

**DISEÑO Y CONSTRUCCION DE UNA MAQUINA EXPENDEDORA DE ESTAMPILLAS PARA LA
UNIVERSIDAD DE NARIÑO**

ANGHELO MARINO LOPEZ RUBIO
JAIRO HUGO CERON BASTIDAS

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERIA ELECTRONICA
SAN JUAN DE PASTO
2008

**DISEÑO Y CONSTRUCCION DE UNA MAQUINA EXPENDEDORA DE ESTAMPILLAS PARA LA
UNIVERSIDAD DE NARIÑO**

ANGHELO MARINO LOPEZ RUBIO
JAIRO HUGO CERON BASTIDAS

Trabajo de Grado presentado como requisito parcial para optar por el título de Ingeniero
Electrónico

Director:
DARÍO FERNANDO FAJARDO FAJARDO

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERIA ELECTRONICA
SAN JUAN DE PASTO
2008

“Las ideas y conclusiones aportadas en la tesis de grado, son responsabilidad exclusiva de los autores”

Artículo 1°. Acuerdo No. 324 de octubre 11 de 1966, emanado del Honorable Consejo Directivo de la Universidad de Nariño.

Nota de aceptación

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Fecha: 5 de septiembre de 2008

AGRADECIMIENTOS

A los directivos y administrativos de la Universidad de Nariño, por darnos la oportunidad, el apoyo y el financiamiento necesarios para llevar a cabo el presente proyecto.

Al Ingeniero Darío Fajardo, por su asesoría prestada en el transcurso de la elaboración de este proyecto.

Al plantel de profesores de Ingeniería Electrónica de la Universidad de Nariño, por brindarnos su apoyo y orientarnos de la mejor manera durante nuestro proceso de formación profesional.

A cada uno de nuestros compañeros y amigos, por los consejos recibidos, los buenos momentos compartidos, por el apoyo incondicional que nos brindaron en todo momento y los fuertes vínculos de amistad que nos mantienen unidos aun por fuera de las aulas de clase y esperamos que perduren durante toda nuestra vida.

Al Ingeniero Jesús Erazo, quien aportó nuevas ideas para el desarrollo del proyecto, y a los Ingenieros Julio Cesar Lasso y Rommel Cabrera, quienes además de su amistad y aprecio, nos suministraron datos acerca de nuevas tecnologías adaptables al desarrollo integral del proyecto.

Agradezco a mis Padres por haberme apoyado incondicionalmente, por sus valiosos consejos, su paciencia, su amor, comprensión y confianza en mi persona para llevar a término parte de este proyecto de vida, que es el estudio.

A Mónica y Danna por haberme dado la inigualable oportunidad de compartir mi vida a su lado y por darme la fuerza para luchar por los sueños.

A Gustavo y Viviana, mis hermanos, por haberme brindado su confianza y amistad especial.

A mis Abuelos que con profundo cariño me han acompañado y aconsejado en todos los momentos y etapas de mi vida.

A mi compañero Jairo Hugo Cerón por haberme brindado su sincera amistad y haberme acogido en su hogar como un hermano.

ANGHELO MARINO LOPEZ RUBIO

Dedico este trabajo a mis padres de quienes recibí el regalo más grande como lo es la educación y por todo el apoyo y los buenos consejos que me han brindado durante todo este tiempo.

A mis abuelos por su compañía incondicional y el cariño que siempre me han manifestado.

A mi hermana Adela, por su comprensión y por enseñarme que pase lo que pase siempre tendré alguien a mi lado con quien puedo contar.

A Anghelo López, un excelente compañero y amigo incondicional, del cual siempre recibí el ánimo y el apoyo necesarios para seguir adelante.

A mis buenos amigos, Andrés, Christian, Fabián, Esteban, José, Fabio, compañeros de locuras, con quienes podre contar toda la vida.

JAIRO HUGO CERON BASTIDAS

RESUMEN

Para realizar el diseño y construcción de la máquina expendedora de estampillas para la Universidad de Nariño, se construye un sistema basado en un microcontrolador PIC16F877A el cual se encarga de manejar varios dispositivos periféricos tales como motores, pantalla LCD, pulsadores, memoria EEPROM externa, Reloj Calendario digital en tiempo real RTC y un sistema encargado de sensar el tamaño y material del que esta constituida la moneda para que el usuario pueda obtener en su tiquete aéreo un sello seco o una estampilla autoadhesiva mediante la inserción de una moneda válida de \$200 pesos.

El diseño también comprende la creación de un módulo de almacenamiento y transferencia de información de registro el cual posee un microcontrolador PIC18F2550 que se encarga de leer los datos de funcionamiento de la máquina, almacenarlos en su memoria interna y posteriormente transmitirlos hacia un computador ya sea de escritorio o portátil mediante una conexión USB.

Palabras clave: Microcontrolador, Pantalla LCD, memoria EEPROM, RTC, conexión USB.

ABSTRACT

In order to realize the design and construction of the “Stamp Distributing Machine” for the “Universidad de Nariño”, is built a system based on a microcontroller PIC16F877A, which controls many different peripheral devices such as DC Motors, Display LCD, buttons, External EEPROM, Real Time Clock RTC and a system which sense the diameter and the material of a selected coin. With this system the user of the machine can get a seal in his airport ticket or a sticker only when the users insert a \$200 pesos coin into the system.

The design comprises the creation of a module for storage and transference of information. This module have a microcontroller PIC18f2550 which read the data operation of the machine, storage this data in this internal EEPROM and later transmit this data into a laptop or desktop system, through an USB connection.

Key Words: Microcontroller, Display LCD, DC Motors, EEPROM, RTC, USB connection.

CONTENIDO

1. MARCO CONCEPTUAL	21
1.1. MICROCONTROLADORES	21
1.2 EL BUS SERIE UNIVERSAL USB	22
1.2.1. Características de transmisión.	23
1.3. EL BUS I2C	27
1.3.1. Condiciones de start y stop.	29
1.3.2. Transferencia de datos.....	29
1.3.3. Formato de una transferencia de datos.	30
1.3.4 Temporización del protocolo del bus I2C	30
1.4 ANTECEDENTES	31
2. DESARROLLO DEL TRABAJO	34
2.1 DESCRIPCION GENERAL DE LA MAQUINA	34
2.2 DESCRIPCION GENERAL DEL MODULO DE ALMACENAMIENTO DE INFORMACION DE REGISTRO.	36
2.3 FUENTE DE VOLTAJE D.C.	37
2.4 HARDWARE DE LA MAQUINA DISPENSADORA DE ESTAMPILLAS	38
2.4.1 Implementación del sistema de reconocimiento y validación de monedas.	38
2.4.2. Sistema de reconocimiento y validación de monedas basado en los sensores de tamaño.	40
2.4.3. Sistema electromecánico de distribución de estampillas y sellos secos.	41
2.4.4 Módulo de distribución de estampillas.....	41
2.4.5. Módulo de sellado en relieve o sello seco	42
2.4.6. Sistema de reconocimiento y validación de monedas basado en la detección del material.	44
2.4.7. Partes del comparador de monedas.	45
2.4.8. Bobinas sensoras.....	45
2.4.9. Circuitos internos del comparador de monedas.	47
2.5 EL MICROCONTROLADOR PIC 16F877A.	50
2.5.1 Diagrama y descripción de pines	52

2.5.2 Configuración del oscilador.....	52
2.5.3 Resumen del set instrucciones.	53
2.5.4 Módulo timer0.	54
2.5.5 Módulo I2C del PIC16f877A.	55
2.6 MEMORIA EXTERNA EEPROM SERIAL 24LC512.	56
2.6.1. Direccionamiento como esclavo.....	58
2.6.2 Conexión de la memoria 24LC512 al PIC16F877A.	59
2.6.3 Escritura de la memoria 24LC512	59
2.6.4 Lectura de la memoria 24LC512	60
2.7 RELOJ-CALENDARIO DIGITAL DS1307.	61
2.7.1 Conexión del DS1307 al PIC16F877A.	62
2.7.2. Registros del DS1307.....	63
2.7.3 Escritura en el DS1307.	65
2.7.4. Lectura del DS1307.	65
2.8 PANTALLA LCD ALFANUMÉRICA.....	66
2.8.1 Conexión del LCD L2432 al microcontrolador PIC16F877A	67
2.9 SISTEMA DE MANDOS PARA LA INTERACCION CON EL PERSONAL DE MANEJO Y MANTENIMIENTO.....	67
2.10 DESARROLLO DEL SOFTWARE DEL MICROCONTROLADOR PIC 16F877A DE LA MÁQUINA DE DISTRIBUCIÓN DE ESTAMPILLAS.	69
2.11 HARDWARE DEL MODULO DE ALMACENAMIENTO DE INFORMACION DE REGISTRO.	116
2.11.1 Microcontrolador PIC18F2550.....	118
2.12 SOFTWARE DEL MICROCONTROLADOR (MÓDULO DE ALMACENAMIENTO DE INFORMACIÓN DE REGISTRO).....	125
2.13 SOFTWARE DE RECEPCION DE DATOS DEL MODULO (ALMACENAMIENTO DE INFORMACION EN PC)	134
3. CONCLUSIONES.....	136
RECOMENDACIONES	138
BIBLIOGRAFIA.....	139

LISTA DE TABLAS

Tabla 1. Pines de conexión del puerto USB	23
Tabla 2. Características eléctricas de la fuente ATX.	37
Tabla 3. Set de Instrucciones del PIC16F877.	54
Tabla 4. Frecuencia de la señal cuadrada del DS1307.	64
Tabla 5. Valores del Configuration Word del PIC16F877A de la máquina.	71
Tabla 6. Configuración del registro ADCON1.....	73
Tabla 8. Configuración del registro OPTION_REG.....	73
Tabla 8. Registro SSPCON	77
Tabla 9. Configuración del registro CONFIG1L.....	121
Tabla 10. Configuración del registro UCON.....	122
Tabla 11. Configuración del registro UCFG.....	122

LISTA DE FIGURAS

Figura 1. Estructura de capas del bus USB.....	24
Figura 2. Flujo de comunicaciones entre dispositivo USB y el host.....	25
Figura 3. Estructura del Bus I2C.	28
Figura 4. Condiciones de start y stop.	29
Figura 5. Transferencia de datos.....	30
Figura 6. Formato de una transferencia de datos.....	30
Figura 7. Definición de los tiempos para el bus I2C.	31
Figura 8. Los ATMs en Grecia (azul), Marruecos (amarillo) y Portugal (rojos).....	32
Figura 9. Diagrama de bloques de la máquina dispensadora de estampillas.	35
Figura 10. Diagrama de bloques del módulo de almacenamiento de información	37
Figura 11. Diagrama de bloques del sistema de reconocimiento y validación de monedas.	38
Figura 12. Esquema del sensor óptico del sistema mecánico de recepción de monedas y su acople de señal al Microcontrolador.	39
Figura 13. Señal del Sensor óptico presente en el sistema mecánico de recepción de monedas.	39
Figura 14. Aspecto real del arreglo de sensores ópticos.....	40
Figura 15. Aspecto real del módulo de distribución de estampillas.....	42
Figura 16. Aspecto real módulo de sellado en relieve.	43
Figura 17. Coin Comparitor CC-16.d.100.	44
Figura 18. Identificación de partes del COIN COMPARITOR CC-16.D.100.....	46
Figura 19. Aspecto real y esquema de las bobinas sensoras del sistema comparador de moneda.	47
Figura 20. Disposición de las bobinas del sistema sensor de moneda.	47
Figura 21. Diagrama de bloques del circuito interno del sistema sensor de moneda.	48
Figura 22. Oscilador para las bobinas del sensor de moneda y su señal de salida.	48
Figura 23. Señal de la bobina de detección del sensor de moneda.....	49
Figura 24. Diagrama de pines del PIC16F877A.	52
Figura 25. Cristal Externo como oscilador.	53

Figura 26. Memoria EEPROM Serial 24LC512.	57
Figura 27. Diagrama de bloques de la EEPROM Serial 24LC512.	58
Figura 28. Direccionamiento de la memoria 24LC512 como esclavo en Bus I2C.	59
Figura 29. Conexión de una memoria EEPROM serie 24LC512 al PIC16F877A.	59
Figura 30. Protocolo de escritura en la memoria EEPROM serie 24LC512.	60
Figura 31. Protocolo de escritura en la memoria EEPROM serie 24LC512.	61
Figura 32. Reloj-Calendarario Digital DS1307.	62
Figura 33. Conexión del PIC16F877A al RTC DS1307.	62
Figura 34. Mapa de Memoria del DS1307.	63
Figura 35. Registros de mayor importancia del DS1307.	64
Figura 36. Registro de control de la señal cuadrada del DS1307.	64
Figura 37. Protocolo de escritura del microcontrolador maestro sobre el DS1307.	65
Figura 38. Protocolo de lectura del DS1307 por parte del microcontrolador maestro. ...	66
Figura 39. Aspecto real del display L2432.	67
Figura 40. Patillaje del LCD L2432.	67
Figura 41. Conexión del PIC16F877A al LCD L2432.	68
Figura 42. Mensaje asignado para la función borrar memoria.	68
Figura 43. Ajuste de hora y fecha.	69
Figura 44. Diagrama del sistema de mandos para el personal de mantenimiento.	70
Figura 45. Diagrama de Flujo de la Rutina Principal.	72
Figura 46. Diagrama de flujo de la subrutina "Grabar mensajes en la memoria EEPROM serial externa".	75
Figura 47. Diagrama de flujo de la subrutina EscribeDato_M24LC512.	76
Figura 48. Diagrama de flujo de la subrutina "Config_transmision_I2C"	77
Figura 49. Diagrama de flujo de la subrutina "Bit_start_I2C"	78
Figura 50. Diagrama de flujo de la subrutina "Transmitir_dato_I2C"	79
Figura 51. Diagrama de flujo de la subrutina "Bit_stop_I2C"	80
Figura 52. Diagrama de flujo de la subrutina "Inicializar Pantalla LCD"	80
Figura 53. Diagrama de flujo de la subrutina "Mensaje inicial a LCD".	82
Figura 54. Diagrama de flujo de la subrutina "LeeDato_M24LC512".	83

Figura 55. Diagrama de flujo de la subrutina “ReStart_I2C”	84
Figura 56. Diagrama de flujo de la subrutina “Recibir_dato_I2C”	85
Figura 57. Diagrama de flujo de la subrutina “LCD_Caracter”	86
Figura 58. Diagrama de flujo de la subrutina “LCD_EscribeLCD”	88
Figura 59. Diagrama de flujo para la subrutina “Inicia_ConteoMoneda”	91
Figura 60. Diagrama de flujo para la subrutina “DetectaValor”	93
Figura 61. Diagrama de flujo para la subrutina “DetectaValor”	94
Figura 62. Diagrama de flujo para la subrutina “DetectaValor”	95
Figura 63. Diagrama de flujo para la subrutina “EraDe50pe”	96
Figura 64. Diagrama de flujo para la subrutina “EraDe100pe”	97
Figura 65. Diagrama de flujo para la subrutina “EraDe500pe”	98
Figura 66. Diagrama de flujo para la subrutina “EraDeOtro”	99
Figura 67. Diagrama de flujo para la subrutina “EraDe200pe”	100
Figura 68. Diagrama de flujo para las subrutinas “Inicia_TMR0” y “Rutina_TMR0”	101
Figura 69. Diagrama de flujo para la subrutina “Mens200falsa”	102
Figura 70. Diagrama de flujo para la subrutina “Mens200Buena”	103
Figura 71. Diagrama de flujo para la subrutina “Motor Sello”	105
Figura 72. Diagrama de flujo para la subrutina “Motor Estampilla”	106
Figura 73. Diagrama de flujo para la subrutina “BORRAR MEMORIA/AJUSTAR HORA/LEER DATOS”	108
Figura 74. Diagrama de flujo para la subrutina “LlenaCerosEE”	109
Figura 75. Diagrama de flujo para la subrutina “Ajustar_Hora”	112
Figura 76. Diagrama de flujo para la subrutina “AjustaAnho”	113
Figura 77. Diagrama de flujo para la subrutina “DS1307_Escribe”	114
Figura 78. Diagrama de flujo para la subrutina “LeerReloj”	115
Figura 79. Diagrama de flujo para la subrutina “MostrarReloj”	116
Figura 80. Conexión del PIC18F2550 con la memoria EEPROM 24LC512 y el reloj DS1307.	117
Figura 81. Diagrama de pines del PIC 18F2550	119
Figura 82. Diagrama de bloques del módulo oscilador del PIC18F2550	120

Figura 83. Conexión del oscilador a cristal en el microcontrolador.	121
Figura 84. Diagrama de bloques del periférico USB.....	123
Figura 85. Diagrama de flujo de la rutina principal del PIC 18F2550 presente en el módulo de almacenamiento y transferencia de información de registro.	127
Figura 86. Diagrama de flujo de la subrutina “Leer_Relej_Memoria”.	128
Figura 87. Diagrama de flujo de la subrutina “Borrar EEPROM del PIC”.	129
Figura 88. Diagrama de flujo de la subrutina “Enviar Datos al PC”.	129
Figura 89. Instalación de nuevo hardware (Máquina de Estampillas).....	132
Figura 90. Instalación de nuevo hardware.	132
Figura 91. Instalación de nuevo hardware.	133
Figura 92. Instalación de nuevo hardware.	133
Figura 93. Instalación de nuevo hardware.	134
Figura 94. Interfaz software transmisión datos PC.....	135

LISTA DE ANEXOS

ANEXO 1. COMPARADOR DE MONEDA.....	141
ANEXO 2. PANTALLA LCD ALFANUMERICA	143

GLOSARIO

ACKNOWLEDGE: Bit de reconocimiento utilizado durante la comunicación a través del bus I2C.

BUS I2C: Inter-Integrated Circuits. Bus de comunicación serial.

BUS USB: Universal Serial Bus. Puerto que sirve para conectar diferentes dispositivos periféricos a una PC.

CONFIGURATION WORD: Palabra de configuración del PIC16F877A.

DATASHEET: Hoja de datos.

DRIVER: programa que interactúa con un dispositivo en particular. Contiene todos los datos necesarios del dispositivo con el que se entrega para que el resto de programas sepan como utilizarlo.

EEPROM: Electrically Erasable PROM. Memoria no volátil borrable eléctricamente.

ESCLAVO: Dispositivo secundario dentro de una comunicación.

FIRMWARE: o Programación en Firme, es un bloque de instrucciones de programa para propósitos específicos, grabado en una memoria tipo ROM, que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo. Al estar integrado en la electrónica del dispositivo es en parte hardware, pero también es software, ya que proporciona lógica y se dispone en algún tipo de lenguaje de programación. Funcionalmente, el firmware es el intermediario (interfaz) entre las órdenes externas que recibe el dispositivo y su electrónica, ya que es el encargado de controlar a ésta última para ejecutar correctamente dichas órdenes externas.

HOST: Dispositivo Maestro dentro de una comunicación o intercambio de datos.

LCD: Liquid Crystal Display. Display o pantalla de cristal líquido.

LED: Light Emitting Diode, diodo emisor de Luz.

MICROCONTROLADOR: es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de E/S, es decir, se trata de un computador completo en un solo circuito integrado.

MSCOMM: El control MScComm proporciona comunicaciones serie para que su aplicación pueda transmitir y recibir datos a través de un puerto serie.

NIBBLE: segmento de palabra de 4 bits.

PC: Personal Computer. Computadora personal.

PIC: (Peripheral Interface Controller), Controlador de interfaz de periféricos.

PUERTO: Punto de conexión regulado por diversos parámetros que permite conexión entre dispositivos.

REGISTRO: posición de memoria rápida por lo general de 8 bits para almacenamiento de datos de distinta índole dentro de un microcontrolador.

RTC: Real Time Clock. Reloj-Calendarario en tiempo real.

SENSAR: equivale al verbo detectar.

SET: Fijar, programar, dar un valor.

SET DE INSTRUCCIONES: Conjunto de instrucciones.

TESTEAR: Probar, medir, detectar el valor actual.

TIMERO: Módulo funcional del PIC16F877A, utilizado principalmente para realizar temporizaciones exactas.

INTRODUCCION

El área de Ingeniería Electrónica siempre se encuentra con desafíos de diseño relacionados con la creación de sistemas que se encarguen de realizar tareas repetitivas durante largos periodos de tiempo y que además de ello se encuentren continuamente realizando labores de control y registro de su funcionamiento.

En la actualidad la Universidad de Nariño posee una fuente de ingresos de gran importancia para sus finanzas y se relaciona con la venta de una estampilla que es uno de los requisitos para que una persona pueda acceder a su tiquete aéreo. El problema se genera en que la forma de cobro y distribución de esta estampilla o sello se hace de manera personal y manual, es decir, se requiere la presencia de una persona en todo momento para que un determinado pasajero pueda pagar este impuesto. Como es de esperarse la constante presencia de una persona es imposible y genera más gastos de los que se producen mediante el cobro del impuesto.

Esta situación hace que se proponga una solución al presente problema mediante la creación de una máquina autónoma capaz de realizar la distribución de sellos y estampillas cuando el usuario inserte una moneda dentro de ella, con lo cual el sistema habilita la entrega de la estampilla o sella el tiquete aéreo del pasajero.

También se plantea que la máquina sea capaz de almacenar un registro de su funcionamiento durante un periodo de tiempo determinado y que este sea accesible desde un sistema tal como una PC.

En el presente documento se encuentra el proceso que se sigue durante la elaboración del diseño y construcción tanto de la máquina dispensadora de estampillas como del módulo de almacenamiento de registro. Se detalla datos técnicos, dispositivos utilizados y se describe el software de control. La construcción de esta máquina da solución al problema de fondo y elimina la necesidad de la presencia de una persona permanentemente desempeñando una labor repetitiva. Por el contrario, al tener las condiciones correctas la máquina es capaz de realizar la tarea para la cual fue diseñada por si sola.

1. MARCO CONCEPTUAL

1.1. MICROCONTROLADORES

Un microcontrolador PIC es un circuito integrado programable que contiene todos los componentes necesarios para controlar el funcionamiento de una tarea determinada, como el control de una lavadora, un teclado de computador, una impresora, un sistema de alarma, etc. Para esto, el microcontrolador utiliza muy pocos componentes asociados. Un sistema con microcontrolador dispone de una memoria donde se almacena el programa que gobierna el funcionamiento del mismo que una vez programado y configurado, sirve para realizar la tarea programada. La utilización de un microcontrolador en un circuito reduce notablemente el tamaño y el número de componentes, y en consecuencia, disminuye el número de averías y el volumen y el peso de los equipos, entre otras ventajas¹.

El microcontrolador es uno de los desarrollos técnicos más notables del siglo XX. En el mercado hay una gran cantidad de ellos, con multitud de posibilidades y características. Cada tipo de microcontrolador sirve para una serie de casos y es el diseñador del sistema quien debe definir cual es el microcontrolador más idóneo para cada uso.

En los últimos años han tenido un gran auge los microcontroladores PIC fabricados por Microchip Technology Inc. Los PIC (Peripheral Interface Controller) son una familia de microcontroladores que ha tenido gran aceptación y desarrollo en los últimos años gracias a que sus buenas características, bajo precio, reducido consumo, pequeño tamaño, gran calidad, fiabilidad y abundancia de información, lo convierten en muy fácil, cómodo y rápido de utilizar.

Esta última arquitectura es la dominante en los microcontroladores actuales ya que la memoria de programa es usualmente ROM, EPROM o FLASH mientras que la memoria de datos es usualmente RAM. Consecuentemente, las tablas de datos pueden estar en la memoria de programa sin que sean perdidas cada vez que el sistema es apagado. Otra ventaja importante en la arquitectura Harvard modificada es que las transferencias de datos pueden ser solapadas con los ciclos de decodificación de instrucciones. Esto quiere decir que la siguiente instrucción puede ser cargada de la memoria de programa mientras se está ejecutando una instrucción que accede a la memoria de datos. La desventaja de la arquitectura Harvard modificada es que se requiere instrucciones especiales para acceder a valores en memoria RAM y ROM haciendo la programación un poco complicada.

¹ Enrique Palacios Municio. Microcontrolador PIC 16F84 Desarrollo de Proyectos. AlfaOmega. 2004.

En cuanto a las técnicas de fabricación, prácticamente la totalidad de los microcontroladores actuales se fabrican con tecnología CMOS (Complementary Metal Oxide Semiconductor). Esta tecnología supera a las técnicas anteriores por su bajo consumo y alta inmunidad al ruido.

Algunos de los principales fabricantes de microcontroladores son Microchip, Motorola, Intel, Atmel, Siemens, Philips, Hitachi o Nacional Semiconductor, entre otros. De entre todos los fabricantes expuestos, Microchip es el que más diversidad posee.

1.2 EL BUS SERIE UNIVERSAL USB

El Universal Serial Bus (bus universal en serie) es un puerto que sirve para conectar periféricos a una computadora. Fue creado en 1996 por siete empresas: IBM, Intel, Northern Telecom, Compaq, Microsoft, Digital Equipment Corporation y NEC.

El estándar incluye la transmisión de energía eléctrica al dispositivo conectado. Algunos dispositivos requieren una potencia mínima, así que se puede conectar varios sin necesitar fuentes de alimentación extra. La mayoría de los concentradores incluyen fuentes de alimentación que brindan energía a los dispositivos conectados a ellos, pero algunos dispositivos consumen tanta energía que necesitan su propia fuente de alimentación. Los concentradores con fuente de alimentación pueden proporcionarle corriente eléctrica a otros dispositivos sin quitarle corriente al resto de la conexión (dentro de ciertos límites).

El diseño del USB tenía en mente eliminar la necesidad de adquirir tarjetas separadas para poner en los puertos bus ISA o PCI, y mejorar las capacidades plug-and-play permitiendo a esos dispositivos ser conectados o desconectados al sistema sin necesidad de reiniciar. Cuando se conecta un nuevo dispositivo, el servidor lo enumera y agrega el software necesario para que pueda funcionar.

El USB puede conectar los periféricos como mouse, teclados, escáneres, cámaras digitales, teléfonos celulares, reproductores multimedia, impresoras, discos duros externos, tarjetas de sonido, sistemas de adquisición de datos y componentes de red. Para dispositivos multimedia como escáneres y cámaras digitales, el USB se ha convertido en el método estándar de conexión. Para impresoras, el USB ha crecido tanto en popularidad que ha empezado a desplazar a los puertos paralelos porque el USB hace sencillo el poder agregar más de una impresora a una computadora personal².

² USB, BUS SERIE UNIVERSAL [Online]. http://es.wikipedia.org/wiki/Bus_de_Serie_Universal.

1.2.1. Características de transmisión. Los dispositivos USB se clasifican en cuatro tipos según su velocidad de transferencia de datos:

- **Baja Velocidad (1.0):** Bitrate de 1.5Mbit/s (192KB/s). Utilizado en su mayor parte por Dispositivos de Interfaz Humana (HID) como los teclados, los ratones y los joysticks.
- **Velocidad Completa (1.1):** Bitrate de 12Mbit/s (1.5MB/s). Esta fue la más rápida antes de que se especificara la USB 2.0 y muchos dispositivos fabricados en la actualidad trabajan a esta velocidad. Estos dispositivos, dividen el ancho de banda de la conexión USB entre ellos basados en un algoritmo FIFO.
- **Alta Velocidad (2.0):** Bitrate de 480Mbit/s (60MB/s).
- **Súper Velocidad (3.0)** Actualmente en fase experimental. Bitrate de 4.8Gbit/s (600MB/s). Esta especificación será lanzada a mediados de 2008 por la compañía Intel, de acuerdo a información recabada de Internet. Las velocidades de los buses serán 10 veces más rápidas que la de USB 2.0 debido a la inclusión de un enlace de fibra óptica que trabaja con los conectores tradicionales de cobre. Se espera que los productos fabricados con esta tecnología lleguen al consumidor en 2009 o 2010.

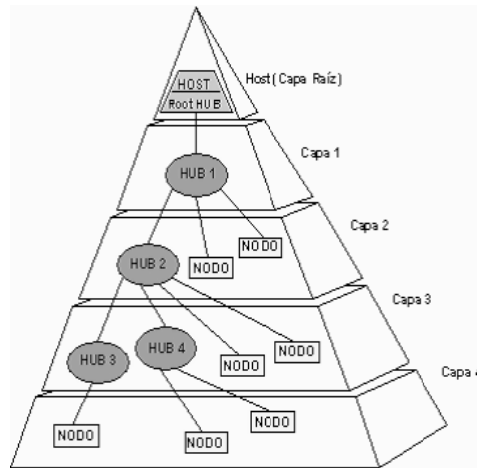
Las señales del USB son transmitidas en un cable de data par trenzado con impedancia de $90\Omega \pm 15\%$ llamados D+ y D-. Éstos, colectivamente utilizan señalización diferencial en half dúplex para combatir los efectos del ruido electromagnético en enlaces largos. D+ y D- usualmente operan en conjunto y no son conexiones simplex. Los niveles de transmisión de la señal varían de 0 a 0.3V para bajos (ceros) y de 2.8 a 3.6V para altos (unos) en las versiones 1.0 y 1.1, y en $\pm 400\text{mV}$ en Alta Velocidad (2.0). En las primeras versiones, los alambres de los cables no están conectados a masa, pero en el modo de alta velocidad se tiene una terminación de 45Ω a tierra o un diferencial de 90Ω para acoplar la impedancia del cable.

Tabla 1. Pines de conexión del puerto USB

Pin	Nombre	Color del Cable	Descripción
1	VCC	Rojo	+5V
2	D-	Blanco	Data -
3	D+	Verde	Data +
4	GND	Negro	Tierra

El estándar USB define una topología de conexión en estrella, tal como se muestra en la figura 1, por medio de la incorporación de varios concentradores (*hubs*) conectados en serie. Cada concentrador se conecta por un lado al computador, que contiene una o dos interfaces de este tipo en la placa base, o a otro concentrador y, por otro lado, se conecta a varios dispositivos o incluso a otro concentrador. De este modo pueden existir periféricos que vengan ya preparados con nuevos conectores USB para incorporar nuevos dispositivos, hasta un total de 127, todos ellos funcionando simultáneamente.

Figura 1. Estructura de capas del bus USB



Fuente: *Universal Serial Bus Revision 2.0 specification [Online]. www.usb.org.*

La topología del bus USB se puede dividir en tres partes.

- *Capa física:* Conexión física de los elementos.
- *Capa lógica:* Las funciones y tareas de los elementos USB.
- *Relación software del cliente-función:* Como es la interacción del software y los elementos o interfaces.

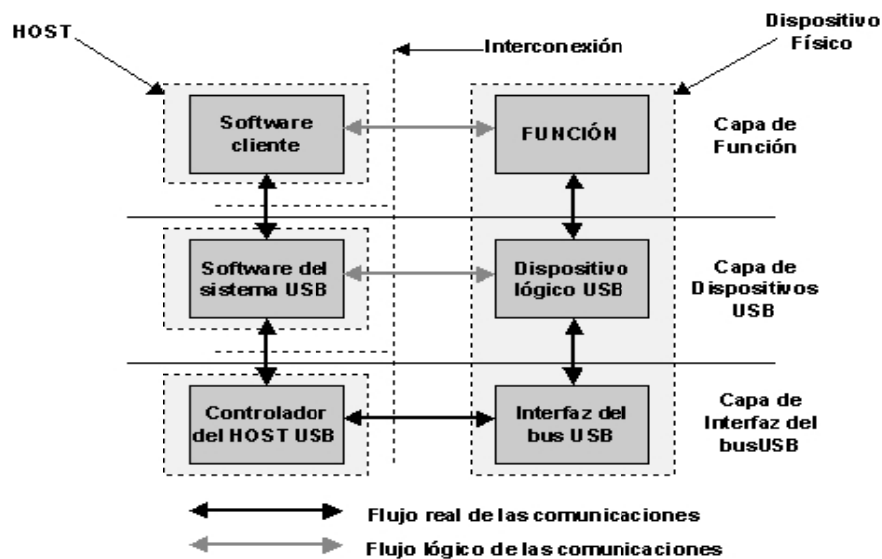
El flujo de datos del bus USB desde un punto de vista lógico hay que entenderlo como una serie de puntos finales (*endpoints*), que se agrupan en conjuntos que dan lugar a interfaces, las cuales permiten controlar la función del dispositivo.

La arquitectura se divide en tres niveles o capas. En el nivel más bajo el controlador de host USB se comunica con la interfaz del bus utilizando el cable USB, mientras que en un nivel superior el software USB del sistema se comunica con el dispositivo lógico utilizando la tubería de control por defecto. En lo que al nivel de función se refiere, el software

cliente establece la comunicación con las interfaces de la función a través de tuberías asociadas a puntos finales.

Cada dispositivo USB está compuesto por unos puntos finales independientes y una dirección única asignada por el sistema en tiempo de conexión, de forma dinámica. A su vez, cada punto final dispone de un identificador único dentro del dispositivo (número de endpoint) que viene asignado de fábrica, además de una determinada orientación del flujo de datos. Cada punto final es por si solo una conexión simple, que soporta un flujo de datos de entrada o de salida.

Figura 2. Flujo de comunicaciones entre dispositivo USB y el host.



Fuente: Universal Serial Bus Revision 2.0 specification [Online]. www.usb.org.

Una tubería USB es una conexión lógica entre un punto final del periférico y el software del host, que permite intercambiar datos entre ellos. La tubería que esta formada por el punto final se denomina tubería de control por defecto. Esta tubería está siempre disponible una vez se ha conectado el dispositivo y ha recibido un reset del bus. El resto de tuberías aparecen después de que se configure el dispositivo. La tubería de control por defecto es utilizada por el software USB del sistema para obtener la identificación y para configurar al periférico.

El host USB tiene las funciones de:

- Detectar la conexión/desconexión de dispositivos y configurarlos.
- Controlar las transferencias de datos y de control que tienen lugar en el bus.

- Realización de auditorías sobre la actividad del sistema.
- Servir como fuente de alimentación a los dispositivos.

El USB define dos líneas para transmitir datos y otras dos para transmitir potencia. Los datos se transmiten de forma balanceada a velocidades entre 1,5 Mbps y 12 Mbps. La señal se envía codificada en un código auto reloj de no retorno a cero invertido (NRZI) para poder incluir junto con los datos información de sincronización.

Debido a que cadenas de unos pueden producir largos periodos en los que la señal no cambia dando lugar a problemas de sincronización, se introducen los bits de relleno. Cada 6 bits consecutivos a "1" se inserta un bit a "0" para forzar un cambio, de esta forma el receptor puede volverse a sincronizar. El relleno bits empieza con el patrón de señal Sync. El "1" que finaliza el patrón de señal Sync es el primer uno en la posible primera secuencia de seis unos.

En las señales a velocidad media o baja, el relleno de bits se utiliza a lo largo de todo el paquete sin excepción. De modo que un paquete con siete unos consecutivos será considerado un error y por lo tanto ignorado. En el caso de la velocidad alta se aplica el relleno de bits a lo largo del paquete, con la excepción de los bits intencionados de error usados en EOP a velocidad alta.

El computador identifica automáticamente el dispositivo que se conecta mientras opera y lo configura sin tener que instalar drivers específicos del fabricante. Al comienzo se detectan los dispositivos conectados midiendo los niveles de voltaje de las líneas. Si un dispositivo está conectado, entonces envía información sobre el tipo o la clase a la que pertenece, qué modo de transferencia utilizará y cuáles son sus necesidades de ancho de banda. El host reconoce el dispositivo buscando en la lista de drivers del sistema operativo y teniendo en cuenta los demás dispositivos conectados le asigna un ancho de banda determinado. De la misma forma también se puede desconectar los dispositivos del sistema. El controlador USB del host asigna un número diferente de dispositivo a cada uno de los periféricos que se conectan a este bus. Para empezar la transferencia, éste envía un paquete que identifica al dispositivo objeto de la transferencia. El protocolo soporta cuatro tipos de transferencias.

Control. Son transferencias que se utilizan para leer información de los descriptores en los registros de los dispositivos (llamados *endpoints*), interpretarla y poder configurarlos.

- **Interrupción.** Usadas en los periféricos del tipo de los controladores de juegos, teclados y ratones, cuya comunicación es unidireccional y poco frecuente.
- **Masiva.** Son transferencias no periódicas que precisan de todo el ancho de banda disponible. Utilizadas por las impresoras y los scanners.

- **Isócrona.** Dedicadas a las transferencias de telecomunicaciones, como voz o vídeo, que garantiza unas tasas de transferencia constantes. Se caracterizan porque el número de pulsos de reloj que transcurren entre la transmisión de dos caracteres es constante, por lo tanto, se está enviando información constantemente entre el host y el dispositivo.

1.3. EL BUS I2C

El bus I2C es un estándar de fábrica que actualmente se encuentra implementado por más de 1000 circuitos integrados de más de 50 compañías. Actualmente numerosos proyectos que utilizan microcontroladores de 8 bits suelen regirse por criterios de diseños muy reiterados:

- Sistema que consta de al menos un microcontrolador y varios dispositivos periféricos como memorias, LCD, convertidores ADC, etc.
- La conexión entre los distintos dispositivos que componen el sistema deben ser fácil de realizar y su costo mínimo.
- Estos sistemas suelen realizar funciones que no requieren una alta tasa de transferencia de datos, generalmente no superior a 100Kbps.
- El sistema no debe depender de los dispositivos conectados a él. De otro modo no sería posible realizar modificaciones o mejoras.

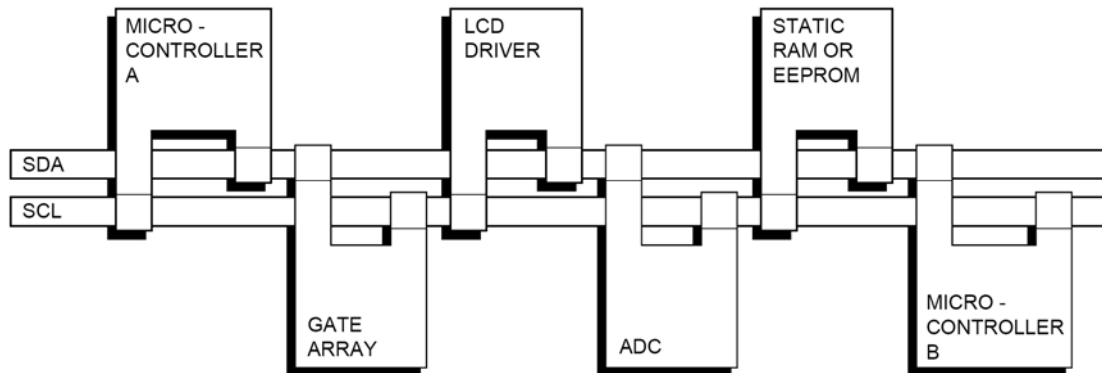
Para implementar un sistema que satisfaga estos criterios se necesita una estructura de bus serie, ya que aunque no tiene ni la misma capacidad ni velocidad de los buses paralelos, requiere poco hardware y un mínimo de cableado. Este bus serie no debe ser simplemente un hilo de conexión de datos, debe incorporar una serie de procedimientos o protocolos para la correcta comunicación entre los componentes del sistema.

Todos estos criterios sirven de base para el fundamento del bus serie I2C para la conexión de circuitos integrados o I2C (Inter Integrated Circuit Bus), desarrollado por Philips Semiconductors.

El bus I2C esta formado por dos hilos que pueden conectar varios dispositivos mediante un hardware muy simple. Esto se puede observar en la figura 3. Por esos dos hilos se produce una comunicación serie, bit a bit. Se transmiten dos señales, una por cada línea:

- **SCL:** (Serial Clock). Es la señal de reloj que se utiliza para la sincronización de los datos.
- **SDA:** (Serial Data). Es la línea para la transferencia serie de los datos.

Figura 3. Estructura del Bus I2C.



Fuente: I2C-Bus Specification and User Manual [Online]. Philips Semiconductors.

Los dispositivos conectados al bus I2C mantienen un protocolo de comunicaciones del tipo maestro/esclavo. Las funciones del maestro y del esclavo se diferencian en:

- El circuito maestro inicia y termina la transferencia de información, además de controlar la señal de reloj.
- El esclavo es un circuito direccionado por el maestro.

La línea SDA es bidireccional, es decir, tanto el maestro como los esclavos pueden actuar como transmisores o receptores de datos, dependiendo de la función del dispositivo. La generación de señales de reloj siempre es responsabilidad del maestro.

Cada dispositivo conectado al bus I2C es reconocido por una única dirección que lo diferencia de los demás circuitos conectados. Los dispositivos compatibles con bus I2C suelen llevar 2 o 3 pines para modificar esta dirección de modo que en el bus no existan dos módulos con la misma dirección.

El hardware del bus I2C se basa en la AND Cableada, las dos líneas SDA y SCL tienen resistencias de Pull-Up cuyo valor depende de la tensión de alimentación, la capacidad del bus y el número de dispositivos conectado, etc.

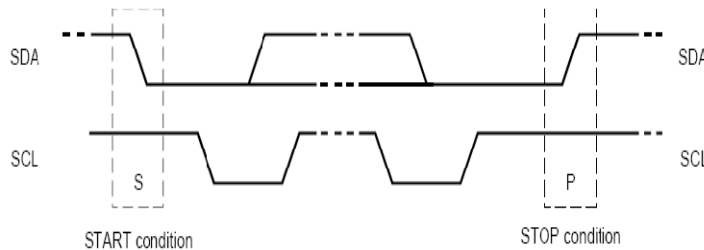
Se pueden transferir datos por el bus a una velocidad máxima de 100K bits por segundo (modo estándar). El número de dispositivos conectados al bus viene limitado por la capacidad máxima admitida que es de 400pF.

1.3.1. Condiciones de start y stop. Para que la transferencia de información pueda ser iniciada el bus no debe estar ocupado. Para indicar que el bus esta libre las líneas SCL y SDA deben estar e nivel alto, una vez verificado esto el transmisor procederá a enviar un bit cada pulso de reloj.

En la transmisión de datos se dan los siguientes pasos:

- Condición de Start: SDA tiene que estar en flanco de bajada mientras que SCL permanece a nivel alto. Esta condición señala el inicio de la transmisión de datos.
- Transmisión de bits de información a cada pulso de reloj.
- Condición de Stop: SDA debe estar en flanco de subida mientras que SCL permanece en nivel alto, esta condición es el fin de la transferencia.

