



Universidad del Azuay
Facultad de Ciencia y Tecnología
Escuela de Ingeniería Electrónica

**“Diseño de un Sistema Inalámbrico para controlar el tráfico vehicular
y peatonal mediante un Sistema SCADA con Semáforos Inteligentes”**

**Trabajo de Graduación previo a la obtención del título de
Ingeniero Electrónico**

Autores

Carlos Felipe Quinde Uyaguari

Byron Gonzalo Arévalo Bustos

Director

Ing. Leonel Pérez

Cuenca, Ecuador

2010

DEDICATORIA

Dedicamos el presente trabajo a nuestras familias, el cual, es una forma de retribuir en algo todo el esfuerzo y sacrificio que nos brindaron a lo largo de nuestras vidas, a nuestros amigos, que siempre estuvieron presentes con su apoyo y a todos los profesores que con su conocimiento siempre nos brindaron lo mejor de sí.

AGRADECIMIENTO

A nuestros padres, por todo su apoyo y amor brindado desinteresadamente, a nuestro director Ing. Leonel Pérez, profesor y amigo, cuyos consejos alentaron a realizar un buen trabajo y a todas las personas que estuvieron presentes con su amistad y confianza.

“Diseño de un Sistema Inalámbrico para controlar el tráfico vehicular y peatonal mediante un Sistema SCADA con Semáforos Inteligentes”

RESUMEN

Este trabajo pretende mejorar en forma rápida, segura y eficiente el tráfico vehicular y peatonal en una ciudad, utilizando nuevas tecnologías, mediante el diseño de un sistema de semaforización inteligente.

El diseño, está basado en tecnología de micro controladores y control inalámbrico “full dúplex” con un alcance de 800 metros de distancia, utilizando técnicas de instrumentación virtual implementados a través de un sistema S.C.A.D.A. (“Supervisory Control And Data Acquisition”), para visualizar el estado de los semáforos con sus respectivas alarmas, modificación de tiempos de encendido de lámparas y configuración de conteo vehicular para una base de datos.

ABSTRACT

This work is intended to improve in sure and efficient quick form and the vehicles traffic and pedestrian flow in any city, creating a new technology through a design of on intelligent semaforization system.

The designed system is based of microcontrollers technologies and wireless control full duplex with a reach of 800 distance meters, features via a S.C.A.D.A. (“Supervisory Control And Data Acquisition”), system implemented using virtual instrumentation techniques, to visualize the state of the traffic lights with their respective alarms, modification of times of ignition of lamps and configuration of vehicular counts for a database.

ÍNDICE DE CONTENIDOS

DEDICATORIA.....	ii
AGRADECIMIENTO.....	iii
RESUMEN.....	iv
ABSTRACT.....	v
ÍNDICE DE CONTENIDOS.....	vi
ÍNDICE DE CUADROS E ILUSTRACIONES.....	viii
ÍNDICE DE ANEXOS.....	x

INTRODUCCIÓN.....	1
-------------------	---

CAPÍTULO 1: TEORÍA Y CONCEPTOS BÁSICOS.

1.1 Semaforización.....	3
1.1.1 Sistema actual.....	3
1.1.2 Sistema inteligente.....	4
1.1.3 Selección de eventos mediante fechas establecidas en el calendario.....	5
1.2 Descripción del microcontrolador.....	6
1.2.1 Sistema maestro.....	7
1.2.2 Sistema esclavo.....	8
1.2.3 Reloj de tiempo real.....	8
1.3 Sensores.....	9
1.3.1 Sensores mecánicos.....	10
1.3.2 Aplicación del sensor mecánico al sistema.....	11
1. 4 Introducción al sistema SCADA.....	11
1.4.1 Componentes del sistema.....	11

1.4.1.1 Unidad de terminal remota.....	12
1.4.1.2 Estación maestra.....	12
1.4.1.3 Interfaz hombre máquina.....	13
1.4.1.4 Infraestructura de comunicación.....	14
1.4.1.5 Comunicación inalámbrica.....	14
1.4.2 Supervisión y monitorización.....	16
1.4.3 Áreas de aplicación.....	16

CAPÍTULO 2: DISEÑO Y CONSTRUCCIÓN DEL HARDWARE.

2.1 Introducción.....	18
2.2 Bloques principales del sistema.....	18
2.3 Circuito maestro.....	19
2.3.1 Contador de vehículos.....	21
2.3.2 Transmisión de datos.....	21
2.3.3 LCD.....	22
2.3.4 Contador descendente peatonal.....	23
2.4 Circuito esclavo.....	24
2.4.1 Sensores de luz.....	25
2.4.2 Circuito conmutador.....	26
2.5 Circuito RTC.....	27
2.6 Fuente de alimentación con batería auxiliar.....	28
2.7 Construcción del circuito impreso.....	30

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE.

3.1 Programación del micro controlador maestro.....	31
3.1.1 Programa principal del sistema maestro.....	31
3.1.2 Diagrama de flujo del programa principal.....	33
3.1.3 Interrupción general.....	34
3.1.4 Diagrama de flujo de interrupciones.....	35
3.2 Programación del circuito esclavo.....	36
3.2.1 Programa principal.....	36
3.3 Programación del circuito RTC.....	38
3.3.1 Diagrama de flujo del RTC.....	38
3.4 Paneles de control y programación gráfica en LabVIEW.....	39
3.4.1 Funcionamiento.....	39
3.4.2 Configuración del puerto serial.....	40
3.4.3 Programación en el diagrama de bloques.....	41
3.4.4 Construcción de semáforos en el panel de control.....	42
3.4.5 Modificación de variables.....	43

CAPÍTULO 4: PRUEBAS FUNCIONALES.

4.1 Equipo de funcionamiento.....	50
4.1.1 Placas físicas.....	52
4.1.2 Maqueta a escala.....	53
4.1.3 Panel de control.....	54

4.2 Récord anecdótico en el desarrollo del prototipo.....	55
4.3 Descripción y resultados de las pruebas realizadas.....	55
 CONCLUSIONES	61
BIBLIOGRAFÍA.....	62
ANEXOS.....	64

Índice de Cuadros e Ilustraciones

Tabla 1.1.3:	Fechas asignadas al micro controlador.....	16
Tabla 2.3:	Descripción de pines del micro controlador maestro.....	31
Tabla 2.4:	Descripción de pines del micro controlador esclavo.....	35
Tabla 3.2.1:	Tabla de verdad de semáforos con valores límites.....	41
Figura 1.2:	Diagrama de pines PIC16F887A.....	17
Figura 1.2.3.a:	Diagrama de pines reloj de tiempo real DS1307.....	19
Figura 1.2.3.b:	Diagrama de pines PIC16F628A.....	20
Figura 2.2:	Diagrama de bloques del sistema.....	30
Figura 2.3.a:	Diagrama del circuito maestro.....	31
Figura 2.3.1:	Circuito controlador de vehículos.....	32
Figura 2.3.2:	Circuito driver multicanal MAX232.....	33
Figura 2.3.3:	Circuito del LCD.....	33
Figura 2.2.1.3:	Circuito controlador peatonal.....	34
Figura 2.4.a:	Diagrama del circuito maestro.....	36
Figura 2.4.1:	Diagrama sensores de luz.....	37
Figura 2.4.2:	Diagrama circuito TRIACS.....	38
Figura 2.5:	Diagrama del circuito RTC.....	39
Figura 2.6:	Diagrama del circuito de fuente con batería auxiliar.....	40
Figura 2.7:	Diagrama del P.C.B. para el circuito maestro-esclavo- RTC.....	64

Figura 2.8:	Diagrama del P.C.B. para el circuito sensores de consumo de corriente.....	65
Figura 2.9:	Diagrama del P.C.B. para el circuito de TRIACS.....	66
Figura 2.10:	Diagrama del P.C.B. para el circuito de la fuente de alimentación.....	67
Figura 3.1.2:	Diagrama de flujo del programa principal.....	44
Figura 3.1.4.:	Diagrama de flujo de interrupciones.....	46
Figura 3.3.1:	Diagrama de flujo del circuito RTC.....	50
Figura 3.4.1.a:	Ventanas de acceso al SCADA.....	51
Figura 3.4.1.b:	Ingreso del código.....	51
Figura 3.4.2:	Diagrama de inicialización, lectura y escritura del puerto serial.....	53
Figura 3.4.3:	Diagrama para modificar la fecha y hora.....	54
Figura 3.4.4:	Visualización de semáforos.....	54
Figura 3.4.5:	Modificación de variables.....	55
Figura 3.4.5.1:	Testeo de ingreso.....	56
Figura 3.4.5.2:	Pregunta de ingreso.....	57
Figura 3.4.5.3:	Recepción de datos para las horas pico.....	57
Figura 3.4.5.4:	Modificación de tiempos de intersección de horas pico.....	58
Figura 3.4.5.5:	Valores modificados y selección de secuencia	59
Figura 3.4.5.6:	Envío de caracteres ASCII al micro controlador.....	60
Figura 4.1:	Equipo de funcionamiento.....	50
Figura 4.1.1:	Placa física del circuito maestro-esclavo.....	51
Figura 4.1.4:	Placa física del circuito de fuente simétrica (+-12V, +-5V).....	52

Figura 4.1.5:	Placa física para los sensores de luz.....	52
Figura 4.1.6:	Placa física del circuito de conmutación con TRIACS.....	53
Figura 4.1.2:	Maqueta a escala.....	54
Figura 4.1.3:	Panel de control.....	54

Índice de Anexos

CAPÍTULO 2.

Anexo A:	Diagrama del P.C.B. maestro-esclavo-RTC.....	64
Anexo B:	Diagrama del P.C.B. sensores de consumo de corriente.....	65
Anexo C:	Diagrama del P.C.B. de TRIACS.....	66
Anexo D:	Diagrama del P.C.B. de la fuente de alimentación	67

CAPÍTULO 3.

Anexo A:	Código del programa principal del micro controlador maestro.....	68
Anexo B:	Código de programa de atención a interrupciones generales del micro controlador maestro	70
Anexo C:	Código del programa principal del micro controlador esclavo.....	77
Anexo D:	Código de programa de atención a interrupciones generales del micro controlador esclavo.....	85
Anexo E:	Código de programa del RTC.....	87

Quinde Uyaguari Carlos Felipe

Arévalo Bustos Byron Gonzalo

Trabajo de Graduación

Ing. Leonel Pérez

Junio del 2010

**Diseño de un Sistema Inalámbrico para controlar el tráfico vehicular y peatonal
mediante un Sistema SCADA con Semáforos Inteligentes**

INTRODUCCIÓN

Los sistemas de semaforización son muy importantes para las ciudades de hoy en día, es por eso que de acuerdo a los conocimientos adquiridos proponemos diseñar un sistema moderno y sobre todo confiable que sea capaz de aplicarse a dichas ciudades con lo último en tecnología que disponemos, es decir; de tercera generación.

La utilización de sistemas de comunicación inalámbricos son cada vez más frecuentes, y son muchas las ventajas obtenidas. De tal manera, este diseño es adaptivo a cualquier tipo de intersección. El sistema SCADA (“Supervisory Control And Data Acquisition”), es de vital importancia para el desarrollo del proyecto, así como la programación de micro controladores, calibración de sensores, realización de las placas físicas, etc.

Mediante el lenguaje de programación gráfico generado en LAbVIEW se creó el sistema SCADA (“Supervisory Control And Data Acquisition”), que es el panel de control en donde se puede monitorear y controlar todos los procesos que sean requeridos. Este diseño consta también con una maqueta a escala, en la cual se pretende mostrar el funcionamiento completo de todo el sistema a demás de realizar las pruebas necesarias; de tal manera aportando al desarrollo de la investigación y creatividad.

CAPÍTULO 1

TEORÍA Y CONCEPTOS BÁSICOS

1.1 Semaforización.

Los semáforos constituyen una medida de solución efectiva y organizada cuando se trata de controlar el flujo de vehículos y personas en una ciudad, su instalación depende de varios factores: Entre los cuales se establece el número de vías, sentido de circulación, volumen de tráfico vehicular, interrupción de tráfico continuo, volumen de peatones, cruce de peatones escolares, frecuencia de accidentes, sistemas de semaforización, o una combinación de todos.

Para controlar el tráfico, peatonal y vehicular en una ciudad, sea grande o pequeña, a través de los años se ha desarrollado diversos métodos, los cuales avanzan de la mano con la tecnología. Es de vital importancia contar con un sistema eficiente que brinde seguridad a peatones y vehículos.

1.1.1 Sistema actual.

La capital azuaya dispone de semáforos de última tecnología. Cien son los sectores de Cuenca que los poseen. El 85% de los aparatos se encuentran en el centro de la ciudad y el resto en los sectores más conflictivos como la avenida Huayna Cápac, 12 de Abril, Héroes de Verdeloma, Unidad Nacional, Ordóñez Lazo, entre otros. Según un estudio de la Unidad de Tránsito del Municipio, en la provincia del Azuay, circulan alrededor de 70 mil

vehículos y solo por el centro de Cuenca pasan a diario 50 mil autos. Pero existen zonas, como la avenida Ordóñez Lazo, por donde se registran hasta 70 mil vehículos diariamente.

El nuevo sistema está basado con tecnología de Letz, es un diseño que abarca una técnica centralizada y computarizada de todo el ordenamiento vehicular, a demás los semáforos ya no serán programados, sino que de acuerdo al flujo vehicular se auto regulan, lo que permite controlar el paso de vehículos y peatones. Los equipos son adaptativos, significa que la semaforización se adapta a la realidad del tráfico, es completamente dinámico, la ventaja de este proyecto es que es inclusiva desde el punto de vista social, cuenta con indicadores acústicos para los invidentes y tiene una central semafórica computarizada.

1.1.2 Sistema inteligente.

El sistema de semaforización inteligente consta de una técnica centralizada para dar lugar al control SCADA (“Supervisory Control And Data Acquisition”), el cual monitorea el tráfico vehicular y peatonal, adquiere el número de vehículos que circula por la intersección y controla el flujo de tránsito para evitar el congestionamiento en la ciudad y dar seguridad a los peatones.

A demás funciona con una serie de eventos marcados en el calendario, la información la proporciona el microcontrolador e indica, como debe funcionar los semáforos esos días, también se puede agregar más eventos mediante el computador. Incluye un sistema de detección de errores, especificando que semáforo o sensor está fallando; posee un generador para realizar un historial vehicular en cualquiera de las intersecciones determinando un cierto horario, los datos se pueden guardar para aplicaciones futuras.

En caso de emergencia se podrá manejar manualmente el encendido de los semáforos, dando prioridad a la vía que la necesite. A su vez se puede ajustar la programación de los tiempos de encendido de las luces, según la conveniencia del tránsito, mediante un operador de control. La programación para los semáforos, sensores de conteo vehicular, y contadores peatonales se realiza en el microcontrolador; las intersecciones son controladas,

monitoreadas y programadas en un centro de control único a través de un sistema dedicado SCADA (“Supervisory Control And Data Acquisition”), el cual se describe más adelante.

1.1.3 Selección de eventos mediante fechas establecidas en el calendario.

Para mejorar el flujo de tránsito en días específicos se evaluó la variación de flujo vehicular y peatonal en fechas importantes que conciernen a la ciudad de Cuenca y otras que son declaradas feriados en la República del Ecuador como se muestra en la tabla 1.1.3.

FESTIVIDAD	DD/MM/AA
Año Nuevo	01/01/2010
Fundación de Cuenca	12/04/09
Día del Trabajo	01/05/09
Batalla de Pichincha	24/05/09
Independencia del Ecuador	10/08/09
Independencia de Cuenca	03/11/09
Navidad	25/12/09

Tabla 1.1.3: Fechas asignadas al microcontrolador.

Fuente: Autores.

1.2 Descripción del microcontrolador.

El PIC 16F877A como se muestra en la figura 1.2, pertenece a la familia de micro controladores elaborada por Microchip Technology, el cual posee características especiales en un solo integrado, como una arquitectura Harvard que trabaja con buses separados que comunican la Memoria de Programas y la Memoria de Datos, el acceso al mismo tiempo a estas memorias dando lugar a multitareas, codificación en una sola palabra, un set de

instrucciones simétricos para poder direccionar registros de propósito general y de propósito específico.

La arquitectura de la memoria de programas en el PIC 16F877A tiene 4 páginas de “2K palabras” siendo igual a 8192 palabras de 14 bits, el micro controlador cuenta con un vector de reset, un vector de interrupción, contador de programas para saber la dirección de la memoria de programas, una pila que es una memoria independiente para respaldar hasta 8 direcciones de retorno diferente; y con referencia a la memoria de datos, se encuentra dividida en 4 páginas llamadas bancos, donde se tienen registros de propósito general y específicos.

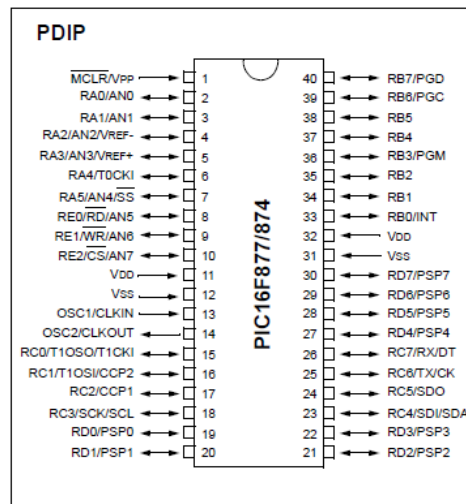


Figura 1.2: Diagrama de pines, PIC16F877A.

Fuente: Autores.

El sistema de control de semáforos está programado en el PIC 16F877A en lenguaje ensamblador mediante el software MPLAB 8.3 con 14 fuentes de interrupción y tres timers; cuenta con dos módulos CCP(“Capture Compare PWM”), comunicación serie USART(“Universal Synchronous Aynchronous Receiver Transmitter”), comunicación en paralelo mediante el protocolo PSP(“Parallel Slave Port”) y conversor A/D de 10bits.

1.2.1 Sistema maestro.

Es el corazón del sistema inteligente, en él se encuentran todas las referencias que el operador necesita, la información de datos está almacenada en la memoria del micro controlador, es decir; los tiempos de conmutación de los semáforos, reloj calendario, fecha /hora de inicio y fin del conteo de vehículos; a demás con las necesidades de la semaforización, el micro informa el fallo y cambio de estado del encendido de luces de los semáforos; consta de una entrada digital al pin (RB0), por la cual ingresa el pulso externo, generado por un reloj de tiempo real; el maestro consta con salidas para la transmisión de datos del LCD, mediante el puerto D, dos pines del puerto E para la conmutación del display de 7 segmentos, cuatro pines del puerto A para el contador descendente.

El operador puede comunicarse inalámbricamente por el puerto serial RS-232 del computador con el sistema maestro mediante el driver bidireccional multicanal MAX-232, conectado a dos pines específicos de comunicación del micro controlador maestro; para lograr la transmisión inalámbrica entre los puertos del computador y del maestro, se adiciona a los terminales de comunicación un módulo de radio frecuencia de alta velocidad con baja potencia (FC-221/AG), con un alcance de hasta 800m de distancia. Está comunicado con el esclavo a través del módulo MSSP (“Master Serial Síncrono Port”), que opera en modo SPI (“Serial Periferic Interfaz”), con tres pines del puerto C, para receptar la información del estado de encendido de los semáforos, mediante los sensores de luces. La descripción de los pines del maestro se encuentra detalladamente en la tabla 2.3 del capítulo 2.

1.2.2 Sistema esclavo.

El esclavo detecta la suspensión de energía eléctrica mediante una entrada del convertidor analógico/digital, y enviar una alarma al maestro y al computador en ese orden; recibe datos en forma binaria del maestro mediante el puerto B, que encienden las luces de los semáforos a través del puerto D, 5 pines del puerto C y uno del puerto B; interpreta el consumo de corriente de las lámparas por medio de los pines convertidores del puerto A y

E, en caso de que dicho voltaje no coincida con el de la secuencia, el esclavo envía banderas de error al maestro mediante los tres pines del puerto C configurados en modo SPI (“Serial Periferic Interfaz”), y el maestro envía al operador la misma bandera de error a la PC, la asignación de los pines del esclavo se indica en la tabla 2.4 del capítulo 2.

1.2.3 Reloj de tiempo real.

La aplicación del reloj de tiempo real DS1307 de Dallas Semiconductor se realiza utilizando un microcontrolador PIC16F628A. El reloj es un dispositivo, en donde se quiere dejar constancia de la fecha y hora de un evento, cuya interfaz se realiza utilizando el bus I2C (“Inter-Integrated Circuit”). Para la aplicación del sistema inteligente, el micro controlador maestro toma solamente la generación de pulso del DS1307. A continuación en las figuras 1.2.3.a y 1.2.3.b, se muestra la asignación de pines del DS1307 y del micro controlador 16F628A.

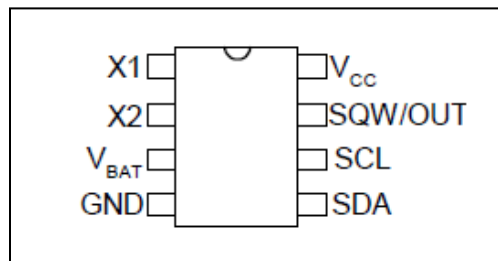


Figura 1.2.3.a: Diagrama de pines del DS1307.

Fuente: Autores.

- VCC: Alimentación de 5 voltios DC
- GND: Referencia a tierra GND.
- VBAT: Entrada de alimentación de una pila de 3 voltios.
- SCL: Entrada de reloj para sincronizar la transferencia de datos en la interfaz.
- SDA: Entrada/salida de datos para la interfaz I2C.
- SQW/OUT: Salida generadora del pulso de 1 segundo receptada por el maestro.
- X1, X2 Conexión al cristal de cuarzo estándar 32.768 Hz.

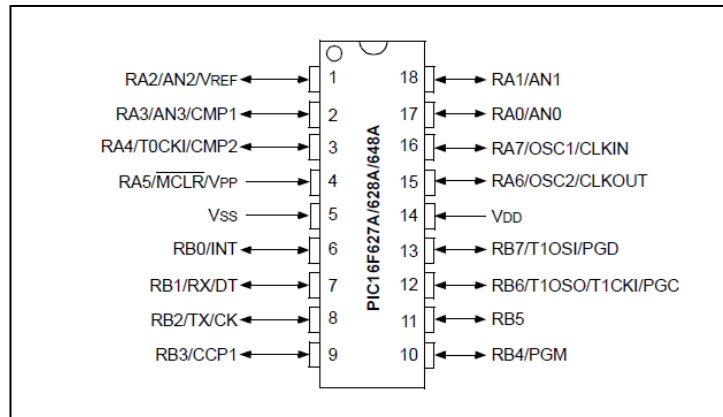


Figura 1.2.3.b: Diagrama de pines, PIC16F628A.

Fuente: Autores.

- VDD: Alimentación de 5 voltios DC
- VSS: Referencia a tierra GND.
- RA7/OSC1, RA6/OSC2/: Conexiones al cristal de 16 MHz.
- RB6/T1OSO: Entrada/salida de datos para la interfaz I2C.
- RB7/T1OSCI: Salida de reloj para sincronizar la transferencia de datos en la interfaz.
- RA5/MCLR: Reset externo.

1.3 Sensores.

Para realizar las mediciones de magnitudes mecánicas, térmicas, eléctricas, físicas, químicas, entre otras, se emplean dispositivos llamados sensores y transductores.

El sensor percibe los cambios de la magnitud en cuestión, como temperatura, posición o concentración química, mientras que el transductor convierte estas mediciones en señales generalmente eléctricas para suministrar la información a elementos de lectura y registro o para el control de las magnitudes medidas.

El Término Sensor (Transductor) se aplica a dispositivos o combinaciones de dispositivos que convierten señales o energía de una forma física en otra. Como sucede en los sistemas de medición donde un transductor proporciona una salida útil en respuesta a una variable específica. La variable es una cantidad, propiedad o condición física que se mide y la salida es la cantidad eléctrica, producida por el sensor, que es función de la variable medida específica.

