

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

**“DISEÑO Y CONSTRUCCIÓN DE UN DISPOSITIVO DE
DESARROLLO DE MICROCONTROLADORES PARA EL
LABORATORIO DE SISTEMAS MICROPROCESADOS,
UTILIZANDO LA TECNOLOGÍA BLUETOOTH PARA LA
DESCARGA DE LOS PROGRAMAS”**

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

**DE LA TORRE GUZMÁN JOSÉ JAVIER
tabero86@hotmail.com**

**DIRECTOR: ING. JAIME VELARDE
jaime.velarde@epn.edu.ec**

Quito, septiembre 2011

DECLARACIÓN

Yo José Javier De la Torre Guzmán, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

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

José Javier De la Torre Guzmán

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por José Javier De la Torre Guzmán, bajo mi supervisión.

Ing. Jaime Velarde
DIRECTOR DEL PROYECTO

AGRADECIMIENTO

Agradezco a mis padres, Javier de la Torre y Marina Guzmán, por el inmenso esfuerzo, dedicación y amor que han puesto y siguen poniendo para que salga adelante.

A mis hermanos, Teresa y Elhías, por compartir tantos lindos momentos junto a mí.

A Tayra Revelo, por su cariño, apoyo y empuje que me ha brindado para cumplir una meta más de mi vida.

A mi gran amigo Ing. Luis Torres, por su ayuda incondicional.

Al Ing. Jaime Velarde, por su apoyo y sabiduría, para guiarme en el desarrollo de este proyecto.

A mis amigos y compañeros de aula, por toda las experiencias vividas en la politécnica.

Javier

DEDICATORIA

A mi padre Javier, mi madre Mariana y a mis hermanos Teresa y Elhías, son la inspiración para seguir avanzando en la vida.

CONTENIDO

1	CAPÍTULO I: SITUACIÓN ACTUAL EN LA IMPLEMENTACIÓN DE LAS PRÁCTICAS EN EL LABORATORIO DE SISTEMAS MICROPROCESADOS.....	1
1.1	MECANISMOS ACTUALES DE LA IMPLEMENTACIÓN DE LAS PRÁCTICAS EN EL LABORATORIO DE SISTEMAS MICROPROCESADOS. ..	1
1.1.1	SITUACIÓN A NIVEL DE HARDWARE	1
1.1.1.1	Microcontrolador.....	2
1.1.1.2	Programador	6
1.1.2	SITUACIÓN A NIVEL DE SOFTWARE.....	6
1.1.2.1	Definición del Problema.....	7
1.1.2.2	Elaboración del Algoritmo	7
1.1.2.3	Escritura del Programa en Mnemónicos	8
1.1.2.4	Traducción del Programa en Lenguaje de Máquina.....	8
1.2	ENCUESTA REALIZADA A LOS ESTUDIANTES DE LABORATORIO DE SISTEMAS MICROPROCESADOS.	8
1.3	EVALUACIÓN DE LA INFORMACIÓN	22
1.3.1	Fortalezas en la Implementación de las Prácticas	22
1.3.2	Debilidades en la Implementación de las Prácticas.....	22
1.3.3	Soluciones que Implementa el Dispositivo del Proyecto	23
2.	CAPÍTULO II: DISEÑO Y CONSTRUCCIÓN DEL DISPOSITVO.....	25
2.1	DISEÑO DEL HARDWARE	26
2.1.1	VISUALIZACIÓN (LCD GRÁFICO).....	27
2.1.2	TECLADO 4X4.....	29
2.1.3	MÓDULO DE BLUETOOTH	30
2.1.3.1	Descripción del Módulo RN-41	30
2.1.3.2	Mini USB-Bluetooth	32
2.1.3.3	Comunicación entre la PC y el dispositivo de desarrollo	33
2.1.4	FUENTE DE ALIMENTACIÓN.....	35
2.1.5	CONSTRUCCIÓN DE LA PLACA DEL PDAT164P.....	36
2.1.5.1	SOFTWARE PROTEUS 7.6	36
2.1.5.1.1	Isis y Ares	36
2.2	DISEÑO DEL SOFTWARE	41
2.2.1	BOOTLOADER	41

2.2.1.1	Secciones Aplicación y Bootloader de la memoria Flash.....	41
2.2.1.1.1	Sección Aplicación	43
2.2.1.1.2	Sección Bootloader (BLS).....	43
2.2.1.2	Lectura Mientras se Escribe y No Lectura Mientras se Escribe en las secciones de la memoria Flash	44
2.2.1.2.1	Lectura Mientras Se Escribe RWW.....	45
2.2.1.2.2	Sección de No Lectura Mientras se Realiza la Escritura NRWW	46
2.2.1.3	Bits de Bloqueo del Bootloader	46
2.2.1.4	Organización de la Flash	46
2.2.1.5	Auto programando la Flash	48
2.2.2	DISEÑO DEL BOOTLOADER.....	49
2.2.3	Descripción del Registro SPMCSR.....	49
2.2.3.1	Descripción del programa computacional Avr Studio 4	52
2.2.3.2	Programador Progisp.....	54
2.2.4	Programación del Bootloader	54
2.2.4.1	Configuración de fusible	55
2.2.4.2	Borrar por página	56
2.2.4.3	Cargando la pagina de Buffer.....	56
2.2.4.4	Escribir por páginas en la memoria.....	57
2.2.4.5	Activación de la Sección RWW.....	57
2.2.4.6	Diagrama de flujo.....	58
2.3	AMBIENTE GRÁFICO	60
2.3.1	VISUAL BASIC.NET	60
2.3.1.1	Interfaz gráfica para la descarga del programa	61
2.3.1.2	Diagrama de flujo de la Interfaz gráfica.....	61
2.4	COSTOS	66
3.	CAPÍTULO TERCERO: EJEMPLOS DE PRÁCTICAS DE LABORATORIO.....	68
3.1	INTRODUCCIÓN	68
3.2	WINAVR.....	68
3.2.1	LIBRERIAS	69
3.2.2	VARIABLES GLOBALES	70
3.2.3	DEFICIONES Y MACROS	70
3.2.4	RUTINAS DE INTERRUPCIÓN	70
3.2.5	DEFINICIÓN DE FUNCIONES.....	72
3.2.6	CUERPO DEL PROGRAMA.....	72

3.2.6.1	Declaración de variables	73
3.2.6.2	Configuración de los puertos.....	73
3.2.6.2.1	DDR _x	73
3.2.6.2.2	PORT _x	73
3.2.6.2.3	PIN _x	74
3.2.6.3	Inicialización de registros.....	74
3.2.6.4	Habilitación de Interrupciones	74
3.2.6.5	Inicialización de funciones y variables	75
3.3	INTRODUCCIÓN AL LENGUAJE C PARA EL COMPILADOR WINAVR	75
3.3.1	TIPOS DE DATOS BÁSICOS	75
3.3.2	OPERADORES	76
3.3.3	ESTRUCTURAS DE CONTROL	77
3.3.4	Decisión Binaria	78
3.3.4.1	Sentencia if.....	78
3.3.4.2	Sentencia if-else	78
3.3.4.3	Sentencia else-if	78
3.3.4.4	Switch.....	79
3.3.5	SENTENCIAS DE REPITICIÓN	79
3.3.5.1	While	79
3.3.5.2	do – While	79
3.3.5.3	For	80
3.3.6	MANEJO DE LO PÓRTICOS.....	80
3.4	REALIZACIÓN DE LAS PRÁCTICAS DE LABORATORIO DE SISTEMAS MICROPROCESADOS EN WINAVR.....	82
3.4.1	RUTINA PARA EL MANEJO DE LA LCD GRÁFICA.....	82
3.4.1.1	Creación de menús para la GLCD.....	83
3.4.1.1.1	Creación de menús en Paint.....	83
3.4.1.1.2	Trasformación de BMP a ASM	84
3.4.1.2	Control de la GLCD gráfica.	85
3.4.1.3	Diagrama de flujo para el manejo de la GLCD.....	87
3.4.1.4	RUTINA PARA EL MANEJO DEL TECLADO 4X4.....	88
3.4.1.5	Diagrama de Flujo par el manejo del Teclado 4x4	89
3.4.2	PRÁCTICA DE CUENTA PERSONAS	90
3.4.2.1	Optoacoplador	90
3.4.2.2	Diagrama del circuito	91

3.4.2.3	Diagrama de flujo.....	92
3.4.3	PRÁCTICA DE CERRADURA ELECTRÓNICA.....	94
3.4.3.1	Diagrama de flujo.....	95
3.4.4	PRÁCTICA DEL VOLTÍMETRO.....	96
3.4.4.1	Diagrama del circuito.....	96
3.4.4.2	Diagrama de flujo.....	97
4.	CAPÍTULO IV: PRUEBAS DE FUNCIONAMIENTO.....	98
4.1	PROGRAMA QUEMADOR.....	98
4.1.1	PARÁMETROS DE LA COMUNICACIÓN.....	99
4.1.2	BOTÓN DE ABRIR PUERTO.....	100
4.1.3	BOTÓN DE BUSCAR.....	100
4.1.4	RUTA DEL ARCHIVO.....	101
4.1.5	PANTALLAS DE INFORMACIÓN.....	101
4.1.5.1	Pantalla de Control.....	102
4.1.5.2	Pantalla de datos.....	102
4.1.6	BOTÓN DE ENVÍO.....	103
4.1.7	BARRA DE PROCESAMIENTO.....	103
4.2	PRUEBAS DE DESCARGAS DE LAS PRÁCTICAS.....	104
4.2.1	DESCARGA DEL PROGRAMA DE LA CERRADURA ELECTRÓNICA.....	104
4.2.2	DESCARGA DEL PROGRAMA DE CUENTA PERSONAS.....	106
4.2.3	DESCARGA DEL PROGRAMA DE VOLTÍMETRO DIGITAL.....	107
5.	CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES.....	110
5.1	CONCLUSIONES.....	110
5.2	RECOMENDACIONES.....	111
	REFERENCIAS BIBLIOGRÁFICAS.....	111
	LIBROS.....	111
	REFERENCIAS ELECTRÓNICAS.....	111
	ANEXOS.....	113
	ANEXO A: ENCUESTAS	
	ANEXO B: MANUAL DE USUARIO DEL EQUIPO	
	ANEXO C: SUMARIO DEL ATMEGA164P	
	ANEXO D: CARACTERÍSTICAS DEL MÓDULO BLUETOOTH RN-41	
	ANEXO E: CARACTERÍSTICAS DE LA GLCD JH12864E	
	ANEXO F: CARACTERÍSTICAS DEL REGULADOR LM7805	

ÍNDICE DE FIGURAS

<i>Fig. 1-1 Gráfica de la Tabulación a la Pregunta No.1 de la Encuesta</i>	<i>9</i>
<i>Fig. 1-2 Gráfica de la Tabulación a la Pregunta No. 2 de la Encuesta</i>	<i>10</i>
<i>Fig. 1-3 Gráfica de la Tabulación a la Pregunta No. 3 de la Encuesta</i>	<i>11</i>
<i>Fig. 1-4 Gráfica de la Tabulación a la Pregunta No. 4 de la Encuesta</i>	<i>12</i>
<i>Fig. 1-5 Gráfica de la Tabulación a la Pregunta No. 5 de la Encuesta</i>	<i>13</i>
<i>Fig. 1-6 Gráfica de la Tabulación a la Pregunta No.6 de la Encuesta</i>	<i>14</i>
<i>Fig. 1-7 Gráfica de la Tabulación a la Pregunta No.7 de la Encuesta</i>	<i>15</i>
<i>Fig. 1-8 Gráfica de la Tabulación a la Pregunta No. 8 de la Encuesta</i>	<i>16</i>
<i>Fig. 1-9 Gráfica de la Tabulación a la Pregunta No. 9 de la Encuesta</i>	<i>17</i>
<i>Fig. 1-10 Gráfica de la Tabulación a la Pregunta No.10 de la Encuesta.....</i>	<i>18</i>
<i>Fig. 1-11 Gráfica de la Tabulación a la Pregunta No.11 de la Encuesta.....</i>	<i>19</i>
<i>Fig. 1-12 Gráfica de la Tabulación a la Pregunta No.12 de la Encuesta.....</i>	<i>20</i>
<i>Fig. 1-13 Gráfica de la Tabulación a la Pregunta No.12 de la Encuesta.....</i>	<i>21</i>
<i>Fig. 2-1 Diagrama de Bloques de PDAT164P.....</i>	<i>26</i>
<i>Fig. 2-2 Diagrama de bloques de la LDC gráfica JHD12864E</i>	<i>27</i>
<i>Fig. 2-3 Diagrama de interconexión de los elementos que conforma el PDAT164P... </i>	<i>29</i>
<i>Fig. 2-4 Módulo Bluetooth RN-41.....</i>	<i>31</i>
<i>Fig. 2-5 Mini USB Bluetooth 2.0.....</i>	<i>32</i>
<i>Fig. 2-6 Muestra la detección automática del módulo RN-41</i>	<i>32</i>
<i>Fig. 2-7 Diagrama de Bloque de la comunicación</i>	<i>33</i>
<i>Fig. 2-8 Conexión del Oscilador de Cristal</i>	<i>34</i>
<i>Fig. 2-9 Diagrama del circuito de la fuente de alimentación.</i>	<i>35</i>
<i>Fig. 2-10 Enrutamiento, ubicación y edición de componentes del PDAT164P.....</i>	<i>37</i>
<i>Fig. 2-11 Pistas del dispositivo realizadas con ARES</i>	<i>37</i>
<i>Fig. 2-12 Placa del PDAT164P.....</i>	<i>38</i>
<i>Fig. 2-13 Montaje del PCB del PDAT164P en la caja</i>	<i>39</i>
<i>Fig. 2-14 Montaje de la GLCD y el teclado 4x4 en la parte delantera de la caja.....</i>	<i>39</i>
<i>Fig. 2-15 Salidas de los pórtricos A y B del microcontrolador y el conector de la fuente de voltaje</i>	<i>40</i>
<i>Fig. 2-16 PDAT164P.....</i>	<i>40</i>
<i>Fig. 2-17 Secciones de Memoria del Microcontrolador Atmega164P.....</i>	<i>42</i>

Fig. 2-18 Secciones RWW vs. NRWW 45

Fig. 2-19 Direccionamiento de la Flash durante SPM 48

Fig. 2-20 Registro de estatus y control para almacenar en la memoria del programa 49

Fig. 2-21 Ventana para escoger el lenguaje de programación..... 53

Fig. 2-22 Ventana de Trabajo del software AVR Studio 4..... 53

Fig. 2-23 Software Progisp 1.6.6 54

Fig. 2-24 Diagrama de flujo general del programa Bootloader..... 59

Fig. 2-25 Interfaz gráfica para la descarga del programa 61

Fig. 2-26 Ejemplo de un archivo de formato intel Hex..... 62

Fig. 2-27 Diagrama de flujo del programa principal de la interfaz gráfica..... 65

Fig. 3-1 Registros asociados a cada p rtico del microcontrolador Atmega 164P..... 73

Fig. 3-2 Dise o del men  principal que va mostrar la GLCD en el Software Paint 84

Fig. 3-3 Conversi n de la imagen en extensi n BMP a lenguaje de m quina..... 85

Fig. 3-4 Funci n para inicializar la GLCD 87

Fig. 3-5 Funci n para habilitar y deshabilitar la se al..... 87

Fig. 3-6 Diagrama de Flujo de la Librer a que maneja la GLCD..... 88

Fig. 3-7 Diagrama de flujo de la Librer a para el manejo del teclado..... 89

Fig. 3-8 Optoacoplador GP3S62..... 90

Fig. 3-9 Diagrama circuital del optoacoplador GP3S62 91

Fig. 3-10 Diagrama circuital de la pr ctica 91

Fig. 3-11 Diagrama de flujo de la pr ctica cuenta personas 93

Fig. 3-12 Diagrama de flujo de la pr ctica de la Cerradura electr nica..... 95

Fig. 3-13 Diagrama del circuito de la pr ctica del volt metro 96

Fig. 3-14 Diagrama de flujo de la pr ctica del volt metro 97

Fig. 4-1 Interfaz QUEMADOR..... 98

Fig. 4-2 Selecci n de los par metros de la comunicaci n..... 99

Fig. 4-3 Bot n de abrir puerto 100

Fig. 4-4 Bot n de explorar 101

Fig. 4-5 Ruta del archivo .HEX..... 101

Fig. 4-6 Pantalla de control 102

Fig. 4-7 Pantalla de Datos 102

Fig. 4-8 Bot n de enviar..... 103

Fig. 4-9 Barra de procesamiento 103

Fig. 4-10 PDAT164P en la posici n de Bootloader..... 104

<i>Fig. 4-11</i>	<i>Descara de la práctica de la Cerradura electrónica.....</i>	<i>105</i>
<i>Fig. 4-12</i>	<i>PDAT164P en la posición de Aplicación</i>	<i>105</i>
<i>Fig. 4-13</i>	<i>Descarga de la práctica de cuenta personas.....</i>	<i>106</i>
<i>Fig. 4-14</i>	<i>Práctica de cuenta personas.....</i>	<i>107</i>
<i>Fig. 4-15</i>	<i>Ejemplo 1 de la práctica voltímetro digital en el PDAT164P.....</i>	<i>108</i>
<i>Fig. 4-16</i>	<i>Ejemplo 2 de la práctica voltímetro digital en el PDAT164P.....</i>	<i>108</i>
<i>Fig. 4-17</i>	<i>Ejemplo 3 de la práctica voltímetro digital en el PDAT164P.....</i>	<i>109</i>

ÍNDICE DE TABLAS

<i>Tabla 1-1 Cuadro de tabulación de la Pregunta No. 1</i>	9
<i>Tabla 1-2 Cuadro de tabulación de la Pregunta No. 2</i>	10
<i>Tabla 1-3 Cuadro de tabulación de la Pregunta No. 3</i>	11
<i>Tabla 1-4 Cuadro de tabulación de la Pregunta No. 4</i>	12
<i>Tabla 1-5 Cuadro de tabulación de la Pregunta No. 5</i>	13
<i>Tabla 1-6 Cuadro de tabulación de la Pregunta No. 6</i>	14
<i>Tabla 1-7 Cuadro de tabulación de la Pregunta No. 7</i>	15
<i>Tabla 1-8 Cuadro de tabulación de la Pregunta No. 8</i>	16
<i>Tabla 1-9 Cuadro de tabulación de la Pregunta No. 9</i>	17
<i>Tabla 1-10 Cuadro de tabulación de la Pregunta No. 10</i>	18
<i>Tabla 1-11 Cuadro de tabulación de la Pregunta No. 11</i>	19
<i>Tabla 1-12 Cuadro de tabulación de la Pregunta No. 12</i>	20
<i>Tabla 1-13 Cuadro de tabulación de la Pregunta No. 13</i>	21
<i>Tabla 2-1 Descripción de los terminales del GLCD</i>	28
<i>Tabla 2-2 Descripción de los Pines de Módulo Bluetooth RN-41</i>	31
<i>Tabla 2-3 Ejemplos de valores de UBRR para un frecuencia de oscilador</i>	34
<i>Tabla 2-4 Modo de operación del Oscilador de Cristal</i>	35
<i>Tabla 2-5 Configuración del Tamaño del Boot</i>	43
<i>Tabla 2-6 Modos de protección de la sección de Aplicación</i>	43
<i>Tabla 2-7 Modos de protección de la sección Bootloader</i>	44
<i>Tabla 2-8 Organización de la memoria Flash del Atmega164p</i>	47
<i>Tabla 2-9 Selección del Tamaño de las Secciones Aplicación y Bootloader</i>	56
<i>Tabla 2-10 Descripción de costos del PDATP164p</i>	67
<i>Tabla 3-1 Interrupciones del microcontrolador ATMEGA 164P</i>	72
<i>Tabla 3-2 Operadores Aritméticos</i>	76
<i>Tabla 3-3 Operadores lógicos y relacionales</i>	77
<i>Tabla 3-4 Operadores Bit a Bit</i>	77
<i>Tabla 3-5 Conjunto de Instrucciones para el control de la GLCD JHD12864E</i>	86

RESÚMEN

El aprendizaje de los sistemas microprocesados se empieza conociendo la estructura interna del microcontrolador, para luego aprender los mnemónicos que nos permita realizar las aplicaciones que puedan resolver los problemas con sistemas microprocesados, pero se debe tener en cuenta que los dispositivos que se conecta a un procesador han evolucionado y el estudiante debe estar capacitado en su utilización, por lo cual este proyecto da una alternativa de manejo de otras tecnologías, como también facilitar la descarga del código fuente de la práctica mediante una comunicación inalámbrica, con esto se pretende un mejor aprendizaje de los microcontroladores facilitando al estudiante el desarrollo de las prácticas.

En el capítulo 1 se indica la actual implementación de las prácticas de laboratorio, luego se realiza una encuesta para conocer la opinión del estudiante sobre la implementación en el laboratorio en base a sus fortalezas como debilidades que luego se toman en cuenta para el diseño del dispositivo.

El capítulo 2 se detalla las características técnicas de los elementos que conforman el dispositivo como son: la LCD gráfica (GLCD) y el módulo Bluetooth, para realizar el diseño del software y hardware del dispositivo, la interfaz gráfica para la PC y finalmente un estimado del costo del equipo.

El capítulo 3 se describe la forma de programar los microcontroladores en un lenguaje de alto nivel como es "C", para lo cual se realizaron tres prácticas de laboratorio en este lenguaje.

El capítulo 4 se presentan las pruebas y funcionamiento de dispositivo de desarrollo, para lo cual se implemento las tres prácticas antes desarrolladas y se explica el manejo de la interfaz gráfica y del dispositivo.

En el capítulo 5 se redacta las conclusiones y recomendaciones que se llegaron luego de terminar este proyecto.

PRESENTACIÓN

Para el desarrollo de las prácticas de laboratorio no existe un profundo conocimiento de la tecnología de hoy, como es la comunicación inalámbrica, GLCD, touchscreen¹, entre otras tecnologías que se pueden acoplar a un microcontrolador para crear sistemas inteligentes que faciliten a la persona común su bienestar, a la vez que su uso sea de manejo fácil.

Los estudiantes de laboratorio deben ser capaces de dar este servicio a la comunidad, más aún que su vida académica se desenvuelve en el ámbito de la electrónica y las telecomunicaciones.

Este proyecto busca esencialmente un cambio en el desarrollo de las prácticas de laboratorio para que estén más acordes con las necesidades actuales de las personas en general y puntualmente en beneficiar el aprendizaje de los estudiantes sobre los microcontroladores facilitando la introducción de datos, salida de los mismos, como también cambios en el software, para lo cual en este proyecto se diseña y construye el dispositivo de desarrollo que facilite todas estas características mencionadas.

¹ Touchscreen .- Pantalla que mediante un toque directo sobre su superficie permite la entrada de datos y órdenes al dispositivo.

1 CAPÍTULO I: SITUACIÓN ACTUAL EN LA IMPLEMENTACIÓN DE LAS PRÁCTICAS EN EL LABORATORIO DE SISTEMAS MICROPROCESADOS

1.1 MECANISMOS ACTUALES DE LA IMPLEMENTACIÓN DE LAS PRÁCTICAS EN EL LABORATORIO DE SISTEMAS MICROPROCESADOS.

En este capítulo se analiza la actual implementación de las prácticas de laboratorio de los estudiantes de la Facultad de Ingeniería Electrónica (FIEE) para la formación en sistemas microprocesados.

1.1.1 SITUACIÓN A NIVEL DE HARDWARE

Para la realización de las prácticas se necesitan dentro del hardware varios dispositivos que se detallan a continuación:

- Microcontrolador.- es el circuito integrado que incluye en su interior las tres unidades funcionales de una computadora: unidad central de procesamiento, memoria y unidades de E/S (entrada/salida).
- Programador.- equipo que sirve para transportar el programa desde la computadora hacia el microcontrolador.
- Protoboard.- placa de uso genérico reutilizable o semipermanente, usada para construir prototipos de circuitos electrónicos con o sin soldadura. Normalmente se utilizan para la realización de pruebas experimentales.
- Dip switch.- es un conjunto de interruptores eléctricos que se presenta en un formato encapsulado que se utiliza para la introducción de datos binarios al microcontrolador.
- Display de 7 segmentos.- dispositivo electrónico que está compuesto de 7 segmentos que se pueden encender o apagar individualmente, y que se utiliza para la salida de datos entre los más comunes decimales y hexadecimales.

- Teclado.- arreglo de pulsadores o teclas que se utiliza para la introducción de datos decimales, y otras funciones de control, mediante un subprograma.
- Otros.

Se han descrito los principales dispositivos electrónicos que se utilizan en el laboratorio para la elaboración de la práctica; cabe recalcar que la interconexión de los mismos depende del estudiante, lo que quiere decir que el correcto funcionamiento de la práctica dependerá si estos fueron conectados de manera correcta entre ellos.

Hay que tener en cuenta que los dispositivos no contienen ninguna protección para sobrecargas, volviéndolos vulnerables, lo que da lugar a que se inutilicen con facilidad.

A continuación se explica sobre el microcontrolador y el programador, ya que estos son los elementos fundamentales para elaborar la práctica.

1.1.1.1 Microcontrolador

El microcontrolador utilizado en el laboratorio es de la familia ATMEGA denominado atmega164P que es un microcontrolador AVR de 8 bit de alto rendimiento y bajo consumo de potencia.

A continuación se detallan algunas de sus características:

- Voltajes de funcionamiento
 - 2.7 - 5.5V (ATmega164P)
 - 1.8 - 5.5V (ATmega164PV)
 - 200 mA es la corriente máxima en los terminales VCC y GND
- ENTRADA PARA EL RESET
 - Para que se active cuando se polariza

- 0 en cualquier instante
- TERMINALES PARA EL CRISTAL
 - 0 – 20 MHz (ATmega164P)
 - 0 – 10 MHz (ATmega164PV)
- PÓRTICOS DE ENTRADA Y SALIDA PARALELA
 - Pórtico A (8 bits)
 - Pórtico B (8 bits)
 - Pórtico C (8 bits)
 - Pórtico D (8 bits)
- CONVERTOR DE ANALÓGICO A DIGITAL
 - 8 canales de un solo terminal
 - 2 canales diferenciales con ganancia programable de x1, x10 y x200
 - 7 canales diferenciales sólo en el encapsulado TQFP²
- COMPARADOR ANALÓGICO
 - La entrada positiva es AIN0
 - La negativa es AIN1
 - Se puede reemplazar AIN1 por las entradas analógicas ADC0. ADC7
- INTERFACE JTAG³ PARA SISTEMA DE DEPURACIÓN
 - En la depuración se tiene acceso a todos periféricos
 - Programación de la Flash, EEPROM, Fusibles y Bits de seguridad.
 - Depuración soportada por el AVR Studio®

² TQFP (Thin Quad Flat Package) encapsulado de circuito integrado para montaje superficial con los conectores de componentes extendiéndose por los cuatro lados.

³ JTAG (Join Test Action Group) nombre común utilizado para la norma IEEE 1149.1 titulada Standard Test Access Port and Boundary-Scan Architecture para test access ports utilizada para analizar PCBs utilizando escaneo de límites.

- INTERFAZ A PERIFÉRICOS SERIALES
 - Full duplex
 - Tres líneas para comunicaciones sincrónicas
 - Operación maestro / esclavo
 - Siete velocidades programables
 - Bandera de fin de la transmisión

- INTERRUPTIONES EXTERNAS (INT0, INT1 e INT2)
 - Pueden activarse por flanco de subida o de bajada, o por nivel cero lógico
 - También se puede generar por software, si son configurados los terminales como salidas

- TEMPORIZADORES / CONTADORES (Timer0 y Timer2 de 8 bits Timer1 de 16 bits)
 - Dispone de unidades comparadoras.
 - Sirven como Generadores de Frecuencia.
 - Poseen relojes pre escalables de 10 bits.
 - Permiten implementar Moduladores por Ancho del Pulso.

- INTERFACE SERIAL CON DOS LÍNEAS (TWI)
 - Operación maestro / esclavo.
 - Puede trabajar como transmisor o como receptor.
 - Velocidad de transferencia hasta 400 KHz.
 - Longitud de la dirección de 7 bits para 127 esclavos.

- RECEPTORES / TRANSMISORES UNIVERSALES SINCRÓNICOS Y ASINCRÓNICOS (USART0 y USART1)
 - Full duplex
 - Velocidad de alta resolución.
 - Tramas de 5, 6, 7, 8 o 9 bits, con 1 o 2 bits de parada.

- Detector de errores de velocidad y en la trama.
- Operación de maestro o esclavo en comunicaciones sincrónicas

- SALIDA DEL RELOJ

- Habilitación de la señal programando el fusible.
- Incluye como fuente al oscilador interno RC.
- Se puede utilizar el sistema pre escalable para realizar la división de la frecuencia del reloj.

- INTERRUPTACIONES POR CAMBIO DE ESTADO

- Cambios entre PCINT0 y PCINT7 se registra en PCI0.
- Cambios entre PCINT8 y PCINT15 se registra en PCI1.
- Cambios entre PCINT16 y PCINT23 se registra en PCI2.
- Cambios entre PCINT24 y PCINT31 se registra en PCI3.

