

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE PARQUEOS

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO
EN ELECTRÓNICA Y TELECOMUNICACIONES**

GABRIELA FERNANDA GAVILÁNEZ PERRAZO

gabyfer_epn@hotmail.com

TANIA BELÉN TAPE VILAÑA

taniataipevilana@hotmail.com

DIRECTOR: ING. MÓNICA DE LOURDES VINUEZA RHOR

monica.vinueza@epn.edu.ec

Quito, Junio 2016

DECLARACIÓN

Nosotros, Gabriela Fernanda Gavilánez Perrazo y Tania Belén Taipe Vilaña, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Gabriela Fernanda Gavilánez Perrazo

Tania Belén Taipe Vilaña

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Gabriela Fernanda Gavilánez Perrazo y Tania Belén Taipe Vilaña, bajo mi supervisión.

Ing. Mónica Vinueza Rhor
DIRECTOR DEL PROYECTO

AGRADECIMIENTOS

A mi papito Dios que siempre me dio la sabiduría y fuerza para poder culminar una etapa más de vida y emprender nuevos retos. A mi madrecita Gladys Perrazo ya que siempre ha estado en los momentos más difíciles de mi vida entendiéndome y brindándome su apoyo incondicional. A mis hermanos Soraya y Andrés por estar conmigo siempre. A mi padre Luis que me dio su comprensión en todo momento.

A mi verdadera amiga y compañera de tesis por su tiempo y paciencia e igualmente a mi novio Fredy ya que siempre me brindó su ayuda incondicional y finalmente a mi querida tutora que siempre nos dio fuerzas para seguir adelante.

Gabriela Fernanda Gaviláñez Perrazo

AGRADECIMIENTOS

A mi madre quien a pesar de todo lo que hemos pasado se encuentra de pie junto a sus hijos, gracias por esperar pacientemente hasta hoy.

A mi hermana Paulina por estar a mi lado desde que nació. A mis sobrinos por ser la alegría de mi vida y mi fuerza desde que nacieron. A mis amigos, Diana y Diego, por su apoyo y confianza.

Gracias a mi compañera de tesis por su amistad, apoyo y dedicación brindados a lo largo de nuestro camino en la tesis; a los profesores que supieron escucharme y alentarme, entre ellos mi tutora por su paciencia y apoyo.

Sobre todo gracias a Dios por su ayuda en momentos difíciles, de los que salí adelante, y las alegrías vividas.

Tania Belén Taipe Vilaña

DEDICATORIA

A mi mejor amiga incondicional mi mamita Gladys ya que siempre ha estado en mis triunfos y derrotas brindándome su apoyo sin esperar nada a cambio y dándome todo su amor; ya que sin ti nada de esto sería posible porque nadie más que tú has estado pendiente de mí, te amo mamá.

Gabriela Fernanda Gavilánez Perrazo

DEDICATORIA

A mi madrecita por su dedicación y apoyo a sus hijos, y mi querida abuelita que ya no está a mi lado pero fue testigo de mi esfuerzo; este logro es por ustedes. Y a la persona que aunque lejos, fue mi fuerza desde siempre y hasta hoy.

Tania Belén Taipe Vilaña

CONTENIDO

PORTADA.....	i
DECLARACIÓN.....	ii
CERTIFICACIÓN.....	iii
AGRADECIMIENTOS.....	iv
DEDICATORIA.....	vi
CONTENIDO.....	viii
PRESENTACIÓN.....	xvii
RESUMEN.....	xviii

CAPÍTULO I

MARCO TEÓRICO.....	1
1.1. MICROCONTROLADOR.....	1
1.1.1. CARACTERÍSTICAS.....	1
1.1.2. FUNCIONAMIENTO.....	2
1.1.3. MICROPROCESADOR 16F877A.....	2
1.1.3.1. Configuración de pines del microcontrador.....	3
1.1.3.2. Distribución de puertos del microcontrolador.....	4
1.1.3.3. Características principales del PIC 16F877A.....	6
1.1.3.4. Observaciones.....	7
1.2. SENSORES INFRARROJOS.....	7
1.2.1. CARACTERÍSTICAS.....	7
1.2.2. LED INFRARROJO (IRLED).....	8
1.2.2.1. Funcionamiento.....	8
1.2.3. SENSOR QRD1114 (Sensor Objeto Reflectante).....	9

1.2.3.1. Características.....	9
1.2.3.2. Especificaciones.....	9
1.2.3.3. Observaciones.....	10
1.3. LEDS Y DISPLAYS DE 7 SEGMENTOS.....	10
1.3.1. DISPLAYS DE 7 SEGMENTOS.....	10
1.3.1.1. Definición.....	10
1.3.1.2. Funcionamiento.....	11
1.3.1.3. Display ánodo común.....	12
1.3.1.4. Observaciones.....	12
1.3.2. LEDS (diodo emisor de luz).....	12
1.3.2.1. Definición.....	12
1.3.2.2. Leds de alta intensidad.....	13
1.3.2.3. Observaciones.....	14
1.4. RESISTENCIAS, CAPACITORES Y REGULADOR DE	14
VOLTAJE.....	
1.4.1. RESISTENCIAS.....	14
1.4.1.1. Características Específicas.....	15
1.4.1.2. Funcionamiento.....	15
1.4.1.3. Observaciones.....	16
1.4.2. CAPACITORES.....	16
1.4.2.1. Definición.....	16
1.4.2.2. Observaciones.....	17
1.4.3. REGULADOR DE VOLTAJE.....	17
1.4.3.1. Definición.....	17
1.4.3.2. Funcionamiento.....	17
1.4.3.3. Regulador de voltaje L7805.....	17

1.4.3.4. Descripción de pines y funcionamiento.....	18
1.4.3.5. Regulador de voltaje externo.....	19
1.4.3.4.1. Componentes.....	19
1.4.3.4.2. Cálculos.....	20
1.5. PROGRAMA ENSAMBLADOR MIKROC.....	24
1.5.1. CARACTERÍSTICAS PRINCIPALES DE MIKROC.....	24
1.5.1.1. Fase de Compilación.....	25
1.5.1.2. Estructura de Programa.....	26
1.5.1.3. Comentarios.....	26
1.5.1.4. Proceso para la programación.....	27
1.5.1.5. Descripción del desarrollo del programa.....	28
CAPÍTULO II	
DISEÑO.....	29
2.1. DISEÑO DEL SISTEMA DE CONTROL DE PARQUEOS.....	29
2.1.1. DIAGRAMA DE BLOQUES DEL SISTEMA.....	29
2.1.2. DIAGRAMA DE FLUJO.....	32
2.1.3. DIAGRAMA DE FLUJO A DETALLE.....	34
2.1.4. CODIFICACIÓN DEL PROGRAMA EN EL MICRO.....	36
2.1.5. DIAGRAMA CIRCUITAL DEL SISTEMA DE CONTROL DE PARQUEOS.....	40
2.1.5.1. Funcionamiento electrónico del estacionamiento.....	42
2.2. PROGRAMACIÓN DEL SISTEMA DE CONTROL DE PARQUEOS.....	42
2.2.1. TABLA DE VERDAD DEL PARQUEADERO.....	42
2.2.2. PROGRAMACIÓN.....	45
2.2.2.1. PROGRAMACIÓN FINAL: 4 PARQUEADEROS.....	45
2.2.3. SIMULACIÓN DEL SISTEMA DE CONTROL DE PARQUEOS.....	48

2.2.3.1. PROGRAMA FINAL: 4 PARQUEADEROS.....	48
CAPÍTULO III	
IMPLEMENTACIÓN.....	53
3.1. ELABORACIÓN DEL PROTOTIPO DEL SISTEMA DE CONTROL DE PARQUEOS.....	53
3.1.1. PROTOTIPO IMPLEMENTADO EN PROTOBOARD.....	53
3.1.1.1. Ubicación del Regulador de Voltaje.....	53
3.1.1.2. Instalación de los sensores.....	54
3.1.1.3. Conexión del Microcontrolador y Displays.....	57
3.1.1.4. Pruebas de Funcionamiento del Prototipo Implementado en el Protoboard.....	61
3.1.2. DISEÑO DE LA ESTRUCTURA Y DISTRIBUCIÓN DEL PROTOTIPO.....	63
3.2. CONSTRUCCIÓN DE LOS CIRCUITOS IMPRESOS.....	65
3.3. PRUEBAS Y RESULTADOS.....	69
3.3.1. PRUEBAS.....	69
3.3.2. RESULTADOS.....	70
CAPÍTULO IV	
CONCLUSIONES Y RECOMENDACIONES.....	71
4.1. CONCLUSIONES.....	71
4.2. RECOMENDACIONES.....	73
4.3. GLOSARIO.....	74
4.4. BIBLIOGRAFÍA.....	75
4.5. ANEXOS.....	78

ANEXO A - RUTEO DE LAS PISTAS DEL CIRCUITO.....	79
ANEXO B - CONSTRUCCIÓN DE CIRCUITOS Y CABLEADO.....	80
ANEXO C - PRUEBAS Y RESULTADOS.....	82
ANEXO D – DATASHEETS.....	84
ANEXO E – LIBRERÍAS DE MIKRO C.....	90
ANEXO F – PROGRAMACION COMPLETA.....	93

ÍNDICE DE FIGURAS

Figura 1.1. Estructura básica de un microcontrolador.....	1
Figura 1.2. Arquitectura Harvard de microcontrolador (acceso a memorias)	2
Figura 1.3. Distribución de pines del PIC16F877A.....	3
Figura 1.4. Distribución de los puertos del PIC16F877A.....	4
Figura 1.5. Capacidad de corriente de los pines y puertos.....	5
Figura 1.6 Configuración primaria de un sensor.....	7
Figura 1.7 Sensor QRD 1114.....	9
Figura 1.8 Displays de 7 segmentos.....	10
Figura 1.9 Display de ánodo común.....	12
Figura 1.10 Leds de alta intensidad.....	14
Figura 1.11 Circuito Regulador de Voltaje.....	18
Figura 1.12 Diagrama circuital de pines y elementos del regulador.....	19
Figura 1.13 Diagrama circuital del regulador de Voltaje externo	20
Figura 1.14 Diagrama circuital de los sensores.....	23
Figura 1.15 Fase de compilación.....	25
Figura 2.1. Diagrama de Bloques Total.....	29
Figura 2.2. Etapa de entrada	30
Figura 2.3. Etapa de Control.....	30
Figura 2.4. Etapa de Salida.....	31
Figura 2.5. Diagrama de Flujo General.....	32
Figura 2.6. Diagrama de Flujo de Sistema de Control de Parqueos.....	35
Figura 2.7. Diseño circuital.....	40
Figura 2.8. Ejemplo 1.....	44

Figura 2.9. Ejemplo 2.....	44
Figura 2.10. Circuito Final.....	49
Figura 2.11. Sitios Libres en el parqueadero.....	50
Figura 2.12. Estacionamiento 1 y 3 estacionados.....	51
Figura 2.13. Sitios estacionados.....	52
Figura 3.1. Regulador de voltaje.....	53
Figura 3.2. Diagrama circuital de un solo sensor.....	54
Figura 3.3. Conexión total de un solo sensor con leds.....	55
Figura 3.4. Diagrama circuital total.....	56
Figura 3.5. Implementación total de los 4 sensores en protoboard.....	56
Figura 3.6. Diagrama Circuital del Microcontrolador.....	57
Figura 3.7. Diagrama Circuital de los Displays.....	58
Figura 3.8. Implementación de Microcontrolador y Displays.....	59
Figura 3.9. Implementación total de prototipo.....	59
Figura 3.10. Diagrama circuital de la implementación total de prototipo.....	60
Figura 3.11. Sitios de parqueos habilitados.....	61
Figura 3.12. Primer sitio de parqueo estacionado.....	62
Figura 3.13. Sitios de parqueo estacionados, contador cero.....	62
Figura 3.14. Diseño de la estructura de parqueadero.....	63
Figura 3.15. Estructura de prototipo de parqueadero.....	64
Figura 3.16. Placa principal del sistema de control de parqueos en ARES.....	65
Figura 3.17. Placa Principal vista inferior.....	66
Figura 3.18. Placa Principal vista superior.....	67
Figura 3.19. Panel de Displays Vista Anterior y Frontal.....	67

Figura 3.20. Prototipo Implementado en su Totalidad.....68

LISTA DE TABLAS

Tabla 1.1. Distribución de los puertos del PIC16F877A.....	5
Tabla 1.2. Características del regulador.....	18
Tabla 2.1. Tabla de Verdad para 4 Parqueos.....	43

LISTA DE ANEXOS

ANEXOS.....	78
ANEXO A – RUTEO DE LAS PISTAS DEL CIRCUITO.....	79
ANEXO B – CONSTRUCCIÓN DE CIRCUITOS Y CABLEADO.....	80
ANEXO C – PRUEBAS Y RESULTADO.....	82
ANEXO D – DATASHEETS.....	84
ANEXO E – LIBRERÍAS DE MIKRO C.....	90
ANEXO F – PROGRAMACION COMPLETA.....	93

PRESENTACIÓN

La falta de información en la entrada de un parqueadero en un establecimiento, para señalar la cantidad de parqueos disponibles y su cantidad, provoca aglutinamiento de vehículos al momento de ingresar y buscar el lugar disponible.

Los microcontroladores como alternativa de control en automatizaciones, han generado una amplia demanda en lugares donde los usuarios requieren información precisa y eficiente: tal es el caso de los parqueaderos de establecimientos donde suele concurrir gran cantidad de personas y es imprescindible llevar un conteo correcto de plazas para parquear los vehículos.

La necesidad de desarrollar este proyecto radica en la eficiencia de sistemas que optimicen los recursos de un parqueadero, evitando las deficiencias en su funcionamiento. De este modo el sistema a desarrollarse contribuirá con un menor costo ante el mercado competente.

Los sensores infrarrojos detectan la posición de los vehículos en cada sitio de parqueo, dentro de su campo de visión, para asegurar que dicho vehículo se encuentre ocupando completamente el lugar indicado.

Los leds y displays permiten visualizar cada estado (estacionado o habilitado) y presencia en cada sitio de parqueo especificando la disponibilidad del mismo.

RESUMEN

Este presente documento se encuentra dividido en 4 capítulos, los cuales detallan el proceso para el alcance de los objetivos planteados.

El capítulo 1 se basa en los fundamentos teóricos de los microcontroladores, sensores infrarrojos, leds, displays de 7 segmentos, resistencias, capacitores, regulador de voltaje y el programa ensamblador MIKROC con el propósito de comprender su funcionamiento y aplicación en este proyecto.

El capítulo 2, está basado en el diseño del proyecto tomando aspectos como: la selección de dispositivos y programa ensamblador, se detalla los materiales utilizados para la implementación del circuito; diseño del sistema de control, programación y finalmente la simulación del sistema de control de parqueos.

El capítulo 3, tiene como objetivo la implementación del circuito para lo cual se siguieron diversos procedimientos: elaboración del prototipo del sistema de control de parqueos, construcción de los circuitos impresos, pruebas y resultados para la obtención del proyecto.

En el capítulo 4, se presentan las conclusiones obtenidos a lo largo de la elaboración de este proyecto, las recomendaciones que facilitarán el desarrollo de futuros proyectos, el glosario el cual facilita ayuda para la verificación de conocimientos o duda obtenidas, mediante el datasheet de cada elemento, y finalmente se encuentra la parte de anexos en la cual se encuentran las pruebas del circuito terminado en la maqueta, pruebas diversas de la programación en MikroC y los datasheet de cada elemento implementado.

CAPÍTULO I: MARCO TEÓRICO

1.1. MICROCONTROLADOR

1.1.1 CARACTERÍSTICAS ^[1]

Los microcontroladores son circuitos integrados programables que poseen las características funcionales de un computador tales como: Memoria, CPU y Unidades de entrada y salida (E/S). Los microcontroladores son empleados en aplicaciones a nivel industrial así como en diversos lugares electrónicos donde son capaces de ejecutar órdenes almacenadas en su memoria.

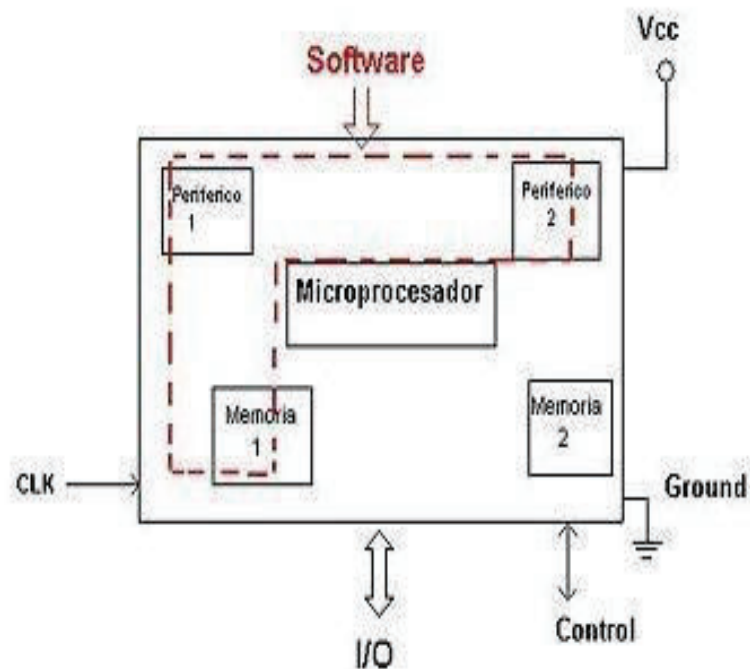


Figura 1.1. Estructura básica de un microcontrolador ^[1]

Como principales características, por lo general, cuenta con: un generador de reloj integrado, Memoria de acceso aleatorio de poca capacidad y/o ROM/EPROM/ EEPROM/flash. Adicional a lo mencionado, un microcontrolador, para su funcionamiento, requiere de programas de control y un cristal de sincronización.

1.1.2 FUNCIONAMIENTO

Los microcontroladores en la actualidad adoptan la arquitectura Harvard que posee dos memorias independientes: memoria solo de datos y memoria solo de instrucciones. Estas memorias pueden realizar operaciones de lectura/ escritura simultáneamente en ambas memorias y poseen sistemas de buses de acceso respectivamente.

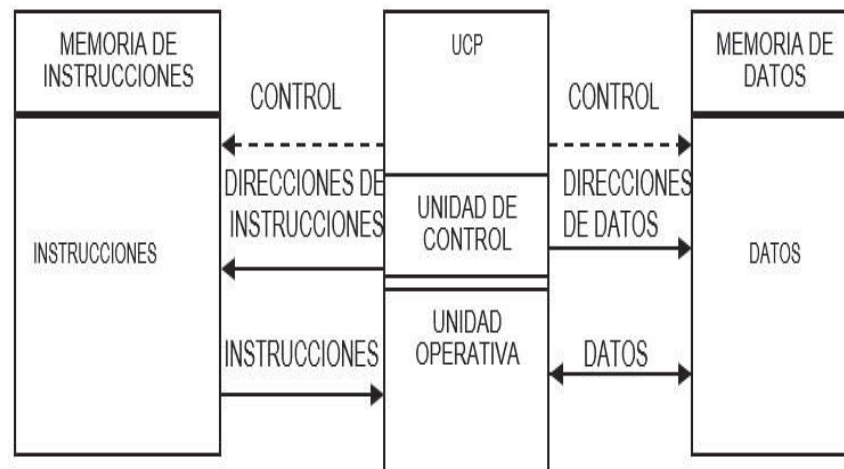


Figura 1.2Arquitectura Harvard de microcontrolador (acceso a memorias)^[1]

1.1.3 MICROCONTROLADOR 16F877A ^[2]

El PIC16F877 es un microcontrolador con memoria de programa tipo FLASH, lo que representa gran facilidad en el desarrollo de prototipos y en su aprendizaje ya que, a diferencia de las versiones EPROM que requieren borrar con luz ultravioleta, permite reprogramarlo nuevamente sin ser borrado con anterioridad.

El PIC16F877 es un microcontrolador de Microchip Technology fabricado en tecnología CMOS, su consumo de potencia es muy bajo y además es completamente estático, esto quiere decir que el reloj puede detenerse y los datos de la memoria no se pierden. El encapsulado más común para este microcontrolador es el DIP (*Dual In-line Pin*) de 40 pines, propio para usarlo en experimentación.

1.1.3.1 Configuración de pines del microcontrolador:

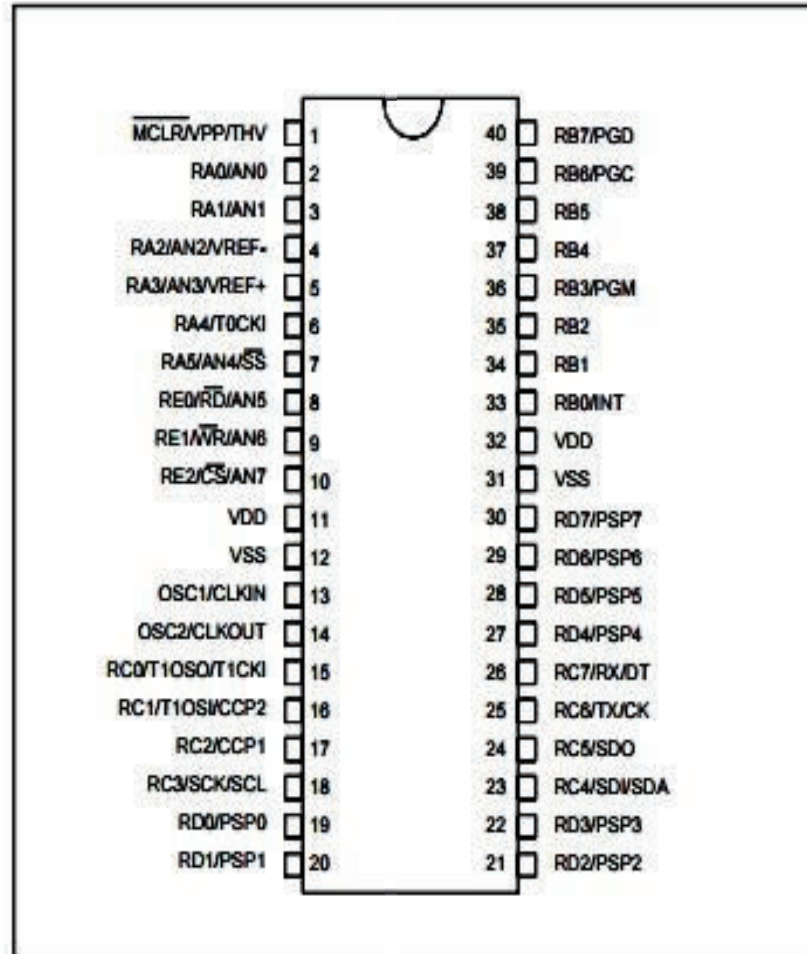


Figura 1.3 Distribución de pines del PIC16F877A ^[16]

Los pines de entrada/salida de este microcontrolador están organizados en cinco puertos, el puerto A con 6 líneas, el puerto B con 8 líneas, el puerto C con 8 líneas, el puerto D con 8 líneas y el puerto E con 3 líneas.

Cada pin de esos puertos se puede configurar como entrada o como salida independiente programando un par de registros diseñados para tal fin. En ese registro un bit en "0" configura el pin del puerto correspondiente como salida y un bit en "1" lo configura como entrada.

1.1.3.2 Distribución de puertos del microcontrolador:

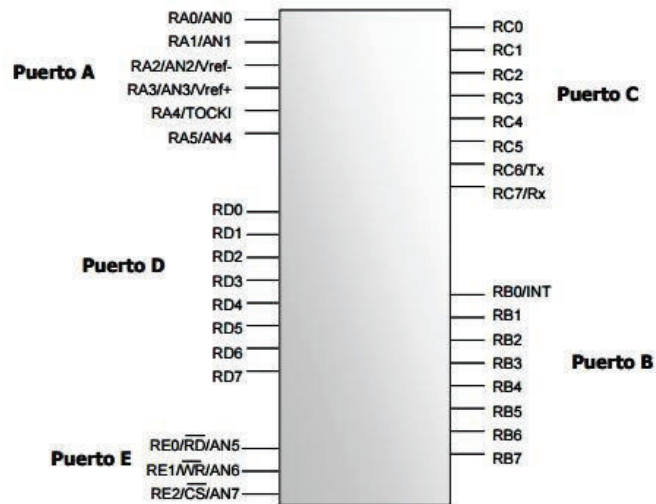


Figura 1.4 Distribución de los puertos del PIC16F877A ^[15]

Los pines del puerto A y del puerto E pueden trabajar como entradas para el convertidor Análogo a Digital interno, donde se conecta una señal proveniente de un sensor o de un circuito analógico para que el microcontrolador la convierta en su equivalente digital y pueda realizar algún proceso de control o de instrumentación digital.

El pin RB0/INT se puede configurar por software para que funcione como interrupción externa, para configurarlo se utilizan unos bits de los registros que controlan las interrupciones. El pin RA4/TOCKI del puerto A puede ser configurado como un pin de entrada/salida o como entrada del temporizador/contador.

Cuando este pin se programa como entrada digital, funciona como un disparador de Schmitt (*Schmitt trigger*), puede reconocer señales un poco distorsionadas y llevarlas a niveles lógicos (*cero y cinco voltios*). Cuando se usa como salida digital se comporta como colector abierto por lo tanto, se debe poner una resistencia de *pull-up* (resistencia externa conectada a un nivel de cinco voltios). Como salida, la lógica es inversa: un "0" escrito al pin del puerto entrega en el pin un "1" lógico.

Además, como salida no puede manejar cargas como fuente, sólo en el modo sumidero. El puerto E puede controlar la conexión en modo microprocesador con otros dispositivos utilizando las líneas RD (lectura), WR (escritura) y CS (selección de chip). En este modo el puerto D funciona como un bus de datos de 8 bits (pines PSP). La máxima capacidad de corriente de cada uno de los pines de los puertos en modo sumidero (sink) o en modo fuente es de 25 mA. La máxima capacidad de corriente total de los puertos es:

Tabla 1.1 Distribución de los puertos del PIC16F877A ^[14]

	PUERTO A	PUERTO B	PUERTO C	PUERTO D
Modo sumidero	150 mA	200 mA	200 mA	200mA
Modo fuente	150 mA	200 mA	200 mA	200mA

El consumo de corriente del microcontrolador para su funcionamiento depende del voltaje de operación, la frecuencia y de las cargas que tengan sus pines. Para un oscilador de 4 MHz el consumo es de aproximadamente 2 mA; aunque este se puede reducir a 40 microamperios cuando se está en el modo reposo o modo de bajo consumo (en este modo el micro se detiene y disminuye el consumo de potencia).

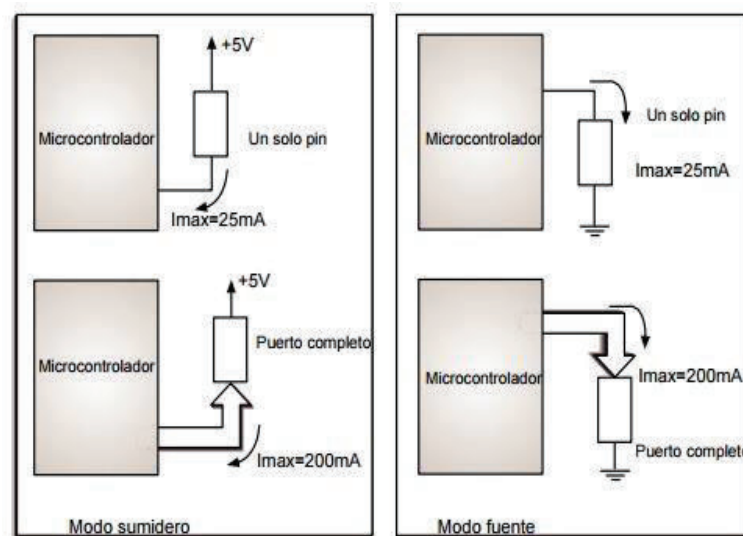


Figura 1.5 Capacidad de corriente de los pines y puertos. ^[14]

1.1.3.3 Características principales del PIC 16F877A ^[3]:

Las principales características del PIC utilizado en el circuito son:

- Memoria de programa:
FLASH, 8 K de instrucciones de 14 bits c/u.
- Memoria de datos: 368 bytes RAM, 256 bytes EEPROM.
- Pila: 8 niveles (14 bits).
- Fuentes de interrupción: 13
- Instrucciones: 35
- Encapsulado: DIP de 40 pines.
- Frecuencia oscilador : 20 MHz (máxima)
- Temporizadores/Contadores: 1 de 8 bits (Timer 0); 1 de 16 bits (Timer 1); 1 de 8 bits (Timer 2) con pre y post escalador. Un perro guardián (WDT)
- Líneas de E/S: 6 del puerto A, 8 del puerto B, 8 del puerto C, 8 del puerto D y 3 del puerto E, además de 8 entradas análogas.
- Dos módulos de Captura, Comparación y PWM: - Captura: 16 bits.
Resolución máx. = 12.5 nseg. - Comparación: 16 bits. □ Resolución máx. = 200 nseg. – PWM □ Resolución máx. = 10 bits.
- Convertidor Análogo/Digital de 10 bits multicanal (8 canales de entrada).
- Puerto serial síncrono (SSP) con bus SPI (modo maestro) y bus I²C (maestro/esclavo).
- USART (Universal Synchronous Asynchronous Receiver Transmitter) con dirección de detección de 9 bits.
- Corriente máxima absorbida/suministrada por línea (pin): 25 mA
- Oscilador:
Soporta 4 configuraciones diferentes: XT, RC, HS, LP.
- Tecnología de Fabricación: CMOS.
- Voltaje de alimentación: 3.0 a 5.5 V DC.
- Puede operar en modo microprocesador.

1.1.3.4 Observaciones:

El microcontrolador utilizado en el proyecto pertenece a la familia 16F87XX, debido a la versatilidad, fácil manejo para la programación, capacidad de memoria y porque se tiene conocimiento de sus conexiones y uso.

1.2 SENSORES INFRARROJOS:

1.2.1 CARACTERÍSTICAS ^[4]:

El sensor infrarrojo es un dispositivo opto-electrónico diseñado para detectar, clasificar determinar la posición de un objeto y detectar formas colores, superficies en condiciones normales y extremas. Este dispositivo es capaz de detectar la radiación electromagnética infrarroja de los objetos, que todo cuerpo posee pero es invisible a la visión humana, siempre y cuando se encuentre dentro del campo de visión. La radiación de los cuerpos está debajo del rango de luz visible pero el sensor puede detectar normalmente; los sensores se clasifican en: Pasivos y Activos.

