

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL

PROYECTO DE GRADO PARA LA OBTENCIÓN DEL
TÍTULO DE INGENIERÍA

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE ALERTA
PARA EL CIERRE DE PUERTAS DE LAS UNIDADES DEL
SISTEMA TROLEBÚS”

JUAN CARLOS FLORES PAREDES

SANGOLQUÍ – ECUADOR

- 2007-

CERTIFICACIÓN

Certificamos que el presente proyecto fue realizado en su totalidad por el señor Juan Carlos Flores Paredes, como requerimiento parcial a la obtención del título de INGENIERO ELECTRÓNICO.

Sr Ing. Hugo Ortiz
DIRECTOR

Sr Ing. Rodolfo Gordillo
CODIRECTOR

RESUMEN

El proyecto realizado comprende el diseño y la implementación de un sistema de alerta prototipo para el cierre de puertas de las unidades de la Unidad Operadora del Sistema Trolebús. El sistema es necesario por cuanto han ocurrido varios accidentes con los usuarios que tratan de ingresar a las unidades y se quedan presionados en las puertas por la falta de señalización oportuna que permita conocer a qué momento se van a cerrar las puertas del trolebús.

El sistema maneja las temporizaciones de las puertas correspondientes a cada hora de los 365 días del año. Un semáforo indica al conductor cuando detenerse y cuando avanzar, también se presenta el tiempo que las puertas permanecerán abiertas mediante un indicador visible a los usuarios dentro del andén y una alerta audible para ellos.

El desarrollo del prototipo involucra tres tipos de software: para el computador, desarrollado en Visual Basic 2005; para el controlador principal, desarrollado en lenguaje C (Dynamic C) y para el controlador de los displays, también desarrollado en lenguaje C (librerías de PCW H 3.2).

Finalmente, el sistema tiene la capacidad de modificar sus parámetros de funcionamiento en el momento que se crea oportuno, directamente manipulando el controlador principal, o mediante el manejo de archivos en el software desarrollado para el computador. Estos archivos proporcionan la capacidad de desarrollar la tabla de temporizaciones anual sin la necesidad de que exista conexión física entre el computador y el controlador principal.

DEDICATORIA

El esfuerzo realizado para la culminación de este proyecto lo dedico a mis padres, cuyo amor y dedicación me han ubicado en el sitio que ocupo en la vida; a mis hermanas, por su apoyo y compañía que me han ayudado en los momentos difíciles.

AGRADECIMIENTO

La realización de este proyecto fue posible gracias al apoyo brindado por la Unidad Operadora del Sistema Trolebús, especialmente el brindado por el Ing. Danilo Rodríguez, quien me brindó el apoyo técnico y logístico necesario para la culminación exitosa de este proyecto.

Agradezco a mi director, a mi codirector y a aquellos profesores del Departamento de Eléctrica y Electrónica de la ESPE que me han sabido transmitir sus conocimientos y las ganas de trabajar a lo largo de mi vida estudiantil.

Especialmente agradezco a Paul Padilla, Alejandro Mera y Fernando Mera, quienes me han brindado su apoyo desde el inicio del proyecto.

PRÓLOGO

Desde inicios del año 2006 se ha detectado un incremento considerable en la cantidad de usuarios que utilizan el Sistema Trolebús, razón por la cual existe una cantidad mayor de usuarios en las paradas del mismo. Como consecuencia, se han ocasionado varios accidentes con los usuarios que tratan de ingresar a las unidades y se quedan presionados en las puertas por la falta de señalización oportuna que permita conocer a qué momento se van a cerrar las puertas del trolebús. Existe un sistema interno de audio en cada trolebús, pero en las horas pico es insuficiente debido a que en la estación no se escucha la información de alerta del cierre de puertas.

Otro inconveniente detectado es que no existe un tiempo fijo en que las unidades deben permanecer estacionadas en cada una de las paradas, ocasionando retrasos en la operación y, en consecuencia, dando lugar a desfases que afectan el nivel de servicio y regularidad del Sistema Trolebús.

Debido a la problemática presentada, la Unidad Operadora del Sistema Trolebús necesita la provisión e instalación del equipo que permita conocer el tiempo de permanencia del trolebús en la estación.

La importancia del proyecto propuesto radica en dos aspectos: el primero es que se reducirá notablemente el número de accidentes producidos al cerrar las puertas del trolebús, mediante una señalización visual y sonora que indique sobre el cierre de puertas. El segundo aspecto es que se proporcionará mayor fluidez y regularidad en la operación del Sistema Trolebús, al proveer un tiempo fijo en que las unidades permanezcan estacionadas en las paradas.

El sistema deberá operar durante 20 horas diarias en un ambiente semi-industrial, el cual se evidencia debido a la interferencia electromagnética

producida por los motores de las puertas del andén y las líneas de alta tensión, propias del Sistema Trolebús; además de la operación del sistema en un ambiente expuesto a polvo y humedad. También debe proporcionarse fidelidad en la transmisión de datos mediante un cableado de aproximadamente 20 metros de longitud.

Finalmente, el sistema está provisto de dos tipos de software: el primero desarrollado para la manipulación de los parámetros de funcionamiento del sistema mediante un computador personal y, el segundo proporcionará una interfaz local cuidadosamente ordenada y estructurada, la cual permitirá la fácil operación del sistema por parte del encargado de la estación.

La documentación presentada está organizada en siete capítulos y la información está estructurada de la siguiente manera:

- El Capítulo 1 presenta una breve introducción acerca del sistema desarrollado, mostrando la descripción y los requerimientos del mismo.
- El Capítulo 2 muestra el fundamento teórico empleado para el diseño del sistema, de los estándares y los equipos utilizados en el diseño y desarrollo de éste.
- El Capítulo 3 presenta el diseño del hardware de cada componente principal del sistema, tomando como referencia el software disponible y los requerimientos de operación del sistema.
- El Capítulo 4 muestra cómo se desarrolló el software, basándose en los requerimientos de operación del sistema y el hardware seleccionado.
- El Capítulo 5 presenta la implementación del sistema, sus placas de circuito impreso y una guía para la instalación del hardware.
- El Capítulo 6 presenta las pruebas y resultados hechas al software y al hardware del sistema implementado.
- El Capítulo 7 muestra el presupuesto para el desarrollo del proyecto, tomando en cuenta que éste es un prototipo.

ÍNDICE DE CONTENIDO

CAPÍTULO 1: INTRODUCCIÓN	21
1.1. ANTECEDENTES	21
1.2. JUSTIFICACIÓN	22
1.3. ALCANCE	23
1.4. DESCRIPCIÓN GENERAL DEL SISTEMA	23
1.5. REQUERIMIENTOS DEL SISTEMA.....	24
1.5.1. Alimentación.....	24
1.5.2. Requerimientos de Hardware.....	24
1.5.3. Requerimientos de Software	25
CAPÍTULO 2: MARCO TEÓRICO	26
2.1. ESTÁNDAR RS-232	26
2.2. ESTÁNDAR RS-485	29
2.2.1. Par Trenzado	31
2.3. MICROCONTROLADOR PIC16F84A-20I/P	31
2.3.1. Características de la CPU RISC	31
2.3.2. Encapsulado 18-Lead PDIP (Plastic Dual In-line)	32
2.3.3. Características de los Periféricos	33
2.3.4. Características Especiales del Microcontrolador.....	33
2.3.5. Características Eléctricas Máximas Admisibles	34
2.4. MÓDULO MiniCom OP6810	35
2.4.1. Introducción.....	35
2.4.2. Descripción	35
2.4.3. Características	35
2.5. RELOJ DE TIEMPO REAL (RTC).....	36
2.6. MÓDULO GRABADOR / REPRODUCTOR DE AUDIO.....	39
2.6.1. Descripción	39
2.6.2. Características	40
2.6.3. Operación de Grabación	41
2.6.4. Operación de Reproducción.....	41
2.6.5. Estado de espera (Standby).....	41

2.7. AMPLIFICADOR DE AUDIO.....	41
2.7.1. Especificaciones	41
2.7.2. Características Técnicas	42
2.8. PROGRAMACIÓN ORIENTADA A OBJETOS (POO).....	42
2.8.1. Características de la POO.....	43
2.8.2. Plataforma de Desarrollo .NET	44
2.9. MANEJO DE ARCHIVOS	47
2.9.1. Característica “My” de Visual Basic 2005 Express.....	48
2.9.2. Manejo de Archivos Mediante “My”	48
2.9.3. Atributos y Niveles de Acceso con Archivos	49

CAPÍTULO 3: DISEÑO DEL HARDWARE..... 50

3.1. DIAGRAMA DE BLOQUES DEL SISTEMA.....	50
3.1.1. Controlador Principal.....	51
3.1.2. Entradas.....	53
3.1.3. Alerta Visual	54
3.1.4. Salidas	54
3.1.5. Alerta Audible.....	54
3.2. DIAGRAMAS Y ESPECIFICACIONES DE LOS ELEMENTOS DEL HARDWARE	55
3.2.1. Controlador Principal.....	55
3.2.2. Entradas.....	57
3.2.3. Alerta Visual	58
3.2.4. Salidas	61
3.2.5. Alerta Audible.....	62

CAPÍTULO 4: DESARROLLO DEL SOFTWARE 64

4.1. DESARROLLO DEL SOFTWARE PARA EL COMPUTADOR PERSONAL	64
4.1.1. Flujograma del Software Desarrollado en Visual Basic 2005.....	65
4.1.2. Tareas del Software Desarrollado en Visual Basic 2005.....	67
4.2. DESARROLLO DEL SOFTWARE PARA EL CONTROLADOR PRINCIPAL	70
4.2.1. Flujograma del Software Desarrollado en Dynamic C.....	71
4.2.2. Tareas del Software Desarrollado en Dynamic C	75

4.3. DESARROLLO DEL SOFTWARE PARA EL MICROCONTROLADOR PIC16F84A	80
4.3.1. Flujoograma del Software Desarrollado en Pic C (PCW H 3.2)	81

CAPÍTULO 5: IMPLEMENTACIÓN 83

5.1. PLACAS DE CIRCUITO IMPRESO	83
5.1.1. Placa de la Interfaz del Controlador Principal	83
5.1.2. Lista de Elementos de la Interfaz del Controlador Principal	84
5.1.3. Placa de los Módulos de Displays.....	85
5.1.4. Lista de Elementos de cada Módulo de Displays	88
5.2. INTEGRACIÓN DEL HARDWARE	90
5.2.1. Interfaz del Controlador Principal:	90
5.2.2. Sensor Infrarrojo:	92
5.2.3. Módulos de Displays:	93
5.2.4. Semáforo:.....	94
5.2.5. Módulo de Voz:	95
5.2.6. Amplificador de Audio:	97
5.3. IMPLEMENTACIÓN.....	98

CAPÍTULO 6: PRUEBAS Y RESULTADOS 101

6.1. CONDICIONES DE PRUEBA	101
6.2. PRUEBAS Y RESULTADOS	102
6.2.1. Pruebas de Hardware	102
6.2.2. Pruebas de Software.....	103

CAPÍTULO 7: ANÁLISIS ECONÓMICO 105

7.1. PRESUPUESTO	105
7.2. ANÁLISIS.....	108

CONCLUSIONES Y RECOMENDACIONES 109

CONCLUSIONES	109
RECOMENDACIONES	111

REFERENCIAS BIBLIOGRÁFICAS 112

ANEXOS 114

ANEXO 1: MANUAL DEL USUARIO.....	114
ANEXO 2: MEMORIA TÉCNICA.....	135
ANEXO 3: CÓDIGO DESARROLLADO PARA EL COMPUTADOR PERSONAL, EN MICROSOFT VISUAL BASIC 2005 EXPRESS EDITION.....	153
ANEXO 4: CÓDIGO DESARROLLADO PARA EL CONTROLADOR PRINCIPAL, EN DYNAMIC C.....	187
ANEXO 5: CÓDIGO DESARROLLADO PARA EL MICROCONTROLADOR PIC16F84A, EN PCW H 3.2.....	233
ANEXO 6: DIAGRAMA ESQUEMÁTICO.....	236

ÍNDICE DE TABLAS

Tabla 2.1. Niveles de Tensión en el Bus RS-232	27
Tabla 2.2. Conectores Empleados en la Conexión RS-232	28
Tabla 2.3. Registros de Datos (RTCxR) del Reloj de Tiempo Real.....	38
Tabla 2.4. Registro de Control (RTCCR dir = 0x01) del Reloj de Tiempo Real....	39
Tabla 2.5. Especificaciones Eléctricas	40
Tabla 5.1. Lista de Elementos de la Interfaz del Controlador Principal	85
Tabla 5.2. Lista de Elementos de cada Módulo de Displays	88
Tabla 6.1. Pruebas y Resultados de Hardware	102
Tabla 6.2. Pruebas y Resultados de Software	103
Tabla 7.1. Presupuesto para el Desarrollo del Prototipo	105

ÍNDICE DE FIGURAS

Figura 2.1. Conector DB-25.....	28
Figura 2.2. Conector DB-9.....	28
Figura 2.3. Voltaje Diferencial y Valores Lógicos	30
Figura 2.4. Conexión Típica de un Enlace RS-485	30
Figura 2.5. Encapsulado 18-Lead PDIP	32
Figura 2.6. Microcontrolador PIC16F84A	33
Figura 2.7. Interfaz Terminal Inteligente OP6810.....	35
Figura 2.8. Módulo Grabador/Reproductor de Audio.....	40
Figura 2.9. Amplificador de Audio	42
Figura 2.10. Modelo de Capas de la Plataforma .NET	45
Figura 2.11. Jerarquía de la Característica “My”	48
Figura 3.1. Diagrama de Bloques del Sistema	51
Figura 3.2. Diagrama de Bloques del Módulo OP6810	52
Figura 3.3. Diagrama Esquemático del Controlador Principal.....	56
Figura 3.4. Diagrama Esquemático de Conexiones del Sensor Infrarrojo.....	57
Figura 3.5. Diagrama de Bloques del Módulo de Displays.....	59
Figura 3.6. Microcontrolador PIC16F84A y Transceptor MAX485	60
Figura 3.7. Circuito de un Segmento del Display	61
Figura 3.8. Diagrama Esquemático de las Salidas del Controlador Principal.....	62
Figura 3.9. Diagrama Esquemático de la Alerta Audible	62
Figura 4.1. Flujograma del Software del Computador Personal (Parte 1)	65
Figura 4.2. Flujograma del Software del Computador Personal (Parte 2)	66
Figura 4.3. Flujograma del Software del Computador Personal (Parte 3)	67
Figura 4.4. Flujograma del Software del Controlador Principal (Parte 1)	72
Figura 4.5. Flujograma del Software del Controlador Principal (Parte 2)	73
Figura 4.6. Flujograma del Software del Controlador Principal (Parte 3)	74
Figura 4.7. Flujograma del Software del Controlador Principal (Parte 4)	75
Figura 4.8. Flujograma de los Menús del Controlador Principal.....	78

Figura 4.9. Flujograma del Software del PIC16F84A	81
Figura 5.1. PCB de la Interfaz del Controlador Principal	83
Figura 5.2. Ubicación de los Elementos en la PCB de la Interfaz del Controlador Principal.....	84
Figura 5.3. PCB de los Módulos de Displays	86
Figura 5.4. Ubicación de los Elementos en la PCB de los Módulos de Displays..	87
Figura 5.5. Diagrama Unifilar de Conexiones del Sistema	90
Figura 5.6. Placa de la Interfaz del Controlador Principal	91
Figura 5.7. Señales del Sensor Ubicadas en el Controlador de una Puerta del Andén.....	93
Figura 5.8. Conectores de Alimentación y RS-485 en la Placa del Módulo de Displays.....	94
Figura 5.9. Esquema de la Conexión del Semáforo.....	95
Figura 5.10. Conectores es en la Placa del Módulo de Voz.....	96
Figura 5.11. Acoplamiento entre el Módulo de Voz y el Amplificador de Audio ...	96
Figura 5.12. Conexiones del Amplificador de Audio.....	97
Figura 5.13. Fotografía de la Caja de Control	98
Figura 5.14. Fotografía del Módulo de Displays Instalado sobre la Segunda Puerta.....	99
Figura 5.15. Vista del Andén “El Calzado” con el Prototipo Instalado	99
Figura 5.16. Fotografía de uno de los Parlantes que Proporcionan la Alerta Audible	100
Figura 5.17. Fotografía del Semáforo ubicado Fuera del Andén, Visible al Conductor.....	100

ÍNDICE DE HOJAS TÉCNICAS

HOJA TÉCNICA N° 1. CONTROLADOR PRINCIPAL: MINICOMP OP6800.....	239
HOJA TÉCNICA N° 2. MICROCONTROLADOR_PIC16F84A.....	250
HOJA TÉCNICA N° 3. MÓDULO DE VOZ: A96020.....	259
HOJA TÉCNICA N° 4. TRANSCEPTOR RS-485: LTC485.....	261
HOJA TÉCNICA N° 5. AMPLIFICADOR DE AUDIO: TDA2616.....	266
HOJA TÉCNICA N° 6. DISPLAY DE 4 PULGADAS.....	272
HOJA TÉCNICA N° 7.FUENTE DE ALIMENTACIÓN.....	275
HOJA TÉCNICA N° 8. RELE DE 5V _{DC}	277
HOJA TÉCNICA N° 9. TRANSISTOR NPN DARLINGTON: TIP110.....	281
HOJA TÉCNICA N° 10. TRANSISTOR NPN: 2N3904.....	284
HOJA TÉCNICA N° 11. TRANSISTOR PNP: 2N3906.....	287

GLOSARIO

.NET	Plataforma de desarrollo que consta de una serie de servicios iguales en todos los lenguajes que mantienen la integridad con los desarrollos existentes y hace posible una interoperatividad entre los lenguajes.
ADO	ActiveX Data Object: Objeto de Datos ActiveX.
Arquitectura Harvard	Utiliza dispositivos de almacenamiento físicamente separados para las instrucciones y para los datos.
Back-up	Respaldo.
BCL	Base Classes Library: Librería de Clases Base.
C++	Extensión del lenguaje de programación C.
Canal Full-Duplex	Canal de transmisión de datos seriales donde los datos pueden viajar en ambos sentidos, simultáneamente.
Canal Half-Duplex	Canal de transmisión de datos seriales donde los datos pueden viajar en una u otra dirección, pero sólo durante un determinado periodo de tiempo; luego la línea debe ser conmutada antes que los datos puedan viajar en la otra dirección.
Canal Simplex	Canal de transmisión de datos seriales donde los datos siempre viajarán en una dirección.
CLR	Common Language Runtime: Lenguaje Común de Tiempo de Ejecución.
CMOS	Complementary Metal Oxide Semiconductor: Semiconductor de Metal Óxido Complementario.
Comunicación Asíncrona	Consiste en agregar marcadores dentro del flujo de bits para ayudar a seguir cada bit de datos.
Comunicación Síncrona	Los dispositivos de envío y recepción de la comunicación son sincronizados utilizando un reloj que cronometra con precisión el tiempo que separa cada bit.

CPU	Central Process Unit: Unidad Central de Proceso.
CTS	Common Type System: Sistema de Tipo Común.
DB-25	Conector de 25 pines, utilizado para comunicación serial o paralela.
DB-9	Conector de 9 pines, utilizado para comunicación serial.
DCE	Data Communications Equipment: Equipo de Comunicación de Datos.
Display	Indicador visual, pantalla de visualización en este caso es un indicador de 7 segmentos.
Distorsión Armónica Total	Cantidad de armónicos que un equipo (amplificador) introduce y que no estaban en la señal original.
DLL	Dynamic Link Library: Librería de Vínculos Dinámicos.
DTE	Data Terminal Equipment: Equipo Terminal de Datos.
Dynamic C	Lenguaje de programación basado en lenguaje C, utilizado para programar los microcontroladores Rabbit.
E/S	Entradas / Salidas.
EIA	Electrical Industries Association: Asociación de Industrias Eléctricas.
Handshaking	Conexión directa. Intercambio de señales, saludo, diálogo, iniciación del diálogo, etc.
HTML	HyperText Markup Language: Lenguaje de Hipertexto Enriquecido.
IC	Integrated Circuit: Circuito Integrado.
ICSP	In-Circuit Serial Programming: Programación Serial en Circuito.
ID	Identificador, sirve para identificar un nodo dentro de una red, en este caso de la red RS-485.
Infrarrojo	Banda de radiación electromagnética con una frecuencia más baja que la microondas y más alta que la óptica. Su longitud de onda es de 10^{-3} a 10^{-6} m
LED	Light Emitting Diode: Diodo Emisor de Luz.
LM317	IC Regulador de voltaje variable

MAX485	IC Transceptor RS-485.
MCLR	Master Clear (Reset): Limpador Maestro (Reiniciación). Señal del Microcontrolador PIC16F84A.
Memoria EEPROM	Electrically Erasable Programmable Read Only Memory: Memoria de Sólo Lectura Programable y Borrable Eléctricamente.
Memoria Flash	Memoria de tipo no volátil, grabable y borra eléctricamente.
Memoria SRAM	Static Random Access Memory: Memoria Estática de Acceso Aleatorio.
MiniCom OP6810	Controlador principal: interfaz terminal inteligente pequeña, de alto desempeño, programable en C y que además dispone de entradas y salidas construidas internamente
Oscilador RC	Oscilador cuya frecuencia se halla determinada por elementos de resistencia y capacidad.
OSI	Open System Interconnect: Sistema de Interconexión abierta.
OST	Oscillator Start-Up Timer: Oscilador del temporizador de Encendido. (Microcontrolador PIC16F84A).
PC	Personal Computer: Computador Personal.
PCB	Printed Circuit Board: Placa de Circuito Impreso.
PCW H 3.2	Software de programación en lenguaje C para PIC's.
PDIP	Plastic Dual In-line Package: Encapsulado Plástico de Doble Línea.
PIC	Familia de microcontroladores, fabricados por la empresa Microchip.
PIC16F84A	Microcontrolador de la familia PIC, con bus de 8 bits, 18 pines y memoria ampliada tipo Flash y EEPROM.
POO	Programación Orientada a Objetos.
POR	Power-On Reset: Reinicio de Encendido. (Microcontrolador PIC16F84A).
PWRT	Power-Up Timer: Temporizador del Encendido (Microcontrolador PIC16F84A).
Rabbit 2000®	Microprocesador de la empresa Rabbit Semiconductor, con

bus de 8 bits, 100 pines y memoria tipo Flash y SRAM.

Relación Señal / Ruido	Margen que hay entre el nivel de referencia (información significativa) y el ruido de fondo de un determinado sistema, medido en decibelios.
Reproducción de Frecuencia	Rango de frecuencia en el que puede operar eficientemente un amplificador de audio.
Resistores de Pull-Down	Resistencia que en un circuito fuerza un estado lógico bajo o "0", cuando el propio dispositivo no pueda generarlo.
Resistores de Pull-Up	Resistencia que en un circuito fuerza un estado lógico alto o "1", cuando el propio dispositivo no pueda generarlo.
RISC	Reduced Instruction Set Computer: Conjunto Reducido de Instrucciones de Computador.
RMS	Root Mean Square: Raíz Cuadrática Media.
RS-232C	(Recommended Standard - 232) Estándar recomendado para comunicación serial en distancias cortas (15m) y velocidades de transmisión de datos bajas (20Kbps). Puede trabajar en comunicación asíncrona o síncrona y tipos de canal simplex, half duplex o full duplex.
RS-485	(Recommended Standard - 485) Estándar recomendado para comunicación serial en ambiente industrial, con alta velocidad de transmisión de datos en largas distancias (35Mbps a 10m y 100Kbps hasta 1200m). Puede trabajar en comunicación half-duplex y multipunto.
RTC	Real Time Clock: Reloj de Tiempo Real.
Runtime	Tiempo de ejecución.
Sleep	Modo de bajo consumo de energía.
Sobrecarga	Se produce si la suma de la potencia de los aparatos que están conectados a un circuito es superior a la potencia para la cual está diseñado dicho circuito.
Standby	Estado de espera .
TDA2616	IC Amplificador de audio.
TTL	Transistor-Transistor Logic: Lógica Transistor-Transistor.

UOST	Unidad Operadora del Sistema Trolebús.
USART	Universal Synchronous Asynchronous Receiver Transmitter: Transmisor Receptor Sincrónico Asincrónico Universal.
VDD	Voltaje de alimentación de un IC (+VDC), VDD se utiliza generalmente para los IC tipo CMOS
VSS	Voltaje de alimentación de un IC (tierra), VSS se utiliza generalmente para los IC tipo CMOS
WDT	Watchdog Timer: Temporizador del Perro Guardián (PIC16F84A). Mecanismo de seguridad que provoca un reinicio del sistema en caso de que éste se haya bloqueado
XML	Extensible Markup Language: Lenguaje de Enriquecimiento Extensible.

CAPÍTULO 1

INTRODUCCIÓN

1.1. ANTECEDENTES

El Municipio del Distrito Metropolitano de Quito, buscando proporcionar un servicio de transporte masivo de calidad, moderno, seguro, ecológico y económico que satisfaga las necesidades de los habitantes del Distrito, inició, en el año 1990, un proceso de planificación realizando múltiples estudios de ingeniería de transporte, los cuales establecieron que el 75% del tráfico de pasajeros urbanos, que utilizan los sistemas públicos, se presentaba en el eje Villaflora-Iñaquito, y que el 25% restante se distribuía en los demás ejes paralelos y algunos transversales. Estas circunstancias desembocaron en la creación de la Unidad Operadora del Sistema Trolebús (UOST), la cual actualmente transporta un promedio de 260 000 pasajeros diarios; cuenta con 39 paradas individuales, 8 de doble sentido y una de integración, las mismas que están a una distancia promedio de 400m, cubriendo una distancia total de aproximadamente 16Km, desde la estación La Y, al norte, hasta la estación Morán Valverde, al sur. Además, la UOST cuenta con tres terminales de transferencia que son abastecidas por 5 rutas alimentadoras en el sur, 6 en el norte, 1 de la Ecovía y 4 en Morán Valverde, con el fin de servir a los lugares periféricos de la ciudad.

Desde inicios del año 2006 se ha detectado un incremento considerable en la cantidad de usuarios que utilizan el Sistema Trolebús, por esta causa existe una cantidad mayor de usuarios en las paradas del mismo.

Debido a esto han ocurrido varios accidentes con los usuarios que tratan de ingresar a las unidades y se quedan presionados en las puertas por la falta de señalización oportuna, dentro del andén de la estación, que permita conocer a

qué momento se van a cerrar las puertas del trolebús. Existe un sistema interno de audio en cada trolebús, pero en las horas pico es insuficiente porque en la estación no se escucha la información de alerta del cierre de puertas.

Otro inconveniente que se está presentando es que no existe un tiempo fijo en que las unidades deben permanecer estacionadas en cada una de las paradas, ocasionando retrasos en la operación y, en consecuencia, dan lugar a desfases que afectan el nivel de servicio y regularidad del Sistema Trolebús.

1.2. JUSTIFICACIÓN

Debido a la problemática presentada, la Unidad Operadora del Sistema Trolebús (UOST) necesita la provisión e instalación del equipo que permita conocer el tiempo de permanencia del trolebús en la estación.

La **importancia** del proyecto propuesto radica en dos aspectos: el primero es que se reducirá notablemente el número de accidentes producidos al cerrar las puertas del trolebús, mediante una señalización visual y sonora que indique sobre el cierre de puertas. El segundo aspecto es que se proporcionará mayor fluidez y regularidad en la operación del Sistema Trolebús, al proveer un tiempo fijo en que las unidades permanezcan estacionadas en las paradas.

La realización de este proyecto se **justifica** en el hecho de que el sistema deberá operar durante 20 horas diarias en un ambiente semi-industrial, el cual se evidencia debido a la interferencia electromagnética producida por los motores de las puertas del andén y las líneas de alta tensión, propias del Sistema Trolebús, además de la operación del sistema en un ambiente expuesto a polvo y humedad. Adicionalmente, debe proporcionarse fidelidad en la transmisión de datos mediante un cableado de aproximadamente 20 metros de longitud. Finalmente, el sistema estará provisto de una interfaz local cuidadosamente ordenada y estructurada, la cual permitirá la fácil operación del sistema por parte del encargado de la estación.

1.3. ALCANCE

Este proyecto plantea satisfacer una necesidad presente en la UOST. Dicha necesidad será cubierta mediante la ejecución del diseño e implementación del prototipo de un sistema electrónico que informe la apertura y cierre de las puertas del trolebús.

El **alcance** del proyecto plantea que dicho prototipo será instalado en **una** estación del Sistema Trolebús, la cual será dispuesta por la administración de la UOST.

Adicionalmente, se tiene una **proyección futura** para que el prototipo sea instalado en 30 estaciones de la UOST, de así convenir a los intereses de dicha empresa.

1.4. DESCRIPCIÓN GENERAL DEL SISTEMA

El Sistema de Alerta para el Cierre de Puertas de las Unidades del Sistema Trolebús esta desarrollado en base a un sistema de control que cumple las siguientes características:

- Detección de la presencia del trolebús en la parada.
- Indicador visual de dos dígitos, visible a los usuarios del Sistema Trolebús, que permite visualización del tiempo que las puertas de acceso al trolebús permanecerán abiertas.
- Señal de alerta audible para los usuarios dentro del andén, para indicar tanto la apertura, como el cierre (manual, por parte del conductor) de las puertas del trolebús.
- Señal de alerta visual para el conductor del trolebús, para mostrar que ha transcurrido su tiempo de estadía en la parada y que puede cerrar las puertas y continuar su recorrido.
- Red y Protocolo de comunicación basados en el estándar industrial RS-485, para comunicar el controlador principal con los indicadores visuales en el andén.
- Tabla de Tiempos programable, para la operación autónoma del sistema durante todo un año, la cual almacena la temporización para la apertura y el

cierre de las puertas del trolebús de cada hora del día, además del tipo de día (ordinario, fin de semana, feriado o personalizado).

- Coordinación del funcionamiento del sistema con un Reloj de Tiempo Real.
- Controlador principal del sistema: módulo MiniCom OP6810.
- Software del controlador principal desarrollado en lenguaje C (Dynamic C 9.25), con características de operación simple y directa por parte del encargado del andén.
- Manejo de un sistema de archivos que permite respaldar la información correspondiente a la Tabla de Tiempo anual de la estación del Sistema Trolebús.
- Conexión entre el controlador principal y el computador para operaciones de programación y respaldo mediante el estándar RS-232.
- Software administrativo del sistema desarrollado bajo Microsoft Visual Basic 2005 Express Edition, el cual permite tanto el almacenamiento, como el respaldo de datos del controlador principal.
- Sistema robusto, provisto de inmunidad frente a las interferencias propias del Sistema Trolebús y capaz de trabajar las 24 horas del día.

1.5. REQUERIMIENTOS DEL SISTEMA

1.5.1. Alimentación

El sistema tiene los siguientes requerimientos en lo referente a su alimentación:

- Voltaje de alimentación: 100 - 120V_{AC} / 50 - 60Hz
- Consumo de corriente: 3.5 - 4.5A

1.5.2. Requerimientos de Hardware

El hardware necesario para la correcta conexión del módulo controlador con el computador son los siguientes:

- Procesador INTEL Pentium III, AMD Athlon XP o superior
- 256 MB de memoria RAM
- Al menos 100 MB de espacio libre en el disco duro
- Puerto Serial con el estándar RS-232C y con conector DB-9 tipo macho.

-
- Cable de datos serial (recto), con conectores DB-9 tipo hembra en ambos extremos.
 - Cable de programación diseñado para el microprocesador Rabbit 2000®.
 - Programador de microcontroladores PIC.

1.5.3. Requerimientos de Software

El software necesario para el computador personal debe poseer las siguientes características:

- Sistema Operativo Windows 98 SE o superior.
- Microsoft .NET Framework 2.0
- Windows Installer 3.1
- Dynamic C 9.25
- DC 9.25 RFU (Para programar el Controlador Principal).
- WINPIC800, IC-PROG o cualquier software desarrollado para grabar microcontroladores PIC.

CAPÍTULO 2

MARCO TEÓRICO

2.1. ESTÁNDAR RS-232

El enlace RS-232C, es denominado así por la norma Americana EIA, cuyo equivalente europeo es V.24. originalmente fue previsto para la comunicación entre una terminal de datos (DTE) y un MODEM (DCE), sin embargo con el tiempo ha experimentado un gran número de cambios aplicados principalmente a enlaces punto a punto entre terminales de datos.

El RS-232 fue originalmente adoptado en 1960 por la EIA. El estándar evolucionó a través de los años y en 1969 la tercera revisión, el RS-232C, fue el estándar elegido por los fabricantes de computadoras personales compatibles con IBM. En 1987 se adoptó la cuarta revisión, el RS-232D, también conocida como EIA-232D. En esta nueva revisión se agregaron 3 líneas de prueba.

La norma básica se ocupa del aspecto físico de la conexión, indicando los tipos de conectores, niveles de la señal y señales de protocolo (señales de “handshaking¹”). RS-232 está diseñado para **distancias cortas**, de unos 15 metros o menos, y para una **velocidad de comunicación baja**, de no más de 20 Kbps. A pesar de ello, muchas veces se utiliza a mayores velocidades con un resultado aceptable. La interfaz puede trabajar en comunicación asíncrona² o síncrona³ y tipos de canal simplex, half duplex o full duplex. En un canal **simplex** los datos siempre viajarán en una dirección, por ejemplo desde DCE a DTE. En un canal **half duplex**, los datos pueden viajar en una u otra dirección, pero sólo durante un determinado periodo de tiempo; luego la línea debe ser conmutada

¹ Handshaking: Conexión directa. Intercambio de señales, saludo, diálogo, iniciación del diálogo, etc.

² Comunicación Asíncrona: Consiste en agregar marcadores dentro del flujo de bits para ayudar a seguir cada bit de datos.

³ Comunicación Síncrona: Los dispositivos de envío y recepción de la comunicación son sincronizados utilizando un reloj que cronometra con precisión el tiempo que separa cada bit.

antes que los datos puedan viajar en la otra dirección. En un canal **full duplex**, los datos pueden viajar en ambos sentidos simultáneamente. Las líneas de handshaking del RS-232 se usan para resolver los problemas asociados con este modo de operación, tal como en qué dirección los datos deben viajar en un instante determinado. Además de esto, si un dispositivo de los que están conectados a una interfase RS-232 procesa los datos a una velocidad menor de la que los recibe, deben conectarse las líneas de handshaking que permiten realizar un control de flujo tal que al dispositivo más lento le de tiempo de procesar la información. Las líneas de handshaking que permiten hacer este control de flujo son las líneas RTS (Solicitud de envío, Tabla 2.2.) y CTS (Autorización de envío, Tabla 2.2.). Los diseñadores del estándar no concibieron estas líneas para que funcionen de este modo, pero dada su utilidad cada en interfase posterior se incluye este modo de uso.

La denominación V.24 de la norma equivalente europea viene del hecho de que los niveles de tensión utilizados son de +12V y -12V (0 y 1 lógicos, respectivamente). En realidad existe una banda de tolerancia para determinar los niveles lógicos de la línea como se muestra en la Tabla 2.1.

Tabla 2.1. Niveles de Tensión en el Bus RS-232

	0 LÓGICO	1 LÓGICO
SALIDAS	+5V a +15V	-5V a -15V
ENTRADAS	+3V a +15V	-3V a -15V

La mayoría de los equipos que implementan puertos RS-232C utilizan un conector DB-25 (Figura 2.1) aún cuando la documentación original del estándar no especifica un conector en especial; la mayoría de las computadoras comenzaron a utilizar el conector DB-9 (Figura 2.2.) dado que 9 son los pines que se requieren para la comunicación asíncrona. Es necesario notar que el documento especifica la cantidad de terminales y su asignación, 20 para las señales, 3 reservados y 2 sin uso. Normalmente el conector macho es en el lado

de la terminal y el conector hembra es en el de comunicaciones, aún si este no es el caso común, como se muestra en la Tabla 2.2.

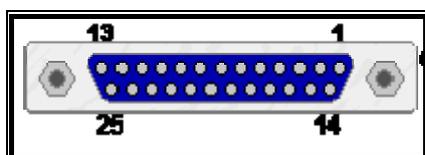


Figura 2.1. Conector DB-25

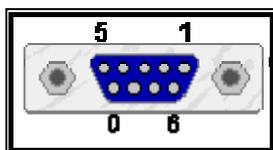


Figura 2.2. Conector DB-9

Tabla 2.2. Conectores Empleados en la Conexión RS-232

Número de pin DB-25	Número de pin DB-9	Nombre de la señal.
1	-	Protección a tierra
7	5	Tierra
2	3	Transmisión de datos (TX)
3	2	Recepción de datos (RX)
4	7	Solicitud de envío (RTS)
5	8	Autorización de envío (CTS)
6	6	Datos listos para enviar (DSR)
20	4	Terminal de datos lista (DTR)
22	9	Detector de llamada
8	1	Detección de portadora (DCD)
21	-	Detección de calidad de señal
23	-	Selector de velocidad de señal de datos.
24	-	Cronómetro de la señal emisora (DTE)

15	-	Cronómetro de la señal de transmisión del transmisor (DCE)
17	-	Cronómetro de la señal del receptor
14	-	TX secundario
16	-	RX secundario
19	-	RTS Secundario
13	-	CTS secundario
12	-	Detector de portadora (CD) secundario
9	-	Reservado Positivo de prueba
10	-	Reservado Negativo de prueba
11	-	Sin conexión
18	-	Sin conexión
25	-	Sin conexión

En algunos catálogos de productos industriales se designa a veces como canal RS-232 un enlace que dispone solo de las líneas RX, TX y Tierra, es decir, ningún tipo de señales de handshaking.

2.2. ESTÁNDAR RS-485

RS-485 se refiere a una especificación eléctrica de un enlace a dos cables de una capa física del modelo OSI, con características de comunicación serial, half-duplex, multipunto.

El estándar especifica una señalización diferencial para la transmisión de datos, es decir mediante la diferencia de voltaje entre los terminales del receptor y transmisor se determina un nivel lógico, como se muestra en la Figura 2.3. Dicha diferencia de potencial debe ser de al menos de 0.2 V para una operación válida, sin embargo se puede aplicar tensiones que van desde +12 V hasta -7 V.

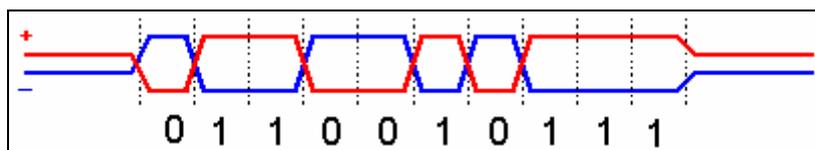


Figura 2.3. Voltaje Diferencial y Valores Lógicos

El estándar RS-485 simplemente especifica características eléctricas del receptor/transmisor y no indica o recomienda un protocolo de datos. Las ventajas de un enlace RS-485 son la configuración de redes locales de bajo costo además de en enlaces multipunto, ofrece conexiones de alta velocidad para transmisión de datos (35Mbps a 10m y 100Kbps hasta 1200m); debido a que utiliza líneas diferenciales balanceadas sobre pares de cable trenzado.

Como se muestra en la Figura 2.4., en una conexión típica multipunto RS-485, se utilizan resistores de terminación en los dos extremos de la línea de transmisión, los cuales ayudan a reducir la sensibilidad al ruido al igual que los resistores de pull-up y pull-down. Sin los resistores de terminación las reflexiones provocadas por las transiciones de alta velocidad pueden causar múltiples transiciones dentro de los datos llegando a corromperlos. El valor de dichos resistores debe ser similar a la impedancia de la línea de transmisión (típicamente 120Ω para cables trenzados).

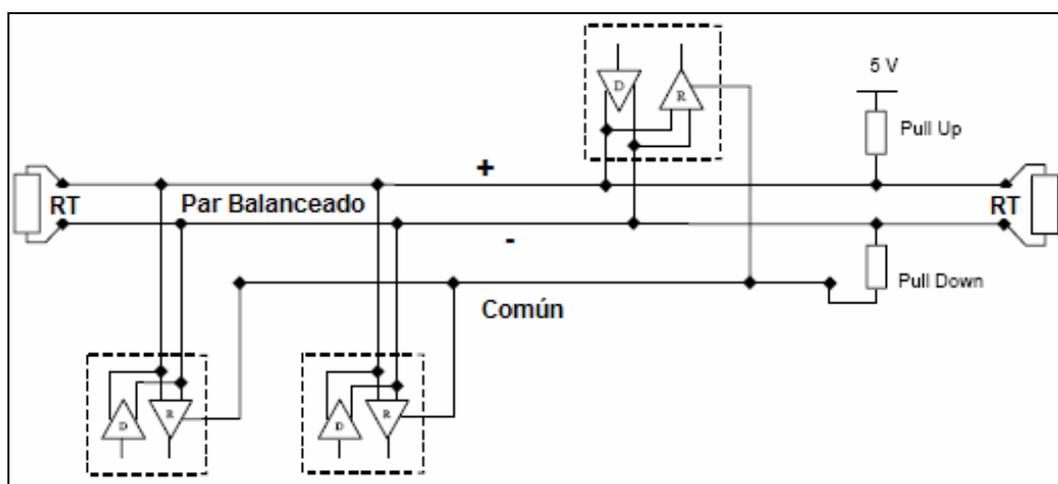


Figura 2.4. Conexión Típica de un Enlace RS-485

2.2.1. Par Trenzado

El medio de transmisión más antiguo es el par trenzado que aún es muy utilizado. Consiste en dos hilos de cobre aislados, de 1mm de diámetro aproximadamente, los conductores se trenzan en forma helicoidal para reducir la interferencia eléctrica con respecto a los pares cercanos que se encuentran a su alrededor. Dos cables paralelos constituyen una antena simple, mientras que si se trenzan no.

Su aplicación más común es el sistema telefónico. Con estos cables se pueden recorrer varios kilómetros sin tener que amplificar las señales, aunque si son necesarios repetidores para distancias más largas. Cuando hay muchos pares trenzados en paralelo, recorriendo una distancia considerable, éstos se agrupan y se cubren con una malla protectora. Los pares dentro de estos grupos podrían sufrir interferencias mutuas si no estuviesen trenzados.

Los pares trenzados pueden usarse para transmisión analógica o digital, y su ancho de banda depende del trenzado del cable y de la distancia que recorre. En muchos casos, pueden obtenerse transmisiones de varios Mbps sobre distancias de pocos kilómetros.

2.3. MICROCONTROLADOR PIC16F84A-20I/P

El PIC16F84A es un microcontrolador de la familia PIC, fabricada por la empresa Microchip.

2.3.1. Características de la CPU RISC

La CPU, o algunas veces simplemente procesador, es el componente en un computador digital que interpreta las instrucciones y procesa los datos contenidos en los programas. RISC constituye un conjunto pequeño de instrucciones simples que necesitan poco tiempo para ejecutarse, las cuales permiten la programación de la CPU.

Las principales características de la CPU RISC del PIC16F84A son las siguientes:

- La arquitectura de la CPU es del tipo Harvard⁴.
- Conjunto de 35 instrucciones de una sola palabra.
- Todas las instrucciones duran un ciclo de máquina, excepto las de salto que duran dos.
- Velocidad de operación:
 - DC – 20MHz, para la frecuencia del reloj de entrada.
 - DC – 200ns, para la duración del ciclo de máquina.
- Frecuencia máxima de funcionamiento 20 MHz (PIC16F84A-20).
- Memoria de programa tipo Flash de 1024 posiciones.
- Memoria de datos RAM de 68 bytes.
- Memoria EEPEROM de datos de 64 bytes.
- Longitud de las instrucciones de 14 bits.
- Longitud de los datos de 8 bits (1 byte).
- Dispone de 4 fuentes de interrupción, las cuales pueden ser habilitadas o deshabilitadas, independientemente, por software:
 - Externa por el pin RB0/INT.
 - Por desbordamiento del Timer 0.
 - Por cambio en las líneas PORTB <7:4>.
 - Por finalización de escritura de la memoria EEPROM de datos.

2.3.2. Encapsulado 18-Lead PDIP (Plastic Dual In-line)

Este tipo de encapsulado es el apropiado para ser utilizado en las placas del circuito impreso. La Figura 2.5. muestra la forma de este encapsulado.

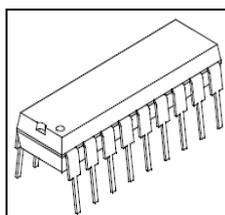


Figura 2.5. Encapsulado 18-Lead PDIP

⁴ Arquitectura Harvard: Utiliza dispositivos de almacenamiento físicamente separados para las instrucciones y para los datos.

Adicionalmente, la Figura 2.6. muestra la distribución de pines del PIC16F84A, correspondiente al encapsulado PDIP.

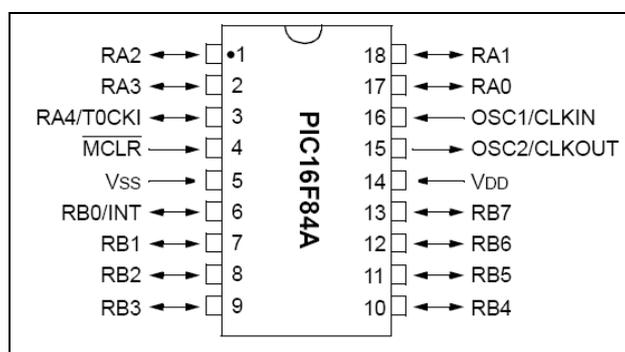


Figura 2.6. Microcontrolador PIC16F84A

2.3.3. Características de los Periféricos

- Dispone de 13 líneas de entrada / salida con control individual de dirección (5 pines el puerto A y 8 pines el puerto B).
- Alta capacidad de corriente por terminal. Proporciona suficiente corriente para gobernar un LED:
 - Consume 25mA por pin cuando está en nivel bajo.
 - Proporciona 20mA por pin cuando está en nivel alto.
- Dispone de un Temporizador / Contador de 8 bits (TMR0) con división de frecuencia programable.

2.3.4. Características Especiales del Microcontrolador

- La memoria Flash de programa admite hasta 1000 ciclos de borrado y escritura.
- La memoria EEPROM de datos admite hasta 1 millón de ciclos de borrado y escritura.
- Garantiza una retención de datos para la memoria EEPROM de datos para un período superior a 40 años.
- Se puede programar el circuito vía serie mediante dos pines (ICSP).
- POR: se genera cuando se detecta un rizo en el voltaje de alimentación (1.2V - 1.7V). PWRT: se mantiene reseteado el circuito por un tiempo determinado (72ms). OST: proporciona un retardo de 1024 ciclos antes que finalice PWRT.

- Dispone de un temporizador Watchdog⁵ (WDT) con su propio oscilador RC⁶ para un funcionamiento fiable.
- Protección del código de programa mediante la activación de un bit de protección.
- Modo de bajo consumo de energía: SLEEP.

2.3.5. Características Eléctricas Máximas Admisibles

- Tensión de cualquier pin respecto de V_{SS}
(excepto V_{DD} , MCLR y RA4)..... -0.3V a ($V_{DD} + 0.3V$)
- Tensión en V_{DD} respecto de V_{SS} -0.3V a 7.5V
- Tensión en MCLR respecto de V_{SS} -0.3V a 14V
- Tensión en RA4 respecto de V_{SS} -0.3V a 8.5V
- Potencia de disipación total..... 800mW
- Máxima corriente por el pin V_{SS} 150mA
- Máxima corriente por el pin V_{DD} 100mA
- Máxima corriente de salida en bajo por
cualquier pin E/S..... 25mA
- Máxima corriente de salida en alto por
cualquier pin E/S..... 20mA
- Máxima corriente de salida en bajo por
el conjunto del puerto A..... 80mA
- Máxima corriente de salida en alto por
el conjunto del puerto A..... 50mA
- Máxima corriente de salida en bajo por
el conjunto del puerto B..... 150mA
- Máxima corriente de salida en alto por
el conjunto del puerto B..... 100mA

⁵ Watchdog: Mecanismo de seguridad que provoca un reinicio del sistema en caso de que éste se haya bloqueado.

⁶ Oscilador RC: Oscilador basado en un circuito Resistencia-Capacitor, tomando como referencia la constante de carga del capacitor.

2.4. MÓDULO MiniCom OP6810

2.4.1. Introducción

El módulo a utilizar como interfaz con el usuario, y como controlador del sistema es la interfaz terminal inteligente OP6810, la cual es pequeña, de alto desempeño y programable en C, y que además dispone de entradas y salidas construidas internamente. Un microprocesador Rabbit 2000, operando a 22.1 MHz, proporciona un rápido procesamiento de datos.

La Figura 2.7. muestra una fotografía del módulo a utilizar como controlador principal del sistema.



Figura 2.7. Interfaz Terminal Inteligente OP6810

2.4.2. Descripción

La interfaz terminal inteligente OP6810 incorpora el potente microprocesador Rabbit 2000, memoria flash, RAM estática, puertos de entrada/salida digitales y puertos seriales RS-232 / RS-485.

2.4.3. Características

Las características técnicas más importantes del módulo OP6810 son las siguientes:

- Display gráfico de 122 × 32 píxeles.
- Teclado con 7 teclas.

- 7 LEDs.
- 24 E/S digitales: 13 entradas digitales filtradas, y 11 salidas con drenaje (sinking) de alta corriente (7 salidas con LEDs indicadores, y 4 salidas digitales de alta corriente con protección de transitorios para manejar cargas inductivas).
- Microprocesador Rabbit 2000® operando a 22.1MHz.
- RAM estática de 128KB, y memoria flash estándar de 256KB.
- Cuatro puertos seriales (dos RS-232 o un RS-232 con RTS/CTS, un RS-485, y un puerto de programación compatible con CMOS).
- Reloj de tiempo real respaldable mediante batería, incluye punto de conexión para batería externa.
- Watchdog.
- Generador de Reset.
- Cumple con las especificaciones NEMA 4 para el montaje del panel frontal a prueba de agua (y polvo).
- Cumple con la certificación **CE** referente a inmunidad y emisiones electromagnéticas.

2.5. RELOJ DE TIEMPO REAL (RTC)

Un RTC es un reloj de computadora (la mayoría de veces en forma de un chip de circuito integrado) que almacena el valor del tiempo actual, incluso cuando el computador está apagado. Esto es utilizado en muchos tipos de computadores, y está presente en todos los computadores personales modernos. Los RTCs también están presentes en muchos sistemas embebidos (embedded systems), como el utilizado en el desarrollo de este proyecto: el módulo MiniCom OP6810.

Los relojes de tiempo real operan con una batería especial que no está conectada a la fuente de alimentación normal. En contraste, los relojes que no son de tiempo real no funcionan cuando el computador está apagado.

En el caso del módulo MiniCom OP6810, el reloj de fecha/hora (RTC) es un contador (tipo rizo: ripple) de 48 bits que es manejado por un oscilador de 32.768KHz. El RTC es un contador tipo rizo modificado, compuesto por seis

contadores separados de 8 bits. Los acarrees son alimentados en todos los seis contadores de 8 bits al mismo tiempo y entonces se tiene el rizo para 8 bits. El tiempo para que tenga lugar este rizo es de unos pocos nanosegundos por bit, y ciertamente no debe exceder 200ns para todos los 8 bits, incluso al operar a bajo voltaje.

Los 48 bits son suficientes para contar hasta 272 años a la frecuencia de reloj de 32KHz. Por convención, se toma como tiempo cero las 12 AM del 1 de enero de 1980. El software de Rabbit Semiconductor ignora el bit de mayor orden, dando al contador una capacidad de 136 años desde el 1 de enero de 1980. Para leer el valor del contador, el valor es primero transferido a un registro de retención de 6 bytes. Entonces los bytes individuales pueden ser leídos de los registros de retención. Para realizar la transferencia, cualquier bit de datos se escribe en RTC0R, el primer registro de retención. El contador puede entonces ser leído como seis bytes de RTC0R hasta RTC5R. El contador y el oscilador de 32KHz son energizados por un pin de energía separado que puede estar suministrado con potencia mientras el resto del chip está apagado. Este diseño hace posible el apoyo de una batería. Puesto que el procesador opera en un reloj diferente que el RTC, existe la posibilidad de desarrollar una transferencia a los registro de retención mientras está tomando lugar un acarreo, resultando en una información incorrecta. Para prevenir esto, el procesador debe leer el reloj dos veces y asegurar que el valor es el mismo en ambas lecturas.

Si el procesador está operando a 32KHz, el procedimiento de lectura del reloj debe ser modificado puesto que varias cuentas de reloj tendrían lugar en el tiempo necesario por el cronometraje lento del procesador para leer el reloj. Una modificación apropiada sería ignorar los bytes más bajos y sólo leer los 5 bytes más altos, que se cuenta una vez cada 256 pulsos de reloj o cada 1/128 segundos. Si la lectura no fue desarrollada en este periodo, pueden ignorarse los bits de menor orden.

Los registros del RTC no pueden fijarse por una operación de escritura, pero pueden ser borrados y contar individualmente, o por subconjuntos. En esta forma,

cualquier registro o el contador entero de 48 bits pueden fijarse en cualquier valor con no más de 256 pasos. Si el cristal de 32KHz no está instalado y el pin de entrada está aterrizado, no tendrá lugar el conteo y los seis registros pueden ser utilizados como una pequeña memoria respaldada por batería. Normalmente esto no sería muy productivo puesto que la circuitería necesaria para proporcionar el interruptor de energización podría también ser utilizado para respaldar con batería una RAM estática de baja potencia.

La Tabla 2.3. y la Tabla 2.4. detallan el Registro de Datos y el Registro de Control utilizados para la manipulación interna del RTC empleado por el sistema embebido que comprende el controlador principal del sistema.

Tabla 2.3. Registros de Datos (RTCxR) del Reloj de Tiempo Real

Reloj de Tiempo Real x Registro de Retención		(RTC0R) R/W	(Dirección = 0x02)
		(RTC1R)	(Dirección = 0x03)
		(RTC2R)	(Dirección = 0x04)
		(RTC3R)	(Dirección = 0x05)
		(RTC4R)	(Dirección = 0x06)
		(RTC5R)	(Dirección = 0x07)
Bit (s)	Valor	Descripción	
7:0	Lectura	Se retorna el valor actual del registro de retención del RTC de 48 bits	
	Escritura	Escribiendo al RTC0R transfiere la cuenta actual del RTC a los seis registros de retención mientras el RTC continua contando	

Tabla 2.4. Registro de Control (RTCCR dir = 0x01) del Reloj de Tiempo Real

Bit (s)	Valor	Descripción
7:0	0x00	<p>Escribiendo un 0x00 al RTCCR no tiene efecto en el contador RTC. Sin embargo, dependiendo de cual fue el comando previo, escribiendo un 0x00 puede</p> <ol style="list-style-type: none"> 1. Deshabilitar la función de incremento de byte o 2. Cancelar el comando de reset del RTC. <p>Si el comando 0xC0 es seguido por el comando 0x00, solo la función de incremento de byte será deshabilitada. El reset del RTC todavía tendrá lugar.</p>
	0x40	Arma el RTC para un reset con código 0x80 o reset y función de incremento de byte con código 0x0C0
	0x80	Resetea todos los seis bytes del contador RTC a 0x00 si procedió el comando de armado 0x40
	0xC0	Resetea todos los seis bytes del contador RTC a 0x00 y entra en modo de incremento de byte. Preceda a este comando con el comando de armado 0x40
7:6	01	Esta combinación de bits debe ser usada con cada escritura del incremento de byte para incrementar los registros del reloj (es) correspondientes a los bits que contienen "1". Ejemplo: 01001101 incrementa los registros 0, 2, 3. El modo de incremento de byte debe ser habilitado. Almacenando 0x00 cancela el modo de incremento de byte.
5:0	0	No causa efecto en el contador RTC
	1	Incrementa el byte correspondiente del contador RTC

2.6. MÓDULO GRABADOR / REPRODUCTOR DE AUDIO

2.6.1. Descripción

El módulo grabador/reproductor de audio (el A96020, en este caso), ofrece la capacidad de almacenamiento verdadero en un solo chip de estado sólido y no requiere software ni apoyo de microcontrolador. Proporciona alta calidad de

grabación y reproducción de dos mensajes de hasta 20 segundos cada uno. En la Figura 2.8. se puede apreciar una fotografía de este módulo.