El dispositivo de entrada recibe la cantidad bajo medición y entrega una forma de onda eléctrica proporcional a él, esto significa que suministran una señal de salida cuando son estimulados por una entrada no mecánica, es decir; un termistor por ejemplo reacciona a las variaciones de la temperatura, una fotocélula a los cambios de la intensidad luminosa, etc.

1.3.1 Sensores mecánicos.

Es un dispositivo que consta de un accionador unido mecánicamente a una serie de contactos. Cuando un objeto entra en contacto físico con el accionador, el dispositivo activa (o acciona) los contactos (o interruptores) para establecer o interrumpir una conexión eléctrica.

Permiten detectar presencia, ausencia, paso, posición; y son muy utilizados por su sencillez de operación, resistencia, robustez, facilidad de instalación, confiabilidad y de apertura positiva de los contactos con gran resistencia a los diversos entornos industriales y buena fidelidad de hasta 0,01 mm en los puntos de activación.

1.3.2 Aplicación del sensor mecánico al sistema.

La función del sensor es la de contar e indicar al operador del SCADA (“Supervisory Control And Data Acquisition”), la hora y cantidad de vehículos que transitan por cada intersección, siendo el maestro el que recepta la señal digital del sensor para enviar un código al computador y que el operador guarde esta información formando así un historial

de tráfico de la intersección asignada, con el fin de poder actualizar el tiempo de conmutación de los semáforos en base al historial registrado, obteniendo así una mejor fluidez vehicular.

1.4 Introducción al sistema SCADA.

SCADA: Supervisory Control And Data Acquisition (adquisición de datos y control de supervisión). Es una aplicación software especialmente diseñada para funcionar sobre ordenadores en el control de producción, proporcionando comunicación con los dispositivos de campo (controladores autónomos...) y controlando el proceso de forma automática desde la pantalla del ordenador.

También provee de toda la información que se genera en el proceso productivo a diversos usuarios, tanto del mismo nivel como de otros supervisores dentro de la empresa (supervisión, control de calidad, control de producción, almacenamiento de datos, etc.).

1.4.1 Componentes del sistema.

Los tres componentes de un sistema SCADA son:

Múltiples unidades de terminal remota, también conocida como RTU ("Remote Terminal Unit") o Estaciones Externas, estación maestra y computador con HMI ("Human Machine Interface"), y por último infraestructura de comunicación.

1.4.1.1 Unidad de terminal remota.

La solución de SCADA ("Supervisory Control And Data Acquisition") a menudo tiene componentes de sistemas de control distribuido, DCS ("Distributed Control System"). El uso de RTUs ("Remote Terminal Unit") o PLCs ("Programmable Logic Control"), sin involucrar computadoras maestras está aumentando, los cuales son autónomos ejecutando procesos de lógica simple.

La complejidad y la naturaleza de este tipo de programación hace que los programadores necesiten cierta especialización y conocimiento sobre los actuadores que van a programar; aunque la programación de los elementos es ligeramente distinta a la programación tradicional, también se usan lenguajes que establecen procedimientos. Esto les permite a los ingenieros de sistemas SCADA implementar programas para ser ejecutados en RTUs o un PLCs.

A diferencia de los sistemas de control distribuido, el lazo de control es generalmente cerrado por el operador. Los sistemas de control distribuido se caracterizan por realizar las acciones de control en forma automática, hoy en día es fácil hallar un sistema SCADA realizando labores de control automático en cualquiera de sus niveles, aunque su labor principal sea de supervisión y control por parte del operador.

1.4.1.2 Estación maestra.

El término "Estación maestra" se refiere a los servidores y al software responsable para comunicarse con el equipo del campo (RTUs, PLCs, etc); aquí se encuentra el software interfaz hombre máquina corriendo para las estaciones de trabajo en el cuarto de control, o en cualquier otro lado.

En un sistema SCADA pequeño, la estación maestra puede estar en un solo computador, a gran escala; puede incluir muchos servidores, aplicaciones de software distribuido, y sitios de recuperación de desastres. El SCADA (“Supervisory Control And Data Acquisition”), usualmente presenta la información al personal operativo de manera gráfica. Esto significa que el operador puede ver un esquema que representa la planta que está siendo controlada y procesada.

1.4.1.3 Interfaz hombre máquina.

Una interfaz hombre - máquina o HMI ("Human Machine Interface") es el aparato que presenta los datos a un operador (humano) y a través del cual éste controla el proceso.

La industria de la interfaz hombre máquina nació esencialmente de la necesidad de estandarizar la manera de monitorear y de controlar múltiples sistemas remotos, PLCs y otros mecanismos de control. Aunque un PLC realiza automáticamente un control pre-programado sobre un proceso, normalmente se distribuyen a lo largo de toda la planta, haciendo difícil recoger los datos de manera manual. Los sistemas SCADA lo hacen de manera automática, históricamente los PLCs no tienen una manera estándar de presentar la información al operador.

La obtención de los datos por el sistema SCADA parte desde el PLC o desde otros controladores y se realiza por medio de algún tipo de red, posteriormente ésta información es combinada y formateada. Un interfaz hombre máquina puede tener también vínculos con una base de datos para proporcionar las tendencias, los datos de diagnostico y manejo de la información.

También un cronograma de procedimientos de mantenimiento, información logística, esquemas detallados para un sensor o máquina en particular. SCADA está presente en aplicaciones sencillas, como controladores de temperatura en un espacio, hasta aplicaciones muy complejas como el control de plantas nucleares y control de tráfico vehicular.

1.4.1.4 Infraestructura de comunicación.

La comunicación por RS-232 ha sido la forma más común de comunicación de un sistema SCADA con un computador. Las primeras implantaciones generaban mensajes bajo un código numérico o nemónico para solicitar que el PLC ejecute una acción o proporcione información, debido a que básicamente los PLCs se programaban de esa forma.

Posteriormente, se cuentan con implantaciones en las que su programación se realizaba con lenguajes de tercera generación (Basic, C, etc.). Las aplicaciones cliente - servidor se conforman por un cliente delegado que recibe las solicitudes del usuario y despliega la información hacia el servidor.

1.4.1.5 Comunicación inalámbrica RS-232.

El interfaz RS-232 tiene como objetivo comunicar dos equipos mediante la comunicación serial, con un rango de voltajes entre +3/+15 voltios que representa un cero lógico y -3/-15 voltios representando a un uno lógico. Para conectar la unidad remota y el software del SCADA (“Supervisory Control And Data Acquisition”), en este caso el micro controlador maestro y la PC respectivamente se utiliza la conexión de tipo Null Modem como se muestra en la figura 1.4.1.5.a, y que consta de tres cables descritos en la figura 1.4.1.5.b, con unión de los pines RTS y CTS en ambos terminales para poder garantizar la misma velocidad entre ellos y sin necesidad de un control de flujo.

La comunicación inalámbrica entre la PC y la unidad remota (placa física), se logra al incluir un módulo de radio frecuencia de banda corta con alta velocidad de datos llamada FC-221/AG, ideal para comunicaciones asíncronas inalámbricas, conveniente para tres interfaces TTL/UART, RS-232 y RS-485. El módulo puede trabajar en bandas de frecuencias de 433 MHz, 868 MHz o 915 MHz, basado en la modulación 3FSK, con alta eficiencia y buena sensibilidad con un alcance de 800 metros, teniendo la opción de utilizar hasta 64 canales con una radio frecuencia de datos de 1200bps hasta 19200 bps. El esquema de interfaz de datos del FC-221/AG se muestra en la figura 1.4.1.5.c

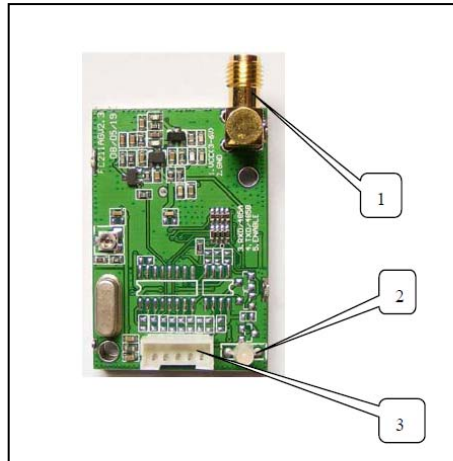


Figura 1.4.1.5.c: Módulo de radiofrecuencia FC-221/AG.

Fuente: Autores.

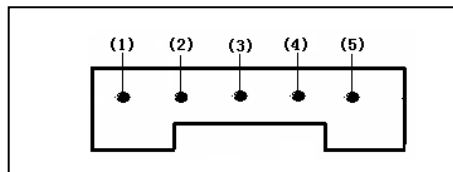


Figura 1.4.1.5.d: Ubicación de pines del módulo.

Fuente: Autores.

En el diagrama 1.4.1.5.d, que hace referencia al bloque 3 de la figura 1.4.1.5.c, muestra los pines de conexión del módulo el cual se describe a continuación:

- **Pin 1:** 4.6v-5.3V (Típicamente 5V).
- **Pin 2:** GND
- **Pin 3:** RXD: Receptor de datos serial.
- **Pin4:** TXD: Transmisor de datos serial.
- **Pin5:** EN: Selección de modo sleep, el módulo trabaja en bajo consumo cuando está en nivel lógico bajo, y en modo normal cuando el nivel lógico es alto.

1.4.2 Supervisión y monitorización.

Mediante tarjetas inteligentes ubicadas en el bus de los PLCs (década de los 80), podían controlar de forma autónoma un monitor y un teclado, intercambiaban información con el CPU del PLC por el bus de datos, ofrecían herramientas “sencillas” de configuración y programación, las prestaciones muy variadas (pantallas alfanuméricas, gráficas, etc.)

Actualmente – Interfaces hombre máquina.

- Interfaces de Operador

De fácil programación, robustos, funciones de comunicación con el dispositivo de control.

- Mediante SCADA.

SCADA (“Supervisory Control And Data Acquisition”), aplicación software diseñada para funcionar en ordenadores de control de producción, proporcionando comunicación con los dispositivos de control supervisando el proceso desde la pantalla del ordenador.

1.4.3 Áreas de aplicación.

1. **Monitorizar procesos.** Químicos, físicos o de transporte en sistemas de suministro de agua, para controlar la generación y distribución de energía eléctrica, de gas o en oleoductos, procesos de distribución y de semaforización.
2. **Gestión de cualquier tipo de procesos.** Facilita la programación de la fabricación.
3. **Mantenimiento.** Proporciona magnitudes de interés tales para evaluar y determinar, índices de control y estado, entre otros.
4. **Control de calidad y estadísticas.** Proporciona de manera automatizada los datos necesarios para calcular índices de estabilidad de la producción y el conteo de número de vehículos para obtener una tabla de datos que proporcione la información de tráfico vehicular del sistema de semaforización.

5. **Administración.** Actualmente pueden enlazarse estos datos del SCADA con un servidor o sistema de planificación de recursos empresariales, e integrarse como un módulo más. Tratamiento histórico de información.

CAPÍTULO 2

DISEÑO Y CONSTRUCCIÓN DEL HARDWARE

2.1 Introducción.

Para el diseño y construcción del hardware (maestro – esclavo), actuadores, sensores y fuente de alimentación, se utiliza el programa Altium Designer Winter 09, que posee herramientas eficientes para realizar la construcción, simulación de circuitos electrónicos y tarjetas de circuito impreso, el desarrollo de las partes físicas de cada uno de los elementos que conforman el hardware del sistema se describen a continuación.

2.2 Bloques principales del sistema.

1.Maestro. Formado por sensores mecánicos, pantalla de visualización LCD, decodificador y display de 7 segmentos, driver multicanal RS232, reloj de tiempo real, fuente de alimentación y esclavo.

2. Esclavo. Integrado por actuadores (TRIACS), sensores de luz y maestro.

3. RTC. Conformado por un micro controlador PIC16F628A, reloj de tiempo real (DS1307) y conectado al maestro.

4. Fuente con batería auxiliar. Conectado a todos los elementos que conforman la tarjeta de control.

La figura 2.2 muestra los bloques principales de la tarjeta controladora.

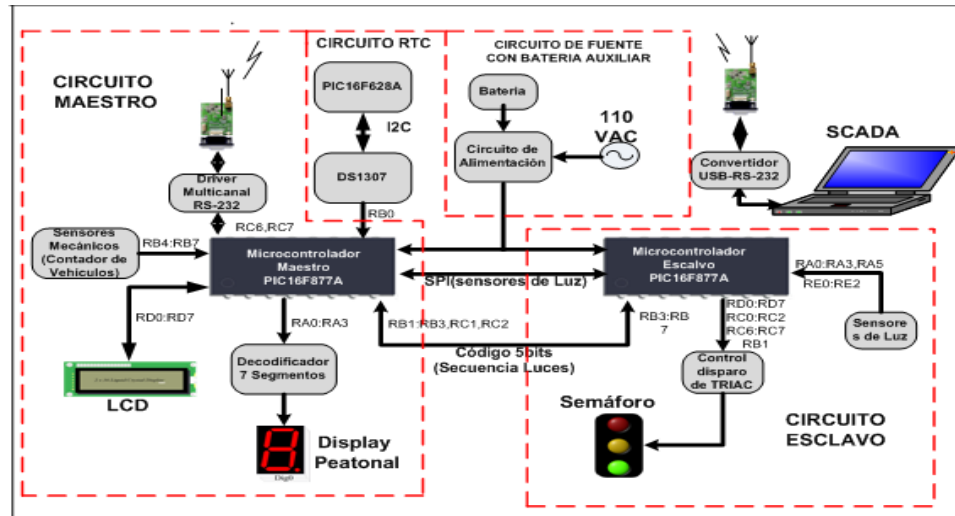


Figura 2.2: Diagrama de bloques del sistema.

Fuente: Autores.

2.3 Circuito maestro.

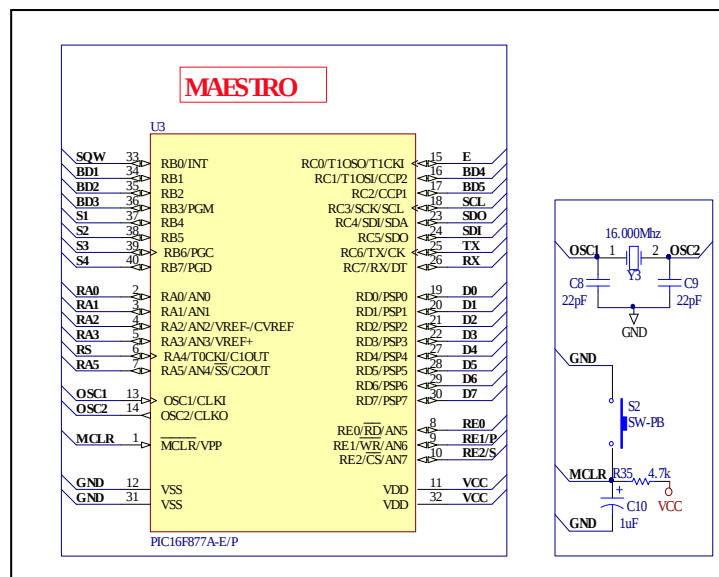
La distribución de pines en el micro controlador maestro se indica en la tabla 2.3. Los pines 2, 3, 4 y 5 están conectados a los pines 7, 1, 2 y 6 de los decodificadores U5 y U6, los pines 6, 15, 19, 20, 21, 22, 27, 28, 29 y 30 están unidos al conector I.D.C. (P5) correspondiente a la entrada del LCD, el pin 7 tiene conexión a una resistencia de base R31 para habilitar el back Light del LCD, los pines 13 y 14 se conectan al cristal Y3, el pin 1 está conectado con el circuito integrado por el pulsante S2, la resistencia R35 y el condensador C10, el pin 33 está conectado con el pin 7 del DS1307(U7), configurado como pull-up con la resistencia R18, los pines 34, 35, 36, 16 y 17 forman un bus de datos conectado al esclavo mediante los pines 36 al 40, los pines 37, 38, 39 y 40 se conectan a las resistencias R21, R22, R23 y R24 del conector I.D.C. (P4), los pines 18, 23 y 24 se conectan a los pines 18, 24 y 23 del micro controlador esclavo respectivamente.

		Microcontrolador (master)	
Puerto A	RA0: Entrada analógica, registro ADCON1, detecta corte de energía eléctrica .	RA4: Salida digital, reloj de sincronización de datos para el lcd.	RA5: Salida digital, enciende el back light del lcd.
Puerto B	RB0: Entrada digital, registro TRISB, pulso externo del reloj.	RB1-RB3: Salidas digitales, registro TRTISB, bus de datos para transmisión de secuencias de los semáforos.	RBA-RB7: Entradas digitales, registro TRISB. RB4 y RB5 cuentan el número de vehículos en la avenida principal. RB6 y RB7 en la secundaria.
Puerto C	RC0: Salida digital, registro TRISC habilitado.	RC1-RC2: Salidas digitales, registro TRISC, transmisión de secuencias de los semáforos.	RC3: Salida digital, registro TRISC, maestro en modo SPI para sincronización con el esclavo.
	RC4: Entrada digital, registro TRISC, ingreso del código de luces de fallo en modo SPI.	RC5: No se ocupa	RC6: Salida digital, registro TXSTA, transición de datos del módulo USART.
	RC7: Entrada digital, registro RCSTA, Recepción asíncrona de datos del módulo USART.		
Puerto D	RD0-RD7: Salidas digitales, bus de datos del lcd.		
Puerto E	RE0: No se ocupa.	RE1-RE2: Salidas digitales, registro ADCON1, pulso para el decodificador bcd de 7 segmentos.	

Tabla 2.3: Descripción de pines del microcontrolador maestro.

Fuente: Autores.

Los pines 25 y 26 están conectados a los pines 11 y 12 del MAX232, los pines 9 y 10 son parte del circuito conmutador de displays conectados a la resistencias R25 y R32, los pines 11, 12, 31 y 32 se unen al conector I.D.C. (P7) que es entrada de la fuente con batería auxiliar, como se muestra en la figura 2.3.a.



2.3.1 Contador de vehículos.

La figura 2.3.1 muestra el circuito contador de vehículos perteneciente al sistema maestro; los pines 37, 38, 39 y 40 a través de las resistencias R21, R22, R23 y R24 están conectados a los pines 37 al 40 del microcontrolador maestro U3 configurados como pull-ups internos y a los pulsantes S3, S4, S5 y S6 que van conectados a tierra (GND).

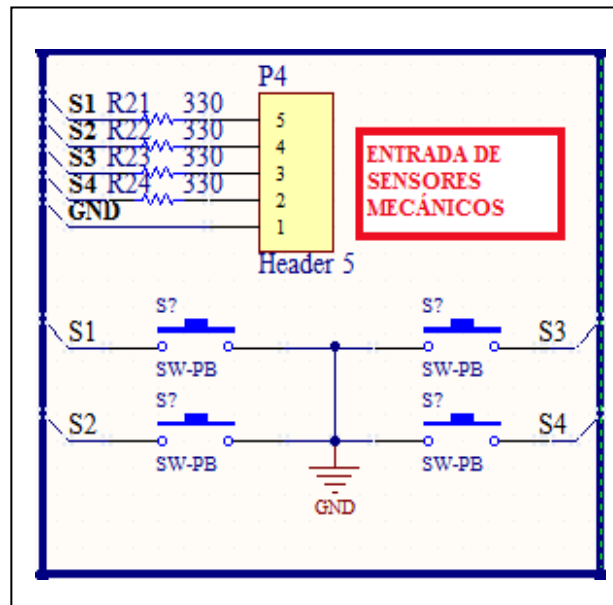


Figura 2.3.1: Circuito contador de vehículos.

Fuente: Autores.

2.3.2 Transmisión de datos.

La comunicación inalámbrica se realiza a partir de la conexión de los pines 25 y 26 del micro controlador maestro al driver multicanal MAX232, los pines del driver 1 y 3 están conectados al condensador C6, los pines 4 y 5 al C4, el pin 2 al C5, el pin 6 al C1, el pin 16 a VCC el pin 15 a GND, y los pines 13 y 14 al conector I.D.C. (P1). El comunicador inalámbrico de alta velocidad FC-221/AG de la compañía Friendcom, se conecta con

igual distribución de pines al mismo conector I.D.C, teniendo en cuenta que los pines RXD y TXD están intercambiados, como se muestra en la figura 2.3.2.

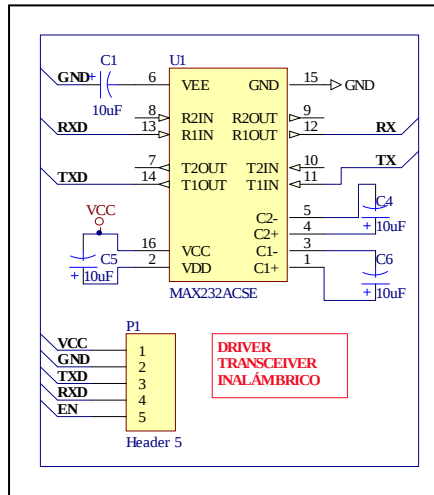


Figura 2.3.2: Circuito driver multicanal max232.

Fuente: Autores.

2.3.3 LCD (Liquid Crystal Display).

La figura 2.3.3 indica los pines 19, 20, 21, 22, 27, 28, 29, 30 del micro controlador maestro que se conectan al conector I.D.C. (P5), el pin conectado a la resistencia de base R31 del transistor 2N3904 (Q4) que sirve para el encendido

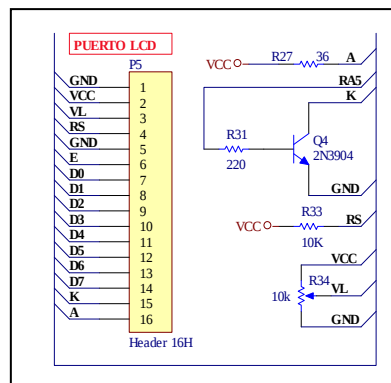


Figura 2.3.3: Circuito del LCD.

Fuente: Autores.

del back light; y el potenciómetro R34 para cambiar el contraste de la pantalla del LCD mediante el pin VL del conector I.D.C.

2.3.4 Contador descendente peatonal.

El contador para los displays en los semáforos peatonales está conectado a partir de los pines 2, 3, 4 y 5 de U3 a los pines, 7, 1, 2 y 6, de los decodificadores SN74LS47 (U5 y U6) respectivamente. Los pines de salida digital 13, 12, 11, 10, 9, 15 y 14 de U5 y U6 se conectan a las resistencias R25 y R32 en ese orden, la salida de R25 está conectada a los conectores I.D.C. (P10, P14 y P11, P15) de los semáforos principales y la salida de R32 conectada a los conectores I.D.C. (P8, P12 y P9, P13) de los semáforos secundarios, como indica la figura 2.3.4.

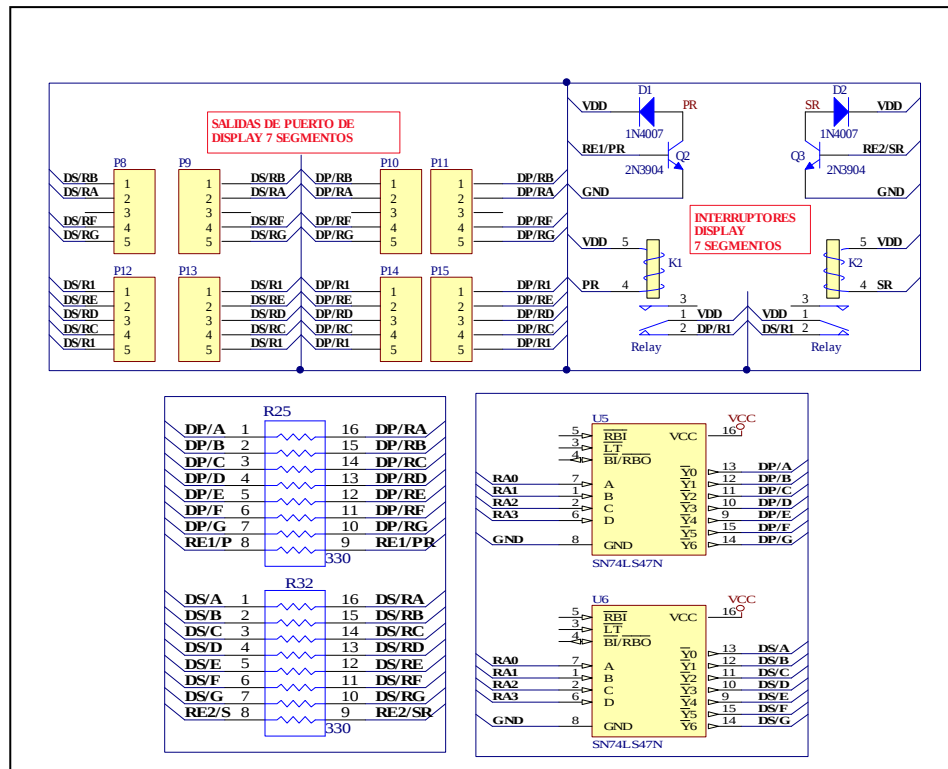


Figura 2.2.1.3: Circuito contador peatonal.