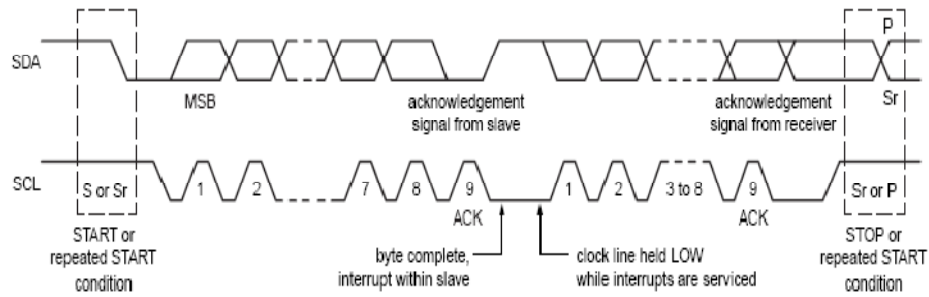
Figura 4. Condiciones de start y stop.



Fuente: I2C-Bus Specification and User Manual [Online]. Philips Semiconductors

1.3.2. Transferencia de datos. Cada dato enviado por la línea SDA debe tener 8 bits (un byte). El número de bytes que se puede enviar no tiene restricción. El byte de datos se transfiere empezando por el bit 7 que es el de mayor peso. Una vez se han enviado los 8 bits el receptor debe enviar un bit de reconocimiento o Acknowledge en el noveno pulso de reloj. El receptor ejecuta esta señal poniendo la línea SDA en bajo. Este bit es obligatorio después de cada transferencia de 8 bits

Figura 5. Transferencia de datos.

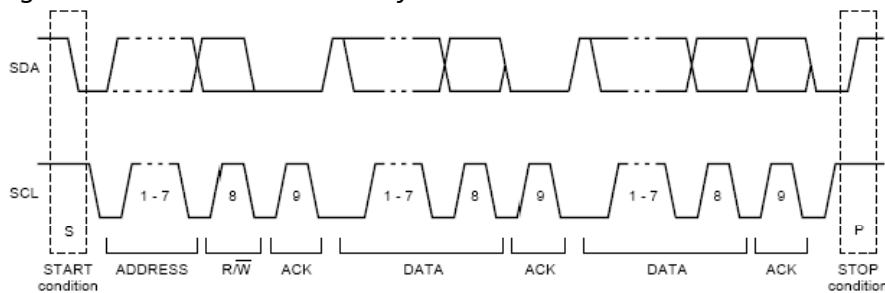


Fuente: *I2C-Bus Specification and User Manual [Online]. Philips Semiconductors*

1.3.3. Formato de una transferencia de datos. Los datos transferidos tienen la forma de la figura 6, donde se observa que para operar un esclavo sobre el bus I2C se requieren seis pasos principales que son:

- a) Un bit de Start que señala el inicio de la transferencia de datos.
- b) Siete bits de direccionamiento de un esclavo.
- c) Un bit de lectura/escritura (R/W) que define si el esclavo es transmisor o receptor.
- d) Un bit de reconocimiento ACK (Acknowledge).
- e) Un mensaje dividido en Bytes (8bits).
- f) Un bit de Stop que indica el fin de la transmisión.

Figura 6. Formato de una transferencia de datos.



Fuente: *I2C-Bus Specification and User Manual [Online]. Philips Semiconductors*

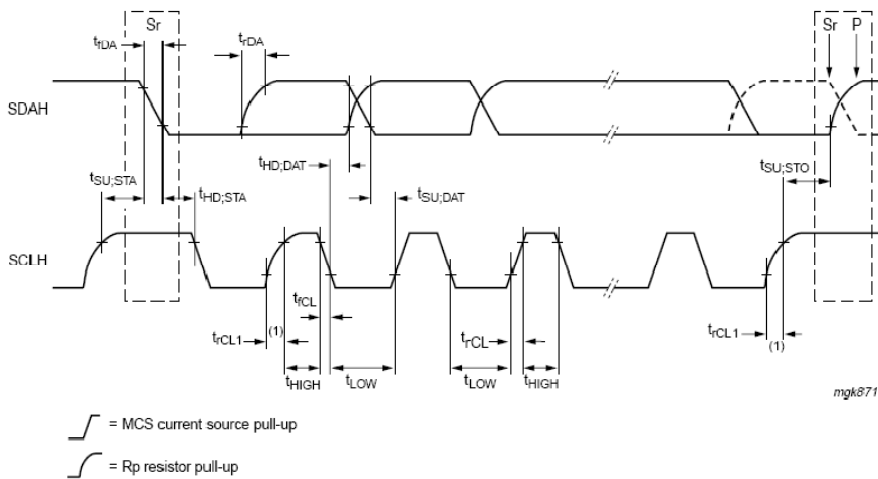
1.3.4 Temporización del protocolo del bus I2C. Los tipos de transferencia de datos en el bus se clasifican en:

- **Standard mode:** Modo estándar, aproximadamente a 100 Kbits por segundo.
- **Fast mode:** Modo rápido, aproximadamente a 400Kb/s.
- **High-Speed Mode:** Modo alta velocidad, mas de 3.4 Mb/s.

El modo estándar es el más común y sus tiempos requeridos vienen reflejados en la figura 6. Los valores mínimos que pueden tener estos tiempos son:

- $t_{HIGH} > 4 \mu s$. Duración del semiperiodo alto del reloj SCL.
- $t_{LOW} > 4.7 \mu s$. Duración del semiperiodo bajo del reloj SCL.
- $t_{HD:STA} > 4 \mu s$. En condición de Start, tiempo que tiene que transcurrir entre el flanco de bajada de la línea SDA y el flanco de bajada de la línea SCL.
- $t_{SU:STO} > 4.7 \mu s$. En la condición de Stop, tiempo entre el flanco de subida de la línea SCL y el flanco de subida de la línea SDA.
- $t_{HD:DAT} > 5 \mu s$. Tiempo de mantenimiento del dato. Tiempo entre el flanco de bajada del reloj SCL y el cambio del dato en la línea SDA.
- $t_{SU:DAT} > 250 ns$. Tiempo de puesta del dato. Tiempo entre el cambio de datos en la línea SDA y el próximo flanco de subida del reloj SCL. Se suele despreciar.
- $t_{BUF} > 4.7 \mu s$. Tiempo en que el bus debe estar libre antes de comenzar una nueva transmisión.

Figura 7. Definición de los tiempos para el bus I2C.



Fuente: I2C-Bus Specification and User Manual [Online]. Philips Semiconductors.

1.4 ANTECEDENTES

Las estampillas, son un tipo de sello que contiene un determinado diseño y que sirve para ser estampado o adherido a un documento con la intención de recaudar un impuesto. Esta idea fue desarrollada en el año de 1837 en Gran Bretaña por el profesor Rowland Hill y es un método de recaudo que ha sobrevivido hasta nuestros días.

En la actualidad no existen referencias sobre trabajos que se haya realizado con objetivos similares a los planteados dentro de este proyecto en las distintas instituciones de la región y del país, sin embargo se cuenta con información de algunas empresas europeas, tales como la francesa AMIEL ELECTRONIX desarrolladora de dispensadores de estampillas y sobres que presentan, junto con la unidad de distribución y pago, teclas de función directas fijas y una bascula electrónica, la portuguesa NEWVISION cuyos diseños se basan en un sistema de tipo modular, que cuentan a la vez con funciones como dispensador de sobres, buzón en su base, bascula digital y dispensador de estampillas de diferentes denominaciones; y la española EPELSA quienes adicionaron a los sistemas básicos presentados por los anteriores diseñadores una impresora de tipo térmico y un rollo de soportes adhesivos tales como los que se puede encontrar en las balanzas digitales de los supermercados. Estas empresas en general desarrollan prototipos de máquinas y sistemas para ser utilizados en las oficinas postales de diferentes países como España y Portugal, las cuales tienen como función el pesaje y la impresión de sobres así como la distribución de estampillas según la tarifa designada al peso del sobre para su envío.

Figura 8. Los ATMs en Grecia (azul), Marruecos (amarillo) y Portugal (rojos).



Fuente: AMIEL Electronics [Online].

El estudio de sistemas de control y de sensores, que permitan determinar y manipular la posición de los objetos, es una gran fuente de información adecuada para el desarrollo de este proyecto. La documentación basada en estos estudios puede brindar datos útiles en cuanto al uso de motores paso-paso, dispositivos mecánicos y sensores ópticos dentro de sistemas que permitan posicionar un elemento en un lugar determinado, y de igual forma presente información acerca de la presencia de ciertos elementos, la cual pueda ser usada como señales de entrada hacia los actuadores.

El uso de un microcontrolador como parte principal de la máquina, gracias a sus características, permite simplificar el sistema, ya que este da la posibilidad de realizar un control cíclico y continuo a la mayoría de las variables empleadas generando así una respuesta inmediata a los cambios presentados en sus entradas. Los avances tecnológicos presentados últimamente en sistemas de control permiten también tener un referente de lo que se espera como resultado final del proyecto en cuanto a la manera en que se aborda cada una de las necesidades planteadas por el usuario. En la figura 8, se puede observar algunos ATM's existentes en el mundo.

2. DESARROLLO DEL TRABAJO

En esta parte se procede a describir las diferentes etapas de diseño, prueba y construcción que conllevan a la consecución del objetivo general y los específicos que inicialmente se plantea para que la máquina posea las funciones requeridas y de esta manera tenga un desempeño en el contexto para el que es concebida desde sus inicios.

2.1 DESCRIPCION GENERAL DE LA MAQUINA

Inicialmente en la etapa de diseño se tiene en cuenta las principales funciones de la Máquina Dispensadora de Estampillas con el fin de que preste un servicio coherente con los requerimientos acordados con el personal de la Universidad de Nariño encargado de manejar la distribución de la estampilla.

En primer lugar se plantea la necesidad de diseñar y construir una máquina que sea capaz de dispensar una estampilla autoadhesiva como la que actualmente se vende en el aeropuerto o que de igual forma posea la opción de entregar al usuario un sello seco en el tiquete aéreo de manera que se agilice el proceso de cobro y evite congestiones en las filas de abordaje dentro del aeropuerto.

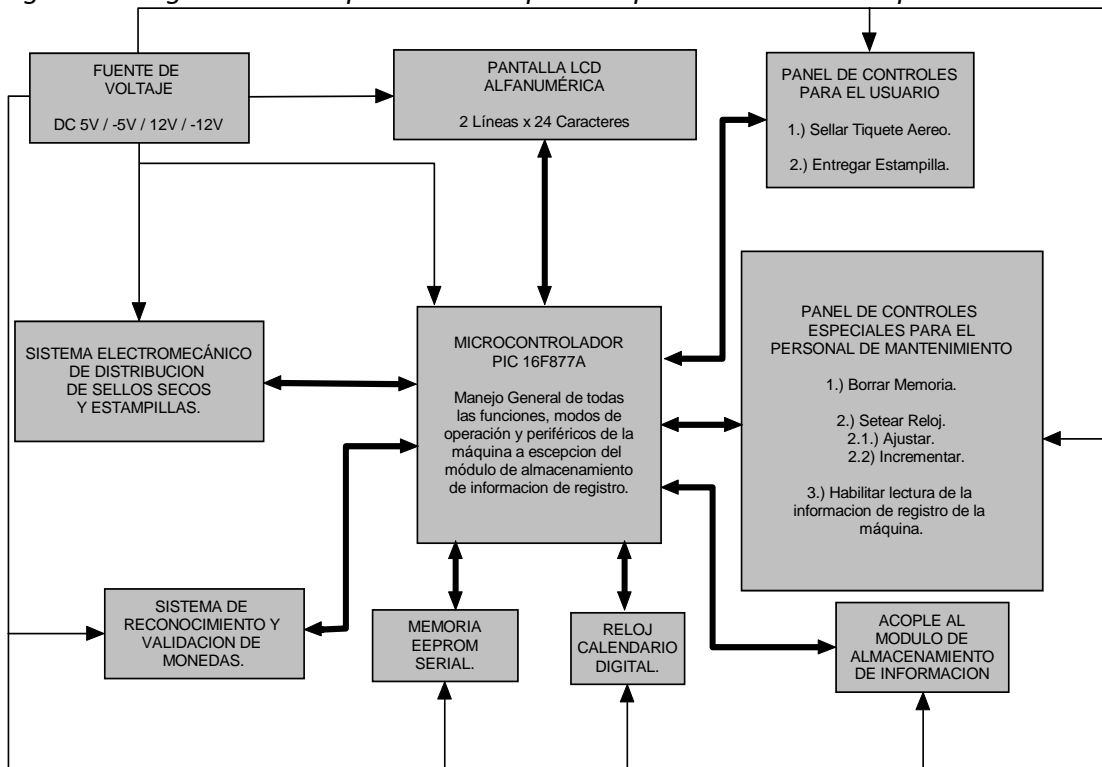
Realizando un análisis mas detallado de la situación, se opta por implementar en la Máquina Dispensadora de Estampillas dos modos de distribución a saber: uno mediante el cual el usuario escoge ubicar su tiquete aéreo en una ranura existente en la máquina produciéndose automáticamente un proceso al interior de ella que permite colocar un sello en alto relieve que certifique de inmediato que el poseedor de dicho tiquete aéreo ya ha realizado el pago del impuesto de estampilla y así evite congestiones en el proceso ya que todo esto debe llevarse a cabo dentro del menor tiempo posible. Por otro lado, la máquina también posee una segunda opción que permite que el usuario en determinado momento pueda retirar de la máquina una estampilla autoadhesiva para que la pueda usar en el momento o transportarla hacia otros lugares lejanos donde se necesita.

La máquina también posee en su interior un sistema que permite almacenar los datos de registro de su funcionamiento y otro externo a ella que admite transportarlos con facilidad y seguridad hacia las oficinas del personal encargado sin sufrir modificación alguna. La existencia de este módulo de almacenamiento de información de registro evita la necesidad de tener que llevar una PC-Portátil hasta el sitio donde se encuentra instalada la máquina, que para este caso particular es el aeropuerto, y descargar o leer

Los datos de relevancia del funcionamiento de la máquina o por el contrario mover la máquina desde su ubicación hasta las oficinas del personal.

El diseño está pensado para que la Máquina Dispensadora de Estampillas “Pro-Desarrollo Udenar” se encuentre en funcionamiento permanente durante largos periodos de tiempo y sin presentar fallos que impidan su funcionamiento normal.

Figura 9. Diagrama de bloques de la máquina dispensadora de estampillas.



En el diagrama de bloques de la figura 9, se encuentra detallado de una manera clara como esta constituida en su totalidad la Máquina Dispensadora de Estampillas en cada una de sus partes principales. En dicho diagrama, y de aquí en adelante, las flechas mas delgadas indican cableado de voltaje para la alimentación de cualquiera de los módulos y las flechas de líneas mas gruesas indican cableado para el intercambio de datos.

Se puede observar que la máquina construida posee varios módulos o bloques importantes y esenciales para su funcionamiento, uno de ellos es la Fuente de Voltaje que entrega la alimentación de corriente y tensión que necesitan todos y cada uno de los bloques funcionales dentro del diagrama por lo tanto se encuentra comunicada, es decir, entrega voltaje a todos ellos.

Esta fuente es regulada y entrega niveles de tensión exactos además de que puede funcionar sin presentar fallos durante largos periodos de tiempo siempre y cuando su conexión a la red eléctrica sea correcta, de tal manera que no represente riesgos para los circuitos que dependen de ella.

El bloque principal que tiene la máquina, que se puede observar en el diagrama de bloques de la figura 9, es el que contiene al Microcontrolador PIC 16F877A-I/P encargado de controlar la totalidad de los sistemas que posee la máquina, por lo cual su modo de funcionamiento se detalla más adelante en este mismo documento. Su trabajo principal es el de procesar todas las señales del circuito de tal modo que la máquina sea capaz de lograr una interacción con el usuario para que finalmente éste obtenga un sello seco o una estampilla autoadhesiva después de realizar el ingreso de una moneda de \$200 (doscientos pesos Colombianos) al sistema.

Adicionalmente a ello también se encarga de controlar las funciones que solamente pueden ser accedidas por el personal de mantenimiento de la máquina que son Borrar la memoria, o sea poner a cero todas las posiciones que contienen los datos de interés del registro de funcionamiento de la máquina, también la función de Ajustar el Reloj/Calendario que contiene la fecha del sistema lo cual es útil para el registro de actividad de la máquina y habilita la lectura de toda la información de registro de la máquina por parte del módulo de almacenamiento.

Una parte muy importante dentro de la máquina e indispensable para su funcionamiento es el Módulo o Sistema de Reconocimiento y Validación de Monedas, el cual se encarga de establecer si las monedas que el usuario ingresa en la máquina son falsas o no, y de este modo realizar el cobro de las estampillas o sellos de la manera mas eficiente y confiable del caso.

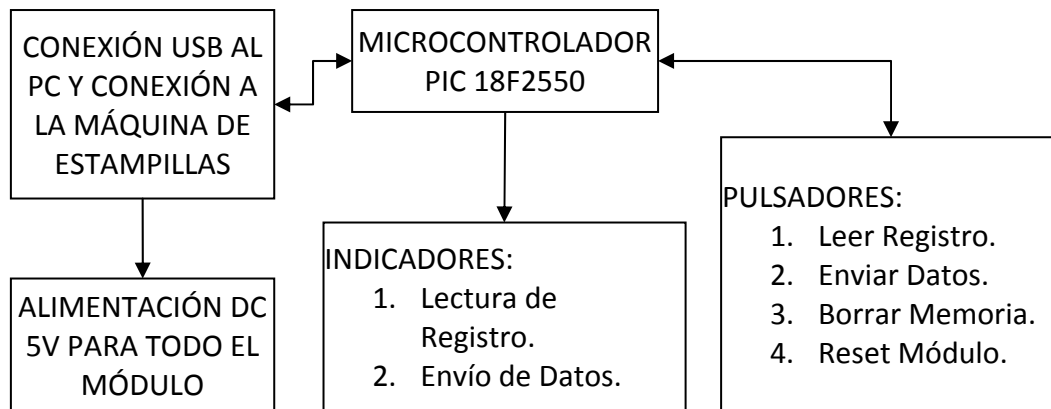
Con el fin de tener la interacción más clara posible con el usuario, la máquina posee una pantalla LCD que permite visualizar mensajes que describen de manera simple y clara lo que esta sucediendo con la máquina en determinado instante y así el usuario esté informado de la actividad de la máquina en todo momento y de este modo pueda tomar una acción respecto a lo que sucede o instrucciones que se deba enviar a la máquina.

2.2 DESCRIPCION GENERAL DEL MODULO DE ALMACENAMIENTO DE INFORMACION DE REGISTRO.

Otra de las partes importantes dentro del funcionamiento de la máquina es el Módulo de Almacenamiento de Información de registro mostrado en la figura 10, el cual es el encargado de leer los datos de importancia de la máquina y guardarlos para poderlos transferir posteriormente a una PC. Este módulo tiene la capacidad de tomar su voltaje de

funcionamiento de la máquina de estampillas cuando se encuentra leyendo la información de registro almacenada en ella y también toma la alimentación del puerto USB del computador donde se encuentre conectada para transmitir la información.

Figura 10. Diagrama de bloques del módulo de almacenamiento de información.



2.3 FUENTE DE VOLTAJE D.C.

Para el desarrollo del proyecto es necesario el uso de una fuente de voltaje que proporcione los niveles de corriente y tensión necesarios para el funcionamiento correcto de todos y cada uno de los elementos que necesitan alimentación D.C. Para escoger el tipo de fuente que utiliza la máquina se tiene cuenta que esta debe ser regulada y poseer un aislamiento adecuado para que proporcione el voltaje los circuitos de la máquina de manera segura y sin sufrir caídas de tensión. La fuente escogida es tipo ATX, cuyas características se encuentran detalladas en la tabla 2.

Tabla 2. Características eléctricas de la fuente ATX.

Numero de Pin	Voltaje (V)
1,2 Naranja	+3.3
3,5,7 Negro	Tierra
4,6 Rojo	+5
8 Blanco	Power Good
9 Violeta	5Vsb
10 Amarillo	+12

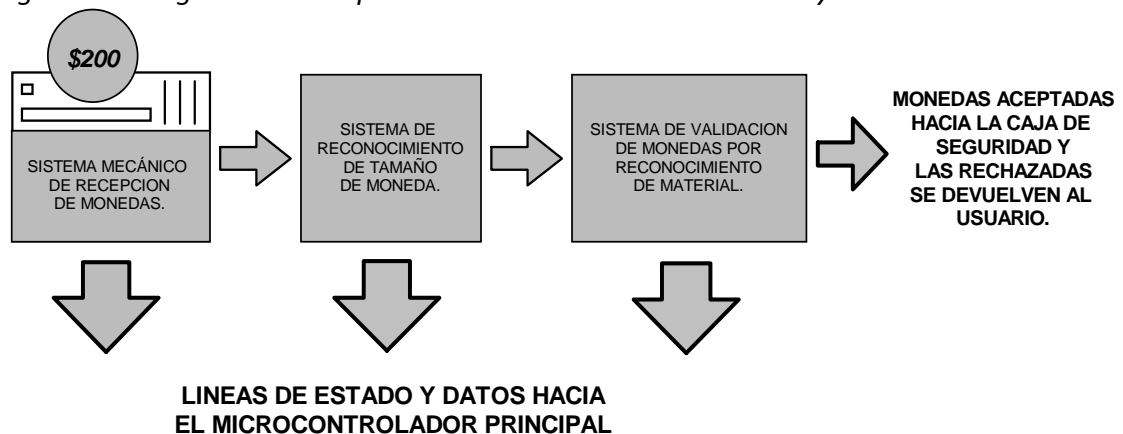
Fuente: Placa de características del Equipo.

2.4 HARDWARE DE LA MAQUINA DISPENSADORA DE ESTAMPILLAS

2.4.1 Implementación del sistema de reconocimiento y validación de monedas. Uno de los módulos que se encuentra descrito en la figura 8 y que es de gran relevancia para que la máquina cumpla con la función para la cual es diseñada es el sistema de reconocimiento y validación de monedas. Dicho sistema es el encargado recibir la moneda que ha sido ingresada por el usuario y someterla a dos pruebas básicas para conocer con exactitud la denominación y legitimidad de ella, que para este caso es la moneda de \$200 (doscientos pesos). Para examinar la moneda, el sistema de reconocimiento y validación utiliza dos métodos bien diferenciados a saber:

1. *Sensores de Tamaño.*
2. *Sensor de Material.*

Figura 11. Diagrama de bloques del sistema de reconocimiento y validación de monedas.



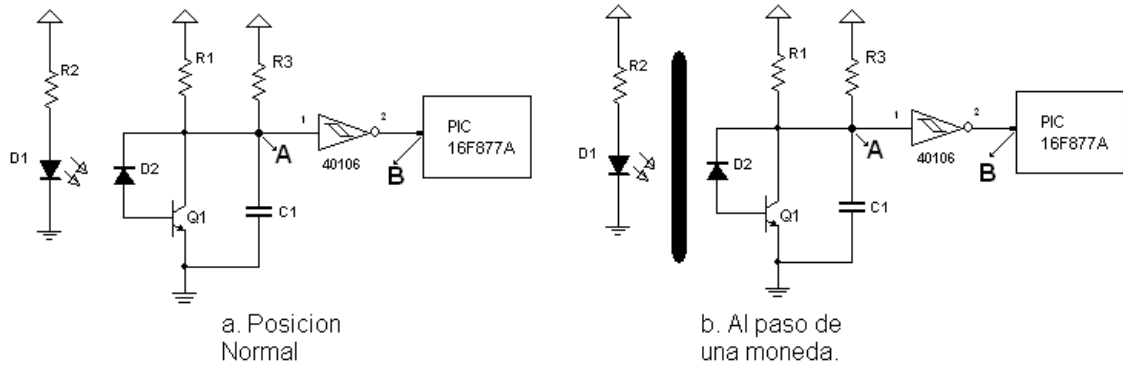
Estos dos sistemas ayudan a realizar una discriminación muy exacta de la moneda de \$200 (doscientos pesos) ya que son capaces de reconocer dos parámetros característicos de esta moneda como son su tamaño y material del que está compuesta.

Esto se puede observar en el diagrama de bloques de la figura 11 en el que se describe de manera general como está compuesto el sistema de reconocimiento y validación de monedas en forma general.

2.4.1.1 Sistema mecánico de recepción de monedas. Cuando el usuario ingresa una moneda de cualquier denominación o tamaño, el sistema mecánico posee un sensor de tipo óptico el cual presenta una señal que cambia de estado de 0V ("0" lógico) a 5V ("1" lógico) cuando la moneda pasa a través de él, después de un corto instante el estado de

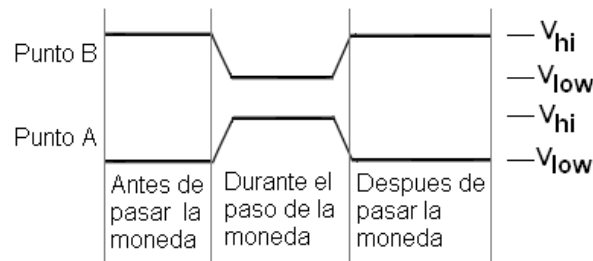
éste vuelve a ser de 0V. El diagrama del circuito de este sensor y su acondicionamiento de señal para determinar su estado se observa en la figura 12.

Figura 12. Esquema del sensor óptico del sistema mecánico de recepción de monedas y su acople de señal al Microcontrolador.



En la figura 12a, se observa que el sistema de detección de una moneda que ingresa a la máquina se encuentra compuesto por un diodo LED emisor infrarrojo (D1) polarizado mediante una resistencia de protección (R2) que se encuentra emitiendo continuamente. Esta radiación hace que el fotodiodo (D2) permita que circule una pequeña corriente en la base del transistor (Q1) y hace que este pase de su estado de saturación a corte y se genere un voltaje bajo en el punto “A”, el cual es sometido a un circuito de acondicionamiento de señal compuesto por un condensador (C1) que quita ruido a dicha señal y ajusta el voltaje para que sea adecuado en nivel, posteriormente se usa un negador tipo Schmitt que proporciona una salida de voltaje estable y cambios rápidos de la señal que proviene del sensor. El comportamiento de la señal que se entrega al Microcontrolador se observa en la figura 12, donde se muestra los niveles de voltaje en los momentos previo, durante y después del paso de una moneda a través de la ranura del sensor que separa al diodo LED infrarrojo D1 del fotodiodo D2.

Figura 13. Señal del Sensor óptico presente en el sistema mecánico de recepción de monedas.



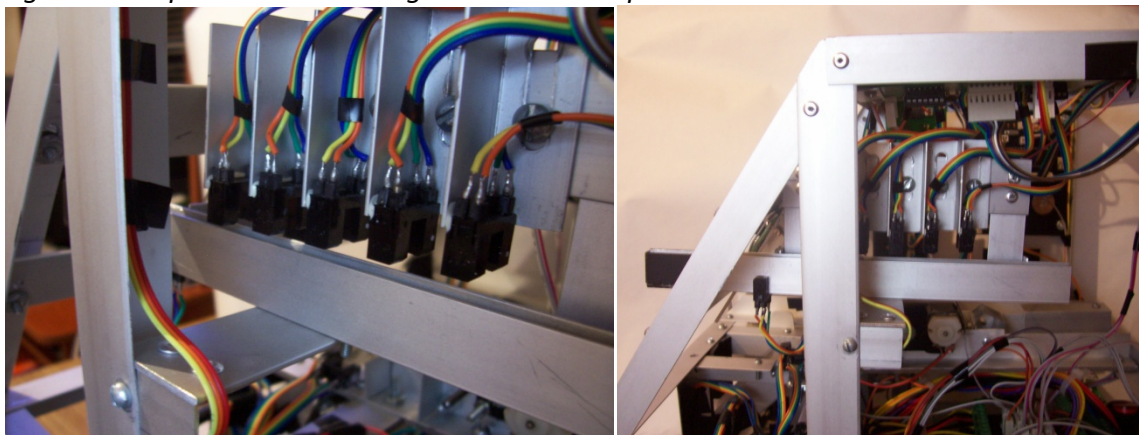
Como se observa en la figura 12b, al momento que pasa la moneda, la señal presente en el punto "A" (Figura 12a), pasa de un estado bajo (V_{low}) a uno alto (V_{hi}) presentándose un cambio inverso en el punto "B" (Figura 12b), es decir, después del negador Schmitt, donde la señal se encuentra continuamente en un nivel alto (V_{hi}) y al paso de la moneda por la ranura del sensor, se presenta un nivel bajo de la señal (V_{low}) durante un tiempo corto equivalente al tiempo que demora en cruzar por completo el diámetro de la moneda a través de la ranura, hecho que es detectado por el Microcontrolador el cual procesa esta señal de manera que sabe que ha ingresado una moneda al sistema y da paso a la detección de los sensores de tamaño. Después del paso de la moneda el sistema vuelve a su estado de reposo.

2.4.2. Sistema de reconocimiento y validación de monedas basado en los sensores de tamaño. Como primera medida es de utilidad el recordar que una correcta validación depende directamente de la facilidad con la cual se pueda obtener un valor determinante del parámetro seleccionado.

La variación del diámetro entre las monedas de diferentes denominaciones puede catalogarse como una de las propiedades más notorias y de fácil caracterización dentro de los sistemas de validación. Por esta razón, se determina el tamaño como una de las variables dentro del diseño del sistema.

Con base al sensor óptico descrito en la figura 12, se realiza un arreglo de 5 sensores infrarrojos tipo barrera Emisor-Receptor, de los cuales cada uno se ubicó a la altura del diámetro de cada una de las monedas circulantes en el país.

Figura 14. Aspecto real del arreglo de sensores ópticos.



En la figura 14, se observa el esquema gráfico de uno de los sensores tipo barrera en el cual se determina como se canaliza la luz infrarroja a través de unas pequeñas ranuras existentes en el encapsulado en forma de herradura del sensor ubicadas frente al emisor y receptor, así mismo se ve que tanto el emisor como el receptor deben ubicarse frente a frente y a la misma altura.

El sensor tipo barrera Emisor-Receptor, como su nombre así lo dice, crea un haz de luz infrarrojo canalizado directamente hacia su receptor, cuando se introduce una moneda en el sistema, el límite superior de esta interrumpe el haz de luz de los sensores de monedas más pequeñas junto con aquel que se encuentra ubicado a la altura exacta de su tamaño, el microcontrolador capta estas señales y mediante su programación es capaz de determinar el valor de la moneda que paso dependiendo de los haces de luz interrumpidos a su paso.

Los sensores infrarrojos tienen una enorme ventaja sobre los sensores de contacto o finales de carrera en este tipo de aplicaciones porque físicamente no hay un contacto que permita desgaste por fricción y por ende genere fallas en la calibración del sensor.

2.4.3. Sistema electromecánico de distribución de estampillas y sellos secos. Esta etapa o sistema se considera como la última interacción de la máquina con el usuario. Dependiendo de las elecciones hechas anteriormente por este, se determina si a la salida final se obtiene una estampilla adhesiva o la aplicación de un sello en relieve sobre el tiquete aéreo.

El sistema en si se divide en dos módulos:

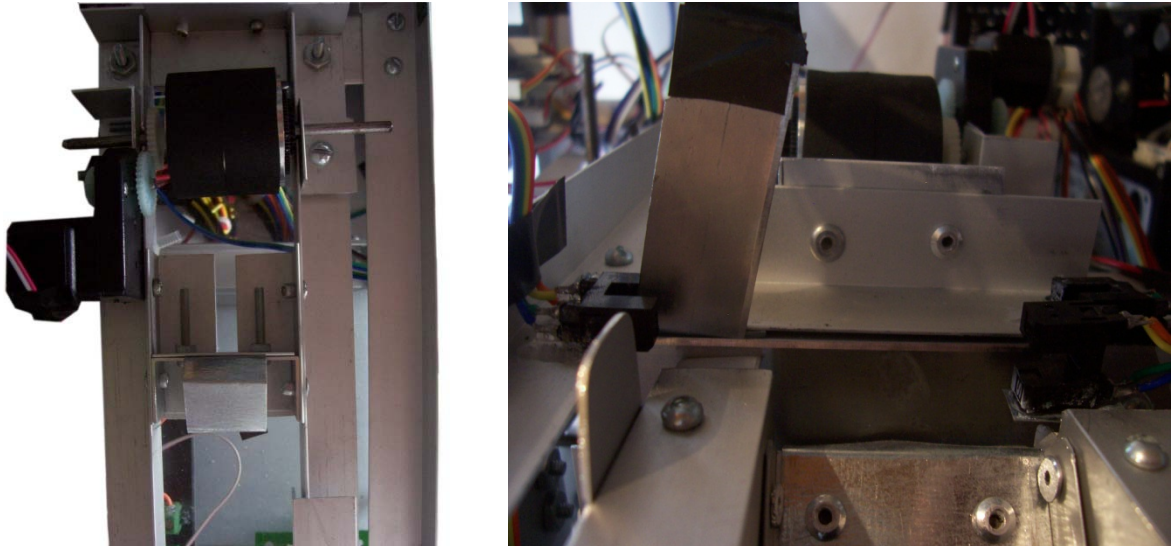
1. *Módulo de distribución de estampillas.*
2. *Módulo de sellado en relieve o sello seco.*

Cada módulo posee su respectiva tarjeta de control, la cual se encarga de adecuar las señales obtenidas para ser transmitidas al microcontrolador y viceversa. Dentro de estas tarjetas se encuentra también una etapa de protección que permite mantener las señales estables evitando sobrecargas que puedan quemar el microcontrolador o presentar señales de ruido sobre los sensores y actuadores.

2.4.4 Módulo de distribución de estampillas. Como su nombre lo indica es el encargado de entregar una estampilla al usuario al final del proceso.

En la figura 15, se encuentra detallado el aspecto real del módulo de distribución de estampillas.

Figura 15. Aspecto real del módulo de distribución de estampillas.



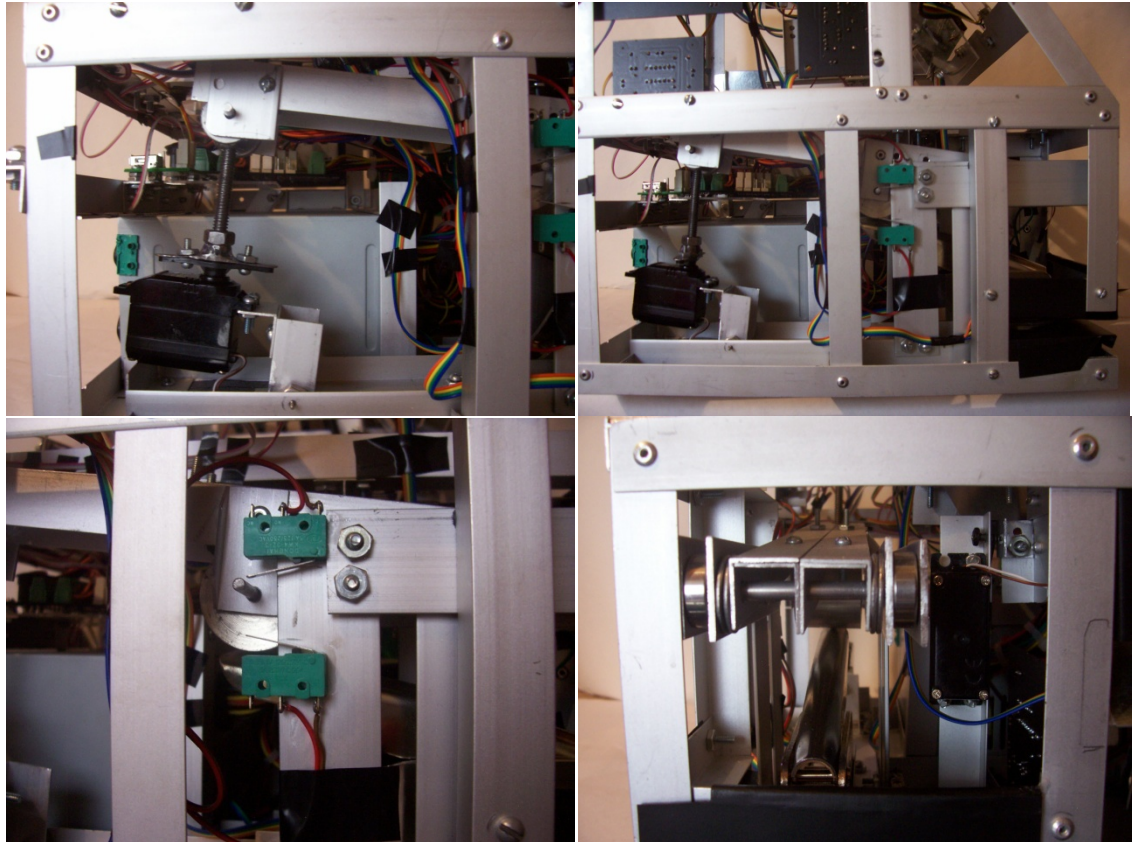
Este módulo, se compone principalmente por dos motores DC y su respectivo circuito de control conformado por un driver L298 y transistores de potencia en corte-saturación, configuración usada para mantener el voltaje en las líneas de entrada estables y funcionales. Este módulo, además, posee un grupo de sensores ópticos tipo barrera que ayudan a determinar la posición del papel y de las cuchillas de corte.

Al recibir el pulso de activación, por medio del control central, el motor de arrastre del papel comienza a halar el papel, arrastrándolo hacia las cuchillas de corte, una vez el primer sensor óptico detecta una marca en la tira de estampillas, el control central procede a desactivar el pulso de movimiento del motor de arrastre e instantáneamente activa el motor de corte que controla las cuchillas.

Luego de activarse el motor de la cuchilla y realizar el corte, esta interfiere el haz de luz de un sensor infrarrojo, por medio de un arreglo de lógica digital, este sensor detiene la cuchilla e invierte el giro del motor para que la cuchilla se mueva ahora en dirección opuesta con el objetivo de devolverla a su punto de origen. Al alcanzar este punto un segundo sensor óptico informa el regreso de la cuchilla a su condición inicial, en este instante la estampilla cae en un pequeño espacio del cual puede ser retirada fácilmente por el usuario.

2.4.5. Módulo de sellado en relieve o sello seco. Como se observa en la figura 16, este pequeño sistema se encuentra constituido principalmente por un servomotor modificado para un funcionamiento como motor de corriente continua.

Figura 16. Aspecto real módulo de sellado en relieve.



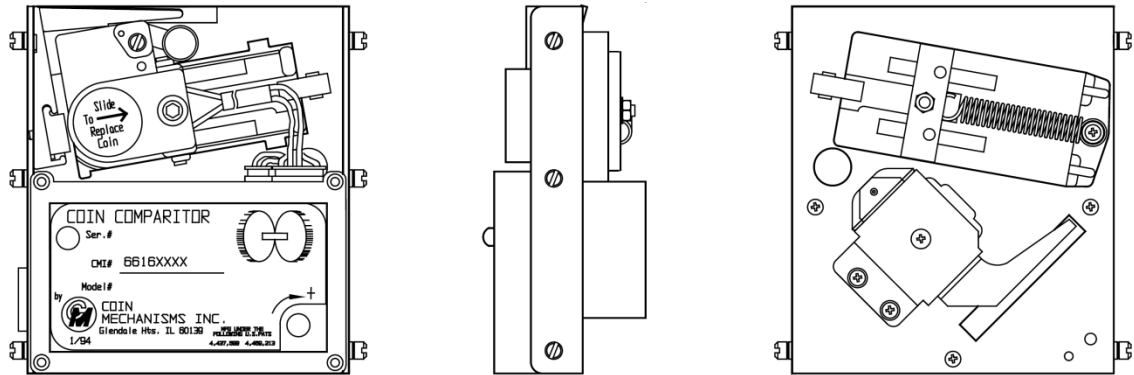
Aprovechando la fuerza de torque del motor, se le adhiere a su eje de rotación un tornillo sin fin, el cual genera un desplazamiento vertical generando una palanca sobre una pinza metálica que lleva en sus extremos el relieve del sello que será impreso en el tiquete aéreo. Esta configuración determinada en un arreglo de doble palanca, incrementa la fuerza del motor y la presión ejercida por la pinza sobre el papel.

Al recibir la señal de activación el motor genera la rotación del tornillo permitiendo el desplazamiento de uno de los extremos del brazo hacia abajo. Al llegar este a un tope limitado por un final de carrera genera una presión sobre la pinza metálica, esta presión se transmite a los extremos de la pinza, que se unen aprisionando el papel, imprimiendo de esta manera el relieve del sello. Así mismo, el final de carrera ubicado en la parte inferior del módulo, activado por la palanca, emite una señal que al ser interpretada por el control principal invierte el giro del motor desplazando la placa hacia arriba, este brazo de aluminio deja de hacer presión sobre la pinza liberando así el tiquete.

En la parte superior del módulo y al finalizar su recorrido, la palanca activa de igual forma un final de carrera cuyo pulso es interpretado como el final del proceso, esto permite al usuario retirar su tiquete aéreo ya sellado.

2.4.6. Sistema de reconocimiento y validación de monedas basado en la detección del material. Desde la etapa inicial del proceso de diseño y construcción de la máquina dispensadora de estampillas, se establece como prioridad la selección del tipo de sistema que se encargue del reconocimiento de las monedas. Dicho sistema debe ser confiable, estable, de fácil calibración, con gran posibilidad de conexión a un microcontrolador, tener un tiempo de acción reducido y poseer la facilidad de adaptarse al tipo de material del que esta constituida la moneda para la que se plantea funcione la máquina.

Figura 17. Coin Comparitor CC-16.d.100.



Fuente: CC-16.100 Dimensions [Online]. ©Coin Mechanisms Inc. Disponible en Internet: <<http://www.coinmech.com>>.

El sistema que se escoge para encargarse de la detección de monedas, es el “COIN COMPARITOR” fabricado por COIN MECHANISMS INC. Este dispositivo se halla actualmente en máquinas tragamonedas o de “casino” y es muy difundido en este medio ya que es bastante confiable y funciona durante prolongados periodos de tiempo sin presentar grandes fallas. Debido a su gran desempeño y especialización en la detección de monedas este sistema es de difícil consecución en el mercado debido a que el fabricante no vende el dispositivo de manera individual sino en cantidades promedio de cien unidades en adelante, además de poseer un elevado precio. Por lo tanto, el sistema es adquirido por intermedio del personal de mantenimiento de las máquinas tragamonedas de un casino local.

El modelo que actualmente se encuentra funcionando en la máquina es el COIN COMPARITOR modelo CC-16.D.100 que se observa en el esquema de la figura 17. Este modelo al igual que todos los de su tipo, se denomina “comparador” ya que utiliza una moneda residente de un valor determinado, que en este caso es la moneda de \$200 pesos (Colombianos) para que sea comparada con las monedas que entran a la máquina; en caso de que sean iguales el sistema acepta la moneda que ha ingresado. En caso contrario la rechaza. Para realizar dicho proceso utiliza un sistema en base a inductores especiales que se basan en el principio del LVDT (Transformador Diferencial de Variación Lineal).

