



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA E
INSTRUMENTACIÓN

PROYECTO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN ELECTRÓNICA E
INSTRUMENTACIÓN

TEMA: “DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE
MÓDULO PARA VISUALIZAR POR MEDIO DE TOUCH
SCREEN, PC Y DISPOSITIVOS MÓVILES CURVAS
CARACTERÍSTICAS DE SENSORES DE TEMPERATURA EN
EL RANGO DE 0 A 200 °C, PARA EL LABORATORIO DE
CIRCUITOS ELECTRÓNICOS DE LA UNIVERSIDAD DE LAS
FUERZAS ARMADAS ESPE EXTENSIÓN LATACUNGA”

AUTORES: EDGAR JAVIER CAJAS OÑA

HERNÁN LEONARDO YANCHAPAXI ARMAS

DIRECTOR: ING. JOSÉ BUCHELI

CODIRECTOR: ING. DAVID RIVAS

LATACUNGA

2015



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

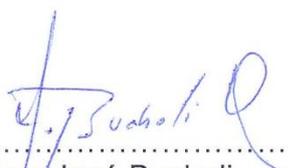
CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN

CERTIFICACIÓN

Certifico que el trabajo de titulación, **“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE MÓDULO PARA VISUALIZAR POR MEDIO DE TOUCH SCREEN, PC Y DISPOSITIVOS MÓVILES CURVAS CARACTERÍSTICAS DE SENSORES DE TEMPERATURA EN EL RANGO DE 0 A 200 °C, PARA EL LABORATORIO DE CIRCUITOS ELECTRÓNICOS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE EXTENSIÓN LATACUNGA”** realizado por los señores **EDGAR JAVIER CAJAS OÑA** y **HERNÁN LEONARDO YANCHAPAXI ARMAS**, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlos y autorizarles a los señores **EDGAR JAVIER CAJAS OÑA** y **HERNÁN LEONARDO YANCHAPAXI ARMAS** para que lo sustente públicamente.

Latacunga, 12 de noviembre del 2015

Atentamente,


.....
Ing. José Bucheli
DIRECTOR


.....
Ing. David Rivas
CODIRECTOR



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN
AUTORÍA DE RESPONSABILIDAD

Nosotros, **EDGAR JAVIER CAJAS OÑA**, con cédula de identidad N° 050326112-5 y **HERNÁN LEONARDO YANCHAPAXI ARMAS**, con cédula de identidad N° 050249564-1, declaramos que este trabajo de titulación **“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE MÓDULO PARA VISUALIZAR POR MEDIO DE TOUCH SCREEN, PC Y DISPOSITIVOS MÓVILES CURVAS CARACTERÍSTICAS DE SENSORES DE TEMPERATURA EN EL RANGO DE 0 A 200 °C, PARA EL LABORATORIO DE CIRCUITOS ELECTRÓNICOS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE EXTENSIÓN LATACUNGA”** ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaramos que este trabajo es de nuestra autoría, en virtud de ello nos declaramos responsables del contenido, veracidad y alcance de la investigación mencionada.

Latacunga, 12 de noviembre del 2015

Edgar Javier Cajas Oña

C.C.: 050326112-5

Hernán Leonardo Yanchapaxi Armas

C.C.: 050249564-1



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN

AUTORIZACIÓN

Nosotros, **EDGAR JAVIER CAJAS OÑA** y **HERNÁN LEONARDO YANCHAPAXI ARMAS**, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación **“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE MÓDULO PARA VISUALIZAR POR MEDIO DE TOUCH SCREEN, PC Y DISPOSITIVOS MÓVILES CURVAS CARACTERÍSTICAS DE SENSORES DE TEMPERATURA EN EL RANGO DE 0 A 200 °C, PARA EL LABORATORIO DE CIRCUITOS ELECTRÓNICOS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE EXTENSIÓN LATACUNGA”** cuyo contenido, ideas y criterios son de nuestra autoría y responsabilidad.

Latacunga, 12 de noviembre del 2015

Edgar Javier Cajas Oña

C.C.: 050326112-5

Hernán Leonardo Yanchapaxi Armas

C.C.: 050249564-1

DEDICATORIA

Este proyecto va dirigido a Dios y la Santísima Virgen de El Quinche, por darme la vida y llenar mi camino de muchas bendiciones permitiendo culminar mi carrera profesional. A mis padres, José y Elena por ser el pilar fundamental quienes me brindaron amor, cariño y a la vez con su ejemplo me apoyaron para culminar mis estudios. A ustedes, mi familia y amigos que con sus palabras de aliento de seguir adelante ayudaron a cristalizar el objetivo propuesto.

Javier.

Dedico este proyecto de Tesis a Dios y a mi familia, en especial a mis padres, mi hermano y mi novia. A Dios porque ha estado conmigo a cada paso que doy, cuidándome y dándome fortaleza para continuar, a mis padres, quienes a lo largo de mi vida han velado por mi bienestar y educación siendo mi soporte en todo momento, depositando su entera confianza en cada reto que se me presentaba sin dudar ni un solo momento de mí, a mi hermano y mi novia por su apoyo y por cada momento compartido en el día a día y durante la realización de este proyecto. Es por Ellos que soy lo que soy ahora. Los amo con mi vida.

Leonardo.

AGRADECIMIENTO

A Dios y la Santísima Virgen de El Quinche, por darme la vida, por sus bendiciones y la fe de poder culminar una de mis metas. A mis Padres, José Cajas y Elena Oña que siempre me formaron con buenos principios y valores, mi eterno agradecimiento ya que me apoyaron en los buenos y malos momentos para culminar mi vida profesional. De manera especial al Ing. José Bucheli y Ing. David Rivas quienes en calidad de tutores aportaron con sus conocimientos y experiencia. A mis amigos por sus palabras de aliento y de confianza.

Javier.

El presente Trabajo de Tesis me gustaría agradecer a Dios por bendecirme para llegar hasta donde he llegado, porque hiciste realidad este sueño anhelado.

A mi Familia por apoyarme en cada decisión que tome y por estar a mi lado en cada momento de mi vida. A mi hermano y mi novia por su ayuda en impulsarme y estar al pendiente para terminar este propósito.

Al Ing. José Bucheli y al Ing. Davis Rivas, por su amistad, su colaboración y sus recomendaciones para que este proyecto salga de lo mejor.

También quiero expresar mis agradecimientos a mis amigos por siempre tener una palabra de aliento y motivación para llegar a cumplir con este objetivo.

Leonardo.

	vii
ÍNDICE DE CONTENIDO	
PORTADA	i
CERTIFICACIÓN	ii
AUTORÍA DE RESPONSABILIDAD	iii
AUTORIZACIÓN	iv
DEDICATORIA	v
AGRADECIMIENTO	vi
ÍNDICE DE CONTENIDO	vii
ÍNDICE DE TABLAS	xii
ÍNDICE DE FIGURAS	xiii
ÍNDICE DE ECUACIONES	xxi
RESUMEN	xxii
ABSTRACT	xxiii

CAPÍTULO I

1	DEFINICIONES GENERALES	1
1.1	Antecedentes.	1
1.2	Planteamiento del problema.	2
1.3	Descripción resumida del proyecto.	3
1.4	Justificación e importancia.	3
1.5	Objetivos generales y específicos.	4
1.5.1	Objetivo general.	4
1.5.2	Objetivos específicos.	4
1.6	Introducción.	5
1.7	Descripción del generador de temperatura.	6
1.8	Dispositivo peltier.	6
1.8.1	Funcionamiento.	7

1.8.2	Esquema de un Módulo Peltier.	7
1.9	Módulo térmico o mufla.	8
1.9.1	Tipos de muflas.	8
1.10	Tipos de sistemas de control.	9
1.10.1	Control ON-OFF.	9
1.10.2	Control proporcional.	9
1.10.3	Control integral.	10
1.10.4	Control derivativo.	10
1.10.5	Control proporcional integral.	11
1.10.6	Control proporcional derivativo.	11
1.10.7	Control proporcional integral derivativo.	12
1.10.8	Control adaptivo.	14
1.10.9	Control neuronal.	15
1.10.10	Control difuso.	16
1.10.11	Control avanzado.	16
1.11	Sensores de temperatura.	16
1.11.1	LM35.	17
1.11.2	Termistor.	19
1.12	Termoresistencias-RTD (Pt100).	21
1.12.1	RTD 2 hilos.	23
1.12.2	RTD 3 hilos.	24
1.12.3	RTD 4 hilos.	24
1.12.4	Construcción de termoresistencias.	25
1.13	Termocupla.	25
1.13.1	Termocupla tipo J.	27
1.13.2	Termocupla tipo K.	27
1.13.3	Ventajas e inconvenientes de las termocuplas.	27

		ix
1.14	Software para el desarrollo.	29
1.14.1	Eclipse.	29
1.14.2	Matlab.	29
1.15	Controladores.	31
1.15.1	PAC.....	31
1.15.2	PLC.....	32
1.15.3	Microcontrolador STM32F407.....	33
1.15.4	Características de la tarjeta.	34
1.15.5	Periféricos y soportes.....	35
1.16	TFT touch screen.....	36
1.16.1	Desarrollo y utilización.	37
1.17	Dispositivos móviles.....	38
1.17.1	Sistemas operativos móviles.....	38
1.18	Protocolos de comunicación.	45
1.18.1	Ethernet.	46
1.18.2	Wifi.....	48

CAPÍTULO II

2	DISEÑO E IMPLEMENTACIÓN.....	50
2.1	Introducción.	50
2.2	Especificaciones y requisitos del sistema.	50
2.2.1	Sensores de temperatura.....	51
2.3	Diseño del generador de temperatura.....	55
2.3.1	Resistencia eléctrica (niquelina).....	56
2.3.2	Módulo peltier (TEC1-12715).....	58
2.3.3	Sistema de refrigeración.	59
2.3.4	Disipador de calor.	60

2.4	Control PID del generador de temperatura.	61	x
2.4.1	Programación del control PID.	62	
2.5	Adquisición de señales de los sensores de temperatura.	67	
2.5.1	MAX 31865 para RTD.....	67	
2.5.2	MAX31855J - MAX31855K para termocuplas.....	69	
2.5.3	Adquisición de datos de RTD y termocuplas.	71	
2.5.4	MCP6004 para termistores y LM35.....	95	
2.5.5	Adquisición de datos para termistores y LM35.	96	
2.6	Diseño y programación del HMI.....	104	
2.6.1	HMI touch screen por comunicación serial.	104	
2.6.2	HMI dispositivo móvil por red wifi.....	127	
2.6.3	HMI PC por red ethernet.	130	
2.7	Diagrama de bloques del sistema de control.	142	
2.8	Diagrama P&ID del sistema.....	142	
2.9	Diagrama eléctrico del sistema.	143	

CAPÍTULO III

3	PRUEBAS Y RESULTADOS OBTENIDOS.....	144	
3.1	Curva característica obtenida del sensor lm35.	144	
3.2	Curvas características obtenidas de los termistores.....	148	
3.2.1	Termistor de 1KΩ.....	149	
3.2.2	Termistor de 5KΩ.....	152	
3.2.3	Termistor de 10KΩ.....	155	
3.3	Curvas características obtenidas de las termocuplas.	158	
3.3.1	Termocupla tipo J.....	158	
3.3.2	Termocupla tipo K.	162	
3.4	Curvas características obtenidas de las RTDs.	166	

		xi
3.4.1	PT100 2 hilos.	167
3.4.2	PT100 3 hilos.	170
3.4.3	Bland altman de los sensores de temperatura.	173
	LM35.	173
	Termistor 1K Ω	175
	Termistor 5K Ω	177
	Termistor 10K Ω	179
	Termocupla J.	181
	Termocupla K.	183
	PT100.	185

CAPÍTULO IV

4 CONCLUSIONES Y RECOMENDACIONES. 187

4.1 Conclusiones. 187

4.2 Recomendaciones. 190

REFERENCIAS BIBLIOGRÁFICAS 192

ANEXOS 196

Anexo A: Glosario De términos

Anexo B: Hoja de especificaciones técnicas

Anexo C: Programación simulink matlab

Anexo D: Programación labview red ethernet

Anexo E: Programación android studio touch screen

Anexo F: Programación android studio dispositivo móvil

Anexo G: Circuito

Anexo H: Estructura externa e interna

Anexo I: Imagenes del proceso

Anexo J: Hojas técnicas

ÍNDICE DE TABLAS

Tabla 1	Parámetros del controlador.....	13
Tabla 2	Lenguajes de programación eclipse	29
Tabla 3	Toolkits de matlab	30
Tabla 4	Estructura de la pila de software del S.O android	39
Tabla 5	Sistema iOS	42
Tabla 6	Modelo de software en windows phone.....	45
Tabla 7	Clasificación de los tipos de ethernet.....	47
Tabla 8	Descripción de pines MAX31855	71
Tabla 9	Voltaje vs temperatura sensor LM35.....	145
Tabla 10	Resistencia vs temperatura del termistor 1K.....	149
Tabla 11	Resistencia vs temperatura del termistor 5K.....	152
Tabla 12	Resistencia vs Temperatura del Termistor 10K	155
Tabla 13	Voltaje vs temperatura termocupla J.....	159
Tabla 14	Voltaje vs temperatura termocupla K	163
Tabla 15	Resistencia vs temperatura PT100 2hilos.....	167
Tabla 16	Resistencia vs temperatura PT100 3hilos.....	170
Tabla 17	Prueba bilateral LM35.....	173
Tabla 18	Prueba bilateral termistor 1K Ω	175
Tabla 19	Prueba bilateral termistor 5K Ω	177
Tabla 20	Prueba bilateral termistor 10K Ω	179
Tabla 21	Prueba bilateral termocupla J	181
Tabla 22	Prueba bilateral termocupla K.....	183
Tabla 23	Prueba bilateral PT100	185

ÍNDICE DE FIGURAS

Figura 1	Dispositivo peltier.....	7
Figura 2	Esquema módulo peltier	8
Figura 3	Diagrama de bloques de un control PID	13
Figura 4	Acción proporcional, integral, derivativo	14
Figura 5	LM35CH.....	18
Figura 6	Conexión LM35CH.....	18
Figura 7	Curva característica LM35CH.....	19
Figura 8	Dispositivo termo resistivo	22
Figura 9	Conexión RTD 2 hilos	23
Figura 10	Conexión RTD 3 hilos	24
Figura 11	Conexión RTD 4 hilos	24
Figura 12	Conexión de termocupla	26
Figura 13	Conexión cable compensado y termocupla	26
Figura 14	Conexión temperatura ambiente.....	28
Figura 15	Tarjeta SMT32F407	33
Figura 16	Ethernet AMP - detección de colisión	46
Figura 17	Diagrama de red wifi	49
Figura 18	Termistor 1K Ω	51
Figura 19	Termistor 5K Ω	52
Figura 20	Termistor 10K Ω	52
Figura 21	Termocupla J	53
Figura 22	Termocupla K	53
Figura 23	LM35.....	54

		xiv
Figura 24	LM35CH.....	54
Figura 25	PT100	55
Figura 26	Generador de temperatura	56
Figura 27	Resistencia eléctrica (niquelina)	57
Figura 28	Módulo peltier	59
Figura 29	Sistema de refrigeración	60
Figura 30	Disipador de calor	61
Figura 31	Diagrama de bloques de lazo cerrado	62
Figura 32	Ingreso al software matlab.....	63
Figura 33	Ingreso ventana simulink	63
Figura 34	Ventana principal de simulink	64
Figura 35	Selección de dispositivos de simulink	65
Figura 36	Diagrama de bloques del control PID	65
Figura 37	Límite de saturación.....	66
Figura 38	Diagrama eléctrico MAX31865	67
Figura 39	Conexión RTD de 4 hilos	68
Figura 40	Conexión RTD de 3 hilos	68
Figura 41	Conexión RTD de 2 hilos	69
Figura 42	Configuración MAX31855	70
Figura 43	Diagrama de bloques MAX31855	70
Figura 44	Diagrama de bloque STM32F407VG.....	71
Figura 45	Configuración de la tarjeta STM32F407VG	72
Figura 46	Adquisición datos (RTD-termocuplas)	72
Figura 47	Configuración protocolo SPI	73
Figura 48	Comunicación arduino	74

		xv
Figura 49	Comunicación tablet	74
Figura 50	Variables globales.....	75
Figura 51	Recolección de datos.....	75
Figura 52	Recolección de datos (PT100-termocuplas).....	76
Figura 53	Contador (a).....	77
Figura 54	Contador (b).....	77
Figura 55	Contador (c).....	78
Figura 56	Diagrama de activación de PT100 uno	79
Figura 57	Configuración de la PT100 uno	80
Figura 58	Diagrama de activación PT100 dos	80
Figura 59	Diagrama de activación PT100 tres	81
Figura 60	Diagrama de activación termocupla J2.....	82
Figura 61	On termocuplas.....	82
Figura 62	Diagrama de activación termocupla K1	83
Figura 63	Diagrama de activación termocupla K2	84
Figura 64	Diagrama de activación termocupla J1	84
Figura 65	Contador (a).....	85
Figura 66	Contador (b).....	86
Figura 67	Contador (c).....	86
Figura 68	Contador (d).....	87
Figura 69	Contador (e).....	87
Figura 70	Configuración de datos PT100 uno.....	88
Figura 71	Diagrama de bloques para obtener la temperatura uno.....	89
Figura 72	Configuración de datos PT100 dos.....	89
Figura 73	Configuración de datos termocupla J1.....	90

		xvi
Figura 74	Complemento A2	91
Figura 75	Diagrama de bloques de datos termocupla J1	91
Figura 76	Configuración de datos termocupla J2.....	92
Figura 77	Configuración de datos termocupla K1	93
Figura 78	Configuración de datos termocupla K2	93
Figura 79	Configuración de datos temperatura control	94
Figura 80	Envío de datos PT10 - termocuplas.....	95
Figura 81	Conexión MCP6004.....	96
Figura 82	Adquisición de datos del LM35	96
Figura 83	Adquisición de datos termistores	97
Figura 84	Matlab function termistores.....	98
Figura 85	Cálculo temperatura termistor (5k).....	99
Figura 86	Cálculo temperatura termistor (1k).....	99
Figura 87	Cálculo temperatura termistor (10k).....	99
Figura 88	Diagrama de bloques de los sensores uno.....	100
Figura 89	Diagrama de bloques de los sensores dos	101
Figura 90	Datos de resistencia y voltaje sensores uno.....	102
Figura 91	Datos de resistencia y voltaje sensores dos	103
Figura 92	Temperatura a resistencia PT100.....	104
Figura 93	Temperatura a voltaje termocupla J1.....	104
Figura 94	Temperatura a voltaje termocupla k1.....	104
Figura 95	Temperatura a voltaje LM35	104
Figura 96	Transmisión STM32F407VG hacia touch screen.....	105
Figura 97	Recepción tarjeta desde touch screen.....	106
Figura 98	Recepción tipo de sensor	106

Figura 99	Recepción set point	106
Figura 100	Inicio android studio	107
Figura 101	Configura un nuevo proyecto	108
Figura 102	Clase de actividad.....	109
Figura 103	Configuración del proyecto	110
Figura 104	Interfaz usuario android studio.....	110
Figura 105	Código para comunicación USB serial.....	111
Figura 106	Asignación de variables	112
Figura 107	Inicializar recepción de datos.....	113
Figura 108	Instanciar variables globales.....	114
Figura 109	Instanciar gráficas.....	115
Figura 110	Timer (500ms).....	115
Figura 111	Códigos para graficar.....	116
Figura 112	Interfaz gráfica	116
Figura 113	Instrucciones botón PT100	117
Figura 114	Instrucciones botones termocuplas.....	118
Figura 115	Instrucciones botón LM35.....	118
Figura 116	Instrucciones botón termistores	119
Figura 117	Instrucciones botones inicio y detener	120
Figura 118	Texto (SP) y botón EN.....	121
Figura 119	Códigos arreglo de vectores para graficar	121
Figura 120	Activar/desactivar servidor	122
Figura 121	Abrir/cerrar el puerto serial	123
Figura 122	Funciones puerto serial.....	124
Figura 123	Recepción datos de variables globales touch screen	125

Figura 124	Switch selección sensores	126
Figura 125	Tx STM32F407VG a dispositivo móvil	127
Figura 126	Transmisión datos wifi	128
Figura 127	Habilitar servidor	130
Figura 128	Función while	130
Figura 129	Transmisión STM32F407VG hacia LabVIEW	131
Figura 130	Inicialización SPI ethernet.....	132
Figura 131	Servidor ethernet	132
Figura 132	Cliente ethernet.....	133
Figura 133	Ingreso software LabVIEW	134
Figura 134	Crear un nuevo proyecto en LabVIEW	134
Figura 135	Crear un proyecto en blanco en LabVIEW	135
Figura 136	Ventana del proyecto	135
Figura 137	Creación del VI	136
Figura 138	Dirección de los datos guardados.....	139
Figura 139	Reporte de datos - curva característica	140
Figura 140	Diagrama de bloques HMI PC por red ethernet.....	141
Figura 141	HMI PC por red ethernet.....	141
Figura 142	Diagrama de bloques.....	142
Figura 143	P&ID del sistema	142
Figura 144	Diagrama eléctrico.....	143
Figura 145	Curvas características LM35 1 y LM35 2 (LabVIEW)	146
Figura 146	Curvas características LM35 1 y LM35 2 (excel)	147
Figura 147	Curvas características termistor 1K (1 y 2) (LabVIEW).....	150
Figura 148	Curvas características termistor 1K (1 y 2) (excel)	151

Figura 149	Curvas Características Termistor 5K (1 y 2) (LabVIEW).....	153
Figura 150	Curvas características termistor 5K (1 y 2) (excel)	154
Figura 151	Curvas características termistor 10K (1 y 2) (LabVIEW).....	156
Figura 152	Curvas características termistor 10K (1 y 2) (excel)	157
Figura 153	Curvas características termocupla J (1 y 2) (LabVIEW)	160
Figura 154	Curvas características termocupla J (1 y 2) (excel)	161
Figura 155	Curvas características termocupla k (1 y 2) (LabVIEW)	164
Figura 156	Curvas características termocupla k (1 y 2) (excel)	165
Figura 157	Curvas características PT100 2hilos (1 y 2) (LabVIEW).....	168
Figura 158	Curvas características PT100 2hilos (1 y 2) (excel).....	169
Figura 159	Curvas características PT100 3hilos (1 y 2) (LabVIEW).....	171
Figura 160	Curvas características PT100 3hilos (1 y 2) (excel).....	172
Figura 161	Temperatura vs tiempo LM35	173
Figura 162	Prueba bilateral LM35.....	174
Figura 163	Bland altman LM35.....	174
Figura 164	Temperatura vs tiempo termistor 1K Ω	175
Figura 165	Prueba bilateral termistor 1K Ω	176
Figura 166	Bland altman termistor 1k Ω	176
Figura 167	Temperatura vs tiempo termistor 5K Ω	177
Figura 168	Prueba bilateral termistor 5K Ω	178
Figura 169	Bland altman termistor 5K Ω	178
Figura 170	Temperatura vs tiempo termistor 10K Ω	179
Figura 171	Prueba bilateral termistor 10K Ω	180
Figura 172	Bland altman termistor 10k Ω	180
Figura 173	Temperatura vs tiempo termocupla J.....	181

		xx
Figura 174	Prueba bilateral termocupla J	182
Figura 175	Bland altman termocupla J	182
Figura 176	Temperatura vs tiempo termocupla K.....	183
Figura 177	Prueba bilateral termocupla K.....	184
Figura 178	Bland altman termocupla K.....	184
Figura 179	Temperatura vs tiempo PT100.....	185
Figura 180	Prueba bilateral PT100	186
Figura 181	Bland altman PT100	186

ÍNDICE DE ECUACIONES

Ecuación 1	Control ON	9
Ecuación 2	Control OFF	9
Ecuación 3	Control proporcional	10
Ecuación 4	Control Integral.....	10
Ecuación 5	Control Derivativo.....	11
Ecuación 6	Control proporcional integral	11
Ecuación 7	Control proporcional derivativo.....	12
Ecuación 8	Control PID.....	14
Ecuación 9	Potencia uno	58
Ecuación 10	Potencia dos	58

RESUMEN

El presente proyecto consiste en el diseño y construcción de un módulo generador de temperatura controlada entre 0 y 200°C, para ilustración de curvas características mostradas en una TFT Touch Screen (LCD gráfico), PC y Dispositivos Móviles utilizando señales de sensores de temperatura PT100 2, 3, 4 hilos, Termistor 1KΩ, 5KΩ, 10KΩ, Termocupla tipo J y K, LM35. Para la construcción del generador de temperatura, está dado por un módulo Peltier (Generador de temperaturas bajas), sistema de refrigeración, y una níquelina (Generador de temperaturas altas). Los datos de los diferentes sensores de temperatura lo adquirimos mediante el software de Matlab – Simulink los mismos que son enviados a la tarjeta STM32F407VG, que por medio del puerto serial de la tarjeta estos datos son visualizados en el HMI de Touch Screen que está programada en el software de Android Studio. De tal manera para la visualización de las curvas características de los sensores en la PC está desarrollado en el software de LabVIEW por medio de una Red Ethernet, y para dispositivos móviles está dado por comunicación Wifi utilizando el software de Android Studio.

PALABRAS CLAVE:

- **GENERADOR DE TEMPERATURA**
- **RED ETHERNET**
- **DISPOSITIVOS MÓVILES**
- **SISTEMA DE CONTROL**
- **SISTEMA DE REFRIGERACIÓN**

ABSTRACT

This project involves the design and construction of a generator module temperature controlled from 0 to 200 ° C, for illustration of curves displayed on a TFT Touch Screen (graphic LCD), PC and mobile devices using signals from temperature sensors PT100 2, 3, 4 wire, Thermistor 1k, 5K Ω , 10k, J and K type thermocouple, LM35. To build the generator temperature, it is given by a Peltier module (low temperature generator), the cooling system, and niquelina (high temperature generator). The data from the various temperature sensors acquired it through software Matlab - Simulink them that are sent to the STM32F407VG card, via the serial port on the card this data is displayed on the HMI touch screen that is scheduled in Studio Android software. So for displaying the characteristic curves of the sensors on the PC it is developed in LabVIEW software via an Ethernet network, and mobile wireless communication is given by using the Android software Studio.

KEYWORDS:

- **GENERATOR TEMPERATURE**
- **ETHERNET NETWORK**
- **MOBILE DEVICES**
- **CONTROL SYSTEM**
- **COOLING SYSTEM**

CAPÍTULO I

1 DEFINICIONES GENERALES

1.1 Antecedentes.

Desde hace varios años, el control de temperatura e identificación de curvas características de sensores dedicados a la obtención de datos de esta variable, se han venido utilizando en innumerables aplicaciones de control y con fines académicos.

Los avances en diferentes ámbitos de la tecnología han permitido al ser humano tener el control de diferentes variables físicas, siendo la temperatura una de las más involucradas en procesos, tanto académicos como industriales; aparte de esto el hombre ha buscado conocer el comportamiento mediante curvas características de distintos sensores de temperatura, instrumentos que son utilizados para el control de dicha variable.

Gracias al apresurado crecimiento tecnológico en instrumentación y control ha provocado un impactado muy importante en la utilización de sensores y otros instrumentos que permite el control de la temperatura, obteniendo así importante información en los aspectos más significativos que posee cada sensor, siendo de gran beneficio para el estudiante al momento de aplicar sus conocimientos en sistemas de control.

Las soluciones electrónicas se han convertido en indispensables para muchos sectores. Su éxito económico no descansa solamente en la optimización de los subsistemas sino en el acercamiento a los sistemas. En la automatización industrial orientada al control de temperatura, una máquina o una planta de producción son vistos no solo en términos mecánicos sino como un sistema completo integrando componentes mecánicos, electrónicos, ingeniería de control y tecnología software.

En conclusión, una rápida evolución tecnológica ha permitido en los últimos años, la creación de innovadores controles de temperatura y la

ilustración de sus curvas características, lo que ha permitido crear sistemas con mayores facilidades de uso y control.

1.2 Planteamiento del problema.

En el Laboratorio de Circuitos Electrónicos de la Universidad De Las Fuerzas Armadas ESPE Extensión Latacunga, unos de los sitios más utilizados por los estudiantes para realizar sus prácticas, es de mucha importancia un generador de temperatura.

Vista esta necesidad se ha diseñado un control de temperatura de 0 a 200°C para mantener estable el valor en cada una de las mediciones, que permitirá graficar la curva características de distintos tipos de sensores de temperatura objeto de este proyecto.

Es por esta razón que el diseño del equipo se realiza a partir de un control de temperatura, la misma que será generada en valores estables, destinados a los sensores de temperatura, con los cuales obtendremos señales que nos permitirán graficar la curva característica de cada uno de ellos, esta será mostrada en una TFT Touch Screen, también tendremos una comunicación Ethernet, Wifi dirigida a un computador personal o a un dispositivo móvil respectivamente.

Planteamiento: Mejorar el estudio de los sensores de temperatura mediante un proceso para obtener las curvas características, en un rango estable de temperatura de 0 a 200°C.

Beneficios: Obtener mejor conocimiento técnico de cada sensor de temperatura como: PT100 2, 3, 4 hilos, Termistor 1KΩ, 5KΩ, 10KΩ, Termocupla tipo J y K, LM35, y visualización de la curva característica de los sensores a utilizar.

Este proyecto por su alto nivel de conocimiento servirá de gran ayuda en el desarrollo del estudiante porque obtendrá mayor información del comportamiento de los diferentes sensores de temperatura gracias a las curvas características.

1.3 Descripción resumida del proyecto.

El proyecto tiene como finalidad la construcción de un módulo generador de temperatura controlada entre 0 y 200°C, para ilustración de curvas características mostradas en una TFT Touch Screen (LCD gráfico), PC y Dispositivos Móviles utilizando señales de sensores de temperatura PT100 2, 3, 4 hilos, Termistor 1K Ω , 5K Ω , 10K Ω , Termocupla tipo J y K, LM35.

Se realizará un análisis del Software de programación del microcontrolador, utilizaremos el programa "BASCOS AVR", como también del Hardware para la construcción del generador de temperatura, en este caso emplearemos, un módulo Peltier (Generador de temperaturas bajas), incluiremos una niquelina (Generador de temperaturas altas) y de los distintos elementos que serán adquiridos para este proyecto.

Posteriormente se efectuará el diseño del módulo, que abarque como característica una apropiada distribución del hardware a ser instalado, logrando así que sus componentes sean fácilmente identificados y manipulados, para luego implementarlo físicamente.

Finalmente se desarrollará la configuración y programación de los diferentes dispositivos para el control de temperatura y HMI, en el mismo que se podrá controlar y manipular la temperatura deseada; así mismo se mostrará la curva característica correspondiente al sensor y por último el desarrollo de manuales de operación.

1.4 Justificación e importancia.

Durante el proceso de pasantía desarrollado en la empresa PRODUCTOS FAMILIA SANCELA DEL ECUADOR S.A., se nos propuso un proyecto, el que consistía en el diseño y construcción de un módulo industrial para comprobación de diferentes tipos de sensores de temperatura, en un rango de 20°C a 500°C, sin embargo el presupuesto no se asignó, por esta razón propusimos un prototipo para el laboratorio de Circuitos Electrónicos donde los estudiantes podrán acudir para obtener

información de sensores de temperatura más comunes para sus prácticas y diseño de sistema de control.

Con la implementación de este proyecto se pretende cubrir las necesidades básicas de manejo de información de los sensores de temperatura PT100 2, 3, 4 hilos, Termistor 1K Ω , 5K Ω , 10K Ω , Termocupla tipo J y K, LM35 con la visualización de las curvas características en un rango de 0 a 200 °C en una TFT Touch Screen, PC y dispositivos móviles por parte de los usuarios, con el fin de maximizar la confiabilidad del sensor de temperatura.

Mediante el proyecto propuesto en este documento se pretende realizar un aporte a los usuarios de la Universidad De Las Fuerzas Armadas “ESPE” extensión Latacunga, dando información adecuada de sensores de temperatura más comunes, gracias a la visualización de la curva característica de cada uno de ellos, para obtener resultados más confiables en un proceso de monitoreo o control.

1.5 Objetivos generales y específicos.

1.5.1 Objetivo general.

- Diseñar e implementar un prototipo de módulo para visualizar por medio de Touch Screen, PC y Dispositivos Móviles curvas características de sensores de temperatura, en el rango de 0 a 200 °C, para el laboratorio de Circuitos Electrónicos de la Universidad de las Fuerzas Armadas ESPE extensión Latacunga.

1.5.2 Objetivos específicos.

- Investigar las características técnicas de los sistemas de control para generar temperaturas de 0 a 200°C.
- Analizar los métodos y/o técnicas utilizados para la adquisición gráfica de las señales de sensores de temperatura PT100 2, 3, 4 hilos, Termistor 1K Ω , 5K Ω , 10K Ω , Termocupla tipo J y K, LM35.
- Analizar repuestos de los sensores de temperatura de acuerdo a las curvas características.

- Conocer y analizar las tecnologías de desarrollo de software para adquirir las curvas características de los sensores de temperatura en una TFT.
- Diseñar e implementar un generador de temperatura en incrementos para un rango de medición de 0 a 200 °C con valores estables.
- Comprobar y analizar en una TFT las diferentes curvas características en un rango de 0 a 200 °C de los sensores de temperatura obtenidos en el proyecto realizado.

1.6 Introducción.

La temperatura es un concepto abstracto, el cual se explica por su efecto en las condiciones del medio ambiente, los objetos y sus propiedades, en general se relaciona con el comportamiento de la materia y en la mayoría de los casos define el estado final de ésta.

Popularmente la temperatura es relacionada a los conceptos de frío y calor. Algo es más caliente si presenta una mayor temperatura, o está más frío si se presenta una disminución en la temperatura. Sin embargo, aunque este concepto es común y aceptable, en la realidad la física define a la temperatura como: “una magnitud escalar relacionada con la energía interna de un sistema termodinámico, definida por el principio cero de la termodinámica, o conocida como la energía sensible, que es la energía asociada a los movimientos de las partículas de ese sistema termodinámico”, esto quiere decir que la temperatura es la cuantificación de la actividad molecular de la materia.

La temperatura es una de las variables físicas que se miden con mayor frecuencia. De acuerdo a la tarea de medición a desarrollar, se debe determinar el tipo adecuado de sensor, teniendo en consideración los principales criterios de selección: rango de medición, exactitud requerida, tiempo de respuesta, accesibilidad del objeto a medir, funciones deseadas (memoria, registro de datos, impresión, etc.), teniendo en cuenta que la temperatura puede medirse de numerosas maneras y con una gran variedad

de elementos, en este documento se estudiará las características de los sensores de uso más común en instrumentación y control de procesos.

1.7 Descripción del generador de temperatura.

Para la generación de calor se utiliza módulos Peltier, teniendo en cuenta que un módulo termoeléctrico o enfriador Peltier es un componente electrónico basado en un semiconductor que funciona como una pequeña bomba de calor. Aplicándole una baja tensión DC, el calor será movido a través del módulo de un lado al otro, es decir, una cara del módulo será enfriada mientras que la otra será calentada simultáneamente. Es importante destacar que este fenómeno es reversible, cambiándole la polaridad de la fuente. Como consecuencia, un Peltier puede usarse para calentar o enfriar, con una gran precisión en la temperatura deseada.

Los Peltier y los refrigeradores mecánicos están gobernados por la misma ley termodinámica. En un refrigerador mecánico, el compresor genera la presión en un líquido refrigerante y la hace circular a través del sistema. En el evaporador o "freezer" el refrigerante hierve absorbiendo energía del medio, causando enfriamiento. El calor absorbido en el evaporador es movido al condensador en el cual el líquido se condensa, liberando la energía que había absorbido en el evaporador. En un Peltier, un semiconductor dopado, esencialmente toma el lugar del líquido refrigerante, el condensador es remplazado por un disipador de calor, y el compresor por una fuente DC. [1]

1.8 Dispositivo peltier.

Es un dispositivo semiconductor que actúa como una bomba de calor, bombeando el calor desde un lado hacia otro de la celda, según la polaridad con que se la conecte. El lado que produce calor es utilizado con un disipador para evitar que se queme, un ejemplo de este dispositivo se muestra en la Figura 1 [2].

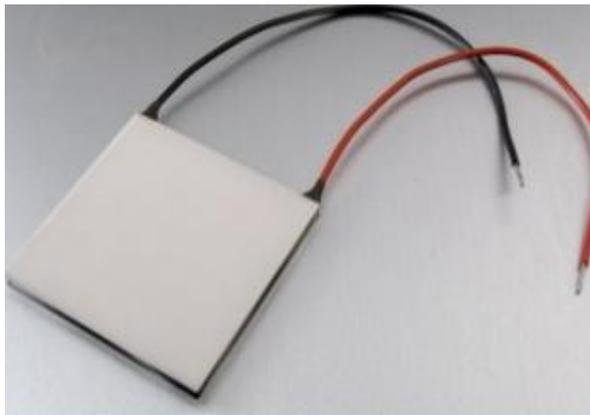


Figura 1 Dispositivo peltier

Fuente: [2]

1.8.1 Funcionamiento.

Cuando se aplica un voltaje de corriente directa a un dispositivo termoeléctrico, se crea una diferencia de temperatura esto facilita la transferencia térmica del lado caliente al frío, induciendo así una corriente térmica; también conocida como el efecto de Peltier.

Este efecto puede utilizarse para cambiar la temperatura de un cierto proceso. En otras palabras, un dispositivo Peltier es una bomba termoeléctrica de calor: cuando la corriente directa circula a través, calor o frío es creado por el movimiento de los electrones. Por lo tanto, puede utilizarse el dispositivo Peltier para calefacción o enfriamiento como un regulador de temperatura.

Las principales ventajas de un dispositivo Peltier son la falta de líquido en el sistema de enfriamiento, la minimización de partes móviles, así como su tamaño pequeño y flexible forma aseguran múltiples opciones de instalación. [3]

1.8.2 Esquema de un Módulo Peltier.

Un dispositivo Peltier consiste en una unión PN. En la cara fría los electrones pasan de un estado de menor energía (en el semiconductor tipo P) a uno de mayor energía (en el de tipo N), en la cara caliente los electrones pasan de un estado de mayor energía a uno de menor energía emitiendo calor en el proceso. De esta forma actúa como una bomba de

calor cuya parte fría o caliente puede invertirse cambiando la polaridad, observe **Figura 2**.

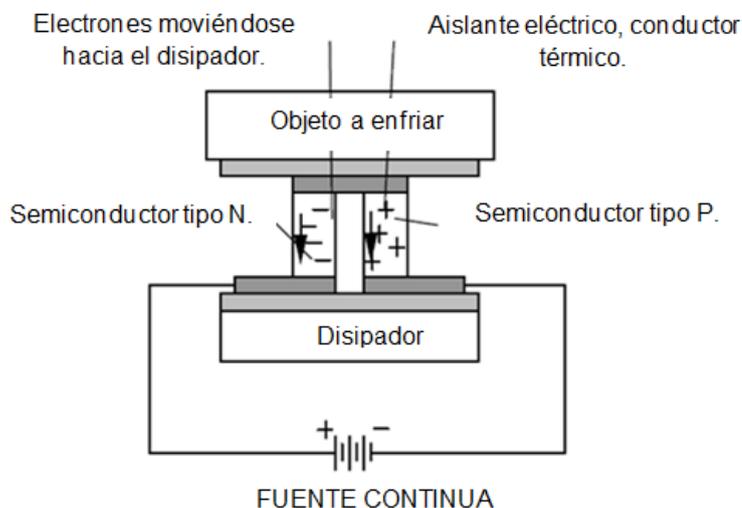


Figura 2 Esquema módulo peltier

Fuente: [3]

1.9 Módulo térmico o mufla.

Una mufla es un tipo de horno que puede alcanzar temperaturas muy altas para cumplir con los diferentes procesos.

Las muflas han sido diseñadas para una gran variedad de aplicaciones dentro de un laboratorio y pueden realizar trabajos como: procesos de control, tratamientos térmicos y secado de precipitados.

1.9.1 Tipos de muflas.

Las muflas cuentan con dos principales tipos, las cuales cubren perfectamente la necesidad que se requiera:

- Una mufla de combustible alcanza temperaturas muy elevadas, pues su fuente de calor está separada totalmente de la cámara de cocción, de tal manera que una muestra no puede ser contaminada con gases de combustión.
- Las muflas eléctricas son aquellas que cuentan con hornos generalmente pequeños con resistencias calefactoras ocultas.

Estas son ampliamente utilizadas en laboratorios, talleres pequeños o consultorios dentales. [4]

1.10 Tipos de sistemas de control.

Existen diferentes tipos de sistemas de control, estos pueden ser de:

- Control ON-OFF.
- Control Proporcional.
- Control Integral.
- Control Derivativo.
- Control Proporcional Integral.
- Control Proporcional Derivativa.
- Control Proporcional Integral Derivativa.
- Control Adaptivo.
- Control Neuronal.
- Control Difuso.
- Control Avanzado.

1.10.1 Control ON-OFF.

Modo de control en el cual la salida del controlador va de un extremo a otro cuando el valor de la variable controlada se desvía del valor deseado.

$$y(t) = M1(ON) \quad e > 0$$

Ecuación 1 Control ON

$$y(t) = M2(OFF) \quad e < 0$$

Ecuación 2 Control OFF

Se observa que la acción On-OFF solo depende del signo del error.

1.10.2 Control proporcional.

La acción de control proporcional, da una salida del controlador que es proporcional al error. Un controlador proporcional puede controlar cualquier sistema estable, pero posee desempeño limitado y error en régimen permanente (off-set).

$$u(t) = k_p * e(t)$$

Ecuación 3 Control proporcional

Donde $u(t)$ es la señal de control, k_p es una ganancia proporcional ajustable y $e(t)$ es la señal del error.

A continuación se describe algunas características del control proporcional:

- Simple.
- Fácil de sintonizar (un solo parámetro.)
- Puede reducir, pero no eliminar, el error en estado estacionario.

1.10.3 Control integral.

Señal de control proporcional a la integral del error:

$$u(t) = k_i \int_0^t e(t) dt$$

Ecuación 4 Control Integral

Donde $u(t)$ es la señal de control, k_i es una ganancia integral ajustable y $e(t)$ es la señal del error.

Normalmente se aplica conjuntamente con una acción proporcional, formando un PI, para encontrar un compromiso entre el transitorio y el permanente de la respuesta, teniendo en cuenta las siguientes características:

- Elimina errores estacionarios.
- Más del 90% de los lazos de control utilizan PI.
- Puede inestabilizar al sistema si T_i (tiempo integral) disminuye mucho.

1.10.4 Control derivativo.

El control derivativo es un ajuste del proceso basado en la tasa de cambio de la corriente del error de control del proceso.

El control derivativo se usa típicamente en los casos donde hay un gran retraso temporal entre el dispositivo controlado y el sensor usado en la realimentación. Este término tiene el efecto total de prevenir que la señal del actuador vaya muy lejos en una dirección u otra, y puede usarse para excederse en la limitación.

$$u(t) = K_d \frac{de(t)}{dt}$$

Ecuación 5 Control Derivativo

Donde $u(t)$ es la señal de control, kd es una ganancia derivativa ajustable y $e(t)$ es la señal del error.

1.10.5 Control proporcional integral.

En un controlador proporcional integral, la acción de control se define mediante:

$$u(t) = ki \int_0^t e(t) dt + \frac{kp}{Ti} \int_0^t e(t) dt$$

Ecuación 6 Control proporcional integral

Donde $u(t)$ es la señal de control, ki es una ganancia integral ajustable, kp es una ganancia proporcional ajustable, Ti es el tiempo integral y $e(t)$ es la señal del error.

El tiempo integral da idea del tiempo que tarda la respuesta temporal en alcanzar el permanente.

Mejora el régimen permanente, ya que el controlador aumenta el tipo del sistema en bucle abierto.

Efecto similar al proporcional en el transitorio.

1.10.6 Control proporcional derivativo.

En un controlador proporcional derivativo, la acción de control se define mediante:

$$u(t) = k_p * e(t) + K_p T_d \frac{de(t)}{d(t)}$$

Ecuación 7 Control proporcional derivativo

Donde $u(t)$ es la señal de control, k_p es una ganancia proporcional ajustable, T_d es el tiempo derivativo y $e(t)$ es la señal del error. Este control presenta las siguientes características:

- Mantener la ganancia y el tiempo integral
- Aumentar el T_d hasta obtener características similares pero con la respuesta más rápida.
- Aumentar ligeramente la ganancia si fuera necesario.

1.10.7 Control proporcional integral derivativo.

El control PID es un mecanismo de control que a través de un lazo de retroalimentación permite regular la velocidad, temperatura, presión y flujo entre otras variables de un proceso en general. El controlador PID calcula la diferencia entre nuestra variable real contra la variable deseada.

Para el funcionamiento del controlador PID combina en un único controlador la mejor característica de estabilidad del controlador PD (Proporcional Derivativo) con la ausencia de error en estado estacionario del controlador PI (Proporcional Integral).

La adición de la acción integral a un controlador PD es esencialmente lo mismo que añadir dicha acción a un controlador Proporcional.

La Tabla 1 muestra cómo varían la estabilidad, la velocidad y el error en estado estacionario cuando se modifican los parámetros del controlador. [5]

Tabla 1

Parámetros del controlador

	Kp Aumenta	Ki Disminuye	Td Aumenta
ESTABILIDAD	Reduce	Disminuye	Aumenta
VELOCIDAD	Aumenta	Aumenta	Aumenta
ERROR ESTADO ESTACIONARIO	No eliminado	Eliminado	No eliminado

Fuente: [5]

En la **Figura 3**, se observa el esquema básico de un control PID. [6]

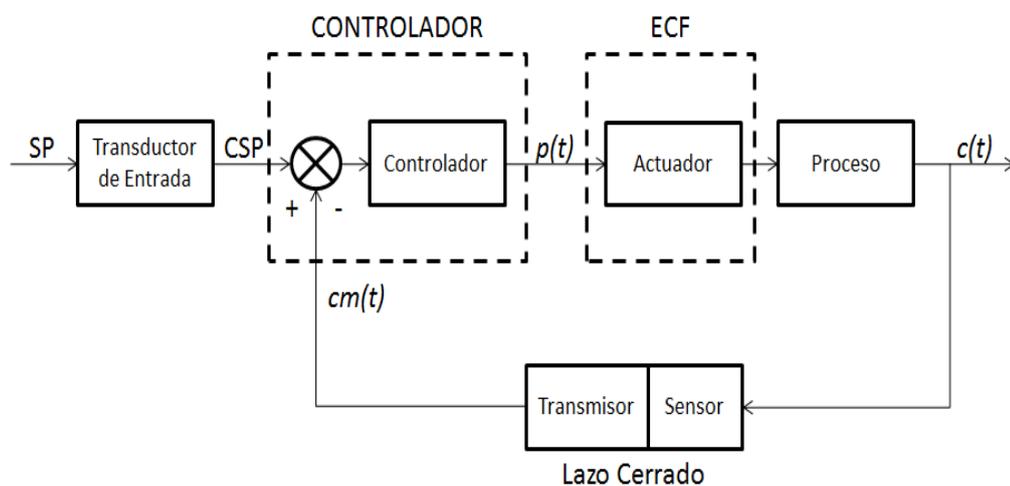


Figura 3 Diagrama de bloques de un control PID

Fuente: [6]

$c(t)$: Variable controlada.

ECF: Elemento de control final.

$p(t)$: Parámetro de control.

$cm(t)$: Variable medida.

CSP: Consigna.

SP: Set point.

El control PID combina las tres acciones que se indican a continuación:

1. P (proporcional).
2. I (Integral).
3. D (Derivativo).

En la Figura 4, se muestra la señal de la acción proporcional, integral y derivativa.

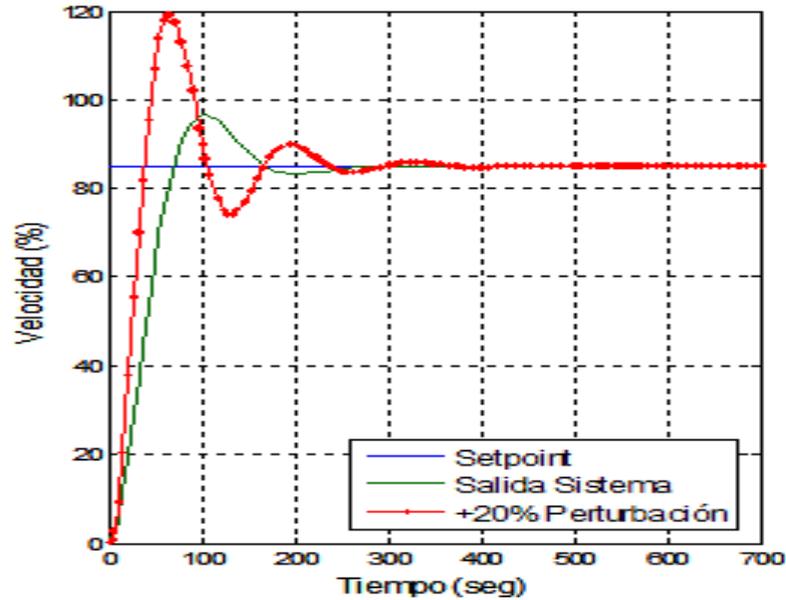


Figura 4 Acción proporcional, integral, derivativo

Fuente: [6]

Controlador PID Continuo:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt}$$

Ecuación 8 Control PID

Donde $u(t)$ es la señal de control, k_p es una ganancia proporcional ajustable, T_i es el tiempo integral, T_d es el tiempo derivativo y $e(t)$ es la señal del error.

1.10.8 Control adaptivo.

Es una variante del control anticipatorio, en donde la respuesta del controlador varía automáticamente basado en los cambios de las condiciones dentro del proceso, es decir, la respuesta del controlador será variable dependiendo del comportamiento actual del proceso.

Para que se lleve a cabo esta adaptación se requiere de algoritmos matemáticos que simulen el proceso en base a los datos tomados en el instante mismo en que se realiza la acción, este resultado va a generar una señal compensadora que garantizará la confiabilidad del sistema.

1.10.9 Control neuronal.

Hace uso de neuronas de inteligencia artificial. La neurona artificial estándar es un elemento de procesamiento que calcula una salida multiplicando su vector de entradas por un vector de pesos y este resultado es aplicado a una función de activación; un conjunto de neuronas conforman una red neuronal. Las Redes Neuronales son parte de la Inteligencia Artificial (AI) caracterizadas por su capacidad de aprendizaje, su velocidad mediante el procesamiento masivo en paralelo de datos y por la facilidad de modelado de sistemas y controladores no lineales, este control presenta las siguientes características:

- Son dispositivos no-lineales.
- Pueden aprender un mapeo.
- Son adaptables.
- Las respuestas están basadas en evidencia.
- Usan contexto, es decir, a más información, la respuesta es más veloz y mejor.
- Son tolerantes a fallas, o sea, que la falla es degradada.
- Su diseño y análisis es uniforme.
- Tienen analogía neuro-biológica.
- Su procesamiento es masivamente, en paralelo, distribuido y realizado con operaciones individuales simples.
- Aprenden por sí mismas.
- Tienen capacidad de generalizar.

Con el Control Neuronal se pueden realizar las siguientes aplicaciones, las mismas que se las encuentran en diferentes procesos:

- Representación de comportamientos de funciones lineales y no lineales.
- Identificación de patrones o sistemas.
- Sistemas de control.
- Reconocimiento de imágenes.
- Reconocimiento de caracteres.

- Reconstrucción de datos.
- Predicción y recomendación para la toma de decisiones.
- Simulación de modelos económicos y financieros.
- Clasificación de objetos.
- Predicciones de clima.

1.10.10 Control difuso.

En el Control difuso podemos encontrar las características referentes a la lógica difusa:

- Variables lingüísticas
- Conjuntos difusos
- Operaciones difusas
- Relaciones difusas

Todo Control difuso debe tener la siguiente estructura específica:

- Esquemas de control estándar
- Estructura de un controlador difuso
- Interfaz real difuso
- Base de reglas lingüísticas
- Mecanismo de inferencia
- Interfaz difuso real

1.10.11 Control avanzado.

Los métodos de control avanzado son aquellos que actúan en forma preventiva, de modo tal que en base a los datos tomados, actúan de modo tal que previenen la ocurrencia de error, por tanto el controlador está ajustando sus parámetros constantemente.

1.11 Sensores de temperatura.

Los sensores de temperatura son dispositivos que transforman los cambios de temperatura en cambios en señales eléctricas que son procesados por equipo eléctrico o electrónico.

El sensor de temperatura, típicamente suele estar formado por el elemento sensor, la vaina que lo envuelve y que está rellena de un material muy conductor de la temperatura, para que los cambios se transmitan rápidamente al elemento sensor y del cable al que se conectarán el equipo electrónico. [7]

Una descripción de cada tipo de sensores de temperatura se dará a continuación:

- Termistores.
- RTD.
- Termocuplas.
- LM35.

1.11.1 LM35.

La serie LM35 son sensores de tipo semiconductor de precisión de temperatura, cuya tensión de salida es linealmente proporcional a la temperatura en grados Celsius (centígrados).

El LM35 tiene una ventaja sobre los sensores de temperatura lineales calibradas en grados Kelvin, como el usuario no está obligado a sustraer una gran tensión constante, a partir de su salida puede obtener un escalamiento conveniente en grados centígrados.

El LM35 no requiere ninguna calibración externa o recorte para proporcionar una precisión típica de $\pm 1/4$ ° C a la temperatura ambiente y $\pm 3/4$ ° C durante un total intervalo de temperatura de -55 a 150 ° C.

La impedancia de salida del LM35 es baja, lineal y precisa calibración inherente, hacer interfaz con la lectura o circuitos de control especialmente fácil. Se puede usar con fuentes de alimentación individuales, o con suministros más y menos estables y tiene muy bajo auto calentamiento, menos de 0.1 ° C. [8]

En la **Figura 7**, se muestra la curva característica del LM35CH.

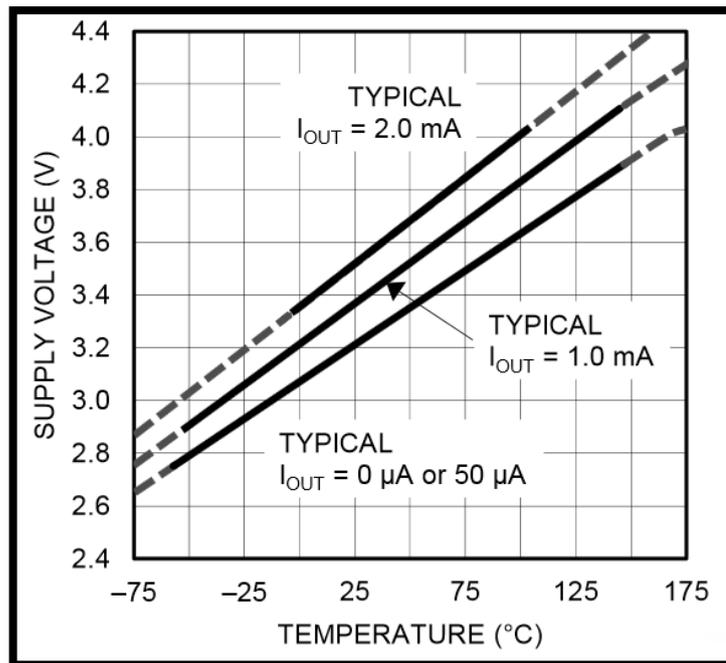


Figura 7 Curva característica LM35CH

Fuente: [9]

1.11.2 Termistor.

Los termistores son resistores variables con la temperatura, que están basados en semiconductores. Existen dos tipos de termistores, dependiendo de si su coeficiente de temperatura es negativo o positivo. Si es negativo se denominan NTC (las cuales se fabrican a base de mezclar y sinterizar óxidos dopados de metales como el níquel, cobalto, manganeso, hierro y cobre), y si es positivo se denominan PTC (basadas en titanato de bario al que se añade titanato de plomo o de circonio para determinar la temperatura de conmutación).