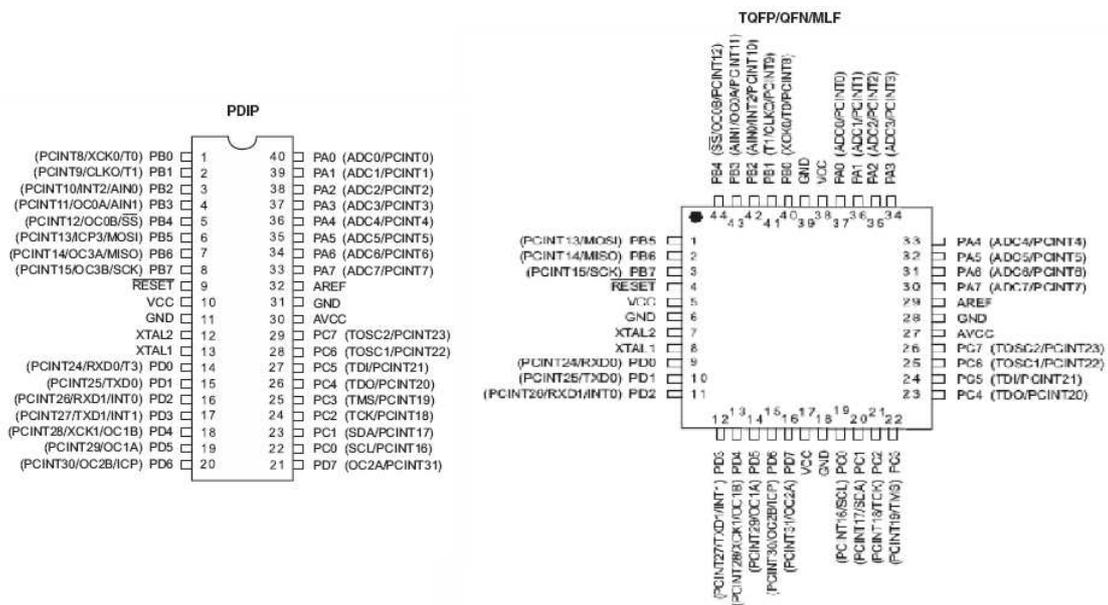


Fig. 1-1 Encapsulado PDIP y TQFP del Microcontrolador Atmega164P

1.1.1.2 Programador

Para desarrollar la práctica se necesita de un equipo que sirva para la transferencia del archivo que contiene el programa desde la PC hacia el microcontrolador, este dispositivo se denomina Programador.

Para la utilización del programador se necesita de un software de instalación para su utilización, se debe indicar que en el laboratorio se utiliza cada vez que lo requiera el programador. La interconexión con la computadora (PC) se produce de manera cableada por medio del puerto USB⁴, también es necesaria la utilización de ambiente gráfico en la PC para la ubicación del programa para cargar en el microcontrolador.

1.1.2 SITUACIÓN A NIVEL DE SOFTWARE

La parte de programación o software de la práctica se define como la creación de una o varias rutinas dentro de un programa el cual se va ejecutar por medio del microcontrolador.

En la actual implementación de la práctica de laboratorio, en lo que concierne al software, se dice que es la creación de una rutina que se va ejecutar infinitamente, lo que en lenguaje de programación, es un lazo infinito donde el programa se ejecuta sin opción de modificación en tiempo real.

La programación está enfocada a aprender los medios para la utilización de las diferentes herramientas que da el microcontrolador como son: los pórtricos de E/S, el procesador aritmético que contiene el microcontrolador, manejo de memorias, el conversor análogo-digital, timers, interrupciones, comunicación serial, para lo cual es indispensable que el estudiante tenga que programar un determinado número de rutinas.

⁴ USB (Universal Serial Bus) o Bus Universal en Serie es un puerto que sirve para conectar periféricos a un ordenador.

El lenguaje de programación utilizado, es aquel que se conoce como Ensamblador, se lo implementa para traducir las instrucciones a lenguaje de máquina, esta traducción es inmediata.

Las instrucciones en ensamblador no son más que mnemónicos⁵ que determinan acciones a ser tomadas directamente por la CPU⁶.

Para la elaboración del programa se siguen los pasos que se detallan:

- Definición de Problema
- Elaboración del Algoritmo
- Escritura del programa en mnemónicos
- Traducción del programa en lenguaje de máquina

1.1.2.1 Definición del Problema

En el ámbito de la definición del problema este se desenvuelve en la comprensión que permita plantear la aplicación que se quiere efectuar en el laboratorio, enfocado en la utilización de las herramientas del microcontrolador, para lo cual se vuelve imprescindible el conocimiento aprendido durante la trayectoria académica.

1.1.2.2 Elaboración del Algoritmo

Para la elaboración del algoritmo es indispensable que el estudiante realice una o varias rutinas que resuelva el problema de la práctica, para lo cual se utiliza ciertos algoritmos para permitir las siguientes aplicaciones entre las que se

⁵ Mnemónicos implementa una representación simbólica de códigos de máquina para facilitar la programación.

⁶ CPU (Central Processing Unit), o simplemente el procesador o microprocesador, es el componente del computador y otros dispositivos programables, que interpreta las instrucciones contenidas en los programas y procesa los datos.

encuentra: activación de pódicos, mostrar en BCD en los displays, realizar las cuatro operaciones matemáticas básicas, etc.

1.1.2.3 Escritura del Programa en Mnemónicos

La computadora como es conocido permite programar en varios lenguajes entre los que están: C, C++, Basic, Ensamblador, para lo cual se usa un cierto número de mnemónicos para traducir a lenguaje de máquina.

1.1.2.4 Traducción del Programa en Lenguaje de Máquina

Para la traducción en lenguaje de máquina se utiliza compiladores que cree el archivo que pueda ejecutar el microcontrolador y que contiene las acciones a ser realizadas por el mismo, para el caso de laboratorio se utiliza el software Avr Studio 4 que es desarrollado por ATMEL, el cual es un software libre que permite programar en lenguaje ensamblador como también en lenguaje C los microcontroladores AVR, para las prácticas se utiliza el primero mencionado, que mediante mnemónicos crea el archivo .HEX que contiene la información en un formato que entienda el microcontrolador.

1.2 ENCUESTA REALIZADA A LOS ESTUDIANTES DE LABORATORIO DE SISTEMAS MICROPROCESADOS.

El diagnóstico situacional del Laboratorio de Sistemas Microprocesados se efectúa mediante la utilización de la técnica de la encuesta, la entrevista y la observación directa en el campo.

La encuesta en los estudiantes de laboratorio de sistemas microprocesados permite detallar los mecanismos actuales de la implementación de las prácticas en los sistemas microprocesados en base a sus fortalezas como debilidades que luego se toman en cuenta para el diseño del dispositivo.

A continuación se pasa a mostrar los resultados de la encuesta:

1. ¿Qué grado de aceptación tiene la forma de desarrollar actualmente las prácticas de Laboratorio?

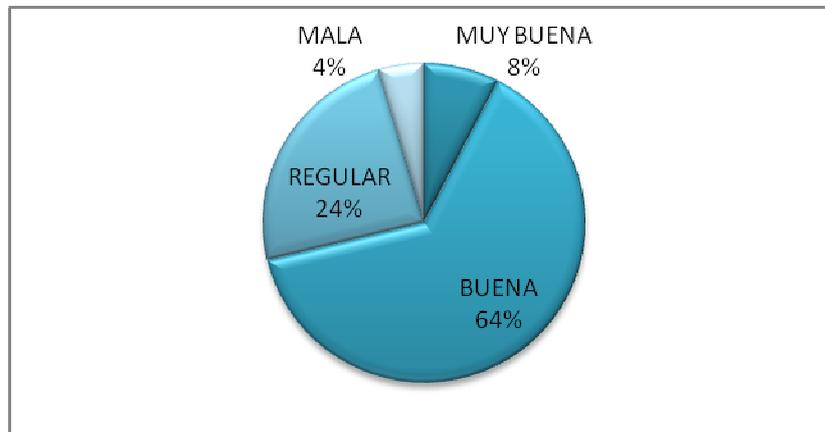


Fig. 1-1 Gráfica de la Tabulación a la Pregunta No.1 de la Encuesta

Cuadro 1		
OPCIÓN	FRECUENCIA	PORCENTAJE
MUY BUENA	5	7,58%
BUENA	42	63,64%
REGULAR	16	24,24%
MALA	3	4,55%
TOTAL	66	100,00%
FUENTE: Estudiantes de Laboratorio de Sistemas Microprocesados		
ELABORACIÓN: Autor de la Investigación		

Tabla 1-1 Cuadro de tabulación de la Pregunta No. 1

ANÁLISIS: Aplicada la encuesta se puede observar en el Cuadro No 1 que hay un alto porcentaje de estudiantes que opinan que la forma de desarrollar las prácticas es buena, lo que quiere decir que si bien existe un porcentaje alto de aceptación no llega a márgenes óptimos, lo que nos permite concluir que es necesario mejorar el método de aprendizaje para el estudiante.

2. ¿Ha tenido usted algún problema con el hardware que implementa en el Laboratorio?

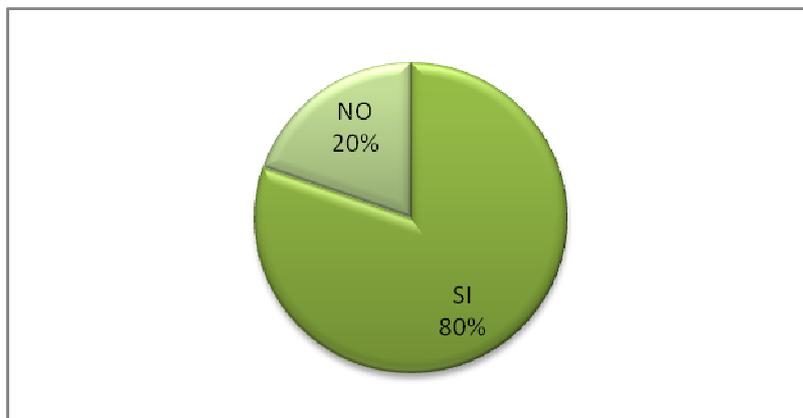


Fig. 1-2 Gráfica de la Tabulación a la Pregunta No. 2 de la Encuesta

Cuadro 2		
OPCIÓN	FRECUENCIA	PORCENTAJE
SI	53	80,30%
NO	13	19,70%
TOTAL	66	100,00%
FUENTE: Estudiantes de Laboratorio de Sistemas Microprocesados		
ELABORACIÓN: Autor de la Investigación		

Tabla 1-2 Cuadro de tabulación de la Pregunta No. 2

ANALISIS: La mayoría de los encuestados opinan que si han tenido algún problema en el hardware al desarrollar las prácticas, esto se debe a que el armado o construcción depende del estudiante algo que es común y entendible en consideración a que el estudiante se halla en un proceso de aprendizaje lo que concluye en que cometa errores comprensibles.

3. ¿El Laboratorio cuenta con los suficientes programadores del microprocesador?

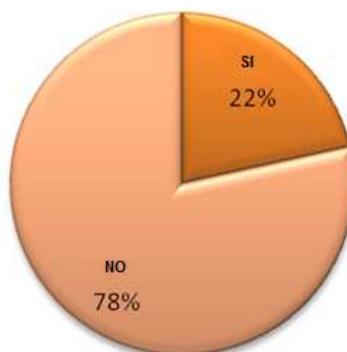


Fig. 1-3 Gráfica de la Tabulación a la Pregunta No. 3 de la Encuesta

Cuadro 3		
OPCIÓN	FRECUENCIA	PORCENTAJE
SI	14	21,54%
NO	51	78,46%
TOTAL	65	100,00%
FUENTE: Estudiantes de Laboratorio de Sistemas Microprocesados		
ELABORACIÓN: Autor de la Investigación		

Tabla 1-3 Cuadro de tabulación de la Pregunta No. 3

ANÁLISIS: Del resultado de la encuesta se puede apreciar que hay insuficientes programadores para el Laboratorio de Sistemas Microprocesados; sin embargo es entendible en la medida que los programadores existentes son adquiridos por el estudiante y no por el laboratorio. Del análisis se concluye que si no se cumple este requerimiento se afecta el desempeño del estudiante debido a que es una herramienta indispensable para el laboratorio, ya que sin él no puede realizar modificaciones de la práctica el estudiante.

4. ¿Qué tipo de problemas tiene al realizar las prácticas?

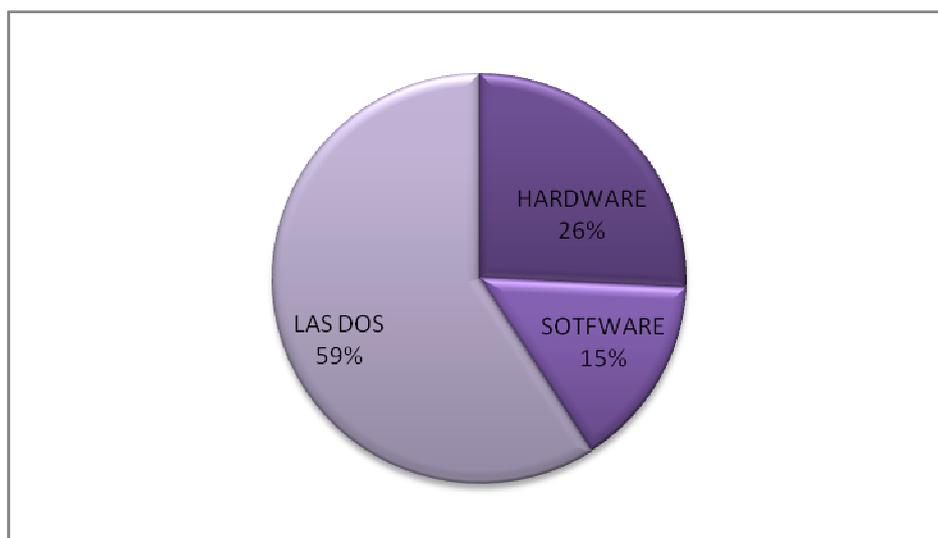


Fig. 1-4 Gráfica de la Tabulación a la Pregunta No. 4 de la Encuesta

Cuadro 4		
OPCIÓN	FRECUENCIA	PORCENTAJE
HARDWARE	17	25,76%
SOTFWARE	10	15,15%
LAS DOS	39	59,09%
TOTAL	66	100,00%
FUENTE: Estudiantes de Laboratorio de Sistemas Microprocesados		
ELABORACIÓN: Autor de la Investigación		

Tabla 1-4 Cuadro de tabulación de la Pregunta No. 4

ANÁLISIS: Se puede apreciar en el Cuadro No 4 que los resultados son ambivalentes. Existe una tendencia mayoritaria de los encuestados, que opinan que el problema en desarrollar la práctica es hardware y el software (opción tercera); si el problema se evidencia en una de las dos situaciones anteriores afecta al estudiante, peor aun cuando se tiene problema en los dos.

5. ¿Puede realizar modificaciones del Software dentro de las prácticas de manera eficiente?

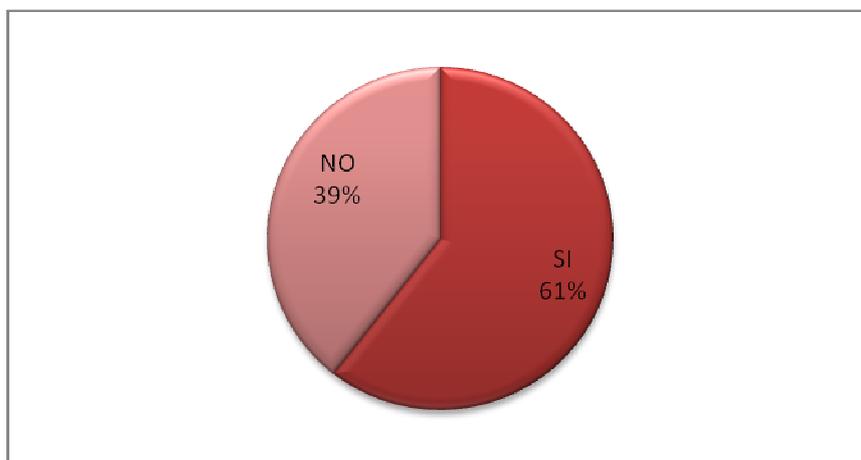


Fig. 1-5 Gráfica de la Tabulación a la Pregunta No. 5 de la Encuesta

Cuadro 5		
OPCIÓN	FRECUENCIA	PORCENTAJE
SI	40	60,61%
NO	26	39,39%
TOTAL	66	100,00%
FUENTE: Estudiantes de Laboratorio de Sistemas Microprocesados		
ELABORACIÓN: Autor de la Investigación		

Tabla 1-5 Cuadro de tabulación de la Pregunta No. 5

ANALISIS: De acuerdo al resultado obtenido se puede identificar que la mayoría de personas encuestadas manifiesta que si puede realizar modificaciones del software, parámetro que permite evidenciar la fortaleza en cuanto a que el estudiante identifica la parte errónea, que a su vez lleva a la conclusión de que es capaz de realizar las correcciones necesarias

6. ¿Cree que es necesario disponer de algún dispositivo que sirva de programador?

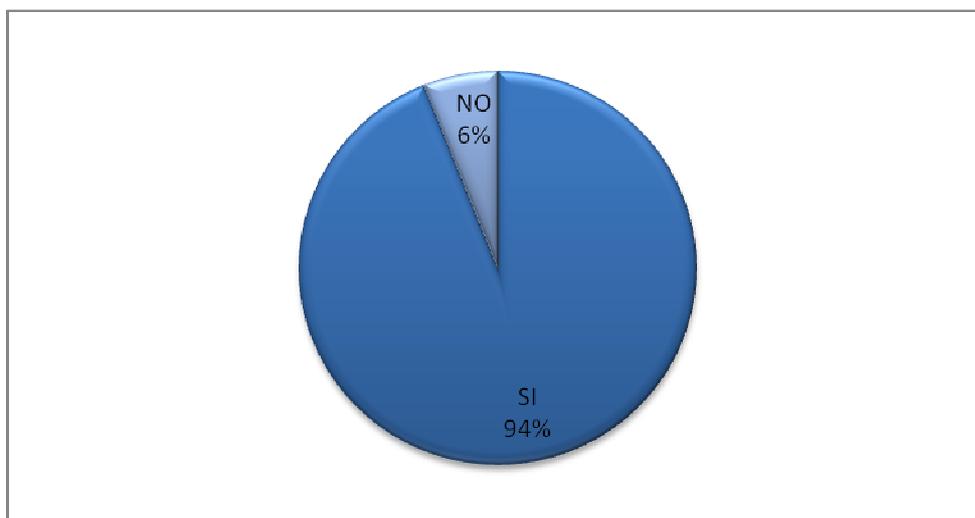


Fig. 1-6 Gráfica de la Tabulación a la Pregunta No.6 de la Encuesta

Cuadro 6		
OPCION	FRECUENCIA	PORCENTAJE
SI	62	93,94%
NO	4	6,06%
TOTAL	66	100,00%
FUENTE: Estudiantes de Laboratorio de Sistemas Microprocesados		
ELABORACIÓN: Autor de la Investigación		

Tabla 1-6 Cuadro de tabulación de la Pregunta No. 6

ANALISIS: En la encuesta existe un pronunciamiento mayoritario en cuanto a que es necesario contar con un dispositivo que sirva de programador, lo que es base cierta para objetivar en la práctica la implementación de este proyecto ya que el dispositivo de desarrollo también va a servir como programador.

7. ¿Cree que es conveniente desarrollar un dispositivo que facilite la implementación de las prácticas? En caso de respuesta negativa termina la encuesta.

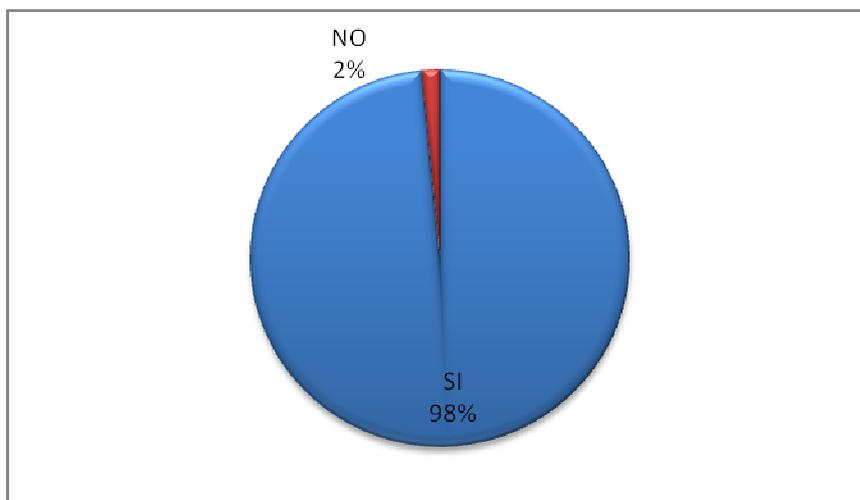


Fig. 1-7 Gráfica de la Tabulación a la Pregunta No.7 de la Encuesta

Cuadro 7		
OPCIÓN	FRECUENCIA	PORCENTAJE
SI	65	98,48%
NO	1	1,52%
TOTAL	66	100,00%
FUENTE: Estudiantes de Laboratorio de Sistemas Microprocesados		
ELABORACIÓN: Autor de la Investigación		

Tabla 1-7 Cuadro de tabulación de la Pregunta No. 7

ANALISIS: El Cuadro No. 7 demuestra que de la totalidad de encuestados en un solo caso se está en desacuerdo con el desarrollo de un dispositivo que facilite la implementación de las prácticas. Los estudiantes reconocen que es necesario de un dispositivo que les ayude en el aprendizaje académico sobre los microcontroladores.

8. ¿Le gustaría un sistema que evite la adquisición del microprocesador y solo sea necesario el programa de la práctica?

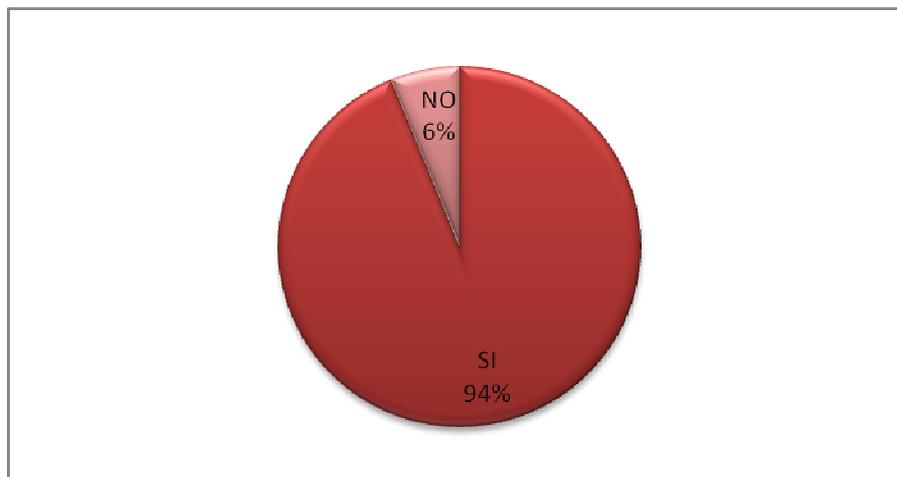


Fig. 1-8 Gráfica de la Tabulación a la Pregunta No. 8 de la Encuesta

Cuadro 8		
OPCION	FRECUENCIA	PORCENTAJE
SI	61	93,85%
NO	4	6,15%
TOTAL	65	100,00%
FUENTE: Estudiantes de Laboratorio de Sistemas Microprocesados		
ELABORACIÓN: Autor de la Investigación		

Tabla 1-8 Cuadro de tabulación de la Pregunta No. 8

ANALISIS: La respuesta es mayoritaria en los encuestados quienes manifiestan una aprobación para que exista un sistema que les evite la adquisición del microcontrolador, esto se debe comprender bajo la óptica que los niveles económicos del estudiante no le permiten adquirir microcontrolador cuando este deja de funcionar.

9. ¿Le gustaría que el dispositivo contenga un sistema de salida gráfica?

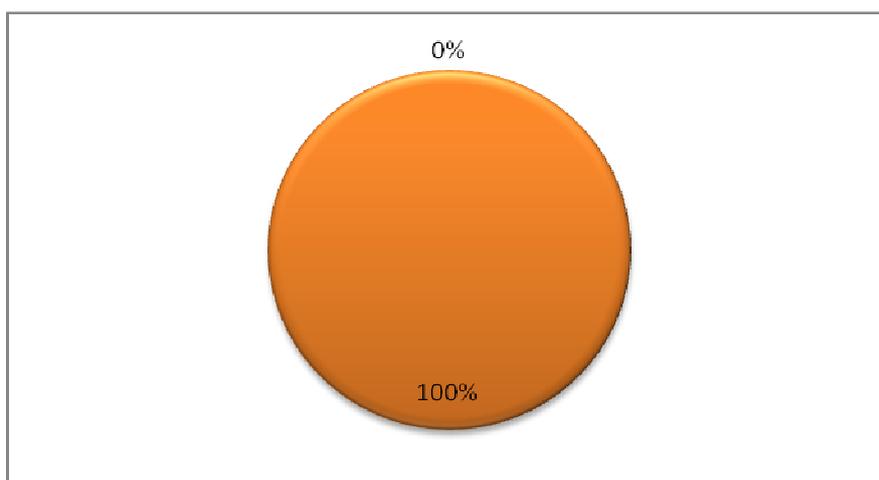


Fig. 1-9 Gráfica de la Tabulación a la Pregunta No. 9 de la Encuesta

Cuadro 9		
OPCIÓN	FRECUENCIA	PORCENTAJE
SI	65	100,00%
NO	0	0,00%
TOTAL	65	100,00%
FUENTE: Estudiantes de Laboratorio de Sistemas Microprocesados		
ELABORACIÓN: Autor de la Investigación		

Tabla 1-9 Cuadro de tabulación de la Pregunta No. 9

ANALISIS: En el Cuadro No. 9 es puntual en su totalidad en referencia a que el dispositivo tenga un sistema de salida gráfica, lo que ratifica la intención de que el dispositivo de desarrollo contenga una LCD gráfica como salida de información.

10. ¿Le gustaría que el dispositivo contenga un dispositivo que facilite el ingreso de los datos?



Fig. 1-10 Gráfica de la Tabulación a la Pregunta No.10 de la Encuesta

Cuadro 10		
OPCIÓN	FRECUENCIA	PORCENTAJE
SI	65	100,00%
NO	0	0,00%
TOTAL	65	100,00%
FUENTE : Estudiantes de Laboratorio de Sistemas Microprocesados		
ELABORACIÓN: Autor de la Investigación		

Tabla 1-10 Cuadro de tabulación de la Pregunta No. 10

ANALISIS: La totalidad de encuestados manifiestan que el dispositivo si debe contener un elemento que les facilite el ingreso de datos, ratificando la conveniencia del planteamiento del proyecto formulado.

11. ¿Cree que es necesario que el sistema permita cargar el programa de manera inalámbrica en las prácticas de Laboratorio?

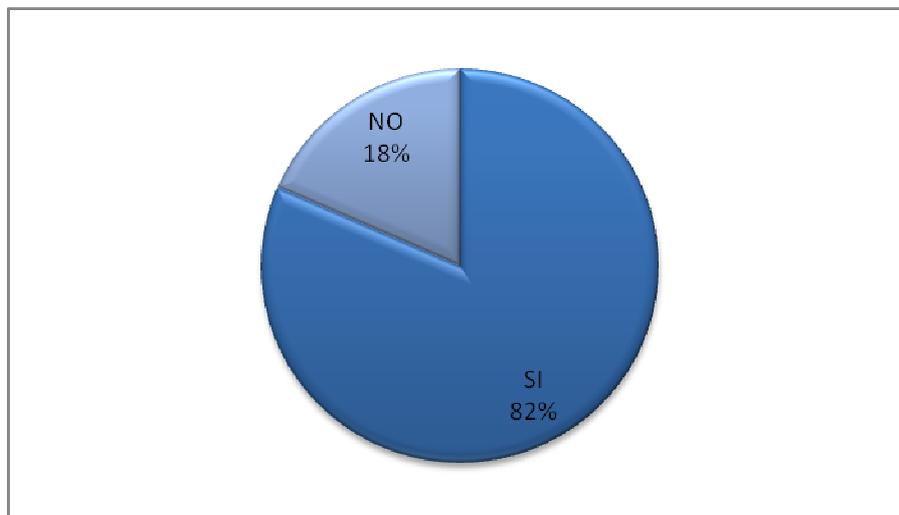


Fig. 1-11 Gráfica de la Tabulación a la Pregunta No.11 de la Encuesta

Cuadro 11		
OPCIÓN	FRECUENCIA	PORCENTAJE
SI	53	81,54%
NO	12	18,46%
TOTAL	65	100,00%
FUENTE: Estudiantes de Laboratorio de Sistemas Microprocesados		
ELABORACIÓN: Autor de la Investigación		

Tabla 1-11 Cuadro de tabulación de la Pregunta No. 11

ANALISIS: Se puede apreciar en el Cuadro No. 11 que el porcentaje tiene alto índice de aceptación, lo que indica que el estudiantado ve necesario la existencia de un mecanismo que les permita cargar el programa de manera inalámbrica, lo que optimiza el desarrollo del diseño del dispositivo, en el cual una de sus funciones es aquella que permite cargar el programa de la práctica mediante Bluetooth desde la PC o laptop hacia el dispositivo descrito.

12. ¿Le gustaría un software que facilite el envío del programa desde la PC hacia el dispositivo de desarrollo?

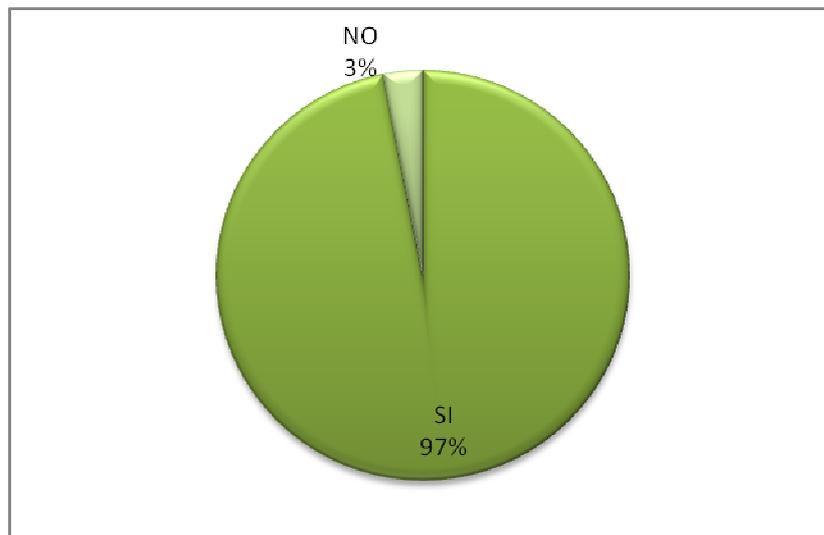


Fig. 1-12 Gráfica de la Tabulación a la Pregunta No.12 de la Encuesta

Cuadro 12		
OPCIÓN	FRECUENCIA	PORCENTAJE
SI	63	96,92%
NO	2	3,08%
TOTAL	65	100,00%
FUENTE : Estudiantes de Laboratorio de Sistemas Microprocesados		
ELABORACIÓN: Autor de la Investigación		

Tabla 1-12 Cuadro de tabulación de la Pregunta No. 12

ANALISIS: La respuesta de los encuestados es afirmativa en su mayoría en la implementación de un programa que permita cargar en el microcontrolador el programa de la práctica para lo cual es necesario de un ambiente gráfico en la PC o laptop para la ubicación y el envío.

