

UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

Sistema de Seguridad manejable por Internet, con base en el Microcontrolador DS80C400

T E S I S

Para obtener el título de:
Ingeniero en Electrónica

Presenta:
Alberto de Jesús Díaz Ortiz

Director de Tesis:
M. C. Felipe Santiago Espinosa

Huajuapán de León, Oaxaca; Diciembre de 2009

AGRADECIMIENTOS

ÍNDICE

INTRODUCCIÓN.....	1
PLANTEAMIENTO DEL PROBLEMA.....	2
JUSTIFICACIÓN.....	4
OBJETIVOS.....	5
CONTENIDO.....	5
ANTECEDENTES Y FUNDAMENTOS.....	7
1.1 SISTEMAS DE SEGURIDAD.....	7
1.1.1 Sensores.....	9
1.2 SISTEMAS EMPOTRADOS.....	10
1.3 TARJETA DE EVALUACIÓN DEL MICROCONTROLADOR DE RED.....	13
1.3.1 El microcontrolador de red DS80C400.....	13
1.4 PROTOCOLOS TCP/IP.....	15
1.5 FAMILIA DE MICROCONTROLADORES AVR.....	17
1.5.1 El microcontrolador ATMEGA16.....	17
1.6 PROTOCOLO SPI.....	18
1.7 PROGRAMACIÓN EN JAVA Y LA PLATAFORMA TINI.....	23
1.7.1 Requisitos de la plataforma de desarrollo.....	24
1.7.2 Componentes de Software.....	24
1.8 LENGUAJE HTML.....	25
1.9 PROGRAMACION EN C.....	25
1.9.1 Consideraciones para el microcontrolador.....	26
DESARROLLO DEL HARDWARE.....	29
2.1 REQUERIMIENTOS DEL HARDWARE.....	29
2.2 DISEÑO.....	31
2.3 SUBSISTEMA DE INTERFAZ LOCAL.....	32
2.3.1 Teclado Matricial.....	32
2.3.2 Colección de Sensores y Actuador.....	33
2.3.3 Pantalla LCD.....	35
2.3.4 Módulo del Microcontrolador ATMEGA16.....	37
2.4 SUBSISTEMA DE INTERFAZ REMOTA.....	39
2.4.1 Módulo del microcontrolador DS80C400.....	39
2.4.2 Ethernet.....	42
2.5 INTEGRACIÓN DE LOS MÓDULOS DEL SISTEMA.....	43
DESARROLLO DEL SOFTWARE.....	47
3.1 REQUERIMIENTOS DEL SOFTWARE.....	47
3.2 DISEÑO.....	48
3.3 MÓDULO DEL MICROCONTROLADOR ATMEGA16.....	50
3.3.1 Archivo: tesis.hex.....	50
3.3.1.1 Se declara ATMEGA como esclavo.....	50
3.3.1.2 Se monitorea estado de sensores.....	52

3.3.1.3	¿Clave de acceso correcta?.....	53
3.3.1.4	Elige una opción.....	53
3.3.1.5	Solicita número de sensor, modo y tipo.....	54
3.3.1.6	Guarda el tipo y modo de un sensor.....	54
3.3.1.7	Tiempo de retardo.....	54
3.3.1.8	Solicita configuración para la sirena y la guarda.....	54
3.3.1.9	Muestra toda la configuración del sistema.....	55
3.3.1.10	Solicita tiempo y código.....	55
3.3.1.11	Guarda el valor del retardo y nuevo código.....	55
3.4	MÓDULO DEL MICROCONTROLADOR DS80C400	56
3.4.1	<i>Archivo TINIWebServer.tini</i>	<i>56</i>
3.4.1.1	Clase: EnviarSPI.....	57
3.4.1.2	Clase: WebWorker.....	57
3.4.1.3	Clase: TINIWebServer.....	58
3.4.2	<i>Hipervínculos Web</i>	<i>60</i>
3.4.2.1	Página: config.html.....	61
3.4.2.2	Página: requer.html.....	64
3.4.2.3	Pantalla: ayuda.html.....	64
3.4.3	<i>Manejo de datos en el Módulo del Microcontrolador DS80C400</i>	<i>67</i>
PRUEBAS Y RESULTADOS		71
4.1	ACCESO AL SISTEMA.....	71
4.1.1	<i>Acceso Local.....</i>	<i>72</i>
4.1.2	<i>Acceso Remoto</i>	<i>72</i>
4.2	CONFIGURACIÓN DE SENSORES.....	73
4.2.1	<i>Configuración Local.....</i>	<i>74</i>
4.2.2	<i>Configuración Remota.....</i>	<i>74</i>
4.3	TIEMPO DE RETARDO	75
4.3.1	<i>Asentamiento Local.....</i>	<i>75</i>
4.3.2	<i>Asentamiento Remoto.....</i>	<i>75</i>
4.4	ESTADO DE LA SIRENA	76
4.4.1	<i>Configuración Local.....</i>	<i>76</i>
4.4.2	<i>Configuración Remota.....</i>	<i>76</i>
4.5	ESTADO DEL SISTEMA	76
4.5.1	<i>Establecimiento Local.....</i>	<i>77</i>
4.5.2	<i>Establecimiento Remoto.....</i>	<i>78</i>
CONCLUSIONES		81
5.1	PERSPECTIVAS.....	82
REFERENCIAS		85
ANEXO A.	MANUAL DEL INICIALIZACIÓN DEL SISTEMA DE EVALUACIÓN DEL MICROCONTROLADOR DE RED DS80C400	89
ANEXO B.	CREACIÓN, DESCARGA Y EJECUCIÓN DE APLICACIONES TINI®.....	95
ANEXO C.	CREACIÓN Y DESCARGA DE ARCHIVOS .HEX AL ATMEGA16	99
ANEXO D.	CLASES GENERADAS PARA MANIPULAR A DISTANCIA EL SISTEMA DE SEGURIDAD	101

ANEXO E. PROGRAMA GENERADO PARA MANIPULAR LOCALMENTE EL SISTEMA DE SEGURIDAD	103
ANEXO F. HIPERVÍNCULOS WEB	105
ANEXO G. LISTA DE MATERIALES Y CIRCUITO IMPRESO	107

LISTA DE FIGURAS

Figura 1. Organización del Sistema de Seguridad.....	2
Figura 2. Módulos con los que cuenta el Sistema de Seguridad.....	3
Figura 1.1. Sistema básico de alarma.....	8
Figura 1.2. Ciclo de desarrollo de Sistemas Empotrados.....	12
Figura 1.3. Diagrama a bloques de la tarjeta de evaluación del DS80C400.....	14
Figura 1.4. Tarjeta de evaluación del microcontrolador de red DS80C400.....	14
Figura 1.5. Microcontrolador DS80C400.....	15
Figura 1.6. Arquitectura del protocolo TCP/IP.....	16
Figura 1.7. Pines del ATMEGA16.....	19
Figura 1.8. Protocolo SPI: comunicación maestro-esclavo.....	20
Figura 1.9. Transferencia de información: registro de desplazamiento.....	20
Figura 1.10. Registro para el control de la transferencia SPI.....	21
Figura 1.11. Modo de transferencia A para el protocolo SPI.....	22
Figura 1.12. Modo de transferencia B para el protocolo SPI.....	23
Figura 1.13. Modo de transferencia C para el protocolo SPI.....	23
Figura 1.14. Modo de transferencia D para el protocolo SPI.....	23
Figura 2.1. Diagrama a bloques del Sistema de Seguridad.....	31
Figura 2.2. Diagrama del teclado matricial.....	32
Figura 2.3. Teclado matricial a) vista superior y b) vista inferior.....	33
Figura 2.4. Conexión del sensor a) NA y b) NC.....	35
Figura 2.5. Pantalla LCD JHD 162A.....	35
Figura 2.6. Diagrama de conexión entre el ATMEGA16 y la pantalla LCD.....	36
Figura 2.7. Diagrama de conexiones del ATMEGA16 con el teclado matricial de 4x4.....	38
Figura 2.8. Diagrama del subsistema de interfaz local.....	39
Figura 2.9. Foto del subsistema de interfaz local.....	40
Figura 2.10. Microcontrolador DS80C400 de MAXIM/Dallas Semiconductor.....	41
Figura 2.11. Módulo del microcontrolador DS80C400.....	42
Figura 2.12. Módulo DSTINIs400: puerto Ethernet y SPI.....	42
Figura 2.13. Diagrama del Sistema de Seguridad.....	45
Figura 2.14. Hardware del Sistema de Seguridad.....	46
Figura 3.1. Diagrama a bloques del software del Sistema de Seguridad.....	49
Figura 3.2. Diagrama de flujo del Sistema de Seguridad.....	51
Figura 3.3. Función: void monitorear().....	52
Figura 3.4. Función: int clave().....	53
Figura 3.5. Programación del DS80C400.....	56
Figura 3.6. Pantalla principal del Sistema de Seguridad.....	61
Figura 3.7. Pantalla para la configuración del Sistema de Seguridad.....	62
Figura 3.8. Opción: Selecciona el Sensor - TIPO.....	63
Figura 3.9. Opción: RETARDO.....	63
Figura 3.10. Opción: SIRENA.....	64
Figura 3.11. Opción: ESTADO.....	64
Figura 3.12. Pantalla que muestra los requerimientos para la manipulación.....	66
Figura 3.13. Pantalla con información para comunicarse con el soporte técnico.....	66
Figura 3.14. Acceso al Sistema de Seguridad a distancia por medio de FTP.....	67

Figura 3.15. Ejemplo para añadir y eliminar el archivo <i>pueba.txt</i> del Sistema de Seguridad a distancia.....	68
Figura 3.16. Pantalla de Telnet ejecutando la aplicación del servidor Web: <i>TINIWebServer.tini</i>	69
Figura 4.1. Solicitud de claves de acceso de forma local.....	72
Figura 4.2. Cambio de clave de acceso de forma local.....	73
Figura 4.3. Solicitud de clave de acceso de forma remota.....	73
Figura 4.4. Configurar el modo de los sensores en forma local.....	74
Figura 4.5. Configurar el tipo de los sensores en forma local.....	74
Figura 4.6. Cambiar tiempo de retardo en forma local.....	75
Figura 4.7. Configurar el estado de la sirena en forma local.....	76
Figura 4.8. Cambios en el estado del sistema de forma local.....	77
Figura 4.9. Cambios en el estado del sistema: ACTIVADO a REPOSO.....	78
Figura 4.10. Cambios en el estado del sistema: ARMADO a REPOSO.....	78
Figura 4.11. Cambios en el estado del sistema: REPOSO a ARMADO.....	79
Figura A. 1.- Prompt del cargador JavaKit.....	91
Figura A. 2.- Instalación típica del hardware.....	92
Figura A. 3.- Salida al cargar el firmware.....	94
Figura B. 1.- Archivo fuente.....	95
Figura B. 2.- Load.cmd.....	97
Figura C. 1.- Programación en AVR Studio de <i>tesis.c</i>	99
Figura C. 2.- Descarga de <i>tesis.hex</i> al ATMEGA16 con PG4UW.....	100
Figura C. 3.- Parámetros para programar el ATMEGA16.....	100
Figura G. 1.- Impreso de la placa para la manipulación local.....	110
Figura G. 2.- Impreso de la placa para la manipulación local con cada uno de sus elementos.....	111
Figura G. 3.- Impreso de la placa para la manipulación remota.....	112
Figura G. 4.- Impreso de la placa para la manipulación remota con cada uno de sus elementos.....	113

LISTA DE TABLAS

Tabla 1.1. Sistemas de seguridad existentes en el mercado.....	9
Tabla 1.2. Modelo, características principales y costo de diferentes sensores.....	11
Tabla 1.3. Modos de transferencia SPI.....	22
Tabla 2.1. Modos de configuración para cada zona.....	30
Tabla 2.2. Tipos de usuario y sus privilegios.....	30
Tabla 2.3. Conexión de los pines del teclado matricial.....	33
Tabla 2.4. Características principales del detector de movimiento.	34
Tabla 2.5. Características principales del contacto magnético.	34
Tabla 2.6. Características principales de la sirena.....	34
Tabla 2.7. Conexión de cada uno de los sensores.....	35
Tabla 2.8. Conexión de los pines de la pantalla LCD con el ATMEGA16.	36
Tabla 2.9. Características del ATMEGA16.	37
Tabla 2.10. Conexión de los pines del puerto SPI entre el subsistema de interfaz local y el remoto.	43
Tabla 3.1. Descripción de las funciones del teclado matricial.	48
Tabla 3.2. Dato a enviar por el puerto SPI para cada opción dentro del acceso	65
Tabla 3.3. Comandos principales para ftp.....	68
Tabla 4.1. Estado del sistema antes y después de configurarlo.....	77
Tabla G. 1.- Lista de materiales empleados en el Sistema de Seguridad.	108

INTRODUCCIÓN

El presente trabajo de tesis consiste en el diseño de un Sistema de Seguridad con base en la plataforma de evaluación del microcontrolador de red DS80C400 y el microcontrolador ATMEGA16, para manipular a distancia la configuración y activación de una alarma, así como para monitorear su estado.

Un Sistema de Seguridad es un conjunto de dispositivos electrónicos colocados estratégicamente en el perímetro de un sitio específico para detectar la presencia, irrupción o invasión de un individuo que no cuente con permisos de acceso [URL 1].

Los sistemas de seguridad están integrados por: controlador, sensores, unidad de operación y dispositivos de salida. El controlador contiene un microprocesador, la batería de respaldo recargable y el transformador de alimentación para el sistema. Los sensores son para la protección del perímetro e interior. El teclado es llamado unidad de operación y sirve para comunicarse con el sistema. Los dispositivos de salida pueden ser sirenas (notificación local de alarma) o módem (para la notificación a la central de monitoreo) [1].

Los sistemas de seguridad comerciales cuentan con una gran variedad de características, pero al incremento de éstas su costo aumenta. Se pueden encontrar sistemas de seguridad que oscilan entre 100 y 500 dólares de las marcas PyRONIX [URL 2], SYSCOM (PIMA y HONEYWELL) [URL 3], STEREN [URL 4], entre otros; la diferencia de estos sistemas y el que se desarrolla en este trabajo es que los sistemas comerciales actuales no cuentan con acceso por Internet; aunque algunos de ellos tienen un comunicador telefónico o transmiten señales por radio, pero solo para la notificación de alarmas.

Con este sistema se puede controlar la seguridad de un área en específico, como casa u oficina, ya sea de forma local o remota; disminuyendo costos y manteniendo las características principales de los sistemas de seguridad ya existentes, con la diferencia del acceso remoto. En la figura 1 se muestra la organización del sistema.

El Sistema de Seguridad soporta sensores de presencia, de humo, de apertura de puertas o ventanas y de ruptura de cristales. Todos estos elementos forman una red que está conectada al microcontrolador ATMEGA16 por sus características de CPU, memoria y puertos de entrada/salida; el cual hará un monitoreo constante del estado de cada uno de los sensores y se comunicará con la plataforma de evaluación del microcontrolador DS80C400 para la interacción de forma remota del usuario con el sistema, por medio del protocolo TCP/IP incluido en éste último microcontrolador.

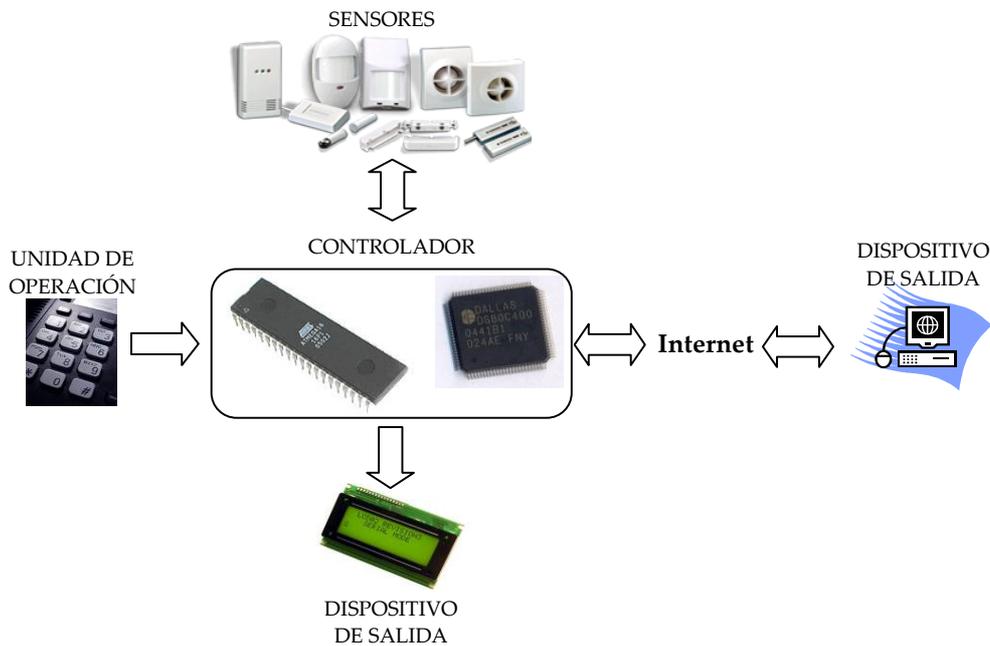


Figura 1. Organización del Sistema de Seguridad.

Una ventaja importante del Sistema de Seguridad aquí desarrollado es la conexión a Internet, lo que requiere de la generación de una página Web que sirva como conexión entre la plataforma de evaluación del microcontrolador de red y el usuario ubicado en cualquier parte del mundo (siempre y cuando tenga acceso a Internet). El microcontrolador ATMEGA16 sirve para manipular al sistema localmente, y lo consigue por medio del teclado y la pantalla de cristal líquido (LCD, *Liquid Crystal Display*) conectados a dos de sus cuatro puertos.

En forma local o remota se puede cambiar la configuración de los sensores, agregar, eliminar o modificar claves de acceso y activar o desactivar el sistema.

PLANTEAMIENTO DEL PROBLEMA

En la actualidad Internet es el medio de comunicación más importante, por lo que a medida que pasa el tiempo, las personas lo utilizan para aplicaciones que satisfagan sus necesidades y con ello generar un estilo de vida más cómodo.

Haciendo caso a la necesidad de utilizar la red para disminuir el esfuerzo al momento de realizar alguna actividad, se desarrolló un Sistema de Seguridad para un área en específico, cuya manipulación puede realizarse por medio de una página Web. Con esta

interfaz es posible modificar los diferentes parámetros de activación, desactivación y funcionamiento del sistema.

El Sistema de Seguridad se apoya en diferentes sensores y la alteración de cualquiera de éstos activa la alarma que inmediatamente produce un mensaje de alerta a través de la página Web, al mismo tiempo que hace sonar una sirena ubicada dentro del área en donde el Sistema de Seguridad está ubicado.

Se cuenta principalmente con los siguientes módulos: colección de sensores y actuador, plataforma de evaluación del microcontrolador de red, microcontrolador ATMEGA16, interfaz física e interfaz gráfica. En la figura 2 se muestra la interacción entre éstos módulos.

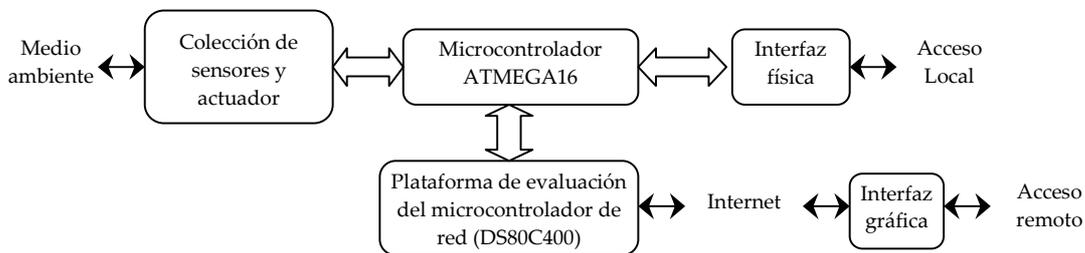


Figura 2. Módulos con los que cuenta el Sistema de Seguridad.

Colección de sensores y actuador

Los diferentes sensores utilizados son de presencia, humo, apertura de puertas o ventanas y ruptura de cristales; todos con forma similar de funcionamiento. Los sensores funcionan como interruptores y pueden ser normalmente abiertos (NA) o normalmente cerrados (NC). Dependiendo de su configuración, la activación de alguno de estos sensores accionará al actuador, que en este caso es una sirena, la cual sonará o no dependiendo de su configuración.

Microcontrolador ATMEGA16

Este módulo tiene la capacidad de cambiar la configuración de los diferentes sensores, agregar, eliminar o modificar claves de acceso y activar o desactivar al Sistema de Seguridad. Para lo anterior cuenta con un dispositivo local.

Dispositivo local

Medio por el que el usuario manipula de forma local al sistema: teclado y pantalla LCD. El teclado para la manipulación del sistema y la pantalla LCD para la observación de la información.

Plataforma de evaluación del microcontrolador de red (DS80C400)

Establece la comunicación a distancia entre el Sistema de Seguridad y el usuario, a través de Internet y por medio de una interfaz gráfica (página Web). La página Web está contenida dentro del microcontrolador DS80C400 a la que se tendrá acceso por medio de la IP asignada a éste.

Interfaz gráfica

La configuración, activación y desactivación de la alarma se puede controlar a través de una página en Internet por medio de una clave de acceso. En esta página se muestra el estado de los sensores y se cambia su configuración, se activa o desactiva la alarma y se agregan, eliminan o modifican claves de acceso; todo en tiempo real.

JUSTIFICACIÓN

En la carrera de Ingeniería en Electrónica existente en la Universidad Tecnológica de la Mixteca (UTM), los alumnos obtienen diferentes conocimientos que sirven para la superación personal y profesional, uno de ellos es el manejo de dispositivos programables, como los microcontroladores y otro, el control de diversos sistemas; ambos están muy relacionados y su implementación en la actualidad y en la vida real tiene mucha importancia.

El problema que en el presente trabajo se resolvió, surgió por la necesidad de manipular a distancia el control del Sistema de Seguridad de un área en específico.

Con éste sistema se tiene control de la activación y desactivación de la alarma y configuración de los sensores desde cualquier punto en donde se cuente con una conexión a Internet. En la actualidad es fácil tener acceso a una conexión a Internet, y por consiguiente, al sistema de alarma a distancia. Y si por algún motivo no se tuviera acceso a Internet, se mantiene la posibilidad de manipular al sistema en forma local.

El sistema le proporciona al usuario facilidades para:

- Vigilar la entrada y salida de personal no autorizado a algún área en específico.
- Notificar acerca de acontecimientos anormales dentro del área asegurada.
- Trabajar en un sistema de red.
- Realizar su control en forma local o remota.
- Garantizar que solo usuarios autorizados podrán manipularlo, con el manejo de claves de acceso.

La importancia de elaborar éste sistema radica en que se plantea un Sistema de Seguridad con otros mecanismos de acceso, pudiendo sustituir a los sistemas existentes.

OBJETIVOS

El objetivo general del presente trabajo fue el diseño y construcción de un Sistema de Seguridad para un área en específico a partir de la plataforma de evaluación del microcontrolador de red DS80C400. La alarma se manipula dentro de su área de acción o bien a distancia por medio de una página Web, para su control se utilizan claves de acceso.

El Sistema de Seguridad es de 6 zonas, es decir que se pueden conectar seis sensores: de presencia, de humo, magnéticos, de ruptura de cristales o una combinación de estos. Para conseguir el buen funcionamiento de todo el conjunto de elementos que componen al sistema, se persiguieron los siguientes objetivos particulares:

- Acondicionar los módulos necesarios para la configuración y control local, monitoreo de sensores y activación del actuador. Todos ellos gobernados por el microcontrolador ATMEGA16.
- Establecer los mecanismos para intercambiar información entre la plataforma de evaluación del microcontrolador DS80C400 y el microcontrolador ATMEGA16.
- Monitorear el estado del sistema por medio de una página Web.
- Desarrollar un programa en Java que permita al usuario configurar el modo de los sensores, modificar claves de acceso y generar alertas a la alteración del estado de la alarma. Todo podrá realizarse en forma local y remota.
- Exponer al sistema frente a diferentes casos y observar su comportamiento.
- Documentar los resultados

CONTENIDO

El presente documento se divide en una introducción y cinco capítulos, en cada capítulo se desarrollan temas específicos en donde se explican los procedimientos realizados para cumplir con los objetivos propuestos.

Introducción.- Plantea el problema, la justificación e importancia, y los objetivos establecidos.

Capítulo 1, Antecedentes y Fundamentos.- Proporciona una descripción de los aspectos necesarios para comprender el funcionamiento de los sistemas de seguridad junto con los diferentes dispositivos conectados al mismo, también hace referencia a las técnicas y dispositivos utilizados para conseguir los objetivos previstos.

El capítulo 2, Desarrollo del Hardware.- Detalla el Hardware necesario para implementar al Sistema de Seguridad, tanto para el subsistema de interfaz local como para el subsistema de interfaz remota, y la forma en que interactúan ambos subsistemas.

Capítulo 3, Desarrollo del Software.- Se ocupa de la descripción del software, explicando el desarrollo de los programas necesarios en el diseño del Sistema de Seguridad. Se incluye la forma en que se desarrollan las funciones involucradas en los programas para el control local y clases de java, junto con las páginas HTML, para el acceso en forma remota.

Capítulo 4, Pruebas y Resultados.- Se expone el Sistema de Seguridad a todos los posibles casos de aplicación con los que se puede enfrentar, se comprueba su funcionamiento con la posibilidad de introducirlo al mercado.

Capítulo 5, Conclusiones.- Muestra los resultados de éste proyecto, las recomendaciones y trabajos futuros.

1

ANTECEDENTES Y FUNDAMENTOS

Este capítulo define lo que es un Sistema de Seguridad y cada uno de los elementos que lo integran; enuncia los tipos de sensores, al igual que los precios y marcas existentes en el mercado. Se explican las características de los dispositivos utilizados en el sistema para lograr los objetivos.

Además, se presenta una introducción a los lenguajes de programación utilizados y la forma en que interactúa el software con el hardware en un sistema empotrado.

1.1 SISTEMAS DE SEGURIDAD

Un Sistema de Seguridad es un elemento pasivo. Esto significa que no evitan una intrusión, pero sí son capaces de advertir de ella, cumpliendo así, una función disuasoria

frente a posibles intrusos. Son capaces además de reducir el tiempo de ejecución de la intrusión, minimizando de esta forma las pérdidas.

La función principal de los Sistemas de Seguridad es advertir el allanamiento en una vivienda o inmueble. Los equipos de alarma pueden estar conectados a una Central Receptora (Central de Monitoreo), con el propietario mismo a través de teléfono o TCP/IP o simplemente cumplir la función disuasoria, activando una sirena [URL 5].

Un Sistema de Seguridad se compone de diferentes dispositivos conectados a la Central de Monitoreo: teclado, detectores de humo, sirena, detectores de presencia, detectores magnéticos y la conexión por teléfono o TCP/IP. La figura 1.1 muestra un sistema básico de alarma [URL 2].

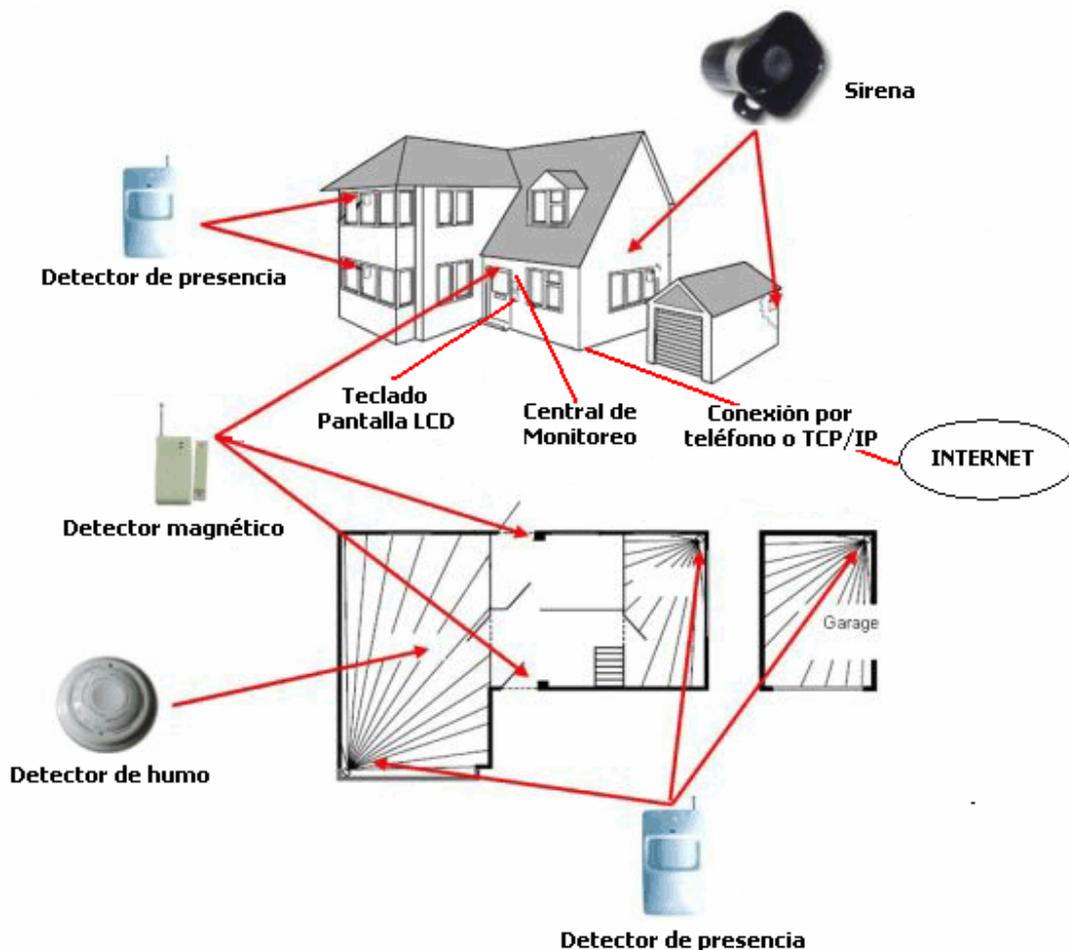


Figura 1.1. Sistema básico de alarma.