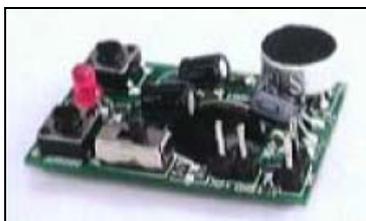


Figura 2.8. Módulo Grabador/Reproductor de Audio

2.6.2. Características

- Tecnología de memoria flash no volátil.
- No requiere batería de respaldo (Back-up).
- Selector para grabar o reproducir.
- Pulsadores separados para cada mensaje.
- Bajo consumo de energía.
- LED indicador para grabación y reproducción.
- Micrófono tipo condensador.
- Alimentación de 5V a 6V
- Dimensiones: 34 x 23 mm

La Tabla 2.5. presenta las especificaciones eléctricas básicas del módulo grabador/reproductor de audio.

Tabla 2.5. Especificaciones Eléctricas

Parámetro	Mín.	Típ.	Máx.	Unidad
Voltaje de operación	5	6	6.5	V
Corriente de operación (Conectado parlante de 8Ω)				
Grabando		45		mA
Reproduciendo		70		mA
Corriente en modo de espera		200	250	μA

2.6.3. Operación de Grabación

- Colocar el selector en la posición REC, presionar M1 o M2 para iniciar la grabación, libere el pulsador cuando termine la grabación. El LED se encenderá durante la grabación. Si el mensaje es mayor de 20s, se dará el pitido dos veces y el LED se apagará y finalizará la grabación.
- Si existe grabado un mensaje, éste será reemplazado por el nuevo.

2.6.4. Operación de Reproducción

- Colocar el selector en la posición PLAY, presionar M1 o M2 para iniciar la reproducción. El mensaje será reproducido de principio a fin. Durante el modo PLAY, si se presiona un pulsador, el mensaje se detendrá inmediatamente.
- Si el mensaje nuevo grabado es menor que el mensaje previo, la porción restante del mensaje previo será borrada.

2.6.5. Estado de espera (Standby)

El módulo entra automáticamente en estado de espera luego que la operación de grabación o reproducción se ha completado.

2.7. AMPLIFICADOR DE AUDIO

2.7.1. Especificaciones

Este amplificador de audio ha sido desarrollado a partir del circuito integrado TDA2616, cumpliendo las siguientes especificaciones:

- Potencia máxima: 2 x 15W (4Ω) o 2 x 10W (8Ω).
- Protección térmica contra cortocircuitos.
- Alimentación:
 - DC Bipolar de $-12V_{DC}$ y $12V_{DC}$.
 - AC de $12V_{RMS}$.

Puede utilizarse cualquier tipo de alimentación ya que el amplificador posee un circuito de rectificación y filtrado.

La Figura 2.9. muestra una fotografía del amplificador de audio utilizado en el sistema, para poder dar una señal de alerta audible a los usuarios dentro del andén.

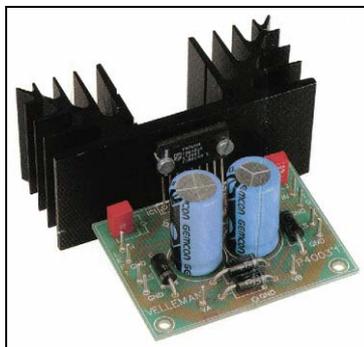


Figura 2.9. Amplificador de Audio

2.7.2. Características Técnicas

- Salida de potencia musical: 2 x 30W / 4Ω
- Salida RMS: 2 x 15W_{RMS} / 4Ω o 2 x 10W_{RMS} / 8Ω.
- Distorsión armónica total: 0.07% (1W / 1KHz).
- Separación de canales 70dB.
- Reproducción de frecuencia: de 7Hz a 60 KHz (-3dB).
- Relación señal / ruido: 98 dB.
- Sensibilidad en la entrada: 300mV / 20KΩ.
- Supresión del ruido seco en la parada y puesta en marcha.
- Protección contra cortocircuitos y sobrecarga: máximo 1 hora.
- Alimentación: 2 x 12 V_{AC} / 2A (recomendado transformador de 50W).
- Dimensiones: 70mm x 50mm.

2.8. PROGRAMACIÓN ORIENTADA A OBJETOS (POO)

La programación orientada a objetos tomó posición como la metodología de programación dominante a mediados de los años ochenta, en gran parte debido a la influencia de C++, una extensión del lenguaje de programación C. Su dominación fue consolidada gracias al auge de las Interfaces gráficas de usuario, para los cuales la programación orientada a objetos está particularmente bien adaptada. En este caso, se habla también de programación orientada a eventos.

2.8.1. Características de la POO

Hay un cierto desacuerdo sobre exactamente qué características de un método de programación o lenguaje le definen como "orientado a objetos", pero hay un consenso general en que las características siguientes son las más importantes:

- **Abstracción:** Cada objeto en el sistema sirve como modelo de un "agente" abstracto que puede realizar trabajo, informar y cambiar su estado, y "comunicarse" con otros objetos en el sistema sin revelar cómo se implementan estas características. Los procesos, las funciones o los métodos pueden también ser abstraídos y cuando lo están, una variedad de técnicas son requeridas para ampliar una abstracción.
- **Encapsulamiento:** también llamado "ocultación de la información". Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una interfaz a otros objetos que especifica cómo pueden interactuar con los objetos de la clase. El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas, solamente los propios métodos internos del objeto pueden acceder a su estado. Esto asegura que otros objetos no pueden cambiar el estado interno de un objeto de maneras inesperadas, eliminando efectos secundarios e interacciones inesperadas. Algunos lenguajes relajan esto, permitiendo un acceso directo a los datos internos del objeto de una manera controlada y limitando el grado de abstracción. La aplicación entera se reduce a un agregado o rompecabezas de objetos.
- **Polimorfismo:** comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado. Cuando esto ocurre en "tiempo de ejecución", esta última característica se llama asignación tardía o asignación dinámica. Algunos lenguajes proporcionan medios más estáticos (en "tiempo de compilación") de polimorfismo, tales como las plantillas y la sobrecarga de operadores de C++.

- **Herencia:** las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que reimplementar su comportamiento. Esto suele hacerse habitualmente agrupando los objetos en clases y estas en árboles o enrejados que reflejan un comportamiento común. Cuando un objeto pertenece a más de una clase se dice que hay herencia múltiple; esta característica no está soportada por algunos lenguajes (como Java).

2.8.2. Plataforma de Desarrollo .NET

La plataforma de desarrollo .NET nace por la necesidad de enfrentar los problemas existentes con la heterogeneidad de tecnologías y lenguajes de programación que aparecieron como respuesta a necesidades tecnológicas del momento. Estas respuestas tecnológicas aisladas produjeron multitud de servicios duplicados, creación de servicios exclusivamente para algunos lenguajes, poca reutilización de código, más complejidad de desarrollo, y soluciones poco documentadas.

La solución definitiva se ha planteado con .NET que consta de una serie de servicios iguales en todos los lenguajes que mantienen la integridad con los desarrollos existentes y hace posible una interoperatividad entre los lenguajes desconocida hasta el momento. Esto es podemos utilizar varios lenguajes diferentes (los permitidos .NET) y todos tendrán disponibles desde el mismo entorno de desarrollo hasta los controles y componentes de programación.

2.8.2.1. Modelo de capas de la plataforma .NET

La mejor forma de comprender cómo funciona .NET es analizar las numerosas capas en las que se divide .NET Framework como se muestra en la Figura 2.10.

.NET Framework constituye la base sobre la que se asienta .NET. Es la pieza fundamental de esta nueva tecnología y es la que proporciona las herramientas y servicios que se necesitan para desarrollar aplicaciones. Podemos agrupar en tres bloques el conjunto de herramientas y servicios:

- El runtime de lenguaje común o entorno de ejecución común (CLR)
- Biblioteca de clases base de la plataforma .NET (.NET Framework Base Classes)
- Motor de generación de la interfaz para crear formularios e interfaces de usuario.

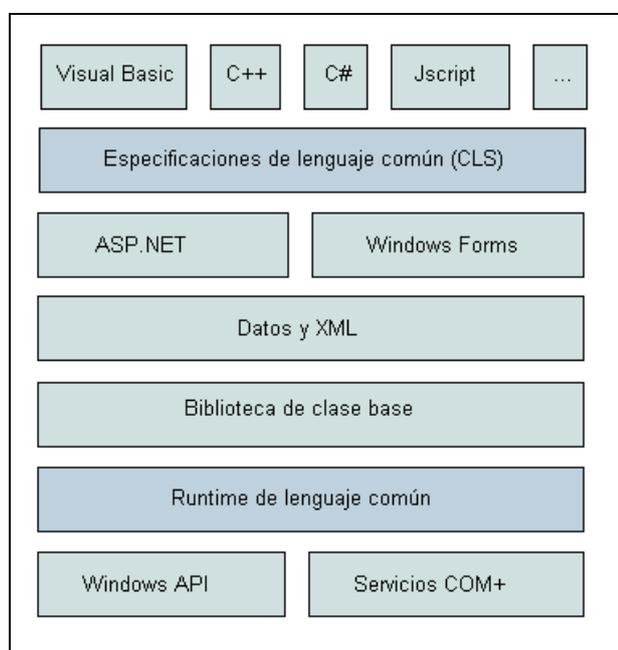


Figura 2.10. Modelo de Capas de la Plataforma .NET

Entorno de ejecución común

El entorno de ejecución común es la primera capa que pertenece a .NET Framework. Esta capa es la responsable de los servicios básicos de .NET, tales como la administración de memoria, la recolección de los elementos no utilizados, el control estructurado de excepciones y del subprocesamiento múltiple. Si .NET se transporta a otras arquitecturas que no estén basadas en Windows el primer paso sería escribir un runtime del lenguaje para el nuevo equipo. El CLR tiene estas características:

- Proporciona mejoras para el programador que antes tenía que elaborar
- Administra el código en tiempo de ejecución: carga en memoria, liberación de memoria, etc.
- Gestiona la seguridad del código ejecutado
- Abre posibilidades a otros fabricantes para incorporar sus lenguajes
- Facilita la distribución e instalación de aplicaciones. Elimina los temibles conflictos de DLL's y versiones de ellas.

Biblioteca de clases

La biblioteca de clases base (BCL) es la parte de .NET Framework que define todos los tipos de datos básicos, tales como System.Object (raíz de la jerarquía de objetos .NET), tipos numéricos y de fechas, tipo string, matrices y colecciones. La BCL contiene también clases que administrarán las características centrales de .NET: entrada/salida de archivos, subprocesamiento, serialización y seguridad. La forma en la que los tipos se implementan en la BCL siguen las especificaciones llamadas Common Type System (CTS). Por ejemplo, estas especificaciones dicta la forma en la que un tipo .NET expone campos, propiedades, métodos y sucesos, también cómo un tipo puede heredar otro tipo.

Capa de datos

La capa de datos y XML contienen las clases .NET que trabajan con bases de datos y con XML. Anteriormente el soporte XML era una compatibilidad proporcionada por un componente externo. En .NET vemos cómo está integrado en su mismo núcleo. Podemos decir que XML es el formato que utiliza .NET para almacenar cualquier tipo de información. La parte de datos es la que se conoce como ADO.NET y es el equivalente en .NET a la tecnología ActiveX Data Object (ADO), ampliamente conocida por los programadores de Visual Basic e Intranets.

Capa de formularios y ASP

Las dos capas siguientes son ASP.NET y Windows Forms. Aquí se sitúan todas las clases que podremos utilizar para generar las páginas web en el primer caso y las ventanas estándares o formularios en las aplicaciones de Windows en el segundo caso.

Aunque estén en el mismo nivel las tecnologías son muy distintas. Web Forms se ejecuta en el servidor y produce HTML y es la base de las conocidas Intranets donde estas páginas devuelven otras páginas web con conjuntos de resultados u otros datos. Windows Forms se ejecuta en el cliente, un equipo Windows.

Un ensamblado consiste en un conjunto de tipos y recursos reunidos para formar la unidad mas elemental de código que puede ejecutar el entorno .NET Framework. Estos ensamblados son los elementos con los que construiremos una aplicación: una aplicación .NET se compone de varios ensamblados. Podemos llamar también a un ensamblado como una DLL lógica (recordamos que una DLL es una biblioteca de enlaces dinámicos). Decimos lógica porque se trata de una lista de ficheros que se referencian en tiempo de ejecución pero que no se compilan para producir un fichero físico, a diferencia de las DLL's tradicionales. Un ensamblado va mucho más allá ya que puede contener otros recursos además de clases como imágenes.

Capas de CLS y lenguajes

En estas capas están las especificaciones de los lenguajes como Visual Basic 2005 Express y su sincronización con el entorno .NET.

2.9. MANEJO DE ARCHIVOS

Una de las principales funciones de un Sistema Operativo es la administración del almacenamiento de información, para lo cual es necesario contar con un "Sistema de Archivos". Con este término se hace referencia, por un lado, a los mecanismos y estructuras que el sistema operativo utiliza para organizar la información en medios físicos tales como discos y diskettes (aspecto físico del sistema de archivos), y por otro a la visión que es ofrecida al usuario para permitir la manipulación de la información almacenada (una abstracción, o perspectiva lógica del sistema de archivos).

2.9.1. Característica “My” de Visual Basic 2005 Express

Visual Basic proporciona nuevas características para el rápido desarrollo de aplicaciones que mejoran la productividad y facilitan su uso mientras mantienen un alto poder de desempeño. Una de estas características, llamada My, proporciona acceso a información e instancias objetivas por defecto que están relacionadas a la aplicación y su ambiente en tiempo de ejecución. Esta información es organizada en un formato que es accesible a través de “IntelliSense” (menús contextuales), y lógicamente delineada de acuerdo al uso.

Los miembros de mayor jerarquía de “My”, son expuestos como objetos. Cada objeto comparte similitudes con un “NameSpace” (agrupaciones de patrones similares) o clase con miembros compartidos, y son expuestos como un conjunto de miembros relacionados.

La Figura 2.11. muestra los objetos de mayor jerarquía de My y sus relaciones con cada uno.

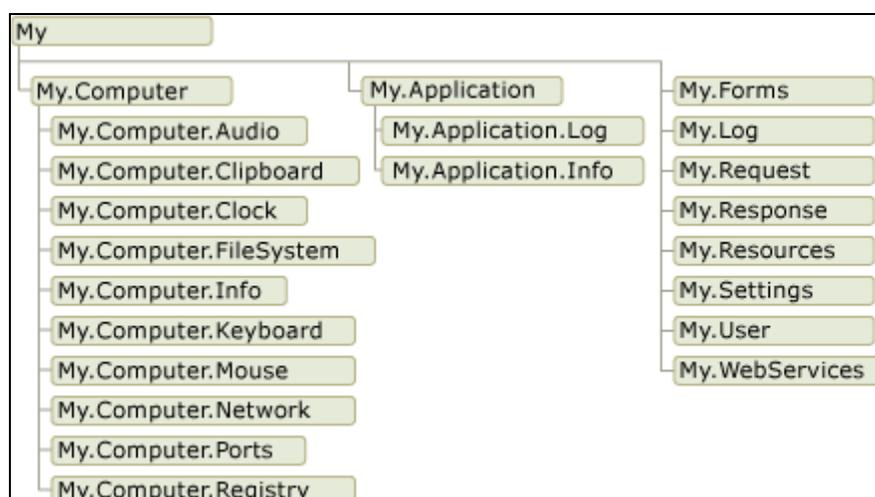


Figura 2.11. Jerarquía de la Característica “My”

2.9.2. Manejo de Archivos Mediante “My”

El manejo de archivos en .NET mediante la característica My proporciona un mejor desempeño en comparación con las clases utilizadas en versiones anteriores de Visual Basic.

El objeto utilizado para el manejo de archivos se encuentra en `My.Computer.FileSystem` de acuerdo al esquema jerarquizado de la Figura 2.11., el cual proporciona las propiedades y métodos necesarios para trabajar con unidades lógicas, archivos y directorios.

2.9.3. Atributos y Niveles de Acceso con Archivos

Es posible controlar, la manera en que los archivos son creados, abiertos, y compartidos mediante la característica `My`, mediante un parámetro adicional que distingue la sobrecarga de los métodos del objeto `My.Computer.FileSystem`.

El parámetro adicional usado por la característica `My` de Visual Basic 2005 para establecer los modos de acceso al archivo es "Append". Si el valor de `Append` es verdadero los datos que se escriban serán adicionados caso contrario serán sobrescritos.

Visual Basic 2005 usa un sistema de excepciones para manejar errores o fallas de la aplicación en tiempo de ejecución. Las excepciones en la escritura de archivos mediante `My`, y sus causas son las siguientes:

- El path o ruta no es válido por una de las siguientes razones: Es una cadena de longitud cero, contiene solamente espacios en blanco, o contiene caracteres no válidos (`ArgumentException`).
- El path o ruta no es válido porque es nulo (`ArgumentNullException`).
- El archivo apunta a una ruta que no existe (`FileNotFoundException` o `DirectoryNotFoundException`).
- El archivo es usado por otro proceso, y un error de I/O ocurre (`IOException`).
- La ruta excede la longitud máxima definida por el sistema (`PathTooLongException`).
- Un archivo contiene una coma o esta en un formato no válido (`NotSupportedException`).
- El usuario carece de los permisos necesarios para acceder a la ruta (`SecurityException`).

CAPÍTULO 3

DISEÑO DEL HARDWARE

3.1. DIAGRAMA DE BLOQUES DEL SISTEMA

El diagrama de bloques permite un mejor entendimiento del sistema, al poder visualizar gráficamente los componentes principales del mismo. La Figura 3.1. muestra este diagrama.

El **controlador principal** es el encargado de centralizar y administrar el flujo de información necesaria para la operación del sistema. Esta operación empieza con la detección del trolebús en la estación, mediante el **sensor infrarrojo**, acción que establece que se encienda la luz roja del **semáforo**. Luego de haber transcurrido un tiempo configurado en el controlador principal, empieza a visualizarse la cuenta regresiva en los **módulos de displays** y se emite el mensaje de apertura de las puertas mediante el **módulo de voz**, el cual se encuentra acoplado al **amplificador de audio** mediante un **transformador de audio**, debido a que la salida del módulo de voz presenta una señal diferencial y este transformador permite aprovechar al máximo dicha señal. Un tiempo (también configurado en el controlador principal) antes de que la cuenta llegue a su fin se emite el mensaje de cierre de puertas y, al finalizar la cuenta, el semáforo enciende la luz verde para indicar al conductor que ha finalizado su tiempo de estadía en la estación y que puede cerrar las puertas y continuar su recorrido.

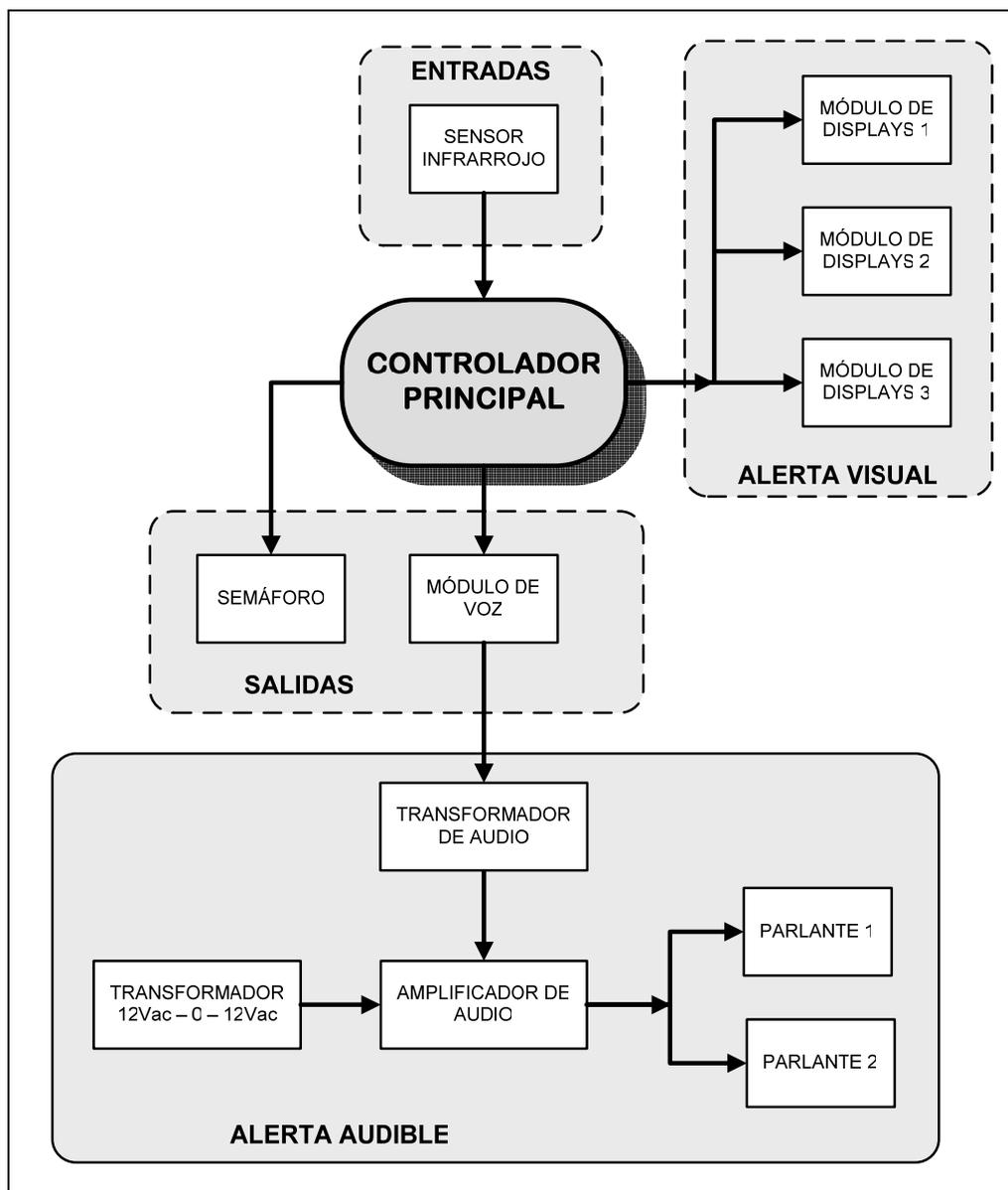


Figura 3.1. Diagrama de Bloques del Sistema

A continuación se proporciona mayor detalle sobre la constitución interna de cada bloque mostrado en la Figura 3.1.

3.1.1. Controlador Principal

El controlador principal es el módulo OP6810 MiniCom, fabricado por la empresa Rabbit Semiconductor.

El sistema necesita reunir varias características, entre las cuales se puede mencionar la necesidad una memoria de tipo no volátil, que permita respaldar la información crítica para la operación del sistema, así como el código del programa. Adicionalmente, son necesarios puertos de comunicación serial para operaciones de modificación y respaldo de los parámetros de operación del sistema (RS-232) y para la transmisión fiable de datos a una distancia considerable (RS-485). Debido a que el controlador principal administra todas las acciones a realizarse, se hace necesaria la presencia de varias entradas y salidas que permitan ejecutar las acciones pertinentes, además de la manipulación de un RTC que gestione las temporizaciones correspondientes al cierre de puertas, de acuerdo a la hora y el día que se esté cursando. El diagrama de bloques correspondiente a la arquitectura del controlador principal se presenta en la Figura 3.2.

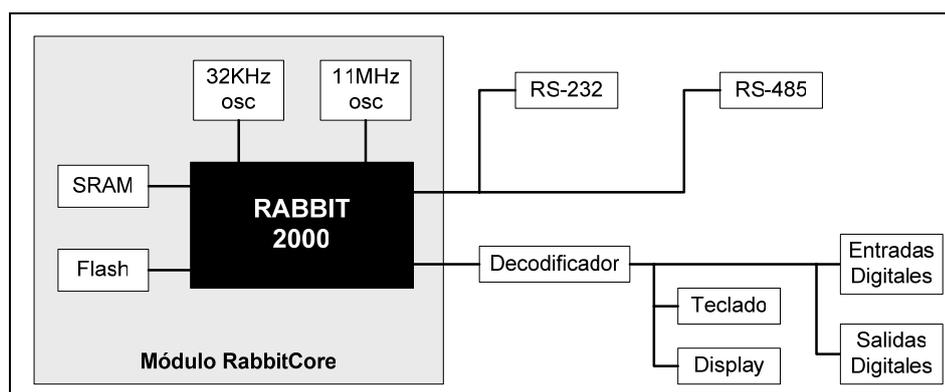


Figura 3.2. Diagrama de Bloques del Módulo OP6810

Como se puede apreciar en la Figura 3.2., existen 12 bloques principales que componen el controlador principal, éstos cumplen las siguientes funciones:

- **SRAM.** Memoria RAM estática, respaldable con una batería externa de 3V, se utiliza para almacenar los datos críticos para la operación del programa, es decir, datos que se desean mantener aún cuando se pierda la alimentación del sistema.
- **Flash.** Memoria flash (no volátil), destinada al almacenamiento del código del programa.

- **32KHz osc.** Oscilador reservado para la operación del RTC, opera con la batería de respaldo.
- **1MHz osc.** Oscilador para el funcionamiento general del controlador principal.
- **RABBIT 2000.** Microcontrolador con el cual opera el controlador principal.
- **RS-232.** Puerto de comunicación serial con el estándar RS-232. En este caso, el puerto es utilizado para la comunicación con el computador.
- **RS-485.** Puerto de comunicación serial con el estándar RS-485. En este caso, el puerto es utilizado para el envío de datos a los módulos de displays.
- **Decodificador.** Este es un decodificador de puertos, el cual permite la manipulación de los periféricos del microcontrolador.
- **Teclado.** Conformado por 7 teclas, que en este caso permiten la manipulación del software almacenado en la memoria flash.
- **Display.** Interfaz gráfica de salida. En este caso permite visualizar las acciones a ejecutar en el software, además de ciertos parámetros de información.
- **Entradas Digitales.** Entradas al controlador principal, en este caso se monitorea solamente la señal de sensor infrarrojo.
- **Salidas Digitales.** Salidas de control que ejecutan una acción determinada por el controlador principal. En este caso se manejan 5 salidas, que se distribuyen 4 para el módulo de voz (modo de reproducción, modo de grabación, reproducción del mensaje de apertura y reproducción del mensaje de cierre) y una salida para manipular las luces del semáforo.

3.1.2. Entradas

El sistema diseñado tiene una sola entrada, que es:

- **Presencia del trolebús en la estación.** Esta entrada es monitoreada de un sensor infrarrojo que se encuentra instalado sobre la segunda puerta del andén. Dicho sensor es el encargado de detectar la presencia del trolebús en la estación y entregar esta señal al controlador de cada puerta. Esta señal es útil para el sistema diseñado, por lo que se toma para ser enviada al controlador principal.

3.1.3. Alerta Visual

La alerta visual está formada por 3 módulos de displays, los cuales muestran a los usuarios dentro del andén el tiempo que permanecerán abiertas las puertas. Los bloques que conforman la alerta visual son:

- **Módulo de Displays 1.**
- **Módulo de Displays 2.**
- **Módulo de Displays 3.**

Los tres módulos y el controlador principal forman una red bajo el estándar RS-485 y cada módulo muestra la misma temporización configurada en el controlador principal, de acuerdo a la hora y al día que esté cursando.

3.1.4. Salidas

El controlador principal se encarga de proporcionar 5 salidas, las cuales son las necesarias y suficientes para el correcto desempeño del sistema. Las salidas son las siguientes:

- **Modo de reproducción.** Dispone al Módulo de voz en modo de reproducción de mensajes, los cuales deben estar previamente grabados en dicho módulo.
- **Modo de grabación.** Dispone al Módulo de voz en modo de grabación de mensajes, los cuales se sobrescribirán sobre los mensajes previamente almacenados.
- **Reproducir mensaje de apertura de puertas.** Reproduce el mensaje que se haya almacenado para la apertura de las puertas del trolebús.
- **Reproducir mensaje de cierre de puertas.** Reproduce el mensaje que se haya almacenado para el cierre de las puertas del trolebús.
- **Cambio de las luces del semáforo.** Cambia las luces del semáforo, de rojo a verde y viceversa, según corresponda el caso.

3.1.5. Alerta Audible

La alerta audible será proporcionada a los usuarios del Sistema Trolebús que se encuentren dentro del andén. Los bloques que esta alerta son:

- **Transformador de Audio.** Permite acoplar la salida del módulo de voz al amplificador de audio, para poder aprovechar al máximo la salida diferencial de dicho módulo.
- **Transformador 12Vac – 0 – 12Vac.** Este transformador se encarga de alimentar al amplificador de audio.
- **Amplificador de Audio.** Amplifica la señal de salida del módulo de voz, de manera que sea audible a los usuarios del Sistema Trolebús.
- **Parlante 1.** Salida del amplificador de audio.
- **Parlante 2.** Salida del amplificador de audio.

3.2. DIAGRAMAS Y ESPECIFICACIONES DE LOS ELEMENTOS DEL HARDWARE

Tomando como referencia los bloques mostrados en la Figura 3.1., a continuación se presenta los diagramas esquemáticos de cada etapa.

3.2.1. Controlador Principal

La Figura 3.3. muestra el diagrama esquemático de la interfaz del controlador principal, como se puede apreciar, este esquema centraliza toda la etapa de control del sistema; dicha figura evidencia lo siguiente:

- **ALIMENTACIÓN (JP2).** Es la entrada de la alimentación del circuito mostrado en la Figura 3.3., esta alimentación es proporcionada por una fuente tipo switching que proporciona $24V_{DC} / 4.5A$.
- **BACK-UP (JP3).** Es la entrada de voltaje de la batería de respaldo (back-up), el cual es igual a 3V y es proporcionado por una batería tipo moneda 2032 de litio, con 220mA-h (miliamperios-hora).
- **RS-485 (JP5).** Es la salida de datos de la red RS-485 del controlador principal. En este caso, únicamente el controlador principal envía datos por esta red y los módulos de displays reciben dichos datos.
- **ENTRADAS/SALIDAS (JP7).** Son las entradas y salidas que manipula el controlador principal, para realizar las acciones pertinentes en el desempeño del sistema. Estas entradas/salidas son tipo TTL¹, se tienen 4 salidas que se

¹ TTL: (Transistor-Transistor Logic) Lógica transistor-transistor, cuya principal característica es que su nivel lógico verdadero es de $5V_{DC}$

encargan se operar el módulo de voz y una entrada que corresponde al sensor infrarrojo que detecta la presencia del trolebús en el andén.

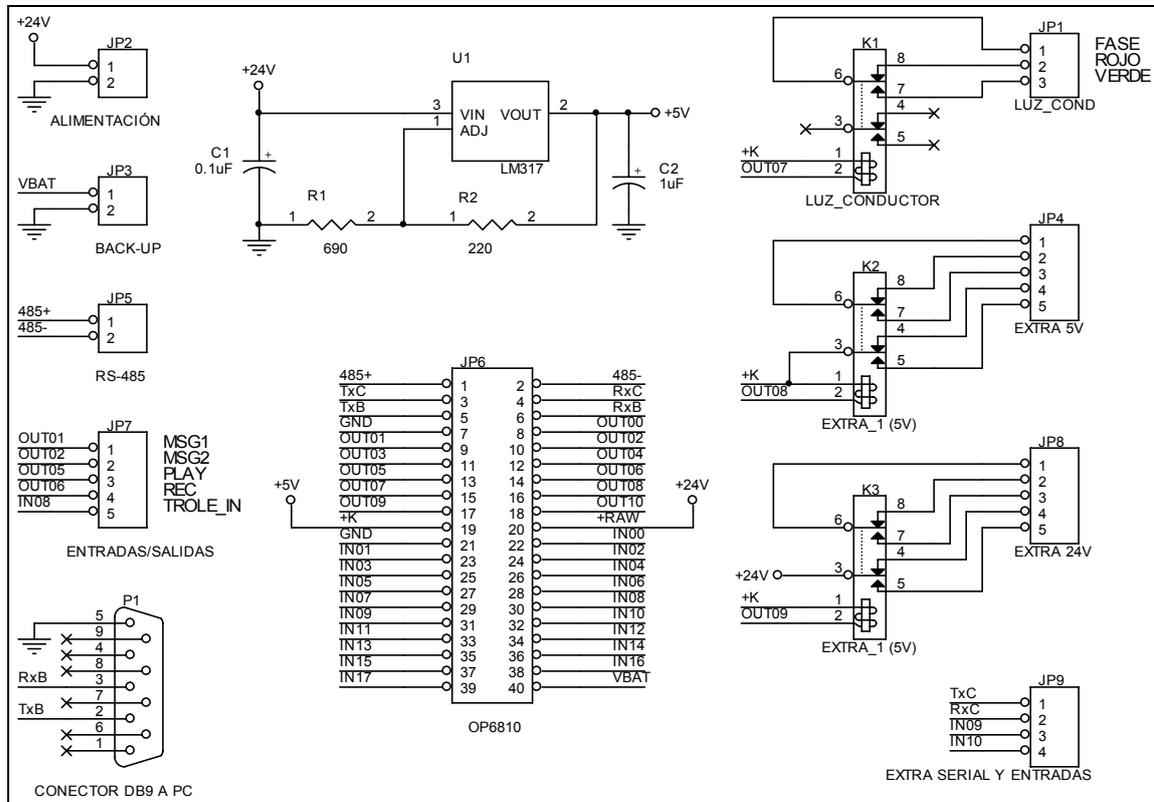


Figura 3.3. Diagrama Esquemático del Controlador Principal

- **CONECTOR DB9 A PC (P1).** Es un conector DB-9 tipo macho, el cual sirve para la conexión del controlador principal con un computador. La conexión se realiza mediante el estándar RS-232 y sirve para operaciones de manipulación y respaldo de los datos del controlador principal.
- **LM317 (U1).** El circuito integrado LM317 es un regulador de voltaje variable, el cual fue seleccionado porque soporta a su entrada un voltaje de $32V_{DC}$, permitiendo su alimentación con la fuente de $24V_{DC}$ (JP2) para obtener los $5V_{DC}$ necesarios para los circuitos TTL del diseño. Los resistores R1 y R2 respetan la ecuación para el cálculo del voltaje de salida del LM317 y los capacitores C1 y C2 son ubicados por recomendación de las hojas técnicas de este circuito integrado.

- **OP6810 (JP6).** Es el módulo OP6810 MiniCom, es decir, éste es en sí el controlador principal de todo el sistema y el resto de elementos mencionados en esta sección son los periféricos.
- **LUZ_COND (JP1).** Son los contactos de un relé que activa las luces del semáforo. Son tres contactos: común, normalmente cerrado (luz roja) y normalmente abierto (luz verde), los cuales forman un contacto seco.
- **EXTRA 5V.** Son las salidas de un relé de dos polos. El primer polo tiene en su contacto común $5V_{DC}$ y el otro polo es un contacto seco.
- **EXTRA 24V.** Son las salidas de un relé de dos polos. El primer polo tiene en su contacto común $24V_{DC}$ y el otro polo es un contacto seco.
- **EXTRA SERIAL Y ENTRADAS.** Posee los pines de transmisión (TxC) y de recepción (RxC) de un puerto serial, además de dos entradas (IN09, IN10) que se encuentran protegidas internamente para manejar cargas inductivas con un voltaje de hasta $40V_{DC}$.

Los tres últimos ítems son entradas y salidas extra que dispone el diseño, con el fin de poder ampliar el campo de aplicación del sistema.

3.2.2. Entradas

El controlador principal maneja una sola entrada: Presencia del trolebús en la estación, la cual se obtiene de un sensor infrarrojo instalado en la segunda puerta del andén y cuyo diagrama esquemático se muestra en la Figura 3.4.

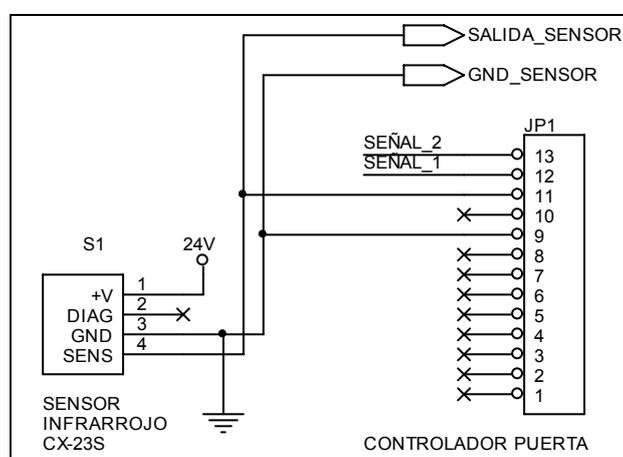


Figura 3.4. Diagrama Esquemático de Conexiones del Sensor Infrarrojo

El diagrama de la Figura 3.4. se encuentra implementado en el controlador de cada puerta del andén; los elementos aquí mostrados son:

- **CX-23S (S1)**. Este es el sensor infrarrojo utilizado en la automatización de las puertas del andén, el cual se encuentra alimentado y su señal es enviada a los controladores de las puertas.
- **CONTROLADOR PUERTA (JP1)**. Son las entradas/salidas de los controladores de las puertas, las señales: SEÑAL_1 y SEÑAL_2 (Figura 3.4.) son señales de control adicionales utilizadas para el control de las puertas y su funcionalidad no son de interés para el desarrollo del sistema diseñado.
- **SALIDA_SENSOR**. Es la señal de salida del sensor infrarrojo, la cual presenta $0V_{DC}$ cuando se detecta la presencia de un trolebús en el andén y $24V_{DC}$ cuando no existe la presencia del mismo.
- **GND_SENSOR**. Es la señal de tierra del sensor y es utilizada para enviar un nivel de voltaje coherente al controlador principal.

3.2.3. Alerta Visual

La alerta visual es proporcionada para informar a los usuarios del Sistema Trolebús el tiempo que las puertas de éste permanecerán abiertas, por lo que dicha alerta está constituida físicamente por tres módulos de displays, uno para cada puerta del andén.

Los módulos de displays están diseñados para funcionar dentro de una red RS-485, en donde el controlador principal es el encargado de llevar la cuenta descendente que es enviada de acuerdo a un ID² que identifica a los módulos de displays dentro del andén.

En la Figura 3.5. se muestra el diagrama de bloques de uno de los módulos de displays (los tres módulos tienen el mismo diseño). La ID del EI el y la cuenta enviada por el controlador el ingresa principal un transceptor de del de través RS-485, convirtiendo los datos recibidos un niveles lógicos TTL para que puedan ser

² ID: Identificador, sirve para identificar un nodo dentro de una red, en este caso de la RS-485

interpretados por el microcontroladores PIC16F84A; el además, las cuentas del se hacen trampas el salidas auxiliares para futuras implementaciones.

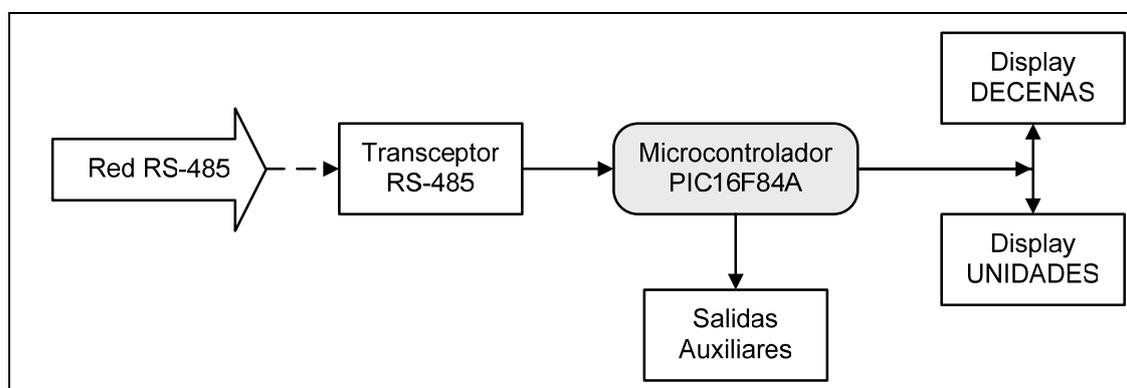


Figura 3.5. Diagrama de Bloques del Módulo de Displays

La red RS-485 implementada mantiene una comunicación unidireccional desde el controlador principal hacia los módulos de displays, para lo cual es necesario configurar el transceptor MAX485, únicamente en modo de recepción, este detalle se muestra en la sección inferior de la Figura 3.6., en donde las señales RE, y DE deben ser conectadas a tierra para cumplir este propósito.

Las salidas Disp_D y Disp_U del microcontrolador (Figura 3.6. y 3.7.) controlan el barrido de las decenas y unidades de la temporización, distinguiendo el display que será encendido de acuerdo a los datos que se encuentren el puerto B del microcontrolador.

Las salidas Lib_1, Lib_2 y Lib_3 (Figura 3.6.) constituyen señales auxiliares que pueden ser utilizadas para controlar dispositivos visuales adicionales que ayuden a mantener la máxima atención de los usuarios.

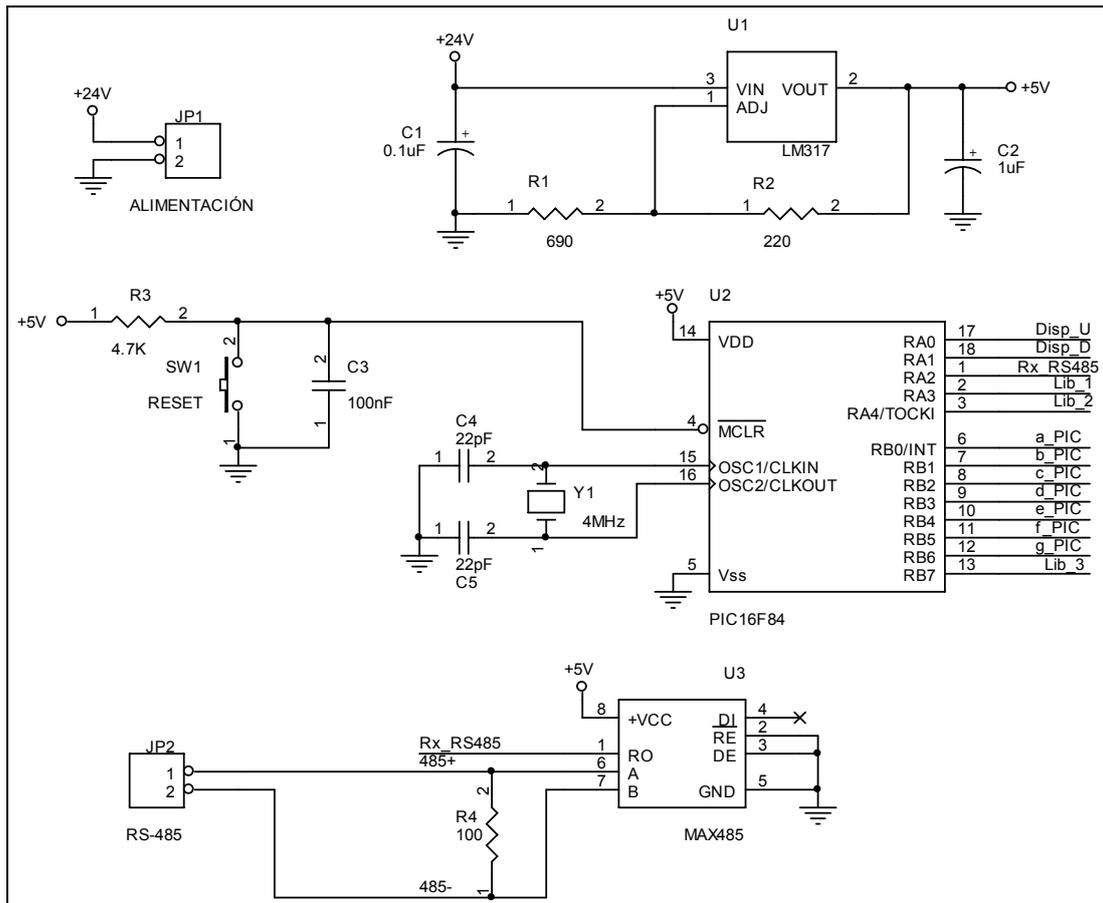


Figura 3.6. Microcontrolador PIC16F84A y Transceptor MAX485

Para realizar el barrido de los displays se tomó algunas consideraciones debido a las diferencias de voltaje de alimentación y de funcionamiento existente entre el display y las señales del microcontrolador, el primero se alimenta con 24V_{DC} y el segundo lo hace con 5V_{DC}. La Figura 3.7. muestra el circuito implementado en cada segmento del display de cátodo común, que incluye arreglos de transistores y elementos de polarización.

El arreglo formado por R1, R2, R3, R4, Q1 y Q2 (Figura 3.7.) actúa como controlador de los segmentos del display, para aislar la diferencia de voltaje antes mencionada, mientras se enciende o apaga el segmento de acuerdo a la señal de control a_PIC (Figura 3.6. y 3.7.) del microcontrolador; cada segmento restante tiene su propia señal de control (b_PIC, c_PIC,..., g_PIC). El arreglo antes mencionado se repite por cada segmento del display, es decir 7 veces y es compartido por las unidades y las decenas de la temporización a ser mostrada.

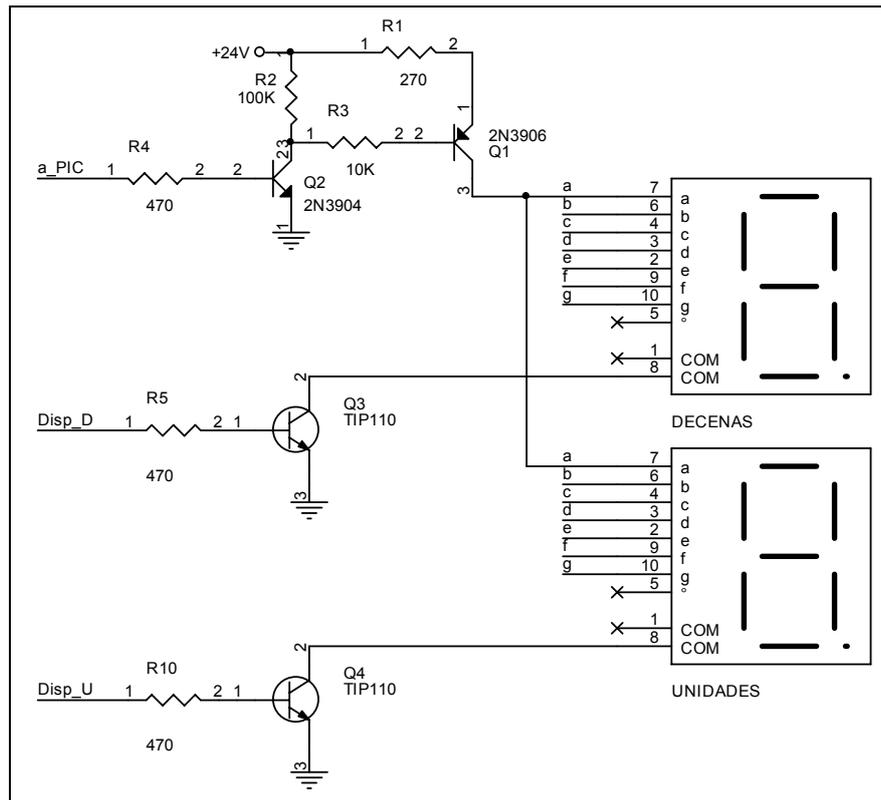


Figura 3.7. Circuito de un Segmento del Display

3.2.4. Salidas

El controlador principal, mediante sus salidas, opera el módulo de voz y el semáforo. La Figura 3.8. muestra el diagrama esquemático correspondiente a estas señales.

El controlador principal se encarga de manipular cinco señales de salida para la operación del sistema. En la Figura 3.8. pueden apreciarse cuatro de estas señales: MSG1, MSG2, PLAY y REC, las cuales se encargan de operar el módulo de voz. La quinta salida manipula dos señales a la vez, mediante los contactos de un relé: ROJO y VERDE, la señal NEUTRO corresponde al neutro de la alimentación de 110V_{AC}, estas últimas tres señales se encargan de la operación del semáforo. En la Figura 3.3. se puede apreciar la señal "FASE" que corresponde al elemento LUZ_COND, esta es la fase de la alimentación de 110V_{AC}, la cual se conmuta mediante relé y cierra el circuito, con la señal NEUTRO, para encender y apagar las luces del semáforo.

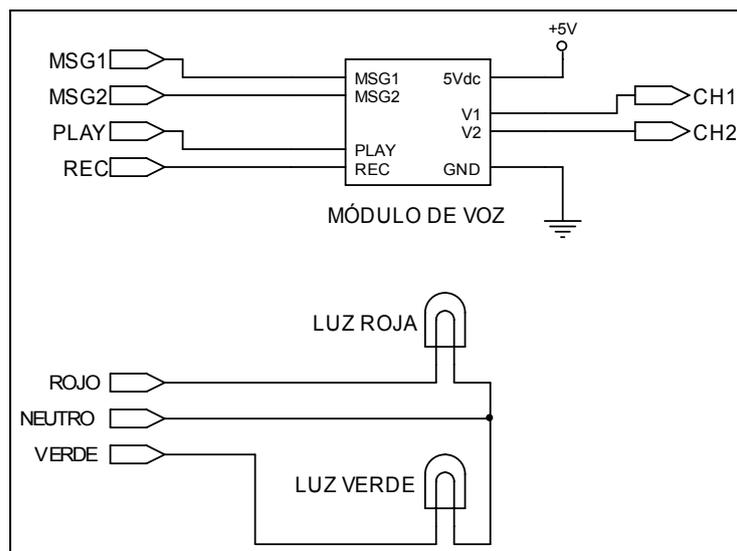


Figura 3.8. Diagrama Esquemático de las Salidas del Controlador Principal

Las señales CH1 y CH2 corresponden a la salida de audio del módulo de voz, la cual es de tipo diferencial.

3.2.5. Alerta Audible

La alerta audible es la salida amplificada del módulo de voz, es decir, una señal audible a los usuarios del Sistema Trolebús, la cual informa que sobre la apertura y el cierre de las puertas del trolebús estacionado frente al andén.

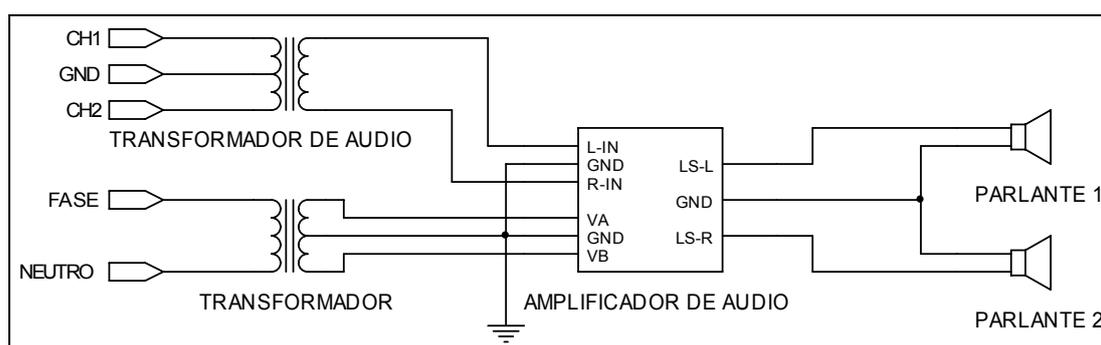


Figura 3.9. Diagrama Esquemático de la Alerta Audible

Como puede apreciarse en la Figura 3.9., la alerta audible esta formada por cuatro secciones, que son:

-
- **TRANSFORMADOR DE AUDIO.** Este transformador se encarga de acoplar la señal de salida del módulo de voz con el amplificador de audio. La salida del módulo de voz es diferencial y este transformador permite aprovechar al máximo dicha señal.
 - **TRANSFORMADOR.** Se alimenta con $110V_{AC}$ y da una salida de $12-0-12 V_{AC}$, la cual alimenta directamente al amplificador de audio.
 - **AMPLIFICADOR DE AUDIO.** Es un amplificador estéreo que proporciona una salida de 15W a una carga de 4Ω , en cada canal.
 - **PARLANTE 1 y PARLANTE 2.** Son las salidas del amplificador de audio. Deben utilizarse parlantes con una impedancia de 4Ω y que soporten una potencia de al menos 15W.

CAPÍTULO 4

DESARROLLO DEL SOFTWARE

El sistema requiere del desarrollo de tres programas, los cuales tienen la capacidad de operar en conjunto e interactuar entre sí para lograr el correcto desempeño del sistema.

A continuación se detalla el desarrollo de dicha programación:

4.1. DESARROLLO DEL SOFTWARE PARA EL COMPUTADOR PERSONAL

El software desarrollado para el computador personal se encarga de administrar la información que se descargará y se respaldará del controlador principal. Esta información comprende los siguientes datos:

- Igualar hora del RTC.
- Respalda / Descargar al controlador las fechas de los días feriados.
- Respalda / Descargar al controlador las temporizaciones de los días feriados.
- Respalda / Descargar al controlador las temporizaciones de los días ordinarios.
- Respalda / Descargar al controlador las temporizaciones de los fines de semana.
- Respalda / Descargar al controlador las fechas de los días personalizados.
- Respalda / Descargar al controlador las temporizaciones de los días personalizados.
- Cambio de clave del controlador.

El programa está desarrollado en Microsoft Visual Basic 2005 Express Edition, el Anexo 3 contiene la codificación de dicho programa.

4.1.1. Flujograma del Software Desarrollado en Visual Basic 2005

Las Figuras 4.1., 4.2. y 4.3. muestran el diagrama de flujo, dividido en tareas, del software desarrollado para el computador personal.