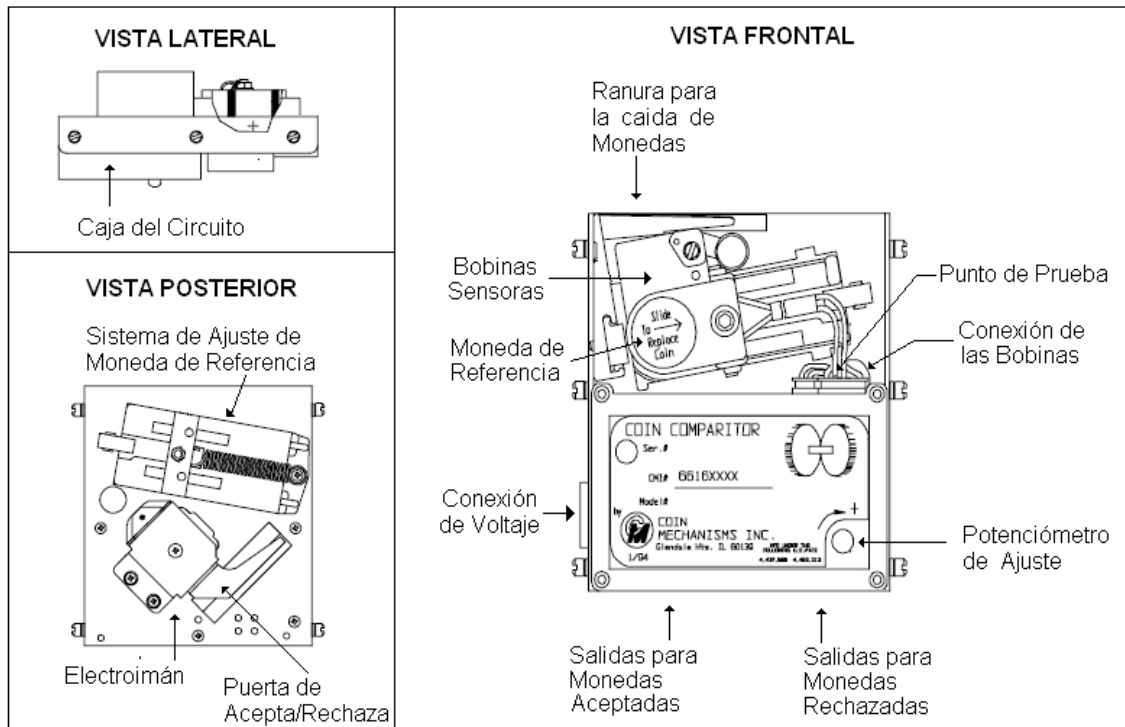
Es de gran importancia para el proyecto saber como funciona este sistema comparador de moneda ya que no se cuenta con ningún tipo de dato técnico del mismo que conduzca a comprender en detalle el manejo de este sistema. Por lo cual es estrictamente necesario llevar a cabo ciertas mediciones en laboratorio que conducen a comprender su funcionamiento. Todo esto con la finalidad de obtener algún tipo de señales que puedan ser interpretadas por el microcontrolador y de esta manera controlar el funcionamiento completo de la máquina; este estudio se detalla a continuación.

2.4.7. Partes del comparador de monedas. El dispositivo comparador de moneda posee varias partes que son esenciales para su funcionamiento. Las partes importantes se observan en la figura 18 y se detallan a continuación:

2.4.8. Bobinas sensoras. Es el sistema encargado de realizar la comparación de la moneda residente y la moneda que ingresa al sistema. Se basa en el principio del Transformador Diferencial de Voltaje Lineal. El aspecto real de este sensor se ve en detalle en la figura 19.

Como se observa en el diagrama de disposición de las bobinas del sistema sensor de moneda (figura 20), la bobina primaria es la que se encuentra comprendida entre los puntos R y A, recibe el voltaje de excitación y genera un flujo que atraviesa la bobina secundaria induciendo una señal de voltaje en los terminales denominados por los puntos B y N. La moneda residente y la moneda que ingresa actúan como núcleos de este sistema y en dependencia de su material cambian la inductancia mutua entre las bobinas, de tal manera que este fenómeno puede ser detectado por el circuito de amplificación y acondicionamiento de señal que se explica más adelante.

Figura 18. Identificación de partes del COIN COMPARITOR CC-16.D.100.



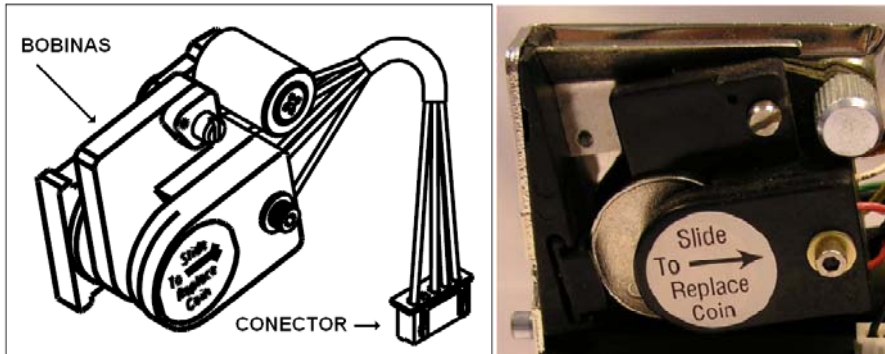
Fuente: CC-16.100 Parts Identification [Online]. ©Coin Mechanisms Inc. Disponible en Internet: <<http://www.coinmech.com>>.

Realizando mediciones en el laboratorio, se encuentra los siguientes datos referentes al sensor:

- Bobina 1: Resistencia eléctrica medida entre R y A = 73.57Ω. Inductancia medida entre R y A = 4.829mH.
- Bobina 2: Resistencia eléctrica medida entre B y N = 36.41Ω. Inductancia medida entre B y N = 2.546mH.

Debido a la constitución y construcción del sistema sensor, es imposible determinar exactamente otros parámetros de importancia tales como número de vueltas en cada bobina, inductancia mutua, calibre del alambre, etc., que pueden ayudar a detallar a fondo el funcionamiento de este sensor, ya que para saberlos implicaría la destrucción del sensor.

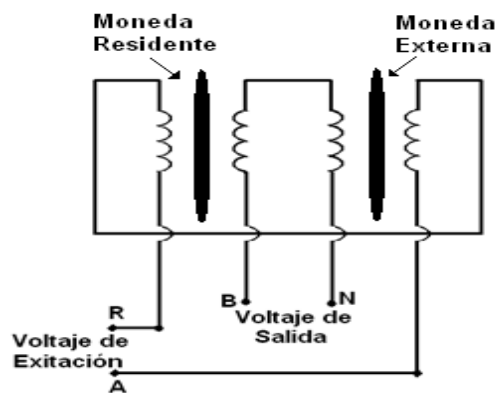
Figura 19. Aspecto real y esquema de las bobinas sensoras del sistema comparador de moneda.



Fuente: CC-16.100 Parts Identification [Online]. ©Coin Mechanisms Inc. Disponible en Internet: <<http://www.coinmech.com>>.

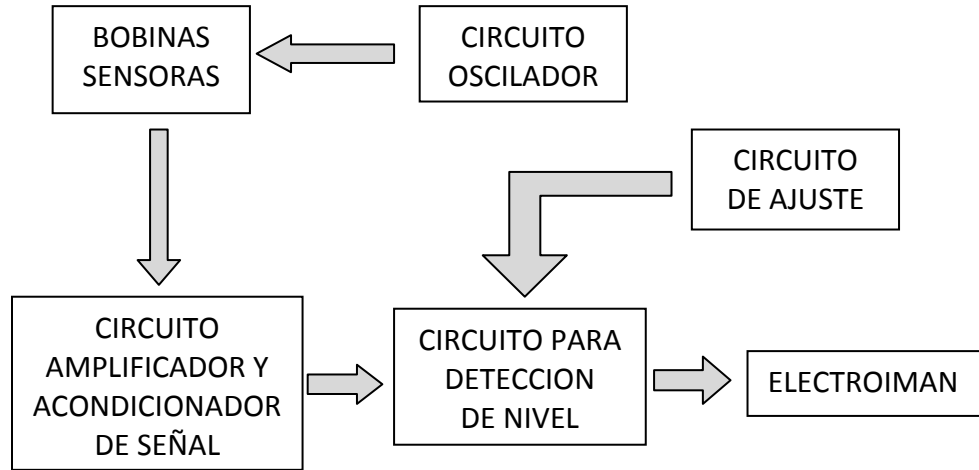
En la figura 19 y 20, se observa que el sistema de bobinas sensoras tiene un espacio destinado a una moneda residente la cual debe ser exactamente igual al tipo de monedas que se desea el sistema acepte, es decir, en este caso la moneda residente debe ser de \$200 (doscientos pesos Colombianos) de tal manera que el sistema acepta todas las monedas que sean del mismo material y rechaza todas las demás. De este modo la moneda residente queda situada paralela a la ranura por donde entran las monedas y entre la bobina sensora central y la bobina de excitación lateral frontal.

Figura 20. Disposición de las bobinas del sistema sensor de moneda.



2.4.9. Circuitos internos del comparador de monedas. Cabe destacar que algunos datos son encontrados por medio de la página de internet del fabricante, a partir de los cuales obtiene escasa información que sirve de guía en el proceso de calibración y funcionamiento de este sistema.

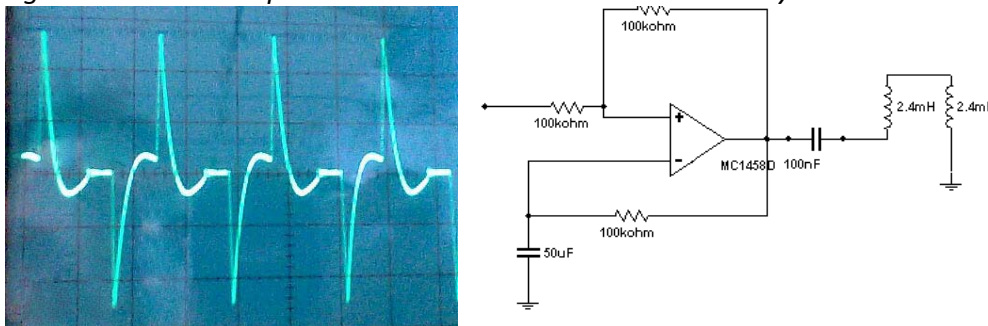
Figura 21. Diagrama de bloques del circuito interno del sistema sensor de moneda.



El plano eléctrico del sistema sensor de monedas no es proporcionado por el fabricante ni tampoco por el vendedor por lo cual es leído de la placa impresa que posee el sistema. Del esquema general del circuito se resume mediante un diagrama de bloques como el de la figura 21 que describe las funciones de cada etapa.

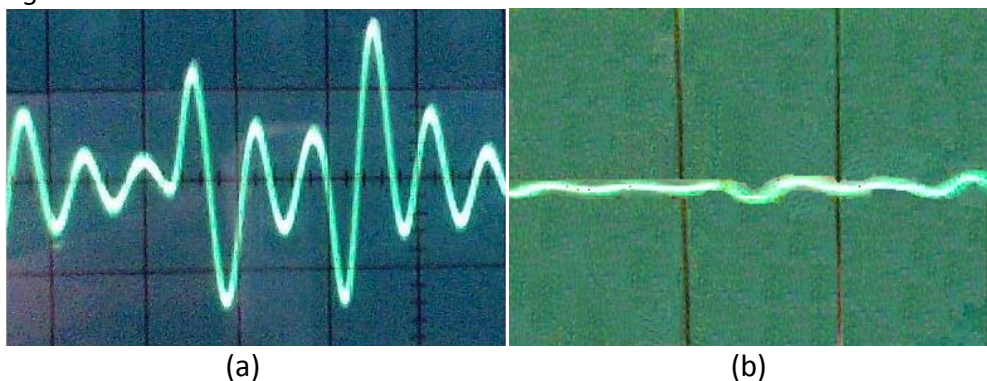
Según el diagrama de bloques se destaca que el circuito posee varias funciones, en primer lugar la bobina principal del sensor de monedas es alimentada por un circuito oscilador en base a un Amplificador Operacional MC1458 fabricado por Philips y varias resistencias, la señal oscilante se mide utilizando un osciloscopio y se determina que era de frecuencia $f = 4545.45 \text{ Hz}$ y voltaje $V = 10 \text{ V}_{PP} = 3.54 \text{ V}_{RMS}$. El circuito encargado de generar esta onda se encuentra en la figura 22, donde también se observa la forma de la señal.

Figura 22. Oscilador para las bobinas del sensor de moneda y su señal de salida.



La señal del oscilador se aplica a la bobina principal y luego es sensado el material de la moneda mediante la bobina central o secundaria, quien entrega este voltaje un circuito amplificador y acondicionador que se encarga de llevar la señal del sensor a niveles aceptables y que son relacionados con el tipo de moneda que entra. Cabe destacar que el nivel de voltaje de la señal que se recibe mediante la bobina sensora central es de un nivel muy bajo y también se usa un osciloscopio para determinar que su voltaje $V = 106\text{mV}$ y su frecuencia es $f = 4545.45\text{Hz}$, que es la misma que tiene el oscilador que alimenta la bobina primaria. La señal de la bobina secundaria, que se puede observar en la figura 23a, es la que entra al circuito bloque amplificación y acondicionamiento de señal. Cuando por las bobinas pasa una moneda de diferente denominación que la residente dicha señal sufre un pequeño cambio en su amplitud sin que se modifiquen sus otras características, pero cuando por las bobinas sensoras pasa una moneda de igual denominación, es decir, idéntica en tamaño, forma y material del que esta construida, la señal de la figura 23a, se convierte en la señal de la figura 23b, donde se ve que disminuye su amplitud y cambia su forma casi hasta el nivel de cero.

Figura 23. Señal de la bobina de detección del sensor de moneda.



El bloque descrito como circuito para detección de nivel se encarga de realizar la activación del electroimán que permite correr la puerta de aceptación o rechazo de monedas. Esto lo hace en dependencia del nivel de la señal que entrega el bloque de amplificación y acondicionamiento de señal, cuando la señal se vuelve casi a su nivel de cero este circuito determina que la moneda ingresada es exactamente como la residente por lo tanto hace que la puerta permita el paso de la moneda hacia otra etapa del sistema, en caso contrario la puerta no se mueve y hace que la moneda ingresada sea retorne hacia el usuario.

2.5 EL MICROCONTROLADOR PIC 16F877A.

Para que la máquina dispensadora de estampillas cumpla con sus funciones necesita de un dispositivo central que le otorgue la capacidad de controlar todos los periféricos, recibir y procesar las señales provenientes de sensores y demás elementos de los que consta la máquina. El dispositivo escogido para tal fin es el PIC 16F877A, que es un Microcontrolador del fabricante Microchip a cuya familia se le denomina PIC y que tiene las siguientes características generales:

- Bus de datos de 8 bits.
- Arquitectura Harvard.
- Tecnología RISC.
- Tecnología CMOS.

Estas características se conjugan para lograr un dispositivo altamente eficiente en el uso de la memoria de datos y programa y por lo tanto en la velocidad de ejecución.

Se selecciona el Microcontrolador PIC 16F877A debido a su facilidad de manejo, abundante información, herramientas de diseño y disponibilidad dentro del mercado local. Otros parámetros que destacan a este Microcontrolador son su velocidad, consumo, tamaño y alimentación.

El PIC 16F877A posee una gran variedad de elementos que son muy útiles para que la máquina dispensadora de estampillas controle todos sus periféricos e interactuar con el usuario de la máquina. Para ello posee varias características que hacen a este Microcontrolador un dispositivo muy versátil, eficiente y práctico para ser empleado en esta aplicación.

Algunas de las características más relevantes de este dispositivo se muestran a continuación:

Características del Núcleo del Microcontrolador.

- Set de 35 instrucciones sencillas.
- Todas sus instrucciones se ejecutan en un ciclo de máquina a excepción de las instrucciones de salto o llamado a una subrutina que ocupan dos ciclos.
- Frecuencia de operación máxima: DC – 20 MHz (Entrada de Reloj)
- Ciclo de instrucción de hasta 200 ns. con reloj de 20 MHz
- Memoria FLASH de programa de 8k x 14 palabras.
- Memoria RAM de datos de 368 x 8 bytes.
- Memoria EEPROM de datos de 256 x 8 bytes.

- Pines de salida compatibles con los PIC 16C73B/74B/76/77.
- Capacidad de 14 fuentes de interrupciones.
- Pila con 8 niveles.
- Modos de direccionamiento directo, indirecto y relativo.
- Power - On Reset (POR).
- Power - Up Timer (PWRT) Y Oscillator Start – Up Timer (OST).
- Watchdog – Timer (WDT) con su propio chip para un oscilador RC para una operación más confiable.
- Protección de código programable.
- Modo SLEEP para ahorro de energía.
- Opciones de oscilador seleccionable.
- Tecnología CMOS FLASH/EEPROM de alta velocidad y baja potencia.
- Diseño completamente estático.
- Programación serial, In-Circuit Serial Programming™ (ICSP) vía dos pines.
- Capacidad de In-Circuit Serial Programming™ (ICSP) con solo 5V.
- El procesador puede acceder a escribir y leer la memoria de programa.
- Amplio rango de voltaje de operación: 2.0V a 5.5V.
- In – Circuit Debugging via 2 pines.
- Rangos de temperatura industrial, comercial y extendido.
- Bajo consumo de potencia: < 0.6mA típica a 3V, 4MHz, 20µA típica a 3V, 32 KHz, y < 1 µA corriente típica en standby.

Características de los Periféricos del Microcontrolador.

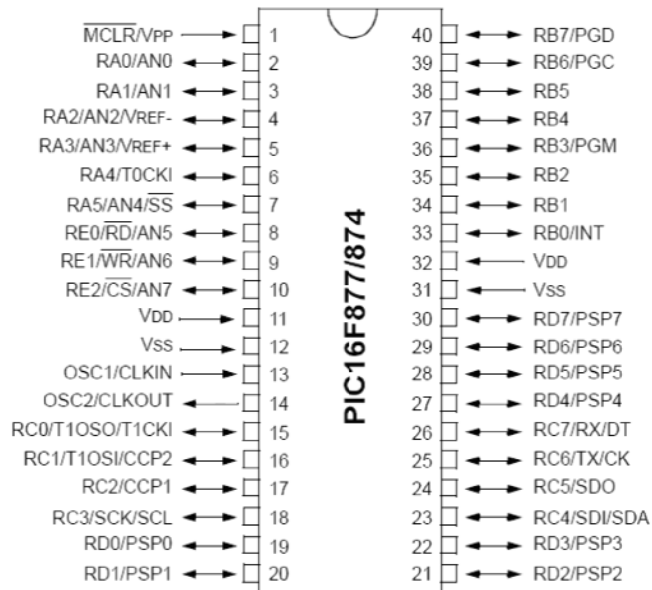
- Timer 0: 8-bit timer/counter con prescaler de 8-bit.
- Timer 1: 16-bit timer/counter con prescaler, puede ser incrementado durante sleep vía reloj/cristal externo.
- Timer 2: 8-bit timer/counter con un registro de periodo de 8 – bit, prescaler y postscaler.
- Dos módulos de Capture, Compare y PWM.
 - Capture es de 16-bit, su máxima resolución es: 12.5 ns.
 - Compare es de 16-bit, su máxima resolución es: 200ns.
 - PWM, su máxima resolución es de: 10-bit.
- Convertidor Analógico a Digital de 10-bit multi-canal.
- Synchronous Serial Port (SSP) con SPI (Master Mode) e I2C (Master/Slave).
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) con detección de dirección de 9-bit.
- Parallel Slave Port (PSP) de 8-bits de ancho, con controles externos de RD, WR y CS (solo para dispositivos de 40/44 pines).
- Brown-out detection circuitry para Brown-out Reset (BOR).

2.5.1 Diagrama y descripción de pines. El PIC 16F877A viene fabricado en varias presentaciones de acuerdo con las necesidades del diseñador, para la máquina se escoge la presentación de 40 pines tipo PDIP (Plastic Dual In Line Package). Su diagrama de pines se puede observar en la figura 23.

En el diagrama de pines de la figura 23 se puede observar que el PIC 16F877A tiene una gran cantidad de funciones las cuales fueron descritas anteriormente en este mismo capítulo y logra diferenciar que la mayoría de los pines tienen o son capaces de cumplir varias funciones dependiendo de cómo estén configurados por el diseñador dentro del programa. Para lograr un entendimiento mayor en este sentido es necesario conocer en detalle las posibles funciones de cada pin y a que módulo pertenece, este hecho se puede observar con detalle en la hoja de datos del elemento.

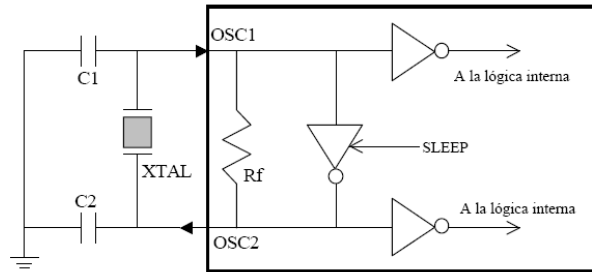
2.5.2 Configuración del oscilador. Para la máquina que se construye se utiliza un oscilador a cristal de 4MHz, ya que para esta aplicación no se requiere una gran velocidad de ejecución. Para escoger los valores correctos de C1 y C2 (Figura 24), el fabricante recomienda seguir los valores recomendados en la hoja de datos del microcontrolador. Estos valores son C1=C2=15pF.

Figura 24. Diagrama de pines del PIC16F877A.



Fuente: Microchip, PIC16F87X Datasheet, 2001 Microchip Technology Inc., DS30292C- página 1.

Figura 25. Cristal Externo como oscilador.



Fuente: Microchip, PIC16F87X Datasheet, 2001 Microchip Technology Inc. Página 121.

En el modo XT un cristal o resonador cerámico se conecta a los pines OSC1/CLKIN y OSC2/CLKOUT para establecer la oscilación (Figura 25). También se puede manejar una fuente de reloj externa que se conecta al pin OSC1/CLKIN.

2.5.3 Resumen del set instrucciones. Porque la máquina dispensadora de estampillas posee una gran variedad de tareas, es de gran importancia conocer el set de instrucciones del PIC 16F877A debido a que el programa principal que posee la máquina controla muchas actividades, módulos y periféricos que posee el microcontrolador con el fin de desempeñar sus funciones de manera rápida y eficiente. Cada instrucción del PIC16F877 es una palabra de 14 bits, dividida en un OPCODE el cual especifica el tipo de instrucción y uno o más operandos los cuales además especifican la operación de la instrucción.

El PIC 16F877A posee un set de 35 instrucciones, todas son ejecutadas en un solo ciclo de instrucción que para este caso es $1\mu\text{S}$ debido a que se utiliza un cristal de 4MHz para generar la señal de reloj del microcontrolador. La tabla 3 muestra el set de instrucciones del PIC16F877A y que son reconocidas por el ensamblador MPASM™ de Microchip MPLAB mediante el cual se crea el código fuente del programa que gobierna todo el sistema.

El PIC16F877A posee una amplia gama de periféricos los cuales le dan un gran potencial de control sobre muchas aplicaciones además de su facilidad de configuración y manejo. Para desarrollar el software de control de la Máquina Dispensadora de Estampillas y entender su funcionamiento es necesario profundizar en el uso de dos módulos que ayudan a simplificar a programa diseñado y hacer un uso mas eficiente de los recursos que proporciona el PIC; estos dos módulos son el denominado "Timer0" y el MSSP I2C, los cuales se detallan a continuación.

Tabla 3. Set de Instrucciones del PIC16F877.

Mnemonic, Operands	Description	Cycles	14-Bit Opcode			Status Affected	Notes		
			MSb	LSb					
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z	2
CLRWF	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff		
NOP	-	No Operation	1	00	0000	0xxx	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1(2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1(2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO,PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETlw	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO,PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

- Note 1:** When an I/O register is modified as a function of itself (e.g., MOVF PORTB, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2:** If this instruction is executed on the TMR0 register (and, where applicable, d – 1), the prescaler will be cleared if assigned to the Timer0 module.
- 3:** If Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is

Fuente: PIC16F87X Datasheet, 2001 Microchip Technology Inc. Página 136.

2.5.4 Módulo timer0. Los PIC16F877 tienen un módulo para el manejo preciso y eficiente de operaciones que involucran tiempo o conteo. Este módulo consta de tres contadores/temporizadores denominados TMR0 (Timer0)/TMR1(Timer1)/TMR2 (Timer2) y dos módulos CCP (Captura, Comparación y PWM) denominados CCP1 y CCP2. Para este caso particular solo interesa el módulo del Timer0, cuyas principales características son:

- El TMR0 es un contador/temporizador de 8 bits.
- Permite acceso de lectura y escritura de sus datos.
- Selección de Reloj Interno o Externo.
- Selección de flanco activo en el reloj externo.
- Pre-escalador de 8 bits programable.
- Solicitud de interrupción programable en el desbordamiento de FFh a 00h.

El registro principal de este módulo es el TMR0 (01h, 101h). Este registro se incrementa continuamente a una frecuencia seleccionable manejada por un pre-escalador y un reloj interno $F_{osc}/4$ (modo temporizador) o bien por un pre-escalador y una señal externa (modo contador). En la hoja de datos se encuentra el diagrama de bloques de este módulo, donde se puede observar los bits que afectan su operación y la manera en que lo hacen.

2.5.5 Módulo I2C del PIC16f877A. El PIC16F877A posee un módulo llamado MSSP (Master Synchronous Serial Port), el cual es una interface serial usado para comunicarse con otros dispositivos o microcontroladores. Estos dispositivos periféricos pueden ser memorias EEPROM seriales, shift registers, drivers para displays, conversores A/D, etc. El módulo MSSP es factible de operar de dos modos:

- Serial Peripheral Interface (SPI).
- Inter-Integrated Circuit (I2C).

2.5.5.1 Modo de operación. El dispositivo maestro genera todos los pulsos de reloj serial y las condiciones de START y STOP. Se completa una transferencia con una condición de STOP o REPEATED START. Si la condición de REPEATED START esta al inicio de la siguiente transferencia serial, el bus I2C no será liberado.

En el modo de Transmisor Maestro, los datos seriales emergen a través de la línea SDA, mientras la señal de reloj serial se transmite por SCL. El primer byte transmitido contiene la dirección de esclavo del dispositivo transmisor (7-bits) y el bit de Read/Write (R/W). En este caso el bit $\overline{R/W}$ será un '0' lógico. Los datos seriales se transmiten 8-bits a la vez. Después de que cada byte se transmite, se recibe un bit de Acknowledge. Las condiciones de START y STOP son enviadas para indicar el inicio y el final de una transferencia serial.

En el modo de Receptor Maestro, el primer byte transmitido contiene la dirección del esclavo o del dispositivo transmisor (7-bits) y el bit $\overline{R/W}$. En este caso el bit $\overline{R/W}$ será un '1' lógico. Así, el primer byte transmitido es la dirección del esclavo de 7-bits seguido por un '1' para indicar recepción de bits. Los datos seriales se reciben vía SDA, mientras SCL es la salida del reloj serial. Los datos seriales se reciben de 8-bits a la vez. Después de que cada byte se recibe, un bit de Acknowledge se envía.

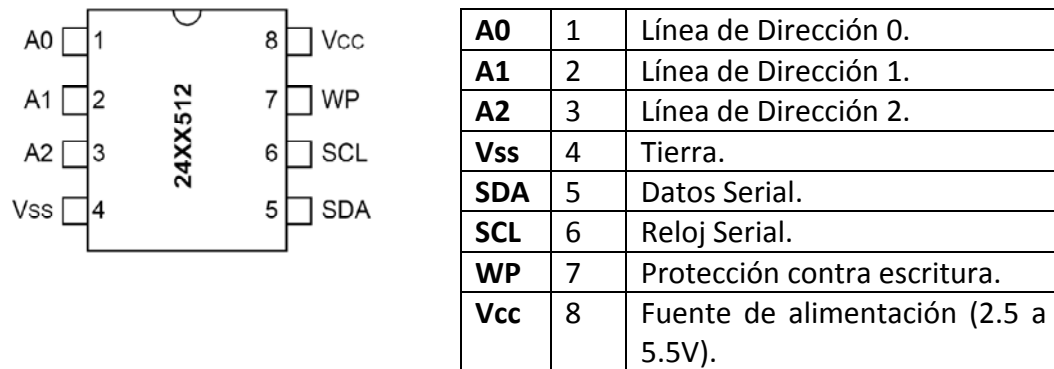
Una secuencia de transmisión típica sería la siguiente:

- El usuario genera una condición de START seteando el bit SEN (START Enable) en el registro SSPCON2.
- Se setea SSPIF. El módulo esperará el tiempo de start requerido antes de cualquier otra operación.
- El usuario carga el SSPBUF con la dirección a transmitir.
- La dirección se desplaza por el pin SDA hasta que todos los 8 bits sean transmitidos.
- El módulo MSSP desplaza el bit ACK desde el dispositivo esclavo y escribe su valor en el registro SSPCON2<6>.
- El módulo MSSP genera una interrupción al final del noveno ciclo de reloj seteando SSPIF.
- El usuario carga el registro SSPBUF con los 8-bits de datos.
- Los datos se desplazan por el pin SDA hasta que todos los 8-bits se transmiten.
- El módulo MSSP desplaza el bit ACK desde el dispositivo esclavo y escribe su valor en el registro SSPCON2<6>.
- El módulo MSSP genera una interrupción al final del noveno pulso de reloj seteando el bit SSPIF.
- El usuario genera una condición de STOP seteando el bit PEN (Stop Enable Bit) en el registro SSPCON2.
- Se genera una interrupción una vez se completa la condición de STOP.

2.6 MEMORIA EXTERNA EEPROM SERIAL 24LC512.

En este proyecto se observa que es necesario el uso de una memoria EEPROM externa a la que posee el microcontrolador de la máquina ya que al hacerse necesario la creación del dispositivo de almacenamiento de información de registro es más fácil y práctico que esta memoria sea externa y pueda ser accesada por dicho módulo en cualquier momento. Además de esto, ya que el microcontrolador maneja varios dispositivos a la vez es posible que se presenten fallas con lo cual se pierde el registro de actividad de la máquina irremediablemente. Al poseer una memoria externa el registro está almacenado de manera más segura y en caso de fallo del microcontrolador principal, su información puede ser recuperada fácilmente por el módulo de Almacenamiento de Información de Registro o en otro caso por cualquier microcontrolador que pueda manejar este tipo de memorias.

Figura 26. Memoria EEPROM Serial 24LC512.



Fuente: 512K I2C™ CMOS Serial EEPROM, Datasheet.

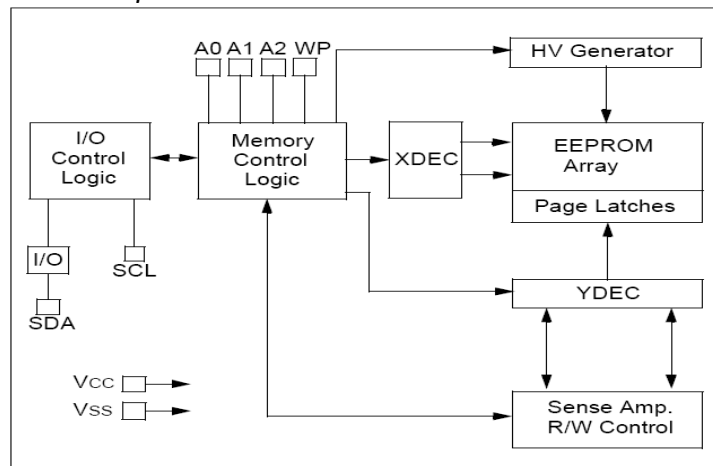
La memoria escogida para lograr los propósitos mencionados anteriormente es la **EEPROM Serial 24LC512**. Este dispositivo fabricado por Microchip posee grandes ventajas que permiten facilidad de uso y manejo tales como:

- Tecnología CMOS de Bajo consumo de potencia.
 - Máxima corriente de escritura 5mA a 5.5V
 - Máxima corriente de lectura 400µA a 5.5V
 - Corriente de Standby (bajo consumo) 100nA, típica a 5.5V
- Bus de Interfaz Serial de 2-cables, compatible con I2C.
- Permite conectar hasta 8 dispositivos en cascada.
- Ciclo auto-temporizado de borrado/escritura.
- Modo de escritura por páginas de 128-bytes.
- Tiempo de ciclo de escritura de 5ms. máximo.
- Protección contra escritura por hardware del arreglo completo (WP). Si WP se conecta a masa el funcionamiento del chip es normal. Si se conecta a Vcc no es posible la escritura de la memoria.
- Entradas tipo Schmitt Trigger para supresión de ruido.
- 1'000.000 de ciclos de borrado/escritura.
- Protección de descarga electrostática > 4000V.
- Retención de datos >200 años.
- Amplios rangos de temperatura de trabajo.

La tecnología 24LC512 es una PROM Borrable Eléctricamente (EEPROM "Electrically Erasable PROM"), de 64K x 8 (512 Kbit) capaz de operar en un rango de voltaje de 1.8V

hasta 5.5V. Es decir, esta estructurada en 64K posiciones de memoria con una longitud de 8 bits cada una. Esta ha sido desarrollada para aplicaciones avanzadas de baja potencia tales como adquisición de datos. Este dispositivo es capaz de leer secuencial y aleatoriamente. En la figura 27 se observa el diagrama de bloques de este dispositivo.

Figura 27. Diagrama de bloques de la EEPROM Serial 24LC512.



Fuente: 512K I2C™ CMOS Serial EEPROM, Datasheet.

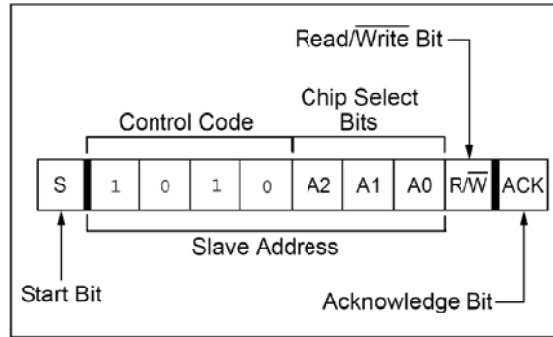
2.6.1. Direccionamiento como esclavo. El direccionamiento de la memoria se rige por el byte de control que envía el microcontrolador maestro, después de la condición de Start, lo cual se encuentra en la figura 28. Esta dirección consta de una parte fija y otra programable. La parte programable se define por la conexión de los pines A0, A1 y A2. El direccionamiento se hace de la siguiente manera:

Después del bit de Start, el byte de control comienza por los bits “1010”.

A continuación viene la parte programable que son los bits A2, A1 y A0 definidos por el conexionado de los pines del mismo nombre.

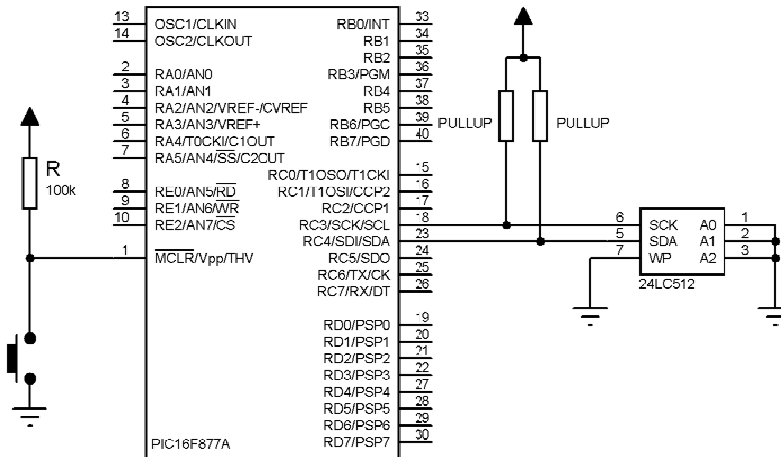
Finalmente el bit R/W, indica si la memoria se escribe o lee. Si R/W= 0, significa que el microcontrolador maestro escribirá en la memoria 24LC512. Si R/W = 1, significa que el maestro leerá un dato de la memoria 24LC512.

Figura 28. Direccinamiento de la memoria 24LC512 como esclavo en Bus I2C.



Fuente: 512K I2C™ CMOS Serial EEPROM, Datasheet.

Figura 29. Conexión de una memoria EEPROM seie 24LC512 al PIC16F877A.



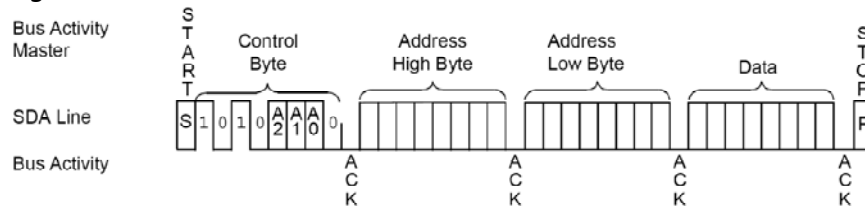
2.6.2 Conexión de la memoria 24LC512 al PIC16F877A. En la figura 29 se observa el diagrama de la conexión de un PIC16F877A con la memoria 24LC512, en modo esclavo. Se aprecia que el Bus I2C lo conforman las conexiones de los pines RC4/SDI/SDA y RC3/SCK/SCL del PIC y las líneas SDA y SCL de la memoria. Como los pines A0, A1 y A2 se conectan a tierra, la dirección para la memoria es b'10100001' en modo Lectura y b'10100000' en modo Escritura. El pin WP también se conecta a tierra para realizar operaciones de escritura sobre la memoria.

2.6.3 Escritura de la memoria 24LC512. La escritura de la memoria por parte del Microcontrolador maestro es una transferencia de datos que tiene un orden bien definido y que sigue los siguientes pasos que hacen parte del protocolo de escritura a esta memoria:

- Primero el microcontrolador maestro envía la condición de START.
- Luego envía el byte de control en modo escritura.
- A continuación el maestro envía el número de la página donde se va a escribir (*Address High Byte*).
- Después el maestro envía el número de posición dentro de la página seleccionada anteriormente (*Address Low Byte*).
- Después se transmiten los datos a escribir. La dirección de la posición de memoria a escribir se incrementa automáticamente.
- Por ultimo el maestro envía la condición de STOP.

La memoria tarda un tiempo máximo de 5ms en escribir un byte en su interior. El procedimiento para la escritura de la memoria se observa en la figura 30.

Figura 30. Protocolo de escritura en la memoria EEPROM serie 24LC512.

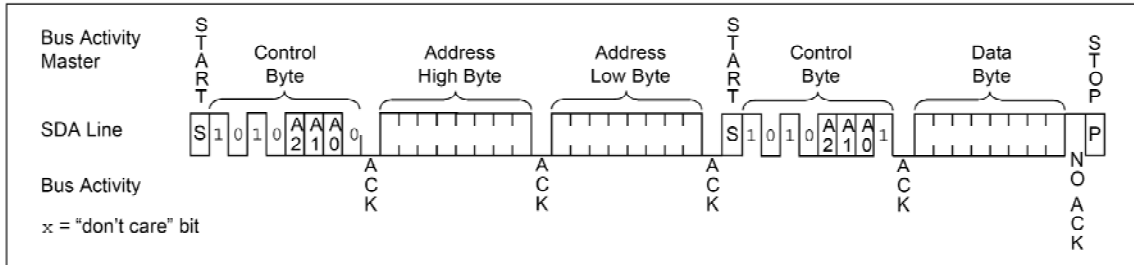


Fuente: 512K I2C™ CMOS Serial EEPROM, Datasheet.

2.6.4 Lectura de la memoria 24LC512. La lectura de los datos de la memoria 24LC512 por parte del microcontrolador sigue el protocolo descrito en la figura 30, el cual comprende los siguientes pasos:

- Primero el microcontrolador maestro envía la dirección como si fuera a escribir lo cual se hace para indicarle a la memoria la dirección desde la cual se va a leer.
- Seguidamente a lo anterior se envía nuevamente la condición de Start.
- A continuación se envía seguidamente el byte de control en modo lectura.
- Después de esto el maestro ya puede leer los datos de la memoria.
- Finalmente se debe enviar una condición de Stop para terminar el proceso de lectura.

Figura 31. Protocolo de escritura en la memoria EEPROM serie 24LC512.



Fuente: 512K I2C™ CMOS Serial EEPROM, Datasheet.

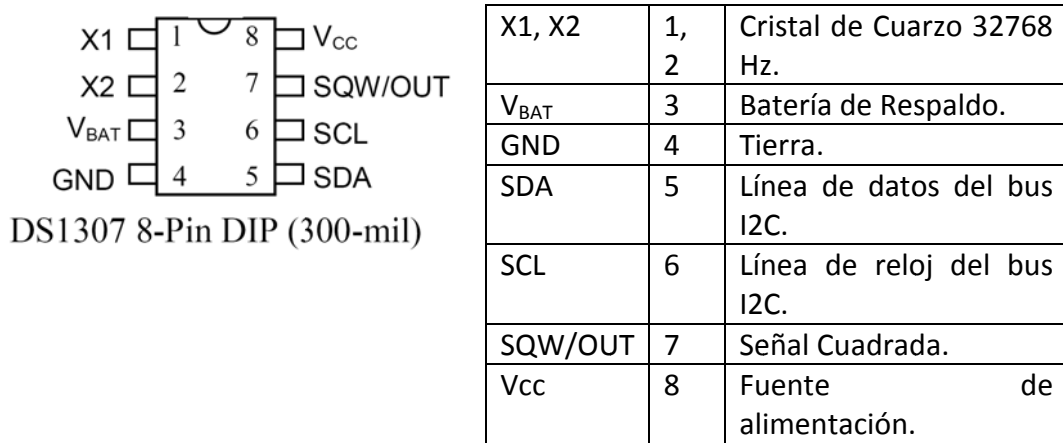
2.7 RELOJ-CALENDARIO DIGITAL DS1307.

En este proyecto es muy importante trabajar en tiempo real, por lo cual se ve la necesidad de utilizar un circuito integrado que se encargue de manejar la hora del sistema y que no dependa directamente del microcontrolador maestro para que funcione de manera continua y sin riesgo de pérdida de la información de la hora actual, en caso de fallo del microcontrolador, con lo cual a través del módulo de almacenamiento de información de registro es posible recuperar la hora y fecha actual de funcionamiento de la máquina sin necesidad de la intervención del microcontrolador principal.