La central de monitoreo es el controlador principal. El teclado proporciona una forma fácil y conveniente de activar y desactivar el Sistema de Seguridad; es el medio de

comunicación entre el usuario y los sensores (detectores). La sirena es una alerta audible que avisa en caso de una emergencia. La conexión por teléfono o Ethernet es la interacción entre el sistema y el exterior [URL 3]. Los sensores se detallarán en párrafos posteriores.

Algunos Sistemas de Seguridad existentes en el mercado se muestran en la tabla 1.1, con sus correspondientes características y costos [URL 2] [URL 3] [URL 4]. Todos los sistemas de seguridad existentes en la actualidad no cuentan con una conexión IP, aunque cuentan con una conexión por radio o teléfono hacia la estación receptora para avisar al momento de detectar alguna diferencia en el estado de sus sensores. Aún con la conexión por radio o teléfono, no controlan a distancia el sistema ni cambian la configuración de sus elementos.

MODELO (MARCA)	CARACTERÍSTICAS PRINCIPALES	COSTO (Dls)
VISTA4864SP (HONEYWELL)	Panel de 8 zonas. Retardo para la activación de la alarma. Niveles de permisos por usuario. Contiene panel de alarma y teclado alfanumérico.	\$185.00
VISTA50P (HONEYWELL)	Panel de 9 zonas. Programación de eventos. Niveles de permiso por usuario. Contiene panel de alarma y transformador.	\$347.00
CAPTAINRX406 (PIMA)	Panel de 6 zonas. Niveles de permiso por usuario. Programación por menús. Contiene panel de alarma y teclado de LED's.	\$123.00
HUNTERPRO (PIMA)	Panel de 9 zonas. Niveles de permiso por usuario. Programación por menús. Solo contiene panel de alarma.	\$126.00
ALA-110 (STEREN)	Panel de 4 zonas. Contiene panel de alarma, un sensor de presencia, una sirena externa y 3 contactos magnéticos.	\$150.00

Tabla 1.1. Sistemas de seguridad existentes en el mercado.

1.1.1 Sensores

Un sensor es un dispositivo que detecta manifestaciones de fenómenos físicos; está diseñado para recibir información del exterior y transformarla en una magnitud, normalmente eléctrica, que sea factible cuantificar y manipular por un sistema electrónico. Pueden ser de indicación directa o estar conectados a un indicador (posiblemente a través de un convertidor analógico a digital, una computadora y un display) de modo que los valores muestreados puedan ser leídos por un humano.

Los sensores típicos para un Sistema de Seguridad son:

- **De presencia.-** Este tipo de sensores tienen como finalidad determinar la presencia de un objeto en un intervalo de distancia especificado. Algunos de los tipos más importantes son: inductivos, de efecto Hall, capacitivos, ultrasónicos e infrarrojos. Su salida puede ser normalmente abierta (NA) o normalmente cerrada (NC). El principio de funcionamiento está basado en la diferencia (moderada) entre la temperatura del ambiente y la temperatura del cuerpo humano. Cuando la temperatura del cuerpo humano difiere de la temperatura del ambiente, la radiación generada por el intruso es empleada para activar una alarma [URL 6].
- **De humo.-** Estos sensores detectan el humo dentro de un área en específico y cambian su estado ante la presencia de éste. Entre sus principios de funcionamiento existen dos tipos fundamentales, los de cámara de ionización y los fotoeléctricos. El primero detecta la descompensación de conductividad producida por el humo entre dos células: la de medida y la de referencia. El segundo detecta radiaciones provenientes de las llamas [URL 7].
- **De apertura de puertas o ventanas.-** Este tipo de sensores consisten en un contacto magnético. El contacto magnético consta de un imán y normalmente el imán cierra el contacto con el sensor. Si se abre una puerta o una ventana, el contacto se abrirá y enviará una señal de alteración en el estado del sensor.
- **De ruptura de cristales.-** Existen dos clases de sensores para la ruptura de cristales. El primero detecta la vibración en el vidrio y el segundo el sonido que produce un cristal al romperse; este último es recomendable porque hay técnicas especiales para quebrar un vidrio sin producir impacto.

La tabla 1.2 muestra los modelos, costos y características principales de los sensores a utilizar en un Sistema de Seguridad [URL 2] [URL 3] [URL 4].

1.2 SISTEMAS EMPOTRADOS

Los sistemas empotrados (*Embedded Systems*) son sistemas electrónicos que realizan una tarea específica bajo la supervisión de un microprocesador y un programa computacional. Son sistemas en tiempo real que realizan funciones de control, procesamiento y/o monitorización [2].

Un sistema empotrado es una mezcla de hardware y software que constituye un componente dentro de un sistema más complejo y se espera que funcione sin intervención humana. A grandes rasgos, el ciclo de desarrollo de un sistema empotrado se expone en la figura 1.2 [URL 9].

SENSOR	MODELO (MARCA)	CARACTERÍSTICAS PRINCIPALES	COSTO (Dls)
De Presencia (Infrarrojo)	998MX (HONEYWELL)	Cobertura de 15x15 m. Incluye filtro para mascotas y largo alcance. Ajuste de padrón vertical y horizontal. Compensación de temperatura automática.	\$ 59.00
	QUEST2260SN (HONEYWELL)	Cobertura de 18.3x22.8 m. Ahorra corriente en instalaciones grandes. Busca perturbaciones ambientales. Modo de sensor inteligente.	\$ 153.00
De Humo	SLR1224BH4W (HOCHIKI)	Detector fotoeléctrico de humo y temperatura fija. Instalación de 4 hilos. Límite de temperatura de 57.2° C	\$ 57.00
	4WTB (SYSTEM SENSOR)	Detector fotoeléctrico de humo. Instalación de 4 hilos. Límite de temperatura de 57° C.	\$ 59.00
De Apertura de puertas y ventanas (Magnéticos)	4959SN (HONEYWELL)	Cubierta de aluminio rugoso e imán con soporte tipo "L" ajustable. Espacio vacío de 5.1 cm (2"). Sellado con sustancias epóxicas para protección contra calor, humedad e impacto.	\$ 32.00
	4939SNGR (HONEYWELL)	Contacto de identificación de punto/direccionable para montaje sobre superficies. Espacio vacío de 3.18 cm (1.25"). Sellado con sustancias epóxicas para protegerlo de la humedad e impacto.	\$ 20.00
De Ruptura de Cristales	ASCS1 (HONEYWELL)	Sensor de audio de ruptura de cristal. Cobertura de 2.4 m en cualquier tipo de cristal. Grosor máximo del cristal 1 cm.	\$ 13.00
	FP05101 (PYRONIX)	Detector microfónico microprocesado con detección acústica. Alcance de 16 m con una cobertura de 360°.	\$ 31.00
	2100EX (HONEYWELL)	Detector dual de audio y vibración Audio detector con cobertura de 7.6 m. Ajuste de sensibilidad de audio y vibración.	\$ 43.00

Tabla 1.2. Modelo, características principales y costo de diferentes sensores.

- **Especificaciones.-** Consiste en realizar un modelo del sistema completo, incluyendo un conjunto de propiedades, una descripción funcional y un conjunto de restricciones.
- **Formato Intermedio.-** Para sistemas complejos, consiste en la partición del sistema en varias funciones de menor complejidad, con el propósito de abordar el problema completo como varios problemas desconectados entre sí.

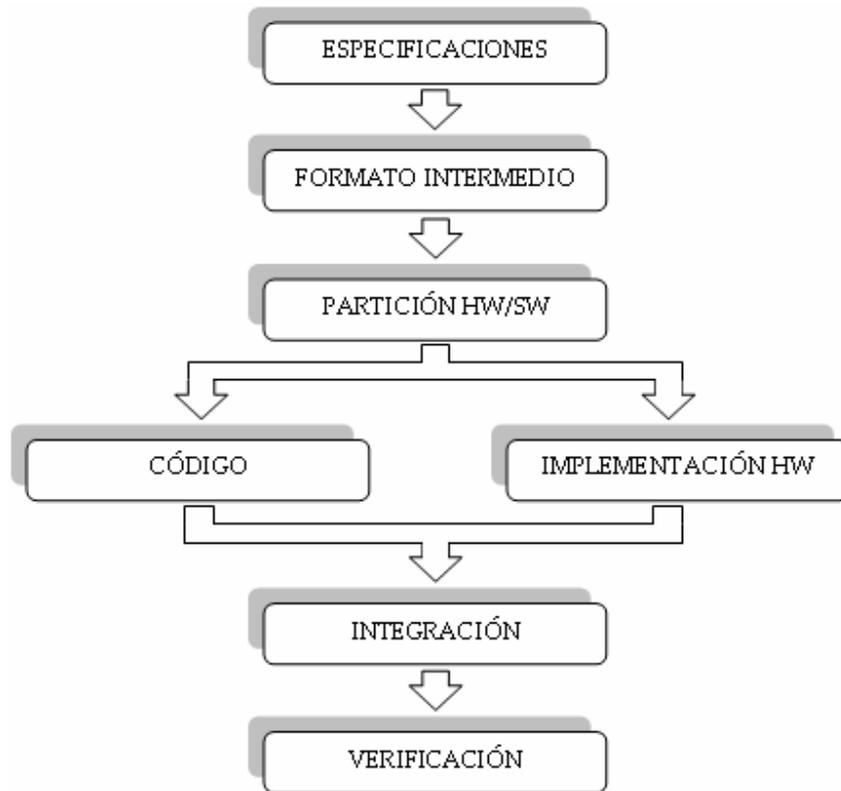


Figura 1.2. Ciclo de desarrollo de Sistemas Empotrados.

- **Partición SW/HW.-** Determina qué funciones se realizarán mediante dispositivos de aplicación específica (hardware) o mediante programación (software).
- **Integración.-** Integra todos los elementos del sistema, de tal manera que todos los dispositivos realicen sus funciones de forma correcta, para que el flujo de datos esté de acuerdo con la operación a realizar.
- **Verificación.-** Realiza una descripción funcional del sistema, esta mediante la realización de pruebas, de las cuales ya se espera un resultado específico, de esta manera se verificará la funcionalidad del sistema.

Este trabajo de tesis tiene sus bases en los sistemas empujados debido a que se manipulan dos sistemas (hardware): tarjeta de evaluación del microcontrolador de red y tarjeta de interfaz local con el microcontrolador de la familia AVR; por medio de diferentes programas (software) escritos en lenguaje JAVA, C y HTML. El software y hardware mencionados se describen a detalle en secciones posteriores.

1.3 TARJETA DE EVALUACIÓN DEL MICROCONTROLADOR DE RED

La tarjeta de evaluación del microcontrolador de red diseñada por Dallas Semiconductor, es una plataforma con medios simples y flexibles para lograr que una amplia variedad de dispositivos se puedan conectar directamente a la red a través de diversos puertos: SPI, I/O, I2C y CAN. Con esta tarjeta se pueden desarrollar aplicaciones para la mayoría de las funciones del microcontrolador DS80C400 [URL 10].

Las principales características de la tarjeta de evaluación del microcontrolador de red son las siguientes:

- Ambiente de desarrollo denominado Interfaz Pequeña de Internet (TINI®, *Tiny Internet Interface*) basado en el lenguaje de programación Java, incluye rutinas para conexión a Internet y validar el diseño del hardware.
- Reloj en tiempo real.
- Dispositivo 1-Wire.
- Controlador de acceso a medios 10/100 Mbps.
- Comunicación serial, basada en el estándar RS-232
- Protocolo serial de bajo nivel, TTL.
- Un puerto CAN 2.0 y tres puertos seriales.
- Puertos I2C y SPI.
- Memoria de programa FLASH de 1 Mb.
- Memoria de datos SRAM no volátil de 1 Mb.

En la figura 1.3 se muestra el Diagrama a bloques de la tarjeta de evaluación del microcontrolador de red en donde el DS80C400 ejecuta directamente el código nativo del entorno de ejecución (*runtime environment*), proporciona el control y la comunicación entre dispositivos y las capacidades para el establecimiento de una red. Las memorias de los datos y de código almacenan los datos del sistema y el entorno de ejecución. El dispositivo decodificador de Ethernet auxilia al microcontrolador para enviar y recibir mensajes a través de la red y los elementos periféricos.

El sistema físico de la tarjeta de evaluación del microcontrolador de red se muestra en la figura 1.4.

1.3.1 El microcontrolador de red DS80C400

El DS80C400 es un microcontrolador de red de 100 terminales (figura 1.5), su controlador de acceso a medios de Ethernet le permite el acceso y comunicación sobre Internet, soportando conexiones TCP simultáneas y transferencias hasta de 5 Mbps. Con la conexión a una red y capacidades de entrada/salida es adecuado para servir como una unidad central de proceso en la red [3] [URL 11]. Puede controlar activamente redes de menor jerarquía con hardware dedicado.

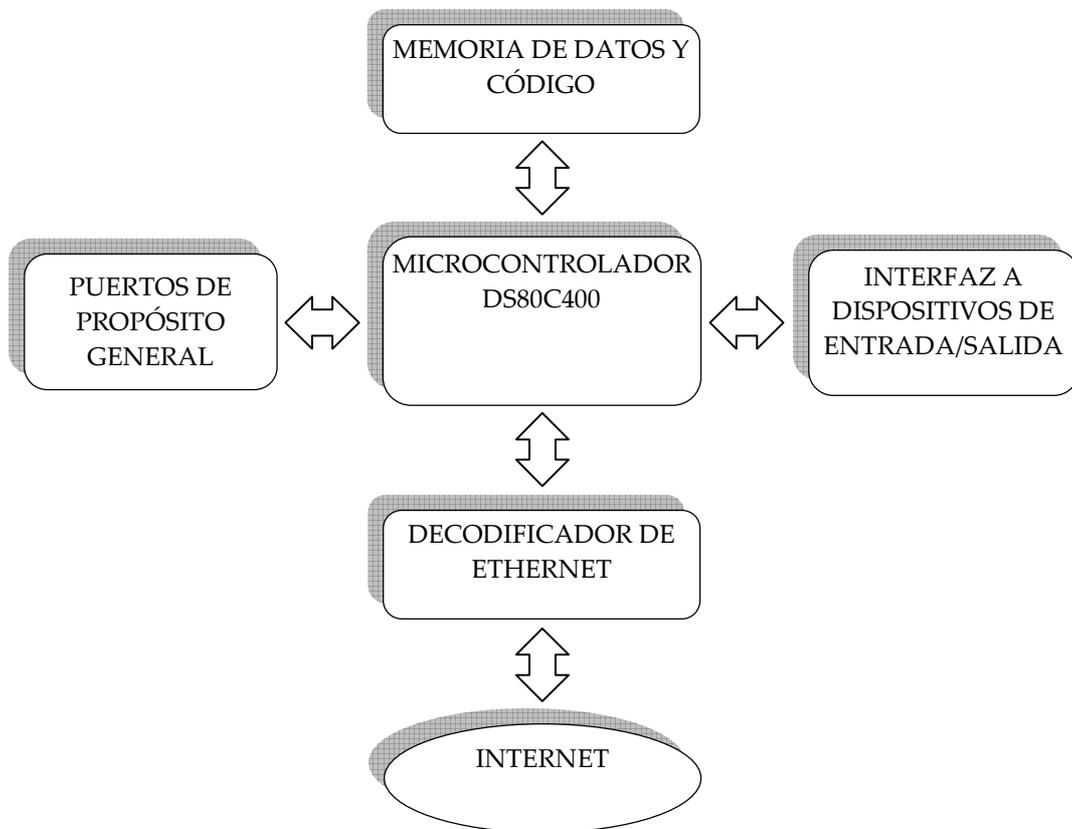


Figura 1.3. Diagrama a bloques de la tarjeta de evaluación del DS80C400.

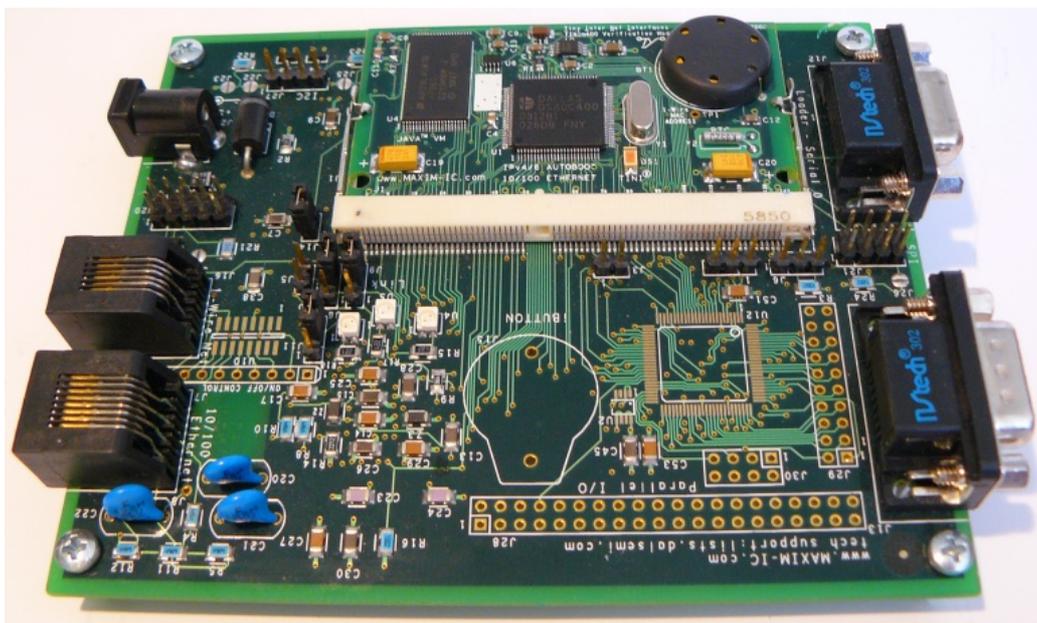


Figura 1.4. Tarjeta de evaluación del microcontrolador de red DS80C400.

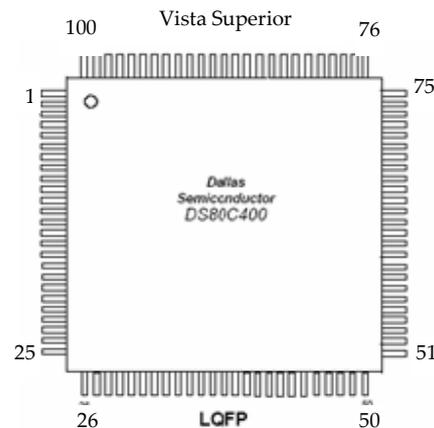


Figura 1.5. Microcontrolador DS80C400.

La conectividad inmediata y el soporte de red se tienen a través de una ROM empotrada de 64 KBytes, que contiene la asistencia para realizar un enlace en la red sobre una conexión de Ethernet usando el protocolo de configuración dinámica de host (DHCP, *Dynamic Host Configuration Protocol*) conjuntamente con el protocolo trivial de transferencia de archivos (TFTP, *Trivial File Transfer Protocol*). DHCP es un conjunto de reglas para administrar automáticamente algunas funciones gestionando redes TCP/IP, otorga direcciones IP y opciones de configuración a ordenadores y estaciones de trabajo en una red. TFTP es un protocolo de transferencia y se utiliza para transferir pequeños archivos entre ordenadores en una red.

1.4 PROTOCOLOS TCP/IP

Un protocolo es un conjunto de reglas que especifican el intercambio de datos u órdenes durante la comunicación entre sistemas; son reglas de comunicación que permiten el flujo de información entre computadoras distintas que manejan lenguajes distintos. Para comunicaciones en Internet, fue creado el protocolo TCP/IP.

El nombre TCP/IP se refiere a un conjunto de protocolos de transmisión de datos y proviene de los 2 principales protocolos que lo conforman [4]:

- Protocolo de Control de Transmisión (TCP, *Transmisión Control Protocol*)
- Protocolo de Internet (IP, *Internet Protocol*)

TCP/IP son los protocolos fundamentales de Internet, también usados en redes de área local. Los estándares de los protocolos son abiertos: interconectan equipos de diferentes fabricantes sin problema, independiente del medio de transmisión físico y tienen un

esquema de direccionamiento amplio y común. Son protocolos de alto nivel estandarizados [5].

Un protocolo establece las reglas formales de comportamiento para que las computadoras puedan comunicarse. Los protocolos TCP/IP no dependen del sistema operativo ni de la computadora (son "abiertos"); cualquiera puede desarrollar productos que se ajusten a las especificaciones de TCP/IP.

La figura 1.6 muestra una representación de la jerarquía de los protocolos TCP/IP a través de un modelo de capas.

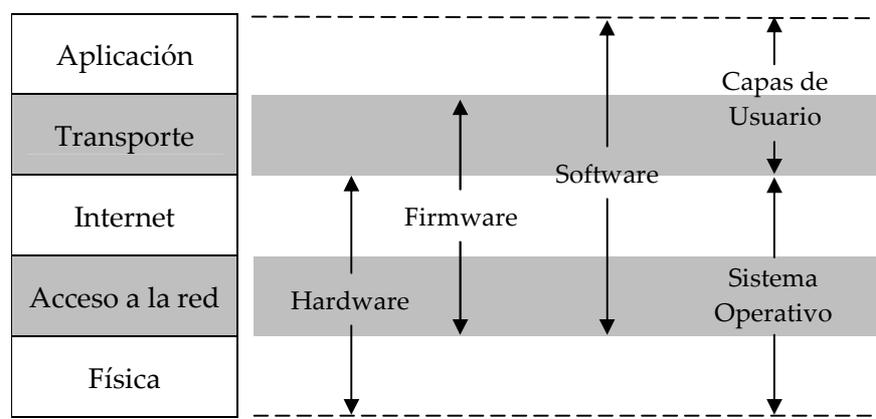


Figura 1.6. Arquitectura del protocolo TCP/IP.

- **Aplicación.-** Incluye protocolos destinados a proporcionar servicios como el del protocolo sencillo de transferencia de correo electrónico (SMTP, *Simple Mail Transfer Protocol*), del protocolo de transferencia de archivos (FTP, *File Transfer Protocol*), de conexión remota (TELNET, *TCP/IP Terminal Emulation Protocol*) y el protocolo de intercambio de hipertexto (HTTP, *Hypertext Transfer Protocol*).
- **Transporte.-** Los protocolos de éste nivel, TCP y el protocolo datagrama de usuario (UDP, *User Datagram Protocol*), se encargan de manejar los datos y proporcionar la fiabilidad necesaria en la transferencia de los mismos.
- **Internet.-** Incluye al protocolo IP que se encarga de enviar los paquetes de información a sus destinos correspondientes. Lo usan los protocolos del nivel de transporte.
- **Acceso a red.-** Es responsable del acceso y encaminamiento de los datos a través de la red.
- **Física.-** Define las características del medio de transmisión.

En éste proyecto de tesis se trabaja en la capa de aplicación; la tarjeta de evaluación del microcontrolador de red DS80C400 contiene la capa física en su hardware que lo conforma y en su software (plataforma TINI) se llevan a cabo las capas de acceso a red, Internet y transporte.

1.5 FAMILIA DE MICROCONTROLADORES AVR

Los microcontroladores de la familia AVR se caracterizan por tener un alto rendimiento, bajo consumo de energía, alta densidad de código, tecnología de memoria excepcional y alta integración. Esta familia es muy numerosa, incluye 71 dispositivos que comparten el núcleo, pero difieren en recursos [URL 12]. De toda esta familia el que se acopla mejor a las necesidades de este trabajo de tesis por su espacio en memoria, cantidad de puertos y precio es el ATMEGA16.

1.5.1 El microcontrolador ATMEGA16

EL ATMEGA16 es un microcontrolador CMOS de baja potencia basado en la arquitectura RISC. Ejecuta instrucciones en un solo ciclo de reloj. El ATMEGA16 alcanza un rendimiento de procesamiento cercano a 1 MIPS por MHz permitiendo que el diseñador del sistema optimice el consumo de energía contra velocidad de procesamiento.

A continuación se presentan las características más importantes del microcontrolador ATMEGA16 obtenidas de la hoja de datos proporcionada por el fabricante (ATMEL):

- Alto rendimiento, Microcontrolador de 8 bits con bajo consumo de energía.
- Arquitectura RISC.
 - 131 poderosas instrucciones – Ejecución en un solo ciclo de reloj.
 - 32 x 8 registros de propósito general.
 - Procesamiento arriba de 16 MIPS.
- Alta resistencia de segmentos de memoria no volátil.
 - 512 bytes en EEPROM.
 - 1 Kbyte en SRAM interna.
 - Ciclos de escritura y borrado: 10,000 en Flash y 100,000 en EEPROM
 - Retención de datos por 20 años a 85° C y de 100 años a 25° C.
 - Sección de código opcional para inicialización.
 - Programación de llaves para seguridad del software.
- Interfaz JTAG.
 - Programación de FLASH, EEPROM y bits de llave a través de la interfaz JTAG.
 - Soporta depuración extensa sobre chip.

- Características periféricas.
 - Dos temporizadores/contadores de 8 bits
 - Un temporizador/contador de 8 bits.
 - Contador en tiempo real con oscilador separado.
 - Cuatro canales PWM.
 - 8 canales ADC de 10 bits.
 - Interfaz serial.
 - USART programable.
 - Interfaz serial SPI, maestro/esclavo.
 - Watchdog programable.
 - Comparador análogo.
- Características especiales del Microcontrolador.
 - Oscilador RC para calibración interna.
 - Fuentes de interrupción externas e internas.
 - Seis modos de espera: *Idle*, reducción de ruido ADC, Power-save, Power-down, Standby y Standby extendido.
- Entrada/Salida.
 - 32 líneas programables de entrada/salida.
- Voltajes de Operación.
 - 2.7 – 5.5 Volts para el ATMEGA16L.
 - 4.5 – 5.5 Volts para el ATMEGA16.
- Grados de velocidad.
 - 0 – 8 MHz para el ATMEGA16L.
 - 0 – 16 MHz para el ATMEGA16.

Las interrupciones en el microcontrolador hacen notar a la CPU la aparición de alguna circunstancia que requiera su intervención. De éste modo, el microcontrolador deja por el momento la tarea que estaba realizando y atiende la interrupción, una vez atendida seguirá con su labor anterior. El ATMEGA16 cuenta con tres fuentes de interrupción externas: INT0 (PORTD2), INT1 (PORTD3) e INT2 (PORTB2).

Este microcontrolador incluye cuatro puertos, todas las terminales con una función alterna. En uno de los puertos se incluye el protocolo SPI Maestro/Esclavo que es empleado en este trabajo de tesis y que se describe en la sección 1.6. Las 40 terminales se enumeran en la figura 1.7.

1.6 PROTOCOLO SPI

El protocolo SPI (*Serial Peripheral Interface*) es un estándar de comunicaciones, usado principalmente para transferir paquetes de información de ocho bits entre circuitos integrados; controla casi cualquier sistema digital que acepte un flujo serial de bits regulado por un reloj [6].

(XCK/T0) PB0	1	40	PA0 (ADC0)
(T1) PB1	2	39	PA1 (ADC1)
(INT2/AIN0) PB2	3	38	PA2 (ADC2)
(OC0/AIN1) PB3	4	37	PA3 (ADC3)
(SS) PB4	5	36	PA4 (ADC4)
(MOSI) PB5	6	35	PA5 (ADC5)
(MISO) PB6	7	34	PA6 (ADC6)
(SCK) PB7	8	33	PA7 (ADC7)
RESET	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2)
XTAL1	13	28	PC6 (TOSC1)
(RXD) PD0	14	27	PC5 (TDI)
(TXD) PD1	15	26	PC4 (TDO)
(INT0) PD2	16	25	PC3 (TMS)
(INT1) PD3	17	24	PC2 (TCK)
(OC1B) PD4	18	23	PC1 (SDA)
(OC1A) PD5	19	22	PC0 (SCL)
(ICP1) PD6	20	21	PD7 (OC2)

Figura 1.7. Pines del ATMEGA16.

