

UNIVERSIDAD DEL VALLE DE GUATEMALA  
FACULTAD DE INGENIERÍA

SISTEMA ADMINISTRADOR DE CASILLEROS CON  
ACCESO POR MEDIO DE CÓDIGO DE BARRAS Y  
MECANISMO DE COBRO AUTOMÁTICO

Juancarlos Jr. Burmester Castillo

Guatemala  
2005



**SISTEMA ADMINISTRADOR DE CASILLEROS CON  
ACCESO POR MEDIO DE CÓDIGO DE BARRAS Y  
MECANISMO DE COBRO AUTOMÁTICO**

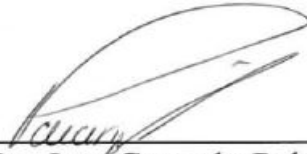
UNIVERSIDAD DEL VALLE DE GUATEMALA  
FACULTAD DE INGENIERÍA

SISTEMA ADMINISTRADOR DE CASILLEROS CON  
ACCESO POR MEDIO DE CÓDIGO DE BARRAS Y  
MECANISMO DE COBRO AUTOMÁTICO

Trabajo de investigación para optar al grado académico de  
Licenciado en Ingeniería Electrónica

presentado por Juancarlos Jr. Burmester Castillo


Guatemala  
2005



---

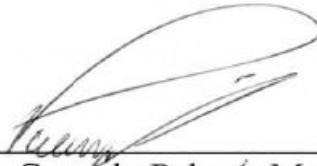
Vo.Bo. Ing. Gonzalo Palaréa Murga  
Asesor de Tesis

**TRIBUNAL EXAMINADOR**



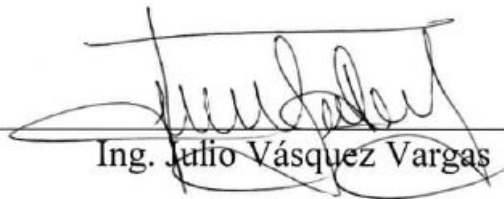
---

Dr. Ing. Manuel López V.



---

Ing. Gonzalo Palaréa Murga



---

Ing. Julio Vásquez Vargas

Fecha de aprobación: 27 de mayo de 2005

## PREFACIO

---

---

Una parte importante de la vida termina hoy; una nueva inicia.

Quiero agradecer a Dios que me permite llegar a la conclusión de una etapa determinante en mi vida...

...quien a través de mis padres me ha enseñado valores morales, de rectitud y honestidad que me han de servir para desenvolverme ahora profesionalmente.

...quien a través de mis maestros me ha enseñado herramientas de conocimiento que ahora me permiten enfrentar el mundo laboral, el mundo real.

...quien a través de mis amigos me ha apoyado y motivado a dar lo mejor.

# ÍNDICE

---

---

Prefacio.....	v
Lista de tablas.....	vii
Lista de figuras.....	viii
Resumen.....	ix
I. Introducción.....	1
II. Objetivos	
A. Generales.....	3
B. Específicos.....	3
III. Desarrollo	
A. Descripción y planteamiento del problema.....	5
B. Análisis descriptivo de la solución planteada.....	7
1. Circuito maestro.....	8
<i>a. Despliegue de mensajes visuales.....</i>	8
<i>b. Reloj externo.....</i>	10
<i>c. Lectura de barras.....</i>	10
<i>d. Reproducción de mensajes audibles.....</i>	11
<i>e. Memoria externa.....</i>	12
<i>f. Recepción de monedas.....</i>	13
<i>g. Comunicación con circuitos esclavos.....</i>	13
2. Circuito esclavo.....	14
<i>a. Indicadores visuales.....</i>	15
<i>b. Estado de las puertas.....</i>	16
<i>c. Activación de solenoides.....</i>	17
<i>d. Comunicación con circuito maestro.....</i>	17
C. Distribución, mapeo y programación.....	18
1. Circuito maestro.....	18
2. Circuito esclavo.....	22
IV. Resultados.....	25
V. Conclusiones y recomendaciones.....	33
VI. Bibliografía.....	35
VII. Anexo A. Mapas de direcciones y memoria.....	37
VIII. Anexo B. Códigos fuente en lenguaje ensamblador.....	43
IX. Anexo C. Manual de operación.....	81

## LISTA DE TABLAS

---

---

Tabla 1.	Distribución de puertos A y B en circuito maestro.....	18
Tabla 2.	Distribución de puertos C, D y E en circuito maestro.....	18
Tabla 3.	Distribución de puertos A y B en circuito esclavo.....	22
Tabla 4.	Distribución de puerto C en circuito esclavo.....	22
Tabla 5.	Equivalencia de direcciones I2C en esclavos.....	23
Tabla 6.	Mapa de memoria RAM en microcontrolador maestro.....	39
Tabla 7.	Mapa de direcciones del bus en circuito maestro.....	40
Tabla 8.	Mapa de memoria EEPROM externa.....	41
Tabla 9.	Mapa de memoria RAM en microcontrolador esclavo.....	42

## LISTA DE FIGURAS

---

---

Figura 1.	Casilleros en la Biblioteca UVG.....	5
Figura 2.	Casilleros en el corredor del edificio C, UVG.....	6
Figura 3.	Distribución del control.....	7
Figura 4.	Funciones del circuito maestro.....	8
Figura 5.	Matriz de LEDs.....	8
Figura 6.	Lector de barras (MS140 de Unitech).....	11
Figura 7.	Identificador de monedas (MS16 de MEI).....	13
Figura 8.	Funciones del circuito esclavo.....	14
Figura 9.	Esquemático del barrido matricial de LEDs.....	15
Figura 10.	Esquemático del barrido matricial de interruptores.....	16
Figura 11.	Diagrama de flujo, rutina principal del microcontrolador maestro.....	20
Figura 12.	Diagrama de flujo, rutina de interrupción del microcontrolador maestro.....	21
Figura 13.	Diagrama de flujo, rutina principal del microcontrolador esclavo.....	24
Figura 14.	Diagrama de flujo, rutina de interrupción del microcontrolador esclavo.....	24
Figura 15.	Selección de columnas de despliegue en el circuito maestro.....	26
Figura 16.	Sección central del circuito maestro.....	27
Figura 17.	Despliegue en matrices de diodos emisores de luz en el circuito maestro.....	28
Figura 18.	Fotografía del circuito maestro.....	29
Figura 19.	Circuito esclavo.....	30
Figura 20.	Fotografía del circuito esclavo.....	31

## RESUMEN

---

---

Los casilleros son espacios de almacenamiento temporal y su administración consiste en el proceso de asignarlos a un usuario específico y un tiempo después permitirle, únicamente a éste, acceso para retirar las pertenencias que en el casillero haya depositado. Esta tarea se realiza automáticamente por el Sistema Administrador de Casilleros, la solución electrónica aquí presentada. El circuito permite a los usuarios utilizar su identificación personal como “llave” para accederlos y realiza la tarea de cobro dependiendo del tiempo de uso de los mismos.

La concepción modular planteada hace flexible su implementación. Se recomienda para áreas en las que la demanda del recurso es alta, liberando así al personal de las tareas relacionadas con la asignación de los casilleros.

# I. INTRODUCCIÓN

---

---

La utilización de mecanismos automatizados se ha venido convirtiendo en elemento indispensable para la eficiencia, rapidez y comodidad en los diferentes aspectos de la vida cotidiana contemporánea y se observa su crecimiento exponencial día a día.

Técnicas tales como la lectura de identificadores codificados han estado presentes en diversas funciones de control automático; en nuestro medio, desde hace más de una década en las cajas receptoras de cadenas de supermercados y de forma más discreta en controles industriales y comerciales diversos. De la misma manera, se pueden mencionar los mecanismos de cobro automatizado cuya presencia es aún más antigua por medio de los teléfonos monederos, parquímetros y máquinas expendedoras de productos varios, por ejemplo.

Hasta hace pocos años el uso de dichas herramientas implicaba altos costos, lo que hacía rentable su uso únicamente en equipos manufacturados en serie y a gran escala. El avance tecnológico que permite el acceso fácil y a bajo costo de los microcontroladores hoy en día, hace viable proponer alternativas de su uso en soluciones prácticas puntuales que hasta hace pocos años eran consideradas inejecutables. A pesar de ello, existe en el medio una subutilización de estas herramientas de automatización y por ende, de las ilimitadas posibilidades que su uso combinado presenta.

Desde 2001 los estudiantes de la Universidad del Valle de Guatemala cuentan con un carné que incluye un segmento de barras que codifica su número y por medio del cual es posible identificar de manera plena a su propietario. Este elemento constituye un paso tecnológicamente avanzado que fue concebido, sin lugar a dudas, para facilitar la identificación del estudiante y su utilización en soluciones electrónicas prácticas. Hasta el momento, sin embargo, no se han implementado las aplicaciones para sacarle debido provecho.

Por lo anterior, es posible identificar una amplia gama de usos que harían más eficiente, tanto para la administración universitaria como para el estudiante, el control y el acceso a los diversos recursos disponibles en el campus.

Este trabajo de graduación concretiza un ejemplo del uso práctico de este tipo de mecanismos automáticos, mediante la creación de un dispositivo administrador de los casilleros de los que dispone el estudiantado del campus central de la UVG, haciéndolos accesibles por medio de código de barras presente en los carnés e incorporando, cuando corresponda, el cobro por el servicio prestado en proporción al tiempo durante el cual éste fue utilizado.

## II. OBJETIVOS

---

---

### A. GENERALES

1. Administrar eficientemente recursos limitados por parte de usuarios temporales.
2. Identificar al usuario del servicio por medio de la utilización de código de barras.
3. Incorporar el cobro monedero directo y automático, según cuotas preestablecidas, por la utilización de un servicio a cargo de un usuario específico.

### B. ESPECÍFICOS

1. Diseñar la capa física de un circuito capaz de administrar casilleros para depósito de pertenencias personales.
2. Diseñar la capa física de un circuito capaz de relacionarse con el administrador para operar las chapas de los casilleros.
3. Programar microcontroladores de la familia PIC18FXX8 que coordinen las funciones de las capas físicas descritas en los puntos anteriores.
4. Aprovechar la decisión de incorporar la identificación por medio de código de barras en el carné de la UVG.
5. Demostrar la viabilidad de la administración automática de casilleros y la potencialidad de este tipo de administración en otros servicios.



### III. DESARROLLO

---

---

#### A. DESCRIPCIÓN Y PLANTEAMIENTO DEL PROBLEMA

El almacenaje temporal de pertenencias es un aspecto con el que se enfrenta el humano a diario. Ya sea por restricciones de los lugares que se atienden o por simple comodidad, es necesario contar con un lugar seguro donde poder guardar por un corto período de tiempo las pertenencias personales. El concepto consiste en permitir a un usuario determinado disponer de un espacio por un tiempo y luego, permitirle únicamente a este usuario acceder a él para retirar los objetos que haya depositado.

La Biblioteca en la Universidad del Valle es un ejemplo sencillo de un lugar que por sus características restringe el acceso de maletines, mochilas y bolsos a sus confines y ofrece al estudiantado casilleros para su respectivo depósito.

Actualmente con sólo plantear la necesidad de un candado, el personal de biblioteca lo entrega gratuitamente a cambio del propio carné del estudiante. Este proceso, sin embargo, implica una carga a las tareas que los bibliotecarios ya tienen a su cargo y requiere de mucho tiempo que, algunos estudiantes, prefieren no invertir.

En los casos en los que el requerimiento de guardar pertenencias no ocurre de manera constante, como en gimnasios o cafeterías por ejemplo, el problema se traduce en una negativa, por parte del usuario, a cancelar una cuota, quizás mensual, por el derecho de

Figura 1. Casilleros en la Biblioteca UVG



usar un servicio que no considera plenamente devengado, es decir, el período mínimo de arrendamiento es muy grande. Es posible, por otro lado, que mientras un usuario está necesitado de un espacio de almacenamiento, existan varios desocupados pero reservados para aquellos que si aceptaron adquirirlos por el mes o semestre.

Figura 2. Casilleros del corredor del edificio C, UVG



Por estas razones se considera que al reducir el tiempo mínimo de arrendamiento se produzca una ocupación más dinámica de los casilleros y, con el objeto de agilizar el proceso de asignación, se introduce la noción de un dispositivo electrónico capaz de administrar arreglos de casilleros como el que se muestra en la fotografía.

La “llave” utilizada para abrir los casilleros puede ser cualquier documento de identificación personal. Es necesario, sin embargo, que éste pueda ser reconocido electrónicamente, como el código de barras ya incluido en el carné de la Universidad del Valle, o en membresías de gimnasios o clubes.

Actualmente los establecimientos pueden quizás tener únicamente unos cuantos casilleros arrendados por largos períodos de tiempo pudiendo tener muchos arrendados por períodos más cortos. Con la implementación de la solución propuesta, el cobro por el servicio se sigue haciendo, pero con tarifas mucho más bajas que permitan a los usuarios preferir la comodidad y seguridad de almacenar sus pertenencias en lugar de llevarlas consigo.

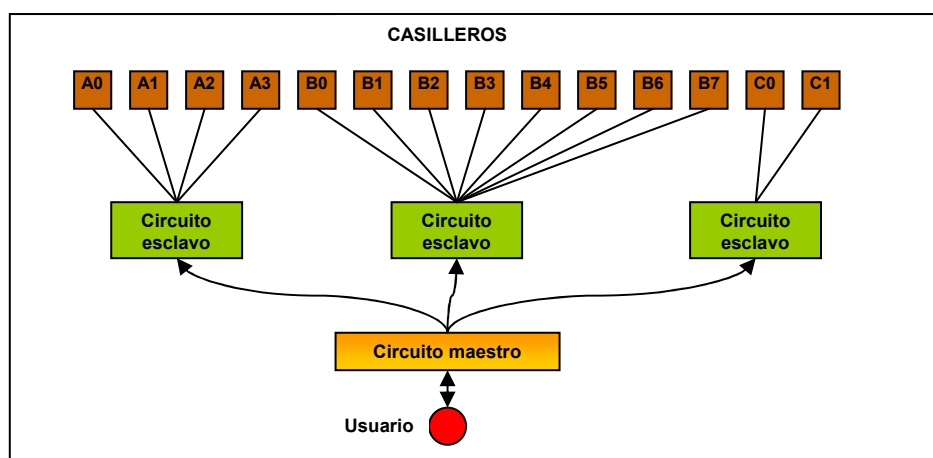
## B. ANÁLISIS DESCRIPTIVO DE LA SOLUCIÓN PLANTEADA

El administrador electrónico de casilleros que se plantea como solución al problema del manejo de los espacios de almacenamiento temporal, consiste en un módulo central llamado maestro y uno o varios módulos secundarios, llamados esclavos.

El circuito maestro es el encargado de establecer el contacto con el usuario, llevar el registro de ocupación, cobrar y ordenar la apertura de cada casillero. Por otro lado, el objetivo de el o los secundarios es llevar a cabo las tareas de operación sobre los propios casilleros. Estas incluyen operar los solenoides presentes en las cerraduras, señalar la vacancia de un espacio usando luces y reportar debidamente el estado de abierto/cerrado de cada una de las puertas de los casilleros que tiene a su cargo.

La concepción modular de la solución, mostrada en la figura 3, permite una alta flexibilidad en cuanto a la cantidad y distribución de los casilleros, pues dependiendo de la necesidad específica de la aplicación se introducen o retiran módulos esclavos. El circuito maestro es entonces una estación central que puede operar hasta 6 circuitos esclavos que a su vez controlan hasta 16 casilleros cada uno, para un total de hasta 96 espacios.

Figura 3. Distribución de control



A continuación se exponen la descripción y consideraciones de los dos circuitos involucrados, iniciando con el circuito maestro.

**1. Circuito maestro.** El corazón del circuito maestro es un PIC18F458 de Microchip. Este microcontrolador fue escogido por ser uno capaz de realizar hasta 10 millones de instrucciones por segundo y contar con 1536 bytes de memoria RAM interna. Además, se implementan en él tres fuentes de interrupción externa y tres diferentes registros para el direccionamiento indirecto. Todas estas cualidades permiten realizar las tareas requeridas por el circuito maestro a un bajo costo.

Se propone considerar el circuito maestro como el conjunto de siete funciones de realización simultánea como se muestra en la figura 4. Seguido, se hace un análisis descriptivo de la implementación electrónica para llevar a cabo cada función.

Figura 4. Funciones del Circuito maestro

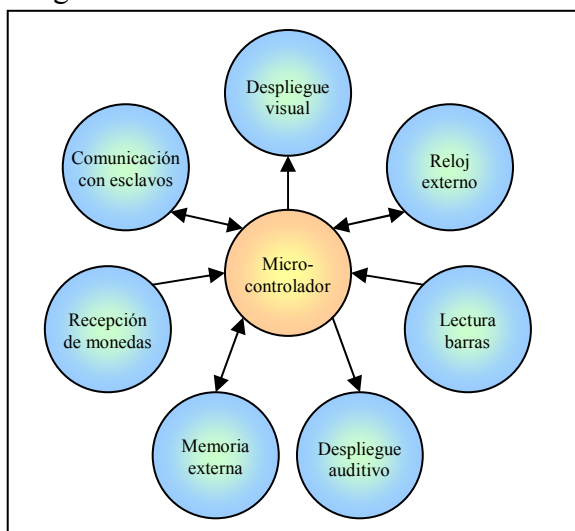
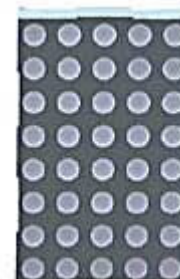


Figura 5. Matriz de LEDs

*a. Despliegue de mensajes visuales.* La emisión de mensajes visuales se realiza a través de la agrupación de arreglos matriciales de diodos emisores de luz como el que aparece en la figura, para crear una sola matriz de 30 columnas y 8 filas.



La operación de esta matriz consiste en activar debidamente cada uno de los diodos para que, en conjunto, creen los caracteres de los mensajes que se han de desplegar. Un primer planteamiento para llevar a cabo esta tarea implica activar las filas con la combinación deseada mientras una sola columna se aterriza y así, columna a columna, crear el mensaje cíclicamente. Aunque la velocidad del microcontrolador permite hacer hasta varios miles de barridos completos por segundo, existe la desventaja de que cada columna permanece encendida únicamente  $1/30$  del periodo del barrido, creando una luz tenue cuando el tiempo dedicado a cada columna es corto y un destello del mensaje, en general, cuando es largo.

Para aumentar la proporción del tiempo dedicado a cada columna, sin complicar excesivamente la programación de la rutina, se planteó la inclusión de tres registros cuya función es almacenar el dato correspondiente a una columna. La matriz de diodos se divide en tres submatrices de diez columnas y ocho filas, siendo estas últimas activadas por cada uno de los registros arriba mencionados. Con este planteamiento, el microcontrolador se ocupa de escribir a los registros y aterrizar de manera conjunta tres columnas a la vez, mejorando la relación a  $1/10$  del periodo de barrido.

Se escogieron registros de la familia HCMOS para que sean capaces de suministrar la corriente necesaria para la activación de los leds, eliminando la necesidad de introducir transistores discretos para el efecto. El microcontrolador ordena aterrizar las columnas a través de un demultiplexor de 16 líneas. Para hacer del circuito uno más compacto, se introduce un empaquetado de pares de transistores tipo Darlington capaces de conducir hasta 600mA, por demás suficiente en la conducción de la corriente consumida por hasta  $8 \cdot 3 = 24$  leds.

El programa del microcontrolador construye dinámicamente los mensajes a desplegar y los almacena en su memoria RAM. El ciclo principal se encarga de escoger cada uno de los valores haciendo uso de direccionamiento indirecto.

*b. Reloj externo.* Quizás el elemento más importante después del microcontrolador mismo sea el reloj de tiempo real (RTC, por sus siglas en inglés). El integrado DP8573 de National Semiconductor fue el escogido para la tarea de llevar el conteo del tiempo de manera independiente al control realizado por el PIC. La utilización de un componente externo para llevar a cabo esta tarea se justifica al tomar en cuenta la importancia de mantener un confiable registro de la hora en la presente aplicación. El reloj externo tiene la cualidad de poder respaldar su funcionamiento en una batería de manera tal que, aún cuando exista una pérdida en la alimentación del circuito general, éste pueda mantener su conteo del tiempo.

El DP8573 es completamente programable, lleva conteo desde los milisegundos hasta los años y es capaz de generar interrupciones cada vez que alguno de estos contadores se incrementa. En esta aplicación se programó la generación de interrupciones cada 1/100seg para que el microcontrolador pueda tener conocimiento del transcurso de fracciones de segundo para realizar diferentes tareas tales como el encendido y apagado de los dos puntos clásicamente observados en los relojes digitales. Es cada vez que ocurre una de estas interrupciones que, al estar configurando la hora del reloj, se aumenta en uno los minutos.

La hora en el administrador de casilleros tiene formato de 24 horas para identificar plenamente la hora del día sin necesidad de recurrir a señalización de antes o pasado meridiano (AM o PM).

*c. Lectura de barras.* Según Marshall Brain, existen más de 20 diferentes tipos de codificación por barras. Esta implementación se desarrolla alrededor del Código 128, un código denso altamente utilizado alrededor del mundo. Después de cierta investigación, se determinó que los números de carné en la Universidad del Valle se codificaron utilizando este formato. Al haber hecho codificaciones independientes se comprobó, sin embargo, que los carnés universitarios carecen del dígito *checksum* que permite la validación de los códigos leídos.

Debido a la complejidad mecánica en la alineación de sensores se optó por utilizar un lector disponible comercialmente para realizar la tarea de interpretación de las barras. En la figura 6 aparece el MS140 de Unitech, el lector escogido. Este dispositivo reporta de manera serial los equivalentes ASCII de los caracteres leídos utilizando el protocolo RS232. Por esta razón, el circuito maestro incorpora la electrónica necesaria para alcanzar los niveles especificados en el protocolo y recibir esta información utilizando el módulo de comunicación serial asíncrona. El programa convierte los códigos ASCII al equivalente BCD de cada dígito con el propósito de almacenar de manera más compacta y manejable el número de carné.

Figura 6. Lector de barras (MS140 de Unitech)



*d. Reproducción de mensajes audibles.* Con el afán de enriquecer la forma en que el sistema administrador de casilleros se relaciona con el usuario, se introdujo la emisión de mensajes auditivos que, junto a los visuales, indican al primero su identificación, el casillero asignado y la necesidad de cancelar un monto por su utilización. El circuito integrado escogido para la implementación de voz es el ISD2560 de Information Storage Devices. Es un dispositivo CMOS que incorpora un oscilador, preamplificador de micrófono y amplificador para bocina.

El chip puede retener información de audio de hasta 60 segundos de duración en memoria no volátil. El modo de operación escogido fue el de mensajes individuales direccionables que permite la construcción de oraciones a partir de palabras discretas. Se hizo necesaria la construcción de un circuito independiente para realizar la grabación de estas palabras.

Una aplicación computacional sintetizadora de voz se utilizó para la emisión de los sonidos y las direcciones fueron escogidas en concordancia con el mapa de direcciones (ver tabla 7 en el Anexo A)

Una vez incorporado al circuito maestro, el ISD2560 se coloca en modo estrictamente de *playback* y debido al amplificador interno de audio, se conecta directamente a la bocina que traducirá la señal eléctrica a una perturbación audible.

*e. Memoria externa.* Para que ésta sea una aplicación robusta, es menester garantizar que aún en el caso de una falla en el suministro eléctrico del módulo central, éste sea capaz de mantener el registro y control de cada espacio de almacenamiento. Para cumplir con este requerimiento, la información de ocupación de cada casillero se almacena en memoria externa eléctricamente programable, específicamente en el integrado AT24C08 de Atmel. Este semiconductor provee 1024x8 (8kbits) de espacio de almacenamiento en un empaquetado de 8 patas. En él, el PIC18F458 almacena los cinco dígitos del carné del usuario que ocupa el casillero, el estado del contador de minutos, de horas y de día de la semana a partir del cual el casillero se empezó a usar para poder eventualmente recuperar la información y a partir de ella determinar ocupación y calcular costos. El hecho de almacenar como parte del registro el día de la semana en que se empieza a usar el servicio permite definir límites de tiempo de almacenamiento de hasta siete días.