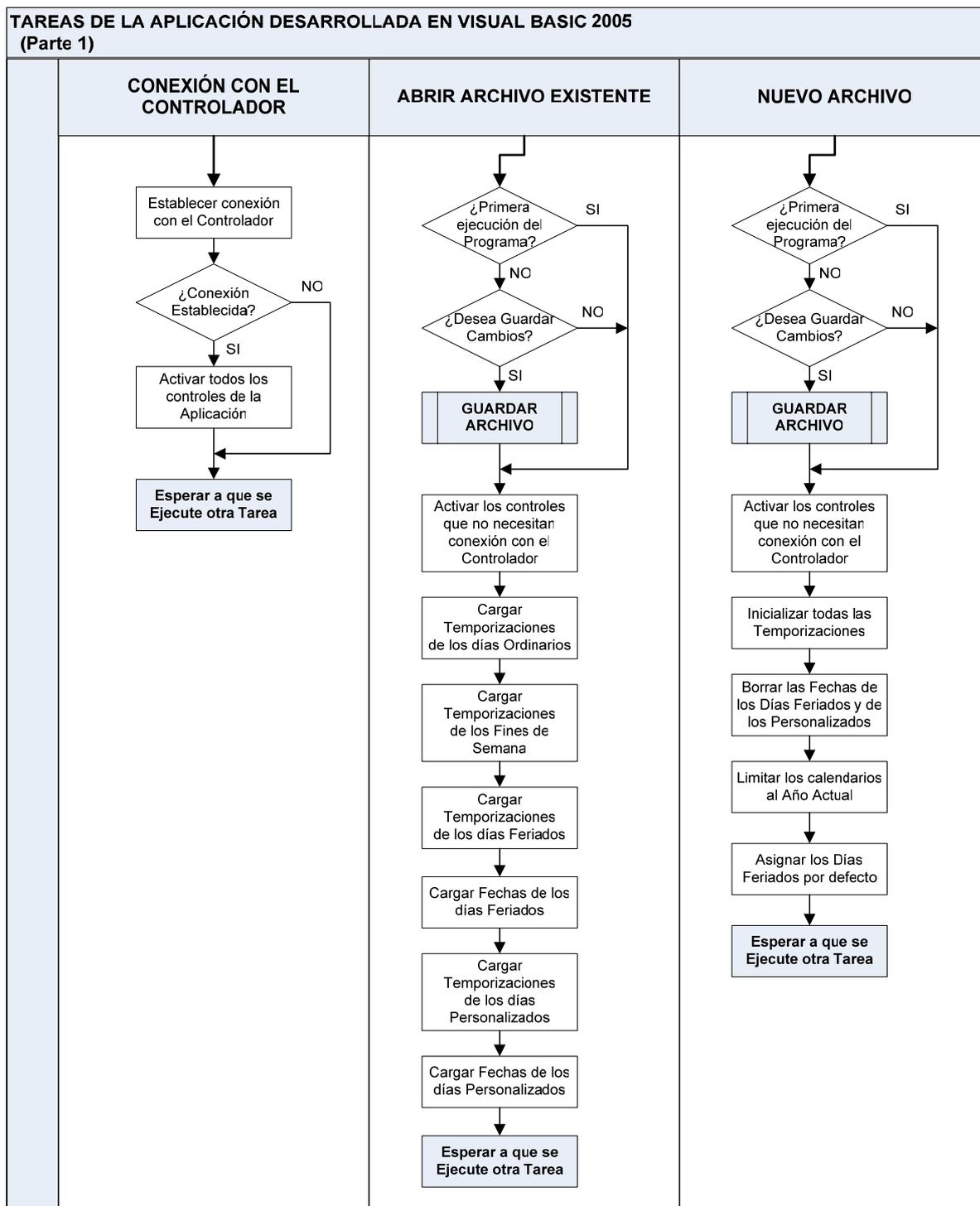


Figura 4.1. Flujograma del Software del Computador Personal (Parte 1)

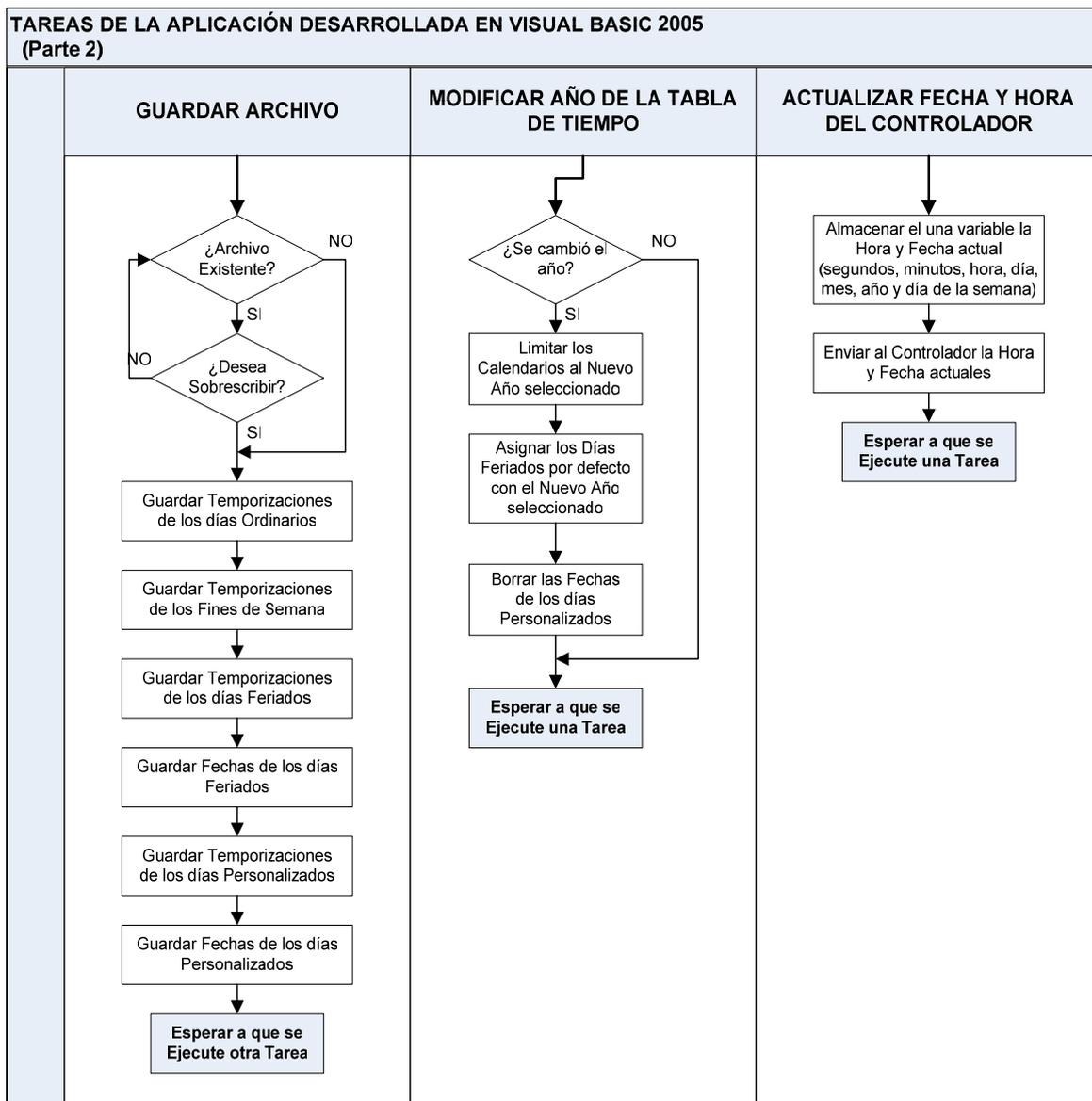


Figura 4.2. Flujoograma del Software del Computador Personal (Parte 2)

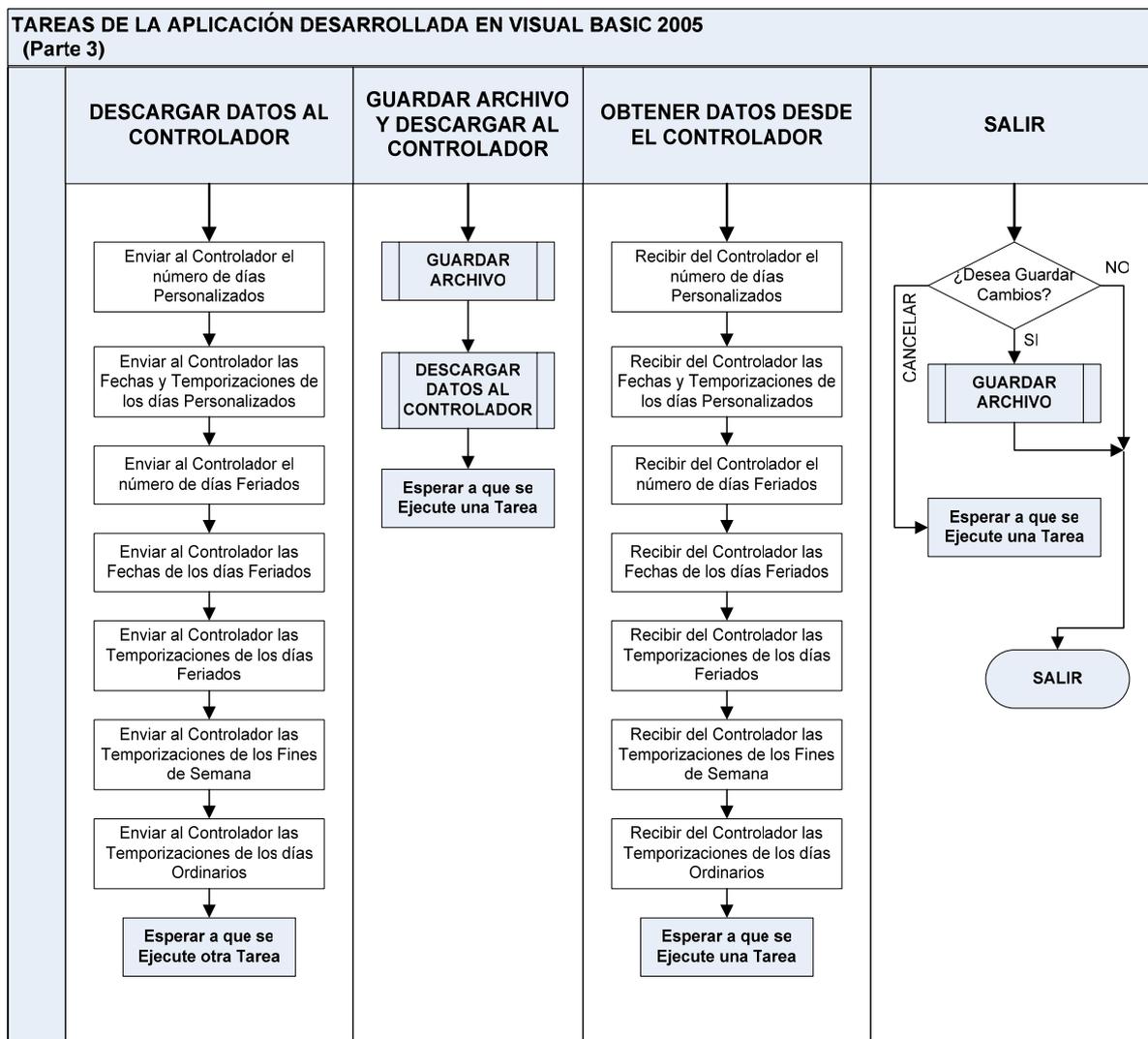


Figura 4.3. Flujoograma del Software del Computador Personal (Parte 3)

El flujoograma del software permite visualizar en forma general el desempeño del programa. Visual Basic permite la ejecución del programa en tiempo real, es decir, se ejecuta el programa mediante tareas simultáneas que se llaman dependiendo de las condiciones que establezca el usuario mediante la operación del software.

4.1.2. Tareas del Software Desarrollado en Visual Basic 2005

El software se divide en diez tareas, mostradas en las figuras correspondientes al flujoograma. Al terminar la ejecución de una de las tareas, el

programa espera a que se ejecute otra, dependiendo de la acción que realice el usuario. Dichas tareas son:

4.1.2.1. Conexión con el Controlador

Esta tarea verifica que se ha establecido la comunicación entre el computador y el controlador principal. Esta comunicación está basada en el estándar RS-232. Cuando se establece la conexión con el controlador principal se activan las herramientas que sirven para el respaldo y la descarga de datos, caso contrario, únicamente quedan activadas las herramientas para trabajar con archivos residentes en el disco duro u otro dispositivo de almacenamiento. Puede observarse el flujograma de esta tarea en la Figura 4.1.

4.1.2.2. Abrir Archivo Existente

Esta tarea permite abrir un archivo que ha sido almacenado con anterioridad en el disco duro u otro dispositivo de almacenamiento. Si se tiene abierto un archivo al ejecutar esta tarea, el programa pedirá que se confirme si se desea guardar los cambios realizados en el archivo actual. Seguidamente, se activan las herramientas que pueden operar sin conexión con el controlador y se cargan todos los datos que han sido almacenados anteriormente. Puede observarse el flujograma de esta tarea en la Figura 4.1.

4.1.2.3. Nuevo Archivo

Esta tarea permite crear un nuevo archivo de datos. Si se tiene abierto un archivo al ejecutar esta tarea, el programa pedirá que se confirme si se desea guardar los cambios realizados en el archivo actual. Seguidamente, se activan las herramientas que pueden operar sin conexión con el controlador, se limitan los calendarios al año actual y se cargan los datos por defecto en el software. Puede observarse el flujograma de esta tarea en la Figura 4.1.

4.1.2.4. Guardar Archivo

Esta tarea permite guardar todos los datos del archivo actual en el disco duro u otro dispositivo de almacenamiento, para su posterior descarga al

controlador o para modificar el archivo. Es posible sobrescribir un archivo guardado anteriormente o guardar un archivo con un nombre nuevo. Puede observarse el flujograma de esta tarea en la Figura 4.2.

4.1.2.5. Modificar Año de da Tabla de Tiempo

Esta tarea permite modificar el año de la tabla de tiempo, con el propósito de crear archivos para años futuros. Al ejecutar esta tarea se limitan los calendarios al año actual y se cargan los datos por defecto en el software. Puede observarse el flujograma de esta tarea en la Figura 4.2.

4.1.2.6. Actualizar Fecha y Hora del Controlador

Esta tarea permite actualizar el RTC del controlador principal, el cual es fundamental para la operación del sistema. Se envía al controlador la fecha y hora actual, que incluye: segundos, minutos, hora, día, mes, año y día de la semana. Para poder ejecutar esta tarea es indispensable que se haya establecido conexión con el controlador. Puede observarse el flujograma de esta tarea en la Figura 4.2.

4.1.2.7. Descargar Datos al Controlador

Esta tarea permite descargar al controlador principal todos los datos que se haya modificado en el archivo actual, es decir, descarga la tabla de tiempo anual. Para poder ejecutar esta tarea es indispensable que se haya establecido conexión con el controlador. Puede observarse el flujograma de esta tarea en la Figura 4.3.

4.1.2.8. Guardar Archivo y Descargar al Controlador

Esta tarea ejecuta en secuencia la tarea “Guardar Archivo” y luego la tarea “Descargar Datos al Controlador”, con todas las condiciones que presentan las dos tareas. Para poder ejecutar esta tarea es indispensable que se haya establecido conexión con el controlador. Puede observarse el flujograma de esta tarea en la Figura 4.3.

4.1.2.9. Obtener Datos Desde el Controlador

Esta tarea permite obtener los datos que se encuentren almacenados en el controlador principal, es decir, carga la tabla de tiempo anual en el software. Para poder ejecutar esta tarea es indispensable que se haya establecido conexión con el controlador. Puede observarse el flujograma de esta tarea en la Figura 4.3.

4.1.2.10. Salir

Esta tarea permite abandonar el programa. Admite salir del programa guardando los cambios o sin hacerlo. Puede observarse el flujograma de esta tarea en la Figura 4.3.

4.2. DESARROLLO DEL SOFTWARE PARA EL CONTROLADOR PRINCIPAL

El software desarrollado para el controlador principal se encarga de realizar el control del sistema, basándose en los datos obtenidos desde el computador personal o con datos cargados por defecto. Los datos por defecto se cargan al perderse la energía cuando la batería de respaldo está desconectada o descargada.

Este software permite la modificación de los parámetros de funcionamiento del sistema y permite la operación de los siguientes eventos:

- Detección del trolebús.
- Manipulación del semáforo.
- Emisión de los mensajes de voz: apertura y cierre de puertas.
- Administración de la tabla de temporizaciones de todas las horas de un año completo, incluyendo: días ordinarios, fines de semana, feriados y días personalizados.
- Control de la cuenta regresiva en los módulos de displays.
- Comunicación con el computador personal para operaciones de respaldo de datos.
- Respaldo de las fechas de los días feriados.
- Respaldo de las temporizaciones de los días feriados.
- Respaldo de las temporizaciones de los días ordinarios.
- Respaldo de las temporizaciones de los fines de semana.

- Respaldo de las fechas de los días personalizados.
- Respaldo de las temporizaciones de los días personalizados.
- Cambio de los parámetros de las temporizaciones: cualquier hora de cualquier día del año.
- Actualización del RTC.

El programa está desarrollado en Dynamic C, el Anexo 4 contiene la codificación de dicho programa.

4.2.1. Flujograma del Software Desarrollado en Dynamic C

El flujograma del software permite visualizar en forma general el desempeño del programa. El software empleado para el desarrollo del programa (Dynamic C) tiene la capacidad de ejecutar el programa en tiempo real, es decir, se ejecuta el programa mediante tareas simultáneas que se llaman dependiendo de las condiciones que se presenten en las entradas del controlador, para proporcionar los valores adecuados en las salidas del mismo.

Las Figuras 4.4., 4.5., 4.6. y 4.7. muestran el diagrama de flujo, dividido en tareas, del software desarrollado para el controlador principal. Las tres primeras figuras muestran tareas específicas que se ejecutarán en el momento oportuno y la Figura 4.7. muestra un modelo general de los menús empleados para que el operador del sistema pueda modificar los parámetros de funcionamiento del controlador principal.

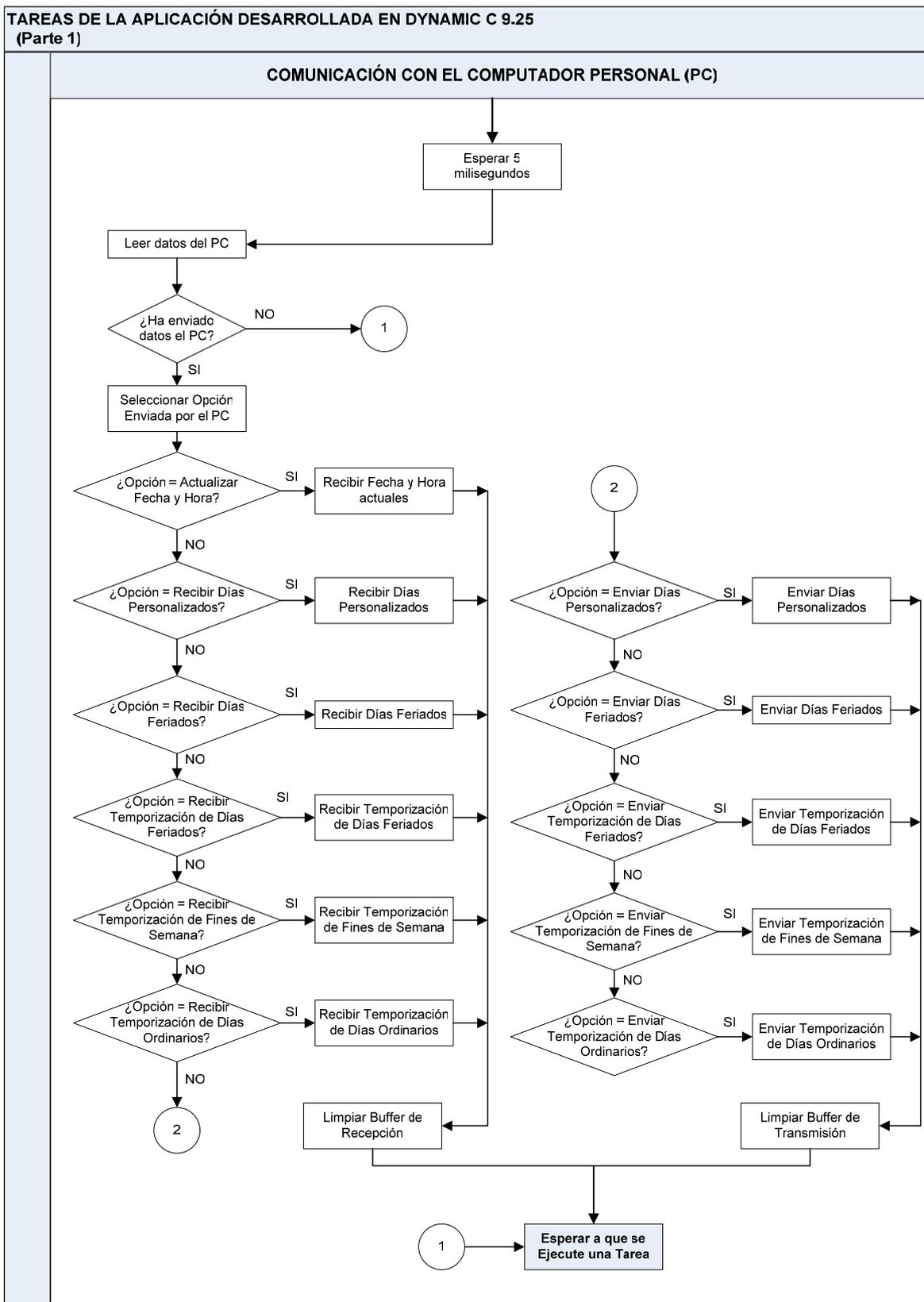


Figura 4.4. Flujograma del Software del Controlador Principal (Parte 1)

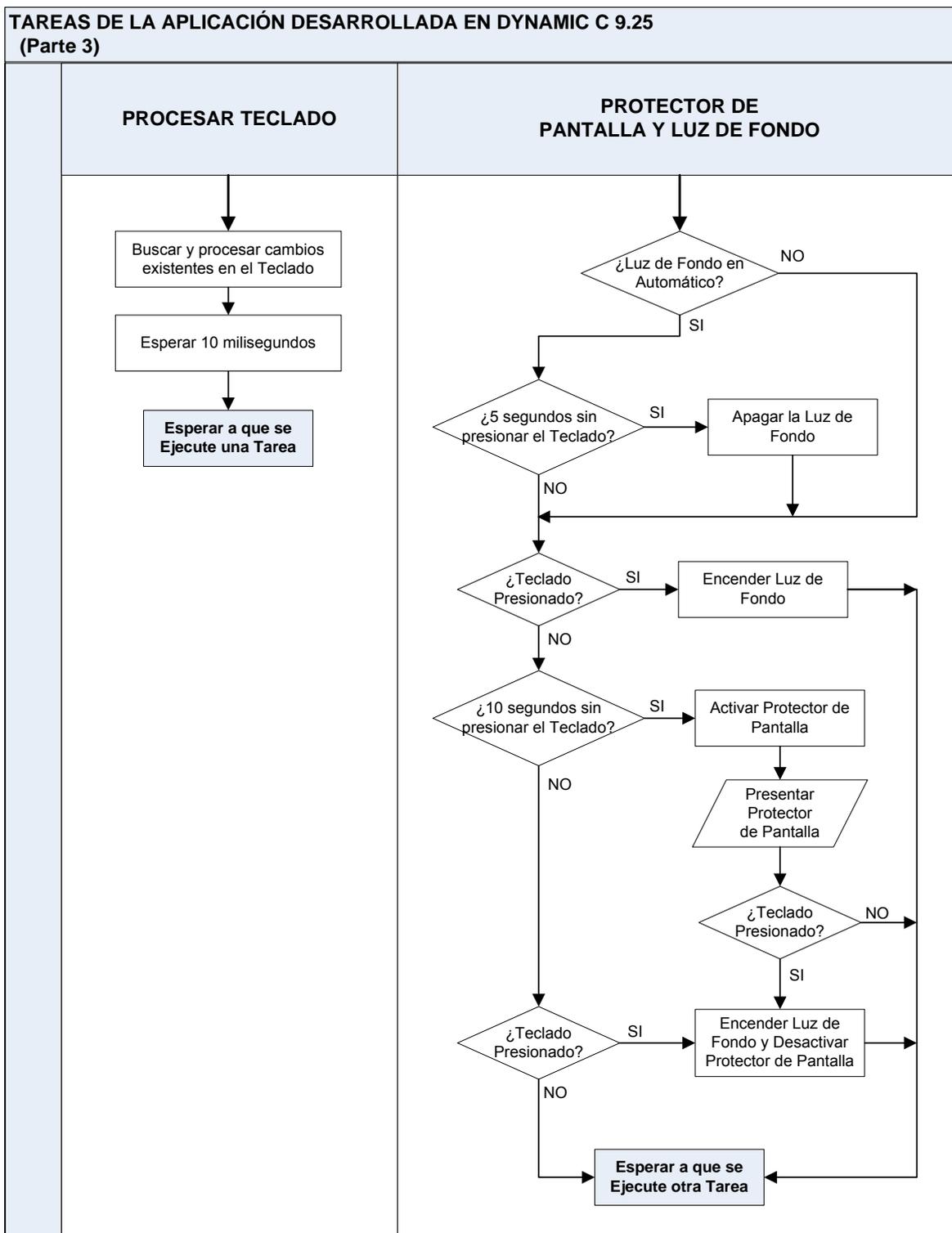


Figura 4.6. Flujograma del Software del Controlador Principal (Parte 3)

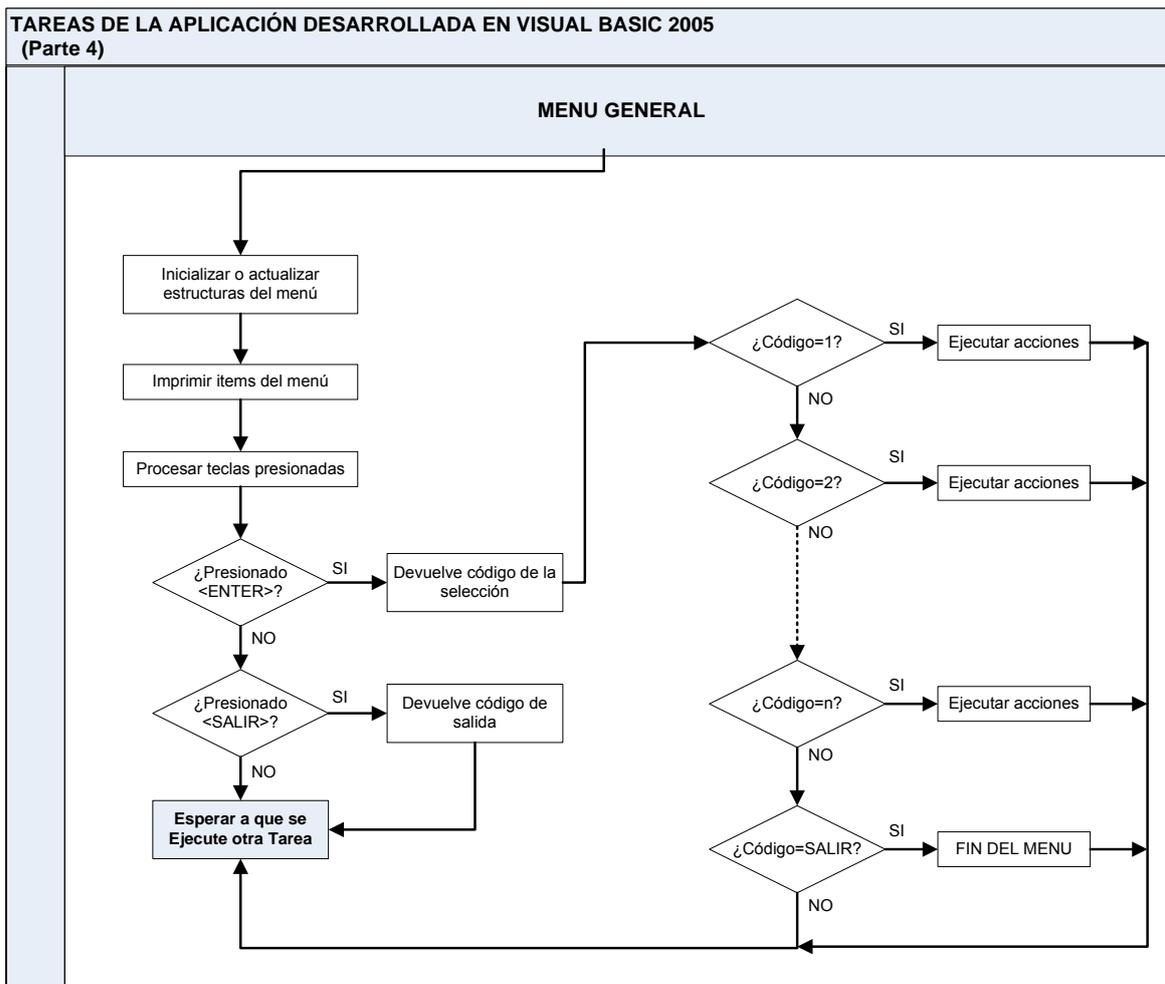


Figura 4.7. Flujograma del Software del Controlador Principal (Parte 4)

4.2.2. Tareas del Software Desarrollado en Dynamic C

El software se divide en seis tareas, mostradas en las figuras correspondientes al flujograma, cinco tareas específicas y una general para el manejo de los menús. Al terminar la ejecución de una de las tareas, el programa espera a que se ejecute otra, dependiendo de la acción que realice el operador o de las condiciones en las entradas. Dichas tareas son:

4.2.2.1. Comunicación con el Computador Personal (PC)

Para la ejecución de esta tarea es indispensable que se haya establecido la conexión (serial RS-232) entre el controlador principal y el computador. Puede observarse el flujograma de esta tarea en la Figura 4.4.

Esta tarea presenta 11 opciones en su programación, las cuales son:

- Actualizar Fecha y Hora.
- Recibir Días Personalizados.
- Recibir Días Feriados.
- Recibir Temporización de Días Feriados.
- Recibir Temporización de Fines de Semana.
- Recibir Temporización de Días Ordinarios.
- Enviar Días Personalizados.
- Enviar Días Feriados.
- Enviar Temporización de Días Feriados.
- Enviar Temporización de Fines de Semana.
- Enviar Temporización de Días Ordinarios.

4.2.2.2. Lectura Sensor

Esta tarea se encarga de detectar la presencia del trolebús frente al andén. Al detectarlo, se ejecuta la tarea de “Temporización”. Puede observarse el flujograma de esta tarea en la Figura 4.5.

4.2.2.3. Temporización

Esta tarea se encarga de la manipulación del semáforo, de la emisión de los mensajes de voz y de enviar la cuenta descendente a los módulos de displays. Para dar lugar a la ejecución de estos eventos, esta tarea lee la fecha y hora actual del RTC y es requisito imprescindible que esté presente el trolebús frente al andén, caso contrario, aborta las acciones y espera a que se ejecute otra tarea. Puede observarse el flujograma de esta tarea en la Figura 4.5.

4.2.2.4. Procesar Teclado

Esta tarea simplemente se encarga de detectar si se ha presionado una tecla cada 10ms, esto es empleado para dar lugar a la ejecución de la tarea “Protector de Pantalla y Luz de Fondo” o para desplazarse y ejecutar acciones dentro de los menús y submenús. Puede observarse el flujograma de esta tarea en la Figura 4.6.

4.2.2.5. Protector de Pantalla y Luz de Fondo

Esta tarea permite manipular el protector de pantalla y la luz de fondo. La luz de fondo puede configurarse para que siempre esté encendida, siempre apagada o en operación automática. En modo automático, si han transcurrido 5 segundos sin presionar el teclado, se apaga la luz de fondo y se enciende al presionar cualquier tecla. El protector de pantalla se presenta siempre luego de haber transcurrido 10 segundos sin presionar el teclado. Puede observarse el flujograma de esta tarea en la Figura 4.6.

4.2.2.6. Menú General

Esta tarea se ejecuta dependiendo del menú o submenú que se seleccione, con el fin de ejecutar las acciones pertinentes. Al empezar la ejecución de un menú se inicializan o actualizan los datos que sean necesarios, luego se presenta en el display del controlador principal el mensaje correspondiente y se procesa el teclado. Si se presiona <Aceptar> en el teclado, se determina la opción seleccionada y se realizan las acciones pertinentes a dicha opción; la tecla <Escape> sale del submenú actual al menú principal. Puede observarse el flujograma de esta tarea en la Figura 4.7.

La Figura 4.8. presenta un diagrama que resume la operación de los menús y submenús del software del Controlador Principal.

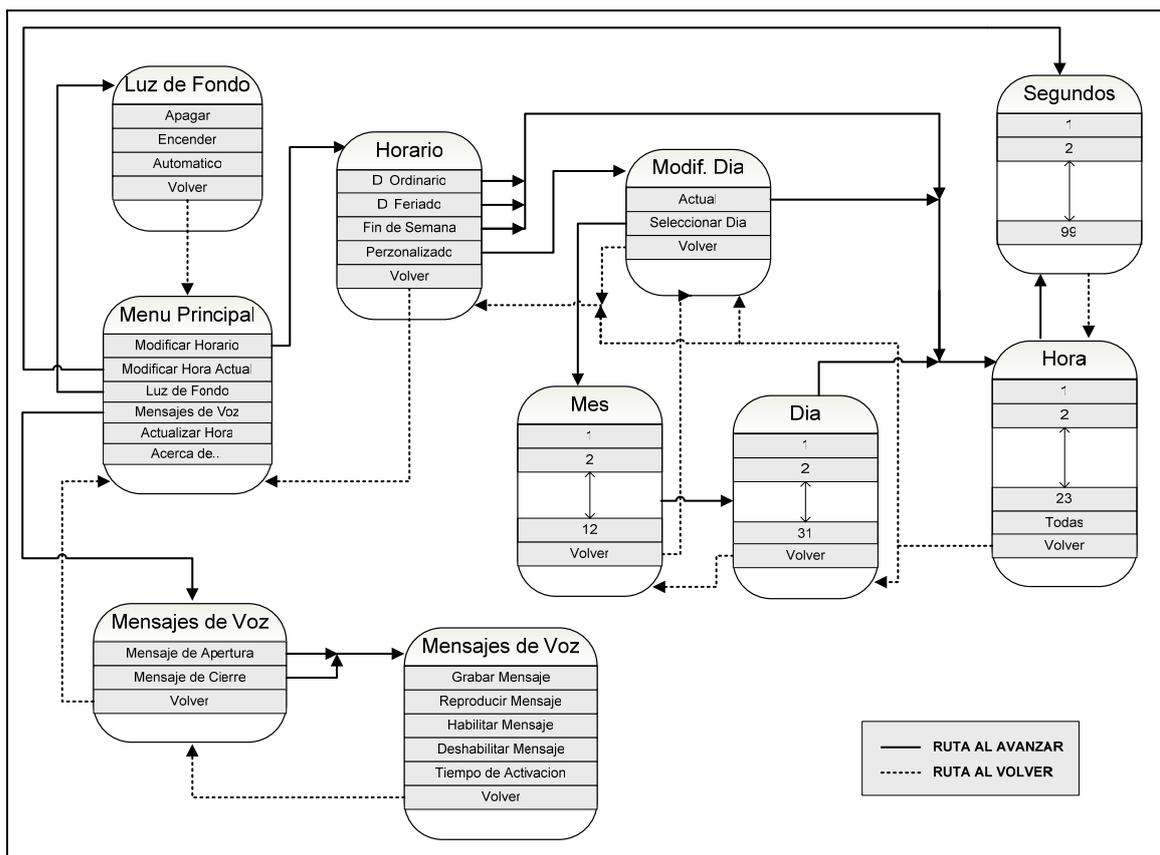


Figura 4.8. Flujo de los Menús del Controlador Principal

En el “Menú Principal” (Figura 4.8.) se encuentran las opciones principales de la operación del software desarrollado para el controlador principal, estas opciones son:

- a. **Modificar Horario.** Esta opción permite modificar la temporización a ser mostrada en los módulos de displays, para el cierre de puertas de las unidades del Sistema Trolebús. Dichas temporizaciones corresponden a:
 - Día Ordinario. Se selecciona la hora de los días ordinarios que se desea modificar, posteriormente se modifica la temporización correspondiente a dicha hora.
 - Día Feriado. Se selecciona la hora de los días feriados que se desea modificar, posteriormente se modifica la temporización correspondiente a dicha hora.

- Fin de Semana. Se selecciona la hora de los fines de semana que se desea modificar, posteriormente se modifica la temporización correspondiente a dicha hora.
- Personalizado. En este caso, se presenta un submenú para poder modificar:
 - Día Actual. Se selecciona la hora que se quiere modificar del día actual y a continuación, la temporización correspondiente a dicha hora.
 - Seleccionar Día. Se selecciona el día del año al que se quiere modificar la temporización. Primeramente se selecciona el mes, luego el día y luego la hora que se quiere modificar del día en cuestión; finalmente, se modifica la temporización correspondiente a dicha hora.

b. Modificar Hora Actual. Esta opción permite modificar la temporización de la hora que esté cursando en el instante en que se hace el cambio. Es decir, permite personalizar la hora presente del día en curso.

c. Luz de Fondo. Esta opción permite personalizar el encendido/apagado de la luz de fondo (backlight). Presenta tres opciones de operación:

- Apagar. Mantiene siempre apagada la luz de fondo.
- Encender. Mantiene siempre encendida la luz de fondo.
- Automático. Se enciende y se apaga la luz de fondo en forma automática, al presionar o dejar sin presionar el teclado un tiempo determinado.

d. Mensajes de Voz. Esta opción permite personalizar las opciones correspondientes a los mensajes de voz, tanto el de apertura, como el de cierre. Permite modificar, para cada mensaje, las siguientes opciones:

- Grabar Mensaje. Permite grabar el mensaje seleccionado. En caso de existir un mensaje ya grabado, será reemplazado por el nuevo. Debe mantenerse presionada la tecla Aceptar (↵) mientras se realiza la grabación del mensaje. Al terminar de grabar el mensaje será reproducido automáticamente para confirmar el correcto estado del mismo.
- Reproducir Mensaje. Reproduce el mensaje seleccionado.
- Habilitar Mensaje. Habilita la reproducción del mensaje seleccionado.
- Deshabilitar Mensaje. Deshabilita la reproducción del mensaje seleccionado.

- **Tiempo de Activación.** Personaliza el tiempo de activación de los mensajes. Para el mensaje de **apertura** se selecciona el tiempo que debe esperar el sistema antes de reproducir el mensaje de apertura, luego de haber detectado la presencia del trolebús. Para el mensaje de **cierre** se selecciona el tiempo en que debe reproducirse el mensaje, antes que finalice la temporización para el cierre de las puertas del trolebús.

e. Actualizar Hora. Esta opción permite actualizar el RTC, el cual constituye la base de la operación del sistema.

f. Acerca de... Despliega información acerca de los desarrolladores del proyecto, para posible soporte técnico.

4.3. DESARROLLO DEL SOFTWARE PARA EL MICROCONTROLADOR PIC16F84A

El software desarrollado para el microcontrolador se encarga de recibir los datos mediante la red RS-485 para mostrar el tiempo que permanecerán abiertas las puertas del trolebús. Los datos a recibir son los siguientes:

- ID (identificador del nodo)
- Número de la cuenta a ser mostrado en los displays.

El objeto de este ID, es diferenciar los módulos de displays que pertenecen a un sentido de circulación del trolebús, cuando el sistema es instalado dentro una parada de doble sentido (norte- sur y sur-norte), en caso de que el ID sea el correcto, los datos serán mostrados en los displays mediante barrido.

El programa está desarrollado mediante las librerías del compilador Pic C (PCW H 3.2), el Anexo 5 contiene la codificación de dicho programa.

4.3.1. Flujograma del Software Desarrollado en Pic C (PCW H 3.2)

Las rutinas implementadas en el microcontrolador PIC16F84A pueden ser explicadas mediante el diagrama de flujo que se muestra en la Figura 4.9., en la cual existen dos operaciones aritméticas:

- **UNIDADES = LECTURA % 10**. El operador “%” significa residuo de la división entera. Es decir que devuelve el residuo de una división.
- **DECENAS = LECTURA / 10**. El operador “/” significa división entera, es decir que devuelve el cociente de una división, despreciando el residuo o truncando la parte decimal.

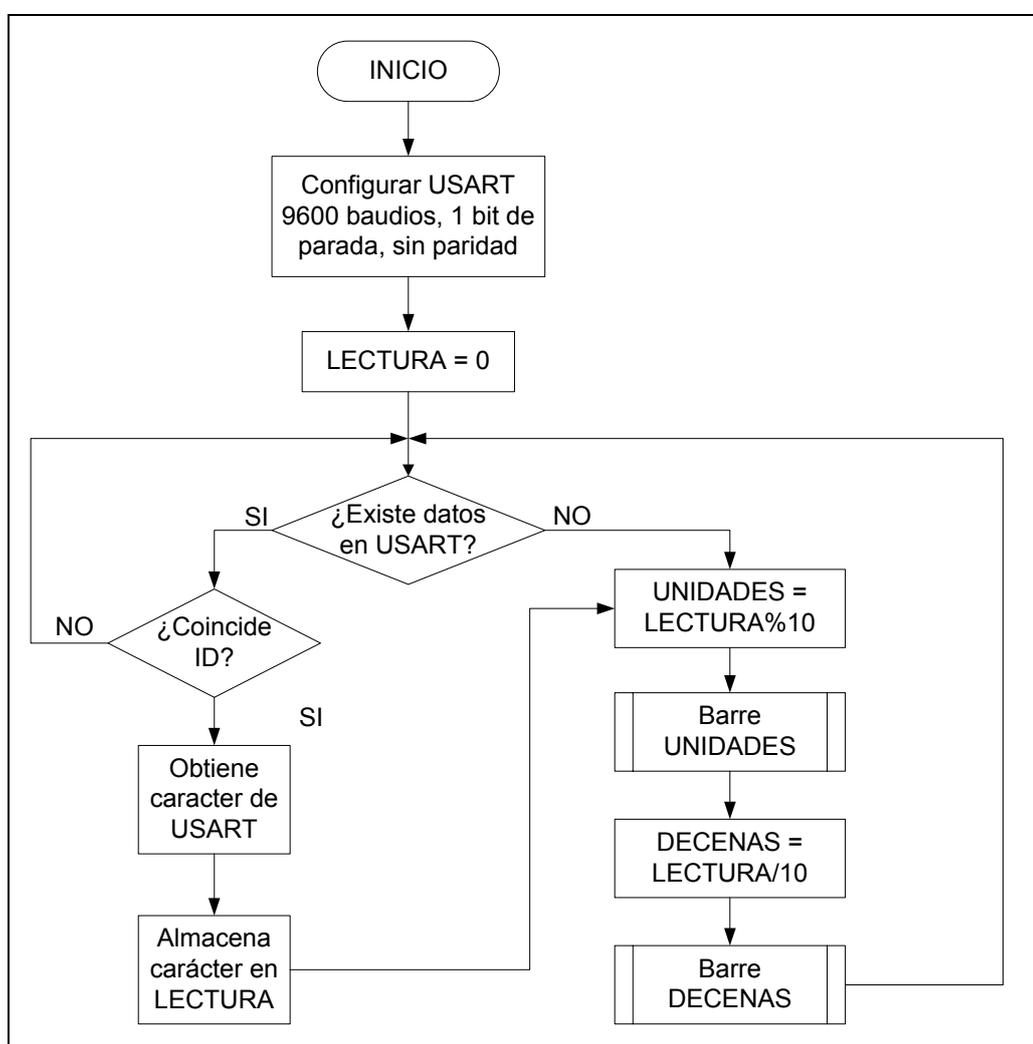


Figura 4.9. Flujograma del Software del PIC16F84A

El microcontrolador PIC16F84A **no** cuenta con un módulo USART¹, por lo que fue implementado por software mediante el compilador PCW H 3.2, que además incluye funciones que permiten preguntar por la llegada de datos mientras se desarrolla las operaciones de barrido, sin ser interrumpidas.

El barrido en los displays se realiza cada 10 milisegundos, es decir que cada par de displays conmuta su activación cada 10 milisegundos para mostrar su valor correspondiente, ya sea las unidades o las decenas. Es posible realizar dicha conmutación en el período de tiempo indicado gracias a que el PIC opera con un oscilador de 20MHz lo que produce un ciclo de instrucción de 5MHz; además, los transistores TIP110 poseen una frecuencia de operación de 20MHz. Los datos de frecuencia mencionados proporcionan coherencia para una conmutación adecuada de los displays.

¹ USART: Universal Synchronous Asynchronous Receiver Transmitter (Transmisor Receptor Universal Sincrónico Asincrónico)

CAPÍTULO 5

IMPLEMENTACIÓN

5.1. PLACAS DE CIRCUITO IMPRESO

Se han diseñado dos placas de circuito impreso o PCB¹, correspondientes a la interfaz del controlador principal y a los módulos de displays.

5.1.1. Placa de la Interfaz del Controlador Principal

La Figura 5.1. muestra la placa del circuito impreso de la interfaz del controlador principal.

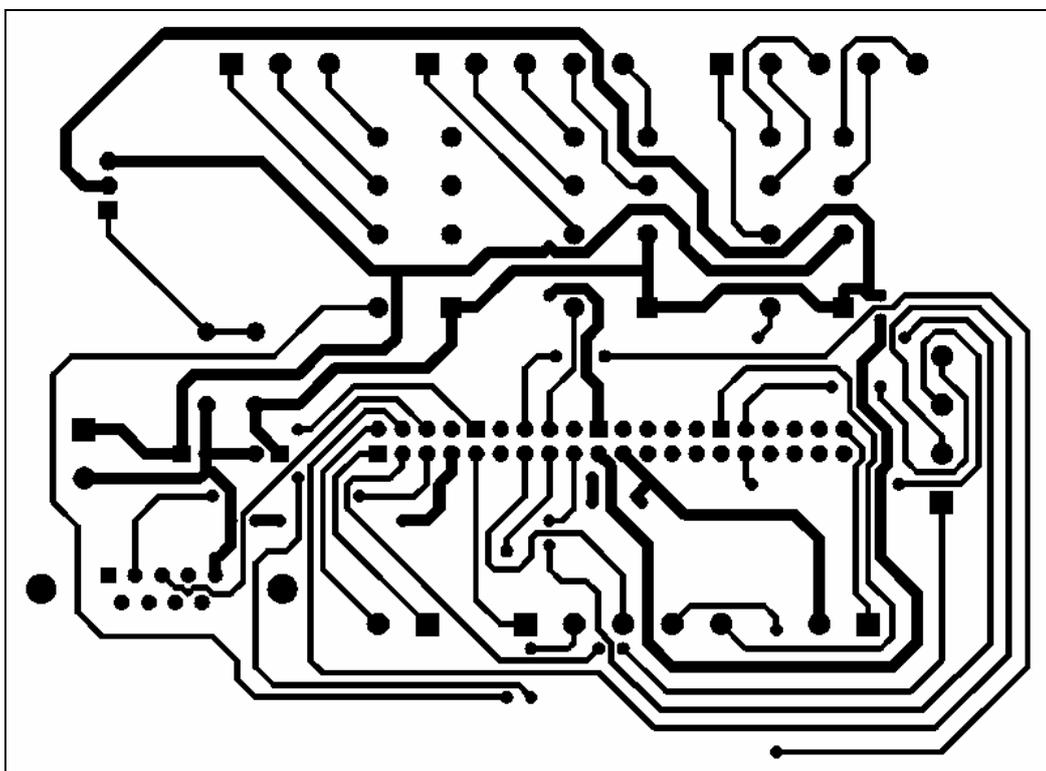


Figura 5.1. PCB de la Interfaz del Controlador Principal

¹ PCB: Printed Circuit Board (Placa de Circuito Impreso)

Adicionalmente, es necesario conocer la ubicación de los elementos en la placa de circuito impreso, lo cual se puede apreciar en la Figura 5.2.

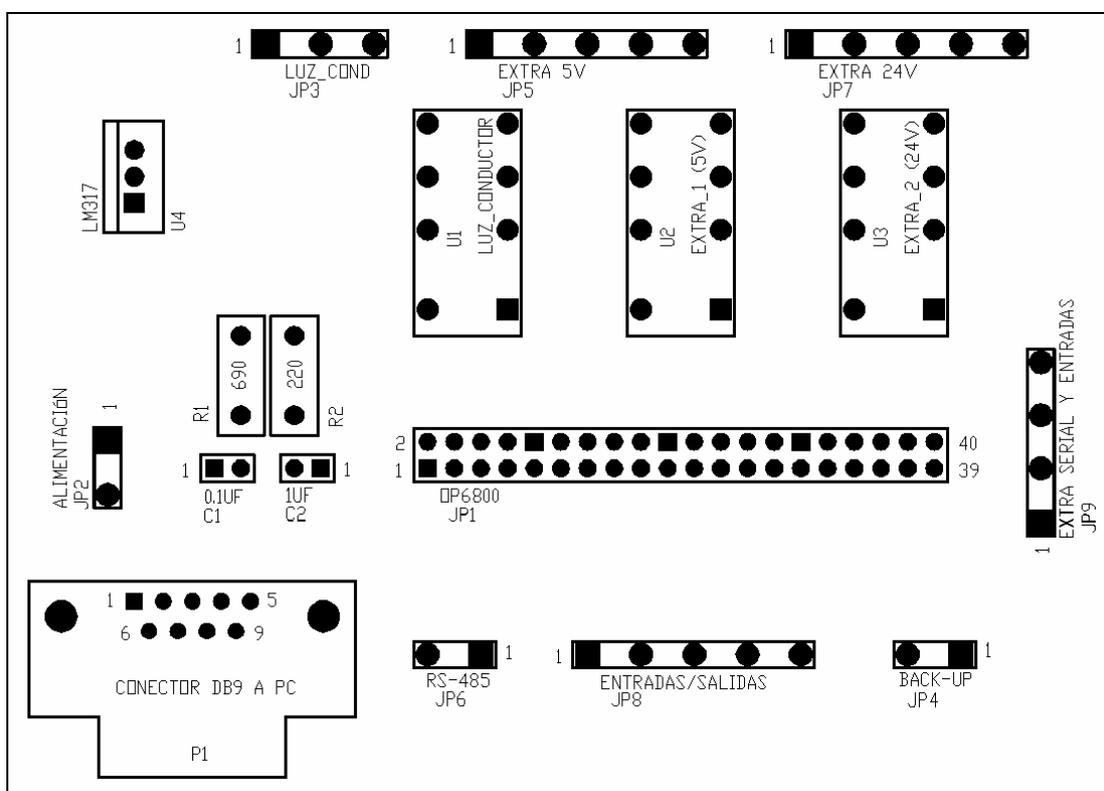


Figura 5.2. Ubicación de los Elementos en la PCB de la Interfaz del Controlador Principal

El detalle de las conexiones de los elementos puede ser apreciado en el Anexo 6, el cual presenta el diagrama esquemático de todo el sistema. Además, la Sección 5.1.2. muestra la lista de elementos de la interfaz del controlador principal.

5.1.2. Lista de Elementos de la Interfaz del Controlador Principal

La Tabla 5.1. detalla los elementos mostrados en la Figura 5.2., de manera que pueda ser interpretada toda la información acerca de la implementación de la PCB de la interfaz del controlador principal.

Tabla 5.1. Lista de Elementos de la Interfaz del Controlador Principal

Cant.	Descripción	Referencia	Valor
1	Capacitor	C1	0.1uF
1	Capacitor	C2	1uF
1	Jumper 50 pines	JP1	OP6800
1	Bornera 2 pines	JP2	ALIMENTACIÓN
1	Bornera 3 pines	JP3	LUZ_COND
1	Bornera 2 pines	JP4	BACK-UP
1	Bornera 5 pines	JP5	EXTRA 5V
1	Bornera 2 pines	JP6	RS-485
1	Bornera 5 pines	JP7	EXTRA 24V
1	Bornera 5 pines	JP8	ENTRADAS/SALIDAS
1	Bornera 4 pines	JP9	EXTRA SERIAL Y ENTRADAS
1	Conector DB-9 tipo macho	P1	CONECTOR DB9 A PC
1	Resistor	R1	690
1	Resistor	R2	220
1	Relé de 5V	U1	LUZ_CONDUCTOR
1	Relé extra de 5V	U2	EXTRA_1 (5V)
1	Relé extra de 5V	U3	EXTRA_2 (24V)
1	Regulador de voltaje variable	U4	LM317

5.1.3. Placa de los Módulos de Displays

El sistema está diseñado de forma tal que los tres módulos de displays deben ser construidos exactamente iguales. La Figura 5.3. muestra la placa del circuito impreso de los módulos de displays.

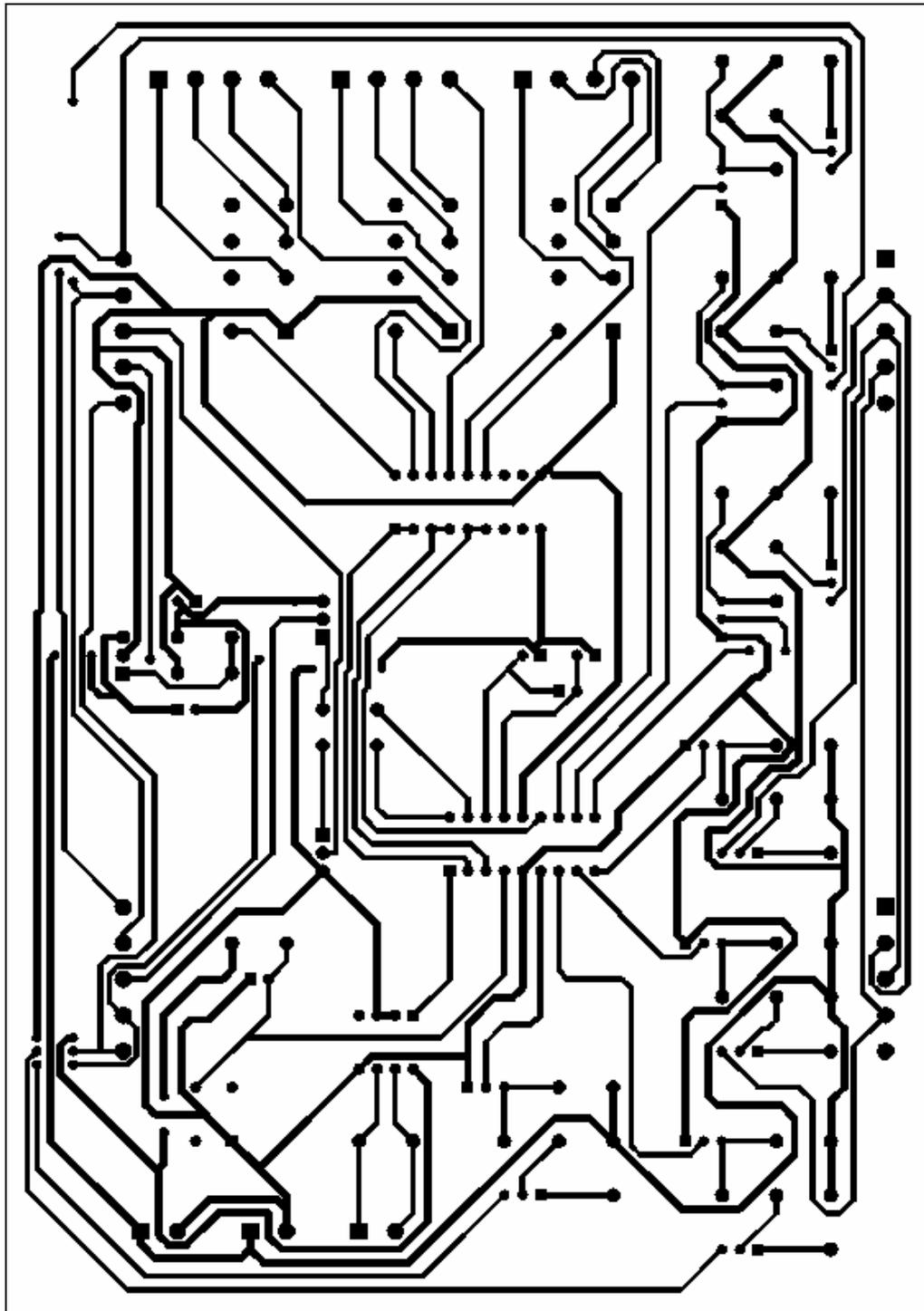


Figura 5.3. PCB de los Módulos de Displays

Al igual que en el caso anterior, es necesario conocer la ubicación de los elementos en la placa de circuito impreso, lo cual se puede apreciar en la Figura 5.4.