El funcionamiento de un termistor se basa en la variación de la resistencia de un semiconductor con la temperatura, debido a la variación de la concentración de portadores. Para los termistores NTC, al aumentar la temperatura, aumentará también la concentración de portadores, por lo que la resistencia será menor, de ahí que el coeficiente sea negativo. Para los termistores PTC, en el caso de un semiconductor con un dopado muy

intenso, éste adquirirá propiedades metálicas, tomando un coeficiente positivo en un margen de temperatura limitado.

Sus principales características son:

- Su rango de temperaturas esta entre -50°C y 150°C , aunque las unidades encapsuladas pueden alcanzar hasta los 300°C .
- En la mayoría de las aplicaciones para una temperatura de 25°C la resistencia varía entre 100 ohm y 100Kohm.
- Tienen un tamaño reducido que hacen que la repuesta a los cambios de temperatura sea rápida (tienen mayor sensibilidad a los cambios de temperatura que otros transconductores).
- Son autocalentables, lo que hace que puedan ser indeseables en algunas aplicaciones, y que otras basen su funcionamiento en ese fenómeno.
- Gracias a la intercambiabilidad (tolerancia con la que es producido un termistor), es posible cambiar un termistor por otro en un sistema, sin necesidad de volver a calibrar el aparato de medida.

Existen varios tipos de configuraciones para los termistores. Éstos son: los tipo perla, tipo disco, tipo chip, tipo arandela y tipo barra. Los termistores tipo perla con cubierta de cristal se caracterizan por tener una excelente estabilidad y fiabilidad a temperaturas superiores a 300°C . Los termistores tipo chip y disco tienen un tamaño mayor que los de perla, lo que permite una potencia de disipación mayor, a expensas, eso sí, de peores tiempos de respuesta. Por su geometría, los termistores de disco suelen tener más potencia de disipación que los chips.

Los termistores ofrecen mayores ventajas que las termoresistencias en cuanto a economía, sensibilidad y velocidad de respuesta. Pero hay que superar en ellos los inconvenientes provocados por la alta no linealidad. [10]

Termistor NTC: Son fabricados a partir de los óxidos de metales de transición (manganeso, cobalto, cobre y níquel) los termistores NTC son semiconductores dependientes de la temperatura. Operan en un rango de -

200° C a + 1000° C. Un termistor NTC debe elegirse cuando es necesario un cambio continuo de la resistencia en una amplia gama de temperaturas. Ofrecen estabilidad mecánica, térmica y eléctrica, junto con un alto grado de sensibilidad.

La excelente combinación de precio y el rendimiento ha dado lugar a una amplia utilización de los termistores NTCs en aplicaciones tales como medición y control de temperatura, compensación de temperatura y medición del flujo de fluidos.

Termistor PTC: Un termistor PTC es un resistor que depende de la temperatura, son fabricación de titanato de bario y deben elegirse cuando se requiere un cambio drástico en la resistencia a una temperatura específica o nivel de corriente. Los termistores PTCs pueden operar en los siguientes modos:

Sensores de temperatura que trabajan en rangos de temperatura entre 60°C a 180°C, por ejemplo, para protección de los bobinados de motores eléctricos y transformadores.

Como fusible de estado sólido de protección contra el exceso de corriente, que van desde mA a varios A (25° C ambiente) a niveles de tensión continua superior a 600V. [10]

1.12 Termoresistencias-RTD (Pt100).

La termoresistencia trabaja según el principio de que en la medida que varía la temperatura, su resistencia se modifica, y la magnitud de esta modificación puede relacionarse con la variación de temperatura. Las termoresistencias de uso más común se fabrican de alambres finos soportados por un material aislante y luego encapsulados. El elemento encapsulado se inserta luego dentro de una vaina o tubo metálico cerrado en un extremo que se llena con un polvo aislante y se sella con cemento para impedir que absorba humedad.

El platino encuentra aplicación dentro de un amplio rango de temperaturas y es el material más estable y exacto. En efecto, la relación

resistencia-temperatura correspondiente al alambre de platino es tan reproducible que la termoresistencia de platino se utiliza como estándar internacional de temperatura desde -260°C hasta 630°C . En la **Figura 8** se observa la señal del dispositivo termo resistivo.

En general el sensor viene introducido dentro de un tubo protector metálico de acero inoxidable o construido de aceros especiales o aleaciones, como el Inconel, Incoloy y Hastelloy. Su construcción puede hacerse con 2, 3 o 4 cables, según la necesidad del proceso. La interconexión entre termoresistencias e instrumentos se realiza con cable común de cobre, que se verá más adelante. [11]

Para nuestro trabajo usaremos un Pt100 que es un sensor de temperatura, el cual consiste en un alambre de platino que a 0°C tiene 100 ohms y que al aumentar la temperatura aumenta su resistencia eléctrica. El incremento de la resistencia es creciente y se aproxima a la linealidad, siendo esto característico del platino de tal forma que mediante tablas es posible encontrar la temperatura exacta a la que corresponde.

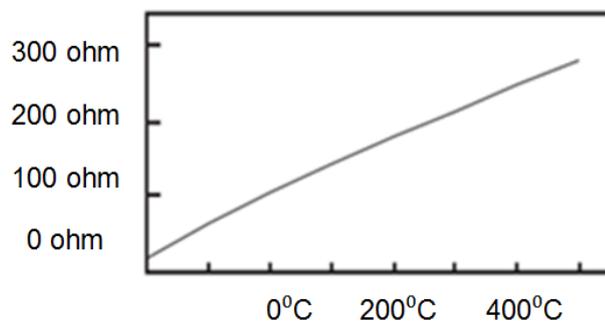


Figura 8 Dispositivo termo resistivo

Fuente: [11]

Normalmente las Pt100 industriales se consiguen encapsuladas en la misma forma que las termocuplas, es decir dentro de un tubo de acero inoxidable u otro material (vaina), en un extremo está el elemento sensible (alambre de platino) y en el otro está el terminal eléctrico de los cables protegido dentro de una caja redonda de aluminio (cabezal).

Por otra parte las ventajas que ofrecen los Pt100 siendo levemente más costosos y mecánicamente no tan rígidos como las termocuplas, las superan especialmente en aplicaciones de bajas temperaturas. (-100 a 200 °C).

Los Pt100 pueden fácilmente entregar precisiones de una décima de grado con la ventaja que la Pt100 no se descompone gradualmente entregando lecturas erróneas, si no que normalmente se abre, con lo cual el dispositivo medidor detecta inmediatamente la falla del sensor y da aviso. Este comportamiento es una gran ventaja en usos como cámaras frigoríficas donde una desviación no detectada de la temperatura podría producir algún daño grave. Además la Pt100 puede ser colocada a cierta distancia del medidor sin mayor problema (hasta unos 30 metros) utilizando cable de cobre convencional para hacer la extensión. [11]

Existen 3 modos de conexión para las Pt100 que a continuación se explicará, cada uno de ellos requiere un instrumento lector distinto. El objetivo es determinar exactamente la resistencia eléctrica $R(t)$ del elemento sensor de platino sin que influya en la lectura la resistencia de los cables R_c .

1.12.1 RTD 2 hilos.

El modo más sencillo de conexión (pero menos recomendado) es con solo dos cables, como se observa en la Figura 9. En este caso las resistencias de los cables R_{c1} y R_{c2} que unen la Pt100 al instrumento se suman generando un error inevitable. El lector medirá el total $R(t) + R_{c1} + R_{c2}$ en vez de $R(t)$. Lo único que se puede hacer es usar cable lo más grueso posible para disminuir la resistencia de R_{c1} y R_{c2} y así disminuir el error en la lectura.



Figura 9 Conexión RTD 2 hilos

Fuente: [11]

1.12.2 RTD 3 hilos.

El modo de conexión de 3 hilos es el más común y resuelve bastante bien el problema de error generado por los cables como se muestra en la Figura 10.

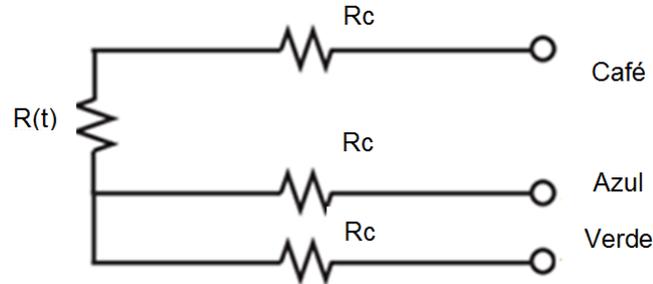


Figura 10 Conexión RTD 3 hilos

Fuente: [11]

El único requisito es que los tres cables tengan la misma resistencia eléctrica pues el sistema de medición se basa (casi siempre) en el "puente de Wheatstone". Por supuesto el lector de temperatura debe ser para este tipo de conexión.

1.12.3 RTD 4 hilos.

El método de 4 hilos es el más preciso de todos, los 4 cables pueden ser distintos (distinta resistencia) pero el instrumento lector es más costoso (véase la Figura 11).

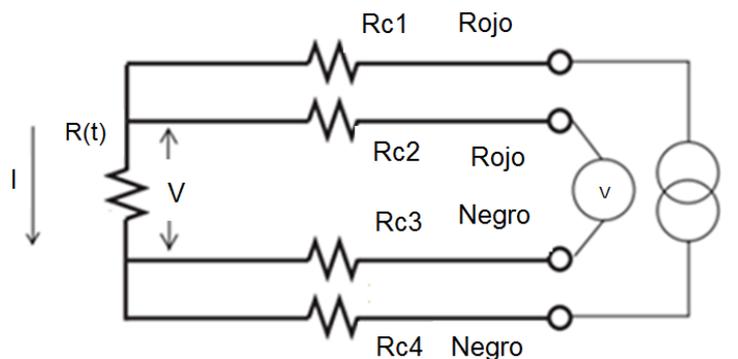


Figura 11 Conexión RTD 4 hilos

Fuente: [11]

Finalmente se deben tener ciertas precauciones de limpieza y protección en la instalación de los Pt100 para prevenir errores por fugas de corriente.

Es frecuente que cables en ambientes muy húmedos se deterioren y se produzca un paso de corriente entre ellos a través de humedad condensada. Aunque mínima, esta corriente "fugada" hará aparecer en el lector una temperatura menor que la real. Estas fugas también pueden ocurrir en óxido, humedad o polvo que cubre los terminales. [11]

Por la descripción hecha de los métodos de medición, queda claro que a diferencia de las termocuplas, no es posible conectar 2 unidades lectoras a un mismo Pt100 pues cada una suministra su corriente de excitación. En el momento de comprar un Pt100 se debe tener presente que existen distintas calidades y precios para el elemento sensor que va en el extremo del Pt100, los de mejor calidad están hechos con un verdadero alambre de platino, en tanto que existen algunos sensores económicos hechos en base a una pintura conductora sobre un substrato de alúmina (cerámica), estos últimos son menos precisos.

En general no se debe montar un Pt100 en lugares sometidos a mucha vibración pues es probable que se fracture.

1.12.4 Construcción de termoresistencias.

El aspecto exterior de las termoresistencias industriales es prácticamente idéntico al de las termocuplas. Se aplican las mismas consideraciones ambientales y de instalación y se debe prestar la misma atención a los conceptos de presión, temperatura, ataque químico, abrasión, vibración, porosidad y velocidad de fluido, requiriéndose los mismos tipos de vainas de protección.

1.13 Termocupla.

Las termocuplas son los sensores de temperatura eléctricos más utilizados en la industria. Una termocupla se hace con dos alambres de distinto material unidos en un extremo, como se muestra la **Figura 12**, que al aplicar temperatura en la unión de los metales se genera un voltaje muy pequeño, del orden de los mV (milivoltios) el cual aumenta con la temperatura. [12]

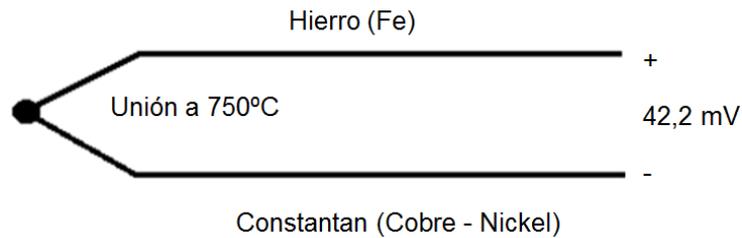


Figura 12 Conexión de termocupla

Fuente: [12]

Estos dispositivos suelen ir encapsulados en vainas, para protegerlos de las condiciones extremas en ocasiones del proceso industrial que tratan de ayudar a controlar, por ejemplo suele utilizarse acero inoxidable para la vaina, de manera que en un extremo está la unión y en el otro el terminal eléctrico de los cables, protegido adentro de una caja redonda de aluminio (cabezal). Además según la distancia a los aparatos encargados de tratar la pequeña señal eléctrica de estos transductores, también deben utilizarse cables compensados para transportar esta señal sin que la modifique o la modifique de una manera fácilmente reconocible y reversible para los dispositivos de tratamiento de la señal. También se da el caso de que los materiales empleados en la termocupla como el platino puro, hagan inviable económicamente extender la longitud de los terminales de medición de la termocupla, observe la Figura 13.

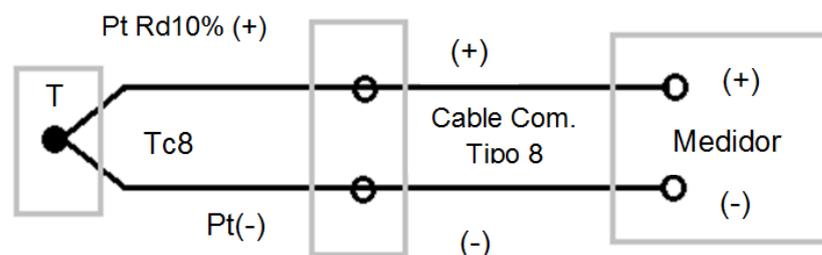


Figura 13 Conexión cable compensado y termocupla

Fuente: [12]

Los cables compensados tienen una polaridad de conexión (+) y (-) que al conectarse con la termocupla se debe respetar. Es importantísimo que estos dos cables compensados sean para el tipo de termocupla que se está usando y además estén conectados con la polaridad correcta (+) con (+) y (-) con (-). De otra forma será imposible obtener una medición sin error.

Las termocuplas podrían clasificarse atendiendo a varios criterios como material del que están construidas, su tolerancia o desviación, etc. Durante varios años ha habido diferentes organismos de estandarización de nacionalidades diferentes intentando normalizar la gran variedad de este tipo de sensores e incluso unificar sus criterios de normalización. [12]

1.13.1 Termocupla tipo J.

Una termocupla tipo J está hecha con un alambre de hierro y otro de constantán. El alambre de hierro se puede reconocer con un imán, además el hierro es gris opaco aunque algunas veces estos alambres se recubren con una delgada capa de cobre para evitar oxidación. El constantán (cobre-níquel) es también magnético pero muy levemente, se reconoce mejor porque es plateado brillante.

1.13.2 Termocupla tipo K.

Las termocuplas K están hechas con chromel (cromo - aluminio) y alumel (aluminio - níquel) ambos de color plateado brillante pero el alumel es levemente magnético por su contenido de níquel.

1.13.3 Ventajas e inconvenientes de las termocuplas.

- **Tipo J:** Para uso continuo en atmósferas oxidantes, reductoras e inertes y en vacío hasta 760° C. Por encima de 540° C, el alambre de hierro se oxida rápidamente, requiriéndose entonces alambre de mayor diámetro para extender su vida en servicio. La ventaja fundamental de la termocupla Tipo J es su bajo costo.
No se deben usar en atmósferas sulfurosas por encima de 540° C. A causa de la oxidación y fragilidad potencial, no se las recomienda para temperaturas inferiores a 0° C. No deben someterse a ciclos por encima de 760° C, aún durante cortos períodos de tiempo, si en algún momento posterior llegaran a necesitarse lecturas exactas por debajo de esa temperatura.
- **Tipo K:** Para uso continuo en vacío y en atmósferas oxidantes, reductoras e inertes.

Su desventaja reside en el hecho de que su límite máximo de temperatura es de tan solo 370° C para un diámetro de 3,25 mm. Resultan adecuadas para mediciones debajo de 0° C, pero se recomienda para ese propósito a las termocuplas Tipo E.

Otra desventaja adicional: la compensación de cero, se debe a que en algún punto, habrá que empalmar los cables de la termocupla con un conductor normal de cobre.

Alinear en ese punto se producirán dos nuevas termocuplas con el cobre como metal para ambas, generando cada una un voltaje proporcional a la temperatura de ambiente en el punto del empalme, observe la Figura 14.

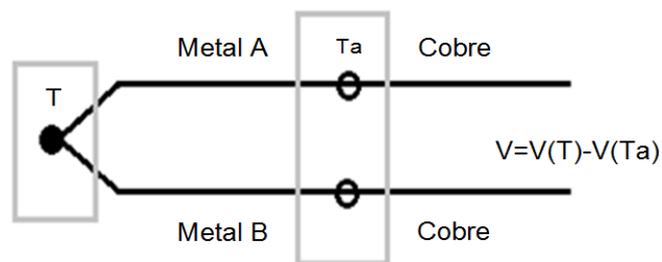


Figura 14 Conexión temperatura ambiente

Fuente: [12]

Antiguamente se solucionaba este problema colocando los empalmes en un baño de hielo a cero grados para que generen cero voltaje ($Ta = 0$ y luego $V(Ta) = 0$).

Actualmente todos los instrumentos modernos miden la temperatura en ese punto (mediante un sensor de temperatura adicional) y la suman para crear la compensación y obtener así la temperatura real. [12]

El punto de empalme (llamado "unión o juntura de referencia") es siempre en el conector a la entrada del instrumento pues ahí está el sensor de temperatura. De modo que es necesario llegar con el cable de la termocupla hasta el mismo instrumento.

1.14 Software para el desarrollo.

1.14.1 Eclipse.

Eclipse es una plataforma de desarrollo de código abierto basada en Java. Por si misma, es simplemente un marco de trabajo y un conjunto de servicios para la construcción del entorno de desarrollo de los componentes de entrada. Afortunadamente, Eclipse tiene un conjunto de complementos, incluidas las Herramientas de Desarrollo de Java (JDT). [13]

Un punto muy importante a notar son los diversos lenguajes de programación utilizados en el desarrollo del proyecto como se muestra en la Tabla 2.

De acuerdo al análisis realizado usando SLOCCount, el lenguaje más utilizado es Java, seguido de ANSI C. [13]

Tabla 2
Lenguajes de programación eclipse

LENGUAJE	LÍNEAS DE CÓDIGO	%
JAVA	1.911.693	92,66%
ANSI C	133.263	6,46%
C++	10.082	0,49%
JSP	3.613	0,18%
SH	2.066	0,10%
PERL	1.468	0,07%
PHP	896	0,04%

Fuente: [13]

1.14.2 Matlab.

MATLAB (Matrix Laboratory) es un programa interactivo de uso general. Es un instrumento computacional simple, versátil y de gran poder para aplicaciones numéricas, simbólicas y gráficas y contiene una gran cantidad de funciones predefinidas para aplicaciones en ciencias e ingeniería.

La interacción se realiza mediante instrucciones (denominadas comandos), y también mediante funciones y programas en un lenguaje estructurado. Los objetos básicos con los cuales opera MATLAB son

matrices. La asignación de memoria a cada variable la realiza MATLAB en forma dinámica y eficiente, por lo que no son necesarias las declaraciones de variables antes de su uso. [14]

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI). Además, se pueden ampliar las capacidades de MATLAB con los Toolkits; y las de Simulink con los paquetes de bloques (blocksets). [14]

En la Tabla 3 se indica los Toolkits y paquetes de bloques con las funcionalidades de Matlab se agrupan en más de 35 cajas de herramientas y paquetes de bloques (para Simulink).

Tabla 3

Toolkits de matlab

MATLAB (Toolkits)	Simulink
Matemáticas y Optimización.	Modelado de punto fijo.
Estadística y Análisis de datos.	Modelado basado en eventos.
Diseño de sistemas de control y análisis.	Modelado físico.
Procesado de señales y comunicaciones.	Gráficos de simulaciones.
Procesado de imagen.	Diseño de sistemas de control y análisis.
Pruebas y medidas.	Procesado de señales y comunicaciones.
Biología computacional.	Generación de código.
Modelado y análisis financiero.	Prototipos de control rápido y SW/HW HIL.
Desarrollo de aplicaciones.	Tarjetas integradas.
Informes y conexiones a bases de datos.	Verificación, validación y comprobación.

Fuente: [14]

Algunas de las características que nos ofrece Matlab se indican a continuación:

- Cálculo numérico rápido y con alta precisión.
- Manejo simbólico.
- Graficación y visualización avanzada.
- Programación mediante un lenguaje de alto nivel.
- Programación estructurada y orientada a objetos.
- Soporte básico para diseño de interfaz gráfica.
- Extensa biblioteca de funciones.
- Aplicaciones especializadas para algunas ramas de ciencias e ingeniería (toolboxes).

La operación en Matlab se describe en cuatro aspectos muy importantes:

- Simple y eficiente.
- Interactivo.
- Sistema de ayuda en línea.
- Interacción con otros entornos.

1.15 Controladores.

A continuación se puede dar a conocer los siguientes controladores más comunes:

- PAC.
- PLC.
- Microcontrolador STM32F407.

1.15.1 PAC.

Un controlador de automatización programable, o PAC, es una tecnología industrial orientada al control automatizado, al diseño de prototipos y a la medición. El PAC se refiere al conjunto formado por un controlador (una CPU típicamente), módulos de entradas y salidas, y uno o múltiples buses de datos que lo interconectan todo.

Este controlador combina eficientemente la fiabilidad de control de un autómatas (controlador lógico programable o PLC) junto a la flexibilidad de monitorización y cálculo de un PC. A veces incluso se le une la velocidad y personalización de la microelectrónica.

Los PACs pueden utilizarse en el ámbito investigador, pero es sobre todo en el industrial, para control de máquinas y procesos, donde más se utiliza. A destacar los siguientes: múltiples lazos cerrados de control independientes, adquisición de datos de precisión, análisis matemático y memoria profunda, monitorización remota, visión artificial, control de movimiento y robótica, seguridad controlada, etc.

Los PAC se comunican usando los protocolos de red abiertos como TCP/IP, OPC (OLE for procesos control), SMTP, puerto serie (con Modbus), y es compatible con los privados (CAN, Profibus, etc.).

1.15.2 PLC.

Un controlador lógico programable, es una computadora utilizada en la ingeniería automática o automatización industrial, para automatizar procesos electromecánicos, tales como el control de la maquinaria de la fábrica en líneas de montaje.

Los PLC son utilizados en muchas industrias y máquinas. A diferencia de las computadoras de propósito general, el PLC está diseñado para múltiples señales de entrada y de salida, rangos de temperatura ampliados, inmunidad al ruido eléctrico y resistencia a la vibración y al impacto. Los programas para el control de funcionamiento de la máquina se suelen almacenar en baterías copia de seguridad o en memorias no volátiles.

Un PLC es un ejemplo de un sistema de tiempo real, donde los resultados de salida deben ser producidos en respuesta a las condiciones de entrada dentro de un tiempo limitado, de lo contrario no producirá el resultado deseado.

Dentro de las ventajas que estos equipos poseen se encuentra que, gracias a ellos, es posible ahorrar tiempo en la elaboración de proyectos,

pudiendo realizar modificaciones sin costos adicionales. Por otra parte, son de tamaño reducido y mantenimiento de bajo costo, además permiten ahorrar dinero en mano de obra y la posibilidad de controlar más de una máquina con el mismo equipo. Sin embargo, y como sucede en todos los casos, los controladores lógicos programables, o PLC's, presentan ciertas desventajas como es la necesidad de contar con técnicos calificados y adiestrados específicamente para ocuparse de su buen funcionamiento.

1.15.3 Microcontrolador STM32F407.

Las STM32F407/417 están diseñadas para aplicaciones médicas, industriales y de consumo donde se requiere un alto nivel de integración y rendimiento, además de un rico conjunto de periféricos dentro de los paquetes tan pequeños como de 10 x 10 mm. A continuación se muestra una fotografía de la tarjeta de desarrollo en la Figura 15. [15]



Figura 15 Tarjeta SMT32F407

Fuente: [15]

Requisitos de Hardware: La única pieza de hardware necesaria para empezar a utilizar la entrenadora es un cable de conexión USB A a mini-B y un puerto USB disponible en su PC. Si desea utilizar el puerto USB OTG necesitará un cable USB de tipo A a micro-B.

Requisitos de Software: Será necesaria una plataforma de Windows para ejecutar las herramientas de desarrollo para microcontroladores STM32. Otras herramientas como Atollic, IAR y Keil están disponibles, estas herramientas también cuentan con versiones de evaluación. Alternativamente, existen aplicaciones de código abierto como ARM GNU toolchain, que puede ser usado para programar los microcontroladores STM32.

Microcontrolador STM32F407VG: Tiene un encapsulado LQFP de 100 pines entre entradas-salidas analógicas y digitales. Este microcontrolador contiene un núcleo ARM Cortex-M4 con FPU (unidad de punto flotante).

Algunas de las características del microcontrolador STM32F407VG son:

- 1 Mbyte de memoria flash.
- 192KBytes de RAM.
- Oscilador RC.
- Alimentación por una sola fuente de 1.8V a 3.6V.
- Puede operar a una velocidad de 168MHz.
- Pines I/O multiplexadas con muchos periféricos internos.
- USB OTG HS/FS.
- Ethernet.
- Controlador de memoria estática de soporte Compact Flash, SRAM, PSRAM, NOR y NAND.
- Interface paralela para LCD.

1.15.4 Características de la tarjeta.

ST-LINK es una herramienta ST Microelectrónica USB JTAG para programación y depuración STM8 STM32. Se trata de una herramienta

externa que se utiliza para microcontroladores STM para programación y depuración JTAG o utilizando SWD (Debug Cable Serial).

Un ST-LINK está incluido en la placa de evaluación STM32F4-Discovery que se puede utilizar para programar el microcontrolador STM32F407VG a bordo.

El incrustado ST-LINK también se puede utilizar para un solo paso a través del código fuente y depurar el microcontrolador de destino. Quitando dos jumpers en la placa, el incrustado ST-LINK puede utilizarse como programador / depurador para programar / depurar un microcontrolador ST en otra tabla a través de un cable desde el conector de SWD.

El embebido ST-LINK se limita a la programación de microcontroladores STM32F a través de un enlace SWD solamente. JTAG no es compatible con el incorporado ST-LINK.

1.15.5 Periféricos y soportes.

- **Dispositivos I/O Interface:** La placa cuenta con cuatro LEDs y un interruptor pulsador programables por el usuario. Un segundo botón pulsador está conectado a la patilla de reposición (NRST) del microcontrolador.
- **Audio:** El STM32F407VG es también conectado a un DAC de audio y micrófono MEMS. Una clavija estéreo de audio está a bordo para conectar auriculares o altavoces a la salida de audio de la DAC.
- **USB:** La tarjeta dispone de un puerto USB OTG toda velocidad disponible en un USB conector micro-B.
- **Sensor de Movimiento:** Un acelerómetro de tres ejes lineales.
- **Encabezados:** Todos los pines I/O de la STM32F407VG están disponibles en conectores macho, 2 por 25 (50 pin) con su cabecera ubicada a cada lado de la placa a lo largo de su longitud. Los nombres de los pines son serigrafiado al lado de los conectores macho. Los cabezales de pin sobresalen de la parte superior del tablero para permitir una fácil conexión de sondas de osciloscopio o analizador lógico para pruebas.

- **Osciladores:** Aunque la STM32F407VG tiene dos osciladores internos RC, utiliza un cristal de 8MHz para el oscilador principal.
- **Puentes de soldadura:** La placa tiene un número de conectores de puente de soldadura que pueden ser soldadas o cerradas desoldada para abrirlas con el fin de hacer o deshacer las conexiones.
- **Consumo de corriente:** Un enlace de puente de la placa se puede quitar lo que permite conectar un amperímetro, si el consumo de corriente del microcontrolador necesita ser medido.

Algunos de los periféricos interconectados que posee la tarjeta ST-LINK son los siguientes:

- 4 Usuarios LEDs programables.
- 1 Pulsador.
- 1 Reset Pulsador.
- 3 Ejes acelerómetro.
- Micrófono
- DAC de audio con amplificador de clase D.
- USB OTG puerto FS.
- Cabeceras de todos los pines de E/S.

Esta tarjeta presenta las siguientes ventajas:

- Incluye un selector incrustado ST-LINK para programar y depurar.
- Alimentado por USB para una evaluación rápida y fácil.
- Excelente documentación.

Desventaja de la tarjeta ST-LINK:

- Sin conector JTAG para probar externo ST-LINK programador / depurador.

1.16 TFT touch screen.

Una pantalla táctil (en inglés Touch Screen) es una pantalla que mediante un toque directo sobre su superficie permite la entrada de datos y órdenes al dispositivo, y a su vez muestra los resultados introducidos

previamente; actuando como periférico de entrada y salida de datos, así como emulador de datos interinos erróneos al no tocarse efectivamente. Este contacto también se puede realizar por medio de un lápiz óptico u otras herramientas similares. Actualmente hay pantallas táctiles que pueden instalarse sobre una pantalla normal, de cualquier tipo (LCD, monitores y televisores CRT, plasma, etc.)

Las pantallas táctiles se hicieron populares por su uso en dispositivos de la industria, ordenadores públicos (como exposiciones de museos, pantallas de información, cajeros automáticos de bancos, etc.) donde los teclados y los ratones no permiten una interacción satisfactoria, intuitiva, rápida, o exacta del usuario. [16]

Desde finales del siglo XX y especialmente en los comienzos del XXI alcanzan un uso habitual en la mayoría de los dispositivos con pantalla: monitores de computadora, teléfonos móviles, tabletas, etc.

1.16.1 Desarrollo y utilización.

La gran mayoría de las tecnologías de pantalla táctil significativas fueron patentadas durante las décadas de 1970 y 1980 y actualmente han expirado. Este hecho ha permitido que desde entonces los diseños de productos y componentes que utilizan dichas tecnologías no estén sujetos a regalías, lo que ha permitido que los dispositivos táctiles se hayan extendido más fácilmente.

Con la creciente aceptación de multitud de productos con una pantalla táctil integrada, el coste marginal de esta tecnología ha sido rutinariamente absorbido en los productos que las incorporan haciendo que prácticamente desaparezca. Como ocurre habitualmente con cualquier tecnología, el hardware y el software asociado a las pantallas táctiles ha alcanzado un punto de madurez suficiente después de más de tres décadas de desarrollo, lo que le ha permitido que actualmente tengan grado muy alto de fiabilidad. Como tal, las pantallas táctiles pueden hallarse en la actualidad en aviones, automóviles, consolas, sistemas de control de maquinaria y dispositivos de mano de cualquier tipo.

1.17 Dispositivos móviles.

Un dispositivo móvil se puede definir como un aparato de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales. [17]

1.17.1 Sistemas operativos móviles.

- Android.
- Bada.
- BlackBerry OS.
- Firefox OS.
- iOS.
- MeeGo.
- Palm OS.
- Symbian OS.
- Windows CE.
- Windows Mobile.
- Windows Phone.

Android: Es una plataforma formada por un conjunto de software en estructura de pila (software stack) que incluye un sistema operativo, software para conectar aplicaciones (middleware) y aplicaciones base. El SDK (Kit de Desarrollo de Software) de Android proporciona varias herramientas y API (Interfaz de Programación de Aplicaciones) que son necesarias para desarrollar aplicaciones Android. Estas aplicaciones se desarrollan en lenguaje Java. De tal manera que se lo utilizan en su mayoría en tablets, la Tabla 4 muestra la arquitectura del sistema operativo Android que está organizado en cuatro grandes bloques (ver Tabla 4).

Tabla 4

Estructura de la pila de software del S.O android

APPLICATIONS				
Home	Contacts	Phone	Browser	...
APPLICATION FRAMEWORK				
Activity Manager		Windows Manager	Content Providers	View System
Package Manager	Telephony Manager	Resource Manager	Location Manager	Notification Manager
LIBRARIES				
Surface Manager	Media Framework	SQLite	ANDROID RUNTIME Core Libraries. Dalvik Virtual Machine.	
OpenGL/ES	FreeType	WebKit		
SGL	SSL	libc		
LINUX KERNEL				
Display Driver	Camera Driver	Flash Memory Driver	Binder (IPC) Driver	
Keypad Driver	Wifi Driver	Audio Drivers	Power Management	

Fuente: [17]

Bada: Samsung Bada es una plataforma o sistema operativo para smartphones lanzado en 2010. Lo que Samsung intenta conseguir con esta plataforma es lo mismo que dice su lema: "Smartphone para todos", su objetivo no es competir con las principales plataformas actuales, cómo Android o IOS. Lo que intenta es adentrar a los usuarios de teléfonos convencionales en el mundo de los smartphones, proporcionando teléfonos inteligentes y a su mismo tiempo rentables.

A pesar de que Bada es soportado por una amplia gama de dispositivos, es lo suficientemente potente como para ofrecer muchas posibilidades, las características principales son:

- Gran control de la interfaz.
- Soporte para flash.
- Samsung App, tienda de aplicaciones propia que cuenta con algo más de 2000 aplicaciones y juegos
- Integración con Facebook y twitter.
- Social hub, integración con mensajería instantánea y emails.

BlackBerry OS: BlackBerry está desarrollado por la compañía RIM (Research In Motion). Los móviles BlackBerry destacan principalmente por su capacidad de enviar y recibir correo electrónico por Internet a través de los operadores que ofrecen este servicio.

Los dispositivos BlackBerry montan el sistema operativo BlackBerry OS, desarrollado por RIM. Este sistema operativo es propietario, con lo cual no hay información pública relevante sobre su diseño ni arquitectura. A día de hoy la última versión del sistema operativo es BlackBerry OS 7. Las versiones del sistema operativo tienen un Kernel que se basa en Java, montando la mayoría de los dispositivos arquitecturas ARM. ARM no construye sus propios chips CPU pero da licencia a terceros para que los fabriquen. El sistema operativo divide la memoria del dispositivo en tres secciones:

1. Memoria de aplicación: espacio de memoria destinado para albergar las aplicaciones.
2. Memoria de dispositivo: espacio para almacenar ficheros y otras fuentes.
3. Memoria para tarjeta: espacio de almacenamiento adicional.

Firefox OS: Es un sistema operativo móvil, basado en HTML5 con núcleo Linux, de código abierto (a diferencia de Android, específicamente la tienda de aplicaciones) para varias plataformas.

Es desarrollado por Mozilla Corporation bajo el apoyo de otras empresas y una gran comunidad de voluntarios de todo el mundo. El sistema operativo está diseñado para permitir a las aplicaciones HTML5 comunicarse directamente con el hardware del dispositivo usando JavaScript y Open Web APIs.

Inicialmente estuvo enfocado en los dispositivos móviles, smartphones y tabletas, específicamente en el sector de gama baja; en el 2013, Telefónica comenzó la venta del primer terminal con Firefox OS, el ZTE Open que fue rápidamente seguido por el teléfono Peak de Geeksphone. También se pudo

aplicar a otros dispositivos como Raspberry Pi, y en el desarrollo de computadores de bajo consumo y televisores (Smart TV y conectores HDMI).

La arquitectura de Firefox OS tiene tres componentes muy importantes:

- Gonk: Conformado por el kernel Linux y una capa de abstracción de hardware.
- Gecko: El entorno de ejecución.
- Gaia: La interfaz gráfica de usuario.

iOS: Desarrollado originalmente para el iPhone y con él, una nueva definición del teléfono móvil. Más tarde fue introducido en el iPod Touch y actualmente en el iPad. Las actualizaciones de este S.O se enumeraron desde la 1.x hasta la 1.1.5.

Un año después, en 2008, se lanzó el iPhone OS 2.0, cuyas actualizaciones llegaron hasta la 2.2. Comenzó entonces la revolución de las aplicaciones móviles y uno de los modelos de negocio más productivos existente hoy en día.

En el año 2009 se lanzó el iPhone OS 3.0 que evolucionó hasta la versión 3.1.3, la cual incluía el Spotlight (para realizar búsquedas en el dispositivo), también ofrecía la posibilidad de incluir la API de Google Maps, las operaciones de copiar/cortar/pegar, interconexión por Bluetooth o P2P y librerías GPS. Esta versión fue soportada por los iPhones e iPads de primera generación.

En 2010, empezó a llamarse iOS y se realizó el lanzamiento de la versión 4.0 del S.O. La versión 4.1.2 fue la última en los iPhone 3G y los iPod Touch de segunda generación.

A finales de 2011 se lanzó la versión iOS 5, con una interfaz mejorada y funcionalidades como la presencia de un asistente personal Siri, facilidades para la sincronización sin cables, un centro de notificaciones mejorado, el servicio de Messages, la navegación web con pestañas, entre otras.

La versión iOS 5 solo se podrá instalar en los Apple TV (segunda generación), iPhone 3GS, iPhone 4, iPhone 4S, iPod Touch (tercera y cuarta generación) y en los dos modelos de iPad actuales. De esta manera debemos tener en cuenta la arquitectura de iOS que se muestra en la Tabla 5.

Tabla 5
Sistema iOS

Cocoa Touch
Media
Core Services
Core OS

Fuente: [17]

MeeGo: Es la unión de los sistemas operativos Maemo de Nokia y Moblin de Intel, con los cuales pretendían competir con el sistema Android de Google.

MeeGo se presentó como un sistema preparado para funcionar en netbooks, dispositivos portátiles, sistemas en vehículos, televisiones y teléfonos multimedia. Básicamente se trata de una distribución Linux con soporte para ARM e Intel/Atom que usa Qt para su interfaz.

El núcleo del sistema operativo Meego es una distribución de Linux, hecho sobre Maemo (sistema de Nokia basado en Debian) y Moblin (sistema de Intel basado en Fedora). Meego es una de las primeras distribuciones de Linux en usar el sistema de archivos Btrfs por omisión, y utiliza repositorios RPM.

Palm OS (WEBOS): Es el sistema operativo de los dispositivos móviles desarrollados por la empresa Palm Inc. Esta empresa tuvo un gran auge con sus agendas electrónicas o PDA. Comenzó su actividad en 1996, creando Palm OS, un sistema operativo fácil de utilizar con pantallas táctiles e interfaces de usuario gráficas.

Palm OS ha tenido una gran evolución de versiones desde la 1.0 hasta la 5.0, luego pasó a llamarse Palm OS Cobalt. Este último es un sistema operativo basado en Linux, que evolucionó para denominarse webOS, en 2009, el primer dispositivo que lo incluía fue Palm Pre.

Symbian OS: El sistema operativo Symbian viene como evolución del sistema operativo Epoc, este fue desarrollado por Psion. Symbian es el resultado de adaptar Psion a dispositivos móviles y tiene diferentes variantes según el dispositivo en el que se utilice.

De todos los fabricantes de móviles es Nokia el que más ampliamente ha utilizado este sistema operativo. Ha desarrollado varias versiones del sistema, que denomina Series, está la Serie60, Serie70, Serie80 y Serie90, siendo la Serie60 la que más amplio uso ha tenido.

Sony Ericsson por ejemplo también utiliza Symbian, con su plataforma UIQ, en sus dispositivos, por ejemplo, en el modelo P800 o Motorola en el A1000.

Como hay muchas combinaciones de SO y UI, habrá también muchos kits de desarrollo (SDK), por ejemplo para la Serie60 hay SDK versión 6.1, 7.0, 8.0 y 9.0. Si nos centramos en la Serie60, tenemos que para cada edición de un SDK hay ampliaciones con mejoras llamadas paquetes de características o FP (Feature Pack), así tenemos:

- SDK for 1st Edition.
- SDK for 1st Edition, FP1.
- SDK for 2st Edition.
- SDK for 2st Edition, FP1.
- SDK for 2st Edition, FP2.

Además, para cada kit de desarrollo hay una variante en función del IDE en el que se instale por lo que habrá variantes diferentes.

Normalmente el entorno de desarrollo viene preparado para estar programado en C o C++ y aporta el emulador de la Serie para la que está desarrollado.

Windows CE: Es un sistema operativo desarrollado por Microsoft para sistemas embebidos. Windows CE no debe confundirse con Windows Embedded Standard, que es un sistema basado en Windows NT; Windows CE está desarrollado independientemente.

La versión actual de Windows Embedded Compact funciona en procesadores Intel x86 y compatibles, además de los tipos MIPS y ARM. Se puede encontrar en teléfonos inteligentes, notebook, hasta en pocket pc y GPS.

Una característica distintiva de Windows CE en comparación con otros sistemas operativos de Microsoft es que gran parte de ella se ofrecen en forma de código fuente. En primer lugar, el código fuente fue ofrecido a varios proveedores, para poder adaptarlo a su hardware. Entonces productos como Platform Builder (un entorno integrado para la creación de imágenes de Windows CE de sistema operativo y la integración, o sistema operativo personalizar los diseños basados en CE) ofrece varios componentes en forma de código fuente al público en general. Sin embargo, una serie de componentes básicos que no necesitan la adaptación a los entornos de hardware específico (aparte de la familia de CPU) siguen siendo distribuidos en forma de sólo binario. El último sistema operativo creado a base de Windows CE fue Windows Phone 7 y sus actualizaciones (7.5 y 7.8)

Windows Mobile: Es un sistema operativo móvil compacto desarrollado por Microsoft, y diseñado para su uso en teléfonos inteligentes y otros dispositivos móviles.

Se basa en el núcleo del sistema operativo Windows CE y cuenta con un conjunto de aplicaciones básicas utilizando las API de Microsoft Windows.

Está diseñado para ser similar a las versiones de escritorio de Windows estéticamente.

Originalmente apareció bajo el nombre de Pocket PC, como una ramificación de desarrollo de Windows CE para equipos móviles con capacidades limitadas.

En la actualidad, la mayoría de los teléfonos con Windows Mobile vienen con un estilete digital, que se utiliza para introducir comandos pulsando en la pantalla.

Windows Phone: Entre las novedades se encuentra la denominada interfaz de usuario “Metro” basada en la utilización de mosaicos dinámicos que muestran información útil al usuario. Además se introduce el concepto de HUB, en donde se centralizan las acciones y las aplicaciones se agrupan por el tipo de actividad que representan.

La Shell y la plataforma de aplicaciones residen en memoria de usuario, mientras que el Kernel, los drivers y el sistema de archivos, networking, el sistema de rendering y gráficos y el sistema de actualizaciones, residen en el espacio de Kernel. El sistema es de 32 bits, con lo cual maneja 4 GB de memoria, dos de los cuales son para procesos y los otros dos para el Kernel, en la siguiente Tabla 6 se muestra el modelo del software.

Tabla 6
Modelo de software en windows phone

EJECUCIÓN DE LA APLICACIÓN		
Modelo Aplicación	Modelo IU	Integración Nube
Kernel		
Hardware		

Fuente: [17]

1.18 Protocolos de comunicación.

Se entiende como protocolo de comunicación al conjunto de reglas usadas por computadoras para comunicarse unas con otras mediante una red.

Un protocolo es una convención o estándar que permite la conexión, comunicación y la transferencia de datos entre dos puntos finales.

A continuación se proyectan algunos ejemplos de protocolos de comunicaciones con la intención de aclarar el concepto y la evolución de los mismos:

- Protocolos punto a punto.
- Comunicación entre redes.
- Protocolos de transmisión de paquetes.
- El protocolo TCP/IP, Protocolo de control de transmisión (TCP)/ Protocolo de Internet (IP).

Los protocolos gestionan dos niveles de comunicación distintos. Las reglas de alto nivel definen como se comunican las aplicaciones, mientras que las de bajo nivel definen como se transmiten las señales y los más utilizados son: Ethernet, Token ring, Token bus, FDDI, CDDI, HDLC, Frame Relay y ATM. [18]

El protocolo de red determina el modo y organización de la información (tanto los datos como los controles) para su transmisión por el medio físico con el protocolo de bajo nivel. Los protocolos de red más comunes son: IPX/SPX, DECnet, X.25, TCP/IP, AppleTalk y NetBEUI.

1.18.1 Ethernet.

Ethernet es una popular tecnología LAN (Red de Área Local) que utiliza el Acceso múltiple con portadora y detección de colisiones entre estaciones con diversos tipos de cables, observe la Figura 16. La misma que hace referencia a su clasificación en la Tabla 7.

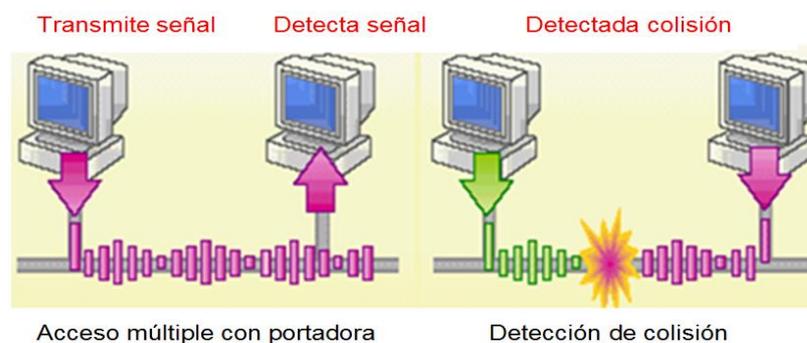


Figura 16 Ethernet AMP - detección de colisión

Fuente: [18]

Tabla 7

Clasificación de los tipos de ethernet

CARACTERÍSTICAS	DESCRIPCIÓN
Método de acceso	CSMA/CD
Velocidad de transferencia	Ethernet standards – 10Mbps Fast Ethernet – 100Mbps Gigabit Ethernet – 1Gbps (1000Mbps)

Fuente: [18]

Las características más importantes de Ethernet son:

- Es pasivo, es decir, no requiere una fuente de alimentación propia, y por tanto, no falla a menos que el cable se corte físicamente o su terminación sea incorrecta.
- Se conecta utilizando una topología de bus en la que el cable está terminado en ambos extremos.
- Utiliza múltiples protocolos de comunicación y puede conectar entornos informáticos heterogéneos, incluyendo NetWare, Unix, Windows Y Macintosh.

La importancia de Ethernet es popular porque permite un buen equilibrio entre velocidad, costo y facilidad de instalación. Estos puntos fuertes, combinados con la amplia aceptación en el mercado y la habilidad de soportar virtualmente todos los protocolos de red populares, hacen a Ethernet la tecnología ideal para la red de la mayoría de usuarios de la informática actual. [19]

El Hardware comúnmente utilizado por Ethernet se muestra a continuación:

- NIC o adaptador de red Ethernet.
- Repetidor.
- Concentrador o hub.
- Puente.
- Router.
- Switch.

1.18.2 Wifi.

En la actualidad es la que ofrece la mayor cantidad de beneficios al costo más bajo entre todas las tecnologías inalámbricas. Es económica, interoperable con equipos de diferentes fabricantes y puede ser extendida para ofrecer funcionalidades mucho más allá de las previstas originalmente por los fabricantes.

Esto se debe a que Wifi utiliza estándares abiertos: enrutadores, Tablet PCs, laptops y teléfonos, Wifi pueden interoperar porque todos adhieren al estándar 802.11

Los Protocolos de Redes Inalámbricas está formado por la familia de protocolos 802.11 es la base de Wifi.

- **802.11a** permite hasta 54 Mbps en las bandas no licenciada a 5GHz.
- **802.11b** permite hasta 11 Mbps en la banda no licenciada a 2.4GHz.
- **802.11g** permite hasta 54 Mbps en la banda no licenciada a 2.4GHz.
- **802.11n** permite hasta 600 Mbps en las bandas no licenciadas a 2.4GHz y 5GHz.

Los dispositivos Wifi pueden operar en alguno de los siguientes modos de funcionamiento:

- Master (AP -Access point)
- Managed (También llamado cliente o estación)
- Ad-hoc (Usado en redes en malla)
- Monitor (No usado normalmente para comunicaciones)

Las Redes Wifi pueden tener muchas utilidades prácticas para todo tipo de entidades, empresas o negocios, como se observa la Figura 17.

- Acceder a una red empresarial desde cualquier punto.
- Acceder a Internet sin necesidad de cables.
- Conectarse sin cables con un PC, PC Portátil, PDA, Teléfono móvil o videoconsola con conexión WIFI.
- Servicio de HotSpot para acceso restringido por tiempo o volumen.

- Acceder a servicios de VoIP sin cables. [20]

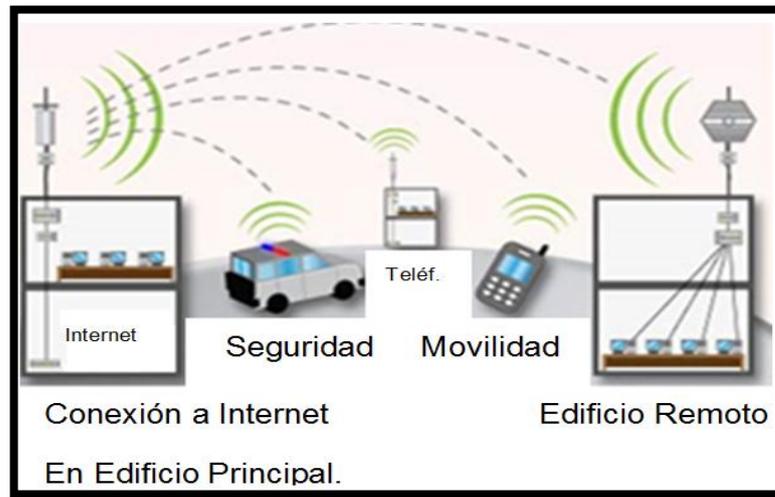


Figura 17 Diagrama de red wifi

Fuente: [20]

CAPÍTULO II

2 DISEÑO E IMPLEMENTACIÓN

2.1 Introducción.

En el presente capítulo detallaremos todo lo relacionado al diseño e implementación de un prototipo de módulo para visualización de curvas características de sensores de temperatura, como también la configuración y programación de los diferentes equipos a utilizar.

2.2 Especificaciones y requisitos del sistema.

El prototipo está diseñado para visualizar las curvas características de sensores de temperatura, con las mismas que se obtendrá las gráficas de las relaciones temperatura – resistencia y temperatura – voltaje, para ello la tarjeta STM32F4 se encarga de tomar datos de la variable de temperatura que son enviadas como señal eléctricas con la ayuda de acondicionadores MAX 31865 (RTD), MAX 31855K - MAX31855J (Termocuplas), Termistores y LM35 son acondicionados directamente con la tarjeta STM32F4, una vez procesada la información se envía todos los datos a través de tres comunicaciones distintas, una por medio de comunicación serial hacia la Touch Screen, otra por red Wifi para un dispositivo móvil y finalmente por red Ethernet a una PC, en donde se generan las curvas deseadas de cada uno de los sensores de temperatura. Para realizar las pruebas hemos diseñado un control de temperatura con rangos de 0 a 200 °C.

La tarjeta STM32F4 recibirá la señal eléctrica de la variable del proceso de temperatura, proveniente de un sensor, ésta tiene la capacidad de controlar la temperatura en el generador en base a un lazo de control, la STM32F4 envía la señal de control a la Niquelina y al Peltier para regular la temperatura en el generador y de este modo mantener la variable constante de acuerdo al valor deseado. Será posible monitorear las variables del proceso en una Touch Screen que tiene un HMI en el cual se puede controlar y visualizar las curvas características de la variable.

Los sensores utilizados en la obtención de datos para el prototipo de módulo de visualización de curvas características de temperatura son: PT100 2, 3, 4 hilos, Termistor 1K Ω , 5K Ω , 10K Ω , Termocupla tipo J y K, LM35.

Los sensores están acoplados a los acondicionadores para obtener señales estándar de voltaje, las cuales son muy importantes para el módulo ya que a partir de ellas se obtienen las curvas de cada sensor.

2.2.1 Sensores de temperatura.

- **Termistor 1K Ω**

Aplicado en medición de la temperatura, detección y control, compensación de temperatura en electrónica industrial y de consumo (ver Figura 18).

Rango de temperatura de funcionamiento: -40 °C a +125 °C.

Tolerancia en R₂₅: ± 2 , ± 3 , ± 5 %



Figura 18 Termistor 1K Ω

Fuente: [21]

- **Termistor 5K Ω**

Estos termistores están hechos de material cerámico NTC. Se usa en medición de la temperatura, detección y control en aplicaciones industriales, de consumo y telecomunicaciones. Para la detección de a bordo o la teledetección exacta (ver Figura 19).

Rango de temperatura de funcionamiento: -40 °C a +125 °C.

Tolerancia en R₂₅: ± 1 , ± 2 , ± 3 , ± 5 %



Figura 19 Termistor 5KΩ

Fuente: [21]

- **Termistor 10KΩ**

Estos termistores tienen un coeficiente de temperatura negativo. El dispositivo consiste en un chip con dos cables de cobre sólido estaño plateados como se observa en la Figura 20. Es lacado gris y un código de color, pero no aislado. Es aplicado en medición de la temperatura, detección y control, compensación de temperatura en electrónica industrial y de consumo.

Rango de temperatura de funcionamiento: -40 °C a +125 °C.

Tolerancia en R_{25} : ± 2 , ± 3 , ± 5 %



Figura 20 Termistor 10KΩ

Fuente: [21]

- **Termocuplas**

En el presente proyecto se considera solo las termocuplas de uso más común como son la tipo J y tipo K [22].

- **Tipo J**

Se seleccionó esta termocupla porque se usan principalmente en la industria del plástico, goma (extrusión e inyección) y fundición de metales a bajas temperaturas (Zamac, Aluminio).

La termocupla J (ver Figura 21), tiene un cable positivo de aleación (hierro), cable negativo de aleación (cobre/níquel) a un

rango de temperatura de -180 a 750 °C a un voltaje máximo de 4.2v.

Tolerancia: +/- 0.5°C.



Figura 21 Termocupla J

Fuente: [22]

- **Tipo K**

Esta termocupla que se indica en la Figura 22 se usa típicamente en fundición y hornos a temperaturas menores de 1300 °C, por ejemplo fundición de cobre y hornos de tratamientos térmicos de ahí la importancia para la selección de la misma.

Esta termocupla tiene un cable positivo de aleación (níquel/cromo), cable negativo de aleación (níquel/aluminio) a un rango de temperatura de -180 a 1300 °C a un voltaje máximo de 54.8v.

Tolerancia: +/- 0.5°C.



Figura 22 Termocupla K

Fuente: [22]

- **LM35**

Un sensor típico seleccionado en este proyecto para análisis es el LM35, que es un tipo de sensor en circuito integrado, con una

precisión calibrada de 1°C y un rango que abarca desde -55° a +150°C [23].

El sensor se presenta en diferentes encapsulados como el LM35CH de la Figura 24, pero el más común es el de la Figura 23, de igual forma que un típico transistor con 3 patas, dos de ellas para alimentarlo y la tercera nos entrega un valor de tensión proporcional a la temperatura medida por el dispositivo.

La salida es lineal y equivale a 10mV/°C por lo tanto:

$$+1500\text{mV} = 150^\circ\text{C}$$

$$+250\text{mV} = 25^\circ\text{C}$$

$$-550\text{mV} = -55^\circ\text{C}$$

El sensor de temperatura puede usarse para compensar un dispositivo de medida sensible a la temperatura ambiente, refrigerar partes delicadas del robot o bien para loggear temperaturas en el transcurso de un trayecto de exploración.

También se puede realizar

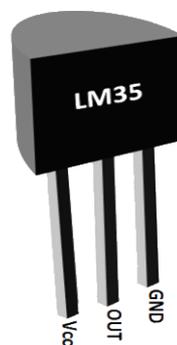


Figura 23 LM35

Fuente: [23]



Figura 24 LM35CH

Fuente: [23]

- **PT100**

Se ha seleccionado para análisis de respuesta a una Pt100 que consiste en un alambre de platino que a 0°C tiene 100 ohms y que al aumentar la temperatura aumenta su resistencia eléctrica. Un Pt100 es un tipo particular de RTD (Dispositivo Termo Resistivo) (ver Figura 25). Teniendo como ventaja que la Pt100 siendo levemente más costosas y mecánicamente no tan rígidas como las termocuplas, las superan especialmente en aplicaciones de bajas temperaturas (-100 a 200 °).