Para su correcto funcionamiento es necesario “formatear”, o bien, pre-programar esta EEPROM con la identificación de los casilleros, para lo cual fue necesario construir un circuito independiente. El proceso implica el grabado de las etiquetas de todos los casilleros que el sistema puede manejar, es decir, desde el A0 hasta el FF, un total de 96 espacios de almacenamiento. Inicialmente en su rutina, el maestro declara ocupados todos aquellos casilleros que según la configuración de los esclavos, no están en disposición de ser utilizados, ya sea por estar fuera de servicio o simplemente por no existir (ver tabla 8, Anexo 8) La comunicación con este dispositivo se lleva a cabo utilizando el protocolo I2C por lo que se encuentra conectado a la misma red que comunica a los esclavos con el maestro.

*f. Recepción de monedas.* Una característica que convierte al sistema administrador de casilleros en una implementación atractiva comercialmente es su capacidad de cobrar automáticamente por la utilización del recurso limitado de casilleros.



El circuito realiza la tarea de cobro por medio del módulo identificador de monedas MS16, de Mars Electronics International (ver figura 7) Este dispositivo se encuentra en teléfonos monederos y por las especificaciones de los mismos, es una solución de bajo consumo eléctrico. Su funcionamiento se centra en la identificación de la forma en la que una aleación específica altera un campo eléctrico. Este dato, junto al peso de una moneda determinada, permite validarla e identificarla. Una vez realizado este proceso, el módulo interrumpe e informa al microcontrolador.

El programa del maestro está diseñado de manera tal que la tarifa de uso por hora puede ser modificable desde 25 centavos hasta un máximo de Q2.50 por hora. El sistema acepta monedas de 25 y 50 centavos de quetzal, así como monedas de un quetzal (Q.1.00)

La recepción de monedas está habilitada únicamente cuando el dispositivo maestro despliega un saldo, que se actualiza debidamente después de introducida cada moneda, hasta que el saldo total ha sido cancelado. Una vez ocurre esto, el registro de EEPROM se elimina, dejando disponible nuevamente el casillero hasta ahora ocupado.

*g. Comunicación con circuitos esclavos.* El circuito maestro se encarga de la lógica de control, el registro y la interfase con el usuario pero son módulos esclavos los encargados de operar directamente las chapas en los casilleros. De esta cuenta, una vez el maestro asigna un nuevo casillero, ordena al respectivo esclavo activar el solenoide que permitirá abrir la puerta. Es por medio de señalización a los esclavos que el maestro autoriza la apertura de un casillero una vez ha sido saldada la cuenta por su utilización.

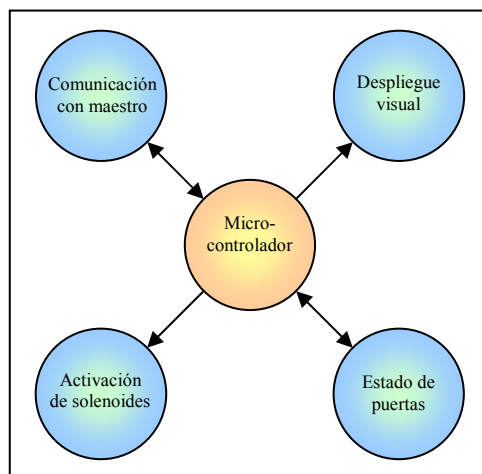
La comunicación entre módulos se realiza a través de una red I2C. Este protocolo permite que entre los diferentes elementos exista un menor número de cables y es importante resaltar que la concepción modular de esta solución permite dimensionar el sistema de acuerdo a las necesidades específicas de cada implementación.

**2. Circuito esclavo.** El circuito esclavo es gobernado por un microcontrolador PIC18F248. Este pequeño computador cuenta con las características de interrupción y direccionamiento indirecto del PIC18F458, pero con memoria de programa y de acceso aleatorio de menor tamaño. Su empaquetado es también más pequeño (sólo 28 pines) razón primordial por la cual se escogió para encargarse de la operación en los circuitos esclavos.

Un detalle interesante sobre los elementos que conforman el circuito esclavo es la inclusión de dos interruptores BCD. El objetivo del primero de ellos es permitir la fácil determinación de la identificación del esclavo. De esta manera se contribuye a la flexibilidad deseada al momento de implementar la solución. El segundo interruptor permite fijar el número máximo de casilleros que un determinado esclavo puede controlar.

Se propone la consideración de un circuito que realice cuatro funciones cíclicamente de acuerdo con la siguiente gráfica.

Figura 8. Funciones del circuito esclavo

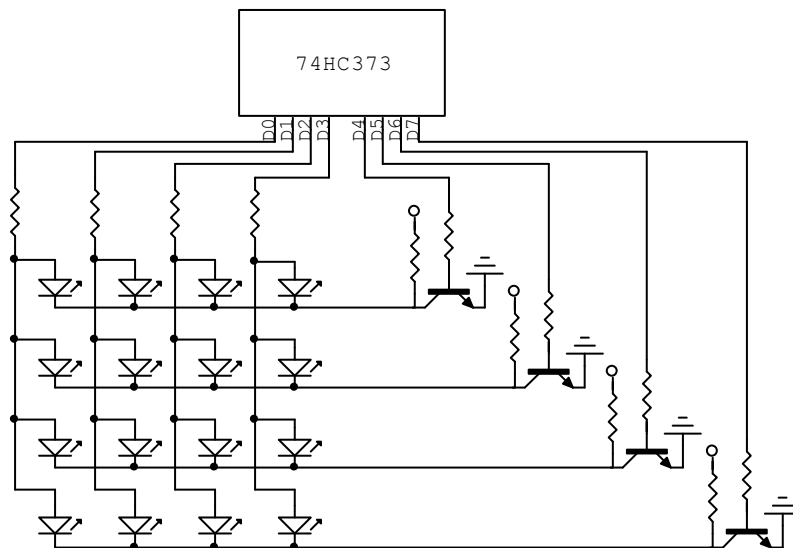


A continuación se describe brevemente la implementación de cada función:

*a. Indicadores visuales.* El circuito esclavo implementa un control sobre un conjunto de hasta 16 diodos emisores de luz que permiten establecer en cualquier momento la disponibilidad de casilleros. Estos leds verdes se ubican en sus puertas y se mantienen encendidos mientras el espacio este desocupado. Una vez se ha recibido una orden de apertura, el esclavo procede a apagar la luz para indicar que el casillero ha sido o está en proceso de ser asignado.

Con el objeto de disminuir al máximo el número de componentes en el circuito, se implementa un barrido matricial que permite controlar con un solo registro de 8 bits el total de 16 leds. El esquemático en la figura 9 permite la fácil comprensión del proceso.

Figura 9. Esquemático del barrido matricial de LEDs

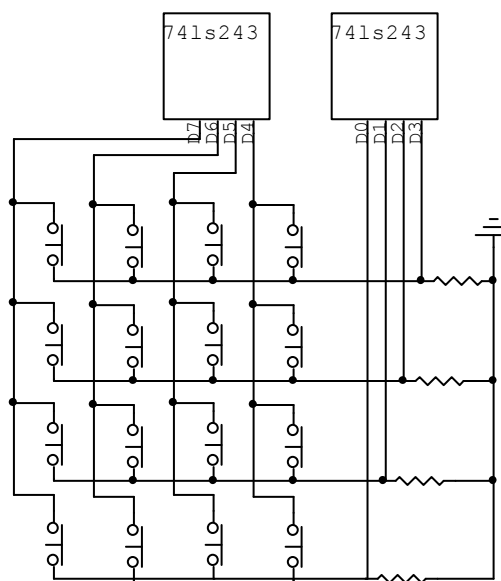


Convenientemente se permite que cada una de las filas de la matriz de leds conduzca desplegando los cuatro datos que en ese momento se encuentran en las columnas. Las terminales del circuito esclavo se encuentran debidamente señalizadas para la instalación de los cables que conducirán estas señales hasta las puertas de cada casillero.

b. *Estado de las puertas.* Es necesario contar con una forma de establecer el estado de las puertas en los casilleros y de esta forma determinar efectivamente cuando uno de ellos se empezó a usar o bien, corroborar que una orden de apertura fue debidamente atendida. Se determinó que una implementación sencilla que permite este control es la inclusión de interruptores en cada una de las puertas. Puesto que la mayor parte del tiempo todas las puertas se mantienen cerradas, se escogieron interruptores normalmente cerrados, de manera que cierran el circuito al que pertenecen en el momento en que las puertas están abiertas. De esta forma se optimiza el consumo eléctrico ya que existe flujo de corriente únicamente durante el momento en el que el interruptor está cerrado, es decir, puerta abierta.

Manteniendo la posición de minimización en el uso de componentes, se decidió implementar un conjunto de 2 *transcievers* 74LS243 en el bus de datos que cuando son habilitados permiten el flujo de información desde el bus hacia los interruptores en uno de los casos y desde los estos hacia el bus en el otro. La figura tal se incluye para ilustrar el proceso.

Figura 10. Esquemático del barrido matricial de interruptores



Aunque los interruptores acá aparecen cerrados es necesario recordar que cuando las puertas estén cerradas, estos estarán accionados (abiertos) reflejando cero voltaje en las terminales D0..D3.

*c. Activación de solenoides.* Sin duda la tarea más importante encomendada al circuito esclavo es la operación de los solenoides que permiten la apertura de las puertas de los casilleros.

Los solenoides escogidos operan con un voltaje de 12V y al ser activados consumen 250mA. Debido a su potencial de operación fue necesario considerar componentes capaces de conducir esta corriente, escogiendo finalmente transistores Darlington en empaquetado NTE2018.

En el circuito existen dos flip-flops de 8 bits encargados de mantener en sus salidas el estado de cada uno de los solenoides. El primero para el grupo de los casilleros 0 al 7 y el segundo para los correspondientes del 8 al F. Las salidas de estos se conectan directamente a cada uno de los pares de transistores Darlington. De esta manera, cuando el microcontrolador escribe un 1 en una de las posiciones de los flip-flops, el par Darlington coloca a tierra su colector permitiendo la conducción eléctrica a través de la bobina del solenoide, de esta manera activándolo.

El programa obliga la condición descrita durante un periodo programable de tiempo o mientras se mantenga abierta la puerta. Una vez ésta se cierra, el solenoide se desactiva, asegurando la puerta.

*d. Comunicación con circuito maestro.* La comunicación maestro-esclavo se realiza a través de una red I2C, usando un cordón de cuatro hilos con terminales para conector RJ11. En él viajan las señales de datos y reloj, así como la referencia (tierra) que se comparte con el maestro y su alimentación de 12V. Este potencial se utiliza para operar los solenoides directamente y de manera regulada para alimentar toda la electrónica TTL.

## C. DISTRIBUCIÓN, MAPEO Y PROGRAMACIÓN

En este capítulo se incluye la información sobre la distribución de los puertos de los microcontroladores en cada uno de los circuitos, la forma en que se distribuyeron las direcciones en el bus correspondiente (mapeo) y las condiciones de flujo del *firmware* desarrollado.

**1. Circuito maestro.** El microcontrolador PIC18F458 se presenta en un empaquetado de 40 pines con 32Kbytes de memoria de programa (ROM), 1536 bytes de memoria de acceso aleatorio (RAM) y cinco puertos distribuidos de la siguiente forma:

Tablas 1 y 2.  
Distribución de puertos en Circuito Maestro

Pin	Nombre	Función
2	RA0	Bus de direcciones, línea A0
3	RA1	Bus de direcciones, línea A1
4	RA2	Bus de direcciones, línea A2
5	RA3	Bus de direcciones, línea A3
6	RA4	Bus de direcciones, línea A4
7	RA5	Bus de direcciones, línea A5
14	RA6	Bus de direcciones, línea A6
33	RB0	Interrupción de voz
34	RB1	Interrupción de reloj externo
35	RB2	Interrupción de monedas
36	RB3	Identificación moneda, dato 0
37	RB4	Identificación moneda, dato 1
38	RB5	Identificación moneda, dato 2
39	RB6	Orden de formateo de EEPROM
40	RB7	Activación de módulo monedero

Pin	Nombre	Función
15	RC0	
16	RC1	
17	RC2	
18	RC3	Red I2C, línea de reloj
23	RC4	Red I2C, línea de datos
24	RC5	
25	RC6	Serial asíncrono, línea de transm..
26	RC7	Serial asíncrono, línea de recepción.
19	RD0	Bus de datos, línea D0
20	RD1	Bus de datos, línea D1
21	RD2	Bus de datos, línea D2
22	RD3	Bus de datos, línea D3
27	RD4	Bus de datos, línea D4
28	RD5	Bus de datos, línea D5
29	RD6	Bus de datos, línea D6
30	RD7	Bus de datos, línea D7
8	RE0	
9	RE1	
10	RE2	Reloj externo, línea de ajuste

La tabla 6 en el anexo A describe la utilización de las posiciones de memoria RAM interna del microcontrolador. Los bancos 3, 4 y 5 no se utilizaron y por ello no aparecen en el mapa. El banco 0 se reservó para las variables de uso general en el programa, todas ellas en la región accesible sin necesidad de especificar banco (*access bank*). Por su lado, el banco 1 contiene toda la información necesaria para el despliegue de los mensajes tanto visuales como audibles. Finalmente el banco 2 contiene la información necesaria para el despliegue visual del saldo, cuando es requerido el pago por el uso del servicio.

El principio de arquitectura digital de la implementación de un bus de direcciones le permite al microcontrolador poder dirigir su comunicación a muchos periféricos de manera relativamente sencilla a través de una debida decodificación. La tabla 7, incluida en el anexo A, presenta la distribución de las 128 direcciones disponibles en el bus de 7 bits en el circuito maestro. El mapeo de estas direcciones se logró con el uso de un conjunto de compuertas lógicas, tanto NAND (74LS00), NOR (74LS02) y NOT (74LS04), así como un demultiplexor de 2-4 (74LS139) y uno de 4-16 (74LS154).

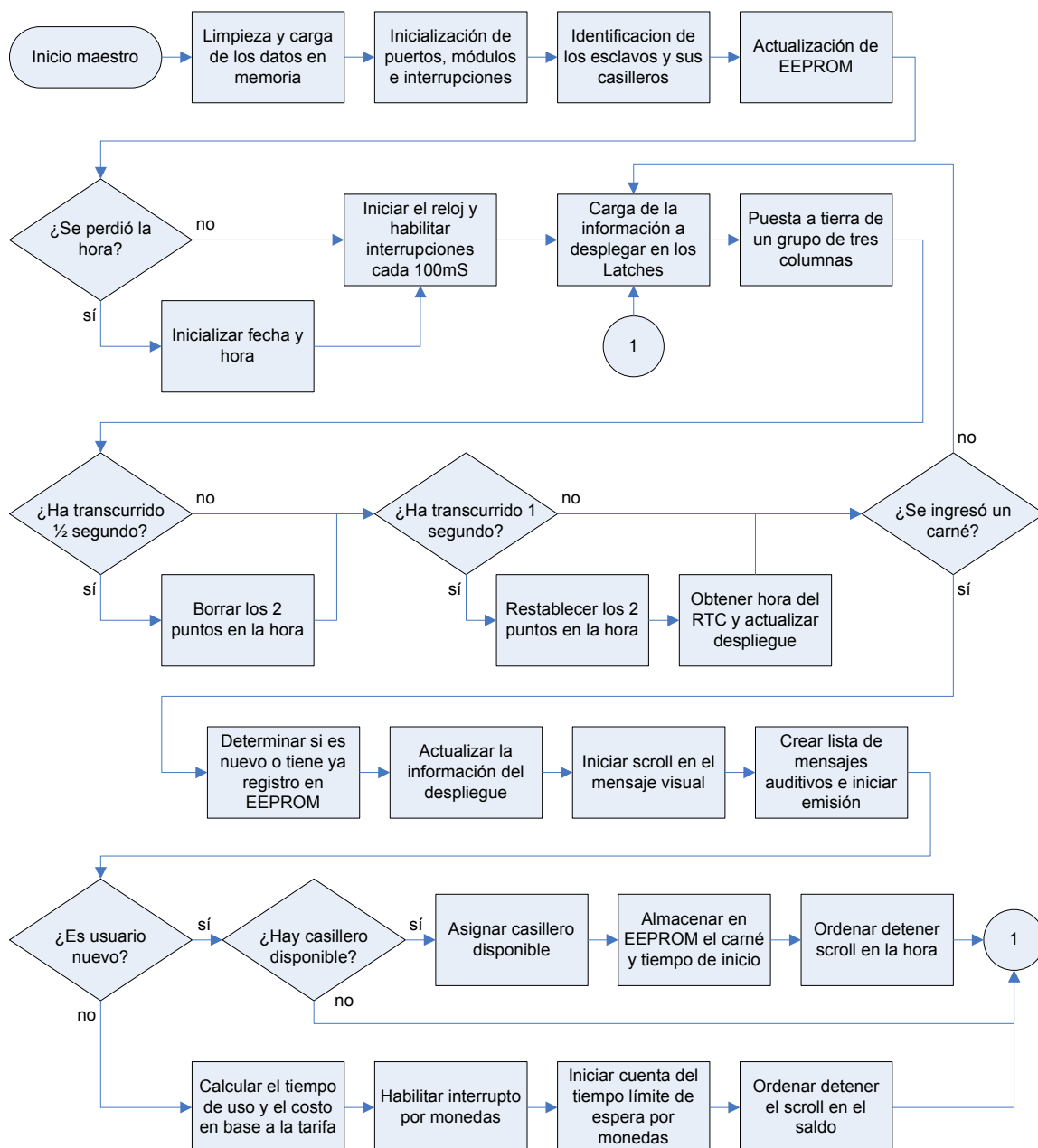
La memoria electrónicamente programable que reside fuera del microcontrolador también tiene su respectivo mapa de memoria. Este se incluye en el anexo A, tabla 8. Como se especificó en el literal B., este mapa contempla la debida colocación de la información del usuario en el renglón correspondiente al casillero que utiliza. Es ésta información la que le permite al microcontrolador determinar si un número de carné que se ingresa tiene ya registro en memoria, es decir, determinar si un usuario es nuevo o ya tiene un casillero asignado.

El formato de las direcciones de memoria EEPROM las dictó la hoja de especificaciones del integrado que se utilizó y cabe mencionar que fue tomado en cuenta en el desarrollo para asignar convenientemente el formato de direcciones I2C que tienen los esclavos como se especificará más adelante.

En el diagrama de la figura 11 se establece el flujo lógico del funcionamiento del programa principal que ejecuta el PIC18F458 para el control del circuito maestro.

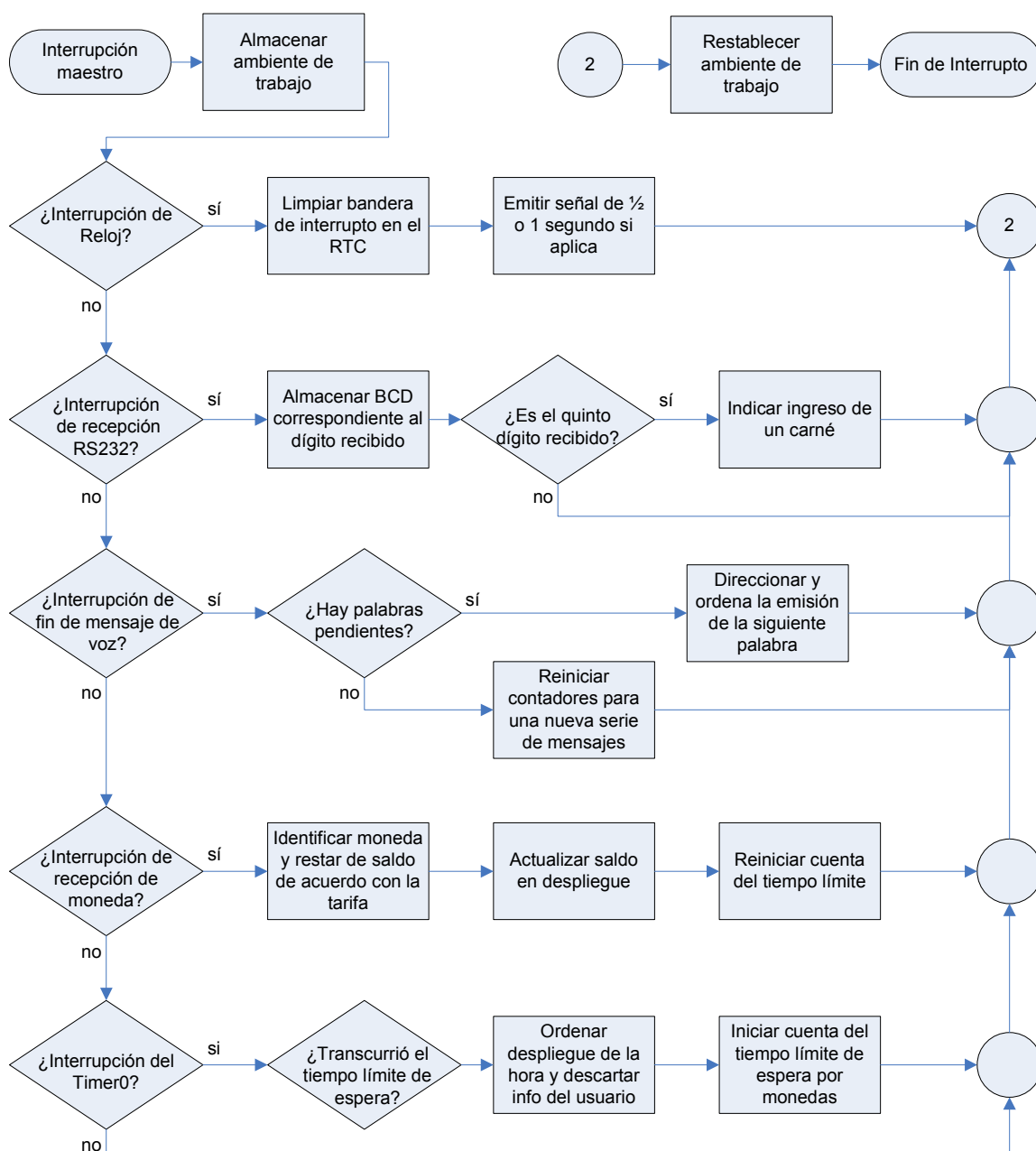
Como se puede observar, la tarea más importante de la rutina principal es obtener el dato de la hora del RTC y desplegar correctamente los mensajes visuales.

Figura 11. Diagrama de flujo, rutina principal del microcontrolador maestro



Es tarea del código que atiende las interrupciones hacer el resto de funciones involucradas, como se muestra en la figura 12.

Figura 12. Diagrama de flujo, rutina de interrupción del microcontrolador maestro



La forma en la que se implementó este flujo en lenguaje ensamblador se detalla en el código fuente del maestro incluido en el anexo B.

**2. Circuito esclavo.** Debido a la disminución en la complejidad de las tareas encomendadas al circuito esclavo, aunado al reducido espacio físico, el PIC18F248 es el microcontrolador encargado de la realización de sus tareas. Se presenta en un empaquetado de 28 pines con 16Kbytes de memoria de programa (ROM), 768 bytes de memoria de acceso aleatorio (RAM) y tres puertos distribuidos de la siguiente manera:

Tablas 3 y 4.  
Distribución de puertos en circuito esclavo

Pin	Nombre	Función
2	RA0	Habilitación del <i>transceiver</i> de filas para los interruptores de las puertas
3	RA1	Habilitación del flip-flop de estados de los solenoides del grupo 2
4	RA2	Habilitación de la salida del latch de los interruptores de identificación
5	RA3	Habilitación del <i>transceiver</i> de cols. para los interruptores de las puertas
6	RA4	
7	RA5	Habilitación del latch de los diodos emisores de luz
10	RA6	
21	RB0	Bus de datos, línea D0
22	RB1	Bus de datos, línea D1
23	RB2	Bus de datos, línea D2
24	RB3	Bus de datos, línea D3
25	RB4	Bus de datos, línea D4
26	RB5	Bus de datos, línea D5
27	RB6	Bus de datos, línea D6
28	RB7	Bus de datos, línea D7

Pin	Nombre	Función
15	RC0	
16	RC1	
17	RC2	
18	RC3	Red I2C, línea de reloj
23	RC4	Red I2C, línea de datos
24	RC5	Habilitación del flip-flop de estados de los solenoides del grupo 1
25	RC6	
26	RC7	

En el anexo A, tabla 9, aparece la tabulación de las posiciones de RAM interna utilizadas por el programa del circuito esclavo. Únicamente se utilizaron posiciones de memoria en el banco de acceso global (*access bank*) con lo que se evitó la necesidad de estar seleccionando el banco en el que se encontraban las variables.

A diferencia del circuito maestro, en este circuito no fue necesaria la implementación de un bus de direcciones puesto que, los periféricos a controlar pueden ser habilitados directamente con pines de los puertos por su reducido número. Esta ventaja elimina la necesidad de incluir compuertas lógicas para decodificar direcciones.