13. ¿Cree que obtendría mayor beneficio en la implementación de las prácticas con este dispositivo?

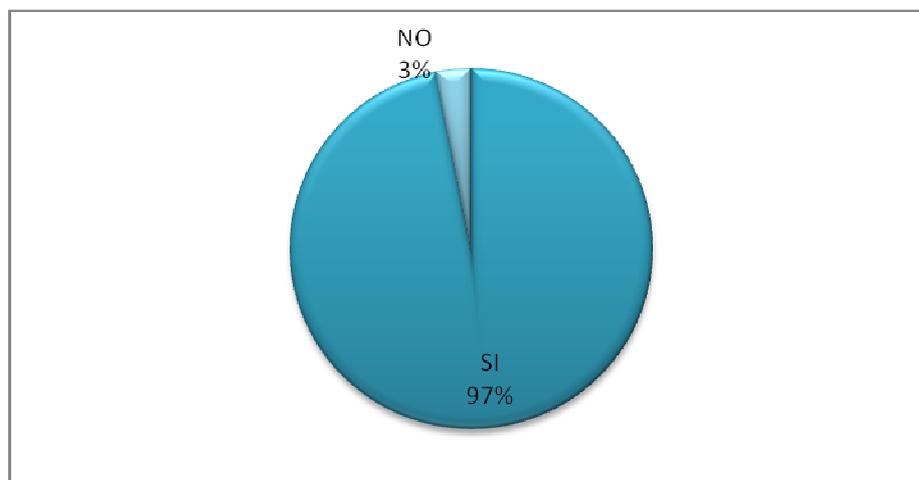


Fig. 1-13 Gráfica de la Tabulación a la Pregunta No.12 de la Encuesta

Cuadro 13		
OPCION	FRECUENCIA	PORCENTAJE
SI	63	96,92%
NO	2	3,08%
TOTAL	65	100,00%
FUENTE : Estudiantes de Laboratorio de Sistemas Microprocesados		
ELABORACIÓN: Autor de la Investigación		

Tabla 1-13 Cuadro de tabulación de la Pregunta No. 13

ANALISIS: Esta pregunta tiene una consideración de gran importancia para el proyecto toda vez que tiende a determinar el beneficio académico del estudiante en su formación tanto que el fin del proyecto es lograr un beneficio social para los estudiantes disminuyendo el egreso económico en la adquisición de elementos, lo que se demuestra en la contestación al cuestionamiento realizando.

1.3 EVALUACIÓN DE LA INFORMACIÓN

Luego del detalle de la forma en que se implementan las prácticas y la información obtenida que nace de las preguntas formuladas en la encuesta se logra determinar las fortalezas y debilidades de la actual implementación de las prácticas de laboratorio.

1.3.1 Fortalezas en la Implementación de las Prácticas

- Actualmente se tiene un grado de aceptación de referencia bueno en cuanto al desarrollo de las prácticas de laboratorio, lo que indica que el estudiante logra un conocimiento básico sobre la programación de los microcontroladores en el desarrollo de sistemas inteligentes simples⁷.
- A nivel de hardware el estudiante tiene un conocimiento del manejo de los elementos que conforman los sistemas como por ejemplo interconexión de: teclado, display, sensores, etc., para la construcción de las prácticas en el laboratorio.
- El lenguaje de programación utilizado es conveniente para la comprensión del funcionamiento del microcontrolador, para que en un futuro se les facilite el manejo de varios lenguajes de programación.
- Los estudiantes sí pueden realizar modificaciones al programa, con un discernimiento de resolver problemas o fallas básicas que ocurren en el software de la práctica.

1.3.2 Debilidades en la Implementación de las Prácticas

- Es común en el laboratorio que el estudiante tenga problemas con el hardware de laboratorio y que no puede encontrar las fallas debido a demasiados elementos y alambres que conforma el dispositivo, lo cual afecta el tiempo en la realización de la práctica en general.

⁷ Sistemas inteligentes simples es un programa que reúne características asimilables al de la inteligencia humana, este término se usa para sistemas incompletos.

- Los insuficientes programadores hace una debilidad importante para el desarrollo normal ya que sin este dispositivo no se puede comprobar las modificaciones en el software.
- Es común que se tenga problemas con fallas debido a corto circuitos en los elementos de los sistemas que se están construyendo, debido a fallas humanas o de fábrica, lo que implica una deficiencia en el desarrollo normal de la práctica.
- En las prácticas de laboratorio no se capacita al estudiante para conocer el funcionamiento, manejo y como se les acoplan a nivel de hardware y software la nuevas tecnologías que existen actualmente a los microcontroladores.

1.3.3 Soluciones que Implementa el Dispositivo del Proyecto

Debido a que la implementación de las prácticas en el laboratorio tiene varias falencias se realizó el diseño de un dispositivo (PDAT164P) que sirve para que el estudiante en el desarrollo de la práctica resuelva los problemas del hardware como: cortocircuitos de los elementos, la adquisición de elementos como son el microcontrolador, permitiendo que no sea necesario la necesidad de llevar al laboratorio ningún sistema que sirva de programador.

Con la creación de este dispositivo se logra que el estudiante desarrolle nuevas habilidades en la programación sobre los microcontroladores enfocándose a la parte de software de la práctica, sin que deje de lado el hardware, ya que es indispensable para el desarrollo de sistemas. Para lo cual se recomienda se realicen algunas prácticas en el uso de este dispositivo, debido a que el estudiante debe tener contactos con los elementos que va usar para construir la parte de hardware de la práctica.

En lo que se refiere al número de las prácticas y al contenido de las mismas, se recomienda se cree dos prácticas nuevas: Primera.- De comunicación inalámbrica ya que el dispositivo permite realizar esta aplicación. Segunda.- Salida de información a una GLCD, para lo cual se recomienda que estas prácticas se efectúen con otro lenguaje de programación como puede ser C o

Basic, debido a que el lenguaje utilizado en el laboratorio es Ensamblador el cual es complejo para realizar este tipo de aplicaciones descritas.

Teniendo en cuenta que hay un número insuficiente de programadores, el dispositivo permite resolver esta falencia; debido que permite cargar el programa de la práctica desde la computadora hacia el dispositivo de desarrollo de una forma directa e inalámbrica beneficiando al estudiante, en el entendido que no tendría que adquirir el programador, y en segundo lugar, la opción de realizar modificaciones del software. Lo que hace que el dispositivo se convierta en un sistema de aprendizaje.

Para el problema de insuficientes computadoras para los alumnos, el estudiante puede llevar sus laptops que tenga tecnología Bluetooth y conectarse al dispositivo para descargar las prácticas sin ninguna complejidad.

2. CAPÍTULO II: DISEÑO Y CONSTRUCCIÓN DEL DISPOSITIVO

En este capítulo se encuentra el diseño y construcción de PDAT164P, el cual se divide en dos partes: Software y Hardware, como también en la realización de un ambiente gráfico para la computadora para la descarga de programas de las prácticas.

En lo referente al diseño de Hardware en el capítulo anterior se describió las características principales del microcontrolador ATMEGA 164P, toda vez que en reunión de profesores de la materia de Sistemas Microprocesados se ha elegido como el dispositivo que se utilizará para el estudio de la materia.

Es pertinente decir que hay varios beneficios al utilizar este microcontrolador para el diseño, como es la capacidad de almacenamiento de la memoria Flash, la opción del gestor de arranque (Bootloader⁸), la capacidad de dos puertos seriales, un puerto serial para la descarga de las prácticas, y el otro libre para realización de las prácticas, fácil nivel de acoplamiento con los periféricos.

El diseño del PDAT164P está compuesto por los siguientes dispositivos: un Teclado 4x4, una LCD gráfico (GLCD), módulo de Bluetooth, una fuente de voltaje y un sistema central compuesto por el microcontrolador antes indicado (ver figura No. 2-1).

En lo concerniente al software, se describe las características que tiene el microcontrolador en la memoria Flash, principalmente en la parte de Bootloader donde va a residir el programa para auto programación del microcontrolador.

En el desarrollo del programa del Bootloader, se toma en cuenta las características de la memoria del microcontrolador, número de bytes a enviar (tamaño de página), control del envío, el tamaño de los programas de las

⁸ Bootloader es un programa sencillo que proporciona un soporte de auto programación al microcontrolador

prácticas y los parámetros de la comunicación, con todas estas características también se realiza una interfaz gráfica para la interacción entre el microcontrolador y la computadora para la descarga del programa de la práctica.

2.1 DISEÑO DEL HARDWARE

En el diseño del Hardware se estudian los diferentes elementos que conforman el dispositivo de desarrollo. En figura No. 2-1 se puede observar el diagrama de bloques del PDAT164P. A continuación se detallan las partes que conforman el PDAT164P de desarrollo:

- Microcontrolador Atmega164P
- LCD Gráfico
- Teclado 4x4
- Modulo de Bluetooth
- Fuente de alimentación

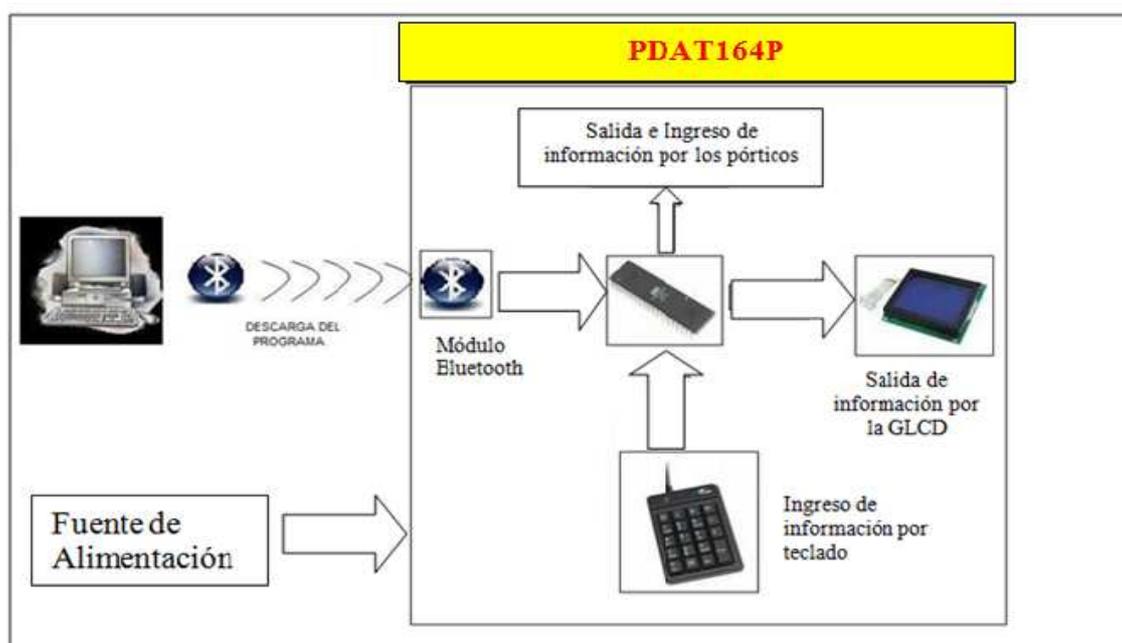


Fig. 2-1 Diagrama de Bloques de PDAT164P.

2.1.1 VISUALIZACIÓN (LCD GRÁFICO)

Para la visualización de los resultados de las prácticas se utiliza un LCD gráfico de 128x64 de marca JHD12864E.

La LCD gráfica JHD12864E, permite interactuar con el microcontrolador, de manera eficaz, debido a las ocho líneas de datos, las mismas que facilitan el manejo de un solo puerto del microcontrolador, para el intercambio de información; entre sus principales características están:

- 128x64 puntos
- Modo de visualización (Azul y Amarrillo)
- Visualización de la dirección o columna de la GLCD en 6 tiempos de reloj.
- 8 líneas de datos paralelos

En el gráfico No. 2-2 se describe el diagrama de bloques de la LCD gráfica JHD12864E.

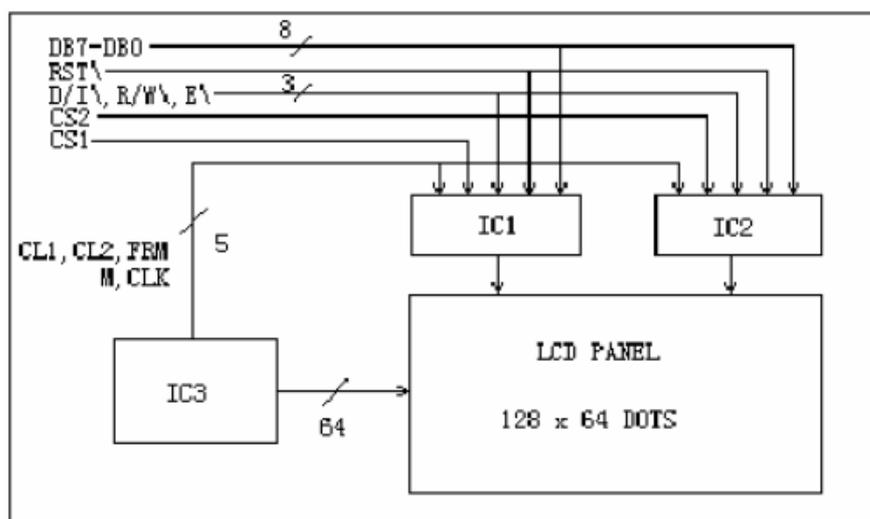


Fig. 2-2 Diagrama de bloques de la LDC gráfica JHD12864E

Como se puede observar en la figura No. 2-2 los bloques IC1 y IC2 manejan la activación de las columnas, el primero la mitad izquierda y el otro la mitad derecha respectivamente, en cambio el bloque IC3 maneja la activación de las filas, la activación de las líneas como columnas se las realizan a nivel de software con las señales de control.

La tabla No. 2-1 muestra la descripción de terminales del GLCD, la cual contiene el símbolo, descripción, funciones de cada uno de las terminales del GLCD para la interconexión con el microcontrolador.

PIN NO.	SÍMBOLO	DESCRIPCIÓN	FUNCIÓN
1	VSS	TIERRA	0V (GND)
2	VDD	FUENTE DE ALIMENTACIÓN PARA CIRCUITO LÓGICO	+5V
3	V0	AJUSTE DEL CONTRASTE DE LA LCD	
4	RS	REGISTRO DE SELECCIÓN DE INSTRUCCIONES O DATOS	RS = 0 ACTIVA LA INTRODUCCIÓN DE INSTRUCCIONES A LA GLCD. RS = 1 ACTIVA LA INTRODUCCIÓN DE DATOS A LA GLCD.
5	RW	SELECCIÓN DE LECTURA O ESCRITURA	RW = 0 : REGISTRO DE ESCRITURA RW = 1 : REGISTRO DE LECTURA
6	E	HABILITAR LA SEÑAL	
7	DB0	LÍNEAS DE DATOS DE ENTRADA O SALIDA	8 BIT: DB0-DB7
8	DB1		
9	DB2		
10	DB3		
11	DB4		
12	DB5		
13	DB6		
14	DB7		
15	CS1	SELECCIÓN DE CHIP	CS1=1,CHIP SELECCIONA LA SEÑAL POR EL IC1
16	CS2	SELECCIÓN DE CHIP	CS2=1,CHIP SELECCIONA LA SEÑAL POR IC2
17	RST	SEÑALDE RESET	RSTB=0 DISPLAY APAGADO, DISPLAY COMIENZA DESDE LA LINEA CERO
18	VEE	TENSIÓN NEGATIVA CON LA CONDUCCIÓN DE LCD	-10V
19	LED+	TENSIÓN DE ALIMENTACIÓN PARA LED+	+5V
20	LED-	TENSIÓN DE ALIMENTACIÓN PARA LED-	0V

Tabla 2-1 Descripción de los terminales del GLCD

Las líneas de datos del GLCD (DB0-DB7) se conectan directamente al p rtico C del microcontrolador y las l neas de control RS, RW, E, CS1 y CS2 se conectan al p rtico D a los terminales 5, 6, 7, 4, 3 respectivamente, la l nea RST del GLCD va al reset del microcontrolador, VEE y V0 va al potenc metro para el ajuste de contraste, el resto de l neas son para alimentaci n del GLCD, por lo cual van a VCC y GND de acuerdo a la tabla No. 2-1. En la figura No. 2-3 se puede ver la interconexi n de los elementos.

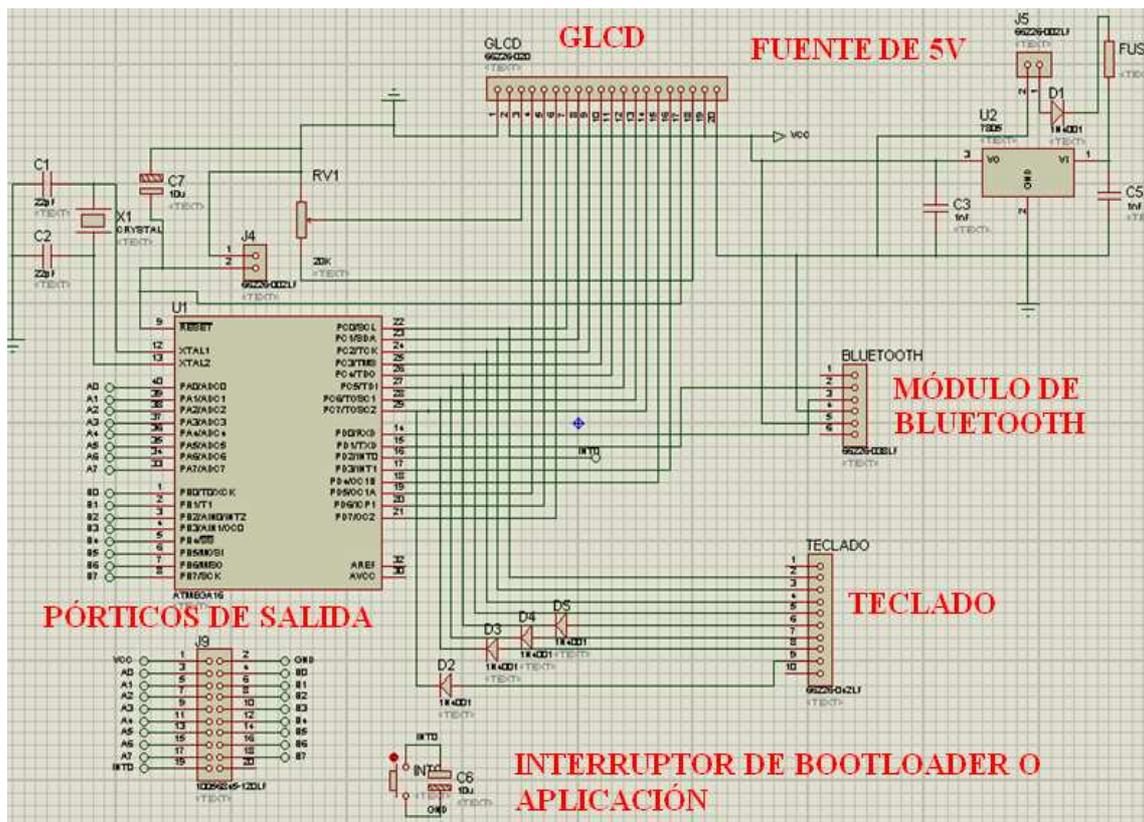


Fig. 2-3 Diagrama de interconexi n de los elementos que conforma el PDAT164P.

2.1.2 TECLADO 4X4

El teclado es un elemento pasivo, que sirve para la introducci n de informaci n al microcontrolador; en el dise o se maneja el mismo p rtico que utilizamos para la conexi n de las l neas de datos del GLCD, el p rtico C, para ahorrar recursos de microcontrolador y tener libre los p rticos A y B para conexiones a otros perif ricos de acuerdo a las necesidades de las pr cticas; para protecci n

del GLCD se conecta 4 diodos IN4007 a las líneas del barrido del teclado, esto es en las terminales 4, 5, 6 y 7 del pórtico C para evitar cortocircuitos por la pulsación de dos o más teclas a la vez.

Para el manejo del GLCD o el teclado este es mediante el programa, activando o desactivando las señales de control de la GLCD de acuerdo a la utilización de los mismos, esto se explicará en el capítulo 3 en la realización de las prácticas. (Ver Figura 2-3).

2.1.3 MÓDULO DE BLUETOOTH

La tecnología inalámbrica Bluetooth se realiza por medio de ondas de radio de corto alcance que trabaja a una frecuencia de 2.4 GHz, cuyo objetivo es simplificar las comunicaciones entre dispositivos electrónicos, como teléfonos móviles, computadores, entre otros dispositivos que poseen este tipo de tecnología.

Para su aplicación en el presente diseño se utiliza el módulo de comunicación de Bluetooth RN-41, este módulo permite el envío y recepción de datos desde el microcontrolador a la PC.

2.1.3.1 Descripción del Módulo RN-41

El módulo RN-41 (ver figura No. 2-4) es de dimensiones pequeñas, de baja potencia, que añade la capacidad inalámbrica, trabaja con múltiples protocolos de interfaz, lo que permite un fácil diseño para solucionar la comunicación en Bluetooth entre dispositivos electrónicos.

El módulo tiene un alto rendimiento debido a que posee EDR⁹ (Enhanced Data Rate) para mejorar la transmisión de datos a una tasa de transferencia de hasta 3 Mbps en distancias de 100 metros. Tiene la capacidad de almacenamiento de

⁹ EDR (Enhanced Data Rate) Le permite mejorar las velocidades de transmisión en hasta 3Mbps a la vez que intenta solucionar algunos errores de la especificación 1.2

un máximo de 8Mbits de memoria flash. A continuación se describen las principales características del módulo Bluetooth RN-41:

- Módulo de Bluetooth Clase I
- Bluetooth v2.0 + EDR de apoyo
- Bajo consumo de energía: Promedio de 30 mA
- UART¹⁰ (SPP¹¹ o HCI¹²) y USB (sólo HCI).
- En SPP tiene velocidades de datos: 240 kbps (esclavo) y 300kbps (maestro)
- Memoria Flash de 8Mb
- Voltaje de funcionamiento: 4.5 – 6 V
- Antena RF integrada en el Chip con amplificador de alta potencia
- Dimensiones 13.4mm x 25.8 mm x2mm
- Puertos UART : TX, RX, RTS, CTS para SPP



Fig. 2-4 Módulo Bluetooth RN-41¹³

El módulo Bluetooth RN-41 está compuesto de 6 pines que se describen en la Tabla No. 2-2

No. de PIN	Descripción
1	RTS
2	RX
3	TX
4	GND
5	VCC
6	CTS

Tabla 2-2 Descripción de los Pines de Módulo Bluetooth RN-41

¹⁰UART (Universal Asynchronous Receiver-Transmitter) Transmisor y Receptor Asíncrono Universal. Éste controla los puertos y dispositivos serie.

¹¹ SPP (Serial Port Profile) Emula una línea serie y provee una interfaz de reemplazo de comunicaciones basadas en RS-232, con las señales de control típicas.

¹² HCI (Host Controller Interface) Proporciona una interfaz de órdenes para el controlador en banda base y para el gestor de enlace, y permite acceder al estado del hardware y a los registros de control.

¹³ Roving Networks Wireless for Less, "hoja de datos de RN-41", DS-RN41-V3.1, 2009

2.1.3.2 Mini USB-Bluetooth

Este módulo (ver figura No. 2-5) es de simple operación, ya que descubre y despliega en pantalla del computador otros dispositivos Bluetooth que están dentro del alcance como se puede ver en la figura No 2-6.



Fig. 2-5 Mini USB Bluetooth 2.0

Provee una interfaz amigable, proporcionando un camino simple para conectarse con otros dispositivos Bluetooth, en este caso para conectarse con el módulo Bluetooth RN-41, para utilizar en su totalidad los servicios disponibles.

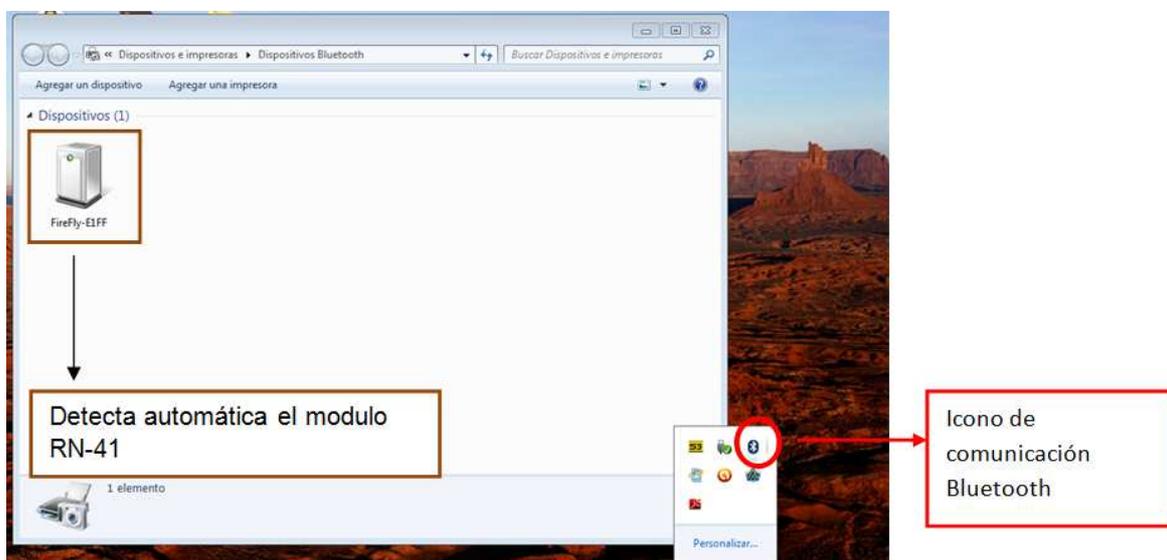


Fig. 2-6 Muestra la detección automática del módulo RN-41

Características del mini USB-Bluetooth 2.0:

- Establece una conexión inalámbrica entre el computador y los dispositivos Bluetooth.

- Sincronización de Datos.
- Conferencias en red.
- Crear Redes Locales (LAN).
- Intercambiar información en forma segura y confidencial.

2.1.3.3 Comunicación entre la PC y el dispositivo de desarrollo

Para la comunicación entre la PC y el PDAT164P se utiliza un adaptador mini USB-Bluetooth que se conecta a un puerto USB del computador, este adaptador transmitirá vía Bluetooth los datos hacia el módulo Bluetooth RN-41 para que este transforme la información a señales RS-232 para la descarga del programa hacia el microcontrolador ATMEGA 164P, el cual procesa la información. En el la figura No. 2-7 se muestra el proceso por el que tiene que pasar la señal para llegar al dispositivo y viceversa.

El módulo RN-41 provee el servicio de puerto serie de envío y recepción de datos RS-232, los cuales se conectan a los pines de RX y TX del microcontrolador respectivamente, para lo cual se configura las velocidades de transferencia en los dos módulos y en el microcontrolador para que trabajen adecuadamente y no haya errores en la comunicación, para el diseño se escogió una trasmisión asincrónica con los siguientes parámetros:

- Velocidad de transmisión: 9,600 Kbps
- Sin paridad
- Un bit de parada
- 8 bits de información

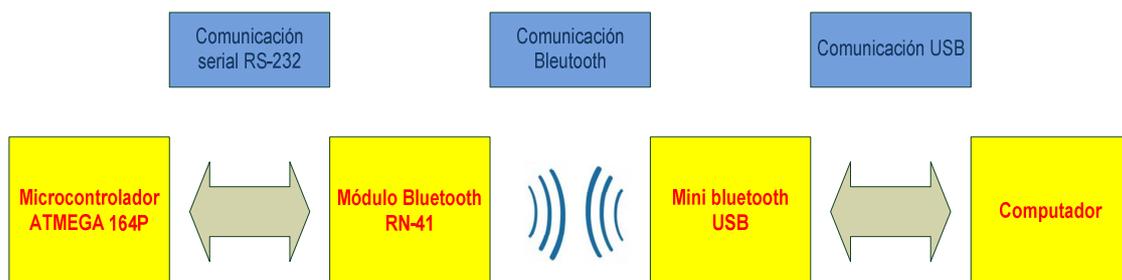


Fig. 2-7 Diagrama de Bloque de la comunicación

Para que no existan errores en la comunicación se utiliza un cristal externo de 11.059200 MHz conectado a las terminales del cristal del microcontrolador y a una velocidad de 9600 bps ya que se tiene una tasa de error de 0,0% como se puede observar en la Tabla No 2-3.

En la tabla también muestra parámetros como UBRR que para configurar la velocidad de transmisión en el microcontrolador y el U2X que sirve para seleccionar si se desea doble velocidad, estos dos parámetros se explicarán en el diseño del software.

