

**ESCUELA POLITÉCNICA DEL EJÉRCITO**

**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA,**

**AUTOMATIZACIÓN Y CONTROL**

**PROYECTO DE GRADO PARA LA OBTENCIÓN DEL TÍTULO**

**DE INGENIERÍA**

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE  
ENTRENAMIENTO MODULAR PARA CONTROL LÓGICO Y DE  
PROCESOS”**

**AUTOR:**

**GALO MATIAS ROBAYO GORDÓN**

**SANGOLQUÍ – ECUADOR**

**2008**

## **CERTIFICACIÓN**

Certificamos que el presente proyecto fue realizado en su totalidad por el señor Galo Matias Robayo Gordón, como requisito parcial para la obtención del título en INGENIERÍA ELECTRÓNICA.

---

Sr. Ing. Hugo Ortiz

Director del Proyecto

---

Sr. Ing. Paúl Ayala

Codirector del Proyecto

## **AGRADECIMIENTO**

Agradezco a Magdalena (†) mi madre por la oportunidad de vivir y el ejemplo de vida, a Gustavo mi padre por el apoyo y la confianza incondicional, a Ana (†) mi abuela por las enseñanzas y el cariño sin límites, y a Oderay mi futura esposa por la inspiración y el amor eterno.

De igual manera, agradezco a mi director Ing. Hugo Ortiz mentalizador de este proyecto.

## **DEDICATORIA**

Dedico este trabajo, el cual marca un antes y un después en mi vida, a la persona que siempre estará presente en mi mente y en mi corazón, mi abuela Anita.

## PRÓLOGO

En la actualidad, la industria en el Ecuador esta modernizándose y automatizándose cada vez más y con mayor rapidez para tener mayor competitividad en el mercado.

Debido a esta realidad, el Departamento de Eléctrica y Electrónica de la Escuela Politécnica del Ejército, consciente en la importancia de ampliar y actualizar el conocimiento acerca de la utilización de Controladores Lógicos Programables (PLC's) ha equipado el correspondiente laboratorio con equipos nuevos de la marca Allen Bradley, entre los que se encuentran PLC's de la serie SLC500 y Micrologix, además de las respectivas licencias del software RSLinx, RSLogix500 y RSView32.

La realización de este proyecto consistió en diseñar e implementar un sistema de entrenamiento guiado que permita de manera didáctica ilustrar a los estudiantes acerca de la aplicación de las diferentes prestaciones de los PLC's y del los paquetes de software antes mencionados, en diferentes técnicas de control lógico y control de procesos.

Para control lógico se diseño e implementó siete tableros (módulos) didácticos que permitirán realizar de manera versátil las aplicaciones de simulación propuestas, las cuales están diseñadas sistemáticamente para ilustrar de manera didáctica las diferentes técnicas de control utilizando PLC's Allen Bradley. Adicionalmente para cada aplicación se desarrolló en RSLogix500 la lógica de control correspondiente.

De igual manera se diseño e implementó siete aplicaciones de simulación en hardware y software, de técnicas de control de procesos tales como PID, Control de Razón, Control en Cascada, Control por Sobreposición, entre otras. Incluyendo el desarrollo en RSLogix500 de la lógica de control correspondiente, además de la creación de una Interfase Humano Maquina (HMI) en RSView32 para cada una de estas aplicaciones.

Finalmente, se diseñó e implementó dos aplicaciones que permiten ilustrar de manera didáctica la configuración de sistemas SCADA. Incluyendo el desarrollo de la lógica de control en RSLogix500 y la creación de la HMI en RSView32 para cada una de estas aplicaciones.

## INDICE

<b>CAPITULO 1 .....</b>	<b>1</b>
1.1. ANTECEDENTES .....	1
1.2. JUSTIFICACIÓN E IMPORTANCIA .....	2
1.3. ALCANCE DEL PROYECTO .....	3
 <b>CAPITULO 2 .....</b>	 <b>4</b>
<b>2.1. PLC ALLEN BRADLEY .....</b>	<b>4</b>
2.1.1. SERIE SLC-500* .....	4
INTRODUCCIÓN .....	4
CARACTERÍSTICAS .....	5
Procesador SLC 5/03.....	6
Procesador SLC 5/05.....	7
PROGRAMACIÓN DE SLC-500 .....	8
Archivos del Procesador (Processor Files).....	8
Direccionamiento de los Archivos de Salida y Entrada (O0: y I1:) .....	9
Direccionamiento de los Archivos de Status (S2:).....	9
Direccionamiento de los Archivos de Bit (B3:) .....	9
Direccionamiento de los Archivos de Temporizador (Timer) (T4:) .....	10
Direccionamiento de los Archivos de Contador (C5:) .....	11
Direccionamiento de los Archivos de Control (R6:).....	12
Direccionamiento de los Archivos de Entero (N7:) .....	12
Formato de direccionamiento de los Archivos de Punto Flotante (F8:).....	12
Conjunto de instrucciones para la programación de SLC-500 .....	13
Instrucciones de Entrada y Salida .....	14
Instrucciones de Conteo .....	17
Instrucciones de Temporización.....	19
Instrucciones de Comparación .....	21
Instrucciones de Transferencia de datos.....	22
Instrucciones Matemáticas. ....	23
Reloj Calendario .....	25
Instrucciones de Transferencia de Datos.....	27
Instrucciones Especiales.....	29
Instrucciones de Flujo de Programa .....	31
Instrucción Proporcional Integral Derivativa (PID).....	31
Instrucciones de comunicación .....	38
2.1.2. SERIE MICROLOGIX 1000* .....	42
INTRODUCCIÓN .....	42
CARACTERÍSTICAS .....	43
PROGRAMACIÓN DE MICROLOGIX-1000 .....	44
Archivos del Procesador (Processor Files).....	44
Direccionamiento .....	44
Conjunto de instrucciones para la programación de Micrologix 1000.....	45
 <b>2.2. INTERFACES DE COMUNICACIÓN* .....</b>	 <b>47</b>
2.2.1. INTERFASE DE COMUNICACIÓN ETHERNET .....	48
Módulo de Interfase Ethernet 1761-NET-ENIW .....	48
Configuración del Módulo de Interfase Ethernet 1761-NET-ENIW .....	49
2.2.2. INTERFASE DE COMUNICACIÓN SERIAL RS-232.....	52
2.2.3. INTERFASE DE COMUNICACIÓN SERIAL RS-485.....	53
Módulo Conversor de Interfase Avanzado 1761-NET-AIC.....	54
Modos de Operación .....	55
Configuración de la dirección del nodo de comunicación .....	55

Diagrama de conexión.....	55
<b>CAPITULO 3.....</b>	<b>56</b>
<b>3.1. SOFTWARE RSLINX*.....</b>	<b>56</b>
3.1.1. INTRODUCCIÓN.....	56
3.1.2. REQUERIMIENTOS DEL SISTEMA .....	58
Requerimientos mínimos de Hardware .....	58
Requerimientos mínimos de Software .....	58
3.1.3. CONFIGURACIÓN DE LOS DRIVERS DE COMUNICACIÓN.....	58
Driver RS-232 de comunicación entre el PLC y la PC .....	58
Configuración del Driver Ethernet de comunicación entre el PLC y la PC .....	61
<b>3.2. SOFTWARE RSLOGIX 500*.....</b>	<b>64</b>
3.2.1. INTRODUCCIÓN.....	64
3.2.2. REQUERIMIENTOS DEL SISTEMA .....	65
Requerimientos mínimos de Hardware .....	65
Requerimientos mínimos de Software .....	66
3.2.3. CONFIGURACIÓN Y CREACIÓN DE UN NUEVO ARCHIVO.....	66
Configuración del canal 0 (DF1 Full-duplex) de comunicación del PLC .....	69
Configuración del canal 1 (Ethernet) del PLC SLC 5/05.....	71
<b>3.3. SOFTWARE RSVIEW32*.....</b>	<b>73</b>
3.3.1. INTRODUCCIÓN.....	73
3.3.2. REQUERIMIENTOS DEL SISTEMA .....	75
3.3.3. CONFIGURACIÓN Y CREACIÓN DE UNA HMI .....	77
Crear un archivo nuevo y configurar de los canales de comunicación directa.....	77
Crear Tags.....	81
Crear Pantallas Gráficas.....	82
Animar Objetos .....	84
Crear Trends (Gráficos de tendencia) .....	85
Crear y configurar Eventos .....	86
Correr el Programa.....	89
<b>CAPITULO 4.....</b>	<b>92</b>
<b>4.1. CONTROL LÓGICO 1: INSTRUCCIONES DE BITS 1.....</b>	<b>92</b>
4.1.1. OBJETIVOS.....	92
4.1.2. MATERIALES Y EQUIPOS .....	92
4.1.3. MARCO TEÓRICO .....	92
4.1.4. PLANTEAMIENTO .....	101
4.1.5. PROCEDIMIENTO .....	102
4.1.6. PRUEBAS Y RESULTADOS .....	103
4.1.7. CONCLUSIONES Y RECOMENDACIONES .....	103
<b>4.2. CONTROL LÓGICO 2: INSTRUCCIONES DE BITS 2.....</b>	<b>104</b>
4.2.1. OBJETIVOS.....	104
4.2.2. MATERIALES Y EQUIPOS .....	104
4.2.3. MARCO TEÓRICO .....	104
4.2.4. PLANTEAMIENTO .....	104
4.2.5. PROCEDIMIENTO .....	105
4.2.6. PRUEBAS Y RESULTADOS .....	106
4.2.7. CONCLUSIONES Y RECOMENDACIONES .....	106
<b>4.3. CONTROL LÓGICO 3: CONTADORES.....</b>	<b>107</b>
4.3.1. OBJETIVOS.....	107
4.3.2. MATERIALES Y EQUIPOS .....	107
4.3.3. MARCO TEÓRICO .....	107
4.3.4. PLANTEAMIENTO .....	111

4.3.5.	PROCEDIMIENTO .....	112
4.3.6.	PRUEBAS Y RESULTADOS .....	113
4.3.7.	CONCLUSIONES Y RECOMENDACIONES .....	113
<b>4.4.</b>	<b>CONTROL LÓGICO 4: TEMPORIZADORES 1.....</b>	<b>114</b>
4.4.1.	OBJETIVOS.....	114
4.4.2.	MATERIALES Y EQUIPOS .....	114
4.4.3.	MARCO TEÓRICO .....	114
4.4.4.	PLANTEAMIENTO .....	116
4.4.5.	PROCEDIMIENTO .....	117
4.4.6.	PRUEBAS Y RESULTADOS .....	118
4.4.7.	CONCLUSIONES Y RECOMENDACIONES .....	118
<b>4.5.</b>	<b>CONTROL LÓGICO 5: TEMPORIZADORES 2.....</b>	<b>119</b>
4.5.1.	OBJETIVOS.....	119
4.5.2.	MATERIALES Y EQUIPOS .....	119
4.5.3.	MARCO TEÓRICO .....	119
4.5.4.	PLANTEAMIENTO .....	119
4.5.5.	PROCEDIMIENTO .....	120
4.5.6.	PRUEBAS Y RESULTADOS .....	120
4.5.7.	CONCLUSIONES Y RECOMENDACIONES .....	120
<b>4.6.</b>	<b>CONTROL LÓGICO 6: ENTRADAS Y SALIDAS ANALÓGICAS .....</b>	<b>121</b>
4.6.1.	OBJETIVOS.....	121
4.6.2.	MATERIALES Y EQUIPOS .....	121
4.6.3.	MARCO TEÓRICO .....	121
4.6.4.	PLANTEAMIENTO .....	131
4.6.5.	PROCEDIMIENTO .....	133
4.6.6.	PRUEBAS Y RESULTADOS .....	133
4.6.7.	CONCLUSIONES Y RECOMENDACIONES .....	133
<b>4.7.</b>	<b>CONTROL LÓGICO 7: REGISTROS DE RELOJ/CALENDARIO.....</b>	<b>134</b>
4.7.1.	OBJETIVOS.....	134
4.7.2.	MATERIALES Y EQUIPOS .....	134
4.7.3.	MARCO TEÓRICO .....	134
4.7.4.	PLANTEAMIENTO .....	135
4.7.5.	PROCEDIMIENTO .....	136
4.7.6.	PRUEBAS Y RESULTADOS .....	137
4.7.7.	CONCLUSIONES Y RECOMENDACIONES .....	137
<b>4.8.</b>	<b>CONTROL LÓGICO 8: TRANSFERENCIA DE DATOS .....</b>	<b>138</b>
4.8.1.	OBJETIVOS.....	138
4.8.2.	MATERIALES Y EQUIPOS .....	138
4.8.3.	MARCO TEÓRICO .....	138
4.8.4.	PLANTEAMIENTO .....	142
4.8.5.	PROCEDIMIENTO .....	143
4.8.6.	PRUEBAS Y RESULTADOS .....	144
4.8.7.	CONCLUSIONES Y RECOMENDACIONES .....	144
<b>4.9.</b>	<b>CONTROL LÓGICO 9: FLUJO DE PROGRAMA.....</b>	<b>145</b>
4.9.1.	OBJETIVOS.....	145
4.9.2.	MATERIALES Y EQUIPOS .....	145
4.9.3.	MARCO TEÓRICO .....	145
4.9.4.	PLANTEAMIENTO .....	146
4.9.5.	PROCEDIMIENTO .....	146
4.9.6.	PRUEBAS Y RESULTADOS .....	146
4.9.7.	CONCLUSIONES Y RECOMENDACIONES .....	146
<b>4.10.</b>	<b>CONTROL DE PROCESOS 1: CONTROL ON-OFF CON HISTÉRESIS.....</b>	<b>147</b>
4.10.1.	OBJETIVOS.....	147

4.10.2.	MATERIALES Y EQUIPOS .....	147
4.10.3.	MARCO TEÓRICO .....	147
4.10.4.	PLANTEAMIENTO .....	162
4.10.5.	PROCEDIMIENTO.....	163
4.10.6.	PRUEBAS Y RESULTADOS .....	164
4.10.7.	CONCLUSIONES Y RECOMENDACIONES .....	164
<b>4.11.</b>	<b>CONTROL DE PROCESOS 2: CONTROL PID DE TEMPERATURA.....</b>	<b>165</b>
4.11.1.	OBJETIVOS.....	165
4.11.2.	MATERIALES Y EQUIPOS .....	165
4.11.3.	MARCO TEÓRICO .....	165
4.11.4.	PLANTEAMIENTO .....	174
4.11.5.	PROCEDIMIENTO.....	175
4.11.6.	PRUEBAS Y RESULTADOS .....	176
4.11.7.	CONCLUSIONES Y RECOMENDACIONES .....	176
<b>4.12.</b>	<b>CONTROL DE PROCESOS 3: CONTROL PID DE VELOCIDAD.....</b>	<b>177</b>
4.12.1.	OBJETIVOS.....	177
4.12.2.	MATERIALES Y EQUIPOS .....	177
4.12.3.	MARCO TEÓRICO .....	177
4.12.4.	PLANTEAMIENTO .....	177
4.12.5.	PROCEDIMIENTO.....	179
4.12.6.	PRUEBAS Y RESULTADOS .....	179
4.12.7.	CONCLUSIONES Y RECOMENDACIONES .....	179
<b>4.13.</b>	<b>CONTROL DE PROCESOS 4: CONTROL DE RAZÓN .....</b>	<b>180</b>
4.13.1.	OBJETIVOS.....	180
4.13.2.	MATERIALES Y EQUIPOS .....	180
4.13.3.	MARCO TEÓRICO .....	180
4.13.4.	PLANTEAMIENTO .....	183
4.13.5.	PROCEDIMIENTO.....	184
4.13.6.	PRUEBAS Y RESULTADOS .....	184
4.13.7.	CONCLUSIONES Y RECOMENDACIONES .....	184
<b>4.14.</b>	<b>CONTROL DE PROCESOS 5: CONTROL POR SOBREPOSICIÓN.....</b>	<b>185</b>
4.14.1.	OBJETIVOS.....	185
4.14.2.	MATERIALES Y EQUIPOS .....	185
4.14.3.	MARCO TEÓRICO .....	185
4.14.4.	PLANTEAMIENTO .....	188
4.14.5.	PROCEDIMIENTO.....	190
4.14.6.	PRUEBAS Y RESULTADOS .....	190
4.14.7.	CONCLUSIONES Y RECOMENDACIONES .....	190
<b>4.15.</b>	<b>CONTROL DE PROCESOS 6: CONTROL EN CASCADA .....</b>	<b>191</b>
4.15.1.	OBJETIVOS.....	191
4.15.2.	MATERIALES Y EQUIPOS .....	191
4.15.3.	MARCO TEÓRICO .....	191
4.15.4.	PLANTEAMIENTO .....	194
4.15.5.	PROCEDIMIENTO.....	195
4.15.6.	PRUEBAS Y RESULTADOS .....	196
4.15.7.	CONCLUSIONES Y RECOMENDACIONES .....	196
<b>4.16.</b>	<b>CONTROL DE PROCESOS 7: CONTROL SELECTIVO .....</b>	<b>197</b>
4.16.1.	OBJETIVOS.....	197
4.16.2.	MATERIALES Y EQUIPOS .....	197
4.16.3.	MARCO TEÓRICO .....	197
4.16.4.	PLANTEAMIENTO .....	198
4.16.5.	PROCEDIMIENTO.....	200
4.16.6.	PRUEBAS Y RESULTADOS .....	201
4.16.7.	CONCLUSIONES Y RECOMENDACIONES .....	201

<b>4.17.</b>	<b>CONTROL DE PROCESOS 8: SCADA 1 COMUNICACIÓN SERIAL.....</b>	<b>202</b>
4.17.1.	OBJETIVOS.....	202
4.17.2.	MATERIALES Y EQUIPOS .....	202
4.17.3.	MARCO TEÓRICO .....	202
4.17.4.	PLANTEAMIENTO .....	213
4.17.5.	PROCEDIMIENTO.....	215
4.17.6.	PRUEBAS Y RESULTADOS .....	215
4.17.7.	CONCLUSIONES Y RECOMENDACIONES .....	215
<b>4.18.</b>	<b>CONTROL DE PROCESOS 9: SCADA 2 COMUNICACIÓN ETHERNET .....</b>	<b>216</b>
4.18.1.	OBJETIVOS.....	216
4.18.2.	MATERIALES Y EQUIPOS .....	216
4.18.3.	MARCO TEÓRICO .....	216
4.18.4.	PLANTEAMIENTO .....	231
4.18.5.	PROCEDIMIENTO.....	233
4.18.6.	PRUEBAS Y RESULTADOS .....	234
4.18.7.	CONCLUSIONES Y RECOMENDACIONES .....	234
<b>CAPITULO 5.....</b>	<b>235</b>	
<b>5.1.</b>	<b>CONTROL LÓGICO 1: INSTRUCCIONES DE BITS 1 .....</b>	<b>235</b>
5.1.1.	OBJETIVOS.....	235
5.1.2.	MATERIALES Y EQUIPOS .....	235
5.1.3.	PLANTEAMIENTO .....	235
5.1.4.	DESARROLLO .....	236
5.1.5.	PRUEBAS Y RESULTADOS .....	241
5.1.6.	CONCLUSIONES Y RECOMENDACIONES .....	241
<b>5.2.</b>	<b>CONTROL LÓGICO 2: INSTRUCCIONES DE BITS 2 .....</b>	<b>242</b>
5.2.1.	OBJETIVOS.....	242
5.2.2.	MATERIALES Y EQUIPOS .....	242
5.2.3.	PLANTEAMIENTO .....	242
5.2.4.	DESARROLLO .....	243
5.2.5.	PRUEBAS Y RESULTADOS .....	247
5.2.6.	CONCLUSIONES Y RECOMENDACIONES .....	247
<b>5.3.</b>	<b>CONTROL LÓGICO 3: CONTADORES.....</b>	<b>248</b>
5.3.1.	OBJETIVOS.....	248
5.3.2.	MATERIALES Y EQUIPOS .....	248
5.3.3.	PLANTEAMIENTO .....	248
5.3.4.	DESARROLLO .....	250
5.3.5.	PRUEBAS Y RESULTADOS .....	254
5.3.6.	CONCLUSIONES Y RECOMENDACIONES .....	254
<b>5.4.</b>	<b>CONTROL LÓGICO 4: TEMPORIZADORES 1.....</b>	<b>255</b>
5.4.1.	OBJETIVOS.....	255
5.4.2.	MATERIALES Y EQUIPOS .....	255
5.4.3.	PLANTEAMIENTO .....	255
5.4.4.	DESARROLLO .....	256
5.4.5.	PRUEBAS Y RESULTADOS .....	262
5.4.6.	CONCLUSIONES Y RECOMENDACIONES .....	262
<b>5.5.</b>	<b>CONTROL LÓGICO 5: TEMPORIZADORES 2.....</b>	<b>263</b>
5.5.1.	OBJETIVOS.....	263
5.5.2.	MATERIALES Y EQUIPOS .....	263
5.5.3.	PLANTEAMIENTO .....	263
5.5.4.	DESARROLLO .....	264
5.5.5.	PRUEBAS Y RESULTADOS .....	268
5.5.6.	CONCLUSIONES Y RECOMENDACIONES .....	268

<b>5.6.</b>	<b>CONTROL LÓGICO 6: ENTRADAS Y SALIDAS ANALÓGICAS .....</b>	<b>269</b>
5.6.1.	OBJETIVOS.....	269
5.6.2.	MATERIALES Y EQUIPOS .....	269
5.6.3.	PLANTEAMIENTO .....	269
5.6.4.	DESARROLLO .....	270
5.6.5.	PRUEBAS Y RESULTADOS .....	275
5.6.6.	CONCLUSIONES Y RECOMENDACIONES .....	275
<b>5.7.</b>	<b>CONTROL LÓGICO 7: REGISTROS DE RELOJ/CALENDARIO.....</b>	<b>276</b>
5.7.1.	OBJETIVOS.....	276
5.7.2.	MATERIALES Y EQUIPOS .....	276
5.7.3.	PLANTEAMIENTO .....	276
5.7.4.	DESARROLLO .....	277
5.7.5.	PRUEBAS Y RESULTADOS .....	280
5.7.6.	CONCLUSIONES Y RECOMENDACIONES .....	281
<b>5.8.</b>	<b>CONTROL LÓGICO 8: TRANSFERENCIA DE DATOS .....</b>	<b>282</b>
5.8.1.	OBJETIVOS.....	282
5.8.2.	MATERIALES Y EQUIPOS .....	282
5.8.3.	PLANTEAMIENTO .....	282
5.8.4.	DESARROLLO .....	283
5.8.5.	PRUEBAS Y RESULTADOS .....	287
5.8.6.	CONCLUSIONES Y RECOMENDACIONES .....	287
<b>5.9.</b>	<b>CONTROL LÓGICO 9: FLUJO DE PROGRAMA.....</b>	<b>288</b>
5.9.1.	OBJETIVOS.....	288
5.9.2.	MATERIALES Y EQUIPOS .....	288
5.9.3.	PLANTEAMIENTO .....	288
5.9.4.	DESARROLLO .....	288
5.9.5.	PRUEBAS Y RESULTADOS .....	293
5.9.6.	CONCLUSIONES Y RECOMENDACIONES .....	293
<b>5.10.</b>	<b>CONTROL DE PROCESOS 1: CONTROL ON-OFF CON HISTERÉISIS.....</b>	<b>294</b>
5.10.1.	OBJETIVOS.....	294
5.10.2.	MATERIALES Y EQUIPOS .....	294
5.10.3.	PLANTEAMIENTO .....	294
5.10.4.	DESARROLLO.....	295
5.10.5.	PRUEBAS Y RESULTADOS .....	304
5.10.6.	CONCLUSIONES Y RECOMENDACIONES .....	305
<b>5.11.</b>	<b>CONTROL DE PROCESOS 2: CONTROL PID DE TEMPERATURA.....</b>	<b>306</b>
5.11.1.	OBJETIVOS.....	306
5.11.2.	MATERIALES Y EQUIPOS .....	306
5.11.3.	PLANTEAMIENTO .....	306
5.11.4.	DESARROLLO.....	308
5.11.5.	PRUEBAS Y RESULTADOS .....	322
5.11.6.	CONCLUSIONES Y RECOMENDACIONES .....	322
<b>5.12.</b>	<b>CONTROL DE PROCESOS 3: CONTROL PID DE VELOCIDAD.....</b>	<b>324</b>
5.12.1.	OBJETIVOS.....	324
5.12.2.	MATERIALES Y EQUIPOS .....	324
5.12.3.	PLANTEAMIENTO .....	324
5.12.4.	DESARROLLO.....	325
5.12.5.	PRUEBAS Y RESULTADOS .....	336
5.12.6.	CONCLUSIONES Y RECOMENDACIONES .....	337
<b>5.13.</b>	<b>CONTROL DE PROCESOS 4: CONTROL DE RAZÓN .....</b>	<b>338</b>
5.13.1.	OBJETIVOS.....	338
5.13.2.	MATERIALES Y EQUIPOS .....	338
5.13.3.	PLANTEAMIENTO .....	338

5.13.4.	DESARROLLO.....	339
5.13.5.	PRUEBAS Y RESULTADOS .....	348
5.13.6.	CONCLUSIONES Y RECOMENDACIONES .....	348
<b>5.14.</b>	<b>CONTROL DE PROCESOS 5: CONTROL POR SOBREPOSICIÓN.....</b>	<b>349</b>
5.14.1.	OBJETIVOS.....	349
5.14.2.	MATERIALES Y EQUIPOS .....	349
5.14.3.	PLANTEAMIENTO .....	349
5.14.4.	DESARROLLO.....	351
5.14.5.	PRUEBAS Y RESULTADOS .....	362
5.14.6.	CONCLUSIONES Y RECOMENDACIONES .....	362
<b>5.15.</b>	<b>CONTROL DE PROCESOS 6: CONTROL EN CASCADA .....</b>	<b>363</b>
5.15.1.	OBJETIVOS.....	363
5.15.2.	MATERIALES Y EQUIPOS .....	363
5.15.3.	PLANTEAMIENTO .....	363
5.15.4.	DESARROLLO.....	365
5.15.5.	PRUEBAS Y RESULTADOS .....	373
5.15.6.	CONCLUSIONES Y RECOMENDACIONES .....	374
<b>5.16.</b>	<b>CONTROL DE PROCESOS 7: CONTROL SELECTIVO .....</b>	<b>375</b>
5.16.1.	OBJETIVOS.....	375
5.16.2.	MATERIALES Y EQUIPOS .....	375
5.16.3.	PLANTEAMIENTO .....	375
5.16.4.	DESARROLLO.....	377
5.16.5.	PRUEBAS Y RESULTADOS .....	386
5.16.6.	CONCLUSIONES Y RECOMENDACIONES .....	387
<b>5.17.</b>	<b>CONTROL DE PROCESOS 8: SCADA 1 COMUNICACIÓN SERIAL.....</b>	<b>388</b>
5.17.1.	OBJETIVOS.....	388
5.17.2.	MATERIALES Y EQUIPOS .....	388
5.17.3.	PLANTEAMIENTO .....	388
5.17.4.	DESARROLLO.....	389
5.17.5.	PRUEBAS Y RESULTADOS .....	398
5.17.6.	CONCLUSIONES Y RECOMENDACIONES .....	398
<b>5.18.</b>	<b>CONTROL DE PROCESOS 9: SCADA 2 COMUNICACIÓN ETHERNET .....</b>	<b>399</b>
5.18.1.	OBJETIVOS.....	399
5.18.2.	MATERIALES Y EQUIPOS .....	399
5.18.3.	PLANTEAMIENTO .....	399
5.18.4.	DESARROLLO.....	402
5.18.5.	PRUEBAS Y RESULTADOS .....	421
5.18.6.	CONCLUSIONES Y RECOMENDACIONES .....	421
<b>CAPITULO 6.....</b>	<b>423</b>	
<b>6.1.</b>	<b>CONTROL LÓGICO 1: INSTRUCCIONES DE BITS 1 .....</b>	<b>423</b>
<b>6.2.</b>	<b>CONTROL LÓGICO 2: INSTRUCCIONES DE BITS 2 .....</b>	<b>423</b>
<b>6.3.</b>	<b>CONTROL LÓGICO 3: CONTADORES.....</b>	<b>423</b>
<b>6.4.</b>	<b>CONTROL LÓGICO 4: TEMPORIZADORES 1.....</b>	<b>424</b>
<b>6.5.</b>	<b>CONTROL LÓGICO 5: TEMPORIZADORES 2.....</b>	<b>424</b>
<b>6.6.</b>	<b>CONTROL LÓGICO 6: ENTRADAS Y SALIDAS ANALÓGICAS.....</b>	<b>424</b>
<b>6.7.</b>	<b>CONTROL LÓGICO 7: REGISTROS DE RELOJ/CALENDARIO.....</b>	<b>425</b>

6.8.	CONTROL LÓGICO 8: TRANSFERENCIA DE DATOS .....	425
6.9.	CONTROL LÓGICO 9: FLUJO DE PROGRAMA.....	425
6.10.	CONTROL DE PROCESOS 1: CONTROL ON-OFF CON HISTÉRESIS.....	425
6.11.	CONTROL DE PROCESOS 2: CONTROL PID DE TEMPERATURA.....	426
6.12.	CONTROL DE PROCESOS 3: CONTROL PID DE VELOCIDAD.....	426
6.13.	CONTROL DE PROCESOS 4: CONTROL DE RAZÓN .....	426
6.14.	CONTROL DE PROCESOS 5: CONTROL POR SOBREPOSICIÓN.....	427
6.15.	CONTROL DE PROCESOS 6: CONTROL EN CASCADA .....	427
6.16.	CONTROL DE PROCESOS 7: CONTROL SELECTIVO .....	427
6.17.	CONTROL DE PROCESOS 8: SCADA 1 COMUNICACIÓN SERIAL.....	427
6.18.	CONTROL DE PROCESOS 9: SCADA 2 COMUNICACIÓN ETHERNET .....	428
<b>CAPITULO 7 .....</b>		<b>429</b>
7.1.	CONCLUSIONES.....	429
7.2.	RECOMENDACIONES.....	430
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>		<b>431</b>
<b>ANEXOS .....</b>		<b>433</b>

# CAPITULO 1

## INTRODUCCIÓN

### 1.1. ANTECEDENTES

Debido a las necesidades de la industria y al acelerado desarrollo tecnológico, se ha difundido mundialmente el uso de los Controladores lógicos programables (*PLC por sus siglas en inglés*) para realizar el control de diferentes procesos en donde el coste de desarrollo y mantenimiento de un sistema de automatización tradicional (basados en circuitos eléctricos con relés, interruptores, contadores, entre otros) es relativamente alto contra el coste de la automatización mediante PLC's, o en procesos en los cuales van a existir cambios en el sistema durante toda su vida operacional, puesto que estos controladores presentan optimas características de confiabilidad, flexibilidad y seguridad (*Estrada, 2006*).

Una de las marcas de PLC's más representativas en la industria tanto a nivel internacional como local, es Allen Bradley, siendo de gran representación las series SLC 500 y Micrologix dentro de los PLC modulares y compactos respectivamente.

Estos equipos ofrecen una excelente relación costo beneficio con amplias capacidades para satisfacer las necesidades de una diversa gama de aplicaciones (*Rockwell Automation 1747-SG001B-ES-P, 2005*).

Además estos PLC's utilizan el software de programación RSLogix 500 que maximiza la productividad simplificando el desarrollo de programas y la resolución de problemas.

Así mismo, puesto que la mayoría de las aplicaciones hoy en día ya no pueden considerarse aplicaciones aisladas en el proceso global, es necesario supervisar y monitorear las distintas variables que intervienen en el proceso. (*Rockwell Automation RSVIEW-BR001A-ES-P, 2005*) Para lo cual se puede emplear la comunicación hacia un computador personal con software dedicado como es RSView32, que consiste en un paquete de software que permite la creación de Interfaces Humano Maquina (HMI).

Por las razones expuestas, en el Laboratorio de Robótica y PLC's del Departamento de Eléctrica y Electrónica de la ESCUELA POLITECNICA DEL EJERCITO se ha iniciado

un necesario proceso de renovación de su infraestructura habiéndose adquirido en primera instancia los equipos y software mencionados con la finalidad de incluirlos como parte de la formación integral de los estudiantes de la carrera de Ingeniería en Electrónica, Automatización y Control.

## **1.2. JUSTIFICACIÓN E IMPORTANCIA**

En la actualidad, la industria en el Ecuador esta modernizándose y automatizándose cada vez más y con mayor rapidez para tener mayor competitividad en el mercado.

Dentro de este proceso de automatización, los PLC's juegan un papel fundamental, debido a sus ya conocidas prestaciones que facilitan la automatización de ciertos procesos.

En nuestro país se emplean PLC's de diversas marcas y modelos, siendo la marca Allen Bradley una de las más representativas y de mayor auge debido a sus excelente calidad y precio.

Debido a esta realidad, el Departamento de Eléctrica y Electrónica de la Escuela Politécnica del Ejército, consciente en la importancia de ampliar y actualizar el conocimiento acerca de la utilización de PLC's ha equipado el correspondiente laboratorio con equipos nuevos de la marca Allen Bradley, entre los que se encuentran PLC's de la serie SLC500 y Micrologix, además de las respectivas licencias del software RSLogix500 y RSView32.

A partir de este hecho surge la inherente necesidad de contar con un sistema de entrenamiento modular guiado que facilite el aprendizaje de los estudiantes acerca de la configuración, programación y aplicación de estos nuevos equipos; que establezca una serie de prácticas específicas que de manera didáctica y metodológica sintetizen y direccionen la información proporcionada por la casa fabricante para que de manera más sencilla pueda ser asimilada por los estudiantes de la carrera, dado que al contar en el laboratorio tanto con equipos Allen Bradley como Modicom se requiere agilizar y sintetizar la ejecución de las prácticas para poder tener una adecuada distribución del tiempo que facilite el estudio de ambas marcas de manera simultánea.

De igual manera se requiere la implementación de tableros que permitan realizar la simulación en hardware de las prácticas de control lógico, lo que ayudará a los estudiantes

a adquirir los conocimientos necesarios, de mejor manera y en menor tiempo. Adicionalmente, se complementarán también estos conocimientos con simulaciones en software.

La principal motivación para la realización de este proyecto, es la de satisfacer las necesidades nombradas, para contribuir al continuo mejoramiento de la educación integral de los alumnos.

### **1.3. ALCANCE DEL PROYECTO**

Se va a diseñar e implementar siete tableros (módulos) didácticos que permitirán realizar de manera versátil aplicaciones de simulación orientadas a ilustrar de manera didáctica diferentes técnicas de control lógico utilizando PLC's Allen Bradley. Para cada aplicación se desarrollará en RSLogix500 la lógica de control correspondiente.

De igual manera se implementarán siete aplicaciones de simulación en hardware y software, de técnicas de control de procesos tales como PID, Control de Razón, Control en Cascada, Control por Sobreposición, entre otras. Cada una de estas aplicaciones incluirá el desarrollo en RSLogix500 de la lógica de control correspondiente, además de la creación de una Interfase Humano Maquina (HMI) en RSView32.

Adicionalmente se implementarán aplicaciones que permitan ilustrar de manera didáctica la configuración de sistemas SCADA, cada una de las cuales incluirá el desarrollo de la lógica de control en RSLogix500 y la creación de la HMI en RSView32.

## CAPITULO 2

### PLC's ALLEN BRADLEY SLC-500 Y MICROLOGIX 1000

#### 2.1. PLC ALLEN BRADLEY

##### 2.1.1. SERIE SLC-500\*

#### INTRODUCCIÓN



**Figura. 2.1. Equipos de la serie SLC-500**

El SLC 500 de Allen-Bradley es una familia de controladores programables de chasis pequeño, de E/S discretas, analógicas y especiales y dispositivos periféricos. Esta familia de procesadores proporciona eficacia y flexibilidad a través de una amplia gama de configuraciones de comunicaciones, características y opciones de memoria. La incorporación de redes incorporadas, ampliaciones de la familia de E/S 1746, disponibilidad de módulos de E/S especiales y un excelente paquete de programación para Windows, el SLC 500 de Allen-Bradley se ha convertido en el principal controlador de pequeño tamaño del mercado.

---

\*La información expuesta en esta sección fue extraída de los textos 3, 4, 5, 7, 8 y 9 citados en las referencias bibliográficas.

## CARACTERÍSTICAS

Las siguientes especificaciones se aplican a todos los componentes modulares SLC 500 a menos que se indique lo contrario.

DESCRIPCIÓN	ESPECIFICACIÓN
Temperatura	En funcionamiento: de 0 °C a +60 °C (de +32 °F a +140 °F)
	Almacenamiento: de -40 °C a +85 °C (de -40 °F a +185 °F)
Humedad:	de 5 a 95 % sin condensación
Vibraciones	En funcionamiento: de 1.0 G a 5 hasta 2000 Hz
	Sin funcionar: de 2.5 G a 5 hasta 2000 Hz
Choque	En funcionamiento: (todos los módulos salvo contacto de relés) 30.0 G (3 impulsos, 11 ms)
	En funcionamiento: (módulos de contacto de relés 1746-OWx y 1746-IOx combo) 10.0 G (3 impulsos, 11 ms)
	Sin funcionar: 50.0 G (3 impulsos, 11 ms)
Caída libre (prueba de caída)	Portátil, 2.268 Kg. (5 libras) o menos a 0.762 m (30 pulg.) (seis caídas)
	Portátil, 2.268 Kg. (5 libras) o mas a 0.1016 m (4 pulg.) (tres caídas en plano)
Inmunidad a ruidos	Norma NEMA ICS 2-230
Compatibilidad electromagnética	Arco eléctrico: 1.5 KV (norma industrial: NEMA ICS 2-230/NEMA ICS 3-304)
	Capacidad para soportar sobretensiones: 3 KV (norma industrial: Norma IEEE 472- 1974/ANSI C37.90/ 90A-1974)
	Ráfaga de fenómeno transitorio rápido (impulso): 2 KV para fuentes de alimentación 1746, 1 KV para 1746 I/O y líneas de comunicaciones de mas de 10 m (32.84 pies), 5 ns de tiempo de elevación
	Descarga electrostática (ESD): 15 KV, 100 pF/modelo de 1.5 KW
	Susceptibilidad electromagnética radiada: walkie-talkie de 5 W a 464.5 MHz y 153.05 MHz
Seguridad	Capacidad nominal dieléctrica: 1500 VCA (norma industrial: UL 508, CSA C22.2 N° 142)

	Aislamiento entre circuitos de comunicación: 500 VCC
	Aislamiento entre backplane y E/S: 1500 VCA
	Inflamabilidad e incendio eléctrico: UL94V-0
Certificaciones	<ul style="list-style-type: none"> <li>▪ listado en CUL</li> <li>▪ Clase I, Grupos A, B, C o D, División 2</li> <li>▪ acorde con la normativa de la CE establecida por las pertinentes directivas</li> <li>▪ certificación de CSA</li> </ul>

**Tabla. 2.1. Especificaciones SLC500**

La Escuela Politécnica de Ejército (ESPE) cuenta con PLC's de las series SLC 5/03 y SLC 5/05 de esta familia, a continuación se describen las características más importantes de estas dos series.

### **Procesador SLC 5/03**

El procesador 1747-L532E 5/03 CPU 16K-Mem OS-302 Series C con el que cuenta la ESPE, posee las siguientes características:

- Tamaño de memoria total de 16 K.
- Control de hasta 4096 puntos de entrada y salida.
- Programación en línea (incluye edición en tiempo de ejecución).
- Canal DH-485 incorporado.
- Canal RS-232 incorporado compatible con full-duplex DF1, maestro/esclavo half-duplex DF1 para SCADA, DH-485 usando un 1761-NET-AIC con cable 1747-CP3 y protocolos ASCII.
- Reloj/calendario en tiempo real incorporado.
- 2 ms de interrupción temporizada seleccionable (STI).
- 0.50 ms de interrupción de entrada discreta (DII).
- Funciones matemáticas avanzadas: trigonométricas, PID, exponenciales, punto flotante (coma flotante) e instrucciones de calculo.
- Direccionamiento indirecto.
- El PROM de la memoria flash proporciona actualizaciones de firmware sin cambiar EPROMS físicamente.

- Módulo de memoria flash EPROM opcional disponible.
- Interruptor de llave: RUN, REMote, PROGram.
- RAM con batería de respaldo.

### **Procesador SLC 5/05**

El procesador 1747-L551 5/05 CPU 16K-Mem OS-501 Series C con el que cuenta la ESPE, posee las siguientes características:

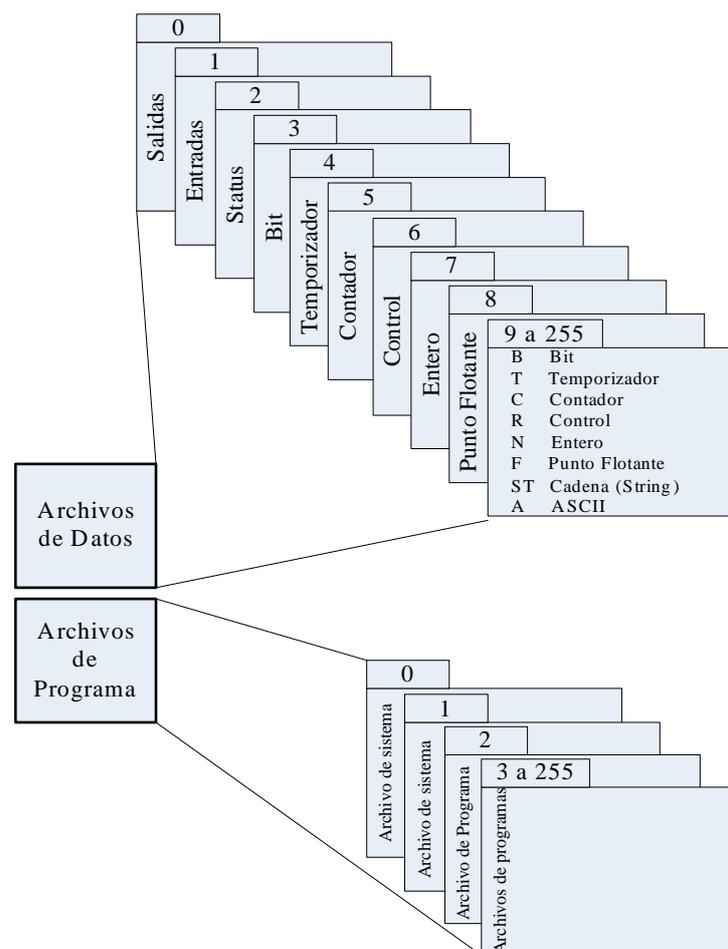
- Tamaños de memoria de programación de 16 K.
- Rendimiento de alta velocidad: 0.90 ms/K es lo típico.
- Control de hasta 4096 puntos de entrada y salida.
- Programación en línea (incluye edición en tiempo de ejecución).
- Canal Ethernet 10Base-T incorporado, que acepta:
  - Comunicación de alta velocidad entre computadoras mediante TCP/ IP
  - Capacidades de comunicación de mensajes con procesadores SLC 5/05, PLC-5 y PLC-5/250, módulo de interfase Ethernet 1785-ENET y puente Ethernet 1756-ENET, así como otros módulos Ethernet para computadoras disponibles comercialmente.
  - SNMP para administración de redes Ethernet estándar
- Canal RS-232 incorporado que acepta full-duplex DF1, maestro/esclavo half-duplex DF1 para SCADA, DH-485 usando un 1761-NET-AIC con cable 1747-CP3 y ASCII.
- Función de paso "passthrough" entre canales de Ethernet a DH-485.
- Función de paso "passthrough" entre canales de Ethernet a DF1.
- Función de paso "passthrough" de E/S remota desde el canal 0 (DF1 o DH485) o el canal 1 (Ethernet) mediante un módulo escáner de E/S remota 1747-SN o 1747-BSN.
- Reloj/calendario en tiempo real incorporado.
- Interrupción temporizada seleccionable (STI) de 1 ms.
- Interrupción de entrada discreta (DII) de 0.50 ms.

- Funciones matemáticas avanzadas: trigonométricas, PID, exponenciales, punto flotante (coma flotante) y las instrucciones de calculo.
- Direccionamiento indirecto.
- El PROM de la memoria flash proporciona actualizaciones de firmware sin cambiar EPROMS físicamente.
- Módulo de memoria flash EPROM opcional disponible.
- Interruptor de llave: RUN, REMote, PROGram (borrado de fallos).
- RAM con batería de respaldo.

## PROGRAMACIÓN DE SLC-500

### Archivos del Procesador (Processor Files)

La memoria de usuario de SLC 500 está compuesta por Archivos de Datos y Archivos de Programa como se muestra en la figura 2.2



**Figura. 2.2. Archivos del Procesador SLC-500**

### Direccionamiento de los Archivos de Salida y Entrada (O0: y I1:)

Los archivos de datos 0 y 1 representan salidas y entradas externas respectivamente.

<i>Formato</i>	<i>Explicación</i>		
<b>O:e.s/b</b>	<b>O</b>	Salida (Output)	
	<b>I</b>	Entrada (Input)	
<b>I:e.s/b</b>	<b>:</b>	Delimitador de elemento	
	<b>e</b>	Número de Slot (en el que se encuentra el módulo de I/O)	El módulo procesador (CPU) es el Slot 0.
	<b>.</b>	Delimitador de palabra	Requerido únicamente si es necesario un número de palabra.
	<b>s</b>	Número de palabra	Requerido si el número de entradas o salidas para el slot excede de 16 (módulos especiales).
	<b>/</b>	Delimitador de bit	
	<b>b</b>	Número de Terminal	Entradas: 0-15  Salidas: 0-15

Tabla. 2.2. Direccionamiento I/O

### Direccionamiento de los Archivos de Status (S2:)

<i>Formato</i>	<i>Explicación</i>		
<b>S:e/b</b>	<b>S</b>	Archivo de Status	
	<b>:</b>	Delimitador de elemento	
	<b>e</b>	Número de elemento	Rango de 0 a 82. Estos son elementos de 1 palabra de longitud y 16 bit por elemento.
	<b>/</b>	Delimitador de bit	
	<b>b</b>	Número de Bit.	Número de Bit dentro del elemento. 0 a 15.

Tabla. 2.3. Direccionamiento STATUS

### Direccionamiento de los Archivos de Bit (B3:)

<i>Formato</i>	<i>Explicación</i>

<b>Bf:e/b</b>	<b>B</b>	Bit	
	<b>f</b>	Número de archivo. Es 3 por defecto. En los SLC puede asignarse un número entre 9 y 255 para almacenamiento adicional.	
	<b>:</b>	Delimitador de elemento	
	<b>e</b>	Número de elemento	Estos son elementos de 1 palabra de longitud y 16 bit por elemento. El rango es entre 0-255.
	<b>/</b>	Delimitador de bit	
	<b>b</b>	Número de Bit	Número de Bit dentro del elemento. 0 a 15.

Tabla. 2.4. Direccionamiento BITS

### Direccionamiento de los Archivos de Temporizador (Timer) (T4:)

<i>Formato</i>	<i>Explicación</i>		
<b>Tf:e.s/b</b>	<b>T</b>	Temporizador (Timer)	
	<b>f</b>	Número de archivo. Es 4 por defecto. En los SLC puede asignarse un número entre 9 y 255 para almacenamiento adicional.	
	<b>:</b>	Delimitador de elemento	
	<b>e</b>	Número de elemento	Estos son elementos de tres palabras cada uno, que conforman el temporizador. El rango es entre 0-255.
	<b>.</b>	Delimitador de palabra	Requerido únicamente si es necesario un número de palabra.
	<b>s</b>	Número de palabra	De 0 a 2 (Tres palabras)
	<b>/</b>	Delimitador de bit	
	<b>b</b>	Número de Bit	De 0 a 15 (16 Bits)

Tabla. 2.5. Direccionamiento TIMER

<b>Ejemplos:</b>	
T4:0/15 or T4:0/EN	Enable bit
T4:0/14 or T4:0/TT	Timer timing bit
T4:0/13 or T4:0/DN	Done bit
T4:0.1 or T4:0.PRE	Valor Preset del timer
T4:0.2 or T4:0.ACC	Valor Acumulado del temporizador

T4:0.1/0 or T4:0.PRE/0	Bit 0 del valor PRESET
T4:0.2/0 or T4:0.ACC/0	Bit 0 del valor Acumulado

**Tabla. 2.6. Ejemplos Direccionamiento TIMER**

### Direccionamiento de los Archivos de Contador (C5:)

<i>Formato</i>	<i>Explicación</i>		
<b>Cf:e.s/b</b>	<b>C</b>	Contador (Counter)	
	<b>f</b>	Número de archivo. Es 5 por defecto. En los SLC puede asignarse un número entre 9 y 255 para almacenamiento adicional.	
	<b>:</b>	Delimitador de elemento	
	<b>e</b>	Número de elemento	Estos son elementos de tres palabras cada uno, que conforman el contador. El rango es entre 0-255.
	<b>.</b>	Delimitador de palabra	Requerido únicamente si es necesario un número de palabra.
	<b>s</b>	Número de palabra	De 0 a 2 (Tres palabras)
	<b>/</b>	Delimitador de bit	
	<b>b</b>	Número de Bit	De 0 a 15 (16 Bits)

**Tabla. 2.7. Direccionamiento COUNTER**

<b>Ejemplos:</b>	
C5:0/15 ó C5:0/CU	Count Up Enable
C5:0/14 ó C5:0/CD	Count Down Enable
C5:0/13 ó C5:0/DN	Done (DN) bit
C5:0/12 ó C5:0/OV	Overflow bit
C5:0/11 ó C5:0/UN	Underflow bit
C5:0.1 ó C5:0.PRE	Valor PRESET del contador
C5:0.2 ó C5:0.ACC	Valor acumulado del contador
C5:0.1/0 ó C5:0.PRE/0	Bit 0 del valor PRESET
C5:0.2/0 ó C5:0.ACC/0	Bit 0 del valor acumulado

**Tabla. 2.8. Ejemplo Direccionamiento COUNTER**

**Direccionamiento de los Archivos de Control (R6:)**

<i>Formato</i>	<i>Explicación</i>		
<b>Rf:e.s/b</b>	<b>R</b>	Control	
	<b>f</b>	Número de archivo. Es 6 por defecto. En los SLC puede asignarse un número entre 9 y 255 para almacenamiento adicional.	
	<b>:</b>	Delimitador de elemento	
	<b>e</b>	Número de elemento	Estos son elementos de tres palabras cada uno. El rango es entre 0-255.
	<b>.</b>	Delimitador de palabra	Requerido únicamente si es necesario un número de palabra.
	<b>s</b>	Número de palabra	
	<b>/</b>	Delimitador de bit	
	<b>b</b>	Número de Bit	De 0 a 15 (16 Bits)

Tabla. 2.9. Direccionamiento CONTROL

**Direccionamiento de los Archivos de Entero (N7:)**

<i>Formato</i>	<i>Explicación</i>		
<b>Nf:e/b</b>	<b>N</b>	Entero	
	<b>f</b>	Número de archivo. Es 7 por defecto. En los SLC puede asignarse un número entre 9 y 255 para almacenamiento adicional.	
	<b>:</b>	Delimitador de elemento	
	<b>e</b>	Número de elemento	Estos son elementos de 1 palabra de longitud y 16 bit por elemento. El rango es entre 0-255.
	<b>/</b>	Delimitador de bit	
	<b>b</b>	Número de Bit	Número de Bit dentro del elemento. 0 a 15.

Tabla. 2.10. Direccionamiento ENTERO

**Formato de direccionamiento de los Archivos de Punto Flotante (F8:)**

<i>Formato</i>	<i>Explicación</i>	
<b>Ff:e</b>	<b>F</b>	Punto Flotante
	<b>f</b>	Número de archivo. Es 8 por defecto. En los SLC puede asignarse un

		número entre 9 y 255 para almacenamiento adicional.
	:	Delimitador de elemento
	e	Número de elemento de Estos son elementos de 2 palabra de longitud y 32 bit por elemento. El rango es entre 0-255.

**Tabla. 2.11. Direccionamiento FLOTANTE**

### Conjunto de instrucciones para la programación de SLC-500

Grupo Funcional	Descripción
Tipo de relé (bit)	Las instrucciones de tipo de relé (bit) monitorean y controlan el estado de los bits. XIC, XIO, OTE, OTL, OTU, OSR
Temporizador y contador	Las instrucciones de temporizador y contador controlan operaciones basadas en tiempo o en el número de eventos. TON, TOF, CTU, CTD, RTO, RES, RHC, TDF
Comparación	Las instrucciones de comparación comparan valores usando una expresión o una instrucción de comparación específica. EQU, NEQ, LES, LEQ, GRT, GEQ, MEQ, LIM.
Cálculo	Las instrucciones de cálculo evalúan operaciones aritméticas usando una expresión o una instrucción aritmética específica. ADD, SUB, MUL, DIV, DDV, CLR, NEG, SQR, SCL, SCP, ABS, CPT, SWP, ASN, ACS, ATN, COS, LN, LOG, SIN, TAN, XPY, RMP.
Lógica	Las instrucciones lógicas realizan operaciones lógicas en los bits. AND, OR, XOR, NOT.
Conversión	Las instrucciones de conversión realizan conversiones entre valores enteros y BCD y entre valores en radianes y grados. TOD, FRD, DCD, DEG, RAD, ENC.
Transferencia	Las instrucciones de transferencia modifican y transfieren los bits. MOV, MVM.
Archivo	Las instrucciones de archivo realizan operaciones en los datos de archivo. COP, FLL, BSL, BSR, FFL, FFU, LFL, LFU, FBC, DDT.
Secuenciador	Las instrucciones de secuenciador monitorean operaciones continuas y repetibles. SQO, SQC, SQL.

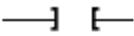
Control de programa	Las instrucciones de flujo de programa cambian el flujo de la ejecución del programa de lógica de escalera. JMP, LBL, JSR, SBR, RET, MCR, TND, SUS, IIM, IOM, END, REF
Interrupción de usuario	Las instrucciones de interrupción del usuario permiten interrumpir el programa basándose en eventos definidos. STD, STE, STS, IID, IIE, RPI, INT
Control de proceso	La instrucción de control de proceso proporciona un control de lazo cerrado. PID
Comunicaciones	Las instrucciones de comunicaciones leen o escriben datos en otra estación. MSG, SVC, BTR, BTW
ASCII	Las instrucciones ASCII leen, escriben, comparan y convierten cadenas ASCII. ABL, ACB, ACI, ACL, ACN, AEX, AHL, AIC, ARD, ARL, ASC, ASR, AWA, AWT

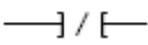
**Tabla. 2.12. Conjunto de instrucciones SLC-500**

A continuación se describen cada una de las principales instrucciones de programación de la familia SLC-500

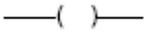
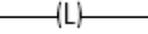
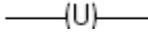
### Instrucciones de Entrada y Salida

Existen varias instrucciones que permiten manipular las entradas o salidas del PLC. Estas pertenecen al grupo de instrucciones conocidas como *INSTRUCCIONES DE BITS*, puesto que actúan sobre un solo bit. En las siguientes tablas se explican las principales:

<i>Instrucciones de Entrada</i>								
SÍMBOLO	NOMBRE	DESCRIPCIÓN						
	Examine if Closed (XIC)	Esta instrucción cambia su estado de acuerdo al estado del bit que representa, de acuerdo a la siguiente tabla: <table border="1" data-bbox="730 1823 1010 1989"> <thead> <tr> <th><i>Bit</i></th> <th><i>XIC</i></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Falso</td> </tr> <tr> <td>1</td> <td>Verdadero</td> </tr> </tbody> </table>	<i>Bit</i>	<i>XIC</i>	0	Falso	1	Verdadero
<i>Bit</i>	<i>XIC</i>							
0	Falso							
1	Verdadero							

	Examine if Open (XIO)	Esta instrucción cambia su estado de acuerdo al estado del bit que representa, de acuerdo a la siguiente tabla: <table border="1" data-bbox="724 338 1003 506"> <thead> <tr> <th><i>Bit</i></th> <th><i>XIO</i></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Verdadero</td> </tr> <tr> <td>1</td> <td>Falso</td> </tr> </tbody> </table>	<i>Bit</i>	<i>XIO</i>	0	Verdadero	1	Falso
<i>Bit</i>	<i>XIO</i>							
0	Verdadero							
1	Falso							
	One-shot Rising (OSR)	Cuando las condiciones que preceden a esta instrucción cambian de Falso a Verdadero, esta se vuelve verdadera durante un scan, luego se vuelve falsa a pesar que las condiciones que la preceden sigan siendo verdaderas. Es decir solo se vuelve verdadera durante un scan, cuando existe una transición Falso-Verdadero de las condiciones que la preceden.						

**Tabla. 2.13. Instrucciones de Entrada**

<i>Instrucciones de Salida</i>		
<b>SÍMBOLO</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>
	Output Energize (OTE)	Se utiliza esta instrucción para encender el bit que representa (salida) cuando las condiciones que la preceden son verdaderas.
	Output Latch (OTL)	Esta instrucción es similar a la anterior (OTE) con la diferencia que una vez que la condiciones que la preceden cambian de Falso a Verdadero, ésta activa el bit y éste queda activado sin importar que las condiciones que la preceden cambien de estado. Es decir se queda enclavado.
	Output Unlatch (OTU)	Esta instrucción es similar a la anterior (OTL), pero trabaja de manera inversa, es decir, una vez que las condiciones que la preceden cambian de Falso a Verdadero, ésta desactiva el bit y éste queda desactivado sin importar que las condiciones que la preceden cambien

		de estado. Es decir se queda enclavado.
--	--	-----------------------------------------

**Tabla. 2.14. Instrucciones de Salida**

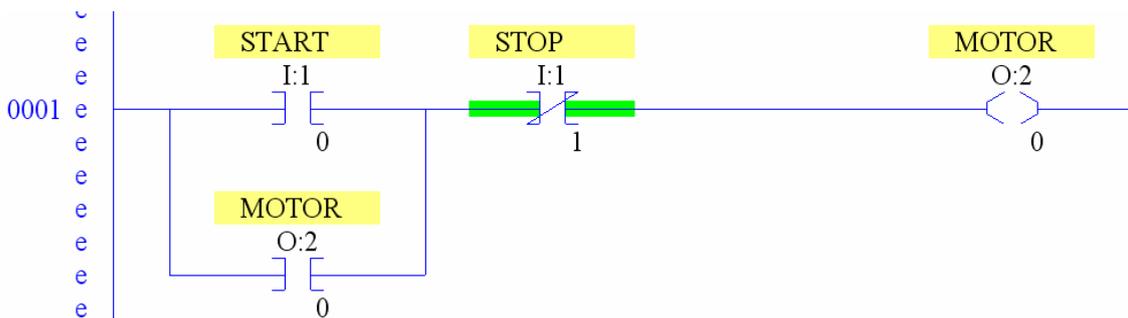


*No se recomienda usar la misma dirección en dos o más instrucciones de salida, a excepción de OTL y OTU que generalmente trabajan en pares.*

**Ejemplos de Programación**

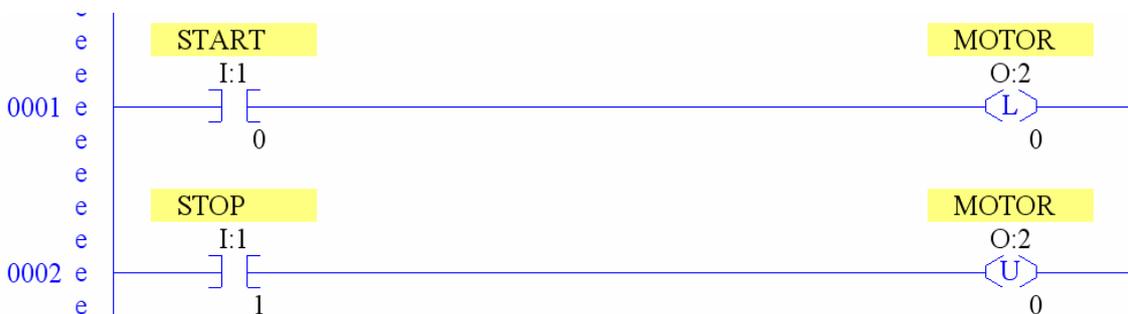
Se desea controlar el encendido y apagado de un motor con dos pulsadores, uno de *START* y uno de *STOP*.

1. El programa Ladder utilizando únicamente como instrucción de salida la instrucción **OPE** se muestra en la **figura 2.3**



**Figura. 2.3. Ejemplo de programación con instrucción OPE**

2. El programa Ladder utilizando únicamente como instrucciones de salida las instrucciones **OTL** y **OTU** se muestra en la **figura 2.4**



**Figura. 2.4. Ejemplo de programación con instrucción OTL Y OTU**

Las direcciones están asignadas de la siguiente manera:

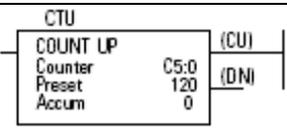
<i>Elemento</i>	<i>Dirección</i>	<i>Descripción</i>
START	I:1/0	Entrada digital 0, cuyo Módulo se encuentra en el Slot 1
STOP	I:1/1	Entrada digital 1, cuyo Módulo se encuentra en el Slot 1
MOTOR	O:2/0	Salida digital 0, cuyo Módulo se encuentra en el Slot 2

**Tabla. 2.15. Direcciones asignada en el ejemplo**

### Instrucciones de Conteo

Existen dos instrucciones principales de conteo, CTU o de conteo ascendente y CTD o de conteo descendente. Ambas poseen los siguientes parámetros:

1. *Dirección del Contador:* Esta debe ser ingresada por el programador, de acuerdo al formato antes explicado
2. *Valor de PRESET:* Este debe ser ingresado por el programador y representa el valor que el acumulado debe alcanzar para activar el bit Done (DN).
3. *El valor acumulado ACCUM:* este representa el valor actual de la cuenta.

<i>Instrucción de Conteo Ascendente</i>		
<b>SÍMBOLO</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>
	Count Up (CTU)	Esta instrucción cuenta de manera creciente las transiciones Falso-Verdadero de las condiciones que la preceden. Pudiendo estas transiciones ser causadas por condiciones de la lógica del programa o por dispositivos de campo externos.

**Tabla. 2.16. Instrucción CTU**

<i>Bits de STATUS de la instrucción CTU</i>		
<i>Este BIT</i>	<i>Es seteado cuando...</i>	<i>Permanece seteado hasta que...</i>
CTU <i>Overflow</i> (OV) Bit 12	El valor acumulado excede el limite y pasa de +32768 a -32767 y continua contando desde ahí	Sea ejecutada una instrucción RES con la misma dirección de la instrucción CTU o hasta que la cuenta sea disminuida con una instrucción CTD y el valor acumulado sea menor o igual a +32768.

<i>Done (DN)</i> Bit 13	El valor acumulado es igual o mayor al valor PRESET	El valor acumulado sea menor al valor de PRESET
CTU <i>Enable (CU)</i> Bit 15	Las condiciones que preceden la instrucción CTU son verdaderas	Las condiciones que preceden la instrucción CTU son falsas o esté activada una instrucción RES con la misma dirección de la instrucción CTU

Tabla. 2.17. Bits de Status de la instrucción CTU

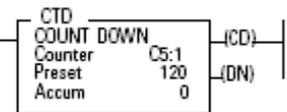
<i>Instrucción de Conteo Descendente</i>		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
	Count Down (CTD)	Esta instrucción cuenta de manera decreciente las transiciones Falso-Verdadero de las condiciones que la preceden. Pudiendo estas transiciones ser causadas por condiciones de la lógica del programa o por dispositivos de campo externos.

Tabla. 2.18. Instrucción CTD

<i>Bits de STATUS de la instrucción CTU</i>		
<i>Este BIT</i>	<i>Es seteado cuando...</i>	<i>Permanece seteado hasta que...</i>
CTD <i>Underflow (UN)</i> Bit 11	El valor acumulado excede el limite y pasa de -32767 a +32768 y continua contando desde ahí	Sea ejecutada una instrucción RES con la misma dirección de la instrucción CTD o hasta que la cuenta sea aumentada con una instrucción CTD y el valor acumulado sea mayor o igual a -32767.
<i>Done (DN)</i> Bit 13	El valor acumulado es igual o mayor al valor PRESET	El valor acumulado sea menor al valor de PRESET
CTD <i>Enable (CU)</i> Bit 14	Las condiciones que preceden la instrucción CTU son verdaderas	Las condiciones que preceden la instrucción CTU son falsas o esté activada una instrucción RES con la misma dirección de la instrucción CTU

Tabla. 2.19. Bits de Status de la instrucción CTD



*La duración de la transición On-Off de la señal de entrada no debe ser menor a dos veces el tiempo de scan. (Asumiendo un ciclo de trabajo del 50%)*

<b>Instrucción de Reset</b>		
<b>SÍMBOLO</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>
—(RES)—	Reset (RES)	Esta instrucción se utiliza para resetear un contador o un temporizador, Cuando las función RES está habilitada esta resetea los Count Up (CTU), Count Down (CTD), Timer On Delay (TON) ó Retentive Timer (RTO) que tengan la misma dirección de la función RES.

**Tabla. 2.20. Instrucción RES**

La instrucción RES resetea todos los bits de Status de los contadores que se vieron anteriormente, y vuelve a cero los valores acumulados tanto de los CTU y CTD.



*Debido a que la instrucción RES vuelve a cero los valores acumulados, si el valor Preset del contador es negativo, esto causa que el bit Done (DN) se active, lo cual deberá ser considerado por el programador*

### **Instrucciones de Temporización**

Existen dos instrucciones principales de temporización, TON o temporizador retardador de encendido y TOF o temporizador retardador de apagado. Ambas poseen los siguientes parámetros:

1. *Dirección del Temporizador:* Esta debe ser ingresada por el programador, de acuerdo al formato antes explicado.
2. *Valor de PRESET:* Este debe ser ingresado por el programador y representa el valor que el acumulado debe alcanzar para activar o desactivar el bit Done (DN) dependiendo si es una instrucción TON ó TOF.
3. *El valor acumulado ACCUM:* este representa el valor actual de la cuenta.

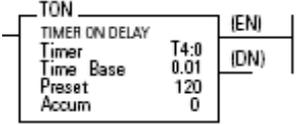
<b>Temporizador retardador de encendido</b>		
<b>SÍMBOLO</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>
	Timer On-delay (TON)	Use esta instrucción para encender o apagar una salida luego que el temporizador ha estado encendido por un intervalo de tiempo definido (Preset). Esta instrucción cuenta intervalos de tiempo cuando las condiciones que la preceden son verdaderas y el valor acumulado es menor al Preset. Si las condiciones que la preceden se vuelven falsas, el temporizador se resetea.

Tabla. 2.21. Instrucción TON

<b>Bits de STATUS de la instrucción TON</b>		
<b>Este BIT</b>	<b>Es seteado cuando...</b>	<b>Permanece seteado hasta que...</b>
Timer Done DN Bit 13	El valor acumulado es igual o mayor que el valor Preset	Las condiciones que preceden a la instrucción TON son falsas
Timer Timing (TT) Bit 14	Las condiciones que preceden a la instrucción TON son verdaderas y el valor acumulado es menor que el Preset	Las condiciones que preceden a la instrucción TON son falsas o cuando el bit Done DN es seteado
Timer Enable EN Bit 15	Las condiciones que preceden la instrucción TON son verdaderas	Las condiciones que preceden la instrucción TOF son falsas

Tabla. 2.22. Bits de Status de la instrucción TON

## Instrucciones de Comparación

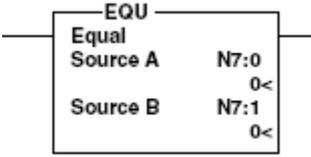
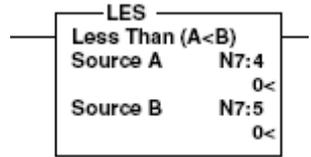
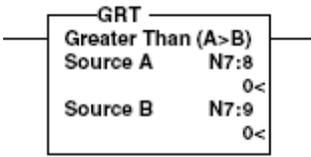
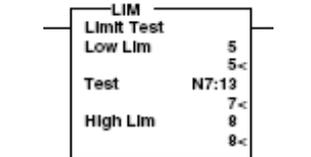
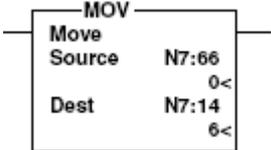
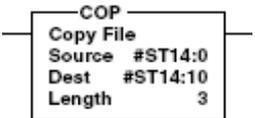
<i>Instrucciones de Comparación</i>		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
	Equal (EQU)	<p>Use esta instrucción para comparar si dos valores son iguales. Si A y B son iguales, la instrucción es verdadera. Si A y B no son iguales, la instrucción es Falsa.</p> <p>A debe ser una dirección, mientras que B puede ser una constante o una dirección. Los enteros negativos son almacenados en complemento a dos.</p>
	Less Than (LES)	<p>Use esta instrucción para comparar si un valor (A), es menor que otro (B). Si <math>A &lt; B</math>, la instrucción es verdadera. Si <math>A &gt; B</math>, la instrucción es falsa.</p>
	Greater Than (GRT)	<p>Use esta instrucción para comparar si un valor (A), es mayor que otro (B). Si <math>A &gt; B</math>, la instrucción es verdadera. Si <math>A &lt; B</math>, la instrucción es falsa.</p>
	Limit Test (LIM)	<p>Use esta instrucción para verificar si un valor está dentro o fuera de un rango especificado.</p>

Tabla. 2.23. Instrucciones de Comparación

**Instrucciones de Transferencia de datos**

<i>Instrucciones de Transferencia de datos</i>																	
<b>SÍMBOLO</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>															
	<p>Move (MOV)</p>	<p>Mueve el valor de la fuente (Source) hacia la dirección del destino (Dest). Si las condiciones que preceden la instrucción son verdaderas. La instrucción mueve los datos una vez cada Scan. La fuente puede ser una dirección o constante. El destino debe ser obligadamente una dirección.</p> <p>Afecta los bits aritméticos de la siguiente manera:</p> <table border="1" data-bbox="746 869 1353 1375"> <thead> <tr> <th colspan="2"><b>BIT</b></th> <th><b>Valor</b></th> </tr> </thead> <tbody> <tr> <td>S:0/0</td> <td>Carry (C)</td> <td>Siempre se resetea</td> </tr> <tr> <td>S:0/1</td> <td>Overflow (V)</td> <td>Siempre se resetea</td> </tr> <tr> <td>S:0/2</td> <td>Zero (Z)</td> <td>Si algún resultado es 0 este se hace 1, o sino se resetea</td> </tr> <tr> <td>S:0/3</td> <td>Sign (S)</td> <td>Si algún resultado es (-) este se hace 1, o sino se resetea</td> </tr> </tbody> </table>	<b>BIT</b>		<b>Valor</b>	S:0/0	Carry (C)	Siempre se resetea	S:0/1	Overflow (V)	Siempre se resetea	S:0/2	Zero (Z)	Si algún resultado es 0 este se hace 1, o sino se resetea	S:0/3	Sign (S)	Si algún resultado es (-) este se hace 1, o sino se resetea
<b>BIT</b>		<b>Valor</b>															
S:0/0	Carry (C)	Siempre se resetea															
S:0/1	Overflow (V)	Siempre se resetea															
S:0/2	Zero (Z)	Si algún resultado es 0 este se hace 1, o sino se resetea															
S:0/3	Sign (S)	Si algún resultado es (-) este se hace 1, o sino se resetea															
	<p>Copy File (COP)</p>	<p>Esta instrucción copia bloques de datos de un lugar a otro. NO utiliza los bits de Status. La fuente es la dirección de los datos que se quieren copiar. El destino es la dirección de inicio donde se va a copiar los datos. Length es el número de elementos que se van a copiar (máximo 128 palabras). Se debe usar el indicador de archivo (#) en la dirección.</p>															

**Tabla. 2.24 Instrucciones de Tranferencia de Datos**



Si se quiere mover una palabra de datos sin afectar los bits aritméticos (S: 0), use la instrucción COP con una longitud (Length) de 1 en lugar de la instrucción MOV.

### Instrucciones Matemáticas.

<i>SUMA</i>		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
	Add (ADD)	Suma un valor (A) a otro (B) y coloca el resultado en el destino.

Tabla. 2.25. Instrucción ADD

La instrucción ADD afecta a los bits aritméticos de la siguiente manera:

BIT		ADD le afecta porque...
S:0/0	Carry (C)	Toma el valor de 1 si se ha generado carry. O sino se resetea
S:0/1	Overflow (V)	Toma el valor de 1 si se genera Overflow (desbordamiento) en el destino. O sino se resetea
S:0/2	Zero (Z)	Toma el valor de 1 si el resultado es 0. O sino se resetea
S:0/3	Sign (S)	Toma el valor de 1 si el resultado es negativo. O sino se resetea

Tabla. 2.26. Bits aritméticos de la instrucción ADD

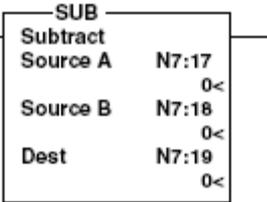
<b>RESTA</b>		
<b>SÍMBOLO</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>
	Subtract (SUB)	Resta un valor (B) de otro (A) y coloca el resultado en el destino.

Tabla. 2.27. Instrucción SUB

La instrucción SUB afecta a los bits aritméticos de la siguiente manera:

<b>BIT</b>		<b>SUB le afecta porque...</b>
S:0/0	Carry (C)	Toma el valor de 1 si se detecta un borrow (préstamo). O sino se resetea
S:0/1	Overflow (V)	Toma el valor de 1 si se genera Underflow (desbordamiento inferior) en el destino. O sino se resetea
S:0/2	Zero (Z)	Toma el valor de 1 si el resultado es 0. O sino se resetea
S:0/3	Sign (S)	Toma el valor de 1 si el resultado es negativo. O sino se resetea

Tabla. 2.28. Bits aritméticos de la instrucción SUB

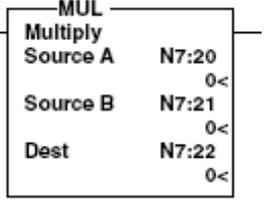
<b>MULTIPLICACIÓN</b>		
<b>SÍMBOLO</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>
	Multiply (MUL)	Multiplica un valor (A) por otro valor (B) y coloca el resultado en el destino.

Tabla. 2.29. Instrucción MUL

La instrucción MUL afecta a los bits aritméticos de la siguiente manera:

<b>BIT</b>		<b>MUL le afecta porque...</b>
S:0/0	Carry (C)	Siempre lo resetea
S:0/1	Overflow	Toma el valor de 1 si se genera Overflow (desbordamiento) en el

	(V)	destino. O sino se resetea
S:0/2	Zero (Z)	Toma el valor de 1 si el resultado es 0. O sino se resetea
S:0/3	Sign (S)	Toma el valor de 1 si el resultado es negativo. O sino se resetea

Tabla. 2.30. Bits aritméticos de la instrucción MUL

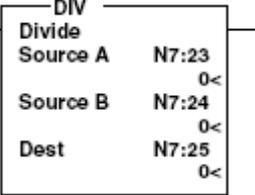
<b>DIVISIÓN</b>		
<b>SÍMBOLO</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>
	Divide (DIV)	Divide un valor (A) para otro valor (B). El cuociente redondeado es colocado en el destino. El cuociente no redondeado es almacenado en la palabra más significativa del registro matemático. El residuo es colocado en la palabra menos significativa del registro matemático.

Tabla. 2.31. Instrucción DIV

La instrucción DIV afecta a los bits aritméticos de la siguiente manera:

<b>BIT</b>	<b>DIV le afecta porque...</b>	
S:0/0	Carry (C)	Siempre lo resetea
S:0/1	Overflow (V)	Toma el valor de 1 si se genera una división para cero o un Overflow (desbordamiento) en el destino. O sino se resetea
S:0/2	Zero (Z)	Toma el valor de 1 si el resultado es 0. O sino se resetea. Es indefinido si existe overflow.
S:0/3	Sign (S)	Toma el valor de 1 si el resultado es negativo. O sino se resetea. Es indefinido si existe overflow.

Tabla. 2.32. Bits aritméticos de la instrucción DIV

### Reloj Calendario

Los procesadores SLC 5/03 o superiores poseen registros incorporados de Reloj/Calendario (Los procesadores Micrologix 1000 no poseen registros de Reloj/Calendario). Estos registros se encuentran dentro de los registros de STATUS del procesador aunque se los clasifica como registros de *Configuración Dinámica*.

La precisión del reloj/calendario en este tipo de procesadores es de  $\pm 54$  segundos/mes a  $+25$  °C ó  $\pm 81$  segundos/mes a  $+60$  °C

A continuación se describen más detalladamente cada uno de estos registros:

Dirección	Nombre	Descripción
S:37	Clock/Calendar Year	Este valor contiene el valor correspondiente al año. Tiene un rango valido de 0 a 65535.
S:38	Clock/Calendar Month	Este valor contiene el valor correspondiente al mes. Tiene un rango valido de 1 a 12. Enero equivale al valor de 1.
S:39	Clock/Calendar Day	Este valor contiene el valor correspondiente al día. Tiene un rango valido de 1 a 31. El primer día del mes equivale al valor de 1.
S:53L	Day-of-Week	Este valor contiene el valor correspondiente al día de la semana. Tiene un rango valido de 0 a 6. El día domingo equivale al valor de 0.
S:40	Clock/Calendar Hours	Este valor contiene el valor correspondiente a la hora. Tiene un rango valido de 0 a 23
S:41	Clock/Calendar Minutes	Este valor contiene el valor correspondiente a los minutos. Tiene un rango valido de 0 a 59.
S:42	Clock/Calendar Seconds	Este valor contiene el valor correspondiente a los segundos. Tiene un rango valido de 0 a 59.

**Tabla. 2.33. Registros Reloj/Calendario**



*Para deshabilitar el reloj calendario ponga ceros a todas la palabras del reloj/calendario (S: 37 a S: 42).*



*El valor de los segundos S: 42 puede no incrementarse exactamente cada segundo, por lo que podría no ser confiable para utilizarse en controles basados en segundos. Si se requiere debe usarse la función STI para tener un control más confiable.*

## Instrucciones de Transferencia de Datos

Existen dos tipos de pilas de Datos: FIFO (First In First Out) y LIFO (Last In First Out). Para trabajar con estos tipos de pilas de datos existen dos funciones para cada tipo, una de carga y otra de descarga.

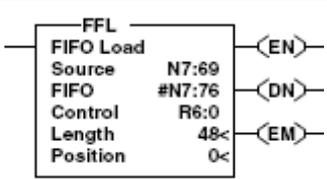
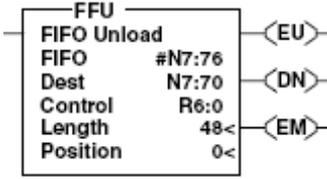
<i>Cargar y Descarga FIFO</i>		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
	FIFO Load (FFL)	Cuando las condiciones que la preceden son verdaderas, la instrucción FFL carga palabras dentro una archivo creado por el usuario llamado pila de datos FIFO
	Unload (FFU)	Cuando las condiciones que la preceden son verdaderas, la instrucción FFU descarga palabras desde una pila FIFO, en el mismo orden en el que entraron.

Tabla. 2.34. Instrucciones FFL y FFU

### Parámetros

- **Source:** es la dirección de una palabra ó una constante de programa (-32768 a 32767) que almacena el valor que va a ser ingresado en la pila FIFO. La instrucción FFL coloca este valor en el siguiente elemento disponible de la pila.
- **Destination:** es la dirección de una palabra que contiene el valor que sale de la pila FIFO. La instrucción FFU descarga un valor desde la pila y lo coloca en esta dirección.
- **FIFO:** Es la dirección de la pila. Esta debe ser una dirección indexada (con el símbolo #) de palabra. Puede ser de tipo entrada, salida, bit, status o entero. Se debe programar la misma dirección para ambas instrucciones (FFL y FFU)
- **Length:** es el máximo número de elementos en una pila, con un límite de 128 palabras. Se debe programar el mismo número en ambas instrucciones.

- **Position:** es la siguiente posición disponible donde la instrucción carga o descarga los datos en la pila. Este número cambia con cada operación de carga y descarga. Se usa el mismo número para ambas instrucciones. Este valor se guarda en el registro de control.
- **Control:** es la dirección de un archivo de tipo CONTROL, en este elemento son almacenados los bits de Status, la longitud (length), y la posición. Se utiliza la misma dirección para ambas instrucciones.

<i>Cargar y Descarga LIFO</i>		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
	LIFO Load (LFL)	Cuando las condiciones que la preceden son verdaderas, la instrucción LFL carga palabras dentro una archivo creado por el usuario llamado pila de datos LIFO
	LIFO Unload (LFU)	Cuando las condiciones que la preceden son verdaderas, la instrucción LFU descarga palabras desde una pila LIFO, en orden opuesto al que entraron.

Tabla. 2.35. Instrucciones LFL y LFU

### Parámetros

- **Source:** es la dirección de una palabra ó una constante de programa (-32768 a 32767) que almacena el valor que va a ser ingresado en la pila LIFO. La instrucción LFL coloca este valor en el siguiente elemento disponible de la pila.
- **Destination:** es la dirección de una palabra que contiene el valor que sale de la pila LIFO. La instrucción LFU descarga un valor desde la pila y lo coloca en esta dirección.
- **LIFO:** Es la dirección de la pila. Esta debe ser una dirección indexada (con el símbolo #) de palabra. Puede ser de tipo entrada, salida, bit, status o entero. Se debe programar la misma dirección para ambas instrucciones (LFL y LFU)

- **Length:** es el máximo número de elementos en una pila, con un límite de 128 palabras. Se debe programar el mismo número en ambas instrucciones.
- **Position:** es la siguiente posición disponible donde la instrucción carga o descarga los datos en la pila. Este número cambia con cada operación de carga y descarga. Se usa el mismo número para ambas instrucciones. Este valor se guarda en el registro de control.
- **Control:** es la dirección de un archivo de tipo CONTROL, en este elemento son almacenados los bits de Status, la longitud (length), y la posición. Se utiliza la misma dirección para ambas instrucciones.



*Excepto cuando se utilicen estas instrucciones de carga y descarga en pares, NUNCA utilice la misma dirección de control para ninguna otra instrucción.*

### Instrucciones Especiales

<i>Compute</i>		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
	Compute (CPT)	Esta instrucción realiza la operación indicada en la expresión. El resultado lo coloca en la dirección Dest.

**Tabla. 2.36. Instrucción CPT**



*Esta instrucción está disponible únicamente en los procesadores SLC 5/03 o superiores. NO en los Micrologix1000.*

Las instrucciones que pueden ser incluidas en la expresión son:

+, -, \* (multiplicación), | (división), SQR, NOT, XOR, OR, AND, TOD, FRD, LN, TAN, ABS, DEG, RAD, SIN, COS, ATN, ASN, ACS, LOG y \*\* (potencia).

Por ejemplo si se quiere hacer la siguiente operación:

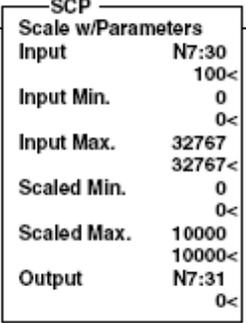
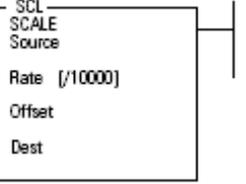
$$\sqrt{a^2 + \frac{(b+c)}{2}}$$

Donde,

$a = N7:0$ ,  $b = N7:1$  y  $c = N7:2$

Entonces se debe ingresar la siguiente expresión:

$$\text{SQR} (( N7:0 ** 2 ) + (( N7:1 + N7:2 ) | 2 ) )$$

<i>Escalar valores</i>		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
	Scale with Parameters (SCP)	Esta instrucción produce un valor escalado que mantiene una relación lineal con el valor de entrada. Esta permite ingresar valores enteros y flotantes.
	Scale Data (SCL)	Cuando la instrucción es verdadera, el valor de Source es multiplicado por el valor Rate, luego el resultado redondeado es sumado con el valor Offset y colocado en la dirección Dest.

**Tabla. 2.37. Instrucciones de Escalado**

En el caso de la función SCL, los parámetros que se debe ingresar se calculan de la siguiente manera:

$$\text{Valor Escalado} = (\text{Valor de entrada} \times \text{Rate}) + \text{Offset}$$

$$\text{Rate} = \frac{(\text{Escalado Maximo} - \text{Escalado Minimo})}{(\text{Entrada Maxima} - \text{Entrada Minima})}$$

$$\text{Offset} = \text{Escalado Minimo} - (\text{Entrada Minima} \times \text{Rate})$$



*La instrucción SCP está disponible únicamente en los procesadores SLC 5/03 o superiores.*

### Instrucciones de Flujo de Programa

<i>Jump to Label (JMP) and Label (LBL)</i>		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
— LBL —	Label (LBL)	Esta instrucción es el objetivo de la instrucción JMP que tenga el mismo número. Se debe colocar esta instrucción en el comienzo del escalón (línea del programa ladder). El número asignado debe ser único.
—(JMP)—	Jump to Label (JMP)	La instrucción JMP causa que el controlador se salte escalones (líneas de ladder). Se puede saltar al mismo Label desde una o más instrucciones JMP.

**Tabla. 2.38. Instrucciones JMP y LBL**

### Instrucción Proporcional Integral Derivativa (PID)

<b>PID</b>		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
	Proportional Integral Derivative (PID)	Ésta es una instrucción de salida que controla propiedades físicas tales como temperatura, presión, nivel de líquidos, flujo, etc.

**Tabla. 2.39. Instrucciones PID**



*Coloque la instrucción PID sin ninguna condición lógica que la preceda, puesto que si una instrucción PID se hace falsa, el término integral se elimina.*

El esquema de un controlador PID aplicado al control de nivel de un líquido, está representado en la figura 2.5

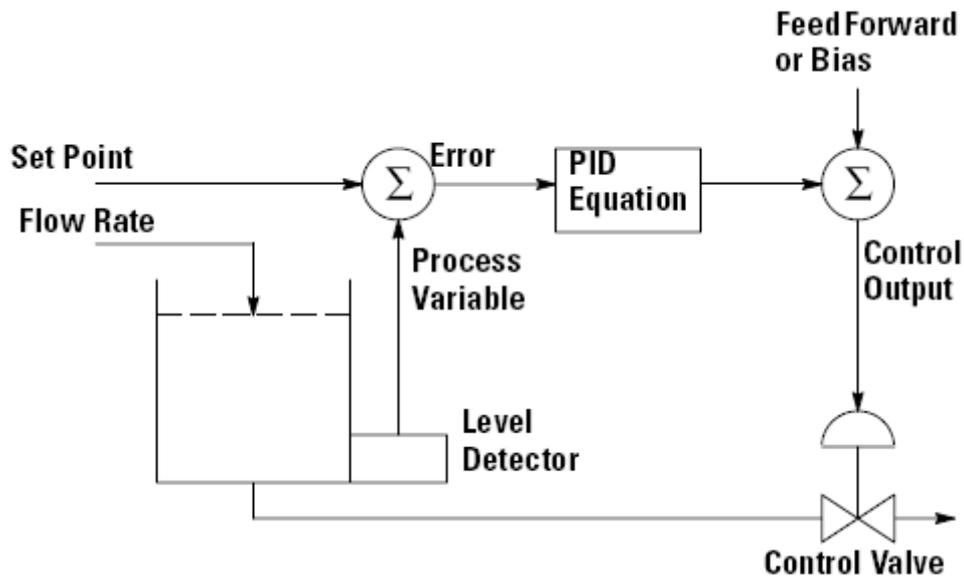


Figura. 2.5. Esquema de un controlador PID aplicado al control de nivel de un líquido

En donde la Ecuación PID controla el proceso enviando una señal de salida a la Válvula de Control. Mientras mayor sea el error entre el Setpoint y la entrada de la Variable del Proceso, mayor será la señal de salida y viceversa. Un valor adicional (Feed Forward o Bias) puede ser sumado a la salida de control como compensación.

### Ecuación PID

La instrucción PID utiliza el siguiente algoritmo:

$$Output = K_c \left[ (E) + \frac{1}{T_i} \int (E)dt + T_D \cdot \frac{D(PV)}{df} \right] + Feed\ Forward / Bias$$

El rango de las constantes de ganancia estándar se muestra en la siguiente tabla:

Término	Rango	Referencia
Ganancia del Controlador $K_c$	0.01 a 327.67 (adimensional)	Proporcional
Reset Time $T_i$	327.67 a 0.02 (minutos/ repetición)	Integral
Rapidez de desviación $T_D$	0.01 a 327.67 (minutos)	Derivativo

Tabla. 2.40. Rango de valores de ganancias estándar



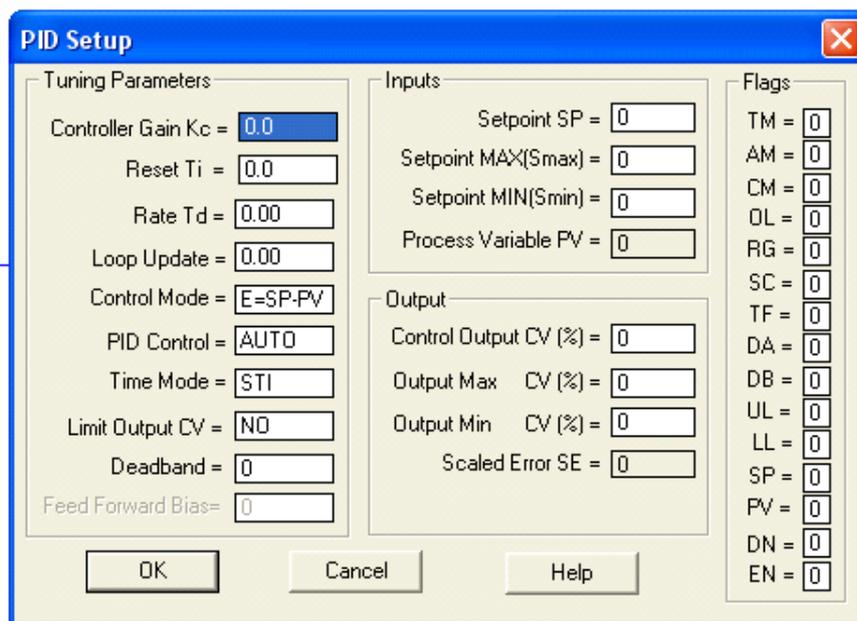
La instrucción PID es de tipo “Entero”, es decir no permite ingresar valores de punto flotante en ninguno de sus parámetros. Si se

*mueven valores de punto flotante a cualquiera de sus parámetros, se produce una conversión de punto flotante a entero.*

## PARÁMETROS

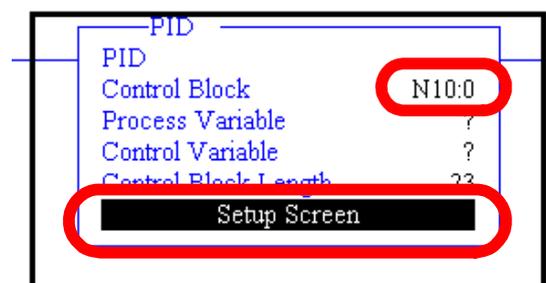
- **Control Block:** Es un archivo que almacena los datos requeridos para que la instrucción pueda operar. Éste archivo tiene una longitud de 23 palabras y pueden ser ingresadas como direcciones de archivos tipo entero. Por ejemplo una entrada de N10:0 asignará los elementos N10:0 hasta N10:22.

La ventana de configuración del Control Block se muestra en la figura 2.6



**Figura. 2.6.** Ventana de configuración del bloque de control PID

Para desplegar ésta ventana se debe dar doble clic en *Setup Screen* como muestra en la figura 2.7



**Figura. 2.7.** Desplegar la ventana de configuración de la instrucción PID



Utilice un archivo de datos UNICO para el Control Block de su PID, es decir no utilice la dirección N7:0 sino cree un nuevo Data File de tipo entero. Por ejemplo N10:0 (ver figura 2.7)

- **Process Variable PV:** es la dirección de un elemento que almacena el valor de entrada del proceso. Esta dirección puede ser la de una palabra en donde se almacene el valor de una entrada analógica ó un entero que contenga el valor pre-escalado (0 a 16383) de la entrada analógica.
- **Control Variable CV:** Es la dirección de un elemento que almacena la salida (rango de 0 a 16383) de la instrucción PID. Normalmente suele ser un valor entero para que usted pueda escalar esta salida al rango que requiera su aplicación y enviarlo a un módulo de salida analógica.

La tabla muestra la estructura del **Control Block** (cada una de las 23 palabras que lo conforman)

<b>Bit</b>	15	14	13	12	11	10	9	8	7	6	5	4	3	2
<b>Word</b>														
Word 0	EN		DN	PV	SP	LL	UL	DB	DA	TF	SC	RG	OL	CM
Word 1	PID Sub Error Code (MSB)													
Word 2	Setpoint SP													
Word 3	Gain Kc													
Word 4	Reset Ti													
Word 5	Rate Td													
Word 6	Feed Forward Bias													
Word 7	Setpoint Maximum (Smax)													
Word 8	Setpoint Minimum (Smin)													
Word 9	Deadband													
Word 10	USO INTERNO – NO CAMBIAR!!													
Word 11	Output Max													
Word 12	Output Min													
Word 13	Loop Update													
Word 14	Scaled Process Variable													

Word 15	Scaled Error SE
Word 16	Output CV% (0-100%)
Word 17	MSW Integral Sum
Word 18	LSW Integral Sum
Word 19	USO INTERNO – NO CAMBIAR!!
Word 20	USO INTERNO – NO CAMBIAR!!
Word 21	USO INTERNO – NO CAMBIAR!!
Word 22	USO INTERNO – NO CAMBIAR!!

**Tabla. 2.41. Bloque de Control PID**

### **Ganancia “Controller Gain” (Kc)**

Kc (palabra 3) es la ganancia “Proporcional”, cuyo rango va desde 0 a 3276.7

(Cuando RG = 0), o de 0 a 327.67 (cuando RG = 1).

### **Reset Time (Ti)**

Ti (palabra 4) es la ganancia “Integral”, cuyo rango va desde 0 to 3276.7 (cuando RG = 0), o de 0 a 327.67 (cuando RG = 1) minutos/repetición.

### **Rate Time (Td)**

Td (palabra 5) es la rapidez de derivación. Su rango es de 0 a 327.67 minutos.

### **Feed Forward/Bias**

El valor que se escriba en esta palabra (palabra #6) es sumado a la salida. Se puede escribir un valor entero entre -16383 y +16383.

### **Error Escalado E**

Es la diferencia entre la variable de procesos y el setpoint, es decir  $E = SP - PV$  ó  $E = PV - SP$  dependiendo del bit de modo de control CM.

### Control (CM)

Cuando es 1,  $E = PV - SP$  lo que causa que la variable de control se incremente cuando la variable del proceso es mayor al setpoint.

Cuando es 1,  $E = SP - PV$  lo que causa que la variable de control se reduzca cuando la variable del proceso es mayor al setpoint.

### Auto / Manual (AM)

Cuando el bit auto/manual es (0), este provoca que el controlador opere en modo automático. Cuando es (1) el controlador opera en modo manual. En modo automático la instrucción controla la variable de control (CV). En modo manual el programador controla la variable de control (CV).



*Durante la sintonización del PID, trabaja en modo Manual.*

### Parámetros de Entrada

Parámetro	Dirección	Rango	Tipo	Tipo de acceso
SP - Setpoint	Palabra #2	0 a 16383*	Control	Lectura/escritura
SPV – Variable de proceso escalada	Palabra #14	0 a 16383	Status	Lectura
SMAX – Setpoint máximo	Palabra #7	-32767 a +32767.	Control	Lectura/escritura
SMIN – Setpoint Mínimo	Palabra #8	-32767 a +32767	Control	Lectura/escritura

**Tabla. 2.42. Parámetros de entrada PID**

### Parámetros de Salida

Parámetro	Dirección	Rango	Tipo	Tipo de acceso
CV - Variable de Control	Def. usuario	0 a 16383	Control	Lectura/escritura

CV% - Porcentaje de salida	Palabra #16	0 a 100	Control	Lectura
OL – Limitar salida	Palabra #0 bit 3	0 ó 1	Control	Lectura/escritura
CVH – Salida Máxima	Palabra #11	0 a 100%	Control	Lectura/escritura
CVL – Salida Mínima	Palabra #12	0 a 100%	Control	Lectura/escritura

**Tabla. 2.43. Parámetros de Salida PID**

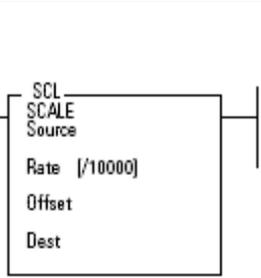
### Escalamiento de Entradas y Salidas

Para la instrucción PID, la escala numérica de la Variable de Proceso (PV) y la de la Variable de Control (CV) es de 0 a 16383. Por lo que, para poder utilizar unidades de ingeniería, se debe primero escalar los rangos de las entradas y salidas analógicas a la escala de 0 a 16383.

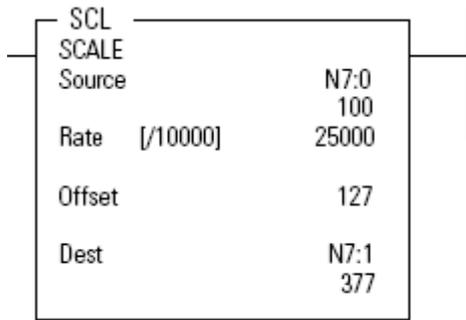
Ejemplos:

- Una entrada analógica con un rango de 4 a 20mA tiene un rango decimal de 3277 a 16384. Este rango decimal debe ser escalado a un rango de 0 a 16383 para poder ser utilizado como PV.
- Para escalar una salida a un rango de 4 a 20 mA cuyo valor decimal sería de 6242 a 31208. En este caso el rango de 0 a 16383 de CV debe ser escalado al rango de 6242 a 31208.

Para realizar este escalamiento, utilice la instrucción SCL, la cual se describe a continuación:

SCL		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
	Scale Data (SCL)	Cuando esta instrucción es verdadera, el valor de la dirección origen (Source) es multiplicado por el valor de (Rate), este resultado es redondeado y sumado al valor de compensación (Offset), para finalmente ser colocado en el destino (Dest).

**Tabla. 2.44. Explicación Instrucción SCL**



Por ejemplo si se tiene la instrucción SCL con los parámetros indicados en la figura, ésta realiza las siguientes operaciones:

La fuente (source) 100 es multiplicada por 25000 y dividida para 10000, y sumada a 127. El resultado

377, es colocado en el destino.

Para calcular los valores de *Offset* y *Rate (Slope)* necesarios para escalar a un rango definido, se deben aplicar las siguientes fórmulas:

$$\text{Valor Escalado} = (\text{valor de entrada} \times \text{Slope}) + \text{offset}$$

$$\text{Slope} = \frac{(\text{Escalado MAX} - \text{Escalado MIN})}{(\text{Entrada MAX} - \text{Entrada MIN})}$$

$$\text{Offset} = \text{Escalado MIN} - (\text{Entrada MIN} \times \text{Slope})$$

**Instrucciones de comunicación**

<i>Mensaje (lectura y/o escritura)</i>		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
	<p>Message Instruction (MSG)</p>	<p>Ésta es una instrucción de salida que permite leer o escribir datos desde un procesador a otro a través de los canales de comunicación.</p>

**Tabla. 2.45. Instrucción MSG**

Para ejecutar esta instrucción, se debe provocar una transición Off-On de las condiciones que la preceden (escalón). No active nuevamente la instrucción, hasta que la instrucción haya completado con o sin éxito el mensaje, lo cual se indica cuando el procesador activa el bit DN o ER.

## Principales Opciones de Configuración de la Instrucción MSG

Existen las siguientes opciones de configuración:

- Lectura/Escritura Punto a Punto (Peer-to-peer Read/Write) en una red local a otro procesador SLC 500
- Lectura/Escritura Punto a Punto (Peer-to-peer Read/Write) en una red remota a otro procesador SLC 500

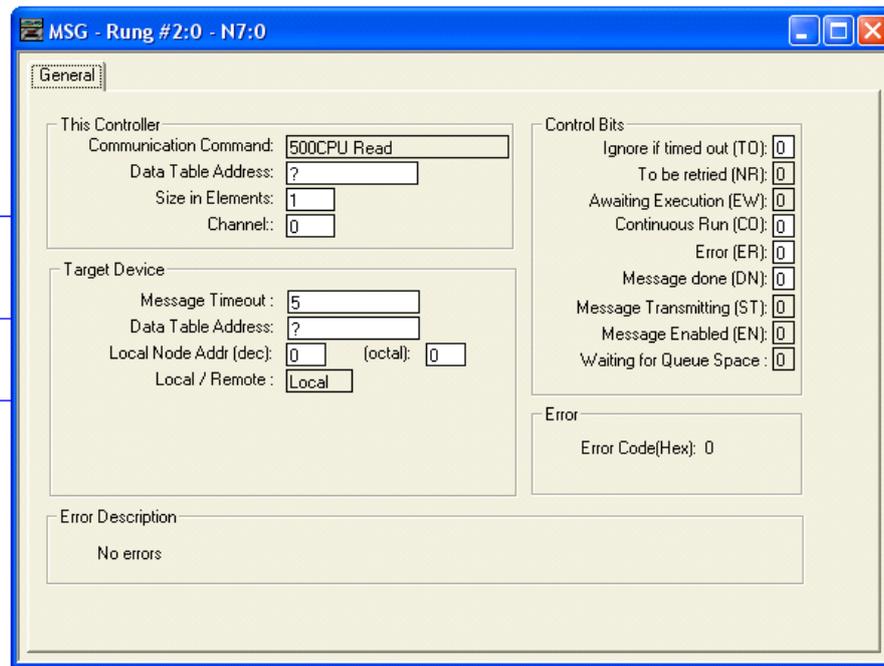
## Parámetros de la instrucción MSG

Cuando se programe esta instrucción, se deben ingresar los siguientes parámetros:

- **Read/Write:** Read indica que el procesador local (procesador en el cual se encuentra la instrucción MSG) está recibiendo datos, Write indica que el procesador local está enviando datos.
- **Target Device:** Identifica el tipo de dispositivo el cual responde al comando MSG. En nuestro caso debe ser 500CPU.
- **Local o Remote:** Identifica si el mensaje es enviado a un dispositivo en una red local o remota.
- **Control Block:** Es una dirección de un archivo de datos de tipo entero que el programador debe seleccionar. Este es un bloque de palabras que contienen los bits de Status, la dirección del objetivo y otros datos asociados a la instrucción MSG.
- **Control Block Length:** Este indica cuantas palabras están siendo utilizadas en el block de control.

## Parámetros de la Pantalla SETUP SCREEN

La pantalla Setup Screen se muestra en la figura 2.8.



**Figura. 2.8. Pantalla de configuración de la instrucción MSG**

### Parámetros “This Controller”

- **Data Table Address**
  - ✓ Para lectura (Read), esta es la dirección de inicio en la cual se recibe los datos desde el dispositivo objetivo (Target Device).
  - ✓ Para Escritura (Write), esta es la dirección de inicio de los datos que son escritos en el dispositivo objetivo (Target Device).
  
- **Size in Elements:** Especifica la longitud del mensaje en elementos. El número máximo de elementos que son transferidos a través de la instrucción MSG está determinado por el tamaño del tipo de datos del destino.
  - ✓ Para lectura (Read), el tipo de datos en el procesador local determina el número máximo de elementos.
  - ✓ Para Escritura (Write), el tipo de datos en el dispositivo objetivo determina el número máximo de elementos.
  
- **Channel:** Especifica el canal de comunicación que está siendo utilizado para transmitir el mensaje.

- ✓ SLC 5/03 - (Canal 0, RS-232) ó (Canal 1, DH-485)
- ✓ SLC 5/05 - (Canal 0, RS-232) ó (Canal 1, Ethernet)

Parámetros “Target Device”

- **Message Timeout:** Especifica la duración en segundos de el temporizador del mensaje. Un valor de 0 segundos significa que no existe temporizador, es decir, un mensaje que ha sido reconocido, espera indefinidamente por la respuesta. Este parámetro tiene un rango valido entre 0 y 255. Cuando se utiliza el canal 1 (ethernet) de los procesadores SLC 5/05, este parámetro no debe modificarse pues es asignado por el controlador.
- **Data Table Address**
  - ✓ Para lectura (Read), Esta es la dirección de inicio de donde los datos están siendo leídos.
  - ✓ Para Escritura (Write), Esta es la dirección de inicio en donde los datos están siendo escritos.
- **Local Node Addr (dec)/(oct):** Especifica el número de nodo del dispositivo que está recibiendo el mensaje. Este parámetro tiene un rango valido que depende del tipo de comunicación utilizada, como se puede ver en la siguiente tabla:

Protocolo	Decimal	Octal
DH-485	0-31	0-37
DH+	0-63	0-77
DF1	0-254	0-376

**Tabla. 2.46. Número de Nodo Instrucción MSG**

- **Ethernet (IP) Address:** Especifica la dirección IP de dispositivo objetivo, es decir del que está recibiendo el mensaje.
- **MultiHop:** Especifica si el mensaje esta siendo o no está siendo enviado a un dispositivo de interfase Ethernet (1761-NET-ENIW).

### 2.1.2. SERIE MICROLOGIX 1000\*

#### INTRODUCCIÓN



Figura. 2.9. Equipos de la serie Micrologix 1000

“Basado en la arquitectura de la familia de controladores SLC 500, el MicroLogix 1000 ofrece alta velocidad, poderosas instrucciones y comunicaciones flexibles para las aplicaciones que requieren soluciones compactas y económicas.

El controlador programable MicroLogix 1000 está disponible en versiones de E/S digitales de 10 puntos, 16 puntos ó 32 puntos. Las versiones analógicas también están disponibles con 20 puntos de E/S digitales y 5 puntos de E/S analógicas.

El circuito de E/S analógicas de las unidades MicroLogix 1000 está incorporado en el controlador base, y no a través de módulos adicionales. Por lo tanto, proporciona un rendimiento analógico de muy alta velocidad y económico.”Rockwell Automation, Controladores Programables Micrologix 1000, 1761-TD001B-ES-P, [www.ab.com](http://www.ab.com), Mayo 2003.

---

\*La información expuesta en esta sección fue extraída de los textos 10, 11 y 12 citados en las referencias bibliográficas.

## CARACTERÍSTICAS

La siguiente muestra las características generales de los PLC's de la serie Micrologix 1000:

Temperatura de operación	Montaje horizontal: 0 °C a +55 °C
	Montaje vertical: 0 °C a +45 °C para E/S digitales, 0 °C a +40 °C para E/S analógicas
Temperatura de almacenamiento	-40 °C a +85 °C
Humedad	5 a 95% sin condensación
Vibración	En operación: 5 Hz a 2 kHz, 0.381 mm (0.015 pulg.) pico a pico, 2.5 g montado en panel, 1 hr por eje
	Fuera de operación: 5 Hz a 2 kHz, 0.762 mm (0.030 pulg.) pico a pico, 5 g, 1 hr por eje
Choque, controladores de 10 y 16 puntos	En operación: 10 g aceleración pico (7.5 g montado en el riel DIN) (11 ± 1 ms de duración) 3 veces en cada dirección, cada eje
	Fuera de operación: 20 g aceleración pico (11 ± 1 ms de duración), 3 veces en cada dirección, cada eje
Choque, controladores de 2 puntos y analógicos	En operación: 7.5 g aceleración pico (5.0 g montado en el riel DIN) (11 ± 1 ms de duración) 3 veces en cada dirección, cada eje
	Fuera de operación: 20 g aceleración pico (11 ± 1 ms de duración), 3 veces en cada dirección, cada eje
Certificaciones	<ul style="list-style-type: none"> <li>▪ Equipo de control industrial en lista UL</li> <li>▪ Equipo de control industrial en lista c-UL para uso en Canadá</li> <li>▪ Equipo de control industrial en lista UL para uso en lugares peligrosos Clase 1, División 2, Grupos A, B, C, D</li> <li>▪ CE Marcado para todas las directivas aplicables</li> </ul>

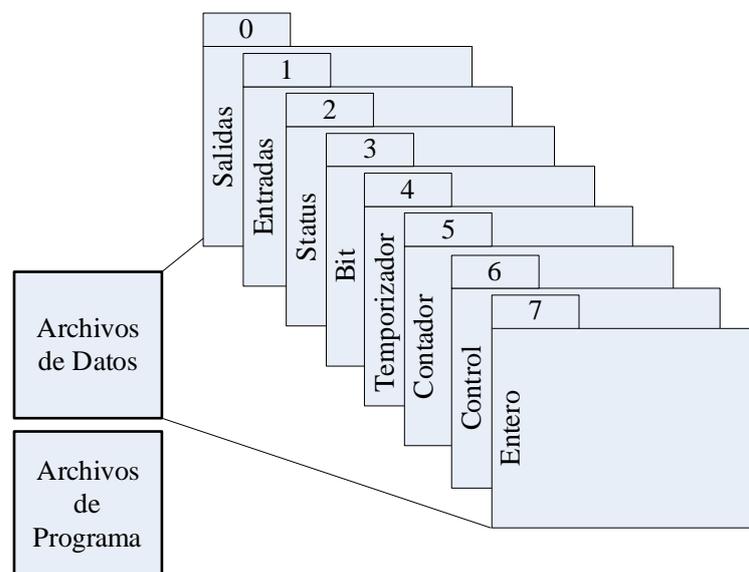
	<ul style="list-style-type: none"> <li>▪ C-Tick Marcado para todas las leyes aplicables</li> </ul>
Descarga electrostática	EN 61000-4-2 a 8 KV
Sensibilidad radiada	EN 61000-4-3 a 10 V/m, 27 MHz a 1000 MHz, 3V/m, 87 MHz a 108 MHz, 174 MHz a 230 MHz y 470 MHz a 790 MHz
Fenómenos transitorios rápidos	EN 61000-4-4, fuente de alimentación 2 K, E/S; com. 1 KV
Aislamiento	1500 VCA

**Tabla. 2.47. Características generales Serie Micrologix**

## PROGRAMACIÓN DE MICROLOGIX-1000

### Archivos del Procesador (Processor Files)

La memoria de usuario de SLC 500 está compuesta por Archivos de Datos y Archivos de Programa como se muestra en la figura 2.10



**Figura. 2.10. Archivos del procesador Micrologix 1000**

### Direccionamiento

El formato de direccionamiento de cada uno de los diferentes tipos de archivos de datos coincide con el formato utilizado en la programación de los PLC's de la serie SLC-500. Referirse a la sección de direccionamiento de la familia SLC-500.

## Conjunto de instrucciones para la programación de Micrologix 1000

La siguiente tabla muestra una tabla comparativa de las instrucciones disponibles en las familias SLC-500 y Micrologix 1000.

INSTRUCCIONES			INSTRUCCIONES			INSTRUCCIONES		
Nº	SLC	Micrologix	Nº	SLC	Micrologix	Nº	SLC	Micrologix
1	ABL	-	39	FBC	-	77	OTL	OTL
2	ABS	-	40	FFL	FFL	78	OTU	OTU
3	ACB	-	41	FFU	FFU	79	PID	-
4	ACI	-	42	FLL	FLL	80	-	RAC
5	ACL	-	43	FRD	FRD	81	RAD	-
6	ACN	-	44	GEQ	GEQ	82	REF	-
7	ACS	-	45	GRT	GRT	83	RES	RES
8	ADD	ADD	46	-	HSC	84	RET	-
9	AEX	-	47	-	HSD	85	RHC	-
10	AHL	-	48	-	HSE	86	RMP	-
11	AIC	-	49	-	HSL	87	RPI	-
12	AND	AND	50	IID	-	88	RTO	RTO
13	ARD	-	51	IIE	-	89	SBR	SBR
14	ARL	-	52	IIM	IIM	90	SCL	SCL
15	ASC	-	53	INT	INT	91	SCP	-
16	ASN	-	54	IOM	IOM	92	SIN	-
17	ASR	-	55	JMP	JMP	93	SQC	SQC
18	ATN	-	56	JSR	JSR	94	SQL	-
19	AWA	-	57	LBL	LBL	95	SQO	-
20	AWT	-	58	LEQ	LEQ	96	SQR	-
21	BSL	BSL	59	LES	LES	97	STD	-
22	BSR	BSR	60	LFL	LFL	98	STE	-
23	BTR	-	61	LFU	LFU	99	STS	-
24	BTW	-	62	LIM	LIM	100	SUB	-
25	CLR	CLR	63	LN	-	101	SUS	-
26	COP	COP	64	LOG	-	102	SVC	-

27	COS	-	65	MCR	MCR	103	SWP	-
28	CPT	-	66	MEQ	MEQ	104	TAN	-
29	CTD	CTD	67	MOV	MOV	105	TDF	-
30	CTU	CTU	68	MSG	MSG	106	TND	-
31	DCD	DCD	69	MUL	MUL	107	TOD	-
32	DDT	-	70	MVM	MVM	108	TOF	-
33	DDV	DDV	71	NEG	NEG	109	TON	-
34	DEG	-	72	NEQ	NEQ	110	XIC	-
35	DIV	DIV	73	NOT	NOT	111	XIO	-
36	ENC	ENC	74	OR	OR	112	XOR	-
37	END	-	75	OSR	OSR	113	XPY	-
38	EQU	EQU	76	OTE	OTE			

**Tabla. 2.48. Instrucciones SLC500 y Micrologix**

Para obtener una descripción más detallada de las principales instrucciones de programación de la familia Micrologix 1000 referirse a la sección de SLC-500.



*Los procesadores Micrologix 1000 no poseen registros de Reloj/Calendario*

## 2.2. INTERFACES DE COMUNICACIÓN\*

Una vez determinada la aplicación que se va a implementar, se debe determinar que tipo de red de comunicación se requiere, para lo que se establece las siguientes directivas:

<b>Si la aplicación requiere:</b>	<b>Utilice esta red:</b>
<ul style="list-style-type: none"> <li>▪ Conexión a módems telefónicos para mantenimiento de programas o recolección de datos remotos.</li> <li>▪ Conexión a módems de línea dedicada o de radio para uso en los sistemas SCADA</li> <li>▪ Funciones de unidad Terminal remota (RTU)</li> </ul>	DF1 Full-Duplex  DF1 Half-Duplex esclavo
<ul style="list-style-type: none"> <li>▪ Compartición con el programa de mantenimiento a nivel de la planta y celdas</li> <li>▪ Compartición de datos entre 32 controladores</li> <li>▪ Carga, descarga y monitoreo del programa a todos los controladores</li> <li>▪ Compatibilidad con múltiples dispositivos HMI de Allen-Bradley</li> </ul>	DH-485
<ul style="list-style-type: none"> <li>▪ Conexión de dispositivos de bajo nivel de múltiples suministradores directamente a los controladores de la planta</li> <li>▪ Compartición de datos entre 64 dispositivos</li> <li>▪ Mejores diagnósticos para ofrecer mejor recolección de datos y detección de fallos</li> <li>▪ Menos cableado y tiempo de puesta en marcha reducido comparado con los sistemas cableados tradicionales</li> </ul>	DeviceNet
<ul style="list-style-type: none"> <li>▪ Carga/descarga del programa</li> <li>▪ Comunicación entre dispositivos similares</li> <li>▪ Comunicación mediante correo electrónico</li> <li>▪ Puerto 10Base-T, con indicadores LED incorporados</li> <li>▪ Capacidad de servidor de Internet (ENIW solamente)</li> </ul>	EtherNet/IP

A continuación se describen los diferentes tipos de redes:

\*La información expuesta en esta sección fue extraída de los textos 13, 14, 15, y 19 citados en las referencias bibliográficas.

### 2.2.1. INTERFASE DE COMUNICACIÓN ETHERNET

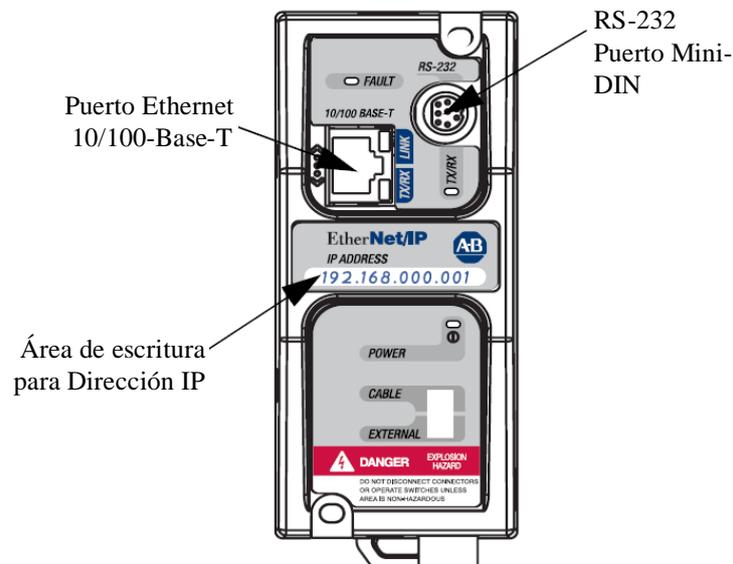
La red Ethernet TCP/ IP es una red de área local diseñada para el intercambio de información a alta velocidad entre equipos y dispositivos relacionados. Con su elevado ancho de banda de hasta 100 Mbps, una red Ethernet permite que muchos equipos, controladores y otros dispositivos se comuniquen a través de distancias de hasta 100m.

En la capa de información, una red Ethernet permite a los sistemas de toda la empresa acceder a datos de la planta.

La red Ethernet ofrece muchas posibilidades porque se puede maximizar la comunicación entre los diversos equipos que ofrecen los distribuidores. El protocolo utilizado por Internet es TCP/IP.

Únicamente el procesador SLC 5/05 posee un puerto de interfase de comunicación ethernet incorporado (Ethernet 10Base-T). Sin embargo el módulo de interfase 1761-NET-ENIW brinda conectividad EtherNet/IP (Ethernet 10/100Base-T) a todos los controladores Micrologix y cualquier dispositivo DF1 full-duplex (SLC 5/03).

#### Módulo de Interfase Ethernet 1761-NET-ENIW



**Figura. 2.11. Descripción del módulo de Interfase Ethernet 1761-NET-ENIW**

El módulo de interfase ENIW, nos permite conectar fácilmente controladores sin puerto ethernet a redes EtherNet/IP para cargar y descargar programas, comunicarse

entre controladores y generar mensajes de e-mail a través de SMTP (Simple Mail Transport Protocol).

### Configuración del Módulo de Interfase Ethernet 1761-NET-ENIW

La configuración del módulo ENIW puede ser ingresada de las siguientes maneras:

- Por medio del software “ENI/ENIW Configuration Utility”, a través del puerto RS-232.
- Por medio del software “ENI/ENIW Configuration Utility”, a través del puerto Ethernet, utilizando el software “Com Port Redirector”
- Por medio de mensajes (instrucción MSG) desde un controlador Allen Bradley al nodo respectivo a través del puerto RS-232. Ej: Para ingresar la dirección IP, se debe enviar esta dirección al nodo 250.

En este caso, por facilidad se lo va a configurar por medio del software “ENI/ENIW Configuration Utility”, a través del puerto RS-232. Para lo que se deben seguir los siguientes pasos:

1. Correr el software ENI/ENIW Utility.

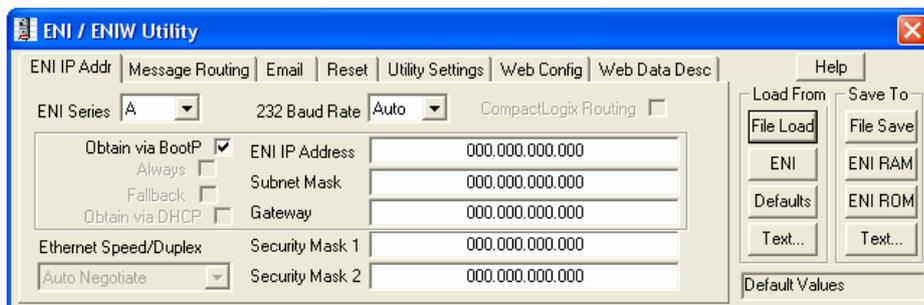


Figura. 2.12. Pantalla de configuración 1 ENIW

2. Dar clic en la pestaña *Utility Settings* para configurar el canal de comunicación.

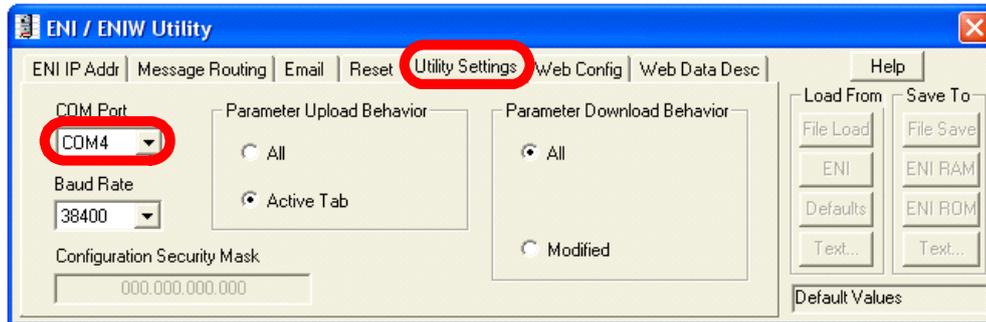


Figura. 2.13. Pantalla de configuración 2 ENIW

3. Dar clic en la pestaña *ENI IP Addr* e ingrese los siguientes parámetros:

- ENI Series: D (los módulos son de la serie D)
- 232 Baud Rate: Seleccionar la velocidad de transmisión serial del dispositivo conectado al módulo. Por ejemplo, si está conectado un PLC de la serie 5/03, está será la velocidad de transmisión del canal 0 (generalmente 19200).
- Obtain via Bootp: Desactivado (Quitar el visto de selección)
- ENI IP Address: Ingresar la dirección IP (por Ej: 192.168.010.002)
- Subset Mask: Ingresar la máscara de subred (por Ej: 255.255.255.0)

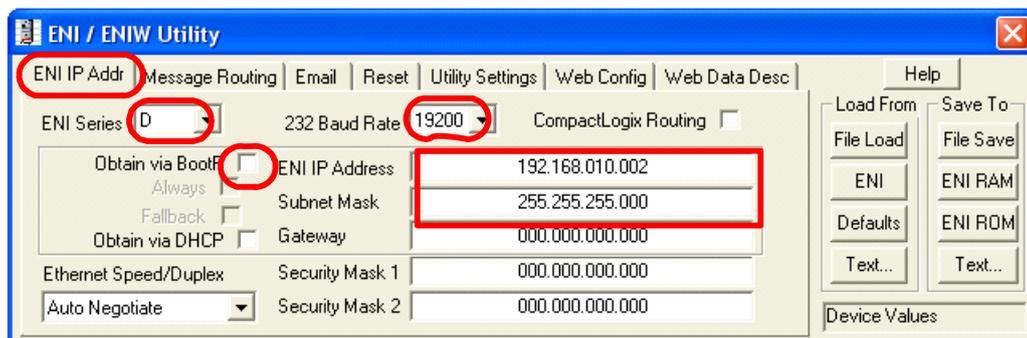


Figura. 2.14. Pantalla de configuración 3 ENIW

4. Dar clic en la pestaña *Message Routing* en donde a un nodo (Dest) se le asigna la dirección IP de un dispositivo de la red. Por ejemplo, si se tiene dos PLC's conectados a una red Ethernet, un 5/05 (conectado directamente a través del canal 1) y un 5/03 (conectado a través del módulo ENIW), y se quiere enviar mensajes desde el 5/03 al 5/05, con una instrucción MSG se envía el mensaje

vía serial DF1 al nodo 0, y el módulo ENIW lo direcciona a la IP asignada al nodo, en este caso sería la dirección IP del 5/05.

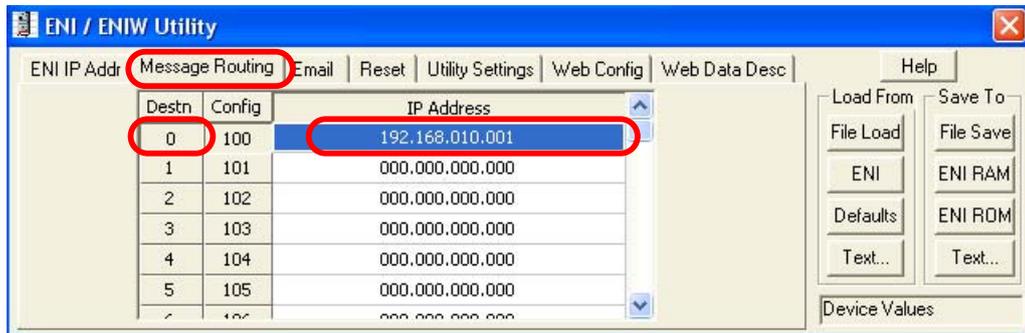


Figura. 2.15. Pantalla de configuración 4 ENIW

5. Dar clic en la pestaña **Email** y configurar los siguientes parámetros:

- Email Server IP Address: Ingresar la dirección del servidor de correo de la red ethernet en la que se encuentra conectado el módulo.
- "From": Escriba la dirección de correo que se desea que aparezca en el email como dirección de origen..
- "To": Ingresar la dirección de correo de destino, ésta se asigna a un número de nodo (Destn) al cual se debe enviar el mensaje via serial para que el módulo ENIW lo envíe como email.

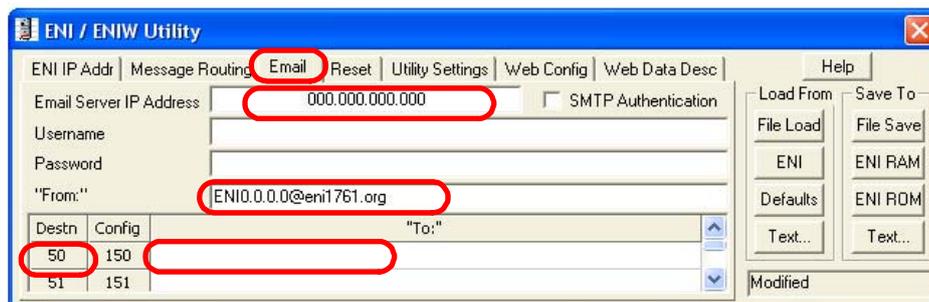
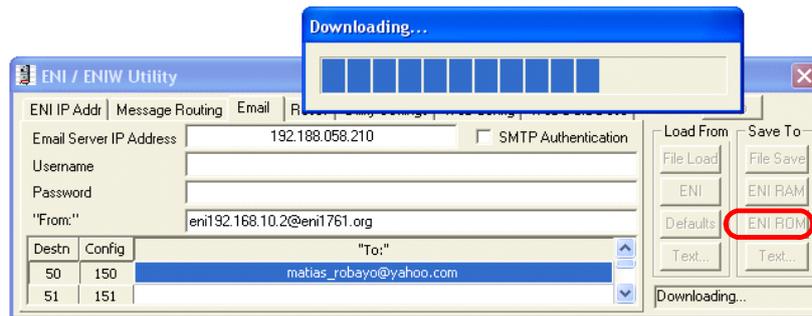


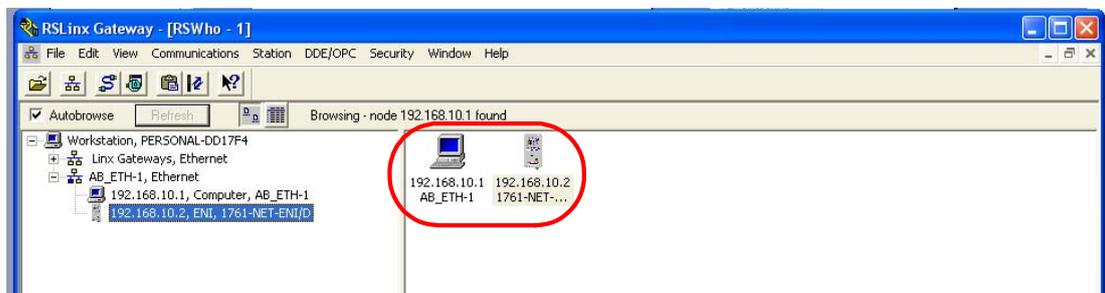
Figura. 2.16. Pantalla de configuración 5 ENIW

6. Finalmente, para grabar la configuración en el módulo, presione sobre **ENI ROM**, con lo que empezará el proceso de grabación



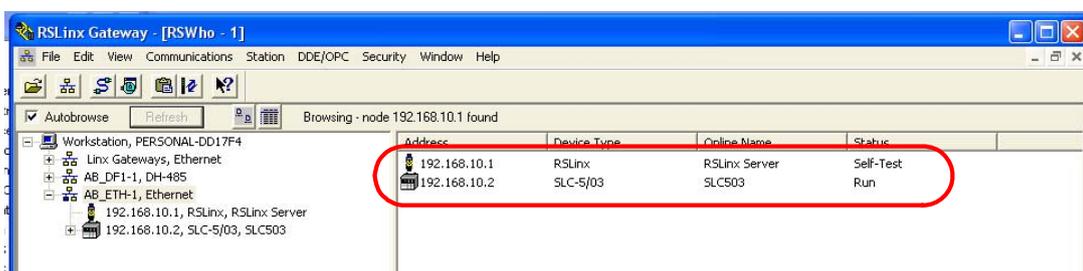
**Figura. 2.17. Pantalla de configuración 6 ENIW**

Para verificar la configuración del módulo, se puede abrir la ventana RSWho del software RSLinx (previamente configurado el driver Ethernet con la dirección del módulo y de la PC) y se verá la siguiente pantalla:



**Figura. 2.18. Verificación de la dirección IP del módulo NET-ENIW**

Posteriormente, cuando ya se configure el PLC 5/03 ó Micrologix, y se lo conecte al puerto serial del módulo, ya no aparecerá la imagen del módulo en esta pantalla, sino la del PLC.



**Figura. 2.19. Verificación de la conexión del módulo NET-ENIW y el PLC SLC 5/03**

## 2.2.2. INTERFASE DE COMUNICACIÓN SERIAL RS-232

RS-232 es un estándar de la Electronics Industries Association (EIA) que especifica las características eléctricas, mecánicas y funcionales para la comunicación serial binaria.

Cabe señalar que RS-232 es una definición de características NO un protocolo.

Uno de los mayores beneficios de la interfase RS-232 es que nos permite integrar módems de radio y teléfono en nuestros sistemas de control (utilizando únicamente el protocolo DF1 apropiado, NO el protocolo DH-485). La distancia a través de la cual nos permite comunicarnos con ciertos dispositivos es virtualmente ilimitada.

DF1 es un protocolo abierto, libremente disponible, desarrollado por Allen-Bradley, que se ha utilizado por más de 15 años. DF1 es utilizado para la transmisión de datos de dos vías, punto a punto y redes multi-drop en aplicaciones locales y de larga distancia.

El protocolo DF1 puede se configurado para operar en “Full Duplex” y “Half Duplex”.

DF1 Full Duplex es utilizado sobre una conexión punto a punto (un dispositivo conectado a otro), este permite comunicaciones bi-direccionales simultáneas y produce un “throughput” alto para aplicaciones de alto desempeño.

DF1 Half Duplex es utilizado para comunicaciones multi-drop maestro/esclavo. Este permite comunicaciones bi-direccionales en un sola dirección a la vez.

### **2.2.3. INTERFASE DE COMUNICACIÓN SERIAL RS-485**

RS-485 es una definición de las características eléctricas, mecánicas y funcionales para la comunicación serial. Cabe señalar que RS-485 es una definición de características NO un protocolo. RS-485 utiliza dispositivos que son capaces de co-existir en un circuito común de datos lo que permite que los datos sean fácilmente compartidos entre los dispositivos.

Los equipos Allen-Bradley utilizan RS-485 half-duplex como interfase física bajo el protocolo DH-485 que define la comunicación entre múltiples dispositivos que co-existen en un único par de cables.

Un red DH-485 ofrece:

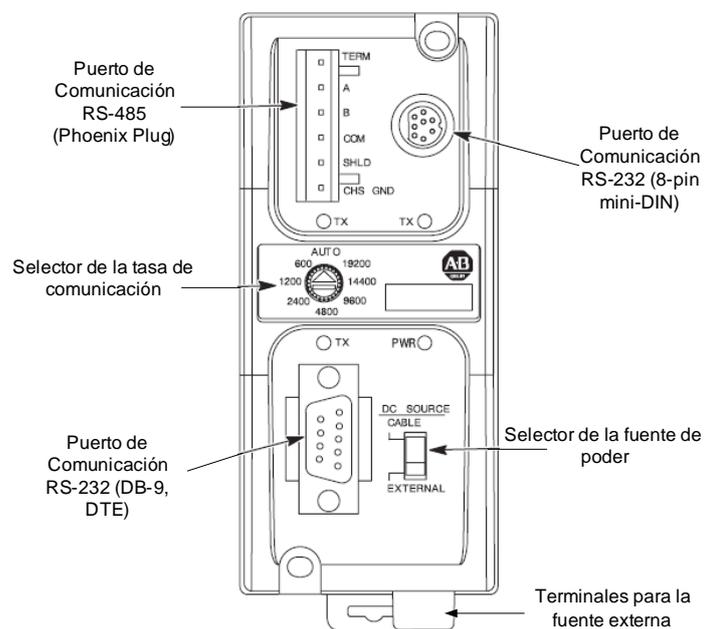
- Interconexión de 32 dispositivos
- Capacidad Multi-master
- Control de acceso token passing
- La habilidad de adicionar o remover modos sin alterar la red

- Longitud máxima de 1219m (4000 pies)

El protocolo DH-485 soporta dos clases de dispositivos: Iniciadores (initiator) y contestadores (responders)

Todos los iniciadores sobre la red pueden iniciar transferencia de mensajes. Para determinar que iniciador tiene el derecho de transmitir, se utiliza el algoritmo token passing.

### Módulo Conversor de Interfase Avanzado 1761-NET-AIC



**Figura. 2.20. Descripción del módulo NET-AIC**

Este módulo permite conectar varios dispositivos puesto que es compatible con una gran variedad de controladores y periféricos de la serie SLC y Micrologix.

Los controladores Micrologix 1000 proveen energía al módulo 1761-NET-AIC a través del cable RS-232 8-pin mini-DIN, sin embargo si se conecta otro tipo de controlador se requiere conectar una fuente externa en los terminales del módulo.

El selector de la tasa de comunicación es utilizado para hacer coincidir la tasa de transmisión del módulo con la de la red a la cual se va a conectar. Normalmente se debe dejar en la posición AUTO, únicamente en ambientes con mucho ruido se debe seleccionar la misma tasa de transmisión de la red

### Modos de Operación

Este módulo puede se utilizado en los siguientes modos de operación:

- Aislador Punto a Punto
- Aislador de RS-232 a RS-485
- Aislador de RS-232 a half-duplex ASCII

### Configuración de la dirección del nodo de comunicación

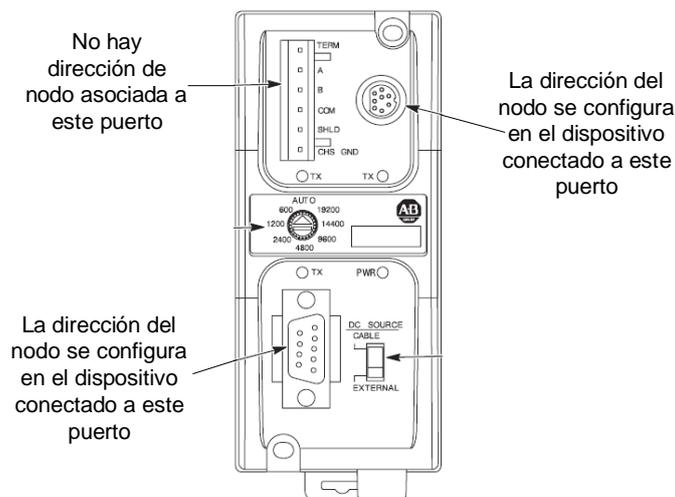


Figura. 2.21. Configuración de los nodos de comunicación del módulo NET-AIC

### Diagrama de conexión

La conexión del módulo en modo aislador Punto a Punto se muestra en la figura 2.22. Si se requiere conectar un PLC de la serie SLC500 se debe conectar una fuente externa al módulo.

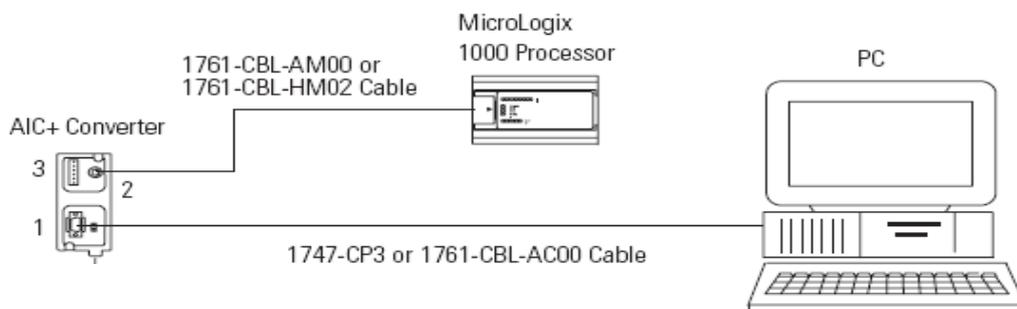


Figura. 2.22. Conexión del módulo NET-AIC en modo aislador punto a punto

## CAPITULO 3

### SOFTWARE ROCKWELL

#### 3.1. SOFTWARE RSLINX\*

##### 3.1.1. INTRODUCCIÓN

RSLinx es un servidor completo de comunicaciones que provee conectividad entre dispositivos de planta y una gran variedad de aplicaciones de software Rockwell tales como RSLogix 500 y RSView32, entre otros. RSLinx puede soportar múltiples aplicaciones de software simultáneamente, comunicándose con una gran variedad de dispositivos en diferentes redes.

RSLinx Classic está disponible en siete versiones, cada una con diferentes requerimientos y funcionalidades. La versión que se encuentra instalada, aparece en la barra de título en la parte superior de la ventana de inicio. Se debe tener en cuenta que si no se instala correctamente los archivos de activación, automáticamente aparece la versión más básica “RSLinx Classic Lite”.



*Ciertos paquetes de software anti-virus, tales como Norton Anti-virus, pueden corromper los archivos de activación. Configure su software antivirus para que evite revisar los archivos EVRSI.SYS y 386SAP.PAR.*

La Escuela Politécnica de Ejército (ESPE) cuenta con la versión **RSLinx Classic Gateway**, la cual amplía las comunicaciones, conectando clientes a través de redes TCP/IP. La versión que se posee, se muestra en la figura 3.1.

---

\*La información expuesta en esta sección fue extraída de los textos 17, 18, y 19 citados en las referencias bibliográficas.



**Figura. 3.1. Versión RSLinx Classic Gateway que posee la ESPE**

Entre las principales aplicaciones de RSLinx Classic Gateway están las siguientes:

- Monitorear datos de PLC's de las series PLC, SLC, o ControlLogix.
- Monitorear la lógica de programación de los procesadores de las familias PLC ó SLC500
- Adquisición de datos utilizando OPC (OLE para Control de Procesos) o DDE (Dynamic Data Exchange) locales para cualquier número de dispositivos. Estos incluyen clientes tales como RSView32, Microsoft Office, Visual Basic y páginas Web.
- Programación con lógica Ladder utilizando productos RSLogix.
- Múltiples clientes RSView32 accediendo a datos a través de un solo RSLinx Classic Gateway (OPC remoto).
- PC remota corriendo RSLogix conectada a la red de la planta a través de un modem para cambios de programación en línea.
- Aplicaciones remotas Microsoft Office para desplegar datos de planta (Ej. Excel).

### 3.1.2. REQUERIMIENTOS DEL SISTEMA

Para utilizar efectivamente Rslinx Classic, su computador personal debe reunir los siguientes requerimientos mínimos de hardware y software:

#### Requerimientos mínimos de Hardware

- Procesador Pentium 100MHz con al menos 32 Megabytes (MB) de memoria RAM.
- Al menos 35 MB disponibles de espacio en disco duro.
- Adaptador gráfico SVGA de 16-color, con resolución de 800x600 o mayor.
- Un Mouse u otro dispositivo apuntador Windows-compatible.
- Una tarjeta de red Ethernet o un dispositivo de comunicación Allen-Bradley.

#### Requerimientos mínimos de Software

- Microsoft Internet Explorer 6.0 (ó posterior).
- Uno de los siguientes sistemas operativos:
  - Microsoft Windows XP, XP SP1 y XP SP2.
  - Microsoft Windows 2003 SP1.
  - Microsoft Windows 2000.
  - Microsoft Windows Me (Millennium Edition).
  - Microsoft Windows 98 (versión 4.10.2222A).
  - Microsoft Windows NT Versión 4.0 (Service Pack 3 o posterior).