La identificación del esclavo se hace a través de los interruptores BCD que se encuentran implementados en el circuito. Los nombres válidos para cada esclavo son los seis últimos valores hexadecimales, es decir, A, B, C, D, E y F. Este valor es leído por el programa en su configuración inicial para crear la correspondiente dirección I2C a la que el circuito ha de responder, de acuerdo con la tabla 5.

Tabla 5. Equivalencias de direcciones I2C en esclavos

ID fisica	Identificación I2C
A	110 1010 0 = D4
B	110 1011 0 = D6
C	110 1100 0 = D8
D	110 1101 0 = DA
E	110 1110 0 = DC
F	110 1111 0 = DE

Esta equivalencia entre dirección física y dirección I2C se estableció con el propósito de estandarizar el formato de las direcciones a las que responde la memoria EEPROM en el maestro y las direcciones a las que tienen que responder los esclavos. Además se consideró que según la especificación de ese protocolo de comunicación serial, el bit menos significativo de la dirección esta reservado para establecer si se está realizando una lectura o una escritura desde el maestro hacia sus esclavos.

En el diagrama de la figura 13 se ilustra el flujo lógico del funcionamiento del programa principal que ejecuta el PIC18F248 para el control del circuito esclavo. La lógica de la rutina de servicio de interrupción aparece diagramada en la figura 14.

Figura 13. Diagrama de flujo, rutina principal del microcontrolador esclavo

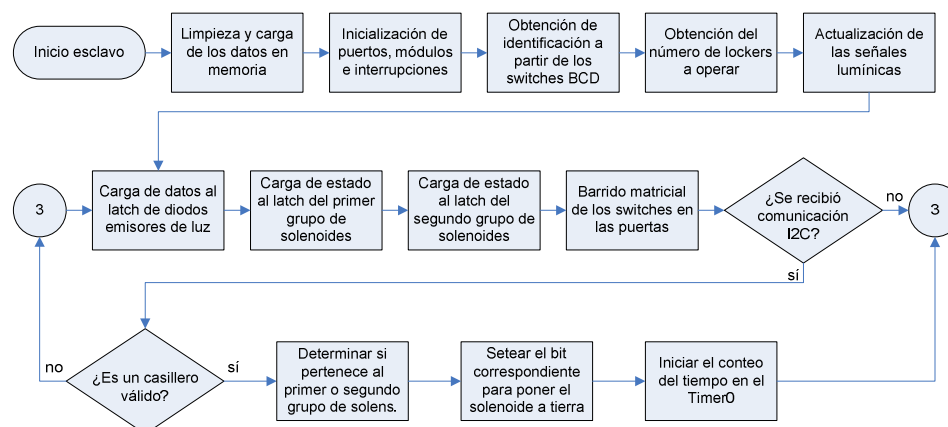
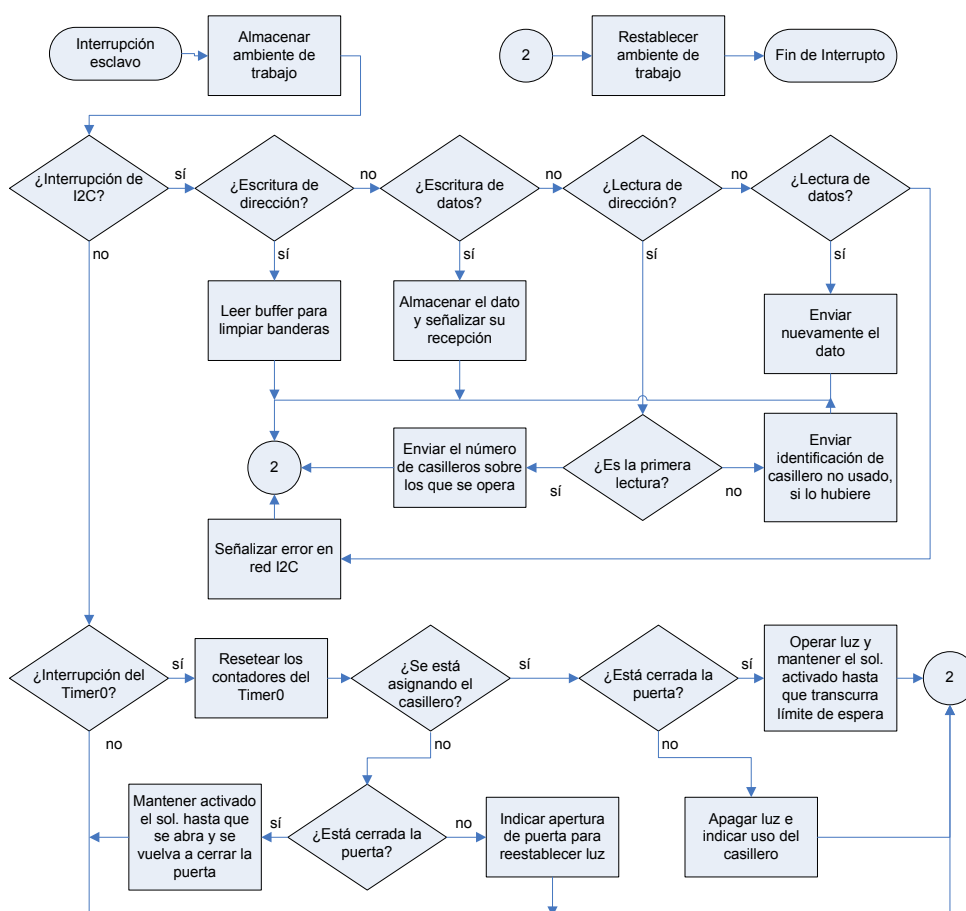


Figura 14. Diagrama de flujo, rutina de interrupción del microcontrolador esclavo



El lector puede consultar el código fuente de estas rutinas con sus respectivos comentarios en el anexo B.

## IV. RESULTADOS

---

---

Este trabajo de investigación tuvo como resultado la creación de un administrador de casilleros con acceso por medio de código de barras y mecanismo de cobro automático. Para el efecto fueron construidos un circuito maestro y un ejemplar de circuito esclavo siguiendo todas las consideraciones anotadas en el literal B del capítulo III.

El circuito maestro fue montado en una placa perforada de 8.5"x11" mientras que el esclavo se construyó en una de 8.5"x5.5". Se escogió esta plataforma de desarrollo para los prototipos por constituirse como un punto medio entre la flexibilidad que ofrece un protoboard y la rigidez, pero debida presentación, de una tarjeta impresa. Como resultado se tiene en ambos circuitos una tarjeta de dos caras. En una de ellas las conexiones se realizaron creando pistas de estaño y cobre que unieron los respectivos contactos en forma horizontal, mientras que en la cara posterior las conexiones se hicieron usando alambre calibre 22 para unir puntos de manera vertical. Los integrados especiales, entiéndase, microcontrolador, latches, flip-flops, demultiplexores, etc. fueron montados en *sockets* que permitían retirarlos del circuito sin necesidad de desoldar sus terminales. Se procuró en todo momento colocar las componentes de manera tal que se aprovechara al máximo el espacio disponible.

En la figuras 15, 16 y 17 se presenta los esquemáticos del circuito maestro. Se optó por dividir en tres partes el circuito con el afán de facilitar su lectura y comprensión. Los puntos de conexión entre las partes se anotan como entradas o salidas en cada una de ellas mientras que los buses de direcciones y datos constituyen uno mismo en cada caso.

En el primer esquemático aparece la electrónica involucrada en la selección de la columna que se pone a tierra para la construcción de los mensajes visuales. En el siguiente esquemático se ilustra la sección central del circuito que incorpora los integrados más relevantes en la realización de las tareas. Finalmente en el tercer esquemático aparece la circuitería relacionada directamente con el despliegue visual.

Figura 15. Selección de columnas de despliegue en el circuito maestro

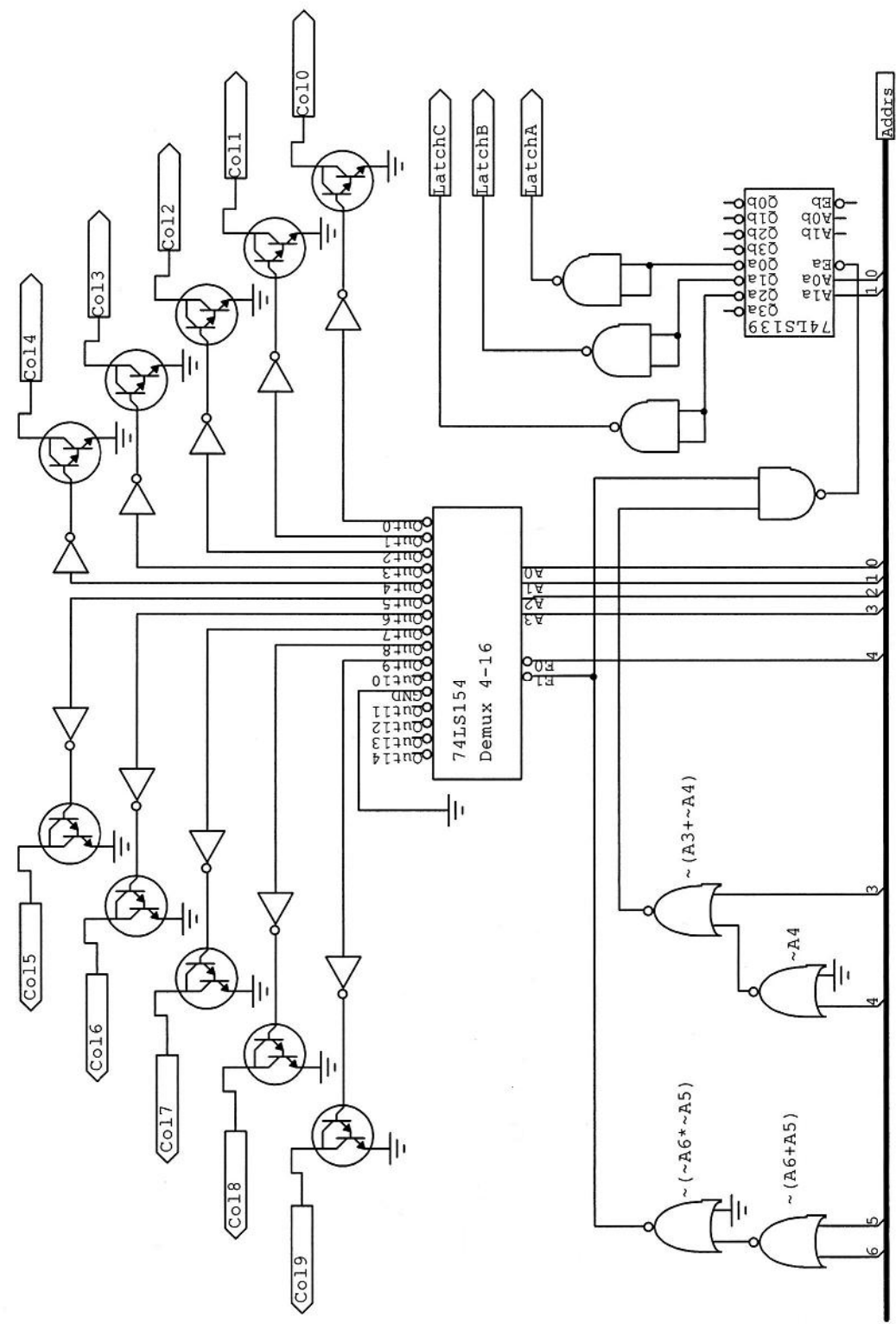


Figura 16. Sección central del circuito maestro

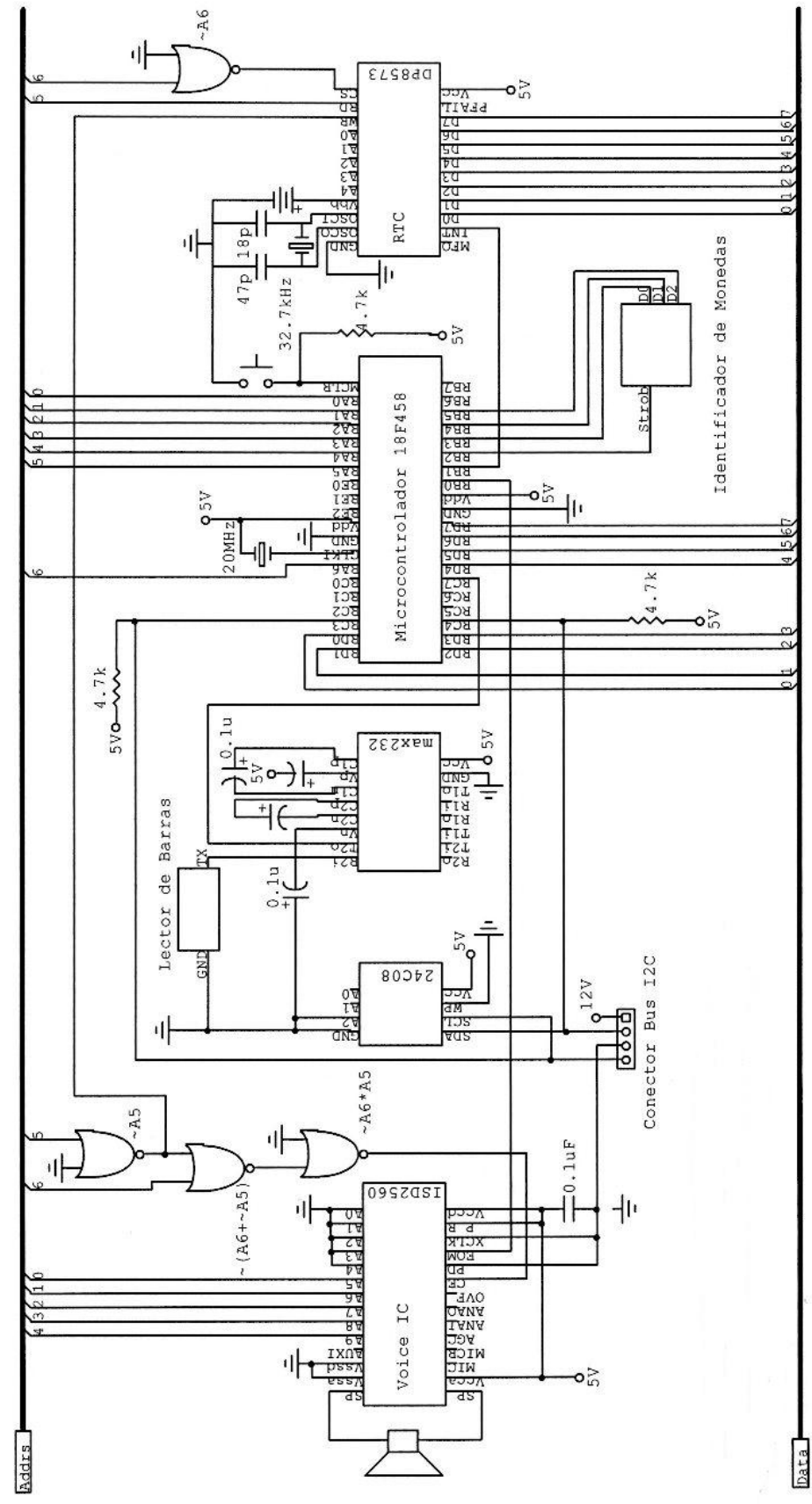


Figura 17. Despliegue en matrices de diodos emisores de luz en el circuito maestro

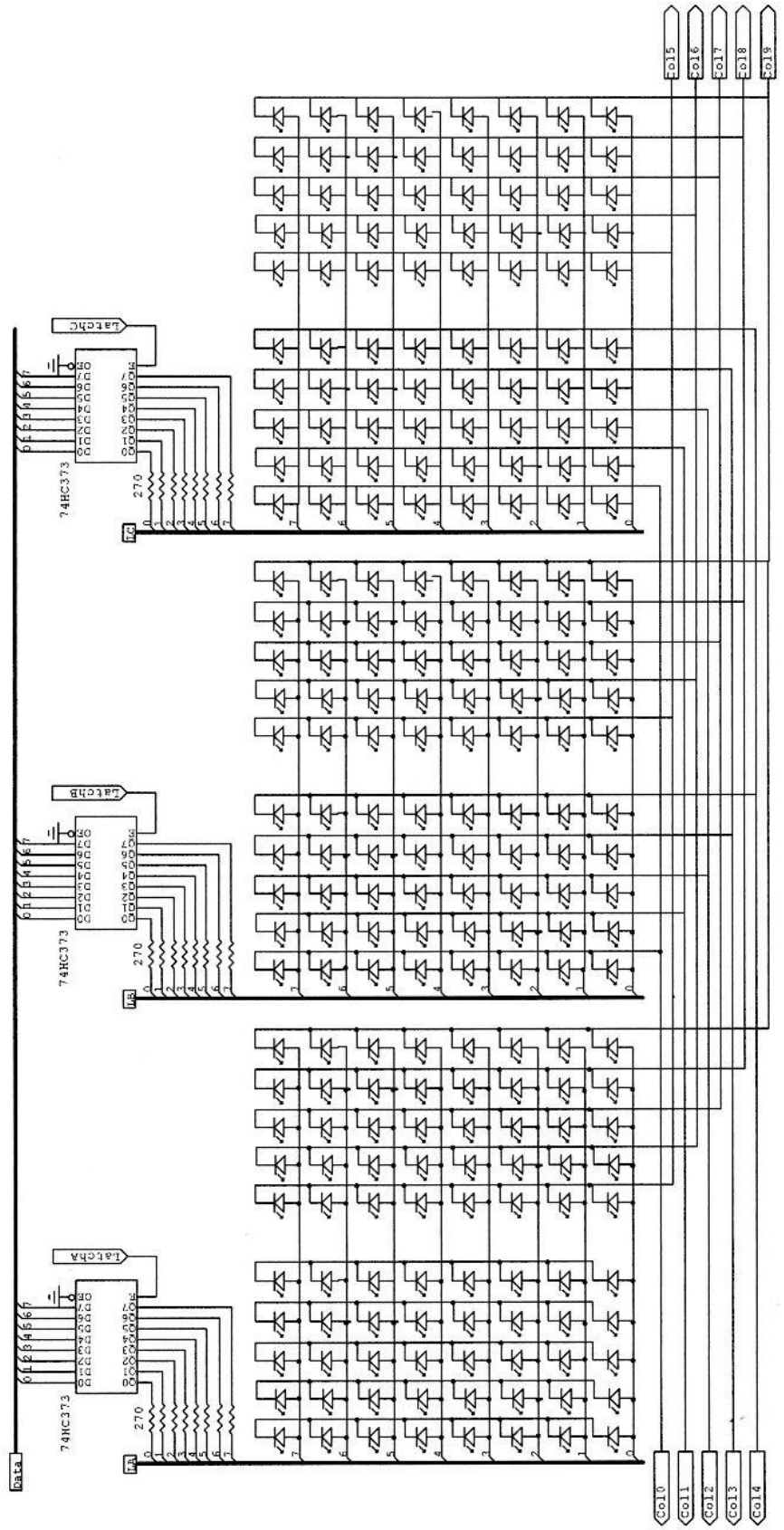
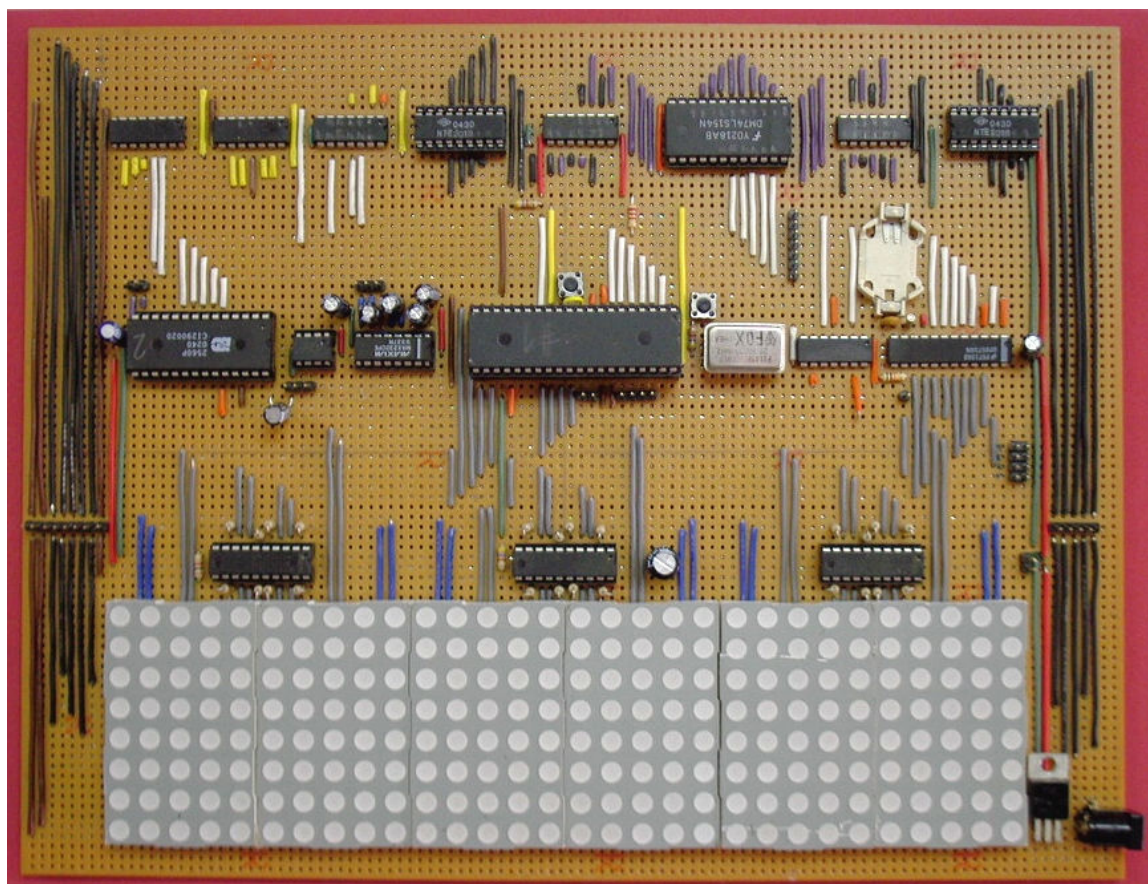


Figura 18. Fotografía del circuito maestro



En la fotografía del circuito maestro arriba se han dejado fuera los elementos exteriores como el lector de barras, el identificador de monedas, la bocina de audio, entre otros. Si se observa detenidamente se puede identificar de manera sencilla en la parte superior la parte de selección de la columna a poner a tierra con su demultiplexor 4-16 y las respectivas compuertas. En la siguiente fila de componentes sobresale el microcontrolador al centro seguido en ambas direcciones de los integrados de alta relevancia como el ISD2560 (integrado controlador de voz) en el extremo izquierdo y en el extremo opuesto el DP8573 (reloj de tiempo real). Por último, se observa la matriz de diodos emisores de luz en la parte inferior con los latches de alta corriente inmediatamente arriba.

En la figura 19 se presenta el esquemático del circuito esclavo.