Baud Rate (bps)	$f_{osc} = 8.0000 \text{ MHz}$				$f_{osc} = 11.0592 \text{ MHz}$				$f_{osc} = 14.7456 \text{ MHz}$			
	U2X = 0		U2X = 1		U2X = 0		U2X = 1		U2X = 0		U2X = 1	
	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error
2400	207	0.2%	416	-0.1%	287	0.0%	575	0.0%	383	0.0%	767	0.0%
4800	103	0.2%	207	0.2%	143	0.0%	287	0.0%	191	0.0%	383	0.0%
9600	51	0.2%	103	0.2%	71	0.0%	143	0.0%	95	0.0%	191	0.0%
14.4k	34	-0.8%	68	0.6%	47	0.0%	95	0.0%	63	0.0%	127	0.0%
19.2k	25	0.2%	51	0.2%	35	0.0%	71	0.0%	47	0.0%	95	0.0%
28.8k	16	2.1%	34	-0.8%	23	0.0%	47	0.0%	31	0.0%	63	0.0%
38.4k	12	0.2%	25	0.2%	17	0.0%	35	0.0%	23	0.0%	47	0.0%
57.6k	8	-3.5%	16	2.1%	11	0.0%	23	0.0%	15	0.0%	31	0.0%
76.8k	6	-7.0%	12	0.2%	8	0.0%	17	0.0%	11	0.0%	23	0.0%
115.2k	3	8.5%	8	-3.5%	5	0.0%	11	0.0%	7	0.0%	15	0.0%
230.4k	1	8.5%	3	8.5%	2	0.0%	5	0.0%	3	0.0%	7	0.0%
250k	1	0.0%	3	0.0%	2	-7.8%	5	-7.8%	3	-7.8%	6	5.3%
0.5M	0	0.0%	1	0.0%	-	-	2	-7.8%	1	-7.8%	3	-7.8%
1M	-	-	0	0.0%	-	-	-	-	0	-7.8%	1	-7.8%
Max ⁽¹⁾	0.5 Mbps		1 Mbps		691.2 kbps		1.3824 Mbps		921.6 kbps		1.8432 Mbps	

Tabla 2-3 Ejemplos de valores de UBRR para un frecuencia de oscilador

La conexión del cristal con el microcontrolador se muestra en la figura No. 2-8, donde se puede observar que van conectados a los pines XTAL2 y XTAL2 del microcontrolador.

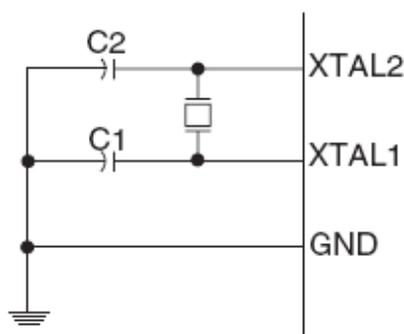


Fig. 2-8 Conexión del Oscilador de Cristal

La tabla No.2-4 muestra los valores de los capacitores de acuerdo a la configuración de los fusibles de la frecuencia de oscilador del microcontrolador.

El dispositivo trabaja a una frecuencia de 11.059200 MHz lo que la tabla da valores entre 12- 22 pF, por lo tanto se ha seleccionado capacitores cerámicos de 22 pF.

CKOPT	CKSEL3..1	Rango de Frecuencia [MHz]	Rango recomendado para capacitores C1 y C2 para uso con Cristal (pF)
1	101	0.4 - 0.9	-
1	110	0.9 – 3.0	12 - 22
1	111	3.0 – 8.0	12 – 22
0	101, 110, 111	$1.0 \leq$	12 - 22

Tabla 2-4 Modo de operación del Oscilador de Cristal

2.1.4 FUENTE DE ALIMENTACIÓN

El PDAT164P requiere ser alimentado con 5V. Por tal razón se va a utilizar una batería que entrega 12V, estos deben ser regulados a 5V, para lo cual utiliza un regulador de voltaje fijo LM7805 con un capacitor de entrada y uno de salida para el rizado. La Figura No. 2-9 muestra el diagrama de la fuente de alimentación.

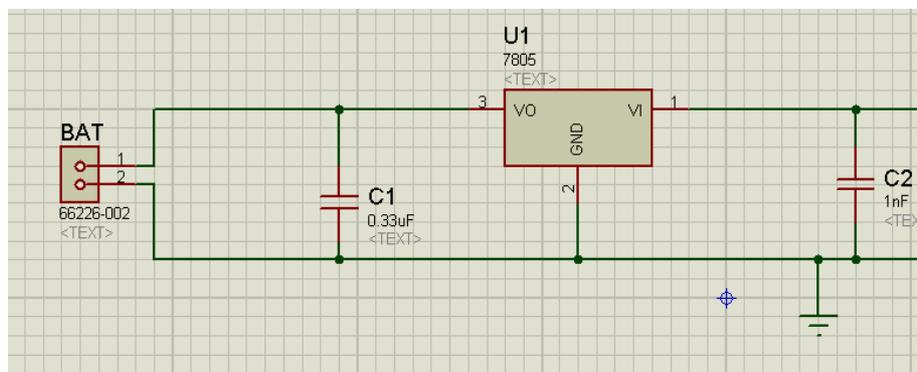


Fig. 2-9 Diagrama del circuito de la fuente de alimentación.

2.1.5 CONSTRUCCIÓN DE LA PLACA DEL PDAT164P

Para la construcción del PDAT164P se realiza las placas para el dispositivo como para la fuente de alimentación, con la ayuda del software Proteus 7.6 que a continuación se explica.

2.1.5.1 SOFTWARE PROTEUS 7.6

Proteus es un paquete de programas de diseño y simulación electrónica, desarrollado por Labcenter Electronics que consta de los dos programas principales: Ares e Isis, y los módulos VSM y Electra

2.1.5.1.1 Isis y Ares

Con ayuda del Isis se diseñó el plano eléctrico del circuito del dispositivo de desarrollo con los componentes que lo conforman: el microprocesador, GLCD, teclado 4x4, módulo de Bluetooth, cristal, capacitores, potenciómetro y conectores, como se puede observar en la figura No. 2-3.

Una vez diseñado el dispositivo se utilizó la herramienta ARES para el enrutamiento, ubicación y edición de componentes del PDAT164P (ver figura No. 2-10), para la fabricación del mismo.

En la figura No. 2-11 se puede observar el circuito impreso del dispositivo que contiene las pistas.

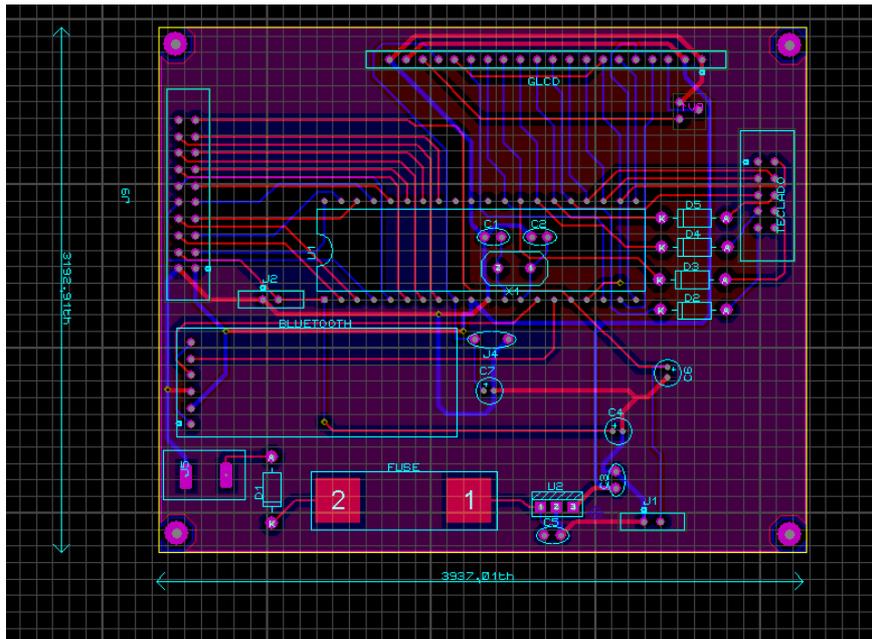


Fig. 2-10 Enrutamiento, ubicación y edición de componentes del PDAT164P

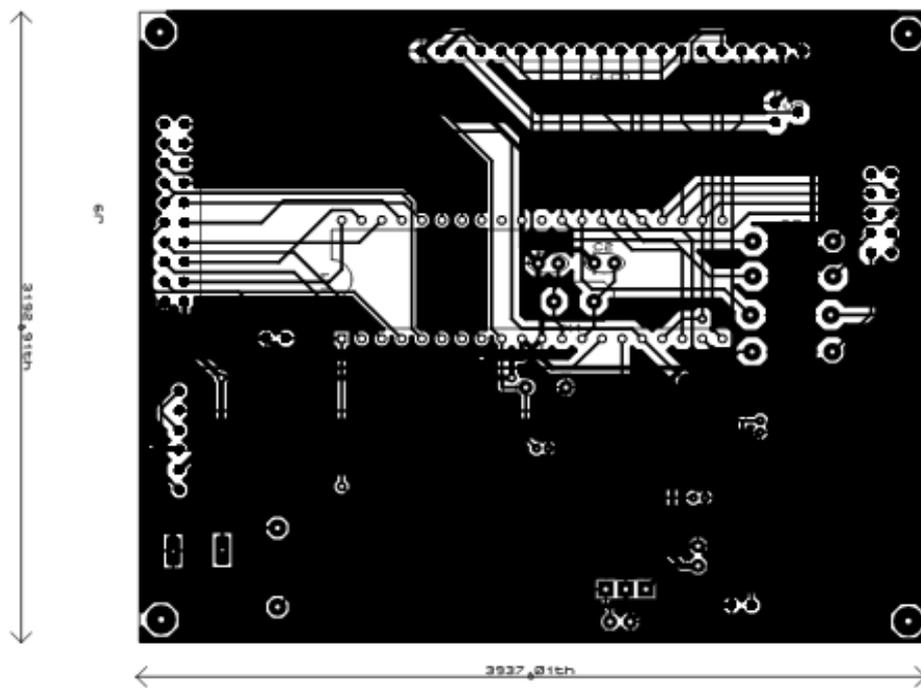


Fig. 2-11 Pistas del dispositivo realizadas con ARES

Una vez realizado el diagrama circuital del PDAT164P, se transfiere a una placa de fibra de vidrio a doble lado, las cuales deben coincidir para que no haya errores para realizar los huecos, y finalmente soldar los elementos que compone el dispositivo. En la Figura No.2-12 se puede observar la placa del PDAT164P con los elementos que conforma ya montados en la misma.

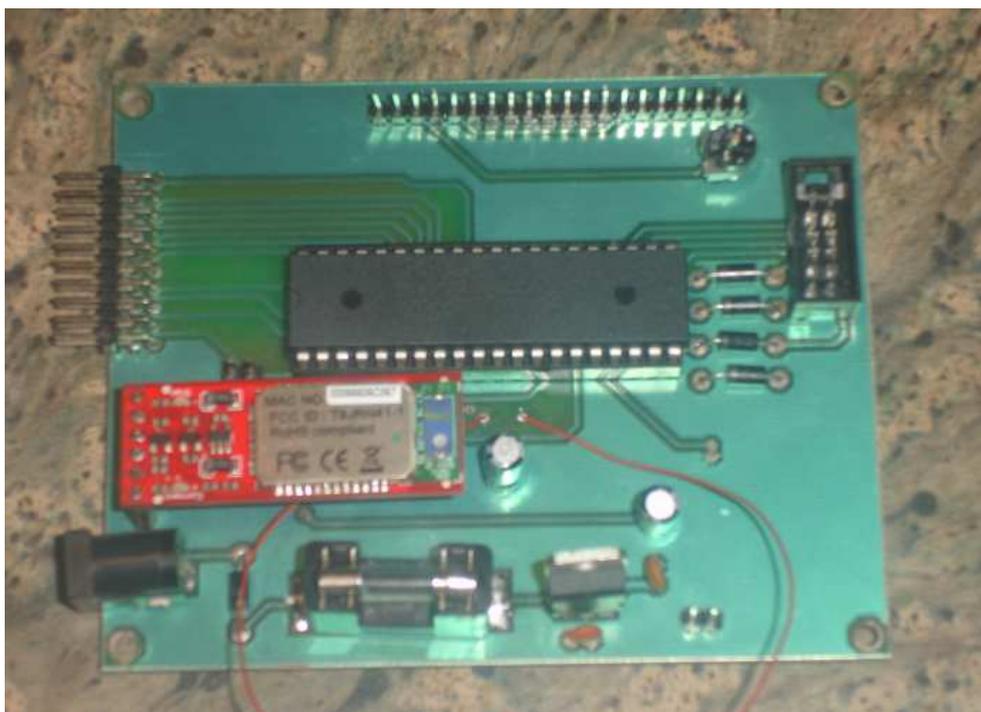


Fig. 2-12 Placa del PDAT164P

Se necesita una caja contenedora para el PDAT164P, por lo que se utiliza una caja de acrílico de medidas 19x10x6 cm, y a partir de estas medidas se elabora el PCB¹⁴ o placa de circuito. En la figura No. 2-13 se observa el circuito impreso con todos sus elementos ya montados en la caja (ver figura No. 2-13).

En la parte delantera de la caja se monta la GLCD y el teclado 4x4 (ver figura No. 2-14) y en la parte lateral se encuentran las salidas de los pórtilo A y B del microcontrolador y el conector de la fuente de 5V (ver figura No. 2-15).

¹⁴ PCB (*printed circuit board*) es un medio para sostener mecánicamente y conectar eléctricamente componentes electrónicos, a través de rutas o pistas de material conductor, grabados en hojas de cobre laminadas sobre un sustrato no conductor, comúnmente baquelita o fibra de vidrio.

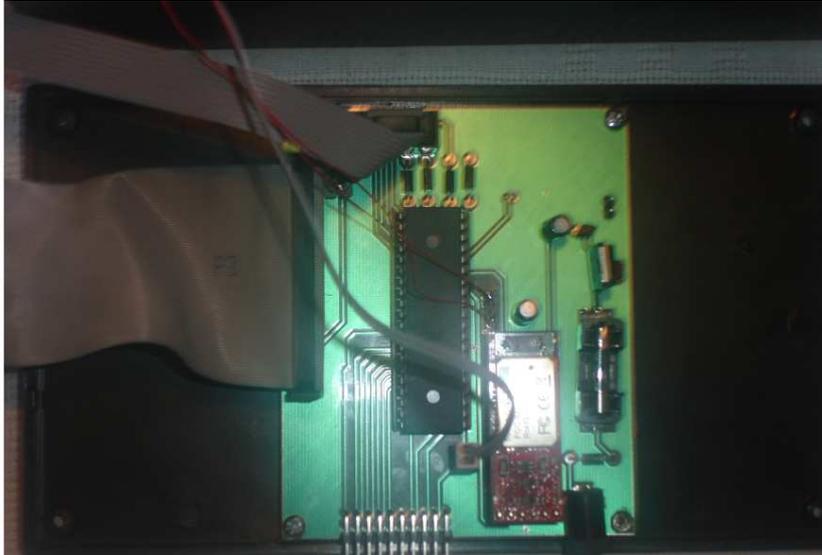


Fig. 2-13 Montaje del PCB del PDAT164P en la caja



Fig. 2-14 Montaje de la GLCD y el teclado 4x4 en la parte delantera de la caja



Fig. 2-15 Salidas de los pórticos A y B del microcontrolador y el conector de la fuente de voltaje

En la figura No. 2-16 se muestra el PDAT164P ya construido y montado en la caja acrílica con todos los elementos que constituye el dispositivo.



Fig. 2-16 PDAT164P

2.2 DISEÑO DEL SOFTWARE

Para el diseño del software se describen las características de la memoria Flash del microcontrolador ATMEGA164P enfocándose en la parte del Bootloader.

2.2.1 BOOTLOADER

Bootloader o Gestor de arranque es un programa sencillo que proporciona un soporte de auto programación. Esto permite realizar aplicaciones flexibles, con actualización del microcontrolador usando el código que reside en la parte de la memoria Flash llamada sección Bootloader.

El programa Bootloader puede ser usado con cualquier interfaz y tecnología disponible, leyendo el código y actualizando la parte de la memoria llamada sección Aplicación.

El código del programa dentro de la sección del Bootloader tiene la capacidad de escribir enteramente en la Flash, incluyendo la parte de la memoria donde reside el Bootloader, modificándolo y borrándolo si es necesario.

Los bits Boot Lock¹⁵ tiene dos configuraciones para seleccionar diferentes niveles de protección de la memoria, estos bits puede ser configurados independientemente.

2.2.1.1 Secciones Aplicación y Bootloader de la memoria Flash.

La memoria Flash está organizada en dos secciones; la sección Aplicación y la del Bootloader (ver figura No. 2-17). El tamaño de las diferentes secciones se configura por el fusible BOOTSZ¹⁶ como se muestra en la Tabla No. 2-5. Estas dos secciones pueden tener diferentes niveles de protección.

¹⁵ Boot lock son bits para seleccionar el modo de protección de la memoria Flash

¹⁶ BOOTSZ son bits para seleccionar el tamaño de las memorias Bootloader y Aplicación

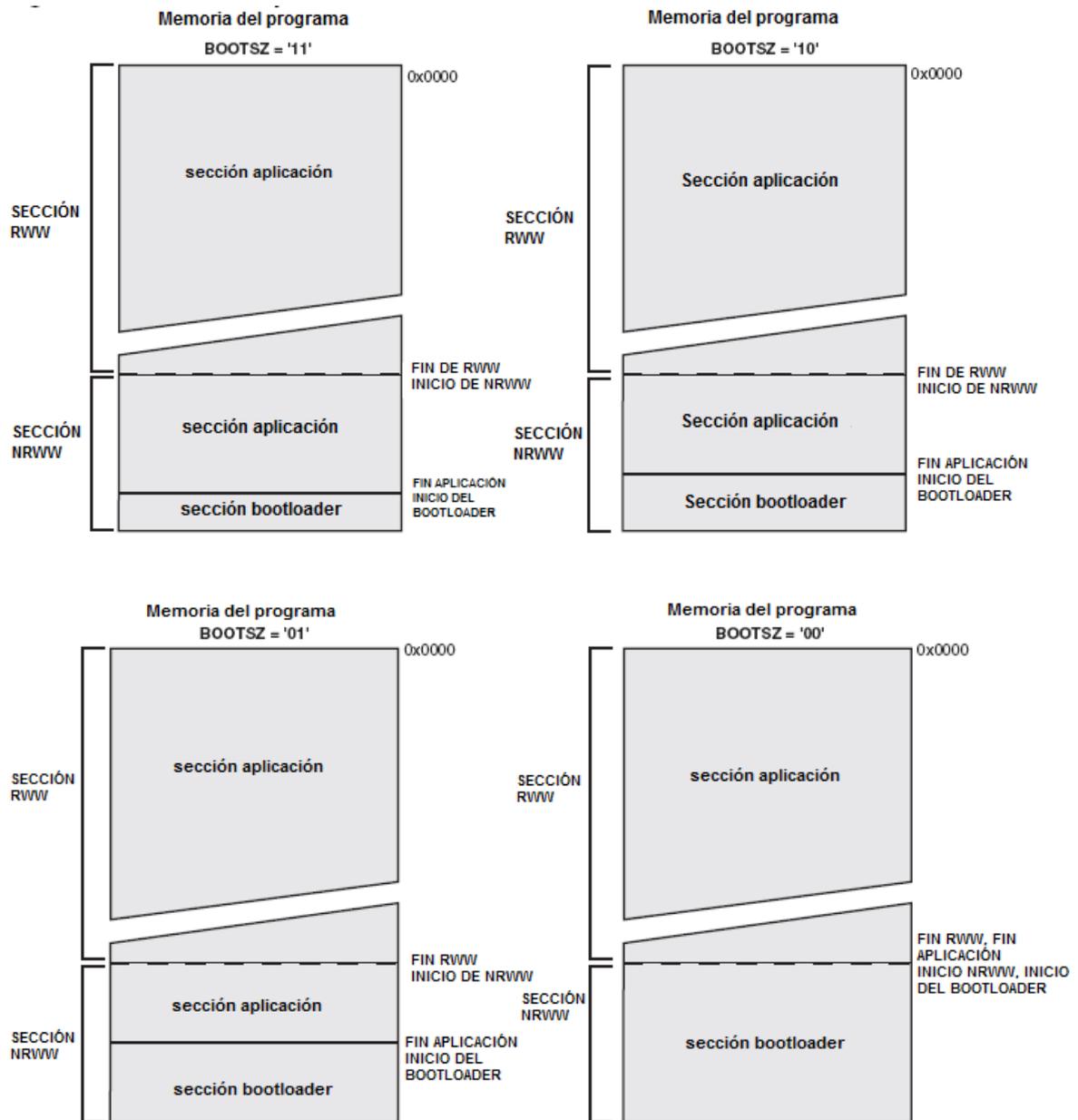


Fig. 2-17 Secciones de Memoria del Microcontrolador Atmega164P¹⁷

BOOTS Z1	BOOTS Z0	TAM. BOOT	PAG.	SECCIÓN APLICACIÓN	SECCIÓN BOOTLOAD ER	FIN DE LA SECCIÓN APLICACI ÓN	INICIO DE LA SECCIÓN BOOTLOAD ER
1	1	128 Palabras	2	0x0000 - 0x1F7F	0x1F80 - 0x1FFF	0x1F7F	0x1F80
1	0	256 Palabras	4	0x0000 - 0x1EFF	0x1F00 - 0x1FFF	0x1EFF	0x1F00

¹⁷ ATMEL, "Hoja de datos del Atmega 164P", 2009, pág. 282

0	1	512 Palabras	8	0x0000 - 0x1DFF	0x1E00 - 0x1FFF	0x1DFF	0x1E00
0	0	1024 Palabras	16	0x0000 - 0x1BFF	0x1C00 - 0x1FFF	0x1BFF	0x1C00

Tabla 2-5 Configuración del Tamaño del Boot

2.2.1.1.1 Sección Aplicación

La sección Aplicación es la parte de la memoria Flash que se usa para guardar el código del programa. La protección para esta sección puede ser seleccionado por la utilización de los bits Boot Lock, (ver la tabla No 2-5).

Esta sección no puede almacenar el código de la aplicación si está deshabilitada la instrucción SPM¹⁸, por lo cual se debe activar antes de guardar el código del programa.

Modo BLB1	BLB12	BLB11	Protección
1	1	1	Sin restricción en las instrucciones SPM y LPM ¹⁹ , para el acceso a la sección Aplicación.
2	1	0	SPM no está permitida para escribir en la sección Aplicación.
3	0	0	Igual a los modos 2 y 4
4	0	1	LMP ejecutada en la sección Bootloader, no permite leer desde la sección de Aplicación. Si un vector está apuntando a la sección del Bootloader, la interrupción está deshabilitada cuando se ejecuta desde la sección Aplicación.

Tabla 2-6 Modos de protección de la sección de Aplicación

2.2.1.1.2 Sección Bootloader (BLS)

A diferencia de la sección Aplicación que es donde se almacena el código del programa, la sección del Bootloader contiene el software que permite guardar y actualizar el programa de la aplicación, para poder guardar este código se

¹⁸ SPM (Store Program Memory) instrucción para almacenar en la memoria del programa

¹⁹ LPM (Load Program Memory) instrucción para leer el código del programa almacenado en la memoria

activa la instrucción SPM, la cual permite acceder a esta parte de la memoria Flash. La de protección para la sección Bootloader puede ser seleccionado por los bits Boot Lock. (Ver tabla No 2-6)

Modo BLB0	BLB02	BLB01	Protección
1	1	1	Sin restricción en las instrucciones SPM y LPM para el acceso a la sección Bootloader.
2	1	0	SPM no está permitida para escribir en la sección Bootloader.
3	0	0	Igual a los modos 2 y 4
4	0	1	LMP ejecutada en la sección Aplicación, no permite leer desde la sección del Bootloader, si un vector está apuntando a la sección Aplicación, la interrupción está deshabilitada cuando se ejecuta desde la sección Bootloader.

Tabla 2-7 Modos de protección de la sección Bootloader

A continuación se detallan las formas de almacenar el código del programa en la memoria Flash del microcontrolador utilizando el programa Bootloader.

2.2.1.2 Lectura Mientras se Escribe y No Lectura Mientras se Escribe en las secciones de la memoria Flash

El microcontrolador soporta dos maneras de cargar el código del programa en la memoria, lee mientras se realiza la escritura, y la otra, el microcontrolador se detiene durante la actualización, en cualquiera de las opciones la actualización depende del número de la página de la memoria donde se quiere colocar el código.

El límite entre RWW y NRWW se puede observar en la figura No. 2-18. La diferencia entre las dos secciones es:

- Cuando se utiliza RWW para borrar o escribir en la localidad de memoria del microcontrolador, la aplicación sigue en ejecución.

- En cambio cuando se utiliza NRWW para borrar o escribir dentro de la Flash, el microcontrolador se detiene por completo durante la operación.

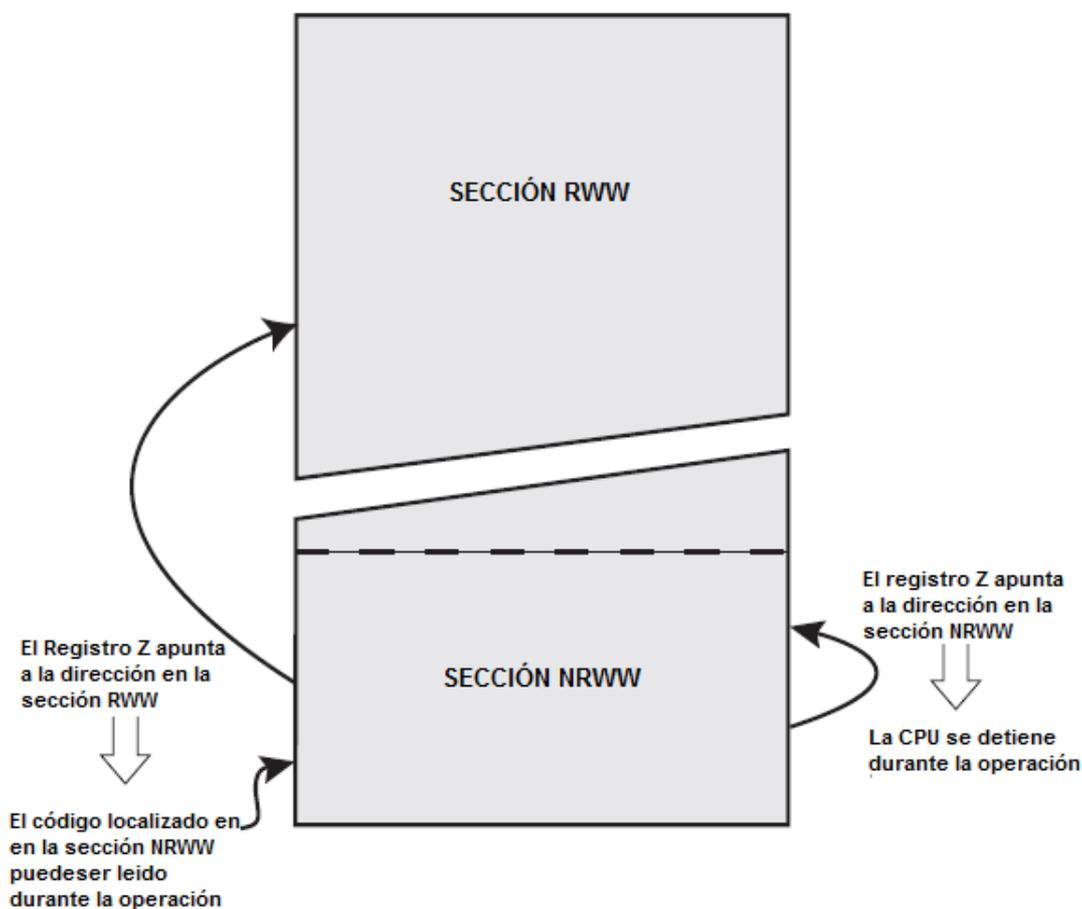


Fig. 2-18 Secciones RWW vs. NRWW²⁰

A continuación se describen las maneras para cargar al código del programa, lectura mientras se escribe (RWW) y no lectura mientras se escribe (NRWW).

2.2.1.2.1 Lectura Mientras Se Escribe RWW

Cuando se utiliza la sección RWW para borrar o escribir una página localizada en la misma sección, la sección NRWW puede ser leída.

Si la aplicación está localizada en la sección RWW, y se utilizan instrucciones para cargar en la memoria del programa, como ejemplo instrucciones de salto o

²⁰ ATMEL, "Hoja de datos del Atmega 164P", 2009, pág. 281.

interrupciones durante la programación, el software finaliza en un estado desconocido. Para evitar esto, las interrupciones son deshabilitadas y movidas a la sección de Bootloader que está localizada en la sección NRWW.

2.2.1.2.2 Sección de No Lectura Mientras se Realiza la Escritura NRWW

La localidad del código que está ubicada en la sección NRWW, se lee cuando el software Bootloader actualiza la página en la sección RWW. Cuando el código del Bootloader actualiza la sección NRWW, el microcontrolador se detiene durante la operación de borrado de página o escritura de la misma.

2.2.1.3 Bits de Bloqueo del Bootloader

Si no se necesita el Bootloader para la actualización de programa, la Flash está disponible para códigos de aplicación.

Caso contrario el Bootloader tiene dos configuraciones separadas de bits de bloqueo del Bootloader con lo cual puede ser configurado independientemente, para dar una flexibilidad única al seleccionar los diferentes niveles de protección. Se puede seleccionar:

- Proteger la Flash para la actualización de software del microcontrolador.
- Proteger solo la sección Bootloader para la actualización del software del microcontrolador.
- Proteger solo la sección de Aplicación de la Flash para la actualización del software del microcontrolador.
- Permitir la actualización del software en la toda la Flash.

2.2.1.4 Organización de la Flash

La memoria FLASH está organizada por páginas; para el ATmega164P son:

- Tamaño de la FLASH: 16 Kilo bytes = 8 Kilo palabras
- Tamaño de la página: 64 palabras = 128 bytes
- Palabra dentro de la página: PC[5:0] = Z6:Z1
- Número de páginas: 128
- Página: PC[12:6] = Z13:Z7
- Bit más significativo del PC: 12

PÁG.	PC		Z	
	INICIO	FINAL	INICIO	FINAL
0	0x0000	0x003F	0x0000	0x007F
1	0x0040	0x007F	0x0080	0x00FF
2	0x0080	0x00BF	0x0100	0x017F
3	0x00C0	0x00FF	0x0180	0x01FF
.
.
.
124	0x1F00	0x1F3F	0x3E00	0x3E7F
125	0x1F40	0x1F7F	0x3E80	0x3EFF
126	0x1F80	0x1FBF	0x3F00	0x3F7F
127	0x1FC0	0x1FFF	0x3F80	0x3FFF

Tabla 2-8 Organización de la memoria Flash del Atmega164p

En el registro Z los bits menos significantes, direccionan las palabras dentro de la página, mientras los bits más significantes direccionan las páginas. Esto se muestra en la Figura No. 2-19.