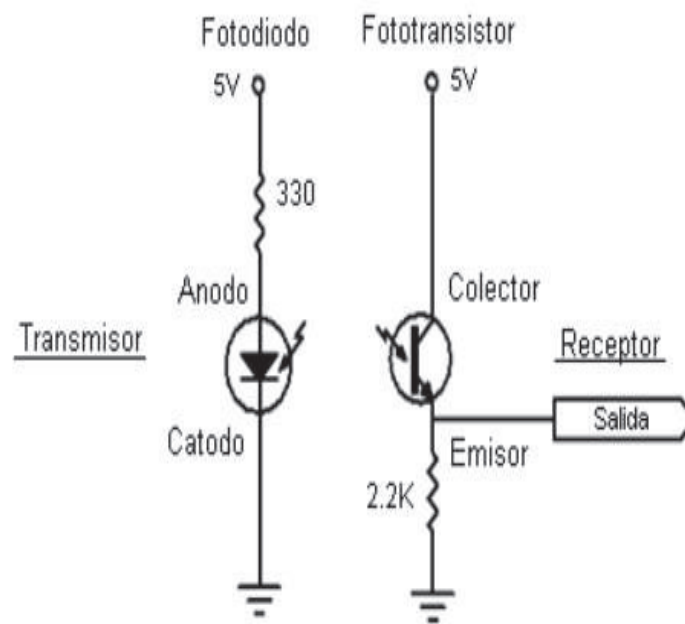


Figura 1.6 Configuración primaria de un sensor ^[4]

1.2.2 LED INFRARROJO (IRLED) ^[19]

El diodo IRLED es un emisor de rayos infrarrojos que son una radiación electromagnética situada en el espectro electromagnético; mientras este espectro es bastante ancho, la mayoría de los LED emitirán luz con una longitud de onda de cerca de 1000nm con un ancho de banda de unos 50nm, esto significa que un LED con valor de 1000nm producirá luz de entre 950 y 1050nm.

Estos diodos se diferencian de los LED por el color de la cápsula que los envuelve que es de color azul o gris, el diámetro de ésta es generalmente de 5 mm.

Estos sensores poseen multitud de aplicaciones como por ejemplo, vigilancia de objetos y personas, medida de temperaturas remotas en aplicaciones industriales, detección de fugas de calor, monitorización y detección de incendios, y diferentes aplicaciones científicas y médicas. La banda del infrarrojo va desde los 0.75 hasta los 1000 nm (nanómetros), y podemos diferenciar las siguientes bandas:

- IR cercano: 0.75 a 3nm.
- IR medio: 3 a 6nm.
- IR lejano: 6 a 15nm.
- IR extremo: 15 a 1000nm.

1.2.2.1 FUNCIONAMIENTO ^[19]

Los rayos infrarrojos (IR) entran dentro del fototransistor donde encontramos un material piroeléctrico, natural o artificial, normalmente formando una lámina delgada dentro del nitrato de galio [Ga (NO₃)₃], nitrato de Cesio (CsNO₃), derivados de la fenilpirazina, y ftalocianina de cobalto. Normalmente están integrados en diversas configuraciones (1, 2, 4 píxeles de material piroeléctrico).

1.2.3 SENSOR QRD1114 (Sensor Objeto Reflectante) ^[18]

El sensor Opto reflexivo QRD1114 es básicamente la incorporación de un fotodiodo IRLED y un fototransistor en un encapsulado cada uno alimentado independientemente a 5V, el IRLED envía una señal infrarroja hacia arriba cuando golpea con un objeto a no más de 1 cm la señal regresa y la base que es un receptor, "la base del transistor" es la que genera el funcionamiento.

1.2.3.1 Características:

Las características del sensor utilizado para el desarrollo del circuito son las siguientes:

- Salida Fototransistor
- Sin detección superficie de contacto
- Fuera de foco para las superficies de detección difusa
- Paquete compacto
- Filtro de la luz del día en el sensor.



Figura 1.7 Sensor QRD 1114 ^[18]

1.2.3.2 Especificaciones:

El sensor QRD1114 está constituido por 4 entradas o pines que sirven como un transistor, el cuál envía una señal para habilitar el funcionamiento. Las especificaciones eléctricas de este sensor se encuentran en el ANEXO D.

1.2.3.3. Observaciones:

Para esta tesis se va utilizar 4 sensores: QRD1114, los cuales van a cada uno de los parqueaderos para detección de cada automóvil al instante de que este ocupe un lugar en el estacionamiento o cuando dejen el mismo.

1.3 DISPLAYS DE 7 SEGMENTOS Y LEDS:

1.3.1 Displays de 7 segmentos: ^[7]

1.3.1.1 Definición:

Los displays de 7 segmentos son componentes desarrollados para la representación de números que se utiliza en los diversos dispositivos electrónicos. Se encuentra compuesto de 7 segmentos (conjunto de leds con conexiones internas apropiadas) y un punto decimal. El display se encuentra formado por pines para ser conectado a un circuito; estos displays pueden ser de ánodo común o de cátodo común.

Los displays utilizados en el diseño del circuito son de ánodo de común.

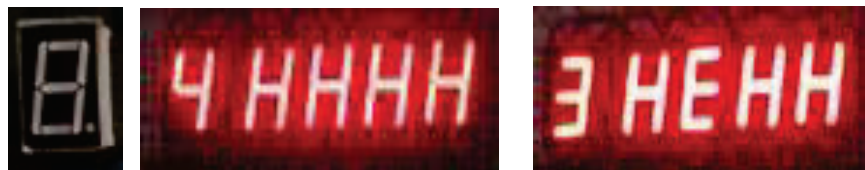


Figura 1.8 Displays de 7 segmentos

Con cada uno de los pines se puede manejar respectivamente cada uno de los segmentos para poder encenderlos o apagarlos. Los pines de la mitad representados por U corresponden a la polarización que se le dará al encapsulado del display, según sea de ánodo o cátodo común, a tierra o Vcc.

Se puede constatar las siguientes afirmaciones para el display de 7 segmentos:

- Los segmentos del display son leds y se nombran con letras de “a” a la “g”.
- Los leds se conectan internamente para reducir el número de pines del integrado.
- En los display tipo cátodo común, el terminal U se conecta al terminal negativo.
- En los display tipo ánodo común, el terminal U se conecta al terminal positivo.

Características Específicas y Generales del Display de 7 Segmentos ^[8]

- Solidez: excelente
- Angulo de visibilidad: 150 grados
- Consumo por dígito: 50 mW
- Vida media en horas: 100000
- Luminosidad: buena
- Facilidad de montaje: excelente □ Vcc general: 1.5 volt.
- La Vcc depende del color del LED.
- Para un color rojo: Vcc=1.7volt.

1.3.1.2 Funcionamiento:

Los leds del display trabajan con baja tensión y con pequeña potencia, y pueden excitarse directamente con compuertas lógicas. Por lo general se usa un codificador (en nuestro caso decimal/BCD) que activando un solo terminal de la entrada del codificador, activa las salidas correspondientes mostrando el número deseado.

Los siete segmentos que forman un dígito con punto decimal a su lado, posee internamente leds con sus extremos llamados ánodo y cátodo.

Al ser encapsulados la totalidad de los extremos (ánodos y cátodos) suman un total de 10 extremos, siendo esto posible debido a la interconexión interna de todos los leds.

1.3.1.3 *Display ánodo común:*

Todos los ánodos de los diodos LED están unidos y a su vez conectados a una fuente de alimentación, lo que facilita la conexión con los otros dispositivos y con el microcontrolador.

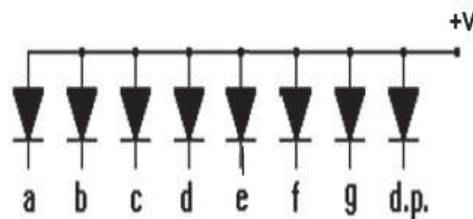


Figura 1.9 Display de ánodo común ^[7]

1.3.1.4 *Observaciones:*

Se utilizaron 5 displays de 7 segmentos de ánodo común para la implementación, debido a que los 4 displays son utilizados para cada parqueadero mientras que el quinto display permite verificar la operación del contador para controlar la entrada y la salida de los automóviles. Los 4 displays pertenecientes a cada parqueadero permiten visualizar si está en estado ocupado E o habilitado H, mientras que el último display muestra los números de parqueaderos habilitados u ocupados según vayan ingresando o saliendo los usuarios.

1.3.2. LEDS (diodo emisor de luz): ^[6]

1.3.2.1 *Definición:*

El led es un diodo que emite luz en respuesta a una corriente eléctrica que fluye a través del mismo. Los diodos leds pueden ser de diversos colores ya que

dependen del material de construcción, también existen diversos leds dependiendo de la aplicación. Los Leds tienen dos patillas de conexión una larga y otra corta. Para que pase la corriente y emita luz se debe conectar la patilla larga al polo positivo y la corta al negativo; en caso contrario la corriente no pasará y no emitirá luz. Su símbolo se encuentra representado tal que el ánodo es la patilla larga.

Los leds funcionan adecuadamente a 2 voltios, pero si se requiere trabajar con tensiones mucho más altas, es preciso conectar una resistencia en serie para que la resistencia recoja parte de la tensión y que solo dos voltios lleguen al led.

Ventajas de los led.

Los leds consumen menos energía que las lámparas convencionales, es decir, que el 80% de la energía se transforma en luz mientras que en las lámparas solo es el 20% y lo demás se convierte en calor. Su tiempo de vida útil es mucho mayor, superior a las 100.000 horas de luz (11 años de continua emisión lumínica), en tanto que las lámparas alcanzan las 5.000 horas. Los leds además presentan ante las lámparas un tamaño reducido, resistencia a las vibraciones, baja emisión de calor, no poseen mercurio, no crean campos magnéticos, disminuye ruidos en las líneas eléctricas no le afecta el encendido intermitente, fiabilidad y duración.

Los diodos leds que se utilizaron son los leds de alta intensidad debido a que ayudarán a identificar de mejor manera los cambios que se producen en el comportamiento y funcionamiento de los sensores.

1.3.2.2 Leds de alta intensidad:

Los leds de alta intensidad, son leds de mayor potencia que un led normal, los cuales representan un mayor aumento en la eficiencia energética, parámetros que miden esencialmente la cantidad de electricidad que el led convierte en luz.

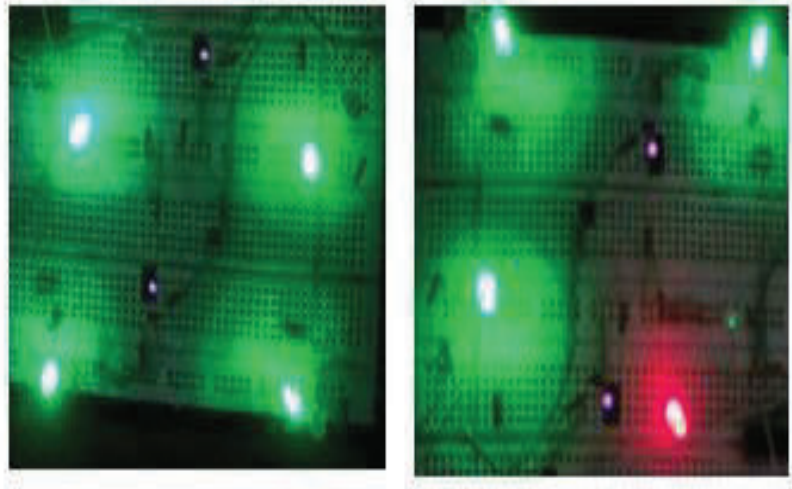


Figura 1.10 Leds de alta intensidad

1.3.2.3. Observaciones:

Se utilizaron 8 leds de alta intensidad 4 leds de color verde y 4 leds de color rojo. Los leds de color verde indican que existe disponibilidad de parqueaderos, mientras que los leds de color rojo reflejan que los parqueaderos se encuentran ocupados.

1.4 RESISTENCIAS, CAPACITORES Y REGULADOR DE VOLTAJE

1.4.1. RESISTENCIAS:

La resistencia es la propiedad física de un elemento o de un dispositivo que impide el flujo de corriente: se representa con el símbolo R. La máxima cantidad de corriente que puede pasar por una resistencia, depende del tamaño de su cuerpo.

Los valores de potencia comunes de las resistencias varían: $\frac{1}{4}$, $\frac{1}{2}$, 1 watt pueden haber de valores mayores.

1.4.1.1 Características Específicas ^[10]

Las resistencias poseen las siguientes características:

- La tolerancia (margen de valores que rodean el valor nominal y el valor real de la resistencia) es determinada por un porcentaje que va desde 0.001% hasta 20% el más utilizada es el de 10%. La tolerancia viene determinada por un código de colores.
- Su rango de funcionamiento viene dado por su capacidad de disipar calor, la tensión y la temperatura máxima que soporta sin deteriorarse.
- El ruido que se ocasiona por los cambios repentinos de corrientes.

1.4.1.2. Funcionamiento ^[9]

- El funcionamiento de las resistencias está basado en la ausencia de conductor perfecto y de esa manera impide el paso de corriente eléctrica como una resistencia eléctrica, pero dependerá siempre del material del que se encuentra formado.
- Su principio de funcionamiento se encuentra bajo la ley de Ohm, la cual establece que en todo conductor, por el cual fluya una corriente eléctrica, se producirá una caída de tensión en sus extremos dependiendo de la resistencia.
- En un circuito electrónico, la resistencia tiene las siguientes funciones: limitación de corriente, carga, polarización, atenuación, divisor de tensión, entre otros.
- Su comportamiento en corrientes alternas desarrolla componentes parásitos que aparecen por los procesos constructivos.
- La potencia a la que trabaja una resistencia está dada por la ecuación:

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

$$P = I^2 * R$$

Código de Colores

Las resistencias se han clasificado de acuerdo a una representación de un código de colores impreso como anillos de colores, que pueden ser 4, en su cuerpo. Sus 2 primeros colores representan los dígitos del valor de la resistencia y el tercero es el factor multiplicador. El último es la tolerancia de la resistencia. En la elaboración del circuito se utiliza diversas resistencias con valores diferentes ya que el elemento a conectarse a cada una de ellas no era el mismo.

1.4.1.3. Observaciones:

Se utilizaron 4 resistencias de 330 ohmios para los sensores, 7 resistencias de 330 ohmios para los displays, 8 resistencias de 5.7K ohmios para los leds, 5 resistencias de 5.7 K ohmios de los transistores para la conexión con los displays.

1.4.2 CAPACITORES:

1.4.2.1 Definición:^[11]

Un capacitor es un elemento de dos terminales formado por dos placas conductoras separadas por un material no conductor; se asemeja mucho a una batería, pues al igual que ésta su función principal es almacenar energía eléctrica, pero de forma diferente. Los capacitores son dispositivos lineales y pasivos, que se utiliza en electricidad y electrónica, capaces de almacenar y liberar energía o carga eléctrica, basándose en campos eléctricos.

La función de un capacitor es almacenar cargas eléctricas de forma instantánea y liberarla de la misma forma en el preciso momento que se requiera. En el circuito se utilizaron diversos capacitores unos que son los electrolíticos de diversas capacidades, voltajes de trabajo y los cerámicos.

1.4.2.2. Observaciones:

Se ocuparon 4 capacitores cerámicos (valor: 0.1uf) en paralelo a las resistencias de 4.7k ohmios para los filtros existentes entre cada sensor y el transistor.

Dos capacitores cerámicos (valor: 22 pf) en paralelo al cristal para el PIC 16F877A. Se utilizaron capacitores entre electrolíticos (valores: 220uf) y cerámicos (valor: 0.1 uf) para el regulador externo.

1.4.3. REGULADOR DE VOLTAJE: ^[13]

1.4.3.1. Definición:

La familia 78xx y 79xx son una gama de integrados dedicados a la regulación de voltaje, hay muchas variables: regulables, fijos, de tensión positiva o negativa. Estos reguladores están conformados por 3 terminales: Vi voltaje de entrada, Vo voltaje de salida y el terminal central o común, con especificaciones similares y que sólo difieren en la tensión de salida suministrada y en la corriente que es capaz de dar ante una demanda de ello.

1.4.3.2. Funcionamiento ^[13]

Todo regulador de voltaje tiene como función convertir un voltaje continuo o alterno de entrada en una salida de voltaje continuo y estable, siempre y cuando se mantenga esa tensión dentro de amplias condiciones de carga.

1.4.3.3 Regulador de voltaje L7805

El 7805 es un regulador de voltaje lineal de tipo de encapsulado TO220, aguanta una corriente máxima de 1 Amperio, un voltaje máximo de 35 Voltios y puede trabajar hasta 120 °C.

1.4.3.4. Descripción de pines y funcionamiento:

El regulador fue implementado en el PROTOBOARD para verificación de funcionamiento correcto del circuito y validaciones de pruebas empleados en el mismo. Se utiliza el circuito integrado de regulador de voltaje que presenta las siguientes características:

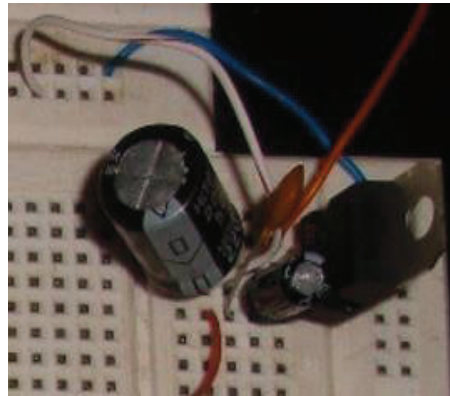


Figura 1.11 Circuito Regulador de Voltaje

Características:

Tabla 1.2 Características del regulador ^[21]

Voltaje de salida	V_0	5V	
Tolerancia	-	$\pm 4\%$	
Voltaje máximo de entrada	V_i	35V	
Corriente de salida	I_0	1A	
Temperatura de operación	T_{opr}	0-70°C	

El L7805 dispone de 3 pines de conexión:

- El pin 1 es para la alimentación.
- El pin 2 es tierra que se comparte para la entrada y la salida, también está unido, a la parte metálica del 7805, un disipador de calor TO-220 para disminuir la temperatura.
- El pin 3 es el pin de salida.

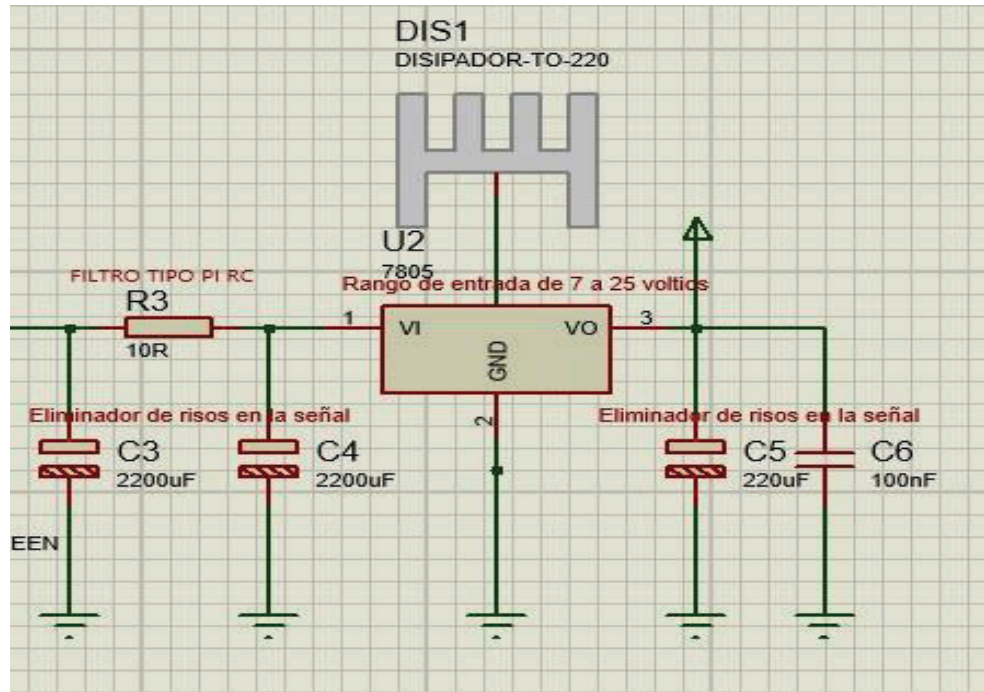


Figura 1.12 Diagrama circuital de pines y elementos del regulador

El rango que se le va a suministrar al regulador será entre 7 a 25 voltios.

1.4.3.5. *Regulador de voltaje externo*

Este regulador de voltaje externo al circuito es utilizado como circuito de protección en el caso que existiera un sobre voltaje.

Este regulador funciona de igual manera que un filtro de protección, el cual permite reducir el ruido externo para obtención de una mejor funcionalidad del circuito.

1.4.3.5.1. *Componentes:*

Los elementos utilizados en este regulador comprenden tanto para el circuito de protección y el filtro de protección: resistencias (valores: 2.2 k ohmios, 1 ohmio), 1 fusible F2AL 250V, 1 diodo LED verde, 1 diodo IN4007, capacitores electrolitos (valor: 2200uf).

El diodo 1N4007 es un diodo de protección inversa para el circuito. Seguido, se encuentra un filtro tipo RC conformado por una resistencia de 10 ohm junto a dos capacitores electrolíticos de 2200 uf que cumplen la función de eliminador de risos en la señal.

Por ende el voltaje de salida que se obtenga al final del circuito de protección ingresará al pin 1 del regulador. La salida de regulador, pin 3, tendrá un capacitor electrolítico de 220 uf y un capacitor cerámico de 100 nf en paralelo para cumplir la función de eliminar risos de la señal obtenida.

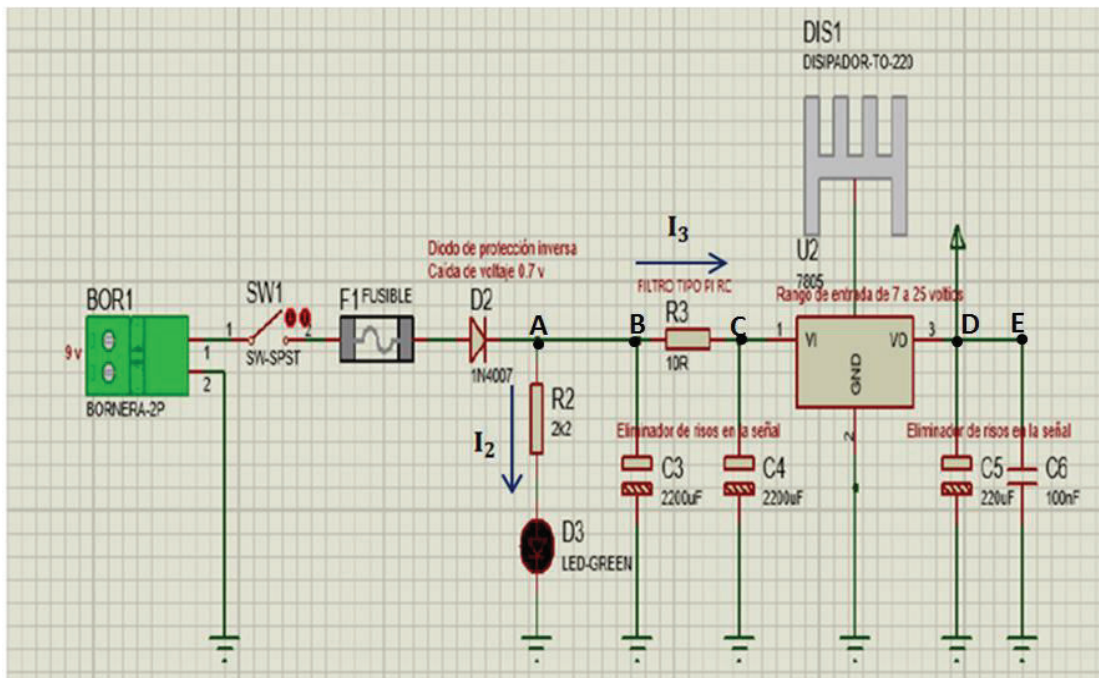


Figura 1.13 Diagrama circuital del regulador de Voltaje externo

1.4.3.5.2. Cálculos:

Datos:

$$V_{\text{entrada}} = 9 \text{ V}$$

$$I_2 = ?$$

$$I_3 = ?$$

$$R_2 = 2,2 \text{ k } \Omega$$

$$R_3 = 10 \text{ } \Omega$$

- *PARA CALCULAR LA CORRIENTE QUE CIRCULA:*

$$V_{diodo} = V_{entrada} - V_{caida\ en\ el\ diodo}$$

$$V_{diodo} = 9v - 0.7v$$

$$V_{diodo} = 8.3V$$

- Aplicando LEY DE OHM:

$$V = R * I$$

$$I = \frac{V}{R}$$

$$I_3 = \frac{V_{diodo}}{R_3}$$

$$I_3 = \frac{8.3\ V}{10\ \Omega}$$

$$I_3 = \mathbf{0.00083mA}$$

- *Para calcular la corriente que circula: I₂*

$$V_{caida\ diodo\ led} = 2.1\ V$$

- Aplicando LEY DE OHM:

$$V = R * I$$

$$I = \frac{V}{R}$$

$$I = \frac{V_{caida\ diodo\ led}}{R_2}$$

$$I_2 = \frac{2.1\ V}{2.2k\ \Omega}$$

$$I_2 = \mathbf{0.9545mA}$$

- *PARA CALCULAR LOS VOLTAJES EN CADA PUNTO DEL CIRCUITO:*

Voltaje en el punto D:

$$V_D = V_{CC}$$

$$V_D = 5 \text{ V}$$

Voltaje en el punto E:

$$V_E = V_D$$

$$V_E = 5 \text{ V}$$

Voltaje en el punto A:

$$V_A = V_{\text{Entrada}} - V_{\text{Caída del diodo}}$$

$$V_A = (9 - 0.6) \text{ V}$$

$$V_A = 8.4 \text{ V}$$

Voltaje en el punto B:

$$V_B = V_A$$

$$V_B = 8.4 \text{ V}$$

○ CIRCUITO DE LOS SENSORES(QRD114)

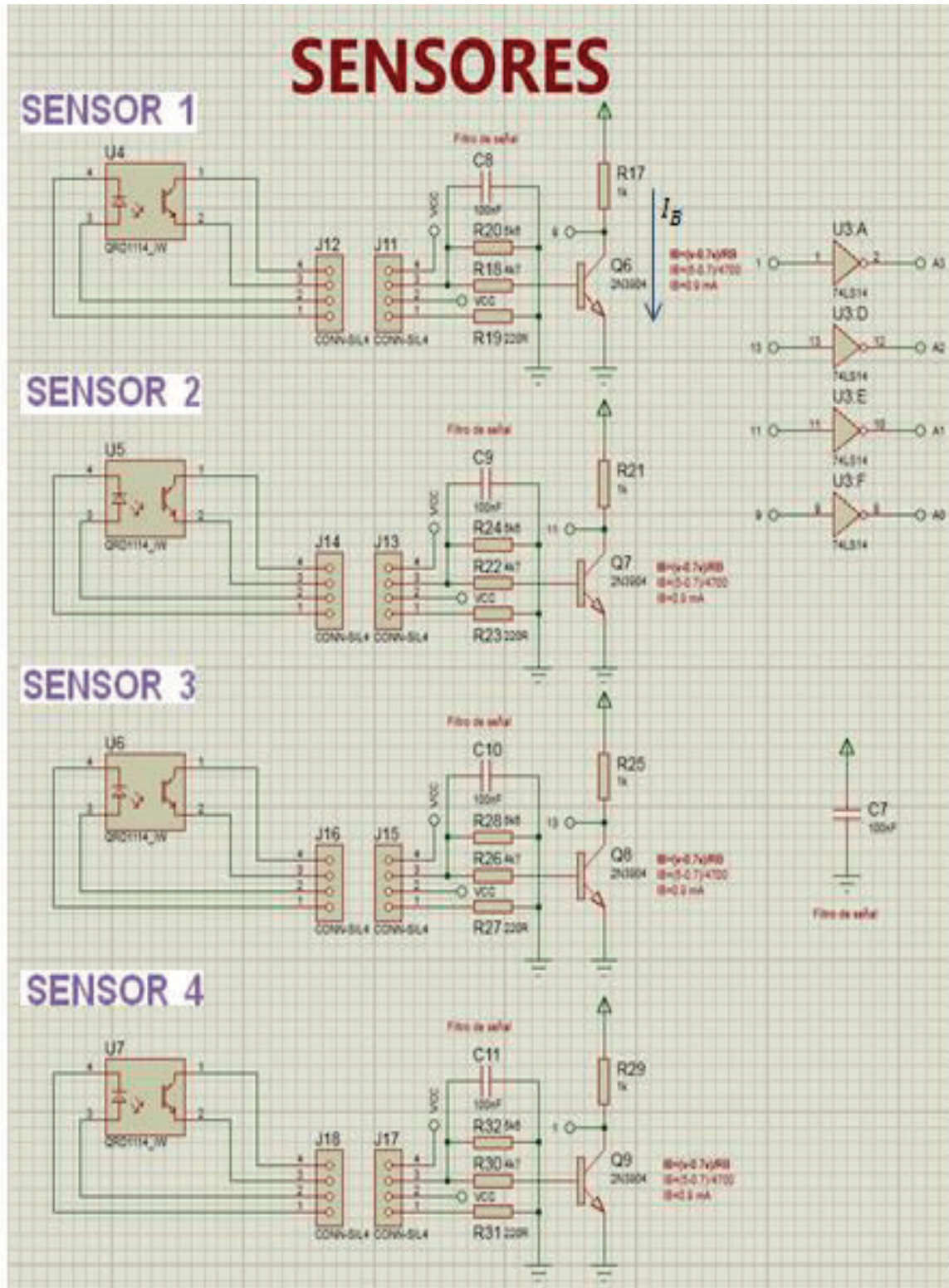


Figura 1.14 Diagrama circuital de los sensores

- *Cálculos:*

Datos:

$$V_{entrada} = 5 \text{ V}$$

$$I_B = ?$$

Por aplicación de transistores:

$$V_E = 0 \text{ V}$$

$$V_B = 0,7 \text{ V}$$

$$V_C = V_{CC} - I_C R_C$$

$$I_B = \frac{V_{CC} - 0,7}{R_B}$$

$$I_B = \frac{V_{CC} - 0,7}{R_B} = \frac{5 - 0,7}{4,7K\Omega} = 0,9 \text{ mA}$$

1.5. PROGRAMA ENSAMBLADOR MIKROC

MikroC PRO for PIC es un compilador utilizado para programar microcontroladores PIC en lenguaje C; es un lenguaje compilado debido a que los archivos fuentes que contienen el código C se traducirán a lenguaje máquina a través del compilador.