Figura 19. Circuito esclavo

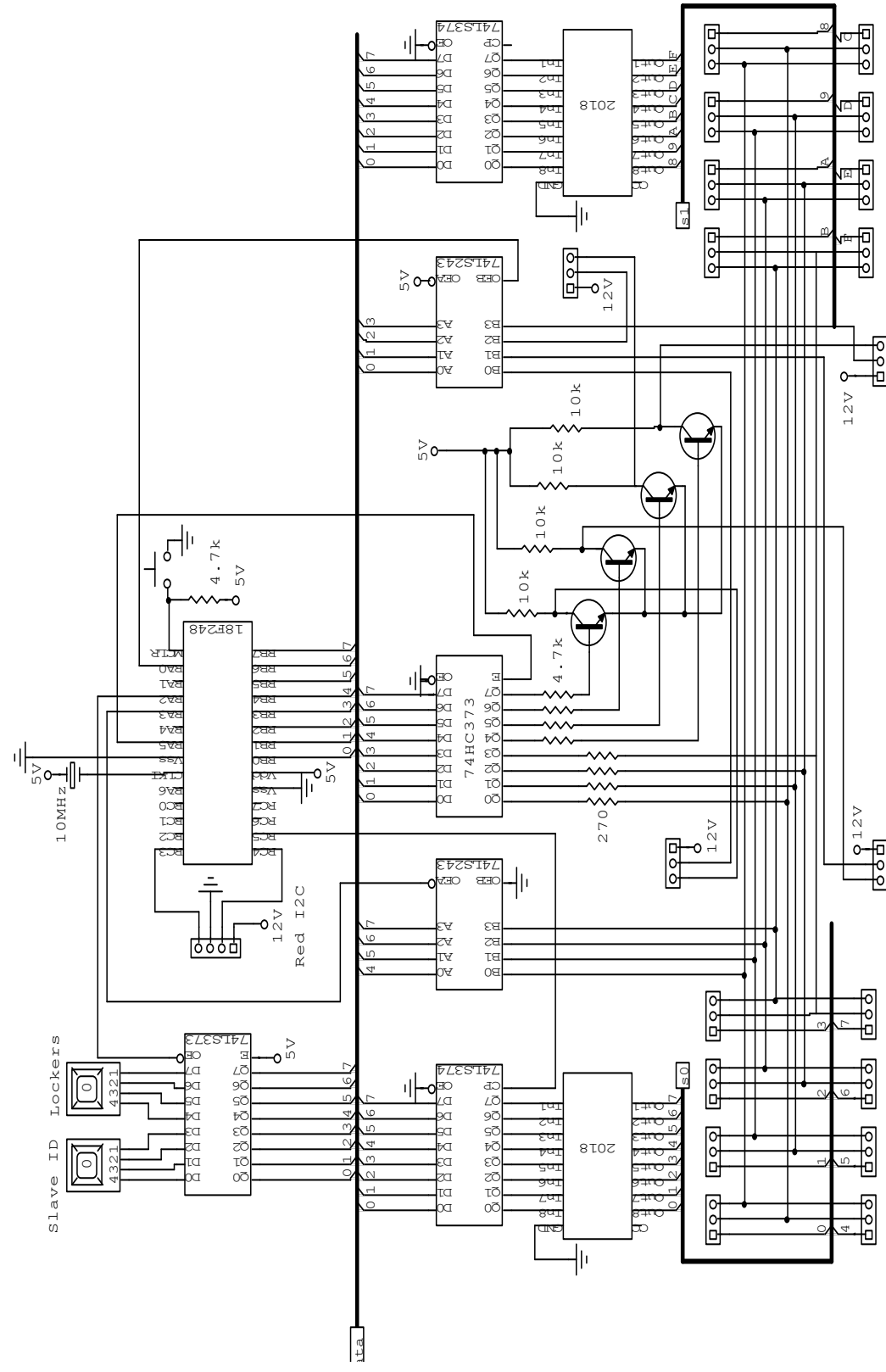
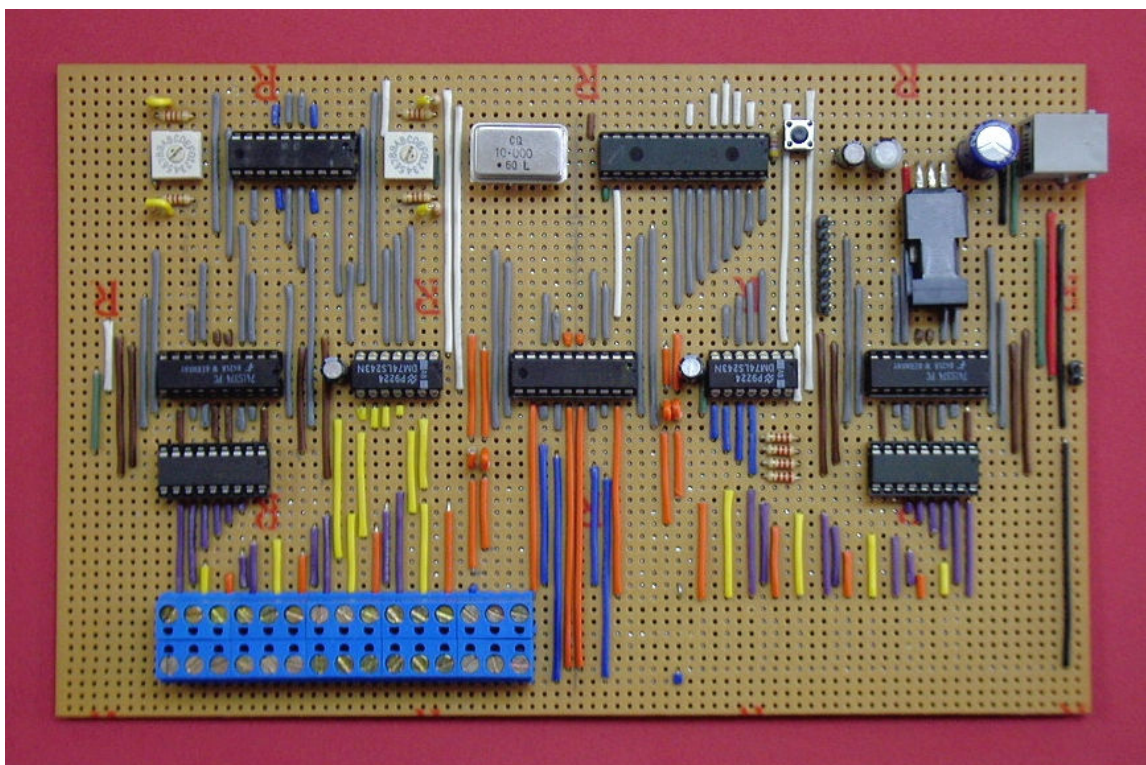


Figura 20. Fotografía del circuito esclavo



La fotografía muestra la implementación física del circuito esclavo. En la esquina superior izquierda de este circuito se pueden observar los dos interruptores BCD utilizados para identificar al esclavo y determinar el número máximo de casilleros a su cargo. El PIC18F248 aparece a su izquierda y en extremo derecho el conector RJ11 a través del cual se establece la comunicación con el circuito maestro. En la siguiente fila de componentes hacia abajo se presentan los flip-flops, *transcievers* y latches utilizados. Finalmente, en azul hasta abajo, aparecen las terminales a las que se conectan los cables que van hacia las chapas de los casilleros, sus indicadores lumínicos e interruptores.



## V. CONCLUSIONES Y RECOMENDACIONES

---

---

El desarrollo de este trabajo de investigación dio la siguiente serie de conclusiones:

- La concepción modular del administrador permite una alta flexibilidad en su implementación al ser cada circuito esclavo independientemente configurable.
- El planteamiento de la división de la matriz de mensajes visuales en subgrupos de menor tamaño permite dedicar más tiempo a cada columna y por ende mejorar la calidad del mensaje desplegado.
- La inclusión de un RTC independiente del microcontrolador permite mantener el conteo del tiempo aún cuando el circuito general está apagado.
- La simbología utilizada para codificación del carné estudiantil de la Universidad del Valle es la de Código 128, pero no incorpora el dígito de validación.
- La memoria externa electrónicamente programable permite mantener registro y control del uso de los casilleros aún cuando eventualmente falle el suministro eléctrico en el circuito principal.
- La implementación de un bus de direcciones en el circuito maestro le permite al microcontrolador gobernar un número de periféricos por demás superior a los que pudo haber dirigido directamente con sus salidas digitales.
- El Administrador de Casilleros con Acceso por medio de Código de Barras y Mecanismo de Cobro Automático es una solución viable para el manejo eficiente, rápido y seguro de espacios de almacenamiento temporal.

Como recomendaciones se pueden hacer las siguientes anotaciones:

- Se recomienda revisar la configuración de la codificación en barras de los carnés estudiantiles en la Universidad del Valle para incluir el dígito de validación y ajustarse de esta manera a la especificación Código 128 de manera completa.
- La implementación de una solución electrónica como la aquí presentada libera al personal de la carga que implica la atención de los casilleros por lo que se recomienda su implementación en áreas de alta demanda del recurso como la Biblioteca de la Universidad del Valle.
- La identificación por código de barras constituye un primer acercamiento al reconocimiento automático. Según lo requiera la aplicación, pueden considerarse otros métodos como la biometría a manera de recomendación para aumentar el nivel de seguridad.
- El cobro por el servicio prestado puede hacerse más amigable al usuario si se incorpora la aceptación de billetes o cargo a cuenta de cada usuario. Su implementación se recomienda para hacer de ésta una aplicación aún más aceptada.
- Para adaptar este dispositivo al rubro comercial, se puede incluir en las funciones del microcontrolador maestro, el control sobre el manejo de alcancías además de un registro sobre el total cobrado desde la última colecta, con el objetivo de administrar los fondos eficientemente.

## VI. BIBLIOGRAFÍA

---

---

- AN735. Using the PICmicro MSSP Module for Master I2C Communications.* 2000. Richard Fischer for Microchip Technology Inc. EEUU. 38 págs.
- AN734. Using the PICmicro SSP for Slave I2C Communication.* 2000. Stephen Bowling for Microchip Technology Inc. EEUU. 14 págs.
- AT24C08, 2 wire serial EEPROM.* 2002. Atmel. EEUU. 12 págs.
- Barcode Symbologies.* 2003. Barcode Island. EEUU. 18 págs.
- Brain, Marshall. 2003. «How UPC Bar Codes Work». *Boletín de las serie electrónica How Stuff Works.* [EEUU]
- DP8573A Real Time Clock (RTC).* 1993. National Semiconductor. EEUU. 16 págs.
- I2C Master Mode. Overview and use of the PICmicro MSSP I2C Interface with a 24xx01x EEPROM.* 2001. Microchip Technology Inc. EEUU. 88 págs.
- ISD2560/75/90/120 Products.* 1998. Information Storage Devices. EEUU. 30 págs.
- MS140I-2 RS232 Interface Slot Reader.* 2001. Unitech Inc. EEUU. 8 págs.
- PIC16F84 Data Sheet.* 2000. Microchip Technology Inc. EEUU. 86 págs.
- PIC16F87X Data Sheet.* 2001. Microchip Technology Inc. EEUU. 258 págs.
- PIC18FXX8 Data Sheet.* 2003. Microchip Technology Inc. EEUU. 395 págs.
- RS232. Tutorial on Data Interface and cables.* 2002. Arcelect Inc. EEUU. 12 págs.
- RS232 Multichannel Drivers/Receivers.* 2003. Maxim Integrated Products. EEUU. 17 págs.

Se usó como referencia bibliográfica las hojas de especificaciones de todos los otros componentes electrónicos utilizados. Estas se descargaron de los respectivos sitios de Internet, entre ellos:

- [www.national.com](http://www.national.com)
- [www.jameco.com](http://www.jameco.com)
- [www.fairchildsemi.com](http://www.fairchildsemi.com)
- [www.ti.com](http://www.ti.com)
- [www.onsemi.com](http://www.onsemi.com)
- [www.motorola.com](http://www.motorola.com)



## VII. ANEXO A

---

---

### Mapas de direcciones y memoria



Tabla 6. Mapa de memoria RAM en microcontrolador maestro

BANCO 0			BANCO 1			BANCO 2		
0 0	FLAG0	129 80	0 0	7E	129 80	7E	0 0	129 80
1 1	FLAG1	129 81	1 1	81	129 81	81	1 1	129 81
2 2	LOCKER	130 82	2 2	81	130 82	82	2 2	130 82
3 3	CARNET1	131 83	3 3	7E	131 83	00	3 3	131 83
4 4	CARNET2	132 84	4 4	00	132 84	00	4 4	132 84
5 5	CARNET3	133 85	5 5	01	133 85	15	5 5	133 85
6 6	MIR	134 86	6 6	41	134 86	0F	6 6	134 86
7 7	MRS	135 87	7 7	FF	135 87	00	7 7	135 87
8 8	DAY	136 88	8 8	01	136 88	1F	8 8	136 88
9 9	TEMP_PLAZA	137 89	9 9	00	137 89	10	9 9	137 89
10 A	RELOJAZO	138 9A	10 A	47	138 9A	06	10 A	138 9A
11 B	COLUMNA	139 9B	11 B	89	139 9B	00	11 B	139 9B
12 C	TWO_BCD	140 9C	12 C	91	140 9C	1F	12 C	140 9C
13 D	ONE_BCD	141 9D	13 D	81	141 9D	10	13 D	141 9D
14 E	CONTADOR	142 9E	14 E	00	142 9E	0F	14 E	142 9E
15 F	ORIGEN_DISP	143 9F	15 F	42	143 9F	00	15 F	143 9F
16 10	POS_CARNET	144 90	16 10	81	144 90	0E	16 10	144 90
17 11	DISTRTO_ALE	145 91	17 11	91	145 91	55	17 11	145 91
18 12	POS_POS	146 92	18 12	9E	146 92	40	18 12	146 92
19 13	TIMER1	147 93	19 13	00	147 93	00	19 13	147 93
20 14	TIMER2	148 94	20 14	70	148 94	12	20 14	148 94
21 15	TIMER3	149 95	21 15	10	149 95	00	21 15	149 95
22 16	RESTATUS	150 96	22 16	00	150 96	00	22 16	150 96
23 17	WORD_ADD	151 97	23 17	FF	151 97	00	23 17	151 97
24 18	DEVICE_ADD	152 98	24 18	00	152 98	00	24 18	152 98
25 19	SLAVE_ADD	153 99	25 19	F1	153 99	00	25 19	153 99
26 1A	MORO_ADD_DISP	154 9A	26 1A	91	154 9A	1A	26 1A	154 9A
27 1B	DEVICE_ADD_DISP	155 9B	27 1B	91	155 9B	00	27 1B	155 9B
28 1C	LOCKER_DISP	156 9C	28 1C	8E	156 9C	00	28 1C	156 9C
29 1D	ADVERTOS	157 9D	29 1D	00	157 9D	00	29 1D	157 9D
30 1E	TINRAL_EPRNOM	158 9E	30 1E	7E	158 9E	00	30 1E	158 9E
31 1F	TINRAL_NC	159 9F	31 1F	91	159 9F	00	31 1F	159 9F
32 20	MORAS_USOL	160 AD	32 20	91	160 AD	00	32 20	160 AD
33 21	MORAS_USOH	161 A1	33 21	4E	161 A1	00	33 21	161 A1
34 22	PROPORCION	162 A2	34 22	00	162 A2	00	34 22	162 A2
35 23	TINR_PROF	163 A3	35 23	80	163 A3	00	35 23	163 A3
36 24	VUELTA31	164 A4	36 24	9F	164 A4	00	36 24	164 A4
37 25	VUELTA32	165 A5	37 25	80	165 A5	00	37 25	165 A5
38 26	PICHA	166 A6	38 26	ED	166 A6	00	38 26	166 A6
39 27	CENTAVOS	167 A7	39 27	00	167 A7	00	39 27	167 A7
40 28	QUINTALES	168 A8	40 28	8E	168 A8	00	40 28	168 A8
41 29	MUM_PALABRA	169 A9	41 29	91	169 A9	00	41 29	169 A9
42 2A	TIEMPO_ACIUM	170 AA	42 2A	91	170 AA	00	42 2A	170 AA
43 2B	STIGUETA	171 AB	43 2B	8E	171 AB	00	43 2B	171 AB
44 2C	TEMP_LAY1	172 AC	44 2C	00	172 AC	00	44 2C	172 AC
45 2D	TEMP_LAY2	173 AD	45 2D	60	173 AD	00	45 2D	173 AD
46 2E	TEMP_PSRM1	174 AE	46 2E	91	174 AE	00	46 2E	174 AE
47 2F	TEMP_PSRM2	175 AF	47 2F	91	175 AF	00	47 2F	175 AF
48 30	TEMP_PSRM3	176 B0	48 30	7E	176 B0	00	48 30	176 B0
49 31	TEMP_PSRM4	177 B1	49 31	00	177 B1	00	49 31	177 B1
50 32	TEMP_PSRM5	178 B2	50 32	7F	178 B2	7E	50 32	178 B2
51 33	TEMP_PSRM6	179 B3	51 33	90	179 B3	81	51 33	179 B3
52 34		180 B4	52 34	80	180 B4	42	52 34	180 B4
53 35		181 B5	53 35	7F	181 B5	00	53 35	181 B5
54 36		182 B6	54 36	00	182 B6	02	54 36	182 B6
55 37		183 B7	55 37	FF	183 B7	15	55 37	183 B7
56 38		184 B8	56 38	91	184 B8	0F	56 38	184 B8
57 39		185 B9	57 39	91	185 B9	00	57 39	185 B9
58 3A		186 BA	58 3A	8E	186 BA	00	58 3A	186 BA
59 3B		187 BB	59 3B	00	187 BB	15	59 3B	187 BB
60 3C		188 BC	60 3C	7E	188 BC	12	60 3C	188 BC
61 3D		189 BD	61 3D	00	189 BD	00	61 3D	189 BD
62 3E		190 BE	62 3E	18	190 BE	18	62 3E	190 BE
63 3F		191 BF	63 3F	01	191 BF	01	63 3F	191 BF
64 40		192 C0	64 40	00	192 C0	00	64 40	192 C0
65 41		193 C1	65 41	FF	193 C1	7E	65 41	193 C1
66 42		194 C2	66 42	81	194 C2	01	66 42	194 C2
67 43		195 C3	67 43	81	195 C3	01	67 43	195 C3
68 44		196 C4	68 44	7E	196 C4	00	68 44	196 C4
69 45		197 C5	69 45	00	197 C5	7E	69 45	197 C5
70 46		198 C6	70 46	FF	198 C6	01	70 46	198 C6
71 47		199 C7	71 47	91	199 C7	01	71 47	199 C7
72 48		200 C8	72 48	91	200 C8	00	72 48	200 C8
73 49		201 C9	73 49	81	201 C9	0E	73 49	201 C9
74 4A		202 CA	74 4A	00	202 CA	15	74 4A	202 CA
75 4B		203 CB	75 4B	FF	203 CB	00	75 4B	203 CB
76 4C		204 CC	76 4C	90	204 CC	00	76 4C	204 CC
77 4D		205 CD	77 4D	90	205 CD	1F	77 4D	205 CD
78 4E		206 CE	78 4E	80	206 CE	10	78 4E	206 CE
79 4F		207 CF	79 4F	00	207 CF	08	79 4F	207 CF
80 50		208 D0	80 50	30 - Volt "Carne"	208 D0	00	80 50	208 D0
81 51	EEPROM_LOCKER	209 D1	81 51	- Volt Dígito 1 Carné	209 D1	0F	81 51	209 D1
82 52	EEPROM_STATUS	210 D2	82 52	- Volt Dígito 2 Carné	210 D2	11	82 52	210 D2
83 53	EEPROM_CARNET1	211 D3	83 53	- Volt Dígito 3 Carné	211 D3	0F	83 53	211 D3
84 54	EEPROM_CARNET2	212 D4	84 54	- Volt Dígito 4 Carné	212 D4	00	84 54	212 D4
85 55	EEPROM_CARNET3	213 D5	85 55	- Volt Dígito 5 Carné	213 D5	12	85 55	213 D5
86 56	EEPROM_MIR	214 D6	86 56	31 - Volt "Castlora"	214 D6	00	86 56	214 D6
87 57	EEPROM_DAY	215 D7	87 57	- Volt Dígito 1 Locker	215 D7	00	87 57	215 D7
88 58		216 D8	88 58	- Volt Dígito 2 Locker	216 D8	00	88 58	216 D8
89 59		217 D9	89 59	32 - Volt "Deposito lo ..."	217 D9	00	89 59	217 D9
90 5A		218 DA	90 5A		218 DA	00	90 5A	218 DA
91 5B		219 DB	91 5B		219 DB	00	91 5B	219 DB
92 5C		220 DC	92 5C		220 DC	00	92 5C	220 DC
93 5D		221 DD	93 5D		221 DD	00	93 5D	221 DD
94 5E		222 DE	94 5E		222 DE	00	94 5E	222 DE
95 5F		223 DF	95 5F		223 DF	00	95 5F	223 DF
96 60		224 E0	96 60		224 E0	00	96 60	224 E0
97 61		225 E1	97 61		225 E1	00	97 61	225 E1
98 62		226 E2	98 62		226 E2	00	98 62	226 E2
99 63		227 E3	99 63		227 E3	00	99 63	227 E3
100 64		228 E4	100 64		228 E4	00	100 64	228 E4
101 65		229 E5	101 65		229 E5	00	101 65	229 E5
102 66		230 E6	102 66		230 E6	00	102 66	230 E6
103 67		231 E7	103 67		231 E7	00	103 67	231 E7
104 68		232 E8	104 68		232 E8	00	104 68	232 E8
105 69		233 E9	105 69		233 E9	00	105 69	233 E9
106 6A		234 EA	106 6A		234 EA	00	106 6A	234 EA
107 6B		235 EB	107 6B		235 EB	00	107 6B	235 EB
108 6C		236 EC	108 6C		236 EC	00	108 6C	236 EC
109 6D		237 ED	109 6D		237 ED	00	109 6D	237 ED
110 6E		238 EE	110 6E		238 EE	00	110 6E	238 EE
111 6F		239 EF	111 6F		239 EF	00	111 6F	239 EF
112 70		240 F0	112 70		240 F0	00	112 70	240 F0
113 71		241 F1	113 71		241 F1	12	113 71	241 F1
114 72		242 F2	114 72		242 F2	00	114 72	242 F2
115 73		243 F3	115 73		243 F3	00	115 73	243 F3
116 74		244 F4	116 74		244 F4	00	116 74	244 F4
117 75		245 F5	117 75		245 F5	00	117 75	245 F5
118 76		246 F6	118 76		246 F6	00	118 76	246 F6
119 77		247 F7	119 77		247 F7	00	119 77	247 F7
120 78		248 F8	120 78		248 F8	00	120 78	248 F8
121 79		249 F9	121 79		249 F9	00	121 79	249 F9
122 7A		250 FA	122 7A		250 FA	00	122 7A	250 FA
123 7B		251 FB	123 7B		251 FB	00	123 7B	251 FB
124 7C		252 FC	124 7C		252 FC	00	124 7C	252 FC
125 7D		253 FD	125 7D		253 FD	00	125 7D	253 FD
126 7E		254 FE	126 7E		254 FE	00	126 7E	254 FE
127 7F		255 FF	127 7F		255 FF	00	127 7F	255 FF

Memoria dentro del ACCESS BANK en sombreado

Q P1 P2 P3 P4





Tabla 9. Mapa de memoria RAM en microcontrolador esclavo

BANCO 0		
0	0	FLAGS
1	1	LED1
2	2	LED2
3	3	LED3
4	4	LED4
5	5	IDENTIDAD
6	6	MAX_LOCKERS
7	7	CONTADOR
8	8	PUERTAS1_DSORD
9	9	PUERTAS2_DSORD
10	A	SOLENOIDES1
11	B	SOLENOIDES2
12	C	TEMP_SSPSTAT
13	D	RX_I2C
14	E	TX_I2C
15	F	TIMER1
16	10	TIMER2
17	11	TIMER3
18	12	
19	13	
20	14	
21	15	
22	16	
23	17	
24	18	
25	19	
26	1A	
27	1B	
28	1C	
29	1D	
30	1E	
31	1F	
32	20	
33	21	
34	22	
35	23	
36	24	
37	25	
38	26	Memoria dentro del ACCESS BANK
39	27	
40	28	
41	29	
42	2A	
43	2B	
44	2C	
45	2D	
46	2E	
47	2F	
48	30	
49	31	
50	32	
51	33	
52	34	
53	35	
54	36	
55	37	
56	38	
57	39	
58	3A	
59	3B	
60	3C	
61	3D	
62	3E	
63	3F	

## VIII. ANEXO B

---

---

Códigos fuente en lenguaje ensamblador



```

;*****
;* UNIVERSIDAD DEL VALLE DE GUATEMALA
;* DEPARTAMENTO DE INGENIERIA ELECTRONICA
;* TRABAJO DE GRADUACION
;* FIRMWARE CIRCUITO MAESTRO - ADMINISTRADOR DE CASILLEROS
;*
;* JUANCARLOS BURMESTER CASTILLO          CARNE 97008
;*****

list p=18f458
#include <p18F458.inc>
ERRORLEVEL -302

CONFIG CONFIG1H, OSCS OFF 1H & ECIO OSC 1H
_CONFIG CONFIG2L, _BOR_ON_2L & _PWRT_ON_2L & _BORV_45_2L
_CONFIG CONFIG2H, _WDT_OFF_2H
CONFIG CONFIG4L, STVR OFF 4L & LVP OFF 4L
CONFIG CONFIG5L, CPO OFF 5L & CP1 OFF 5L & CP2 OFF 5L & CP3 OFF 5L
_CONFIG CONFIG5H, _CPB_OFF_5H & _CPD_OFF_5H
_CONFIG CONFIG6L, WRTO OFF 6L & WRT1 OFF 6L & WRT2 OFF 6L & WRT3 OFF 6L
_CONFIG CONFIG6H, _WRTB_OFF_6H & _WRTC_OFF_6H & _WRTD_OFF_6H
_CONFIG CONFIG7L, _EBTRO_OFF_7L & _EBTR1_OFF_7L & _EBTR2_OFF_7L & _EBTR3_OFF_7L
CONFIG CONFIG7H, EBTRB OFF 7H

;CONSTANTES
;*****
#define ritmo          H'01'
#define centavosxhora D'100'

;VARIABLES
;*****

CBLOCK H'00'
FLAGS0
FLAGS1
LOCKER
CARNET1
CARNET2
CARNET3
MIN
HRS
DAY
PER FLAG
RELOJAZO
COLUMNA
TWO BCD
ONE BCD
CONTADOR
ORIGEN DISP
POS_CARNET
DIGITO_RX
RX I2C
TIMER1
TIMER2
TIMER3
ESTATUS
WORD_ADD
DEVICE_ADD
SLAVE_ADD
WORD_ADD_DISP
DEVICE_ADD_DISP
LOCKER_DISP
ACIERTOS
TREAL_EEPROM
TREAL_PIC
HORAS_USOL