Fuente: Autores.

Los pines 9 y 10 de U3 se conectan a las resistencias de base R25 y R22 de los transistores Q2 y Q3, los colectores están conectados a los relés K1, K2 y a los diodos D1, D2, conformando el circuito conmutador de displays.

2.4 Circuito esclavo.

La distribución de pines en el micro controlador maestro se indica en la tabla 2.4. El conector I.D.C. (P2), pertenece a las entradas de los sensores de luz, las cuales están conectadas a los pines 2, 3, 4, 5, 7, 8 y 9, del micro controlador U4. Los pines 36 al 40 de U4 pertenecen al bus de datos que comunica con el maestro, los pines 19 al 22, 27 al 30, 15 al 17, 25, 26 y 34, son dedicados al circuito de conmutación de lámparas que van al conector I.D.C. (P3). Los pines 13 y 14 están conectados al cristal Y1 y a los condensadores C2 y C3. El pin 1 se encuentra conectado a la resistencia R15, al pulsante S1 y al condensador C7.

	Microcontrolador (esclavo)		
Puerto A	RA0-RA5: Entradas analógicas, registro ADCON1, sistema de fallo de luz.		
Puerto B	RB0-RB2: No se ocupa.	RB3-RB7: Entradas digitales, registro TRISB, recepta los datos de secuencia de los semáforos.	
Puerto C	RC1-RC2-RC4-RC6-RC7: Salidas digitales, registro TRISC, encendido y apagado de luces en los semáforos.	RC3: Entrada digital, registro TRISC, recepta el reloj que genera el maestro en modo SPI.	RC5: Salida digital, registro TRISC, envía código de luces de fallo en modo SPI.
Puerto D	RD0-RD7: Salidas digitales, registro TRISD, para el semáforo.		
Puerto E	RE0-RE3: Entradas digitales, registro ADCON1, sistema de fallo de luz.		

Tabla 2.4: Descripción de pines del microcontrolador esclavo.

Fuente: Autores.

Los pines 12 y 31 están conectados a tierra (GND), y los pines 11 y 32 conectados a 5 voltios(VCC), los pines 18, 23 y 24, forman parte del bus de datos en modo SPI (“Serial Periferic Interfaz”), que se conectan al micro controlador maestro, como se muestra en la figura 2.4.a.

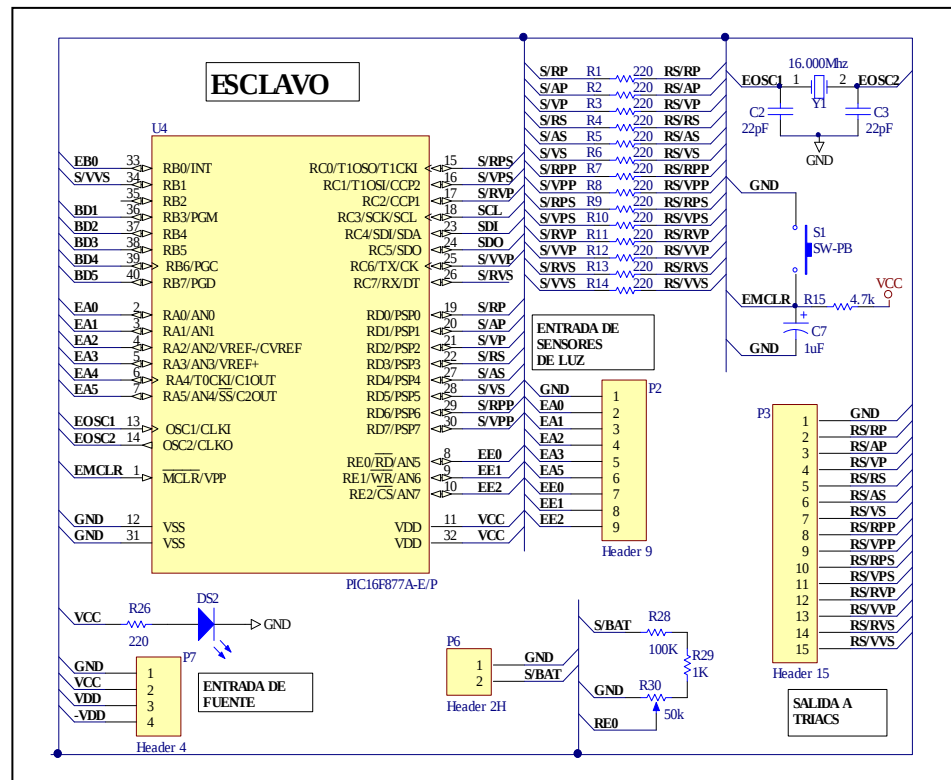


Figura 2.4.a: Diagrama del circuito maestro.

Fuente: Autores.

2.4.1 Sensores de consumo de corriente.

El circuito convierte el voltaje AC de cada lámpara en voltaje DC, reduciendo 110 voltios (AC) a un rango de (0,9 a 1,5V) DC, mediante un diodo rectificador y un condensador que permite filtrar el voltaje, obteniendo un voltaje que depende de la variación del encendido de luz del semáforo.

El operacional TL084 cumple la función de sumar los voltajes reducidos y rectificadas de dos lámparas del mismo color sean principales o secundarios, a la salida del operacional se obtiene un voltaje negativo, el cual se invierte con un circuito inversor, la salida del circuito está conectada a la entrada analógica del micro controlador esclavo mediante el conector I.D.C. (P3), como se muestra en la figura 2.4.1.

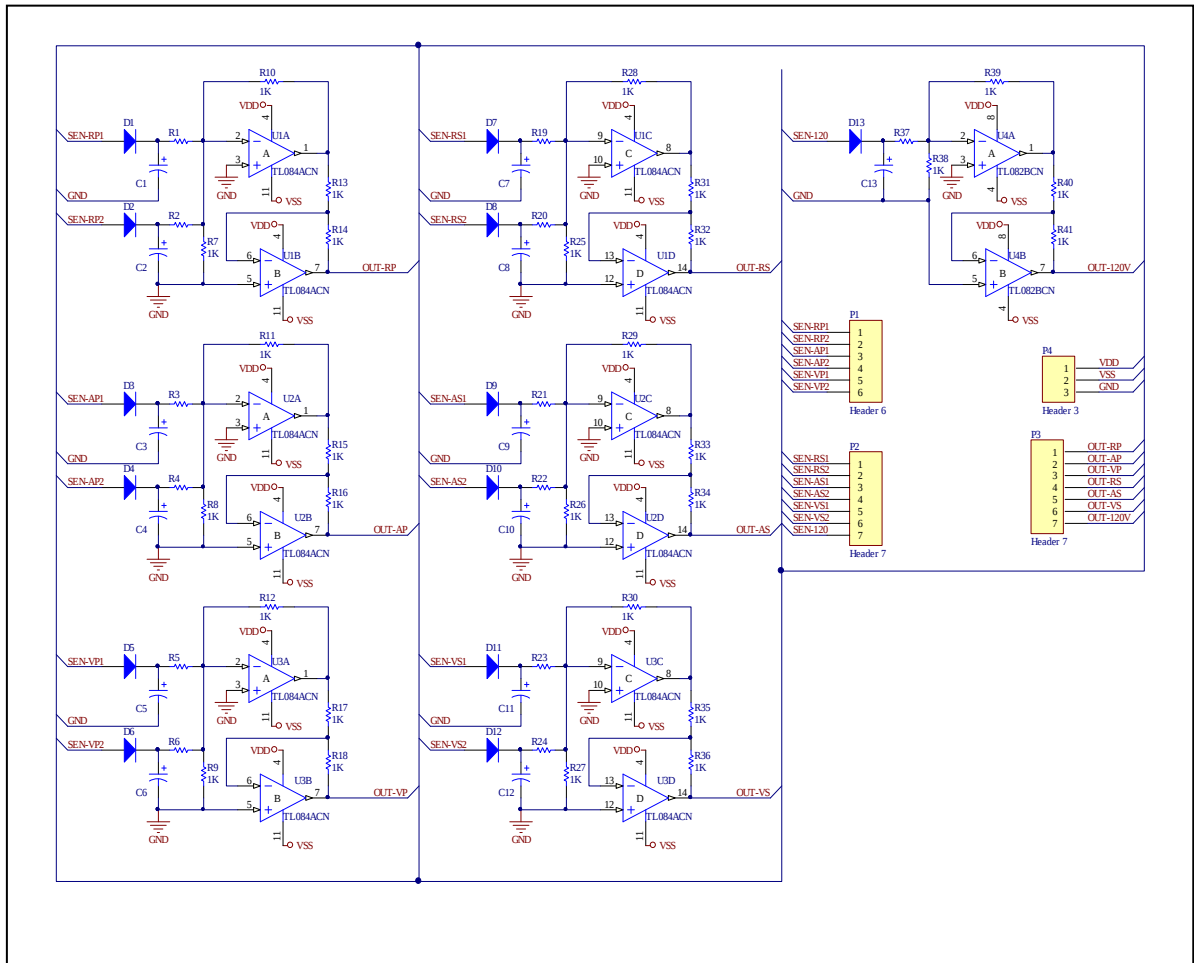


Figura 2.4.1: Diagrama sensores de luz.

Fuente: Autores.

2.4.2 Circuito conmutador.

Para encender las luces de los semáforos se utiliza un circuito conmutador integrado por TRIACS BT136 (Q1,...,Q14), resistencias (R1,...,R14), condensadores electrolíticos (C1,...,C14) y diodos 1N4007 (D1,...,D14), para evitar corrientes contrarias y proteger las salidas del micro controlador esclavo. El conector I.D.C. (P1), pertenece a la salida digital del esclavo que se conectan al circuito, como se muestra en la figura 2.4.2.

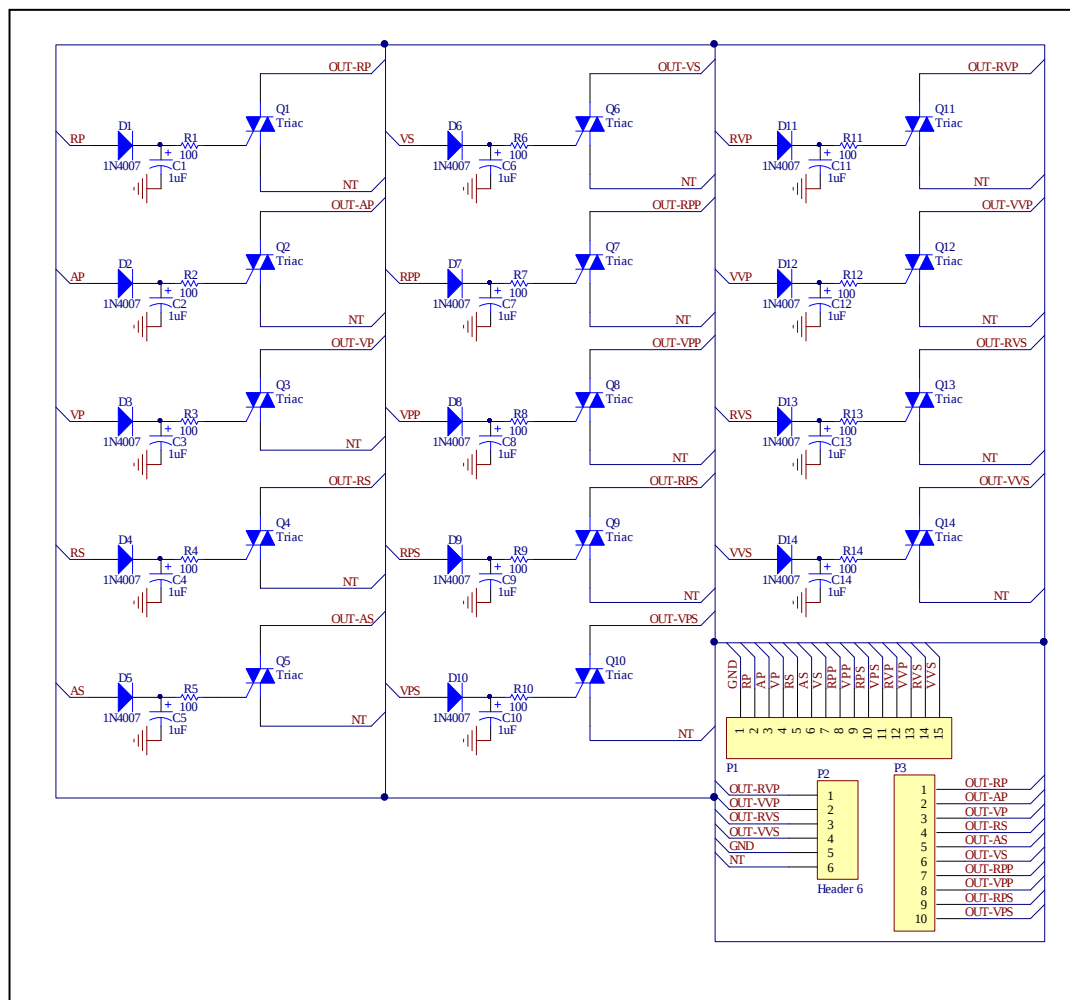


Figura 2.4.2: Diagrama circuito TRIACS.

Fuente: Autores.

2.5 Circuito RTC.

La figura 2.5 muestra el circuito RTC (“ReaL Time Clock”), el cual funciona con un microcontrolador PIC16F628A (U2) cuyos pines 12 y 13, están conectados a los pines 5, y 6 del DS1307 (U7) respectivamente. Para la alimentación de U2 se conecta el pin 5 a tierra (GND) y el pin 14 a 5 voltios (VCC), el pin 4 a la resistencia R16. Los pines 1 y 2 de U7 están conectados al cristal Y2, el pin 7 a la resistencia R18 (pull-ups) y la salida a la resistencia de base R20 perteneciente al transistor Q1, el colector al LED DS1 con una

resistencia de colector R19, el cual visualiza el pulso; el pin 5 se conecta a la resistencia R17 (pull –ups), el pin 3 a la batería del U7 y el pin 4 a GND, y para la alimentación se conecta el pin 8 a 5voltios (VCC).

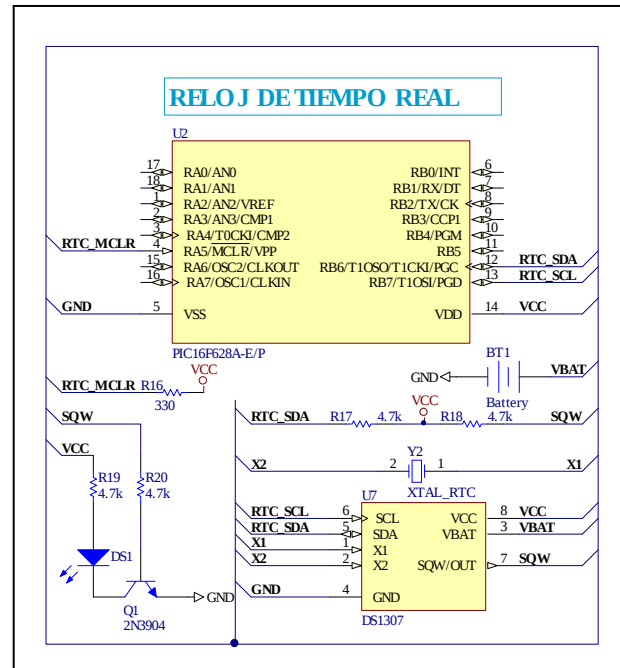


Figura 2.5: Diagrama del circuito RTC.

Fuente: Autores.

2.6 Fuente de alimentación con batería auxiliar.

El circuito maestro - esclavo dispone de una fuente para que, en el momento que exista un corte de energía eléctrica, los datos no se pierdan, por lo tanto se diseñó una fuente simétrica con batería auxiliar que funciona de la siguiente manera.

El integrado 555 mediante el pin 3 genera una señal oscilante con forma de onda cuadrada de 0 a 12V, cuando el pin se encuentra en un nivel lógico bajo, el condensador C13 se carga a través de D2 y D4, hasta que la tensión sea próxima a 12V. Si el pin 3 está en nivel lógico alto, la tensión en el punto de unión de C13 y D4 pasará a un valor dos veces más grande, el diodo D4 está polarizado inversamente, por lo que se bloquea

cuando se encuentre en este estado de conducción. El condensador C15 se carga con un valor de tensión superior a 12V, a la salida de C15 se conecta un regulador de tensión, a partir de un transistor NPN (Q1) y un diodo zener (D9) como referencia, el potenciómetro (R6) regula el voltaje de salida, como indica la figura 2.6.

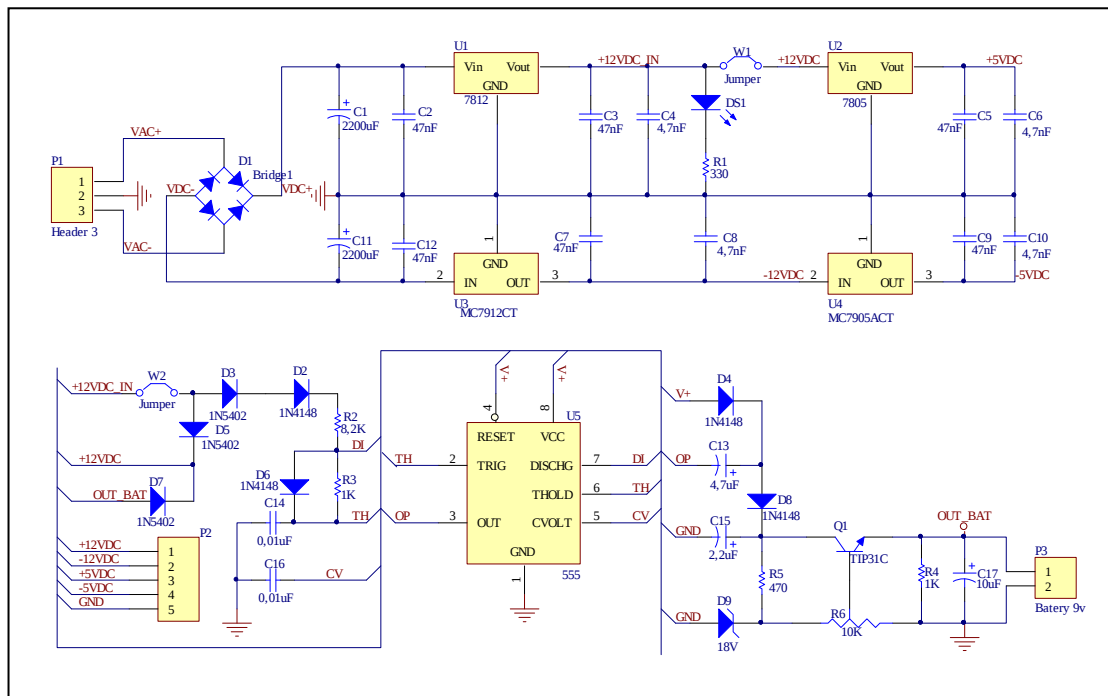


Figura 2.6: Diagrama del circuito de fuente con batería auxiliar.

Fuente: Autores.

La fuente es importante porque el sistema debe estar siempre en funcionamiento ya que en él se encuentran los datos del reloj calendario, el tiempo de encendido de luces, fecha y hora de activación y desactivación de las secuencias del encendido de luces y de los sensores mecánicos dedicados al conteo vehicular.

2.7 Construcción del circuito impreso.

Los circuitos mencionados tienen como objeto formar parte del funcionamiento físico en el sistema inteligente, para lo cual se construyen los circuitos impresos P.C.B. (“Printed Circuit Board”) de la siguiente manera.

1. Maestro – esclavo – RTC.
2. Conmutador de semáforos.
3. Sensores de luz.
4. Fuente de alimentación.

Los circuitos impresos se observan en los anexos A, B, C y D del capítulo 2.

CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE

3.1 Programación del microcontrolador maestro.

3.1.1 Programa Principal del sistema maestro.

El programa del circuito maestro está escrito en lenguaje ensamblador, compuesto de la siguiente estructura:

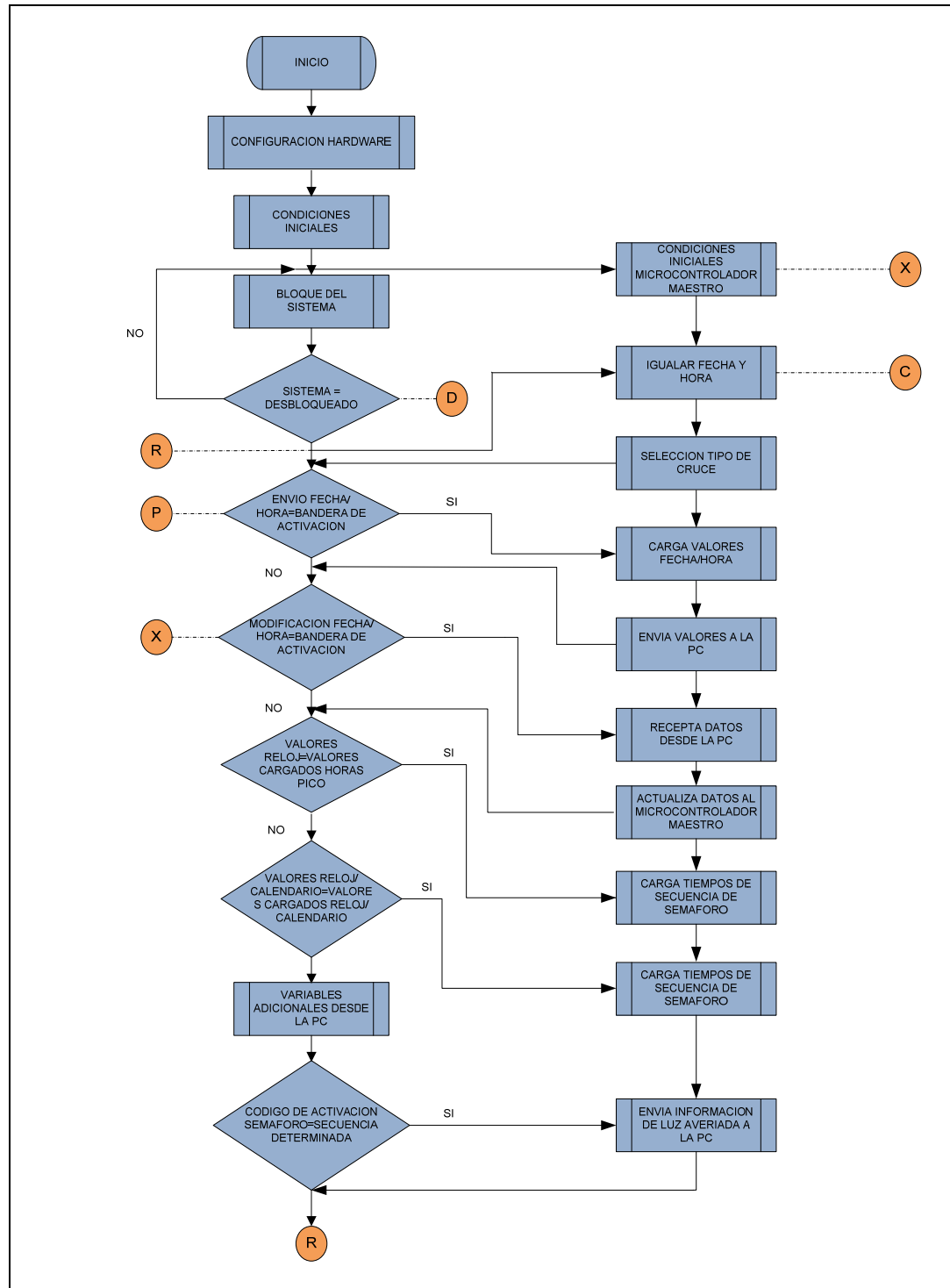
- Selección del procesador a utilizar, en este caso PIC16F877A.
- Declaración de los registros y variables en la memoria de datos.
- Vectores dentro de la memoria de programas, (interrupción y reset).
- Bloque de configuración en donde primero se especifica la dirección de la memoria de programa donde se encuentra el bloque, dando a conocer entradas y salidas para el LCD 2x16, display de 7 segmentos, sensores mecánicos, reloj de tiempo real, bus de datos, configuración de interrupciones, Timers y comunicación serial USART (“Universal Synchronous Aynchronous Receiver Transmitter”).
- Condiciones iniciales a las variables asignadas y a los puertos de salida.
- Subrutina de inicialización del LCD

Realizadas las acciones mencionadas se activan las respectivas banderas de interrupción, se inicia la presentación del cartel de bloqueo total en la pantalla y se espera por la

interrupción para que sea desbloqueada. En el momento que se desbloquea el sistema, inicia el programa principal que consta de las siguientes encuestas:

- Existencia de lámparas averiadas en los semáforos (receptada por interrupción mediante una bandera de aviso, enviada por el esclavo). Si existe avería, el maestro envía al operador del SCADA (“Supervisory Control And Data Acquisition”), la misma bandera, para que sea interpretada.
- Encuesta de activación de bandera para el inicio del conteo de vehículos mediante el puerto B, (bandera de activación receptada por interrupción mediante el puerto serial USART), “Universal Synchronous Aynchronous Receiver Transmitter”.
- Envío de información mediante el puerto serial, tiempos de las horas pico, reloj de tiempo real, tiempos del encendido de luces de la intersección, hora de encendido de luces intermitentes, fecha y hora de inicio y fin del conteo de vehículos, tiempos de carga del encendido de luces en el inicio y fin de las horas pico.
- Encuesta de días feriados. El programa principal del micro controlador maestro se observa en el anexo A del capítulo 3.