### 3.1.3. CONFIGURACIÓN DE LOS DRIVERS DE COMUNICACIÓN

#### Driver RS-232 de comunicación entre el PLC y la PC

Para configurar correctamente el driver de comunicación, se deben seguir las siguientes instrucciones:

1. Encender el PLC y conectar el cable serial del PLC al computador.
2. Correr el software RSLinx y dar clic en *Configure Drivers*

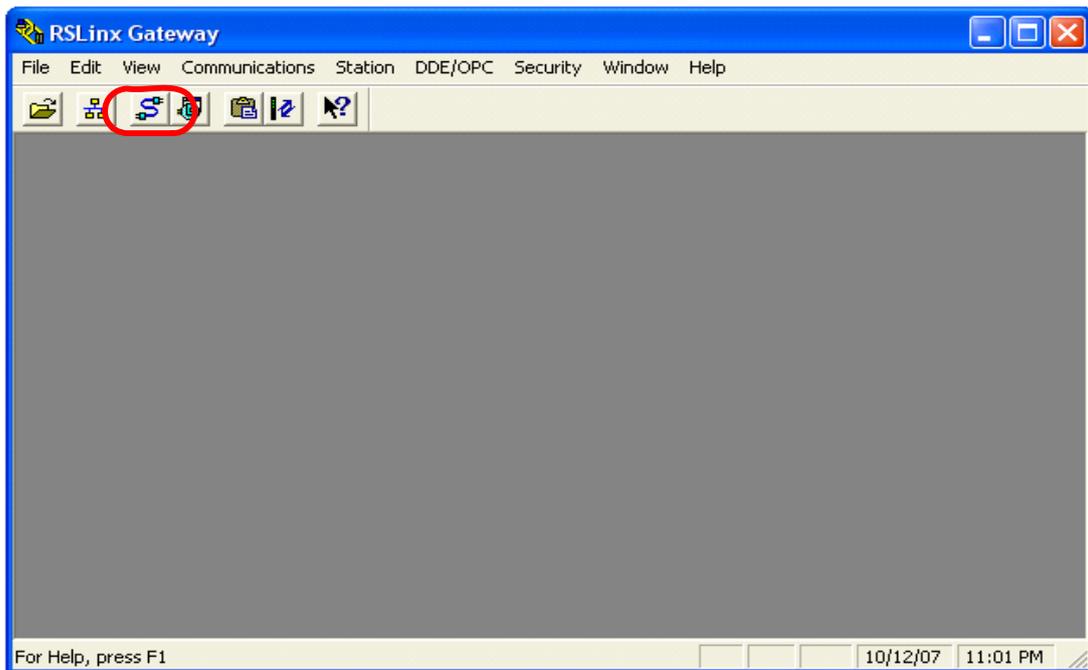


Figura. 3.2. Configuración del Driver de comunicación serial RS232 (1)

3. En la pantalla que aparece, seleccionar el driver de comunicación “*RS-232 DF1 devices*” y hacer clic en *Add New* como se indica en la figura 3.3.

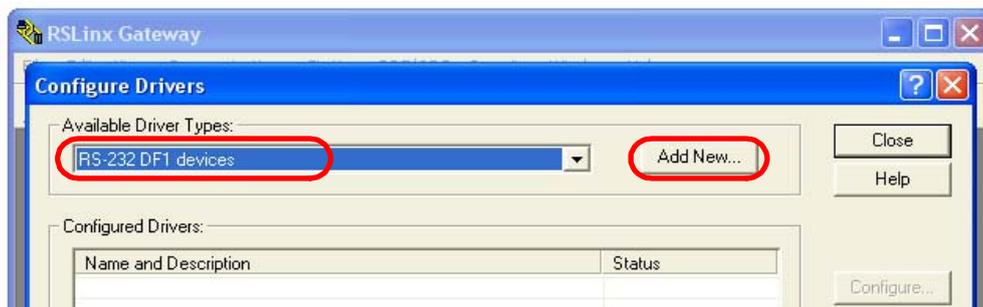


Figura. 3.3. Configuración del Driver de comunicación serial RS232 (2)

4. Escribir un nuevo nombre para el Driver o dejar el que viene por defecto y presionar *OK*.



Figura. 3.4. Configuración del Driver de comunicación serial RS232 (3)

5. Seleccionar el puerto de serial al que se encuentra conectado el PLC, seleccionar el dispositivo *SLC-CHO/Micro/PanelView*, presionar *Auto-Configure*, una vez que salga el mensaje de configuración exitosa presionar *OK*

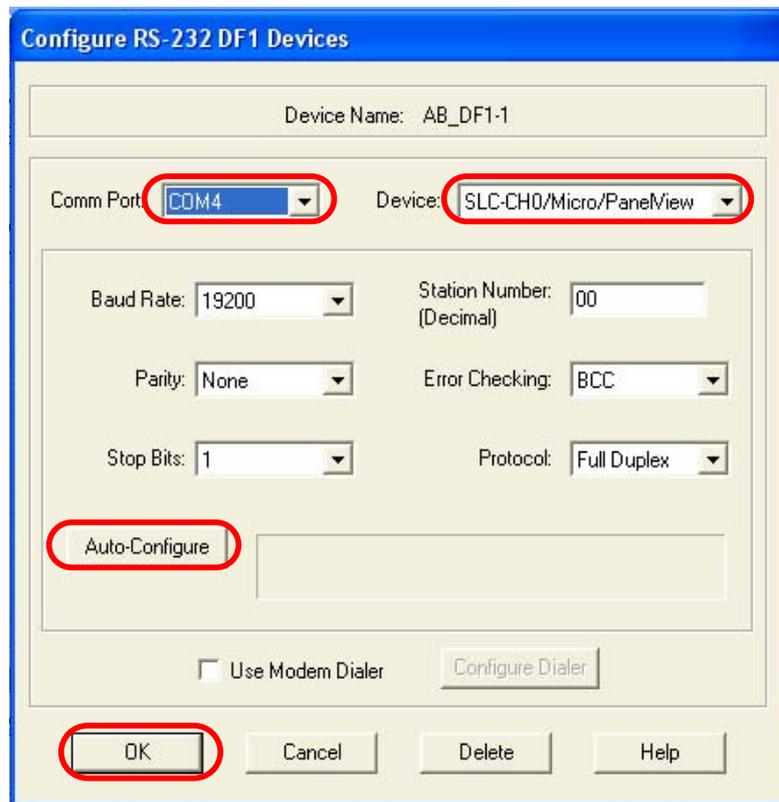


Figura. 3.5. Configuración del Driver de comunicación serial RS232 (4)

**Nota:** Si la auto configuración no es exitosa podrían aparecer los siguientes mensajes:

a. *Failed to find baud and parity! Check all cables and switch settings!*

Esto puede indicar que el Puerto serial para la computadora no esta habilitado, el cable esta dañado o no esta conectado correctamente, o el protocolo para el canal del procesador no esta configurado para comunicación RS-232 full duplex.

b. *Unable to verify settings due to packet time-out! (or Unable to verify settings due to a NAK!)* Check all cables and configuration and try again.

Estos dos mensajes usualmente indican que el canal en el procesador no esta configurado para comunicación RS-232 full duplex.

c. *Unable to open specified port for configuration testing!*

Hay conflicto en el Puerto serial –El esta siendo usado por otro driver en RSLinx o por un dispositivo diferente tal como un modem.

6. Verificar que el driver esté funcionando correctamente, y presionar **Close** con lo que se volverá a la pantalla de inicio.

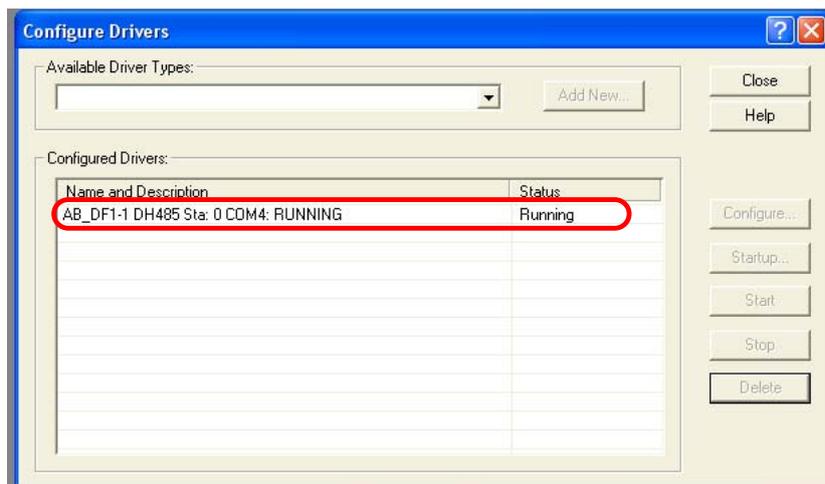


Figura. 3.6. Verificación del driver de comunicación serial RS232

### Configuración del Driver Ethernet de comunicación entre el PLC y la PC

1. Correr el software RSLinx y dar clic en **Configure Drivers**



Figura. 3.7. Configuración del Driver de comunicación Ethernet (1)

2. En la pantalla que aparece, seleccionar el driver de comunicación “*Ethernet devices*” y hacer clic en *Add New* como se indica en la figura 3.8.

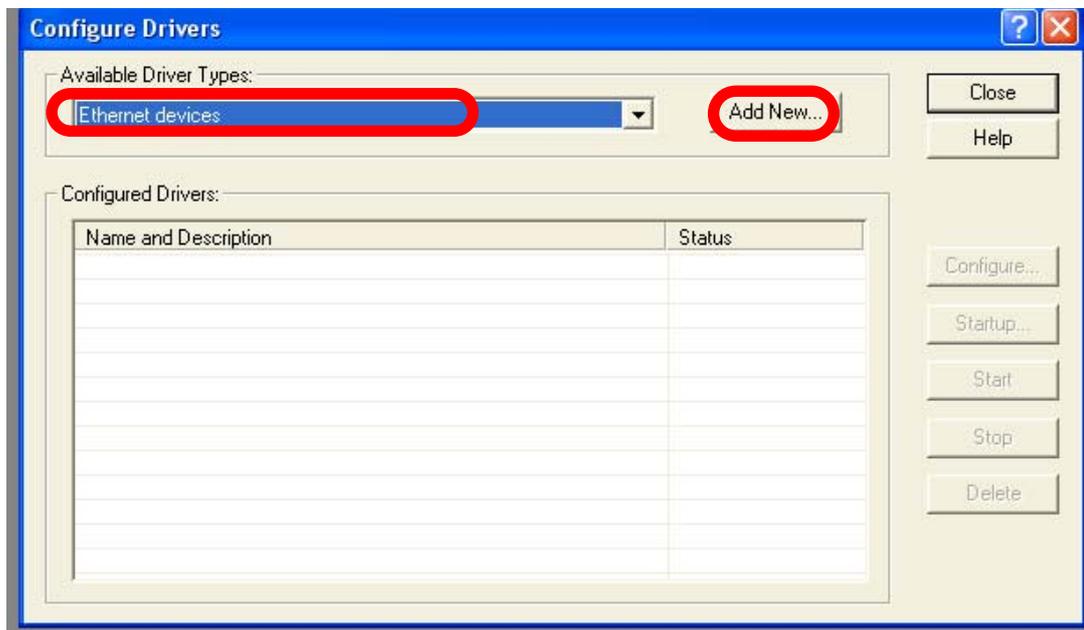


Figura. 3.8. Configuración del Driver de comunicación Ethernet (2)

3. Escribir un nuevo nombre para el Driver o dejar el que viene por defecto y presionar *OK*.

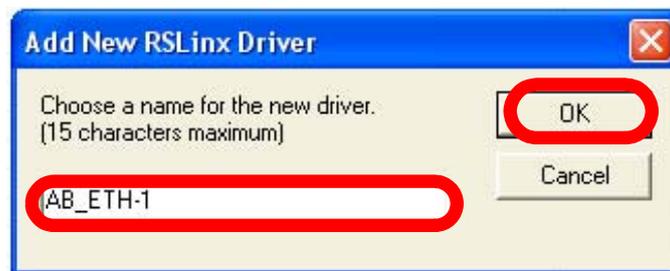


Figura. 3.9. Configuración del Driver de comunicación Ethernet (3)

4. Ingrese las direcciones IP, presionando *Add New* cada vez que se requiera ingresar una nueva dirección. Se deben ingresar la direcciones de todos los equipos que se vayan a conectar a la red (PLC's, PC, Módulos de interfase Ethernet, etc.) y finalmente presionar *OK*

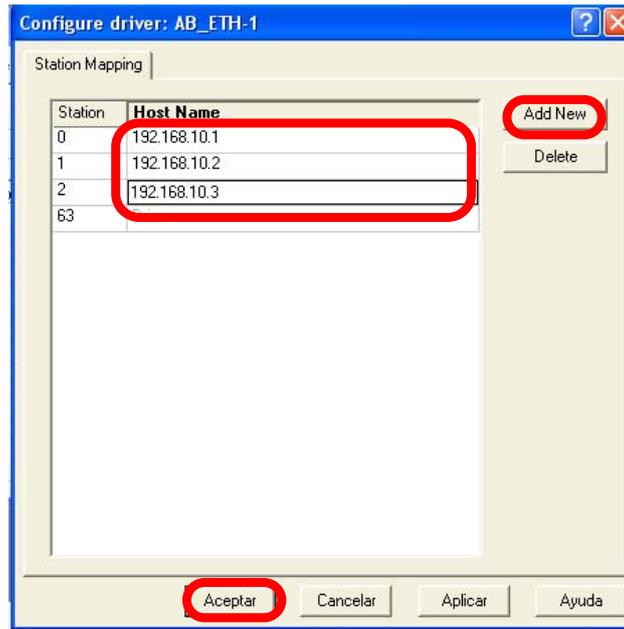


Figura. 3.10. Configuración del Driver de comunicación Ethernet (4)

5. Finalmente, verificar que el driver esté funcionando correctamente, y presionar *Close* con lo que se volverá a la pantalla de inicio.

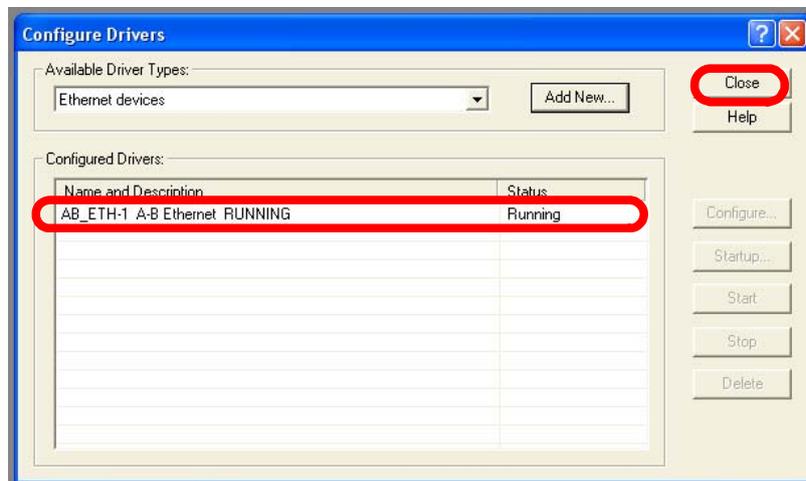


Figura. 3.11. Verificación del driver de comunicación serial Ethernet

## 3.2. SOFTWARE RSLOGIX 500\*

### 3.2.1. INTRODUCCIÓN

El software RSLogix 500 es un paquete que corre en Windows de 32 bits, que permite la programación de lógica Ladder para los procesadores SLC 500 y MicroLogix. RSLogix 500 es compatible todos los programa de SLC 500 y MicroLogix creados con cualquier otro paquete de programación de Rockwell Software's

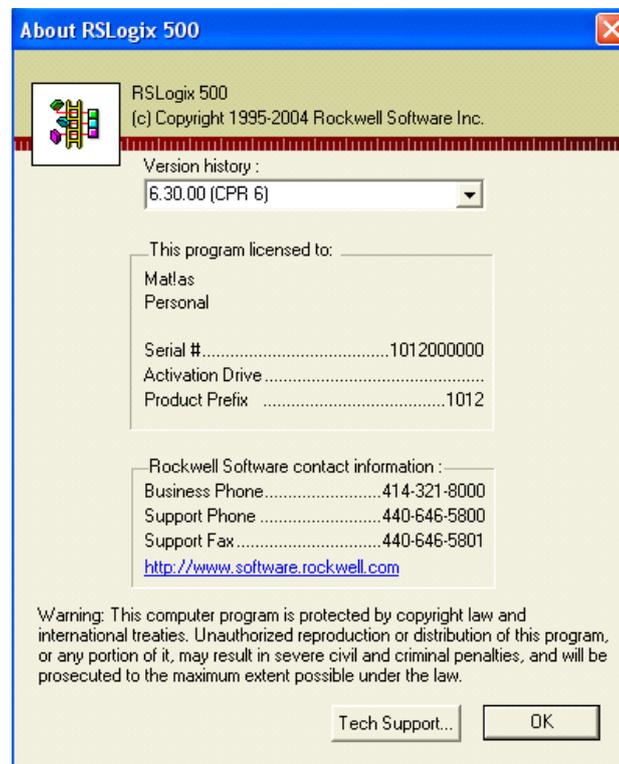
Las principales características que el software RSLogix 500 incluye son:

- Un editor ladder de forma libre que permite al programador concentrarse en la lógica de la aplicación en lugar de la sintaxis de programación.
- Un poderoso verificador de proyecto, que permite crear una lista de los errores que se presenten en la programación para la posterior verificación y corrección de los mismos. Edición drag-and-drop para mover rápidamente elementos.
- Asistente (Wizard) de direcciones que facilita el ingreso de direcciones y reduce los errores involuntarios.
- Interfase llamada "project tree" que permite acceder a todas la carpetas y archivos contenidos en el proyecto
- Monitor de datos que permite ver elementos de datos separados y observar interacciones.

---

\*La información expuesta en esta sección fue extraída del texto 19 citado en las referencias bibliográficas.

La Escuela Politécnica de Ejército (ESPE) cuenta con la versión **RSLogix 500 6.30.00 (CPR 6)** como se muestra en la figura 3.12.



**Figura. 3.12. Versión de RSLogix 500 que posee la ESPE**

### 3.2.2. REQUERIMIENTOS DEL SISTEMA

Para utilizar efectivamente RSLogix 500, su computador personal debe reunir los siguientes requerimientos mínimos de hardware y software:

#### Requerimientos mínimos de Hardware

- Procesador Pentium II (500 MHz) o superior
- 128 MB de memoria RAM.
- 45 MB de espacio disponible en disco duro.
- Adaptador gráfico 256-color SVGA con resolución de 800x600
- CD-ROM
- Un Mouse u otro dispositivo apuntador Windows-compatible.

## Requerimientos mínimos de Software

- Uno de los siguientes sistemas operativos:
  - Microsoft Windows 98
  - Microsoft Windows 2000
  - Windows NT 4.0 con Service Pack 6 o superior
  - Windows XP
  
- RSLinx Classic versión 2.31.00 o superior.

### 3.2.3. CONFIGURACIÓN Y CREACIÓN DE UN NUEVO ARCHIVO

Para crear un programa ladder con el software RSLogix500, se deberá seguir los siguientes pasos:

1. Configurar el driver que va a utilizar el PLC (Explicado detalladamente en la sección anterior).
2. Abrir el programa RSLogix500 y presionar el icono de *New* como se muestra en la figura 3.13.

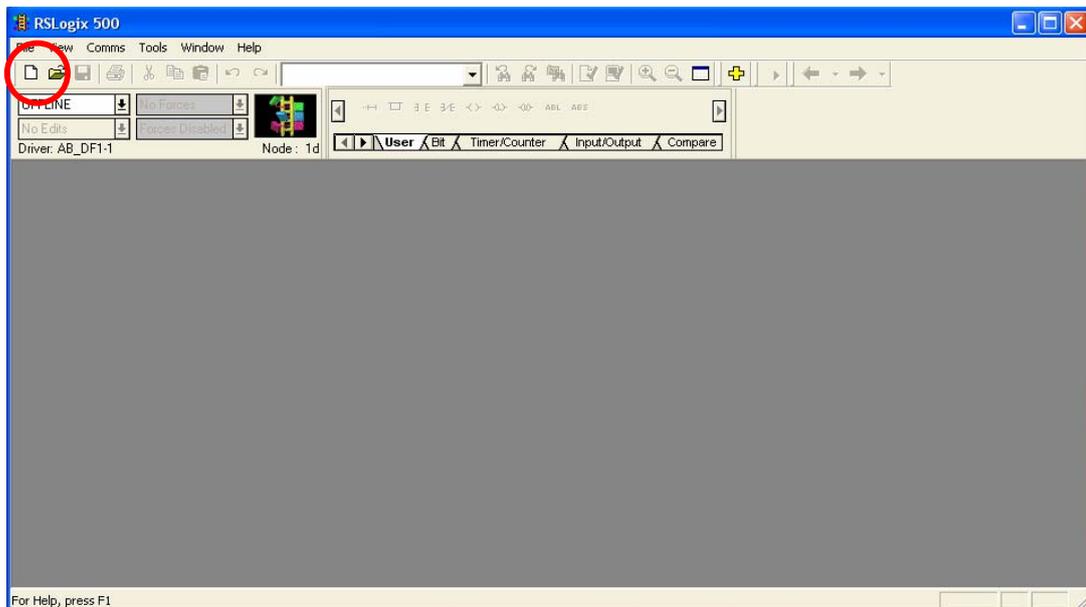


Figura. 3.13. Creación de un nuevo archivo en RSLogix 500

3. Poner nombre al Procesador, escoger el tipo de PLC que se tiene de entre la lista, seleccionar el Driver (antes configurado) y presionar **OK**

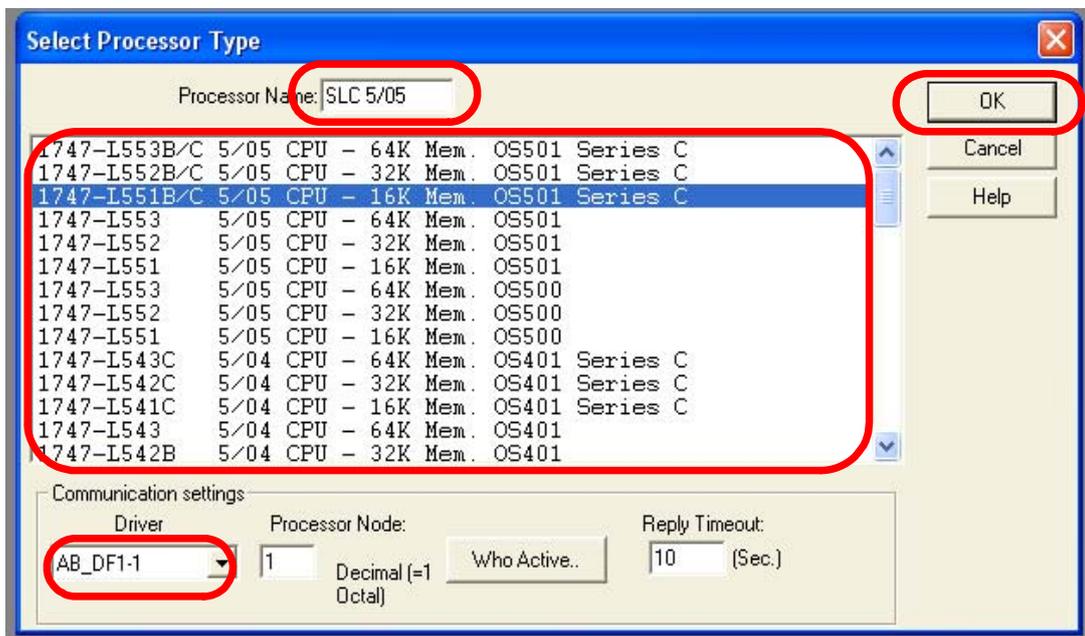


Figura. 3.14. Selección del tipo de PLC en RSLogix 500

4. En la ventana de la izquierda, abrir la carpeta **Controller** y dar clic en **IO Configuration**

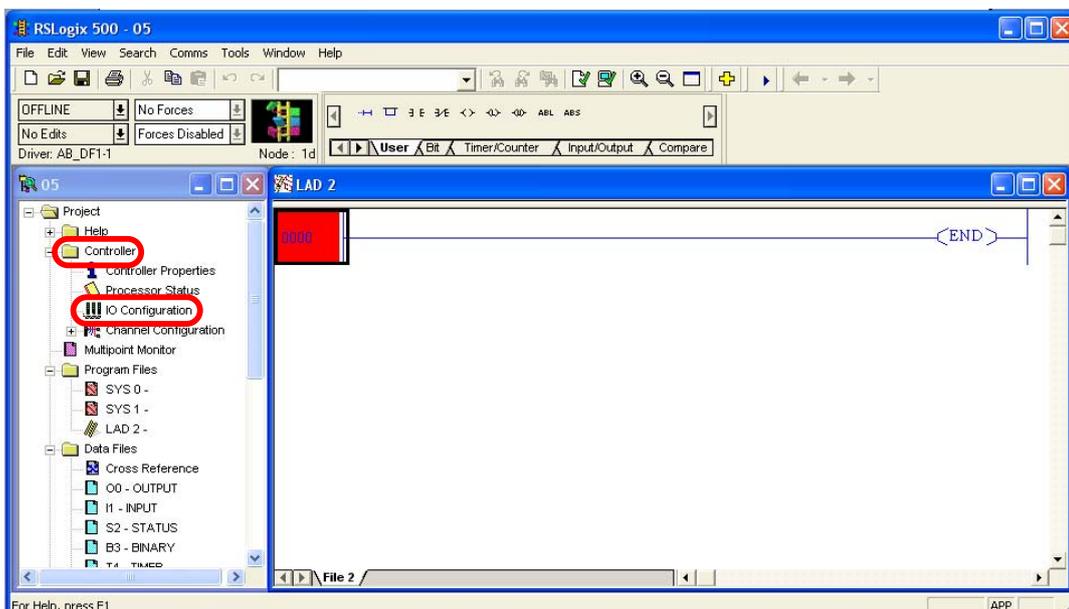


Figura. 3.15. Configuración de los módulos I/O en RSLogix 500 (1)

5. Dar clic en **Read IO Config** para leer automáticamente la configuración de los módulos instalados. Si se quiere hacer de manera manual (si no se dispone al momento del PLC), se debe seleccionar de uno en uno los módulos de entre la lista de la izquierda.

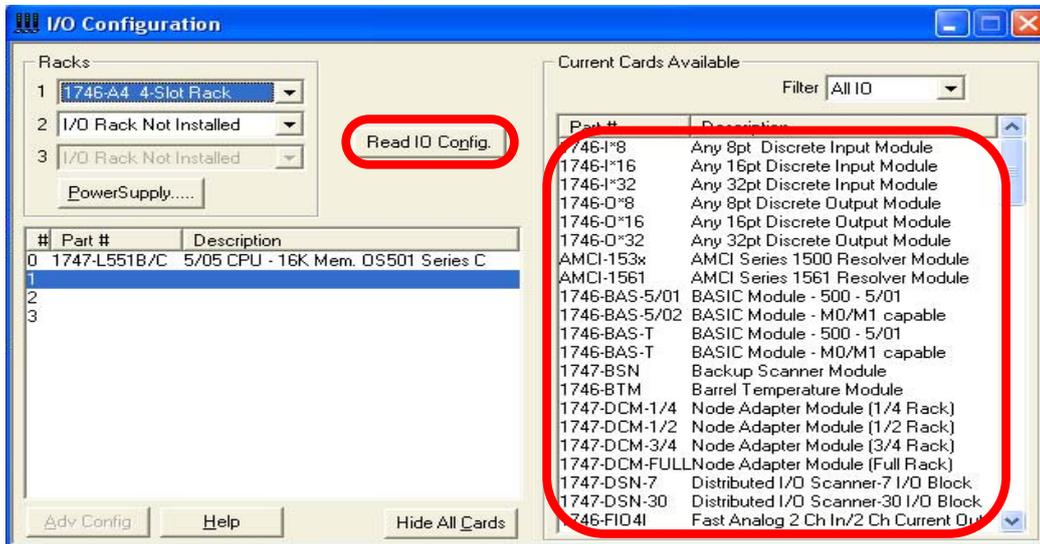


Figura. 3.16. Configuración de los módulos I/O en RSLogix 500 (2)

6. Si se presionó **Read IO Config**, seleccionar el Driver y presionar **Read IO Config** para que el software lea directamente del PLC la ubicación y el tipo de módulos que se encuentran instalados.

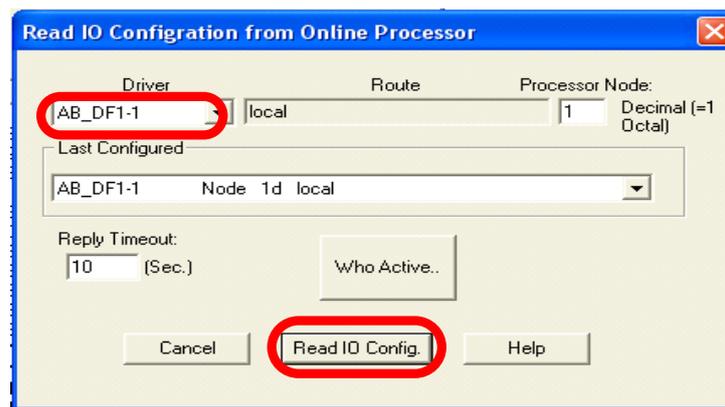


Figura. 3.17. Configuración de los módulos I/O en RSLogix 500 (3)

7. Cerrar la ventana **IO Configuration** y empezar a programar.
8. Una vez terminado el programa, se debe guardarlo en el computador y posteriormente **bajarlo** al PLC, seleccionando la opción **Download**

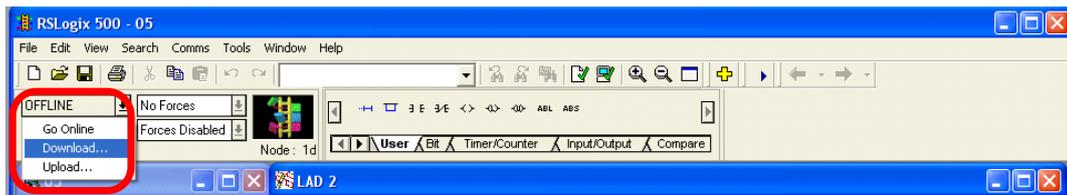


Figura. 3.18. Descargar el programa ladder en el PLC

### Configuración del canal 0 (DF1 Full-duplex) de comunicación del PLC

Para configurar correctamente el canal 0 de comunicación del PLC se debe seguir las siguientes instrucciones:

1. Dar clic en **Controller** (para desplegar las opciones) y dar doble clic en **Channel Configuration**

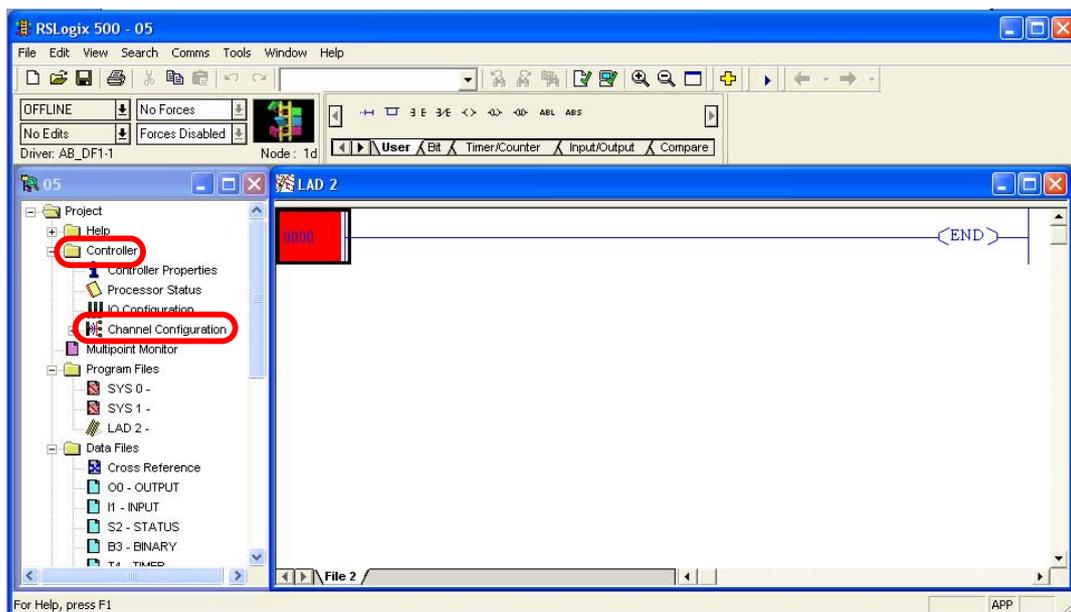


Figura. 3.19. Configuración del canal 0 DF1 Full-Duplex del PLC (1)

2. En el casillero **Diagnostic File** (en la sección Channel 0) asignar un número que sea mayor al número de Data Files que se estén utilizando (ver en la lista de la izquierda de la imagen). Para nuestro caso es 9. Luego dar clic en la pestaña **Chan 0 - System**

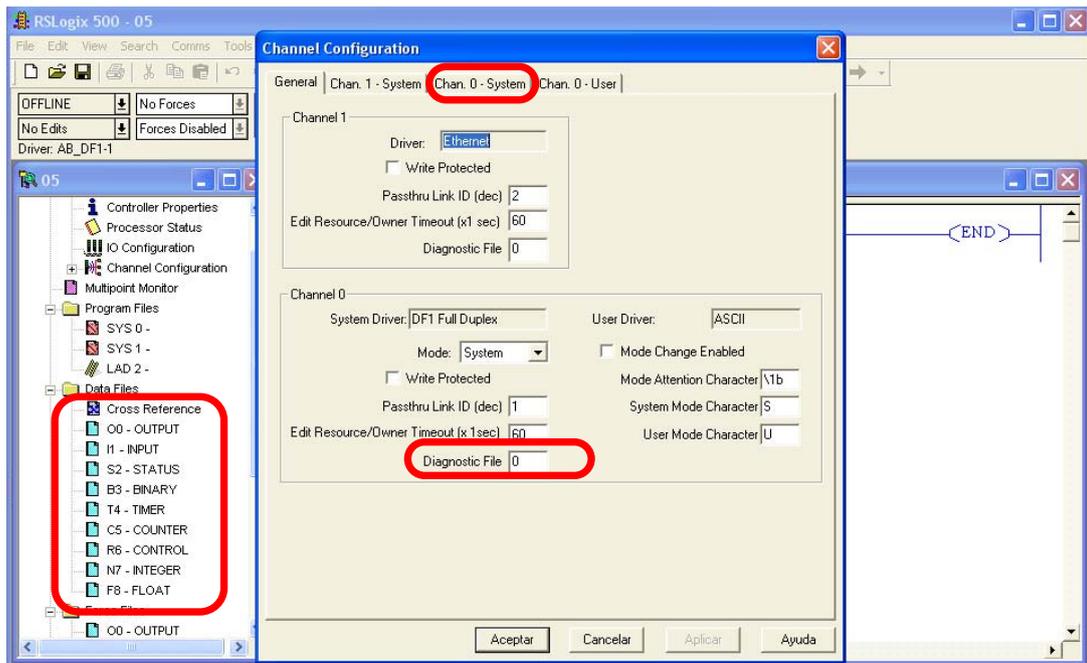


Figura. 3.20. Configuración del canal 0 DF1 Full-Duplex del PLC (2)

3. Aparecerá la siguiente pantalla en la que se puede configurar los parámetros de comunicación, para posteriormente presionar *Aceptar*.

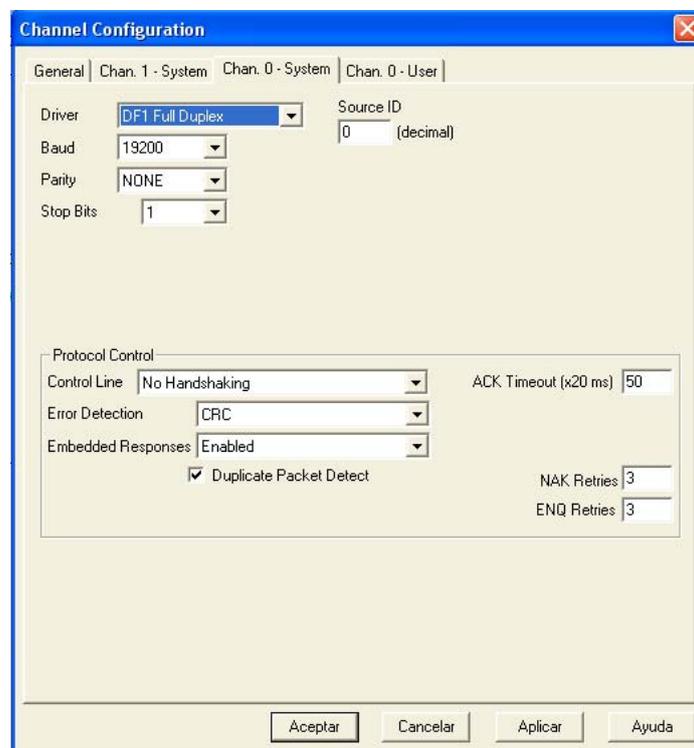


Figura. 3.21. Configuración del canal 0 DF1 Full-Duplex del PLC (3)

## Configuración del canal 1 (Ethernet) del PLC SLC 5/05

Para configurar correctamente el canal 1 (Ethernet) de comunicación del PLC SLC 5/05 se debe seguir las siguientes instrucciones:

1. Dar clic en **Controller** (para desplegar las opciones) y dar doble clic en **Channel Configuration**

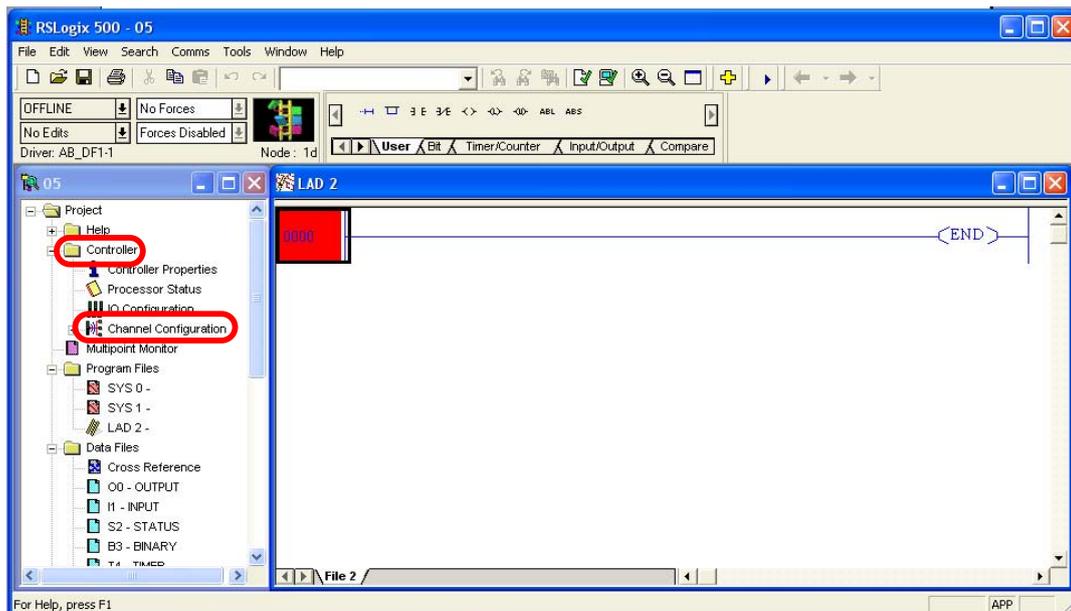


Figura. 3.22. Configuración del canal 1 Ethernet del PLC (1)

2. En el casillero Diagnostic File (de la sección Channel 1) asignar un número que sea mayor al número de Data Files que se estén utilizando (ver en la lista de la izquierda de la imagen). Para nuestro caso es 9. Luego dar clic en la pestaña Chan 1 – System.

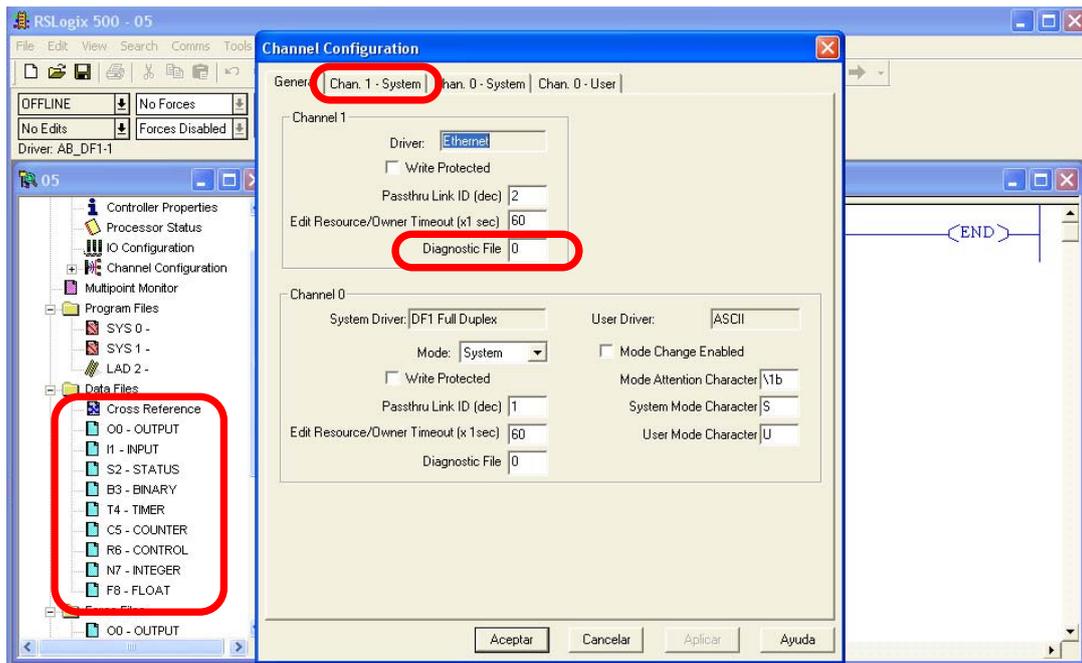


Figura. 3.23. Configuración del canal 1 Ethernet del PLC (2)

3. Ingrese la dirección IP y la máscara de subred, y finalmente presionar *Aceptar*

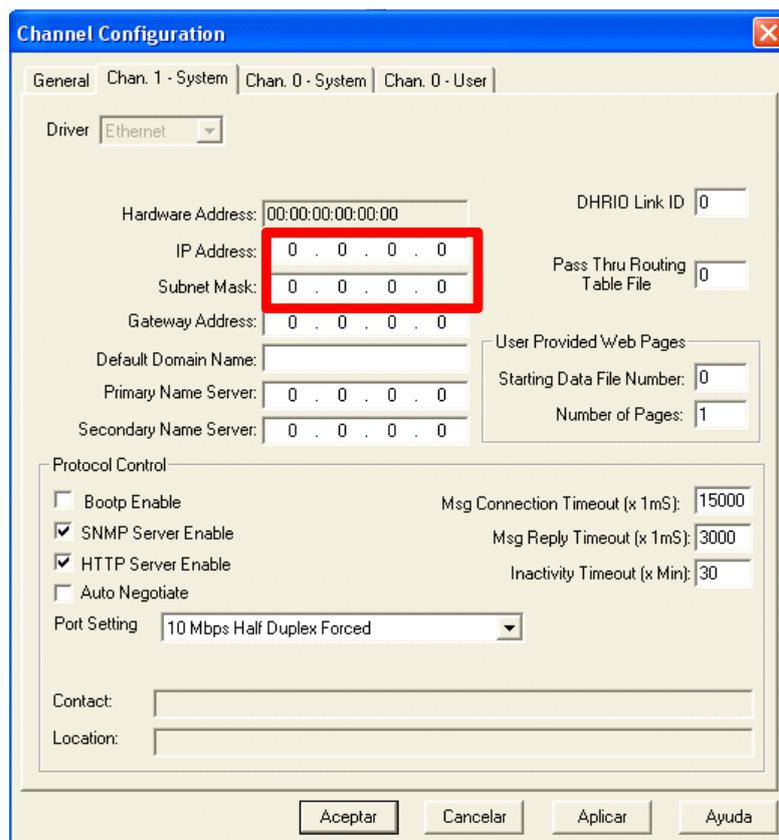


Figura. 3.24. Configuración del canal 1 Ethernet del PLC (3)

### 3.3. SOFTWARE RSVIEW32\*

#### 3.3.1. INTRODUCCIÓN

RSView32 es un software basado en Windows para el desarrollo y la ejecución de aplicaciones de interfaz operador-máquina “HMI”. Le proporciona todas las herramientas que necesita para crear y ejecutar eficazmente las aplicaciones de monitoreo y control supervisor.

**RSView32 Works.-** RSView32 Works contiene los editores necesarios para generar una aplicación completa de interfaz operador-máquina y contiene el software requerido para ejecutar las aplicaciones generadas. Utilice los editores para crear aplicaciones tan simples o sofisticadas como las necesite. Cuando haya terminado de desarrollar su aplicación, cambie al modo de ejecución o utilice RSView32 Runtime (que viene incluido junto con RSView32 Works y utiliza menos memoria) y ejecute su aplicación.

Con RSView32, puede:

- Utilizar la capacidad del contenedor RSView32 ActiveX y OLE para aprovechar la tecnología avanzada. Por ejemplo, puede incrustar RSTools™, Visual Basic® u otros componentes ActiveX en las pantallas gráficas de RSView32 para ampliar las capacidades de éste.
- Crear y editar pantallas con las herramientas propias de los programas de Microsoft que Ud. está utilizando. Mediante sofisticados gráficos y animaciones basados en objetos, más las técnicas simples de arrastrar, colocar y cortar-pegar, se simplifica la configuración de la aplicación.
- Utilizar el modelo de objetos RSView32 y VBA para compartir datos con otros programas de Windows, tales como Microsoft Access y SQL Server, interactuar con otros programas de Windows tales como Microsoft Excel, así como personalizar y extender RSView32 adaptándolo a sus necesidades específicas.
- Utilizar gráficos de las bibliotecas de gráficos RSView32 o importar archivos de otros paquetes de dibujo tales como CorelDRAW™ y Adobe® Photoshop®.

---

\*La información expuesta en esta sección fue extraída de los textos 20 y 16 citados en las referencias bibliográficas.

- Desarrollar rápidamente su aplicación utilizando herramientas de productividad RSView32 tales como el Asistente de comandos, el Examinador de tags y Object Smart Path™ (OSP) - (ruta inteligente de objeto).
- Evitar introducir información repetida. Importe una base de datos de un PLC o SLC de Allen-Bradley con el Examinador de bases de datos de PLC.
- Utilizar las funciones de alarmas de RSView32 para monitorear incidentes ocurridos en el proceso con varios niveles de gravedad. Cree resúmenes de varias alarmas para obtener datos específicos sobre las alarmas en lugar de examinar las alarmas de la totalidad del sistema.
- Crear tendencias que muestren variables del proceso graficadas en relación al tiempo. Muestre datos en tiempo real o histórico hasta con 16 plumas (tags) en cada tendencia.
- Registrar datos simultáneamente en varios archivos de registro o bases de datos ODBC remotas para proporcionar diversos registros de los datos de producción. Lleve los datos registrados directamente a programas de otros fabricantes tales como Microsoft Excel y Seagate Crystal Reports™ sin necesidad de convertir los archivos.
- Bloquear el sistema por medio de la desactivación de las claves de Windows de modo que los usuarios sólo puedan utilizar la aplicación RSView32.

**RSView32 Runtime.-** RSView32 Runtime contiene el software necesario para ejecutar aplicaciones RSView32. RSView32 Runtime también contiene un subconjunto de editores RSView32 Works, de manera que usted pueda editar partes seleccionadas de un proyecto durante el tiempo de ejecución. RSView32 Runtime puede obtenerse en paquete junto con RSView32 Works o puede comprarse por separado. Con RSView32 Runtime, su aplicación utiliza menos memoria para la ejecución.

La Escuela Politécnica de Ejército (ESPE) cuenta con la versión **RSView32 7.20.00 (CPR 7)** como se muestra en la figura 3.25. Además cuenta con una llave de activación RSView.D32K la cual permite trabajar hasta con 32000 tags.

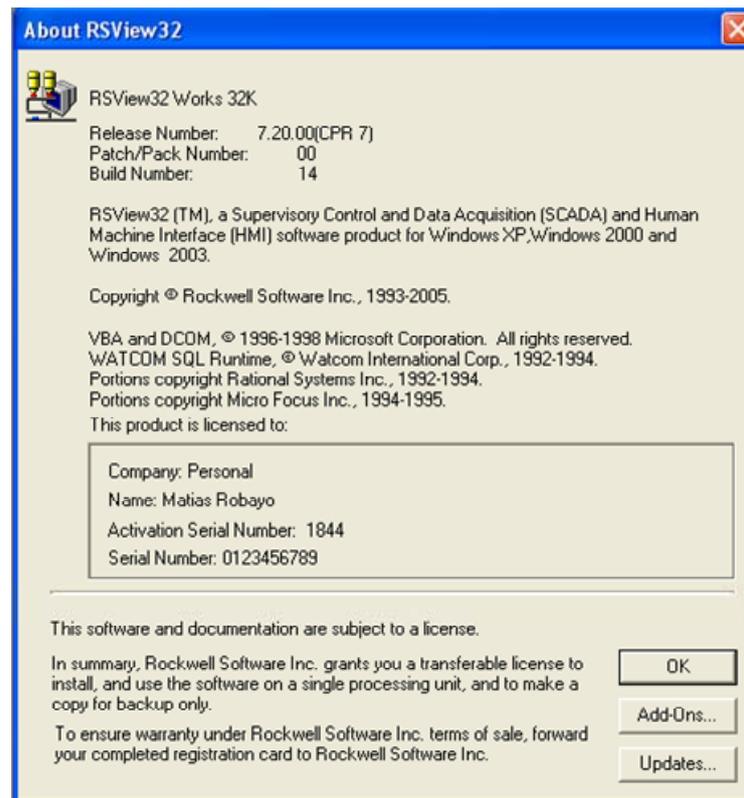


Figura. 3.25. Versión del software RSView32 que posee la ESPE

### 3.3.2. REQUERIMIENTOS DEL SISTEMA

El hardware y software a usar con RSView32 dependen de lo que el proyecto le exigirá al sistema. Cuanto mayor sea la exigencia, más poderoso será el sistema que necesite. Las siguientes recomendaciones se basan en la experiencia de campo. Es posible que su aplicación funcione en una plataforma de menor potencia que la recomendada. En las siguientes tablas, los requisitos que el proyecto exige del sistema están representados por puntos. Sume los puntos del paso 1 y luego use la tabla del paso 2 para determinar el hardware y software necesario.

#### Paso 1. Sume sus puntos

Para	Si esta	Puntos
RSView 32 Project	Ejecutando pantallas o editando su proyecto usando RSView 32 Works.	2
	Usando más de 30.000 tags	2
	Usando más de 5.000 tags	1

	Cambiando pantallas con un promedio de más de 5 veces por minuto	1
	Usando más de 200 objetos en pantallas	2
	Usando controles ActiveX en las pantallas	2
<b>Alarmas</b>	Monitoreando más de 1.500 tags para las alarmas	2
	Monitoreando más de 300 tags para las alarmas	1
<b>Sistema de pantalla activa, clientes activos Simultáneamente en OPC o DDE</b>	Usando generalmente más de 5 clientes activos	2
	Usando por lo menos un cliente	1
<b>Registro de datos o tendencias históricas</b>	Registrando menos de 100 tags en un modelo	1
	Registrando entre 100 y 1.000 tags en un modelo	2
	Registrándose en más de 1 modelo	2
<b>Detector de eventos, tags derivados o tendencias en tiempo real</b>	Usando cualquiera	1
<b>VBA</b>	Usando cualquiera	2
<b>Adiciones</b>	Usando cualquiera, para cada uno	1
<b>Sume sus puntos aquí</b>	_____ →	

## Paso 2. Determine la configuración recomendada del sistema

<b>Puntos</b>	<b>Mínimo hardware necesario</b>	<b>Mínimo software necesario</b>
<b>8 o más</b>	Pentium II 400 MHz RAM de 128 MB	Windows 2000 Professional o Windows NT Workstation 4.0 con Service Pack 4 ó Service Pack 6
<b>6 a 7</b>	Pentium II 300 MHz RAM de 128 MB	Windows 2000 Professional o Windows NT Workstation 4.0 con Service Pack 4 ó Service Pack 6
<b>3 a 5</b>	Pentium 200 MHz RAM de 64 MB	Windows 2000 Professional o Windows NT Workstation 4.0 con Service Pack 4 ó Service Pack 6
<b>1 a 2</b>	Pentium 100 MHz	Windows 2000 Professional o Windows NT

RAM de 24 MB

Workstation 4.0 con Service Pack 4 ó Service Pack 6  
ó Windows 9x

### 3.3.3. CONFIGURACIÓN Y CREACIÓN DE UNA HMI

A continuación se describirán los pasos principales para configurar y crear una HMI utilizando el software RSView32.

#### Crear un archivo nuevo y configurar de los canales de comunicación directa

Para crear y nuevo archivo y configurar correctamente el canal de comunicación, se deben seguir las siguientes instrucciones:

1. Configurar el driver con el Software RSLinx
2. Correr el software RSView32 y dar clic en *New*.

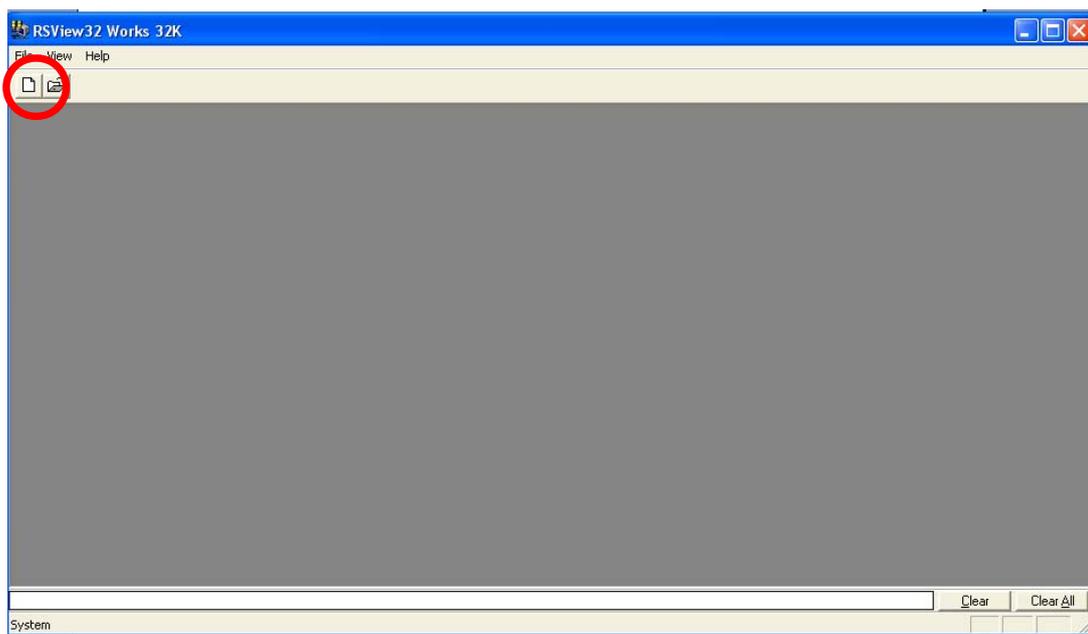


Figura. 3.26. Creación de un archivo nuevo en RSView32 (1)

3. Seleccionar la ubicación donde se quiere crear el archivo, asignar un nombre y presionar *Abrir*.

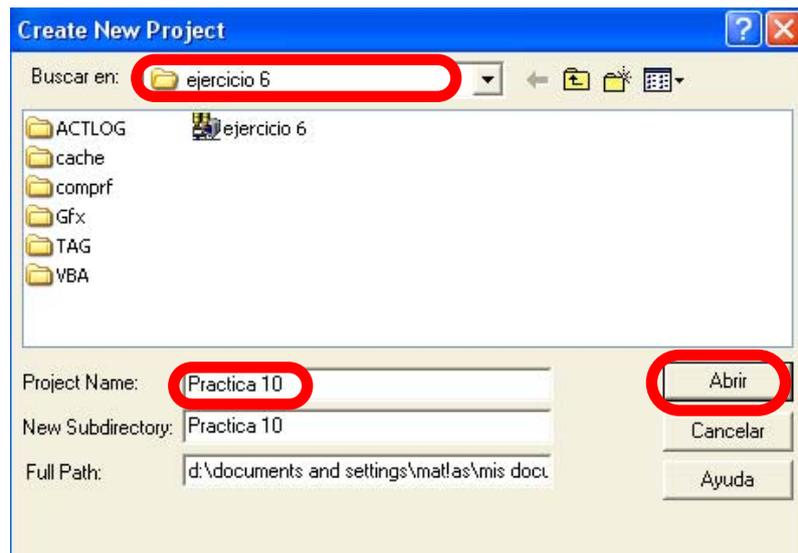


Figura. 3.27. Creación de un archivo nuevo en RSVIEW32 (2)

4. Dar clic en *System* para desplegar las opciones, dar doble clic en *Channel*

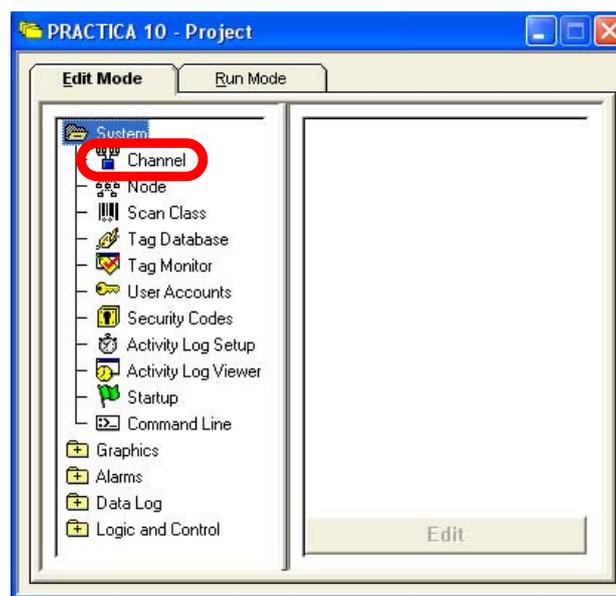


Figura. 3.28. Configuración del canal de comunicación en RSVIEW32 (1)

5. Escoger el número de canal (en este caso 1), en *Network Type* escoger DH-485 (Para conectarse a un SLC con un puerto RS-232), en *Messages* escriba un número entre 1 y 10 (este es el número de mensajes que el software puede enviar por un canal, antes de requerir una respuesta) se recomienda colocar un número alto (10), y si se producen errores de comunicación reducirlo gradualmente puesto que éste número dependerá de la capacidad de

almacenamiento del buffer de PLC. Para configurar un canal Ethernet en *Network Type* escoger TCP/IP. Finalmente presione OK

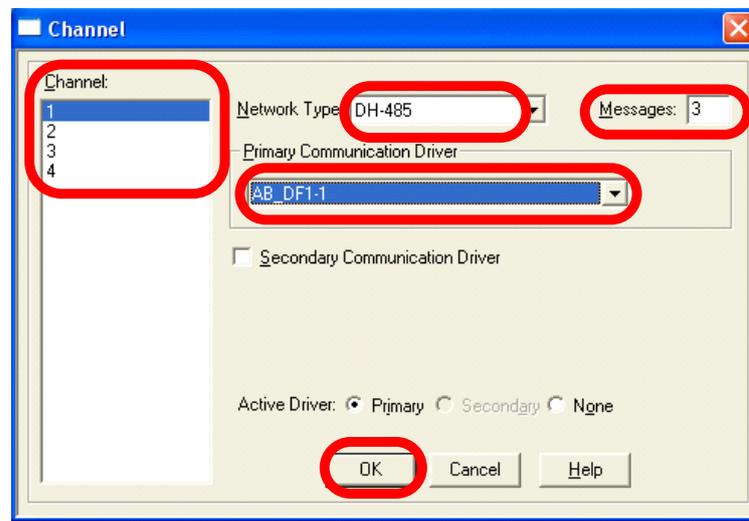


Figura. 3.29. Configuración del canal de comunicación en RSView32 (2)

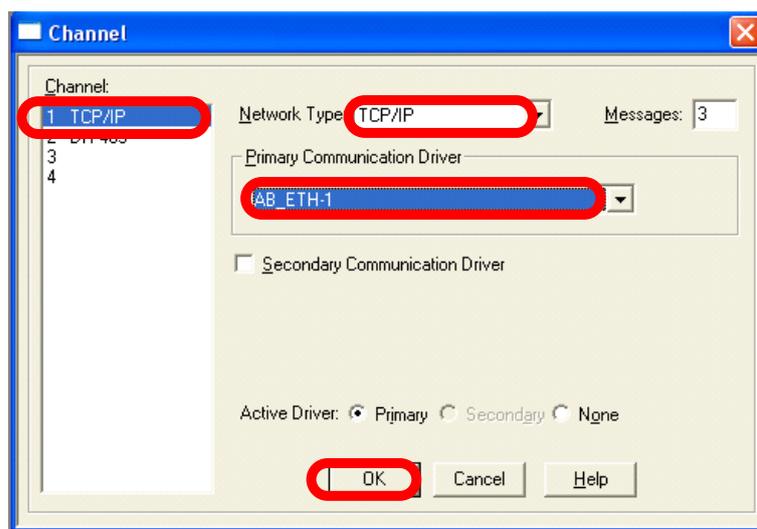


Figura. 3.30. Configuración del canal de comunicación en RSView32 (3)

6. Dar doble clic en *Node*

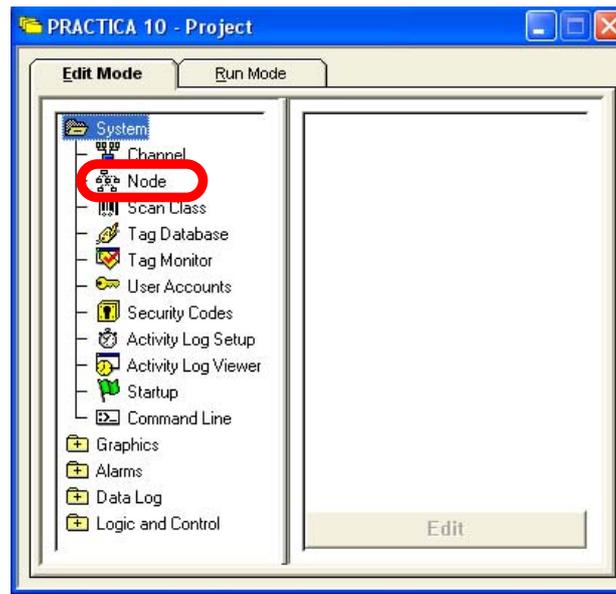


Figura. 3.31. Configuración del nodo de comunicación en RSVIEW32 (1)

7. Seleccionar *Direct Driver*, en *Name* asignar un nombre al nodo, en *Channel* escoger el canal creado anteriormente, en *Station* presionar el botón de explorar para seleccionar el PLC al que se quiere conectar, en *Type* seleccionar el tipo de PLC *SLC 5 (Enhanced)*, y presionar *Aceptar*:

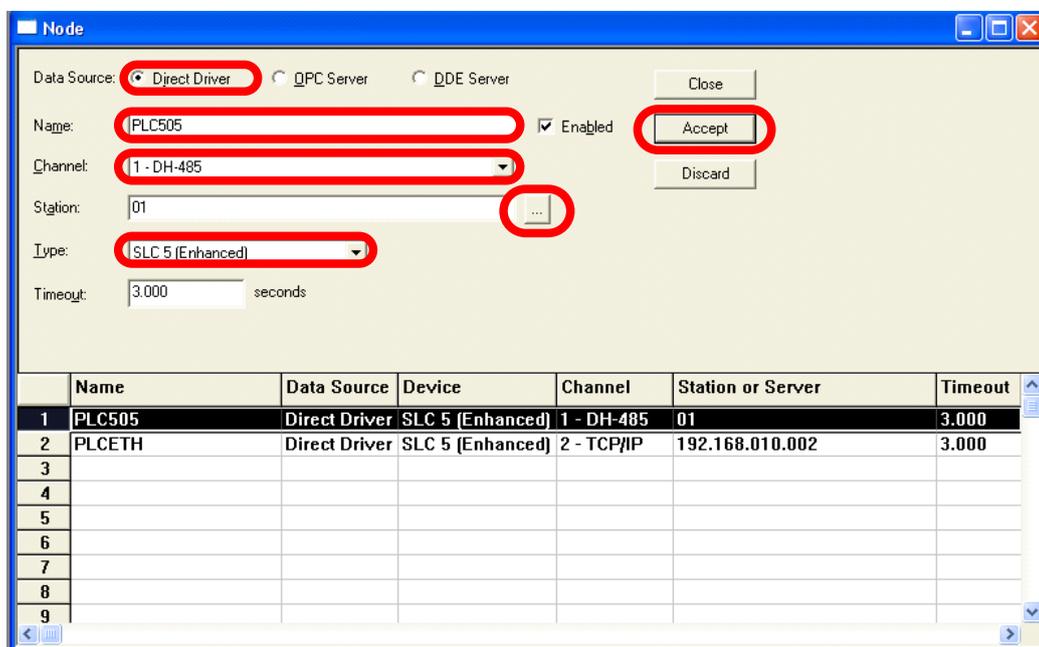


Figura. 3.32. Configuración del nodo de comunicación en RSVIEW32 (2)

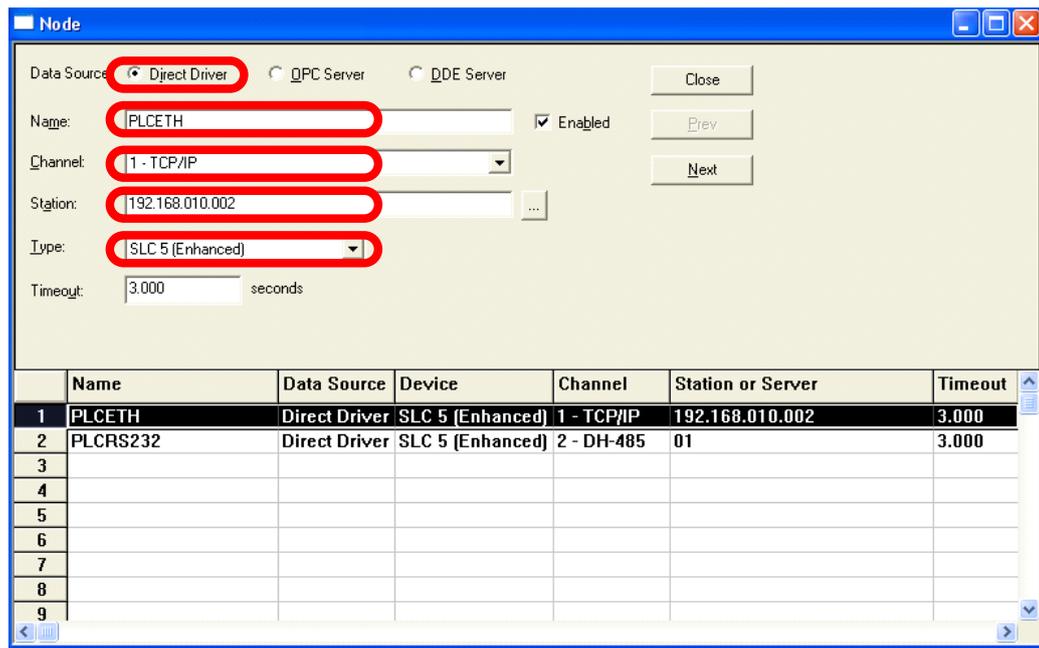


Figura. 3.33. Configuración del nodo de comunicación en RSView32 (3)

## Crear Tags

1. Dar doble clic en *Tag Database*

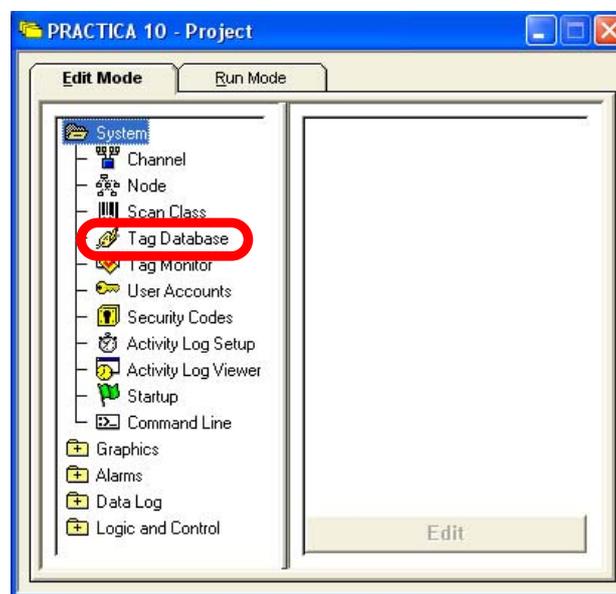


Figura. 3.34. Creación de tags en RSView32

2. En *Name* asignar un nombre (Max. 255 caracteres, no podrá contener espacio Ej: Motor1, Motor-1 ó Motor\_1. Pero NO *Motor 1*). En *Type* seleccionar el tipo de tag que se requiere. En *Description* se puede escribir una pequeña

descripción del tag, esta es opcional. En **Data Source** seleccionar *Memory* si el tag recibe únicamente datos internos del software RSVIEW32, seleccionar *Device* si el tag recibe datos de una fuente externa (PLC), para lo que se desplegará las siguientes casillas:

- Node Name: Seleccionar el nodo creado anteriormente.
- Address: Especificar la dirección de los datos en el PLC a los cuales el tag debe estar conectado. La sintaxis dependerá del tipo de PLC que se esté utilizando.

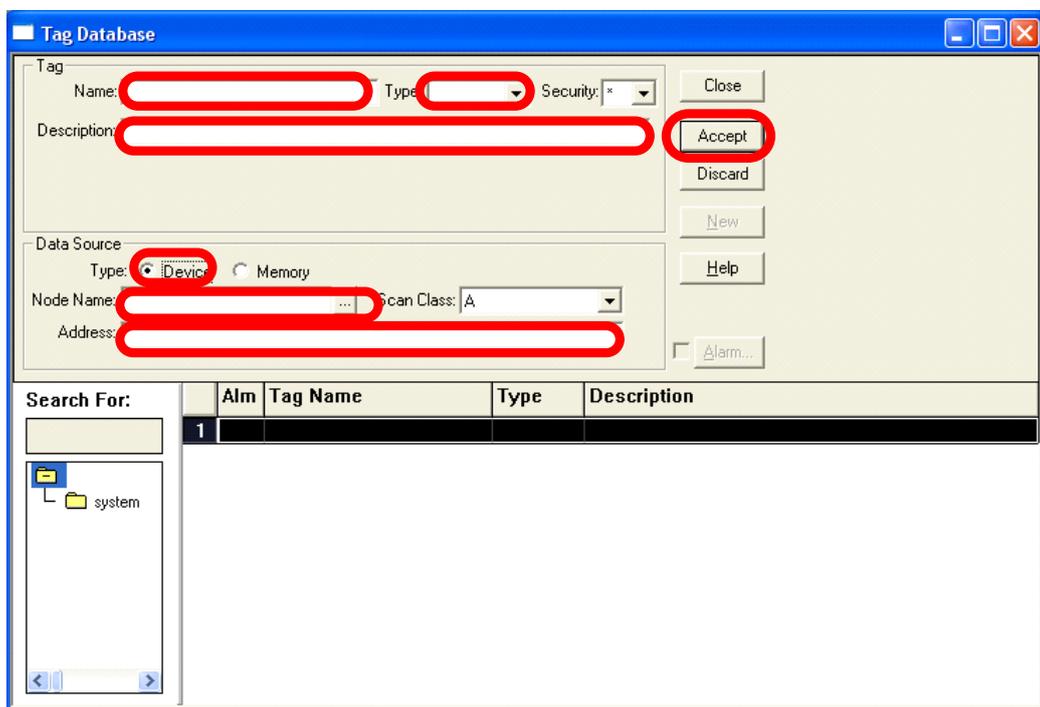


Figura. 3.35. Configurar tags en RSVIEW32



*Si se requiere activar o desactivar una salida del PLC a través de un “interruptor” dentro de la HMI, éste no deberá apuntar directamente a la salida del PLC, sino deberá apuntar a un registro de bit, el cual dentro del programa ladder controlará la salida del PLC.*

## Crear Pantallas Gráficas

1. Dar clic en **Graphics** para desplegar las opciones, dar doble clic en **Display**

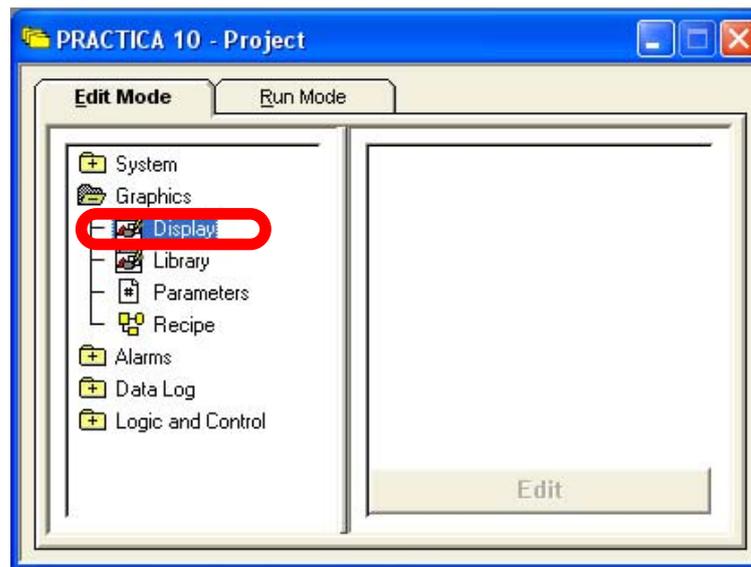


Figura. 3.36. Creación de pantallas gráficas en RSView32 (1)

2. Aparecerá una ventana en blanco, que será en donde se creará la HMI. Además se desplegará la barra de herramientas de dibujo, colores, alineamiento, etc. (sin no aparecen, seleccionarlas en el menú *View*). Adicionalmente, si se quiere insertar gráficos prediseñados se debe dar doble clic en *Library* y copiarlos en nuestra ventana.

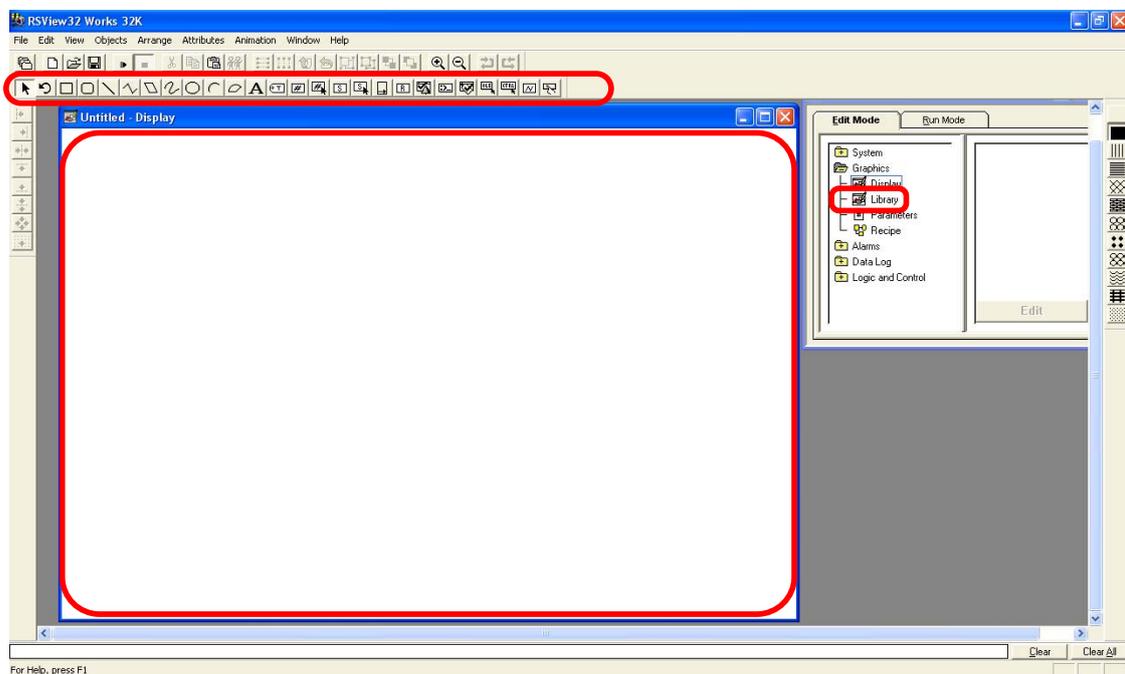
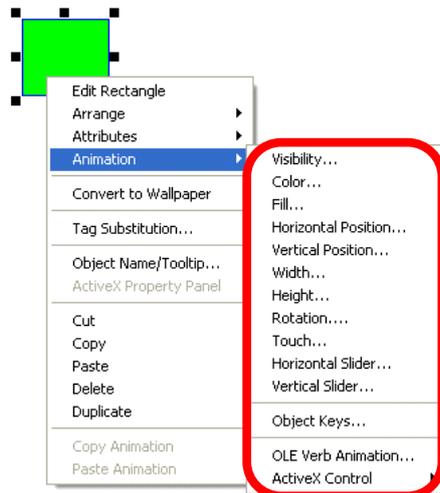


Figura. 3.37. Creación de pantallas gráficas en RSView32 (2)

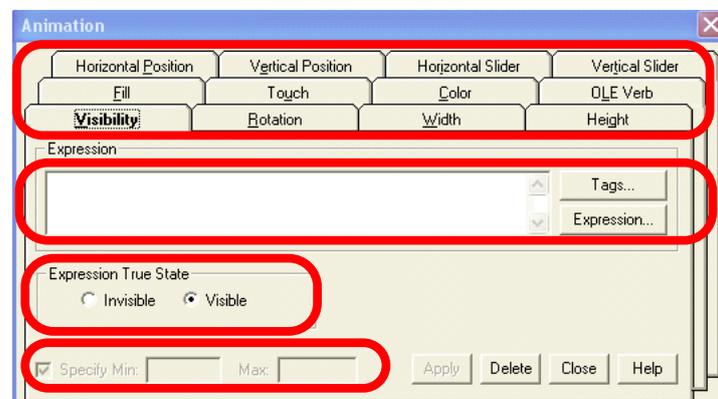
## Animar Objetos

1. Una vez creados los objetos gráficos (botones, medidores, leds, etc.) se puede crear animaciones a través de la ventana de animación o al dar clic derecho sobre el objeto y seleccionar *Animation*.



**Figura. 3.38. Animación de objetos en RSView32 (1)**

2. Como se puede ver en la figura 3.39, la ventana de aplicación consta de 4 secciones principales:
  - Tipos de animación
  - Área de la expresión
  - Resultado de a expresión
  - Valores mínimos y máximos



**Figura. 3.39. Animación de objetos en RSView32 (2)**

3. Para probar la animación creada, se deberá cambiar a Modo **Test**, presionando el botón **Test Run** de la barra de herramientas.

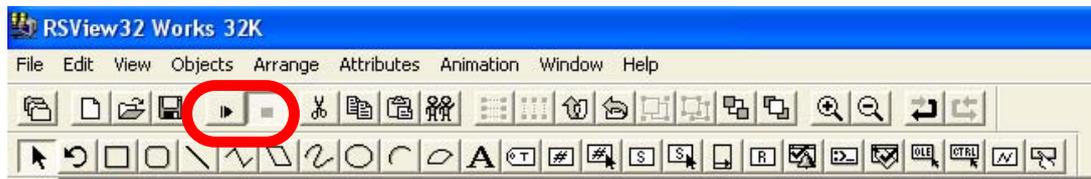


Figura. 3.40. Modo Test Run en RSView32



*El modo TEST, NO es lo mismo que “Correr” el programa puesto que el modo TEST no realiza los cambios de apariencia o posición de la ventana, realizados en el cuadro de dialogo “DISPLAY SETTINGS”.*

### Crear Trends (Gráficos de tendencia)

1. Seleccionar la barra de herramientas Trend, dando clic en el icono respectivo como se ve en la figura 3.41., y seleccionando el área que ocupara esta gráfica de tendencia (trend) en nuestra pantalla.

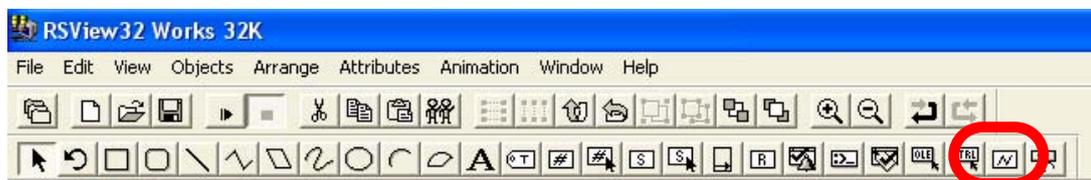


Figura. 3.41. Creación de Trends en RSView32 (1)

2. En la ventana que aparece, presionando la ceja **Trend Configuration**, se encuentran las diferentes opciones de configuración del gráfico.

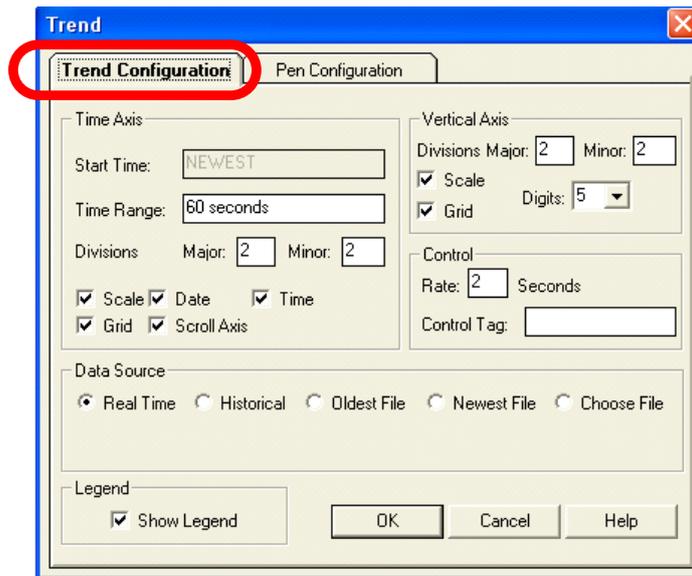


Figura. 3.42. Creación de Trends en RSView32 (2)

3. Presionando la ceja *Pen Configuration*, aparece la ventana en la que se debe asignar un tag a la línea que queremos que se dibuje, pudiendo cambiar también el estilo de la línea.

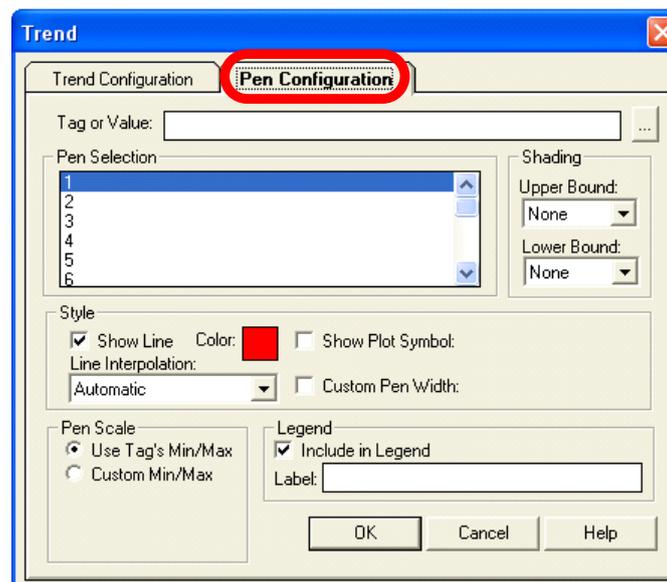


Figura. 3.43. Creación de Trends en RSView32 (3)

## Crear y configurar Eventos

Los eventos son expresiones que inician o ejecutan acciones.

Las expresiones son ecuaciones que contienen valores, Tags, operaciones matemáticas, lógica if–then–else y otras funciones propias de RSVIEW32.

Las acciones son comandos, símbolos y macros de RSVIEW32, una acción puede por ejemplo mostrar una pantalla de error utilizando los comandos de pantalla.

Estos Eventos se utilizan cuando se requiere que se valide una expresión constantemente (mientras se ejecute el evento) y se realice una acción dependiendo del resultado.

1. Para crear un nuevo evento se debe dar clic en “Logic and Control” para desplegar los submenús. Posteriormente se debe dar doble clic en “Events”.

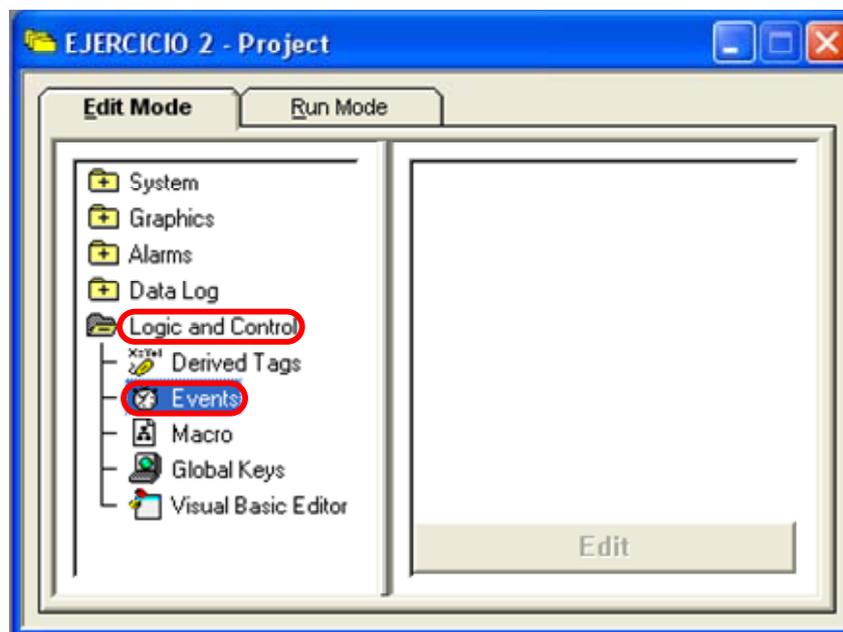


Figura. 3.44. Creación de Eventos en RSVIEW32 (1)

2. Aparece la siguiente pantalla, en la cual en *Action* se debe ingresar la acción se debe realizar el evento, cada vez que la expresión *Expression* sea verdadera. En *Description* se pone una pequeña descripción de la función del evento, si bien esto es opcional es recomendable para facilitar la posterior revisión. En *Expression* se ingresa la expresión que debe ser evaluada, para lo cual existen varias opciones en la parte inferior. Por último, la sección inferior de la ventana nos muestra las diferentes acciones que se han creado en el evento.

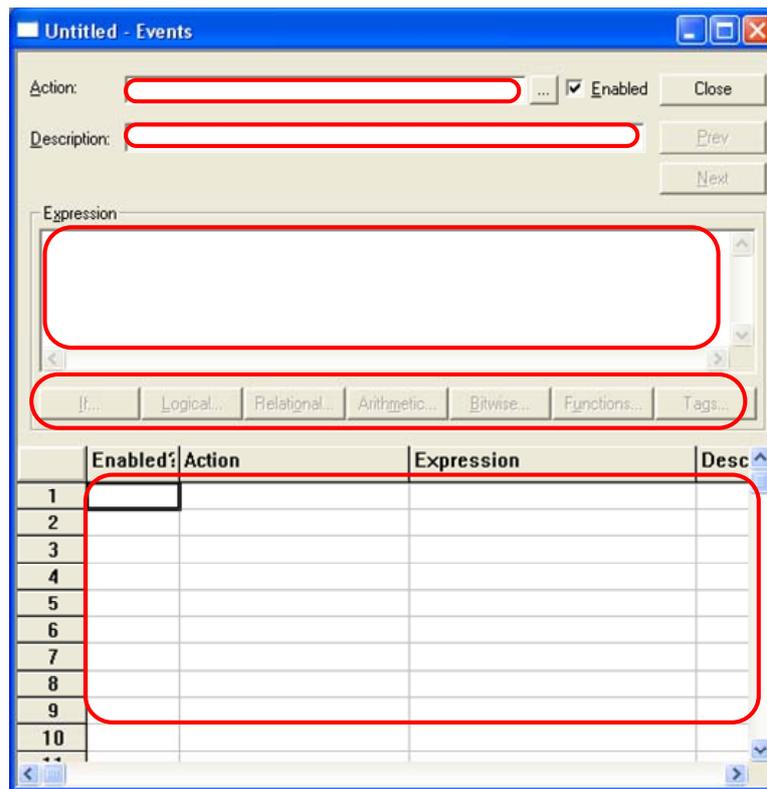


Figura. 3.45. Creación de Eventos en RSView32 (2)

3. Para configurar como se va a evaluar el Evento (Continuo o en Intervalos) se debe dar clic en Setup y posteriormente en Event Setup, lo que despliega la pantalla de configuración.

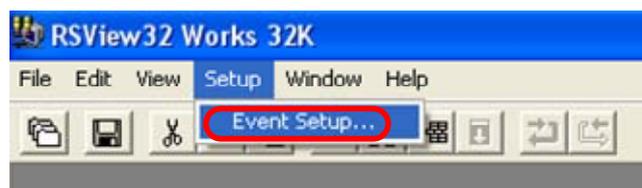


Figura. 3.46. Creación de Eventos en RSView32 (3)

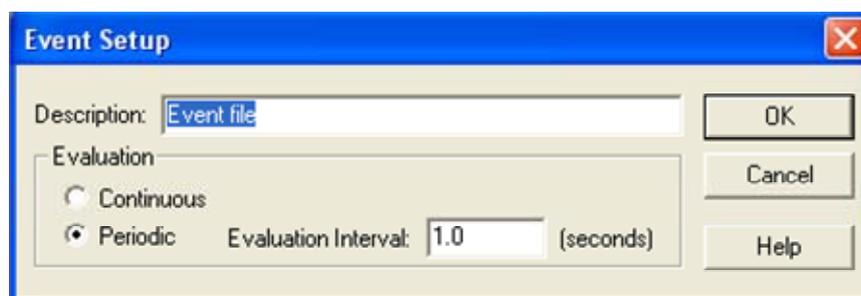


Figura. 3.47. Creación de Eventos en RSView32 (4)

4. Una vez creado el evento, este puede ser activado o desactivado a nuestra conveniencia con los comandos *EventOn* y *EventOff* en cualquier línea de comando. (por ejemplo cuando se presione un botón o cuando aparezca una pantalla).

### Correr el Programa

1. Para configurar los parámetros de cómo queremos que corra nuestro programa, se debe ingresar a la ventana de opciones, dando doble clic en *Startup*.

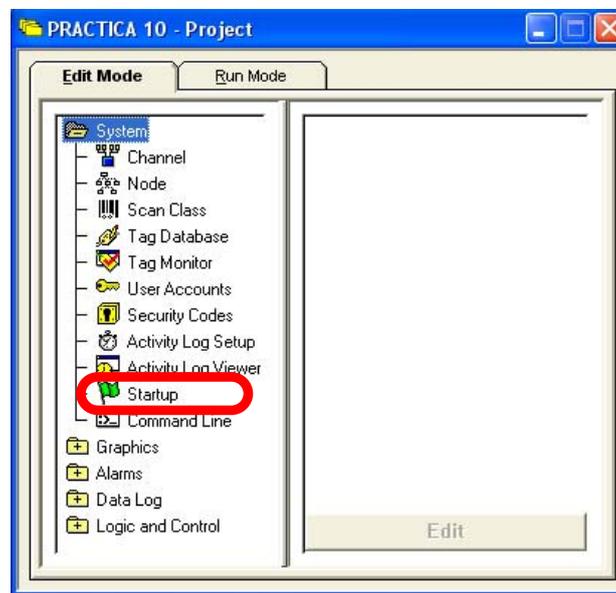


Figura. 3.48. Configurar la corrida del programa en RSView32 (1)

2. Aparece la ventana de *Preferences*, en la que se pueden seleccionar diferentes opciones para que aparezcan o no en la pantalla al momento de correr el programa, por ejemplo los botones de minimizar y maximizar.

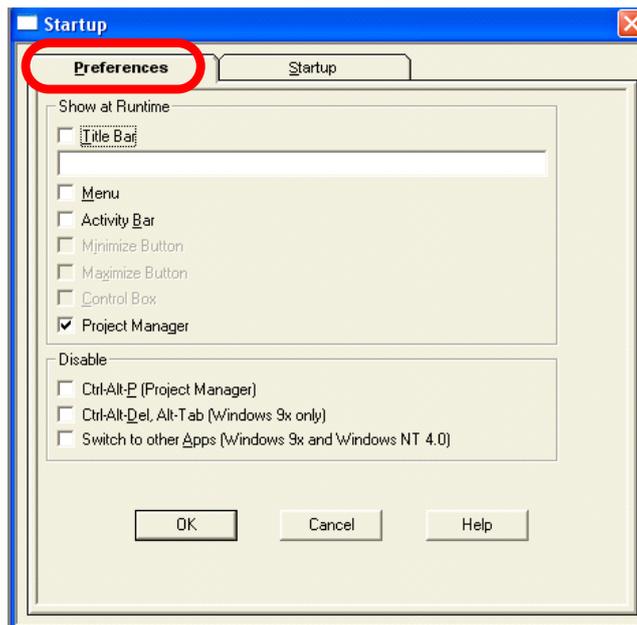


Figura. 3.49. Configurar la corrida del programa en RSView32 (2)

3. Presionando la pestaña de *Startup*, aparecen nuevas opciones como por ejemplo definir cual va a ser nuestra pantalla de inicio.

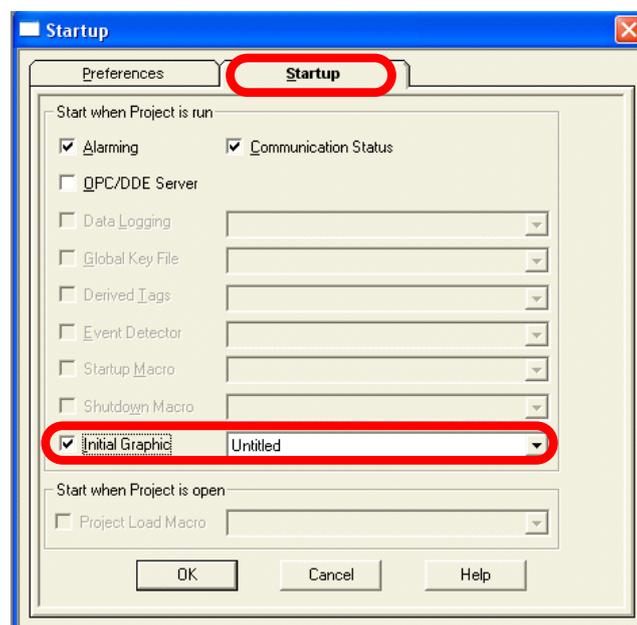


Figura. 3.50. Configurar la corrida del programa en RSView32 (3)

4. Finalmente, para correr el programa presione la pestaña Run Mode, y finalmente Run Project.



Figura. 3.51 Configurar la corrida del programa en RSView32 (4)

## CAPITULO 4

# DISEÑO DE EXPERIMENTOS PARA EL SISTEMA DE ENTRENAMIENTO

### 4.1. CONTROL LÓGICO 1: INSTRUCCIONES DE BITS 1

#### 4.1.1. OBJETIVOS

- Familiarizarse con RSLogix500 (software de programación) y con RSLinx (software de comunicación)
- Utilizar las entradas y salidas discretas del PLC
- Familiarizarse con el direccionamiento lógico para los elementos con imagen de entradas o salidas.
- Programar aplicaciones utilizando instrucciones de bits

#### 4.1.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 ó Micrologix.
- ✓ 1 Módulo de Entradas Discretas\*.
- ✓ 1 Módulo de Salidas Discretas\*.
- ✓ 1 Computador Personal, con instalación de RSLogix500 y RSLinx
- ✓ 1 Tablero Didáctico de Simulación AB-01
- ✓ 1 Fuente de alimentación +24 VDC
- ✓ Cables de Conexión

(\*No se requiere si se utiliza un PLC de la serie Micrologix)

#### 4.1.3. MARCO TEÓRICO

### Configuración del Driver serial de comunicación entre el PLC y la PC

Para configurar correctamente el driver de comunicación, se deben seguir las siguientes instrucciones:

1. Conectar el cable serial del PLC al computador y encender el PLC
2. Ejecutar el software RSLinx y dar clic en *Configure Drivers*

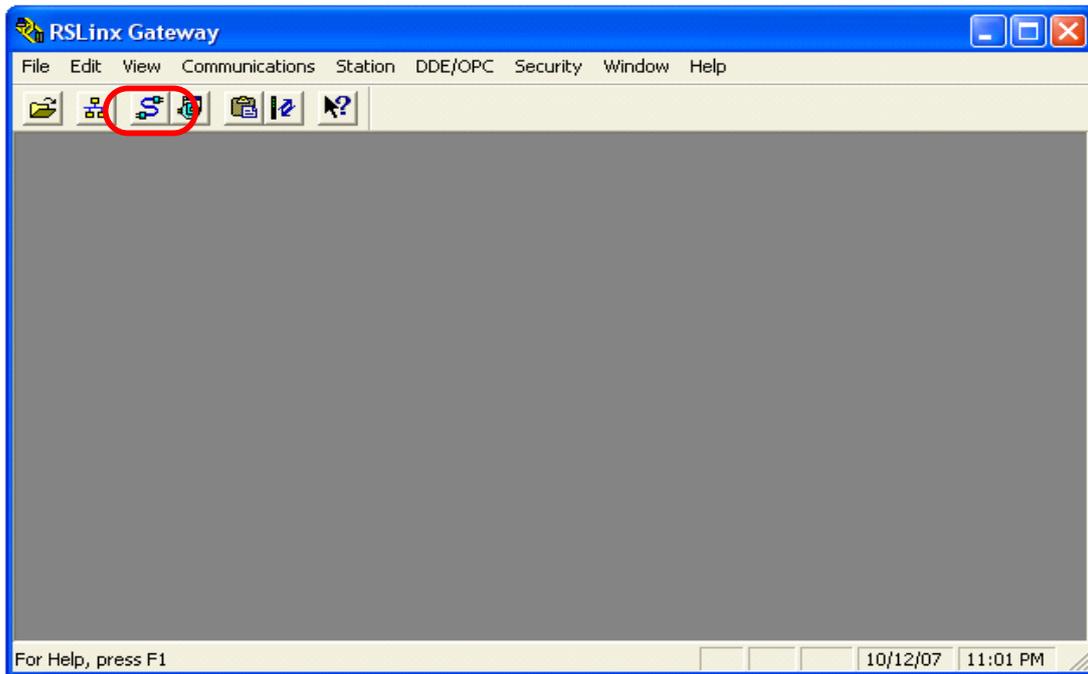


Figura. 4.1. Configuración del Driver de comunicación serial RS232 (1)

3. En la pantalla que aparece, seleccionar el driver de comunicación “*RS-232 DF1 devices*” y hacer clic en *Add New* como se indica en la figura 4.2.

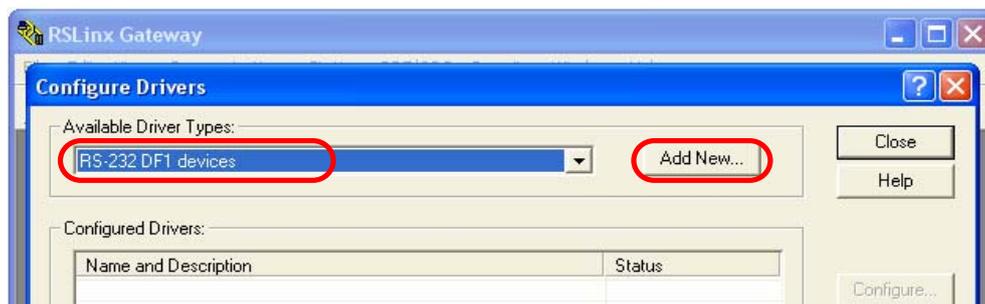


Figura. 4.2. Configuración del Driver de comunicación serial RS232 (2)

4. Escribir un nuevo nombre para el Driver o aceptar el que viene por defecto y presionar *OK*.



Figura. 4.3. Configuración del Driver de comunicación serial RS232 (3)

5. Seleccionar el puerto serial al que se encuentra conectado el PLC, seleccionar el dispositivo *SLC-CHO/Micro/PanelView*, presionar *Auto-Configure*, una vez que aparezca el mensaje de configuración exitosa presionar *OK*

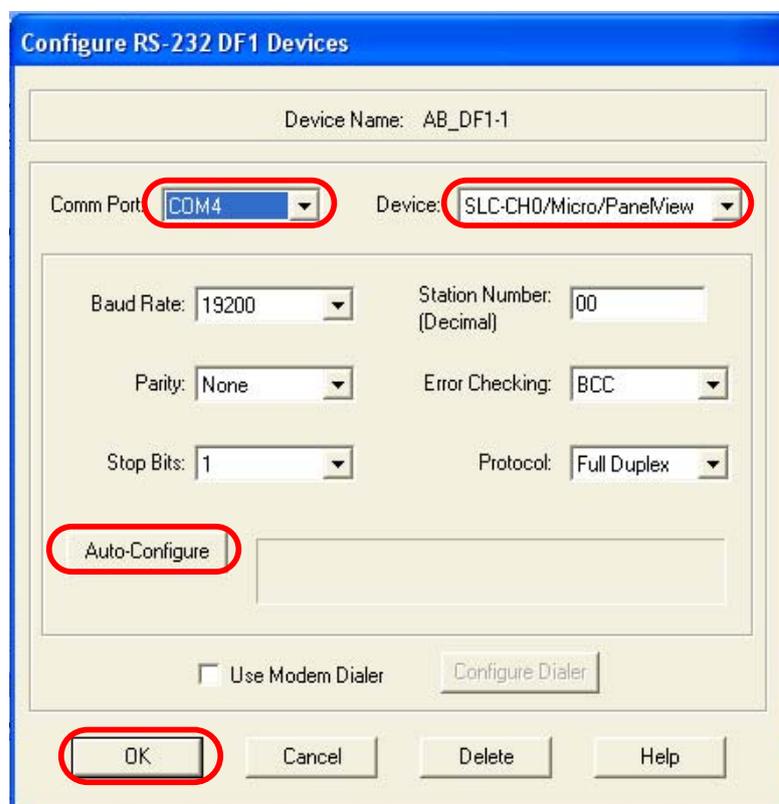


Figura. 4.4. Configuración del Driver de comunicación serial RS232 (4)

**Nota:** Si la auto configuración no es exitosa podrían aparecer los siguientes mensajes:

- a. *Failed to find baud and parity! Check all cables and switch settings!*

Esto puede indicar que el Puerto serial para la computadora no esta habilitado, el cable esta dañado o no esta conectado correctamente, o el protocolo para el canal del procesador no esta configurado para comunicación RS-232 full duplex.

b. *Unable to verify settings due to packet time-out! (or Unable to verify settings due to a NAK!)* Check all cables and configuration and try again.

Estos dos mensajes usualmente indican que el canal en el procesador no esta configurado para comunicación RS-232 full duplex.

c. *Unable to open specified port for configuration testing!*

Hay conflicto en el Puerto serial (está siendo usado por otro driver en RSLinx o por un dispositivo diferente tal como un MODEM).

6. Verificar que el driver esté funcionando correctamente, y presionar **Close** con lo que se volverá a la pantalla de inicio.

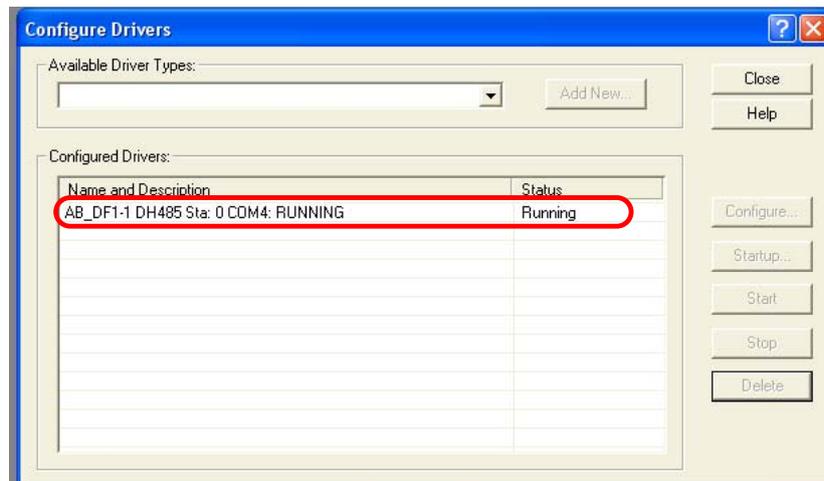


Figura. 4.5. Verificación del driver de comunicación serial RS232

## Introducción al software RSLogix500

Para crear un programa ladder con el software RSLogix500, se deberá seguir los siguientes pasos:

1. Configurar el driver que va a utilizarse en la comunicación con el PLC (Explicado en la sección anterior).
2. Ejecutar el programa RSLogix500 y presionar el icono de *New* como se muestra en la figura 4.6.

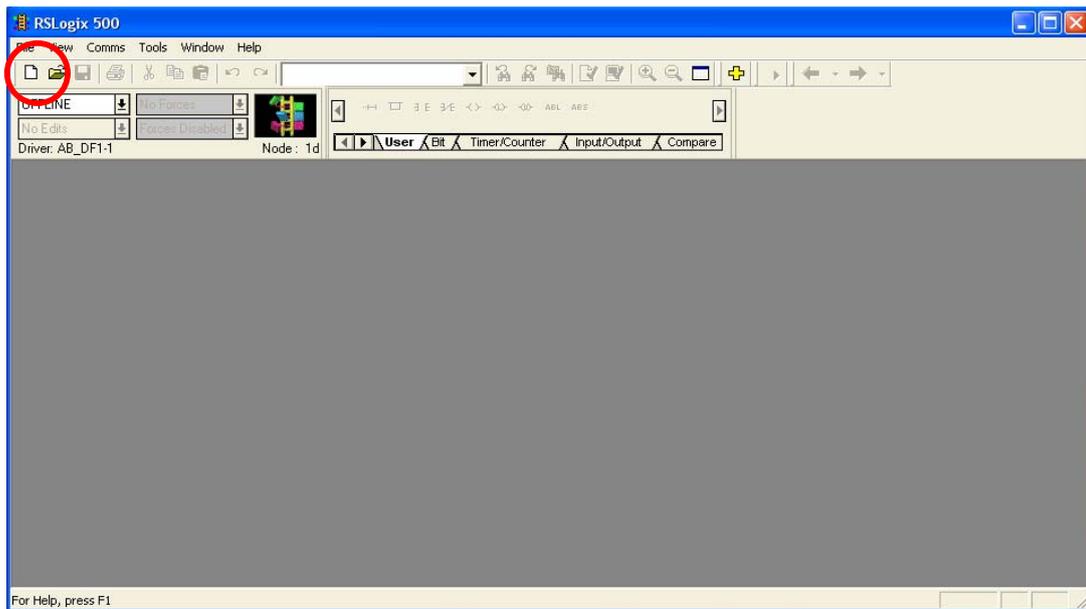


Figura. 4.6. Creación de un nuevo archivo en RSLogix 500

3. Asignar nombre al Procesador, escoger el tipo de PLC que se tiene de entre la lista, seleccionar el Driver (antes configurado) y presionar **OK**

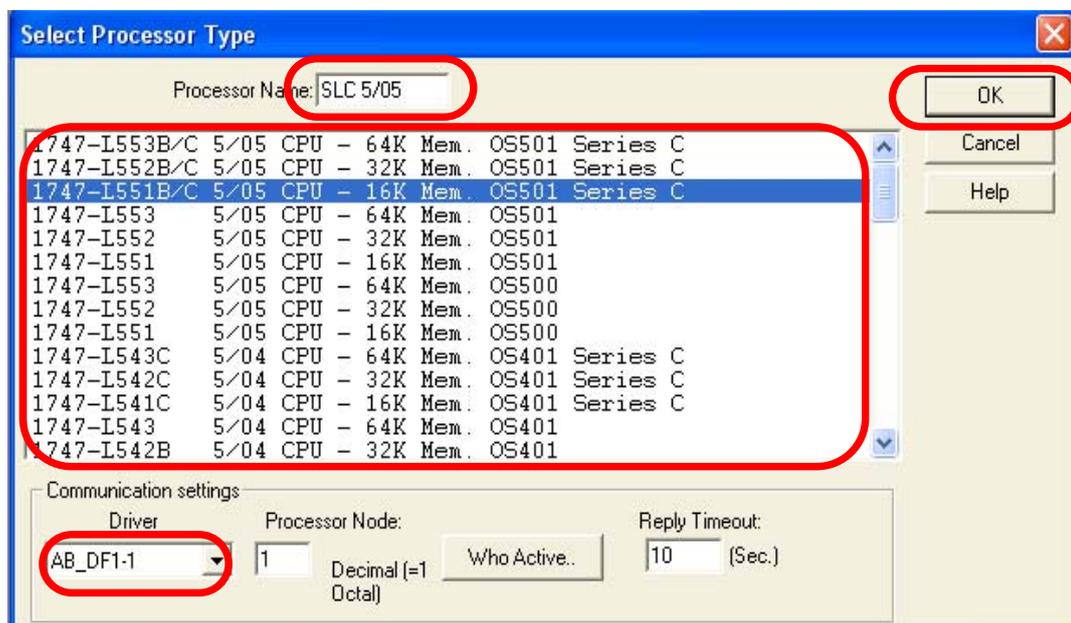


Figura. 4.7. Selección del tipo de PLC en RSLogix 500

4. En la ventana de la izquierda, abrir la carpeta **Controller** y dar clic en **IO Configuration**

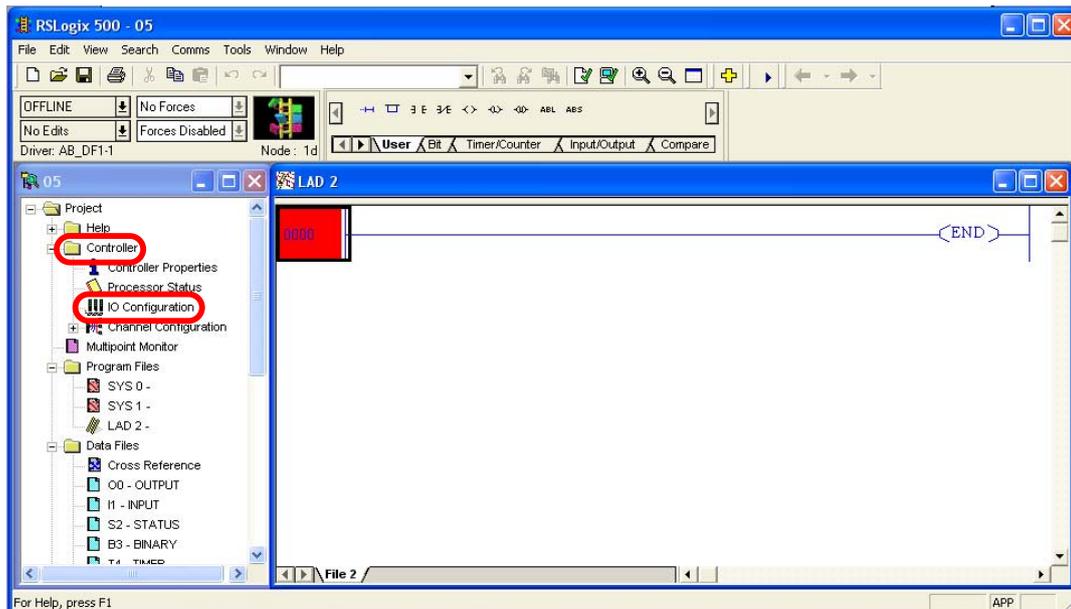


Figura. 4.8. Configuración de los módulos I/O en RSLogix 500 (1)

5. Dar clic en *Read IO Config* para leer automáticamente la configuración de los módulos instalados. Si se quiere hacer de manera manual (si no se dispone al momento el PLC), se debe seleccionar de uno en uno los módulos de entre la lista mostrada.

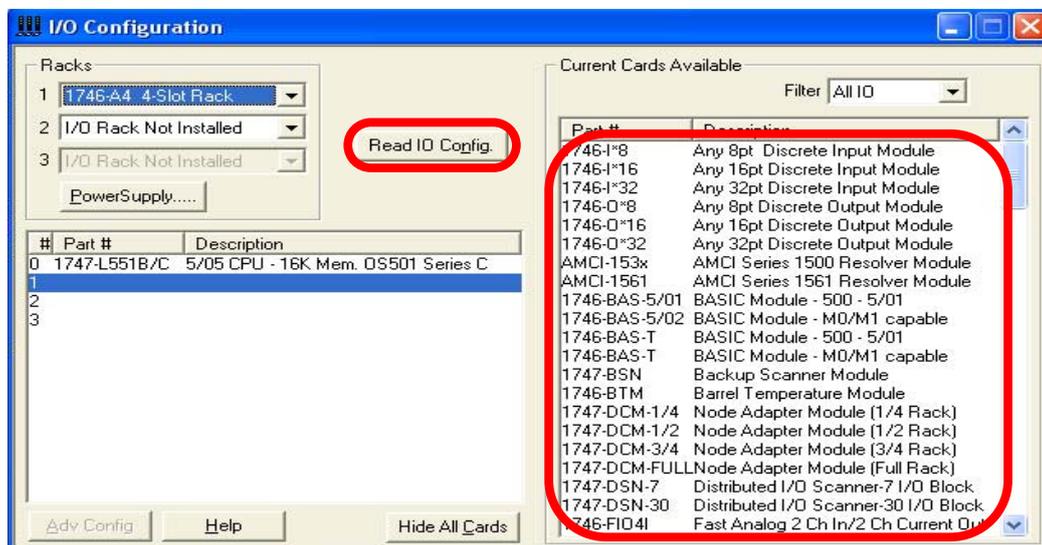


Figura. 4.9. Configuración de los módulos I/O en RSLogix 500 (2)

6. Si se presionó *Read IO Config*, seleccionar el Driver y presionar *Read IO Config* para que el software lea directamente del PLC la ubicación y el tipo de módulos que se encuentran instalados.

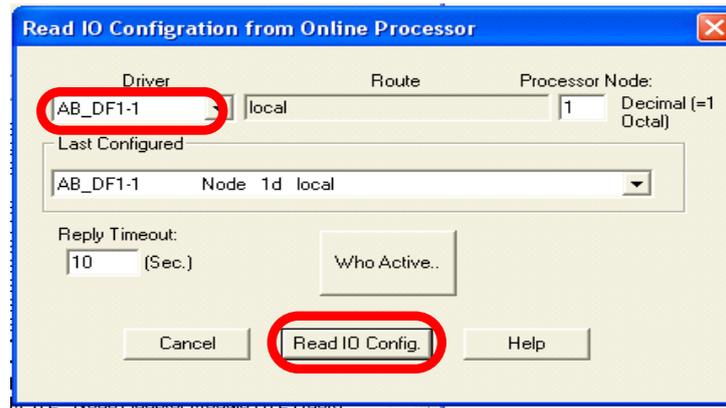


Figura. 4.10. Configuración de los módulos I/O en RSLogix 500 (3)

7. Cerrar la ventana *IO Configuration* y empezar a programar.
8. Una vez terminado el programa, se debe guardarlo en el computador y posteriormente transferirlo al PLC, seleccionando la opción *Download*

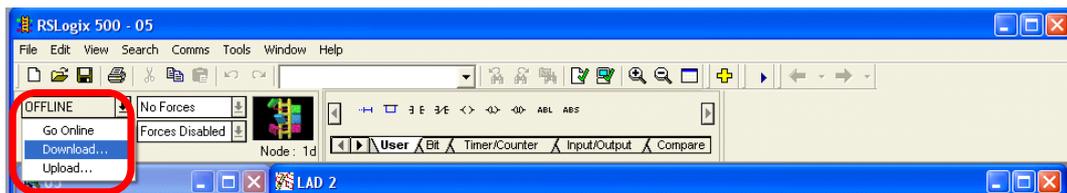


Figura. 4.10. Descargar el programa ladder en el PLC

## Introducción a la Programación

### Formatos de direccionamiento para Entradas y Salidas

La siguiente tabla explica detalladamente el formato de direccionamiento para entradas y salidas:

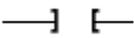
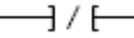
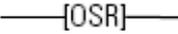
<i>Formato</i>	<i>Explicación</i>	
<b>O:e.s/b</b>	<b>O</b>	Salida (Output)
	<b>I</b>	Entrada (Input)
	<b>:</b>	Delimitador de elemento
<b>I:e.s/b</b>	<b>e</b>	Número de Slot (en el que se encuentra el módulo de I/O) El módulo procesador (CPU) es el Slot 0.
	<b>.</b>	Delimitador de palabra Requerido únicamente si es necesario un número de palabra.

<b>s</b>	Número de palabra	Requerido si el número de entradas o salidas para el slot excede de 16 (módulos especiales).
<b>/</b>	Delimitador de bit	
<b>b</b>	Número de Terminal	Entradas: 0-15 Salidas: 0-15

Tabla. 4.1. Direccionamiento I/O

### Instrucciones de Entrada y Salida

Existen varias instrucciones que permiten manejar las entradas o salidas del PLC. Estas pertenecen al grupo de instrucciones conocidas como *INSTRUCCIONES DE BITS*, puesto que actúan sobre un solo bit. En las siguientes tablas se explican las principales:

<i>Instrucciones de Entrada</i>								
SÍMBOLO	NOMBRE	DESCRIPCIÓN						
	Examine if Closed (XIC)	Esta instrucción cambia su estado de acuerdo al estado del bit que representa, de acuerdo a la siguiente tabla: <table border="1" data-bbox="730 1234 1010 1402"> <thead> <tr> <th><i>Bit</i></th> <th><i>XIC</i></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Falso</td> </tr> <tr> <td>1</td> <td>Verdadero</td> </tr> </tbody> </table>	<i>Bit</i>	<i>XIC</i>	0	Falso	1	Verdadero
<i>Bit</i>	<i>XIC</i>							
0	Falso							
1	Verdadero							
	Examine if Open (XIO)	Esta instrucción cambia su estado de acuerdo al estado del bit que representa, de acuerdo a la siguiente tabla: <table border="1" data-bbox="724 1646 1005 1814"> <thead> <tr> <th><i>Bit</i></th> <th><i>XIO</i></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Verdadero</td> </tr> <tr> <td>1</td> <td>Falso</td> </tr> </tbody> </table>	<i>Bit</i>	<i>XIO</i>	0	Verdadero	1	Falso
<i>Bit</i>	<i>XIO</i>							
0	Verdadero							
1	Falso							
	One-shot	Cuando las condiciones que preceden a esta instrucción cambian de Falso a Verdadero, esta se vuelve verdadera						

	Rising (OSR)	durante un scan, luego se vuelve falsa a pesar que las condiciones que la preceden sigan siendo verdaderas. Es decir solo se vuelve verdadera durante un scan, cuando existe una transición Falso-Verdadero de las condiciones que la preceden.
--	--------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla. 4.2. Instrucciones de Entrada

<i>Instrucciones de Salida</i>		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
—( )—	Output Energize (OTE)	Se utiliza esta instrucción para encender el bit que representa (salida) cuando las condiciones que la preceden son verdaderas.
—(L)—	Output Latch (OTL)	Esta instrucción es similar a la anterior (OTE) con la diferencia que una vez que la condiciones que la preceden cambian de Falso a Verdadero, ésta activa el bit y éste queda activado sin importar que las condiciones que la preceden cambien de estado. Es decir se queda enclavado.
—(U)—	Output Unlatch (OTU)	Esta instrucción es similar a la anterior (OTL), pero trabaja de manera inversa, es decir, una vez que las condiciones que la preceden cambian de Falso a Verdadero, ésta desactiva el bit y éste queda desactivado sin importar que las condiciones que la preceden cambien de estado. Es decir se queda enclavado.

Tabla. 4.3. Instrucciones de Salida



*No se recomienda usar la misma dirección en dos o más instrucciones de salida, a excepción de OTL y OTU que generalmente trabajan en pares.*

### Ejemplos de Programación

*Se desea controlar el encendido y apagado de un motor con dos pulsadores, uno de START y uno de STOP.*

3. El programa Ladder utilizando únicamente como instrucción de salida la instrucción *OTE* se muestra en la figura 4.11

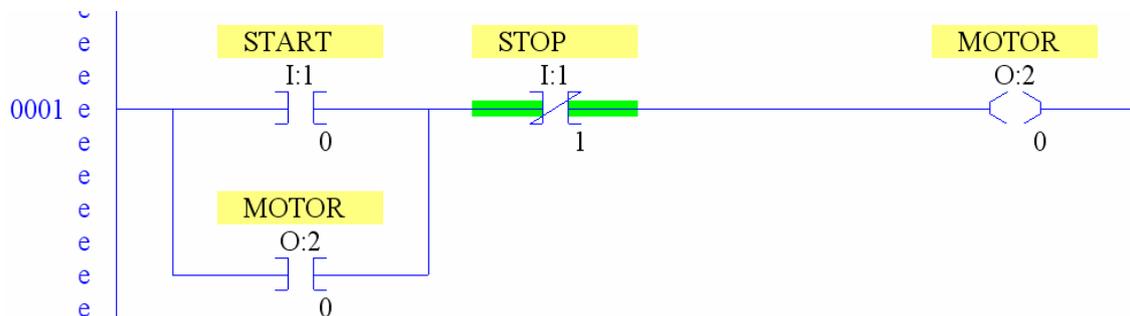


Figura. 4.11. Ejemplo de programación con instrucción OTE

4. El programa Ladder utilizando únicamente como instrucciones de salida las instrucciones *OTL* y *OTU* se muestra en la figura 4.12.

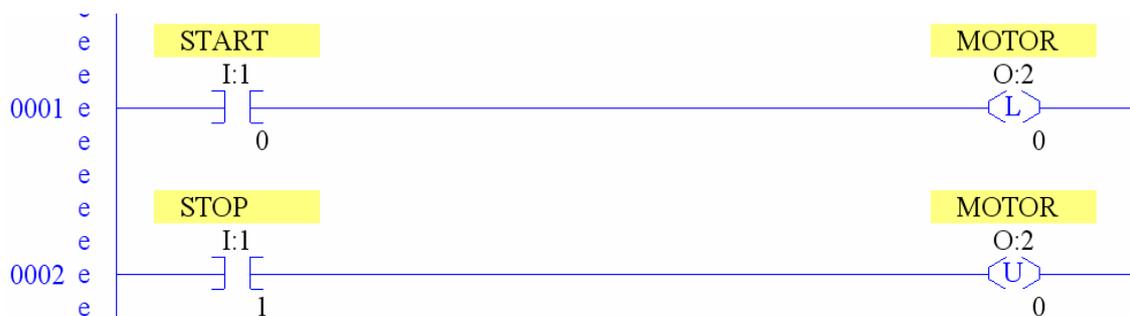


Figura. 4.12. Ejemplo de programación con instrucción OTL Y OTU

Como se puede ver en ambos ejemplos, las direcciones están asignadas de la siguiente manera:

<i>Elemento</i>	<i>Dirección</i>	<i>Descripción</i>
START	I:1/0	Entrada discreta 0, cuyo Módulo se encuentra en el Slot 1
STOP	I:1/1	Entrada discreta 1, cuyo Módulo se encuentra en el Slot 1
MOTOR	O:2/0	Salida discreta 0, cuyo Módulo se encuentra en el Slot 2

Tabla. 4.4. Direcciones del ejemplo de Instrucciones I/O

#### 4.1.4. PLANTEAMIENTO

Se tienen tres motores como se muestra en la figura 4.13, cada uno con un botón de START y otro de STOP. Se desea que la secuencia de encendido sea: Primero el motor

M1, luego el motor M2 y finalmente el motor M3. Mientras que para la secuencia de apagado se desea que el primero en desactivarse sea el motor M3, luego el motor M2 y finalmente el motor M1. No se podrá activar o desactivar los motores, en una secuencia diferente a la antes mencionada.

1. Desarrolle el programa Ladder utilizando únicamente como instrucción de salida la instrucción OTE.
2. Desarrolle el programa Ladder utilizando únicamente como instrucciones de salida las instrucciones OTL y OTU.

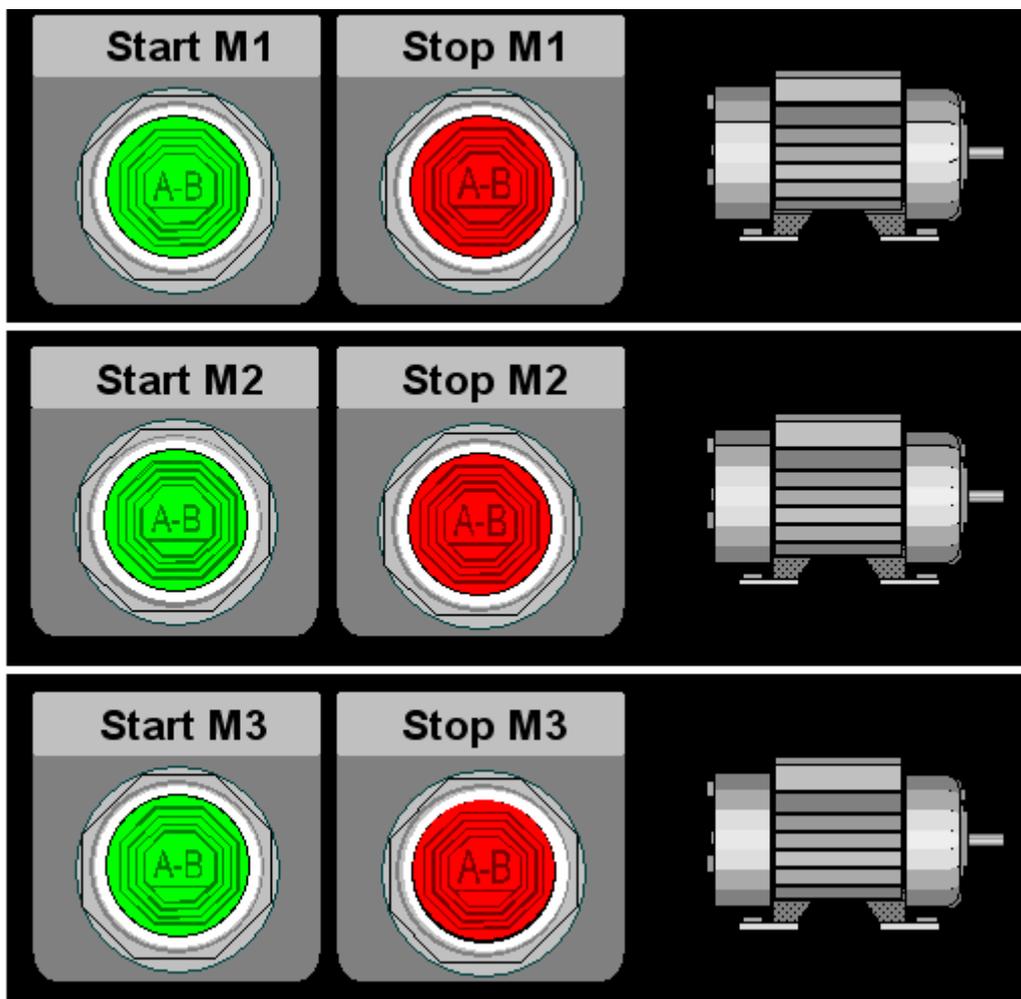


Figura. 4.13. Planteamiento Práctica #1

#### 4.1.5. PROCEDIMIENTO

Para la realización de esta práctica el procedimiento a seguir es el siguiente:

1. Configurar el driver en RSLinx
2. Ejecutar el software RSLogix500, crear un nuevo archivo y configurar el tipo de PLC y sus módulos (de ser el caso) en el programa.
3. Realizar el programa Ladder que cumpla con la lógica de control planteada en esta práctica.
4. Transferir el programa.
5. Ejecutar el programa en el PLC y probar si la lógica del programa desarrollado funciona correctamente en el Modo ONLINE.
6. Conectar las Entradas y Salidas del PLC al Tablero de simulación correspondiente, de acuerdo a la asignación de éstas en el programa.
7. Verificar si el PLC trabaja correctamente de acuerdo al planteamiento de la práctica.

Repetir los pasos anteriores para la segunda parte de la práctica.

#### **4.1.6. PRUEBAS Y RESULTADOS**

#### **4.1.7. CONCLUSIONES Y RECOMENDACIONES**

## 4.2. CONTROL LÓGICO 2: INSTRUCCIONES DE BITS 2

### 4.2.1. OBJETIVOS

- Utilizar las entradas y salidas discretas del PLC
- Programar aplicaciones de control lógico utilizando instrucciones de bits.

### 4.2.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 ó Micrologix.
- ✓ 1 Módulo de Entradas Discretas\*.
- ✓ 1 Módulo de Salidas Discretas\*.
- ✓ 1 Computador Personal, con instalación de RSLogix500 y RSLinx
- ✓ 1 Tablero Didáctico de Simulación AB-02
- ✓ 1 Fuente de alimentación +24 VDC
- ✓ Cables de Conexión

(\*No se requiere si se utiliza un PLC de la serie Micrologix)

### 4.2.3. MARCO TEÓRICO

Referirse al MARCO TEÓRICO de la práctica anterior.

### 4.2.4. PLANTEAMIENTO

Se tiene una tolva que almacena asfalto, como se indica en la figura 4.14. Esta posee dos motores, el motor *M1* o de *Llenado* que alimenta de material a la tolva y el motor *M2* o de *Vaciado* que permite el desalojo del material. También se dispone de tres detectores electromecánicos de nivel tipo boya (con contacto normalmente abierto) *S1*, *S2* y *S3* instalados en la tolva como se indica en la figura 4.14.

Se desea que la tolva pueda operar en dos modos: Manual y Automático. En el Modo Manual, el operador puede activar y desactivar los motores de acuerdo a su conveniencia, mientras que en el Modo Automático, la operación se realizará de acuerdo a las siguientes condiciones:

El motor M1 se enciende cuando el nivel del material está por debajo de S2 y se apaga cuando el material alcanza el nivel de S3.

El motor M2 se enciende cuando el material alcanza el nivel de S2 y se apaga cuando el material está por debajo de S1.

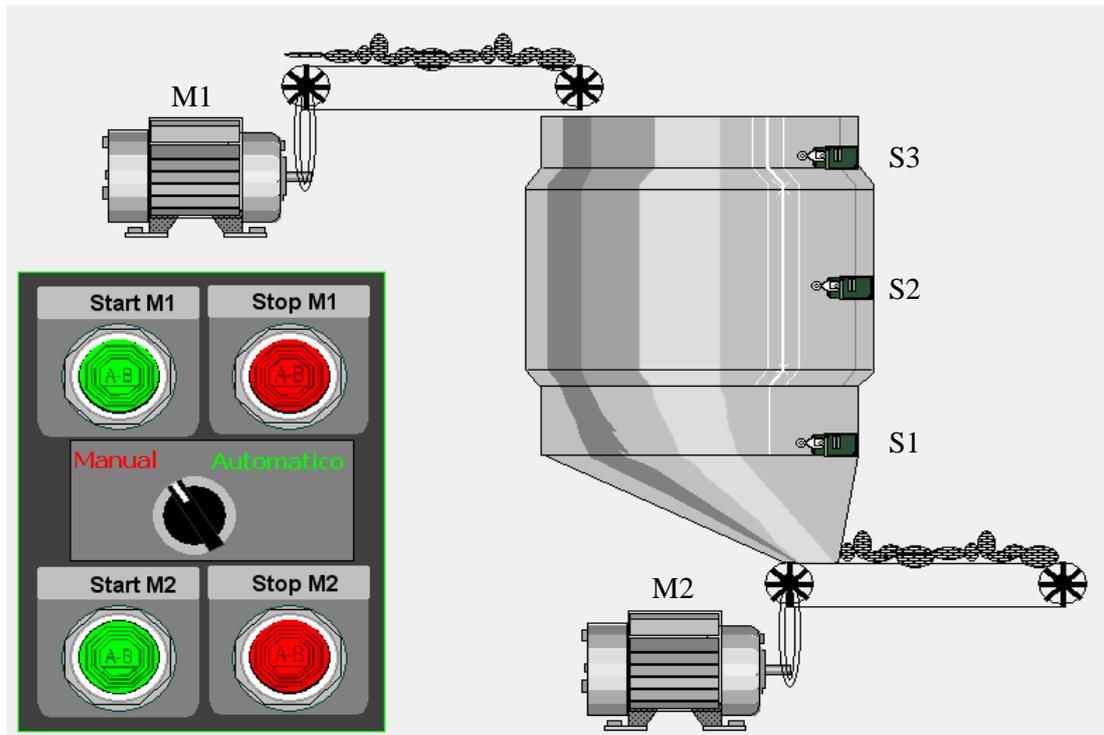


Figura. 4.14. Planteamiento Práctica #2

#### 4.2.5. PROCEDIMIENTO

Para la realización de esta práctica el procedimiento a seguir es el siguiente:

1. Configurar el driver en RSLinx
2. Ejecutar RSLogix500, crear un nuevo archivo y configurar el tipo de PLC y sus módulos (de ser el caso) en el programa
3. Realizar el programa Ladder que cumpla con la lógica de control planteada en esta práctica.
4. Transferir el programa.
5. Ejecutar el programa en el PLC y probar si la lógica del programa desarrollado funciona correctamente en el Modo ONLINE.

6. Conectar las Entradas y Salidas del PLC al Tablero de simulación correspondiente, de acuerdo a la asignación de éstas en el programa.
7. Verificar si el PLC trabaja correctamente de acuerdo al planteamiento de la práctica.

#### **4.2.6. PRUEBAS Y RESULTADOS**

#### **4.2.7. CONCLUSIONES Y RECOMENDACIONES**

### 4.3. CONTROL LÓGICO 3: CONTADORES

#### 4.3.1. OBJETIVOS

- Conocer la operación de las Instrucciones de Conteo.
- Programar aplicaciones de control utilizando Instrucciones de Conteo.

#### 4.3.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 ó Micrologix.
- ✓ 1 Módulo de Entradas Discretas\*.
- ✓ 1 Módulo de Salidas Discretas\*.
- ✓ 1 Computador Personal, con instalación de RSLogix500 y RSLinx
- ✓ 1 Tablero Didáctico de Simulación AB-03
- ✓ 1 Fuente de alimentación +24 VDC
- ✓ Cables de Conexión

(\*No se requiere si se utiliza un PLC de la serie Micrologix)

#### 4.3.3. MARCO TEÓRICO

##### Formato de direccionamiento para Contadores

Cada contador está formado por un archivo de datos de 3 palabras (16 bits cada palabra). La palabra 0 es una palabra de control que contiene los bits de Status del contador. La palabra 1 contiene el valor de PRESET. La palabra 2 contiene el valor acumulado ACCUM.

La siguiente tabla explica el formato de direccionamiento empleado en las instrucciones de conteo:

<i>Formato</i>	<i>Explicación</i>	
<b>Cf:e.s/b</b>	<b>C</b>	Contador (Counter)
	<b>f</b>	Número de archivo. Es 5 por defecto. En los SLC puede asignarse un número entre 9 y 255 para almacenamiento adicional.

	:	Delimitador de elemento	
	e	Número de elemento	Estos son elementos de tres palabras cada uno, que conforman el contador. El rango es entre 0-255.
	.	Delimitador de palabra	Requerido únicamente si es necesario un número de palabra.
	s	Número de palabra	De 0 a 2 (Tres palabras)
	/	Delimitador de bit	
	b	Número de Bit	De 0 a 15 (16 Bits)

**Tabla. 4.5. Direccionamiento COUNTER**

<b>Ejemplos:</b>	
C5:0/15 ó C5:0/CU	Count Up Enable
C5:0/14 ó C5:0/CD	Count Down Enable
C5:0/13 ó C5:0/DN	Done (DN) bit
C5:0/12 ó C5:0/OV	Overflow bit
C5:0/11 ó C5:0/UN	Underflow bit
C5:0.1 ó C5:0.PRE	Valor PRESET del contador
C5:0.2 ó C5:0.ACC	Valor acumulado del contador
C5:0.1/0 ó C5:0.PRE/0	Bit 0 del valor PRESET
C5:0.2/0 ó C5:0.ACC/0	Bit 0 del valor acumulado

**Tabla. 4.6. Ejemplos Direccionamiento COUNTER**

### Instrucciones de Conteo

Existen dos instrucciones principales de conteo, CTU o de conteo ascendente y CTD o de conteo descendente. Ambas poseen los siguientes parámetros:

1. *Dirección del Contador:* Esta debe ser ingresada por el programador, de acuerdo al formato antes explicado

2. *Valor de PRESET*: Este debe ser ingresado por el programador y representa el valor que el acumulador debe alcanzar para activar el bit Done (DN).
3. *El valor acumulado ACCUM*: este representa el valor actual de la cuenta.

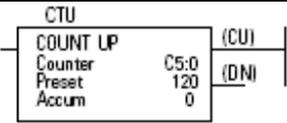
<i>Instrucción de Conteo Ascendente</i>		
<b>SÍMBOLO</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>
	Count Up (CTU)	Esta instrucción cuenta de manera creciente las transiciones Falso-Verdadero de las condiciones que la preceden. Pudiendo estas transiciones ser causadas por condiciones de la lógica del programa o por dispositivos de campo externos.

Tabla. 4.7. Instrucción CTU

<i>Bits de STATUS de la instrucción CTU</i>		
<i>Este BIT</i>	<i>Es seteado cuando...</i>	<i>Permanece seteado hasta que...</i>
CTU <i>Overflow</i> (OV) Bit 12	El valor acumulado excede el limite y pasa de +32768 a -32767 y continua contando desde ahí	Sea ejecutada una instrucción RES con la misma dirección de la instrucción CTU o hasta que la cuenta sea disminuida con una instrucción CTD y el valor acumulado sea menor o igual a +32768.
<i>Done (DN)</i> Bit 13	El valor acumulado es igual o mayor al valor PRESET	El valor acumulado sea menor al valor de PRESET
CTU <i>Enable</i> (CU) Bit 15	Las condiciones que preceden la instrucción CTU son verdaderas	Las condiciones que preceden la instrucción CTU son falsas o esté activada una instrucción RES con la misma dirección de la instrucción CTU

Tabla. 4.8. Bits de Status de la Instrucción CTU

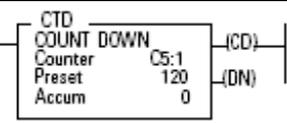
<b>Instrucción de Conteo Descendente</b>		
<b>SÍMBOLO</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>
	Count Down (CTD)	Esta instrucción cuenta de manera decreciente las transiciones Falso-Verdadero de las condiciones que la preceden. Pudiendo estas transiciones ser causadas por condiciones de la lógica del programa o por dispositivos de campo externos.

Tabla. 4.9. Instrucción CTD

<b>Bits de STATUS de la instrucción CTU</b>		
<b>Este BIT</b>	<b>Es seteado cuando...</b>	<b>Permanece seteado hasta que...</b>
CTD <i>Underflow</i> (UN) Bit 11	El valor acumulado excede el limite y pasa de -32767 a +32768 y continua contando desde ahí	Sea ejecutada una instrucción RES con la misma dirección de la instrucción CTD o hasta que la cuenta sea aumentada con una instrucción CTD y el valor acumulado sea mayor o igual a -32767.
<i>Done</i> (DN) Bit 13	El valor acumulado es igual o mayor al valor PRESET	El valor acumulado sea menor al valor de PRESET
CTD <i>Enable</i> (CU) Bit 14	Las condiciones que preceden la instrucción CTU son verdaderas	Las condiciones que preceden la instrucción CTU son falsas o esté activada una instrucción RES con la misma dirección de la instrucción CTU

Tabla. 4.10. Bits de Status de la Instrucción CTD



*La duración de la transición On-Off de la señal de entrada no debe ser menor a dos veces el tiempo de scan. (Asumiendo un ciclo de trabajo del 50%)*

<b>Instrucción de Reset</b>		
<b>SÍMBOLO</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>
	Reset (RES)	Esta instrucción se utiliza para resetear un contador o un temporizador, Cuando las función RES está habilitada esta resetea los Count Up (CTU), Count Down (CTD), Timer On

		Delay (TON) ó Retentive Timer (RTO) que tengan la misma dirección de la función RES.
--	--	--------------------------------------------------------------------------------------

**Tabla. 4.11. Instrucción RES**

La instrucción RES resetea todos los bits de Status de los contadores que se vieron anteriormente, y vuelve a cero los valores acumulados tanto de los CTU y CTD.



*Debido a que la instrucción RES vuelve a cero los valores acumulados, si el valor Preset del contador es negativo, esto causa que el bit Done (DN) se active, lo cual deberá ser considerado por el programador*

#### 4.3.4. PLANTEAMIENTO

En una banda transportadora de botellas de refrescos, como la que se ilustra en la figura 4.15, se dispone de dos sistemas detectores: uno de control de peso S1, que entregará una señal de activación ON cuando el peso del producto este fuera de una tolerancia establecida y otro de presencia del producto aprobado, a través del detector capacitivo S2. Desarrollar un programa para que, a través del PLC se realice la siguiente rutina de trabajo:

- Cada 12 productos aprobados, se activará un sistema elevador de cajas para el embalaje.
- Se contabilizará el número total de productos aprobados y embalados.
- Se contabilizará el número total de cajas embaladas
- Si el número de productos defectuosos es mayor a 20, se encenderá un foco de alarma, por lo que el sistema deberá tener un botón de reconocimiento de alarma que además encere el contador de productos defectuosos
- Es sistema cuenta con un pulsador para resetear todos los contadores.