Para ello se escoge el DS1307 que es un reloj en tiempo real RTC (Real Time Clock) que posee un bus de conexión tipo I2C. Este dispositivo es un reloj- calendario digital fabricado por Dallas Semiconductors y es muy útil en aplicaciones que requieren registro y uso de tiempo. Sus principales características son:

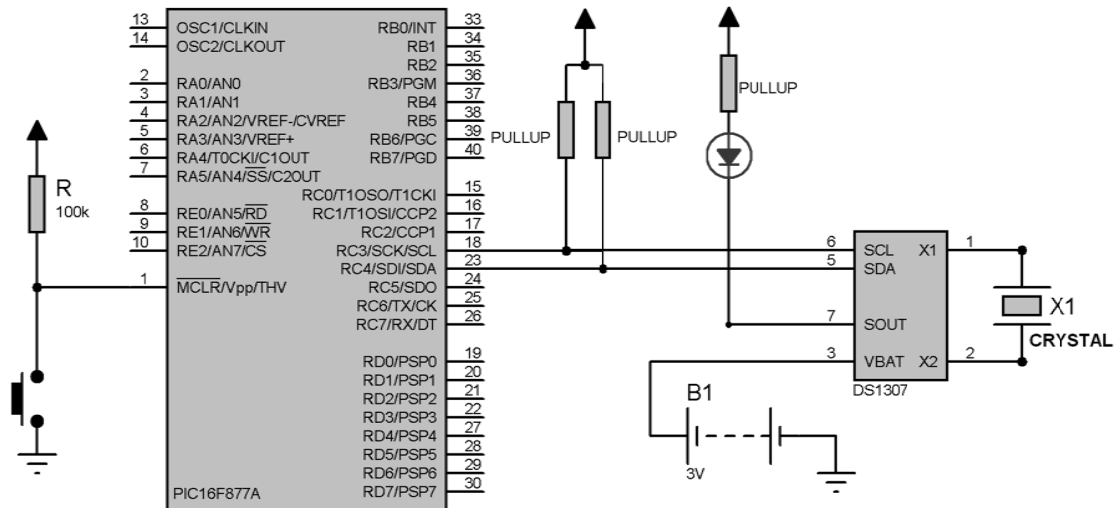
- Cuenta los segundos, minutos, horas, día de la semana (lunes, martes,...), día del mes, meses y años, válidos hasta el 2100.
- Almacena los datos en formato BCD para que se pueda trabajar directamente con ellos.
- Tiene 56 bytes de RAM no volátil para almacenamiento de datos.
- Proporciona una onda programable en su pin SQW/OUT.
- Tiene un circuito interno de respaldo para alimentación en caso de fallo de la alimentación principal por lo cual es capaz de mantener la fecha y hora cuando el sistema se encuentra apagado.
- Se alimenta desde 4.5V hasta 5.5V.
- Posee un modo de bajo consumo de menos de 500nA en modo de respaldo.
- Utiliza un cristal de cuarzo propio de 32.768 Hz para lograr tiempos exactos y no depender del microcontrolador.
- El último día del mes se ajusta automáticamente a 28, 29, 30 o 31, según corresponda. También tiene en cuenta años bisiestos.

Figura 32. Reloj-Calendarario Digital DS1307.



Fuente: Dallas Semiconductor DS1307 64x8 Serial-Real Time Clock Datasheet.

Figura 33. Conexión del PIC16F877A al RTC DS1307.



2.7.1 Conexión del DS1307 al PIC16F877A. En la figura 33 se observa la conexión del microcontrolador maestro al reloj-calendarario digital RTC. Se usan las mismas líneas RC4/SDI/SDA y RC3/SCK/SCL que se conectan a la memoria 24LC512 que posee la máquina. El reloj se conecta a dichas líneas del PIC y utiliza también las resistencias de Pull-Up para su correcto funcionamiento. Debido a que el PIC16F877A está en modo maestro, es capaz de manejar tanto la memoria como el reloj presentes en la máquina aunque se encuentren conectadas en el mismo bus.

2.7.2. Registros del DS1307. En la figura 34, se muestra el mapa de direcciones de la memoria RAM del DS1307. Los principales registros se encuentran comprendidos desde la posición 00h hasta la 07h, desde la posición 08h hasta la 3Fh hay 56 posiciones de memoria RAM que por lo general sirven para almacenar datos de importancia.

Figura 34. Mapa de Memoria del DS1307.

00H	SECONDS
	MINUTES
	HOURS
	DAY
	DATE
	MONTH
	YEAR
07H	CONTROL
08H	RAM
3FH	56 x 8

Fuente: Dallas Semiconductor DS1307 64x8 Serial-Real Time Clock Datasheet.

En la figura 35, se encuentran en detalle los registros del DS1307 que almacenan los datos de interés para el manejo del tiempo real. De ellos se puede destacar la siguiente información de interés:

- El formato de todos los datos esta dado en BCD.
- El bit 7 del registro 00h es el bit de puesta en marcha Clock Halt (CH). Si CH = 0 se pone en marcha el reloj, si CH = 1 se impide el funcionamiento del reloj.
- El formato de las horas puede seguir el modelo americano de 12 horas (bit 12/24 = 1) o el modelo europeo de 24 horas (bit 12/24 = 0).
- El contenido del registro 07h controla la señal cuadrada del pin SQW/OUT, según lo que se nota en la figura 35 y la Tabla 4.

Figura 35. Registros de mayor importancia del DS1307.

	BIT7								BIT0
00H	CH	10 SECONDS			SECONDS				00-59
	0	10 MINUTES			MINUTES				00-59
	0	12 24	10 HR A/P	10 HR	HOURS				01-12 00-23
	0	0	0	0	0	DAY			1-7
	0	0	10 DATE		DATE				01-28/29 01-30 01-31
	0	0	0	10 MONTH	MONTH				01-12
	10 YEAR				YEAR				00-99
07H	OUT	0	0	SQWE	0	0	RS1	RS0	

Fuente: Dallas Semiconductor DS1307 64x8 Serial-Real Time Clock Datasheet.

Figura 36. Registro de control de la señal cuadrada del DS1307.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	0	0	SQWE	0	0	RS1	RS0

Fuente: Dallas Semiconductor DS1307 64x8 Serial-Real Time Clock Datasheet.

El registro de control, se encuentra en la posición 07h y controla la señal cuadrada que puede ser de cuatro valores exactos que dependen de los bits RS1 y RS0 según la tabla 5. También se debe tener en cuenta que el bit 4 SQWE debe estar en alto para que haya una onda cuadrada en el pin SQW/OUT y además de ello el bit 7 OUT indica el nivel lógico en el que se encuentra la señal cuadrada.

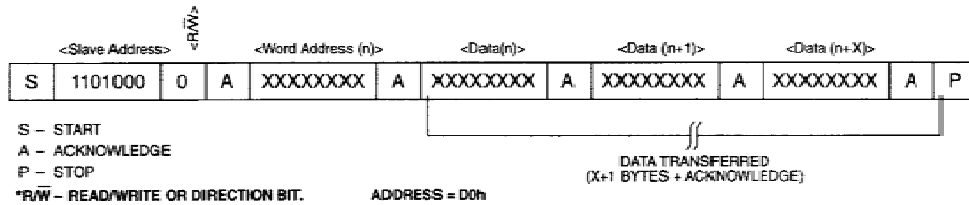
Tabla 4. Frecuencia de la señal cuadrada del DS1307.

RS1	RS0	SQW OUTPUT FREQUENCY
0	0	1Hz
0	1	4.096kHz
1	0	8.192kHz
1	1	32.768kHz

Fuente: Dallas Semiconductor DS1307 64x8 Serial-Real Time Clock Datasheet.

2.7.3 Escritura en el DS1307. El proceso de transferencia de datos del microcontrolador maestro a al DS1307 se hace mediante el bus I2C y se observa en la figura 37. El protocolo se describe a continuación:

Figura 37. Protocolo de escritura del microcontrolador maestro sobre el DS1307



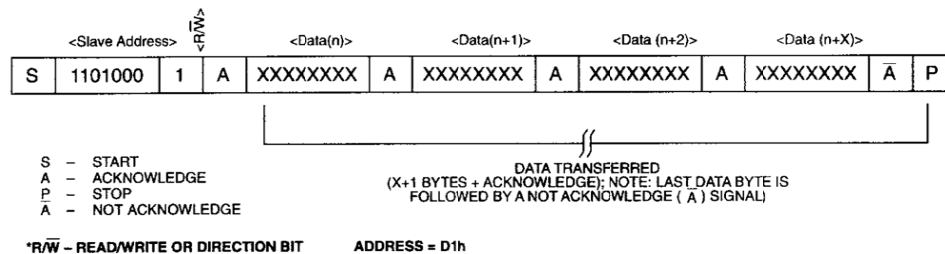
Fuente: Dallas Semiconductor DS1307 64x8 Serial-Real Time Clock Datasheet.

- Primero el microcontrolador maestro envía la condición de Start.
- Luego envía la dirección del DS1307 (Slave Address) en modo escritura que es la b'11010000' o D0h.
- Seguidamente el maestro envía un puntero con la primera dirección del registro a escribir (Word Address).
- Después se transmiten los datos a escribir. La dirección del registro a escribir incrementa automáticamente.
- Cuando termina de escribir el microcontrolador maestro envía la condición de Stop.

2.7.4. Lectura del DS1307. La lectura de datos del DS1307 por parte del microcontrolador maestro se esquematiza según se observa en la figura 38. El protocolo que procede es el siguiente:

- Primero el microcontrolador maestro envía la condición de Start.
- Luego envía la dirección del DS1307 (Slave Address) en modo lectura que es la b'11010001' o D1h.
- Después el maestro lee los datos de los registros. La dirección del registro a leer se incrementa automáticamente.
- Por ultimo el microcontrolador maestro envía la condición de Stop.

Figura 38. Protocolo de lectura del DS1307 por parte del microcontrolador maestro.



Fuente: Dallas Semiconductor DS1307 64x8 Serial-Real Time Clock Datasheet

2.8 PANTALLA LCD ALFANUMÉRICA.

Debido a que la máquina necesita interactuar con el usuario se hace necesario el uso de un Display o Visualizador LCD el cual permite mostrar determinados mensajes al usuario de tal modo que sea posible facilitar al máximo su uso y se entienda con sencillez el estado actual de la máquina.

Para la máquina dispensadora de estampillas se escogió el Display L2432, el cual es un LCD (Liquid Crystal Display) de bajo consumo, pantalla plana, alto contraste, posee un Controlador CMOS interno. El controlador interno es un generador de caracteres ROM/RAM. Todas las funciones del LCD son gestionadas mediante dicho controlador que se puede conectar fácilmente con un microcontrolador, que en este caso es el PIC16F877A. Las principales características del L2432 son:

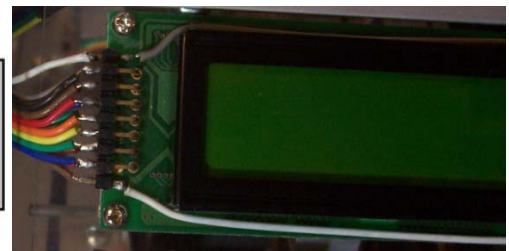
- Bajo consumo de potencia.
- Dos líneas de 24 caracteres cada una.
- Matriz de 5x7 puntos.
- Movimiento del cursor y cambio de su aspecto
- Generador de Caracteres ROM para 192 tipos de caracteres.
- Generador de Caracteres RAM para ocho tipos de caracteres.
- Display Data RAM de 80x8 bits (80 caracteres máximo).
- Interfaz de conexión por bus de 4-bits o bus de 8-bits.
- Diversas instrucciones para varias funciones.
- Circuito oscilador interno.
- Voltaje de operación de 5V.
- Reset automático al encenderse.
- Procesador CMOS.
- Rango de temperatura de operación de 0°C a 50°C.

Figura 39. Aspecto real del display L2432.



El L2432 posee 14 pines descritos la figura 40 y sus respectivas funciones en la tabla 1 del Anexo 2. Se conecta fácilmente a cualquier sistema microcontrolador y posee varias funciones que le permiten diversos usos.

Figura 40. Patillaje del LCD L2432.



2.8.1 Conexión del LCD L2432 al microcontrolador PIC16F877A. Debido a que el proyecto requiere una gran cantidad de líneas para el control de las diversas partes que posee la máquina, es necesario utilizar la menor cantidad de pines para cada aplicación, en este caso se escoge la configuración de conexión de la pantalla LCD mediante comunicación por bus de 4-bits.

2.9 SISTEMA DE MANDOS PARA LA INTERACCION CON EL PERSONAL DE MANEJO Y MANTENIMIENTO.

La máquina dispensadora de estampillas posee tres opciones principales que solo pueden ser modificadas por el personal de mantenimiento o técnico que se encargue de ella debido a que sirven para modificar parámetros que afectan el funcionamiento normal de la máquina. Estas opciones solo se activan cuando la puerta que permite el acceso al interior de la máquina se encuentra abierta, en caso contrario no pueden ser manipuladas, dichas funciones, son:

Borrar Memoria: para ejecutar esta función la máquina posee un pulsador denominado “BORRAR MEMORIA”, que se observa en la figura 42 que al ser activado da la orden al microcontrolador de que ponga en cero (“0”) todas las posiciones de la memoria EEPROM Serial 24LC512 que poseen datos referentes al registro de funcionamiento. Esta opción es útil en caso de que algún mal funcionamiento de la máquina genere errores en el registro, también es de utilidad cuando el numero máximo de monedas que puede contar la máquina se alcance (FFFFh o 65535 como máximo), o simplemente por conveniencia del usuario. En la figura 42 se observa el mensaje que aparece después de ejecutar la opción de borrar el registro de actividad de la máquina.

Figura 41. Conexión del PIC16F877A al LCD L2432.

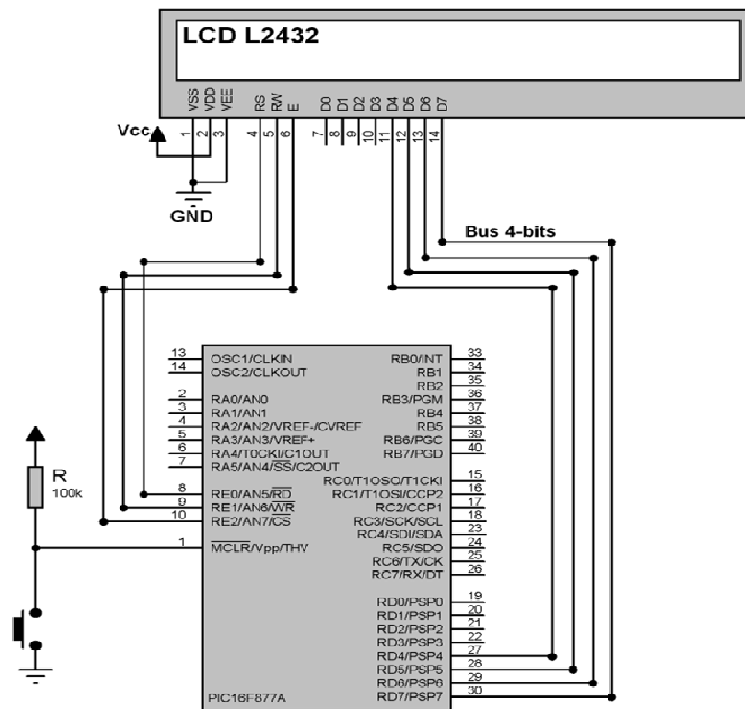


Figura 42. Mensaje asignado para la función borrar memoria.



Ajuste de Hora y Fecha: debido a que la máquina necesita tener actualizada la fecha y hora en la que se encuentra funcionando es necesario que posea controles para modificar dichos parámetros del sistema según lo decida el personal de mantenimiento. Para ello tiene tres pulsadores, los cuales se observan en la figura 43. El pulsador denominado “AJUSTE HORA”, se encarga de llevar a la máquina a la subrutina que permite modificar los registros del DS1307 que se encargan de almacenar la hora y fecha del sistema. Una vez dentro de la subrutina, el usuario puede realizar modificaciones mediante los pulsadores “AJUSTAR SIGUIENTE” e “INCREMENTAR”. Una vez ajustada la hora el sistema inicializa el funcionamiento del reloj-calendario DS1307 para que empiece a oscilar cada segundo, luego pasa nuevamente a esperar órdenes del usuario. Como se observa en la figura 43, en la pantalla LCD de la máquina se visualiza claramente los parámetros de hora y fecha, el indicador de la parte superior derecha muestra que se están modificando, en este caso, los minutos.

Figura 43. Ajuste de hora y fecha.

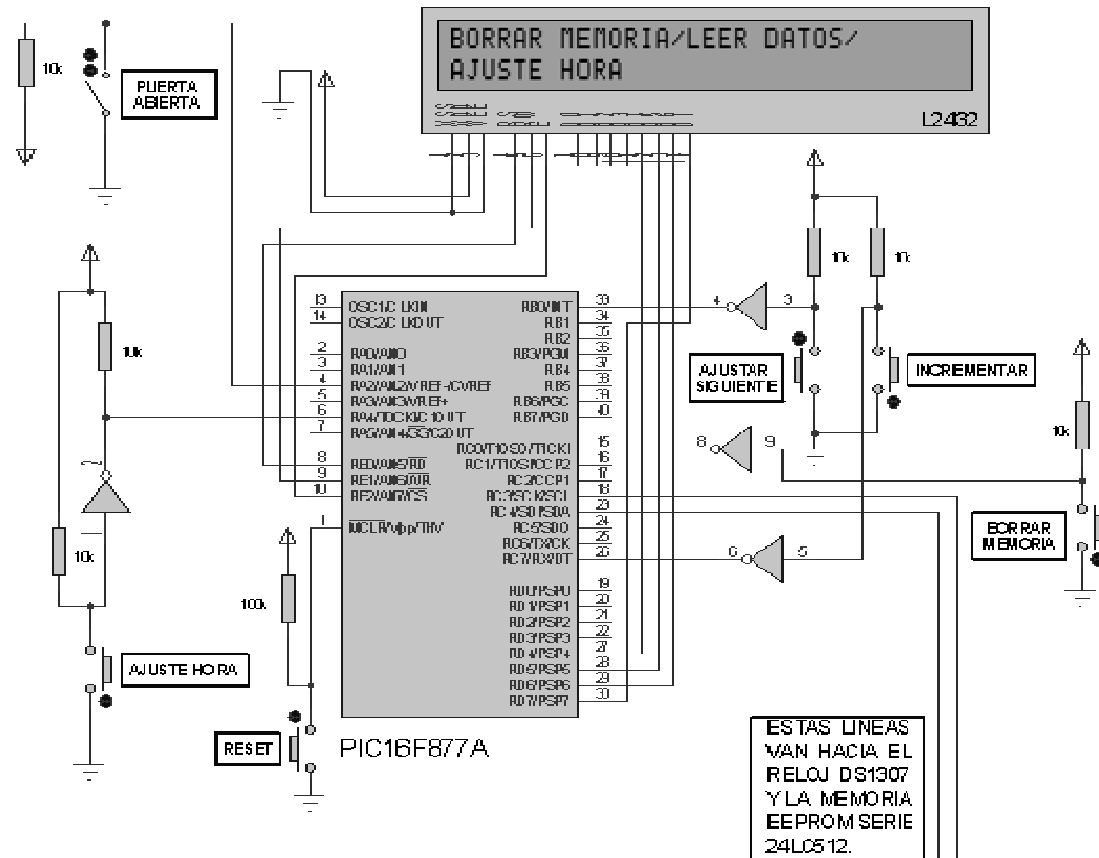


Leer Datos: esta opción simplemente se visualiza en la pantalla LCD de la máquina para que el usuario sepa que el Módulo de Almacenamiento de Información de Registro lee los datos en cualquier momento sin intervención del microcontrolador principal, lo cual presta agilidad al proceso. Esta función se detalla a fondo en la sección referente al módulo de almacenamiento de información de registro.

2.10 DESARROLLO DEL SOFTWARE DEL MICROCONTROLADOR PIC 16F877A DE LA MÁQUINA DE DISTRIBUCIÓN DE ESTAMPILLAS.

La programación del microcontrolador PIC16F877a de la máquina dispensadora de estampillas se describe a través de diagramas de flujo básicos, y es una manera resumida de representar el código fuente del programa real que es muy extenso para ser tratado de manera individual; el código completo del programa comprende todas las ordenes que tiene programado el Software de la máquina de manera que se logren controlar todos los periféricos, señales y funciones que conllevan a que el dispositivo cumpla con el objetivo que inicialmente fue planteado en su etapa de diseño.

Figura 44. Diagrama del sistema de mandos para el personal de mantenimiento.



El software para realizar el programa que contiene el microcontrolador es el MPLAB IDE V7.60.00.00 Certified, que es software libre proporcionado por MICROCHIP que es el fabricante del PIC16F877A. Este software es ideal para el desarrollo del proyecto debido a que maneja el lenguaje ensamblador lo cual permite controlar el dispositivo desde las instrucciones más básicas y brinda la ventaja de conocer con certeza lo que sucede cada instante con el microcontrolador. Además nos permite manejar cada módulo del PIC a voluntad y crear nuestras propias subrutinas de manejo para cada elemento que constituye el sistema.

- **RUTINA PRINCIPAL**

Como se plantea inicialmente el software es el encargado de controlar la totalidad de los sistemas que constituyen la máquina dispensadora de estampillas. La rutina principal describe el proceso en general que lleva a cabo la máquina durante su proceso de control de todas las variables, periféricos y módulos que posee. Su funcionamiento en general se describe en la figura 44.

Inicialmente el programa empieza por determinar la disposición del registro de configuración del PIC16F877A. Se determina que para nuestro proyecto la CONFIGURATION WORD debía tener el valor:

Tabla 5. Valores del Configuration Word del PIC16F877A de la máquina.

Bit13										Bit 0			
CP1	CP0	DEBUG	-	WRT	CPD	LVP	BODEN	CP1	CP0	PWRTE	WDTE	FOSC1	FOSCO
1	1	1		0	0	0	0	1	1	1	0	0	1

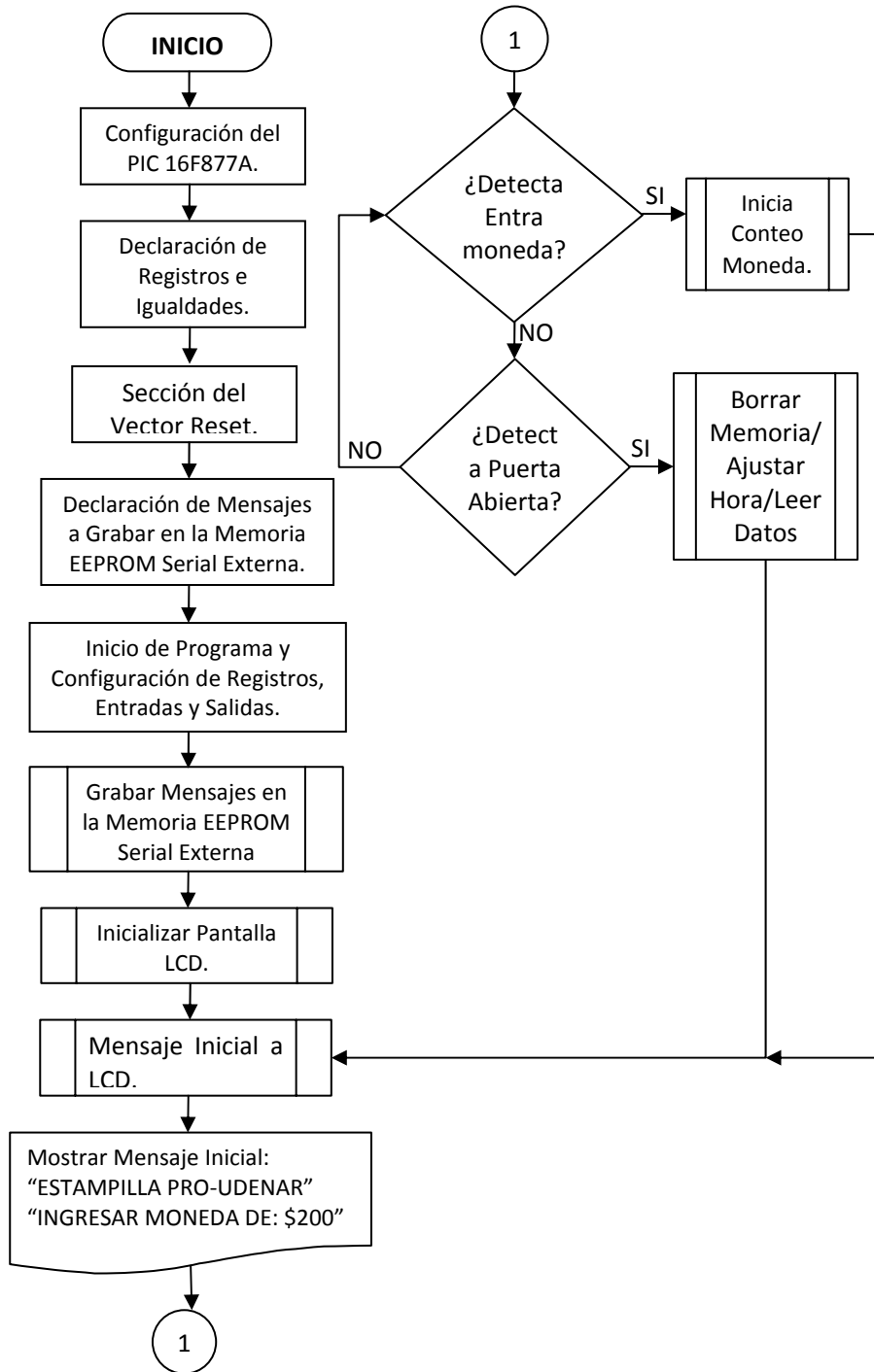
Para que este registro tenga la configuración determinada por la tabla 5, se debe escribir la siguiente línea de código en la primera posición del programa:

```
__CONFIG
_CP_OFF&_WDT_OFF&_PWRTE_ON&_XT_OSC&_LVP_OFF&_BODEN_
OFF&_CPD_OFF.
```

Después de haber configurado la primera línea del programa se pasa a declarar todos los registros importantes, para lo cual se utiliza la función CBLOCK 0x21 la que asigna nombres a los registros de propósito general iniciando en la posición 021h del Banco0 del PIC. Esto ayuda a facilitar la programación debido a que se hace más confuso trabajar con las posiciones de memoria como tal que con los nombres que representan la función que se encuentra desempeñando dicho registro dentro del programa. Dichos registros se observan detalladamente en el código fuente del programa.

También facilita la programación la declaración de varias igualdades que permitan utilizar nombres representativos en lugar de números, por ejemplo, es mas conveniente trabajar con la igualdad "DS1307_DireccionLectura" que con la cifra hexadecimal 0xD1 que es el valor real dicha dirección. De esta manera se asigna nombres a otras direcciones de importancia tales como las del reloj-calendario digital, la memoria EEPROM Serial, caracteres por línea del LCD etc. En esta zona del programa también se asignan igualdades a los pines del PIC en función de la variable que están leyendo de los dispositivos periféricos de la máquina, por ejemplo, la etiqueta "LCD_PinEnable" se asigna al pin 2 del PORTE; así cada vez que se lea la etiqueta "LCD_PinEnable" se esta leyendo el estado del pin 2 del Puerto E.

Figura 45. Diagrama de Flujo de la Rutina Principal.



Todas las asignaciones de variables se encuentran explicadas en detalle en el código del programa.

También se debe configurar todos los puertos del PIC debido a que por la gran cantidad de datos y periféricos que maneja la máquina, se requiere pre-inicializar los registros TRIS de cada puerto para setear correctamente los bits que permita configurar los pines como entradas o salidas. Debido a que los pines del Puerto A y el Puerto E pueden ser configurados como entradas/salidas analógicas o digitales se debe inicializar el Registro ADCON1 de tal manera que todos los pines queden funcionando como entradas/salidas digitales (tabla 6).

Tabla 6. Configuración del registro ADCON1.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0
0	0	0	0	0	1	1	0

Seguidamente los registros TRISA y TRISB se cargan con el valor 0xFF lo cual pone un '1' lógico en todos sus pines determinando así que todos los pines del Puerto A y el Puerto B se configuran como entradas digitales. De la misma manera los registros TRISD y TRISE se cargan con el valor 0x00 de manera que todos los pines de los Puertos D y E quedan configurados como salidas digitales.

Ya que los pines del Puerto C tienen diversas funciones su configuración varía según el módulo que se este usando, por ejemplo, al activar el I2C las líneas RA3/SCL y RA4/SDA se configuran como entradas o salidas según el tipo de transmisión que se encuentre en progreso. Sin embargo el registro TRISC se carga inicialmente con b'10011101' con lo cual los pines 7, 4, 3, 2, 0 son entradas digitales y los pines 6, 5, 1 quedan como salidas digitales.

Finalmente, se configura el registro OPTION_REG (tabla 8), de tal manera que permita el uso del módulo TIMERO que posee el PIC16F877A para ser usado en la subrutina de detección de mondas.

Tabla 8. Configuración del registro OPTION_REG.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RBPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
0	0	0	0	0	1	1	1

La configuración del registro OPTION_REG esta dada de tal manera que se activan las resistencias de Pull-Up del Puerto B, la fuente de reloj del Timer0 se asigna al ciclo de reloj interno, el prescaler se asigna al módulo del Timer0 y la relación de incremento del Timer0 es de 1:256.

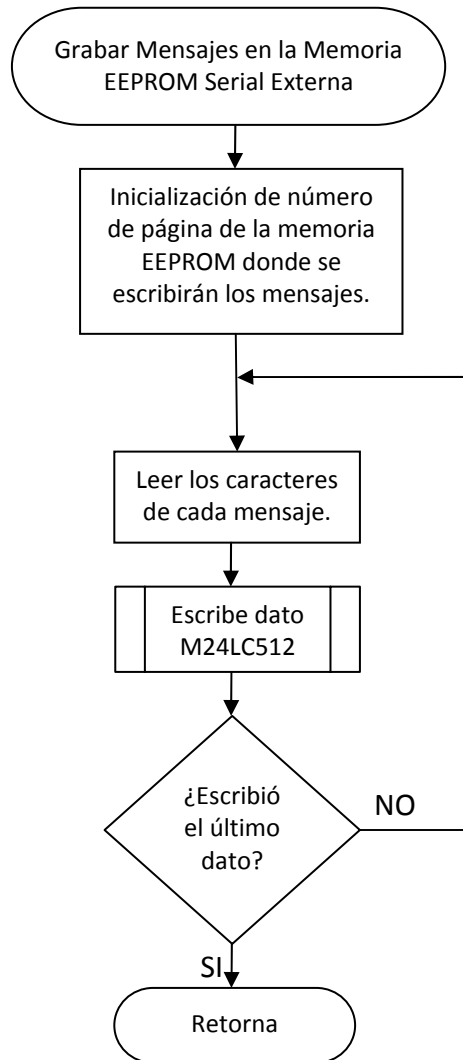
Después de configurar completamente todos los puertos, periféricos y módulos del PIC, empieza el microcontrolador a realizar las acciones más importantes las cuales se encuentran descritas en las subrutinas “Grabar Mensajes en la Memoria EEPROM serial externa”, “Inicializar pantalla LCD” y “Mensaje inicial a LCD” que se amplían a continuación. Después de ejecutar dichas subrutinas el sistema imprime el mensaje “ESTAMPILLA PRO-UDENAR”/“INGRESAR MONEDA DE: \$200” en la pantalla LCD, mediante los cuales se da aviso al usuario que la máquina solo aceptará monedas de \$200 pesos colombianos para funcionar normalmente. Después de mostrar el mensaje en cuestión, la máquina pasa a examinar los pines que le permiten saber si el usuario ingresa una moneda de cualquier valor al sistema con lo cual se sigue a la subrutina de “Conteo moneda y detecta valor” o por el contrario si la compuerta principal de la máquina se encuentra abierta, lo cual da paso a la subrutina “Borrar memoria/Ajustar Hora/Leer Datos” que solo puede ser manejada por el personal de mantenimiento.

- **SUBROUTINA “GRABAR MENSAJES EN LA MEMORIA EEPROM SERIAL EXTERNA”**

Esta subrutina es creada con el fin de grabar los mensajes que aparecen en el módulo LCD dentro de las primeras páginas de la memoria EEPROM Serial 24LC512 de tal manera que sean guardados de manera segura y sobretodo se almacene gran cantidad de mensajes que permitan una interacción fluida con el usuario.

El proceso de grabar datos en la memoria EEPROM externa se hace mediante el uso del módulo I2C del PIC, el cual debe configurarse adecuadamente de manera que no se cometa errores en la transmisión. Para ello se llama la subrutina “EscribeDato_M24LC512” quien se encarga de escribir en la memoria los caracteres leídos de cada mensaje predeterminado por los diseñadores. Posteriormente, al proceso de escritura de un carácter en la memoria, se verifica si es el último y según ello se decide si se pasa a escribir otro o finaliza el proceso de escritura.

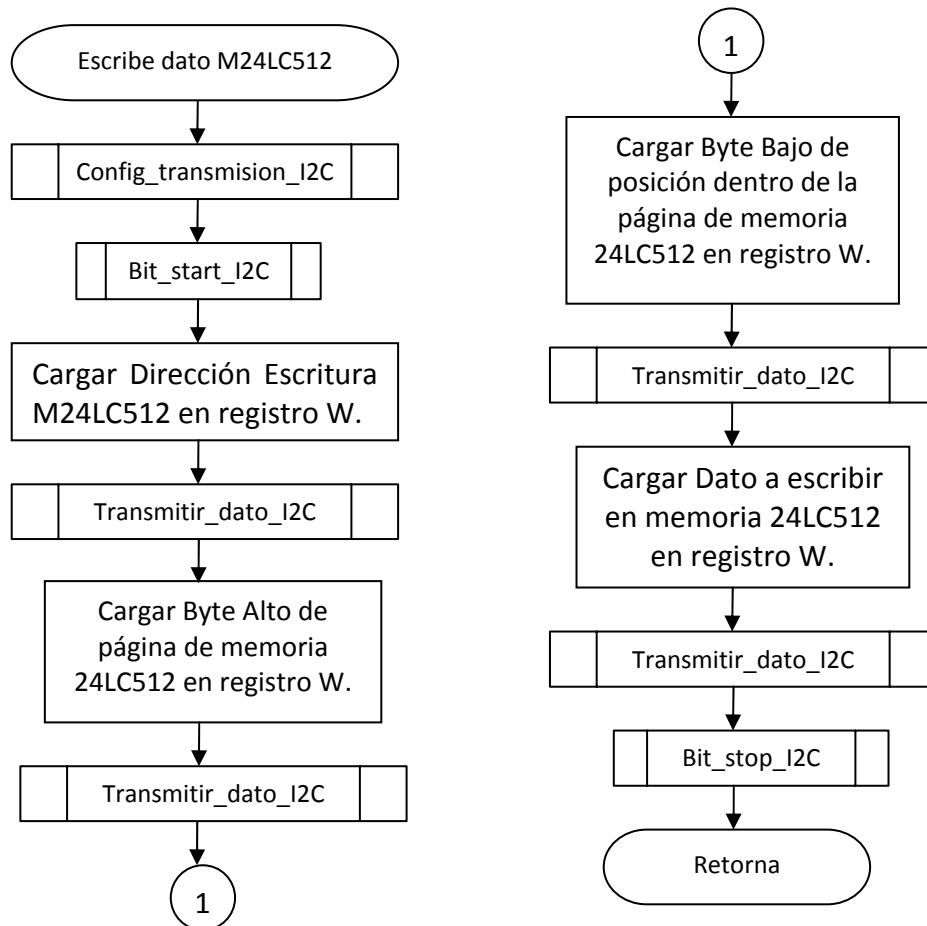
Figura 46. Diagrama de flujo de la subrutina “Grabar mensajes en la memoria EEPROM serial externa”.



- **SUBROUTINA “EscribeDato_M24LC512”**

Dentro de esta subrutina se llama a varias subrutinas más pequeñas y que son de gran importancia ya que se refieren al manejo del módulo I2C del PIC. Estas subrutinas se explican a continuación.

Figura 47. Diagrama de flujo de la subrutina *EscribeDato_M24LC512*.



El proceso de escritura de un carácter o número requiere el uso de varias subrutinas pequeñas que son la base para el uso del módulo I2C del PIC. Estas se describen a continuación.

- **SUBROUTINA “config_transmision_I2C”**

Se encarga de cargar todos los registros de manera que el módulo I2C funcione sin errores y esté adaptado a los requerimientos del sistema y el software de control. Dentro de ella se configura primero que todo el registro SSPADD, el cual se carga con el valor 0x09 que quiere decir que la velocidad de transmisión será de 100 KHz. También se configura el registro PIR1, del cual solo se pone a ‘0’ el bit SSPIF, lo cual quiere decir que no ha ocurrido ninguna interrupción debida al SSP. Finalmente se configura el registro SSPCON

para seleccionar el modo de funcionamiento del módulo I2C del PIC, esto se hace según la tabla 8.

Figura 48. Diagrama de flujo de la subrutina "Config_transmision_I2C"

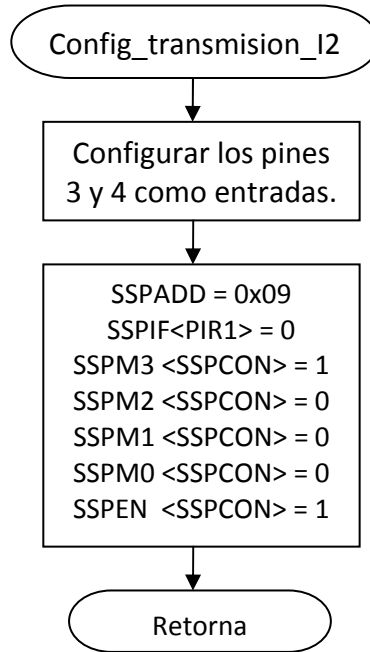


Tabla 8. Registro SSPCON

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
0	0	1	0	1	0	0	0

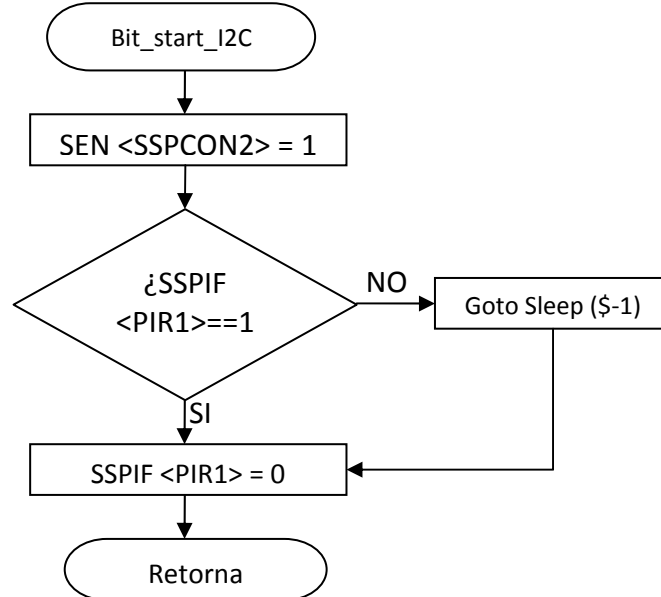
Esto quiere decir que se escoge que el módulo I2C funcione en modo maestro, con frecuencia de oscilación de reloj = $F_{osc} / (4 * (SSPADD + 1))$, y también queda activado el bit SSPEN con lo cual se habilita el uso del puerto serial del PIC y configura los pines SDA y SCL como líneas de datos y reloj del puerto serial.

- **SUBROUTINA "Bit_start_I2C"**

Esta subrutina sirve para enviar una condición de Start en las líneas SDA y SCL de modo que se pueda iniciar la comunicación vía I2C usando el puerto serie del PIC. Esta rutina pone en '1' el bit SEN del registro SSPCON2, con lo cual se inicia la condición de Start.

Luego lee el bit SSPIF del registro PIR, que indica si se ha completado la condición de Start generada por el módulo SSP. En caso de estar en '1' quiere decir que la condición se completo correctamente, seguidamente limpia este bit (0=>SSPIF) y retorna.

Figura 49. Diagrama de flujo de la subrutina "Bit_start_I2C"

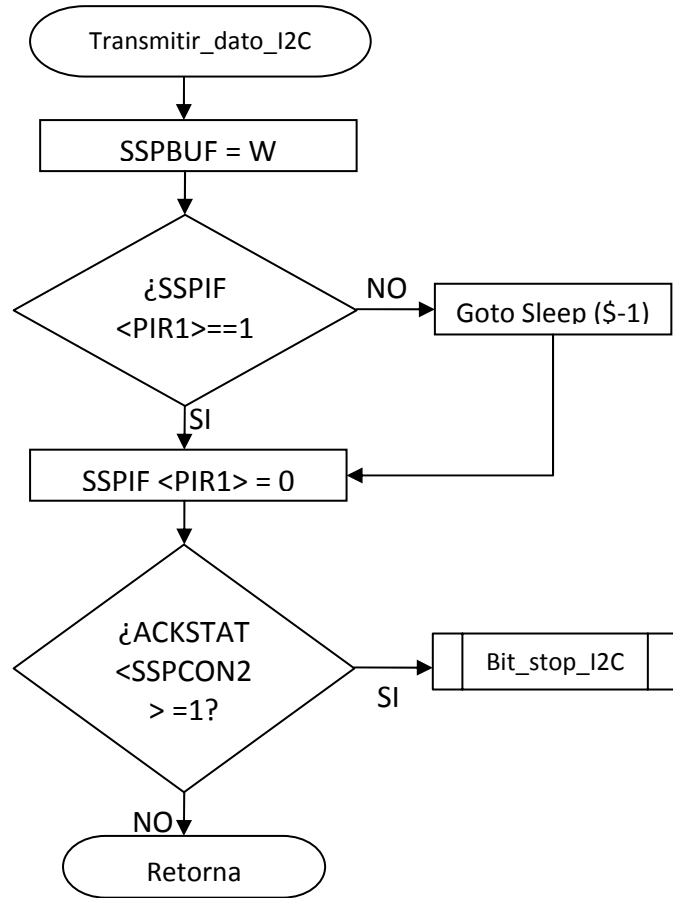


- **SUBROUTINA "Transmitir_dato_I2C"**

Esta rutina se encarga de transmitir un dato previamente cargado en el registro de trabajo W y transmitirlo hacia el esclavo, para ello toma la información del registro W y la copia en el registro SSPBUF, que es el registro de Buffer de transmisión/recepción del módulo SSP. Cuando se acaba de transferir la información del registro W al SSPBUF, se inicia automáticamente la transmisión de datos hacia el esclavo a través de las líneas SDA y SCL.