Figura 25 PT100

Fuente: [12]

2.3 Diseño del generador de temperatura.

En la Figura 26 se muestra la estructura del generador de temperatura que está diseñado para alcanzar rangos de 0 a 200 °C, teniendo en cuenta que para rangos menores a la temperatura ambiente utilizamos un módulo Peltier; y para rangos superiores a la temperatura ambiente necesitamos de una Niquelina de 2200 W, además fue necesario el uso de un disipador de alta potencia que nos ayuda a la disipación del calor que es generado a un lado del Peltier, también fue preciso la utilización de ventiladores para disipar el calor generado por la Niquelina y así evitar que ésta se queme.



Figura 26 Generador de temperatura

2.3.1 Resistencia eléctrica (niquelina).

La resistencia calefactora utilizada en nuestro generador la fabricamos con tubería Incoloy (súper-aleaciones), que están hechas con base de níquel (ver Figura 27), y tienen características que incluyen gran resistencia a la corrosión en ambientes acuosos, excelente fuerza y resistencia a la oxidación en altas temperaturas.

Las súper-aleaciones resistentes a la corrosión son ampliamente usadas en ambientes extremos donde la resistencia a la corrosión y a las altas temperaturas es supremamente importante para un buen resultado en el producto final. Estas súper-aleaciones son ampliamente usadas en industrias de procesos químicos y petroquímicos, plantas eléctricas, e industrias petroleras y gas. [24]

Para el diseño de la Niquelina nos basamos en los siguientes aspectos:

$$\text{Área del Disipador: } 11\text{cm} * 26.5\text{cm} = 291.5\text{cm}^2$$

$$\text{Material de la Resistencia: Tubería Incoloy } \phi \frac{1}{2} \rightarrow \text{AceroInoxidable}$$

$$\text{Para una Resistencia Tubular de } \frac{1}{2} \text{ pulgada} = 20 \frac{W}{cm}$$

$$\text{Diámetro del Tubo} = 12.69\text{mm}$$

$$\text{Aislante} \rightarrow \text{Óxido de Magnesio}$$

Procedimiento:

- Calcular la dimensión del Tubo Incoloy.

Necesitamos 2200W a 110V → 200°C

$$\text{Longitud Tubo Incoloy} = \frac{2200W}{20 \frac{W}{cm}} = 110cm$$

- Introducir en el Tubo Incoloy la niquelina tipo resorte.
- Colocar dos arandelas a los lados del Tubo para sujetar los extremos de la niquelina tipo resorte.
- Rellenar con el óxido de magnesio el Tubo Incoloy completamente.
- Colocar los conectores en cada uno de los extremos de la resistencia.
- Luego se coloca la resistencia en una Teficadora por tres veces, para compactar completamente el aislante, reduciendo el diámetro de la resistencia.

Teficadora:

1. – *Ingresar de 12.69mm → Sale a 12.1mm*

2. – *Ingresar de 12.1mm → Sale a 11.6mm*

3. – *Ingresar de 11.6mm → Sale a 11.1mm*

- Finalmente doblamos la resistencia en forma de M, de tal manera que alcance dentro del área del disipador.



Figura 27 Resistencia eléctrica (niquelina)

Cálculos:

Por cada 1000W es 100°C

$$P = 20 \frac{W}{cm} * 110cm = 2200W$$

$$V = 110V$$

$$P = V * I$$

Ecuación 9 Potencia uno

$$I = \frac{P}{V} = \frac{2200W}{110V} = 20A$$

$$P = \frac{V^2}{R}$$

Ecuación 10 Potencia dos

$$R = \frac{V^2}{P} = \frac{110V^2}{2200W} = 5.5\Omega$$

2.3.2 Módulo peltier (TEC1-12715).

La estructura del módulo termoelectrico se describe generalmente como elementos semiconductores de tipo n y tipo p intercaladas entre dos placas de cerámica y conectados en serie. Los elementos son, bombas de calor de estado sólido, sin vibraciones, libres de ruido, donde el calor se desplaza de una superficie a otra cuando se aplica una corriente eléctrica directa en él. Si el calor en el lado caliente se disipa al entorno ambiental por un disipador de calor, este conjunto se convierte en una unidad de refrigeración. No sólo se utiliza para la transferencia de calor, los módulos termoelectricos son ampliamente utilizados para generar energía eléctrica mediante la conversión de calor a corriente eléctrica lo que permite la recuperación de calor residual [25].

El presente proyecto desarrollado ofrece un sistema sin partes móviles, siendo de tamaño pequeño y ligero de peso (ver Figura 28), el módulo termoelectrico podrá ser utilizado en servicios militares, médicos,

industriales, consumidores, laboratorios, electro-óptico y áreas de telecomunicaciones para la refrigeración, calefacción y generación de energía.

Algunas de las características del TEC1-12715 se describen a continuación:

- Variación Temperatura máximas: $< 90\text{ }^{\circ}\text{C}$
- Variación Temperatura mínima: $< -3.5\text{ }^{\circ}\text{C}$
- Potencia: 150W
- Tensión Máxima: 15.4V (La alimentación normal es a 12V DC, pero puede llegar hasta 15.4V DC)
- Corriente: 15.6 A



Figura 28 Módulo peltier

Fuente: [25]

2.3.3 Sistema de refrigeración.

El sistema de refrigeración (ver Figura 29) ayuda al módulo Peltier alcanzar temperaturas bajas hasta 0°C , el mismo que consta de un evaporador que al encontrar un ambiente de baja presión que el fluido refrigerante, pasa del estado líquido para el estado gaseoso, absorbiendo en el proceso calor del ambiente interno del refrigerador.

De tal manera el compresor promueve el bombeo del fluido refrigerante, que al retornar del evaporador en estado gaseoso es succionado y bombeado para el condensador, causando baja presión en el evaporador y alta presión en el condensador, además de elevar aún más la temperatura del gas.

El condensador tiene como principal papel proporcionar la disipación del calor absorbido por el fluido refrigerante a lo largo del sistema de refrigeración. Es en el condensador que el gas recalentado al perder calor por el medio ambiente, pasa del estado gaseoso al estado líquido. En cambio el filtro de secador es un elemento filtrante con material de disecación, con la finalidad de retener impurezas o humedad que pueda haber en el sistema.

El elemento de control tiene como finalidad crear resistencia a la circulación de fluido refrigerante causando una gran diferencia de presión entre condensador y evaporador.



Figura 29 Sistema de refrigeración

Fuente: [26]

2.3.4 Disipador de calor.

Un disipador es un elemento que se utiliza para bajar la temperatura de algunos componentes electrónicos. Su funcionamiento se basa en la segunda ley de la termodinámica, transfiriendo el calor de la parte caliente que se desea disipar al aire. Este proceso permite aumentar la superficie de contacto con el aire permitiendo una eliminación más rápida del calor excedente.

Un disipador extrae el calor del componente que refrigera y lo expulsa al exterior, normalmente al aire. Para ello es necesaria una buena conducción de calor a través del mismo, se fabrican de aluminio por su ligereza, muy importante es su bajo peso, cabe aclarar que el peso es importante ya que la tecnología avanza y por lo tanto se requieren disipadores más ligeros y con eficiencia suficiente para la transferencia de calor hacia el exterior. [27]

En la Figura 30, observamos nuestro disipador de alta potencia, para mejores resultados, se decidió adquirir un reemplazo de un disipador de Drive, ya que éste nos brinda las prestaciones adecuadas para la disipación de calor del generador de temperatura y con mayor ayuda hacia el módulo Peltier.

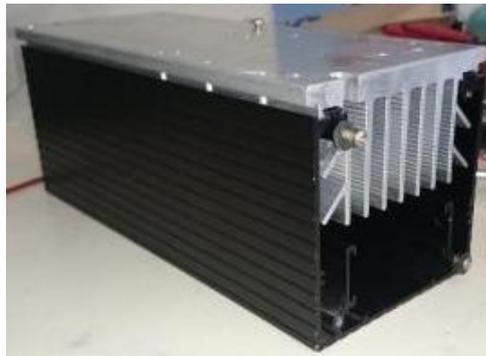


Figura 30 Disipador de calor

Fuente: [27]

2.4 Control PID del generador de temperatura.

Nuestro generador de temperatura, con niquelina y módulo Peltier, disipador de calor y ventiladores, se integra en una sola estructura y está diseñado para trabajar en un rango de temperatura de 0 a 200 °C.

El diagrama de la Figura 31 ilustra el lazo de control, donde, por medio de una consigna ingresada al HMI se envía un valor de Set Point que es ingresado a la tarjeta SMT32F4 donde está desarrollado el sistema de control, el mismo que tiene como objetivo controlar la niquelina y el Peltier para diferentes rangos de temperatura.

Como referencia tendremos la temperatura ambiente dentro de la estructura y cuando necesitemos temperaturas mayores a la ambiente

entrará en funcionamiento la niquelina, mientras que para temperaturas inferiores a la ambiente funciona el Módulo Peltier, se debe aclarar que los ventiladores son utilizados para disipar las temperaturas elevadas generadas por estos dos actuadores; para la realimentación del control usamos como sensor una PT100, con la cual corregiremos el error existente en el proceso.

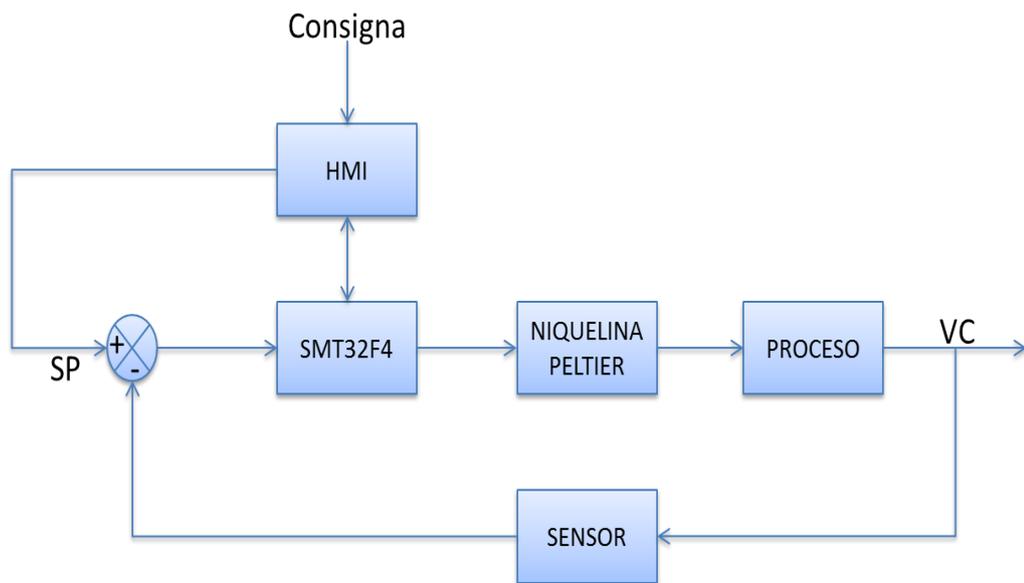


Figura 31 Diagrama de bloques de lazo cerrado

2.4.1 Programación del control PID.

Para la programación de nuestro Control PID se trabajó en Simulink, un entorno de programación visual, que funciona sobre Matlab.

A continuación se describe los pasos del modelo del control PID realizado en diagramas de bloques en Simulink.

Se selecciona y abre el software MATLAB como se indica en la Figura 32 ya sea buscando el software o dando clic en el ícono de acceso rápido del mismo.

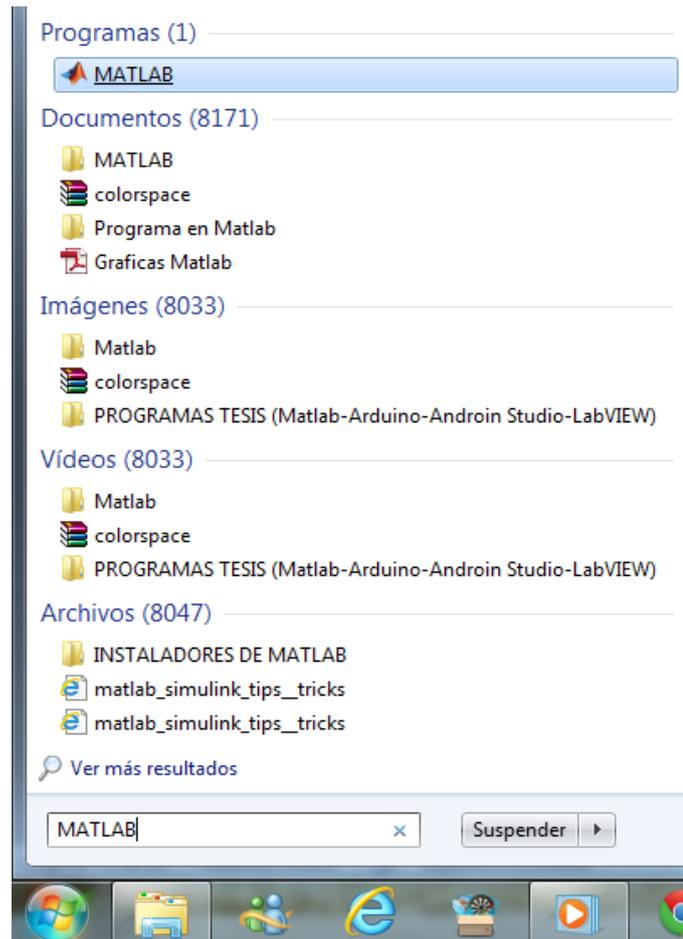


Figura 32 Ingreso al software matlab

En la pestaña de Simulink dar doble clic para ingresar a la ventana de trabajo y crear un nuevo proyecto, como se indica en la Figura 33.

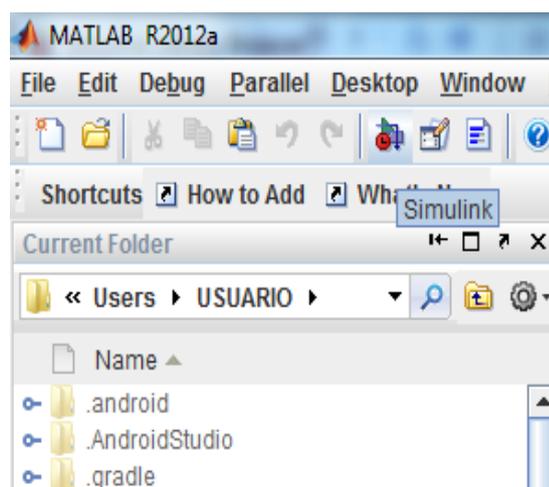


Figura 33 Ingreso ventana simulink

Una vez ingresado a la ventana de Simulink, creamos un nuevo proyecto, clic File – New – Model y nos muestra la ventana de trabajo (ver Figura 34).

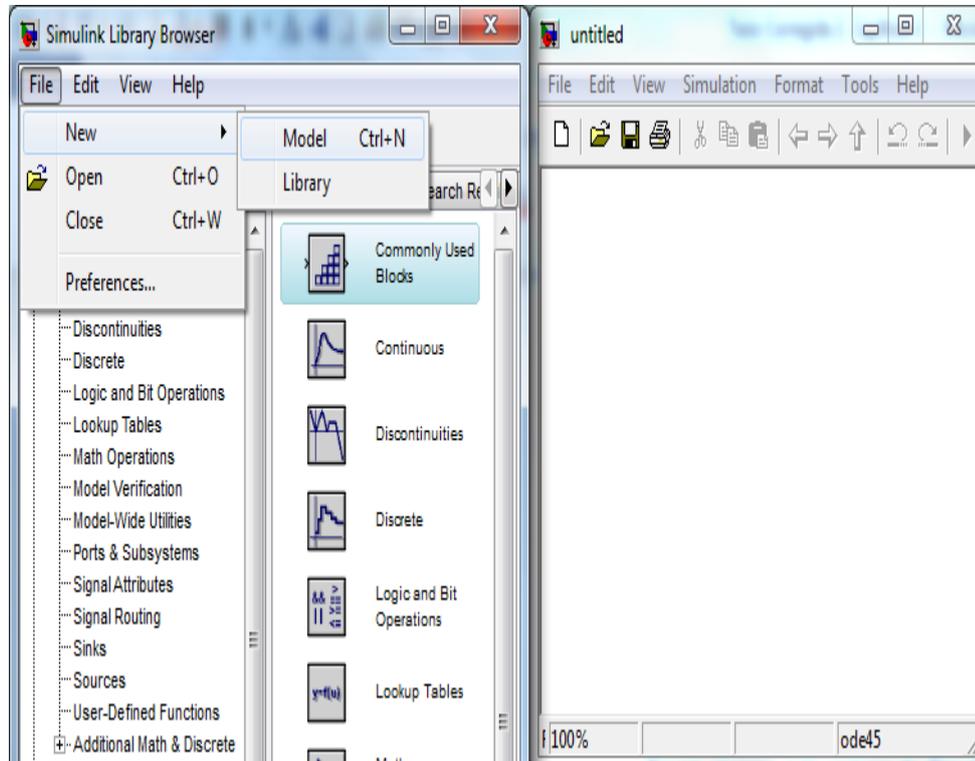


Figura 34 Ventana principal de simulink

A continuación se desplaza hacia la pantalla de trabajo todos los elementos a utilizar (Iconos de: Configuración de la Tarjeta, Recolección de Datos, Transmisión de Datos, Control PID, SPI para Termocuplas y RTDs, UART1 para Arduino, UART3 Comunicación con la Tablet, Variables Globales) como ilustra la Figura 35.

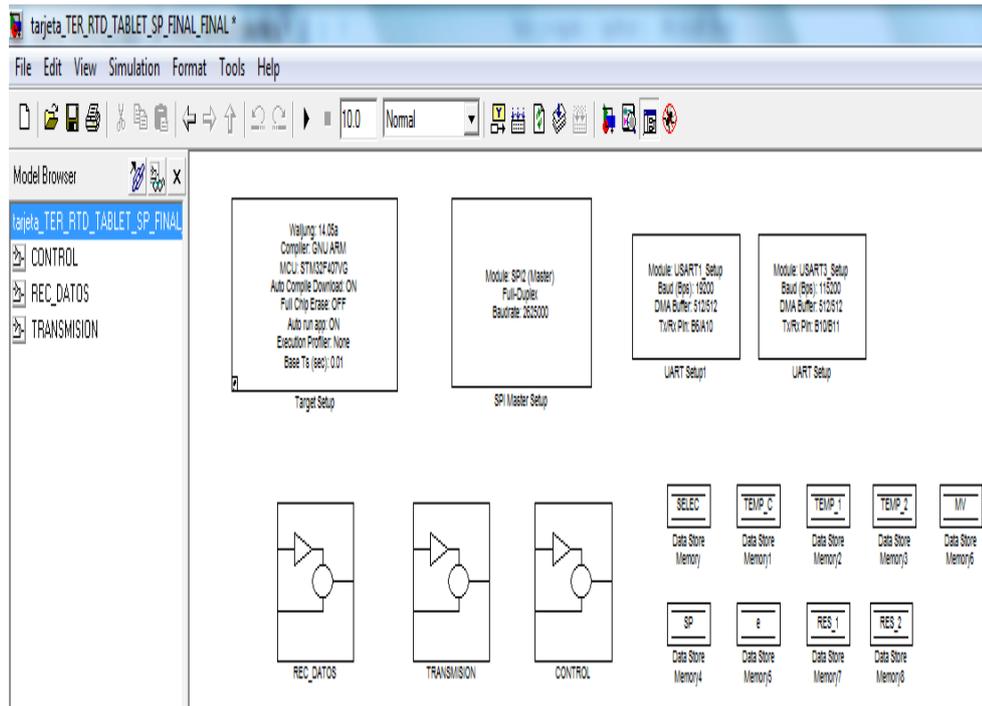


Figura 35 Selección de dispositivos de simulink

Clic en el subsistema del bloque de control, y se desplaza una nueva ventana en la que contiene el diagrama de bloques del control PID como muestra la Figura 36.

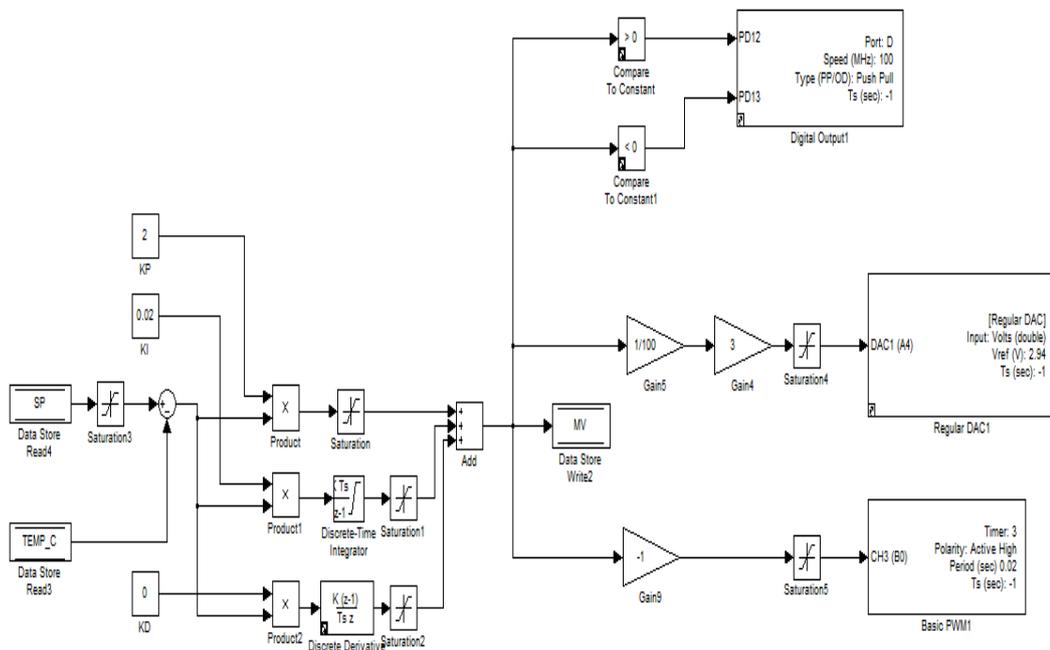


Figura 36 Diagrama de bloques del control PID

La nomenclatura utilizada en el diagrama de bloques es la siguiente:

SP: Este icono Contiene el valor del Set Point.

TEMP_C: Contiene el valor de la temperatura de control.

KP: Es la constante del proporcional.

KI: Es la constante del integral.

KD: Es la constante del derivativo.

ADD: El bloque Suma realiza adición o sustracción de sus entradas. Este bloque puede sumar o restar escalar, vectorial, o entradas de la matriz.

MV: Almacena los datos de la escritura realizada en el proceso.

DAC: Convertidor de digital a analógico (DAC) del bloque para generar señales analógicas.

Los datos del **MV** hay que enviarlos a un **DAC**, antes lo dividimos para 100 y multiplicamos por 3 que es el voltaje y se le agrega una saturación de 2.5 como muestra la Figura 37.

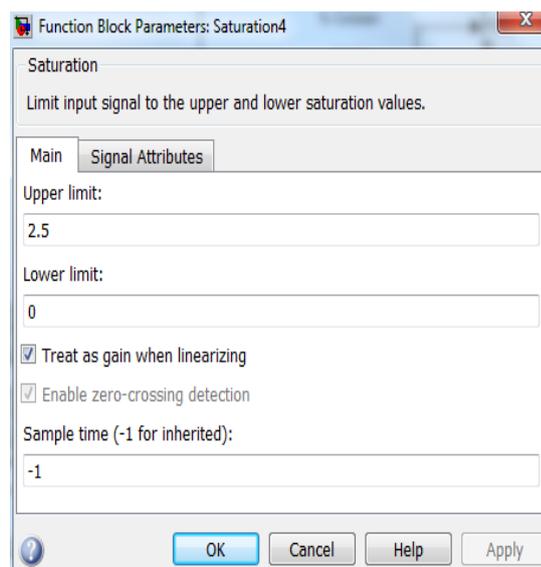


Figura 37 Límite de saturación

Hay tres configuraciones diferentes de RTD. La tarjeta está adaptada para el tipo de RTD (PT100), pero los cables de puente se pueden utilizar en la entrada de la terminal del tornillo para cambiar fácilmente entre las RTD de 2, 3 y 4 hilos para su aplicación. Dicho de otra manera, esta tarjeta de PT100, se puede utilizar para leer los RTD de 2, 3 o 4 hilos con sólo añadir o sustraer cables de puente entre los terminales de entrada del módulo.

En la Figura 39 observamos la conexión para una RTD de 4 hilos.

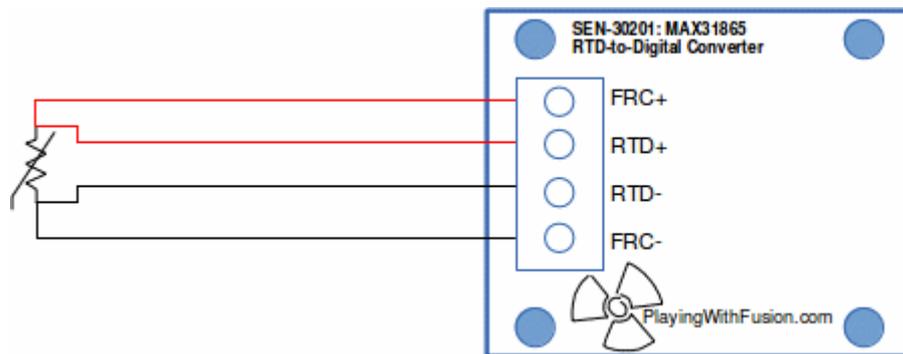


Figura 39 Conexión RTD de 4 hilos

Fuente: [28]

Para la conexión de 3 hilos se observa en la Figura 40 como debemos conectarla.

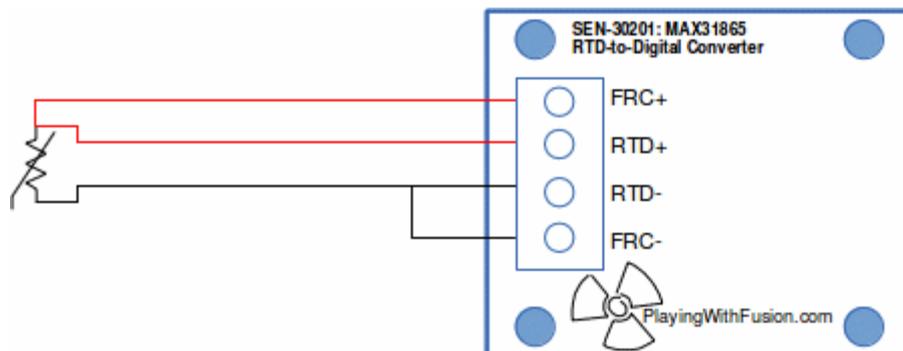


Figura 40 Conexión RTD de 3 hilos

Fuente: [28]

Finalmente para una conexión de dos hilos será necesario realizar las conexiones como se muestra en la Figura 41.

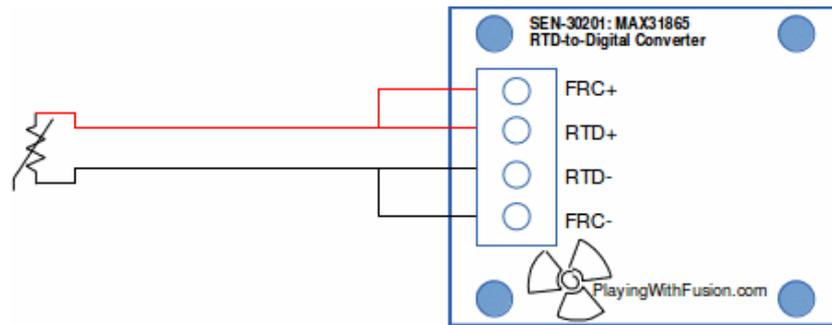


Figura 41 Conexión RTD de 2 hilos

Fuente: [28]

2.5.2 MAX31855J - MAX31855K para termocuplas.

Los MAX31855J - MAX31855K realizan la compensación de junta fría y digitaliza la señal de una termocupla tipo K, J. Los datos se emiten en una forma de 14 bits, SPI. Este convertidor tiene una resolución de 0.25 °C, permite lecturas de hasta +1800 °C y tan bajas como -270 °C, y exhibe una precisión de ± 2 °C para temperaturas que van desde -200 °C a +700 °C para termopares tipo K y J [29].

El dispositivo incluye hardware de acondicionamiento de señal para convertir la señal del termopar en un voltaje compatible con los canales de entrada del ADC. Las entradas T + y T- se conectan al circuito interno que reduce la introducción de errores de ruido a partir de los hilos del termopar (ver Figura 42 y Figura 43). Antes de convertir los voltajes termoeléctricos en los valores de temperatura equivalentes, es necesario compensar la diferencia entre el lado termopar unión fría (temperatura ambiente dispositivo) y una referencia virtual 0 °C. Para un termopar tipo K y J, la tensión de cambios es por sobre 41 $\mu\text{V}/^\circ\text{C}$.

Se tomó en cuenta que la MAX31855 asume una relación lineal entre la temperatura y voltaje. Debido a que todos los termopares exhiben un cierto nivel de no linealidad, aplicar la corrección apropiada para datos de salida del dispositivo.

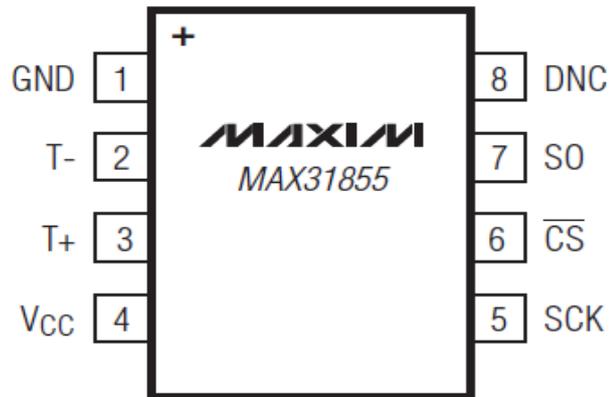


Figura 42 Configuración MAX31855

Fuente: [29]

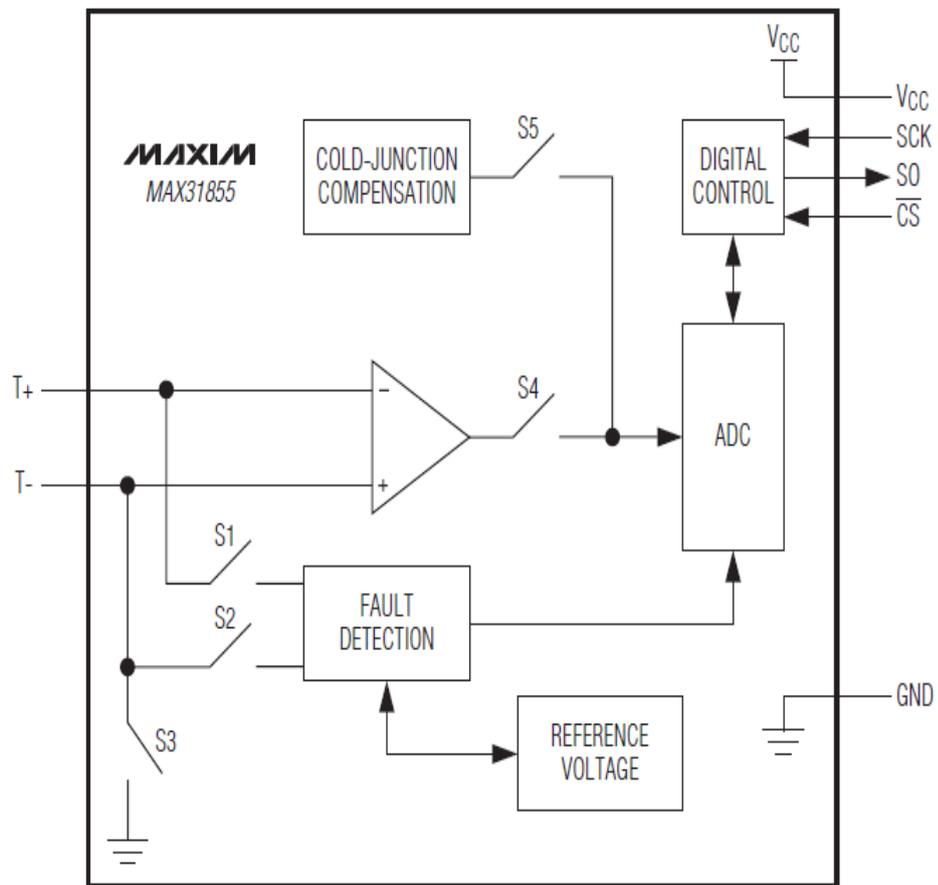


Figura 43 Diagrama de bloques MAX31855

Fuente: [29]

En la Tabla 8 se explica la función de cada pin del MAX31855.

Tabla 8
Descripción de pines MAX31855

PIN	NOMBRE	FUNCIÓN
1	GND	Tierra.
2	T-	Entrada negativa de la Termocupla. No conectar a GND.
3	T+	Entrada positiva de la Termocupla.
4	VCC	Voltaje de alimentación.
5	SCK	Señal de reloj.
6	\overline{CS}	Activa o Desactiva la interfaz serial.
7	SO	Salida de datos.
8	DNC	No conectar.

Fuente: [29]

2.5.3 Adquisición de datos de RTD y termocuplas.

Para la adquisición de datos se utilizó la siguiente secuencia de programación en la tarjeta STM32F407VG, como se ilustra en la Figura 44.

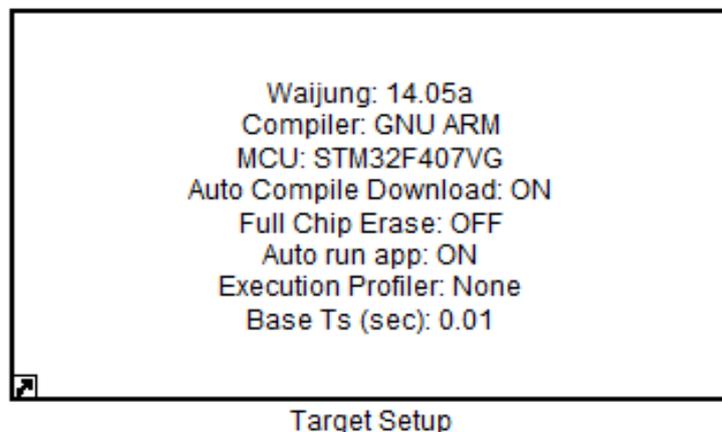


Figura 44 Diagrama de bloque STM32F407VG

Al dar doble clic en el diagrama de bloque de la tarjeta se despliega el diagrama de bloque de configuración de la tarjeta STM32F407VG, donde nos muestra los siguientes parámetros (ver Figura 44).

Copiler: GNU ARM (Compilador del microcontrolador).

MCU: STM32F407VG (LQFP100); STM32F4DISCOVERY (Escoger el microcontrolador).

CLOCK CONFIGURATION: Default.

Programmer/Debugger: ST_Link (Es el programador).

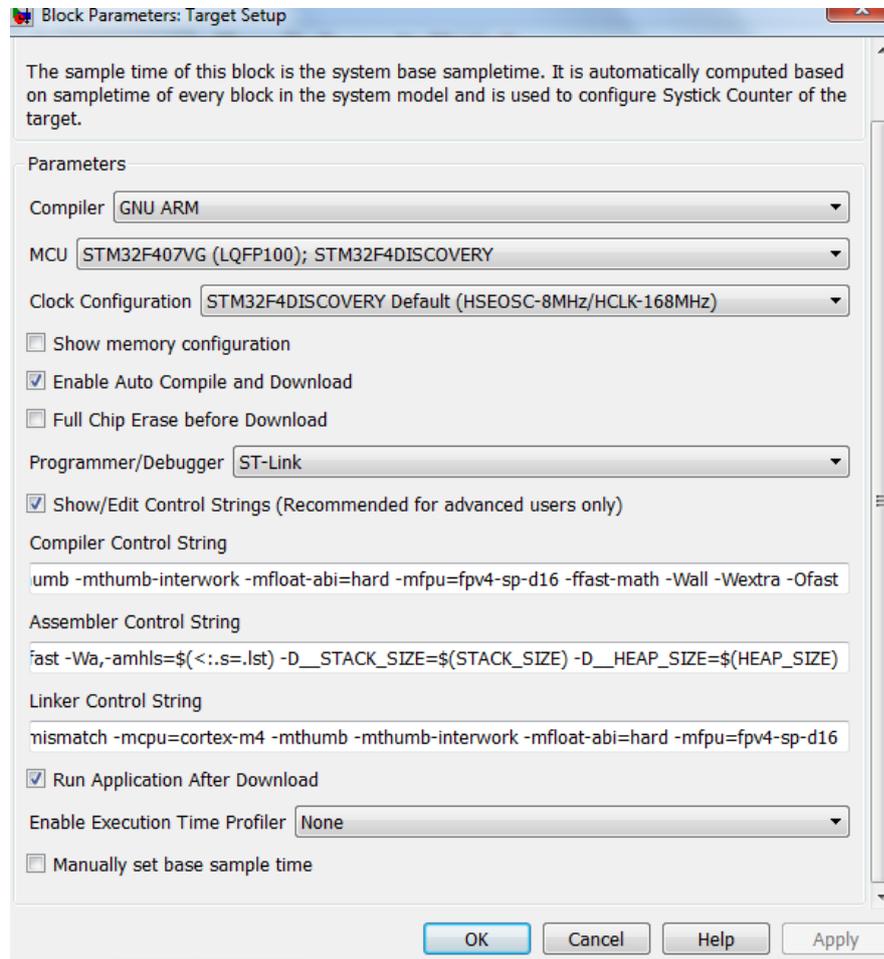


Figura 45 Configuración de la tarjeta STM32F407VG

Para la adquisición de los datos de las Termocuplas y PT100 utilizamos el siguiente Bloque SPI (ver Figura 46).



Figura 46 Adquisición datos (RTD-termocuplas)

Una vez ingresado al bloque se nos desplegará una pantalla de configuración para el SPI (interfaz de periféricos serie), este es un protocolo síncrono. La sincronización y la transmisión de datos se realizan por medio de 4 señales (ver Figura 47).

SCLK (Clock): Es el pulso que marca la sincronización. Con cada pulso de este reloj, se lee o se envía un bit.

MOSI (Master Output Slave Input): Salida de datos del Master y entrada de datos al Slave. También llamada SIMO.

MISO (Master Input Slave Output): Salida de datos del Slave y entrada al Master. También conocida por SOMI.

SS/Select: Para seleccionar un Slave, o para que el Master le diga al Slave que se active. También llamada SSTE.

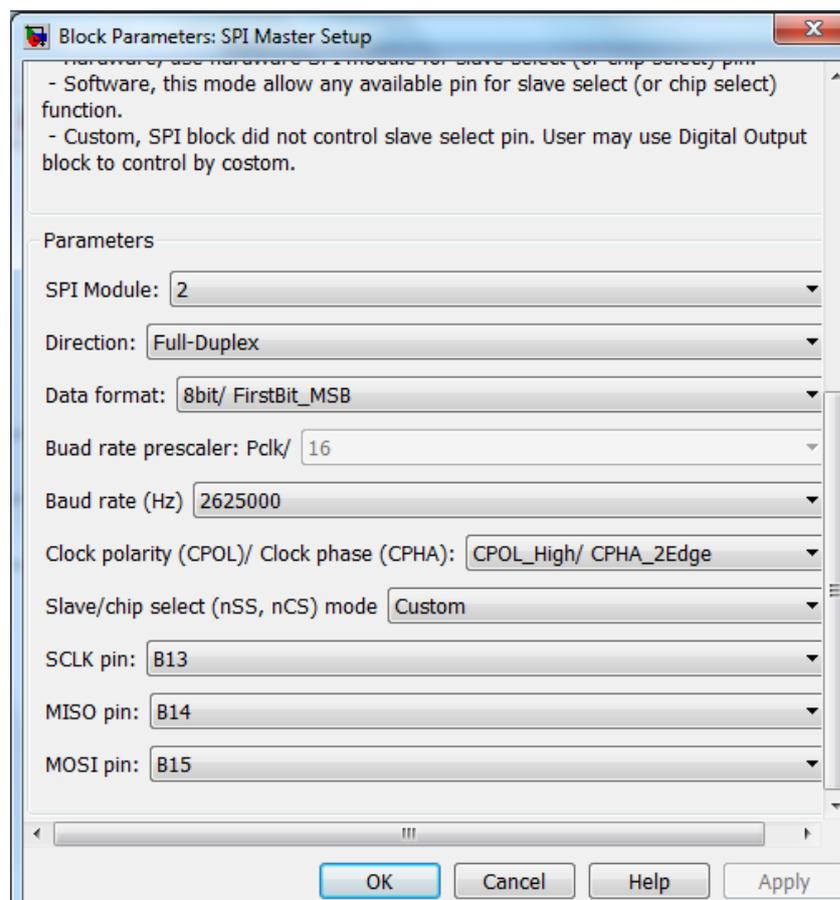


Figura 47 Configuración protocolo SPI

El siguiente bloque, UART1 es para la comunicación con el Arduino, para que se comunique con LabVIEW y realice la red Ethernet (ver Figura 48).

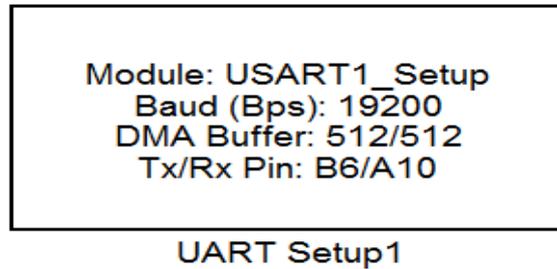


Figura 48 Comunicación arduino

El bloque del UART3 es para la comunicación con la Tablet (ver Figura 49).

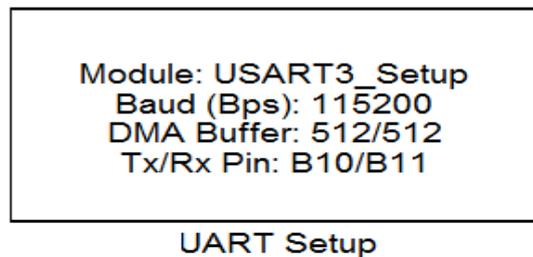


Figura 49 Comunicación tablet

Los siguientes bloques son las variables globales, a las que se pueden acceder desde cualquier bloque, ya sea desde el bloque de escritura o desde el bloque de lectura (ver Figura 50).

SELEC: Para seleccionar el tipo de sensor.

TEMP_C: Es la temperatura de control.

TEMP_1: Es la temperatura del sensor uno.

TEMP_2: Es la temperatura del sensor dos.

MV: Es la variable manipulada.

SP: Set Point.

e: Error.

RES_1 y RES_2: Muestra da las unidades de medida de los sensores, si es PT100 y Termistores nos da resistencia en cambio sí es Termocuplas o LM35 son voltaje.

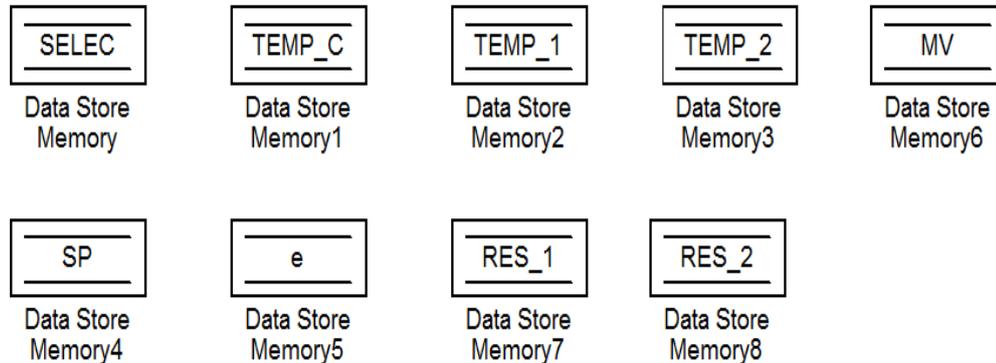


Figura 50 Variables globales

Para la recolección de datos utilizamos el bloque de la Figura 51.

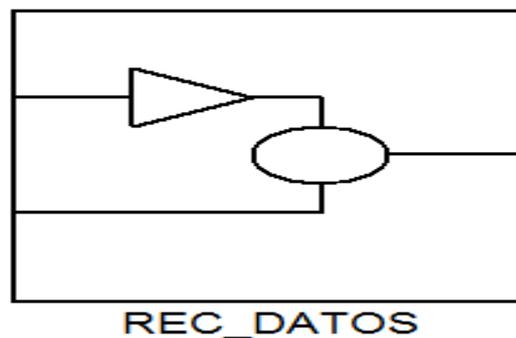


Figura 51 Recolección de datos

Damos doble clic en el bloque de la Figura 51 y se despliega el diagrama de bloque de inicialización y datos de los sensores de temperaturas de la PT100 y Termocuplas (ver Figura 52).

Para el envío y recepción de datos por protocolo SPI se necesita del bloque de inicialización para la tarjeta MAX31865.

El **Step** es el icono que tiene dos estados 0 y 1; cuando está en 0 realiza la inicialización y cuando está en 1 realiza la recolección de datos.

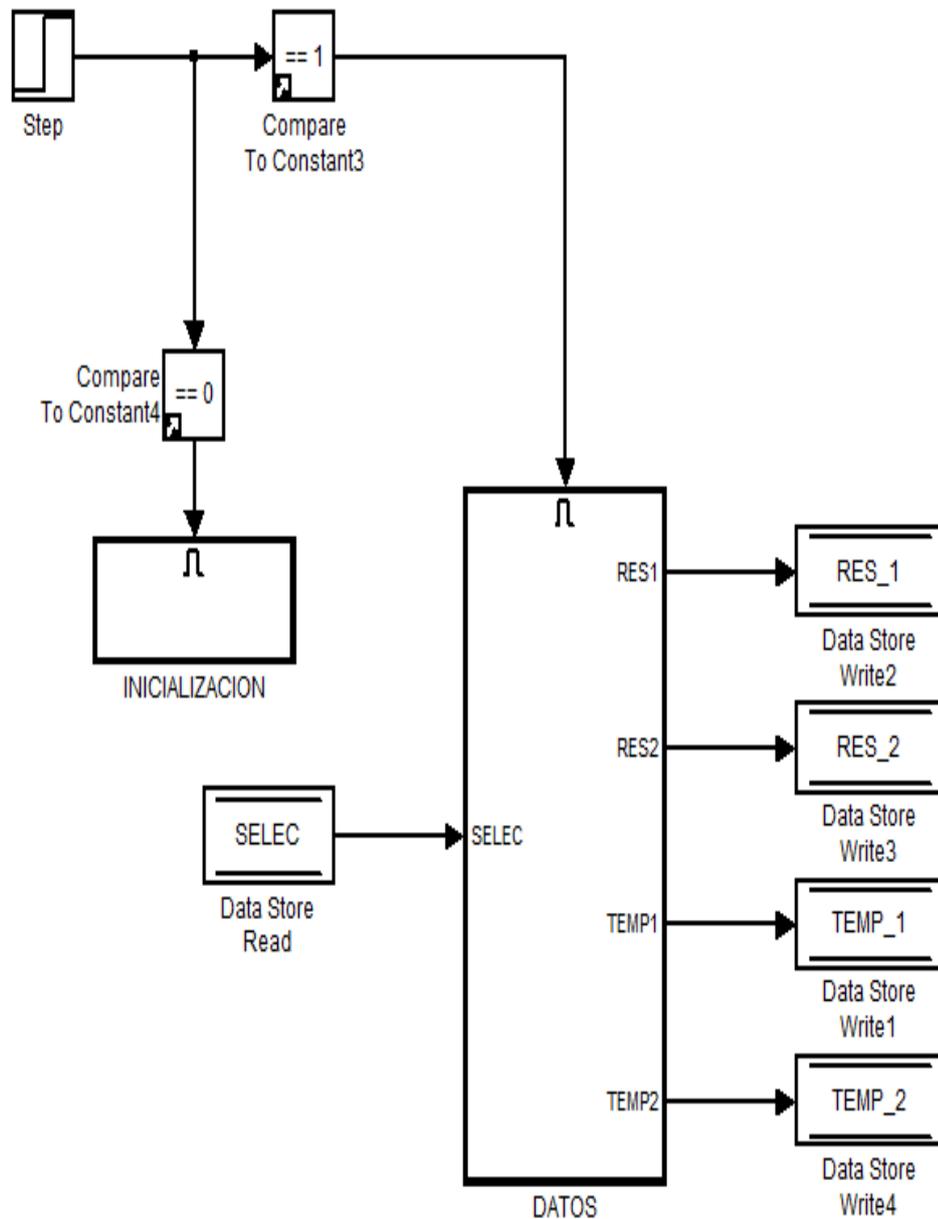


Figura 52 Recolección de datos (PT100-termocuplas)

Dar doble clic en el icono de inicialización y se despliega un diagrama de bloques el que contiene un contador de 13 elementos, como se muestra en las Figura 53, Figura 54, Figura 55.

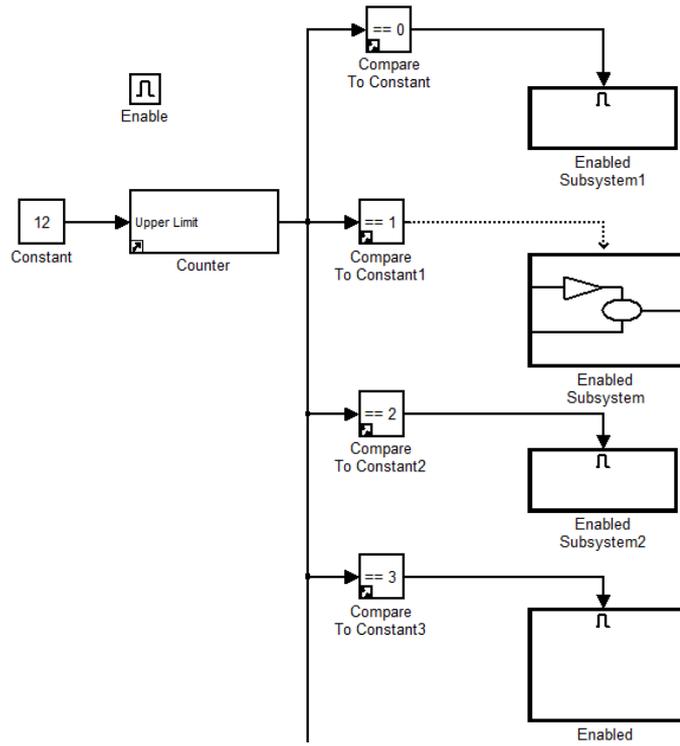


Figura 53 Contador (a)

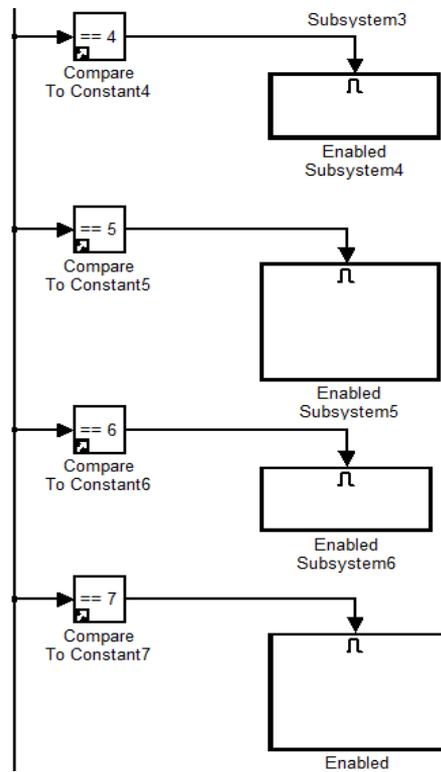


Figura 54 Contador (b)

CONTINÚA ➡

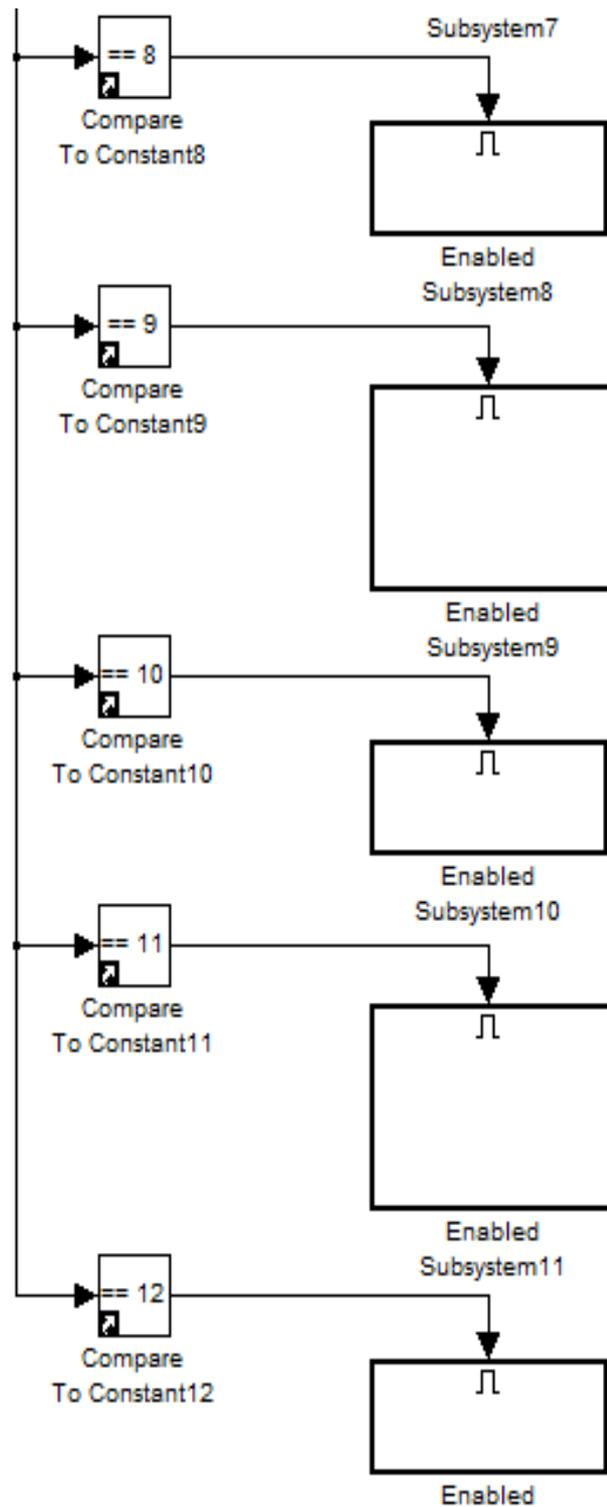


Figura 55 Contador (c)

Nota: Los contadores mencionados anteriormente en las Figura 53, Figura 54, Figura 55; es un solo contador de 13 elementos que contienen la configuración de la PT100 y Termocuplas.

Cuando el contador está en (**==0**), realiza la siguiente configuración de la PT100 número 1 (PC1). Teniendo en cuenta, que el botón de selección de los sensores envía una acción SPI para que el Chip SELEC se coloque en bajo (ver Figura 56).