1.5.1. CARACTERÍSTICAS PRINCIPALES DE MIKROC ^[20]:

El lenguaje utilizado por MikroC es muy similar al lenguaje C estándar pero difiere en aspectos tales como: en mejoras, facilidad de programación de los microcontroladores PIC, limitaciones de la arquitectura de Hardware de los PIC microcontroladores.

1.5.1.1 Fase de Compilación

Al escribir el programa fuente (código de MikroC) e iniciar el proceso de compilación, el compilador busca las directivas del preprocesador, que comienzan con #, dentro del código, para modificar el código fuente de acuerdo a las directivas. Se eliminan las informaciones inútiles del código, como comentarios y espacios en blanco.

Se procede a traducir el código a un archivo binario llamado .mcl. Se recupera la información necesaria para que opere el programa de los archivos externos con linker y agrupa en un solo archivo .dbg.

Como punto final el generador hex originará un archivo .hex que servirá para poder cargar y ejecutar en el microcontrolador. Todos estos pasos para realizar la compilación se llama building.

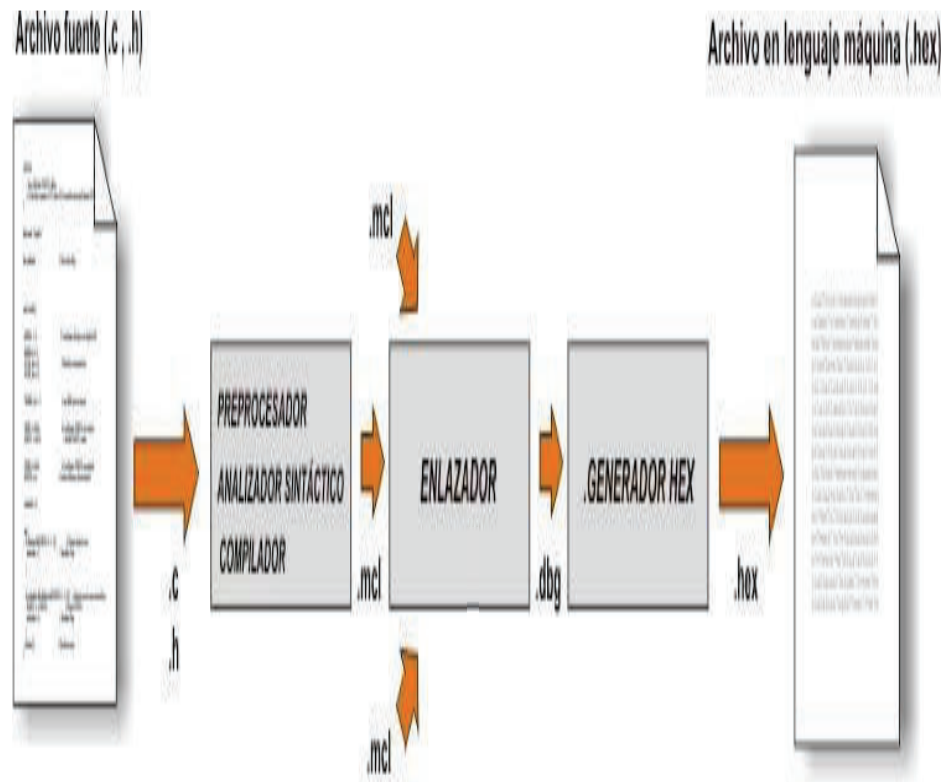


Figura 1.15 Fase de compilación ^[15]

1.5.1.2. Estructura del Programa ^[15]

El programa se divide en 4 partes:

1. Activar y configurar el convertidor A/D incorporado;
2. Medir el valor analógico;

Para poder realizar esto, es necesario dividir en diversos trozos de operaciones pero que sea capaz de escribirse como una sola función.

Los programas en MikroC poseen una función llamada main(), primera función a ejecutar, que encerrará en llaves {} todas las “operaciones” o sentencias a ejecutarse (resto de funciones).

La función main () es obligatoria al inicio del programa en MikroC, en cambio las sentencias serán opcionales según sea el requerimiento de la programación. El lenguaje C distingue mayúsculas de minúsculas por lo que un adecuado uso de la manera de escribir el programa es necesario. Todas las sentencias deberán terminar con un punto y como para que el compilador pueda traducirlas por separado a lenguaje máquina.

1.5.1.3. Comentarios ^[15]

Los comentarios aclaran las instrucciones del programa o abastecen de información del mismo y constituyen una parte importante en el programa.

En situaciones en que se requiera mejorar modificar, actualizar o simplificar el programa, ayudan a interpretar la programación adecuadamente.

Si los comentarios son muy largos, se asignará una secuencia especial

/*... (Texto)...*/, pero si son cortos antes de escribir los comentarios se asignará //(texto). La utilización del software para el circuito es uno de los procesos más indispensables ya que permite conseguir que todos los dispositivos que son

conectados al microcontrolador desempeñen su función esperada de manera correcta.

En la actualidad existen diversos lenguajes de programación los cuales se han ido desarrollando o modificando en función a las necesidades o tendencias tecnológicas; por tal razón se ha optado por la utilización del compilador MikroC PRO for PIC.

La utilización del mismo conjuntamente con el programa PROTEUS permitirá la verificación y corrección de errores del sistema.

1.5.1.4. Proceso para la Programación ^[17]:

Programar un microcontrolador es planificar la manera que se va a modificar el comportamiento del PIC en cuanto a la funcionalidad de los recursos y periféricos según las instrucciones asignadas por el programador.

El proceso de transferencia de código de programación al PIC se lo realiza de la siguiente manera:

- **Editar**

Se refiere a escribir el programa, armar las instrucciones en un lenguaje específico para poder indicarle al PIC lo que tiene que hacer.

- **Compilar**

Para la compilación es necesario tomar en cuenta que el PIC como otros circuitos integrados interpreta niveles de voltaje (código binario). Las instrucciones que se graban en la memoria deben ser ingresadas en este código (binario); por la dificultad de y alto conocimiento del funcionamiento interno del PIC resulta poco

práctico escribir en este lenguaje. Por aquello existen lenguajes de alto nivel que facilitan la programación, ahorrar tiempo y brindar mayor facilidad de comprensión para entender dicho programa.

Compilar es convertir un programa escrito en lenguaje de alto nivel en lenguaje o código que el PIC pueda entender.

- **Grabar**

Es el proceso en el cual se graba el programa en el PIC, mediante una tarjeta electrónica y software; se pasa el programa compilado de la PC al PIC.

Dependiendo del grabador se esperan unos segundos y se siguen los pasos dando click y está finalizado el proceso de grabación.

- **Comprobar**

Al finalizar todo el proceso de programación del PIC, se realiza la verificación del funcionamiento del programa.

1.5.1.5. Descripción del desarrollo del programa:

Una vez identificadas las condiciones y la lógica que debe tener el programa, se procede a describir de manera general las instrucciones y funciones que se necesita, para el desarrollo del programa se utilizan las herramientas y librerías que ofrece MikroC PRO para el manejo de periféricos.

Para la visualización de las librerías que se encuentran en MikroC, se pueden verificar en los anexos E.

CAPÍTULO II: DISEÑO

2.1 DISEÑO DEL SISTEMA DE CONTROL DE PARQUEOS

Este sistema de control de parqueos proporciona información de la capacidad total del parqueadero y su número actual de disponibilidad de plazas para la ubicación vehicular de los usuarios.

Todos los elementos utilizados son en base a la programación y la elaboración del diseño circuital para proporcionar la correcta información a los usuarios. El diseño circuital fue realizado en el programa CIRCUIT MAKER el cual brinda un fácil manejo de los elementos para la verificación del funcionamiento y correcciones de ser necesarias al momento de querer implementarlo.

2.1.1 DIAGRAMA DE BLOQUES DEL SISTEMA:

El diagrama de bloques consta de 3 etapas:

- Entrada
- Control
- Salida

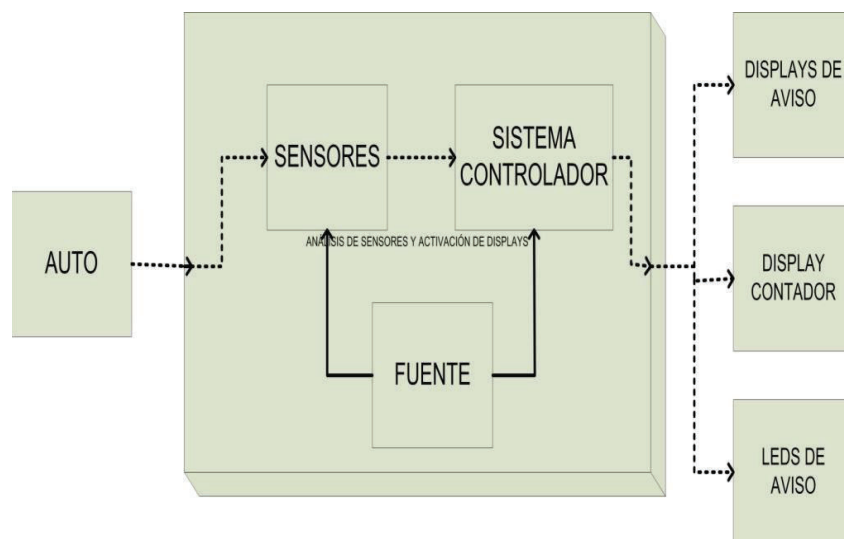


Figura 2.1 Diagrama de Bloques Total

1. ETAPA DE ENTRADA:

Está formada por los dispositivos que intervienen para la ejecución de la etapa de control, en esta etapa constituye únicamente los vehículos.



Figura 2.2 Etapa de entrada

2. ETAPA DE CONTROL:

Esta etapa está constituida por la fuente, los sensores y el sistema controlador (programación en el PIC). En esta etapa se realiza el análisis del funcionamiento de los sensores y la activación de los displays y leds, a través de la interacción de los sensores con el microcontrolador.

Se desarrolla el envío de información de señales de los sensores, que se visualizarán en la etapa final de los displays, contador y leds.

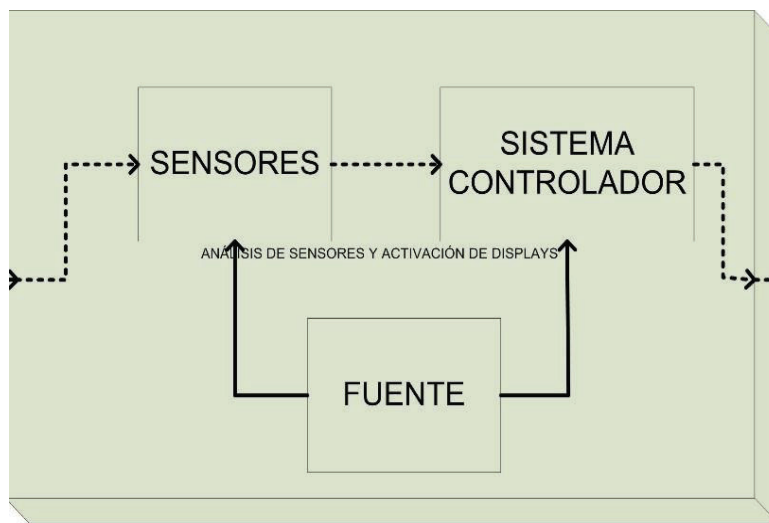


Figura 2.3 Etapa de control

3. ETAPA DE SALIDA:

La interacción de información tiene como resultado la visualización de la información acerca de los sitios de parqueos:

- En los displays se mostrará si en el sitio está un vehículo estacionado o si el sitio se encuentra habilitado.
- En el display del contador se apreciará el número de estacionamientos disponibles en tiempo real.
- El par de leds ubicados en cada sitio de parqueo, informarán si el mismo se encuentra disponible encendiendo un led verde, o si se encuentra ocupado mediante el encendido de un led rojo.

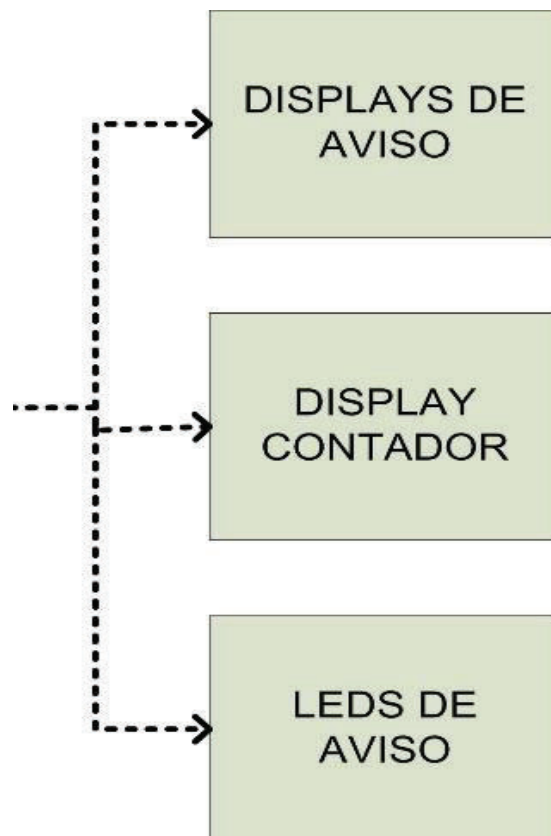


Figura 2.4 Etapa de Salida

2.1.2 DIAGRAMA DE FLUJO GENERAL

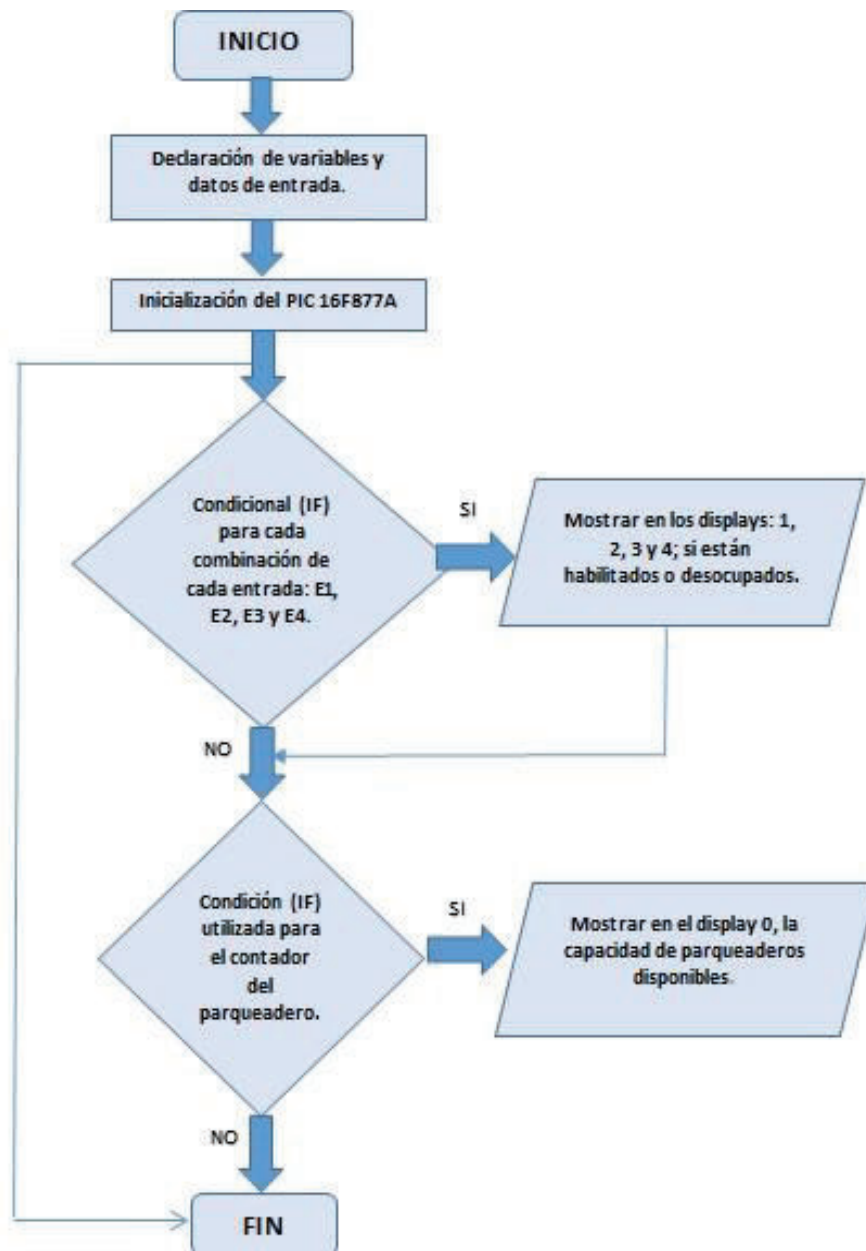


Figura 2.5 Diagrama de Flujo General

El diagrama de flujo general consta de tres etapas:

1. Etapa de declaración de variables
2. Condicionales
3. Representación de resultados.

1. ETAPA DE DECLARACIÓN DE VARIABLES:

Esta etapa consiste en el ingreso de los datos, puertos, definición de variables de entrada al PIC. Los sensores, cuando un vehículo está estacionado, emiten la señal que atraviesa al 74LS14P para invertirla y su información sea apta para transferirla hacia todos los pines del puerto A, de A0 hasta A3. Para que puedan ser validado y compilado, el resultado entre el programa MikroC y el PIC 16F877A es necesario que los datos sean los mismos y esté escrita la función que desempeñan ya que en esta etapa se encuentra la inicialización del PIC.

2. CONDICIONALES:

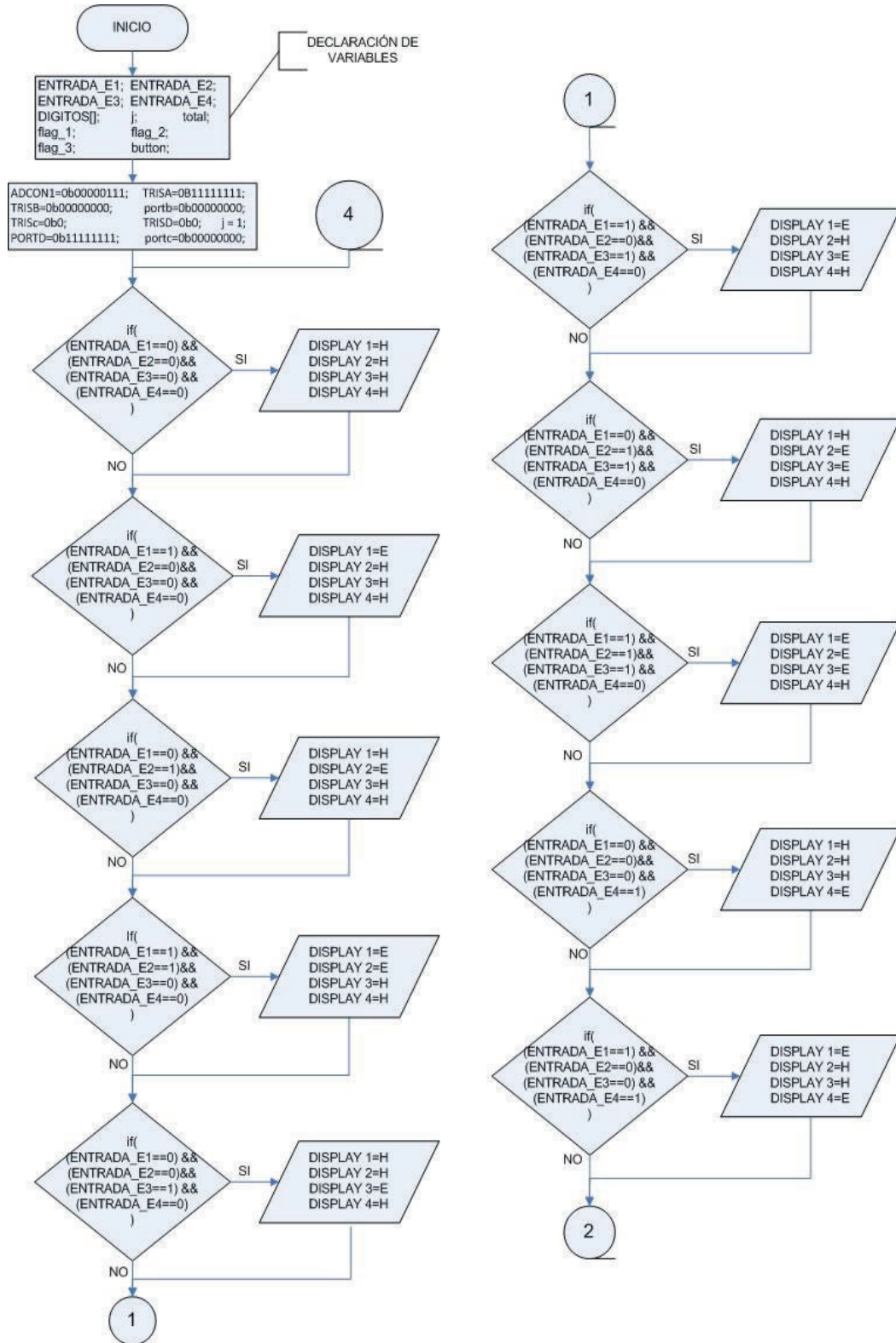
La información ingresada por los pines del microcontrolador, es transmitida y analizada para realizar el control del sistema de parqueos: conteo de los sitios habilitados y visualizaciones en el panel de displays. Esta etapa consiste en la generación de los bucles y condicionales generados con cada una de las combinaciones por el número de parqueaderos.

El bucle permanece mientras se efectúe cada combinación una vez culminado el proceso y no disponiendo de otra combinación el proceso llega a su fin. El bucle se maneja bajo el condicional IF.

3. REPRESENTACIÓN DE RESULTADOS:

En esta fase el resultado final que se haya generado en la anterior etapa por la combinación, se muestra en los dispositivos de salida: displays. El microcontrolador procesa la información emitida por el sensor y envía su resultado a través de los puertos B y D, y la presenta visualmente en el display del contador de sitios libres y panel de displays. Los dispositivos de salida indican el estado actual del parqueadero: habilitado u ocupado para que el usuario disponga de información al momento de requerir hacer uso del mismo.

2.1.3 DIAGRAMA DE FLUJO A DETALLE:



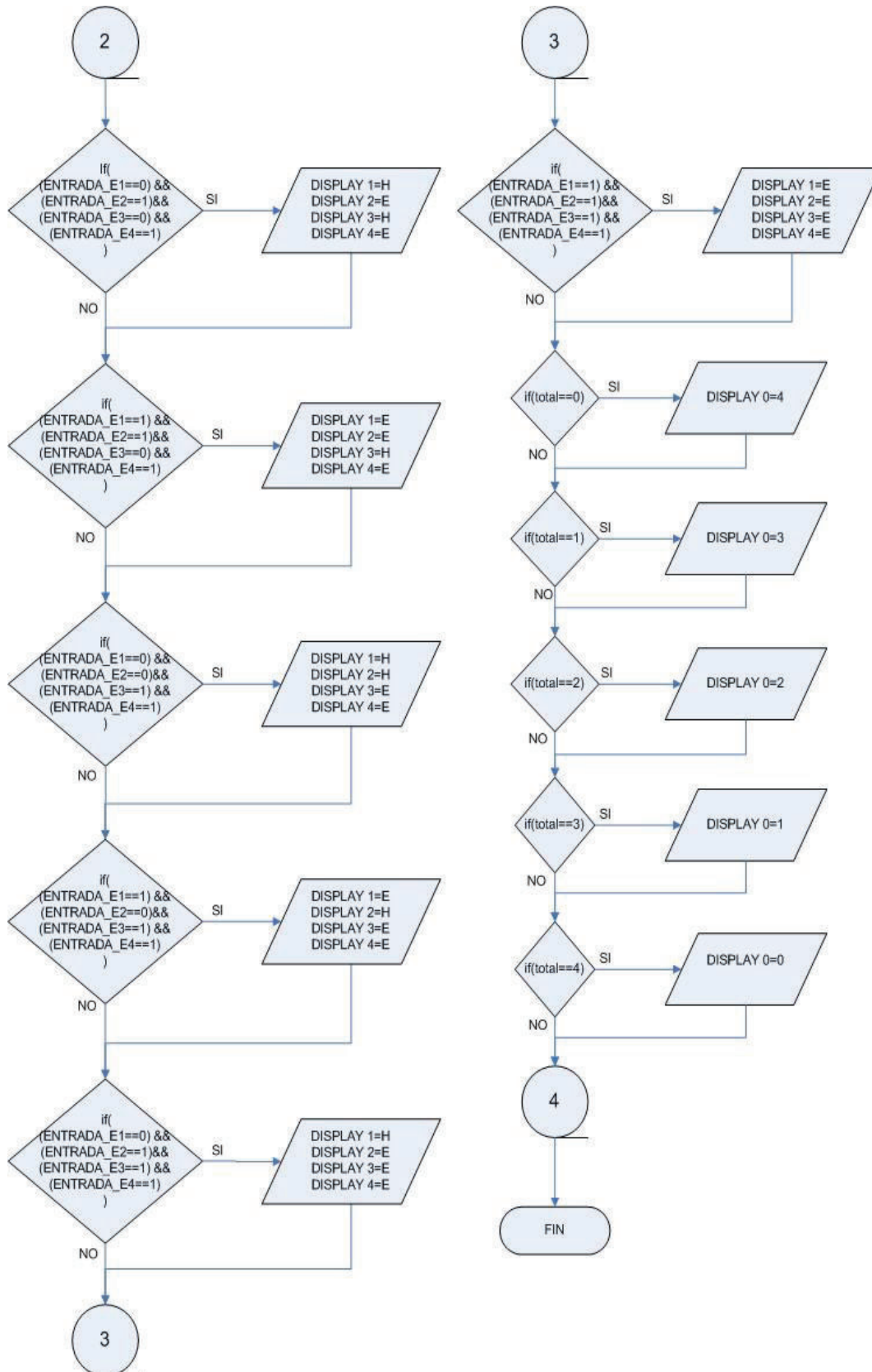
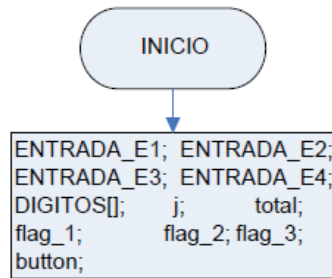


Figura 2.6. Diagrama de Flujo de Sistema de Control de Parqueos

2.1.4 CODIFICACIÓN DEL PROGRAMA EN EL MICRO:



```

#define ENTRADA_E1 PORTA.F0
#define ENTRADA_E2 PORTA.F1
#define ENTRADA_E3 PORTA.F2
#define ENTRADA_E4 PORTA.F3
// #define sens_int PORTA.F4 /
// #define sens_out PORTA.F4 const unsigned short DIGITOS [] =
{
0x3F, //Código del dígito 0
0x06, //Código del dígito 1
0x5B, //Código del dígito 2
0x4F, //Código del dígito 3
0x66, //Código del dígito 4
0x6D, //Código del dígito 5
0x7D, //Código del dígito 6
0x07, //Código del dígito 7
0x79, //Código del dígito 8 codigo E estacionado
0x76, //Código del dígito 9 codigo H habilitado
};
  
```

```

ADCON1=0b00000111; TRISA=0B11111111;
TRISB=0b00000000; portb=0b00000000;
TRISC=0b0; TRISD=0b0; j = 1;
PORTD=0b11111111; portc=0b00000000;
  
```

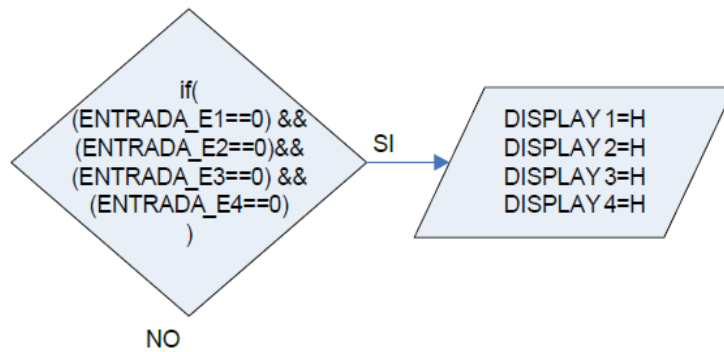
ADCON1=0b00000111; //CONVIERTO PUERTO A EN DIGITAL

TRISA=0B11111111; // PUERTO A COMO ENTRADA

```

TRISB=0b00000000;// puerto B como salida
portb=0b00000000;// puerto esta 0L
TRISC=0b0;// puerto B como salida
portc=0b00000000;
TRISD=0b0;
PORTD=0b11111111;
j = 1;

```



```

if((ENTRADA_E1==0) && (ENTRADA_E2==0)&&(ENTRADA_E3==0) &&
(ENTRADA_E4==0))
{ //1
TOTAL=0 ; //PARQUEADERO VACIO
PORTB = ~DIGITOS [9]; //Visualiza las unidades.
PORTd.F1=0; //Activa en alto el primer display
tiempo(A); //Retado de 10m segundos
PORTd.f1=1; //Desactiva todos los displays.
PORTB = ~DIGITOS [9]; //Visualiza las decenas.
PORTd.F2=0; //Activa en alto el primer display
tiempo(A); //Retado de 10m segundos
PORTd.f2=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las decenas.
PORTd.F3=0; //Activa en alto el segundo display
tiempo(A); //Retado de 10m segundos
PORTd.f3=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las centenas.

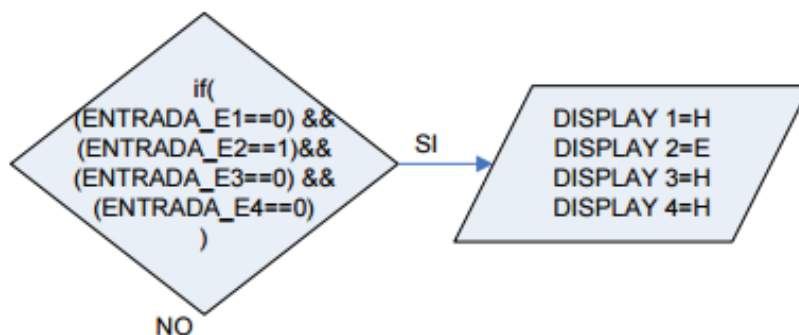
```



```

PORTd.F4=0; //Activa en alto el tercer display
tiempo(A); //Retado de 10m segundos
PORTd.f4=1; //Desactiva todos los displays.
}
TOTAL=1 ;
PORTB = ~DIGITOS[8]; //Visualiza las unidades.
PORTd.F1=0; //Activa en alto el primer display
tiempo(A); //Retado de 10m segundos
PORTd.f1=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las decenas.
PORTd.F2=0; //Activa en alto el primer display
tiempo(A); //Retado de 10m segundos
PORTd.f2=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las decenas.
PORTd.F3=0; //Activa en alto el segundo display
tiempo(A); //Retado de 10m segundos
PORTd.f3=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las centenas.
PORTd.F4=0; //Activa en alto el tercer display
delay_ms(5); //Retado de 10m segundos
PORTd.f4=1; //Desactiva todos los displays.
}