Los dispositivos conectados al bus son definidos como maestro y esclavos. Un maestro es aquel que inicia la transferencia de información sobre el bus y genera las señales de reloj y control. Un esclavo es un dispositivo controlado por el maestro. Cada esclavo es controlado sobre el bus a través de una línea selectora llamada "chip select" o "select slave". La comunicación maestro-esclavo se ilustra en la imagen 1.8.

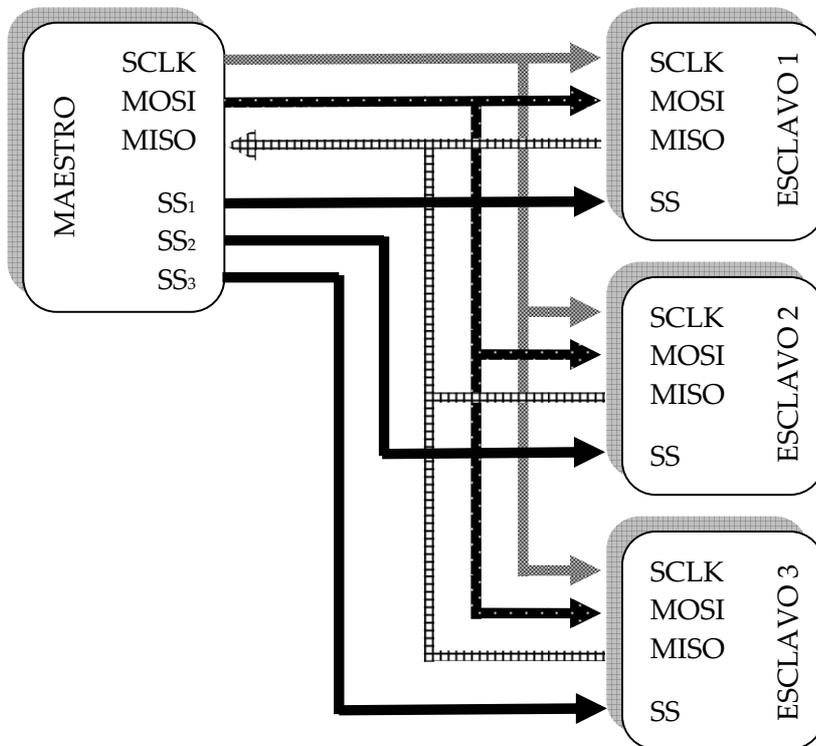


Figura 1.10. Registro para el control de la transferencia SPI.

donde:

SPIE.- Habilita la interrupción SPI (SPI Interrupt Enable).

Este bit provoca la ejecución de una interrupción si las condiciones del micro se lo permiten.

SPE.- Habilita la comunicación SPI.

Cuando el bit SPE es puesto en 1, se habilita la comunicación SPI.

DORD.- Orden de los datos.

Cuando el bit DORD es puesto en 1, el bit menos significativo del dato a transmitir es enviado primero.

Cuando el bit DORD es puesto en 0, el bit más significativo del dato a transmitir es enviado primero.

MSTR.- Selección Maestro/Esclavo.

Este bit selecciona el modo SPI Maestro cuando es puesto en 1, o el modo SPI Esclavo cuando es puesto en 0

CPOL.- Polaridad del Reloj.

Cuando este bit es puesto en 1, SCK es puesto en alto mientras está inactivo.

Cuando este bit es puesto en 0, SCK es puesto en bajo mientras está inactivo.

CPHA.- Fase del Reloj.

Los ajustes del bit CPHA determinan en que flanco del reloj los datos son desplazados hacia dentro o hacia fuera.

Si el bit CPHA es puesto en 0, los datos sobre la línea MOSI son detectados cada flanco de bajada y los datos sobre la línea MISO son detectados cada flanco de subida.

SPR1 y SPR2.- Selección del tipo de reloj SPI, 1 y 0.

Estos dos bits controlan la velocidad del SCK, se aplican al elemento configurado como Maestro. SPR1 y SPR2 no tienen efecto en el Esclavo.

Todas las transferencias de los datos, son sincronizadas por la línea de reloj del bus SPI. Un BIT es transferido por cada ciclo de reloj. Como se observa en el registro SPCR, las interfaces SPI tienen 2 bits de configuración, llamados CPOL y CPHA. Cada BIT tiene 2 estados, lo cual permite 4 diferentes combinaciones, las cuales son incompatibles una de la otra. Por lo que si dos dispositivos SPI desean comunicarse entre sí, estos deben tener la misma Polaridad de Reloj (CPOL) y la misma Fase de Reloj (CPHA).

Existen cuatro combinaciones de fase y polaridad de SCK definidos por el protocolo SPI, que son determinados por los bits de control CPOL y CPHA. Los diferentes modos de transferencia de datos en SPI se muestran en las figuras 1.11, 1.12, 1.13 y 1.14.

CPOL	CPHA	MODO SPI
1	1	A
0	1	B
1	0	C
0	0	D

Tabla 1.3. Modos de transferencia SPI.

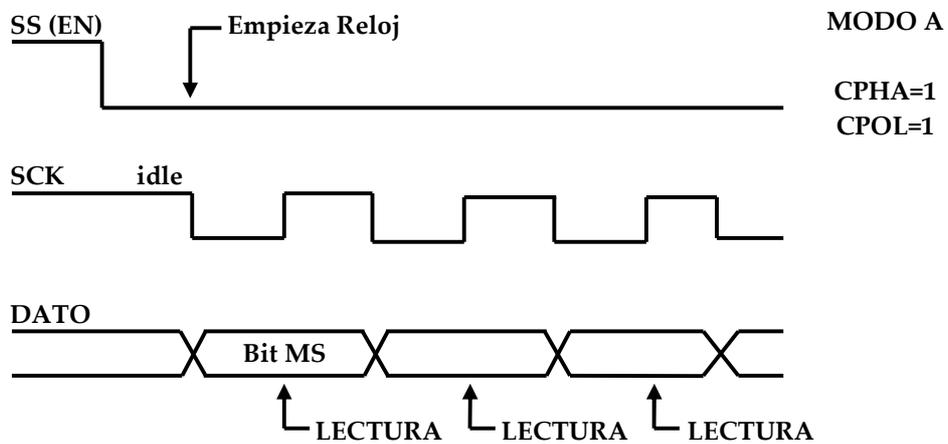


Figura 1.11. Modo de transferencia A para el protocolo SPI.

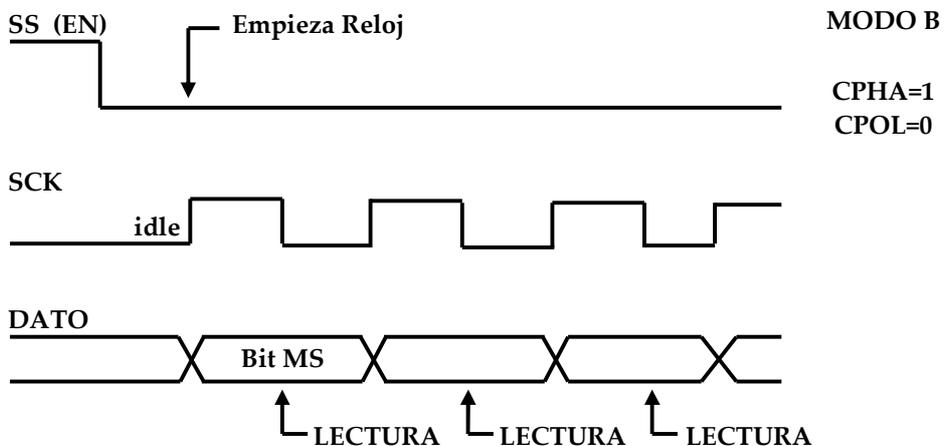


Figura 1.12. Modo de transferencia B para el protocolo SPI.

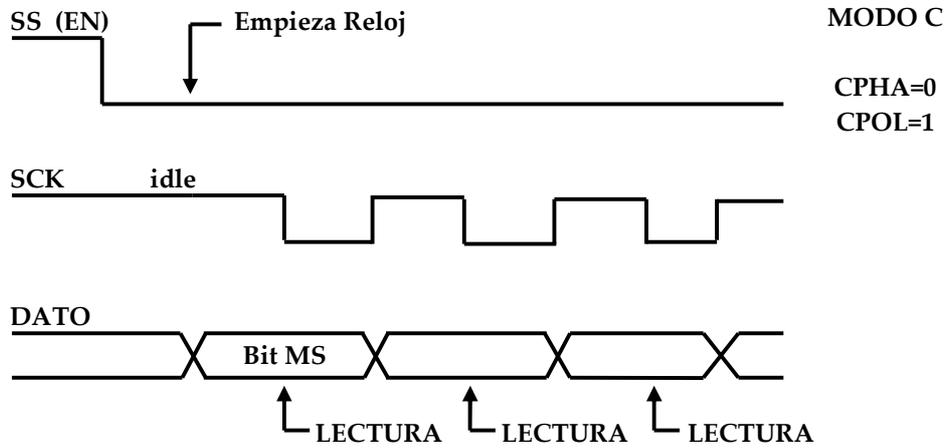


Figura 1.13. Modo de transferencia C para el protocolo SPI.

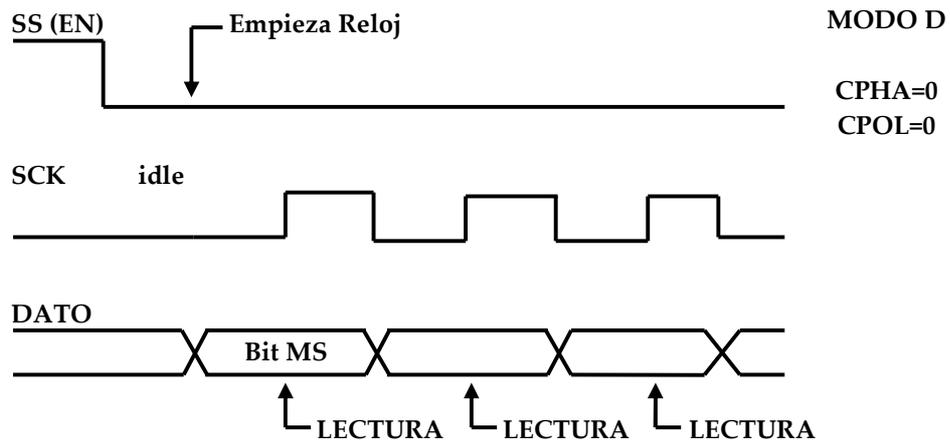


Figura 1.14. Modo de transferencia D para el protocolo SPI.

Se puede ver en las figuras 1.11, 1.12, 1.13 y 1.14 que en todos los casos, los datos se leen a la mitad del ciclo de reloj después de que se ponen en la línea de datos.

1.7 PROGRAMACIÓN EN JAVA Y LA PLATAFORMA TINI

Java es un lenguaje de programación orientado a objetos, independiente de la plataforma. Con la programación en Java se pueden realizar distintos tipos de aplicaciones, como los

applets (aplicaciones especiales), que se ejecutan dentro de un navegador al ser cargada una página HTML. Por lo general los *applets* son programas pequeños y de propósitos específicos [7].

Otra de las utilidades de la programación en Java es el desarrollo de aplicaciones del tipo *stand-alone*, que son programas autocontenidos ejecutados en forma independiente; es decir, que cualquier tipo de aplicación se puede realizar con ella. Java permite la modularidad por lo que se pueden hacer rutinas individuales que sean usadas por más de una aplicación [8].

Para el desarrollo de este trabajo de tesis fue necesaria la utilización de la plataforma TINI (*Tiny Internet Interface*) distribuida por Dallas Semiconductor. Este ambiente de desarrollo es un sistema operativo basado en el lenguaje de programación Java que incluye rutinas para conexión a Internet, contenido dentro del sistema de evaluación del microcontrolador de red DS80C400. A grandes rasgos se puede decir que es una plataforma integrada en una placa (*System on a Board*) y con un ambiente de operación Java [URL 13].

La programación en Java permite tomar ventaja de las extensas librerías disponibles para la plataforma TINI y que están integradas en el controlador principal a las que se puede acceder automáticamente cuando se programa vía TINI.

1.7.1 Requisitos de la plataforma de desarrollo

Se utiliza el término "plataforma de desarrollo" para referirse a la computadora usada para crear, construir, y cargar las aplicaciones. Ésta es la máquina en la que funciona el kit de desarrollo Java (JDK, *Java Developer's Kit*) y el ambiente de prueba que está conectado con el sistema usando Ethernet y/o un cable serial.

Ya que las herramientas empleadas se han escrito en Java, las aplicaciones de TINI® se pueden realizar en cualquiera de los sistemas operativos siguientes:

- MAC OSX.
- Solaris.
- Linux.
- OS Win32 (Windows 95, 98, NT, 2000, XP).

La instalación del entorno de ejecución de TINI® se puede hacer a través de un puerto USB, requisito que satisface cualquier computadora o estación de trabajo.

1.7.2 Componentes de Software

Se refiere a los programas necesarios para que la plataforma TINI® opere y ayude en la creación de aplicaciones. MAXIM/Dallas Semiconductor proporciona dos herramientas para cargar los programas. La primera es el kit de herramientas del microcontrolador (MTK, *Microcontroller Tools Kit*), recomendable para usuarios nuevos de TINI®, y por el momento solo disponible para la plataforma Windows. Básicamente es un archivo ejecutable que sirve para comunicarse con el cargador de inicialización/ROM de la mayoría de los microcontroladores de Dallas Semiconductor. Simplifica la configuración de funciones especiales y la carga y descarga de código, no es necesario software adicional para hacerlo funcionar.

La segunda herramienta es el JDK o JavaKit, un programa con una interfaz para el cargador. El JDK requiere el siguiente software para operar:

- Java Development Environment (JDE).
- Java Communications Application Programming Interface (API).
- TINI® Software Development Kit (TINI SDK).

JDE ofrece el ambiente de compilación, el segundo proporciona la infraestructura de comunicación serial RS-232 y colabora con el ambiente de operación, ambos se usan para el desarrollo de aplicaciones en Java. Entre el JDE y el TINI SDK ofrecen las herramientas de Java, el compilador y la máquina virtual de Java (JVM, *Java Virtual Machine*). Estos programas también se pueden obtener de manera libre en Internet [URL 14].

Una característica más del JDK es la inclusión de un convertidor TINI®, una herramienta con interfaz de línea de comandos para convertir archivos de clase Java a aplicaciones TINI®.

1.8 LENGUAJE HTML

HTML (*HyperText Markup Language*) es el lenguaje predominante para la construcción de páginas Web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes [9].

El lenguaje consta de etiquetas que tienen esta forma o <P> donde cada etiqueta significa una cosa y casi todas las etiquetas tienen su correspondiente etiqueta de cierre. Por ejemplo y . Así que el HTML no es más que una serie de etiquetas que se utilizan para definir la forma o estilo que se quiere aplicar a un documento.

1.9 PROGRAMACION EN C

El lenguaje C es un lenguaje de programación de propósito general que ofrece estructuras de control de flujo sencillas y un conjunto amplio de operadores. Se le considera un lenguaje de nivel medio ya que permite optimizar al máximo los recursos de la máquina, por ende, hace que la dificultad y que los errores que se puedan cometer programando aumenten. Esto lo convierte en un lenguaje potente y con un campo de aplicación ilimitado [10].

El lenguaje de programación C fue creado en 1972 por Ken Thompson y Dennis M. Ritchie en los Laboratorios Bell, orientado a la implementación de sistemas operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.

C es un tipo de programación en donde a la máquina se le dicta un conjunto de instrucciones, se le dice cómo hacer algo; éstas instrucciones se escriben en forma de lista y son ejecutadas en orden. A este tipo de programación se le llama “Programación Imperativa” [11].

1.9.1 Consideraciones para el microcontrolador

La familia de microcontroladores AVR fueron diseñados desde su comienzo para la ejecución eficiente de código C compilado. Las ventajas principales de programar en C se deben a que es un lenguaje muy eficiente puesto que es posible utilizar sus características de bajo nivel para realizar implementaciones óptimas, proporciona facilidades para realizar programas modulares y/o utilizar código o bibliotecas existentes y tiene riqueza de operadores y expresiones [12].

Los principales criterios a considerar para programar un microcontrolador en lenguaje C, son:

- Los tipos de variables: bit (1 bit), char (8 bits), int (16 bits), long int (32 bits), float (32 bits) y double (32 bits); y pueden ser declaradas en tres clases de almacenamiento.
- Las clases de almacenamiento: auto, static o register.
- Los operadores sobre bits: ~, <<, >>, &, ^, |; trabajan sobre operandos que no son punto flotante y afectan el resultado al nivel de bit.
- Los operadores lógicos; && y ||; tratan a los operandos como expresiones falsas o verdaderas.
- Los tipos de memoria en los que se puede almacenar información: SRAM, FLASH y EEPROM; cada una de ellas con características específicas. Por ejemplo los apuntadores son almacenados en la SRAM aunque pueden apuntar a FLASH o

EEPROM, las constantes pueden colocarse en la memoria del programa (FLASH) o para que los datos se conserven aún después de quitar el suministro de energía se usa la EEPROM.

2

DESARROLLO DEL HARDWARE

En éste capítulo se describen los componentes físicos necesarios para implementar al Sistema de Seguridad. Se detallan los elementos requeridos para manipular el sistema en forma local y remota.

2.1 REQUERIMIENTOS DEL HARDWARE

El Sistema de Seguridad cuenta con seis zonas y cada zona debe ser configurada con su respectivo modo y tipo. El modo puede ser: normalmente abierto (NA) o normalmente cerrado (NC); y el tipo: tipo 01 (entrada/salida), tipo 02 (perímetro) o tipo 03 (ruta de entrada interior). La configuración de cada zona puede ser de forma local o remota. Los tres diferentes tipos de configuración de las zonas se detallan en tabla 2.1.

CONFIGURACIÓN	DESCRIPCIÓN
Tipo 01 Entrada / Salida	Para áreas usadas como entrada y salida. Proporcionará un retardo de tiempo (configurado por el usuario) antes de accionar la alarma.
Tipo 02 Perímetro (24 horas)	Para todos los detectores o contactos en puertas y ventanas exteriores poco usadas. Proporcionará la activación instantánea de la alarma.
Tipo 03 Ruta de entrada interior	Para áreas como un vestíbulo o pasillo por el que se tiene que pasar al entrar (para llegar hasta la consola). Proporcionará una alarma retardada si primero se activa el área entrada/salida. Instantánea en caso contrario.

Tabla 2.1. Modos de configuración para cada zona.

Para tener un mejor control de acceso al sistema es necesaria la implantación de diferentes tipos de usuarios, en donde solo uno puede configurar todo el sistema y otro solo puede activar o desactivar la alarma. La tabla 2.2 muestra los diferentes tipos de usuarios y sus privilegios. La configuración de las zonas mostradas en la tabla 2.1 puede hacerla únicamente el usuario denominado “instalador” ya que tiene los permisos necesarios para llevar a cabo esa acción.

PERMISOS	USUARIO		
	INSTALADOR	PRIMARIO	SECUNDARIO
Definir modo y tipo de las zonas	X		
Configurar estado de la sirena	X		
Establecer estado del sistema		X	X
Modificar tiempo de retardo	X		
Cambiar código	X	X	
Añadir/Eliminar usuarios		X	

Tabla 2.2. Tipos de usuario y sus privilegios.

Al ser instalado por primera vez el sistema o reinicializado en su totalidad, las zonas toman su valor por defecto: normalmente abierto (NA) para los 6 zonas y de tipo 01 para las dos primeras, 02 para las dos siguientes y 03 para las dos últimas.

El Sistema de Seguridad cuenta con tres estados: ARMADO, REPOSO y ACTIVADO. Se encuentra ARMADO cuando esté perceptivo a los cambios en el estado de los sensores tipo 01, 02 y 03; en estado de REPOSO el sistema sólo puede ser alterado si algún sensor del tipo 02 ha sido estimulado y por último el sistema es ACTIVADO cuando la alarma ha detectado algún intruso dentro del área.

El estado del sistema puede ser cambiado de ARMADO a REPOSO, REPOSO a ARMADO, ACTIVADO a REPOSO, REPOSO a ACTIVADO y ARMADO a ACTIVADO. Los tres primeros cambios se hacen en forma local o remota y los dos últimos son automáticos al momento de detectar algún intruso dentro del área asegurada.

2.2 DISEÑO

El sistema de alarma requiere de dos subsistemas, uno para la manipulación de la alarma en forma local y el otro para su manipulación en forma remota; ambos subsistemas comunicados por el puerto SPI. Para el subsistema de interfaz remota se utiliza la tarjeta de evaluación del DS80C400 y para la interfaz local al ATMEGA16, que en tres de sus cuatro puertos se conecta la colección de sensores, la pantalla LCD, el teclado y la sirena.

El diagrama de la figura 2.1 muestra la organización del sistema, se observan los módulos de los que está compuesto para después explicar cada uno de ellos.

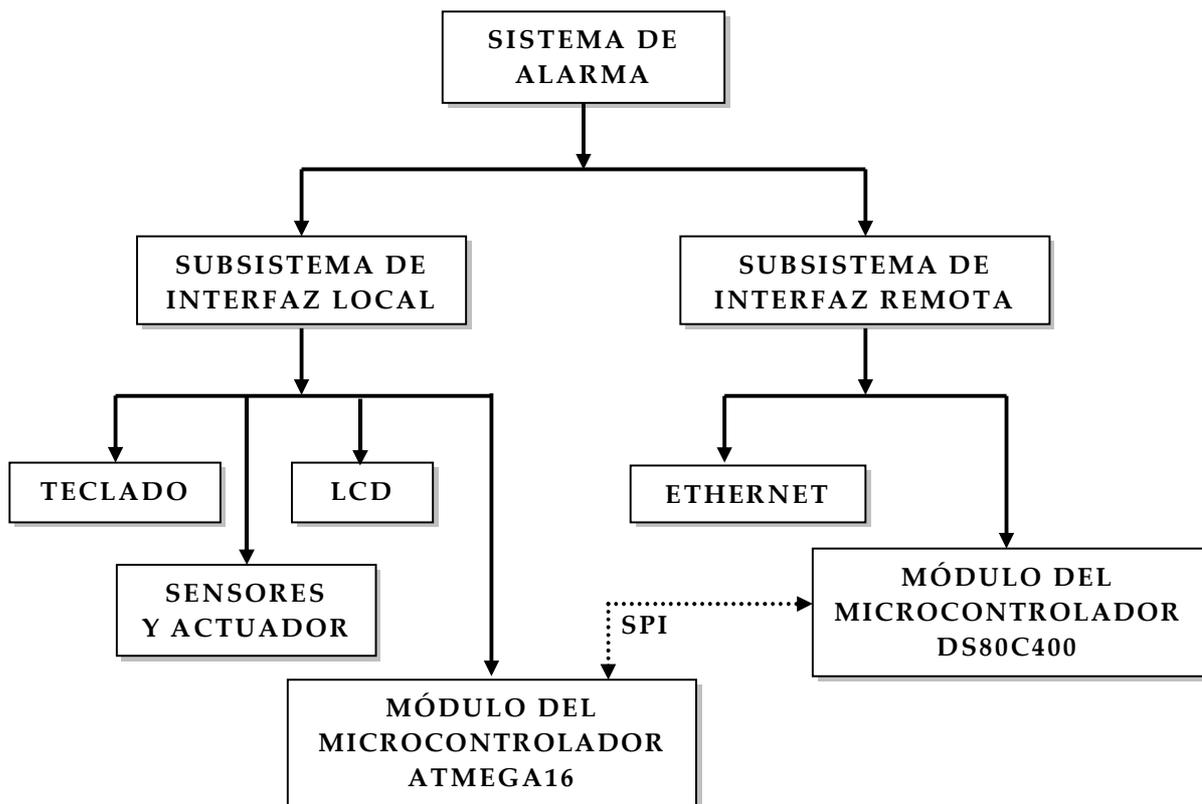


Figura 2.1. Diagrama a bloques del Sistema de Seguridad.

2.3 SUBSISTEMA DE INTERFAZ LOCAL

Con el subsistema de interfaz local se puede configurar al Sistema de Seguridad en forma local, está conformado de diversos dispositivos: el teclado para navegar entre las diferentes opciones y configurar, activar o desactivar el sistema, la pantalla LCD para visualizar la información a la que se está accediendo por medio del teclado, los sensores para monitorear el área protegida, el actuador (sirena) para avisar en caso de intrusión y el módulo del microcontrolador ATMEGA16 que se encarga de coordinar a todos los demás elementos para que el sistema funcione correctamente.

2.3.1 Teclado Matricial

El teclado matricial utilizado es de 4 x 4 teclas, para abarcar los 10 dígitos más teclas con funciones especiales que sirven para el control del Sistema de Seguridad. Es manipulado por el circuito integrado MM74C923N fabricado por Fairchild Semiconductor, el cual es un decodificador de teclado con el que se facilita su interacción con los demás módulos del sistema [URL 15].

Con el teclado matricial se cubren las necesidades básicas que cualquier usuario obtendría de algún Sistema de Seguridad comercial. Su diagrama se muestra en la figura 2.2 en donde se observan las conexiones entre los 16 botones (*push botton*) y el MM74C923N, el cual requiere de una compuerta NOT (DM74LS04N) para habilitar la salida de los datos. Las 8 terminales de los 16 botones, divididas en 4 filas: Y1, Y2, Y3 y Y4 y 4 columnas: X1, X2, X3 y X4, se conectan directamente al MM74C923N de donde se obtienen solo cinco pines a la salida, cuatro pines con el valor de la tecla presionada (A, B, C, D) y uno para avisar cuando un dato está disponible porque se acaba de presionar una tecla (DD).

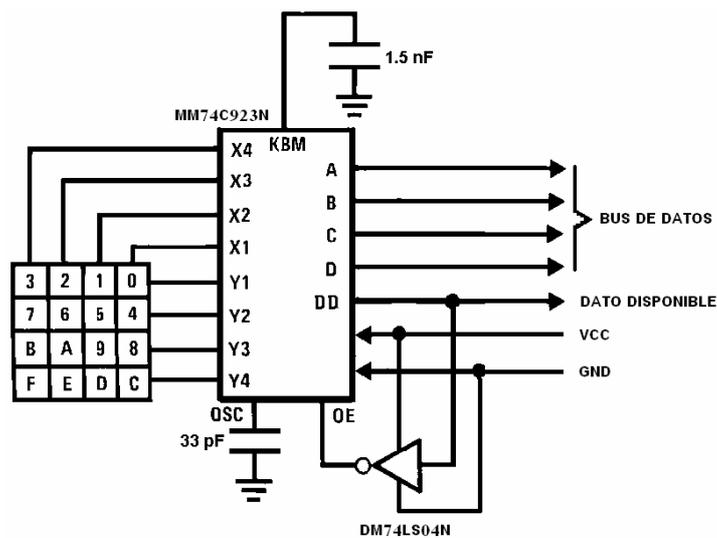


Figura 2.2. Diagrama del teclado matricial.

La placa del teclado en su vista superior e inferior se muestra en la figura 2.3 junto con la ubicación de cada uno de sus pines de entrada: VCC y GND y salida: A, B, C, D y DD.

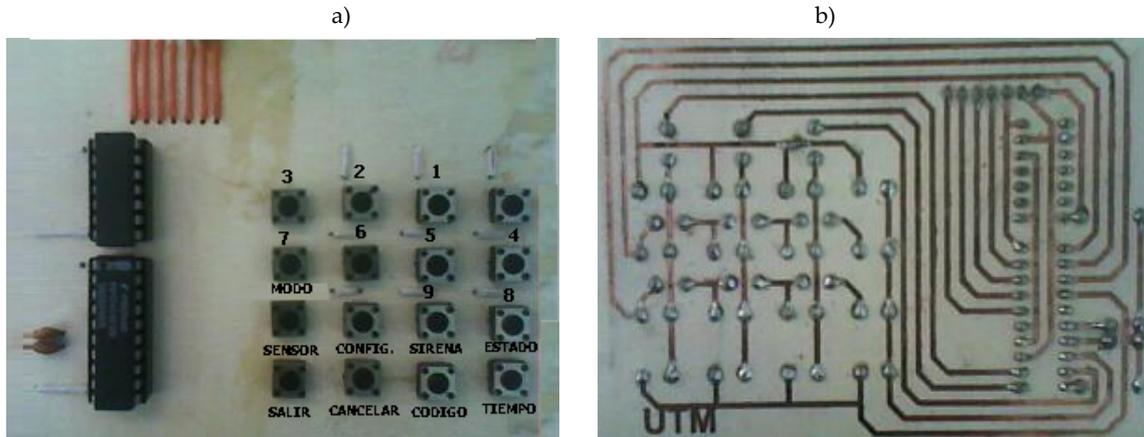


Figura 2.3. Teclado matricial a) vista superior y b) vista inferior.