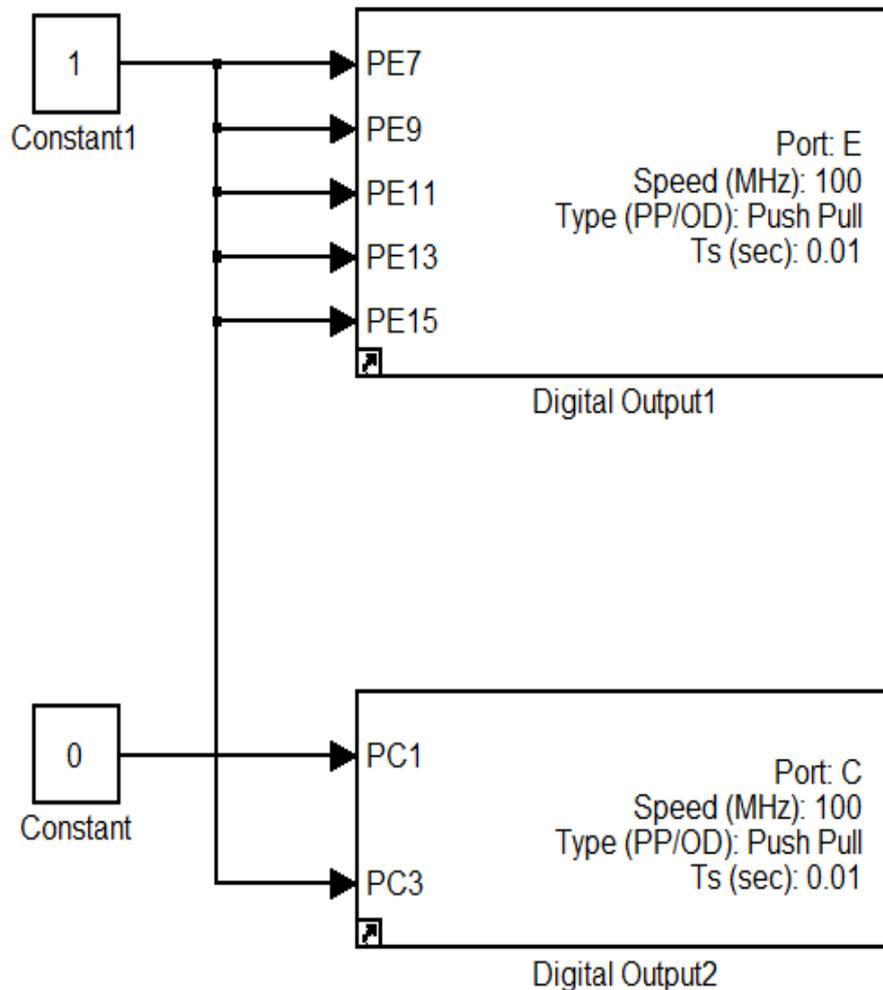
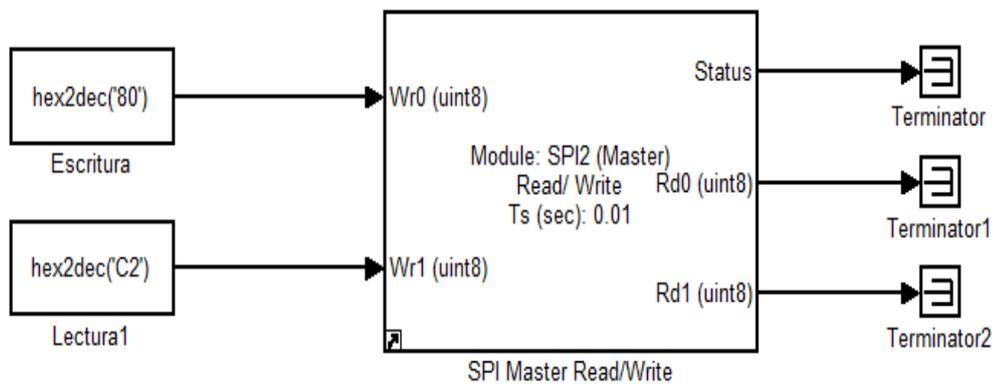


Figura 56 Diagrama de activación de PT100 uno

El contador en (**==1**), realiza la configuración para la PT100 uno. En esta ocasión debemos enviar las palabras de control el hex2dec('80') y hex2dec('C2') que nos da el datasheet. Una vez enviada las palabras de control inmediatamente se envían los datos (ver Figura 57).



Write config to IC
 bit 7: Vbias -> 1 (ON)
 bit 6: conversion mode -> 1 (AUTO)
 bit 5: 1-shot -> 0 (off)
 bit 4: 3-wire select -> 0 (2/4 wire config)
 bit 3-2: fault detection cycle -> 0 (none)
 bit 1: fault status clear -> 1 (clear any fault)
 bit 0: 50/60 Hz filter select -> 0 (60 Hz)

Figura 57 Configuración de la PT100 uno

Cuando en contador está en (**==2**), realiza la configuración para la PT100 dos (PC3) (ver Figura 58).

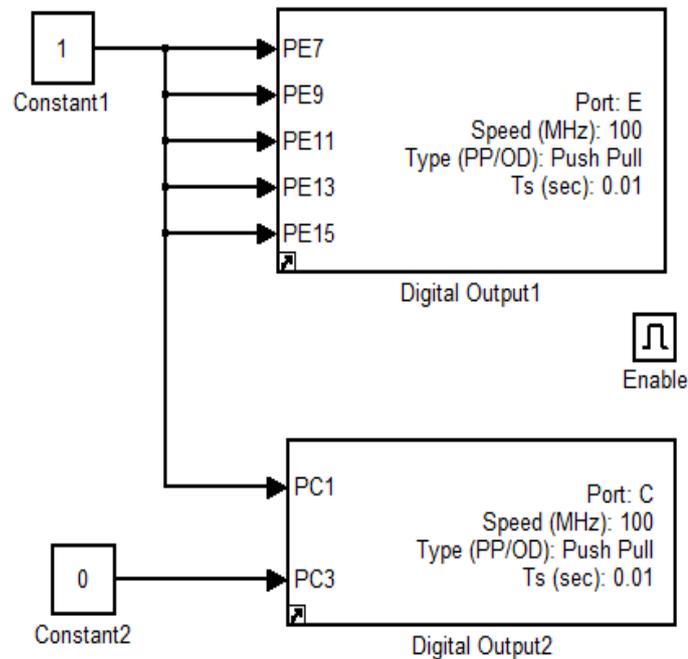


Figura 58 Diagrama de activación PT100 dos

El contador cuando está en (**==3**), realiza la misma configuración de la PT100 uno que se indica en la Figura 57.

El contador cuando está en (**==4**), realiza la configuración de activación de la placa de control de la PT100 tres (PE15) (ver Figura 59).

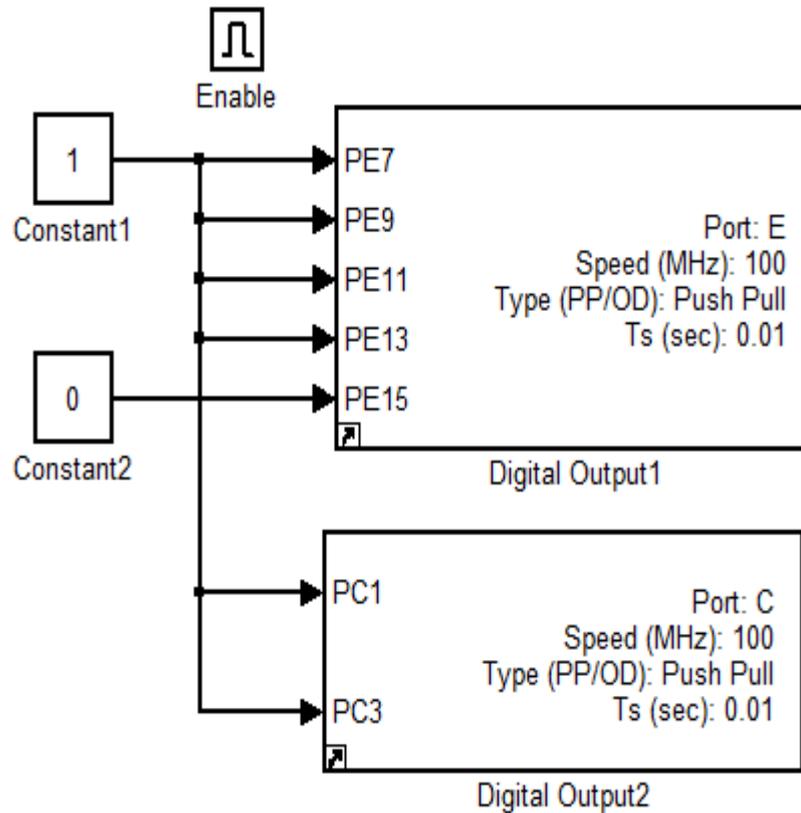


Figura 59 Diagrama de activación PT100 tres

El contador en (**==5**), realiza la misma configuración de la Figura 57, con las mismas palabras de control.

El contador en (**==6**), realiza la configuración de activación para la Termocupla J2 (PE9) como ilustra la Figura 60.

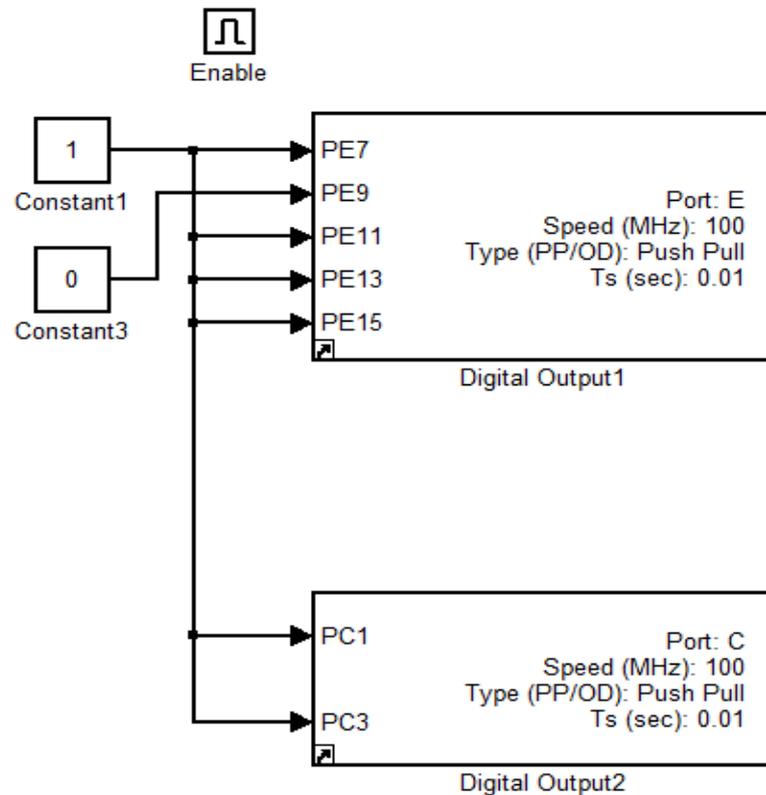


Figura 60 Diagrama de activación termocupla J2

El contador en (**==7**), debemos tener en cuenta que las termocuplas no necesitan inicialización, por lo tanto no hay que configurarlas; porque las termocuplas ya envían datos (ver Figura 61).

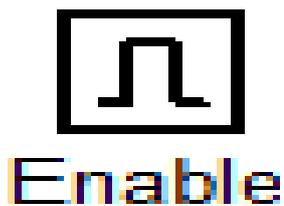


Figura 61 On termocuplas

El contador en (**==8**), realiza la configuración de activación para las Termocuplas k (PE11) como ilustra la Figura 62.

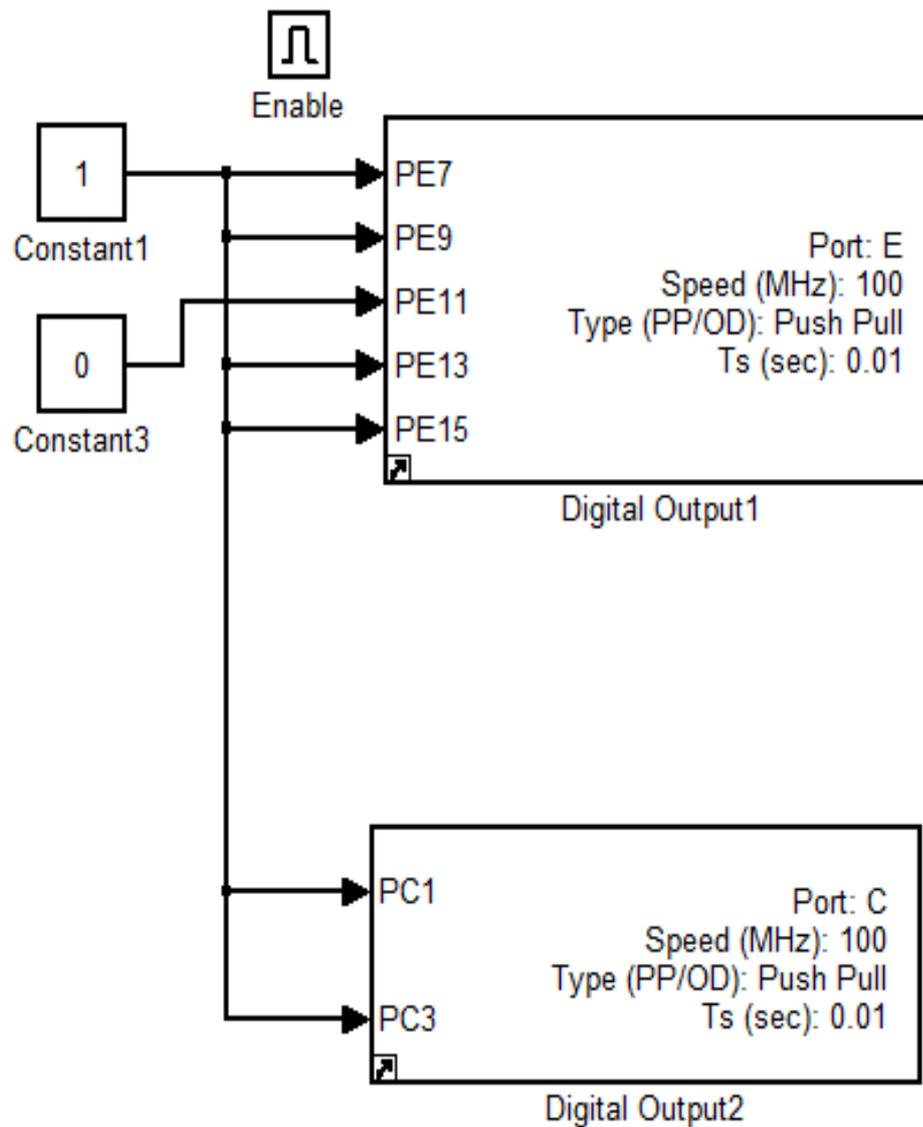


Figura 62 Diagrama de activación termocupla K1

El contador en (**==9**), debemos tener en cuenta que las termocuplas no necesitan inicialización, solo habilitación como se muestra en la Figura 61.

El contador en (**==10**), realiza la configuración de activación para las Termocuplas K (PE13) como ilustra la Figura 63.

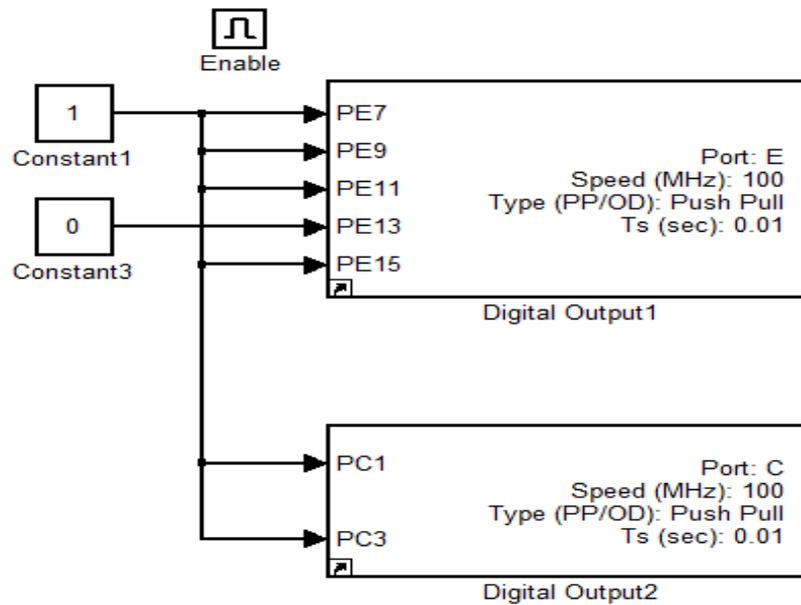


Figura 63 Diagrama de activación termocupla K2

El contador en (**==11**), solo habilitan las termocuplas como se muestra en la Figura 61.

El contador en (**==12**), realiza la configuración de activación para la Termocupla J1 (PE7) como ilustra la Figura 64.

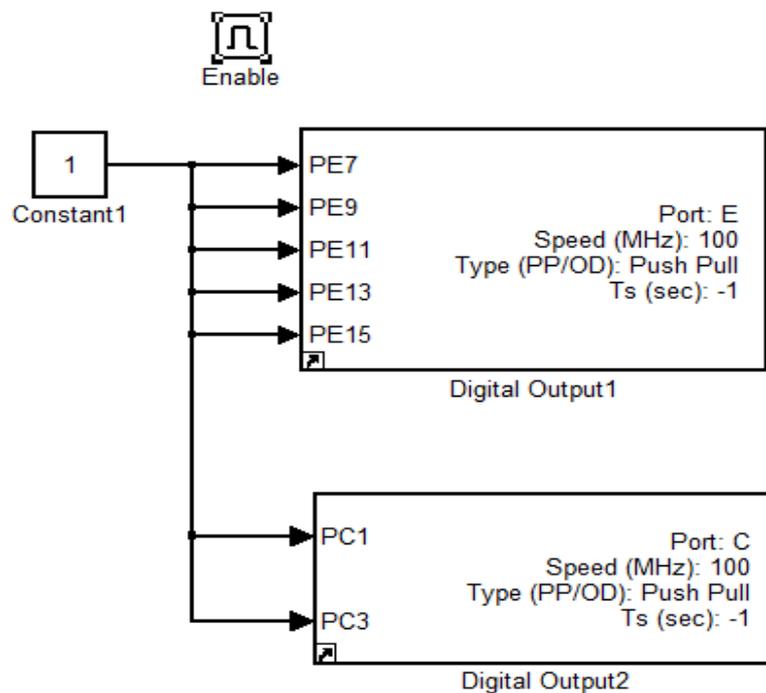


Figura 64 Diagrama de activación termocupla J1

Con doble clic en el icono de **Datos** de la Figura 52, aquí se nos desplegará el diagrama de bloques de un contador de 15 elementos para la recolección de datos de la PT100 y Termocuplas, como se muestra en las Figura 65, Figura 66, Figura 67, Figura 68, Figura 69.

En este bloque de la Figura 52, mediante el **SELEC** nos indica el tipo de sensor que va a leer y a la salida nos da las unidades de medida de la PT100 y Termistores en resistencia, mientras que para el LM35 y Termocuplas entregan voltaje; a la salida también nos muestra la temperatura 1 del sensor 1 y la temperatura 2 del sensor 2.

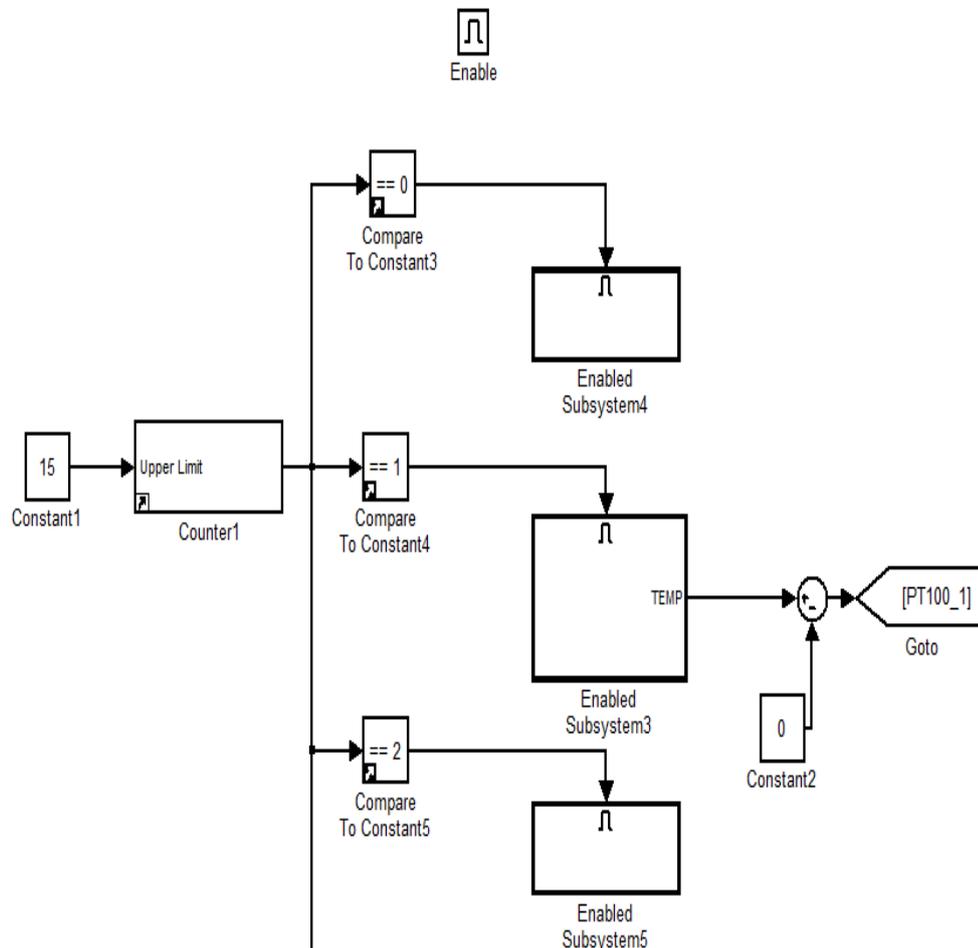


Figura 65 Contador (a)

CONTINÚA ⇨

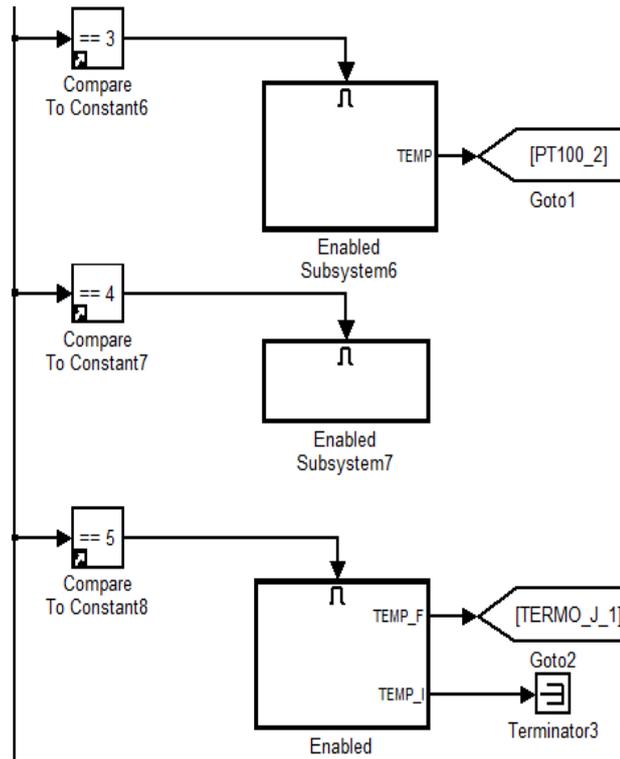


Figura 66 Contador (b)

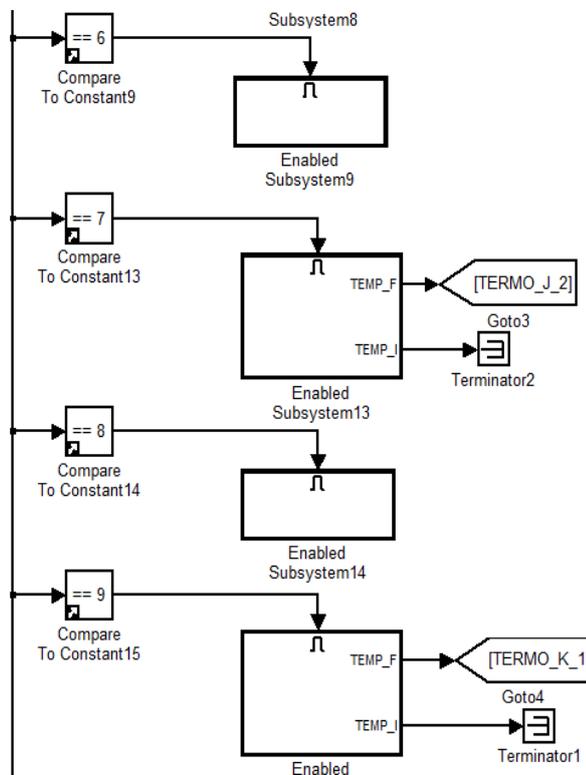


Figura 67 Contador (c)

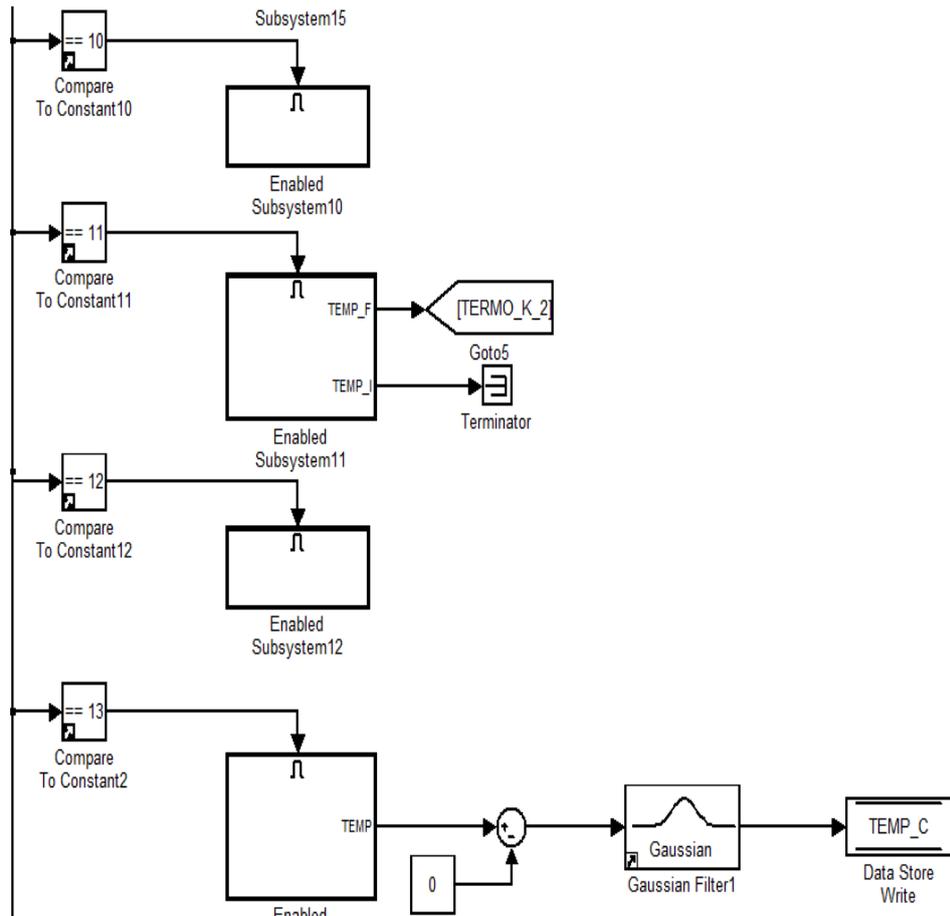


Figura 68 Contador (d)

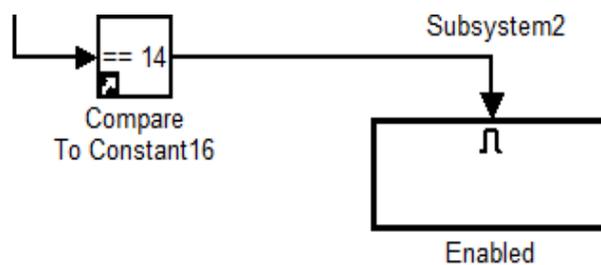


Figura 69 Contador (e)

Nota: Los contadores mostrados en las figuras anteriores, es un solo contador de 15 elementos que contienen los datos de la PT100 y Termocuplas.

Contador en (**==0**), damos clic y se despliega el diagrama de bloques de la configuración de la PT100 uno (ver Figura 70).

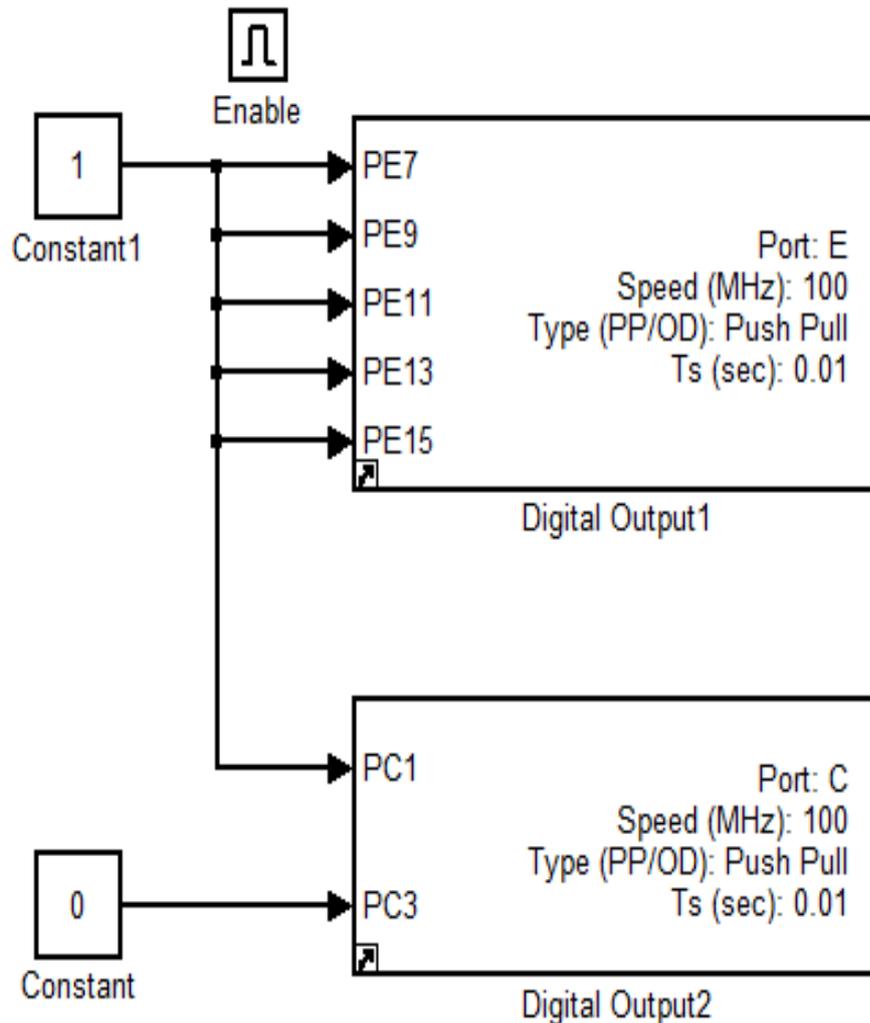


Figura 70 Configuración de datos PT100 uno

Contador en (**==1**), lee y envía las palabras de control `hex2dec('80')` y `hex2dec('C2')` y el SPI master nos devuelve dos bytes que contiene la temperatura, pero eso dos bytes con resolución de 14 bits, entonces con **Shift Arithmetic** lo que hace es desplazar 8 lugares y sumar con el otro dato para obtener 16bits y a la salida multiplicar por una ganancia de 0.017 y luego sumar por una constante de -278.58 dando lugar a un número entero y obtener un número de Temperatura, por último el **Discrete FIR Filter** filtrar la señal para no tener variación en los valores de temperatura (ver Figura 71).

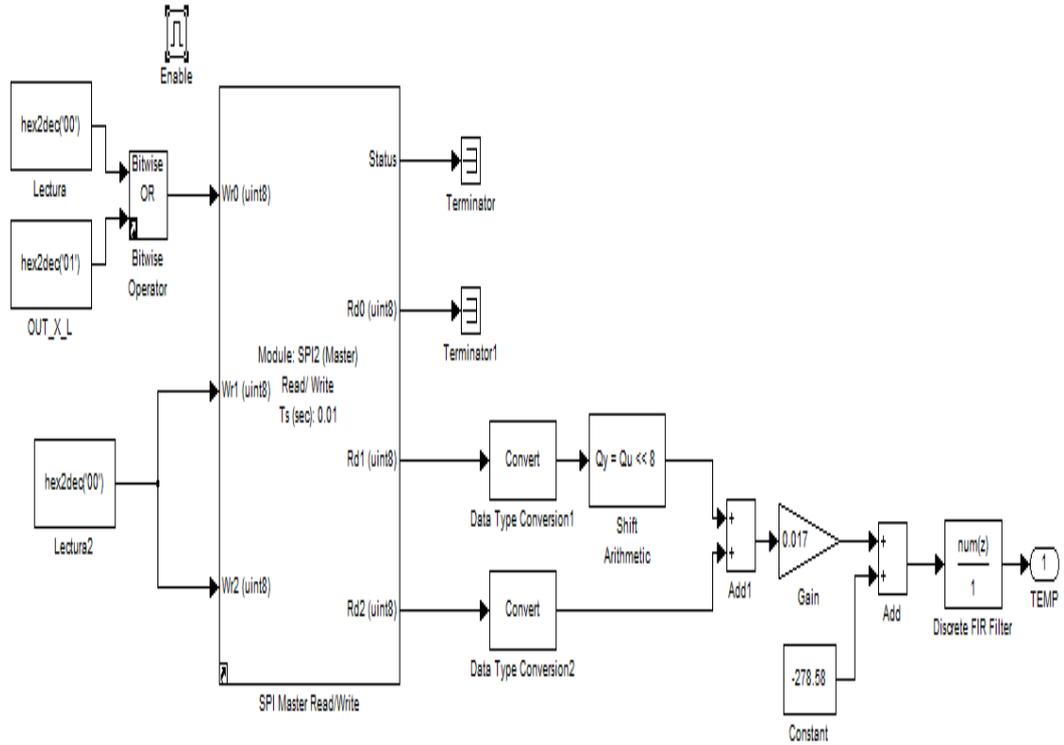


Figura 71 Diagrama de bloques para obtener la temperatura uno

Contador en (==2), damos clic y se despliega la configuración del diagrama de bloques de la PT100 dos (ver Figura 72).

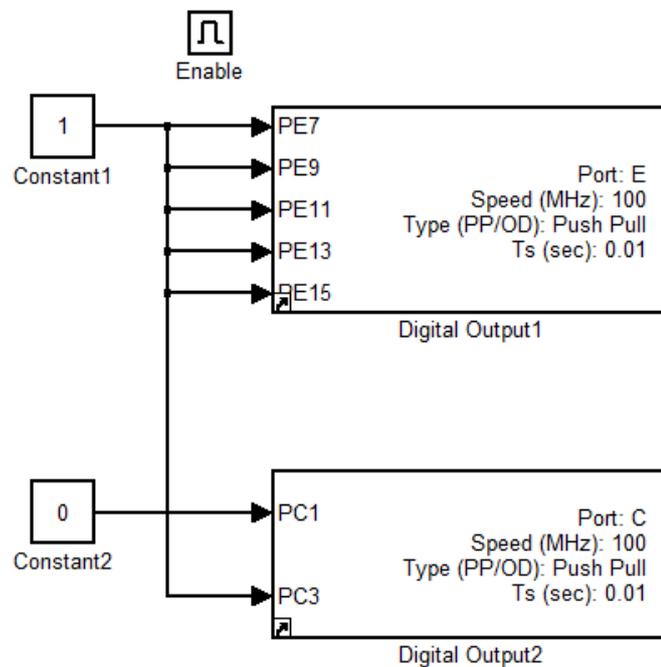


Figura 72 Configuración de datos PT100 dos

Contador en (**==3**), obtenemos el mismo diagrama de bloque para la PT100 dos como muestra la Figura 71, teniendo en cuenta que se está escribiendo por el mismo bus y el que da la ubicación por el cual estamos trabajando es el Chip SELEC.

Contador en (**==4**), se tiene la configuración del diagrama de bloques de la Termocupla J1 (ver Figura 73).

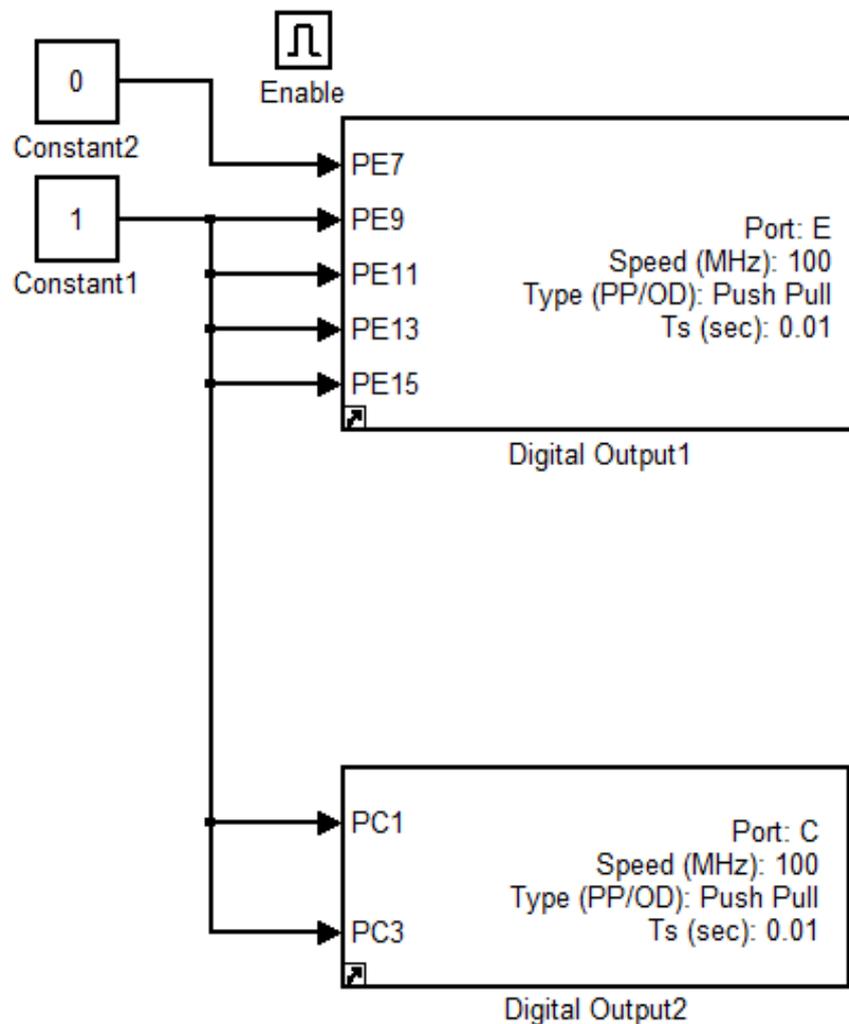


Figura 73 Configuración de datos termocupla J1

Contador en (**==5**), realiza la configuración de la Termocupla tipo J1. Donde la temperatura de la termocupla es relativa, lo que hace la termocupla es medir la diferencia de la temperatura de la junta fría y la temperatura del generador (ver Figura 75).

El **Chip SELEC** por si solo nos envía la temperatura absoluta ósea la temperatura final.

El **MATLAB FUNCTION** nos envía un formato de complemento A2; de un entero de complemento A2 a un double para enviar valores negativos.

Clic en **MATLAB FUNCTION** para revisar el programa del complemento A2 que se presenta en la Figura 74.

```
function x = twos2dec(t2)
x = t2;
xneg = log2(x) >= 14 - 1;
if xneg
    x=-double(bitcmp(uint16(x), 14) + 1);
end
end
```

Figura 74 Complemento A2

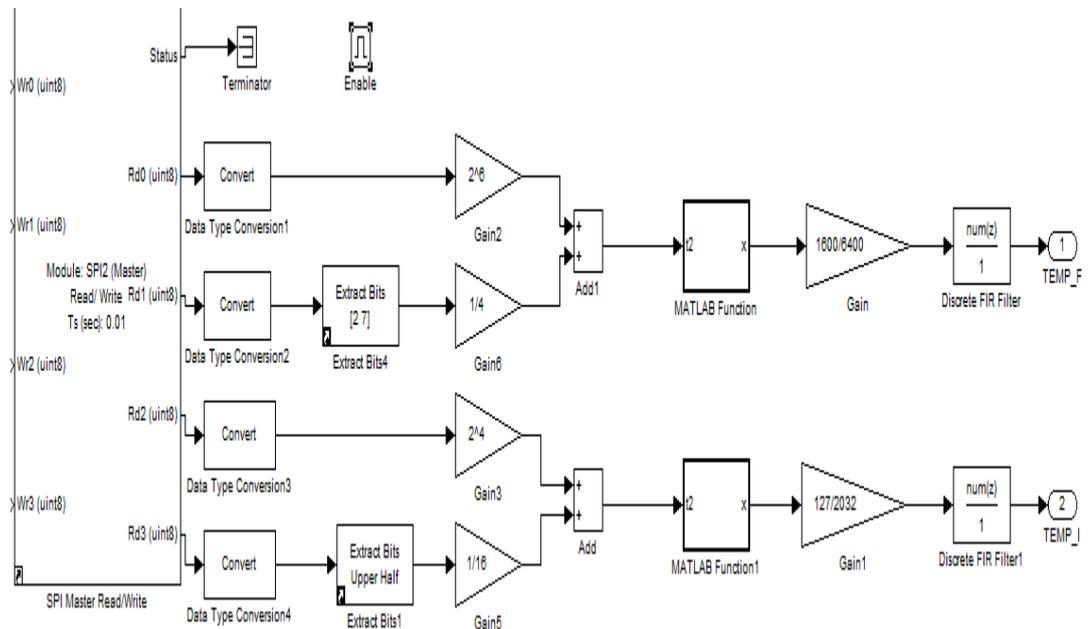


Figura 75 Diagrama de bloques de datos termocupla J1

Contador en **(==6)**, indica la configuración de la Termocupla J2 (ver Figura 76).

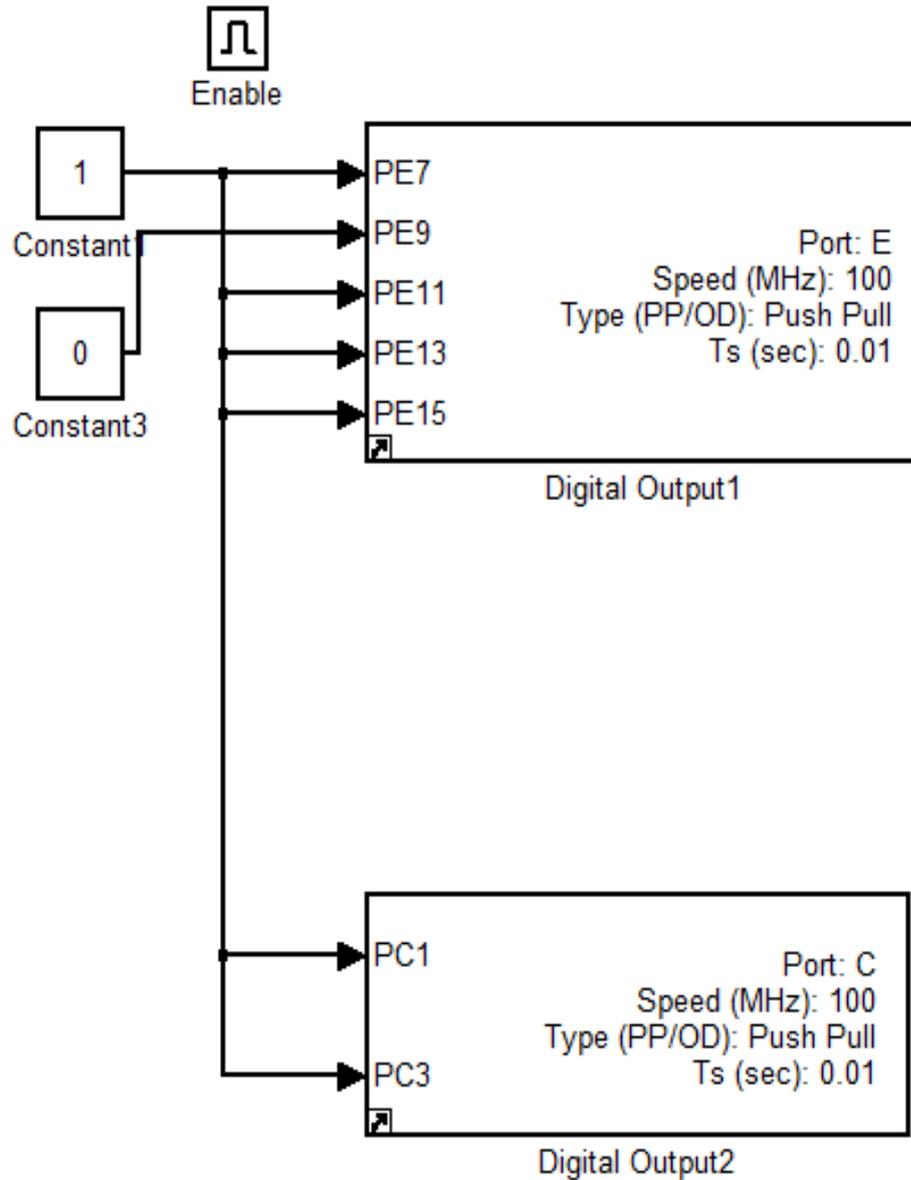


Figura 76 Configuración de datos termocupla J2

Contador en (**==7**), realiza la configuración de la Termocupla tipo J2. Donde el diagrama de bloques es el mismo de la Termocupla J1 de la Figura 75.

Contador en (**==8**), ejecuta la configuración de datos de la Termocupla tipo K1 (ver Figura 77).

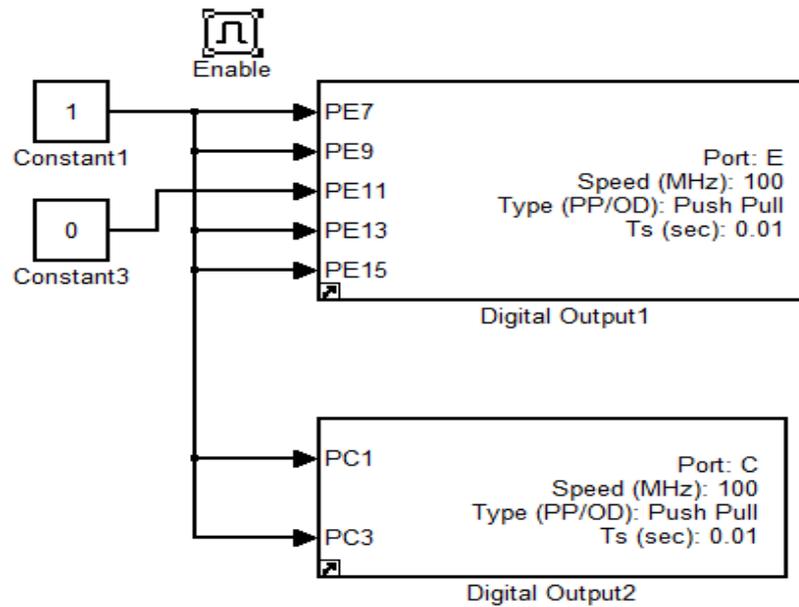


Figura 77 Configuración de datos termocupla K1

Contador en (**==9**), efectúa la configuración de la Termocupla tipo K1. Donde el diagrama de bloques es el mismo de la Termocupla J1 de la Figura 75.

Contador en (**==10**), realiza la configuración de datos de la Termocupla tipo K2 (ver Figura 78).

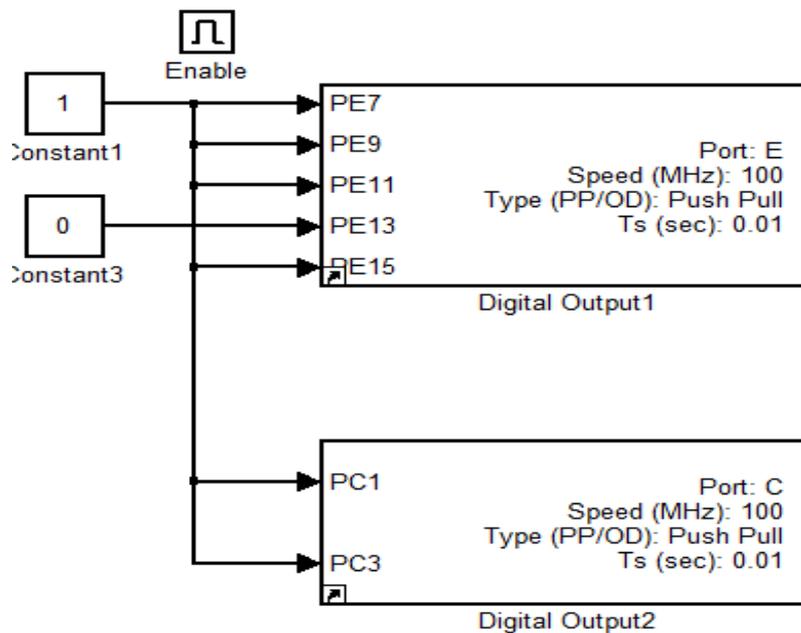


Figura 78 Configuración de datos termocupla K2

Contador en (**==11**), se encuentra la configuración de la Termocupla tipo K2. Donde el diagrama de bloques es el mismo de la Termocupla J1 de la Figura 75.

Contador en (**==12**), se encuentra la configuración de la Temperatura de control (ver Figura 79).

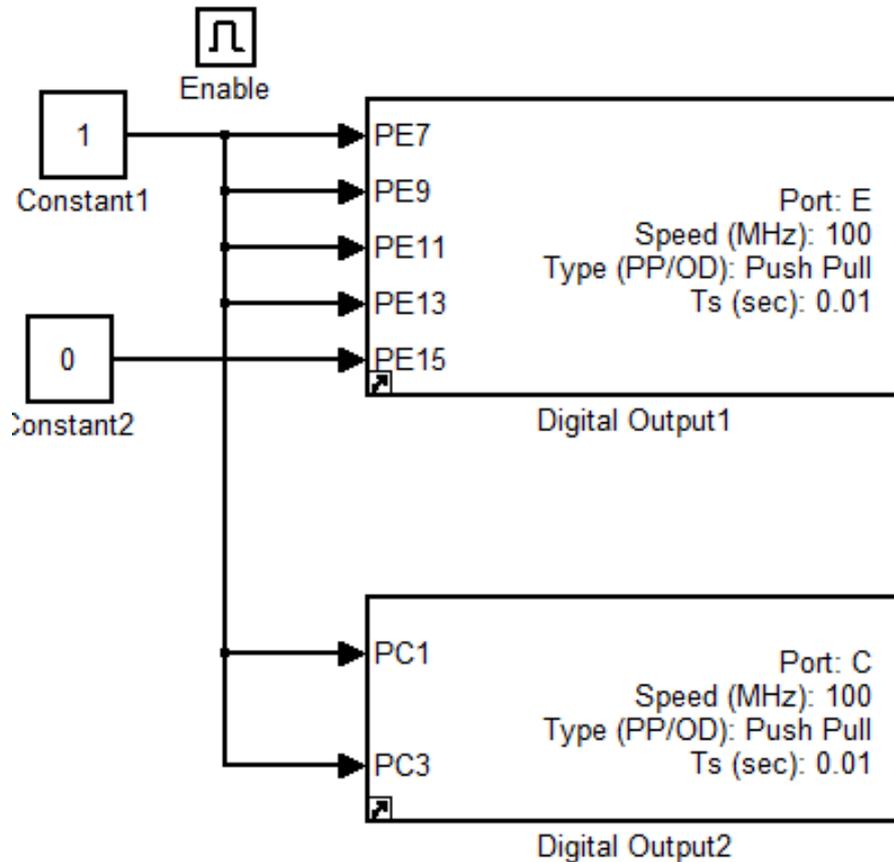


Figura 79 Configuración de datos temperatura control

Contador en (**==13**), se encuentra la configuración de la Temperatura de Control. Donde el diagrama de bloques es el mismo de la PT100 uno, de la Figura 72.

Contador en (**==14**), realiza la configuración del envío de datos de la PT100 y Termocuplas J, K (ver Figura 80).

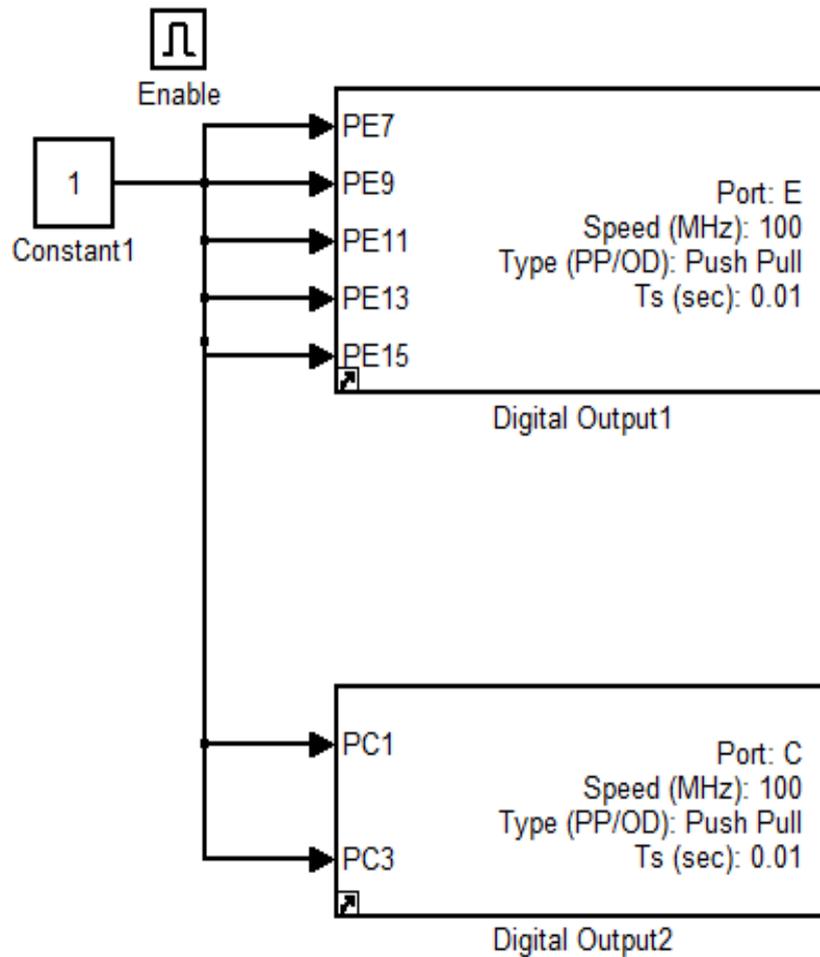


Figura 80 Envío de datos PT10 - termocuplas

2.5.4 MCP6004 para termistores y LM35.

Se ha seleccionado el Microchip MCP6004 de la familia de amplificadores operacionales el cual está diseñado específicamente para aplicaciones de propósito general. Esta familia tiene un ancho de banda 1MHz de ganancia y margen de fase de 90°. También mantiene un margen de fase de 45° con una carga capacitiva de 500 pF. Esta familia funciona a partir de una única tensión de alimentación tan bajo como 1.8V. El MCP6004 está disponible en los rangos de temperaturas industriales (-40°C a 85°C) y extendidas (-40°C a 125°C), con un rango de potencia de suministro de 1.8V a 6.0V. (Ver Figura 81)

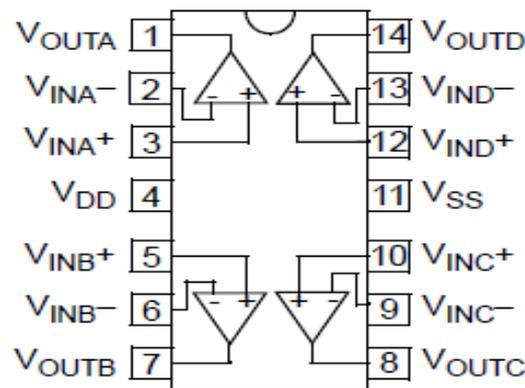


Figura 81 Conexión MCP6004

Fuente: [30]

2.5.5 Adquisición de datos para termistores y LM35.

Para la adquisición de datos de Termistores y LM35 se debe tener en cuenta que sus salidas son señales analógicas, por lo tanto utilizamos un ADC (Analógico Conversor Digital).