3.1.2 Diagrama de flujo del programa principal.



Fuente: Autores.

3.1.3 Interrupción general.

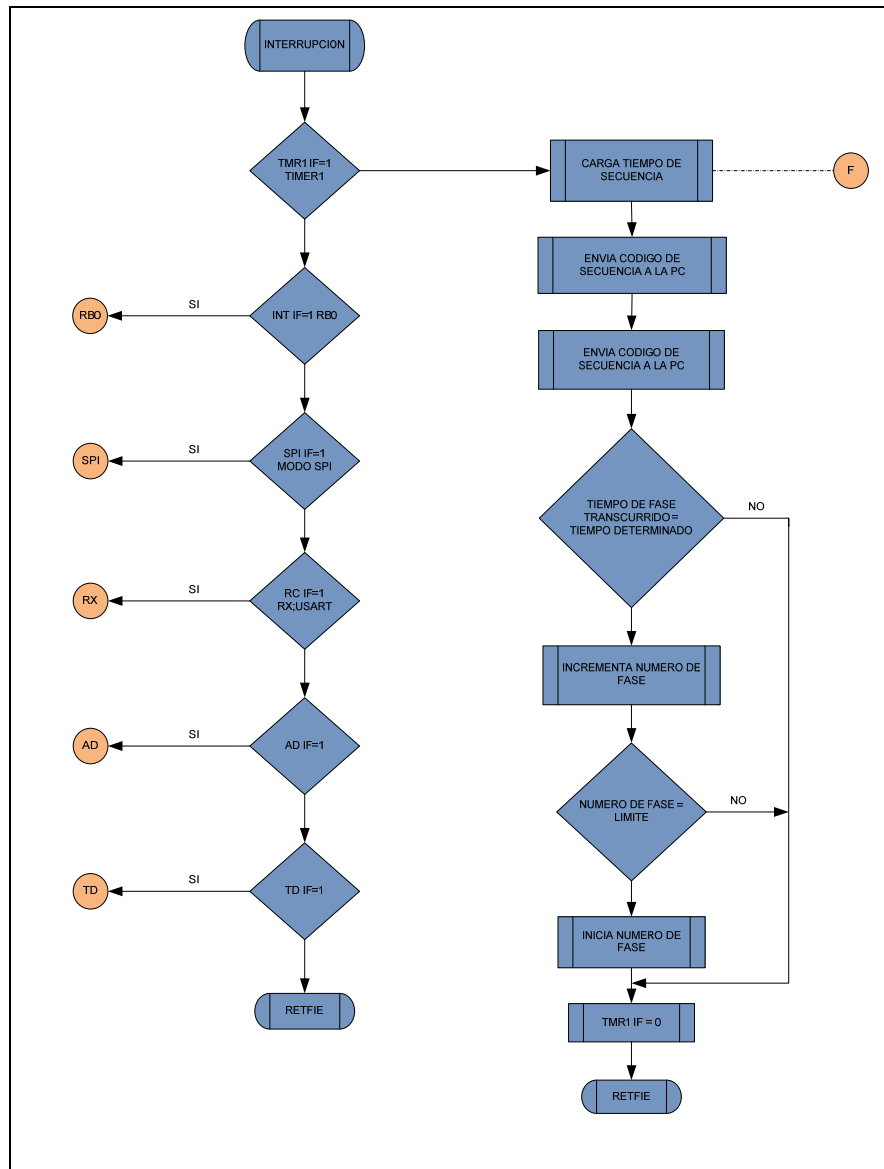
El micro controlador maestro posee una subrutina de atención a interrupción general, la cual está formada por las siguientes encuestas:

- **Encuesta de desbordamiento del Timer1 cada 16.4ms.** Para el encendido de los semáforos, interpreta si es intersección de una o doble vía, si se enciende en forma intermitente o si están averiados, cada tiempo de encendido está representado por “fase_#”, el cuál contiene el código binario de la secuencia del encendido de los semáforos enviada al esclavo, mediante el bus de datos por los pines respectivos, transmisión de una bandera por el puerto serie representando a la fase en curso.
- **Encuesta del pin RB0, en flanco de subida.** Para dar inicio al conteo ascendente de los segundos del reloj del tiempo real y así crear la visualización de fecha y hora en la pantalla del LCD 2x16.
- **Encuesta de la bandera de recepción de datos en modo SPI (“Serial Periferic Interfaz”).** Se recibe datos que concierne a la avería en alguna de las lámparas, los cuales son enviados por el esclavo.
- **Encuesta de la bandera de recepción de datos del puerto serial USART (“Universal Synchronous Asynchronous Receiver Transmitter”).** El micro controlador recibe los datos en código ASCII, interpretado para las funciones de desbloqueo de todo el sistema, cambio de tiempos de fase, cambio de tiempos de fase nocturno y diurno (luces intermitentes), modificación de reloj calendario, cambio de hora y tiempo de fase de las horas pico, modificación de hora de inicio y fin del conteo vehicular, tipo de intersección (una o doble vía), envío de fecha y hora , petición de valores de horas pico, petición de valores de las fases, petición de la cantidad de vehículos, petición de la hora para encendido de luces intermitentes, petición de horas de inicio y fin de conteo de vehículos, petición

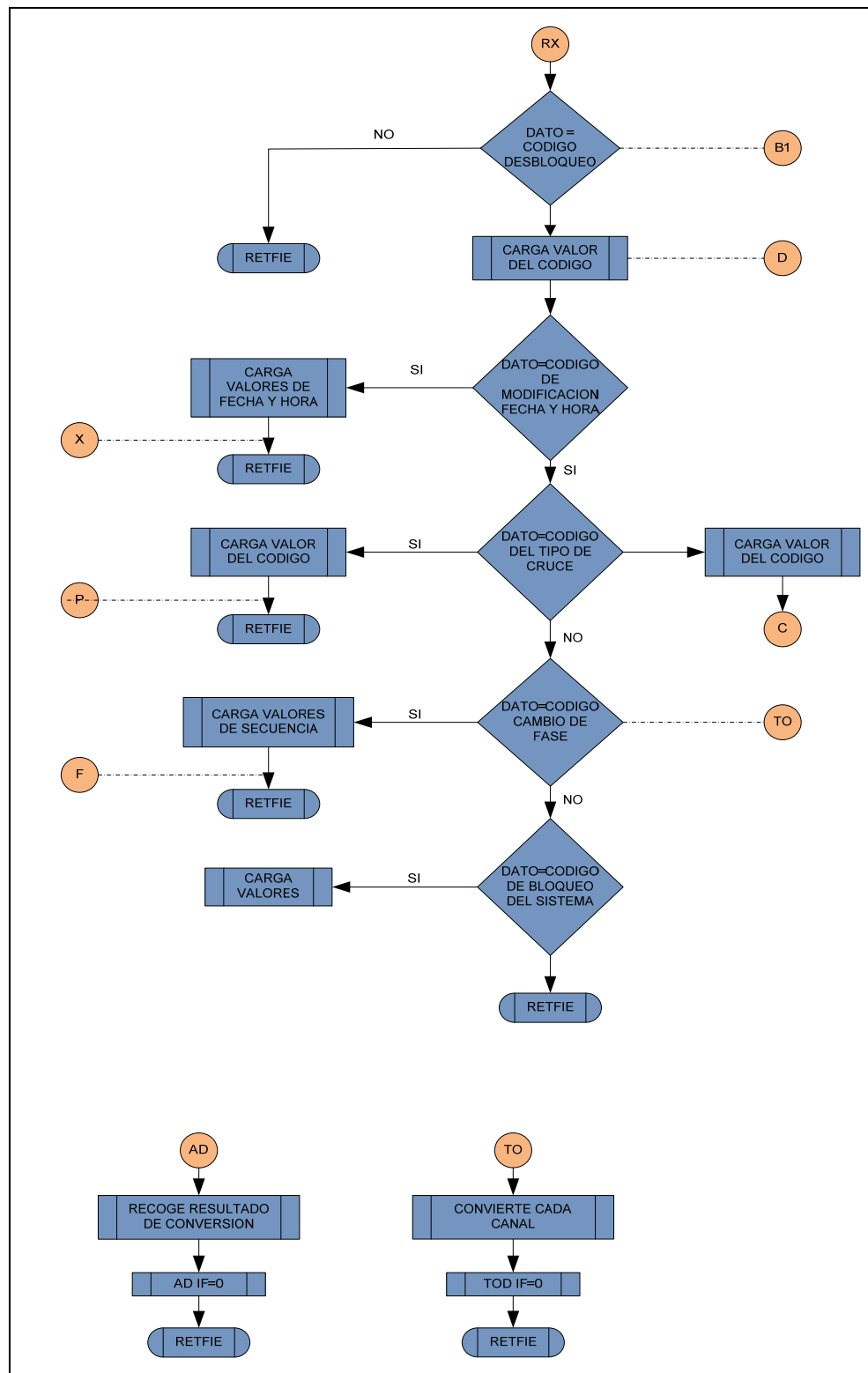
de tiempos que se va a cargar para las fases de las horas pico, petición de tipo de intersección, y cambio de luces manuales.

El programa de atención a interrupciones generales del micro controlador maestro se observa en el anexo B del capítulo 3.

3.1.4 Diagramas de flujo de interrupciones.



Fuente: Autores.



Fuente: Autores.

3.2 Programación del circuito Esclavo.

3.2.1 Programa Principal.

El programa del circuito esclavo está compuesto por los siguientes pasos:

- Selección del procesador a utilizar, en este caso PIC16F877A.
- Declaración de los registros y variables en la memoria de datos.
- Vectores dentro de la memoria de programas, (interrupción y reset).
- Bloque de configuración, en el cual se especifica primero la dirección de la memoria de programa donde se encuentra el bloque, dando a conocer las respectivas entradas y salidas para la activación de TRIACS, bus de datos, sensores de luz , configuración de registros de entradas analógicas, salidas digitales, interrupciones, Timers y comunicación serial SPI (“Serial Peripheral Interface”).
- Condiciones iniciales a las variables asignadas, puertos de salidas y activación de interrupciones.

Cuando el sistema maestro inicia el encendido de luces, el esclavo entra al programa principal el cual consta de la siguiente encuesta:

- Comparación entre los valores de variables asignadas y las banderas que impone las entradas analógicas según el estado de encendido que se encuentren los semáforos, en caso de no coincidir, entra a una subrutina para enviar en modo SPI (“Serial Periferic Interfaz”), al sistema maestro la bandera impuesta por la entrada analógica.

El sistema esclavo posee una subrutina de atención a interrupción general, la cual está formada por las siguientes encuestas de banderas:

- Encuesta por cambio de estado de las entradas (RB3-RB7) del puerto B; cada código binario está asignado a una tabla, la cual envía a una subrutina de activación de pines para los TRIACS y asigna valores a las variables para los buffers respectivos.
- Encuesta por desbordamiento del Timer 0, el cual entra a una subrutina para convertir el canal seleccionado de analógico a digital.
- Encuesta de fin de conversión del conversor A/D, compuesta por una subrutina para guardar el valor de la conversión en un canal especificado.

La encuesta está formada por subrutinas de calibración de los canales analógicos, cada canal está conformado por la suma de las corrientes que consumen las lámparas verdes de los semáforos principales, el valor de cada canal está limitado según el encendido de los semáforos, tomando como referencia la tabla 3.2.1 que muestra los resultados de los valores obtenidos.

Verde principal 1	Verde principal 2	Valor de canal analógico (hex)	Mínimo	Máximo
0	0	0xdd	0xdc	0xdf
0	1	0xcf	0xcb	0xd4
1	0	0xc4	0xc0	0xc9
1	1	0xaf	0xab	0xb3

Tabla 3.2.1: Tabla de verdad de semáforos con valores límites.

Fuente: Autores.

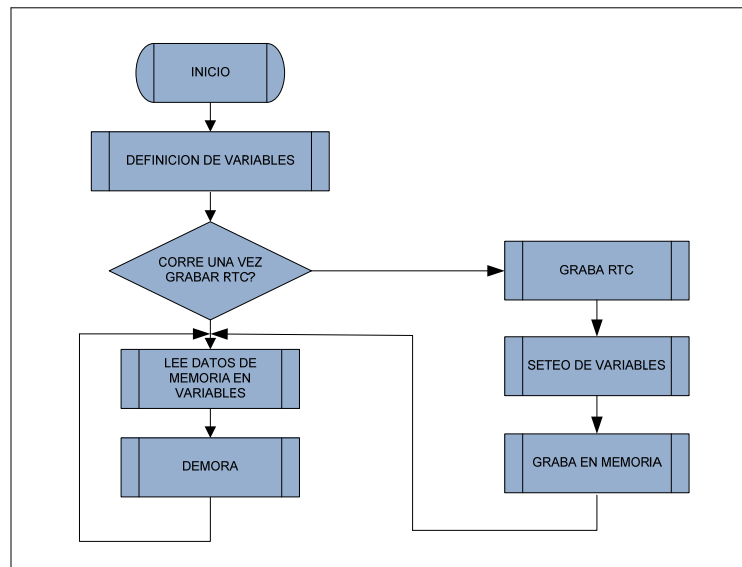
En caso, de que los valores analógicos estén fuera de la tabla, se crea una bandera de error y se envía al sistema maestro, y luego a la PC del operador.

3.3 Programación del circuito RTC.

El programa del circuito RTC (“ReaTime Clock”), está escrito en lenguaje BASIC formado por la definición de registros, variables, y un programa principal que se describe a continuación.

- Lee datos del RTC y guarda en sus variables respectivas.
- Seteo de variables, esta acción se realiza si es primera vez que corre el programa.

3.3.1 Diagrama de flujo del RTC.



Fuente: Autores.

3.4 Paneles de Control y Programación gráfica en LabVIEW.

El sistema SCADA (“Supervisory Control And Data Acquisition”), está programado en lenguaje grafico G mediante el software LabVIEW de la compañía National Instruments, que brinda dos ventanas: **Panel frontal**. En donde se puede crear paneles de control y de visualización y **Diagrama de bloques**. En el cual se encuentra la programación propuesta.

3.4.1 Funcionamiento.

El SCADA muestra dos ventanas de acceso, la primera informa si desea ingresar al sistema, (figura 4.3.1.a), si la opción es “CANCELAR” sale de la ventana y del programa, en caso de “ACEPTAR” entra a una segunda ventana, pidiendo la contraseña respectiva (la figura 4.3.1.b), si el operador ingresa la contraseña incorrecta, el programa se cancela por completo. La comunicación se realiza mediante la configuración del módulo del puerto serial, utilizando los drivers de la A.P.I. V.I.S.A. (Virtual Instrument System Architecture) para LabVIEW.

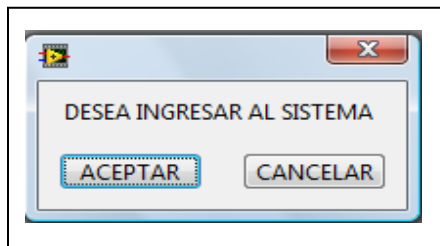


Figura 3.4.1.a: Ventanas acceso SCADA.

Fuente: Autores.

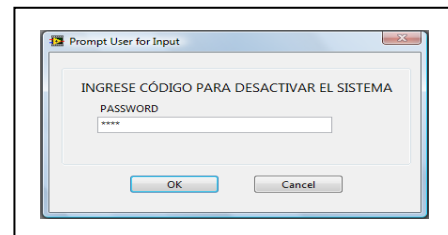


Figura 3.4.1.b: Ingreso del código.

Fuente: Autores.

Una vez ingresado al SCADA (“Supervisory Control And Data Acquisition”), el programa indica si es primera vez que se configura la unidad de control (placa física); de ser así entonces aparecerán ventanas para igualar la fecha/hora y seleccionar el tipo de intersección, (una o doble vía); en caso de ingresar al sistema más de una vez, aparece directamente el panel que corresponda al tipo de intersección configurado en la primera vez, con el estado actual del encendido de las lámparas de los semáforos, sus respectivos datos de tiempo de conmutación, reloj calendario, inicio y fin del conteo vehicular con fecha y hora, inicio y fin de luces intermitentes con la hora correspondiente. El panel de control muestra botones que indica la opción de ingresar a la modificación de una tabla de datos completa mencionada anteriormente.

3.4.2 Configuración del puerto serial

Se configura con los elementos de la A.P.I. V.I.S.A. (Virtual Instrument System Architecture), ubicada en la ventana de diagrama de bloques, en la paleta Functions>>Instrumental I/O>>Serial, donde se encuentra la función “Configure Serial Port”. Esta función permite configurar el puerto serial nombrado al módulo, en cuanto a velocidad de transmisión de “9600” baudios, “8” bits de datos, ningún bit de paridad, “1” bit de parada y ningún control de flujo, estos datos de comunicación deben ser igual al del microcontrolador maestro.

Una vez inicializada la sesión V.I.S.A, se utiliza “V.I.S.A. Read” para la lectura, se configura el ingreso del tamaño del buffer en “bytes”, utilizando el buffer de salida para poder concatenar una constante vacía en caso de que el tamaño del buffer sea cero, esto ayuda a limpiar el buffer de entrada de datos.

La escritura a través del puerto serial se puede realizar utilizando “V.I.S.A. Write”. El dato a enviar se conecta al buffer de entrada de la función; por último se cierra la sesión V.I.S.A, para liberar el puerto y poder dar otra función mediante “VISA Close”,

adicionando fuera del lazo un controlador de error llamado “Simple error Handler” como se muestra en la figura 3.4.2.

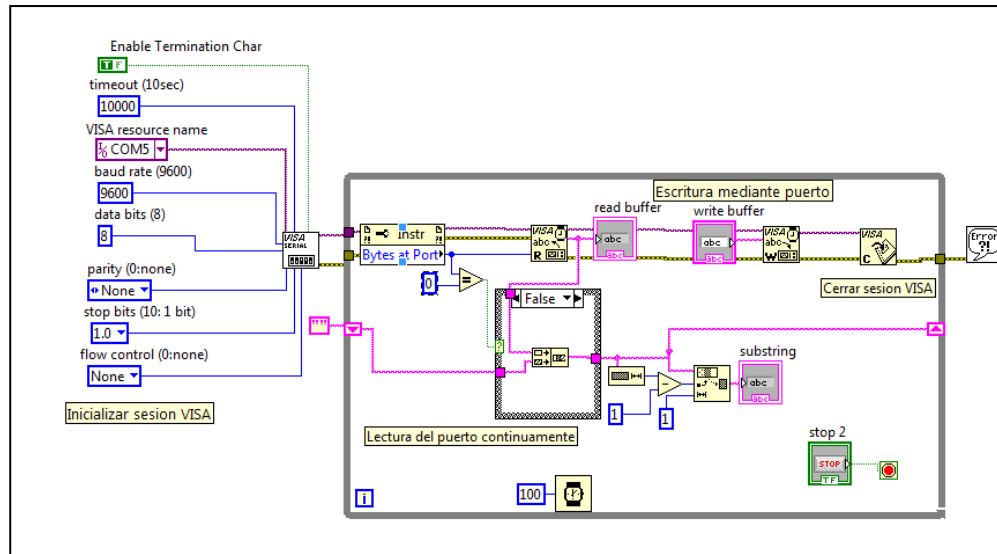


Figura 3.4.2: Diagrama de inicialización, lectura y escritura del puerto serial.

Fuente: Autores.

3.4.3 Programación en el diagrama de bloques.

La modificación de la fecha y hora se establece mediante la arquitectura de Máquina de Estado, mediante el botón “ACEPTAR” se da paso a una ventana de confirmación de ingreso para la respectiva modificación, en el siguiente paso se envía un carácter en código ASCII que interpreta el micro controlador maestro, con una determinada demora, y el envío de un string vacío “” mediante una secuencia en pila, para luego manipular los datos e ingresar al envío de caracteres, estructurado por un lazo llamado “While Loop” con su respectivo contenido, que envía string por string a través del buffer de salida al micro controlador maestro con una demora de 100 mili segundos, como se muestra en la figura 3.4.3.

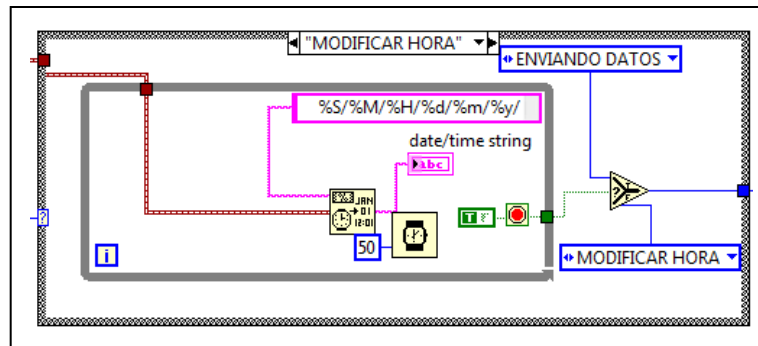


Figura 3.4.3: Diagrama para modificar la fecha y hora.

Fuente: Autores.

3.4.4 Construcción de Semáforos en el panel de control.

- Una vía, el cual está formado por dos semáforos vehiculares y dos peatonales, como se muestra en la figura 3.4.4.