El teclado matricial (como se muestra más adelante) se conecta directamente al módulo del microcontrolador ATMEGA16. En la tabla 2.3 se hace referencia a los pines de salida del teclado matricial y a qué pin del ATMEGA16 va conectado cada uno de ellos.

PIN	Microcontrolador ATMEGA16
A	PC0
B	PC1
C	PC2
D	PC3
DD	PC4
VCC	VCC
GND	GND

Tabla 2.3. Conexión de los pines del teclado matricial.

2.3.2 Colección de Sensores y Actuador

Existen cuatro tipos de sensores importantes para un Sistema de Seguridad comercial (sección 1.1.1), de los cuales se consideraron dos tipos en la implementación de éste trabajo: de presencia y de apertura de puertas o ventanas. Para detectar si hay actividad dentro de un área protegida se hace uso del sensor de movimiento modelo 998PI de la marca ADEMCO, se considera adecuado por su precio y cobertura, comparado con otras marcas existentes en el mercado, esto se verifica en la tabla 2.4. Para monitorear la

apertura de puertas o ventanas se utilizaron contactos magnéticos marca ALEPH modelo DC2541WH por su bajo costo y su funcionamiento simple. Las principales características se enumeran en la tabla 2.5.

	Honeywell	Rokonet	Ademco	Pyronix
Modelo	IS2560T	RK410PR	998PI	FP05782
Cobertura (m)	18 x 13	12 x 12	15.2 x 15.2	10 x 10
Consumo (mA)	17	12	16	15
Dimensiones (cm)	11.2 x 6 x 4	8.9 x 5.9 x 3.9	6.7 x 11.1 x 5.4	6 x 4.2 x 8.8
Voltaje (VDC)	12	12	12	12
Precio (DLS)	30	22	25.99	12

Tabla 2.4. Características principales del detector de movimiento.

	Honeywell	Tane Alarm Products	Aleph
Modelo	79392	TANE45WH	DC2541WH
Tipo de salida con/sin el imán puesto	NC o NA	NC / NA	NC / NA
Tipo de Montaje	Superficie	Madera, Aluminio	Superficie
Dimensiones (cm)	2.55 x 0.49	2.73 x .06 x 1.61	6 x 0.86 x 1.47
Precio (DLS)	10.00	2.30	2.03

Tabla 2.5. Características principales del contacto magnético.

El actuador es la sirena que sonará si al activarse la alarma no se encuentra configurada en modo silencioso; la sirena utilizada es de la marca SYSCOM modelo R58GS por las ventajas que tiene sobre las otras marcas y modelos de características similares (tabla 2.6).

	Syscom	System Sensor	Honeywell	Steren
Modelo	R58GS	HR	748LC	TRS-588PB
Db	119	110	112	92
Consumo (mA)	250	75	1200	250
Voltaje (V)	6 – 12	12 – 24	9.6	12
Precio (DLS)	17	38	42	24

Tabla 2.6. Características principales de la sirena.

Los seis posibles sensores a colocar en el Sistema de Seguridad se conectan al puerto A del ATMEGA16, esta conexión entre el microcontrolador y los sensores se muestra en la tabla 2.7. Los sensores colocados en estos pines pueden funcionar como circuitos normalmente abiertos (NA) o normalmente cerrados (NC).

Los seis sensores básicamente se comportan como interruptores normalmente cerrados; uno de sus extremos se conecta en las terminales del ATMEGA16 (tabla 2.7) y el otro a tierra, debe existir una resistencia de 10 MΩ conectada entre la terminal del microcontrolador y la fuente de voltaje. Si no se utilizan los seis sensores, debe colocarse la

resistencia del microcontrolador a tierra. Las conexiones se ejemplifican en la figura 2.4 a) y b).

SENSOR	Microcontrolador ATMEGA16
1	PA0
2	PA1
3	PA2
4	PA3
5	PA4
6	PA5

Tabla 2.7. Conexión de cada uno de los sensores.

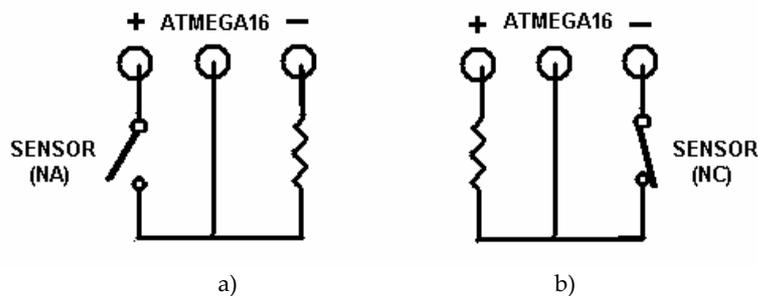


Figura 2.4. Conexión del sensor a) NA y b) NC.

2.3.3 Pantalla LCD

La pantalla LCD utilizada es manufacturada por TIANMA con número de serie JHD 162A, la cual muestra el contenido en 16 caracteres por 2 filas. Este dispositivo puede ser manejado con una interfaz de 4 u 8 bits de datos y para éste trabajo se utilizó la interfaz de 4 bits [URL 16]. Se necesitan señales de control para inicializar y desplegar los datos en la pantalla, estas señales son generadas y transferidas por el módulo del microcontrolador ATMEGA16 junto con la información que se necesita desplegar. En la figura 2.5 está la pantalla de LCD y la ubicación de cada uno de sus 16 pines.

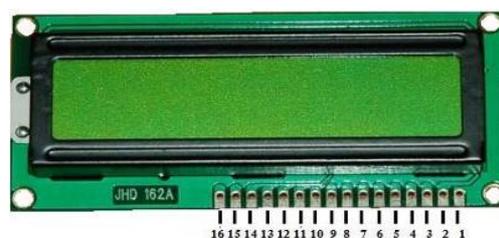


Figura 2.5. Pantalla LCD JHD 162A.

La tabla 2.8 muestra la configuración de los 16 pines del LCD y cuáles de ellos se conectan al microcontrolador ATMEGA16 (esto último se detalla en la sección 2.3.4). El diagrama de la conexión entre el ATMEGA16 y la pantalla LCD se muestra en la figura 2.6.

PIN	Pantalla LCD	Microcontrolador ATMEGA16
1	VSS	GND
2	VCC	VCC
3	VEE	GND
4	RS	PD4
5	R/W	GND
6	E	PD5
7	DB0	----
8	DB1	----
9	DB2	----
10	DB3	----
11	DB4	PD0
12	DB5	PD1
13	DB6	PD2
14	DB7	PD3
15	LED+	----
16	LED-	----

Tabla 2.8. Conexión de los pines de la pantalla LCD con el ATMEGA16.

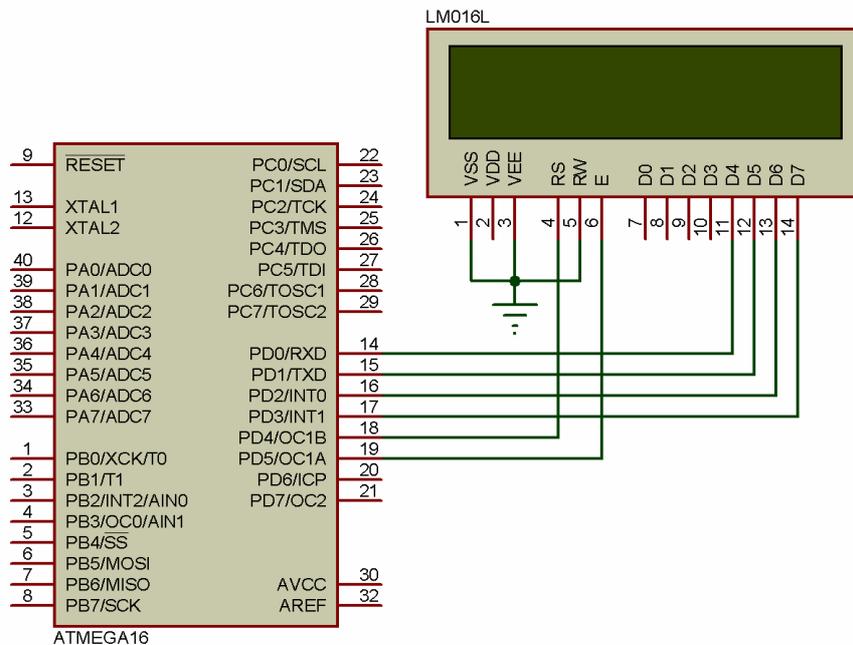


Figura 2.6. Diagrama de conexión entre el ATMEGA16 y la pantalla LCD.

2.3.4 Módulo del Microcontrolador ATMEGA16

Este módulo coordina y maneja a los módulos anteriores para que el subsistema de interfaz local funcione correctamente. En su memoria EEPROM almacena los datos necesarios para que el sistema mantenga su configuración (aun sin energía), se encarga de desplegar información en la pantalla de LCD, recibe datos desde el teclado matricial, por medio de led's muestra el estado de la alarma y a su activación hace sonar una sirena (si la configuración se lo permite). También cuenta con un botón de inicialización (reset) para regresar al microcontrolador a su estado inicial.

La parte principal de éste módulo es el microcontrolador ATMEGA16, sus características principales para el desarrollo de esta tesis son: 16 Kbytes de memoria Flash, 512 bytes de memoria EEPROM, 1 Kbyte de memoria SRAM, 32 pines de entrada/salida y un puerto SPI. En la tabla 2.9 se resumen las características más importantes de éste microcontrolador.

FLASH	16
EEPROM	512
SRAM	1024
Máximos pines de Entrada/Salida	32
Frecuencia Máxima (MHz)	16
Vcc (V)	2.7 - 5.5
Temporizadores de 12-bits	1
Temporizadores de 8-bits	2
PWM	4
RTC	Si
SPI Maestro/Esclavo	1
USART	1
TWI	Si
ISP	Si
Canales A/D de 10-bits	8
Comparador Analógico	Si
Watchdog	Si
Oscilador Interno	Si
Multiplicador por Hw	Si
Interrupciones	21
Interrupciones externas	3
Memoria de Auto Programación	Si
Encapsulados	PDIP 40 TQFP 44 MLF 44
Interfaz JTAG	Si

Tabla 2.9. Características del ATMEGA16.

Como se ha mencionado en secciones anteriores; el teclado matricial de 4x4, la pantalla LCD, los sensores y el actuador se conectan directamente a éste módulo. En la pantalla LCD se muestran las diferentes funciones del sistema y para seleccionar alguna se hace uso del teclado matricial de 4x4. La colección de sensores (de presencia y de apertura de puertas o ventanas) detectan los cambios que ocurren en el entorno y el actuador (sirena) proporciona un aviso auditivo en caso de activación de la alarma. El ATMEGA16 es el encargado de recopilar la información detectada por los sensores, interpreta la información proveniente desde el teclado para atender las peticiones del usuario y guarda el estado de la alarma para transferirle esa información al subsistema de interfaz remota.

Los 32 pines de entrada/salida del ATMEGA16 se dividen en cuatro puertos: A, B, C y D; cada uno de estos puertos se utiliza para conectar alguno de los dispositivos ya mencionados: pantalla LCD, teclado matricial de 4x4, sensores y actuador.

En el puerto A se conectan los sensores, el puerto B sirve para comunicarse con el subsistema de interfaz remota por medio del protocolo SPI, en el puerto C se conecta la sirena y el teclado matricial de 4x4 y por último, por medio del puerto D se envían datos y señales de control a la pantalla de LCD.

El diagrama con la asignación de terminales para la conexión de los diferentes dispositivos con el microcontrolador ATMEGA16 se muestra en la figura 2.7.

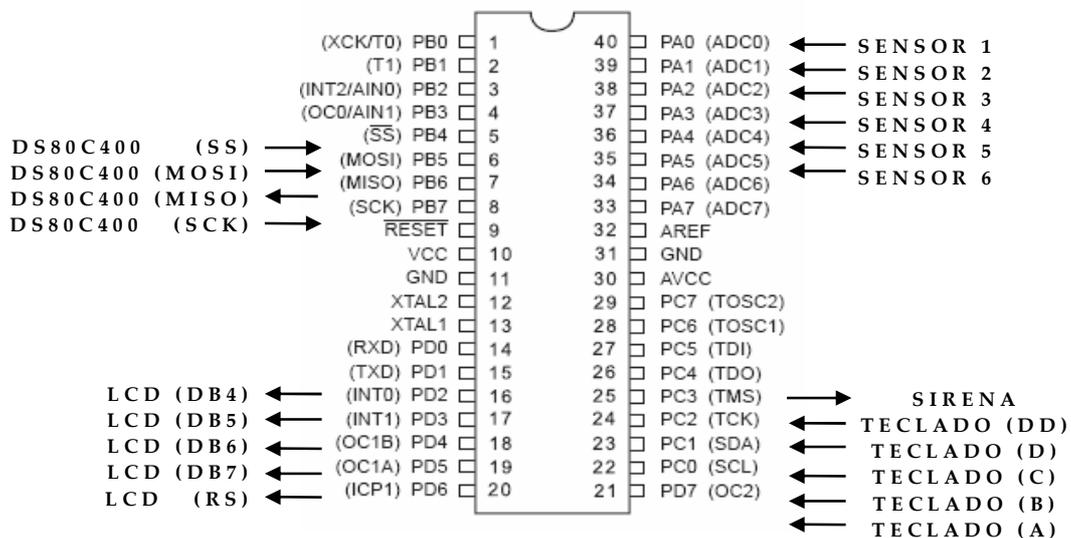


Figura 2.7. Diagrama de conexiones del ATMEGA16 con el teclado matricial de 4x4, la pantalla LCD, los sensores y el actuador.

El diagrama del subsistema de interfaz local con todos sus elementos se muestra en la figura 2.8 y en la figura 2.9 una imagen del mismo con todos sus componentes.

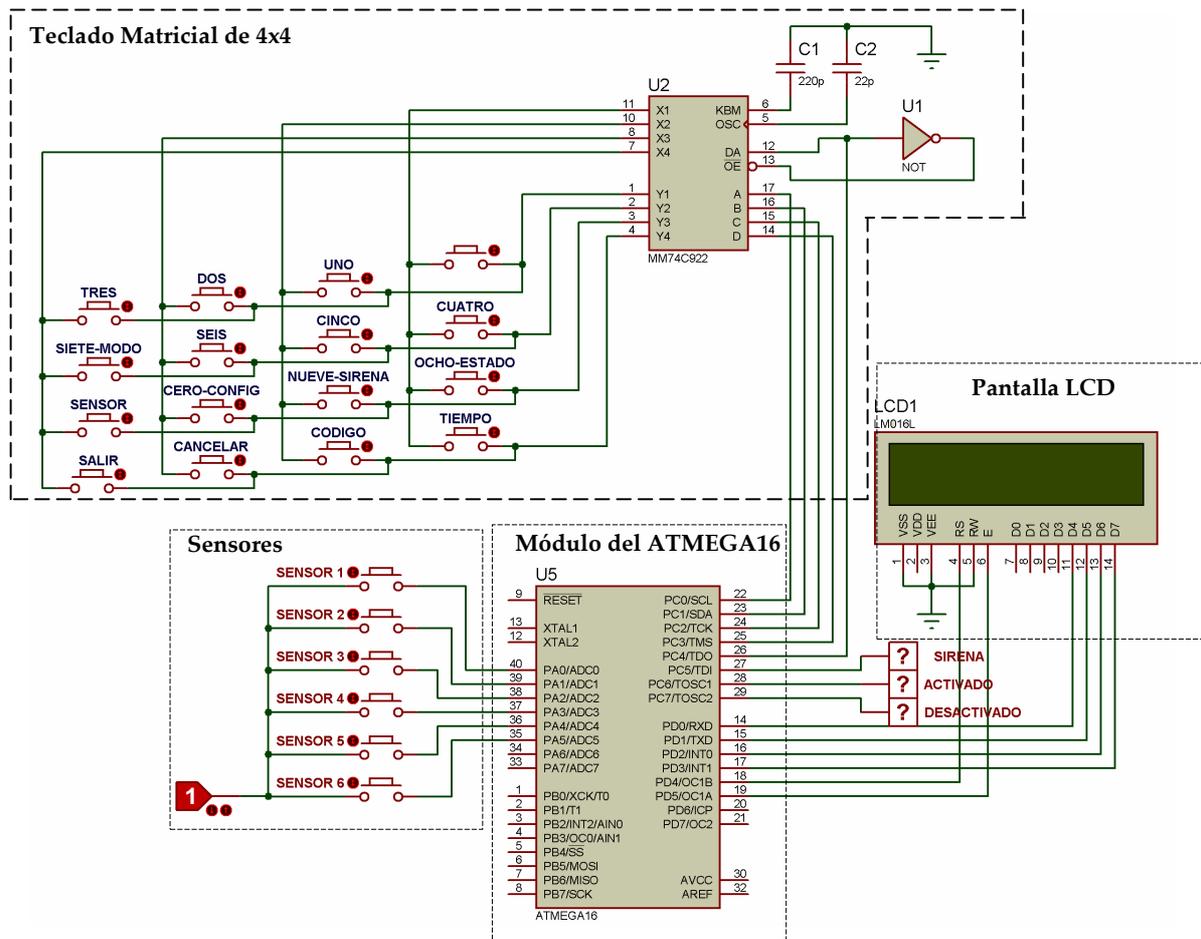


Figura 2.8. Diagrama del subsistema de interfaz local

2.4 SUBSISTEMA DE INTERFAZ REMOTA

Con el subsistema de interfaz remota se configura a distancia el Sistema de Seguridad, está conformado principalmente por dos dispositivos: el módulo del microcontrolador DS80C400 y el puerto Ethernet. Por medio del puerto Ethernet se tiene acceso al Sistema de Seguridad desde Internet y el módulo del microcontrolador DS80C400 se encarga de enviar la información obtenida desde Internet al subsistema de interfaz remota por medio del puerto SPI. Su diagrama a bloques se muestra en la figura 1.3.

2.4.1 Módulo del microcontrolador DS80C400

Este módulo es una plataforma de evaluación del microcontrolador de red DS80C400, para demostrar sus diversas aplicaciones y es creada por su mismo fabricante, Dallas Semiconductor en una versión única [URL 17].

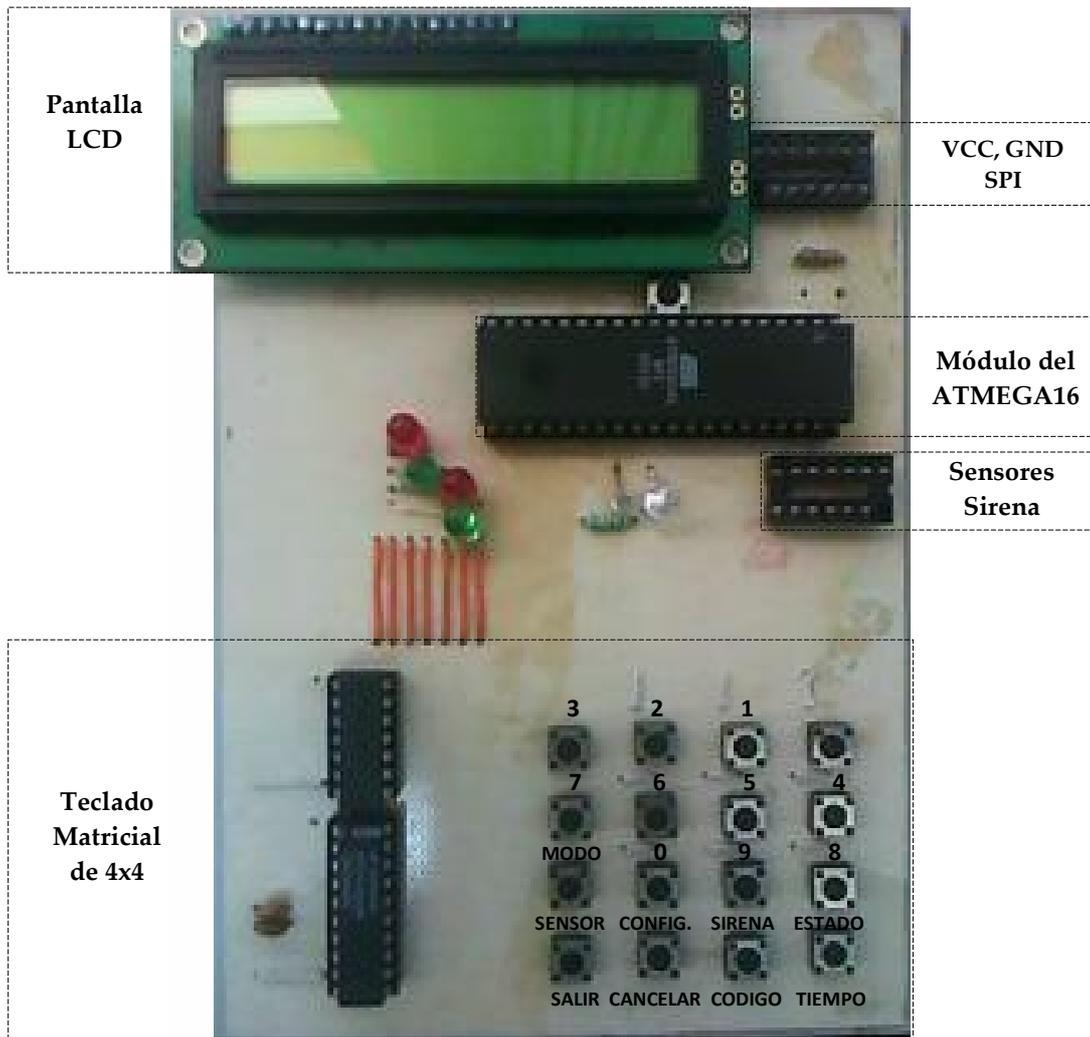


Figura 2.9. Foto del subsistema de interfaz local.

El elemento principal de éste módulo es el DS80C400 en el que se programa un entorno gráfico para tener acceso al Sistema de Seguridad desde Internet y con ello manipularlo a distancia. Este microcontrolador con ayuda incorporada para comunicación de entrada/salida, posee varios puertos con terminales de propósito general para realizar tareas simples de control, además de la posibilidad del uso de Ethernet. Sus principales características se enlistan a continuación y su aspecto físico se muestra en la figura 2.10.

- Arquitectura de alto rendimiento.
 - Ciclo de instrucción de 54 ns.
 - Frecuencia de operación de 75 Mhz.
 - Espacio de direccionamiento externo de 16 MBytes.
 - Cuatro apuntadores de datos con auto incremento/decremento.
 - Acelerador matemático de 16/32 Bits.

- Trabajo de red multitarea e I/O.
 - Controlador de acceso a medios 10/100 Mbps.
 - CAN 2.0 B.
 - Controlador de red 1-Wire.
 - Tres puertos seriales.
 - Hasta 8 puertos bidireccionales de 8 bits (64 terminales I/O).
- ROM Firmware.
 - Soporte de cargador de red sobre Ethernet usando DHCP y TFTP.
 - Pila de red TCP/IP accesible.
 - Soporta IPv4 e IPv6.
 - Implementa UDP, TCP, DHCP, ICMP e IGMP.
- 10/100 Ethernet MAC.
 - IEEE 802.3 flexible para las interfaces MII (10/100 Mbps) y ENDEC (Encode/Decode) (10Mbps) que permite selección de operación en baja potencia de la capa física (PHY).
 - 8 KBytes de memoria para Rx/Tx de paquete de datos con unidad de control de buffer.
 - Operación half o full-duplex con control de flujo.
- Sistema lógico primario integrado.
 - 16 fuentes de interrupción, con 6 externas.
 - Cuatro temporizadores/contadores de 16 bits.
 - *Watchdog Timer* programable.



Figura 2.10. Microcontrolador DS80C400 de MAXIM/Dallas Semiconductor.

El DS80C400 en conjunto con la memoria FLASH externa, su batería y otros dispositivos hacen que éste funcione correctamente, crean el módulo del microcontrolador DS80C400 mostrado en la figura 2.11 y cuyas características principales son:

- Servidor del entorno de desarrollo TINI® (en conjunto con el DSTINIs400).
- Interfaz MII para conexiones de red incluyendo 10/100 Base-T, Optical, y HomePNA.
- Tres puertos seriales.

- Controlador de red 1-Wire.
- Puerto CAN 2.0B.
- Reloj en tiempo real.
- Memoria Flash ROM de 1 MB para almacenar aplicaciones.
- Memoria SRAM de 1MB para almacenar datos.



Figura 2.11. Módulo del microcontrolador DS80C400.

2.4.2 Ethernet

El puerto Ethernet dentro del Sistema de Seguridad sirve para ampliar los límites de manipulación del mismo; por este medio se observa, manipula y controla a distancia el Sistema de Seguridad en su totalidad.

El módulo del DS80C400 se conecta a una tarjeta de evaluación denominada DSTINIs400 también fabricada por Dallas Semiconductor; ésta tarjeta contiene al puerto Ethernet que sirve para interconectar al módulo del microcontrolador DS80C400 con Internet y proporcionar al subsistema de interfaz remota su objetivo principal. El DSTINIs400 se muestra en la figura 2.12 con su respectivo puerto Ethernet y el puerto SPI, la utilización de éste último se verá en la sección 2.5.

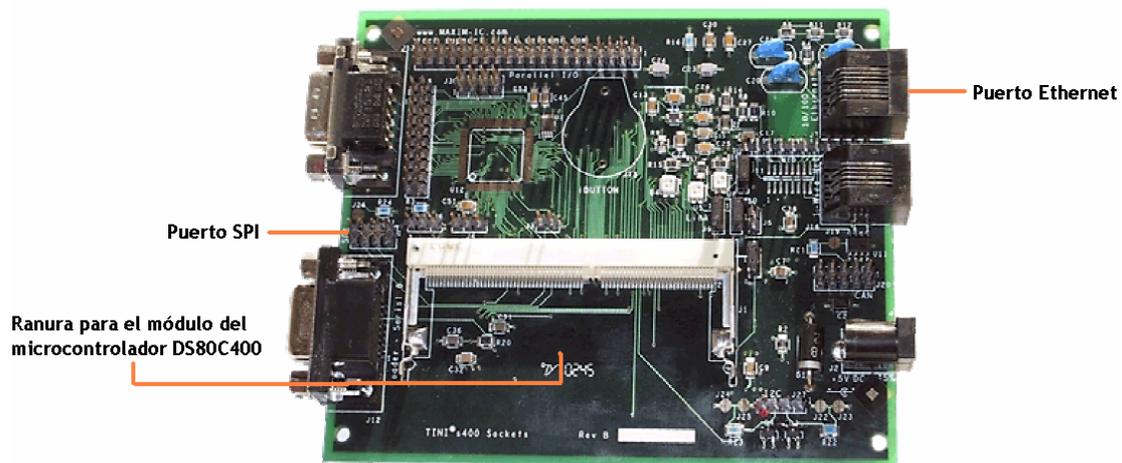


Figura 2.12. Módulo DSTINIs400: puerto Ethernet y SPI.

Los recursos empleados de la tarjeta de evaluación DSTINIs400 para el Sistema de Seguridad son el puerto Ethernet y el puerto SPI, aunque cuenta con más elementos importantes para otras posibles aplicaciones, entre los cuales tenemos:

- Dos puertos seriales.
- Controlador de red 1-Wire.
- Puerto CAN 2.0B.
- Puerto I2C.

2.5 INTEGRACIÓN DE LOS MÓDULOS DEL SISTEMA

En esta etapa se integran los dos subsistemas ya mencionados en este capítulo; los elementos contenidos en el subsistema de interfaz local se involucran con los elementos del subsistema de interfaz remota. La comunicación entre subsistemas se hace por medio del puerto SPI ya que ambos subsistemas contienen dicho puerto.

La comunicación por el puerto SPI es relativamente simple ya que en ambos subsistemas dentro de su estructura está contenido este tipo de protocolo. Para que exista comunicación por el puerto SPI entre dos elementos debe existir un elemento que sirva como maestro y otro que funcione como esclavo. Como se mencionó en la sección 1.6, este puerto maneja cuatro pines para la comunicación entre dispositivos y transferencia de información. Para éste Sistema de Seguridad el subsistema de interfaz remoto es el SPI maestro y el subsistema de interfaz local es el SPI esclavo, con esto los subsistemas de interfaz remota y local se mantienen en comunicación constante para dar aviso en caso de cambios en el estado de la alarma o de la configuración.

Ambos subsistemas, el de interfaz local y remota tienen su puerto SPI en el microcontrolador ATMEGA16 y en la tarjeta de evaluación DSTINIs400 respectivamente. La conexión maestro-esclavo entre los subsistemas se muestra en la tabla 2.10.

Módulo DSTINIs400	Microcontrolador ATMEGA16
SCK	PB7 (SCK)
MOSI	PB5 (MOSI)
MISO	PB6 (MISO)
SS	PB4 (SS)

Tabla 2.10. Conexión de los pines del puerto SPI entre el subsistema de interfaz local y el remoto.