Para adquirir los datos de los sensores LM35, se utiliza **Discrete FIR Filter**, este bloque es para filtrar la señal, luego se realiza un escalamiento para pasar a voltaje $\left(\frac{2,94}{4095}\right)$, y se multiplica por una ganancia de 100 (100mV/°C), como se muestra en la Figura 82.

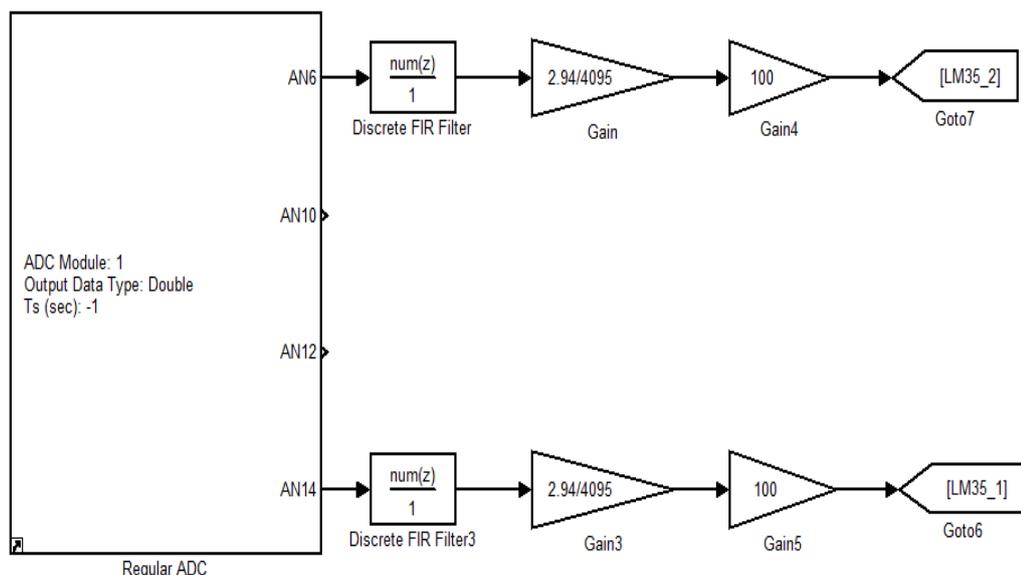


Figura 82 Adquisición de datos del LM35

Para la adquisición de datos de los Termistores (1K Ω , 5K Ω , 10K Ω), primero se utiliza un ADC, luego un **Discrete FIR Filter** y por último un escalamiento de $\left(\frac{2,94}{4095}\right)$ como muestra la Figura 83.

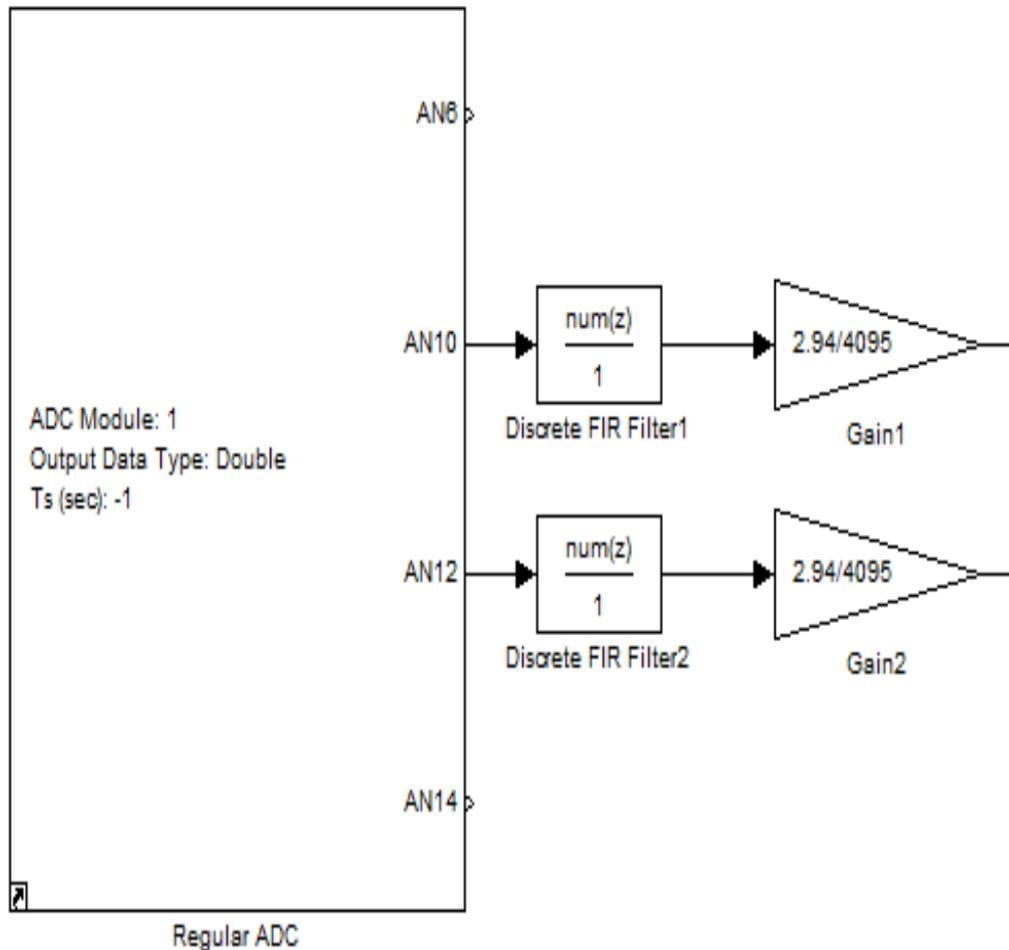


Figura 83 Adquisición de datos termistores

Las señales acondicionadas de los termistores son procesadas mediante MATLAB, para esto se utiliza **MATLAB FUNCTION**, como muestra la Figura 84.

Con esta función se realiza la programación para calcular la temperatura de cada uno de los termistores ya sea de 1K Ω , 5K Ω , 10K Ω , como se indica en la Figura 85, Figura 86, Figura 87.

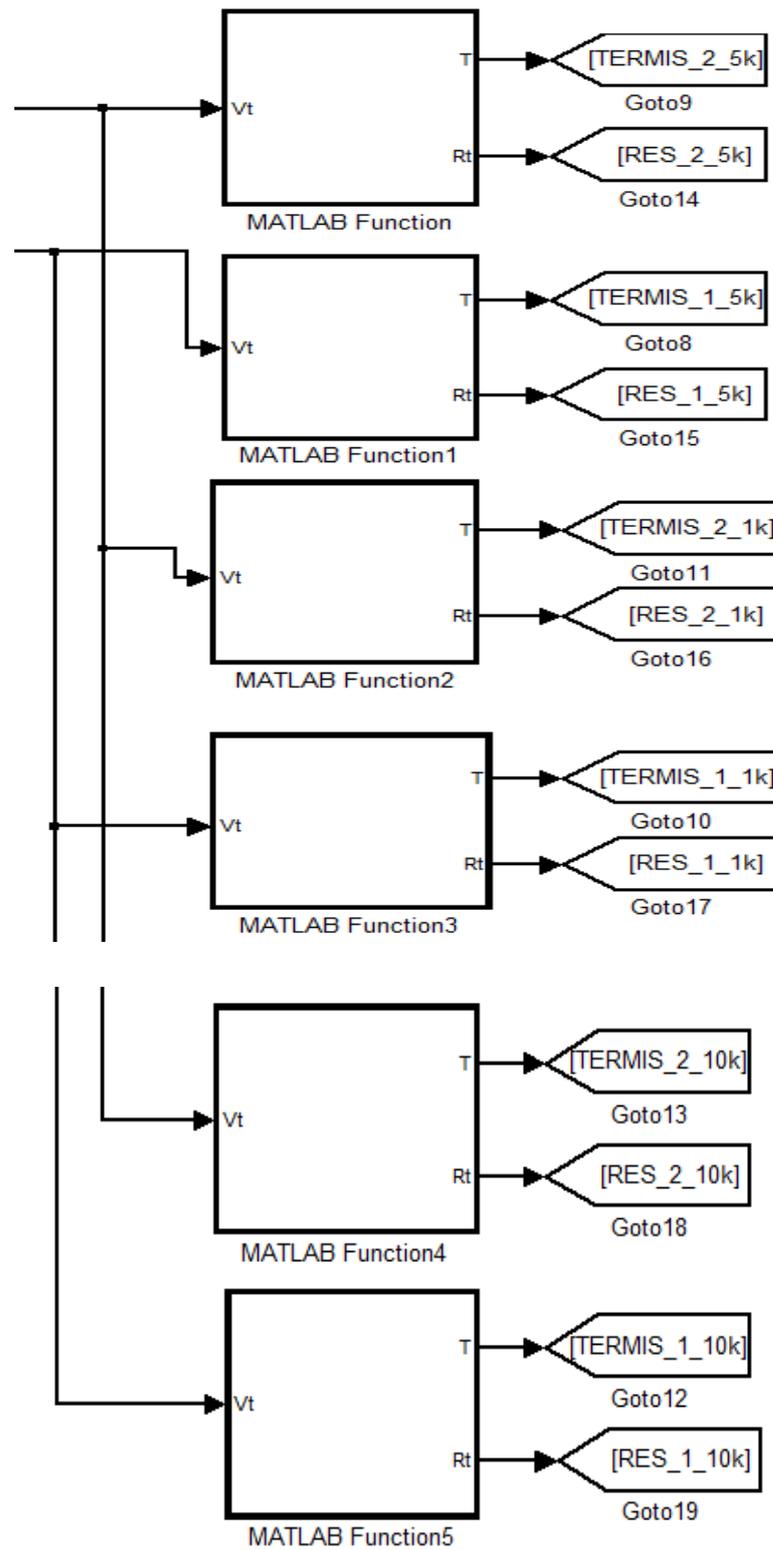


Figura 84 Matlab function termistores

```
function [T,Rt] = temp_5k(Vt)
R = 10000;
V = 4.42;
Rt = Vt*R/(V-Vt);
T = 668.9*Rt^(-0.1525)-157.6;
```

Figura 85 Cálculo temperatura termistor (5k)

```
function [T,Rt] = temp_1k(Vt)
R = 10000;
V = 4.42;
Rt = Vt*R/(V-Vt);
T = 486*Rt^(-0.145)-153.2;
```

Figura 86 Cálculo temperatura termistor (1k)

```
function [T,Rt] = temp_10k(Vt)
R = 10000;
V = 4.42;
Rt = Vt*R/(V-Vt);
T = 601.7*Rt^(-0.142)-138;
```

Figura 87 Cálculo temperatura termistor (10k)

La temperatura calculada en el **MATLAB FUNCTION**, de los pares de sensores (LM35, PT100, Termocupla Tipo J, k, Termistores 1kΩ, 5kΩ, 10kΩ), se aplican a un **switch** con **SELEC** que empieza en uno y un bloque de saturación con siete como elemento máximo, al final se le da una ganancia de 100, (ver Figura 88), la Figura 89 para los sensores de prueba.

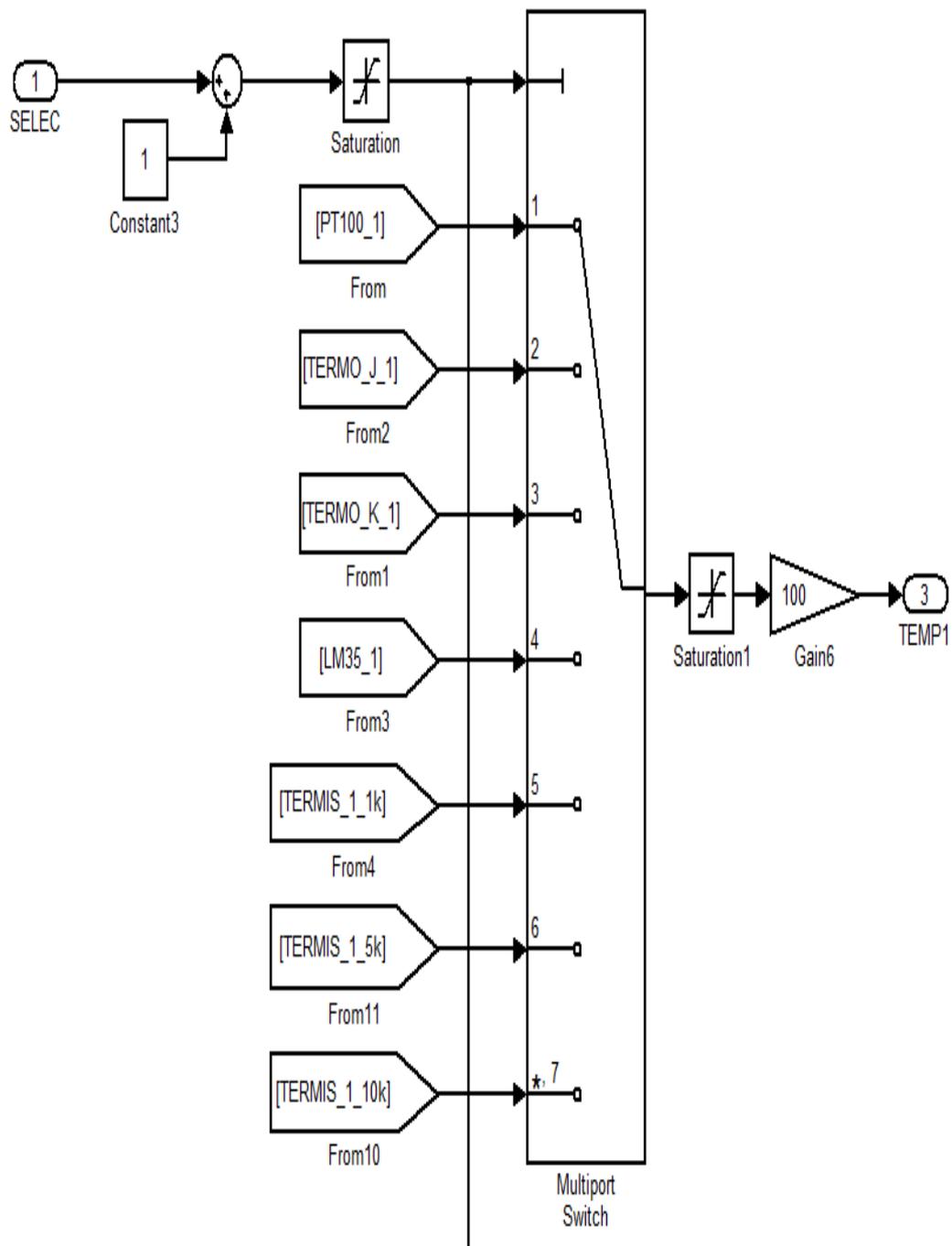


Figura 88 Diagrama de bloques de los sensores uno

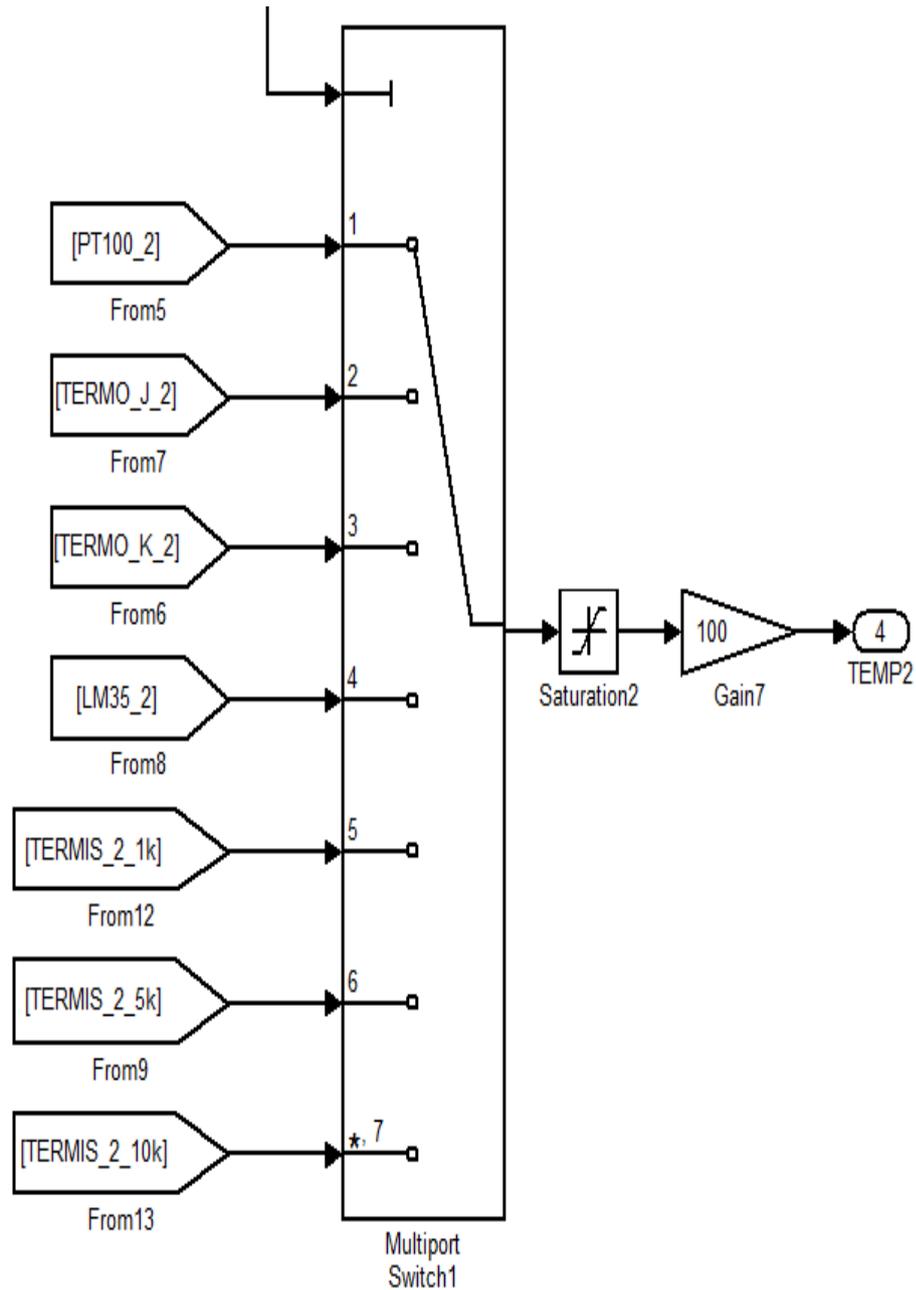


Figura 89 Diagrama de bloques de los sensores dos

Para obtener las resistencias y los voltajes equivalentes a las resistencias o voltajes de la PT100, termistor, LM35 o termocupla para los dos sensores de prueba se utiliza **MATLAB FUNCTION**, cuyo diagrama de bloques se indica en la Figura 90 y Figura 91.

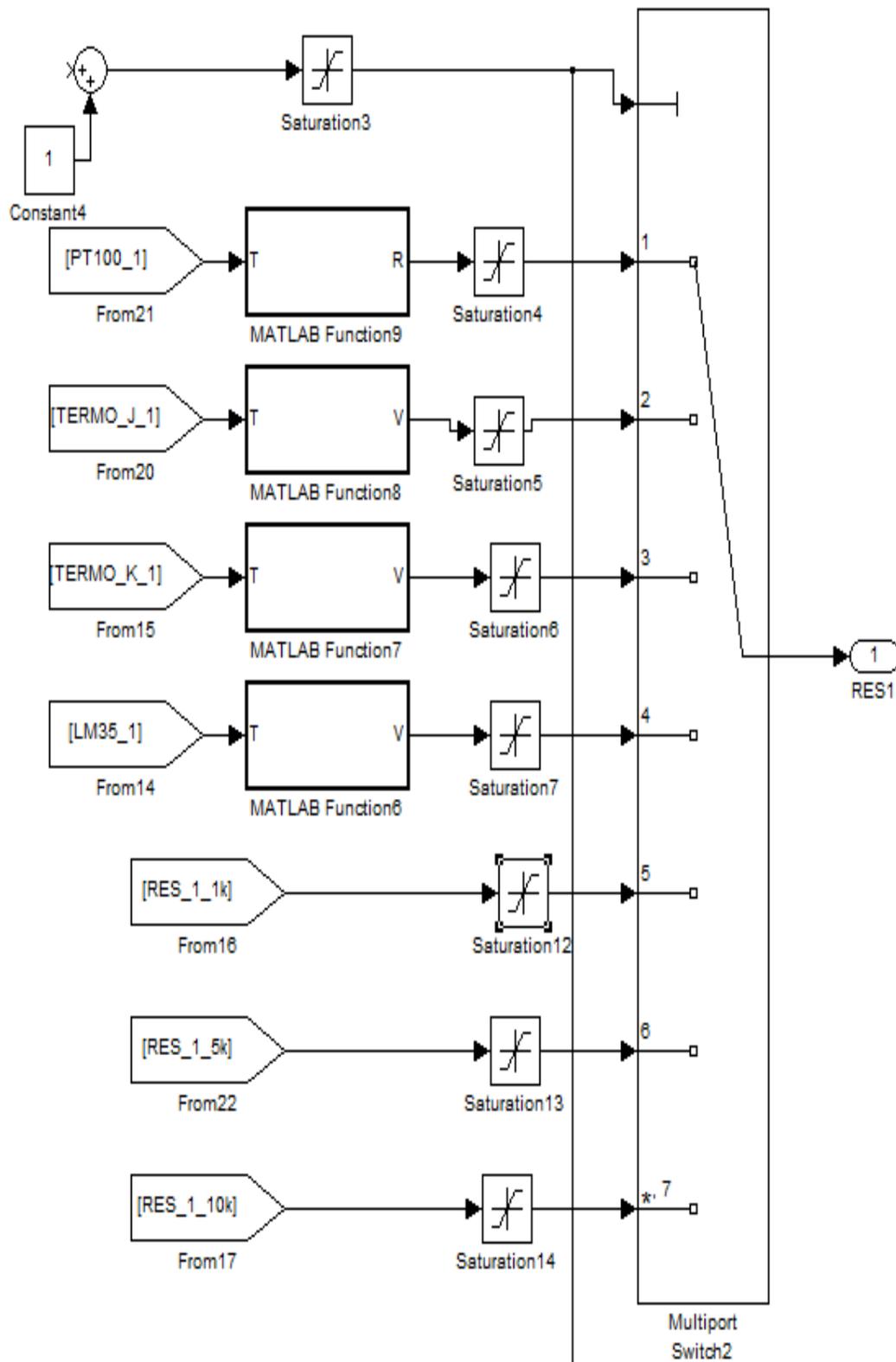


Figura 90 Datos de resistencia y voltaje sensores uno

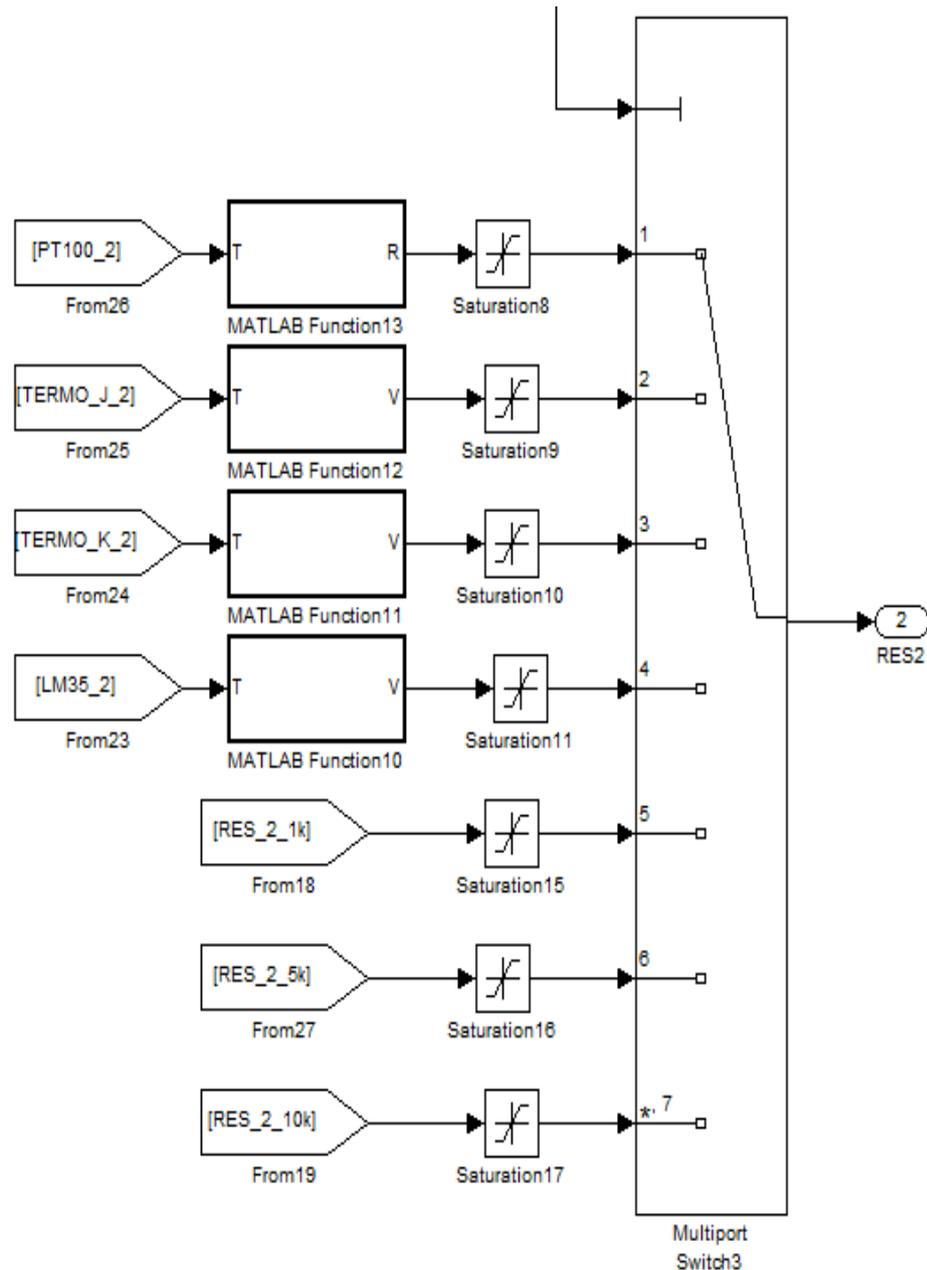


Figura 91 Datos de resistencia y voltaje sensores dos

La programación en **MATLAB FUNCTION** para el sensor uno, se visualiza en las Figura 92, Figura 93, Figura 94, Figura 95.

La programación es similar para el sensor dos, se observa en la Figura 92, Figura 93, Figura 94, Figura 95.

```
function R = temp_to_r(T)
R0 = 100;
a = 3.90830e-3;
b = -5.77500e-7;
R = R0*(1+a*T+b*T^2)*1000;
```

Figura 92 Temperatura a resistencia PT100

```
function V = temp_to_v(T)
a = -1.0306020e-7 ;
b = 1.9323799e-2 ;
c = -T;
V = (-b + sqrt(b^2-4*a*c))/(2*a);
```

Figura 93 Temperatura a voltaje termocupla J1

```
function V = temp_to_v(T)
a = -6.0883423e-8 ;
b = 2.5132785e-2;
c = -T;
V = (-b + sqrt(b^2-4*a*c))/(2*a);
```

Figura 94 Temperatura a voltaje termocupla k1

```
function V = temp_to_v(T)
V = T*10;
```

Figura 95 Temperatura a voltaje LM35

2.6 Diseño y programación del HMI.

En el diseño y programación del HMI se usa el software Android Studio para Touch Screen y Dispositivos Móviles; y lo referente a la interfaz de la PC fue diseñado y programado en LabVIEW, estos software son de fácil manejo y muy amigables con el usuario.

2.6.1 HMI touch screen por comunicación serial.

Para el HMI en el dispositivo Touch Screen se seleccionó Android Studio, con el cual realizamos la interfaz gráfica, la recepción de datos desde

la tarjeta de control y transmisión de datos hacia la tarjeta de control. Mientras que los datos recibidos en la Touch Screen son enviados desde la tarjeta STM32F407VG.

En la transmisión de datos desde la Tarjeta STM32F407VG hacia la Touch Screen se envía en primer lugar las temperaturas (TEMP_1 y TEMP_2), luego la temperatura de control (TEMP_C). Este proceso se realiza con todas las otras variables como son la selección de opciones de sensores (SELEC), unidades de medida de los sensores (RES_1 y RES_2), set point (SP), variable manipulada (MV), proceso que se indica en la Figura 96.

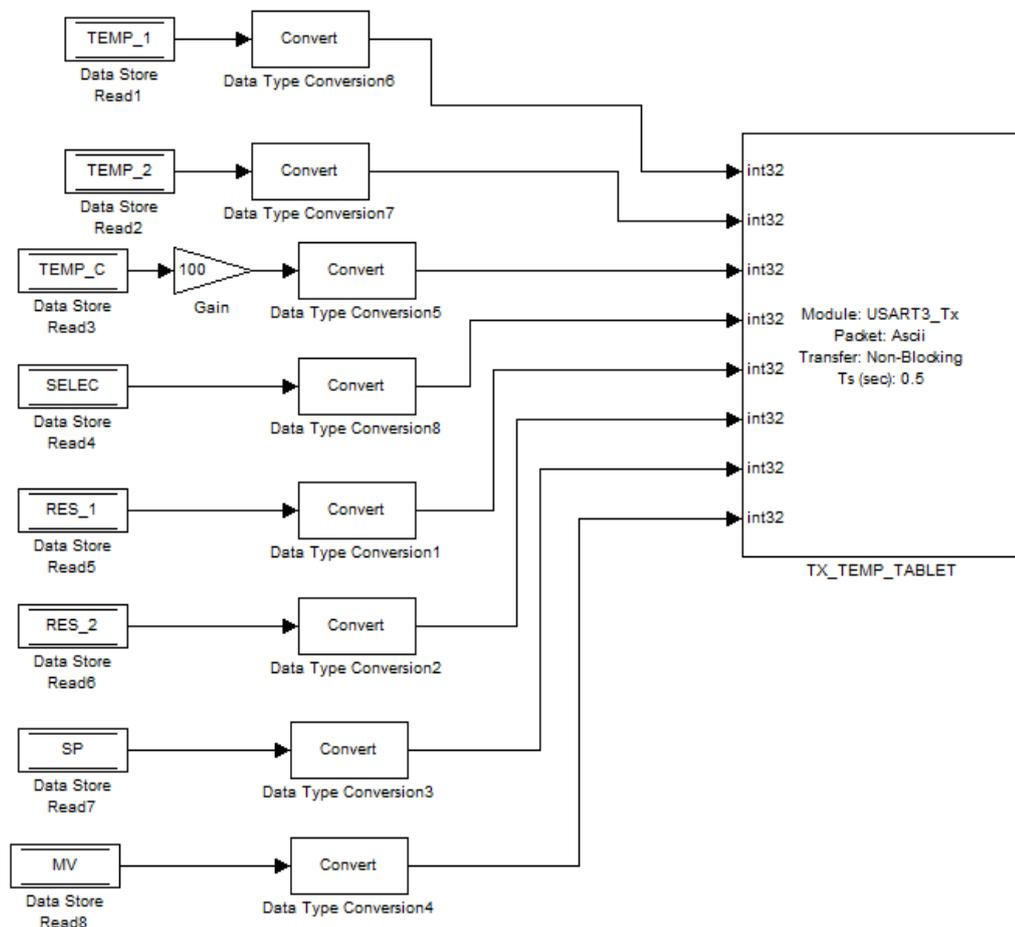


Figura 96 Transmisión STM32F407VG hacia touch screen

Para la transmisión de datos de la Touch Screen hacia la tarjeta se realiza el proceso que muestra en la Figura 97, aquí podemos observar los bloques de recepción datos para selección de sensores (SELEC) y el set point (SP).

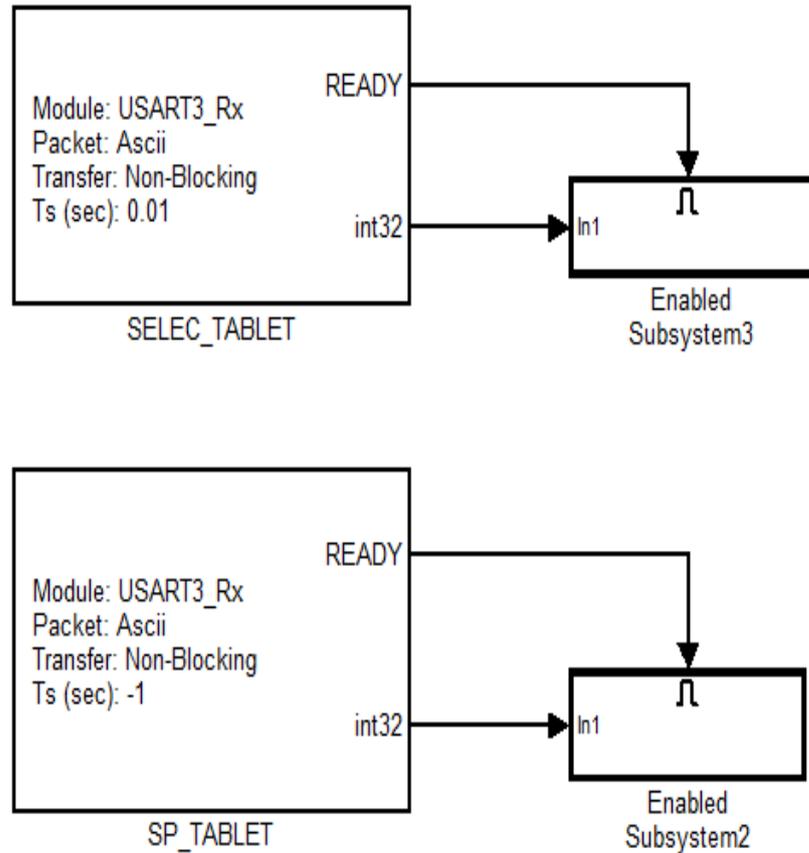


Figura 97 Recepción tarjeta desde touch screen

Mediante **Enabled Subsystem3**, el dato recibido de la Touch Screen es convertido a un valor entero para la selección del tipo de sensor (Figura 98). El mismo proceso se realiza para la variable del set point (SP) con **Enabled Subsystem2** (Figura 99).

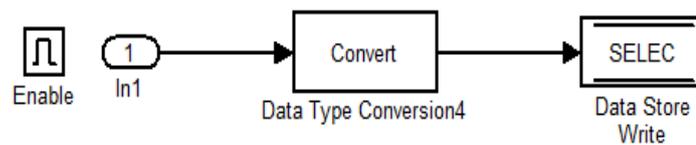


Figura 98 Recepción tipo de sensor

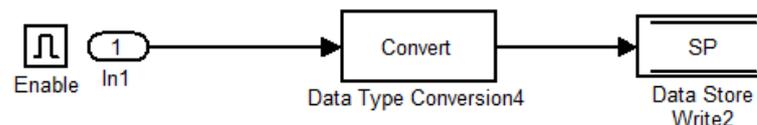


Figura 99 Recepción set point

Mediante la explicación a través de los diagramas de bloque indicados anteriormente se concluye la transmisión y recepción de datos desde y hacia la tarjeta.

Para el diseño del HMI, la recepción y transmisión en la Touch Screen con el software de Android Studio se procede como se explica a continuación.

Al abrir el programa se despliega una pantalla de bienvenida con varias opciones, como muestra la Figura 100.



Figura 100 Inicio android studio

Primeramente, seleccionamos **New Project** (Nuevo proyecto) de la lista de opciones. En la Figura 101 se observa una lista de opciones para configurar un nuevo proyecto.

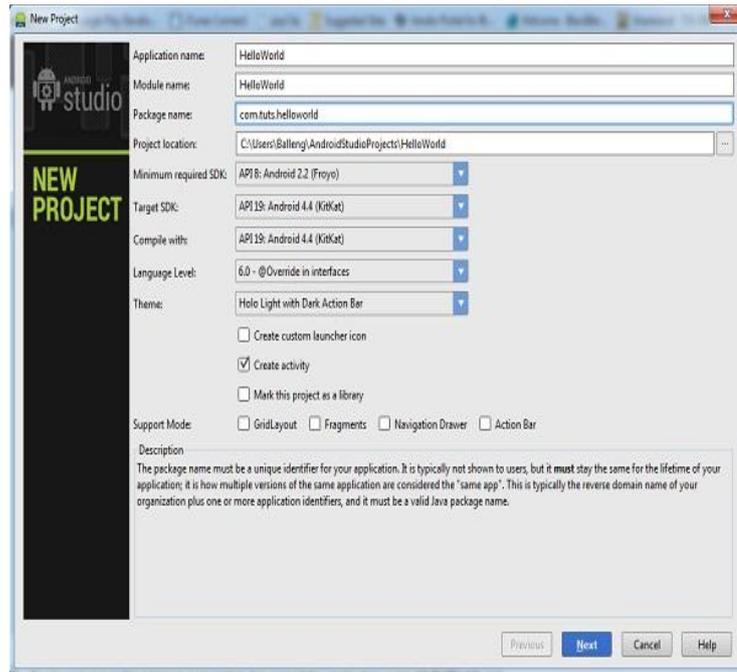


Figura 101 Configura un nuevo proyecto

Nombre de la aplicación: El nombre de la aplicación. Será mostrado por ejemplo en Google Play.

Nombre del módulo: Utilizado por Android Studio. Suele ser igual al nombre de la aplicación.

Nombre del paquete: Un identificador único de nuestra aplicación.

SDK mínimo requerido: Versión de Android mínima necesaria para correr la aplicación.

SDK objetivo: Máxima versión de Android con la que podrá ejecutarse la aplicación.

Nivel de lenguaje: Versión de Java con la que se desarrollará la aplicación.

Icono de la aplicación: El ícono que representará la aplicación.

Para la creación de nuestro HMI en Android Studio realizamos los siguientes pasos:

En la Figura 101, hemos establecido el nombre de la aplicación a **HelloWorld** y también el nombre del módulo a **HelloWorld**. Un módulo es una unidad de funcionalidad simple que puede ser compilada, ejecutada, testeada y depurada de forma independiente. Los módulos contienen el código fuente, los scripts de creación y cualquier parámetro adicional necesario para poder realizar su tarea específica.

Cuando creamos un nuevo proyecto también podemos establecer el nombre del paquete del proyecto. Por defecto, Android Studio establece el nombre del paquete del proyecto con el último nombre del módulo, pero podemos cambiarlo y establecer el nombre que queramos.

Otras configuraciones son la ubicación del proyecto en nuestro ordenador, la versión de SDK mínima y objetivo, la versión de SDK con la que será compilado el proyecto y el tema del proyecto. También podremos decirle a Android Studio que nos cree una clase **Activity** y un icono de inicio.

Para esta aplicación no vamos a crear un icono personalizado, por lo que desmarcamos la opción **Create custom launch icon**. Pulsamos **Next** para continuar con la configuración de nuestro proyecto.

Como dejamos seleccionada la opción **Create activity** en el paso anterior, se nos preguntará por la configuración de la clase **Activity** que Android Studio creará (Figura 102), pulsar Next.

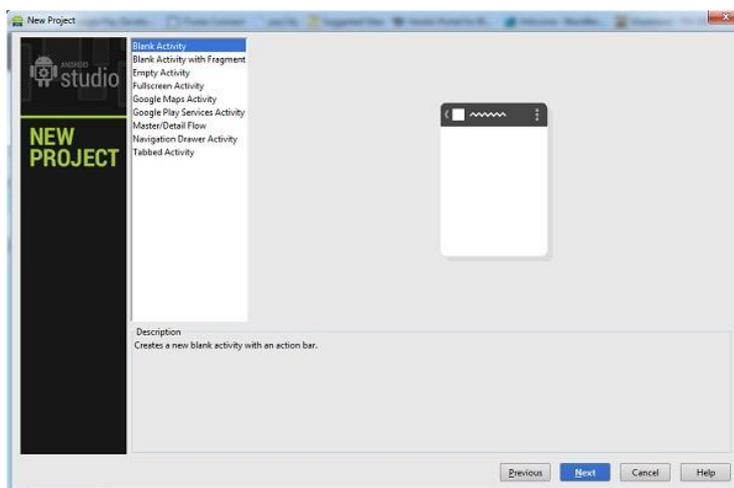


Figura 102 Clase de actividad

En la siguiente ventana que aparece se nos preguntará el nombre de la clase **Activity**, el layout principal, y el fragment layout. También podemos establecer el tipo de navegación, en este caso lo hemos dejado en **None**, y pulsamos Finish, Figura 103.

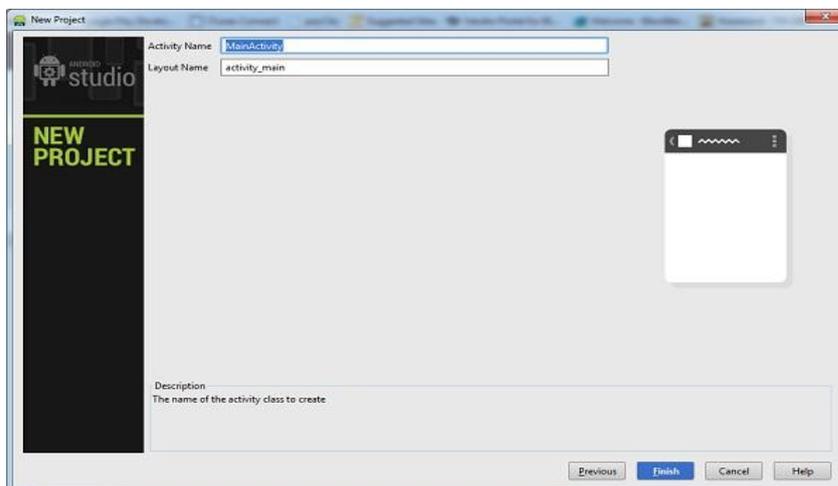


Figura 103 Configuración del proyecto

A continuación se despliega la interfaz de usuario de Android Studio con el explorador de proyectos a la izquierda y el editor de código a la derecha (Figura 104).

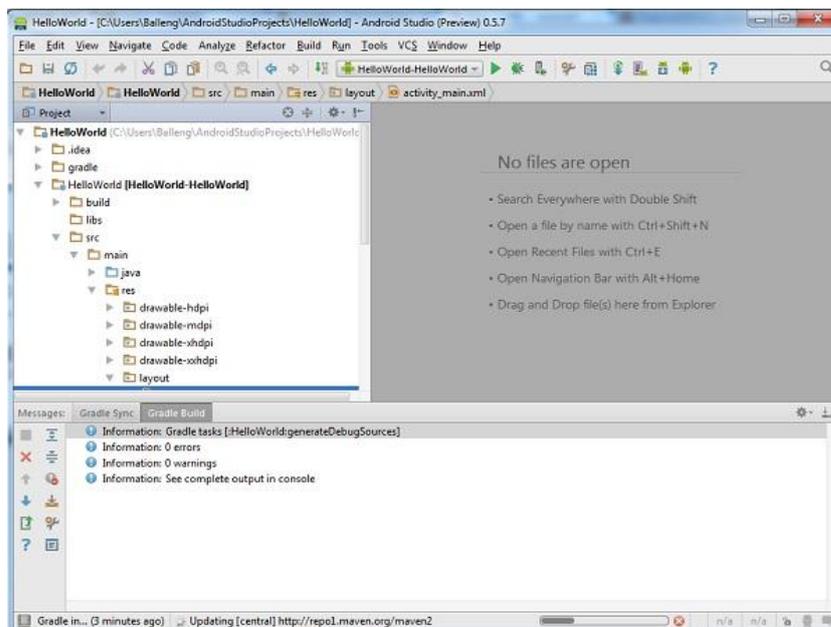


Figura 104 Interfaz usuario android studio

El Software Android Studio nos muestra dos pantallas, una de programación y otra de diseño de HMI.

Para la recepción de datos trabajamos en la pantalla de programación específicamente en **MainActivity**, en donde se crean por defecto todas las librerías y códigos para la comunicación USB Serial, debe tenerse en cuenta que la Touch Screen necesita estar conectada al puerto serial, con la finalidad de instalar la aplicación a la misma (Figura 105).

```
/** Simple container for a UsbDevice and its driver. */
private static class DeviceEntry {
    public UsbDevice device;
    public UsbSerialDriver driver;

    DeviceEntry(UsbDevice device, UsbSerialDriver driver) {
        this.device = device;
        this.driver = driver;
    }
}
```

Figura 105 Código para comunicación USB serial

En el **SerialConsoleActivity** se crea la clase en donde asignamos las variables con las cuales recibimos cada uno de los datos provenientes de la tarjeta de control (Figura 106).

```

package com.example.hmi_robotica;

import ...

public class SerialConsoleActivity extends Activity implements DataDisplay{

    private XYPlot plot,plot2;
    ArrayList<Double> Vector = new ArrayList<>();
    ArrayList<Double> Vector2 = new ArrayList<>();
    public double DatoX,temp1,temp2,tempc, res1, res2;
    int tiempo=50;
    int selec=0;
    int cc=0;
    int sensor=0;
    double sp=0;
    int inicio=1;
    //String sas="";

    private final String TAG = SerialConsoleActivity.class.getSimpleName();

    private static UsbSerialDriver sDriver = null;

    private TextView mTitleTextView;
    private TextView mTextView;

    MyServer server = new MyServer();
    TextView serverMessage, mTemp1, mTemp2, mTempc, mSelec, mRes1, mRes2, mUni1, mUni2;
    EditText mSp;
    Button bPt100, bTermoj, bTermok, bLm35, bTermistor1, bTermistor5, bTermistor10, bSp, bInicio, bDetener;
    //public String rxSerialData;

```

Figura 106 Asignación de variables

En la Figura 107 se indica el código creado para arrancar la aplicación en el instante que recibe los datos desde la tarjeta.

```

public static Integer[] ThumbsIds= {
    R.drawable.tt,
    R.drawable.alluc,
    R.drawable.jj,
    R.drawable.cc,
    R.drawable.dd,
};

private final ExecutorService mExecutor = Executors.newSingleThreadExecutor();

private SerialInputOutputManager mSerialIoManager;

private final SerialInputOutputManager.Listener mListener =
    new SerialInputOutputManager.Listener() {

        @Override
        public void onRunError(Exception e) { Log.d(TAG, "Runner stopped."); }

        @Override
        public void onNewData(final byte[] data) {
            SerialConsoleActivity.this.runOnUiThread(() -> {
                SerialConsoleActivity.this.updateReceivedData(data);
            });
        }
    };
};

```

Figura 107 Inicializar recepción de datos

Después instanciamos las variables, con esto nos referimos a relacionar las variables creadas en Android Studio con las variables globales que tenemos en la tarjeta de control (Figura 108).

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.serial_console);
    //serverMessage = (TextView) findViewById(R.id.textView3);
    mTemp1 = (TextView) findViewById(R.id.TEMP_1);
    mTemp2 = (TextView) findViewById(R.id.TEMP_2);
    mTempc = (TextView) findViewById(R.id.TEMP_C);
    mSelec = (TextView) findViewById(R.id.SELEC);
    mUni1 = (TextView) findViewById(R.id.textView3);
    mUni2 = (TextView) findViewById(R.id.textView4);
    mSp = (EditText) findViewById(R.id.SP);
    mRes1 = (TextView) findViewById(R.id.RES_1);
    mRes2 = (TextView) findViewById(R.id.RES_2);

    bPt100 = (Button) findViewById(R.id.PT100);
    bTermoj = (Button) findViewById(R.id.TERMO_J);
    bTermok = (Button) findViewById(R.id.TERMO_K);
    bLm35 = (Button) findViewById(R.id.LM35);
    bTermistor1 = (Button) findViewById(R.id.TER1);
    bTermistor5 = (Button) findViewById(R.id.TER5);
    bTermistor10 = (Button) findViewById(R.id.TER10);
    bSp = (Button) findViewById(R.id.BSP);
    bInicio = (Button) findViewById(R.id.INICIO);
    bDetener = (Button) findViewById(R.id.DETENER);
}

```

Figura 108 Instanciar variables globales

También instanciamos las variables para realizar la gráfica, en donde la instrucción **plot.setRangeValueFormat(new DecimalFormat("###.#"))** nos permite graficar tres enteros y un decimal, la relación de la gráfica es temperatura (Escala de 0 a 225 °C en pasos de 25 °C) respecto tiempo (segundos) (Figura 109).

```

plot = (XYPlot) findViewById(R.id.Grafica);
plot.getGraphWidget().setDomainValueFormat(new DecimalFormat("0"));
plot.setDomainStepMode(XYStepMode.INCREMENT_BY_VAL);
plot.setDomainStepValue(tiempo/10);
plot.setRangeStepMode(XYStepMode.INCREMENT_BY_VAL);
plot.setRangeStepValue(25);
plot.setRangeValueFormat(new DecimalFormat("###.#"));
plot.setRangeBoundaries(0, 225, BoundaryMode.FIXED);
plot.setDomainBoundaries(0, tiempo, BoundaryMode.FIXED);

plot2 = (XYPlot) findViewById(R.id.Grafica2);
plot2.getGraphWidget().setDomainValueFormat(new DecimalFormat("0"));
plot2.setDomainStepMode(XYStepMode.INCREMENT_BY_VAL);
plot2.setDomainStepValue(tiempo/10);
plot2.setRangeStepMode(XYStepMode.INCREMENT_BY_VAL);
plot2.setRangeStepValue(25);
plot2.setRangeValueFormat(new DecimalFormat("###.#"));
plot2.setRangeBoundaries(0, 225, BoundaryMode.FIXED);
plot2.setDomainBoundaries(0, tiempo, BoundaryMode.FIXED);

```

Figura 109 Instanciar gráficas

En la Figura 110 se muestra la opción de un Timer de 500 ms, que va a llamar a la función **Refrescar** para actualizar los datos de temperatura y graficarlos.

```

Timer timer = new Timer();
timer.scheduleAtFixedRate(() -> { Refrescar(); }, 0, 500);

```

Figura 110 Timer (500ms)

Con los códigos indicados de la Figura 111 se permite visualizar las dos gráficas en la Touch Screen.

```

XYSeries series = new SimpleXYSeries(Vector, SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "señal");
LineAndPointFormatter seriesFormat=new LineAndPointFormatter(Color.rgb(127, 255, 0),0x000000,0x000000,null);
plot.clear();
plot.addSeries(series, seriesFormat);

XYSeries series2 = new SimpleXYSeries(Vector2, SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "señal");
LineAndPointFormatter seriesFormat2=new LineAndPointFormatter(Color.rgb(127, 255, 0),0x000000,0x000000,null);
plot2.clear();
plot2.addSeries(series2, seriesFormat2);

```

Figura 111 Códigos para graficar

En el diseño del HMI relacionamos cada variable global y vectores con cada uno de los elementos gráficos que componen nuestra interfaz gráfica (Figura 112).

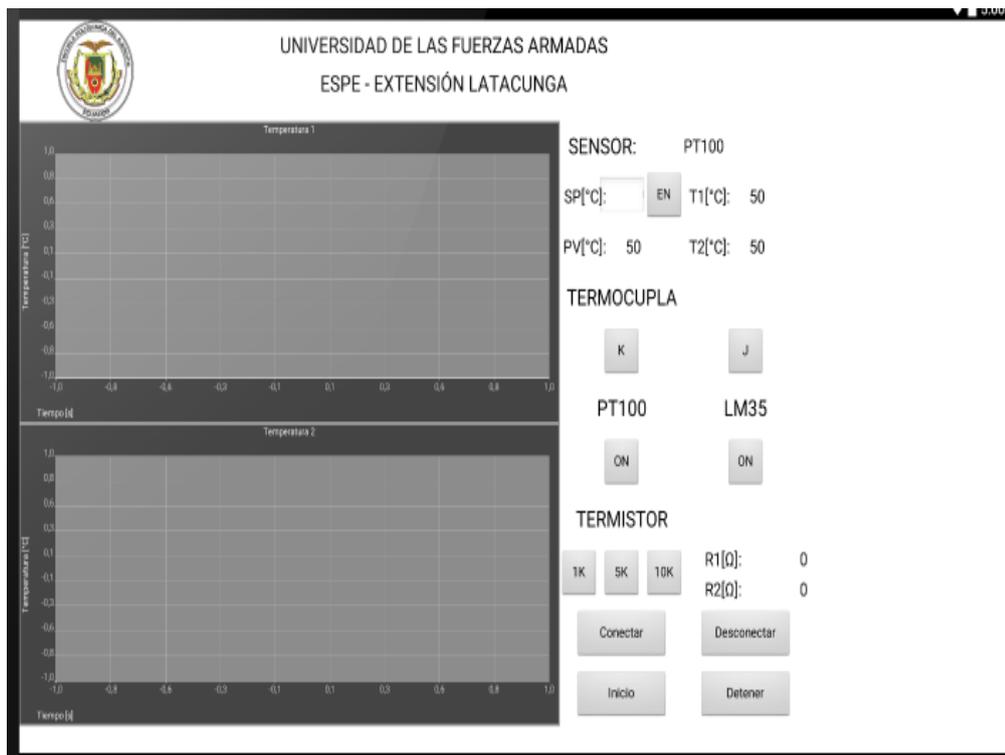


Figura 112 Interfaz gráfica

Diseñado el HMI programamos las dos gráficas, botones y cuadro de textos.

Como anteriormente instanciamos cada variable global con las variables creadas, lo que nos queda es relacionar cada botón de la gráfica con cada una de las variables creadas para enviar una instrucción a través de la comunicación serial. Esta instrucción es relacionada con el SELEC de la tarjeta de control, permitiendo así la elección del tipo de sensor a ser medido.

A continuación se explica las instrucciones para cada uno de los botones.

PT100



Para este botón se aplica las instrucciones de la Figura 113.

```
bPt100.setOnClickListener((view) -> {
    byte[] opt;
    String se="SELEC=0\n";
    opt = se.getBytes();
    try {
        sDriver.write(opt, 15);
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
});
```

Figura 113 Instrucciones botón PT100

TERMOCUPLA



Para los botones de termocuplas se aplican las siguientes instrucciones de la Figura 114.

```

bTermoj.setOnClickListener((view) -> {
    byte[] opt;
    String se="SELEC=1\n";
    opt = se.getBytes();
    try {
        sDriver.write(opt, 15);
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
});

bTermok.setOnClickListener((view) -> {
    byte[] opt;
    String se="SELEC=2\n";
    opt = se.getBytes();
    try {
        sDriver.write(opt, 15);
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
});

```

Figura 114 Instrucciones botones termocuplas

LM35



Para este botón se aplica el código de la Figura 115.

```

bLm35.setOnClickListener((view) -> {
    byte[] opt;
    String se="SELEC=3\n";
    opt = se.getBytes();
    try {
        sDriver.write(opt, 15);
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
});

```

Figura 115 Instrucciones botón LM35

TERMISTOR



Para los botones de termistores se aplican las siguientes instrucciones como se muestra en la Figura 116.

```

bTermistor1.setOnClickListener((view) -> {
    byte[] opt;
    String se="SELEC=4\n";
    opt = se.getBytes();
    try {
        sDriver.write(opt, 15);
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
});

bTermistor5.setOnClickListener((view) -> {
    byte[] opt;
    String se="SELEC=5\n";
    opt = se.getBytes();
    try {
        sDriver.write(opt, 15);
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
});

bTermistor10.setOnClickListener((view) -> {
    byte[] opt;
    String se="SELEC=6\n";
    opt = se.getBytes();
    try {
        sDriver.write(opt, 15);
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
});

```

Figura 116 Instrucciones botón termistores



Para esta selección se aplican las siguientes instrucciones que son enviadas a la tarjeta de control, su función es iniciar y detener el proceso en la tarjeta, como se muestra en la Figura 117.

```

bInicio.setOnClickListener((view) -> {
    byte[] hola;
    String sas="INI=1\n";
    inicio=1;
    hola = sas.getBytes();
    try {
        sDriver.write(hola, 15);
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
});

bDetener.setOnClickListener((view) -> {
    byte[] hola;
    String sas="INI=0\n";
    inicio=0;
    hola = sas.getBytes();
    try {
        sDriver.write(hola, 15);
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
});

```

Figura 117 Instrucciones botones inicio y detener



Para este botón lo primero que se realiza es tomar el dato del cuadro de texto y enviarlo a la tarjeta de control presionando el botón **EN**, el código fuente se muestra en la Figura 118.