```

```

HORAS USOH
PROPORCION
VUELTAS1
VUELTAS2
FICHA
CENTAVOS
QUETZALES
NUM PALABRA
TIEMPO ACOM
ETIQUETA
TEMP LATA
TEMP LATD
TEMP_FSR0H
TEMP_FSR0L
TEMP_FSR1H
TEMP_FSR1L
TEMP_FSR2H
TEMP_FSR2L
ENDC

CBLOCK H'50'
EEPROM LOCKER
EEPROM_STATUS
EEPROM_CARNET1
EEPROM_CARNET2
EEPROM_CARNET3
EEPROM_MIN
EEPROM_HRS
EEPROM_DAY
ENDC

;*****

;EQUIVALENCIAS
;*****
;en FLAGS0:
#define medio_seg           H'00'
#define un_seg              H'01'
#define carnet              H'02'
#define quinto_malo        H'03'
#define old_user            H'04'
#define uno_disp            H'05'
#define saldo               H'06'

;*****

ORG 00h
GOTO MAIN
ORG 08h
GOTO INTERRUPTO

MAIN
CALL INIT_MEMORY
CALL INIT_PUERTOS
CALL INIT_RELOJ
CALL DELAY
CALL DELAY
CALL DELAY ;espera que permite a los esclavos
CALL DELAY ;configurarse para responder a las
CALL DELAY ;ordenes de inicializacion
CALL DELAY
CALL INIT_ESCLAVOS

LFSR 0, H'1E2'

loop_principal
MOVFF INDF0, LATD ;el dato apuntado por INDF0
MOVWF FSROL, W, 0 ;se coloca en el bus de datos

```

```

ADDLW D'10'
MOVWF FSR0L,0
MOVLW H'10'           ;y se hace una escritura al
MOVWF LATA,0         ;latch A - la primera columna
CALL WAIT
MOVLW H'13'
MOVWF LATA,0
MOVFF INDF0,LATD     ;el dato apuntado por INDF0
MOVF FSR0L,W,0      ;se coloca en el bus de datos
ADDLW D'10'
MOVWF FSR0L,0
MOVLW H'11'         ;y se hace una escritura al
MOVWF LATA,0       ;latch B - la segunda columna
CALL WAIT
MOVLW H'13'
MOVWF LATA,0

MOVFF INDF0,LATD     ;el dato apuntado por INDF0
MOVLW H'12'         ;se escribe al latch C -
MOVWF LATA,0       ;la tercera columna
CALL WAIT
MOVLW H'13'
MOVWF LATA,0

MOVFF COLUMNA,LATA  ;se pone a tierra el conjunto
CALL DELAY         ;de columnas indicado por columna
INCF COLUMNA,F,0   ;para pasar a la siguiente

BTFSS PORTB,06h,0   ;se ha activado el borrado de EEPROM?
CALL FORMAT EEPROM ;si: ejecutar FORMAT EEPROM

BTFSS FLAGS0,medio_seg,0 ;ha transcurrido medio segundo?
BRA no0.5seg       ;no: ir a no0.5seg
BCF FLAGS0,medio_seg,0 ;si: limpiar la bandera y
LFSR 1,H'1F1'      ;borrar los dos puntitos
CLRF INDF1,0

no0.5seg
BTFSS FLAGS0,un_seg,0 ;ha transcurrido un segundo?
BRA nol.0seg       ;no: ir a nol.0seg
BCF FLAGS0,un_seg,0 ;si: limpiar la bandera,
LFSR 1,H'1E7'      ;actualizar los datos de minutos
MOVF HRS,W,0       ;y horas a desplegar y
CALL A_DISPLAY
MOVLW H'12'         ;reestablecer los dos puntitos
MOVWF POSTINC1,0
CLRF POSTINC1,0
MOVF MIN,W,0
CALL A_DISPLAY

nol.0seg
BTFSC FLAGS0,carnet,0
CALL NEW_CARNET

MOVF FSR0L,W,0
ADDLW H'ED'         ;resta 19 al puntero FSRO
MOVWF FSR0L,0
MOVF ORIGEN_DISP,W,0
ADDLW D'10'

CPFSEQ FSR0L,0     ;se barrio ya la pantalla complete?
BRA loop_principal ;no: reiniciar el ciclo

MOVF ORIGEN_DISP,W,0
MOVWF FSR0L,0
CLRF COLUMNA,0    ;inicio del barrido, reset COLUMNA y

BTFSC FLAGS0,saldo,0 ;existe saldo pendiente?
BRA hay_saldo     ;si: saltar a hay_saldo

```

```

BTFSC FLAGS0,old user,0
CALL DESASIGNAR

MOVLW H'01'           ;no: cargar 01 en la parte alta
MOVWF FSR0H,0        ;del puntero
BRA loop_principal   ;reiniciar el ciclo

hay_saldo
MOVF ORIGEN_DISP,W,0 ;se llego ya al final del mensaje?
XORLW H'E2'
BNZ loop_principal   ;no: ir al loop principal

LFSR 1,H'2EA'        ;carga del puntero de busqueda en la tabla
MOVLW H'10'          ;existen decenas en quetzales?
CPFSLT QUETZALES,0
BRA $+8              ;si: saltar
BSF FLAGS0,quinto_malo,0 ;no: indicar que solo existe un BCD en el byte
SWAPF QUETZALES,W,0 ;cargar el valor de las unidades
BRA $+4              ;y saltar al llamado de A_DISPLAY
MOVF QUETZALES,W,0   ;cargar el valor de las unidades
CALL A_DISPLAY
BCF FLAGS0,quinto_malo,0 ;limpiar la bandera
MOVLW H'01'
MOVWF POSTINC1,0     ;colocar el punto decimal
CLRF POSTINC1,0
MOVF CENTAVOS,W,0    ;hacer el despliegue de los centavos
CALL A_DISPLAY

loop_precio
CLRF POSTINC1,0      ;y borrar todos los bytes restantes
MOVF FSR1L,F,0       ;en el despliegue
BNZ loop_precio
MOVLW H'02'          ;cargar 2 a la parte alta del puntero
MOVWF FSR0H,0        ;para mostrar el monto
BRA loop_principal

;*****
INIT MEMORY          ;INICIALIZACION DE MEMORIA
LFSR 0,H'00'         ;Rutina de borrado de los bancos 0, 1, 2 y 3
CLRF POSTINC0,0
BTFSS FSR0H,02h
BRA $-4

LFSR 0,H'100'        ;Carga del equivalente de cada caracter

MOVLW H'7E'          ;0
MOVWF POSTINC0,0
MOVLW H'81'
MOVWF POSTINC0,0
MOVLW H'81'
MOVWF POSTINC0,0
MOVLW H'7E'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'01'          ;1
MOVWF POSTINC0,0
MOVLW H'41'
MOVWF POSTINC0,0
MOVLW H'FF'
MOVWF POSTINC0,0
MOVLW H'01'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'47'          ;2
MOVWF POSTINC0,0
MOVLW H'89'
MOVWF POSTINC0,0

```

```

MOVLW H'91'
MOVWF POSTINC0,0
MOVLW H'61'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'42' ;3
MOVWF POSTINC0,0
MOVLW H'81'
MOVWF POSTINC0,0
MOVLW H'91'
MOVWF POSTINC0,0
MOVLW H'6E'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'70' ;4
MOVWF POSTINC0,0
MOVLW H'10'
MOVWF POSTINC0,0
MOVLW H'10'
MOVWF POSTINC0,0
MOVLW H'FF'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'F1' ;5
MOVWF POSTINC0,0
MOVLW H'91'
MOVWF POSTINC0,0
MOVLW H'91'
MOVWF POSTINC0,0
MOVLW H'8E'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'7E' ;6
MOVWF POSTINC0,0
MOVLW H'91'
MOVWF POSTINC0,0
MOVLW H'911
MOVWF POSTINC0,0
MOVLW H'4E'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'80' ;7
MOVWF POSTINC0,0
MOVLW H'8F'
MOVWF POSTINC0,0
MOVLW H'90'
MOVWF POSTINC0,0
MOVLW H'EO'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'6E' ;8
MOVWF POSTINC0,0
MOVLW H'91'
MOVWF POSTINC0,0
MOVLW H'91'
MOVWF POSTINC0,0
MOVLW H'6E'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'60' ;9
MOVWF POSTINC0,0

```

```

MOVLW H'91'
MOVWF POSTINC0,0
MOVLW H'91'
MOVWF POSTINC0,0
MOVLW H'7E'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'7F' ;A
MOVWF POSTINC0,0
MOVLW H'90'
MOVWF POSTINC0,0
MOVLW H'90'
MOVWF POSTINC0,0
MOVLW H'7F'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'FF' ;B
MOVWF POSTINC0,0
MOVLW H'91'
MOVWF POSTINC0,0
MOVLW H'91'
MOVWF POSTINC0,0
MOVLW H'6E'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'7E' ;C
MOVWF POSTINC0,0
MOVLW H'81'
MOVWF POSTINC0,0
MOVLW H'81'
MOVWF POSTINC0,0
MOVLW H'42'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'FF' ;D
MOVWF POSTINC0,0
MOVLW H'81'
MOVWF POSTINC0,0
MOVLW H'81'
MOVWF POSTINC0,0
MOVLW H'7E'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'FF' ;E
MOVWF POSTINC0,0
MOVLW H'91'
MOVWF POSTINC0,0
MOVLW H'91'
MOVWF POSTINC0,0
MOVLW H'81'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'FF' ;F
MOVWF POSTINC0,0
MOVLW H'90'
MOVWF POSTINC0,0
MOVLW H'90'
MOVWF POSTINC0,0
MOVLW H'80'
MOVWF POSTINC0,0
CLRF POSTINC0,0

```

```

LFSR 0,H'150'
MOVLW H'30' ;colocacion mensaje "Carne"
MOVWF INDF0,0
LFSR 0,H'156'
MOVLW H'31' ;colocacion mensaje "Casillero"
MOVWF INDF0,0
LFSR 0,H'159'
MOVLW H'32' ;colocacion mensaje "Deposite lo indicado"
MOVWF INDF0, 0

LFSR 0,H'180'

MOVLW H'7E' ;C
MOVWF POSTINC0,0
MOVLW H'81'
MOVWF POSTINC0,0
MOVLW H'42'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'02' ;a
MOVWF POSTINC0,0
MOVLW H'15'
MOVWF POSTINC0,0
MOVLW H'0F'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'1F' ;r
MOVWF POSTINC0,0
MOVLW H'10'
MOVWF POSTINC0,0
MOVLW H'08'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'1F' ;n
MOVWF POSTINC0,0
MOVLW H'10'
MOVWF POSTINC0,0
MOVLW H'0F'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'0E' ;e
MOVWF POSTINC0,0
MOVLW H'55'
MOVWF POSTINC0,0
MOVLW H'4D'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'12' ;:
MOVWF POSTINC0,0
CLRF POSTINC0,0
CLRF POSTINC0,0

LFSR 0,H'1B0'

CLRF POSTINC0,0
CLRF POSTINC0,0
MOVLW H'7E' ;C
MOVWF POSTINC0, 0
MOVLW H'81'
MOVWF POSTINC0,0
MOVLW H'42'
MOVWF POSTINC0,0
CLRF POSTINC0,0

```

```

MOVLW H'02'           ;a
MOVWF POSTINC0,0
MOVLW H'151
MOVWF POSTINC0,0
MOVLW H'0F1
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'09'           ;s
MOVWF POSTINC0,0
MOVLW H'15'
MOVWF POSTINC0,0
MOVLW H'12'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'16'           ;i
MOVWF POSTINC0,0
MOVLW H'01'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'7E'           ;l
MOVWF POSTINC0,0
MOVLW H'01'
MOVWF POSTINC0,0
MOVLW H'01'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'7E'           ;l
MOVWF POSTINC0,0
MOVLW H'01'
MOVWF POSTINC0,0
MOVLW H'01'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'0E'           ;e
MOVWF POSTINC0,0
MOVLW H'15'
MOVWF POSTINC0,0
MOVLW H'0D'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'1F'           ;r
MOVWF POSTINC0,0
MOVLW H'10'
MOVWF POSTINC0,0
MOVLW H'08'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'0E'           ;o
MOVWF POSTINC0,0
MOVLW H'11'
MOVWF POSTINC0,0
MOVLW H'0E'
MOVWF POSTINC0,0
CLRF POSTINC0,0

MOVLW H'12'           ;:
MOVWF POSTINC0,0
CLRF POSTINC0,0
CLRF POSTINC0,0

```

```

LFSR 0,H'1E2'
CLRF POSTINC0,0
CLRF POSTINC0,0
CLRF POSTINC0,0
CLRF POSTINC0,0
CLRF POSTINC0,0

LFSR 0,H'1F1'
MOVLW H'12'
MOVWF POSTINC0,0
CLRF POSTINC0,0

LFSR 0,H'2E4'
MOVLW H'7E'           ;Q
MOVWF POSTINC0,0
MOVLW H'85'
MOVWF POSTINC0,0
MOVLW H'83'
MOVWF POSTINC0,0
MOVLW H'7F'
MOVWF POSTINC0,0
CLRF POSTINC0,0

;VALORES INICIALES PARA CIERTAS VARIABLES

MOVLW H'E2'
MOVWF ORIGEN_DISP,0
MOVLW H'03'
MOVWF POS_CARNET, 0

loop proporcion
INCF CONTADOR,F,0
MOVF PROPORCION,W,           ;0
ADDLW D'25'
MOVWF PROPORCION, 0
XORLW centavosxhora
BNZ loop proporcion
MOVWF CONTADOR,PROPORCION

RETURN

;-----
INIT PUERTOS                ;INICIALIZACION DE PUERTOS
CLRF LATA,0                 ;valores iniciales en las
CLRF LATB,0                 ;salidas de los puertos
MOVLW B'00100000'
MOVWF LATC,0
CLRF LATD, 0
MOVLW B'00000011'
MOVWF LATE,0
CLRF TRISA,0                ;bus de direcciones (todas salidas)
MOVLW B'01111111'          ;entradas de los interruptos externos
MOVWF TRISB,0               ;e indicador de las monedas
MOVLW B'10011111'          ;bus i2c y comunicacion serial
MOVWF TRISC,0
CLRF TRISD,0                ;bus de datos (todas salidas)
MOVLW B'00000100'          ;control sobre escritura y lectura
MOVWF TRISE,0               ;del reloj
MOVLW B'00000110'          ;todas los pines del puerto A como
MOVWF ADCON1,0              ;entradas/salidas digitales
MOVLW B'00000111'          ;deshabilitacion del modulo de comparadores
MOVWF CMCON,0

;DE I2C
MOVLW B'00101000'          ;habilitacion del puerto serial sincrono,
MOVWF SSPCON1,0             ;I2C master mode
MOVLW B'10000000'          ;slew rate deshabilitado
MOVWF SSPSTAT,0

```

```

MOVW D'99' ;velocidad de 50kbps
MOVWF SSPADD,0

;DEL TIMER
MOVW B'00000111' ;timer off, 16 bit, internal inst.,
MOVWF TOCON,0 ;low to high, prescaler 1:256
CLRF WREG,0
MOVWF TMR0H,0 ;contadores en 0
MOVWF TMR0L,0

;DE INTERRUPTOS
BCF RCON,IPEN,0 ;deshabilitacion de prioridades
BCF INTCON3,INT1IF,0 ;limpieza de bandera int de reloj
BCF INTCON2,INTEDG1,0 ;en flanco negativo
BSF INTCON3,INT1IE,0 ;habilitacion de int de reloj

BCF INTCON3,INT2IF,0 ;limpieza de bandera int de fichas

BCF INTCON,INT0IF,0 ;limpieza de bandera int de voz
BSF INTCON2,INTEDGO,0 ;en flanco positivo

BCF TXSTA,SYNC,0 ;modo asincrono
BSF TXSTA,BRGH,0 ;high speed baud rate
MOVW D'129' ;9600 bps
MOVWF SPBRG,0 ;recepcion continue
BSF RCSTA,CREN,0 ;habilitacion del puerto serial
BCF RCSTA,SPEN,0 ;limpieza de bandera int de barras
BSF PIR1,RCIF,0 ;habilitacion de int de barras
BSF PIE1,RCIE,0

BCF INTCON,TMR0IF,0 ;limpieza de bandera int timer0
BSF INTCON,TMR0IE,0 ;habilitacion de int timer0

BSF INTCON,PEIE,0 ;habilitacion de ints perifericos
BSF INTCON,GIE,0 ;habilitacion global de interruptos
RETURN

;-----
INIT_ESCLAVOS ;RUTINA DE COMUNICACION INICIAL
MOVW H'20' ;CON LOS CIRCUITOS ESCLAVOS
MOVWF SLAVE_ADD,0 ;empezar con esclavo H'20' = A
ciclo_max_lockers
BSF SSPCON2,SEN,0 ;generar Start
CALL waitMSSP
INCF SLAVE_ADD,W,0 ;indica lectura I2C
MOVWF SSPBUF,0
CALL waitMSSP
BTFSC SSPCON2,ACKSTAT,0 ;respondio el esclavo al llamado?
BRA no_existe ;no: esclavo no existe
BSF SSPCON2,RCEN,0 ;si: recibir del esclavo el numero
CALL waitMSSP ;de locker que opera
MOVWF SSPBUF,RX_I2C
BSF SSPCON2,ACKDT,0
BSF SSPCON2,ACKEN,0 ;emitir ACK de recepcion del dato
CALL waitMSSP
BSF SSPCON2,PEN,0 ;generar stop
CALL waitMSSP
INCF RX_I2C,F,0 ;a partir del siguiente locker
BZ esclavo_completo ;se han de declarar todos ocupados
BRA ciclo_ocupados ;para no disponer de ellos
no_existe
BSF SSPCON2,PEN,0 ;generar stop
CALL waitMSSP ;y no disponer de los casilleros a
CLRF RX_I2C,0 ;partir del locker 0
ciclo_ocupados
CALL WAIT
CALL GET_ADDRESSES ;obtencion de las direcciones en EEPROM
BSF SSPCON2,RSEN,0 ;generar Restart

```

```

CALL waitMSSP
MOVFF DEVICE_ADD,SSPBUF ;escribir direccion del chip EEPROM
CALL waitMSSP
BTFSC SSPCON2,ACKSTAT,0 ;esta lista la memoria?
GOTO ciclo_ocupados ;no: intentar de nuevo
MOVFF WORD_ADD,SSPBUF ;si: escribir la palabra a modificar
CALL waitMSSP
BTFSC SSPCON2,ACKSTAT,0
GOTO I2Cfail
MOVLW H'01' ;escribir H'01' (indicacion de ocupado)
MOVWF SSPBUF, 0
CALL waitMSSP
BTFSC SSPCON2,ACKSTAT,0
GOTO I2Cfail
BSF SSPCON2,PEN,0 ;generar stoP
CALL waitMSSP
INCF RX_I2C,F, 0
MOVLW B'00001111'
ANDWF RX_I2C,W,0 ;se hizo ya hasta el locker F?
BNZ ciclo_ocupados ;no: repetir para siguiente locker esclavo complete
CLRf RZ_I2C,0 ;ahora se procede a informar al esclavo
ciclo status EEPROM ;el status de cada locker segun la EEPROM
CALL GET_ADDRESSES ;obtencion de direcciones
BSF SSPCON2,RSEN,0 ;generar Restart
CALL waitMSSP
MOVFF DEVICE_ADD,SSPBUF ;direccion de la memoria EEPROM
CALL waitMSSP ;para consultar el estado del locker
BTFSC SSPCON2,ACKSTAT,0 ;esta lista la memoria?
BRA ciclo_status_EEPROM ;no: intentar de nuevo
MOVFF WORD_ADD,SSPBUF ;si: escribir la palabra a consultar
CALL waitMSSP
BTFSC SSPCON2,ACKSTAT,0
GOTO I2Cfail
BSF SSPCON2,RSEN,0 ;generar un Restart
CALL waitMSSP
INCF DEVICE_ADD,W,0 ;indicar el comando de lectura
MOVWF SSPBUF, 0
CALL waitMSSP
BTFSC SSPCON2,ACKSTAT,0
GOTO I2Cfail
BSF SSPCON2,RCEN,0 ;pasar a modo receptor
CALL waitMSSP
MOVFF SSPBUF,ESTATUS ;el valor de la consulta se guarda
BSF SSPCON2,ACKDT,0 ;en ESTATUS
BSF SSPCON2,ACKEN,0
CALL waitMSSP
BSF SSPCON2,PEN,0 ;generar stoP
CALL waitMSSP

BSF SSPCON2,SEN,0 ;generar un Start
CALL waitMSSP
MOVFF SLAVE_ADD,SSPBUF ;para hablar con el esclavo
CALL waitMSSP
BTFSC SSPCON2,ACKSTAT,0 ;si se estuviera informando a un esclavo
BRA $+H'10' ;no existente salta a la sig. consulta
MOVFF ESTATUS,SSPBUF ;transmitir ESTATUS
CALL waitMSSP
BTFSC SSPCON2,ACKSTAT,0
GOTO I2Cfail
BSF SSPCON2,PEN,0 ;generar un stop
CALL waitMSSP
INCF RX_I2C,F,0 ;se informo ya sobre los 16 casilleros?
MOVLW B'00001111'
ANDWF RX_I2C,W,0
BNZ ciclo_status_EEPROM ;no: repetir el ciclo para el sig. locker

MOVLW H'10' ;sumar 10 para hablarle al siguiente
ADDWF SLAVE_ADD, F, 0 ;esclavo

```

```

MOVLW H'80'
CPFSEQ SLAVE ADD,0           ;se hizo ya para los 6 posibles esclavos?
BRA ciclo_max_lockers      ;no: repita el ciclo complete

```

```

RETURN

```

```

;-----
INIT RELOJ                    ;INICIALIZACION DEL RTC

    SETF TRISD,0              ;bus de datos como entrada

    MOVLW H'43'
    MOVWF LATA,0              ;periodic flag reg
    CALL WAIT
    BCF LATE,00h,0           ;read!
    CALL WAIT
    MOVFF PORTD,PER FLAG
    BSF LATE,00h,0

    CLRF TRISD,0              ;bus de datos como salida

    BTSS PER FLAG,H'06'      ;se perdio info de la hora?
    BRA start_clock          ;no: inicie el reloj

    MOVLW B'01001111'        ;si: configurar nuevamente...
    MOVWF LATD,0              ;banco de registro=1 y reset de
    MOVLW H'60'               ;banderas de interrupcion
    MOVWF LATA,0              ;main status reg
    CALL WAIT
    BCF LATE,01h,0           write!
    CALL WAIT
    BSF LATE,01h,0

    MOVLW B'00000001'        ;deshabilitacion de interruptos
    MOVWF LATD,0              ;en bateria, stop clock, modo 24H,
    MOVLW H'61'               ;anno biciesto hace un anno
    MOVWF LATA,0              ;real time reg
    CALL WAIT
    BCF LATE,01h,0           ;write!
    CALL WAIT
    BSF LATE,01h,0

    CLRF LATD,0               ;deshabilitacion de interruptos
    MOVLW H'64'               ;de alarma
    MOVWF LATA,0              ;int Ctrl reg 1
    CALL WAIT
    BCF LATE,01h,0           ;write!
    CALL WAIT
    BSF LATE,01h,0

    MOVLW B'00001111'        ;banco de registros = 0
    MOVWF LATD,0
    MOVLW H'60'
    MOVWF LATA,0              ;main status reg
    CALL WAIT
    BCF LATE,01h,0           ;write!
    CALL WAIT
    BSF LATE,01h,0

    CLRF LATD,0               ;funcionamiento con bateria
    MOVLW H'43'
    MOVWF LATA,0              ;periodic flag reg
    CALL WAIT
    BCF LATE,01h             ;write!
    CALL WAIT
    BSF LATE,01h

    CLRF LATD,0               ;00 BCD
    MOVLW H'67'