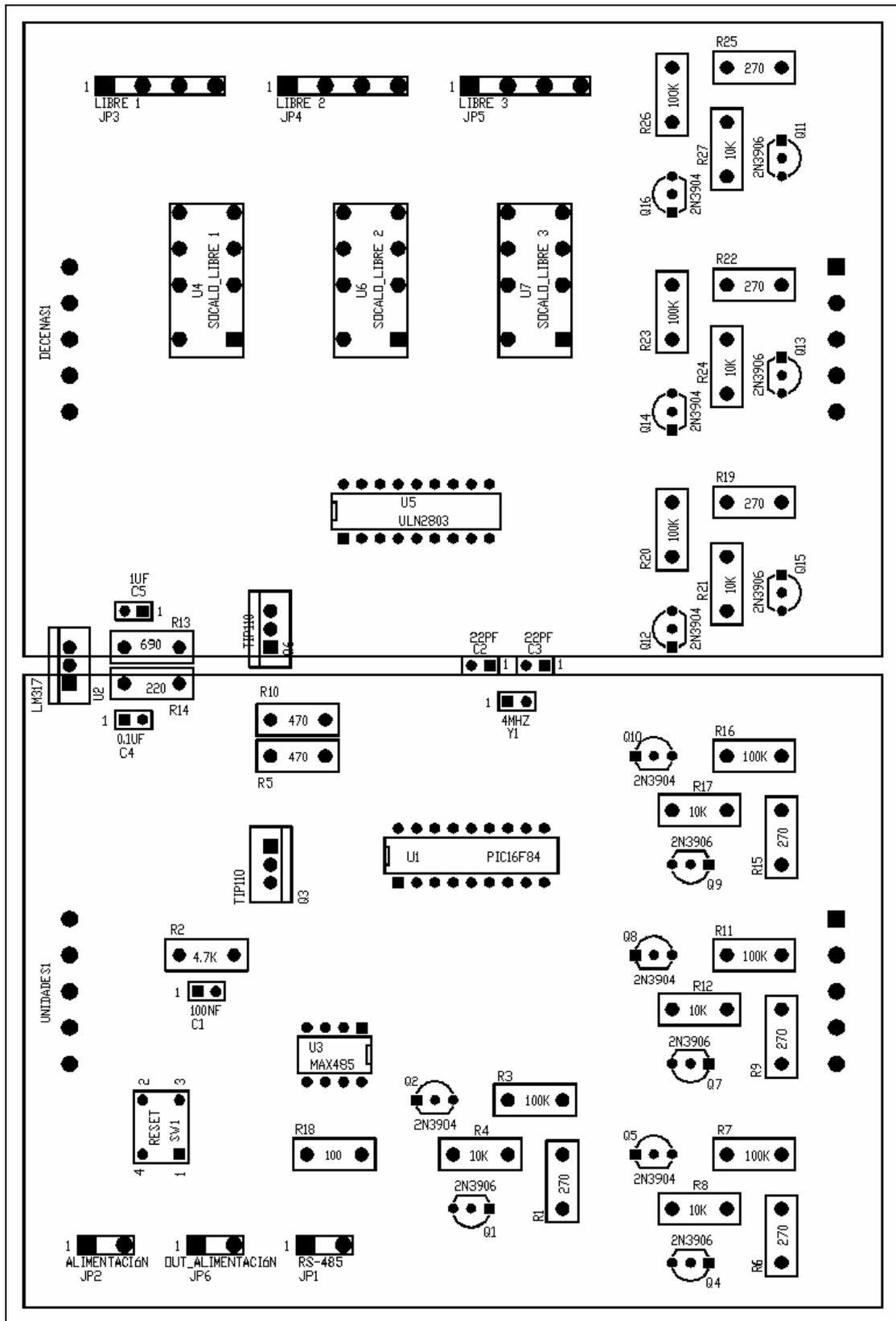


Figura 5.4. Ubicación de los Elementos en la PCB de los Módulos de Displays

El detalle de las conexiones de los elementos puede ser apreciado en el Anexo 6, el cual presenta el diagrama esquemático de todo el sistema. Además, la Sección 5.1.4. muestra la lista de elementos de cada módulo de displays.

5.1.4. Lista de Elementos de cada Módulo de Displays

La Tabla 5.2. detalla los elementos mostrados en la Figura 5.4., de manera que pueda ser interpretada toda la información acerca de la implementación de la PCB de los módulos de displays.

Tabla 5.2. Lista de Elementos de cada Módulo de Displays

Cant.	Descripción	Referencia	Valor
1	Capacitor	C1	100nF
2	Capacitor	C2,C3	22pF
1	Capacitor	C4	0.1uF
1	Capacitor	C5	1uF
2	Display de 4 pulgadas	UNIDADES1, DECENAS1	DISPLAY 4"
1	Bornera 2 pines	JP1	RS-485
1	Bornera 2 pines	JP2	ALIMENTACIÓN
1	Bornera 4 pines	JP3	LIBRE 1
1	Bornera 4 pines	JP4	LIBRE 2
1	Bornera 4 pines	JP5	LIBRE 3
1	Bornera 2 pines	JP6	OUT_ALIMENTACIÓN
7	Transistor PNP	Q1,Q4,Q7,Q9, Q11,Q13,Q15	2N3906
7	Transistor NPN	Q2,Q5,Q8,Q10, Q12,Q14,Q16	2N3904
2	Transistor Darlington NPN	Q6,Q3	TIP110
7	Resistor	R1,R6,R9,R15, R19,R22,R25	270
1	Resistor	R2	4.7K

7	Resistor	R3,R7,R11,R16, R20,R23,R26	100K
7	Resistor	R4,R8,R12,R17, R21,R24,R27	10K
9	Resistor	R5,R10,R28,R29, R30,R31,32,R33,R34	470
1	Resistor	R13	690
1	Resistor	R14	220
1	Resistor	R18	100
1	Pulsador NA	SW1	RESET
1	Microcontrolador	U1	PIC16F84
1	Regulador de voltaje variable	U2	LM317
1	Transceptor RS-485	U3	MAX485
1	Relé extra de 5V	U4	Socalo_LIBRE 1
1	Arreglo de transistores Darlington	U5	ULN2803
1	Relé extra de 5V	U6	Socalo_LIBRE 2
1	Relé extra de 5V	U7	Socalo_LIBRE 3
1	Cristal	Y1	20MHz

5.2. INTEGRACIÓN DEL HARDWARE

La Figura 5.5. muestra un diagrama unifilar correspondiente a las conexiones al sistema, este diagrama resume las conexiones a realizarse al momento de la implementación del sistema.

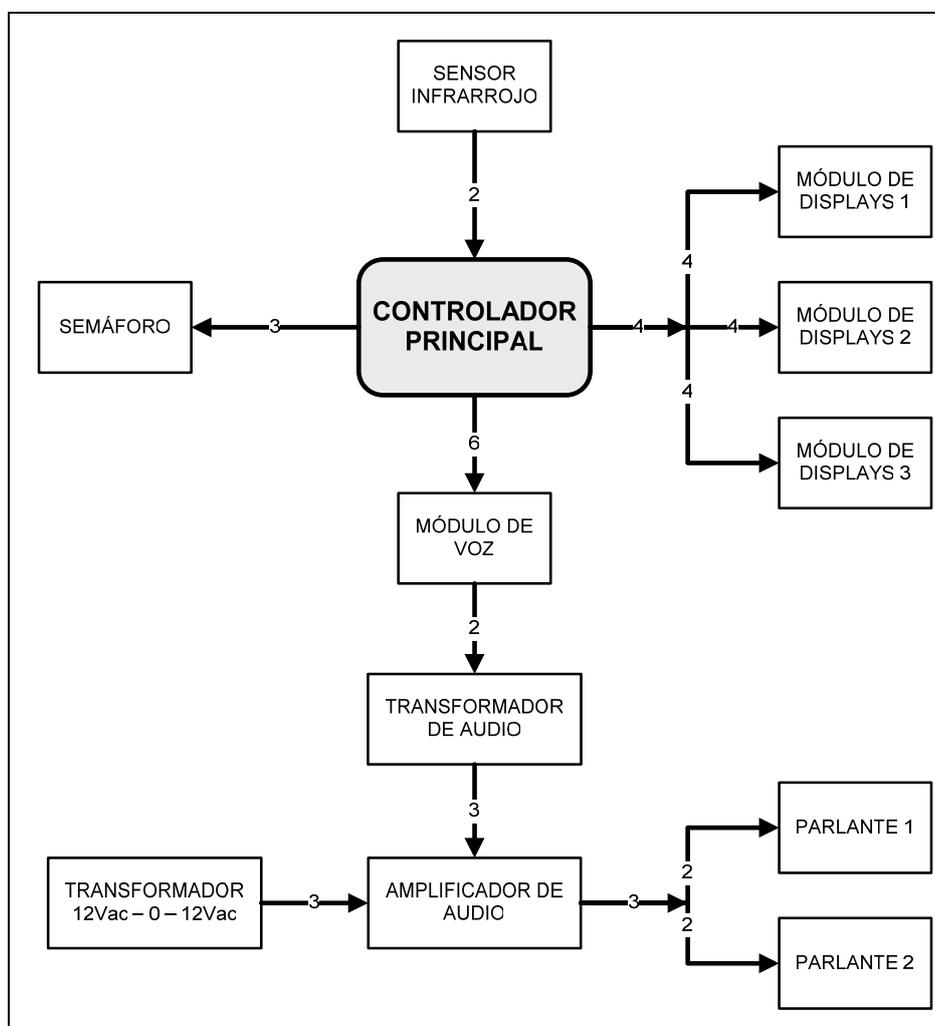


Figura 5.5. Diagrama Unifilar de Conexiones del Sistema

5.2.1. Interfaz del Controlador Principal:

El controlador principal, se encarga de enviar y recibir todas las señales de control necesarias para el correcto desempeño del sistema. La Figura 5.6. muestra la placa de interfaz del controlador principal, en dicha figura se encuentran etiquetadas las señales para la instalación del cableado.

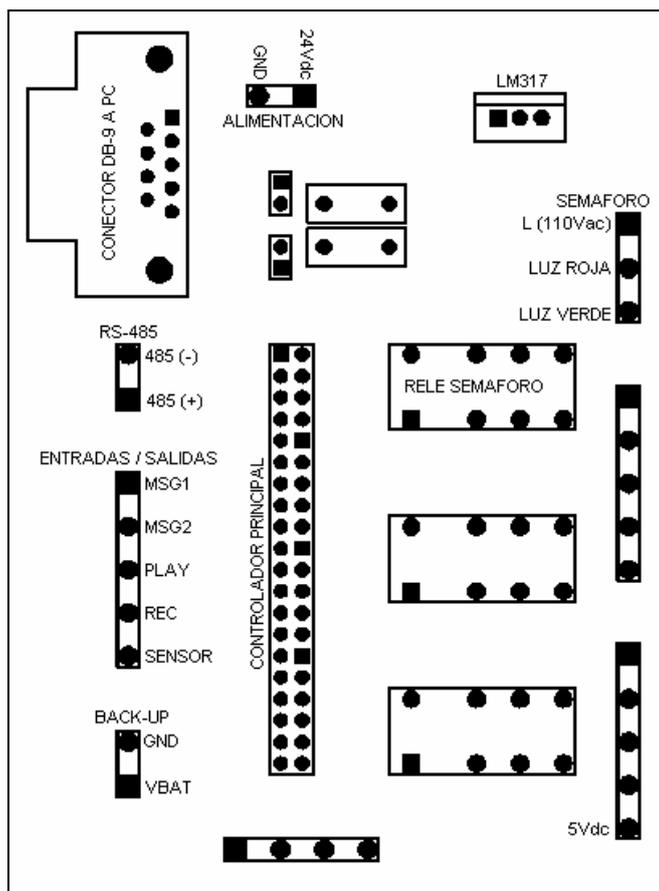


Figura 5.6. Placa de la Interfaz del Controlador Principal

- **CONECTOR DB-9 A PC.** Este es el conector para ubicar el cable serial (recto) para las operaciones de almacenamiento y respaldo de datos del controlador principal.
- **RS-485.** Son las señales asociadas a la red RS-485 utilizada para la presentación de datos en los displays.
 - **485(-).** Señal diferencial negativa de la red.
 - **485(+).** Señal diferencial positiva de la red.
- **ENTRADAS / SALIDAS.** Son las entradas y salidas necesarias para el control del sistema.
 - **MSG1.** Salida para la activación del mensaje de apertura de puertas.
 - **MSG2.** Salida para la activación del mensaje de cierre de puertas.
 - **PLAY.** Salida para la activación del modo de reproducción de mensajes de audio.

- **REC.** Salida para la activación del modo de grabación de mensajes de audio.
- **SENSOR.** Entrada de la señal de activación del sensor.
- **BACK-UP.** Es la entrada de voltaje de la batería de respaldo o back-up.
 - **GND.** Tierra del sistema (negativo de la batería)
 - **VBAT.** Voltaje de la batería de respaldo (positivo de la batería).
- **ALIMENTACIÓN.** Es la entrada de voltaje necesaria para la operación del controlador principal.
 - **24Vdc.** Entrada de 24 V_{DC}.
 - **GND.** Tierra del sistema.
- **CONTROLADOR PRINCIPAL.** Es el conector designado a la entrada de los pines del controlador principal.
- **SEMÁFORO.** Son las señales de salida para la conexión del semáforo.
 - **L (110Vac).** Es la fase de señal de 110Vac (El neutro llega hasta donde esté ubicado el semáforo).
 - **LUZ ROJA.** Señal de activación de la luz roja del semáforo.
 - **LUZ VERDE.** Señal de activación de la luz verde del semáforo.
- **5Vdc.** Es una salida de voltaje que alimentará al módulo que se encarga de almacenar los mensajes de voz.

5.2.2. Sensor Infrarrojo:

El sensor infrarrojo es el encargado de detectar la presencia del trolebús en la estación. Este sensor el mismo utilizado para abrir las puertas del andén, por lo que ya se encuentra alimentado y simplemente se debe tomar su señal de activación y enviarla al controlador principal.

Como se puede apreciar en la Figura 5.5., se deben conectar 2 cables entre el sensor y el controlador principal:

- Señal de activación del sensor (Borne No. 11 en la Figura 5.7.)
- Señal de Tierra (Borne No. 9 en la Figura 5.7.)

La Figura 5.7. muestra la ubicación de estas señales en el controlador de una de las puertas del andén. Dichas señales pueden ser tomadas de cualquier puerta.



Figura 5.7. Señales del Sensor Ubicadas en el Controlador de una Puerta del Andén

5.2.3. Módulos de Displays:

Los módulos de displays están destinados a indicar a los usuarios del Sistema Trolebús el tiempo que permanecerán abiertas las puertas del trolebús estacionado frente al andén.

Como se puede apreciar en la Figura 5.5., se necesitan 4 cables de conexión entre cada módulo de displays y el controlador principal, cabe recalcar que los 4 cables son los mismos para cada módulo y son los siguientes:

- Voltaje de alimentación (24 Vdc).
- Tierra del sistema (GND).
- Señal diferencial positiva de la red RS-485 (485(+)).
- Señal diferencial negativa de la red RS-485 (485(-)).

La Figura 5.8. muestra la ubicación de estas señales en el módulo de displays.

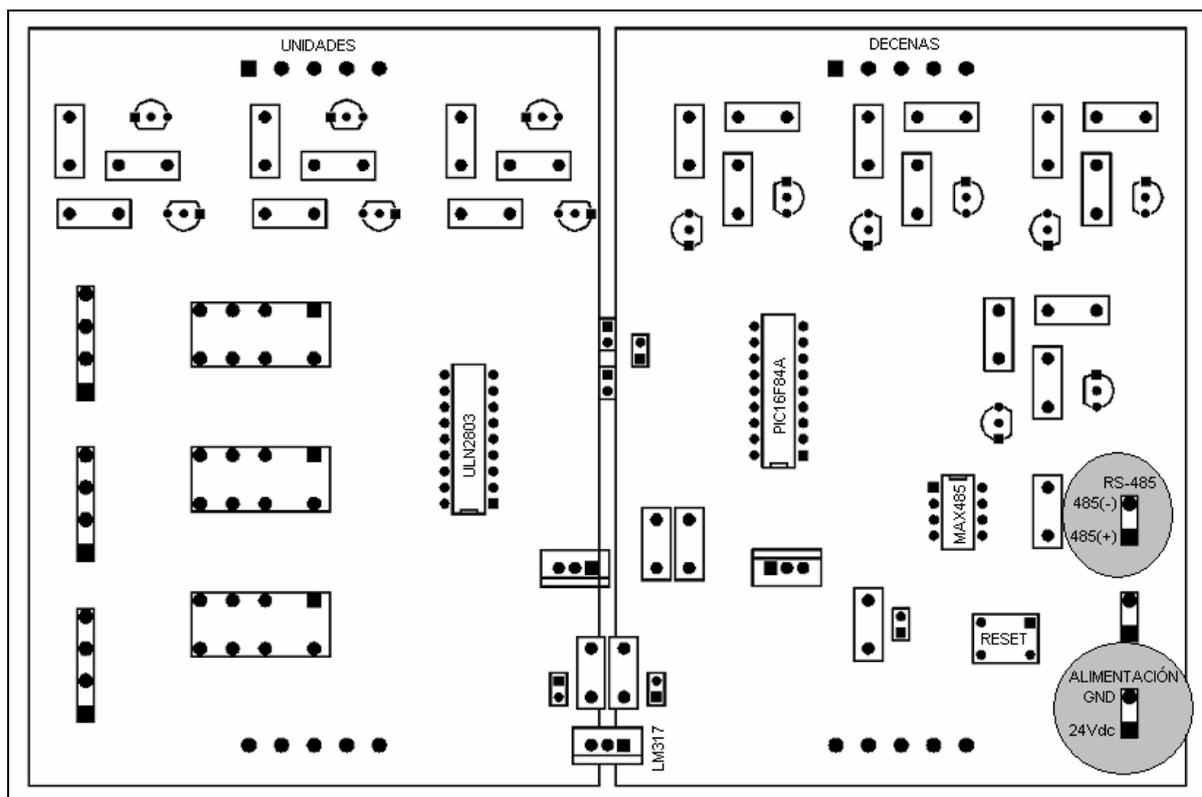


Figura 5.8. Conectores de Alimentación y RS-485 en la Placa del Módulo de Displays

5.2.4. Semáforo:

El semáforo constituye la señal de alerta visual para el conductor del trolebús, para mostrarle que ha transcurrido su tiempo de estadía en la parada y que puede cerrar las puertas y continuar su recorrido.

Como se puede apreciar en la Figura 5.5., se necesitan 3 cables para la conexión del semáforo, son los siguientes:

- Luz roja.
- Luz verde.
- Neutro de la alimentación de 110VAC.

Deben realizarse las conexiones del semáforo de acuerdo a lo indicado en la Figura 5.9.

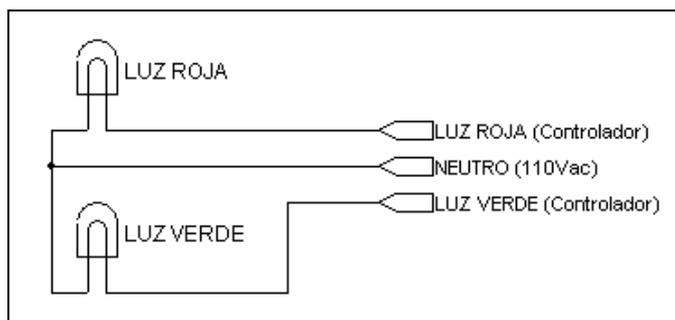


Figura 5.9. Esquema de la Conexión del Semáforo.

Las señales mostradas en la Figura 5.9. son:

- LUZ ROJA (Controlador). Señal de activación de la luz roja, proveniente del controlador principal.
- NEUTRO (110Vac). Neutro de la alimentación de los 110VAC, la fase se encuentra en el controlador principal y es éste último el encargado de cerrar el circuito de activación de las luces del semáforo.
- LUZ VERDE (Controlador). Señal de activación de la luz verde, proveniente del controlador principal.

5.2.5. Módulo de Voz:

El módulo de voz es el que almacena los mensajes de advertencia que dispone el sistema, es decir, el mensaje de apertura de puertas y el de cierre de puertas.

Como se puede apreciar en la Figura 5.5., se necesitan 6 cables para la conexión del módulo de voz con el controlador principal, son los siguientes:

- Activación del mensaje 1 (MSG1).
- Activación del mensaje 2 (MSG2).
- Modo de reproducción de mensajes (PLAY).
- Modo de grabación de los mensajes (REC).
- Voltaje de alimentación del módulo (5Vdc).
- Tierra del sistema (GND).

La Figura 5.10. muestra la ubicación de los conectores de estas señales en el módulo voz.

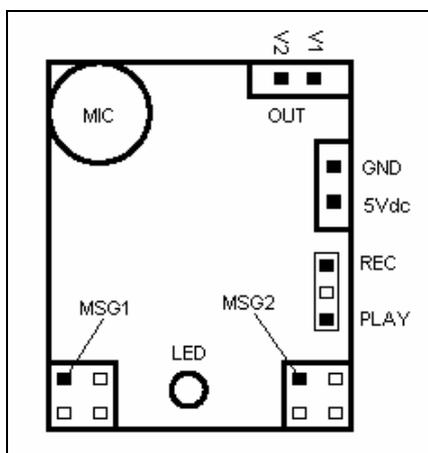


Figura 5.10. Conectores en la Placa del Módulo de Voz

Adicionalmente, la salida del módulo de voz (OUT, en la Figura 5.10.) debe ser conectada a un Transformador de Audio, para poder acoplar correctamente el módulo de voz con el amplificador de audio. La Figura 5.11. muestra el circuito de acoplamiento.



Figura 5.11. Acoplamiento entre el Módulo de Voz y el Amplificador de Audio

Las señales mostradas en la Figura 5.11. son las siguientes:

- **V1.** Salida 1 del módulo de voz.
- **GND.** Tierra del sistema.
- **V2.** Salida 2 del módulo de voz.
- **CH1.** Salida de audio 1.
- **CH2.** Salida de audio 2.

Adicionalmente, se recomienda reubicar el micrófono (MIC, en la Figura 5.10.) en un lugar accesible para el momento de realizar la grabación de los mensajes de voz.

5.2.6. Amplificador de Audio:

El amplificador de audio se encarga de proporcionar una señal audible a los usuarios del Sistema Trolebús. Como se puede apreciar en la Figura 5.5., el amplificador tiene 6 entradas y 3 salidas, las cuales son:

- **VA.** Voltaje de entrada (negativo).
- **GND.** Tierra del sistema (fuente de alimentación).
- **VB.** Voltaje de entrada (positivo).
- **L-IN.** Entrada de audio al canal 1.
- **GND.** Tierra del sistema (entrada de audio).
- **R-IN.** Entrada de audio al canal 2.
- **LS-L.** Salida 1 del amplificador.
- **LS-R.** Salida 2 del amplificador.

La Figura 5.12. muestra las conexiones a realizarse en el amplificador de audio.

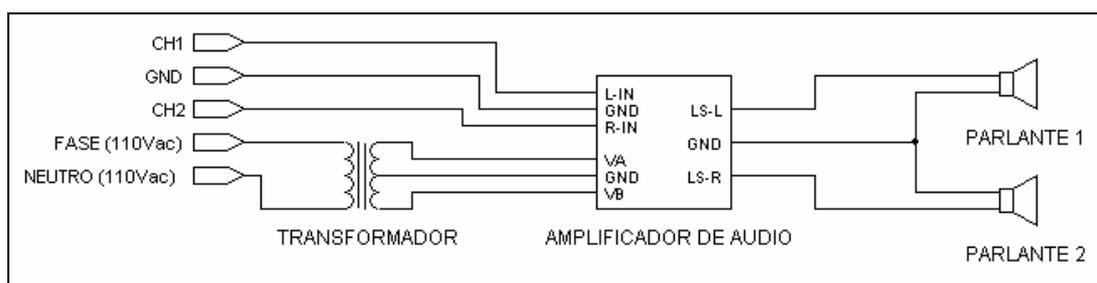


Figura 5.12. Conexiones del Amplificador de Audio.

Las etiquetas de las entradas adicionales corresponden a:

- **CH1.** Salida 1 del transformador de audio.
- **GND.** Tierra del sistema.
- **CH2.** Salida 2 del transformador de audio.

- **FASE (110Vac).** Entrada de voltaje al transformador, para la fuente del amplificador.
- **NEUTRO (110Vac).** Entrada de voltaje al transformador, para la fuente del amplificador.

Los parlantes serán conectados en las salidas del amplificador, es decir en las señales **LS-L** y **LS-R**, además de conectar el cátodo de los parlantes a la tierra del sistema.

5.3. IMPLEMENTACIÓN

El sistema ha sido implementado en el andén de la estación “El Calzado”, en sentido sur-norte. Para la implementación se siguió los procedimientos mostrados en este capítulo y a continuación se presentan algunas fotografías del sistema.



Figura 5.13. Fotografía de la Caja de Control



Figura 5.14. Fotografía del Módulo de Displays Instalado sobre la Segunda Puerta



Figura 5.15. Vista del Andén "El Calzado" con el Prototipo Instalado

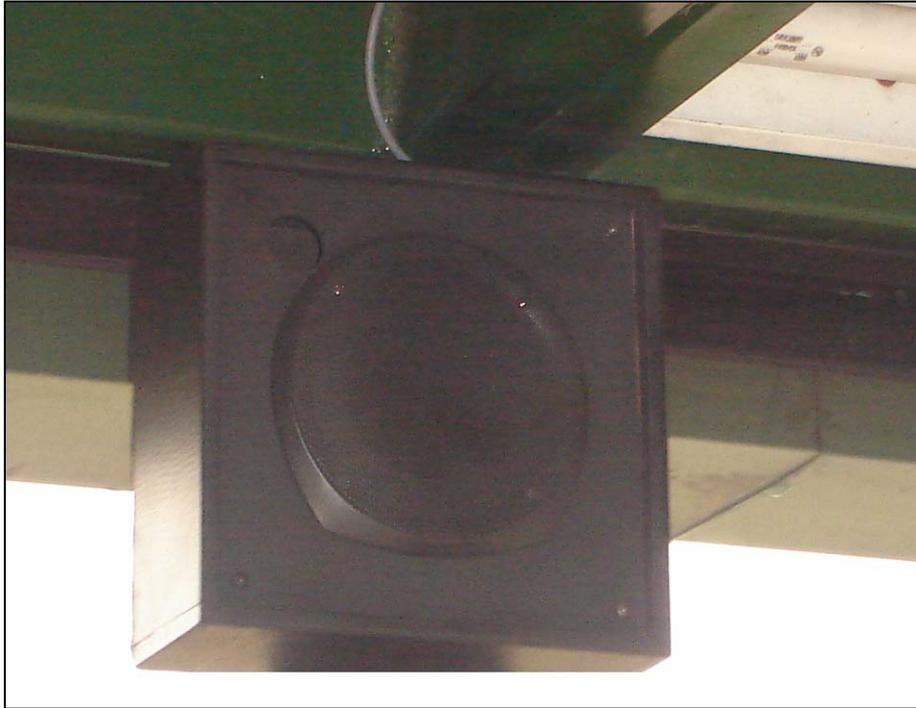


Figura 5.16. Fotografía de uno de los Parlantes que Proporcionan la Alerta Audible



Figura 5.17. Fotografía del Semáforo ubicado Fuera del Andén, Visible al Conductor

CAPÍTULO 6

PRUEBAS Y RESULTADOS

6.1. CONDICIONES DE PRUEBA

El sistema se encuentra instalado en el andén “El Calzado” de la Unidad Operadora del Sistema Trolebús y se encuentra a prueba desde el día 15 de diciembre de 2006.

Para realizar las pruebas de funcionamiento, el sistema ha sido configurado y conectado bajo las siguientes condiciones:

- Voltaje de alimentación: 110V_{AC} / 60Hz.
- Propiedades de la comunicación serial con el computador:
 - Estándar: RS-232
 - Velocidad de comunicación: 9600 baudios
 - Bits de datos: 8
 - Bits de parada: 1
 - Bits de paridad: Ninguna
- Propiedades de la comunicación serial con los módulos de displays:
 - Estándar: RS-485
 - Velocidad de comunicación: 9600 baudios
 - Bits de datos: 8
 - Bits de parada: 1
 - Bits de paridad: Ninguna
- Volumen del amplificador al máximo.
- Semáforo encendido.

6.2. PRUEBAS Y RESULTADOS

6.2.1. Pruebas de Hardware

Las pruebas de hardware están realizadas en base al consumo de corriente del sistema sometido a varias condiciones. Los resultados a estas pruebas pueden apreciarse en la Tabla 6.1.

Tabla 6.1. Pruebas y Resultados de Hardware

Pruebas	Resultados
1. Sistema energizado.	0.70 A
2. Sistema energizado, semáforo encendido.	1.00 A
3. Sistema energizado, semáforo encendido, reproduciendo un mensaje de voz.	1.60 A
4. Sistema energizado, semáforo encendido, reproduciendo un mensaje de voz, trolebús presente.	2.75 A

6.2.1.1. Prueba de Hardware 1. Esta prueba consistió en medir la corriente que entrega la fuente de alimentación del sistema, cuando éste se encuentra encendido, pero está apagado el semáforo, no se está reproduciendo ningún mensaje de voz y no se detecta la presencia de un trolebús frente al andén. Este valor muestra el consumo de corriente mínimo del sistema y es un valor coherente con el dimensionamiento de la fuente de alimentación (4.5A).

6.2.1.2. Prueba de Hardware 2. Esta prueba consistió en medir la corriente que entrega la fuente de alimentación del sistema, cuando éste y el semáforo están encendidos, pero no se está reproduciendo ningún mensaje de voz y no se detecta la presencia de un trolebús frente al andén. Este valor muestra el consumo de corriente en estado de espera, ya que al encender estos dos elementos, el sistema está listo para operar con todas sus características. El valor obtenido presenta coherencia con el dimensionamiento de la fuente de alimentación.

6.2.1.3. Prueba de Hardware 3. Esta prueba consistió en medir la corriente que entrega la fuente de alimentación del sistema, cuando éste y el semáforo están encendidos, se está reproduciendo un mensaje de voz, pero no se detecta la presencia de un trolebús frente al andén. Este valor muestra el consumo de corriente del sistema sin la presencia del trolebús. Al igual que en los casos anteriores, el valor obtenido presenta coherencia con el dimensionamiento de la fuente de alimentación.

6.2.1.4. Prueba de Hardware 4. Esta prueba consistió en medir la corriente que entrega la fuente de alimentación del sistema, cuando éste y el semáforo están encendidos, se está reproduciendo ningún mensaje de voz y se detecta la presencia de un trolebús frente al andén, es decir, en el caso en que el sistema demanda el máximo consumo de corriente. El valor obtenido confirma el hecho que el dimensionamiento de la fuente de alimentación fue realizado en forma adecuada.

6.2.2. Pruebas de Software

Las pruebas de software están realizadas en base al tiempo que toman las operaciones de respaldo y de descarga de datos al controlador principal, resultados que se pueden apreciar en la Tabla 6.2.

Tabla 6.2. Pruebas y Resultados de Software

Pruebas	Resultados
1. Tiempo de descarga con datos por defecto.	17 s
2. Tiempo de descarga de 365 días personalizados.	4 min, 45 s
3. Tiempo de respaldo con datos por defecto.	14 s
4. Tiempo de respaldo de 365 días personalizados.	4 min

6.2.2.1. Prueba de Software 1. Esta prueba consistió en medir el tiempo transcurrido al descargar los parámetros de funcionamiento por defecto, desde el computador hacia el controlador principal. Se obtuvo un valor

satisfactorio, debido a que el medio de transmisión de datos cumple con el estándar RS-232.

6.2.2.2. Prueba de Software 2. Esta prueba consistió en medir el tiempo transcurrido al descargar los parámetros de funcionamiento del sistema, con los 365 días del año personalizados, desde el computador hacia el controlador principal. Se obtuvo un valor satisfactorio, ya que el caso de personalizar 365 días es remoto, pero es una capacidad que presenta el sistema.

6.2.2.3. Prueba de Software 3. Esta prueba consistió en medir el tiempo transcurrido al respaldar los parámetros de funcionamiento por defecto, desde el controlador principal hacia el computador. Se obtuvo un valor satisfactorio y coherente con el valor obtenido en la prueba de software 1.

6.2.2.4. Prueba de Software 4. Esta prueba consistió en medir el tiempo transcurrido al respaldar los parámetros de funcionamiento del sistema, con los 365 días del año personalizados, desde el controlador principal hacia el computador. Se obtuvo un valor satisfactorio y coherente con el valor obtenido en la prueba de software 2.

CAPÍTULO 7

ANÁLISIS ECONÓMICO

7.1. PRESUPUESTO

El propósito de mostrar el presupuesto necesario para el desarrollo del prototipo del Sistema de Alerta para el Cierre de Puertas de las Unidades del Sistema Trolebús es reflejar el hecho de que el sistema puede ser desarrollado con ingeniería nacional y a un precio razonable. La Tabla 7.1. muestra el presupuesto referencial para el desarrollo del prototipo.

Tabla 7.1. Presupuesto para el Desarrollo del Prototipo

Cantidad	Ítem	Cantidad	Elemento	Detalle / Marca	Valor Unitario	Valor Total	Suma Total
1	Caja de Control	1	Amplificador de Audio	Velleman	45.30	45.30	792.80
		5	Bornera	2 terminales	0.45	2.25	
		2	Bornera	3 terminales	0.70	1.40	
		1	Caja Controlador Principal	Metálica	15.00	15.00	
		2	Capacitor	1uF	0.10	0.20	
		1	Circuito Integrado	LM317	1.60	1.60	
		1	Conector para placa	DB-9 macho	0.40	0.40	
		1	Fuente de Poder	24V _{DC} - 5A	68.30	68.30	
		2	Header	1 fila	0.35	0.70	
		2	Interruptores ON/OFF	Tipo palanca	0.35	0.70	
		2	Módulo Controlador	Rabbit	295.20	590.40	
		2	Módulo de Voz	2x20 segundos	19.50	39.00	

	1	Perilla	Potenciómetro	0.50	0.50		
	1	Placa Interfaz Controlador Principal	--	16.00	16.00		
	1	Potenciómetro	Estéreo (tandem)	1.20	1.20		
	1	Relé	Bobina a 5V	1.70	1.70		
	2	Resistencia	1/2W	0.03	0.06		
	1	Transformador	Salida: 12 – 0 – 12 V _{AC}	5.00	5.00		
	1	Transformador de Audio	--	3.00	3.00		
	1	Zócalo DIP	16 pines	0.09	0.09		
3							
3	Caja de Displays	3	Caja Display	En Acrílico	50.00	150.00	437.73
		6	Bornera	2 terminales	0.45	2.70	
		6	Capacitor	22pF	0.10	0.60	
		3	Capacitor	100nF	0.10	0.30	
		6	Capacitor	1uF	0.10	0.60	
		4	Circuito Integrado	MAX485	4.90	19.60	
		4	Circuito Integrado	PIC16F84A	5.90	23.60	
		3	Circuito Integrado	LM317	1.60	4.80	
		3	Cristal	20MHz	1.20	3.60	
		6	Display de 7 Segmentos	4 pulgadas	18.20	109.20	
		3	Placa Display	--	35.00	105.00	
		3	Pulsador	NA	0.15	0.45	
		21	Resistencia	1W	0.09	1.89	
		102	Resistencia	1/2W	0.03	3.06	
		21	Transistor	2N3904	0.22	4.62	
		21	Transistor	2N3906	0.20	4.20	
		6	Transistor	TIP110	0.50	3.00	
		3	Zócalo DIP	8 pines	0.06	0.18	
3	Zócalo DIP	18 pines	0.11	0.33			

1	Semáforo	1	Caja Semáforo	En aluminio	15.00	15.00	30.00
		1	Materiales	2 focos, 2 boquillas, 2 lunas (roja y verde)	15.00	15.00	
1	Adicionales	2	Barra de Silicón	--	0.24	0.48	91.91
		1	Cable de Programación	Rabbit	35.20	35.20	
		2	Caja Acústica	6 pulgadas	7.00	14.00	
		1	Cinta Aislante	Tape	0.40	0.40	
		2	Conector con Tapa	DB-9 hembra	0.75	1.50	
		1	Lija de Agua	No 180	0.23	0.23	
		2	Lunitas Acrílico	1.35x0.34 mm	3.00	6.00	
		1	Material menudo	--	10.00	10.00	
		2	Metro Cable 3 hilos	AWG18	0.38	0.76	
		2	Parlante	100W-4Ω, 6"	7.60	15.20	
		1	Pegamento	Brujita	1.55	1.55	
		1	Pintura en Spray	Plateado	4.24	4.24	
		1	Pintura en Spray	Negro Mate	1.42	1.42	
		80	Remache POP	1/8 x 5/16	0.00	0.38	
		14	Remache POP	3/16 x 1	0.04	0.55	
1	Instalación	70	Amarra plástica	10cm	0.00	0.21	111.71
		10	Amarra plástica	40cm	0.03	0.25	
		15	Colepatos	--	0.05	0.75	
		1	Espiral	--	4.70	4.70	
		1	Metro Cable de Audio	--	1.40	1.40	
		2	Metro Cable Gemelo	AWG14	0.80	1.60	
		10	Metro Cable Gemelo	AWG18	0.37	3.70	
		30	Metro Cable Multipar	4 pares	0.86	25.80	
		30	Metro de Cable Sucre	3x14	2.08	62.40	
		3	Metro perfil de Aluminio	15x3 mm	2.70	8.10	

		1	Metro perfil de Aluminio para soporte	--	2.80	2.80	
					TOTAL		1464.15

7.2. ANÁLISIS

Debe tomarse en cuenta que el sistema desarrollado es un prototipo, por lo que existen elementos adicionales que se adquirió por daño, pérdida o respaldo de éstos. Además, existen ciertos componentes del sistema que deben ser adquiridos una sola vez y que ya se los dispone.

Estos hechos hacen que el prototipo tenga un costo de USD\$ 1464.15. En caso de que se decida reproducir el sistema, el costo se reducirá a aproximadamente USD\$ 1100, considerando únicamente los materiales estrictamente necesarios.

Cabe señalar que los elementos fueron adquiridos en el mes de **noviembre de 2006**, por lo que los costos corresponden a esta fecha y deberán considerarse posibles reajustes en los costos para futuras implementaciones.

Cabe señalar también que por tratarse de un prototipo desarrollado bajo el esquema de un proyecto de grado no se ha considerado ningún costo referente al desarrollo de ingeniería.

Finalmente, no se realiza estrictamente un análisis económico, en vista que la **rentabilidad** del proyecto no es palpable económicamente, debido a que éste está enfocado a elevar el nivel de satisfacción de los usuarios del Sistema Trolebús, al proporcionar un mejor servicio. Esto no produce un nivel de utilidad económica directa a la UOST, pero se considera que “un cliente satisfecho siempre regresa”, lo que al mediano y largo plazo incrementará su utilidad económica y el nivel de la calidad de su servicio.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Se ha logrado diseñar e implementar un sistema prototipo para la UOST, el cual cumple con los requerimientos de operación establecidos al inicio del proyecto, dicho sistema permite alertar a los usuarios dentro del andén sobre el cierre de las puertas del trolebús, en forma audible y visible.
- Se ha diseñado el sistema de forma que el controlador principal sea el encargado de centralizar y administrar el flujo de información necesaria para la operación del sistema, gracias a la disponibilidad en su hardware de entradas y salidas, además de un teclado y un display que permiten la manipulación del software cargado en él.
- Se ha desarrollado la programación del controlador principal mediante multitareas, lo que permite el desempeño del sistema en tiempo real.
- El sistema diseñado incluye el software necesario para las operaciones de respaldo y descarga de la información empleada para la operación del sistema, mediante un computador personal.
- Se ha dotado al sistema de autonomía mediante el empleo del RTC incluido en el controlador principal, además de la capacidad de manipular la tabla de temporizaciones anual almacenada en la memoria del controlador.
- El sistema diseñado presenta la capacidad de respaldar las variables críticas para la operación del sistema, mediante la memoria SRAM incluida en el controlador principal, siendo necesario para esto una batería de respaldo.

- Se ha utilizado el estándar RS-232, debido a que presenta las características adecuadas para la conexión entre el controlador principal y un computador, debido a sus características de conexión directa con un computador y a su distancia de conexión, 12m a 20Kbps.
- Se ha utilizado el estándar RS-485, debido a que presenta las características apropiadas para la conexión entre el controlador principal y los módulos de displays, ya que permite la transmisión de datos fiable a velocidades y distancias relativamente altas, 100Kbps a 1200m.
- Se han diseñado los módulos de displays para funcionar dentro de una red RS-485, donde el controlador principal es el encargado de llevar la cuenta descendente y el microcontrolador PIC16F84A es el encargado de presentar dicha cuenta mediante barrido.
- El módulo de voz empleado proporciona una solución eficaz y económica para cumplir con los requerimientos del sistema, en lo concerniente a la alerta audible.
- El diseño del sistema incluye un transformador de audio, el cual permite referenciar adecuadamente la señal de audio diferencial proporcionada por el módulo de voz, para que sea aprovechada toda la amplitud de dicha señal.
- La manipulación de los parámetros de funcionamiento en forma directa mediante el controlador principal, le brinda al sistema un alto nivel de flexibilidad en su operación.
- El análisis económico realizado muestra que los requerimientos expuestos por la UOST pueden ser satisfechos mediante ingeniería nacional a un costo razonable.

RECOMENDACIONES

- Se recomienda revisar el estado de la batería de respaldo en un período menor a 3 años, debido a que ese es el tiempo que recomienda el fabricante.

- Se debe considerar que en caso de perderse la energía cuando la batería de respaldo se ha agotado, al restablecerse la energía, el sistema reiniciará su funcionamiento con valores por defecto y el RTC tendrá información incorrecta, siendo recomendable realizar las correcciones pertinentes en los parámetros de operación del sistema.

- Para las operaciones de almacenamiento y respaldo de datos mediante el computador personal, se recomienda utilizar un cable de 12 metros como máximo, ya que el estándar RS-232 no soporta distancias mayores.

- Se recomienda utilizar un par trenzado para la implementación de la red bajo el estándar RS-485, ya que este es un requerimiento realizado por la EIA.

- Al realizar cambios en la operación del sistema, mediante el hardware del controlador principal, debe considerarse que esta información ya no es coherente con el archivo desarrollado en el software para el computador.

- Se recomienda verificar el correcto estado y operación del sensor infrarrojo instalado en el andén, ya que este es un elemento adicional en la implementación del prototipo y es fundamental en la operación del mismo.

REFERENCIAS BIBLIOGRÁFICAS

- PALACIOS, Enrique – REMIRO, Fernando – LÓPEZ, Lucas, *Microcontrolador PIC16F84 Desarrollo de Proyectos*, Segunda Edición, Editorial Alfa-Omega, Perú, 2006.
- GRANIZO, Evelio, *Programación Orientada a Objetos en C++ Teoría y Ejercicios*, Editorial ESPE, primera edición, Quito-Ecuador, 1997.
- CASTAÑEDA, Juan – SABANA, Maribel, *Visual Basic 2005 Express Como debe de ser*, Editorial Megabyte, primera edición, Lima-Perú, 2005.
- SERRANO, Jorge, *Manual de Introducción a Microsoft® Visual Basic® 2005 Express Edition*, 2006.
- HORROCKS, Kris – CAMPBELL Sea – HATCHARD Derek – BERNHARDT Peter – SWIGART Scott, *Introducing Visual Basic 2005 for Developers*, primera edición, Microsoft, 2005.
- RABBIT SEMICONDUCTOR, *Rabbit 2000® Microprocessor User's Manual*, Estados Unidos de América, 1999.
- Z-WORLD, *MiniCom (OP6810) C-Programmable Operator Interface User's Manual*, Estados Unidos de América, 2002.
- Z-WORLD, *Dynamic C User's Manual*, Estados Unidos de América, 2004.
- Microsoft MSDN 2005 Express Edition, <http://www.msdn.com>
- PIC16F84, <http://es.wikipedia.org/wiki/PIC16F84A>

-
- Perro Guardián, http://es.wikipedia.org/wiki/Perro_guardi%C3%A1n
 - CPU, <http://es.wikipedia.org/wiki/CPU>
 - RISC, <http://es.wikipedia.org/wiki/RISC>
 - Arquitectura Harvard, http://es.wikipedia.org/wiki/Arquitectura_Harvard
 - Comunicación Síncrona,
<http://mx.geocities.com/pcmuseo/mecatronica/puertoserialsincrono.htm>
 - H, <http://es.tldp.org/Otros/diccionario-us-es/diccionario-us-es-0.1.6/H.html>
 - Comunicación Asíncrona,
<http://mx.geocities.com/pcmuseo/mecatronica/puertoserialasincrono.htm>
 - RS-232, <http://es.wikipedia.org/wiki/RS-232>
 - EIA-485, <http://en.wikipedia.org/wiki/RS-485>
 - Base de Datos, http://es.wikipedia.org/wiki/Base_de_datos
 - Manejo de Archivos,
http://www.investigacion.frc.utn.edu.ar/labsis/Publicaciones/apunte_linux/ma.html

ANEXO 1

MANUAL DEL USUARIO

**PROTOTIPO DEL SISTEMA DE ALERTA
PARA EL CIERRE DE PUERTAS DE LAS
UNIDADES DEL SISTEMA TROLEBÚS**

MANUAL DEL USUARIO

**ESCUELA POLITÉCNICA DEL EJÉRCITO – 2007.
PROYECTO DE GRADO.**

TABLA DE CONTENIDOS:

1. PRESENTACIÓN:	3
2. IMPORTANTE:	3
3. ADVERTENCIA:	3
4. ALIMENTACIÓN DEL SISTEMA:	3
5. CARACTERÍSTICAS:	3
6. MANIPULACIÓN DEL SOFTWARE DEL CONTROLADOR PRINCIPAL:	4
6.1. Pantalla Inicial.....	5
6.2. Contraseña de Ingreso.....	5
6.3. Menú Principal	6
6.3.1. Modificar Horario.....	6
6.3.2. Modificar Hora Actual.....	6
6.3.3. Luz de Fondo.	6
6.3.4. Mensajes de Voz.....	6
6.3.5. Actualizar Hora.....	7
6.3.6. Acerca de.....	7
7. MANIPULACIÓN DEL SOFTWARE ADMINISTRATIVO:	8
7.1. Menú Archivo	9
7.1.1. Abrir Archivo Existente	10
7.1.2. Nuevo Archivo.....	10
7.1.3. Guardar Archivo	11
7.1.4. Cambiar Usuario y/o Contraseña	11
7.1.5. Salir.....	12
7.2. Menú Herramientas.....	12
7.2.1. Conexión con el Controlador.....	12
7.2.2. Modificar Año de la Tabla de Tiempo.....	13
7.2.3. Actualizar Fecha y Hora del Controlador.....	13
7.2.4. Descargar Datos al Controlador	13
7.2.5. Guardar Archivo y Descargar al Controlador	14
7.2.6. Obtener Datos Desde el Controlador	14
7.2.7. Descargar la Contraseña para el Controlador.....	15
7.3. Menú Ayuda.....	16
7.3.1. Ayuda	16
7.3.2. Acerca de	16
7.4. Pestaña Seleccionar Días Feriados.....	16
7.5. Pestaña Configuración del Tipo de Día.....	17
7.6. Pestaña Personalizar Día.	18
8. ASISTENCIA TÉCNICA:	20

PROTOTIPO DEL SISTEMA DE ALERTA PARA EL CIERRE DE PUERTAS DE LAS UNIDADES DEL SISTEMA TROLEBÚS

PRESENTACIÓN:

El siguiente manual contiene la información necesaria para la manipulación del SISTEMA DE ALERTA PARA EL CIERRE DE PUERTAS DE LAS UNIDADES DEL SISTEMA TROLEBÚS.

IMPORTANTE:

Lea detenidamente este manual antes de operar el sistema. Conserve este manual para futuras referencias.

ADVERTENCIA:

Siempre respete las normas de seguridad cuando use este sistema. No intente modificar el hardware y/o el software del sistema sin antes recibir el adecuado asesoramiento técnico.

ALIMENTACIÓN DEL SISTEMA:

- Voltaje de alimentación: 100 - 120VAC / 50 - 60Hz
- Consumo de corriente: 3.5 - 4.5A

CARACTERÍSTICAS:

El Sistema de Alerta para el Cierre de Puertas de la Unidades del Sistema Trolebús presenta las siguientes características:

- Indicador visual de dos dígitos, visible a los usuarios del Sistema Trolebús, que permite visualización del tiempo que las puertas de acceso al trolebús permanecerán abiertas.
- Señal de alerta audible para los usuarios dentro del andén, para indicar tanto la apertura, como el cierre (manual, por parte del conductor) de las puertas del trolebús.
- Señal de alerta visual para el conductor del trolebús, para mostrar que ha transcurrido su tiempo de estadía en la parada y que puede cerrar las puertas y continuar su recorrido.
- Tabla de Tiempos programable, para la operación autónoma del sistema durante todo un año, la cual almacena la temporización para la apertura y el cierre de las puertas del trolebús de cada hora del día, además del tipo de día (ordinario, fin de semana, feriado o personalizado).

- Manejo de un sistema de archivos que permite respaldar la información correspondiente a la Tabla de Tiempo anual de la estación del Sistema Trolebús.
- Conexión entre el controlador principal y el computador para operaciones de programación y respaldo mediante el estándar RS-232.
- Software administrativo del sistema desarrollado bajo Microsoft Visual Basic, el cual permite tanto el almacenamiento, como el respaldo de datos del controlador principal.
- Modificación de la operación del sistema en tiempo real, para permitir la reprogramación del sistema, dependiendo de las condiciones actuales de la estación y su demanda de usuarios.
- Sistema robusto, provisto de inmunidad frente a las interferencias propias del Sistema Trolebús y capaz de trabajar las 24 horas del día.

MANIPULACIÓN DEL SOFTWARE DEL CONTROLADOR PRINCIPAL:

Como se puede apreciar en la Figura A1.1. el controlador principal posee integrado en su hardware un LCD, siete luces indicadoras y un teclado de siete teclas, las cuales permiten la manipulación del software del mismo.



Figura A1.1. Controlador Principal del Sistema.

La Tabla A1.1. muestra una lista con la descripción de las luces indicadoras del controlador principal. Adicionalmente, la Tabla A1.2. muestra una lista con las funciones asignadas a cada tecla

Tabla A1.1. Descripción de las Luces Indicadoras del Controlador Principal

No.	COLOR	DESCRIPCIÓN (Luz Encendida)
1	Rojo	Trolebús presente en la Estación
2	Verde	Emisión del mensaje de apertura de puertas (pulso)
3	Verde	Emisión del mensaje de cierre de puertas (pulso)
4	Verde	No asignado
5	Verde	No asignado
6	Amarillo	Modo de reproducción de mensajes.
7	Amarillo	Modo de grabación de mensajes.

Tabla A1.2. Función de las teclas del Controlador Principal

TECLA	FUNCIÓN
	Escape. Permite salir del menú en el que se encuentre. En caso de estar modificando la temporización regresa al menú anterior, caso contrario, regresa al menú principal.
	Incremento uno a uno. Permite desplazarse hacia el menú o submenú superior o incrementar los valores numéricos de uno en uno.
	Decremento uno a uno. Permite desplazarse hacia el menú o submenú inferior o decrementar los valores numéricos de uno en uno.
	No tiene función asignada.
	Decremento múltiple. Permite desplazarse en el menú o submenú 4 opciones hacia abajo o permite decrementar los valores numéricos de 10 en 10.
	Incremento múltiple. Permite desplazarse en el menú o submenú 4 opciones hacia arriba o permite incrementar los valores numéricos de 10 en 10.
	Aceptar. Permite ingresar al menú o submenú seleccionado, o acepta el valor asignado en un submenú destinado a este tipo de función.

Una vez comprendida las funciones del hardware, se detalla a continuación la operación del software desarrollado.

El sistema presenta las siguientes pantallas y menús:

6.1. *Pantalla Inicial*

Al encender el sistema, el controlador principal presenta una pantalla inicial, la cual indica la hora y fecha del sistema, cuando se presiona cualquier tecla en el módulo esta pantalla es reemplazada por la de Contraseña de Ingreso. La pantalla inicial vuelve a ser presentada 10 segundos después de no haberse presionado ninguna tecla del controlador principal.

6.2. *Contraseña de Ingreso*

Antes de poder ingresar a los menús de configuración del sistema, el controlador principal requiere que el usuario ingrese una combinación predefinida de 4 teclas. Si la combinación de teclas es inválida o si han pasado 5 segundos desde la última tecla presionada, el controlador vuelve a la Pantalla Inicial. Cuando se ha ingresado al software del controlador principal y no se realiza ninguna acción durante un tiempo determinado, se vuelve a presentar la Pantalla Inicial y se necesita ingresar nuevamente la contraseña para poder realizar modificaciones de operación.

La única manera de **cambiar** la Contraseña de Ingreso es mediante el Software Administrativo.

6.3. Menú Principal

6.3.1. Modificar Horario

Esta opción permite modificar la temporización a ser mostrada en los módulos de displays, para el cierre de puertas de las unidades del Sistema Trolebús. Dichas temporizaciones corresponden a:

- a. **Día Ordinario.** Se selecciona la hora de los días ordinarios que se desea modificar, posteriormente se modifica la temporización correspondiente a dicha hora.
- b. **Día Feriado.** Se selecciona la hora de los días feriados que se desea modificar, posteriormente se modifica la temporización correspondiente a dicha hora.
- c. **Fin de Semana.** Se selecciona la hora de los fines de semana que se desea modificar, posteriormente se modifica la temporización correspondiente a dicha hora.
- d. **Personalizado.** En este caso, se presenta un submenú para poder modificar:
 - **Día Actual.** Se selecciona la hora que se quiere modificar del día actual y a continuación, la temporización correspondiente a dicha hora.
 - **Seleccionar Día.** Se selecciona el día del año al que se quiere modificar la temporización. Primeramente se selecciona el mes, luego el día y luego la hora que se quiere modificar del día en cuestión; finalmente, se modifica la temporización correspondiente a dicha hora.

6.3.2. Modificar Hora Actual.

Esta opción permite modificar la temporización de la hora que esté cursando en el instante en que se hace el cambio. Es decir, permite personalizar la hora presente del día en curso.

6.3.3. Luz de Fondo.

Esta opción permite personalizar el encendido/apagado de la luz de fondo (backlight). Presenta tres opciones de operación:

- a. **Apagar.** Mantiene siempre apagada la luz de fondo.
- b. **Encender.** Mantiene siempre encendida la luz de fondo.
- c. **Automático.** Se enciende y se apaga la luz de fondo en forma automática, al presionar o dejar sin presionar el teclado un tiempo determinado.

6.3.4. Mensajes de Voz.

Esta opción permite personalizar las opciones correspondientes a los mensajes de voz, tanto el de apertura, como el de cierre. Permite modificar, para cada mensaje, las siguientes opciones:

- a. **Grabar Mensaje.** Permite grabar el mensaje seleccionado. En caso de existir un mensaje ya grabado, será reemplazado por el nuevo. Debe mantenerse presionada la tecla Aceptar (↵) mientras se realiza la grabación del mensaje. Al terminar de grabar el mensaje será reproducido automáticamente para confirmar el correcto estado del mismo.
- b. **Reproducir Mensaje.** Reproduce el mensaje seleccionado.
- c. **Habilitar Mensaje.** Habilita la reproducción del mensaje seleccionado.
- d. **Deshabilitar Mensaje.** Deshabilita la reproducción del mensaje seleccionado.
- e. **Tiempo de Activación.** Personaliza el tiempo de activación de los mensajes. Para el mensaje de **apertura** se selecciona el tiempo que debe esperar el sistema antes de reproducir el mensaje de apertura, luego de haber detectado la presencia del trolebús. Para el mensaje de **cierre** se selecciona el tiempo en que debe reproducirse el mensaje, antes que finalice la temporización para el cierre de las puertas del trolebús.

6.3.5. Actualizar Hora.

Esta opción permite actualizar el Reloj de Tiempo Real, el cual constituye la base de la operación del sistema.

6.3.6. Acerca de...

Despliega información acerca de los desarrolladores del proyecto, para posible soporte técnico.