Para verificar la transmisión se lee el bit SSPIF del registro PIR1 que también sirve para indicar si una transmisión o recepción ha tenido lugar. Inmediatamente este bit se activa, se limpia (0 =>SSPIF) y se pasa a leer el bit ACKSTAT del registro SSPCON2, el cual debe estar en '0' si la transmisión es correcta, es decir, el esclavo ha recibido los datos correctamente. Si no esta en cero salta a la condición de Stop y termina la transmisión.

Figura 50. Diagrama de flujo de la subrutina "Transmitir_dato_I2C"



- **SUBROUTINA "Bit_stop_I2C"**

Esta subrutina sirve para generar una condición de Stop en las líneas SDA y SCL del PIC, para ello se pone en '1' el bit PEN del registro SSPCON2, con lo cual se inicia la condición de Stop. Para verificar si la condición es alcanzada correctamente se lee el bit SSPIF del registro PIR1, si este está en '1' la condición se completa de manera correcta y se procede a limpiar el bit (0=>SSPIF).

Figura 51. Diagrama de flujo de la subrutina "Bit_stop_I2C"

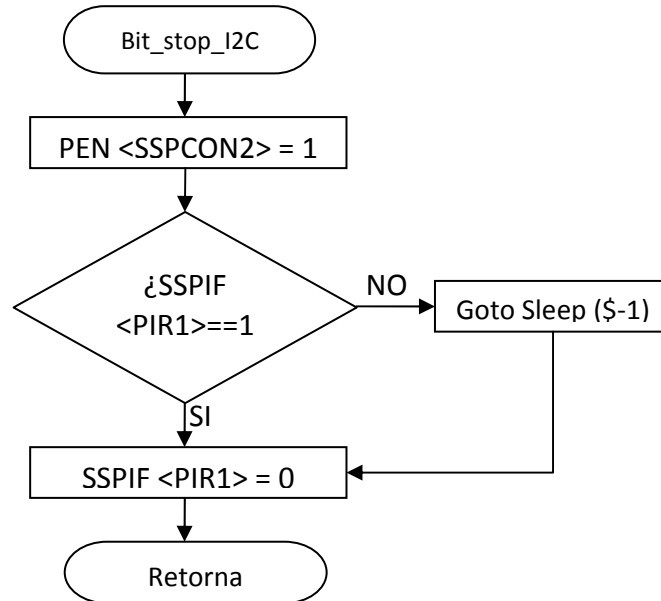
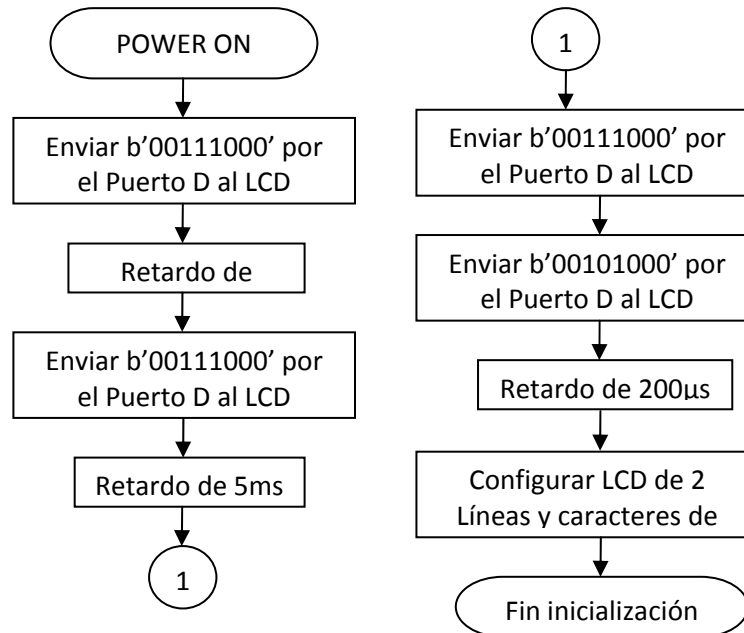


Figura 52. Diagrama de flujo de la subrutina "Inicializar Pantalla LCD"



- ***SUBROUTINA “INICIALIZAR PANTALLA LCD”***

El fabricante de la pantalla LCD L2432 recomienda se siga un proceso exacto para poder inicializar el módulo. Esto consiste en enviar ciertas órdenes secuencialmente e intercalando varios retardos entre ellas, el proceso se puede esquematizar según la figura 52.

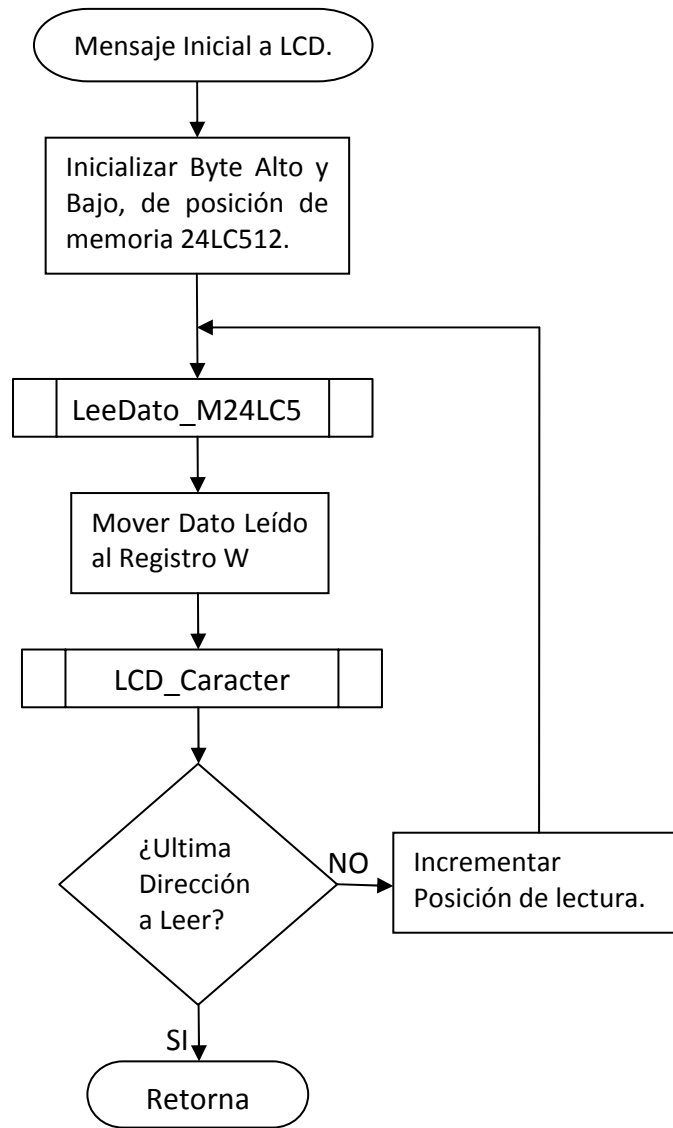
- ***SUBROUTINA “MENSAJE INICIAL A LCD”***

Esta subrutina se encarga de leer el mensaje de la memoria EEPROM serial externa 24LC512 y enviarlo hacia el módulo LCD de manera que en el se pueda visualizar por completo el mensaje escogido. Este es un proceso complejo que se resume mediante el diagrama de flujo de la figura 53 y para su mayor comprensión se refiere al código fuente.

Cabe destacar que para el manejo del módulo LCD se destina los tres bits que posee el Puerto E y los cuatro bits más significativos del Puerto D, del PIC16F877A, estos se encuentran definidos en la sección del código fuente donde se declara los registros importantes y las variables necesarias para el uso del LCD, y se detallan a continuación:

- LCD_Dato: Guarda el dato a escribir en el módulo LCD.
- LCD_GuardaDato: Guarda el dato a enviar al módulo LCD.
- LCD_GuardaTRISD: Guarda la configuración del TRISD para evitar fallos.
- LCD_Auxiliar1: Registro auxiliar para el manejo del módulo LCD.
- LCD_Auxiliar2: Registro auxiliar para el manejo del módulo LCD.
- LCD_ApuntaCaracter: Indica la posición del carácter a visualizar respecto de la posición de todos los mensajes (Posición de la etiqueta “Mensajes”).
- LCD_ValorCaracter: Código ASCII del carácter a visualizar.
- LCD_CursorPosicion: Contabiliza la posición del cursor dentro de la pantalla.
- LCD_CaracteresPorLinea = .24: Número de caracteres por línea de la pantalla.
- LCD_PinRS = PORTE, 0: Este pin se conecta al pin RS de la pantalla LCD.
- LCD_PinRW = PORTE, 1: Este pin se conecta al pin RW de la pantalla LCD.
- LCD_PinEnable = PORTE, 2: Este pin se conecta al pin E (Enable) de la pantalla LCD.
- LCD_BusDatos = PORTD: Define que se utiliza el Puerto D para el manejo de los pines de datos del módulo LCD.

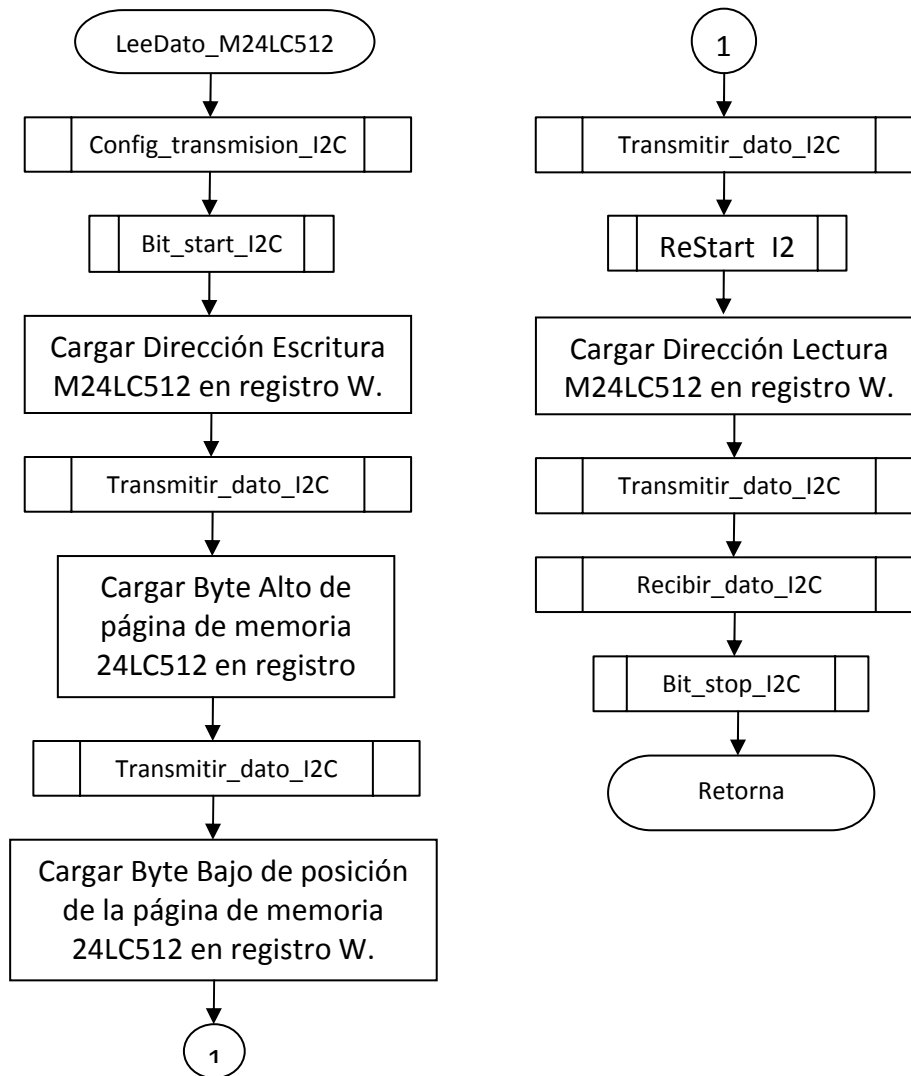
Figura 53. Diagrama de flujo de la subrutina "Mensaje inicial a LCD".



- **SUBROUTINA "LeeDato_M24LC512"**

Esta subrutina permite leer cualquier dato almacenado en una determinada posición de memoria dentro de la EEPROM Serial Externa 24LC51, ya sean caracteres o números. Esta subrutina hace uso de otras ya explicadas anteriormente y se basa en el protocolo de escritura de la memoria que se explica en la sección anteriormente en este documento.

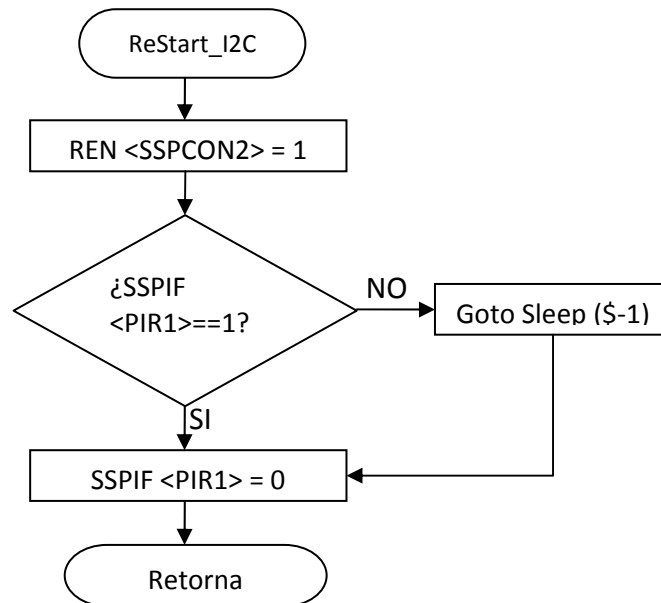
Figura 54. Diagrama de flujo de la subrutina "LeeDato_M24LC512".



- **SUBROUTINA "ReStart_I2C"**

Esta subrutina permite enviar una condición de "ReStart" o Start Repetido, lo cual es de gran ayuda para simplificar la programación porque sin esta condición se debe enviar primero una condición de Stop, esperar un retardo en tiempo y posteriormente enviar de nuevo la condición de Start para iniciar el proceso de lectura. Al igual que otras subrutinas que utilizan el módulo I2C del PIC16F877A esta también manipula la configuración del registro SSPCON2, ya que para enviar la condición de "ReStart" se debe poner en '1' el bit REN. Inmediatamente se pasa a leer el bit SSPIF<PIR1>, cuando este en '1' lógico quiere decir que el módulo I2C completo de poner la condición de ReStart en las líneas SDA y SCL, mientras tanto el sistema espera en modo Sleep.

Figura 55. Diagrama de flujo de la subrutina "ReStart_I2C".



- **SUBROUTINA "Recibir_dato_I2C"**

Esta subrutina se encarga de leer los datos del dispositivo esclavo direccionado por el PIC que en este caso es la memoria EEPROM Serial 24LC512 pero también servirá para otros procesos mas adelante.

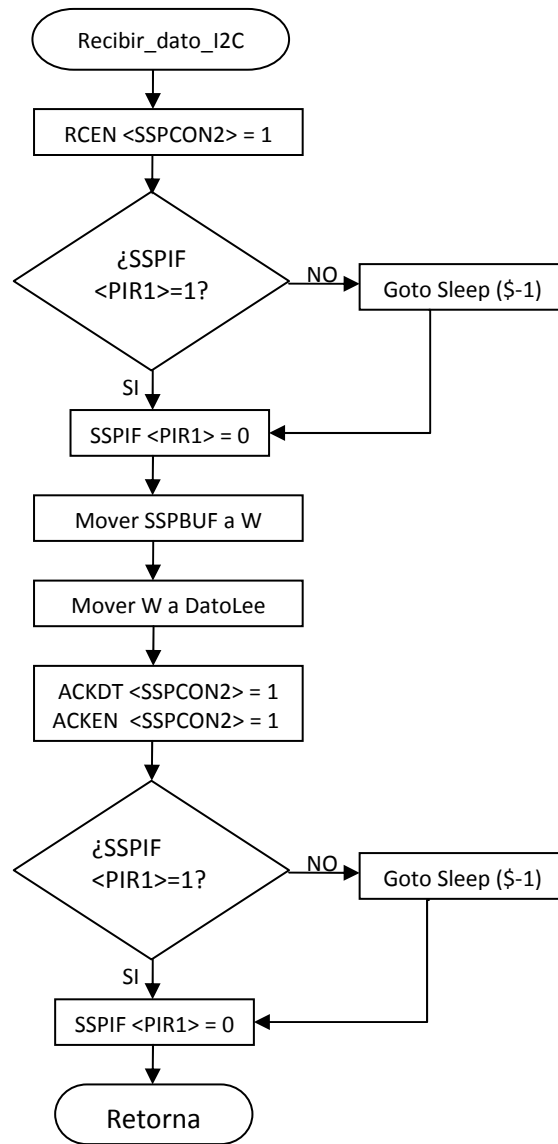
En primer lugar esta rutina pone en '1' el bit RCEN del registro SSPCON2, con el fin de habilitar el modo maestro de recepción del módulo I2C. A continuación se lee el bit SSPIF del registro PIR1; si esta en '1' quiere decir que ha sucedido una recepción por parte del módulo, se debe llevar a cero por software.

Si esta en '0' espera en modo Sleep a que el módulo termine la recepción de datos. Los datos recibidos se graban en el registro SSPBUF, de donde pueden ser transferidos a otro registro, en este caso el registro de denomina "DatoLee".

Una vez completado esto, el maestro pone el bit de reconocimiento de que los datos han llegado, para se debe poner un '1' en el bit ACKDT del registro SSPCON2 y también se pone un '1' en el bit ACKEN para permitir que el bit de reconocimiento o "Acknowledge" este activo en las líneas SDA y SCL.

Después de haber puesto el bit de reconocimiento debe probar de nuevo el bit SSPIF del registro PIR1 para saber si la condición de "Acknowledge" se ha terminado por el módulo SSP. Finalizado esto se retorna a la subrutina principal.

Figura 56. Diagrama de flujo de la subrutina "Recibir_dato_I2C".



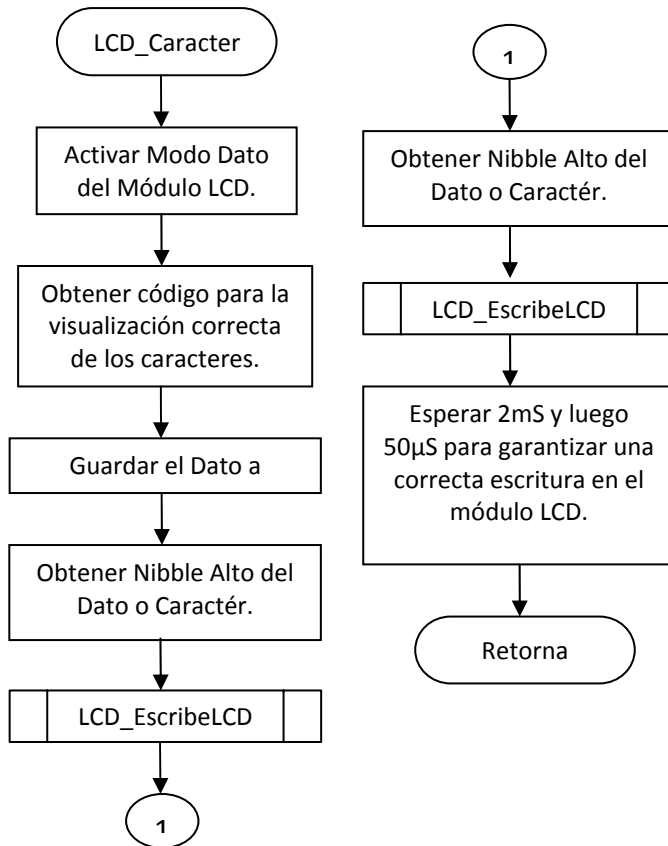
- **SUBROUTINA "LCD_Caracter"**

Esta subrutina se encarga de escribir en la memoria DDRAM del LCD L2432 el carácter ASCII introducido a través del registro W. Trabaja en modo dato. También se encarga de obtener el código ASCII del carácter para su correcta visualización. Guarda el dato y lo envía usando el modo de transmisión de 4-bits del módulo LCD.

Esta subrutina se compone de varios procesos y rutinas internas que conlleva el manejo del módulo LCD de la máquina, por lo tanto para entender a fondo la parte del programa

dedicada para tal fin se debe analizar en detalle el código fuente del programa. En la figura 57 se observa este proceso de manera general.

Figura 57. Diagrama de flujo de la subrutina “LCD_Caracter”.



- **SUBROUTINA “LCD_EscribeLCD”**

Esta subrutina permite enviar el dato del registro de trabajo W al bus de datos y producir un pequeño pulso en el pin “Enable” del LCD. Para no alterar el contenido de las líneas de la parte baja del Puerto D que no son utilizadas para el LCD (pines RD3:RD0), primero se lee estas líneas y después se vuelve a enviar este dato sin cambiarlo. Para ello primero se queda con el nibble alto del dato y lo guarda. Luego lee la información actual de la parte baja del puerto D que no se debe alterar, con esto se logra enviar la parte alta del dato de entrada y en la parte baja lo que había antes.

Esta rutina también tiene en cuenta que se debe almacenar la configuración que tenía antes TRISD, luego se configura las cuatro líneas de la parte alta del puerto D como salidas y las cuatro líneas de la parte baja se dejan como estaban.

Finalmente, se debe recuperar el dato a enviar y luego se lo envía al módulo LCD. Para ello el programa permite el funcionamiento de la pantalla mediante un pequeño pulso y termina impidiendo el funcionamiento del LCD. Para terminar se restaura el antiguo valor en la configuración del Puerto D y retorna.

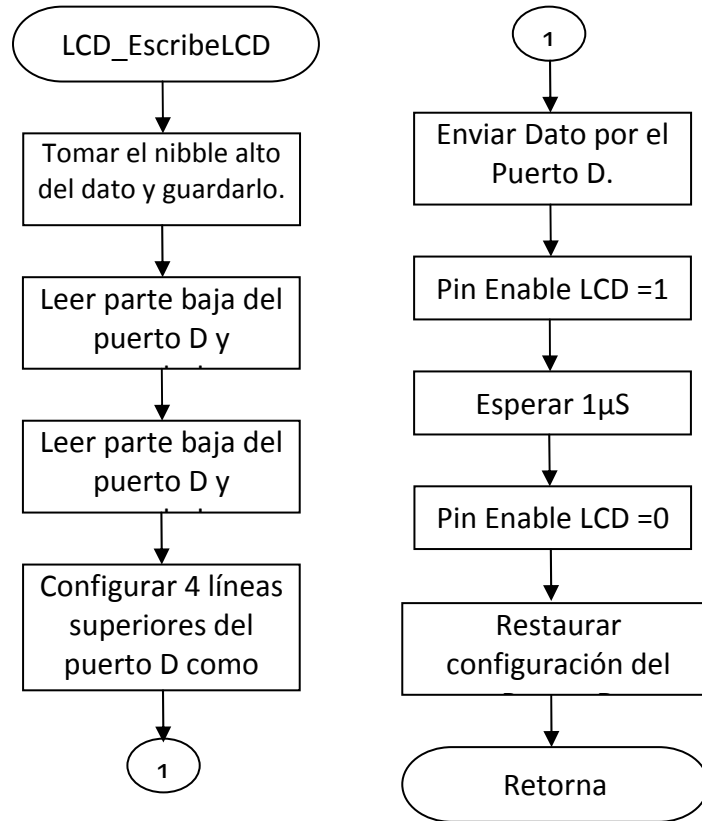
- **SUBROUTINAS PARA CONTEO DE MONEDAS Y DETECCION DE VALOR DE LA MONEDA.**

Como ya se menciona antes, la rutina principal se encarga permanentemente de testear las dos entradas principales del sistema que son el caso en que entre una moneda de cualquier valor para realizar una petición de estampilla o sello, o que se abra la puerta de la máquina, lo cual indicaría que se desea realizar acciones de ajuste y mantenimiento a la máquina.

Esta subrutina y las relacionadas con ella se encargan de controlar el sistema cuando el usuario normal ingresa una moneda al interior del sistema, se encarga de incrementar los contadores respectivos según las señales que reciba del sistema detector de monedas; así por ejemplo, si los sensores de moneda entregan señales indicando que entra una moneda de \$200 pesos, el software de la máquina incrementa el contador de monedas de \$200 y pasa a determinar si el detector validó o no dicha moneda para tomar la acción respectiva.

Las subrutinas de esta parte del software de control de la máquina esta diseñada de la siguiente manera: después de que el sistema detecta la entrada de una moneda pasa a verificar si esta activado el sensor de tamaño de la moneda de \$50, si este no se activa durante cierto límite de tiempo se deduce que la moneda que ingresó al sistema es de un tamaño menor a la de \$50 y es sistema no la acepta. Luego retorna a la rutina principal. En caso de que se active el sensor de tamaño de \$50, el sistema pasa a verificar si se activa el sensor de tamaño de la moneda de \$100; si no se activa quiere decir que la moneda que ingresó el usuario era de \$50 pesos, el sistema la rechaza y pasa a la rutina principal. De este modo se hace la detección para todas las denominaciones que actualmente se encuentran vigentes, que son las de \$50, \$100, \$500 y \$200 pesos. La máquina tiene un último sensor de tamaño en caso de que se inserte una moneda u objeto no válido cuyo diámetro sea mayor que la moneda de \$200 pesos.

Figura 58. Diagrama de flujo de la subrutina “LCD_EscribeLCD”.



Las subrutinas de esta parte del software de control de la máquina esta diseñada de la siguiente manera: después de que el sistema detecta la entrada de una moneda pasa a verificar si esta activado el sensor de tamaño de la moneda de \$50, si este no se activa durante cierto límite de tiempo se deduce que la moneda que ingresó al sistema es de un tamaño menor a la de \$50 y es sistema no la acepta. Luego retorna a la rutina principal. En caso de que se active el sensor de tamaño de \$50, el sistema pasa a verificar si se activa el sensor de tamaño de la moneda de \$100; si no se activa quiere decir que la moneda que ingresó el usuario era de \$50 pesos, el sistema la rechaza y pasa a la rutina principal. De este modo se hace la detección para todas las denominaciones que actualmente se encuentran vigentes, que son las de \$50, \$100, \$500 y \$200 pesos. La máquina tiene un último sensor de tamaño en caso de que se inserte una moneda u objeto no válido cuyo diámetro sea mayor que la moneda de \$200 pesos.

Esta parte del software se encarga de determinar y dar aviso al usuario de que clase de moneda esta ingresando al sistema y, en caso de ser necesario, dice que para obtener un sello o estampilla se debe ingresar una moneda de \$200.

Para almacenar los datos de los contadores de monedas ingresadas al sistema se destina utilizar dos Bytes para cada uno, es decir, cada contador puede incrementarse desde '0000h' hasta 'FFFFh' o sea 65535. Esto con el fin de dar un rango bastante amplio para que la máquina pueda estar en uso sin necesidad de leer el registro de la máquina. Se determina que los datos de los contadores de monedas de la máquina se encuentren guardados en la página 3Fh a partir de la posición 00h, la dirección de la EEPROM serial externa 24LC512 donde se encuentra cada dato es:

- ContMonedasL = 3F00h, es el byte bajo del contador de número de monedas ingresadas a la máquina.
- ContMonedasH = 3F01h, es el byte alto del contador de número de monedas ingresadas a la máquina.
- Cont50pL y Cont50pH, que son los bytes bajo y alto del contador de monedas de \$50 pesos ingresadas a la máquina. Se encuentran en la dirección 3F02h y 3F03h.
- Cont100pL y Cont100pH, que son los bytes bajo y alto del contador de monedas de \$100 pesos ingresadas a la máquina. Se encuentran en la dirección 3F04h y 3F05h.
- Cont500pL y Cont500pH, que son los bytes bajo y alto del contador de monedas de \$500 pesos ingresadas a la máquina. Se encuentran en la dirección 3F06h y 3F07h.
- ContOtroL y ContOtroH, que son los bytes bajo y alto del contador de monedas de otro valor desconocido que se ha ingresadas a la máquina. Se encuentran en la dirección 3F08h y 3F09h.
- Cont200pL y Cont200pH, que son los bytes bajo y alto del contador de monedas de \$200 pesos ingresadas a la máquina. Se encuentran en la dirección 3F0Ah y 3F0Bh.
- ContRechL y ContRechH, que son los bytes bajo y alto del contador de monedas de \$200 pesos rechazadas por el sistema detector de monedas. Se encuentran en la dirección 3F0Ch y 3F0Dh.
- ContAceptaL y ContAceptaH, que son los bytes bajo y alto del contador de monedas de \$200 pesos ingresadas a la máquina y que fueron validadas como buenas. Se encuentran en la dirección 3F0Eh y 3F0Fh.
- ContSelloL y ContSelloH, que son los bytes bajo y alto del contador de cantidad de sellos que han entregado. Se encuentran en la dirección 3F10h y 3F11h.

- ContEstampL y ContEstampH, que son los bytes bajo y alto del contador de cantidad de estampillas que entrega la máquina. Se encuentran en la dirección 3F12h y 3F13h.

Lo que se observa en la figura 59, es el inicio de la subrutina que controla el proceso de conteo de monedas que ingresan a la máquina. Lo primero es leer la posición 3700h de la memoria EEPROM Externa 24LC512, donde se encuentra almacenado el byte bajo del contador de monedas ingresadas a la máquina denominado “ContMonedasL”.

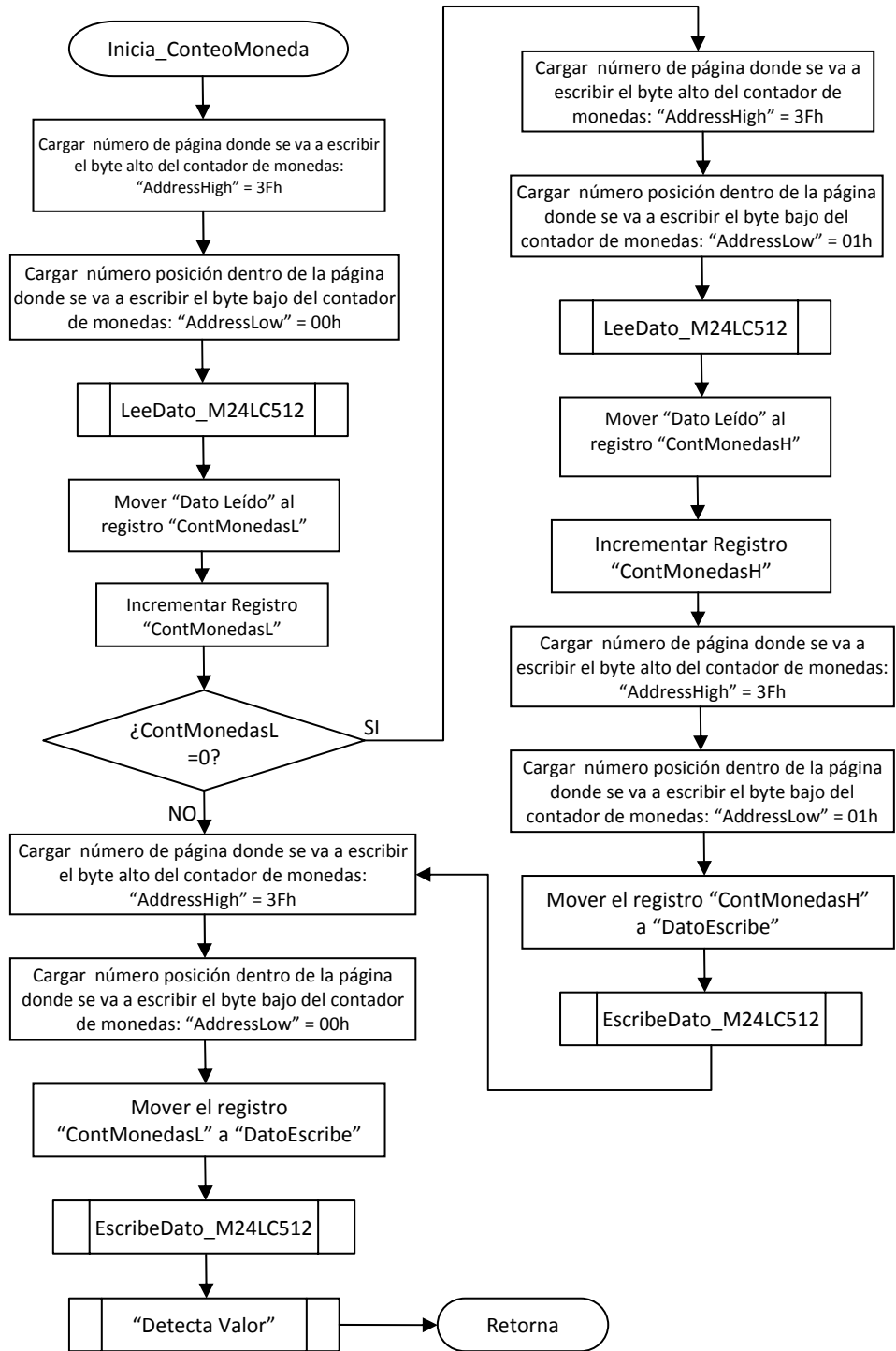
Este registro se incrementa en ‘1’ para registrar la entrada de una moneda, si supera su valor máximo ‘FFh’ y pasa nuevamente a ‘0’ el programa salta a leer nuevamente la memoria EEPROM Externa, pero esta vez carga la página 3F01h que es donde se encuentra almacenado el byte alto del contador de monedas ingresadas a la máquina llamado “ContMonedasH”. En este caso el registro también se incrementa en ‘1’. La rutina para ambos casos finaliza grabando los datos guardados en los registros “ContMonedasL” y “ContMonedasH” en sus respectivas posiciones en la EEPROM Externa.

Algunas subrutinas como “LeeDato_M24LC512” y “EscribeDato_M24LC512” se adaptan perfectamente a la rutina actual y por lo tanto solo se explica aquellas que no se tratan anteriormente. También se debe aclarar que los registros “Dato Leído” y “Dato Escribe” son posiciones de memoria del PIC16F877A que se usan para guardar temporalmente los datos que se leen de la memoria EEPROM Externa y los que se graba en ella, respectivamente.

- **SUBRUTINAS DETECCION DE VALOR DE LA MONEDA.**

Cuando el sistema detecta que una moneda ingresó en el sistema, llama a la rutina “Inicia_ConteoMoneda”, la cual hace un llamado a la subrutina “Detecta Valor”, que se encarga de leer las señales que provienen del sistema de detección y validación de monedas. Esta parte del software determina el valor de la moneda que ingresa el usuario en base a las señales entregadas por los sensores de tamaño y luego pasa a verificar si la moneda es aceptada o no por el detector de material de la moneda.

Figura 59. Diagrama de flujo para la subrutina "Inicia_ConteoMoneda".



Las señales que envía el sistema sensor de monedas se entregan en los siete pines más significativos del Puerto B del PIC16F877A, y dentro del programa cada pin se encuentra determinado por un nombre específico que destaca su función, estos se han denominado de la siguiente manera:

- EntraMoneda = PORTB, 1: Define la entrada del sensor óptico que informa del ingreso de una moneda de cualquier denominación dentro de la máquina.
- Moneda50pe = PORTB, 2: Define la entrada del sensor óptico que permite identificar las monedas de tamaño igual o superior a la de \$ 50 (pesos Colombianos).
- Moneda100pe = PORTB, 3: Define la entrada del sensor óptico que permite identificar las monedas de tamaño igual o superior a la de \$100 (pesos Colombianos).
- Moneda200pe = PORTB, 4: Define la entrada del sensor óptico que permite identificar las monedas de tamaño igual o superior a la de \$200 (pesos Colombianos).
- Moneda500pe = PORTB, 5: Define la entrada del sensor óptico que permite identificar las monedas de tamaño igual o superior a la de \$500 (pesos Colombianos).
- MonedaAccept = PORTB, 6: Define la entrada del sensor óptico que permite identificar si una moneda de \$200 pesos Colombianos fue aceptada o no por el Sistema Sensor de Moneda.
- MonedaOtro = PORTB, 7: Define la entrada del sensor óptico que permite identificar las monedas de tamaño superior a la de \$500 (pesos Colombianos).
- OpcionSello = PORTA, 0: Permite determinar cuantas veces fue elegida la opción de Sellar Tiquete Aéreo.
- OpcionEstamp = PORTA, 1: Permite determinar cuantas veces fue elegida la opción de Entregar Estampilla.

Se utiliza un registro auxiliar llamado TMR0aux, para las subrutinas que emplean el manejo del módulo Timer0 del PIC, este registro almacena el número de veces que se tiene que repetir el conteo del Timer0 para poder testear la señal de cada sensor de tamaño cada 20ms durante 1 segundo.

Figura 60. Diagrama de flujo para la subrutina "DetectaValor".

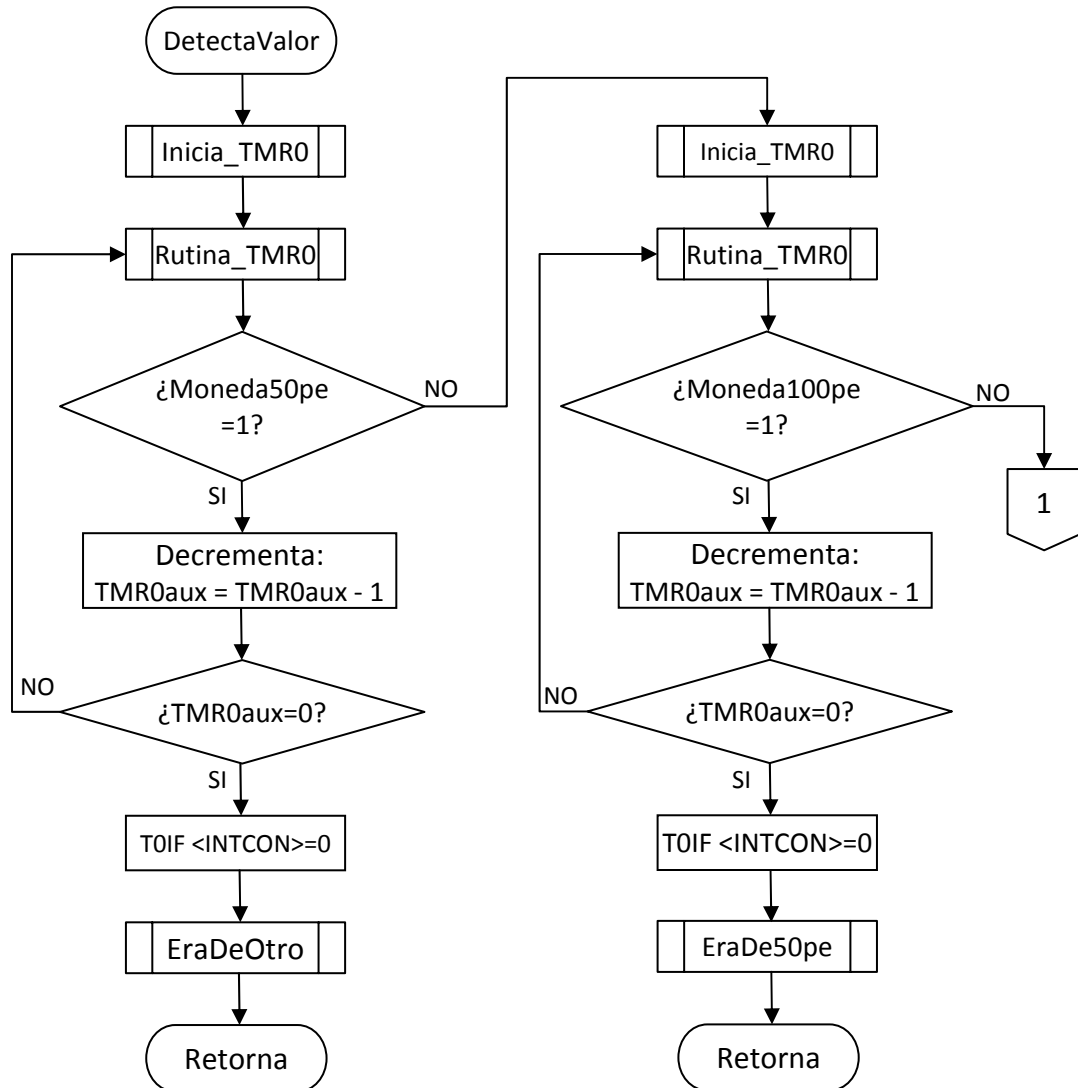


Figura 61. Diagrama de flujo para la subrutina "DetectaValor".