```

```

MOVWF LATA,0           ;min counter
CALL WAIT
BCF LATE,01h, 0       ;write!
CALL WAIT
BSF LATE,01h,0

MOVLW B'00010010'     ;12 BCD
MOVWF LATD,0
MOVLW H'68'
MOVWF LATA,0         ;hrs counter
CALL WAIT
BCF LATE,01h, 0       ;write!
CALL WAIT
BSF LATE,01h,0

MOVLW B'00000001'     ;01 BCD
MOVWF LATD,0
MOVLW H'6E'
MOVWF LATA, 0        ;day of week counter
CALL WAIT
BCF LATE,01h,0       ;write!
CALL WAIT
BSF LATE,01h,0

MOVLW B'00000001'     ;01 BCD
MOVWF LATD,0
MOVLW H'6A'
MOVWF LATA,0         ;month counter
CALL WAIT
BCF LATE,01h,0       ;write!
CALL WAIT
BSF LATE,01h,0

MOVLW B'00000101'     ;05 BCD
MOVWF LATD,0
MOVLW H'6B'
MOVWF LATA,0         ;year counter
CALL WAIT
BCF LATE,01h,0       ;write!
CALL WAIT
BSF LATE, 01h, 0

start clock
MOVLW B'01001111'     ;banco de registros = 1
MOVWF LATD, 0
MOVLW H'60'
MOVWF LATA,0         ;main status reg
CALL WAIT
BCF LATE,01h,0       ;write!
CALL WAIT
BSF LATE,01h,0

MOVLW B'00001000'     ;habilitacion de interrupto cada
MOVWF LATD,0         ;100 mseg
MOVLW H'63'
MOVWF LATA,0         ;int Ctrl reg 0
CALL WAIT
BCF LATE,01h,0       ;write!
CALL WAIT
BSF LATE,01h,0

MOVLW B'00001001'     ;start clock
MOVWF LATD,0
MOVLW H'61'
MOVWF LATA,0         ;real time reg
CALL WAIT
BCF LATE,01h,0       ;write!
CALL WAIT

```

```

BSF LATE,01h,0

RETURN

;-----
A_DISPLAY                                ;RUTINA ENCARGADA DE DETERMINAR
                                           ;EQUIVALENCIAS ENTRE BCD Y DESPLIEGUE
LFSR 2,H'100'                             ;direccion original de la tabla
MOVWF TWO BCD,0                            ;almacenamiento de los BCD a desplegar
ANDLW H'F0'
MOVWF ONE BCD, 0                          ;el BCD mas significative
SWAPF ONE BCD,W,0
MOVLW D'5'                                 ;multiplicado por 5
MOVF PRODL,W,0                             ;para determinar el desplazamiento
ADDWF FSR2L,F,0                            ;en la tabla de equivalencias
MOVLW D'5'
MOVWF CONTADOR,0
MOVFF POSTINC2,POSTINC1                    ;copiado de la secuencia correspondiente
DECFSZ CONTADOR,F,0                        ;5 veces
BRA $-6
BTFSC FLAGS0,quinto malo,0                ;hay solo un BCD en el byte?
RETURN                                     ;si: retorne
LFSR 2,H'100'                             ;direccion original de la tabla
MOVF TWO BCD,W,0
ANDLW H'0F'                                ;obtencion del BCD menos significative
MULLW D'5'                                 ;multiplicado por 5
MOVF PRODL,W,0                             ;para determinar el desplazamiento
ADDWF FSR2L,F,0                            ;en la tabla de equivalencias
MOVLW D'5'
MOVWF CONTADOR,0
MOVFF POSTINC2,POSTINC1                    ;copiado de la secuencia correspondiente
DECFSZ CONTADOR,F,0                        ;5 veces
BRA $-6
RETURN

;-----
NEW_CARNET                                ;RUTINA DE IDENTIFICACION DE GARNET
BCF FLAGS0,carnet,0                        ;borra la bandera
CLRF DIGITO_RX,0                           ;reset al contador de digitos recibidos
MOVLW H'03'                                 ;reset del posicionador de digitos recibidos
MOVWF POS_CARNET,0

CALL USER_TYPE                             ;determinar si es usuario nuevo o antike

BTFSC FLAGS0,old_user,0                    ;es un usuario nuevo?
BRA registrado                             ;no: saltar a registrado
BTFSS FLAGS0,uno disp,0                    ;si: hay un locker disponible?
BRA no_hay                                 ;no: saltar a no_hay
CALL ASIGN LOCKER                           ;si: asignarlo

registrado
BTFSC FLAGS0,old_user,0                    ;es un usuario antiguo?
CALL CALC COBRO                             ;si: calcular el cobro a realizar

CALL MENSAJES_VOZ

LFSR 1,H'197'
MOVF CARNET1,W,0                           ;conversion de BCD a digitos desplegables
CALL A_DISPLAY
MOVF CARNET2,W,0
CALL A_DISPLAY
BSF FLAGS0,quinto malo,0                   ;identificacion de la presencia de
MOVF CARNET3,W,0                           ;un solo BCD en el byte
CALL A_DISPLAY
BCF FLAGS0,quinto malo,0

LFSR 1,H'1D8'
MOVF LOCKER,W,0                             ;conversion del locker a par desplegable

```

```

CALL A DISPLAY

MOVW H'62' ;colocacion del inicio del barrido
MOVWF ORIGEN_DISP,0 ;para desplegar el mensaje completo
no_hay
CLRF CARNET1,0 ;borrado de las variables de
CLRF CARNET2,0 ;almacenamiento temporal
CLRF CARNET3,0
RETURN

;-----
USER TYPE ;RUTINA DE IDENTIFICACION DEL TIPO DE USUARIO
CALL LOCKER_NO_USADO? ;determiner si hubo un locker no usado
CLRF WORD_ADD,0 ;inicializacion de variables para la
MOVW H'AO' ;busqueda en la EEPROM via I2C
MOVWF DEVICE_ADD,0
BCF FLAGS0,uno disp,0
I2Cread
LFSR 2,H'050' ;puntero para la colocacion de bytes leidos
MOVW D'8' ;para realizar la lectura 8 veces
MOVWF CONTADOR,0 ;Restart para iniciar lectura
BSF SSPCON2,RSEN,0 ;envio de la direccion de la EEPROM
CALL waitMSSP
MOVWF DEVICE_ADD,SSPBUF ;envio de la direccion a partir de la cual
CALL waitMSSP ;se empieza a leer
BTFSC SSPCON2,ACKSTAT,0
BRA I2Cread
MOVWF WORD_ADD,SSPBUF ;palabra de control para la lectura
CALL waitMSSP
BTFSC SSPCON2,ACKSTAT,0
GOTO I2Cfail
BSF SSPCON2,RSEN,0
CALL waitMSSP
MOVWF DEVICE_ADD,W,0 ;palabra de control para la lectura
ADDLW H'O1'
MOVWF SSPBUF,0
CALL waitMSSP
BTFSC SSPCON2,ACKSTAT,0
GOTO I2Cfail
loop read
BSF SSPCON2,RCEN,0 ;habilitar el modo receptor
CALL waitMSSP
MOVWF SSPBUF,POSTINC2 ;colocacion del byte leído

BSF SSPCON2,ACKDT,0 ;habilitar NACK
DECFSZ CONTADOR,F,0 ;pero si se pretende seguir leyendo
BCF SSPCON2,ACKDT,0 ;se convierte en ACK
BSF SSPCON2,ACKEN,0 ;envio del acknowledge correspondiente
CALL waitMSSP
BTFSS SSPCON2,ACKDT,0
BRA loop_read

BSF SSPCON2,PEN,0 ;enviar un stop
CALL waitMSSP
BTFSC EEPROM_STATUS,00h,0 ;esta ocupado el locker en este registro?
BRA ocupado ;si: vaya a ocupado
BTFSC FLAGS0,uno_disp,0 ;no: se almaceno ya un locker desocupado?
BRA sig_registro ;si: continúe con el siguiente registro
MOVWF EEPROM_LOCKER,LOCKER_DISP ;no: guarde el locker como uno disponible con
MOVWF WORD_ADD,WORD_ADD_DISP ;su direccion de palabra y
MOVWF DEVICE_ADD,DEVICE_ADD_DISP ;su direccion de dispositivo
BSF FLAGS0,uno_disp,0 ;...un locker disponible ha sido encontrado
BRA sig_registro ;continúe con el siguiente registro
ocupado
CLRF ACIERTOS,0 ;numero de aciertos = 0
MOVWF CARNET1,W,0 ;compare cada byte del carnet ingresado
XORWF EEPROM_CARNET1,W,0 ;con los leídos de la EEPROM, por cada
BTFSC STATUS,Z,0 ;acierto incremente la variable en uno

```

```

INCF ACIERTOS,F,0
MOVWF CARNET2,W,0
XORWF EEPROM CARNET2,W,0
BTFSC STATUS,Z,0
INCF ACIERTOS,F,0
MOVWF CARNET3,W,0
XORWF EEPROM CARNET3,W,0
BTFSC STATUS,Z,0
INCF ACIERTOS,F,0
MOVLW D'3'
CPFSEQ ACIERTOS,0 ;se produjeron 3 aciertos?
BRA sig_registro ;no: continúe con el siguiente registro
BSF FLAGS0,old_user,0 ;si: identificar como usuario antiguo
RETURN
sig_registro
MOVWF WORD_ADD,W,0 ;sumar 8 a la palabra para leer el
ADDLW D'8' ;siguiente renglon
MOVWF WORD_ADD,0
BTFSS STATUS,C,0
BRA $+8
MOVWF DEVICE_ADD,W,0 ;si se produjo un carry a partir de la
ADDLW D'2" ;suma, incrementar en dos la direccion del
MOVWF DEVICE_ADD,0 ;dispositivo
MOVLW H'FF'
CPFSEQ EEPROM_LOCKER,0 ;se leyo ya el ultimo registro?
BRA I2Cread ;no: haga una nueva lectura
RETURN

;-----
LOCKER NO USADO? ;RUTINA DE CONSULTA CON EL ESCLAVO
MOVLW H'20' ;SOBRE UN LOCKER QUE NO HAYA SIDO USADO
MOVWF SLAVE_ADD,0 ;empezando por el esclavo A
ciclo_no_usados
BSF SSPCON2,RSEN,0 ;generar Restart
CALL waitMSSP
INCF SLAVE_ADD,W,0 ;comando de lectura
MOVWF SSPBUF,0
CALL waitMSSP
BTFSC SSPCON2,ACKSTAT,0 ;respondio el esclavo?
BRA no_respondio ;no: saltar a no respondio
BSF SSPCON2,RCEN,0 ;si: pasar a modo receptivo
CALL waitMSSP
MOVWF SSPBUF,RX_I2C ;almacenar el dato recibido BSF SSPCON2,ACKDT,0
BSF SSPCON2,ACKEN,0 ;emitir ACK por la recepcion del dato CALL waitMSSP
BSF SSPCON2,PEN,0 ;generar stoP
CALL waitMSSP
MOVWF RX_I2C,F,0 ;se recibio un cero?
BZ no_reusar ;si: no hubo locker que ho haya sido usado
si_reusar ;no: proceder a borrar el registro
MOVLW B'00001111' ;previamente almacenado para este locker
ANDWF RX_I2C,F,0

CALL GET_ADDRESSES ;obtencion de las direcciones en EEPROM

BSF SSPCON2,SEN,0 ;generar Start
CALL waitMSSP
MOVWF DEVICE_ADD,SSPBUF ;hablar a EEPROM
CALL waitMSSP
BTFSC SSPCON2,ACKSTAT,0
GOTO I2Cfail
MOVWF WORD_ADD,SSPBUF ;direccion de la palabra a modificar
CALL waitMSSP
BTFSC SSPCON2,ACKSTAT,0
GOTO I2Cfail
CLRF WREG,0 ;escribir un cero (indicacion de vacancia)
MOVWF SSPBUF,0
CALL waitMSSP
BTFSC SSPCON2,ACKSTAT,0

```

```

    GOTO I2Cfail
no_respondio
    BSF SSPCON2,PEN,0          ;generar un stoP
    CALL waitMSSP
no_reusar
    MOVLW H'10'                ;sumar para consultar con el sig. esclavo
    ADDWF SLAVE_ADD,F,0
    MOVLW H'80'
    CPFSEQ SLAVE_ADD,0        ;se hizo ya para los 6 posibles?
    BRA ciclo_no_usados      ;no: repetir ciclo de consulta
    RETURN

;-----
GET_ADDRESSES                ;RUTINA DE OBTENCION DE LA DIRECCION
    MOVLW H'20'                ;EN EEPROM CORRESPONDIENTE AL ESCLAVO
    SUBWF SLAVE_ADD,W,0        ;Y LOCKER QUE SE VA A MODIFICAR
    ANDLW B'01100000'
    SWAPF WREG, F, 0
    ADDLW H'A0'
    MOVWF DEVICE_ADD,0        ;para la direccion del dispositivo
    MOVF SLAVE_ADD,W,0
    ANDLW B'00010000'
    ADDWF RX_I2C,W,0
    RLNCF WREG, F, 0
    RLNCF WREG, F, 0
    RLNCF WREG,F,0
    INCF WREG, F, 0
    MOVWF WORD_ADD,0          ;para la ubicacion de la palabra
    RETURN

;-----
ASIGN_LOCKER                 ;RUTINA DE ASIGNACION DE UN LOCKER
    INCF WORD_ADD_DISP,F,0    ;apuntar al status del renglon del locker disp
    LFSR 2,H'03'              ;puntero inicio para la carga de los datos
    MOVLW D'6'                ;la escritura se realizara 6 veces
    MOVWF CONTADOR,0

    BSF SSPCON2,SEN,0        ;envio de un Start
    CALL waitMSSP
    MOVFF DEVICE_ADD_DISP,SSPBUF ;envio de la direccion de dispositivo
    CALL waitMSSP
    BTFSC SSPCON2,ACKSTAT,0
    GOTO I2Cfail
    MOVFF WORD_ADD_DISP,SSPBUF ;envio de la direccion de palabra
    CALL waitMSSP
    BTFSC SSPCON2,ACKSTAT,0
    GOTO I2Cfail
    MOVLW H'01'                ;indicacion que el locker ahora esta
    MOVWF SSPBUF,0            ;ocupado
    CALL waitMSSP
    BTFSC SSPCON2,ACKSTAT,0
    GOTO I2Cfail
I2Cwrite
    MOVFF POSTINC2,SSPBUF      ;envio del siguiente byte a escribir
    CALL waitMSSP
    BTFSC SSPCON2,ACKSTAT,0
    GOTO I2Cfail
    DECFSZ CONTADOR,F,0        ;se escribio ya el registro completo?
    BRA I2Cwrite              ;no: haga una nueva escritura
    BSF SSPCON2,PEN, 0        ;si: envíe un stoP
    CALL waitMSSP
    MOVFF LOCKER_DISP,LOCKER  ;el locker disponible pasa a ser el asignado
    RETURN

;-----
waitMSSP                      ;RUTINA DE ESPERA PARA EL MODULO MSSP
    BTFSS PIR1,SSPIF,0
    BRA $-2

```

```

BCF PIR1,SSPIF,0
RETLW 0

;-----
I2Cfail                                     ;TRAMPA POR OCURRENCIA DE ERROR I2C
    BSF SSPCON2,PEN,0
    CALL waitMSSP
;    BSF LATB,07h,0
    GOTO $

;-----
COMM SLAVE                                  ;RUTINA DE COMUNICACION CON EL ESCLAVO
    BSF SSPCON2,SEN,0                       ;generar un Start
    CALL waitMSSP
    MOVF LOCKER,W,0                         ;el nibble alto de LOCKER indica el
    ANDLW B'01110000'                       ;esclavo con quien se tiene que hablar
    MOVWF SSPBUF,0
    CALL waitMSSP
    BTFSC SSPCON2,ACKSTAT,0
    GOTO I2Cfail
    MOVF LOCKER,W,0                         ;el nibble bajo de LOCKER indica el
    ANDLW B'00001111'                       ;casillero que se quiere abrir
    MOVWF SSPBUF, 0
    CALL waitMSSP
    BTFSC SSPCON2,ACKSTAT,0
    GOTO I2Cfail
    BSF SSPCON2,PEN,0                       ;generar un stop
    CALL waitMSSP
    RETURN

;
CALC_COBRO                                  ;RUTINA DE CALCULO DEL COBRO A REALIZAR
    MOVFF EEPROM_LOCKER, LOCKER             ;el locker leído pasa a ser el asignado

    CLRF HORAS_USOL,0                       ;inicializacion de variables
    CLRF HORAS_USOH,0
    MOVF EEPROM_DAY,W,0                     ;al registro de dia leído
    SUBWF DAY,W,0                           ;reste el dia actual
    MOVLW D'24'                             ;y multiplique la resta por 24
    MOVF PRODL,W,0                          ;para obtener las horas uso por dias
    ADDWF HORAS_USOL,F,0                    ;sume el resultado a la variable

    SWAPF HRS,W,0                           ;obtencion de la equivalencia entre el
    ANDLW B'00000011'                       ;valor BCD y un decimal para poder
    MULLW D'10'                             ;hacer las restas que resultan en
    MOVFF PRODL,TREAL_PIC                   ;las horas uso
    MOVF HRS,W,0
    ANDLW B'00001111'
    ADDWF TREAL_PIC, F, 0

    SWAPF EEPROM_HRS,W,0
    ANDLW B'00000011'
    MOVLW D'10'
    MOVFF PRODL,TREAL_EEPROM
    MOVF EE_PROM_HRS, W, 0
    ANDLW B'00001111'
    ADDWF TREAL_EEPROM,W,0
    SUBWF TREAL_PIC,W,0
    ADDWF HORAS_USOL,F,0

    SWAPF MIN,W,0                           ;obtencion de la equivalencia para
    ANDLW B'00001111'                       ;los registros de minutos
    MOVLW D'10'
    MOVFF PRODL,TREAL_PIC
    MOVF MIN,W,0
    ANDLW B'00001111'
    ADDWF TREAL_PIC,F,0

```

```

SWAPF EEPROM_MIN,W,0
ANDLW B'00001111'
MULLW D'10'
MOVFF PRODL,TREAL_EEPROM
MOVF EEPROM_MIN,W,0
ANDLW B'00001111'
ADDWF TREAL_EEPROM,W,0
SUBWF TREAL_PIC,W,0
BZ $+6
BTFSS STATUS,N,0
INCF HORAS_USOL,F,0

MOVF HORAS_USOL,F,0           ;en el caso en que ni un minuto haya
BTFSC STATUS,Z,0           ;transcurrido se toma ya como utilizado
INCF HORAS_USOL,F,0           ;una hora

MOVLW D'168'               ;se utilice el locker por mas de 168 horas,
CPFSLT HORAS_USOL,0         ;(una semana)?
MOVWF HORAS_USOL,0         ;cargue ese numero como maximo

MOVF PROPORCION,W,0         ;dependiendo de la tarifa se saca una
MULWF HORAS_USOL,0         ;equivalencia en numero de unidades de
MOVFF PRODL,HORAS_USOL     ;25centavos que se ha de pagar
MOVFF PRODH,HORAS_USOH

BSF T0CON,TMR0ON,0

BSF INTCON3,INT2IE,0       ;habilite la interrupcion por fichas,
BSF LATB,07h,0            ;encienda el modulo identificador
BSF INTCON3,INT2IF,0       ;y genere artificialmente una
RETURN

;-----
DESASIGNAR                   ;RUTINA ENCARGADA DE BORRAR EL REGISTRO
                               ;UNA VEZ FUE SALDADA LA DEUDA
BCF FLAGS0,old user,0     ;limpiar la bandera
BSF SSPCON2,SEN,0         ;iniciar comunicacion I2C
CALL waitMSSP
MOVFF DEVICE_ADD,SSPBUF
CALL waitMSSP
BTFSC SSPCON2,ACKSTAT,0
GOTO I2Cfail
MOVF WORD_ADD,W,0
ADDLW H'01'
MOVWF SSPBUF,0
CALL waitMSSP
BTFSC SSPCON2,ACKSTAT,0
GOTO I2Cfail
CLRF WREG,0               ;escribir 0's en el byte que controla
MOVWF SSPBUF,0           ;el renglon del presente locker
CALL waitMSSP
BTFSC SSPCON2,ACKSTAT,0
GOTO I2Cfail
BSF SSPCON2,PEN,0
CALL waitMSSP

CALL COMM_SLAVE
RETURN

;-----
MENSAJES_VOZ                 ;RUTINA DE CREACION DE LA PILA
                               ;DE LOS MENSAJES DE VOZ A PRONUNCIAR
LFSR 2,H'151'            ;colocacion del puntero

SWAPF CARNET1,W,0         ;obtencion del primer digito del carne
ANDLW B'00001111'
ADDLW H'20'               ;para la obtencion de la direccion del bus
MOVWF POSTINC2,0         ;que activa precisamente es palabra

```

```

MOVWF CARNET1,W,0           ;obtencion del segundo digito del carne
ANDLW B'00001111'
ADDLW H'20'
MOVWF POSTINC2,0

SWAPF CARNET2,W,0           ;obtencion del tercer digito del carne
ANDLW B'00001111'
ADDLW H'20'
MOVWF POSTINC2,0
MOVWF CARNET2,0             ;obtencion del cuarto digito del carne
ANDLW B'00001111'
ADDLW H'20'
MOVWF POSTINC2,0

SWAPF CARNETS,W,0           ;obtencion del quinto digito del carne
ANDLW B'00001111'
ADDLW H'20'
MOVWF POSTINC2,0

INCF FSR2L,F,0

SWAPF LOCKER,W,0            ;obtencion del primer caracter del locker
ANDLW B'00001111'
ADDLW H'20'
MOVWF POSTINC2,0
MOVWF LOCKER,W,0            ;obtencion del segundo caracter del locker
ANDLW B'00001111'
ADDLW H'20'
MOVWF POSTINC2,0

MOVLW H'50'                 ;inicializacion de la variable
MOVWF NUM_PALABRA,0
BCF INTCON,INTOIF,0         ;limpieza de bandera previa
BSF INTCON,INTOIE,0        ;habilitacion de int de voz
BSF INTCON,INT0IF,0        ;generacion de un interrupto
RETURN

;-----
FORMAT_EEPROM               ;RUTINA ENCARGADA DE BORRAR TODOS
                             ;LOS REGISTROS DE LA EEPROM
CLRF LATD,0                 ;limpiar el bus de datos
MOVLW H'10'
MOVWF LATA,0                ;y cargar ceros en los tres latches
CALL WAIT                   ;para borrar el despliegue
MOVLW H'11'
MOVWF LATA,0
CALL WAIT
MOVLW H'12'
MOVWF LATA,0
CALL WAIT
MOVLW H'13'                 ;dejar la direccion H'13' en el bus
MOVWF LATA,0
MOVLW H'A0'                 ;direccionar a EEPROM
MOVWF DEVICE_ADD,0
CLRF WORD_ADD,0
MOVLW H'A0'                 ;cargar primer valor a etiquetar
MOVWF ETIQUETA,0
MOVLW D'7'
MOVWF CONTADOR,0
I2Cwrite formateo
BSF SSPCON2,RSEN,0         ;generar un Restart
CALL waitMSSP
MOVWF DEVICE_ADD,SSPBUF    ;cargar direccion de EEPROM
CALL waitMSSP
BTFS SSPCON2,ACKSTAT,0     ;esta el chip listo?
BRA I2Cwrite formateo      ;no: intente de nuevo
MOVWF WORD_ADD,SSPBUF      ;si: indique la palabra a modificar
CALL waitMSSP