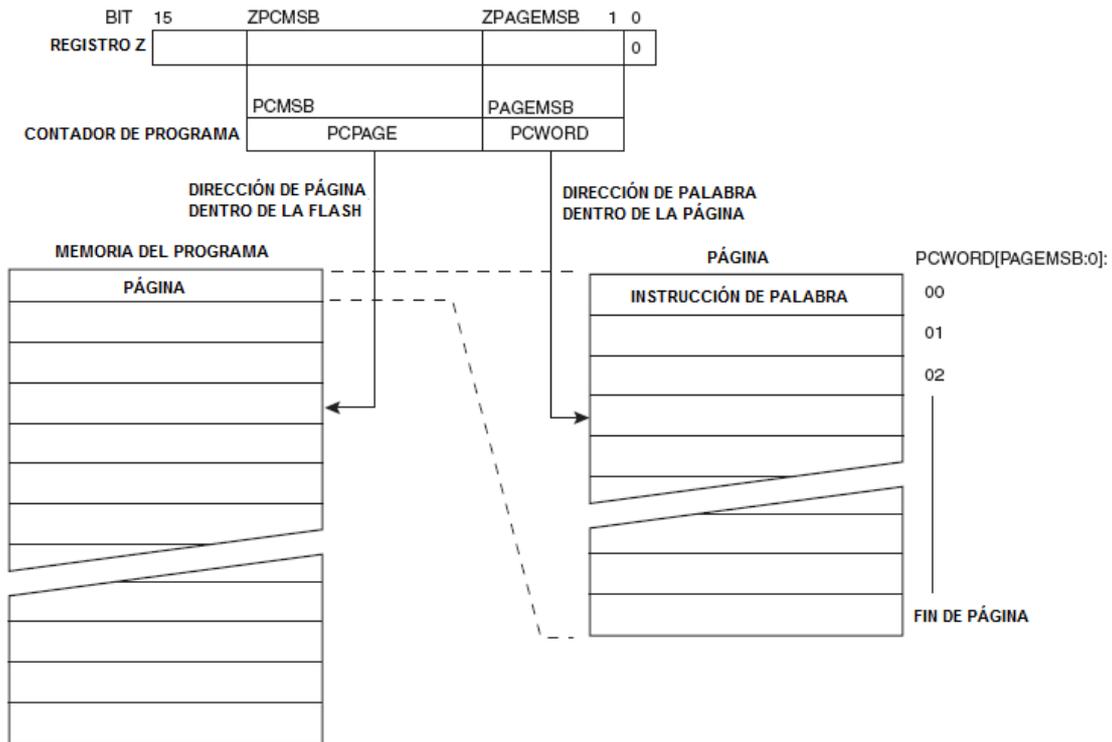


Fig. 2-19 Direccionamiento de la Flash durante SPM²¹

2.2.1.5 Auto programando la Flash

La memoria del Programa se actualiza por páginas, antes de programar con el dato almacenado en el buffer de la página temporal, la página debe ser borrada. El buffer de la página temporal se llena con una palabra al tiempo que se usa la instrucción SPM y el buffer puede ser llenado antes del comando de borrado de página. Hay varias opciones de llenar el buffer antes de borrado de página, que a continuación se detallan:

Alternativa 1

- Se Llena el buffer de la página temporal
- Se borra la página
- Se realiza la escritura de la página

Alternativa 2

- Se borra la página
- Se llena el buffer de la página
- Realiza la escritura de la página

²¹ ATMEL, "Hoja de datos del Atmega 164P", 2009, pág. 284.

Si solo una parte de las páginas necesitan ser cambiadas, el resto de las páginas se almacena en un buffer temporal antes de borrar, y luego se vuelve a escribir. Cuando se usa la alternativa 1, el Bootloader proporciona una lectura efectiva; lo que permite al usuario primero leer la página, hacer los cambios necesarios, luego escribir de nuevo el dato modificado. En cambio, en la alternativa 2 no es posible leer los datos antiguos mientras se carga desde la página.

2.2.2 DISEÑO DEL BOOTLOADER

2.2.3 Descripción del Registro SPMCSR

EL registro SPMCSR (Store Program Memory Control and Status Register) contiene los bits de control necesarios para las operaciones del BootLoader (ver figura No. 2-20).

Bit	7	6	5	4	3	2	1	0
0x37 (0x57)	SPMIE	RWWSB	SIGRD	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN
LECTURA/ESCRITURA	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
VALOR INICIAL	0	0	0	0	0	0	0	0

Fig. 2-20 Registro de estatus y control para almacenar en la memoria del programa

- **Bit 7 – SPMIE: Habilitación de Interrupciones SPM**

Cuando el bit SPMIE es uno lógico, y el bit I del registro del estado es uno, la interrupción estaría lista para ser habilitada, y la interrupción SPM sería ejecutada mientras el bit SPMEN es borrado.

- **Bit 6 – RWWSB: Lectura mientras la sección de escritura está ocupada**

Cuando la operación de auto programación (borrado de página o escritura de página) en la sección RWW se inicializan, RWWSB es uno lógico por hardware.

Cuando el bit RWWSB es uno, la sección RWW no puede ser accedida, después de la auto programación el bit RWWSB se borra y RWWSRE se escribe a uno. También hay la alternativa de que el bit RWWSB

automáticamente sea limpiado si la operación de cargar la página es inicializada.

- **Bit 5 – SIGRD: Lectura de la Fila de Firma**

Si en este bit se escribe uno lógico al mismo tiempo que en SPMEN, la siguiente instrucción LPM dentro de tres ciclos de reloj leerá un byte desde la fila de firma en el registro Z. La instrucción SPM dentro de cuatro ciclos después de que SIGRD y SPMEN son puestos en uno no tendría efecto.

Esta operación se reserva para usos futuros y no se usa.

- **Bit 4 – RWWSRE: Lectura-Mientras- Escribe habilita la sección de lectura**

Cuando se realiza la operación de auto programación (Borrado de Página o escritura de Página) en la sección RWW, la sección RWW se bloquea para la lectura (RWWSB se pone en uno lógico por hardware). Para rehabilitar la sección RWW, el software del usuario espera hasta que la programación sea completada (SPMEN se borre).

Entonces si el bit RWWSRE se escribe uno al mismo tiempo que SPMEN dentro de cuatro ciclos de reloj rehabilita la sección RWW. La sección RWW no podría ser rehabilitada mientras la flash está ocupada con el borrado de página o escritura de página (SPMEN es uno).

Si RWWSRE es escrito en uno lógico mientras la Flash está cargando, la operación de cargar la Flash abortaría y el dato cargado sería perdido.

- **Bit 3 – BLBSET: Poner en uno el Bit del Boot Lock Bit Set**

Si este bit se escribe uno al mismo tiempo que en el bit SPMEN, la siguiente instrucción SPM dentro de cuatro ciclos de reloj establece los bits Boot Lock, de acuerdo a los datos en R0; los datos en R1 y la dirección en el puntero Z son ignoradas debido a que se utilizó para configurar los bits de bloqueo.

El bit BLBSET automáticamente se borra por completo después de establecer los bits de bloqueo, o si no la instrucción SPM se ejecutada dentro de cuatro ciclos de reloj.

Cuando la instrucción LPM se ejecuta dentro de tres ciclos después de que BLBSET y SPMEN se ponen en uno lógico en el Registro SPMCSR, leerá los bits de bloqueo o los bits del Fusible los cuales se cargaran en el registro de destino, esto cuando Z sea 0x0001.

- **Bit 2 – PGWRT: Escritura de Página**

Si PGWRT se escribe uno al mismo tiempo que SPMEN, la siguiente instrucción SPM dentro de cuatro ciclos de reloj ejecuta la escritura de página, con el dato almacenado en el buffer temporal. La dirección de página se tomada desde la parte alta del puntero Z [Z13:Z7], mientras la dirección de palabra se toma de la parte baja de Z [Z6:Z1], en este instante los datos en R1 y R0 son ignorados.

El bit PGWRT se auto borrará cuando termina la escritura de la página, o de lo contrario, la instrucción SPM se ejecuta dentro de cuatro ciclos de reloj. La CPU se detendrá durante la operación de escritura de página cuando se direcciona a la sección NRWW.

- **Bit 1 – PGERS: Borrado de Página**

Si en este bit se escribe uno al mismo tiempo que SPMEN, la siguiente instrucción dentro de cuatro ciclos de reloj ejecuta el borrado de página. La dirección de página es tomada desde la parte alta del puntero Z y los datos en R1 y R0 son ignorados. El bit PGERS se borrará automáticamente cuando se complete el borrado de la página, o si no la instrucción SPM se ejecuta dentro de cuatro ciclos de reloj. La CPU se detiene durante la operación de borrado de página si se direcciona a la sección NRWW.

- **Bit 0 – SP MEN: Habilitar la capacidad de almacenar y borrar en la memoria**

Este bit habilita la instrucción SPM para los siguientes cuatro ciclos de reloj; si se escribe junto alguno de los siguientes bits RWWSRE, BLBSET, PGWRT o PGERS, la siguiente instrucción SPM tendría un significado especial de acuerdo a la descripción anterior de cada bit del registro SPMCSR, como por ejemplo si se ejecuta PGERS con SP MEN la siguiente instrucción SPM dentro de cuatro ciclos de reloj borrará una página de la memoria.

Si solo SP MEN es escrito, la siguiente instrucción SPM cargaría el valor en R1:R0 en el buffer temporal de página direccionado por el puntero Z. El LSB²² del puntero Z [Z6:Z1] es ignorado. El bit SP MEN se limpiaría automáticamente por completo cuando termina de ejecutarse la instrucción SPM, o si no la instrucción SPM es ejecutada dentro de cuatro ciclos de reloj. Durante el borrado de página y escritura de página, el bit SP MEN permanece en alto hasta que la operación se completa.

2.2.3.1 Descripción del programa computacional Avr Studio 4

Para la programación del software de Bootloader se ha elegido el programa computacional Avr Studio 4 (ver figura No. 2-21) desarrollado por ATMEL ya que es un software libre que se utiliza para programar los microcontroladores AVR.

Para la elaboración del Bootloader se ha escogido el lenguaje de programación Ensamblador, ya que permite un manejo adecuado de la memoria del microcontrolador.

²² LSB (Byte menos significativo) situado en la posición de menor valor potencial de un conjunto dado de bytes

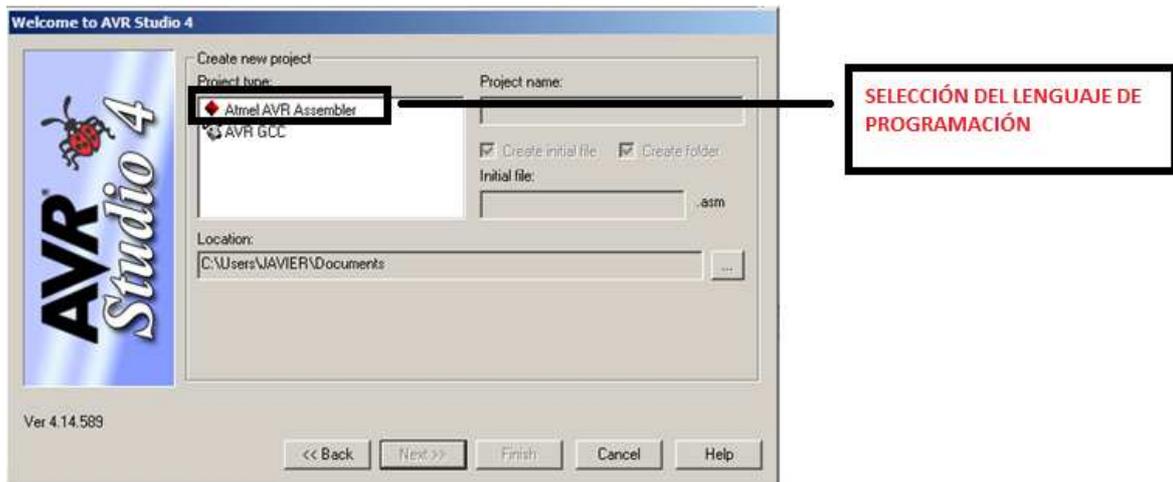


Fig. 2-21 Ventana para escoger el lenguaje de programación.

(La figura 2-22 muestra el ambiente de programación del software Avr Studio 4)

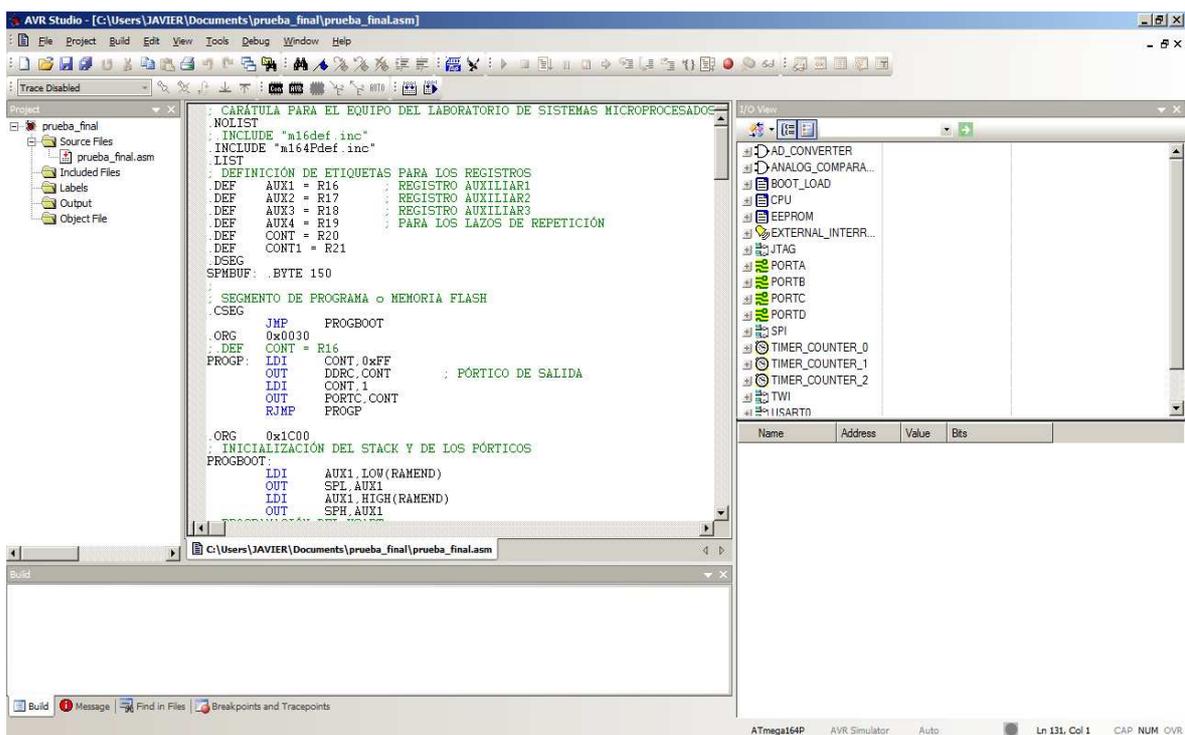


Fig. 2-22 Ventana de Trabajo del software AVR Studio 4

2.2.3.2 Programador Progisp

Ya que el microcontrolador viene vacío de fábrica, sin ningún programa cargado en la memoria flash, necesitamos de un programador para cargar por primera vez el Bootloader, para lo cual se usó el programador Progisp1.6.6 (ver figura No. 2-23), el programa residirá en la parte de la memoria llamada Bootloader, para lo cual en el programador seleccionamos los fusibles de tamaño de Bootloader, inicio del vector reset, la frecuencia del reloj que se utilizará y como es la primera vez que se va a cargar el Bootloader se escoge una programación ISP serial.

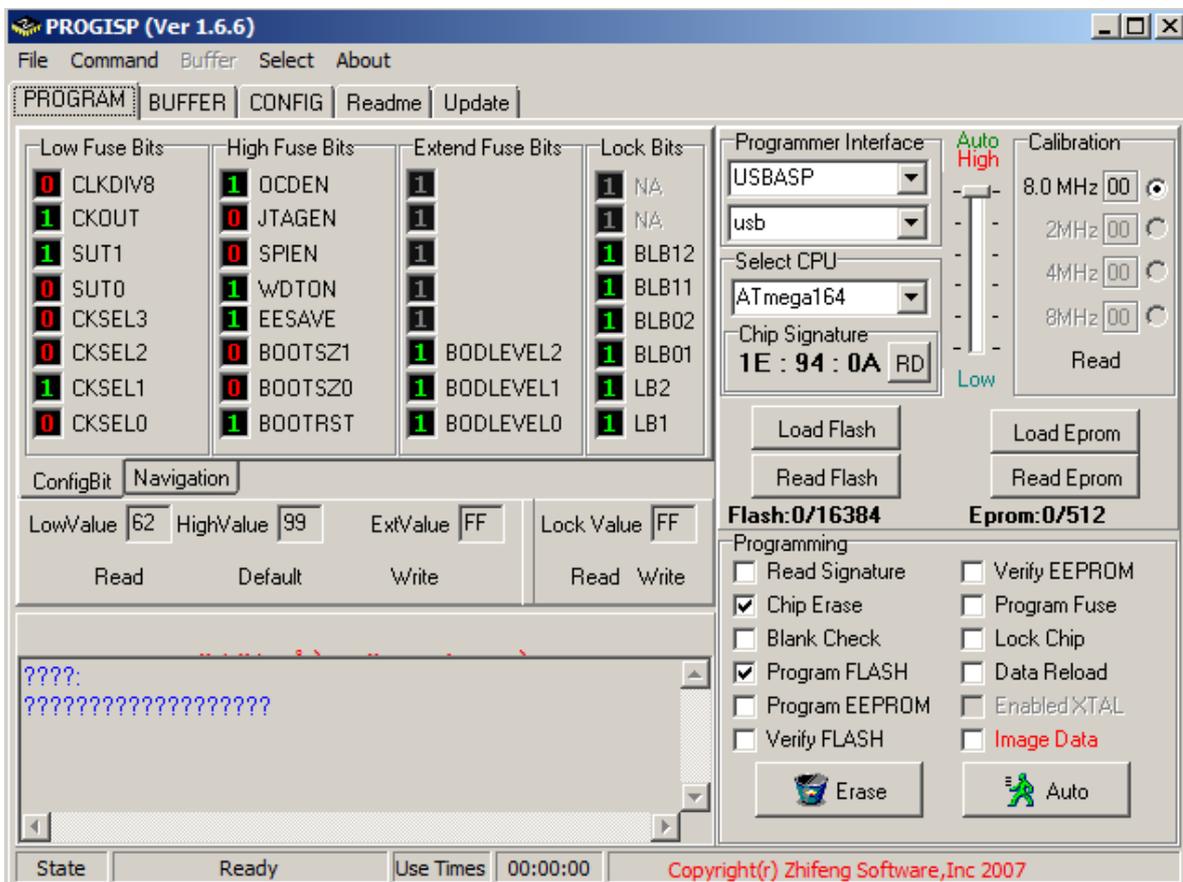


Fig. 2-23 Software Progisp 1.6.6

2.2.4 Programación del Bootloader

Para la programación del Bootloader se utiliza el lenguaje de programación Ensamblador, ya que permite un manejo adecuado y administración de la memoria.

Es necesario para el diseño inicializar en la sección de Bootloader, esto se puede lograr mediante un disparo con un comando de recepción vía USART²³, por medio de una interfaz SPI para inicializar el vector del reset, o una instrucción de salto JMP.

En el diseño se utiliza el programador Progisp para que el vector reset comience en la sección de Bootloader, activando el fusible de reset del Boot, para que el vector apunte a la parte del Bootloader de la Flash.

Una vez configurados los fusibles no pueden ser cambiados por el microcontrolador, esto significa que una vez programados los fusibles del reset, este vector siempre apuntará al Bootloader y el fusible solo puede ser cambiado a través de la interfaz de programación serial o paralela.

La dirección seleccionada para que resida el Bootloader es 0x1C00, que permite tener 1024 palabras de escritura, siendo la mayor capacidad de almacenamiento de aplicación que tiene el microcontrolador, debido a que las aplicaciones se han programado en lenguaje de alto nivel como es el lenguaje C, el mismo que utiliza mayores recursos de la memoria, por último se configuran los niveles de protección de la memoria.

2.2.4.1 Configuración de fusible

Para la protección de la sección Aplicación se configuran los fusibles BLB11 y BLB12 en uno lógico, se selecciona esta configuración para poder cargar y borrar cualquier programa sin ninguna restricción.

Como el programa de Bootloader no se va a cambiar en el diseño sino que se queda permanente, se selecciona el modo de protección donde no permita cambiar ni borrar el software, la configuración es BLB12 en uno lógico y BLB11 en cero lógico.

²³ USART (Transmisor y Receptor Sincrónico/Asincrónico Universal) se trata de un periférico para la transmisión de datos en formato serie, utilizando técnicas de transmisión sincrónica o asincrónica, según se configure el periférico.

Para que el microcontrolador arranque desde la sección de Bootloader, el vector de reset debe apuntar a la esta sección, para lo cual se configura el BOOTRST en cero lógico. El tamaño seleccionado se puede ver en la Tabla No. 2-9

BOOT SZ1	BOOT SZ0	TAM. BOOT	PÁG	SECC. APLICACIÓN	SECC. BOOTLOADER	FIN DE LA SECC. APLICACIÓN	INICIO DE LA SECC. BOOTLOADER
0	0	1024 Palabras	16	0x0000 - 0x1BFF	0x1C00 - 0x1FFF	0x1BFF	0x1C00

Tabla 2-9 Selección del Tamaño de las Secciones Aplicación y Bootloader

2.2.4.2 Borrar por página

El registro Z se utiliza para seleccionar la página que se va a borrar. El registro Z va a apuntar al inicio de la página para borrar de byte en byte. En el dispositivo el tamaño de página es de 64 palabras (128 bytes), los seis primeros bits del registro Z se ignoran. Para borrar una página, mediante programación los bits SP MEN y PGERS en el registro SPMCRS se activan y luego se ejecuta la instrucción de SPM.

2.2.4.3 Cargando la pagina de Buffer

Para escribir nuevos datos en una página de la memoria, primero se debe almacenar en la página del buffer, la cual es de 128 bytes (64 palabras). Esta página se llena palabra por palabra en R0 y R1, los bits menos significativos del registro Z [Z6:Z1] son ignorados debido a que se carga directamente desde la SRAM, luego la página del buffer escribe en una sola operación en la memoria Flash, para lo cual utiliza los bits [Z13:Z7] para direccionar la página y los bits [Z6:Z1] para direccionar la palabra que se desea escribir en la memoria del programa.

Para el desarrollo del software se van a recibir por el puerto serial 256 caracteres vía Bluetooth, los cuales se transforman a bytes mediante una

subrutina, para luego almacenar palabra por palabra en la memoria SRAM de microcontrolador para luego guardar en la página del buffer.

2.2.4.4 Escribir por páginas en la memoria

Para escribir la palabra en la memoria de la página, se carga la palabra en los registros R1 y R0. Se ajusta el registro Z para que apunte a la palabra correcta y se pone en uno lógico SPMEN en el Registro SPMCRS para que ejecute la instrucción de escritura de la página en la memoria. La instrucción SPM a continuación se ejecutada dentro de los cuatro ciclos.

Cuando el Buffer de la página se carga con nuevos datos, se escribe en la memoria Flash. Para ello en el registro Z se establece la dirección donde se quiere escribir la página. A continuación se ajustan los bits SPMEN y PGWRT en el Registro SPMCR y se ejecuta la instrucción SPM dentro de los cuatro ciclos.

2.2.4.5 Activación de la Sección RWW

Al realizar un borrado de página o escritura de página en la sección de RWW, el bit RWWSB se pone uno lógico automáticamente, para indicar que la sección RWW inaccesible. Para volver activar la sección RWW se debe poner en uno lógico el bit RWWSB junto a SPMEN y ejecutar la instrucción SPM dentro de los cuatros ciclos de reloj siguientes.

El contenido de los registros Z y la R1:R0 se omiten cuando se utiliza la función RWWSRE. Cuando se accede a la sección RWW no hay necesidad de volver a habilitarlo después de borrar o escribir.

A continuación se muestra la rutina para borrar y guardar una página en la memoria del programa.

```

ULTIM: ; SE INICIALIZA EN LA SRAM
LDI    YL,LOW(SPMBUF)
LDI    YH,HIGH(SPMBUF)
RCALL  WRTPAG ; RUTINA PARA ESCRIBIR Y BORRAR UNA PAGINA

; SE INICIALIZA EN LA SIGUIENTE PÁGINA
LDI    AUX4,128
ADD    ZL,AUX4
LDI    AUX4,0
ADC    ZH,AUX4
RCALL  UNSEG
RCALL  UNSEG
RCALL  UNSEG
RET
;RJMP  CARGAR1

WRTPAG: LDI    AUX1,0B00000011;CARGA EN EL REGISTRO AUX1(PGERS=1 Y SPMEN=1)PARA BORRAR UNA PÁGINA
RCALL  PROCESO ;LLAMA A LA RUTINA PARA VER SI EL BIT 0 (SPMEN = 0) Y CARGAR SPMCSR = AUX1
LDI    AUX1,0B00010001; HABILITA LA SECCIÓN RWW (RWSRE=1 Y SPMEN=1)
RCALL  PROCESO; LLAMA A LA RUTINA PARA VER SI EL BIT 0 (SPMEN = 0) Y CARGAR SPMCSR = AUX1

;
FILBUF: LDI    CONT,64; ; SE INICIALIZA EN 64 (64 # DE PALABRAS O 128 BYTES)
LD      R0,Y+ ; TIENE GUARDA EN R0 Y R1 LA PALABRA (DATOS)
LD      R1,Y+
LDI    AUX1,0B00000001 ; SPMEN=1 PARA GUARDAR LOS DATOS (SPM STORE PROGRAM MEMORY)
RCALL  PROCESO; LLAMA A LA RUTINA PARA VER SI EL BIT 0 (SPMEN = 0) Y CARGAR SPMCSR = AUX1
ADIW   ZH,ZL,2; SUMA 2 BYTES A REGISTRO Z, PARA PODER CARGAR LA SIGUIENTE PALABRA
DEC    CONT ; DECREMENTA EL CONTADOR
BRNE   FILBUF ; CUANDO CONT=0 SE TERMINO DE CARGAR TODA LA PÁGINA Y REGRESA PARA CARGAR LA SIGUIENTE PÁGINA

SUBI   ZL,128; RESTA 128 PARA QUE SE INICIALICE EN LA SIGUIENTE PAGINA DEBIDO QUE SE SUMO 2 EN EL PROCESO
SECI   ZH,0
LDI    AUX1,0B00000101; CARGA EN REGISTRO AUX1 (PGWRT=1 Y SPMEN=1) PARA ESCRIBIR UNA PÁGINA
RCALL  PROCESO ; LLAMA A LA RUTINA PARA VER SI EL BIT 0 (SPMEN = 0) Y CARGAR SPMCSR = AUX1
LDI    AUX1,0B00010001; HABILITA LA SECCIÓN RWW (RWSRE=1 Y SPMEN=1)
RCALL  PROCESO ; LLAMA A LA RUTINA PARA VER SI EL BIT 0 (SPMEN = 0) Y CARGAR SPMCSR = AUX1
RET

PROCESO:
DEMORA: IN      AUX2,SPMCSR ; GUARDA EL REGISTRO SPMCSR EN AUX2
SBRC   AUX2,0 ; VERIFICA QUE EL BIT 0 (SPMEN=0) PARA CONTINUAR
RJMP   DEMORA ; RETARDO
IN     AUX2,SREG; GUARDA LA BANDERA SREG = AUX2
CLI    ; DESACTIVA LA INTERRUPCIONES
OUT    SPMCSR,AUX1 ; CARGA AUX1 EN EL REGISTRO SPMCSR (PARA BORRAR O ESCRIBIR UN PÁGINA)
SPM    ; INSTRUCCION PARA GUARDAR EN LA MEMORIA DEL PROGRAMA
OUT    SREG,AUX2; REGRESA ELBANDERA INICIAL
RET

```

2.2.4.6 Diagrama de flujo

En la figura No. 2-24 se muestra el diagrama de flujo del programa principal del Bootloader, para almacenar en la memoria SRAM, borrar y escribir en la memoria del microcontrolador y la activación de nuevo de lectura de la sección RWW.