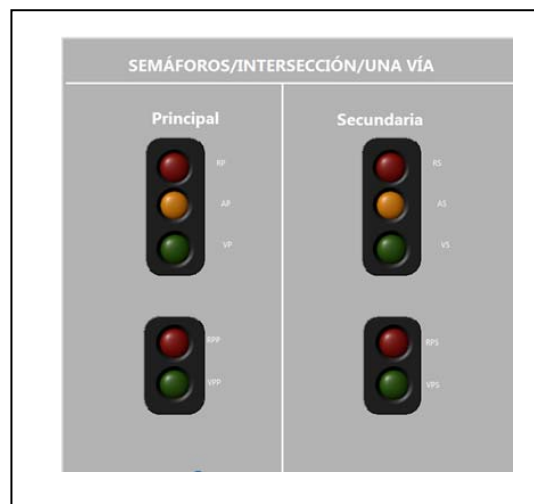


Figura 3.4.4: Visualización de semáforos.

Fuente: Autores.

3.4.5 Modificación de variables.

El panel consta de las siguientes opciones:

1. Modificación de fecha y hora.
2. Modificación de tiempos de intersección.
3. Modificación de fecha y hora para el inicio y fin del conteo vehicular.
4. Modificación de las horas pico.
5. Modificación de los tiempos en las intersecciones en horas pico.
6. Modificación de la hora de inicio y fin para luces intermitentes.
7. Modificación de contraseñas.



The image shows a software interface titled "MODIFICACIÓN DE VARIABLES" in a red header bar. Below the header, there is a list of seven menu items, each with a red "INGRESAR" button next to it. The items are: "Fecha/hora" (with a sub-label "HORA/FECHA" and a display showing "00:00:00 DD/MM/YYYY"), "Tiempos de fase", "Conteo de Vehículos", "Horas Pico", "Tiempos de Horas Pico", "Hora Intermitente (Nocturna/Diurna)", and "Contraseña". To the left of the first item, there is a small "OK" button.

Figura 3.4.5: Modificación de variables.

Fuente: Autores.

Las modificaciones de las variables se realizan mediante la arquitectura de Máquina de Estado; compuesta por estructuras de caso, ventanas de diálogo, estructuras de secuencias planas (flat), ventanas de modificación de variables y lazos While loop.

La arquitectura de Máquina de Estado está formada por seis casos etiquetados por los siguientes nombres: Inicio, pregunta, envía bandera, modifica, envía datos y fin.

1. Inicio.- Se encuentra uno de los botones del panel de modificación de variables, en caso de ser presionado va a la etiqueta “pregunta”, caso contrario va a “inicio”, como se indica en la figura 3.4.5.1.

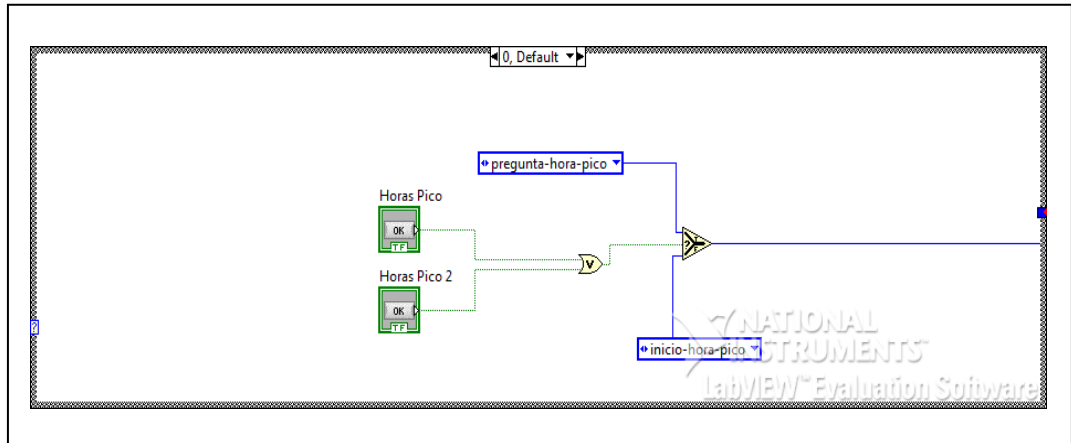


Figura 3.4.5.1: Testeo de ingreso.

Fuente: Autores.

2. Pregunta.- En este caso aparece una ventana de diálogo con la opción de ingresar las variables, en caso de presionar el botón “Aceptar” va a la siguiente etiqueta “envía bandera”, caso contrario se presiona el botón “Cancelar” va a “inicio”, como se muestra en la figura 3.4.5.2.

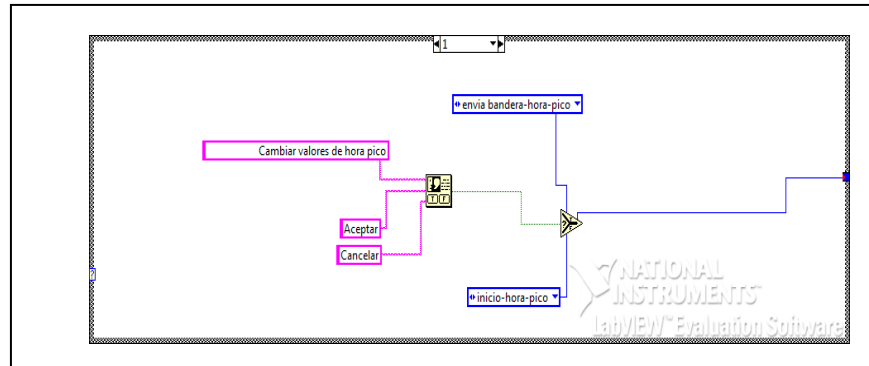


Figura 3.4.5.2: Pregunta de ingreso.

Fuente: Autores.

3. Envía bandera.- Conformada por una estructura de secuencia plana de tres pasos, el primero envía la letra “D” al buffer de salida del puerto serial V.I.S.A. (Virtual Instrument System Architecture), con una demora de 100ms, el siguiente envía una constante vacía “” al buffer con el mismo valor de demora y el último paso especifica la siguiente secuencia a ejecutarse mediante una constante numérica, como se muestra en la figura 3.4.5.3.

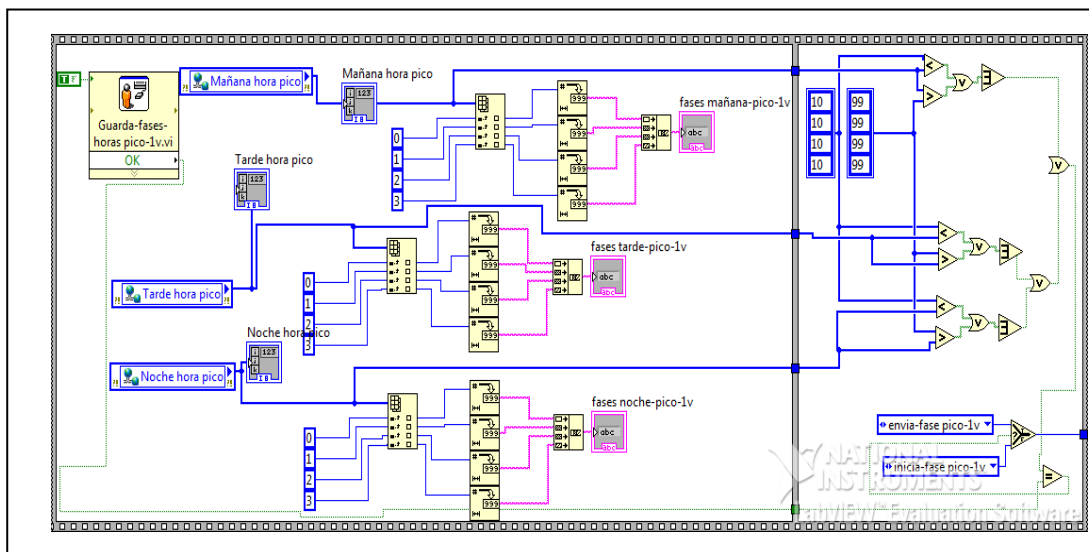


Figura 3.4.5.3: Recepción de datos para las horas pico.

Fuente: Autores.

4. Modifica.- Formada por la estructura de secuencia plana de dos pasos, en el primero se encuentra el sub vi modificado por el elemento del cuadro de diálogos, dentro de él, están las variables de control en el panel frontal, y en el diagrama de bloques se encuentra una programación por defecto; se adiciona un indicador de las variables mencionadas, de los indicadores se crea una nueva variable compartida, la cual se cambia a modo de escritura y se une con un nodo de propiedad creado desde el indicador; en la primera secuencia se adicionan las variables compartidas de escritura del sub vi, también se crea un indicador para guardar los cambios respectivos, en algunos casos se concatenan todas las modificaciones, como se muestra en la figura 3.4.5.4.

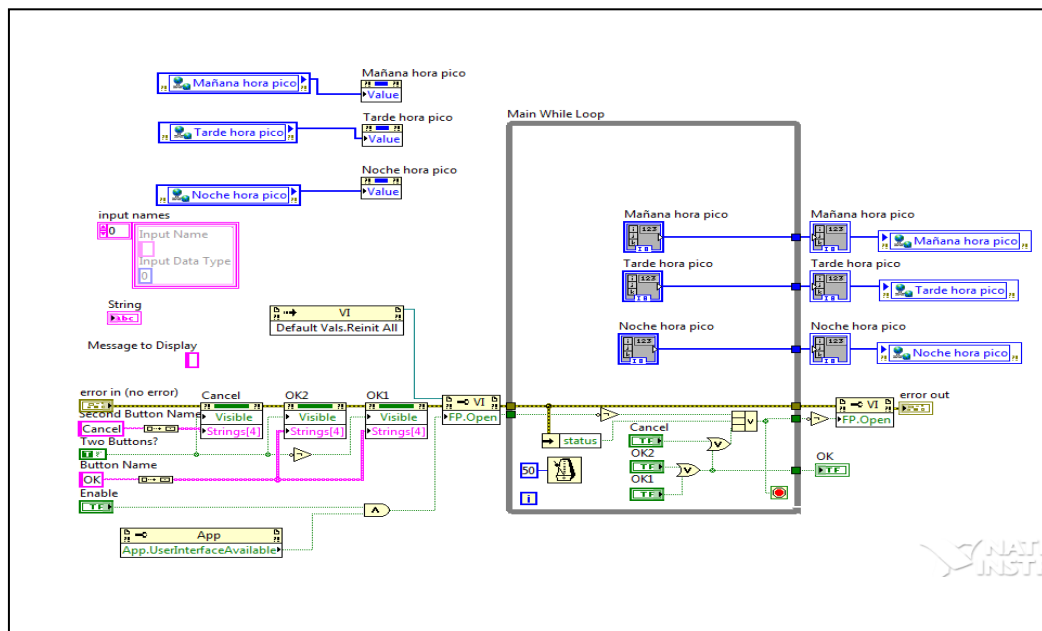


Figura 3.4.5.4: Modificación de tiempos de intersección de horas pico.

Fuente: Autores.

En el segundo paso de la secuencia plana se encuentra el selector de opciones, en caso de aceptar los cambios en el primer paso va a la secuencia “envía”, si se

presiona el botón “cancelar” va a la secuencia “fin”, como se muestra en la figura 3.4.5.5

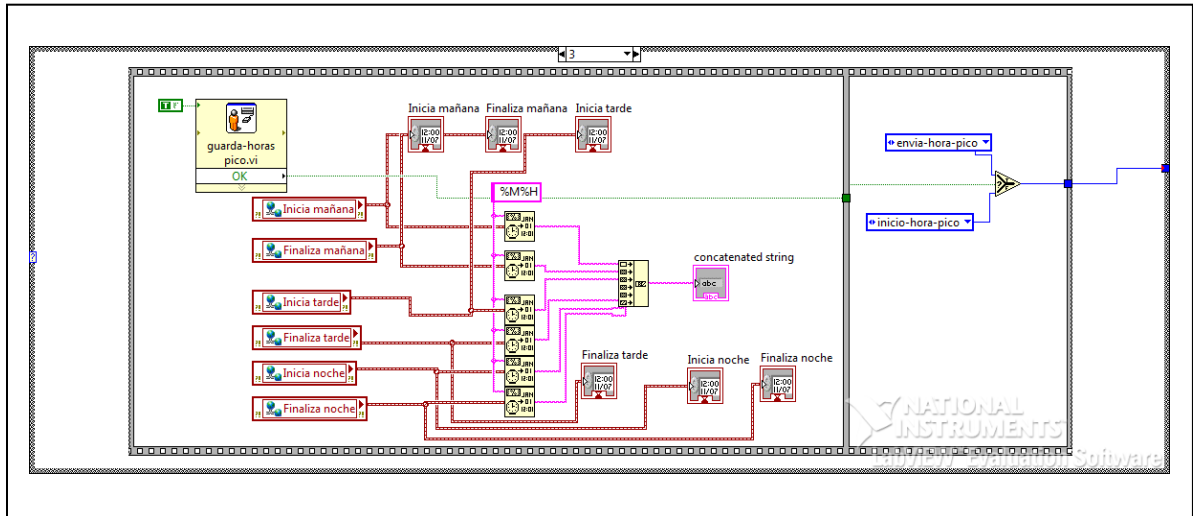


Figura 3.4.5.5: Valores modificados y selección de secuencia.

Fuente: Autores.

5. Envía.- Para enviar los valores modificados se utiliza el primer paso de la estructura de secuencia plana, en la que se encuentra la estructura “while loop” que a su vez está formada por una estructura de secuencia plana, en la primera secuencia envía un carácter al buffer de salida del puerto serial V.I.S.A. (Virtual Instrument System Architecture), luego se envía un carácter vacío “”, y por último envía un carácter específico para cada modificación. El segundo paso de la primera estructura de secuencia plana se especifica en la siguiente secuencia a ejecutarse, mediante una constante numérica, como se muestra en la figura 3.4.5.6.

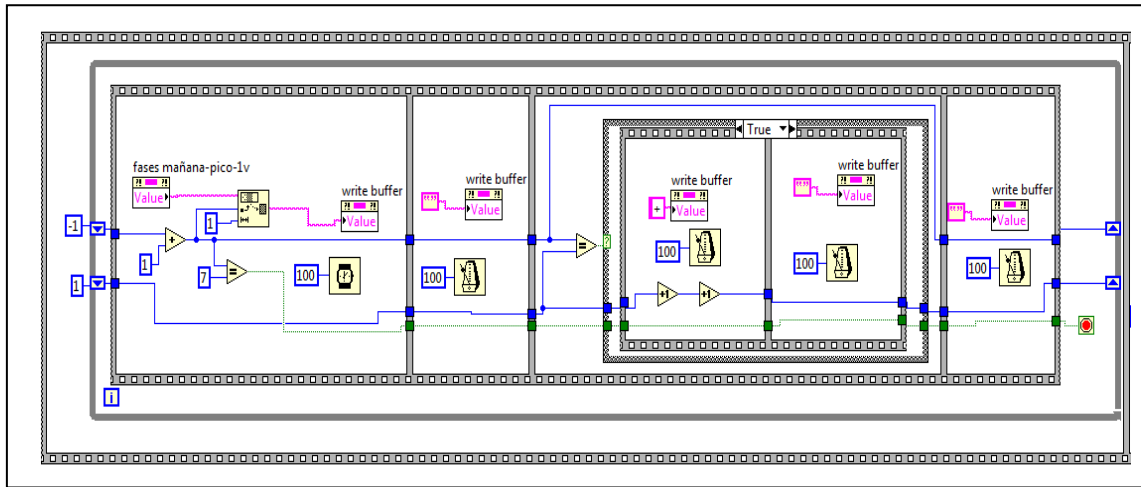


Figura 3.4.5.6: Envío de caracteres ASCII al micro controlador.

Fuente: Autores.

6. Fin.- Especifica la siguiente secuencia a ejecutarse que es “inicio”, para continuar el ciclo de testeo y modificación.

CAPÍTULO 4

PRUEBAS FUNCIONALES

4.1 Equipo de funcionamiento.

En la figura 4.1 se observan los componentes utilizados para el desarrollo del proyecto, el mismo que consta de placas físicas, maqueta a escala, panel de control, a continuación se describen cada uno de ellos.

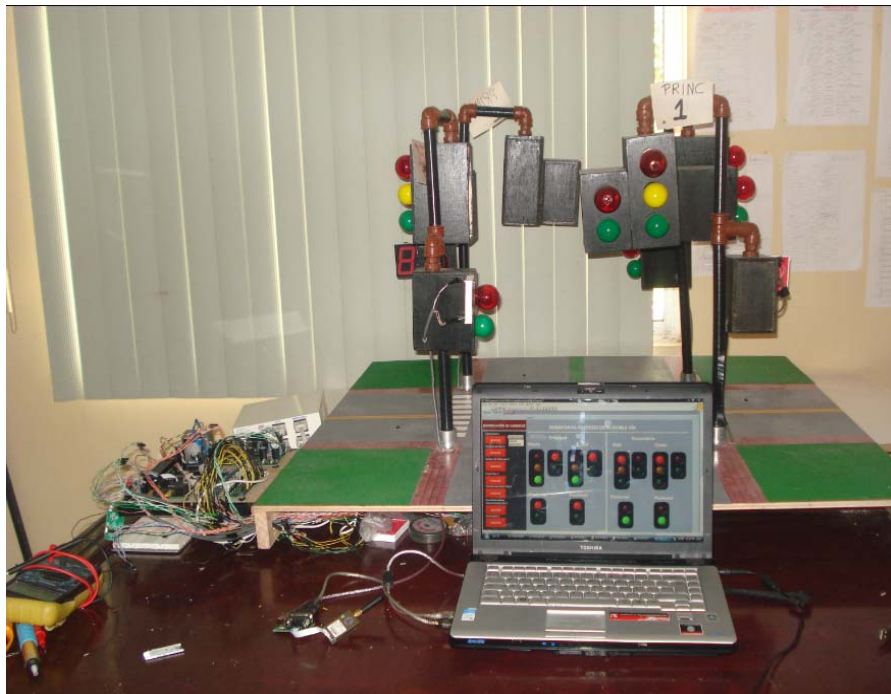


Figura 4.1: Equipo de funcionamiento.

Fuente: Autores.

4.1.1 Placas físicas.

La figura 4.1.1, muestra la placa física maestro-esclavo, en donde se observan los micro controladores maestro y esclavo, LCD (Liquid Crystal Display), relés de conmutación, driver multicanal MAX 232, decodificadores SN74SL47 con sus respectivas conexiones para el circuito de control, DS1307 con el micro controlador 16F628A para el circuito del reloj de tiempo real y el transceiver para la comunicación inalámbrica.

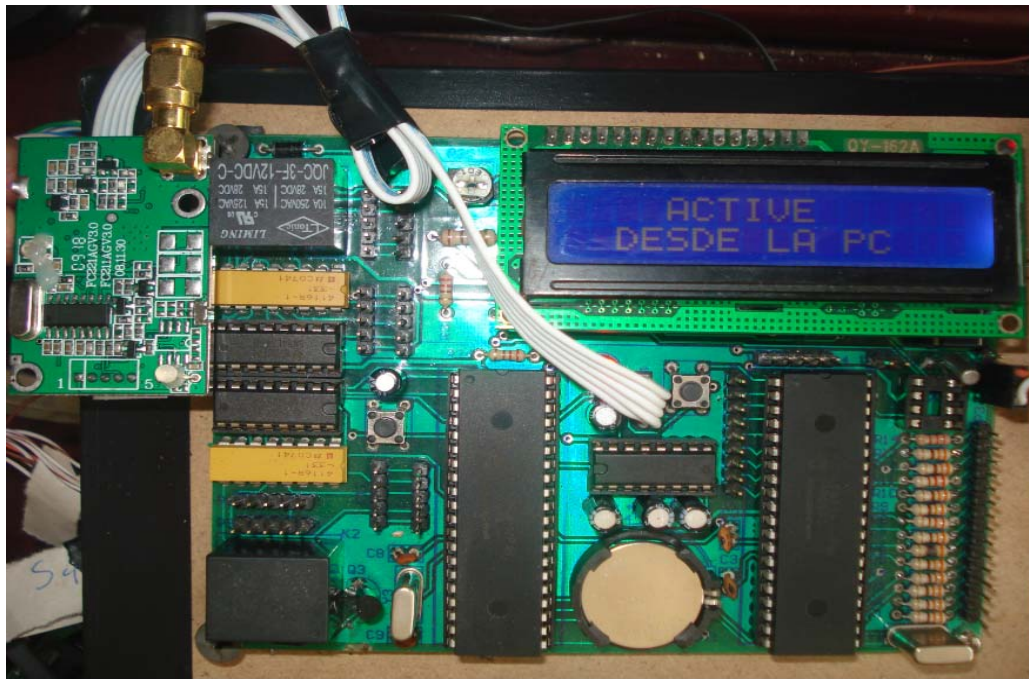


Figura 4.1.1: Placa física del circuito maestro-esclavo.

Fuente: Autores.

En la figura 4.1.4, se muestra la placa física del circuito de la fuente simétrica (+12 V y +5 V), con batería auxiliar, en la cual se encuentra un transformador de 3 Amperios, condensadores electrolíticos de 2200uF y un integrado 555. Este circuito alimenta a todas las placas físicas del proyecto mediante las respectivas borneras.

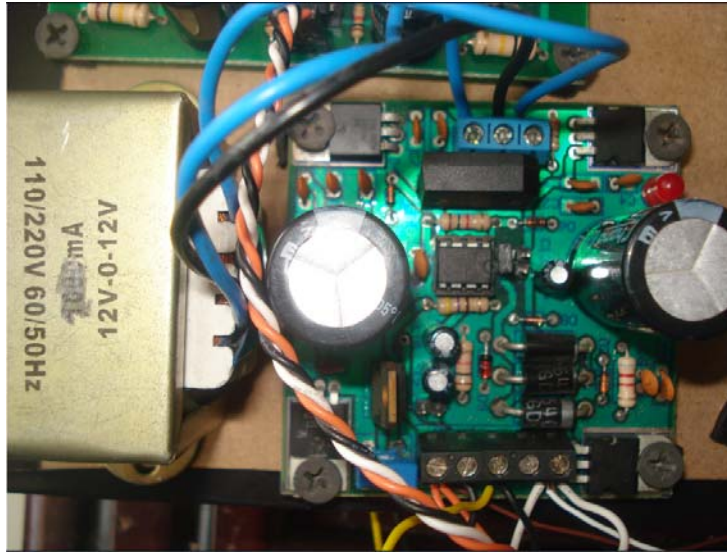


Figura 4.1.4: Placa física del circuito de la fuente simétrica (+-12V y +-5V).

Fuente: Autores.

La figura 4.1.5, muestra la placa física para los sensores de luz, la misma que contienen condensadores de 470uF y diodos para rectificar y filtrar la corriente AC proporcionada por las lámparas, operacionales TL084, TL082 para sumar las corrientes de las lámparas y conectores I.D.C.

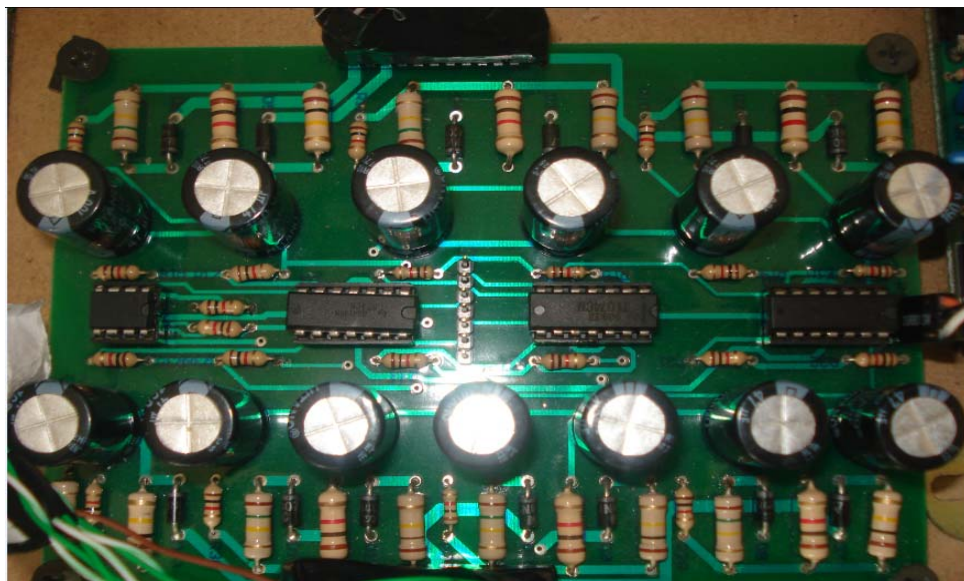


Figura 4.1.5: Placa física para los sensores de luz.