El sistema contará con las siguientes señales de entrada y salida:

Nombre	Descripción
RECHAZAR	Salida que activa el brazo que rechaza el producto
ALARMA	Salida que activa la alarma
ELEVADOR	Salida que activa el elevador de cajas
CARGAR EN CAJA	Salida que activa el brazo que coloca el producto aprobado en

	la caja
S1	Sensor de control de peso, que se activa cuando el producto se debe rechazar.
RECHAZADO	Señal de entrada que indica que el producto ya fue rechazado.
AKW ALARMA	Pulsador de reconocimiento de Alarma
S2	Sensor de presencia del producto aprobado, listo para ser cargado.
CARGADO	Señal de entrada que indica que el producto ya fue cargado.
FIN ELEVADOR	Señal de entrada que indica que el elevador ya regresó a su lugar, después de haber movido la caja.
ENCERAR CUENTAS	Pulsador para encerar los contadores.

Tabla. 4.12. Señales I/O Práctica #3

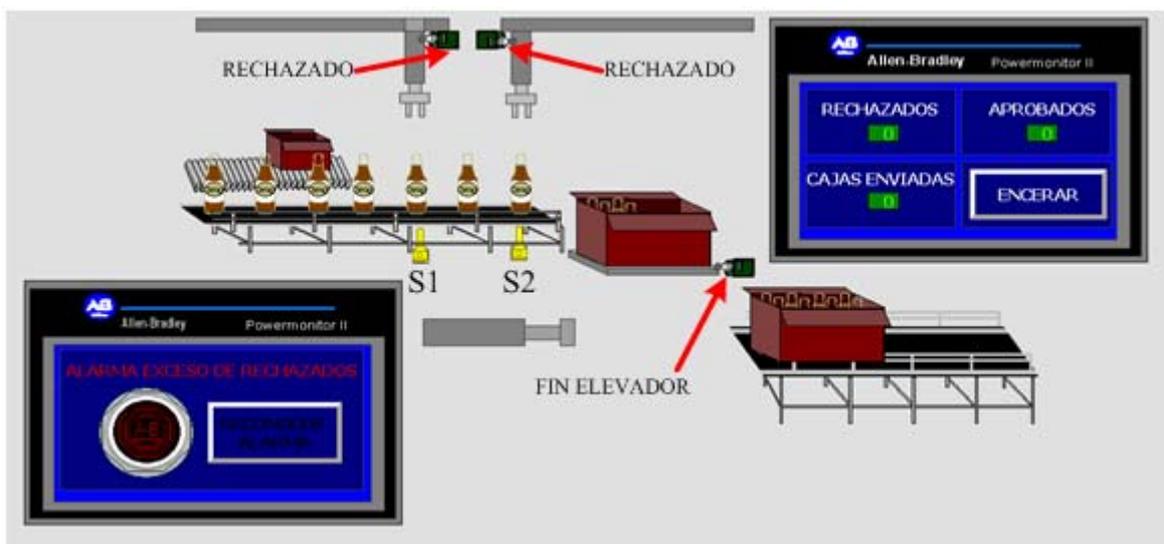


Figura. 4.15. Planteamiento Práctica #3

#### 4.3.5. PROCEDIMIENTO

Para la realización de esta práctica el procedimiento a seguir es el siguiente:

1. Configurar el driver en RSLinx
2. Ejecutar RSLogix500, crear un nuevo archivo y configurar el tipo de PLC y sus módulos (de ser el caso) en el programa
3. Realizar el programa Ladder que cumpla con la lógica de control planteada en esta práctica.

4. Transferir el programa.
5. Ejecutar el programa en el PLC y probar si la lógica del programa desarrollado funciona correctamente en el Modo ONLINE.
6. Conectar las Entradas y Salidas del PLC al Tablero de simulación correspondiente, de acuerdo a la asignación de éstas en el programa.
7. Verificar si el PLC trabaja correctamente de acuerdo al planteamiento de la práctica.

#### **4.3.6. PRUEBAS Y RESULTADOS**

#### **4.3.7. CONCLUSIONES Y RECOMENDACIONES**

## 4.4. CONTROL LÓGICO 4: TEMPORIZADORES 1

### 4.4.1. OBJETIVOS

- Conocer la operación de las Instrucciones de Temporización.
- Programar aplicaciones de control utilizando Instrucciones de Temporización.

### 4.4.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 ó Micrologix.
- ✓ 1 Módulo de Entradas Discretas\*.
- ✓ 1 Módulo de Salidas Discretas\*.
- ✓ 1 Computador Personal, con instalación de RSLogix500 y RSLinx
- ✓ 1 Tablero Didáctico de Simulación AB-04
- ✓ 1 Fuente de alimentación +24 VDC
- ✓ Cables de Conexión

(\*No se requiere si se utiliza un PLC de la serie Micrologix)

### 4.4.3. MARCO TEÓRICO

#### Formato de direccionamiento para Temporizadores

Cada Temporizador esta formado por un archivo de datos de 3 palabras (16 bits cada palabra). La palabra 0 es una palabra de control que contiene los bits de Status del Temporizador. La palabra 1 contiene el valor de PRESET. La palabra 2 contiene el valor acumulado ACCUM.

La siguiente tabla explica detalladamente el formato de direccionamiento empleado en las instrucciones de temporización:

<i>Formato</i>	<i>Explicación</i>	
<b>Tf:e.s/b</b>	<b>T</b>	Temporizador (Timer)
	<b>f</b>	Número de archivo. Es 4 por defecto. En los SLC puede asignarse un número entre 9 y 255 para almacenamiento adicional.

	:	Delimitador de elemento	
	e	Número de elemento	Estos son elementos de tres palabras cada uno, que conforman el temporizador. El rango es entre 0-255.
	.	Delimitador de palabra	Requerido únicamente si es necesario un número de palabra.
	s	Número de palabra	De 0 a 2 (Tres palabras)
	/	Delimitador de bit	
	b	Número de Bit	De 0 a 15 (16 Bits)

**Tabla. 4.13. Direccionamiento TIMER**

<b>Ejemplos:</b>	
T4:0/15 or T4:0/EN	Enable bit
T4:0/14 or T4:0/TT	Timer timing bit
T4:0/13 or T4:0/DN	Done bit
T4:0.1 or T4:0.PRE	Preset value of the timer
T4:0.2 or T4:0.ACC	Valor Acumulado del temporizador
T4:0.1/0 or T4:0.PRE/0	Bit 0 del valor PRESET
T4:0.2/0 or T4:0.ACC/0	Bit 0 del valor Acumulado

**Tabla. 4.14. Ejemplos Direccionamiento TIMER**

### Instrucciones de Temporización

Existen dos instrucciones principales de temporización, TON o temporizador retardador de encendido y TOF o temporizador retardador de apagado. Ambas poseen los siguientes parámetros:

1. *Dirección del Temporizador:* Esta debe ser ingresada por el programador, de acuerdo al formato antes explicado.
2. *Valor de PRESET:* Este debe ser ingresado por el programador y representa el valor que el acumulado debe alcanzar para activar o desactivar el bit Done (DN) dependiendo si es una instrucción TON ó TOF.
3. *El valor acumulado ACCUM:* este representa el valor actual de la cuenta.

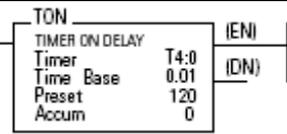
<b>Temporizador retardador de encendido</b>		
<b>SÍMBOLO</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>
	Timer On-delay (TON)	Use esta instrucción para encender o apagar una salida luego que el temporizador ha estado encendido por un intervalo de tiempo definido (Preset). Esta instrucción cuenta intervalos de tiempo cuando las condiciones que la preceden son verdaderas y el valor acumulado es menor al Preset. Si las condiciones que la preceden se vuelven falsas, el temporizador se resetea.

Tabla. 4.15. Instrucción TON

<b>Bits de STATUS de la instrucción TON</b>		
<b>Este BIT</b>	<b>Es seteado cuando...</b>	<b>Permanece seteado hasta que...</b>
Timer Done DN Bit 13	El valor acumulado es igual o mayor que el valor Preset	Las condiciones que preceden a la instrucción TON son falsas
Timer Timing (TT) Bit 14	Las condiciones que preceden a la instrucción TON son verdaderas y el valor acumulado es menor que el Preset	Las condiciones que preceden a la instrucción TON son falsas o cuando el bit Done DN es seteado
Timer Enable EN Bit 15	Las condiciones que preceden la instrucción TON son verdaderas	Las condiciones que preceden la instrucción TOF son falsas

Tabla. 4.16. Bits de Status de la instrucción TON

#### 4.4.4. PLANTEAMIENTO

Programar e implementar un sistema de alarma contra incendios para una fábrica como el que se muestra en la figura 4.16. , el cual tiene un panel de control que dispone de tres

lámparas indicadoras L1, L2 y L3; además posee cuatro botones: uno de RESET y tres de ENTERADO (uno para cada área). Se cuenta con tres sensores/detectores de llama S1, S2 y S3, uno en cada área principal de la fábrica. En el instante en que cualquiera de los sensores se activa, la lámpara indicadora correspondiente se activa intermitentemente a una frecuencia de 1 Hz y suena una sirena; al presionar el botón de ENTERADO dicha luz dejara de titilar para quedar encendida permanentemente. Ninguna de las luces ni la sirena podrán ser apagadas con el botón de RESET si el PLC sigue recibiendo la señal de presencia de llama o si no se ha presionado previamente el botón de ENTERADO. La operación indebida del botón de RESET provocará que el sistema regrese al estado anterior.

Nota: para que funcionen intermitentemente las luces utilice únicamente 2 temporizadores. Considere también que ante incendios simultáneos en dos o más áreas las lámparas deben titilar sincrónicamente.



Figura. 4.16. Planteamiento Práctica #4

#### 4.4.5. PROCEDIMIENTO

Para la realización de esta práctica el procedimiento a seguir es el siguiente:

1. Configurar el driver en RSLinx
2. Ejecutar RSLogix500, crear un nuevo archivo y configurar el tipo de PLC y sus módulos (de ser el caso) en el programa

3. Realizar el programa Ladder que cumpla con la lógica de control planteada en esta práctica.
4. Transferir el programa.
5. Ejecutar el programa en el PLC y probar si la lógica del programa desarrollado funciona correctamente en el Modo ONLINE.
6. Conectar las Entradas y Salidas del PLC al Tablero de simulación correspondiente, de acuerdo a la asignación de éstas en el programa.
7. Verificar si el PLC trabaja correctamente de acuerdo al planteamiento de la práctica.

#### **4.4.6. PRUEBAS Y RESULTADOS**

#### **4.4.7. CONCLUSIONES Y RECOMENDACIONES**

## 4.5. CONTROL LÓGICO 5: TEMPORIZADORES 2

### 4.5.1. OBJETIVOS

- Familiarizarse con el direccionamiento lógico para las instrucciones de Temporización.
- Conocer el funcionamiento y aplicación de las Instrucciones de Temporización.
- Familiarizarse con el uso de los Bits de Status de las Instrucciones de Temporización.

### 4.5.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 ó Micrologix.
- ✓ 1 Módulo de Entradas Discretas\*.
- ✓ 1 Módulo de Salidas Discretas\*.
- ✓ 1 Computador Personal, con instalación de RSLogix500 y RSLinx
- ✓ 1 Tablero Didáctico de Simulación AB-05
- ✓ 1 Fuente de alimentación +24 VDC
- ✓ Cables de Conexión

(\*No se requiere si se utiliza un PLC de la serie Micrologix

### 4.5.3. MARCO TEÓRICO

Referirse al MARCO TEÓRICO de la práctica anterior.

### 4.5.4. PLANTEAMIENTO

Se desea controlar la apertura de una puerta corrediza de garaje como la que se muestra en la figura 4.17. En los extremos de la puerta se han ubicado dos interruptores de fin de carrera S2 y S3 que se activan cuando la puerta se abre o se cierra totalmente, respectivamente. La puerta se abre cuando el controlador recibe la señal enviada por un detector infrarrojo S1. Una vez que la puerta se abre totalmente, ésta permanece en ese estado por 5 segundos. Adicionalmente, el sistema cuenta con dos luces que indican si la puerta se está abriendo o cerrando, respectivamente.

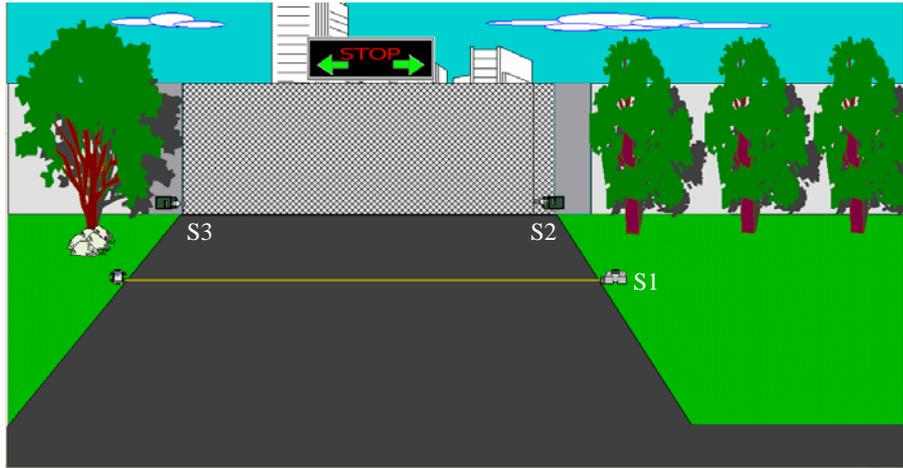


Figura. 4.17. Planteamiento Práctica #5

#### 4.5.5. PROCEDIMIENTO

Para la realización de esta práctica el procedimiento a seguir es el siguiente:

1. Configurar el driver en RSLinx
2. Ejecutar RSLogix500, crear un nuevo archivo y configurar el tipo de PLC y sus módulos (de ser el caso) en el programa
3. Realizar el programa Ladder que cumpla con la lógica de control planteada en esta práctica.
4. Transferir el programa.
5. Ejecutar el programa en el PLC y probar si la lógica del programa desarrollado funciona correctamente en el Modo ONLINE.
6. Conectar las Entradas y Salidas del PLC al Tablero de simulación correspondiente, de acuerdo a la asignación de éstas en el programa.
7. Verificar si el PLC trabaja correctamente de acuerdo al planteamiento de la práctica.

#### 4.5.6. PRUEBAS Y RESULTADOS

#### 4.5.7. CONCLUSIONES Y RECOMENDACIONES

## 4.6. CONTROL LÓGICO 6: ENTRADAS Y SALIDAS ANALÓGICAS

### 4.6.1. OBJETIVOS

- Configurar y utilizar las entradas y salidas analógicas.
- Conocer el formato de direccionamiento de datos de tipo Entero.
- Familiarizarse con las instrucciones de Comparación.
- Familiarizarse con las instrucciones de Transferencia de datos
- Familiarizarse con las instrucciones Matemáticas.

### 4.6.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 ó Micrologix 1000 Analog.
- ✓ 1 Módulo de Entradas Discretas\*.
- ✓ 1 Módulo de Salidas Discretas\*.
- ✓ 1 Módulo de Entradas Analógicas\*.
- ✓ 1 Módulo de Salidas Analógicas\*.
- ✓ 1 Computador Personal, con instalación de RSLogix500 y RSLinx
- ✓ 1 Tablero Didáctico de Simulación AB-06
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

(\*No se requiere si se utiliza un PLC de la serie Micrologix 1000 Analog)

### 4.6.3. MARCO TEÓRICO

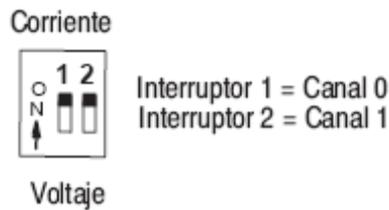
#### Módulos Analógicos Combinados NIO4V

El módulo 1746-NIO4V incluye 2 entradas de Voltaje o Corriente (Seleccionado por el usuario) y 2 salidas de Voltaje.

#### Configuración del Módulo 1746-NIO4V

Los módulos NIO4V tienen dos interruptores individuales marcados 1 y 2 como se muestra en la figura 4.18. Estos interruptores controlan el modo de entrada de los canales de

entrada 0 y 1. Un interruptor en la posición ON configura el canal para entrada de corriente. Un interruptor en la posición OFF configura el canal para entrada de voltaje.



**Figura. 4.18. Selector entradas de voltaje o corriente**



*Se debe tener cuidado para evitar conectar una fuente de voltaje a un canal configurado para entrada de corriente. El módulo se puede dañar u operar de manera incorrecta.*

### **Conexión del Módulo NIO4V**

Pautas para la conexión del módulo:

- Todos los terminales comunes analógicos (ANL COM) están conectados eléctricamente dentro del módulo. ANL COM *no* está conectado a tierra dentro del módulo.
- Los voltajes en los terminales IN+ y IN– deben mantenerse dentro de  $\pm 20$  voltios en relación a ANL COM para asegurar una operación correcta del canal de entrada. Esto aplica para la operación del canal de entrada de corriente y voltaje.
- Las salidas de voltaje (OUT 0 y OUT 1) están referenciadas a ANL COM.
- La resistencia de carga (R1) para un canal de salida de voltaje debe ser igual o mayor que 1 K ohms.

La conexión de las entradas y salidas del módulo analógico NIO4V se muestra en la figura 4.19.

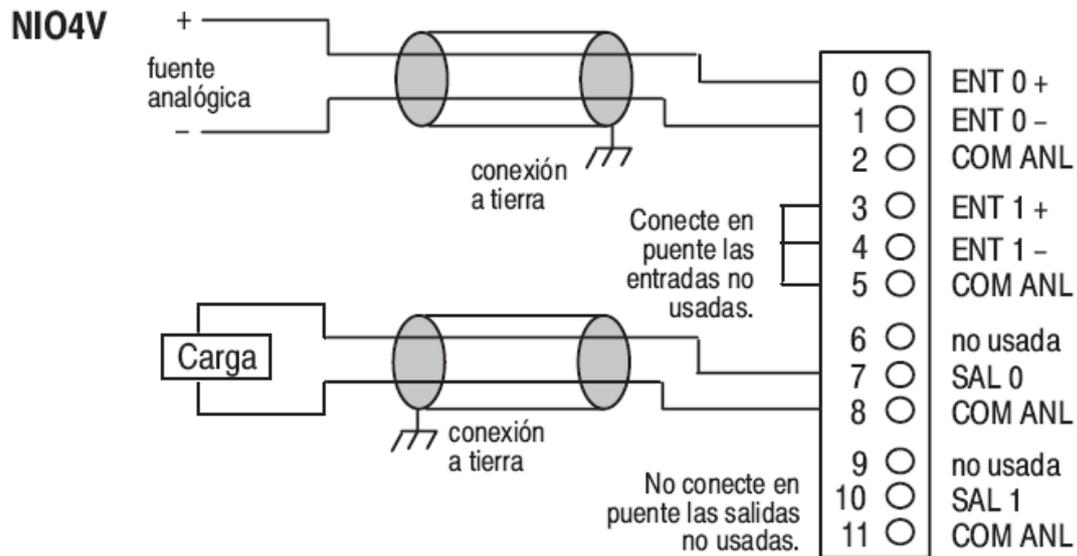


Figura. 4.19. Conexión I/O del módulo NIO4V

**Direccionamiento del Módulo NIO4V**

Cada canal de entrada de los módulos NIO4V se direcciona como una sola palabra en la tabla de imagen de entrada y cada canal de salida del módulo se direcciona como una sola palabra en la tabla de imagen de salida.

Los módulos NIO4V usan un total de 2 palabras de entrada y 2 palabras de salida.

Los valores de entrada convertidos de los canales 0 y 1 se direccionan como palabras 0 y 1 de la ranura en donde reside el módulo. Los valores de salida de los canales de salida 0 y 1 se direccionan como palabras de salida 0 y 1 de la ranura en donde reside el módulo.

**Ejemplo** – Si se desea direccionar el canal de salida 0 del módulo NIO4V en la ranura 3, se direccionaría como palabra de salida 0 en la ranura 3 (O:3.0).

Es decir se direccionan de manera similar al direccionamiento de las entradas y salidas discretas con la diferencia de que no se especifica el número de terminal.

<i>Formato</i>	<i>Explicación</i>	
<b>O:e.s</b>	<b>O</b>	Salida (Output)
	<b>I</b>	Entrada (Input)
	<b>:</b>	Delimitador de elemento
	<b>e</b>	Número de Slot (en el que se   El módulo procesador (CPU) es el

<b>I:e.s</b>		encuentra el módulo de I/O)	Slot 0.
	.	Delimitador de palabra	
	s	Número de palabra	0 ó 1 dependiendo de la salida o entrada que se requiera.

**Tabla. 4.17. Direccionamiento I/O**

### Conversión de datos de Entrada Analógica

Las entradas analógicas convierten señales de corriente y voltaje en valores binarios de complemento a dos de 16 bits.

La siguiente tabla identifica los límites de entrada de corriente y voltaje para los canales de entrada, el número de bits significativos para la aplicación usando límites de entrada menores que la escala completa, y su resolución.

<i>Límites de Voltaje/Corriente</i>	<i>Representación decimal</i>	<i>Número de bits significativos</i>	<i>Resolución por LSB (bit menos significativo)</i>
-10 VCC a (+10 VCC - 1LSB)	-32768 a +32767	16 Bits	305.176 $\mu$ V
0 a (10 VCC - 1LSB)	0 a 32767	15 Bits	
0 a 5 VCC	0 a 16384	14 Bits	
1 a 5 VCC	3277 a 16384	13.67 Bits	
-20 mA a +20 mA	-16384 a +16384	15 Bits	1.22070 $\mu$ A
0 a +20 mA	0 a 16384	14 Bits	
4 a +20mA	3277 a 16384	13.67 Bits	

**Tabla. 4.18. Límites de entrada de corriente y voltaje**

1. Para determinar un voltaje aproximado que el valor de entrada representa, utilice la ecuación siguiente:

$$\frac{10\text{ V}}{32768} \times \text{Valor de entrada} = \text{voltaje de entrada (V)}$$

El valor de entrada es el valor decimal de la palabra en la imagen de entrada para la entrada analógica correspondiente. Por ejemplo, si un valor de entrada de -16021 está en la imagen de entrada, el voltaje de entrada calculado será:

$$\frac{10 V}{32768} \times -16021 = -4.889221 (V)$$

Éste es el valor calculado, el valor actual puede variar según las limitaciones de exactitud del módulo.

2. Para determinar una corriente aproximada que un valor de entrada representa, se puede utilizar la ecuación siguiente:

$$\frac{20 mA}{16384} \times Valor\ de\ entrada = corriente\ de\ entrada\ (mA)$$

El valor de entrada es el valor decimal de la palabra en la imagen de entrada para la entrada analógica correspondiente. Por ejemplo, si un valor de entrada de 4096 está en la imagen de entrada, la corriente de entrada calculada será:

$$\frac{20 mA}{16384} \times 4096 = 5 (mA)$$

Éste es el valor calculado, el valor actual puede variar según las limitaciones de exactitud del módulo.

### **Conversión de datos de Salida Analógica**

Las salidas analógicas convierten un valor binario de complemento a dos en una señal de salida analógica. Debido a que los canales de salida analógica tienen un convertidor de 14 bits, los 14 bits más significativos de este número de 16 bits son los bits que convierte el canal de salida.

Los módulos NIO4V tienen capacidad para dos salidas de voltaje, con límites desde -10 hasta +10 VCC.

La siguiente tabla identifica los límites de voltaje para canales de salida, el número de bits significativos para aplicaciones que usan límites de salida menores que la escala completa, y su resolución.

<i>Límites de Voltaje</i>	<i>Representación decimal para la palabra de salida</i>	<i>Número de bits significativos</i>	<i>Resolución por LSB (bit menos significativo)</i>
-10 VCC a (+10 VCC - 1LSB)	-32768 a +32764	14 Bits	1.22070 mV
0 a (10 VCC - 1LSB)	0 a 32764	13 Bits	
0 a 5 VCC	0 a 16384	12 Bits	
1 a 5 VCC	3277 a 16384	11.67 Bits	

**Tabla. 4.19. Límites de salida de voltaje**

Nota: la Resolución por LSB (bit menos significativo) es 1.22070 mV, en donde la posición de LSB en la palabra de salida se indica así:

														LSB	X		X
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		

X = Bit no utilizado

LSB = Bit Menos Significativo

Use la siguiente ecuación para determinar el valor decimal de la salida de voltaje:

$$\frac{32768}{10 VCC} \times \text{Salida de voltaje deseada (VCC)} = \text{Valor decimal de Salida}$$

Por ejemplo, si se desea un valor de salida de 1 VCC, el valor que se debe colocar en la palabra correspondiente en la imagen de salida se puede calcular de la siguiente manera:

$$\frac{32768}{10 VCC} \times 1 VCC = 3277$$

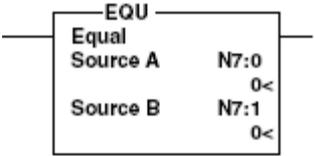
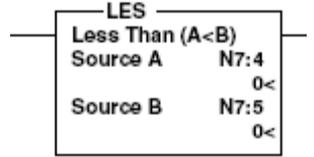
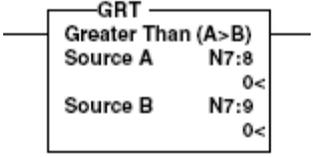
### Formato de direccionamiento de datos de tipo Entero

<i>Formato</i>	<i>Explicación</i>	
<b>Nf:e/b</b>	<b>N</b>	Entero (Integer)
	<b>f</b>	Número de archivo. Es 7 por defecto. En los SLC puede asignarse un número entre 9 y 255 para almacenamiento adicional.
	<b>:</b>	Delimitador de elemento
	<b>e</b>	Número de Estos son elementos de 1 palabra (16 bit) cada

	elemento	uno. El rango es entre 0-255.
/	Delimitador de bit	
<b>b</b>	Número de Bit	De 0 a 15 (16 Bits)

Tabla. 4.20. Direccionamiento ENTERO

### Instrucciones de Comparación

<i>Instrucciones de Comparación</i>		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
	Equal (EQU)	<p>Use esta instrucción para comparar si dos valores son iguales. Si A y B son iguales, la instrucción es verdadera. Si A y B no son iguales, la instrucción es Falsa.</p> <p>A debe ser una dirección, mientras que B puede ser una constante o una dirección. Los enteros negativos son almacenados en complemento a dos.</p>
	Less Than (LES)	<p>Use esta instrucción para comparar si un valor (A), es menor que otro (B). Si A&lt;B, la instrucción es verdadera. Si A&gt;B, la instrucción es falsa.</p> <p>A debe ser una dirección, mientras que B puede ser una constante o una dirección. Los enteros negativos son almacenados en complemento a dos.</p>
	Greater Than (GRT)	<p>Use esta instrucción para comparar si un valor (A), es mayor que otro (B). Si A&gt;B, la instrucción es verdadera. Si A&lt;B, la instrucción es falsa.</p> <p>A debe ser una dirección, mientras que B puede ser una constante o una dirección. Los</p>

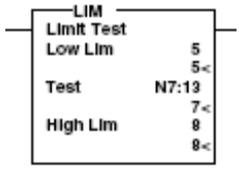
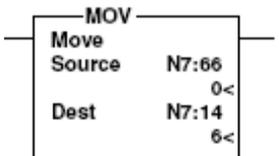
		<p>enteros negativos son almacenados en complemento a dos.</p>
	<p>Limit Test (LIM)</p>	<p>Use esta instrucción para verificar si un valor está dentro o fuera de un rango especificado.</p>

Tabla. 4.21. Instrucciones de Comparación

**Instrucciones de Transferencia de datos**

<i>Instrucciones de Transferencia de datos</i>																	
SÍMBOLO	NOMBRE	DESCRIPCIÓN															
	<p>Move (MOV)</p>	<p>Mueve el valor de la fuente (Source) hacia la dirección del destino (Dest). Si las condiciones que preceden la instrucción son verdaderas. La instrucción mueve los datos una vez cada Scan. La fuente puede ser una dirección o constante. El destino debe ser obligadamente una dirección.</p> <p>Afecta los bits aritméticos de la siguiente manera:</p> <table border="1" data-bbox="742 1310 1348 1814"> <thead> <tr> <th colspan="2">BIT</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>S:0/0</td> <td>Carry (C)</td> <td>Siempre se resetea</td> </tr> <tr> <td>S:0/1</td> <td>Overflow (V)</td> <td>Siempre se resetea</td> </tr> <tr> <td>S:0/2</td> <td>Zero (Z)</td> <td>Si algún resultado es 0 este se hace 1, o sino se resetea</td> </tr> <tr> <td>S:0/3</td> <td>Sign (S)</td> <td>Si algún resultado es (-) este se hace 1, o sino se resetea</td> </tr> </tbody> </table>	BIT		Valor	S:0/0	Carry (C)	Siempre se resetea	S:0/1	Overflow (V)	Siempre se resetea	S:0/2	Zero (Z)	Si algún resultado es 0 este se hace 1, o sino se resetea	S:0/3	Sign (S)	Si algún resultado es (-) este se hace 1, o sino se resetea
BIT		Valor															
S:0/0	Carry (C)	Siempre se resetea															
S:0/1	Overflow (V)	Siempre se resetea															
S:0/2	Zero (Z)	Si algún resultado es 0 este se hace 1, o sino se resetea															
S:0/3	Sign (S)	Si algún resultado es (-) este se hace 1, o sino se resetea															

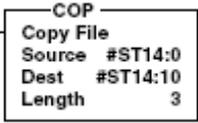
	Copy File (COP)	Esta instrucción copia bloques de datos de un lugar a otro. NO utiliza los bits de Status. La fuente es la dirección de los datos que se quieren copiar. El destino es la dirección de inicio donde se va a copiar los datos. Length es el número de elementos que se van a copiar (máximo 128 palabras). Se debe usar el indicador de archivo (#) en la dirección.
-----------------------------------------------------------------------------------	--------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla. 4.22. Instrucciones de Transferencia de Datos



*Si se quiere mover una palabra de datos sin afectar los bits aritméticos (S:0), use la instrucción COP con una longitud (Length) de 1 en lugar de la instrucción MOV.*

### Instrucciones Matemáticas.

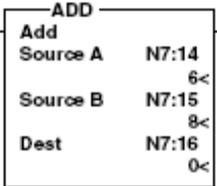
<i>SUMA</i>		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
	Add (ADD)	Suma un valor (A) a otro (B) y coloca el resultado en el destino.

Tabla. 4.23. Instrucción ADD

La instrucción ADD afecta a los bits aritméticos de la siguiente manera:

BIT		ADD le afecta porque...
S:0/0	Carry (C)	Toma el valor de 1 si se ha generado carry. O sino se resetea
S:0/1	Overflow (V)	Toma el valor de 1 si se genera Overflow (desbordamiento) en el destino. O sino se resetea
S:0/2	Zero (Z)	Toma el valor de 1 si el resultado es 0. O sino se resetea

S:0/3	Sign (S)	Toma el valor de 1 si el resultado es negativo. O sino se resetea
-------	----------	-------------------------------------------------------------------

Tabla. 4.24. Bits de la Instrucción ADD

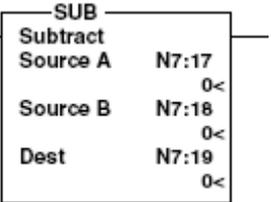
<b>RESTA</b>		
<b>SÍMBOLO</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>
	Subtract (SUB)	Resta un valor (B) de otro (A) y coloca el resultado en el destino.

Tabla. 4.25. Instrucción SUB

La instrucción SUB afecta a los bits aritméticos de la siguiente manera:

<b>BIT</b>		<b>SUB le afecta porque...</b>
S:0/0	Carry (C)	Toma el valor de 1 si se detecta un borrow (préstamo). O sino se resetea
S:0/1	Overflow (V)	Toma el valor de 1 si se genera Underflow (desbordamiento inferior) en el destino. O sino se resetea
S:0/2	Zero (Z)	Toma el valor de 1 si el resultado es 0. O sino se resetea
S:0/3	Sign (S)	Toma el valor de 1 si el resultado es negativo. O sino se resetea

Tabla. 4.26. Bits de la Instrucción SUB

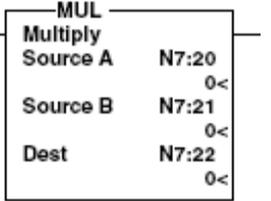
<b>MULTIPLICACIÓN</b>		
<b>SÍMBOLO</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>
	Multiply (MUL)	Multiplica un valor (A) por otro valor (B) y coloca el resultado en el destino.

Tabla. 4.27. Instrucción MUL

La instrucción MUL afecta a los bits aritméticos de la siguiente manera:

BIT		MUL le afecta porque...
S:0/0	Carry (C)	Siempre lo resetea
S:0/1	Overflow (V)	Toma el valor de 1 si se genera Overflow (desbordamiento) en el destino. O sino se resetea
S:0/2	Zero (Z)	Toma el valor de 1 si el resultado es 0. O sino se resetea
S:0/3	Sign (S)	Toma el valor de 1 si el resultado es negativo. O sino se resetea

Tabla. 4.28. Bits de la instrucción MUL

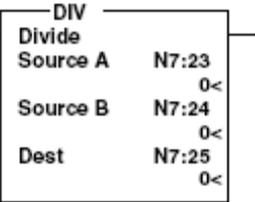
DIVISIÓN		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
	Divide (DIV)	Divide un valor (A) para otro valor (B). El cuociente redondeado es colocado en el destino. El cuociente no redondeado es almacenado en la palabra más significativa del registro matemático. El residuo es colocado en la palabra menos significativa del registro matemático.

Tabla. 4.29. Instrucción DIV

La instrucción DIV afecta a los bits aritméticos de la siguiente manera:

BIT		DIV le afecta porque...
S:0/0	Carry (C)	Siempre lo resetea
S:0/1	Overflow (V)	Toma el valor de 1 si se genera una división para cero o un Overflow (desbordamiento) en el destino. O sino se resetea
S:0/2	Zero (Z)	Toma el valor de 1 si el resultado es 0. O sino se resetea. Es indefinido si existe overflow.
S:0/3	Sign (S)	Toma el valor de 1 si el resultado es negativo. O sino se resetea. Es indefinido si existe overflow.

Tabla. 4.30. Bits de la Instrucción DIV

#### 4.6.4. PLANTEAMIENTO

Se tiene un tanque de almacenamiento de un líquido como se muestra en la figura 4.20., el cual incluye dos válvulas de accionamiento eléctrico para carga y descarga del fluido. La

activación de la válvula V1 permite la carga y V2 la descarga, respectivamente. Adicionalmente el tanque posee un medidor ultrasónico de nivel de 0 a 20 metros el cual entrega una salida de 0 a 10 voltios respectivamente.

Se desea que el tanque pueda operar en dos modos: MANUAL y AUTOMÁTICO.

En el modo MANUAL, el operador puede activar y desactivar las válvulas de acuerdo a su conveniencia, sin considerar el nivel en que se encuentra la gasolina.

En el modo AUTOMÁTICO, la lógica de operación se realiza de acuerdo a las siguientes condiciones:

- La válvula V1 se activa cuando el nivel del líquido está por debajo de 15m y se desactiva cuando el nivel alcance los 20 m.

La válvula V2 se activa cuando el nivel sobrepase los 15m y se desactiva cuando el nivel está por debajo de 5m.

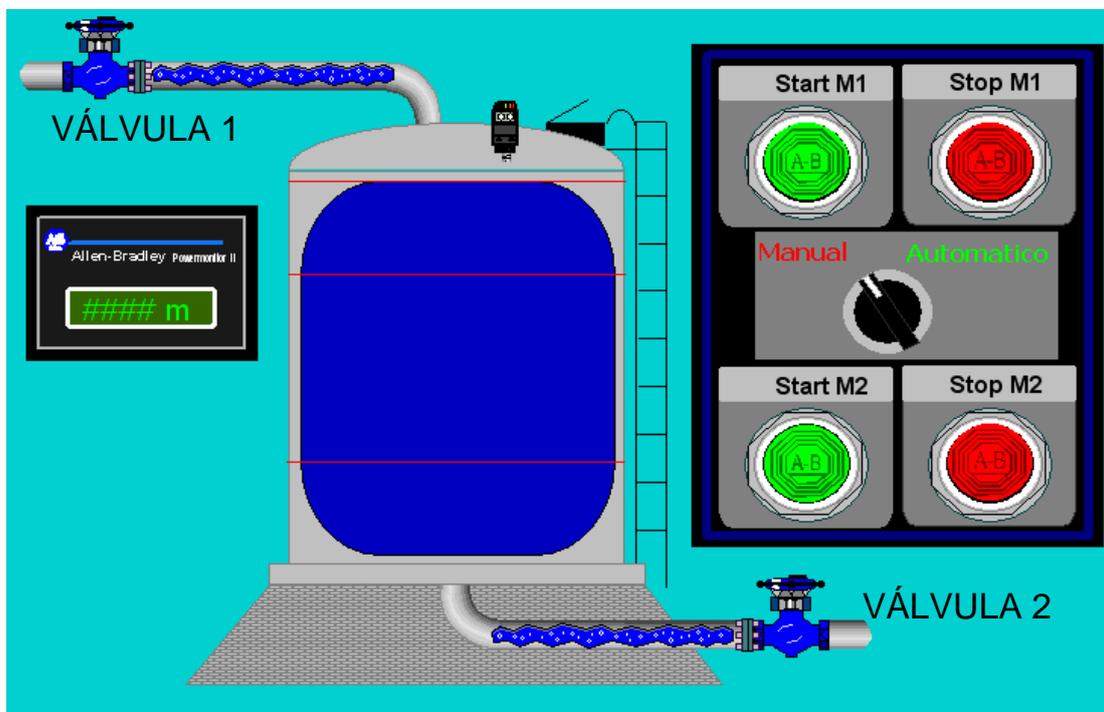


Figura. 4.20. Planteamiento Práctica #6

#### **4.6.5. PROCEDIMIENTO**

Para la realización de esta práctica el procedimiento a seguir es el siguiente:

1. Configurar el driver en RSLinx
2. Ejecutar RSLogix500, crear un nuevo archivo y configurar el tipo de PLC y sus módulos (de ser el caso) en el programa
3. Realizar el programa Ladder que cumpla con la lógica de control planteada en esta práctica.
4. Transferir el programa.
5. Ejecutar el programa en el PLC y probar si la lógica del programa desarrollado funciona correctamente en el Modo ONLINE.
6. Conectar las Entradas y Salidas del PLC al Tablero de simulación correspondiente, de acuerdo a la asignación de éstas en el programa.
7. Verificar si el PLC trabaja correctamente de acuerdo al planteamiento de la práctica.

#### **4.6.6. PRUEBAS Y RESULTADOS**

#### **4.6.7. CONCLUSIONES Y RECOMENDACIONES**

## 4.7. CONTROL LÓGICO 7: REGISTROS DE RELOJ/CALENDARIO

### 4.7.1. OBJETIVOS

- Familiarizarse con el uso y aplicación de los Registros de Reloj/Calendario
- Familiarizarse con las instrucciones de Comparación.

### 4.7.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 5/03 0 superior
- ✓ 1 Módulo de Entradas Discretas.
- ✓ 1 Módulo de Salidas Discretas.
- ✓ 1 Computador Personal, con instalación de RSLogix500 y RSLinx
- ✓ 1 Tablero Didáctico de Simulación AB-07
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

### 4.7.3. MARCO TEÓRICO

#### Reloj Calendario

Los procesadores SLC 5/03 o superiores poseen registros incorporados de Reloj/Calendario. Estos registros se encuentran dentro de los registros de STATUS del procesador aunque se los clasifica como registros de *Configuración Dinámica*.

La precisión del reloj/calendario en este tipo de procesadores es de  $\pm 54$  segundos/mes a  $+25$  °C ó  $\pm 81$  segundos/mes a  $+60$  °C

A continuación se describen más detalladamente cada uno de estos registros:

Dirección	Nombre	Descripción
S:37	Clock/Calendar Year	Este valor contiene el valor correspondiente al año. Tiene un rango valido de 0 a 65535.
S:38	Clock/Calendar Month	Este valor contiene el valor correspondiente al mes. Tiene un rango valido de 1 a 12. Enero

		equivale al valor de 1.
S:39	Clock/Calendar Day	Este valor contiene el valor correspondiente al día. Tiene un rango valido de 1 a 31. El primer día del mes equivale al valor de 1.
S:53L	Day-of-Week	Este valor contiene el valor correspondiente al día de la semana. Tiene un rango valido de 0 a 6. El día domingo equivale al valor de 0.
S:40	Clock/Calendar Hours	Este valor contiene el valor correspondiente a la hora. Tiene un rango valido de 0 a 23
S:41	Clock/Calendar Minutes	Este valor contiene el valor correspondiente a los minutos. Tiene un rango valido de 0 a 59.
S:42	Clock/Calendar Seconds	Este valor contiene el valor correspondiente a los segundos. Tiene un rango valido de 0 a 59.

**Tabla. 4.31. Registros Reloj/Calendario**



*Para deshabilitar el reloj calendario ponga ceros a todas la palabras del reloj/calendario (S:37 a S:42).*



*El valor de los segundos S:42 puede no incrementarse exactamente cada segundo, por lo que podría no ser confiable para utilizarse en controles basados en segundos. Si se requiere debe usarse la función STI para tener un control más confiable.*

**Nota:** *Para referencias acerca de las instrucciones de comparación, revisar el marco teórico de la práctica anterior.*

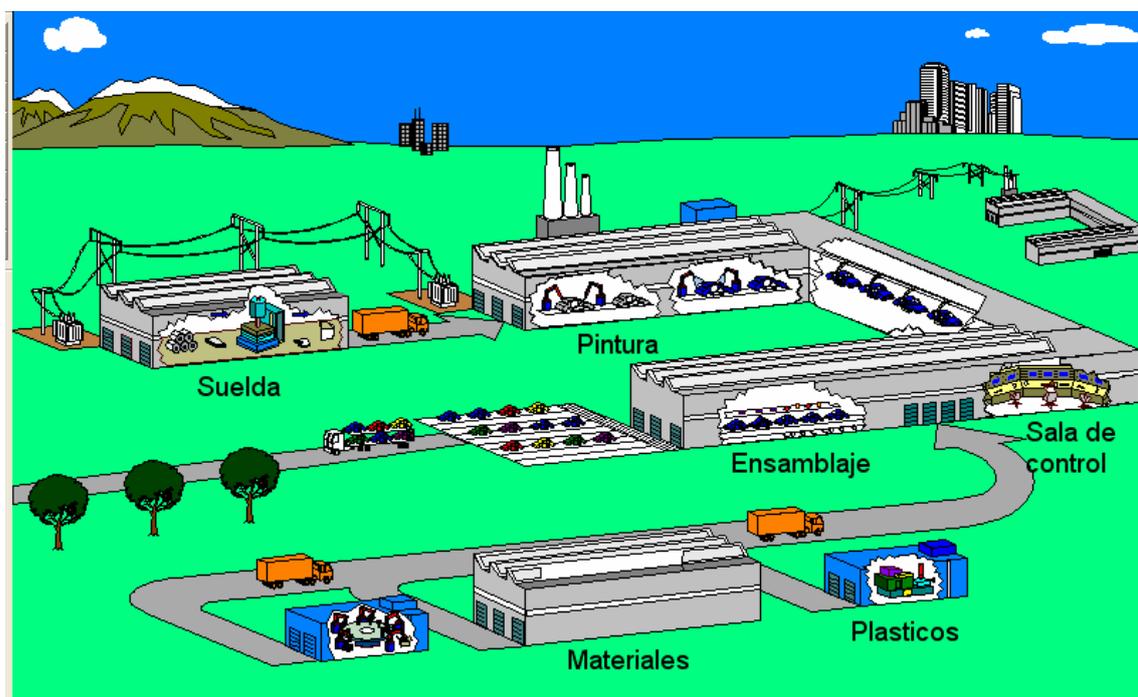
#### 4.7.4. PLANTEAMIENTO

Se necesita implementar un sistema de control automático de iluminación para una planta ensambladora de vehículos como la que se muestra en la figura 4.21. El sistema contará con un único switch de Start/Stop. Cuando esté encendido, deberá encender y apagar la iluminación de las diferentes áreas de la planta de acuerdo a los horarios establecidos en la siguiente tabla:

<i>ÁREA</i>	<i>HORA DE ENCENDIDO</i>	<i>HORA DE APAGADO</i>
Materiales y Suelda	06:00	18:00
Pintura	06:30	20:00
Ensamblaje y Plásticos	07:00	23:00
Sala de control	06:00	23:30

**Tabla. 4.32. Horarios Práctica #7**

**Nota:** Estos horario de trabajo se deben cumplir únicamente de lunes a viernes. Los fines de semana las luminarias debe de permanecer apagadas.



**Figura. 4.21. Planteamiento Práctica #7**

#### 4.7.5. PROCEDIMIENTO

Para la realización de esta práctica el procedimiento a seguir es el siguiente:

1. Configurar el driver en RSLinx
2. Ejecutar RSLogix500, crear un nuevo archivo y configurar el tipo de PLC y sus módulos (de ser el caso) en el programa
3. Realizar el programa Ladder que cumpla con la lógica de control planteada en esta práctica.
4. Transferir el programa.

5. Ejecutar el programa en el PLC y probar si la lógica del programa desarrollado funciona correctamente en el Modo ONLINE.
6. Conectar las Entradas y Salidas del PLC al Tablero de simulación correspondiente, de acuerdo a la asignación de éstas en el programa.
7. Verificar si el PLC trabaja correctamente de acuerdo al planteamiento de la práctica.

#### **4.7.6. PRUEBAS Y RESULTADOS**

#### **4.7.7. CONCLUSIONES Y RECOMENDACIONES**

## 4.8. CONTROL LÓGICO 8: TRANSFERENCIA DE DATOS

### 4.8.1. OBJETIVOS

- Familiarizarse con el uso y aplicación de las instrucciones de transferencia de datos.
- Familiarizarse con el uso y aplicación de las instrucciones matemáticas.

### 4.8.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 ó Micrologix 1000 Analog.
- ✓ 1 Módulo de Entradas Discretas\*.
- ✓ 1 Módulo de Salidas Discretas\*.
- ✓ 1 Módulo de Entradas Analógicas\*.
- ✓ 1 Módulo de Salidas Analógicas\*.
- ✓ 1 Computador Personal, con instalación de RSLogix500 y RSLinx
- ✓ 1 Tablero Didáctico de Simulación AB-08
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

(\*No se requiere si se utiliza un PLC de la serie Micrologix 1000 Analog)

### 4.8.3. MARCO TEÓRICO

#### Instrucciones de Transferencia de Datos

Existen dos tipos de pilas de Datos: FIFO (First In First Out) y LIFO (Last In First Out). Para trabajar con estos tipos de pilas de datos existen dos funciones para cada tipo, una de carga y otra de descarga.

<i>Cargar y Descarga FIFO</i>		
SÍMBOLO	NOMBRE	DESCRIPCIÓN

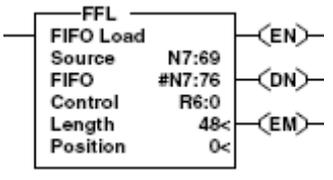
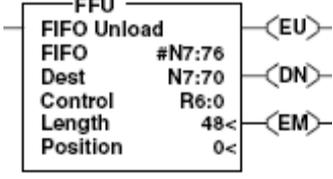
	<p>FIFO Load (FFL)</p>	<p>Cuando las condiciones que la preceden son verdaderas, la instrucción FFL carga palabras dentro una archivo creado por el usuario llamado pila de datos FIFO</p>
	<p>Unload (FFU)</p>	<p>Cuando las condiciones que la preceden son verdaderas, la instrucción FFU descarga palabras desde una pila FIFO, en el mismo orden en el que entraron.</p>

Tabla. 4.33. Instrucciones FFL y FFU

### Parámetros

- **Source:** es la dirección de una palabra ó una constante de programa (-32768 a 32767) que almacena el valor que va a ser ingresado en la pila FIFO. La instrucción FFL coloca este valor en el siguiente elemento disponible de la pila.
- **Destination:** es la dirección de una palabra que contiene el valor que sale de la pila FIFO. La instrucción FFU descarga un valor desde la pila y lo coloca en esta dirección.
- **FIFO:** Es la dirección de la pila. Esta debe ser una dirección indexada (con el símbolo #) de palabra. Puede ser del tipo entrada, salida, bit, status o entero. Se debe programar la misma dirección para ambas instrucciones (FFL y FFU)
- **Length:** es el máximo número de elementos en una pila, con un límite de 128 palabras. Se debe programar el mismo número en ambas instrucciones.
- **Position:** es la siguiente posición disponible donde la instrucción carga o descarga los datos en la pila. Este número cambia con cada operación de carga y descarga. Se usa el mismo número para ambas instrucciones. Este valor se guarda en el registro de control.
- **Control:** es la dirección de un archivo de tipo CONTROL, en este elemento son almacenados los bits de Status, la longitud (length), y la posición. Se utiliza la misma dirección para ambas instrucciones.

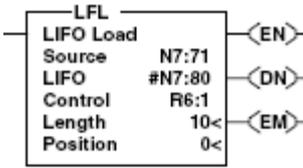
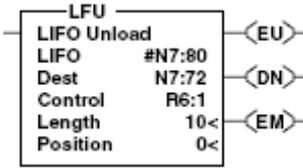
<i>Cargar y Descarga LIFO</i>		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
	LIFO Load (LFL)	Cuando las condiciones que la preceden son verdaderas, la instrucción LFL carga palabras dentro una archivo creado por el usuario llamado pila de datos LIFO
	LIFO Unload (LFU)	Cuando las condiciones que la preceden son verdaderas, la instrucción LFU descarga palabras desde una pila LIFO, en orden opuesto al que entraron.

Tabla. 4.34. Instrucciones LFL y LFU

### Parámetros

- **Source:** es la dirección de una palabra ó una constante de programa (-32768 a 32767) que almacena el valor que va a ser ingresado en la pila LIFO. La instrucción LFL coloca este valor en el siguiente elemento disponible de la pila.
- **Destination:** es la dirección de una palabra que contiene el valor que sale de la pila LIFO. La instrucción LFU descarga un valor desde la pila y lo coloca en esta dirección.
- **LIFO:** Es la dirección de la pila. Esta debe ser una dirección indexada (con el símbolo #) de palabra. Puede ser del tipo entrada, salida, bit, status o entero. Se debe programar la misma dirección para ambas instrucciones (LFL y LFU)
- **Length:** es el máximo número de elementos en una pila, con un límite de 128 palabras. Se debe programar el mismo número en ambas instrucciones.
- **Position:** es la siguiente posición disponible donde la instrucción carga o descarga los datos en la pila. Este número cambia con cada operación de carga y descarga. Se usa el mismo número para ambas instrucciones. Este valor se guarda en el registro de control.
- **Control:** es la dirección de un archivo de tipo CONTROL, en este elemento son almacenados los bits de Status, la longitud (length), y la posición. Se utiliza la misma dirección para ambas instrucciones.



*Excepto cuando se utilicen estas instrucciones de carga y descarga en pares, NUNCA utilice la misma dirección de control para ninguna otra instrucción.*

<i>Compute</i>		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
	Compute (CPT)	Esta instrucción realiza la operación indicada en la expresión. El resultado lo coloca en la dirección Dest.

**Tabla. 4.35. Instrucción CPT**



*Esta instrucción está disponible únicamente en los procesadores SLC 5/03 o superiores. NO en los Micrologix1000.*

Las instrucciones que pueden ser incluidas en la expresión son:

+, -, \* (multiplicación), | (división), SQR, NOT, XOR, OR, AND, TOD, FRD, LN, TAN, ABS, DEG, RAD, SIN, COS, ATN, ASN, ACS, LOG y \*\* (potencia).

Por ejemplo si se quiere hacer la siguiente operación:

$$\sqrt{a^2 + \frac{(b+c)}{2}}$$

Donde,

a = N7:0, b = N7:1 y c = N7:2

Entonces se debe ingresar la siguiente expresión:

SQR ( ( N7:0 \*\* 2 ) + ( ( N7:1 + N7:2 ) | 2 ) )

<i>Escalar valores</i>		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
	Scale with Parameters (SCP)	Esta instrucción produce un valor escalado que mantiene una relación lineal con el valor de entrada. Esta permite ingresar valores enteros y flotantes.
	Scale Data (SCL)	Cuando la instrucción es verdadera, el valor de Source es multiplicado por el valor Rate, luego el resultado redondeado es sumado con el valor Offset y colocado en la dirección Dest.

**Tabla. 4.36. Instrucciones de Escalado**

En el caso de la función SCL, los parámetros que se debe ingresar se calculan de la siguiente manera:

$$\text{Valor Escalado} = (\text{Valor de entrada} \times \text{Rate}) + \text{Offset}$$

$$\text{Rate} = \frac{(\text{Escalado Maximo} - \text{Escalado Minimo})}{(\text{Entrada Maxima} - \text{Entrada Minima})}$$

$$\text{Offset} = \text{Escalado Minimo} - (\text{Entrada Minima} \times \text{Rate})$$



*La instrucción SCP está disponible únicamente en los procesadores SLC 5/03 o superiores.*

#### 4.8.4. PLANTEAMIENTO

En un horno de pintura, se tiene un sensor de temperatura que trabaja en un rango de 0 a 100 °C, el cual envía una señal al PLC en el rango de 0 a 10 V en relación proporcional y lineal con la temperatura. Se requiere que cuando se presione START el PLC tome 6 muestras (una cada 10 segundos) de la temperatura, las mismas que serán almacenadas en

una pila de registros; una vez tomadas las 6 muestras se deberá activar una luz indicadora, calcular el promedio y almacenarlo en un registro. Para cancelar la toma de muestras se deberá presionar STOP.

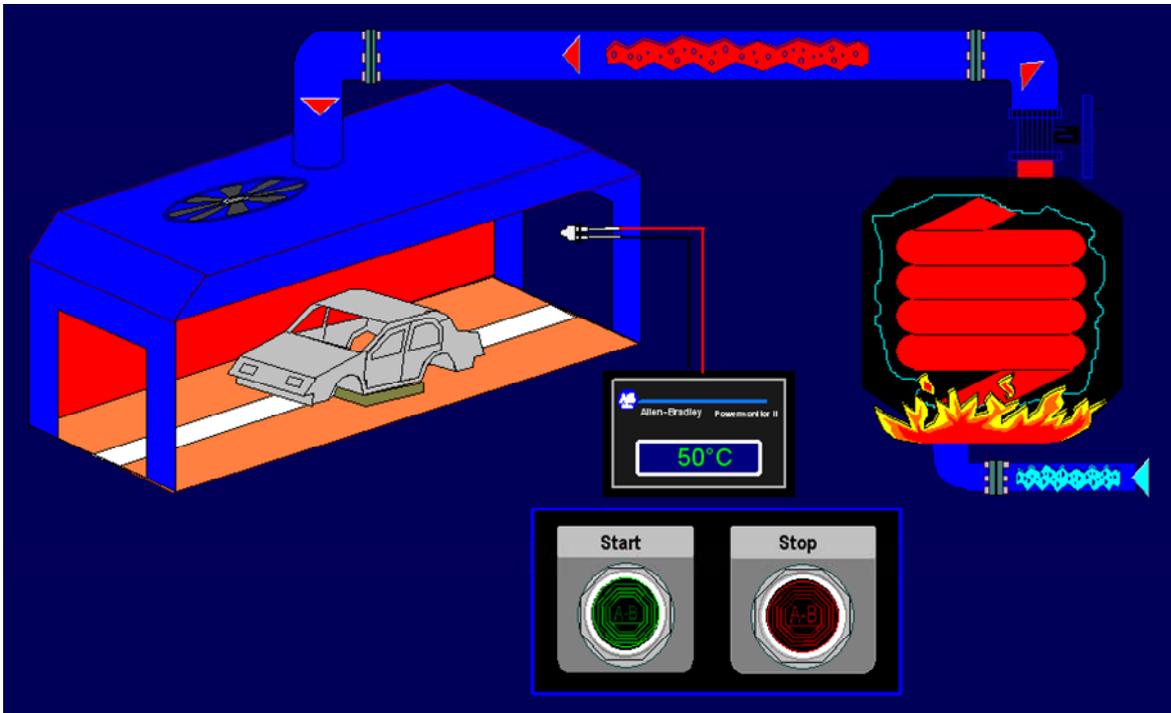


Figura. 4.22. Planteamiento Práctica #8

#### 4.8.5. PROCEDIMIENTO

Para la realización de esta práctica el procedimiento a seguir es el siguiente:

1. Configurar el driver en RSLinx
2. Ejecutar RSLogix500, crear un nuevo archivo y configurar el tipo de PLC y sus módulos (de ser el caso) en el programa
3. Realizar el programa Ladder que cumpla con la lógica de control planteada en esta práctica.
4. Transferir el programa.
5. Ejecutar el programa en el PLC y probar si la lógica del programa desarrollado funciona correctamente en el Modo ONLINE.
6. Conectar las Entradas y Salidas del PLC al Tablero de simulación correspondiente, de acuerdo a la asignación de éstas en el programa.

7. Verificar si el PLC trabaja correctamente de acuerdo al planteamiento de la práctica.

#### **4.8.6. PRUEBAS Y RESULTADOS**

#### **4.8.7. CONCLUSIONES Y RECOMENDACIONES**

## 4.9. CONTROL LÓGICO 9: FLUJO DE PROGRAMA

### 4.9.1. OBJETIVOS

- Familiarizarse con el uso y aplicación de las instrucciones de Flujo de Programa.

### 4.9.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 ó Micrologix.
- ✓ 1 Módulo de Entradas Discretas\*.
- ✓ 1 Módulo de Salidas Discretas\*.
- ✓ 1 Computador Personal, con instalación de RSLogix500 y RSLinx
- ✓ 1 Tablero Didáctico de Simulación AB-01
- ✓ 1 Fuente de alimentación +24 VDC
- ✓ Cables de Conexión

(\*No se requiere si se utiliza un PLC de la serie Micrologix)

### 4.9.3. MARCO TEÓRICO

#### Instrucciones de Flujo de Programa

<i>Jump to Label (JMP) and Label (LBL)</i>		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
— LBL —	Label (LBL)	Esta instrucción es el objetivo de la instrucción JMP que tenga el mismo número. Se debe colocar esta instrucción en el comienzo del escalón (línea del programa ladder). El número asignado debe ser único.
—(JMP)—	Jump to Label (JMP)	La instrucción JMP causa que el controlador se salte escalones (líneas de ladder). Se puede saltar al mismo Label desde una o más instrucciones JMP.

Tabla. 4.37 Instrucciones JMP y LBL

#### **4.9.4. PLANTEAMIENTO**

Se tienen tres motores, cada uno con un único botón de START/STOP (Utilizar el Tablero Didáctico AB-01, pero trabajar con un solo botón para el encendido y apagado). Se desea que la secuencia de encendido sea: Primero el motor M1, luego el motor M2 y finalmente el motor M3. Mientras que para la secuencia de apagado se desea que el primero en desactivarse sea el motor M3, luego el motor M2 y finalmente el motor M1. No se podrá activar o desactivar los motores, en una secuencia diferente a la antes mencionada.

#### **4.9.5. PROCEDIMIENTO**

Para la realización de esta práctica el procedimiento a seguir es el siguiente:

1. Configurar el driver en RSLinx
2. Ejecutar RSLogix500, crear un nuevo archivo y configurar el tipo de PLC y sus módulos (de ser el caso) en el programa
3. Realizar el programa Ladder que cumpla con la lógica de control planteada en esta práctica.
4. Transferir el programa.
5. Ejecutar el programa en el PLC y probar si la lógica del programa desarrollado funciona correctamente en el Modo ONLINE.
6. Conectar las Entradas y Salidas del PLC al Tablero de simulación correspondiente, de acuerdo a la asignación de éstas en el programa.
7. Verificar si el PLC trabaja correctamente de acuerdo al planteamiento de la práctica.

#### **4.9.6. PRUEBAS Y RESULTADOS**

#### **4.9.7. CONCLUSIONES Y RECOMENDACIONES**

## 4.10. CONTROL DE PROCESOS 1: CONTROL ON-OFF CON HISTÉRESIS

### 4.10.1. OBJETIVOS

- Familiarizarse con la técnica de control ON-OFF con histéresis.
- Familiarizarse con el software RSView32.
- Crear una HMI (Interfase Humano Máquina) que interactúe con el PLC.

### 4.10.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 ó Micrologix 1000 Analog.
- ✓ 1 Módulo de Entradas Discretas\*.
- ✓ 1 Módulo de Salidas Discretas\*.
- ✓ 1 Módulo de Entradas Analógicas\*.
- ✓ 1 Computador Personal, con instalación de RSLogix500, RSLinx y RSView32
- ✓ 1 Módulo Degem de simulación “AIR FLOW TEMPERATURE CONTROL SYSTEM PCT-2”.
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

(\*No se requiere si se utiliza un PLC de la serie Micrologix 1000 Analog)

### 4.10.3. MARCO TEÓRICO

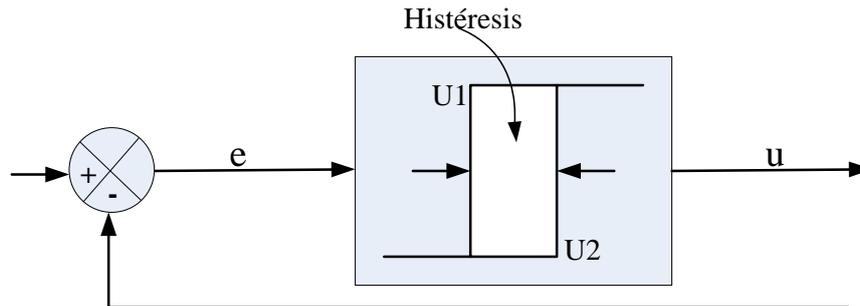
#### Control ON-OFF con Histéresis\*

El control On-Off es la forma más simple de control. En este, el elemento final de control se mueve rápidamente entre una de dos posiciones.

Este tipo de controlador se emplea usualmente con una denominada *banda diferencial*, *histéresis* o *zona muerta* en la que el elemento final de control permanece en su última posición para valores de la variable comprendidos dentro de la *banda diferencial* como se muestra en la figura 4.23.

---

\*La información expuesta de este tema fue extraída del texto 1 citado en las referencias bibliográficas.



**Figura. 4.23. Banda diferencial o Histéresis**

Los ajustes de control se basan en variar el Setpoint y la Banda Diferencial.

Estos controladores funcionan satisfactoriamente si el proceso tiene una velocidad de reacción lenta. Se caracteriza porque las dos posiciones extremas del elemento final de control permiten una entrada y salida de energía al proceso ligeramente superior e inferior con respecto a las necesidades de la operación normal. Es evidente que la variable controlada oscila continuamente y que estas oscilaciones varían en frecuencia y magnitud si se presentan cambios de carga en el proceso.

### **Introducción al software RSView32**

A continuación se describirán los pasos principales para configurar y crear una HMI utilizando el software RSView32.

### **Crear un archivo nuevo y configurar de los canales de comunicación directa**

Para crear y nuevo archivo y configurar correctamente el canal de comunicación, se deben seguir las siguientes instrucciones:

1. Configurar el driver con el Software RSLinx (Referirse a la práctica #1).
2. Ejecutar el software RSView32 y dar clic en *New*.

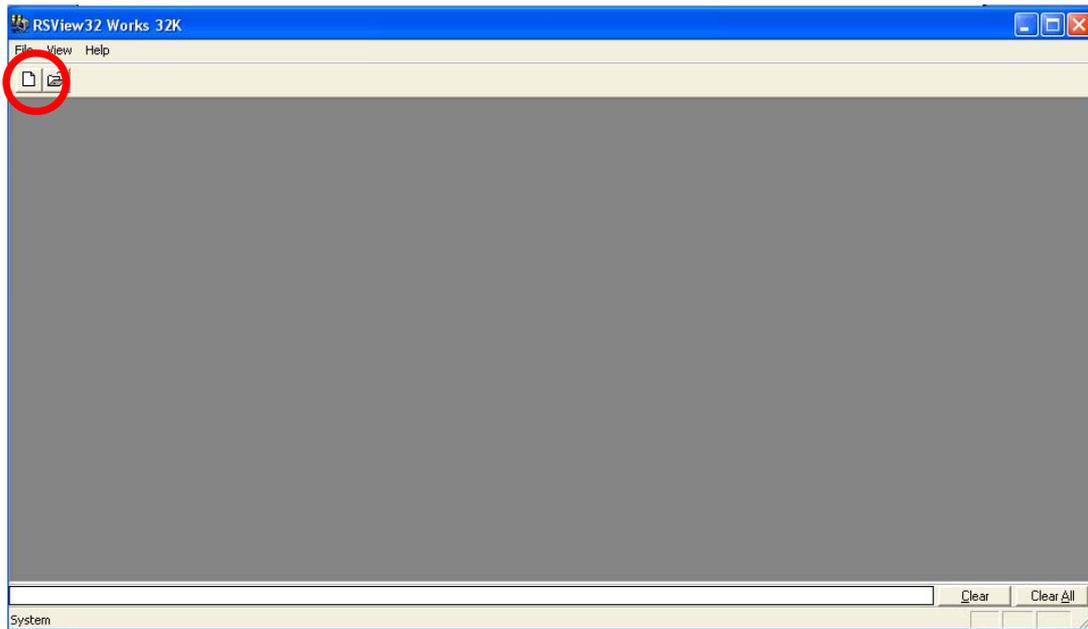


Figura. 4.24. Creación de un archivo nuevo en RSView32 (1)

3. Seleccionar la ubicación donde se quiere crear el archivo, asignar un nombre y presionar *Abrir*.

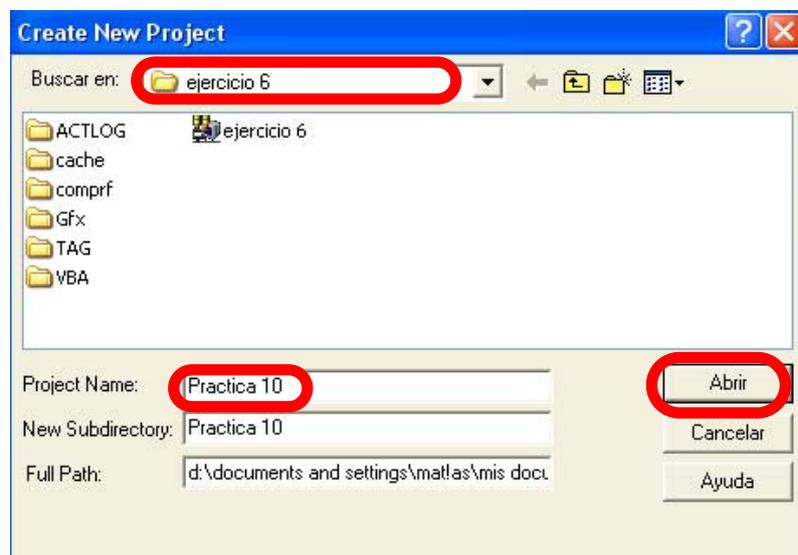


Figura. 4.25. Creación de un archivo nuevo en RSView32 (2)

4. Dar clic en *System* para desplegar las opciones, dar doble clic en *Channel*

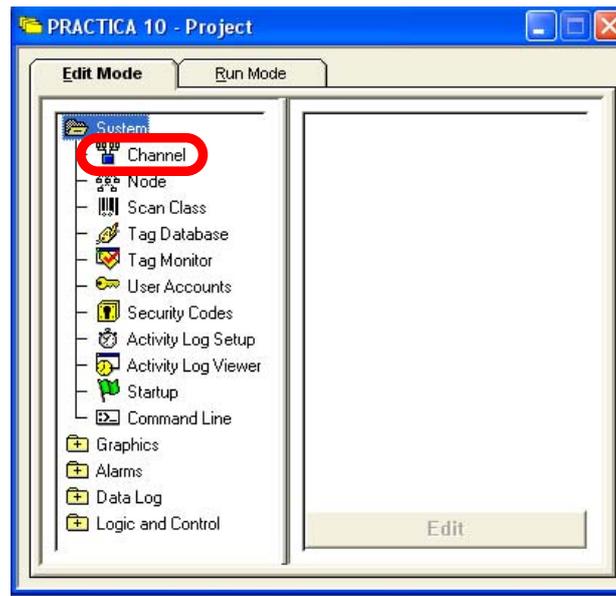


Figura. 4.26. Configuración del canal de comunicación en RSVIEW32 (1)

5. Escoger el número de canal (en este caso 1), en **Network Type** escoger DH-485 (Para conectarse a un SLC con un puerto RS-232), en **Messages** escriba un número entre 1 y 10 (este es el número de mensajes que el software puede enviar por un canal, antes de requerir una respuesta) se recomienda colocar un número alto (10), y si se producen errores de comunicación reducirlo gradualmente puesto que éste número dependerá de la capacidad de almacenamiento del buffer de PLC.

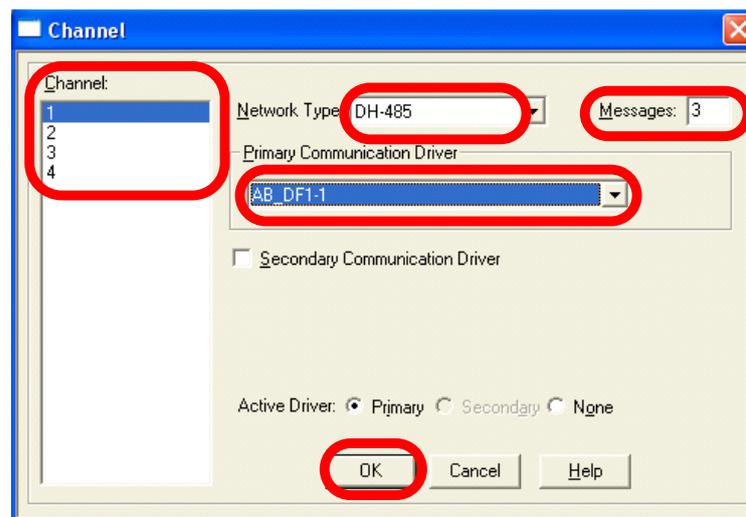


Figura. 4.27. Configuración del canal de comunicación en RSVIEW32 (2)

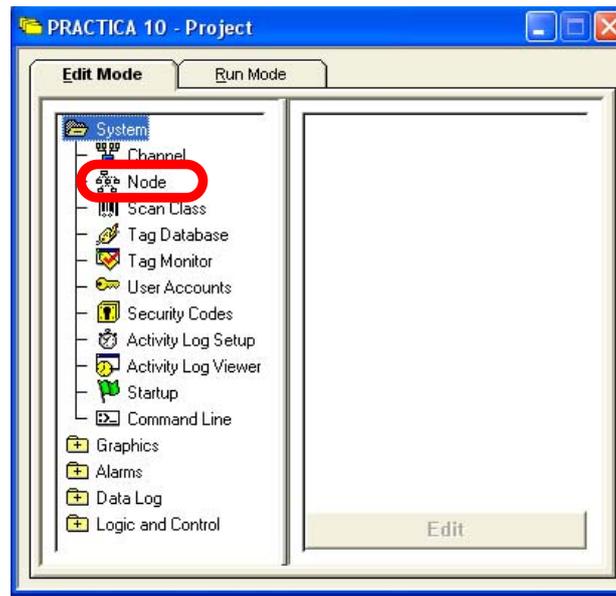
6. Dar doble clic en *Node*

Figura. 4.28. Configuración del nodo de comunicación en RSView32 (1)

7. Seleccionar *Direct Driver*, en *Name* asignar un nombre al nodo, en *Channel* escoger el canal creado anteriormente, en *Station* presionar el botón de explorar para seleccionar el PLC al que se quiere conectar, en *Type* seleccionar el tipo de PLC *SLC 5 (Enhanced)*, y presionar *Aceptar*:

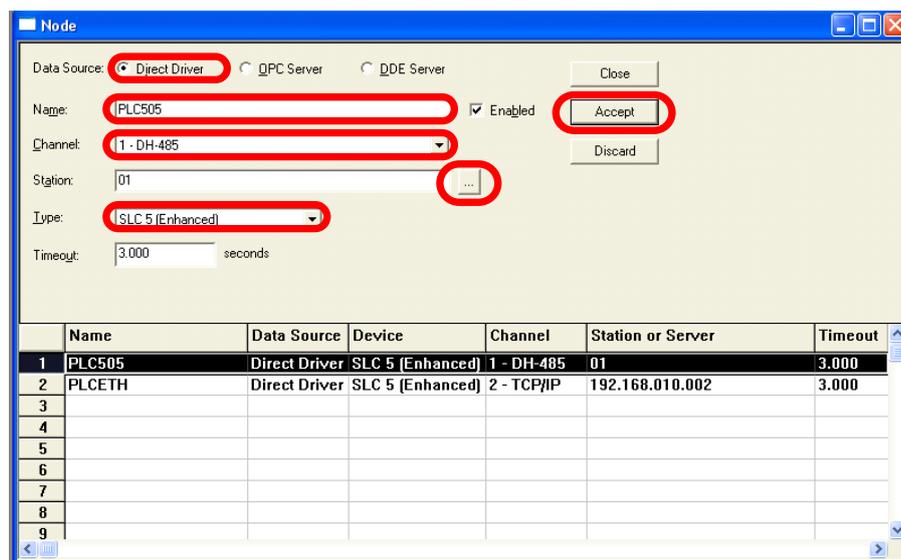


Figura. 4.29. Configuración del nodo de comunicación en RSView32 (2)

## Crear Tags

1. Dar doble clic en *Tag Database*

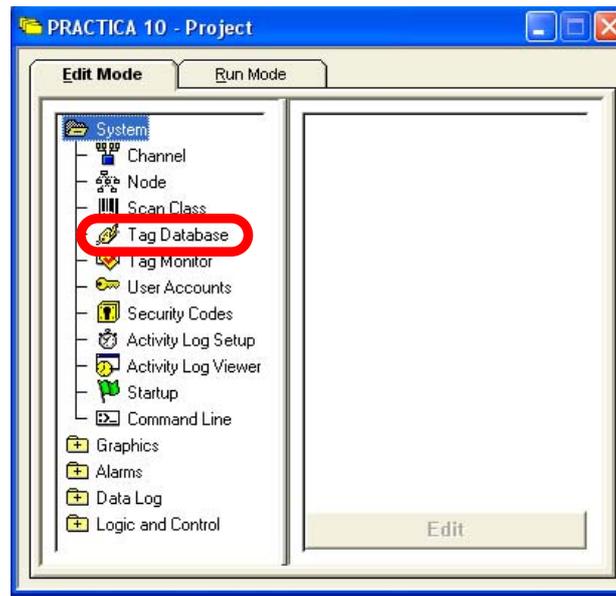


Figura. 4.30. Creación de tags en RSView32

2. En *Name* asignar un nombre (Max. 255 caracteres, no podrá contener espacio Ej: Motor1, Motor-1 ó Motor\_1. Pero NO *Motor 1*). En *Type* seleccionar el tipo de tag que se requiere. En *Description* se puede escribir una pequeña descripción del tag, esta es opcional. En *Data Source* seleccionar *Memory* si el tag recibe únicamente datos internos del software RSView32, seleccionar *Device* si el tag recibe datos de una fuente externa (PLC), para lo que se desplegará las siguientes casillas:

**Node Name:** Seleccionar el nodo creado anteriormente.

**Address:** Especificar la dirección de los datos en el PLC a los cuales el tag debe estar conectado. La sintaxis dependerá del tipo de PLC que se este utilizando.

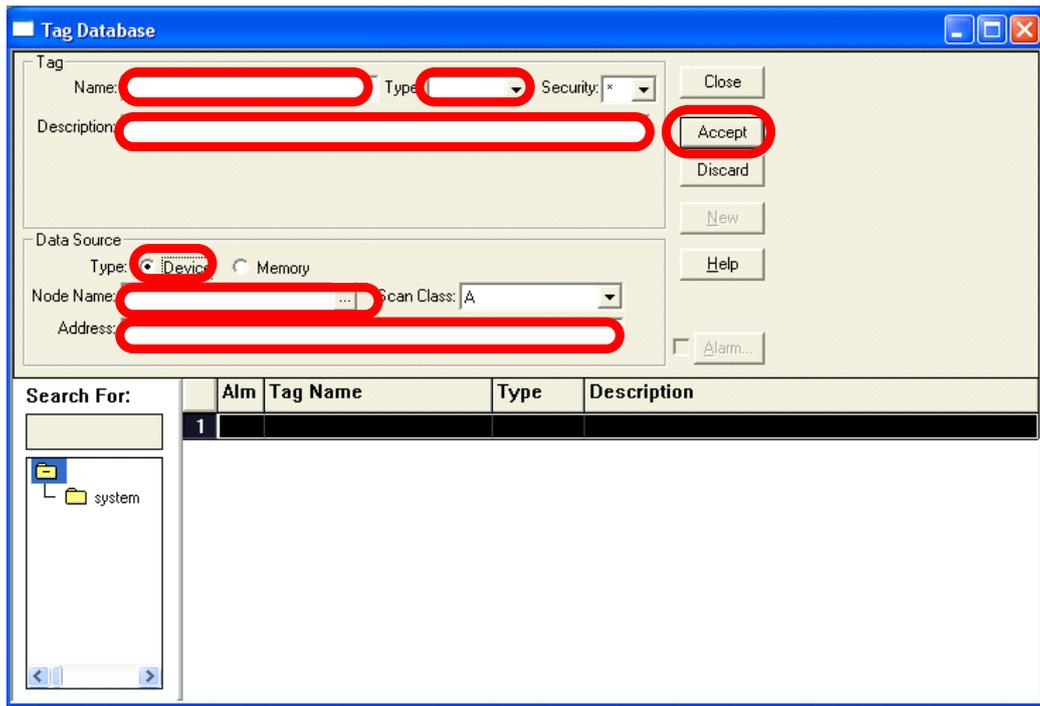


Figura. 4.31. Configurar tags en RSVIEW32



Si se requiere activar o desactivar una salida del PLC a través de un “interruptor” dentro de la HMI, éste no deberá apuntar directamente a la salida del PLC, sino deberá apuntar a un registro de bit, el cual dentro del programa ladder controlará la salida del PLC.

### Crear Pantallas Gráficas

1. Dar clic en **Graphics** para desplegar las opciones, dar doble clic en **Display**

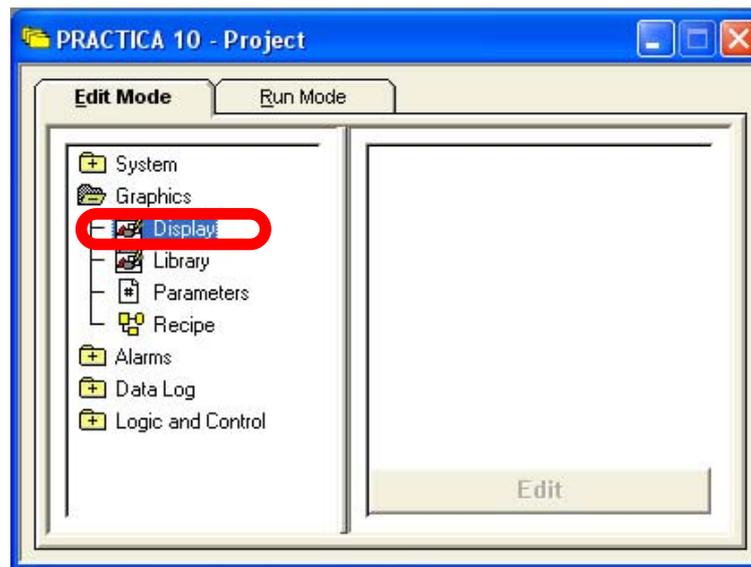


Figura. 4.32. Creación de pantallas gráficas en RSView32 (1)

2. Aparecerá una ventana en blanco, que será en donde se creará la HMI. Además se desplegará la barra de herramientas de dibujo, colores, alineamiento, etc. (sin no aparecen, seleccionarlas en el menú *View*). Adicionalmente, si se quiere insertar gráficos prediseñados se debe dar doble clic en *Library* y copiarlos en nuestra ventana.

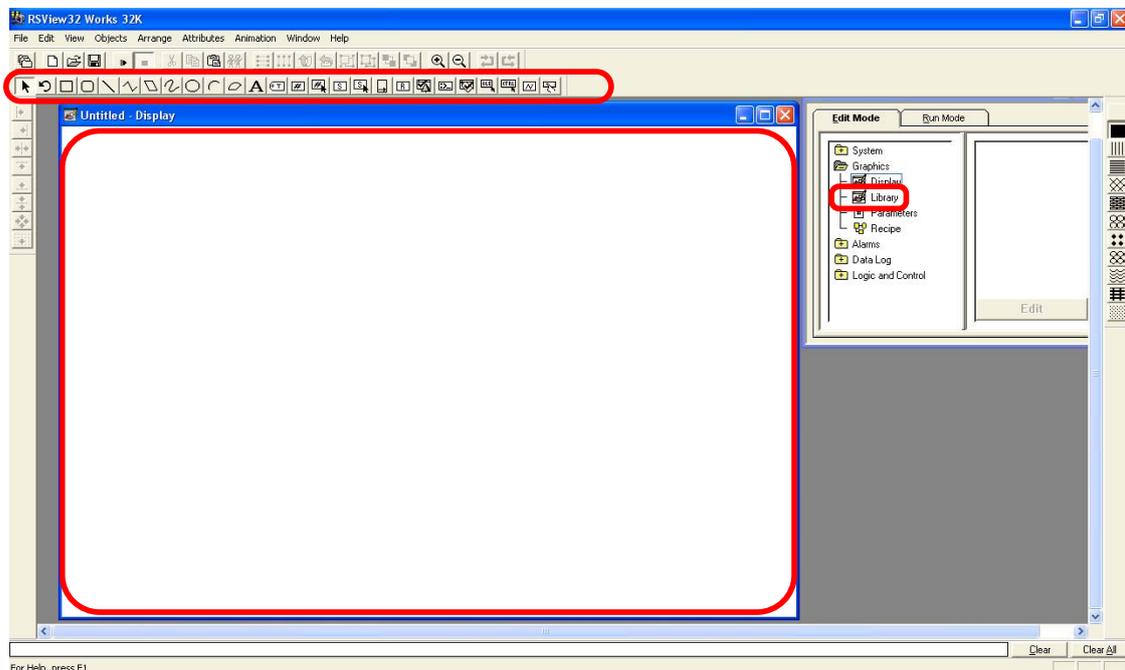


Figura. 4.33. Creación de pantallas gráficas en RSView32 (2)

## Animar Objetos

1. Una vez creados los objetos gráficos (botones, medidores, leds, etc.) se puede crear animaciones a través de la ventana de animación o al dar clic derecho sobre el objeto y seleccionar *Animation*.

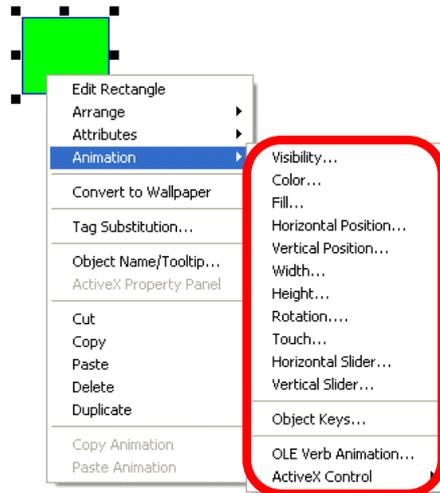


Figura. 4.34. Animación de objetos en RSView32 (1)

2. Como se puede ver en la figura, la ventana de aplicación consta de 4 secciones principales:
  - Tipos de animación
  - Área de la expresión
  - Resultado de a expresión
  - Valores mínimos y máximos

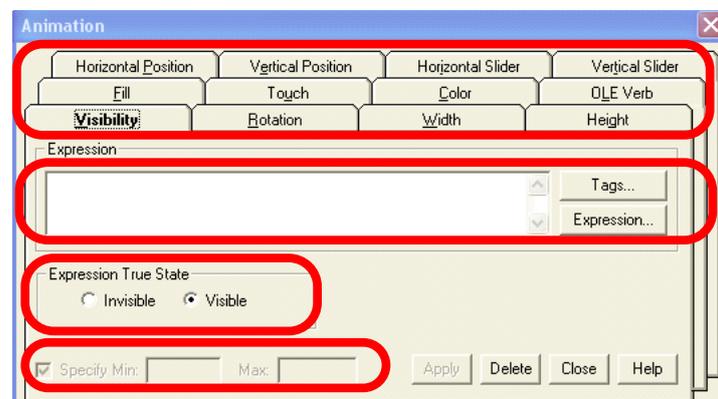


Figura. 4.35. Animación de objetos en RSView32 (2)

3. Para probar la animación creada, se deberá cambiar a Modo **Test**, presionando el botón **Test Run** de la barra de herramientas.



Figura. 4.36. Modo Test Run en RSView32



*El modo TEST, NO es lo mismo que “Correr” el programa puesto que el modo TEST no realiza los cambios de apariencia o posición de la ventana, realizados en el cuadro de dialogo “DISPLAY SETTINGS”.*

### Crear Trends (Gráficos de tendencia)

1. Seleccionar la barra de herramientas Trend, dando clic en el icono respectivo como se ve en la figura 4.37., y seleccionando el área que ocupara esta gráfica de tendencia (trend) en nuestra pantalla.

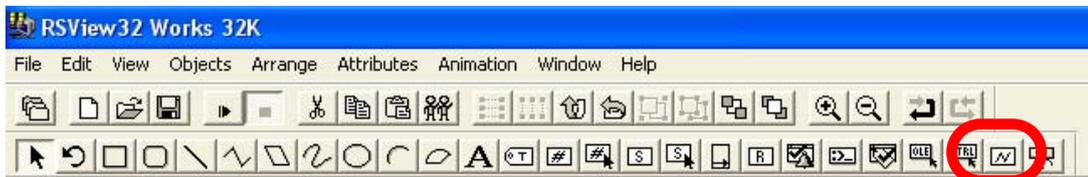


Figura. 4.37. Creación de Trends en RSView32 (1)

2. En la ventana que aparece, presionando la ceja **Trend Configuration**, se encuentran las diferentes opciones de configuración del gráfico.

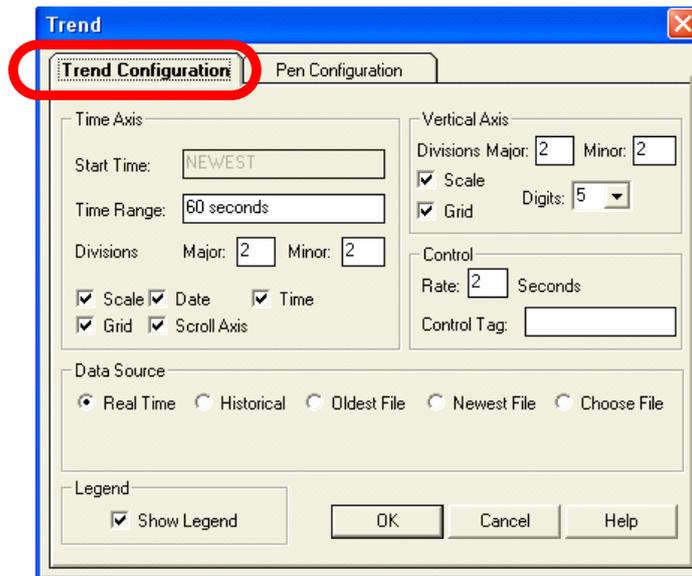


Figura. 4.38. Creación de Trends en RSView32 (2)

3. Presionando la ceja *Pen Configuration*, aparece la ventana en la que se debe asignar un tag a la línea que queremos que se dibuje, pudiendo cambiar también el estilo de la línea.

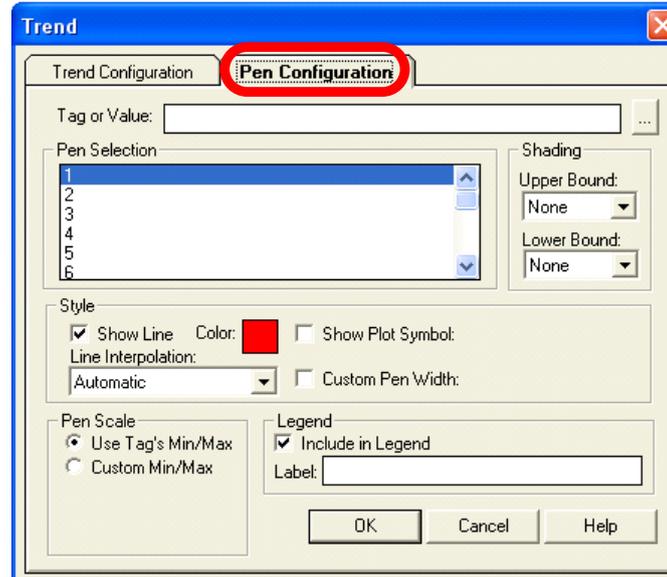


Figura. 4.39. Creación de Trends en RSView32 (3)

## Crear y configurar Eventos

Los eventos son expresiones que inician o ejecutan acciones.

Las expresiones son ecuaciones que contienen valores, Tags, operaciones matemáticas, lógica if–then–else y otras funciones propias de RSVIEW32.

Las acciones son comandos, símbolos y macros de RSVIEW32, una acción puede por ejemplo mostrar una pantalla de error utilizando los comandos de pantalla.

Estos Eventos se utilizan cuando se requiere que se valide una expresión constantemente (mientras se ejecute el evento) y se realice una acción dependiendo del resultado.

1. Para crear un nuevo evento se debe dar clic en “Logic and Control” para desplegar los submenús. Posteriormente se debe dar doble clic en “Events”.

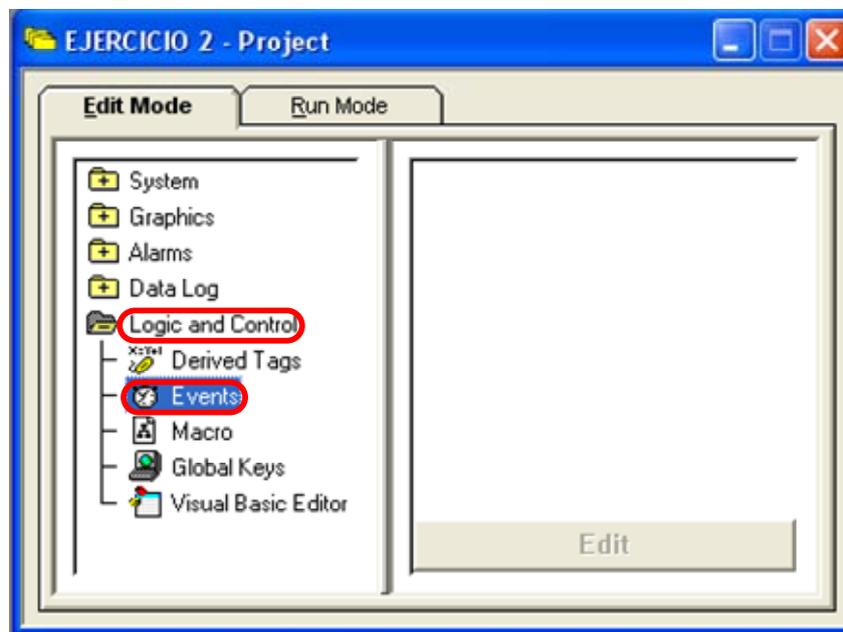


Figura. 4.40. Creación de Eventos en RSVIEW32 (1)

2. Aparece la siguiente pantalla, en la cual en *Action* se debe ingresar la acción se debe realizar el evento, cada vez que la expresión *Expression* sea verdadera. En *Description* se pone una pequeña descripción de la función del evento, si bien esto es opcional es recomendable para facilitar la posterior revisión. En *Expression* se ingresa la expresión que debe ser evaluada, para lo cual existen varias opciones en la parte inferior. Por último, la sección inferior de la ventana nos muestra las diferentes acciones que se han creado en el evento.

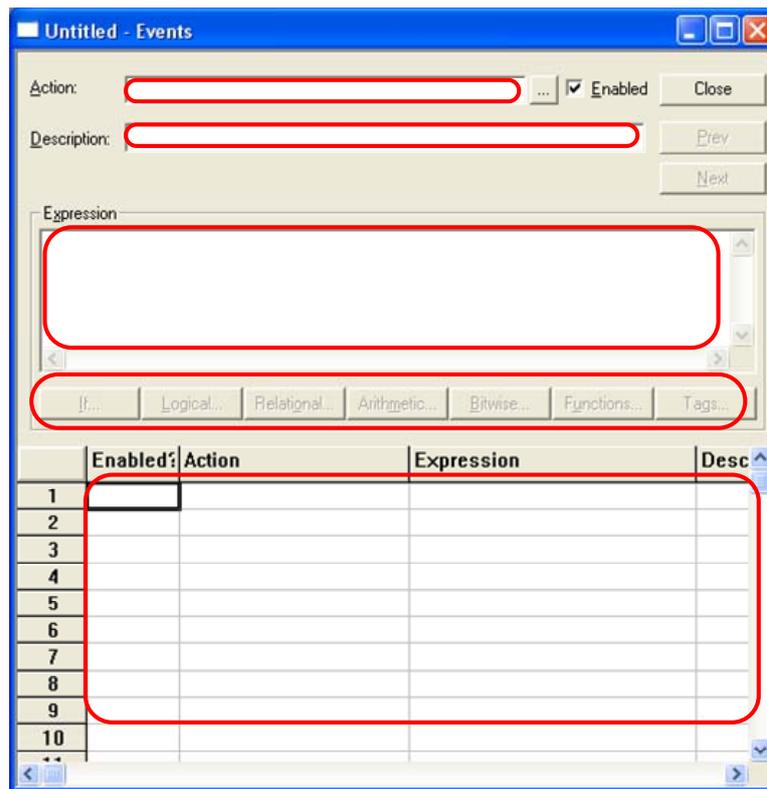


Figura. 4.41. Creación de Eventos en RSView32 (2)

3. Para configurar como se va a evaluar el Evento (Continuo o en Intervalos) se debe dar clic en Setup y posteriormente en Event Setup, lo que despliega la pantalla de configuración.

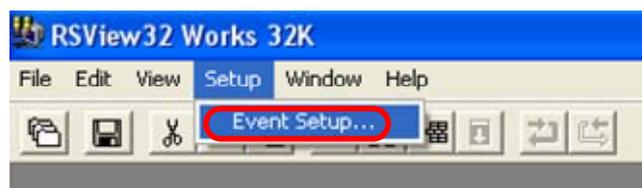


Figura. 4.42. Creación de Eventos en RSView32 (3)

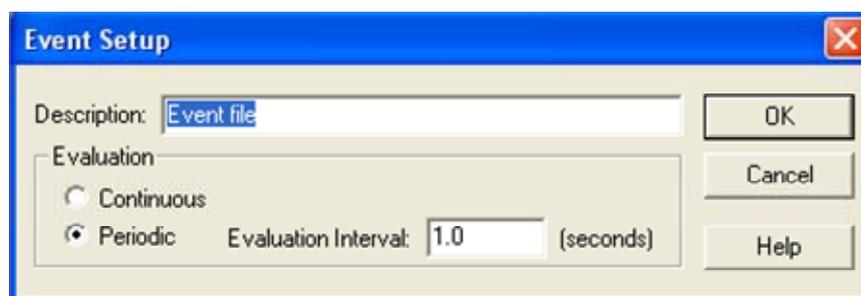


Figura. 4.43. Creación de Eventos en RSView32 (4)

- Una vez creado el evento, este puede ser activado o desactivado a nuestra conveniencia con los comandos *EventOn* y *EventOff* en cualquier línea de comando. (por ejemplo cuando se presione un botón o cuando aparezca una pantalla).

### Ejecutar el Programa

- Para configurar los parámetros de cómo queremos que se ejecute nuestro programa, se debe ingresar a la ventana de opciones, dando doble clic en *Startup*.

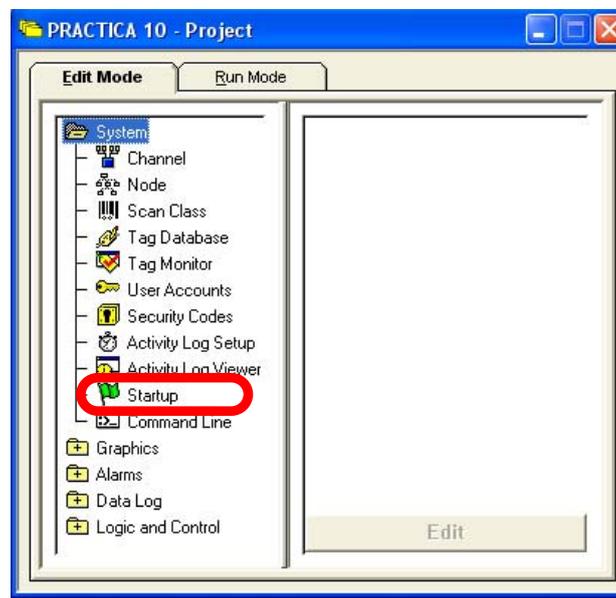


Figura. 4.44. Configurar la corrida del programa en RSVIEW32 (1)

- Aparece la ventana de *Preferences*, en la que se pueden seleccionar diferentes opciones para que aparezcan o no en la pantalla al momento de correr el programa, por ejemplo los botones de minimizar y maximizar.

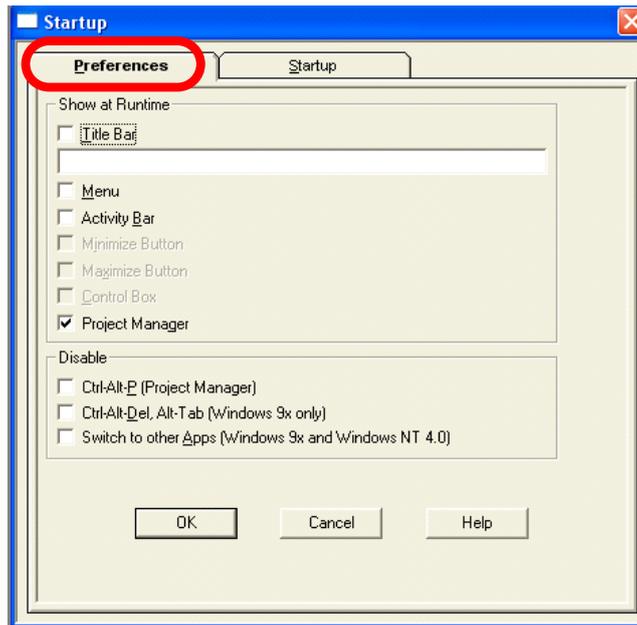


Figura. 4.45. Configurar la corrida del programa en RSVIEW32 (2)

3. Presionando la pestaña de *Startup*, aparecen nuevas opciones como por ejemplo definir cual va a ser nuestra pantalla de inicio.

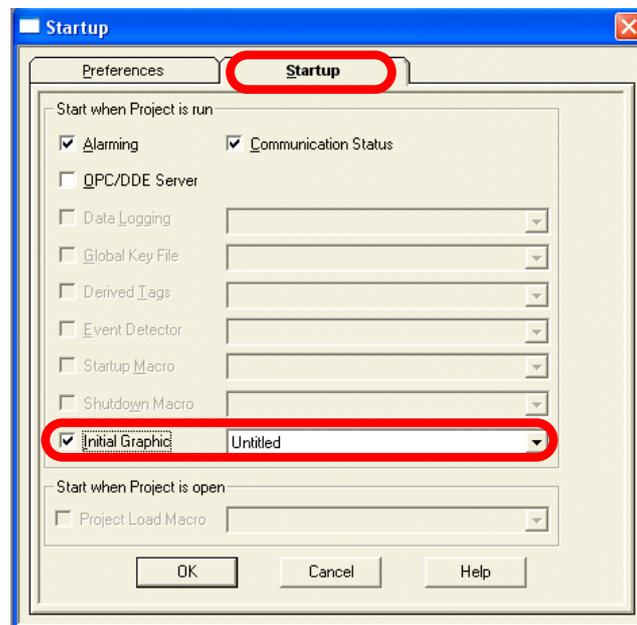


Figura. 4.46. Configurar la corrida del programa en RSVIEW32 (3)

4. Finalmente, para correr el programa presione la pestaña Run Mode, y finalmente Run Project.

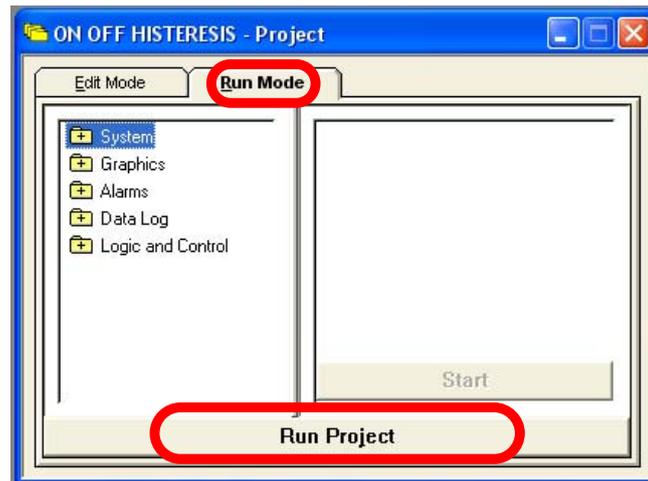


Figura. 4.47. Configurar la corrida del programa en RSView32 (4)

#### 4.10.4. PLANTEAMIENTO

Diseñar e Implementar un sistema de control de temperatura para un horno de secado de madera que trabaja en un rango de 50°C a 65°C. La técnica de control a utilizarse será la de ON-OFF con Histéresis. El control se lo hará a través de un PLC, al cual ingresa la señal proveniente de un sensor de temperatura que trabaja en un rango de 20°C a 70 °C, enviando una señal al PLC de 0V a 5 V en relación proporcional y lineal con la temperatura del horno. El controlador deberá trabajar en dos modos:

Manual: permite al usuario encender y apagar el horno, indistintamente de la temperatura.

Automático: El encendido y apagado del horno lo hace el controlador ON-OFF con Histéresis.

Adicionalmente con el software RSView32, crear una HMI que permita interactuar al operador con el sistema controlador, la cual deberá tener el siguiente diseño referencial:

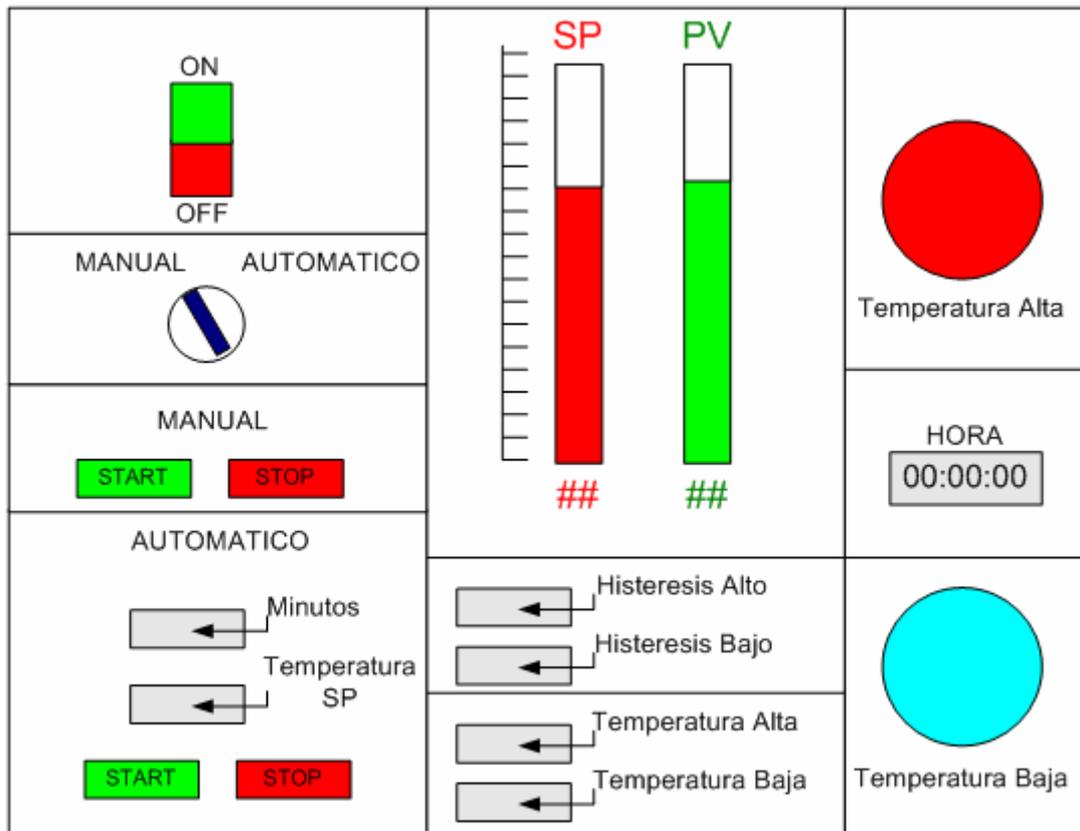


Figura. 4.48. Planteamiento Práctica #10

Nota: Las luces de alarma deberán estar asociadas a una archivo .wav para tener también alarma audible.

#### 4.10.5. PROCEDIMIENTO

Para la realización de esta práctica el procedimiento a seguir es el siguiente:

1. Configurar el driver en RSLinx
2. Ejecutar RSLogix500, crear un nuevo archivo y configurar el tipo de PLC y sus módulos (de ser el caso) en el programa.
3. Realizar el programa Ladder que cumpla con la lógica de control planteada en esta práctica.
4. Transferir el programa.
5. Ejecutar RSView32, configurar el canal de comunicación, configurar el nodo de comunicación.
6. Crear la interfase HMI que interactúe con el PLC
7. Conectar las entradas y salidas analógicas y digitales que se requieran del PLC.

8. Ejecutar el programa en el PLC y la HMI, probar si la lógica del programa del PLC y la HMI desarrollada funcionan correctamente, de acuerdo al problema planteado.

#### **4.10.6. PRUEBAS Y RESULTADOS**

#### **4.10.7. CONCLUSIONES Y RECOMENDACIONES**

## 4.11. CONTROL DE PROCESOS 2: CONTROL PID DE TEMPERATURA

### 4.11.1. OBJETIVOS

- Evaluar el desempeño de las técnicas PI y PID en el control de temperatura de un flujo de aire.
- Conocer el funcionamiento, configuración y aplicación de la Instrucción SCL
- Conocer el funcionamiento, configuración y aplicación de la Instrucción PID.
- Familiarizarse con el software RSView32.
- Crear una HMI (Interfase Humano Máquina) que interactúe con el PLC.

### 4.11.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 (5/03 o superior)
- ✓ 1 Módulo de Entradas Discretas.
- ✓ 1 Módulo de Salidas Discretas.
- ✓ 1 Módulo de Entradas Analógicas.
- ✓ 1 Módulo de Salidas Analógicas.
- ✓ 1 Computador Personal, con instalación de RSLogix500, RSLinx y RSView32.
- ✓ 1 Módulo Degem de simulación “AIR FLOW TEMPERATURE CONTROL SYSTEM PCT-2”.
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

### 4.11.3. MARCO TEÓRICO

#### Técnica de Control Proporcional Integral (PID)

Las tres componentes de un controlador PID son: parte **Proporcional**, acción **Integral** y acción **Derivativa**. El peso de la influencia que cada una de estas partes tiene en la suma final, viene dado por la *constante proporcional*, el *tiempo integral* y el *tiempo derivativo*, respectivamente. La formulación matemática de esta técnica de control viene dada por:

$$m(t) = m_0 + K_c e(t) + \left( \frac{K_c}{T_i} \right) \int e(t) dt + K_c T_D \left( \frac{\partial e(t)}{\partial t} \right)$$

Donde:

$m(t) = \text{Salida del controlador}$

$e(t) = \text{Señal de error} \Rightarrow \text{Punto de control} - \text{Señal que se controla}$

$K_c = \text{Ganancia del controlador}$

$m_0 = \text{Valor base (Bias)}$

$T_i = \text{Tiempo de integración o reajuste (Reset Time)}$

$T_D = \text{Rapidez de derivación (minutos)}$

En términos de control la contribución de cada componente es la siguiente:

- **Proporcional:** Esta componente PID toma un papel importante cuando la señal de *error* es grande, pero su acción se ve mermada con la disminución de dicha señal. Este efecto tiene como consecuencia la aparición de un *error permanente*, que hace que la parte proporcional nunca llegue a solucionar por completo el error del sistema. La parte proporcional no considera el tiempo, por tanto la mejor manera de solucionar el *error permanente* y hacer que el sistema contenga alguna componente que tenga en cuenta la variación con respecto al tiempo es incluyendo y configurando las acciones integral y derivativa.
- **Integral:** tiene como propósito disminuir y eliminar el error en estado estacionario, provocado por el modo proporcional.
- **Derivativo:** La acción derivativa se manifiesta cuando hay un cambio en el valor absoluto del error; (si el error es constante, solamente actúan los modos proporcional e integral). La función de la acción derivativa es mantener el error al mínimo corrigiéndolo proporcionalmente con la velocidad con que se produce; de esta manera evita que el error se incremente.

## Instrucción Proporcional Integral Derivativa (PID)

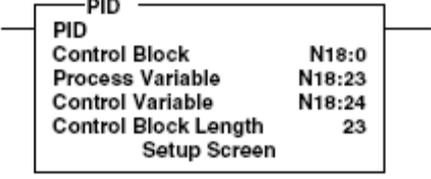
PID		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
	Proporcional Integral Derivative (PID)	Ésta es una instrucción de salida que controla propiedades físicas tales como temperatura, presión, nivel de líquidos, flujo, etc.

Tabla. 4.38. Instrucción PID



Coloque la instrucción PID sin ninguna condición lógica que la preceda, puesto que si una instrucción PID se hace falsa, el término integral se elimina.

El esquema de un controlador PID aplicado al control de nivel de un líquido, está representado en la figura 4.49.

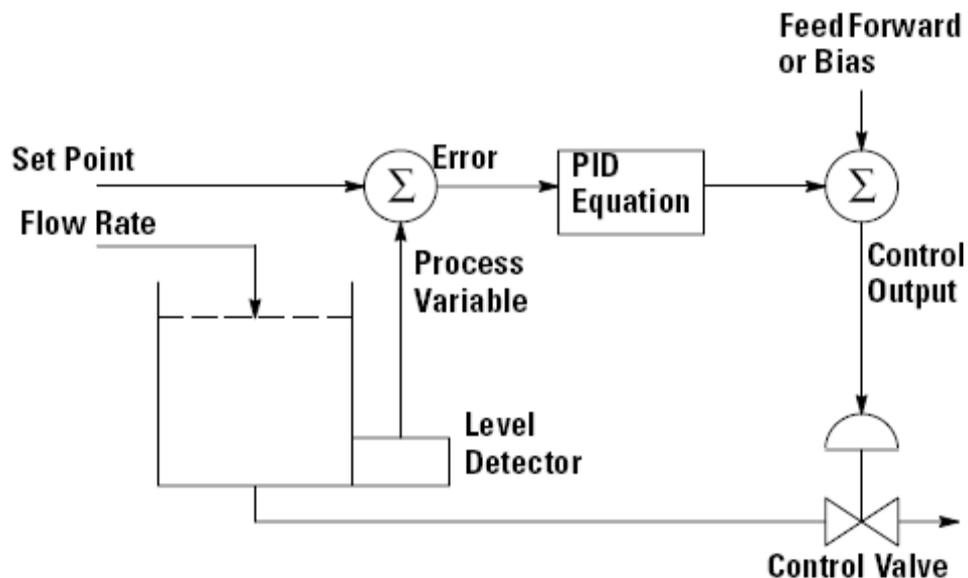


Figura. 4.49. Esquema de un controlador PID aplicado al control de nivel de un líquido

En donde la Ecuación PID controla el proceso enviando una señal de salida a la Válvula de Control. Mientras mayor sea el error entre el Setpoint y la entrada del la Variable del Proceso, mayor será la señal de salida y viceversa. Un valor adicional (Feed Forward o Bias) puede ser sumado a la salida de control como compensación.

### Ecuación PID

La instrucción PID utiliza el siguiente algoritmo:

$$Output = K_c \left[ (E) + \frac{1}{T_i} \int (E) dt + T_D \cdot \frac{D(PV)}{df} \right] + Feed Forward / Bias$$

El rango de las constantes de ganancia estándar se muestra en la siguiente tabla:

Término	Rango	Bit RG	Referencia
Ganancia del Controlador $K_c$	0.1 a 3276.7 (adimensional)	RG = 0	Proporcional
	0.01 a 327.67 (adimensional)	RG = 1	
Reset Time $T_i$	3276.7 a 0.1 (min./ repetición)	RG = 0	Integral
	327.67 a 0.01 (min./ repetición)	RG = 1	
Rapidez de desviación $T_D$	0.01 a 327.67 (minutos)	RG = 0 ó 1	Derivativo

Tabla. 4.38. Ganancias Estándar PID

**\*Reset and Gain Enhancement Bit (RG):** Cuando es 1, el valor  $T_i$  y la ganancia  $K_c$  se dividen para 100. Cuando es 0, el valor  $T_i$  y la ganancia  $K_c$  se dividen para 10.



*La instrucción PID es de tipo “Entero”, es decir no permite ingresar valores de punto flotante en ninguno de sus parámetros. Si se mueven valores de punto flotante a cualquiera de sus parámetros, se produce una conversión de punto flotante a entero. Si se desea ingresar una ganancia de 25.4, se debe ingresar el valor entero de 254 si el bit  $RG=0$ , y 2540 si  $RG=1$ .*

### PARÁMETROS

- **Control Block:** Es un archivo que almacena los datos requeridos para que la instrucción pueda operar. Éste archivo tiene una longitud de 23 palabras y pueden ser ingresadas como direcciones de archivos tipo entero. Por ejemplo una entrada de N10:0 asignará los elementos N10:0 hasta N10:22.

La ventana de configuración del Control Block se muestra en la figura 4.50.

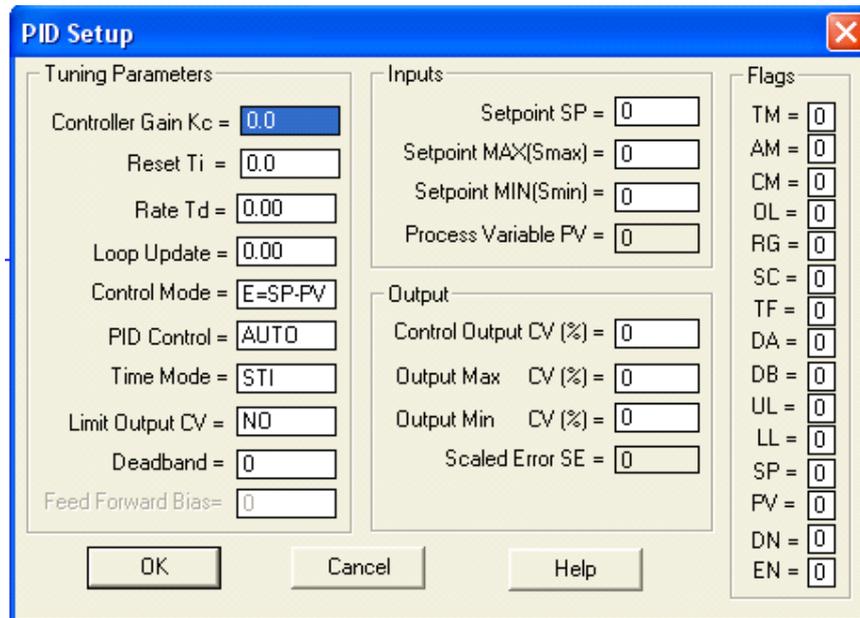


Figura. 4.50. Ventana de configuración del bloque de control PID

Para desplegar ésta ventana se debe dar doble clic en *Setup Screen* como muestra en la figura 4.51.

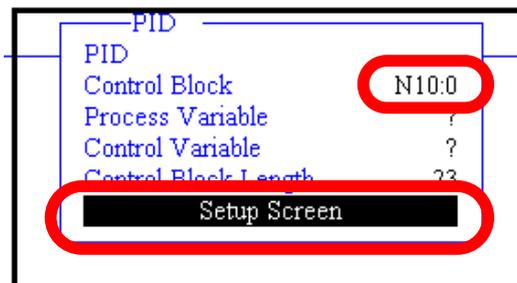


Figura. 4.51. Desplegar la ventana de configuración de la instrucción PID



Utilice un archivo de datos UNICO para el Control Block de su PID, es decir no utilice la dirección N7:0 sino cree un nuevo Data File de tipo entero. Por ejemplo N10:0 (ver figura 4.51.)

- **Process Variable PV:** es la dirección de un elemento que almacena el valor de entrada del proceso. Esta dirección puede ser la de una palabra en donde se almacene el valor de una entrada analógica ó un entero que contenga el valor pre-escalado (0 a 16383) de la entrada analógica.

- **Control Variable CV:** Es la dirección de un elemento que almacena la salida (rango de 0 a 16383) de la instrucción PID. Normalmente suele ser un valor entero para que usted pueda escalar esta salida al rango que requiera su aplicación y enviarlo a un módulo de salida analógica.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Word 0	EN		DN	PV	SP	LL	UL	DB	DA	TF	SC	RG	OL	CM
Word 1	PID Sub Error Code (MSB)													
Word 2	Setpoint SP													
Word 3	Gain Kc													
Word 4	Reset Ti													
Word 5	Rate Td													
Word 6	Feed Forward Bias													
Word 7	Setpoint Maximum (Smax)													
Word 8	Setpoint Minimum (Smin)													
Word 9	Deadband													
Word 10	USO INTERNO – NO CAMBIAR!!													
Word 11	Output Max													
Word 12	Output Min													
Word 13	Loop Update													
Word 14	Scaled Process Variable													
Word 15	Scaled Error SE													
Word 16	Output CV% (0-100%)													
Word 17	MSW Integral Sum													
Word 18	LSW Integral Sum													
Word 19	USO INTERNO – NO CAMBIAR!!													
Word 20	USO INTERNO – NO CAMBIAR!!													
Word 21	USO INTERNO – NO CAMBIAR!!													
Word 22	USO INTERNO – NO CAMBIAR!!													

Tabla. 4.40. Control Block PID

**Ganancia “Controller Gain” (Kc)**

Kc (palabra 3) es la ganancia “Proporcional”, cuyo rango va desde 0 a 3276.7

(cuando RG = 0), o de 0 a 327.67 (cuando RG = 1).

**Reset Time (Ti)**

Ti (palabra 4) es la ganancia “Integral”, cuyo rango va desde 0 to 3276.7 (cuando RG = 0), o de 0 a 327.67 (cuando RG = 1) minutos/repetición.

**Rate Time (Td)**

Td (palabra 5) es la rapidez de derivación. Su rango es de 0 a 327.67 minutos.

**Feed Forward/Bias**

El valor que se escriba en esta palabra (palabra #6) es sumado a la salida. Se puede escribir un valor entero entre -16383 y +16383.

**Error Escalado E**

Es la diferencia entre la variable de procesos y el setpoint, es decir  $E = SP - PV$  ó  $E = PV - SP$  dependiendo del bit de modo de control CM.

**Control (CM)**

Cuando es 1,  $E = PV - SP$  lo que causa que la variable de control se incremente cuando la variable del proceso es mayor al setpoint.

Cuando es 0,  $E = SP - PV$  lo que causa que la variable de control se reduzca cuando la variable del proceso es mayor al setpoint.

**Auto / Manual (AM)**

Cuando el bit auto/manual es (0), este provoca que el controlador opere en modo automático. Cuando es (1) el controlador opera en modo manual. En modo automático la instrucción controla la variable de control (CV). En modo manual el programador controla la variable de control (CV).

### Parámetros de Entrada

Parámetro	Dirección	Rango	Tipo	Tipo de acceso
SP - Setpoint	Palabra #2	0 a 16383*	Control	Lectura/escritura
SPV – Variable de proceso escalada	Palabra #14	0 a 16383	Status	Lectura
SMAX – Setpoint máximo	Palabra #7	-32767 a +32767.	Control	Lectura/escritura
SMIN – Setpoint Mínimo	Palabra #8	-32767 a +32767	Control	Lectura/escritura

Tabla. 4.41. Parámetros de Entrada PID

### Parámetros de Salida

Parámetro	Dirección	Rango	Tipo	Tipo de acceso
CV - Variable de Control	Def. usuario	0 a 16383	Control	Lectura/escritura
CV% - Porcentaje de salida	Palabra #16	0 a 100	Control	Lectura
OL – Limitar salida	Palabra #0 bit 3	0 ó 1	Control	Lectura/escritura
CVH – Salida Máxima	Palabra #11	0 a 100%	Control	Lectura/escritura
CVL – Salida Mínima	Palabra #12	0 a 100%	Control	Lectura/escritura

Tabla. 4.42. Parámetros de salida PID

### Escalamiento de Entradas y Salidas

Para la instrucción PID, la escala numérica de la Variable de Proceso (PV) y la de la Variable de Control (CV) es de 0 a 16383. Por lo que, para poder utilizar unidades de ingeniería, se debe primero escalar los rangos de las entradas y salidas analógicas a la escala de 0 a 16383.

Ejemplos:

- Una entrada analógica con un rango de 4 a 20mA tiene un rango decimal de 3277 a 16384. Este rango decimal debe ser escalado a un rango de 0 a 16383 para poder ser utilizado como PV.

- Para escalar una salida a un rango de 4 a 20 mA cuyo valor decimal sería de 6242 a 31208. En este caso el rango de 0 a 16383 de CV debe ser escalado al rango de 6242 a 31208.

Para realizar este escalamiento, utilice la instrucción SCL, la cual se describe a continuación:

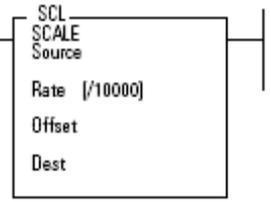
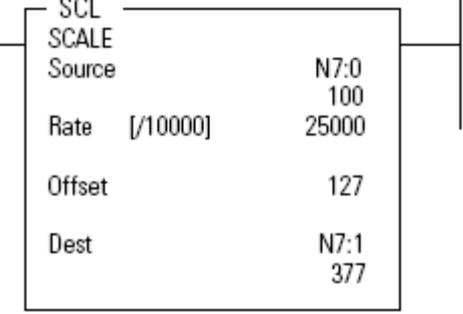
SCL		
SÍMBOLO	NOMBRE	DESCRIPCIÓN
	Scale Data (SCL)	Cuando esta instrucción es verdadera, el valor de la dirección origen (Source) es multiplicado por el valor de (Rate), este resultado es redondeado y sumado al valor de compensación (Offset), para finalmente ser colocado en el destino (Dest).

Tabla. 4.43. Explicación Instrucción SCL

	<p>Por ejemplo si se tiene la instrucción SCL con los parámetros indicados en la figura, ésta realiza las siguientes operaciones:</p> <p>La fuente (source) 100 es multiplicada por 25000 y dividida para 10000, y sumada a 127. El resultado 377, es colocado en el destino.</p>
-------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Para calcular los valores de *Offset* y *Rate(Slope)* necesarios para escalar a un rango definido, se deben aplicar las siguientes fórmulas:

$$\text{Valor Escalado} = (\text{valor de entrada} \times \text{Slope}) + \text{offset}$$

$$\text{Slope} = \frac{(\text{Escalado MAX} - \text{Escalado MIN})}{(\text{Entrada MAX} - \text{Entrada MIN})}$$

$$\text{Offset} = \text{Escalado MIN} - (\text{Entrada MIN} \times \text{Slope})$$

#### 4.11.4. PLANTEAMIENTO

Diseñar e Implementar un sistema de control de temperatura para una estufa bacteriológica que trabaja en un rango de 50°C a 60°C. La técnica de control a utilizarse será la de *PI* ó *PID* (evaluar el desempeño de cada técnica). El control se lo hará a través de un PLC, al cual ingresa la señal proveniente de un sensor de temperatura que trabaja en un rango de 20°C a 70 °C, enviando una señal al PLC de 0V a 5V en relación lineal y proporcional con la temperatura de la estufa. El error en estado estacionario máximo permitido entre la temperatura real (PV) y la deseada (Setpoint) es de +/- 0.1°C.

Adicionalmente con el software RSView32, crear una HMI que permita interactuar al operador con el sistema controlador, la cual deberá tener las siguientes características:

Dos cuentas de usuario: INGENIERO (con contraseña) y OPERADOR

Pantalla Principal con el siguiente diseño referencial:

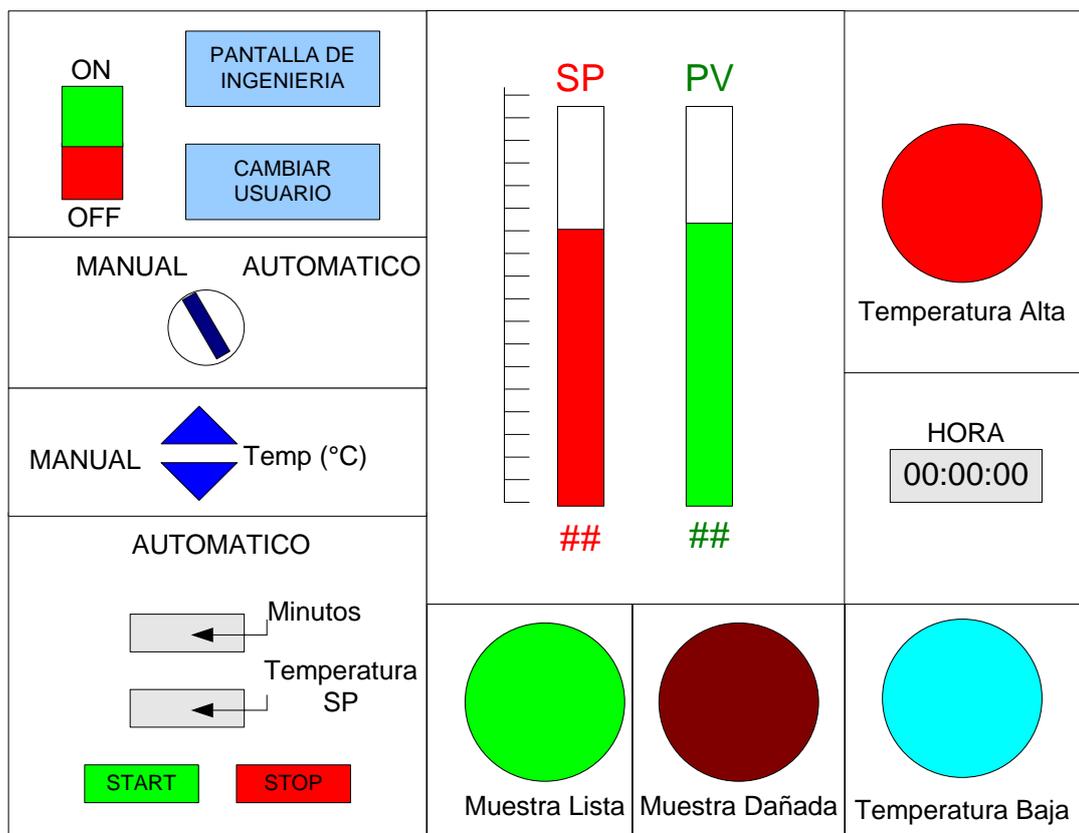


Figura. 4.51. Planteamiento Práctica #11

- ✓ El botón de “Pantalla de Ingeniería” despliega dicha pantalla únicamente si se encuentra trabajando con el usuario “INGENIERO”
- ✓ Las luces de alarma deberán estar asociadas a una archivo .wav para tener también alarma audible.
- ✓ Una vez transcurrido el tiempo ingresado, se deberá activar la alarma de “Muestra Lista”
- ✓ Si la variación entre la temperatura real (PV) y la deseada (Setpoint) es mayor a  $\pm 0.1^{\circ}\text{C}$  durante el transcurso del tiempo, se deberá activar la alarma “Muestra Dañada”.
- ✓ El botón START, en modo automático arrancará el tiempo y deberá ser presionado una vez que el controlador ya se haya estabilizado.

Pantalla de Ingeniería (Acceso permitido únicamente para el usuario INGENIERO)

- ✓ Control de activación del controlador.
- ✓ Visualización de estado de las luces de alarma.
- ✓ Selección de la Temperatura de referencia o Set Point.
- ✓ Ingreso de parámetros de calibración del controlador.
- ✓ Gráfica de tendencia (Trend) de Temperatura de la Estufa y Setpoint.
- ✓ Gráfica de tendencia (Trend) de % de Voltaje de Salida del controlador.

#### 4.11.5. PROCEDIMIENTO

Para la realización de esta práctica el procedimiento a seguir es el siguiente:

1. Configurar el driver en RSLinx
2. Ejecutar RSLogix500, crear un nuevo archivo y configurar el tipo de PLC y sus módulos (de ser el caso) en el programa.
3. Realizar el programa Ladder que cumpla con la lógica de control planteada en esta práctica.
4. Transferir el programa.
5. Ejecutar RSView32, configurar el canal de comunicación, configurar el nodo de comunicación.
6. Crear la interfase HMI que interactúe con el PLC

7. Conectar las entradas y salidas analógicas y digitales que se requieran del PLC.
8. Sintonizar el controlador.
9. Probar si la lógica del programa del PLC y la HMI desarrollada funcionan correctamente, de acuerdo al problema planteado.

#### **4.11.6. PRUEBAS Y RESULTADOS**

#### **4.11.7. CONCLUSIONES Y RECOMENDACIONES**

## 4.12. CONTROL DE PROCESOS 3: CONTROL PID DE VELOCIDAD

### 4.12.1. OBJETIVOS

- Evaluar el desempeño de las técnicas P, PI y PID en el control de velocidad de un motor DC
- Conocer el funcionamiento, configuración y aplicación de la Instrucción SCL
- Conocer el funcionamiento, configuración y aplicación de la Instrucción PID.
- Familiarizarse con el software RSView32.
- Crear una HMI (Interfase Humano Máquina) que interactúe con el PLC.

### 4.12.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 (5/03 o superior)
- ✓ 1 Módulo de Entradas Discretas.
- ✓ 1 Módulo de Salidas Discretas.
- ✓ 1 Módulo de Entradas Analógicas.
- ✓ 1 Módulo de Salidas Analógicas.
- ✓ 1 Computador Personal, con instalación de RSLogix500, RSLinx y RSView32.
- ✓ 1 Módulo Degem de simulación PCT-1 “DC SPEED CONTROL SYSTEM”.
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

### 4.12.3. MARCO TEÓRICO

Referirse al marco teórico de la práctica anterior

### 4.12.4. PLANTEAMIENTO

Diseñar e Implementar un sistema de control de velocidad para una bomba, que trabaja en el rango de 0 - 3000 RPM (0-10VDC). La técnica de control a utilizarse será la de *PI* ó *PID* (evaluar el desempeño de cada técnica). El control se lo hará a través de un PLC, al cual ingresa la señal proveniente de un sensor el cual envía una señal al PLC de 0V a 6VDC en relación lineal a la velocidad de la bomba. El error en estado estacionario máximo permitido es de +/- 20 RPM.

Adicionalmente con el software RSVIEW32, crear una HMI que permita interactuar al operador con el sistema controlador, la cual deberá tener las siguientes características:

Dos cuentas de usuario: INGENIERO (con contraseña) y OPERADOR

Pantalla Principal con el siguiente diseño referencial:

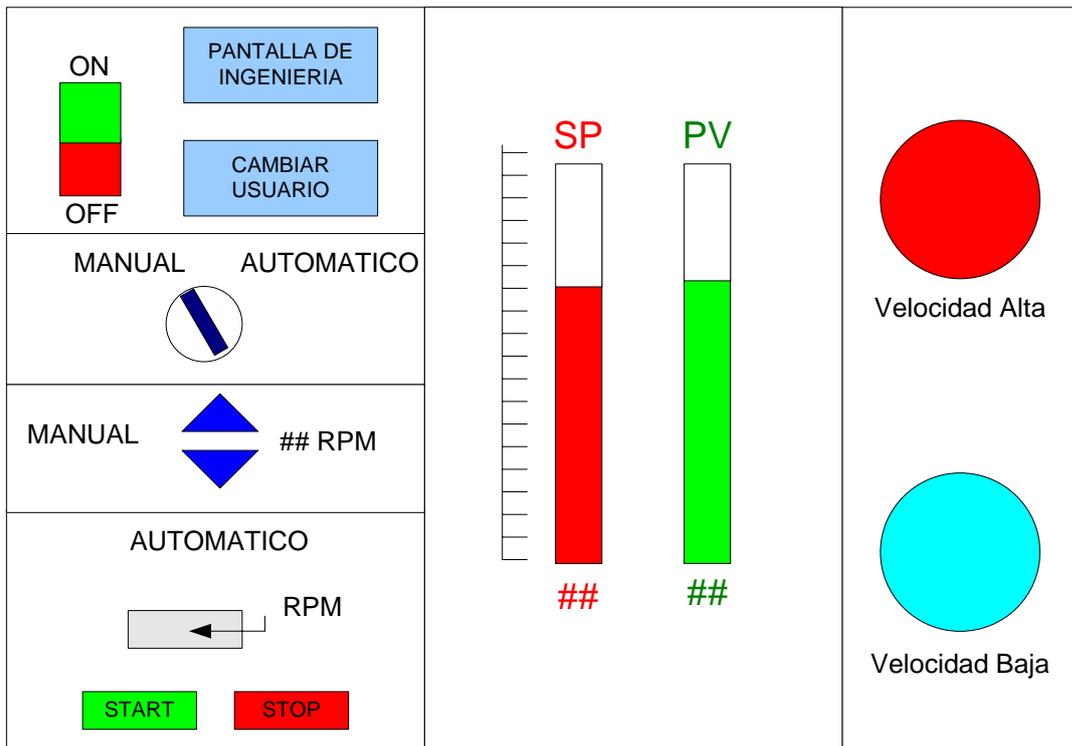


Figura. 4.52. Planteamiento Práctica #12

- ✓ El botón de “Pantalla de Ingeniería” despliega dicha pantalla únicamente si se encuentra trabajando con el usuario “INGENIERO”
- ✓ Las luces de alarma deberán estar asociadas a un archivo .wav para tener también alarma audible.

Pantalla de Ingeniería (Acceso permitido únicamente para el usuario INGENIERO)

- ✓ Control de activación del controlador.
- ✓ Visualización de estado de las luces de alarma.
- ✓ Selección de la Velocidad de referencia o Set Point.
- ✓ Ingreso de parámetros de calibración del controlador.
- ✓ Gráfica de tendencia (Trend) de velocidad de la Bomba y Setpoint.

- ✓ Gráfica de tendencia (Trend) de Voltaje de Salida del controlador.

#### **4.12.5. PROCEDIMIENTO**

Para la realización de esta práctica el procedimiento a seguir es el siguiente:

1. Configurar el driver en RSLinx
2. Ejecutar RSLogix500, crear un nuevo archivo y configurar el tipo de PLC y sus módulos (de ser el caso) en el programa.
3. Realizar el programa Ladder que cumpla con la lógica de control planteada en esta práctica.
4. Transferir el programa.
5. Ejecutar RSView32, configurar el canal de comunicación, configurar el nodo de comunicación.
6. Crear la interfase HMI que interactúe con el PLC
7. Conectar las entradas y salidas analógicas y digitales que se requieran del PLC.
8. Sintonizar el controlador.
9. Probar si la lógica del programa del PLC y la HMI desarrollada funcionan correctamente, de acuerdo al problema planteado.

#### **4.12.6. PRUEBAS Y RESULTADOS**

#### **4.12.7. CONCLUSIONES Y RECOMENDACIONES**

### 4.13. CONTROL DE PROCESOS 4: CONTROL DE RAZÓN

#### 4.13.1. OBJETIVOS

- Familiarizarse con la técnica de Control de Razón.
- Crear una HMI (Interfase Humano Máquina) que interactúe con el PLC.

#### 4.13.2. MATERIALES Y EQUIPOS

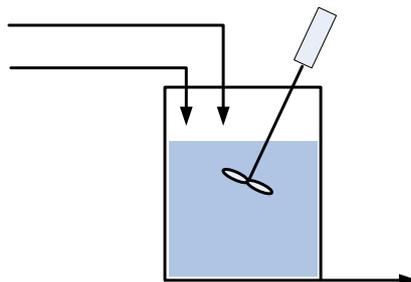
- ✓ 1 CPU Allen Bradley de la Serie SLC 500 (5/03 o superior)
- ✓ 1 Módulo de Entradas Discretas.
- ✓ 1 Módulo de Salidas Discretas.
- ✓ 1 Módulo de Entradas Analógicas.
- ✓ 1 Módulo de Salidas Analógicas.
- ✓ 1 Computador Personal, con instalación de RSLogix500, RSLinx y RSView32.
- ✓ 2 Módulos Degem de simulación PCT-1 “DC SPEED CONTROL SYSTEM”.
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

#### 4.13.3. MARCO TEÓRICO

##### Control de Razón\*

Esta es una técnica de control muy común en procesos industriales. Para ejemplificar este método de control, suponga que se debe mezclar dos corrientes de líquidos A y B, en cierta proporción o razón R, en donde

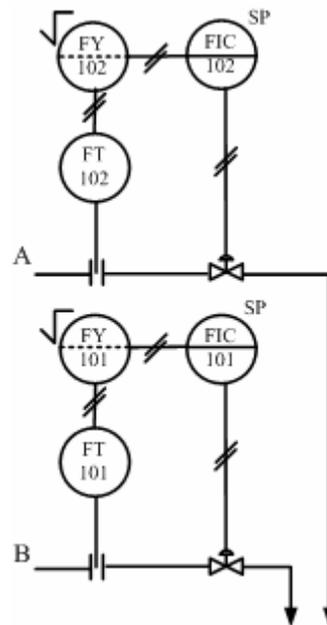
$$R = \frac{B}{A}$$



\*La información expuesta de este tema fue extraída del texto 1 citado en las referencias bibliográficas.

**Figura. 4.53. Aplicación de control de Razón**

La figura 4.54. muestra una posible solución para cumplir con este requerimiento, esta consiste en controlar cada flujo mediante un circuito de flujo en el cual el punto de control de los controladores se fija de manera tal que los líquidos se mezclan en la proporción correcta.

**Figura. 4.54. Diagrama de control de Razón**

Sin embargo, si no se puede controlar el flujo del líquido A, sino únicamente medirlo, flujo que se conoce como *flujo libre*. De alguna manera la corriente B debe variar conforme varía la corriente A, es decir se debe controlar la corriente B para mantener la mezcla en la razón correcta.

Existen dos maneras de realizar el control de razón:

1. **Primer Método:** Consiste en medir el flujo libre y multiplicarlo por la razón que se desea ( $B=RA$ ), para obtener el flujo que se requiere de la corriente B (Setpoint del controlador de la corriente B).

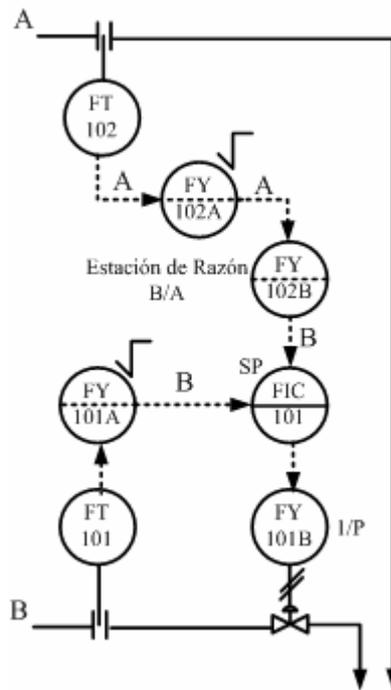


Figura. 4.55. Primer método de control de Razón

2. **Segundo Método:** Consiste en medir ambas corrientes y dividir las para obtener la razón de flujo real a través del sistema. Esta razón que se calcula se envía al controlador de la corriente B, el cual variará el flujo B para mantener esta razón en un valor deseado o Setpoint ingresado en el controlador.

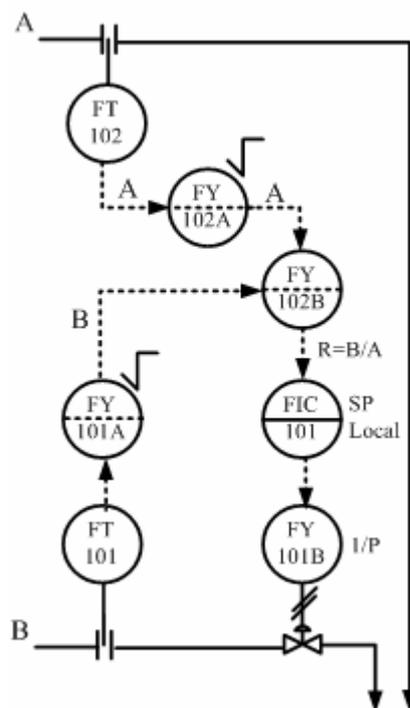


Figura. 4.56. Segundo método de control de Razón

En la industria se utilizan ambos esquemas de control, sin embargo, se prefiere el primer método porque es más lineal.