La Figura A1.2. presenta un diagrama de flujo, el cual permite aclarar y resumir la operación del software del Controlador Principal.

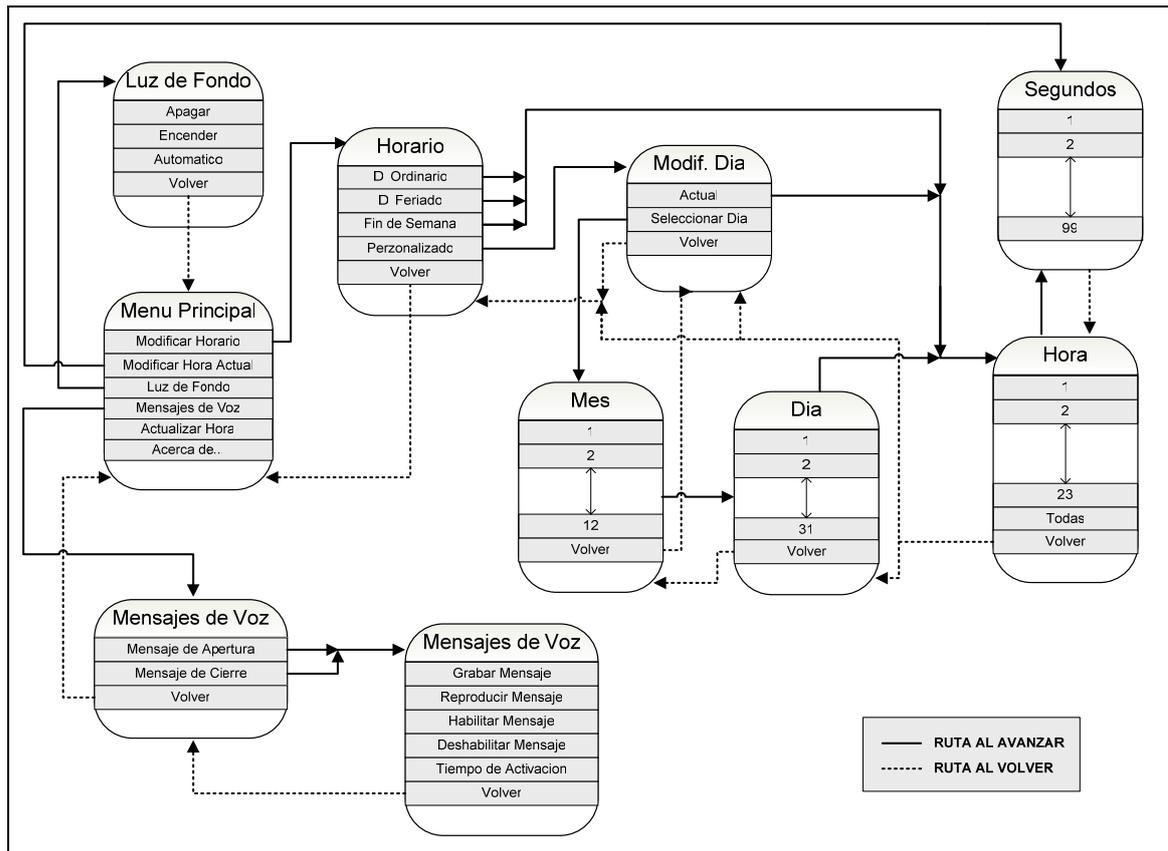


Figura A1.2. Diagrama de Flujo de la Operación del Controlador Principal

MANIPULACIÓN DEL SOFTWARE ADMINISTRATIVO:

El Software Administrativo constituye la aplicación desarrollada para la administración del Sistema de Alerta mediante un computador personal. El proceso de instalación de este software se detalla en el documento titulado "Memoria Técnica" (Sección 13.1).

A continuación se detalla la manipulación del Software Administrativo:

Al iniciar la aplicación se presenta la Pantalla Inicial de la aplicación, la cual se puede apreciar en la Figura A1.3.

La Pantalla Inicial es presentada durante tres segundos y a continuación se presenta la pantalla de "Inicio de Sesión", Figura A1.4., en la cual debe ingresarse el nombre del usuario y la contraseña.



Figura A1.3. Pantalla Inicial del Software Administrativo

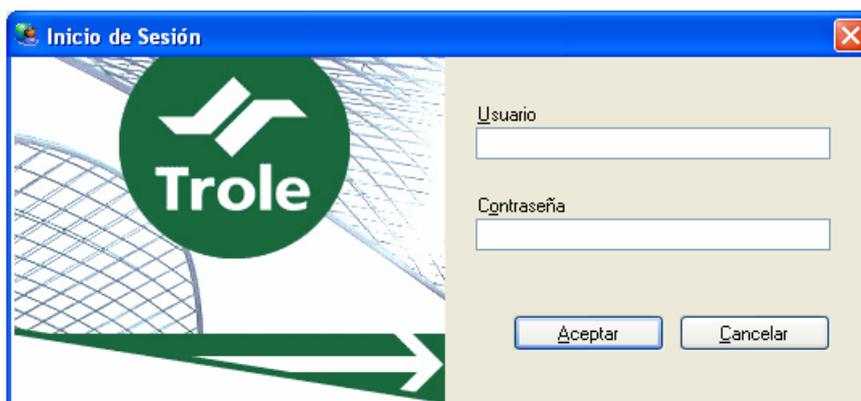


Figura A1.4. Pantalla Inicio de Sesión

El software tiene por defecto los siguientes datos de inicio de sesión:

Usuario: usuario
Contraseña: trolebus

Se recomienda cambiar estos datos, con el fin de proteger la información desarrollada mediante este software. Para ello se debe seguir la ruta: <Archivo \ Cambiar Usuario y/o Contraseña>, mayor detalle de esta operación en la Sección 7.1.4.

7.1. Menú Archivo

El menú Archivo presenta 5 opciones, las cuales pueden apreciarse en la Figura A1.5. y se detallan a continuación:



Figura A1.5. Menú Archivo

7.1.1. Abrir Archivo Existente. Esta opción permite cargar en información anteriormente guardada.

Al seleccionar esta opción, se pregunta al usuario si desea guardar los cambios realizados en el archivo actual, en caso de querer guardar los cambios, la operación es igual a “Guardar Archivo” (Sección 7.1.3.). En caso de no querer guardar los cambios, se abre el cuadro de diálogos “Abrir” mostrado en la Figura A1.6.

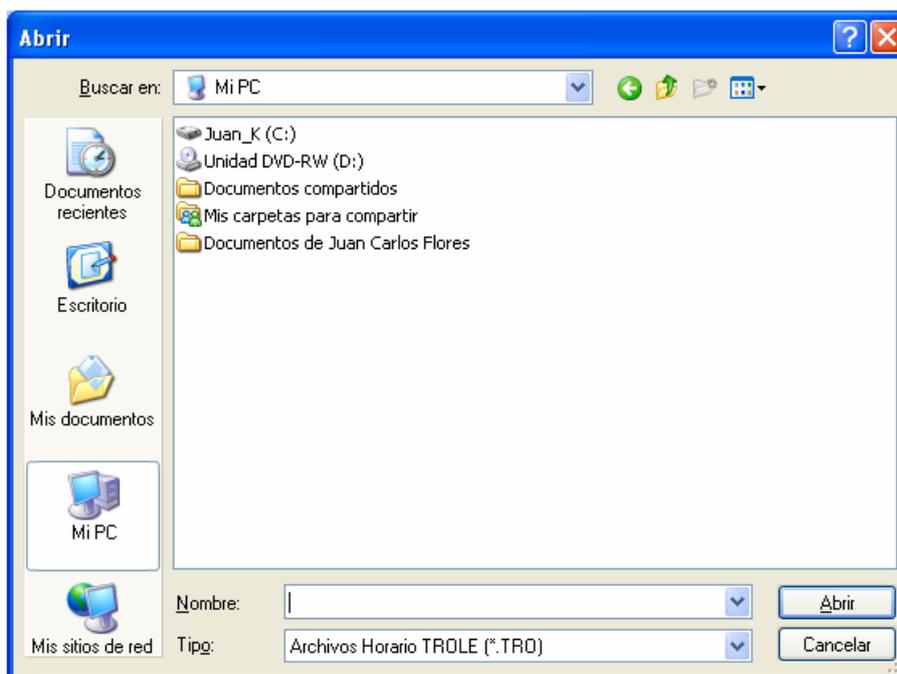


Figura A1.6. Abrir Archivo

Este cuadro de diálogos permite ubicar la ruta del archivo guardado y que se quiere abrir. El tipo de archivo que maneja el software es *.TRO.

Al seleccionar un archivo válido y hacer click en “Abrir”, se carga la información contenida en éste.

7.1.2. Nuevo Archivo. Esta opción permite crear un nuevo archivo, partiendo de los datos por defecto que se cargan en la aplicación.

Al igual que en el caso anterior, se pregunta al usuario si desea guardar los cambios realizados en el archivo actual, en caso de querer guardar los cambios, la operación es igual a “Guardar Archivo” (Sección 7.1.3.). En

caso de no querer guardar los cambios, se cargan los datos por defecto en la aplicación.

7.1.3. Guardar Archivo. Esta opción permite grabar la información que se encuentra en ese instante cargada en la aplicación.

Al aceptar esta opción, se abre el cuadro de diálogos “Guardar como”, mostrado en la Figura A1.7. Aquí se elige la ubicación en la que se desea almacenar el archivo, se le da un nombre válido (con la extensión *.TRO) y se da click en “Guardar”.

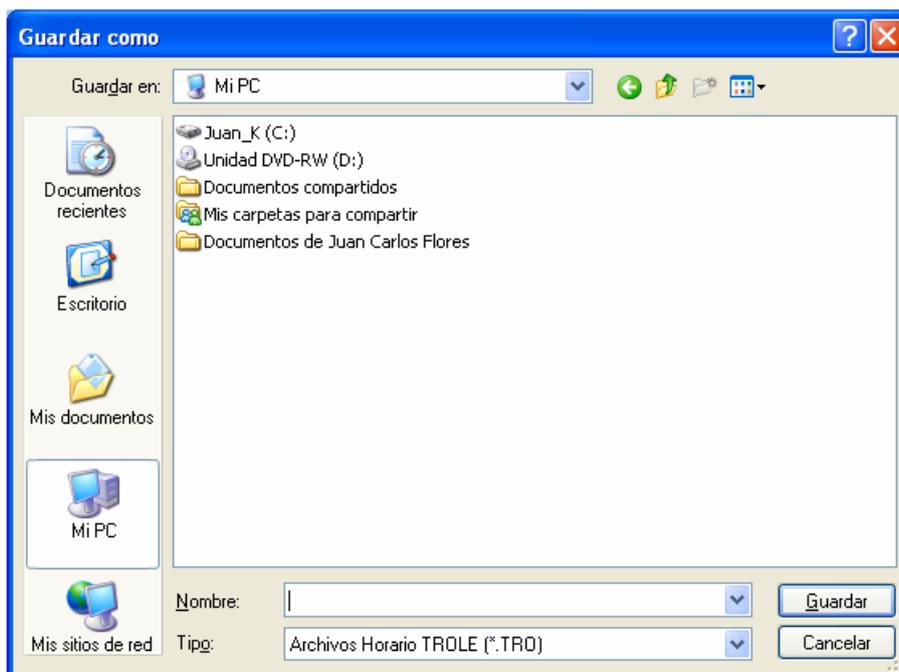


Figura A1.7. Pantalla Guardar como

7.1.4. Cambiar Usuario y/o Contraseña. Esta opción permite al usuario cambiar su nombre y/o su contraseña, según crea conveniente.

Al aceptar esta opción, se muestra la pantalla “Cambiar Usuario y/o Contraseña”, en la cual deben ser ingresados los datos mostrados en la Figura A1.8.

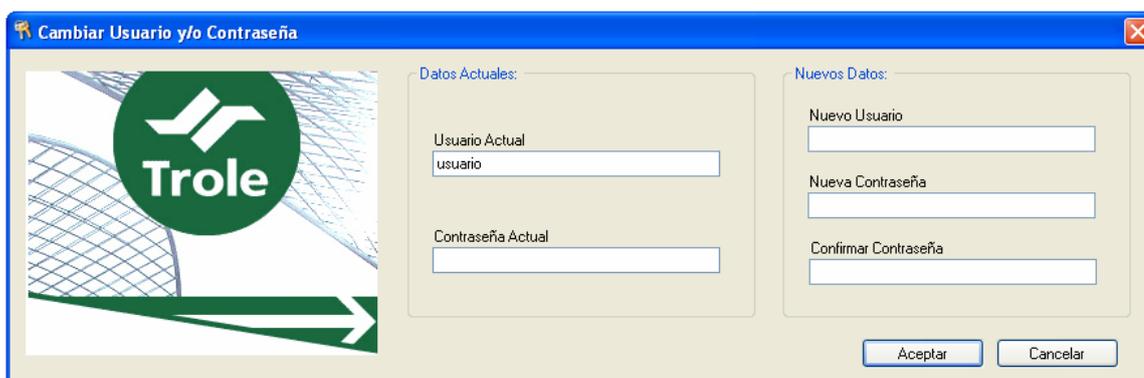


Figura A1.8. Pantalla Cambiar Usuario y/o Contraseña

Para poder cambiar el nombre del usuario y/o la contraseña, deben primero llenarse los “Datos Actuales”. Este es un requisito para poder ingresar los “Nuevos Datos”. Los nuevos datos pueden ser cambiados según la conveniencia del usuario del software, la única **restricción** es que la contraseña debe ser de al menos 5 caracteres.

Adicionalmente, se presentan mensajes de información que indican si fueron cambiados los datos de forma correcta, o en el caso contrario, se indica el tipo de error que se cometió al ingresar los datos.

7.1.5. **Salir.** Esta opción permite cerrar la aplicación.

Al aceptar esta opción, se pregunta al usuario si desea guardar los cambios realizados en el archivo actual, en caso de querer guardar los cambios, la operación es igual a “Guardar Archivo” (Sección 7.1.3.) y en el caso contrario se cierra la aplicación.

7.2. **Menú Herramientas.**

El menú Herramientas presenta 7 opciones, las cuales pueden apreciarse en la Figura A1.9. y se detallan a continuación:

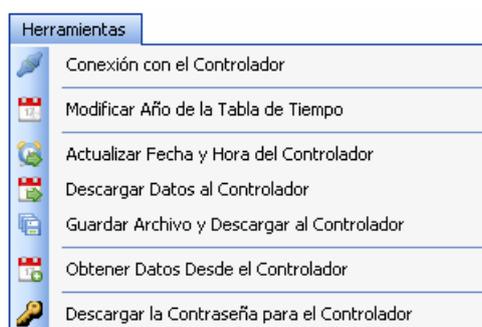


Figura A1. 9. Menú Herramientas

7.2.1. **Conexión con el Controlador.** Esta opción permite establecer la conexión del computador con el controlador principal.

Al aceptar esta opción se abre el cuadro de diálogos “Conexión”, mostrado en la Figura A1.10.

Este cuadro únicamente permite modificar el número del puerto de comunicación serial (COM) para establecer la comunicación. Adicionalmente, se proporciona detalles sobre velocidad, número de bits de datos y la paridad con la que trabaja la aplicación, para evitar incoherencias en la configuración del puerto serial del computador.

Una vez que se ha establecido satisfactoriamente la conexión, se activan las herramientas de la aplicación que necesitan una comunicación activa para su desempeño.

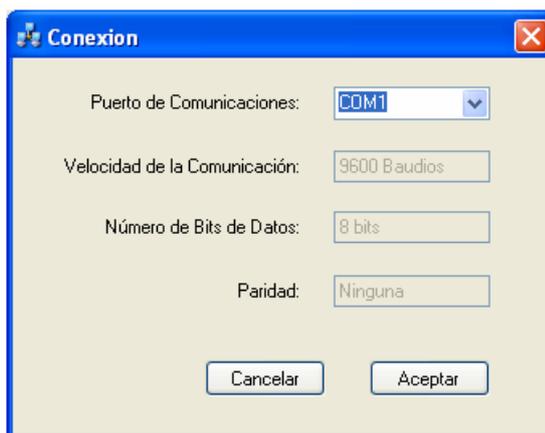


Figura A1.10. Pantalla Conexión

7.2.2. Modificar Año de la Tabla de Tiempo. Esta opción permite cambiar el año de la Tabla de Tiempo Anual que maneja el Sistema de Alerta. Esto resulta útil en el caso de querer realizar con anticipación la Tabla de Tiempo de un año diferente al que esté cursando.

Al aceptar esta opción se abre el cuadro de diálogos "Modificar Año", mostrado en la Figura A1.11., el cual simplemente permite modificar el año.



Figura A1.11. Pantalla Modificar Año

7.2.3. Actualizar Fecha y Hora del Controlador. Esta opción actualiza el Reloj de Tiempo Real (RTC) que maneja el controlador principal, el cual constituye la base de la operación del Sistema de Alerta.

Al aceptar esta opción, automáticamente se actualiza el RTC y se presenta un mensaje de confirmación de la acción realizada.

7.2.4. Descargar Datos al Controlador. Esta opción descarga al controlador la información que se encuentre en ese momento en la aplicación, esté o no guardada en un archivo.

Al aceptar esta opción, se desactivan los controles que pueden afectar a la descarga de datos. Además, en la sección inferior derecha puede ser apreciada una barra de estado del proceso de descarga, además de una breve descripción de los datos que están siendo descargados, lo cual

puede ser apreciado en la Figura A1.12. Todos los datos de fechas y temporizaciones se descargan automáticamente para el correcto desempeño del Sistema de Alerta.

Al finalizar la descarga se presenta un mensaje de confirmación de la acción realizada y se vuelven a activar los controles de la aplicación.

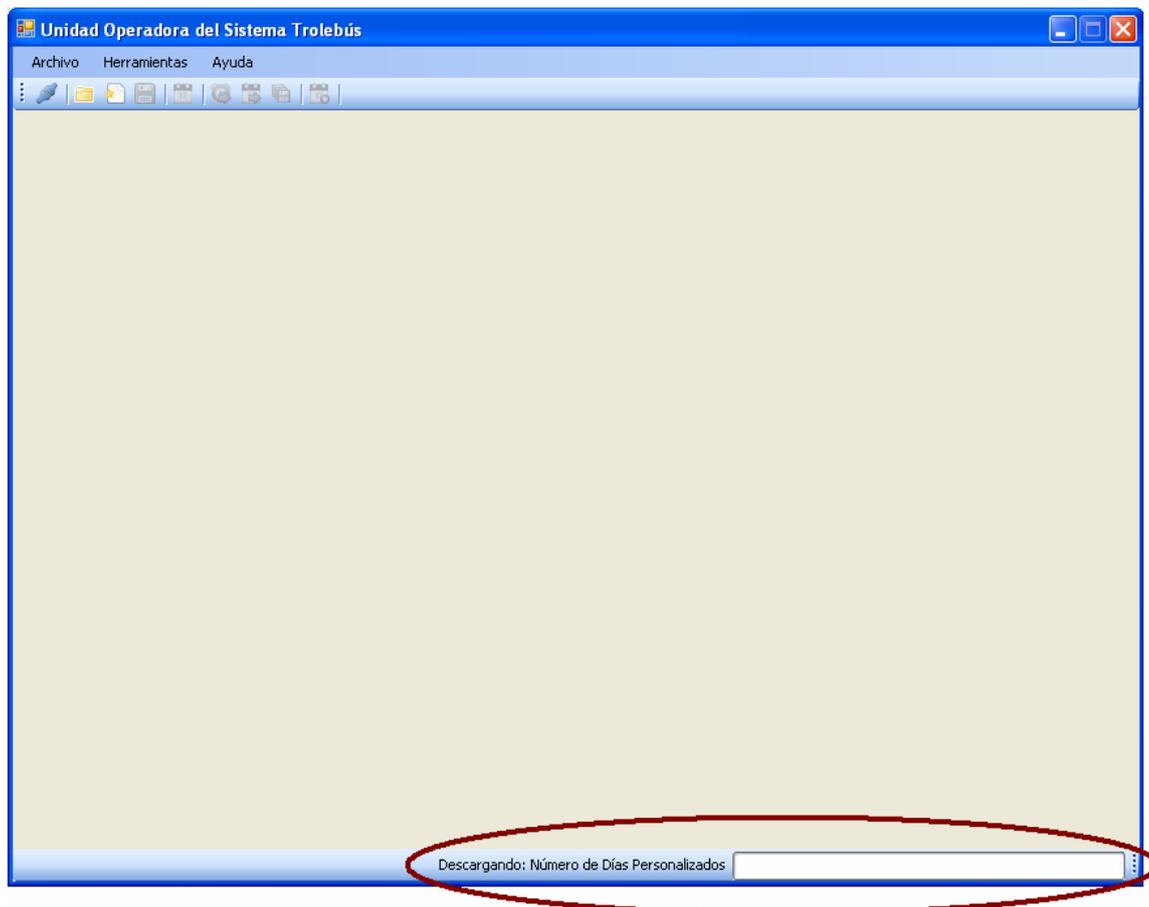


Figura A1.12. Estado de la Pantalla al Descargar Datos al Controlador

7.2.5. Guardar Archivo y Descargar al Controlador. Esta opción guarda la información presente en la aplicación y a continuación la descarga al controlador.

Esta opción es una combinación de las opciones “Guardar Archivo” (Sección 7.1.3.) y “Descargar Datos al Controlador” (Sección 7.2.4.).

7.2.6. Obtener Datos Desde el Controlador. Esta opción obtiene la información que se encuentre almacenada en el controlador principal.

Al aceptar esta opción, se desactivan los controles que pueden afectar a la transmisión de datos. Además, en la sección inferior derecha se puede apreciar una barra de estado del proceso, además de una breve descripción de los datos que están siendo subidos al computador; lo cual puede ser apreciado en la Figura A1.13. Al finalizar el proceso se presenta un mensaje de confirmación de la acción realizada y se vuelven a activar los controles de la aplicación.

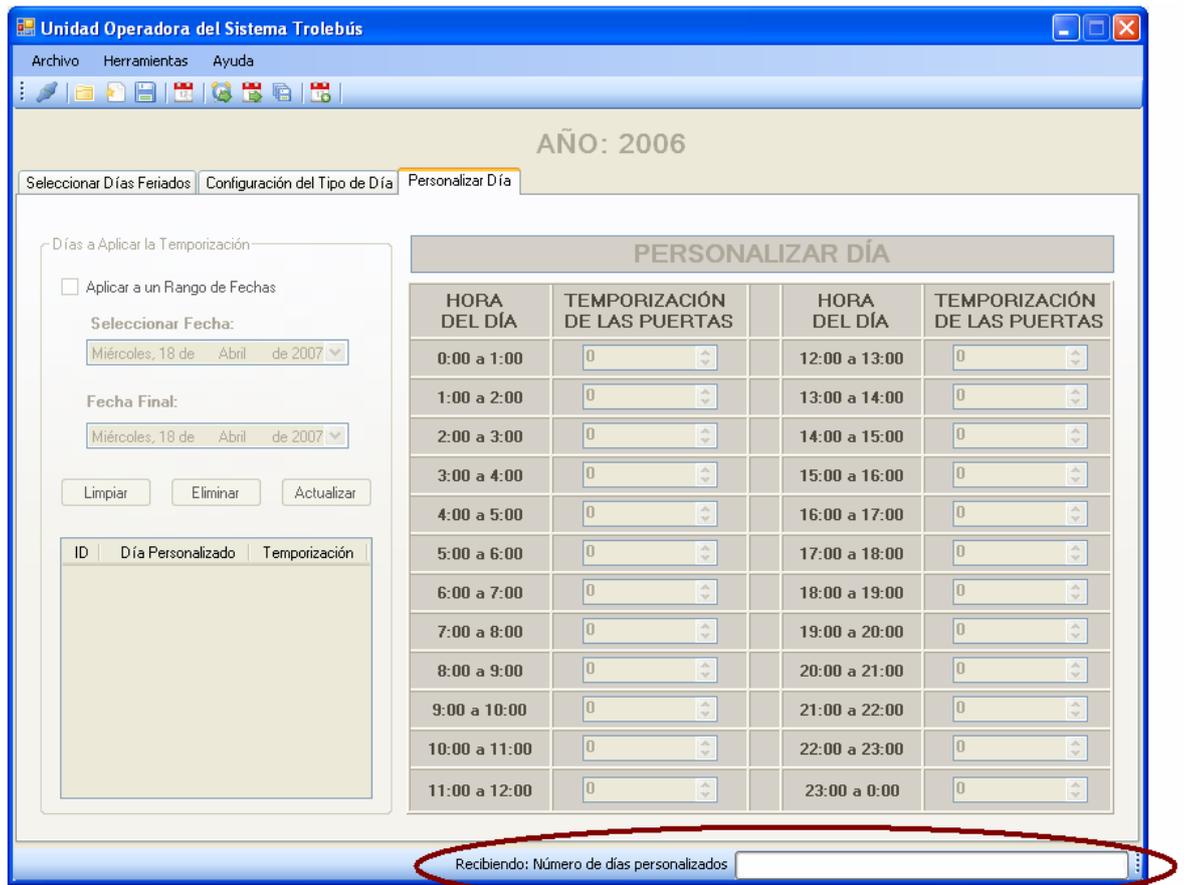


Figura A1.13. Estado de la Pantalla al Obtener Datos Desde el Controlador

- 7.2.7. Descargar la Contraseña para el Controlador.** Esta opción permite descargar la contraseña para la manipulación del controlador principal. Al aceptar esta opción, se abre el cuadro de diálogos mostrado en la Figura A1.14.



Figura A1.14. Pantalla Descargar la Contraseña para el Controlador

La contraseña a ingresar debe formar una combinación de cuatro teclas, existen mensajes de confirmación para verificar que la contraseña se

ingrese adecuadamente y para indicar que la contraseña ha sido descargada satisfactoriamente.

Para el ingreso de la contraseña se hace click en cada botón de la sección “Teclado del Controlador” (Figura A1.14.), hasta formar dicha combinación. Cabe aclarar que esta es la única manera de cambiar la contraseña del Controlador Principal.

7.3. Menú Ayuda.

El menú Ayuda presenta 2 opciones, las cuales pueden apreciarse en la Figura A1.15. y se detallan a continuación:



Figura A1.15. Menú Ayuda

7.3.1. Ayuda. Esta opción abre un cuadro de diálogos que informa como puede acceder a la información mostrada en este manual.

7.3.2. Acerca de... Esta opción proporciona información concerniente al desarrollo de la aplicación.

La Figura A1.16. permite apreciar el cuadro de diálogos que muestra dicha información.

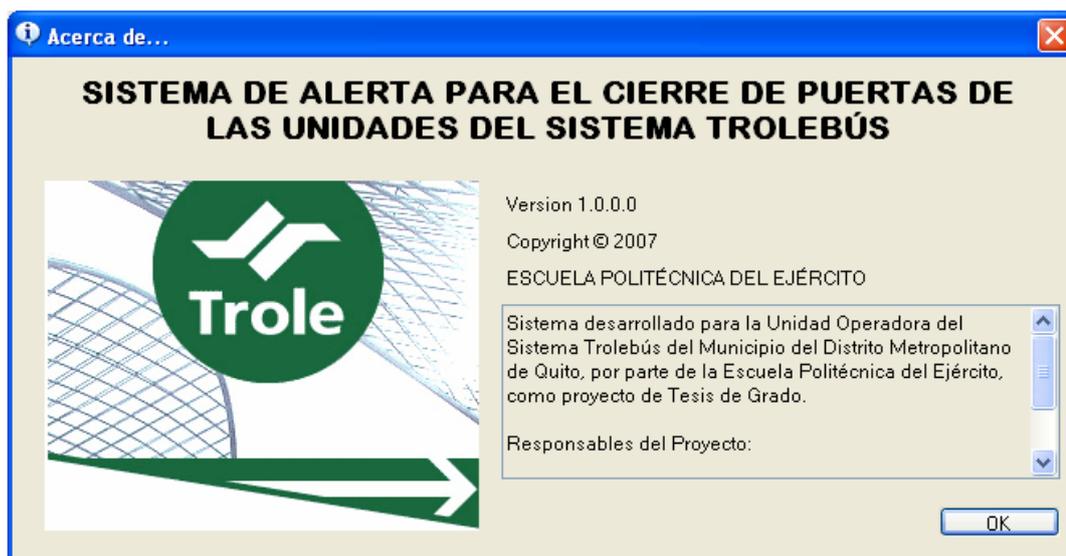


Figura A1.16. Pantalla Acerca de...

7.4. Pestaña Seleccionar Días Feriados.

Esta pestaña permite administrar las fechas de los días feriados, la Figura A1.17. muestra las opciones que se pueden manipular mediante esta pestaña.

Seleccionar Días Feriados | Configuración del Tipo de Día | Personalizar Día

Tipo de Día

Día Ordinario

Fin de Semana

Día Feriado

DÍA ORDINARIO			
HORA DEL DÍA	TEMPORIZACIÓN DE LAS PUERTAS	HORA DEL DÍA	TEMPORIZACIÓN DE LAS PUERTAS
0:00 a 1:00	10	12:00 a 13:00	10
1:00 a 2:00	10	13:00 a 14:00	10
2:00 a 3:00	10	14:00 a 15:00	10
3:00 a 4:00	10	15:00 a 16:00	10
4:00 a 5:00	10	16:00 a 17:00	10
5:00 a 6:00	10	17:00 a 18:00	10
6:00 a 7:00	10	18:00 a 19:00	10
7:00 a 8:00	10	19:00 a 20:00	10
8:00 a 9:00	10	20:00 a 21:00	10
9:00 a 10:00	10	21:00 a 22:00	10
10:00 a 11:00	10	22:00 a 23:00	10
11:00 a 12:00	10	23:00 a 0:00	10

Figura A1.18. Pestaña Configuración del Tipo de Día

En la sección izquierda de la pantalla activa la casilla de verificación respectiva al tipo de día al que se quiere asignar las temporizaciones. Esta pestaña administra los tipos de día:

1. **Día Ordinario.**
2. **Fin de Semana.**
3. **Día Feriado.**

En la tabla que ocupa la mayor parte de la pantalla debe ingresarse la temporización respectiva (en segundos) de cada hora del día, correspondiente al tipo de día seleccionado.

7.6. Pestaña Personalizar Día.

Esta pestaña permite personalizar, de forma individual, un día cualquiera del año, pudiendo personalizar los 365 días del año. La Figura A1.19. muestra las opciones que se pueden manipular mediante esta pestaña.

ASISTENCIA TÉCNICA:

Para consultas y soporte técnico puede comunicarse con:

- Juan Carlos Flores.
Telf: 09-7199-594 e-mail: jcacmil@gmail.com

ANEXO 2

MEMORIA TÉCNICA

**PROTOTIPO DEL SISTEMA DE ALERTA
PARA EL CIERRE DE PUERTAS DE LAS
UNIDADES DEL SISTEMA TROLEBÚS**

MEMORIA TÉCNICA

**ESCUELA POLITÉCNICA DEL EJÉRCITO – 2007.
PROYECTO DE GRADO.**

TABLA DE CONTENIDOS:

1. PRESENTACIÓN:	3
2. IMPORTANTE:	3
3. ADVERTENCIA:	3
4. SEGURIDAD:	3
5. ALIMENTACIÓN DEL SISTEMA:	3
6. REQUERIMIENTOS DEL SISTEMA:	3
6.1. Requerimientos de Hardware:	3
6.2. Requerimientos de Software:.....	4
7. CARACTERÍSTICAS:	4
8. CONTROLADOR PRINCIPAL	5
8.1. Luces Indicadoras del Controlador Principal:.....	5
8.2. Teclado del Controlador Principal:	5
9. CONDICIONES INICIALES:	6
10. DESCRIPCIÓN DE LA OPERACIÓN:	6
11. PUESTA EN MARCHA:	6
12. DIAGRAMA DE CONEXIONES:	7
12.1. Interfaz del Controlador Principal:	8
12.2. Sensor Infrarrojo:.....	9
12.3. Módulos de Displays:	10
12.4. Semáforo:.....	11
12.5. Módulo de Voz:.....	12
12.6. Amplificador de Audio:.....	13
13. INSTALACIÓN DEL SOFTWARE:	14
13.1. Instalación del Software Administrativo.	14
13.2. Descarga del Software del Microcontrolador PIC16F84A.....	14
13.3. Descarga del Software del Controlador Principal.	14
14. GUÍA DE SOLUCIÓN DE PROBLEMAS:	15
15. ASISTENCIA TÉCNICA:	17

PROTOTIPO DEL SISTEMA DE ALERTA PARA EL CIERRE DE PUERTAS DE LAS UNIDADES DEL SISTEMA TROLEBÚS

PRESENTACIÓN:

El siguiente manual contiene importante información del SISTEMA DE ALERTA PARA EL CIERRE DE PUERTAS DE LAS UNIDADES DEL SISTEMA TROLEBÚS, en lo referente a seguridad, características, funcionamiento, guía de solución de problemas y asistencia técnica.

IMPORTANTE:

Lea detenidamente este manual antes de operar el sistema. Conserve este manual para futuras referencias.

ADVERTENCIA:

Siempre respete las normas de seguridad cuando use este sistema. No intente modificar el hardware y/o el software del sistema sin antes recibir el adecuado asesoramiento técnico.

4. SEGURIDAD:

- Refiérase al Diagrama de Conexiones (Sección 12) antes de realizar cualquier prueba con sistema.
- Asegúrese que la alimentación del sistema sea la indicada en este manual (Sección 5).
- Evite la humedad excesiva en los diferentes componentes del sistema.

ALIMENTACIÓN DEL SISTEMA:

- Voltaje de alimentación: 100 - 120VAC / 50 - 60Hz
- Consumo de corriente: 3.5 - 4.5A

REQUERIMIENTOS DEL SISTEMA:

6.1. *Requerimientos de Hardware:*

- Procesador INTEL Pentium III, AMD Athlon XP o superior.
- 256 MB de memoria RAM.
- Al menos 100 MB de espacio libre en el disco duro.
- Puerto Serial con el estándar RS-232 y con conector DB-9 tipo macho.
- Cable de datos serial (recto), con conectores DB-9 tipo hembra en ambos extremos.

- Cable de programación para el Controlador Principal.
- Programador de microcontroladores PIC.

6.2. Requerimientos de Software:

- Sistema Operativo Windows 98 SE o superior.
- Microsoft .NET Framework 2.0
- Windows Installer 3.1
- DC 9.25 RFU (Para programar el Controlador Principal).
- WINPIC800, IC-PROG o cualquier software desarrollado para grabar microcontroladores PIC.

CARACTERÍSTICAS:

El Sistema de Alerta presenta las siguientes características:

- Indicador visual de dos dígitos, visible a los usuarios del Sistema Trolebús, que permite visualización del tiempo que las puertas de acceso al trolebús permanecerán abiertas.
- Señal de alerta audible para los usuarios dentro del andén, para indicar tanto la apertura, como el cierre (manual, por parte del conductor) de las puertas del trolebús.
- Señal de alerta visual para el conductor del trolebús (semáforo), para mostrar que ha transcurrido su tiempo de estadía en la parada y que puede cerrar las puertas y continuar su recorrido.
- Tabla de Tiempos programable, para la operación autónoma del sistema durante todo un año, la cual almacena la temporización para la apertura y el cierre de las puertas del trolebús de cada hora del día, además del tipo de día (ordinario, fin de semana, feriado o personalizado).
- Manejo de un sistema de archivos que permite respaldar la información correspondiente a la Tabla de Tiempo anual de la estación del Sistema Trolebús.
- Conexión entre el controlador principal y el computador para operaciones de programación y respaldo mediante el estándar RS-232.
- Software administrativo del sistema desarrollado bajo Microsoft Visual Basic, el cual permite tanto el almacenamiento, como el respaldo de datos del controlador principal.
- Modificación de la operación del sistema en tiempo real, para permitir la reprogramación del sistema, dependiendo de las condiciones actuales de la estación y su demanda de usuarios.
- Sistema robusto, provisto de inmunidad frente a las interferencias propias del Sistema Trolebús y capaz de trabajar las 24 horas del día.

CONTROLADOR PRINCIPAL

La Figura A2.1. muestra una imagen del controlador principal. Como se puede apreciar, éste posee integrado un LCD, siete luces indicadoras y un teclado de siete teclas.



Figura A2.1. Controlador Principal del Sistema.

8.1. Luces Indicadoras del Controlador Principal:

El controlador principal posee 7 luces indicadoras, cuya función se encuentra descrita en la Tabla A2.1.

Tabla A2.1. Descripción de las Luces Indicadoras del Controlador Principal

No.	COLOR	DESCRIPCIÓN (Luz Encendida)
1	Rojo	Trolebús presente en la Estación
2	Verde	Emisión del mensaje de apertura de puertas (pulso)
3	Verde	Emisión del mensaje de cierre de puertas (pulso)
4	Verde	No asignado
5	Verde	No asignado
6	Amarillo	Modo de reproducción de mensajes.
7	Amarillo	Modo de grabación de mensajes.

8.2. Teclado del Controlador Principal:

El controlador principal está provisto de 7 teclas, cuya función se encuentra descrita en la Tabla A2.2.

Tabla A2.2. Función de las Teclas del Controlador Principal

TECLA	FUNCIÓN
	Escape. Permite salir del menú en el que se encuentre. En caso de estar modificando la temporización regresa al menú anterior, caso contrario, regresa al menú principal.
	Incremento uno a uno. Permite desplazarse hacia el menú o submenú superior o incrementar los valores numéricos de uno en uno.
	Decremento uno a uno. Permite desplazarse hacia el menú o submenú inferior o decrementar los valores numéricos de uno en uno.
	No tiene función asignada.

	Decremento múltiple. Permite desplazarse en el menú o submenú 4 opciones hacia abajo o permite decrementar los valores numéricos de 10 en 10.
	Incremento múltiple. Permite desplazarse en el menú o submenú 4 opciones hacia arriba o permite incrementar los valores numéricos de 10 en 10.
	Aceptar. Permite ingresar al menú o submenú seleccionado, o acepta el valor asignado en un submenú destinado a este tipo de función.

CONDICIONES INICIALES:

Al ser encendido el Sistema de Alerta (incluyendo el semáforo), éste presenta las siguientes condiciones iniciales:

- El controlador presenta su Pantalla Inicial.
- Está encendida la sexta luz indicadora del controlador principal (Modo de Reproducción activado).
- Todos los displays presentan el número cero.
- El semáforo tiene encendida la luz verde.

DESCRIPCIÓN DE LA OPERACIÓN:

Al ser detectado un trolebús en la estación, el Sistema de Alerta opera de la siguiente manera:

- Se enciende la primera luz indicadora del controlador principal (Trolebús presente en la estación, Tabla A2.1.) mientras esté presente un trolebús.
- El semáforo cambia y se enciende la luz roja.
- El sistema espera el tiempo que ha sido configurado para la emisión del mensaje de apertura.
- En los módulos de displays, oscila (2 veces) el valor de la temporización que permanecerán abiertas las puertas del trolebús.
- Da un pulso la luz indicadora del mensaje de alerta de apertura de puertas (Emisión del mensaje de apertura de puertas, Tabla A2.1.) y se emite dicho mensaje.
- Inicia la cuenta de la temporización definida para la hora actual.
- Al faltar un tiempo preestablecido en el controlador principal para que finalice la cuenta de la temporización, da un pulso la luz indicadora del mensaje de alerta del cierre de puertas (Emisión del mensaje de cierre de puertas, Tabla A2.1.) se emite dicho mensaje y oscila el valor mostrado en los módulos de displays.
- Finaliza la cuenta y se enciende la luz verde en el semáforo.

PUESTA EN MARCHA:

- Realice las conexiones del sistema **basándose estrictamente** en el Diagrama de Conexiones (Sección 12).
- Alimente el sistema de acuerdo a lo mencionado en este manual (Sección 5).

- Encienda el sistema con el interruptor presente en la parte frontal de la caja del controlador principal.
- Encienda el semáforo con el interruptor ubicado en la parte superior de la caja del controlador principal.
- Ajuste el volumen.
- Espere a que llegue un trolebús a la estación y verifique la operación del sistema (Sección 10).
- El sistema tiene valores de funcionamiento por defecto, pero éste debe ser programado adecuadamente para su correcto desempeño.

DIAGRAMA DE CONEXIONES:

La Figura A2.2. muestra el diagrama de conexiones correspondiente al Sistema de Alerta, el cual resume las conexiones a realizarse al momento de la implementación del sistema. A continuación se proporciona mayor detalle en lo concerniente a la instalación misma del sistema.

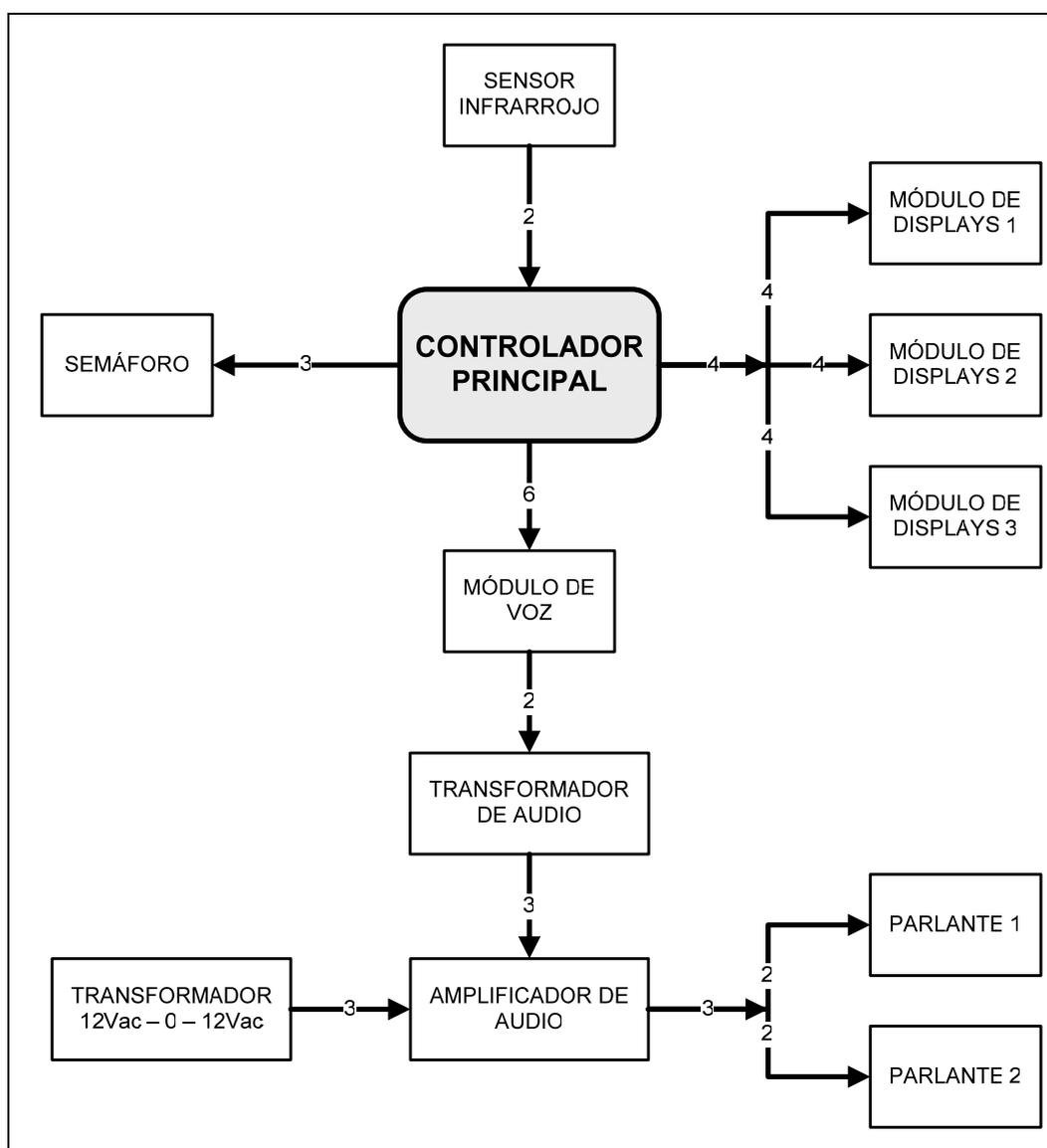


Figura A2.2. Diagrama de Conexiones del Sistema de Alerta

12.1. *Interfaz del Controlador Principal:*

El controlador principal, se encarga de enviar y recibir todas las señales de control necesarias para el correcto desempeño del sistema. La Figura A2.3. muestra la placa de interfaz del controlador principal, en dicha figura se encuentran etiquetadas las señales para la instalación del cableado, dichas señales son:

1. **CONECTOR DB-9 A PC.** Este es el conector para ubicar el cable serial (recto) para las operaciones de almacenamiento y respaldo de datos del controlador principal.
2. **RS-485.** Son las señales asociadas a la red RS-485 utilizada para la presentación de datos en los displays.
 - **485(-).** Señal diferencial negativa de la red.
 - **485(+).** Señal diferencial positiva de la red.
3. **ENTRADAS / SALIDAS.** Son las entradas y salidas necesarias para el control del sistema.
 - **MSG1.** Salida para la activación del mensaje de apertura de puertas.
 - **MSG2.** Salida para la activación del mensaje de cierre de puertas.
 - **PLAY.** Salida para la activación del modo de reproducción de mensajes de audio.
 - **REC.** Salida para la activación del modo de grabación de mensajes de audio.
 - **SENSOR.** Entrada de la señal de activación del sensor.
4. **BACK-UP.** Es la entrada de voltaje de la batería de respaldo o back-up.
 - **GND.** Tierra del sistema (negativo de la batería)
 - **VBAT.** Voltaje de la batería de respaldo (positivo de la batería).
5. **ALIMENTACIÓN.** Es la entrada de voltaje necesaria para la operación del controlador principal.
 - **24Vdc.** Entrada de 24 V_{DC}.
 - **GND.** Tierra del sistema.
6. **CONTROLADOR PRINCIPAL.** Es el conector designado a la entrada de los pines del controlador principal.
7. **SEMÁFORO.** Son las señales de salida para la conexión del semáforo.
 - **L (110Vac).** Es la fase de señal de 110Vac (El neutro llega hasta donde esté ubicado el semáforo).
 - **LUZ ROJA.** Señal de activación de la luz roja del semáforo.
 - **LUZ VERDE.** Señal de activación de la luz verde del semáforo.
8. **5Vdc.** Es una salida de voltaje que alimentará al módulo que se encarga de almacenar los mensajes de voz.

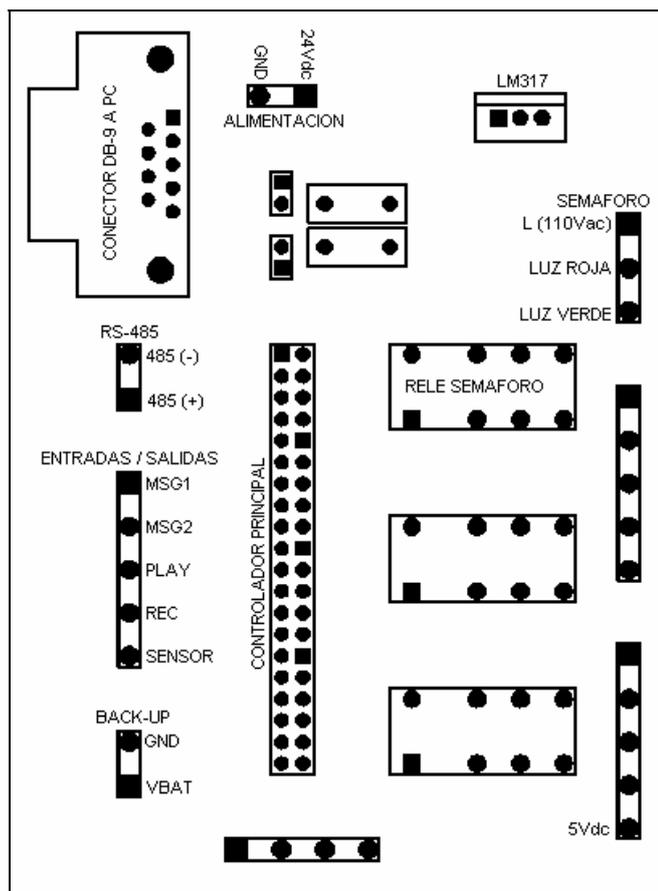


Figura A2.3. Placa de la Interfaz del Controlador Principal

12.2. Sensor Infrarrojo:

El sensor infrarrojo es el encargado de detectar la presencia del trolebús en la estación. Este sensor es el mismo utilizado para abrir las puertas del andén, por lo que ya se encuentra alimentado y simplemente se debe tomar su señal de activación y enviarla al controlador principal.

Este sensor se encuentra instalado sobre la segunda puerta del andén, pero su señal de activación se encuentra en los controladores de cada puerta del andén.

Como se puede apreciar en la Figura A2.2., se deben conectar simplemente 2 cables entre el sensor y el controlador principal:

1. Señal de activación del sensor (Borne No. 11 en la Figura A2.4.).
2. Señal de Tierra (Borne No. 9 en la Figura A2.4.).

La Figura A2.4. muestra la ubicación de estas señales en el controlador de una de las puertas del andén. Dichas señales pueden ser tomadas de cualquier puerta.



Figura A2.4. Señales del Sensor ubicadas en el Controlador de una puerta del andén

12.3. Módulos de Displays:

Los módulos de displays están destinados a indicar a los usuarios del Sistema Trolebús el tiempo que permanecerán abiertas las puertas del trolebús estacionado frente al andén.

Como se puede apreciar en la Figura A2.2., se necesitan 4 cables de conexión entre cada módulo de displays y el controlador principal, cabe recalcar que los 4 cables son los mismos para cada módulo y son los siguientes:

1. Voltaje de alimentación (24 Vdc).
2. Tierra del sistema (GND).
3. Señal diferencial positiva de la red RS-485 (485(+)).
4. Señal diferencial negativa de la red RS-485 (485(-)).

La Figura A2.5. muestra la ubicación de estas señales en el módulo de displays.

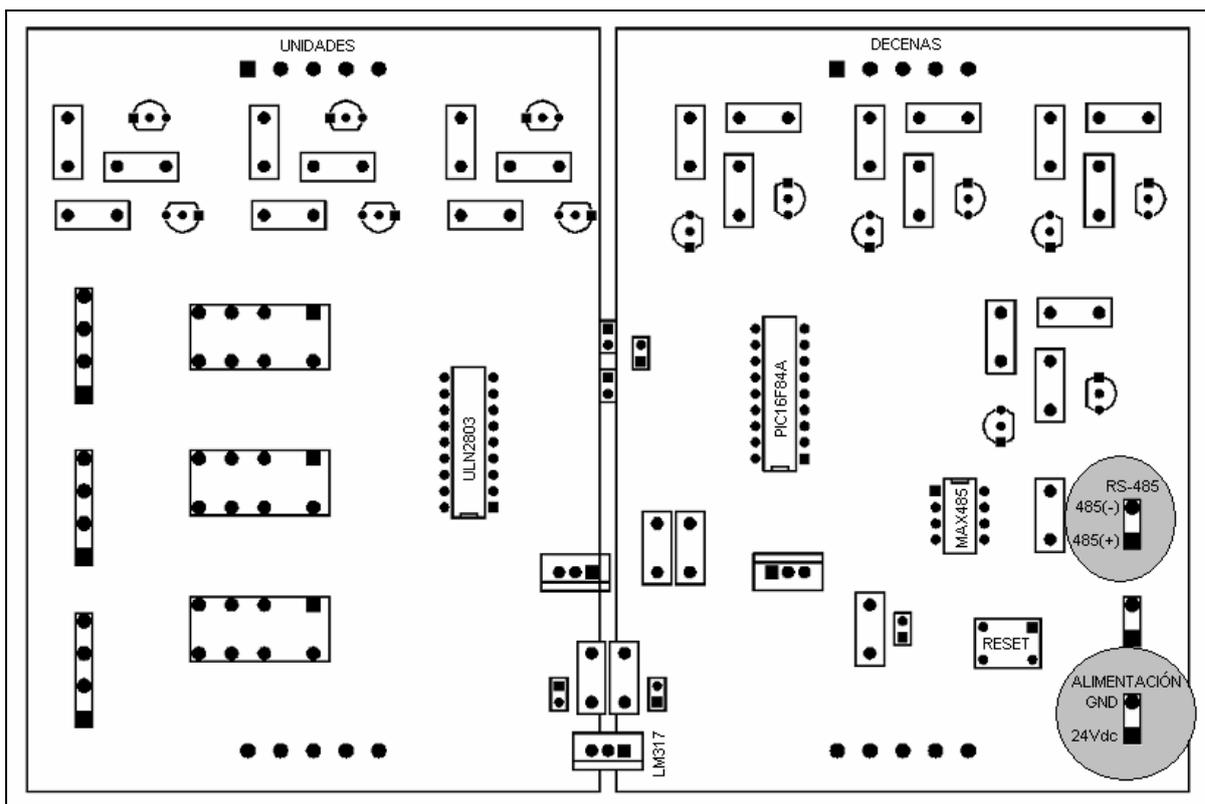


Figura A2.5. Conexiones en la Placa del Módulo de Displays

12.4. Semáforo:

El semáforo constituye la señal de alerta visual para el conductor del trolebús, para mostrarle que ha transcurrido su tiempo de estadía en la parada y que puede cerrar las puertas y continuar su recorrido.

Como se puede apreciar en la Figura A2.2., se necesitan 3 cables para la conexión del semáforo, son los siguientes:

1. Luz roja.
2. Luz verde.
3. Neutro de la alimentación de 110V_{AC}.

Deben realizarse las conexiones del semáforo de acuerdo a lo indicado en la Figura A2.6.

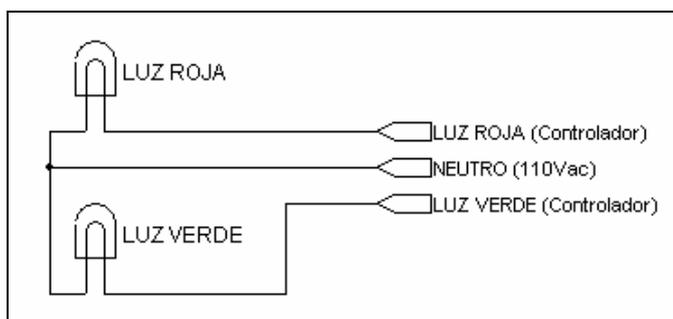


Figura A2.6. Esquema de la Conexión del Semáforo.

Las señales mostradas en la Figura A2.6. son:

1. LUZ ROJA (Controlador). Señal de activación de la luz roja, proveniente del controlador principal.
2. NEUTRO (110Vac). Neutro de la alimentación de los 110V_{AC}, la fase se encuentra en el controlador principal y es éste último el encargado de cerrar el circuito de activación de las luces del semáforo.
3. LUZ VERDE (Controlador). Señal de activación de la luz verde, proveniente del controlador principal.

12.5. Módulo de Voz:

El módulo de voz es el que almacena los mensajes de advertencia que dispone el sistema, es decir, el mensaje de apertura de puertas y el de cierre de puertas.

Como se puede apreciar en la Figura A2.2., se necesitan 6 cables para la conexión del módulo de voz con el controlador principal, son los siguientes:

1. Activación del mensaje 1 (MSG1).
2. Activación del mensaje 2 (MSG2).
3. Modo de reproducción de mensajes (PLAY).
4. Modo de grabación de los mensajes (REC).
5. Voltaje de alimentación del módulo (5Vdc).
6. Tierra del sistema (GND).

La Figura A2.7. muestra la ubicación de estas señales en el módulo voz.