Fuente: Autores

En la figura 4.1.6, se muestra la placa física del circuito de conmutación con TRIACS, con sus respectivos condensadores y borneras para la alimentación de las lámparas.

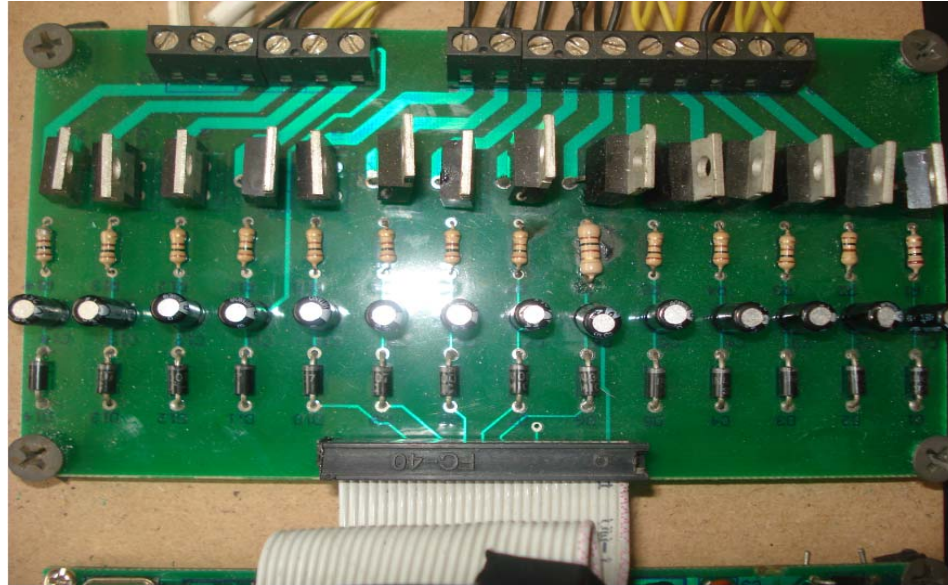


Figura 4.1.6: Placa física del circuito de conmutación con TRIACS.

Fuente: Autores.

4.1.2 Maqueta a escala.

Debido a la necesidad de poner en práctica lo antes mencionado es necesario contar con un modelo a escala para dicha demostración, en la figura 4.1.2, se observan los semáforos, displays de 7 segmentos y pulsantes mecánicos que serán de mucha ayuda para verificar el correcto funcionamiento del equipo.



Figura 4.1.2: Maqueta a escala.

Fuente: Autores.

4.1.3 Panel de control.

En la figura 4.1.3, se muestra el panel de control o sistema SCADA (“Supervisory Control And Data Acquisition”), para controlar y monitorear los semáforos de viraje y peatonales, acceso a la modificación de variables como fecha/hora, tiempos de face, tiempo de horas pico, luces intermitentes y contraseña.



Figura 4.1.3: Panel de control.

Fuente: Autores.

4.2 Récord anecdótico en el desarrollo del prototipo.

En la prueba de funcionalidad de la placa física del circuito de TRIACS se comprobó que el espacio entre las huellas para los pines de los TRIACS BT-136 era demasiado pequeño, por lo que al encender la placa se observó que algunos de los TRIACS se quemaron, debido a dicho problema; por lo que se recomienda en la construcción del P.C.B. (Printed Circuit Board), considerar que al trabajar con altos voltajes debe existir la separación necesaria entre ellas.

En el circuito de la fuente con batería auxiliar se llegó a la conclusión de cambiar el transformador de 1 Amperio de corriente por uno de 2 Amperios, ya que en el momento de encender todo el equipo disminuían los voltajes en la fuente simétrica de ± 12 V y se apagaban los micro controladores en la placa del circuito maestro-esclavo.

4.3 Resultados de las pruebas realizadas.

Las pruebas de simulación de control y transmisión de datos del sistema, se realizaron en el software PROTEUS 7.5, que tiene la capacidad de simular virtualmente el código realizado en lenguaje ensamblador del micro controlador con la finalidad de asegurar el correcto funcionamiento de la placa física previo a la elaboración del P.C.B. Todos los circuitos se armaron en varios “Proto Boards”, para comprobar el funcionamiento total del sistema de control, transmisión de datos, conmutación y así ensamblar todos los componentes de los circuitos anteriores en las placas físicas.

La configuración de los transceivers inalámbricos pertenecientes a la PC y a la placa de control se realizó mediante el software FC221, configurados con el mismo canal, igual número de bit de datos, radiofrecuencia en Baudios y bits de paridad, logrando con éxito la transmisión y recepción de datos.

CONCLUSIONES

De acuerdo a las pruebas realizadas en el diseño y construcción del hardware, software y maqueta a escala, ponemos a consideración las siguientes conclusiones:

Los objetivos planteados en cada etapa del proyecto se alcanzaron de acuerdo a las expectativas, en forma global se puede asegurar que el sistema está perfectamente capacitado para ser aplicado en cualquier ciudad. A demás, para llegar a una comunicación inalámbrica, no se necesita dispositivos de alto costo y muy robustos, por lo que, se implementan módulos de radiofrecuencias de baja potencia para transmitir datos entre dos terminales a una distancia de 800m, se disminuyen costos y mano de obra, lo cual resulta muy importante.

Debido a que el sistema se aplica a una maqueta a escala, puede ser utilizado como material didáctico para los estudiantes de la universidad en algunas materias de nuestra carrera.

BIBLIOGRAFÍA

REFERENCIAS BIBLIOGRÁFICAS.

- ANGULO, José; ANGULO, Ignacio. *Microcontroladores PIC: Diseño Práctico De Aplicaciones*. España. Editorial McGraw-Hill/Interamericana de España S.A, 2003.
- ANGULO USATEGUI, José María; GARCÍA ZAPIRAIN, Begoña; ANGULO MARTÍNEZ, Ignacio. *Microcontroladores Avanzados DsPIC: Controladores Digitales De Señales. Arquitectura, Programación Y Aplicaciones*. España. Editorial Thomson Paraninfo, 2006.
- ALTIUM DESIGNER. Tutorial: Creating Library Components. U.S.A. 2008.
- AXELSON, Jan. *Serial Port Complete. Programming and Circuits for RS-232*. Lakeview Research. USA.
- BATES, Martin. *Interfacing PIC Microcontrollers. Embedded Design by Interactive Simulation*. Reino Unido. Editorial Elsevier Ltd. 2007.
- FRIENDCOM. *Manual Del Usuario. FC-221/AG ISM Narrow-band RF Module*. USA. 2009.
- HERRERA, Luis A. *Sistemas de supervisión y Adquisición De Datos*. Argentina. Universidad De Buenos Aires. 2008.
- LAZARO, Antonio. *LABVIEW Gi Programación Gráfica para Control e Instrumentación*. USA. Editorial Paraninfo. 2001.

- REMIRO DOMÍNGUEZ, Fernando. *Curso De Programación De Microcontroladores PIC: En MPLAB*. España. Editorial Resistor, 2000.
- SANCHEZ, Julio; CANTON, Maria P. *Microcontroller Programming. The Microchip PIC*. USA. Editoril Taylor and Francis. 2007.
- TOCCI, Ronald. *Sistemas Digitales: Principios y Aplicaciones*. México. Editorial Prentice Hall. 2007
- UVa. Departamento De Electricidad y Electronica. *Manual De Proteus. Circuitos Digitales*. España. 2006
- WILMSHURST, Tim. *Designing Embedded Systems with PIC Microcontrollers. Principles and applications*. Reino Unido. Editorial Elsevier Ltd. 2007.

REFERENCIAS ELECTRÓNICAS.

- MICROCHIP TECHNOLOGY Inc. PIC 16F877A Datasheet. microchip.com. U.S.A. 2003. Disponible en Web: <<http://www.microchip.com/downloads/en/DeviceDoc/16F877A.pdf>>.
- NATIONAL INSTRUMENTS. NI Vision Builder for Automated Inspection Development Toolkit Tutorial. ni.com. U.S.A. 2009. Web: <<http://www.ni.com/ToolkitTutorial.pdf>>.
-
- NATIONAL INSTRUMENTS. Tutorial: How to do a Serial Loopback Test. ni.com. U.S.A. 2003. Disponible <<http://www.ni.com/>>.

- PHILIPS Semiconductors. TRIACS BT136 series. Esensitive gate. Datasheetcatalog.com. USA. 2001. Disponible <<http://www.PHILIPS.com/datasheetcatalog.pdf>>.
- TEXAS INSTRUMENTS. MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS Datasheet. .focus.ti.com. U.S.A. 2002. Disponible <<http://www.texasinstruments.com/downloads/datasheets max232.pdf>>.
- MICROCHIP. *PIC16F627A/628A/648A Data Sheet*. [en línea] U.S.A. Microchip Technology Inc, 2009 [ref. de noviembre de 2009]. Disponible en Web: <<http://ww1.microchip.com/downloads/en/DeviceDoc/40044G.pdf>>.

ANEXOS

CAPÍTULO 2 DISEÑO Y CONSTRUCCIÓN DEL HARDWARE

ANEXO A

Diagrama del P.C.B. (“Printed Circuit Board”), maestro-esclavo-RTC.

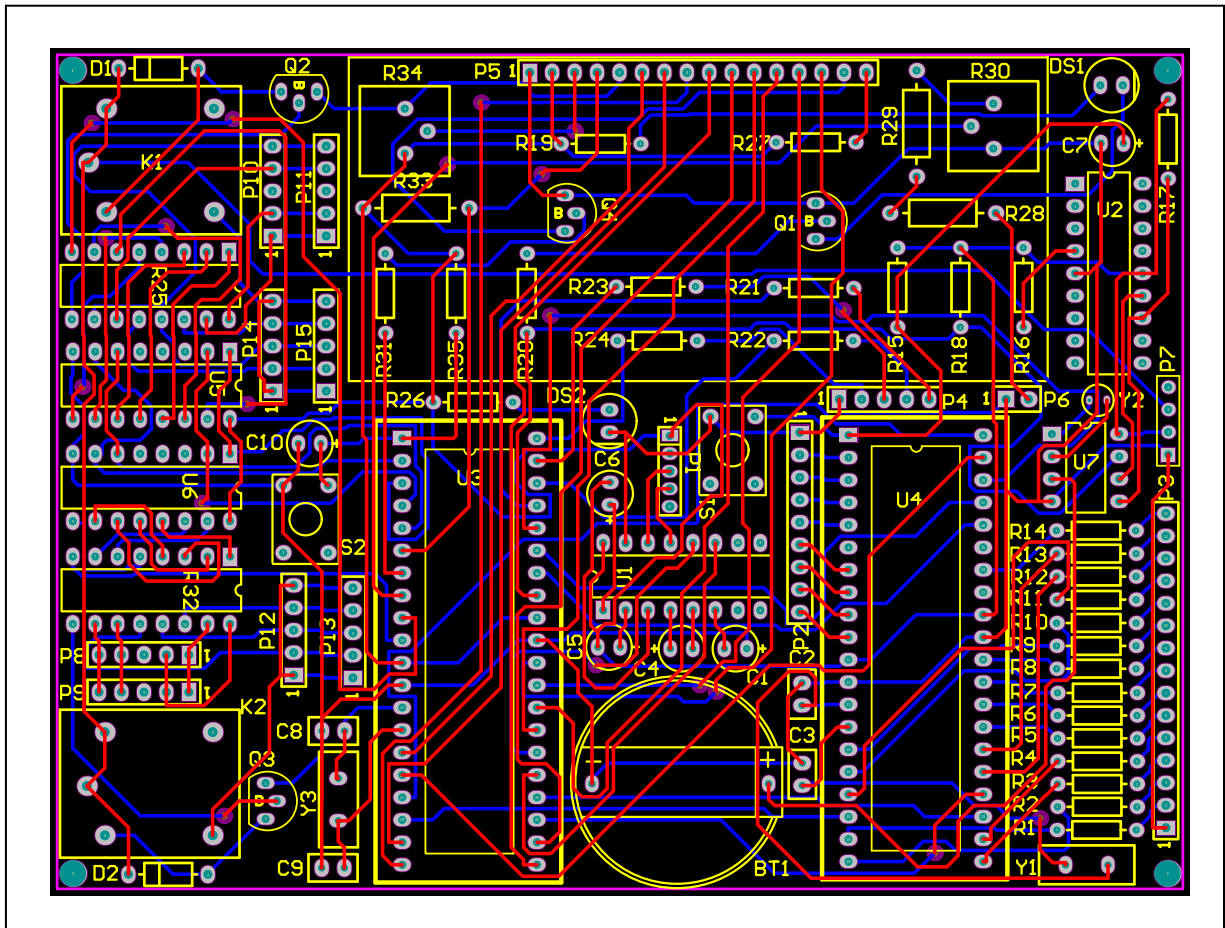


Figura 2.7: Diagrama del P.C.B. (“Printed Circuit Board”), para el circuito maestro-esclavo-RTC.

Fuente: Autores.

CAPÍTULO 2

DISEÑO Y CONSTRUCCIÓN DEL HARDWARE

ANEXO B

Diagrama del P.C.B. (“Printed Circuit Board”), sensores de consumo de corriente.

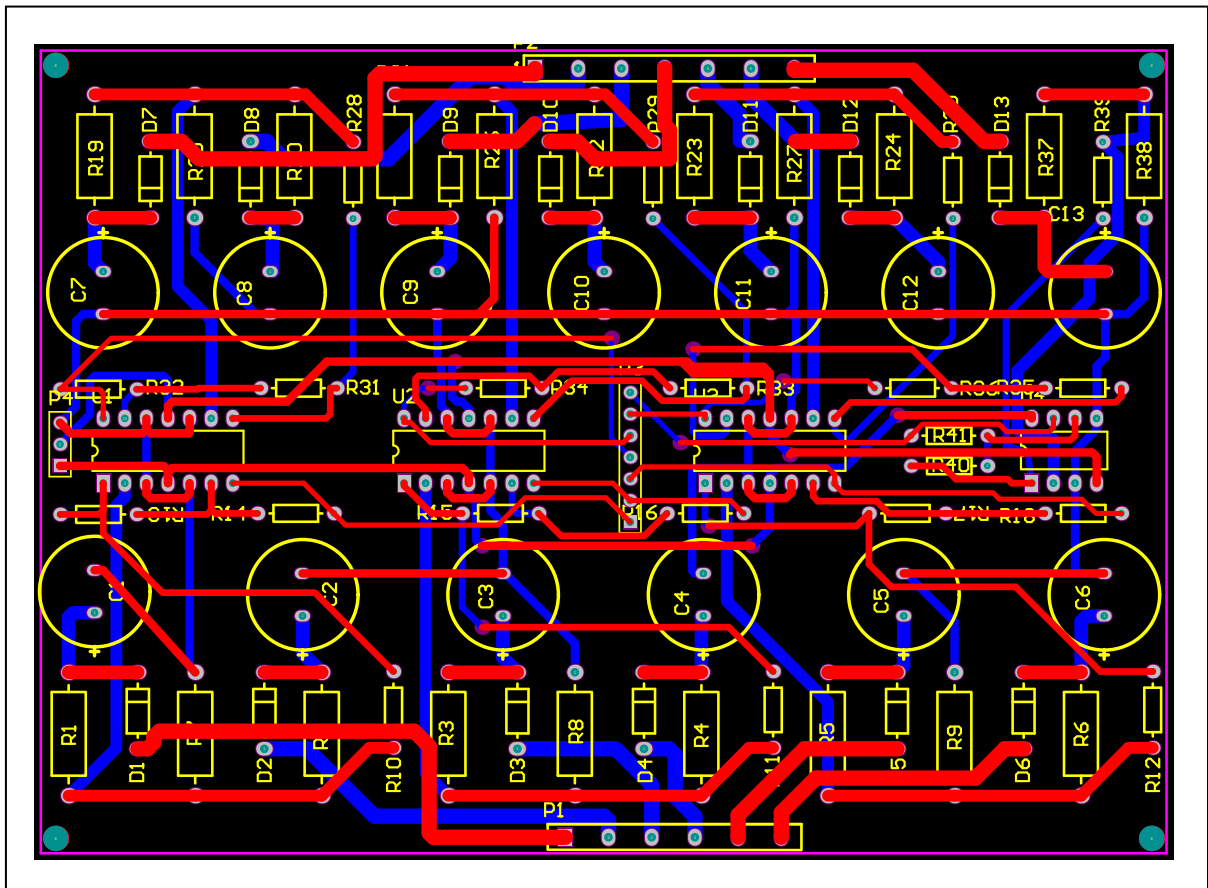


Figura 2.8: Diagrama del P.C.B. (“Printed Circuit Board”), para el circuito sensores de consumo de corriente.

Fuente: Autores.

CAPÍTULO 2

DISEÑO Y CONSTRUCCIÓN DEL HARDWARE

ANEXO C

Diagrama del P.C.B. (“Printed Circuit Board”), de TRIACS.

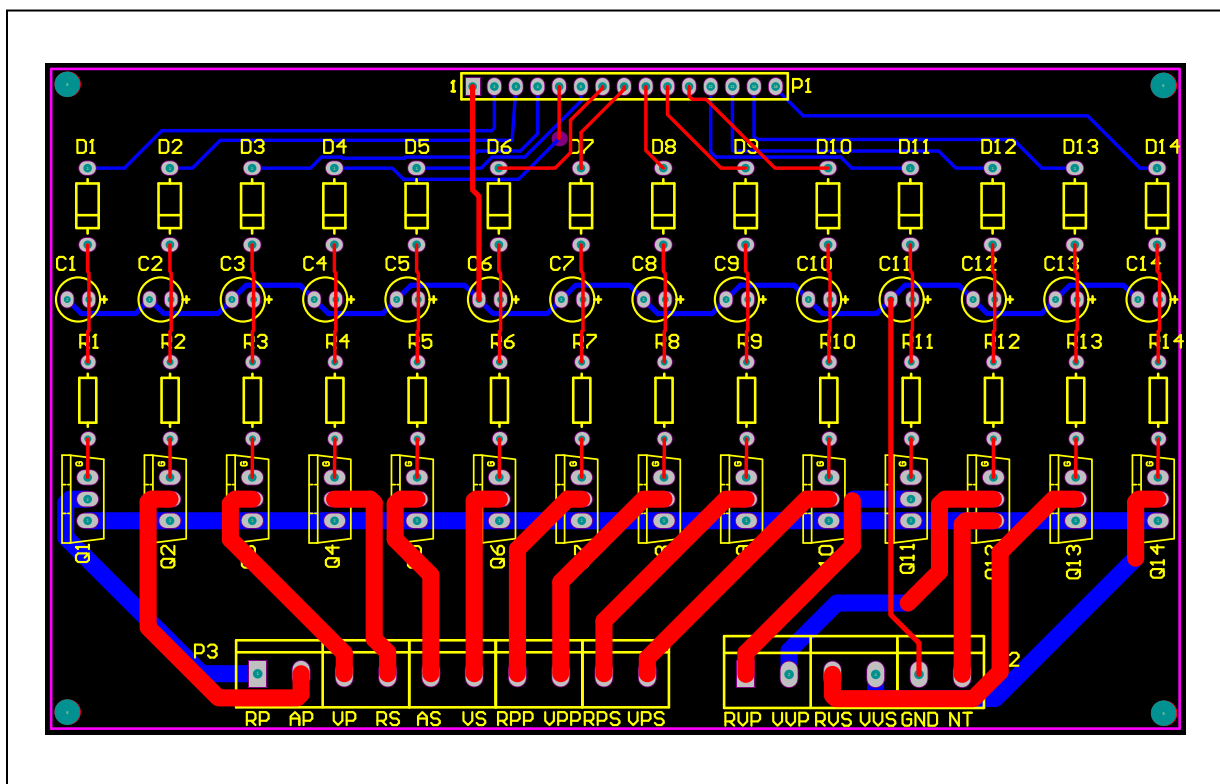


Figura 2.9: Diagrama del P.C.B. (“Printed Circuit Board”), para el circuito de TRIACS.

Fuente: Autores.

CAPÍTULO 2

DISEÑO Y CONSTRUCCIÓN DEL HARDWARE

ANEXO D

Diagrama del P.C.B. (“Printed Circuit Board”), de la fuente de alimentación.

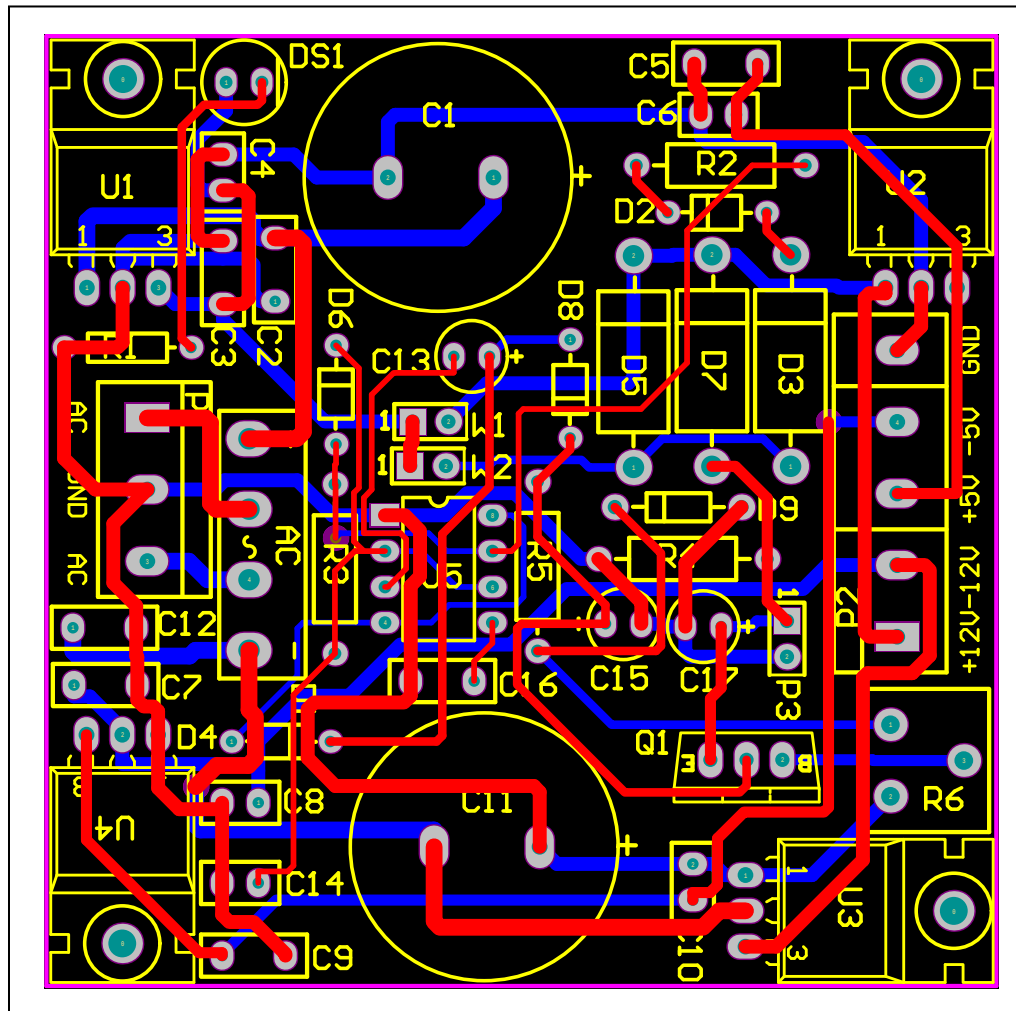


Figura 2.10: Diagrama del P.C.B. (“Printed Circuit Board”), para el circuito de la fuente de alimentación.