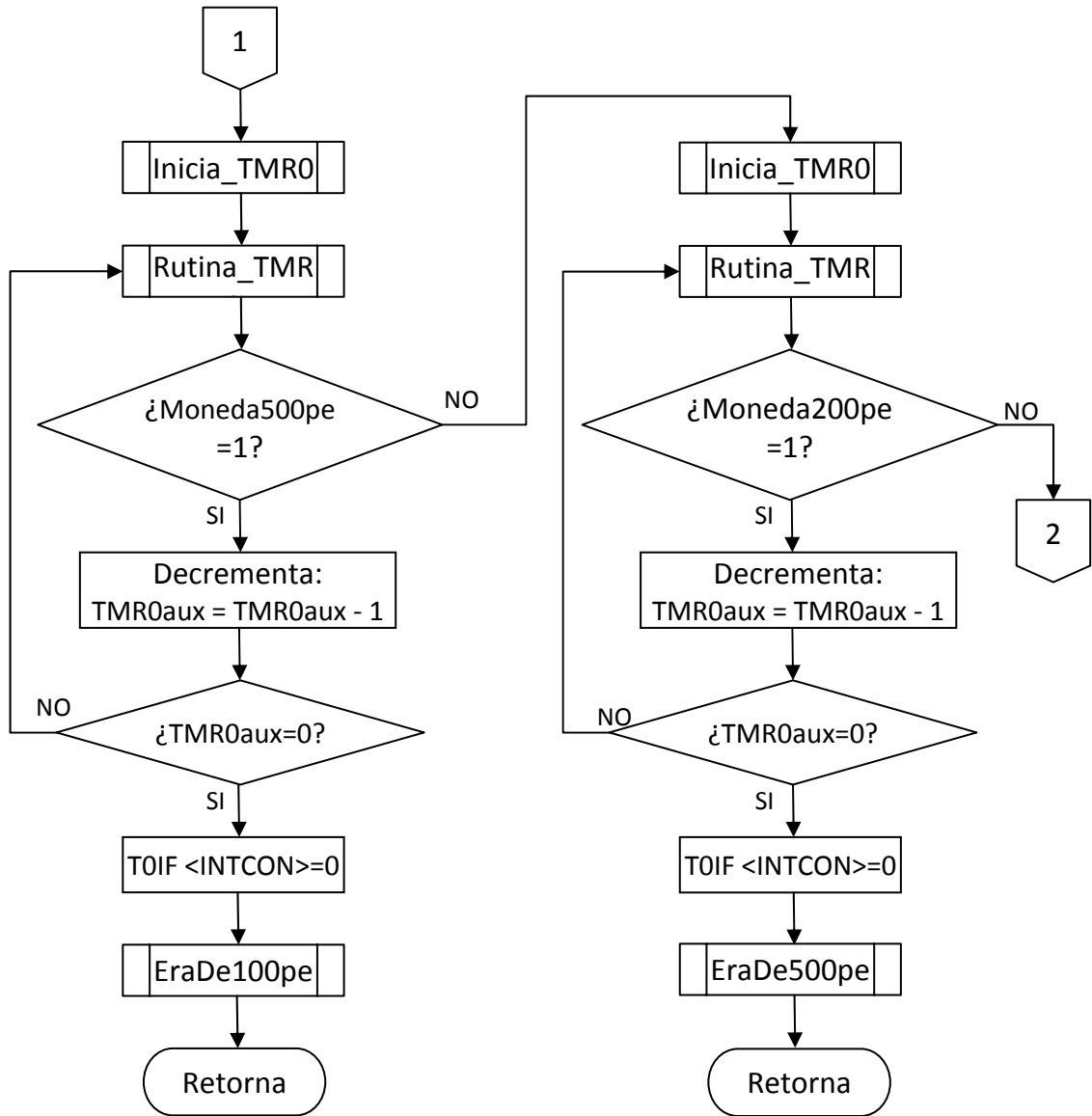
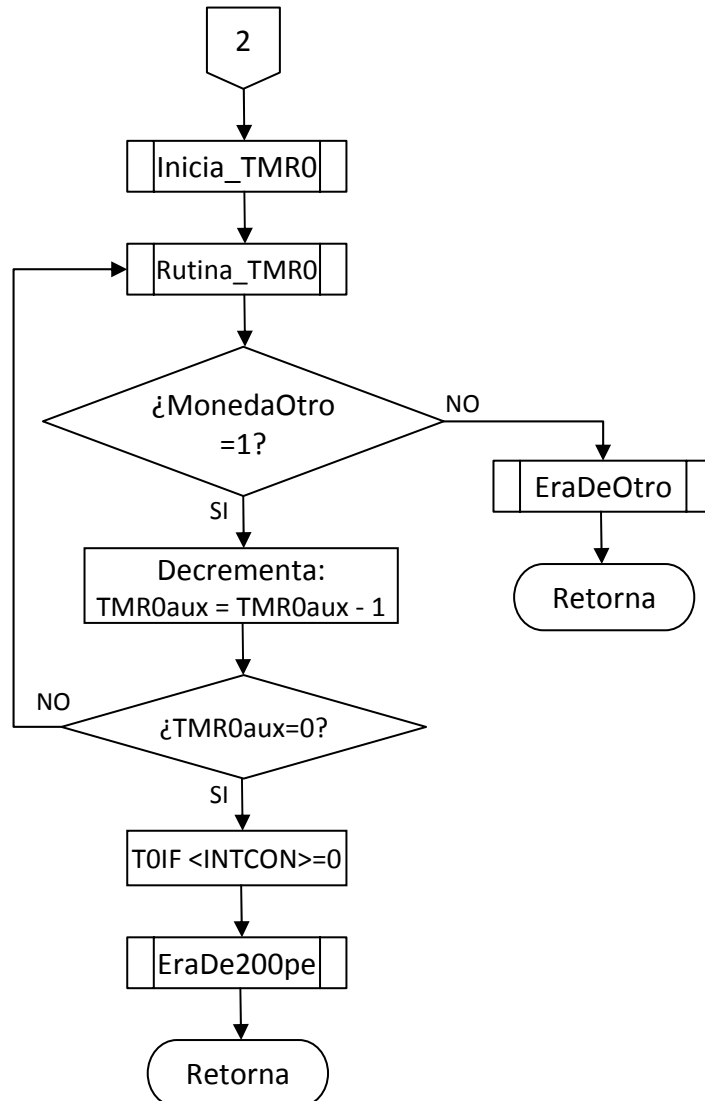


Figura 62. Diagrama de flujo para la subrutina "DetectaValor".

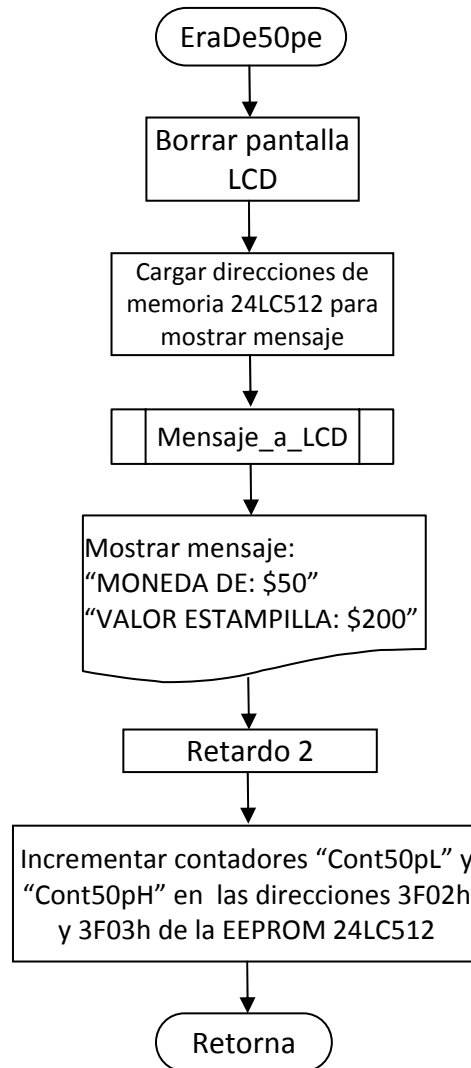


- **SUBROUTINA "EraDe50pe".**

Esta subrutina se encarga de mostrar en la pantalla LCD de la máquina un mensaje indicando que el usuario ingresa una moneda de \$50 pesos, la cual no es válida para que el sistema entregue un sello o una estampilla. También se encarga de registrar este suceso utilizando los registros "Cont50pL" y "Cont50pH", que son los bytes bajo y alto del contador de monedas de \$50 pesos ingresadas a la máquina. Para incrementar los contadores de las monedas de \$50 pesos, se utiliza un proceso exactamente igual al de la subrutina "Inicia_ConteoMoneda" (Figura 59), por lo cual este proceso se resume en un solo bloque que describa su función.

Lo único que cambia para este caso actual son los nombres de los registros y las posiciones de la memoria EEPROM serial externa donde se encuentran almacenados, y el echo de que no se llamaría a la subrutina “DetectaValor” sino que se retorna directamente al programa principal donde se muestra permanentemente el mensaje inicial hasta que ingrese otra moneda y se detecte y muestre su respectivo valor.

Figura 63. Diagrama de flujo para la subrutina “EraDe50pe”.

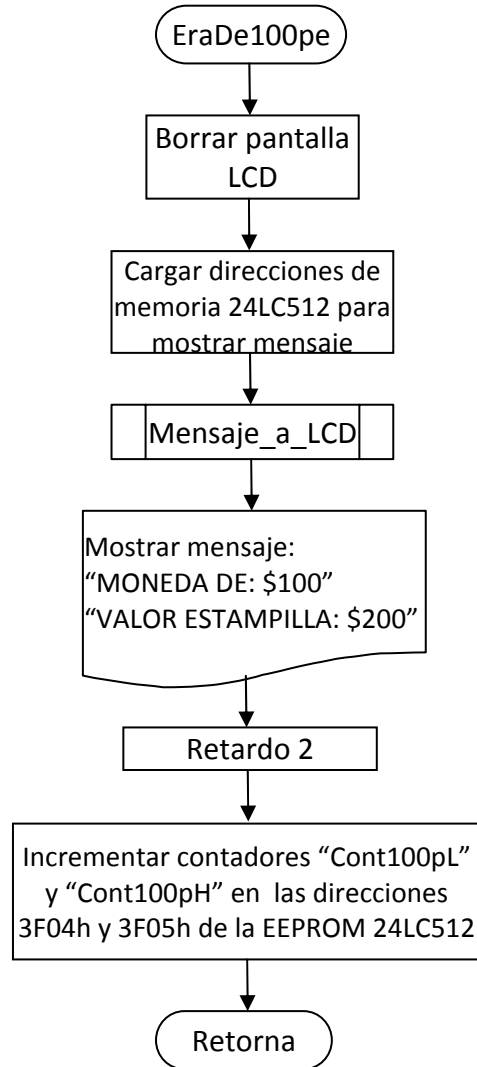


- **SUBROUTINA “EraDe100pe”.**

Para esta rutina se tiene las mismas consideraciones que para la subrutina “EraDe50pe”, ya que se basan en el mismo principio de funcionamiento y en los mismos parámetros de

diseño. Se tiene en cuenta que necesariamente se deben cambiar los registros donde se almacenan los contadores, que para este caso son “Cont100pL” y “Cont100pH”, que son los bytes bajo y alto del contador de monedas de \$100 pesos ingresadas a la máquina.

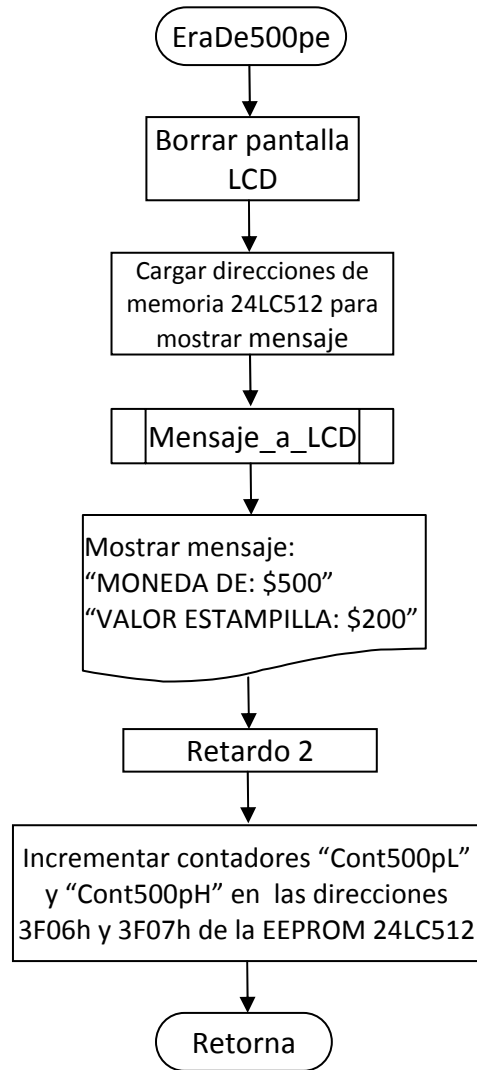
Figura 64. Diagrama de flujo para la subrutina “EraDe100pe”.



SUBROUTINA “EraDe500pe”.

Para esta rutina se tiene en cuenta que los registros donde se almacena los contadores son “Cont500pL” y “Cont500pH”, que son los bytes bajo y alto del contador de monedas de \$500 pesos ingresadas a la máquina.

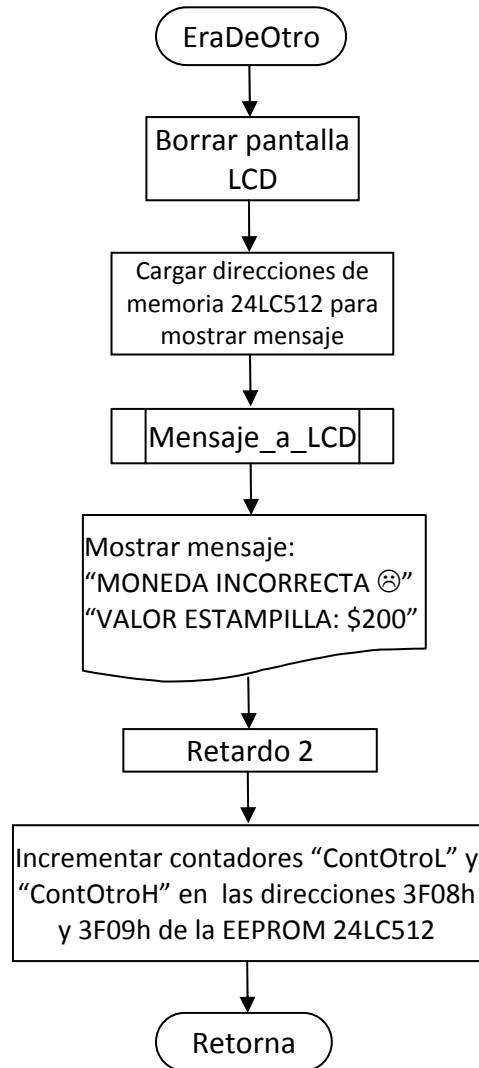
Figura 65. Diagrama de flujo para la subrutina "EraDe500pe".



- **SUBROUTINA "EraDeOtro".**

Se tiene en cuenta que se han de cambiar los registros donde se almacenan los contadores, que para este caso son "ContOtroL" y "ContOtroH", que son los bytes bajo y alto del contador de monedas de otras denominaciones ingresadas a la máquina.

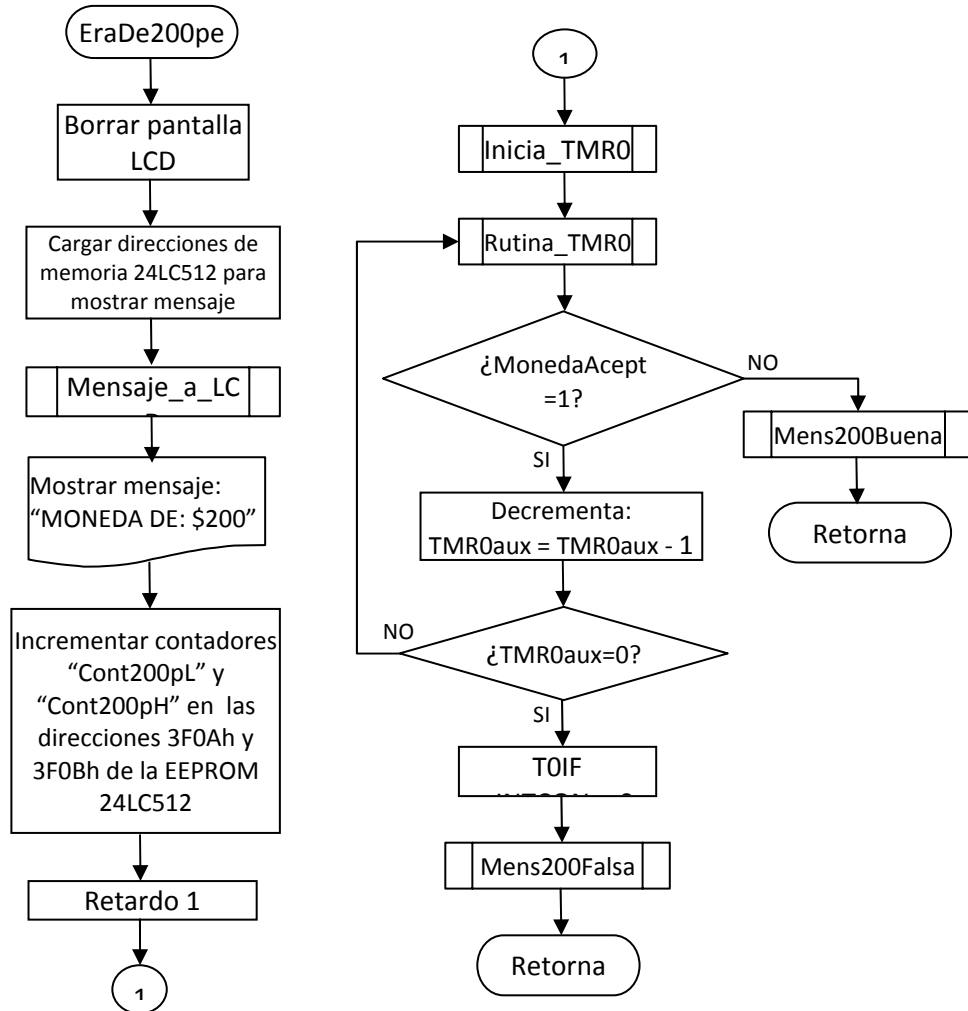
Figura 66. Diagrama de flujo para la subrutina "EraDeOtro".



- **SUBROUTINA "EraDe200pe".**

Esta subrutina y todas las relacionadas con ella se encargan de leer las señales que provienen del sistema sensor de moneda el cual está configurado para aceptar únicamente monedas de \$200 pesos. Dicha moneda es la de mayor tamaño entre las que actualmente se encuentran en circulación en el territorio colombiano, por lo tanto es la última que pasa por los sensores de tamaño, lo cual es detectado y registrado por el programa. Después que la moneda pasa a través de los sensores de tamaño el programa debe esperar unos instantes a que el sistema detector de monedas por el tipo de material expulse la moneda como aceptada o rechazada, según sea su constitución.

Figura 67. Diagrama de flujo para la subrutina "EraDe200pe".

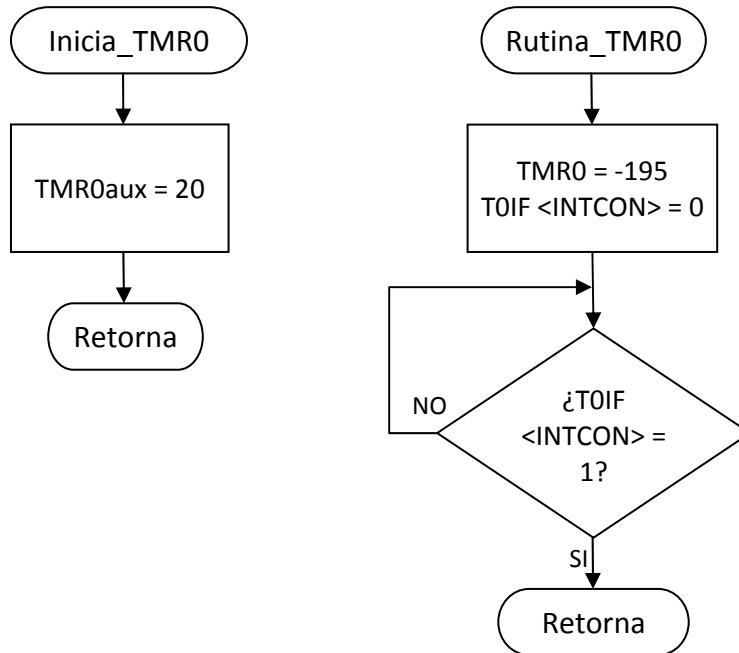


Si la moneda se rechaza, el sistema muestra en pantalla un mensaje alusivo a este hecho, devuelve la moneda al usuario y pasa nuevamente a ejecutar la subrutina principal. En el caso de que la moneda sea aceptada por el sistema, en la pantalla LCD se muestra un mensaje con dos opciones para el usuario: Sellar Tiquete Aéreo o Entregar Estampilla. Según sea la elección, el programa ordena a los motores encargados de los sellos y las estampillas para ejecutar dichas acciones a través del Driver L298. Una vez ejecutada la acción, el programa retorna a ejecutar la rutina principal a espera de que el usuario inserte una moneda nuevamente.

- **SUBROUTINAS “Inicia_TMRO” y “Rutina_TMRO”.**

Subrutina para controlar el tiempo de los sensores de moneda usando el Timer0. Como el PIC trabaja a una frecuencia de 4 MHz el TMR0 evoluciona cada microsegundo. El bucle central se hace en un tiempo de 50 ms. Para ello se utiliza un prescaler de 256 y el TMR0 tiene que contar 195. Efectivamente $195 \times 256 = 49920 \mu s = 50 \text{ ms}$. Para conseguir una temporización de 1 segundo aproximadamente habrá que repetir 20 veces el lazo de 50 ms. Se debe testear el bit TOIF del registro INTCON ya que este se pone en ‘1’ cuando el contador TMR0 se ha desbordado, es decir, en este caso indica que han pasado exactamente 50ms.

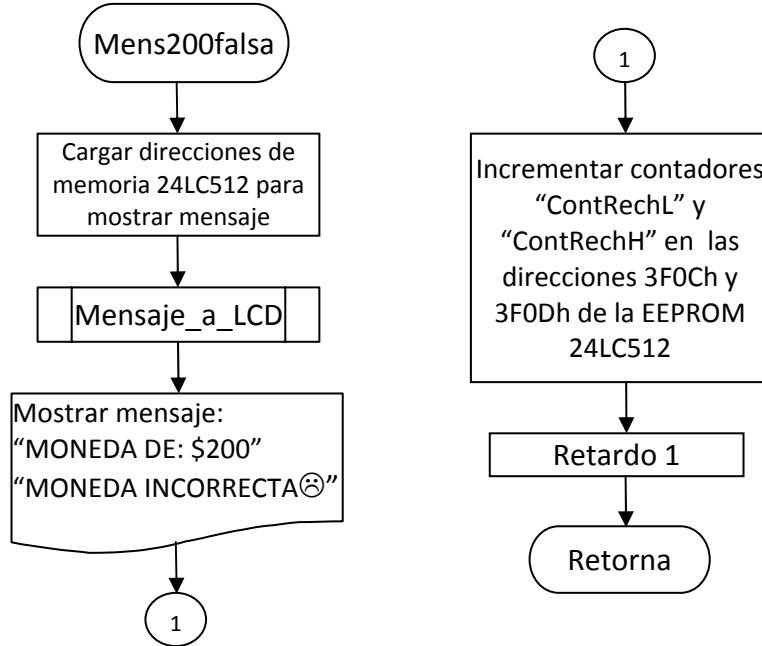
Figura 68. Diagrama de flujo para las subrutinas “Inicia_TMRO” y “Rutina_TMRO”.



- **SUBROUTINA “Mens200falsa”.**

Esta subrutina muestra un mensaje en pantalla cuando el usuario ingresado una moneda de \$200 pesos falsa, registra este acontecimiento en los contadores “ContRechL” y “ContRechH”, que se encuentran en las posiciones de memoria EEPROM Serial Externa 3F0Ch y 3F0Dh respectivamente. Se observa en la figura 69.

Figura 69. Diagrama de flujo para la subrutina "Mens200falsa".

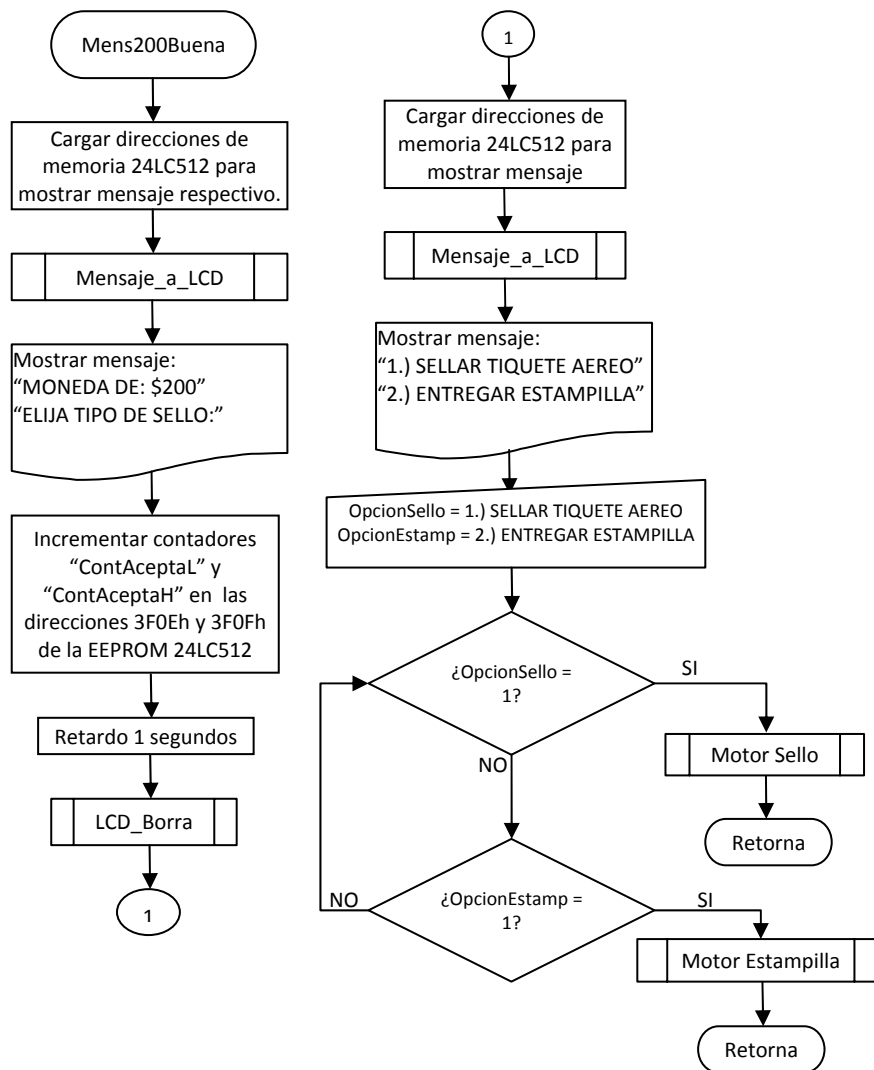


- **SUBROUTINA "Mens200Buena".**

Si el usuario ingresa a la máquina una moneda de \$200 que es aceptada tanto por los sensores de tamaño como por el sistema de detección del material de la moneda esta se considera como aceptada o "buena", lo cual quiere decir que es legal. El sistema se encarga de mostrar un mensaje en pantalla que da aviso al usuario que su moneda es aceptada y que debe elegir entre dos opciones para que el sistema dispense ya sea una estampilla autoadhesiva o selle su tiquete aéreo, para ello la máquina posee dos pulsadores cerca de la pantalla LCD de la máquina.

Para registrar cada uno de estos eventos la máquina posee registros donde se guarda dicha información: la cantidad de monedas de \$200 pesos aceptadas se guarda en los registros "ContAceptaL" y "ContAceptaH" que se graban en las posiciones 3F0Eh y 3F0Fh de la memoria 24LC512, la cantidad de sellos entregados se guarda en los registros "ContSelloL" y "ContSelloH" en las posiciones 3F10h y 3F11h, y la cantidad de estampillas entregadas se guarda en los registros "ContEstampL" y "ContEstampH" en las posiciones 3F12h y 3F13h.

Figura 70. Diagrama de flujo para la subrutina "Mens200Buena".



• **SUBROUTINA "MOTOR SELLO".**

Esta subrutina se encarga del movimiento del motor que mueve el sistema que sella el tiquete aéreo cuando el usuario lo inserta previamente en el espacio de la máquina destinado para esta acción. Estos cuatro pines se conectan con el Driver L298 que se encarga de manejar el motor DC que controla la entrega de Sello.

- EN1_Sello = PORTD, 0
- Rev_Avance = PORTD,1
- Stop_Avance = PORTA, 3

- Stop_Rev = PORTA,5
- IN_Tiquete = PORTC, 2

Reciben habilitación de dos finales de carrera ubicados dentro del recorrido del sello de la máquina. También posee un sensor óptico que sirve para determinar si el usuario ya inserta el tiquete en la ranura, el sistema no procede a sellar mientras no haya papel o tiquete dentro de la máquina.

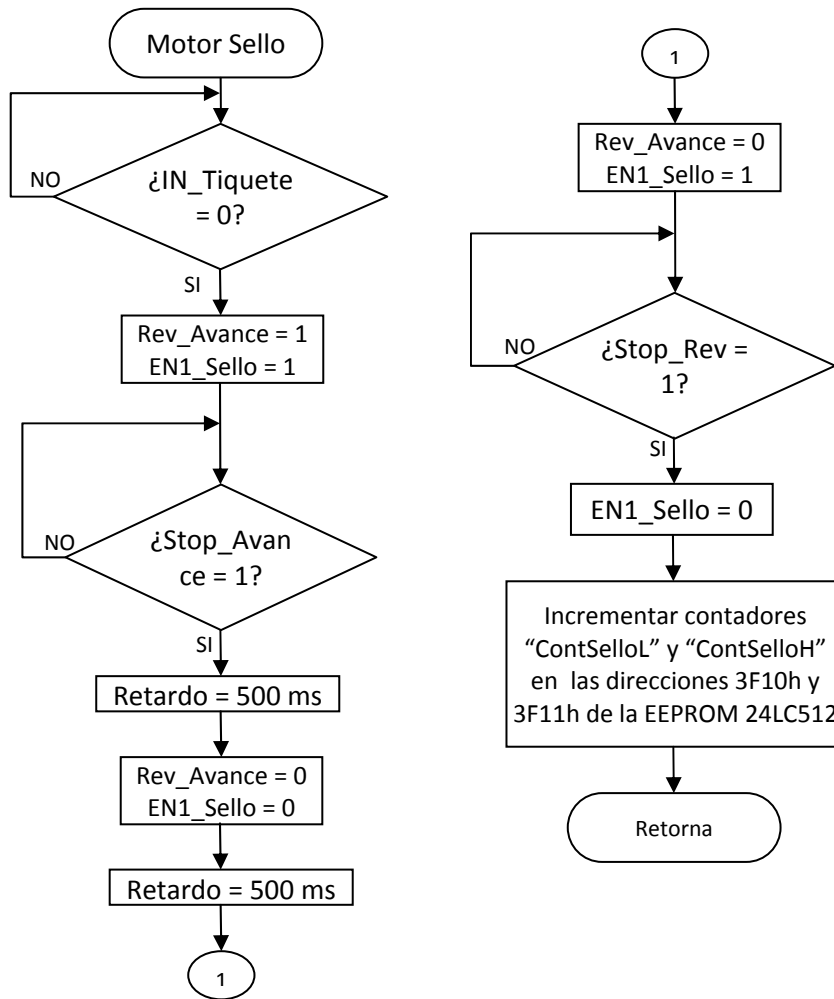
Esta subrutina esta configurada para que inicialmente verifique que el usuario inserta el tiquete dentro de la máquina, para poder habilitar el proceso de sellado. Esto se hace verificando el estado del pin denominado "IN_Tiquete" el cual se encuentra normalmente en '1' de manera que el programa se encuentra testeando permanentemente este pin.

Cuando cambia a '0' quiere decir que el usuario fija el tiquete correctamente, entonces se da paso al proceso de sellado. En esta parte, el motor empieza a avanzar hacia el tiquete habilitando los pines denominados "EN1_Sello" y "Rev_Avance".

El motor baja hasta encontrar el final de carrera que se conecta al pin denominado "Stop_Avance" el cual indica que el sello es puesto exitosamente en el papel. Inmediatamente se deshabilita el funcionamiento del motor mediante el driver y se invierte la polaridad de tal manera que al ser habilitado nuevamente el motor haga que el sello retorne a su posición inicial lo cual se verifica con la activación del final de carrera superior conectado al pin denominado "Stop_Rev". Este proceso se puede resumir mediante el diagrama de bloques de la figura 71.

El motor baja hasta encontrar el final de carrera que se conecta al pin denominado "Stop_Avance" el cual indica que el sello es puesto exitosamente en el papel. Inmediatamente se deshabilita el funcionamiento del motor mediante el driver y se invierte la polaridad de tal manera que al ser habilitado nuevamente el motor haga que el sello retorne a su posición inicial lo cual se verifica con la activación del final de carrera superior conectado al pin denominado "Stop_Rev". Este proceso se puede resumir mediante el diagrama de bloques de la figura 72.

Figura 71. Diagrama de flujo para la subrutina "Motor Sello".

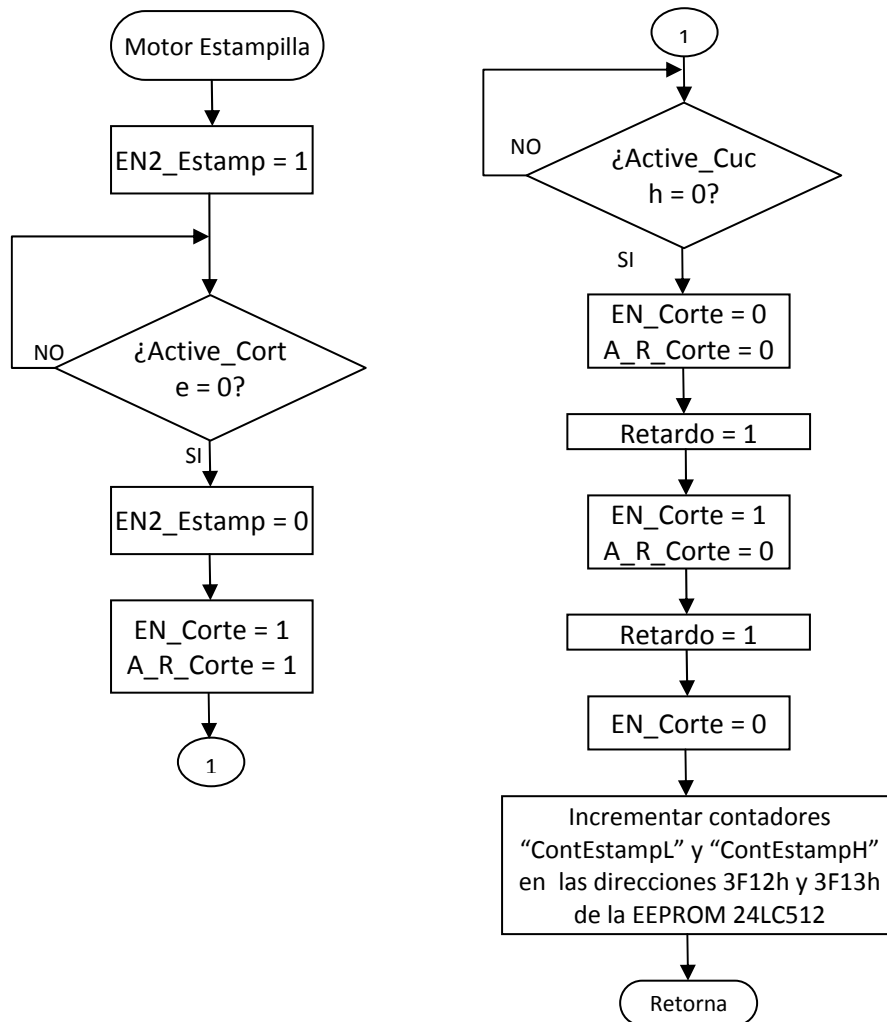


• **SUBROUTINA "MOTOR ESTAMPILLA".**

Esta subrutina habilita o deshabilita cinco pines que se conectan con dos chips Driver L298 que se encargan de manejar el motor DC que controla la entrega de la Estampilla y el motor DC que se encarga del corte de la misma.

- EN2_Estamp = PORTC, 1
- EN_Corte = PORTC, 5
- A_R_Corte = PORTC, 6
- Active_Corte = PORTD, 2
- Active_Cuch = PORTD, 3

Figura 72. Diagrama de flujo para la subrutina "Motor Estampilla".



Para el control de esta subrutina se debe configurar los bits del PuertoD adecuadamente ya que comparten los bits mas significativos se utilizan para el control del módulo LCD.

En primer lugar esta subrutina se activa cuando el usuario requiere obtener una estampilla autoadhesiva, para ello en un inicio se activa el motor que mueve el rollo de estampillas hasta el punto de corte habilitando el pin EN2_Estamp seguidamente a ello la rutina testea el pin Active_Corte el cual indica que la estampilla llega a la cuchilla que hace el corte.

En este punto el motor que mueve la estampilla se detiene y se activa el motor que mueve la cuchilla de corte mediante los pines EN_Corte y A_R_Corte. Para determinar si el sistema realiza el corte se dispone a leer el pin Active_Cuch. Si este se encuentra en bajo,

quiere decir que se corta la estampilla exitosamente y se da la orden al motor de que invierta la polaridad de la corriente mediante el pin A_R_Corte y se activa de nuevo hasta que llegue a su posición inicial donde espera ordenes para comenzar el proceso nuevamente.

- **SUBROUTINA “BORRAR MEMORIA/AJUSTAR HORA/LEER DATOS”.**

Cuando la compuerta de la máquina se encuentra abierta el personal de mantenimiento tiene acceso tres funciones que configuran y modifican el registro de actividad de la máquina estas son:

- BORRAR MEMORIA
- AJUSTAR HORA
- LEER DATOS

- **SUBROUTINA “LlenaCerosEE”.**

La primera función es “Borrar Memoria”, la cual se encarga de poner en cero las posiciones de la memoria EEPROM serial externa 24LC512 que contienen los registros de los contadores de cantidad de monedas, sellos y estampillas que procesa la máquina.

Para poder ejecutar esta función se encuentra un pulsador conectado al pin ‘0’ del PUERTO C, el cual denominamos Pin_Borrar para efectos de programación. Esta subrutina se puede explicar mediante el diagrama de flujo de la figura 74.

En el registro “CantidadCeros” se guarda la cantidad de posiciones de memoria a partir de la posición 3F00h que se quiere dejar en cero. Esta cantidad solo se modifica a nivel del software del microcontrolador.

Figura 73. Diagrama de flujo para la subrutina "BORRAR MEMORIA/AJUSTAR HORA/LEER DATOS".

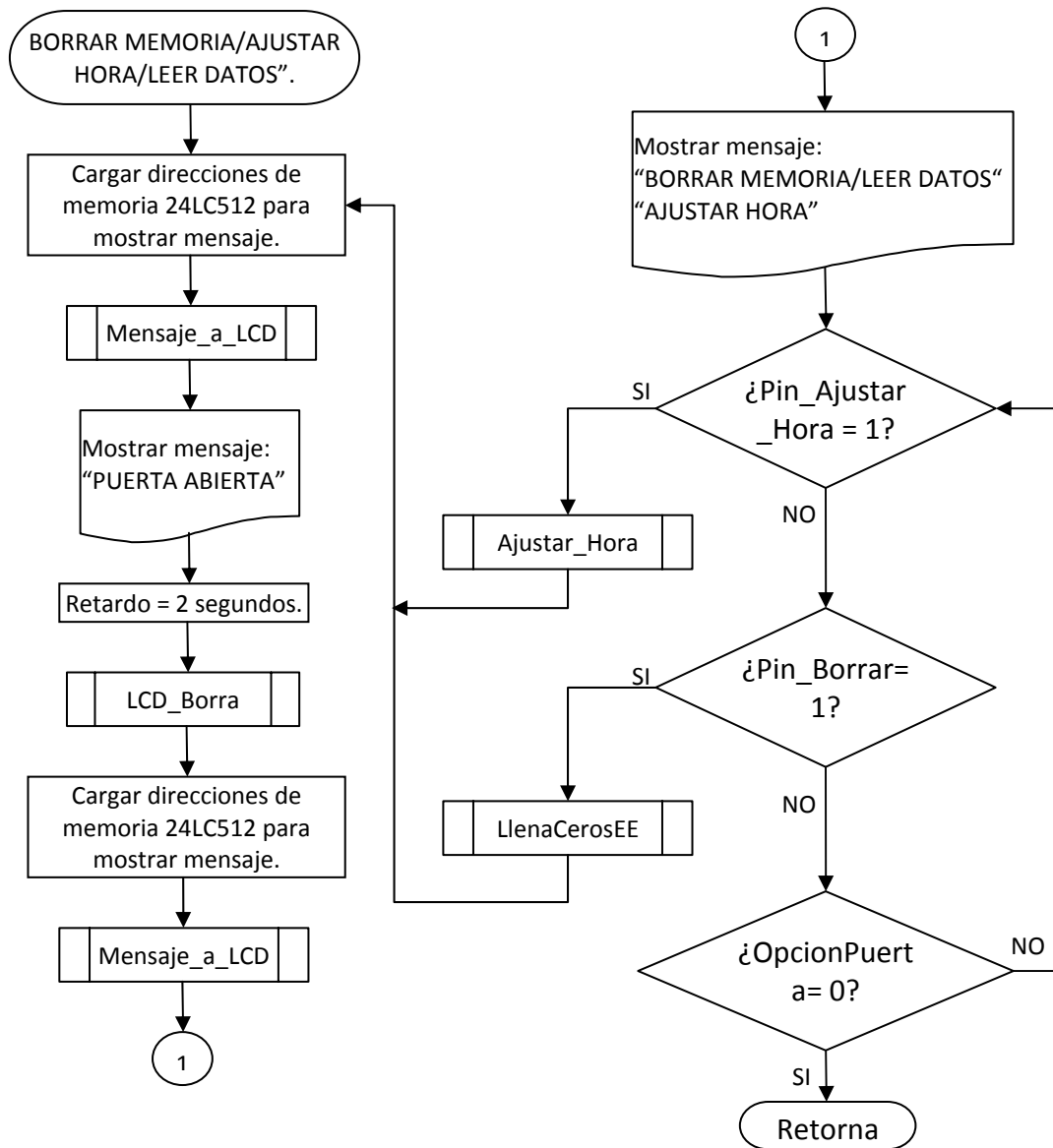
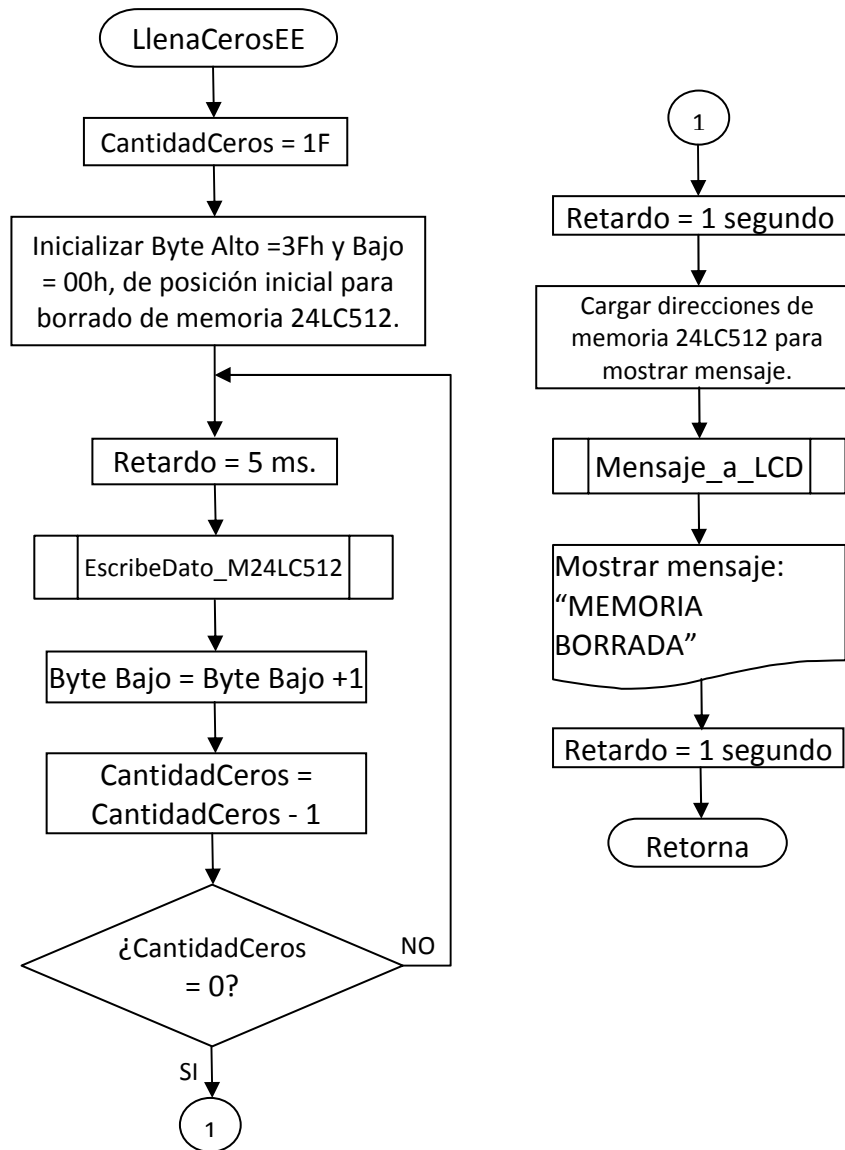


Figura 74. Diagrama de flujo para la subrutina "LlenaCerosEE".