La alimentación del circuito se obtiene de una fuente comercial, se puede ocupar cualquiera que a su salida tenga 6.5 V y potencia de 9 W; por precio y las características ya

mencionadas se utilizó la fuente DELUXE UNIVERSAL AC-DC ADAPTER, marca MITZU, modelo MCA-34 RD, con salida variable de 1.5 a 12 V. y potencia de 9 W.

Se necesita la fuente de alimentación de 6.5 Volts para mantener energizado al subsistema de interfaz remota y una de 5 Volts para el de interfaz local. La fuente de alimentación de 6.5 Volts se conecta directamente a su subsistema, el subsistema de interfaz local obtienen su alimentación de los pines 2 y 4 del conector j21 (puerto SPI) de la tarjeta de evaluación DSTINIs400 ubicada en el subsistema de interfaz remoto y con ello alimenta el teclado, la pantalla LCD y el microcontrolador ATMEGA16.

El diagrama completo del Sistema de Seguridad se muestra en la figura 2.13 en donde se observa que por medio del puerto SPI interactúan el subsistema de interfaz local y el subsistema de interfaz remota. La figura 2.14 es una imagen del sistema real de seguridad funcionando.

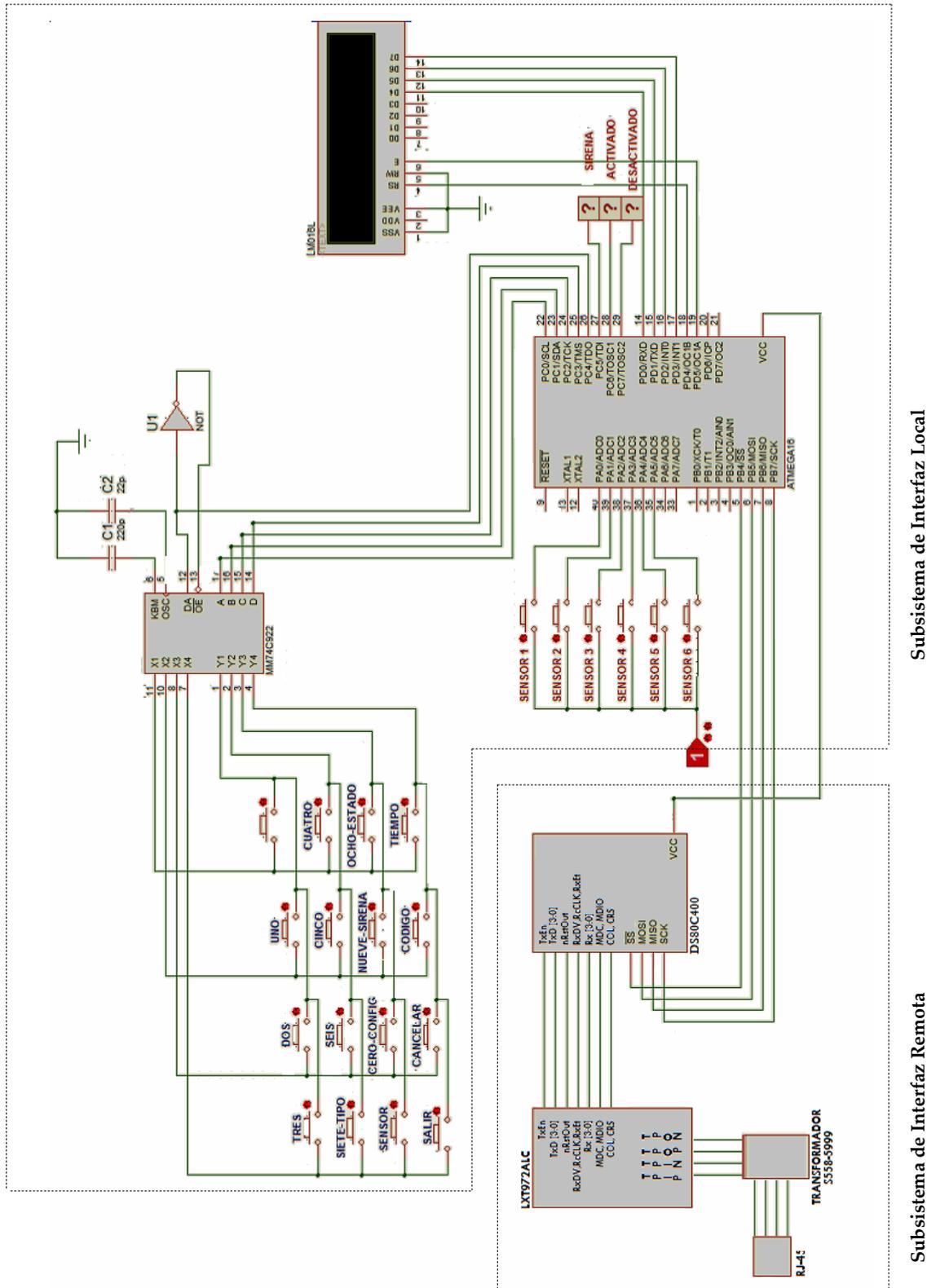


Figura 2.13. Diagrama del Sistema de Seguridad.

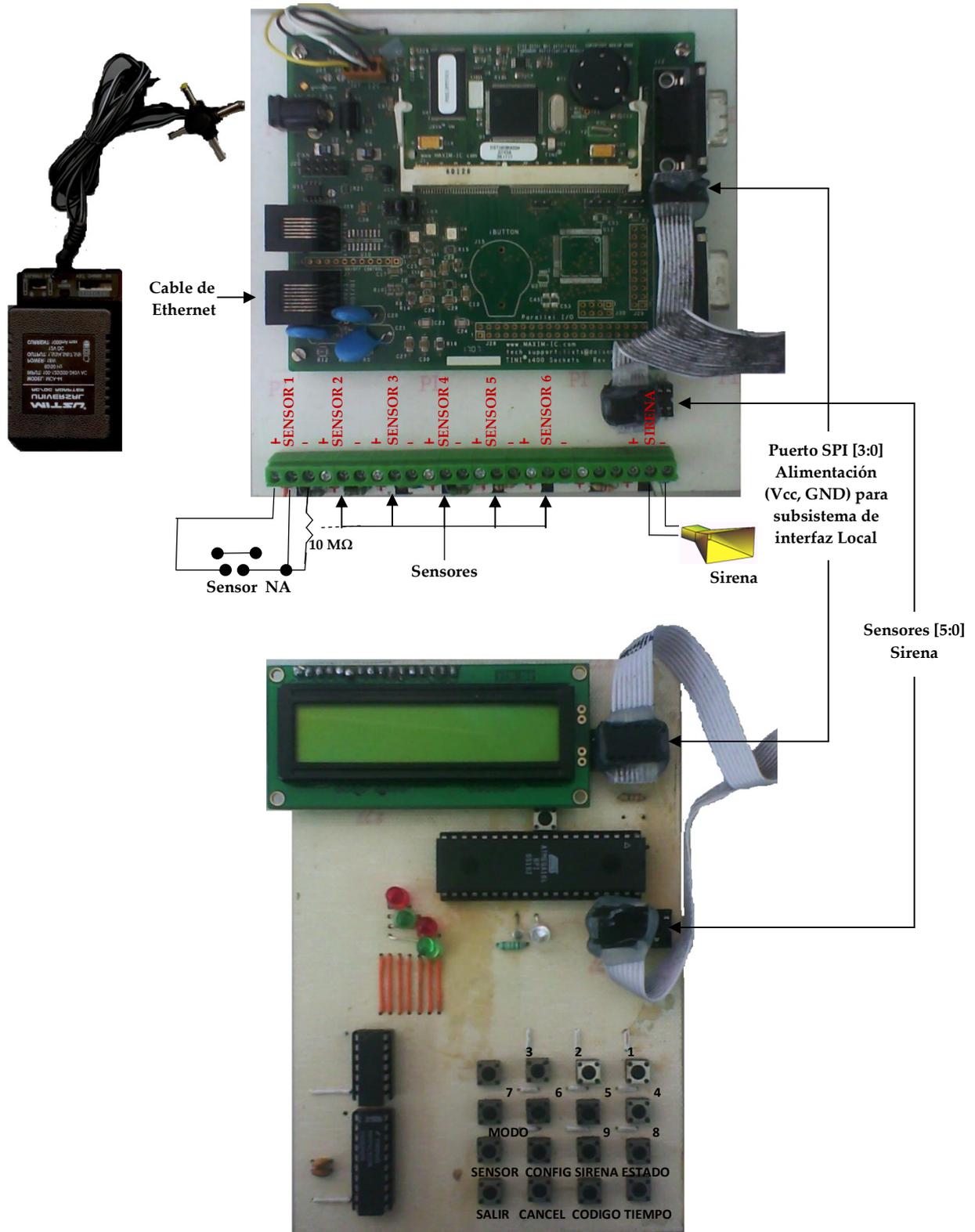


Figura 2.14. Hardware del Sistema de Seguridad.

3

DESARROLLO DEL SOFTWARE

En este capítulo se explican los programas que se realizaron para desarrollar el presente proyecto de tesis, algunos programas son extensos por lo que solo se describen las partes esenciales y el código completo se encuentra en un CD adjunto a este documento.

3.1 REQUERIMIENTOS DEL SOFTWARE

El Sistema de Seguridad para mayor control y facilidad en su manipulación hace uso de menús ya sea en forma local o remota. El acceso en forma local a los diferentes menús se hace por medio del teclado matricial y el acceso remoto es por medio de vínculos ubicados en la página Web principal del sistema.

En forma local, la función de cada uno de los botones del teclado matricial se especifica en la tabla 3.1.

BOTÓN	FUNCIÓN ESPECIAL	DESCRIPCIÓN
1		Teclas numéricas.
2		
3		
4		
5		
6		
7	MODO	Además del valor numérico, permite configurar cada uno de los seis sensores en los dos diferentes modos (NA o NC).
8	ESTADO	Además del valor numérico, activa o desactiva al sistema de alarma.
9	SIRENA	Además del valor numérico, activa o desactiva el modo silencioso, es decir, sin sirena.
0	CONFIGURACIÓN	Además del valor numérico, muestra la configuración de los sensores, de la sirena y el tiempo de retardo.
	SENSOR	Muestra la opción para configurar cada uno de los seis sensores en las tres diferentes zonas (01, 02 o 03).
	TIEMPO	Cambia el tiempo de retardo del sistema de alarma.
	CÓDIGO	Cambia el código de acceso de cuatro dígitos del sistema de alarma.
	CANCELAR	Cancela la operación en curso.
	SALIR	Salida rápida del sistema.

Tabla 3.1. Descripción de las funciones del teclado matricial.

3.2 DISEÑO

El software desarrollado se encuentra dividido en dos partes: el subsistema de interfaz local y el de interfaz remota. Ambos manipulan al Sistema de Seguridad; el primero de forma local por medio de un programa en lenguaje C, el cual una vez compilado es guardado y ejecutado dentro del módulo del microcontrolador ATMEGA16. El segundo de forma remota a través de archivos en código html, clases de java y archivos de procesamiento por lotes, almacenados dentro del módulo del microcontrolador DS80C400.

De esta forma el módulo del ATMEGA16 manipula la información recibida desde el teclado para desplegarla en la pantalla LCD, monitorea el estado de los sensores para la posible activación de la sirena e informa al subsistema de interfaz remota acerca del estado del sistema. El módulo del DS80C400 contiene la página Web visualizada a través de

Internet con la que se pueden configurar todas las opciones contenidas en el sistema y monitorear el estado de la alarma.

La estructura y dependencia de los diferentes programas realizados para cumplir con los requerimientos del sistema se visualizan en la figura 3.1, posteriormente se detalla cada uno de estos archivos.

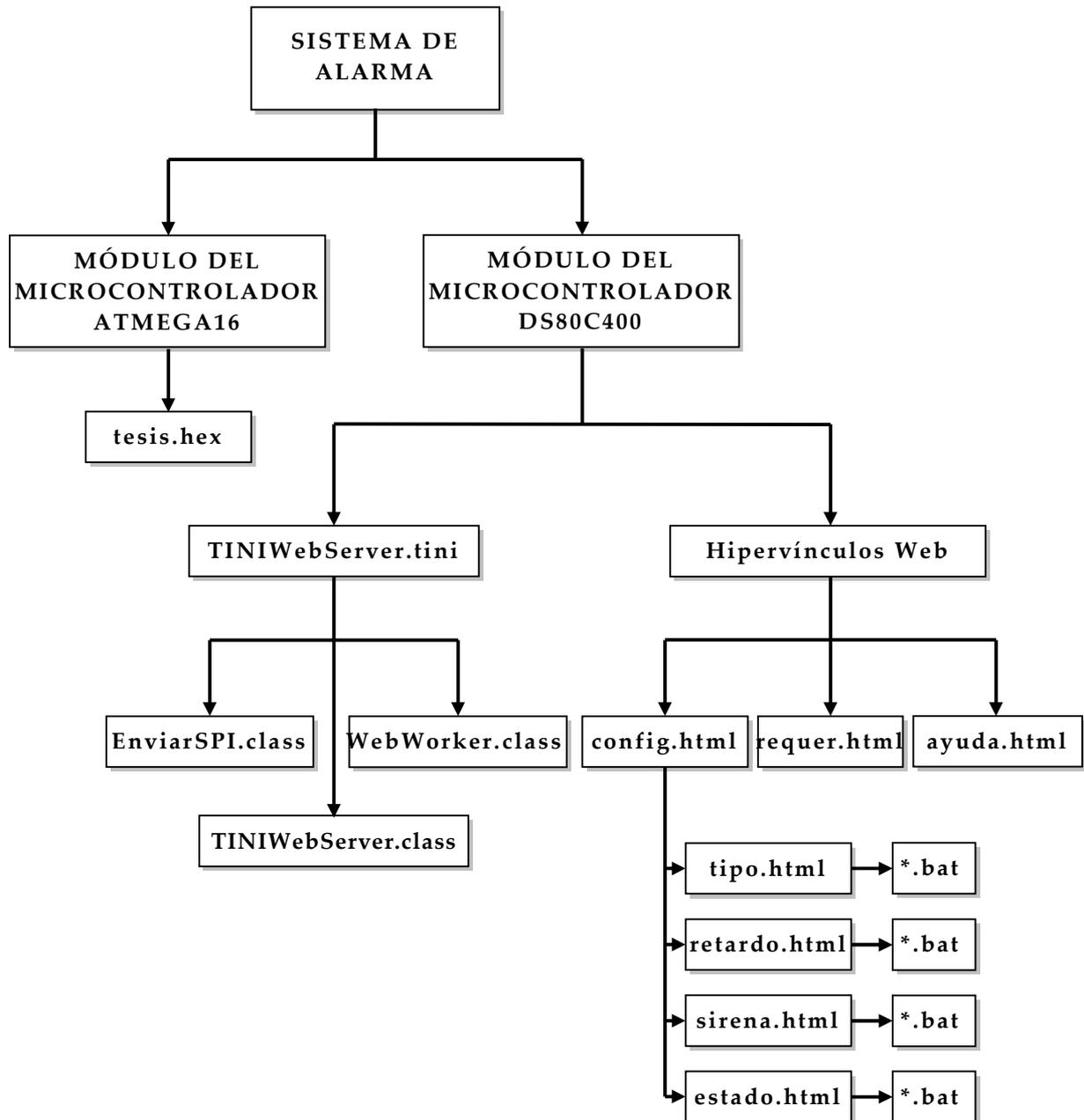


Figura 3.1. Diagrama a bloques del software del Sistema de Seguridad.

3.3 MÓDULO DEL MICROCONTROLADOR ATMEGA16

El módulo del microcontrolador ATMEGA16 es parte del subsistema de interfaz local y su funcionamiento está determinado por un programa llamado *tesis.hex* (lenguaje C) y sirve para la manipulación del Sistema de Seguridad en forma local. Este programa no se enfoca solo a los datos obtenidos en forma local, también recibe información del subsistema de interfaz remota para su manipulación. A continuación se detalla la función del archivo *tesis.hex*.

3.3.1 Archivo: tesis.hex

Este programa guarda todas las configuraciones, monitorea los sensores, controla la sirena y establece la comunicación con el subsistema de interfaz remota (específicamente con el módulo del microcontrolador DS80C400). Su diagrama de flujo es el mostrado en la figura 3.2.

El programa representado en el diagrama de flujo de la figura 3.2 configura las entradas y salidas, luego evalúa si existe alguna entrada del teclado y si es así, inicializa las configuraciones del sistema; de lo contrario habilita al microcontrolador como esclavo y espera la contraseña de acceso al mismo. Después de haber tecleado la contraseña correcta se puede elegir alguna de las opciones disponibles en el teclado (si tiene los permisos necesarios): MODO (0x07), ESTADO (0x08), SIRENA (0x09), CONFIGURACION (0x0A), SENSOR (0x0B), TIEMPO (0x0C) y CODIGO (0x0D); dependiendo de la opción seleccionada el flujo del programa tomara el camino correspondiente a dicha opción.

Tesis.hex maneja diferentes sub-funciones () para la manipulación del Sistema de Seguridad en forma local, por lo que en su diagrama de flujo (figura 3.2) existen 16 llamadas a éstas, las que se explican a continuación.

3.3.1.1 Se declara ATMEGA como esclavo.

Función: void slave_init ()

Esta función inicializa al ATMEGA16 como un esclavo para la comunicación SPI con el DS80C400, se configura al puerto B como una interfaz de comunicación con otro dispositivo habilitando las banderas del protocolo SPI, de lo contrario los pines del puerto funcionarían solo como entradas o salidas.

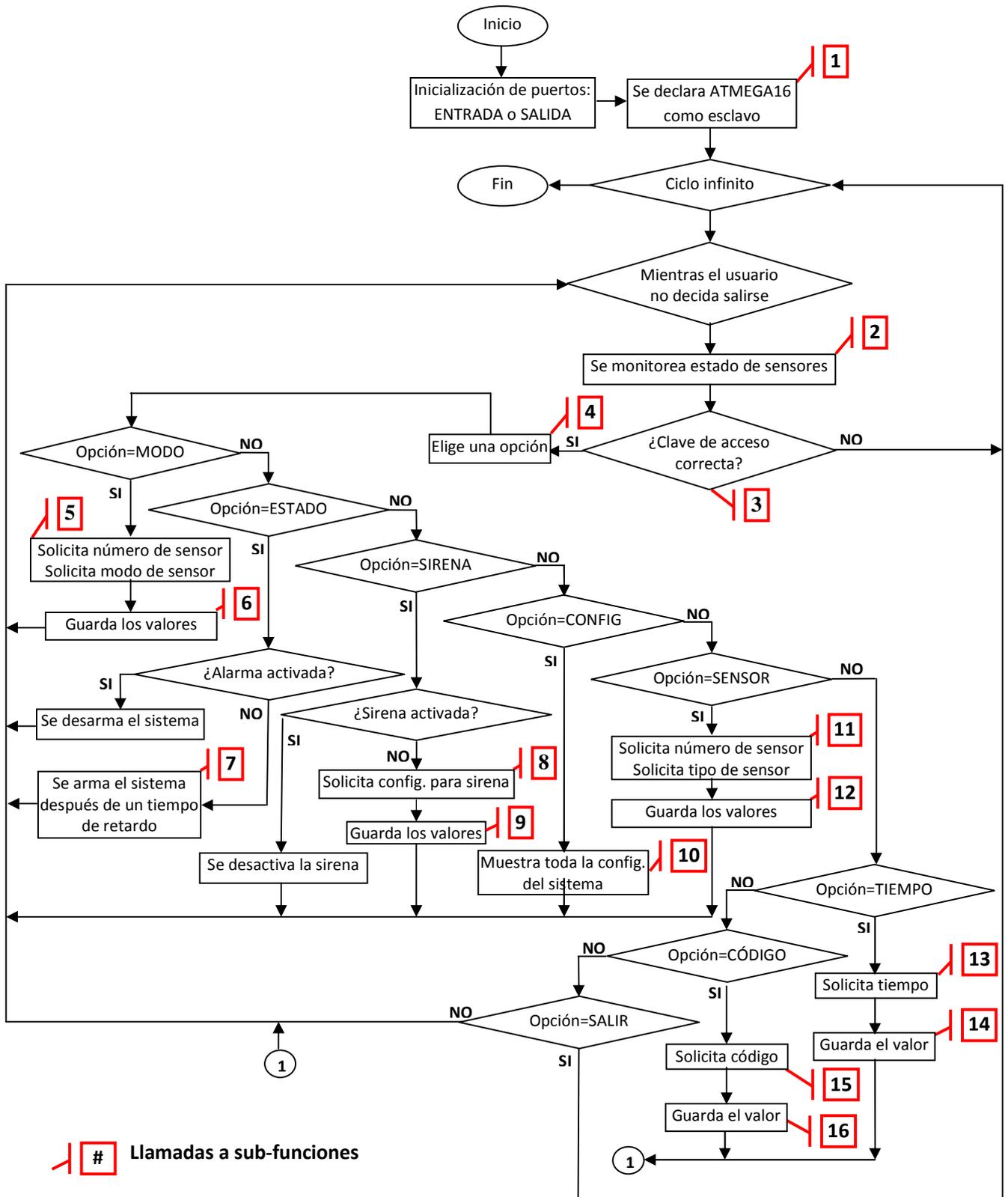


Figura 3.2. Diagrama de flujo del Sistema de Seguridad.

3.3.1.2 2 Se monitorea estado de sensores.

Función: void monitorear ()

Se encarga de monitorear el estado de los sensores, enciende la sirena si se activa la alarma y si su configuración se lo permite, y guarda en la EEPROM del ATMEGA16 los cambios en el sistema para el aviso a través de Internet. La figura 3.3 muestra el diagrama de flujo de ésta función.

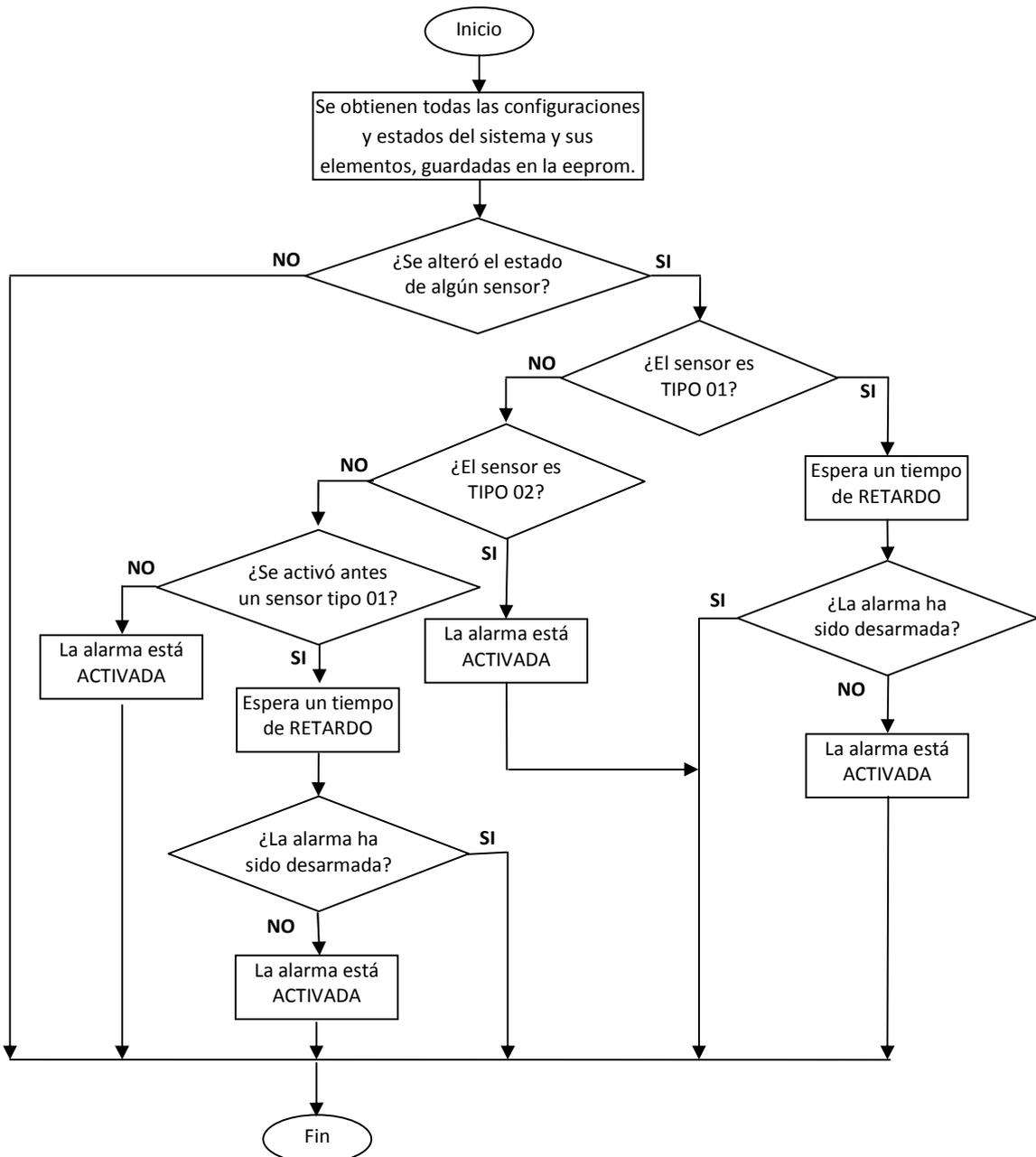


Figura 3.3. Función: void monitorear().

Primero se obtiene la configuración actual del estado de la alarma, la sirena, modo y tipo de los sensores y el retardo del sistema, toda esta información respaldada en la EEPROM, luego se revisa cada uno de los sensores y si alguno está activo se verifica su tipo. Si el sensor activado está configurado como tipo 1 entonces espera un tiempo de retardo y si después de ese tiempo sigue activo el sistema se activará, si el sensor activado es de tipo 2 la activación del sistema es inmediato y si es de tipo 3 se espera un tiempo de retardo para que después de haber transcurrido ese tiempo el sistema sea activado.

3.3.1.3 3 ¿Clave de acceso correcta?

Función: int clave ()

Cuando el sistema está en reposo, lo primero a introducir es la clave de acceso y para esto se tiene la función *clave()*. Acepta cuatro dígitos tecleados por el usuario, los guarda en una variable entera y verifica si es correcta. El diagrama de flujo de esta función se muestra en la figura 3.4.

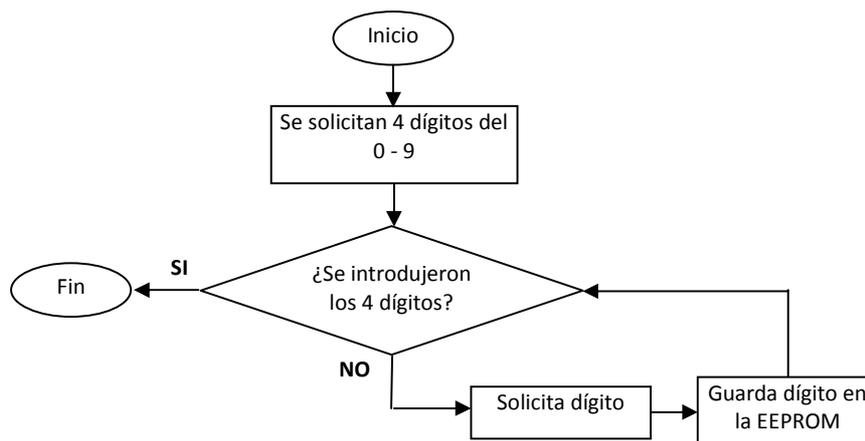


Figura 3.4. Función: int clave().

Si una clave de acceso es válida se tiene acceso al sistema, pero dependiendo del tipo de clave es la cantidad de permisos que se tienen para la manipulación del mismo.

3.3.1.4 4 Elige una opción

Función: int opción ()

Después de teclear una clave de acceso correcta y detectar el tipo de usuario, se puede elegir entre varias opciones presentes en el teclado. Reconoce los números del CERO al NUEVE y las opciones especiales de: MODO, ESTADO, SIRENA, CONFIGURACIÓN, SENSOR, TIEMPO, CÓDIGO, CANCELAR y SALIR. La función retorna el valor de la opción seleccionada por el usuario.

3.3.1.5 **5** **11** Solicita número de sensor, modo y tipoFunción: `int num_sensor ()`Función: `int tipo_sensor ()`Función: `int obtener_NANC ()`

Si se accede a las opciones MODO o SENSOR del teclado matricial, se utilizan estas funciones. La función `num_sensor()` retorna el número de sensor (1-6) elegido por el usuario para configurarlo. El `tipo_sensor()` proporciona los tres posibles tipos de configuración (tabla 2.1) para los sensores con los que se pueden mantener activos: 01, 02 ó 03. El modo de los sensores lo retorna la función `obtener_NANC()` en donde se puede elegir entre normalmente abierto (NA) o normalmente cerrado (NC).