Fuente: Autores.

CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE

ANEXO A

Código del programa principal del micro controlador maestro.

```

;=====
===
;PROGRAMA PRINCIPAL
;=====
===
main
;=====
===
;ENCUESTA PARA RECEPCIÓN DE SPI
movf spi_alto,w
sublw 0x00
btfss STATUS,2
goto preg_I
goto preg_bajo
preg_I movf spi_alto,w
movwf TXREG ;ENVIA DATO EN BUFFER DE SALIDA
clrf spi_alto
nop
bcf PIR1,TXIF
call dem_5ms
preg_bajo movf resp_spi,w
sublw 0x00
btfss 3,2
goto envia_bajo

```

```

goto test_conteo
envia_bajo movf resp_spi,w
movwf TXREG ;ENVIA DATO EN BUFFER DE SALIDA
clrf resp_spi
nop
bcf PIR1,TXIF
call dem_5ms
;=====
===
;TESTEO DE BANDERA DE INICIALIZACIÓN DE CONTEO DE VEHÍCULOS
test_conteo movf start_conteo,w
sublw d'1'
btfss STATUS,2
goto test_go_hora
goto Princ_1
;-----
test_go_hora
movf pet_valor,w
sublw d'0'
btfss STATUS,2
goto go_valores
goto next_test
;=====
=====;TESTEO DE INICIO Y FINALIZACIÓN DE HORAS PICO EN LA
MAÑANA, TARDE Y ;NOCHE
next_test goto event_mañana
;-----
;TESTEO DE HORA Y FECHA DE LAS CALLES PRINCIPAL Y SECUNDARIA
conteo_#calle goto inicia_calle1

```

```

;=====
=====
=====
go_valores movf system_start ,w
sublw d'1'
btfs STATUS,2 ;PREGUNTA SI CÓDIGO DE DESBLOQUEO DE TODO ES
SISTEMA ES UNO??
goto next_test ;NO ES UNO; REGRESA A PROGRAMA PRINCIPAL
movf pet_valor,w ;PREGUNTA SI BANDERA DE TESTEO
sublw d'1' ;ES CORRESPONDIENTE PARA EMPEZAR A ENVIAR
btfs STATUS,2 ;DATOS DE FECHA Y HORA
goto test_send_pico ;VA A TESTEA OTRA BANDERA
;=====
;ENVIA DATOS A PC DE FECHA Y HORA:
;MINUTO,HORA, DÍA, MES Y AÑO
movlw 'A' ;ENVÍA BANDERA DE INICIO DE FECHA Y HORA
call tx_trans
movlw 0x2d ;INGRESA LA DIRECCIÓN A MANIPULAR
movwf FSR ;APUNTA A LA DIRECCIÓN ANTERIOR
movlw 0x31 ;INGRESA LÍMITE DE CONTADOR
movwf cont_tx
goto send_to_pc
;-----
;-----
tx_trans movwf TXREG ;ENVIA DATO EN BUFFER DE SALIDA
nop
bcf PIR1,TXIF
call dem_5ms
return
;=====
=====

```

```

;=====
=====
;ENVIA VALORES CORESPONDIENTES A PC
send_to_pc movf INDF,w ;MUEVE EL VALOR DEL FSR A W
call conv_tx;goto conv_tx ;CONVIERTE EL DECIMAL EN ASCII
movf FSR,w
subwf cont_tx,w;lw 0x31 ;RESTA SI ES EL ÚLTIMO VALOR A ENVIAR
btfss STATUS,2 ;PREGUNTA SI ES ÚLTIMO
goto add_dir_tx ;VA A INCREMENTAR SIGUIENTE DIRECCIÓN
goto clear_dir_tx;LIMPIA EL CONTADOR DE DIRECCIÓN
add_dir_tx incf FSR,1;INCREMENTA CONTADOR
goto send_to_pc ;LOOP BACK
clear_dir_tx bcf STATUS,5 ;BANCO 0
clrf pet_valor ;LIMPIA BANDERA DE PETICIÓN DE HORA
;TERMINA ENVIAR FECHA Y HORA
;PASA A ENVIAR BANDERA DE FINALIZACION DE ENVIO
movlw 'Z'
call tx_trans
goto next_test ;SIGUE AL PROGRAMA PRINCIPAL

```

CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE

ANEXO B

Código de programa de atención a interrupciones generales del micro controlador maestro.

```

;=====
===
;=====
===
; SUBROUTINA DE INTERUPCION
;=====
===
IT
;-----
;Respalda registros
movwf wtemp ;RESPALDO AC
swapf STATUS,w
movwf stat_temp
movf FSR,0
movwf fsr_temp ;RESPALDO EL FSR
;-----
; PREGUNTO SI SE DESBORDO EL TIMER 1
TIMER_1 btfss PIR1,0 ; BIT TMR1IF = 1???
goto RB0_INT ;NO HAY INT, VA A PREGUNTAR POR INT DE RB0
movlw 0x08
movwf PCLATH
call semaforo ;SI HAY INT, VA A FUNCIONAR SEMÁFORO
clrf PCLATH

```

```

goto back_t1
;-----
;PREGUNTO SI FLANCO DE SUBIDA DE RB0=1
RB0_INT btfss INTCON,1 ; BIT INTF = 1???
goto INT_SPI ;NO HAY INT, VA A PREGUNTAR POR INT DE SPI
goto RELOJ ;SI HAY INT VA A INCREMENTAR EL SEGUNDO DE RELOJ
;-----
;ENCUESTA DE LA BANDERA DE INTERRUPCIÓN EN RECEPCIÓN DE DATO
EN MODO SPI
INT_SPI btfss PIR1,SSPIF ;PREGUNTA SI OCURRIÓ INTERRUPCIÓN POR
SPI????
goto RS232_RX ;back
goto SI_SPI
;-----
RS232_RX btfss PIR1,RCIF
goto TIMER_0
goto RX_USART
;-----
TIMER_0 btfss INTCON,2
goto back
goto A_D
;-----
A_D bcf INTCON,2
goto back
;=====
====
RX_USART movf RCREG,w
movwf dato_rx ;RESPALDO DATO EN UN REGISTRO
;-----
; P R I M E R P A S O
    
```

;TESTEO DE RECEPCIÓN DE ASCII PARA DESBLOQUEO DE TODO EL SISTEMA

start_all **movf** dato_rx,w

sublw 'A'

btfss STATUS,2

goto fin_sistema

movlw d'1'

movwf system_start ;SISTEMA DESBLOQUEADO

movlw d'1'

movwf block

;movlw 'H'

;movwf conectado

goto back_rx

;-----

;TESTEO DE RECEPCIÓN DE ASCII PARA BLOQUEO DE TODO EL SISTEMA

fin_sistema **movf** dato_rx,w

sublw 'B'

btfss STATUS,2

goto set_fase

clrf block

;movlw 'I'

;movwf conectado

goto back_rx

;-----

;TESTEO DE RECEPCIÓN DE ASCII PARA CAMBIAR TIEMPO DE FASE

set_fase **movf** dato_rx,w

sublw 'C'

btfss STATUS,2

goto mod_fase ;VA A TESTEAR BANDERA DE FINALIZACION DE FASE

movf block,w

sublw d'1'

```

btfss STATUS,2
goto back_rx
movlw d'1' ;SOLO RESPONDE A CAMBIOS DE TIEMPOS DE FASES
movwf flag_decim
movlw '-'
movwf resp_select
goto back_rx

;-----
;TESTEO DE RECEPCIÓN DE ASCII PARA FINALIZAR LA MODIFICACION DE
FASES
mod_fase movf dato_rx,w
sublw '-'
btfss STATUS,2
goto intermitente
movf block,w
sublw d'1'
btfss STATUS,2
goto back_rx
movf flag_decim,w
sublw d'1'
btfss STATUS,2
goto back_rx
goto ok_dato_fase

;-----
;TESTEO DE RECEPCIÓN DE ASCII PARA CAMBIAR TIEMPO NOCTURNO Y
DIURNO
intermitente movf dato_rx,w
sublw 'D'
btfss STATUS,2
goto end_interm ;VA A TESTEAR BANDERA DE FINALIZACION DE FASE
movf block,w

```

```

sublw d'1'
btfss STATUS,2
goto back_rx
movlw d'1' ;SOLO RESPONDE A CAMBIOS DE TIEMPOS DE FASES
movwf flag_decim
movlw 0x72
movwf select_direc
movlw '!'
movwf resp_select
goto back_rx
;-----
;TESTEO DE RECEPCIÓN DE ASCII PARA FINALIZAR LA MODIFICACION DE
FASES
end_interm movf dato_rx,w
sublw '!'
btfss STATUS,2
goto mod_clk
movf block,w
sublw d'1'
btfss STATUS,2
goto back_rx
movf flag_decim,w
sublw d'1'
btfss STATUS,2
goto back_rx
goto ok_dato_save
;-----
---
;TESTEO DE RECEPCIÓN DE ASCII PARA MODIFICAR RELOJ/CALEDARIO
mod_clk movf dato_rx,w
sublw 'E'

```

```

btfss STATUS,2
goto end_mod_clk
movf block,w
sublw d'1'
btfss STATUS,2
goto back_rx
movlw d'1'
movwf flag_decim
movwf select_direc
movlw '/'
movwf resp_select
goto back_rx
;-----
;TESTEO DE RECEPCIÓN DE ASCII PARA FINALIZAR LA MODIFICACIÓN DE
RELOJ CALENDARIO
end_mod_clk movf dato_rx,w
sublw '/'
btfss STATUS,2
goto hora_pico
movf block,w
sublw d'1'
btfss STATUS,2
goto back_rx
movf flag_decim,w
sublw d'1'
btfss STATUS,2
goto back_rx
goto ok_dato_save;ok_dato_clk
;-----
;TESTEO DE RECEPCIÓN DE ASCII PARA MODIFICAR TIEMPOS DE HORAS
PICO

```

```

hora_pico movf dato_rx,w
sublw 'F'
btfss STATUS,2
goto end_hora_pico
movf block,w
sublw d'1'
btfss STATUS,2
goto back_rx
movlw d'1'
movwf flag_decim
movlw 0x5a
movwf select_direc
movlw '*'
movwf resp_select
goto back_rx
;-----
;TESTEO DE RECEPCIÓN DE ASCII PARA FINALIZAR TIEMPOS DE HORAS
PICO
end_hora_pico movf dato_rx,w
sublw '*'
btfss STATUS,2
goto mod_time_fase_pico
movf block,w
sublw d'1'
btfss STATUS,2
goto back_rx
movf flag_decim,w
sublw d'1'
btfss STATUS,2
goto back_rx
goto ok_dato_save

```

```

;_____
;TESTEO DE DIRECCIÓN PARA GUARDAR TIEMPO DE FASE EN
MAÑANA,TARDE,NOCHE ;EN HORAS PICO
ok_dato_banco1
bsf STATUS,5 ; BANCO 1
movf select_direc,w
movwf FSR ;APUNTA A LA DIRECCIÓN
movf cont_b1,w
addwf FSR,1
bcf STATUS,5 ;BANCO 0
movf resp_num,w
bsf STATUS,5 ; BANCO 1
movwf INDF ;GRABA RESULTADO
;_____
bcf STATUS,5 ;BANCO0
movf resp_select,w
sublw '+'
btfss STATUS,2
goto numeral
goto acum_timepico
numeral movf resp_select,w
sublw '#'
btfss STATUS,2
goto back_rx
goto acum_conteo_carros
;_____
acum_timepico
movf sem_ok,w
sublw 'a'
btfss STATUS,2
goto acum_fase_4v

```

```

goto acum_fase_2v ;ES INTERCECCIÓN DE UNA VÍA
;_____
;PREGUNTO SI ES ULTIMA FASE
acum_fase_4v bsf STATUS,5 ;BANCO1
movf cont_b1,w
sublw d'23'
btfss STATUS,2
goto add_dir_b1 ;INCREMENTA A SIGUIENTE DIRECCIÓN
goto clear_dir_b1
;_____
;LA CALLE ES DE UNA VÍA;LA DIRECCIÓN INCREMENTA CON UN VALOR
DE 4
acum_fase_2v bsf STATUS,5 ;BANCO1
movf cont_b1,w
sublw d'22'
btfss STATUS,2
goto add_fase_pico_2v ;INCREMENTA A SIGUIENTE DIRECCIÓN
goto clear_dir_b1
;_____
add_fase_pico_2v
;incf cont_b1,1
incf cont_b1,1
goto add_dir_b1
;_____
clear_dir_b1 clrf cont_b1
bcf STATUS,5 ;BANCO0
clrf num_ok
clrf resp_num
movlw d'9'
movwf diez
clrf flag_decim

```

```

goto back_rx
;_____
add_dir_b1
incf cont_b1,1
bcf STATUS,5 ;BANCO0
clrf num_ok
clrf resp_num
movlw d'9'
movwf diez
goto back_rx
;_____
;PREGUNTO SI ES ULTIMA CALLE
acum_conteo_carros movf sem_ok,w
sublw 'a'
btfss STATUS,2
goto acum_conteo_4v
goto acum_conteo_2v
acum_conteo_4v bsf STATUS,5 ;BANCO1
movf cont_b1,w
sublw d'31'
btfss STATUS,2
goto add_dir_b1 ;INCREMENTA A SIGUIENTE DIRECCIÓN
goto clear_dir_b1
acum_conteo_2v bsf STATUS,5 ;BANCO1
movf cont_b1,w
sublw d'15'
btfss STATUS,2
goto add_dir_b1 ;INCREMENTA A SIGUIENTE DIRECCIÓN
goto clear_dir_b1
;_____
;_____

```

;GUARDA EL NUMERO CONVERTIDO EN DECIMAL EN DIA

save_day **movf** day,w

call convierte ;SEPARA LA UNIDAD Y DECENA DEL NUMERO DECIMAL

movlw 0x86 ;CARGA DIRECCIÓN DE PARTE ALTA

movwf decena

movlw 0x87 ;CARGA DIRECCIÓN DE PARTE BAJA

movwf unidad

call write_data ;ESCRIBE UN NUMERO DECIMAL A LCD

return

;GUARDA EL NUMERO CONVERTIDO EN DECIMAL EN MES

save_mes **movf** mes,w

call convierte ;SEPARA LA UNIDAD Y DECENA DEL NUMERO DECIMAL

movlw 0x89 ;CARGA DIRECCIÓN DE PARTE ALTA

movwf decena

movlw 0x8a ;CARGA DIRECCIÓN DE PARTE BAJA

movwf unidad

call write_data ;ESCRIBE UN NUMERO DECIMAL A LCD

return

;GUARDA EL NUMERO CONVERTIDO EN DECIMAL EN AÑO

save_year **movf** year,w

call convierte ;SEPARA LA UNIDAD Y DECENA DEL NUMERO DECIMAL

movlw 0x8e ;CARGA DIRECCIÓN DE PARTE ALTA

movwf decena

movlw 0x8f ;CARGA DIRECCIÓN DE PARTE BAJA

movwf unidad

call write_data ;ESCRIBE UN NUMERO DECIMAL A LCD

return

```

;GUARDA EL NUMERO CONVERTIDO EN DECIMAL EN HORA
save_hora movf hora,w
call convierte ;SEPARA LA UNIDAD Y DECENA DEL NUMERO DECIMAL
movlw 0xc8 ;CARGA DIRECCIÓN DE PARTE ALTA
movwf decena
movlw 0xc9 ;CARGA DIRECCIÓN DE PARTE BAJA
movwf unidad
call write_data ;ESCRIBE UN NUMERO DECIMAL A LCD
return

```

```

;GUARDA EL NUMERO CONVERTIDO EN DECIMAL EN MINUTO
save_minuto movf minuto,w
call convierte ;SEPARA LA UNIDAD Y DECENA DEL NUMERO DECIMAL
movlw 0xcb ;CARGA DIRECCIÓN DE PARTE ALTA
movwf decena
movlw 0xcc ;CARGA DIRECCIÓN DE PARTE BAJA
movwf unidad
call write_data ;ESCRIBE UN NUMERO DECIMAL A LCD
return

```

```

;GUARDA EL NUMERO CONVERTIDO EN DECIMAL EN SEGUNDERO
save_second movf second,w
call convierte ;SEPARA LA UNIDAD Y DECENA DEL NUMERO DECIMAL
movlw 0xce ;CARGA DIRECCIÓN DE PARTE ALTA
movwf decena
movlw 0xcf ;CARGA DIRECCIÓN DE PARTE BAJA
movwf unidad
call write_data ;ESCRIBE UN NUMERO DECIMAL A LCD
return

```

```

;-----
;LIMPIA LA BANDERA DE RECEPCIÓN
back_rx bcf PIR1,RCIF
goto back
;=====
=====
back_spi bcf PIR1,SSPIF ; LIMPIA LA BANDERA DE INTERRUPCIÓN DE SPI
goto back
;=====
=====
back_rb0 bcf INTCON,1 ;LIMPIA LA BANDERA DE LA INTERRUPCIÓN RB0
goto back
;=====
=====
back_t1 bcf PIR1,0 ;LIMPIA BANDERA DE TIMER 1
goto back
;=====
=====
back movf fsr_temp,w
movwf FSR ;restituye FSR
swapf stat_temp,w
movwf STATUS ;restituye STATUS
swapf wtemp,f
swapf wtemp,w ;restituye W sin afectar las banderas del STATUS.
;la instrucción "swapf f ", no afecta el contenido del
STATUS
Retfie

```

CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE

ANEXO C

Código del programa principal del micro controlador esclavo.

```

;=====
=====
;PROGRAMA PRINCIPAL
;=====
=====
;=====
;PROGRAMA PRINCIPAL
again ;goto again
btfss PORTB,4
goto again
bsf INTCON,5
;=====
;COMPARA EL VALOR DE LAS LUCES ROJAS
;DE LOS SEMAFOROS PRINCIPALES _1Y2 (RA0=BUFFER0)
comp_sp1_rojo ;goto comp_sp1_2v
movf dato_ra0,w
subwf dato_buffer0,w
btfss 3,2
goto go_error_to_master_sp1_rojo ;NO ES IGUAL
goto comp_sp1_amall ;NO EXISTE ERROR
go_error_to_master_sp1_rojo ;ENVIA ASCII A MASTER
movlw 'I'
call envia_master
movf dato_buffer0,w

```

```

call envia_master
goto comp_sp1_amall
;=====
;COMPARA EL VALOR DE LAS LUCES AMARILLAS
;DE LOS SEMAFOROS PRINCIPALES _1Y2 (RA1=BUFFER1)
comp_sp1_amall
movf dato_ra1,w
subwf dato_buffer1,w
btfss 3,2
goto go_error_to_master_sp1_amall
goto comp_sp1_verd
go_error_to_master_sp1_amall
movlw 'J'
call envia_master
movf dato_buffer1,w
call envia_master
goto comp_sp1_verd
;=====
;COMPARA EL VALOR DE LAS LUCES VERDES
;DE LOS SEMAFOROS PRINCIPALES _1Y2 (RA2=BUFFER2)
comp_sp1_verd
movf dato_ra2,w
subwf dato_buffer2,w
btfss 3,2
goto go_error_to_master_sp1_verd
goto comp_ss1_rojo
go_error_to_master_sp1_verd
movlw 'K'
call envia_master
movf dato_buffer2,w
call envia_master

```

```

goto comp_ss1_rojo
;=====
;COMPARA EL VALOR DE LAS LUCES ROJAS
;DE LOS SEMAFOROS SECUNDARIOS_1Y2 (RA5=BUFFER5)
comp_ss1_rojo
movf dato_ra5,w
subwf dato_buffer5,w
btfss 3,2
goto go_error_to_master_ss1_rojo
goto comp_ss1_amall
go_error_to_master_ss1_rojo
movlw 'L'
call envia_master
movf dato_buffer5,w
call envia_master
goto comp_ss1_amall
;=====
;COMPARA EL VALOR DE LAS LUCES AMARILLAS
;DE LOS SEMAFOROS SECUNDARIOS_1Y2 (RE0=BUFFER6)
comp_ss1_amall
movf dato_re0,w
subwf dato_buffer6,w
btfss 3,2
goto go_error_to_master_ss1_amall
goto comp_ss1_verd
go_error_to_master_ss1_amall
movlw 'M'
call envia_master
movf dato_buffer6,w
call envia_master
goto comp_ss1_verd

```

```

;=====
;COMPARA EL SEMÁFORO DE VIRAJE PRINCIPAL DE SUR A
NORTE(RE0=BUFFER5)
comp_ss1_verd
movf dato_re1,w
subwf dato_buffer7,w
btfss 3,2
goto go_error_to_master_ss1_verd
goto comp_sp1_rojo
go_error_to_master_ss1_verd
movlw 'N'
call envia_master
movf dato_buffer7,w
call envia_master
goto comp_sp1_rojo
;=====
=
envia_master return;bcf PIR1,SSPIF;return
btfsc PIR1,SSPIF
goto envia_master
movwf SSPBUF
bcf PIR1,SSPIF
call dem_100ms
call dem_100ms
return
;=====
=====
;=====
=====
;ACTIVACIÓN DE FASE PARA EL ENCENDIDO DE LAS LÁMPARAS
fase_1 movlw b'01001100' ;ENCIENDE VERDE DE SEMÁFORO PRINCIPAL

```

```

movwf PORTD ;ROJO SECUNDARIA Y PEATONAL PRINCIPAL
bcf PORTC,0 ;APAGADO ROJO PEATÓN SECUNDARIA
bsf PORTC,1 ;ENCENDIDO VERDE PEATÓN SECUNDARIA
bsf PORTC,2 ;ENCENDIDO ROJO VIRAJE PRINCIPAL
bcf PORTC,6 ;APAGADO VERDE VIRAJE PRINCIPAL
bsf PORTC,7 ;ENCENDIDO ROJO VIRAJE SECUNDARIA
bcf PORTB,1 ;APAGADO VERDE VIRAJE SECUNDARIA
movlw 'm'
movwf dato_ra0 ;ROJOS DE SEMÁFOROS PRINCIPALES_1Y2
movlw 'm'
movwf dato_ra1 ;AMARILLOS DE SEMÁFOROS PRINCIPALES_1Y2
movlw 'p'
movwf dato_ra2 ;VERDES DE SEMÁFOROS PRINCIPALES_1Y2
movlw 'p'
movwf dato_ra5 ;ROJOS DE SEMÁFOROS SECUNDARIOS_1Y2
movlw 'm'
movwf dato_re0 ;AMARILLOS DE SEMÁFOROS SECUNDARIOS_1Y2
movlw 'm'
movwf dato_re1 ;VERDES DE SEMÁFOROS SECUNDARIOS_1Y2
return
;-----
fase_2 movlw b'01001010' ;ENCIENDE AMARILLO DE SEMÁFORO PRINCIPAL
movwf PORTD ;ROJO SECUNDARIA Y PEATONAL PRINCIPAL
bcf PORTC,0 ;APAGADO ROJO PEATÓN SECUNDARIA
bsf PORTC,1 ;ENCENDIDO VERDE PEATÓN SECUNDARIA
bsf PORTC,2 ;ENCENDIDO ROJO VIRAJE PRINCIPAL
bcf PORTC,6 ;APAGADO VERDE VIRAJE PRINCIPAL
bsf PORTC,7 ;ENCENDIDO ROJO VIRAJE SECUNDARIA
bcf PORTB,1 ;APAGADO VERDE VIRAJE SECUNDARIA
movlw 'm'
movwf dato_ra0 ;ROJOS DE SEMÁFOROS PRINCIPALES_1Y2