- **SUBROUTINA "Ajustar_Hora".**

Para acceder al ajuste de la hora del sistema, la máquina posee tres pulsadores que se encargan de dicha labor:

- ModoAjuste = PORTB, 0: Pulsador que permite Ajustar el siguiente parámetro, de Fecha y Hora.

- Incrementar = PORTC, 7: Pulsador que permite incrementar el valor del registro de Fecha y Hora que se esta ajustando.
- Pin_Ajustar_Hora = PORTA, 4: Pulsador para iniciar el Modo de Ajuste de Fecha y Hora.

También se hace necesario declarar algunos registros que son necesarios para el funcionamiento correcto de todas las subrutinas del ajuste del reloj. Los principales son:

- Segundo: Guarda los segundos.
- Minuto: Guarda los minutos.
- Hora: Guarda las horas.
- DiaSemana: Guarda el día de la semana, tal como Lunes, Martes, Etc.
- Día: Guarda el día del mes.
- Mes: Guarda el mes.
- Anho: Guarda el año.
- FlagsAjuste: Guarda las banderas para establecer los ajustes de día, mes, año, hora, etc.

También se define la posición de las banderas que indican el ajuste de los diferentes parámetros de hora y fecha.

- F_Anho =FlagsAjuste, 5
- F_Mes =FlagsAjuste, 4
- F_Dia =FlagsAjuste, 3
- F_Hora = FlagsAjuste, 2
- F_Minuto =FlagsAjuste, 1
- F_Salida =FlagsAjuste, 0

De manera muy general se describe el funcionamiento de la subrutina “Ajustar_Hora” a través de varios pasos muy sencillos pero que en el código del ensamblador posee cierta dificultad para ser comprendidos, por lo tanto para tener una idea detallada de esta subrutina se debe leer cuidadosamente el código fuente del programa de control creado.

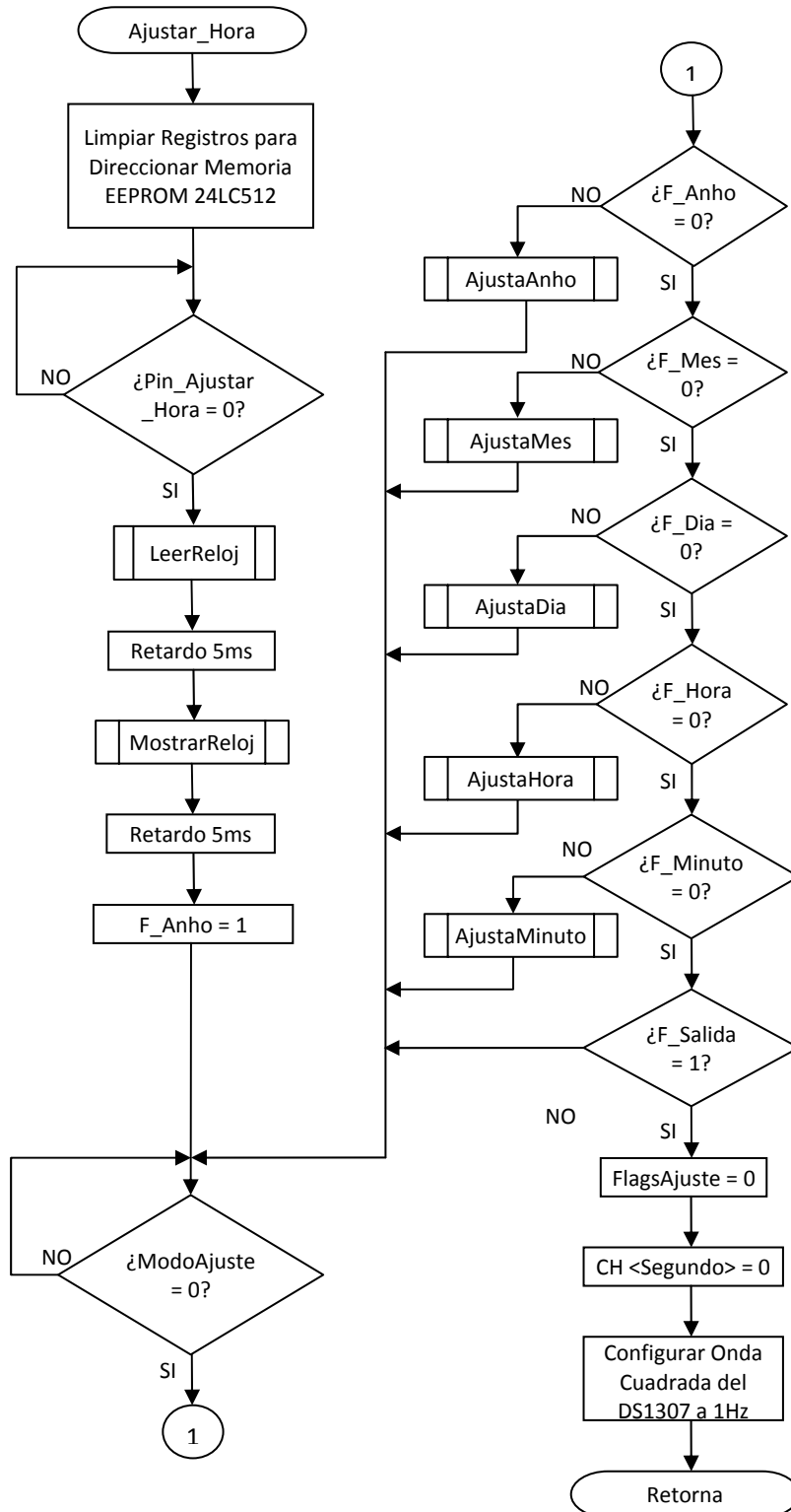
Dichos pasos son:

1. En primer lugar la rutina espera a que se deje de pulsar el botón “Pin_Ajustar_Hora” e inmediatamente lee los registros del reloj-calendario DS1307 desde la posición 00h hasta la 06h y los graba en sus respectivos registros dentro del PIC16F877A así:

- a. Segundo = 00h
 - b. Minuto = 01h
 - c. Hora = 02h
 - d. DiaSemana = 03h
 - e. Dia = 04h
 - f. Mes = 05h
 - g. Anho = 06h (Año)
2. Una vez leídos cada uno de los registros del DS1307 la rutina se encarga de mostrar el contenido del reloj por primera vez en pantalla, y muestra una etiqueta en la parte superior derecha del LCD que dice “/Año/”, lo cual indica que se puede incrementar el registro “Anho” para su respectivo ajuste mediante el pulsador “Incrementar”.
 3. Al mismo tiempo que se puede incrementar el contenido del primer registro, la subrutina también se encuentra testeando el pulsador “ModoAjuste” el cual hace que la rutina cambie la etiqueta superior derecha de la pantalla indicando que se puede modificar el registro “Mes”.
 4. De este mismo modo se puede cambiar el contenido de todos los registros hasta llegar al registro “Minutos”. En este punto, si la subrutina detecta que el usuario pulsó el botón “ModoAjuste” nuevamente, se sale de esta subrutina.
 5. Finalizado el ajuste de todos los registros, el programa ajusta la onda cuadrada que entrega el Reloj-Calendario digital en su pin SQW/OUT y pone un ‘0’ en el bit 7 (CH Clock Halt) del registro 00h del DS1307 de tal manera que el reloj empieza su marcha cada segundo.
 6. Después de estos pasos el programa retorna al mensaje “BORRAR MEMORIA/LEER DATOS/AJUSTE HORA”. En este punto el usuario ingresa de nuevo a la subrutina “Ajuste_Hora” pulsando el botón “Pin_Ajuste_Hora”, en cuyo caso se vuelven a ejecutar todos los pasos anteriores con la salvedad que el reloj tendrá ya en sus registros guardado la configuración anterior que dio el usuario.

De manera resumida los anteriores 6 pasos se observan en el diagrama de la figura 75, y a continuación algunas de las subrutinas contenidas dentro de dicho esquema.

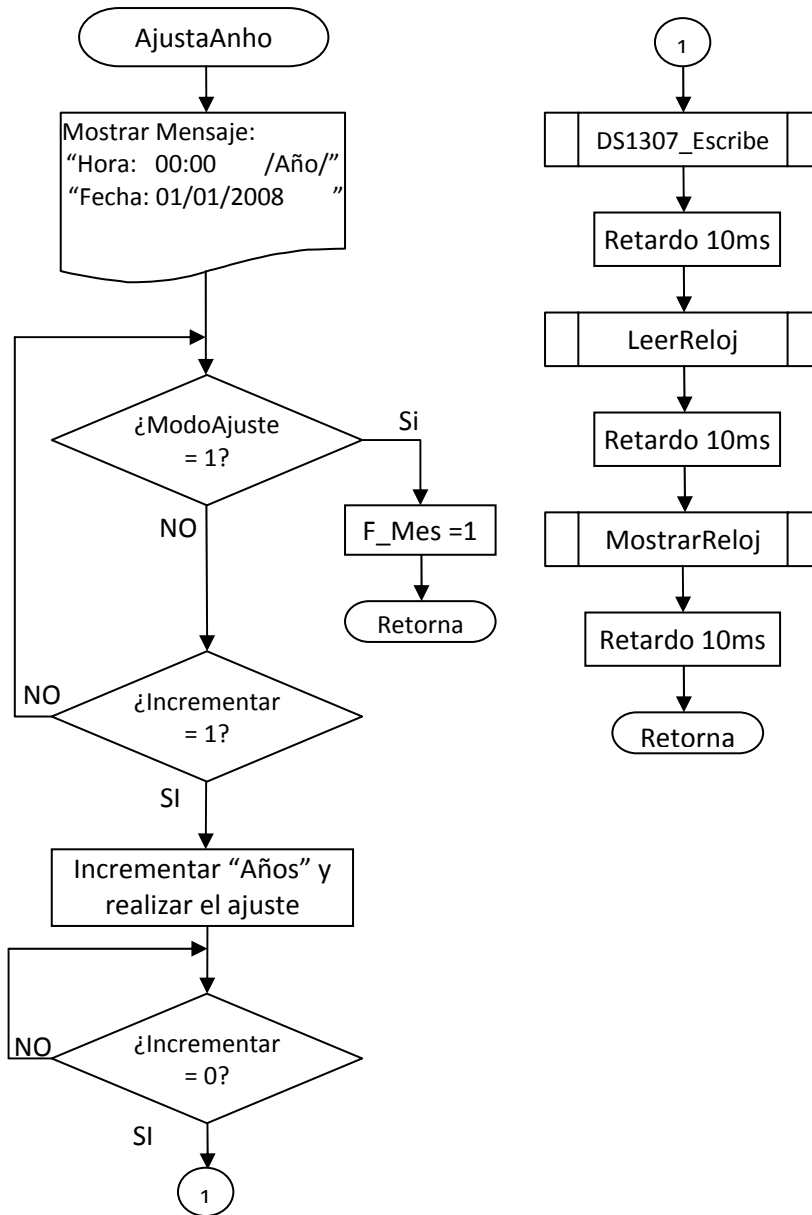
Figura 75. Diagrama de flujo para la subrutina "Ajustar_Hora".



- **SUBROUTINA “AjustaAnho”**

Esta subrutina permite al usuario incrementar el contenido del registro “Anho” y visualizar esta acción en el Display LCD, además llama a otra que permite incrementar únicamente hasta el año 2030, lo cual es un límite definido para que en caso de error el usuario no tenga que incrementar los años hasta el 2100 que es el límite superior y luego volver.

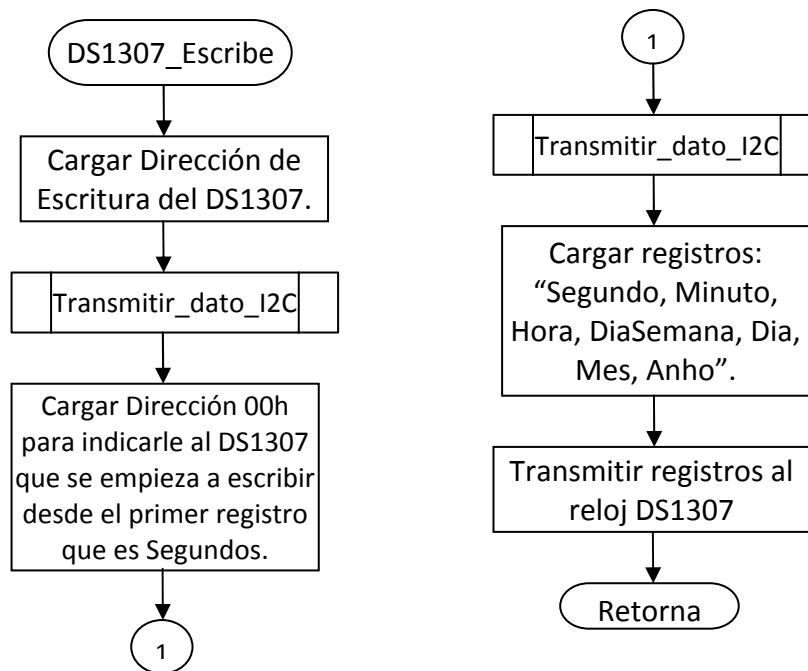
Figura 76. Diagrama de flujo para la subrutina “AjustaAnho”.



Las subrutinas “AjustaMes”, “AjustaDia”, “AjustaHora” y “AjustaMinuto” tienen una estructura igual a la que muestra el diagrama de flujo de la figura 77, con la salvedad que por ejemplo para la rutina “AjustaHora”, el registro que sufre un incremento o modificación es “Hora”, y en la parte superior derecha de la pantalla LCD se observa el mensaje “/Hora/” lo cual indica que actualmente el usuario puede incrementar únicamente el registro de Hora. De esta misma manera se pueden modificar todos los registros que hacen parte del Reloj DS1307.

La rutina “DS1307_Escribe” permite escribir los valores de los registros Anho, Mes, Dia, Hora y Minuto que han sido modificados o no por el usuario en el chip DS1307 a través de los pines SDA y SCL del bus I2C, donde el PIC16F877A actúa como dispositivo maestro. El esquema general de esta subrutina se puede observar en la figura 77.

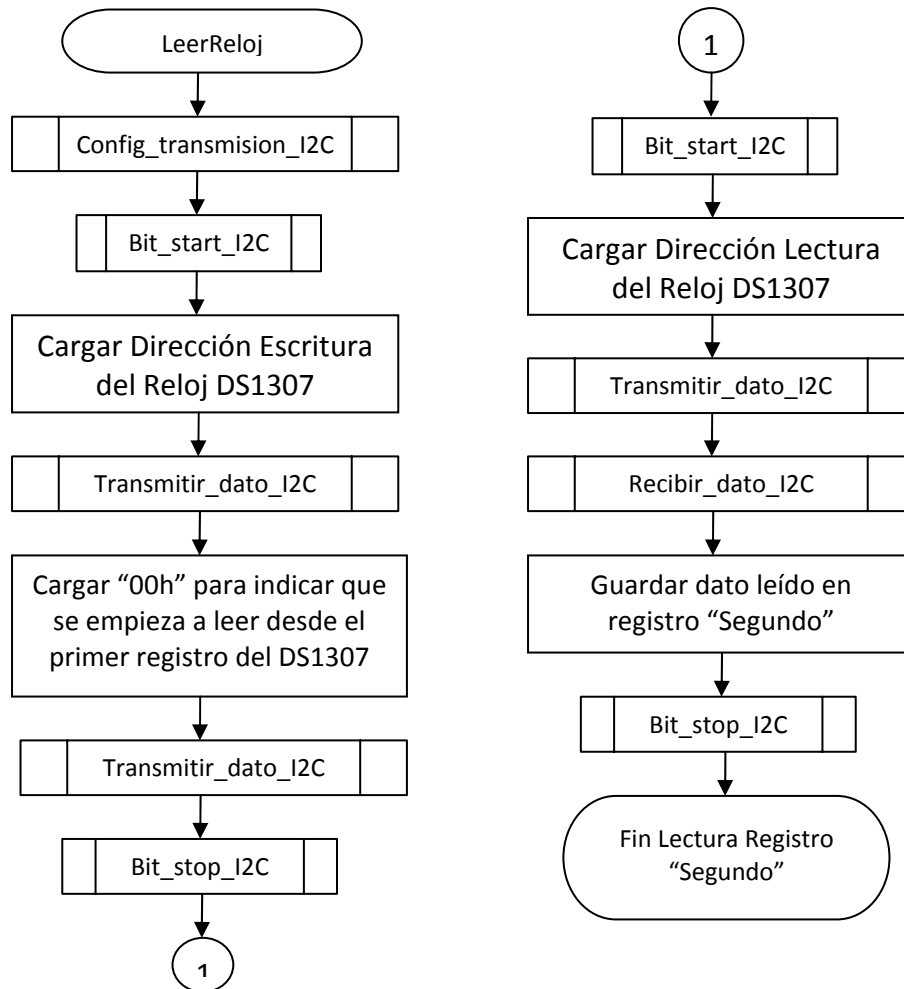
Figura 77. Diagrama de flujo para la subrutina “DS1307_Escribe”.



La subrutina “LeerReloj” que se describe en la Figura 78, sirve para leer todos los registros del reloj DS1307 y escribirlos en otros de manera temporal dentro del PIC 16F877A para que puedan ser modificados según se define en rutinas anteriores, este proceso se hace mediante el módulo I2C del microcontrolador que permite realizar operaciones de lectura de manera muy fácil, similar al método utilizado con la memoria EEPROM 24LC512.

Todas las subrutinas explicadas anteriormente que pertenecen al módulo I2C, también son válidas para esta parte del programa, las que se utilizan con mayor frecuencia son: “Bit_stop_I2C”, “Bit_start_I2C”, “ReStart_I2C”, “Transmitir_dato_I2C” y “Config_transmision_I2C”.

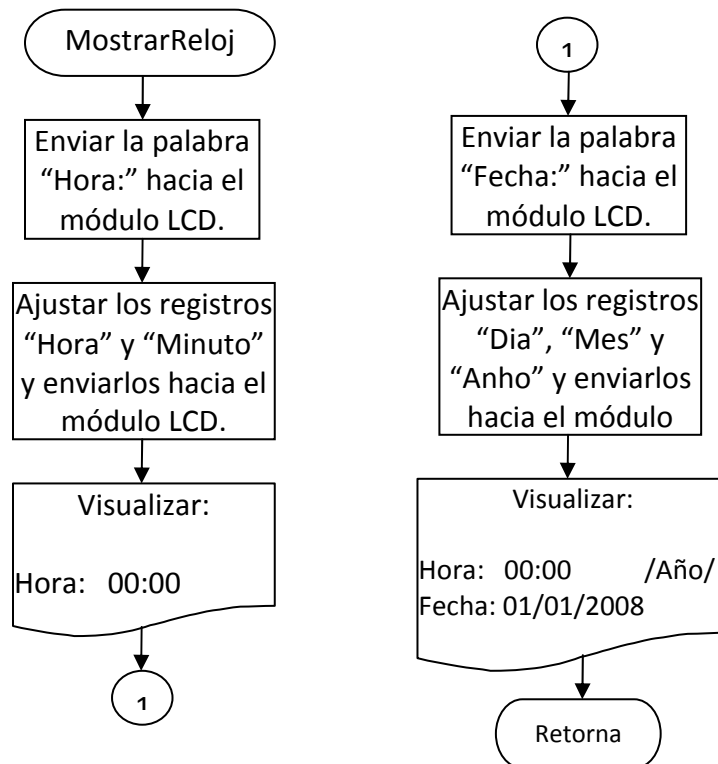
Figura 78. Diagrama de flujo para la subrutina “LeerReloj”.



De manera similar se leen los demás registros del reloj-calendario digital, se usan las mismas subrutinas en igual orden, lo único que cambia es que la siguiente rutina utiliza la dirección 01h para leer el registro que contiene los datos de “Minuto” y guardarlos en el PIC 16F877A de manera temporal. De la misma manera para todos los demás registros; 02h=Hora, 03h=Día de la semana, 04h=Día, 05h=Mes, 06h=Año. Todos estos datos se leen de manera individual pero al momento de llamar a la subrutina “LeerReloj”, se leen todos de una vez usando este método.

La rutina “MostrarReloj” se encarga, en resumen, de llevar los datos leídos del reloj calendario digital hacia la pantalla LCD, pero aplicándoles un formato que facilite su visualización y rápida comprensión, esto se describe en la figura 79.

Figura 79. Diagrama de flujo para la subrutina “MostrarReloj”.



2.11 HARDWARE DEL MODULO DE ALMACENAMIENTO DE INFORMACION DE REGISTRO.

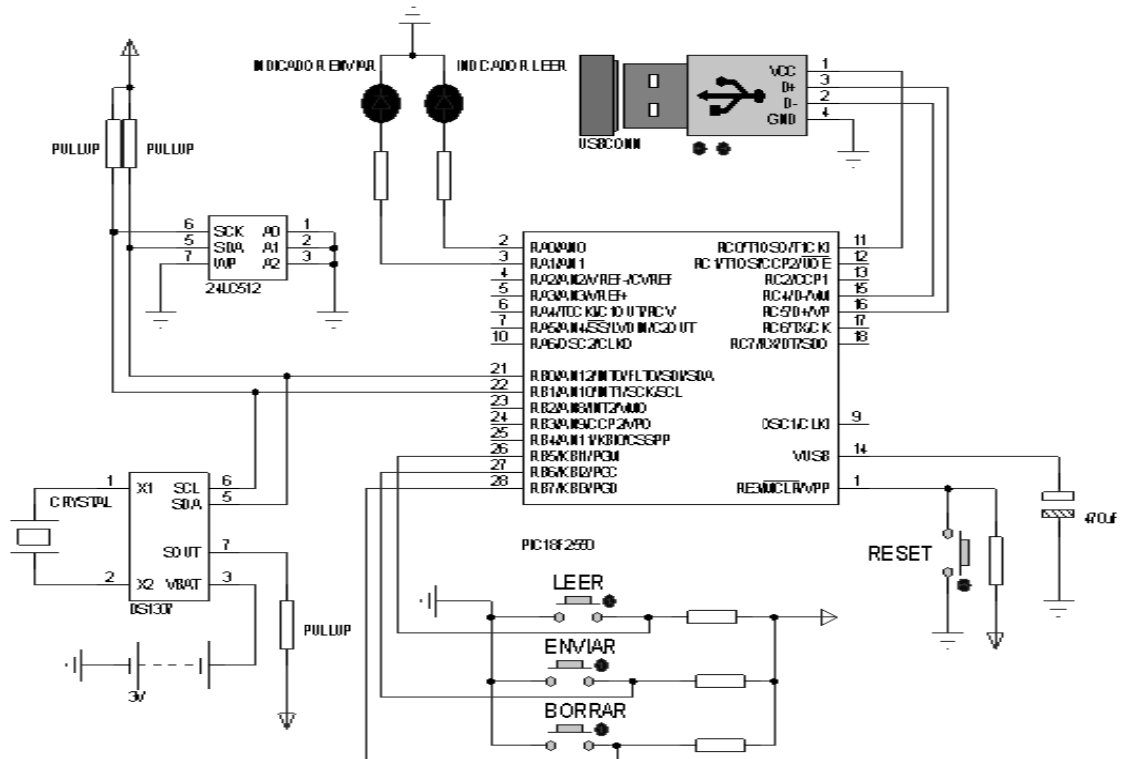
El módulo de almacenamiento de información de registro es el encargado de transferir los datos de la máquina hacia una PC. Se ha diseñado para cumplir cuatro funciones principales que son:

- Transmitir datos desde su memoria EEPROM interna hacia la PC por medio de una interfaz USB y un programa que se encarga de recibir los datos.
- Leer la información de registro de actividades de la máquina dispensadora de estampillas, lo cual implica que debe leer los datos de la memoria EEPROM Serial

Externa 24LC512 y del Reloj DS1307, para posteriormente grabarlos en su memoria EEPROM interna.

- Borrar los datos de la memoria EEPROM Interna llevando todas sus posiciones a cero, para ser usada solo en caso de ser necesario.
- Reset del programa que posee el PIC.

Figura 80. Conexión del PIC18F2550 con la memoria EEPROM 24LC512 y el reloj DS1307.



El diagrama general del circuito que forman el módulo de transferencia de información, la memoria EEPROM Serial 24LC512 y el reloj DS1307, se observa en la figura 80.

Claramente se observa que el PIC 18F2550 es el elemento central del módulo, los pulsadores de funciones se encuentran conectados a los tres bits mas significativos del Puerto B. El módulo también posee dos leds indicadores, los cuales se activan cada vez que le módulo se encuentra en proceso de lectura de datos de la máquina o envío de datos hacia la PC, se encuentran conectados a los dos bits menos significativos del Puerto A.

En el Puerto C del microcontrolador 18F2550, se encuentra la conexión USB del módulo, el pin RC0 se conecta a Vcc del puerto USB de la PC, el pin de tierra se conecta con la tierra general del circuito. Los pines RC4/D- y RC5/D+ se conectan con sus pines equivalentes del Puerto USB.

Las líneas de datos SDA y reloj SCL del bus I2C se conectan a los pines RB0/SDA y RB1/SCL, del PIC18F2550. Físicamente en el módulo de transferencia de información se utiliza el mismo conector del puerto USB para realizar la interfaz módulo-máquina dispensadora y módulo-PC.

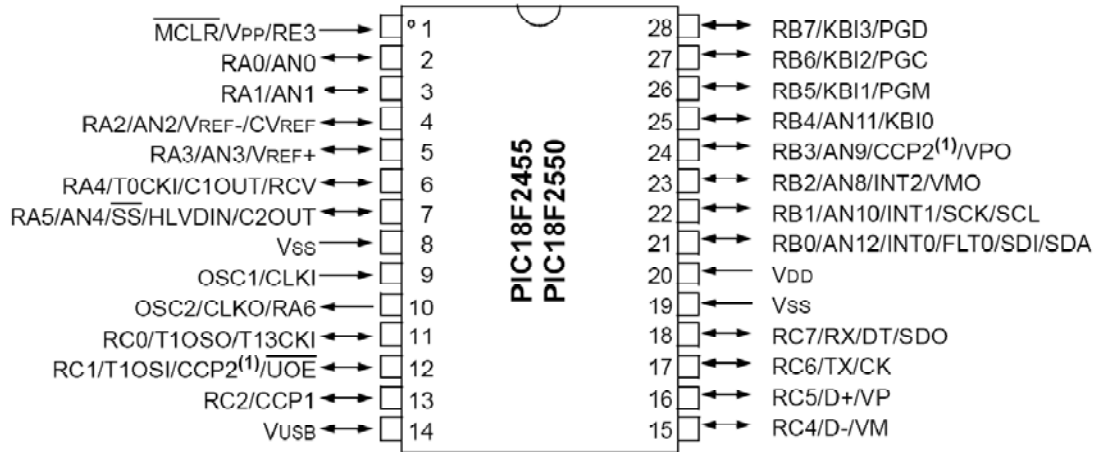
2.11.1 Microcontrolador PIC18F2550. Para escoger el tipo de microcontrolador para desempañar todas las funciones es necesario que cumpla con cuatro requerimientos: poseer memoria EEPROM Interna, módulo I2C, interfaz USB y tamaño reducido. El dispositivo que cumple con estos cuatro requerimientos para el presente proyecto es el PIC 18F2550 del fabricante Microchip. Las principales características de este chip son las siguientes:

- Desarrollado de acuerdo al protocolo USB V2.0.
- Soporta transferencia de datos a baja velocidad (1.5Mbps) y alta velocidad (12Mbps).
- Transceiver (Transmisor/Receptor) USB interno con su propio chip regulador de voltaje.
- Modos de operación de bajo consumo de corriente desde 0.1µA en modo sleep hasta 5.8µA de corriente típica en modo normal.
- Cuatro modos de oscilador a cristal, incluyendo PLL de alta precisión para el módulo USB.
- Rango de voltaje de funcionamiento de 2V hasta 5.5V.
- Frecuencia de operación de 48Mhz.
- Memoria Flash de Programa de 32 Kbytes.
- Memoria SRAM de datos de 2Kbytes.
- Memoria EEPROM de 256 bytes.
- Cuatro puertos de entrada/salida: A, B, C,D.
- 19 fuentes de interrupción.
- Set extendido de 83 instrucciones con arquitectura optimizada para compilador en lenguaje C.

2.11.1.1. Diagrama de pines y encapsulado del PIC18F2550. El PIC 18F2550 posee un encapsulado de 28 pines tipo PDIP, lo cual lo hace de tamaño muy reducido teniendo en cuenta la gran cantidad de funciones que posee internamente. Esto lo hace ideal para

cumplir con las funciones requeridas por el proyecto y da solución a los problemas planteados en la etapa de diseño.

Figura 81. Diagrama de pines del PIC 18F2550.



Fuente: PIC18F2550 Datasheet [Online], 2004 Microchip Technology Inc. Página 2.

2.11.1.2 Configuración del oscilador del PIC18F2550. El PIC18F2550 incorpora diferentes modos de oscilador y sistemas de reloj para el microcontrolador, además, la existencia en su estructura del módulo USB con sus requerimientos de una fuente de reloj estable, hacen necesario proveer una fuente de reloj separada que sea acorde con los modos de baja y alta velocidad del USB. El diagrama de bloques del módulo oscilador del PIC18F2550 se encuentra detallado en la figura 82.

La operación del oscilador del PIC 18F2550 esta controlada por dos registros de configuración y dos registros de control. Los registros CONFIG1L y CONFIG1H, seleccionan el modo del oscilador y las opciones del prescaler/postcaler del USB. Estos registros solo pueden ser configurados antes de programar el dispositivo y permanecen de esta manera hasta que el PIC es reprogramado. El registro OSCCON selecciona el modo de reloj activo, el registro OSCTUNE se usa para reducir la frecuencia INTRC.

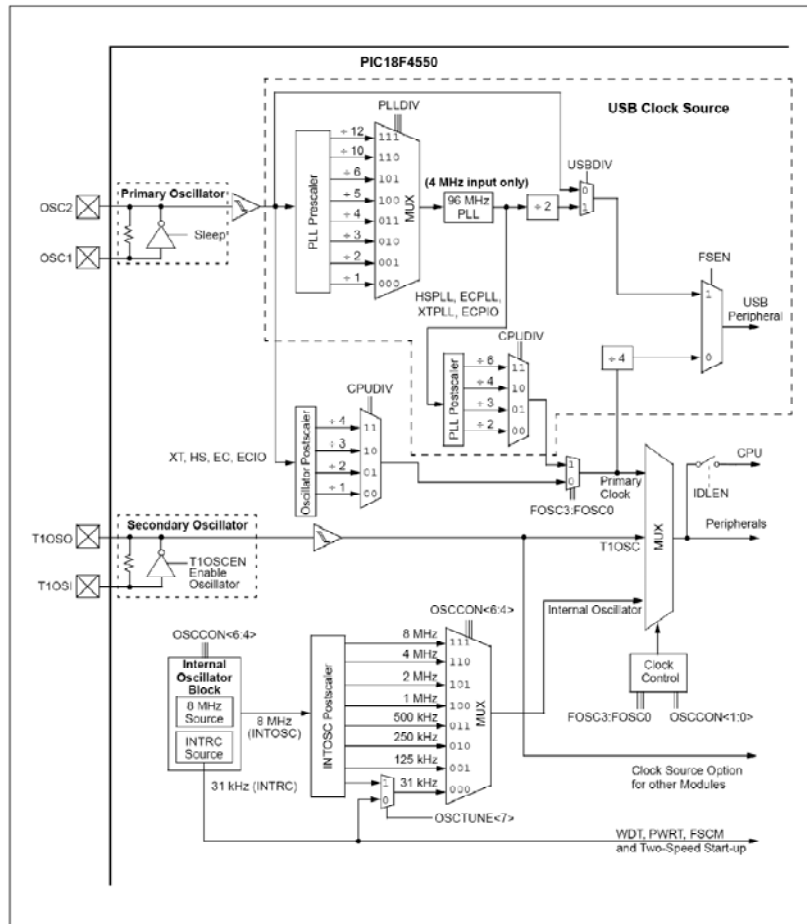
El PIC18F2550 puede ser operado en doce modos diferentes de oscilador, estos se eligen configurando los bits FOSC3:FOSC0.

Estos modos son:

- XT: Cristal/resonador cerámico.
- XTPLL: Cristal/resonador cerámico con PLL activo.

- HS: Cristal de alta velocidad/resonador cerámico.
- HSPLL: Cristal de alta velocidad/resonador cerámico con PLL activo.
- EC: Reloj externo con salida fosc/4.
- ECIO: Reloj externo con I/O en RA6.
- ECPLL: Reloj externo con PLL activo y salida fosc/4 en RA6.
- ECPIO: Reloj externo con PLL activo, I/O en RA6.
- INTHS: Oscilador interno usado como fuente de reloj del microcontrolador, Oscilador HS usado como fuente de reloj del USB.
- INTXT: Oscilador interno usado como fuente de reloj del microcontrolador, Oscilador XT usado como fuente de reloj del USB.
- INTIO: Oscilador interno usado como fuente de reloj del microcontrolador, Oscilador EC usado como fuente de reloj del USB, I/O digital en RA6.
- INTCKO: Oscilador interno usado como fuente de reloj del microcontrolador, Oscilador EC usado como fuente de reloj del USB, salida fosc/4 en RA6.

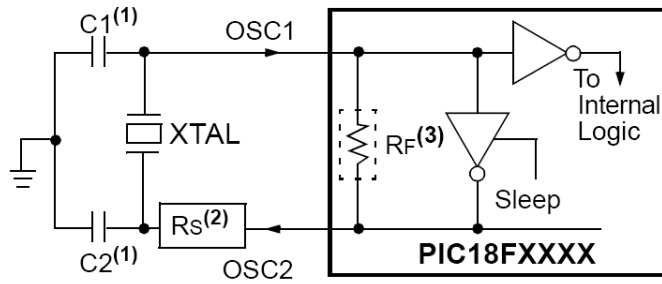
Figura 82. Diagrama de bloques del módulo oscilador del PIC18F2550.



Fuente: PIC18F2550 Datasheet [Online], 2004 Microchip Technology Inc.

Para el módulo de transmisión de información de registro se determina que es suficiente utilizar un oscilador a cristal de 4MHz, el cual se conecta según recomendación del fabricante como se muestra en la figura 83, los condensadores recomendados son $C1 = C2 = 27\mu\text{F}$.

Figura 83. Conexión del oscilador a cristal en el microcontrolador.



Fuente: PIC18F2550 Datasheet [Online], 2004 Microchip Technology Inc.

Para que la configuración de la figura 83 sea posible se deben configurar adecuadamente los registros CONFIG1L y CONFIG1H, para ello debemos conocer como están constituidos y que función cumple cada uno de sus bits.

- **REGISTRO CONFIG1L (Configuration Register 1 low byte).**

Tabla 9. Configuración del registro CONFIG1L.

Bit 7								Bit 0
-	-	USBDIV	CPUDIV1	CPUDIV0	PLLDIV2	PLLDIV1	PLLDIV0	
-	-	1	0	0	0	0	0	

Para que el módulo de almacenamiento y transferencia de información de registro funcione adecuadamente se tiene en cuenta la configuración de los registros CONFIG1L, CONFIG1H y OSCCON. Como se determino en las tablas 7, 8 y 9, se puede determinar que el PIC18F2550 se encuentra funcionando en modo XT de 4MHz, y la velocidad del módulo USB queda determinada en 48MHz para que sea compatible con la versión de USB 2.0.

Los bits PLLDIV (bits 2:0) del registro CONFIG1L se ponen a '0' para no hacer ninguna división y así la frecuencia de 4MHz del oscilador externo entra directamente en el PLL, el valor del bit USBDIV (bit 5) se lo lleva a '1' para hacer la división del PLL el cual tiene una frecuencia de 96MHz, obteniendo de esta forma una frecuencia para el módulo de USB a 48MHz. Los bits CPUDIV (bits 4:3) se ponen en '0' para que la frecuencia del oscilador

primario (4MHz) pase igual a los demás módulos del microcontrolador y solo las transmisiones USB sean de 48MHz.

El bit IESO del registro CONFIG1H permite el intercambio del oscilador interno/externo, pero como para este módulo de transmisión de información solo se utiliza el oscilador externo, se deshabilita esta opción colocándolo en '0', los bits de FOSC3:FOSCO se configuran con '0' lo cual determina que se usa un oscilador tipo XT. El registro OSCTUNE sirve para calibrar y ajustar la frecuencia del oscilador interno pero ya que únicamente se utiliza el oscilador primario como fuente del reloj, el valor de este registro no interesa.

2.11.1.3 Configuración del módulo USB del PIC18F2550. El control del módulo USB del PIC 18F2550 se lleva a cabo mediante la configuración de 22 registros que son:

- Registro de control USB (UCON): contiene los bits necesarios para controlar el comportamiento del módulo USB durante las transferencias. Este se observa en la tabla 10.

Tabla 10. Configuración del registro UCON.

Bit 7				Bit 0			
-	PPBRST	SEO	PKTDIS	USBEN	RESUME	SUSPND	-
-	0	0	1	0	0	0	-

Registro de configuración USB (UCFG): se encarga de habilitar el transceiver, las resistencias Pull-Up internas y la velocidad del dispositivo. Se puede observar en la tabla 11.

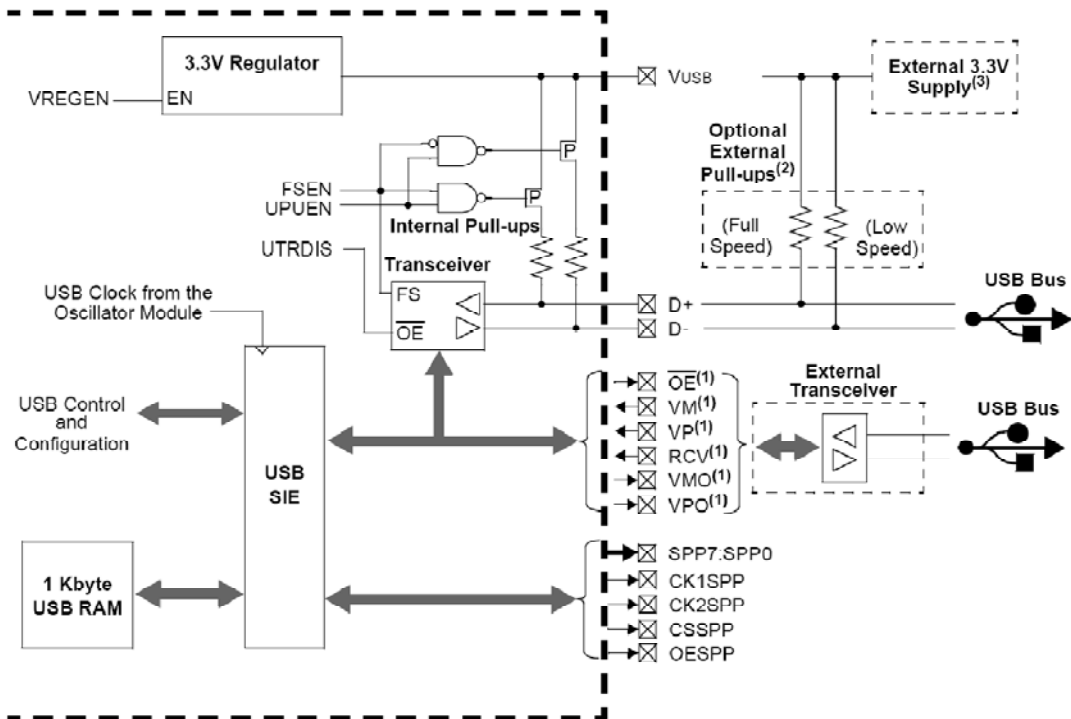
Tabla 11. Configuración del registro UCFG

Bit 7				Bit 0			
UTEYE	UOEMON	-	UPUEN	UTRDIS	FSEN	PPB1	PPB0
0	0	-	1	0	1	0	0

- Registro de estado de transferencia USB (USTAT).
- Registro de dirección de dispositivo USB (UADDR).
- Registros de número de frame (UFRMH:UFRML).
- Registros (del 0 al 15) de habilitación de puntos finales (UEPn).

2.11.1.4 Configuración del periférico USB del PIC 18F2550. El módulo USB posee un regulador de 3.3V interno que se utiliza para suministrar voltaje al transceiver interno. Esto es debido a que USB utiliza 3.3V para realizar la comunicación con otro dispositivo como un PC. Por lo tanto, se necesita que el módulo USB este alimentado a 3.3V mientras que el resto de la lógica del PIC se encuentra funcionando con una tensión de 5V. La fuente que utiliza el transceiver se denomina V_{USB} . Este regulador se controla mediante el bit VREGEN del registro CONFIG2L. Cuando se encuentra en '1', el voltaje de 3.3V se puede medir en el pin V_{USB} . Para que este módulo funcione de manera adecuada se debe activar las resistencias de Pull-Up, las cuales solo pueden ser utilizadas si el transceiver interno se encuentra activado. El diagrama de bloques de dicho módulo se observa en la figura 84.