3.3.1.6 **6** **12** Guarda el tipo y modo de un sensorFunción: `void config_sensor (int , int)`Función: `void config_NANC (int , int)`

Ambas funciones respaldan en la EEPROM del ATMEGA16 lo retornado por las funciones `num_sensor()`, `tipo_sensor()` y `obtener_NANC()`. Para `config_sensor(int ,int)` los parámetros recibidos son el número de sensor y el tipo de configuración y para `config_NANC(int ,int)` son también el número de sensor más el modo de configuración.

3.3.1.7 **7** Tiempo de retardoFunción: `void delay (int)`

Proporciona un retardo de un *número* de segundos en el transcurso del programa. Esta función es utilizada para determinar los tiempos de entrada y respetar el tiempo de espera establecido por el usuario antes de activar el Sistema de Seguridad.

3.3.1.8 **8** **9** Solicita configuración para la sirena y la guardaFunción: `int obtener_sirena ()`Función: `void config_sirena (int)`

Cuando se accesa a la opción SIRENA de las funciones especiales del teclado se hace uso de estas funciones. Proporciona la posibilidad de configurar el modo silencioso de la sirena, retorna la opción tecleada por el usuario: 1.- ON o 2.- OFF y guarda en la memoria EEPROM del ATMEGA16 la nueva configuración establecida por el usuario. De esta forma se determina si habrá un aviso auditivo al momento de activarse el Sistema de Seguridad.

3.3.1.9 10 Muestra toda la configuración del sistema

Función: void configuracion ()

La opción CONFIGURACIÓN de las funciones especiales del teclado invoca a esta función, en donde se lee de la EEPROM del ATMEGA16 y se despliega en la pantalla LCD la siguiente información:

- Tipo de los sensores.
- Tiempo de retardo.
- Configuración de la sirena y
- Modo de los sensores.

3.3.1.10 13 15 Solicita tiempo y código

Función: int obtener_tiempo ()
Función: int obtener_codigo ()

Para hacer uso de estas funciones se debe elegir del teclado matricial cualquiera de las opciones: TIEMPO o CÓDIGO; la función *obtener_tiempo()* es utilizada para la tecla TIEMPO y *obtener_codigo()* para la tecla CÓDIGO. Ambas funciones reciben datos desde el teclado y almacenan esta información para su posterior utilización en el acceso al sistema y el retardo a la activación del mismo. La función *obtener_tiempo()* retorna un valor entre 001 y 999, y *obtener_codigo()* tiene un rango entre 0000 y 9999.

3.3.1.11 14 16 Guarda el valor del retardo y nuevo código

Función: void cambiar_tiempo (int)
Función: void cambiar_codigo (int)

La función *cambiar_tiempo(int)* recibe el nuevo tiempo de retardo y la función *cambiar_codigo(int)* el nuevo código de acceso. Estas dos funciones guardan en la EEPROM del ATMEGA16 los nuevos valores para futuros tiempos de espera y claves de acceso al sistema.

El código del programa completo (*tesis.c*) se encuentra en un CD adjunto a éste documento dentro de la carpeta "Programa_AVR", las líneas son compiladas con ayuda del AVR Studio el cual genera un archivo titulado *tesis.hex*. Este archivo se descarga al microcontrolador ATMEGA16 por medio del software PG4UW y el programador universal BK PRECISION. El proceso se detalla en el ANEXO C.

3.4 MÓDULO DEL MICROCONTROLADOR DS80C400

El subsistema de interfaz remota cuenta con el módulo del microcontrolador DS80C400, dentro de este módulo se guarda el programa *TINIWebServer.tini* elaborado en código java y con el cual se logran los siguientes objetivos:

- Implementa un servidor Web.
- Monitorea el estado del Sistema de Seguridad.
- Establece la comunicación con el protocolo SPI entre el subsistema de interfaz local y remoto, para el envío y recepción de información.

3.4.1 Archivo TINIWebServer.tini

El archivo *TINIWebServer.tini* contiene la página principal para la manipulación del Sistema de Seguridad a distancia y surge de una serie de archivos hechos en java; se necesita el archivo con la extensión *.tini* para que al ser introducido al DS80C400 sea reconocido y pueda ser ejecutado. Para lo anterior se sigue el procedimiento ilustrado en la figura 3.5. La creación, descarga y ejecución de aplicaciones TINI se detalla en el ANEXO B.

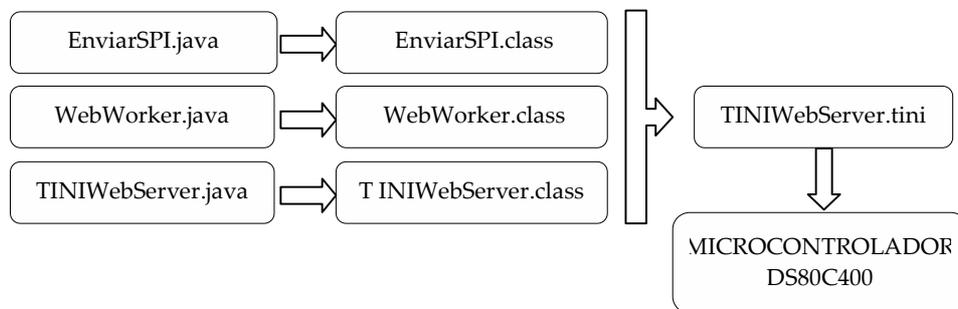


Figura 3.5. Programación del DS80C400.

Las clases generadas para que el Sistema de Seguridad funcione adecuadamente son las siguientes:

- EnviarSPI.- Establece la comunicación SPI.
- WebWorker.- Implementa un servidor Web.
- TINIWebServer.- Genera una página Web y hace uso de las clases EnviarSPI y WebWorker.

3.4.1.1 Clase: EnviarSPI

Esta clase construye un módulo SPI, hace uso de la clase SPI establecida en *com.dalsemi.comm*, de donde se utilizan un constructor y un método detallados a continuación.

public SPI()

Construye un objeto SPI con los estados de configuración por defecto, incluye: *DEFAULT_CLOCK_DELAY*, permite interrupciones durante la transferencia, usa la línea para selección de esclavo (*SS*) activa en bajo, usa la señal de reloj (*SCLK*) puesta en bajo, utiliza el bit más significativo para empezar la transmisión y el modo de 8 bits para transferencia de datos.

public int xmith(byte[] ba, int off, int len)

Escribe *len* bytes de datos sobre MOSI y retorna los datos simultáneamente leídos desde MISO. La fuente de los datos está contenido en el arreglo *ba* empezando en la posición *off*. Los datos leídos son retornados en la misma posición dentro del arreglo. Sus parámetros son:

ba : Contiene datos a ser escritos en el esclavo SPI.
off : Offset para empezar dentro del arreglo *ba*.
len : Número de bytes a escribir.

3.4.1.2 Clase: WebWorker

La clase *WebWorker* ejecuta un servidor Web, hace uso de la clase *HTTPServer* establecida en *com.dalsemi.tininet.http*. El constructor y los métodos utilizados para esta clase se explican a continuación.

public static final int DEFAUL_HTTP_PORT

Número de puerto por defecto: Puerto 80.

public HTTPServer(int httpPort)

Crea un *HTTPServer* usando el puerto *httpPort*. Su parámetro es el siguiente:

httpPort : Número de puerto para el servidor.

public void setIndexPage(java.lang.String indexPage)

Establece el índice de la página del servidor. El parámetro para este constructor es:

indexPage : Página a ser usada por el servidor como la página de índice por defecto.

public java.lang.String getHTTPRoot()

Devuelve la ruta HTTP del servidor.

```
public void setLogFilename(java.lang.String logFileName)
```

Establece el nombre del archivo de registro. El parámetro que recibe es:

logFileName : Nombre del archivo del registro.

```
public int serviceRequest()
```

Atiende las peticiones de los clientes HTTP y las solicitudes de los servicios de apoyo que se detecten.

```
public void setHTTPRoot(java.lang.String httpRoot)
```

Establece la raíz HTTP. Todos los archivos ubicados en el directorio de *httpRoot* pueden ser accedidos por las peticiones HTTP. Archivos entre la raíz del sistema de archivos y *httpRoot* no pueden ser accedidos con las peticiones HTTP. Su parámetro es:

httpRoot : Ruta que indica la raíz del servidor.

```
public java.lang.String getIndexPage()
```

Retorna la página de inicio predeterminada del servidor.

3.4.1.3 Clase: TINIWebServer

Esta clase es un servidor Web que utiliza la clase *com.dalsemi.system* junto con las generadas en los dos programas anteriores, *WebWorker.class* para habilitar el servidor HTTP y *EnviarSPI.class* para enviar los datos a través del puerto SPI. Los hilos usados en este programa son para responder las peticiones sobre el puerto 80 (TCP), para actualizar el índice de la página Web y para la comunicación por medio del puerto SPI.

La página Web que se genera con la clase *TINIWebServer* se hace por medio de una cadena de caracteres declarada en java, esta cadena incluye todos los parámetros establecidos por el código HTML. La página se divide en diferentes fragmentos para su fácil manipulación, así que se declaran varias cadenas y por medio de la clase *FileOutputStream* se interpretan estas cadenas como código HTML. El código del programa en java necesario para generar esta clase incluye todos los métodos y constructores que a continuación se describen.

```
public static final int HOUR_OF_DAY
```

Obtiene y establece la hora del día. Se utiliza para el reloj de 24 horas.

```
public static final int MINUTE
```

Obtiene y establece el minuto dentro de la hora.

```
public static final int SECOND
```

Obtiene y establece el Segundo dentro del minuto.

public FileOutputStream (File file)

Estos objetos son útiles para la escritura de archivos de texto. Primero se necesita abrir el archivo para después escribir en él. Una vez abierto el archivo, se pueden escribir bytes de datos utilizando el método *write()*.

file : Archivo a ser abierto para escribir.

public byte[] getBytes()

Codifica una cadena en una secuencia de bytes utilizando la plataforma de caracteres por defecto, almacena el resultado en una nueva cadena de bytes.

public int length()

Devuelve la longitud de una cadena.

public static String toString(int i) :

Convierte el entero que se pasa al método en su representación como cadena.

i : El entero a ser convertido.

public void write(String str, int off, int len)

Escribe la porción de una cadena.

str : La cadena.

off : Offset desde donde se van a empezar a escribir los caracteres.

len : Numero de caracteres a escribir.

public Thread(Runnable target)

Asigna un nuevo hilo.

target : Objeto cuyo método es llamado para su ejecución.

public static Calendar getInstance()

Devuelve una instancia de *GregorianCalendar* con la fecha y hora del momento.

public final void setName(String name)

Cambia el nombre de este hilo por uno igual al del argumento *name*.

name : El nuevo nombre para el hilo.

public string getWebRoot()

Obtiene la ruta del directorio del servidor Web.

public string getWebPage()

Obtiene la página por defecto del servidor Web.

La secuencia de los constructores y métodos utilizados en las tres clases anteriores, se muestran en el ANEXO D. El código completo de cada clase se encuentra guardado en el CD adjunto a éste documento, dentro de la carpeta "TINI".

Las clases: *EnviarSPI*, *WebWorker* y *TINIWebServer*; se obtienen a partir de sus correspondientes archivos escritos en código java y la forma más simple para generar las clases consiste en la ubicación de los archivos **.java* dentro de una carpeta (*src*) para compilarlos de la siguiente manera [URL 18]:

```
javac -bootclasspath %TINI_ROOT%\bin\tiniclases.jar -classpath
%TINI_ROOT%\bin\owapi_dependencies_TINI.jar; %TINI_ROOT%\bin\modules.jar;
%classpath% -d bin src\*.java
```

Una vez obtenidas las clases *EnviarSPI*, *WebWorker* y *TINIWebServer* deben adecuarse para su ejecución bajo el entorno desarrollado por MAXIM/Dallas Semiconductor, TINI o pequeña Interfaz de Internet. Para convertir estos archivos, ejecutables bajo este entorno de aplicación es necesaria la siguiente instrucción:

```
java -classpath %TINI_ROOT%\bin\tini.jar; %classpath% BuildDependency
-add SPI400 -add I2CALL -x %TINI_ROOT%\bin\owapi_dep.txt
-p %TINI_ROOT%\bin\owapi_dependencies_TINI.jar; %TINI_ROOT%\bin\modules.jar
-f bin -o bin\TINIWebServer.tini -d %TINI_ROOT%\bin\tini.db -add
OneWireContainer10; HTTPSERVER
```

Del comando anterior se obtiene el archivo *TINIWebServer.tini*, el cual se descarga en el módulo del DS80C400, no es necesario que se introduzcan los archivos **.java* o los **.class*. Hecho esto se ejecuta el programa desde una terminal de Telnet, lo anterior se detalla en la sección 3.4.3 y el ANEXO B.

Una vez que el programa se está ejecutando en el sistema, se puede acceder a él desde Internet con la dirección IP asignada al mismo. De esta forma se visualiza la página principal del Sistema de Seguridad generada por *TINIWebServer.tini* (figura 3.6); desde donde se pueden realizar diferentes tareas o bien desplegar información importante por medio de los vínculos ubicados dentro de su estructura.

En las siguientes secciones se detallan los vínculos a los que se pueden acceder desde la página principal del Sistema de Seguridad.

3.4.2 Hipervínculos Web

Como se observa en la figura 3.6 desde la página principal se accede a diferentes opciones del sistema, las cuales son:



Figura 3.6. Pantalla principal del Sistema de Seguridad

CONFIGURAR (*config.html*):-

Configura el tipo de cada uno de los sensores, el tiempo de retardo, el modo de la sirena y el estado del sistema.

REQUERIMIENTOS (*requer.html*):-

Indica los requerimientos mínimos de una computadora para tener acceso y manipular el Sistema de Seguridad a distancia.

AYUDA (*ayuda.html*):-

Contiene referencias para contactar al soporte técnico.

A continuación se describen las páginas Web creadas para la manipulación del Sistema de Seguridad a distancia. En el ANEXO F se presenta el esquema de cada una de estas páginas y sus códigos completos se muestran en el CD adjunto, dentro de la carpeta "Páginas".

3.4.2.1 Página: **config.html**

Al intentar acceder a ésta página es necesaria una clave de acceso, su contenido se presenta en la figura 3.7 en donde también se observan diversos hipervínculos: para la selección del sensor y posteriormente el tipo, para el establecimiento del tiempo de retardo, para el modo de la sirena y para el estado del sistema.



Figura 3.7. Pantalla para la configuración del Sistema de Seguridad.

En su estructura, la página contiene cuatro funciones:

1. Dependiendo de la opción elegida en la lista desplegable (-Seleccione el SENSOR-), al presionar el botón TIPO se abre una nueva página correspondiente al sensor seleccionado en donde se muestran los posibles modos de configuración: 01, 02 o 03.
2. Al dar un clic sobre RETARDO se abra una nueva ventana en donde se elije entre diferentes retardos (en segundos) para establecerle al sistema: 10, 20, 30, 40, 50, 60, 80, 100, 120, 150, 180 y 240.
3. Al presionar SIRENA se accesa a una nueva pantalla para configurar el modo audible (ON) o silencioso (OFF).
4. Si se necesita ARMAR al sistema o mantenerlo en REPOSO, es necesario elegir la opción de ESTADO en donde por medio de una nueva página se tiene control sobre esto.

Todas las ventanas que se abren a partir de la página *config.html* tienen un formato similar y sirven para enviar datos a través del puerto SPI, las páginas se muestran en las figuras 3.8, 3.9, 3.10 y 3.11 con sus diferentes opciones. Con cada una de esas opciones se ejecuta un archivo por lotes que es el encargado de enviar un dato a través del puerto SPI; esto se hace por medio de la siguiente función:

```
<A HREF= "../batch/ARCHIVO.bat ">OPCION</A>
```

en donde OPCION es la palabra que observa el usuario y *ARCHIVO.bat* es el nombre del archivo que contiene el dato a ser enviado por el puerto SPI para esa OPCION. El *ARCHIVO.bat* es de la siguiente forma:

```
cd c:\ss  
C:\java\bin\java -cp .;sshfactory.jar envioTelnet DATO
```

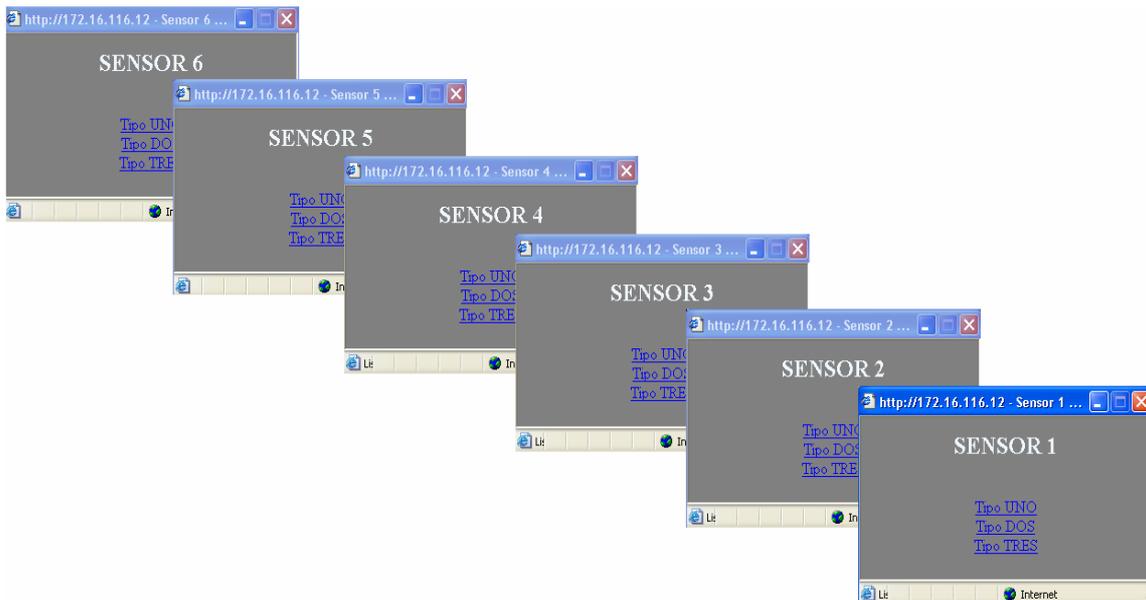


Figura 3.8. Opción: Selecciona el Sensor - TIPO

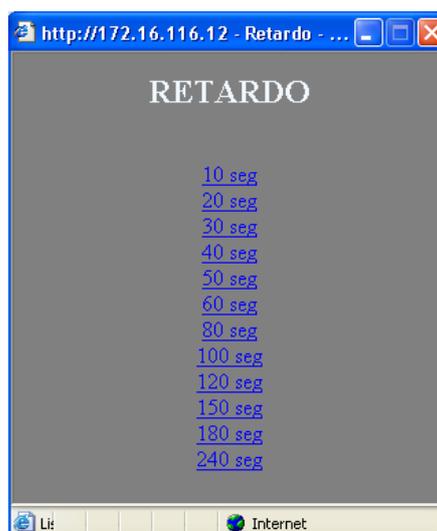


Figura 3.9. Opción: RETARDO



Figura 3.10. Opción: SIRENA

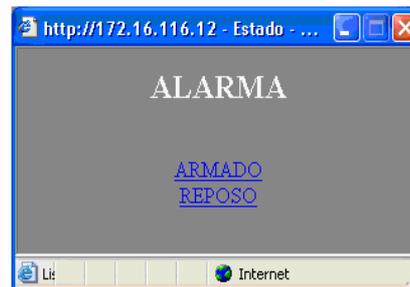


Figura 3.11. Opción: ESTADO

Por medio del programa *envioTelnet* automáticamente se abre la conexión Telnet, se introduce el usuario, la clave de acceso y se ejecuta un archivo que envía por el puerto SPI el DATO que es diferente para cada opción, los datos de cada una de las opciones se muestran en la tabla 3.2. Pero para que lo anterior tenga resultado, es necesario tener instalado SSH Factory que es un conjunto de componentes basados en Java para comunicarse con SSH y los servidores Telnet [URL 19].

3.4.2.2 Página: *requer.html*

Se tiene acceso a esta página por medio de la opción REQUERIMIENTOS desde la pantalla principal del Sistema de Seguridad y proporciona información acerca de los requerimientos de cómputo para poder manipular a distancia el sistema y visualizar su estado.

Desde este punto se puede descargar el SSH Factory si no se tiene instalado en la computadora, este archivo debe instalarse sobre la unidad C:\ss de lo contrario no será reconocido por el sistema y no se podrá enviar información por medio de la página Web a través del puerto SPI.

La figura 3.12 muestra la página *requer.html*, para ésta no se necesita clave de acceso debido a que la información que aquí se maneja no es relevante.

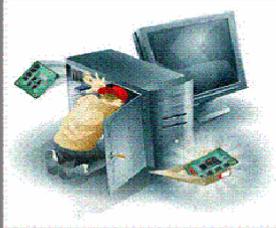
3.4.2.3 Pantalla: *ayuda.html*

Con un clic sobre la opción AYUDA de la pantalla principal del Sistema de Seguridad se accesa a la página mostrada en la figura 3.13, en donde se proporciona información del soporte técnico para cualquier duda o información acerca del sistema.

		DECIMAL	BINARIO								ASCII
SENSOR 1	1	65	0	1	0	0	0	0	0	1	A
	2	66	0	1	0	0	0	0	1	0	B
	3	67	0	1	0	0	0	0	1	1	C
SENSOR 2	1	70	0	1	0	0	0	1	1	0	F
	2	71	0	1	0	0	0	1	1	1	G
	3	72	0	1	0	0	1	0	0	0	H
SENSOR 3	1	75	0	1	0	0	1	0	1	1	K
	2	76	0	1	0	0	1	1	0	0	L
	3	77	0	1	0	0	1	1	0	1	M
SENSOR 4	1	80	0	1	0	1	0	0	0	0	P
	2	81	0	1	0	1	0	0	0	1	Q
	3	82	0	1	0	1	0	0	1	0	R
SENSOR 5	1	85	0	1	0	1	0	1	0	1	U
	2	86	0	1	0	1	0	1	1	0	V
	3	87	0	1	0	1	0	1	1	1	W
SENSOR 6	1	90	0	1	0	1	1	0	1	0	Z
	2	91	0	1	0	1	1	0	1	1	[
	3	92	0	1	0	1	1	1	0	0	\
RETARDO	10	33	0	0	1	0	0	0	0	1	!
	20	48	0	0	1	1	0	0	0	0	0
	30	35	0	0	1	0	0	0	1	1	#
	40	36	0	0	1	0	0	1	0	0	\$
	50	49	0	0	1	1	0	0	0	1	1
	60	50	0	0	1	1	0	0	1	0	2
	80	39	0	0	1	0	0	1	1	1	'
	100	40	0	0	1	0	1	0	0	0	(
	120	41	0	0	1	0	1	0	0	1)
	150	51	0	0	1	1	0	0	1	1	3
	180	52	0	0	1	1	0	1	0	0	4
ESTADO	240	44	0	0	1	0	1	1	0	0	,
	CONECTAR	63	0	0	1	1	1	1	1	1	?
	DESCONECTAR	95	0	0	1	1	1	1	1	0	>
SIRENA	OFF	54	0	0	1	1	0	1	1	0	6
	ON	55	0	0	1	1	0	1	1	1	7
ALARMA	APAGADA	56	0	0	1	1	1	0	0	0	8
	ENCENDIDA	57	0	0	1	1	1	0	0	1	9

Tabla 3.2. Datos a enviar por el puerto SPI para cada opción dentro del acceso al sistema por medio de Internet.

REQUERIMIENTOS



Aquí te decimos las características de tu computadora y te damos lo necesario para que puedas manipular a distancia el Sistema de Seguridad, con base en el Microcontrolador DS80C400

Características de la computadora

- Memoria RAM (mínima): 256 Mb
- Disco Duro (mínimo): 20 Gb
- Windows 98 o posterior
- Tarjeta de red

Software necesario

- Internet Explorer
- SSH Factory [DESCARGAR](#)

[Regresar](#)

Figura 3.12. Pantalla que muestra los requerimientos para la manipulación del Sistema de Seguridad a distancia.

AYUDA



Para saber más acerca del Sistema de Seguridad, con base en el Microcontrolador DS80C400 comunícate a la siguiente dirección de correo electrónico

alberto_di.or@hotmail.com

[Regresar](#)

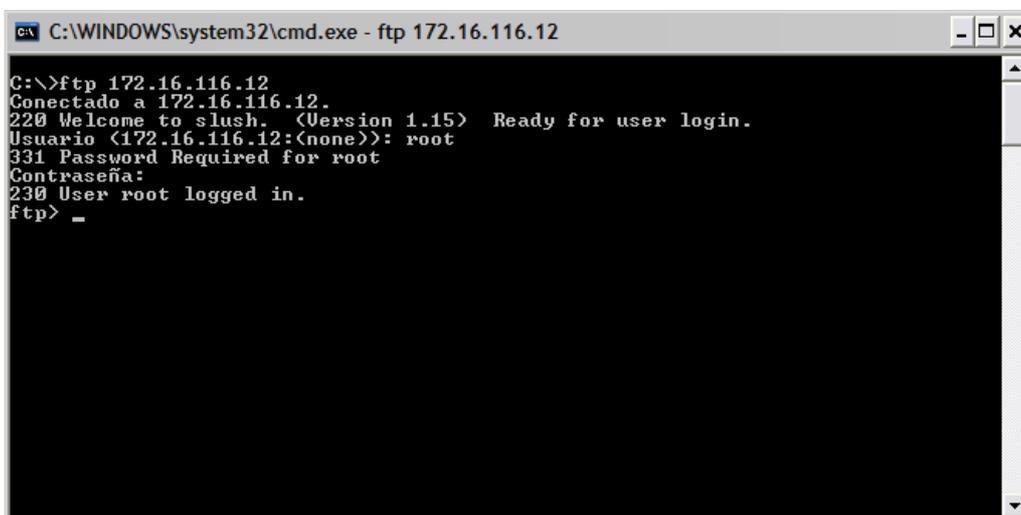
Figura 3.13. Pantalla con información para comunicarse con el soporte técnico del Sistema de Seguridad a distancia.

3.4.3 Manejo de datos en el Módulo del Microcontrolador DS80C400

Para introducir archivos al módulo del microcontrolador DS80C400 es necesario abrir una terminal de MS-DOS y teclear:

```
C:\ ftp DIRECCION_IP
```

donde DIRECCION_IP es el número que identifica de manera lógica y jerárquica a la interfaz del sistema dentro de la red, este número es fijo, para que de esta forma se permita su localización en la red (por ejemplo 172.16.116.12). Una vez ejecutado este comando aparecerá una pantalla como la de la figura 3.14 en donde se proporciona un *login* y *password* que para el primer caso es “root” y para el segundo “tini”.



```
C:\WINDOWS\system32\cmd.exe - ftp 172.16.116.12
C:\>ftp 172.16.116.12
Conectado a 172.16.116.12.
220 Welcome to slush. <Version 1.15> Ready for user login.
Usuario <172.16.116.12:(none)>: root
331 Password Required for root
Contraseña:
230 User root logged in.
ftp> _
```

Figura 3.14. Acceso al Sistema de Seguridad a distancia por medio de FTP.

Todo lo que se teclea después de *ftp>* afectará al sistema, siempre que sea un comando válido. Desde este punto se pueden quitar o añadir archivos del sistema. Para quitar archivos se utiliza el comando: *del* NOMBRE_ARCHIVO; y para añadir el comando: *put* NOMBRE_ARCHIVO. Antes de añadir un archivo es necesario establecer el tipo de transferencia binaria con la instrucción: *bin*. Como ejemplo se va a añadir y a borrar el archivo llamado *prueba.txt* ubicado en C:/ de la computadora desde donde se accede al ftp, el resultado se muestra en la figura 3.15.

Para tener un mejor panorama de lo que se puede hacer con ftp dentro del Sistema de Seguridad se construye la tabla 3.3 en donde se enlistan los comandos principales junto con su descripción.

```

C:\WINDOWS\system32\cmd.exe - ftp 172.16.116.12
C:\>ftp 172.16.116.12
Conectado a 172.16.116.12.
220 Welcome to slush. (Version 1.15) Ready for user login.
Usuario (172.16.116.12:(none)): root
331 Password Required for root
Contraseña:
230 User root logged in.
ftp> bin
200 Type set to Binary
ftp> put prueba.txt
200 Command successful.
150 BINARY connection open, putting prueba.txt
226 Closing data connection.
ftp>
ftp>
ftp>
ftp> del prueba.txt
250 File deleted.
ftp> _

```

Figura 3.15. Ejemplo para añadir y eliminar el archivo *prueba.txt* del Sistema de Seguridad a distancia.

COMANDO	DESCRIPCIÓN
ASCII	Establecer tipo de transferencia ascii.
BINARY	Establecer tipo de transferencia binaria.
BYE	Finalizar la sesión ftp y salir.
CLOSE	Finalizar la sesión ftp.
DELETE	Eliminar archivo remoto.
DIR	Mostrar el contenido del directorio remoto.
DISCONNECT	Finalizar la sesión ftp.
HELP	Imprimir información de ayuda local.
LCD	Cambiar el directorio de trabajo local.
LS	Mostrar el contenido del directorio remoto.
MKDIR	Crear directorio en el equipo remoto.
PUT	Enviar un archivo.
PWD	Imprimir el directorio de trabajo del equipo remoto.
QUIT	Finalizar la sesión ftp y salir.
RENAME	Cambiar el nombre del archivo.
RMDIR	Quitar directorio en el equipo remoto.