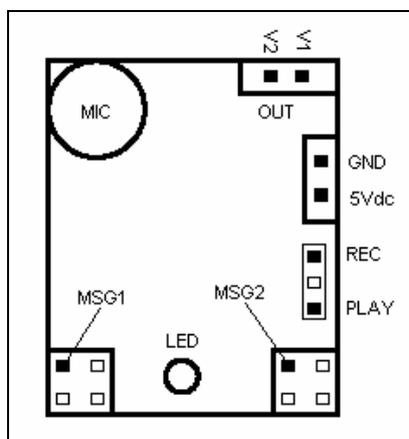


Figura A2. 7. Conexiones en la Placa del Módulo de Voz

Adicionalmente, la salida del módulo de voz (OUT, en la Figura A2.7.) debe ser conectada a un Transformador de Audio, para poder acoplar correctamente el módulo de voz con el amplificador de audio. La Figura A2.8. muestra el circuito de acoplamiento.



Figura A2.8. Acoplamiento entre el Módulo de Voz y el Amplificador de Audio

Las señales mostradas en la Figura A2.8. son las siguientes:

1. **V1.** Salida 1 del módulo de voz.
2. **GND.** Tierra del sistema.
3. **V2.** Salida 2 del módulo de voz.
4. **CH1.** Salida de audio 1.
5. **CH2.** Salida de audio 2.

Adicionalmente, se recomienda reubicar el micrófono (MIC, en la Figura A2.7.) en un lugar accesible para el momento de realizar la grabación de los mensajes de voz.

12.6. Amplificador de Audio:

El amplificador de audio se encarga de proporcionar una señal audible a los usuarios del Sistema Trolebús. Como se puede apreciar en la Figura A2.2., el amplificador tiene 6 entradas y 3 salidas, las cuales son:

1. **VA.** Voltaje de entrada (negativo).
2. **GND.** Tierra del sistema (fuente de alimentación).
3. **VB.** Voltaje de entrada (positivo).
4. **L-IN.** Entrada de audio al canal 1.
5. **GND.** Tierra del sistema (entrada de audio).
6. **R-IN.** Entrada de audio al canal 2.
7. **LS-L.** Salida 1 del amplificador.
8. **LS-R.** Salida 2 del amplificador.

La Figura A2.9. muestra las conexiones a realizarse en el amplificador de audio.

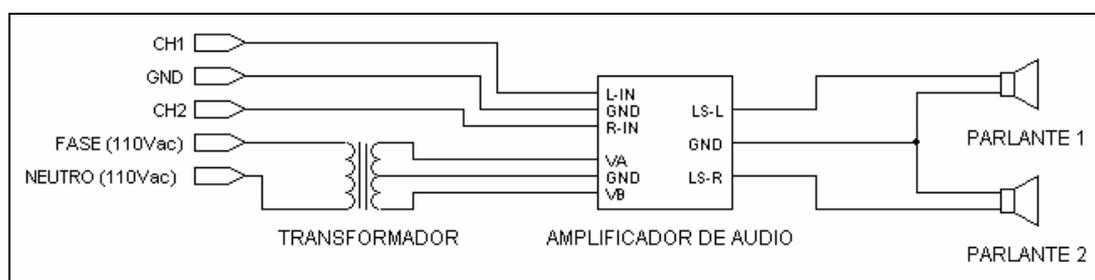


Figura A2.9. Conexiones del Amplificador de Audio.

Las etiquetas de las entradas adicionales corresponden a:

1. **CH1.** Salida 1 del transformador de audio.
2. **GND.** Tierra del sistema.
3. **CH2.** Salida 2 del transformador de audio.
4. **FASE (110Vac).** Entrada de voltaje al transformador, para la fuente del amplificador.
5. **NEUTRO (110Vac).** Entrada de voltaje al transformador, para la fuente del amplificador.

Los parlantes serán conectados en las salidas del amplificador, es decir en las señales **LS-L** y **LS-R**, además de conectar el cátodo de los parlantes a la tierra del sistema.

INSTALACIÓN DEL SOFTWARE:

La documentación entregada a la Unidad Operadora del Sistema Trolebús incluye un disco en el que se encuentra dicha documentación y todo el software necesario para poder manipular el Sistema de Alerta en su totalidad.

13.1. *Instalación del Software Administrativo.*

Para instalar la aplicación se debe ejecutar el programa **SETUP**, que es el método de publicación desarrollado, este programa instalador verificará los requerimientos de software, en caso de no cumplir con estos, se procede a la descarga automática desde el sitio web del vendedor del paquete. Si ya se posee los paquetes de software requerido, es aconsejable realizar su instalación antes de ejecutar el programa **SETUP**.

El programa de instalación viene acompañado con el manifiesto de publicación, en donde se encuentra información sobre el autor y la versión del paquete.

Estos archivos se encuentran en CD entregado a la UOST, en la ruta:

D:\ESPE-UOST\Software\Software Administrativo

13.2. *Descarga del Software del Microcontrolador PIC16F84A.*

Para la operación de los módulos de displays, es imprescindible descargar el software desarrollado para el PIC16F84A.

Dicho software se entrega en el CD adjunto, este es un archivo con extensión *.HEX, y está nombrado como: **micro.hex**

Este archivo se encuentra en la ruta:

D:\ESPE-UOST\Software\Software PIC16F84A

El software necesario para la descarga del programa es: **WINPIC800, IC-PROG** o cualquier software desarrollado para grabar microcontroladores PIC.

13.3. *Descarga del Software del Controlador Principal.*

Para la operación del Sistema de Alerta para el Cierre de Puertas de las Unidades del Sistema Trolebús, es preciso descargar el software desarrollado para el Controlador Principal.

Dicho software se entrega en el CD adjunto, este es un archivo con extensión *.BIN, y está nombrado como: **s_alerta.bin**

Este archivo se encuentra en la ruta:

D:\ESPE-UOST\Software\Software Controlador Principal

El software necesario para la descarga del programa es: **DC 9.25 RFU**. La Figura A2.10. muestra la pantalla principal de dicho software. Para la descarga del programa del Controlador Principal se sigue la ruta: "File \ Load Flash Image...". Seguidamente se abre un cuadro de diálogos donde se selecciona el archivo *s_alerta.bin* de la ubicación ya mencionada dentro del CD. Para iniciar la descarga se hace click en el botón "Ok".

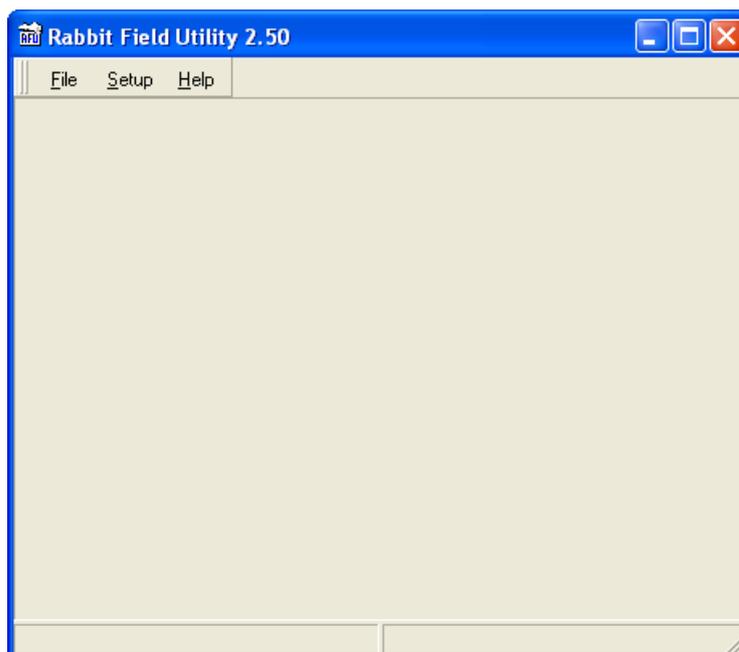


Figura A2.10. Pantalla Principal del Software DC 9.25 RFU

GUÍA DE SOLUCIÓN DE PROBLEMAS:

La Tabla A2.3. resume varios problemas y soluciones que pueden ser presentados durante la puesta en marcha y la operación del Sistema de Alerta para el Cierre de Puertas de la Unidades del Sistema Trolebús.

Tabla A2.3. Guía de Solución de Problemas

SÍNTOMA	POSIBLE PROBLEMA	SOLUCIÓN
El sistema no emite los mensajes de voz.	La alimentación no está conectada.	Asegúrese que le llega la alimentación adecuada al sistema.
	El volumen está demasiado bajo.	Suba el volumen con la perilla destinada al ajuste del mismo.
	Los mensajes de voz no están activados o grabados.	Active los mensajes en el controlador principal, en la ruta: <Mensajes de voz \ Mensaje de Apertura (Mensaje de Cierre) \ Habilitar Mensaje> y presione <Enter (↵)>
	Los parlantes no están conectados.	Asegúrese que las conexiones del elemento están realizadas de manera apropiada.
	El amplificador de audio no está conectado.	
	El módulo de voz no está conectado.	
El sistema no realiza la cuenta.	El módulo de Displays no está alimentado.	Asegúrese que le llegan 24V _{DC} al módulo de Displays.
	La red RS-485 no está conectada.	Asegúrese que la red RS-485 está conectada

		apropiadamente.
	El sensor está desconectado.	Asegúrese que el sensor está conectado y que opera correctamente.
	El microcontrolador está desprogramado.	Asegúrese que el microcontrolador PIC16F84A contiene el programa respectivo.
El semáforo no está trabajando.	La alimentación no está conectada.	Asegúrese que le llega la alimentación adecuada al sistema y al semáforo.
	El interruptor de Encendido/Apagado del semáforo está en la posición de apagado.	Cambie de posición el interruptor de Encendido/Apagado del semáforo.
	El semáforo no está conectado.	Asegúrese que el semáforo está conectado correctamente.
	Los focos dentro del semáforo están quemados.	Reemplace el o los focos del semáforo.
	El relé destinado al control del semáforo está dañado.	Reemplace el relé de la placa del controlador principal.
Los indicadores visuales oscilan.	La red RS-485 está mal conectada.	Intercambie los cables de la red RS-485
El controlador principal está encendido, pero no lo está el resto del sistema.	La alimentación del sistema está realizada incorrectamente.	Asegúrese que le llega el voltaje adecuado al elemento que está apagado.
No se presenta nada en la pantalla del controlador principal.	El controlador principal está desprogramado.	Asegúrese que el controlador principal contiene el programa correspondiente.
El sistema está encendido pero no opera.	El controlador principal está desprogramado.	Asegúrese que el controlador principal contiene el programa correspondiente.
	El sensor está desconectado.	Asegúrese que el sensor está conectado y que opera correctamente.
No se puede establecer conexión con el Computador (No se puede cargar ni descargar datos hacia o desde el controlador principal).	La alimentación no está conectada.	Asegúrese que le llega la alimentación adecuada al sistema, o al menos al controlador principal.
	No está conectado el controlador principal con el Computador	Asegúrese que el cable serial está conectado adecuadamente.
	El puerto serial (COM) es incorrecto.	Asegúrese de haber asignado correctamente el puerto serial.

ASISTENCIA TÉCNICA:

Para consultas y soporte técnico puede comunicarse con:

- Juan Carlos Flores.
Telf: 09-7199-594 e-mail: jcacmil@gmail.com

ANEXO 3

**CÓDIGO DESARROLLADO PARA EL COMPUTADOR PERSONAL,
EN MICROSOFT VISUAL BASIC 2005 EXPRESS EDITION**

FORMULARIO “Acerca de”

Public NotInheritable Class About

```
Private Sub About_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    LabelVersion.Text = String.Format("Version {0}", My.Application.Info.Version.ToString)
    LabelCopyright.Text = "Copyright " + Chr(169) + " 2007"
End Sub
```

```
Private Sub OKButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles OKButton.Click
    Me.Close()
End Sub
```

End Class

FORMULARIO “Cambiar_Password”

Public Class Cambiar_Password

```
Private Sub Cambiar_Password_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    txt_usuario_act.Text = My.Settings.User
End Sub
```

```
Private Sub Aceptar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Aceptar.Click
    If txt_usuario_act.Text = My.Settings.User And txt_pass_act.Text = My.Settings.Password
Then
        If txt_usuario_new.Text <> "" Then
            My.Settings.User = txt_usuario_new.Text
        End If
```

```
        If txt_pass_new.Text = txt_pass_conf.Text Then
            If txt_pass_new.Text.Length < 5 Then
                MsgBox("La Nueva Contraseña debe ser de al menos 5 caracteres",
MsgBoxStyle.Exclamation, "Cambiar Contraseña")
            Else
                My.Settings.Password = txt_pass_new.Text
                MsgBox("Se ha cambiado correctamente el Usuario y Contraseña.",
MsgBoxStyle.Information, "Cambiar Contraseña")
                Me.Close()
            End If
```

```
        Else
            MsgBox("La confirmación de la Contraseña no es correcta." + Chr(13) + "Confirme los
Datos.", MsgBoxStyle.Information, "Cambiar Contraseña")
        End If
```

```
    Else
        MsgBox("Usuario y/o Contraseña Actual incorrecto" + Chr(13) + "Verifique los datos y
vuelva a intentarlo", MsgBoxStyle.Information, "Cambiar Usuario y/o Password")
    End If
End Sub
```

```
Private Sub Cancelar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Cancelar.Click
    Me.Close()
End Sub

End Class
```

FORMULARIO "Conexion"

```
Public Class Conexion
```

```
Private Sub Conexion_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
    Dim Puertos() As String
    Dim i, j As Integer
    Dim n As Integer = 0

    'Muestra todos los puertos COM disponibles.
    Puertos = System.IO.Ports.SerialPort.GetPortNames()
    For i = 0 To Puertos.Length - 1
        For j = i + 1 To Puertos.Length - 1
            If Puertos(i) = Puertos(j) Then
                n = n + 1
            End If
        Next
    Next

    If n > 0 Then
        n = 0
        combo_puerto.Items.Add(Puertos(i))
    End If
Next
End Sub
```

```
Private Sub combo_puerto_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles combo_puerto.SelectedIndexChanged
    Principal.SerialPort1.PortName = combo_puerto.Text
End Sub
```

```
Private Sub cmd_aceptar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles cmd_aceptar.Click
    If prueba_conexion() = True Then
        MsgBox("Puerto " + Principal.SerialPort1.PortName + " inicializado.",
MsgBoxStyle.Information, "Conexión")
        Principal.Nuevo.Enabled = True
        Principal.Abrir.Enabled = True
        Principal.GuardarArchivo.Enabled = True
        Principal.ActualizarRTC.Enabled = True
        Principal.GuardarYDescargar.Enabled = True
        Principal.Descargar2Controlador.Enabled = True

        Principal.NuevoToolStripMenuItem.Enabled = True
        Principal.AbrirToolStripMenuItem.Enabled = True
        Principal.GuardarArchivoToolStripMenuItem.Enabled = True
        Principal.ActualizarFechaYHoraToolStripMenuItem.Enabled = True
        Principal.DescargarAlControladorToolStripMenuItem.Enabled = True
        Principal.GuardarArchivoYDescargarAlControladorToolStripMenuItem.Enabled = True
        Principal.DescargarContraseñaParaElControladorToolStripMenuItem.Enabled = True
    End If
End Sub
```

```
Else
    MsgBox("Puerto " + Principal.SerialPort1.PortName + " no disponible. Verificar las
conexiones.", MsgBoxStyle.Information, "Conexión")
End If

Me.Close()
End Sub

Private Sub cmd_cancelar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmd_cancelar.Click
    Me.Close()
End Sub

End Class
```

FORMULARIO "Inicio de Sesión"

```
Public Class Inicio_de_Sesion

    Dim intentos As Integer = 3

    Private Sub cmd_Aceptar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmd_Aceptar.Click
        If txt_Password.Text = My.Settings.Password And txt_User.Text = My.Settings.User Then
            Me.Hide()
            Principal.Show()
        Else
            MsgBox("USUARIO Y/O CONTRASEÑA INCORRECTA.." + Chr(13) + "Sistema de Alerta
para el Cierre de Puertas de las Unidades del Sistema Trolebús" + Chr(13) + "Unidad Operadora
del Sistema Trolebús - Escuela Politécnica del Ejército" + Chr(13) + "Intentos Restantes: " +
(intentos - 1).ToString, MsgBoxStyle.Information, "Inicio de Sesión")
            intentos -= 1
            If intentos = 0 Then
                Me.Close()
            End If
        End If
    End Sub

    Private Sub cmd_Cancelar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmd_Cancelar.Click
        Me.Close()
    End Sub

End Class
```

FORMULARIO "Modif_Año"

```
Public Class Modif_año

    Dim año_1 As Integer

    Private Sub Modif_año_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
        año.Minimum = Now.Year
        año.Maximum = 3000
        año.Value = Principal.Calendario_inicio.MinDate.Year
    End Sub

End Class
```

```
    anio_1 = anio.Value  
End Sub
```

```
Private Sub cmd_aceptar_Click_1(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles cmd_aceptar.Click
```

```
    Principal.Titulo_anio.Text = "AÑO: " + anio.Value.ToString
```

```
    Principal.Calendario_inicio.MinDate = "01/01/" + Now.Year.ToString
```

```
    Principal.Calendario_fin.MinDate = "01/01/" + Now.Year.ToString
```

```
    Principal.dia_Feriado.MinDate = "01/01/" + Now.Year.ToString
```

```
If Principal.Calendario_inicio.MinDate.Year > Principal.Calendario_inicio.MaxDate.Year Then
```

```
    Principal.Calendario_inicio.MinDate = "01/01/" + anio.Value.ToString
```

```
    Principal.Calendario_inicio.MaxDate = "31/12/" + anio.Value.ToString
```

```
    Principal.Calendario_fin.MinDate = "01/01/" + anio.Value.ToString
```

```
    Principal.Calendario_fin.MaxDate = "31/12/" + anio.Value.ToString
```

```
    Principal.dia_Feriado.MinDate = "01/01/" + anio.Value.ToString
```

```
    Principal.dia_Feriado.MaxDate = "31/12/" + anio.Value.ToString
```

```
Else
```

```
    Principal.Calendario_inicio.MaxDate = "31/12/" + anio.Value.ToString
```

```
    Principal.Calendario_inicio.MinDate = "01/01/" + anio.Value.ToString
```

```
    Principal.Calendario_fin.MaxDate = "31/12/" + anio.Value.ToString
```

```
    Principal.Calendario_fin.MinDate = "01/01/" + anio.Value.ToString
```

```
    Principal.dia_Feriado.MaxDate = "31/12/" + anio.Value.ToString
```

```
    Principal.dia_Feriado.MinDate = "01/01/" + anio.Value.ToString
```

```
End If
```

```
' Carga los días Feriados por defecto
```

```
Principal.Lista_Feriados.Items.Clear()
```

```
Cargar_Feriado("1/1/" + anio.Value.ToString) 'Año Nuevo
```

```
Cargar_Feriado("27/2/" + anio.Value.ToString) 'Batalla de Tarqui
```

```
Cargar_Feriado("1/4/" + anio.Value.ToString) 'Dia del Maestro
```

```
Cargar_Feriado("24/5/" + anio.Value.ToString) 'Batalla de Pichincha
```

```
Cargar_Feriado("10/8/" + anio.Value.ToString) 'Primer Grito de la Independencia
```

```
Cargar_Feriado("9/10/" + anio.Value.ToString) 'Independencia de Guayaquil
```

```
Cargar_Feriado("2/11/" + anio.Value.ToString) 'Día de Difuntos
```

```
Cargar_Feriado("3/11/" + anio.Value.ToString) 'Independencia de Cuenca
```

```
Cargar_Feriado("6/12/" + anio.Value.ToString) 'Fundación de Quito
```

```
Cargar_Feriado("25/12/" + anio.Value.ToString) 'Navidad
```

```
Cargar_Feriado("31/12/" + anio.Value.ToString) 'Fin de Año
```

```
' Limpia la lista de Dias Personalizados
```

```
If anio_1 <> anio.Value Then
```

```
    Principal.Lista_Personalizados.Items.Clear()
```

```
End If
```

```
Me.Close()
```

```
End Sub
```

```
Private Sub cmd_cancelar_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles cmd_cancelar.Click
```

```
    Me.Close()
```

```
End Sub
```

```
End Class
```

FORMULARIO "Password_Controlador"

```
Public Class Password_Controlador
```

```
    Dim password(3) As Byte  
    Dim i As Integer
```

```
    Private Sub Password_Controlador_Load(ByVal sender As Object, ByVal e As  
System.EventArgs) Handles Me.Load  
        For i = 0 To 3  
            password(i) = 0  
        Next  
        i = 0  
        Pass_Controlador.Text = ""  
    End Sub
```

```
    Private Sub Limpiar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles Limpiar.Click  
        For i = 0 To 3  
            password(i) = 0  
        Next  
        i = 0  
        Pass_Controlador.Text = ""  
    End Sub
```

```
    Private Sub Tecla_L_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles Tecla_L.Click  
        If i <= 3 Then  
            Pass_Controlador.Text += "*"   
            password(i) = 76 'L  
            i += 1  
        Else  
            MsgBox("La Contraseña para el Controlador debe ser una Combinación de 4 Teclas",  
MsgBoxStyle.Information, "Contraseña para el Controlador")  
        End If  
    End Sub
```

```
    Private Sub Tecla_U_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles Tecla_U.Click  
        If i <= 3 Then  
            Pass_Controlador.Text += "*"   
            password(i) = 85 'U  
            i += 1  
        Else  
            MsgBox("La Contraseña para el Controlador debe ser una Combinación de 4 Teclas",  
MsgBoxStyle.Information, "Contraseña para el Controlador")  
        End If  
    End Sub
```

```
    Private Sub Tecla_D_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles Tecla_D.Click  
        If i <= 3 Then  
            Pass_Controlador.Text += "*"   
            password(i) = 68 'D  
            i += 1  
        End If  
    End Sub
```

```
Else
    MsgBox("La Contraseña para el Controlador debe ser una Combinación de 4 Teclas",
    MsgBoxStyle.Information, "Contraseña para el Controlador")
End If
End Sub
```

```
Private Sub Tecla_R_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Tecla_R.Click
    If i <= 3 Then
        Pass_Controlador.Text += "*"
        password(i) = 82 'R
        i += 1
    Else
        MsgBox("La Contraseña para el Controlador debe ser una Combinación de 4 Teclas",
        MsgBoxStyle.Information, "Contraseña para el Controlador")
    End If
End Sub
```

```
Private Sub Tecla_m_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Tecla_m.Click
    If i <= 3 Then
        Pass_Controlador.Text += "*"
        password(i) = 45 '-'
        i += 1
    Else
        MsgBox("La Contraseña para el Controlador debe ser una Combinación de 4 Teclas",
        MsgBoxStyle.Information, "Contraseña para el Controlador")
    End If
End Sub
```

```
Private Sub Tecla_p_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Tecla_p.Click
    If i <= 3 Then
        Pass_Controlador.Text += "*"
        password(i) = 43 '+'
        i += 1
    Else
        MsgBox("La Contraseña para el Controlador debe ser una Combinación de 4 Teclas",
        MsgBoxStyle.Information, "Contraseña para el Controlador")
    End If
End Sub
```

```
Private Sub Tecla_E_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Tecla_E.Click
    If i <= 3 Then
        Pass_Controlador.Text += "*"
        password(i) = 69 'E
        i += 1
    Else
        MsgBox("La Contraseña para el Controlador debe ser una Combinación de 4 Teclas",
        MsgBoxStyle.Information, "Contraseña para el Controlador")
    End If
End Sub
```

```

Private Sub Descargar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Descargar.Click
    If Pass_Controlador.Text.Length = 4 Then
        'Se envía la contraseña para el Rabbit, con la función 26
        Dim respuesta() As Byte = TxSerial(26, password)

        If respuesta(0) <> 26 Or respuesta(1) <> 0 Then
            MsgBox("Error al Descargar la Contraseña para el Controlador ",
            MsgBoxStyle.Information, "Contraseña para el Controlador")
            For i = 0 To 3
                password(i) = 0
            Next
            i = 0
            Pass_Controlador.Text = ""
        Else
            MsgBox("Contraseña para el Controlador Descargada Satisfactoriamente",
            MsgBoxStyle.Information, "Contraseña para el Controlador")
            Me.Close()
        End If
    Else
        MsgBox("La Contraseña para el Controlador debe ser una Combinación de 4 Teclas",
        MsgBoxStyle.Information, "Contraseña para el Controlador")
        For i = 0 To 3
            password(i) = 0
        Next
        i = 0
        Pass_Controlador.Text = ""
    End If
End Sub
    
```

```

Private Sub Cancelar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Cancelar.Click
    Me.Close()
End Sub
    
```

End Class

FORMULARIO "Presentacion"

```

Public NotInheritable Class Presentacion
    
```

```

Private Sub Presentacion_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
    If My.Application.Info.Title <> "" Then
        ApplicationTitle.Text = My.Application.Info.Title
    Else
        'If the application title is missing, use the application name, without the extension
        ApplicationTitle.Text =
        System.IO.Path.GetFileNameWithoutExtension(My.Application.Info.AssemblyName)
    End If
    
```

```

Version.Text = System.String.Format(Version.Text, My.Application.Info.Version.Major,
My.Application.Info.Version.Minor)
Copyright.Text = "Copyright " + Chr(169) + " 2007"
End Sub
    
```

End Class

FORMULARIO "Principal_UOST"

Public Class Principal

Public Tabla_Tipo(23) As NumericUpDown ' Arreglo de Controles de la pestaña: Configuración del Tipo de Día

Dim Tabla_Personaliza(23) As NumericUpDown ' Arreglo de Controles de la pestaña: Personalizar Día

Public tabla_Ordinario(23) As Byte ' Temporizaciones de Días Ordinarios

Public tabla_FinSemana(23) As Byte ' Temporizaciones de Fines de Semana

Public tabla_Feriado(23) As Byte ' Temporizaciones de Días Feriados

Public tabla_Personalizado(365, 23) As Byte ' Temporizaciones de Días Personalizados

Dim i, j, k As Integer ' Contadores Generales

Dim cont_fecha As Date ' Variable auxiliar, contador para fechas

Dim Cambios As Boolean

Private Sub Principal_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

' Realiza el arreglo de la Tabla de Tipo de Día

For i = 0 To 11

Tabla_Tipo(i) = Tabla_Tiempo.GetControlFromPosition(1, i + 1)

Next

For i = 0 To 11

Tabla_Tipo(i + 12) = Tabla_Tiempo.GetControlFromPosition(4, i + 1)

Next

' Realiza el arreglo de la Tabla de Personalizados

For i = 0 To 11

Tabla_Personaliza(i) = Tabla_Personalizar.GetControlFromPosition(1, i + 1)

Next

For i = 0 To 11

Tabla_Personaliza(i + 12) = Tabla_Personalizar.GetControlFromPosition(4, i + 1)

Next

' Activa/Desactiva opciones que necesitan requisitos previos

GuardarArchivo.Enabled = False

ActualizarRTC.Enabled = False

Descargar2Controlador.Enabled = False

GuardarYDescargar.Enabled = False

Subir4Controlador.Enabled = False

Modificar_AnioT.Enabled = False

GuardarArchivoToolStripMenuItem.Enabled = False

ActualizarFechaYHoraToolStripMenuItem.Enabled = False

ModificarAnioDeLaTablaDeTiempoToolStripMenuItem.Enabled = False

DescargarAlControladorToolStripMenuItem.Enabled = False

GuardarArchivoYDescargarAlControladorToolStripMenuItem.Enabled = False

SubirDesdeElControladorToolStripMenuItem.Enabled = False

' DescargarContraseñaParaElControladorToolStripMenuItem.Enabled = False

TabControl1.Visible = False

Titulo_anio.Visible = False

Cambios = False

d_ordinario.Checked = True

Sel_Rango.Checked = True

Sel_Rango.Checked = False

cmd_eliminar_personalizado.Enabled = False

cmd_limpiar_personalizado.Enabled = False

```
Barra.Visible = False
Descargando.Visible = False
End Sub
```

```
Private Sub Abrir_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Abrir.Click, AbrirToolStripMenuItem.Click
```

```
    ' Dim ok_m As Boolean
    Dim primera As Boolean
```

```
    If TabControl1.Visible = True Then
        primera = False
        If MsgBox("¿Desea Guardar los Cambios Realizados en el Archivo Actual?",
MsgBoxStyle.YesNo, "Abrir Archivo") = MsgBoxResult.Yes Then
            Guardar_Archivo()
        End If
    Else
        primera = True
    End If
```

```
    Open1.Filter = "Archivos Horario TROLE (*.TRO)|*.TRO"
    Open1.FilterIndex = 0
```

```
    If Open1.ShowDialog() = Windows.Forms.DialogResult.OK Then
        Dim arreglo_open() As Byte = My.Computer.FileSystem.ReadAllBytes(Open1.FileName)
        Dim ind As Integer
        Dim n_ferriados, n_personalizados As Integer
        Dim fecha_s As String
        Dim aux_s As String
        Dim aux_year As Campos
        Dim item As ListViewItem
        Dim actualizar_year As Boolean = False
        Dim year As Integer = Now.Year
```

```
        ' Habilita controles
        Cambios = True
        GuardarArchivo.Enabled = True
        Modificar_AnioT.Enabled = True
```

```
        GuardarArchivoToolStripMenuItem.Enabled = True
        ModificarAnioDeLaTablaDeTiempoToolStripMenuItem.Enabled = True
        TabControl1.Visible = True
        Titulo_anio.Visible = True
```

```
        Lista_Ferriados.Items.Clear()
        Lista_Personalizados.Items.Clear()
        d_ordinario.Checked = True
```

```
        ' Realiza la apertura del Archivo
        ind = 0
        ' Carga las Temporizaciones de los Dias Ordinarios
        For i = 0 To 23
            tabla_Ordinario(i) = arreglo_open(ind)
            Tabla_Tipo(i).Value = tabla_Ordinario(i)
            ind += 1
        Next
```

```
        ' Carga las Temporizaciones de los Fines de Semana
        For i = 0 To 23
            tabla_FinSemana(i) = arreglo_open(ind)
            ind += 1
```

```
Next

' Carga las Temporizaciones de los Días Feriados
For i = 0 To 23
    tabla_Feriado(i) = arreglo_open(ind)
    ind += 1
Next

' Carga Numero de Días Feriados
n_feriados = arreglo_open(ind)
ind += 1

' Carga las Fechas de los Días Feriados
If n_feriados > 0 Then
    cmd_eliminar.Enabled = True
    cmd_limpiar.Enabled = True
    For i = 0 To n_feriados - 1
        fecha_s = ""
        fecha_s = arreglo_open(ind).ToString + "/" & " " & Dia
        ind += 1
        fecha_s += arreglo_open(ind).ToString + "/" & " " & Mes
        ind += 1
        aux_year.dato1 = arreglo_open(ind)
        ind += 1
        aux_year.dato2 = arreglo_open(ind)
        ind += 1
        aux_year.dato3 = arreglo_open(ind)
        ind += 1
        fecha_s += aux_year.datoL.ToString & " Año

        item = New ListViewItem(Lista_Feriados.Items.Count() + 1.ToString)
        item.SubItems.Add(fecha_s)
        Lista_Feriados.Items.Add(item)
    Next
    year = aux_year.datoL & " Año a ser cargado en el formulario Modif_año
Else
    cmd_eliminar.Enabled = False
    cmd_limpiar.Enabled = False
    actualizar_year = True
End If

' Carga Numero de Días Personalizados
n_personalizados = arreglo_open(ind)
ind += 1

' Carga Temporizaciones de los Días Personalizados
If n_personalizados > 0 Then
    For i = 0 To n_personalizados - 1
        For j = 0 To 23
            tabla_Personalizado(i, j) = arreglo_open(ind)
            ind += 1
        Next
    Next
End If

' Carga las Fechas de los Días Personalizados
If n_personalizados > 0 Then
    cmd_eliminar_personalizado.Enabled = True
    cmd_limpiar_personalizado.Enabled = True
    For i = 0 To n_personalizados - 1
```

```
    fecha_s = ""
    fecha_s = arreglo_open(ind).ToString + "/" + Dia
    ind += 1
    fecha_s += arreglo_open(ind).ToString + "/" + Mes
    ind += 1
    aux_year.dato1 = arreglo_open(ind)
    ind += 1
    aux_year.dato2 = arreglo_open(ind)
    ind += 1
    aux_year.dato3 = arreglo_open(ind)
    ind += 1
    fecha_s += aux_year.datoL.ToString + " Año"

    item = New ListViewItem(Lista_Personalizados.Items.Count() + 1.ToString)
    item.SubItems.Add(fecha_s)

    aux_s = ""
    For j = 0 To 23
        aux_s += tabla_Personalizado(i, j).ToString + "."
    Next
    item.SubItems.Add(aux_s)

    Lista_Personalizados.Items.Add(item)
Next

If actualizar_year Then
    year = aux_year.datoL + " Año a ser cargado en el formulario Modif_anio"
End If
Else
    cmd_eliminar_personalizado.Enabled = False
    cmd_limpiar_personalizado.Enabled = False
    Calendario_inicio.MinDate = "01/01/" + Now.Year.ToString
    Titulo_anio.Text = "AÑO: " + Now.Year.ToString
End If

' Modifica el año del Programa, de ser necesario
If Calendario_inicio.MinDate.Year <> year Then
    Calendario_inicio.MinDate = "01/01/" + year.ToString
    Modif_anio.anio.Value = year
    Titulo_anio.Text = "AÑO: " + year.ToString

    Calendario_inicio.MinDate = "01/01/" + year.ToString
    Calendario_fin.MinDate = "01/01/" + year.ToString
    dia_Feriado.MinDate = "01/01/" + year.ToString

    If Calendario_inicio.MinDate.Year > Calendario_inicio.MaxDate.Year Then
        Calendario_inicio.MinDate = "01/01/" + Modif_anio.anio.Value.ToString
        Calendario_inicio.MaxDate = "31/12/" + Modif_anio.anio.Value.ToString
        Calendario_fin.MinDate = "01/01/" + Modif_anio.anio.Value.ToString
        Calendario_fin.MaxDate = "31/12/" + Modif_anio.anio.Value.ToString
        dia_Feriado.MinDate = "01/01/" + Modif_anio.anio.Value.ToString
        dia_Feriado.MaxDate = "31/12/" + Modif_anio.anio.Value.ToString
    End If
    If Calendario_inicio.MinDate.Year < Calendario_inicio.MaxDate.Year Then
        Calendario_inicio.MaxDate = "31/12/" + Modif_anio.anio.Value.ToString
        Calendario_inicio.MinDate = "01/01/" + Modif_anio.anio.Value.ToString
        Calendario_fin.MaxDate = "31/12/" + Modif_anio.anio.Value.ToString
        Calendario_fin.MinDate = "01/01/" + Modif_anio.anio.Value.ToString
        dia_Feriado.MaxDate = "31/12/" + Modif_anio.anio.Value.ToString
        dia_Feriado.MinDate = "01/01/" + Modif_anio.anio.Value.ToString
    End If
End If
```

```
End If
End If
End If
End Sub
```

```
Private Sub Nuevo_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Nuevo.Click, NuevoToolStripMenuItem.Click
```

```
Dim ok_m As Boolean
Dim ok As Boolean = True
```

```
cmd_eliminar.Enabled = True
cmd_limpiar.Enabled = True
```

```
If TabControl1.Visible = True Then
If MsgBox("¿Desea Guardar los Cambios Realizados en el Archivo Actual?",
MsgBoxStyle.YesNo, "Nuevo Archivo") = MsgBoxResult.Yes Then
```

```
Guardar_Archivo()
ok_m = True
```

```
Else
ok_m = False
```

```
End If
```

```
Else
ok = True
```

```
End If
```

```
If ok Or ok_m Then
```

```
' Habilita controles
Cambios = True
GuardarArchivo.Enabled = True
Modificar_AnioT.Enabled = True
```

```
GuardarArchivoToolStripMenuItem.Enabled = True
ModificarAnioDeLaTablaDeTiempoToolStripMenuItem.Enabled = True
```

```
TabControl1.Visible = True
Titulo_anio.Visible = True
Modif_anio.anio.Value = Now.Year
Titulo_anio.Text = "AÑO: " + Now.Year.ToString
```

```
' Inicializa los arregos de temporizaciones
```

```
For i = 0 To 23
```

```
tabla_Ordinario(i) = 10
tabla_FinSemana(i) = 12
tabla_Feriado(i) = 14
```

```
For j = 0 To 365
```

```
tabla_Personalizado(j, i) = 0
```

```
Next
```

```
Tabla_Personaliza(i).Value = tabla_Personalizado(0, i)
```

```
Tabla_Tipo(i).Value = tabla_Ordinario(i)
```

```
Next
```

```
' Borra el contenido de las Listas
```

```
Lista_Feriados.Items.Clear()
```

```
Lista_Personalizados.Items.Clear()
```

```
' Limita los calendarios al año actual
```

```
Calendario_inicio.MinDate = "01/01/" + Now.Year.ToString
```

```
Calendario_inicio.MaxDate = "31/12/" + Now.Year.ToString
```

```
Calendario_fin.MinDate = "01/01/" + Now.Year.ToString
```

```
    Calendario_fin.MaxDate = "31/12/" + Now.Year.ToString
    dia_Feriado.MinDate = "01/01/" + Now.Year.ToString
    dia_Feriado.MaxDate = "31/12/" + Now.Year.ToString

    ' Carga los días Feriados por defecto
    Cargar_Feriado("1/1/" + Now.Year.ToString) 'Año Nuevo
    Cargar_Feriado("27/2/" + Now.Year.ToString) 'Batalla de Tarqui
    Cargar_Feriado("1/4/" + Now.Year.ToString) 'Dia del Maestro
    Cargar_Feriado("24/5/" + Now.Year.ToString) 'Batalla de Pichincha
    Cargar_Feriado("10/8/" + Now.Year.ToString) 'Primer Grito de la Independencia
    Cargar_Feriado("9/10/" + Now.Year.ToString) 'Independencia de Guayaquil
    Cargar_Feriado("2/11/" + Now.Year.ToString) 'Día de Difuntos
    Cargar_Feriado("3/11/" + Now.Year.ToString) 'Independencia de Cuenca
    Cargar_Feriado("6/12/" + Now.Year.ToString) 'Fundación de Quito
    Cargar_Feriado("25/12/" + Now.Year.ToString) 'Navidad
    Cargar_Feriado("31/12/" + Now.Year.ToString) 'Fin de Año
End If
End Sub

Private Sub ModificarAnioDeLaTablaDeTiempoToolStripMenuItem_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles ModificarAnioDeLaTablaDeTiempoToolStripMenuItem.Click, Modificar_AnioT.Click
    Modif_anio.Show()
End Sub

Private Sub ActualizarFechaYHoraToolStripMenuItem_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles ActualizarFechaYHoraToolStripMenuItem.Click, ActualizarRTC.Click
    ' Se obtiene una cadena con la fecha y hora actual
    Dim fecha() As Byte = {Now.Second(), Now.Minute(), Now.Hour(), Now.Day(), Now.Month(),
    (Now.Year() - 1900), Now.DayOfWeek()}

    'Se envía los datos de hora y fecha al Rabbit, con la función 10
    Dim respuesta() As Byte = TxSerial(10, fecha)

    If respuesta(1) = 0 Then
        MsgBox("Fecha y Hora actualizadas", MsgBoxStyle.Information, "Fecha y Hora")
    Else
        ' MsgBox("Error en Actualizacion de Fecha y Hora", MsgBoxStyle.Information,
"Grabación")
    End If
End Sub

Private Sub ConexionToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ConexionToolStripMenuItem.Click, b_Conexion.Click
    Conexion.Show()
End Sub

' Día Ordinario
Private Sub d_ordinario_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles d_ordinario.CheckedChanged
    If d_ordinario.Checked = True Then
        titulo.Text = "DÍA ORDINARIO"
        For i = 0 To 23
            Tabla_Tipo(i).Value = tabla_Ordinario(i)
        Next
    End If
End Sub
```

```
End If

If d_ordinario.Checked = False Then
    For i = 0 To 23
        tabla_Ordinario(i) = Tabla_Tipo(i).Value
    Next
End If
End Sub

' Fin de Semana
Private Sub f_semana_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles f_semana.CheckedChanged
    If f_semana.Checked = True Then
        titulo.Text = "FIN DE SEMANA"
        For i = 0 To 23
            Tabla_Tipo(i).Value = tabla_FinSemana(i)
        Next
    End If

    If f_semana.Checked = False Then
        For i = 0 To 23
            tabla_FinSemana(i) = Tabla_Tipo(i).Value
        Next
    End If
End Sub

' Feriado
Private Sub d_feriado_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles d_feriado.CheckedChanged
    If d_feriado.Checked = True Then
        titulo.Text = "DÍA FERIADO"
        For i = 0 To 23
            Tabla_Tipo(i).Value = tabla_Feriado(i)
        Next
    End If

    If d_feriado.Checked = False Then
        For i = 0 To 23
            tabla_Feriado(i) = Tabla_Tipo(i).Value
        Next
    End If
End Sub

Private Sub dia_Feriado_CloseUp(ByVal sender As Object, ByVal e As System.EventArgs)
Handles dia_Feriado.CloseUp
    Dim item As ListViewItem
    Dim fecha_sel As Date
    Dim aumentar As Boolean = True

    cmd_limpiar.Enabled = True
    cmd_eliminar.Enabled = True

    fecha_sel = dia_Feriado.Value

    For i = 0 To Lista_Feriados.Items.Count() - 1
        If (fecha_sel.Day.ToString + "/" + fecha_sel.Month.ToString + "/" + fecha_sel.Year.ToString)
= Lista_Feriados.Items(i).SubItems(1).Text Then
```

```
        aumentar = False
    End If
Next

    If aumentar = True Then
        item = New ListViewItem(Lista_Feriados.Items.Count() + 1.ToString)
        item.SubItems.Add(dia_Feriado.Value.Day.ToString + "/" + dia_Feriado.Value.Month.ToString + "/" + dia_Feriado.Value.Year.ToString)
        Lista_Feriados.Items.Add(item)
        If Lista_Feriados.Items.Count() > 1 Then
            Lista_Feriados.ListViewItemSorter = New ListViewItemComparer(1) 'Ordena los items por Fecha
            For i = 0 To Lista_Feriados.Items.Count() - 1
                Lista_Feriados.Items(i).SubItems(0).Text = (i + 1).ToString 'Asigna un nuevo ID a todos los elementos de la lista
            Next
        End If
    End If

    ' Elimina los datos de la Lista de dias Personalizados que se agregan a la Lista de dias Feriados
    i = 0
    If Lista_Feriados.Items.Count > 0 Then
        Do While i <= Lista_Personalizados.Items.Count - 1
            For j As Integer = 0 To Lista_Feriados.Items.Count - 1
                If Lista_Personalizados.Items(i).SubItems(1).Text = Lista_Feriados.Items(j).SubItems(1).Text Then
                    Lista_Personalizados.Items(i).Remove()
                    i = -1
                    If Lista_Personalizados.Items.Count() > 1 Then
                        Lista_Personalizados.ListViewItemSorter = New ListViewItemComparer(1) 'Ordena los items por Fecha
                    End If
                    For h As Integer = 0 To Lista_Personalizados.Items.Count() - 1
                        Lista_Personalizados.Items(h).SubItems(0).Text = (h + 1).ToString 'Asigna un nuevo ID a todos los elementos de la lista
                    Next
                End If
            Exit For
        End If
    Next
    i += 1
Loop
End If

' Activa/Desactiva los Botones, segun corresponda
If Lista_Feriados.Items.Count() > 0 Then
    cmd_eliminar.Enabled = True
Else
    cmd_eliminar.Enabled = False
End If
End Sub

Private Sub Lista_Feriados_ColumnClick(ByVal sender As Object, ByVal e As System.Windows.Forms.ColumnClickEventArgs) Handles Lista_Feriados.ColumnClick
    Lista_Feriados.ListViewItemSorter = New ListViewItemComparer(e.Column)

    For i As Integer = 0 To Lista_Feriados.Items.Count() - 1
        Lista_Feriados.Items(i).SubItems(0).Text = (i + 1).ToString
    Next
End Sub
```

```
Next  
End Sub
```

```
Private Sub cmd_eliminar_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles cmd_eliminar.Click  
    Dim indices As ListView.SelectedListViewItemCollection = Lista_Feriados.SelectedItems  
    'Obtiene los items seleccionados  
  
    If indices.Count() Then 'Si existe un item seleccionado  
        Lista_Feriados.Items.Remove(indices(0)) 'Remueve el item del listview  
        If Lista_Feriados.Items.Count() > 1 Then  
            Lista_Feriados.ListViewItemSorter = New ListViewItemComparer(1) 'Ordena los items  
por Fecha  
            For i As Integer = 0 To Lista_Feriados.Items.Count() - 1  
                Lista_Feriados.Items(i).SubItems(0).Text = (i + 1).ToString 'Asigna un nuevo ID a  
todos los elementos de la lista  
            Next  
        End If  
    Else  
        MsgBox("Seleccione un elemento de la lista para eliminar", MsgBoxStyle.Exclamation,  
"Eliminar Día Feriado")  
    End If  
  
    If Lista_Feriados.Items.Count() = 0 Then  
        cmd_eliminar.Enabled = False  
        cmd_limpiar.Enabled = False  
    End If  
End Sub
```

```
Private Sub Calendario_fin_CloseUp(ByVal sender As Object, ByVal e As System.EventArgs)  
Handles Calendario_fin.CloseUp  
    If Calendario_fin.Value < Calendario_inicio.Value Then  
        MsgBox("La Fecha Final debe ser mayor o igual a la Inicial", MsgBoxStyle.Exclamation,  
"Rango Incorrecto")  
        Calendario_fin.Value = Calendario_inicio.Value  
    End If  
End Sub
```

```
Private Sub Calendario_inicio_CloseUp(ByVal sender As Object, ByVal e As System.EventArgs)  
Handles Calendario_inicio.CloseUp  
    If Sel_Rango.Checked = False Then  
        Calendario_fin.Value = Calendario_inicio.Value  
    End If  
  
    If Calendario_inicio.Value > Calendario_fin.Value Then  
        MsgBox("La Fecha Inicial debe ser menor o igual a la Final", MsgBoxStyle.Exclamation,  
"Rango Incorrecto")  
        Calendario_inicio.Value = Calendario_fin.Value  
    End If  
End Sub
```

```
Private Sub Sel_Rango_CheckedChanged_1(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Sel_Rango.CheckedChanged  
    If Sel_Rango.Checked = True Then  
        Tit_Inicial.Text = "Fecha Inicial:"  
        Calendario_fin.Enabled = True
```

```

        Calendario_fin.Value = Calendario_inicio.Value
    End If

    If Sel_Rango.Checked = False Then
        Tit_Inicial.Text = "Seleccionar Fecha:"
        Calendario_fin.Enabled = False
        Calendario_fin.Value = Calendario_inicio.Value
    End If
End Sub

Private Sub Titulo_anio_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Titulo_anio.Click
    Modif_anio.Show()
End Sub

Private Sub TabControl1_Selecting(ByVal sender As Object, ByVal e As
System.Windows.Forms.TabControlCancelEventArgs) Handles TabControl1.Selecting
    If TabControl1.SelectedIndex = 2 Then
        For i = 0 To 23
            Tabla_Personaliza(i).Value = tabla_Personalizado(0, i)
        Next
    End If

    If TabControl1.SelectedIndex = 0 Or TabControl1.SelectedIndex = 1 Then
        For i = 0 To 23
            tabla_Personalizado(0, i) = Tabla_Personaliza(i).Value
        Next
    End If
End Sub

Private Sub cmd_limpiar_personalizado_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmd_limpiar_personalizado.Click
    If MsgBox("¿Está seguro de eliminar Todas las Fechas Personalizadas?",
MsgBoxStyle.Exclamation + MsgBoxStyle.OkCancel, "Eliminar Días Personalizados") =
MsgBoxResult.Ok Then
        Lista_Personalizados.Items.Clear()
        cmd_limpiar_personalizado.Enabled = False
        cmd_eliminar_personalizado.Enabled = False
        For i = 0 To 23
            Tabla_Personaliza(i).Value = 0
        Next
    End If
End Sub

Private Sub cmd_eliminar_personalizado_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmd_eliminar_personalizado.Click
    Dim indices As ListView.SelectedListViewItemCollection =
Lista_Personalizados.SelectedItems 'Obtiene los items seleccionados
    Dim arreglo_c() As Char
    Dim arreglo_s(23) As String
    Dim ind As Integer

    If indices.Count() Then 'Si existe un item seleccionado
        Lista_Personalizados.Items.Remove(indices(0)) 'Remueve el item del listview
        If Lista_Personalizados.Items.Count() > 1 Then