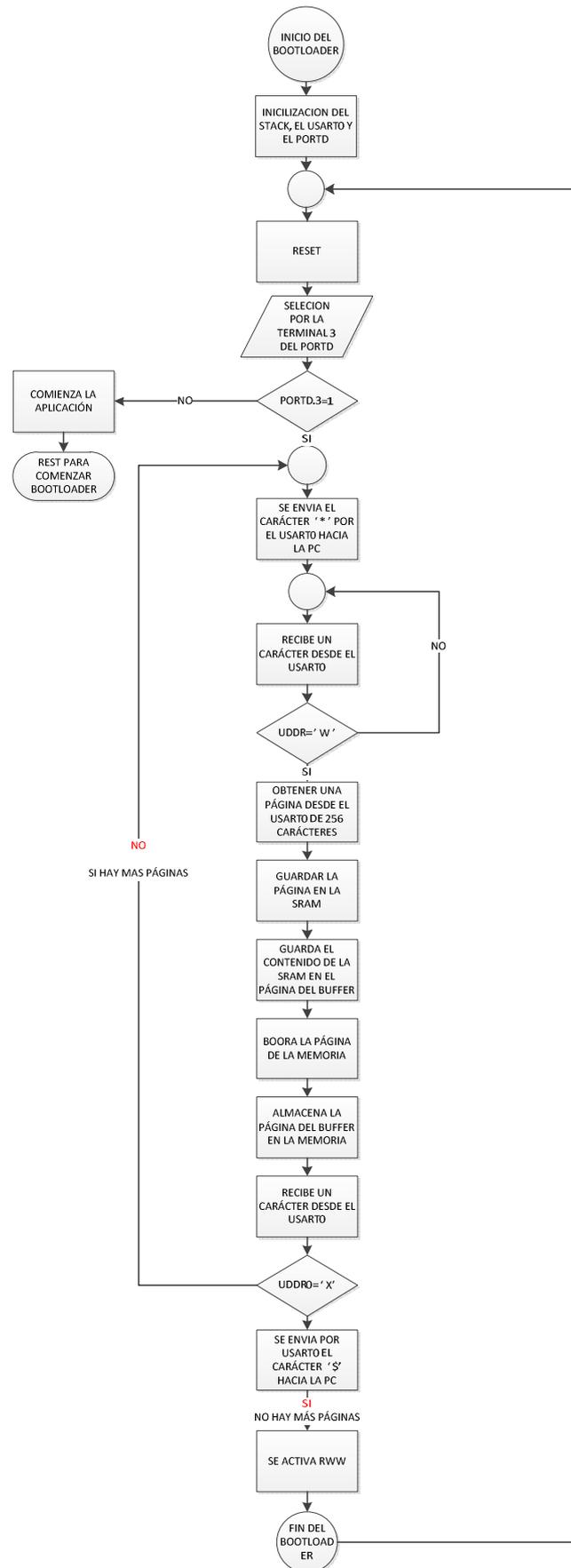


Fig. 2-24 Diagrama de flujo general del programa Bootloader

2.3 AMBIENTE GRÁFICO

Una vez realizado el software de Bootloader, se elaboró un ambiente visual para la descarga hacia el microcontrolador desde la PC, tomando en cuenta las características programadas en el Bootloader, como los parámetros de programación, para lo cual se ha utilizado una programación mediante clase utilizando el Software Visual Studio 2008.

2.3.1 VISUAL BASIC.NET

Visual Basic es un lenguaje de programación orientado a eventos. Este lenguaje de programación es un dialecto de BASIC, con importantes agregados para simplificar la programación utilizando una ambiente de desarrollo totalmente gráfico, para facilitar la creación de interfaces gráficas.

Visual Basic .NET (VB.NET) es un lenguaje de programación orientado a objetos que se puede considerar una evolución de Visual Basic implementada sobre el Framework .Net²⁴. El manejo de las instrucciones es similar a versiones anteriores de Visual Basic, facilitando así el desarrollo de aplicaciones más avanzadas con herramientas modernas.

Se utiliza el ambiente VB.NET integrado en Microsoft Visual Studio 2008. Al igual que con todos los lenguajes de programación basados en .NET, los programas escritos en VB .NET requieren el Framework .NET

Visual Basic (Visual Studio) contiene un entorno de desarrollo integrado o IDE²⁵ que incluye un editor de textos para edición del código, un depurador, un compilador (y enlazador) y un constructor de interfaz gráfica o GUI.

²⁴ Framework .NET que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones

²⁵ IDE(integrated development environment) es un programa informático compuesto por un conjunto de herramientas de programación

2.3.1.1 Interfaz gráfica para la descarga del programa

El ambiente gráfico permite que el estudiante seleccione la velocidad de transmisión, el puerto que se desea abrir, bits de paridad, bits de parada y el número de bits de información, todos estos son parámetros para la transferencia del programa de la práctica, También consta con un cuadro de diálogo para cargar el archivo de la práctica. En la figura No. 2-25 se puede ver la interfaz gráfica desarrollada en Visual Studio 2008.



Fig. 2-25 Interfaz gráfica para la descarga del programa

2.3.1.2 Diagrama de flujo de la Interfaz gráfica

En la figura No. 2-27 muestra el diagrama de flujo del programa principal de la interfaz gráfica, para la descarga del código de la aplicación hacia el dispositivo. A continuación se explican los pasos seguidos para la descarga de programa:

- Primero se seleccionan los parámetros para la comunicación
- Abre el puerto donde esté conectado el módulo Bluetooth
- Se carga el Archivo de extensión .Hex
- Los archivos de las prácticas son de formato Intel Hex. En la figura No. 2-26 se puede ver un ejemplo de un archivo de este formato. Se puede observar que estos archivos están compuestos por línea de caracteres

que siempre comienza con el carácter “:”, este carácter es el primero que se descarta, luego viene dos caracteres que son el número total de datos en la línea, este número está en hexadecimal, estos dos caracteres indicaran en el programa cuántos datos se debe tomar para guardar en el vector que después se enviará, después viene la dirección compuesta por 4 caracteres estos caracteres también se descartan, de ahí vienen dos caracteres que identifican el tipo de registro, los cuales pueden ser 00 si se trata de datos, 01 para indicar fin de archivo, 02 para dirección extendida, luego vienen los datos, estos son los que se separan para guardar en el vector, y una vez obtenidos los datos del archivo .Hex se envían hacia el dispositivo y por último vienen dos bytes de checksum.

```
:10010000214601360121470136007EFE09D2190140
:100110002146017EB7C20001FF5F16002148011988
:10012000194E79234623965778239EDA3F01B2CAA7
:100130003F0156702B5E712B722B732146013421C7
:00000001FF
```

Código de inicio
 Longitud
 Dirección
 Tipo de registro
 Datos
 Checksum

Fig. 2-26 Ejemplo de un archivo de formato intel Hex

- A continuación se muestra el código para abrir el archivo y obtener la longitud de los datos del archivo .Hex, como lo que entiende la computadora son caracteres Ascii se transforma a decimal para saber el valor:

```
' se abre el archivo
' se obtiene el total de caracteres de información del archivo
' se guarda en el vector que tiene que ser un numero entero multiplo de 256
Do While Not archivo.Peek = -1
linea = archivo.ReadLine ' Lee una línea de caracteres del archivo .HEX
```

```

n = Len(linea) 'longitud de la línea de caracteres
Dim i As Integer = 0

' RUTINA PARA OBTENER LA LONGITUD DE DATOS PARA CREAR UN VECTOR
For i = 1 To 3
    datos = Mid(linea, i, 1) ' descarta el caracter ':'
    If i = 2 Then
        pdigito = Mid(linea, i, 1) ' para obtener la longitud del capo de datos
    End If
    If i = 3 Then
        Dim dig10 As Integer
        dig10 = AscW(pdigito) 'transforma a ascii el primer caracter del capo longitud
        pdigito2 = Mid(linea, i, 1) 'obtien el segundo caracter del campo longitud
        ascdigito = AscW(pdigito2) ' tranforma a ascii el segundo caracter
        'rutina para transformar a decimal el segundo caracteres del campo longitud
        If ascdigito < 65 Then
            ascdigito = ascdigito - 48
        ElseIf ascdigito >= 65 Then
            ascdigito = ascdigito - 55
        End If
        ' si el primer caracter del campo longitud es 49 ascii o sea 1
        If dig10 = 49 Then
            obtenemos 16 en decimal el primer caracter del capo longitud
            ascdigito = 16 '
        End If
        longtotal += ascdigito * 2
    End If
Next
' muestra los datos (comienza en 10 hasta ascdigito*2)
datos = Mid(linea, 10, ascdigito * 2)
cont = cont + 1
Loop

```

- Una vez obtenido la longitud de los datos se calcula el número de las páginas para crear un vector de longitud (número de páginas x 256-1) para guardar los datos, debido a que en la memoria del microcontrolador se descargan por páginas de 256 caracteres o 128 bytes de datos.
- Como no va a salir un número entero las páginas se utiliza la función `longvector=Math.Round(aux1)` para redondear, pero si la resta entre el valor redondeado y el valor en fracción es mayor que cero se suma uno al valor redondeado, ya que la función `Math.Round` redondea al

inmediato superior cualquier entero con valor de fracción mayor a 5, pero se pueden tener valores enteros con valores de fracción menores a 5 donde redondea al inmediato inferior, operación que está mal, por lo cual se le suma uno.

```
Dim aux1 As Double = (longtotal / 256) ' aux1 se guarda el número de páginas
Dim longvector As Integer = Math.Round(aux1) ' redondea en longvector debido a que el resultado no es entero
    ' si la resta de aux- longvector es mayor que 0 se suma uno a longvector
    If aux1 - longvector > 0 Then
        longvector = longvector + 1 ' número de páginas
    End If
List2.Items.Add("TOTAL DE PÁGINAS")
List2.Items.Add(longvector)
' creamos un vector con la longitud de 256 x num de páginas
Dim data(longvector * 256 - 1) As String
```

- Se genera el vector con la longitud obtenida, que es de longitud $\text{longvector} * 256 - 1$, como los datos no siempre van hacer un múltiplo de 256 el vector se completa con el carácter "F", para que no afecte en la descarga hacia el dispositivo.

```
' se completa el vector en el caracter F para que sea un numero entero multiplo de 256
If counter < longvector * 256 - 1 Then
    Dim k As Integer = 0
    For k = counter To (longvector * 256 - 1)
        data(k) = "F"
    Next
End If
```

- Y por último se envía el vector generado por el puerto serial hacia el dispositivo.

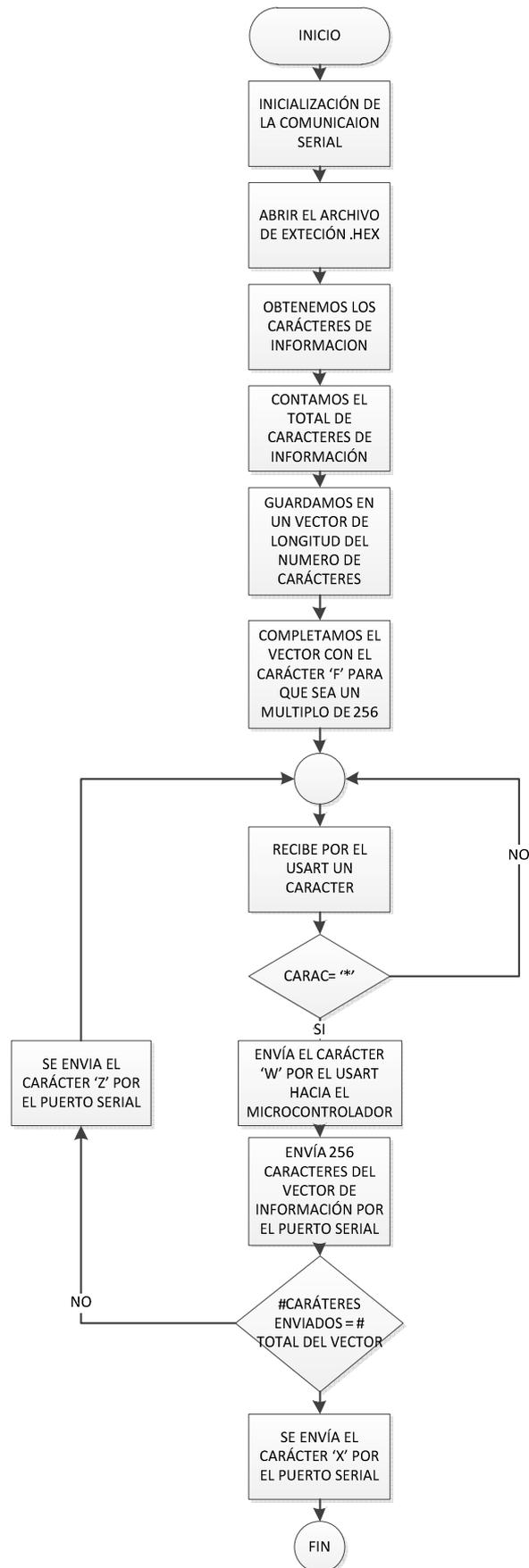


Fig. 2-27 Diagrama de flujo del programa principal de la interfaz gráfica

2.4 COSTOS

A continuación se realiza una cotización del proyecto, el cual incluye el precio de los elementos que conforman el prototipo como también la mano de obra para la realización del mismo. (Ver tabla 2-10)

DESCRIPCIÓN	ELEMENTOS	CANTIDAD	COSTO POR UNIDAD (\$)	COSTO TOTAL (\$)
PLACA DE TESIS	Microcontrolador ATMEGA 164P	1	7,5	7,50
	Modulo de Bluetooth T9JRN41	1	79	79,00
	LCD gráfico 128x64 JHD12864E	1	30	30,00
	Teclado 4x4	1	9	9,00
	Regulador de voltaje LM7805	1	0,55	0,55
	Potenciómetro tipo araña de 10k	1	0,17	0,17
	Cristal de cuarzo 11.0592 MHz	1	0,49	0,49
	Capacitor cerámico de 100nF	2	0,08	0,16
	Capacitores cerámico de 22pF	2	0,07	0,14
	Capacitores Electrolíticos de 10uF	4	0,06	0,24
	Diodo 1N4007	5	0,17	0,85
	Fusible de 1A	1	0,45	0,45
	Pulsadores	1	0,27	0,27
	Interruptor	1	0,37	0,37
	Espadines de 90 grados	1	0,75	0,75
	Espadines	2	0,1	0,20
	Sócalo de 40 terminales	2	0,22	0,44
	Jack	1	0,1	0,10
	Conector macho	1	0,12	0,12
	Circuito impreso	1	44,5	44,50
ACCESORIOS	Caja del proyecto	1	7	7,00
	Cable UTP Cat. 5e	1	0,33	0,33
	Mini USB Bluetooth	1	5	5,00

TOTAL DE MATERIALES				187,63
MANO DE OBRA	Mano de Obra (Horas)	360	16	5760
COSTO DEL PROYECTO				5947,63

Tabla 2-10 Descripción de costos del PDATP164p

3. CAPÍTULO TERCERO: EJEMPLOS DE PRÁCTICAS DE LABORATORIO

3.1 INTRODUCCIÓN

Para la realización de las prácticas se utiliza el software computacional Avr Studio 4, con su compilador WinAvr, que es compatible con ANSI C, adicionalmente se dispone de librerías específicas para el manejo de los diferentes recursos.

3.2 WINAVR

Antes de la programación primero se presenta los componentes de una plantilla genérica para establecer programas escritos para el compilador C de WinAvr:

```

//*****//
//Titulo :
// Autor :
// Fecha:
// Version:
// Target MCU : Atmel AVR series
//*****//
//-----//
//***** Inclusión de Librerías *****//
//-----//
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <avr/pgmspace.h>
#include <inttypes.h>
#include "lcd_lib.h"
//-----//
//***** Definiciones de Variables Globales *****//

```

```

// ***** Definiciones y Macros *****//
// ***** Rutinas de Interrupción *****//
// ***** Declaración de Funciones *****//
//-----//
// ***** Cuerpo del Programa *****//
//-----//

int main( void ){
// ***** Declaración de Variables *****//
// ***** Configuración de los Puertos de E/S *****//
// ***** Inicialización de Registros *****//
// ***** Habilitación de Interrupciones *****//
// ***** Inicialización de Funciones y Variables *****//
//-----//
***** Ciclo Principal de Ejecución *****//
//-----//

while (1)
{
}
//Fin while
return 0;
}
//***** Definición De Funciones *****//

```

El compilador WinAvr dispone de una serie de librerías que pueden incluirse dentro del programa, algunas son estándar C y otras propias de WinAvr

3.2.1 LIBRERIAS

El “# include” es una directiva que indica al compilador que busque el archivo que se encuentra dentro de los símbolos menor que y mayor que “<>”, estas librerías siempre van al comienzo de los programas, debido a que es lo primero que va a compilar. Como ejemplos se tienen el io.h, esta librería incluye la información de entrada y salida del microcontrolador, para el manejo de pines y el delay.h que proporciona las definiciones de la demora.

3.2.2 VARIABLES GLOBALES

Las variables comunes, solo son accesibles desde la función que se halla definida, si se requiere que la variable sea accesible desde cualquier parte del programa, es necesario declararla como global.

Para esto se define antes de la función main(). Ejemplo.

```
volatile int var1, var2
```

Es aconsejable, aunque no obligatorio definir las variables globales como volátiles para indicarle al compilador que no sobrescriba esta posición de memoria.

3.2.3 DEFICIONES Y MACROS

Mediante ciertas directivas se puede asignar a algún símbolo definido por el programador una determinada expresión. A continuación los siguientes ejemplos:

```
# define      M                2
#define      _BV                (1<<(bit))
#define      set_bit(value, bit) (value|((value)&~_BV(bit)))
```

3.2.4 RUTINAS DE INTERRUPCIÓN

La ejecución normal del programa sigue una secuencia de instrucciones. Sin embargo, a veces, esta secuencia normal se debe interrumpir para responder a una situación de alta prioridad, tanto dentro como fuera de la microcontrolador. Cuando estos eventos de mayor prioridad se producen, el microcontrolador debe suspender temporalmente el funcionamiento normal y ejecutar acciones de eventos específicos llamados rutinas de interrupción. Una vez que el evento de mayor prioridad ha sido reparado, el microcontrolador continúa procesando el programa normal. En el ejemplo se activa la interrupción 2 de

microcontrolador, el cual se detiene para sumar uno a la variable a y hace un proceso de retardo y luego procesa normalmente el programa principal.

```
ISR(INT2_vect)
{
a++;
delay_ms(100);
}
```

No es necesario hacer un salto implícito a esta función, ni ejecutar alguna instrucción de retorno.

El nombre de los diferentes vectores de interrupción se definen en la librería <avr/io.h>, y va de acuerdo al microcontrolador, En la tabla No 3-1 se observan las interrupciones para el Atmega 164P.

1	S0000	REST	Terminal externo, reajusta el encendido, Watdog y el JTAG
2	S0002	INT0	Solicitud de interrupción externa 0
3	S0004	INT1	Solicitud de interrupción externa 1
4	S0006	INT2	Solicitud de interrupción externa 2
5	S0008	PCINT0	Terminal de solicitud de interrupción de cambio 0
6	S000A	PCINT1	Terminal de solicitud de interrupción de cambio 1
7	S000C	PCINT2	Terminal de solicitud de interrupción de cambio 2
8	S000E	PCINT3	Terminal de solicitud de interrupción de cambio 3
9	S0010	WDT	Interrupción de tiempo de espera Watchdog
10	S0012	TIMER2_COMPA	Temporizador/Contador2 comparador A
11	S0014	TIMER2_COMPB	Temporizador/Contador2 comparador B
12	S0016	TIMER2_OVF	Temporizador/Contador2 por desbordamiento
13	S0018	TIMER1_CAPT	Temporizador/Contador1 por captura de eventos
14	S001A	TIMER1_COMPA	Temporizador/Contador1 comparador A

15	S001C	TIMER1_COMPB	Temporizador/Contador1 comparador B
16	S001E	TIMER1_OVF	Temporizador/Contador1 por desbordamiento
17	S0020	TIMER0_COMPA	Temporizador/Contador0 comparador A
18	S0022	TIMER0_COMPB	Temporizador/Contador0 comparador B
19	S0024	TIMER_OVF	Temporizador/Contador0 por desbordamiento
20	S0026	SPI_STC	SPI por transferencia de serie completa
21	S0028	USART0_RX	USART0 Rx Completa
22	S002A	USART0_UDER	USART0 por registro de datos vacía
23	S002C	USART0_TX	USART0 Tx Completa
24	S002E	ANLOG_COMP	Comparador analogo
25	S0030	ADC	ADC conversion complete
26	S0032	EE_READY	Preparado la EEPROM
27	S0034	TWI	Interfaz serial de 2 líneas
28	S0036	SPM_READY	Preparado par almacenar en la memoria del programa
29	S0038	USART1_RX	USART1 Rx Completa
30	S003A	USART1_UDRE	USART1 por registro de datos vacía
31	S003C	USART1_TX	USART1 Tx Completa

Tabla 3-1 Interrupciones del microcontrolador ATMEGA 164P

3.2.5 DEFINICIÓN DE FUNCIONES

En este campo debe colocarse la declaración de las funciones a emplear con el prototipo de la función. Ejemplo.

```
void printpantalla(int ubica, int tamano);
```

3.2.6 CUERPO DEL PROGRAMA

El cuerpo del programa siempre empieza con la función main() desde donde empieza su ejecución. Ejemplo.

```
int main( void )
{
}
```

3.2.6.1 Declaración de variables

Las variables que se definen tienen un ámbito local, no son accesibles desde las funciones.

3.2.6.2 Configuración de los puertos

El ATMEGA164P está equipado con cuatro pórtricos de 8 bits de propósito general, para la utilización de I/O digital los puertos se denominan como PORTA, PORTB, PORTC, y PORTD. Todos estos puertos también tienen funciones alternativas. Como se muestra en la Figura No. 3-1, cada puerto tiene tres registros asociados:

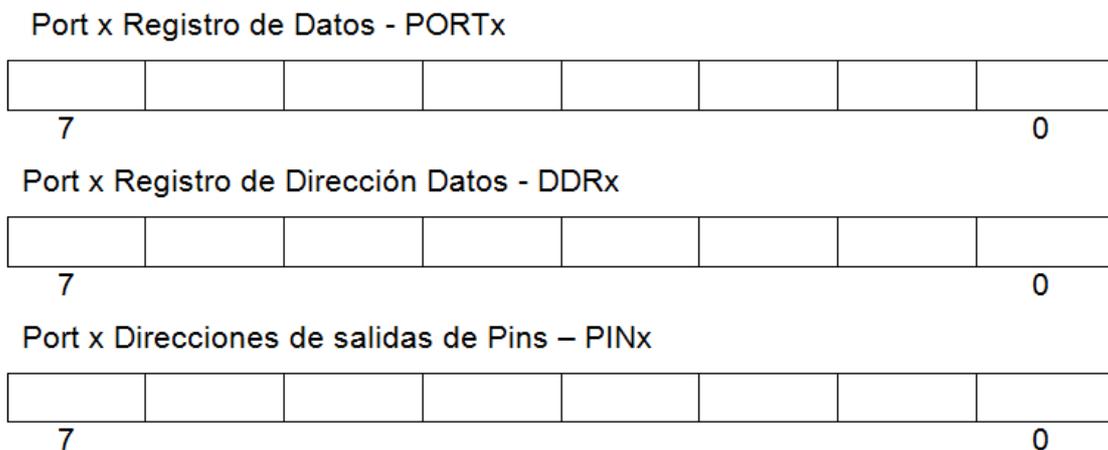


Fig. 3-1 Registros asociados a cada pórtrico del microcontrolador Atmega 164P

3.2.6.2.1 DDRx

Este registro permite configurar los pórtricos del microcontrolador como entrada o salida

- 1 configura como salida
- 0 configura como entrada

3.2.6.2.2 PORTx

El registro se emplea para enviar información por el pórtrico. En el caso que un determinado pin se configure como entrada, en el registro PORTx se coloca uno lógico, lo que se habilita la correspondiente resistencia de pull-up.

3.2.6.2.3 *PIN_x*

Con el registro Pinx se permite leer la información que ingresa al microcontrolador, lo cual permite leer los estados de los pines.

Los Pines del puerto se suelen configurar en el inicio de un programa, ya sea de entrada o salida, y sus valores iniciales se establecen más adelante. Por lo general, los ocho pines para un puerto determinado se configuran al mismo tiempo. Un ejemplo del código se proporciona a continuación para mostrar cómo se configuran los puertos.

```
// Configura PORTA como salida
PORTA=0;DDRA = 0xFF;
// Configura PORTB como salida
PORTB=0;DDRB = 0xFF;
// Configura PORTC como entrada
PORTC=0;DDRC = 0x00;
// Configura PORTD como entrada y salida
PORTD=0;DDRD = 0x0F;
```

3.2.6.3 Inicialización de registros

Se establecen los valores de los registros de propósito específicos como interrupciones, conversor análogo digital, timers, etc., de acuerdo con las configuraciones de los diferentes recursos de microcontrolador. Ejemplo

```
\ \ Registro de Control de Interrupciones A
EICRA=0x00;
```

3.2.6.4 Habilitación de Interrupciones

Habilitación de los diferentes bits de interrupción. Ejemplo.

```
\ \ Registro de activación de interrupciones
EIMSK=0x05;
```

3.2.6.5 Inicialización de funciones y variables

Se debe establecer el estado inicial de las distintas variables y ejecutar las funciones de inicialización requeridas. En el ejemplo que sigue se llama función prinpantalla y se inicia con los valores 0 y 1024.

```
prinpantalla(0,1024);
```

3.3 INTRODUCCIÓN AL LENGUAJE C PARA EL COMPILADOR WINAVR

3.3.1 TIPOS DE DATOS BÁSICOS

Como ejemplo se tiene la declaración variables de 8 bits, empleado especialmente para representación de caracteres.

```
char
```

El Ansi C los enteros se pueden implementar, los tipos int y long int, para datos de 16 y 32 bits.

```
Int (-32768 a 32767)
```

```
Unsigned int (0 a 65535)
```

```
Long int (-2147483648 a 2147483647)
```

La librería #include <inttypes.h> implementa otros tipos adicionales:

```
Int8_t      uint8_t
```

```
Int16_t     uint16_t
```

```
Int32_t     uint32_t
```

```
Int64_t     uint64_t
```

float y double son punto flotante de 32 bits

3.3.2 OPERADORES

La tabla No. 3-2 muestran los operadores aritméticos que se usa en el compilador WinAvr.

Operador	Nombre	Ejemplo	Definición
*	Multiplicación	$X*Y$	Multiplica x por y
/	División	X/Y	Divide x para y
%	Modulo	$X\%Y$	Proporcionan la resta de x dividido por y
+	Suma	$X+Y$	Suma X mas Y
-	Resta	$X-Y$	Resta y de x
++	Incremento	$X++$	Incremento de x después de usarlo
--	Decremento	$--X$	Decremento de y antes de usarlo
-	Negación	$-X$	Multiplica x por -1

Tabla 3-2 Operadores Aritméticos

En la tabla No. 3-3 se muestran los operadores lógicos y relacionales que se usan comúnmente en el compilador WinAvr.

Operador	Nombre	Ejemplo	Definición
>	Mayor que	$X > Y$	1 si x es mayor que y, de otra manera 0
>=	Mayor o igual que	$X \geq Y$	1 si x es mayor o igual que y, de otra manera 0
<	Menor que	$X < Y$	1 si x es menor que y, de otra manera 0
<=	Menor o igual que	$X \leq Y$	1 si x es menor o igual a y, de otra manera 0
==	Igual a	$X == Y$	1 si x igual a y, de otra manera 0
!=	No igual a	$X != Y$	1 si x no es igual a y, de otra manera 0
!	Operador NOT	$!X$	1 si x es 0, de otra manera 0

&&	Operador AND	X && Y	0 si x o y es 0, de otra manera 1
	Operador OR	X Y	0 si ambos X y Y son 0, de otra manera 1

Tabla 3-3 Operadores lógicos y relacionales

En la Tabla No. 3-4 se indican los operadores bit a bit necesarios para simplificar la nomenclatura de los programas.

Operador	Nombre	Ejemplo	Definición
~	Complemento bit a bit NOT	~X	Cambian los bits 1 por 0 y los bits 0 por 1
&	Bit a bit AND	X&Y	Bit a bit AND de X y Y
	Bit a bit OR	X Y	Bit a bit OR de X y Y
^	Bit a bit or exclusivo	X^Y	Bit a Bit XOR de X y Y
<<	Desplazamiento a la izquierda	X<<2	Bits en X se desplazan a la izquierda 2 bits
>>	Desplazamiento a la derecha	X>>3	Bits en X se desplazan a la derecha 3 bits

Tabla 3-4 Operadores Bit a Bit

3.3.3 ESTRUCTURAS DE CONTROL

Un lenguaje debe disponer de dos formas básicas para controlar el flujo del programa²⁶, que son:

- a) Decisión. Ejecutar sentencia entre acciones alternativas.
- b) Repetición. Se repite una secuencia de sentencias hasta que se cumpla una determinada condición.

²⁶ El Flujo de programa es la secuencia que el microcontrolador sigue para ejecutarlo

3.3.4 Decisión Binaria

La decisión binaria decide la ejecución de las sentencias entre dos alternativas.

3.3.4.1 Sentencia if

La sentencia if permite escoger entre realiza una acción o no hacerlo. Es la forma más sencilla de una sentencia If para elegir entre ejecutar una sentencia o saltarla.

3.3.4.2 Sentencia if-else

La sentencia if-else puede escoger entre dos acciones diferentes. Permite la selección entre dos sentencias; elige la una y la otra no la toma en cuenta

3.3.4.3 Sentencia else-if

La sentencia else-if es la misma sentencia if-else, pero está se utiliza para elegir una única acción entre más de dos posibles alternativas.

La estructura general de estas tres sentencias son:

```

if (expression1)
{
-----;
-----;
} else if (expression2)
{
-----;
-----;
} else if (expression3)
{ -----;
-----;
} else
{
-----;
-----;
}

```

3.3.4.4 Switch

Utiliza la sentencia switch cuando se necesita elegir una o varias acciones entre varias alternativas, por lo tanto, esta sentencia no es excluyente. La sentencia general para la sentencia switch es:

```
switch (expression)
{
  Case constant expression1:
  -----;
  -----;
  Break;
  Case constant expression2:
  -----;
  -----;
  Break;
  Default constant expression2:
  -----;
  -----;
  Break;
}
```

3.3.5 SENTENCIAS DE REPITICIÓN

3.3.5.1 While

La sentencia while crea un bucle o lo que se conoce como lazo que repite hasta que la expresión de control llamada “test o expresión” se vuelva falsa. La sentencia general para la sentencia while es:

```
while(expression)
{
  -----;
  -----;
}
```

3.3.5.2 do – While

La sentencia do-while crea un lazo que permite repetir hasta que la expresión de control sea falsa. Se diferencia con la sentencia while en que la condición del test debe estar al final del cuerpo del lazo. La estructura general de la sentencia do-while es:

```
do
{ -----;
-----;
} while(expression);
```

Ejemplo;

```
while ( i < 128)
{
PORTB = 2*i;
i++;
}
```

3.3.5.3 For

Este lazo agrupa en un solo lugar las tres acciones: a) Inicializar un contador, b) Compararlo con un límite; y c) Incrementarlo cada vez que se ejecute el lazo.