#### 4.13.4. PLANTEAMIENTO

En una planta de Pintura, se desea controlar la razón de mezcla de dos diferentes tipos de pintura, para lo cual se debe controlar la velocidad (RPM) de dos bombas que extraen la pintura de los tanques de almacenamiento. Empleando la técnica de Control de Razón, diseñar e implementar un sistema de control de velocidad para ambas bombas que permita ingresar la razón de mezcla deseada. En ambas bombas se cuenta con sensores de Velocidad que envían al controlador una señal de 0 a 6 VDC en relación lineal a la velocidad de la bomba (0 a 3000 RPM).

Adicionalmente con el software RSView32, crear una HMI que permita interactuar al operador con el sistema controlador, la cual deberá tener las siguientes características:

Dos cuentas de usuario: INGENIERO (con contraseña) y OPERADOR

Pantalla Principal con el siguiente esquema referencial:

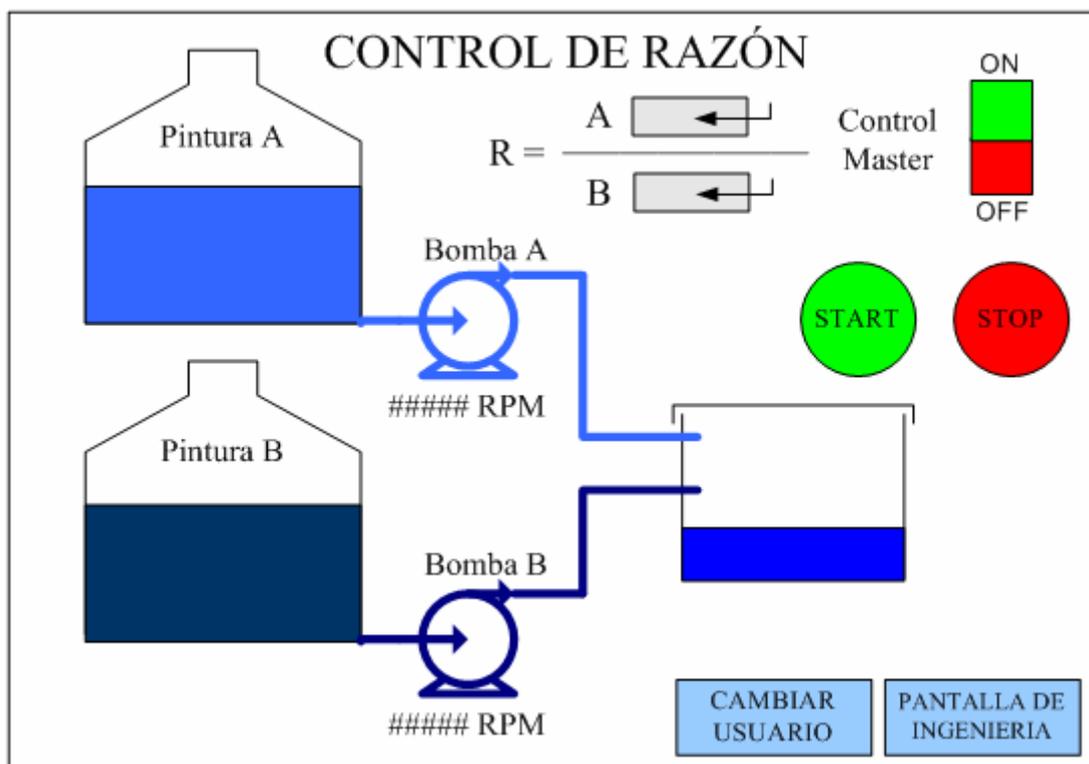


Figura. 4.57. Planteamiento Práctica #13

- ✓ El botón de “Pantalla de Ingeniería” despliega dicha pantalla únicamente si se encuentra trabajando con el usuario “INGENIERO”

Pantalla de Ingeniería (Acceso permitido únicamente para el usuario INGENIERO)

- ✓ Control de activación del controlador.
- ✓ Selección de la Razón De Mezcla.
- ✓ Ingreso de parámetros de calibración del controlador.
- ✓ Gráfica de tendencia (Trend) de Velocidad de la Bomba 1 y Bomba 2.
- ✓ Gráfica de tendencia (Trend) de Voltaje de Salida del controlador.

#### **4.13.5. PROCEDIMIENTO**

Para la realización de esta práctica el procedimiento a seguir es el siguiente:

1. Configurar el driver en RSLinx
2. Ejecutar RSLogix500, crear un nuevo archivo y configurar el tipo de PLC y sus módulos (de ser el caso) en el programa.
3. Realizar el programa Ladder que cumpla con la lógica de control planteada en esta práctica.
4. Transferir el programa.
5. Ejecutar RSView32, configurar el canal de comunicación, configurar el nodo de comunicación.
6. Crear la interfase HMI que interactúe con el PLC
7. Conectar las entradas y salidas analógicas y digitales que se requieran del PLC.
8. Sintonizar los controladores.
9. Probar si la lógica del programa del PLC y la HMI desarrollada funcionan correctamente, de acuerdo al problema planteado.

#### **4.13.6. PRUEBAS Y RESULTADOS**

#### **4.13.7. CONCLUSIONES Y RECOMENDACIONES**

## 4.14. CONTROL DE PROCESOS 5: CONTROL POR SOBREPOSICIÓN

### 4.14.1. OBJETIVOS

- Familiarizarse con la técnica de Control por Sobreposición.
- Crear una HMI (Interfase Humano Máquina) que interactúe con el PLC.

### 4.14.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 (5/03 o superior)
- ✓ 1 Módulo de Entradas Discretas.
- ✓ 1 Módulo de Salidas Discretas.
- ✓ 1 Módulo de Entradas Analógicas.
- ✓ 1 Módulo de Salidas Analógicas.
- ✓ 1 Computador Personal, con instalación de RSLogix500, RSLinx y RSView32.
- ✓ 1 Módulo Degem de simulación PCT-2 “AIR FLOW TEMPERATURE CONTROL SYSTEM”.
- ✓ 1 Módulo Degem de simulación PCT-1 “DC SPEED CONTROL SYSTEM”
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

### 4.14.3. MARCO TEÓRICO

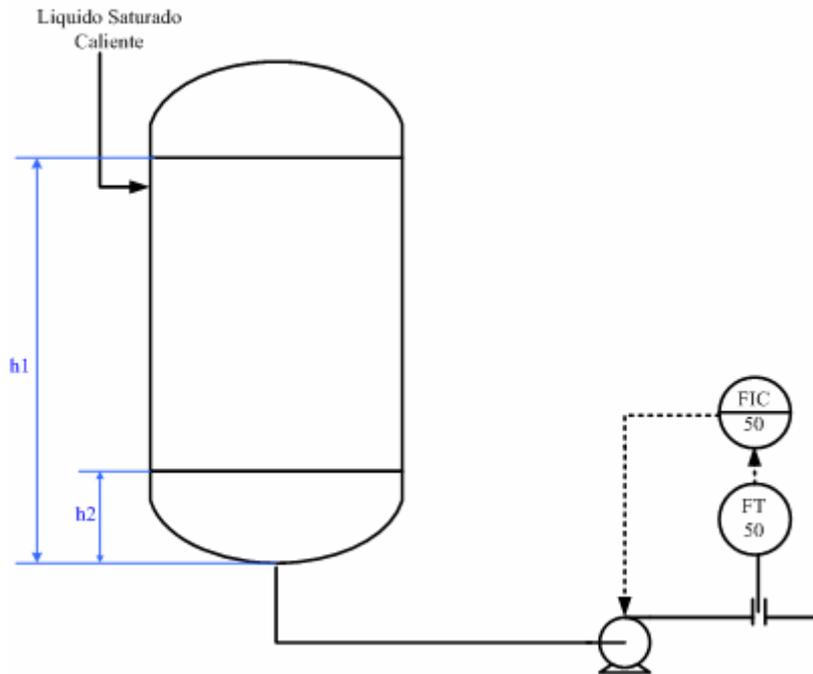
#### Control por Sobreposición\*

Esta técnica de control se utiliza generalmente como un control de protección para mantener variables del proceso dentro de los límites.

Para ejemplificar este método de control, considérese el proceso que aparece en la figura 4.58., en el tanque entra un líquido saturado y de ahí nuevamente se bombea, bajo control, al proceso.

---

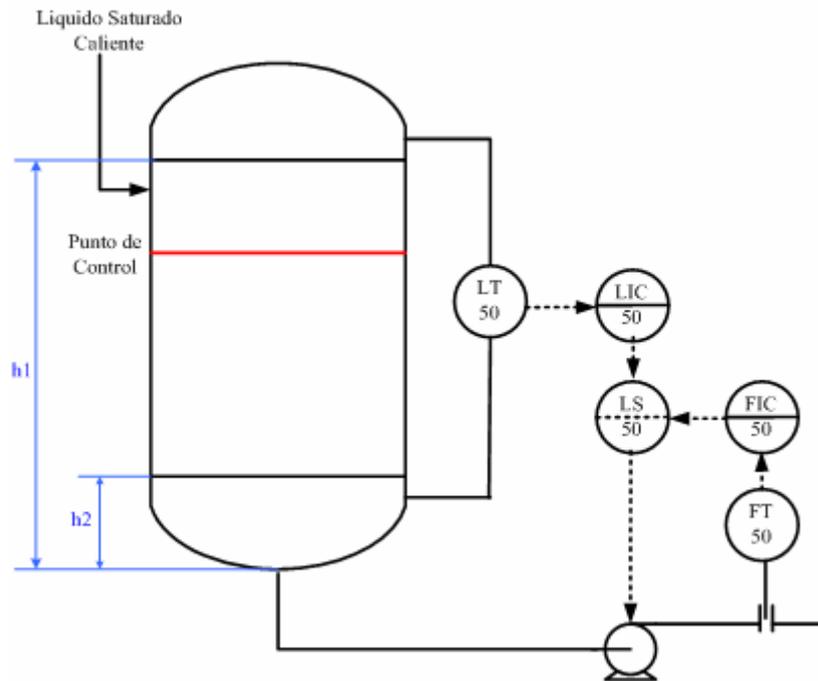
\*La información expuesta de este tema fue extraída del texto 1 citado en las referencias bibliográficas.



**Figura. 4.58. Aplicación de control por Sobreposición**

En operación normal, el nivel del tanque está a la altura  $h_1$ , si por cualquier circunstancia el nivel del líquido baja a  $h_2$ , no se tendrá suficiente volumen positivo neto de succión (VPNS), de lo que resulta cavitación en la bomba. Para evitar esto, es necesario diseñar un sistema de control con el que se evite esta condición.

Para solucionar esta situación, se debe implementar un controlador de nivel, el cual también controle el bombeo de la bomba pero en relación al nivel del líquido y no en relación al flujo deseado para el proceso. Adicionalmente se debe utilizar un relé de selección baja, el cual recibirá la señal de ambos controladores y enviará a la bomba la señal mas baja. Este sistema de control se representa gráficamente en la figura 4.59.



**Figura. 4.59. Diagrama de control por Sobreposición**

Es importante acotar que el controlador de flujo debe ser un controlador con acción inversa, en cambio el controlador de nivel debe ser un controlador de acción directa.

Bajo condiciones normales de operación, el nivel está en  $h_1$ , el cual se encuentra por arriba del punto de control del controlador de nivel y por tanto, desde este controlador se tratará de acelerar la bomba tanto como sea posible incrementando al máximo la salida, es decir en este caso, el relé enviará a la bomba la señal del controlador de flujo puesto que este tendrá un salida menor.

Si por alguna razón el nivel del tanque empieza a bajar, tan pronto como el nivel sea menor que el punto de control del controlador de nivel, este controlador tratará de hacer más lento el bombeo mediante la reducción de su salida. Cuando esta salida sea menor a la salida del controlador de flujo, en el relé de selección de baja se elige la salida del controlador de nivel para manejar la bomba.

Una consideración importante al diseñar un sistema de control por Sobreposición es que, si en cualquiera de los controladores existe modo integral de control, se requerirá entonces una protección contra reajuste excesivo.



Dos cuentas de usuario: INGENIERO (con contraseña) y OPERADOR

Pantalla Principal con el siguiente diseño referencial:

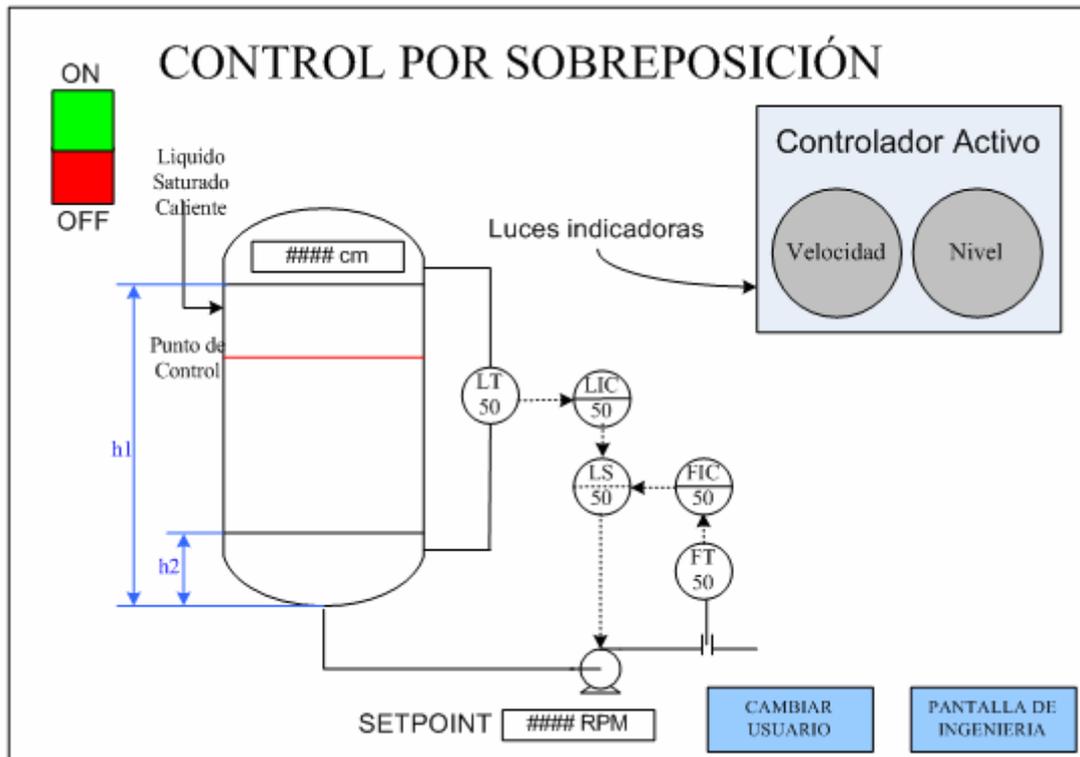


Figura. 4.61. Planteamiento Práctica #14 (2)

- ✓ El botón de “Pantalla de Ingeniería” despliega dicha pantalla únicamente si se encuentra trabajando con el usuario “INGENIERO”

Pantalla de Ingeniería (Acceso permitido únicamente para el usuario INGENIERO)

- ✓ Control Master de activación del controlador.
- ✓ Selección de la Velocidad deseada.
- ✓ Ingreso de los niveles de fluido  $h_1$  y  $h_2$
- ✓ Ingreso de parámetros de calibración del sistema controlador.
- ✓ Gráfica de tendencia (Trend) de Velocidad de la Bomba real (PV) y deseada (Setpoint).
- ✓ Gráfica de tendencia (Trend) de Nivel del liquido real (PV) y deseado (Setpoint).
- ✓ Gráfica de tendencia (Trend) de Voltaje de Salida de los controladores.

#### **4.14.5. PROCEDIMIENTO**

Para la realización de esta práctica el procedimiento a seguir es el siguiente:

1. Configurar el driver en RSLinx
2. Ejecutar RSLogix500, crear un nuevo archivo y configurar el tipo de PLC y sus módulos (de ser el caso) en el programa.
3. Realizar el programa Ladder que cumpla con la lógica de control planteada en esta práctica.
4. Transferir el programa.
5. Ejecutar RSView32, configurar el canal de comunicación, configurar el nodo de comunicación.
6. Crear la interfase HMI que interactúe con el PLC
7. Conectar las entradas y salidas analógicas y digitales que se requieran del PLC.
8. Sintonizar los controladores.
9. Probar si la lógica del programa del PLC y la HMI desarrollada funcionan correctamente, de acuerdo al problema planteado.

#### **4.14.6. PRUEBAS Y RESULTADOS**

#### **4.14.7. CONCLUSIONES Y RECOMENDACIONES**

## 4.15. CONTROL DE PROCESOS 6: CONTROL EN CASCADA

### 4.15.1. OBJETIVOS

- Familiarizarse con la técnica de Control en Cascada.
- Crear una HMI (Interfase Humano Máquina) que interactúe con el PLC.

### 4.15.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 (5/03 o superior)
- ✓ 1 Módulo de Entradas Discretas.
- ✓ 1 Módulo de Salidas Discretas.
- ✓ 1 Módulo de Entradas Analógicas.
- ✓ 1 Módulo de Salidas Analógicas.
- ✓ 1 Computador Personal, con instalación de RSLogix500, RSLinx y RSView32.
- ✓ 1 Módulo Degem de simulación PCT-2 “AIR FLOW TEMPERATURE CONTROL SYSTEM”.
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

### 4.15.3. MARCO TEÓRICO

#### Control en Cascada\*

Existen algunas ocasiones en que el desempeño de un esquema de control feedback puede mejorarse notablemente mediante el empleo de un esquema de control denominado “en cascada”. La idea básica detrás del empleo de controladores en cascada la podemos explicar mejor empleando el siguiente ejemplo.

Considérese el sistema de control para el intercambiador de calor que aparece en la figura 4.62. en este sistema la temperatura con que sale el líquido que se procesa se controla mediante la manipulación de la posición de la válvula de vapor.

---

\*La información expuesta de este tema fue extraída del texto 1 citado en las referencias bibliográficas.

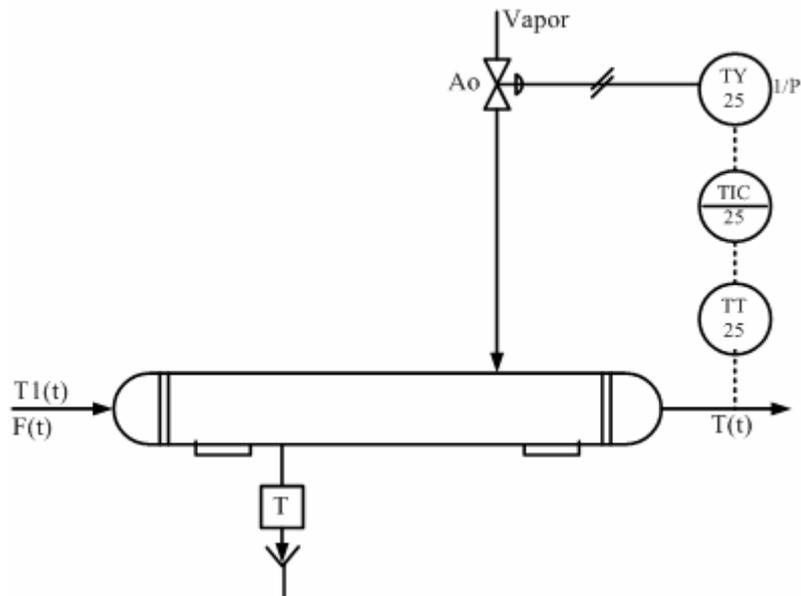


Figura. 4.62. Aplicación de control en Cascada

Como se puede ver, no se manipula el flujo de vapor, éste depende de la posición de la válvula de vapor y de la caída de presión a través de la válvula. Si se presenta una elevación de presión en la tubería de vapor, es decir, si la presión se incrementa antes de la válvula, esto cambia el flujo de vapor; esta perturbación se puede compensar por medio del circuito de control de temperatura únicamente después de que la temperatura del proceso se desvíe del punto de control.

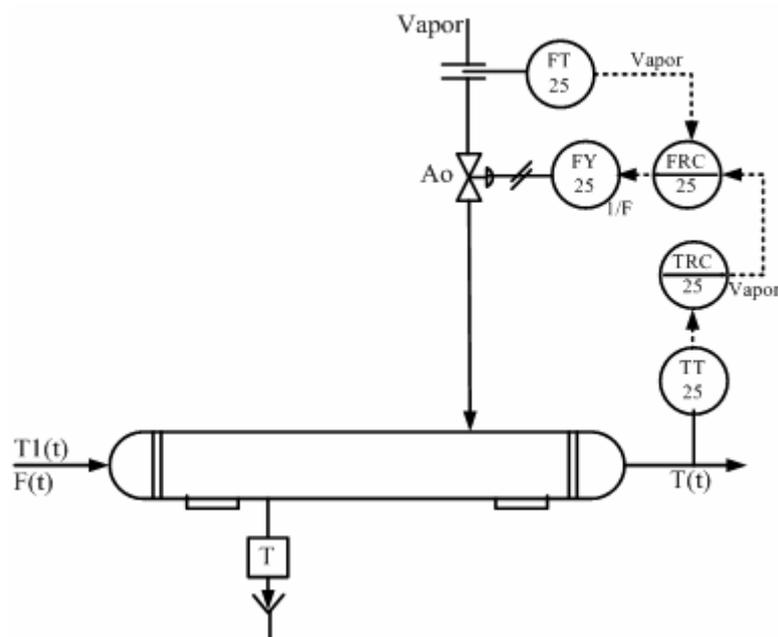
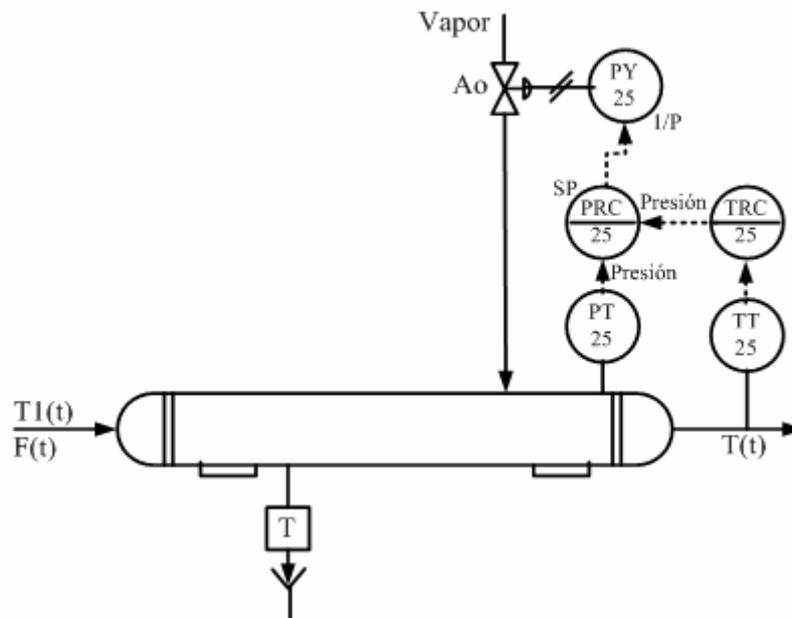


Figura. 4.63. Diagrama de control en Cascada (1)

En la figura 4.63. se muestra un esquema de cascada en el que se añadió un circuito de flujo, el setpoint del controlador de flujo se ajusta con el controlador de temperatura, ahora cualquier cambio en el flujo se compensa por medio del circuito de flujo. El significado físico de la señal que sale del controlador de temperatura es el flujo de vapor que se requiere para mantener la temperatura en el punto de control (setpoint). En este esquema el controlador de flujo es de acción inversa debido a los requerimientos del proceso y a la acción de la válvula de control. El controlador de temperatura también es de acción inversa, esto es, si se incrementa la temperatura de salida se requiere que el flujo de vapor se decremente y por lo tanto se debe disminuir la salida del controlador de temperatura hacia el controlador de flujo.

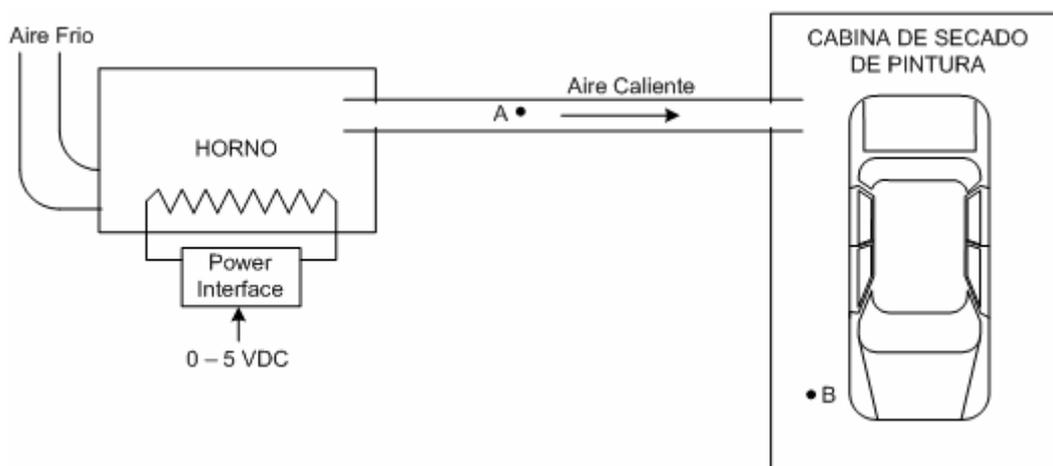


**Figura. 4.64. Diagrama de control en Cascada (2)**

Con el esquema de cascada que aparece en la figura 4.64 se logra el mismo control, pero ahora la variable secundaria es la presión de vapor en el casquillo del intercambiador. Cualquier cambio en el flujo de vapor afecta rápidamente la presión en el casquillo, y cualquier cambio de presión se compensa mediante el circuito de presión. Si se puede instalar un sensor de presión en el intercambiador, entonces la implementación de éste último esquema se hace menos costosa ya que no se requiere un orificio con sus respectivos accesorios, lo cual puede resultar caro. Sin embargo ambos esquemas de control son comunes en la industria

#### 4.15.4. PLANTEAMIENTO

En una planta de pintura de autos, se desea controlar la temperatura de una cabina de secado de pintura que trabaje en un rango de temperatura de 40 a 50 °C. Esta cabina es calentada a través de un flujo de aire caliente que llega a través de una tubería desde un horno, puesto que no se puede controlar el flujo de aire, se debe controlar el sistema calentador del Horno. Además se poseen dos sensores de temperatura, uno ubicado en la tubería de aire (A) y el otro ubicado dentro de la cabina (B) como se muestra en la figura 4.65.



**Figura. 4.65. Planteamiento Práctica #15 (1)**

Diseñar e implementar un sistema de Control en Cascada que permita realizar dicho control. El error estacionario máximo permitido es de  $\pm 0.1^{\circ}\text{C}$ .

Adicionalmente con el software RSView32, crear una HMI que permita interactuar al operador con el sistema controlador, la cual deberá tener las siguientes características:

Dos cuentas de usuario: INGENIERO (con contraseña) y OPERADOR

Pantalla Principal con el siguiente diseño referencial:

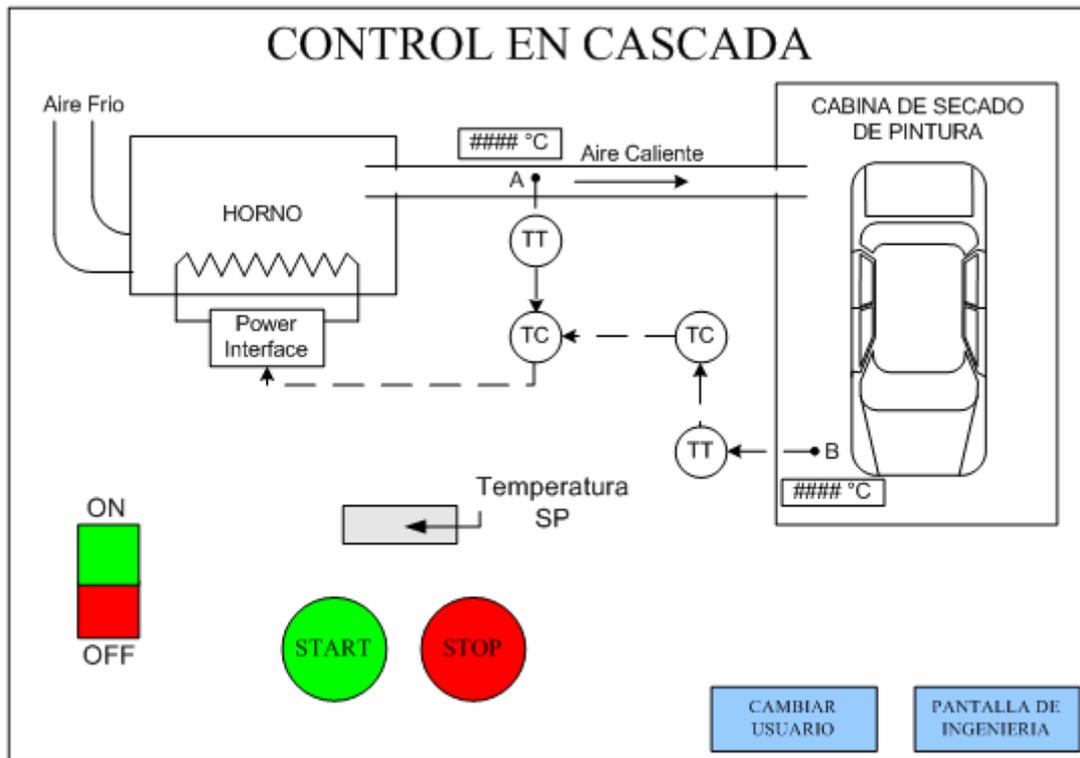


Figura. 4.66. Planteamiento Práctica #15 (2)

- ✓ El botón de "Pantalla de Ingeniería" despliega dicha pantalla únicamente si se encuentra trabajando con el usuario "INGENIERO"

Pantalla de Ingeniería (Acceso permitido únicamente para el usuario INGENIERO)

- ✓ Control de activación del controlador.
- ✓ Selección de la Temperatura de la Cabina (Setpoint).
- ✓ Ingreso de parámetros de calibración de ambos controladores (maestro y esclavo).
- ✓ Gráfica de tendencia (Trend) de la Temperatura real y la deseada del controlador maestro.
- ✓ Gráfica de tendencia (Trend) de la Temperatura real y la deseada del controlador esclavo.
- ✓ Gráfica de tendencia (Trend) de Voltaje de Salida de ambos controladores.

#### 4.15.5. PROCEDIMIENTO

Para la realización de esta práctica el procedimiento a seguir es el siguiente:

1. Configurar el driver en RSLinx
2. Ejecutar RSLogix500, crear un nuevo archivo y configurar el tipo de PLC y sus módulos (de ser el caso) en el programa.
3. Realizar el programa Ladder que cumpla con la lógica de control planteada en esta práctica.
4. Transferir el programa.
5. Ejecutar RSView32, configurar el canal de comunicación, configurar el nodo de comunicación.
6. Crear la interfase HMI que interactúe con el PLC
7. Conectar las entradas y salidas analógicas y digitales que se requieran del PLC.
8. Sintonizar los controladores.
9. Probar si la lógica del programa del PLC y la HMI desarrollada funcionan correctamente, de acuerdo al problema planteado.

#### **4.15.6. PRUEBAS Y RESULTADOS**

#### **4.15.7. CONCLUSIONES Y RECOMENDACIONES**

## 4.16. CONTROL DE PROCESOS 7: CONTROL SELECTIVO

### 4.16.1. OBJETIVOS

- Familiarizarse con la técnica de Control Selectivo.
- Crear una HMI (Interfase Humano Máquina) que interactúe con el PLC.

### 4.16.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 (5/03 o superior)
- ✓ 1 Módulo de Entradas Discretas.
- ✓ 1 Módulo de Salidas Discretas.
- ✓ 1 Módulo de Entradas Analógicas.
- ✓ 1 Módulo de Salidas Analógicas.
- ✓ 1 Computador Personal, con instalación de RSLogix500, RSLinx y RSView32.
- ✓ 1 Módulo Degem de simulación PCT-2 “AIR FLOW TEMPERATURE CONTROL SYSTEM”.
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

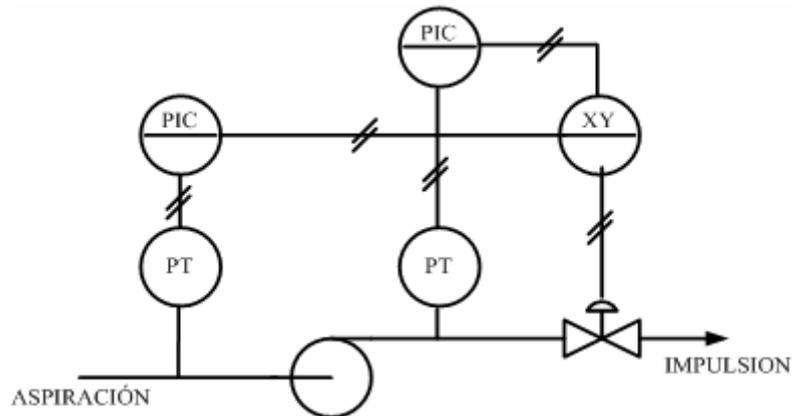
### 4.16.3. MARCO TEÓRICO\*

Este tipo de control selectivo se emplea para limitar la variable del proceso en un valor alto o bajo con el objeto de evitar daños en el producto o en el proceso.

Entre las aplicaciones típicas se encuentra el bombeo de oleoductos, proceso mostrado en la figura 4.67. En esta aplicación hay dos controladores de presión, uno de aspiración y otro de impulsión cuya señal de salida es seleccionada por un relé selector en comunicación con la válvula de control.

---

\*La información expuesta de este tema fue extraída del texto 1 citado en las referencias bibliográficas.



**Figura. 4.67. Aplicación de control Selectivo**

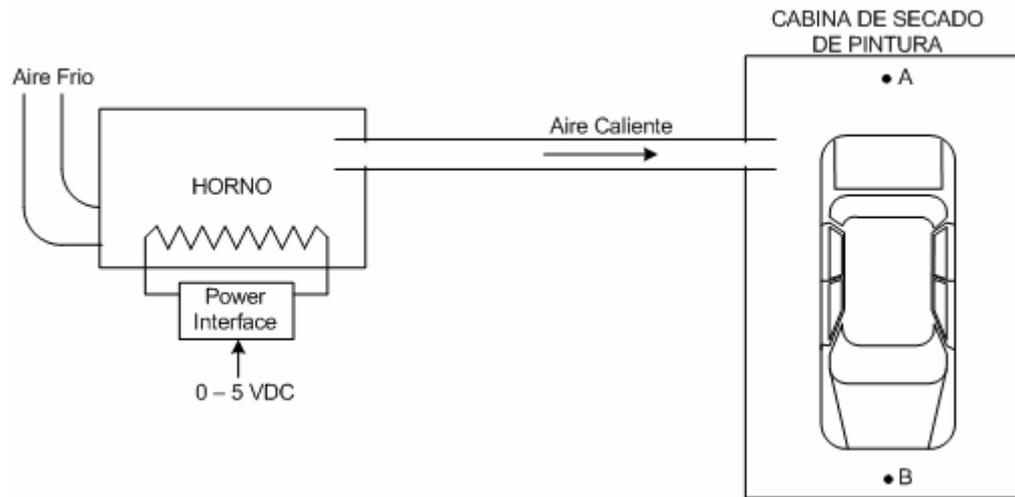
El control se efectúa en condiciones de funcionamiento normales con el controlador de impulsión y cuando por cualquier avería baja la presión de aspiración de la bomba por debajo del límite de seguridad debe entrar en funcionamiento el controlador de aspiración.

Para conseguirlo, el controlador de aspiración es de acción inversa y el setpoint del es inferior a los valores normales de trabajo, y el relé selector selecciona la mínima de las dos señales que le llegan.

De este modo, el control normal se efectuará con el controlador de impulsión y cuando baje demasiado la presión de aspiración y llegue a ser inferior a su setpoint, la señal de salida disminuye y se hace menor a la salida del controlador de impulsión, con lo que el relé selector la selecciona y la válvula pasa a ser controlada directamente por la presión de aspiración.

#### 4.16.4. PLANTEAMIENTO

En una planta de pintura de autos, se desea controlar la temperatura de una cabina de secado de pintura que trabaje en un rango de temperatura de 40 a 50 °C. Esta cabina es calentada a través de un flujo de aire caliente que llega a través de una tubería desde un horno, puesto que no se puede controlar el flujo de aire, se debe controlar el sistema calentador del Horno. Además se poseen dos sensores de temperatura, uno ubicado en la parte frontal de la cabina (A) y el otro ubicado en la parte posterior de la cabina (B) como se muestra en la figura 4.68.



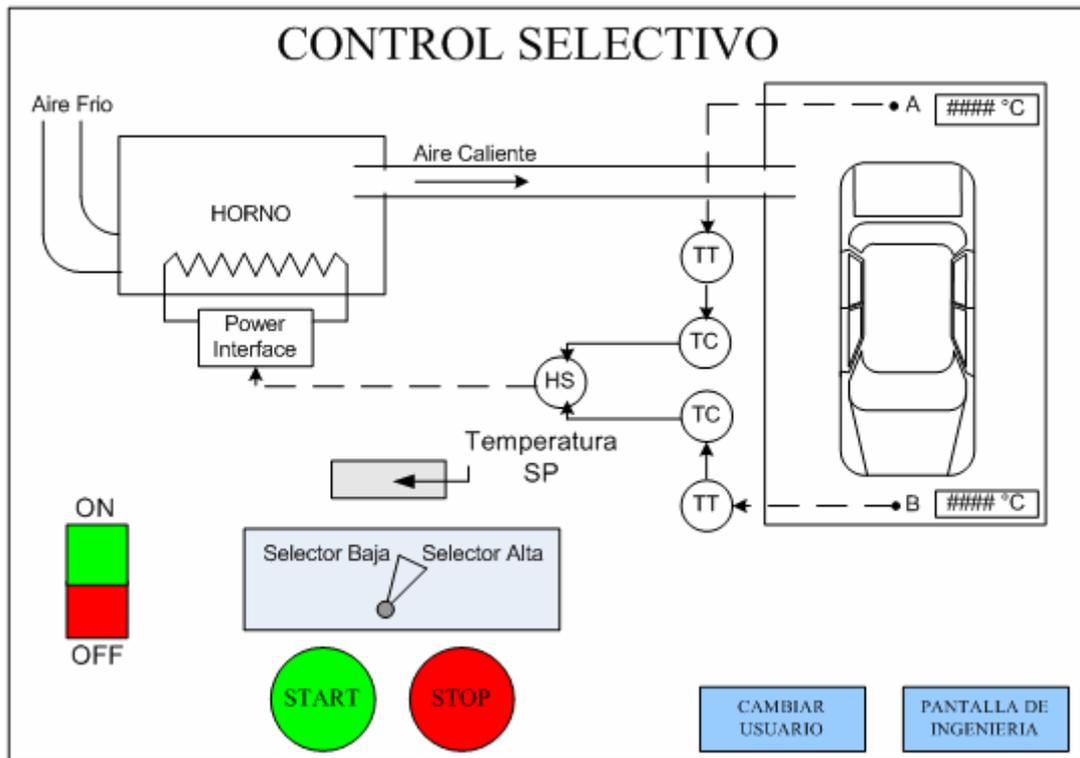
**Figura. 4.68. Planteamiento Práctica #16 (1)**

Diseñar e implementar un sistema de Control Selectivo que permita realizar el control de la temperatura de la cabina. El error en estado estacionario permitido es máximo de  $\pm 0.1^{\circ}\text{C}$ .

Adicionalmente con el software RSView32, crear una HMI que permita interactuar al operador con el sistema controlador, la cual deberá tener las siguientes características:

Dos cuentas de usuario: INGENIERO (con contraseña) y OPERADOR

Pantalla Principal con el siguiente diseño referencial:



**Figura. 4.69. Planteamiento Práctica #16 (2)**

- ✓ El botón de “Pantalla de Ingeniería” despliega dicha pantalla únicamente si se encuentra trabajando con el usuario “INGENIERO”

Pantalla de Ingeniería (Acceso permitido únicamente para el usuario INGENIERO)

- ✓ Control de activación del controlador.
- ✓ Selector de Baja y Alta
- ✓ Selección de la Temperatura de la Cabina (Setpoint).
- ✓ Ingreso de parámetros de calibración de los controladores.
- ✓ Gráfica de tendencia (Trend) de la Temperatura real y la deseada del controlador A.
- ✓ Gráfica de tendencia (Trend) de la Temperatura real y la deseada del controlador B.
- ✓ Gráfica de tendencia (Trend) de Voltaje de Salida de ambos controladores.

#### 4.16.5. PROCEDIMIENTO

Para la realización de esta práctica el procedimiento a seguir es el siguiente:

1. Configurar el driver en RSLinx
2. Ejecutar RSLogix500, crear un nuevo archivo y configurar el tipo de PLC y sus módulos (de ser el caso) en el programa.
3. Realizar el programa Ladder que cumpla con la lógica de control planteada en esta práctica.
4. Transferir el programa.
5. Ejecutar RSView32, configurar el canal de comunicación, configurar el nodo de comunicación.
6. Crear la interfase HMI que interactúe con el PLC
7. Conectar las entradas y salidas analógicas y digitales que se requieran del PLC.
8. Sintonizar los controladores.
9. Probar si la lógica del programa del PLC y la HMI desarrollada funcionan correctamente, de acuerdo al problema planteado.

#### **4.16.6. PRUEBAS Y RESULTADOS**

#### **4.16.7. CONCLUSIONES Y RECOMENDACIONES**

## **4.17. CONTROL DE PROCESOS 8: SCADA 1 COMUNICACIÓN SERIAL**

### **4.17.1. OBJETIVOS**

- Configurar correctamente el canal (0) de comunicación serial del PLC SLC 5/05.
- Conocer las aplicaciones, configuración y conexión del módulo conversor de interfase avanzado 1761-NET-AIC.
- Configurar los Radio Modems Seriales SRM6100
- Crear una HMI que interactúe con el PLC y permita controlar de manera remota las entradas y salidas del PLC.

### **4.17.2. MATERIALES Y EQUIPOS**

- ✓ 1 CPU Allen Bradley SLC 5/05
- ✓ 1 Módulo de Entradas Discretas.
- ✓ 1 Módulo de Salidas Discretas.
- ✓ 1 Radio Modem Master Data-Linc SRM6100
- ✓ 1 Radio Modem Remote Data-Linc SRM6100
- ✓ 1 Módulo conversor de interfase avanzado 1761-NET-AIC
- ✓ 1 Computador Personal, con instalación de RSLogix500, RSLinx y RSView32.
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión Serial

### **4.17.3. MARCO TEÓRICO**

#### **Configuración de los RADIO MODEM SRM6100**

Para configurar los RADIO MODEM se puede emplear el programa Hyper Terminal de Windows XP, por lo cual se deberán seguir los siguientes pasos:

1. Conecte el radio modem al puerto serial de computador
2. Ejecute el programa Hyper Terminal
3. Abra una nueva conexión

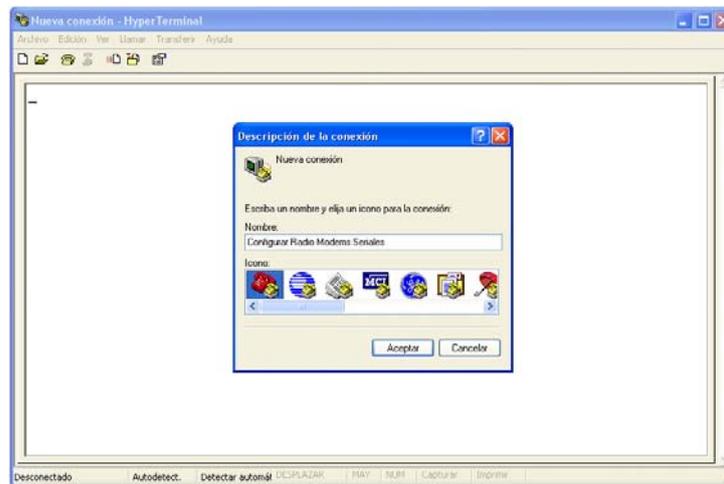


Figura. 4.70. Nueva conexión Hyperterminal

4. Seleccione el puerto serial que se va a utilizar (COM4)



Figura. 4.71. Seleccionar puerto en Hyperterminal

5. Configure los siguientes parámetros de comunicación

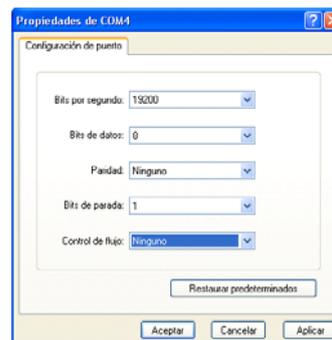


Figura. 4.72. Parámetros de comunicación en Hyperterminal

6. Conéctese con el radio modem

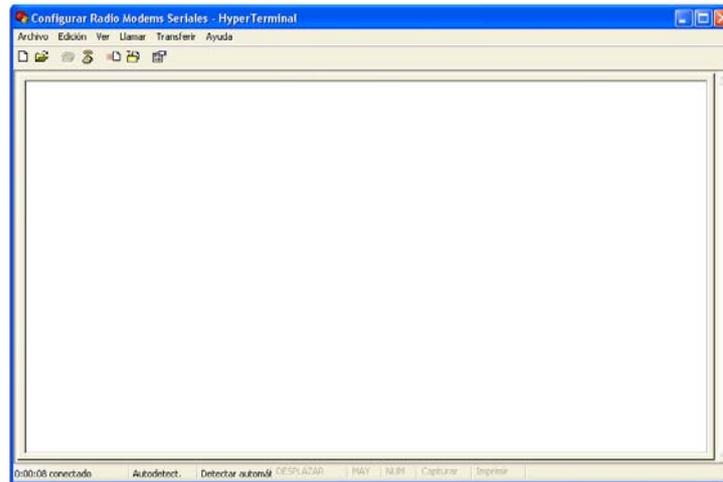


Figura. 4.73. Conexión de Hyperterminal con los Radio módems

7. Presione el botón **Config** de radio modem con la ayuda de un clip, lo que desplegará la pantalla de configuración

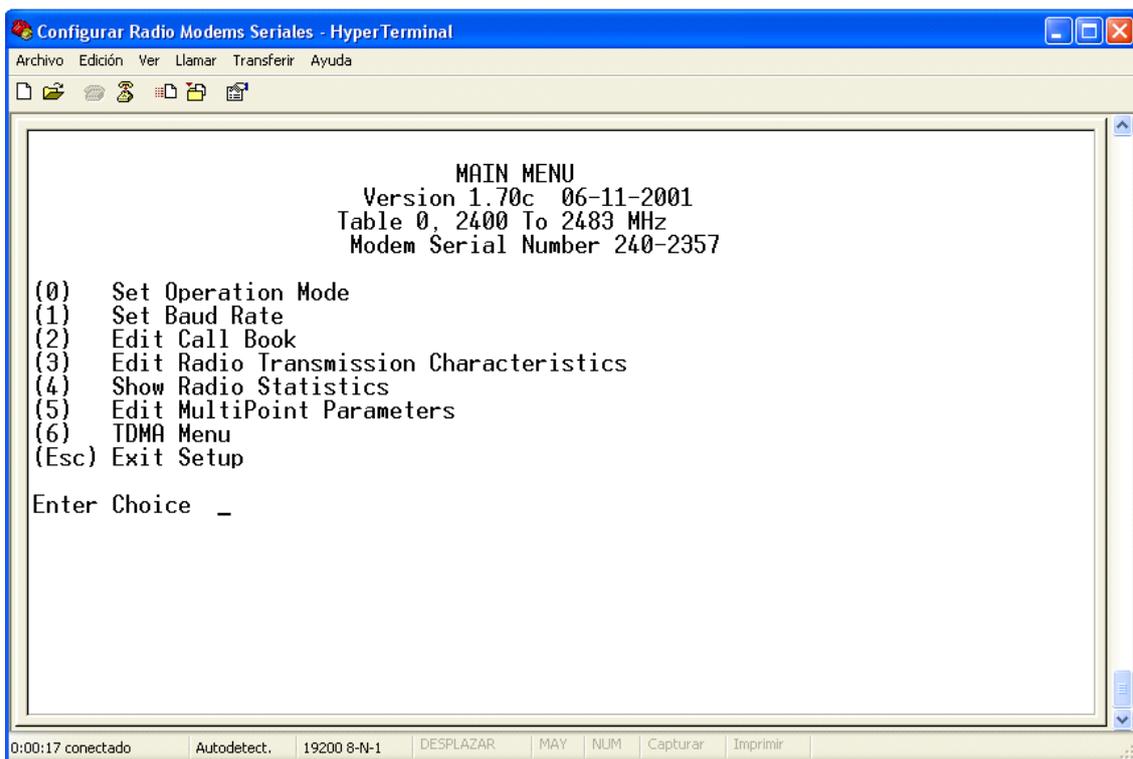


Figura. 4.74. Menú Principal de Configuración de los Radio Modem

A continuación se describirá cada uno de los parámetros a configurar:

### (0) Set Operation Mode

Los SRM6100 pueden operar en los siguientes modos:

- (0) Point to Point Master.
- (1) Point to Point Slave.
- (2) Point to MultiPoint Master
- (3) Point to MultiPoint Slave
- (4) Point to Point Slave/Repeater
- (5) Point to Point Repeater
- (6) Point to Point Slave/Master Switchable
- (7) Point to MultiPoint Repeater

El modo de operación seleccionado dependerá del tipo de comunicación que se requiera.

**(1) Set Baud Rate**

Se pueden seleccionar las siguientes velocidades:

- (1) 115,200
- (2) 76,800
- (3) 57,600
- (4) 38,400
- (5) 19,200
- (6) 9,600
- (7) 4,800
- (8) 2,400

- (9) 1,200
- (A) Data, Parity 0
- (B) MODBus RTU 0
- (Esc) Exit to Main Menu

El número seleccionado en la opción (A) Data, Parity configura los siguientes parámetros:

Data, Parity	Data Bits	Parity	Stop Bits
0	8	None	1
1	7	Even	1
2	7	Odd	1
3	8	None	2
4	8	Even	1
5	8	Odd	1

**Tabla. 4.44. Parámetro Data, Parity**

La opción (B) MODbusRTU habilita (1) y deshabilita (0) el soporte para el protocolo ModBus RTU

## (2) Edit Call Book

### MODEM CALL BOOK

Entry to Call is 00

Entry	Number	Repeater1	Repeater2
-------	--------	-----------	-----------

(0) 240-2334

(1) 000-0000

(2) 000-0000

(3) 000-0000

- (4) 000-0000
- (5) 000-0000
- (6) 000-0000
- (7) 000-0000
- (8) 000-0000
- (9) 000-0000
- (C) Change Entry to Use (0-9) or A(ALL)
- (Esc) Exit to Main Menu

Enter all zeros (000-0000) as your last number in list

En esta pantalla, se debe ingresar el número CALL del radio al cual se quiere conectar, por ejemplo en una conexión Punto-a-Punto en la configuración del master se deberá ingresar el número CALL del remoto (esclavo) y viceversa.

### **(3) Edit Radio Transmission Characteristics**

Esta opción permite modificar las siguientes características de transmisión de los radio módems:

- (0) FreqKey
- (1) Max Packet Size
- (2) Min Packet Size
- (3) Xmit Rate
- (4) RF Data Rate
- (5) RF Xmit Power

- (6) Slave Security
- (7) RTS to CTS
- (8) Retry Time Out
- (9) Lowpower Mode
- (Esc) Exit to Main Menu

Para tener mayor referencia acerca de cada uno de estos parámetros, referirse al manual proporcionado por DATA-LINC GROUP que consta en el ANEXO 1

#### (4) Show Radio Statistics

Esta opción muestra las estadísticas de la transmisión de datos, a continuación se muestra un ejemplo:

#### MODEM STATISTICS

Master-Slave Distance(m) 0006912

Number of Disconnects 1

Radio Temperature 31

Local Remote1 Remote2 Remote3

Average Noise Level 25 24

Average Signal Level 94 94

Overall Rcv Rate (%) 10

89F77E

Press <Ret> for Freq Table, <Esc> to return to main menu

### **(5) Edit MultiPoint Parameters**

Esta opción permite configurar los siguientes parámetros de comunicación, cuando se trabaja en modo multipunto:

- (0) Number Repeaters
- (1) Master Packet Repeat
- (2) Max Slave Retry
- (3) Retry Odds
- (4) DTR Connect
- (5) Repeater Frequency
- (6) NetWork ID
- (7) Reserved
- (8) MultiMasterSync
- (9) 1 PPS Enable/Delay
- (A) Slave/Repeater
- (B) Diagnostics
- (C) SubNet ID
- (D) Radio ID

Los mismos que no se recomienda cambiarlos si no se tiene pleno conocimiento de su incidencia, lo cual se detalla en el manual del ANEXO 1

## Configuración del canal 0 (DF1 Full-duplex) de comunicación del PLC

Para configurar correctamente el canal 0 de comunicación del PLC se debe seguir las siguientes instrucciones:

1. Dar clic en **Controller** (para desplegar las opciones) y dar doble clic en **Channel Configuration**

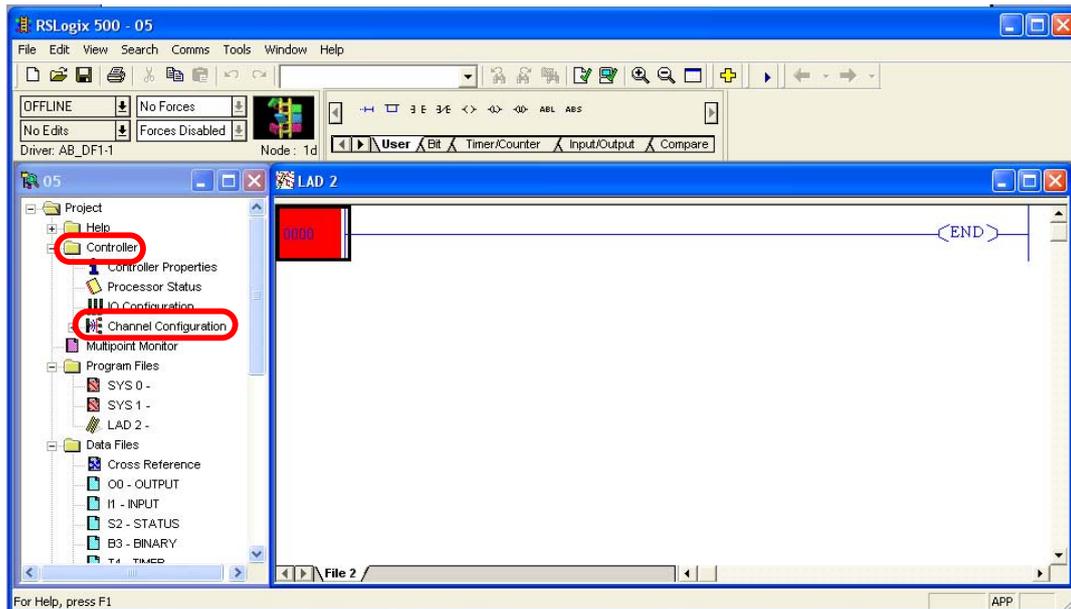


Figura. 4.75. Configuración del canal 0 DF1 Full-Duplex del PLC (1)

2. En el casillero **Diagnostic File** (en la sección Channel 0) asignar un número que sea mayor al número de Data Files que se estén utilizando (ver en la lista de la izquierda de la imagen). Para nuestro caso es 9. Luego dar clic en la pestaña **Chan 0 - System**

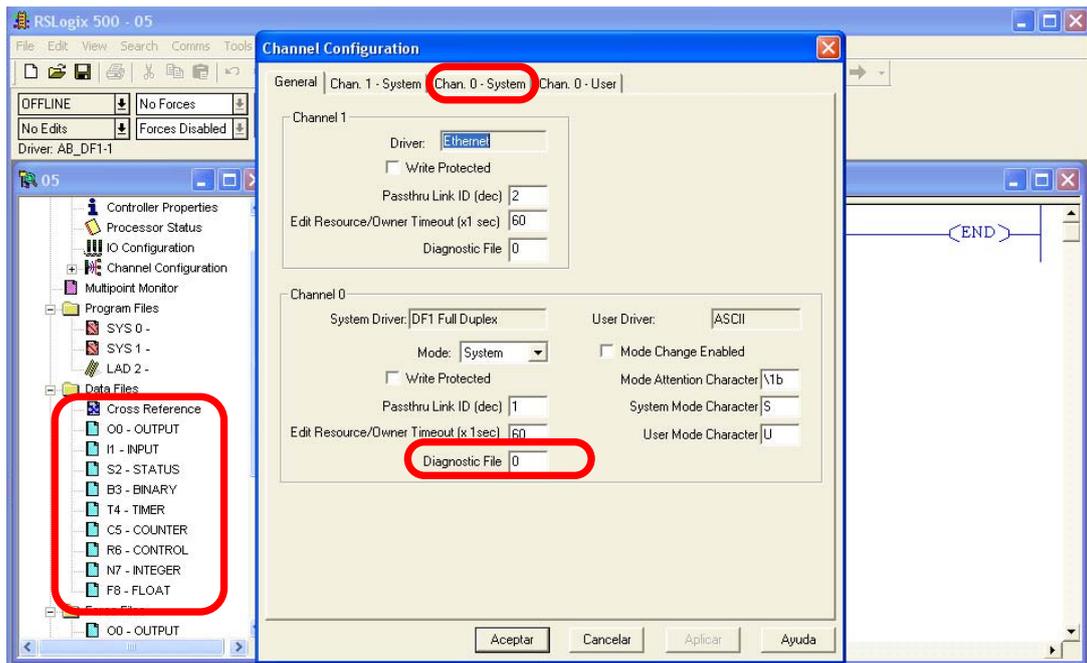


Figura. 4.76. Configuración del canal 0 DF1 Full-Duplex del PLC (2)

3. Aparecerá la siguiente pantalla en la que se puede configurar los parámetros de comunicación, para posteriormente presionar *Aceptar*.

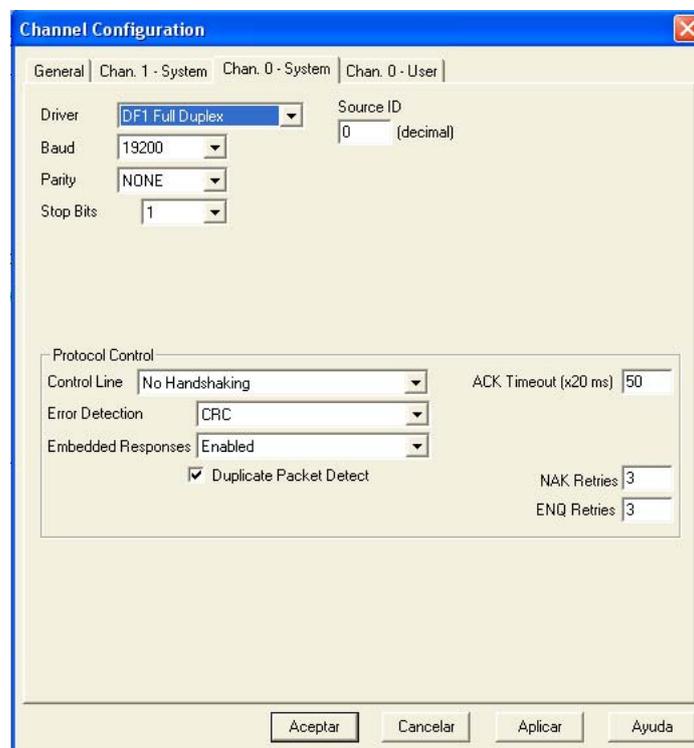
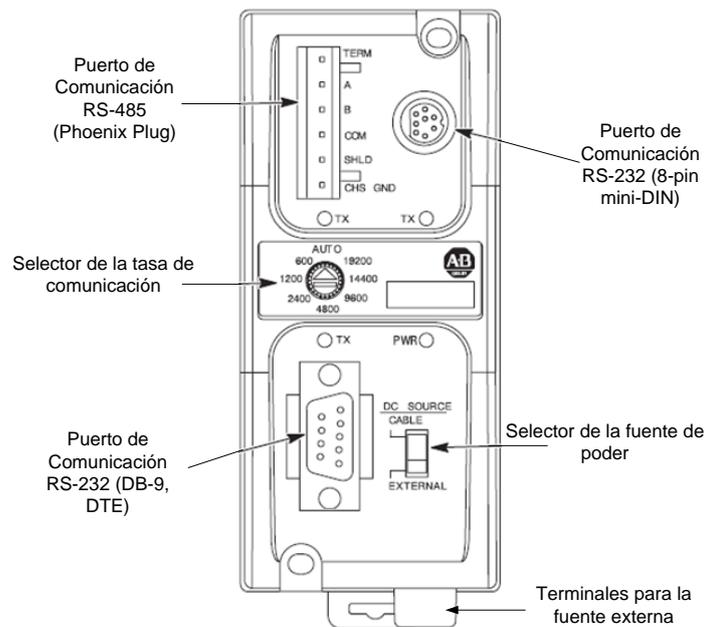


Figura. 4.77. Configuración del canal 0 DF1 Full-Duplex del PLC (7)



**Figura. 4.78. Descripción del Módulo 1761-NET-AIC**

Este módulo permite conectar varios dispositivos puesto que es compatible con una gran variedad de controladores y periféricos de la serie SLC y Micrologix.

Los controladores Micrologix 1000 proveen energía al módulo 1761-NET-AIC a través del cable RS-232 8-pin mini-DIN, sin embargo si se conecta otro tipo de controlador se requiere conectar una fuente externa en los terminales del módulo.

El selector de la tasa de comunicación es utilizado para hacer coincidir la tasa de transmisión del módulo con la de la red a la cual se va a conectar. Normalmente se debe dejar en la posición AUTO, únicamente en ambientes con mucho ruido se debe seleccionar la misma tasa de transmisión de la red

### **Modos de Operación**

Este módulo puede ser utilizado en los siguientes modos de operación:

- Aislador Punto a Punto
- Aislador de RS-232 a RS-485
- Aislador de RS-232 a half-duplex ASCII

### **Configuración de la dirección del nodo de comunicación**

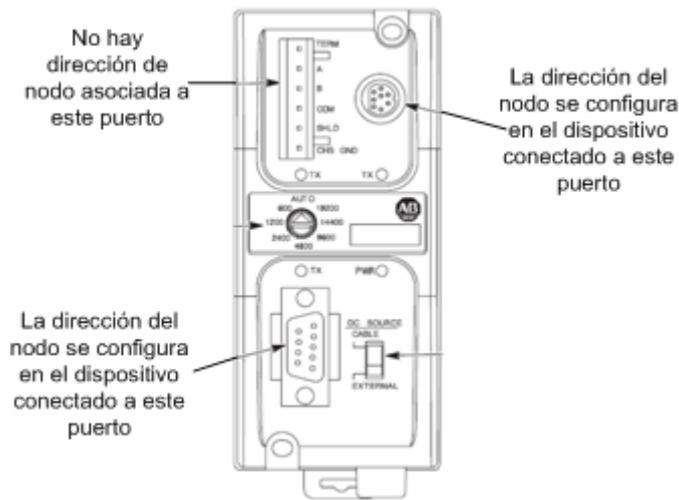


Figura. 4.79. Configuración de los nodos de comunicación del módulo NET-AIC

**Diagrama de conexión**

La conexión del módulo en modo aislador Punto a Punto se muestra en la siguiente figura 4.80. Si se requiere conectar un PLC de la serie SLC500 se debe conectar una fuente externa al módulo.

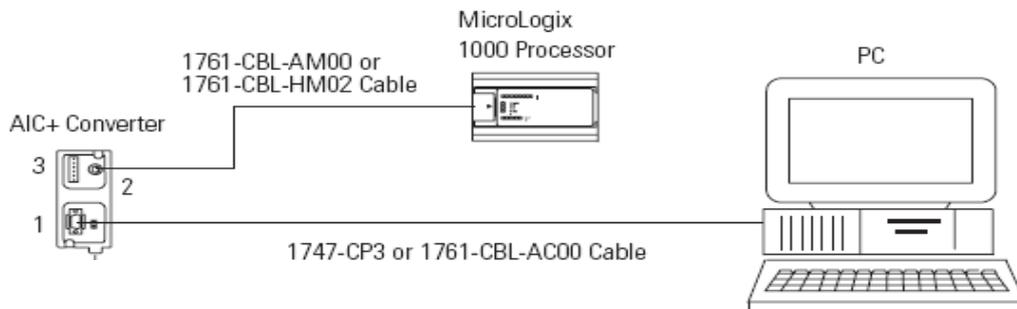


Figura. 4.80. Conexión del módulo NET-AIC en modo aislador punto a punto

**4.17.4. PLANTEAMIENTO**

Se desea conectar los siguientes equipos en la configuración mostrada en la figura 4.81.

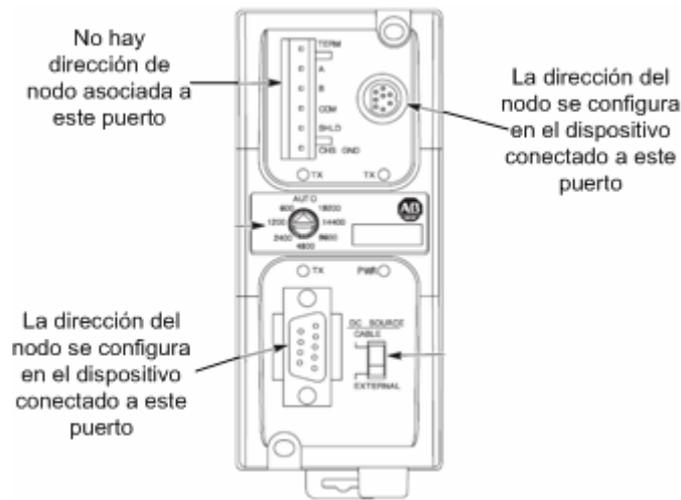


Figura. 4.81. Planteamiento Práctica #17 (1)

El computador personal deberá poseer una HMI (realizada en RSView32) que permita monitorear las Entradas Discretas y controlar las Salidas Discretas del PLC.

La HMI deberá contar con una sola pantalla con el siguiente esquema:

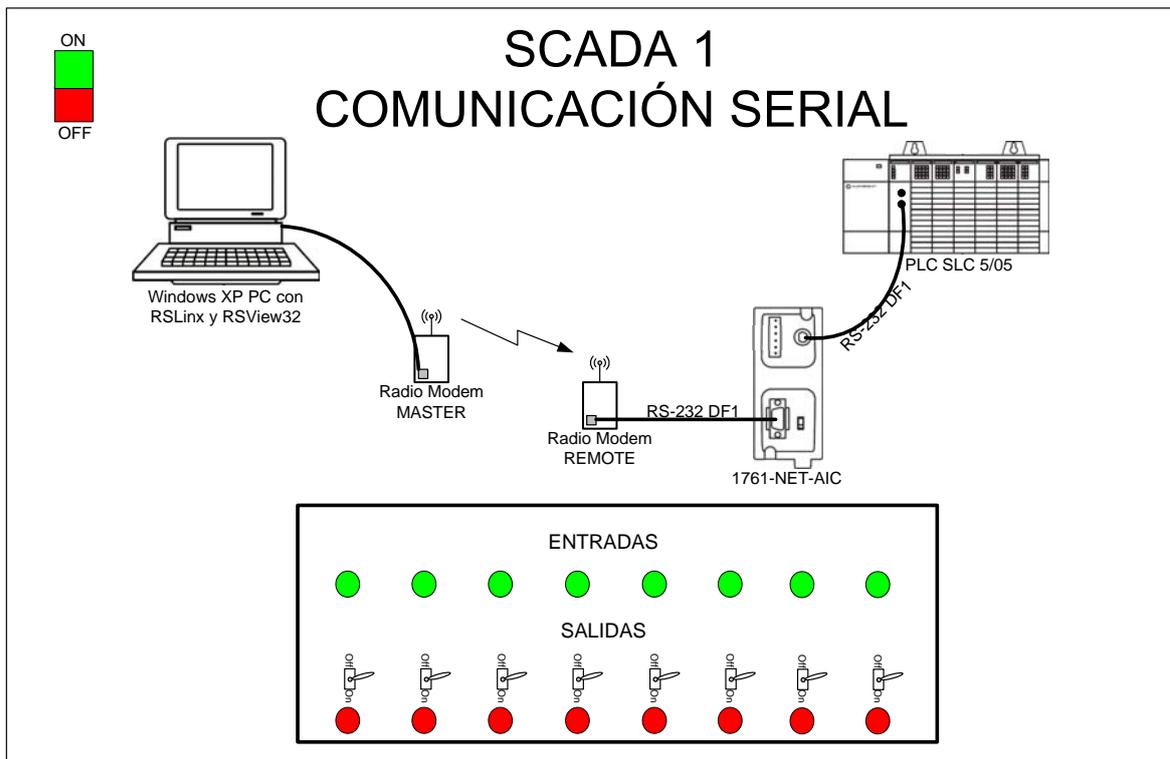


Figura. 4.82. Planteamiento Práctica #17 (2)

#### **4.17.5. PROCEDIMIENTO**

Para la realización de esta práctica el procedimiento a seguir es el siguiente:

1. Configurar el Hyper Terminal para conectarse a los Radio Modems
2. Configurar los Radio Modems para que se enlacen entre si.
3. Conectar la PC, el PLC, el módulo 1761-NET-AIC+ y los Radio Modems de acuerdo a la configuración planteada.
4. Configurar el Driver Serial en RSLinx para comunicarse con el PLC SLC 5/05
5. Abrir el software RSLogix500, crear un nuevo archivo y configurar el PLC SLC 5/05, sus módulos y los dos canales de comunicación serial.
6. Realizar el programa Ladder que cumpla con los requerimientos definidos en el planteamiento de la práctica.
7. Transferir el programa.
8. Abrir el software RSView32, configurar el canal de comunicación, configurar el nodo de comunicación.
9. Crear la interfase HMI que interactúe con el PLC de acuerdo al problema planteado.
10. Correr el PLC y la HMI, probar si la lógica del programa del PLC y la HMI desarrollada funcionan correctamente, de acuerdo al problema planteado.

#### **4.17.6. PRUEBAS Y RESULTADOS**

#### **4.17.7. CONCLUSIONES Y RECOMENDACIONES**

## **4.18. CONTROL DE PROCESOS 9: SCADA 2 COMUNICACIÓN ETHERNET**

### **4.18.1. OBJETIVOS**

- Conocer el funcionamiento, configuración y aplicación de la Instrucción MSG.
- Conectar PLC's 5/03 a una red Ethernet, utilizando los módulos de interfase ethernet
- Configurar el canal 1 de los PLC's 5/05 para conectarlos a una red Ethernet
- Enviar mensajes de alarma vía e-mail, a través de los módulos de interfase ethernet
- Crear una HMI que interactúe con el PLC y permita controlar de manera remota varios procesos.

### **4.18.2. MATERIALES Y EQUIPOS**

- ✓ 1 CPU Allen Bradley SLC 5/03
- ✓ 1 CPU Allen Bradley SLC 5/05
- ✓ 2 Módulos de Entradas Discretas.
- ✓ 2 Módulos de Salidas Discretas.
- ✓ 2 Radio Modems Ethernet Data-Linc SRM6310E
- ✓ 1 Switch Ethernet
- ✓ 1 Módulo de Interfase Ethernet 1761-NET-ENIW
- ✓ 1 Computador Personal, con instalación de RSLogix500, RSLinx y RSView32.
- ✓ 1 Punto de conexión a la red ethernet de la ESPE
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ 4 Cables de conexión Ethernet
- ✓ 1 Cable 1761-CBL-PM02 PC to Micrologix
- ✓ 1 Cable Serial C232/SRM61/PC (Programación Radio Modem)

### **4.18.3. MARCO TEÓRICO**

#### **Configuración de los Radio Modem SRM6310E**

La configuración del Hyper Terminal, para conectarse con los Radio Modems es la misma que se describió en la práctica anterior.

Los parámetros de configuración de los Radio Modems Ethernet son los mismos que los de los Radio Modems Seriales (Práctica anterior) con excepción de algunos parámetros propios de la comunicación ethernet.

Sin embargo para realizar esta práctica, únicamente se requiere configurar el modo de operación (opción 0), Baud Rate (opción 1) y el libro de llamadas (opción 2).

Si se requiere conocer acerca de los demás parámetros de configuración de los Radio Modem SM6310E se recomienda consultar el Manual de Usuario proporcionado por DATA-LINC GROUP, el mismo que consta en el ANEXO 2.

### Configuración del Driver Ethernet de comunicación entre el PLC y la PC

Para configurar correctamente el driver de comunicación ethernet, se deben seguir las siguientes instrucciones:

1. Correr el software RSLinx y dar clic en *Configure Drivers*

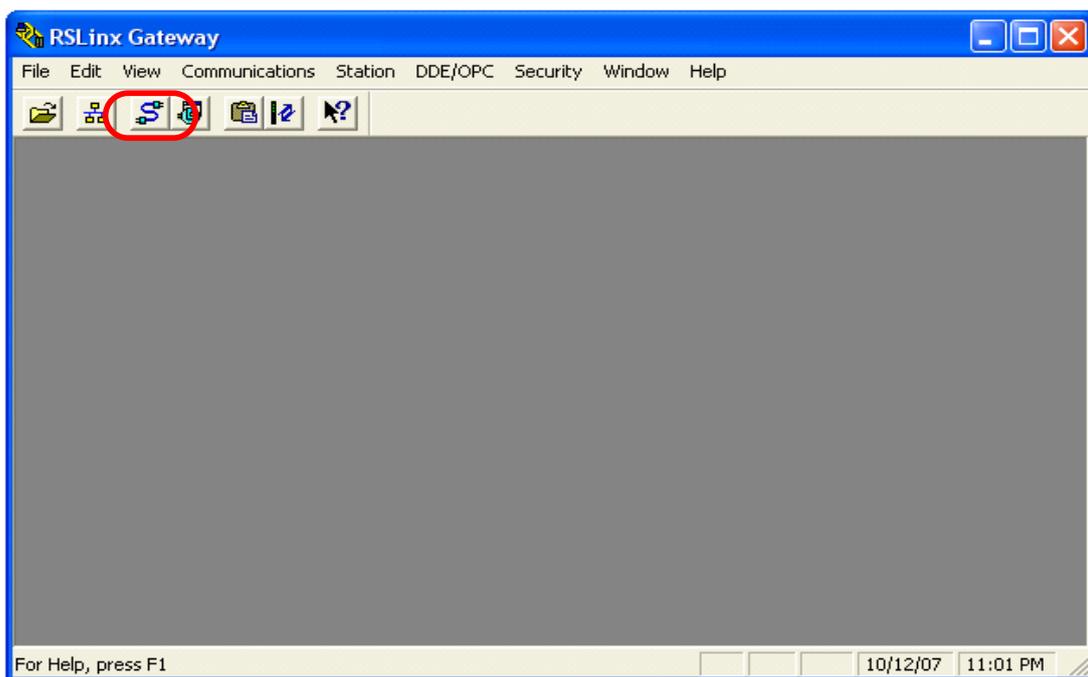


Figura. 4.83. Configuración del Driver de comunicación Ethernet (1)

2. En la pantalla que aparece, seleccionar el driver de comunicación “*Ethernet devices*” y hacer clic en *Add New* como se indica en la figura 4.84..

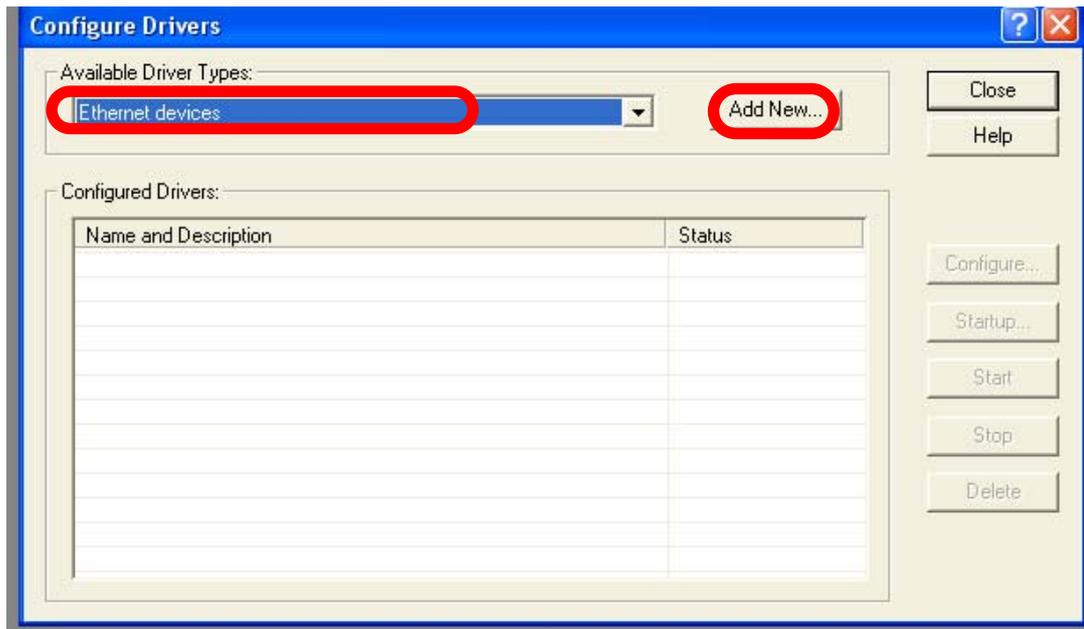


Figura. 4.84. Configuración del Driver de comunicación Ethernet (2)

3. Escribir un nuevo nombre para el Driver o dejar el que viene por defecto y presionar **OK**.

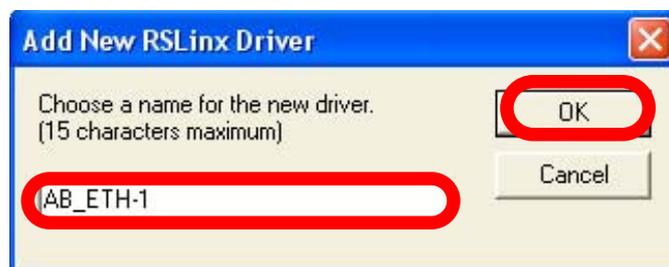
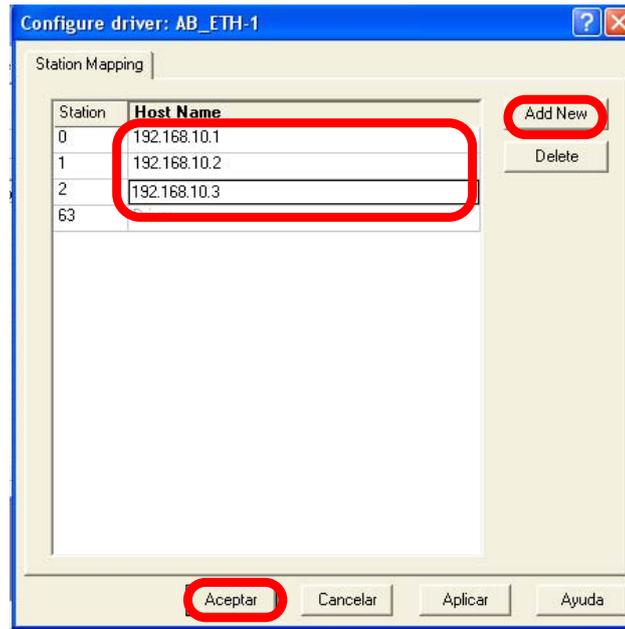


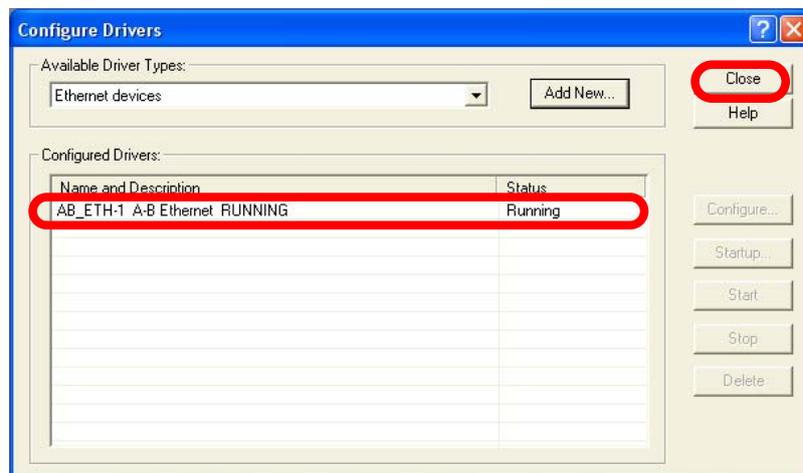
Figura. 4.85. Configuración del Driver de comunicación Ethernet (3)

4. Ingrese las direcciones IP, presionando **Add New** cada vez que se requiera ingresar una nueva dirección. Se deben ingresar la direcciones de todos los equipos que se vayan a conectar a la red (PLC's, PC, Módulos de interfase Ethernet, etc.) y finalmente presionar **OK**



**Figura. 4.86. Configuración del Driver de comunicación Ethernet (4)**

5. Finalmente, verificar que el driver esté funcionando correctamente, y presionar **Close** con lo que se volverá a la pantalla de inicio.



**Figura. 4.87. Verificación del driver de comunicación serial Ethernet**

### Configuración del canal 1 (Ethernet) de comunicación del SLC 5/05

Para configurar correctamente el canal 1 (Ethernet) de comunicación del PLC SLC 5/05 se debe seguir las siguientes instrucciones:

1. Dar clic en **Controller** (para desplegar las opciones) y dar doble clic en **Channel Configuration**

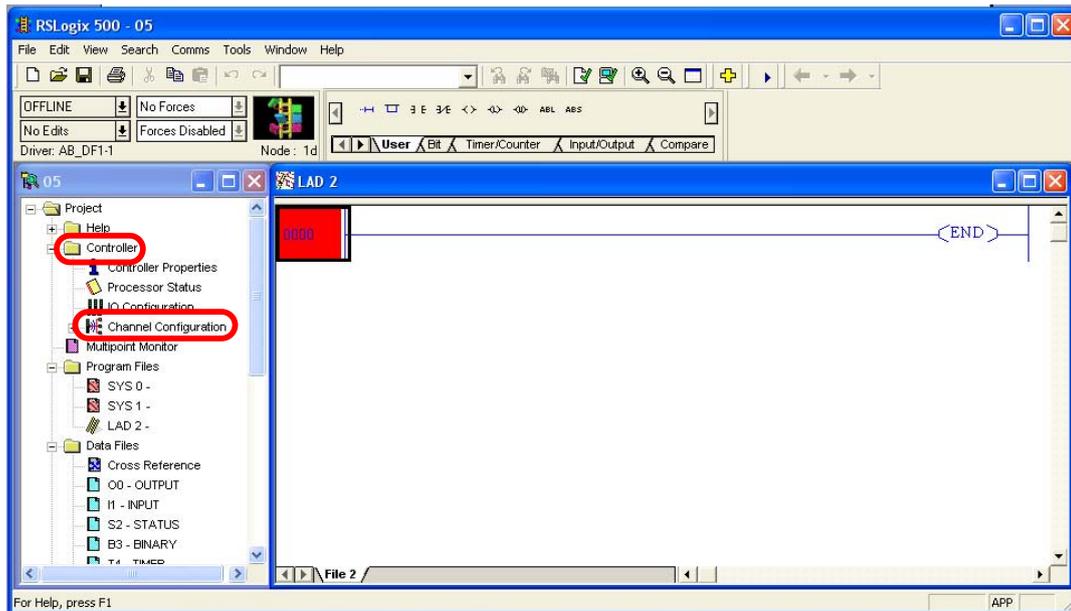


Figura. 4.88. Configuración del canal 1 Ethernet del PLC (1)

2. En el casillero Diagnostic File (de la sección Channel 1) asignar un número que sea mayor al número de Data Files que se estén utilizando (ver en la lista de la izquierda de la imagen). Para nuestro caso es 9. Luego dar clic en la pestaña Chan 1 – System.

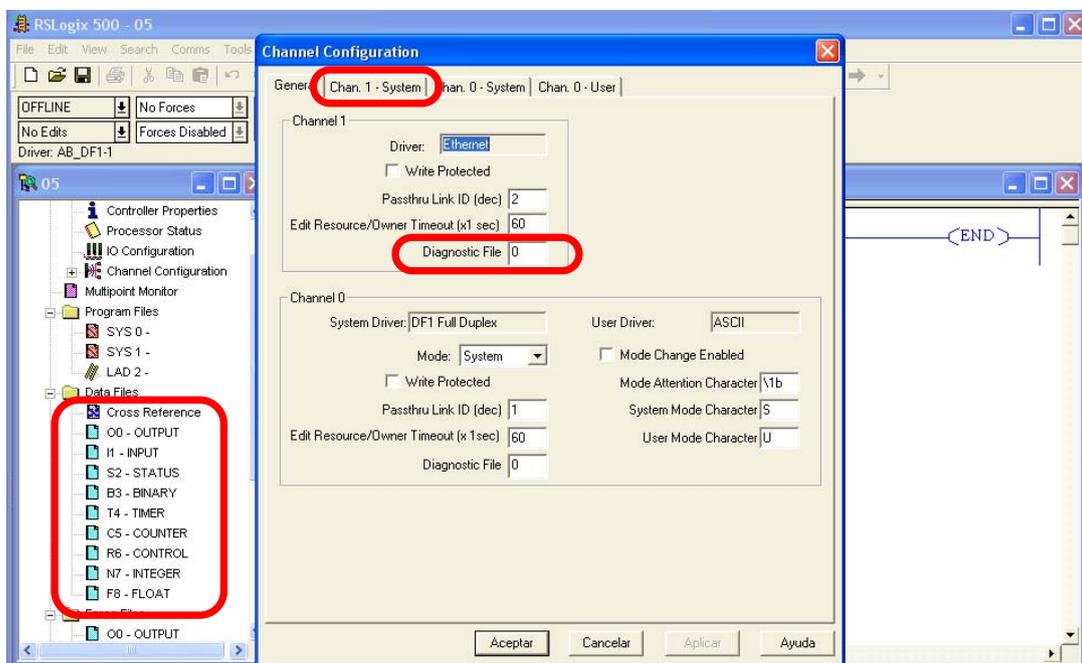


Figura. 4.89. Configuración del canal 1 Ethernet del PLC (2)

3. Ingrese la dirección IP y la máscara de subred, y finalmente presionar *Aceptar*

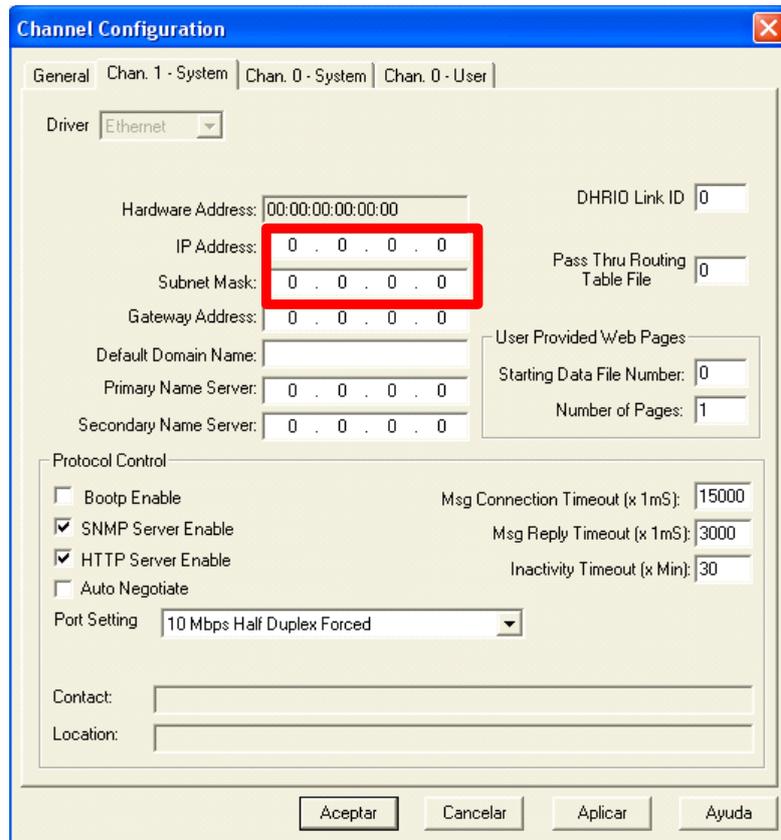


Figura. 4.90. Configuración del canal 1 Ethernet del PLC (3)

**Instrucciones de comunicación**

<i>Mensaje (lectura y/o escritura)</i>		
<b>SÍMBOLO</b>	<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>
	<p>Message Instruction (MSG)</p>	<p>Ésta es una instrucción de salida que permite leer o escribir datos desde un procesador a otro a través de los canales de comunicación.</p>

Tabla. 4.45. Instrucción MSG

Para ejecutar esta instrucción, se debe provocar una transición Off-On de las condiciones que la preceden (escalón). No active nuevamente la instrucción, hasta que la instrucción haya completado con o sin éxito el mensaje, lo cual se indica cuando el procesador activa el bit DN o ER.

## Principales Opciones de Configuración de la Instrucción MSG

Existen las siguientes opciones de configuración:

- Lectura/Escritura Punto a Punto (Peer-to-peer Read/Write) en una red local a otro procesador SLC 500
- Lectura/Escritura Punto a Punto (Peer-to-peer Read/Write) en una red remota a otro procesador SLC 500

## Parámetros de la instrucción MSG

Cuando se programe esta instrucción, se deben ingresar los siguientes parámetros:

- **Read/Write:** Read indica que el procesador local (procesador en el cual se encuentra la instrucción MSG) está recibiendo datos, Write indica que el procesador local está enviando datos.
- **Target Device:** Identifica el tipo de dispositivo el cual responde al comando MSG. En nuestro caso debe ser 500CPU.
- **Local o Remote:** Identifica si el mensaje es enviado a un dispositivo en una red local o remota.
- **Control Block:** Es una dirección de un archivo de datos de tipo entero que el programador debe seleccionar. Este es un bloque de palabras que contienen los bits de Status, la dirección del objetivo y otros datos asociados a la instrucción MSG.
- **Control Block Length:** Este indica cuantas palabras están siendo utilizadas en el block de control.

## Parámetros de la Pantalla SETUP SCREEN

La pantalla Setup Screen se muestra en la figura 4.91.

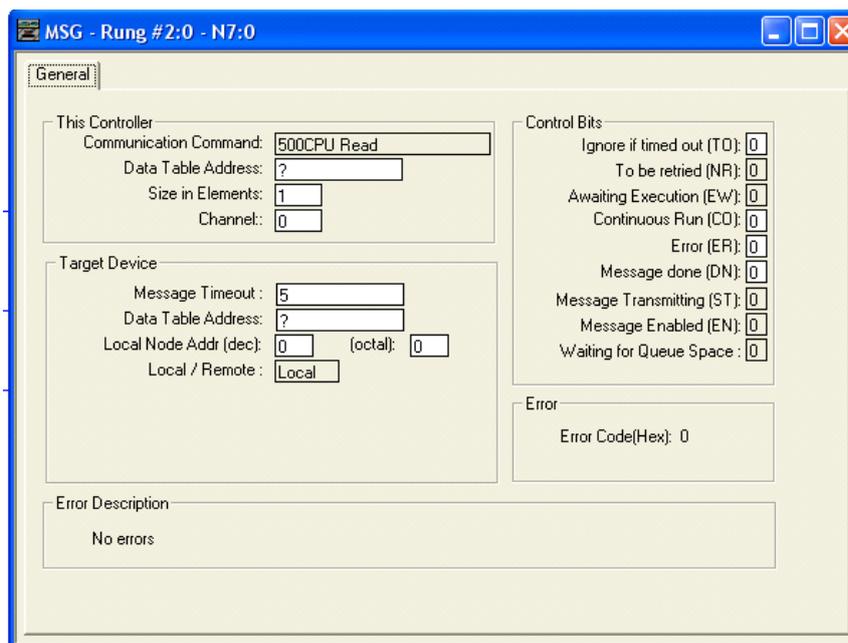


Figura. 4.91. Pantalla de configuración de la instrucción MSG

#### Parámetros “This Controller”

- **Data Table Address**
  - ✓ Para lectura (Read), esta es la dirección de inicio en la cual se recibe los datos desde el dispositivo objetivo (Target Device).
  - ✓ Para Escritura (Write), esta es la dirección de inicio de los datos que son escritos en el dispositivo objetivo (Target Device).
- **Size in Elements:** Especifica la longitud del mensaje en elementos. El número máximo de elementos que son transferidos a través de la instrucción MSG está determinado por el tamaño del tipo de datos del destino.
  - ✓ Para lectura (Read), el tipo de datos en el procesador local determina el número máximo de elementos.
  - ✓ Para Escritura (Write), el tipo de datos en el dispositivo objetivo determina el número máximo de elementos.
- **Channel:** Especifica el canal de comunicación que está siendo utilizado para transmitir el mensaje.
  - ✓ SLC 5/03 - (Canal 0, RS-232) ó (Canal 1, DH-485)

- ✓ SLC 5/05 - (Canal 0, RS-232) ó (Canal 1, Ethernet)

### Parámetros “Target Device”

- **Message Timeout:** Especifica la duración en segundos de el temporizador del mensaje. Un valor de 0 segundos significa que no existe temporizador, es decir, un mensaje que ha sido reconocido, espera indefinidamente por la respuesta. Este parámetro tiene un rango valido entre 0 y 255. Cuando se utiliza el canal 1 (ethernet) de los procesadores SLC 5/05, este parámetro no debe modificarse pues es asignado por el controlador
- **Data Table Address**
  - ✓ Para lectura (Read), Esta es la dirección de inicio de donde los datos están siendo leídos.
  - ✓ Para Escritura (Write), Esta es la dirección de inicio en donde los datos están siendo escritos.
- **Local Node Addr (dec)/(oct):** Especifica el número de nodo del dispositivo que está recibiendo el mensaje. Este parámetro tiene un rango valido que depende del tipo de comunicación utilizada, como se puede ver en la siguiente tabla:

Protocolo	Decimal	Octal
DH-485	0-31	0-37
DH+	0-63	0-77
DF1	0-254	0-376

**Tabla. 4.46. Local Node Instrucción MSG**

- **Ethernet (IP) Address:** Especifica la dirección IP de dispositivo objetivo, es decir del que está recibiendo el mensaje.
- **MultiHop:** Especifica si el mensaje esta siendo o no está siendo enviado a un dispositivo de interfase Ethernet (1761-NET-ENIW).

### **Configuración del canal directo de comunicación Ethernet en el Software RSView32**

Una vez creado un nuevo proyecto (revisar Práctica **Control de Procesos 1**), realizar los siguientes pasos:

1. Dar clic en **System** para desplegar las opciones, dar doble clic en **Channel**

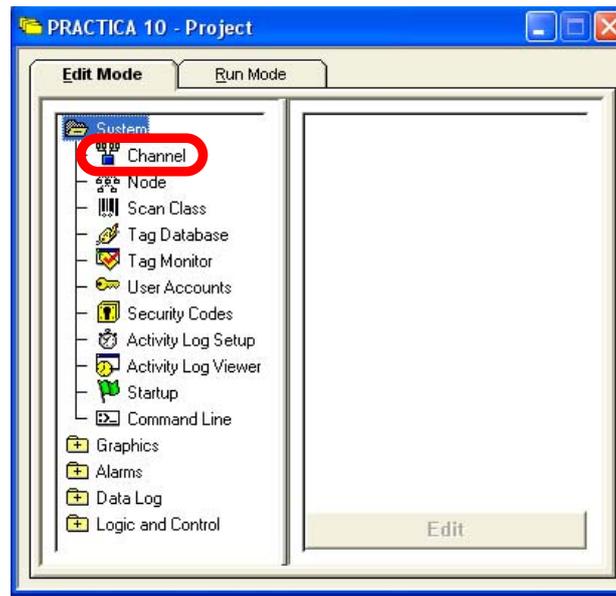


Figura. 4.92. Configuración del canal Ethernet en RSView32 (1)

2. Escoger el número de canal (en este caso 1), en **Network Type** escoger TCP/IP, en **Messages** escriba un número entre 1 y 10 (este es el número de mensajes que el software puede enviar por un canal, antes de requerir una respuesta), en **Primary Communication Driver** seleccione el driver antes configurado en RSLinx. Finalmente presione OK

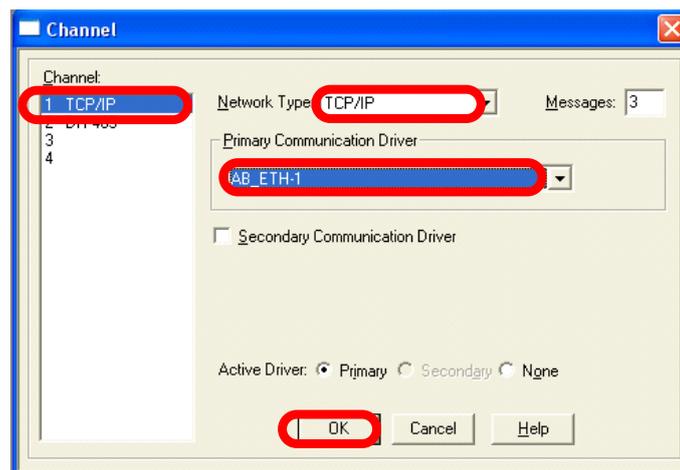


Figura. 4.93. Configuración del canal Ethernet en RSView32 (2)

3. Dar doble clic en **Node**

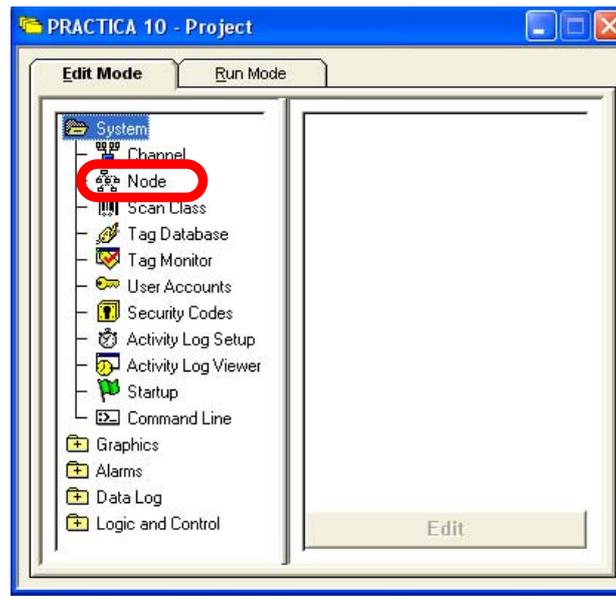


Figura. 4.94. Configuración del canal Ethernet en RSView32 (3)

4. Seleccionar *Direct Driver*, en *Name* asignar un nombre al nodo, en *Channel* escoger el canal creado anteriormente, en *Station* presionar el botón de explorar para seleccionar el PLC al que se quiere conectar, en *Type* seleccionar el tipo de PLC *SLC 5 (Enhanced)*, y presionar *Aceptar*:

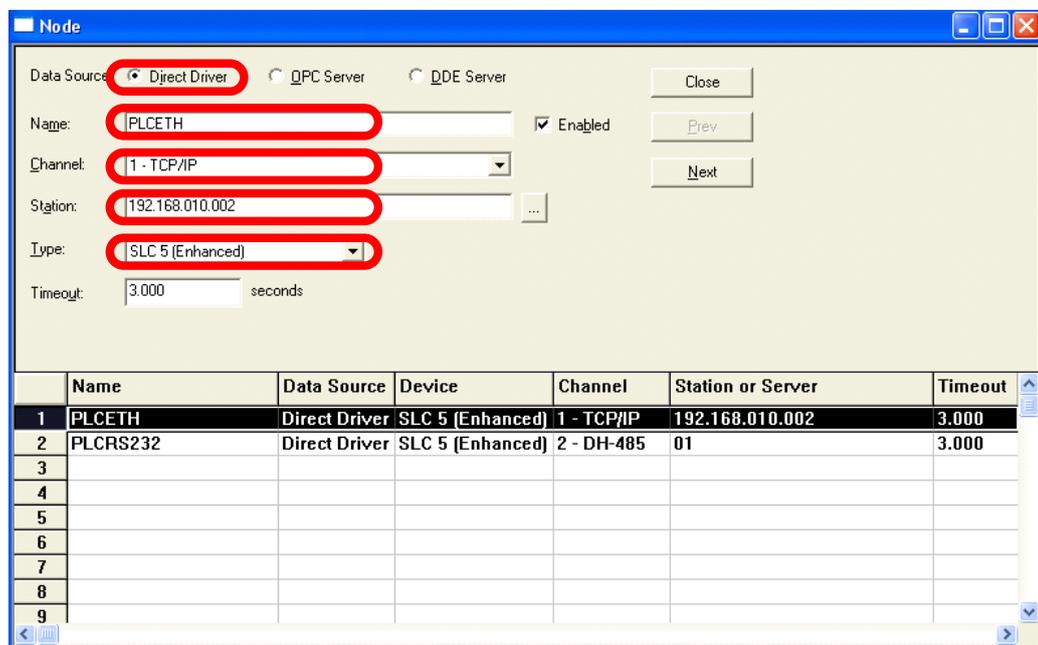
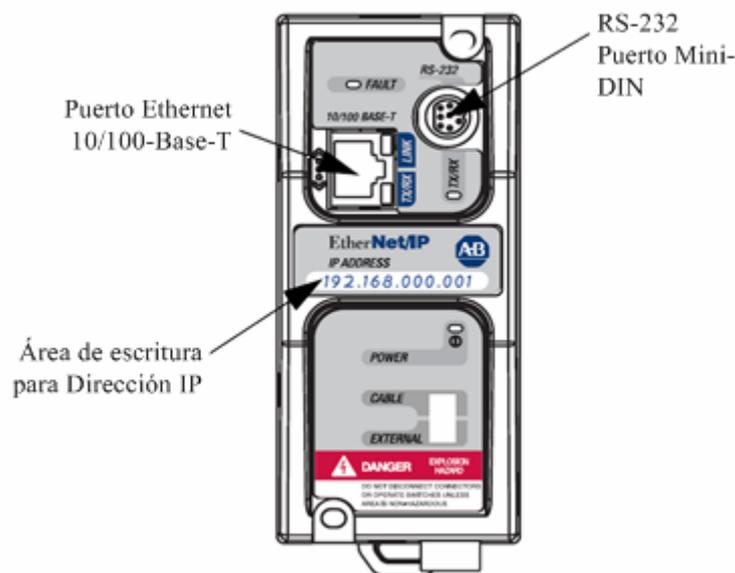


Figura. 4.95. Configuración del canal Ethernet en RSView32 (4)

### Módulo de Interfase Ethernet 1761-NET-ENIW



**Figura. 4.96. Descripción del módulo de Interfase Ethernet 1761-NET-ENIW**

El módulo de interfase ENIW, nos permite conectar fácilmente controladores sin puerto ethernet a redes EtherNet/IP para cargar y descargar programas, comunicarse entre controladores y generar mensajes de e-mail a través de SMTP (Simple Mail Transport Protocol).

### Configuración del Módulo de Interfase Ethernet 1761-NET-ENIW

La configuración del módulo ENIW puede ser ingresada de las siguientes maneras:

- Por medio del software “ENI/ENIW Configuration Utility”, a través del puerto RS-232.
- Por medio del software “ENI/ENIW Configuration Utility”, a través del puerto Ethernet, utilizando el software “Com Port Redirector”
- Por medio de mensajes (instrucción MSG) desde un controlador Allen Bradley al nodo respectivo a través del puerto RS-232. Ej: Para ingresar la dirección IP, se debe enviar esta dirección al nodo 250.

En este caso, por facilidad se lo va a configurar por medio del software “ENI/ENIW Configuration Utility”, a través del puerto RS-232. Para lo que se deben seguir los siguientes pasos:

1. Correr el software ENI/ENIW Utility.

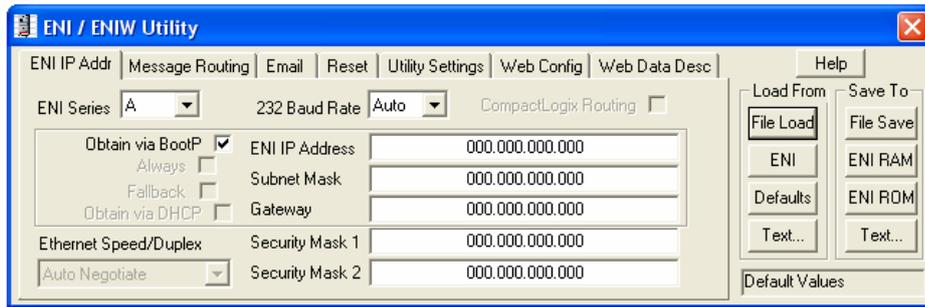


Figura. 4.97. Pantalla de configuración 1 ENIW

2. Dar clic en la pestaña *Utility Settings* para configurar el canal de comunicación.



Figura. 4.98. Pantalla de configuración 2 ENIW

3. Dar clic en la pestaña *ENI IP Addr* e ingrese los siguientes parámetros:

- ENI Series: D (los módulos son de la serie D)
- 232 Baud Rate: Seleccionar la velocidad de transmisión serial del dispositivo conectado al módulo. Por ejemplo, si está conectado un PLC de la serie 5/03, está será la velocidad de transmisión del canal 0 (generalmente 19200).
- Obtain via Bootp: Desactivado (Quitar el visto de selección)
- ENI IP Address: Ingresar la dirección IP (por Ej: 192.168.010.002)
- Subset Mask: Ingresar la máscara de subred (por Ej: 255.255.255.0)

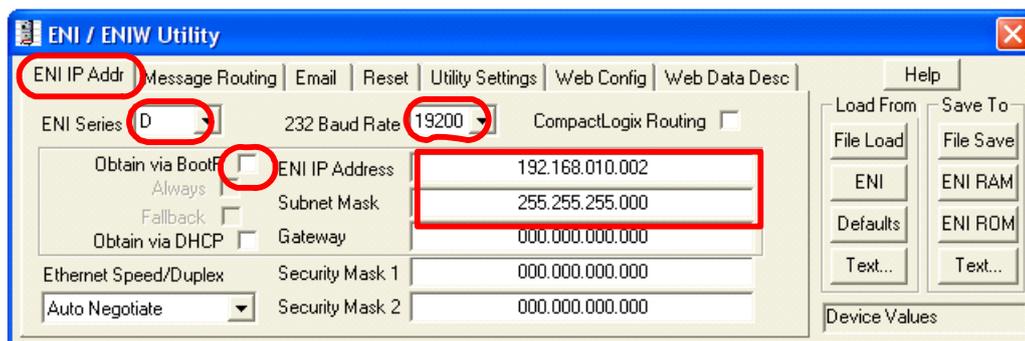


Figura. 4.99. Pantalla de configuración 3 ENIW

4. Dar clic en la pestaña **Message Routing** en donde a un nodo (Dest) se le asigna la dirección IP de un dispositivo de la red. Por ejemplo, si se tiene dos PLC's conectados a una red Ethernet, un 5/05 (conectado directamente a través del canal 1) y un 5/03 (conectado a través del módulo ENIW), y se quiere enviar mensajes desde el 5/03 al 5/05, con una instrucción MSG se envía el mensaje vía serial DF1 al nodo 0, y el módulo ENIW lo direcciona a la IP asignada al nodo, en este caso sería la dirección IP del 5/05.

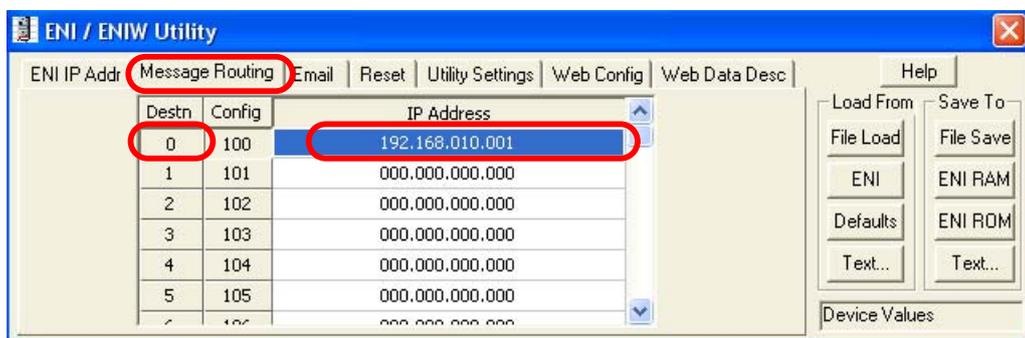


Figura. 4.100. Pantalla de configuración 4 ENIW

5. Dar clic en la pestaña **Email** y configurar los siguientes parámetros:
  - Email Server IP Address: Ingresar la dirección del servidor de correo de la red ethernet en la que se encuentra conectado el módulo.
  - "From": Escriba la dirección de correo que se desea que aparezca en el email como dirección de origen..
  - "To": Ingresar la dirección de correo de destino, ésta se asigna a un número de nodo (Destn) al cual se debe enviar el mensaje via serial para que el módulo ENIW lo envíe como email.

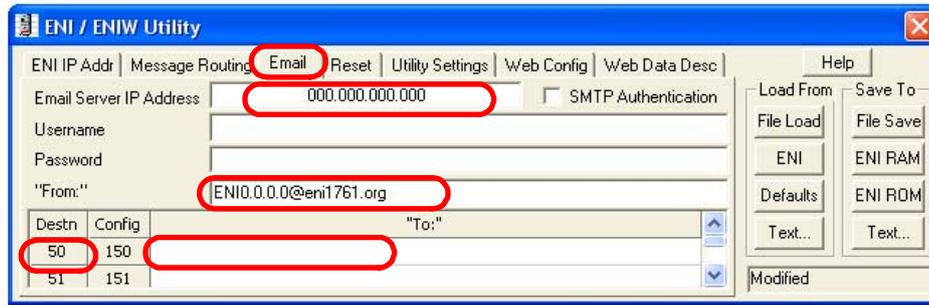


Figura. 4.101. Pantalla de configuración 5 ENIW

6. Finalmente, para grabar la configuración en el módulo, presione sobre **ENI ROM**, con lo que empezará el proceso de grabación

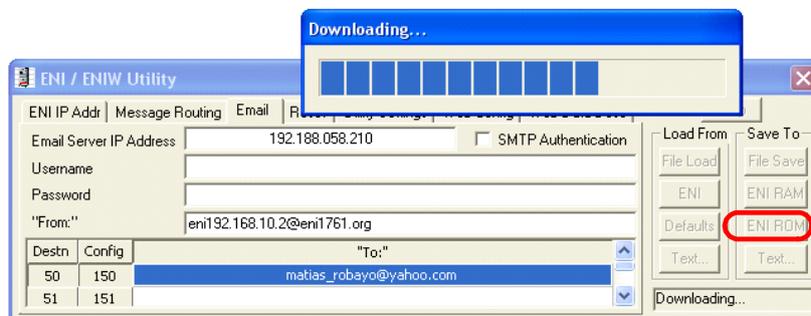


Figura. 4.102. Pantalla de configuración 6 ENIW

Para verificar la configuración del módulo, se puede abrir la ventana RSWho del software RSLinx (previamente configurado el driver Ethernet con la dirección del módulo y de la PC) y se verá la siguiente pantalla:

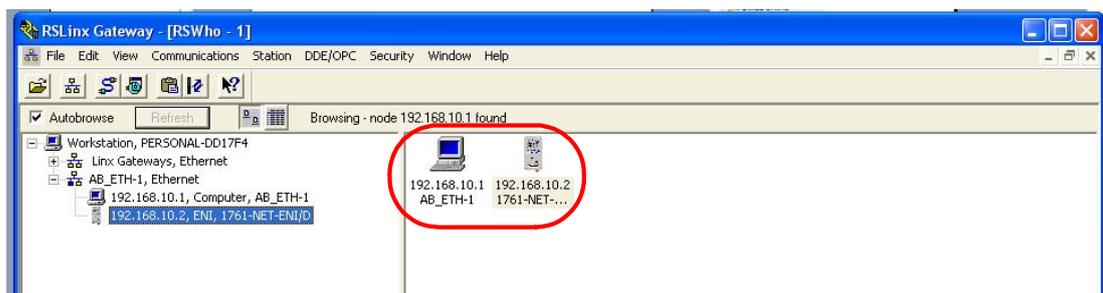


Figura. 4.103. Verificación de la dirección IP del módulo NET-ENIW

Posteriormente, cuando ya se configure el PLC 5/03 ó Micrologix, y se lo conecte al puerto serial del módulo, ya no aparecerá la imagen del módulo en esta pantalla, sino la del PLC.

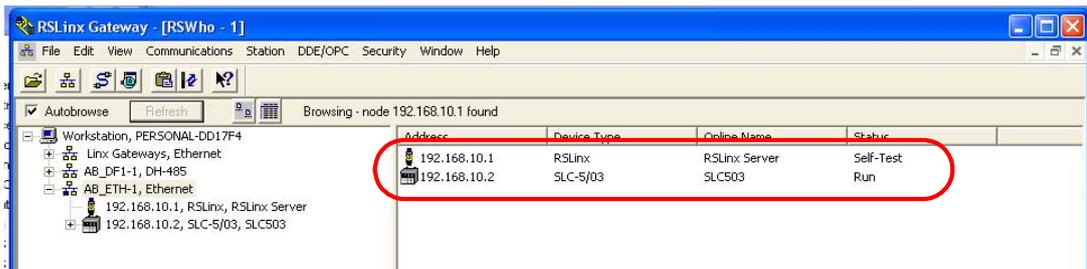


Figura. 4.104. Verificación de la conexión del módulo NET-ENIW y el PLC SLC 5/03

#### 4.18.4. PLANTEAMIENTO

Se desea conectar los siguientes equipos en la configuración mostrada en la figura 4.105.

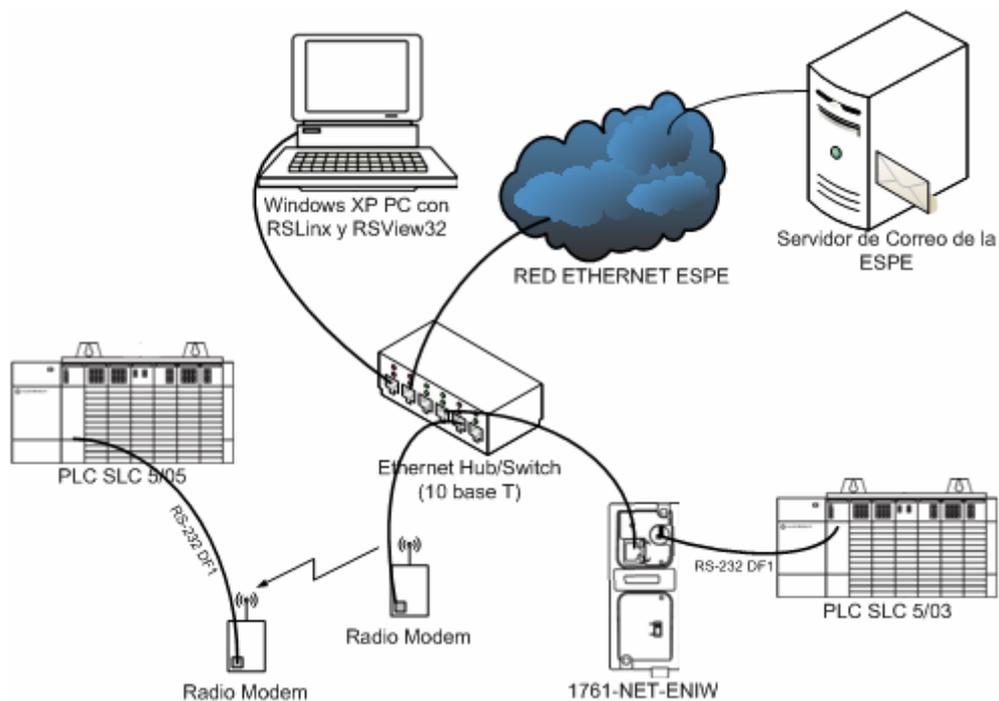


Figura. 4.105. Planteamiento Práctica #18 (1)

El computador personal deberá poseer una HMI (realizada en RSView32) que permita monitorear las Entradas Discretas y controlar las Salidas Discretas de los tres PLC's.

La HMI deberá contar con 3 pantallas con los siguientes esquemas:

#### PANTALLA PRINCIPAL

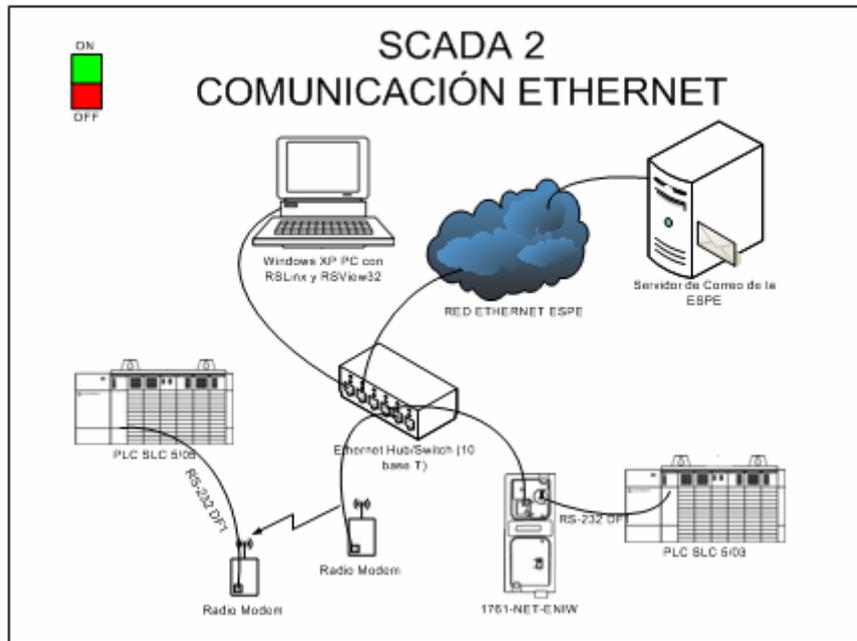


Figura. 4.106. Planteamiento Práctica #18 (2)

Al hacer clic con el Mouse sobre los PLC's, se desplegará la pantalla respectiva que permita monitorear y controlar las entradas y Salidas Discretas de dicho PLC. A continuación de muestra el esquema de cada pantalla.

**SLC 5/05**

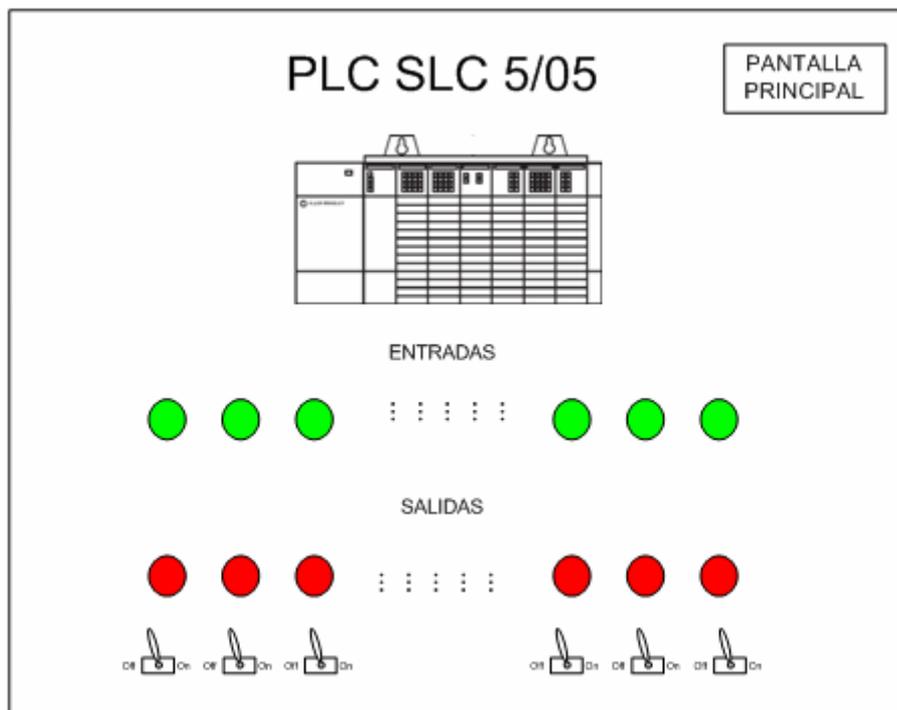
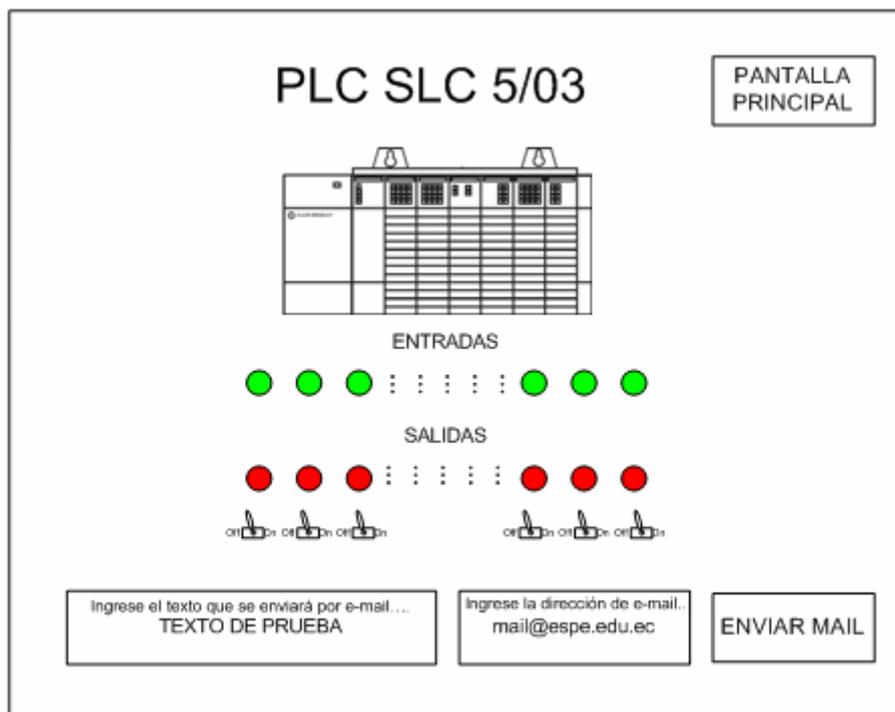


Figura. 4.107. Planteamiento Práctica #18 (3)

**SLC 5/03**

**Figura. 4.108. Planteamiento Práctica #18 (4)**

En esta pantalla, se incluirá la opción de envío de e-mail, a través del módulo 1761-NET-ENIW. Para lo que se requiere conocer la dirección IP del servidor de e-mail de la ESPE.

**Nota:** El número de entradas y salidas dependerá del PLC y los módulos utilizados.

#### **4.18.5. PROCEDIMIENTO**

Para la realización de esta práctica el procedimiento a seguir es el siguiente:

1. Configurar el Hyper Terminal para conectarse a los Radio Modems
2. Configurar los Radio Modems para que se enlacen entre si.
3. Configurar el Driver Serial en RSLinx para comunicarse con el PLC SLC 5/05
4. Abrir el software RSLogix500, crear un nuevo archivo y configurar el PLC SLC 5/05, sus módulos y el canal de comunicación ethernet.
5. Realizar el programa Ladder que cumpla con los requerimientos definidos en el planteamiento de la práctica.
6. Transferir el programa.
7. Configurar el Driver Serial en RSLinx para comunicarse con el PLC SLC 5/03

8. Abrir el software RSLogix500, crear un nuevo archivo y configurar el PLC SLC 5/03, sus módulos y el canal de comunicación serial.
9. Realizar el programa Ladder que cumpla con los requerimientos definidos en el planteamiento de la práctica.
10. Transferir el programa.
11. Configurar el módulo de interfase ethernet y conectarlo al plc SLC 5/03
12. Configurar la Dirección IP de la PC
13. Conectar la PC, PLC's, módulo 1761-NET-ENIW, Radio Modems y el Switch Ethernet de acuerdo a la configuración planteada
14. Configurar el Driver Ethernet en RSLinx que permita conectarse con los PLC's
15. Abrir el software RSView32, configurar los canales de comunicación, configurar los nodos de comunicación.
16. Crear la interfase HMI que interactúe con los PLC's de cada estación de control.
17. Correr los PLC's y la HMI, probar si la lógica del programa de los PLC's y la HMI desarrollada funcionan correctamente, de acuerdo al problema planteado.

#### **4.18.6. PRUEBAS Y RESULTADOS**

#### **4.18.7. CONCLUSIONES Y RECOMENDACIONES**

## CAPITULO 5

# IMPLEMENTACIÓN DE LOS EXPERIMENTOS DEL SISTEMA DE ENTRENAMIENTO

### 5.1. CONTROL LÓGICO 1: INSTRUCCIONES DE BITS 1

#### 5.1.1. OBJETIVOS

- Familiarizarse con RSLogix500 (software de programación) y con RSLinx (software de comunicación)
- Utilizar las entradas y salidas discretas del PLC
- Familiarizarse con el direccionamiento lógico para los elementos con imagen de entradas o salidas.
- Programar aplicaciones utilizando instrucciones de bits

#### 5.1.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 ó Micrologix.
- ✓ 1 Módulo de Entradas Discretas\*.
- ✓ 1 Módulo de Salidas Discretas\*.
- ✓ 1 Computador Personal, con instalación de RSLogix500 y RSLinx
- ✓ 1 Tablero Didáctico de Simulación AB-01
- ✓ 1 Fuente de alimentación +24 VDC
- ✓ Cables de Conexión

(\*No se requiere si se utiliza un PLC de la serie Micrologix)

#### 5.1.3. PLANTEAMIENTO

Se tienen tres motores como se muestra en la figura 5.1., cada uno con un botón de START y otro de STOP. Se desea que la secuencia de encendido sea: Primero el motor M1, luego el motor M2 y finalmente el motor M3. Mientras que para la secuencia de apagado se desea que el primero en desactivarse sea el motor M3, luego el motor M2 y finalmente el motor M1. No se podrá activar o desactivar los motores, en una secuencia diferente a la antes mencionada.

1. Desarrolle el programa Ladder utilizando únicamente como instrucción de salida la instrucción OTE.
2. Desarrolle el programa Ladder utilizando únicamente como instrucciones de salida las instrucciones OTL y OTU.

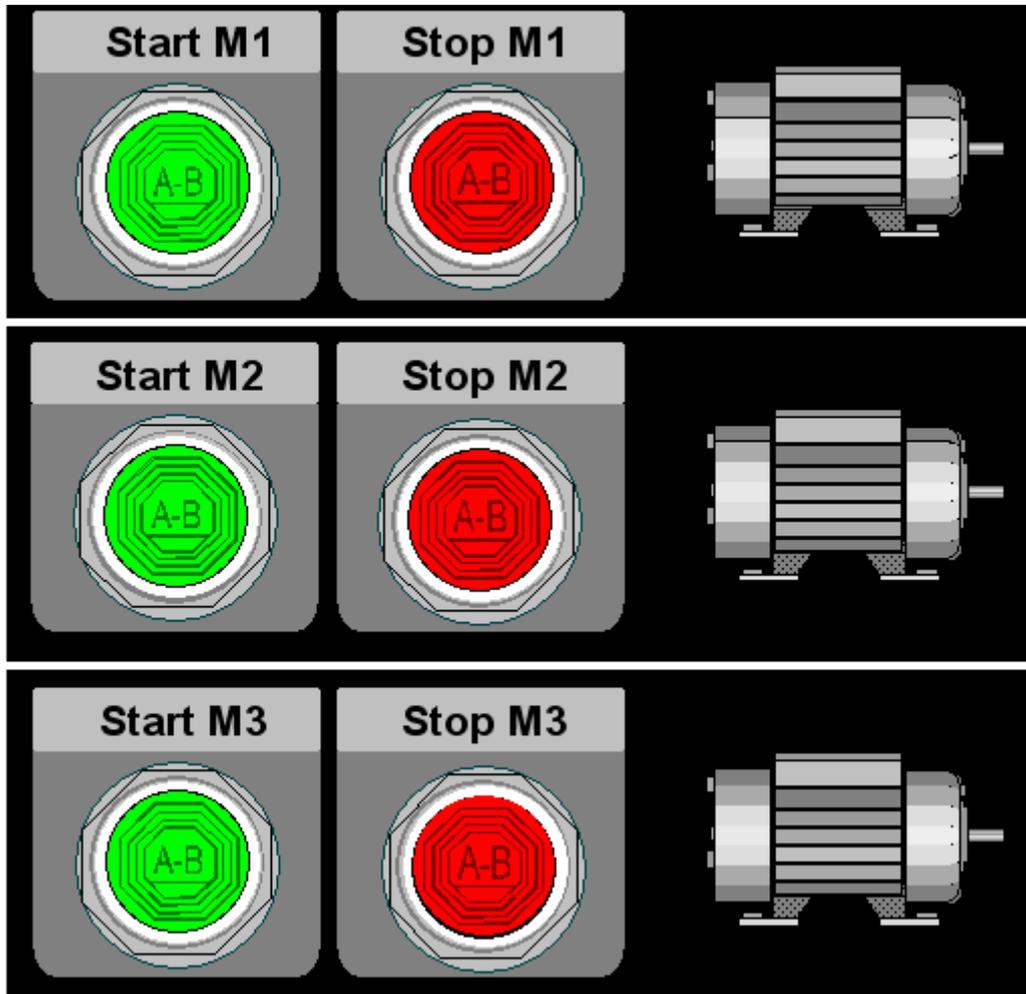


Figura. 5.1. Planteamiento Práctica #1

#### 5.1.4. DESARROLLO

1. Configuración de driver en RSLinx

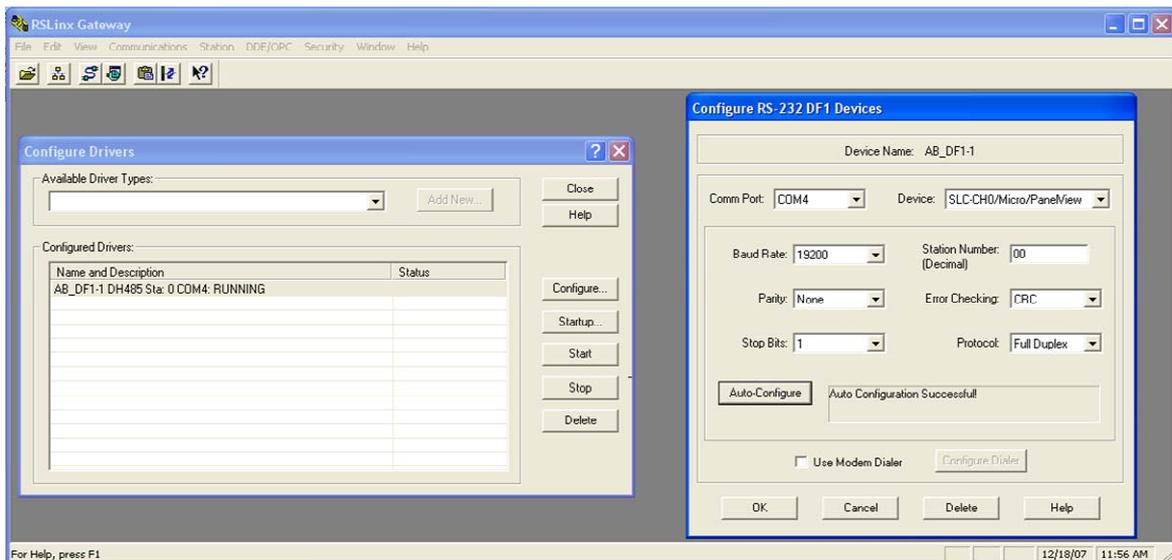


Figura. 5.2. Desarrollo Práctica #1 (1)

## 2. Creación del archivo y configuración del PLC en el software RSLogix500

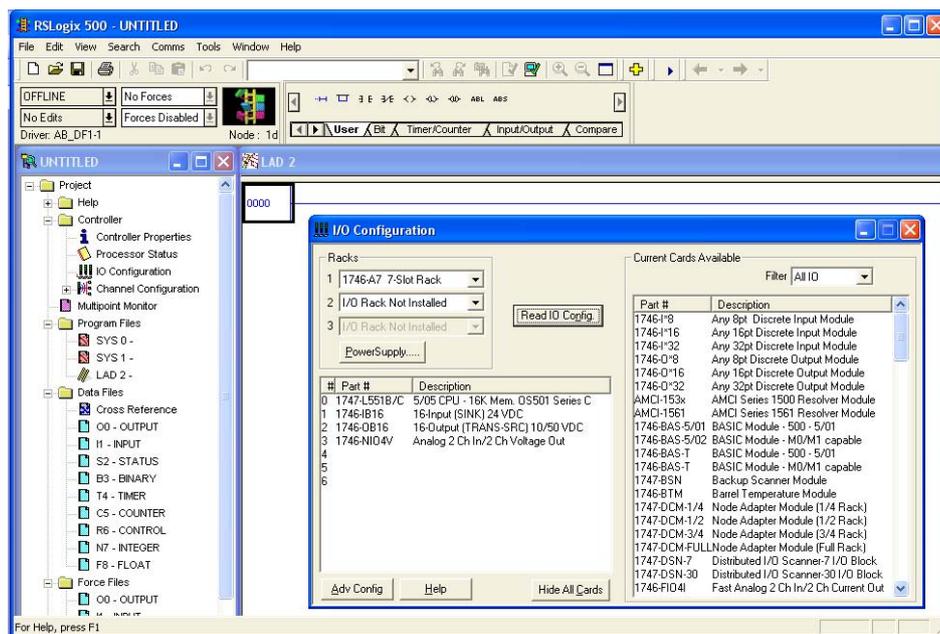


Figura. 5.3. Desarrollo Práctica #1 (2)

## 3. Programación Ladder

- a. Programa Ladder que cumple con la lógica de control planteada en esta práctica (primera parte), utilizando únicamente como instrucción de salida la instrucción OTE.

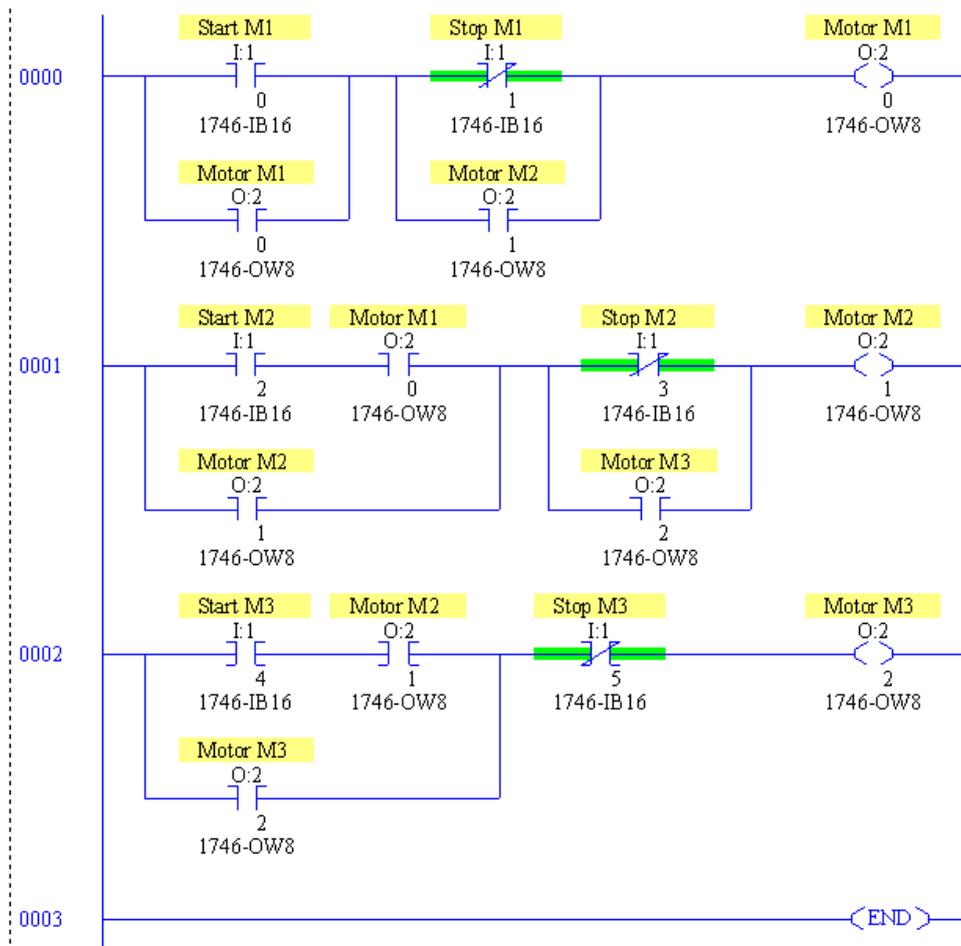


Figura. 5.4. Desarrollo Práctica #1 (3)

- b. Programa Ladder que cumple con la lógica de control planteada en esta práctica (segunda parte), utilizando únicamente como instrucciones de salida las instrucciones OTL y OTU.

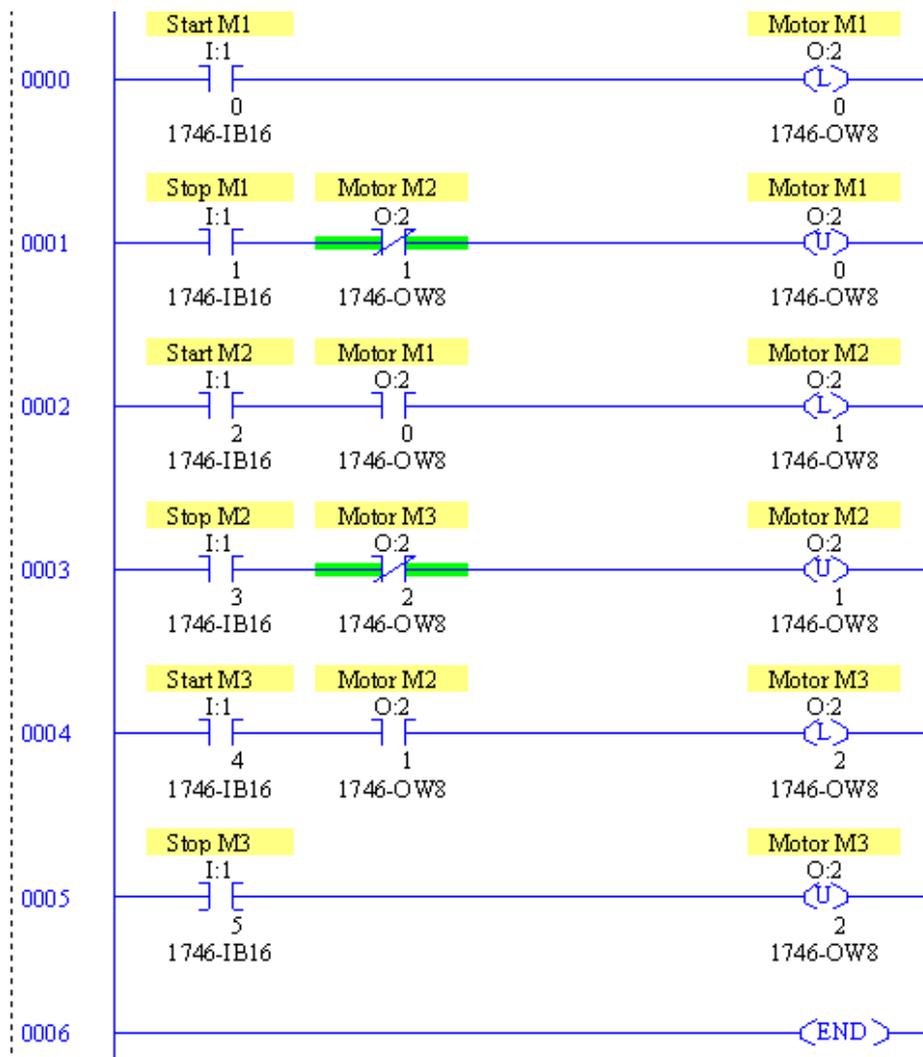


Figura. 5.5. Desarrollo Práctica #1 (4)

4. Transferir el programa.
5. En modo ONLINE se realizó las respectivas pruebas para verificar que el programa realizado cumpla satisfactoriamente con la lógica planteada en esta práctica. A continuación se presenta una pantalla demostrativa de esta verificación, en la que se puede ver que a pesar de que se activó la entrada Start M3, el motor M3 no se enciende puesto de acuerdo a la secuencia de encendido, primero se debe encender el motor M2.