Figura 84. Diagrama de bloques del periférico USB.



Fuente: PIC18F2550 Datasheet [Online], 2004 Microchip Technology Inc.

2.11.1.5 Configuración del periférico I2C del PIC18F2550. El módulo I2C del PIC18F2550 se maneja de manera muy similar al módulo I2C del PIC16F877A que posee la máquina por lo cual se determina que estos dos son completamente compatibles con la memoria 24LC512 y el reloj DS1307, de donde el módulo de transferencia y almacenamiento de información de registro lee la información para posteriormente descargarla en una PC vía USB.

Los registros de control del módulo I2C del PIC18F2550 son:

- SSPCON1: Registro de control 1 del MSSP, para I2C.
- SSPCON2: Registro de control 2 del MSSP, para I2C.
- SSPSTAT: Registro de estado del MSSP.
- SSPBUF: Registro del Buffer de transmisión/recepción serial.
- SSPSR: Registro de desplazamiento del MSSP, no accesible directamente.
- SSPADD: Registro de dirección del MSSP.

Todos estos registros coinciden tanto en su nombre como en sus bits internos de configuración con lo anteriormente tratado para el PIC16F877A, por lo cual no es necesario volverlos a detallar.

2.11.1.6 Configuración de la EEPROM del PIC18F2550. La EEPROM del PIC18F2550 es un arreglo de memoria no volátil separada de la memoria RAM de datos y la memoria de programa, y se usa para guardar datos durante largos periodos de tiempo. Tiene una capacidad de 256 bytes, y en este caso la EEPROM del PIC se utiliza para almacenar los datos del registro de funcionamiento de la máquina de modo que sean transportados hacia la PC sin riesgo de pérdidas. Cuatro registros son necesarios para el manejo de la memoria EEPROM y son:

- EECON1
- EECON2
- EEDATA
- EEADR

La EEPROM de datos permite leer y escribir bytes de datos. Cuando se conecta el bloque de memoria de datos, el registro EEDATA guarda los 8 bits de datos que van a ser leídos o escritos.

El registro EEADR guarda la dirección de la EEPROM donde se guarda o lee los datos. Los registros EECON1 y EECON2, controlan el acceso a la memoria de datos EEPROM y deben ser configurados para lectura y escritura según el fabricante.

Para leer una posición de la memoria, se escribe en primer lugar la dirección en el registro EEADR, limpiar el bit de control EEPGD ($EECON1<7>$) y poner en alto el bit de control RD ($EECON1<0>$). El dato se encuentra disponible en el siguiente ciclo de instrucción, por lo tanto el registro EEDATA puede leerse en la siguiente instrucción. Esta información permanece en el registro EEDATA hasta que se lea o escriba otro dato. Esto se puede resumir como consta a continuación:

```

MOVLW    DATA_EE_ADDR
MOVWF    EEADR
BCF      EECON1, EEPGD
BCF      EECON1, CFGS      ; Acceso a la EEPROM
BSF      EECON1, RD        ; Lectura de la EEPROM
MOVF    EEDATA, W          ; W = EEDATA

```

Para escribir un dato en una posición de memoria primero se escribe la dirección en el registro EEADR y el dato a grabar esta en el registro EEDATA. Se sigue exactamente el siguiente procedimiento para escribir cada byte, ya que esta es una secuencia determinada por el fabricante:

```

MOVLW    DATA_EE_ADDR
MOVWF    EEADR
MOVLW    DATA_EE_DATA
MOVWF    EEDATA
BCF      EECON1, EPGD
BCF      EECON1, CFGS
BSF      EECON1, WREN
BCF      INTCON, GIE
MOVLW    55h

```

Secuencia Requerida

```

MOVWF    EECON2
MOVLW    0AAh
MOVWF    EECON2
BSF      EECON1, WR
BSF      INTCON, GIE
BCF      EECON1, WREN

```

Adicionalmente se tiene en cuenta que el bit WREN en el registro EECON1, debe ser puesto en alto para habilitar la escritura. Este mecanismo está diseñado para prevenir escrituras accidentales en la EEPROM debido a errores de código. Después de iniciar el proceso de escritura el contenido de los registros EECON1, EEADR y EEDATA no puede ser modificado.

2.12 SOFTWARE DEL MICROCONTROLADOR (MÓDULO DE ALMACENAMIENTO DE INFORMACIÓN DE REGISTRO).

Debido que el módulo USB del PIC18F2550 se posee gran cantidad de registros para configurar su funcionamiento y también a la gran complejidad que posee el intercambio de datos hacia la PC mediante este medio, se decide utilizar un lenguaje de programación

de mayor nivel que el ensamblador. Esto permite simplificar funciones que en lenguaje ensamblador del MPLAB de microchip serian demasiado extensas y llevarían a cometer gran cantidad de errores.

Por ello el software del módulo de almacenamiento y transferencia de información de registro se realiza en un programa Compilador Lenguaje C, que genera los archivos necesarios para el funcionamiento del microcontrolador. Además de ello, este lenguaje posee una amplia gama de librerías de subrutinas que permiten manejar de manera sencilla y directa el módulo USB, I2C y la memoria EEPROM.

El programa principal se encarga de inicializar todos los pines destinados a las funciones especiales del módulo, también se encarga de inicializar el registro que contiene las ordenes para el programador y de inicializar todas las librerías de subrutinas que permiten el manejo de la memoria EEPROM Serial Externa 24LC512, el reloj DS1307, y el módulo USB del PIC.

La rutina principal del módulo, se encarga de estar leyendo permanentemente los tres pulsadores de funciones de tal manera que el programa genere un salto a la subrutina correspondiente a la acción que se desea desempeñar mediante dicho pulsador. El diagrama de flujo de la rutina principal del módulo de almacenamiento de información de registro se puede observar en la figura 85.

- **SUBROUTINA LEER_RELOJ_MEMORIA.**

Esta subrutina se activa cuando se pulsa el Pin RB5 o Leer Datos. La subrutina principalmente inicializa todas las funciones del módulo I2C del PIC de tal manera que lee los datos de importancia de los registros del Reloj DS1307 y de la memoria EEPROM Serial Externa 24LC512. De este modo la información de funcionamiento de la máquina se graba al interior del módulo y posteriormente para que los datos no se pierdan se almacenan en las primeras posiciones de la memoria EEPROM del PIC 18F2550. Durante el tiempo de ejecución de esta subrutina se mantiene activado un led que indica que el microcontrolador se encuentra en un proceso de lectura de datos, al finalizar este indicador se apaga y se retorna a la rutina principal a espera de una nueva orden.

Figura 85. Diagrama de flujo de la rutina principal del PIC 18F2550 presente en el módulo de almacenamiento y transferencia de información de registro.

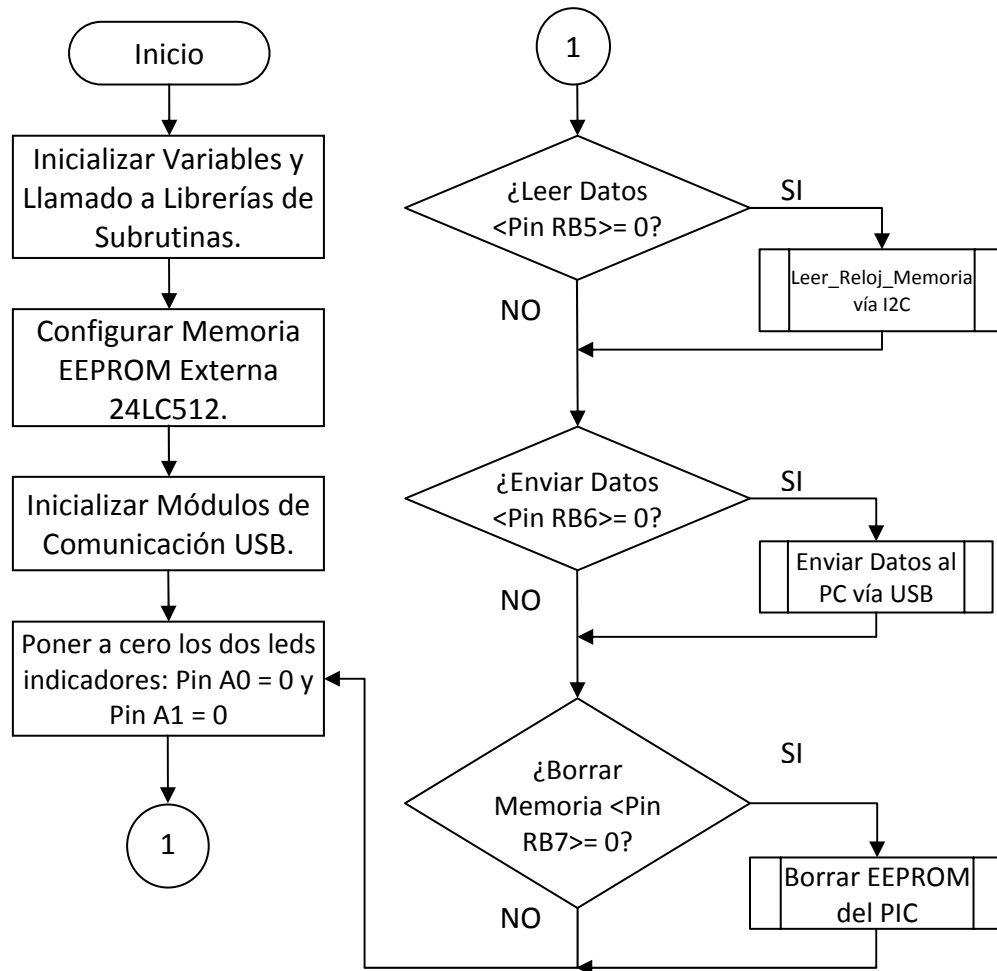
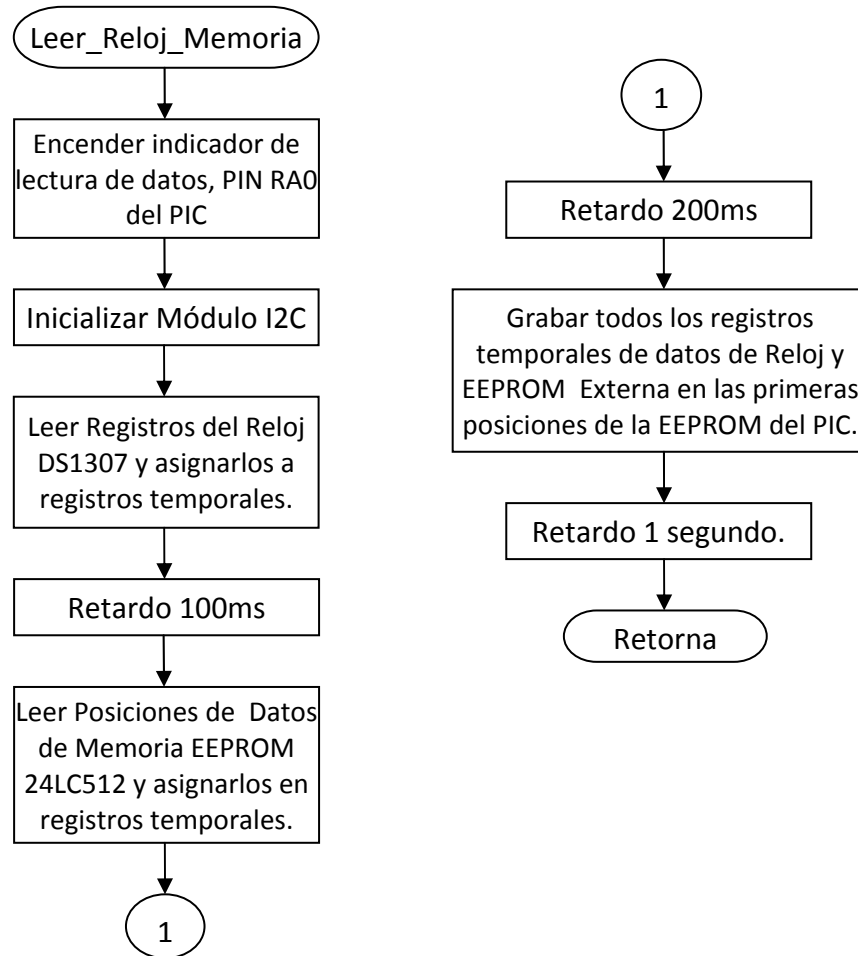


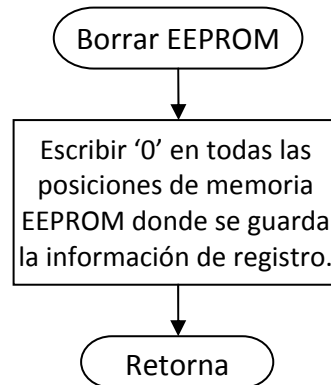
Figura 86. Diagrama de flujo de la subrutina "Leer_Reloj_Memoria".



- **SUBROUTINA BORRAR EEPROM DEL PIC.**

Esta subrutina simplemente graba ceros en todas las posiciones de la memoria EEPROM de datos del PIC18F2550, y el usuario tiene especial cuidado con ella debido a que esto provoca que los datos del registro almacenados en el módulo no se recuperen sino leyéndolos nuevamente de la máquina. El diagrama de bloques de esta subrutina se encuentra el diagrama de la figura 87.

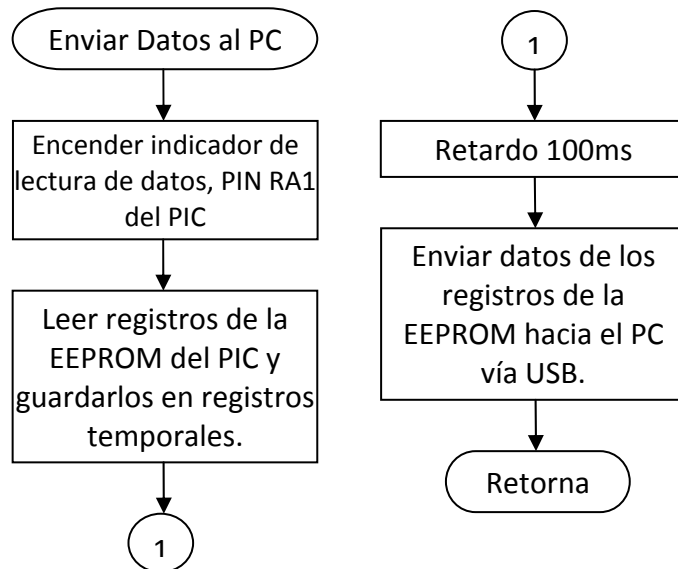
Figura 87. Diagrama de flujo de la subrutina "Borrar EEPROM del PIC".



- **SUBROUTINA ENVIAR DATOS AL PC.**

Esta subrutina permite enviar los datos contenidos dentro de la memoria EEPROM del PIC 18F2550 hacia el PC mediante el módulo USB. Para ello inicializa el módulo de comunicación USB, el cual se habilita cuando el módulo de almacenamiento de información de registro se encuentra conectado el puerto USB del PC.

Figura 88. Diagrama de flujo de la subrutina "Enviar Datos al PC".



Inicialmente esta subrutina se encarga de preparar los datos que se encuentran grabados en la EEPROM del PIC, para lo cual los carga inicialmente en varios registros temporales, desde los cuales se envía los datos hacia la PC mediante la función “usb_cdc_putc” del lenguaje C.

Este proceso se encuentra detallado en el código fuente del programa. El diagrama de bloques de esta subrutina se encuentra en la figura 88.

También posee un led indicador el cual se encuentra conectado al Pin RA1 del PIC18F2550 del módulo y permanece activado mientras el módulo se encuentre enviando datos hacia el PC. Cuando termina de enviar datos este led indicador se apaga.

Para que la transferencia de datos hacia el PC sea posible, se implementa una comunicación bidireccional serial entre el módulo de almacenamiento de información de registro y el PC el cual utiliza un puerto COMM serie estándar pero emulado vía USB 2.0 a velocidad completa. Para esto sea posible se realiza modificaciones en el Firmware del PIC18F2550 para que sea reconocido por el Windows como un dispositivo serie estándar y lo registre como un puerto COMM virtual. Para que esto sea posible se utiliza el driver *mchpcdc.inf* desarrollado por Microchip para Windows XP el cual debe estar instalado correctamente en la PC donde se desee conectar el módulo USB.

Hay varios archivos que ayudan a realizar la conexión entre el módulo y el PC, estos archivos contienen la definición de los registros y funciones USB. Los necesarios para nuestra aplicación son:

- **18F2550.h**: Librería de cabecera estándar para dispositivos PIC18F2550.
- **pic18_usb.h**: Capa hardware de los dispositivos PIC18F2550.
- **USB.C**: Manejador de peticiones USB estándar.
- **USB.H**: Prototipos de funciones, definiciones y variables globales USB.
- **usb_cdc.h**: Librería para adicionar un puerto virtual en el PC a través de USB usando el CDC (Communication Device Class).
- **USB_Cdc_Estampillas.h**: Configuración y descriptores de dispositivos USB.

En la librería *USB_Cdc_Estampillas.h*, se definen las estructuras y parámetros necesarios para la conexión USB. Esta incluye las estructuras y parámetros necesarios para una correcta conexión con el driver del PC. De esta librería se modifica el VID&PID y la tabla *USB_STRING_DESC[]*.

El **VID** (Identificación del Vendedor), es un número de 16 bits, este código identifica al fabricante del hardware a conectar. En este caso se utiliza el número 04D8h que identifica a Microchip. El **PID** (Identificación del Producto), es un número de 16 bits, este código identifica al dispositivo en concreto (hardware a conectar). Para el microcontrolador

18F2550 se utiliza el número 000Bh que identifica a la familia de los PIC18 de este fabricante.

Al conectar el dispositivo de almacenamiento de información de registro al PC, el sistema operativo de este (Windows XP) se conecta con el firmware, el S.O. recibirá el VID&PID y busca entre sus drivers instalados para encontrar el que corresponde a esta identificación. Como este no corresponde a un dispositivo genérico se debe indicar la ubicación del driver adecuado. Este driver debe estar configurado para conectar con un hardware cuyo VID&PID sea el mismo. Como el driver para puertos serie ya existe en Windows, se instala el driver *mchpcdc.inf* para que el S.O se conecte correctamente con el firmware que tiene el PIC.

La tabla `USB_STRING_DESC` de la librería `USB_Cdc_Estampillas.h`, contiene la descripción del dispositivo detectado por el driver de Windows XP y permite mostrar la descripción del dispositivo en la lista del hardware instalado en el sistema. Esta se divide en tres partes:

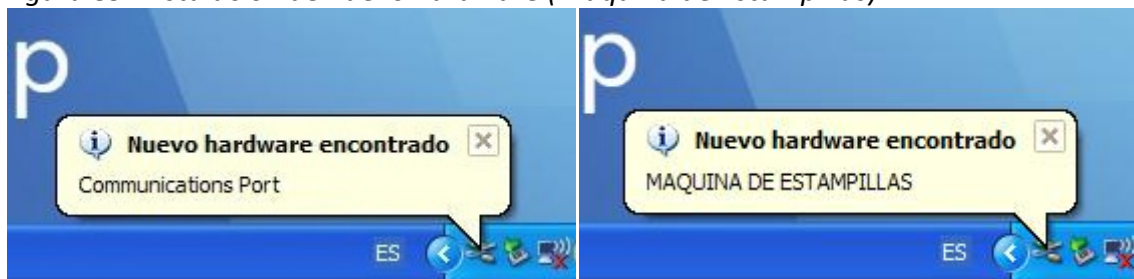
- La `USB_STRING_DESC` que contiene las descripciones requeridas y una tabla llamada `USB_STRING_DESC_OFFSET` que contiene los desplazamientos con respecto al inicio de `USB_STRING_DESC` en donde se encuentran las correspondientes cadenas.
- `USB_STRING_DESC_OFFSET` tiene tres números que indican donde comienza el correspondiente dato en la tabla `USB_STRING_DESC`. Así un contenido de {0, 4, 12} indica que el primer string comienza en el byte 0, el segundo en el byte 4 y el tercero se encuentra a partir del byte número 12. Si se cambia la longitud de cualquiera de los strings, se debe reordenar esta tabla correspondientemente con solo contar los caracteres y apuntar en esta tabla el número de byte donde comienza cada uno de ellos.
- `USB_STRING_DESC` contiene los tres strings que se va a transmitir con el descriptor USB. Cada uno de ellos tiene la misma estructura que consta de un primer byte que indica la longitud total de la correspondiente cadena, un segundo byte que indica el tipo de dato que viene a continuación y por último tantos bytes como sean necesarios en el contenido del string.

Para que al momento de conectar el dispositivo por primera vez al computador se pueda observar un mensaje que lo identifique se debe configurar la siguiente cadena dentro de la librería `USB_Cdc_Estampillas.h`:

22, USB_DESC_STRING_TYPE, 'M', 0, 'A', 0, 'Q', 0, 'U', 0, 'I', 0, 'N', 0, 'A', 0, ' ', 0, 'D', 0, 'E', 0, ' ', 0, 'E', 0, 'S', 0, 'T', 0, 'A', 0, 'M', 0, 'P', 0, 'I', 0, 'L', 0, 'L', 0, 'A', 0, 'S', 0, en este string se define el nombre del dispositivo como: "MAQUINA DE ESTAMPILLAS".

Al conectar el dispositivo móvil al PC por primera vez, aparece el asistente de Windows para la instalación de nuevo hardware donde se observa algo similar a lo que esta en la figura 89.

Figura 89. Instalación de nuevo hardware (Máquina de Estampillas)



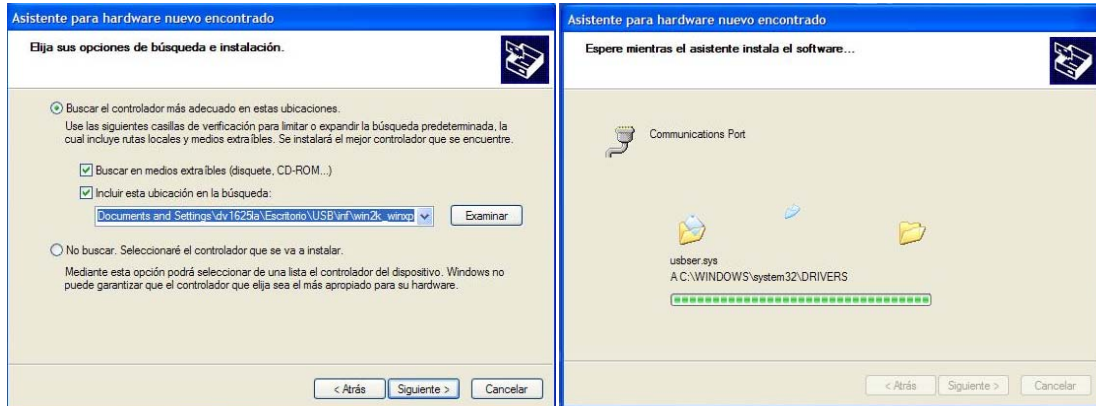
Posteriormente el asistente pregunta si se quiere conectar a Windows Update. Se marca la opción *No* y se da clic sobre <Siguiente>. En la siguiente pantalla se marca la opción de "Instalar desde una lista o ubicación específica". Se da clic sobre <Siguiente>: a la próxima pantalla, donde se marca la opción de "Buscar el controlador más adecuado en estas ubicaciones". Se presiona sobre "Incluir esta ubicación en la búsqueda" y a continuación sobre *Examinar*, se selecciona la ubicación de la carpeta que contiene el driver *mchpcdc.inf*, y se le da clic en <Aceptar> y en <Siguiente>.

Figura 90. Instalación de nuevo hardware.



En ese momento, comienza a copiarse los archivos necesarios en el computador.

Figura 91. Instalación de nuevo hardware.



Al terminar de copiar, aparece una pantalla indicando que la instalación del driver está finalizada.

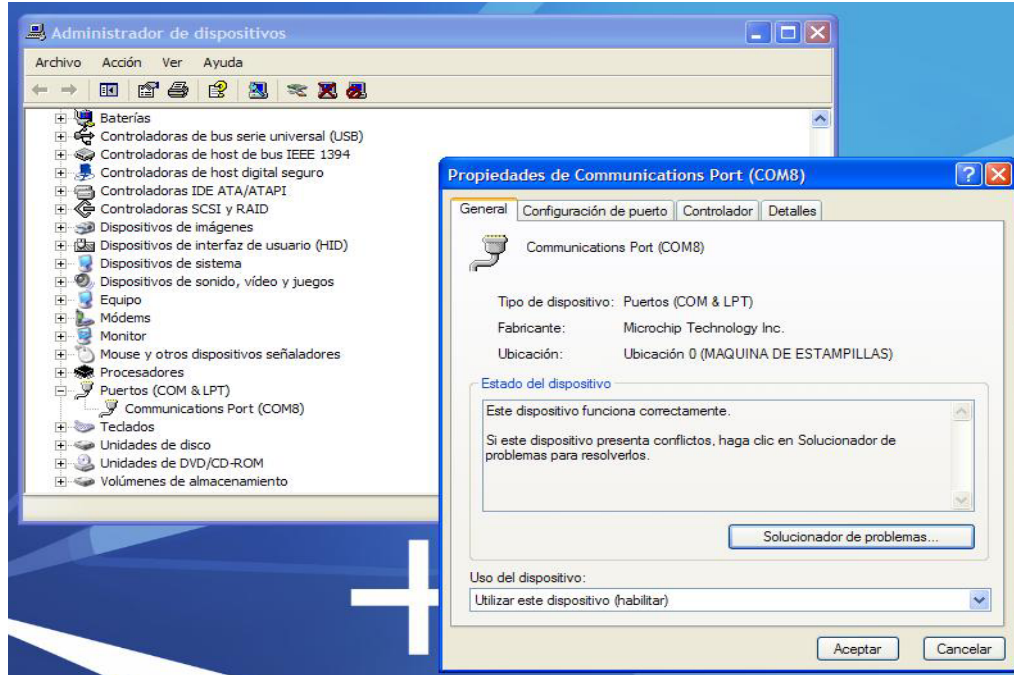
Figura 92. Instalación de nuevo hardware.



Si la instalación del dispositivo que se realiza no tuvo ningún inconveniente, en el Administrador de dispositivos debe aparecer un puerto COM virtual nuevo.

En las siguientes ocasiones que se conecte el dispositivo al computador, este es reconocido de forma automática, ya no es necesario indicar ningún tipo de información.

Figura 93. Instalación de nuevo hardware.



2.13 SOFTWARE DE RECEPCION DE DATOS DEL MODULO (ALMACENAMIENTO DE INFORMACION EN PC)

Gracias al módulo de almacenamiento de información de registro es posible llevar un control periódico de los procesos realizados por la máquina, la visualización y almacenamiento de esta información para su posterior manipulación, puede ser llevada a cabo a través de un computador mediante un software de recepción de datos creado en Visual Basic 6.0.

Como se ve en la figura 94, la interfaz del software con el usuario es muy simple y fácil de comprender. Al iniciar el software y luego de conectar el módulo de transmisión al computador el usuario deberá seleccionar el puerto al cual se a conectado el dispositivo, luego de esto procederá a abrir el puerto, en esta instancia el programa entra en el modo de lectura de datos esperando a que el módulo transmita toda la información contenida en el, al recibir la información el usuario procederá a cerrar el puerto y finalmente mediante un clic en el botón guardar, el software le permitirá al usuario seleccionar un archivo en Microsoft Word para guardar en el la información.

Figura 94. Interfaz software transmisión datos PC



El programa establece la comunicación con el módulo a través de la emulación de un puerto serial COMM virtual el cual facilita el establecimiento de protocolos para enlace. La programación del mismo se basa en comandos simples y variables enteras sencillas, para la parte de almacenamiento de datos, en cuanto a la parte de comunicación se emplea los comandos propios del módulo MSCOMM, estos pueden encontrarse dentro de la ayuda de Visual Basic 6.0.

3. CONCLUSIONES

El desarrollo del proyecto concluye con la materialización del diseño de la máquina dispensadora de estampillas con un módulo de almacenamiento de información de registro, tal y como se había planteado en los objetivos específicos planteados en el anteproyecto.

Se consigue implementar exitosamente el sistema sensor de moneda de tal manera que logra distinguir dos parámetros de importancia como son el material y el tamaño, de manera que se logra una discriminación muy precisa de la moneda de \$200 pesos colombianos. Además de ello se logra obtener señales apropiadas que son leídas por el microcontrolador y que reflejan con gran exactitud las características de la moneda permitiendo al software interno interactuar de manera correcta con el usuario.

Tanto en procesos de control de pequeña magnitud como en la solución de dificultades en grandes industrias la combinación entre mecánica y electrónica permite generar infinitas alternativas para resolver problemas de diseño electrónico que necesiten interacción de elementos mecánicos para su funcionamiento. Esto fue aplicado en nuestro caso de tal manera que se logra la construcción de un sistema electromecánico de distribución de sellos y estampillas que enlaza de manera muy eficiente los distintos actuadores del sistema con un controlador digital como es el PIC16F877A.

También se concluye con la construcción de un módulo o sistema de almacenamiento y transferencia de información de registro el cual tiene una interfaz USB para realizar la conexión con el PC de manera que es adaptable a cualquier computadora moderna ya sea tipo Desktop o Laptop. Este módulo diseñado y construido también lee los datos del registro de funcionamiento de la máquina utilizando el Bus I2C y no requiere de la intervención del microcontrolador que maneja la máquina lo cual reduce el software de control necesario y le da independencia al módulo de almacenamiento del registro.

El módulo de transferencia de información tiene la ventaja de dar un alto grado de seguridad a los datos del registro de funcionamiento de la máquina debido a que es el único elemento que puede tener acceso a dicha información. Esto es importante ya que la máquina almacena dinero y por ello es de gran utilidad que el registro no pueda ser modificado ni leído mediante una memoria o dispositivo cualquiera.

El software creado para el control de la máquina dispensadora de estampillas se desarrolla en lenguaje ensamblador lo que proporciona un mayor grado de control sobre todos los dispositivos, periféricos y funciones que posee el microcontrolador PIC16F877A, permitiéndonos profundizar en la investigación de tal manera que se comprende como se enlaza el microcontrolador con los demás módulos del sistema desde su lógica mas básica. Esto permitió minimizar los errores al máximo y proporciona una visión más amplia de la labor de un diseñador dentro de un sistema con determinado grado de complejidad.

El lenguaje de programación C de gran ayuda para el diseño del software del módulo de almacenamiento y transferencia de información de registro ya que posee los recursos necesarios para desarrollar todas las subrutinas necesarias para el funcionamiento del módulo USB ya que si se programara dichas funciones en lenguaje ensamblador el código resultante seria demasiado extenso y proclive a varios errores que demorarían el proceso de diseño.

El proceso de diseño y construcción de la máquina dispensadora de estampillas así como del módulo de almacenamiento y transferencia de información de registro, lleva a investigar una gran variedad de procesos y conceptos que refuerzan, complementan y profundizan varios de los temas tratados durante la formación como ingenieros que electrónicos.

RECOMENDACIONES

Mejorar el sistema sensor de monedas, de manera que se amplíe su capacidad de reconocimiento a más de una denominación, para así, aumentar las opciones del usuario, tales como; seleccionar más de una estampilla y entregar regresos.

Actualizar el sistema de visualización de los datos mediante el uso de un Display LCD gráfico que permita presentar las opciones de una forma más agradable y ordenada al usuario.

Diseñar un sistema de auto diagnóstico, que proporcione el estado actual de la máquina y facilite la detección de errores en su funcionamiento por parte del operario.

Implementar un sistema de distribución de estampillas que maneje más de un modelo de estampillas y diversos valores de las mismas.

Reemplazar los elementos de las tarjetas electrónicas por dispositivos de superficie que generen una reducción en el espacio que ocupa cada una de ellas.

Integrar un sistema de recolección de datos que maneje una plataforma manipulable a través de mensajes de texto, con el fin de permitir la lectura de información desde estaciones remotas.

Crear un grupo de investigación y diseño dentro del área de desarrollo de hardware.

BIBLIOGRAFIA

ANALOG PRODUCTS DATABOOK, National Semiconductor, 2004 Edition.

DISTRIBUIDORES ATM'S DE PRODUCTOS POSTALES Y PARAPOSTALES. Prototipos de equipos distribuidores de sellos postales. En: www.ateeme.net [2007]

DS1307 Serial Real -Time Clock datasheet [Online]. Dallas Semiconductor. 2000. Disponible en Internet: < http://www.maxim-ic.com/quick_view2.cfm/qv_pk/2688>.

DUQUE, Edison. Curso Básico De Microcontroladores PIC. Colombia: CEKIT S.A. 1997. 115p.

EEPROM I2C 24LC256 datasheet [Online]. Microchip Technology Inc. 2004. Disponible en Internet:
<http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2057&ty=&dt=Data+Sheets§ion=Data+Sheets&ssUserText=24LC256>.

ESTAMPILLA UDENAR. Ley 542 de 1999. En: www.udenar.edu.co [Referencias sobre la legislación acerca de la expedición de la estampilla y formas de cobro]. [2007].

I2C-Bus Specification And User Manual [Online]. Philips Semiconductors. Versión 3.0, junio 19, 2007. Disponible en Internet: < <http://www.semiconductors.philips.com/I2C>>; 15p.

INSTITUTO COLOMBIANO DE NORMAS TECNICAS Y CERTIFICACION (ICONTEC). Documentación. Presentación de Tesis, trabajos de grado y otros trabajos de investigación. Quinta actualización. Santa Fé de Bogotá: ICONTEC. 112 p.: il. (NTC 1486).

LASSO, Julio Cesar y CABRERA, Rommel Oswaldo. Automatización Sistema de Control de Tiempo del Transporte de Buses Urbanos. San Juan de Pasto, Universidad de Nariño, 2008.

LINEAR APPLICATIONS HANDBOOK, National Semiconductor, 2003 Edition.

MALONEY, Timothy. Electrónica Industrial Moderna 3ª Edición, Prentice Hall, 1997.

PIC18F2550 datasheet [Online]. Microchip Technology Inc. 2004. Disponible en Internet:<
http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1335&DocName=en010280>.

PIC16F628A datasheet [Online]. Microchip Technology Inc. 2005. Disponible en Internet:
< [http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE &nodeId=1335&DocName=en010210](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1335&DocName=en010210)>.

PIC16F87X DATA SHEET, Microchip. [2007]

“ROWLAND HILL”. Microsoft® Encarta® 2006 [CD]. Microsoft Corporation, 2006. [Historia de la estampa]. [2007].

TLP434A datasheet [Online]. Laipac Technology Inc. 2001. Disponible en Internet: <
http://www.laipac.com/easy_434_eng.htm >.

Universal Serial Bus Revision 2.0 specification [Online]. www.usb.org. Revisión 2.0 Abril 27, 2000. Disponible en Internet: http://www.usb.org/develo pers/docs/usb_20.pdf; 650p.

ANEXO 1. COMPARADOR DE MONEDA.

1. RANURA PARA LA CAIDA DE MONEDAS

Es el espacio superior del módulo comparador el cual se encarga de recibir la moneda que ingresa el usuario y conducirla hacia las bobinas sensoras.

2. CONEXIÓN DE VOLTAJE

Este sistema posee la ventaja de que puede ser alimentado de varias maneras ya que acepta niveles de tensión $12 V_{DC}$, $12V_{AC}$ y $24V_{AC}$. Para este caso y debido a la fuente de alimentación que posee la máquina se decidió hacer funcionar el sistema sensor de monedas en modo de $12 V_{DC}$, con el cual se obtiene gran sensibilidad y se quita la necesidad de tener una fuente de tensión de AC.

3. PROCEDIMIENTO PARA CALIBRAR SELECTIVIDAD DEL SISTEMA DETECTOR DE MONEDAS

El sistema detector de monedas debe calibrarse de manera adecuada para que detecte las monedas del valor deseado. Para este propósito el sistema posee un potenciómetro de ajuste el cual se encarga de calibrar los niveles de tensión del circuito de tal manera que se puede determinar la sensibilidad del sensor.

Para lograr un ajuste correcto del comparador deben realizarse tres pruebas:

FULL REJECT: esto quiere decir que el sistema queda calibrado para rechazar todas las monedas que ingresen al sistema, incluso aquellas que son iguales a la moneda residente. Para ello se debe girar el potenciómetro de ajuste (figura 12) en sentido horario hasta llegar a su extremo, entonces el sistema se energiza y se prueba para varios tipos de monedas. Se observa que efectivamente el sistema rechaza de inmediato cualquiera de ellas.

FULL ACCEPT: quiere decir que el sistema se calibra de tal manera que acepta cualquier moneda que el usuario ingrese al sistema. Para lograr esta configuración se debe girar el potenciómetro de ajuste (figura 12) en sentido anti horario hasta llegar a su extremo. Después se energiza el sistema y se prueba para varios tipos de monedas; se observa que el sistema si acepta cualquier moneda ingresada.

EXCELLENT DISCRIMINATION: mediante este procedimiento se ajusta el sistema de manera que acepte únicamente las monedas del mismo valor y constitución que la moneda residente. Este proceso se realiza de la siguiente manera:

- Ajustar el potenciómetro en sentido horario hasta que las monedas apropiadas sean rechazadas.
- Insertar una moneda falsa o una pieza de metal similar a una moneda, también debe ser rechazada.
- Reajustar el potenciómetro en sentido anti horario de tal manera que solo se acepten las monedas apropiadas.
- Repetir los pasos anteriores mientras sea necesario hasta que se obtenga la mejor discriminación posible.
- Después de lograr un buen margen de discriminación, se deben ingresar por lo menos seis monedas del valor correcto y una falsa; el sistema rechaza únicamente la moneda falsa. Si esto no se logra se deben repetir los pasos anteriores.

ANEXO 2. PANTALLA LCD ALFANUMERICA

Tabla 1. Funciones de los pines del LCD L2432.

PIN N°	SÍMBOLO	DESCRIPCIÓN
1	V_{SS}	Patilla de tierra de alimentación
2	V_{DD}	Patilla de alimentación de 5 V
3	V_O	Patilla de contraste del cristal líquido. Normalmente se conecta a un potenciómetro a través del cual se aplica una tensión variable entre 0 y +5V que permite regular el contraste del cristal líquido.
4	RS	Selección del registro de control/registro de datos: RS=0 Selección del registro de control RS=1 Selección del registro de datos
5	R/W	Señal de lectura/escritura R/W=0 El módulo LCD es escrito R/W=1 El módulo LCD es leído
6	E	Señal de activación del módulo LCD: E=0 Módulo desconectado E=1 Módulo conectado
7-14	D0-D7	Bus de datos bi-direccional. A través de estas líneas se realiza la transferencia de información entre el módulo LCD y el sistema informático que lo gestiona

El Display LCD L2432 posee tres modos de funcionamiento que son: Modo Comando, Modo Carácter o Dato, y Modo Lectura del “Busy Flag” o LCD Ocupada. Bajo estos tres modos de funcionamiento puede ejecutar varias funciones tales como:

- *Clear Display*: Borra pantalla y devuelve el cursor a la posición inicial.
- *Return Home*: cursor a dirección origen. Devuelve el cursor a al posición original de la DDRAM quedando intacto su contenido.
- *Entry Mode Set*: Modo entrada. Establece las características de escritura de los datos *Shift e Increment/Decrement*:

S=0: la información visualizada en pantalla no se desplaza al escribir un nuevo carácter.

S=1: la información visualizada se desplaza al escribir un nuevo carácter. La pantalla se desplaza en el sentido indicado por el bit *I/D* cuando el cursor llega al filo de la pantalla.

I/D=1: incremento automático de la posición del cursor. La posición de la DDRAM se incrementa automáticamente tras cada lectura o escritura a la misma.

I/D=0: decremento de la posición del cursor. Se decrementa el puntero de la DDRAM.

- *Display*
- *Control:* control de la pantalla:

B=0: Blink OFF, no hay efecto de parpadeo del cursor.

B=1: Blink ON, efecto de parpadeo con un cursor rectangular.

C=0: Cursor OFF, el cursor no se visualiza.

C=1: Cursor ON, el cursor es visualizado.

D=0: Display OFF, el Display se apaga.

D=1: Display ON, el Display se enciende.

- *Cursor and Display Shift:* control de los desplazamientos del cursor y de la pantalla:

R/L=0: Left. A la izquierda.

R/L=1: Right. A la derecha.

S/C=0: el efecto de desplazamiento se aplica solo sobre el cursor sin alterar el contenido de la DDRAM.

S/C=1: el efecto de desplazamiento se aplica sobre todo el Display.

- *Function Set:* características de control hardware:

F=0: Font. Caracteres de 5x7 puntos.

F=1: Font. Caracteres de 5x10 puntos.

N=0: Number Line. Pantalla de 1 línea.

N=1: Number Line. Pantalla de 2 líneas.

DL=0: Data Length. Comunicación con 4 bits. Indica al Display LCD que solamente se va a utilizar las líneas D7, D6, D5 y D4 para enviarle datos y que se hará enviando primero el nibble alto y a continuación el nibble bajo del dato.

DL=1: Data Length. Comunicación mediante 8-bits.

- *Set CGRAM Address.* Se escribe sobre la dirección CGRAM señalada.
- *Set DDRAM Address.* Esta instrucción se utiliza para modificar el puntero a la DDRAM. Esto sirve para escribir en cualquier posición de la pantalla.
- *Read Busy Flag.* Lee el BF indicando si hay una operación interna en curso y lee, además, el contenido de la dirección DDRAM apuntada.

Tabla 2. Comandos del Display LCD L2432.

COMANDO	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
<i>Clear Display</i>	0	0	0	0	0	0	0	0	0	1
<i>Return Home</i>	0	0	0	0	0	0	0	0	1	*
<i>Entry Mode Set</i>	0	0	0	0	0	0	0	1	I/D	S
<i>Display Control</i>	0	0	0	0	0	0	1	D	C	B
<i>Cursor and Display Shift</i>	0	0	0	0	0	1	S/C	R/L	*	*
<i>Function Set</i>	0	0	0	0	1	DL	N	F	*	*
<i>Set CGRAM address</i>	0	0	0	1	CGRAM Address					
<i>Set DDRAM address</i>	0	0	1	DDRAM Address						
<i>Read Busy Flag</i>	0	1	BF	DDRAM Address						
<i>Write RAM</i>	1	0	Write Data							
<i>Read RAM</i>	1	1	Read Data							

Fuente: Microcontrolador PIC16F84 Desarrollo de Proyectos.