Tabla 3.3. Comandos principales para ftp.

Cuando el Sistema de Seguridad a distancia contenga la información necesaria para que funcione correctamente (todos los archivos explicados en este capítulo), se debe ejecutar la aplicación del servidor Web (*TINIWebServer.tini*) de lo contrario la página principal no podrá ser accesada y por consiguiente el sistema no puede ser manipulado ni se observará su estado. Esto se realiza con una sesión de Telnet. En la figura 3.16 se muestra la pantalla

de Telnet, su login es "root" y su password es "tini", ésta se abre desde una ventana de MS-DOS con la instrucción:

```
C:\> telnet DIRECCION_IP
```

Como se puede observar en la figura 3.16 la instrucción para ejecutar la aplicación del servidor Web es:

```
tini0707fa /> java TINIWebServer.tini &
```

Con lo anterior se tiene acceso al Sistema de Seguridad por medio de una página Web, desde Internet Explorer, Mozilla Firefox, Opera o algún otro explorador, con solo escribir en la barra de direcciones la DIRECCION_IP establecida al sistema, y de esta forma se accesa a él de forma gráfica.

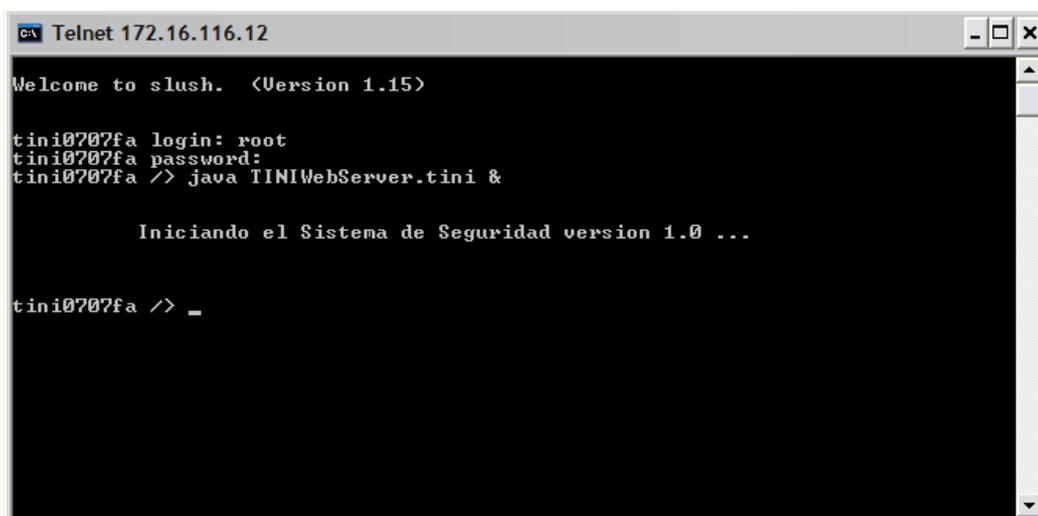


Figura 3.16. Pantalla de Telnet ejecutando la aplicación del servidor Web: *TINIWebServer.tini*.

Si se desea operar desde un sistema operativo UNIX, LINUX, etc. al modulo DS80C400 se puede instalar una maquina virtual sobre este Open Source, de esta forma se consigue el manejo de los datos como se ha mencionado en éste apartado.

En el siguiente capítulo se describe el funcionamiento del software exponiéndolo a los diferentes casos a enfrentar al momento de ser implementado.

4

PRUEBAS Y RESULTADOS

En esta sección se describen los resultados de la puesta en marcha del sistema para verificar su confiabilidad y robustez; exponiéndolo a los diferentes casos para los que fue elaborado como son restricción del acceso, configuración de los sensores y establecimiento del retardo. Se presentan las pruebas para diferentes casos de usabilidad y se comentan cada una de éstas con sus respectivos resultados. También se mencionan las características principales del sistema y algunos puntos importantes para obtener su mayor rendimiento.

4.1 ACCESO AL SISTEMA

El acceso al sistema puede ser en forma local o remota, en ambos casos la clave es diferente. En el acceso remoto se hace uso de una página Web diseñada específicamente para el Sistema de Seguridad y el acceso local es a través del subsistema de interfaz local y su teclado matricial adjunto a él.

Para configurar los sensores o la sirena, cambiar las contraseñas de acceso al sistema, modificar el retardo y activar o desactivar la alarma es necesario que se introduzca primero la clave de acceso. A continuación se muestra la forma de acceder al sistema en forma local y remota.

4.1.1 Acceso Local

Para el acceso de forma local existen diferentes claves (numéricas), las claves varían dependiendo del usuario y sus permisos (tabla 2.2). Cuando el sistema se encuentra en reposo, con la leyenda "SISTEMA DE SEGURIDAD", se introduce la clave correcta y se tiene acceso a las opciones establecidas en el teclado para la manipulación de la alarma (figura 4.1).

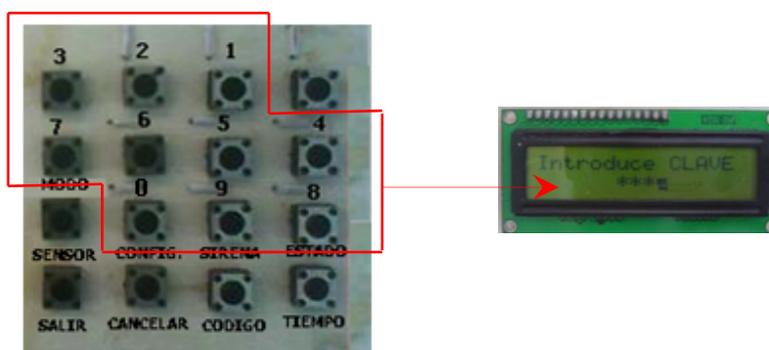


Figura 4.1. Solicitud de claves de acceso de forma local.

Si se necesita cambiar la clave de acceso por motivos de seguridad, el teclado cuenta con el botón (CÓDIGO) para cambiarla si tiene los privilegios. Se presiona el botón CÓDIGO entonces aparece el texto "CÓDIGO 0000-9999" en la pantalla LCD, por medio del teclado matricial se introducen los números del 0 al 9 hasta completar cuatro dígitos que será la nueva clave de acceso. Esta clave funcionará para la siguiente vez que se intente acceder al sistema.

La forma en que se presenta la solicitud de clave de acceso se visualiza en la figura 4.1 y en la figura 4.2 se muestran como modificarla.

4.1.2 Acceso Remoto

En el caso del acceso remoto la contraseña es alfanumérica. Para este método solo existen dos tipos de usuario: Instalador y Secundario, la solicitud "Introduce clave" se muestra al presionar el botón CONFIGURACIÓN de la página principal del Sistema de Seguridad. La forma en que se presenta la solicitud de la clave de acceso para el Instalador se observa en la figura 4.3.

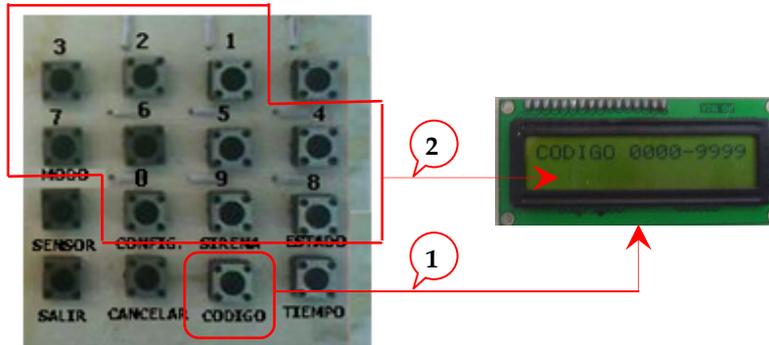


Figura 4.2. Cambio de clave de acceso de forma local.



Figura 4.3. Solicitud de clave de acceso de forma remota.

4.2 CONFIGURACIÓN DE SENSORES

Se debe tomar en cuenta que al cambiar la configuración de los sensores de forma remota los valores para los mismos son cambiados inmediatamente, en el caso del cambio en forma local éste se realiza al momento de salir del entorno del sistema.

4.2.1 Configuración Local

La configuración de los sensores en forma local se realiza por medio del teclado alfanumérico ubicado en el subsistema de interfaz local. Para configurar el modo se hace uso del botón "MODO" y el número de sensor que se desea configurar (del 1 al 6). La forma de realizar esta configuración es la mostrada en la figura 4.4.

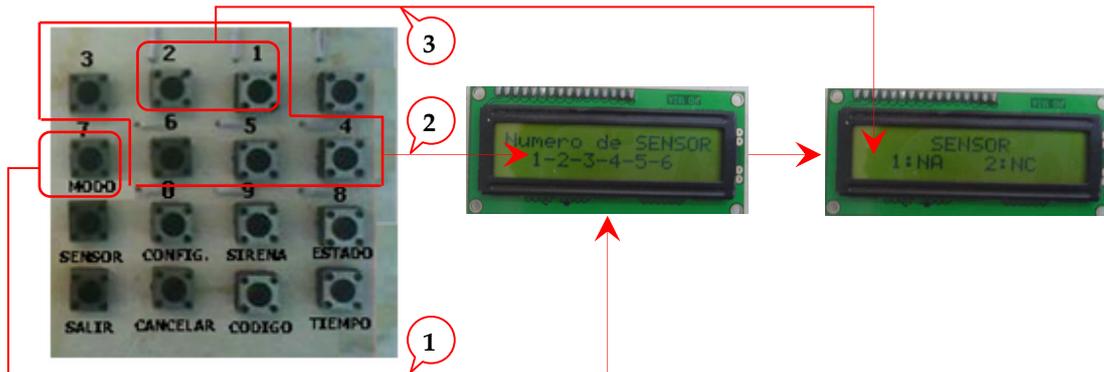


Figura 4.4. Configurar el modo de los sensores en forma local.

Y para configurar el tipo solo se tiene que oprimir el botón "SENSOR" y proporcionar los datos que en la pantalla se solicitan. La forma de hacerlo se muestra en la figura 4.5.

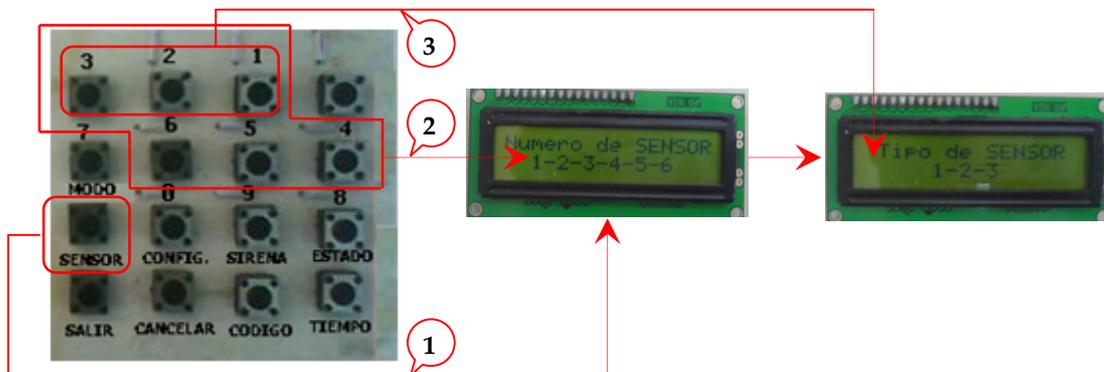


Figura 4.5. Configurar el tipo de los sensores en forma local.

4.2.2 Configuración Remota

Al configurar los sensores en forma remota, aparecen pantallas conforme se avanza en el proceso; la forma de hacerlo es simple ya que solo se elige la opción deseada e inmediatamente se envían los datos al sistema físico.

Para configurar el tipo de los sensores; en la pantalla principal del Sistema de Seguridad (figura 3.6) se da clic en la opción CONFIGURACIÓN y se desplegará una nueva pantalla (figura 3.7), en esta pantalla se selecciona el sensor y se presiona el botón TIPO, posteriormente aparecerá alguna de las pantallas correspondientes al número de sensor (figura 3.8) y se escogerá el tipo requerido.

4.3 TIEMPO DE RETARDO

El tiempo de retardo al configurarse de forma local tiene un rango entre 000 y 999 segundos, pero al configurarse de forma remota solo se puede escoger entre cantidades preestablecidas de tiempo.

4.3.1 Asentamiento Local

Para establecer el tiempo de retardo del sistema de forma local es necesario oprimir el botón "TIEMPO" ubicado en el teclado matricial del subsistema de interfaz local, y teclear el nuevo tiempo entre 000 y 999 segundos.

La figura 4.6 ejemplifica como hacer el cambio del tiempo de retardo de forma local.

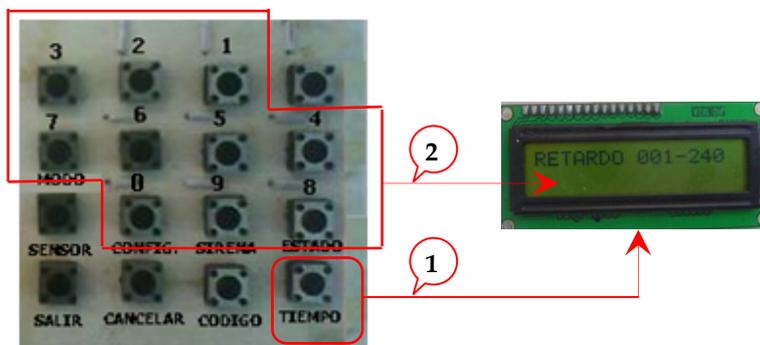


Figura 4.6. Cambiar tiempo de retardo en forma local.

4.3.2 Asentamiento Remoto

Dentro de la página Web principal del Sistema de Seguridad se encuentra la opción CONFIGURACIÓN, se le da clic y se accede a la pantalla titulada CONFIGURACIÓN, se escoge la opción RETARDO y aparecerá una nueva pantalla en donde se encuentran las 12 opciones diferentes de tiempos de retardo, entre 10 y 240 segundos (figura 3.9).

4.4 ESTADO DE LA SIRENA

Al momento de activarse la alarma, la sirena sonará si su configuración se lo permite, ésta se puede configurar de forma local o remota.

4.4.1 Configuración Local

Dentro del teclado matricial ubicado a un costado del subsistema de interfaz local existe un botón llamado “SIRENA”, al pulsarlo presenta un submenú para escoger el tipo de configuración deseada: ON para mantenerla activa y OFF para que se mantenga en reposo. La figura siguiente muestra como se hace la configuración y como aparece en la pantalla de LCD la información solicitada.

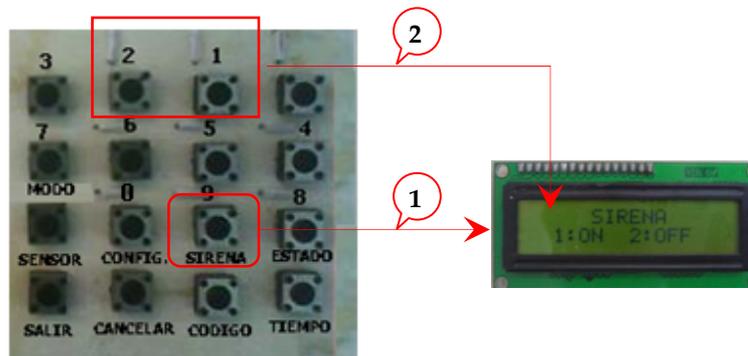


Figura 4.7. Configurar el estado de la sirena en forma local.

4.4.2 Configuración Remota

Para acceder a la configuración de la sirena de forma remota se tiene que seguir la siguiente ruta: Abrir la página Web principal del Sistema de Seguridad, dar clic en CONFIGURACIÓN – SIRENA y en la pantalla que se despliegue (figura 3.10) escoger ON u OFF e inmediatamente los datos son enviados y guardados en el subsistema de interfaz local.

4.5 ESTADO DEL SISTEMA

Se cuenta con dos led's para visualizar el estado de la alarma; el led rojo significa que el sistema está ARMADO, el led azul que esta REPOSO, y cuando ambos led's están apagados el sistema ha sido ACTIVADO.

Cuando el sistema está ACTIVADO, en forma remota se envía automáticamente información a la página Web principal del Sistema de Seguridad y aparece un mensaje de alerta, localmente se enciende la sirena si su configuración se lo permite y los dos led's (azul y rojo) ubicados en el subsistema de interfaz local se apagan.

Resumiendo lo anterior se tiene la tabla 4.1 en donde se especifica el led encendido para cada estado del sistema y el nuevo valor del estado al presionar el botón ESTADO.

ESTADO		LED			
Antes	Después	Antes		Después	
		ROJO	AZUL	ROJO	AZUL
ACTIVADO	REPOSO				SI
ARMADO	REPOSO	SI			SI
REPOSO	ARMADO		SI	SI	

Tabla 4.1. Estado del sistema antes y después de configurarlo.

4.5.1 Establecimiento Local

En el subsistema de interfaz local, por medio del teclado matricial adjunto a éste, se puede configurar el estado de la alarma con el botón ESTADO. Si el estado de la alarma es ARMADO el nuevo estado del sistema será REPOSO, si se encuentra en el estado REPOSO entonces tomará el valor de ARMADO y si el sistema está ACTIVADO el estado nuevo será REPOSO. Para realizar el cambio en el estado basta con pulsar el botón ESTADO como se muestra en la figura 4.8 y dependiendo del nuevo estado los led's rojo y azul se apagaran o encenderán.

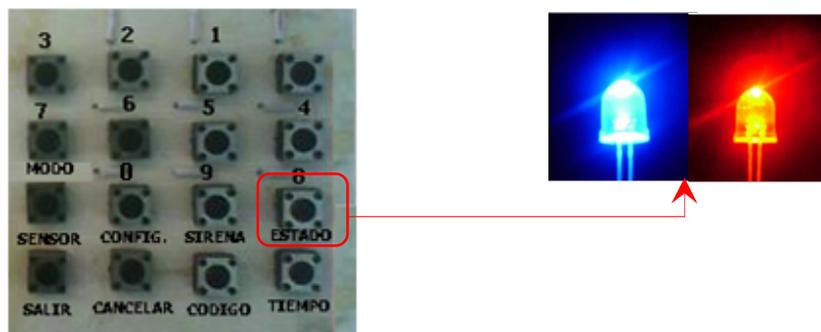


Figura 4.8. Cambios en el estado del sistema de forma local.

4.5.2 Establecimiento Remoto

Para hacer el cambio de estado en forma remota es necesario acceder a la ruta CONFIGURACION – ESTADO dentro de la página Web principal del Sistema de Seguridad y escoger cualquiera de las dos opciones que se nos presentan: ARMADO o REPOSO (figura 3.11). Localmente se cambiará el estado del sistema y de los led's rojo y azul. Las figuras 4.9, 4.10 y 4.11 muestran cómo se comporta el sistema en cada uno de los estados antes y después de la actualización del mismo.

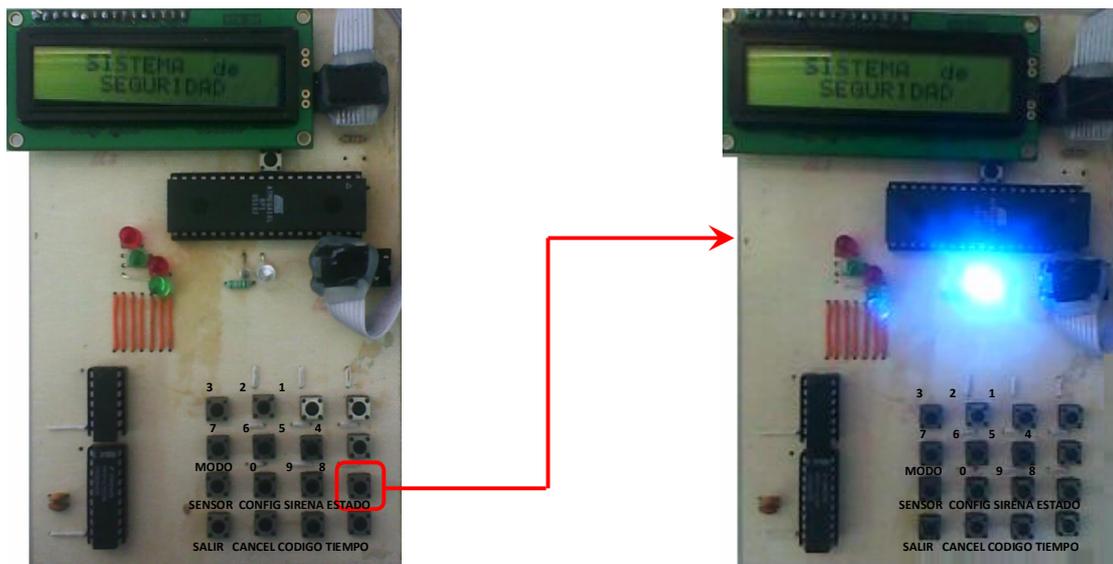


Figura 4.9. Cambios en el estado del sistema: ACTIVADO a REPOSO.

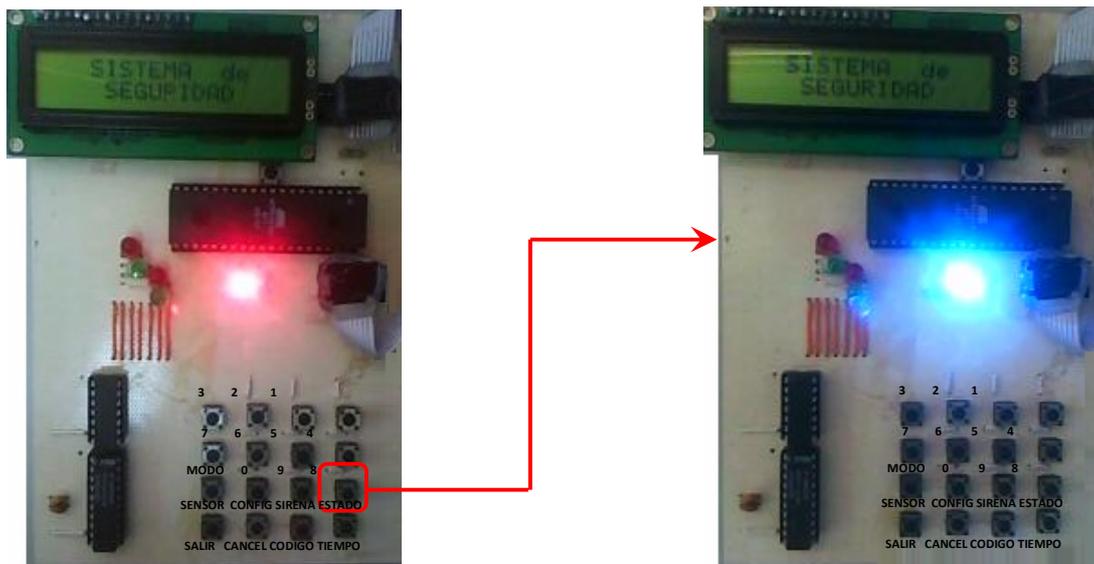


Figura 4.10. Cambios en el estado del sistema: ARMADO a REPOSO.

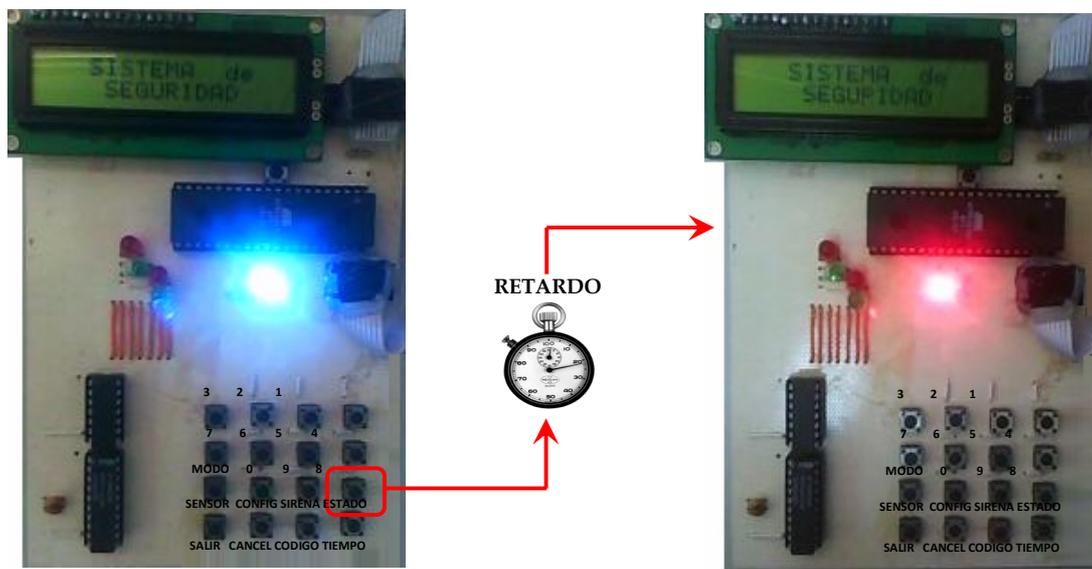


Figura 4.11. Cambios en el estado del sistema: REPOSO a ARMADO.

5

CONCLUSIONES

Al realizar éste proyecto de tesis acerca de un Sistema de Seguridad manejable por Internet, se obtuvieron resultados que permiten presentar el siguiente conjunto de conclusiones:

- La construcción de sistemas de propósito específico con conexión a internet no requieren de una computadora dedicada, sino que se puede construir con base en microcontroladores.
- La configuración y control local del sistema, monitoreo de sensores y activación del activador se realizó por medio de un programa en código C ejecutado desde el microcontrolador ATMEGA16.
- El monitoreo y acceso al sistema en forma remota se llevó a cabo por medio de una página Web contenida dentro del microcontrolador DS80C400.

- El intercambio de información entre la plataforma de evaluación del microcontrolador DS80C400 y el microcontrolador ATMEGA16 se consiguió a través del puerto SPI incluido en ambos microcontroladores.
- Se comprueba que el uso de protocolos estándares permite comunicar dispositivos de diferentes fabricantes operados con distintos lenguajes de programación.

Un obstáculo importante que se presentó durante la realización del Sistema de Seguridad fue que el manual proporcionado por Dallas Semiconductor para hacer funcionar la tarjeta de evaluación del DS80C400 está incorrecto; los archivos cargados dentro de la tarjeta no son los que en el manual y disco de instalación proporcionan sino unos obtenidos de la versión 1.15 de TINI. Los archivos son el *tini_400.tbin* y *slush_400.tbin* señalados en el ANEXO A.

Si en futuros proyectos se pretende dar otra aplicación a la comunicación SPI entre la tarjeta de evaluación del DS80C400 y un microcontrolador de la familia AVR, debe tomarse en cuenta la cantidad de información y de instrucciones que se van a manejar dentro del AVR ya que de eso depende el modelo a ocupar. Con el Sistema de Seguridad se ocupó el 97% de la memoria del ATMEGA16, para aplicaciones más complejas se puede hacer uso del ATMEGA32 o uno con mayores capacidades.

5.1 PERSPECTIVAS

No debe pensarse que con la implantación de éste sistema se tiene completamente asegurada algún área en específico, éste solo es un medio preventivo ya que únicamente avisa cuando alteran el estado en alguno de los sensores, así que no puede descartarse el factor humano. Se pueden añadir al sistema chapas electrónicas, cámaras IP, lector de huellas digitales y otros elementos para ampliar sus mecanismos de monitoreo, pero ninguno de estos podrá evitar por completo la intrusión de personas no autorizadas a las instalaciones.

Si las intenciones del lector es dar seguimiento a este proyecto, se podría implementar un diseño de hardware para que el Sistema de Seguridad sea inalámbrico, al igual que todos sus componentes. O bien hacer la comunicación de ambos microcontroladores por otro puerto, por ejemplo I2C o CAN. Si a sistemas de seguridad se refiere, podría existir al momento del acceso de los usuario una ruta específica para cada uno de ellos, así al teclear su clave, el usuario tendrá acceso a las instalaciones y por medio de chapas electrónicas el sistema marcará su ruta, cualquier desvío del camino será inútil ya que las puertas y ventanas estarán bloqueadas.

Con la combinación de la plataforma de evaluación del microcontrolador DS80C400 y el microcontrolador ATMEGA16 se pueden realizar de forma remota muchas actividades

que se realizan cotidianamente en un hogar; por ejemplo encender o apagar las luces, abrir o cerrar puertas, activar o desactivar la alarma, controlar aparatos eléctricos, entre otras cosas.