```



```

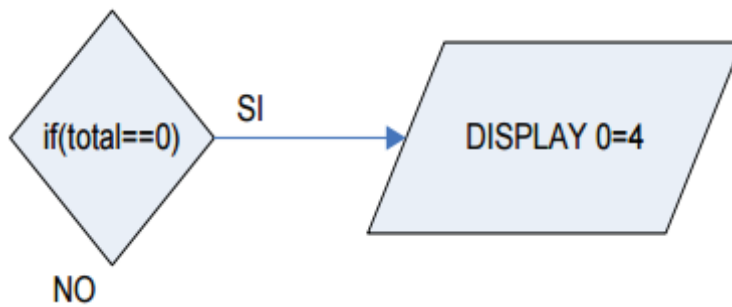
}
PORTB = ~DIGITOS[8]; //Visualiza las decenas.
PORTd.F2=0; //Activa en alto el primer display

```

```

tiempo(A); //Retado de 10m segundos
PORTd.f2=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[8]; //Visualiza las decenas.
PORTd.F3=0; //Activa en alto el segundo display
tiempo(A); //Retado de 10m segundos
PORTd.f3=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[8]; //Visualiza las centenas.
PORTd.F4=0; //Activa en alto el tercer display
delay_ms(5); //Retado de 10m segundos
PORTd.f4=1; //Desactiva todos los displays.
}

```



```

if(total==0)
{
PORTB = ~DIGITOS[4];
PORTd.F0=0; //Activa en alto el cuarto display
tiempo(A); //Retado de 10m segundos
PORTd.f0=1; //Desactiva todos los displays.
}

```

2.1.5. DIAGRAMA CIRCUITAL DEL SISTEMA DE CONTROL PARQUEOS

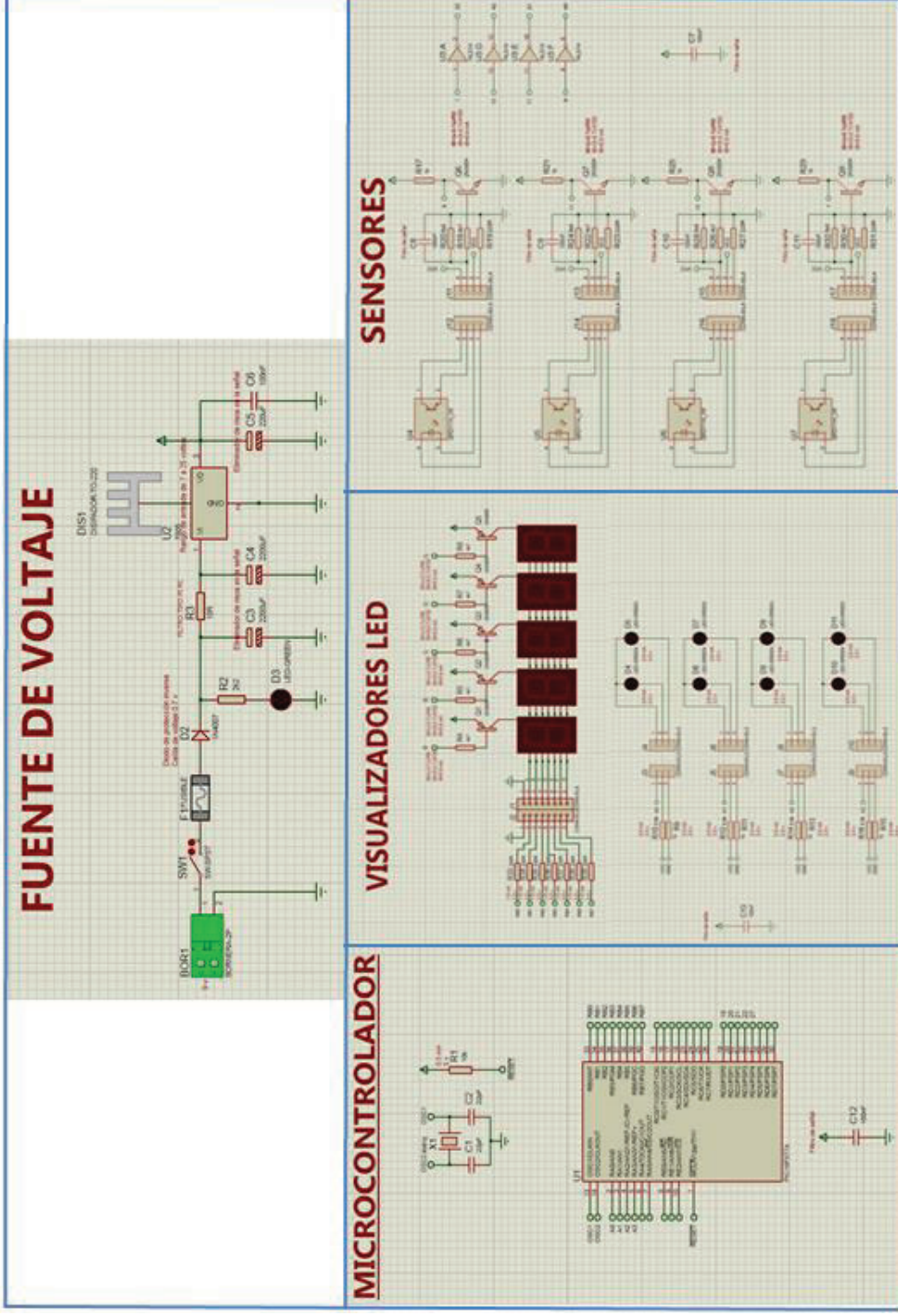


Figura 2.7. Diseño Circuitual

El microcontrolador 16F877A es mucho más eficaz y eficiente que sus homólogos, como el microcontrolador 16F876. Para el presente proyecto se utilizó el 16F877A debido a su número de 40 pines para entradas y/o salidas. Su ampliación respecto a líneas de programación su capacidad de memoria flash de 8Kb, memoria de datos RAM de 368 bytes, memoria de datos EEPROM 256 bytes, Convertidor Análogo/Digital de 10 bits multicanal (8 canales de entrada) Voltaje de alimentación: 3.0 a 5.5 V DC, su consumo de corriente de 2Ma. El microcontrolador posee mejores prestaciones como sus 356 líneas de programación y la facilidad de su familia en cuanto a manejo de su programación en Mikro C que es accesible y fácil de manejar por los programadores en cuanto a su facilidad de instrucciones y sentencias.

Los sensores infrarrojos QRD1114 permiten, a diferencia de la mayoría de sensores infrarrojos, mejorar su recepción de la señal, una detección de las superficies sin tener un contacto directo, campo de visión más amplio en distancias respecto al objeto y al sensor, filtro de luz del día en el sensor, consumo de corriente de 50mA y un manejo de voltaje en su IRLED y fototransistor independiente cada uno de 5V.

Los display de 7 segmentos son visualizadores mucho más accesibles para los usuarios, de buena luminosidad, fácil configuración y manejo de sus entradas, consumo por dígito de 50 mW, y su voltaje de 1,7 V para displays rojos. Su implementación en el proyecto radica en la facilidad para visualizar los resultados normales y de barridos, y la accesibilidad en el mercado para su implementación a diferencia de un Lcd que requiere más instrucciones y un manejo más complejo de su estructura que no siempre puede ser incorporado fácilmente.

Los leds de alta potencia poseen una mayor potencia que los leds normales, teniendo un mejoramiento en su eficiencia energética que determina la cantidad de electricidad que el led emplea en convertirla en luz, excelente flujo luminoso, alta eficiencia, voltaje de consumo regular de 2.1 V, potencia de 1W, mínima emisión de radiaciones.

2.1.5.1 Funcionamiento electrónico del estacionamiento:

Al momento de que un usuario ingresa al parqueadero el sensor QRD1114 detecta una señal de OL, circula una corriente aproximada de 20mA por el diodo interno a la resistencia 4.7kohm a 5V, mientras que el transistor que es la salida se conecta el emisor a GND y el colector necesita una resistencia denominada pull-up con el valor de 5.7Kohm, la cual se conecta a 5V. La salida va a ser cercana a OV cuando hay un objeto reflectivo (automóvil) delante del sensor. En ausencia de objeto delante la salida va a ser cercana a 5V (asumiendo que la tensión de alimentación es 5V).

Después de pasar por el sensor la señal resultante ingresa al inversor 74LS14P para que invertirla, debido a que el PIC 16f877A funciona con 1L.

Una vez invertida la señal ingresa al PIC para poder procesar la información de los sensores y que sea transferida hacia los dispositivos de salida: displays y leds.

2.2. PROGRAMACIÓN DEL SISTEMA DE CONTROL DE PARQUEOS

En el desarrollo de la programación se tuvieron que realizar algunos cambios debido a la capacidad del PIC 16F877A y el programa compilador.

2.2.1. TABLA DE VERDAD DEL PARQUEADERO:

Para desarrollar cada una de las programaciones fue necesario realizar la tabla de verdad para poder verificar todas las combinaciones posibles, según el número de parqueaderos y poder verificar la capacidad de la memoria.

Tabla de verdad para 4 parqueos incluido el inversor:

#	Parqueaderos				Contador	TOTAL DISPONIBLE
	EN 1	EN 2	EN 3	EN 4		
1	1	1	1	1		0
2	0	1	1	1		1
3	1	0	1	1		1
4	0	0	1	1		2
5	1	1	0	1		1
6	0	1	0	1		2
7	1	0	0	1		2
8	0	0	0	1		3
9	1	1	1	0		1
10	0	1	1	0		2
11	1	0	1	0		2
12	0	0	1	0		3
13	1	1	0	0		2
14	0	1	0	0		3
15	1	0	0	0		3
16	0	0	0	0		4

Combinaciones

1L = OCUPADO
= ROJO = 8 E

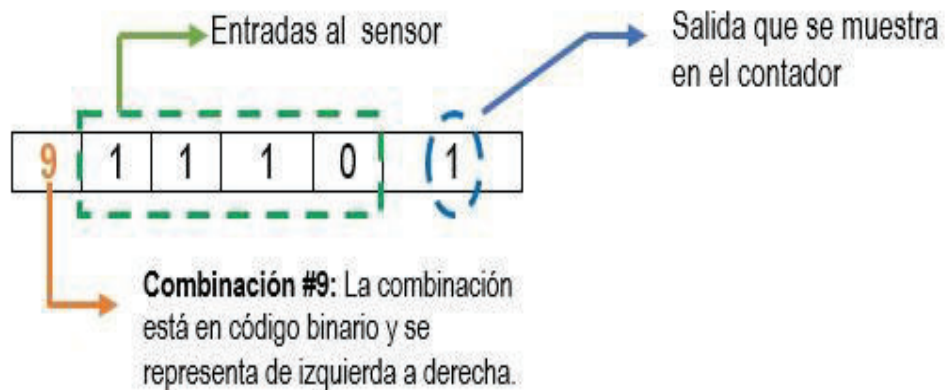
0L
= DESOCUPADO
= VERDE = 9 H

Entrada

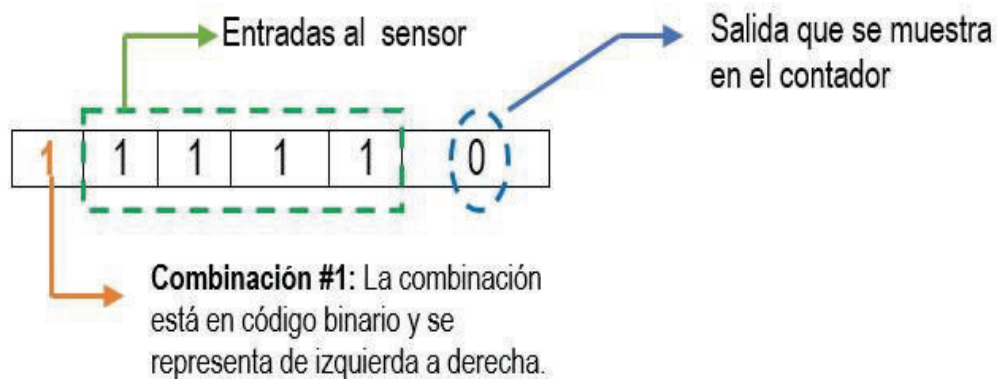
Salida

Tabla 2.1. Tabla de Verdad para 4 Parqueos

Para la verificación del funcionamiento de la tabla antes descrita, se valida con 2 ejemplos:

Ejemplo 1:**Figura 2.8.** Ejemplo 1

En la combinación # 9 de la tabla mostrada se tiene el código binario **1110**, el cual representa la posición de cada auto en el parqueadero. La posición 4 indica que está habilitado y el resto está ocupado; esto quiere decir que la disponibilidad de parqueaderos se reduce a 1.

Ejemplo 2:**Figura 2.9.** Ejemplo 2

En la combinación # 1 de la tabla mostrada se tiene el código binario **1111**, el cual representa la posición de cada auto en el parqueadero. Las posiciones están ocupadas; esto quiere decir que la disponibilidad es cero.

2.2.2. PROGRAMACIÓN:

Para la implementación y validación del circuito, se realizaron múltiples pruebas de la programación a implementarse, tomándose en cuenta la capacidad de memoria del PIC, las necesidades requeridas y los elementos empleados.

Por tal razón fueron necesarias algunas pruebas las cuales facilitaron identificar los errores cometidos y corrección de los mismos. A continuación se muestra el programa corregido y final elaborado en MikroC.

2.2.2.1. PROGRAMACIÓN FINAL: 4 PARQUEADEROS:

```
#define ENTRADA_E1 PORTA.F0
#define ENTRADA_E2 PORTA.F1
#define ENTRADA_E3 PORTA.F2
#define ENTRADA_E4 PORTA.F3
//#define sens_int PORTA.F4
//#define sens_out PORTA.F4 const unsigned
short DIGITOS[] = {
0x3F, //Código del dígito 0
0x06, //Código del dígito 1
0x5B, //Código del dígito 2
0x4F, //Código del dígito 3
0x66, //Código del dígito 4
0x6D, //Código del dígito 5
0x7D, //Código del dígito 6
0x07, //Código del dígito 7
```

Definición de variables
de entrada al PIC


```
0x79, //Código del dígito 8 codigo E estacionado
0x76, //Código del dígito 9 codigo H habilitado
tiempo (int A) {
delay_us(3000); //3 ms
rutina(int B)
}
void main() {
unsigned short A; //short, signed o int es lo mismo
//short b;
//signed c;
//short d; //short e; short total;
char j; bit flag_1;
bit flag_2;
bit flag_3;
bit button;
```

Definición de
variables de
entrada al PIC

```
ADCON1=0b00000111;//CONVIERTO PUERTO A EN
DIGITAL
TRISA=0B11111111; // PUERTO A COMO ENTRADA
TRISB=0b00000000;// puerto B como salida
portb=0b00000000;// puerto esta 0L
TRISC=0b0;// puerto B como salida
portc=0b00000000;
TRISD=0b0;
PORTD=0b11111111; //PWM1_Init(5000); //5000 Hz
//PWM1_Start();
// start PWM
j = 1;
```

Inicialización
del PIC

```
while(1)
{
if((ENTRADA_E1==0) &&
(ENTRADA_E2==0)&&(ENTRADA_E3==0)&&
(ENTRADA_E4==0)&&(ENTRADA_E5==0))
{//1
TOTAL=0 ; //PARQUEADERO VACIO
PORTB = ~DIGITOS[9]; //Visualiza las unidades.
PORTd.F1=0;
//Activa en alto el primer display tiempo(A);
//Retado de 10m segundos
PORTd.f1=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las decenas.
PORTd.F2=0; //Activa en alto el primer display tiempo(A);
//Retado de 10m segundos PORTd.f2=1; //Desactiva todos
los displays.
PORTB = ~DIGITOS[9]; //Visualiza las decenas.
PORTd.F3=0; //Activa en alto el segundo display
tiempo(A); //Retado de 10m segundos
PORTd.f3=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las centenas.
PORTd.F4=0; //Activa en alto el tercer display
tiempo(A); //Retado de 10m segundos
PORTd.f4=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las unidades de mil.
PORTd.F5=0; //Activa en alto el cuarto display
tiempo(A); //Retado de 10m segundos
PORTd.f5=1; //Desactiva todos los displays.
}
```

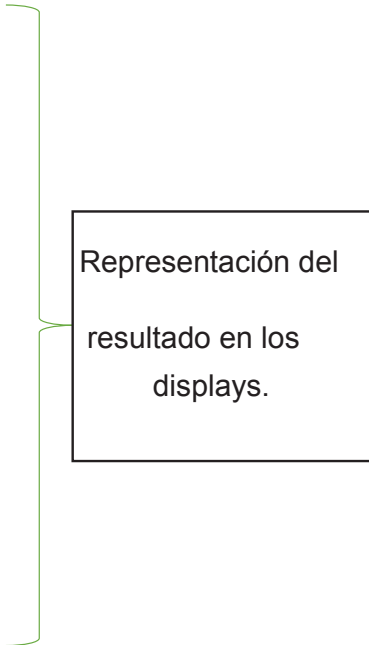
Condiciones
y sentencias
para cada
combinación

```

if(total==0)
PORTB = ~DIGITOS[4];
PORTd.F0=0; //Activa en alto el cuarto display
tiempo(A); //Retado de 10m segundos
PORTd.f0=1; //Desactiva todos los displays.
}

if(total==4) {
PORTB = ~DIGITOS[0];
PORTd.F0=0; //Activa en alto el cuarto display
tiempo(A); //Retado de 10m segundos
PORTd.f0=1; //Desactiva todos los displays.
}

```



Representación del
resultado en los
displays.

La programación completa del circuito verificar en el ANEXO F.

2.2.3. SIMULACIÓN DEL SISTEMA DE CONTROL DE PARQUEOS

Los circuitos presentados a continuación se muestran con cada una de las pruebas realizadas en el PROTEUS para los casos presentados en la programación para la culminación del circuito del PIC 16F877A:

2.2.3.1. PROGRAMA FINAL: 4 PARQUEADEROS

Por la capacidad del PIC 16F877A al momento de programar y realizar las diversas pruebas con ayuda del simulador, se procede a verificar que lo más óptimo es dejar la programación para 4 parqueaderos debido a la repetición del barrido de los displays y la capacidad del microcontrolador conocido.

Tomar en cuenta que adicional se maneja el contador y por ende son 5 displays y no 4. Las líneas de programación deben satisfacer las condiciones de barrido y capacidad del PIC, para satisfacer correcta y eficientemente el diseño.

A continuación se muestra el proyecto final:

Proyecto Final:

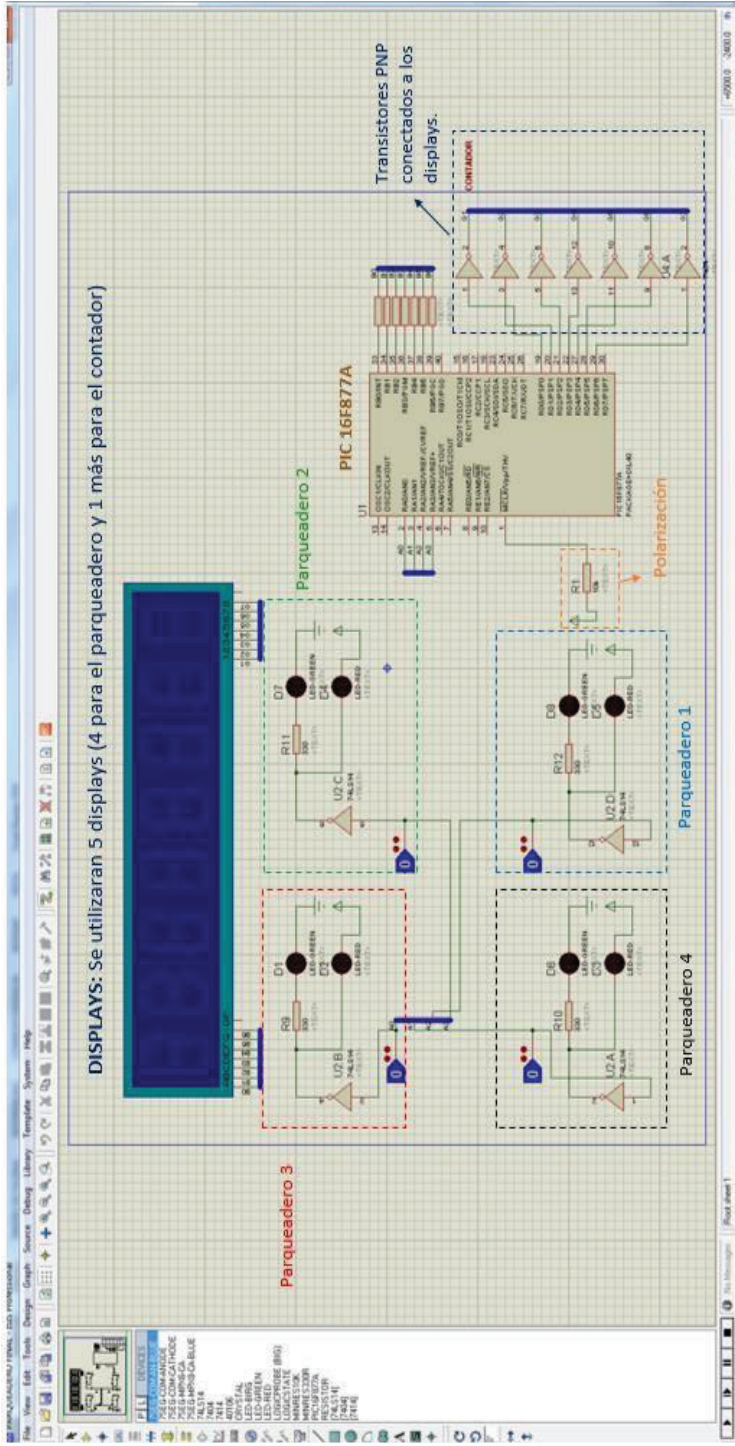


Figura 2.10. Circuito Final

En la figura 2.10 se observa la simulación del proyecto final realizada en PROTEUS; el PIC 16F877A valida las instrucciones ingresadas a través del MikroC para poder enviar la información a todos los dispositivos, los displays utilizados fueron 5: 4 displays usados para cada parqueadero y 1 display que realiza la función del contador. Los displays visualizan los parqueaderos habilitados, ocupados y el conteo de los mismos al instante que ingresa o sale un automóvil.

Sitios libre en el parqueadero:

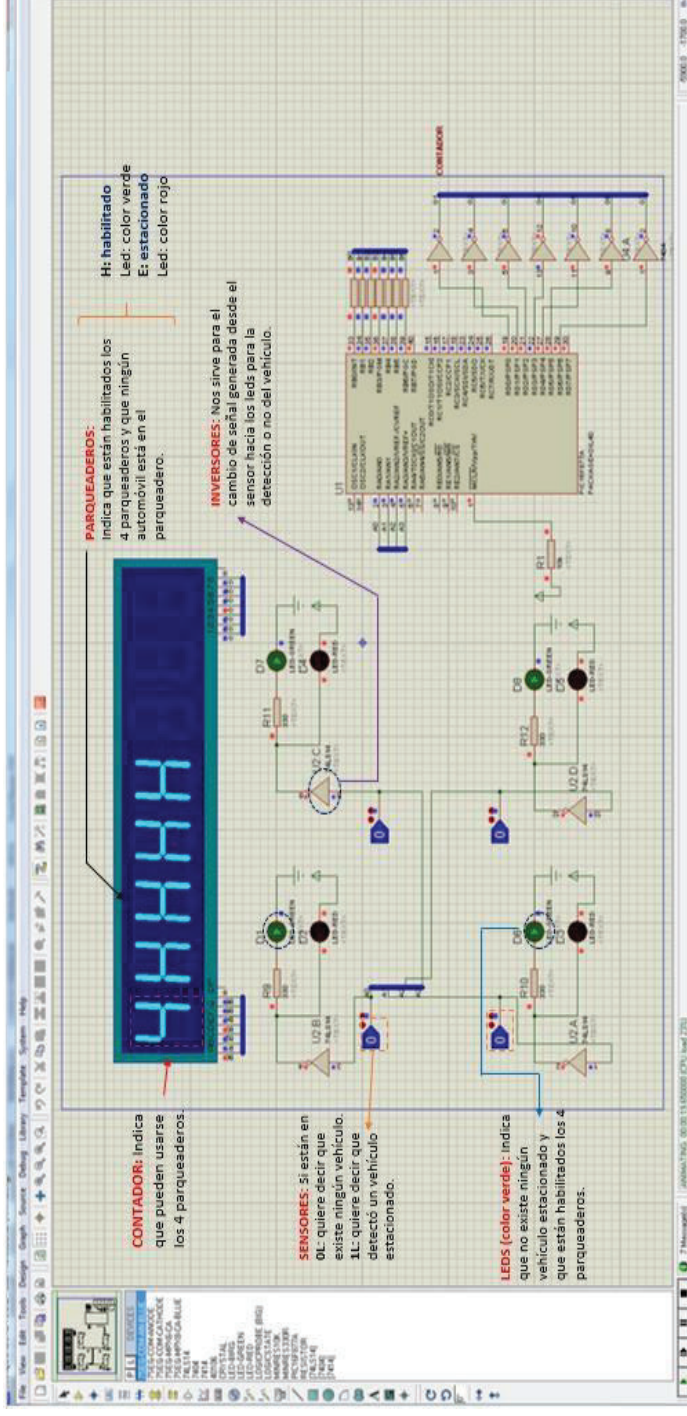


Figura 2.11. Sitios Libres en el parqueadero

En la figura 2.11 se puede apreciar que no existe ningún automóvil estacionado ya que los displays indican que los 4 parqueaderos están en H: habilitados (OL: parqueaderos desocupados), el contador indica que los 4 parqueaderos pueden ser usados para que el usuario disponga de esa información y sepa el lugar donde se puede estacionar.

Los leds verdes son indicativos de que los parqueaderos están disponibles. Finalmente los sensores en OL: indican que no detectan ningún vehículo estacionado y que existen sitios libres.

Utilización del estacionamiento:

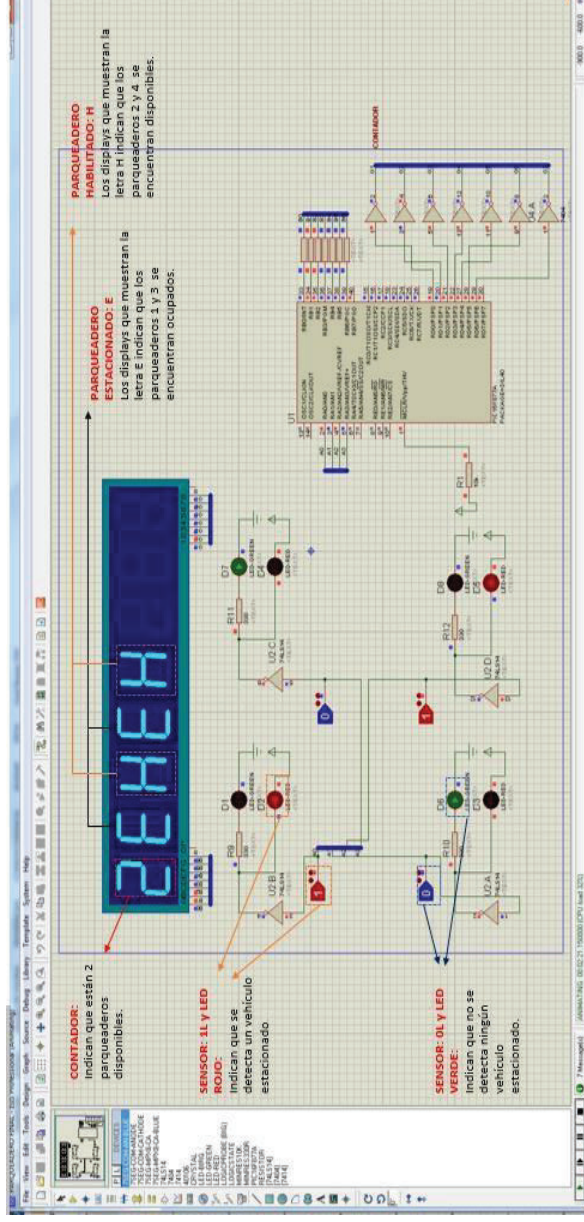


Figura 2.12. Estacionamiento 1 y 3 ocupados

En la figura 2.12 se visualiza que existen 2 parqueaderos ocupados y 2 parqueaderos disponibles en el estacionamiento, lo cual se verifica mediante el uso del contador ya que indica la capacidad de parqueaderos disponibles. Los parqueaderos disponibles son los parqueaderos 2 y 4 que se reflejan en el panel de control de los displays con la letra H, de igual manera para los parqueaderos ocupados se verifica que son los parqueaderos 1 y 3 representados por la letra E. Finalmente se aprecia que los sensores conjuntamente con los leds indican el estado de cada parqueadero al momento de ingresar o salir un usuario en el estacionamiento.