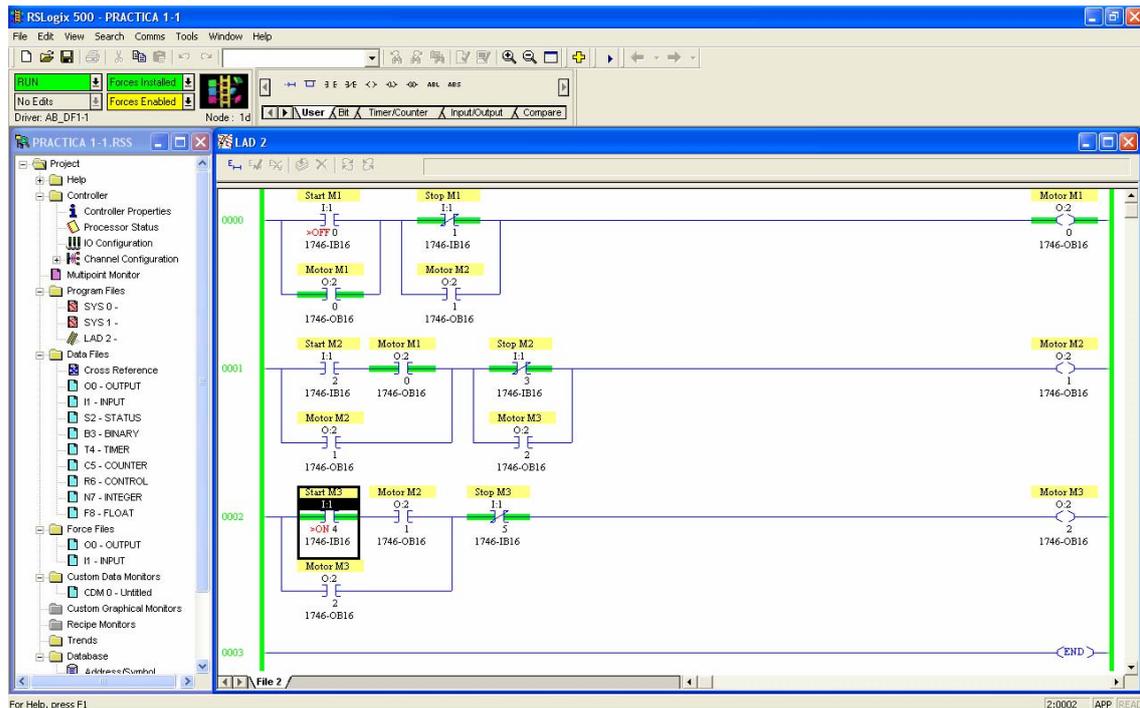


Figura. 5.6. Desarrollo Práctica #1 (5)

6. La conexión de las Entradas y Salidas del PLC al Tablero de simulación correspondiente, se la hizo de acuerdo a la asignación de éstas en el programa:

Nombre	Tipo	Dirección
Start M1	Entrada Discreta	I:1/0
Stop M1	Entrada Discreta	I:1/1
Start M2	Entrada Discreta	I:1/2
Stop M2	Entrada Discreta	I:1/3
Start M3	Entrada Discreta	I:1/4
Stop M3	Entrada Discreta	I:1/5
Motor M1	Salida Discreta	O:2/0
Motor M2	Salida Discreta	O:2/1
Motor M3	Salida Discreta	O:2/2

Tabla. 5.1. Direcciones utilizadas Práctica #1

7. Se verificó que el PLC trabaja correctamente de acuerdo al planteamiento de la práctica, con ayuda del correspondiente tablero didáctico de simulación.

### **5.1.5. PRUEBAS Y RESULTADOS**

Durante el desarrollo de la práctica se realizaron las pruebas respectivas tanto en software (Modo ONLINE del PLC) como en hardware (Tableros didácticos de simulación), a través de las cuales se verificó que el PLC trabaja correctamente de acuerdo a la lógica del problema planteado en esta práctica.

### **5.1.6. CONCLUSIONES Y RECOMENDACIONES**

#### **Conclusiones**

- Los paquetes de software RSLogix500 y RSLinx permiten conectarse fácilmente a los PLC SLC500 y Micrologix
- Las instrucciones OTE, OTL y OTU permiten controlar fácilmente el estado de las Salidas Discretas, de acuerdo a la lógica de nuestro programa.
- El formato de direccionamiento lógico para los elementos con imagen de entradas o salidas facilita la programación y posterior revisión de la lógica ladder puesto que sintetiza toda la información que describe a dicha imagen.

#### **Recomendaciones**

- Para evitar provocar daños al PLC por una mala programación, se recomienda siempre verificar primero la lógica de programación a través del modo en línea (ONLINE), y posteriormente realizar las pruebas en Hardware.
- Para facilitar la programación y posterior revisión del programa, se recomienda asignar una pequeña descripción a cada dirección que se utilice, la misma que aparece cada vez que se emplee dicha dirección.

## 5.2. CONTROL LÓGICO 2: INSTRUCCIONES DE BITS 2

### 5.2.1. OBJETIVOS

- Utilizar las entradas y salidas discretas del PLC
- Programar aplicaciones de control lógico utilizando instrucciones de bits.

### 5.2.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 ó Micrologix.
- ✓ 1 Módulo de Entradas Discretas\*.
- ✓ 1 Módulo de Salidas Discretas\*.
- ✓ 1 Computador Personal, con instalación de RSLogix500 y RSLinx
- ✓ 1 Tablero Didáctico de Simulación AB-02
- ✓ 1 Fuente de alimentación +24 VDC
- ✓ Cables de Conexión

(\*No se requiere si se utiliza un PLC de la serie Micrologix)

### 5.2.3. PLANTEAMIENTO

Se tiene una tolva que almacena asfalto, como se indica en la figura 5.7. Esta posee dos motores, el motor *M1* o de *Llenado* que alimenta de material a la tolva y el motor *M2* o de *Vaciado* que permite el desalojo del material. También se dispone de tres detectores electromecánicos de nivel tipo boya (con contacto normalmente abierto) *S1*, *S2* y *S3* instalados en la tolva como se indica en la figura 5.7.

Se desea que la tolva pueda operar en dos modos: Manual y Automático. En el Modo Manual, el operador puede activar y desactivar los motores de acuerdo a su conveniencia, mientras que en el Modo Automático, la operación se realizará de acuerdo a las siguientes condiciones:

El motor *M1* se enciende cuando el nivel del material está por debajo de *S2* y se apaga cuando el material alcanza el nivel de *S3*.

El motor M2 se enciende cuando el material alcanza el nivel de S2 y se apaga cuando el material está por debajo de S1.

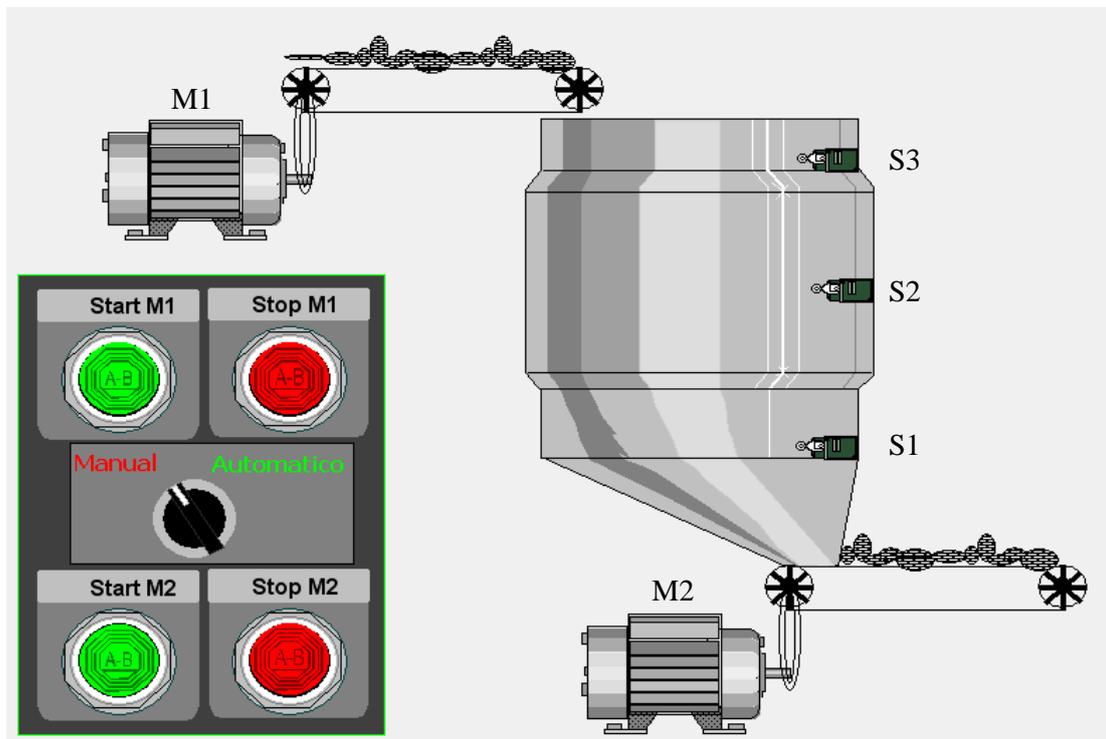


Figura. 5.7. Planteamiento Práctica #2

## 5.2.4. DESARROLLO

### 1. Configuración de driver en RSLinx

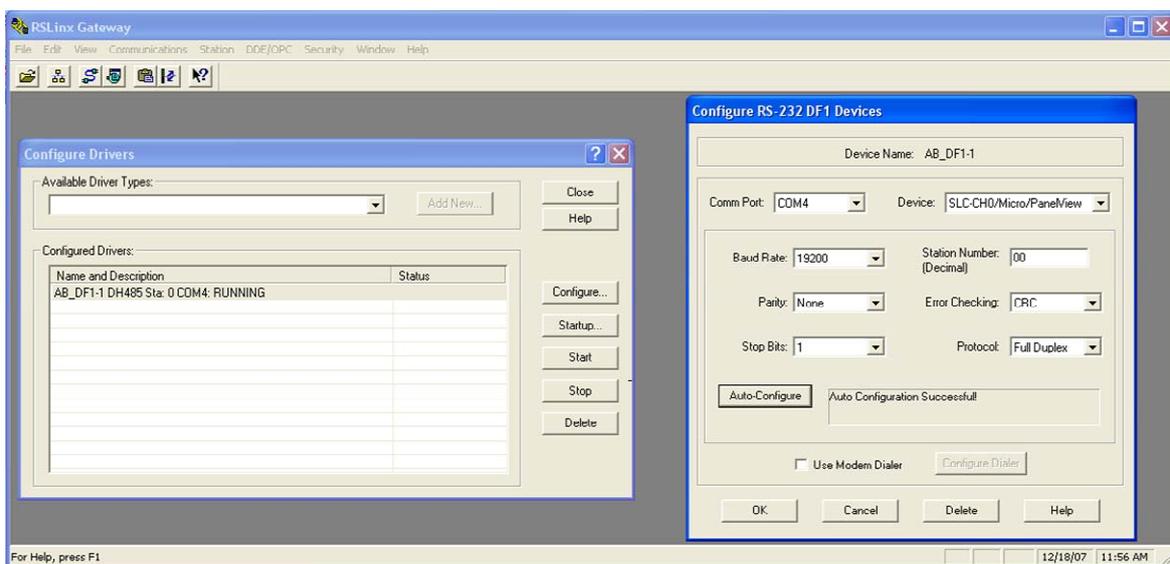


Figura. 5.8. Desarrollo Práctica #2 (1)

## 2. Creación del archivo y configuración del PLC en el software RSLogix500

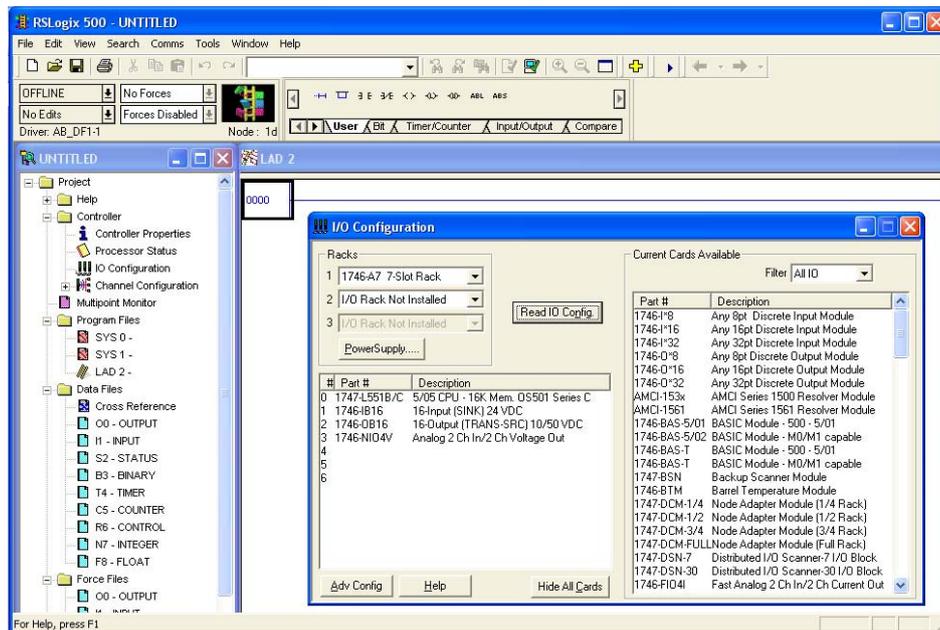


Figura. 5.9. Desarrollo Práctica #2 (2)

## 3. La programación Ladder que cumple con la lógica de control planteada en esta práctica es la siguiente:

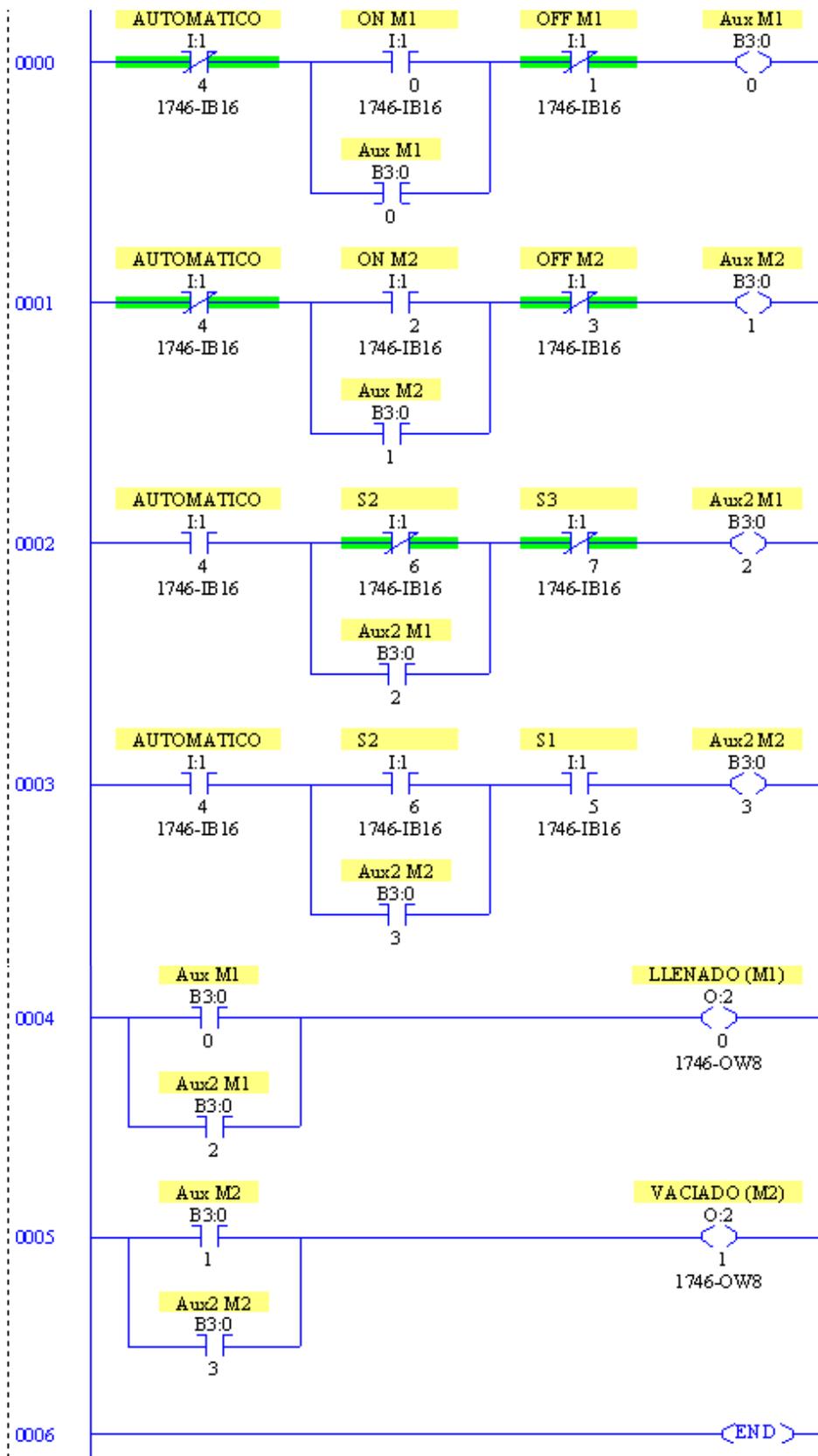


Figura. 5.10. Desarrollo Práctica #2 (3)

4. Transferir el programa.
5. En modo ONLINE se realizó las respectivas pruebas para verificar que el programa realizado cumpla satisfactoriamente con la lógica planteada en esta práctica. A continuación se presenta una pantalla demostrativa de esta verificación, en la que se puede ver que ambos motores (LLENADO Y VACIADO) se encuentran activados puesto que los sensores de nivel S1 y S2 se encuentran activados. Lo cual coincide con la lógica planteada en esta práctica.

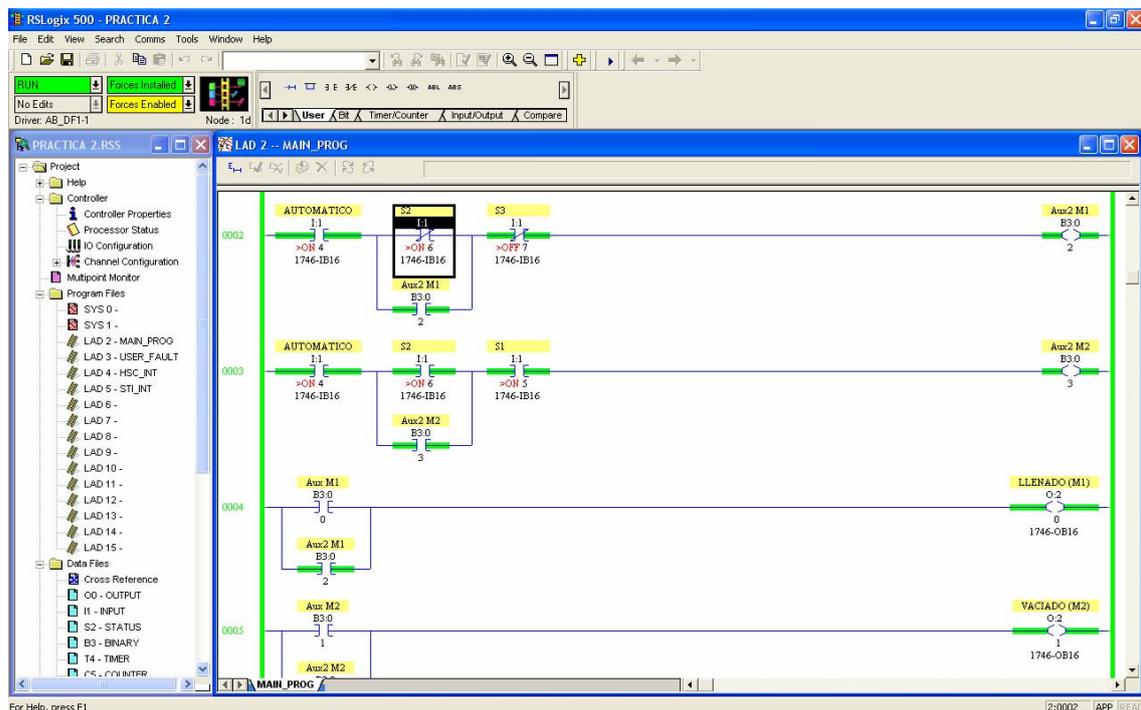


Figura. 5.11. Desarrollo Práctica #2 (4)

6. La conexión de las Entradas y Salidas del PLC al Tablero de simulación correspondiente, se la hizo de acuerdo a la asignación de éstas en el programa:

Nombre	Tipo	Dirección
ON M1	Entrada Discreta	I:1/0
OFF M1	Entrada Discreta	I:1/1
ON M2	Entrada Discreta	I:1/2
OFF M2	Entrada Discreta	I:1/3
AUTOMÁTICO	Entrada Discreta	I:1/4
S1	Entrada Discreta	I:1/5

S2	Entrada Discreta	I:1/6
S3	Entrada Discreta	I:1/7
AUX1 M1	Bit	B3:0/0
AUX1 M2	Bit	B3:0/1
AUX2 M1	Bit	B3:0/2
AUX2 M2	Bit	B3:0/3
LLENADO (M1)	Salida Discreta	O:2/0
VACIADO (M2)	Salida Discreta	O:2/1

**Tabla. 5.2. Direcciones utilizadas Práctica #2**

7. Se verificó que el PLC trabaja correctamente de acuerdo al planteamiento de la práctica, con ayuda del correspondiente tablero didáctico de simulación.

### **5.2.5. PRUEBAS Y RESULTADOS**

Durante el desarrollo de la práctica se realizaron las pruebas respectivas tanto en software (Modo ONLINE del PLC) como en hardware (Tableros didácticos de simulación), a través de las cuales se verificó que el PLC trabaja correctamente de acuerdo a la lógica del problema planteado en esta práctica.

### **5.2.6. CONCLUSIONES Y RECOMENDACIONES**

#### **Conclusiones**

- Las instrucciones de Entrada y Salida en la lógica de programación permiten interactuar a los PLC's con las señales externas para el Control de Procesos.

#### **Recomendaciones**

- Se recomienda no utilizar dos o más instrucciones de salida OTE asignadas a una misma de dirección.

### 5.3. CONTROL LÓGICO 3: CONTADORES

#### 5.3.1. OBJETIVOS

- Conocer la operación de las Instrucciones de Conteo.
- Programar aplicaciones de control utilizando Instrucciones de Conteo.

#### 5.3.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 ó Micrologix.
- ✓ 1 Módulo de Entradas Discretas\*.
- ✓ 1 Módulo de Salidas Discretas\*.
- ✓ 1 Computador Personal, con instalación de RSLogix500 y RSLinx
- ✓ 1 Tablero Didáctico de Simulación AB-03
- ✓ 1 Fuente de alimentación +24 VDC
- ✓ Cables de Conexión

(\*No se requiere si se utiliza un PLC de la serie Micrologix)

#### 5.3.3. PLANTEAMIENTO

En una banda transportadora de botellas de refrescos, como la que se ilustra en la figura 5.12, se dispone de dos sistemas detectores: uno de control de peso S1, que entregará una señal de activación ON cuando el peso del producto este fuera de una tolerancia establecida y otro de presencia del producto aprobado, a través del detector capacitivo S2. Desarrollar un programa para que, a través del PLC se realice la siguiente rutina de trabajo:

- Cada 12 productos aprobados, se activará un sistema elevador de cajas para el embalaje.
- Se contabilizará el número total de productos aprobados y embalados.
- Se contabilizará el número total de cajas embaladas
- Si el número de productos defectuosos es mayor a 20, se encenderá un foco de alarma, por lo que el sistema deberá tener un botón de reconocimiento de alarma que además encere el contador de productos defectuosos
- Es sistema cuenta con un pulsador para resetear todos los contadores.

El sistema contará con las siguientes señales de entrada:

Nombre	Descripción
RECHAZAR	Salida que activa el brazo que rechaza el producto
ALARMA	Salida que activa la alarma
ELEVADOR	Salida que activa el elevador de cajas
CARGAR EN CAJA	Salida que activa el brazo que coloca el producto aprobado en la caja
S1	Sensor de control de peso, que se activa cuando el producto se debe rechazar.
RECHAZADO	Señal de entrada que indica que el producto ya fue rechazado.
AKW ALARMA	Pulsador de reconocimiento de Alarma
S2	Sensor de presencia del producto aprobado, listo para ser cargado.
CARGADO	Señal de entrada que indica que el producto ya fue cargado.
FIN ELEVADOR	Señal de entrada que indica que el elevador ya regresó a su lugar, después de haber movido la caja.
ENCERAR CUENTAS	Pulsador para encerar los contadores.

Tabla. 5.3. Señales utilizadas Práctica #3

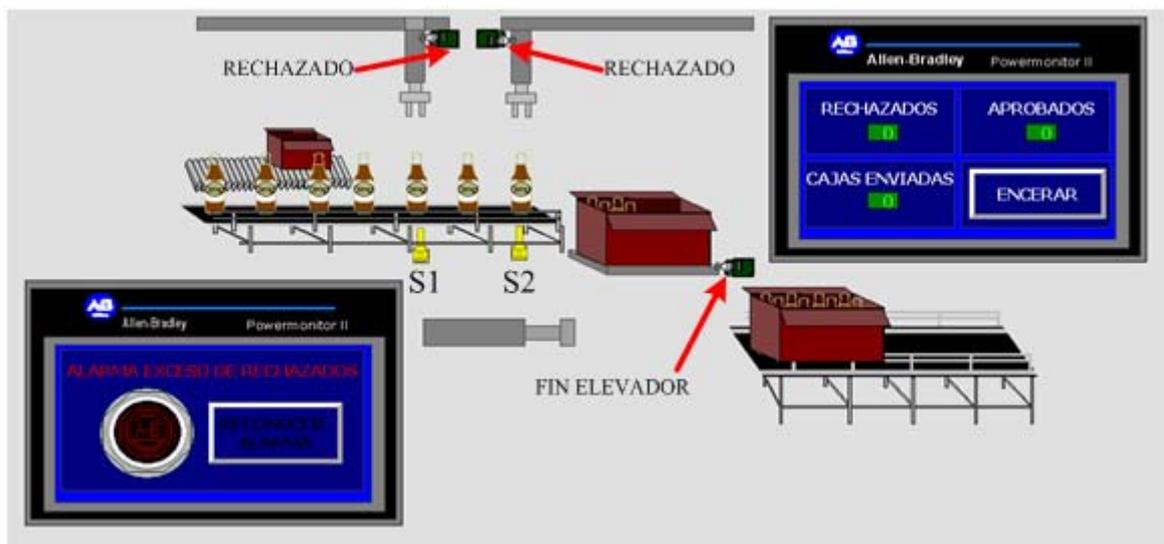


Figura. 5.12. Planteamiento Práctica #3

## 5.3.4. DESARROLLO

### 1. Configuración de driver en RSLinx

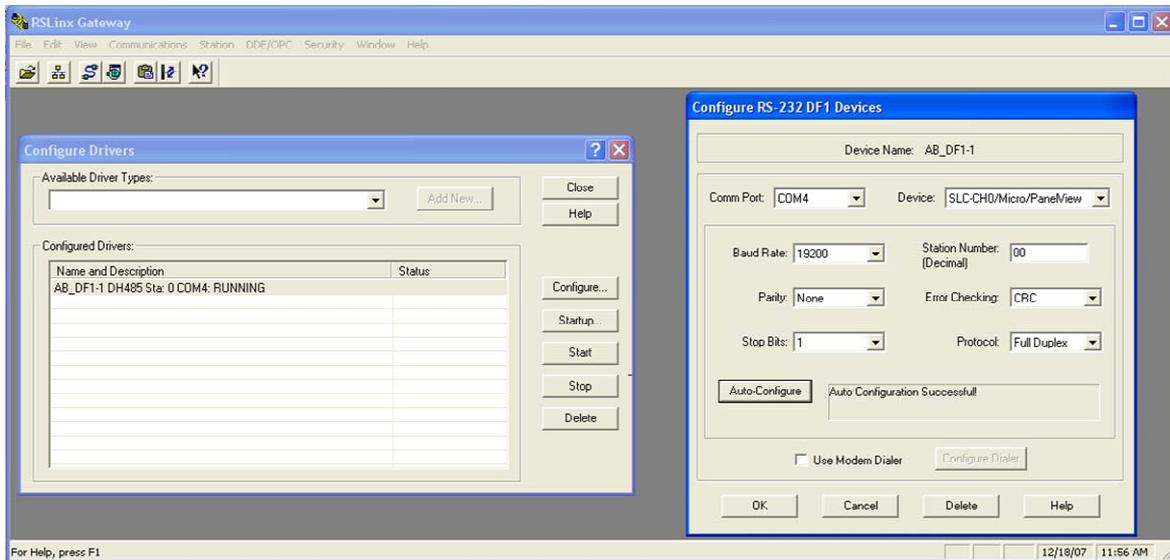


Figura. 5.13. Desarrollo Práctica #3 (1)

### 2. Creación del archivo y configuración del PLC en el software RSLogix500

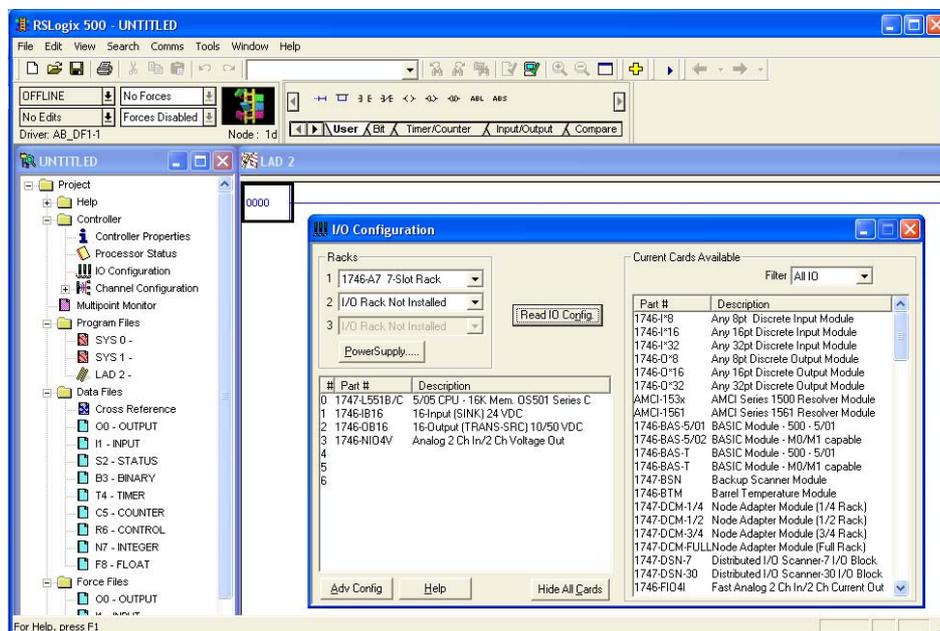


Figura. 5.14. Desarrollo Práctica #3 (2)

3. La programación Ladder que cumple con la lógica de control planteada en esta práctica es la siguiente:

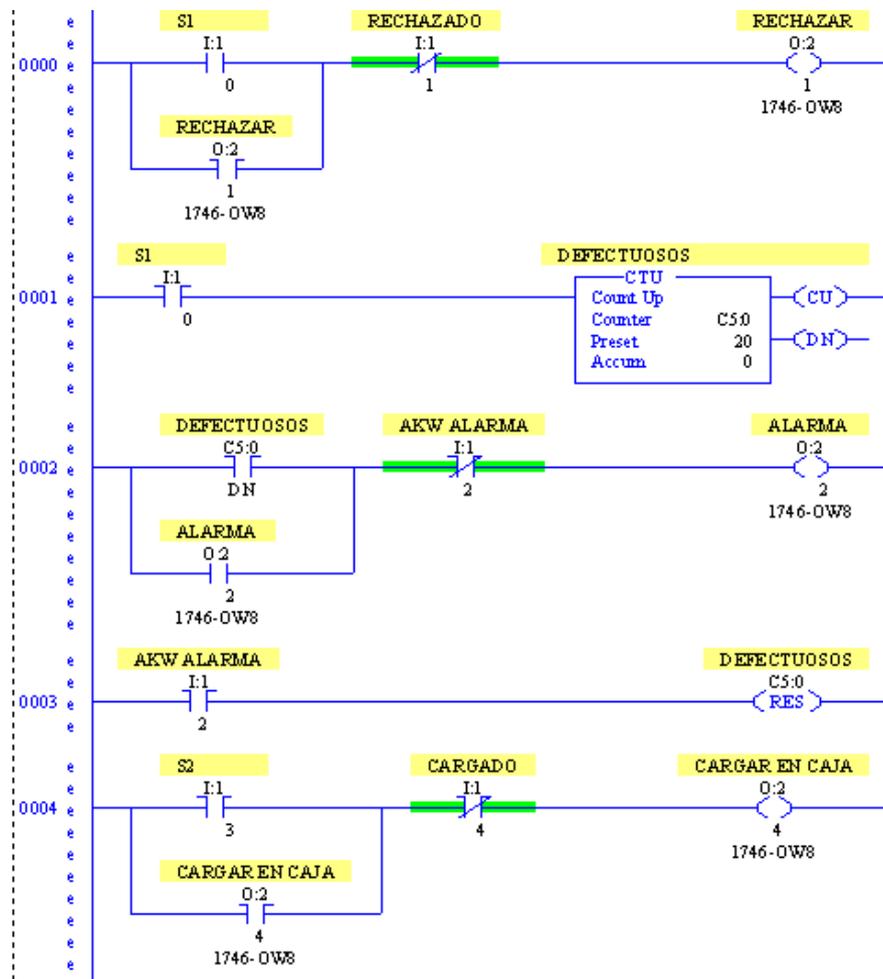


Figura. 5.15. Desarrollo Práctica #3 (3)

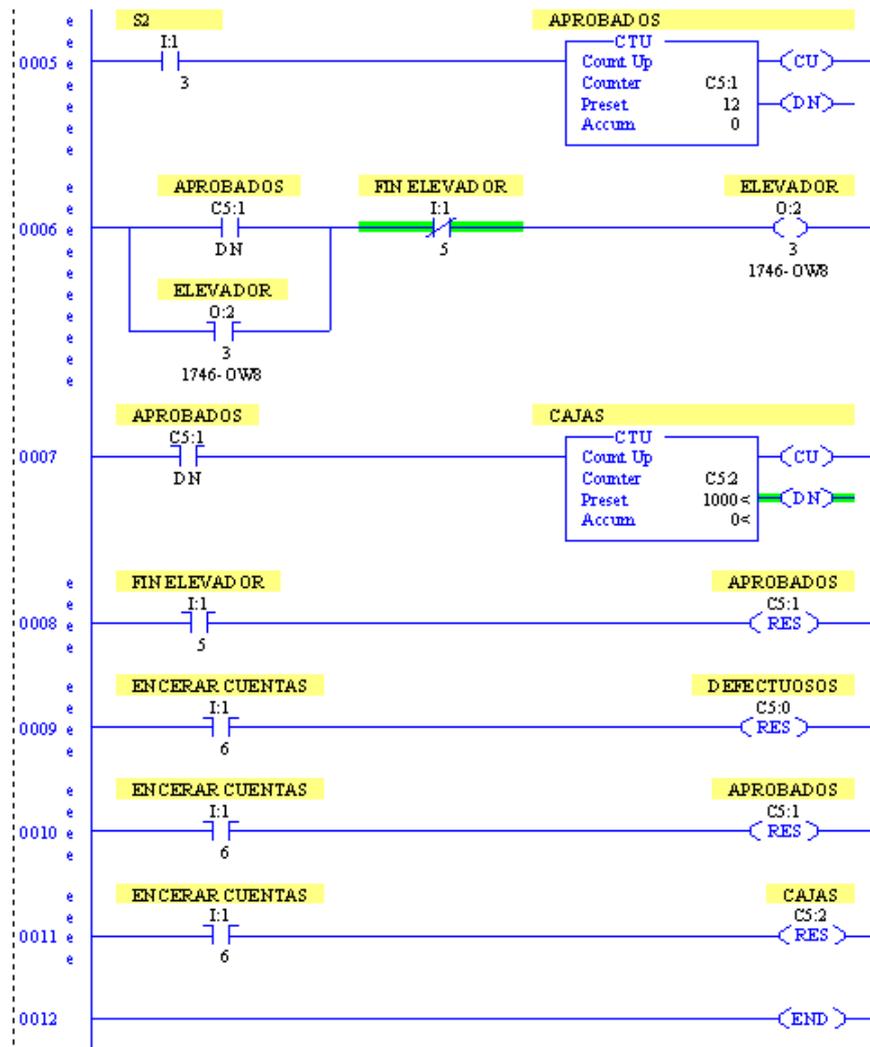


Figura. 5.16. Desarrollo Práctica #3 (4)

4. Transferir el programa.
5. En modo ONLINE se realizó las respectivas pruebas para verificar que el programa realizado cumpla satisfactoriamente con la lógica planteada en esta práctica. A continuación se presenta una pantalla demostrativa de esta verificación, en la que se puede ver que cuando se activa el sensor S2 de presencia del producto aprobado, el PLC activa la salida “CARGAR EN CAJA” e incrementa la cuenta de APROBADOS.

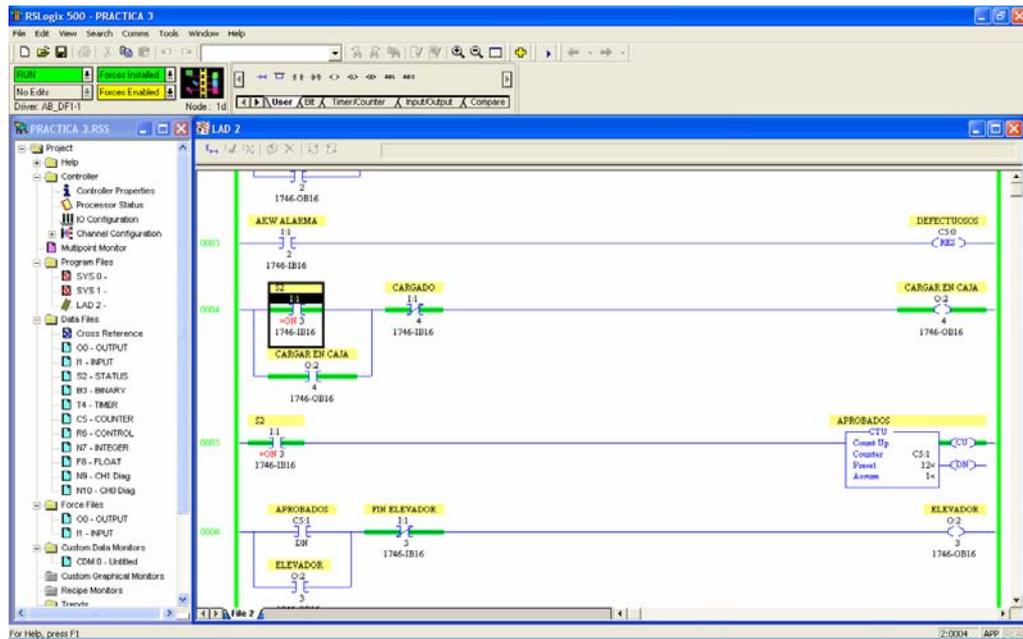


Figura. 5.17. Desarrollo Práctica #3 (5)

6. La conexión de las Entradas y Salidas del PLC al Tablero de simulación correspondiente, se la hizo de acuerdo a la asignación de éstas en el programa:

Nombre	Tipo	Dirección
RECHAZAR	Salida Discreta	O:2/1
ALARMA	Salida Discreta	O:2/2
ELEVADOR	Salida Discreta	O:2/3
CARGAR EN CAJA	Salida Discreta	O:2/4
S1	Entrada Discreta	I:1/0
RECHAZADO	Entrada Discreta	I:1/1
AKW ALARMA	Entrada Discreta	I:1/2
S2	Entrada Discreta	I:1/3
CARGADO	Entrada Discreta	I:1/4
FIN ELEVADOR	Entrada Discreta	I:1/5
ENCERAR CUENTAS	Entrada Discreta	I:1/6
DEFECTUOSOS	Contador	C5:0
APROBADOS	Contador	C5:1
CAJAS	Contador	C5:2

Tabla. 5.4. Direcciones utilizadas Práctica #3

7. Se verificó que el PLC trabaja correctamente de acuerdo al planteamiento de la práctica, con ayuda del correspondiente tablero didáctico de simulación.

### **5.3.5. PRUEBAS Y RESULTADOS**

Durante el desarrollo de la práctica se realizaron las pruebas respectivas tanto en software (Modo ONLINE del PLC) como en hardware (Tableros didácticos de simulación), a través de las cuales se verificó que el PLC trabaja correctamente de acuerdo a la lógica del problema planteado en esta práctica.

### **5.3.6. CONCLUSIONES Y RECOMENDACIONES**

#### **Conclusiones**

- El uso de instrucciones de conteo ascendente (CTU) o de conteo descendente (CTD) está determinado por los requerimientos planteados en la lógica de nuestra aplicación.

#### **Recomendaciones**

- La duración de la transición On-Off de las señales de entrada no debe ser menor a dos veces el tiempo de scan. (Asumiendo un ciclo de trabajo del 50%), para que las reconozcan las instrucciones de conteo
- Se recomienda reducir el valor de PRESET de los contadores, para facilitar la simulación en software (modo ONLINE), lo que reduce significativamente el tiempo necesario para la verificación de la lógica del programa.

## 5.4. CONTROL LÓGICO 4: TEMPORIZADORES 1

### 5.4.1. OBJETIVOS

- Conocer la operación de las Instrucciones de Temporización.
- Programar aplicaciones de control utilizando Instrucciones de Temporización.

### 5.4.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 ó Micrologix.
- ✓ 1 Módulo de Entradas Discretas\*.
- ✓ 1 Módulo de Salidas Discretas\*.
- ✓ 1 Computador Personal, con instalación de RSLogix500 y RSLinx
- ✓ 1 Tablero Didáctico de Simulación AB-04
- ✓ 1 Fuente de alimentación +24 VDC
- ✓ Cables de Conexión

(\*No se requiere si se utiliza un PLC de la serie Micrologix)

### 5.4.3. PLANTEAMIENTO

Programar e implementar un sistema de alarma contra incendios para una fábrica como el que se muestra en la figura 5.18, el cual tiene un panel de control que dispone de tres lámparas indicadoras L1, L2 y L3; además posee cuatro botones: uno de RESET y tres de ENTERADO (uno para cada área). Se cuenta con tres sensores/detectores de llama S1, S2 y S3, uno en cada área principal de la fábrica. En el instante en que cualquiera de los sensores se activa, la lámpara indicadora correspondiente se activa intermitentemente a una frecuencia de 1 Hz y suena una sirena; al presionar el botón de ENTERADO dicha luz dejara de titilar para quedar encendida permanentemente. Ninguna de las luces ni la sirena podrán ser apagadas con el botón de RESET si el PLC sigue recibiendo la señal de presencia de llama o si no se ha presionado previamente el botón de ENTERADO. La operación indebida del botón de RESET provocará que el sistema regrese al estado anterior.

Nota: para que funcionen intermitentemente las luces utilice únicamente 2 temporizadores. Considere también que ante incendios simultáneos en dos o más áreas las lámparas deben titilar sincrónicamente.



Figura. 5.18. Planteamiento Práctica #4

## 5.4.4. DESARROLLO

### 1. Configuración de driver en RSLinx

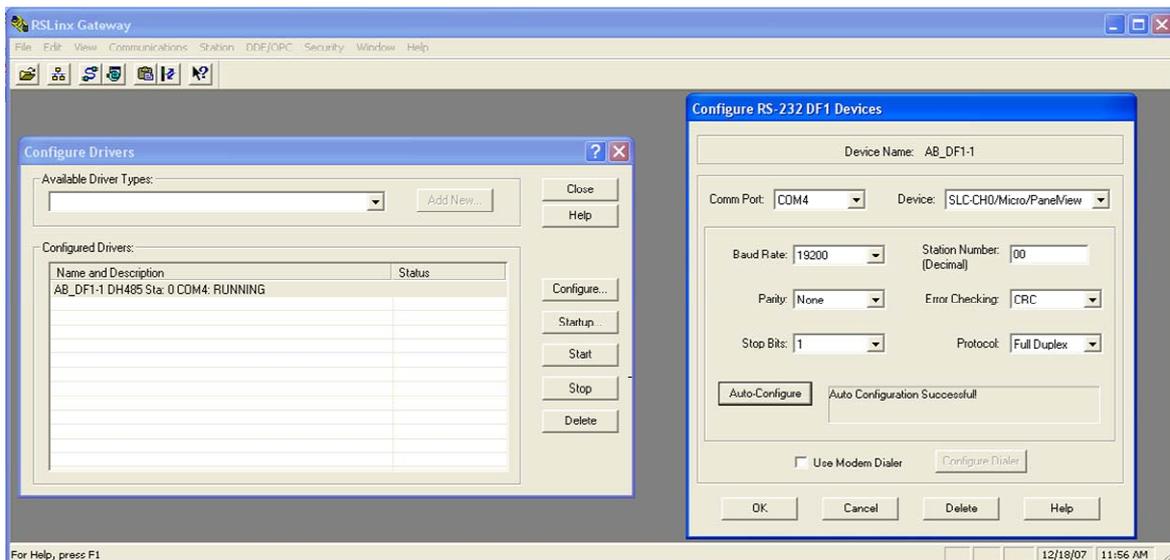


Figura. 5.19. Desarrollo Práctica #4 (1)

## 2. Creación del archivo y configuración del PLC en el software RSLogix500

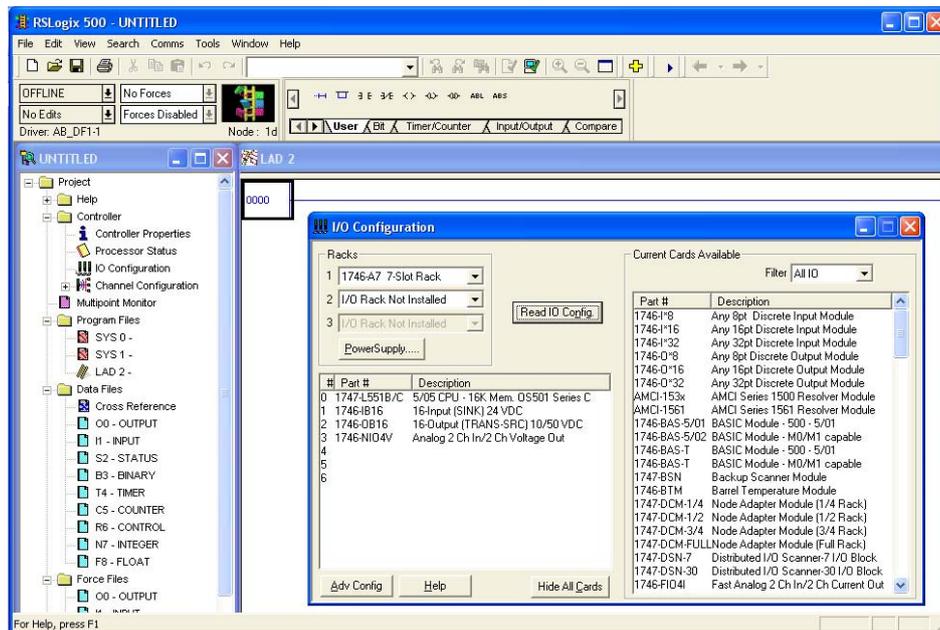


Figura. 5.20. Desarrollo Práctica #4 (2)

## 3. La programación Ladder que cumple con la lógica de control planteada en esta práctica es la siguiente:

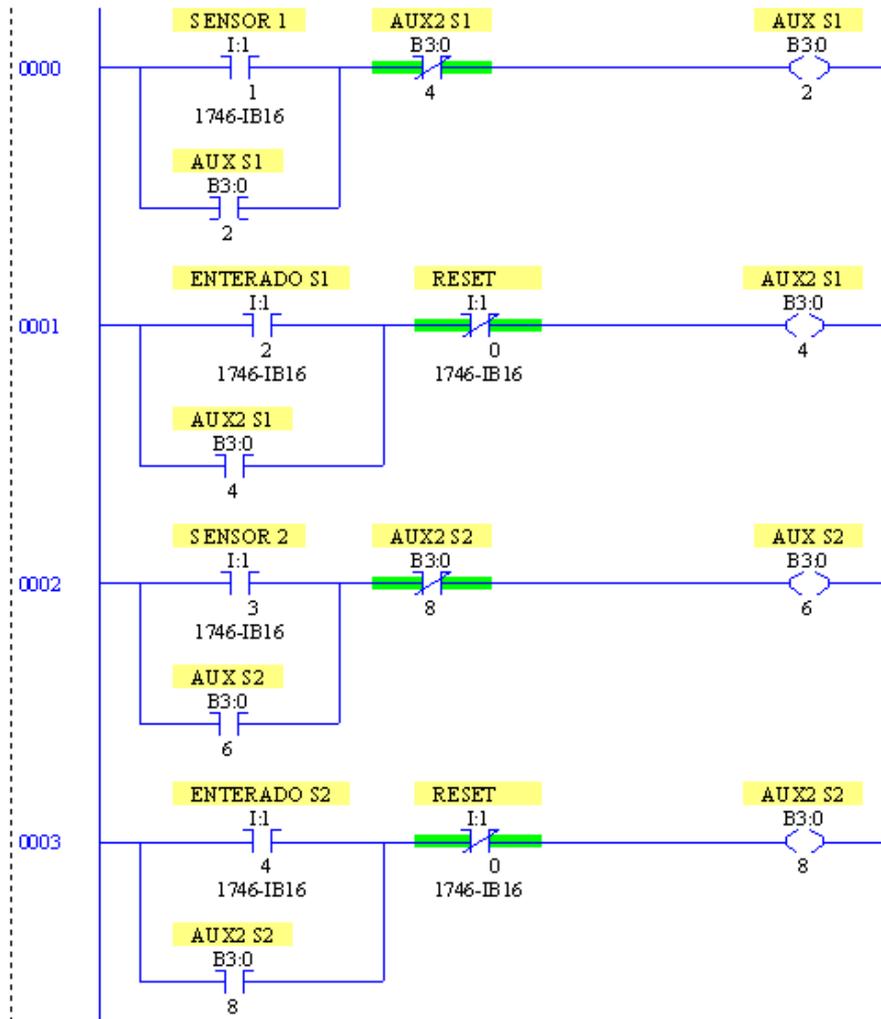


Figura. 5.21. Desarrollo Práctica #4 (3)

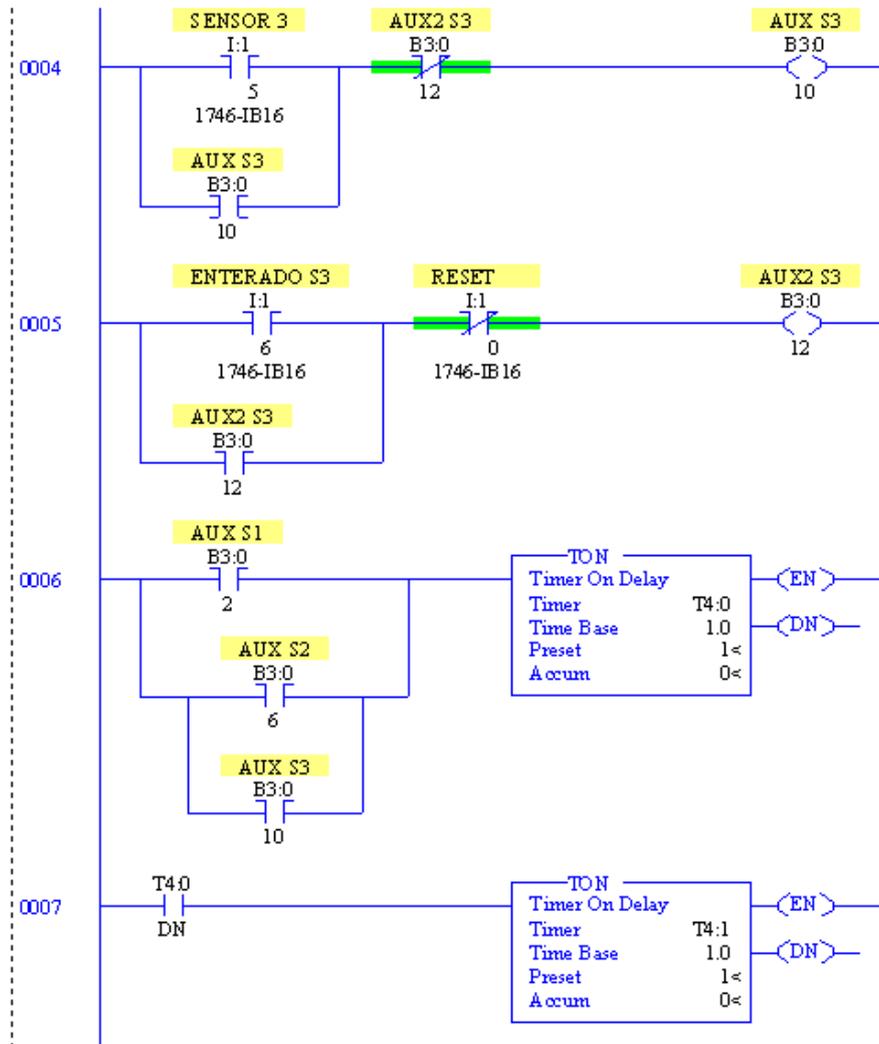


Figura. 5.22. Desarrollo Práctica #4 (4)

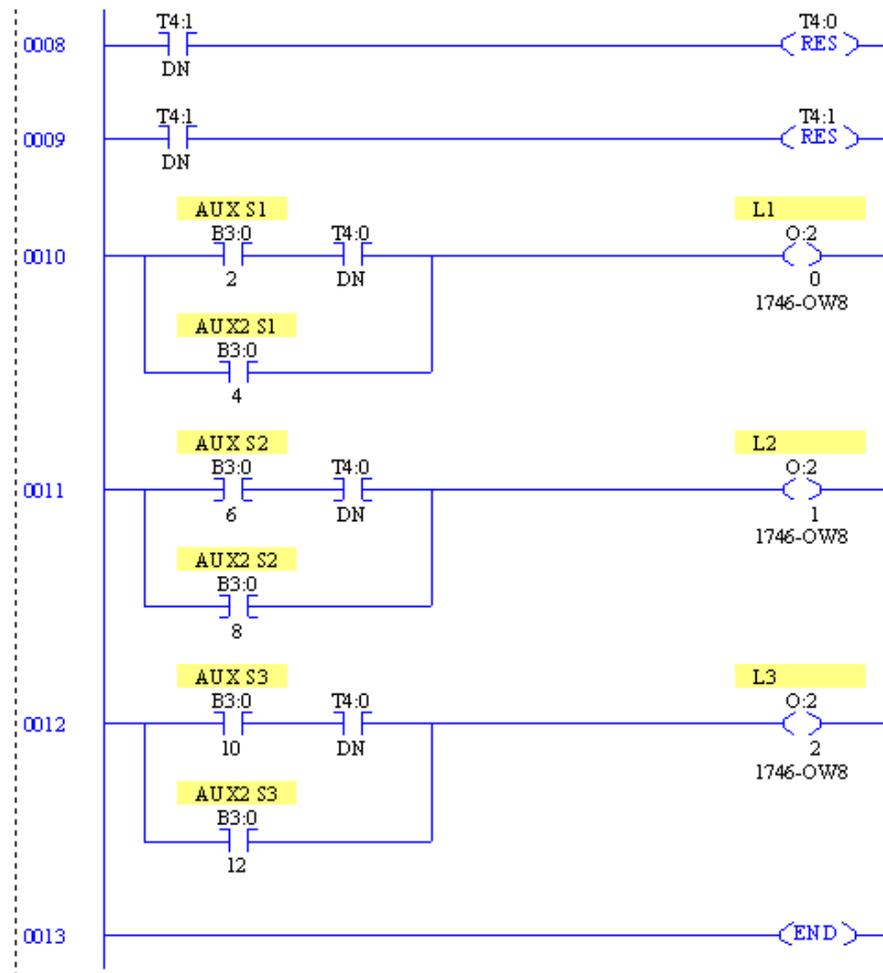


Figura. 5.23. Desarrollo Práctica #4 (5)

4. Transferir el programa.
5. En modo ONLINE se realizó las respectivas pruebas para verificar que el programa realizado cumpla satisfactoriamente con la lógica planteada en esta práctica. A continuación se presenta una pantalla demostrativa de esta verificación, en la que se puede ver que cuando se activa S3, el PLC activa el temporizador lo cual hace parpadear la luz L3

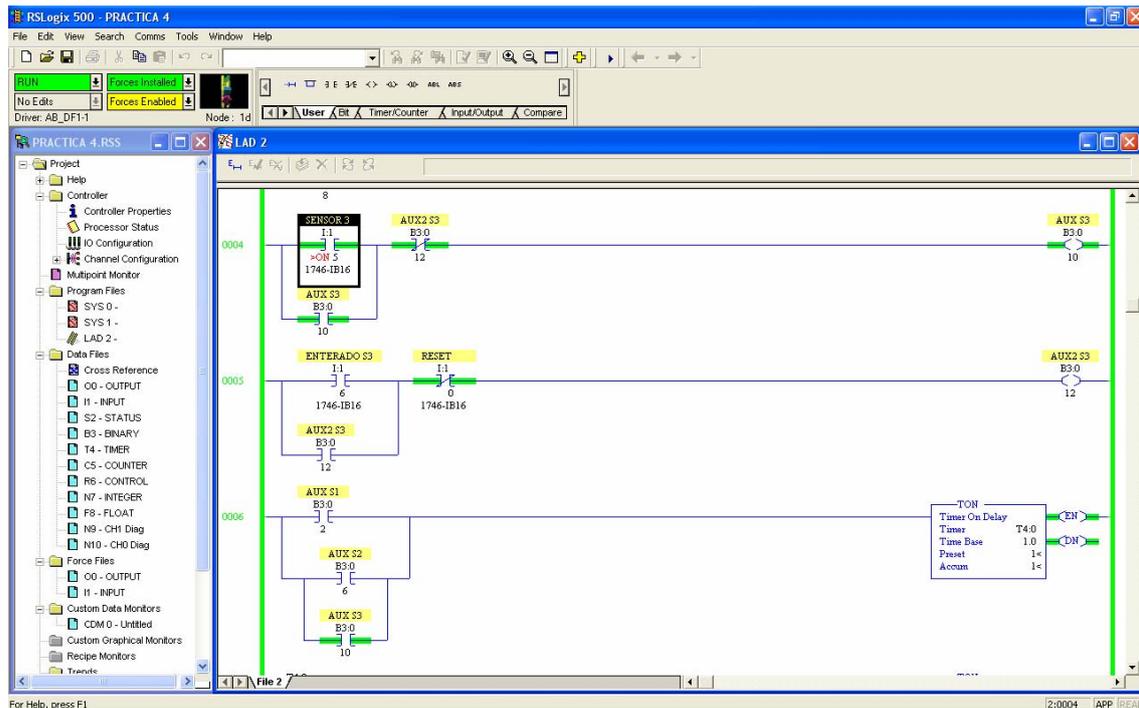


Figura. 5.24. Desarrollo Práctica #4 (6)

6. La conexión de las Entradas y Salidas del PLC al Tablero de simulación correspondiente, se la hizo de acuerdo a la asignación de éstas en el programa:

Nombre	Tipo	Dirección
L1	Salida Discreta	O:2/0
L2	Salida Discreta	O:2/1
L3	Salida Discreta	O:2/2
RESET	Entrada Discreta	I:1/0
SENSOR 1	Entrada Discreta	I:1/1
ENTERADO S1	Entrada Discreta	I:1/2
SENSOR 2	Entrada Discreta	I:1/3
ENTERADO S2	Entrada Discreta	I:1/4
SENSOR 3	Entrada Discreta	I:1/5
ENTERADO S3	Entrada Discreta	I:1/6
AUX S1	Bit	B3:0/2
AUX2 S1	Bit	B3:0/4
AUX S2	Bit	B3:0/6
AUX2 S2	Bit	B3:0/8

AUX S3	Bit	B3:0/10
AUX2 S3	Bit	B3:0/12

**Tabla. 5.5. Direcciones utilizadas Práctica #4**

7. Se verificó que el PLC trabaja correctamente de acuerdo al planteamiento de la práctica, con ayuda del correspondiente tablero didáctico de simulación.

#### **5.4.5. PRUEBAS Y RESULTADOS**

Durante el desarrollo de la práctica se realizaron las pruebas respectivas tanto en software (Modo ONLINE del PLC) como en hardware (Tableros didácticos de simulación), a través de las cuales se verificó que el PLC trabaja correctamente de acuerdo a la lógica del problema planteado en esta práctica.

#### **5.4.6. CONCLUSIONES Y RECOMENDACIONES**

##### **Conclusiones**

- Cada Temporizador esta formado por un archivo de datos de 3 palabras (16 bits cada palabra). La palabra 0 es una palabra de control que contiene los bits de Status del Temporizador. La palabra 1 contiene el valor de PRESET. La palabra 2 contiene el valor acumulado ACCUM.

##### **Recomendaciones**

- Se recomienda utilizar únicamente dos temporizadores para hacer parpadear todas las luces, lo que garantiza que éstas estén siempre sincronizadas.

## 5.5. CONTROL LÓGICO 5: TEMPORIZADORES 2

### 5.5.1. OBJETIVOS

- Familiarizarse con el direccionamiento lógico para las instrucciones de Temporización.
- Conocer el funcionamiento y aplicación de las Instrucciones de Temporización.
- Familiarizarse con el uso de los Bits de Status de las Instrucciones de Temporización.

### 5.5.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 ó Micrologix.
- ✓ 1 Módulo de Entradas Discretas\*.
- ✓ 1 Módulo de Salidas Discretas\*.
- ✓ 1 Computador Personal, con instalación de RSLogix500 y RSLinx
- ✓ 1 Tablero Didáctico de Simulación AB-05
- ✓ 1 Fuente de alimentación +24 VDC
- ✓ Cables de Conexión

(\*No se requiere si se utiliza un PLC de la serie Micrologix

### 5.5.3. PLANTEAMIENTO

Se desea controlar la apertura de una puerta corrediza de garaje como la que se muestra en la figura 5.25. En los extremos de la puerta se han ubicado dos interruptores de fin de carrera S2 y S3 que se activan cuando la puerta se abre o se cierra totalmente, respectivamente. La puerta se abre cuando el controlador recibe la señal enviada por un detector infrarrojo S1. Una vez que la puerta se abre totalmente, ésta permanece en ese estado por 5 segundos. Adicionalmente, el sistema cuenta con dos luces que indican si la puerta se está abriendo o cerrando, respectivamente.

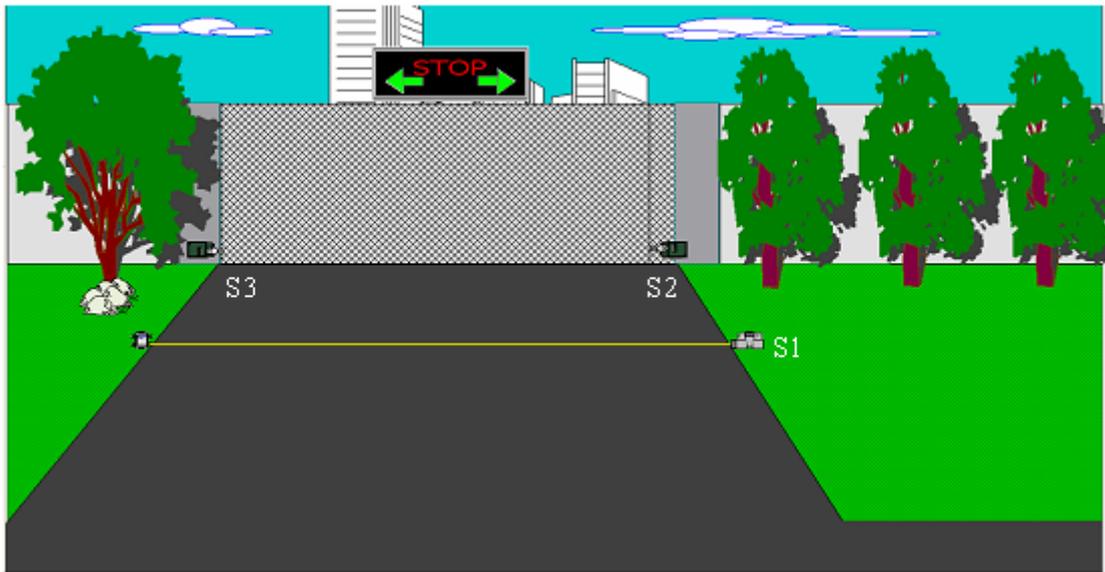


Figura. 5.25. Planteamiento Práctica #5

## 5.5.4. DESARROLLO

### 1. Configuración de driver en RSLinx

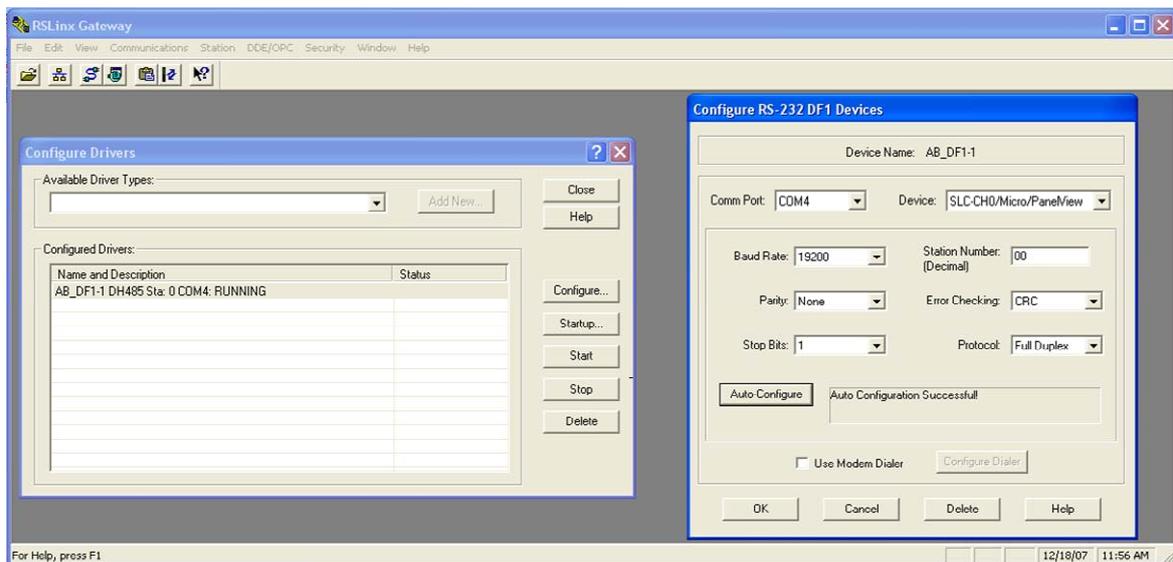


Figura. 5.26. Desarrollo Práctica #5 (1)

### 2. Creación del archivo y configuración del PLC en el software RSLogix500

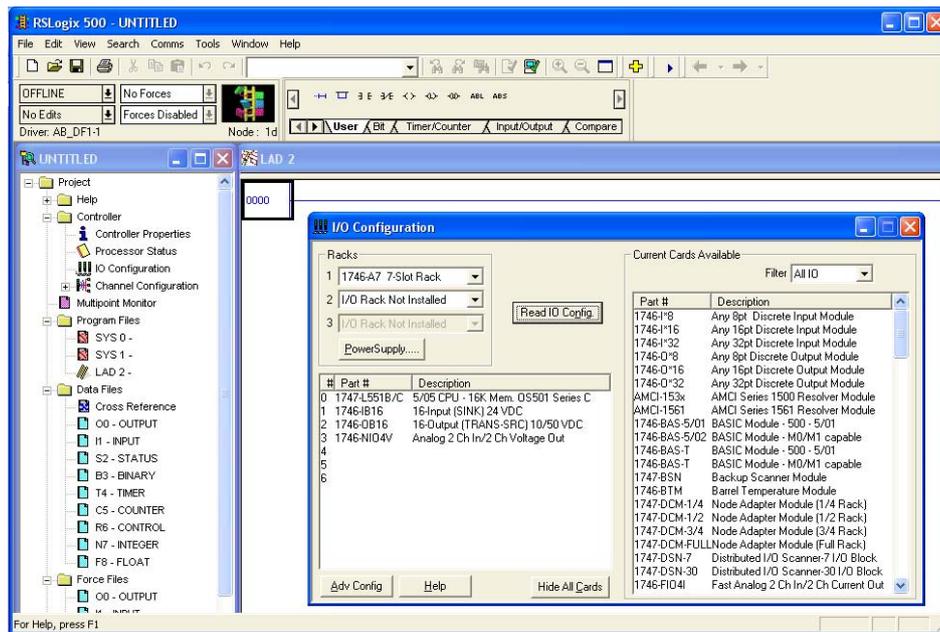


Figura. 5.27. Desarrollo Práctica #5 (2)

3. La programación Ladder que cumple con la lógica de control planteada en esta práctica es la siguiente:

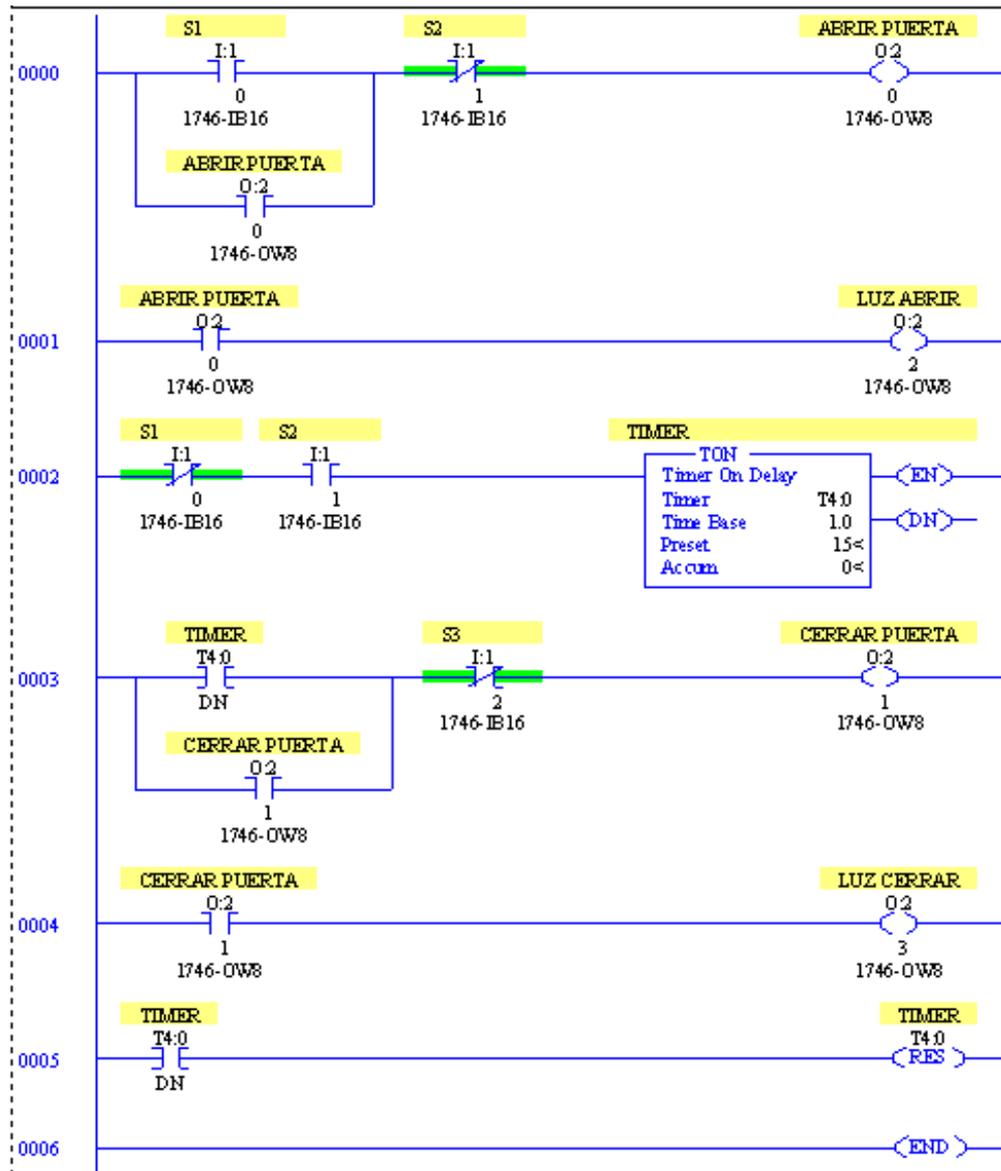


Figura. 5.28. Desarrollo Práctica #5 (3)

4. Transferir el programa.
5. En modo ONLINE se realizó las respectivas pruebas para verificar que el programa realizado cumpla satisfactoriamente con la lógica planteada en esta práctica. A continuación se presenta una pantalla demostrativa de esta verificación, en la que se puede ver que si el sensor S1 esta desactivado y el sensor S2 activado, el temporizador activa la salida CERRAR PUERTA después de transcurridos 15 segundos.

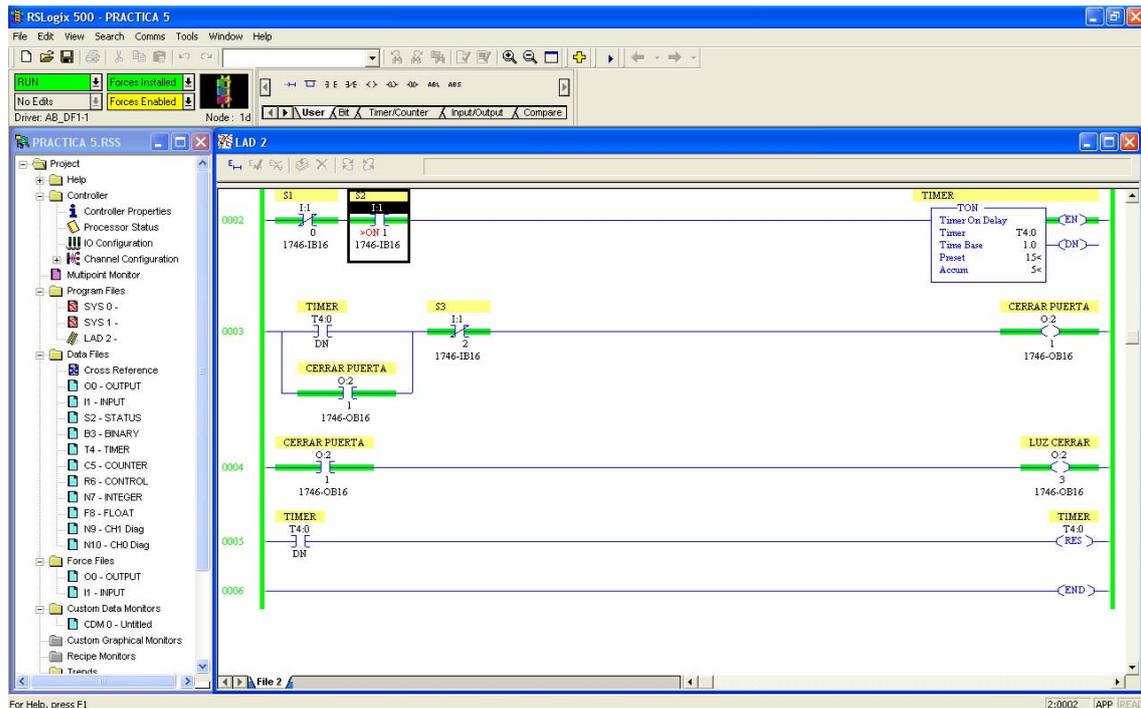


Figura. 5.29. Desarrollo Práctica #5 (4)

6. La conexión de las Entradas y Salidas del PLC al Tablero de simulación correspondiente, se la hizo de acuerdo a la asignación de éstas en el programa:

Nombre	Tipo	Dirección
ABRIR PUERTA	Salida Discreta	O:2/0
CERRAR PUERTA	Salida Discreta	O:2/1
LUZ ABRIR	Salida Discreta	O:2/2
LUZ CERRAR	Salida Discreta	O:2/3
S1	Entrada Discreta	I:1/0
S2	Entrada Discreta	I:1/1
S3	Entrada Discreta	I:1/2
TIMER	Temporizador	T4:0

Tabla. 5.6. Direcciones utilizadas Práctica #5

7. Se verificó que el PLC trabaja correctamente de acuerdo al planteamiento de la práctica, con ayuda del correspondiente tablero didáctico de simulación.

### **5.5.5. PRUEBAS Y RESULTADOS**

Durante el desarrollo de la práctica se realizaron las pruebas respectivas tanto en software (Modo ONLINE del PLC) como en hardware (Tableros didácticos de simulación), a través de las cuales se verificó que el PLC trabaja correctamente de acuerdo a la lógica del problema planteado en esta práctica.

### **5.5.6. CONCLUSIONES Y RECOMENDACIONES**

#### **Conclusiones**

- El uso de las instrucciones de temporización Timer On-delay (TON) ó Timer Off-delay (TOF), dependen de la lógica utilizada en la programación ladder, pues ambas pueden ser utilizadas para una misma aplicación

#### **Recomendaciones**

- Se recomienda realizar el mismo programa utilizando la instrucción TOF (Timer Off Delay) para familiarizarse con su funcionamiento.

## 5.6. CONTROL LÓGICO 6: ENTRADAS Y SALIDAS ANALÓGICAS

### 5.6.1. OBJETIVOS

- Configurar y utilizar las entradas y salidas analógicas.
- Conocer el formato de direccionamiento de datos de tipo Entero.
- Familiarizarse con las instrucciones de Comparación.
- Familiarizarse con las instrucciones de Transferencia de datos
- Familiarizarse con las instrucciones Matemáticas.

### 5.6.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 ó Micrologix 1000 Analog.
- ✓ 1 Módulo de Entradas Discretas\*.
- ✓ 1 Módulo de Salidas Discretas\*.
- ✓ 1 Módulo de Entradas Analógicas\*.
- ✓ 1 Módulo de Salidas Analógicas\*.
- ✓ 1 Computador Personal, con instalación de RSLogix500 y RSLinx
- ✓ 1 Tablero Didáctico de Simulación AB-06
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

(\*No se requiere si se utiliza un PLC de la serie Micrologix 1000 Analog)

### 5.6.3. PLANTEAMIENTO

Se tiene un tanque de almacenamiento de un líquido como se muestra en la figura 5.30, el cual incluye dos válvulas de accionamiento eléctrico para carga y descarga del fluido. La activación de la válvula V1 permite la carga y V2 la descarga, respectivamente. Adicionalmente el tanque posee un medidor ultrasónico de nivel de 0 a 20 metros el cual entrega una salida de 0 a 10 voltios respectivamente.

Se desea que el tanque pueda operar en dos modos: MANUAL y AUTOMÁTICO.

En el modo MANUAL, el operador puede activar y desactivar las válvulas de acuerdo a su conveniencia, sin considerar el nivel en que se encuentra la gasolina.

En el modo AUTOMÁTICO, la lógica de operación se realiza de acuerdo a las siguientes condiciones:

- La válvula V1 se activa cuando el nivel del líquido está por debajo de 15m y se desactiva cuando el nivel alcance los 20 m.
- La válvula V2 se activa cuando el nivel sobrepase los 15m y se desactiva cuando el nivel está por debajo de 5m.

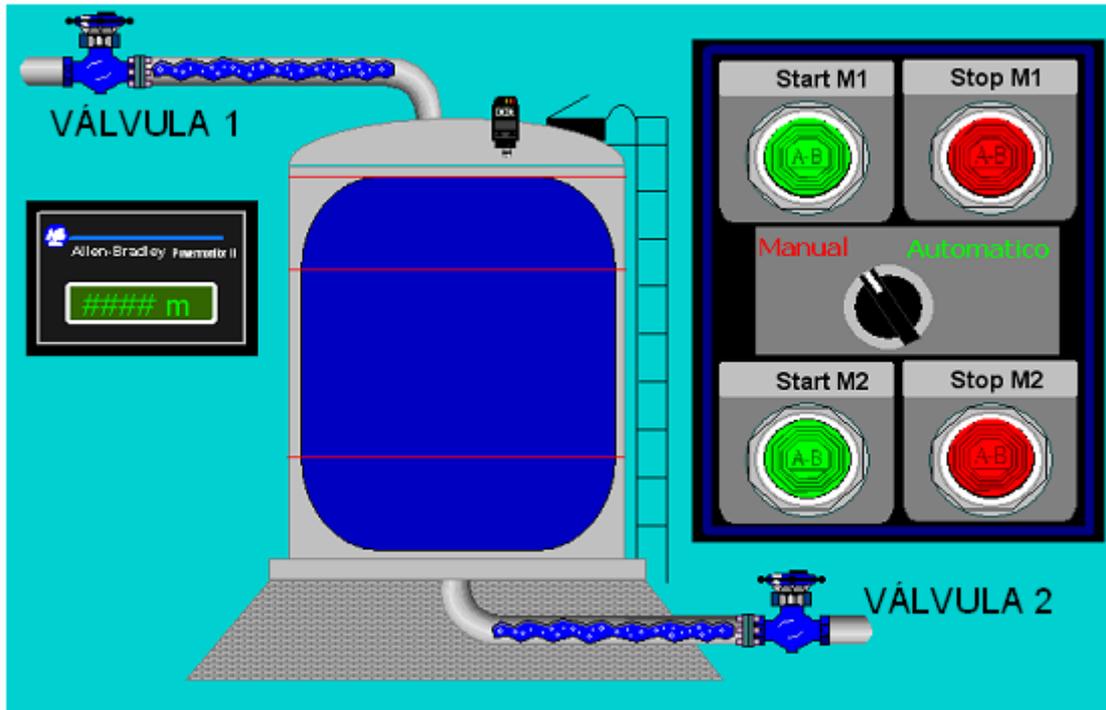


Figura. 5.30. Planteamiento Práctica #6

#### 5.6.4. DESARROLLO

1. Configuración de driver en RSLinx

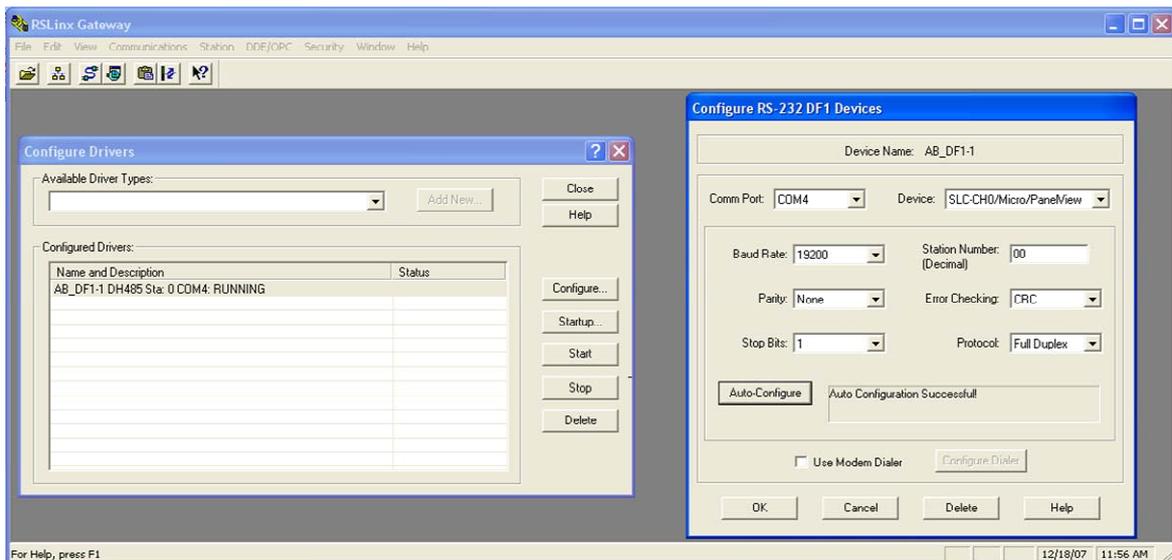


Figura. 5.31. Desarrollo Práctica #6 (1)

## 2. Creación del archivo y configuración del PLC en el software RSLogix500

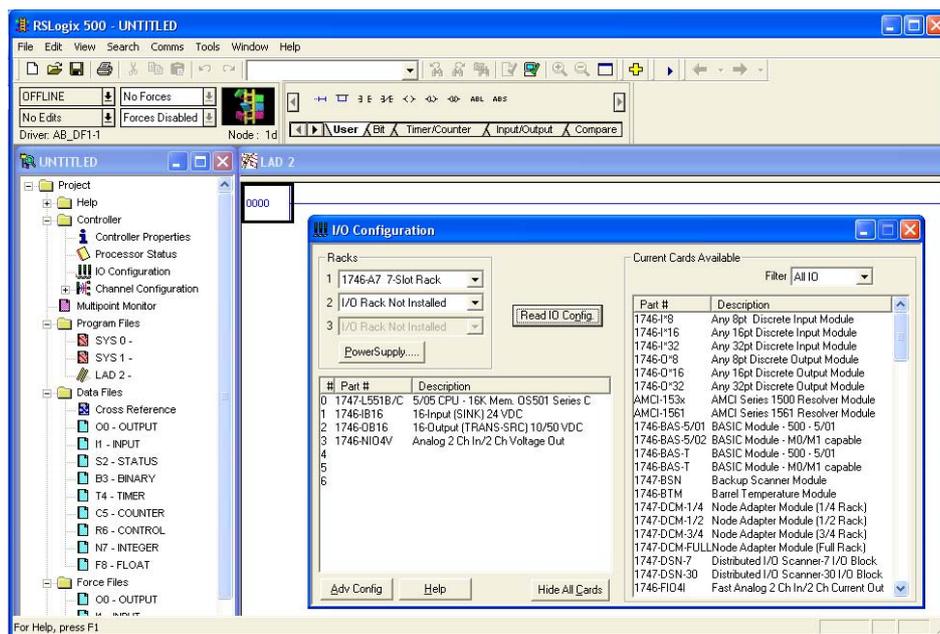


Figura. 5.32. Desarrollo Práctica #6 (2)

## 3. La programación Ladder que cumple con la lógica de control planteada en esta práctica es la siguiente:

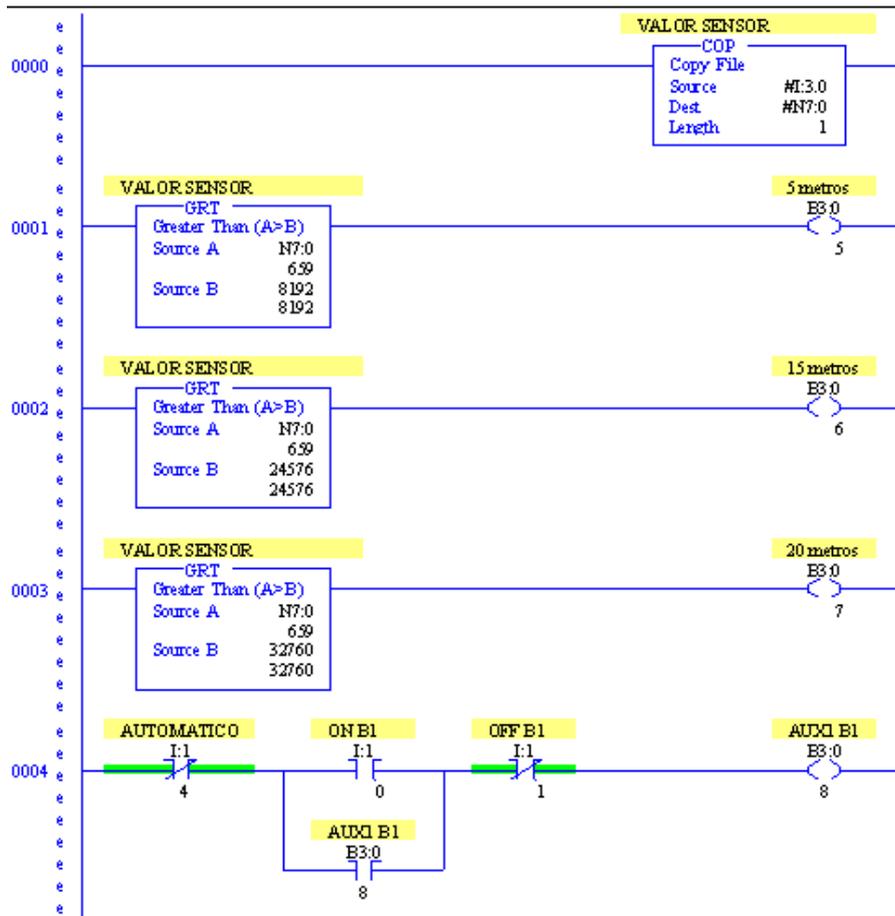


Figura. 5.32. Desarrollo Práctica #6 (3)

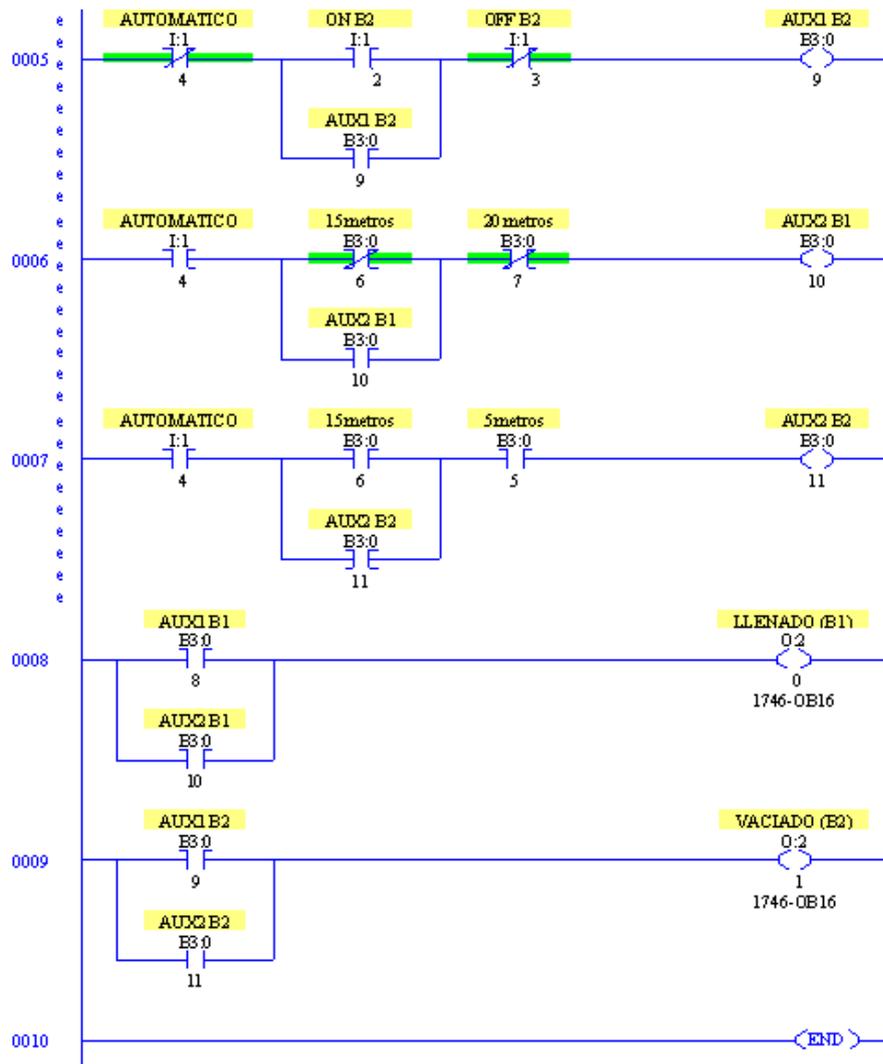


Figura. 5.33. Desarrollo Práctica #6 (4)

Para calcular los valores del registro para cada uno de los niveles (5m, 15m y 20m), se aplicó la siguiente fórmula:

$$\frac{10V}{32768} \times \text{Valor de entrada} = \text{voltaje de entrada (V)}$$

De donde se obtuvieron los siguientes datos:

Altura	Voltaje	Valor del registro
5m	2.5 v	8192
15m	7.5 v	24576
20m	10 v	32760

Tabla. 5.7. Valores de altura

4. Transferir el programa.
5. En modo ONLINE se realizó las respectivas pruebas para verificar que el programa realizado cumpla satisfactoriamente con la lógica planteada en esta práctica. A continuación se presenta una pantalla demostrativa de esta verificación, en la que se puede ver que cuando el sistema opera en modo automático y nivel del líquido es inferior a 5m, el PLC activa la salida LLENADO (B1).

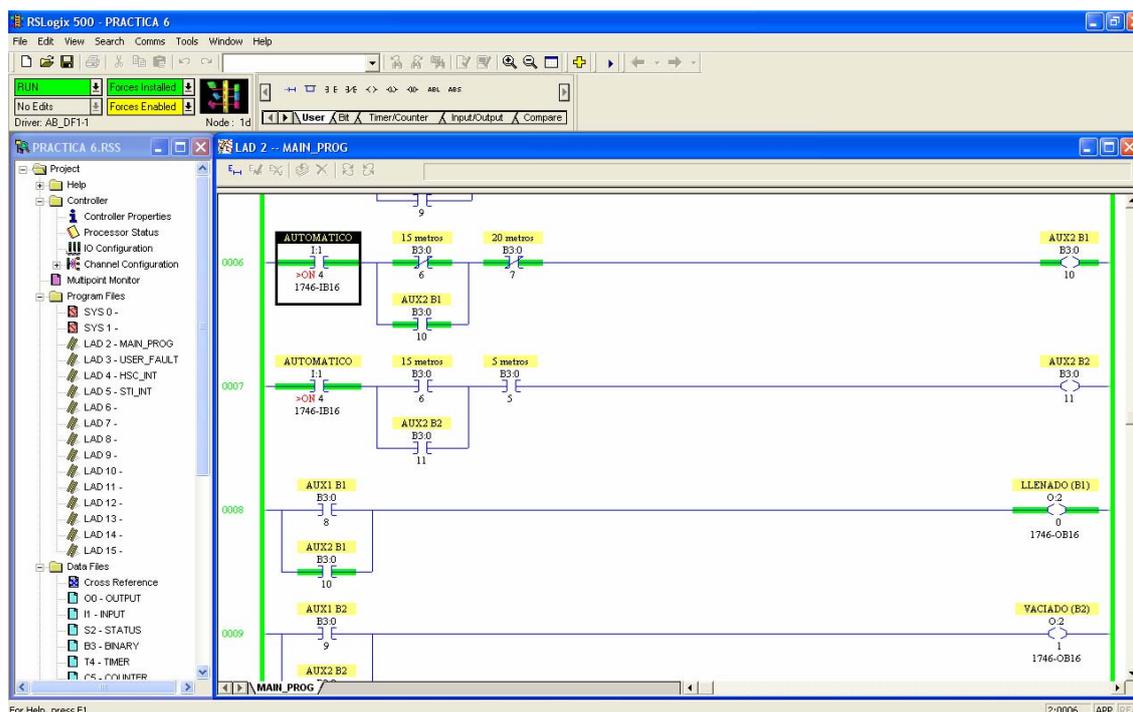


Figura. 5.34. Desarrollo Práctica #6 (5)

6. La conexión de las Entradas y Salidas del PLC al Tablero de simulación correspondiente, se la hizo de acuerdo a la asignación de éstas en el programa:

Nombre	Tipo	Dirección
ON B1	Entrada Discreta	I:1/0
OFF B1	Entrada Discreta	I:1/1
ON B2	Entrada Discreta	I:1/2
OFF B2	Entrada Discreta	I:1/3
AUTOMÁTICO	Entrada Discreta	I:1/4
5 metros	Bit	B3:0/5
15 metros	Bit	B3:0/6

20 metros	Bit	B3:0/7
AUX1 B1	Bit	B3:0/8
AUX1 B2	Bit	B3:0/9
AUX2 B1	Bit	B3:0/10
AUX2 B2	Bit	B3:0/11
LLENADO (B1)	Salida Discreta	O:2/0
VACIADO (B2)	Salida Discreta	O:2/1

**Tabla. 5.8. Direcciones utilizadas Práctica #6**

7. Se verificó que el PLC trabaja correctamente de acuerdo al planteamiento de la práctica, con ayuda del correspondiente tablero didáctico de simulación.

### 5.6.5. PRUEBAS Y RESULTADOS

Durante el desarrollo de la práctica se realizaron las pruebas respectivas tanto en software (Modo ONLINE del PLC) como en hardware (Tableros didácticos de simulación), a través de las cuales se verificó que el PLC trabaja correctamente de acuerdo a la lógica del problema planteado en esta práctica.

### 5.6.6. CONCLUSIONES Y RECOMENDACIONES

#### Conclusiones

- Las entradas analógicas convierten señales de corriente y voltaje en valores binarios de complemento a dos de 16 bits.
- En las Salidas Discretas la Resolución por LSB (bit menos significativo) es 1.22070 mV
- En las Entradas Discretas la Resolución por LSB (bit menos significativo) es 305.176  $\mu$ V.

#### Recomendaciones

- Para mover el valor de una entrada analógica a otro registro, se recomienda utilizar la instrucción COP con una longitud (Length) de 1 en lugar de la instrucción MOV para no afectar los bits aritméticos (S:0)
- Se recomienda conectar en puente las entradas analógicas no utilizadas.

## 5.7. CONTROL LÓGICO 7: REGISTROS DE RELOJ/CALENDARIO

### 5.7.1. OBJETIVOS

- Familiarizarse con el uso y aplicación de los Registros de Reloj/Calendario
- Familiarizarse con las instrucciones de Comparación.

### 5.7.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 5/03 0 superior
- ✓ 1 Módulo de Entradas Discretas.
- ✓ 1 Módulo de Salidas Discretas.
- ✓ 1 Computador Personal, con instalación de RSLogix500 y RSLinx
- ✓ 1 Tablero Didáctico de Simulación AB-07
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

### 5.7.3. PLANTEAMIENTO

Se necesita implementar un sistema de control automático de iluminación para una planta ensambladora de vehículos como la que se muestra en la figura 5.35. El sistema contará con un único switch de Start/Stop. Cuando esté encendido, deberá encender y apagar la iluminación de las diferentes áreas de la planta de acuerdo a los horarios establecidos en la siguiente tabla:

<i>ÁREA</i>	<i>HORA DE ENCENDIDO</i>	<i>HORA DE APAGADO</i>
Materiales y Suelda	06:00	18:00
Pintura	06:30	20:00
Ensamblaje y Plásticos	07:00	23:00
Sala de control	06:00	23:30

**Tabla. 5.9. Horarios Práctica #7**

**Nota:** Estos horario de trabajo se deben cumplir únicamente de lunes a viernes. Los fines de semana las luminarias debe de permanecer apagadas.

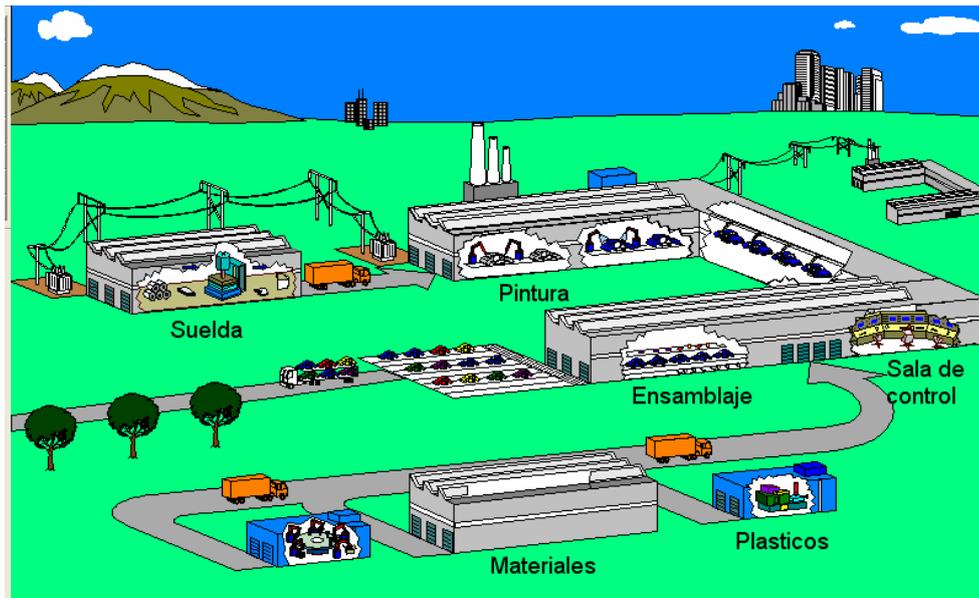


Figura. 5.35. Planteamiento Práctica #7

## 5.7.4. DESARROLLO

### 1. Configuración de driver en RSLinx

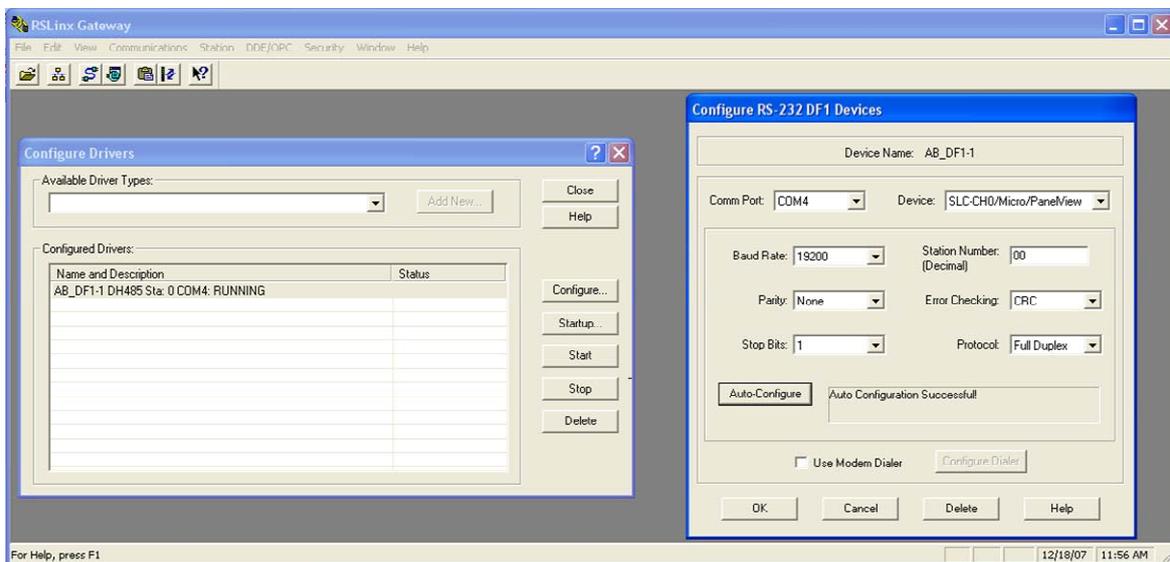


Figura. 5.36. Desarrollo Práctica #7 (1)

### 2. Creación del archivo y configuración del PLC en el software RSLogix500

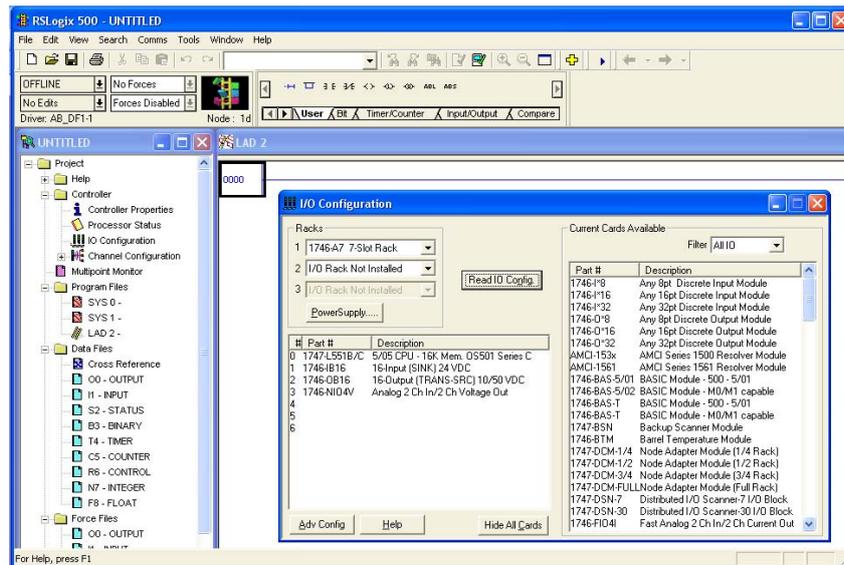
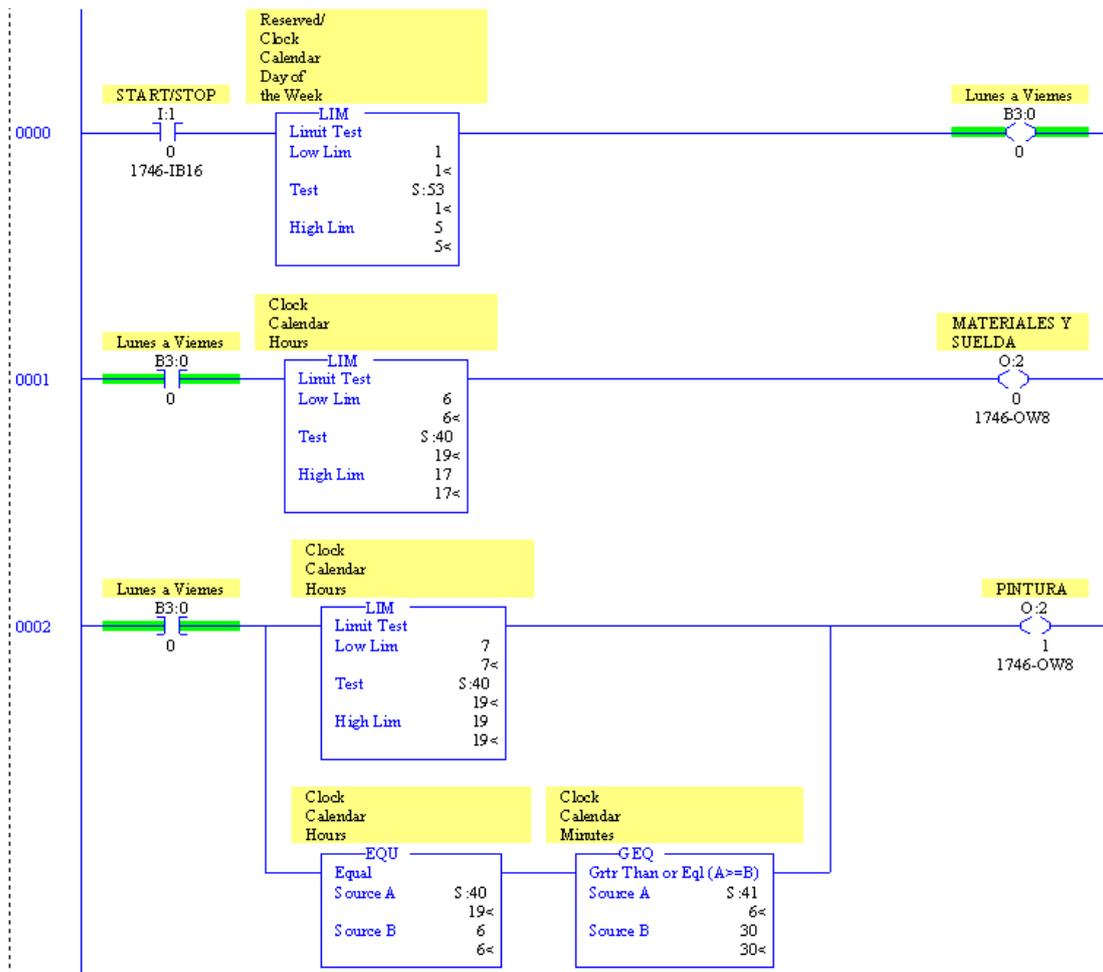


Figura. 5.37. Desarrollo Práctica #7 (2)

3. La programación Ladder que cumple con la lógica de control planteada en esta práctica es la siguiente:



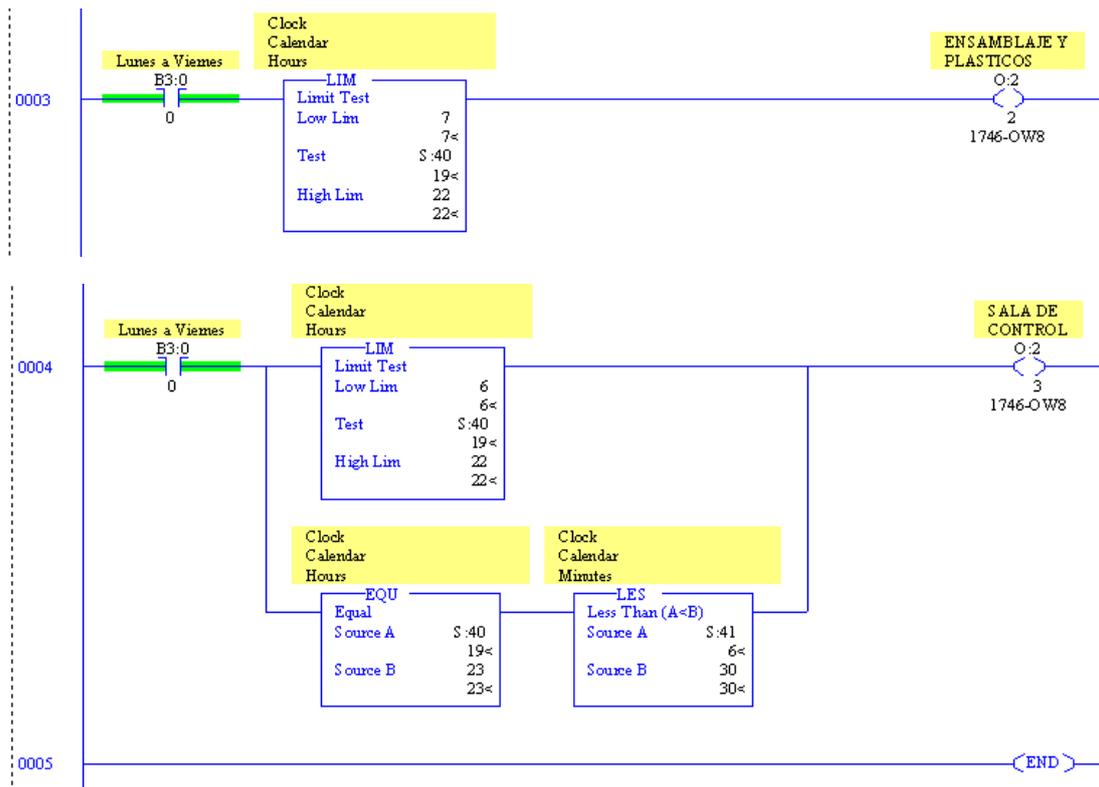


Figura. 5.39. Desarrollo Práctica #7 (4)

4. Transferir el programa.
5. En modo ONLINE se realizó las respectivas pruebas para verificar que el programa realizado cumpla satisfactoriamente con la lógica planteada en esta práctica. A continuación se presenta una pantalla demostrativa de esta verificación, en la que se puede ver que si la fecha en el PLC es 1 de Noviembre del 2007 (jueves) y son las 12:19, el PLC activa la salida MATERIALES Y SUELDA.

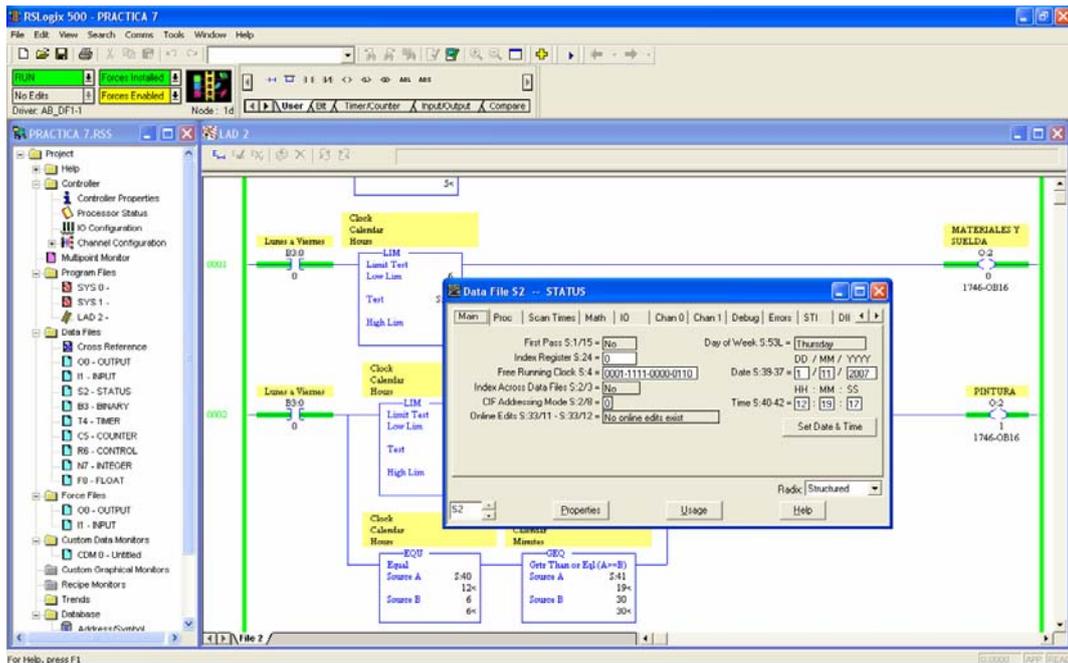


Figura. 5.40. Desarrollo Práctica #7 (5)

6. La conexión de las Entradas y Salidas del PLC al Tablero de simulación correspondiente, se la hizo de acuerdo a la asignación de éstas en el programa:

Nombre	Tipo	Dirección
START/STOP	Entrada Discreta	I:1/0
Materiales y Suelda	Salida Discreta	O:2/0
Pintura	Salida Discreta	O:2/1
Ensamblaje y Plásticos	Salida Discreta	O:2/2
Sala de control	Salida Discreta	O:2/3

Tabla. 5.10. Direcciones utilizadas Práctica #7

7. Se verificó que el PLC trabaja correctamente de acuerdo al planteamiento de la práctica, con ayuda del correspondiente tablero didáctico de simulación.

### 5.7.5. PRUEBAS Y RESULTADOS

Durante el desarrollo de la práctica se realizaron las pruebas respectivas tanto en software (Modo ONLINE del PLC) como en hardware (Tableros didácticos de simulación), a través de las cuales se verificó que el PLC trabaja correctamente de acuerdo a la lógica del problema planteado en esta práctica.

## 5.7.6. CONCLUSIONES Y RECOMENDACIONES

### Conclusiones

- Los Registros de Reloj/Calendario permiten programar eventos de acuerdo a la hora, día o fecha en la que se requiere que se ejecuten.
- El valor de los segundos S:42 puede no incrementarse exactamente cada segundo, por lo que no es confiable para utilizarse en controles basados en segundos.
- Las instrucciones de Comparación se utilizan para condicionar la continuidad lógica del escalón en el que se encuentra la instrucción a la comparación de dos valores.

### Recomendaciones

- Se recomienda realizar la misma práctica sin incluir en la programación la instrucción LIM, lo que permitirá familiarizarse aun más con las instrucciones de comparación.

## 5.8. CONTROL LÓGICO 8: TRANSFERENCIA DE DATOS

### 5.8.1. OBJETIVOS

- Familiarizarse con el uso y aplicación de las instrucciones de transferencia de Datos.
- Familiarizarse con el uso y aplicación de las instrucciones matemáticas.

### 5.8.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 ó Micrologix 1000 Analog.
- ✓ 1 Módulo de Entradas Discretas\*.
- ✓ 1 Módulo de Salidas Discretas\*.
- ✓ 1 Módulo de Entradas Analógicas\*.
- ✓ 1 Módulo de Salidas Analógicas\*.
- ✓ 1 Computador Personal, con instalación de RSLogix500 y RSLinx
- ✓ 1 Tablero Didáctico de Simulación AB-08
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

(\*No se requiere si se utiliza un PLC de la serie Micrologix 1000 Analog)

### 5.8.3. PLANTEAMIENTO

En un horno de pintura, se tiene un sensor de temperatura que trabaja en un rango de 0 a 100 °C, el cual envía una señal al PLC en el rango de 0 a 10 V en relación proporcional y lineal con la temperatura. Se requiere que cuando se presione START el PLC tome 6 muestras (una cada 10 segundos) de la temperatura, las mismas que serán almacenadas en una pila de registros; una vez tomadas las 6 muestras se deberá activar una luz indicadora, calcular el promedio y almacenarlo en un registro. Para cancelar la toma de muestras se deberá presionar STOP.

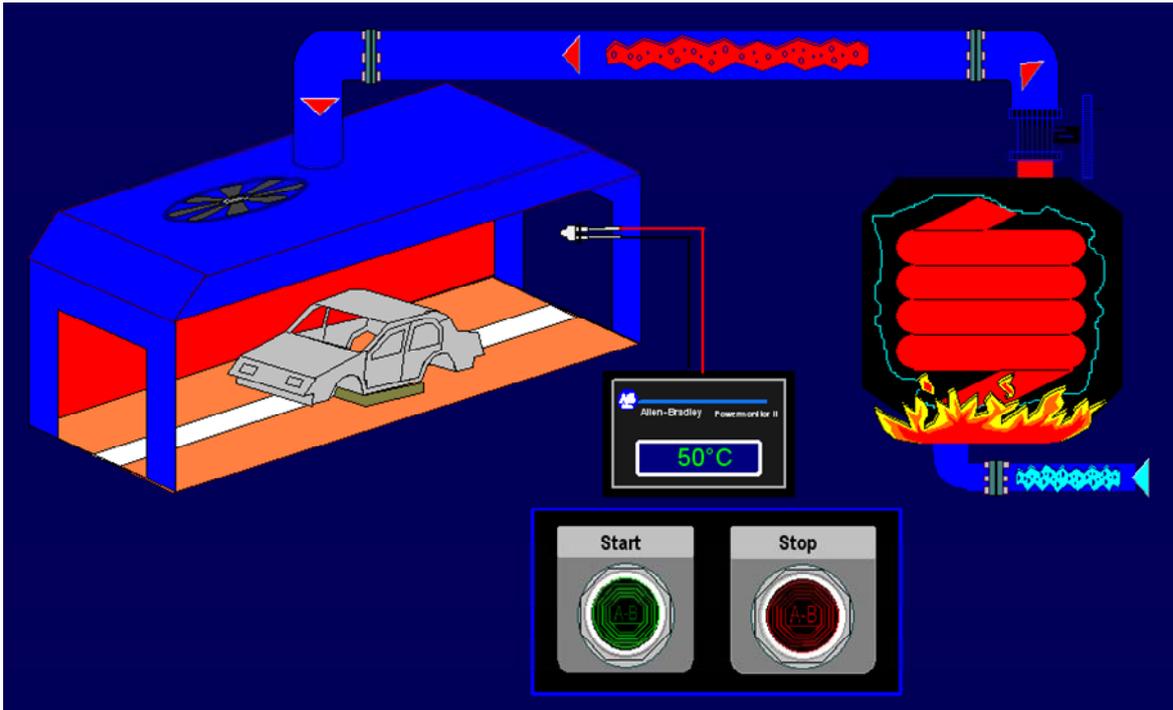


Figura. 5.41. Planteamiento Práctica #8

#### 5.8.4. DESARROLLO

##### 1. Configuración de driver en RSLinx

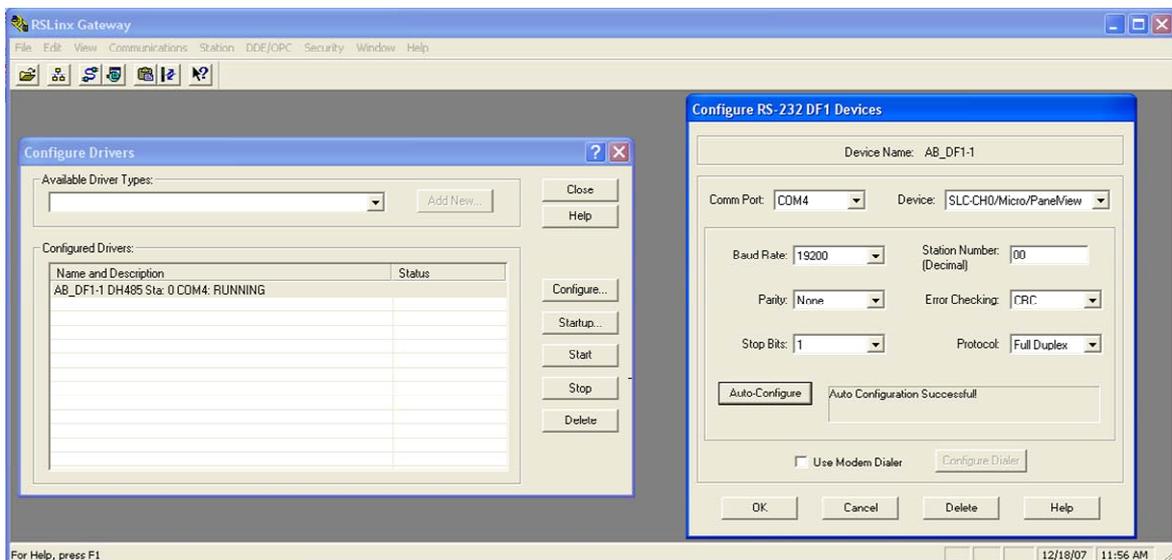


Figura. 5.42. Desarrollo Práctica #8 (1)

##### 2. Creación del archivo y configuración del PLC en el software RSLogix500

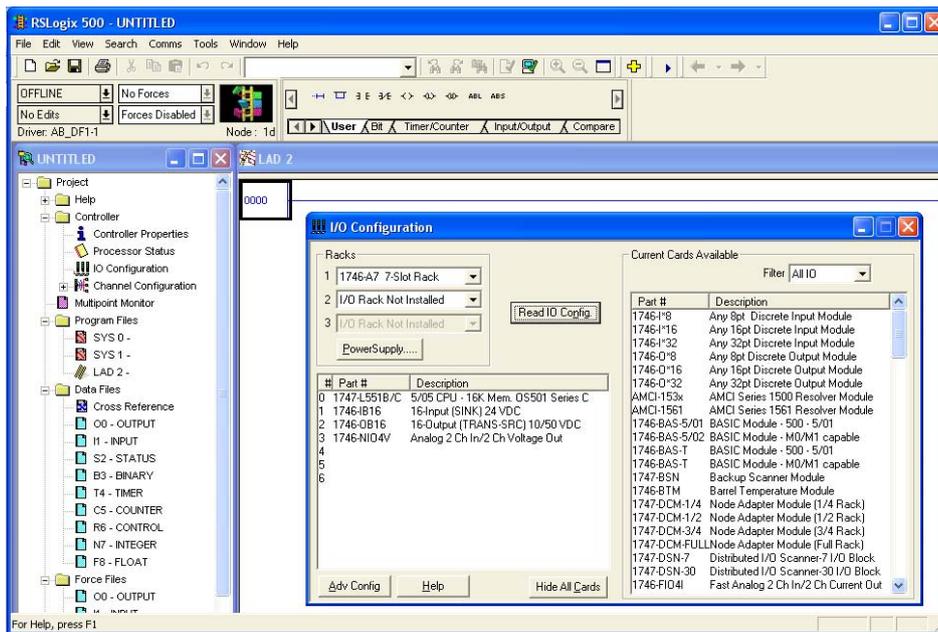
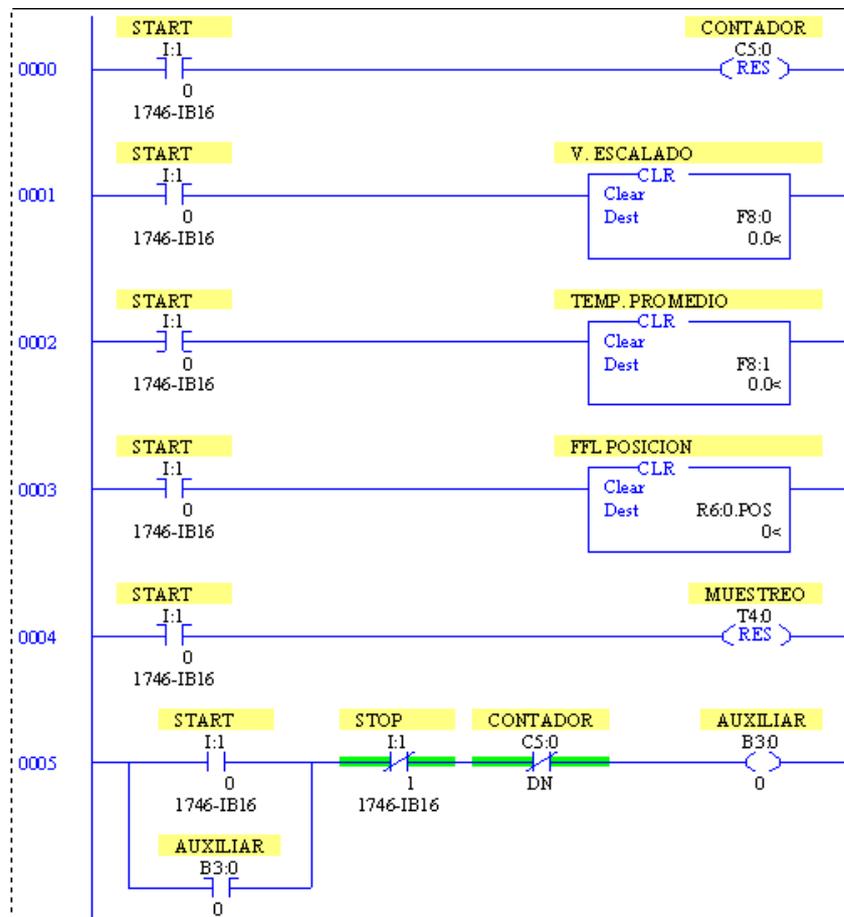


Figura. 5.43. Desarrollo Práctica #8 (2)

3. La programación Ladder que cumple con la lógica de control planteada en esta práctica es la siguiente:



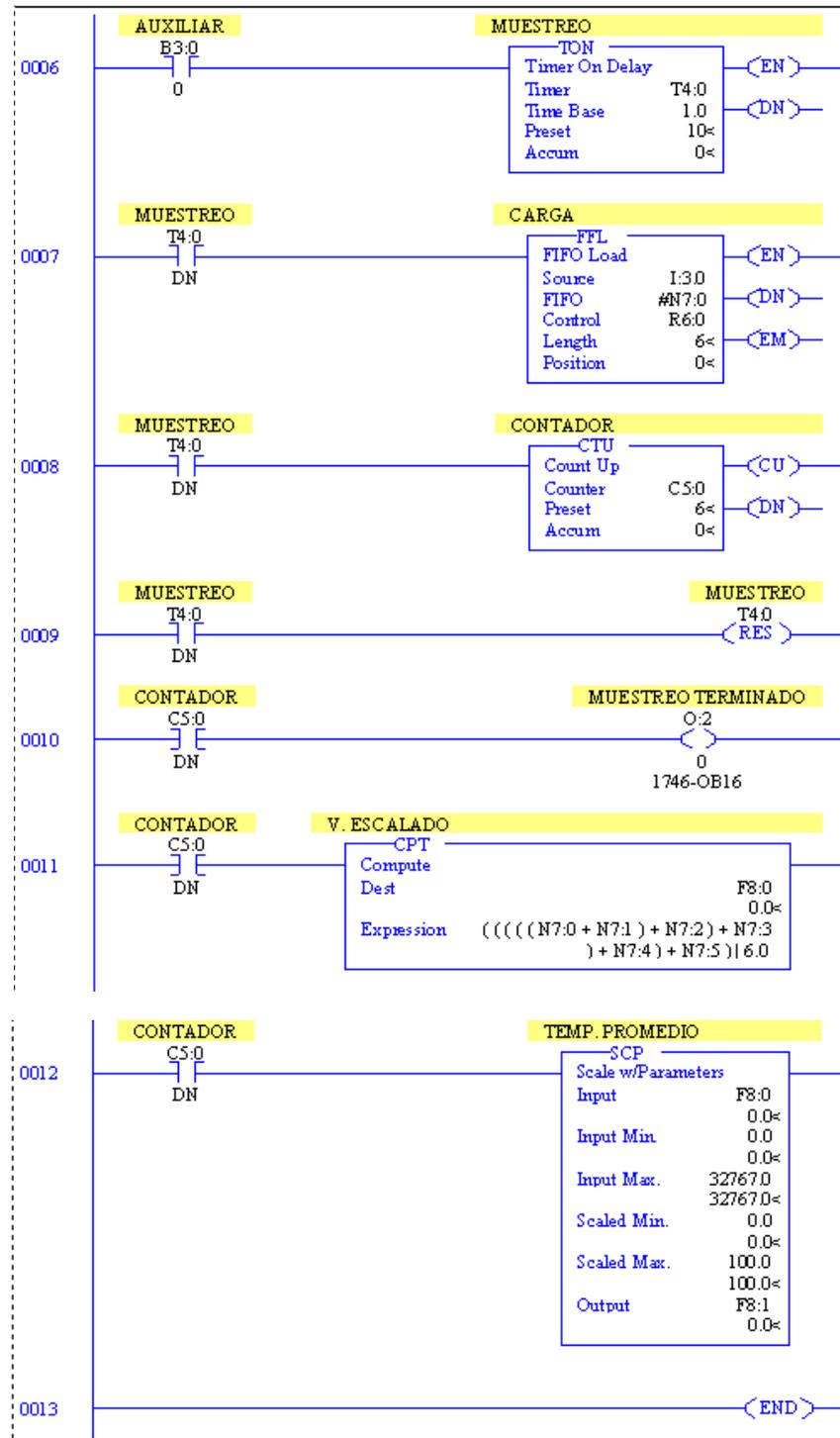


Figura. 5.44. Desarrollo Práctica #8 (3)

4. Transferir el programa.
5. En modo ONLINE se realizó las respectivas pruebas para verificar que el programa realizado cumpla satisfactoriamente con la lógica planteada en esta práctica. A continuación se presenta una pantalla demostrativa de esta verificación, en la que se puede ver los registros en donde se han almacenado

los valores muestreados y también la temperatura promedio obtenida en este caso que es de 70.97 °C

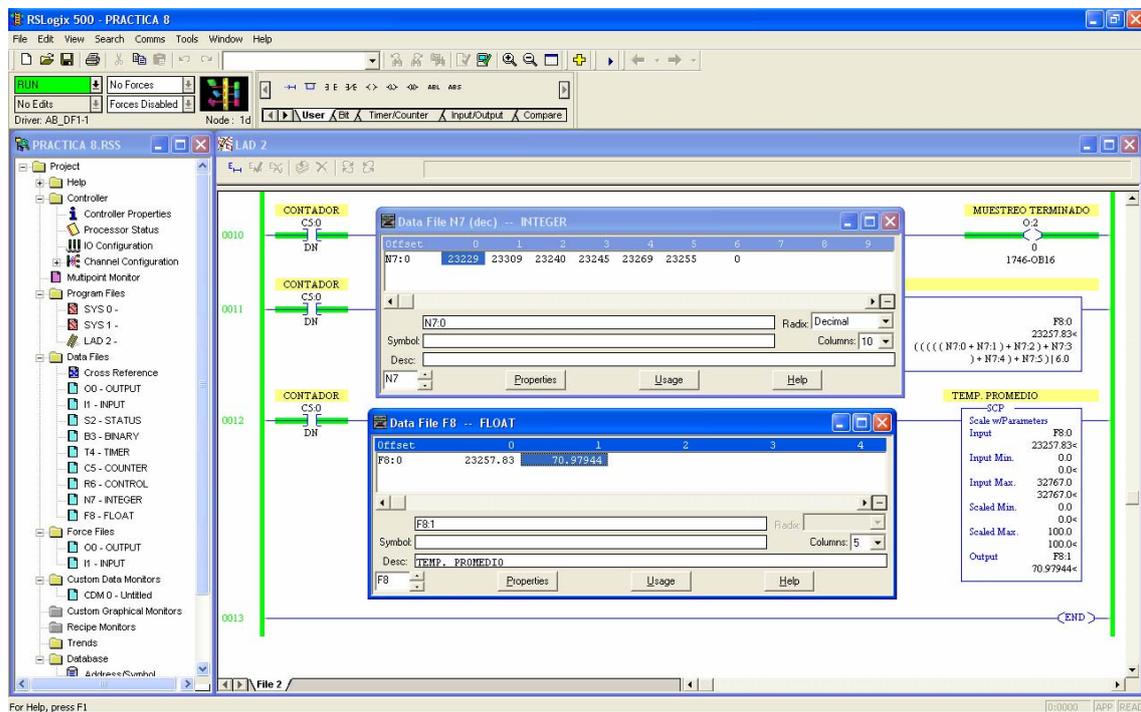


Figura. 5.45. Desarrollo Práctica #8 (4)

6. La conexión de las Entradas y Salidas del PLC al Tablero de simulación correspondiente, se la hizo de acuerdo a la asignación de éstas en el programa:

Nombre	Tipo	Dirección
START	Entrada Discreta	I:1/0
STOP	Entrada Discreta	I:1/1
MUESTREO TERMINADO	Salida Discreta	O:2/0
SENSOR TEMPERATURA	Entrada Analógica	I:3.0
TEMP. PROMEDIO	Registro	F8:1

Tabla. 5.11. Direcciones utilizadas Práctica #8

7. Se verificó que el PLC trabaja correctamente de acuerdo al planteamiento de la práctica, con ayuda del correspondiente tablero didáctico de simulación.

### **5.8.5. PRUEBAS Y RESULTADOS**

Durante el desarrollo de la práctica se realizaron las pruebas respectivas tanto en software (Modo ONLINE del PLC) como en hardware (Tableros didácticos de simulación), a través de las cuales se verificó que el PLC trabaja correctamente de acuerdo a la lógica del problema planteado en esta práctica.

### **5.8.6. CONCLUSIONES Y RECOMENDACIONES**

#### **Conclusiones**

- El tipo de pila en la que se almacenará los datos, dependerá estrictamente de la aplicación, y dependiendo del tipo de pila que se utilice se deberá emplear las respectivas instrucciones de carga y descarga.
- El adecuado escalado de los valores con los que se va a trabajar, es indispensable para facilitar el análisis de los datos en unidades de ingeniería.

#### **Recomendaciones**

- Se recomienda realizar la misma práctica sin incluir en la programación la instrucción CPT (compute), lo que permitirá familiarizarse aun más con las instrucciones matemáticas.
- Se recomienda trabajar en unidades crudas y únicamente transformar el valor de temperatura promedio a unidades de ingeniería, para facilitar la programación y para obtener mayor exactitud.

## **5.9. CONTROL LÓGICO 9: FLUJO DE PROGRAMA.**

### **5.9.1. OBJETIVOS**

- Familiarizarse con el uso y aplicación de las instrucciones de Flujo de Programa.

### **5.9.2. MATERIALES Y EQUIPOS**

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 ó Micrologix.
- ✓ 1 Módulo de Entradas Discretas\*.
- ✓ 1 Módulo de Salidas Discretas\*.
- ✓ 1 Computador Personal, con instalación de RSLogix500 y RSLinx
- ✓ 1 Tablero Didáctico de Simulación AB-01
- ✓ 1 Fuente de alimentación +24 VDC
- ✓ Cables de Conexión

(\*No se requiere si se utiliza un PLC de la serie Micrologix)

### **5.9.3. PLANTEAMIENTO**

Se tienen tres motores, cada uno con un único botón de START/STOP (Utilizar el Tablero Didáctico AB-01, pero trabajar con un solo botón para el encendido y apagado). Se desea que la secuencia de encendido sea: Primero el motor M1, luego el motor M2 y finalmente el motor M3. Mientras que para la secuencia de apagado se desea que el primero en desactivarse sea el motor M3, luego el motor M2 y finalmente el motor M1. No se podrá activar o desactivar los motores, en una secuencia diferente a la antes mencionada.

### **5.9.4. DESARROLLO**

1. Configuración de driver en RSLinx

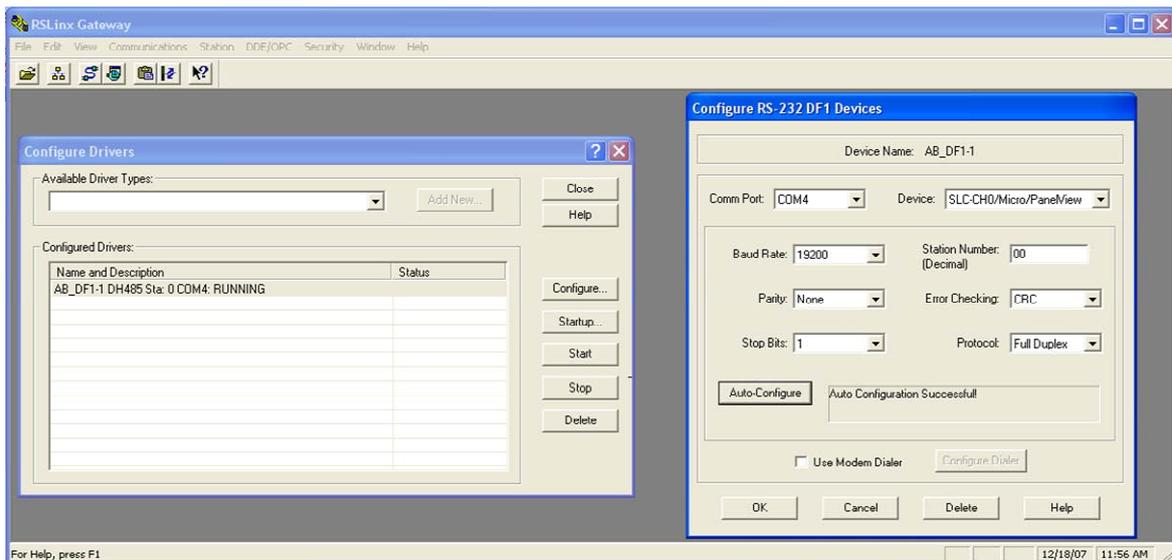


Figura. 5.46. Desarrollo Práctica #9 (1)

## 2. Creación del archivo y configuración del PLC en el software RSLogix500

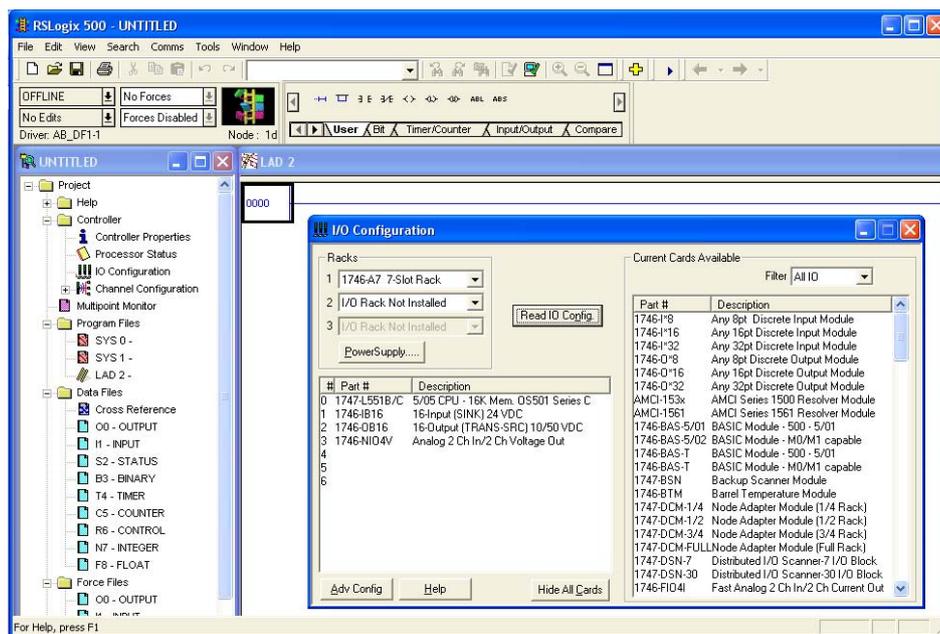
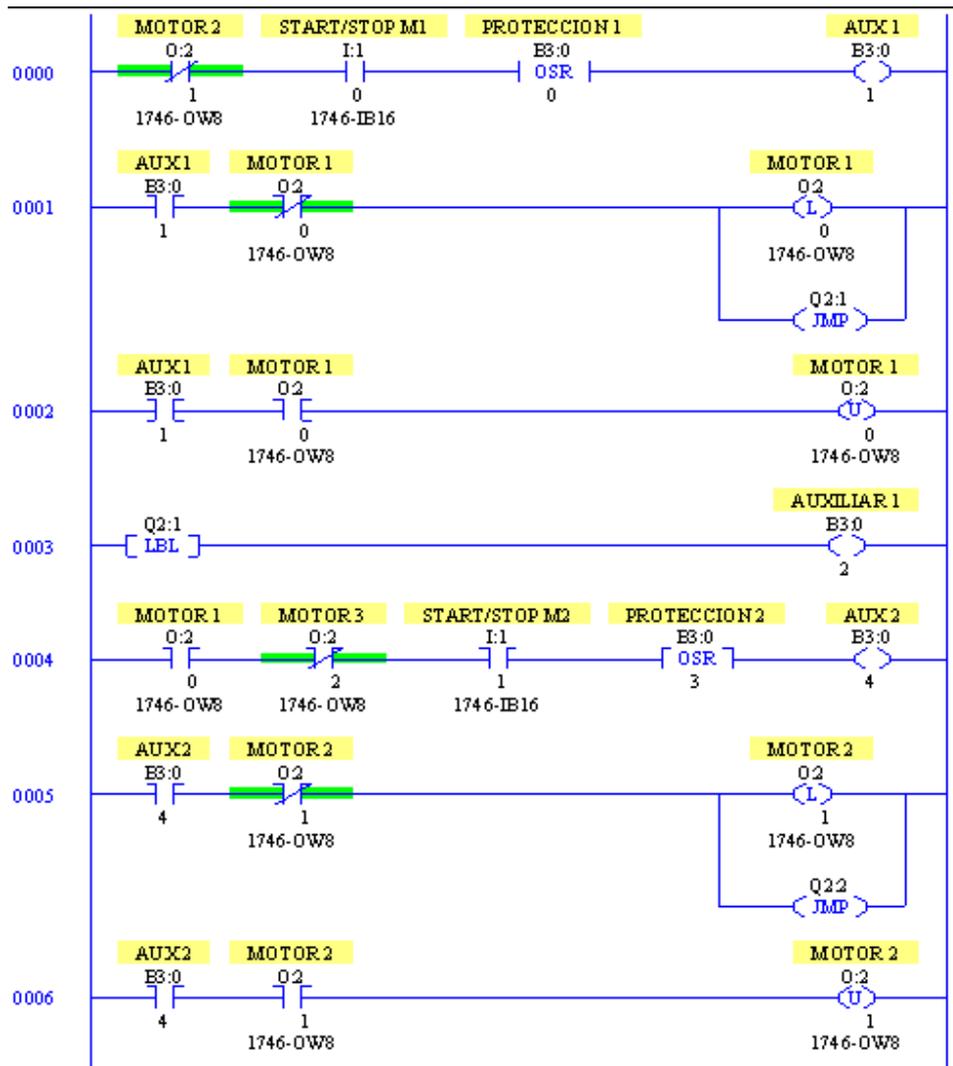


Figura. 5.47. Desarrollo Práctica #9 (2)

## 3. La programación Ladder que cumple con la lógica de control planteada en esta práctica es la siguiente:



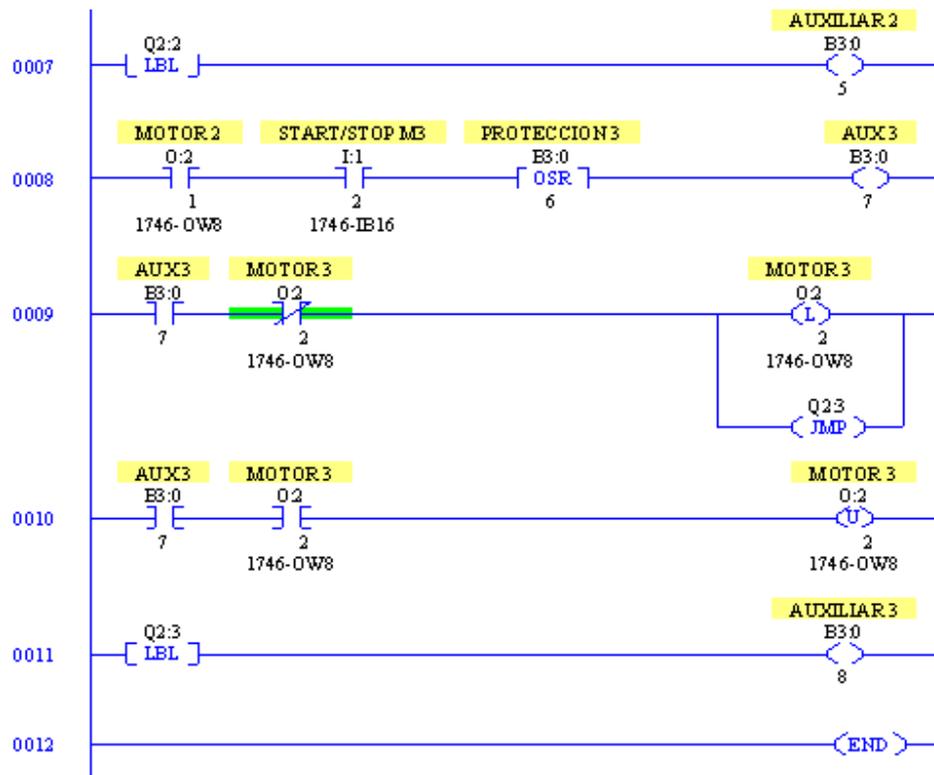


Figura. 5.48. Desarrollo Práctica #9 (3)

4. Transferir el programa.
5. En modo ONLINE se realizó las respectivas pruebas para verificar que el programa realizado cumpla satisfactoriamente con la lógica planteada en esta práctica. A continuación se presenta una pantalla demostrativa de esta verificación, en la que se puede ver que a pesar de que se activó la entrada START/STOP M3, el motor M3 no se enciende puesto de acuerdo a la secuencia de encendido, primero se debe encender el motor M2.