Es decir, en un lazo for se pueden sustituir las tres acciones en una sola sentencia.

```
for(expression1; expression2; expression3)
{
-----;
-----;
}
```

3.3.6 MANEJO DE LO PÓRTICOS

Para una mejor programación se debe tener un mayor manejo de los pÓrticos, para utilizar las varias funciones que pueden realizar los mismos, para lograrlo se utilizan varios operadores.

Ejemplos.

- Define los pines 0, 1, 2, 3, 4 como salida y el resto de pines como entradas

```
DDRB = 0x1F;
```

```
DDRB =( 1<< DDB0)|( 1<< DDB1)|( 1<< DDB2)|( 1<< DDB3)|( 1<< DDB4);
```

- Se activa la Pull up en el terminal 2 del puerto B

```
PORTB= 0x04;
```

```
PORTB= PORTB | 0x04; Realiza PORTB OR 0X04
```

```
PORTB|=( 1<<PB2); Activa 1 lógico en la terminal 2 y realiza (PORTB OR PB2)
```

- Pone las terminal 4 y 5 en alto sin alterar las demás.

```
PORTB |=( 1<<PB4)|( 1<<PB5);
```

- Pone los pines 4 y 5 en 0 sin alterar los demás.

```
PORTB &= ~( (1<<PB4)|(1<<PB5) );
```

- Formas de examinar un pin.

```
if(PINC &(1<<PINC1) ) {Será verdadero si el pin 1 está en alto.
```

```
/* action */
```

```
}
```

```
if(!(PINB &(1<<PINB2)) ) { Será verdadero si el pin 1 está en bajo.
```

```
/* action */
```

```
}
```

Para simplificar la escritura pueden definirse ciertos macros. En los siguientes se define para el manejo de los bits utilizados en la librería del GLCD.

```
#define SETBIT(ADDRESS,BIT)
```

```
(ADDRESS |= (1<<BIT))
```

```
#define CLEARBIT(ADDRESS,BIT)
```

```
(ADDRESS &= ~(1<<BIT))
```

3.4 REALIZACIÓN DE LAS PRÁCTICAS DE LABORATORIO DE SISTEMAS MICROPROCESADOS EN WINAVR

Con la introducción para la programación en C, se realizan las prácticas de Laboratorio de Sistemas Microprocesados, utilizando el compilador WinAvr, antes explicado, por lo cual se ha elaborado 3 prácticas de laboratorio, las cuales son:

- Cuenta Personas
- Cerradura Electrónica
- Voltímetro

Se va explicar las rutinas para el manejo de la LCD gráfica y Teclado, elementos principales del dispositivo y de las prácticas.

3.4.1 RUTINA PARA EL MANEJO DE LA LCD GRÁFICA

Para el manejo de la LCD gráfica se toma en consideración sus características técnicas, y del diseño de hardware del dispositivo.

Primero se crea las pantallas o menús que va mostrar el GLCD, para lo cual se utiliza la herramienta Paint y el programa BMP to ASM para transformar los bits a caracteres hexadecimales, los mismos que se utilizan para mostrar en la GLCD.

Se tiene en consideración que GLCD y el teclado están conectados al mismo pòrtico, para lo cual tenemos que inicializar cada vez que se vaya a usar uno u otro elemento.

Finalmente se realiza las rutinas para mostrar la salida de información de las prácticas en la GLCD.

3.4.1.1 Creación de menús para la GLCD

Para la elaboración de los menús y de la información que va a mostrar la GLCD se crea tablas con la información en formato asm para lo cual utilizamos un programa que convierta imágenes creadas en el software Paint de Windows, que están en formato BMP a ASM para que entienda el microcontrolador.

3.4.1.1.1 Creación de menús en Paint

Para la elaboración de una pantalla de menú primero se construye el dibujo en el software Paint para lo cual se utiliza características monocromáticas, de resolución de 128x64 pixeles en formato bitmap (.BMP).

El Bitmap es el formato propio del programa Paint, que viene con el sistema operativo Windows. Puede guardar imágenes de 24 bits (16,7 millones de colores), 8 bits (256 colores) y menos. Configura estos archivos a una compresión sin pérdida de calidad.

Los archivos con extensión .BMP, en los sistemas operativos Windows, representan la sigla BitMaP (o también Bit Mapped Picture), o sea mapa de bits.

En la figura No. 3-2 se observa la imagen que va a mostrar la GLCD creada en Paint.

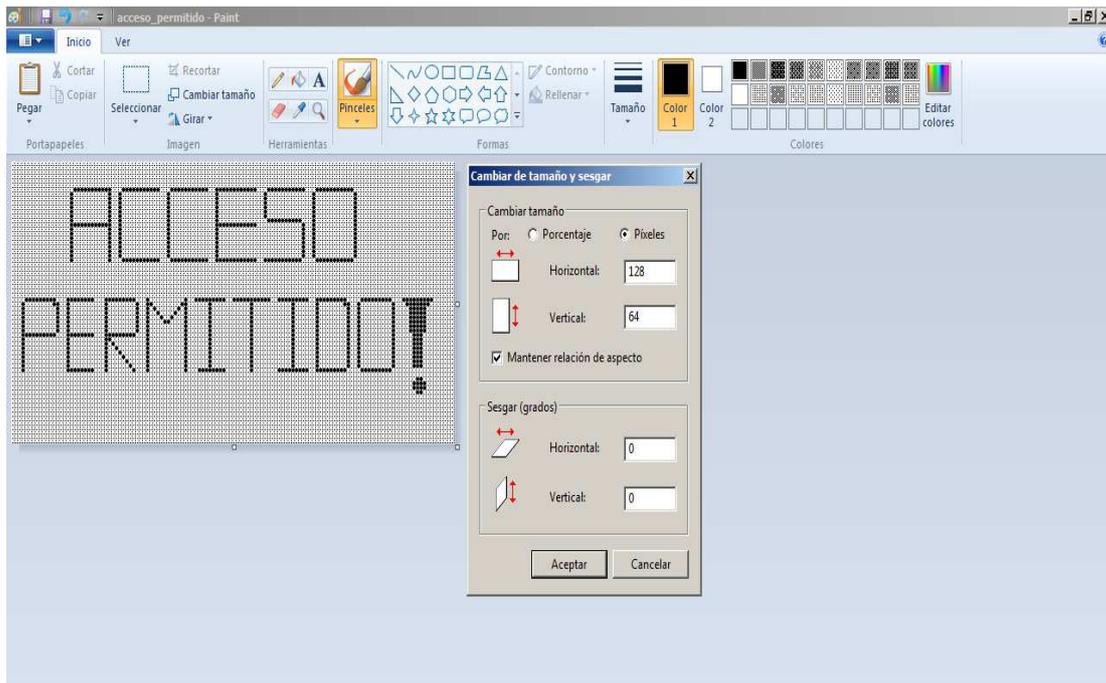


Fig. 3-2 Diseño del menú principal que va mostrar la GLCD en el Software Paint

Teniendo la imagen monocromática de resolución 128x64 píxeles, tamaño de la GLCD utilizada en formato .BMP, se pasa a convertir a lenguaje ensamblador. ASM que entiende el microcontrolador.

3.4.1.1.2 *Trasformación de BMP a ASM*

Una vez obtenida la imagen en formato bitmap se transforma a lenguaje ensamblador utilizando un software llamado BMP to ASM, el cual se puede observar en la figura No. 3-3.

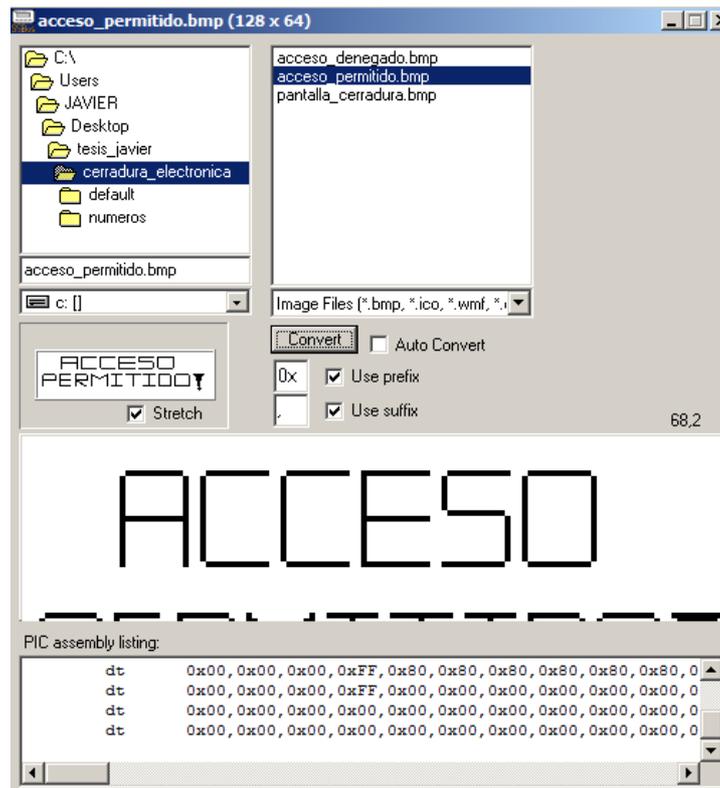


Fig. 3-3 Conversión de la imagen en extensión BMP a lenguaje de máquina

En el software se selecciona el archivo de extensión .BMP que contendrá la imagen a mostrar y se escoge stretch para que la imagen entre en el tamaño de la GLCD y se selecciona el prefijo y el sufijo que están en la conversión.

Para facilitar la programación se escogió el prefijo "0x" y el sufijo "," para separar los caracteres hexadecimales y se presiona convertir.

Finalmente el programa entrega una serie de caracteres hexadecimales que contiene la información de la imagen que se va a mostrar en la GLCD.

3.4.1.2 Control de la GLCD gráfica.

La tabla No. 3-5 se muestra el conjunto de instrucciones de control de la LCD gráfica JHD12864E. Estas instrucciones son enviadas desde el microcontrolador a la GLCD. Se observa la instrucción para encender y apagar la GLCD y para el manejo del mismo.

Instrucción	RS	R/W	DB 7	DB 6	DB 5	DB 4	DB 3	DB 2	DB 1	DB 0	Función
Display on/off	L	L	L	L	H	H	H	H	H	L/H	Controles de la pantalla encendida o apagada. L:OFF , H: ON
Conjunto de direcciones (dirección Y)	L	L	L	H	Direcciones Y (0-63)					Establece la dirección de Y en la dirección de contador Y	
Conjunto de páginas (dirección X)	L	L	H	L	H	H	H	Pagina (0-7)			Establecer la dirección de X en la dirección de registro X
Comienzo de línea de display (dirección Z)	L	L	H	H	Comienzo de línea del display (0-63)					indica la memoria RAM de datos de la pantalla que aparecen en la parte superior de la pantalla	
Estado de lectura	L	H	BU SY	L	ON/ OF F	RE SET	L	L	L	L	Estado de lectura: BUSY: L: lee H: En operación ON/OFF L: Encendido H: Apagado RESET L: Normal H: Reset
Escribir datos en la pantalla	H	L	Escribir datos								Escribe datos (DB0-7) en la memoria RAM de la pantalla. Después de instrucción escritura, la dirección Y se incrementa en 1 de forma automática
Leer datos	H	H	Leer Datos								Leer datos (DB0-7) desde la pantalla de RAM para el bus de datos

Tabla 3-5 Conjunto de Instrucciones para el control de la GLCD JHD12864E

En la figura No. 3-4 constan las funciones de inicialización de la GLCD de acuerdo al conjunto de instrucciones de la tabla No. 3-5. Primero se habilita CS1 y CS2 poniendo 1 lógico en el pines del puerto D, se reinicia RS y RW de GLCD, luego se inicializa la línea del display, después se habilita la señal y se deshabilita; eso se observa en la figura No. 3-5 en la función strobe, después se inicializa el número de paginas y de direcciones, al final se deshabilita CS1 y CS2.

```

void initglcd (void)
{
    PORTC=0;
    SETBIT(PORTD,CS1);           //Habilita CS1 del display
    SETBIT(PORTD,CS2);           //Habilita CS2 del display
    CLEARBIT(PORTD,RS);          //Limpia RS
    CLEARBIT(PORTD,RW);          //Limpia RW
    delay_ms(2);
    PORTC = 0b11000000;          //Inicializa la linea del display =0 (0-63)
    estrobe();
    delay_ms(2);
    PORTC = 0b01000000;          //Conjunto de direcciones = 0 (0-63)
    estrobe();
    delay_ms(2);
    PORTC = 0b10111000;          //Conjunto de página= 0 (0-7)
    estrobe();
    delay_ms(2);
    PORTC = 0b00111111;          //Enciende el display
    estrobe();
    CLEARBIT(PORTD,CS1);          //desabilita el CS1 del diplay
    CLEARBIT(PORTD,CS1);          //desabilita el CS12del diplay
}

```

Fig. 3-4 Función para inicializar la GLCD

```

void estrobe (void)
{
    unsigned char i;
    SETBIT(PORTD,EN);             //GLCD 'E' pósito Alto
    for(i=0;i<2;i++)
    {}
    CLEARBIT(PORTD,EN);           //GLCD 'E' pósito Bajo
    for(i=0;i<2;i++)
    {}
}

```

Fig. 3-5 Función para habilitar y deshabilitar la señal

3.4.1.3 Diagrama de flujo para el manejo de la GLCD.

En la figura No. 3-6 se muestra el diagrama de flujo de la librería que maneja la GLCD, donde contiene la inicialización de los vectores de información como las diferentes funciones para imprimir en la GLCD, como también las funciones de inicialización y posicionamiento, de acuerdo a la tabla de instrucciones (ver tabla No. 3-5)



Fig. 3-6 Diagrama de Flujo de la Librería que maneja la GLCD

3.4.1.4 RUTINA PARA EL MANEJO DEL TECLADO 4X4

El teclado es un dispositivo pasivo compuesto de 8 pines, que es controlado mediante software; hay varias formas de manejarlo, la que se utiliza en el presente trabajo es mediante un barrido permanente, leyendo si cambia a un estado bajo alguno de los 4 primeros pines que están conectados al teclado.

El barrido permanente cambia pin a pin, dentro de los cuatros últimos pines. Si se presiona una tecla se lee un estado bajo y el microprocesador, obtiene el valor de la tecla presionada.

Como la GLCD y el teclado están conectados al mismo puerto del microcontrolador, se tiene que inicializar dos veces en cada lazo de procesamiento; uno para la GLCD y el otro para el teclado.

A continuación se muestra el diagrama de flujo del manejo del Teclado

3.4.1.5 Diagrama de Flujo par el manejo del Teclado 4x4

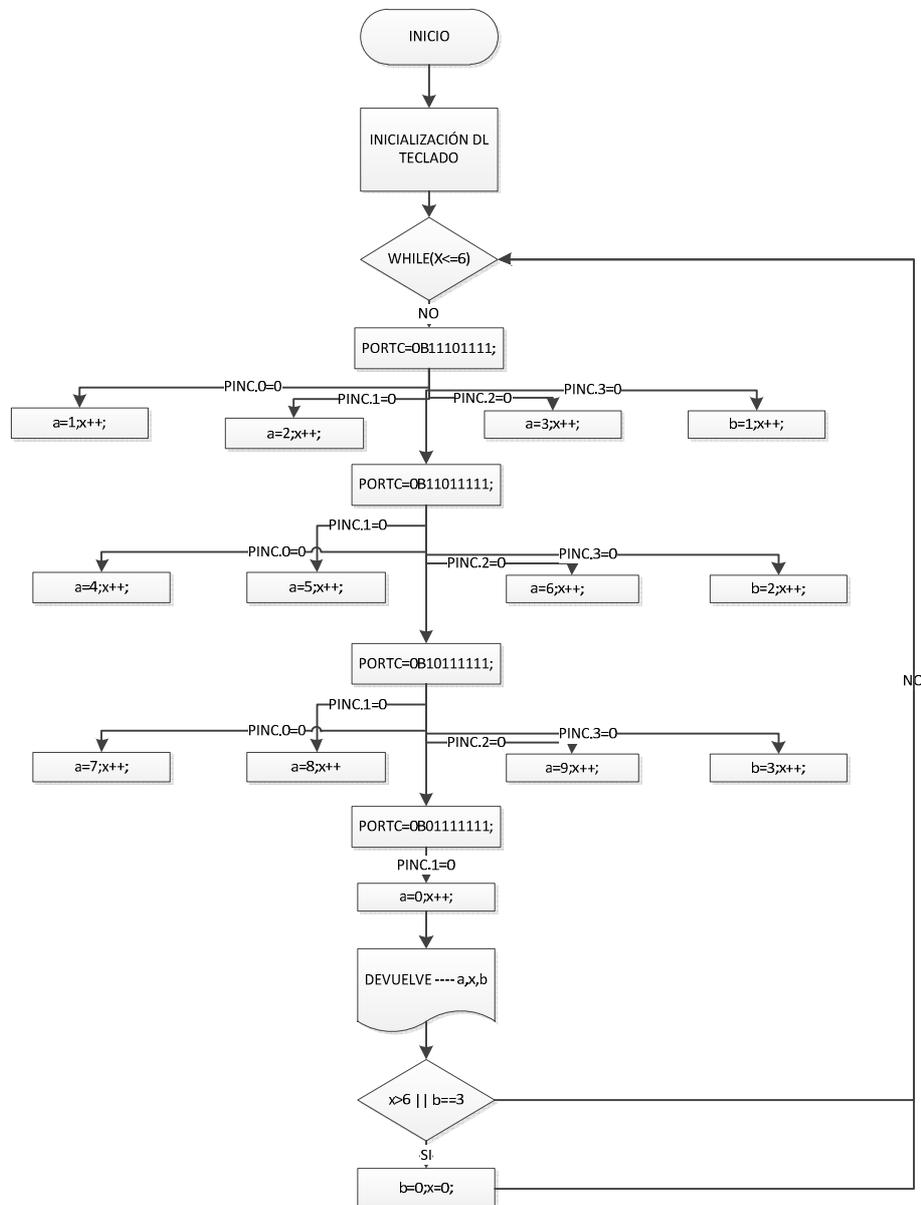


Fig. 3-7 Diagrama de flujo de la Librería para el manejo del teclado

3.4.2 PRÁCTICA DE CUENTA PERSONAS

Esta práctica consiste en contar el número de personas que van a ingresar, salir y las que se hallan actualmente dentro de un establecimiento, esta selección se efectúa por medio del teclado; el número de personas se visualiza en la GLCD; para lo cual se utiliza la activación de interrupción externas 0 y 2, cuando ingrese o salga alguien, con lo cual se interrumpe el normal proceso de programa y se ejecute la aplicación de suma o resta de ser el caso para la interrupción correspondiente y para obtener el número actual de personas dentro se llevara a cabo una resta entre el número de personas que entraron y las que salieron detectados por los sensores utilizados.

Para poder contar las personas que ingresaron y las que salieron se usa un elemento electrónico llama optoacoplador.

3.4.2.1 Optoacoplador

Un optoacoplador, también llamado optoaislador o aislador acoplado ópticamente, es un dispositivo de emisión y recepción que funciona como un interruptor influenciado mediante la luz emitida por un diodo LED que satura un componente optoelectrónico, normalmente en forma de fototransistor. De este modo se combinan en un solo dispositivo semiconductor, un foto emisor y un foto receptor, cuya conexión entre ambos es óptica.

Para la realización de la práctica se utiliza el optoacoplador de marca SHARP modelo GP3S62 que se puede observar en la figura No. 3-8



Fig. 3-8 Optoacoplador GP3S62

Este elemento electrónico está conformado por un fotodiodo que actúa como transmisor y un fototransistor que cumple la función de receptor. Como se puede visualizar en la figura No. 3-9

El fotodiodo como el fototransistor van a tener una conexión óptica, cuando algo obstruya entre la línea de vista de estos dos elementos, va a suceder que se saturará el transistor, y por ende el emisor va a salir cero lógico, lo que luego va a procesar el microcontrolador que determina la entrada o salida de alguien, todo esto depende del optoacoplador que este conectado a la interrupción correspondiente de salida o de entrada.

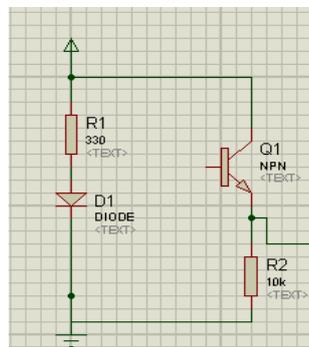


Fig. 3-9 Diagrama circuital del optoacoplador GP3S62

3.4.2.2 Diagrama del circuito

El diagrama circuital para esta práctica es la siguiente:

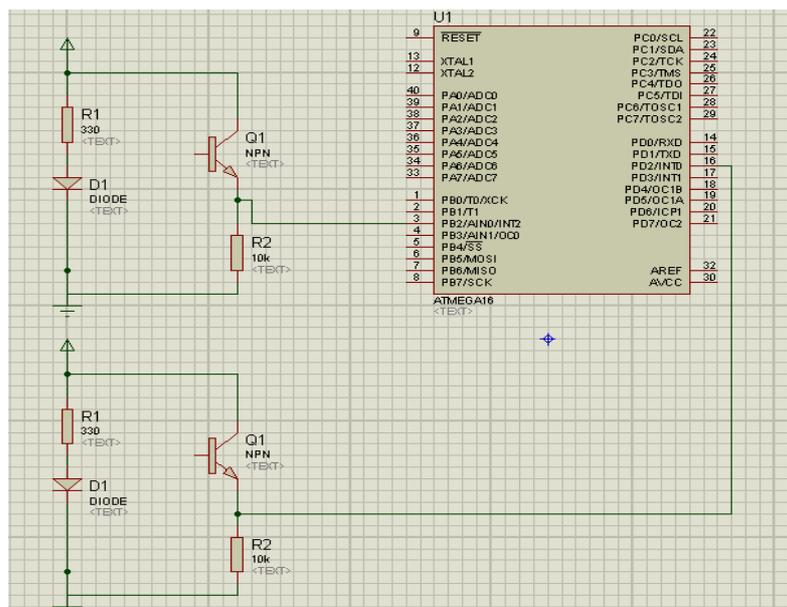


Fig. 3-10 Diagrama circuital de la práctica

Como se utiliza el dispositivo de desarrollo, lo único que necesitamos es conectar los optoacopladores al dispositivo, para lo cual se usa las interrupciones 0 y 2 que están libres, la interrupción 0 (PIND 16) va a contar las personas que entran y la interrupción 2 (PINB 2) van a contar las personas que salen.

3.4.2.3 Diagrama de flujo

En la figura No. 3-11 muestra el diagrama de flujo de la práctica de cuenta personas. El código se encuentra en el Anexo No. 3.

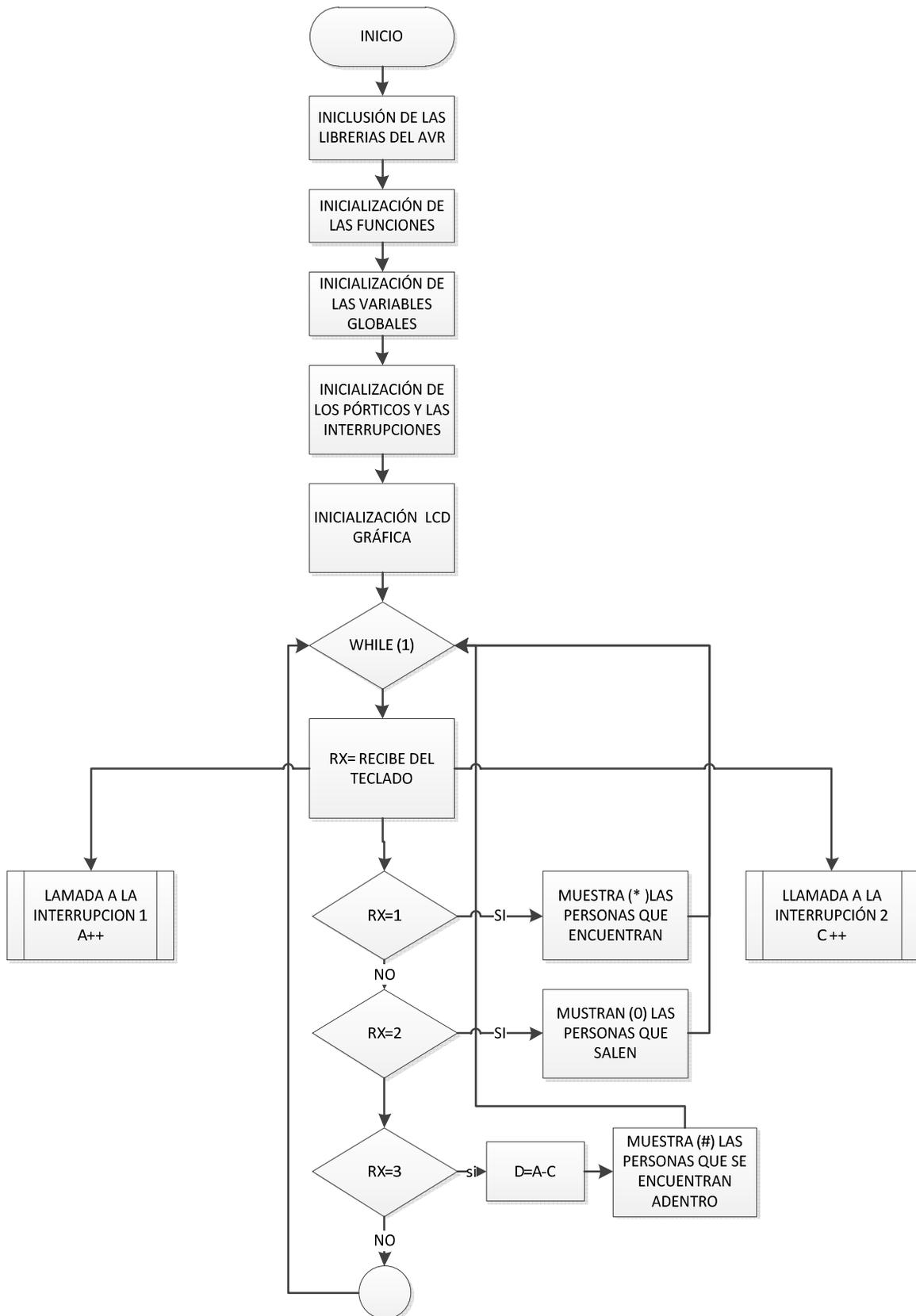


Fig. 3-11 Diagrama de flujo de la práctica cuenta personas

3.4.3 PRÁCTICA DE CERRADURA ELECTRÓNICA

En esta práctica utiliza el teclado 4x4 para introducir la clave, validarla o quitarla, se visualiza los números que se ingresa en la GLCD, como pantallas de acceso permitido y o de acceso denegado cuando ingresemos la clave mal.

En esta práctica no es necesario ningún elemento electrónico externo, para poder realizar la misma, es suficiente con la utilización del dispositivo de desarrollo.

A continuación se observa el diagrama de flujo de la práctica de la cerradura electrónica.

3.4.3.1 Diagrama de flujo

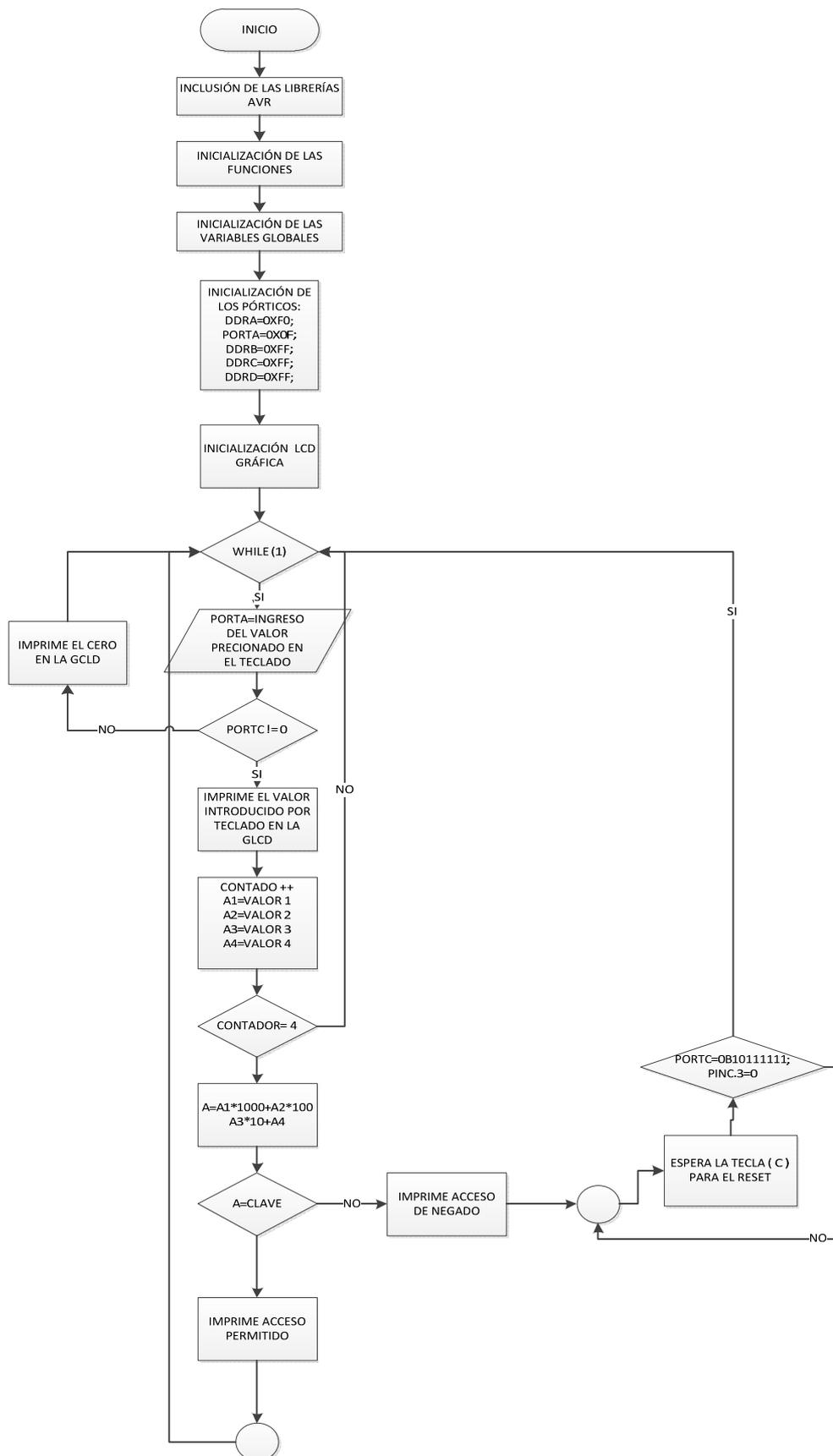


Fig. 3-12 Diagrama de flujo de la práctica de la Cerradura electrónica

3.4.4 PRÁCTICA DEL VOLTÍMETRO

En esta práctica utilizamos el conversor analógico digital de microcontrolador Atmega 164p, para la realización de un voltímetro digital. El funcionamiento de la práctica es la siguiente:

Se convierte la señal analógica ingresada por el PINA 1 de un divisor de voltaje formado por un potenciómetro variable, que nos permitirá comprobar la conversión de la señal analógica a una señal digital del voltaje en el potenciómetro que se visualiza en la GLCD.