```

```
        Lista_Personalizados.ListViewItemSorter = New ListViewItemComparer(1) 'Ordena los
items por Fecha
        For i As Integer = 0 To Lista_Personalizados.Items.Count() - 1
            Lista_Personalizados.Items(i).SubItems(0).Text = (i + 1).ToString 'Asigna un nuevo ID
a todos los elementos de la lista
            Next
        End If
    Else
        MsgBox("Seleccione un elemento de la lista para eliminar", MsgBoxStyle.Exclamation,
"Eliminar Día Feriado")
    End If

If Lista_Personalizados.Items.Count() = 0 Then
    cmd_eliminar_personalizado.Enabled = False
    cmd_limpiar_personalizado.Enabled = False
Else
    ' Actualiza el arreglo de Dias Personalizados con la Lista de Dias Personalizados
    For j = 0 To Lista_Personalizados.Items.Count() - 1
        ' Se hace arreglo el contenido de la Lista "Temporización", la fila j
        arreglo_c = Lista_Personalizados.Items(j).SubItems(2).Text.ToCharArray()
        ' Inicialización de variables
        ind = 0
        For i = 0 To 23
            arreglo_s(i) = "0"
        Next
        ' Separa y almacena los datos que estan separados por puntos en la Lista
(Temporizacion)
        For i = 0 To arreglo_c.Length - 1
            If arreglo_c(i) <> "." Then
                arreglo_s(ind) += arreglo_c(i)
            Else
                ind += 1
            End If
        Next
        ' Actualiza el arreglo de dias Personalizados
        For k = 0 To 23
            tabla_Personalizado(j, k) = arreglo_s(k)
        Next
    Next
End If
End Sub

Private Sub cmd_actualizar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmd_actualizar.Click
    Dim item As ListViewItem
    Dim fecha_sel As Date
    Dim aumentar As Integer
    Dim rango_f As Integer
    Dim aux_s As String

    rango_f = Calendario_fin.Value.DayOfYear - Calendario_inicio.Value.DayOfYear
    fecha_sel = Calendario_inicio.Value
    aumentar = 0

    For j = 0 To rango_f
        For i = 0 To Lista_Personalizados.Items.Count() - 1
            If (fecha_sel.Day.ToString + "/" + fecha_sel.Month.ToString + "/" +
fecha_sel.Year.ToString) = Lista_Personalizados.Items(i).SubItems(1).Text Then
                aumentar += 1 ' Fecha existente en la Lista
```

```
        Lista_Personalizados.Items(i).SubItems(2).Text = ""
        For k = 0 To 23
            tabla_Personalizado(i, k) = Tabla_Personaliza(k).Value
            Lista_Personalizados.Items(i).SubItems(2).Text += tabla_Personalizado(i,
k).ToString + "."
        Next
    End If
Next

If aumentar = 0 Then ' Se incrementan las fechas en la lista
    item = New ListViewItem(Lista_Personalizados.Items.Count() + 1.ToString)
    ' Agrega la Fecha como subítem
    item.SubItems.Add(fecha_sel.Day.ToString + "/" + fecha_sel.Month.ToString + "/" +
fecha_sel.Year.ToString)

    ' Incrementa fecha a la Lista
    For k = 0 To 23
        tabla_Personalizado(Lista_Personalizados.Items.Count(), k) =
Tabla_Personaliza(k).Value
    Next

    aux_s = ""
    For k = 0 To 23
        aux_s += tabla_Personalizado(Lista_Personalizados.Items.Count(), k).ToString + "."
    Next

    ' Agrega la Temporización como subítem
    item.SubItems.Add(aux_s)
    ' Agrega el ítem a la Lista
    Lista_Personalizados.Items.Add(item)
End If

fecha_sel = fecha_sel.AddDays(1)
aumentar = 0
Next

' Ordena los ítems por Fecha
If Lista_Personalizados.Items.Count() > 1 Then
    Lista_Personalizados.ListViewItemSorter = New ListViewItemComparer(1)
    ' Asigna un nuevo ID a todos los elementos de la lista
    For i = 0 To Lista_Personalizados.Items.Count() - 1
        Lista_Personalizados.Items(i).SubItems(0).Text = (i + 1).ToString
    Next
    Actualizar_Arreglo_Personalizados()
End If

' Elimina los datos de la Lista de días Feriados que se encuentran en la Lista de días
Personalizados
i = 0
If Lista_Personalizados.Items.Count > 0 Then
    Do While i <= Lista_Feriados.Items.Count - 1
        For j As Integer = 0 To Lista_Personalizados.Items.Count - 1
            If Lista_Feriados.Items(i).SubItems(1).Text =
Lista_Personalizados.Items(j).SubItems(1).Text Then
                Lista_Feriados.Items(i).Remove()
                i = -1
            If Lista_Feriados.Items.Count() > 1 Then
                Lista_Feriados.ListViewItemSorter = New ListViewItemComparer(1) ' Ordena los
ítems por Fecha
                For h As Integer = 0 To Lista_Feriados.Items.Count() - 1
```

```
                Lista_Feritados.Items(h).SubItems(0).Text = (h + 1).ToString 'Asigna un nuevo  
ID a todos los elementos de la lista
```

```
                Next  
            End If  
        Exit For  
    End If  
Next  
i += 1  
Loop  
End If
```

```
' Activa/Desactiva los Botones, segun corresponda  
If Lista_Personalizados.Items.Count() > 0 Then  
    cmd_eliminar_personalizado.Enabled = True  
    cmd_limpiar_personalizado.Enabled = True  
Else  
    cmd_eliminar_personalizado.Enabled = False  
    cmd_limpiar_personalizado.Enabled = False  
End If  
End Sub
```

```
Private Sub Lista_Personalizados_Click(ByVal sender As Object, ByVal e As  
System.EventArgs) Handles Lista_Personalizados.Click  
    Dim indices As ListView.SelectedListItemCollection =  
Lista_Personalizados.SelectedItems 'Obtiene los items seleccionados  
    Dim num As Integer
```

```
    Calendario_inicio.MinDate = "01/01/" + Now.Year.ToString  
    Calendario_fin.MinDate = "01/01/" + Now.Year.ToString  
    dia_Feriado.MinDate = "01/01/" + Now.Year.ToString
```

```
    If Calendario_inicio.MinDate.Year > Calendario_inicio.MaxDate.Year Then  
        Calendario_inicio.MinDate = "01/01/" + Modif_anio.anio.Value.ToString  
        Calendario_inicio.MaxDate = "31/12/" + Modif_anio.anio.Value.ToString  
        Calendario_fin.MinDate = "01/01/" + Modif_anio.anio.Value.ToString  
        Calendario_fin.MaxDate = "31/12/" + Modif_anio.anio.Value.ToString  
        dia_Feriado.MinDate = "01/01/" + Modif_anio.anio.Value.ToString  
        dia_Feriado.MaxDate = "31/12/" + Modif_anio.anio.Value.ToString  
    End If
```

```
    If Calendario_inicio.MinDate.Year < Calendario_inicio.MaxDate.Year Then  
        Calendario_inicio.MaxDate = "31/12/" + Modif_anio.anio.Value.ToString  
        Calendario_inicio.MinDate = "01/01/" + Modif_anio.anio.Value.ToString  
        Calendario_fin.MaxDate = "31/12/" + Modif_anio.anio.Value.ToString  
        Calendario_fin.MinDate = "01/01/" + Modif_anio.anio.Value.ToString  
        dia_Feriado.MaxDate = "31/12/" + Modif_anio.anio.Value.ToString  
        dia_Feriado.MinDate = "01/01/" + Modif_anio.anio.Value.ToString  
    End If
```

```
    num = indices(0).SubItems(0).Text - 1  
    Sel_Rango.Checked = False  
    Tit_Inicial.Text = "Seleccionar Fecha:"  
    Calendario_inicio.Value = CType(indices(0).SubItems(1).Text, Date)  
    Calendario_fin.Enabled = False  
    Calendario_fin.Value = Calendario_inicio.Value  
    For i = 0 To 23  
        Tabla_Personaliza(i).Value = tabla_Personalizado(num, i)  
    Next  
End Sub
```

```
Private Sub Descargar2Controlador_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Descargar2Controlador.Click, DescargarAlControladorToolStripMenuItem.Click
    Descargar_a_Controlador()
End Sub
```

```
Private Sub Subir4Controlador_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Subir4Controlador.Click, SubirDesdeElControladorToolStripMenuItem.Click
    Dim respuesta(128) As Byte
    Dim descargar(2) As Byte
    Dim ent_aux As Campos
    Dim n_personalizados As Integer
    Dim n_feriados As Integer
    Dim person(365) As String
    Dim feria(365) As String
    Dim aux_s As String
    Dim item As ListViewItem

    Dim ok_m As Boolean
    Dim ok As Boolean = True

    If TabControl1.Visible = True Then
        If MsgBox("¿Desea Guardar los Cambios Realizados en el Archivo Actual?", MsgBoxStyle.YesNo, "Obtener Datos Desde el Controlador") = MsgBoxResult.Yes Then
            Guardar_Archivo()
            ok_m = True
        Else
            ok_m = False
        End If
    Else
        ok = True
    End If

    If ok Or ok_m Then

        ' Habilita controles
        Titulo_anio.Visible = True
        TabControl1.Visible = True
        TabControl1.SelectedIndex = 2
        Me.Refresh()

        ' Inicializacion de datos necesarios
        TabControl1.Enabled = False
        Titulo_anio.Enabled = False
        Lista_Feriados.Items.Clear()
        Lista_Personalizados.Items.Clear()
        d_ordinario.Checked = True
        Barra.Value = 0
        Barra.Visible = True
        Descargando.Visible = True

        ' Recibe Numero de Dias Personalizados [funcion 19]
        Descargando.Text = "Recibiendo: Número de días personalizados"
        Barra.Maximum = 1
        Me.Refresh()
        respuesta = TxSerial(19, descargar)
        If respuesta(0) = 19 Then
            ent_aux.dato1 = respuesta(1)
        End If
    End If
End Sub
```

```
ent_aux.dato2 = respuesta(2)
n_personalizados = ent_aux.datoS
Barra.Value = 1
Else
  MsgBox("Error al recibir Número de Días Personalizados", MsgBoxStyle.Information,
"Recepción de Datos")
  Descargando.Text = "Recepción Errónea: Número de Días Personalizados"
  Me.Refresh()
End If
Barra.Value = 0

' Recibe datos de Dias Personalizados (dia, mes, temporizaciones)[funcion 20]
Descargando.Text = "Recibiendo: Fechas y Temporizaciones de los Días Personalizados"
If n_personalizados > 1 Then
  Barra.Maximum = n_personalizados - 1
Else
  Barra.Maximum = 1
End If
Me.Refresh()
For i = 0 To n_personalizados - 1
  respuesta = TxSerial(20, descargar)
  If respuesta(0) = 20 Then
    item = New ListViewItem(Lista_Personalizados.Items.Count() + 1.ToString)
    item.SubItems.Add(respuesta(1).ToString + "/" + respuesta(2).ToString + "/" +
Now.Year.ToString)
    aux_s = ""
    For k = 0 To 23
      tabla_Personalizado(i, k) = respuesta(k + 3)
      aux_s += tabla_Personalizado(i, k).ToString + "."
    Next
    item.SubItems.Add(aux_s)
    ' Agrega el item a la Lista
    Lista_Personalizados.Items.Add(item)
    Barra.Value = i
  Else
    MsgBox("Error al recibir Fechas y Temporizaciones de los Días Personalizados",
MsgBoxStyle.Information, "Recepción de Datos")
    Descargando.Text = "Recepción Errónea: Fechas Temporizaciones de los Días
Personalizados"
    Me.Refresh()
  End If
Next
Barra.Value = 0

' Recibe Numero de Dias Feriados [funcion 21]
Descargando.Text = "Recibiendo: Número de Días Feriados"
Barra.Maximum = 1
Me.Refresh()
respuesta = TxSerial(21, descargar)
If respuesta(0) = 21 Then
  n_feriados = respuesta(1)
  Barra.Value = 1
Else
  MsgBox("Error al recibir Número de Días Feriados", MsgBoxStyle.Information,
"Recepción de Datos")
  Descargando.Text = "Recepción Errónea: Número de Días Feriados"
  Me.Refresh()
End If
Barra.Value = 0
```

```
' Recibe datos de Dias Feriados (dia, mes)[funcion 22]
Descargando.Text = "Recibiendo: Fechas de los Días Feriados"
Barra.Maximum = n_feriados - 1
Me.Refresh()
For i = 0 To n_feriados - 1
    respuesta = TxSerial(22, descargar)
    If respuesta(0) = 22 Then
        item = New ListViewItem(Lista_Feriados.Items.Count() + 1.ToString)
        item.SubItems.Add(respuesta(1).ToString + "/" + respuesta(2).ToString + "/" +
Now.Year.ToString)
        Lista_Feriados.Items.Add(item)
        Barra.Value = i
    Else
        MsgBox("Error al recibir Fechas de los Días Feriados", MsgBoxStyle.Information,
"Recepción de Datos")
        Descargando.Text = "Recepción Errónea: Fechas de los Días Feriados"
        Me.Refresh()
    End If
Next
Barra.Value = 0

' Recibe Temporizacion de Días Feriados [funcion 23]
Descargando.Text = "Recibiendo: Temporizaciones de los Días Feriados"
Barra.Maximum = 23
Me.Refresh()
respuesta = TxSerial(23, descargar)
If respuesta(0) = 23 Then
    For i = 0 To 23
        tabla_Feriado(i) = respuesta(i + 1)
        Barra.Value = i
    Next
Else
    MsgBox("Error al recibir Temporización de los Días Feriados", MsgBoxStyle.Information,
"Recepción de Datos")
    Descargando.Text = "Recepción Errónea: Temporizaciones de los Días Feriados"
    Me.Refresh()
End If
Barra.Value = 0

' Recibe Temporizacion de Fines de Semana [funcion 24]
Descargando.Text = "Recibiendo: Temporizaciones de los Fines de Semana"
Barra.Maximum = 23
Me.Refresh()
respuesta = TxSerial(24, descargar)
If respuesta(0) = 24 Then
    For i = 0 To 23
        tabla_FinSemana(i) = respuesta(i + 1)
        Barra.Value = i
    Next
Else
    MsgBox("Error al recibir Temporización de los Fines de Semana",
MsgBoxStyle.Information, "Recepción de Datos")
    Descargando.Text = "Recepción Errónea: Temporizaciones de los Fines de Semana"
    Me.Refresh()
End If
Barra.Value = 0

' Recibe Temporizacion de Días Ordinarios [funcion 25]
Descargando.Text = "Recibiendo: Temporizaciones de los Días Ordinarios"
Barra.Maximum = 23
```

```
Me.Refresh()
respuesta = TxSerial(25, descargar)
If respuesta(0) = 25 Then
    For i = 0 To 23
        tabla_Ordinario(i) = respuesta(i + 1)
        Tabla_Tipo(i).Value = tabla_Ordinario(i)
        Barra.Value = i
    Next
Else
    MsgBox("Error al recibir Temporización de los Días Ordinarios",
MsgBoxStyle.Information, "Recepción de Datos")
    Descargando.Text = "Recepción Errónea: Temporizaciones de los Días Ordinarios"
    Me.Refresh()
End If

Descargando.Text = "Recepción Finalizada "
MsgBox("Recepción Finalizada...", MsgBoxStyle.Information, "Recepción de Datos")
TabControl1.Enabled = True
Titulo_ano.Enabled = True
Barra.Value = 0
Barra.Visible = False
Descargando.Visible = False
Me.Refresh()

If Lista_Feridos.Items.Count() > 0 Then
    cmd_eliminar.Enabled = True
    cmd_limpiar.Enabled = True
End If

If Lista_Personalizados.Items.Count() > 0 Then
    cmd_eliminar_personalizado.Enabled = True
    cmd_limpiar_personalizado.Enabled = True
End If

End If
End Sub

Private Sub cmd_limpiar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles cmd_limpiar.Click
    If MsgBox("¿Está seguro de eliminar Todos los Días Feridos?", MsgBoxStyle.Exclamation +
MsgBoxStyle.OkCancel, "Eliminar Días Feridos") = MsgBoxResult.Ok Then
        Lista_Feridos.Items.Clear()
        cmd_limpiar.Enabled = False
        cmd_eliminar.Enabled = False
    End If
End Sub

Private Sub GuardarArchivo_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles GuardarArchivo.Click, GuardarArchivoToolStripMenuItem.Click
    Guardar_Archivo()
End Sub

Private Sub GuardarYDescargar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles GuardarYDescargar.Click,
GuardarArchivoYDescargarAIControladorToolStripMenuItem.Click
    Guardar_Archivo()
    Descargar_a_Controlador()
```

End Sub

```
Private Sub SalirToolStripMenuItem_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SalirToolStripMenuItem.Click
    Dim resp As Integer
    If Cambios Then
        resp = MsgBox("¿Desea Guardar los Cambios Realizados en el Archivo Actual?",
        MsgBoxStyle.YesNoCancel, "Salir")
        ' si=6, no=7, cancel=2

        If resp = 2 Then ' Cancelar
            Exit Sub
        End If

        If resp = 6 Then ' Si
            Guardar_Archivo()
        End If
    End If

    ' Confirma que el puerto serial esté cerrado
    If SerialPort1.IsOpen = True Then
        SerialPort1.Close()
    End If

    ' Cierra todas las ventanas de la aplicación
    Inicio_de_Sesion.Close()
    Conexion.Close()
    Modif_anio.Close()
    Me.Close()
End Sub
```

```
Private Sub AcercaDeToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AcercaDeToolStripMenuItem.Click
    About.Show()
End Sub
```

```
Private Sub CambiarContraseñaToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CambiarContraseñaToolStripMenuItem.Click
    Cambiar_Password.Show()
End Sub
```

```
Private Sub DescargarContraseñaParaElControladorToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DescargarContraseñaParaElControladorToolStripMenuItem.Click
    Password_Controlador.Show()
End Sub
```

```
Private Sub AyudaToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AyudaToolStripMenuItem1.Click
    MsgBox("Refiérase al <Manual del Usuario (Sección 7)> entregado con documentación," + Chr(13) + "o en el CD en la ruta: D:\ESPE-UOST\Documentación\Manual del Usuario.pdf",
    MsgBoxStyle.OkOnly, "Ayuda")
End Sub
```

End Class


```
End If

If respuesta(1) = 0 Then
    Return (1)
End If

End Function

' Funcion que carga los dias feriados por defecto
Public Sub Cargar_Feriado(ByVal fecha As String)
    Dim item As ListViewItem

    item = New ListViewItem(Principal.Lista_Feriados.Items.Count() + 1.ToString)
    item.SubItems.Add(fecha)
    Principal.Lista_Feriados.Items.Add(item)
End Sub

' Función que actualiza el arreglo de dias personalizados, respecto a la Lista_Personalizados
Public Sub Actualizar_Arreglo_Personalizados()
    Dim i, j, ind As Integer
    Dim arr_c() As Char
    Dim arr_s(23) As String

    For i = 0 To Principal.Lista_Personalizados.Items.Count() - 1
        arr_c = Principal.Lista_Personalizados.Items(i).SubItems(2).Text.ToCharArray
        ind = 0
        For j = 0 To 23
            arr_s(j) = ""
        Next

        For j = 0 To arr_c.Length - 1
            If arr_c(j) <> "." Then
                arr_s(ind) += arr_c(j)
            Else
                ind += 1
            End If
        Next

        For j = 0 To 23
            Principal.tabla_Personalizado(i, j) = arr_s(j)
        Next
    Next
End Sub

' Funcion que Descarga los Datos al Controlador
Public Sub Descargar_a_Controlador()
    Dim i, j As Integer
    Dim fecha_sel As Date
    Dim descargar(128) As Byte
    Dim respuesta(128) As Byte
    Dim empezar_desc_p As Boolean = False
    Dim empezar_desc_f As Boolean = False
    Dim ent_aux As Campos

    Principal.TabControl1.Enabled = False
    Principal.Titulo_anio.Enabled = False
```

```
' Actualiza ultimo dia configurado
If Principal.d_ordinario.Checked Then
  For i = 0 To 23
    Principal.tabla_Ordinario(i) = Principal.Tabla_Tipo(i).Value
  Next
End If

If Principal.f_semana.Checked Then
  For i = 0 To 23
    Principal.tabla_FinSemana(i) = Principal.Tabla_Tipo(i).Value
  Next
End If

If Principal.d_feriado.Checked Then
  For i = 0 To 23
    Principal.tabla_Feriado(i) = Principal.Tabla_Tipo(i).Value
  Next
End If

' Se envía al controlador el Numero de Dias Personalizados [funcion 12]
Principal.Barra.Visible = True
Principal.Descargando.Visible = True
Principal.Descargando.Text = "Descargando: Número de Días Personalizados"
Principal.Refresh()
Principal.Barra.Value = 0
Principal.Barra.Maximum = 1

ent_aux.datoS = Principal.Lista_Personalizados.Items.Count()
descargar(0) = ent_aux.dato1
descargar(1) = ent_aux.dato2
respuesta = TxSerial(12, descargar)

If respuesta(0) = 12 And respuesta(1) = 0 Then
  empezar_desc_p = True
  Principal.Barra.Value = 1
Else
  MsgBox("Error al enviar Numero de Días Personalizados ", MsgBoxStyle.Information,
"Descarga de Datos")
  empezar_desc_p = False
End If
Principal.Barra.Value = 0

' Se envía al controlador Fecha (dia y mes) y Temporizacion de cada Dia Personalizado
[funcion 13]
If empezar_desc_p Then
  Principal.Descargando.Text = "Descargando: Días Personalizados"
  Principal.Refresh()
  If Principal.Lista_Personalizados.Items.Count() > 1 Then
    Principal.Barra.Maximum = Principal.Lista_Personalizados.Items.Count() - 1
  Else
    Principal.Barra.Maximum = 1
  End If

  For i = 0 To Principal.Lista_Personalizados.Items.Count() - 1
    Principal.Barra.Value = i
    fecha_sel = Principal.Lista_Personalizados.Items(i).SubItems(1).Text
    descargar(0) = fecha_sel.Day
    descargar(1) = fecha_sel.Month
    For j = 0 To 23
      descargar(j + 2) = Principal.tabla_Personalizado(i, j)
```

```
        Next
        respuesta = TxSerial(13, descargar)
        If respuesta(0) <> 13 Or respuesta(1) <> 0 Then
            MsgBox("Error al enviar los Días Personalizados ", MsgBoxStyle.Information,
"Descarga de Datos")
            Exit For
        End If
    Next
    Principal.Barra.Value = 0
    Principal.Descargando.Text = "Descarga Completa: Días Personalizados"
    Principal.Refresh()
    empezar_desc_p = False
End If

' Se envía al controlador el Numero de Dias Feriados [funcion 14]
Principal.Descargando.Text = "Descargando: Número de Días Feriados"
Principal.Refresh()
Principal.Barra.Maximum = 1

descargar(0) = Principal.Lista_Feriados.Items.Count()
respuesta = TxSerial(14, descargar)
If respuesta(0) = 14 And respuesta(1) = 0 Then
    empezar_desc_f = True
    Principal.Barra.Value = 1
Else
    MsgBox("Error al enviar Numero de Días Feriados ", MsgBoxStyle.Information, "Descarga
de Datos")
    empezar_desc_f = False
End If
Principal.Barra.Value = 0

' Se envía al controlador la Lista de las Fechas (día y mes) de cada Dia Feriado [funcion 15]
If empezar_desc_f Then
    Principal.Descargando.Text = "Descargando: Fechas de Días Feriados"
    Principal.Refresh()
    Principal.Barra.Maximum = Principal.Lista_Feriados.Items.Count() - 1

    For i = 0 To Principal.Lista_Feriados.Items.Count() - 1
        Principal.Barra.Value = i
        fecha_sel = Principal.Lista_Feriados.Items(i).SubItems(1).Text
        descargar(0) = fecha_sel.Day
        descargar(1) = fecha_sel.Month
        respuesta = TxSerial(15, descargar)
        If respuesta(0) <> 15 Or respuesta(1) <> 0 Then
            MsgBox("Error al enviar Días Feriados", MsgBoxStyle.Information, "Descarga de
Datos")
            Exit For
        End If
    Next
    Principal.Barra.Value = 0
    Principal.Descargando.Text = "Descarga Completa: Fechas de Días Feriados"
    Principal.Refresh()
    empezar_desc_f = False
End If

' Se envía al controlador la temporización de los Días Feriados [funcion 16]
Principal.Descargando.Text = "Descargando: Temporización de Días Feriados"
Principal.Refresh()
Principal.Barra.Maximum = 23
For i = 0 To 23
```

```
        descargar(i) = Principal.tabla_Feriado(i)
        Principal.Barra.Value = i
    Next
    respuesta = TxSerial(16, descargar)
    If respuesta(0) = 16 And respuesta(1) = 0 Then
        Principal.Descargando.Text = "Descarga Completa: Temporización de Días Feriados"
        Principal.Refresh()
    Else
        MsgBox("Error al enviar Temporización de Días Feriados ", MsgBoxStyle.Information,
"Descarga de Datos")
        Principal.Descargando.Text = "Descarga Errónea: Temporización de Días Feriados"
        Principal.Refresh()
    End If
    Principal.Barra.Value = 0

' Se envía al controlador la temporización de los Fines de Semana [funcion 17]
Principal.Descargando.Text = "Descargando: Temporización Fines de Semana"
Principal.Refresh()
Principal.Barra.Maximum = 23
For i = 0 To 23
    descargar(i) = Principal.tabla_FinSemana(i)
    Principal.Barra.Value = i
Next
respuesta = TxSerial(17, descargar)
If respuesta(0) = 17 And respuesta(1) = 0 Then
    Principal.Descargando.Text = "Descarga Completa: Temporización de Fines de Semana"
    Principal.Refresh()
Else
    MsgBox("Error al enviar Temporización de Fines de Semana", MsgBoxStyle.Information,
"Descarga de Datos")
    Principal.Descargando.Text = "Descarga Errónea: Temporización de Fines de Semana"
    Principal.Refresh()
End If
Principal.Barra.Value = 0

' Se envía al controlador la temporización de los Días Ordinarios [funcion 18]
Principal.Descargando.Text = "Descargando: Temporización Días Ordinarios"
Principal.Refresh()
Principal.Barra.Maximum = 23
For i = 0 To 23
    descargar(i) = Principal.tabla_Ordinario(i)
    Principal.Barra.Value = i
Next
respuesta = TxSerial(18, descargar)
If respuesta(0) = 18 And respuesta(1) = 0 Then
    Principal.Descargando.Text = "Descarga Completa: Temporización de Días Ordinarios"
    Principal.Refresh()
Else
    MsgBox("Error al enviar Temporización de Días Ordinarios", MsgBoxStyle.Information,
"Descarga de Datos")
    Principal.Descargando.Text = "Descarga Errónea: Temporización de Días Ordinarios"
    Principal.Refresh()
End If
Principal.Descargando.Text = "Descarga Completa "
MsgBox("Descarga Completa...", MsgBoxStyle.Information, "Descarga de Datos")
Principal.TabControl1.Enabled = True
Principal.Titulo_anio.Enabled = True
Principal.Barra.Value = 0
Principal.Barra.Visible = False
Principal.Descargando.Visible = False
```

```
Principal.Refresh()
End Sub

' Funcion que Guarda en un Archivo todos los Datos
Public Sub Guardar_Archivo()
    Dim i, j As Integer
    Dim ind As Integer
    Dim n_feriados, n_personalizados As Integer
    Dim arreglo_save(1024) As Byte
    Dim fecha As Date
    Dim aux_year As Campos

    ' Actualiza ultimo dia configurado
    If Principal.d_ordinario.Checked Then
        For i = 0 To 23
            Principal.tabla_Ordinario(i) = Principal.Tabla_Tipo(i).Value
        Next
    End If

    If Principal.f_semana.Checked Then
        For i = 0 To 23
            Principal.tabla_FinSemana(i) = Principal.Tabla_Tipo(i).Value
        Next
    End If

    If Principal.d_feriado.Checked Then
        For i = 0 To 23
            Principal.tabla_Feriado(i) = Principal.Tabla_Tipo(i).Value
        Next
    End If

    ' Guarda Temporizaciones de los Dias Ordinarios
    ind = 0
    For i = 0 To 23
        arreglo_save(ind) = Principal.tabla_Ordinario(i) ' 0 a 23
        ind += 1
    Next

    ' Guarda Temporizaciones de los Fines de Semana
    For i = 0 To 23
        arreglo_save(ind) = Principal.tabla_FinSemana(i) ' 24 a 47
        ind += 1
    Next

    ' Guarda Temporizaciones de los Dias Feriados
    For i = 0 To 23
        arreglo_save(ind) = Principal.tabla_Feriado(i) ' 48 a 71
        ind += 1
    Next

    ' Guarda Numero de Días Feriados
    n_feriados = Principal.Lista_Feriados.Items.Count()
    arreglo_save(ind) = n_feriados ' 72
    ind += 1

    ' Guarda las Fechas de los Días Feriados
    For i = 0 To n_feriados - 1
        fecha = Principal.Lista_Feriados.Items(i).SubItems(1).Text
        arreglo_save(ind) = fecha.Day
```

```
    ind += 1
    arreglo_save(ind) = fecha.Month
    ind += 1
    aux_year.datoL = fecha.Year
    arreglo_save(ind) = aux_year.dato1
    ind += 1
    arreglo_save(ind) = aux_year.dato2
    ind += 1
    arreglo_save(ind) = aux_year.dato3
    ind += 1
Next

' Guarda Numero de Días Personalizados
n_personalizados = Principal.Lista_Personalizados.Items.Count()
arreglo_save(ind) = n_personalizados          '(n_feridos+1)*5 + 1
ind += 1

' Guarda Temporizaciones de los Días Personalizados
For i = 0 To n_personalizados - 1
    For j = 0 To 23
        arreglo_save(ind) = Principal.tabla_Personalizado(i, j)
        ind += 1
    Next
Next

' Guarda las Fechas de los Dias Personalizados
For i = 0 To n_personalizados - 1
    fecha = Principal.Lista_Personalizados.Items(i).SubItems(1).Text
    arreglo_save(ind) = fecha.Day
    ind += 1
    arreglo_save(ind) = fecha.Month
    ind += 1
    aux_year.datoL = fecha.Year
    arreglo_save(ind) = aux_year.dato1
    ind += 1
    arreglo_save(ind) = aux_year.dato2
    ind += 1
    arreglo_save(ind) = aux_year.dato3
    ind += 1
Next

' Guarda TODO el Arreglo creado para Guardar
Principal.Save1.Filter = "Archivos Horario TROLE (*.TRO)*.TRO"
Principal.Save1.FilterIndex = 0
If Principal.Save1.ShowDialog() = Windows.Forms.DialogResult.OK Then
    If My.Computer.FileSystem.FileExists(Principal.Save1.FileName) Then
        My.Computer.FileSystem.DeleteFile(Principal.Save1.FileName)
    End If
    My.Computer.FileSystem.WriteAllBytes _
    (Principal.Save1.FileName, arreglo_save, False) 'true agrega, false sobrescribe
End If
End Sub

' Clase y Funcion para ordenar las fechas de los dias Feriados
Class ListViewItemComparer
    Implements IComparer
    Private col As Integer

    Public Sub New()
```

```
        col = 0
    End Sub

    Public Sub New(ByVal column As Integer)
        col = column
    End Sub

    Public Function Compare(ByVal x As Object, ByVal y As Object) As Integer Implements
System.Collections.IComparer.Compare
        Dim ini, fin As Date
        Dim ini_s, fin_s As String

        If col = 1 Then
            ini_s = CType(x, ListViewItem).SubItems(col).Text
            fin_s = CType(y, ListViewItem).SubItems(col).Text
            ini = CType(ini_s, Date)
            fin = CType(fin_s, Date)

            If ini.Date > fin.Date Then
                Return (1)
            End If

            If ini.Date < fin.Date Then
                Return (-1)
            End If

            If ini.Date = fin.Date Then
                Return (0)
            End If
        End If
    End Function
End Class

End Module
```

ANEXO 4

**CÓDIGO DESARROLLADO PARA EL
CONTROLADOR PRINCIPAL, EN DYNAMIC C**

```

#class static //Change default: local vars now stored on stack.
#memmap xmem //Required to reduce root memory usage

#define DOUTBUFSIZE 255 //Tamaño del Buffer para RS 485
#define DINBUFSIZE 31
#define BINBUFSIZE 255 //Tamaño del Buffer para RS 232
#define BOUTBUFSIZE 255
#define BAUD232 9600 //Serial baud rate
#define N_DIAS 366
#define MAX_FERIADOS 255
#define MAX_PERSONALIZADOS 366

// Salidas
#define LED_ROJO 0
#define LUZ 7
#define MSG_1 1
#define MSG_2 2
#define PLAY 5
#define REC 6

//Entradas
#define SENSOR 8

//Horarios
#define ORDINARIO 0
#define FERIADO 1
#define FIN_SEMANA 2
#define PERSONALIZADO 3
#define TPANTALLA 10 //tiempo para activar protector de pantalla
#define TBACKLIGHT 5 //tiempo para apagar el backlight

//-----
// Macro's for MENU system.... to be set for your application
//-----
#define NUM_MENUS 11 // Used to set the size of a structure
// array to hold menu information, set
// to actual number of menus that you
// are going to be using.

// Define the unique menu number...is used for an index into the menu
// structure array.
#define LVL_MAINMENU 0
#define LVL_BACKLIGHT 1
#define LVL_MOD_DIA 2
#define LVL_SELEC_HOR 3
#define LVL_HOR_DIA 4
#define LVL_SELEC_DIA31 5
#define LVL_SELEC_DIA30 6
#define LVL_SELEC_DIA28 7
#define LVL_SELEC_MES 8
#define LVL_MSG_VOZ 9
#define LVL_OP_VOZ 10

//-----
// Menu options.....set as needed for your application
//-----
// Can insert/delete menu options. The highlight bar is setup
// to start with the first MENU option and stop at the last
// menu option in the MENU.
//

```

// When adding/deleting menu options you must match up the
// case statements to the menu option number.

```
const char *main_menu [] =
{
    " -MENU PRINCIPAL-",
    "> Modificar Horario",
    "> Mod. Hora Actual",
    "> Luz de Fondo",
    "> Mensajes de Voz",
    "> Actualizar Hora",
    "> Acerca de...",
    ""
};

const char *backlight [] =
{
    " -LUZ DE FONDO-",
    "> Apagar",
    "> Encender",
    "> Automatico",
    "< Volver",
    ""
};

const char *mod_dia [] =
{
    " -PERSONALIZAR DIA-",
    "> Actual",
    "> Seleccionar Dia",
    "< Volver",
    ""
};

const char *selec_hor [] =
{
    " -SELEC. HORARIO-",
    "> Dia Ordinario",
    "> Dia Feriado",
    "> Fin de Semana",
    "> Personalizado",
    "< Volver",
    ""
};

const char *msg_voz [] =
{
    " -MENSAJE DE VOZ-",
    "> Mensaje de Apertura",
    "> Mensaje de Cierre",
    "< Volver",
    ""
};

const char *op_voz [] =
{
    " -OPCIONES DE VOZ-",
    "> Grabar Mensaje",
    "> Reproducir Mensaje",
    "> Habilitar Mensaje",
    "> Deshabilitar Mensaje",
    "> Tiempo de Activacion",
    "< Volver",
    ""
};
```

```
const char *hor_dia [] =
{
    "-SELECCIONAR HORA-",
    "> 0 a 1", "> 1 a 2", "> 2 a 3",
    "> 3 a 4", "> 4 a 5", "> 5 a 6",
    "> 6 a 7", "> 7 a 8", "> 8 a 9", "> 9 a 10",
    "> 10 a 11", "> 11 a 12", "> 12 a 13",
    "> 13 a 14", "> 14 a 15", "> 15 a 16",
    "> 16 a 17", "> 17 a 18", "> 18 a 19",
    "> 19 a 20", "> 20 a 21", "> 21 a 22",
    "> 22 a 23", "> 23 a 0",
    "> Todas",
    "< Volver",
    ""
};

const char *selec_mes [] =
{
    "-SELECCIONAR MES-",
    "> Enero", "> Febrero", "> Marzo",
    "> Abril", "> Mayo", "> Junio",
    "> Julio", "> Agosto", "> Septiembre",
    "> Octubre", "> Noviembre", "> Diciembre",
    "< Volver",
    ""
};

const char *selec_dia31 [] =
{
    "-SELECCIONAR DIA-",
    "> Dia 1", "> Dia 2", "> Dia 3", "> Dia 4", "> Dia 5",
    "> Dia 6", "> Dia 7", "> Dia 8", "> Dia 9", "> Dia 10",
    "> Dia 11", "> Dia 12", "> Dia 13", "> Dia 14", "> Dia 15",
    "> Dia 16", "> Dia 17", "> Dia 18", "> Dia 19", "> Dia 20",
    "> Dia 21", "> Dia 22", "> Dia 23", "> Dia 24", "> Dia 25",
    "> Dia 26", "> Dia 27", "> Dia 28", "> Dia 29", "> Dia 30",
    "> Dia 31",
    "< Volver",
    ""
};

const char *selec_dia30 [] =
{
    "-SELECCIONAR DIA-",
    "> Dia 1", "> Dia 2", "> Dia 3", "> Dia 4", "> Dia 5",
    "> Dia 6", "> Dia 7", "> Dia 8", "> Dia 9", "> Dia 10",
    "> Dia 11", "> Dia 12", "> Dia 13", "> Dia 14", "> Dia 15",
    "> Dia 16", "> Dia 17", "> Dia 18", "> Dia 19", "> Dia 20",
    "> Dia 21", "> Dia 22", "> Dia 23", "> Dia 24", "> Dia 25",
    "> Dia 26", "> Dia 27", "> Dia 28", "> Dia 29", "> Dia 30",
    "< Volver",
    ""
};

const char *selec_dia28 [] =
{
    "-SELECCIONAR DIA-",
    "> Dia 1", "> Dia 2", "> Dia 3", "> Dia 4", "> Dia 5",
    "> Dia 6", "> Dia 7", "> Dia 8", "> Dia 9", "> Dia 10",
    "> Dia 11", "> Dia 12", "> Dia 13", "> Dia 14", "> Dia 15",
    "> Dia 16", "> Dia 17", "> Dia 18", "> Dia 19", "> Dia 20",
    "> Dia 21", "> Dia 22", "> Dia 23", "> Dia 24", "> Dia 25",
    "> Dia 26", "> Dia 27", "> Dia 28",
    "< Volver",
    ""
};
```

```

};

// Do auto calculation of the number of menu options for a given MENU.
#define NUM_MAINMENU_OPTS sizeof(main_menu)/sizeof(int)
#define NUM_BACKLIGHT_OPTS sizeof(backlight)/sizeof(int)
#define NUM_MOD_DIA_OPTS sizeof(mod_dia)/sizeof(int)
#define NUM_SELEC_HOR_OPTS sizeof(selec_hor)/sizeof(int)
#define NUM_HOR_DIA_OPTS sizeof(hor_dia)/sizeof(int)
#define NUM_SELEC_MES_OPTS sizeof(selec_mes)/sizeof(int)
#define NUM_SELEC_DIA31_OPTS sizeof(selec_dia31)/sizeof(int)
#define NUM_SELEC_DIA30_OPTS sizeof(selec_dia30)/sizeof(int)
#define NUM_SELEC_DIA28_OPTS sizeof(selec_dia28)/sizeof(int)
#define NUM_MSG_VOZ_OPTS sizeof(msg_voz)/sizeof(int)
#define NUM_OP_VOZ_OPTS sizeof(op_voz)/sizeof(int)

//-----
// START.....Macros and Structure for menu system
// !!! The following section is not intended to be changed.
//-----
#define MENU_INIT          0
#define MENU_NO_CHANGE    1
#define MENU_REFRESH      2
#define SALIR              1000

struct menu_infor
{
    int current_offset;
    int new_offset;
    int lasthighlight;
    int highlight;
};

struct menu_infor menu[NUM_MENUS];
//-----
// END.....Macros and Structure for menu system
//-----

// Bitmap : Img_OP_2_bmp
// Buffer Size : 256
// Monochrome : Black Foreground, White Background
// Mode : Landscape
// Height : 32 pixels.
// Width : 60 pixels.
// Init : glXPutBitmap (leftedge,topedge,60,32,Img_OP_2_bmp);
xdata Img_OP_2_bmp {
'\x00', '\x00', '\x00', '\x1F', '\xFF', '\xFF', '\x80', '\x0F',
'\x00', '\x00', '\x00', '\x3F', '\xFF', '\xFF', '\xC0', '\x0F',
'\x00', '\x00', '\x00', '\x7F', '\xFF', '\xFF', '\xE0', '\x0F',
'\x01', '\x10', '\x00', '\xFF', '\xFF', '\xFF', '\xF0', '\x0F',
'\x01', '\x10', '\x00', '\xFF', '\xFF', '\xFF', '\xF0', '\x0F',
'\x01', '\x10', '\x01', '\xFF', '\xFD', '\xFF', '\xF8', '\x0F',
'\x01', '\x10', '\x03', '\xFF', '\xF8', '\xFF', '\xFC', '\x0F',
'\x00', '\xE0', '\x03', '\xFF', '\xF0', '\xFF', '\xFC', '\x0F',
'\x00', '\x00', '\x03', '\xFF', '\xE1', '\xC1', '\xFC', '\x0F',
'\x00', '\x00', '\x07', '\xFF', '\xC3', '\x80', '\xFE', '\x0F',
'\x00', '\xE0', '\x07', '\xFF', '\x87', '\x0F', '\xFE', '\x0F',
'\x01', '\x10', '\x0F', '\xF8', '\x0E', '\x1F', '\xFF', '\x0F',
'\x01', '\x10', '\x0F', '\xFC', '\x1C', '\x3F', '\xFF', '\x0F',
'\x01', '\x10', '\x0F', '\xFF', '\xF8', '\x7F', '\xFF', '\x0F',
'\x00', '\xE0', '\x0F', '\xFF', '\xF8', '\xFF', '\xFF', '\x0F',

```

```

'\x00', '\x00', '\x0F', '\xFF', '\xFD', '\xFF', '\xFF', '\x0F',
'\x00', '\x00', '\x0F', '\xFF', '\xFF', '\xFF', '\xFF', '\x0F',
'\x00', '\xF0', '\x0F', '\xFF', '\xFF', '\xFF', '\xFF', '\x0F',
'\x01', '\x00', '\x0F', '\xE0', '\xFF', '\xEF', '\xFF', '\x0F',
'\x00', '\xE0', '\x07', '\xFB', '\xFF', '\xEF', '\xFE', '\x0F',
'\x00', '\x10', '\x07', '\xFB', '\xAC', '\x6C', '\x7E', '\x0F',
'\x01', '\xE0', '\x03', '\xFB', '\x9B', '\xAB', '\xBC', '\x0F',
'\x00', '\x00', '\x03', '\xFB', '\xBB', '\xA8', '\x7C', '\x0F',
'\x00', '\x00', '\x01', '\xFB', '\xBB', '\xAB', '\xF8', '\x0F',
'\x01', '\xF0', '\x01', '\xFB', '\xBC', '\x6C', '\x78', '\x0F',
'\x00', '\x40', '\x00', '\xFF', '\xFF', '\xFF', '\xF0', '\x0F',
'\x00', '\x40', '\x00', '\x7F', '\xFF', '\xFF', '\xE0', '\x0F',
'\x00', '\x40', '\x00', '\x3F', '\xFF', '\xFF', '\xC0', '\x0F',
'\x00', '\x40', '\x00', '\x1F', '\xFF', '\xFF', '\x80', '\x0F',
'\x00', '\x00', '\x00', '\x0F', '\xFF', '\xFF', '\x00', '\x0F',
'\x00', '\x00', '\x00', '\x03', '\xFF', '\xFC', '\x00', '\x0F',
'\x00', '\x00', '\x00', '\x01', '\xFF', '\xF8', '\x00', '\x0F'
};

// Bitmap : au_bmp
// Buffer Size : 238
// Monochrome : Black Foreground, White Background
// Mode : Landscape
// Height : 34 pixels.
// Width : 55 pixels.
// Init : glXPutBitmap (leftedge,topedge,55,34,au_bmp);
xdata au_bmp {
'\x3F', '\xFF', '\xFF', '\xFF', '\xF8', '\x00', '\x01',
'\x60', '\x00', '\x00', '\x00', '\x1F', '\xFF', '\xFD',
'\x4F', '\xFF', '\xF8', '\x00', '\x00', '\x00', '\x05',
'\xDF', '\xFF', '\xF1', '\xFF', '\xFF', '\xFA', '\x55',
'\x98', '\x92', '\x30', '\xFF', '\xFF', '\xFA', '\x55',
'\x9A', '\xDA', '\xB6', '\x67', '\xFF', '\xF2', '\xD5',
'\x9A', '\xDA', '\xB3', '\x03', '\xFF', '\xE6', '\xD5',
'\x98', '\xDA', '\x31', '\xE7', '\x00', '\x0E', '\xD5',
'\x9F', '\xFF', '\xF4', '\xFE', '\x7F', '\xF8', '\xD5',
'\x99', '\x18', '\x92', '\x3C', '\xE0', '\x01', '\xD5',
'\x9D', '\x5A', '\xD1', '\x99', '\xCF', '\xFF', '\x15',
'\x9D', '\x5A', '\xD4', '\x9B', '\x18', '\x00', '\x35',
'\x9D', '\x18', '\xD6', '\xDB', '\x73', '\xFF', '\xF5',
'\x9F', '\xFF', '\xF6', '\xDB', '\x66', '\x7F', '\xF5',
'\x9F', '\xFF', '\xF6', '\xDB', '\x6C', '\x00', '\x05',
'\x98', '\x98', '\x96', '\xDB', '\x6E', '\x7F', '\xF5',
'\x9A', '\xDA', '\xD6', '\xDB', '\x6F', '\xFF', '\xF5',
'\x9A', '\xDA', '\xD6', '\xDB', '\x6E', '\x00', '\x05',
'\x98', '\xD8', '\xD6', '\xDB', '\x6E', '\x7F', '\xF5',
'\x9F', '\xFF', '\xF6', '\xDB', '\x6F', '\xFF', '\xF5',
'\x98', '\x92', '\x36', '\xDB', '\x62', '\x77', '\x75',
'\x9A', '\xDA', '\xB6', '\xDB', '\x78', '\x22', '\x35',
'\x9A', '\xDA', '\xB6', '\xDB', '\x1E', '\x62', '\x35',
'\x98', '\xDA', '\x36', '\xDB', '\x87', '\xF7', '\x75',
'\x9F', '\xFF', '\xF6', '\xD9', '\xF0', '\x77', '\x75',
'\x99', '\x89', '\x16', '\xDC', '\x1C', '\x37', '\x75',
'\x9D', '\xAD', '\x56', '\xDF', '\x8E', '\x77', '\x75',
'\x9D', '\xAD', '\x56', '\xDF', '\xCF', '\xF7', '\x75',
'\x9D', '\x8D', '\x16', '\xD8', '\x7F', '\xF7', '\x75',
'\x9F', '\xFF', '\xF6', '\xDB', '\x0F', '\xF7', '\x75',
'\xCF', '\xFF', '\xF2', '\xDB', '\xE1', '\xF7', '\x75',
'\x40', '\x00', '\x10', '\x00', '\x00', '\x00', '\x05',
'\x7F', '\xFF', '\xFF', '\xFF', '\xFF', '\xFF', '\xFD',

```

```

'\x00', '\x00', '\x00', '\x00', '\x00', '\x00', '\x01'
};

// Init : glXPutBitmap (leftedge, topedge, 32, 32, pass_bmp);
xdata pass_bmp {
'\x80', '\x00', '\x00', '\x01',
'\x3F', '\xFF', '\xFF', '\xFC',
'\x7F', '\xFF', '\xFF', '\xFE',
'\x7F', '\xFF', '\xFF', '\xFE',
'\x7F', '\xFF', '\xFF', '\xFE',
'\x7F', '\xFF', '\xC7', '\xFE',
'\x7F', '\xFF', '\x00', '\xFE',
'\x7F', '\xFE', '\x7E', '\x7E',
'\x7F', '\xFC', '\xFF', '\x3E',
'\x7F', '\xF9', '\xFF', '\xBE',
'\x7F', '\xFB', '\xF3', '\xBE',
'\x7F', '\xFB', '\xF3', '\x9E',
'\x7F', '\xFB', '\xFF', '\x9E',
'\x7F', '\xF9', '\xFF', '\xBE',
'\x7F', '\xF9', '\xFF', '\xBE',
'\x7F', '\xF1', '\xFF', '\x3E',
'\x7F', '\xE3', '\xFE', '\x7E',
'\x7F', '\xC7', '\x80', '\xFE',
'\x7F', '\x8F', '\x03', '\xFE',
'\x7F', '\x1F', '\x3F', '\xFE',
'\x7E', '\x3F', '\x3F', '\xFE',
'\x7C', '\x7E', '\x3F', '\xFE',
'\x7C', '\xF0', '\x7F', '\xFE',
'\x7C', '\xE0', '\xFF', '\xFE',
'\x7C', '\x0F', '\xFF', '\xFE',
'\x7C', '\x1F', '\xFF', '\xFE',
'\x7F', '\xFF', '\xFF', '\xFE',
'\x7F', '\xFF', '\xFF', '\xFE',
'\x7F', '\xFF', '\xFF', '\xFE',
'\x7F', '\xFF', '\xFF', '\xFE',
'\x3F', '\xFF', '\xFF', '\xFC',
'\x80', '\x00', '\x00', '\x01'
};

//-----
// Macro's
//-----
#define MAXDISPLAYROWS 4
#define ASCII 0
#define NUMBER 1

//-----
// Structures, arrays, variables
//-----
fontInfo fi6x8, fi8x10, fi12x16;
windowFrame textWindow;

typedef struct
{
    int data;
    char *ptr;
} fieldupdate;

struct tm CurTime;
char szTime[40];

```

```

char szString[20];
const char Days[] = {"DomLunMarMieJueVieSab"};
const char Months[] = {"EneFebMarAbrMayJunJulAgoSepOctNovDic"};

int beeperTick, timerTick ;
int max_menu_options;
int max_cmds_options;
unsigned long ulTime;
char *keybuffer;

typedef struct
{
    word _crc;
    char msg1En;
    char msg2En;
    char msg1Tiempo;
    char msg2Tiempo;
} MENSAJES;

typedef struct
{
    char mes;
    char dia;
    char horas[24];
} HORARIOS;          //Horarios personalizados

typedef struct
{
    char mes;
    char dia;
} L_FERIADOS;      //Lista de dias feriados

CoData temporizacion, Lec_sensor, Menus; //Tareas

protected shared HORARIOS H_Personalizados[MAX_PERSONALIZADOS]; //HORARIOS
PERZONALIZADOS
protected shared int tot_Personalizados;//Indice de los horario perzonalizados

protected shared L_FERIADOS L_Feritados[MAX_FERIADOS];//Lista de dias feriados
protected shared char H_Feritados[24]; //Horario de dias feriados
protected shared char tot_Feritados;//Indice de los horario perzonalizados
protected shared char H_Ordinario[24]; //Horario de dias ordinarios
protected shared char H_FinSemana[24]; //Horario de dias feriados
protected shared int dias_recibidos; //auxiliarpara la recepcion de datos
protected shared MENSAJES msgVar; // estructura para los mensajes
protected shared char password[4];
int t_pantalla;
int backlightAUTO;

//Prototipos de Funciones
void msDelay(unsigned int delay) ;
int ProcessKeyField(int mode, fieldupdate *field);
int GetKeypadOption(int *offset, int *highlight, int num_options );
int display_menu ( char **line, int *state, int level, int num_options );
void FormatDateTime ( void );
int dispDate( void );
void date_prompt(char *ptr, int *col, int *row);
cofunc void SetDateTime( void );
void SignOnMessage(void);
int backlight_menu( void );

```



```

digOut((int)MSG_2,1);
digOut((int)PLAY,0); //Modo de Reproduccion
digOut((int)REC,1); //Modo de Grabacion
disp_buf[0]=1;//id del display
disp_buf[1]=101;// segmento g
wr_RS485(disp_buf,2);

//-----
// Display Sign-on message and wait for keypress
//-----
SignOnMessage();

//-----
// Main program loop for the MENU system
//-----
t_pantalla=0;
backlightAUTO=TRUE;
protector=TRUE;
CoPause(&Menus);
loopinit();
loophead();

costate // Comunicacion con la PC _____
{
    wfd cln=cof_serBread( (char *)buf_in, (int)BINBUFSIZE, 5);
    if(cln!=0)
        wfd comPC(buf_in,cln);
    serBrdFlush();
} // FIN de Comunicacion con la PC _____

costate Lec_sensor always_on //Lectura del Sensor _____
{
    waitfor(!digIn((int)SENSOR));
    printf("\nLo sintio!! se activa temporizacion");
    CoBegin(&temporizacion);
    CoPause(&Lec_sensor);
} //Fin de Lectura del Sensor _____

costate temporizacion //Temporizacion de las salidas _____
{
    printf("\nEntro a temporizar");
    tm_rd(&rtc);//Lee hora actual
    digOut((int)LUZ,0);
    digOut((int)LED_ROJO,0);
    wfd temp_i=buscarHorario(rtc.tm_mon,rtc.tm_mday,rtc.tm_hour,rtc.tm_wday);
    printf("\nya busco");
    waitfor(DelaySec((long) msgVar.msg1Tiempo)); // espera a que pare el trole
    printf("\nya espero");
    //Activa mensaje de ingreso
    if( ( !digIn((int)SENSOR) ) && ( msgVar.msg1En==TRUE ) )
    {
        digOut((int)MSG_1,0);
        waitfor(DelayMs(200));
        digOut((int)MSG_1,1);
    }
    for(i=0;i<1;i++)//titila displays antes de iniciar cuenta
    {
        if(digIn((int)SENSOR)) break; // si el trole sale antes de tiempo
        disp_buf[0]=1;//id del display
        disp_buf[1]=temp_i;// segundos
    }
}

```

```

        wr_RS485(displ_buf,2);
        waitfor(DelayMs(500));
        displ_buf[0]=1;//id del display
        displ_buf[1]=100;// segundos
        wr_RS485(displ_buf,2);
        waitfor(DelayMs(500));
    }
    for(temp_i;temp_i>0;temp_i--)// Realiza cuenta regresiva
    {
        if(digIn((int)SENSOR))break; // si el trole sale antes de tiempo
        if((temp_i==msgVar.msg2Tiempo)&&(msgVar.msg2En==TRUE))//Activa
                                                    mensaje de Salida
    {
        digOut((int)MSG_2,0);
        waitfor(DelayMs(200));
        digOut((int)MSG_2,1);
    }
        displ_buf[0]=1;//id del display
        displ_buf[1]=temp_i;// segundos
        wr_RS485(displ_buf,2);
        printf("\temp_i=%d",temp_i);
        waitfor(DelayMs(500));
        if( temp_i<=msgVar.msg2Tiempo )
        {
            displ_buf[0]=1;//id del display
            displ_buf[1]=100;// segundos
            wr_RS485(displ_buf,2);
        }
        waitfor(DelayMs(500));
    }

    if( ( temp_i==msgVar.msg2Tiempo ) && ( msgVar.msg2En==(char)TRUE ) )//Activa
                                                    mensaje de Salida
    {
        digOut((int)MSG_2,0);
        waitfor(DelayMs(200));
        digOut((int)MSG_2,1);
    }
    digOut((int)LUZ,1);
    displ_buf[0]=1;//id del display
    displ_buf[1]=0;// segundos
    wr_RS485(displ_buf,2);
    waitfor(digIn((int)SENSOR));
    digOut((int)LED_ROJO,1);
    displ_buf[0]=1;//id del display
    displ_buf[1]=101;// segmento g
    wr_RS485(displ_buf,2);
    waitfor(DelaySec(3));
    CoResume(&Lec_sensor);
} //Temporizacion de las salidas_____

costate //Procesa teclado
{
    keyProcess ();
    waitfor(DelayMs(10));
}

costate Menus always_on//menus _____
{
    // Display the MAIN MENU

```



```

        wfd P_Pantalla();
        CoBegin(&Menus);
        state=MENU_REFRESH;
        t_pantalla=(int)TPANTALLA;
    }
} else
    if(t_pantalla==(int)TBACKLIGHT)t_pantalla=(int)TBACKLIGHT;
    t_pantalla--;
    waitfor(DelaySec(1));
}
} //main()
}

//FUNCIONES AUXILIARES
//FUNCION PARA BUSCAR UN HORARIO
cofunc char buscarHorario(char mes, char dia, char hora, char dia_semana)
{
    //devuelve la temporizacion
    int i;

    //Personalizado
    for(i=0;i<tot_Personalizados;i++)
    {
        if(mes==H_Personalizados[i].mes && dia==H_Personalizados[i].dia)
        {
            printf("\nPer Mes=%d, Dia=%d",H_Personalizados[i].mes,
                H_Personalizados[i].dia);
            printf("\nH_Per=%d",H_Personalizados[i].horas[hora]);
            return(H_Personalizados[i].horas[hora]);
        }
        yield;
    }

    //Feriados
    for(i=0;i<tot_Feriados;i++)
    {
        if(mes==L_Feriados[i].mes && dia==L_Feriados[i].dia)
        {
            printf("\nFer Mes=%d, Dia=%d",L_Feriados[i].mes , L_Feriados[i].dia);
            printf("\nH_Fer=%d",H_Feriados[hora]);
            return(H_Feriados[hora]);
        }
        yield;
    }

    //Dias Ordinarios y Fines de Semana
    if(dia_semana==6 || dia_semana==0)
    {
        printf("\nH_Fini=%d",H_FinSemana[hora]);
        return(H_FinSemana[hora]);
    }
    else
    {
        printf("\nH_ordi=%d",H_Ordinario[hora]);
        return(H_Ordinario[hora]);
    }
    printf("\nNo encontro Dia");
    return -1;
}
}

```

```

//FUNCION PARA ESCRIBIR EL PUERTO RS232*****
void wr_serial(char *buf, int tam)
{
    serBwrite((char *)buf,tam); // envia respuesta
    while (serBwrFree() != BOUTBUFSIZE);
    // then, wait until the Tx data register and the Tx shift register
    // are both empty
    while (BitRdPortI(SBSR, 3) || BitRdPortI(SBSR, 2));
    serBwrFlush(); //listo el buffer para la siguiente lectura
    serBrdFlush();
}

//FUNCION PARA ESCRIBIR EL PUERTO RS485*****
void wr_RS485(char *buf, int tam)
{
    ser485Tx(); // enable transmitter
    serDwrite((char *)buf,tam); // envia respuesta
    while (serDwrFree() != DOUTBUFSIZE);
    // then, wait until the Tx data register and the Tx shift register
    // are both empty
    while (BitRdPortI(SDSR, 3) || BitRdPortI(SDSR, 2));
    serDwrFlush(); //listo el buffer para la siguiente lectura
    serDrdFlush();
}

//FUNCION PARA INICIALIZAR EL PUERTO RS232*****
void iniSerial() //Inicializa el puerto serial
{
    // Open serial port
    serBopen(BAUD232);
    serBparity(PARAM_NOPARITY);
    serBdatabits(PARAM_8BIT);
    serBflowcontrolOff();
    iniRS485();//temporal hasta tener el OP6810
    serMode(0);
    // Can use serial mode 0 for 3 wire RS232 operation
    // and serMode must be executed after the serXopen
    // function(s). (Usada solo con PowerCoreFLEX prototyping board)
    // Clear serial buffers
    serBwrFlush();
    serBrdFlush();
    // Clear serial buffers
    serBwrFlush();
    serBrdFlush();
}

//FUNCION PARA INICIALIZAR EL PUERTO RS485*****
void iniRS485()
{
    // Open serial port
    serDopen(BAUD232);
    serDparity(PARAM_NOPARITY);
    serDdatabits(PARAM_8BIT);
    serDflowcontrolOff();
}

//FUNCION QUE PROCESA LOS DATOS RECIBIDOS DEL PC*****
cofunc void comPC(char* buf, int tam)
{
    struct tm rtc;// estructura para manejar datos del reloj de tiempo real

```

```

char respuesta[BOUTBUFSIZE];
int i;

union//Union para la recepcion de datos
{
    char bytes[2];
    int entero;
} campos;

switch(buf[0])//indica que funcion se va a realizar
{
    case 10://Iguala reloj de tiempo real
        rtc.tm_sec=buf[1];
        rtc.tm_min=buf[2];
        rtc.tm_hour=buf[3];
        rtc.tm_mday=buf[4];
        rtc.tm_mon=buf[5];
        rtc.tm_year=buf[6];
        rtc.tm_wday=buf[7];
        tm_wr(&rtc); //iguala el reloj de tiempo real
        SEC_TIMER = mktime(&rtc); //iguala la variable de tiempo del sistema
        printf("\nigual fecha(mes/dia/año hh/mm/ss): %02d/%02d/%04d
                %02d:%02d:%02d\n\n",rtc.tm_mon,rtc.tm_mday,1900+rtc.tm_year,
                rtc.tm_hour, rtc.tm_min, rtc.tm_sec);
        respuesta[0]=3; //Numero de datos a enviar
        respuesta[1]=10;
        respuesta[2]=0;
        wr_serial(respuesta, (int)respuesta[0]); //Indica la operacion se realizo
                                                correctamente
        break;

    case 11://Comprueba conexion con el PC
        respuesta[0]=3; //Numero de datos a enviar
        respuesta[1]=11;
        respuesta[2]=0;
        wr_serial(respuesta, (int)respuesta[0]); //Indica la operacion se realizo
                                                correctamente
        break;

    case 12://Numero de dias personalizados a recibir
        campos.bytes[0]=buf[1];
        campos.bytes[1]=buf[2];
        tot_Personalizados=campos.entero;
        dias_recibidos=0;
        respuesta[0]=3; //Numero de datos a enviar
        respuesta[1]=12;
        respuesta[2]=0;
        wr_serial(respuesta, (int)respuesta[0]); //Indica la operacion se realizo
                                                correctamente
        break;

    case 13://Recepcion de dias personalizados
        H_Personalizados[dias_recibidos].dia=buf[1];
        H_Personalizados[dias_recibidos].mes=buf[2];
        for(i=3;i<=26;i++)
            H_Personalizados[dias_recibidos].horas[i-3]=buf[i];
        dias_recibidos++;
        respuesta[0]=3; //Numero de datos a enviar
        respuesta[1]=13;
        respuesta[2]=0;