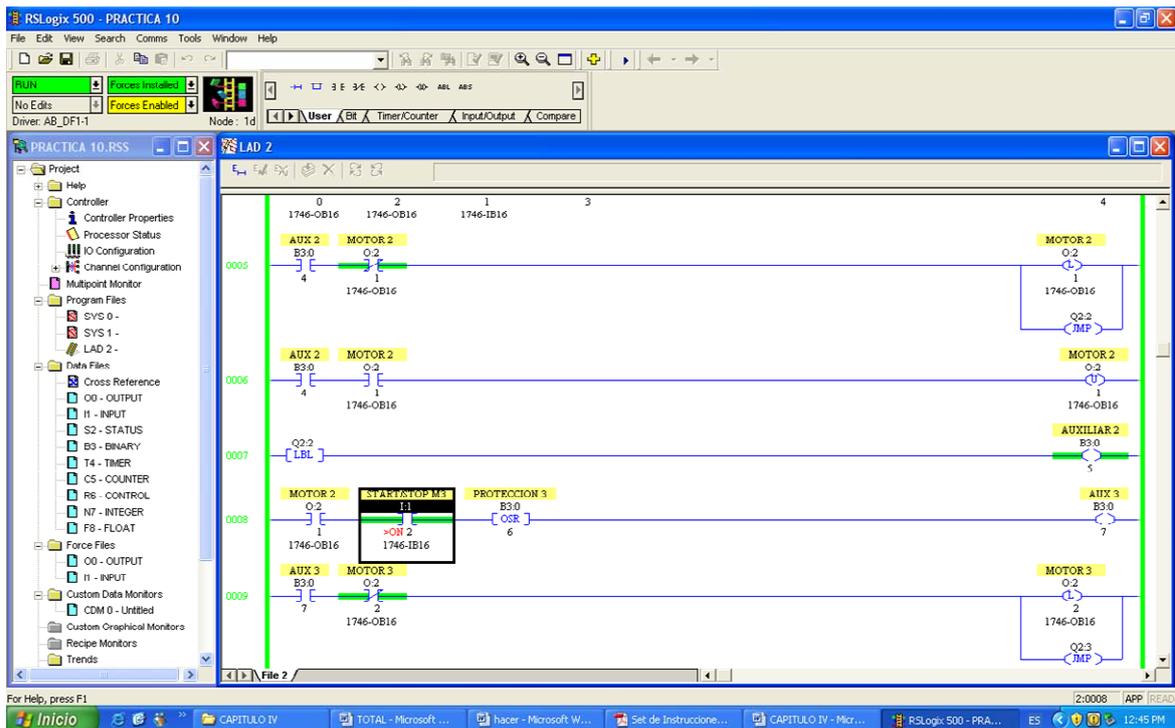


Figura. 5.49. Desarrollo Práctica #9 (4)

6. La conexión de las Entradas y Salidas del PLC al Tablero de simulación correspondiente, se la hizo de acuerdo a la asignación de éstas en el programa:

Nombre	Tipo	Dirección
START/STOP M1	Entrada Discreta	I:1/0
START/STOP M2	Entrada Discreta	I:1/1
START/STOP M3	Entrada Discreta	I:1/2
MOTOR 1	Salida Discreta	O:2/0
MOTOR 2	Salida Discreta	O:2/1
MOTOR 3	Salida Discreta	O:2/2
PROTECCION 1	Bit	B3:0/0
AUX 1	Bit	B3:0/1
AUXILIAR 1	Bit	B3:0/2
PROTECCION 2	Bit	B3:0/3
AUX 2	Bit	B3:0/4
AUXILIAR 2	Bit	B3:0/5
PROTECCION 3	Bit	B3:0/6
AUX 3	Bit	B3:0/7

AUXILIAR 3	Bit	B3:0/8
------------	-----	--------

**Tabla. 5.12. Direcciones utilizadas Práctica #9**

7. Se verificó que el PLC trabaja correctamente de acuerdo al planteamiento de la práctica, con ayuda del correspondiente tablero didáctico de simulación.

### **5.9.5. PRUEBAS Y RESULTADOS**

Durante el desarrollo de la práctica se realizaron las pruebas respectivas tanto en software (Modo ONLINE del PLC) como en hardware (Tableros didácticos de simulación), a través de las cuales se verificó que el PLC trabaja correctamente de acuerdo a la lógica del problema planteado en esta práctica.

### **5.9.6. CONCLUSIONES Y RECOMENDACIONES**

#### **Conclusiones**

- Las instrucciones de flujo de programa permiten al programador controlar y determinar la forma en que se desea que corra el programa, volviéndolo más eficiente, de acuerdo a la aplicación.

#### **Recomendaciones**

- Se recomienda utilizar la instrucción OSR como protección, para evitar que la cuando se presione el pulsador la salida cambie de estado varias veces puesto que el pulsador generalmente se mantiene presionado por más de un scan.

## 5.10. CONTROL DE PROCESOS 1: CONTROL ON-OFF CON HISTÉRESIS

### 5.10.1. OBJETIVOS

- Familiarizarse con la técnica de control ON-OFF con histéresis.
- Familiarizarse con el software RSView32.
- Crear una HMI (Interfase Humano Máquina) que interactúe con el PLC.

### 5.10.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 ó Micrologix 1000 Analog.
- ✓ 1 Módulo de Entradas Discretas\*.
- ✓ 1 Módulo de Salidas Discretas\*.
- ✓ 1 Módulo de Entradas Analógicas\*.
- ✓ 1 Computador Personal, con instalación de RSLogix500, RSLinx y RSView32
- ✓ 1 Módulo Degem de simulación PCT-2 “AIR FLOW TEMPERATURE CONTROL SYSTEM”.
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

(\*No se requiere si se utiliza un PLC de la serie Micrologix 1000 Analog)

### 5.10.3. PLANTEAMIENTO

Diseñar e Implementar un sistema de control de temperatura para un horno de secado de madera que trabaja en un rango de 50°C a 65°C. La técnica de control a utilizarse será la de ON-OFF con Histéresis. El control se lo hará a través de un PLC, al cual ingresa la señal proveniente de un sensor de temperatura que trabaja en un rango de 20°C a 70 °C, enviando una señal al PLC de 0V a 5 V en relación proporcional y lineal con la temperatura del horno. El controlador deberá trabajar en dos modos:

Manual: permite al usuario encender y apagar el horno, indistintamente de la temperatura.

Automático: El encendido y apagado del horno lo hace el controlador ON-OFF con Histéresis.

Adicionalmente con el software RSView32, crear una HMI que permita interactuar al operador con el sistema controlador, la cual deberá tener el siguiente diseño referencial:

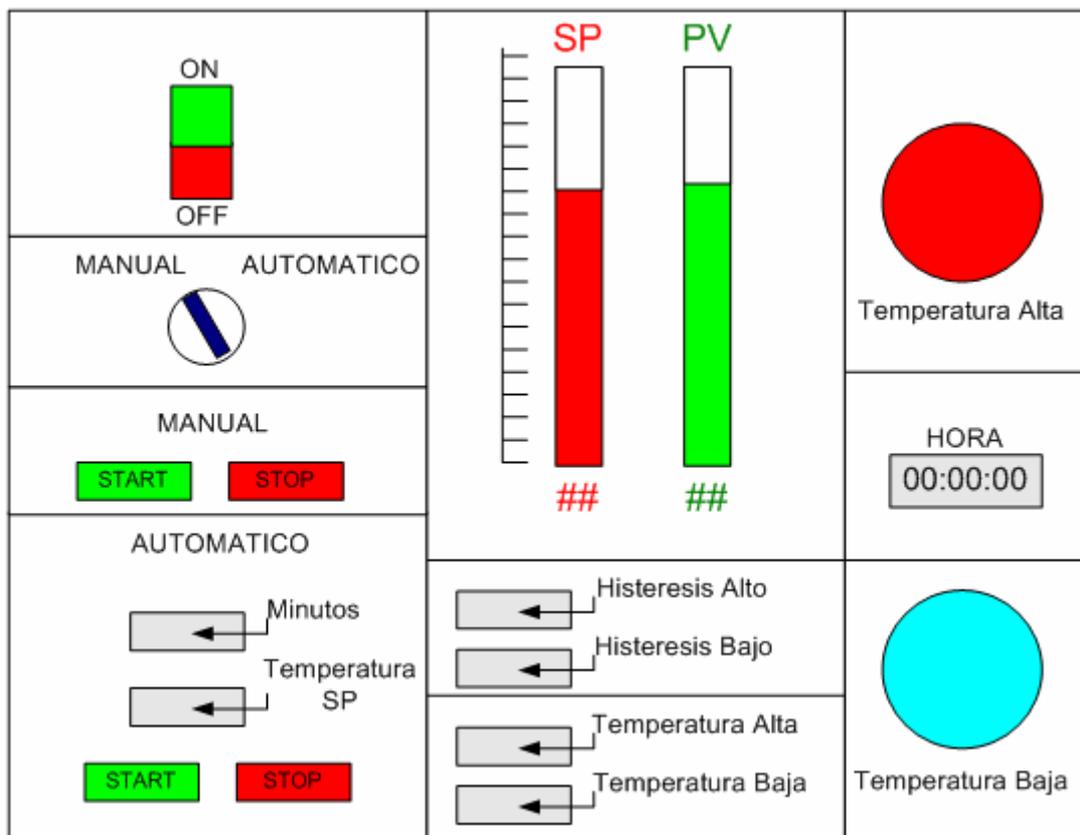


Figura. 5.50. Planteamiento Práctica #10

Nota: Las luces de alarma deberán estar asociadas a una archivo .wav para tener también alarma audible.

#### 5.10.4. DESARROLLO

1. Configuración de driver en RSLinx

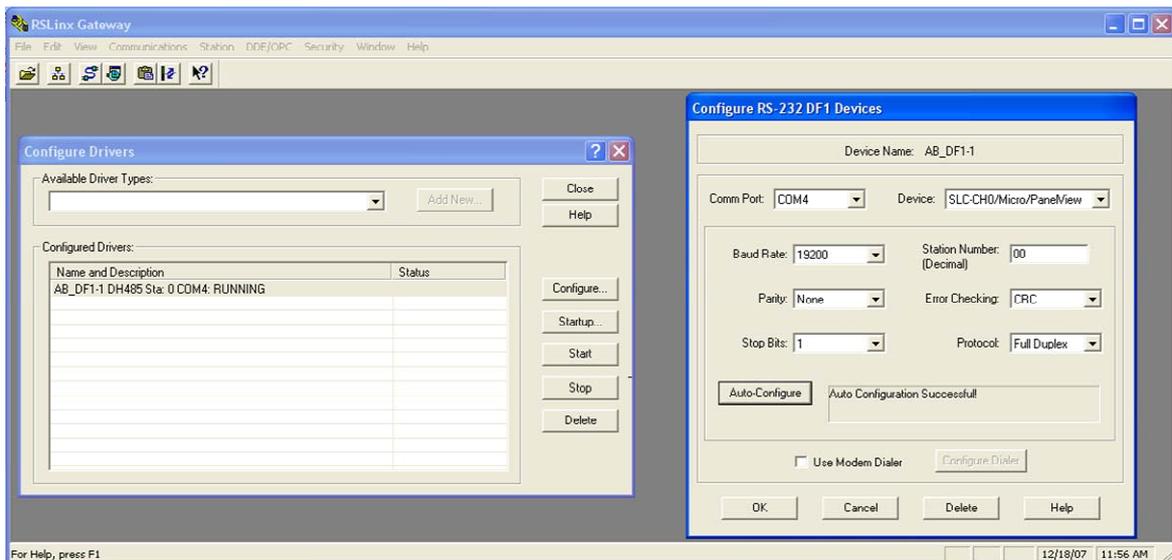


Figura. 5.51. Desarrollo Práctica #10 (1)

2. Creación del archivo y configuración del PLC en el software RSLogix500.

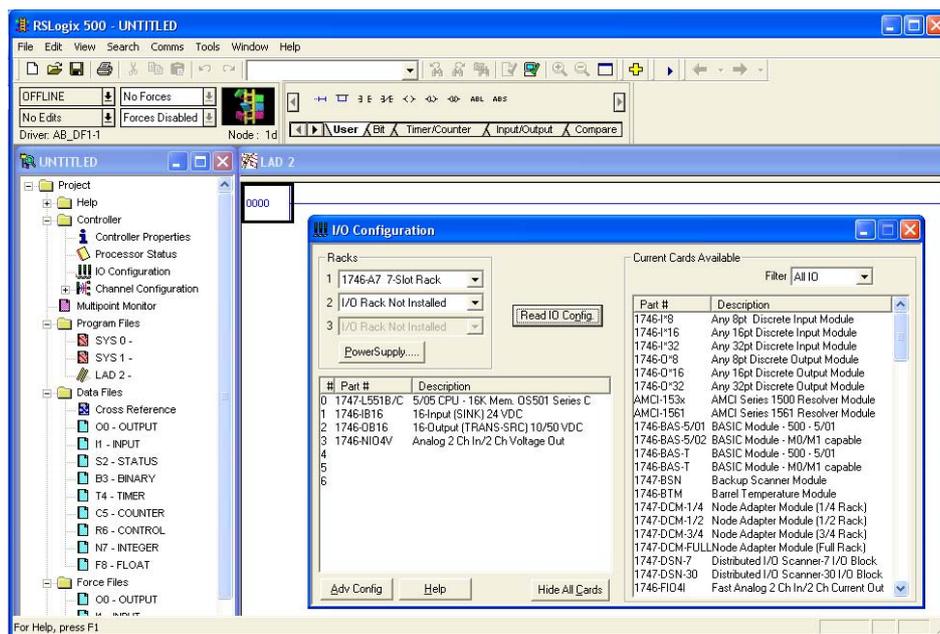


Figura. 5.52. Desarrollo Práctica #10 (2)

3. La programación Ladder que cumple con la lógica de control planteada en esta práctica es la siguiente:

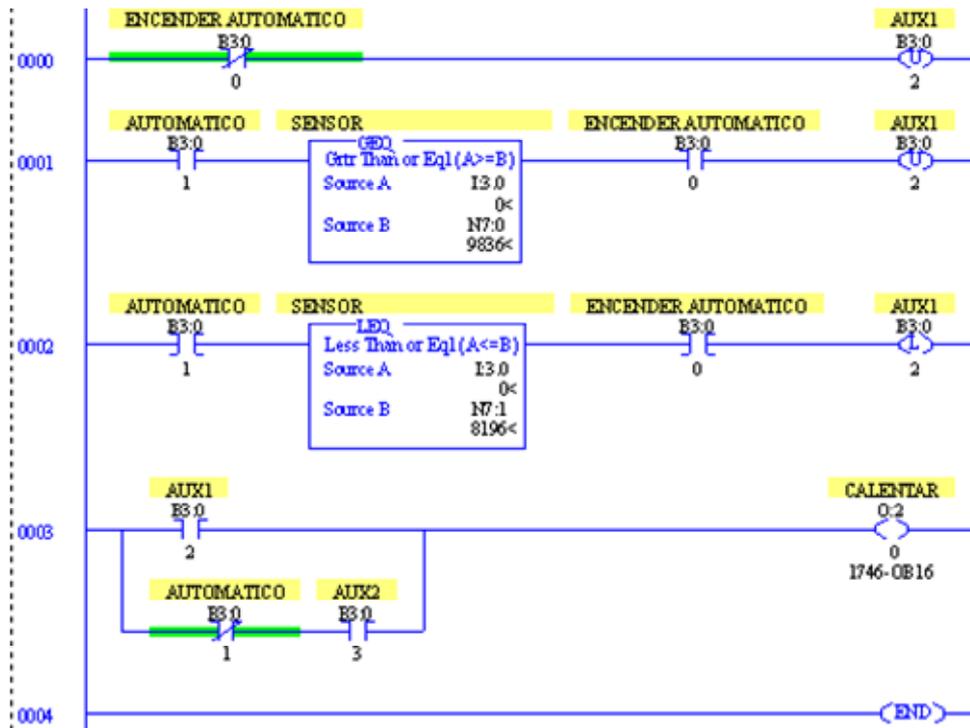


Figura. 5.53. Desarrollo Práctica #10 (3)

- La asignación de direcciones en el PLC de los diferentes registros, entradas y salidas utilizadas, se la hizo de acuerdo a la siguiente tabla:

Nombre	Tipo	Dirección
SENSOR	Entrada Analógica	I:3:0
CALENTAR	Salida Digital	O:2/0
ENCENDER AUTOMÁTICO	Bit	B3:0/0
AUTOMÁTICO	Bit	B3:0/1
AUX1	Bit	B3:0/2
AUX2	Bit	B3:0/3
SUPERIOR	Entero	N7:0
INFERIOR	Entero	N7:1

Tabla. 5.13. Direcciones utilizadas Práctica #10

- Abrir el software RSView32, configurar el canal de comunicación, configurar el nodo de comunicación.



Figura. 5.54. Desarrollo Práctica #10 (4)

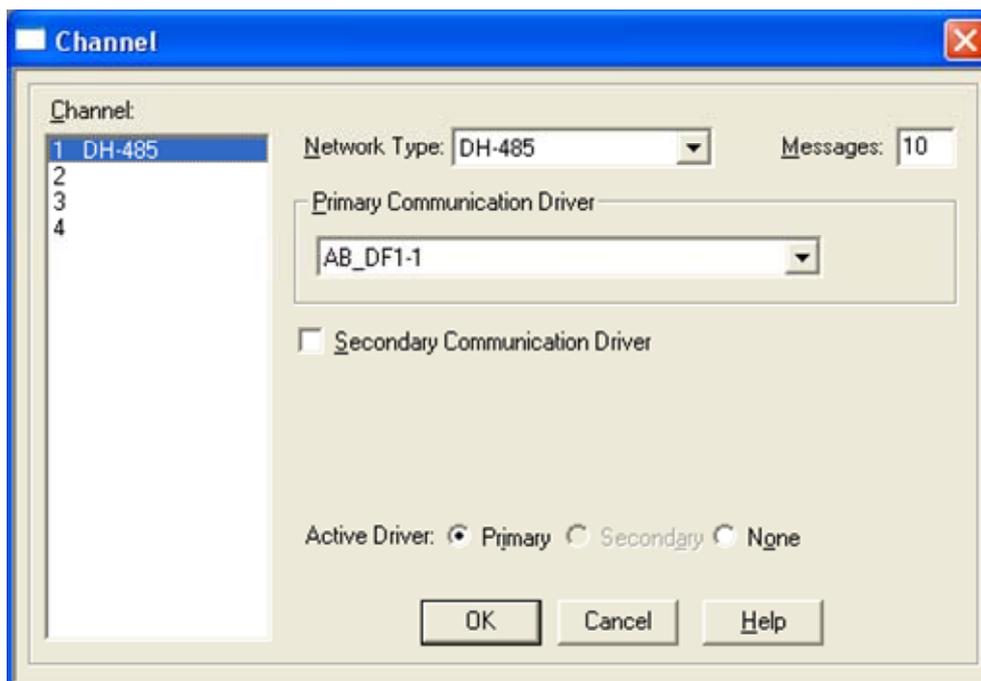


Figura. 5.55. Desarrollo Práctica #10 (5)

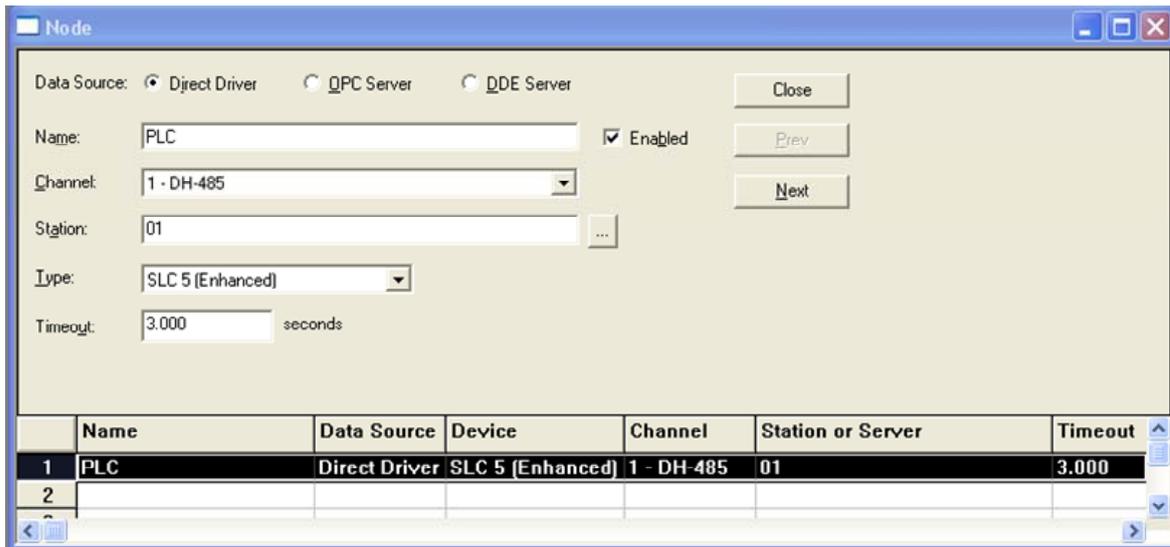


Figura. 5.56. Desarrollo Práctica #10 (6)

## 6. Creación de la HMI que interactúe con el PLC

### INTERFASE HUMANO MÁQUINA (HMI)

La HMI creada en base al esquema planteado, es la siguiente:

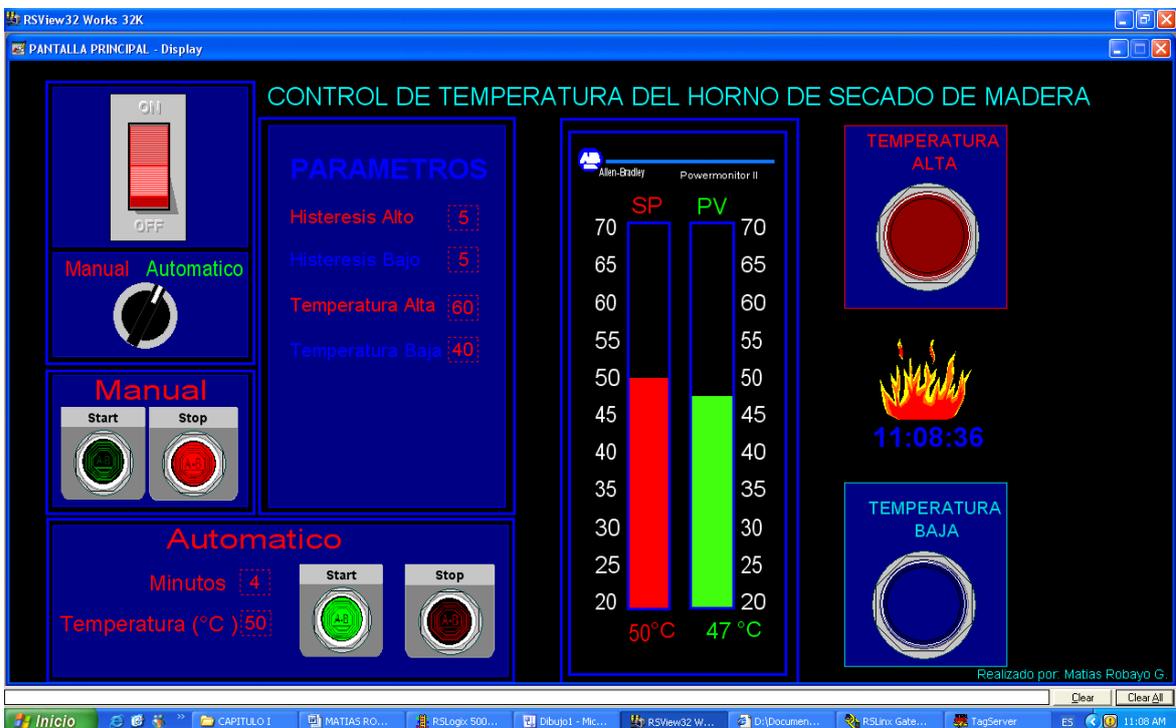


Figura. 5.57. Desarrollo Práctica #10 (7)

## TAGS

En la siguiente tabla se muestran los tags empleados en la programación de la HMI y sus características:

NOMBRE	TIPO	CARACTERÍSTICAS					
		Alarmed: No	Security: *	Device Tag	Node Name: PLC	Address: B3:0/1	Scan Class: A
AUTOMATICO	Digital	Off Label: Off	On Label: On				
CALENTAR	Digital	Off Label: Off	On Label: On				
ENCENDER	Digital	Off Label: Off	On Label: On	Memory Tag	Initial Value: Off		
HIST_ALTO	Analog	Minimum : 0	Maximum : 10	Scale : 1	Offset : 0	Units:	Data Type: (Default)
HIST_BAJO	Analog	Minimum : 0	Maximum : 10	Scale : 1	Offset : 0	Units:	Data Type: (Default)
HISTERESIS_ALTO	Analog	Minimum : 31	Maximum : 65	Scale : 0.003050	Offset : 20.500000	Units:	Data Type: Integer
HISTERESIS_BAJO	Analog	Minimum : 30	Maximum : 64	Scale : 0.003050	Offset : 18.500000	Units:	Data Type: Integer
MINUTOS	Analog	Minimum : 0	Maximum : 100	Scale : 1	Offset : 0	Units:	Data Type: (Default)
PV	Analog	Minimum : 20	Maximum : 70	Scale : 0.003050	Offset : 20	Units:	Data Type: (Default)
START_AUTOMATICO	Digital	Off Label: Off	On Label: On				
START_MANUAL	Digital	Off Label:	On Label:				

		Off	On				
TEMP_ALTA	Analog	Alarmed: No	Security: *	Memory Tag	Initial Value: 60		
		Minimum : 20	Maximum : 70	Scale : 1	Offset : 0	Units:	Data Type: (Default)
TEMP_BAJA	Analog	Alarmed: No	Security: *	Memory Tag	Initial Value: 40		
		Minimum : 20	Maximum : 70	Scale : 1	Offset : 0	Units:	Data Type: (Default)
TEMPERATURA	Analog	Alarmed: No	Security: *	Memory Tag	Initial Value: 50		
		Minimum : 30	Maximum : 65	Scale : 1	Offset : 0	Units:	Data Type: (Default)

Tabla. 5.14. Tags utilizados Práctica #10

Estos tags se encuentran configurados en la ventana *Tag Database* de acuerdo a las características que se encuentran en la tabla anterior, a continuación se muestra un ejemplo de esta configuración:

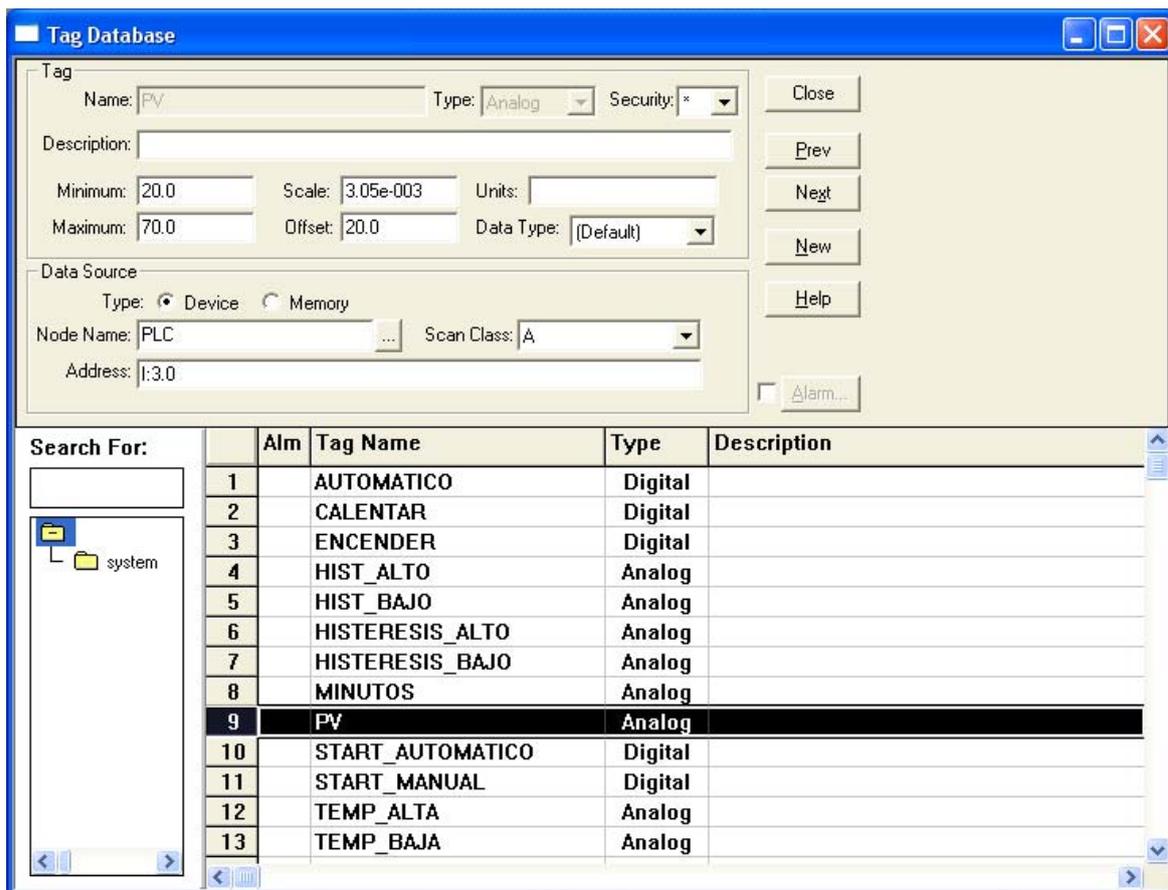


Figura. 5.58. Desarrollo Práctica #10 (8)

Como se puede ver en la figura 5.58, el tag PV es de tipo análogo y se encuentra asociado a la dirección I:3.0 (entrada analógica) a través del nodo PLC, tiene un Mínimo de 20 y un Máximo de 70 puesto que ese es el rango de temperaturas en unidades de ingeniería. Además, se encuentra escalado para transformar las unidades crudas en unidades de ingeniería con las siguientes fórmulas:

$$Scale = \frac{(Escalado \text{ Máximo} - Escalado \text{ Mínimo})}{(Entrada \text{ Máxima} - Entrada \text{ Mínima})} = \frac{(70 - 20)}{(16383 - 0)} = 0.00305$$

$$Offset = Escalado \text{ Mínimo} - (Entrada \text{ Mínima} \times Scale) = 20 - (0 \times 0.00305) = 20$$

## EVENTOS

Como se muestra en la figura 5.59, para la programación de esta HMI se emplearon tres eventos: APAGA, HISTERESIS, MINUTOS. Estos eventos se activan cada vez que se presione el botón START en modo automático, y se desactivan cuando la variable MINUTOS = 0 o cuando se presiona el botón STOP en modo automático.

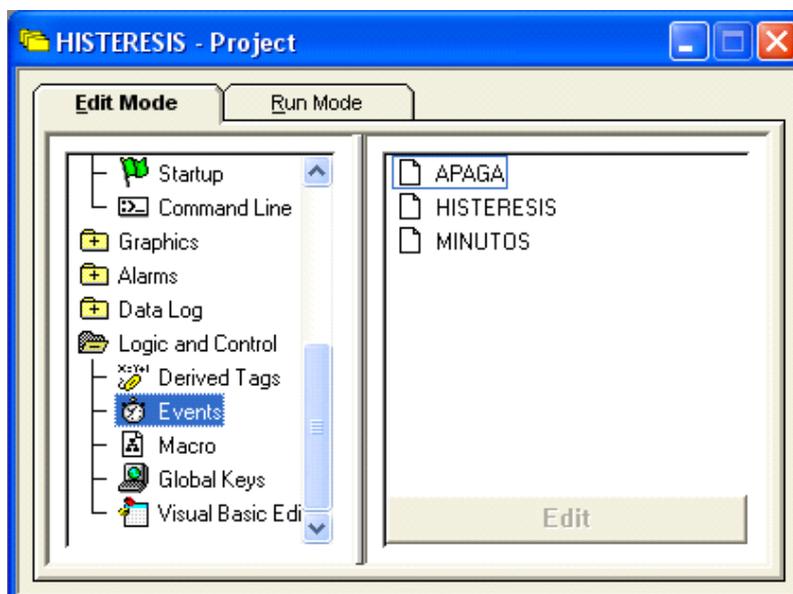


Figura. 5.59. Desarrollo Práctica #10 (9)

A continuación se describen cada uno de estos eventos

**APAGA:** Este evento apaga el sistema cuando la variable (tag) minutos es igual a cero, es decir cuando el tiempo programado para el secado de la madera, ya ha transcurrido.

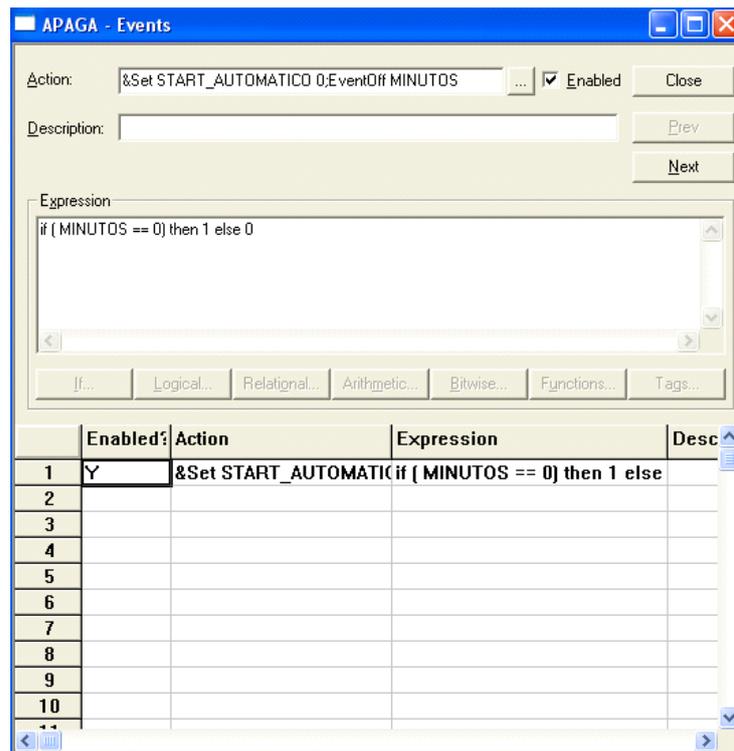


Figura. 5.60. Desarrollo Práctica #10 (10)

**MINUTOS:** Este evento reduce en 1 la variable MINUTOS cada vez que culmina un intervalo de 1 minuto.

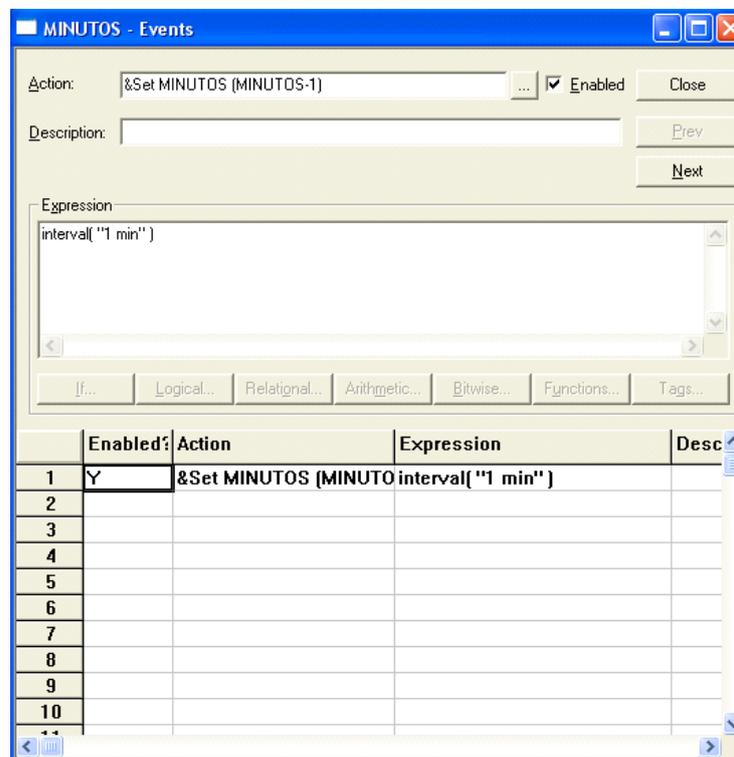


Figura. 5.61. Desarrollo Práctica #10 (11)

**HISTERESIS:** Este evento calcula el valor de temperatura superior e inferior, en base a la suma de la temperatura ingresada (setpoint) más el valor de histéresis superior e inferior que se ha seteado en la HMI.

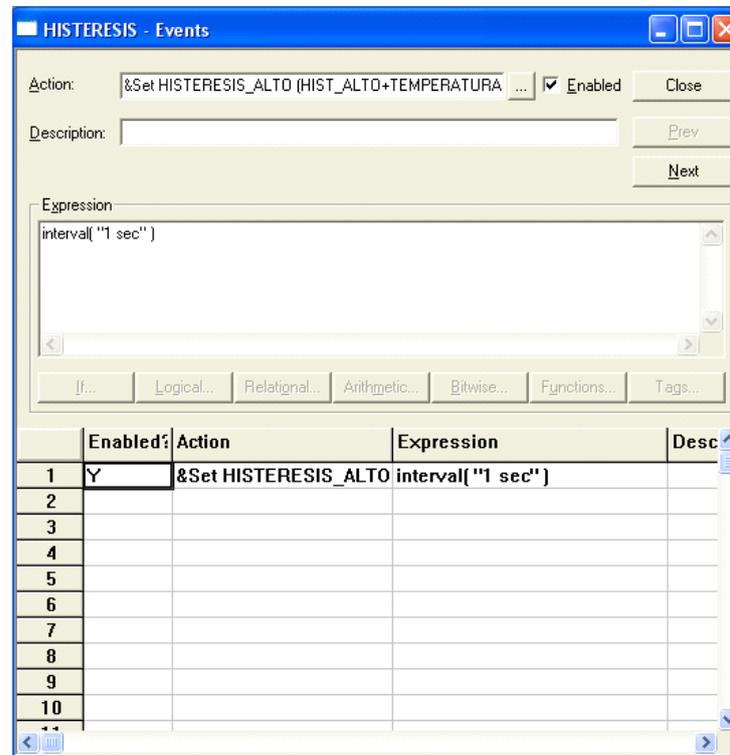


Figura. 5.62. Desarrollo Práctica #10 (12)

- Finalmente, se realizaron las pruebas necesarias para verificar que el controlador funciona correctamente de acuerdo al problema planteado.

#### 5.10.5. PRUEBAS Y RESULTADOS

Durante el desarrollo de la práctica se realizaron las pruebas respectivas tanto del software (programación del PLC y de la HMI) como del hardware (Conexión del PLC y del Módulo Degem de simulación “AIR FLOW TEMPERATURE CONTROL SYSTEM PCT-2”), a través de las cuales se verificó que el PLC y la HMI trabajan correctamente de acuerdo a la lógica del problema planteado en esta práctica.

### 5.10.6. CONCLUSIONES Y RECOMENDACIONES

#### Conclusiones

- La lógica de control On-Off con Histéresis funciona satisfactoriamente en procesos que tienen una velocidad de reacción lenta, si la velocidad de reacción del proceso es muy rápida, en la variable controlada se producen oscilaciones con frecuencias demasiado altas para un correcto funcionamiento.
- La HMI (Interfase Humano Máquina) permite al operador interactuar fácilmente tanto con el controlador como con el proceso controlado.

#### Recomendaciones

- Se recomienda asignar nombres iguales a las variables de la programación ladder del PLC y a los tags de la HMI que se encuentren asociados, para facilitar la revisión y corrección de posibles errores en la programación.
- Se recomienda considerar la inercia del proceso para evitar que la temperatura exceda los límites establecidos por los valores de histéresis.

## 5.11. CONTROL DE PROCESOS 2: CONTROL PID DE TEMPERATURA

### 5.11.1. OBJETIVOS

- Evaluar el desempeño de las técnicas PI y PID en el control de temperatura de un flujo de aire.
- Conocer el funcionamiento, configuración y aplicación de la Instrucción SCL
- Conocer el funcionamiento, configuración y aplicación de la Instrucción PID.
- Familiarizarse con el software RSView32.
- Crear una HMI (Interfase Humano Máquina) que interactúe con el PLC.

### 5.11.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 (5/03 o superior)
- ✓ 1 Módulo de Entradas Discretas.
- ✓ 1 Módulo de Salidas Discretas.
- ✓ 1 Módulo de Entradas Analógicas.
- ✓ 1 Módulo de Salidas Analógicas.
- ✓ 1 Computador Personal, con instalación de RSLogix500, RSLinx y RSView32.
- ✓ 1 Módulo Degem de simulación PCT-2 “AIR FLOW TEMPERATURE CONTROL SYSTEM”.
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

### 5.11.3. PLANTEAMIENTO

Diseñar e Implementar un sistema de control de temperatura para una estufa bacteriológica que trabaja en un rango de 50°C a 60°C. La técnica de control a utilizarse será la de *PI* ó *PID* (evaluar el desempeño de cada técnica). El control se lo hará a través de un PLC, al cual ingresa la señal proveniente de un sensor de temperatura que trabaja en un rango de 20°C a 70 °C, enviando una señal al PLC de 0V a 5V en relación lineal y proporcional con la temperatura de la estufa. El error en estado estacionario máximo permitido entre la temperatura real (PV) y la deseada (Setpoint) es de +/- 0.1°C.

Adicionalmente con el software RSView32, crear una HMI que permita interactuar al operador con el sistema controlador, la cual deberá tener las siguientes características:

Dos cuentas de usuario: INGENIERO (con contraseña) y OPERADOR

Pantalla Principal con el siguiente diseño referencial:

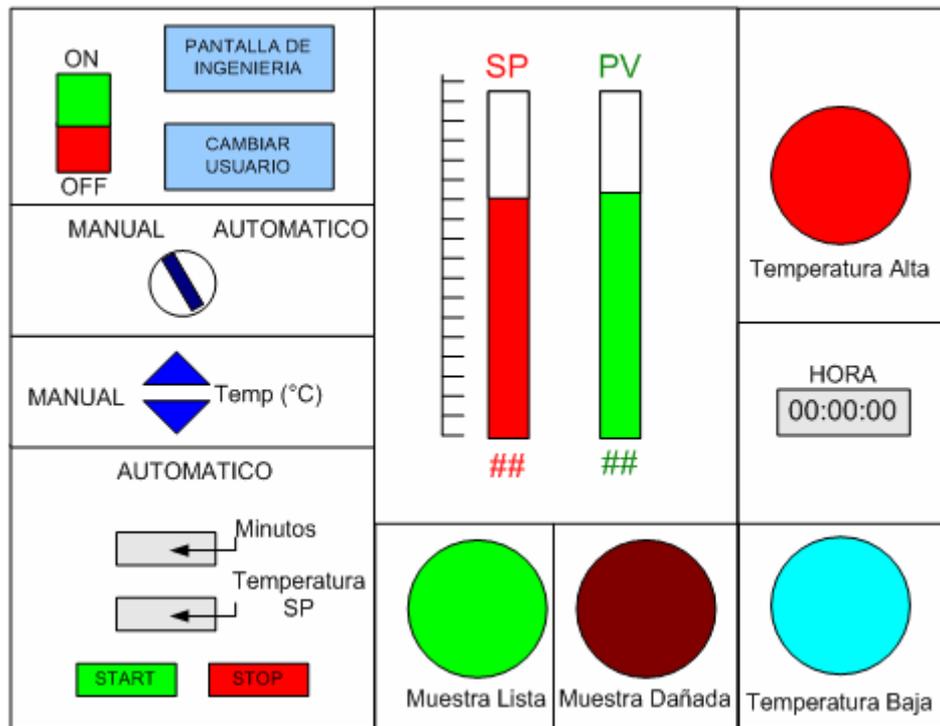


Figura. 5.63. Planteamiento Práctica #11

- ✓ El botón de “Pantalla de Ingeniería” despliega dicha pantalla únicamente si se encuentra trabajando con el usuario “INGENIERO”
- ✓ Las luces de alarma deberán estar asociadas a un archivo .wav para tener también alarma audible.
- ✓ Una vez transcurrido el tiempo ingresado, se deberá activar la alarma de “Muestra Lista”
- ✓ Si la variación entre la temperatura real (PV) y la deseada (Setpoint) es mayor a  $\pm 0.1^{\circ}\text{C}$  durante el transcurso del tiempo, se deberá activar la alarma “Muestra Dañada”.
- ✓ El botón START, en modo automático arrancará el tiempo y deberá ser presionado una vez que el controlador ya se haya estabilizado.

Pantalla de Ingeniería (Acceso permitido únicamente para el usuario INGENIERO)

- ✓ Control de activación del controlador.

- ✓ Visualización de estado de las luces de alarma.
- ✓ Selección de la Temperatura de referencia o Set Point.
- ✓ Ingreso de parámetros de calibración del controlador.
- ✓ Gráfica de tendencia (Trend) de Temperatura de la Estufa y Setpoint.
- ✓ Gráfica de tendencia (Trend) de % de Voltaje de Salida del controlador.

#### 5.11.4. DESARROLLO

##### 1. Configuración de driver en RSLinx

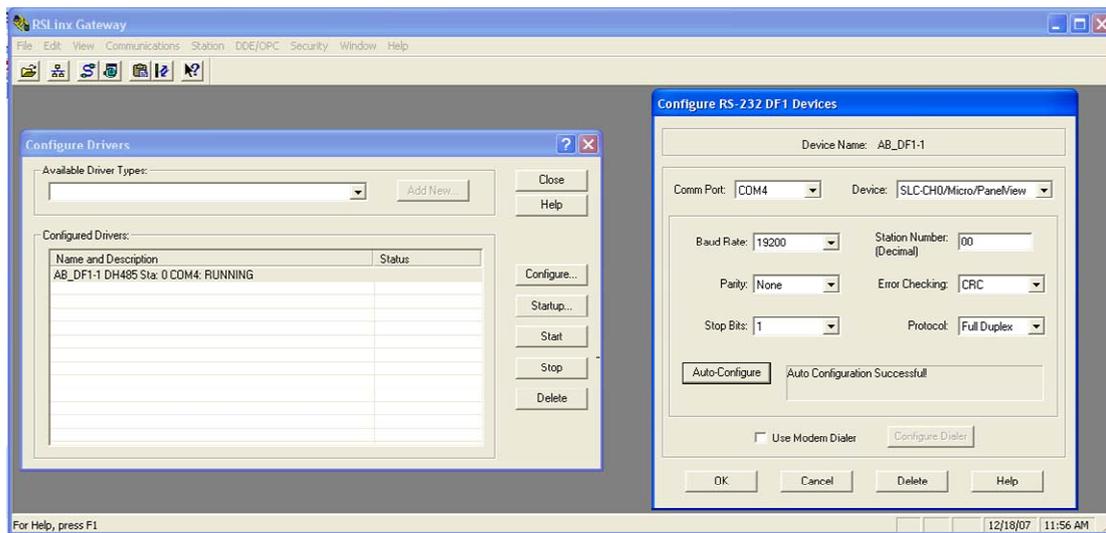


Figura. 5.64. Desarrollo Práctica #11 (1)

##### 2. Creación del archivo y configuración del PLC en el software RSLogix500.

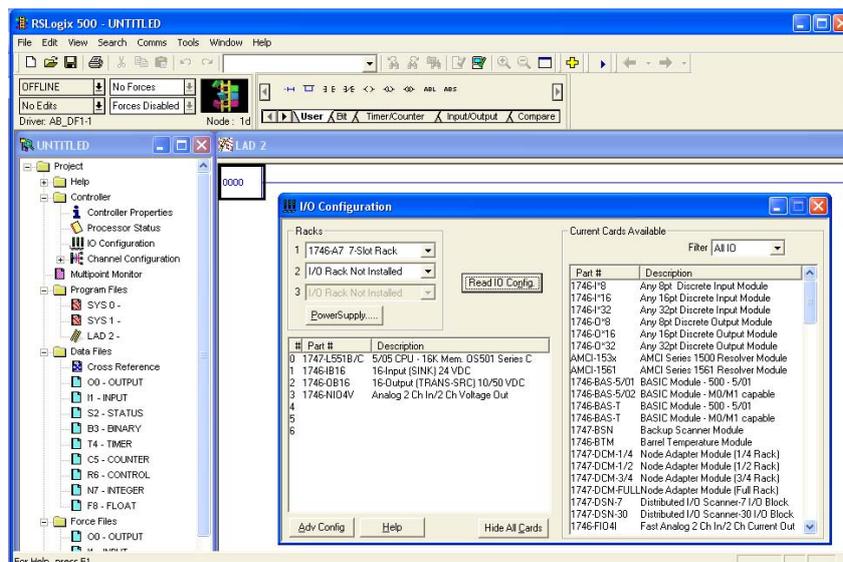
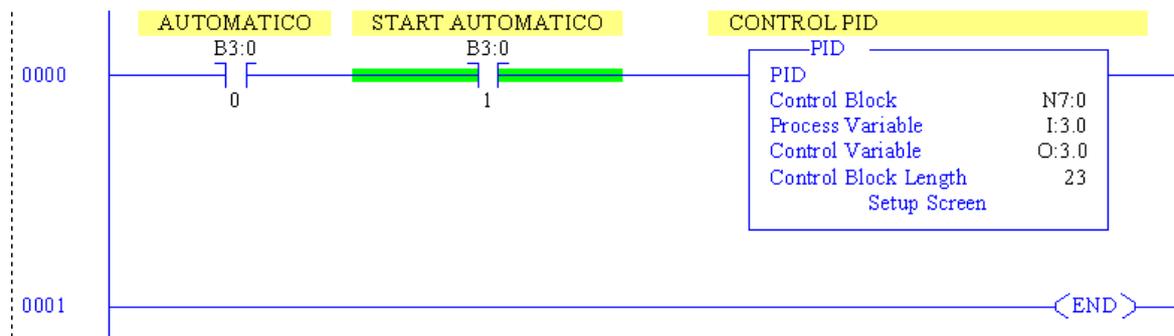


Figura. 5.65. Desarrollo Práctica #11 (2)

3. La programación Ladder que cumple con la lógica de control planteada en esta práctica es la siguiente:



**Figura. 5.67. Desarrollo Práctica #11 (3)**

4. La asignación de direcciones en el PLC de los diferentes registros, entradas y salidas utilizadas, se la hizo de acuerdo a la siguiente tabla:

Nombre	Tipo	Dirección
PV	Entrada Analógica	I:3.0
SALIDA ANALOGICA	Salida Analógica	O:3/0
AUTOMÁTICO	Bit	B3:0/0
START AUTOMATICO	Bit	B3:0/1
CONTROL PID	Entero	N7:0 - N7:22

**Tabla. 5.15. Direcciones utilizadas Práctica #11**

5. Abrir el software RSView32, configurar el canal de comunicación, configurar el nodo de comunicación.

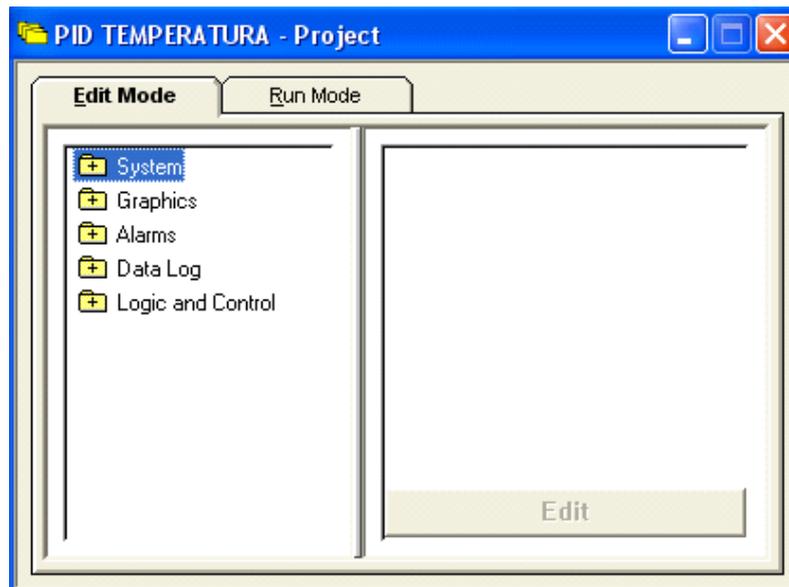


Figura. 5.68. Desarrollo Práctica #11 (4)

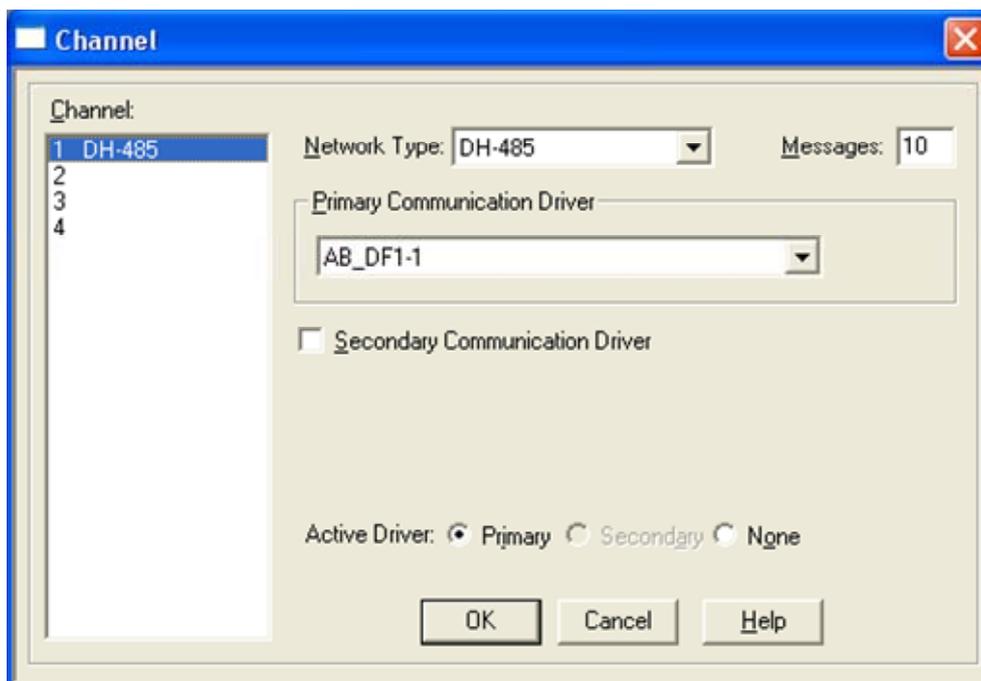


Figura. 5.69. Desarrollo Práctica #11 (5)

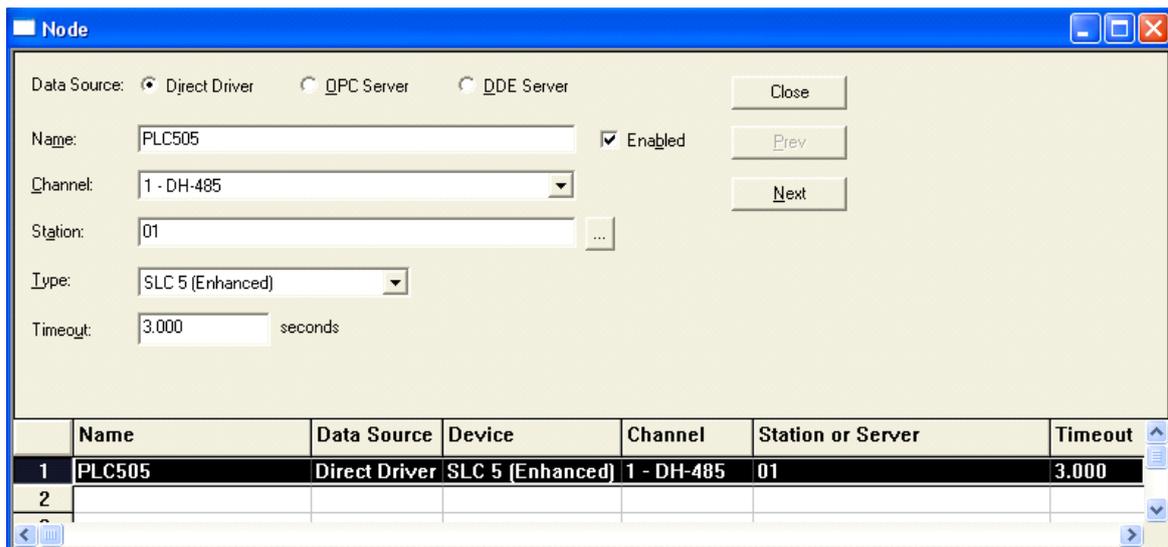


Figura. 5.70. Desarrollo Práctica #11 (6)

## 6. Creación de la HMI que interactúe con el PLC

### INTERFASE HUMANO MÁQUINA (HMI)

La HMI creada en base al esquema planteado consta de dos pantallas:

#### Pantalla Principal

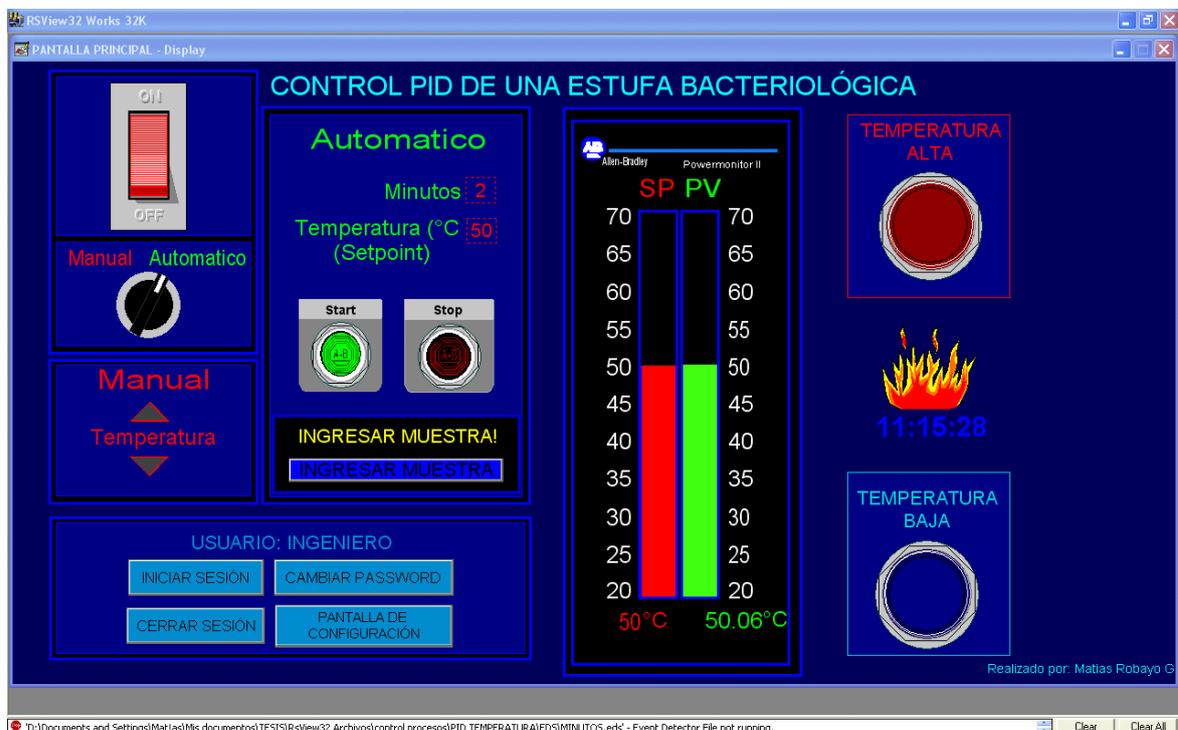


Figura. 5.71. Desarrollo Práctica #11 (7)

## Pantalla de Control



Figura. 5.72. Desarrollo Práctica #11 (8)

## TAGS

En la siguiente tabla se muestran los tags empleados en la programación de la HMI y sus características:

NOMBRE	TIPO	CARACTERÍSTICAS					
AUTOMATICO	Digital	Alarmed:	Security: *	Device	Node	Address:	Scan Class:
		No		Tag	Name:	B3:0/0	A
ENCENDER	Digital	Off Label:	On Label:	Memory	Initial		
		Off	On	Tag	Value: Off		
GANANCIA_Kc	Analog	Alarmed:	Security: *	Device	Node	Address:	Scan Class:
		No		Tag	Name:	N7:3	A
INGRESAR_MUESTRA	Digital	Minimum :	Maximum :	Scale :	Offset :	Units:	Data Type:
		0	32767	1	0		(Default)
INGRESAR_MUESTRA	Digital	Alarmed:	Security: *	Memory	Initial		
		No		Tag	Value: Off		
INGRESAR_MUESTRA	Digital	Off Label:	On Label:				
		Off	On				

MINUTOS	Analog	Alarmed: No	Security: *	Memory Tag	Initial Value: 0		
		Minimum : 0	Maximum : 100	Scale : 1	Offset : 0	Units:	Data Type: (Default)
MUESTRA_DANADA	Digital	Alarmed: No	Security: *	Memory Tag	Initial Value: Off		
		Off Label: Off	On Label: On				
MV	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:16	Scan Class: A
		Minimum : 0	Maximum : 100	Scale : 1	Offset : 0	Units:	Data Type: (Default)
PV	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: I:3.0	Scan Class: A
		Minimum : 20	Maximum : 70	Scale : 0.003050	Offset : 20	Units:	Data Type: (Default)
RATE_TIME	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:5	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
RESET_TIME	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:4	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
SALIDA_ANALOGICA	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: O:3.0	Scan Class: A
		Minimum : 0	Maximum : 12000	Scale : 1	Offset : 0	Units:	Data Type: Integer
SET_POINT	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:2	Scan Class: A
		Minimum : 20	Maximum : 70	Scale : 1	Offset : 0	Units:	Data Type: (Default)
START_AUTOMATICO	Digital	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: B3:0/1	Scan Class: A
		Off Label: Off	On Label: On				

**Tabla. 5.16. Tags utilizados Práctica #11**

Estos tags se encuentran configurados en la ventana *Tag Database* de acuerdo a las características que se encuentran en la tabla anterior, a continuación se muestra un ejemplo de esta configuración:

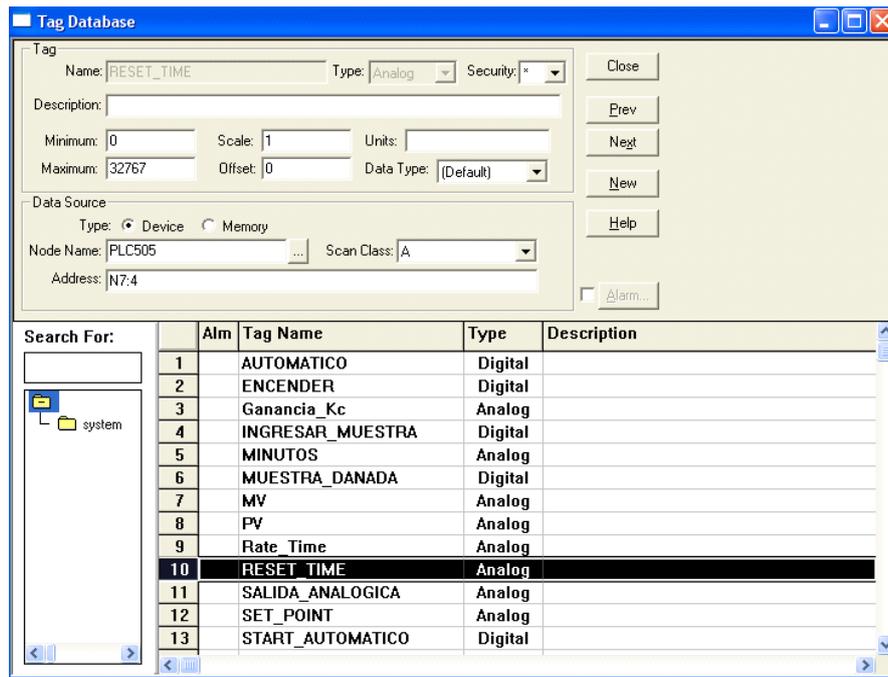


Figura. 5.73. Desarrollo Práctica #11 (9)

Como se puede ver en la figura 5.73, el tag RESET\_TIME es de tipo analógico y se encuentra asociado a la dirección N7:4 (Quinta posición del Control Block que va desde N7:0 hasta N7:22) a través del nodo PLC505, tiene un Mínimo de 0 y un Máximo de 32767 puesto que ese es el rango de valores permitidos para el Reset Time en la instrucción PID.

## USUARIOS

Como se muestra en la figura 5.74, esta HMI posee dos cuentas de usuario, “Default” e “Ingeniero”.

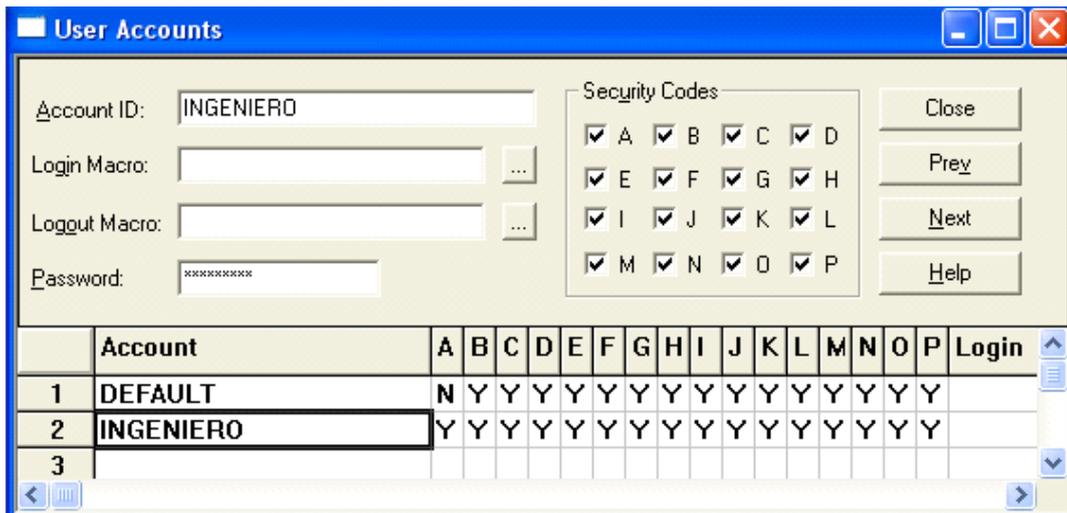


Figura. 5.74. Desarrollo Práctica #11 (10)

Default: Es el usuario que se muestra como “Operador” cuando se corre el programa, como se puede ver no posee acceso al código de seguridad A. Para restringir el acceso de este usuario a la Pantalla de Configuración, a dicha pantalla se le ha asignado el código de seguridad A, como se muestra en la figura 5.75.

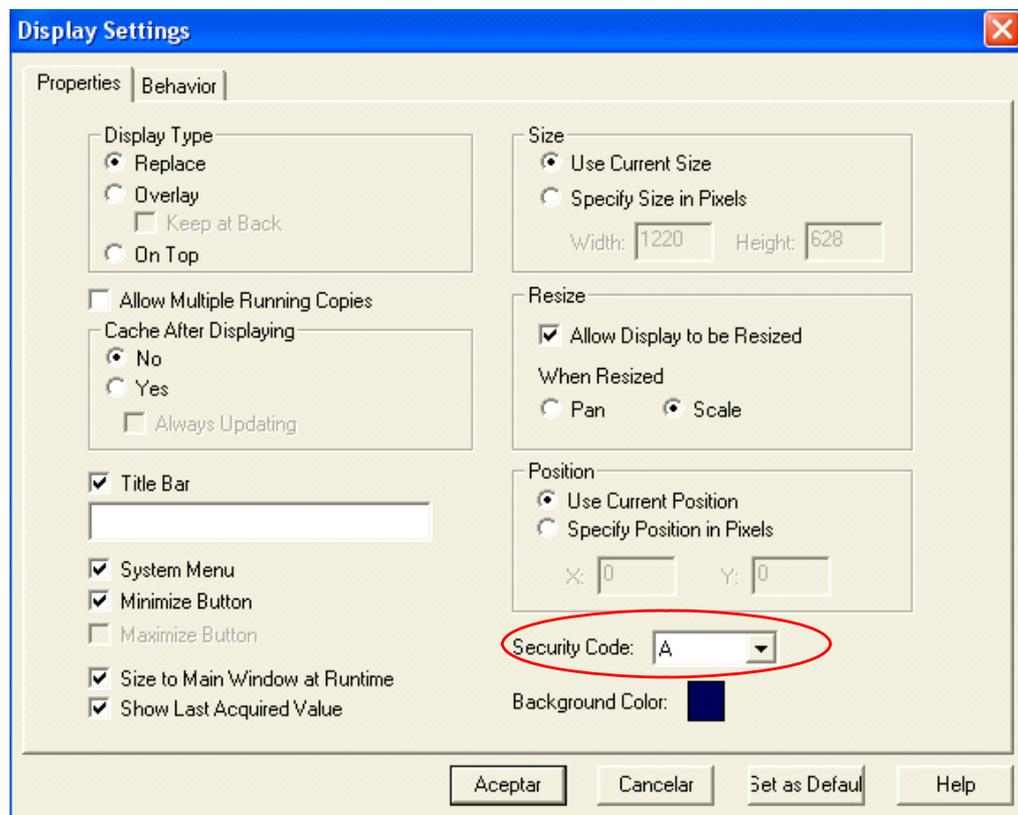


Figura. 5.75. Desarrollo Práctica #11 (11)

Ingeniero: Este usuario tiene acceso a todos los códigos de seguridad. Inicialmente tiene asignado el Password: *ingeniero*, el mismo que puede ser cambiado en la HMI (figura 5.76.).

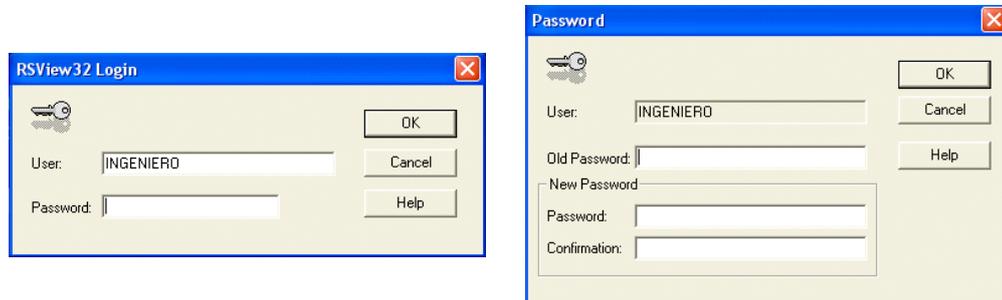


Figura. 5.76. Desarrollo Práctica #11 (12)

## EVENTOS

Como se muestra en la siguiente figura, para la programación de esta HMI se emplearon cuatro eventos: APAGA, ESTABLE, MINUTOS Y MUESTRA DANADA. Estos eventos se activan cada vez que se presione el botón START en modo automático, y se desactivan cuando la variable  $MINUTOS = 0$  o cuando se presiona el botón STOP en modo automático.

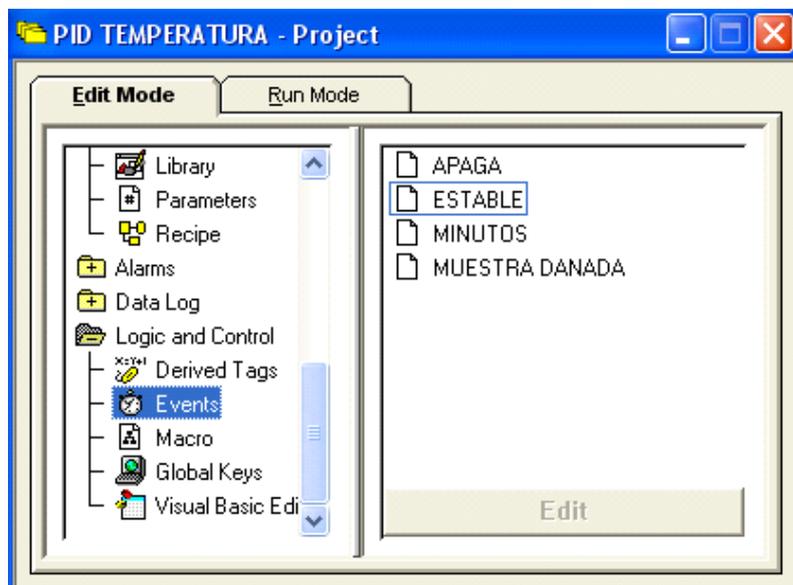


Figura. 5.77. Desarrollo Práctica #11 (13)

A continuación se describen cada uno de estos eventos

**APAGA:** Este evento apaga el sistema cuando la variable (tag) minutos es igual a cero, es decir cuando el tiempo programado para el secado de la madera, ya ha transcurrido.

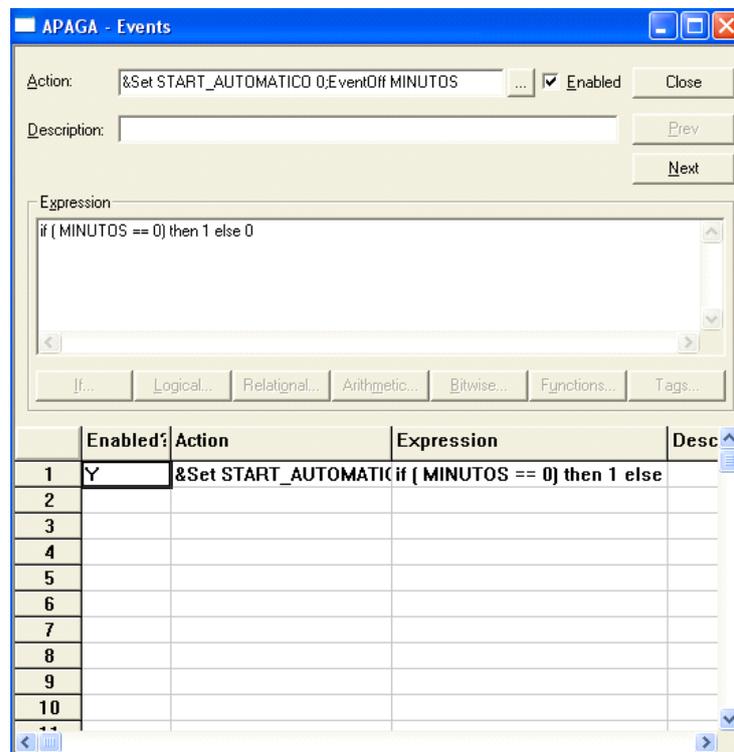


Figura. 5.78. Desarrollo Práctica #11 (14)

**ESTABLE:** Este evento despliega el mensaje de INGRESAR MUESTRA un minuto después de que se encienda el controlador. Tiempo suficiente para que se establezca el controlador.

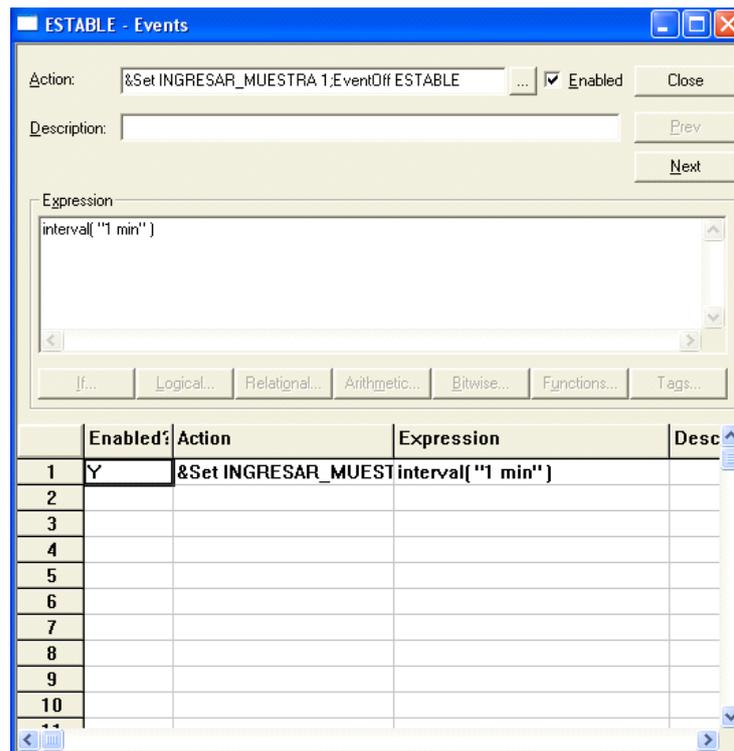


Figura. 5.79. Desarrollo Práctica #11 (15)

**MINUTOS:** Este evento reduce en 1 la variable MINUTOS cada vez que culmina un intervalo de 1 minuto.

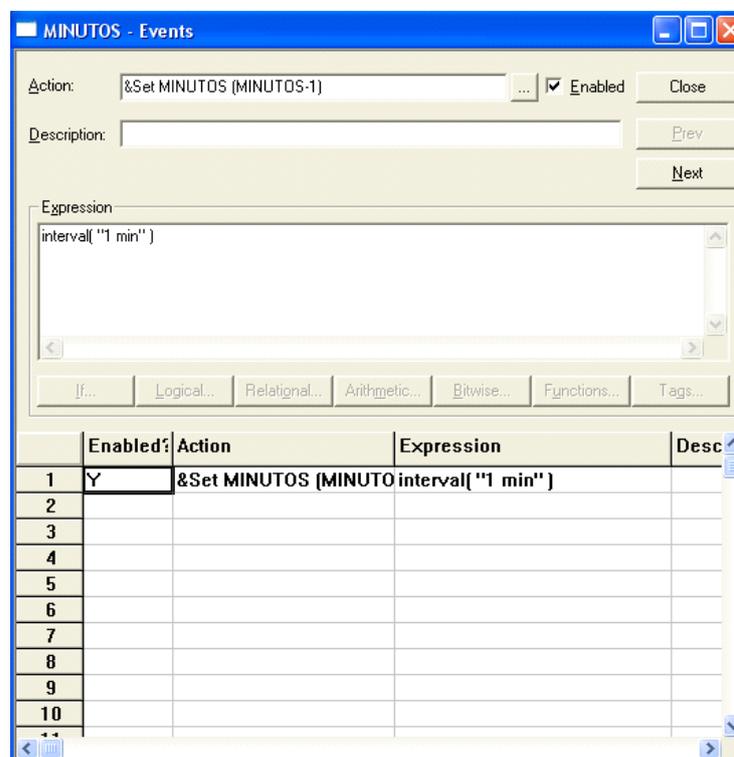


Figura. 5.80. Desarrollo Práctica #11 (16)

**MUESTRA DANADA:** Este evento apaga el controlador y despliega el mensaje de MUESTRA DAÑADA si la temperatura sale del rango de  $\pm 0.1$  °C con respecto al Setpoint, siempre y cuando ya se haya ingresado la muestra.

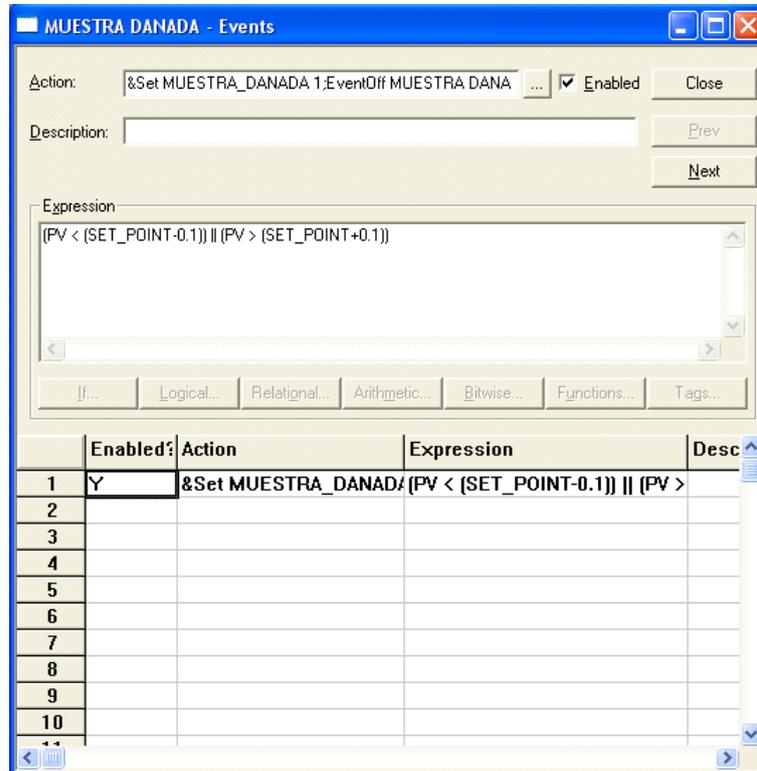


Figura. 5.81. Desarrollo Práctica #11 (17)

## 7. Sintonización de Controlador

Para sintonizar el controlador en esta práctica, se utilizó el método de lazo abierto o respuesta al escalón y se aplicó las fórmulas de Ziegler-Nichols.

Para lo cual, con el controlador trabajando en modo MANUAL se estableció una salida del 40% con lo que la temperatura se fijó en 45°C. Una vez estable la temperatura se cambió la salida al 50% con lo que se obtuvo la curva de respuesta al escalón de la planta, con ayuda de un Trend en RSVIEW32 como se puede ver en la figura 5.82.

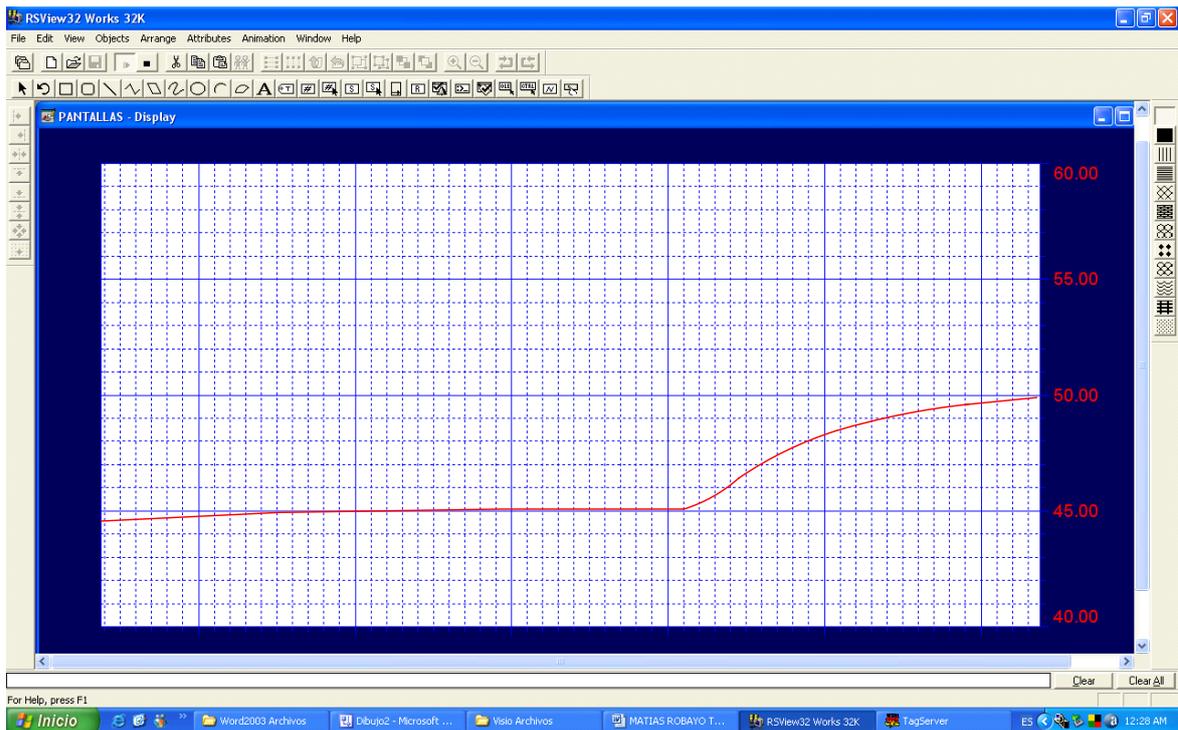


Figura. 5.82. Desarrollo Práctica #11 (18)

Tomando en cuenta que cada división vertical, representa un intervalo de 8 segundos, y cada división horizontal representa un incremento de  $1^{\circ}\text{C}$ , se obtuvieron los siguientes valores:

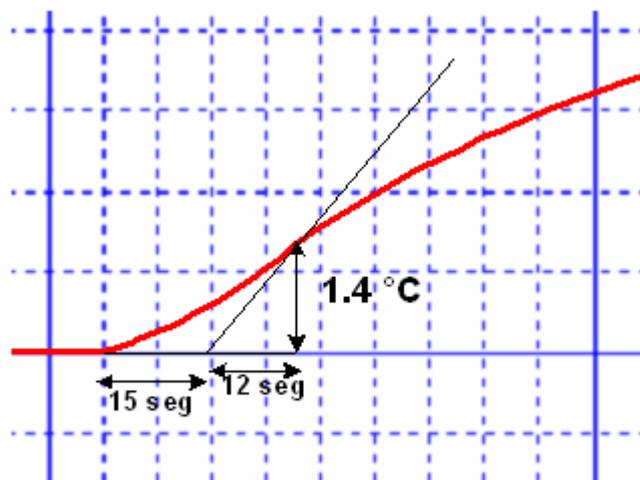


Figura. 5.83. Desarrollo Práctica #11 (19)

Es decir:

$\Delta m = 10\%$  (Variación del escalón de 40% al 50% de la salida del controlador)

$L = 15 \text{ seg} \Rightarrow 0.25 \text{ min}$

$$B = 12 \text{ seg} \Rightarrow 0.20 \text{ min}$$

$A = 1.4^{\circ}\text{C} \Rightarrow 2.8\%$  (Tomando en cuenta que la variación máxima es de  $50^{\circ}\text{C}$  es decir de  $20^{\circ}\text{C}$  a  $70^{\circ}\text{C}$ )

Con estos datos se calculó:

$$R_r = \frac{A}{B} = \frac{2.8\%}{0.2 \text{ min}} = 14 \text{ \%}/\text{min}$$

Aplicando las fórmulas de Ziegler-Nichols se obtienen los siguientes datos:

<b>Ziegler-Nichols. Método de respuesta al escalón</b>				
<b>Control</b>	<b><i>PB</i></b>	<b><i>Kc</i></b>	<b><i>Ti</i></b>	<b><i>Td</i></b>
<b>P</b>	$\frac{100 \cdot R_r \cdot L}{\Delta m} = 35\%$	$\frac{100}{PB} = 2.857$		
<b>PI</b>	$\frac{111 \cdot R_r \cdot L}{\Delta m} = 38.85\%$	$\frac{100}{PB} = 2.574$	$3.33 \cdot L = 0.8325$	
<b>PID</b>	$\frac{83 \cdot R_r \cdot L}{\Delta m} = 29.05\%$	$\frac{100}{PB} = 3.442$	$2 \cdot L = 0.5$	$0.5 \cdot L = 0.125$

**Tabla. 5.17. Valores de sintonización PID**

Finalmente, se realizaron las pruebas respectivas del desempeño de cada una de los tres tipos de controladores y se determinó que el controlador PID tenía un mejor desempeño.

En la figura 5.84 se muestra el desempeño obtenido del controlador PID cuando se varía el setpoint.



Figura. 5.84. Desarrollo Práctica #11 (20)

Los valores finales de sintonización del controlador para esta planta son:

$$Kc = 3.5 \quad Ti = 0.5 \quad Td = 0.12$$

8. Finalmente, se realizaron las pruebas necesarias para verificar que el controlador funciona correctamente de acuerdo al problema planteado.

#### 5.11.5. PRUEBAS Y RESULTADOS

Durante el desarrollo de la práctica se realizaron las pruebas respectivas tanto del software (programación del PLC y de la HMI) como del hardware (Conexión del PLC y del Módulo Degem de simulación “AIR FLOW TEMPERATURE CONTROL SYSTEM PCT-2”), a través de las cuales se verificó que el PLC y la HMI trabajan correctamente de acuerdo a la lógica del problema planteado en esta práctica.

#### 5.11.6. CONCLUSIONES Y RECOMENDACIONES

##### Conclusiones

- En esta práctica, el controlador PID tiene un mejor desempeño que el PI, debido a las características propias de la planta.

**Recomendaciones**

- Se recomienda para la sintonización, obtener la curva de respuesta al escalón de la planta en el rango de temperatura en el cual el controlador va a trabajar normalmente es decir entre  $50^{\circ}$  y  $60^{\circ}$  para esta práctica.

## 5.12. CONTROL DE PROCESOS 3: CONTROL PID DE VELOCIDAD

### 5.12.1. OBJETIVOS

- Evaluar el desempeño de las técnicas P, PI y PID en el control de velocidad de un motor DC
- Conocer el funcionamiento, configuración y aplicación de la Instrucción SCL
- Conocer el funcionamiento, configuración y aplicación de la Instrucción PID.
- Familiarizarse con el software RSView32.
- Crear una HMI (Interfase Humano Máquina) que interactúe con el PLC.

### 5.12.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 (5/03 o superior)
- ✓ 1 Módulo de Entradas Discretas.
- ✓ 1 Módulo de Salidas Discretas.
- ✓ 1 Módulo de Entradas Analógicas.
- ✓ 1 Módulo de Salidas Analógicas.
- ✓ 1 Computador Personal, con instalación de RSLogix500, RSLinx y RSView32.
- ✓ 1 Módulo Degem de simulación PCT-1 “DC SPEED CONTROL SYSTEM”
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

### 5.12.3. PLANTEAMIENTO

Diseñar e Implementar un sistema de control de velocidad para una bomba, que trabaja en el rango de 0 - 3000 RPM (0-10VDC). La técnica de control a utilizarse será la de *PI* ó *PID* (evaluar el desempeño de cada técnica). El control se lo hará a través de un PLC, al cual ingresa la señal proveniente de un sensor el cual envía una señal al PLC de 0V a 6VDC en relación lineal a la velocidad de la bomba. El error en estado estacionario máximo permitido es de +/- 20 RPM.

Adicionalmente con el software RSView32, crear una HMI que permita interactuar al operador con el sistema controlador, la cual deberá tener las siguientes características:

Dos cuentas de usuario: INGENIERO (con contraseña) y OPERADOR

Pantalla Principal con el siguiente diseño referencial:

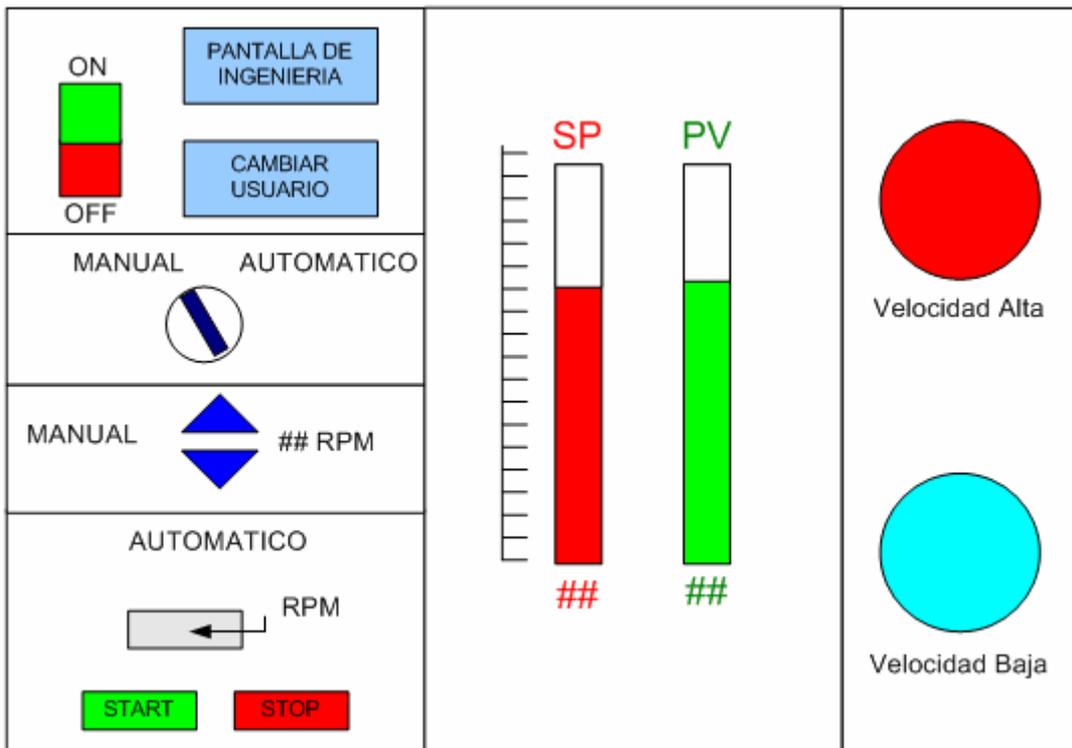


Figura. 5.85. Planteamiento Práctica #12

- ✓ El botón de “Pantalla de Ingeniería” despliega dicha pantalla únicamente si se encuentra trabajando con el usuario “INGENIERO”
- ✓ Las luces de alarma deberán estar asociadas a un archivo .wav para tener también alarma audible.

Pantalla de Ingeniería (Acceso permitido únicamente para el usuario INGENIERO)

- ✓ Control de activación del controlador.
- ✓ Visualización de estado de las luces de alarma.
- ✓ Selección de la Velocidad de referencia o Set Point.
- ✓ Ingreso de parámetros de calibración del controlador.
- ✓ Gráfica de tendencia (Trend) de velocidad de la Bomba y Setpoint.
- ✓ Gráfica de tendencia (Trend) de Voltaje de Salida del controlador.

#### 5.12.4. DESARROLLO

1. Configuración de driver en RSLinx

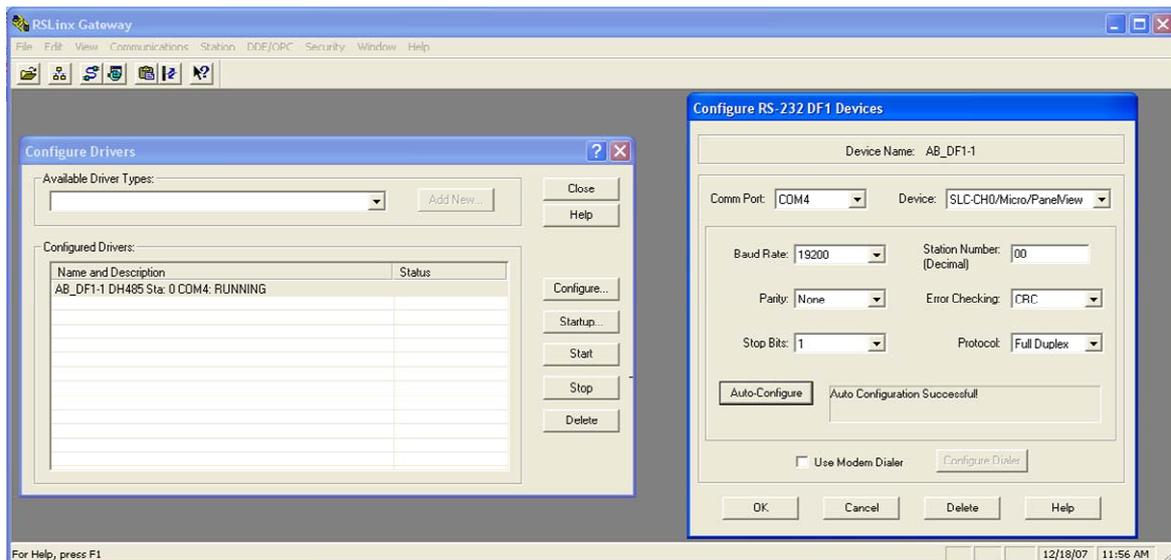


Figura. 5.86. Desarrollo Práctica #12 (1)

2. Creación del archivo y configuración del PLC en el software RSLogix500.

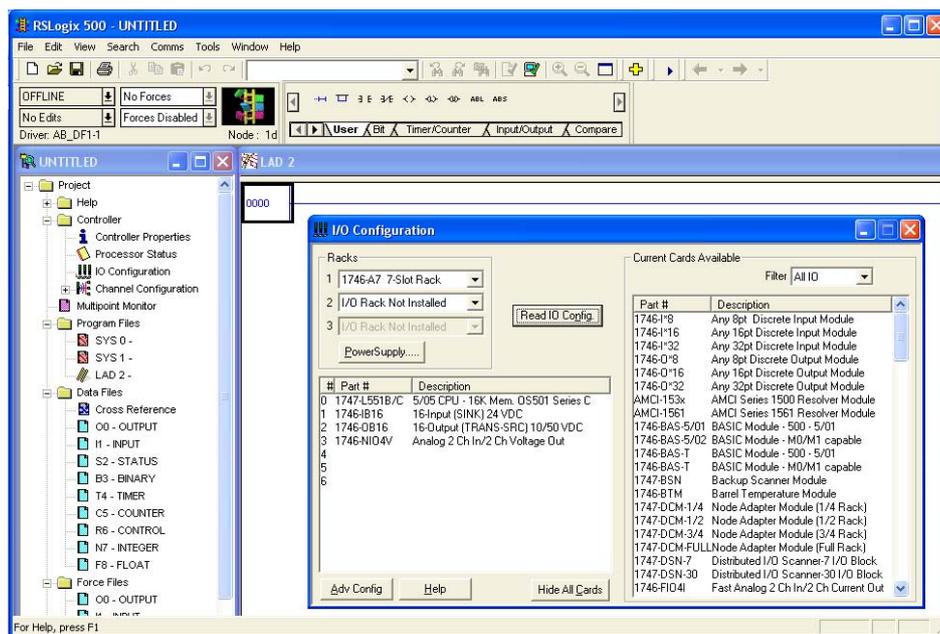


Figura. 5.87. Desarrollo Práctica #12 (2)

3. La programación Ladder que cumple con la lógica de control planteada en esta práctica es la siguiente:

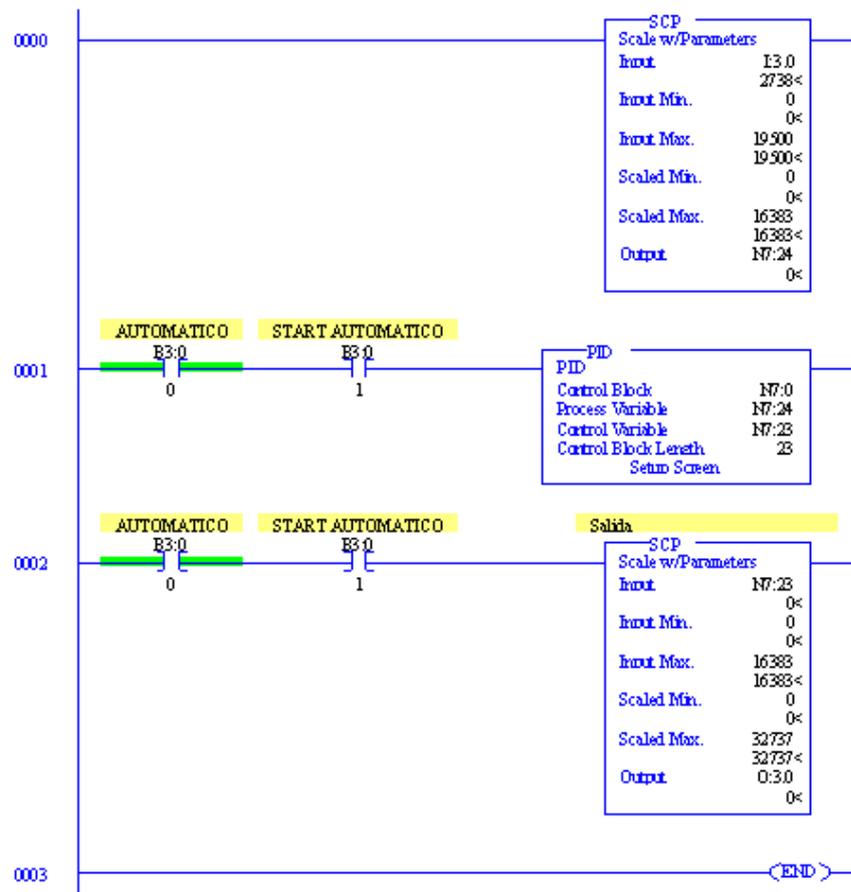


Figura. 5.88. Desarrollo Práctica #12 (3)

4. La asignación de direcciones en el PLC de los diferentes registros, entradas y salidas utilizadas, se la hizo de acuerdo a la siguiente tabla:

Nombre	Tipo	Dirección
PV	Entrada Analógica	I:3.0
SALIDA ANALOGICA	Salida Analógica	O:3/0
AUTOMÁTICO	Bit	B3:0/0
START AUTOMATICO	Bit	B3:0/1
CONTROL PID	Entero	N7:0 - N7:22
SALIDA PID	Entero	N7:23
ENTRADA ESCALADA	Entero	N7:24

Tabla. 5.18. Direcciones utilizadas Práctica #12

5. Abrir el software RSView32, configurar el canal de comunicación, configurar el nodo de comunicación.

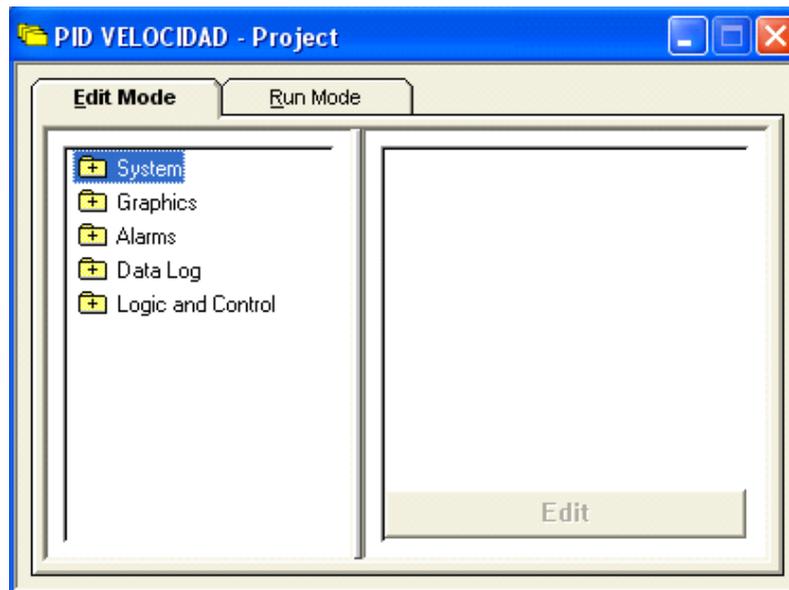


Figura. 5.89. Desarrollo Práctica #12 (4)

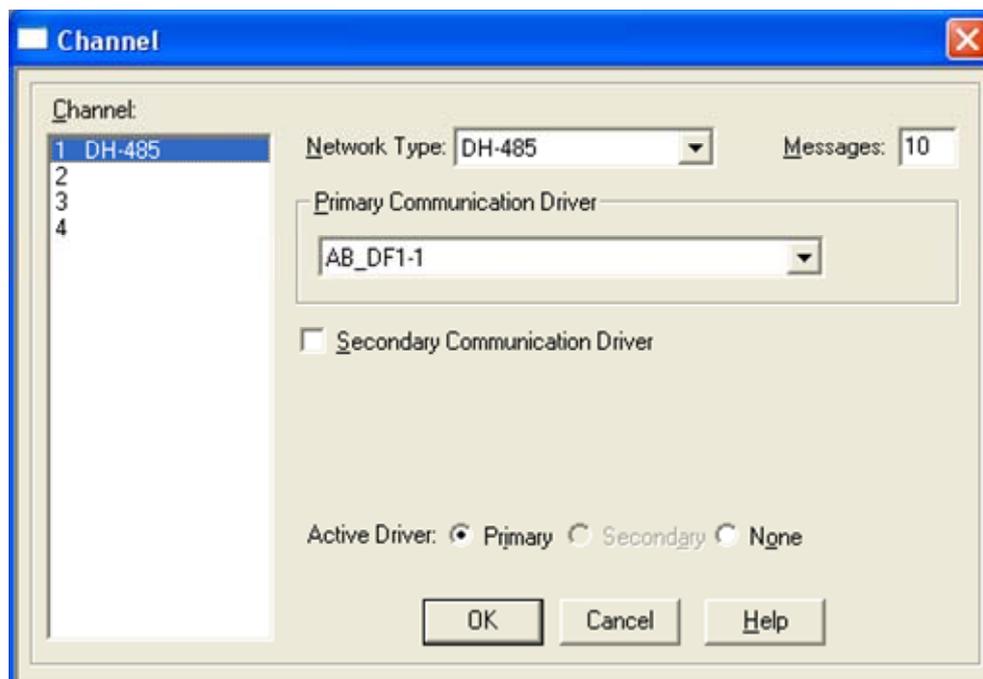


Figura. 5.90. Desarrollo Práctica #12 (5)

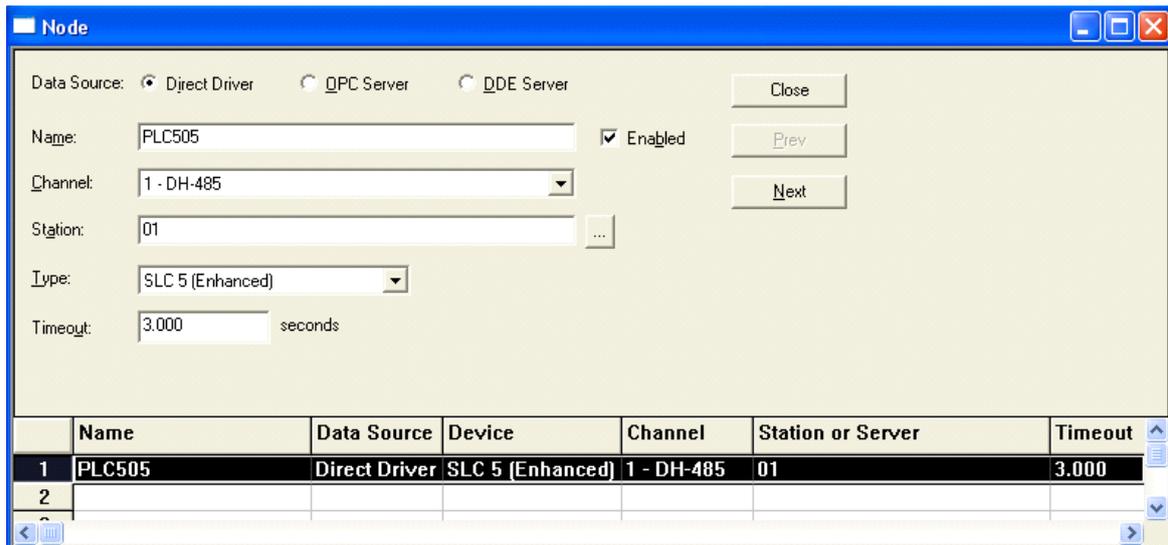


Figura. 5.91. Desarrollo Práctica #12 (6)

## 6. Creación de la HMI que interactúe con el PLC

### INTERFASE HUMANO MÁQUINA (HMI)

La HMI creada en base al esquema planteado consta de dos pantallas:

#### Pantalla Principal



Figura. 5.92. Desarrollo Práctica #12 (7)

**Pantalla de Control**



Figura. 5.93. Desarrollo Práctica #12 (8)

**TAGS**

En la siguiente tabla se muestran los tags empleados en la programación de la HMI y sus características:

NOMBRE	TIPO	CARACTERÍSTICAS					
AUTOMATICO	Digital	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: B3:0/0	Scan Class: A
		Off Label: Off	On Label: On				
ENCENDER	Digital	Alarmed: No	Security: *	Memory Tag	Initial Value: Off		
		Off Label: Off	On Label: On				
Ganancia_Kc	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:3	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
MV	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:16	Scan Class: A
		Minimum :	Maximum :	Scale : 1	Offset : 0	Units:	Data Type:

		0	100				(Default)
PV	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:24	Scan Class: A
		Minimum : 0	Maximum : 3000	Scale : 0.183000	Offset : 0	Units:	Data Type: (Default)
RATE_TIME_Td	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:5	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
RESET_TIME_Ti	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:4	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: Integer
SALIDA_ANALOGICA	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: O:3.0	Scan Class: A
		Minimum : 0	Maximum : 32737	Scale : 1	Offset : 0	Units:	Data Type: Integer
SET_POINT	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:2	Scan Class: A
		Minimum : 0	Maximum : 3000	Scale : 1	Offset : 0	Units:	Data Type: (Default)
START_AUTOMATICO	Digital	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: B3:0/1	Scan Class: A
		Off Label: Off	On Label: On				

**Tabla. 5.19. Tags utilizados Práctica #12**

Estos tags se encuentran configurados en la ventana *Tag Database* de acuerdo a las características que se encuentran en la tabla anterior, a continuación se muestra un ejemplo de esta configuración:

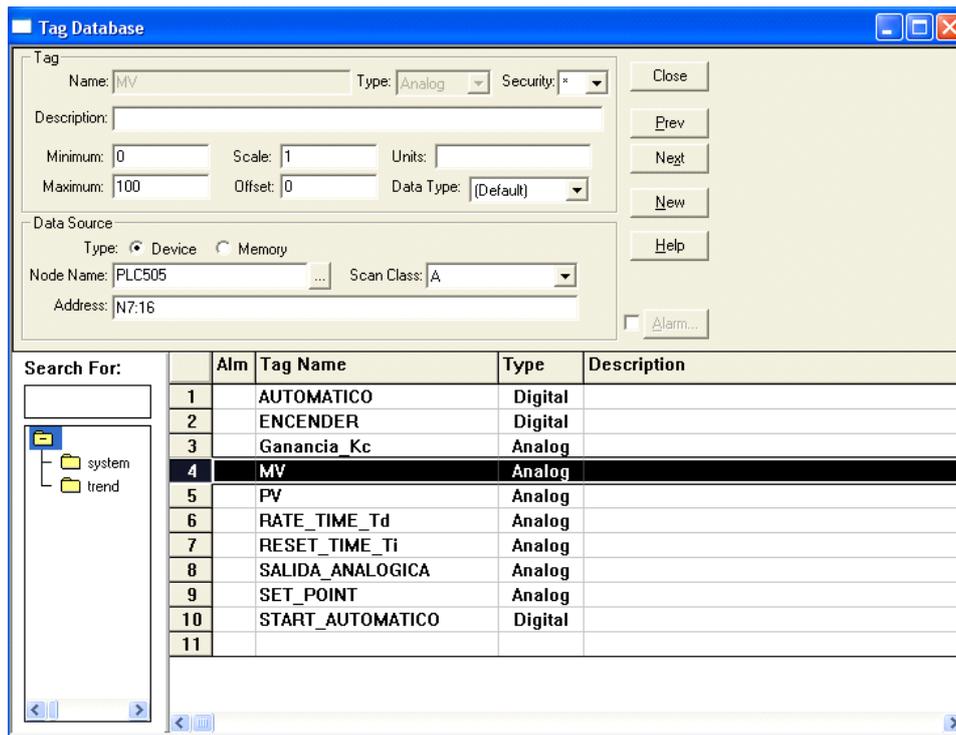


Figura. 5.94. Desarrollo Práctica #12 (9)

Como se puede ver en la figura anterior, el tag MV es de tipo analógico y se encuentra asociado a la dirección N7:16 (17<sup>va</sup> posición del Control Block que va desde N7:0 hasta N7:22) a través del nodo PLC505, tiene un Mínimo de 0 y un Máximo de 100 puesto que ese es el rango de valores de la variable MV (0 a 100%) en la instrucción PID.

**USUARIOS**

Como se muestra en la figura 5.95 , esta HMI posee dos cuentas de usuario, “Default” e “Ingeniero”.

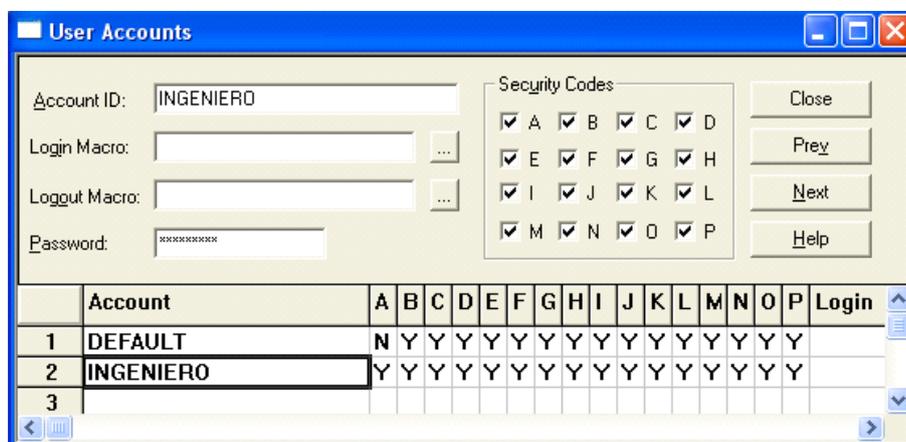


Figura. 5.95. Desarrollo Práctica #12 (10)

**Default:** Es el usuario que se muestra como “Operador” cuando se corre el programa, como se puede ver no posee acceso al código de seguridad A. Para restringir el acceso de este usuario a la Pantalla de Configuración, a dicha pantalla se le ha asignado el código de seguridad A, como se muestra en la figura 5.96.

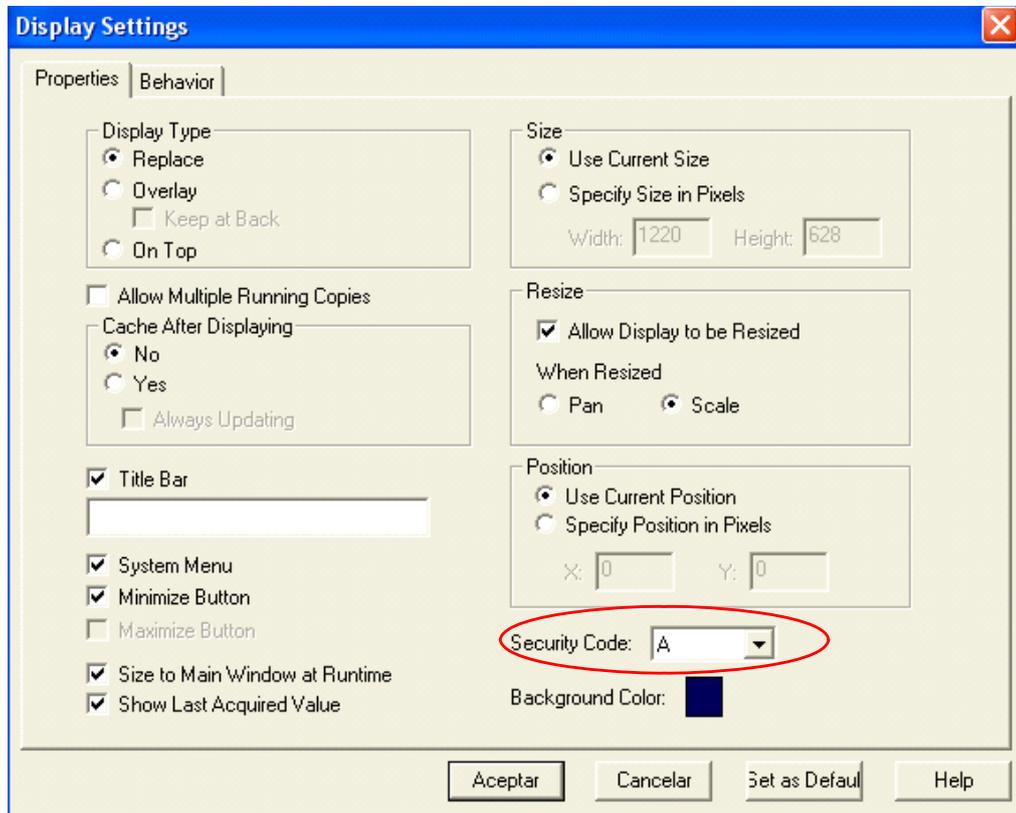


Figura. 5.96. Desarrollo Práctica #12 (11)

**Ingeniero:** Este usuario tiene acceso a todos los códigos de seguridad. Inicialmente tiene asignado el Password: *ingeniero*, el mismo que puede ser cambiado en la HMI (figura 5.97).

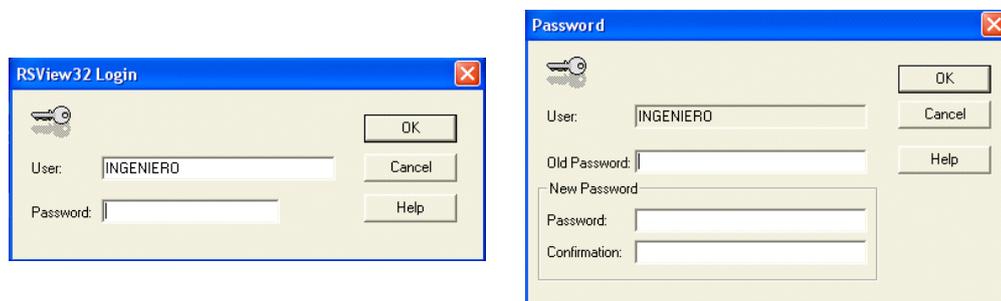


Figura. 5.97. Desarrollo Práctica #12 (12)

## 7. Sintonización de Controlador

Debido a la alta velocidad de respuesta de la planta, se aplicó el método de respuesta en frecuencia (Lazo cerrado) en lugar del método de respuesta al escalón (Lazo abierto), para determinar las constantes de sintonización.

Con software RSView32, se creó una HMI que permita ingresar los parámetros  $K_c$ ,  $T_i$  y  $T_d$ . Dicha pantalla también permite visualizar gráficamente el valor de PV (velocidad del motor) a través de un trend.

En el PLC se grabó el programa ladder mostrado en la sección 3 del desarrollo de esta práctica y se conectó el controlador PID al sistema con las funciones integral y derivativa eliminadas de modo que solo sea un control proporcional.

Posteriormente, con el PLC y la HMI corriendo, se aumentó lentamente la ganancia del controlador hasta un valor de  $K_u=15.5$  con el que se presentó un comportamiento oscilatorio como se muestra en la figura 5.98.

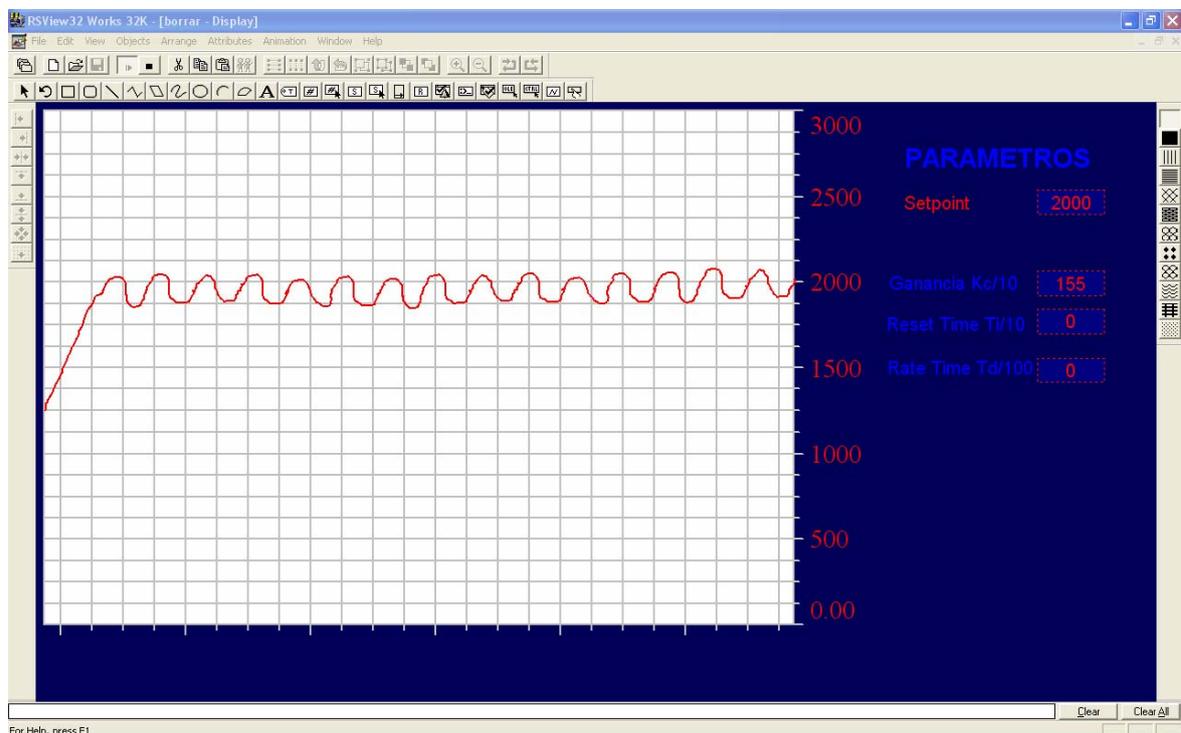


Figura. 5.98. Desarrollo Práctica #12 (13)

De donde se pudo obtener los siguientes valores:

$K_u=15.5$  (Cabe recordar que el controlador divide para 10 el valor ingresado en la HMI)

$T_u=7$  seg (el Trend se encuentra configurado de tal modo que cada división del eje horizontal equivale a un tiempo de 5 seg)

Debido a que los parámetros de sintonización del PLC se ingresan en minutos el valor del periodo es:

$T_u= 0.1166$  minutos

Una vez determinados los 2 valores  $K_u$  y  $T_u$  se utilizaron las formulas de Ziegler-Nichols con lo que se obtuvieron los siguientes valores:

<b>Ziegler-Nichols. Método de respuesta en frecuencia</b>			
<b>Control</b>	<b><math>K_c</math></b>	<b><math>T_i</math></b>	<b><math>T_d</math></b>
<b>P</b>	$0.5 \cdot K_u = 7.75$		
<b>PI</b>	$0.45 \cdot K_u = 6.975$	$\frac{T_u}{1.2} = 0.097$	
<b>PID</b>	$0.6 \cdot K_u = 9.3$	$0.5 \cdot T_u = 0.0583$	$\frac{T_u}{8} = 0.0145$

**Tabla. 5.20. Valores de sintonización PID**

Finalmente, se realizaron las pruebas respectivas del desempeño de cada una de los tres tipos de controladores y se determinó que el controlador PI tenía un mejor desempeño.

En la figura 5.99. se muestra el desempeño obtenido del controlador PI cuando se varía el setpoint.

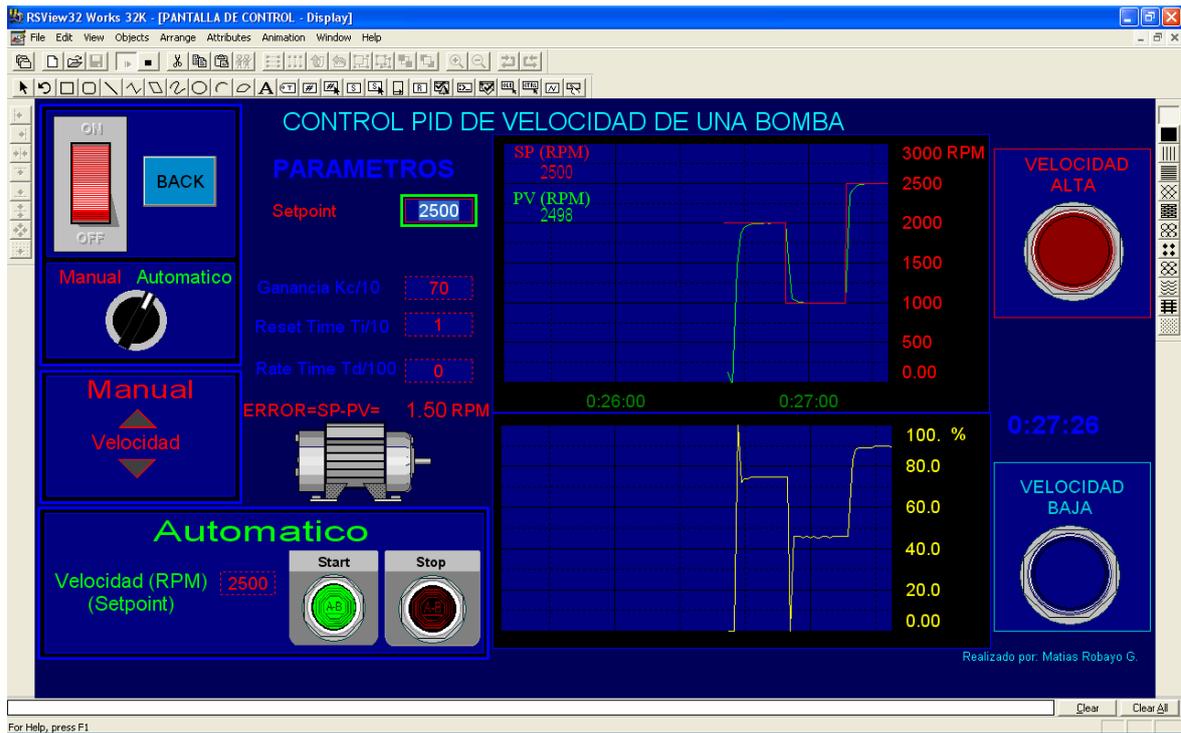


Figura. 5.99. Desarrollo Práctica #12 (14)

Los valores finales de sintonización del controlador para esta planta son:

$$Kc = 7$$

$$Ti = 0.1$$

$$Td = 0$$

8. Finalmente, se realizaron las pruebas necesarias para verificar que el controlador funciona correctamente de acuerdo al problema planteado.

#### 5.12.5. PRUEBAS Y RESULTADOS

Durante el desarrollo de la práctica se realizaron las pruebas respectivas tanto del software (programación del PLC y de la HMI) como del hardware (Conexión del PLC y del Módulo Degem de simulación PCT-1 “DC SPEED CONTROL SYSTEM”), a través de las cuales se verificó que el PLC y la HMI trabajan correctamente de acuerdo a la lógica del problema planteado en esta práctica.

### **5.12.6. CONCLUSIONES Y RECOMENDACIONES**

#### **Conclusiones**

- En esta práctica, el controlador PI tiene un mejor desempeño que el PID, debido a las características propias de la planta.
- Cuando el tiempo de reacción de la planta es muy rápido, no se puede aplicar el método de respuesta al escalón para sintonizar el controlador.

#### **Recomendaciones**

- Se recomienda verificar previamente que el módulo de simulación se encuentre trabajando correctamente y establecer las condiciones de funcionamiento del mismo, puesto que estas pueden variar respecto al módulo empleado.

### 5.13. CONTROL DE PROCESOS 4: CONTROL DE RAZÓN

#### 5.13.1. OBJETIVOS

- Familiarizarse con la técnica de Control de Razón.
- Crear una HMI (Interfase Humano Máquina) que interactúe con el PLC.

#### 5.13.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 (5/03 o superior)
- ✓ 1 Módulo de Entradas Discretas.
- ✓ 1 Módulo de Salidas Discretas.
- ✓ 1 Módulo de Entradas Analógicas.
- ✓ 1 Módulo de Salidas Analógicas.
- ✓ 1 Computador Personal, con instalación de RSLogix500, RSLinx y RSView32.
- ✓ 2 Módulos Degem de simulación PCT-1 “DC SPEED CONTROL SYSTEM”
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

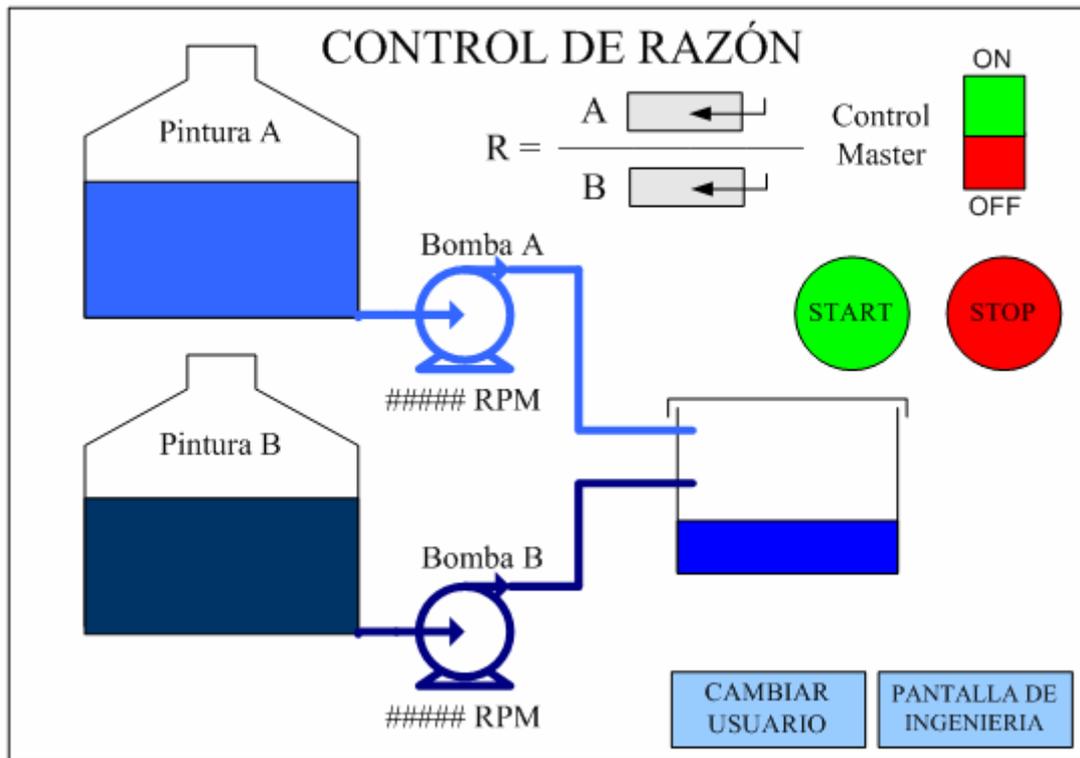
#### 5.13.3. PLANTEAMIENTO

En una planta de Pintura, se desea controlar la razón de mezcla de dos diferentes tipos de pintura, para lo cual se debe controlar la velocidad (RPM) de dos bombas que extraen la pintura de los tanques de almacenamiento. Empleando la técnica de Control de Razón, diseñar e implementar un sistema de control de velocidad para ambas bombas que permita ingresar la razón de mezcla deseada. En ambas bombas se cuenta con sensores de Velocidad que envían al controlador una señal de 0 a 6 VDC en relación lineal a la velocidad de la bomba (0 a 3000 RPM).

Adicionalmente con el software RSView32, crear una HMI que permita interactuar al operador con el sistema controlador, la cual deberá tener las siguientes características:

Dos cuentas de usuario: INGENIERO (con contraseña) y OPERADOR

Pantalla Principal con el siguiente esquema referencial:



**Figura. 5.100. Planteamiento Práctica #13**

- ✓ El botón de “Pantalla de Ingeniería” despliega dicha pantalla únicamente si se encuentra trabajando con el usuario “INGENIERO”

Pantalla de Ingeniería (Acceso permitido únicamente para el usuario INGENIERO)

- ✓ Control de activación del controlador.
- ✓ Selección de la Razón De Mezcla.
- ✓ Ingreso de parámetros de calibración del controlador.
- ✓ Gráfica de tendencia (Trend) de Velocidad de la Bomba 1 y Bomba 2.
- ✓ Gráfica de tendencia (Trend) de Voltaje de Salida del controlador.