```

bSp.setOnClickListener((view) -> {
    byte[] hola;
    String sas=mSp.getText().toString();
    sp=Double.parseDouble(sas);
    sas="SP="+sas+"\n";
    hola = sas.getBytes();
    try {
        sDriver.write(hola, 15);
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
});

```

Figura 118 Texto (SP) y botón EN

Las instrucciones indicadas en la Figura 119 es para formar todo el arreglo de vectores “x” y vectores “y” que se van a graficar en la Touch Screen, y con el código **LineAndPointFormatter seriesFormat** graficamos los vector en diferentes gráficas.

```

private void Refrescar() { this.runOnUiThread(Graf); }
private Runnable Graf = () -> {
    if(inicio==1) {
        if (DatoX < tiempo) {
            DatoX = DatoX + 0.5;
            Vector.add(DatoX);
            Vector.add(temp1);
            Vector2.add(DatoX);
            Vector2.add(temp2);
        } else {
            for (int i = 1; i < 2 * tiempo / 0.5 - 1; i = i + 2) {
                Vector.set(i, Vector.get(i + 2));
                Vector2.set(i, Vector2.get(i + 2));
            }
            Vector.set(2 * tiempo * 2 - 1, temp1);
            Vector2.set(2 * tiempo * 2 - 1, temp2);
        }
    }

    XYSeries series = new SimpleXYSeries(Vector, SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "senal");
    LineAndPointFormatter seriesFormat = new LineAndPointFormatter(Color.rgb(127, 255, 0), 0x000000, 0x000000, null);
    plot.clear();
    plot.addSeries(series, seriesFormat);
    plot.redraw();

    XYSeries series2 = new SimpleXYSeries(Vector2, SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "senal");
    LineAndPointFormatter seriesFormat2 = new LineAndPointFormatter(Color.rgb(127, 255, 0), 0x000000, 0x000000, null);
    plot2.clear();
    plot2.addSeries(series2, seriesFormat2);
    plot2.redraw();
};

```

Figura 119 Códigos arreglo de vectores para graficar



Con los botones de Conectar y Desconectar, activamos y desactivamos el servidor (Figura 120).

```
public void connect(View view) {  
    //if(cc==0) {  
        //    server.setEventListener(this);  
        server.startListening();  
        //    cc=1;  
    //    }  
}  
  
public void disconnect(View view) {  
  
    server.stopServer();  
    //cc=0;  
}
```

Figura 120 Activar/desactivar servidor

El **onPause** es para cerrar el Driver del Puerto Serial, cuando se pausa la aplicación, teniendo en cuenta que es una función propia del Android Studio.

onResume que se despliega al abrir la aplicación, es abrir al Puerto Serial en donde se ingresa la velocidad de Transmisión que debe tener la Tarjeta de control (Figura 121).

```

@Override
protected void onPause() {
    super.onPause();
    stopIoManager();
    if (sDriver != null) {
        try {
            sDriver.close();
        } catch (IOException e) {
            // Ignore.
        }
        sDriver = null;
    }
    finish();
}

@Override
protected void onResume() {
    super.onResume();
    Log.d(TAG, "Resumed, sDriver=" + sDriver);
    if (sDriver == null) {
        //mTitleTextView.setText("No serial device.");
    } else {
        try {
            sDriver.open();
            sDriver.setParameters(115200, 8, UsbSerialDriver.STOPBITS_1, UsbSerialDriver.PARITY_NONE);
        } catch (IOException e) {
            Log.e(TAG, "Error setting up device: " + e.getMessage(), e);
            //mTitleTextView.setText("Error opening device: " + e.getMessage());
            try {
                sDriver.close();
            } catch (IOException e2) {
                // Ignore.
            }
            sDriver = null;
            return;
        }
        mTitleTextView.setText("Serial device: " + sDriver.getClass().getSimpleName());
    }
    onDeviceStateChange();
}

```

Figura 121 Abrir/cerrar el puerto serial

Debemos tener en cuenta que el **stopIoManager** y **startIoManager** son funciones que necesita el Puerto Serial y están designadas por defecto como muestra la Figura 122.

```

private void stopIoManager() {
    if (mSerialIoManager != null) {
        Log.i(TAG, "Stopping io manager ..");
        mSerialIoManager.stop();
        mSerialIoManager = null;
    }
}

private void startIoManager() {
    if (sDriver != null) {
        Log.i(TAG, "Starting io manager ..");
        mSerialIoManager = new SerialInputOutputManager(sDriver, mListener);
        mExecutor.submit(mSerialIoManager);
    }
}

private void onDeviceStateChange() {
    stopIoManager();
    startIoManager();
}

```

Figura 122 Funciones puerto serial

El **updateReceivedData** es para recibir los datos de las variables globales en la Touch Screen. Para codificar estos datos debemos realizar bit a bit y obtener el valor de la temperatura en centena, decena, unidad, decimas, centésimas, luego sumar estos datos y obtener la temperatura uno. El **bigStr** es para ver si los datos son positivos o negativos.

Para la **res1** y **res2** estos ya no tienen decimales, por lo tanto hay que multiplicarlos por 100000 y a la salida nos da en μV , mV o en $\text{m}\Omega$ (Figura 123).

```

private void updateReceivedData(byte[] data) {

    server.rxData(new String(data),1);

    String serverTx = server.txData();
    //if (serverTx.length()>4){
        byte[] hola;
        hola = serverTx.getBytes();
        try {
            sDriver.write(hola, 15);
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}

temp1=(data[1]-48)*(((data[1]-48)>=0)?1:0)*100+(data[2]-48)*(((data[2]-48)>0)?1:0)*10+(data[3]-48)*
(((data[3]-48)>0)?1:0)+(data[4]-48)*(((data[4]-48)>0)?1:0)*0.1+(data[5]-48)*(((data[5]-48)>0)?1:0)*0.01;
String bigStr = new String(data);
if (bigStr.indexOf("-")>1 & bigStr.indexOf("-")<=5)
    temp1=-1*temp1;

temp2=(data[7]-48)*(((data[7]-48)>0)?1:0)*100+(data[8]-48)*(((data[8]-48)>0)?1:0)*10+(data[9]-48)*
(((data[9]-48)>0)?1:0)+(data[10]-48)*(((data[10]-48)>0)?1:0)*0.1+(data[11]-48)*(((data[11]-48)>0)?1:0)*0.01;
if (bigStr.indexOf("-")>7 & bigStr.indexOf("-")<=11)
    temp2=-1*temp2;

tempc=(data[13]-48)*(((data[13]-48)>0)?1:0)*100+(data[14]-48)*(((data[14]-48)>0)?1:0)*10+(data[15]-48)*
(((data[15]-48)>0)?1:0)+(data[16]-48)*(((data[16]-48)>0)?1:0)*0.1+(data[17]-48)*(((data[17]-48)>0)?1:0)*0.01;
if (bigStr.indexOf("-")>13 & bigStr.indexOf("-")<=17)
    tempc=-1*tempc;

selec=(data[19]-48);

res1=(data[21]-48)*(((data[21]-48)>0)?1:0)*100000+(data[22]-48)*(((data[22]-48)>0)?1:0)*10000+(data[23]-48)*(((data[23]-48)>0)?1:0)*
1000+(data[24]-48)*(((data[24]-48)>0)?1:0)*100+(data[25]-48)*(((data[25]-48)>0)?1:0)*10+(data[26]-48)*(((data[26]-48)>0)?1:0);

res2=(data[28]-48)*(((data[28]-48)>0)?1:0)*100000+(data[29]-48)*(((data[29]-48)>0)?1:0)*10000+(data[30]-48)*(((data[30]-48)>0)?1:0)*
1000+(data[31]-48)*(((data[31]-48)>0)?1:0)*100+(data[32]-48)*(((data[32]-48)>0)?1:0)*10+(data[33]-48)*(((data[33]-48)>0)?1:0);

```

Figura 123 Recepción datos de variables globales touch screen

Para los textos de la pantalla del HMI se utiliza un **Switch** de selección:

El **CASE en 0** tenemos PT100 y la unidad en mΩ.

El **CASE en 1** Termocupla J la unidad en μV.

El **CASE en 2** Termocupla K la unidad en μV.

El **CASE en 3** LM35 la unidad en mV.

El **CASE en 4** Termistor de 1KΩ la unidad en Ω.

El **CASE en 5** Termistor de 5K Ω la unidad en Ω .

El **CASE en 6** Termistor de 10K Ω la unidad en Ω .

Por último procedemos a imprimir los datos (Figura 124).

```

switch (selec){
  case 0:
    mSelec.setText("PT100");
    mUni1.setText("R1 [m $\Omega$ ");
    mUni2.setText("R2 [m $\Omega$ ");
    break;
  case 1:
    mSelec.setText("TERMOCUPLA J");
    mUni1.setText("V1 [uV]");
    mUni2.setText("V2 [uV]");
    break;
  case 2:
    mSelec.setText("TERMOCUPLA K");
    mUni1.setText("V1 [uV]");
    mUni2.setText("V2 [uV]");
    break;
  case 3:
    mSelec.setText("LM35");
    mUni1.setText("V1 [mV]");
    mUni2.setText("V2 [mV]");
    break;
  case 4:
    mSelec.setText("TERMISTOR 1k");
    mUni1.setText("R1 [ $\Omega$ ");
    mUni2.setText("R2 [ $\Omega$ ");
    break;
  case 5:
    mSelec.setText("TERMISTOR 5k");
    mUni1.setText("R1 [ $\Omega$ ");
    mUni2.setText("R2 [ $\Omega$ ");
    break;
  case 6:
    mSelec.setText("TERMISTOR 10k");
    mUni1.setText("R1 [ $\Omega$ ");
    mUni2.setText("R2 [ $\Omega$ ");
    break;
  default:
    break;
}
mTemp1.setText(String.format("%.1f", temp1));
mTemp2.setText(String.format("%.1f", temp2));
mTempc.setText(String.format("%.1f", tempc));
mRes1.setText(String.format("%.0f", res1));
mRes2.setText(String.format("%.0f", res2));

```

Figura 124 Switch selección sensores

2.6.2 HMI dispositivo móvil por red wifi.

Para el HMI de los dispositivos móviles el programa se basa en la programación del HMI de la Touch Screen, teniendo como diferencia que el dispositivo móvil solo recibirá datos.

La transmisión de datos desde la tarjeta STM32F407VG hacia el dispositivo móvil sigue el siguiente proceso.

En la Figura 125, con la acción **rxSerialData** lo que hace es recibir los datos la tarjeta por el puerto serial y transmitir estos mismos al dispositivo móvil, entonces éste servidor realiza la misma decodificación de datos como se mostró en la Figura 123.

```
public class MyServer {
    Thread m_objThread;
    ServerSocket m_server;
    String m_strMessage;
    String rxSerialData;
    int eAct=0;
    DataDisplay m_dataDisplay;
    Object m_connected;
    int cont=0;
    int stop=0;
    String strMessage="";

    public MyServer () {
    }

    public void rxData(String data, int env) {
        rxSerialData=data;
        eAct = env;
    }
}
```

Figura 125 Tx STM32F407VG a dispositivo móvil

En la acción del **stopServer**, la variable **stop** se pone en 1 para detener el servidor, teniendo en cuenta que un servidor siempre está asociado a un lazo, el mismo que es un proceso aparte del principal. En la Figura 126 muestra cómo declarar el puerto, Tablet puerto 2001, Arduino puerto 23.

```

public void stopServer() { stop=1; }

public void startListening(){
    m_objThread=new Thread((Runnable) () -> {
        // TODO Auto-generated method stub
        try {
            m_server=new ServerSocket(2001);
            Socket connectedSocket=m_server.accept();
            Message clientmessage;
            ObjectInputStream ois = new ObjectInputStream(connectedSocket.getInputStream());
            ObjectOutputStream oos = new ObjectOutputStream(connectedSocket.getOutputStream())

            clientmessage = Message.obtain();
            strMessage = (String)ois.readObject();
            clientmessage.obj = strMessage;
            //mHandler.sendMessage(clientmessage);

            while(1==1) {
                if (eAct==1){
                    /*
                    clientmessage = Message.obtain();
                    strMessage = (String)ois.readObject();
                    clientmessage.obj = strMessage;
                    mHandler.sendMessage(clientmessage);
                    //Thread.sleep(500);

                    */
                    eAct=0;
                    if(stop==1){
                        oos.writeObject("chao");
                        break;
                    }
                    oos.writeObject(rxSerialData);
                }
            }

            stop=0;
            ois.close();
            oos.close();
            m_server.close();
            //m_objThread.stop();
        }
    }
}

```

Figura 126 Transmisión datos wifi

A continuación explicaremos la programación de recepción de datos desde la Touch Screen hacia el dispositivo móvil.

Los siguientes parámetros que se describen tienen la misma función en HMI Touch Screen y HMI del Dispositivo Móvil, analizado anteriormente.

1. Asignación de Variables. (Figura 106)
2. Inicializar recepción de datos. (Figura 107)
3. Instanciar variables globales. (Figura 108)
4. Instanciar Graficar. (Figura 109)
5. Timer (500 ms). (Figura 110)
6. Código para gráfico. (Figura 111)
7. Interfaz Gráfica. (Figura 112)
8. Instrucción botón PT100. (Figura 113)
9. Instrucciones botones Termocuplas. (Figura 114)
10. Instrucción botón LM35. (Figura 115)
11. Instrucciones botones Termistores. (Figura 116)
12. Instrucciones botones Inicio - Detener. (Figura 117)
13. Instrucciones cuadro de texto (SP) y botón EN. (Figura 118)
14. Código arreglo de vectores para Graficar. (Figura 119)
15. Switch Selección Sensores. (Figura 124)

La diferencia en los comando del HMI Touch Screen con HMI Dispositivos Móvil se muestra a continuación en las Figura 127, Figura 128.

Al presionar el **botón conectar** funciona el comando **Thread** para conectarse con el servidor, en este caso con el Dispositivo Móvil con IP **192.168.0.103** y un puerto 2001, el **ObjectOutputStream oos**, **ObjectInputStream ois** es el flujo de entrada y flujo salida para envío y recepción de datos.

```

public void Start(View view){
    m_objThreadClient=new Thread(new Runnable() {
        @Override
        public void run() {
            // TODO Auto-generated method stub
            try {
                clientSocket = new Socket("192.168.0.103",2001);
                ObjectOutputStream oos = new ObjectOutputStream(clientSocket.getOutputStream());
                ObjectInputStream ois = new ObjectInputStream(clientSocket.getInputStream());
                Message serverMessage ;
                String strMessage;
                int i=0;

                oos.writeObject("c"+i);
            }
        }
    });
}

```

Figura 127 Habilitar servidor

En la Figura 128 el **while1** contiene el código principal, la acción **serverMessage** es obtener datos por Wifi y **readObject** es leer datos y guardarlos en el **strMessage**, estos datos le pasan a un **Array** de byte que se guarda en la variable **data** obteniendo la misma configuración de datos para las variables globales recibidas.

```

while(1==1) {

    /*
    Thread.sleep(480);
    oos.writeObject("c"+i);
    */

    serverMessage = Message.obtain();
    strMessage = (String)ois.readObject();
    serverMessage.obj = strMessage;
    mHandler.sendMessage(serverMessage);

    data = strMessage.toCharArray();
}

```

Figura 128 Función while

2.6.3 HMI PC por red ethernet.

Para este HMI se escogió LabVIEW, teniendo este la característica de solo recibir datos desde la tarjeta de control.

En la transmisión de datos desde la Tarjeta STM32F407VG hacia la Touch Screen se realiza el siguiente proceso:

En primer lugar se envían las temperaturas (TEMP_1 y TEMP_2) hacia la Touch Screen, luego la temperatura de control (TEMP_C) y de manera similar con cada una de las variables globales explicadas anteriormente como: la selección de opciones de sensores (SELEC), unidades de medida de los sensores (RES_1 y RES_2), set point (SP), variable manipulada (MV), todas estas variables se muestran en la Figura 129.

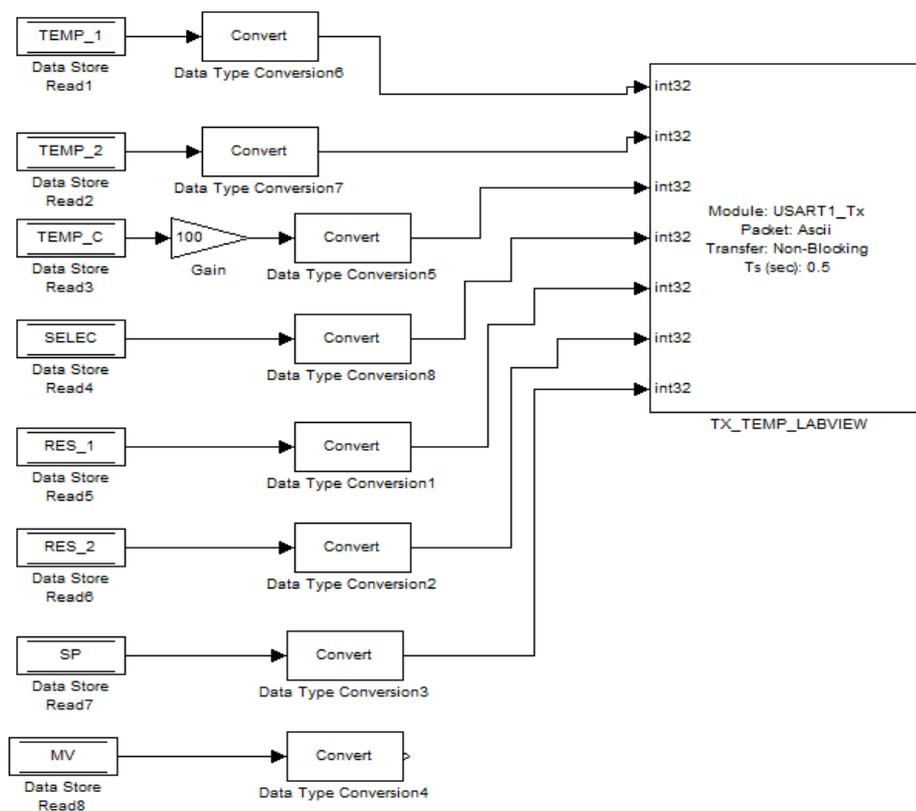


Figura 129 Transmisión STM32F407VG hacia LabVIEW

El arduino adquiere los datos del puerto serial de la Tarjeta STM32F407VG y transmite esos mismos a través de Ethernet a LabVIEW, esto se desarrolla inicializando la variable **SPI Ethernet**, declarando la **MAC** de Ethernet, envía las variables (**sendDataS**, **sendDataSS**, **sendDataE**, **sendDataEE**), como se indica en la. Figura 130.

```

#include <SPI.h>
#include <Ethernet.h>

// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network.
// gateway and subnet are optional:
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
//IPAddress ip(192,168,0, 27);
//IPAddress gateway(192,168,0, 1);
//IPAddress subnet(255, 255, 255, 0);
String sendDataS=""; // Allocate some space for the string
String sendDataSS=""; // Allocate some space for the string
char inCharS; // Where to store the character read
String sendDataE=""; // Allocate some space for the string
String sendDataEE=""; // Allocate some space for the string
char inCharE; // Where to store the character read
int borrar=0;
// telnet defaults to port 23

```

Figura 130 Inicialización SPI ethernet

En la figura Figura 131, el **EthernetServer**, lo inicializamos en el puerto 23, este mismo puerto debe estar en LabVIEW y Arduino, declaramos la velocidad de transmisión del serial de **19200 baudios**, declaramos **30** localidades como espacios de memorias para el envío de datos.

```

EthernetServer server(23);
boolean alreadyConnected = false; // whether or not the client was

void setup() {
  // initialize the ethernet device
  //Ethernet.begin(mac, ip, gateway, subnet);
  Ethernet.begin(mac);
  // start listening for clients
  server.begin();
  // Open serial communications and wait for port to open:
  Serial.begin(19200);
  sendDataSS.reserve(30);
  sendDataS.reserve(30);
  sendDataEE.reserve(30);
  sendDataE.reserve(30);
}

```

Figura 131 Servidor ethernet

EthernetClient client = server.available, con esta acción preguntamos si está habilitado el servidor, activar el cliente serial, preguntamos si han llegado datos del serial, entonces realizamos un barrido de los datos que llegan hasta que sea igual a 10 y esto almacenamos en la variable **sendDataSS** por último enviamos dicha variable a través de Ethernet a LabVIEW (Figura 132).

```

void loop() {
  // wait for a new client:
  EthernetClient client = server.available();
  if (borrar==0){
    // Serial.flush();
  }

  if (client) {
    if(Serial.available()>0){
      borrar=1;
      inCharS = Serial.read(); // Read a character
      while(inCharS!=10){
        sendDataS+=(char) (inCharS);
        inCharS = Serial.read();
      }
      sendDataSS=sendDataS;
      sendDataS="";
    }