```

```
        wr_serial(respuesta, (int)respuesta[0]); //Indica la operacion se realizo
                                                correctamente
        break;

case 14: //Numero de dias feriados a recibir
        tot_Feridos=buf[1];
        dias_recibidos=0;
        respuesta[0]=3; //Numero de datos a enviar
        respuesta[1]=14;
        respuesta[2]=0;
        wr_serial(respuesta, (int)respuesta[0]); //Indica la operacion se realizo
                                                correctamente
        break;

case 15: // Recepcion de dias feriados
        L_Feridos[dias_recibidos].dia=buf[1];
        L_Feridos[dias_recibidos].mes=buf[2];
        L_Feridos[dias_recibidos].dia);
        dias_recibidos++;
        respuesta[0]=3; //Numero de datos a enviar
        respuesta[1]=15;
        respuesta[2]=0;
        wr_serial(respuesta, (int)respuesta[0]); //Indica la operacion se realizo
                                                correctamente
        break;

case 16: //Horario Feriados
        for(i=1;i<=24;i++)
            H_Feridos[i-1]=buf[i];
        respuesta[0]=3; //Numero de datos a enviar
        respuesta[1]=16;
        respuesta[2]=0;
        wr_serial(respuesta, (int)respuesta[0]); //Indica la operacion se realizo
                                                correctamente
        break;

case 17://Horario Fin de Semana
        for(i=1;i<=24;i++)
            H_FinSemana[i-1]=buf[i];
        respuesta[0]=3; //Numero de datos a enviar
        respuesta[1]=17;
        respuesta[2]=0;
        wr_serial(respuesta, (int)respuesta[0]); //Indica la operacion se realizo
                                                correctamente
        break;

case 18://Horario Dia Ordinario
        for(i=1;i<=24;i++)
            H_Ordinario[i-1]=buf[i];
        respuesta[0]=3; //Numero de datos a enviar
        respuesta[1]=18;
        respuesta[2]=0;
        wr_serial(respuesta, (int)respuesta[0]); //Indica la operacion se realizo
                                                correctamente
        break;

case 19://Numero de dias personalizados a enviar
        campos.entero=tot_Personalizados;
        respuesta[0]=4;//datos a enviar
        respuesta[1]=19;//codigo de la funcion
```

```
    respuesta[2]=campos.bytes[0];
    respuesta[3]=campos.bytes[1];
    dias_recibidos=0;
    for(i=0;i<=3;i++)
        printf("\nresp[%d]=%d",i,respuesta[i]);
    wr_serial(respuesta, (int)respuesta[0]); //Responde
    break;

case 20://Envio de dias perzonalizados
    respuesta[0]=28;//datos a enviar
    respuesta[1]=20;
    respuesta[2]=H_Personalizados[dias_recibidos].dia;
    respuesta[3]=H_Personalizados[dias_recibidos].mes;
    for(i=0;i<=23;i++)
        respuesta[i+4]= H_Personalizados[dias_recibidos].horas[i];
    dias_recibidos++;
    wr_serial(respuesta, (int)respuesta[0]); //Responde
    break;

case 21://Numero de dias feriados a enviar
    respuesta[0]=3;//datos a enviar
    respuesta[1]=21;
    respuesta[2]=tot_Feriados;
    dias_recibidos=0;
    wr_serial(respuesta,(int)respuesta[0]); //Responde
    break;

case 22: //Envio de dias feriados
    respuesta[0]=4;//datos a enviar
    respuesta[1]=22;
    respuesta[2]=L_Feriados[dias_recibidos].dia;
    respuesta[3]=L_Feriados[dias_recibidos].mes;
    dias_recibidos++;
    wr_serial(respuesta,(int)respuesta[0]); //Responde
    break;

case 23://Horario Feriados (envio)
    respuesta[0]=26;//datos a enviar
    respuesta[1]=23;
    for(i=0;i<=23;i++)
        respuesta[i+2]= H_Feriados[i];
    wr_serial(respuesta, (int)respuesta[0]); //Responde
    break;

case 24: //Horario Fin de Semana (envio)
    respuesta[0]=26;//datos a enviar
    respuesta[1]=24;
    for(i=0;i<=23;i++)
        respuesta[i+2]= H_FinSemana[i];
    wr_serial(respuesta, (int)respuesta[0]); //Responde
    break;

case 25://Horario Dia Ordinario (envio)
    respuesta[0]=26;//datos a enviar
    respuesta[1]=25;
    for(i=0;i<=23;i++)
        respuesta[i+2]= H_Ordinario[i];
    wr_serial(respuesta, (int)respuesta[0]); //Responde
    break;
```



```

        *highlight = 0;
        wKey = -1;
        break;

    case 'U':    // Scroll-up by one line
        *highlight -= 1;
        if(*highlight < 0)
        {
            *offset -= 1;
            *highlight = 0;
        }
        if(*offset == 0 && *highlight == 0)
            *highlight = 1;
        wKey = -1;
        break;

    case 'D':    // Scroll-down by one line
        if((*offset + *highlight) + 1 < (num_options - 1))
        {
            *highlight += 1;
            if(*highlight > 3)
            {
                *offset += 1;
                *highlight = 3;
            }
        }
        wKey = -1;
        break;

    case 'E':    // Select option
        wKey = *offset + *highlight;
        break;

    case 'L':
        wKey=-2;
        break;

    default:
        wKey = -1;
        break;
}
}
return(wKey);
}

```

```

int display_menu ( char **line, int *state, int level, int num_options)
{
    int menu_option;
    int i;
    struct menu_infor *ptr;
    ptr = &menu[level];
    menu_option = 0;           // Initially set to no option selected
    switch(*state)
    {
        case 0: // Set menu parameters in menu structure array
            ptr->current_offset = 0;           // Initialize menu line index
            ptr->highlight = 1;               // Assumes all menus have a heading
            ptr->new_offset = !ptr->current_offset;
            keyProcess ();
            // Make sure no key is being pressed initially

```

```

        if(keyGet() == 0)
        {
            *state = 4;
            break;
        }
        break;

case 1: // Get the user's option
        menu_option = GetKeypadOption(&ptr->current_offset, &ptr->highlight,
                                     num_options);

        if(menu_option == -1)
        {
            // Check if user selected the scrolling option
            glSetBrushType(PIXXOR);
            glBlock (0, ptr->lasthighlight*8, 122, 8);
            glSetBrushType(PIXBLACK);
            // Set menu option to zero due to scrolling operation
            menu_option = 0;
            *state = 4;
        }
        if(menu_option== -2) menu_option=(int)SALIR; //SALIR
        break;

case 2: // Refresh menu options
        glBuffLock();
        glBlankScreen();
        for(i=0; i < 4; i++)
        {
            // Display up to 4 lines of menu options
            TextGotoXY(&textWindow, 0, i);
            TextPrintf(&textWindow, "%s", line[ptr->current_offset]);
            if(*line[ptr->current_offset + 1] == '\0')
            {
                break;
            }
            ptr->current_offset++;
        }
        // Reset the offset back to the first option displayed
        ptr->current_offset = ptr->current_offset-i;
        ptr->new_offset = ptr->current_offset;
        glSetBrushType(PIXXOR);
        glBlock (0, ptr->highlight*8, 122, 8);
        glSetBrushType(PIXBLACK);
        glBuffUnlock();
        ptr->lasthighlight = ptr->highlight;
        *state = 1;
        break;

case 4: // Display menu option
        if(ptr->current_offset != ptr->new_offset)
        {
            glBuffLock();
            glBlankScreen();
            for(i=0; i < 4; i++)
            {
                // Display up to 4 lines of menu options
                TextGotoXY(&textWindow, 0, i);
                TextPrintf(&textWindow, "%s", line[ptr->current_offset]);
                if(*line[ptr->current_offset + 1] == '\0')
                {
                    break;
                }
            }
        }
    
```

```

        ptr->current_offset++;
    }
    glBuffUnlock();
    // Reset the offset back to the first option displayed
    ptr->current_offset = ptr->current_offset-i;
    ptr->new_offset = ptr->current_offset;
}
glSetBrushType(PIXXOR);
glBlock (0, ptr->highlight*8, 122, 8);
glSetBrushType(PIXBLACK);
ptr->lasthighlight = ptr->highlight;
*state = 1;
break;

default:
    *state = 0;
    break;
}
return(menu_option);
}

//-----
// Format the Date and Time for the LCD display
//-----
void FormatDateTime ( void )
{
    char Day[4], Mon[4];
    ulTime = read_rtc ();           // get the RTC value
    mktime( &CurTime, ulTime );   // convert seconds to date values
    strncpy ( Day, &Days[CurTime.tm_wday*3], 3 );
    strncpy ( Mon, &Months[(CurTime.tm_mon-1)*3], 3 );
    Day[3] = 0;
    Mon[3] = 0;
    sprintf ( szTime, "%s %s %d, %d \n%02d:%02d:%02d",
        Day, Mon, CurTime.tm_mday, CurTime.tm_year+1900,
        CurTime.tm_hour, CurTime.tm_min, CurTime.tm_sec );
}

//-----
// Display the Date and Time on the LCD display
//-----
int dispDate( void )
{
    static int status;
    int wKey;
    costate
    {
        // Get current Date/Time
        status = 0;
        ulTime = read_rtc ();           // get the RTC value
        mktime( &CurTime, ulTime );   // convert seconds to date values
        FormatDateTime();               // convert to text
        waitFor(DelayMs(5));
        // Display Date and Time
        glBuffLock();
        TextGotoXY(&textWindow, 0, 0);
        TextPrintf(&textWindow, "%s\n", szTime);
        waitFor(DelayMs(5));
        // Display user exit message
        TextGotoXY(&textWindow, 0, 3);
    }
}

```

```

        TextPrintf(&textWindow, "Press Key to EXIT");
        waitfor(DelayMs(5));
        glBuffUnlock();
        // Wait for key to be pressed to exit
        waitfor(((wKey = keyGet()) != 0) || DelayMs(100));
        if(wKey != 0)
        {
            t_pantalla=(int)TPANTALLA;
            glBlankScreen();
            status = 1;
        }
    }
    return(status);
}

//-----
// Date and Time prompt message routine
//-----
void date_prompt(char *ptr, int *col, int *row)
{
    glBlankScreen();
    TextGotoXY(&textWindow, 0, 0);
    TextPrintf(&textWindow, "%s", ptr);
    TextCursorPosition(&textWindow, col, row);
    TextGotoXY(&textWindow, 0, 3);
    TextPrintf(&textWindow, "Continuar...<ENTER>");
}

//-----
// Set Date and Time
//-----
cofunc void SetDateTime( void )
{
    int wKey;
    int col, row;
    char buffer[256];
    fieldupdate dateTime;
    struct tm reloj;
    // Setup for FAST key repeat after holding down key for 12 ticks
    keyConfig ( 6,'E',0, 12, 1, 1, 1 );
    keyConfig ( 2,'D',0, 12, 1, 1, 1 );
    keyConfig ( 5,'+',0, 12, 1, 1, 1 );
    keyConfig ( 1,'U',0, 12, 1, 1, 1 );
    keyConfig ( 4,'-',0, 12, 1, 1, 1 );
    keyConfig ( 0,'L',0, 0, 0, 0, 0 );
    date_prompt("-MODIF. FECHA/HORA-\nIngrese Año: ", &col, &row);
    tm_rd(&reloj);
    dateTime.data = (int)reloj.tm_year+1900;
    while(1)
    {
        sprintf(buffer, "%04d", dateTime.data);
        TextGotoXY(&textWindow, col, row);
        TextPrintf(&textWindow, "%s", buffer);
        while((wKey = ProcessKeyField(NUMBER, &dateTime)) == 0) yield;
        if(dateTime.data < 1900 || dateTime.data > 2047)
        {
            dateTime.data = 2006;
        }
        if(wKey == 'E')
        {

```

```

        if( dateTime.data >= 1900 && dateTime.data < 2048)
        {
            CurTime.tm_year = dateTime.data - 1900; // offset from 1900
            break;
        }
    }
    if(wKey == 'L') return; //Sale del menu
    yield;
}
date_prompt("-MODIF. FECHA/HORA-\nIngrese el mes: ", &col, &row);
dateTime.data = (int)reloj.tm_mon;
while(1)
{
    sprintf(buffer, "%02d", dateTime.data);
    TextGotoXY(&textWindow, col, row);
    TextPrintf(&textWindow, "%s", buffer);
    while((wKey = ProcessKeyField(NUMBER, &dateTime)) == 0) yield;
    if(wKey == 'E')
    {
        if( dateTime.data >= 1 && dateTime.data < 13 )
        {
            CurTime.tm_mon = dateTime.data;
            break;
        }
    }
    if(dateTime.data < 1 || dateTime.data > 12)
    {
        dateTime.data = (dateTime.data < 1) ? 12 : 1;
    }
    if(wKey == 'L') return; //Sale del menu
    yield;
}
date_prompt("-MODIF. FECHA/HORA-\nIngrese dia del mes: ", &col, &row);
dateTime.data = (int)reloj.tm_mday;
while(1)
{
    sprintf(buffer, "%02d", dateTime.data);
    TextGotoXY(&textWindow, col, row);
    TextPrintf(&textWindow, "%s", buffer);
    while((wKey = ProcessKeyField(NUMBER, &dateTime)) == 0) yield;
    if(wKey == 'E')
    {
        if( dateTime.data >= 1 && dateTime.data < 32) {
            CurTime.tm_mday = dateTime.data;
            break;
        }
    }
    if(dateTime.data < 1 || dateTime.data > 31)
    {
        dateTime.data = (dateTime.data < 1) ? 31 : 1;
    }
}
if(wKey == 'L') return; //Sale del menu
yield;
}
date_prompt("-MODIF. FECHA/HORA-\nIngrese hora (24hr): ", &col, &row);
dateTime.data = (int)reloj.tm_hour;
while(1)
{
    sprintf(buffer, "%02d", dateTime.data);
    TextGotoXY(&textWindow, col, row);

```

```

TextPrintf(&textWindow, "%s", buffer);
while((wKey = ProcessKeyField(NUMBER, &dateTime)) == 0)yield ;
if(wKey == 'E')
{
    if(dateTime.data >= 0 && dateTime.data < 24) {
        CurTime.tm_hour = dateTime.data;
        break;
    }
}
if(dateTime.data < 0 || dateTime.data > 23)
{
    dateTime.data = (dateTime.data < 0) ? 23 : 0;
}
if(wKey == 'L') return; //Sale del menu
yield;
}
date_prompt("-MODIF. FECHA/HORA-\nIngrese minuto: ", &col, &row);
dateTime.data = (int)reloj.tm_min;
while(1)
{
    sprintf(buffer, "%02d", dateTime.data);
    TextGotoXY(&textWindow, col, row);
    TextPrintf(&textWindow, "%s", buffer);
    while((wKey = ProcessKeyField(NUMBER, &dateTime)) == 0) yield;
    if(wKey == 'E')
    {
        if( dateTime.data >= 0 && dateTime.data < 60) {
            CurTime.tm_min = dateTime.data;
            break;
        }
        if(wKey == 'E')
        {
            break;
        }
    }
    if(dateTime.data < 0 || dateTime.data > 59)
    {
        dateTime.data = (dateTime.data < 0) ? 59 : 0;
    }
}
if(wKey == 'L') return; //Sale del menu
yield;
}
CurTime.tm_sec = 0;
ulTime = mktime ( &CurTime );           // get seconds from 1/1/1980
write_rtc ( ulTime );                    // set the real time clock
 keypadDef();
glBlankScreen();
while(1)
{
    // Get current Date/Time
    FormatDateTime();                      // convert to text
    // Display Date and Time
    glBuffLock();
    TextGotoXY(&textWindow, 0, 0);
    TextPrintf(&textWindow, "-FECHA/HORA ACTUAL-\n%s\n", szTime);
    // Display user exit message
    TextGotoXY(&textWindow, 0, 3);
    TextPrintf(&textWindow, "Salir...<ENTER>");
    glBuffUnlock();
    keyProcess ();
}

```

```

        if((wKey = keyGet()) != 0)
        {
            t_pantalla=(int)TPANTALLA;
            glBlankScreen();
            break;
        }
    yield;
}

//-----
// Comprueba que no se haya perdido la informacion
//-----
void SignOnMessage(void)
{
    char wKey;
    if(test_important_values()==0)//se han perdido los valores del sistema
    {
        ini_variables();
        glBlankScreen();
        glBuffLock();
        glPlotLine(0, 6, 120, 6);
        glPlotLine(0, 4, 120, 4);
        TextGotoXY(&textWindow, 0, 1);
        TextPrintf(&textWindow, " SE HA PERDIDO LA\n INFORMACION\n");
        glPlotLine(0,28, 120, 28);
        glPlotLine(0,26, 120, 26);
        glBuffUnlock();
        while((wKey=keyGet())!='E')
        {
            keyProcess();
        }
        glBlankScreen();
        glBuffLock();
        TextGotoXY(&textWindow, 0,0);
        TextPrintf(&textWindow, "El sistema iniciara con valores por defecto. Revise la
                                conexion de la \n");

        glBuffUnlock();
        while((wKey=keyGet())!='E')
        {
            keyProcess();
        }
        glBlankScreen();
        glBuffLock();
        TextGotoXY(&textWindow, 0,0);
        TextPrintf(&textWindow, "bateria de back-up oreemplacela. Presione <ENTER>
                                para continuar... \n");

        glBuffUnlock();
        while((wKey=keyGet())!='E')
        {
            keyProcess();
        }
    }
}

//MENUS
//-----
// Routine for sub_menu backlight options
//-----
int backlight_menu( void )

```

```

{
    static int done;
    static int state;
    int option;
    costate
    {
        done = FALSE;
        state = MENU_INIT;
        while(!done)
        {
            // Display the MAIN MENU
            waitfor((option = display_menu(backlight, &state, LVL_BACKLIGHT,
                                         NUM_BACKLIGHT_OPTS)) > 0);

            // Get menu option from the user
            switch(option)
            {
                // Turn Backlight OFF
                case 1:
                    backlightAUTO=FALSE;
                    glBackLight(0);
                    state = MENU_NO_CHANGE;
                    break;

                // Turn Backlight ON
                case 2:
                    backlightAUTO=FALSE;
                    glBackLight(1);
                    state = MENU_NO_CHANGE;
                    break;

                //AUTOMATICO
                case 3:
                    backlightAUTO=TRUE;
                    glBackLight(1);
                    state = MENU_NO_CHANGE;
                    break;

                case 4://volver
                    done = TRUE;
                    break;

                case (int)SALIR://SALIR
                    done=(int)SALIR;
                    break;

                // User made invalid selection
                default:
                    break;
            }
        }
    }
    return(done);
}

//-----
// Routine for sub_menu Modificar Dia options
//-----
static int mod_dia_menu( void )
{
    static int done;

```



```

state = MENU_INIT;
while(!done)
{
    // Display the MAIN MENU
    waitfor((option = display_menu(selec_mes, &state, LVL_SELEC_MES,
                                  NUM_SELEC_MES_OPTS)) > 0);

    // Get menu option from the user
    switch(option)
    {
        case 1: // ENERO
        case 3: //MARZO
        case 5: //MAYO
        case 7: //JULIO
        case 8: //AGOSTO
        case 10: //NOVIEMBRE
        case 12: // DICIEMBRE
            waitfor((seleccion=selec_dia_menu(31,(char)option))>0);//<<<<<<
            state = MENU_REFRESH;
            if(seleccion==(int)SALIR) done=(int)SALIR;
            break;

        case 4: //ABRIL
        case 6: //MJUNIO
        case 9: //SEPTIEMPBRE
        case 11: //OCTUBRE
            waitfor((seleccion=selec_dia_menu(30,(char)option))>0);//<<<<<
            state = MENU_REFRESH;
            if(seleccion==(int)SALIR) done=(int)SALIR;
            break;

        case 2: //FEBRERO
            waitfor((seleccion=selec_dia_menu(28,(char)option))>0);//<<<<<<
            state = MENU_REFRESH;
            if(seleccion==(int)SALIR) done=(int)SALIR;
            break;

        case 13: //volver
            done = TRUE;
            break;

        case (int)SALIR://SALIR
            done=(int)SALIR;
            break;

        // User made invalid selection
        default:
            break;
    }
}
return(done);
}

//-----
// Routine for sub_menu SELECCIONAR DIA options
//-----
static int selec_dia_menu(char nDias, char mes )
{
    static int done;
    static int state;

```

```
int option, seleccion;
costate
{
    done = FALSE;
    state = MENU_INIT;
    while(!done)
    {
        // Display the MAIN MENU
        if(nDias==31) waitfor((option = display_menu(selec_dia31, &state,
            LVL_SELEC_DIA31, NUM_SELEC_DIA31_OPTS)) > 0);
        if(nDias==30) waitfor((option = display_menu(selec_dia30, &state,
            LVL_SELEC_DIA30, NUM_SELEC_DIA30_OPTS)) > 0);
        if(nDias==28) waitfor((option = display_menu(selec_dia28, &state,
            LVL_SELEC_DIA28, NUM_SELEC_DIA28_OPTS)) > 0);
        // Get menu option from the user
        switch(option)
        {
            case 1:
            case 2:
            case 3:
            case 4:
            case 5:
            case 6:
            case 7:
            case 8:
            case 9:
            case 10:
            case 11:
            case 12:
            case 13:
            case 14:
            case 15:
            case 16:
            case 17:
            case 18:
            case 19:
            case 20:
            case 21:
            case 22:
            case 23:
            case 24:
            case 25:
            case 26:
            case 27:
            case 28:
            case 29:
                if(nDias==28 && option==29)//volver
                {
                    done = TRUE;
                    break;
                }

            case 30:
            case 31:
                if(nDias==30 && option==31)//volver
                {
                    done = TRUE;
                    break;
                }
                printf("\nPersonalizado (DIA S) dia=%d, mes=%d",option,mes);
        }
    }
}
```

```

        waitfor((seleccion=hor_dia_menu((int)PERSONALIZADO,
                                        mes,option))>0);//<
        state = MENU_REFRESH;
        if(seleccion==(int)SALIR) done=(int)SALIR;
        break;

    case 32: //volver
        done = TRUE;
        break;

    case (int)SALIR://SALIR
        done=(int)SALIR;
        break;

    // User made invalid selection
    default:
        break;
    }
}
}
return(done);
}

//-----
// Routine for sub_menu SELECCIONAR HORARIO options
//-----
static int selec_hor_menu( void )
{
    static int done;
    static int state;
    int option;
    int seleccion;
    costate
    {
        done = FALSE;
        state = MENU_INIT;
        while(!done)
        {
            // Display the MAIN MENU
            waitfor((option = display_menu(selec_hor, &state, LVL_SELEC_HOR,
                                           NUM_SELEC_HOR_OPTS)) > 0);

            // Get menu option from the user
            switch(option)
            {
                // D. ORDinario
                case 1:
                    waitfor((seleccion=hor_dia_menu((int)ORDINARIO,0,0))>0);//<<
                    state = MENU_REFRESH;
                    if(seleccion==(int)SALIR) done=(int)SALIR;
                    break;

                // D. Feriado
                case 2:
                    waitfor((seleccion=hor_dia_menu((int)FERIADO,0,0))>0);//<<
                    state = MENU_REFRESH;
                    if(seleccion==(int)SALIR) done=(int)SALIR;
                    break;

                //Fin Semana
                case 3:

```



```

        case 12:
        case 13:
        case 14:
        case 15:
        case 16:
        case 17:
        case 18:
        case 19:
        case 20:
        case 21:
        case 22:
        case 23:
        case 24:
        case 25: //todas
            glBlankScreen();
            wfd TempHor( modo, mes, dia,(char)option );
            state=MENU_REFRESH;
            break;

        case 26: //volver
            done = TRUE;
            break;

        case (int)SALIR://SALIR
            done=(int)SALIR;
            break;

        // User made invalid selection
        default:
            break;
    }
}
}
return(done);
}

//-----
// MODIFICA TEMPORIZACION DE LAS HORAS
//-----
static cofunc void TempHor( int modo, char mes, char dia, char hora )
{
    int wKey;
    int col, row, i, j, tipo;
    char buffer[256];
    char tiempo;
    fieldupdate dateTime;
    printf("\nPersonalizado (TEMP) dia=%d, mes=%d",dia,mes);
    // Setup for FAST key repeat after holding down key for 12 ticks
    keyConfig ( 6,'E',0, 12, 1, 1, 1 );
    keyConfig ( 2,'D',0, 12, 1, 1, 1 );
    keyConfig ( 5,'+',0, 12, 1, 1, 1 );
    keyConfig ( 1,'U',0, 12, 1, 1, 1 );
    keyConfig ( 4,'-',0, 12, 1, 1, 1 );
    keyConfig ( 0,'L',0, 0, 0, 0, 0 );
    switch(modo)
    {
        case (int)PERSONALIZADO:
            date_prompt(" -TEMPORIZACION-\nIngrese Segundos: ", &col, &row);
            if(hora == 25)//todas las horas
                dateTime.data = 10; //10 segundos
    }
}

```

```

else //hora especifica
{
    wfd tiempo= buscarHorario((char)mes,(char)dia,(char)(hora-1),1);
    dateTime.data = (int)tiempo;//Temporizacion actual
}
while(1)
{
    sprintf(buffer, "%02d", dateTime.data);
    TextGotoXY(&textWindow, col, row);
    TextPrintf(&textWindow, "%s", buffer);
    while((wKey = ProcessKeyField(NUMBER, &dateTime)) == 0) yield;
    if(dateTime.data < 0 || dateTime.data > 99)
    {
        if(hora==25)//todas las horas
            dateTime.data=10;//10 segundos
        else//hora especifica
            dateTime.data = (int)tiempo;//Temporizacion actual
    }
    if(wKey == 'E')
    {
        if( dateTime.data >= 0 && dateTime.data <= 99)
        {
            for(i=0;i<tot_Personalizados;i++)
            {
                if(mes==H_Personalizados[i].mes &&
                    dia==H_Personalizados[i].dia)
                    break;
                yield;
            }
            if(i==tot_Personalizados)//no es dia personalizado
            {
                if(hora==25)//todas las horas
                    for(j=0;j<24;j++)
                        H_Personalizados[i].horas[j]=
                            (char)dateTime.data;
                else //hora especifica
                {
                    for(j=0;j<24;j++)
                        wfd
                            H_Personalizados[i].horas[j]=buscarHorario((
                                char)mes,(char)dia,(char)j,1);//Horario anterior
                    H_Personalizados[i].horas[hora-1]=
                        (char)dateTime.data;
                }
                H_Personalizados[i].dia=(char)dia;
                H_Personalizados[i].mes=(char)mes;
                printf("\nPersonalizado DIA= %d, MES=%d",dia,
                    mes);
                printf("\nPersonalizado DIA= %d,MES=%d",
                    H_Personalizados[i].dia, H_Personalizados[i].mes);
                tot_Personalizados++;//actualiza personalizados
            }
        }
        else // si es personalizado
        {
            if(hora==25)//todas las horas
                for(j=0;j<24;j++)
                    H_Personalizados[i].horas[j]=(char)
                        dateTime.data;
            else //hora especifica
                H_Personalizados[i].horas[hora-1]=

```

```

                                                                    (char)dateTime.data;
                                                                    }
                                                                    break;
                                                                    }
                                                                    }
                                                                    }
                                                                    if(wKey == 'L') return; //Sale del menu
                                                                    yield;
                                                                    }
                                                                    break;

case (int)ORDINARIO:
    date_prompt(" -TEMPORIZACION-\nIngrese Segundos: ", &col, &row);
    if(hora == 25)//todas las horas
        dateTime.data = 10; //10 segundos
    else //hora especifica
        dateTime.data = H_Ordinario[hora-1];//Temporizacion actual
    while(1)
    {
        sprintf(buffer, "%02d", dateTime.data);
        TextGotoXY(&textWindow, col, row);
        TextPrintf(&textWindow, "%s", buffer);
        while((wKey = ProcessKeyField(NUMBER, &dateTime)) == 0) yield;
        if(dateTime.data < 0 || dateTime.data > 99)
        {
            if(hora==25)//todas las horas
                dateTime.data=10;//10 segundos
            else//hora especifica
                dateTime.data = H_Ordinario[hora-1];//Temporizacion actual
        }
        if(wKey == 'E')
        {
            if( dateTime.data >= 0 && dateTime.data <= 99)
            {
                if(hora==25)//todas las horas
                    for(i=0;i<24;i++)
                        H_Ordinario[i]= dateTime.data;
                else //hora especifica
                    H_Ordinario[hora-1]= dateTime.data;
                break;
            }
        }
        if(wKey == 'L') return; //Sale del menu
        yield;
    }
    break;

case (int)FERIADO:
    date_prompt(" -TEMPORIZACION-\nIngrese Segundos: ", &col, &row);
    if(hora == 25)//todas las horas
        dateTime.data = 10; //10 segundos
    else //hora especifica
        dateTime.data = H_Feridos[hora-1];//Temporizacion actual
    while(1)
    {
        sprintf(buffer, "%02d", dateTime.data);
        TextGotoXY(&textWindow, col, row);
        TextPrintf(&textWindow, "%s", buffer);
        while((wKey = ProcessKeyField(NUMBER, &dateTime)) == 0) yield;
        if(dateTime.data < 0 || dateTime.data > 99)
        {

```

```

        if(hora==25)//todas las horas
            dateTime.data=10;//10 segundos
        else//hora especifica
            dateTime.data = H_Feriados[hora-1];//Temporizacion actual
    }
    if(wKey == 'E')
    {
        if( dateTime.data >= 0 && dateTime.data <= 99)
        {
            if(hora==25)//todas las horas
                for(i=0;i<24;i++)
                    H_Feriados[i]= dateTime.data;
            else //hora especifica
                H_Feriados[hora-1]= dateTime.data;
            break;
        }
    }
    if(wKey == 'L') return; //Sale del menu
    yield;
}
break;

case (int)FIN_SEMANA:
    date_prompt(" -TEMPORIZACION-\nIngrese Segundos: ", &col, &row);
    if(hora == 25)//todas las horas
        dateTime.data = 10; //10 segundos
    else //hora especifica
        dateTime.data = H_FinSemana[hora-1];//Temporizacion actual
    while(1)
    {
        sprintf(buffer, "%02d", dateTime.data);
        TextGotoXY(&textWindow, col, row);
        TextPrintf(&textWindow, "%s", buffer);
        while((wKey = ProcessKeyField(NUMBER, &dateTime)) == 0) yield;
        if(dateTime.data < 0 || dateTime.data > 99)
        {
            if(hora==25)//todas las horas
                dateTime.data=10;//10 segundos
            else//hora especifica
                dateTime.data=H_FinSemana[hora-1];//Temporizac. actual
        }
        if(wKey == 'E')
        {
            if( dateTime.data >= 0 && dateTime.data <= 99)
            {
                if(hora==25)//todas las horas
                    for(i=0;i<24;i++)
                        H_FinSemana[i]= dateTime.data;
                else //hora especifica
                    H_FinSemana[hora-1]= dateTime.data;
                break;
            }
        }
        if(wKey == 'L') return; //Sale del menu
        yield;
    }
    break;

default:
    break;

```

```

    }
    keypadDef();
    glBlankScreen();
    glBuffLock();
    glPlotLine(0, 6, 120, 6);
    glPlotLine(0, 4, 120, 4);
    TextGotoXY(&textWindow, 0, 1);
    TextPrintf(&textWindow, " HORARIO HA SIDO\n  MODIFICADO\n");
    glPlotLine(0,28, 120, 28);
    glPlotLine(0,26, 120, 26);
    glBuffUnlock();
    waitFor(DelaySec(2));
    glBlankScreen();
}

//-----
// Routine for msg_voz options
//-----
int msg_voz_menu( void )
{
    static int done;
    static int state, seleccion;
    int option;
    costate
    {
        done = FALSE;
        state = MENU_INIT;
        while(!done)
        {
            // Display the MAIN MENU
            waitFor((option = display_menu(msg_voz, &state, LVL_MSG_VOZ,
                NUM_MSG_VOZ_OPTS)) > 0);

            // Get menu option from the user
            switch(option)
            {
                // APRETURA
                case 1:
                    waitFor((seleccion= op_voz_menu((int)MSG_1))>0);//<<
                    state = MENU_REFRESH;
                    if(seleccion==(int)SALIR) done=(int)SALIR;
                    break;

                // CIERRE
                case 2:
                    waitFor((seleccion= op_voz_menu((int)MSG_2))>0);//<<
                    state = MENU_REFRESH;
                    if(seleccion==(int)SALIR) done=(int)SALIR;
                    break;

                case 3://volver
                    done = TRUE;
                    break;

                case (int)SALIR://SALIR
                    done=(int)SALIR;
                    break;

                // User made invalid selection
                default:
                    state = MENU_REFRESH;
            }
        }
    }
}

```

```

        break;
    }
}
return(done);
}

//-----
// Routine for op_voz options
//-----
int op_voz_menu( int modo )
{
    static int done;
    static int state;
    int option;
    costate
    {
        done = FALSE;
        state = MENU_INIT;
        while(!done)
        {
            // Display the MAIN MENU
            waitfor((option = display_menu(op_voz, &state, LVL_OP_VOZ,
                                           NUM_OP_VOZ_OPTS)) > 0);

            // Get menu option from the user
            switch(option)
            {
                // GRABAR
                case 1:
                    glBlankScreen();
                    wfd RecMsg( modo );
                    state=MENU_REFRESH;
                    break;

                // REPRODUCIR
                case 2:
                    digOut(modo,0);
                    waitfor(DelayMs(200));
                    digOut(modo,1);
                    break;

                //HABILITAR
                case 3:
                    if(modo==MSG_1)
                        msgVar.msg1En=TRUE;
                    if(modo==MSG_2)
                        msgVar.msg2En=TRUE;
                    //actualizar
                    rechecksum_important_values();
                    glBlankScreen();
                    glBuffLock();
                    glPlotLine(0, 6, 120, 6);
                    glPlotLine(0, 4, 120, 4);
                    TextGotoXY(&textWindow, 0, 1);
                    TextPrintf(&textWindow, " MENSAJE HA SIDO\n
                                                                           HABILITADO\n");

                    glPlotLine(0,28, 120, 28);
                    glPlotLine(0,26, 120, 26);
                    glBuffUnlock();
                    waitfor(DelaySec(2));
            }
        }
    }
}

```

```

        glBlankScreen();
        state = MENU_REFRESH;
        break;

//DESHABILITAR
case 4:
    if(modo==(int)MSG_1)
        msgVar.msg1En=FALSE;
    if(modo==(int)MSG_2)
        msgVar.msg2En=FALSE;
    //actualizar
    rechecksum_important_values();
    glBlankScreen();
    glBuffLock();
    glPlotLine(0, 6, 120, 6);
    glPlotLine(0, 4, 120, 4);
    TextGotoXY(&textWindow, 0, 1);
    TextPrintf(&textWindow, " MENSAJE HA SIDO\n
        DESHABILITADO\n");

    glPlotLine(0,28, 120, 28);
    glPlotLine(0,26, 120, 26);
    glBuffUnlock();
    waitfor(DelaySec(2));
    glBlankScreen();
    state = MENU_REFRESH;
    break;

//TIEMPO
case 5:
    glBlankScreen();
    wfd TempMsg( modo );
    state=MENU_REFRESH;
    break;

case 6://volver
    done = TRUE;
    break;

case (int)SALIR://SALIR
    done=(int)SALIR;
    break;

// User made invalid selection
default:
    break;
    }
}
}
return(done);
}

//-----
// MOFIFICA TEMPORIZACION DE LOS MENSAJES
//-----
cofunc void TempMsg( int modo )
{
    int wKey;
    int col, row, i, j, tipo;
    char buffer[256];
    int tiempo;

```

```

fieldupdate dateTime;
// Setup for FAST key repeat after holding down key for 12 ticks
keyConfig ( 6,'E',0, 12, 1, 1, 1 );
keyConfig ( 2,'D',0, 12, 1, 1, 1 );
keyConfig ( 5,'+',0, 12, 1, 1, 1 );
keyConfig ( 1,'U',0, 12, 1, 1, 1 );
keyConfig ( 4,'-',0, 12, 1, 1, 1 );
keyConfig ( 0,'L',0, 0, 0, 0, 0 );
switch(modos)
{
case (int)MSG_1:
    date_prompt("-MENSAJE APERTURA-\nIngrese Segundos: ", &col, &row);
    dateTime.data =(int) msgVar.msg1Tiempo;//Temporizacion actual
    while(1)
    {
        sprintf(buffer, "%02d", dateTime.data);
        TextGotoXY(&textWindow, col, row);
        TextPrintf(&textWindow, "%s", buffer);
        while((wKey = ProcessKeyField(NUMBER, &dateTime)) == 0) yield;
        if(dateTime.data < 0 || dateTime.data > 10)
        {
            dateTime.data =(int) msgVar.msg1Tiempo;//Temporizacion actual
        }
        if(wKey == 'E')
        {
            if( dateTime.data >= 0 && dateTime.data <= 10)
            {
                msgVar.msg1Tiempo=(char)dateTime.data;
                //actualizar
                rechecksum_important_values();
                break;
            }
        }
        if(wKey == 'L') return; //Sale del menu
        yield;
    }
    break;

case (int)MSG_2:
    date_prompt("-MENSAJE DE CIERRE-\nIngrese Segundos: ", &col, &row);
    dateTime.data = (int)msgVar.msg2Tiempo;//Temporizacion actual
    while(1)
    {
        sprintf(buffer, "%02d", dateTime.data);
        TextGotoXY(&textWindow, col, row);
        TextPrintf(&textWindow, "%s", buffer);
        while((wKey = ProcessKeyField(NUMBER, &dateTime)) == 0) yield;
        if(dateTime.data < 0 || dateTime.data > 99)
        {
            dateTime.data = (int)msgVar.msg2Tiempo;//Temporizacion actual
        }
        if(wKey == 'E')
        {
            if( dateTime.data >= 0 && dateTime.data <= 99)
            {
                msgVar.msg2Tiempo=(char)dateTime.data;
                //actualizar
                rechecksum_important_values();
                break;
            }
        }
    }
}

```

```

        }
        if(wKey == 'L') return; //Sale del menu
        yield;
    }
    break;

default:
    break;
}
keypadDef();
glBlankScreen();
glBuffLock();
glPlotLine(0, 6, 120, 6);
glPlotLine(0, 4, 120, 4);
TextGotoXY(&textWindow, 0, 1);
TextPrintf(&textWindow, " TIEMPOS HAN SIDO\n  MODIFICADOS\n");
glPlotLine(0,28, 120, 28);
glPlotLine(0,26, 120, 26);
glBuffUnlock();
waitfor(DelaySec(2));
glBlankScreen();
}

//-----
// GRABA MENSAJE
//-----
cofunc void RecMsg( int modo )
{
    int wKey;
    // Setup for FAST key repeat after holding down key for 12 ticks
    keyConfig ( 6,'E','e',0, 0, 0, 0 );
    keyConfig ( 0,'L',0,0, 0, 0, 0 );
    glBlankScreen();
    glBuffLock();
    glPlotLine(0, 6, 120, 6);
    glPlotLine(0, 4, 120, 4);
    TextGotoXY(&textWindow, 0, 1);
    TextPrintf(&textWindow, "MANTENGA PRESIONADO\n<ENTER> PARA GRABAR\n");
    glPlotLine(0,28, 120, 28);
    glPlotLine(0,26, 120, 26);
    glBuffUnlock();
    while((wKey=keyGet())!='E')
    {
        keyProcess();
        if(wKey=='L') return;
        yield;
    }
    t_pantalla=10;
    glBlankScreen();
    glBuffLock();
    glPlotLine(0, 6, 120, 6);
    glPlotLine(0, 4, 120, 4);
    TextGotoXY(&textWindow, 0, 1);
    TextPrintf(&textWindow, "LIBERE <ENTER> PARA\n  TERMINAR\n");
    glPlotLine(0,28, 120, 28);
    glPlotLine(0,26, 120, 26);
    glBuffUnlock();
    digOut((int)PLAY,1);
    digOut((int)REC,0);
    digOut(modo,0);
}

```

```

while((wKey=keyGet())!='e') //suelta tecla
{
    keyProcess();
    yield;
}
t_pantalla=(int)TPANTALLA;
digOut(modo,1);
digOut((int)REC,1);
digOut((int)PLAY,0);
waitfor(DelayMs(200));
keypadDef();
glBlankScreen();
glBuffLock();
glPlotLine(0, 6, 120, 6);
glPlotLine(0, 4, 120, 4);
TextGotoXY(&textWindow, 0, 1);
TextPrintf(&textWindow, " MENSAJE HA SIDO\n GRABADO\n");
glPlotLine(0,28, 120, 28);
glPlotLine(0,26, 120, 26);
glBuffUnlock();
waitfor(DelaySec(2));
glBlankScreen();
digOut(modo,0);
waitfor(DelayMs(200));
digOut(modo,1);
}

```

```

////////////////////////////////////

```

```

void ini_variables()

```

```

{
    int i;
    //Inicializacion de horarios
    for(i=0;i<24;i++)
    {
        H_Ordinario[i]=10;
        H_FinSemana[i]=10;
        H_Feriados[i]=10;
    }
    L_Feriados[0].mes=1;
    L_Feriados[0].dia=1;
    L_Feriados[1].mes=2;
    L_Feriados[1].dia=27;
    L_Feriados[2].mes=4;
    L_Feriados[2].dia=1;
    L_Feriados[3].mes=5;
    L_Feriados[3].dia=24;
    L_Feriados[4].mes=8;
    L_Feriados[4].dia=10;
    L_Feriados[5].mes=10;
    L_Feriados[5].dia=9;
    L_Feriados[6].mes=11;
    L_Feriados[6].dia=2;
    L_Feriados[7].mes=11;
    L_Feriados[7].dia=3;
    L_Feriados[8].mes=12;
    L_Feriados[8].dia=6;
    L_Feriados[9].mes=12;
    L_Feriados[9].dia=25;
    L_Feriados[10].mes=12;
    L_Feriados[10].dia=31;
}

```

```

tot_Feridos=11;
tot_Personalizados=0;
//Contraseña
password[0]=85; //'U'
password[1]=68; //'D'
password[2]=43; //'+'
password[3]=45; //'-'
//Mensajes
msgVar.msg1En=(char)TRUE;
msgVar.msg1Tiempo=2;
msgVar.msg2En=(char)TRUE;
msgVar.msg2Tiempo=4;
rechecksum_impimportant_values();
}

////////////////////////////////////
int test_impimportant_values()
{
    word      chk;
    chk = getcrc((char*)&msgVar + sizeof(msgVar._crc),
                sizeof(msgVar)-sizeof(msgVar._crc),0 );
    printf( "chk = %04x  want = %04x\n", chk, msgVar._crc );
    return (int)(chk == msgVar._crc);
} /* end test_impimportant_values() */

////////////////////////////////////
void rechecksum_impimportant_values()
{
    /*ASSERT:*/ if( sizeof(msgVar) > 255 ) { printf("CRC has max size 255 bytes"); exit(2); }
    msgVar._crc = getcrc( (char*)&msgVar + sizeof(msgVar._crc), sizeof(msgVar)-
                                                                    sizeof(msgVar._crc), 0 );

    printf( "chk = %04x \n", msgVar._crc );
} /* end rechecksum_impimportant_values() */

////////////////////////////////////
static cofunc void P_Pantalla()
{
    int signMesgDone, i, loop;
    char buffer[256];
    char wKey;
    unsigned long auxT, auxT2;
    t_pantalla=(int)TPANTALLA;
    glBackLight(1);
    auxT2=MS_TIMER;
    glBlankScreen();
    glXPutBitmap (0,0,60,32,Img_OP_2_bmp);
    signMesgDone = FALSE;
    ulTime = read_rtc ();                // get the RTC value
    mktime( &CurTime, ulTime );        // convert seconds to date values
    glPrintf (55, 16,  &fi6x8, " %02d/%02d/%02d", (int)CurTime.tm_mon, (int)CurTime.tm_mday,
                                                    ((int)CurTime.tm_year+1900)%100);
    glPrintf (55, 24,  &fi6x8, " %02d:%02d:%02d", (int)CurTime.tm_hour, (int)CurTime.tm_min,
                                                    (int)CurTime.tm_sec);

    auxT=MS_TIMER;
    while(!signMesgDone)
    {
        i=0;
        sprintf(buffer, "UNIDAD OPERADORA DEL SISTEMA TROLEBUS");
        strcat(buffer, "  Presione cualquier tecla para continuar...  ");
        while(buffer[i] != '\0' && !signMesgDone)

```



```

        if(teclas[n_teclas]!= 0)
        {
            printf("\nteclas[%d]=%c",n_teclas,teclas[n_teclas]);
            n_teclas++;
            auxT=MS_TIMER;
            t_pantalla=(int)TPANTALLA;
        }
        if(n_teclas==4)
        {
            signMesgDone=TRUE;
            break;
        }
        if((auxT+5000)<MS_TIMER)
        {
            t_pantalla=(int)TPANTALLA;
            glBackLight(1);
            glBlankScreen();
            glXPutBitmap (0,0,60,32,img_OP_2_bmp);
            return(0);
        }
        yield;
    }
    yield;
}
}
for(i=0;i<4;i++) printf("\npass[%d]=%c",i,password[i]);
if((password[0]==teclas[0]) && (password[1]==teclas[1]) && (password[2]==teclas[2]) &&
    (password[3]==teclas[3]) )
{
    glBlankScreen();
    t_pantalla=(int)TPANTALLA;
    printf("\nacepto");
    return(1);
}
else
{
    t_pantalla=(int)TPANTALLA;
    glBackLight(1);
    glBlankScreen();
    glXPutBitmap (0,0,60,32,img_OP_2_bmp);
    printf("\nfallo");
    return(0);
}
}

////////////////////////////////////
cofunc void AcercaDe()
{
    int signMesgDone, i, loop;
    char buffer[256];
    char wKey;
    glBlankScreen();
    glXPutBitmap (0,0,55,34,au_bmp);
    glPlotLine(56,10,LCD_XS,10);
    glPlotLine(56,9,LCD_XS,9);
    glPlotLine(56,22,LCD_XS,22);
    glPlotLine(56,23,LCD_XS,23);
    msDelay(500);
    signMesgDone = FALSE;
    while(!signMesgDone)

```

```
{
    i=0;
    sprintf(buffer, "Juan Flores 097199594");
    //strcat(buffer, "  Presione tecla < para continuar... ");
    while(buffer[i] != '\0' && !signMesgDone)
    {
        gIHScroll(56, 12, 72, 8, -6);
        glPrintf (116, 12, &fi6x8, "%c", buffer[i++]);
        for(loop=0; loop < 165; loop++)
        {
            msDelay(1);
            keyProcess ();
            if(keyGet() != 0)
            {
                signMesgDone = TRUE;
            }
            yield;
        }
        yield;
    }
}
```

ANEXO 5

**CÓDIGO DESARROLLADO PARA EL
MICROCONTROLADOR PIC16F84A, EN PCW H 3.2.**

```
#include "C:\prueba\RS232_display.h"
#define ID 1

void putch_to_device(char num);
short salida;
int8 i;

void main()
{
    char num=0;
    char aux;
    i=0;
    salida=0;

    while(1)
    {
        if(kbhit())
        {
            aux=getc();
            if (aux==ID)
            {
                num=getc();
                i=0;
                salida=0;
            }
        }

        if(i==200) i=0;

        if (i==0)
        {
            if(num>=100)
                putch_to_device(num);
        }
        else
            putch_to_device((num%10)&0x0F);

        if (i==100)
        {
            if(num>=100)
                putch_to_device(num);
            else
                putch_to_device((num/10)&0x0F);
        }
        i++;
    }
}

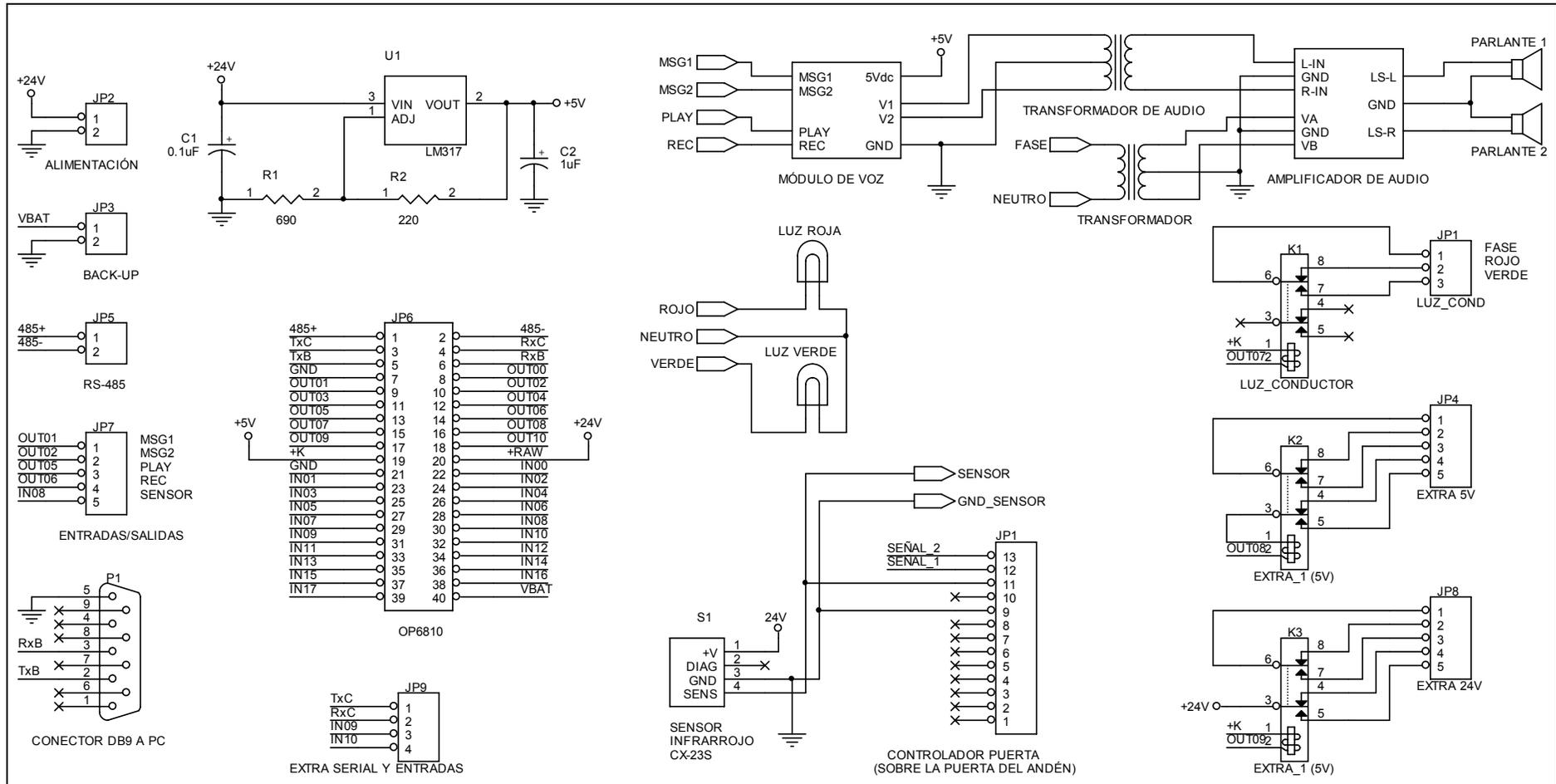
void putch_to_device(char num)
{
    output_b ((0x00));
    output_high(PIN_A0);
    output_high(PIN_A1);
    output_low(PIN_A0);
    output_low(PIN_A1);
    switch(num)
    {
        case 0:
            output_b ((0x3F));
            break;
    }
}
```

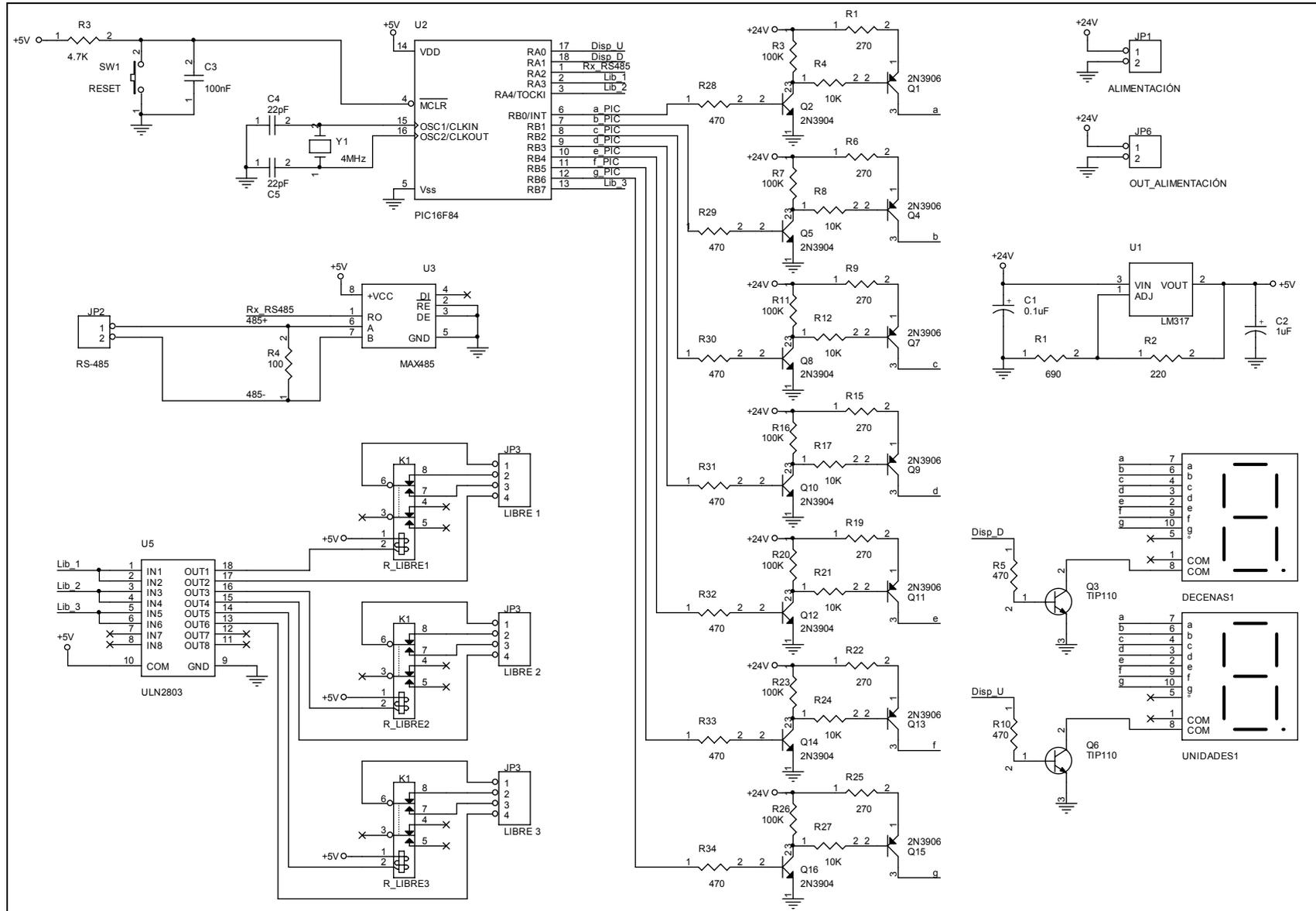
```
        case 1:
            output_b ((0x06));
            break;
        case 2:
            output_b ((0x5B));
            break;
        case 3:
            output_b ((0x4F));
            break;
        case 4:
            output_b ((0x66));
            break;
        case 5:
            output_b ((0x6D));
            break;
        case 6:
            output_b ((0x7D));
            break;
        case 7:
            output_b ((0x07));
            break;
        case 8:
            output_b ((0x7F));
            break;
        case 9:
            output_b ((0x6F));
            break;
        case 100: //Apagado
            output_b ((0x00));
            break;
        case 101: //Segmento G encendido
            output_b ((0x40));
            break;
    }

    salida=~salida;
    output_bit(PIN_A0, salida);
    output_bit(PIN_A1, ~salida);
}
```

ANEXO 6

DIAGRAMA ESQUEMÁTICO





FECHA DE ENTREGA

Ing. Víctor Proaño Rosero
COORDINADOR DE CARRERA

Sr. Juan Carlos Flores Paredes
ALUMNO