#### 5.13.4. DESARROLLO

1. Configuración de driver en RSLinx

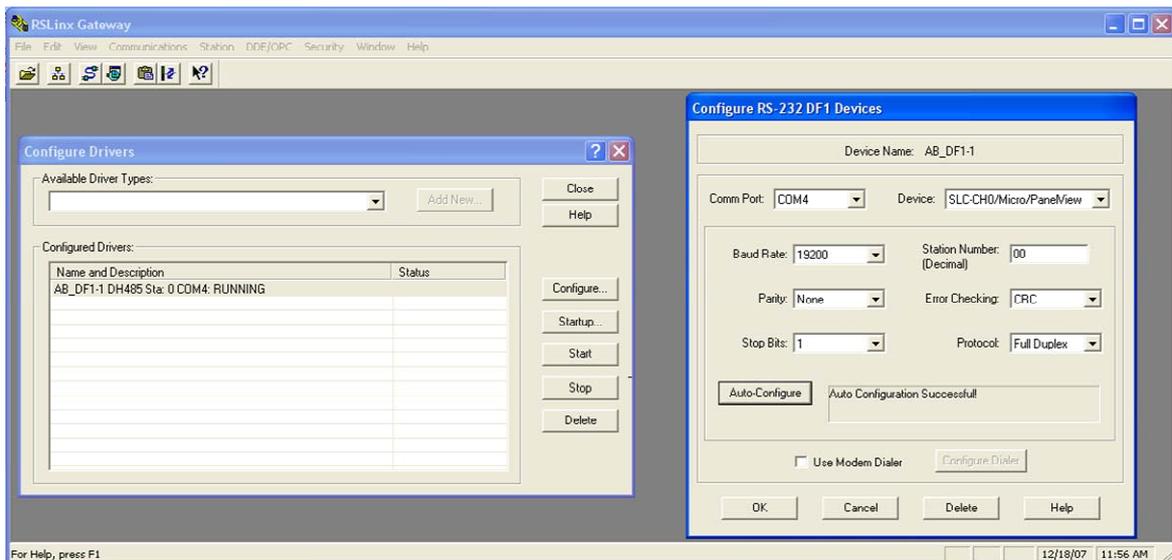


Figura. 5.101. Desarrollo Práctica #13 (1)

2. Creación del archivo y configuración del PLC en el software RSLogix500.

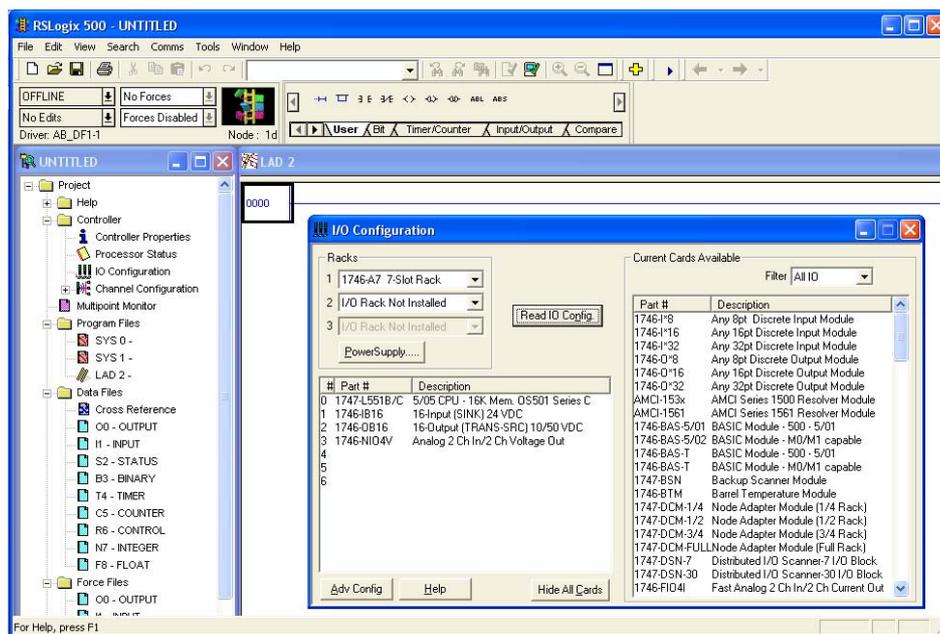


Figura. 5.102. Desarrollo Práctica #13 (2)

3. La programación Ladder que cumple con la lógica de control planteada en esta práctica es la siguiente:

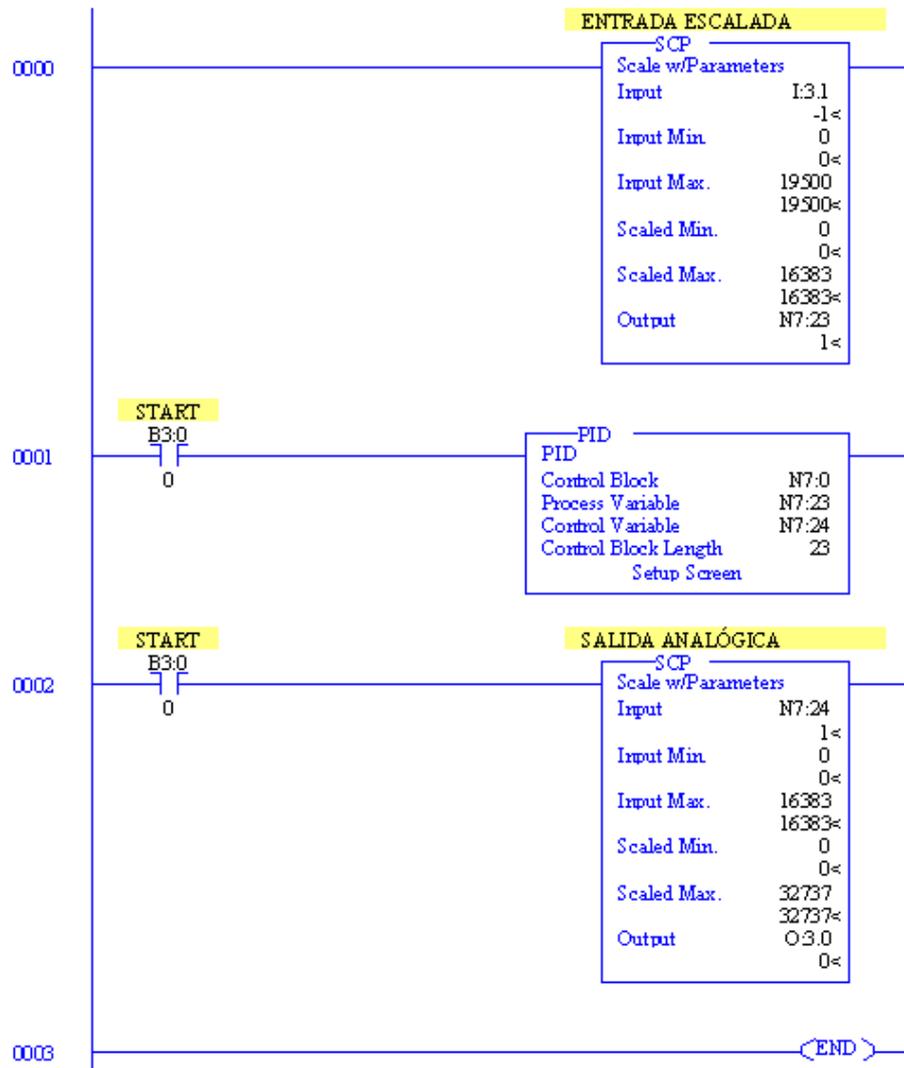


Figura. 5.103. Desarrollo Práctica #13 (3)

4. La asignación de direcciones en el PLC de los diferentes registros, entradas y salidas utilizadas, se la hizo de acuerdo a la siguiente tabla:

Nombre	Tipo	Dirección
PV A	Entrada Analógica	I:3.0
PV B	Entrada Analógica	I:3.1
SALIDA ANALOGICA	Salida Analógica	O:3/0
START	Bit	B3:0/0
CONTROL PID	Entero	N7:0 - N7:22
ENTRADA ESCALADA	Entero	N7:23
SALIDA PID	Entero	N7:24

Tabla. 5.21. Direcciones utilizadas Práctica #13

5. Abrir el software RSVIEW32, configurar el canal de comunicación, configurar el nodo de comunicación.

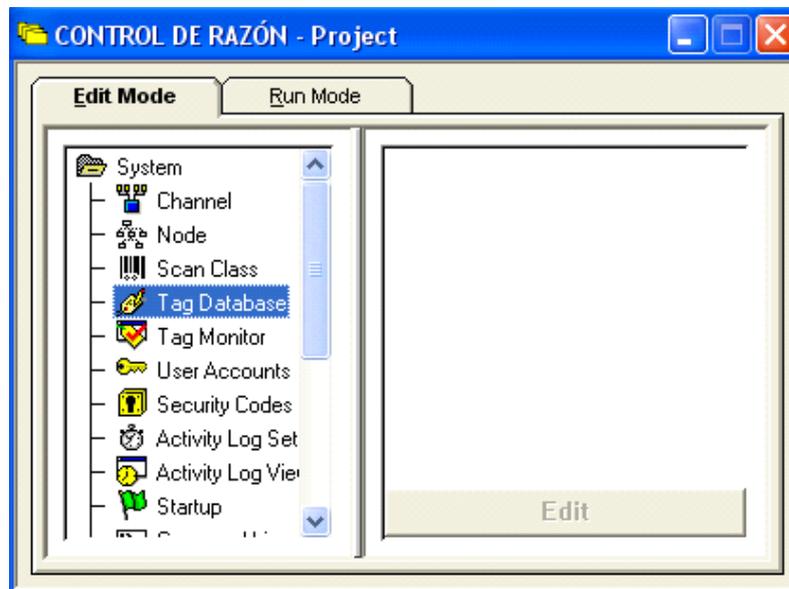


Figura. 5.104. Desarrollo Práctica #13 (4)

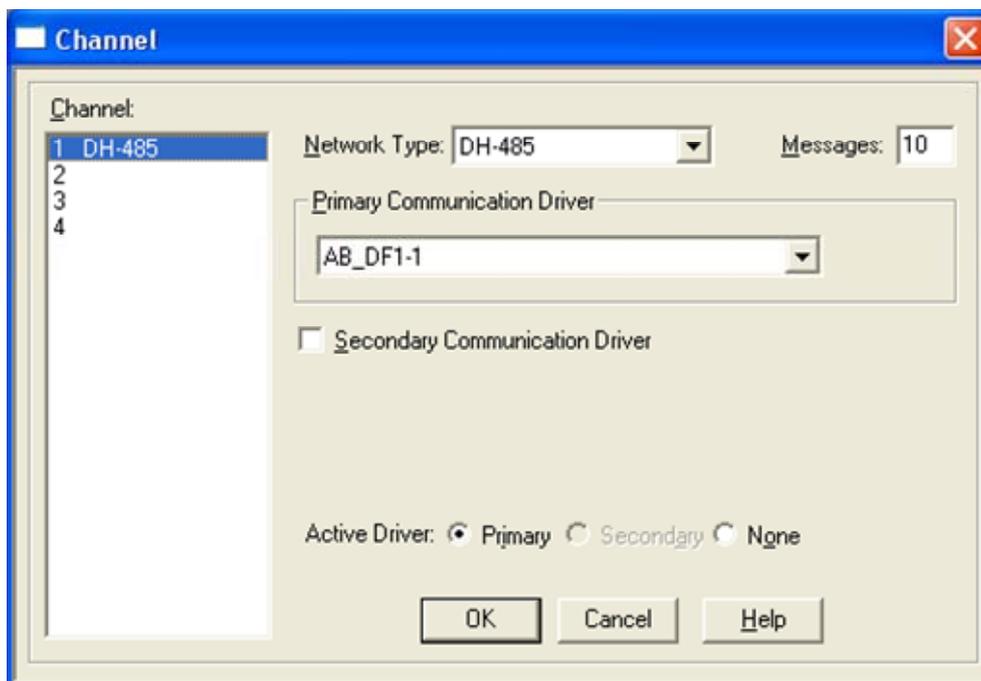


Figura. 5.105. Desarrollo Práctica #13 (5)

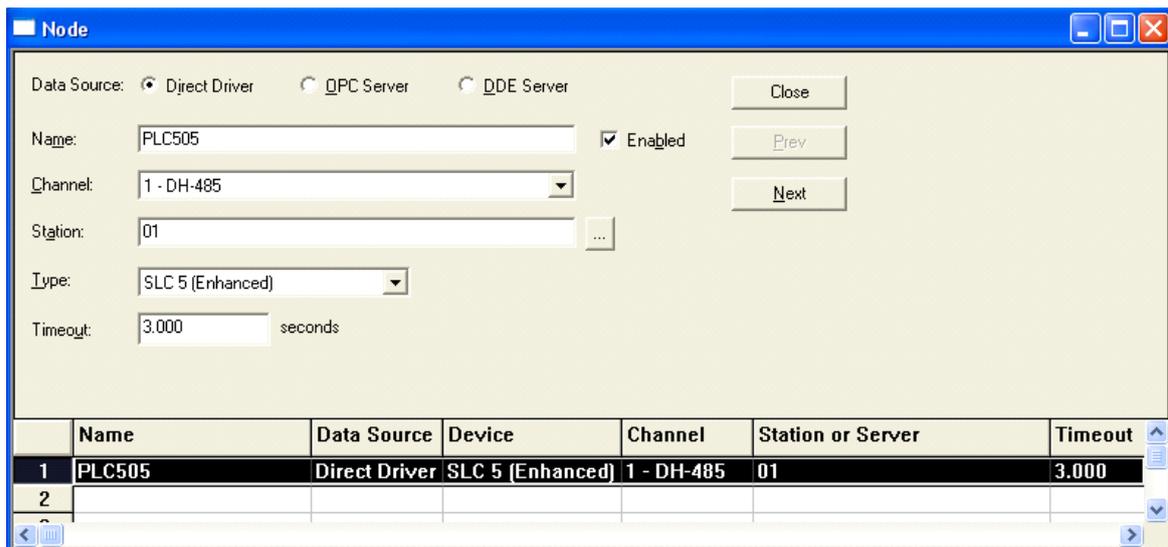


Figura. 5.106. Desarrollo Práctica #13 (6)

## 6. Creación de la HMI que interactúe con el PLC

### INTERFASE HUMANO MÁQUINA (HMI)

La HMI creada en base al esquema planteado consta de dos pantallas:

#### Pantalla Principal

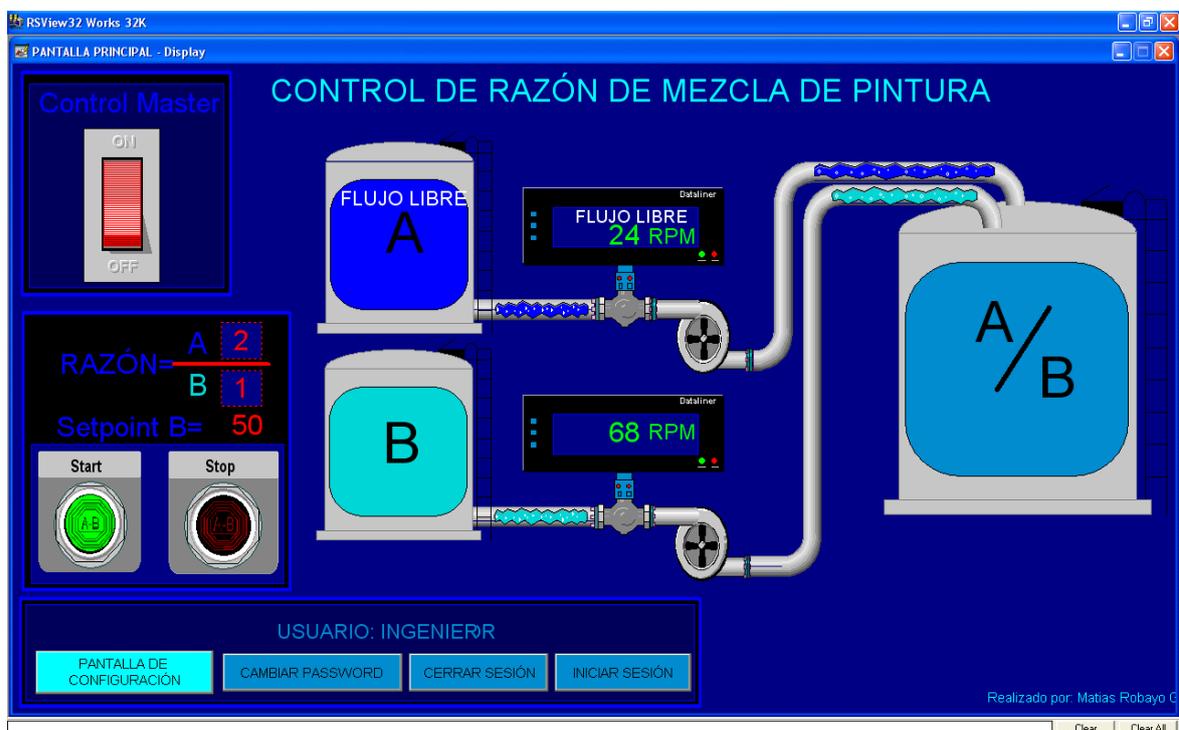


Figura. 5.107. Desarrollo Práctica #13 (7)

**Pantalla de Control**

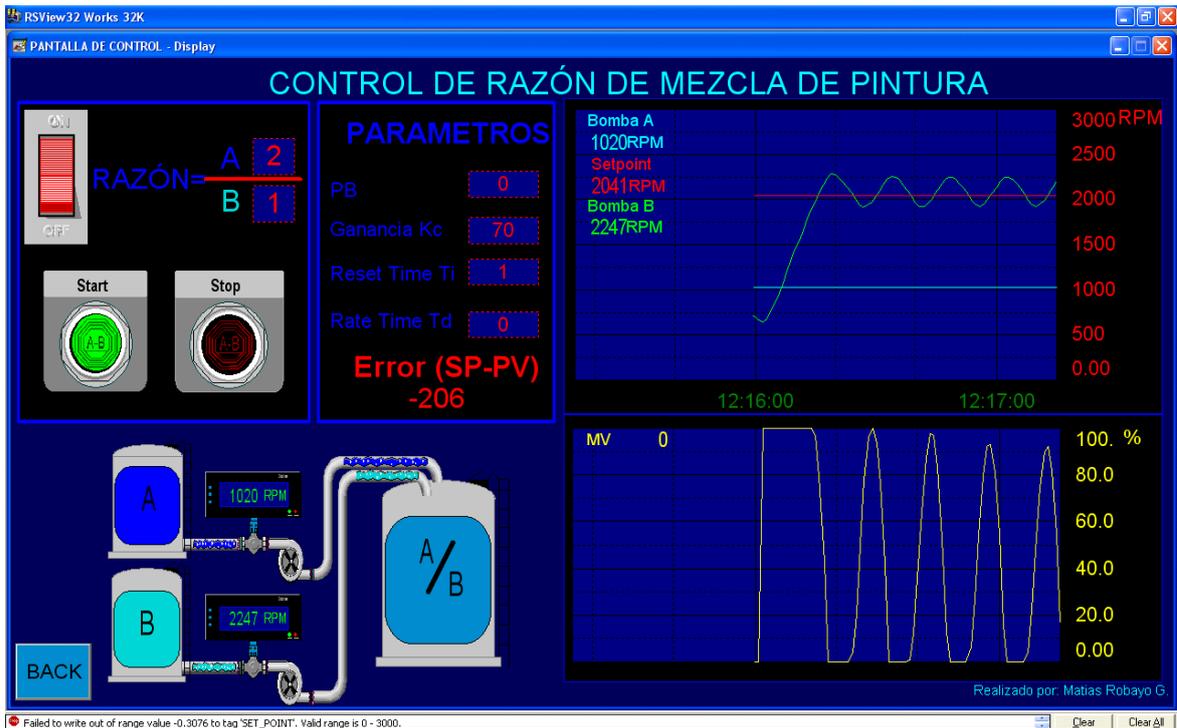


Figura. 5.108. Desarrollo Práctica #13 (8)

**TAGS**

En la siguiente tabla se muestran los tags empleados en la programación de la HMI y sus características:

NOMBRE	TIPO	CARACTERÍSTICAS					
ENCENDER	Digital	Alarmed:	Security: *	Memory Tag	Initial Value: Off		
		No					
Ganancia_Kc	Analog	Alarmed:	Security: *	Device Tag	Node Name: PLC505	Address: N7:3	Scan Class: A
		No					
MV	Analog	Alarmed:	Security: *	Device Tag	Node Name: PLC505	Address: N7:16	Scan Class: A
		No					
PB	Analog	Alarmed:	Security: *	Device Tag	Node Name: PLC505	Address: N7:9	Scan Class: A
		No					
		Minimum :	Maximum :	Scale : 1	Offset : 0	Units:	Data Type: (Default)

		0	70				(Default)
PV	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:23	Scan Class: A
		Minimum : 0	Maximum : 3000	Scale : 0.183000	Offset : 0	Units:	Data Type: (Default)
PV_A	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: I:3.0	Scan Class: A
		Minimum : 0	Maximum : 3000	Scale : 0.153800	Offset : 0	Units:	Data Type: (Default)
Rate_Time	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:5	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
RAZON_DENOMINADOR	Analog	Alarmed: No	Security: *	Memory Tag	Initial Value: 1		
		Minimum : 0	Maximum : 9	Scale : 1	Offset : 0	Units:	Data Type: (Default)
RAZON_NUMERADOR	Analog	Alarmed: No	Security: *	Memory Tag	Initial Value: 1		
		Minimum : 0	Maximum : 9	Scale : 1	Offset : 0	Units:	Data Type: (Default)
RESET_TIME	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:4	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
SALIDA_ANALOGICA	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: O:3.0	Scan Class: A
		Minimum : 0	Maximum : 12000	Scale : 1	Offset : 0	Units:	Data Type: Integer
SET_POINT	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:2	Scan Class: A
		Minimum : 0	Maximum : 3000	Scale : 1	Offset : 0	Units:	Data Type: (Default)
START	Digital	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: B3:0/0	Scan Class: A
		Off Label: Off	On Label: On				

**Tabla. 5.22. Tags utilizados Práctica #13**

Estos tags se encuentran configurados en la ventana *Tag Database* de acuerdo a las características que se encuentran en la tabla anterior, a continuación se muestra un ejemplo de esta configuración:

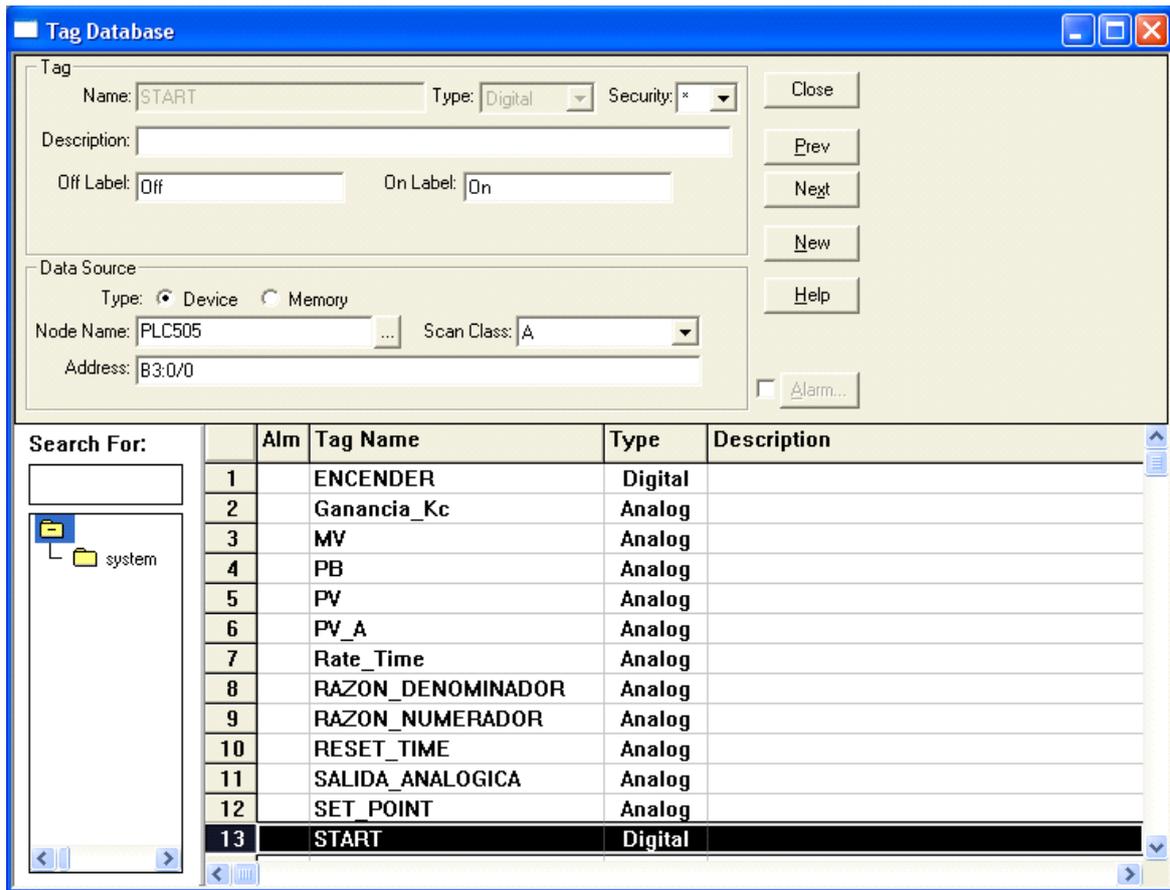


Figura. 5.109. Desarrollo Práctica #13 (9)

Como se puede ver en la figura 5.109., el tag START es de tipo digital y se encuentra asociado a la dirección B3:0/0 a través del nodo PLC505.

**USUARIOS**

Como se muestra en la siguiente figura 5.110., esta HMI posee dos cuentas de usuario, “Default” e “Ingeniero”.

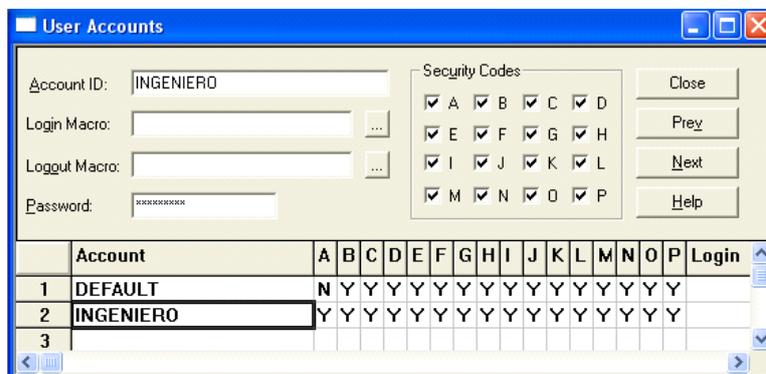


Figura. 5.110. Desarrollo Práctica #13 (10)

**Default:** Es el usuario que se muestra como “Operador” cuando se corre el programa, como se puede ver no posee acceso al código de seguridad A. Para restringir el acceso de este usuario a la Pantalla de Configuración, a dicha pantalla se le ha asignado el código de seguridad A, como se muestra en la figura 5.111.

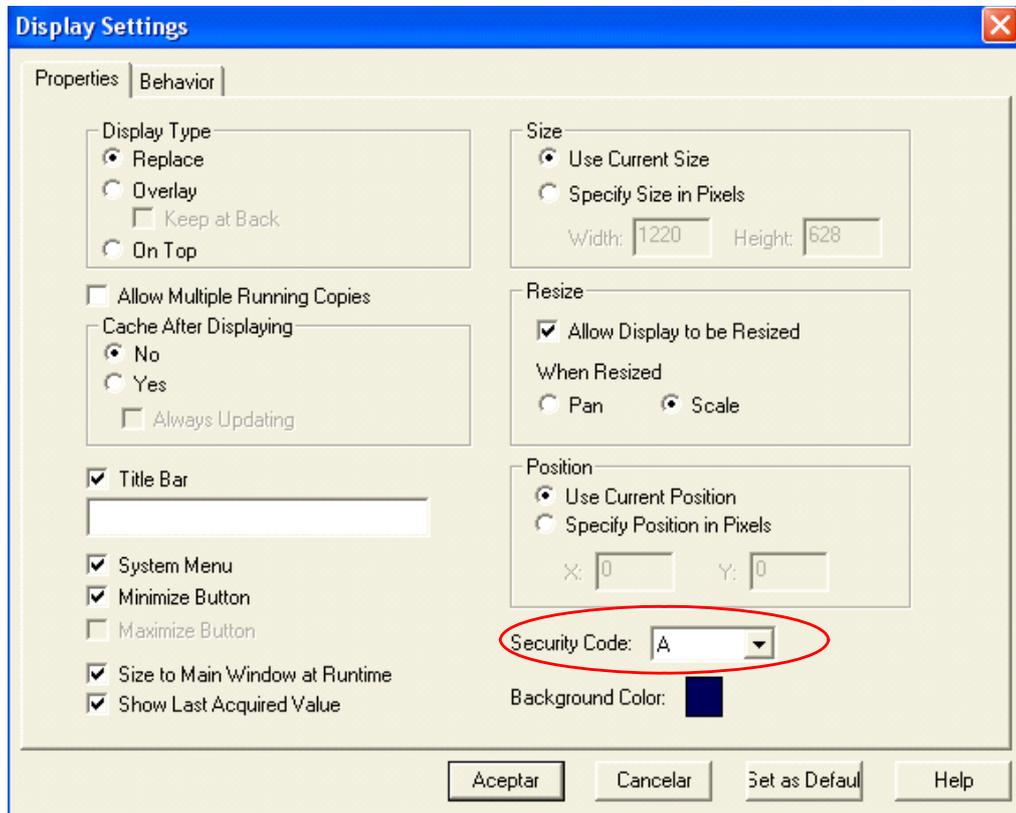


Figura. 5.111. Desarrollo Práctica #13 (11)

**Ingeniero:** Este usuario tiene acceso a todos los códigos de seguridad. Inicialmente tiene asignado el Password: *ingeniero*, el mismo que puede ser cambiado en la HMI (figura 5.112).

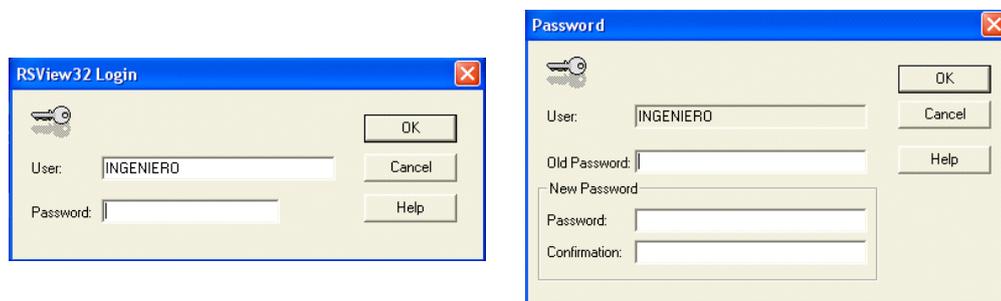


Figura. 5.112. Desarrollo Práctica #13 (12)

## 7. Sintonización de Controlador

Se utilizaron los mismos valores de sintonización que se obtuvieron en la sintonización de la práctica anterior debido a que la planta utilizada es la misma.

Es decir, se emplearon los siguientes valores:

$$\boxed{Kc = 7} \quad \boxed{Ti = 0.1} \quad \boxed{Td = 0}$$

8. Finalmente, se realizaron las pruebas necesarias para verificar que el controlador funciona correctamente de acuerdo al problema planteado.

### 5.13.5. PRUEBAS Y RESULTADOS

Durante el desarrollo de la práctica se realizaron las pruebas respectivas tanto del software (programación del PLC y de la HMI) como del hardware (Conexión del PLC y de los 2 Módulos Degem de simulación PCT-1 “DC SPEED CONTROL SYSTEM”), a través de las cuales se verificó que el PLC y la HMI trabajan correctamente de acuerdo a la lógica del problema planteado en esta práctica.

### 5.13.6. CONCLUSIONES Y RECOMENDACIONES

#### Conclusiones

- El control de razón es muy útil en aplicaciones en las que se requiere mantener una proporción o razón entre dos variables (temperatura, velocidad, etc.) puesto que permite mantener esta razón incluso cuando únicamente se puede controlar una de las dos variables.

#### Recomendaciones

- Se recomienda validar los valores de razón que se ingresan en la HMI, para evitar que el setpoint obtenido se salga de rango de trabajo permitido.

## 5.14. CONTROL DE PROCESOS 5: CONTROL POR SOBREPOSICIÓN

### 5.14.1. OBJETIVOS

- Familiarizarse con la técnica de Control por Sobreposición.
- Crear una HMI (Interfase Humano Máquina) que interactúe con el PLC.

### 5.14.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 (5/03 o superior)
- ✓ 1 Módulo de Entradas Discretas.
- ✓ 1 Módulo de Salidas Discretas.
- ✓ 1 Módulo de Entradas Analógicas.
- ✓ 1 Módulo de Salidas Analógicas.
- ✓ 1 Computador Personal, con instalación de RSLogix500, RSLinx y RSView32.
- ✓ 1 Módulo Degem de simulación PCT-2 “AIR FLOW TEMPERATURE CONTROL SYSTEM”.
- ✓ 1 Módulo Degem de simulación PCT-1 “DC SPEED CONTROL SYSTEM”
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

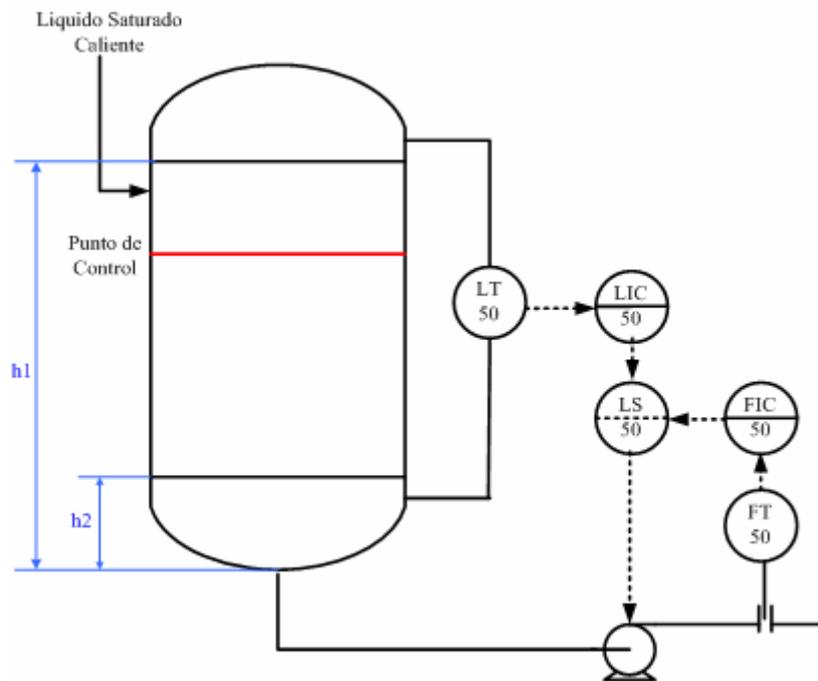
### 5.14.3. PLANTEAMIENTO

Se tiene el proceso indicado en la figura 5.113. (Proceso analizado en el marco teórico), en el que en operación normal el nivel del tanque debe estar a una altura  $h_1 = 4$  m, si por cualquier circunstancia el nivel del líquido baja a la altura  $h_2 = 1$  m, no se tendrá suficiente volumen positivo neto de succión (VPNS), de lo que resulta cavitación en la bomba.

Diseñar e implementar un sistema de control por Sobreposición que permita controlar la velocidad (RPM) de la bomba, evitando que se produzca cavitación en la misma. Para lo que se requerirá dos controladores uno de Velocidad de la bomba (Módulo de Velocidad) y otro de nivel del líquido (Módulo de Temperatura) como se indica en la figura 5.113.

Para el controlador de Velocidad se tiene un sensor de Velocidad que envía una señal de 0 a 6 VDC en relación lineal a la velocidad de la bomba (0 a 3000 RPM)

Para el controlador de Nivel se tiene un sensor de Nivel que envía una señal de 0 a 5 VDC en relación lineal al nivel del líquido (0 a 5 Metros)



**Figura. 5.113. Planteamiento Práctica #14 (1)**

Adicionalmente con el software RSView32, crear una HMI que permita interactuar al operador con el sistema controlador, la cual deberá tener las siguientes características:

Dos cuentas de usuario: INGENIERO (con contraseña) y OPERADOR

Pantalla Principal con el siguiente diseño referencial:

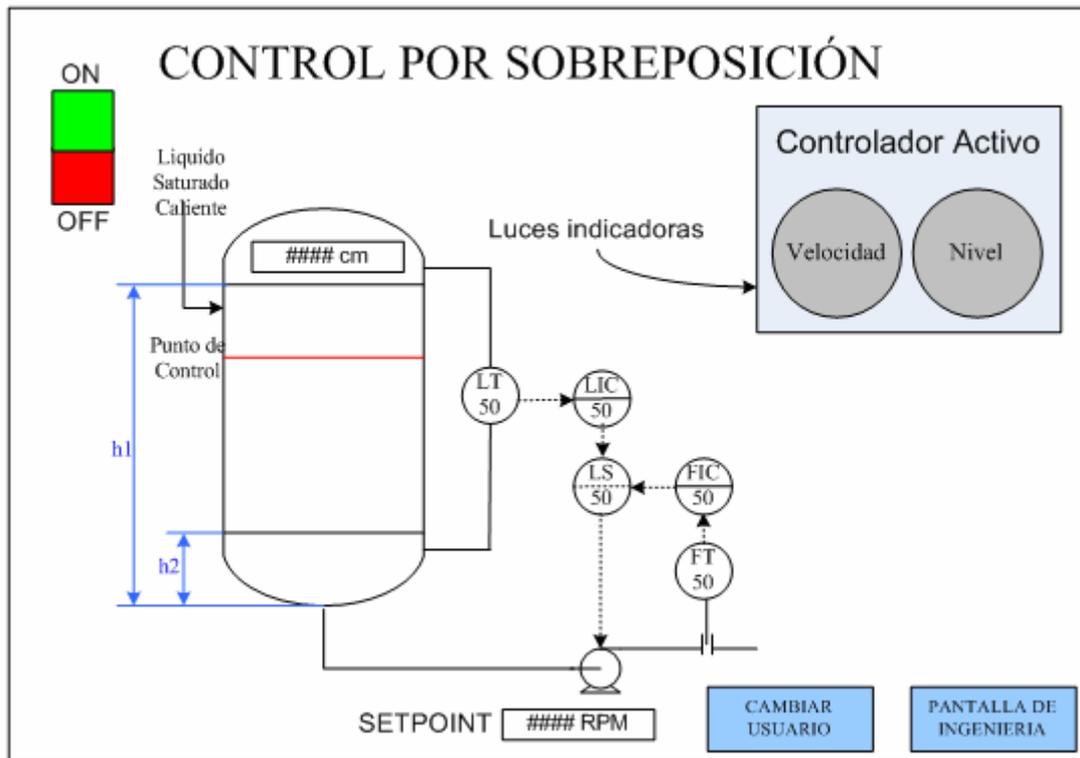


Figura. 5.114. Planteamiento Práctica #14 (2)

- ✓ El botón de “Pantalla de Ingeniería” despliega dicha pantalla únicamente si se encuentra trabajando con el usuario “INGENIERO”

Pantalla de Ingeniería (Acceso permitido únicamente para el usuario INGENIERO)

- ✓ Control Master de activación del controlador.
- ✓ Selección de la Velocidad deseada.
- ✓ Ingreso de los niveles de fluido  $h1$  y  $h2$
- ✓ Ingreso de parámetros de calibración del sistema controlador.
- ✓ Gráfica de tendencia (Trend) de Velocidad de la Bomba real (PV) y deseada (Setpoint).
- ✓ Gráfica de tendencia (Trend) de Nivel del líquido real (PV) y deseado (Setpoint).
- ✓ Gráfica de tendencia (Trend) de Voltaje de Salida de los controladores.

#### 5.14.4. DESARROLLO

1. Configuración de driver en RSLinx

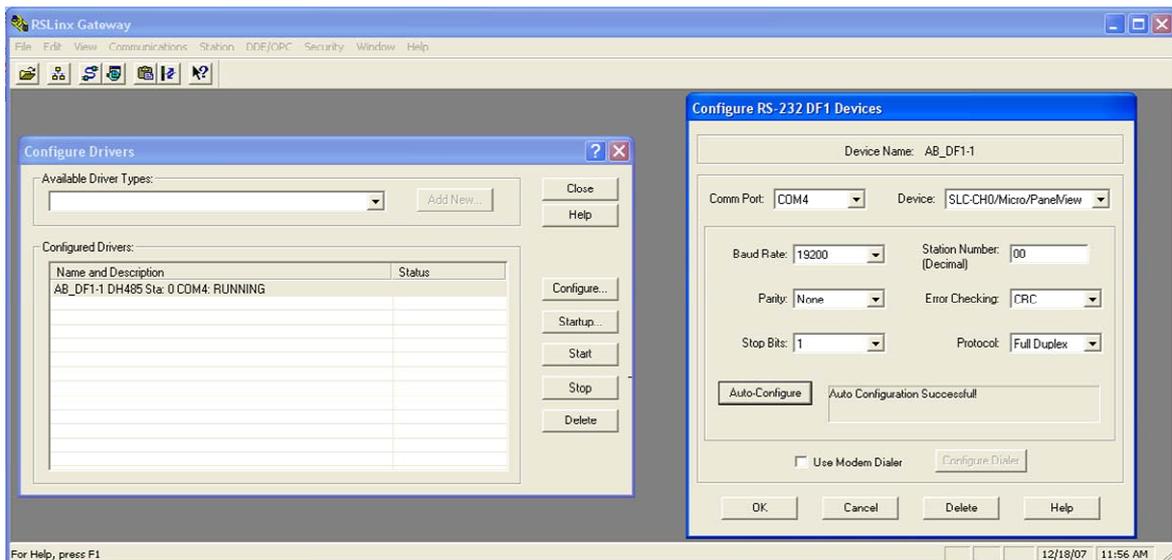


Figura. 5.115. Desarrollo Práctica #14 (1)

2. Creación del archivo y configuración del PLC en el software RSLogix500.

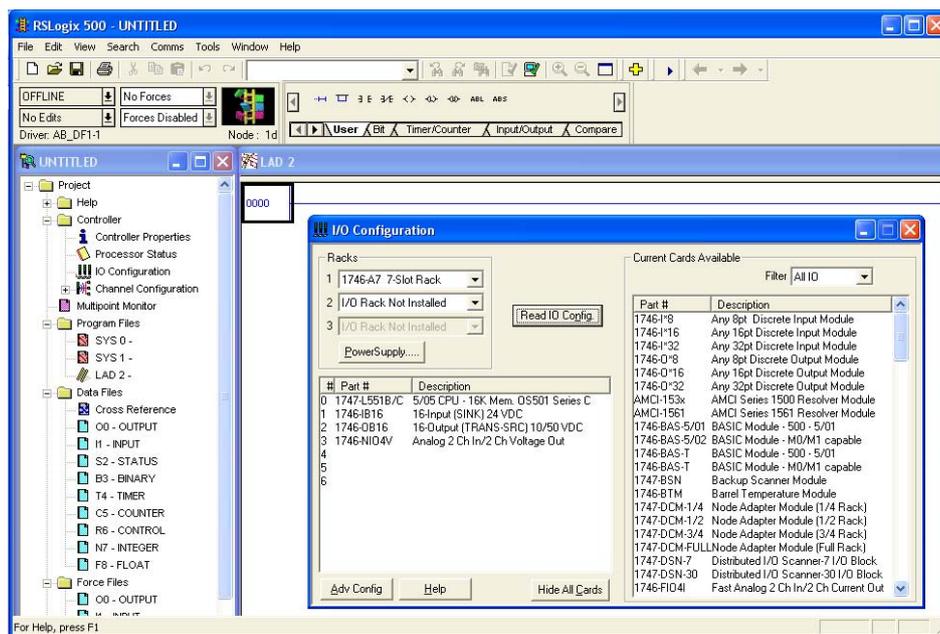
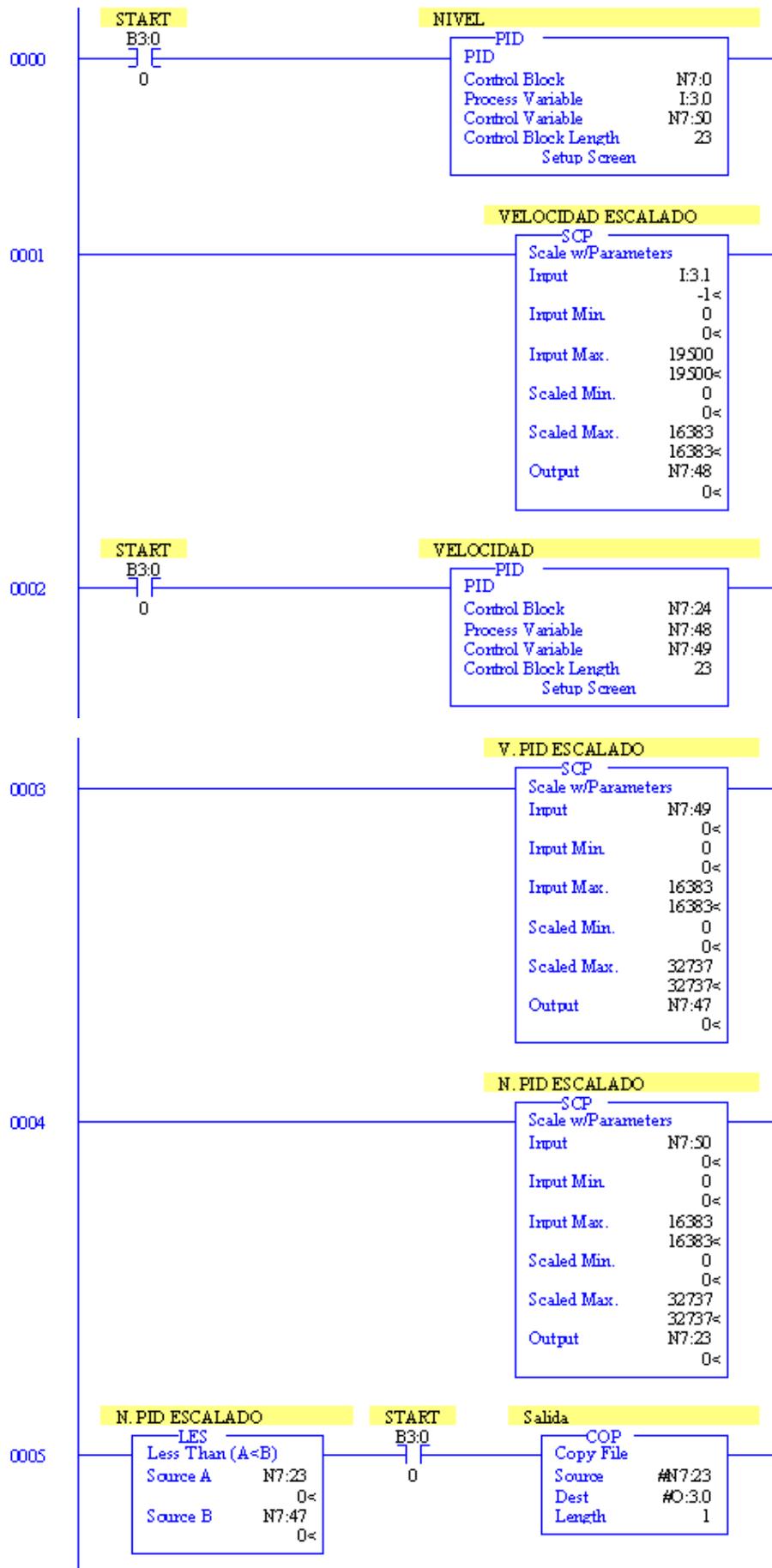


Figura. 5.116. Desarrollo Práctica #14 (2)

3. La programación Ladder que cumple con la lógica de control planteada en esta práctica es la siguiente:



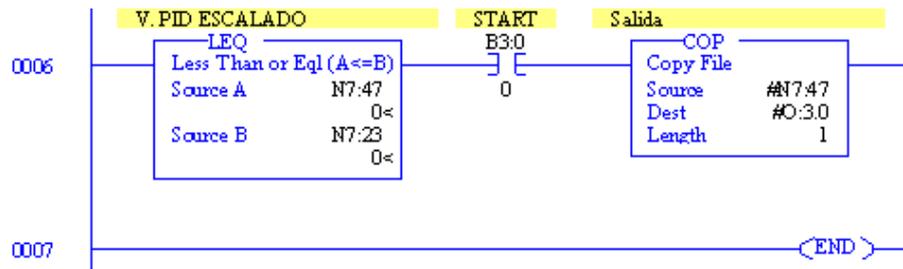


Figura. 5.117. Desarrollo Práctica #14 (3)

4. La asignación de direcciones en el PLC de los diferentes registros, entradas y salidas utilizadas, se la hizo de acuerdo a la siguiente tabla:

Nombre	Tipo	Dirección
PV NIVEL	Entrada Analógica	I:3.0
PV VELOCIDAD	Entrada Analógica	I:3.1
SALIDA	Salida Analógica	O:3/0
START	Bit	B3:0/0
CONTROL PID NIVEL	Entero	N7:0 - N7:22
CONTROL PID VELOCIDAD	Entero	N7:24 - N7:46
VELOCIDAD ESCALADO	Entero	N7:48
N. PID ESCALADO	Entero	N7:23
V. PID ESCALADO	Entero	N7:47

Tabla. 5.23. Direcciones utilizadas Práctica #14

5. Abrir el software RSView32, configurar el canal de comunicación, configurar el nodo de comunicación.

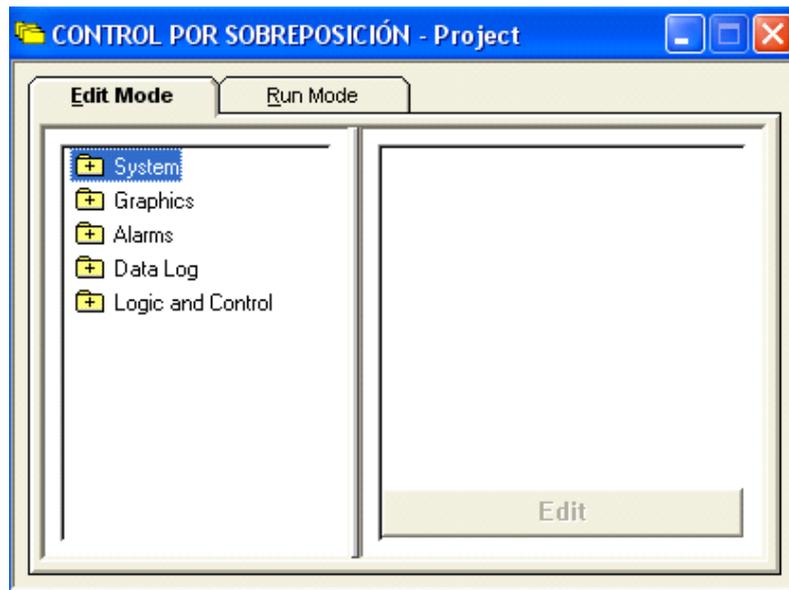


Figura. 5.118. Desarrollo Práctica #14 (4)

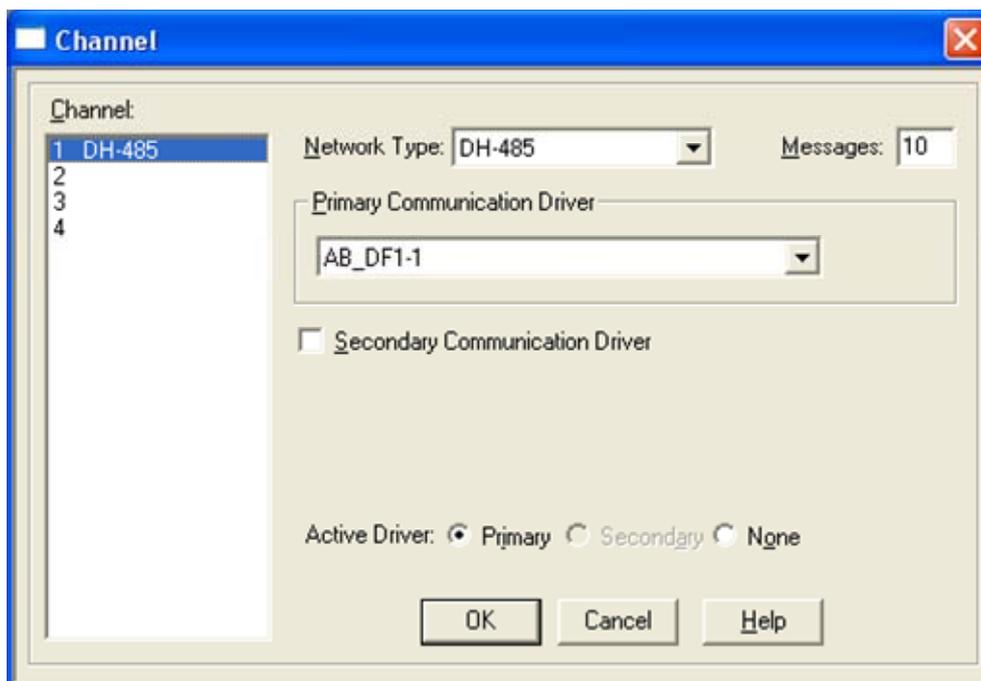


Figura. 5.119. Desarrollo Práctica #14 (5)

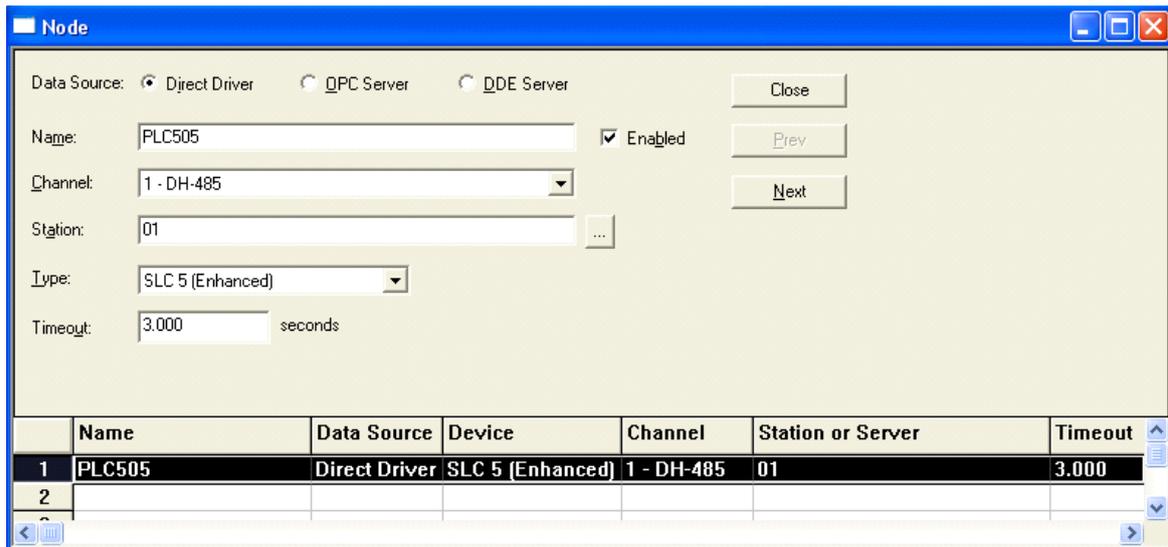


Figura. 5.120. Desarrollo Práctica #14 (6)

## 6. Creación de la HMI que interactúe con el PLC

### INTERFASE HUMANO MÁQUINA (HMI)

La HMI creada en base al esquema planteado consta de dos pantallas:

#### Pantalla Principal



Figura. 5.121. Desarrollo Práctica #14 (7)

**Pantalla de Control**



Figura. 5.122. Desarrollo Práctica #14 (8)

**TAGS**

En la siguiente tabla se muestran los tags empleados en la programación de la HMI y sus características:

NOMBRE	TIPO	CARACTERÍSTICAS					
ENCENDER	Digital	Alarmed:	Security: *	Memory	Initial		
		No		Tag	Value:		
GANANCIA_NIVEL	Analog	Off Label:	On Label:	Device	Node	Address:	Scan
		Off	On	Tag	Name:	N7:3	Class: A
GANANCIA_VELOCIDAD	Analog	Alarmed:	Security: *	Scale : 1	Offset : 0	Units:	Data
		No		Maximum : 32767	PLC505		Type: (Default)
MV_NIVEL	Analog	Alarmed:	Security: *	Device	Node	Address:	Scan
		No		Tag	Name:		
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data
				PLC505			Type: (Default)

		No		Tag	Name: PLC505	N7:16	Class: A
		Minimum : 0	Maximum : 100	Scale : 1	Offset : 0	Units:	Data Type: (Default)
MV_VELOCIDAD	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:40	Scan Class: A
		Minimum : 0	Maximum : 100	Scale : 1	Offset : 0	Units:	Data Type: (Default)
PV_NIVEL	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: I:3.0	Scan Class: A
		Minimum : 0	Maximum : 5	Scale : 0.000305	Offset : 0	Units:	Data Type: (Default)
PV_VELOCIDAD	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: I:3.1	Scan Class: A
		Minimum : 0	Maximum : 3000	Scale : 0.153800	Offset : 0	Units:	Data Type: (Default)
RATE_TIME_NIVEL	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:5	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
RATE_TIME_VELOCIDAD	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:29	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
RESET_TIME_NIVEL	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:4	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
RESET_TIME_VELOCIDAD	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:28	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
SALIDA_ANALOGICA	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: O:3.0	Scan Class: A
		Minimum : 0	Maximum : 32000	Scale : 1	Offset : 0	Units:	Data Type: Integer
SETPOINT_NIVEL	Analog	Alarmed:	Security: *	Device	Node	Address:	Scan

		No		Tag	Name: PLC505	N7:2	Class: A
		Minimum : 0	Maximum : 5	Scale : 0.000305	Offset : 0	Units:	Data Type: (Default)
SETPOINT_VELOCIDAD	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:26	Scan Class: A
		Minimum : 0	Maximum : 3000	Scale : 1	Offset : 0	Units:	Data Type: (Default)
START	Digital	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: B3:0/0	Scan Class: A
		Off Label: Off	On Label: On				

Tabla. 5.24. Tags utilizados Práctica #14

Estos tags se encuentran configurados en la ventana *Tag Database* de acuerdo a las características que se encuentran en la tabla anterior, a continuación se muestra un ejemplo de esta configuración:

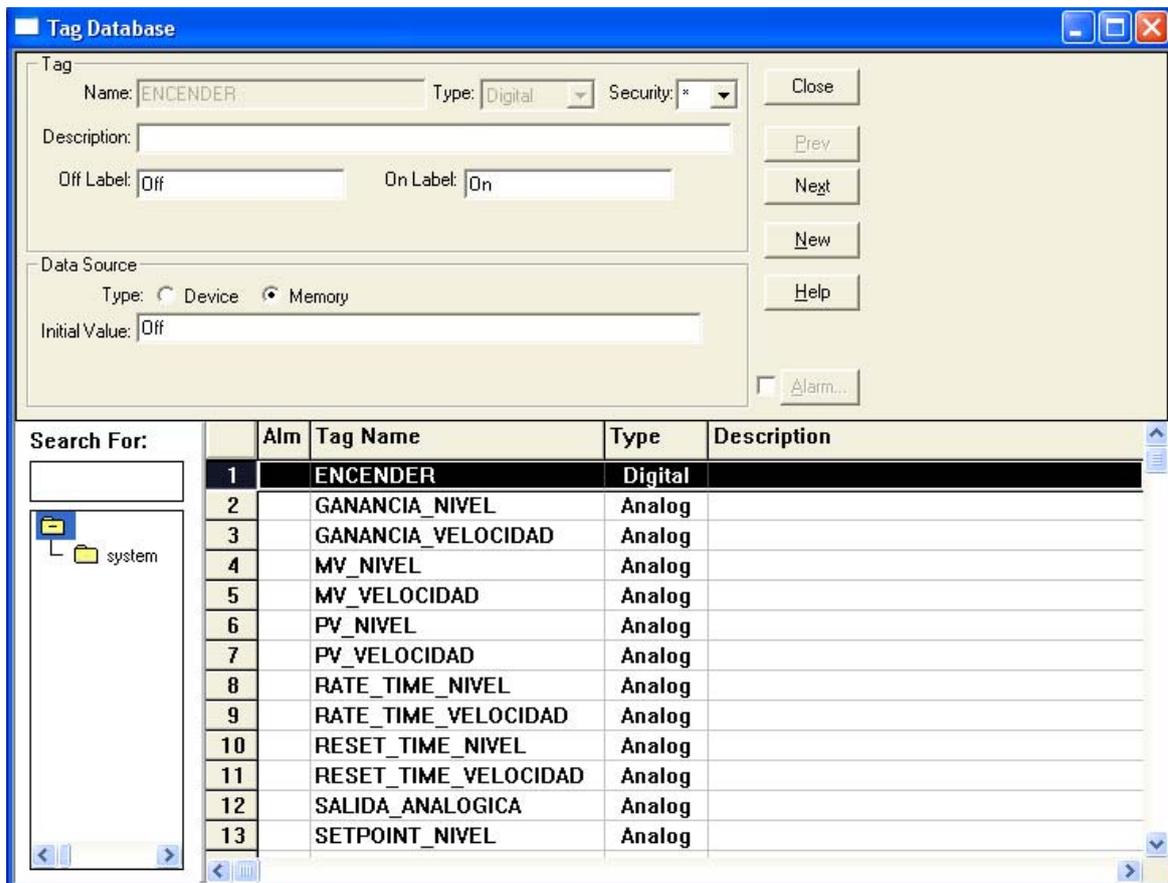


Figura. 5.123. Desarrollo Práctica #14 (9)

Como se puede ver en la figura 5.123., el tag ENCENDER es de tipo digital y no se encuentra asociado a ninguna dirección ya que es de tipo Memory.

## USUARIOS

Como se muestra en la siguiente figura 5.124., esta HMI posee dos cuentas de usuario, “Default” e “Ingeniero”.

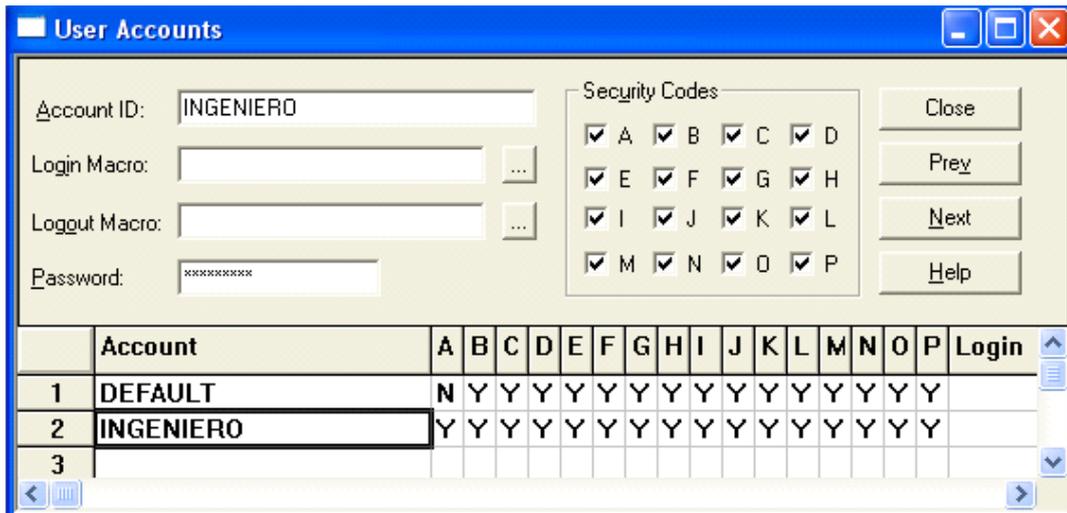
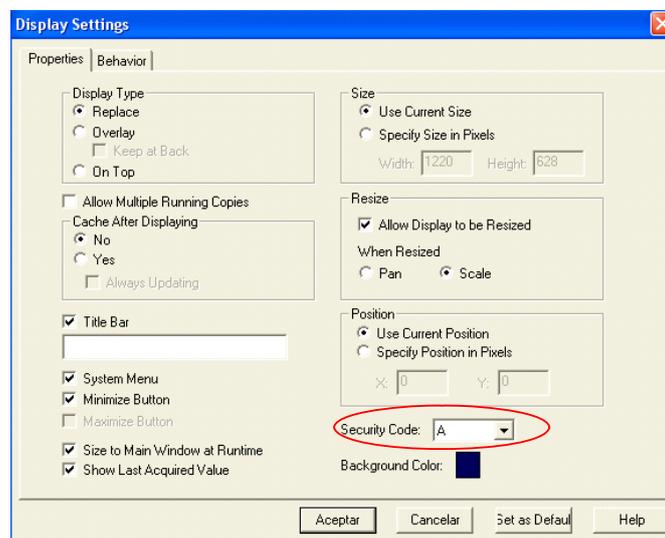


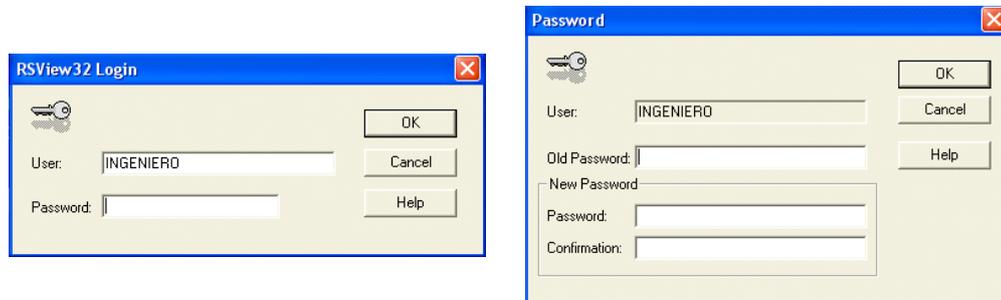
Figura. 5.124. Desarrollo Práctica #14 (10)

Default: Es el usuario que se muestra como “Operator” cuando se corre el programa, como se puede ver no posee acceso al código de seguridad A. Para restringir el acceso de este usuario a la Pantalla de Configuración, a dicha pantalla se le ha asignado el código de seguridad A, como se muestra en la figura 5.125.



**Figura. 5.125. Desarrollo Práctica #14 (11)**

Ingeniero: Este usuario tiene acceso a todos los códigos de seguridad. Inicialmente tiene asignado el Password: *ingeniero*, el mismo que puede ser cambiado en la HMI (figura 5.126).

**Figura. 5.126. Desarrollo Práctica #14 (12)**

## 7. Sintonización de Controlador

### Controlador de Velocidad:

Para el controlador de Velocidad, se utilizaron los mismos valores de sintonización que se obtuvieron en la sintonización de la práctica “PID DE VELOCIDAD” debido a que la planta utilizada es la misma.

Es decir, se emplearon los siguientes valores:

$$\boxed{Kc = 7} \quad \boxed{Ti = 0.1} \quad \boxed{Td = 0}$$

### Controlador de Nivel (Temperatura):

Para el controlador de Temperatura, se utilizaron los mismos valores de sintonización que se obtuvieron en la sintonización de la práctica “PID DE TEMPERATURA” debido a que la planta utilizada es la misma.

Es decir, se emplearon los siguientes valores:

$$\boxed{Kc = 3.5} \quad \boxed{Ti = 0.5} \quad \boxed{Td = 0.12}$$

8. Finalmente, se realizaron las pruebas necesarias para verificar que el controlador funciona correctamente de acuerdo al problema planteado.

### **5.14.5. PRUEBAS Y RESULTADOS**

Durante el desarrollo de la práctica se realizaron las pruebas respectivas tanto del software (programación del PLC y de la HMI) como del hardware (Conexión del PLC, del Módulo Degem de simulación “AIR FLOW TEMPERATURE CONTROL SYSTEM PCT-2” y del Módulo Degem de simulación PCT-1 “DC SPEED CONTROL SYSTEM”), a través de las cuales se verificó que el PLC y la HMI trabajan correctamente de acuerdo a la lógica del problema planteado en esta práctica.

### **5.14.6. CONCLUSIONES Y RECOMENDACIONES**

#### **Conclusiones**

- La técnica de control por sobreposición permite cambiar de controlador cuando las variables del proceso se encuentran fuera de ciertos límites, razón por la cual este tipo de control se utiliza como un control de protección.

#### **Recomendaciones**

- Al diseñar un sistema de control por sobreposición, se recomienda garantizar que exista protección contra reajuste excesivo en ambos controladores.

## 5.15. CONTROL DE PROCESOS 6: CONTROL EN CASCADA

### 5.15.1. OBJETIVOS

- Familiarizarse con la técnica de Control en Cascada.
- Crear una HMI (Interfase Humano Máquina) que interactúe con el PLC.

### 5.15.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 (5/03 o superior)
- ✓ 1 Módulo de Entradas Discretas.
- ✓ 1 Módulo de Salidas Discretas.
- ✓ 1 Módulo de Entradas Analógicas.
- ✓ 1 Módulo de Salidas Analógicas.
- ✓ 1 Computador Personal, con instalación de RSLogix500, RSLinx y RSView32.
- ✓ 1 Módulo Degem de simulación PCT-2 “AIR FLOW TEMPERATURE CONTROL SYSTEM”.
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

### 5.15.3. PLANTEAMIENTO

En una planta de pintura de autos, se desea controlar la temperatura de una cabina de secado de pintura que trabaje en un rango de temperatura de 40 a 50 °C. Esta cabina es calentada a través de un flujo de aire caliente que llega a través de una tubería desde un horno, puesto que no se puede controlar el flujo de aire, se debe controlar el sistema calentador del Horno. Además se poseen dos sensores de temperatura, uno ubicado en la tubería de aire (A) y el otro ubicado dentro de la cabina (B) como se muestra en la figura 5.127.

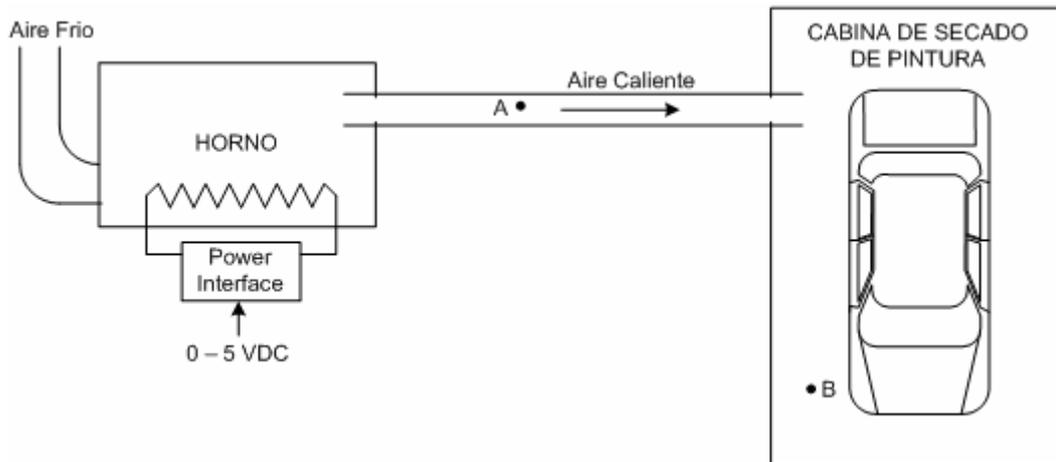


Figura. 5.127. Planteamiento Práctica #15 (1)

Diseñar e implementar un sistema de Control en Cascada que permita realizar dicho control. El error estacionario máximo permitido es de  $\pm 0.1^{\circ}\text{C}$ .

Adicionalmente con el software RSView32, crear una HMI que permita interactuar al operador con el sistema controlador, la cual deberá tener las siguientes características:

Dos cuentas de usuario: INGENIERO (con contraseña) y OPERADOR

Pantalla Principal con el siguiente diseño referencial:

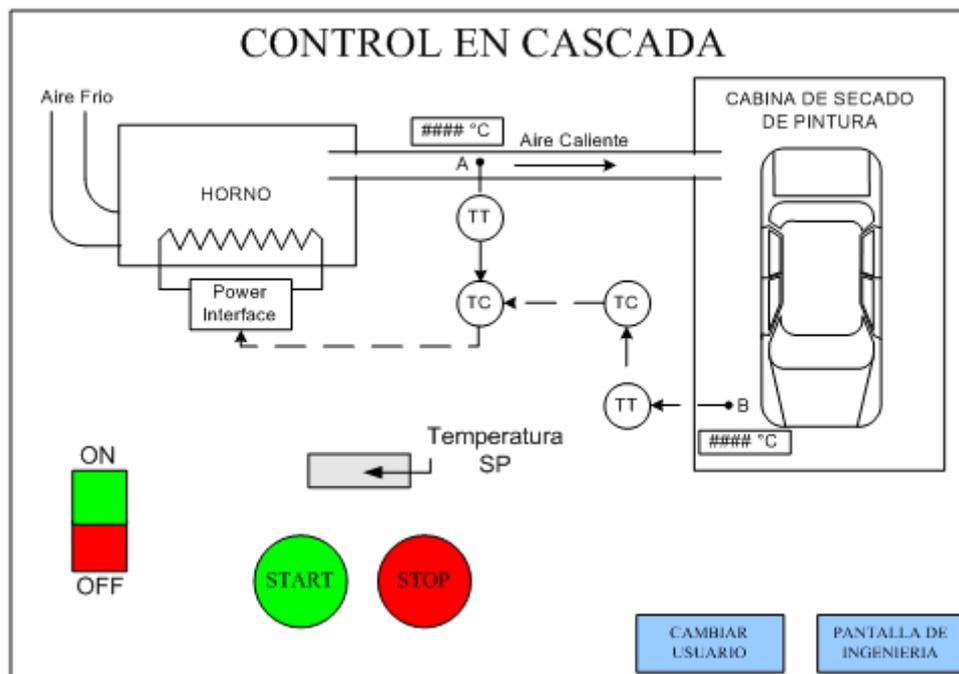


Figura. 5.128. Planteamiento Práctica #15 (2)

- ✓ El botón de “Pantalla de Ingeniería” despliega dicha pantalla únicamente si se encuentra trabajando con el usuario “INGENIERO”

Pantalla de Ingeniería (Acceso permitido únicamente para el usuario INGENIERO)

- ✓ Control de activación del controlador.
- ✓ Selección de la Temperatura de la Cabina (Setpoint).
- ✓ Ingreso de parámetros de calibración de ambos controladores (maestro y esclavo).
- ✓ Gráfica de tendencia (Trend) de la Temperatura real y la deseada del controlador maestro.
- ✓ Gráfica de tendencia (Trend) de la Temperatura real y la deseada del controlador esclavo.
- ✓ Gráfica de tendencia (Trend) de Voltaje de Salida de ambos controladores.

#### 5.15.4. DESARROLLO

##### 1. Configuración de driver en RSLinx

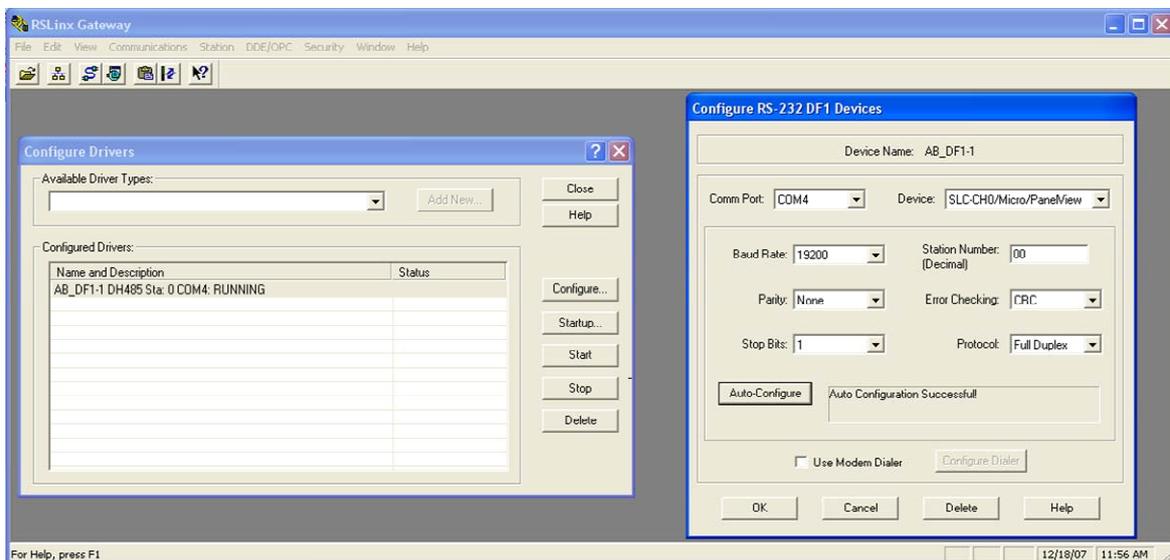


Figura. 5.129. Desarrollo Práctica #15 (1)

##### 2. Creación del archivo y configuración del PLC en el software RSLogix500.

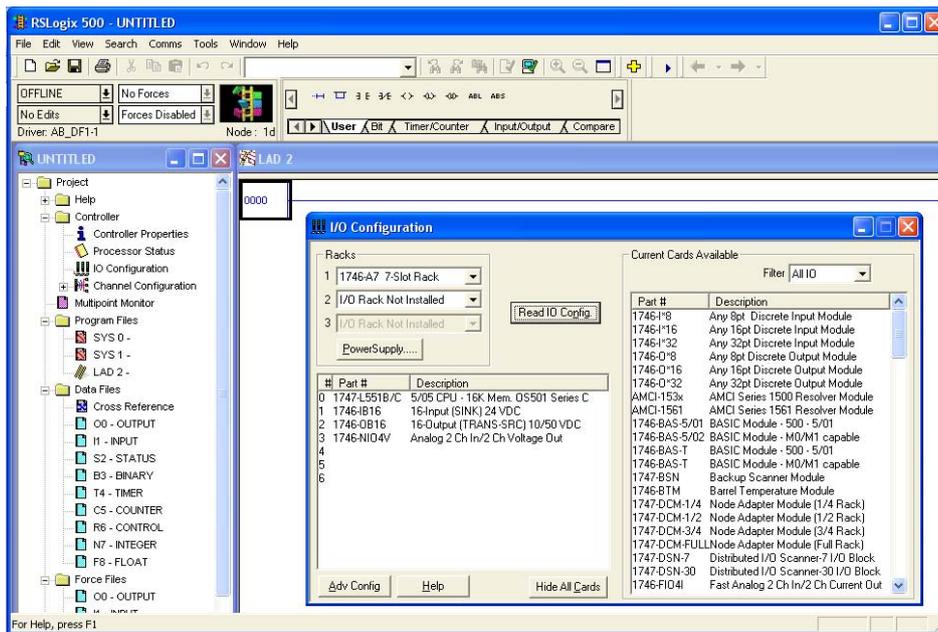


Figura. 5.130. Desarrollo Práctica #15 (2)

3. La programación Ladder que cumple con la lógica de control planteada en esta práctica es la siguiente:

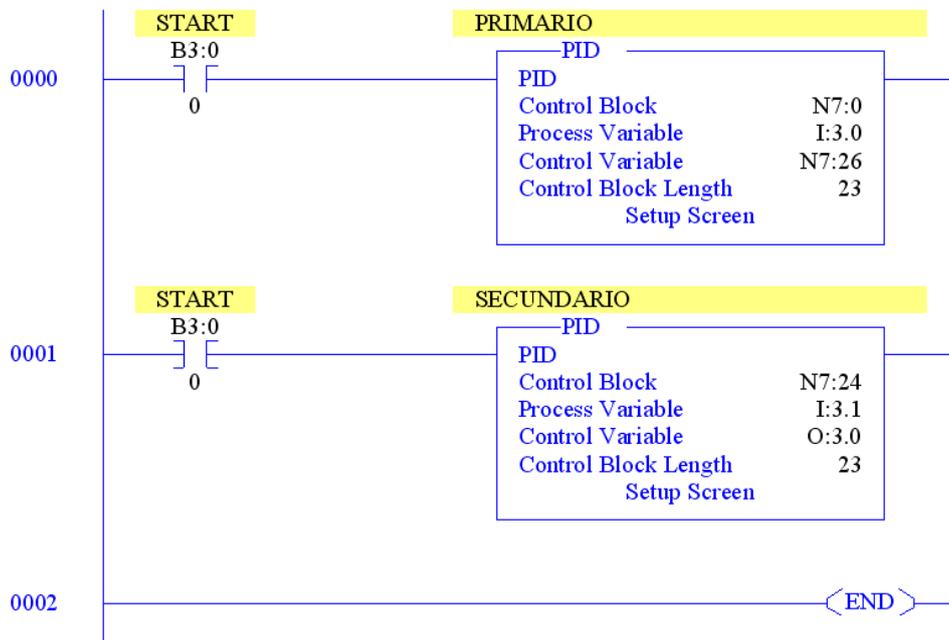


Figura. 5.131. Desarrollo Práctica #15 (3)

4. La asignación de direcciones en el PLC de los diferentes registros, entradas y salidas utilizadas, se la hizo de acuerdo a la siguiente tabla:

Nombre	Tipo	Dirección
PV PRIMARIO	Entrada Analógica	I:3.0
PV SECUNDARIO	Entrada Analógica	I:3.1
SALIDA	Salida Analógica	O:3/0
START	Bit	B3:0/0
CONTROL PRIMARIO	Entero	N7:0 - N7:22
CONTROL SECUNDARIO	Entero	N7:24 - N7:46

Tabla. 5.25. Direcciones utilizadas Práctica #15

5. Abrir el software RSView32, configurar el canal de comunicación, configurar el nodo de comunicación.

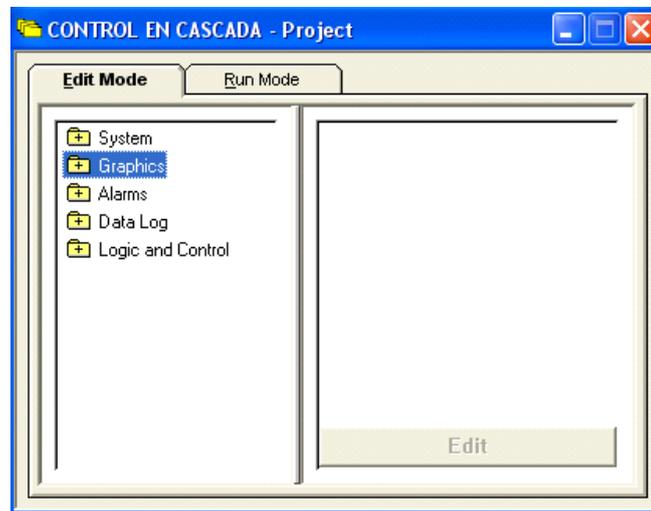


Figura. 5.132. Desarrollo Práctica #15 (4)

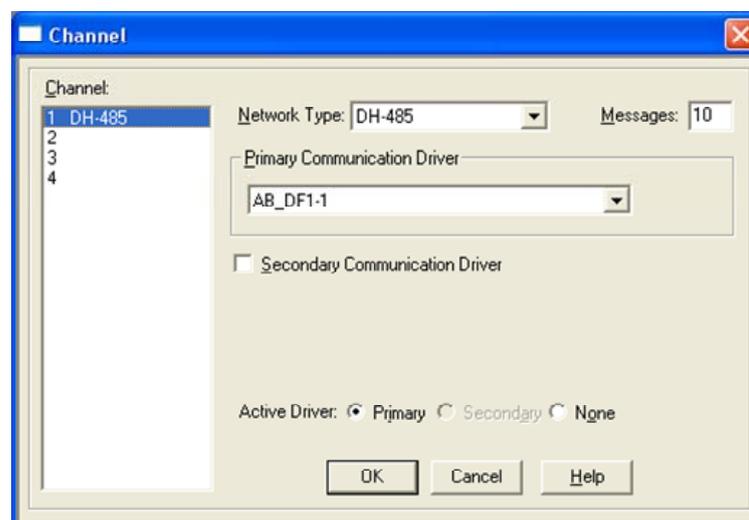


Figura. 5.133. Desarrollo Práctica #15 (5)

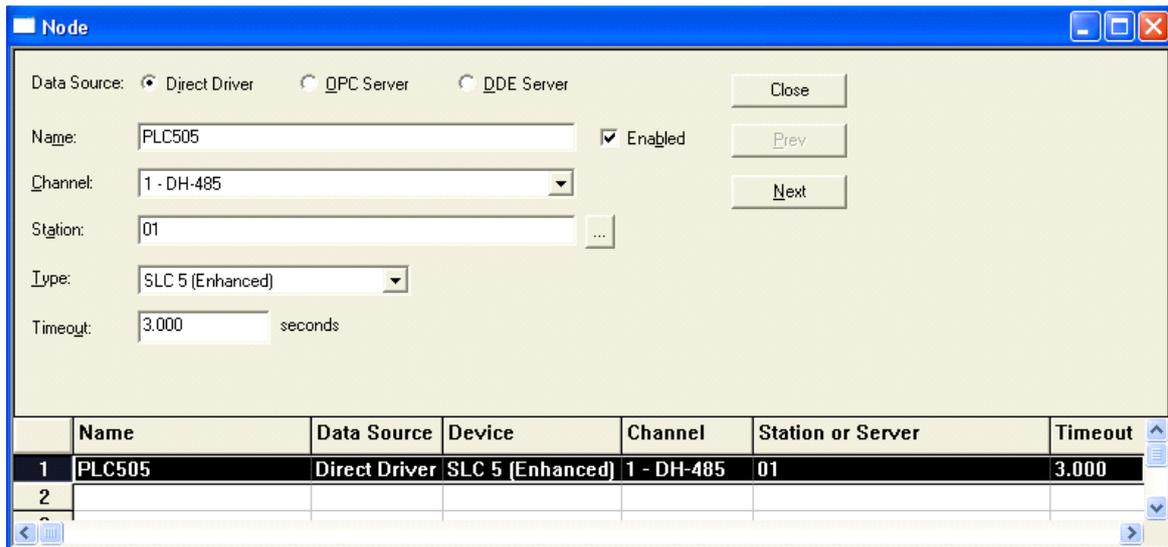


Figura. 5.134. Desarrollo Práctica #15 (6)

## 6. Creación de la HMI que interactúe con el PLC

### INTERFASE HUMANO MÁQUINA (HMI)

La HMI creada en base al esquema planteado consta de dos pantallas:

#### Pantalla Principal

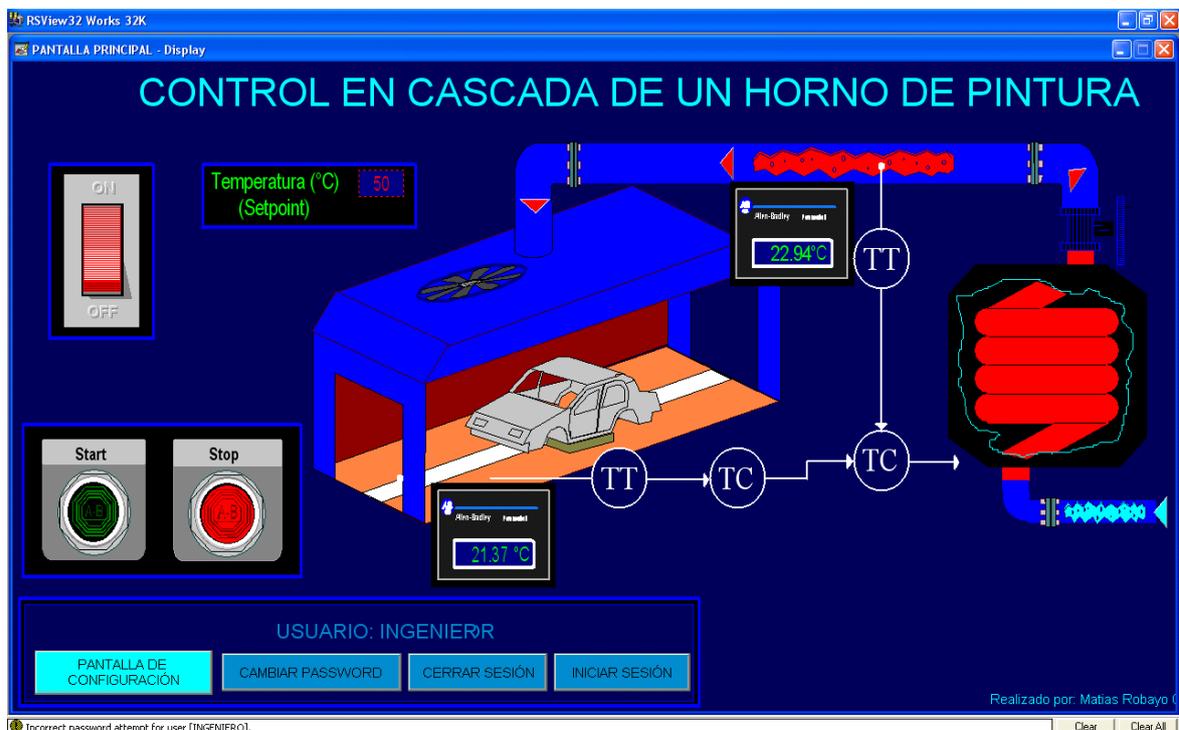


Figura. 5.135. Desarrollo Práctica #15 (7)

## Pantalla de Control



Figura. 5.136. Desarrollo Práctica #15 (8)

## TAGS

En la siguiente tabla se muestran los tags empleados en la programación de la HMI y sus características:

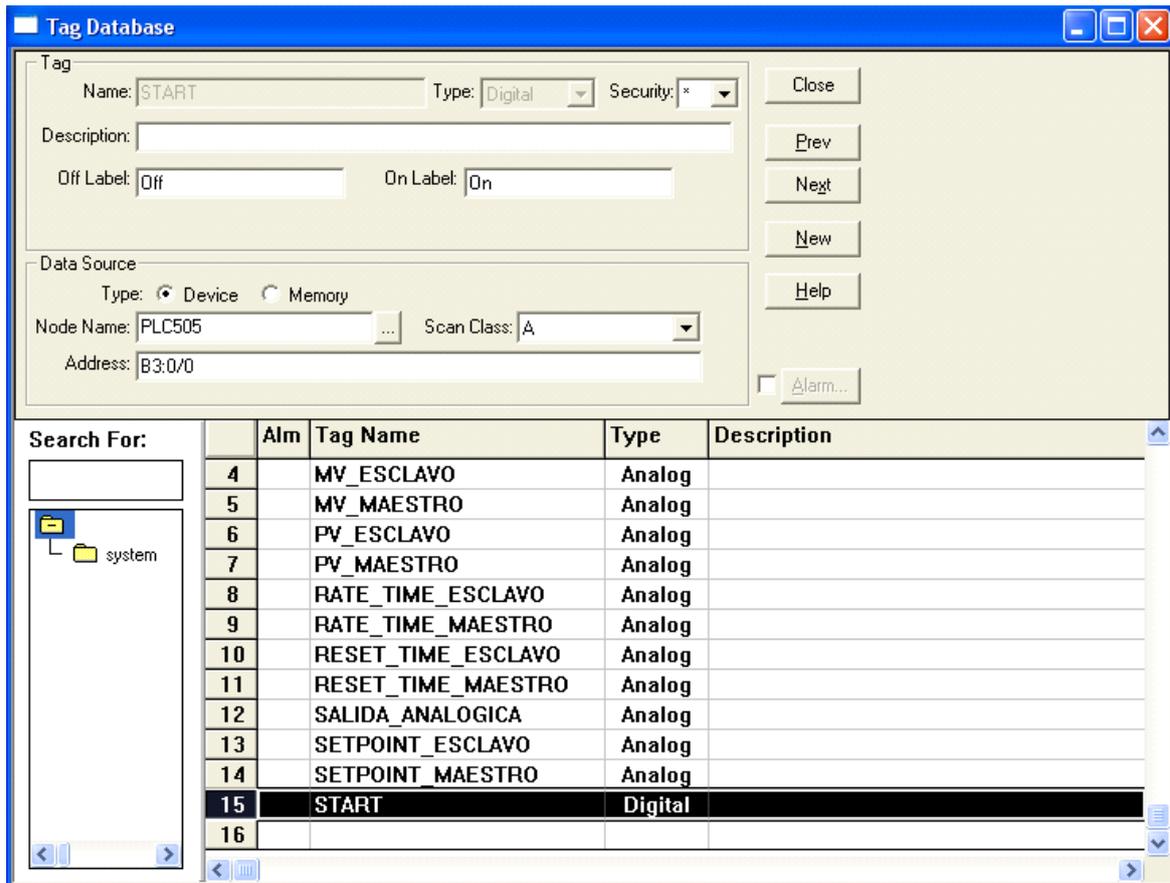
NOMBRE	TIPO	CARACTERÍSTICAS					
ENCENDER	Digital	Alarmed: No	Security: *	Memory Tag	Initial Value: Off		
		Off Label: Off	On Label: On				
GANANCIA_ESCLAVO	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:27	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
GANANCIA_MAESTRO	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:3	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
MV_ESCLAVO	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:40	Scan Class: A
		Minimum :	Maximum :	Scale : 1	Offset : 0	Units:	Data Type:

		0	100				(Default)
MV_MAESTRO	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:16	Scan Class: A
		Minimum : 0	Maximum : 100	Scale : 1	Offset : 0	Units:	Data Type: (Default)
PV_ESCLAVO	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: I:3.1	Scan Class: A
		Minimum : 20	Maximum : 70	Scale : 0.003050	Offset : 20	Units:	Data Type: (Default)
PV_MAESTRO	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: I:3.0	Scan Class: A
		Minimum : 20	Maximum : 70	Scale : 0.003050	Offset : 20	Units:	Data Type: (Default)
RATE_TIME_ESCLAVO	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:29	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
RATE_TIME_MAESTRO	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:5	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
RESET_TIME_ESCLAVO	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:28	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
RESET_TIME_MAESTRO	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:4	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
SALIDA_ANALOGICA	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: O:3.0	Scan Class: A
		Minimum : 0	Maximum : 163838	Scale : 1	Offset : 0	Units:	Data Type: (Default)
SETPOINT_ESCLAVO	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:26	Scan Class: A
		Minimum : 20	Maximum : 70	Scale : 0.003050	Offset : 20	Units:	Data Type: (Default)
SETPOINT_MAESTRO	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:2	Scan Class: A
		Minimum : 20	Maximum : 70	Scale : 1	Offset : 0	Units:	Data Type: (Default)
START	Digital	Alarmed: No	Security: *	Device Tag	Node Name:	Address: B3:0/0	Scan Class: A

					PLC505		
		Off Label: Off	On Label: On				

**Tabla. 5.26. Tags utilizados Práctica #15**

Estos tags se encuentran configurados en la ventana *Tag Database* de acuerdo a las características que se encuentran en la tabla anterior, a continuación se muestra un ejemplo de esta configuración:



**Figura. 5.137. Desarrollo Práctica #15 (9)**

Como se puede ver en la figura 5.137, el tag START es de tipo digital y se encuentra asociado a la dirección B3:0/0 a través del nodo de comunicación PLC505

## USUARIOS

Como se muestra en la siguiente figura 5.138, esta HMI posee dos cuentas de usuario, "Default" e "Ingeniero".

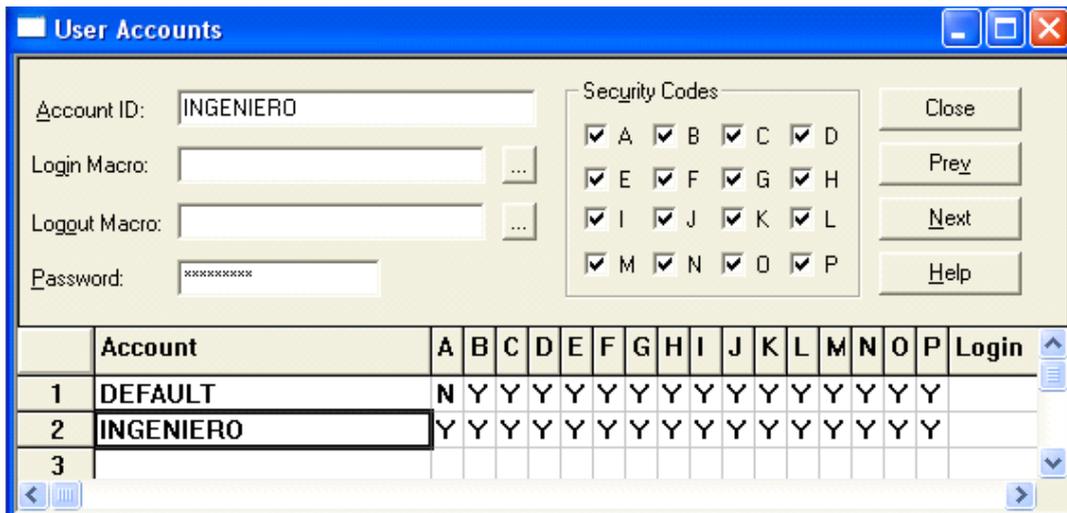


Figura. 5.138. Desarrollo Práctica #15 (10)

Default: Es el usuario que se muestra como “Operador” cuando se corre el programa, como se puede ver no posee acceso al código de seguridad A. Para restringir el acceso de este usuario a la Pantalla de Configuración, a dicha pantalla se le ha asignado el código de seguridad A, como se muestra en la figura 5.139.

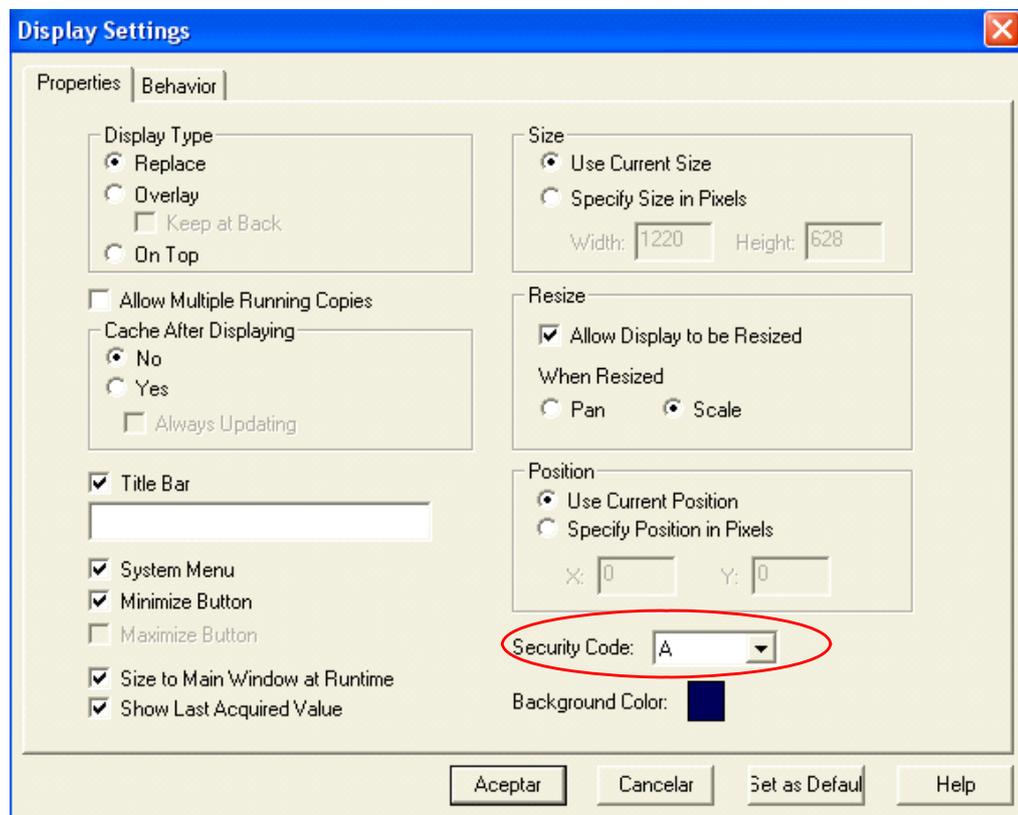


Figura. 5.139. Desarrollo Práctica #15 (11)

**Ingeniero:** Este usuario tiene acceso a todos los códigos de seguridad. Inicialmente tiene asignado el Password: *ingeniero*, el mismo que puede ser cambiado en la HMI (figura 5.140.).

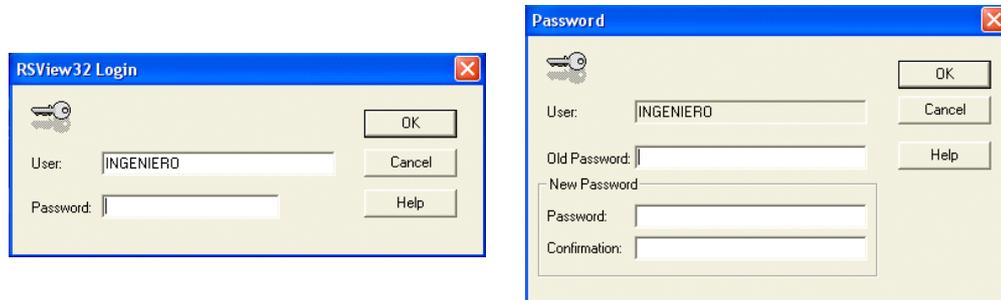


Figura. 5.140. Desarrollo Práctica #15 (12)

## 7. Sintonización de Controlador

### Controlador Primario

Para el controlador Primario, se utilizaron los mismos valores de sintonización que se obtuvieron en la sintonización de la práctica “PID DE TEMPERATURA” debido a que la planta utilizada es la misma.

Es decir, se emplearon los siguientes valores:

$$\boxed{Kc = 3.5} \quad \boxed{Ti = 0.5} \quad \boxed{Td = 0.12}$$

### Controlador Secundario

Para el controlador secundario se empleó únicamente un control proporcional con una ganancia  $\boxed{Kc = 10}$

8. Finalmente, se realizaron las pruebas necesarias para verificar que el controlador funciona correctamente de acuerdo al problema planteado.

### 5.15.5. PRUEBAS Y RESULTADOS

Durante el desarrollo de la práctica se realizaron las pruebas respectivas tanto del software (programación del PLC y de la HMI) como del hardware (Conexión del PLC y del Módulo Degem de simulación “AIR FLOW TEMPERATURE CONTROL

SYSTEM PCT-2”), a través de las cuales se verificó que el PLC y la HMI trabajan correctamente de acuerdo a la lógica del problema planteado en esta práctica.

#### **5.15.6. CONCLUSIONES Y RECOMENDACIONES**

##### **Conclusiones**

- El control en cascada permite mejorar notoriamente el desempeño de un esquema de control realimentado en ciertos procesos, razón por la cual es una técnica muy utilizada en procesos industriales.

##### **Recomendaciones**

- Se recomienda verificar el tipo de controlador (de acción inversa ó acción directa) que se debe emplear tanto en el controlador primario como secundario, para obtener un correcto desempeño del sistema.

## 5.16. CONTROL DE PROCESOS 7: CONTROL SELECTIVO

### 5.16.1. OBJETIVOS

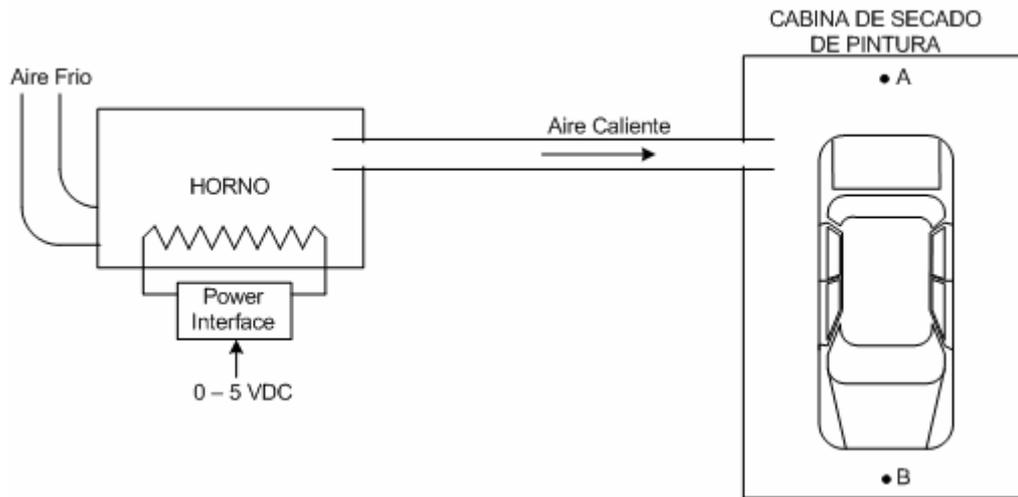
- Familiarizarse con la técnica de Control Selectivo.
- Crear una HMI (Interfase Humano Máquina) que interactúe con el PLC.

### 5.16.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley de la Serie SLC 500 (5/03 o superior)
- ✓ 1 Módulo de Entradas Discretas.
- ✓ 1 Módulo de Salidas Discretas.
- ✓ 1 Módulo de Entradas Analógicas.
- ✓ 1 Módulo de Salidas Analógicas.
- ✓ 1 Computador Personal, con instalación de RSLogix500, RSLinx y RSView32.
- ✓ 1 Módulo Degem de simulación PCT-2 “AIR FLOW TEMPERATURE CONTROL SYSTEM”.
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión

### 5.16.3. PLANTEAMIENTO

En una planta de pintura de autos, se desea controlar la temperatura de una cabina de secado de pintura que trabaje en un rango de temperatura de 40 a 50 °C. Esta cabina es calentada a través de un flujo de aire caliente que llega a través de una tubería desde un horno, puesto que no se puede controlar el flujo de aire, se debe controlar el sistema calentador del Horno. Además se poseen dos sensores de temperatura, uno ubicado en la parte frontal de la cabina (A) y el otro ubicado en la parte posterior de la cabina (B) como se muestra en la figura 5.141.



**Figura. 5.141. Planteamiento Práctica #16 (1)**

Diseñar e implementar un sistema de Control Selectivo que permita realizar el control de la temperatura de la cabina. El error en estado estacionario permitido es máximo de  $\pm 0.1^{\circ}\text{C}$ .

Adicionalmente con el software RSView32, crear una HMI que permita interactuar al operador con el sistema controlador, la cual deberá tener las siguientes características:

Dos cuentas de usuario: INGENIERO (con contraseña) y OPERADOR

Pantalla Principal con el siguiente diseño referencial:

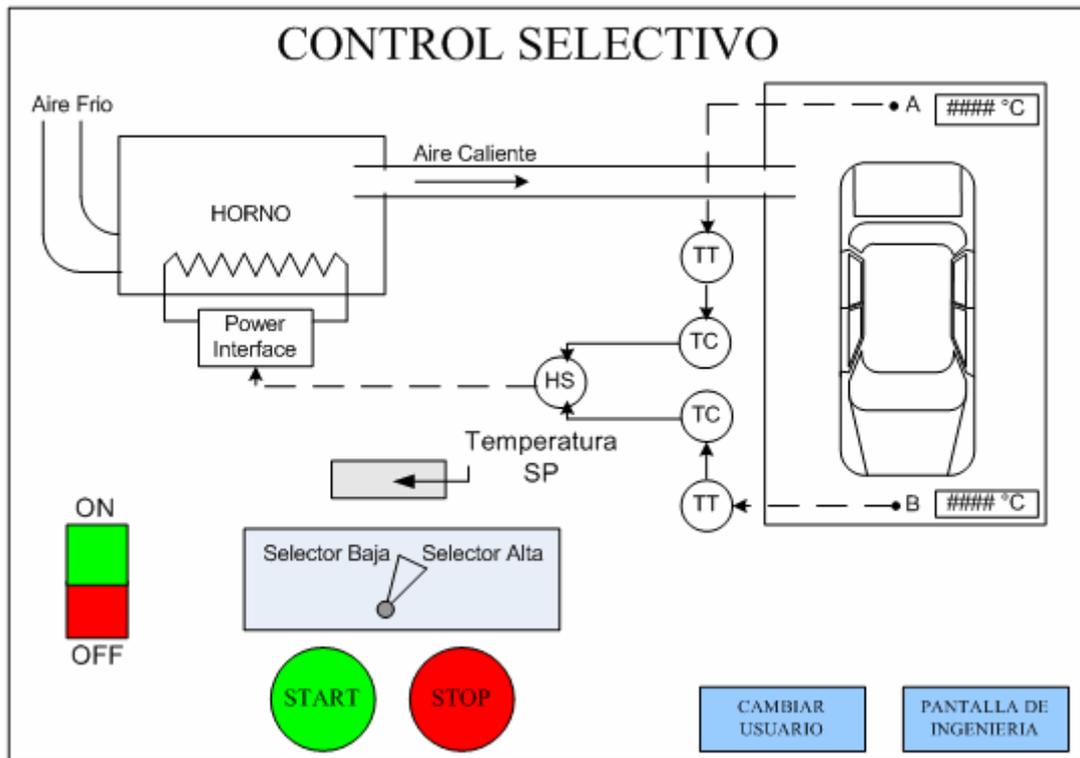


Figura. 5.142. Planteamiento Práctica #16 (2)

- ✓ El botón de “Pantalla de Ingeniería” despliega dicha pantalla únicamente si se encuentra trabajando con el usuario “INGENIERO”

Pantalla de Ingeniería (Acceso permitido únicamente para el usuario INGENIERO)

- ✓ Control de activación del controlador.
- ✓ Selector de Baja y Alta
- ✓ Selección de la Temperatura de la Cabina (Setpoint).
- ✓ Ingreso de parámetros de calibración de los controladores.
- ✓ Gráfica de tendencia (Trend) de la Temperatura real y la deseada del controlador A.
- ✓ Gráfica de tendencia (Trend) de la Temperatura real y la deseada del controlador B.
- ✓ Gráfica de tendencia (Trend) de Voltaje de Salida de ambos controladores.

#### 5.16.4. DESARROLLO

1. Configuración de driver en RSLinx

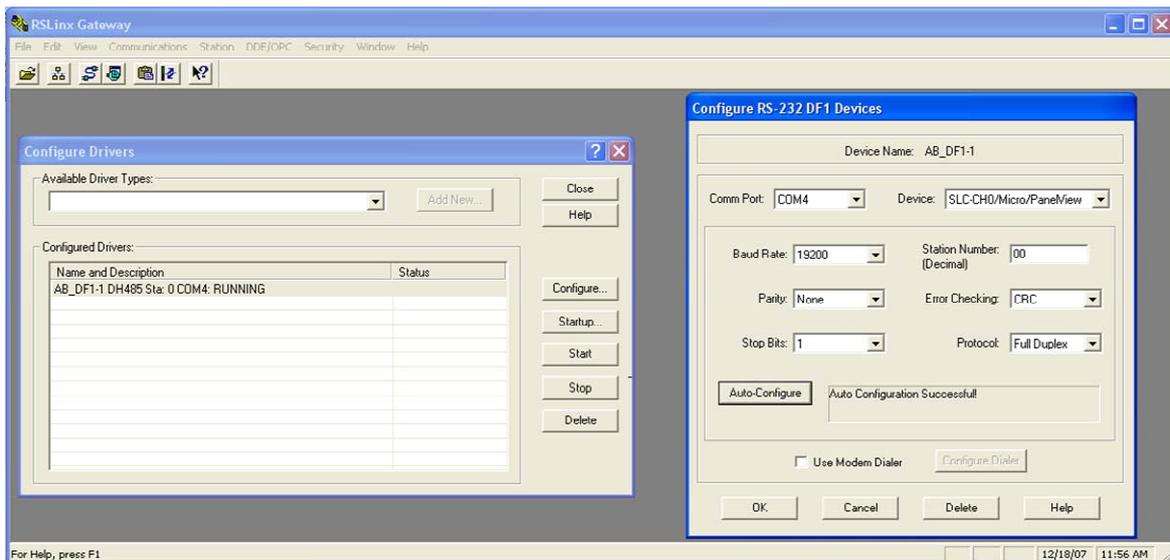


Figura. 5.143. Desarrollo Práctica #16 (1)

2. Creación del archivo y configuración del PLC en el software RSLogix500.

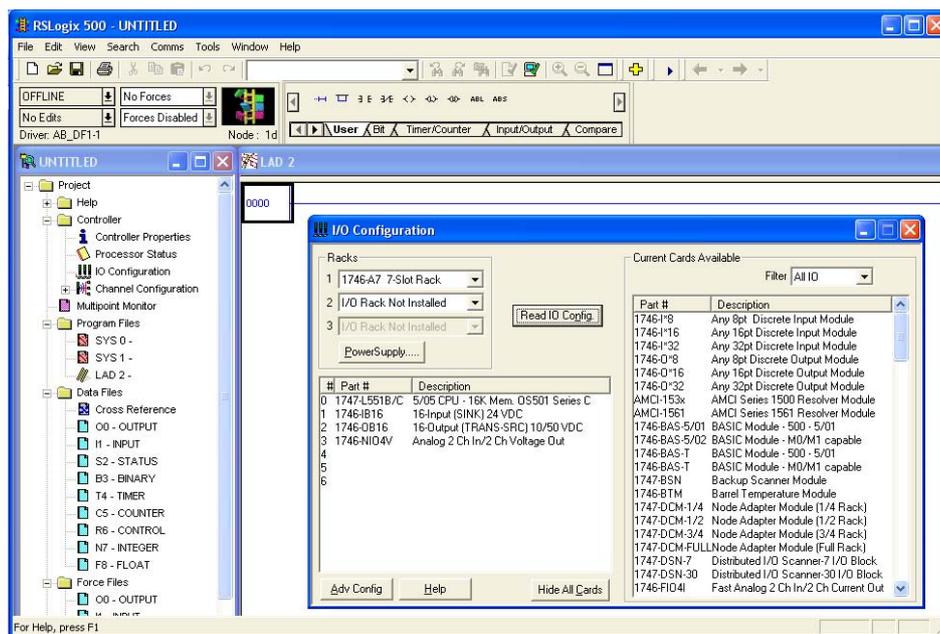


Figura. 5.144. Desarrollo Práctica #16 (2)

3. La programación Ladder que cumple con la lógica de control planteada en esta práctica es la siguiente:

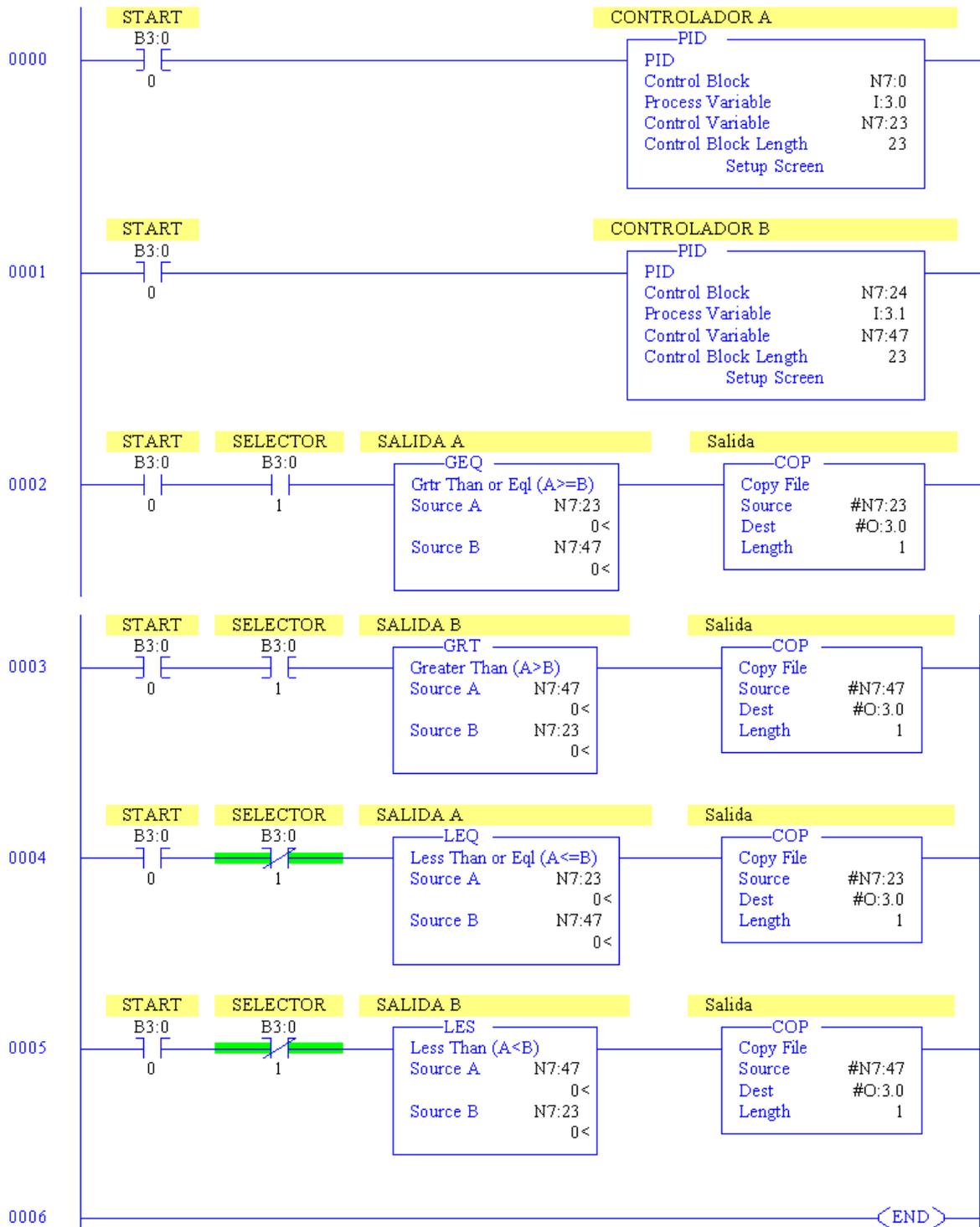


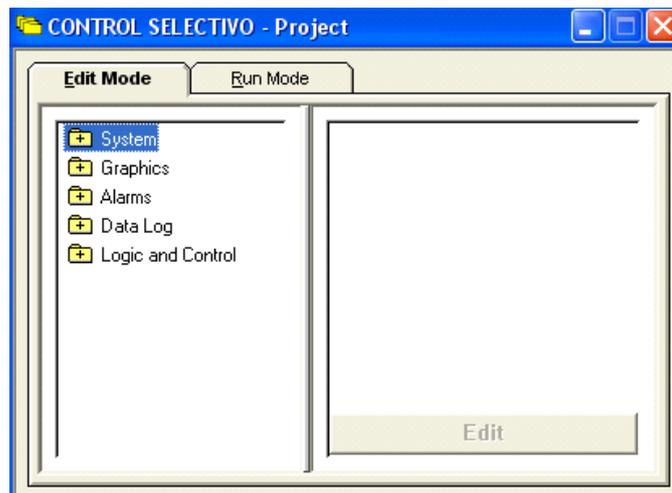
Figura. 5.145. Desarrollo Práctica #16 (3)

- La asignación de direcciones en el PLC de los diferentes registros, entradas y salidas utilizadas, se la hizo de acuerdo a la siguiente tabla:

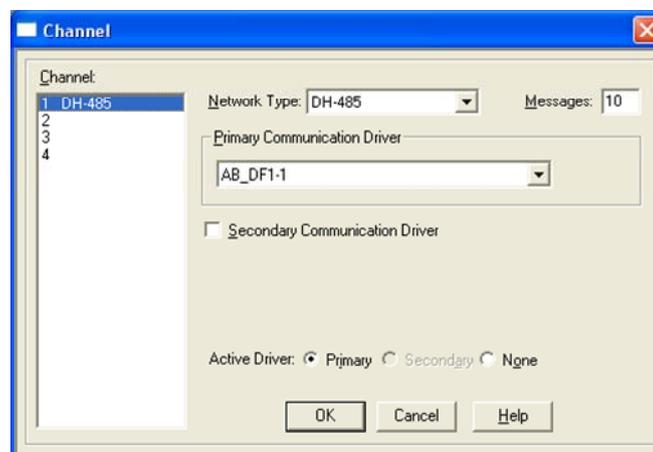
Nombre	Tipo	Dirección
PV A	Entrada Analógica	I:3.0
PV B	Entrada Analógica	I:3.1
SALIDA	Salida Analógica	O:3/0
START	Bit	B3:0/0
SELECTOR	Bit	B3:0/1
CONTROL A	Entero	N7:0 - N7:22
CONTROL B	Entero	N7:24 - N7:46

**Tabla. 5.27. Direcciones utilizadas Práctica #16**

5. Abrir el software RSView32, configurar el canal de comunicación, configurar el nodo de comunicación.



**Figura. 5.146. Desarrollo Práctica #16 (4)**



**Figura. 5.147. Desarrollo Práctica #16 (5)**

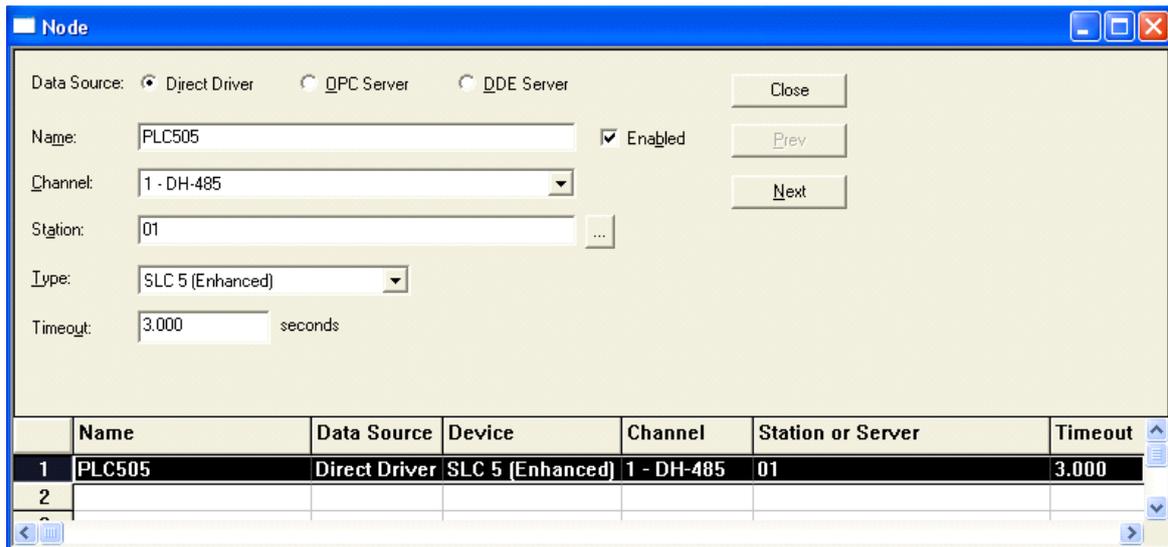


Figura. 5.148. Desarrollo Práctica #16 (6)

## 6. Creación de la HMI que interactúe con el PLC

### INTERFASE HUMANO MÁQUINA (HMI)

La HMI creada en base al esquema planteado consta de dos pantallas:

#### Pantalla Principal

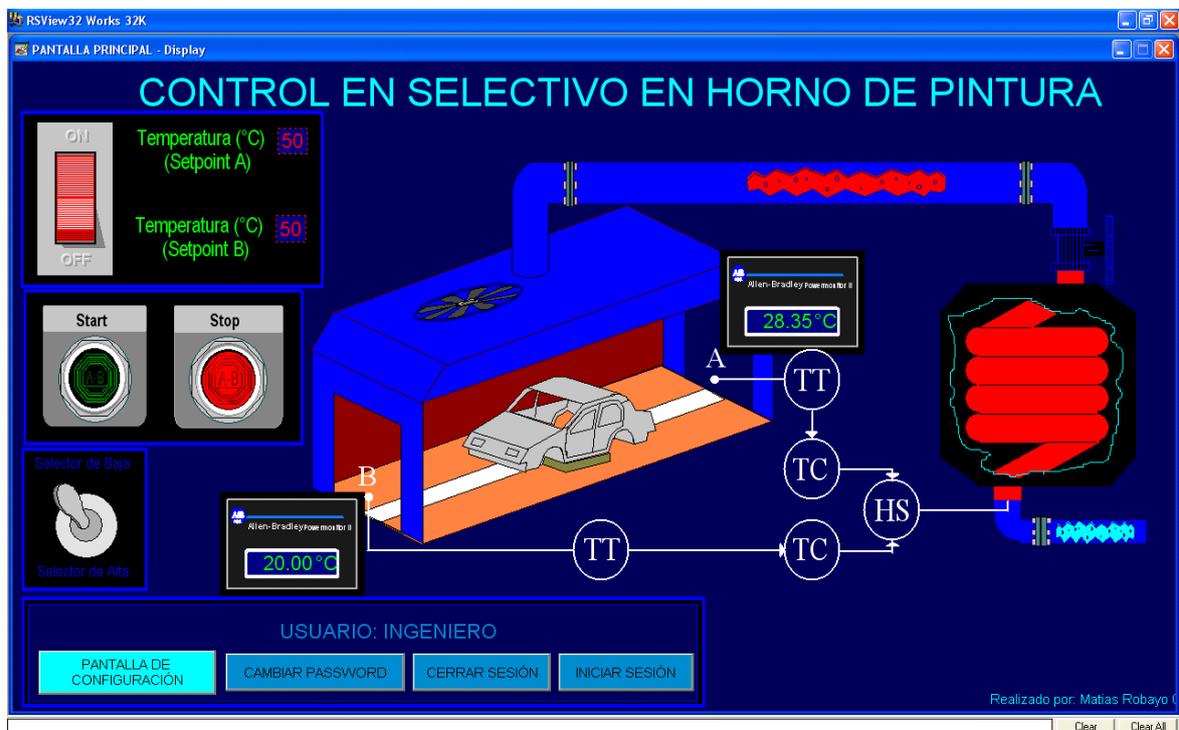


Figura. 5.149. Desarrollo Práctica #16 (7)

**Pantalla de Control**



Figura. 5.150. Desarrollo Práctica #16 (8)

**TAGS**

En la siguiente tabla se muestran los tags empleados en la programación de la HMI y sus características:

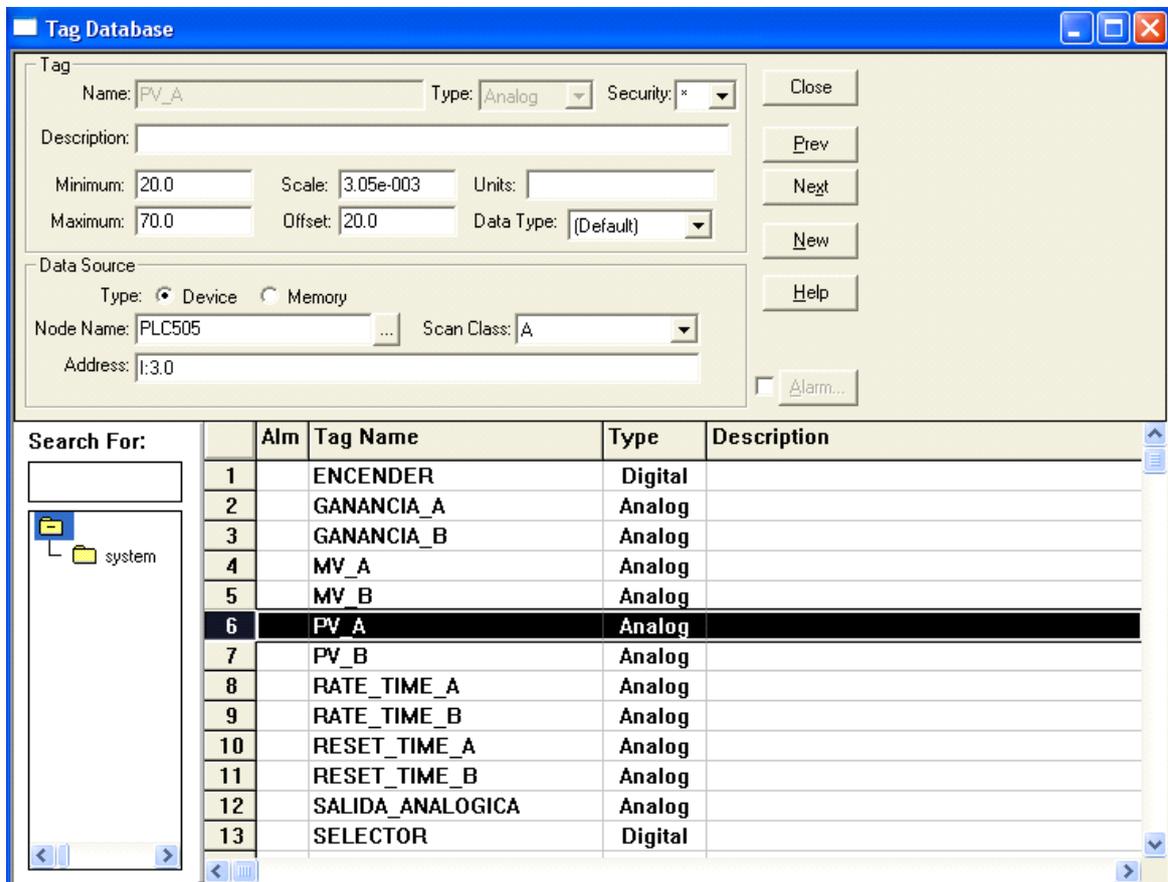
NOMBRE	TIPO	CARACTERÍSTICAS					
ENCENDER	Digital	Alarmed: No	Security: *	Memory Tag	Initial Value: Off		
		Off Label: Off	On Label: On				
GANANCIA_A	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:3	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
GANANCIA_B	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:27	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
MV_A	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:16	Scan Class: A
		Minimum :	Maximum :	Scale : 1	Offset : 0	Units:	Data Type:

		0	100				(Default)
MV_B	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:40	Scan Class: A
		Minimum : 0	Maximum : 100	Scale : 1	Offset : 0	Units:	Data Type: (Default)
PV_A	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: I:3.0	Scan Class: A
		Minimum : 20	Maximum : 70	Scale : 0.003050	Offset : 20	Units:	Data Type: (Default)
PV_B	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: I:3.1	Scan Class: A
		Minimum : 20	Maximum : 70	Scale : 0.003050	Offset : 20	Units:	Data Type: (Default)
RATE_TIME_A	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:5	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
RATE_TIME_B	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:29	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
RESET_TIME_A	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:4	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
RESET_TIME_B	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:28	Scan Class: A
		Minimum : 0	Maximum : 32767	Scale : 1	Offset : 0	Units:	Data Type: (Default)
SALIDA_ANALOGICA	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: O:3.0	Scan Class: A
		Minimum : 0	Maximum : 163838	Scale : 1	Offset : 0	Units:	Data Type: (Default)
SELECTOR	Digital	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: B3:0/1	Scan Class: A
		Off Label: Off	On Label: On				
SETPOINT_A	Analog	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: N7:2	Scan Class: A
		Minimum : 20	Maximum : 70	Scale : 1	Offset : 0	Units:	Data Type: (Default)
SETPOINT_B	Analog	Alarmed: No	Security: *	Device Tag	Node Name:	Address: N7:26	Scan Class: A

					PLC505		
		Minimum : 20	Maximum : 70	Scale : 1	Offset : 0	Units:	Data Type: (Default)
START	Digital	Alarmed: No	Security: *	Device Tag	Node Name: PLC505	Address: B3:0/0	Scan Class: A
		Off Label: Off	On Label: On				

**Tabla. 5.28. Tags utilizados Práctica #16**

Estos tags se encuentran configurados en la ventana *Tag Database* de acuerdo a las características que se encuentran en la tabla anterior, a continuación se muestra un ejemplo de esta configuración:



**Figura. 5.151. Desarrollo Práctica #16 (9)**

Como se puede ver en la figura 5.151., el tag PV\_A es de tipo análogo y se encuentra asociado a la dirección I:3.0 (entrada analógica) a través del nodo PLC, tiene un Mínimo de 20 y un Máximo de 70 puesto que ese es el rango de temperaturas en unidades de ingeniería. Además, se encuentra escalado para transformar las unidades crudas en unidades de ingeniería con las siguientes fórmulas:

$$Scale = \frac{(Escalado \textit{ M}{\acute{a}}ximo - Escalado \textit{ M}{\acute{i}}nimo)}{(Entrada \textit{ M}{\acute{a}}xima - Entrada \textit{ M}{\acute{i}}nima)} = \frac{(70 - 20)}{(16383 - 0)} = 0.00305$$

$$Offset = Escalado \textit{ M}{\acute{i}}nimo - (Entrada \textit{ M}{\acute{i}}nima \times Scale) = 20 - (0 \times 0.00305) = 20$$

## USUARIOS

Como se muestra en la siguiente figura 5.152., esta HMI posee dos cuentas de usuario, “Default” e “Ingeniero”.

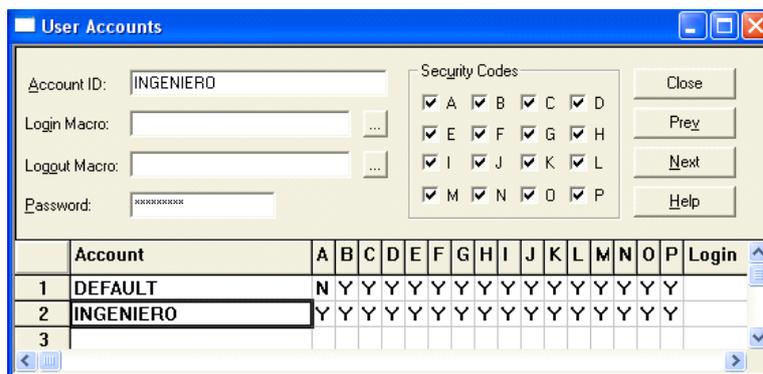


Figura. 5.152. Desarrollo Práctica #16 (10)

Default: Es el usuario que se muestra como “Operador” cuando se corre el programa, como se puede ver no posee acceso al código de seguridad A. Para restringir el acceso de este usuario a la Pantalla de Configuración, a dicha pantalla se le ha asignado el código de seguridad A, como se muestra en la figura 5.153.

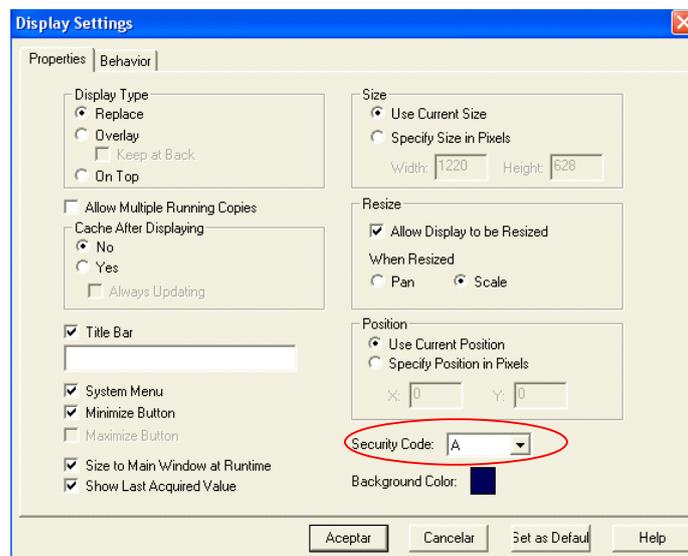


Figura. 5.153. Desarrollo Práctica #16 (11)

Ingeniero: Este usuario tiene acceso a todos los códigos de seguridad. Inicialmente tiene asignado el Password: *ingeniero*, el mismo que puede ser cambiado en la HMI (figura 5.154.).

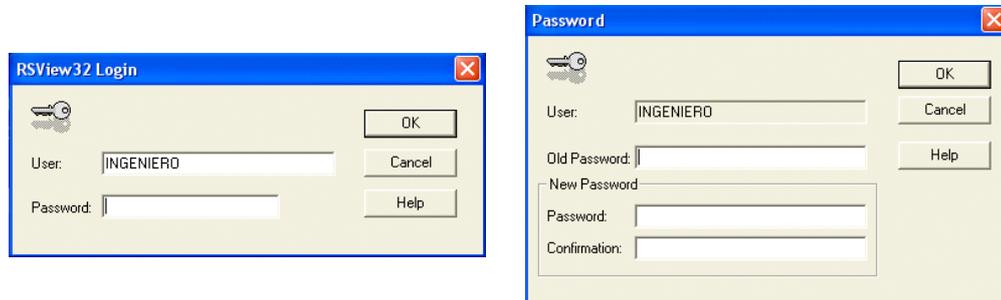


Figura. 5.154. Desarrollo Práctica #16 (12)

## 7. Sintonización de Controlador

Para ambos controladores, se utilizaron los mismos valores de sintonización que se obtuvieron en la sintonización de la práctica “PID DE TEMPERATURA” debido a que la planta utilizada es la misma.

Es decir, se emplearon los siguientes valores:

$$\boxed{Kc = 3.5} \quad \boxed{Ti = 0.5} \quad \boxed{Td = 0.12}$$

- Finalmente, se realizaron las pruebas necesarias para verificar que el controlador funciona correctamente de acuerdo al problema planteado.

### 5.16.5. PRUEBAS Y RESULTADOS

Durante el desarrollo de la práctica se realizaron las pruebas respectivas tanto del software (programación del PLC y de la HMI) como del hardware (Conexión del PLC y del Módulo Degem de simulación “AIR FLOW TEMPERATURE CONTROL SYSTEM PCT-2”), a través de las cuales se verificó que el PLC y la HMI trabajan correctamente de acuerdo a la lógica del problema planteado en esta práctica.

### **5.16.6. CONCLUSIONES Y RECOMENDACIONES**

#### **Conclusiones**

- El control selectivo permite limitar en un valor alto o bajo la variable del proceso para evitar que se produzcan daños en los equipos o en el proceso.

#### **Recomendaciones**

- Se recomienda programar toda la lógica de control en el PLC y no en la HMI, lo que garantizará que el controlador siga funcionando correctamente a pesar de que se pierda la comunicación entre la HMI y el PLC.

## 5.17. CONTROL DE PROCESOS 8: SCADA 1 COMUNICACIÓN SERIAL

### 5.17.1. OBJETIVOS

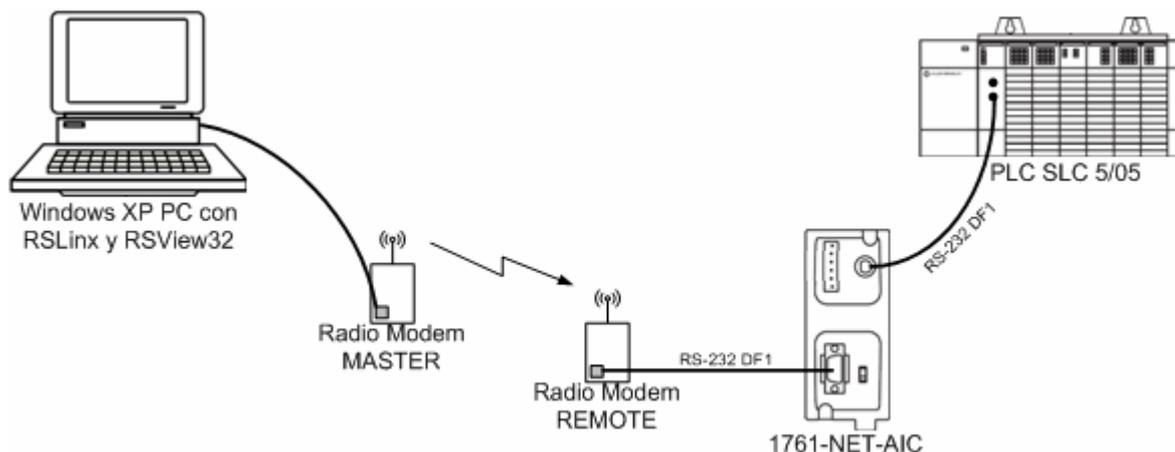
- Configurar correctamente el canal (0) de comunicación serial del PLC SLC 5/05.
- Conocer las aplicaciones, configuración y conexión del módulo conversor de interfase avanzado 1761-NET-AIC.
- Configurar los Radio Modems Seriales SRM6100
- Crear una HMI que interactúe con el PLC y permita controlar de manera remota las entradas y salidas del PLC.

### 5.17.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley SLC 5/05
- ✓ 1 Módulo de Entradas Discretas.
- ✓ 1 Módulo de Salidas Discretas.
- ✓ 1 Radio Modem Master Data-Linc SRM6100
- ✓ 1 Radio Modem Remote Data-Linc SRM6100
- ✓ 1 Módulo conversor de interfase avanzado 1761-NET-AIC
- ✓ 1 Computador Personal, con instalación de RSLogix500, RSLinx y RSView32.
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ Cables de Conexión Serial.

### 5.17.3. PLANTEAMIENTO

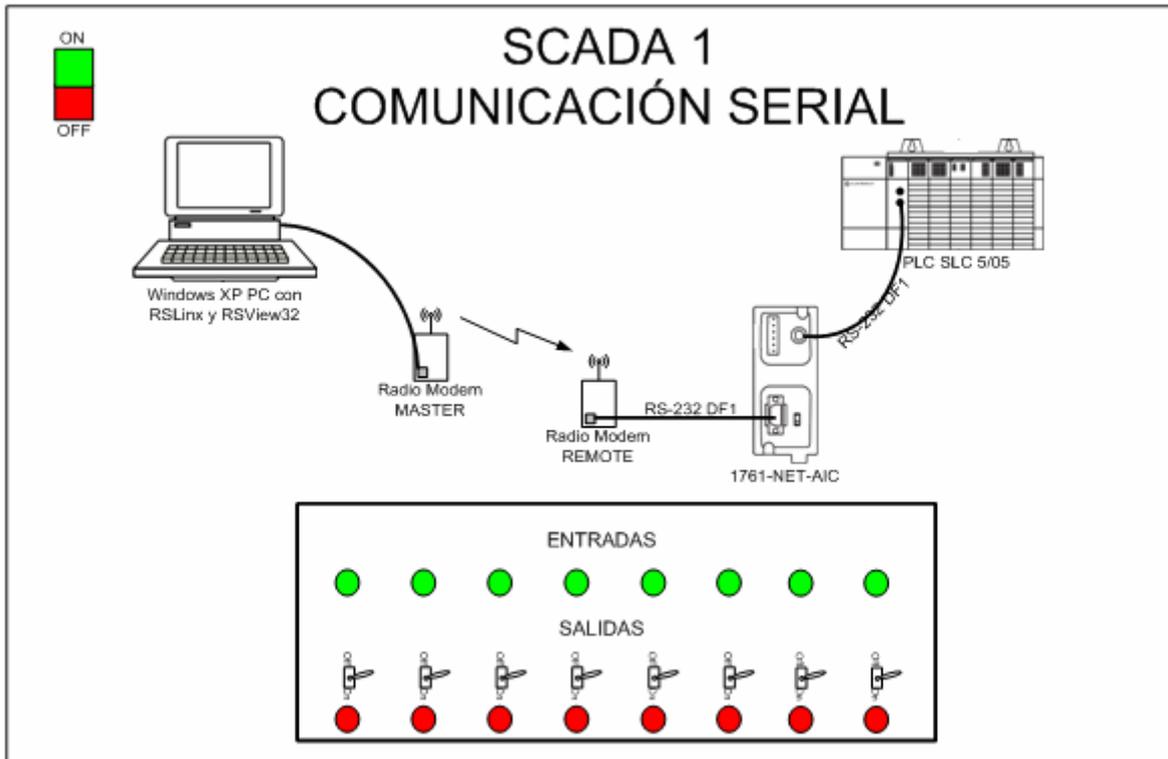
Se desea conectar los siguientes equipos en la configuración mostrada en la figura 5.155.



**Figura. 5.155. Planteamiento Práctica #17 (1)**

El computador personal deberá poseer una HMI (realizada en RSVIEW32) que permita monitorear las Entradas Discretas y controlar las Salidas Discretas del PLC.