```

```

    BTFSC SSPCON2,ACKSTAT,0
    GOTO I2Cfail
    MOVFF ETIQUETA,SSPBUF           ;escribir en la primera posicion la etiqueta
    CALL waitMSSP
    BTFSC SSPCON2,ACKSTAT,0
    GOTO I2Cfail
ciclo formateo
    MOVLW H'00'                     ;y en las subsecuentes 7 escribir ceros
    MOVWF SSPBUF, 0
    CALL waitMSSP
    BTFSC SSPCON2,ACKSTAT,0
    GOTO I2Cfail
    DECFSZ CONTADOR,F,0
    BRA ciclo_formateo
    BSF SSPCON2,PEN,0               ;generar un stop
    CALL waitMSSP
    INCF ETIQUETA,F,0              ;incrementar la etiqueta
    MOVLW D'7'                     ;y reiniciar variables
    MOVWF CONTADOR,0
    MOVF WORD_ADD,W,0
    ADDLW D'8'
    MOVWF WORD_ADD,0
    BTFSS STATUS,C,0
    GOTO I2Cwrite_formateo
    MOVF DEVICE_ADD,W,0
    ADDLW D'2'
    MOVWF DEVICE_ADD,0
    XORLW B'10100110'             ;hasta que se haya escrito los 96 registros
    BNZ I2Cwrite_formateo
    BTFSS PORTB,06h,0             ;esperar hasta que la llave haya sido retirada
    BRA $-2
    RETURN

;-----
DELAY                               ;RUTINA DE RETRASO EN LA PUESTA A TIERRA
    MOVLW H'06'
    MOVWF TIMER1,0
ciclo_delay
    MOVLW H'FF'
    MOVWF TIMER2,0
    DECFSZ TIMER2,F,0
    BRA $-2
    DECFSZ TIMER1,F,0
    BRA ciclo_delay
    RETURN

;-----
WAIT                               ;RUTINA DE RETRASO EN LA ESCRITURA A PUERTOS
    MOVLW H'10'
    MOVWF TIMER3,0
    DECFSZ TIMER3,F,0
    BRA $-2
    RETURN

;-----
INTERRUPTO                          ;RUTINA DE SERVICIO DE INTERRUPCIONES
    MOVFF LATA,TEMP_LATA          ;almacenamiento del ambiente
    MOVFF LAID,TEMP_LATD
    MOVFF FSR0H,TEMP_FSR0H
    MOVFF FSR0L,TEMP_FSR0L
    MOVFF FSR1H,TEMP_FSR1H
    MOVFF FSR1L,TEMP_FSR1L
    MOVFF FSR2H,TEMP_FSR2H
    MOVFF FSR2L,TEMP_FSR2L

    BTFSC INTCON3,INT1IF,0        ;fue una int del reloj?
    BRA INT_RELOJ                 ;salte a INT_RELOJ

```

```

BTFSC PIR1,RCIF,0           ;fue una int de recepcion serial?
BRA INT RX                  ;salte a INT RX

BTFSC INTCON,TMR0IF,0
BRA INT TIMERO

BTFSC INTCON3,INT2IF,0     ;fue una int de insercion de monedas?
BRA INT FICHA              ;salte a INT FICHA

BTFSC INTCON,INTOIF,0     ;fue una int de fin de mensaje de voz?
BRA INT VOZ                ;salte a INT VOZ

BRA EXIT_INTERRUPTO        ;salida por generacion de int erronea

;-----
INT_RELOJ
BCF INTCON3,INT1IF,0       ;resetea la bandera del PIC
MOVLW (H'E3'-ritmo)
CPFSLT ORIGEN_DISP,0      ;rutina para determinar el ritmo del
BRA $+6                    ;scrolling del mensaje
MOVLW ritmo
ADDWF ORIGEN_DISP,F,0

CLRF TRISD,0              ;bus de datos como salida

MOVLW B'01001111'         ;resetea la bandera del RTC
MOVWF LATD,0
MOVLW H'60'
MOVWF LATA,0              ;main status reg
CALL WAIT
BCF LATE,01h,0            ;write!
CALL WAIT
BSF LATE,01h,0
INCF RELOJAZO,F,0         ;control sobre el numero de interrupcion

BTFSC PORTE,02h,0         ;se esta ajustando el reloj?
BRA $+6                    ;no: continuar en sin ajuste
MOVLW D'10'              ;si: pretender que transcurrio un seg
MOVWF RELOJAZO,0

MOVF RELOJAZO,W,0
XORLW D'5'
BTFSC STATUS,Z,0         ;ha transcurrido medio segundo?
BSF FLAGS0,medio_seg,0  ;si: encender la bandera medio seg
MOVLW D'10'
CPFSEQ RELOJAZO,0        ;ha transcurrido un segundo?
BRA EXIT_INTERRUPTO      ;no: salir del interrupto

;ha transcurrido un segundo...
CLRF RELOJAZO,0          ;reiniciar cuenta
BSF FLAGS0,un_seg,0     ;encender la bandera un_seg
SETF TRISD,0            ;bus de datos como entrada

MOVLW H'47'
MOVWF LATA,0             ;minutes counter reg
CALL WAIT
BCF LATE,00h,0
CALL WAIT
MOVFF PORTD,MIN         ;guarda los minutos en MIN
BSF LATE,00h,0

MOVLW H'48'
MOVWF LATA,0             ;hours counter reg
CALL WAIT
BCF LATE,00h,0          ;read!
CALL WAIT
MOVFF PORTD,HRS         ;guarda las horas en HRS
BSF LATE,00h,0

```

```

MOVLW H'4E'
MOVWF LATA, 0 ;days counter reg
CALL WAIT
BCF LATE,00h,0 ;read!
CALL WAIT
MOVFF PORTD, DAY ;guarda el dia de la semana en DAY
BSF LATE,00h,0
CLRf TRISD,0 ;bus de datos como salida

BTFSC PORTE,02h,0 ;se esta ajustando el reloj?
BRA EXIT_INTERRUPTO ;no: continuar en next
MOVf MIN,W,0 ;si: sumar 1 a los minutos
ADDLW H'01'
DAW ;ajuste BCD a la suma
BTFSC STATUS,DC,0 ;se produjo un digit carry?
ADDLW H'10' ;si: sumar 1 a las decenas
MOVWF MIN,0 ;actualizar la variable
BCF STATUS,DC,0 ;limpiar la bandera digit carry

MOVLW H'60' ;el incremento corresponde al
CPFSEQ MIN,0 ;minuto 60?
BRA act_minutos ;no: continuar con actualizacion de minutos
CLRf MIN,0 ;si: rollover a 00 y actualizar
MOVf HRS,W,0 ;la variable HRS
ADDLW H'01'
DAW
BTFSC STATUS,DC,0 ;se produjo un digit carry?
ADDLW H'10' ;si: sumar 1 a las decenas
MOVWF HRS,0 ;actualizar la variable HRS
BCF STATUS,DC,0 ;limpiar la bandera digit carry
XORLW H'24' ;el incremento corresponde a
BTFSC STATUS,Z,0 ;la hora 24?
CLRf HRS,0 ;si: rollover a 00
act_horas
MOVFF HRS,LATD ;actualizar horas en el RTC
MOVLW H'68'
MOVWF LATA,0 ;hrs counter
CALL WAIT
BCF LATE,01h,0 ;write!
CALL WAIT
BSF LATE,01h,0

act_minutos
MOVFF MIN,LATD ;actualizar minutos en el RTC
MOVLW H'67'
MOVWF LATA,0 ;min counter
CALL WAIT
BCF LATE,01h,0 ;write!
CALL WAIT
BSF LATE,01h,0
BRA EXIT_INTERRUPTO ;salir de la interrupcion

;-----
INT_RX
CLRf FSR1H,0 ;colocacion del puntero para acomodar
MOVFF POS_CARNET,FSR1L ;los caracteres recibidos serialmente
MOVf RCREG,W,0
ANDLW B'00001111'
ADDWF INDF1,F,0 ;obtencion del equivalente BCD y su
BTFSS DIGITO_RX,00h,0 ;correcta colocacion en los bytes
SWAPF INDF1,F,0
BTFSC DIGITO_RX,00h,0
INCF POS_CARNET,F,0
INCF DIGITO_RX,F,0
MOVf DIGITO_RX,W,0
XORLW D'5'
BTFSC STATUS,Z,0 ;se recibio ya el ultimo caracter?

```

```

BSF FLAGS0,carnet,0          ;si: marque la recepcion de un carnet
BRA EXIT INTERRUPTO        ;salir de la interrupcion

;- - - - -
INT_VOZ
  BTFSC INTCON3,INT2IF,0
  BRA EXIT INTERRUPTO

  BCF INTCON,INTOIF,0      ;resetea la bandera del PIC
  LFSR 2,H'100'
  MOVF NUM PALABRA,W,0
  MOVWF FSR2L,0
  MOVWF INDF2,LATA
  BCF LATC,05h,0          ;habilita el chip de voz
  CALL WAIT
  BSF LATC,05h,0          ;deshabilita el chip de voz
  INCF NUM PALABRA,F,0

  MOVLW H'58'
  CPFSGT NUM_PALABRA,0
  BRA EXIT INTERRUPTO

  MOVLW H'5A'
  CPFSLT NUM PALABRA,0
  BRA $+6

  BTFSC FLAGS0,old user,0
  BRA EXIT INTERRUPTO

  BTFSS FLAGS0,old user,0
  CALL COMM SLAVE

  MOVLW H'50'
  MOVWF NUM PALABRA,0
  BCF INTCON, INT0IE,0    ;deshabilitacion de int de voz
  BRA EXIT INTERRUPTO

;- - - - -
INT_FICHA
  BCF INTCON3,INT2IF,0    ;resetea la bandera del PIC
  MOVWF PORTB,FICHA      ;obtencion de la identificacion de la
  MOVLW B'00111000'      ;la moneda insertada
  ANDWF FICHA,F,0

  CLRF TIEMPO_ACUM,0
  CLRF WREG,0
  MOVWF TMR0H,0          ;contadores en 0
  MOVWF TMR0L,0
  MOVE FICHA, W,0
  XORLW B'00010000'      ;fue una moneda de 25 centavos?
  BNZ no_ficha25c       ;no: value la siguiente opcion
  DECF HORAS_USOL,F,0    ;si: decrementar en uno la variable
  BC cuanto_saldo       ;es negative? no: calcular costo
  DECF HORAS_USOH,F,0    ;si: decrementar la parte alta
  BC cuanto_saldo       ;es negative? no: calcular costo
  CLRF HORAS_USOL,0      ;si: costo saldado, ambas variables
  CLRF HORAS~USOH,0     ;son cero
  BRA cuanto_saldo      ;calcular costo

no_ficha25c
  MOVE FICHA,W,0
  XORLW B'00011000'      ;fue una moneda de 50 centavos?
  BNZ no_ficha50c       ;no: value la siguiente opcion
  MOVLW D'2'            ;si: decrementar en dos la variable
  SUBWF HORAS_USOL,F,0
  BC cuanto_saldo       ;es negative? no: calcular costo
  DECF HORAS_USOH,F,0    ;si: decrementar la parte alta
  BC cuanto_saldo       ;es negative? no: calcular costo

```

```

    CLRF HORAS_USOL,0           ;si: costo saldado, ambas variables
    CLRF HORAS_USOH,0         ;son cero
    BRA cuanto_saldo         ;calcular costo

no_ficha50c
    MOVF FICHA,W,0
    XORLW B'00100000'        ;fue una moneda de 1 quetzal?
    BNZ cuanto_saldo        ;no: calcular costo (moneda no valida)
    MOVLW D'4'              ;si: decrementar en 4 la variable
    SUBWF HORAS_OSOL,F,0
    BC cuanto_saldo         ;es negative? no: calcular costo
    DECF HORAS_USOH,F,0     ;si: decrementar la parte alta
    BC cuanto_saldo        ;es negative? no: calcular costo
    CLRF HORAS_USOL,0       ;si: costo saldado, ambas variables
    CLRF HORAS_USOH,0       ;son cero

cuanto_saldo
    CLRF CENTAVOS,0         ;inicializacion de las variables
    CLRF QUETZALES,0
    MOVF HORAS_USOL,F,0
    BNZ debe_pisto
    MOVF HORAS_USOH,F,0    ;es alguna de las variables es diferente
    BNZ debe_pisto        ;de cero? si: se debe dinero
    BCF FLAGS0,saldo,0    ;no: limpiar la bandera
    BCF INTCON3,INT2IE,0  ;deshabilitar el int de fichas y
    BCF LATB,07h,0        ;apagar el modulo de monedas
    BRA EXIT_INTERRUPTO   ;salir del interrupto

debe_pisto
    BSF FLAGS0,saldo,0    ;set la bandera de saldo pendiente
    MOVFF HORAS_USOL,VUELTAS1 ;carga de contadores para el ciclo
    MOVFF HORAS_USOH,VUELTAS2

loop_dinero
    MOVF CENTAVOS,W,0     ;sumar 25 por cada vuelta
    ADDLW H'25'
    DAW
    MOVWF CENTAVOS,0
    BNC no_carry_qtzs    ;se genero un carry? no: ir a no_carry_qtzs
    MOVF QUETZALES,W,0   ;si: actualizar el valor de quetzales
    ADDLW H'01'
    DAW
    BTFSC STATUS,DC,0
    ADDLW H'10'
    MOVWF QUETZALES,0
    XORLW H'99'          ;se llego al maximo cobrable?
    BZ EXIT_INTERRUPTO  ;si: salir del interrupto

no_carry_qtzs
    DECFSZ VUELTAS1,F,0
    BRA loop_dinero
    DECF VUELTAS2,F,0
    BC loop_dinero
    BRA EXIT_INTERRUPTO

; - - - - -
INT_TIMER0
    BCF INTCON,TMR0IF,0
    INCF TIEMPO_ACUM,F,0
    MOVLW H'08'
    CPFSEQ TIEMPO_ACUM,0
    BRA EXIT_INTERRUPTO
    BCF TOCON,TMROON,0
    CLRF TIEMPO_ACUM,0
    CLRF WREG,0
    MOVWF TMR0H,0        ;contadores en 0
    MOVWF TMR0L,0

    BCF FLAGS0,saldo,0
    BCF INTCON3,INT2IE,0

```

```
BCF FLAGS0,old user,0
BRA EXIT INTERRUPTO

EXIT_INTERRUPTO

MOVFF TEMP_FSR2L,FSR2L      ;reestablecer el ambiente
MOVFF TEMP_FSR2H,FSR2H
MOVFF TEMP_FSR1L,FSR1L
MOVFF TEMP_FSR1H, FSR1H
MOVFF TEMP_FSR0L,FSR0L
MOVFF TEMP_FSR0H,FSR0H
MOVFF TEMP_LATD,LATD      ;reestablecer valores en puertos
MOVFF TEMP_LATA,LATA      ;de direcciones y datos
RETFIE FAST

END
```

```

;*****
;* UNIVERSIDAD DEL VALLE DE GUATEMALA
;* DEPARTAMENTO DE INGENIERIA ELECTRONICA
;* TRABAJO DE GRADUACION
;* FIRMWARE CIRCUITO ESCLAVO - ADMINISTRADOR DE CASILLEROS
;*
;* JUANCARLOS BURMESTER CASTILLO CARNE 97008
;*****

list p=18F248
#include <p18f248.inc>
ERRORLEVEL -302

__CONFIG __CONFIG1H, _OSCS_OFF_1H & _ECIO_OSC_1H
__CONFIG __CONFIG2L, BOR ON 2L & PWRT ON 2L & BORV 45 2L
__CONFIG __CONFIG2H, WDT OFF 2H
__CONFIG __CONFIG4L, _STVR_OFF_4L & _LVP_OFF_4L & _DEBUG_OFF_4L
__CONFIG __CONFIG5L, CPO OFF 5L & CP1 OFF 5L
__CONFIG __CONFIG5H, CPB OFF 5H & CPD OFF 5H
__CONFIG __CONFIG6L, _WRT0_OFF_6L & _WRT1_OFF_6L
__CONFIG __CONFIG6H, _WRTB_OFF_6H & _WRTC_OFF_6H & _WRTD_OFF_6H
__CONFIG __CONFIG7L, _EBTR0_OFF_7L & _EBTR1_OFF_7L
__CONFIG __CONFIG7H, _EBTRB_OFF_7H

;CONSTANTES
;*****
#define ESPERA D'31'
;*****

;VARIABLES
;*****

CBLOCK H'00'
  FLAGS
  LEDS1
  LEDS2
  LEDS3
  LEDS4
  IDENTIDAD
  MAX_LOCKERS
  CONTADOR
  PUERTAS1_DSORD
  PUERTAS2_DSORD
  SOLENOIDES1
  SOLENOIDES2
  TEMP_SSPSTAT
  RX_I2C
  TX_I2C
  TIMER1
  TIMER2
  TIMER3
  ENDC

;*****

;EQUIVALENCIAS
;*****
#define SW FILAS H'00' ;en LATA
#define LATCH_SOL2 H'01'
#define NOT OE SW ID H'02'
#define NOT SW COLS H'03'
#define LATCH_LEDS H'05'
#define LATCH SOLS1 H'05' ;en LATC

#define COMANDOI2C H'00' ;en FLAGS0
#define CONFIG_MASTER H'01'

```

```

#define PUERTA                H'00'
#define LUZ                   H'01'
#define SOL                    H'02'
#define SE_USO                 H'03'    ;en variables de control
#define SACO_COSAS             H'04'    ;de cada locker
#define ASIGNAR                H'05'
#define MOD_LOCKER             H'06'
#define REUSAR                 H'07'

;*****

    ORG 00h
    GOTO MAIN
    ORG 08h
    GOTO INTERRUPTO

MAIN
    CALL INIT_MEMORY
    CALL INIT_PUERTOS
    CALL INIT_VALORES
    CALL INIT_MAESTRO

    LFSR 0,H'0001'

loop_principal
    CLRF TRISB,0                ;todas las lineas como salidas

    CALL LUCES                  ;obtencion del estado de los leds

    MOVFF POSTINC0,LATB
    BSF LATA,LATCH_LEDS,0      ;carga del dato al latch de leds
    CALL WAIT
    BCF LATA,LATCH_LEDS,0

    CALL POLEO_PUERTAS         ;chequeo del estado de las puertas

    CALL SOLS                   ;obtencion del estado de los solenoides

    MOVFF SOLENOIDES1,LATB     ;carga del estado del primer grupo
    BSF LATC,LATCH_SOLS1,0     ;de solenoides a su latch
    CALL WAIT
    BCF LATC,LATCH_SOLS1,0     ;solenoides 0, 1, 2, 3, 4, 5, 6, 7

    CALL WAIT

    MOVFF SOLENOIDES2,LATB     ;carga del estado del segundo grupo
    BSF LATA,LATCH_SOLS2,0     ;de solenoides a su latch
    CALL WAIT
    BCF LATA,LATCH_SOLS2,0     ;solenoides 8, 9, A, B, C, D, E, F

    CALL DELAY                  ;retraso

    BTFSC FLAGS,COMANDOI2C,0   ;se ha recibido orden I2C?
    CALL ACTUAR                 ;si: llamar a actuar

    MOVLW H'05'                ;se barrieron ya los 4 valores de leds?
    CPFSEQ FSR0L,0
    BRA loop_principal         ;no: repita el ciclo
    LFSR 0,H'0001'             ;si: reinicie la variable y
    BRA loop_principal         ;repita el ciclo

;*****
INIT_MEMORY                    ;INICIALIZACION DE MEMORIA
    LFSR 0,H'0000'             ;Rutina de borrado de los bancos 0 y 1
    CLRF POSTINC0,0
    BTFSS FSR0H,01h
    BRA $-4

```

```

RETURN

;*****
INIT_PUERTOS                                ;INICIALIZACION DE PUERTOS
MOVW B'00001100'                            ;valores iniciales en las
MOVWF LATA,0                                ;salidas de los puertos
CLRF LATB,0
CLRF LATC,0
CLRF TRISA,0                                ;todas salidas
SETF TRISB,0                                ;todas entradas (bus datos)
MOVW B'00011111'
MOVWF TRISC,0
MOVW B'00000110'                            ;todos los pines del puerto A como
MOVWF ADCON1,0                              ;entradas/salidas digitales

;DE I2C
CLRF SSPCON2,0                              ;clock stretching solo para el envio
MOVW B'100110110'                          ;habilitacion del puerto serial sincrono,
MOVWF SSPCON1,0                              ;I2C slave mode, 7 bit address
MOVW B'10000000'
MOVWF SSPSTAT,0                             ;slew rate deshabilitado

;DEL TIMER
MOVW B'00000010'                            ;timer off, 16 bit, internal inst.
MOVWF TOCON,0                               ;low to high, prescaler 1:8
CLRF WREG,0
MOVWF TMR0H,0                               ;contadores en 0
MOVWF TMR0L,0
BSF TOCON,TMR0ON,0                          ;arranca el conteo

;DE INTERRUPTOS
BCF RCON,IPEN,0                             ;deshabilita prioridad en interruptos
BCF PIR1,SSPIF,0                            ;limpia la bandera de I2C
BSF PIR1,SSPIF,0                            ;habilita interrupto de I2C
BCF INTCON,TMR0IF,0                         ;limpia la bandera del timer 0
BSF INTCON,TMR0IE,0                         ;habilita interrupto del timer 0
BSF INTCON,PEIE,0                           ;habilitacion de ints perifericos
BSF INTCON,GIE,0                            ;habilitacion global de interruptos
RETURN

;*****
INIT_VALORES
BCF LATA,NOT OE SW ID,0                     ;lectura de los switches BCD de
CALL WAIT                                    ;identificacion del esclavo
MOVWF PORTB, IDENTIDAD
BSF LATA,NOT OE SW ID,0
MOVF IDENTIDAD,W,0
ANDW B'01110000'                            ;obtencion de la identificacion del switch
MOVWF SSPADD,0                              ;como su direccion I2C
MOVF IDENTIDAD,W,0                          ;el nibble bajo de identidad
ANDW B'00001111'                            ;corresponde al numero max de lockers
MOVWF MAX LOCKERS,0                         ;sobre los que se va a operar
CLRF TRISB,0

MOVWF MAX LOCKERS,CONTADOR
LFSR 1,H'0030'                              ;encender la luz de todos
BSF INDF1,LUZ,0                             ;los lockers operables
INCF FSR1L,F,0
DECF CONTADOR,F,0
BC $-6
RETURN

;*****
INIT_MAESTRO
BSF LATC,07h,0

```

```

    BTFSS FLAGS,CONFIG MASTER,0
    BRA $-2
    LFSR 0,H'0030'
ciclo_act_status
    BTFSS FLAGS,COMANDOI2C,0
    BRA $-2
    BCF FLAGS,COMANDOI2C,0
    MOVF RX I2C,F,0
    BZ locker libre
locker_ocupado
    MOVLW B'00001000'
    MOVWF POSTINC0,0
    BRA $+4
locker_libre
    INCF FSR0L,F, 0
    MOVLW H'40'
    CPFSEQ FSR0L,0
    BRA ciclo act status
    RETURN
;*****
LUCES                                     ;RUTINA DE OBTENCION DEL ESTADO
    CLRF LEDS1,0                          ;DE LOS LEDS
    CLRF LEDS2,0
    CLRF LEDS3,0                          ;limpieza previa de las variables
    CLRF LEDS4,0
    LFSR 1,H'0030'

    MOVLW B'00010000'                     ;si el bit de LUZ esta prendido para
    MOVWF LEDS1,0                          ;el locker, enciendolo en leds
    BTFSC POSTINC1,LUZ,0                   ;casillero 0
    BSF LEDS1,00h,0
    BTFSC POSTINC1,LUZ,0                   ;casillero 1
    BSF LEDS1,01h,0
    BTFSC POSTINC1,LUZ,0                   ;casillero 2
    BSF LEDS1,02h,0
    BTFSC POSTINC1,LUZ,0                   ;casillero 3
    BSF LEDS1,03h,0

    MOVLW B'00100000'
    MOVWF LEDS2,0
    BTFSC POSTINC1,LUZ,0                   ;casillero 4
    BSF LEDS2,00h,0
    BTFSC POSTINC1,LUZ,0                   ;casillero 5
    BSF LEDS2,01h,0
    BTFSC POSTINC1,LUZ,0                   ;casillero 6
    BSF LEDS2,02h,0
    BTFSC POSTINC1,LUZ,0                   ;casillero 7
    BSF LEDS2,03h,0

    MOVLW B'01000000'
    MOVWF LEDS3,0
    BTFSC POSTINC1,LUZ,0                   ;casillero 8
    BSF LEDS3,00h,0
    BTFSC POSTINC1,LUZ,0                   ;casillero 9
    BSF LEDS3,01h, 0
    BTFSC POSTINC1,LUZ,0                   ;casillero A
    BSF LEDS3,02h,0
    BTFSC POSTINC1,LUZ,0                   ;casillero B
    BSF LEDS3,03h,0

    MOVLW B'10000000'
    MOVWF LEDS4,0
    BTFSC POSTINC1,LUZ,0                   ;casillero C
    BSF LEDS4,00h,0
    BTFSC POSTINC1,LUZ,0                   ;casillero D
    BSF LEDS4,01h,0
    BTFSC POSTINC1,LUZ,0                   ;casillero E