3.4.4.1 Diagrama del circuito

El diagrama del circuito de la práctica esta conformado de un potenciómetro variable de 10K Ω que servirá de señal analógica, debido a que forma un divisor de voltaje que va ir cambiando de acuerdo a lo que varíe el valor de resistencia del potenciómetro. En la figura No. 3-13 se muestra la conexión del potenciómetro al ADC0 (PINA 0) del microcontrolador Atmega 164p.

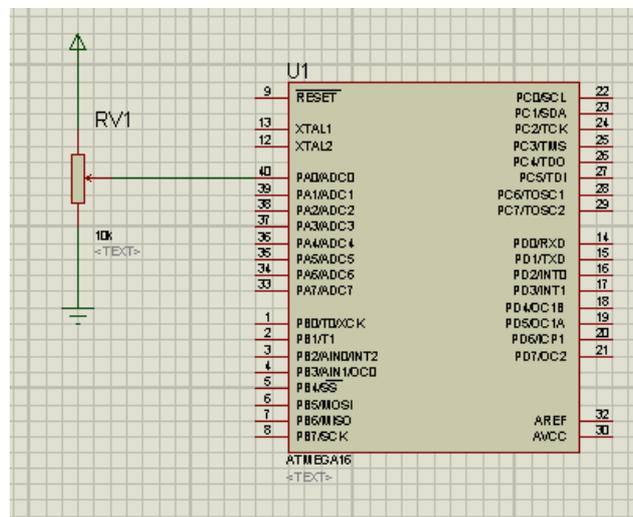


Fig. 3-13 Diagrama del circuito de la práctica del voltímetro

El diagrama de flujo de la práctica del voltímetro digital se observa a continuación.

3.4.4.2 Diagrama de flujo

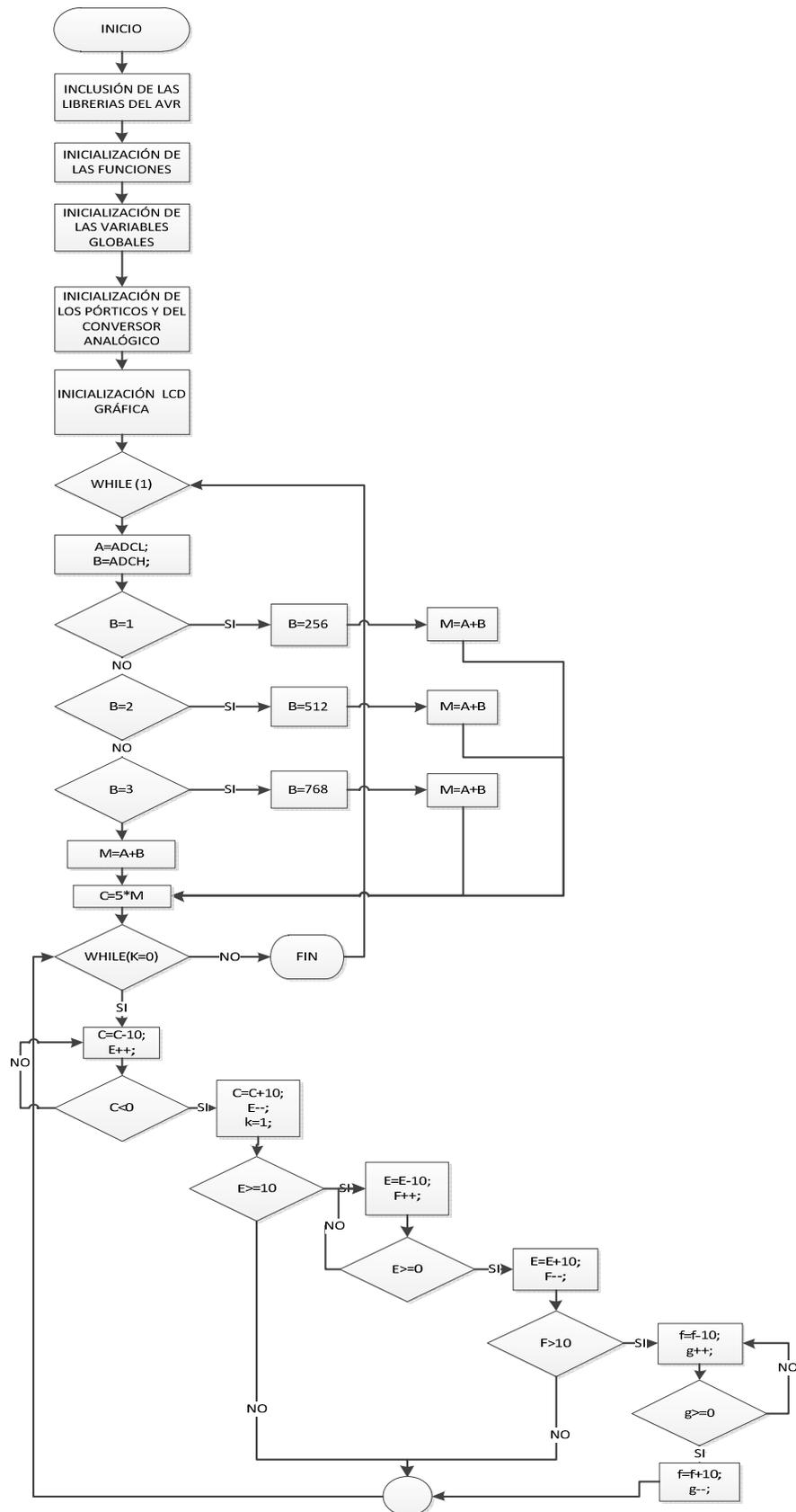


Fig. 3-14 Diagrama de flujo de la práctica del voltímetro

4. CAPÍTULO IV: PRUEBAS DE FUNCIONAMIENTO

Este capítulo trata sobre el funcionamiento y el desempeño del PDAT164P en la elaboración de las prácticas de laboratorio de Sistemas Microprocesados por parte del estudiante, para cual se explica el manejo del mismo y las consideraciones que debe tener.

4.1 PROGRAMA QUEMADOR

El software llamado QUEMADOR se desarrollo para que el estudiante tenga una interfaz amigable para la búsqueda y envío de la práctica hacia el PDAT164P desde la computadora.

En la figura No. 4-1 se puede observar la interfaz gráfica QUEMADOR

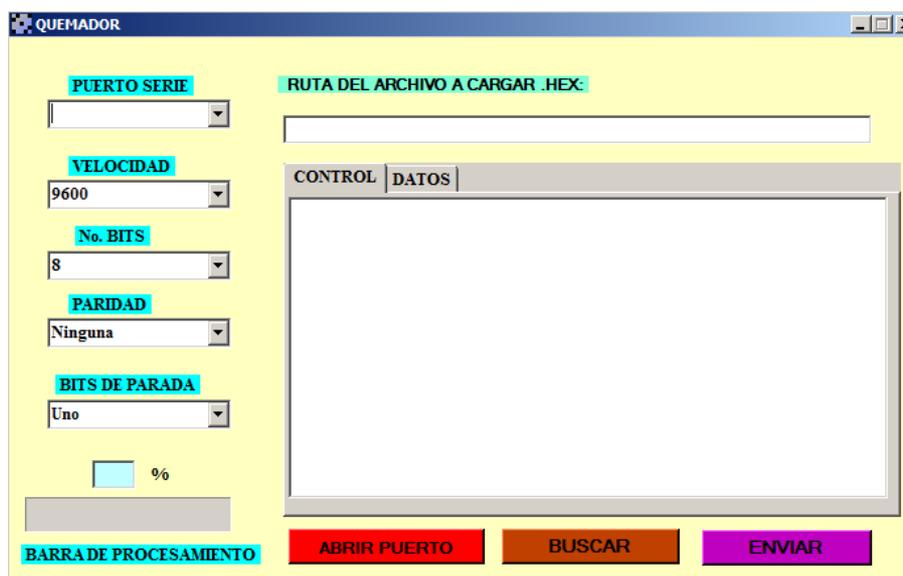


Fig. 4-1 Interfaz QUEMADOR

Esta interfaz contiene las siguientes opciones:

- Parámetros de la comunicación.
- Botón de abrir el puerto.
- Botón de Buscar.
- La ruta de Archivo a enviar.

- Pantallas de información.
- Botón de envió.
- Barra de procesamiento

4.1.1 PARÁMETROS DE LA COMUNICACIÓN

En esta opción escogemos los parámetros como son: el puerto serial, la velocidad de transmisión, números de bits de información, bits de paridad y bits de parada, ya que estos parámetros dependen de la configuración tanto del puerto asignado al mini Bluetooth para la selección del puerto serial, al igual que de la configuración del Usart del microcontrolador, estos parámetros son:

- COM16 (asignado al mini Bluetooth por la PC).
- Velocidad: 9.6 kbps.
- No. bits de parada: 8
- Paridad: Ninguna.
- Bits de parada: 1

En la figura No. 4-2 se puede observar la selección de los parámetros de la comunicación.

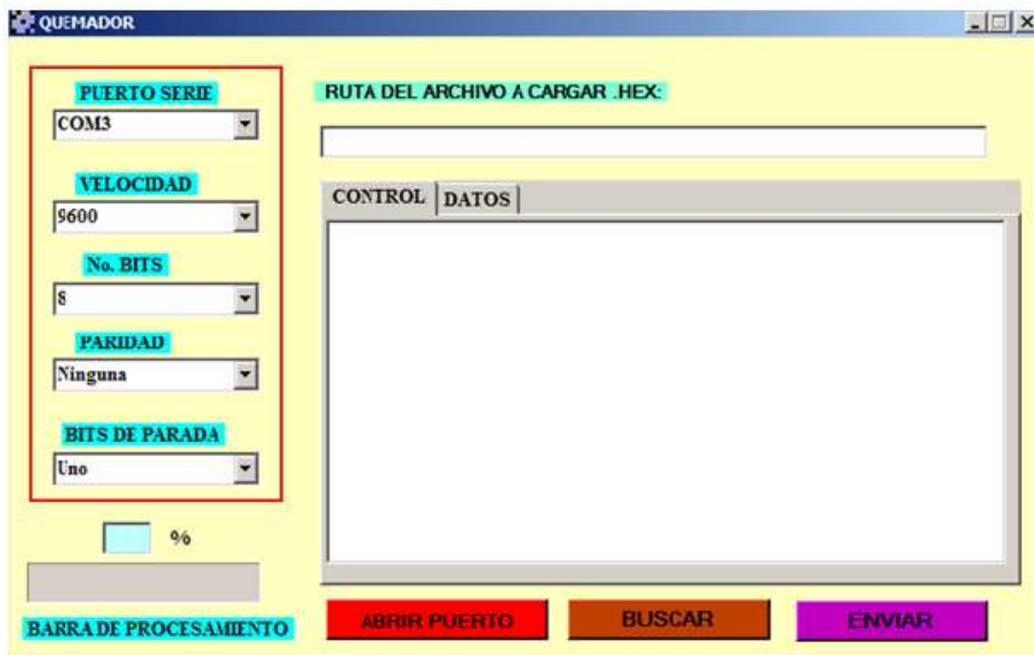


Fig. 4-2 Selección de los parámetros de la comunicación.

4.1.2 BOTÓN DE ABRIR PUERTO

Este botón sirve para que una vez seleccionados los parámetros de la comunicación se proceda a abrir el puerto por donde se efectuará la transferencia del archivo entre la computadora y el PDAT164P mediante comunicación inalámbrica, cabe recalcar que el dispositivo debe estar conectado a la fuente de voltaje.

En la figura No. 4-3 se puede observar como en la pestaña de control se despliega el mensaje se abrió el puerto COMXX, esto indica que ya se estableció la comunicación entre la PC y PDAT164P.

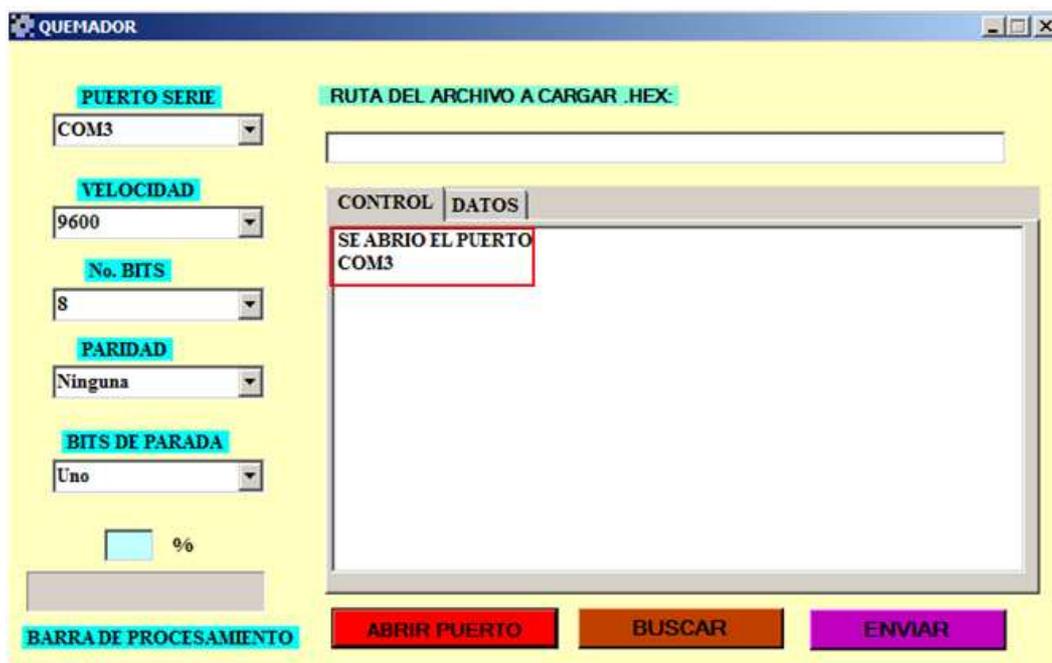


Fig. 4-3 Botón de abrir puerto

4.1.3 BOTÓN DE BUSCAR

El botón de buscar permite explorar en el computador el archivo de extensión .HEX que se desea descargar hacia el PDAT164P, para lo cual se despliega una ventana donde se puede hallar el archivo deseado.

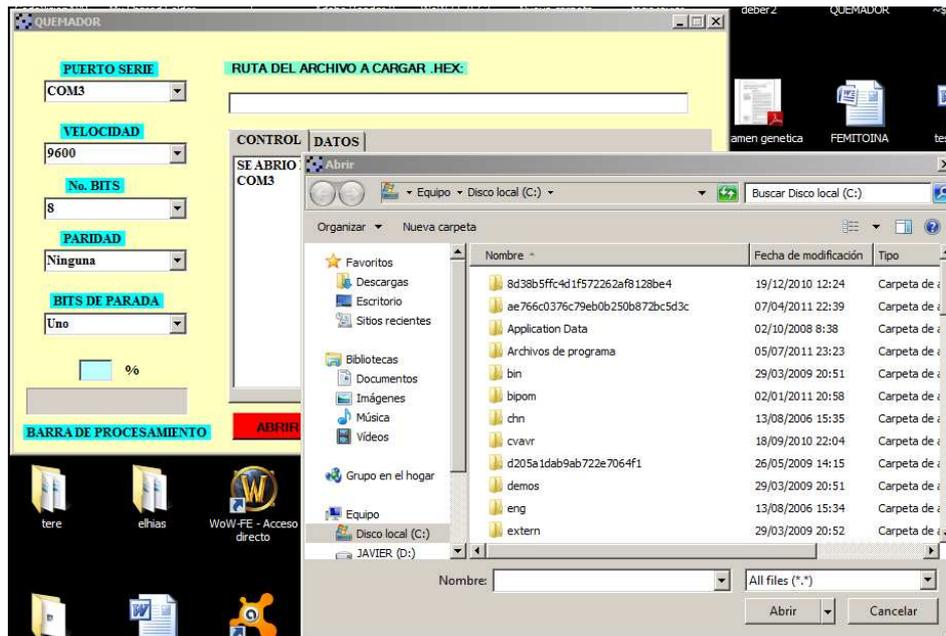


Fig. 4-4 Botón de explorar

4.1.4 RUTA DEL ARCHIVO

En la figura No. 4-5 podemos observar que en esta sección se puede ver la dirección donde está ubicado el archivo de extensión .HEX



Fig. 4-5 Ruta del archivo .HEX

4.1.5 PANTALLAS DE INFORMACIÓN

Esta es compuesta por dos pestañas llamadas de control y de datos.

4.1.6 BOTÓN DE ENVIO

Este botón sirve para iniciar la descarga del archivo de la práctica hacia el PDAT164P, antes se debe poner el interruptor en la posición de Bootloader en el dispositivo. (Ver figura No. 2-8).



Fig. 4-8 Botón de enviar

4.1.7 BARRA DE PROCESAMIENTO

Muestra el estado de la descarga del archivo de la práctica hacia el PDAT164P desde la computadora. (Ver figura No. 2-9).



Fig. 4-9 Barra de procesamiento

4.2 PRUEBAS DE DESCARGAS DE LAS PRÁCTICAS

Para probar el funcionamiento del PDAT164P se va a enviar las prácticas mediante comunicación inalámbrica, las cuales se realizaron en el capítulo anterior, para que mediante el Bootloader el microcontrolador se auto programe.

4.2.1 DESCARGA DEL PROGRAMA DE LA CERRADURA ELECTRÓNICA

Primero se debe conectar el PDAT164P a la fuente de voltaje, como se quiere descargar un programa el interruptor debe estar en la posición de Bootloader. (Ver figura No. 4-10).



Fig. 4-10 PDAT164P en la posición de Bootloader

Una vez que el Prototipo esta en la posición de Bootloader se utiliza el software QUEMADOR para la descarga de la práctica, como se explicó anteriormente.

En la figura No. 4-11 se observa la finalización de la descarga del programa de la cerradura electrónica.

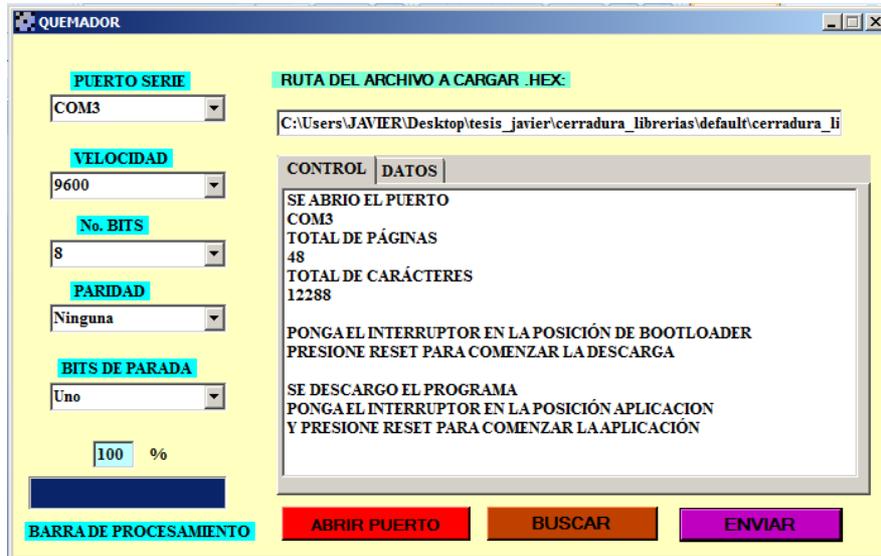


Fig. 4-11 Descarga de la práctica de la Cerradura electrónica

Como nos indica la pantalla de control para ir a la aplicación se debe poner el PDAT164P en la posición de "aplicación" y pulsar el reset.



Fig. 4-12 PDAT164P en la posición de Aplicación

4.2.2 DESCARGA DEL PROGRAMA DE CUENTA PERSONAS

Para descargar otro programa hacia el PDAT164P se carga el nuevo programa de la práctica, se coloca el interruptor en la posición de Bootloader y se presiona reset.

La figura No. 4-13 muestra la descarga del programa de cuenta personas.

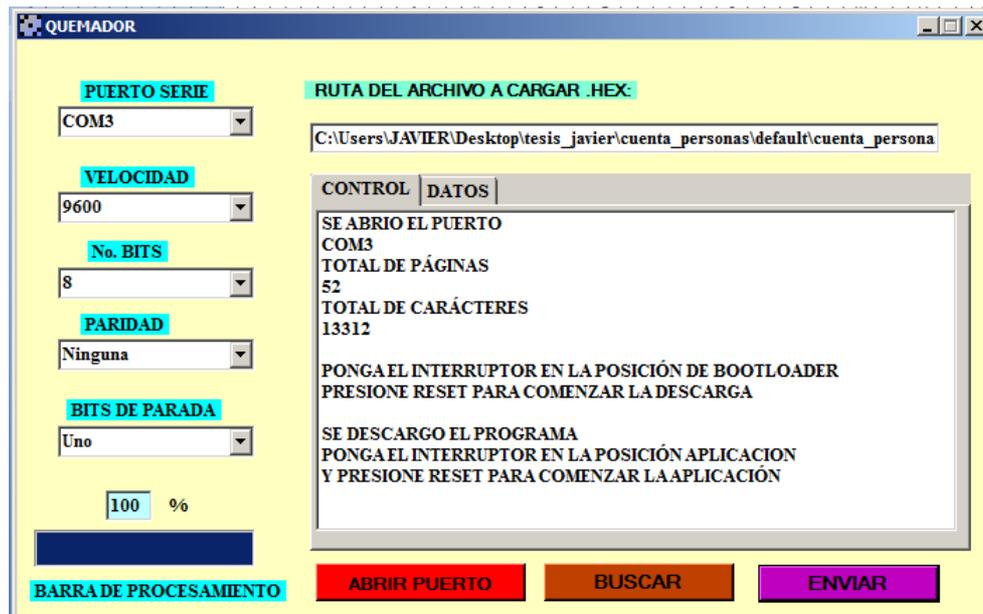


Fig. 4-13 Descarga de la práctica de cuenta personas

El PDAT164P permite obtener información de elementos externos, como es el caso de esta práctica volviéndose necesario el uso de los optoacopladores para realizar la interrupción y contabilizar la entrada y salida de personas. Para lo cual tiene dos pórticos para entrada y salida de información los pórticos A y B, como también las salidas de VCC y GND para circuitos externos.

En la figura No. 4-14 muestra la práctica de cuenta personas.



Fig. 4-14 Práctica de cuenta personas

4.2.3 DESCARGA DEL PROGRAMA DE VOLTÍMETRO DIGITAL

Igual que en los dos casos anteriores, se vuelve hacer lo mismo que ya se indicó anteriormente. Se pone en la posición de Bootloader luego se utiliza el software QUEMADOR y se descarga el programa de la práctica y finalmente en la posición de Aplicación y se presiona el botón de reset para que se ejecute el programa que esta cargado en la memoria.

En la figura No. 4-15 se muestra la práctica de voltímetro digital, igual que en la práctica de cuenta personas se necesita de un circuito externo que le de información al PDAT164P para que procese y muestre en la GLCD, en este caso el voltaje analógico, para que lo digitalice y lo muestre.

A continuación se muestra varias salidas de la práctica.



Fig. 4-15 Ejemplo 1 de la práctica voltímetro digital en el PDAT164P



Fig. 4-16 Ejemplo 2 de la práctica voltímetro digital en el PDAT164P



Fig. 4-17 Ejemplo 3 de la práctica voltímetro digital en el PDAT164P

5. CAPITULO V: CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Una vez terminado el proyecto se llega a concluir lo siguiente:

- ✓ Con el diseño y construcción del PDAT164P se soluciona la deficiencia de programadores en el Laboratorio de Sistemas Microprocesados y permite realizar las prácticas de una manera más eficiente, con lo cual el estudiante tiene un mayor desarrollo académico.
- ✓ Con la utilización del equipo el estudiante ya no necesita llevar elementos como son: microprocesador, programador, protoboard, etc., con lo cual puede enfocarse a nivel de software para desarrollar mejores sistemas inteligentes a nivel de programación. También van dirigidas a un beneficio social para los estudiantes del Laboratorio de Sistemas Microprocesados de la Escuela Politécnica Nacional, en relación al aspecto económico.
- ✓ El PDAT164P tiene la posibilidad de conectarse a otros dispositivos o circuitos electrónicos por los púrticos libres, cuando sea el caso de obtener información de sensores u otros dispositivos electrónicos, ampliando el aprendizaje del estudiante, y así el estudio no solo se enfoca a los elementos que lo constituye.
- ✓ Con el programa Gestor de Arranque que reside en la memoria de Bootloader, el microcontrolador se puede auto programar, en otras palabras el estudiante únicamente con la programación por una sola vez mediante un dispositivo ISP de este programa, ya no tiene la necesidad de utilizar otro programador, sino recurrir a la interfaz gráfica realizada en este proyecto para la descarga del programa y una comunicación serial RS-232 entre la PC y el microcontrolador.

- ✓ A través de la realización de este proyecto se creó un gestor de arranque (Bootloader), el cual se utiliza para auto programar el microcontrolador, pero la utilidad del mismo tiene una variedad de aplicaciones en el mundo de la electrónica, ya que permite la actualización de los sistemas controlados por un microprocesador como por ejemplo en la Domótica o cualquier sistema inteligente; sin necesidad de desmontar ningún dispositivo, sino mediante una comunicación inalámbrica modificar los sistemas.

5.2 RECOMENDACIONES

- ✓ Se recomienda que se fabrique mayor cantidad del PDAT164P para equipar el laboratorio de Sistemas Microprocesados con este prototipo, para que los estudiantes lo usen y puedan realizar las prácticas, para lo cual se adjunta el PCB del dispositivo y el código fuente del Gestor de arranque.
- ✓ Se recomienda realizar una práctica inicial para la utilización y manejo de la GLCD del equipo, para que sea la base de la realización de las demás prácticas actuales tales como: calculadora digital, cuenta personas, frecuencímetro, etc., además de la creación de una práctica serial inalámbrica, debido a que el dispositivo permite la realización de la misma.
- ✓ El PDAT164P fue desarrollado con el microcontrolador Atmega 164p, pero su aplicación de auto programación se puede extender a otros microcontroladores de la familia ATMEL, el cual puede ser tema de otros proyectos de titulación.
- ✓ Para bajar el costo del equipo en vez de realizar una comunicación vía Bluetooth entre la computadora y el PDAT164P, se puede utilizar, la descarga del programa de la práctica utilizando un interfaz RS-232, obteniendo el mismo resultado.

- ✓ Para la realización de las prácticas en la que se utilice la GLCD se recomienda el uso de un lenguaje de alto nivel como es C o Basic, ya que el manejo en un lenguaje como ensamblador sería tedioso y extenso para el estudiante.

REFERENCIAS BIBLIOGRÁFICAS

LIBROS:

- [1] BARRETT, Steven, PACK, Daniel, "Atmel AVR Microcontroller Primer: Programming and Interfacing", Morgan & Claypool Publisher series, USA, 2008.
- [2] PARDUE, Joe, "C Programming for Microcontrollers", Editorial Smiley Micros, Knoxville, 2005.
- [3] GRANIZO, Evelio, "Lenguaje C Teoría y Ejercicios", Editorial ESPE, Ecuador, 1999.
- [4] BLACO, Luis, "Programación en Visual Basic .NET", Grupos EIDOS, Madrid, 2002.
- [5] GRUNDGEIGER, Dave, "Programming Visual Basic .NET", O'Reille, USA, 2002.

REFERENCIAS ELECTRÓNICAS:

- [6] Atmel, "Hoja de datos de Atmega164p", 2009.
http://www.atmel.com/dyn/resources/prod_documents/doc7674.pdf
- [7] Atmel, "AVR106: C functions for reading and writing to Flash memory", 2006.
http://www.atmel.com/dyn/resources/prod_documents/doc2575.pdf
- [8] Atmel, "AVR: Self Programming", 2004.
http://www.atmel.com/dyn/resources/prod_documents/doc1644.pdf
- [9] Atmel, "Efficient C Coding for AVR", 2004.
http://www.atmel.com/dyn/resources/prod_documents/doc1497.pdf

[10] Atmel, "Atmel's Self-Programming Flash Microcontrollers", 2003.

http://www.atmel.com/dyn/resources/prod_documents/doc2464.pdf

[11] Rovinc Networks "Hoja de datos de RN-41", 2009

<http://www.rovingnetworks.com/documents/RN-41.pdf>.

[12] Visual-paradigm, "Programin in VB.NET"

<http://media.visual->

paradiqm.com/media/documents/dbva40npg/pdf/dbva_dotnet_programme_r_guide_chapter7.pdf

ANEXOS