Sitios estacionados:

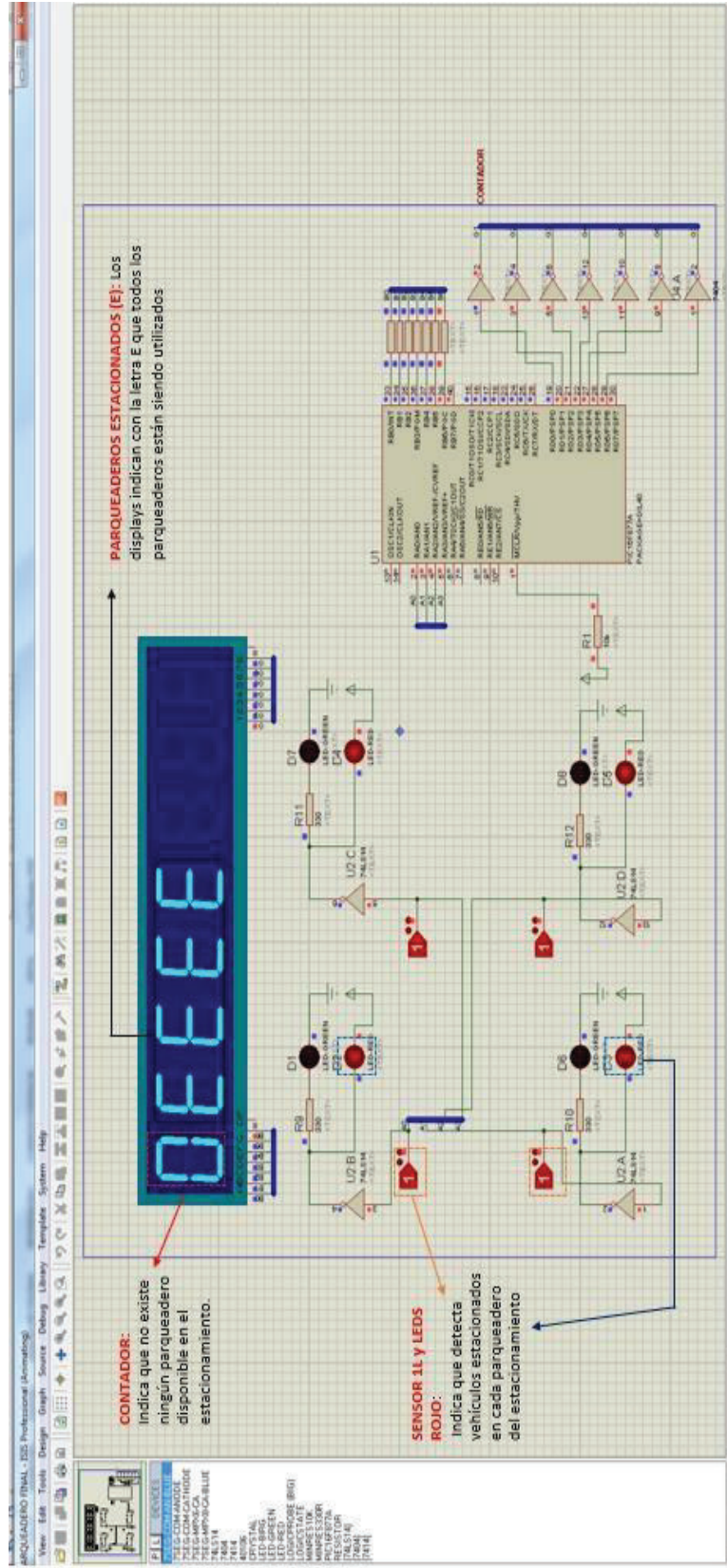


Figura 2.13. Sitios estacionados

En la figura 2.13 se puede apreciar que no existe ningún sitio disponible en el estacionamiento debido a que todos los parqueaderos se encuentran ocupados. Los leds rojos son indicativos de que los parqueaderos no están disponibles. Finalmente los sensores en 1L: indican que se detectan vehículos estacionados en cada uno de los parqueaderos.

CAPÍTULO III: IMPLEMENTACIÓN

3.1 ELABORACIÓN DEL PROTOTIPO DEL SISTEMA DE CONTROL DE PARQUEOS

3.1.1. PROTOTIPO IMPLEMENTADO EN PROTOBOARD

La elaboración del prototipo del sistema de control de parqueos, se realizó en protoboard para verificar y corregir posibles fallas y errores en conexiones y elementos. Su desarrollo consta de diversa etapas descritas a continuación.

3.1.1.1. Ubicación del Regulador de Voltaje

Como siguiente paso se adecua la ubicación de un regulador de voltaje que se describió en el capítulo II anterior.

El regulador de voltaje L7805 admitirá una señal de entrada de voltaje hasta 12 voltios, suministrando un voltaje de 5 voltios. El voltaje de salida del regulador será el voltaje de entrada para el circuito (placa del circuito impreso) que se encuentra desarrollado en el protoboard.

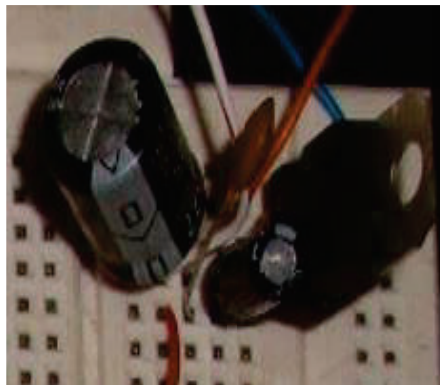


Figura 3.1. Regulador de voltaje

3.1.1.2. Instalación de los sensores

Esta etapa corresponde al dominio de conocimientos respecto a los sensores infrarrojos opto-reflexivos QRD1114, su conectividad y funcionamiento se verá reflejada en diodos emisores de luz (LEDS) de colores verde y rojo.

Su función es indicar si el sitio de parqueo está Habilitado (H) o estacionado (E) respectivamente. Para que los leds funcionen de forma óptima se necesita el uso de un circuito integrado 74LS14P que genere los niveles de señal OL Y 1L y se enciendan los leds adecuadamente. Su esquema es el siguiente:

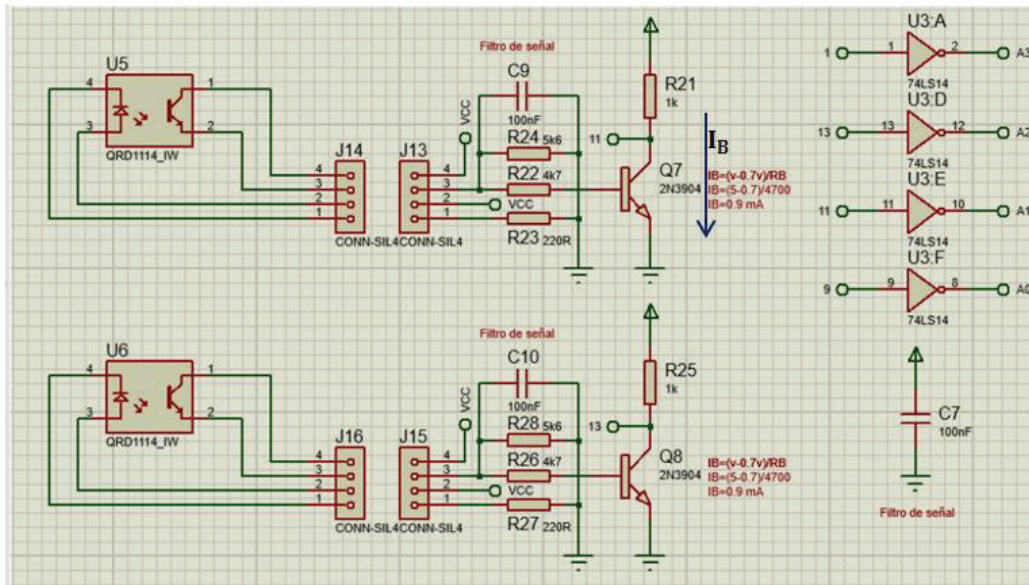


Figura 3.2. Diagrama circuital de un solo sensor

- *Cálculos:*

Datos:

$$V_{entrada} = 5 \text{ V}$$

$$I_B = ?$$

Por aplicación de transistores:

$$V_E = 0 \text{ V}$$

$$V_B = 0,7 \text{ V}$$

$$V_C = V_{CC} - I_C R_C$$

$$I_B = \frac{V_{CC} - 0,7}{R_B}$$

$$I_B = \frac{V_{CC} - 0,7}{R_B} = \frac{5 - 0,7}{4,7 \text{ K}\Omega} = 0,9 \text{ mA}$$

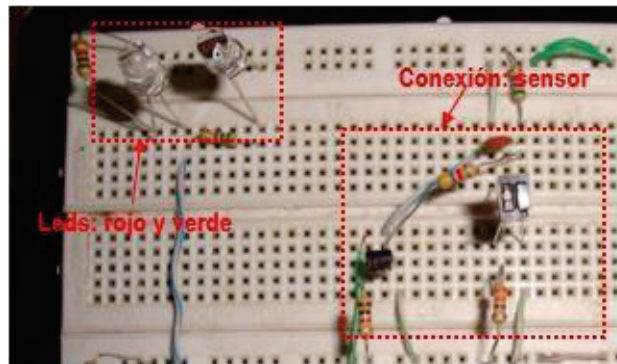
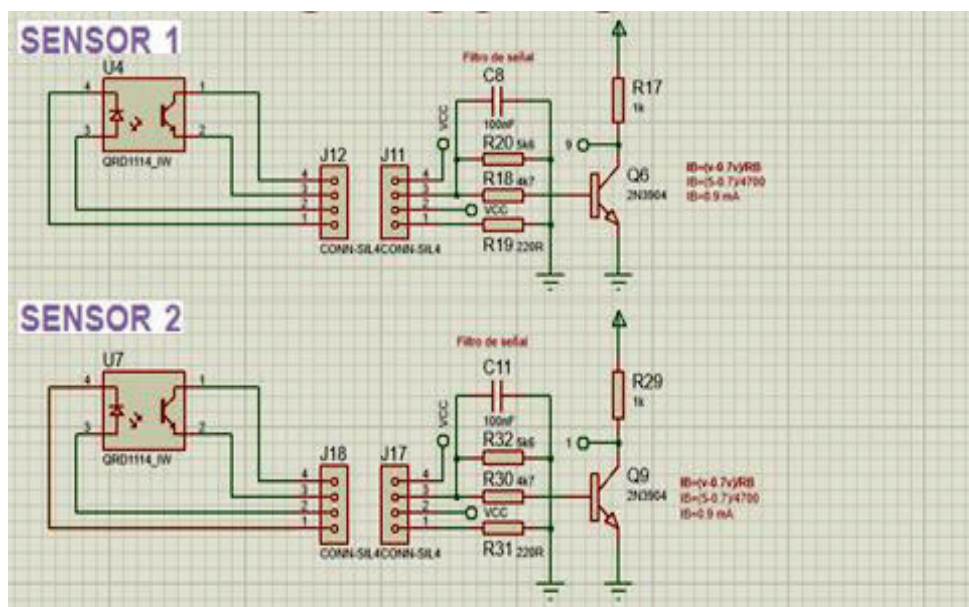


Figura 3.3. Conexión total de un solo sensor con leds

El mismo procedimiento de implementación se desarrollará para los tres siguientes sitios de parqueos dentro de un estacionamiento, siguiendo un orden secuencial.



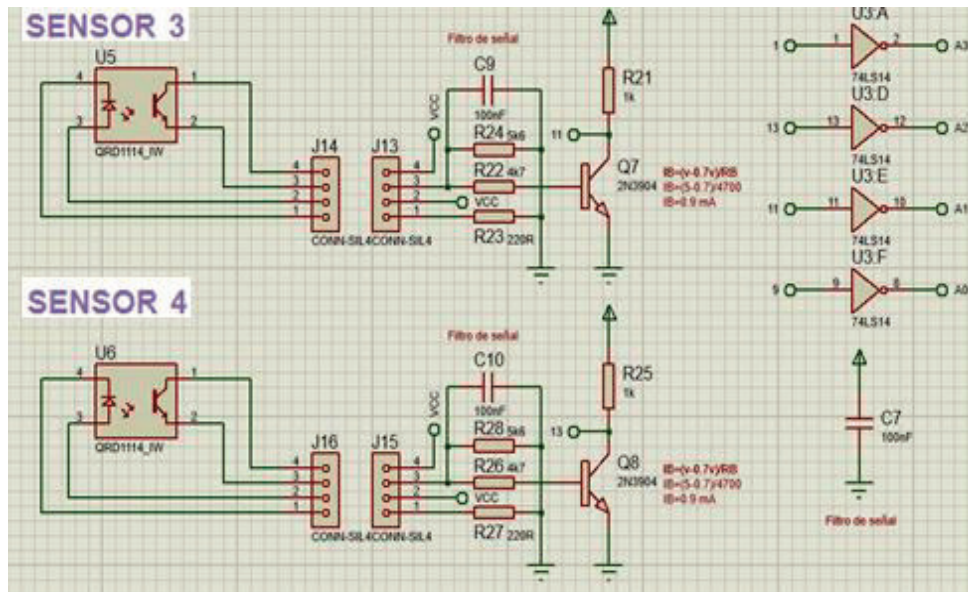


Figura 3.4. Diagrama circuital total

La figura 3.4. indica el diagrama de la construcción total de los 4 parqueaderos con cada uno de sus componentes.

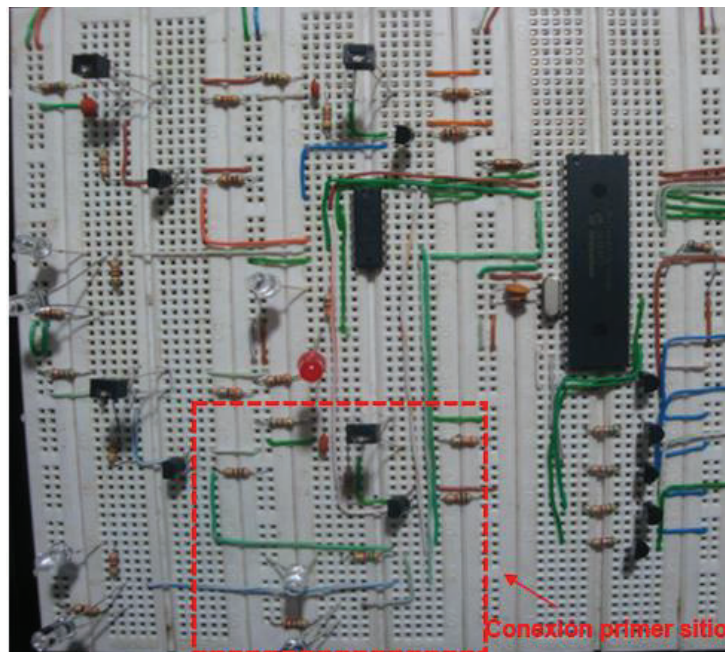


Figura 3.5. Implementación total de los 4 sensores en protoboard

La figura 3.5. muestra la implementación de todos los parqueaderos en forma secuencial de acuerdo al diseño.

3.1.1.3. Conexión del Microcontrolador y Displays

La conexión del microcontrolador se dispondrá de acuerdo a la necesidad de las entradas de las señales emitidas por los sensores a través del circuito integrado 74LS14P; las salidas corresponderán a la distribución de señales tanto para:

- el contador UP/DOWN (ascendente/descendente) que se ubicará a la entrada del parqueadero (prototipo), y
- para el panel de control que contiene los displays, los cuales informarán al usuario acerca de los sitios de parqueo indicando si están habilitados (H) o estacionados (E) respectivamente.

El microcontrolador manejará la programación realizada en MikroC, cuyo objetivo es de emitir la información, de los sitios habilitados o estacionados, recibida de las señales de los sensores.

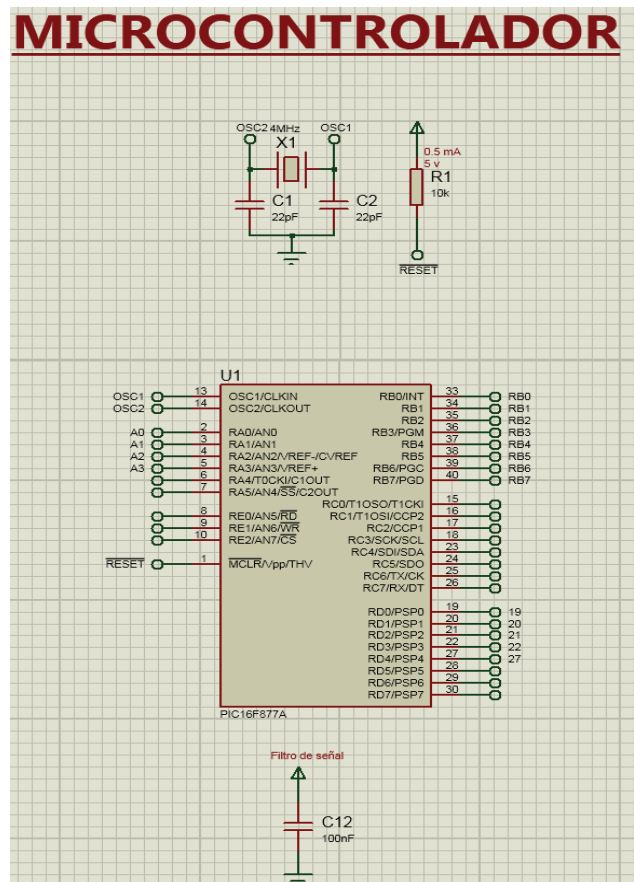


Figura 3.6. Diagrama Circuitual Microcontrolador

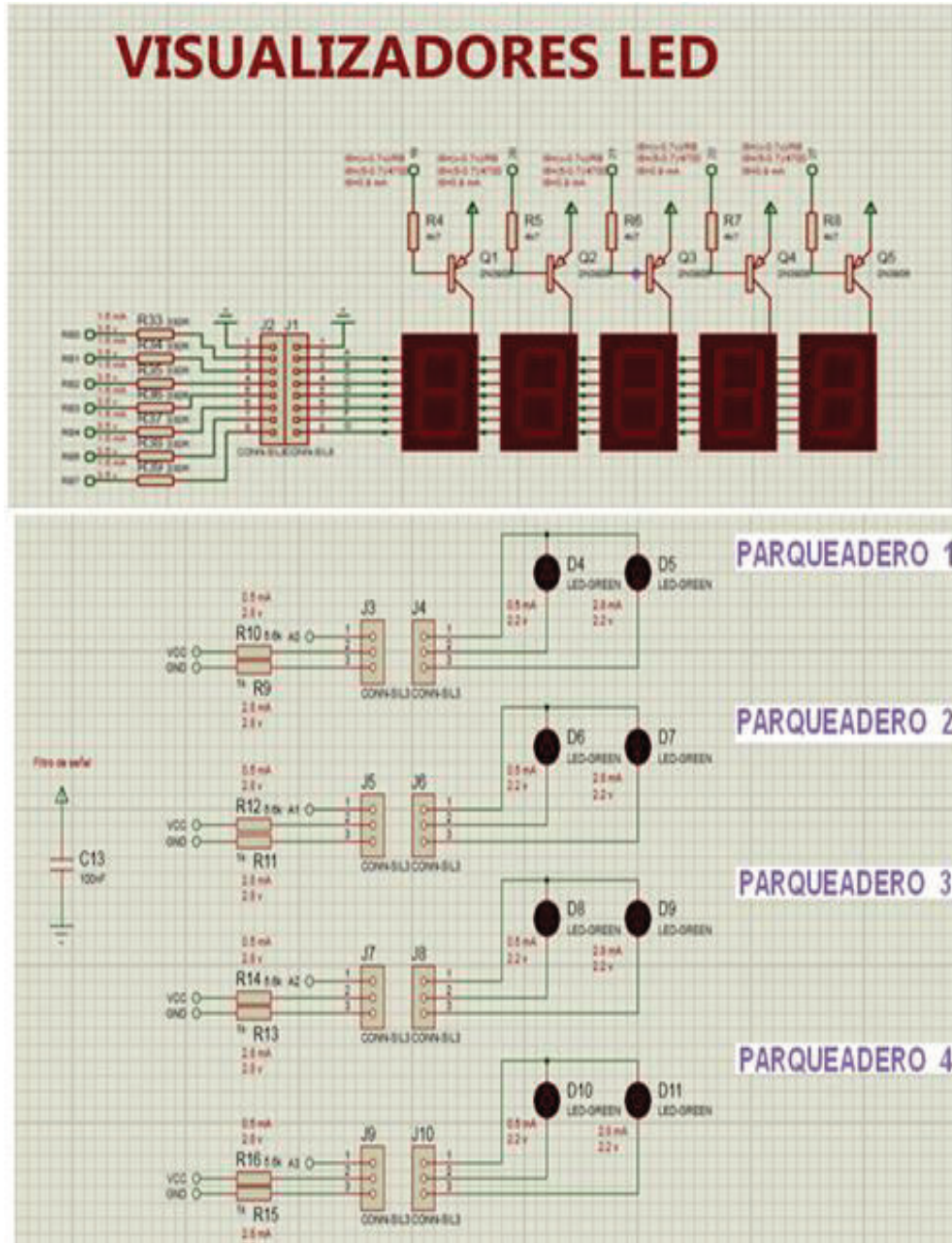


Figura 3.7. Diagrama Circuital de los Displays

Las figuras 3.6 y 3.7, especifican la esquematización del circuito del microcontrolador con los displays.

La siguiente figura 3.8., muestra la conexión de cada salida del microcontrolador con los displays del contador y del parqueadero.

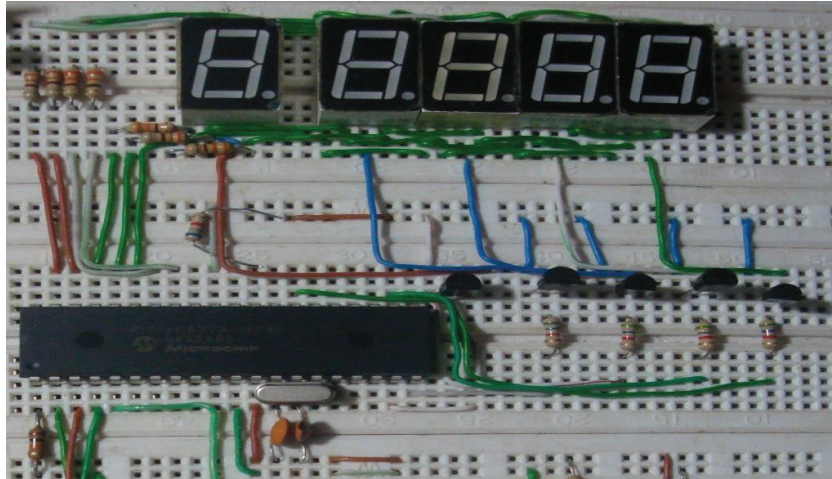


Figura 3.8. Implementación de Microcontrolador y Displays

La totalidad de la implementación presentará una distribución de las tres etapas, antes mencionadas, en el protoboard así como las conexiones entre ellas y polarizaciones de los respectivos elementos utilizados.

El siguiente gráfico muestra el conjunto de las tres etapas: ubicación del regulador de voltaje, instalación de los sensores y conexión del microcontrolador y displays.

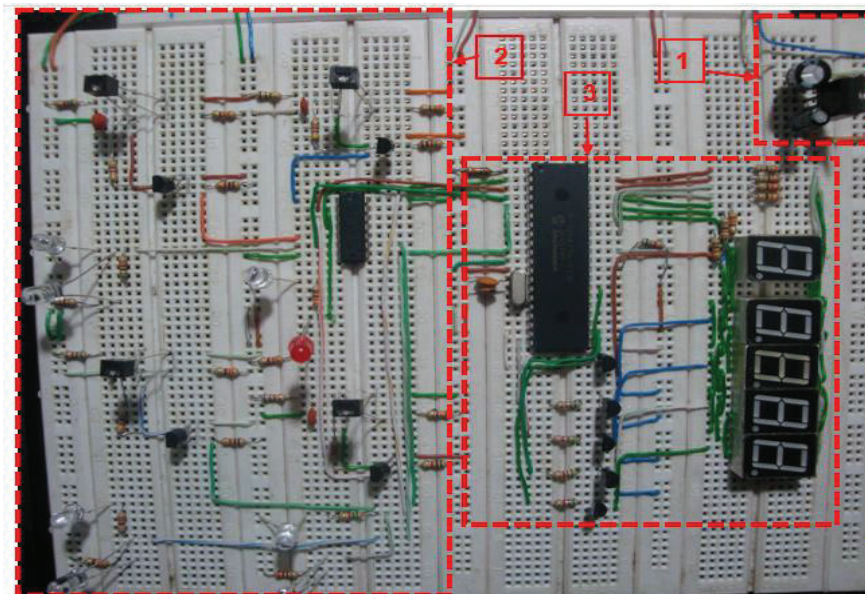


Figura 3.9. Implementación total de prototipo

La figura 3.10 muestra el diagrama circuital de las tres etapas unificadas.

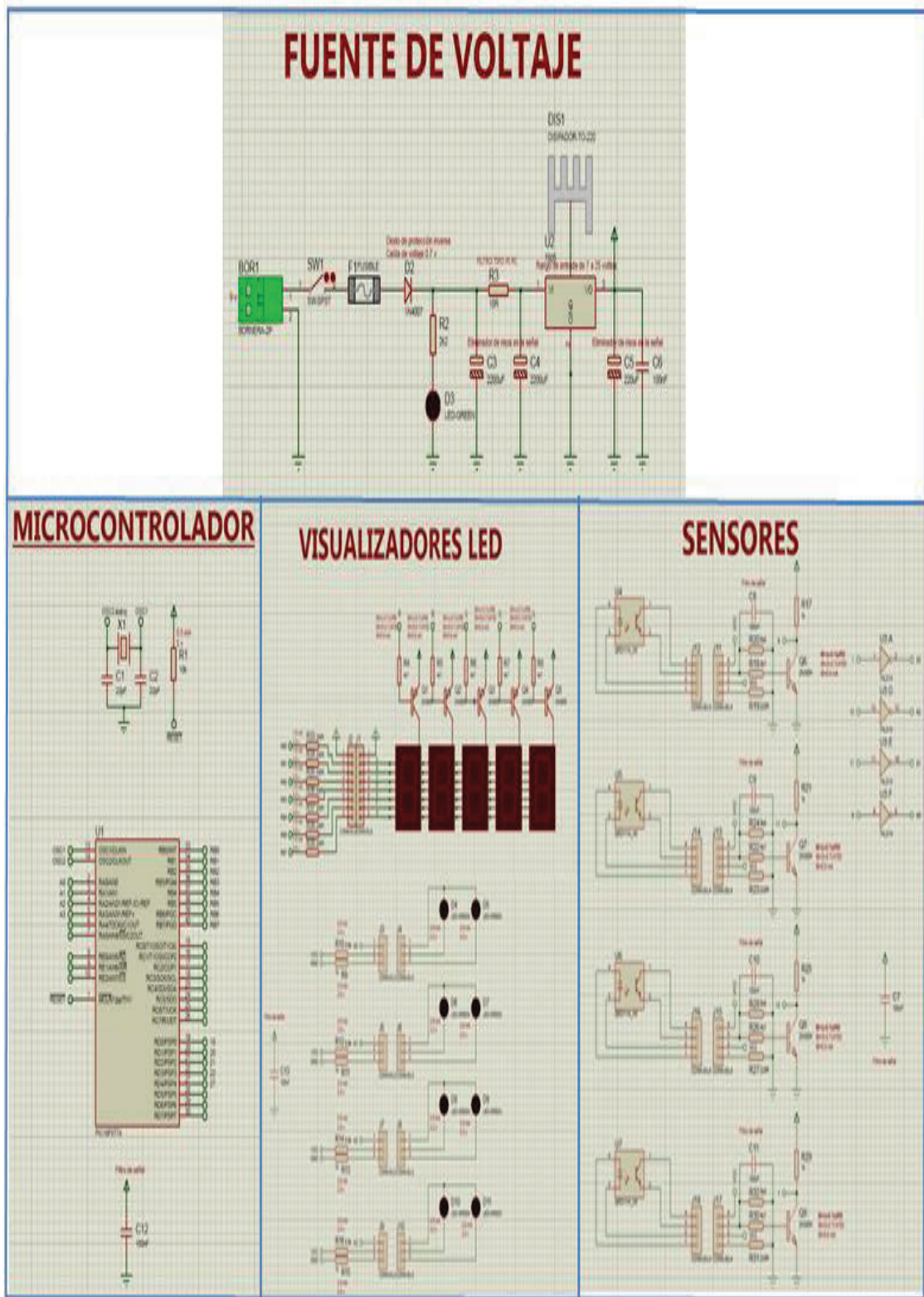


Figura 3.10 Diagrama circuital de la implementación total de prototipo

3.1.1.4. Pruebas de Funcionamiento del Prototipo Implementado en el Protoboard

Con la finalidad de discernir posibles fallas y controlarlas, antes de diseñar la placa principal del sistema de control de parqueos y demás circuitos, se procedió a probar su funcionalidad en protoboard. Una vez descartados los errores que se presentaron, los cuales se indican en la sección 3.3.1.Pruebas, se comprobó el adecuado funcionamiento de todos los diagramas y programación realizados.

Las siguientes figuras 3.11., 3.12., 3.13., muestran las pruebas realizadas:

Sitios habilitados: muestra la totalidad de los sitios de parqueos disponibles; los leds de color verde indican su disponibilidad, y en el panel de displays la letra H que significa que el sitio se encuentra habilitado o disponible.

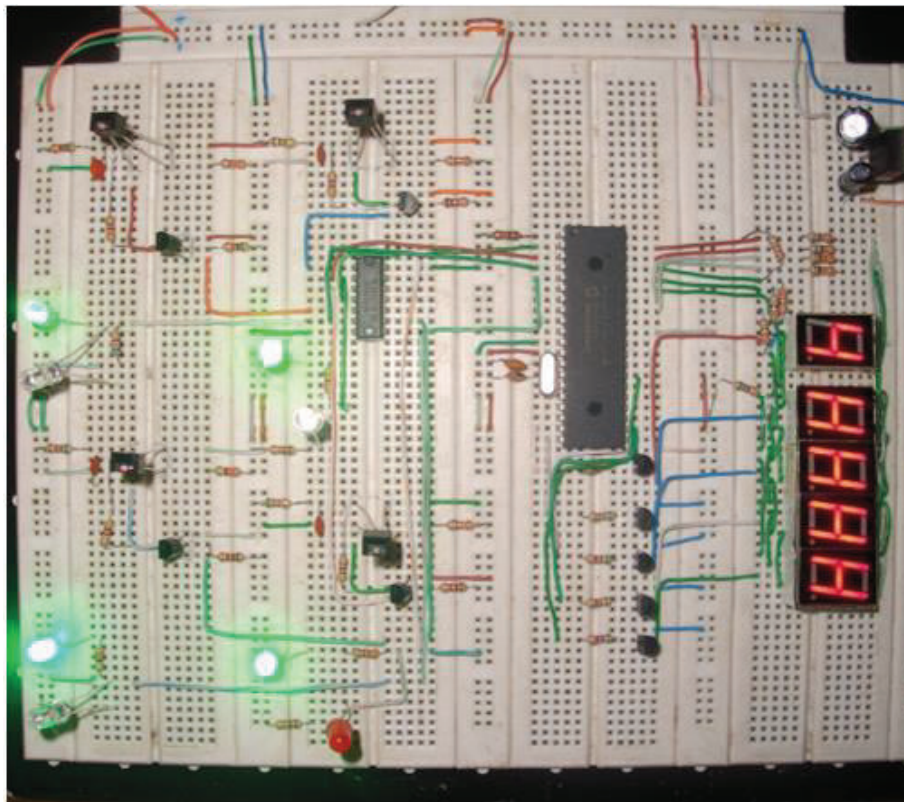


Figura 3.11. Sitios de parqueos habilitados

Contador decreciente: se indica el primer estacionamiento ocupado activando el led rojo; en consecuencia en el panel de control muestra el sitio ocupado, mediante la letra E (estacionado), y una disminución en el contador indicando el número de parqueos disponibles.