```

```

BSF LEDS4,02h,0
BTFSC POSTINC1,LUZ,0           ;casillero F
BSF LEDS4,03h,0
RETURN

;*****
SOLS                             ;RUTINA DE OBTENCION DEL ESTADO
  CLRf SOLENOIDES1,0           ;DE LOS SOLENOIDES
  CLRf SOLENOIDES2,0           ;limpieza de variables
  LFSR 1,H'0030'               ;si el solenoide esta activado para
                               ;el locker, enciendolo en SOLENOIDES

  BTFSC POSTINC1,SOL,0         ;casillero 0
  BSF SOLENOIDES1,00h,0
  BTFSC POSTINC1,SOL,0         ;casillero 1
  BSF SOLENOIDES1,01h,0
  BTFSC POSTINC1,SOL,0         ;casillero 2
  BSF SOLENOIDES1,02h,0
  BTFSC POSTINC1,SOL,0         ;casillero 3
  BSF SOLENOIDES1,03h,0
  BTFSC POSTINC1,SOL,0         ;casillero 4
  BSF SOLENOIDES1,04h,0
  BTFSC POSTINC1,SOL,0         ;casillero 5
  BSF SOLENOIDES1,05h,0
  BTFSC POSTINC1,SOL,0         ;casillero 6
  BSF SOLENOIDES1,06h,0
  BTFSC POSTINC1,SOL,0         ;casillero 7
  BSF SOLENOIDES1,07h,0

  BTFSC POSTINC1,SOL,0         ;casillero 8
  BSF SOLENOIDES2,07h,0
  BTFSC POSTINC1,SOL,0         ;casillero 9
  BSF SOLENOIDES2,06h,0
  BTFSC POSTINC1,SOL,0         ;casillero A
  BSF SOLENOIDES2,05h,0
  BTFSC POSTINC1,SOL,0         ;casillero B
  BSF SOLENOIDES2,04h,0
  BTFSC POSTINC1,SOL,0         ;casillero C
  BSF SOLENOIDES2,03h,0
  BTFSC POSTINC1,SOL,0         ;casillero D
  BSF SOLENOIDES2,02h,0
  BTFSC POSTINC1,SOL,0         ;casillero E
  BSF SOLENOIDES2,01h,0
  BTFSC POSTINC1,SOL,0         ;casillero F
  BSF SOLENOIDES2,00h,0

RETURN

;*****
POLEO PUERTAS                     ;RUTINA DE CHEQUEO DEL ESTADO DE PUERTAS
  CLRf LATB,0
  MOVLW B'00001111'           ;configuracion de entradas/salidas
  MOVWF TRISB,0
  NOP
  NOP
  BCF LATA,NOT SW COLS,0      ;habilitacion del transceiver columnas
  BSF LATA,SW_FILAS,0        ;habilitacion del transceiver filas

  MOVLW B'00010000'           ;colocacion de 1 en la primera columna
  MOVWF LATB,0
  CALL WAIT
  MOVF PORTB,W,0              ;recepcion de la respuesta en las filas
  ANDLW B'00001111'
  MOVWF PUERTAS1_DSORD,0      ;almacenamiento en nibble bajo

  MOVLW B'00100000'           ;colocacion de 1 en la segunda columna
  MOVWF LATB,0
  CALL WAIT

```

```

MOVF PORTB,W,0           ;recepcion de la respuesta en las filas
ANDLW B'00001111'
SWAPF WREG,W,0
ADDWF PUERTAS1_DSORD,F,0 ;almacenamiento en nibble alto

MOVLW B'01000000'       ;colocacion de 1 en la tercera columna
MOVWF LATB,0
CALL WAIT
MOVF PORTB,W,0         ;recepcion de la respuesta en las filas
ANDLW B'00001111'
MOVWF PUERTAS2_DSORD,0 ;almacenamiento en nibble bajo

MOVLW B'10000000'       ;colocacion de 1 en la cuarta columna
MOVWF LATB,0
CALL WAIT
MOVF PORTB,W,0         ;recepcion de la respuesta en las filas
ANDLW B'00001111'
SWAPF WREG,W,0
ADDWF PUERTAS2_DSORD,F,0 ;almacenamiento en nibble alto

CLRF LATB,0            ;limpieza del bus
NOP
NOP
BCF LATA,SW FILAS,0    ;deshabilitacion del transciever filas
BSF LATA,NOT_SW_COLS,0 ;deshabilitacion del transciever columnas

CLRF TRISB,0          ;todas salidas

LFSR 1,H'0030'

BTFS PUERTAS1_DSORD,00h,0 ;del casillero 0
BSF INDF1,PUERTA,0
BTFS PUERTAS1_DSORD,00h,0
BCF INDF1,PUERTA,0
INCF FSR1L,F,0

BTFS PUERTAS1_DSORD,05h,0 ;del casillero 1
BSF INDF1,PUERTA,0
BTFS PUERTAS1_DSORD,05h,0
BCF INDF1,PUERTA,0
INCF FSR1L,F,0

BTFS PUERTAS2_DSORD,02h,0 ;del casillero 2
BSF INDF1,PUERTA,0
BTFS PUERTAS2_DSORD,02h,0
BCF INDF1,PUERTA,0
INCF FSR1L,F,0

BTFS PUERTAS2_DSORD,07h,0 ;del casillero 3
BSF INDF1,PUERTA,0
BTFS PUERTAS2_DSORD,07h,0
BCF INDF1,PUERTA,0
INCF FSR1L,F,0

BTFS PUERTAS1_DSORD,04h,0 ;del casillero 4
BSF INDF1,PUERTA,0
BTFS PUERTAS1_DSORD,04h,0
BCF INDF1,PUERTA,0
INCF FSR1L,F,0

BTFS PUERTAS2_DSORD,01h,0 ;del casillero 5
BSF INDF1,PUERTA,0
BTFS PUERTAS2_DSORD,01h,0
BCF INDF1,PUERTA,0
INCF FSR1L,F,0

BTFS PUERTAS2_DSORD,06h,0 ;del casillero 6
BSF INDF1,PUERTA,0

```

```

BTFS PUERTAS2_DSORD,06h,0
BCF INDF1,PUERTA,0
INCF FSR1L,F,0

BTFS PUERTAS2_DSORD,00h,0 ;del casillero 7
BSF INDF1,PUERTA,0
BTFS PUERTAS2_DSORD,00h,0
BCF INDF1,PUERTA,0
INCF FSR1L,F,0

BTFS PUERTAS2_DSORD,05h,0 ;del casillero 8
BSF INDF1,PUERTA,0
BTFS PUERTAS2_DSORD,05h,0
BCF INDF1,PUERTA,0
INCF FSR1L,F,0

BTFS PUERTAS2_DSORD,04h,0 ;del casillero 9
BSF INDF1,PUERTA,0
BTFS PUERTAS2_DSORD,04h,0
BCF INDF1,PUERTA,0
INCF FSR1L,F,0

BTFS PUERTAS1_DSORD,01h,0 ;del casillero A
BSF INDF1,PUERTA,0
BTFS PUERTAS1_DSORD,01h,0
BCF INDF1,PUERTA,0
INCF FSR1L,F,0

BTFS PUERTAS1_DSORD,06h,0 ;del casillero B
BSF INDF1,PUERTA,0
BTFS PUERTAS1_DSORD,06h,0
BCF INDF1,PUERTA,0
INCF FSR1L,F,0

BTFS PUERTAS2_DSORD,03h,0 ;del casillero C
BSF INDF1,PUERTA,0
BTFS PUERTAS2_DSORD,03h,0
BCF INDF1,PUERTA,0
INCF FSR1L,F,0

BTFS PUERTAS1_DSORD,02h,0 ;del casillero D
BSF INDF1,PUERTA,0
BTFS PUERTAS1_DSORD,02h,0
BCF INDF1,PUERTA,0
INCF FSR1L,F,0

BTFS PUERTAS1_DSORD,07h,0 ;del casillero E
BSF INDF1,PUERTA,0
BTFS PUERTAS1_DSORD,07h,0
BCF INDF1,PUERTA,0
INCF FSR1L,F,0

BTFS PUERTAS1_DSORD,03h,0 ;del casillero F
BSF INDF1,PUERTA,0
BTFS PUERTAS1_DSORD,03h,0
BCF INDF1,PUERTA,0

RETURN

;*****
ACTUAR ;RUTINA DE ACTIVACION DE UN SOLENOIDE
BCF FLAGS,COMANDOI2C,0 ;limpiar la bandera
MOVW RX I2C,W,0
MOVWF CONTADOR,0
SUBWF MAX_LOCKERS,W,0
BTFS STATUS,C,0 ;es un locker operable?
RETURN ;no: ignore la orden

```

```

LFSR 1,H'0030'
MOVWF RX I2C,W,0          ;ubicacion del puntero en la posicion
ADDWF FSR1L,F,0          ;del locker en el que se actua
BSF INDF1,MOD_LOCKER,0   ;marcarlo como modificable
BTFSS INDF1,SE_USO,0     ;ya fue usado?
BSF INDF1,ASIGNAR,0     ;no, indique asignacion
BTFSC INDF1,SE_USO,0
BCF INDF1,ASIGNAR,0     ;si, indique desasignacion
BSF INDF1,SOL,0         ;accionar el solenoide respective
BTFSS INDF1,ASIGNAR,0
RETURN
MOVLW H'10'              ;si el locker se asigna...
ADDWF FSR1L,F,0         ;cargar el contador de espera en la
MOVWF ESPERA            ;asignacion
MOVWF INDF1,0
RETURN

;*****
DELAY                    ;RUTINA DE RETRASO EN ILUMINACION
    MOVLW H'06'          ;DE UN LED
    MOVWF TIMER1,0
ciclo delay
    MOVLW H'FF'
    MOVWF TIMER2, 0
    DECFSZ TIMER2,F,0
    BRA $-2
    DECFSZ TIMER1,F,0
    BRA ciclo delay
    RETURN

;*****
WAIT                    ;RUTINA DE RETRASO EN ESCRITURA
    MOVLW H'10'          ;A PUERTOS
    MOVWF TIMER3,0
    DECFSZ TIMER3,F,0
    BRA $-2
    RETURN

;*****
INTERRUPTO

    BTFSC PIR1,SSPIF,0   ;es una interrupcion de I2C?
    BRA INT_MSSP         ;si: saltar a INT_MSSP
    BTFSC INTCON,TMR0IF,0 ;es una interrupcion del timer?
    BRA INT_TIMER        ;si: saltar a INT_TIMER
    BRA EXIT_INTERRUPTO

;-----
INT_MSSP                ;se recibio un mensaje I2C...
    BCF PIR1,SSPIF,0    ;limpiar bandera de interrupcion
    MOVF SSPSTAT,W,0
    ANDLW B'00101101'   ;para determinar el origen de la
    MOVWF TEMP SSPSTAT,0 ;interrupcion
estad1
    MOVLW B'00001001'   ;write, last byte address, buffer full
    CPFSEQ TEMP SSPSTAT,0 ;S=1, D_A=0, R_W=0, BF=1
    BRA estado2
    MOVF SSPBUF,W,0     ;hacer una lectura para resetear banderas
    BRA EXIT_INTERRUPTO
estad2
    MOVLW B'00101001'   ;write, last byte data, buffer full
    CPFSEQ TEMP SSPSTAT,0 ;S=1, D_A=1, R_W=0, BF=1
    BRA estado3
    MOVWF SSPBUF,RX_I2C ;el dato recibido se almacena y se
    BSF FLAGS,COMANDOI2C,0 ;sennaliza su recepcion
    BRA EXIT_INTERRUPTO
estad3
    MOVLW B'00001100'   ;read, last byte address, buffer empty
    ;S=1, D_A=0, RJSN1, BF=0

```

```

    CPFSEQ TEMP SSPSTAT,0
    BRA estado4
do_I2C write
    BTFSC SSPSTAT,BF,0           ;esta el buffer lleno?
    BRA do_I2C write           ;si: espere a que se descargue
cargar_info
    BCF SSPCON1,WCOL,0         ;limpiar bandera de colision
    LFSR 2,H'0030'
    BTFSC FLAGS,CONFIG_MASTER,0
    BRA ciclo_reusar
    MOVF MAX_LOCKERS,W,0
    MOVWF SSPBUF,0
    BSF FLAGS,CONFIG_MASTER,0
    BRA solo
ciclo_reusar
    BTFSS INDF2,REUSAR,0       ;identificar que locker tiene
    BRA siguiente             ;indicacion de REUSAR
    BCF INDF2,REUSAR,0
    MOVF FSR2L,W,0           ;una vez encuentre uno lo pone en
    MOVWF SSPBUF, 0         ;el buffer de transmision
    BRA solo
siguiente
    INCF FSR2L,F,0           ;busqueda en los 16 lockers
    MOVLW H'40'
    CPFSEQ FSR2L,0
    BRA ciclo_reusar
    CLRF WREG,0             ;si no se encuentra ninguno se
    MOVWF SSPBUF,0         ;envia un byte de ceros
solo
    BTFSC SSPCON1,WCOL,0     ;hubo colision en la escritura?
    BRA cargar_info         ;si: repita el proceso
    BSF SSPCON1,CKP,0       ;no: libere el reloj para transmitir
    BSF FLAGS,CONFIG_MASTER,0
    BRA EXIT_INTERRUPTO
estado4
    MOVLW B'00101100'       ;read, last byte data, buffer empty
    CPFSEQ TEMP SSPSTAT,0   ;S=1, D A=1, R W=1, BF=0
    BRA estado5             ;se solicito nuevo dato?
    BRA do_I2C_write       ;si: enviarlo nuevamente
estado5
    MOVLW B'00101000'       ;slave reset por NACK del master
    CPFSEQ TEMP_SSPSTAT,0   ;S=1, D A=1, R W=0, BF=0
    BRA I2Cerror           ;no fue ninguna de las razones?
    BRA EXIT_INTERRUPTO    ;reporte error
I2Cerror
    BSF LATC,06h,0         ;sennalizacion de error I2C
    NOP
    BRA EXIT_INTERRUPTO

;-----
INT_TIMER
    BCF INTCON,TMR0IF,0     ;ha transcurrido 1/4 de segundo
    LFSR 2,H'0030'         ;limpiar la bandera
ciclo_decremento
    BTFSS INDF2,MOD_LOCKER,0 ;esta marcado para modificar?
    BRA continue_ciclo     ;no: continue el ciclo
    BTFSS INDF2,ASIGNAR,0   ;esta en proceso de asignacion?
    BRA proc_desasignar    ;no: ir a proc_desasignar
proc_asignar
    BTFSC INDF2,PUERTA,0    ;esta la puerta abierta?
    BRA puerta_abiertal    ;si: vaya a puerta abierta
puerta_cerradal
    MOVLW H'10'           ;revisar el estado del contador asociado
    MOVF PLUSW2,F,0       ;esta el contador en cero?
    BZ contador_cero      ;si: ir a contador cero
    DECF PLUSW2,F,0       ;no: decrementarlo y
    BTG INDF2,LUZ,0       ;cambiar el estado de la luz
    BRA continue_ciclo

```

```

contador_cero
    BCF INDF2,SOL,0           ;desactivar solenoide
    BCF INDF2,MOD_LOCKER,0   ;limpiar indicacion de modificacion
    BTFSC INDF2,SE_USO,0     ;fue utilizado?
    BRA continue_ciclo      ;si: continuar ciclo
    BSF INDF2,LUZ,0         ;no: encender la luz
    BSF INDF2,REUSAR,0      ;indicar opcion de reuso
    BRA continue_ciclo

puerta_abierta1
    BSF INDF2,SE_USO,0       ;la puerta se abrio, senalizar su uso
    BCF INDF2,LUZ,0         ;apagar la luz
    MOVLW H'10'
    CLRF PLUSW2,0           ;borrar el contador
    BRA continue_ciclo

proc_desasignar
    BTFSC INDF2,PUERTA,0    ;esta la puerta abierta?
    BRA puerta_abierta2     ;si: ir a puerta abierta2
puerta_cerrada2
    BTFSS INDF2,SACO_COSAS,0 ;saco ya el usuario sus pertenencias?
    BRA continue_ciclo     ;no: continue el ciclo
    BSF INDF2,LUZ,0        ;si: encender la luz,
    BCF INDF2,SOL,0        ;desactivar el solenoide
    BCF INDF2,SE_USO,0     ;borrar indicacion de uso
    BCF INDF2,SACO_COSAS,0 ;borrar bandera de retire de pertenencias
    BCF INDF2,MOD_LOCKER,0 ;borrar bandera de modificacion
    BCF INDF2,REUSAR,0    ;borrar bandera de reutilizacion
    BRA continue_ciclo

puerta_abierta2
    BSF INDF2,SACO_COSAS,0  ;al abrir la puerta, se sennaliza
                           ;el retiro de pertenencias

continue_ciclo
    INCF FSR2L,F,0         ;pasar al siguiente locker
    MOVF MAX_LOCKERS,W,0   ;hasta el numero maximo operable
    ADDLW H'31'
    CPFSEQ FSR2L,0
    BRA ciclo_decremento
    BRA EXIT_INTERRUPTO
EXIT_INTERRUPTO
    RETFIE FAST

END

```

## IX. ANEXO C

---

---

Manual de operación



## *INTRODUCCIÓN*

Este documento acompaña como referencia al informe de tesis titulado SISTEMA ADMINISTRADOR DE CASILLEROS CON ACCESO POR MEDIO DE CÓDIGO DE BARRAS Y MECANISMO DE COBRO AUTOMÁTICO. A continuación se describen de manera sencilla los puntos que deben tener presente, tanto el usuario como el operario del sistema, para su debido manejo.

Inicialmente se plantean los lineamientos para la utilización del sistema en la parte *Usuario*, quien contrata el servicio de almacenamiento. Seguido, en la sección *Operario*, se detallan los aspectos de la administración.

La interfase del Sistema Administrador de Casilleros está diseñada de manera tal que sea amigable y fácil de usar. La persona que requiere de un espacio de almacenamiento debe acercarse a la unidad central, donde solicita un casillero haciendo uso de su identificación. El módulo central indica con mensajes de texto y un mensaje audible cuál es el espacio disponible y lo abre para su ocupación. Una vez colocadas sus pertenencias, el usuario cierra la puerta del casillero y ésta inmediatamente se asegura. Cuando la persona desee recuperar lo guardado, nuevamente se acerca al módulo central, éste le recordara cuál es el espacio que había sido asignado y procede a solicitar el pago correspondiente por la utilización del servicio. Cuando la deuda ha sido saldada en su totalidad, el casillero es liberado nuevamente para permitir el retiro de las pertenencias.

### **Los casilleros**

---

El Sistema administra arreglos de casilleros, cada uno de estos cuenta con una señal lumínica que indica el estado del casillero. La luz se encuentra encendida cuando el espacio está disponible. Esta misma luz destella cuando el casillero está en proceso de ser asignado por un tiempo predeterminado, durante el cual el casillero puede ser abierto para ocuparlo. Si este tiempo transcurre sin que se abra la puerta, el casillero queda nuevamente disponible. Cuando la luz está apagada el espacio se encuentra ocupado. Una vez todos los casilleros han sido ocupados, la solicitud de uno nuevo será atendida hasta que uno sea desocupado.

### **El carné**

---

La identificación que se utiliza como llave para acceder a los casilleros es el carné de la Universidad del Valle de Guatemala, que incorpora un código de barras. Al momento de solicitar la asignación o desasignación de un espacio, éste ha de ser deslizado por el lector.

## **El pago**

---

Cuando el servicio de almacenamiento se termina de usar, el usuario debe identificarse nuevamente en el módulo central. El Sistema determinará el pago que debe realizarse de acuerdo a las tarifas preestablecidas y solicitará el monto con un mensaje luminoso. El Sistema acepta monedas de 25 y 50 centavos de quetzal, así como de Q1. Una vez se cancela el monto, el casillero se libera para que se puedan retirar las pertenencias.

*sección OPERARIO*

El Sistema Administrador de Casilleros fue construido prestando atención a la flexibilidad que debe prestarse para acomodarlo fácilmente a cualquier tamaño de implementación. El módulo central es el encargado de llevar el control y atender a los usuarios mientras que los módulos periféricos son los que operan directamente los casilleros. El arranque del Sistema es sumamente sencillo y únicamente se requiere de un cable de comunicación entre módulos. Más adelante se detalla la conexión que tiene que existir entre los casilleros y los módulos periféricos.

## **Conexiones y cables**

---

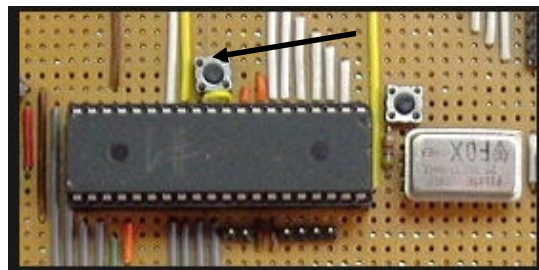
La red I2C que intercomunica al módulo central con los periféricos se establece con un CSA FT1, el cable telefónico de cuatro hilos con terminales RJ11 que se conecta convenientemente a cada circuito. A través de éste viajan las señales de comunicación serial, la alimentación y referencia del periférico, por lo que es necesario que esta conexión sea debidamente realizada antes de arrancar el Sistema.

Los módulos periféricos cuentan con una serie de terminales donde se establecen las conexiones a cada uno de los elementos de los casilleros. Todos aquellos casilleros sobre los que se va a operar deben tener sus elementos relacionados con su respectivo módulo.

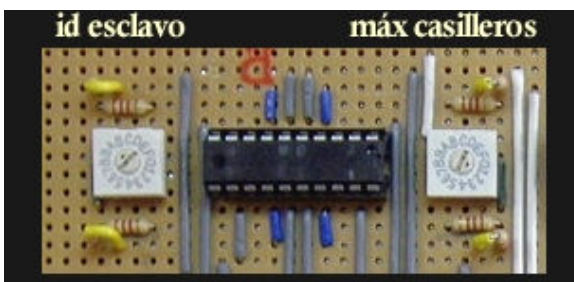
## Inicialización del sistema

---

- *El reloj.* La operación del reloj se encuentra asegurada por una batería que permite su debido funcionamiento aún en la ausencia de alimentación. Si el conteo de la hora se ha mantenido después de una interrupción en el fluido eléctrico, el Sistema procede a operar con ese dato. De lo contrario, el sistema arranca con las 12:00am, en cuyo caso, el botón de ajuste de la hora (ver figura abajo) puede ser presionado para configurarla correctamente. La hora tiene el formato de 24 horas.



- *Los esclavos.* Todos los esclavos deben ser identificados por medio de los interruptores antes de iniciar el Sistema. Es necesario también definir el máximo de casilleros que cada uno de ellos puede operar antes del arranque. La comunicación inicial entre los módulos sincroniza la disponibilidad de casilleros a partir de esta información.



## Manejo de la EEPROM

---

Un integrado en el módulo central mantiene el registro de los casilleros para todo el Sistema. Durante la operación normal, esta memoria se actualiza con cada acción de

asignación y desasignación que ocurre. Si fuera necesario, el módulo central incluye un mecanismo que permite reiniciar por completo la memoria. Debido a la certeza que tiene que existir para realizar esta acción, es necesaria la utilización de una llave. Después de realizada una reinicialización de EEPROM, es necesario apagar el sistema y arrancarlo nuevamente de manera que la sincronización entre módulos ocurra.

### **Apertura manual**

---

Los casilleros incluyen un mecanismo de apertura manual. En caso de emergencia una llave puede abrir la chapa para el servicio del casillero. Es importante mencionar que al realizar esta acción, el módulo periférico puede confundir el estado de cada espacio.

### **Configuración personalizada**

---

Existen varios parámetros que pueden ser personalizados para las aplicaciones. Entre ellos se puede mencionar, la tarifa que por hora o fracción se cobra por el uso del servicio, desde 25 centavos, hasta un máximo de Q 2.50.

En *firmware*, es posible ajustar a los requerimientos específicos el tiempo que el módulo periférico espera para declarar como usado un casillero; tiempo durante el cual la puerta permanece sin seguro para poder depositar las pertenencias.

La velocidad con la que se mueve el mensaje visual es otro parámetro que puede ajustarse de manera sencilla en el propio programa del módulo central.

Para hacer cambios relacionados con los parámetros mencionados anteriormente referirse a las constantes declaradas al inicio de los programas de operación de los microcontroladores.