La HMI deberá contar con una sola pantalla con el siguiente esquema:

**Figura. 5.156. Planteamiento Práctica #17 (2)**

#### 5.17.4. DESARROLLO

Para la realización de esta práctica el procedimiento a seguir es el siguiente:

1. Configuración del Hyper Terminal para la comunicación con los Radio Modems



**Figura. 5.157. Desarrollo Práctica #17 (1)**

2. La configuración del Radio Modem MASTER es la siguiente:

**(0) Set Operation Mode**

(0) Point to Point Master

**(1) Set Baud Rate**

(5) 19,200

(A) Data, Parity 0

(B) MODBus RTU 0

**(2) Edit Call Book**

Entry Number Repeater1 Repeater2

(0) 240-2334 (Call number del Radio Modem Remote)

Todas las demás entradas se encuentran con 000-0000

**(3) Edit Radio Transmission Characteristics**

- (0) FreqKey 5
- (1) Max Packet Size 9
- (2) Min Packet Size 2
- (3) Xmit Rate 1
- (4) RF Data Rate 2
- (5) RF Xmit Power 5
- (6) Slave Security 0
- (7) RTS to CTS 0
- (8) Retry Time Out
- (9) Lowpower Mode 0

3. La configuración del Radio Modem SLAVE es la siguiente:

**(0) Set Operation Mode**

- (1) Point to Point Slave

**(1) Set Baud Rate**

- (5) 19,200
- (A) Data, Parity 0
- (B) MODBus RTU 0

**(2) Edit Call Book**

Entry Number Repeater1 Repeater2

- (0) 240-2357 (Call number del otro Radio Modem)

---

Todas las demás entradas se encuentran con 000-0000

**(3) Edit Radio Transmission Characteristics**

- (0) FreqKey 5
- (1) Max Packet Size 9
- (2) Min Packet Size 2
- (3) Xmit Rate 1
- (4) RF Data Rate 2
- (5) RF Xmit Power 5
- (6) Slave Security 0
- (7) RTS to CTS 0
- (8) Retry Time Out 255
- (9) Lowpower Mode 0

4. Configuración del Driver Serial en RSLinx

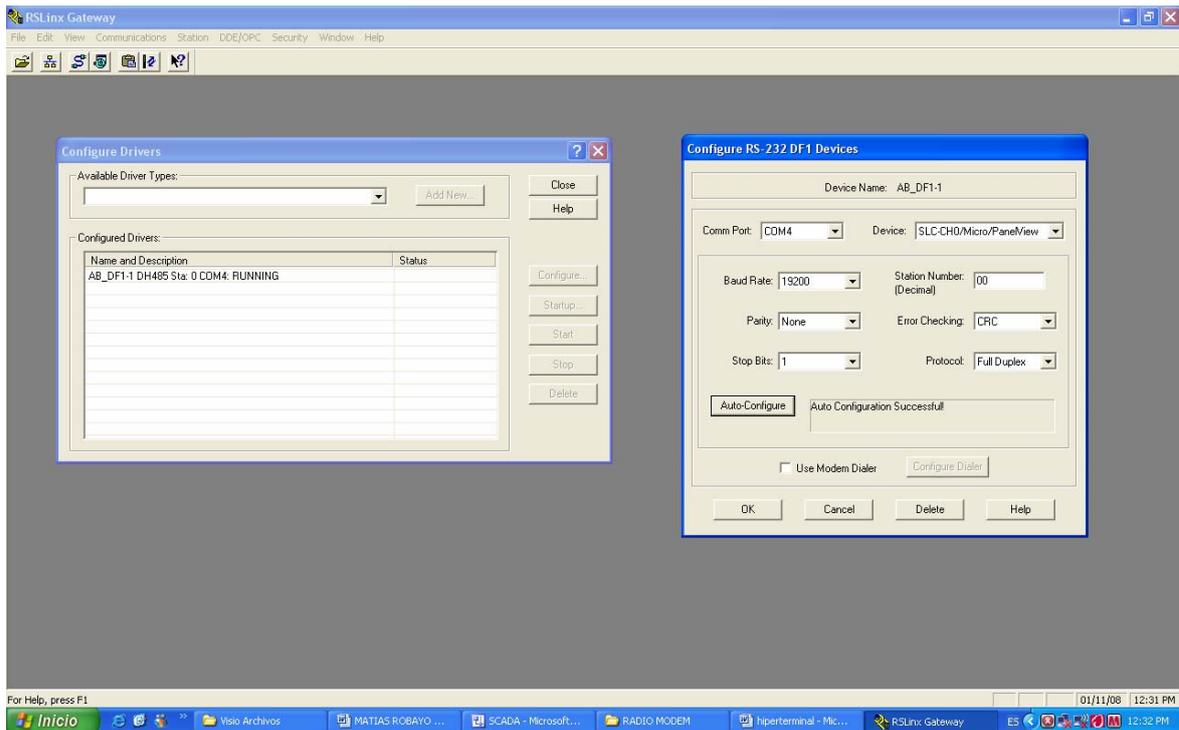


Figura. 5.158. Desarrollo Práctica #17 (2)

## 5. Configuración del canal 0 (DF1 Full Duplex) del PLC SLC 5/05

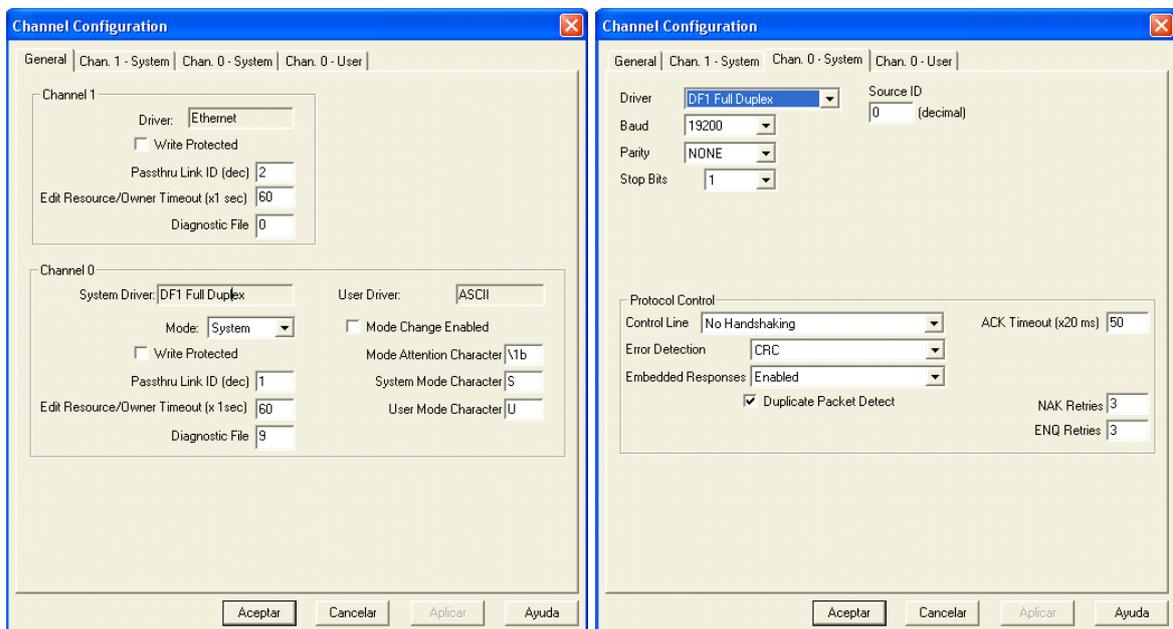


Figura. 5.159. Desarrollo Práctica #17 (3)

6. Debido a que se requiere un registro auxiliar para activar o desactivar las salidas desde la HMI, la programación Ladder que cumple con la lógica de control planteada en esta práctica es la siguiente:

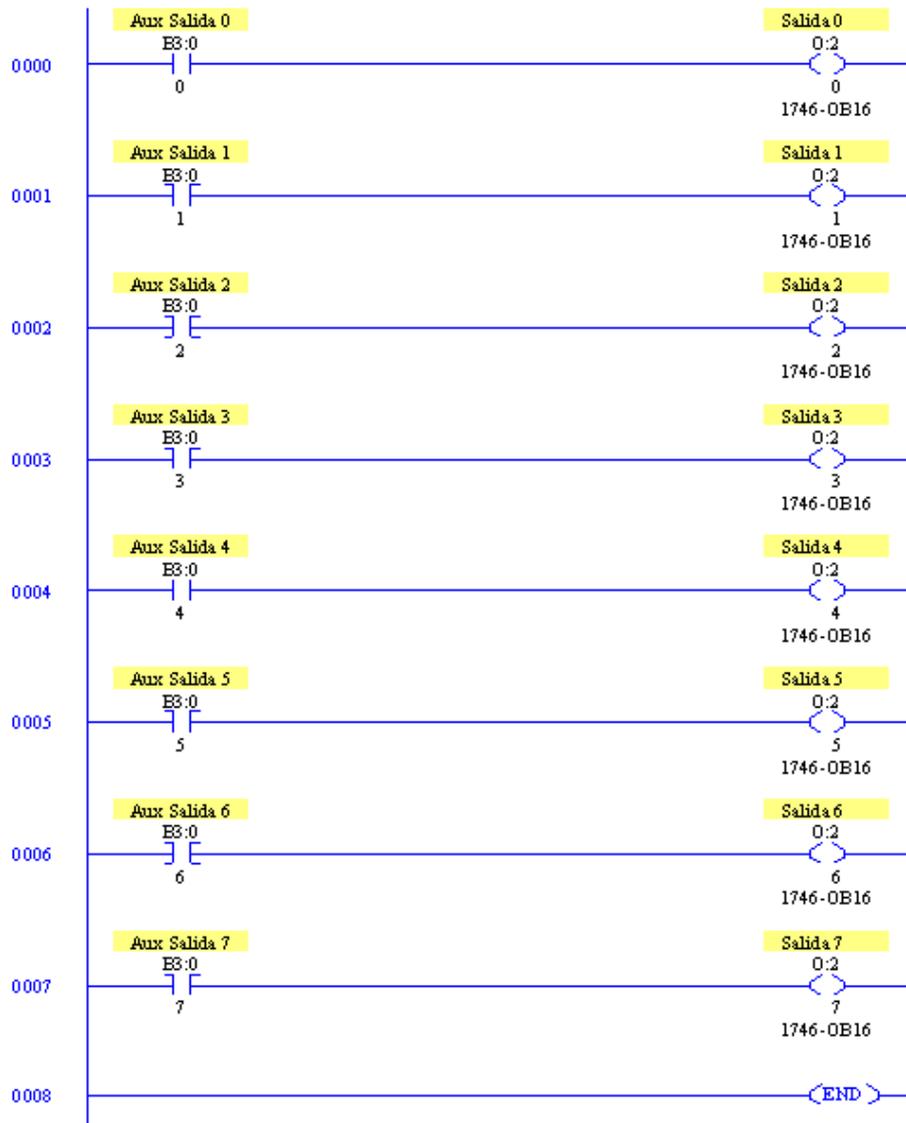


Figura. 5.160. Desarrollo Práctica #17 (4)

7. Abrir el software RSView32, configurar el canal de comunicación, configurar el nodo de comunicación.

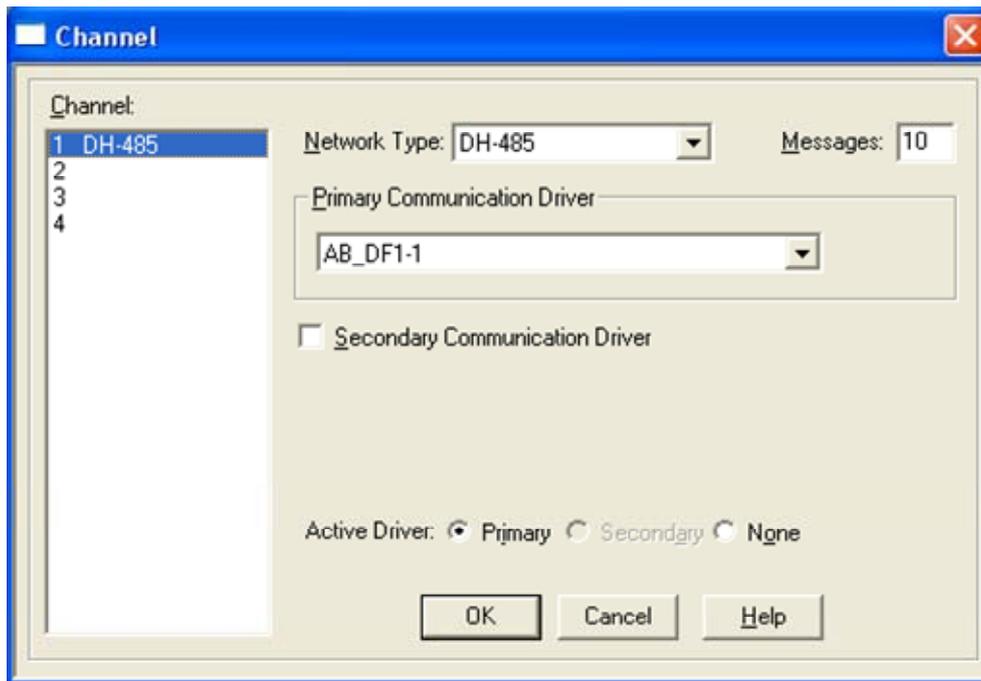


Figura. 5.161. Desarrollo Práctica #17 (5)

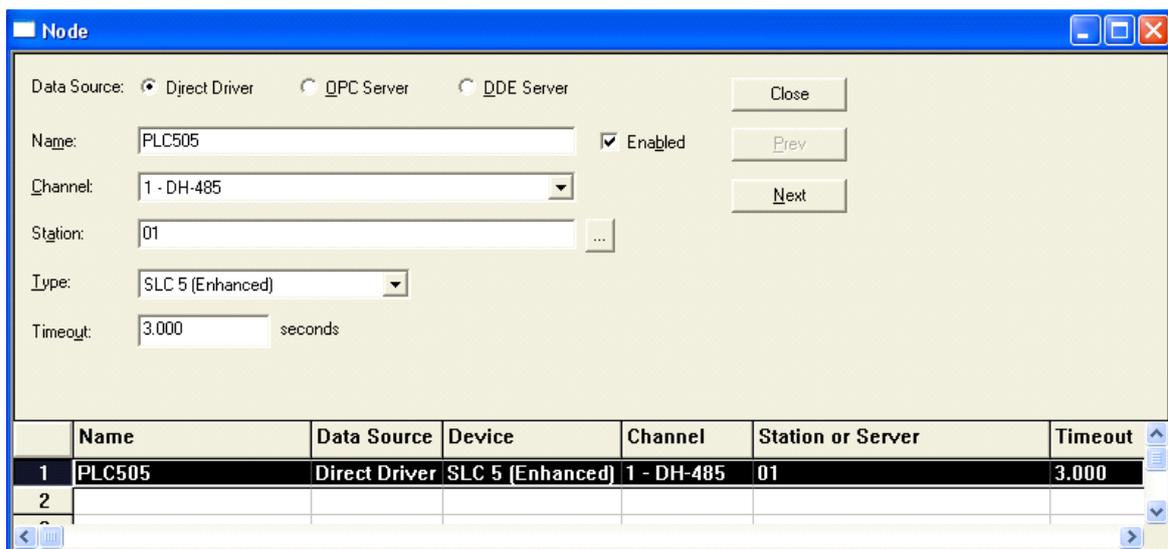


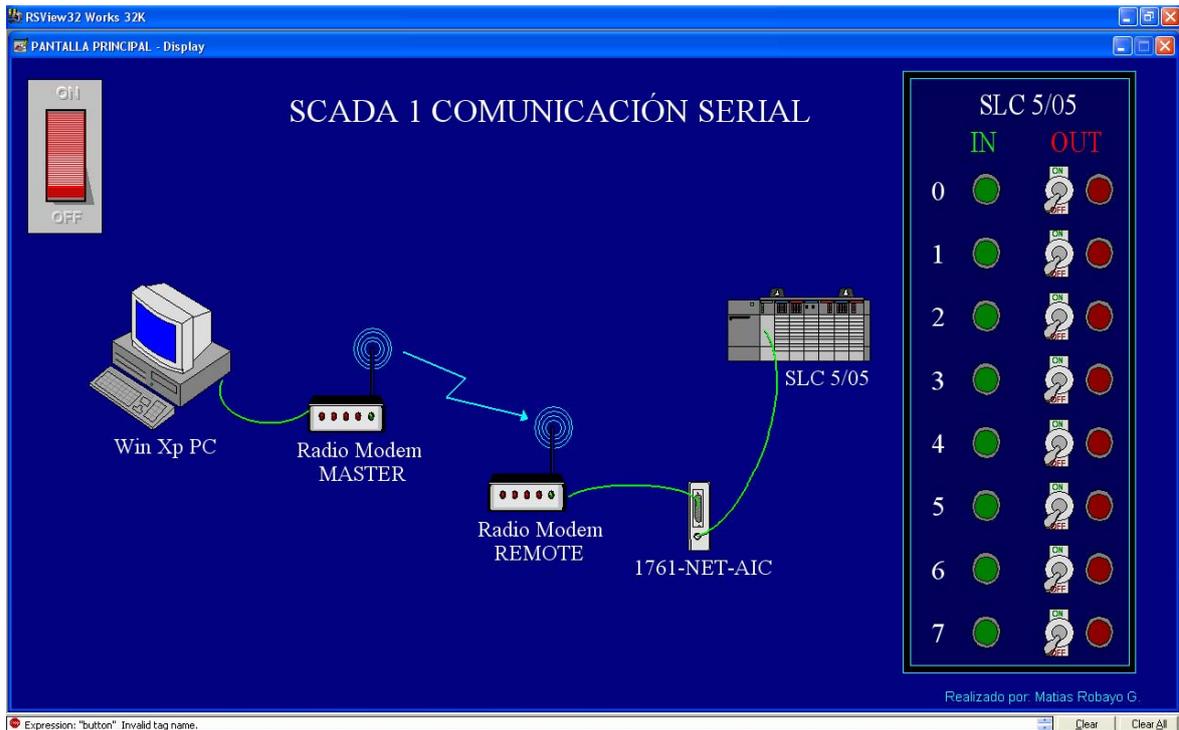
Figura. 5.162. Desarrollo Práctica #17 (6)

## 9. Creación de la HMI que interactúe con el PLC

### INTERFASE HUMANO MÁQUINA (HMI)

La HMI creada en base al esquema planteado consta la siguiente pantalla:

**Pantalla Principal**



**Figura. 5.163. Desarrollo Práctica #17 (7)**

**TAGS**

En la siguiente tabla se muestran los tags empleados en la programación de la HMI:

NOMBRE	TIPO	CARACTERÍSTICAS					
ENCENDER	Digital	Alarmed: No	Security: *	Memory Tag	Initial Value: Off		
		Off Label: Off	On Label: On				
SLC505_IN0	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: I:1/0	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_IN1	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: I:1/1	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_IN2	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: I:1/2	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_IN3	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: I:1/3	Scan Class: A
		Off Label:	On Label:				

		Off	On				
SLC505_IN4	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: I:1/4	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_IN5	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: I:1/5	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_IN6	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: I:1/6	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_IN7	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: I:1/7	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_OUT0	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: B3:0/0	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_OUT1	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: B3:0/1	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_OUT2	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: B3:0/2	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_OUT3	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: B3:0/3	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_OUT4	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: B3:0/4	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_OUT5	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: B3:0/5	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_OUT6	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: B3:0/6	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_OUT7	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: B3:0/7	Scan Class: A
		Off Label: Off	On Label: On				

Tabla. 5.29. Tags utilizados Práctica #17

8. Finalmente, se realizaron las pruebas necesarias para verificar que la HMI y los PLC's funcionan correctamente y que se encuentran correctamente configurados.

### **5.17.5. PRUEBAS Y RESULTADOS**

Durante el desarrollo de la práctica se realizaron las pruebas respectivas tanto del software (programación de los PLC y de la HMI) como del hardware (Conexión de los PLC's, módulo ENIW y Radio Modems), a través de las cuales se verificó que todos los dispositivos trabajan correctamente de acuerdo a la lógica y configuración del problema planteado en esta práctica.

### **5.17.6. CONCLUSIONES Y RECOMENDACIONES**

#### **Conclusiones**

- El módulo 1761-NET-AIC+ brinda total protección al puerto serial de los radio módems al actuar como un aislador punto a punto en la conexión serial.
- Los radio modem Seriales permiten ampliar significativamente (hasta 24 Km.) la distancia máxima de comunicación de una red de datos serial.

#### **Recomendaciones**

- Se recomienda NO cambiar los parámetros "Radio Transmission Characteristics" de los radio módems a no ser que se tenga pleno conocimiento de lo que representan.
- Se recomienda para esta práctica, no configurar el canal 0 de los PLC SLC5/03 en modo "DF1 Radio Modem" puesto que este protocolo no es compatible con los protocolos DF1 Half y Full Duplex que son los que permiten comunicarse a la PC con el PLC.

## 5.18. CONTROL DE PROCESOS 9: SCADA 2 COMUNICACIÓN ETHERNET

### 5.18.1. OBJETIVOS

- Conocer el funcionamiento, configuración y aplicación de la Instrucción MSG.
- Conectar PLC's 5/03 a una red Ethernet, utilizando los módulos de interfase ethernet
- Configurar el canal 1 de los PLC's 5/05 para conectarlos a una red Ethernet
- Enviar mensajes de alarma vía e-mail, a través de los módulos de interfase ethernet
- Crear una HMI que interactúe con el PLC y permita controlar de manera remota varios procesos.

### 5.18.2. MATERIALES Y EQUIPOS

- ✓ 1 CPU Allen Bradley SLC 5/03
- ✓ 1 CPU Allen Bradley SLC 5/05
- ✓ 2 Módulos de Entradas Discretas.
- ✓ 2 Módulos de Salidas Discretas.
- ✓ 2 Radio Modems Ethernet Data-Linc SRM6310E
- ✓ 1 Switch Ethernet
- ✓ 1 Módulo de Interfase Ethernet 1761-NET-ENIW
- ✓ 1 Computador Personal, con instalación de RSLogix500, RSLinx y RSView32.
- ✓ 1 Punto de conexión a la red ethernet de la ESPE
- ✓ 1 Fuente de alimentación variable 0-24 VDC
- ✓ 4 Cables de conexión Ethernet
- ✓ 1 Cable 1761-CBL-PM02 PC to Micrologix
- ✓ 1 Cable Serial C232/SRM61/PC (Programación Radio Modem)

### 5.18.3. PLANTEAMIENTO

Se desea conectar los siguientes equipos en la configuración mostrada en la figura 5.164.

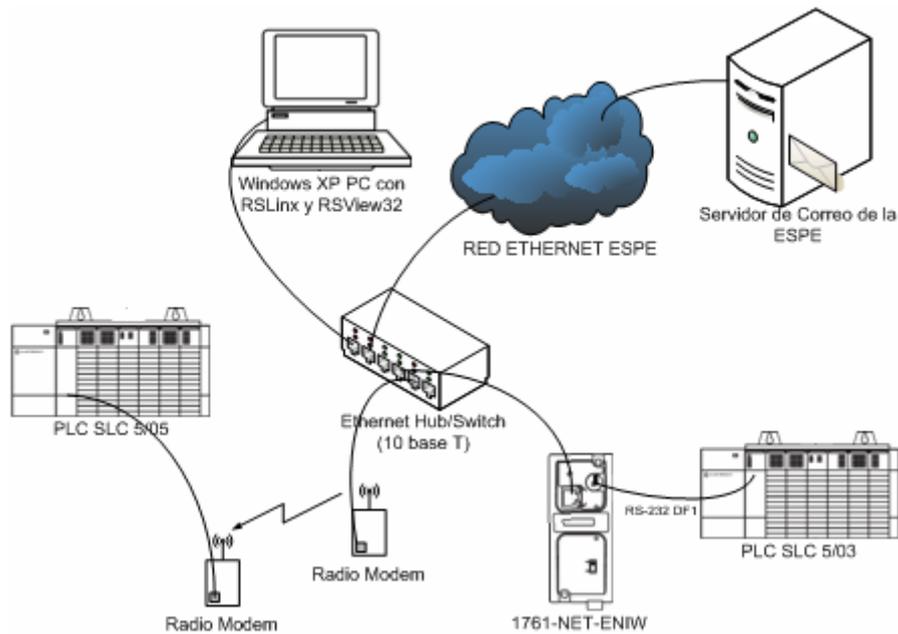


Figura. 5.164. Planteamiento Práctica #18 (1)

El computador personal deberá poseer una HMI (realizada en RSView32) que permita monitorear las Entradas Discretas y controlar las Salidas Discretas de los tres PLC's.

La HMI deberá contar con 3 pantallas con los siguientes esquemas:

### PANTALLA PRINCIPAL

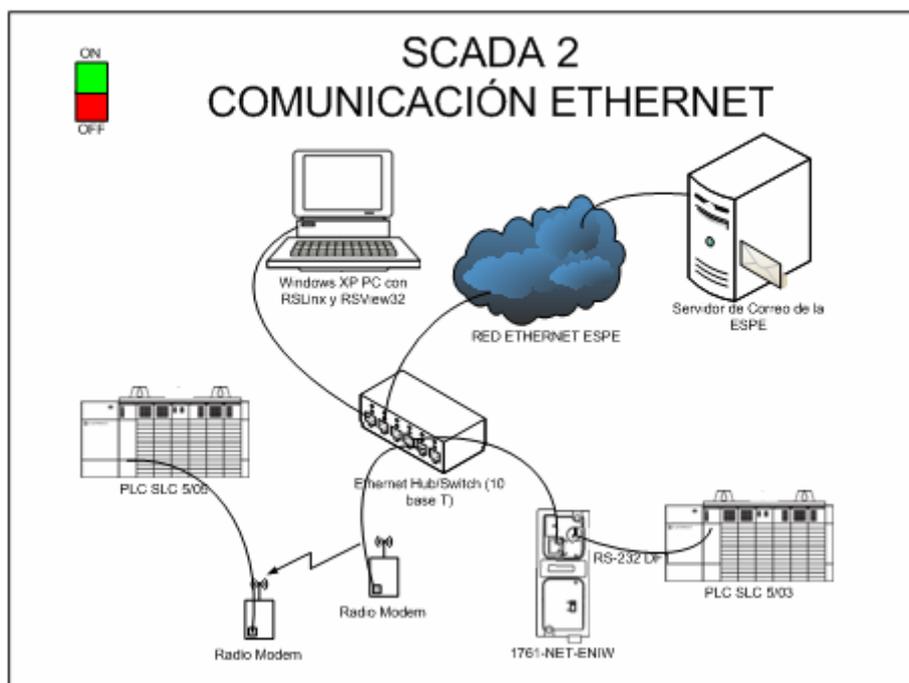
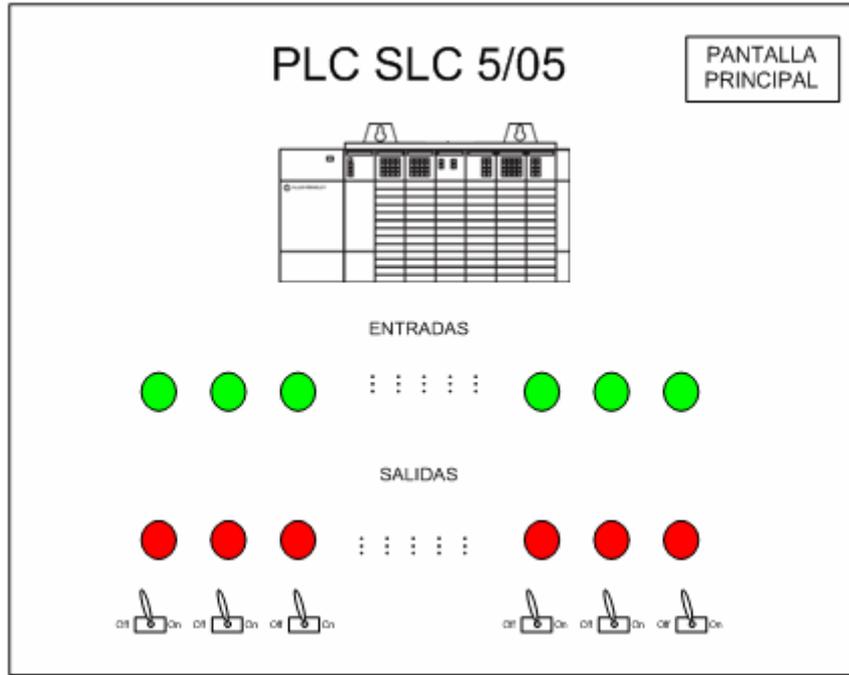


Figura. 5.165. Planteamiento Práctica #18 (2)

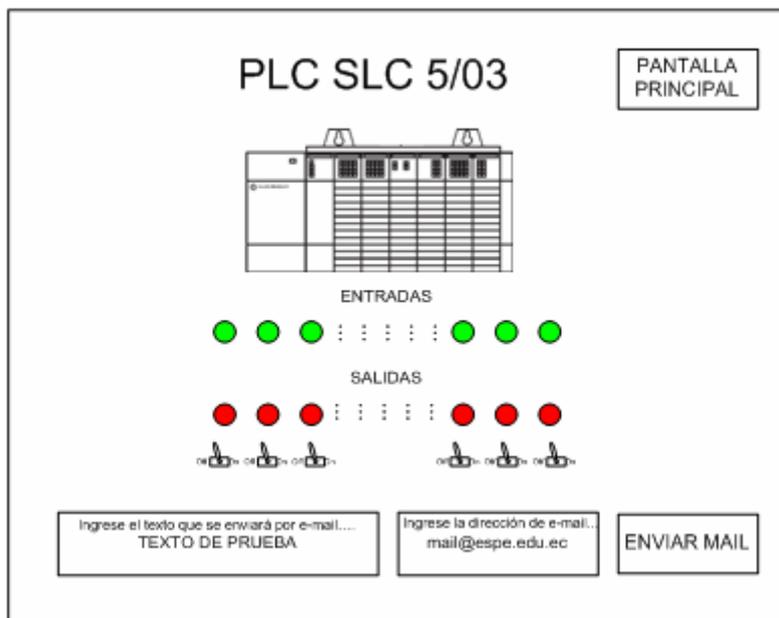
Al hacer clic con el Mouse sobre los PLC's, se desplegará la pantalla respectiva que permita monitorear y controlar las entradas y Salidas Discretas de dicho PLC. A continuación de muestra el esquema de cada pantalla.

**SLC 5/05**



**Figura. 5.166. Planteamiento Práctica #18 (3)**

**SLC 5/03**



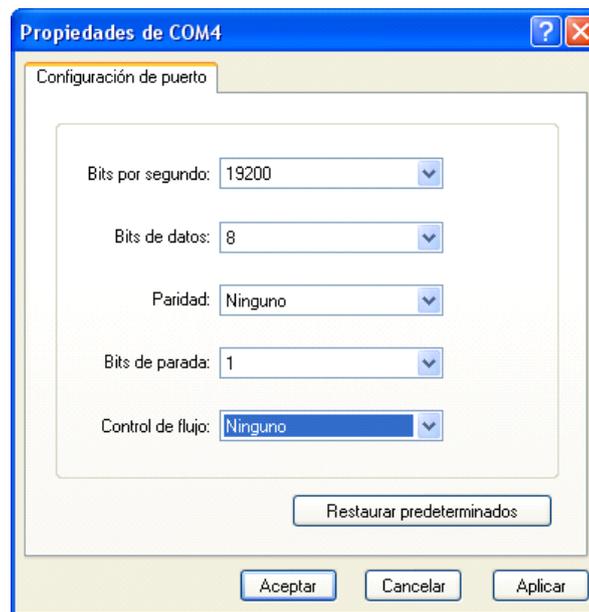
**Figura. 5.167. Planteamiento Práctica #18 (4)**

En esta pantalla, se incluirá la opción de envío de e-mail, a través del módulo 1761-NET-ENIW. Para lo que se requiere conocer la dirección IP del servidor de e-mail de la ESPE.

**Nota:** El número de entradas y salidas dependerá del PLC y los módulos utilizados.

#### 5.18.4. DESARROLLO

1. Configuración del Hyper Terminal para la comunicación con los Radio Modems



**Figura. 5.168. Desarrollo Práctica #18 (1)**

2. La configuración del Radio Modem MASTER es la siguiente:

**(0) Set Operation Mode**

- (0) Point to Point Master

**(1) Set Baud Rate**

- (0) 230,400

- (A) Data, Parity 0

- (B) MODBus RTU 0

- (C) RS232/485 0
- (D) Setup Port 3
- (E) TurnOffDelay 0 TurnOnDelay 0
- (F) FlowControl 0

## (2) Edit Call Book

Entry Number Repeater1 Repeater2

- (0) 242-2823 (Call number del otro Radio Modem)

Todas las demás entradas se encuentran con 000-0000

## (3) Edit Radio Transmission Characteristics

- (0) FreqKey 5
- (1) Max Packet Size 9
- (2) Min Packet Size 0
- (3) Xmit Rate 1
- (4) RF Data Rate 2
- (5) RF Xmit Power 5
- (6) Slave Security 0
- (7) RTS to CTS 0
- (8) Retry Time Out 255
- (9) Lowpower Mode 0
- (A) High Noise 0

(B) MCU Speed 1

(C) RemoteLED 0

3. La configuración del Radio Modem SLAVE es la siguiente:

**(0) Set Operation Mode**

(1) Point to Point Slave

**(1) Set Baud Rate**

(0) 230,400

(A) Data, Parity 0

(B) MODBus RTU 0

(C) RS232/485 0

(D) Setup Port 3

(E) TurnOffDelay 0 TurnOnDelay 0

(F) FlowControl 0

**(2) Edit Call Book**

Entry Number Repeater1 Repeater2

(0) 242-2853 (Call number del otro Radio Modem)

Todas las demás entradas se encuentran con 000-0000

**(3) Edit Radio Transmission Characteristics**

(0) FreqKey 5

(1) Max Packet Size 9

- (2) Min Packet Size 0
- (3) Xmit Rate 1
- (4) RF Data Rate 2
- (5) RF Xmit Power 5
- (6) Slave Security 0
- (7) RTS to CTS 0
- (8) Retry Time Out 255
- (9) Lowpower Mode 0
- (A) High Noise 0
- (B) MCU Speed 1
- (C) RemoteLED 0

#### 4. Configuración del canal 1 (Ethernet) del PLC SLC 5/05

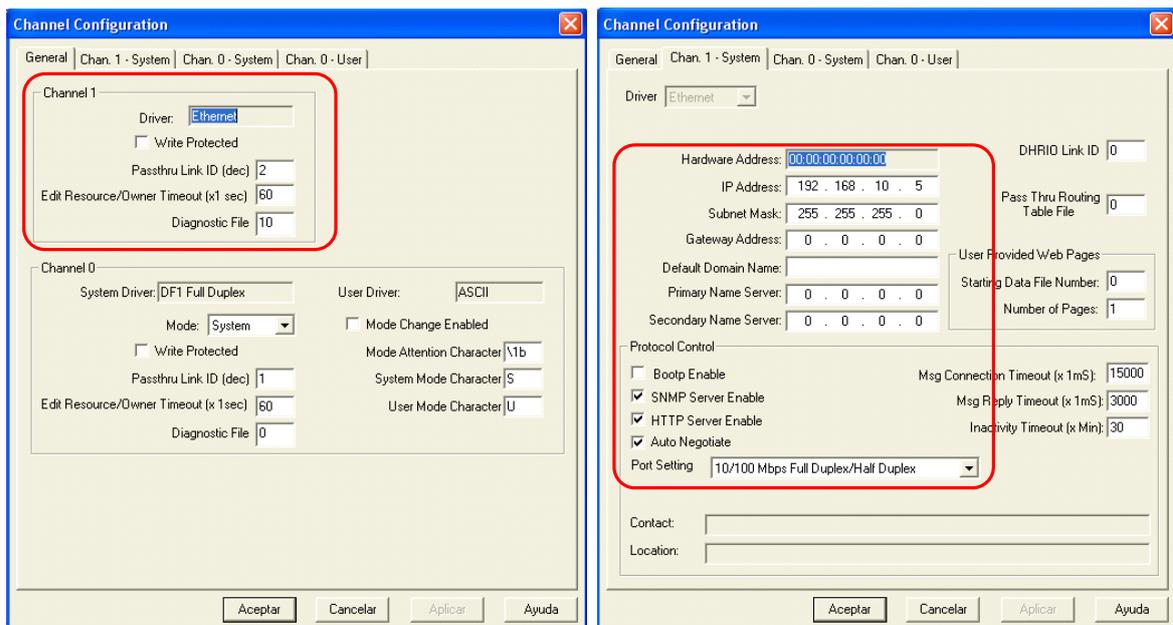


Figura. 5.169. Desarrollo Práctica #18 (2)

5. Debido a que se requiere un registro auxiliar para activar o desactivar las salidas desde la HMI, la programación Ladder que cumple con la lógica de control planteada en esta práctica es la siguiente:

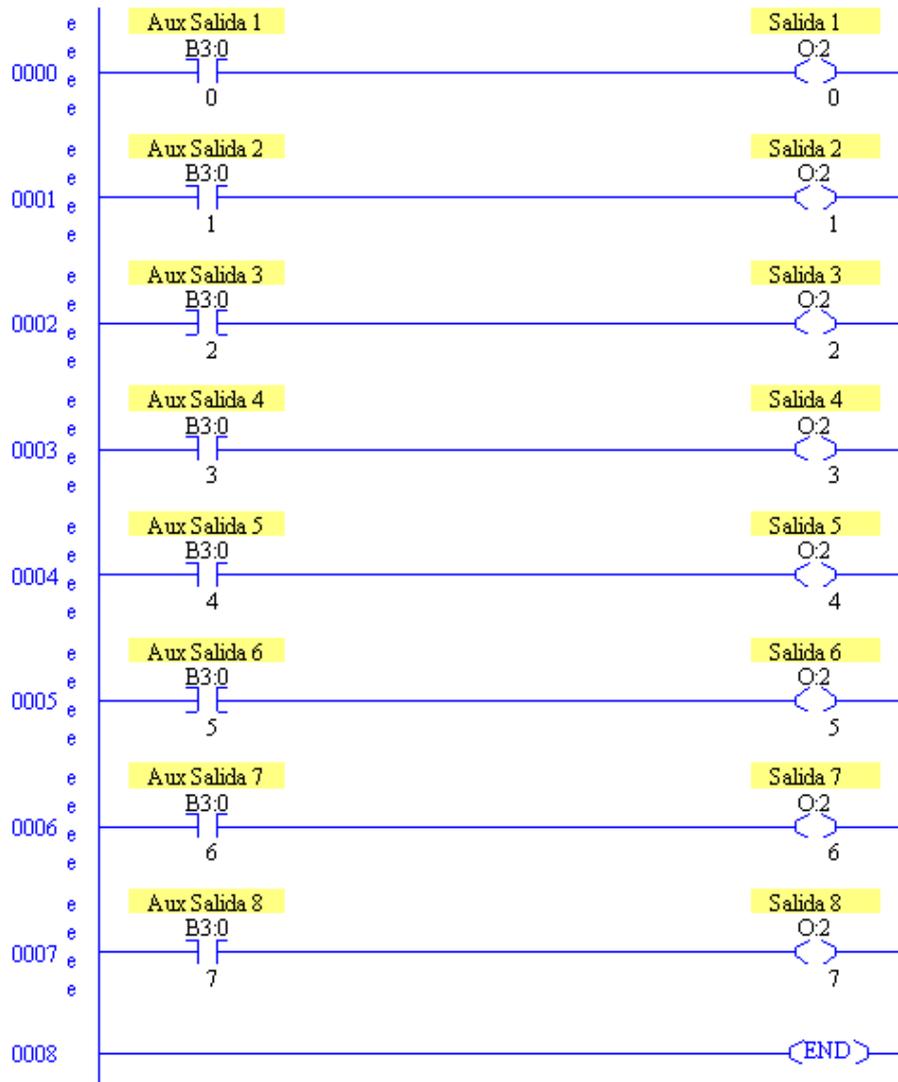


Figura. 5.170. Desarrollo Práctica #18 (3)

6. Configuración del canal 0 (DF1 Full Duplex) del PLC SLC 5/03

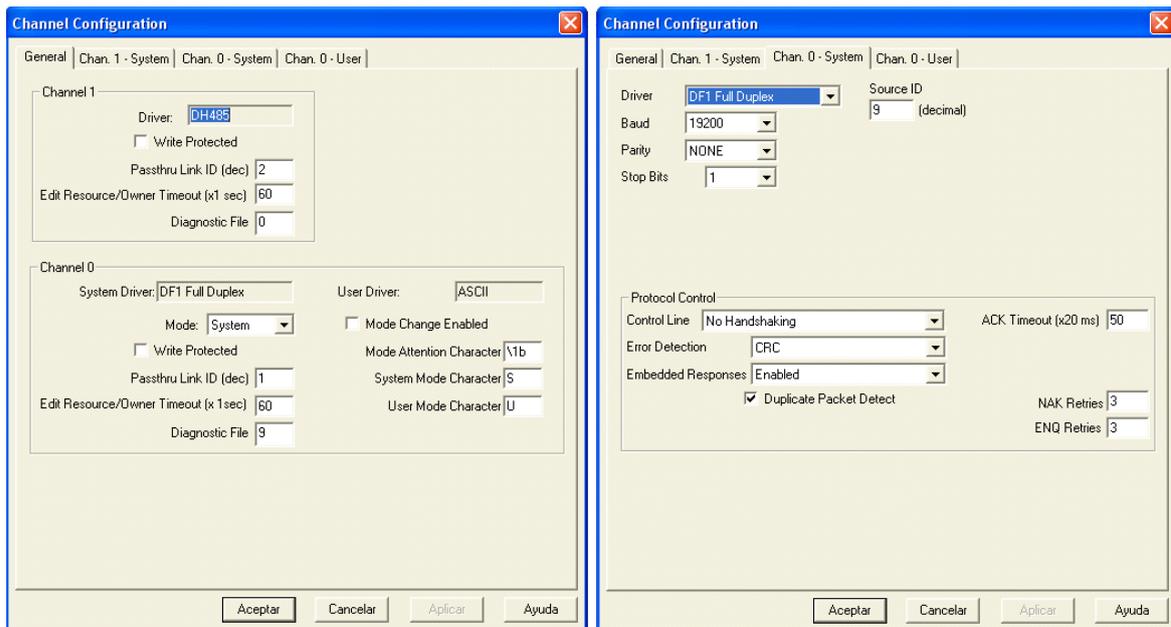


Figura. 5.171. Desarrollo Práctica #18 (4)

7. La programación Ladder que cumple con la lógica de control planteada en esta práctica es la siguiente:

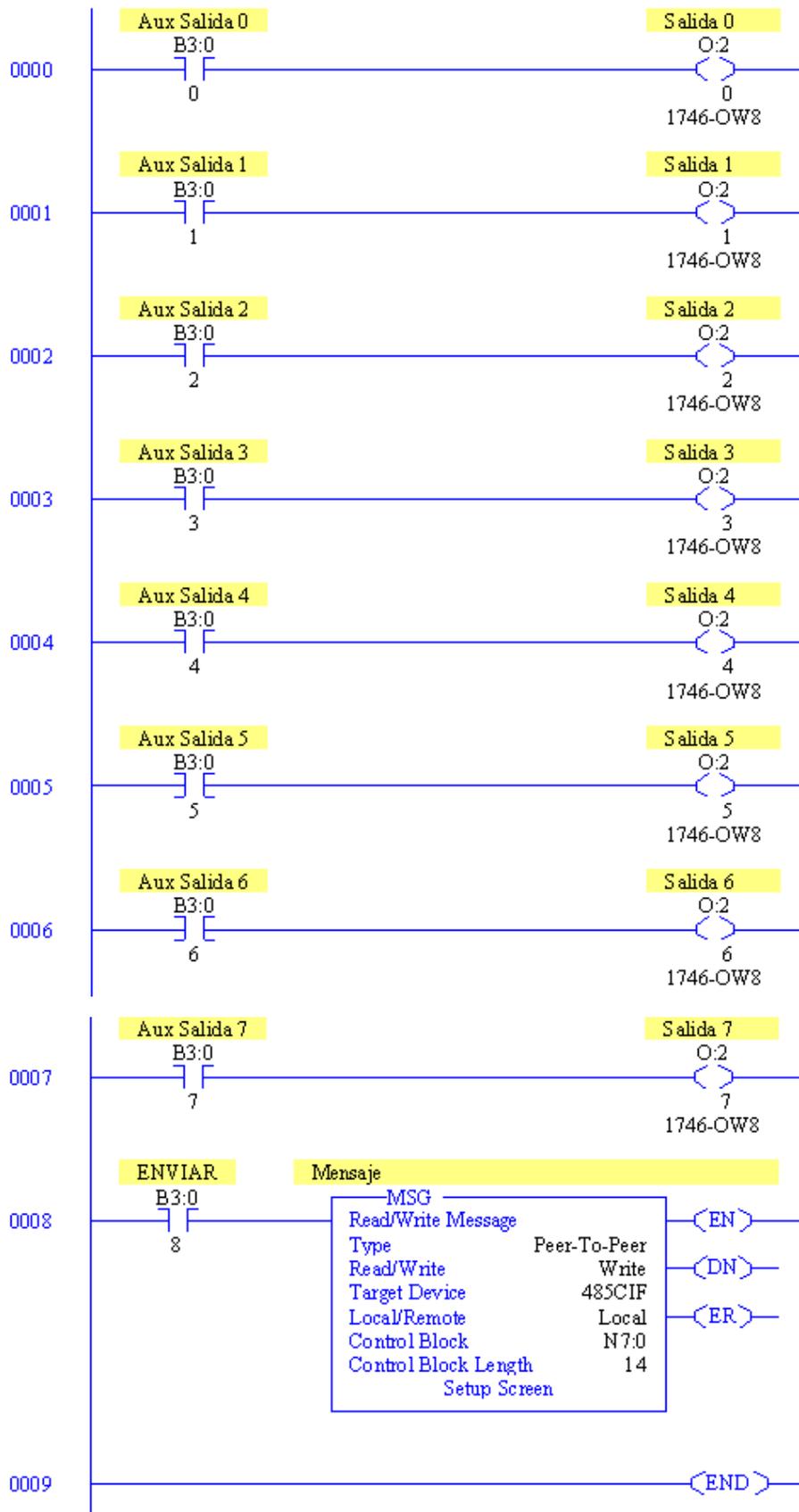


Figura. 5.172. Desarrollo Práctica #18 (5)

En donde la instrucción MSG se encuentra configurada de la siguiente manera:

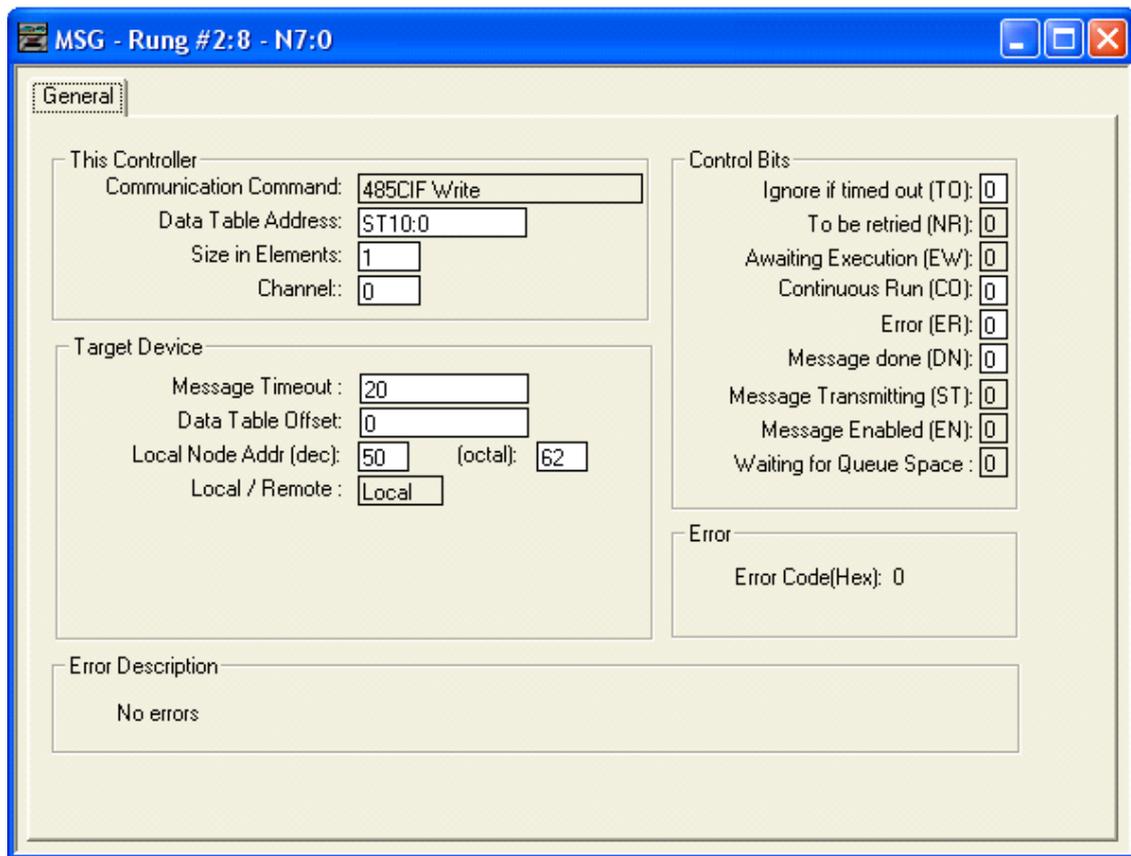


Figura. 5.173. Desarrollo Práctica #18 (6)

En donde la dirección que se encuentra en *Data Table Address* “ST10:0” contiene el texto del mensaje que se va a enviar, como se muestra en la figura 5.174.

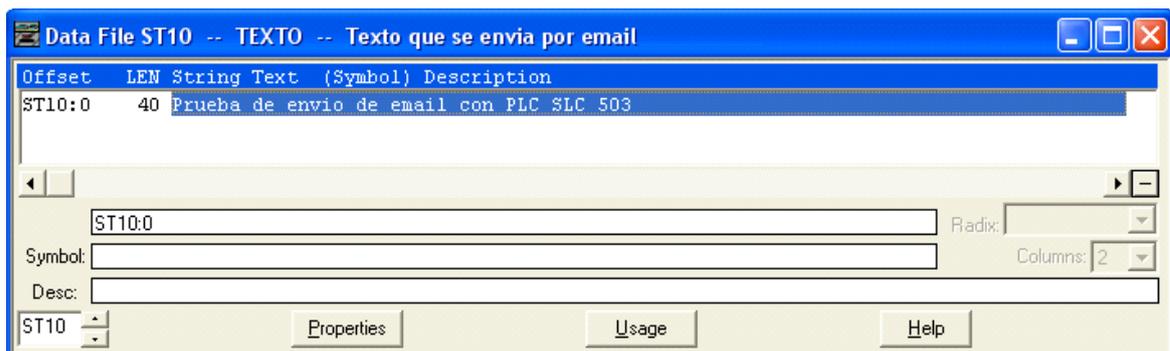


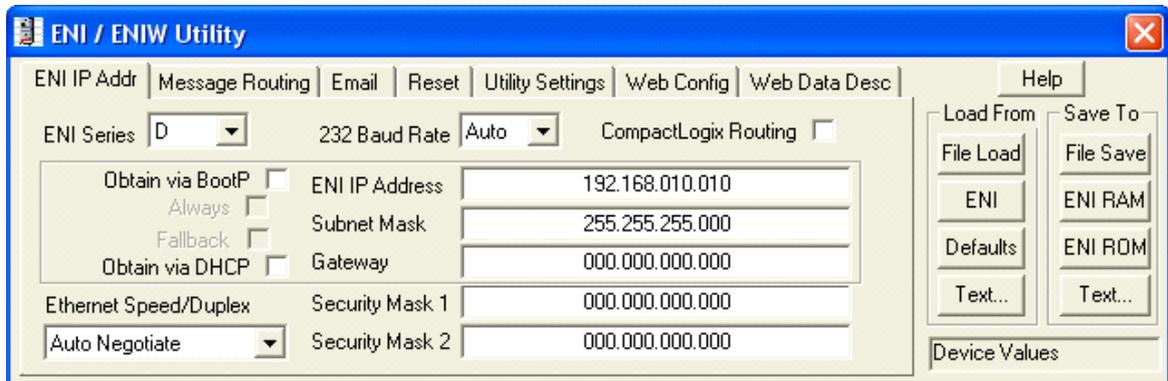
Figura. 5.174. Desarrollo Práctica #18 (7)

En *Local Node Addr (dec)* se ingresó el número de nodo (50) en el cual se encuentra configurada la dirección de e-mail en el módulo NET-ENIW. Ver figura 5.176.

## 8. Configuración del módulo de interfase ethernet NET-ENIW

Para configurar el módulo 1761-NET-ENIW se empleó el software “ENI/ENIW Utility”.

Como se muestra en la figura 5.175. la dirección IP asignada a este módulo es la 192.168.10.10



**Figura. 5.175. Desarrollo Práctica #18 (8)**

En la figura 5.176. Se muestra como se configuró la dirección IP del servidor de correo de la ESPE. (La dirección 192.188.58.210 es la IP asignada al servidor de correo de la ESPE al momento de realizar la práctica, la misma que puede estar sujeta a cambios por lo que deberá ser verificada al momento de realizar esta práctica). También se muestra la dirección de correo a la cual se enviará el e-mail (esta podrá ser cambiada posteriormente a través de la HMI).



**Figura. 5.176. Desarrollo Práctica #18 (9)**

La figura 5.177. Muestra los parámetros de comunicación que se emplearon para grabar la configuración en la memoria del módulo.

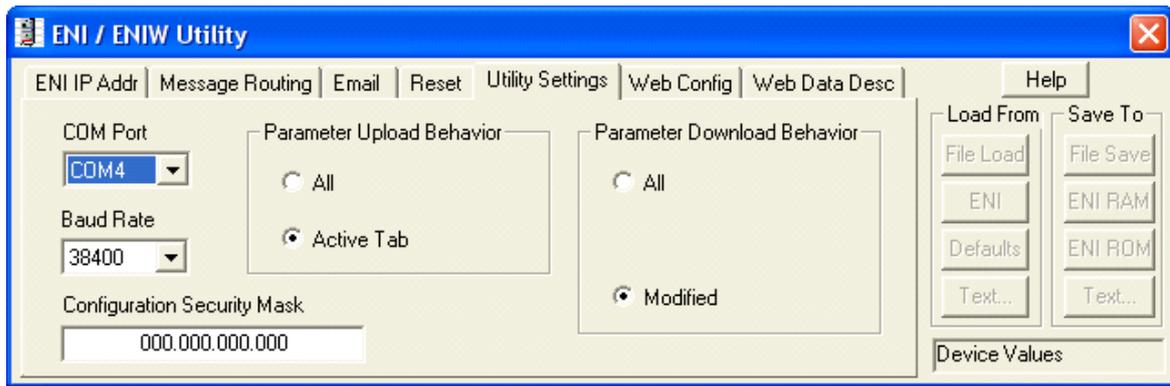


Figura. 5.177. Desarrollo Práctica #18 (10)

Finalmente se grabó la configuración en la memoria del módulo.

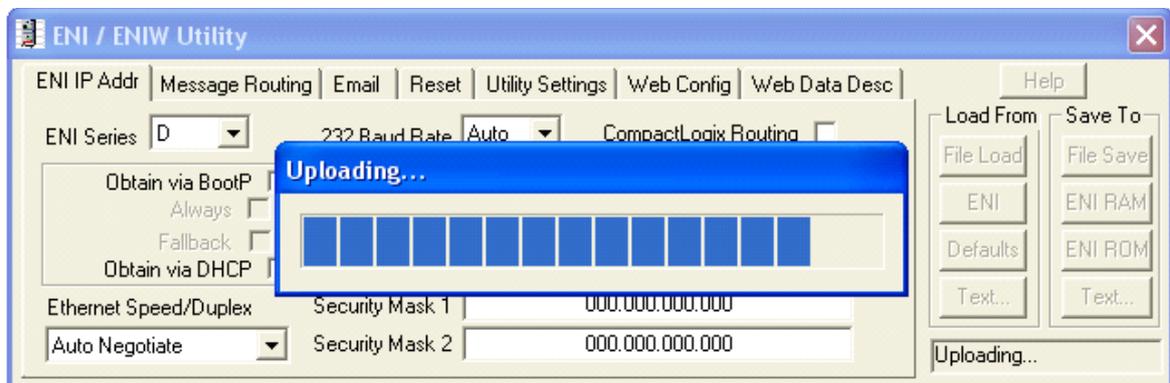
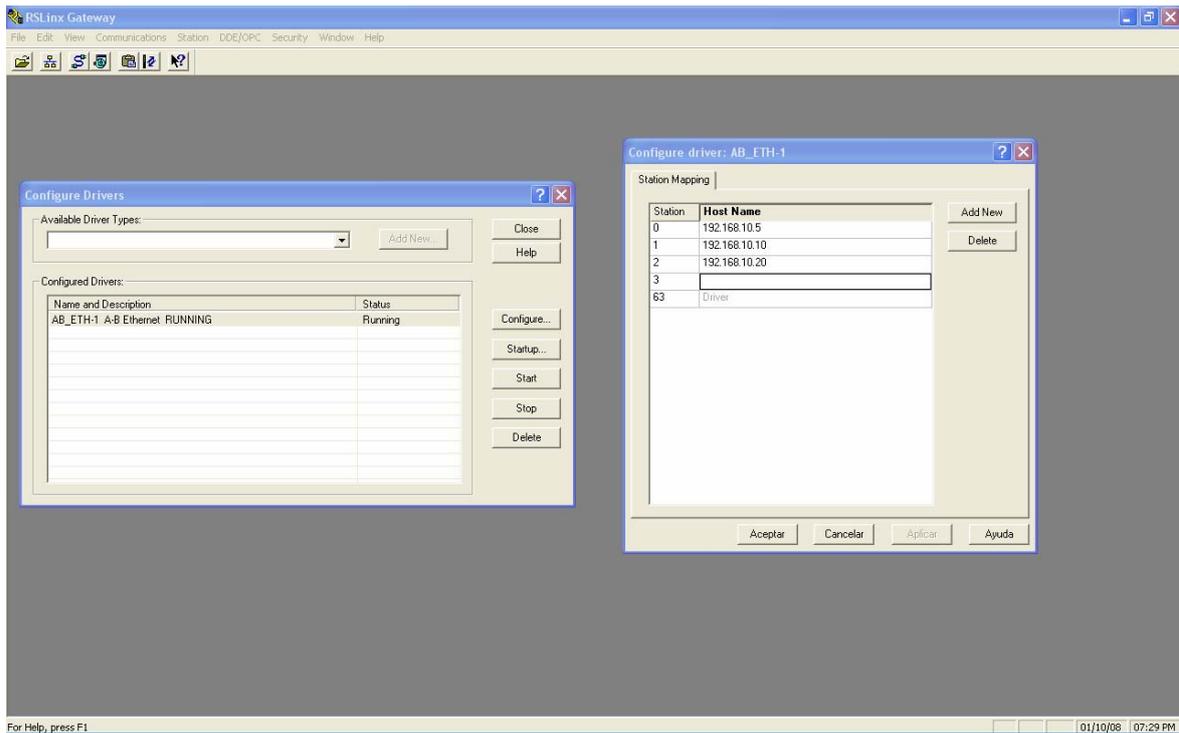


Figura. 5.178. Desarrollo Práctica #18 (11)



*Para grabar la configuración en el módulo, se recomienda presionar el botón **ENI ROM** en lugar del botón **ENI RAM**, puesto que esto garantiza que se grave toda la información en dicho módulo.*

## 9. Configuración del Driver de comunicación Ethernet en RSLinx



**Figura. 5.179. Desarrollo Práctica #18 (12)**

Como se puede ver en la figura 5.179., en el driver se han configurado las siguientes direcciones IP:

Dirección IP	Equipo al que se asignará...
192.168.10.5	PLC SLC 5/05
192.168.10.10	Módulo ENIW conectado al PLC SLC 5/03
192.168.10.20	PC con RSLinx, RSView32 y RSLogix500

**Tabla. 5.30. IP asignadas Práctica #18**



*Las Direcciones asignadas a cada uno de los equipos deberán constar en la red ethernet de la ESPE.*

## 10. Configuración de la dirección IP de la PC

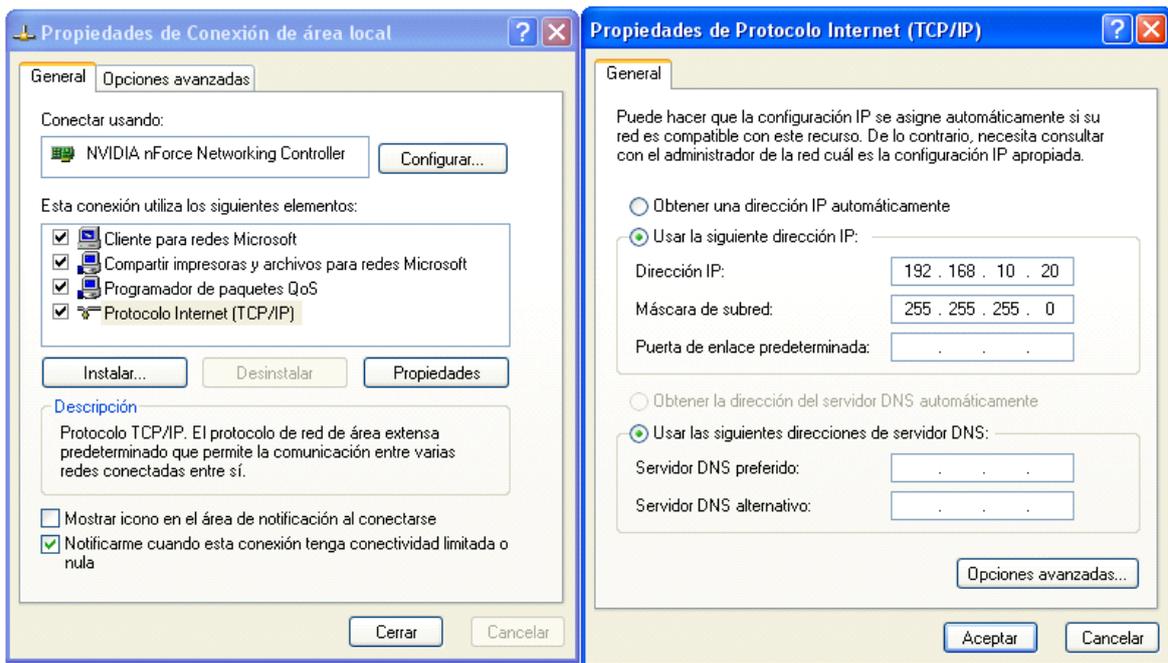


Figura. 5.180. Desarrollo Práctica #18 (13)

## 11. Verificación del funcionamiento del driver Ethernet

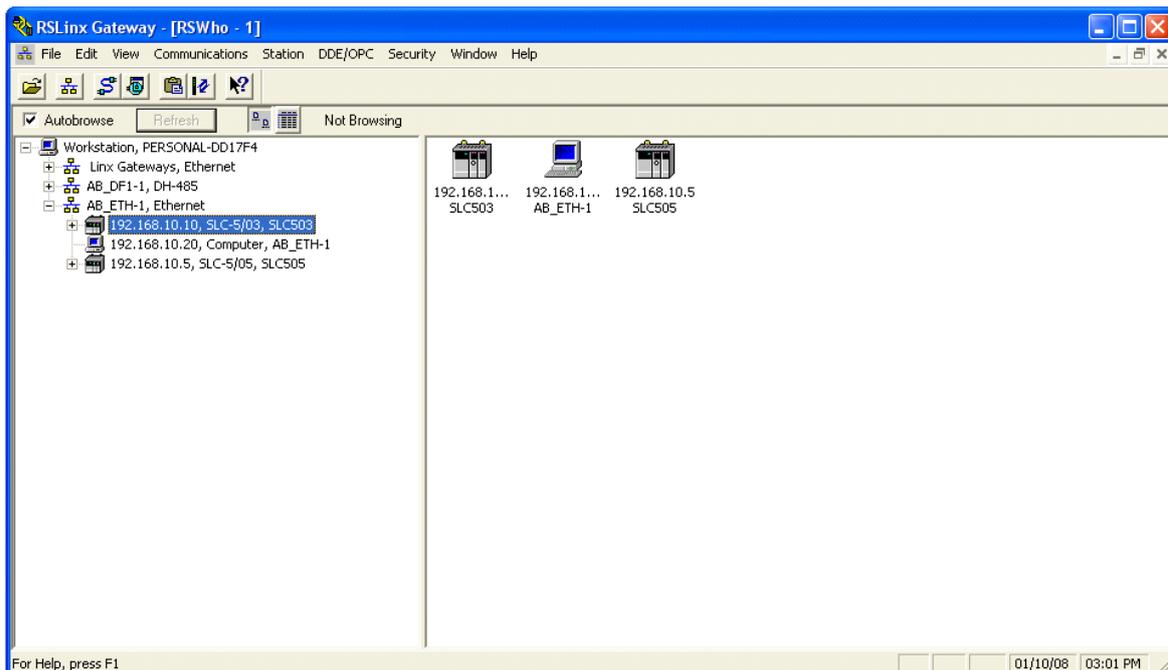


Figura. 5.181. Desarrollo Práctica #18 (14)

Como se puede ver los Radio Modems y el Módulo NET-ENIW son transparentes en la comunicación con los PLC's.

12. Abrir el software RSVIEW32, configurar el canal de comunicación, configurar el nodo de comunicación.

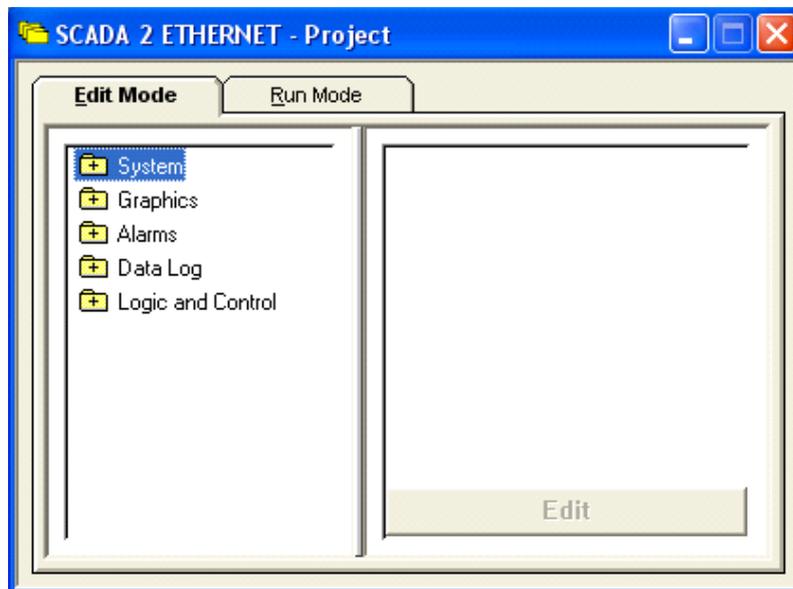


Figura. 5.182. Desarrollo Práctica #18 (15)

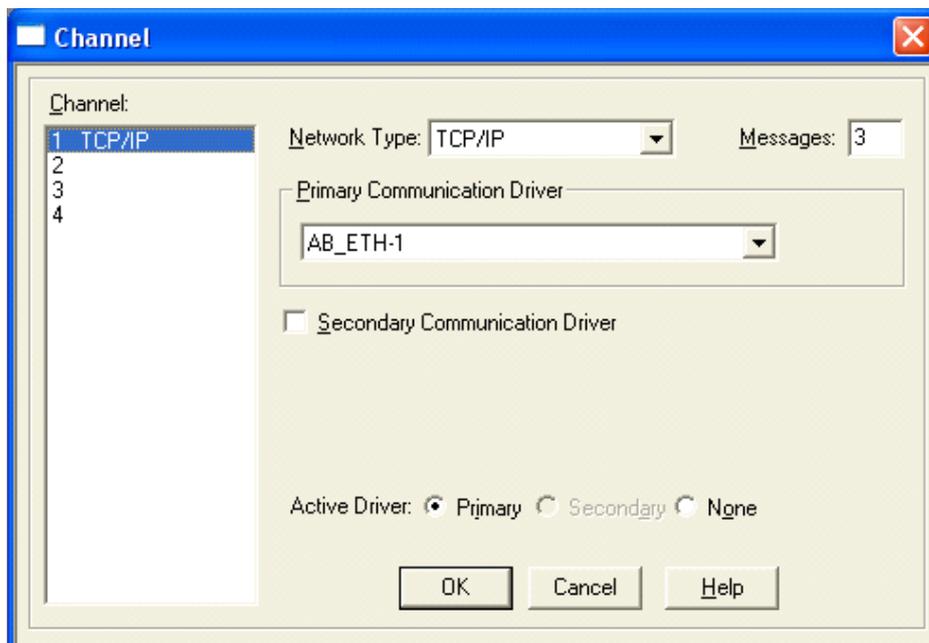


Figura. 5.183. Desarrollo Práctica #18 (16)

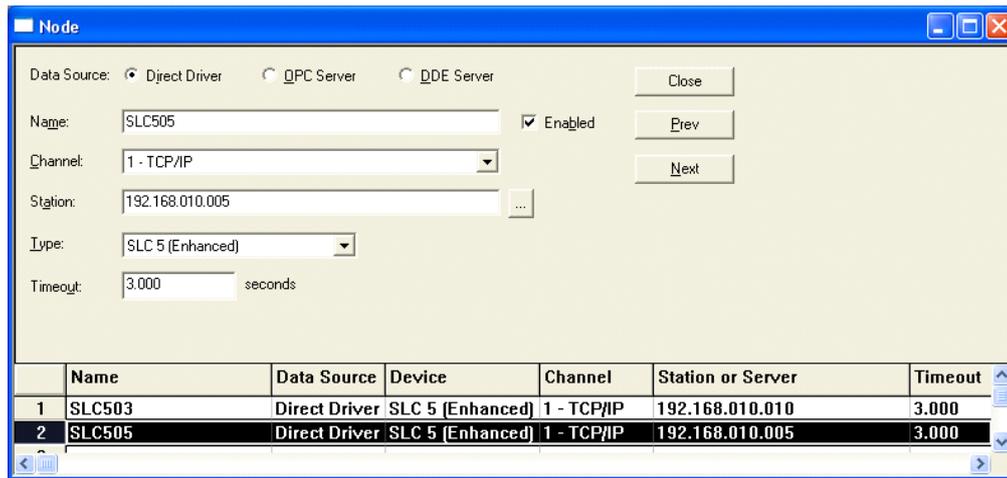


Figura. 5.184. Desarrollo Práctica #18 (17)

- Creación de la interfase HMI que interactúe con los PLC's de cada estación de control.

## INTERFASE HUMANO MÁQUINA (HMI)

La HMI creada en base al esquema planteado consta de cuatro pantallas:

### Pantalla Principal

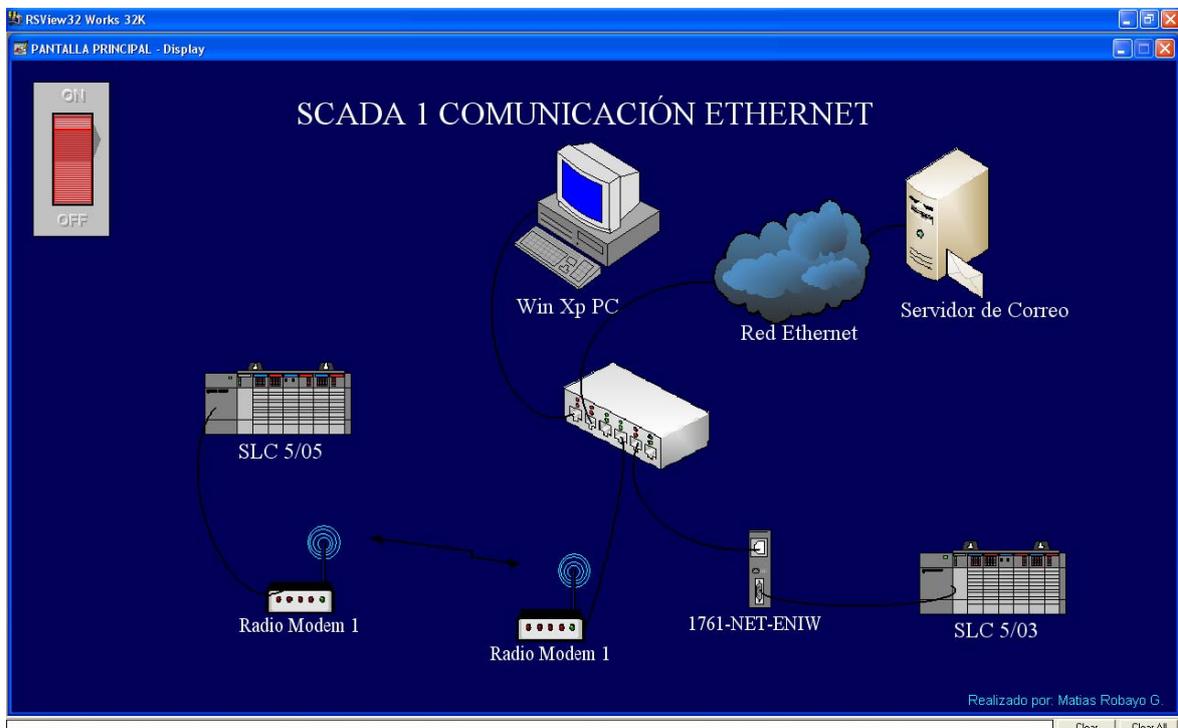
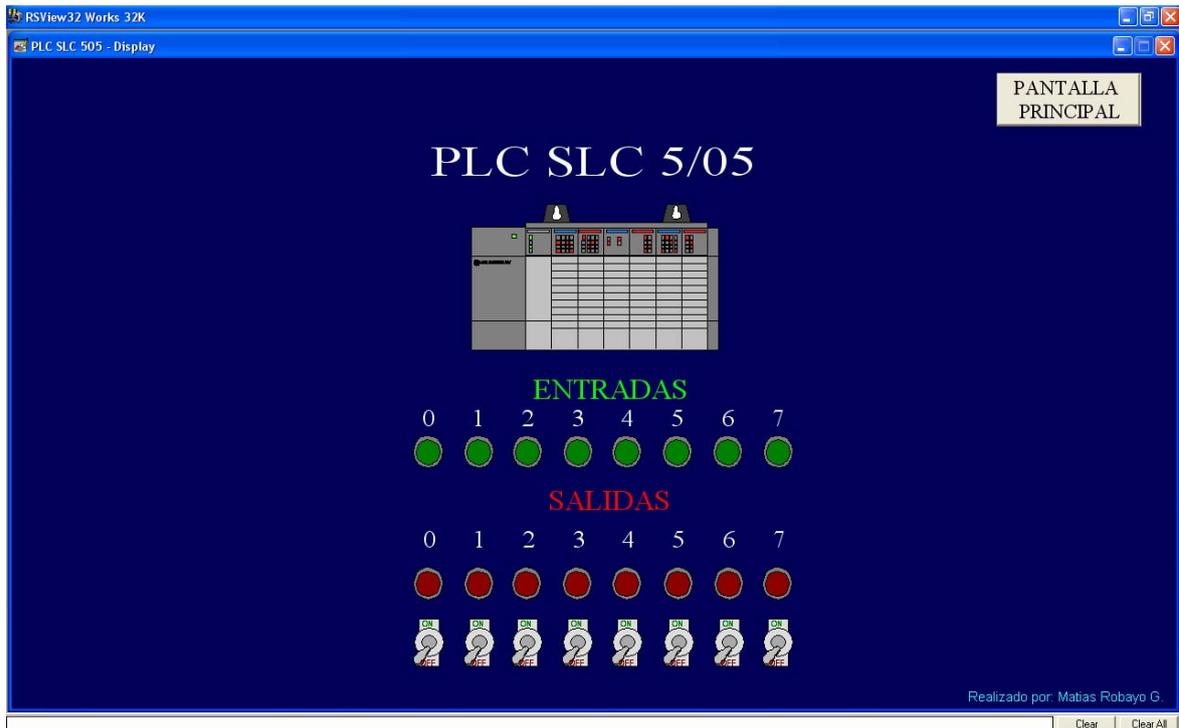


Figura. 5.185. Desarrollo Práctica #18 (18)

**PLC SLC 5/05**



**Figura. 5.186. Desarrollo Práctica #18 (19)**

**PLC SLC 5/03**



**Figura. 5.187. Desarrollo Práctica #18 (20)**

## TAGS

En la siguiente tabla se muestran los tags empleados en la programación de la HMI y sus características:

NOMBRE	TIPO	CARACTERÍSTICAS					
ENCENDER	Digital	Alarmed: No	Security: *	Memory Tag	Initial Value: Off		
		Off Label: Off	On Label: On				
ENVIAR	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC503	Address: B3:0/8	Scan Class: A
		Off Label: Off	On Label: On				
SLC503_IN0	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC503	Address: I:1/0	Scan Class: A
		Off Label: Off	On Label: On				
SLC503_IN1	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC503	Address: I:1/1	Scan Class: A
		Off Label: Off	On Label: On				
SLC503_IN2	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC503	Address: I:1/2	Scan Class: A
		Off Label: Off	On Label: On				
SLC503_IN3	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC503	Address: I:1/3	Scan Class: A
		Off Label: Off	On Label: On				
SLC503_IN4	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC503	Address: I:1/4	Scan Class: A
		Off Label: Off	On Label: On				
SLC503_IN5	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC503	Address: I:1/5	Scan Class: A
		Off Label: Off	On Label: On				
SLC503_IN6	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC503	Address: I:1/6	Scan Class: A
		Off Label: Off	On Label: On				
SLC503_IN7	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC503	Address: I:1/7	Scan Class: A
		Off Label: Off	On Label: On				
SLC503_OUT0	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC503	Address: B3:0/0	Scan Class: A
		Off Label:	On Label:				

		Off	On				
SLC503_OUT1	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC503	Address: B3:0/1	Scan Class: A
		Off Label: Off	On Label: On				
SLC503_OUT2	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC503	Address: B3:0/2	Scan Class: A
		Off Label: Off	On Label: On				
SLC503_OUT3	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC503	Address: B3:0/3	Scan Class: A
		Off Label: Off	On Label: On				
SLC503_OUT4	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC503	Address: B3:0/4	Scan Class: A
		Off Label: Off	On Label: On				
SLC503_OUT5	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC503	Address: B3:0/5	Scan Class: A
		Off Label: Off	On Label: On				
SLC503_OUT6	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC503	Address: B3:0/6	Scan Class: A
		Off Label: Off	On Label: On				
SLC503_OUT7	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC503	Address: B3:0/7	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_IN0	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: I:1/0	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_IN1	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: I:1/1	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_IN2	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: I:1/2	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_IN3	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: I:1/3	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_IN4	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: I:1/4	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_IN5	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: I:1/5	Scan Class: A

		Off Label: Off	On Label: On				
SLC505_IN6	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: I:1/6	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_IN7	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: I:1/7	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_OUT0	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: B3:0/0	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_OUT1	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: B3:0/1	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_OUT2	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: B3:0/2	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_OUT3	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: B3:0/3	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_OUT4	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: B3:0/4	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_OUT5	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: B3:0/5	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_OUT6	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: B3:0/6	Scan Class: A
		Off Label: Off	On Label: On				
SLC505_OUT7	Digital	Alarmed: No	Security: *	Device Tag	Node Name: SLC505	Address: B3:0/7	Scan Class: A
		Off Label: Off	On Label: On				
TEXTO	String	Alarmed: No	Security: *	Device Tag	Node Name: SLC503	Address: ST10:0	Scan Class: A
		Length : 40					

**Tabla. 5.31. Tags utilizados Práctica #18**

14. Finalmente, se realizaron las pruebas necesarias para verificar que la HMI y los PLC's funcionan correctamente y que se encuentran correctamente

configurados. El envío se de e-mail se verificó como se puede ver en la figura 5.188.

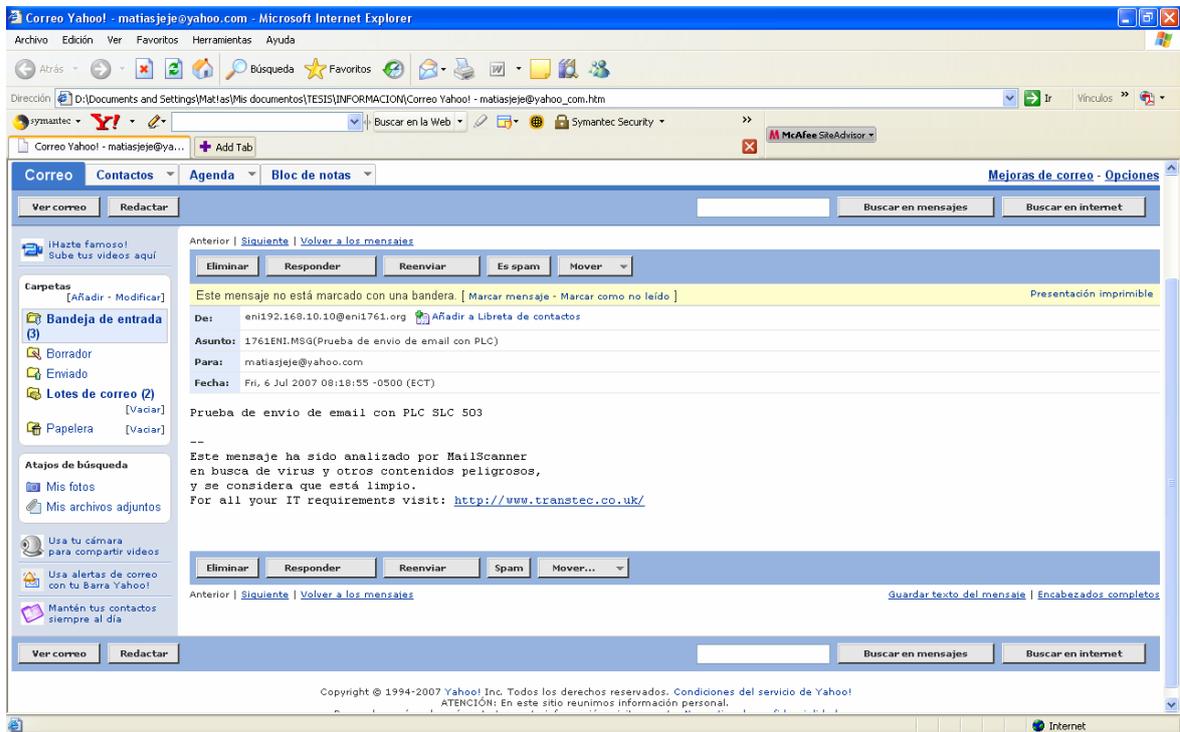


Figura. 5.188. Desarrollo Práctica #18 (21)

A continuación se presentan las pantallas de acceso al PLC SLC 5/05 y al módulo NET-ENIW, que se obtienen ingresando la dirección IP de cada uno en el Internet Explorer.

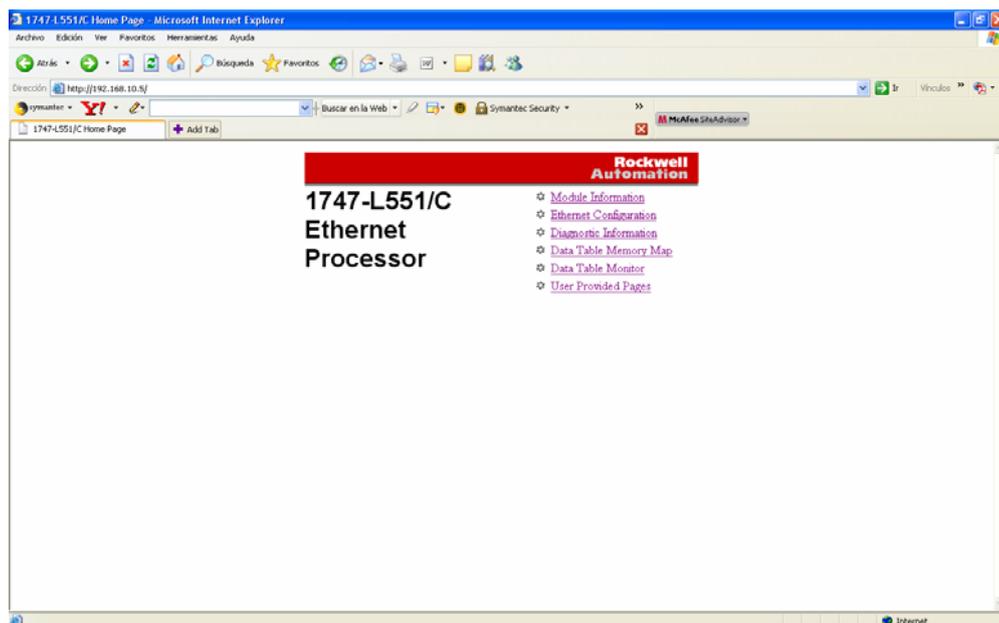


Figura. 5.189. Desarrollo Práctica #18 (22)

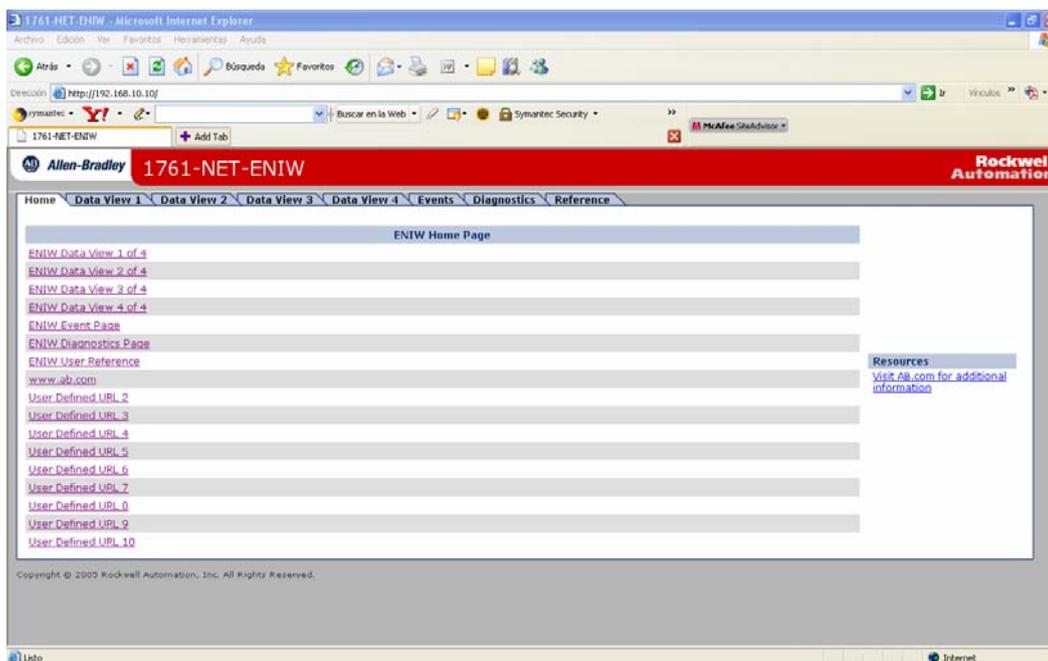


Figura. 5.190. Desarrollo Práctica #18 (23)

### 5.18.5. PRUEBAS Y RESULTADOS

Durante el desarrollo de la práctica se realizaron las pruebas respectivas tanto del software (programación de los PLC y de la HMI) como del hardware (Conexión de los PLC's, módulo ENIW y Radio Modems), a través de las cuales se verificó que todos los dispositivos trabajan correctamente de acuerdo a la lógica y configuración del problema planteado en esta práctica.

### 5.18.6. CONCLUSIONES Y RECOMENDACIONES

#### Conclusiones

- Los PLC's SLC 5/05 al poseer un canal de comunicación ethernet incorporado, simplifican la configuración del PLC en sistemas que requieran conexión a este tipo de redes.
- El módulo 1761-NET-ENIW brinda total conectividad ethernet a los PLC's SLC 5/03, siendo este módulo totalmente "Transparente" para la comunicación.
- Los radio modem Ethernet permiten ampliar significativamente (hasta 16 Km.) la distancia máxima de comunicación de una red ethernet.

- El módulo 1761-NET-ENIW permite enviar mensajes cortos vía e-mail a través de la programación de instrucción MSG en el PLC al cual se encuentra conectado.
- Tanto el PLC SLC 5/05 como el módulo 1761-NET-ENIW poseen la funcionalidad de servidor Web, lo que permite monitorearlos a través del explorador de Internet

### **Recomendaciones**

- Se recomienda verificar que las Direcciones IP asignadas a cada uno de los dispositivos sean validas en la red ethernet de la ESPE, puesto que dicha red puede estar sujeta a cambios.
- Se recomienda NO cambiar los parámetros “Radio Transmission Characteristics” de los radio módems a no ser que se tenga pleno conocimiento de lo que representan.
- Se recomienda para esta práctica, no configurar el canal 0 de los PLC SLC5/03 en modo “DF1 Radio Modem” puesto que este protocolo no es compatible con los protocolos DF1 Half y Full Duplex que son los que permiten comunicarse a la PC con el PLC.

## CAPITULO 6

### PRUEBAS Y RESULTADOS

#### 6.1. CONTROL LÓGICO 1: INSTRUCCIONES DE BITS 1

Para verificar que el programa ladder realizado en el desarrollo de esta práctica cumple satisfactoriamente con el planteamiento de la misma, se realizaron pruebas tanto en software (Modo ONLINE del PLC en RSLogix 500) como en hardware (Tablero didáctico de simulación AB-01) en las que se verificó que la secuencia de encendido y apagado de los tres motores cumple con la secuencia planteada en la práctica.

#### 6.2. CONTROL LÓGICO 2: INSTRUCCIONES DE BITS 2

Para verificar que el programa ladder realizado en el desarrollo de esta práctica cumple satisfactoriamente con el planteamiento de la misma, se realizaron pruebas tanto en software (Modo ONLINE del PLC en RSLogix 500) como en hardware (Tablero didáctico de simulación AB-02) en las que se verificó que el encendido y apagado de los motores M1 (llenado) y M2 (vaciado) se realiza correctamente de acuerdo a la secuencia planteada en la práctica, la misma que depende de los sensores S1, S2 y S3 (en modo Automático).

#### 6.3. CONTROL LÓGICO 3: CONTADORES

Para verificar que el programa ladder realizado en el desarrollo de esta práctica cumple satisfactoriamente con el planteamiento de la misma, se realizaron pruebas tanto en software (Modo ONLINE del PLC en RSLogix 500) como en hardware (Tablero didáctico de simulación AB-03) en las que se verificó que:

- La señal RECHAZAR se activa únicamente cuando el sensor S1 se activa, y se desactiva cuando se enciende la señal RECHAZADO
- La señal ELEVADOR se activa cuando han sido aprobados 12 productos, y se desactiva cuando se enciende la señal FIN ELEVADOR
- La señal CARGAR EN CAJA se activa cuando se activa el sensor S2

- La señal ALARMA se activa si el número de productos defectuosos es mayor a 20 y se desactiva únicamente si se presiona el botón de reconocimiento AKW ALARMA.
- Los contadores se enceran cuando se presiona el botón de encerar.

Con lo que se garantizó que el sistema se encuentra trabajando correctamente de acuerdo a la lógica planteada.

#### **6.4. CONTROL LÓGICO 4: TEMPORIZADORES 1**

Para verificar que el programa ladder realizado en el desarrollo de esta práctica cumple satisfactoriamente con el planteamiento de la misma, se realizaron pruebas tanto en software (Modo ONLINE del PLC en RSLogix 500) como en hardware (Tablero didáctico de simulación AB-04) en las que se verificó que las señales de alarma se encienden de acuerdo al sensor que se active. Adicionalmente se verificó que las señales de alarma titilan sincrónicamente con una frecuencia de 1HZ hasta que se presione el botón respectivo de reconocimiento.

#### **6.5. CONTROL LÓGICO 5: TEMPORIZADORES 2**

Para verificar que el programa ladder realizado en el desarrollo de esta práctica cumple satisfactoriamente con el planteamiento de la misma, se realizaron pruebas tanto en software (Modo ONLINE del PLC en RSLogix 500) como en hardware (Tablero didáctico de simulación AB-05) en las que se verificó que las señales de ABRIR PUERTA y CERRAR PUERTA se encienden de acuerdo a la lógica planteada en la práctica, la misma que depende de los sensores S1, S2 y S3.

#### **6.6. CONTROL LÓGICO 6: ENTRADAS Y SALIDAS ANALÓGICAS**

Para verificar que el programa ladder realizado en el desarrollo de esta práctica cumple satisfactoriamente con el planteamiento de la misma, se realizaron pruebas tanto en software (Modo ONLINE del PLC en RSLogix 500) como en hardware (Tablero didáctico de simulación AB-06) en las que se verificó que la activación y desactivación de las válvulas de V1 y V2 se realiza de acuerdo a la lógica planteada en la práctica, la

misma que depende del nivel de combustible que se lee a través de la entrada analógica.

### **6.7. CONTROL LÓGICO 7: REGISTROS DE RELOJ/CALENDARIO**

Para verificar que el programa ladder realizado en el desarrollo de esta práctica cumple satisfactoriamente con el planteamiento de la misma, se realizaron pruebas tanto en software (Modo ONLINE del PLC en RSLogix 500) como en hardware (Tablero didáctico de simulación AB-07) en las que se verificó que la activación y desactivación de las luces se realiza de acuerdo a los horarios planteados en la práctica.

### **6.8. CONTROL LÓGICO 8: TRANSFERENCIA DE DATOS**

Para verificar que el programa ladder realizado en el desarrollo de esta práctica cumple satisfactoriamente con el planteamiento de la misma, se realizaron pruebas tanto en software (Modo ONLINE del PLC en RSLogix 500) como en hardware (Tablero didáctico de simulación AB-08) en las que se verificó que la tomas de muestras (valores de temperatura) se la realiza en intervalos de 10 segundo y se los almacena en una pila de registros.

### **6.9. CONTROL LÓGICO 9: FLUJO DE PROGRAMA**

Para verificar que el programa ladder realizado en el desarrollo de esta práctica cumple satisfactoriamente con el planteamiento de la misma, se realizaron pruebas tanto en software (Modo ONLINE del PLC en RSLogix 500) como en hardware (Tablero didáctico de simulación AB-09) en las que se verificó que la secuencia de encendido y apagado de los tres motores cumple con la secuencia planteada en la práctica, utilizando únicamente un botón para el apagado y encendido de cada motor.

### **6.10. CONTROL DE PROCESOS 1: CONTROL ON-OFF CON HISTÉRESIS**

Para verificar que el programa ladder y la HMI realizados en el desarrollo de esta práctica cumplen satisfactoriamente con el planteamiento de la misma, se realizaron pruebas tanto del software (HMI en RSView32) como del hardware (PLC y Módulo Degem de simulación PCT-2 “AIR FLOW TEMPERATURE CONTROL SYSTEM”)

en las que se verificó que el controlador trabaja correctamente de acuerdo a las especificaciones planteadas en la práctica, y que la HMI permite al operador interactuar fácilmente con el controlador y con el proceso.

### **6.11. CONTROL DE PROCESOS 2: CONTROL PID DE TEMPERATURA**

Para verificar que el programa ladder y la HMI realizados en el desarrollo de esta práctica cumplen satisfactoriamente con el planteamiento de la misma, se realizaron pruebas tanto del software (HMI en RSView32) como del hardware (PLC y Módulo Degem de simulación PCT-2 “AIR FLOW TEMPERATURE CONTROL SYSTEM”) en las que se verificó que el controlador trabaja correctamente de acuerdo a las especificaciones planteadas en la práctica, y que la HMI permite al operador interactuar fácilmente con el controlador y con el proceso.

### **6.12. CONTROL DE PROCESOS 3: CONTROL PID DE VELOCIDAD**

Para verificar que el programa ladder y la HMI realizados en el desarrollo de esta práctica cumplen satisfactoriamente con el planteamiento de la misma, se realizaron pruebas tanto del software (HMI en RSView32) como del hardware (PLC y Módulo Degem de simulación PCT-1 “DC SPEED CONTROL SYSTEM”) en las que se verificó que el controlador trabaja correctamente de acuerdo a las especificaciones planteadas en la práctica, y que la HMI permite al operador interactuar fácilmente con el controlador y con el proceso.

### **6.13. CONTROL DE PROCESOS 4: CONTROL DE RAZÓN**

Para verificar que el programa ladder y la HMI realizados en el desarrollo de esta práctica cumplen satisfactoriamente con el planteamiento de la misma, se realizaron pruebas tanto del software (HMI en RSView32) como del hardware (PLC y Módulos Degem de simulación PCT-1 “DC SPEED CONTROL SYSTEM”) en las que se verificó que el controlador trabaja correctamente de acuerdo a las especificaciones planteadas en la práctica, y que la HMI permite al operador interactuar fácilmente con el controlador y con el proceso.

#### **6.14. CONTROL DE PROCESOS 5: CONTROL POR SOBREPOSICIÓN**

Para verificar que el programa ladder y la HMI realizados en el desarrollo de esta práctica cumplen satisfactoriamente con el planteamiento de la misma, se realizaron pruebas tanto del software (HMI en RSView32) como del hardware (PLC y Módulos Degem de simulación PCT-1 “DC SPEED CONTROL SYSTEM” y PCT-2 “AIR FLOW TEMPERATURE CONTROL SYSTEM”) en las que se verificó que el controlador trabaja correctamente de acuerdo a las especificaciones planteadas en la práctica, y que la HMI permite al operador interactuar fácilmente con el controlador y con el proceso.

#### **6.15. CONTROL DE PROCESOS 6: CONTROL EN CASCADA**

Para verificar que el programa ladder y la HMI realizados en el desarrollo de esta práctica cumplen satisfactoriamente con el planteamiento de la misma, se realizaron pruebas tanto del software (HMI en RSView32) como del hardware (PLC y Módulo Degem de simulación PCT-2 “AIR FLOW TEMPERATURE CONTROL SYSTEM”) en las que se verificó que el controlador trabaja correctamente de acuerdo a las especificaciones planteadas en la práctica, y que la HMI permite al operador interactuar fácilmente con el controlador y con el proceso.

#### **6.16. CONTROL DE PROCESOS 7: CONTROL SELECTIVO**

Para verificar que el programa ladder y la HMI realizados en el desarrollo de esta práctica cumplen satisfactoriamente con el planteamiento de la misma, se realizaron pruebas tanto del software (HMI en RSView32) como del hardware (PLC y Módulo Degem de simulación PCT-2 “AIR FLOW TEMPERATURE CONTROL SYSTEM”) en las que se verificó que el controlador trabaja correctamente de acuerdo a las especificaciones planteadas en la práctica, y que la HMI permite al operador interactuar fácilmente con el controlador y con el proceso.

#### **6.17. CONTROL DE PROCESOS 8: SCADA 1 COMUNICACIÓN SERIAL**

Para verificar que la HMI y la configuración de los PLCs y Radio Modems realizados en el desarrollo de esta práctica cumplen satisfactoriamente con el planteamiento de la

misma, se realizaron pruebas tanto del software (HMI en RSView32) como del hardware (PLC, Radio Modems y módulo 1761-NET-AIC+) en las que se verificó que:

- El PLC se encuentra conectado y trabaja correctamente de acuerdo a las especificaciones planteadas en la práctica.
- La HMI permite al operador interactuar fácilmente con el controlador y con el proceso.
- Los Radio Modems trabajan correctamente, siendo transparentes en la comunicación

#### **6.18. CONTROL DE PROCESOS 9: SCADA 2 COMUNICACIÓN ETHERNET**

Para verificar que la HMI y la configuración de los PLCs y Radio Modems realizados en el desarrollo de esta práctica cumplen satisfactoriamente con el planteamiento de la misma, se realizaron pruebas tanto del software (HMI en RSView32) como del hardware (PLCs, Radio Modems y módulo 1761-NET-ENIW) en las que se verificó que:

- Los PLCs se encuentran conectados y trabajan correctamente de acuerdo a las especificaciones planteadas en la práctica.
- El módulo 1761-NET-ENIW se encuentra configurado correctamente y permite conectar al PLC 5/03 a la red Ethernet.
- La HMI permite al operador interactuar fácilmente con el controlador y con el proceso.
- Los Radio Modems trabajan correctamente, siendo transparentes en la comunicación

## CAPITULO 7

### CONCLUSIONES Y RECOMENDACIONES

#### 7.1. CONCLUSIONES

- El diseño del sistema de entrenamiento modular con enfoque sistemático, didáctico y metodológico facilita el aprendizaje de los estudiantes acerca del manejo de PLC's Allen Bradley y Software Rockwell aplicados en el control lógico y de procesos.
- Las prestaciones de programación y configuración que ofrece el paquete de programación RSLogix500, maximizan el rendimiento, ahorran tiempo de desarrollo y mejoran la productividad de los proyectos que emplean procesadores SLC 500 y MicroLogix de Allen-Bradley.
- El diseño del entorno y la arquitectura integrada del software RSView32, reducen el tiempo de aprendizaje y capacitación del estudiante para el desarrollo de aplicaciones HMI cuando se utiliza controladores de Rockwell Automation.
- La sencilla instalación y configuración del módulo 1761-NET-ENIW facilita la conexión de PLC's SLC500 (SLC 5/04 o inferiores) y Micrologix 1000 a redes de datos Ethernet.
- El completo desarrollo sistemático y progresivo de las prácticas propuestas en este sistema de entrenamiento, garantiza al estudiante que se encuentra capacitado para el correcto manejo y operación de los PLC's SLC-500 y Micrologix aplicados en el control lógico y de procesos.
- La fácil conexión y la didáctica representación grafica que poseen los tableros de simulación, permiten al estudiante realizar simulaciones en hardware de todas las prácticas de control lógico, empleando mucho menor tiempo y mejorando sustancialmente su aprendizaje.

- La manera didáctica, sistemática y metodológica en que se plantean las prácticas permiten sintetizar y direccionar la información proporcionada por la casa fabricante para que de manera más sencilla pueda ser asimilada por los estudiantes de la carrera de Ingeniería en Electrónica Automatización y Control.
- El contenido tratado en este sistema de entrenamiento modular abarca completamente los temas incluidos en la planificación integral de estudio de la Escuela Politécnica del Ejército para el Aprendizaje de Control Lógico y de Procesos.

## **7.2. RECOMENDACIONES**

- Seguir en forma ordenada la planificación propuesta para el desarrollo de las prácticas.
- Verificar los diagramas de conexión dados por la casa fabricante para cada uno de los equipos, antes de realizar cualquier conexión evitando así cualquier daño por mala manipulación.
- Consultar los textos indicados en la bibliografía, si se requiere información adicional sobre algún tema específico.
- Procurar la adquisición de equipos adicionales que permitan complementar el sistema de entrenamiento con prácticas referentes principalmente al uso de redes DeviceNet y ControlNet
- Procurar igualmente la adquisición de nuevas plantas (procesos) que permitan verificar las técnicas de control planteadas en este proyecto pero bajo otras condiciones.
- A fin de abastecer las necesidades propias del Lab. De Robótica y PLCs se recomienda equipar las siete mesas del laboratorio con el sistema de entrenamiento prototipo generado en el presente proyecto.

## REFERENCIAS BIBLIOGRÁFICAS

1. Ortiz, Hugo, **Instrumentación y Sistemas de Control**, 1998.
2. Lipták, Béla, **Instrument Engineers Handbook**, 3ra Edición, Tunning PID Controllers, Pags 144-157.
3. Rockwell Automation, **SLC 500 Instruction Set**, 1746 1747-RM011E-EN-P, [www.ab.com](http://www.ab.com), Enero 2006.
4. Rockwell Automation, **SLC™ 500 4-Channel Analog I/O Modules**, 1746-UM005B-EN-P, [www.ab.com](http://www.ab.com), Junio 2004.
5. Rockwell Automation, **SLC™ 500 4-Channel Analog I/O Modules Installation Instructions**, 1746-IN008C-EN-P, [www.ab.com](http://www.ab.com), Mayo 2004.
6. Rockwell Automation, **Quick Start for Experienced Users**, 1747-10.4, [www.ab.com](http://www.ab.com), Julio 1998.
7. Rockwell Automation, **Módulos de E/S analógicas**, 1746-6.4ES, [www.ab.com](http://www.ab.com), Enero 1996.
8. Rockwell Automation, **SLC 5/03, SLC 5/04, and SLC 5/05 Modular Processors**, 1747-IN009E-MU-P, [www.ab.com](http://www.ab.com), Abril 1996.
9. Rockwell Automation, **Digital I/O Modules**, 1746-IN027B-EN-P, [www.ab.com](http://www.ab.com), Enero 2007.
10. Rockwell Automation, **MicroLogix 1000 Programmable Controllers User Manual**, 1761-6.3, [www.ab.com](http://www.ab.com), Julio 1998.
11. Rockwell Automation, **MicroLogix 1000 Programmable Controllers Installation Instructions**, 1761-IN001B-MU-P, [www.ab.com](http://www.ab.com), Mayo 2002.
12. Rockwell Automation, **MicroLogix™ Ethernet Interface**, 1761-UM006E-EN-P, [www.ab.com](http://www.ab.com), Agosto 2006.
13. Rockwell Automation, **MicroLogix™ Ethernet Interface**, 1761-IN007B-MU-P, [www.ab.com](http://www.ab.com), Marzo 2004.
14. Rockwell Automation, **AIC+ Advanced Interface Converter**, 1761-UM004B-EN-P, [www.ab.com](http://www.ab.com), Junio 2006.
15. Rockwell Automation, **RSView32 User's Guide**, 1761 VW32-UM001B-EN-E, [www.ab.com](http://www.ab.com), Mayo 2005.
16. Rockwell Automation, **Controladores Programables Micrologix 1000**, 1761-TD001B-ES-P, [www.ab.com](http://www.ab.com), Mayo 2003.

17. Caicedo Mario, Sierra Hugo, **Desarrollo de Simuladores Didácticos para Controladores Lógicos Programables**, 1998.
18. Estrada, Henry, **Fundamentos de Controladores Lógicos Programables**, [www.geocities.com/ingenieria\\_control/control1.htm](http://www.geocities.com/ingenieria_control/control1.htm), Agosto 2007.
19. Rockwell Automation, **Sistemas SLC-500 Guía de Selección**, 1747-SG001B-ES-P, [www.ab.com](http://www.ab.com), Marzo 2005.
20. Rockwell Automation, **Descripción de Productos RSView**, RSVIEW-BR001A-ES-P, [www.ab.com](http://www.ab.com), Mayo 2005.

## **ANEXOS**

## INDICE DE FIGURAS

Figura. 2.1. Equipos de la serie SLC-500.....	4
Figura. 2.2. Archivos del Procesador SLC-500.....	8
Figura. 2.3. Ejemplo de programación con instrucción OTE.....	16
Figura. 2.4. Ejemplo de programación con instrucción OTL Y OTU.....	16
Figura. 2.5. Esquema de un controlador PID aplicado al control de nivel de un líquido....	32
Figura. 2.6. Ventana de configuración del bloque de control PID.....	33
Figura. 2.7. Desplegar la ventana de configuración de la instrucción PID.....	33
Figura. 2.8. Pantalla de configuración de la instrucción MSG.....	40
Figura. 2.9. Equipos de la serie Micrologix 1000.....	42
Figura. 2.10. Archivos del procesador Micrologix 1000.....	44
Figura. 2.11. Descripción del módulo de Interfase Ethernet 1761-NET-ENIW.....	48
Figura. 2.12. Pantalla de configuración 1 ENIW.....	49
Figura. 2.13. Pantalla de configuración 2 ENIW.....	50
Figura. 2.14. Pantalla de configuración 3 ENIW.....	50
Figura. 2.15. Pantalla de configuración 4 ENIW.....	51
Figura. 2.16. Pantalla de configuración 5 ENIW.....	51
Figura. 2.17. Pantalla de configuración 6 ENIW.....	52
Figura. 2.18. Verificación de la dirección IP del módulo NET-ENIW.....	52
Figura. 2.19. Verificación de la conexión del módulo NET-ENIW y el PLC SLC 5/03....	52
Figura. 2.20. Descripción del módulo NET-AIC.....	54
Figura. 2.21. Configuración de los nodos de comunicación del módulo NET-AIC.....	55
Figura. 2.22. Conexión del módulo NET-AIC en modo aislador punto a punto.....	55
Figura. 3.1. Versión RSLinx Classic Gateway que posee la ESPE.....	57
Figura. 3.2. Configuración del Driver de comunicación serial RS232 (1).....	59
Figura. 3.3. Configuración del Driver de comunicación serial RS232 (2).....	59
Figura. 3.4. Configuración del Driver de comunicación serial RS232 (3).....	59
Figura. 3.5. Configuración del Driver de comunicación serial RS232 (4).....	60
Figura. 3.6. Verificación del driver de comunicación serial RS232.....	61
Figura. 3.7. Configuración del Driver de comunicación Ethernet (1).....	61
Figura. 3.8. Configuración del Driver de comunicación Ethernet (2).....	62
Figura. 3.9. Configuración del Driver de comunicación Ethernet (3).....	62
Figura. 3.10. Configuración del Driver de comunicación Ethernet (4).....	63
Figura. 3.11. Verificación del driver de comunicación serial Ethernet.....	63
Figura. 3.12. Versión de RSLogix 500 que posee la ESPE.....	65
Figura. 3.13. Creación de un nuevo archivo en RSLogix 500.....	66
Figura. 3.14. Selección del tipo de PLC en RSLogix 500.....	67
Figura. 3.15. Configuración de los módulos I/O en RSLogix 500 (1).....	67
Figura. 3.16. Configuración de los módulos I/O en RSLogix 500 (2).....	68
Figura. 3.17. Configuración de los módulos I/O en RSLogix 500 (3).....	68
Figura. 3.18. Descargar el programa ladder en el PLC.....	69
Figura. 3.19. Configuración del canal 0 DF1 Full-Duplex del PLC (1).....	69
Figura. 3.20. Configuración del canal 0 DF1 Full-Duplex del PLC (2).....	70
Figura. 3.21. Configuración del canal 0 DF1 Full-Duplex del PLC (3).....	70
Figura. 3.22. Configuración del canal 1 Ethernet del PLC (1).....	71
Figura. 3.23. Configuración del canal 1 Ethernet del PLC (2).....	72
Figura. 3.24. Configuración del canal 1 Ethernet del PLC (3).....	72
Figura. 3.25. Versión del software RSView32 que posee la ESPE.....	75
Figura. 3.26. Creación de un archivo nuevo en RSView32 (1).....	77

Figura. 3.27. Creación de un archivo nuevo en RSView32 (2).....	78
Figura. 3.28. Configuración del canal de comunicación en RSView32 (1).....	78
Figura. 3.29. Configuración del canal de comunicación en RSView32 (2).....	79
Figura. 3.30. Configuración del canal de comunicación en RSView32 (3).....	79
Figura. 3.31. Configuración del nodo de comunicación en RSView32 (1).....	80
Figura. 3.32. Configuración del nodo de comunicación en RSView32 (2).....	80
Figura. 3.33. Configuración del nodo de comunicación en RSView32 (3).....	81
Figura. 3.34. Creación de tags en RSView32.....	81
Figura. 3.35. Configurar tags en RSView32 .....	82
Figura. 3.36. Creación de pantallas gráficas en RSView32 (1).....	83
Figura. 3.37. Creación de pantallas gráficas en RSView32 (2).....	83
Figura. 3.38. Animación de objetos en RSView32 (1).....	84
Figura. 3.39. Animación de objetos en RSView32 (2).....	84
Figura. 3.40. Modo Test Run en RSView32 .....	85
Figura. 3.41. Creación de Trends en RSView32 (1) .....	85
Figura. 3.42. Creación de Trends en RSView32 (2) .....	86
Figura. 3.43. Creación de Trends en RSView32 (3) .....	86
Figura. 3.44. Creación de Eventos en RSView32 (1).....	87
Figura. 3.45. Creación de Eventos en RSView32 (2).....	88
Figura. 3.46. Creación de Eventos en RSView32 (3).....	88
Figura. 3.47. Creación de Eventos en RSView32 (4).....	88
Figura. 3.48. Configurar la corrida del programa en RSView32 (1).....	89
Figura. 3.49. Configurar la corrida del programa en RSView32 (2).....	90
Figura. 3.50. Configurar la corrida del programa en RSView32 (3).....	90
Figura. 3.51 Configurar la corrida del programa en RSView32 (4).....	91
Figura. 4.1. Configuración del Driver de comunicación serial RS232 (1).....	93
Figura. 4.2. Configuración del Driver de comunicación serial RS232 (2).....	93
Figura. 4.3. Configuración del Driver de comunicación serial RS232 (3).....	94
Figura. 4.4. Configuración del Driver de comunicación serial RS232 (4).....	94
Figura. 4.5. Verificación del driver de comunicación serial RS232.....	95
Figura. 4.6. Creación de un nuevo archivo en RSLogix 500 .....	96
Figura. 4.7. Selección del tipo de PLC en RSLogix 500.....	96
Figura. 4.8. Configuración de los módulos I/O en RSLogix 500 (1) .....	97
Figura. 4.9. Configuración de los módulos I/O en RSLogix 500 (2) .....	97
Figura. 4.10. Configuración de los módulos I/O en RSLogix 500 (3) .....	98
Figura. 4.10. Descargar el programa ladder en el PLC .....	98
Figura. 4.11. Ejemplo de programación con instrucción OTE.....	101
Figura. 4.12. Ejemplo de programación con instrucción OTL Y OTU.....	101
Figura. 4.13. Planteamiento Práctica #1 .....	102
Figura. 4.14. Planteamiento Práctica #2 .....	105
Figura. 4.15. Planteamiento Práctica #3 .....	112
Figura. 4.16. Planteamiento Práctica #4 .....	117
Figura. 4.17. Planteamiento Práctica #5 .....	120
Figura. 4.18. Selector entradas de voltaje o corriente .....	122
Figura. 4.19. Conexión I/O del módulo NIO4V .....	123
Figura. 4.20. Planteamiento Práctica #6.....	132
Figura. 4.21. Planteamiento Práctica #7 .....	136
Figura. 4.22. Planteamiento Práctica #8.....	143
Figura. 4.23. Banda diferencial o Histéresis.....	148
Figura. 4.24. Creación de un archivo nuevo en RSView32 (1).....	149

Figura. 4.25. Creación de un archivo nuevo en RSView32 (2).....	149
Figura. 4.26. Configuración del canal de comunicación en RSView32 (1).....	150
Figura. 4.27. Configuración del canal de comunicación en RSView32 (2).....	150
Figura. 4.28. Configuración del nodo de comunicación en RSView32 (1).....	151
Figura. 4.29. Configuración del nodo de comunicación en RSView32 (2).....	151
Figura. 4.30. Creación de tags en RSView32.....	152
Figura. 4.31. Configurar tags en RSView32 .....	153
Figura. 4.32. Creación de pantallas gráficas en RSView32 (1).....	154
Figura. 4.33. Creación de pantallas gráficas en RSView32 (2).....	154
Figura. 4.34. Animación de objetos en RSView32 (1).....	155
Figura. 4.35. Animación de objetos en RSView32 (2).....	155
Figura. 4.36. Modo Test Run en RSView32 .....	156
Figura. 4.37. Creación de Trends en RSView32 (1) .....	156
Figura. 4.38. Creación de Trends en RSView32 (2) .....	157
Figura. 4.39. Creación de Trends en RSView32 (3) .....	157
Figura. 4.40. Creación de Eventos en RSView32 (1).....	158
Figura. 4.41. Creación de Eventos en RSView32 (2).....	159
Figura. 4.42. Creación de Eventos en RSView32 (3).....	159
Figura. 4.43. Creación de Eventos en RSView32 (4).....	159
Figura. 4.44. Configurar la corrida del programa en RSView32 (1).....	160
Figura. 4.45. Configurar la corrida del programa en RSView32 (2).....	161
Figura. 4.46. Configurar la corrida del programa en RSView32 (3).....	161
Figura. 4.47. Configurar la corrida del programa en RSView32 (4).....	162
Figura. 4.48. Planteamiento Práctica #10 .....	163
Figura. 4.49. Esquema de un controlador PID aplicado al control de nivel de un líquido	167
Figura. 4.50. Ventana de configuración del bloque de control PID .....	169
Figura. 4.51. Desplegar la ventana de configuración de la instrucción PID .....	169
Figura. 4.51. Planteamiento Práctica #11 .....	174
Figura. 4.52. Planteamiento Práctica #12 .....	178
Figura. 4.53. Aplicación de control de Razón .....	181
Figura. 4.54. Diagrama de control de Razón.....	181
Figura. 4.55. Primer método de control de Razón.....	182
Figura. 4.56. Segundo método de control de Razón.....	182
Figura. 4.57. Planteamiento Práctica #13 .....	183
Figura. 4.58. Aplicación de control por Sobreposición.....	186
Figura. 4.59. Diagrama de control por Sobreposición.....	187
Figura. 4.60. Planteamiento Práctica #14 (1) .....	188
Figura. 4.61. Planteamiento Práctica #14 (2) .....	189
Figura. 4.62. Aplicación de control en Cascada.....	192
Figura. 4.63. Diagrama de control en Cascada (1) .....	192
Figura. 4.64. Diagrama de control en Cascada (2) .....	193
Figura. 4.65. Planteamiento Práctica #15 (1) .....	194
Figura. 4.66. Planteamiento Práctica #15 (2) .....	195
Figura. 4.67. Aplicación de control Selectivo .....	198
Figura. 4.68. Planteamiento Práctica #16 (1) .....	199
Figura. 4.69. Planteamiento Práctica #16 (2) .....	200
Figura. 4.70. Nueva conexión Hyperterminal .....	203
Figura. 4.71. Seleccionar puerto en Hyperterminal.....	203
Figura. 4.72. Parámetros de comunicación en Hyperterminal .....	203
Figura. 4.73. Conexión de Hyperterminal con los Radio módems.....	204

Figura. 4.74. Menú Principal de Configuración de los Radio Modem.....	204
Figura. 4.75. Configuración del canal 0 DF1 Full-Duplex del PLC (1).....	210
Figura. 4.76. Configuración del canal 0 DF1 Full-Duplex del PLC (2).....	211
Figura. 4.77. Configuración del canal 0 DF1 Full-Duplex del PLC (7).....	211
Figura. 4.78. Descripción del Módulo 1761-NET-AIC.....	212
Figura. 4.79. Configuración de los nodos de comunicación del módulo NET-AIC .....	213
Figura. 4.80. Conexión del módulo NET-AIC en modo aislador punto a punto .....	213
Figura. 4.81. Planteamiento Práctica #17 (1) .....	214
Figura. 4.82. Planteamiento Práctica #17 (2) .....	214
Figura. 4.83. Configuración del Driver de comunicación Ethernet (1).....	217
Figura. 4.84. Configuración del Driver de comunicación Ethernet (2).....	218
Figura. 4.85. Configuración del Driver de comunicación Ethernet (3).....	218
Figura. 4.86. Configuración del Driver de comunicación Ethernet (4).....	219
Figura. 4.87. Verificación del driver de comunicación serial Ethernet.....	219
Figura. 4.88. Configuración del canal 1 Ethernet del PLC (1).....	220
Figura. 4.89. Configuración del canal 1 Ethernet del PLC (2).....	220
Figura. 4.90. Configuración del canal 1 Ethernet del PLC (3).....	221
Figura. 4.91. Pantalla de configuración de la instrucción MSG.....	223
Figura. 4.92. Configuración del canal Ethernet en RSView32 (1).....	225
Figura. 4.93. Configuración del canal Ethernet en RSView32 (2).....	225
Figura. 4.94. Configuración del canal Ethernet en RSView32 (3).....	226
Figura. 4.95. Configuración del canal Ethernet en RSView32 (4).....	226
Figura. 4.96. Descripción del módulo de Interfase Ethernet 1761-NET-ENIW .....	227
Figura. 4.97. Pantalla de configuración 1 ENIW .....	228
Figura. 4.98. Pantalla de configuración 2 ENIW .....	228
Figura. 4.99. Pantalla de configuración 3 ENIW .....	229
Figura. 4.100. Pantalla de configuración 4 ENIW .....	229
Figura. 4.101. Pantalla de configuración 5 ENIW .....	230
Figura. 4.102. Pantalla de configuración 6 ENIW .....	230
Figura. 4.103. Verificación de la dirección IP del módulo NET-ENIW .....	230
Figura. 4.104. Verificación de la conexión del módulo NET-ENIW y el PLC SLC 5/03	231
Figura. 4.105. Planteamiento Práctica #18 (1) .....	231
Figura. 4.106. Planteamiento Práctica #18 (2) .....	232
Figura. 4.107. Planteamiento Práctica #18 (3) .....	232
Figura. 4.108. Planteamiento Práctica #18 (4) .....	233
Figura. 5.1. Planteamiento Práctica #1 .....	236
Figura. 5.2. Desarrollo Práctica #1 (1) .....	237
Figura. 5.3. Desarrollo Práctica #1 (2) .....	237
Figura. 5.4. Desarrollo Práctica #1 (3) .....	238
Figura. 5.5. Desarrollo Práctica #1 (4) .....	239
Figura. 5.6. Desarrollo Práctica #1 (5) .....	240
Figura. 5.7. Planteamiento Práctica #2.....	243
Figura. 5.8. Desarrollo Práctica #2 (1) .....	243
Figura. 5.9. Desarrollo Práctica #2 (2) .....	244
Figura. 5.10. Desarrollo Práctica #2 (3) .....	245
Figura. 5.11. Desarrollo Práctica #2 (4) .....	246
Figura. 5.12. Planteamiento Práctica #3 .....	249
Figura. 5.13. Desarrollo Práctica #3 (1) .....	250
Figura. 5.14. Desarrollo Práctica #3 (2) .....	250
Figura. 5.15. Desarrollo Práctica #3 (3) .....	251

Figura. 5.16. Desarrollo Práctica #3 (4) .....	252
Figura. 5.17. Desarrollo Práctica #3 (5) .....	253
Figura. 5.18. Planteamiento Práctica #4 .....	256
Figura. 5.19. Desarrollo Práctica #4 (1) .....	256
Figura. 5.20. Desarrollo Práctica #4 (2) .....	257
Figura. 5.21. Desarrollo Práctica #4 (3) .....	258
Figura. 5.22. Desarrollo Práctica #4 (4) .....	259
Figura. 5.23. Desarrollo Práctica #4 (5) .....	260
Figura. 5.24. Desarrollo Práctica #4 (6) .....	261
Figura. 5.25. Planteamiento Práctica #5 .....	264
Figura. 5.26. Desarrollo Práctica #5 (1) .....	264
Figura. 5.27. Desarrollo Práctica #5 (2) .....	265
Figura. 5.28. Desarrollo Práctica #5 (3) .....	266
Figura. 5.29. Desarrollo Práctica #5 (4) .....	267
Figura. 5.30. Planteamiento Práctica #6 .....	270
Figura. 5.31. Desarrollo Práctica #6 (1) .....	271
Figura. 5.32. Desarrollo Práctica #6 (2) .....	271
Figura. 5.32. Desarrollo Práctica #6 (3) .....	272
Figura. 5.33. Desarrollo Práctica #6 (4) .....	273
Figura. 5.34. Desarrollo Práctica #6 (5) .....	274
Figura. 5.35. Planteamiento Práctica #7 .....	277
Figura. 5.36. Desarrollo Práctica #7 (1) .....	277
Figura. 5.37. Desarrollo Práctica #7 (2) .....	278
Figura. 5.39. Desarrollo Práctica #7 (4) .....	279
Figura. 5.40. Desarrollo Práctica #7 (5) .....	280
Figura. 5.41. Planteamiento Práctica #8 .....	283
Figura. 5.42. Desarrollo Práctica #8 (1) .....	283
Figura. 5.43. Desarrollo Práctica #8 (2) .....	284
Figura. 5.44. Desarrollo Práctica #8 (3) .....	285
Figura. 5.45. Desarrollo Práctica #8 (4) .....	286
Figura. 5.46. Desarrollo Práctica #9 (1) .....	289
Figura. 5.47. Desarrollo Práctica #9 (2) .....	289
Figura. 5.48. Desarrollo Práctica #9 (3) .....	291
Figura. 5.49. Desarrollo Práctica #9 (4) .....	292
Figura. 5.50. Planteamiento Práctica #10 .....	295
Figura. 5.51. Desarrollo Práctica #10 (1) .....	296
Figura. 5.52. Desarrollo Práctica #10 (2) .....	296
Figura. 5.53. Desarrollo Práctica #10 (3) .....	297
Figura. 5.54. Desarrollo Práctica #10 (4) .....	298
Figura. 5.55. Desarrollo Práctica #10 (5) .....	298
Figura. 5.56. Desarrollo Práctica #10 (6) .....	299
Figura. 5.57. Desarrollo Práctica #10 (7) .....	299
Figura. 5.58. Desarrollo Práctica #10 (8) .....	301
Figura. 5.59. Desarrollo Práctica #10 (9) .....	302
Figura. 5.60. Desarrollo Práctica #10 (10) .....	303
Figura. 5.61. Desarrollo Práctica #10 (11) .....	303
Figura. 5.62. Desarrollo Práctica #10 (12) .....	304
Figura. 5.63. Planteamiento Práctica #11 .....	307
Figura. 5.64. Desarrollo Práctica #11 (1) .....	308
Figura. 5.65. Desarrollo Práctica #11 (2) .....	308

Figura. 5.67. Desarrollo Práctica #11 (3) .....	309
Figura. 5.68. Desarrollo Práctica #11 (4) .....	310
Figura. 5.69. Desarrollo Práctica #11 (5) .....	310
Figura. 5.70. Desarrollo Práctica #11 (6) .....	311
Figura. 5.71. Desarrollo Práctica #11 (7) .....	311
Figura. 5.72. Desarrollo Práctica #11 (8) .....	312
Figura. 5.73. Desarrollo Práctica #11 (9) .....	314
Figura. 5.74. Desarrollo Práctica #11 (10) .....	315
Figura. 5.75. Desarrollo Práctica #11 (11) .....	315
Figura. 5.76. Desarrollo Práctica #11 (12) .....	316
Figura. 5.77. Desarrollo Práctica #11 (13) .....	316
Figura. 5.78. Desarrollo Práctica #11 (14) .....	317
Figura. 5.79. Desarrollo Práctica #11 (15) .....	318
Figura. 5.80. Desarrollo Práctica #11 (16) .....	318
Figura. 5.81. Desarrollo Práctica #11 (17) .....	319
Figura. 5.82. Desarrollo Práctica #11 (18) .....	320
Figura. 5.83. Desarrollo Práctica #11 (19) .....	320
Figura. 5.84. Desarrollo Práctica #11 (20) .....	322
Figura. 5.85. Planteamiento Práctica #12 .....	325
Figura. 5.86. Desarrollo Práctica #12 (1) .....	326
Figura. 5.87. Desarrollo Práctica #12 (2) .....	326
Figura. 5.88. Desarrollo Práctica #12 (3) .....	327
Figura. 5.89. Desarrollo Práctica #12 (4) .....	328
Figura. 5.90. Desarrollo Práctica #12 (5) .....	328
Figura. 5.91. Desarrollo Práctica #12 (6) .....	329
Figura. 5.92. Desarrollo Práctica #12 (7) .....	329
Figura. 5.93. Desarrollo Práctica #12 (8) .....	330
Figura. 5.94. Desarrollo Práctica #12 (9) .....	332
Figura. 5.95. Desarrollo Práctica #12 (10) .....	332
Figura. 5.96. Desarrollo Práctica #12 (11) .....	333
Figura. 5.97. Desarrollo Práctica #12 (12) .....	333
Figura. 5.98. Desarrollo Práctica #12 (13) .....	334
Figura. 5.99. Desarrollo Práctica #12 (14) .....	336
Figura. 5.100. Planteamiento Práctica #13 .....	339
Figura. 5.101. Desarrollo Práctica #13 (1) .....	340
Figura. 5.102. Desarrollo Práctica #13 (2) .....	340
Figura. 5.103. Desarrollo Práctica #13 (3) .....	341
Figura. 5.104. Desarrollo Práctica #13 (4) .....	342
Figura. 5.105. Desarrollo Práctica #13 (5) .....	342
Figura. 5.106. Desarrollo Práctica #13 (6) .....	343
Figura. 5.107. Desarrollo Práctica #13 (7) .....	343
Figura. 5.108. Desarrollo Práctica #13 (8) .....	344
Figura. 5.109. Desarrollo Práctica #13 (9) .....	346
Figura. 5.110. Desarrollo Práctica #13 (10) .....	346
Figura. 5.111. Desarrollo Práctica #13 (11) .....	347
Figura. 5.112. Desarrollo Práctica #13 (12) .....	347
Figura. 5.113. Planteamiento Práctica #14 (1) .....	350
Figura. 5.114. Planteamiento Práctica #14 (2) .....	351
Figura. 5.115. Desarrollo Práctica #14 (1) .....	352
Figura. 5.116. Desarrollo Práctica #14 (2) .....	352

Figura. 5.117. Desarrollo Práctica #14 (3) .....	354
Figura. 5.118. Desarrollo Práctica #14 (4) .....	355
Figura. 5.119. Desarrollo Práctica #14 (5) .....	355
Figura. 5.120. Desarrollo Práctica #14 (6) .....	356
Figura. 5.121. Desarrollo Práctica #14 (7) .....	356
Figura. 5.122. Desarrollo Práctica #14 (8) .....	357
Figura. 5.123. Desarrollo Práctica #14 (9) .....	359
Figura. 5.124. Desarrollo Práctica #14 (10) .....	360
Figura. 5.125. Desarrollo Práctica #14 (11) .....	361
Figura. 5.126. Desarrollo Práctica #14 (12) .....	361
Figura. 5.127. Planteamiento Práctica #15 (1) .....	364
Figura. 5.128. Planteamiento Práctica #15 (2) .....	364
Figura. 5.129. Desarrollo Práctica #15 (1) .....	365
Figura. 5.130. Desarrollo Práctica #15 (2) .....	366
Figura. 5.131. Desarrollo Práctica #15 (3) .....	366
Figura. 5.132. Desarrollo Práctica #15 (4) .....	367
Figura. 5.133. Desarrollo Práctica #15 (5) .....	367
Figura. 5.134. Desarrollo Práctica #15 (6) .....	368
Figura. 5.135. Desarrollo Práctica #15 (7) .....	368
Figura. 5.136. Desarrollo Práctica #15 (8) .....	369
Figura. 5.137. Desarrollo Práctica #15 (9) .....	371
Figura. 5.138. Desarrollo Práctica #15 (10) .....	372
Figura. 5.139. Desarrollo Práctica #15 (11) .....	372
Figura. 5.140. Desarrollo Práctica #15 (12) .....	373
Figura. 5.141. Planteamiento Práctica #16 (1) .....	376
Figura. 5.142. Planteamiento Práctica #16 (2) .....	377
Figura. 5.143. Desarrollo Práctica #16 (1) .....	378
Figura. 5.144. Desarrollo Práctica #16 (2) .....	378
Figura. 5.145. Desarrollo Práctica #16 (3) .....	379
Figura. 5.146. Desarrollo Práctica #16 (4) .....	380
Figura. 5.147. Desarrollo Práctica #16 (5) .....	380
Figura. 5.148. Desarrollo Práctica #16 (6) .....	381
Figura. 5.149. Desarrollo Práctica #16 (7) .....	381
Figura. 5.150. Desarrollo Práctica #16 (8) .....	382
Figura. 5.151. Desarrollo Práctica #16 (9) .....	384
Figura. 5.152. Desarrollo Práctica #16 (10) .....	385
Figura. 5.153. Desarrollo Práctica #16 (11) .....	385
Figura. 5.154. Desarrollo Práctica #16 (12) .....	386
Figura. 5.155. Planteamiento Práctica #17 (1) .....	389
Figura. 5.156. Planteamiento Práctica #17 (2) .....	389
Figura. 5.157. Desarrollo Práctica #17 (1) .....	390
Figura. 5.158. Desarrollo Práctica #17 (2) .....	393
Figura. 5.159. Desarrollo Práctica #17 (3) .....	393
Figura. 5.160. Desarrollo Práctica #17 (4) .....	394
Figura. 5.161. Desarrollo Práctica #17 (5) .....	395
Figura. 5.162. Desarrollo Práctica #17 (6) .....	395
Figura. 5.163. Desarrollo Práctica #17 (7) .....	396
Figura. 5.164. Planteamiento Práctica #18 (1) .....	400
Figura. 5.165. Planteamiento Práctica #18 (2) .....	400
Figura. 5.166. Planteamiento Práctica #18 (3) .....	401

Figura. 5.167. Planteamiento Práctica #18 (4) .....	401
Figura. 5.168. Desarrollo Práctica #18 (1) .....	402
Figura. 5.169. Desarrollo Práctica #18 (2) .....	405
Figura. 5.170. Desarrollo Práctica #18 (3) .....	406
Figura. 5.171. Desarrollo Práctica #18 (4) .....	407
Figura. 5.172. Desarrollo Práctica #18 (5) .....	408
Figura. 5.173. Desarrollo Práctica #18 (6) .....	409
Figura. 5.174. Desarrollo Práctica #18 (7) .....	409
Figura. 5.175. Desarrollo Práctica #18 (8) .....	410
Figura. 5.176. Desarrollo Práctica #18 (9) .....	410
Figura. 5.177. Desarrollo Práctica #18 (10) .....	411
Figura. 5.178. Desarrollo Práctica #18 (11) .....	411
Figura. 5.179. Desarrollo Práctica #18 (12) .....	412
Figura. 5.180. Desarrollo Práctica #18 (13) .....	413
Figura. 5.181. Desarrollo Práctica #18 (14) .....	413
Figura. 5.182. Desarrollo Práctica #18 (15) .....	414
Figura. 5.183. Desarrollo Práctica #18 (16) .....	414
Figura. 5.184. Desarrollo Práctica #18 (17) .....	415
Figura. 5.185. Desarrollo Práctica #18 (18) .....	415
Figura. 5.186. Desarrollo Práctica #18 (19) .....	416
Figura. 5.187. Desarrollo Práctica #18 (20) .....	416
Figura. 5.188. Desarrollo Práctica #18 (21) .....	420
Figura. 5.189. Desarrollo Práctica #18 (22) .....	420
Figura. 5.190. Desarrollo Práctica #18 (23) .....	421

## INDICE DE TABLAS

Tabla. 2.1. Especificaciones SLC500.....	6
Tabla. 2.2. Direccionamiento I/O .....	9
Tabla. 2.3. Direccionamiento STATUS .....	9
Tabla. 2.4. Direccionamiento BITS.....	10
Tabla. 2.5. Direccionamiento TIMER.....	10
Tabla. 2.6. Ejemplos Direccionamiento TIMER.....	11
Tabla. 2.7. Direccionamiento COUNTER.....	11
Tabla. 2.8. Ejemplo Direccionamiento COUNTER.....	11
Tabla. 2.9. Direccionamiento CONTROL.....	12
Tabla. 2.10. Direccionamiento ENTERO.....	12
Tabla. 2.11. Direccionamiento FLOTANTE.....	13
Tabla. 2.12. Conjunto de intrucciones SLC-500 .....	14
Tabla. 2.13. Instrucciones de Entrada .....	15
Tabla. 2.14. Instrucciones de Salida.....	16
Tabla. 2.15. Direcciones asignada en el ejemplo .....	17
Tabla. 2.16. Instrucción CTU .....	17
Tabla. 2.17. Bits de Status de la instrucción CTU.....	18
Tabla. 2.18. Instrucción CTD .....	18
Tabla. 2.19. Bits de Status de la instrucción CTD.....	18
Tabla. 2.20. Instrucción RES .....	19
Tabla. 2.21. Instrucción TON .....	20
Tabla. 2.22. Bits de Status de la instrucción TON .....	20
Tabla. 2.23. Instrucciones de Comparación .....	21
Tabla. 2.24 Instrucciones de Tranferencia de Datos .....	22
Tabla. 2.25. Instrucción ADD .....	23
Tabla. 2.26. Bits aritméticos de la instrucción ADD.....	23
Tabla. 2.27. Instrucción SUB .....	24
Tabla. 2.28. Bits aritméticos de la instrucción SUB.....	24
Tabla. 2.29. Instrucción MUL .....	24
Tabla. 2.30. Bits aritméticos de la instrucción MUL.....	25
Tabla. 2.31. Instrucción DIV .....	25
Tabla. 2.32. Bits aritméticos de la instrucción DIV .....	25
Tabla. 2.33. Registros Reloj/Calendario.....	26
Tabla. 2.34. Instrucciones FFL y FFU.....	27
Tabla. 2.35. Instrucciones LFL y LFU .....	28
Tabla. 2.36. Instrucción CPT.....	29
Tabla. 2.37. Instrucciones de Escalado.....	30
Tabla. 2.38. Instrucciones JMP y LBL.....	31
Tabla. 2.39. Instrucciones PID .....	31
Tabla. 2.40. Rango de valores de ganancias estándar .....	32
Tabla. 2.41. Bloque de Control PID .....	35
Tabla. 2.42. Parámetros de entrada PID .....	36
Tabla. 2.43. Parámetros de Salida PID.....	37
Tabla. 2.44. Explicación Instrucción SCL.....	37
Tabla. 2.45. Instrucción MSG .....	38
Tabla. 2.46. Número de Nodo Instrucción MSG.....	41
Tabla. 2.47. Características generales Serie Micrologix .....	44

Tabla. 2.48. Instrucciones SLC500 y Micrologix .....	46
Tabla. 4.1. Direccionamiento I/O .....	99
Tabla. 4.2. Instrucciones de Entrada .....	100
Tabla. 4.3. Instrucciones de Salida .....	100
Tabla. 4.4. Direcciones del ejemplo de Instrucciones I/O .....	101
Tabla. 4.5. Direccionamiento COUNTER.....	108
Tabla. 4.6. Ejemplos Direccionamiento COUNTER .....	108
Tabla. 4.7. Instrucción CTU .....	109
Tabla. 4.8. Bits de Status de la Instrucción CTU .....	109
Tabla. 4.9. Instrucción CTD .....	110
Tabla. 4.10. Bits de Status de la Instrucción CTD .....	110
Tabla. 4.11. Instrucción RES.....	111
Tabla. 4.12. Señales I/O Práctica #3.....	112
Tabla. 4.13. Direccionamiento TIMER .....	115
Tabla. 4.14. Ejemplos Direccionamiento TIMER.....	115
Tabla. 4.15. Instrucción TON.....	116
Tabla. 4.16. Bits de Status de la instrucción TON .....	116
Tabla. 4.17. Direccionamiento I/O .....	124
Tabla. 4.18. Límites de entrada de corriente y voltaje .....	124
Tabla. 4.19. Límites de salida de voltaje .....	126
Tabla. 4.20. Direccionamiento ENTERO.....	127
Tabla. 4.21. Instrucciones de Comparación .....	128
Tabla. 4.22. Instrucciones de Transferencia de Datos .....	129
Tabla. 4.23. Instrucción ADD .....	129
Tabla. 4.24. Bits de la Instrucción ADD .....	130
Tabla. 4.25. Instrucción SUB .....	130
Tabla. 4.26. Bits de la Instrucción SUB .....	130
Tabla. 4.27. Instrucción MUL .....	130
Tabla. 4.28. Bits de la instrucción MUL .....	131
Tabla. 4.29. Instrucción DIV .....	131
Tabla. 4.30. Bits de la Instrucción DIV .....	131
Tabla. 4.31. Registros Reloj/Calendario.....	135
Tabla. 4.32. Horarios Práctica #7 .....	136
Tabla. 4.33. Instrucciones FFL y FFU.....	139
Tabla. 4.34. Instrucciones LFL y LFU .....	140
Tabla. 4.35. Instrucción CPT.....	141
Tabla. 4.36. Instrucciones de Escalado.....	142
Tabla. 4.37 Instrucciones JMP y LBL.....	145
Tabla. 4.38. Instrucción PID.....	167
Tabla. 4.38. Ganancias Estándar PID .....	168
Tabla. 4.40. Control Block PID.....	170
Tabla. 4.41. Parámetros de Entrada PID .....	172
Tabla. 4.42. Parámetros de salida PID.....	172
Tabla. 4.43. Explicación Instrucción SCL.....	173
Tabla. 4.44. Parámetro Data, Parity.....	206
Tabla. 4.45. Instrucción MSG .....	221
Tabla. 4.46. Local Node Instrucción MSG.....	224
Tabla. 5.1. Direcciones utilizadas Práctica #1 .....	240
Tabla. 5.2. Direcciones utilizadas Práctica #2.....	247
Tabla. 5.3. Señales utilizadas Práctica #3.....	249

Tabla. 5.4. Direcciones utilizadas Práctica #3.....	253
Tabla. 5.5. Direcciones utilizadas Práctica #4.....	262
Tabla. 5.6. Direcciones utilizadas Práctica #5.....	267
Tabla. 5.7. Valores de altura.....	273
Tabla. 5.8. Direcciones utilizadas Práctica #6.....	275
Tabla. 5.9. Horarios Práctica #7 .....	276
Tabla. 5.10. Direcciones utilizadas Práctica #7.....	280
Tabla. 5.11. Direcciones utilizadas Práctica #8.....	286
Tabla. 5.12. Direcciones utilizadas Práctica #9.....	293
Tabla. 5.13. Direcciones utilizadas Práctica #10.....	297
Tabla. 5.14. Tags utilizados Práctica #10.....	301
Tabla. 5.15. Direcciones utilizadas Práctica #11.....	309
Tabla. 5.16. Tags utilizados Práctica #11.....	313
Tabla. 5.17. Valores de sintonización PID .....	321
Tabla. 5.18. Direcciones utilizadas Práctica #12.....	327
Tabla. 5.19. Tags utilizados Práctica #12.....	331
Tabla. 5.20. Valores de sintonización PID .....	335
Tabla. 5.21. Direcciones utilizadas Práctica #13.....	341
Tabla. 5.22. Tags utilizados Práctica #13.....	345
Tabla. 5.23. Direcciones utilizadas Práctica #14.....	354
Tabla. 5.24. Tags utilizados Práctica #14.....	359
Tabla. 5.25. Direcciones utilizadas Práctica #15.....	367
Tabla. 5.26. Tags utilizados Práctica #15.....	371
Tabla. 5.27. Direcciones utilizadas Práctica #16.....	380
Tabla. 5.28. Tags utilizados Práctica #16.....	384
Tabla. 5.29. Tags utilizados Práctica #17.....	397
Tabla. 5.30. IP asignadas Práctica #18 .....	412
Tabla. 5.31. Tags utilizados Práctica #18.....	419

## **HOJA DE ENTREGA**

Este proyecto de grado fue entregado al Departamento de Eléctrica y Electrónica y reposa en la Escuela Politécnica del Ejército desde:

Sangolquí, a \_\_\_\_\_ del 2008

---

Sr. Galo Matias Robayo Gordón

AUTOR

---

Ing. Víctor Proaño

COORDINADOR DE CARRERA DE INGENIERÍA EN  
ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

## **ANEXOS**