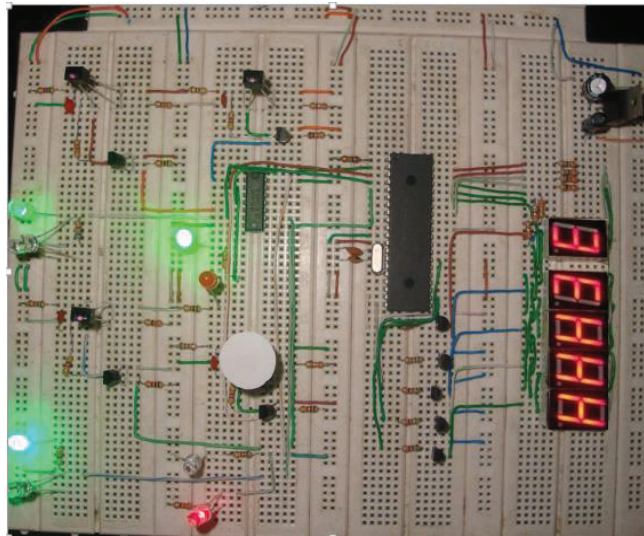


Figura 3.12. Primer sitio de parqueo estacionado

Contador a cero: todos los sitios se encuentran ocupados y activan los leds rojos; el panel de control muestra los displays con la letra E (estacionado) y el contador a marcando cero disponibilidad de sitios de parqueos.

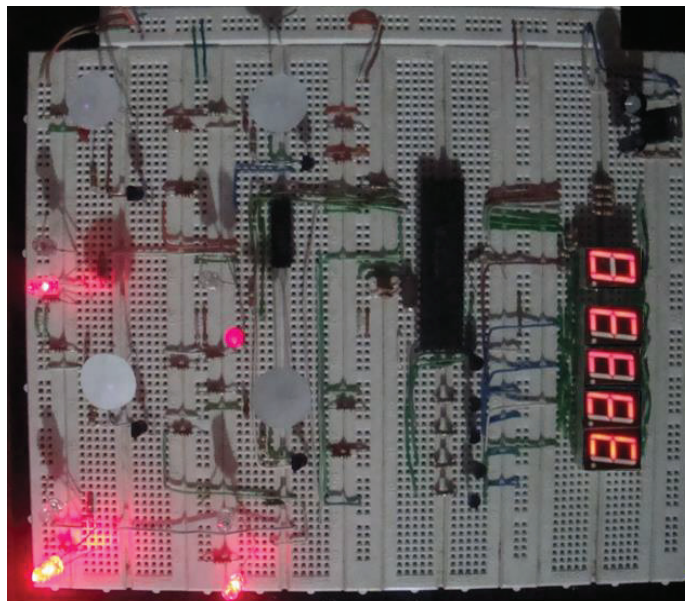


Figura 3.13. Sitios de parqueo estacionados, contador cero

3.1.2. DISEÑO DE LA ESTRUCTURA Y DISTRIBUCIÓN DEL PROTOTIPO

El diseño de la estructura y distribución del prototipo, se desarrolló en el programa Autocad para facilitar su visualización y mejor optimización de cada una de sus partes.

Las partes señaladas en el siguiente esquema constan de los 4 sitios de parqueo a 8 cm a desnivel con la entrada y/o salida, junto con el panel de displays y un contador UP/DOWN.

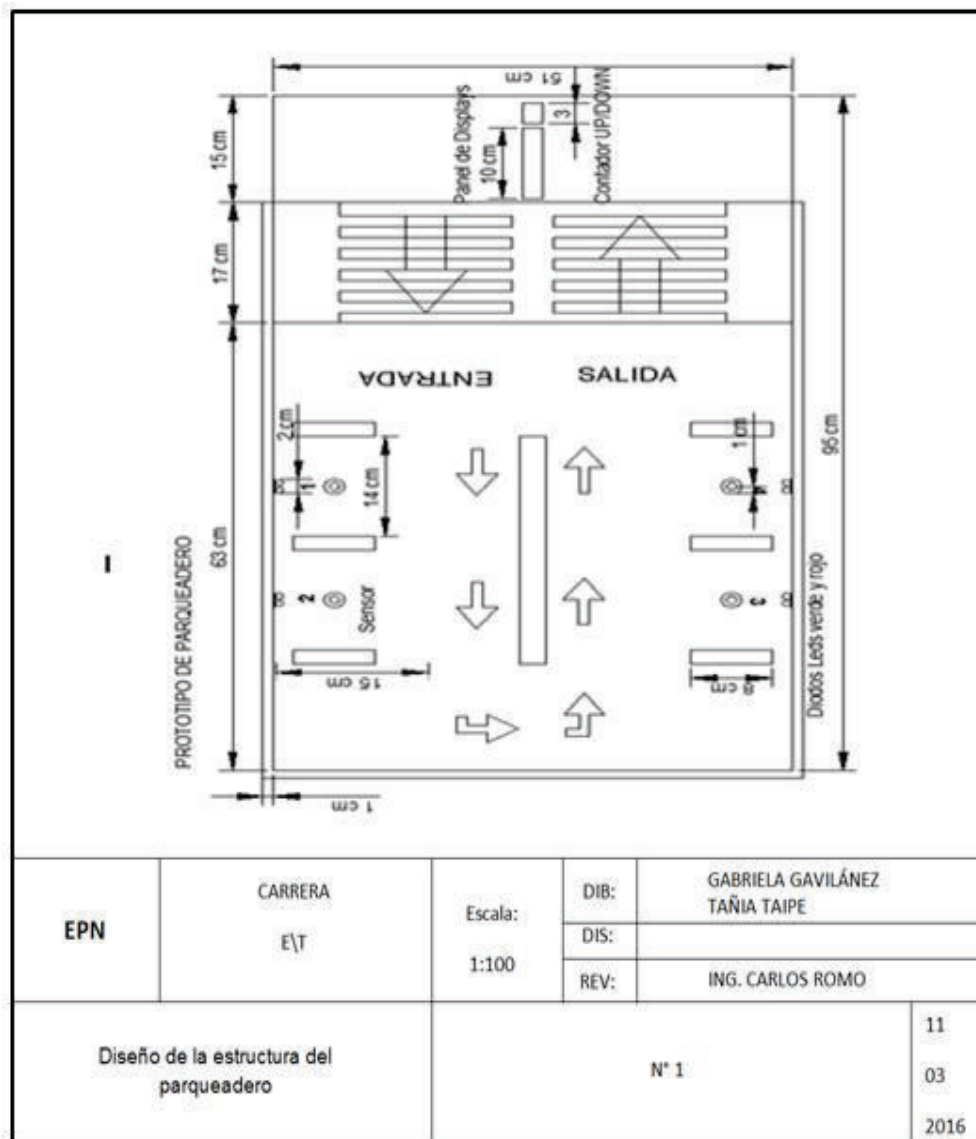


Figura 3.14. Diseño de la estructura de parqueadero

Como se puede apreciar en la figura 3.13, se ha designado las acotaciones respectivas a cada una de las dimensiones del parqueadero en su totalidad, definiendo medidas a escala (de 1 a 100) de espacios y elementos dentro de dicho prototipo. Cada sitio de parqueo cuenta con un área específica de 15 cm de largo por 14 cm de ancho. Cada uno de ellos contará con diodos leds de color verde y rojo incorporados en la pared frontal del sitio de parqueo, e indicarán si el sitio está habilitado (libre) o estacionado respectivamente.

Conjuntamente se implementará sensores infrarrojos en la parte inferior central (piso) de cada sitio de parqueo para detectar la presencia de los vehículos. Adicional a lo antes mencionado, se implementará señalizaciones en el lugar de estacionamiento que facilitarán el buen uso de los sitios de parqueos y entrada y/o salida del prototipo de estacionamiento.

Como paso final se procedió a la elaboración de la estructura del prototipo con:

- Madera, de 1 cm de espesor, para las paredes y piso.
- Papel contact para el revestimiento del piso y paredes. En el piso se llevará impreso las señalizaciones y sitios de parqueos.

El siguiente gráfico demuestra la estructura terminada

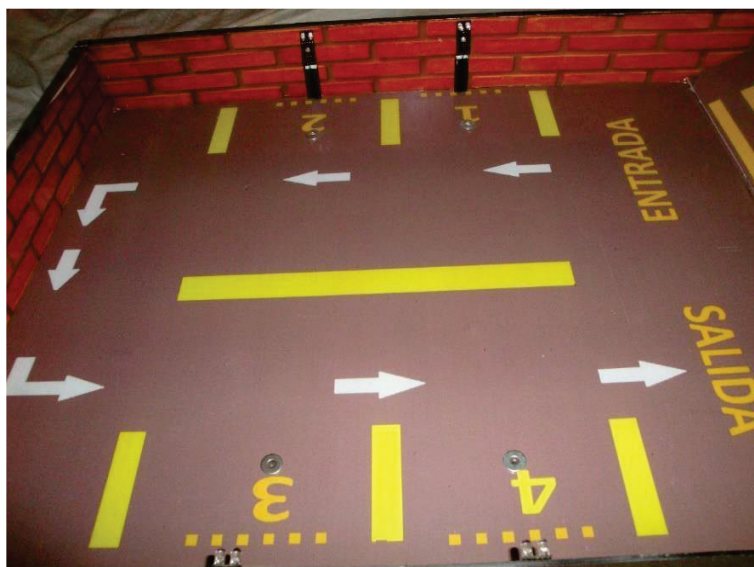


Figura 3.15. Estructura de prototipo de parqueadero

3.2 CONSTRUCCIÓN DE CIRCUITOS

La construcción de los circuitos se realizó de tal manera que su manejo y manipulación aporten adecuadamente las funcionalidades respectivas para cada una de las partes del prototipo del parqueadero. Su elaboración se desarrolló mediante los esquemas ya formulados en anteriores capítulos con programas como ARES, Proteus y, que fueron comprobados en el protoboard antes de ser implementados. ARES permite una elaboración y visualización óptimas de sus elementos en la placa electrónica, así como las conexiones entre ellos, mediante la generación automática de pistas y comprobar su adecuado posicionamiento automático de los elementos.

El siguiente gráfico muestra la elaboración de la placa electrónica con los elementos y sus pistas correspondientes.

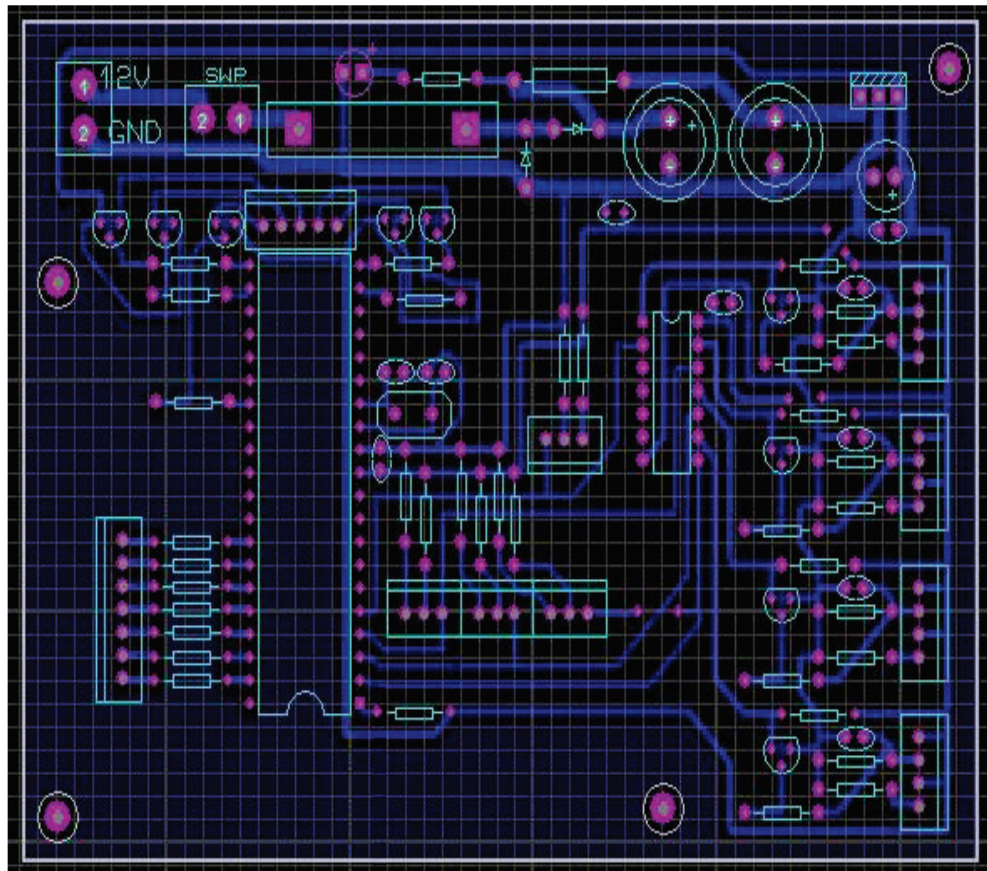


Figura 3.16. Placa principal del sistema de control de parqueos en ARES

Se puede apreciar la placa con el ruteo de las pistas del circuito en el anexo A. La placa principal se elaboró en baquelita de fibra de vidrio que brinda un mejor rendimiento de sus pistas y mayor resistencia de la placa; sus pistas impresas (conexiones entre elementos) fueron aplicadas en la baquelita mediante una lámina de termo transferencia, que sirve para colocar el diagrama del circuito desarrollado en ARES, y como paso final llevar a la baquelita a ser quemada con ácido. Al terminar de quemar la placa, se limpia con tiñer las pistas de tinta que quedaron y se procedió a comprobar que la placa no tenga corto circuitos.

Luego de comprobar la placa, se procedió a implementar los elementos adecuadamente; se dispuso la perforación de los agujeros a través de los que se soldaron en su mayoría los elementos, sócalos para displays, circuito integrado inverso y microcontrolador; junto con borneras para las conexiones.

El siguiente gráfico muestra la placa con la implementación dispuesta.

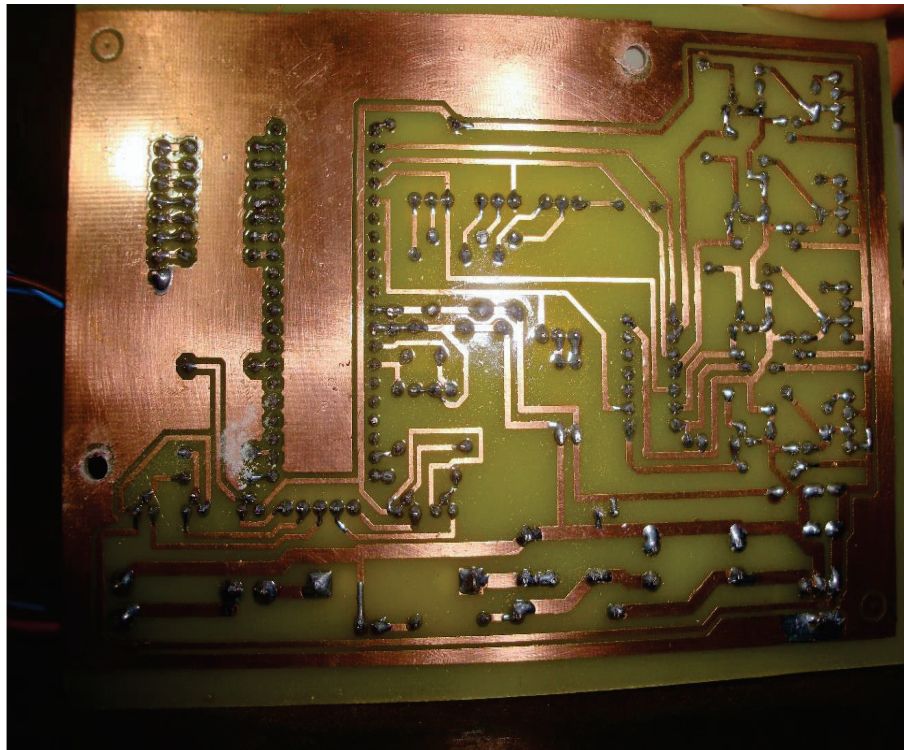


Figura 3.17. Placa Principal vista inferior

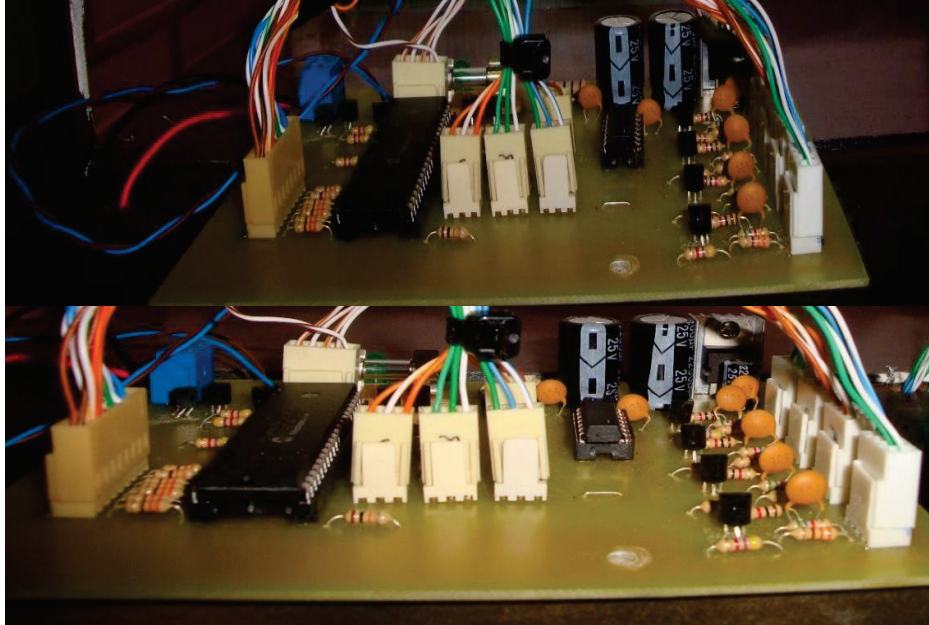


Figura 3.18. Placa principal vista superior

La placa del panel de control se realizó en baquelita perforada tipo galleta para una mejor manejo de los displays, debido a sus múltiples conexiones con la placa principal. El siguiente gráfico muestra su implementación.



Figura 3.19. Panel de Displays vista anterior y frontal

El resto de placas, para el contador y leds de visualización verde y rojo en sitios de parqueos, fueron elaborados en baquelita perforada tipo galleta tal como fue implementado el panel de displays, e incorporados en sus respectivos sitios de la estructura del prototipo de parqueadero. Sus diseños terminados se pueden apreciar en los anexo B.

El cableado subterráneo del prototipo se encontrará dispuesto para la conexión entre la placa principal y el resto de módulos. El cableado se lo puede apreciar en el anexo B.

Al determinar la ubicación de las placas e implementarlas se finalizó el prototipo completamente

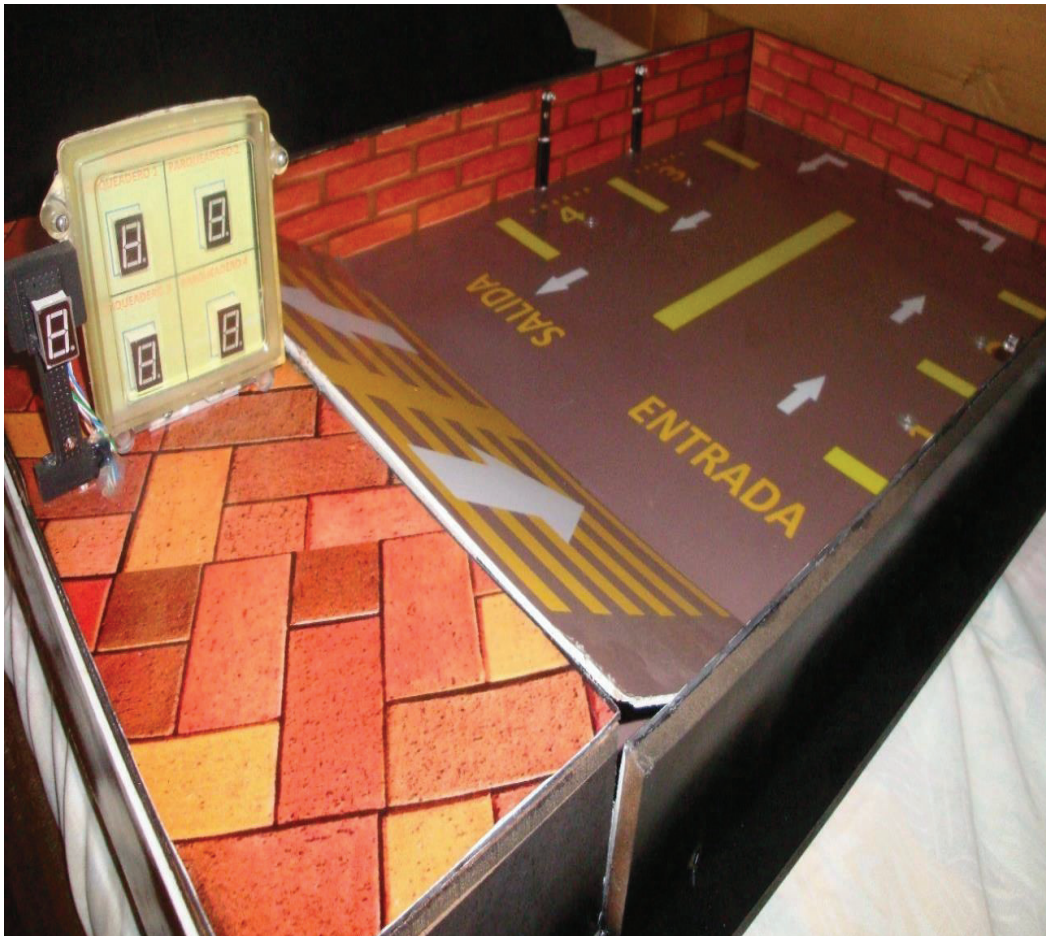


Figura 3.20. Prototipo implementado en su totalidad

3.3. PRUEBAS Y RESULTADOS

3.3.1. PRUEBAS

En la programación se apreció que la capacidad de compilación de MikroC soportaba un número específico de líneas, motivo por el cual la compilación tuvo fallas, y se disminuyó el número de parqueos.

Las polarizaciones de los elementos como transistores y sensores presentaron fallas en la funcionalidad del circuito mostrando una deficiencia; por lo cual se corrigió los valores numéricos de los elementos pasivos.

En la placa principal se verificó que sus elementos estén debidamente soldados y colocados acorde al diseño, así se comprobaría si los buses de datos entre la placa principal y las secundarias se encuentran en las condiciones adecuadas. Las pistas debían estar adecuadamente definidas para evitar una falta de conductividad.

Se verificó en la fuente de regulación añadida, que el voltaje entregado hacia el circuito sea el adecuado y establecido para que los elementos funcionen de forma correcta. El cableado se implementó según las necesidades de forma subterránea, y se verificó su conductividad desde los circuitos hacia sus dispositivos terminales como sensores, panel de displays, contador y leds.

La distancia a la que trabajan los sensores, luego de implementarlos en la estructura de la maqueta, se estableció con el propósito de corregir sus fallos y evitar que no detecte al automóvil estacionado. Su distancia estimada y verificada es de 1 cm aproximadamente.

Cada una de las partes en la placa y el resto de circuitos fueron debidamente ajustadas y protegidas, a fin de evitar cortocircuitos o interferencias con sus allegados.

3.3.2. RESULTADOS

El voltaje aplicado en el prototipo oscila entre los 4 y 12 V; dicho rango permite que los elementos en el prototipo del parqueadero funcionen, y se visualicen los resultados en los displays y los leds.

Al funcionar por primera vez el prototipo de parqueadero, se detectó un fallo en el contador debido a la falta de contacto entre el display y el sócalo de la placa. Se corrigió de inmediato y comprobó que funcionaba adecuadamente, registrando el valor numérico de los sitios habilitados, dentro del prototipo del parqueadero.

El cableado, responsable de transmitir la información hacia los dispositivos terminales, se encontraba en perfecto estado y se reajustó sus extremos con el propósito de que no ocurra errores en revisiones y consultas futuras.

Al comprobar cada uno de los sitios de parqueo empleando un vehículo, se pudo demostrar si en realidad los sensores funcionaban a la distancia establecida. Su funcionamiento era el adecuado y su registro hacia los leds, y displays fueron los más precisos posibles en tiempo real.

En conjunto, al ocupar todos los estacionamientos el contador registraba un cero, en el panel de displays todos los sitios de parqueo establecían una E de estacionado, y los leds rojos se encontraban encendidos. Al contrario si estaban todos los sitios habilitados el contador registraba la totalidad de sitios habilitados, en el panel de displays todos los sitios de parqueo establecían una H de habilitado, y los leds verdes se encontraban encendidos. Para mejor visualización de resultados dirigirse al anexo C.

CAPÍTULO IV:

CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

Los dispositivos necesarios para la implementación del control de parqueos como: sensores opto reflectivos o infrarrojos, microcontrolador, resistencias, displays, leds, transistores, fueron definidos según las necesidades de disminución de ruido, corriente, voltaje, polarizaciones de cada elemento del sistema de control de parqueos.

El microcontrolador PIC17F877A permite que las señales de los sensores infrarrojos sean procesadas, y su información se transmita para realizar el control del sistema de parqueos, apreciación de los leds, su conteo respectivo y la visualización en el panel de displays del parqueadero; así se logra una óptima comunicación en tiempo real de la transmisión de la información para los usuarios.

La programación del microcontrolador se realizó utilizando el lenguaje C para configurar en el compilador MikroC, porque es de fácil manejo de sus librerías y programación, Se desarrolló las funcionalidades del sistema de parqueos, usando adecuadamente cada una de las variables, operadores, estructuras de control entre otros.

Los sensores infrarrojos poseen un campo de visión limitado, para su adecuado funcionamiento, que asegura la efectividad de respuesta ante la presencia del vehículo, cubriendo la necesidad de información precisa y actualizada para los usuarios en la entrada del parqueadero.

Los displays y leds de alta intensidad visualizan la información emitida por los sensores de manera instantánea, razón por la cual su empleo es indispensable en situaciones de factibilidad de localización y empleo, continuo trabajo y voltajes admisibles del control de parqueos.

La simulación del circuito del sistema de control de parqueos se desarrolló en el simulador ISI Proteus, el cual ayuda a tener un previo análisis de la funcionalidad del circuito en la vida real.

La implementación del prototipo del sistema del control de parqueos se realizó mediante la elaboración de la placa, empleo de los materiales elegidos, incorporación de materiales a la placa e implementación de la estructura del prototipo como un parqueadero, determinando así ubicación de los sitios de parqueos óptimos, colocación de placas de circuitos entre otros.

4.2. RECOMENDACIONES

Se recomienda antes de realizar la programación, se verifique el tipo de lenguaje a utilizar, porque no todos los lenguajes disponen de las herramientas necesarias, para que sean validadas en el PIC utilizado.

La información acerca del funcionamiento de los displays, sensores y leds, debe ser respaldada mediante un adecuado conocimiento de sus protecciones y limitaciones, para evitar fallas o accidentes inminentes de una mala manipulación.

Se recomienda tener un amplio rango de opciones para los sensores infrarrojos, en su empleo ante diversas situaciones, tales como el campo de visión, su tiempo de respuesta ante un evento, protección necesaria, etc.

El cableado del prototipo debe ser etiquetado lo más preciso posible, para evitar equivocaciones al momento de establecer las conexiones entre los sensores y la placa principal.

La protección del circuito mediante el regulador de voltaje y la fuente de alimentación debe ser revisada continuamente antes de la conexión, para evitar sobre-voltajes, cortocircuitos, daños en los componentes del prototipo o fallas en sus respuestas.

GLOSARIO

- **I2C:** Es un bus de comunicaciones en serie usado en la industria, principalmente para comunicar microcontroladores y sus periféricos en sistemas integrados
- **Tecnología Nano watt:** Tecnología CMOS que consume energía cuando cambia su estado, lo que proporciona su estado de bajo consumo frente a otros tipos de tecnología de semiconductor.
- **AUSART:** Son periféricos de serie síncronos, utiliza 9 bits para distinguir entre los datos y dirección.
- **EUSART:** Es una interface serial de comunicación, puede adaptarse a múltiples periféricos o dispositivos que transfieren información de esta manera como una PC.
- **SPI:** Es una interfaz periférica serie que solo se diseñó para comunicaciones de corta distancia dentro de un mismo dispositivo.
- **MSSP:** proveedor de servicios de seguridad gestionada ○ **Reflectancia:** Capacidad de un cuerpo de reflejar la luz. ○ **IRLED:** Es un LED infrarrojo el cual emite una frecuencia no visible.
- **Material piroeléctrico:** Son partículas que contienen polarizaciones espontáneas producidas por dipolos orientados.
- **Stack:** es una lista ordenada o estructura de datos en la que el modo de acceso a sus elementos
- **Fototransistor:** Es igual a un transistor común, con la diferencia que el primero puede trabajar de 2 formas: como transistor normal con la corriente de base I_b (modo común) o como fototransistor, cuando la luz que incide en este elemento hace las veces de corriente de base. I_p (modo de iluminación).