    //sendDataSS="Hola"+String(borrar,DEC);
    if(borrar=1){
      client.println(sendDataSS);
      delay(500);
    }
    borrar=0;
  }
}

```

Figura 132 Cliente ethernet

La programación de Recepción de datos desde la tarjeta STM32F407VG hacia la PC, realizado con el Software LabVIEW es como sigue.

Abrir el software LabVIEW como se indica en la Figura 133.

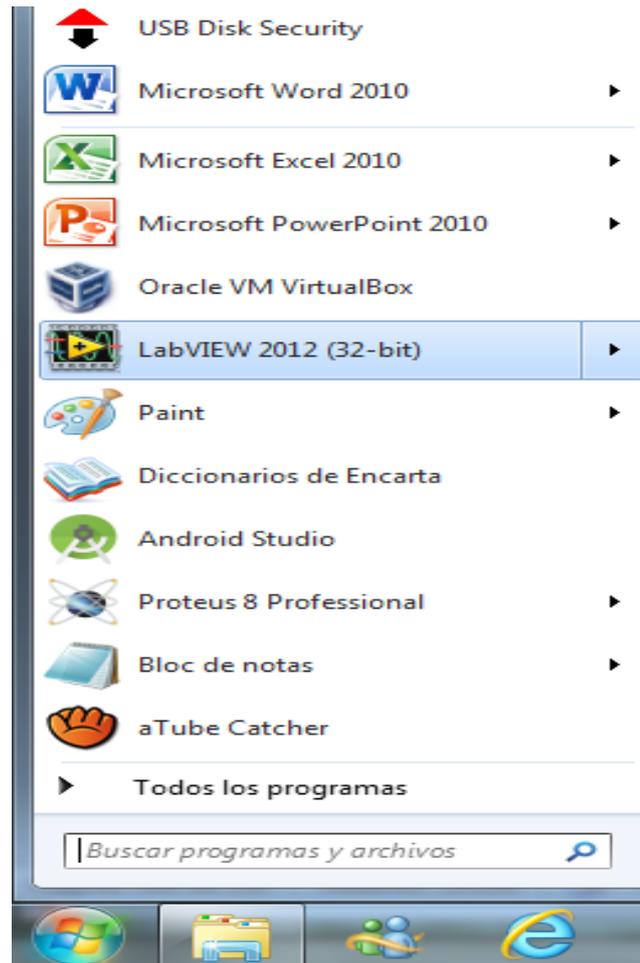


Figura 133 Ingreso software LabVIEW

Dar clic en crear un nuevo proyecto como se indica en la Figura 134, se despliega una nueva ventana ahí elegir proyecto en blanco como indica en la Figura 135 y dar clic en finalizar.



Figura 134 Crear un nuevo proyecto en LabVIEW

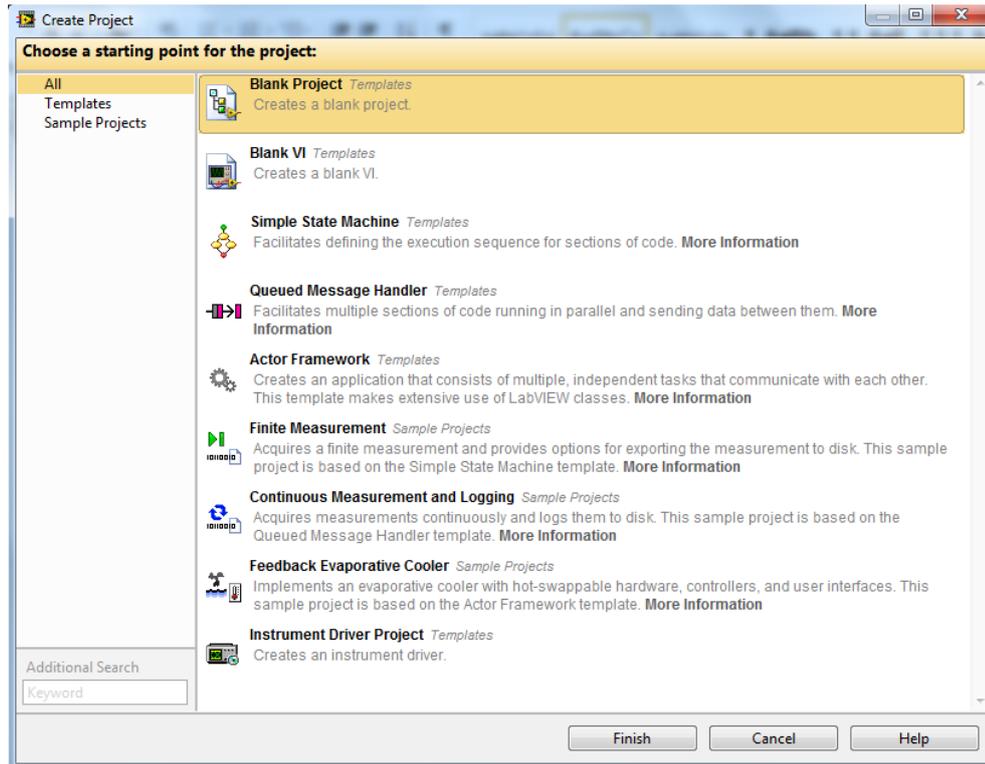


Figura 135 Crear un proyecto en blanco en LabVIEW

Se abre una ventana en donde se encuentra el proyecto creado como se indica en la Figura 136.

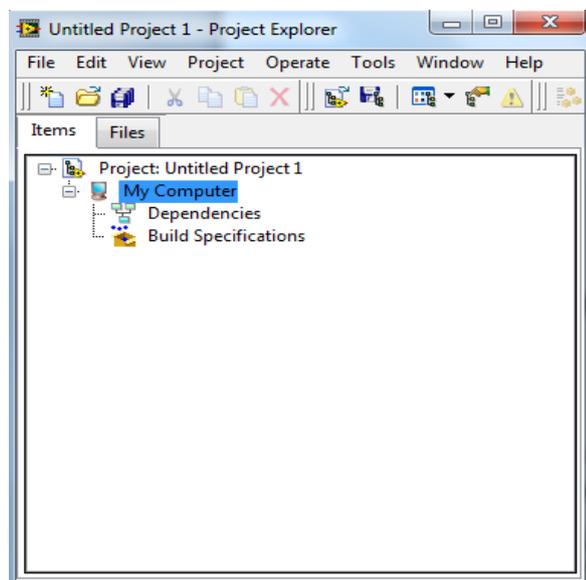


Figura 136 Ventana del proyecto

Dar clic derecho en **My Computer/New/VI** y se abren las pantallas de diagramas de bloques y en el panel frontal de LabVIEW, como indica en la Figura 137.

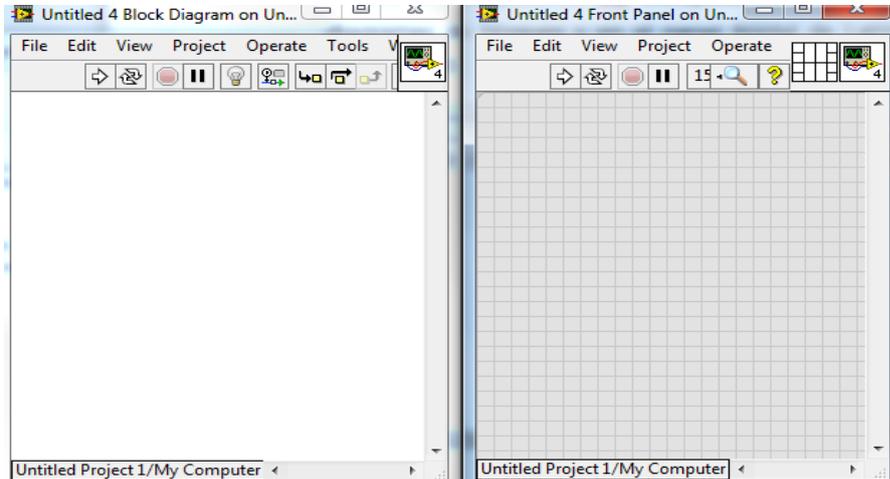
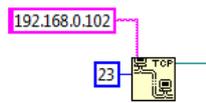
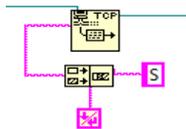


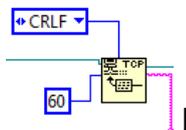
Figura 137 Creación del VI



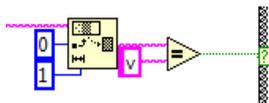
Este bloque es para la inicialización del servidor, tenemos la dirección IP 192.168.0.102 del servidor del Arduino y el puerto 23.



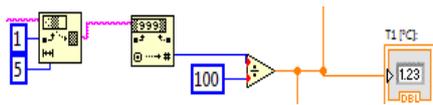
Bloque de envío de datos pero sabes que LabVIEW solo va a recibir los datos.



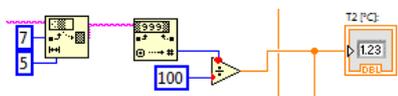
Bloque de recepción de datos, 60 es el número de bytes que vamos a recibir, dato más CRLF que es el byte de terminación.



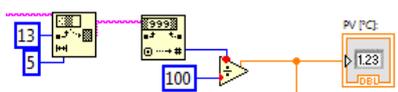
En este bloque preguntamos si el primer dato que recibimos es igual a **V**, después discriminamos a los siguientes bloques.



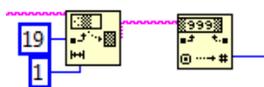
Este bloque describe que desde el byte uno con dimensión de cinco bytes es la T1 °C.



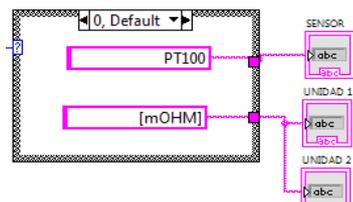
Este bloque describe que desde el byte siete con dimensión de cinco bytes es la T2 °C.



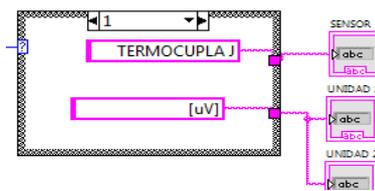
Este bloque describe que desde el byte trece con dimensión de cinco bytes es la temperatura del PV °C.



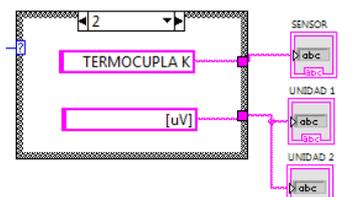
Este bloque describe que desde el byte diecinueve con dimensión de un byte es el SELEC de 0 a 6.



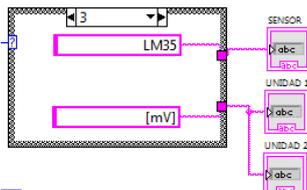
Bloque de SELEC, cuando está en 0 el sensor es PT100 y la unidad [mΩ].



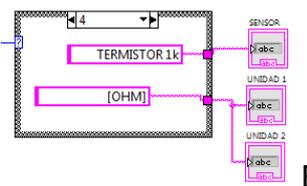
Bloque de SELEC, cuando está en 1 el sensor es Termocupla J y la unidad [μV].



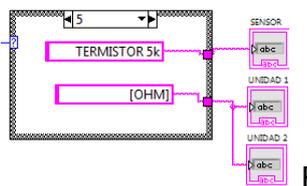
Bloque de SELEC, cuando está en 2 el sensor es Termocupla K y la unidad [μV].



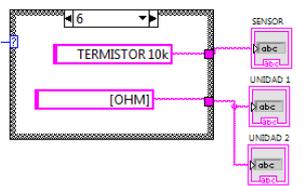
— Bloque de SELEC, cuando está en 3 el sensor es LM35 y la unidad [mV].



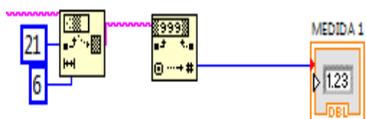
Bloque de SELEC, cuando está en 4 el sensor es Termistor de 1KΩ y la unidad [Ω].



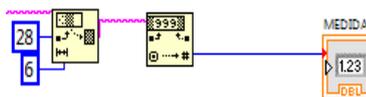
Bloque de SELEC, cuando está en 5 el sensor es Termistor de 5KΩ y la unidad [Ω].



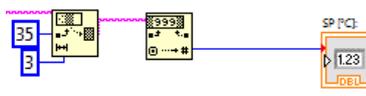
Bloque de SELEC, cuando está en 6 el sensor es Termistor de 10KΩ y la unidad [Ω].



Este bloque describe que desde el byte veintiuno con dimensión de seis bytes la medida uno.



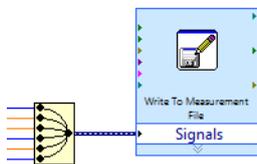
Este bloque describe que desde el byte veintiocho con dimensión de seis bytes la medida dos.



Este bloque describe que desde el byte treintainco con dimensión de tres bytes el SP °C.



Bloque para realizar las gráficas.



Bloque1 que une todas las señales y las guardan en el bloque 2 en **Write To Measurement File**.

Dando clic en el bloque de **Write To Measurement File**, ingresamos la dirección en donde se guardan los Datos como se muestra la Figura 138.

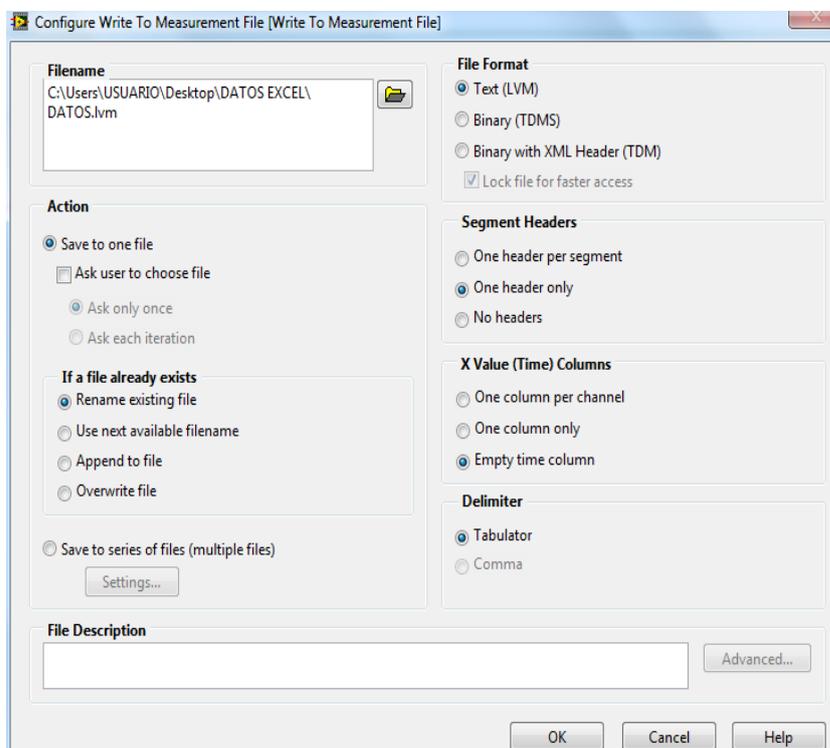


Figura 138 Dirección de los datos guardados

En el diagrama de bloques de la Figura 139 se detalla el reporte de los datos de cada uno de los sensores de temperatura, enviados desde LabVIEW a Excel, creando una tabla de datos en donde se muestra la temperatura del sensor 1 y el sensor 2, de la misma manera muestra la unidad de medida de cada sensor, también genera una gráfica BMP en donde ilustra la curva característica del sensor de temperatura.

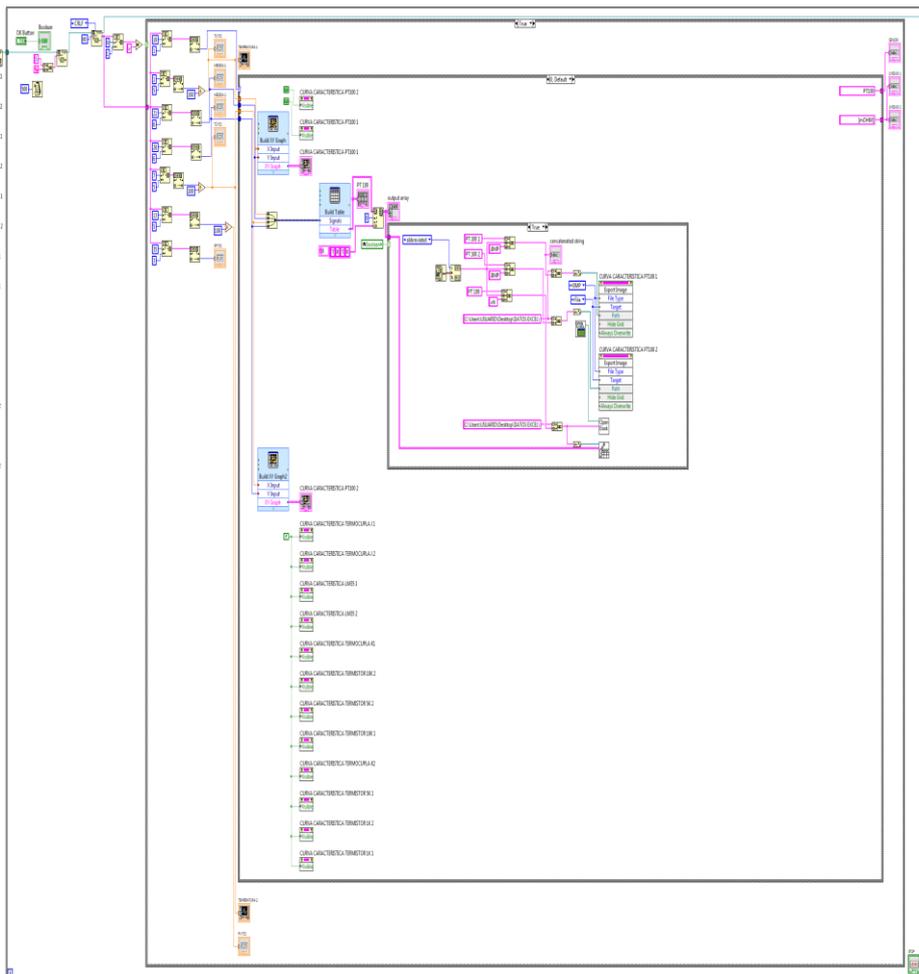


Figura 140 Diagrama de bloques HMI PC por red ethernet

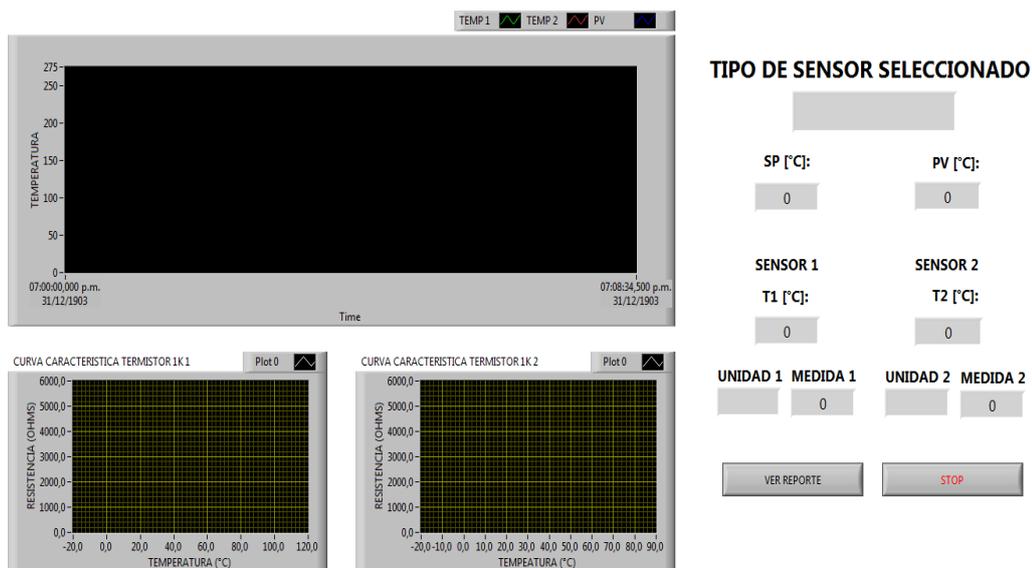


Figura 141 HMI PC por red ethernet

2.7 Diagrama de bloques del sistema de control.

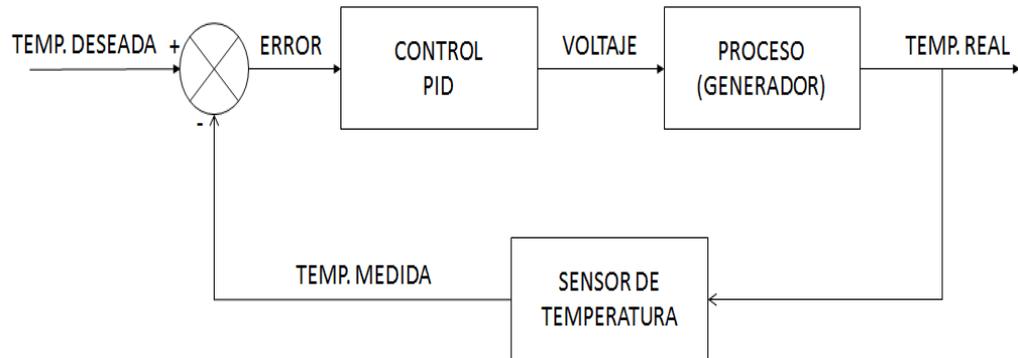


Figura 142 Diagrama de bloques

2.8 Diagrama P&ID del sistema.

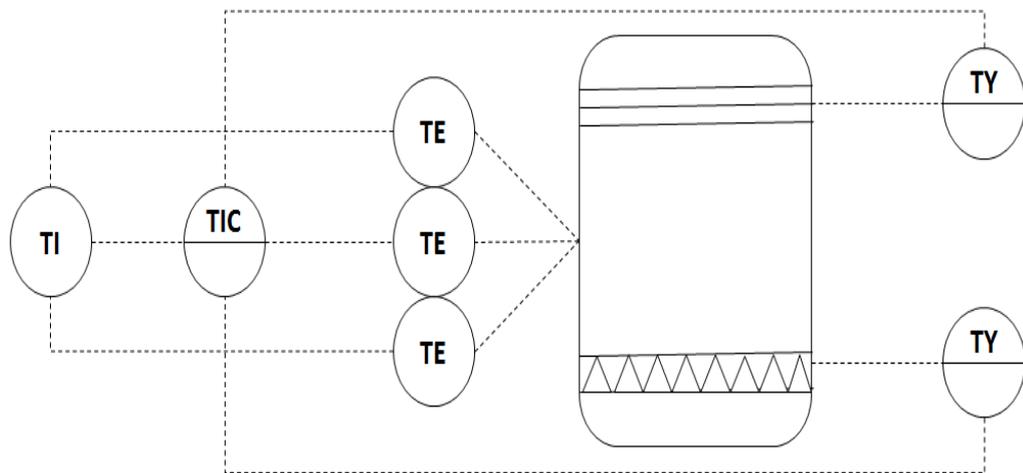


Figura 143 P&ID del sistema

2.9 Diagrama eléctrico del sistema.

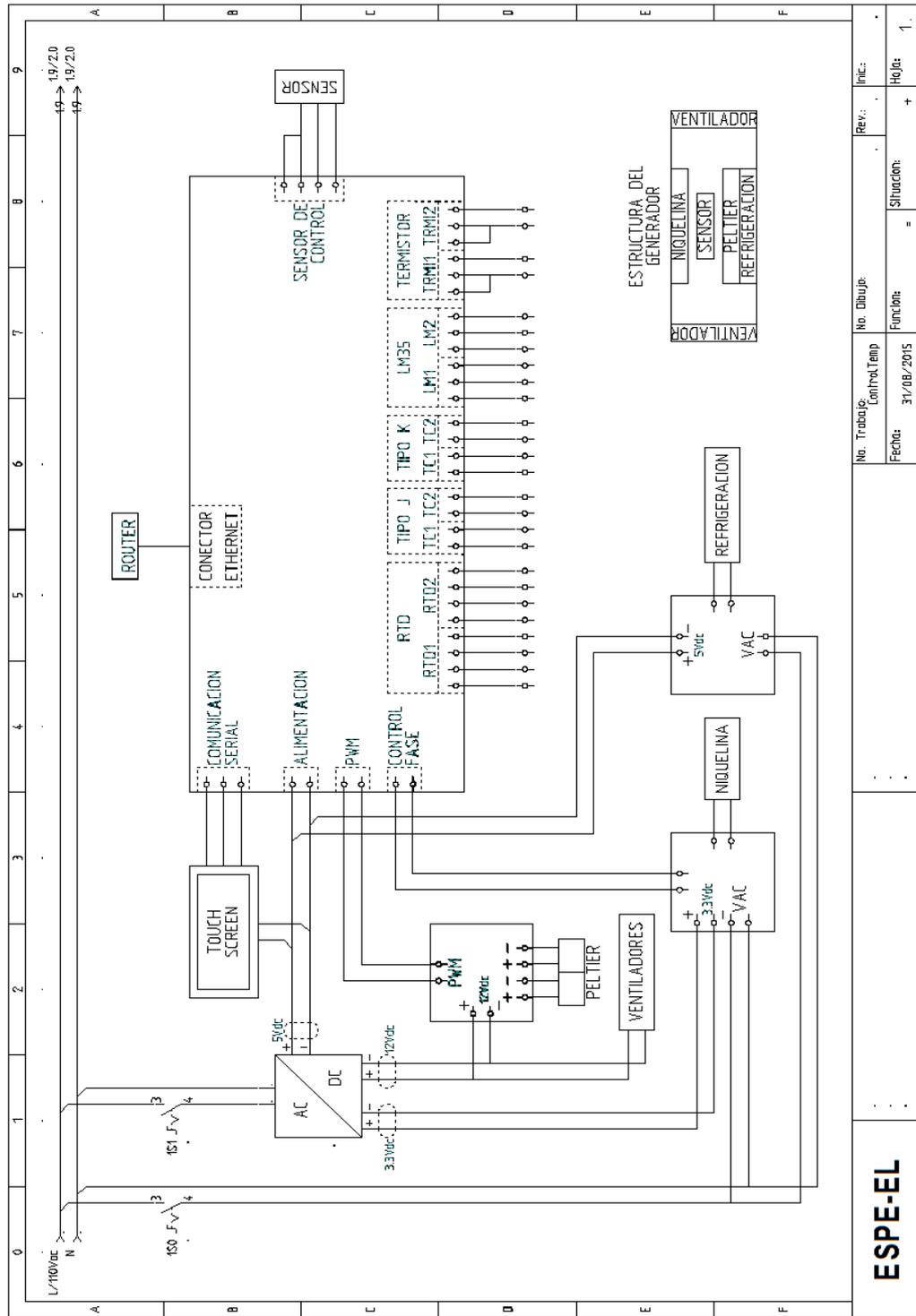


Figura 144 Diagrama eléctrico

CAPÍTULO III

3 PRUEBAS Y RESULTADOS OBTENIDOS.

En este capítulo se realizó el análisis del comportamiento práctico de los sensores de temperatura, las pruebas realizadas con el equipo ayudaron a conocer las variaciones en la medida de temperatura entre dos sensores del mismo tipo, estos resultados nos ayudan a comprender el funcionamiento de cada uno de estos elementos, demostrando la diferencia en características y funcionamiento de estos sensores en base a curvas características obtenidas por medio de un generador de temperatura. En tanto que los resultados de las pruebas ayudaron a la determinación de importantes conclusiones.

3.1 Curva característica obtenida del sensor Im35.

Para obtener la curva característica del sensor LM35 debemos tener en cuenta que su rango de trabajo es de -55° a $+150^{\circ}\text{C}$, las pruebas realizadas fueron tomadas para dos sensores del mismo tipo en un rango de 0° a 100°C , obteniendo como resultado los valores indicados en la Tabla 9 y las curvas características desarrolladas en LabVIEW y Excel que se indican en la Figura 145 y Figura 146.

Puede notarse una ligera diferencia de medida de temperatura en el mismo instante, debido posiblemente a la ubicación de los sensores y agentes externos como temperatura y humedad.

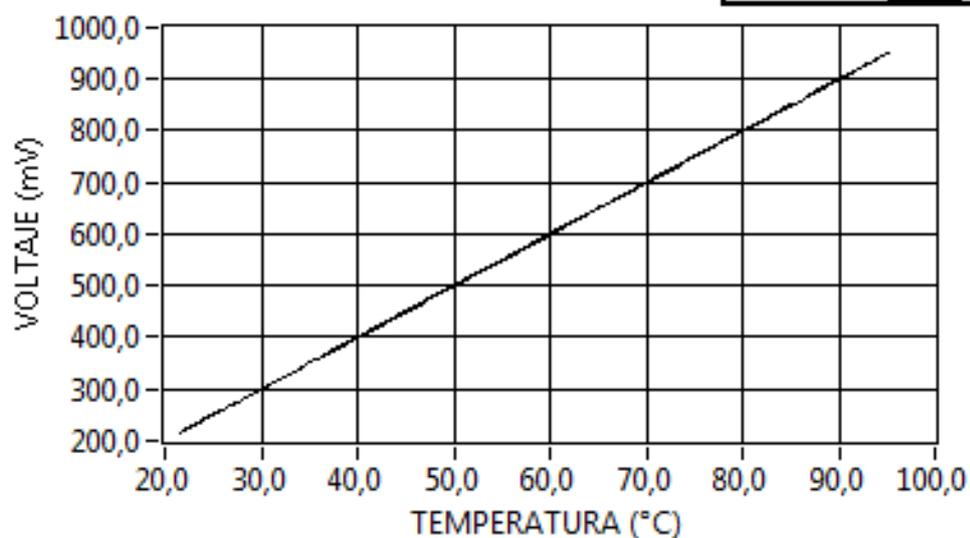
Además como se observa en la Figura 145 en la cual tenemos una escala en decenas y la Figura 146 con escala en décimas, vemos que los dos sensores comparados no presentan linealidad y a la vez existe pequeñas perturbaciones en la medición, esto puede darse por ruido existente en el interior del generador en la recirculación de fluido en el sistema de refrigeración.

Tabla 9

Voltaje vs temperatura sensor LM35

TEMP1 (°C)	VOLTAJE 1 (mV)	TEMP 2 (°C)	VOLTAJE 2 (mV)
22,81	228	18,3	183
22,34	223	17,33	173
22,32	223	18,47	184
22,18	221	18,62	186
22,38	223	18,59	185
64,96	649	67,42	674
70,41	704	68,07	680
68,11	681	68,85	688
73,77	737	69,39	693
73,1	731	70,4	704
66,48	664	70,18	701
71,46	714	71,43	714
68,54	685	71,8	718
73,5	735	72,03	720
73,99	739	73,21	732
73,74	737	74,36	743
74,22	742	74,25	742
75,81	758	75,72	757
73,28	732	75,71	757
77,17	771	76,56	765
84,33	843	91,2	912
84,05	840	91,71	917
79,53	795	92,04	920
86,42	864	93,1	931
87,53	875	93,87	938
92,38	923	94,75	947
90,18	901	95,44	954
88,76	887	95,97	959
86,29	862	96,16	961
92,01	920	97,31	973
89,67	896	98,17	981
92,96	929	99,96	999
92,6	926	100,39	1003
93,76	937	100,97	1009
93,73	937	101,84	1018
95,22	952	102,78	1027

CURVA CARACTERISTICA LM35 1



CURVA CARACTERISTICA LM35 2

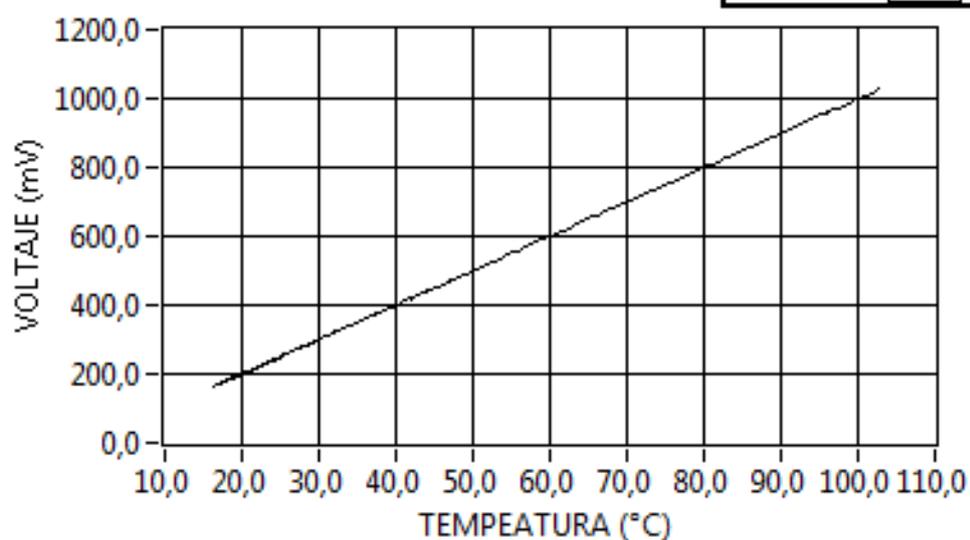


Figura 145 Curvas características LM35 1 y LM35 2 (LabVIEW)

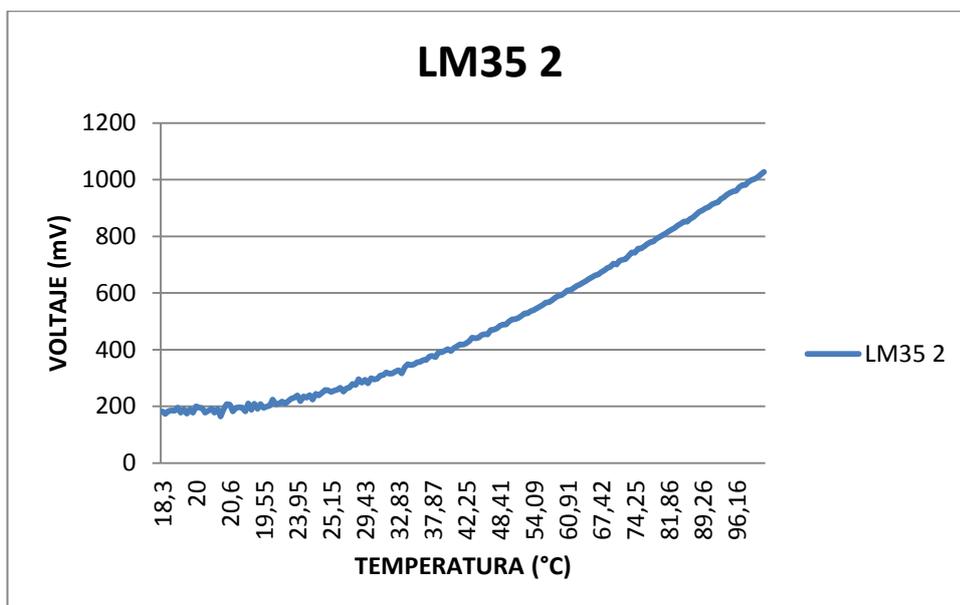
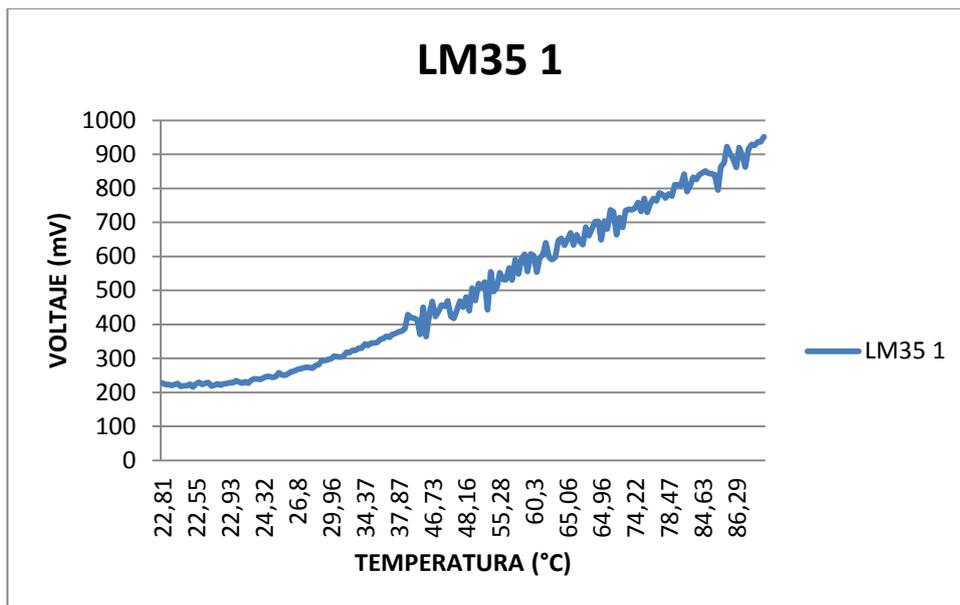


Figura 146 Curvas características LM35 1 y LM35 2 (excel)

3.2 Curvas características obtenidas de los termistores.

En la adquisición de la curva característica de los termistores tipo NTC de $1k\Omega$, $5k\Omega$ y $10k\Omega$, las pruebas se las realizó en un rango de 0° a 100°C , (su rango de trabajo es de -40° a $+125^\circ\text{C}$), luego de las mediciones se obtuvo los valores indicados en las Tabla 10, Tabla 11 y Tabla 12.

Por medio de los valores medidos y mostrados en la Tabla 10, Tabla 11 y Tabla 12 se obtienen las curvas características entre dos termistores de $1K\Omega$, $5K\Omega$, $10K\Omega$ respectivamente, estas curvas se muestran en la Figura 147, Figura 148, Figura 149, Figura 150, Figura 151, Figura 152.

Como se puede notar las medidas de temperaturas en los dos sensores difieren ligeramente en el instante de la medición, debido posiblemente a la ubicación y al proceso de generación, también influyen agentes externos como temperatura y humedad que afectan en el instante de toma de datos.

Para las curvas de la Figura 147, Figura 149 y Figura 151 tenemos una escala en saltos de 20 unidades y para las Figura 148, Figura 150 y Figura 152 su variación en la escala es en décimas; para los dos termistores de $1K\Omega$, $5K\Omega$, $10K\Omega$ comparados se observa que no existe linealidad, es exacto, estable y muy sensible en la medición.

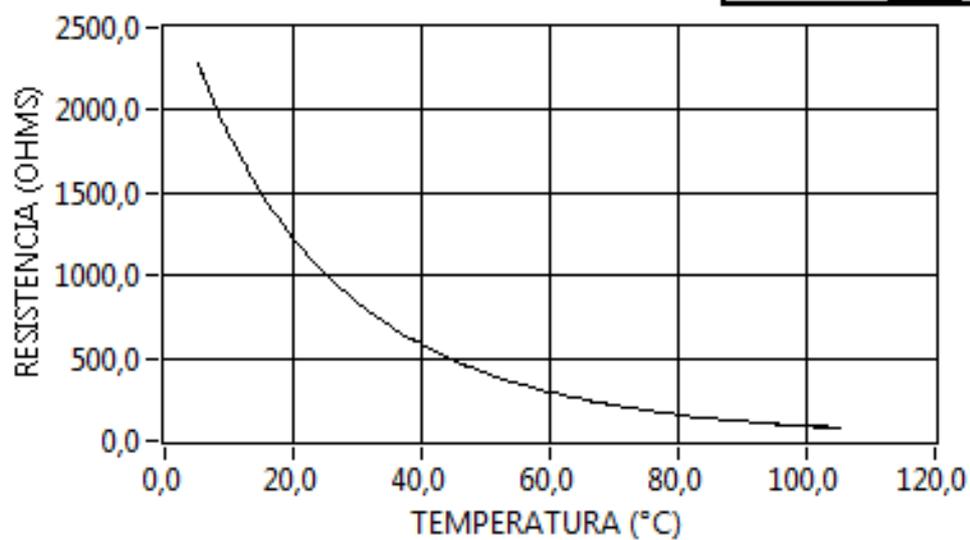
3.2.1 Termistor de 1K Ω .

Tabla 10

Resistencia vs temperatura del termistor 1K

TEMP1 (°C)	RESISTENCIA 1 (Ω)	TEMP 2 (°C)	RESISTENCIA 2 (Ω)
0,35	2824	5,19	2279
0,3	2830	5,11	2288
0,25	2836	5,11	2287
0,42	2815	5,29	2270
0,44	2812	5,34	2265
1,85	2640	7,43	2068
2,06	2616	7,65	2050
2,24	2594	7,78	2038
2,25	2593	7,96	2022
2,64	2550	8,5	1976
2,86	2524	8,88	1944
3,16	2492	9,3	1910
4,68	2331	11,14	1767
4,69	2329	11,12	1769
5,16	2282	11,49	1742
54,58	350	64,05	257
55,31	342	64,22	256
55,25	343	64,39	255
56,02	334	65,59	245
82,97	145	89,54	119
83,61	142	90,03	118
84,51	138	91,87	112
85,78	133	92,4	110
86,21	131	92,62	110
85,86	133	92,51	110
87,93	125	91,66	113
89,16	121	93,16	108
99,96	89	101,62	85
101,77	85	105,23	77
100,09	89	103,83	80

CURVA CARACTERISTICA TERMISTOR 1K 1



CURVA CARACTERISTICA TERMISTOR 1K 2

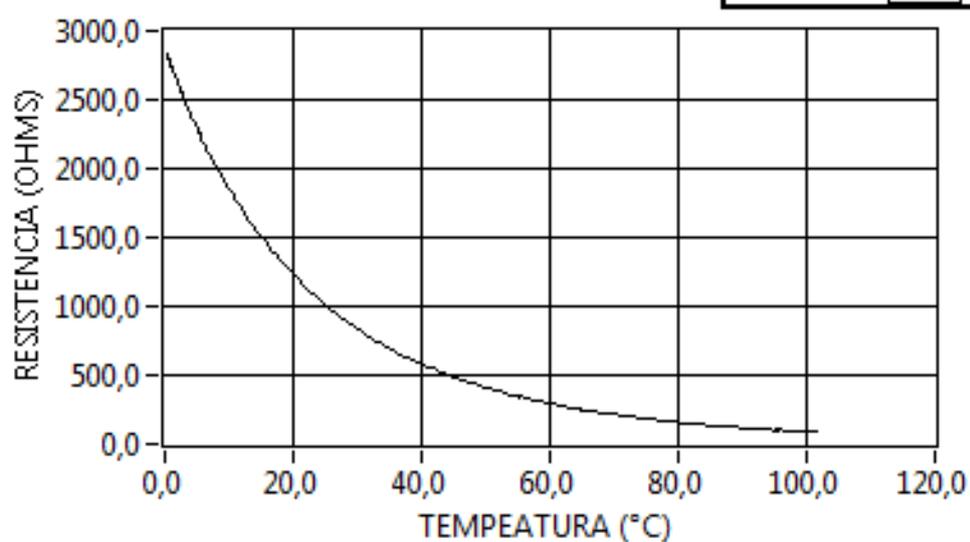


Figura 147 Curvas características termistor 1K (1 y 2) (LabVIEW)

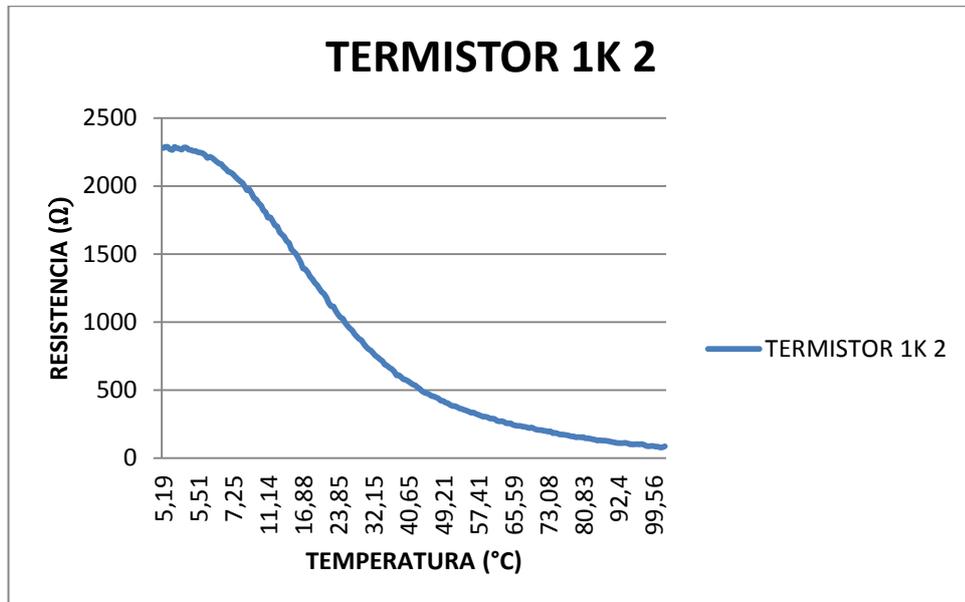
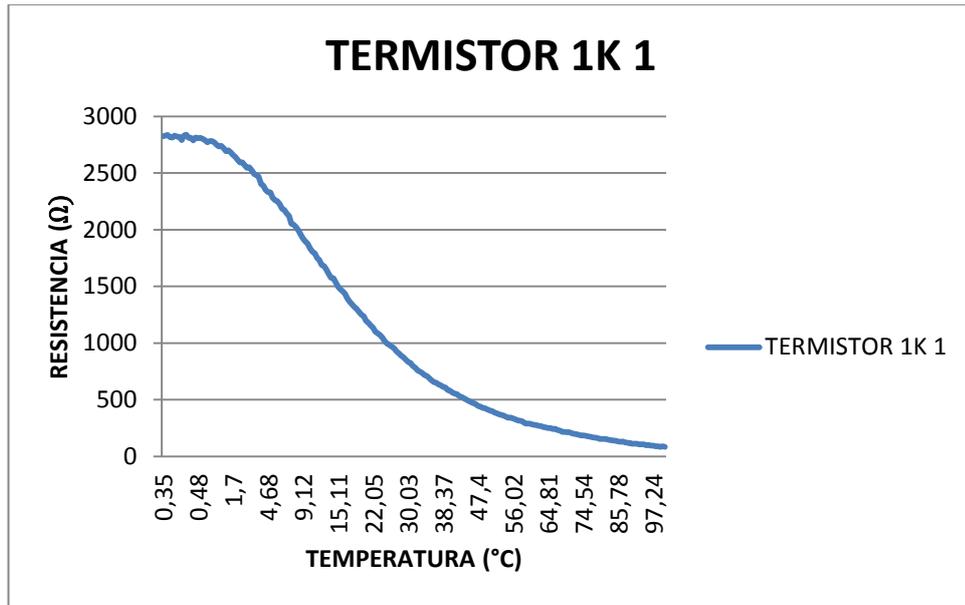


Figura 148 Curvas características termistor 1K (1 y 2) (excel)

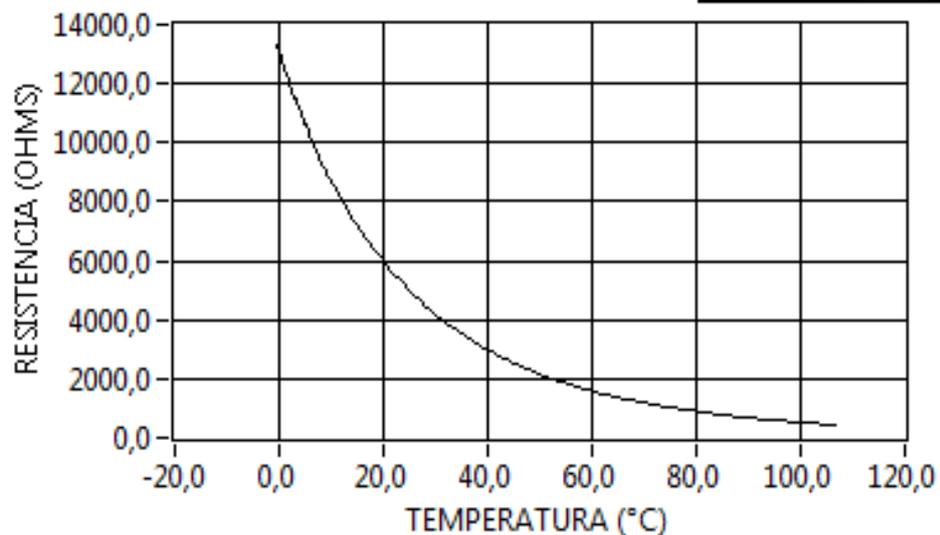
3.2.2 Termistor de 5K Ω .

Tabla 11

Resistencia vs temperatura del termistor 5K

TEMP1 (°C)	RESISTENCIA 1 (Ω)	TEMP 2 (°C)	RESISTENCIA 2 (Ω)
-6,48	17228	-0,07	13118
-6,76	17439	-0,36	13277
-6,57	17296	-0,16	13168
-6,76	17442	-0,28	13233
1,29	12399	5,93	10265
1,8	12142	6,3	10114
2,12	11985	6,55	10014
10,24	8659	12,41	7959
10,62	8528	12,79	7841
11,2	8338	13,36	7672
11,85	8133	13,79	7548
12,25	8006	14	7485
12,74	7859	14,49	7348
13,58	7607	15,03	7198
14,05	7473	15,44	7087
14,75	7274	16,06	6924
15,08	7186	16,33	6852
15,9	6965	16,97	6691
60,83	1538	59,83	1585
61,62	1502	60,43	1557
62,79	1451	61,62	1502
63,5	1421	62,07	1482
98,34	544	97,9	550
99,76	524	99,18	532
100,31	517	99,74	525
101,39	503	100,83	510
102,01	495	101,19	506

CURVA CARACTERISTICA TERMISTOR 5K 1

Plot 0 

CURVA CARACTERISTICA TERMISTOR 5K 2

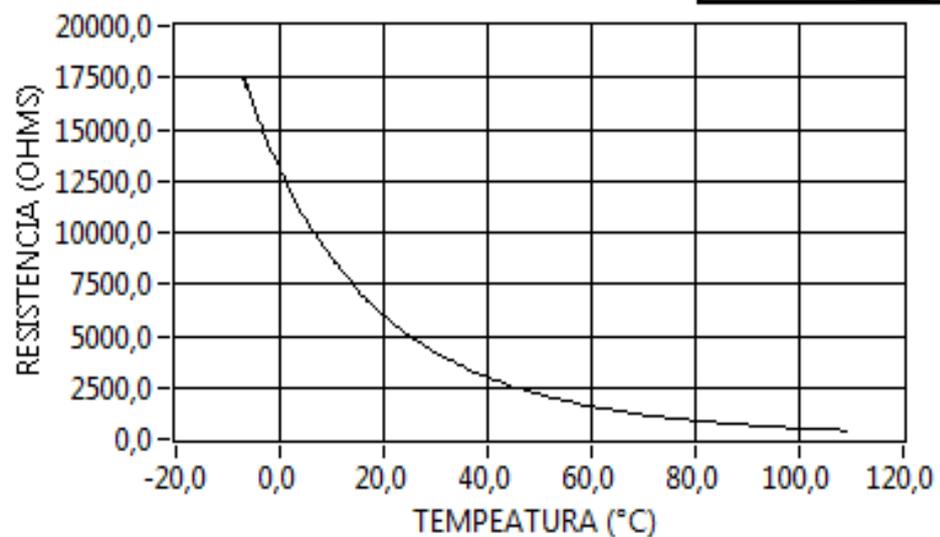
Plot 0 

Figura 149 Curvas Características Termistor 5K (1 y 2) (LabVIEW)

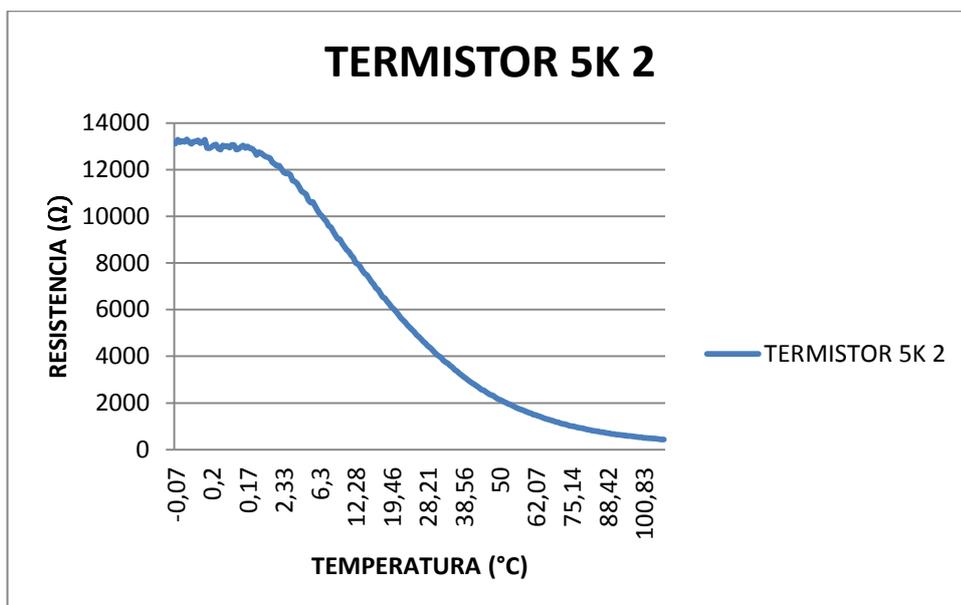
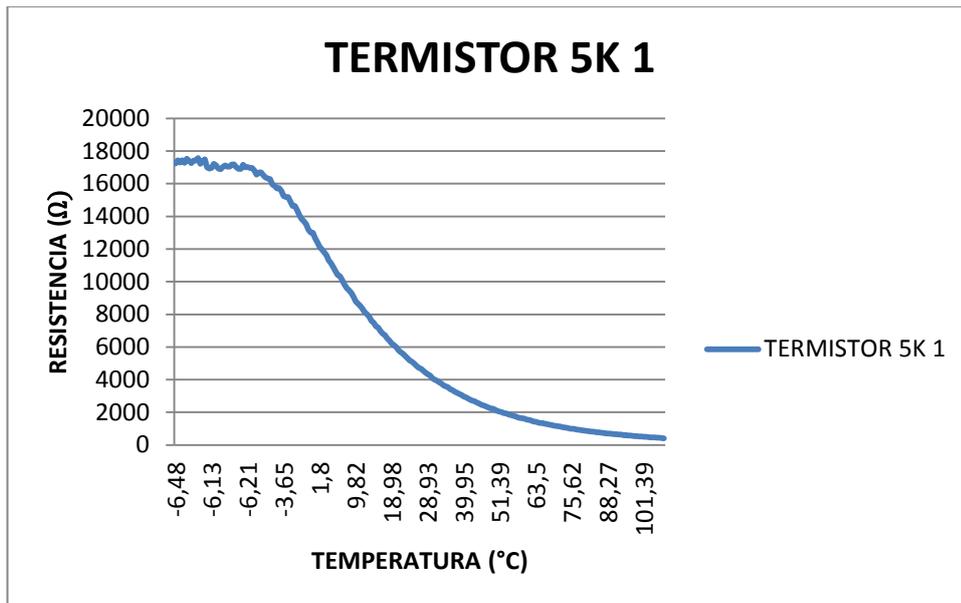


Figura 150 Curvas características termistor 5K (1 y 2) (excel)

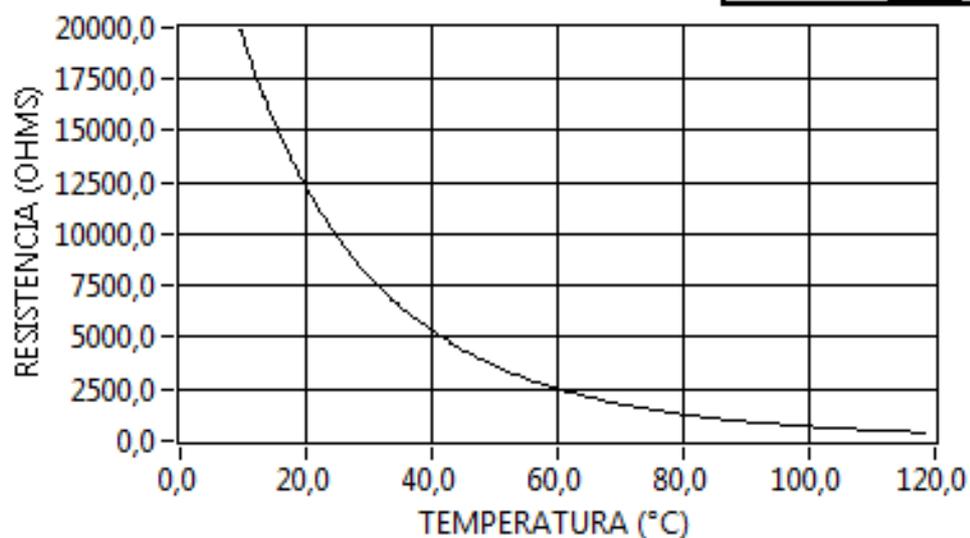
3.2.3 Termistor de 10K Ω .

Tabla 12

Resistencia vs Temperatura del Termistor 10K

TEMP1 (°C)	RESISTENCIA 1 (Ω)	TEMP 2 (°C)	RESISTENCIA 2 (Ω)
9,58	19864	9,58	19864
9,6	19853	9,6	19850
10,12	19363	15,78	14872
10,57	18954	16,23	14568
11,07	18509	16,97	14085
11,14	18454	17,07	14018
11,19	18406	17,37	13828
29,49	8147	40,75	5153
30,01	7972	41,31	5040
30,32	7870	42,03	4901
50,08	3601	66,35	2008
50,91	3491	67,23	1948
51,26	3447	67,83	1908
68,46	1867	86,68	1029
69	1833	87,18	1013
69,44	1806	88	988
70,31	1754	88,79	964
89,62	939	108,65	533
90,17	923	108,72	532
100,65	673	118,27	407
101,12	664	118,05	410

CURVA CARACTERISTICA TERMISTOR 10K 1



CURVA CARACTERISTICA TERMISTOR 10K 2

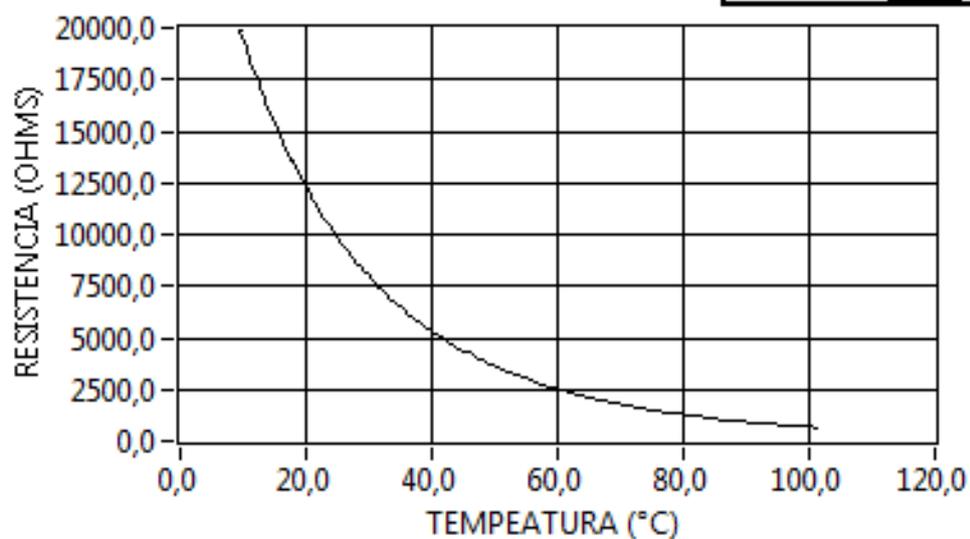


Figura 151 Curvas características termistor 10K (1 y 2) (LabVIEW)

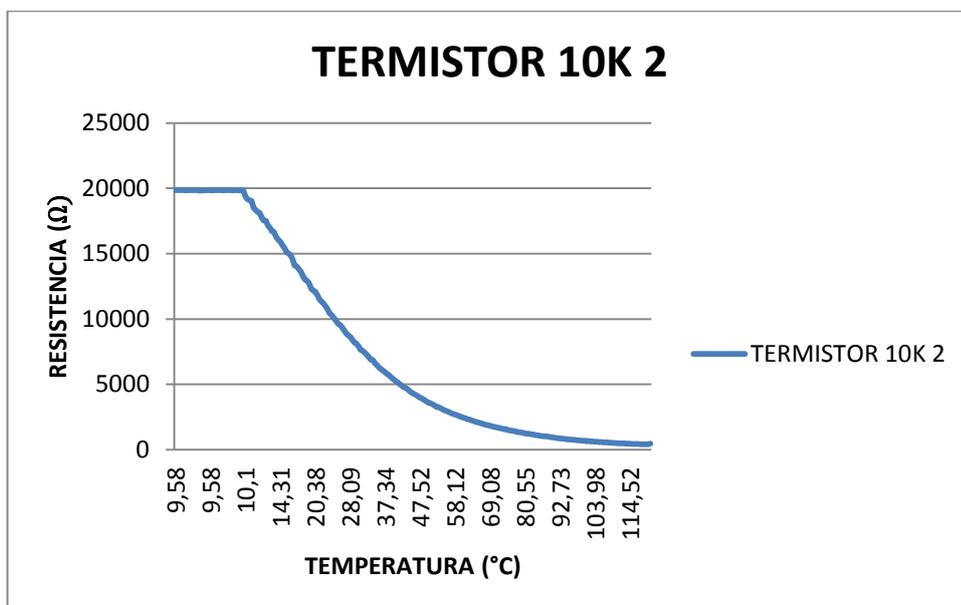
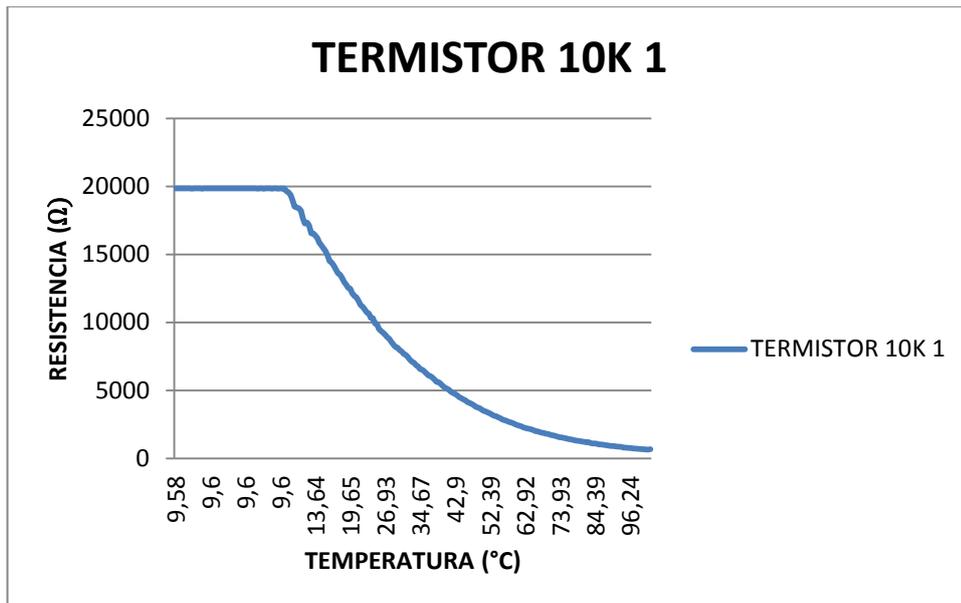


Figura 152 Curvas características termistor 10K (1 y 2) (excel)

3.3 Curvas características obtenidas de las termocuplas.

3.3.1 Termocupla tipo J.

Para obtener la curva característica de la termocupla tipo J, debemos tener en cuenta que su rango de trabajo de -180 a 750 °C, para nuestras pruebas trabajamos en un rango de 0° a 100 °C, obteniendo como resultado los valores indicados en la Tabla 13.

En base a estos valores se obtuvieron las curvas características entre dos termocuplas tipo J mediante LabVIEW y Excel, estos resultados se muestran en la Figura 153 y Figura 154.

Puede notarse de la misma manera una ligera diferencia entre los dos sensores del mismo tipo, posiblemente debido a la ubicación de los sensores y agentes externos como temperatura y humedad.

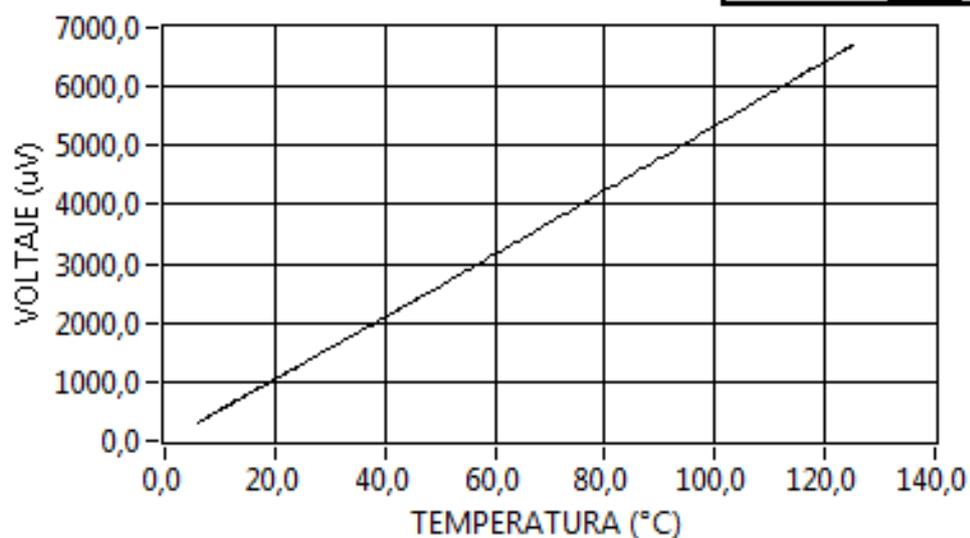
Además como se observa en la Figura 153, en la cual tenemos una escala en pasos de 20 unidades y la Figura 154 con escala en décimas, vemos que los dos sensores comparados se acercan a la linealidad.

Tabla 13

Voltaje vs temperatura termocupla J

TEMP1 (°C)	VOLTAJE 1 (uV)	TEMP 2 (°C)	VOLTAJE 2 (uV)
6	311	1,21	62
6,2	321	1,36	70
6,4	331	1,71	88
10	518	4,01	207
10,2	529	3,81	197
10,35	537	4,01	207
10,6	550	3,96	205
19	988	14,86	772
19,3	1004	15,21	790
19,6	1019	15,41	801
20,05	1043	16,16	840
40,5	2119	43,11	2258
41	2146	43,46	2276
41,45	2170	43,96	2303
50,7	2661	57,41	3019
89,15	4732	103,01	5491
90,25	4792	103,36	5510
91,25	4847	103,81	5535
99,6	5304	110,61	5910
100,45	5350	111,31	5949
101,2	5392	111,76	5974
101,9	5430	111,61	5965
102,95	5488	112,41	6009
103,75	5532	114,11	6103

CURVA CARACTERISTICA TERMOCUPLA J 1



CURVA CARACTERISTICA TERMOCUPLA J 2

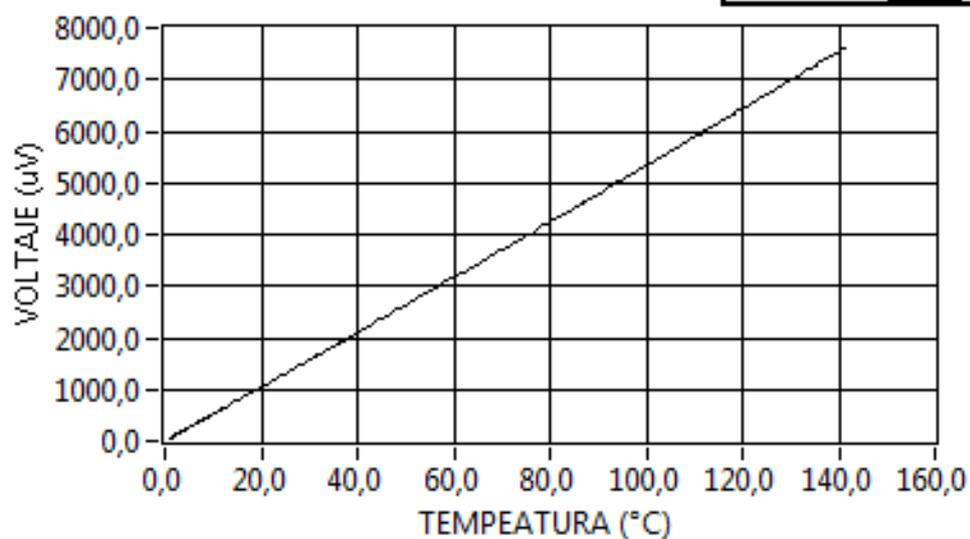


Figura 153 Curvas características termocupla J (1 y 2) (LabVIEW)

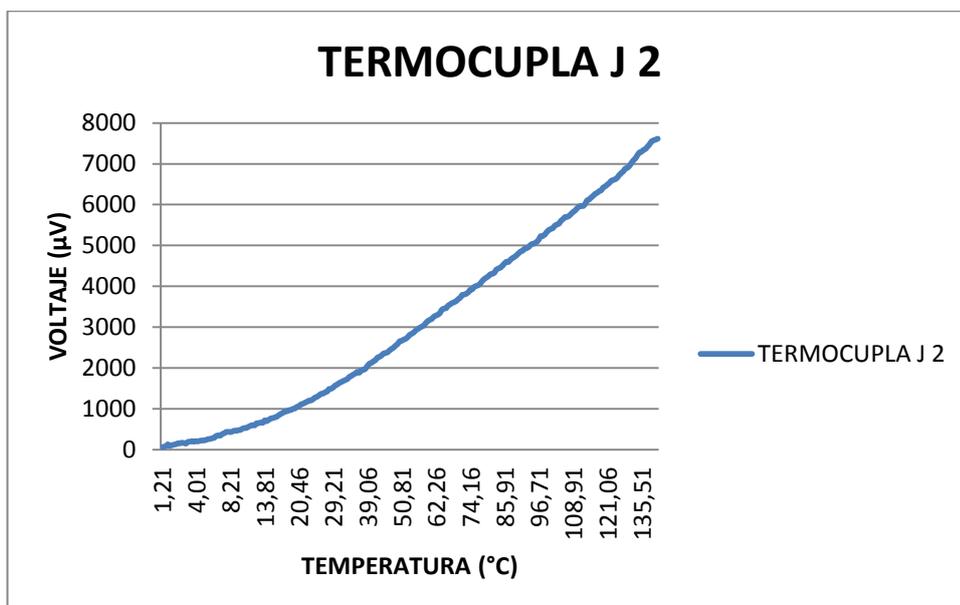
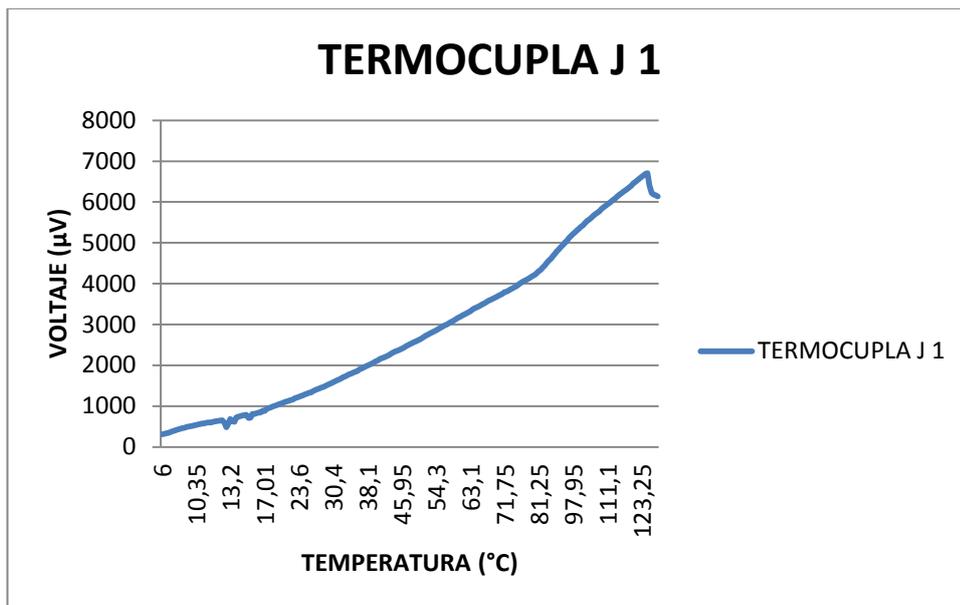


Figura 154 Curvas características termocupla J (1 y 2) (excel)

3.3.2 Termocupla tipo K.

Para graficar la curva característica de la termocupla tipo k, debemos fijarnos en su rango de trabajo de -180 a 1300 °C, en nuestras pruebas trabajamos en un rango de 0° a 100 °C, obteniendo como resultado lo indicado en la Tabla 14.

En base a los valores medidos indicados en la Tabla 14 se obtuvo las curvas características de las dos termocuplas tipo k mediante LabVIEW y Excel, que se muestran en la Figura 155 y Figura 156.

Se puede notar pequeñas diferencias en las mediciones realizadas al mismo tiempo con las dos termocuplas tipo k.

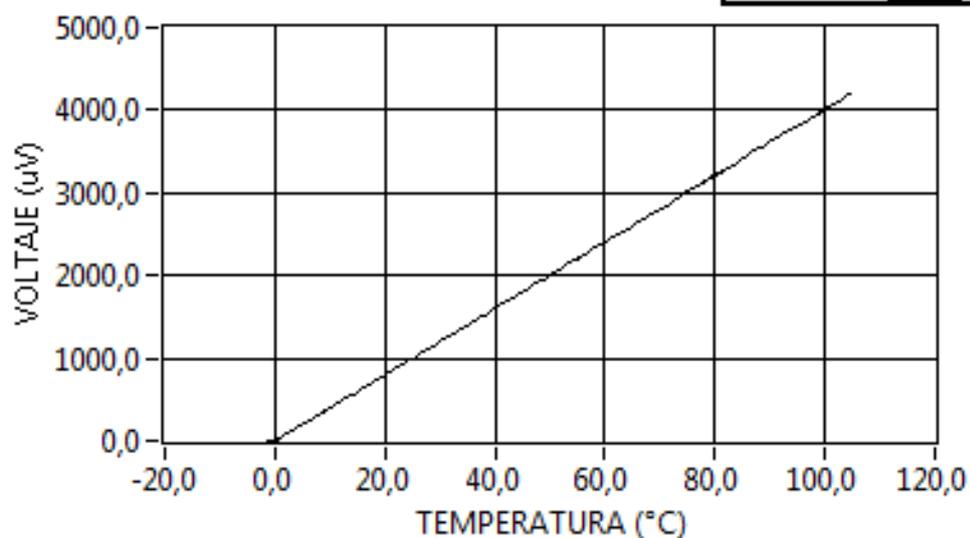
Debemos considerar que en la Figura 155 tenemos una escala en pasos de 20 unidades, mientras que en la Figura 156 su escala es en décimas, aquí podemos ver que los dos sensores comparados no presentan linealidad.

Tabla 14

Voltaje vs temperatura termocupla K

TEMP1 (°C)	VOLTAJE 1 (uV)	TEMP 2 (°C)	VOLTAJE 2 (uV)
-0,38	0	-0,25	0
0,15	5	0,15	5
-0,73	0	0,25	9
0,71	28	9,15	364
1,06	42	9,4	374
1,16	46	9,65	384
1,51	60	10,25	408
2,01	80	10,6	422
10,66	424	24,05	959
10,81	430	24,7	985
11,51	458	25,25	1007
59,71	2389	80,25	3218
60,11	2405	81,2	3256
60,66	2427	81,95	3286
89,86	3607	109,95	4422
90,81	3645	110,65	4450
98,86	3971	116,2	4676
100,46	4036	116,9	4704
100,66	4044	117,55	4731
101,16	4065	118,25	4759
101,51	4079	118,9	4786

CURVA CARACTERISTICA TERMOCUPLA K1



CURVA CARACTERISTICA TERMOCUPLA K2

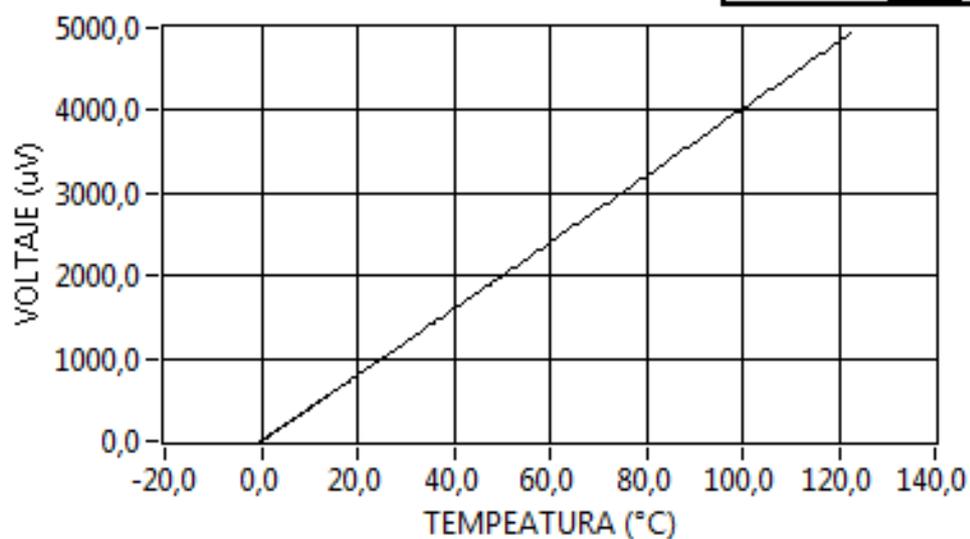


Figura 155 Curvas características termocupla k (1 y 2) (LabVIEW)

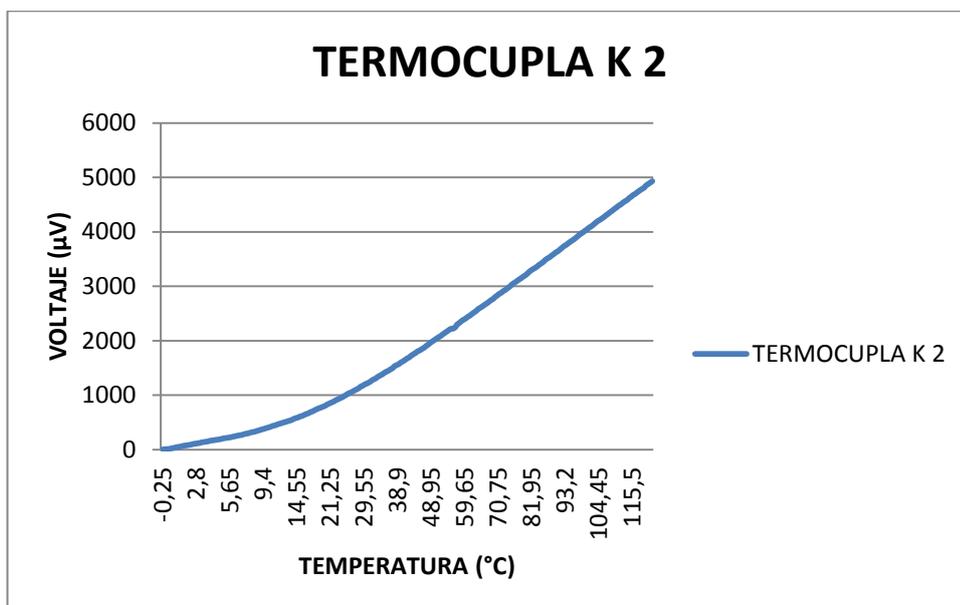
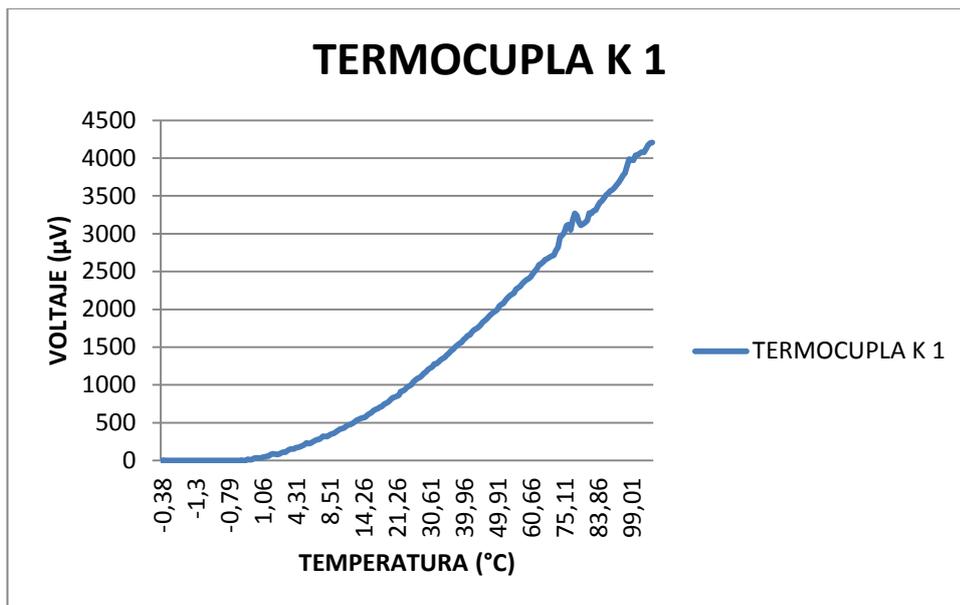


Figura 156 Curvas características termocupla k (1 y 2) (excel)

3.4 Curvas características obtenidas de las RTDs.

Para la presentación de la curva característica de las PT100, las pruebas se las realizó en un rango de 0° a 100 °C y 0° a 200 °C, conociendo que su rango de trabajo es de -260 a 630 °C, los resultados de las mediciones se indican en la Tabla 15 y Tabla 16, estas mediciones corresponden a configuraciones con PT100 de 2 y 3 Hilos.

Con las mediciones obtenidas se presentan las curvas características entre dos PT100 de 2 Hilos y dos PT100 de 3 Hilos, que se muestran en la Figura 157, Figura 158, Figura 159 y Figura 160, con la ayuda del software (LabVIEW y Excel)

Se observan que las mediciones difieren en una pequeñísima cantidad.

Las curvas creadas en LabVIEW utilizan una escala en pasos de 20 unidades y para las figuras graficadas en Excel su variación de escala es en décimas; estos valores y gráficas demuestran que estos sensores son casi lineales.

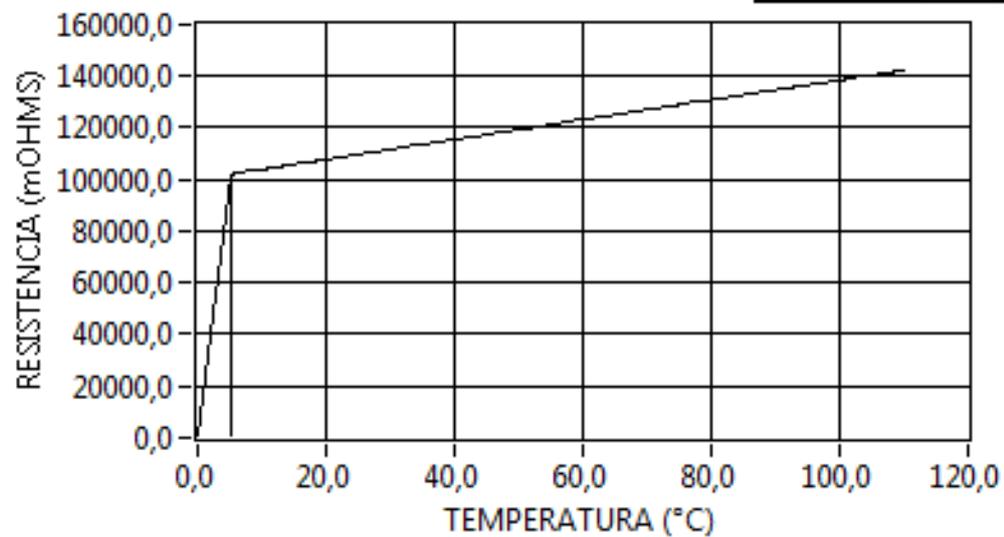
3.4.1 PT100 2 hilos.

Tabla 15

Resistencia vs temperatura PT100 2hilos

TEMP1 (°C)	RES 1 (mΩ)	TEMP 2 (°C)	RES 2 (mΩ)
5,06	101978	9,16	103577
6,86	102681	11,25	104393
8,25	103222	11,47	104475
8,78	103428	11,62	104533
8,9	103476	11,79	104602
9,01	103518	11,97	104671
21,85	108513	37,32	114506
22,07	108600	37,8	114690
22,3	108690	38,26	114872
38,84	115092	65,34	125292
39,26	115255	65,98	125537
39,63	115400	66,61	125777
40,04	115557	66,78	125845
40,43	115710	66,1	125582
40,84	115867	66,31	125665
40,08	115573	67,4	126082
39,61	115392	70,01	127082
40,02	115549	70,67	127334
42,13	116363	72,69	128104
43,84	117025	73,37	128367
44,21	117169	74,04	128622
61,12	123673	101,38	139028
61,48	123811	103,02	139652
107,64	141403	198,7	175378
107,94	141514	200,51	176043
108,24	141629	201	176226
108,54	141743	201,52	176416
108,83	141853	202,02	176601

CURVA CARACTERISTICA PT100 1



CURVA CARACTERISTICA PT100 2

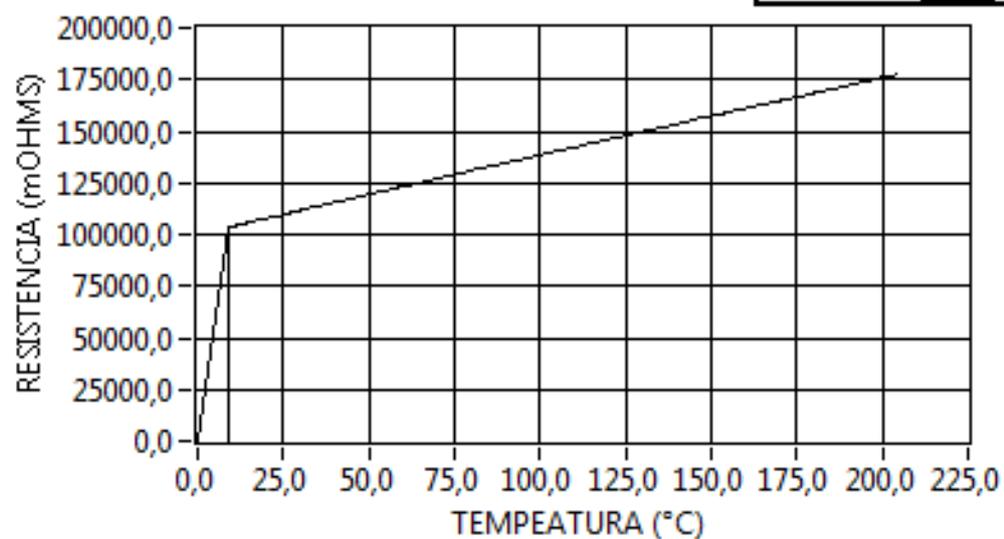


Figura 157 Curvas características PT100 2hilos (1 y 2) (LabVIEW)

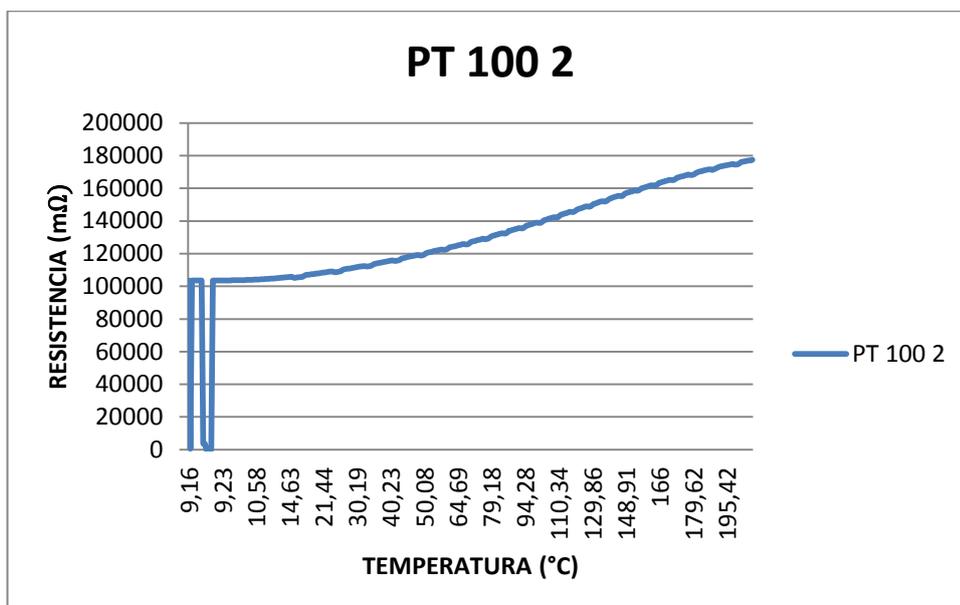
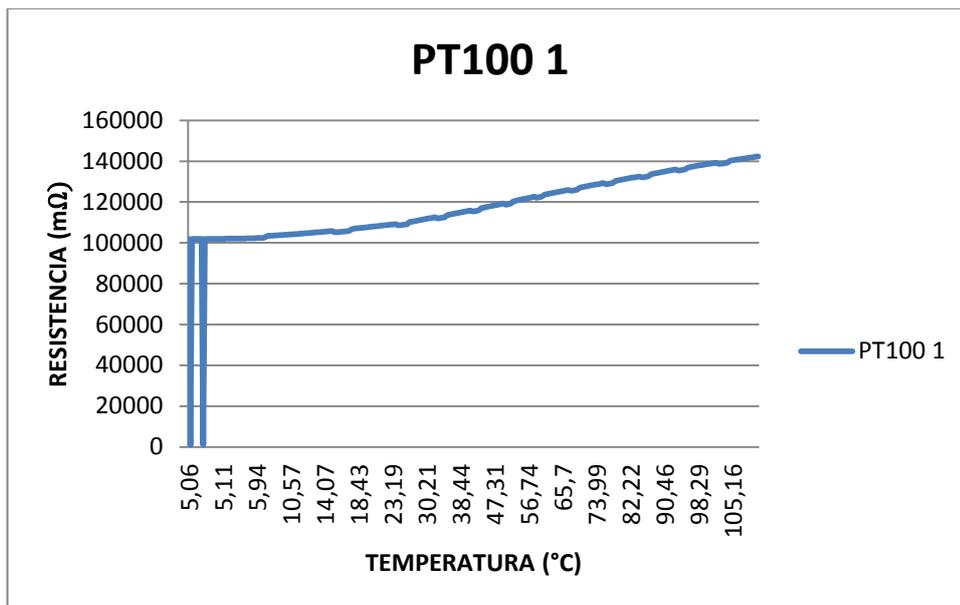


Figura 158 Curvas características PT100 2hilos (1 y 2) (excel)

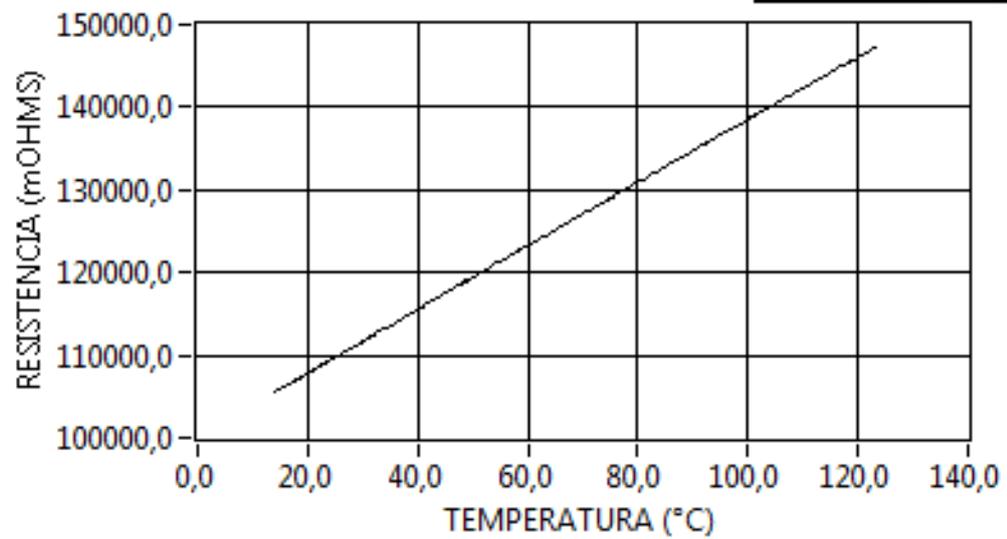
3.4.2 PT100 3 hilos.

Tabla 16

Resistencia vs temperatura PT100 3hilos

TEMP1 (°C)	RES 1 (mΩ)	TEMP 2 (°C)	RES 2 (mΩ)
13,99	105456	11,65	104545
17,55	106842	12,72	104964
17,57	106850	12,74	104971
17,61	106866	12,79	104990
17,64	106879	12,79	104990
17,68	106892	12,78	104987
17,73	106911	12,82	105003
17,77	106927	12,85	105014
17,8	106940	12,89	105030
17,81	106945	12,93	105046
17,97	107006	13,05	105093
18,02	107024	13,09	105109
18,09	107054	13,12	105120
18,16	107080	13,16	105133
18,17	107083	13,2	105152
18,22	107104	13,22	105159
18,28	107125	13,24	105165
18,29	107130	13,29	105186
18,39	107170	13,32	105197
18,47	107202	13,33	105202
18,55	107233	13,43	105239
18,62	107257	13,48	105260
18,7	107289	13,54	105281
18,79	107326	13,58	105297
18,88	107360	13,62	105316

CURVA CARACTERISTICA PT100 1



CURVA CARACTERISTICA PT100 2

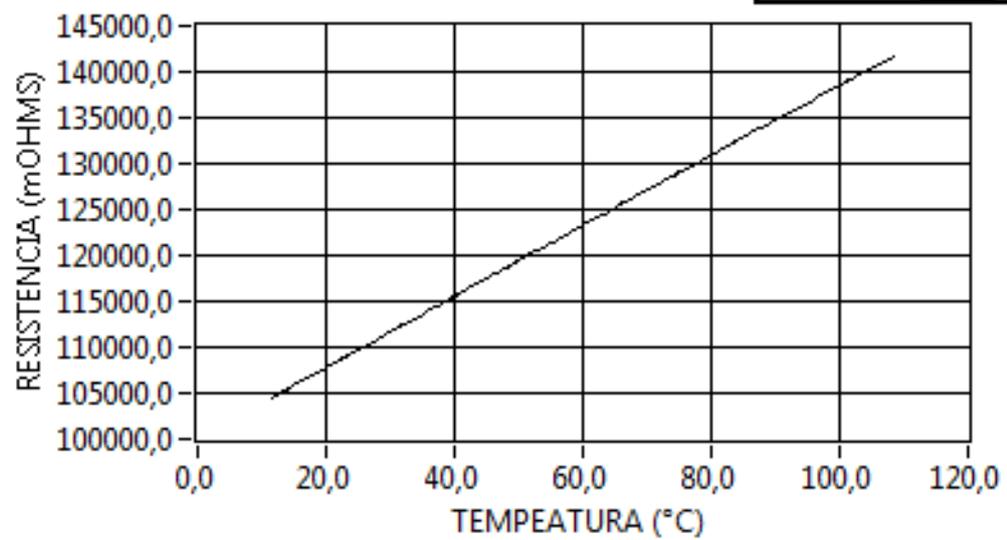


Figura 159 Curvas características PT100 3hilos (1 y 2) (LabVIEW)

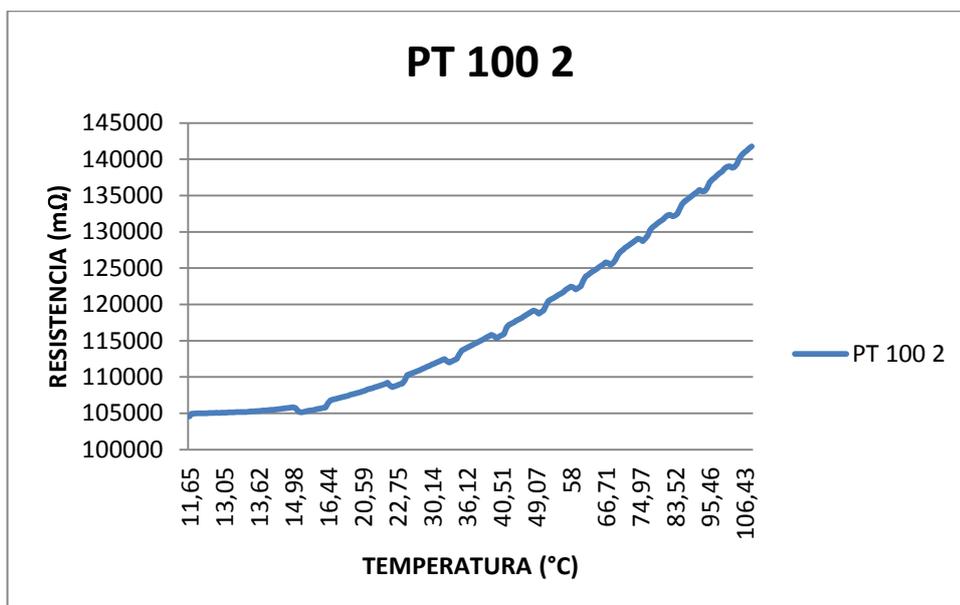
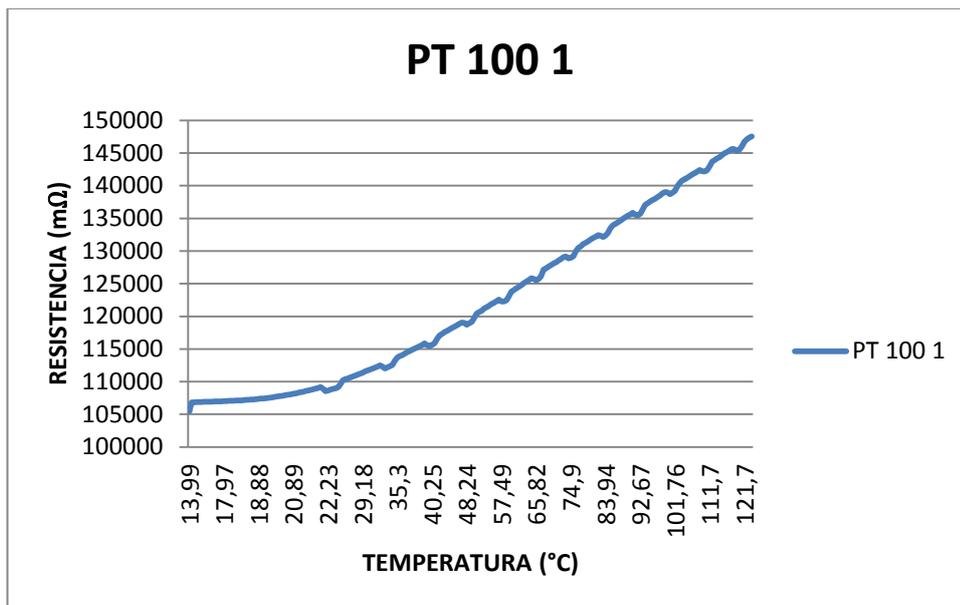


Figura 160 Curvas características PT100 3hilos (1 y 2) (excel)

3.4.3 Bland altman de los sensores de temperatura.

LM35.

En la Figura 161 visualizamos la gráfica de dos sensores LM35, donde se observa gráficas similares en un mismo proceso de temperatura.