```

```

movlw 'p'
movwf dato_ra1 ;AMARILLOS DE SEMÁFOROS PRINCIPALES _1Y2
movlw 'm'
movwf dato_ra2 ;VERDES DE SEMÁFOROS PRINCIPALES _1Y2
movlw 'p'
movwf dato_ra5 ;ROJOS DE SEMÁFOROS SECUNDARIOS _1Y2
movlw 'm'
movwf dato_re0 ;AMARILLOS DE SEMÁFOROS SECUNDARIOS _1Y2
movlw 'm'
movwf dato_re1 ;VERDES DE SEMÁFOROS SECUNDARIOS _1Y2
return
;-----
fase_3 movlw b'01001001' ;ENCIENDE ROJO DE SEMÁFORO PRINCIPAL
movwf PORTD ;ROJO SECUNDARIA Y PEATONAL PRINCIPAL
bsf PORTC,0 ;ENCENDIDO ROJO PEATÓN SECUNDARIA
bcf PORTC,1 ;APAGADO VERDE PEATÓN SECUNDARIA
bcf PORTC,2 ;APAGADO ROJO VIRAJE PRINCIPAL
bsf PORTC,6 ;ENCENDIDO VERDE VIRAJE PRINCIPAL
bsf PORTC,7 ;ENCENDIDO ROJO VIRAJE SECUNDARIA
; bcf PORTB,1 ;APAGADO VERDE VIRAJE SECUNDARIA
movlw 'p'
movwf dato_ra0 ;ROJOS DE SEMÁFOROS PRINCIPALES _1Y2
movlw 'm'
movwf dato_ra1 ;AMARILLOS DE SEMÁFOROS PRINCIPALES _1Y2
movlw 'm'
movwf dato_ra2 ;VERDES DE SEMÁFOROS PRINCIPALES _1Y2
movlw 'p'
movwf dato_ra5 ;ROJOS DE SEMÁFOROS SECUNDARIOS _1Y2
movlw 'm'
movwf dato_re0 ;AMARILLOS DE SEMÁFOROS SECUNDARIOS _1Y2
movlw 'm'

```

```

movwf dato_re1 ;VERDES DE SEMÁFOROS SECUNDARIOS_1Y2
return
;-----
fase_4 movlw b'01001001' ;ENCIENDE ROJO DE SEMÁFORO PRINCIPAL
movwf PORTD ;ROJO SECUNDARIA Y PEATONAL PRINCIPAL
bsf PORTC,0 ;ENCENDIDO ROJO PEATÓN SECUNDARIA
bcf PORTC,1 ;APAGADO VERDE PEATÓN SECUNDARIA
bcf PORTC,2 ;ENCENDIDO ROJO VIRAJE PRINCIPAL
bcf PORTC,6 ;APAGADO VERDE VIRAJE PRINCIPAL
bsf PORTC,7 ;ENCENDIDO ROJO VIRAJE SECUNDARIA
;_____
movlw 'p'
movwf dato_ra0 ;ROJOS DE SEMÁFOROS PRINCIPALES_1Y2
movlw 'm'
movwf dato_ra1 ;AMARILLOS DE SEMÁFOROS PRINCIPALES _1Y2
movlw 'm'
movwf dato_ra2 ;VERDES DE SEMÁFOROS PRINCIPALES_1Y2
movlw 'p'
movwf dato_ra5 ;ROJOS DE SEMÁFOROS SECUNDARIOS_1Y2
movlw 'm'
movwf dato_re0 ;AMARILLOS DE SEMÁFOROS SECUNDARIOS_1Y2
movlw 'm'
movwf dato_re1 ;VERDES DE SEMÁFOROS SECUNDARIOS_1Y2
return
;-----
fase_5 movlw b'10100001' ;ENCIENDE ROJO DE SEMÁFORO PRINCIPAL
movwf PORTD ;VERDE SECUNDARIA Y PEATONAL PRINCIPAL
bsf PORTC,0 ;ENCENDIDO ROJO PEATÓN SECUNDARIA
bcf PORTC,1 ;APAGADO VERDE PEATÓN SECUNDARIA
bsf PORTC,2 ;ENCENDIDO ROJO VIRAJE PRINCIPAL
bcf PORTC,6 ;APAGADO VERDE VIRAJE PRINCIPAL

```

bsf PORTC,7 ;ENCENDIDO ROJO VIRAJE SECUNDARIA

;

movlw 'p'

movwf dato_ra0 ;ROJOS DE SEMÁFOROS PRINCIPALES_1Y2

movlw 'm'

movwf dato_ra1 ;AMARILLOS DE SEMÁFOROS PRINCIPALES _1Y2

movlw 'm'

movwf dato_ra2 ;VERDES DE SEMÁFOROS PRINCIPALES_1Y2

movlw 'm'

movwf dato_ra5 ;ROJOS DE SEMÁFOROS SECUNDARIOS_1Y2

movlw 'm'

movwf dato_re0 ;AMARILLOS DE SEMÁFOROS SECUNDARIOS_1Y2

movlw 'p'

movwf dato_re1 ;VERDES DE SEMÁFOROS SECUNDARIOS_1Y2

return

;

fase_6 **movlw** b'10010001' ;ENCIENDE ROJO DE SEMÁFORO PRINCIPAL

movwf PORTD ;ENCIENDE AMARILLA SECUNDARIA Y ROJO PEATONAL
PRINCIPAL

bsf PORTC,0 ;ENCENDIDO ROJO PEATÓN SECUNDARIA

bcf PORTC,1 ;APAGADO VERDE PEATÓN SECUNDARIA

bsf PORTC,2 ;ENCENDIDO ROJO VIRAJE PRINCIPAL

bcf PORTC,6 ;APAGADO VERDE VIRAJE PRINCIPAL

bsf PORTC,7 ;ENCENDIDO ROJO VIRAJE SECUNDARIA

;

movlw 'p'

movwf dato_ra0 ;ROJOS DE SEMÁFOROS PRINCIPALES_1Y2

movlw 'm'

movwf dato_ra1 ;AMARILLOS DE SEMÁFOROS PRINCIPALES _1Y2

movlw 'm'

movwf dato_ra2 ;VERDES DE SEMÁFOROS PRINCIPALES_1Y2

```

movlw 'm'
movwf dato_ra5 ;ROJOS DE SEMÁFOROS SECUNDARIOS_1Y2
movlw 'p'
movwf dato_re0 ;AMARILLOS DE SEMÁFOROS SECUNDARIOS_1Y2
movlw 'm'
movwf dato_re1 ;VERDES DE SEMÁFOROS SECUNDARIOS_1Y2
return
;-----
fase_7 movlw b'01001001' ;ENCIENDE ROJO DE SEMÁFORO PRINCIPAL
movwf PORTD ;ENCIENDE ROJO SECUNDARIA Y PEATONAL PRINCIPAL
bsf PORTC,0 ;ENCENDIDO ROJO PEATÓN SECUNDARIA
bcf PORTC,1 ;APAGADO VERDE PEATÓN SECUNDARIA
bsf PORTC,2 ;ENCENDIDO ROJO VIRAJE PRINCIPAL
bcf PORTC,6 ;APAGADO VERDE VIRAJE PRINCIPAL
bcf PORTC,7 ;ENCENDIDO ROJO VIRAJE SECUNDARIA
bsf PORTB,1 ;APAGADO VERDE VIRAJE SECUNDARIA
;-----
movlw 'p'
movwf dato_ra0 ;ROJOS DE SEMÁFOROS PRINCIPALES_1Y2
movlw 'm'
movwf dato_ra1 ;AMARILLOS DE SEMÁFOROS PRINCIPALES_1Y2
movlw 'm'
movwf dato_ra2 ;VERDES DE SEMÁFOROS PRINCIPALES_1Y2
movlw 'p'
movwf dato_ra5 ;ROJOS DE SEMÁFOROS SECUNDARIOS_1Y2
movlw 'm'
movwf dato_re0 ;AMARILLOS DE SEMÁFOROS SECUNDARIOS_1Y2
movlw 'm'
movwf dato_re1 ;VERDES DE SEMÁFOROS SECUNDARIOS_1Y2
return
;-----

```

```

fase_8 movlw b'01001001' ;ENCIENDE ROJO DE SEMÁFORO PRINCIPAL
movwf PORTD ;ENCIENDE ROJO SECUNDARIA Y PEATONAL PRINCIPAL
bsf PORTC,0 ;ENCENDIDO ROJO PEATÓN SECUNDARIA
bcf PORTC,1 ;APAGADO VERDE PEATÓN SECUNDARIA
bsf PORTC,2 ;ENCENDIDO ROJO VIRAJE PRINCIPAL
bcf PORTC,6 ;APAGADO VERDE VIRAJE PRINCIPAL
bcf PORTC,7 ;ENCENDIDO ROJO VIRAJE SECUNDARIA
bcf PORTB,1 ;APAGADO VERDE VIRAJE SECUNDARIA
;-----
movlw 'p'
movwf dato_ra0 ;ROJOS DE SEMÁFOROS PRINCIPALES_1Y2
movlw 'm'
movwf dato_ra1 ;AMARILLOS DE SEMÁFOROS PRINCIPALES _1Y2
movlw 'm'
movwf dato_ra2 ;VERDES DE SEMÁFOROS PRINCIPALES_1Y2
movlw 'p'
movwf dato_ra5 ;ROJOS DE SEMÁFOROS SECUNDARIOS_1Y2
movlw 'm'
movwf dato_re0 ;AMARILLOS DE SEMÁFOROS SECUNDARIOS_1Y2
movlw 'm'
movwf dato_re1 ;VERDES DE SEMÁFOROS SECUNDARIOS_1Y2
return
return
;-----
;TODAS LAS LUCES DE LOS SEMÁFOROS APAGADOS
fase_off clrf PORTD
bcf PORTC,0 ;APAGADO ROJO PEATÓN SECUNDARIA
bcf PORTC,1 ;APAGADO VERDE PEATÓN SECUNDARIA
bcf PORTC,2 ;APAGADO ROJO VIRAJE PRINCIPAL
bcf PORTC,6 ;APAGADO VERDE VIRAJE PRINCIPAL
bcf PORTC,7 ;APAGADO ROJO VIRAJE SECUNDARIA

```

```

bcf PORTB,1 ;APAGADO VERDE VIRAJE SECUNDARIA
return
;-----
;TODAS LAS LUCES DE LOS SEMÁFOROS ENCENDIDAS
fase_dañado_on movlw b'11111111'
movwf PORTD
bsf PORTC,0 ;ENCENDIDO ROJO PEATÓN SECUNDARIA
bsf PORTC,1 ;ENCENDIDO VERDE PEATÓN SECUNDARIA
bsf PORTC,2 ;ENCENDIDO ROJO VIRAJE PRINCIPAL
bsf PORTC,6 ;ENCENDIDO VERDE VIRAJE PRINCIPAL
bsf PORTC,7 ;ENCENDIDO ROJO VIRAJE SECUNDARIA
bsf PORTB,1 ;ENCENDIDO VERDE VIRAJE SECUNDARIA
return
;-----
fase_1_2v movlw b'01001100' ;ENCIENDE VERDE DE SEMÁFORO PRINCIPAL
movwf PORTD ;ROJO SECUNDARIA Y PEATONAL PRINCIPAL
bcf PORTC,0 ;APAGADO ROJO PEATÓN SECUNDARIA
bsf PORTC,1 ;ENCENDIDO VERDE PEATÓN SECUNDARIA
bcf PORTC,2 ;ENCENDIDO ROJO VIRAJE PRINCIPAL
bcf PORTC,6 ;APAGADO VERDE VIRAJE PRINCIPAL
bcf PORTC,7 ;ENCENDIDO ROJO VIRAJE SECUNDARIA
bcf PORTB,1 ;APAGADO VERDE VIRAJE SECUNDARIA
movlw 'm'
movwf dato_ra0 ;ROJOS DE SEMÁFOROS PRINCIPALES_1Y2
movlw 'm'
movwf dato_ra1 ;AMARILLOS DE SEMÁFOROS PRINCIPALES_1Y2
movlw 'p'
movwf dato_ra2 ;VERDES DE SEMÁFOROS PRINCIPALES_1Y2
movlw 'p'
movwf dato_ra5 ;ROJOS DE SEMÁFOROS SECUNDARIOS_1Y2
movlw 'm'

```

```

movwf dato_re0 ;AMARILLOS DE SEMÁFOROS SECUNDARIOS _1Y2
movlw 'm'
movwf dato_re1 ;VERDES DE SEMÁFOROS SECUNDARIOS_1Y2
return
;-----
fase_2_2v movlw b'01001010' ;ENCIENDE AMARILLO DE SEMÁFORO
PRINCIPAL
movwf PORTD ;ROJO SECUNDARIA Y PEATONAL PRINCIPAL
bcf PORTC,0 ;APAGADO ROJO PEATÓN SECUNDARIA
bsf PORTC,1 ;ENCENDIDO VERDE PEATÓN SECUNDARIA
bcf PORTC,2 ;ENCENDIDO ROJO VIRAJE PRINCIPAL
bcf PORTC,6 ;APAGADO VERDE VIRAJE PRINCIPAL
bcf PORTC,7 ;ENCENDIDO ROJO VIRAJE SECUNDARIA
bcf PORTB,1 ;APAGADO VERDE VIRAJE SECUNDARIA
movlw 'm'
movwf dato_ra0 ;ROJOS DE SEMÁFOROS PRINCIPALES_1Y2
movlw 'p'
movwf dato_ra1 ;AMARILLOS DE SEMÁFOROS PRINCIPALES _1Y2
movlw 'm'
movwf dato_ra2 ;VERDES DE SEMÁFOROS PRINCIPALES_1Y2
movlw 'p'
movwf dato_ra5 ;ROJOS DE SEMÁFOROS SECUNDARIOS_1Y2
movlw 'm'
movwf dato_re0 ;AMARILLOS DE SEMÁFOROS SECUNDARIOS _1Y2
movlw 'm'
movwf dato_re1 ;VERDES DE SEMÁFOROS SECUNDARIOS_1Y2
return
;-----
fase_3_2v movlw b'10100001' ;ENCIENDE ROJO DE SEMÁFORO PRINCIPAL
movwf PORTD ;VERDE SECUNDARIA Y PEATONAL PRINCIPAL
bsf PORTC,0 ;ENCENDIDO ROJO PEATÓN SECUNDARIA

```

```

bcf PORTC,1 ;APAGADO VERDE PEATÓN SECUNDARIA
bcf PORTC,2 ;ENCENDIDO ROJO VIRAJE PRINCIPAL
bcf PORTC,6 ;APAGADO VERDE VIRAJE PRINCIPAL
bcf PORTC,7 ;ENCENDIDO ROJO VIRAJE SECUNDARIA
bcf PORTB,1 ;APAGADO VERDE VIRAJE SECUNDARIA
movlw 'p'
movwf dato_ra0 ;ROJOS DE SEMÁFOROS PRINCIPALES_1Y2
movlw 'm'
movwf dato_ra1 ;AMARILLOS DE SEMÁFOROS PRINCIPALES _1Y2
movlw 'm'
movwf dato_ra2 ;VERDES DE SEMÁFOROS PRINCIPALES_1Y2
movlw 'm'
movwf dato_ra5 ;ROJOS DE SEMÁFOROS SECUNDARIOS_1Y2
movlw 'm'
movwf dato_re0 ;AMARILLOS DE SEMÁFOROS SECUNDARIOS _1Y2
movlw 'p'
movwf dato_re1 ;VERDES DE SEMÁFOROS SECUNDARIOS_1Y2
return
;-----
fase_4_2v movlw b'10010001' ;ENCIENDE ROJO DE SEMÁFORO PRINCIPAL
movwf PORTD ;ENCIENDE AMARILLA SECUNDARIA Y ROJO PEATONAL
PRINCIPAL
bsf PORTC,0 ;ENCENDIDO ROJO PEATÓN SECUNDARIA
bcf PORTC,1 ;APAGADO VERDE PEATÓN SECUNDARIA
bcf PORTC,2 ;ENCENDIDO ROJO VIRAJE PRINCIPAL
bcf PORTC,6 ;APAGADO VERDE VIRAJE PRINCIPAL
bcf PORTC,7 ;ENCENDIDO ROJO VIRAJE SECUNDARIA
bcf PORTB,1 ;APAGADO VERDE VIRAJE SECUNDARIA
movlw 'p'
movwf dato_ra0 ;ROJOS DE SEMÁFOROS PRINCIPALES_1Y2
movlw 'm'

```

```

movwf dato_ra1 ;AMARILLOS DE SEMÁFOROS PRINCIPALES _1Y2
movlw 'm'
movwf dato_ra2 ;VERDES DE SEMÁFOROS PRINCIPALES_1Y2
movlw 'm'
movwf dato_ra5 ;ROJOS DE SEMÁFOROS SECUNDARIOS_1Y2
movlw 'p'
movwf dato_re0 ;AMARILLOS DE SEMÁFOROS SECUNDARIOS _1Y2
movlw 'm'
movwf dato_re1 ;VERDES DE SEMÁFOROS SECUNDARIOS_1Y2
return
fase_interm_on
movlw b'01001010' ;ENCIENDE VERDE DE SEMÁFORO PRINCIPAL
movwf PORTD ;ROJO SECUNDARIA Y PEATONAL PRINCIPAL
bsf PORTC,0 ;APAGADO ROJO PEATÓN SECUNDARIA
bcf PORTC,1 ;APAGADO VERDE PEATÓN SECUNDARIA
bsf PORTC,2 ;ENCENDIDO ROJO VIRAJE PRINCIPAL
bcf PORTC,6 ;APAGADO VERDE VIRAJE PRINCIPAL
bsf PORTC,7 ;ENCENDIDO ROJO VIRAJE SECUNDARIA
bcf PORTB,1 ;APAGADO VERDE VIRAJE SECUNDARIA
return

```

CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE

ANEXO D

Código de programa de atención a interrupciones generales del micro controlador esclavo.

```

=====
===
; SUBROUTINA DE INTERUPCION
=====
===
IT ;Respalda registros
movwf wtemp ;RESPALDO AC
swapf STATUS,w
movwf stat_temp
movf FSR,0
movwf fsr_temp ;RESPALDO EL FSR
;-----
;-----
;INICIA TESTEO DE INTERRUPCIONES MEDIANTE ENCUESTA DE
BANDERAS
btfsc INTCON,0
goto SI_PORTB
btfsc INTCON,2
goto TIMER_0
; btfsc PIR1,SSPIF
; goto spi_go
btfss PIR1,ADIF

```

```

goto back
;-----
;-----
;FIN DE CONVERSIÓN POR INTERRUPCIÓN DEL CONVERSOR A/D
;GUARDA VALOR DE CONVERSIÓN EN BUFFER SELECCIONADO.
A_D ;goto back_ad
movlw 0x2C ;INICIO CON UNA DIRECCIÓN.
movwf FSR ;APUNTA A LA DIRECCIÓN.
movf buffer,w ;BUFFER AL INICIO ES CERO.
addwf FSR,1 ;SUMA LA DIRECCIÓN PARA SELECCIONAR EL BUFFER
movf ADRESH,w ;MUEVE EL VALOR CONVERTIDO DEL BUFFER
SELECCIONADO AL
ACUMULADOR
movwf INDF ;GUARDA VALOR CONVERTIDO A LA DIRECCIÓN ACTUAL.
bcf ADCON0,0 ;APAGA CONVERSOR A/D
;VA A PREGUNTAR POR LOS LIMITES DE LOS SENSORES
movf buffer,w ;ESCOGE EL BUFFER PARA TESTEAR LÍMITES
call test_buffers;DE VOLTAJE DE LAS LAMPARAS DE LOS SEMÁFOROS
;-----
movf buffer,w ;VA A PREGUNTAR SI ES EL ÚLTIMO
sublw d'6' ;BUFFER A GUARDAR Y CONVERTIR
btfss 3,2 ;
goto add_buffer ;NO ES EL ÚLTIMO,VA A CAMBIAR DE BUFFER_N
clrf buffer ;SI ES EL ÚLTIMO,INICIA CON BUFFER0
back_ad bcf PIR1,ADIF ;LIMPIA BANDERA DE INTERRUPCIÓN
goto back ;VA A RESTITUCIÓN DE REGISTROS
;-----
add_buffer incf buffer ;INCREMENTA EN UNO AL BUFFER; BUFFER_N
bcf PIR1,ADIF ;LIMPIA BANDERA DE INTERRUPCIÓN
goto back ;VA A RESTITUCIÓN DE REGISTROS
;-----

```

```

spi_go bcf PIR1,SSPIF
goto back
;-----
;CONVERSIÓN DEL CANAL SELECCIONADO
TIMER_0 ;goto back_t0
movf buffer,w ;NUMERO DE BUFFER
call ADvalues ;SELECCIONA EL BUFFER PARA LA CONVERSIÓN
movwf ADCON0 ;CARGA EL # DE BUFFER AL REGISTRO
bsf ADCON0,0 ;GO! INICIA LA CUANTIFICACIÓN
call dem_40us ;TIEMPO DE ADQUISICIÓN
bsf ADCON0,2
goto back_t0 ;VA A LIMPIAR LA BANDERA DEL TIMER 0
;-----
;-----
SI_PORTB
movf PORTB,w ;MUEVO AL ACUMULADOR
movwf registro ;RESPALDO DATO
swapf registro,w ;GIRO EL REGISTRO PARA ESCOGER LOS BITS
movwf new_fase ;RESPALDO DATO
movf new_fase,w
andlw d'15' ;ENMARCARO DATO
movwf new_fase
btfss PORTB,3
goto next
movlw d'16'
addwf new_fase,1
next movf new_fase,w
call tab_fases ;LLAMO A SELECCIONAR EL TIPO DE FASE
; clrf PORTB
goto back_rb
;-----

```

```

back_t0 bcf INTCON,2 ;LIMPIA BANDERA DE INTERRUPCIÓN
CORRESPONDIENTE
goto back ;VA A RESTITUCIÓN DE REGISTROS
back_rb bcf INTCON,0 ;LIMPIA BANDERA DE INTERRUPCIÓN
CORRESPONDIENTE
;-----
;RESTITUCIÓN DE REGISTROS
back movf fsr_temp,w
movwf FSR ;restituye FSR
swapf stat_temp,w
movwf STATUS ;restituye STATUS
swapf wtemp,f ;la instrucción "swapf f ", no afecta el contenido del
STATUS
swapf wtemp,w ;restituye W sin afectar las banderas del STATUS.
retfie

```

CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE

ANEXO E

Código de programa del RTC.

```

DEFINE I2C_SCLOUT 1
CPIN VAR Portb.7
DPIN VAR Portb.6
segu VAR BYTE
minu VAR BYTE
hora VAR BYTE
diaS VAR BYTE
diaF VAR BYTE
mes VAR BYTE
anio VAR BYTE
dato VAR BYTE
actualizado VAR BIT
EEPROM 0,[0]
READ 0,actualizado
IF actualizado =0 THEN grabarRTC
INICIO:
I2CREAD DPIN,CPIN,%11010000,0,[segu]
I2CREAD DPIN,CPIN,%11010000,1,[minu]
I2CREAD DPIN,CPIN,%11010000,2,[hora]
I2CREAD DPIN,CPIN,%11010000,3,[diaS]
I2CREAD DPIN,CPIN,%11010000,4,[diaF]
I2CREAD DPIN,CPIN,%11010000,5,[mes]
I2CREAD DPIN,CPIN,%11010000,6,[anio]

```

PAUSE 500

GOTO INICIO

,******subrutina*

*grabar******

grabarRTC:

I2CWRITE DPIN,CPIN,%11010000,0,[\$00]

PAUSE 10

I2CWRITE DPIN,CPIN,%11010000,1,[\$30]

PAUSE 10

I2CWRITE DPIN,CPIN,%11010000,2,[\$13]

PAUSE 10

I2CWRITE DPIN,CPIN,%11010000,3,[\$2]

PAUSE 10

I2CWRITE DPIN,CPIN,%11010000,4,[\$27]

PAUSE 10

I2CWRITE DPIN,CPIN,%11010000,5,[\$9]

PAUSE 10

I2CWRITE DPIN,CPIN,%11010000,6,[\$04]

PAUSE 10

I2CWRITE DPIN,CPIN,%11010000,7,[\$10]

PAUSE 10

WRITE 0,1

GOTO INICIO

END