BIBLIOGRAFÍA:

1. Microcontroladores, Obtenido de:
 © Monografias.com S.A. Miércoles, 24 de Octubre de 2007 a las 04:33
<http://www.monografias.com/trabajos12/microco/microco.shtml#ixzz3lnhijy Be4>
2. El mundo de los microcontroladores - Microcontroladores PIC – Programación en C con ejemplos, Obtenido de:
 Copyright © 1998 - 2015. MikroElektronika.
<http://www.mikroe.com/chapters/view/79/capitulo-1-el-mundo-de-losmicrocontroladores>.
3. Microcontroladores PIC, Obtenido de:
 Web oficial del fabricante de los PIC en inglés y chino.
<http://perso.wanadoo.es/pictob/micropic.htm>
4. Sensores infrarrojos, Obtenido de:
 Vía | Sensores infrarrojos. Joaquín Ferruz Melero.
<http://www.xatakaciencia.com/tecnologia/sensores-infrarrojos>
5. ¿Cómo es un led?, Obtenido de: © Tecnología Area
<http://www.areatecnologia.com/electronica/como-es-un-led.html>
6. Diodo LED, Obtenido de:
 © Monografias.com S.A. Miércoles, 29 de Mayo de 2013 a las 01:41
<http://www.monografias.com/trabajos60/diodo-led/diodoled2.shtml#ixzz3UWLI3sL>
7. Displays de 7 segmentos, Obtenido de: [Powered by Komento](#), 07 Mayo 2009
<http://tecnoface.com/tutoriales/12-practicando-con-displays-de-7segmentos/19-conceptos-basicos-de-un-display-de-7-segmentos>
8. LEDS Y DISPLAY DE 7 SEGMENTOS, Obtenido de:
 © Monografias.com S.A Jueves, 7 de Junio de 2012 a las 10:16.
<http://www.monografias.com/trabajos11/leds/leds2.shtml#ixzz3Q3KMexBH>
9. ¿Cómo funciona una resistencia?, Obtenido de:
 Escrito por Isaiah David | Traducido por Juan Manuel Rodríguez.
http://www.ehowenespanol.com/funciona-resistencia-como_166511/

10. Resistencias Tutorial de Electrónica, Obtenido de:

LinkedIn Corporation © 2015. Published on 06 de julio de 2012.

<http://es.slideshare.net/Jomicast/resistencias-13564623>

11. Tutorial capacitores, Obtenido de:

[Teoría de Circuitos I](#) - Última modificación: Mayo 28, 2002.

<http://www.ing.unp.edu.ar/electronica/asignaturas/ee016/tutoriales/capacitores/capacitores.htm>

12. Boninas y Capacitores, Obtenido de:

© Monografias.com S.A. 2 A.M. <http://www.monografias.com/trabajos100/boninas-y-capacitores/boninas-y-capacitores.shtml#ixzz3UiFFK4tp>

13. Reguladores de tensión, Obtenido de:

Reguladores de voltaje tema 11.

http://paginas.fisica.uson.mx/horacio.munguia/aula_virtual/Cursos/Instrumentacion%20II/Documentos/Regulacion%20voltaje.pdf

14. Programación de los microcontroladores - Microcontroladores PIC –

Programación en C con ejemplos, Obtenido de:

Copyright 1998-2015. MikroElektronika.

<http://www.mikroe.com/chapters/view/80/capitulo-2-programacion-de-losmicrocontroladores/>

15. Guía de referencia rápida a MIKROC, Obtenido de:

http://www.mikroe.com/downloads/get/945/es_c_syntax_v102.pdf

16. Data Sheet PIC 16F87XA, Obtenido de:

págs. 170. Consultado el 31 de mayo de 2009

<http://ww1.microchip.com/downloads/en/DeviceDoc/30569b.pdf>

17. Programación de los microcontroladores - Microcontroladores PIC –

Programación en C con ejemplos, Obtenido de:

Copyright © 1998 - 2014. Mikroelektronika.

<http://www.mikroe.com/chapters/view/80/capitulo-2-programacion-de-losmicrocontroladores/>

18. Sensor QRD1114, Obtenido de:

©CopyrightI+DElectrónica2011.

http://www.didacticaselectronicas.com/index.php?page=shop.product_details&flypage=flypage.tpl&product_id=656&category_id=134&option=com_virtuemart&Itemid=132&vmcchk=1&Itemid=132

19. Sensor Infrarrojo, Obtenido de:

Publicado por OptimusTronic en Copyright © 2000-2015 Looping Media, S.L.

<http://catorcepi.wikispaces.com/file/view/3678453-SENSOR->

[INFRARROJO-Teoria-y-practica.pdf](#)

20. MikroC PRO for PIC, Obtenido de:

Copyright © Targetware 2007-2013

<http://www.software.com.ar/mikroc-pro-for-pic.html>

21. LLOYD. Temes (2004), “Comunicación Electrónica”, México.

22. DORF. Svoboda (2006), “Circuitos Eléctricos”, México, Sexta edición.

23. BOYLESTAD Nashelsky (2003), “Teoría de circuitos y dispositivos

Electrónicos”, Pearson Prentice Hall. México.

24. REYES A. Carlos (2004), “Microcontroladores PIC”, Rispergraf, 2da edición.

25. ANGULO U. José María; ANGULO M. Ignacio; ROMERO Y. Susana

(2006), “Microcontroladores PIC, Diseño práctico de aplicaciones: segunda parte: PIC 16F87X”, McGraw-Hill, 2da edición.

26. Cucás Carlos, Mateus Jessica (Agosto, 2006), Construcción de un prototipo electrónico que informe sobre lugares vacantes a los usuarios de un parqueadero, QUITO/ EPN

<http://bibdigital.epn.edu.ec/handle/15000/1942>

27. Verdezoto, Geovanny (Mayo, 2014), Aplicación de la domótica para el control del sistema de temperatura, monitoreo, apertura y cierre automático de la puerta del garaje e iluminación interior y exterior de una vivienda.

<http://bibdigital.epn.edu.ec/handle/15000/7397>

ANEXOS

A. RUTEO DE LAS PISTAS DEL CIRCUITO

B. CONSTRUCCIÓN DE CIRCUITOS Y CABLEADO

C. PRUEBAS Y RESULTADOS

D. DATASHEETS

E. LIBRERÍAS EN MIKRO C

F. PROGRAMACIÓN COMPLETA

ANEXO A - RUTEO DE LAS PISTAS DEL CIRCUITO

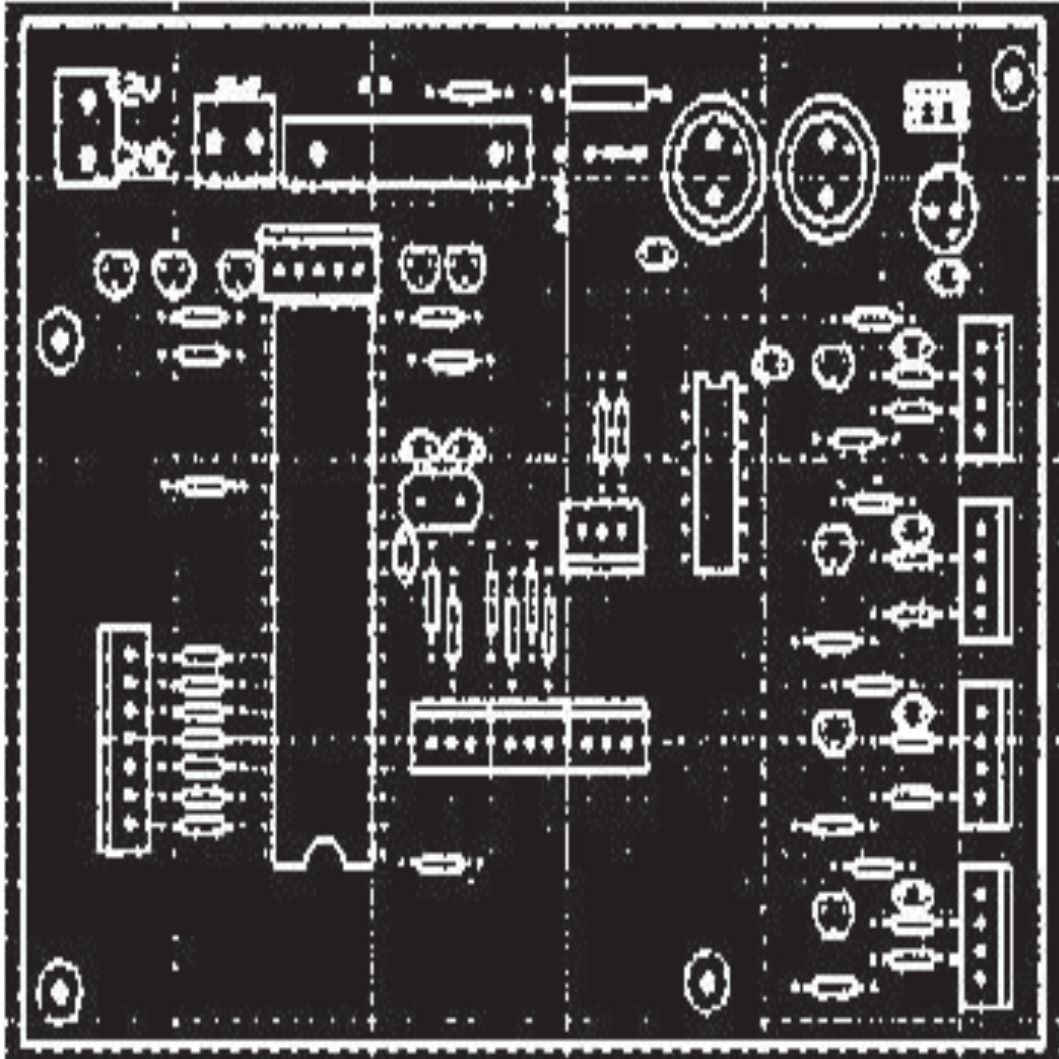


Figura 1. Ruteo de las pistas del circuito.

ANEXO B - CONSTRUCCIÓN DE CIRCUITOS Y CABLEADO

* Circuitos en vaquelita perforado tipo galleta para el contador y los leds (verde y rojo, en cada sitio de parqueo).

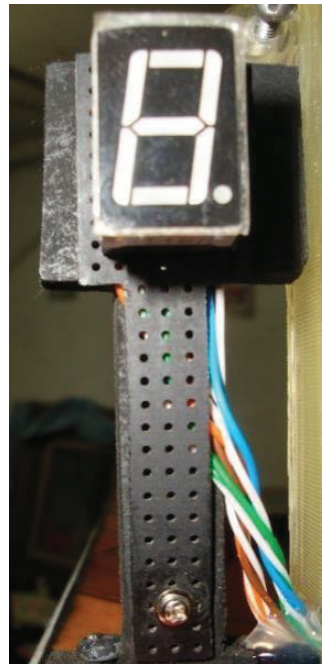


Figura 2. Display contador en baquelita perforada.



Figura 3. Leds, rojo y verde, en baquelita perforada.

* Cableado subterráneo

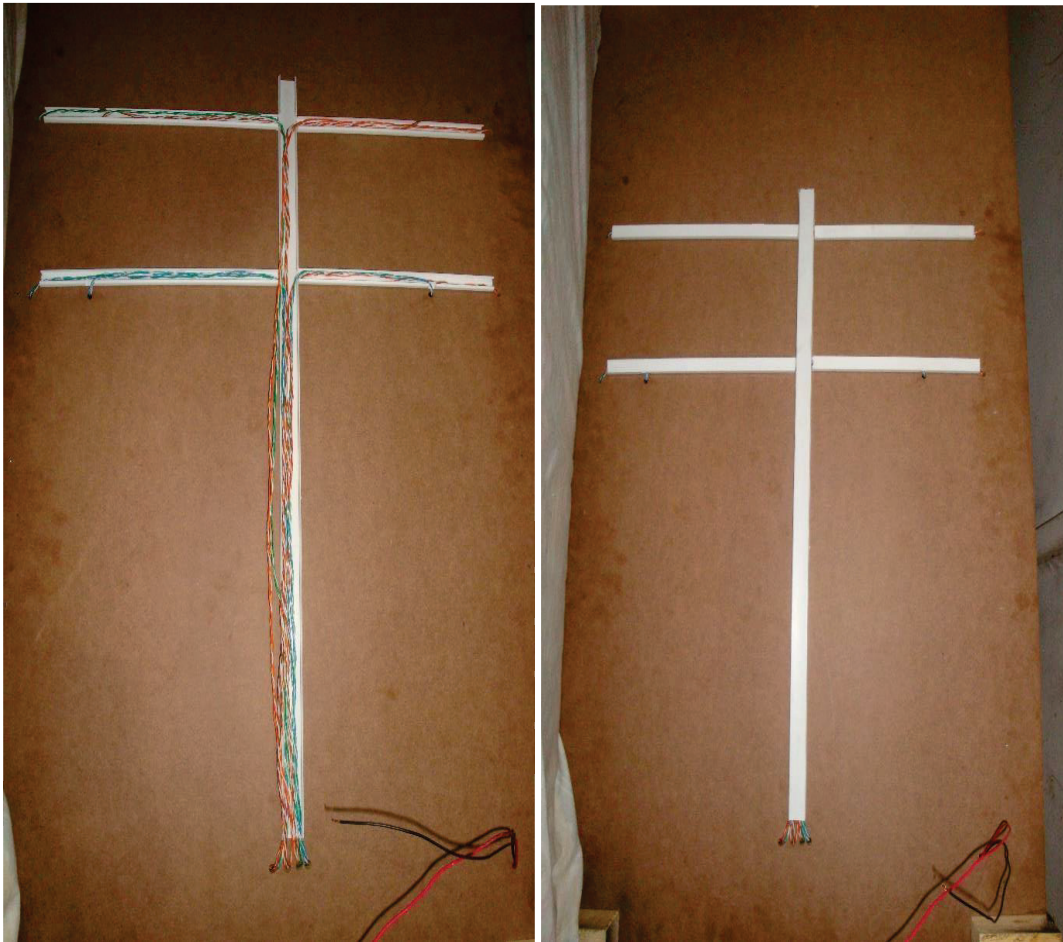


Figura 4. Vista interior y exterior del cableado en las canaletas.

ANEXO C - PRUEBAS Y RESULTADOS

Estacionamientos habilitados

Los leds verdes indican la disponibilidad de los sitios de parqueo y en el panel se visualiza las letras H de habilitado de cada sitio correspondiente. El contador indica 4 sitios habilitados.

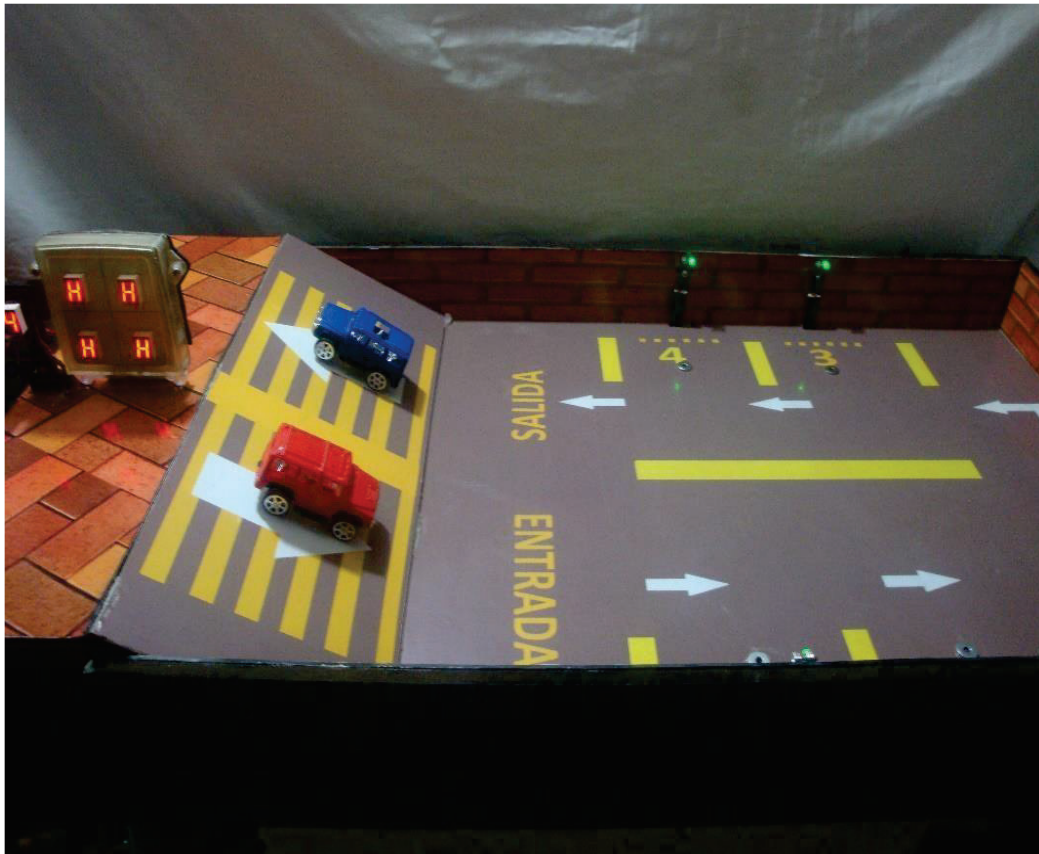


Figura 5. Estacionamientos habilitados.

Sitios de parqueo 2 y 3 estacionados

Los sitios de parqueo 2 y 3 se encuentran ocupados, se visualiza leds rojos en estos sitios, y el panel de display en la entrada indica la letra E de estacionado en los sitios correspondientes. El contador indica 2 sitios habilitados.



Figura 5. Sitios de parqueo 2 y 3 estacionados.

Sitios de parques estacionados

Los sitios de parqueo se encuentran totalmente ocupados, se visualiza leds rojos en todos los sitios, y el panel de display en la entrada indica la letra E de estacionado en todos los sitios. El contador indica 0 sitios habilitados.

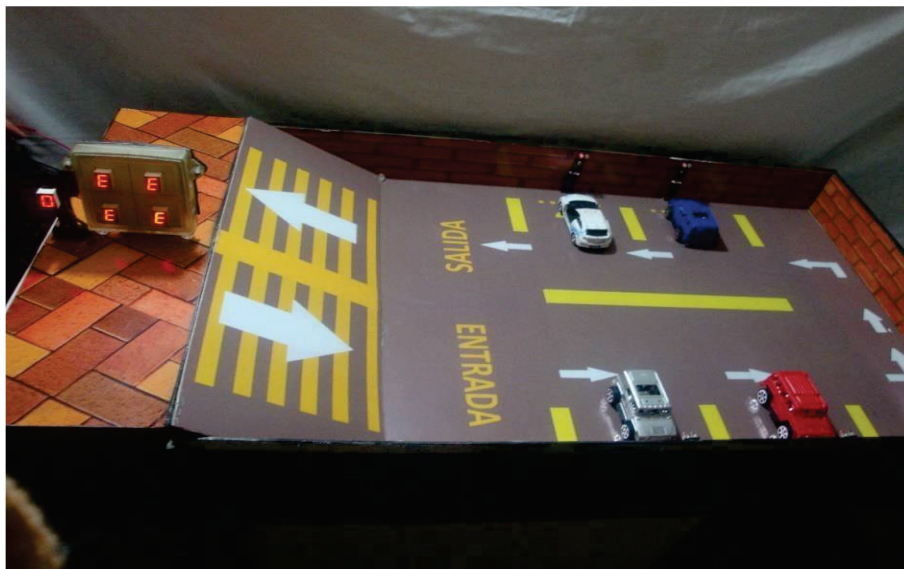


Figura 1. Sitios de parqueo estacionados.

ANEXO D – DATASHEETS

1.4.1 DATASHEET: PIC 16F877A

PIC16F87XA

1.0 DEVICE OVERVIEW

This document contains device specific information about the following devices:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

PIC16F873A/876A devices are available only in 28-pin packages, while PIC16F874A/877A devices are available in 40-pin and 44-pin packages. All devices in the PIC16F87XA family share common architecture with the following differences:

- The PIC16F873A and PIC16F874A have one-half of the total on-chip memory of the PIC16F876A and PIC16F877A
- The 28-pin devices have three I/O ports, while the 40/44-pin devices have five
- The 28-pin devices have fourteen interrupts, while the 40/44-pin devices have fifteen
- The 28-pin devices have five A/D input channels, while the 40/44-pin devices have eight
- The Parallel Slave Port is implemented only on the 40/44-pin devices

The available features are summarized in Table 1-1. Block diagrams of the PIC16F873A/876A and PIC16F874A/877A devices are provided in Figure 1-1 and Figure 1-2, respectively. The pinouts for these device families are listed in Table 1-2 and Table 1-3.

Additional information may be found in the PICmicro® Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

TABLE 1-1: PIC16F87XA DEVICE FEATURES

Key Features	PIC16F873A	PIC16F874A	PIC16F876A	PIC16F877A
Operating Frequency	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
Flash Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory (bytes)	128	128	256	256
Interrupts	14	15	14	15
I/O Ports	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C	Ports A, B, C, D, E
Timers	3	3	3	3
Capture/Compare/PWM modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Analog Comparators	2	2	2	2
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions
Packages	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN

1.4.2. DATASHEET: SENSOR QRD1114

Electrical / Optical CharacteristicsValues are at $T_A = 25^\circ\text{C}$ unless specified otherwise.

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Units
INPUT (Emitter)						
V_F	Forward Voltage	$I_F = 20\text{ mA}$			1.7	V
I_R	Reverse Leakage Current	$V_R = 5\text{ V}$			100	μA
λ_{PE}	Peak Emission Wavelength	$I_F = 20\text{ mA}$		940		nm
OUTPUT (Sensor)						
BV_{CEO}	Collector-Emitter Breakdown	$I_C = 1\text{ mA}$	30			V
BV_{ECO}	Emitter-Collector Breakdown	$I_E = 0.1\text{ mA}$	5			V
I_D	Dark Current	$V_{CE} = 10\text{ V}, I_F = 0\text{ mA}$			100	nA
COUPLED						
$I_{C(ON)}$	QRD1113 Collector Current	$I_F = 20\text{ mA}, V_{CE} = 5\text{ V},$ $D = 0.050\text{ inch}^{(5, 7)}$	0.300			mA
$I_{C(ON)}$	QRD1114 Collector Current	$I_F = 20\text{ mA}, V_{CE} = 5\text{ V},$ $D = 0.050\text{ inch}^{(5, 7)}$	1			mA
$V_{CE(BAT)}$	Collector Emitter Saturation Voltage	$I_F = 40\text{ mA}, I_C = 100\text{ }\mu\text{A},$ $D = 0.050\text{ inch}^{(5, 7)}$			0.4	V
I_{CX}	Cross Talk	$I_F = 20\text{ mA}, V_{CE} = 5\text{ V},$ $E_E = 0^{(6)}$		0.2	10.0	μA
t_r	Rise Time	$V_{CE} = 5\text{ V}, R_L = 100\text{ }\Omega,$ $I_{C(ON)} = 5\text{ mA}$		10		μs
t_f	Fall time	$I_{C(ON)} = 5\text{ mA}$		50		μs

Notes:

5. D is the distance from the sensor face to the reflective surface.
6. Crosstalk (I_{CX}) is the collector current measured with the indicated current on the input diode and with no reflective surface.
7. Measured using Eastman Kodak natural white test card with 90% diffused reflecting as a reflecting surface.

1.4.3. DATASHEET: INTEGRADO INVERSOR 74LS14P

SN54/74LS13 • SN54/74LS14

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
V_{T+}	Positive-Going Threshold Voltage	1.5		2.0	V	$V_{CC} = 5.0\text{ V}$
V_{T-}	Negative-Going Threshold Voltage	0.6		1.1	V	$V_{CC} = 5.0\text{ V}$
$V_{T+} - V_{T-}$	Hysteresis	0.4	0.8		V	$V_{CC} = 5.0\text{ V}$
V_{IK}	Input Clamp Diode Voltage		-0.65	-1.5	V	$V_{CC} = \text{MIN}, I_{IN} = -18\text{ mA}$
V_{OH}	Output HIGH Voltage	54	2.5	3.4	V	$V_{CC} = \text{MIN}, I_{OH} = -400\text{ }\mu\text{A}, V_{IN} = V_{IL}$
		74	2.7	3.4	V	
V_{OL}	Output LOW Voltage	54, 74	0.25	0.4	V	$V_{CC} = \text{MIN}, I_{OL} = 4.0\text{ mA}, V_{IN} = 2.0\text{ V}$
		74	0.35	0.5	V	$V_{CC} = \text{MIN}, I_{OL} = 8.0\text{ mA}, V_{IN} = 2.0\text{ V}$
I_{T+}	Input Current at Positive-Going Threshold		-0.14		mA	$V_{CC} = 5.0\text{ V}, V_{IN} = V_{T+}$
I_{T-}	Input Current at Negative-Going Threshold		-0.18		mA	$V_{CC} = 5.0\text{ V}, V_{IN} = V_{T-}$
I_{IH}	Input HIGH Current		1.0	20	μA	$V_{CC} = \text{MAX}, V_{IN} = 2.7\text{ V}$
				0.1	mA	$V_{CC} = \text{MAX}, V_{IN} = 7.0\text{ V}$
I_{IL}	Input LOW Current			-0.4	mA	$V_{CC} = \text{MAX}, V_{IN} = 0.4\text{ V}$
I_{OS}	Short Circuit Current (Note 1)	-20		-100	mA	$V_{CC} = \text{MAX}, V_{OUT} = 0\text{ V}$
I_{CC}	Power Supply Current					$V_{CC} = \text{MAX}$
	Total, Output HIGH	LS13	2.9	6.0	mA	
		LS14	8.6	16		
	Total, Output LOW	LS13	4.1	7.0	mA	
	LS14	12	21			

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS ($T_A = 25^\circ\text{C}$)

Symbol	Parameter	Max		Unit	Test Conditions
		LS13	LS14		
t_{PLH}	Propagation Delay, Input to Output	22	22	ns	$V_{CC} = 5.0\text{ V}$ $C_L = 15\text{ pF}$
t_{PHL}	Propagation Delay, Input to Output	27	22	ns	

1.4.4. DATASHEET: TRANSISTORES 2N3904 y 2N3906

1.4.4.2. TRANSISTORES 2N3904

2N3903, 2N3904

ELECTRICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$ unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit	
OFF CHARACTERISTICS					
Collector-Emitter Breakdown Voltage (Note 2.) ($I_C = 1.0\text{ mAdc}$, $I_B = 0$)	$V_{(BR)CEO}$	40	—	Vdc	
Collector-Base Breakdown Voltage ($I_C = 10\text{ }\mu\text{Adc}$, $I_E = 0$)	$V_{(BR)CBO}$	60	—	Vdc	
Emitter-Base Breakdown Voltage ($I_E = 10\text{ }\mu\text{Adc}$, $I_C = 0$)	$V_{(BR)EBO}$	6.0	—	Vdc	
Base Cutoff Current ($V_{CE} = 30\text{ Vdc}$, $V_{EB} = 3.0\text{ Vdc}$)	I_{BL}	—	50	nAdc	
Collector Cutoff Current ($V_{CE} = 30\text{ Vdc}$, $V_{EB} = 3.0\text{ Vdc}$)	I_{CEX}	—	50	nAdc	
ON CHARACTERISTICS					
DC Current Gain (Note 2.) ($I_C = 0.1\text{ mAdc}$, $V_{CE} = 1.0\text{ Vdc}$)	β_{FE}	20	—	—	
($I_C = 1.0\text{ mAdc}$, $V_{CE} = 1.0\text{ Vdc}$)		40	—		
($I_C = 10\text{ mAdc}$, $V_{CE} = 1.0\text{ Vdc}$)		35	—		
($I_C = 50\text{ mAdc}$, $V_{CE} = 1.0\text{ Vdc}$)		70	—		
($I_C = 100\text{ mAdc}$, $V_{CE} = 1.0\text{ Vdc}$)		50	150		
		100	300		
		30	—		
		60	—		
		15	—		
		30	—		
Collector-Emitter Saturation Voltage (Note 2.) ($I_C = 10\text{ mAdc}$, $I_B = 1.0\text{ mAdc}$) ($I_C = 50\text{ mAdc}$, $I_B = 5.0\text{ mAdc}$)	$V_{CE(sat)}$	—	0.2 0.3	Vdc	
Base-Emitter Saturation Voltage (Note 2.) ($I_C = 10\text{ mAdc}$, $I_B = 1.0\text{ mAdc}$) ($I_C = 50\text{ mAdc}$, $I_B = 5.0\text{ mAdc}$)	$V_{BE(sat)}$	0.65 —	0.85 0.95	Vdc	
SMALL-SIGNAL CHARACTERISTICS					
Current-Gain - Bandwidth Product ($I_C = 10\text{ mAdc}$, $V_{CE} = 20\text{ Vdc}$, $f = 100\text{ MHz}$)	f_T	250 300	—	MHz	
Output Capacitance ($V_{CB} = 5.0\text{ Vdc}$, $I_E = 0$, $f = 1.0\text{ MHz}$)	C_{ob0}	—	4.0	pF	
Input Capacitance ($V_{EB} = 0.5\text{ Vdc}$, $I_C = 0$, $f = 1.0\text{ MHz}$)	C_{ib0}	—	8.0	pF	
Input Impedance ($I_C = 1.0\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$)	h_{ie}	1.0 1.0	8.0 10	k Ω	
Voltage Feedback Ratio ($I_C = 1.0\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$)	η_{ve}	0.1 0.5	5.0 8.0	$\times 10^{-4}$	
Small-Signal Current Gain ($I_C = 1.0\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$)	h_{fe}	50 100	200 400	—	
Output Admittance ($I_C = 1.0\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$)	h_{oe}	1.0	40	μmhos	
Noise Figure ($I_C = 100\text{ }\mu\text{Adc}$, $V_{CE} = 5.0\text{ Vdc}$, $R_S = 1.0\text{ k}\Omega$, $f = 1.0\text{ kHz}$)	NF	—	6.0 5.0	dB	
SWITCHING CHARACTERISTICS					
Delay Time	$(V_{CC} = 3.0\text{ Vdc}$, $V_{BE} = 0.5\text{ Vdc}$, $I_C = 10\text{ mAdc}$, $I_{B1} = 1.0\text{ mAdc}$)	t_d	—	35	ns
Rise Time		t_r	—	35	ns
Storage Time	$(V_{CC} = 3.0\text{ Vdc}$, $I_C = 10\text{ mAdc}$, $I_{B1} = I_{B2} = 1.0\text{ mAdc}$)	t_s	—	175	ns
Fall Time		t_f	—	200	ns
				50	ns