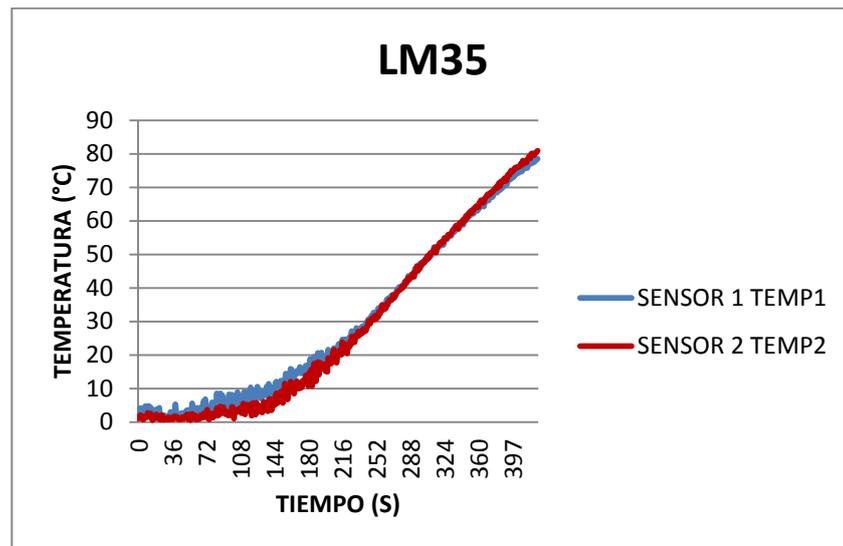


Figura 161 Temperatura vs tiempo LM35

La Figura 162 indica la prueba bilateral para dos muestras relacionadas, donde el intervalo de confianza para la diferencia entre las medias al 95% es de $]-1,319; -0,929[$ obteniendo datos resultantes como indica la Tabla 17.

Tabla 17
Prueba bilateral LM35

Diferencia	-1,124
t (Valor observado)	-11,333
t (Valor crítico)	1,966
GL	423
valor-p (bilateral)	< 0,0001
alfa	0,05

De ésta manera la interpretación de la prueba está dada por:

H_0 : La diferencia entre las medias es igual a 0.

H_a : La diferencia entre las medias es diferente de 0.

Puesto que el valor $-p$ computado es menor que el nivel de significación $\alpha = 0,05$, se debe rechazar la hipótesis nula H_0 , y aceptar la hipótesis alternativa H_a . El riesgo de rechazar la hipótesis nula H_0 cuando es verdadera es inferior al 0,01%.

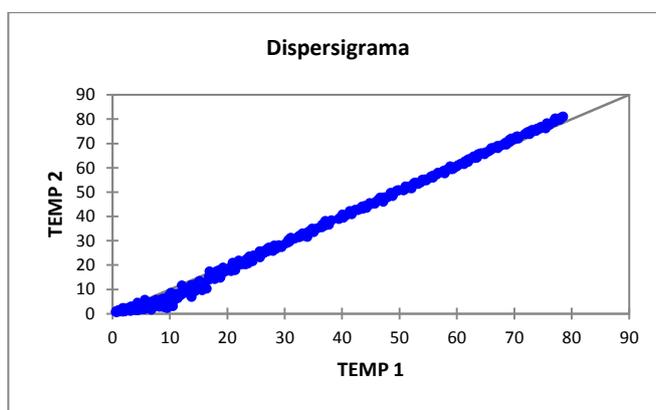


Figura 162 Prueba bilateral LM35

El análisis de Bland Altman tiene como intervalo de confianza de $]-5,126; 2,878[$, la Figura 163 muestra que el sensor LM35 en el rango de 5°C a 15°C sale del intervalo de confianza.

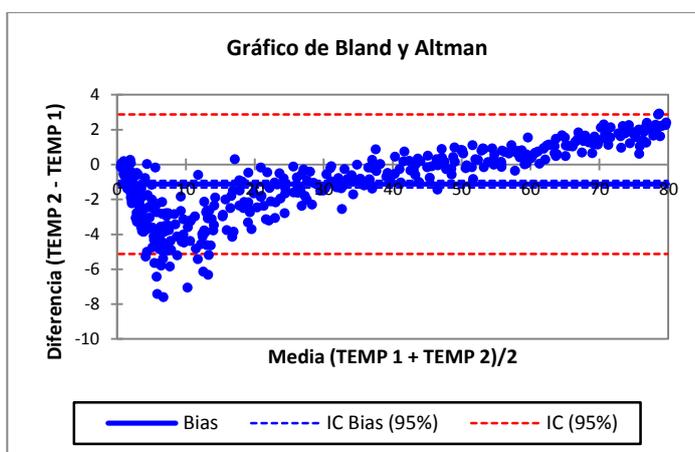


Figura 163 Bland altman LM35

Termistor 1K Ω .

La Figura 164 muestra la gráfica de dos sensores Termistor 1K Ω , obteniendo como resultado gráficas similares dentro de un mismo proceso de temperatura.

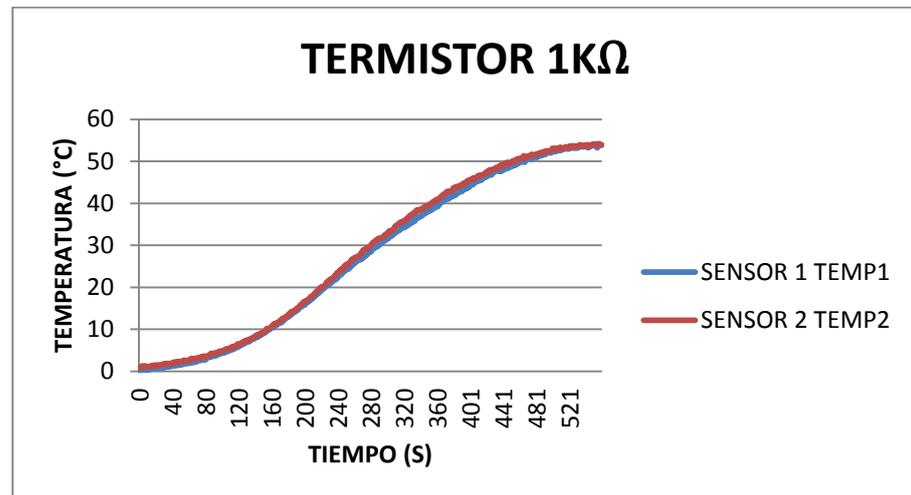


Figura 164 Temperatura vs tiempo termistor 1K Ω

La Figura 165 indica la prueba bilateral para dos muestras relacionadas, donde el intervalo de confianza para la diferencia entre las medias al 95% es de]0,796; 0,873[obteniendo datos resultantes como indica la Tabla 18.

Tabla 18
Prueba bilateral termistor 1K Ω

Diferencia	0,732
t (Valor observado)	10,173
t (Valor crítico)	1,964
GL	611
valor-p (bilateral)	< 0,0001
alfa	0,05

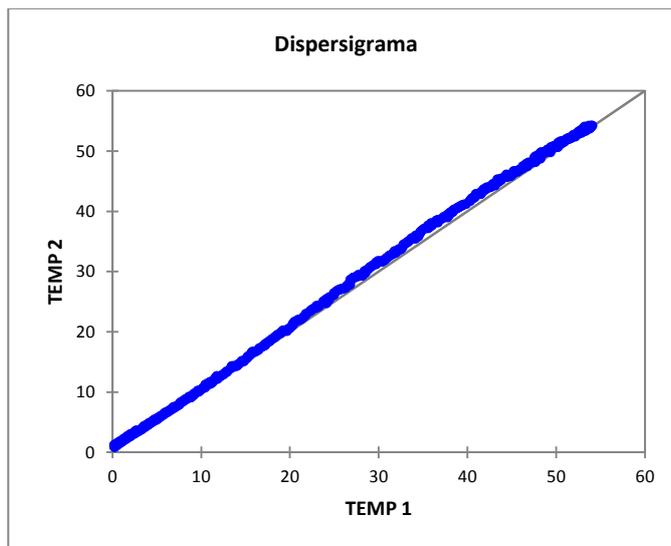


Figura 165 Prueba bilateral termistor 1K Ω

El análisis de Bland Altman tiene un intervalo de confianza de $]-0,081; 1,751[$ donde la Figura 166 muestra que el Termistor 1K Ω en el rango de temperatura de 30 a 45 °C sobrepasa el intervalo de confianza.

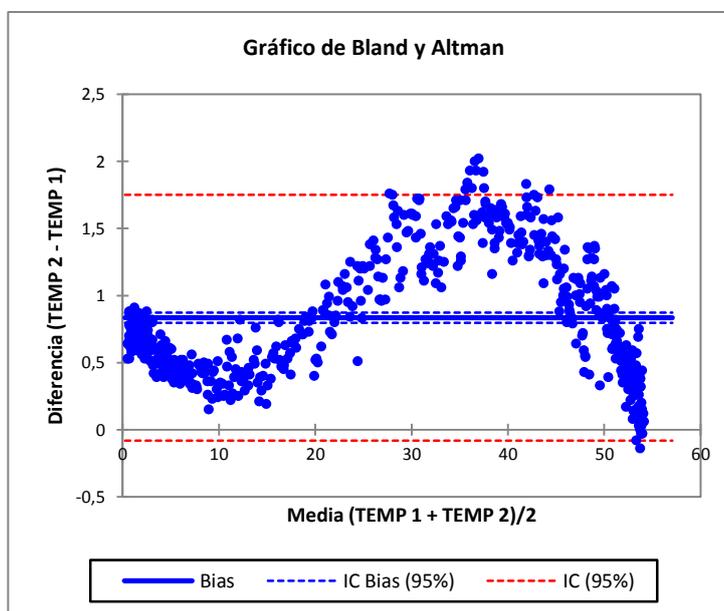


Figura 166 Bland altman termistor 1k Ω

Termistor 5K Ω .

La Figura 167 muestra la gráfica de dos sensores de Termistor 5K Ω , obteniendo como resultado gráficas similares dentro de un mismo proceso de temperatura.

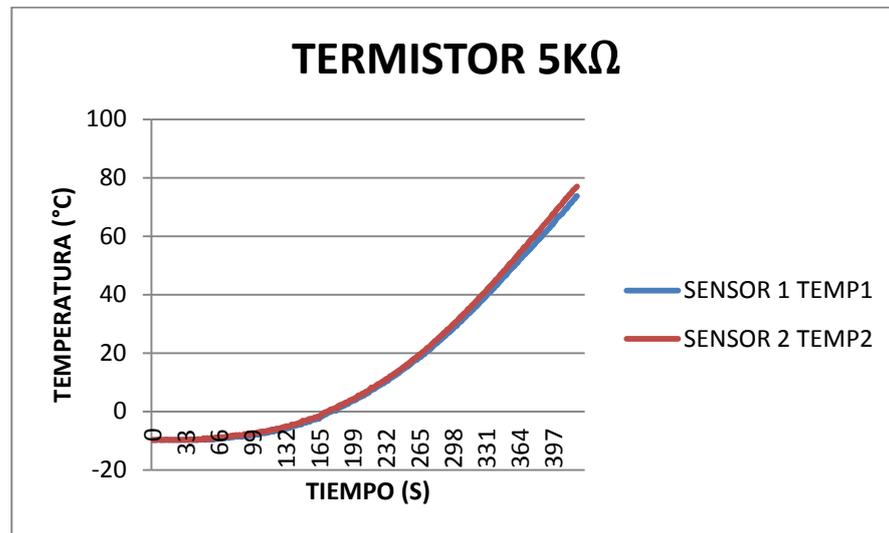


Figura 167 Temperatura vs tiempo termistor 5K Ω

La Figura 168 indica la prueba bilateral para dos muestras, donde el intervalo de confianza para la diferencia entre las medias al 95% es de]1,039; 1,225[obteniendo resultados como indica la Tabla 19.

Tabla 19

Prueba bilateral termistor 5K Ω

Diferencia	1,132
t (Valor observado)	23,939
t (Valor crítico)	1,966
GL	420
valor-p (bilateral)	< 0,0001
alfa	0,05

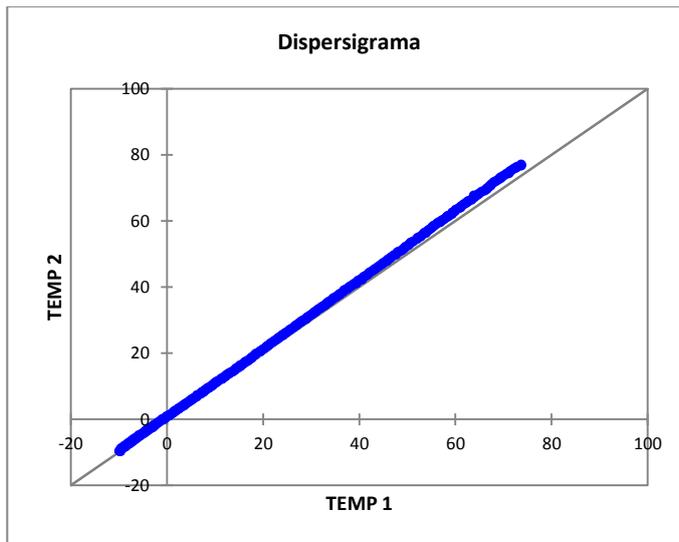


Figura 168 Prueba bilateral termistor 5K Ω

El análisis de Bland Altman tiene como intervalo de confianza de]-0,770; 3,034[donde la Figura 169 muestra que el sensor Termistor 5K Ω a partir de los 60 °C está fuera del intervalo de confianza.

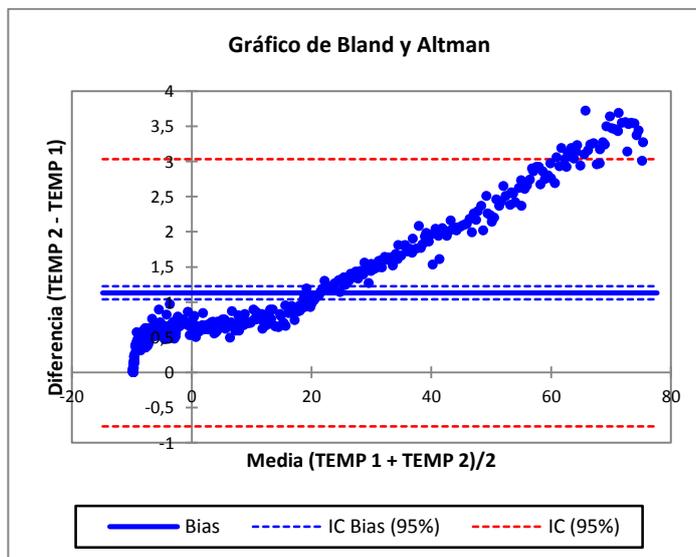


Figura 169 Bland altman termistor 5K Ω

Termistor 10K Ω .

La Figura 170 muestra la gráfica de dos sensores Termistor 10K Ω , obteniendo como resultado gráficas similares dentro de un mismo proceso de temperatura.

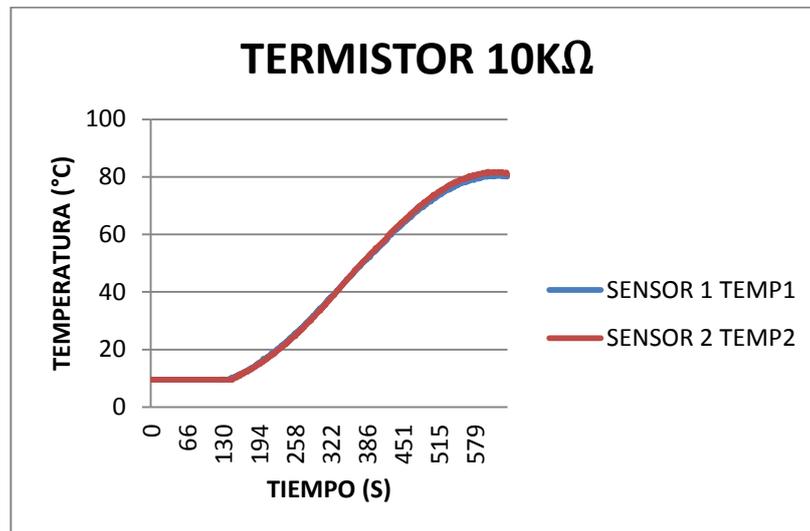


Figura 170 Temperatura vs tiempo termistor 10K Ω

La Figura 171 indica la prueba bilateral para dos muestras relacionadas, donde el intervalo de confianza para la diferencia entre las medias al 95% es de]0,229; 0,324[obteniendo datos resultantes como indica la Tabla 20.

Tabla 20
Prueba bilateral termistor 10K Ω

Diferencia	0,277
t (Valor observado)	11,460
t (Valor crítico)	1,964
GL	634
valor-p (bilateral)	< 0,0001
alfa	0,05

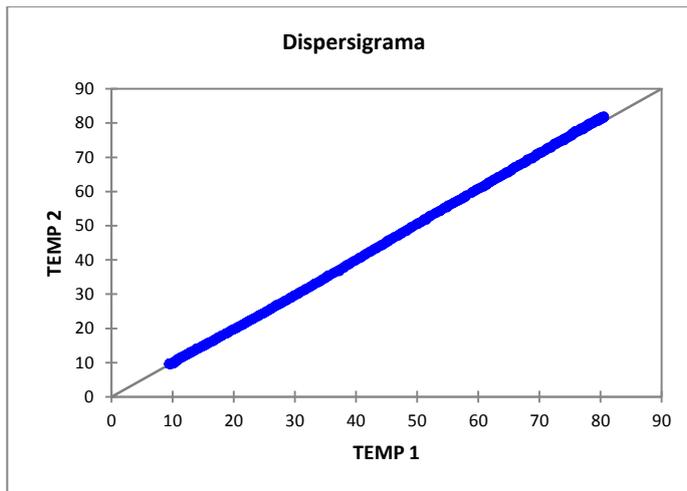


Figura 171 Prueba bilateral termistor 10KΩ

El análisis de Bland Altman tiene como intervalo de confianza de $]-0,916; 1,469[$ donde la Figura 172 muestra que el sensor Termistor 10KΩ en 78 °C sobre pasa su intervalo de confianza.

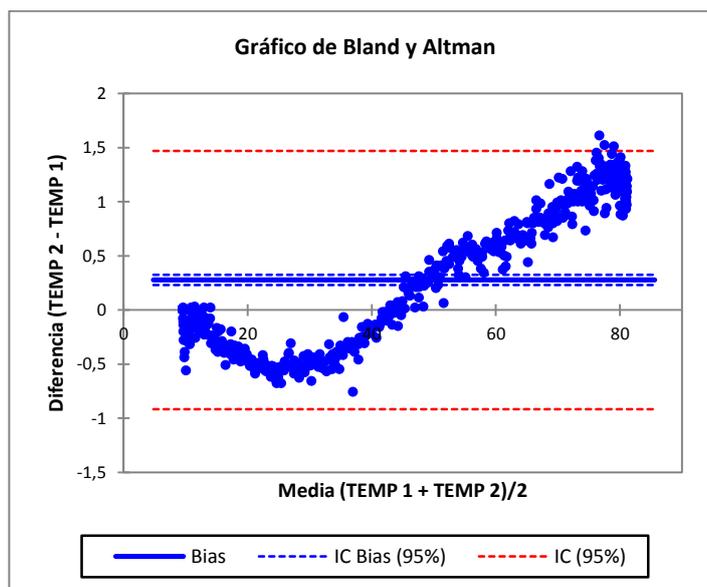


Figura 172 Bland altman termistor 10kΩ

Termocupla J.

La Figura 173 muestra la gráfica de dos sensores Termocupla tipo J, obteniendo como resultado gráficas similares dentro de un mismo proceso de temperatura.

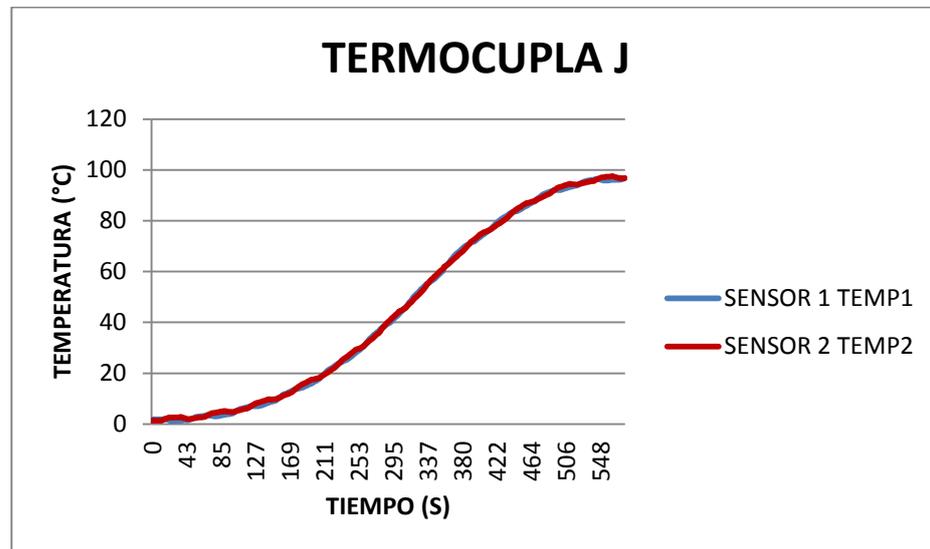


Figura 173 Temperatura vs tiempo termocupla J

La Figura 174 indica la prueba bilateral para dos muestras relacionadas, donde el intervalo de confianza para la diferencia entre las medias al 95% es de $]-0,306; -0,177[$ obteniendo datos resultantes como indica la Tabla 21.

Tabla 21

Prueba bilateral termocupla J

Diferencia	-0,241
t (Valor observado)	-7,390
t (Valor crítico)	1,964
GL	576
valor-p (bilateral)	< 0,0001
alfa	0,05

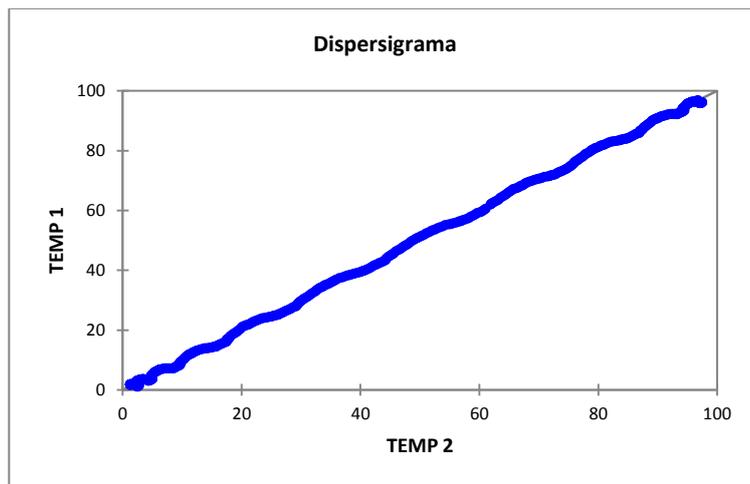


Figura 174 Prueba bilateral termocupla J

El análisis de Bland Altman tiene como intervalo de confianza de $]-1,779; 1,297[$ donde la Figura 175 muestra que el sensor Termocupla tipo J se encuentra dentro del rango del intervalo de confianza .

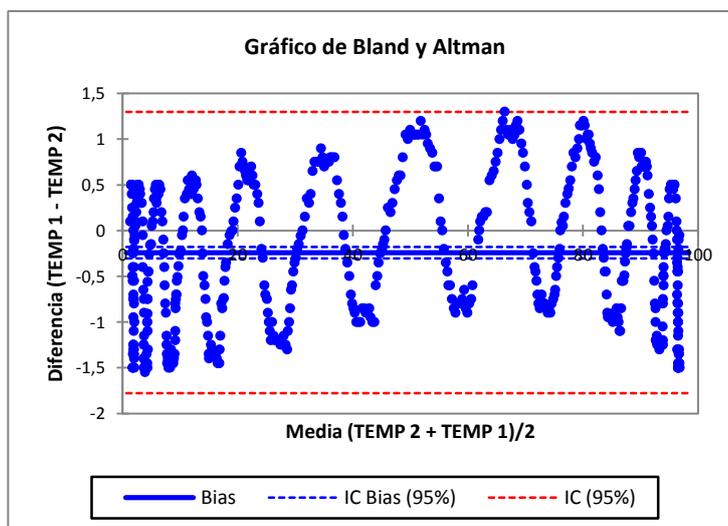


Figura 175 Bland altman termocupla J

Termocupla K.

La Figura 176 muestra la gráfica de dos sensores Termocupla tipo K, obteniendo como resultado gráficas similares dentro de un mismo proceso de temperatura.

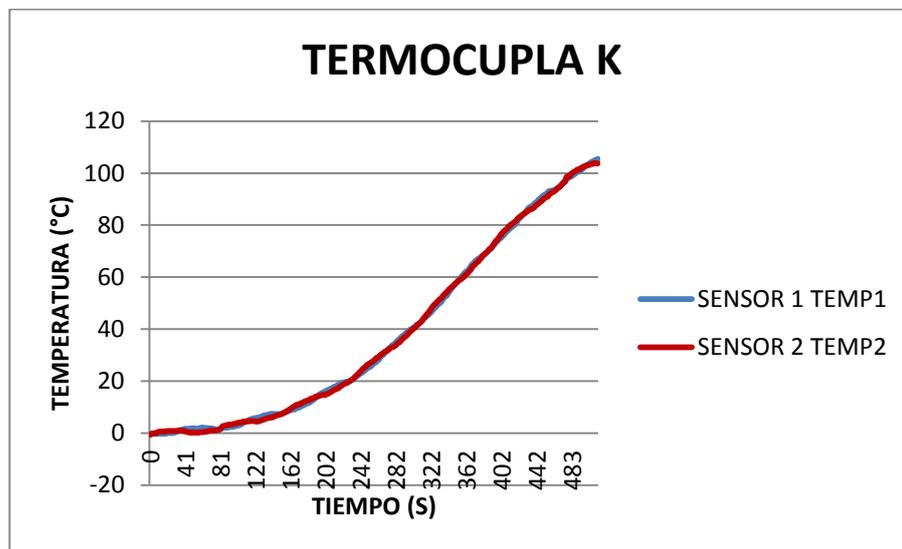


Figura 176 Temperatura vs tiempo termocupla K

La Figura 177 indica la prueba bilateral para dos muestras relacionadas, donde el intervalo de confianza para la diferencia entre las medias al 95% es de $] -0,135; 0,030[$ obteniendo datos resultantes como indica la Tabla 22.

Tabla 22
Prueba bilateral termocupla K

Diferencia	-0,052
t (Valor observado)	-1,244
t (Valor crítico)	1,965
GL	510
valor-p (bilateral)	0,214
alfa	0,05

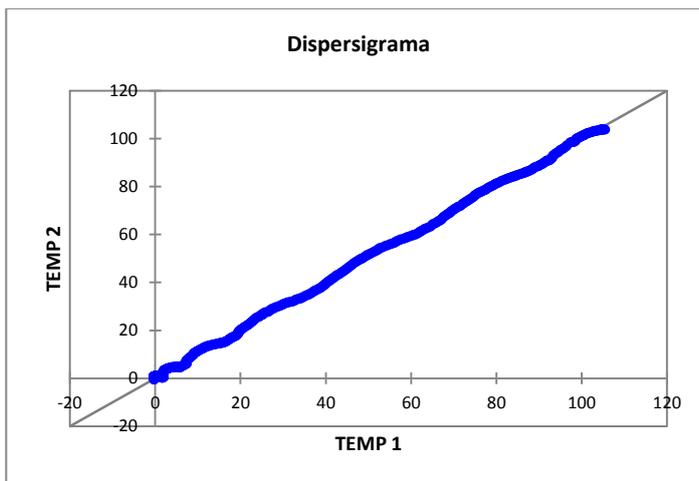


Figura 177 Prueba bilateral termocupla K

El análisis de Bland Altman tiene como intervalo de confianza de $]-1,919; 1,814[$ donde la Figura 178 muestra que el sensor Termocupla tipo K se encuentra dentro del rango del intervalo de confianza.

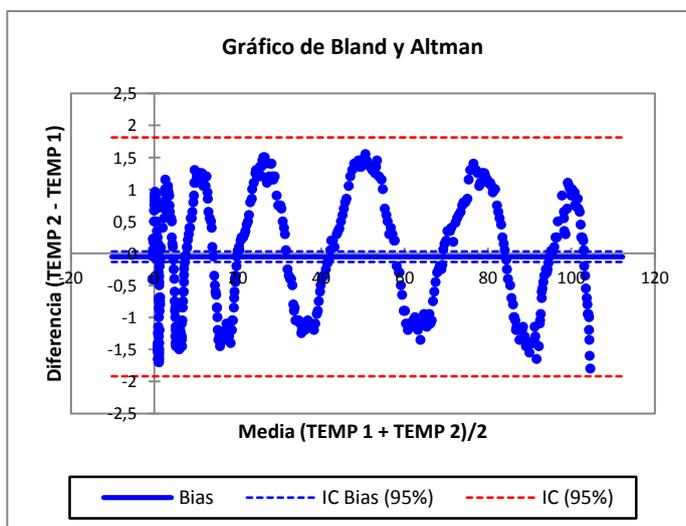


Figura 178 Bland altman termocupla K

PT100.

La Figura 179 muestra la gráfica de dos sensores PT100, obteniendo como resultado gráficas similares dentro de un mismo proceso de temperatura.

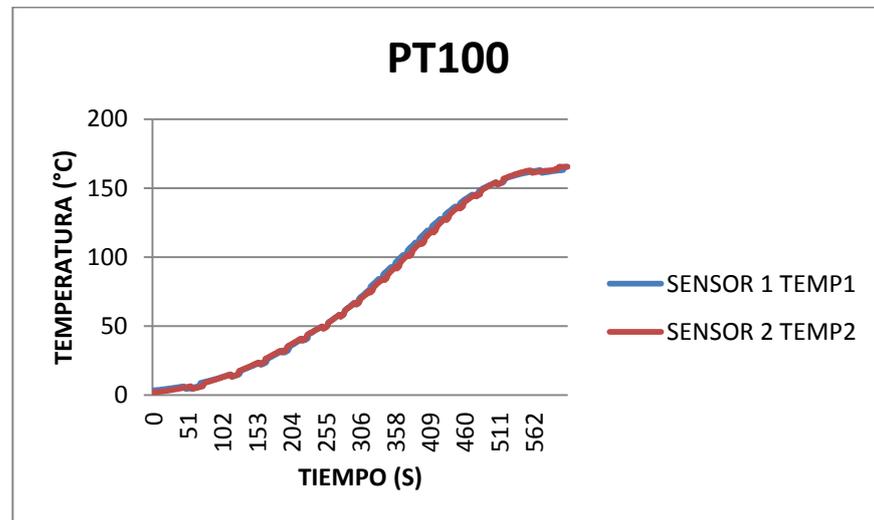


Figura 179 Temperatura vs tiempo PT100

La Figura 180 indica la prueba bilateral para dos muestras relacionadas, donde el intervalo de confianza para la diferencia entre las medias al 95% es de $]0,591; 0,874[$ obteniendo datos resultantes como indica la Tabla 23.

Tabla 23
Prueba bilateral PT100

Diferencia	0,732
t (Valor observado)	10,173
t (Valor crítico)	1,964
GL	611
valor-p (bilateral)	< 0,0001
alfa	0,05

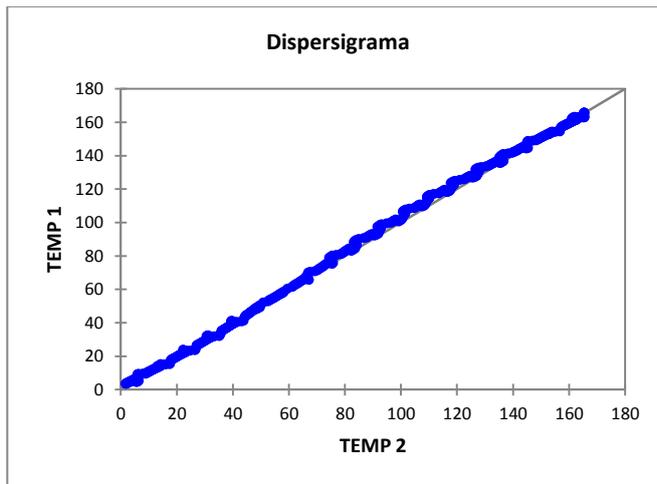


Figura 180 Prueba bilateral PT100

El análisis de Bland Altman tiene como intervalo de confianza de $]-2,758; 4,223[$ donde la Figura 181 muestra que el sensor PT100 en una temperatura de 30 °C se sub estima el intervalo de confianza mientras que en el rango de 80 a 130 °C se sobre estima el intervalo de confianza.

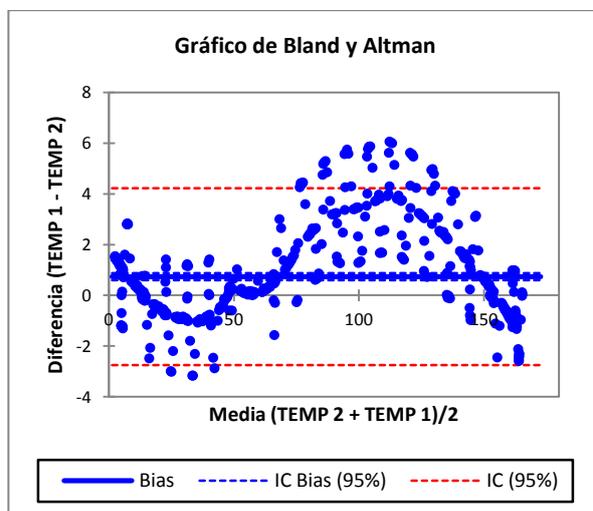


Figura 181 Bland altman PT100

CAPÍTULO IV

4 CONCLUSIONES Y RECOMENDACIONES.

4.1 Conclusiones.

- Se diseñó e implementó un prototipo de módulo para visualizar por medio de Touch Screen, PC y Dispositivos Móviles curvas características de sensores de temperatura en el rango de 0 a 200 °C, para el laboratorio de Circuitos Electrónicos de la Universidad de las Fuerzas Armadas ESPE Extensión Latacunga.
- Con la implementación del sistema de refrigeración se obtuvo una mayor área de enfriamiento de la que proporciona por sí solo el módulo Peltier, con esto se logró alcanzar temperaturas menores a la ambiente.
- Un proceso de temperatura es un proceso lento, el tiempo en estabilizarse es más prolongado, característica considerada para toma de mediciones, el rango, y los incrementos de temperaturas.
- El control utilizado es parte importante del proyecto, su funcionamiento correcto permite, realizar las mediciones de los sensores de prueba para la obtención de las curvas características en diferentes rangos de temperatura, comprendido entre 0° y 200°C.
- El sistema desarrollado es amigable en lo que se refiere a la adquisición, la forma de graficar los datos de las mediciones, la interface gráfica utiliza dispositivos que evitan realizar procesos complejos.

- El prototipo construido permite ayudar a la detección de daños mínimos o completos entre sensores del mismo tipo, con la ayuda de las curvas características generadas por éste, el reporte final dará a conocer si un sensor está en buenas condiciones o presenta algún defecto para seguir utilizándolo.
- MATLAB y Simulink son una excelente opción para el análisis y simulación de las señales por la facilidad con la que se pueden visualizar los resultados, por la sencillez de las instrucciones y por su variedad en las aplicaciones.
- En cuanto a la programación de la TFT el software Android Studio facilitó la creación del HMI con sus gráficas y cuadros de textos que reciben datos en tiempo real.
- Mediante LabVIEW y Excel se obtuvo de forma clara y precisa las curvas características de la respuesta de dos sensores de temperatura del mismo tipo.
- El tiempo de estabilidad del sistema es directamente proporcional a la temperatura deseada o Set Point, es decir, el sistema se tomará más tiempo en estabilizarse cuando controla temperaturas altas.
- Con las curvas características obtenidas en el sistema logramos conocer cuánto los sensores de temperatura a prueba se acercan a la curva característica de un elemento patrón.
- Al realizar la comparación entre dos sensores de temperatura del mismo tipo se tomó en cuenta su exactitud, precisión, resolución, parámetros que ayudan a conocer las variaciones en cada medición y así dar un diagnóstico.

- Mediante el sistema de análisis Bland Altman se calcula la diferencia de medida para los dos sensores de prueba. El gráfico obtenido muestra un diagrama de dispersión de las diferencias representativas frente a las medias de las dos mediciones.
- Se aplicó Bland Altman para el análisis de la Pt 100, donde la media de las variables se representa en el eje horizontal y las diferencias representa en el eje vertical, que mostró la cantidad de diferencia entre las dos medidas.

4.2 Recomendaciones.

- Es recomendable que el sistema que cubre al generador de temperatura sea hermético para evitar influencias ajenas en las mediciones.
- Es necesario dimensionar adecuadamente la niquelina para la generación de calor porque al tener una niquelina sobredimensionada el proceso de control será mucho más lento.
- Para un mejor análisis en las curvas características mostradas en la TFT se recomienda el uso de una pantalla mucho más amplia en la que se pueda visualizar las variaciones mínimas en la medición.
- Antes de comenzar a utilizar los diferentes tipos de sensor de temperatura investigar su rango de trabajo para evitar que sean expuestos a temperaturas fuera de sus especificaciones y así evadir desperfectos en el instrumento.
- Debido a las características del generador de temperatura es necesario que los sensores estén lo más cerca posible del sistema de enfriamiento y del sistema de calefacción para lograr tener una respuesta mucho más rápida en la toma de datos.
- Se recomienda el uso de un disipador muy grande para el dispositivo Peltier ya que éste genera demasiado calor en uno de sus lados, ayudándonos con esto a eliminar el calor adicional.
- Los sensores deben estar totalmente aislados de la carcasa que cubre al generador de temperatura para evitar interferencias.

- Es necesario que la niquelina tenga ventilación externa para evitar daños internos en su estructura y de la misma manera lograr un enfriamiento mucho más rápido.
- Tener en cuenta la configuración de pines de cada sensor en los que es necesario saber su polaridad y así evitar daños en éste.
- Se recomienda resetear la tarjeta arduino de ethernet antes de iniciar la comunicación, debido a que en el buffer se quedan almacenados datos de mediciones anteriores.
- Debemos revisar la IP que fue asignada a la tarjeta arduino ethernet porque no existe una IP estática asignada.

REFERENCIAS BIBLIOGRÁFICAS

- M. VIGLIANO, «Monografías.com,» s.f s.f s.f. [En línea]. Available:
- 1] <http://www.monografias.com/trabajos16/modulos-peltier/modulos-peltier.shtml>. [Último acceso: 3 Septiembre 2014].
- F. R. J. V. J. G. Y. M. S. A. AGUSTÍN BUNGE, «Heladera Refigerante Mediante Efecto Peltier Cooler,» San Juan El Precursor, San Isidro, 2005. Available: <http://descargas.cetronic.es/EstudioPeltier.pdf>. [Último acceso: 6 Octubre 2014].
- E. A. M. B. L. R. V. ANTONIO GOMEZ ROA., «Control De Temperatura Mediante Un Sistema De Enfriamiento Termoeléctrico Peltier,» de *Segundo Congreso Virtual, Microcontroladores y Sus Aplicaciones*, México, D.F. 2010. Available: <http://e-spacio.uned.es/fez/eserv/taee:congreso-2010-1003/S01A03.pdf>. [Último acceso: 15 Enero 2015].
- «¿Cuál es la función principal de una mufla?,» *QUIMINET*, 10 Octubre 2011. Available: <http://www.quiminet.com/articulos/cual-es-la-funcion-principal-de-una-mufla-2587039.htm>. [Último acceso: 10 Marzo 2015].
- P. GARCÍA, «¿Qué es el control PID?,» *Noticias Del Mercado*, Septiembre 2013. Available: <https://franklinlinkmx.wordpress.com/2013/09/05/que-es-el-control-pid/>. [Último acceso: 12 Abril 2015].
- F. M. GARCÍA, «Dpto. Informática y Automática,» ETSI de Informática, UNED, 11 Enero 2007. [En línea]. Available: <http://www.dia.uned.es/~fmorilla/MaterialDidactico/EI%20controlador%20PID.pdf>. [Último acceso: 15 Mayo 2015].
- S. D. TEMPERATURA, «medirtemperatura.com,» 13 Octubre 2006. [En línea]. Available: <http://nergiza.com/como-hacer-un-registrador-de-temperatura-con-arduino/>. [Último acceso: 16 Febrero 2015].

- J. QUIROZ, «Uso sensores de temperatura lineal LM35,» Attribution
8] Non-commercial, 2013. Available:
<http://www.ti.com/lit/ds/symlink/lm35.pdf>. [Último acceso: 18 Abril 2015].
- T. INSTRUMENTS, *Temperature Sensor Switch*, Texas, s.f. Available:
9] <http://www.ti.com/europe/downloads/2-%203-%204-Wire%20RTD%20Measurement.pdf>. [Último acceso: 25 Junio 2015].
- J. GARCIA, «Dos Tipos de Medidores de Temperatura: Termo-
10] resistencias y Termistores,» Departamento de Electrónica, Universidad
Técnica Federico Santa María, Valparaíso, 2000. Available:
<http://materias.fi.uba.ar/7609/material/S0307MedicionTemperatura2.pdf>
. [Último acceso: 8 Julio 2015].
- ARIAN, «Pt100, su operación, instalación y tablas,» *Control &*
11] *Instrumentación*, 2011. Available: <http://www.arian.cl/downloads/nt-004.pdf>. [Último acceso: 8 Mayo 2015].
- ARIAN, «Que són y como funcionan las termocuplas ?,» *Control &*
12] *Instrumentación*, 2011. Available: <http://www.arian.cl/downloads/nt-002.pdf>. [Último acceso: 25 Agosto 2015].
- D. GALLARDO, «Iniciándose en la plataforma Eclipse,»
13] *developerWorks*, 2012. Available:
<http://www.ibm.com/developerworks/ssa/library/os-ecov/>. [Último
acceso: 1 Febrero 2015].
- L. R. OJEDA, «MATLAB Conceptos Básicos y Programación,» de
14] *Tutorial*, 2007, p. 32. Available:
http://blog.espol.edu.ec/icm00794/files/2015/03/TUTORIAL_DE_MATLAB.pdf. [Último acceso: 28 Mayo 2015].
- Anónimo, «STlife.augmented,» 2014. [En línea]. Available:
15] <http://translate.google.com.ec/translate.www.st.com%2Fweb%2Fen%2Fcatalog%2Fm2FSS1577>. [Último acceso: 4 Septiembre 2014].

- J. VELASCO, «Think Big,» 21 Enero 2014. [En línea]. Available:
16] <http://blogthinkbig.com/accesibilidad-en-la-era-de-los-dispositivos-de-pantalla-tactil/>. [Último acceso: 17 Septiembre 2014].
- J. M. F. LUNA, «Tipos de dispositivos móviles,» *DECSAI*, nº 18071,
17] 2006. Available:
http://augusta.uao.edu.co/moodle/file.php/2896/pres_dispositivos_moviles-2011.pdf. [Último acceso: 27 Octubre 2014].
- Anónimo, «Potocolos de Comunicación, Interfaz Hombre Máquina,»
18] Quito, 2009, p. 35. Available:
<http://iaci.unq.edu.ar/materias/laboratorio2/HMI%5CIntroduccion%20HMI.pdf>. [Último acceso: 30 Noviembre 2014].
- C. A. M. B. I. C. A. L. L. N. HORACIO ARAÚZ, «Tecnología de
19] Redes,» [En línea]. Available:
<http://www.angelfire.com/planet/netstechnology/ethernet.htm>. [Último acceso: 3 Octubre 2014].
- «Ibersystems,» [En línea]. Available: <http://www.redeswifi.info/>. [Último
20] acceso: 6 Septiembre 2014].
- H. FREE, «NTC Thermostors, Radial Leded, Standar Precision,»
21] Vishay BCcomponents, 24 Agosto 2012. [En línea]. Available:
www.vishay.com/resistors-non-linear/curve-computation-list/. [Último acceso: 16 Octubre 2014].
- A. C. JESUS BAUSA, «SENSORES DE TEMPERATURA,»
22] SENSOVANT, 20 Julio 2013. [En línea]. Available: <http://server-die.alc.upv.es/asignaturas/LSED/2003-04/SensoresTemperatura.pdf>.
[Último acceso: 2 Enero 2015].
- «LM35CH,» TEXAS INSTRUMENTS, 8 Mayo 2011. [En línea].
23] Available: <http://pdf1.alldatasheet.com/datasheet-pdf/view/8878/NSC/LM35C.html>. [Último acceso: 3 Junio 2015].

- D. Humble, «MEGA MEX,» Purchase Terms · Sales Terms · Disclaimer,
24] [En línea]. Available: <http://www.megamex.com/span/incoloy.htm>.
[Último acceso: 12 Marzo 2015].
- J. L. Giodano, «Calentador Termoeléctrico,» 8 Enero 2005. [En línea].
25] Available: <http://www.profisica.cl/comofuncionan/como.php?id=15>.
[Último acceso: 25 Julio 2015].
- R. J. Dossat, «Instalaciones Frigoríficas,» MARCOMBO, 20 Agosto
26] 2001. [En línea]. Available:
<http://www.embraco.com/Default.aspx?tabid=262>. [Último acceso: 5
Febrero 2015].
- T. W. G. GENTIL, «SLIDESHARE,» [En línea]. Available:
27] <http://es.slideshare.net/waltergentil96/disipador-de-calor-16161946>.
[Último acceso: 12 MARZO 2015].
- C. Headquarters, «Maxim Integrated,» [En línea]. Available:
28] <http://www.maximintegrated.com/en/products/analog/sensors-and-sensor-interface/MAX31865.html>. [Último acceso: 18 Marzo 2015].
- «COLD-JUNCTION COMPENSATED THERMOCUPLE-TO-DIGITAL,»
29] MAXIM Innovation Delivered, 2 Marzo 2015. [En línea]. Available:
<http://datasheets.maximintegrated.com/en/ds/MAX31855.pdf>. [Último
acceso: 15 Agosto 2015].
- M. TECHNOLOGY, «Amplificador Operacional,» Mouser Electronics, 18
30] Febrero 2015. [En línea]. Available: <http://www.datasheet-pdf/74939/MICROCHIP/MCP6004.html>. [Último acceso: 2 Agosto
2015].
- K. KELLER, «Termistores,» Texas Instruments, 15 Octubre 2013. [En
31] línea]. Available: <http://www.ti.com.cn/cn/lit/an/slua621/slua621.pdf>.
[Último acceso: 10 Diciembre 2014].

ANEXOS



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN
CERTIFICACIÓN

Se certifica que el presente trabajo fue desarrollado por los señores **EDGAR JAVIER CAJAS OÑA** y **HERNÁN LEONARDO YANCHAPAXI ARMAS**.

Latacunga, 12 Noviembre 2015

CERTIFICAN:

ING. JOSÉ BUCHELI
DIRECTOR

ING. DAVID RIVAS
CODIRECTOR

APROBADO POR:

ING. FRANKLIN SILVA M.

**DIRECTOR DE LA CARRERA DE INGENIERÍA EN ELECTRÓNICA E
INSTRUMENTACIÓN**

CERTIFICADO POR:

DR. RODRIGO VACA
SECRETARIO ACADÉMICO

UNIDAD DE ADMISIÓN Y REGISTRO