2. Pulse Test: Pulse Width $\leq 300\text{ }\mu\text{s}$; Duty Cycle $\leq 2\%$.

1.4.4.3. TRANSISTORES 2N3906

2N3906

ELECTRICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$ unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit	
OFF CHARACTERISTICS					
Collector–Emitter Breakdown Voltage (Note 2.) ($I_C = 1.0\text{ mAdc}$, $I_B = 0$)	$V_{(BR)CEO}$	40	–	Vdc	
Collector–Base Breakdown Voltage ($I_C = 10\ \mu\text{Adc}$, $I_E = 0$)	$V_{(BR)CBO}$	40	–	Vdc	
Emitter–Base Breakdown Voltage ($I_E = 10\ \mu\text{Adc}$, $I_C = 0$)	$V_{(BR)EBO}$	5.0	–	Vdc	
Base Cutoff Current ($V_{CE} = 30\text{ Vdc}$, $V_{EB} = 3.0\text{ Vdc}$)	I_{BL}	–	50	nAdc	
Collector Cutoff Current ($V_{CE} = 30\text{ Vdc}$, $V_{EB} = 3.0\text{ Vdc}$)	I_{CEX}	–	50	nAdc	
ON CHARACTERISTICS (Note 2.)					
DC Current Gain ($I_C = 0.1\text{ mAdc}$, $V_{CE} = 1.0\text{ Vdc}$) ($I_C = 1.0\text{ mAdc}$, $V_{CE} = 1.0\text{ Vdc}$) ($I_C = 10\text{ mAdc}$, $V_{CE} = 1.0\text{ Vdc}$) ($I_C = 50\text{ mAdc}$, $V_{CE} = 1.0\text{ Vdc}$) ($I_C = 100\text{ mAdc}$, $V_{CE} = 1.0\text{ Vdc}$)	h_{FE}	60 80 100 60 30	– – 300 – –	–	
Collector–Emitter Saturation Voltage ($I_C = 10\text{ mAdc}$, $I_B = 1.0\text{ mAdc}$) ($I_C = 50\text{ mAdc}$, $I_B = 5.0\text{ mAdc}$)	$V_{CE(sat)}$	– –	0.25 0.4	Vdc	
Base–Emitter Saturation Voltage ($I_C = 10\text{ mAdc}$, $I_B = 1.0\text{ mAdc}$) ($I_C = 50\text{ mAdc}$, $I_B = 5.0\text{ mAdc}$)	$V_{BE(sat)}$	0.65 –	0.85 0.95	Vdc	
SMALL–SIGNAL CHARACTERISTICS					
Current–Gain – Bandwidth Product ($I_C = 10\text{ mAdc}$, $V_{CE} = 20\text{ Vdc}$, $f = 100\text{ MHz}$)	f_T	250	–	MHz	
Output Capacitance ($V_{CB} = 5.0\text{ Vdc}$, $I_E = 0$, $f = 1.0\text{ MHz}$)	C_{obo}	–	4.5	pF	
Input Capacitance ($V_{EB} = 0.5\text{ Vdc}$, $I_C = 0$, $f = 1.0\text{ MHz}$)	C_{ibo}	–	10	pF	
Input Impedance ($I_C = 1.0\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$)	h_{ie}	2.0	12	k Ω	
Voltage Feedback Ratio ($I_C = 1.0\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$)	f_{re}	0.1	10	$\times 10^{-4}$	
Small–Signal Current Gain ($I_C = 1.0\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$)	h_{fe}	100	400	–	
Output Admittance ($I_C = 1.0\text{ mAdc}$, $V_{CE} = 10\text{ Vdc}$, $f = 1.0\text{ kHz}$)	h_{oe}	3.0	60	μmhos	
Noise Figure ($I_C = 100\ \mu\text{Adc}$, $V_{CE} = 5.0\text{ Vdc}$, $R_S = 1.0\text{ k}\Omega$, $f = 1.0\text{ kHz}$)	NF	–	4.0	dB	
SWITCHING CHARACTERISTICS					
Delay Time	$(V_{CC} = 3.0\text{ Vdc}$, $V_{BE} = 0.5\text{ Vdc}$, $I_C = 10\text{ mAdc}$, $I_{B1} = 1.0\text{ mAdc}$)	t_d	–	35	ns
Rise Time		t_r	–	35	ns
Storage Time	$(V_{CC} = 3.0\text{ Vdc}$, $I_C = 10\text{ mAdc}$, $I_{B1} = I_{B2} = 1.0\text{ mAdc}$)	t_s	–	225	ns
Fall Time	$(V_{CC} = 3.0\text{ Vdc}$, $I_C = 10\text{ mAdc}$, $I_{B1} = I_{B2} = 1.0\text{ mAdc}$)	t_f	–	75	ns

2. Pulse Test: Pulse Width $\leq 300\ \mu\text{s}$; Duty Cycle $\leq 2\%$.

1.4.5. DATASHEET: REGULADOR DE VOLTAJE L7805


LM7805C, LM7812C, LM7815C
www.ti.com

SNOSBR7D – MAY 2000 – REVISED APRIL 2013



These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

ABSOLUTE MAXIMUM RATINGS⁽¹⁾⁽²⁾

Input Voltage ($V_O = 5V, 12V$ and $15V$)		35V
Internal Power Dissipation ⁽³⁾		Internally Limited
Operating Temperature Range (T_A)		0°C to +70°C
Maximum Junction Temperature	(TO-3 Package)	150°C
	(NDE Package)	150°C
Storage Temperature Range		-65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	TO-3 Package	300°C
	TO-220 Package NDE	230°C

- (1) Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. For ensured specifications and the test conditions, see Electrical Characteristics.
- (2) If Military/Aerospace specified devices are required, please contact the Texas Instruments Sales Office/Distributors for availability and specifications.
- (3) Thermal resistance of the TO-3 package is typically 4°C/W junction to case and 35°C/W case to ambient. Thermal resistance of the TO-220 package (NDE) is typically 4°C/W junction to case and 50°C/W case to ambient.

ELECTRICAL CHARACTERISTICS LM78XXC⁽¹⁾

0°C ≤ T_J ≤ 125°C unless otherwise noted.

Output Voltage			5V			12V			15V			Units	
Input Voltage (unless otherwise noted)			10V			19V			23V				
Symbol	Parameter	Conditions	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
V_O	Output Voltage	$T_J = 25^\circ\text{C}, 5\text{ mA} \leq I_O \leq 1\text{ A}$	4.8	5	5.2	11.5	12	12.5	14.4	15	15.6	V	
		$P_D \leq 15\text{ W}, 5\text{ mA} \leq I_O \leq 1\text{ A}$	4.75		5.25	11.4		12.6	14.2		15.7	V	
ΔV_O	Line Regulation	$I_O = 500\text{ mA}$	$T_J = 25^\circ\text{C}$	3	50	4	120	4	150			mV	
			ΔV_{IN}	(7 ≤ V_{IN} ≤ 25)	14.5 ≤ V_{IN} ≤ 30)								
			0°C ≤ T_J ≤ +125°C	50	120	150							
		$I_O \leq 1\text{ A}$	ΔV_{IN}	(8 ≤ V_{IN} ≤ 20)	(15 ≤ V_{IN} ≤ 27)	(18.5 ≤ V_{IN} ≤ 30)							
			$T_J = 25^\circ\text{C}$	50	120	150							
			ΔV_{IN}	(7.5 ≤ V_{IN} ≤ 20)	(14.6 ≤ V_{IN} ≤ 27)	(17.7 ≤ V_{IN} ≤ 30)							
ΔV_O	Load Regulation	$T_J = 25^\circ\text{C}$	$5\text{ mA} \leq I_O \leq 1.5\text{ A}$		10	50		12	120		12	150	mV
			$250\text{ mA} \leq I_O \leq 750\text{ mA}$			25			60			75	mV
			$5\text{ mA} \leq I_O \leq 1\text{ A}, 0^\circ\text{C} \leq T_J \leq +125^\circ\text{C}$			50			120			150	mV
I_Q	Quiescent Current	$I_O \leq 1\text{ A}$	$T_J = 25^\circ\text{C}$			8			8			8	mA
			0°C ≤ T_J ≤ +125°C			8.5			8.5			8.5	mA
ΔI_Q	Quiescent Current Change	$5\text{ mA} \leq I_O \leq 1\text{ A}$	$T_J = 25^\circ\text{C}, I_O \leq 1\text{ A}$			0.5			0.5			0.5	mA
			$V_{MIN} \leq V_{IN} \leq V_{MAX}$			1.0			1.0			1.0	mA
			$I_O \leq 500\text{ mA}, 0^\circ\text{C} \leq T_J \leq +125^\circ\text{C}$			1.0			1.0			1.0	mA
			ΔV_{IN}	(7 ≤ V_{IN} ≤ 25)	(14.5 ≤ V_{IN} ≤ 30)	(17.5 ≤ V_{IN} ≤ 30)							

- (1) All characteristics are measured with capacitor across the input of 0.22 μF, and a capacitor across the output of 0.1 μF. All characteristics except noise voltage and ripple rejection ratio are measured using pulse techniques ($t_w \leq 10\text{ ms}$, duty cycle ≤ 5%). Output voltage changes due to changes in internal temperature must be taken into account separately.

ANEXO E - LIBRERÍAS DE MIKRO C

Las librerías representan un código compilado, escrito en MikroC, que contiene un conjunto de variables y constantes. Existen librerías: ANSI C estándar, misceláneas y para el hardware.

Librería ANSI C estándar

LIBRERÍA	DESCRIPCIÓN
ANSI C Ctype Library	Utilizada principalmente para probar o para convertir los datos
ANSI C Math Library	Utilizada para las operaciones matemáticas de punto flotante
ANSI C Stdlib Library	Contiene las funciones de librerías estándar
ANSI C String Library	Utilizada para realizar las operaciones de cadenas y de manipulaciones de memoria

Librerías Misceláneas

LIBRERÍA	DESCRIPCIÓN
Button Library	Utilizada para desarrollar los proyectos
Conversion Library	Utilizada para la conversión de tipos de datos
Sprint Library	Utilizada para formatear los datos con facilidad
PrintOut Library	Utilizada para formatear los datos e imprimirlos
Time Library	Utilizada para cálculos de tiempo (formato UNIX time)
Trigonometry Library	Utilizada para la implementación de funciones trigonométricas fundamentales
Setjmp Library	Utilizada para los saltos de programa

Librerías para el Hardware

LIBRERÍA	DESCRIPCIÓN
ADC Library	Utilizada para el funcionamiento del convertidor A/D
CAN Library	Utilizada para las operaciones con el módulo CAN
CANSPI Library	Utilizada para las operaciones con el módulo CAN externo (MCP2515 o MCP2510)
Compact Flash Library	Utilizada para las operaciones con las tarjetas de memoria Compact Flash
EEPROM Library	Utilizada para las operaciones con la memoria EEPROM incorporada
EthernetPIC18Fxx3 60 Library	Utilizada para las operaciones con el módulo Ethernet incorporado
Flash Memory Library	Utilizada para las operaciones con la memoria Flash incorporada
Graphic Lcd Library	Utilizada para las operaciones con el módulo LCD gráfico con resolución 128x64
I2C Library	Utilizada para las operaciones con el módulo de comunicación serial I2C incorporado
Keypad Library	Utilizada para las operaciones con el teclado (botones de presión 4x4)
Lcd Library	Utilizada para las operaciones con el LCD (de 2x16 caracteres)
Manchester Code Library	Utilizada para la comunicación utilizando el código Manchester
Multi Media Card Library	Utilizada para las operaciones con las tarjetas multimedia MMC flash
One Wire Library	Utilizada para las operaciones con los circuitos utilizando la comunicación serial One Wire

Port Expander Library	Utilizada para las operaciones con el extensor de puertos MCP23S17
PS/2 Library	Utilizada para las operaciones con el teclado estándar
	PS/2
PWM Library	Utilizada para las operaciones con el módulo PWM incorporado
RS-485 Library	Utilizada para las operaciones con los módulos utilizando la comunicación serial RS485
Software I2C Library	Utilizada para simular la comunicación I2C con software
Software SPI Library	Utilizada para simular la comunicación SPI con software
Software UART Library	Utilizada para simular la comunicación UART con software
Sound Library	Utilizada para generar la señal de audio
SPI Library	Utilizada para las operaciones con el módulo SPI incorporado
SPI Ethernet Library	Utilizada para la comunicación SPI con el módulo ETHERNET (ENC28J60)
SPI Graphic Lcd Library	Utilizada para la comunicación SPI de 4 bits con el LCD gráfico
SPI LCD Library	Utilizada para la comunicación SPI de 4 bits con el LCD (de 2x16 caracteres)
SPI Lcd8 Library	Utilizada para la comunicación SPI de 8 bits con el LCD
SPI T6963C Graphic Lcd Library	Utilizada para la comunicación SPI con el LCD gráfico
UART Library	Utilizada para las operaciones con el módulo UART incorporado
USB Hid Library	Utilizada para las operaciones con el módulo USB incorporado

ANEXO F - PROGRAMACION COMPLETA

PROGRAMACIÓN FINAL: 4 PARQUEADEROS:

```
#define ENTRADA_E1 PORTA.F0
#define ENTRADA_E2 PORTA.F1
#define ENTRADA_E3 PORTA.F2
#define ENTRADA_E4 PORTA.F3
//#define sens_int PORTA.F4 //#define sens_out PORTA.F4 const unsigned short
DIGITOS[] =
{
0x3F, //Código del dígito 0
0x06, //Código del dígito 1
0x5B, //Código del dígito 2
0x4F, //Código del dígito 3
0x66, //Código del dígito 4
0x6D, //Código del dígito 5
0x7D, //Código del dígito 6
0x07, //Código del dígito 7
0x79, //Código del dígito 8 codigo E estacionado
0x76, //Código del dígito 9 codigo H habilitado
};
tiempo(int A)
{
delay_us(3000); //3 ms
}
rutina(int B)
{
```

```

}
void main()
{
    unsigned short A;    //short, signed o int es lo mismo
    //short b;
    //short e;
    short total;
    char j; bit
    flag_1; bit
    flag_2; bit
    flag_3; bit
    button;
    ADCON1=0b00000111;//CONVIERTO PUERTO A EN DIGITAL
    TRISA=0B11111111; // PUERTO A COMO ENTRADA
    TRISB=0b00000000;// puerto B como salida portb=0b00000000;// puerto esta 0L
    TRISC=0b0;// puerto B como salida portc=0b00000000;
    TRISD=0b0;
    PORTD=0b11111111;
    //PWM1_Init(5000); //5000 Hz
    //PWM1_Start(); // start PWM
    j = 1; while(1)
    {
        if((ENTRADA_E1==0)  &&  (ENTRADA_E2==0)&&(ENTRADA_E3==0)  &&
        (ENTRADA_E4==0))
            //1
    TOTAL=0 ; //PARQUEADERO VACIO
    PORTB = ~DIGITOS[9]; //Visualiza las unidades.
    PORTd.F1=0; //Activa en alto el primer display tiempo(A); //Retado de 10m
    segundos

```

```

PORTd.f1=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las decenas.
PORTd.F2=0; //Activa en alto el primer display tiempo(A); //Retado de 10m
segundos
PORTd.f2=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las decenas.
PORTd.F3=0; //Activa en alto el segundo display tiempo(A); //Retado de 10m
segundos
PORTd.f3=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las centenas.
PORTd.F4=0; //Activa en alto el tercer display tiempo(A); //Retado de 10m
segundos
PORTd.f4=1; //Desactiva todos los displays.
}
if((ENTRADA_E1==1) && (ENTRADA_E2==0)&&(ENTRADA_E3==0) &&
(ENTRADA_E4==0))
{
//2
TOTAL=1 ;
PORTB = ~DIGITOS[8]; //Visualiza las unidades.
PORTd.F1=0; //Activa en alto el primer display tiempo(A); //Retado de 10m
segundos
PORTd.f1=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las decenas.
PORTd.F2=0; //Activa en alto el primer display tiempo(A); //Retado de 10m
segundos
PORTd.f2=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las decenas.
PORTd.F3=0; //Activa en alto el segundo display tiempo(A); //Retado de 10m
segundos PORTd.f3=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las centenas.
PORTd.F4=0; //Activa en alto el tercer display delay_ms(5); //Retado de 10m
segundos

```

```

PORTd.f4=1; //Desactiva todos los displays.
}
if((ENTRADA_E1==0) && (ENTRADA_E2==1)&&(ENTRADA_E3==0) &&
(ENTRADA_E4==0))
{//3
TOTAL=1 ;
PORTB = ~DIGITOS[9]; //Visualiza las unidades.
PORTd.F1=0; //Activa en alto el primer display tiempo(A); //Retado de 10m
segundos
PORTd.f1=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[8]; //Visualiza las decenas.
PORTd.F2=0; //Activa en alto el primer display tiempo(A); //Retado de 10m
segundos
PORTd.f2=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las decenas.
PORTd.F3=0; //Activa en alto el segundo display tiempo(A); //Retado de 10m
segundos
PORTd.f3=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las centenas.
PORTd.F4=0; //Activa en alto el tercer display delay_ms(5); //Retado de 10m
segundos
PORTd.f4=1; //Desactiva todos los displays.
}
if((ENTRADA_E1==1) && (ENTRADA_E2==1)&&(ENTRADA_E3==0) &&
(ENTRADA_E4==0))
{//4
TOTAL=2 ;
PORTB = ~DIGITOS[8]; //Visualiza las unidades.
PORTd.F1=0; //Activa en alto el primer display tiempo(A); //Retado de 10m
segundos

```

```

PORTd.f1=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[8]; //Visualiza las decenas.
PORTd.F2=0; //Activa en alto el primer display tiempo(A); //Retado de 10m
segundos
PORTd.f2=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las decenas.
PORTd.F3=0; //Activa en alto el segundo display tiempo(A); //Retado de 10m
segundos
PORTd.f3=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las centenas.
PORTd.F4=0; //Activa en alto el tercer display delay_ms(5); //Retado de 10m
segundos
PORTd.f4=1; //Desactiva todos los displays.
}
if((ENTRADA_E1==0) && (ENTRADA_E2==0)&&(ENTRADA_E3==1) &&
(ENTRADA_E4==0))
    { //5
TOTAL=1 ;
PORTB = ~DIGITOS[9]; //Visualiza las unidades.
PORTd.F1=0; //Activa en alto el primer display
tiempo(A); //Retado de 10m segundos
PORTd.f1=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las decenas.
PORTd.F2=0; //Activa en alto el primer display tiempo(A); //Retado de 10m
segundos
PORTd.f2=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[8]; //Visualiza las decenas.
PORTd.F3=0; //Activa en alto el segundo display tiempo(A); //Retado de 10m
segundos
PORTd.f3=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las centenas.

```

```

PORTd.F4=0; //Activa en alto el tercer display   delay_ms(5); //Retado de 10m
segundos
PORTd.f4=1; //Desactiva todos los displays.
}

if((ENTRADA_E1==1)   &&   (ENTRADA_E2==0)&&(ENTRADA_E3==1)   &&
(ENTRADA_E4==0))

  {//6

    TOTAL=2 ;

    PORTB = ~DIGITOS[8]; //Visualiza las unidades.
    PORTd.F1=0; //Activa en alto el primer display   tiempo(A); //Retado de 10m
segundos
    PORTd.f1=1; //Desactiva todos los displays.
    PORTB = ~DIGITOS[9]; //Visualiza las decenas.
    PORTd.F2=0; //Activa en alto el primer display   tiempo(A); //Retado de 10m
segundos
    PORTd.f2=1; //Desactiva todos los displays.

    PORTB = ~DIGITOS[8]; //Visualiza las decenas.
    PORTd.F3=0; //Activa en alto el segundo display   tiempo(A); //Retado de 10m
segundos
    PORTd.f3=1; //Desactiva todos los displays.
    PORTB = ~DIGITOS[9]; //Visualiza las centenas.
    PORTd.F4=0; //Activa en alto el tercer display   delay_ms(5); //Retado de 10m
segundos
    PORTd.f4=1; //Desactiva todos los displays.

  }

if((ENTRADA_E1==0)   &&   (ENTRADA_E2==1)&&(ENTRADA_E3==1)   &&
(ENTRADA_E4==0))

  {//7

    TOTAL=2 ;

    PORTB = ~DIGITOS[9]; //Visualiza las unidades.
    PORTd.F1=0; //Activa en alto el primer display   tiempo(A); //Retado de 10m

```

```

segundos  PORTd.f1=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[8]; //Visualiza las decenas.
PORTd.F2=0; //Activa en alto el primer display  tiempo(A); //Retado de 10m
segundos
PORTd.f2=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[8]; //Visualiza las decenas.
PORTd.F3=0; //Activa en alto el segundo display  tiempo(A); //Retado de 10m
segundos
PORTd.f3=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las centenas.
PORTd.F4=0; //Activa en alto el tercer display  delay_ms(5); //Retado de 10m
segundos
PORTd.f4=1; //Desactiva todos los displays.
}

if((ENTRADA_E1==1) && (ENTRADA_E2==1)&&(ENTRADA_E3==1) &&
(ENTRADA_E4==0))
{//8  1=8 Y 0=9
TOTAL=3 ;

PORTB = ~DIGITOS[8]; //Visualiza las unidades.
PORTd.F1=0; //Activa en alto el primer display  tiempo(A); //Retado de 10m
segundos
PORTd.f1=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[8]; //Visualiza las decenas.
PORTd.F2=0; //Activa en alto el primer display  tiempo(A); //Retado de 10m
segundos
PORTd.f2=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[8]; //Visualiza las decenas.
PORTd.F3=0; //Activa en alto el segundo display tiempo(A); //Retado de 10m
segundos
PORTd.f3=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las centenas.

```



```

    PORTd.F4=0; //Activa en alto el tercer display    delay_ms(5); //Retado de 10m
    segundos
    PORTd.f4=1; //Desactiva todos los displays.

}

if((ENTRADA_E1==0)    &&    (ENTRADA_E2==0)&&(ENTRADA_E3==0)    &&
(ENTRADA_E4==1))

    {//9    1=8 Y 0=9

TOTAL=1 ;

    PORTB = ~DIGITOS[9]; //Visualiza las unidades.
    PORTd.F1=0; //Activa en alto el primer display    tiempo(A); //Retado de 10m
    segundos
    PORTd.f1=1; //Desactiva todos los displays.
    PORTB = ~DIGITOS[9]; //Visualiza las decenas.
    PORTd.F2=0; //Activa en alto el primer display    tiempo(A); //Retado de 10m
    segundos    PORTd.f2=1; //Desactiva todos los displays.
    PORTB = ~DIGITOS[9]; //Visualiza las decenas.
    PORTd.F3=0; //Activa en alto el segundo display    tiempo(A); //Retado de 10m
    segundos    PORTd.f3=1; //Desactiva todos los displays.
    PORTB = ~DIGITOS[8]; //Visualiza las centenas.
    PORTd.F4=0; //Activa en alto el tercer display    delay_ms(5); //Retado de 10m
    segundos
    PORTd.f4=1; //Desactiva todos los displays.

}

if((ENTRADA_E1==1)    &&    (ENTRADA_E2==0)&&(ENTRADA_E3==0)    &&
(ENTRADA_E4==1))

    {//10    1=8 Y 0=9

TOTAL=2 ;

    PORTB = ~DIGITOS[8]; //Visualiza las unidades.
    PORTd.F1=0; //Activa en alto el primer display    tiempo(A); //Retado de 10m
    segundos PORTd.f1=1; //Desactiva todos los displays.
    PORTB = ~DIGITOS[9]; //Visualiza las decenas.

```

```

PORTd.F2=0; //Activa en alto el primer display tiempo(A); //Retado de 10m
segundos PORTd.f2=1; //Desactiva todos los displays.

PORTB = ~DIGITOS[9]; //Visualiza las decenas.

PORTd.F3=0; //Activa en alto el segundo display tiempo(A); //Retado de 10m
segundos PORTd.f3=1; //Desactiva todos los displays.

PORTB = ~DIGITOS[8]; //Visualiza las centenas.

PORTd.F4=0; //Activa en alto el tercer display delay_ms(5); //Retado de 10m
segundos
PORTd.f4=1; //Desactiva todos los displays.

}

if((ENTRADA_E1==0) && (ENTRADA_E2==1)&&(ENTRADA_E3==0) &&
(ENTRADA_E4==1))

    { //11 1=8 Y 0=9

        TOTAL=2 ;

PORTB = ~DIGITOS[9]; //Visualiza las unidades.
PORTd.F1=0; //Activa en alto el primer display tiempo(A); //Retado de 10m
segundos PORTd.f1=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[8]; //Visualiza las decenas.
PORTd.F2=0; //Activa en alto el primer display tiempo(A); //Retado de 10m
segundos
PORTd.f2=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las decenas.
PORTd.F3=0; //Activa en alto el segundo display tiempo(A); //Retado de 10m
segundos
PORTd.f3=1; //Desactiva todos los displays.

PORTB = ~DIGITOS[8]; //Visualiza las centenas.
PORTd.F4=0; //Activa en alto el tercer display delay_ms(5); //Retado de 10m
segundos
PORTd.f4=1; //Desactiva todos los displays.

}

if((ENTRADA_E1==1) && (ENTRADA_E2==1)&&(ENTRADA_E3==0) &&

```

```

(ENTRADA_E4==1))
{//12  1=8 Y 0=9
TOTAL=3 ;
PORTB = ~DIGITOS[8]; //Visualiza las unidades.
PORTd.F1=0; //Activa en alto el primer display
tiempo(A); //Retado de 10m segundos
PORTd.f1=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[8]; //Visualiza las decenas.
PORTd.F2=0; //Activa en alto el primer display tiempo(A); //Retado de 10m
segundos
PORTd.f2=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las decenas.
PORTd.F3=0; //Activa en alto el segundo display tiempo(A); //Retado de 10m
segundos
PORTd.f3=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[8]; //Visualiza las centenas.
PORTd.F4=0; //Activa en alto el tercer display delay_ms(5); //Retado de 10m
segundos
PORTd.f4=1; //Desactiva todos los displays.
}
if((ENTRADA_E1==0) && (ENTRADA_E2==0)&&(ENTRADA_E3==1) &&
(ENTRADA_E4==1))
{//13  1=8 Y 0=9
TOTAL=2 ;
PORTB = ~DIGITOS[9]; //Visualiza las unidades.
PORTd.F1=0; //Activa en alto el primer display tiempo(A); //Retado de 10m
segundos
PORTd.f1=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[9]; //Visualiza las decenas.

```

```

PORTd.F2=0; //Activa en alto el primer display tiempo(A); //Retado de 10m
segundos
PORTd.f2=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[8]; //Visualiza las decenas.
PORTd.F3=0; //Activa en alto el segundo display tiempo(A); //Retado de 10m
segundos
PORTd.f3=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[8]; //Visualiza las centenas.
PORTd.F4=0; //Activa en alto el tercer display delay_ms(5); //Retado de 10m
segundos
PORTd.f4=1; //Desactiva todos los displays.
}
if((ENTRADA_E1==1) && (ENTRADA_E2==0)&&(ENTRADA_E3==1) &&
(ENTRADA_E4==1))
    { //14 1=8 Y 0=9
    TOTAL=3 ;
    PORTB = ~DIGITOS[8]; //Visualiza las unidades.
    PORTd.F1=0; //Activa en alto el primer display tiempo(A); //Retado de 10m
segundos
    PORTd.f1=1; //Desactiva todos los displays.
    PORTB = ~DIGITOS[9]; //Visualiza las decenas.
    PORTd.F2=0; //Activa en alto el primer display tiempo(A); //Retado de 10m
segundos
    PORTd.f2=1; //Desactiva todos los displays.
    PORTB = ~DIGITOS[8]; //Visualiza las decenas.
    PORTd.F3=0; //Activa en alto el segundo display tiempo(A); //Retado de 10m
segundos
    PORTd.f3=1; //Desactiva todos los displays.
    PORTB = ~DIGITOS[8]; //Visualiza las centenas.
    PORTd.F4=0; //Activa en alto el tercer display delay_ms(5); //Retado de 10m
segundos
    PORTd.f4=1; //Desactiva todos los displays.

```

```

}

if((ENTRADA_E1==0)  &&  (ENTRADA_E2==1)&&(ENTRADA_E3==1)  &&
(ENTRADA_E4==1))

  {//15  1=8 Y 0=9

    TOTAL=3 ;

    PORTB = ~DIGITOS[9]; //Visualiza las unidades.
    PORTd.F1=0; //Activa en alto el primer display  tiempo(A); //Retado de 10m
segundos
    PORTd.f1=1; //Desactiva todos los displays.
    PORTB = ~DIGITOS[8]; //Visualiza las decenas.
    PORTd.F2=0; //Activa en alto el primer display  tiempo(A); //Retado de 10m
segundos
    PORTd.f2=1; //Desactiva todos los displays.

    PORTB = ~DIGITOS[8]; //Visualiza las decenas.
    PORTd.F3=0; //Activa en alto el segundo display tiempo(A); //Retado de 10m
segundos
    PORTd.f3=1; //Desactiva todos los displays.
    PORTB = ~DIGITOS[8]; //Visualiza las centenas.
    PORTd.F4=0; //Activa en alto el tercer display  delay_ms(5); //Retado de 10m
segundos
    PORTd.f4=1; //Desactiva todos los displays.

  }

if((ENTRADA_E1==1)  &&  (ENTRADA_E2==1)&&(ENTRADA_E3==1)  &&
(ENTRADA_E4==1))

  {//16  1=8 Y 0=9

    TOTAL=4 ;

    PORTB = ~DIGITOS[8]; //Visualiza las unidades.
    PORTd.F1=0; //Activa en alto el primer display  tiempo(A); //Retado de 10m
segundos
    PORTd.f1=1; //Desactiva todos los displays.
    PORTB = ~DIGITOS[8]; //Visualiza las decenas.

```

```

PORTd.F2=0; //Activa en alto el primer display tiempo(A); //Retado de 10m
segundos
PORTd.f2=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[8]; //Visualiza las decenas.
PORTd.F3=0; //Activa en alto el segundo display tiempo(A); //Retado de 10m
segundos
PORTd.f3=1; //Desactiva todos los displays.
PORTB = ~DIGITOS[8]; //Visualiza las centenas.
PORTd.F4=0; //Activa en alto el tercer display delay_ms(5); //Retado de 10m
segundos
PORTd.f4=1; //Desactiva todos los displays.
}
////////////////////////////////////// if(total==0)
{
PORTB = ~DIGITOS[4];
PORTd.F0=0; //Activa en alto el cuarto display tiempo(A); //Retado de 10m
segundos
PORTd.f0=1; //Desactiva todos los displays.
}
if(total==1)
{
PORTB = ~DIGITOS[3];
PORTd.F0=0; //Activa en alto el cuarto display tiempo(A); //Retado de 10m
segundos
PORTd.f0=1; //Desactiva todos los displays.
} if(total=2)
{
PORTB = ~DIGITOS[2];
PORTd.F0=0; //Activa en alto el cuarto display tiempo(A); //Retado de 10m
segundos
PORTd.f0=1; //Desactiva todos los displays.
}

```

```
    }    if(total==3)
    {
        PORTB = ~DIGITOS[1];

        PORTd.F0=0; //Activa en alto el cuarto display    tiempo(A); //Retado de 10m
segundos
        PORTd.f0=1; //Desactiva todos los displays.
    }    if(total==4)
    {
        PORTB = ~DIGITOS[0];

        PORTd.F0=0; //Activa en alto el cuarto display
]
0    tiempo(A); //Retado de 10m segundos    PORTd.f0=1; //Desactiva todos los
displays.
    }
}
}
```