El comando a distancia hace posible que los expertos en algún área puedan aplicar sus conocimientos aún cuando se encuentren lejos de donde se requieran, como en telemedicina. Con los elementos utilizados en éste trabajo podría controlarse un brazo manipulador desde cualquier lugar con acceso a Internet.

Se puede intentar crear una aplicación para celulares, en donde por medio de un entorno gráfico accesen a la tarjeta y manipulen lo que se encuentra conectado a ella, por ejemplo una alarma para automóviles.

REFERENCIAS

- [1] "Sistemas de Seguridad Electrónicos". España : Orbis/Marcombo, 1986.
- [2] VILLARROEL SALCEDO, José Luis. "Sistemas Empotrados". Departamento de Informática e Ingeniería de Sistemas, Centro Politécnico Superior. Agosto, 2005.
- [3] OVIEDO EDISON, Rubén Alberto. "Sistema mínimo de propósito general basado en el microcontrolador DS80C400 con operación en un sistema de red". Universidad Tecnológica de la Mixteca. Julio, 2007.
- [4] STALLINGS, William. "Data and Computer Communications". España: Prentice Hall, 2000.
- [5] SNADER, Jon C. "Effective TCP". USA : Addison-Wesley , 2000.
- [6] LÓEZ PÓEZ, Eric. "Protocolo SPI (Serial Peripheral Interface)". Ingeniería en Microcontroladores, 2009.
- [7] JOYANES, FERNANDEZ. "JAVA 2, Manual de Programación". Osborne, McGraw-Hill, 2005.
- [8] GARCIA DE JALÓN, "Aprenda Java como si estuviera en primero". Escuela Superior de Ingenieros Industriales de San Sebastián, Universidad de Navarra. Febrero, 2000.
- [9] DELL, Tom. "Dynamic HTML for WebMasters". San Diego California: AP Professional, 1999
- [10] SEQUERA A, Richard. "Lenguaje C". Marzo, 2007.
- [11] ZEIGLER, Bernard. "Objects and Systems : Principled design with implementations in C++ and JAVA". USA: Springer -Verlag, 1996.
- [12] Fundación Código Libre Dominicano. "Artículo: Ventajas de código C". Santo Domingo, Santiago, 2005.
Stereon. "Catálogo de Productos, STEREN, Shop". 2009.
- [13] "Redes de Computadores Ethernet". México : Subdirección de Producción, 1993.
- [14] MACKENZIE, I. Scott "The 8051 Microcontroller". USA : Prentice-Hall, 1999.

- [15] PEATMAN, John B. "Design with MICROCONTROLLERS". USA : McGraw-Hill , 1988.
- [16] EMBREE, Paul M. \ Danieli, Damon, Coaut. "C++ Algorithms for Digital Signal Processing". New Jersey : Prentice-Hall, 1999.
- [17] BARR, Michael. "Programming Embedded Systems in C and C++". USA: O'Reilly, 1999.
- [18] POWELL, Thomas A. \ SANCHEZ GARCIA, Ignacio. "HTML 4: Manual de Referencia". Madrid: McGraw-Hill Interamericana, 2001

Sitios de Internet

- [URL 1] "Evolución de los Sistemas de Seguridad". Tema: Sistemas de Seguridad.
<http://www.maquinariapro.com/sistemas/sistema-de-seguridad.html>.
Última visita: 19 de octubre de 2007
- [URL 2] "Sistemas de Seguridad". Tema: Sistemas de Seguridad.
<http://www.sekiguchi.com.mx>. Última visita: 19 de octubre de 2007
- [URL 3] "Distribuidor de sistemas de emergencias". Tema: Sistemas de alarma.
<http://www.syscom.com.mx>. Última visita: 19 de octubre de 2007
- [URL 4] "Soluciones en Electrónica". Tema: Sistemas de seguridad.
<http://www.sterenshop.com.mx>. Última visita: 19 de octubre de 2007
- [URL 5] "How Stuff Works, Inc". Tema: Sistemas de seguridad.
<http://home.howstuffworks.com/burglar-alarm.htm>
Última visita: 10 de Febrero de 2009
- [URL 6] "Sensores de proximidad inductiva". Tema: Salidas NA y NC.
http://literature.rockwellautomation.com/idc/groups/literature/documents/ca/c115-ca523_-es-p.pdf
<http://www-lsi.die.upm.es/~carreras/ISSE/isse.html>
Última visita: 19 de octubre de 2007
- [URL 7] "Belt Ibérica S.A. Analistas de Prevención". Tema: Detección de incendios.
<http://www.belt.es/expertos/experto.asp?id=2781>
Última visita: 19 de octubre de 2007
- [URL 8] "El centro de tesis, documentos, publicaciones y recursos educativos más amplio de la Red". Tema: Protocolos TCP/IP.
<http://www.monografias.com/trabajos15/arquitectura-tcp/arquitectura-tcp.shtml>
Última visita: 12 de septiembre de 2007
- [URL 9] "Página personal de Felipe Santiago Espinosa". Tema: Sistemas Empotrados.
<http://mixteco.utm.mx/~fsantiag/>. Última visita: 2 de Enero de 2009.
- [URL 10] "MAXIM". Tema: DS80C400 Evaluation Kit
http://www.maxim-ic.com/quick_view2.cfm/qv_pk/4983
Última visita: 05 de Febrero de 2009.
- [URL 11] "Networked Microcontroller Offers the Integration Available in an 8051 Device". Tema: Microcontrolador DS80C400.
http://www.maxim-ic.com/quick_view2.cfm?qv_pk=3609.
Última visita: 31 de marzo de 2009.
- [URL 12] "Atmel Products – Microcontrollers". Tema: AVR® 8-Bit RISC.
<http://www.atmel.com/products/AVR/>. Última visita: 02 de Febrero de 2008.

- [URL 13] "Método de diseño de sistemas integrados orientado a objetos". Tema: Plataforma TINI.
http://www.cs.uic.edu/~jiparrag/publications/JIP_2004_01.pdf
Última visita: 08 de septiembre de 2007
- [URL 14] "Página personal de Felipe Santiago Espinosa". Tema: Arquitectura de Microcomputadoras.
<http://mixteco.utm.mx/~fsantiag/>. Última visita: 15 de Enero de 2009.
- [URL 15] "ALLDATASHEET". Tema: MM74C923N.
<http://www.alldatasheet.com/view.jsp?Searchword=MM74C923N>.
Última consulta: 20 de Septiembre de 2008.
- [URL 16] "ALLDATASHEET". Tema: JHD 162A
<http://www.alldatasheet.com/view.jsp?Searchword=JHD162A>
Última visita: 20 de Septiembre de 2008.
- [URL 17] "Wikitrónica: la wiki más electrónica". Tema: SPI.
[http://wikitronica.net/index.php/SPI+flujo+de+se%C3%B1ales+en+spi&cd=8\\$hl=es&ct=clnk&gl=mx&client=firefox-a](http://wikitronica.net/index.php/SPI+flujo+de+se%C3%B1ales+en+spi&cd=8$hl=es&ct=clnk&gl=mx&client=firefox-a). Última visita: 28 de Abril de 2008
- [URL 18] "MAXIM. Innovation Delivered". Tema: TINI® Network Microcontroller.
<http://www.maxim-ic.com/products/microcontrollers/tini/>. Última visita: 21 de Enero de 2009.
- [URL 19] "JSCAPE". Tema: SSH and Telnet Component Suite for Java.
<http://www.jscape.com/sshfactory/>. Última visita: 18 de Junio de 2009
- [URL 20] "ATMEL". Tema: AVR Studio
http://www.atmel.com/dyn/Products/tools_card.asp?tool_id=2725.
Última visita: 02 de Junio de 2009.
- [URL 21] "BK PRECISION". Tema: Programador Universal BK Precision.
<http://www.bkprecision.com/>. Último visita: 28 de Mayo de 2009.

ANEXO A. Manual del inicialización del sistema de evaluación del microcontrolador de red DS80C400

INICIANDO CON EL SISTEMA

Este anexo es de guía para el proceso de configuración inicial de la tarjeta de evaluación del DS80C400. Siguiendo las instrucciones siguientes, el usuario deberá establecer comunicación con el sistema y cargar el firmware. A continuación se incluye una lista de los materiales y software requeridos.

Requerimientos de Hardware

- El sistema mínimo basado en el microcontrolador DS80C400.
- Computadora Personal con Sistema Operativo que reúna los requerimientos del JDK de Sun Microsystems y con puerto RS232.
- Cable serial.
- Cable Ethernet cruzado.
- Fuente de alimentación de 6.5V DC.

Requerimientos de Software

- JDK (Java Development Kit) de Sun Microsystems version 1.4.2.
- Disponible en: <http://www.java.sun.com/javase/index.jsp> o <ftp:dalsemi.com/pub/tini/index.html>.
- Java Communications API de Sun Microsystems.
Disponible en: <http://java.sun.com/products/javacomm/index.jsp>.
- TINI Software Development Kit Version 1.1 o posterior.
Disponible en: <http://www.ibutton.com/TINI/software/index.html>.

Proceso de Inicialización

1. Instalando el JDK

El JDK es un programa ejecutable, se puede instalar en cualquier ubicación, la cual se referirá como *JAVA_HOME*. Una vez que la instalación se ha completado se puede ver la siguiente estructura de archivos.

```
06/18/2009 10:29 PM <DIR> bin
05/18/2009 07:00 AM 4,519 COPYRIGHT
06/18/2009 10:29 PM <DIR> demo
06/18/2009 10:29 PM <DIR> include
```

```

06/18/2009 10:29 PM <DIR>          jre
06/18/2009 10:29 PM <DIR>          lib
06/18/2009 10:29 PM          17,932 LICENSE
06/18/2009 10:29 PM          28,905 LICENSE.rtf
06/18/2009 10:29 PM          16,703 README.html
06/18/2009 10:29 PM           9,215 README.txt
05/18/2009 07:00 AM          11,849,754 src.zip
06/18/2009 10:29 PM          10,367 THIRDPARTYLICENSEREADME.txt

```

NOTA: Las fechas mostradas en el listado son las correspondientes a la fecha de instalación del JDK. Los nombres de archivo y tamaño mostrados son para la versión JDK 1.4.2.

2. Instalando las Comunicaciones API de Java

1. Descomprimir el archivo en cualquier directorio, el cual será consecuentemente referido como *COMMAPI*. En este directorio se encontrarán los siguientes archivos:

```

06/18/2009 03:59 PM          3,335 apichanges.html
06/18/2009 04:00 PM          28,043 comm.jar
06/18/2009 03:59 PM          8,141 COMM2.0_license.txt
06/18/2009 03:59 PM          5,374 CommAPI_FAQ.txt
05/18/2009 11:44 AM <DIR>          javadocs
06/18/2009 03:59 PM           467 javax.comm.properties
06/18/2009 03:59 PM          2,182 jdk1.2.html
06/18/2009 03:59 PM          3,715 PlatformSpecific.html
06/18/2009 03:59 PM          3,913 Readme.html
06/18/2009 03:59 PM          1,821 ReceiveBehavior.html
05/18/2009 11:44 AM <DIR>          samples
06/18/2009 04:00 PM          27,648 win32com.dll

```

2. Copiar win32com.dll de COMMAPI al directorio %JRE_HOME%\jre\bin.
3. Copiar comm.jar de COMMAPI al directorio %JRE_HOME%\jre\lib\ext.
4. Copiar javax.comm.properties de COMMAPI a %JRE_HOME%\jre\lib.
5. Ir a TINI_HOME y crear un archivo .BAT con la siguiente línea (descripción sensible a mayúsculas y minúsculas):

```
%JAVA_HOME%\bin\java -classpath %TINI_HOME%\bin\tini.jar JavaKit
```

6. Dar doble clic al archivo batch creado.

Esto pone a funcionar el JDK, el programa que se usa para interactuar con el sistema, como se muestra en la figura A.1. Se usará JDK frecuentemente, por eso el archivo batch ahorra tiempo. Si todo se instaló apropiadamente, se podrán ver los puertos COM que se tienen en la computadora.

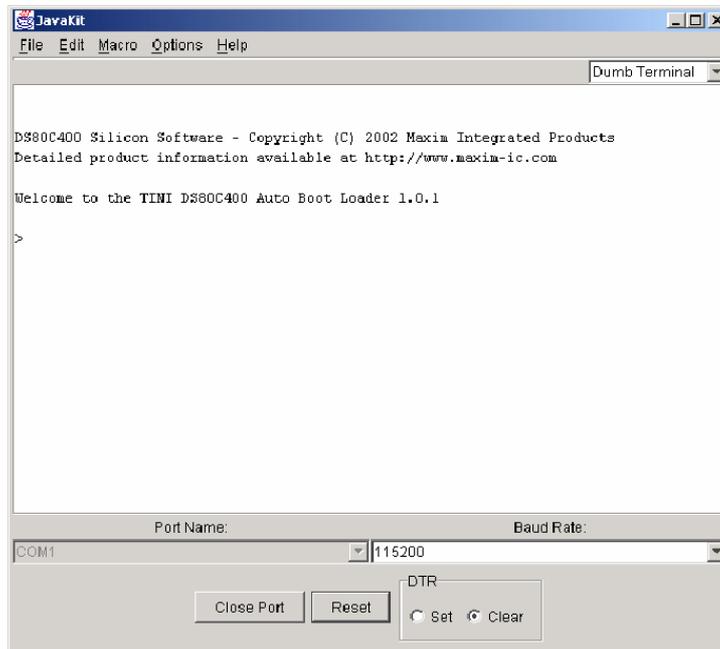


Figura A. 1.- Prompt del cargador JavaKit.

3. Conectando el sistema

1. Verificar que el sistema no esté alimentado.
2. Colocar el *jumper* DTR-Reset.
3. Conectar el cable SERIAL a la tarjeta.
4. Conectar el cable cruzado Ethernet al puerto Ethernet de la computadora, para enlazar la conexión entre la computadora y la tarjeta del DS80C400.
5. Conectar el adaptador de alimentación a la tarjeta del DS80C400.
6. Conectar el adaptador de alimentación a la toma de alimentación.

El resultado de estos pasos se muestra en la figura A.2.

4. Instalando el software TINI

Descomprimir el paquete en cualquier directorio, éste se referirá en adelante como TINI_HOME. En este directorio se deben encontrar estos archivos:

06/18/2009	04:35 PM	<DIR>	bin
06/18/2009	04:35 PM	<DIR>	docs
06/18/2009	04:35 PM	<DIR>	examples
05/15/2003	09:15 AM		3,413 index.html
05/15/2003	09:15 AM		13,178 License.txt
06/18/2009	04:35 PM	<DIR>	native

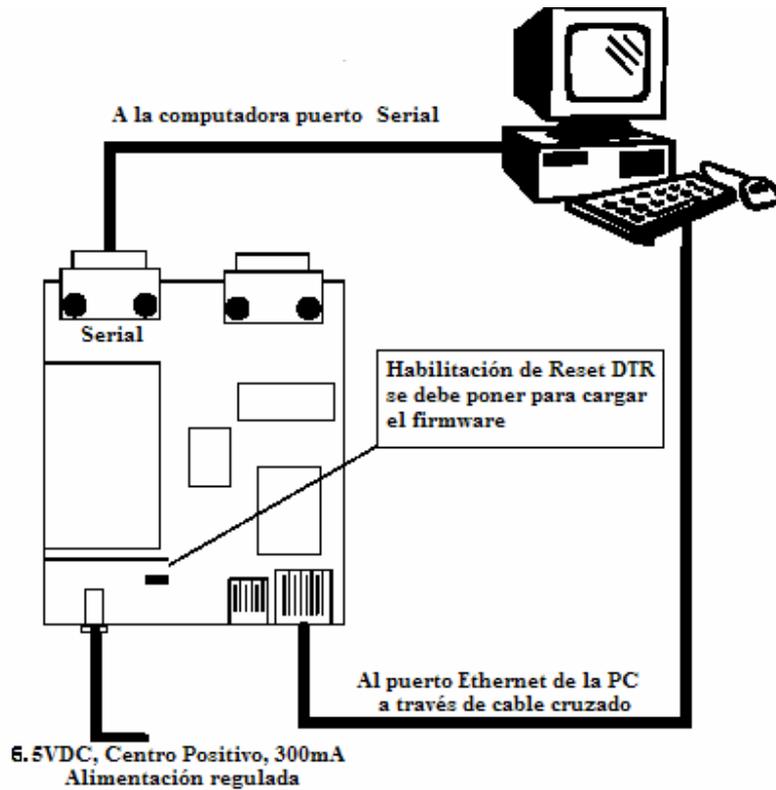


Figura A. 2.- Instalación típica del hardware.

```

05/15/2003  09:15 AM                18,050 README.txt
06/18/2009  04:35 PM   <DIR>          src
3 File(s)                    34,641 bytes
    
```

5. Cargando el Firmware

Hay tres cosas que se necesitan para descargar y ejecutar programas en el DS80C400. Lo primero es el firmware TINI, que está en el archivo %TINI-HOME%\bin\tini.hex. Después la biblioteca de las clases Java, que se encuentra en %TINI-HOME%\bin\tiniapi.hex. Finalmente, se necesita la línea de comandos interactiva que se encuentra en %TINI-HOME%\examples\slush\Slush.hex.

1. En JDK, seleccionar el puerto COM a usar para la entrada/salida serial. Este es cualquier puerto COM para conectar el sistema a la computadora.
2. Asegurarse que la velocidad sea de 115200.
3. Presionar el botón OPEN PORT.

4. Cuando el botón de RESET se habilite, presionarlo. Esto envía una señal de arranque al sistema. Este procedimiento inicia el "cargador" que permite comunicarse con el sistema y empotrar el firmware.
5. En el área de texto del prompt deben aparecer las siguientes líneas:

```
TINI loader 05-15-00 17:45  
Copyright (C) 2000 Dallas Semiconductor. All rights reserved.  
>
```

6. Seleccionar en el menú FILE la opción LOAD. Cargar el siguiente archivo: %TINI_HOME%\bin\tini.tbin. El archivo incluye una descripción de los bancos de información, por lo que no es necesario especificar la dirección de descarga. Debe reportar si el (los) banco (os) se cargó (aron) correctamente, lo que tomará algunos segundos.
7. Seleccionar FILE/LOAD una vez más y cargar el siguiente archivo: %TINI-HOME%\bin\slush.tbin.
8. Limpiar el área con lo siguiente:

```
> bank 18    //cambios para el banco 18  
> fill 0     //llena el banco 18 con 0's  
> bank 0     //cambios para banco 0  
> fill 0     //llena el banco 18 con 0's
```

9. Si todo se ha cargado, escribir 'e' para ejecutar. Se debe tener una salida similar a la que muestra la figura A.3.

Después de presionar cualquier tecla se debe escribir el nombre de usuario y posteriormente la contraseña de entrada:

```
Welcome to slush. (Version 1.01)
```

```
TINI login: root  
TINI password:
```

```
TINI />
```

NOTA: Hay dos cuentas de usuario por defecto, 'guest' con el password 'guest' y 'root' con password 'tini'

10. Escribir "help" para obtener ayuda. También se puede realizar un comando específico. Por ejemplo: "help ipconfig".
11. Ahora se puede introducir la dirección IP usando el comando *ipconfig*. Por ejemplo:

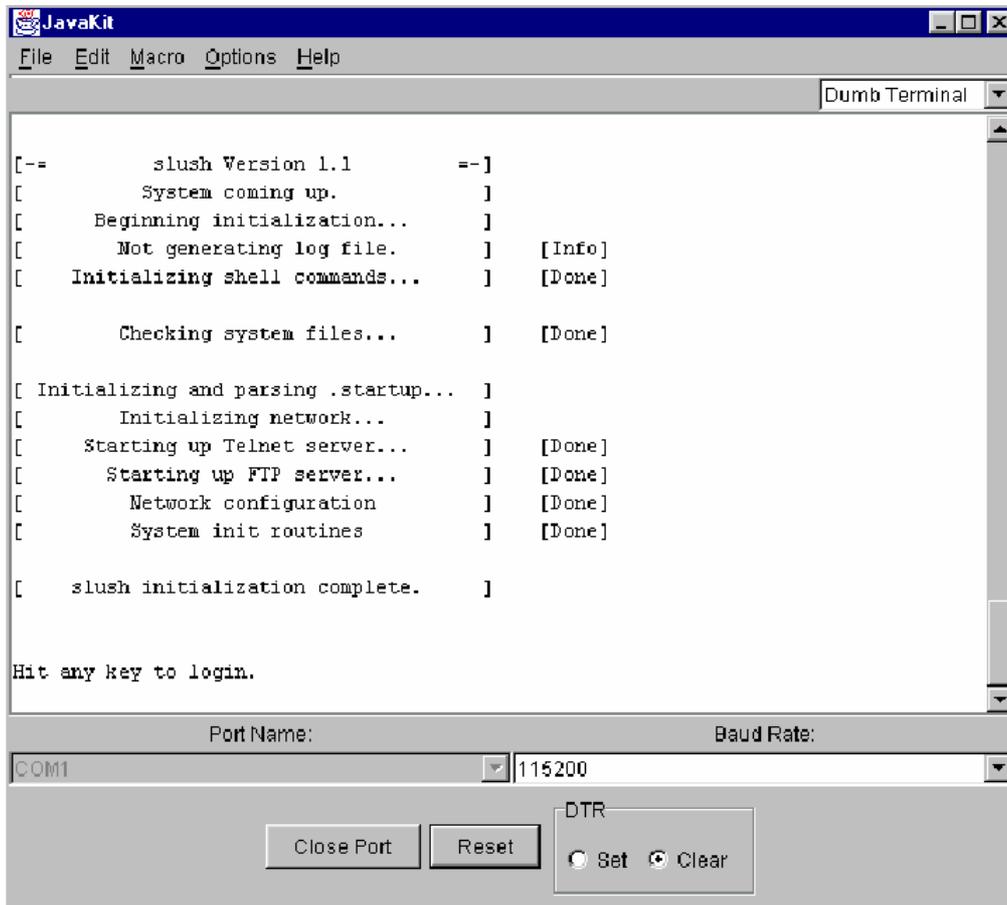


Figura A. 3.- Salida al cargar el firmware.

```
TINI /> ipconfig -a IP_del_sistema -m máscara_de_red
```

Nota: La opción -a se debe acompañar con la opción -m.

12. Si se pone la información de red correctamente usando ipconfig, se puede usar TELNET y FTP en el sistema. Ahora se puede probar con un ping al sistema para verificar que se encuentra en red.

NOTA: Una vez que se ha ingresado al sistema, todos los comandos son sensibles a mayúsculas y minúsculas.

6. Hecho

¡Se ha completado! El sistema está listo para funcionar.

ANEXO B. Creación, descarga y ejecución de aplicaciones TINI®

Este anexo es de ayuda para la realización de aplicaciones, como se carga y se ejecuta en el sistema de evaluación del DS80C400. Al igual que los lenguajes de programación para aprender se usa al aceptado programa “Hola Mundo”.

A través de una sesión TELNET, se corre la aplicación, se despliega la salida, se interactúa con los archivos de sistema y con los procesos de control.

B.1 Hola Mundo

Aunque este ejemplo no resalta exactamente las características de Java, es adecuado para describir el proceso paso a paso del desarrollo de una aplicación. Típicamente, para el desarrollo y prueba de aplicaciones se requieren 5 pasos:

1. Crear el archivo fuente.
2. Compilar el archivo fuente.
3. Convertir a archivo de clase.
4. Cargar la imagen convertida.
5. Ejecutar la imagen convertida.

Para la creación del archivo fuente se puede usar cualquier editor de texto o algún bloc de notas para programadores, por supuesto cuidando guardar el archivo con extensión *.java*, los siguientes tres pasos se realizan a través de la línea de comandos de MS-DOS. Y para ejecutar la aplicación se debe ingresar al sistema y usar su línea de comandos.

Paso 1: Crear el archivo fuente. Crear y guardar un archivo nombrado *HelloWorld.java* conteniendo el código de la figura B.1.

```
Class HelloWorld {  
    Public static void main (String[] args)  
    {  
        System.out.println("Hola Mundo");  
    }  
}
```

Figura B. 1.- Archivo fuente.

Paso 2: Compilando el archivo fuente: Se compila *HelloWorld.java* a un archivo de clase usando el compilador de Java, cambiando el prompt al directorio que contiene el archivo recién creado y se ejecuta el siguiente comando:

```
javac HelloWorld.java
```

Si el compilado produce error, generalmente se debe a que las clases de TINI® no están en el mismo directorio, entonces se debe especificar a través del parámetro “-bootclasspath” y así el compilador sabrá donde se ubican las librerías que se usan, como el siguiente comando:

```
javac -bootclasspath c:\tini1.12\bin\tiniclasses.jar HelloWorld.java
```

Si es satisfactorio, se tiene un nuevo archivo llamado *HelloWorld.class* en el directorio de trabajo actual.

Paso 3: Convirtiendo el archivo clase. Para este paso se usa la utilidad *TINIConvertor* para generar una salida binaria adecuada para la ejecución en TINI®. Esta aplicación está implícita en el TINI SDK (*tini.jar*) y se ejecuta en la línea de comandos.

TINIConvertor requiere de los parámetros: archivo fuente o directorio (-f), base de datos API (-d) y archivo de salida (-o).

```
java -classpath c:\tini1.02\bin\tini.jar TINIConvertor -f HelloWorld.class -d
c:\tini\tini1.02\bin\tini.db -o HelloWorld.tini
```

En el ejemplo, la aplicación consiste solo de una clase, mientras que la mayoría de las aplicaciones consisten de varias clases en uno o más paquetes y para ejecutarlas hay que cambiar el nombre por el del directorio de la raíz de los paquetes.

TINIConvertor genera un archivo con el nombre dado con la opción -o en este caso: *HelloWorld.tini*.

Paso 4: Cargar la imagen convertida. Se usa el cliente FTP, provisto con el sistema operativo, para conectar a TINI® y transferir la imagen binaria al sistema de archivos de TINI®.

```
C:\tini1.02\HelloWorld>ftp 172.16.116.12
Connected to 172.16.116.12.
220 Welcome to slush. (Version 1.02) Ready for user login.
User (172.16.116.12:(none)): root
331 root login allowed. Password required.
Password: ****
230 User root logged in.
ftp>
```

Después de establecer la conexión y registrarse, se puede transferir HelloWorld.tini. Primero se escribe "bin" en el prompt de FTP para asegurar que la imagen binaria no se altere en la transferencia.

```
ftp> bin
200 Type set to Binary
```

Se transfiere usando el comando put.

```
ftp> put HelloWorld.tini
200 PORT Command successful.
150 BINARY connection open, putting HelloWorld.tini
226 Closing data connection.
ftp: 171 bytes sent in 0.00Seconds 171000.00Kbytes/sec.
```

Finalmente se cierra la sesión FTP escribiendo *bye* o *quit* en el prompt.

```
ftp> bye
221 Goodbye.
```

Es posible revisar que el archivo se transfirió satisfactoriamente usando el comando *ls* en el prompt de la sesión TELNET.

```
TINI /> ls -l
total 3
drwxr-x 1 root admin  9 Mar 28 14:45.
-rwxr-- 1 root admin 27 Mar 28 15:46 HelloWorld.tini
drwxr-x 1 root admin 10 Mar 28 14:45 etc
```

El archivo *HelloWorld.tini* ahora aparece en el directorio raíz del sistema de archivos.

Otra forma de hacer la transferencia sin usar línea de comandos, y que muchos desarrolladores prefieren, es con archivos creados con las instrucciones. Por ejemplo, con el Windows FTP, se crea un archivo con el contenido de la figura B.2.

```
root
tini
bin
put HelloWorld.tini
bye
```

Figura B. 2.- Load.cmd.

Si se le llama `load.cmd` se usa lo siguiente para transferir el archivo sin ninguna interacción con el prompt de comando de FTP.

```
C:\TINI\tini1.02\myapps\HelloWorld>ftp -s:load.cmd 172.16.116.12
```

Usando la opción `-s` el cliente FTP lee el archivo especificado y ejecuta cada línea como si se escribiera manualmente en el prompt.

Paso 5: Ejecutando la imagen convertida. Ahora sólo hay que ejecutar la aplicación usando el comando `java` en la línea de comandos dentro del sistema. Para ingresar al sistema se debe abrir una sesión TELNET con la dirección del sistema.

```
C:/>telnet 172.16.116.12
```

Después de ingresar el nombre de usuario y contraseña se tiene la línea de comandos del sistema y se puede ejecutar la aplicación.

```
TINI/>java HelloWorld.tini
Hola Mundo
TINI/>
```

HelloWorld.tini realizó y produjo la salida esperada. Al terminar el programa, el control de la sesión del usuario regresa a la línea de comandos.

ANEXO C. Creación y descarga de archivos .hex al ATMEGA16

Para crear un archivo en código C que sirva para manipular las entradas y salidas de un microcontrolador, se hace uso del entorno de desarrollo integrado AVR Studio, desarrollado por ATMEL [URL 20]. Las líneas de código compiladas por medio de este programa generan un archivo con extensión .hex. Este archivo se puede descargar al microcontrolador por medio del software PG4UW desarrollado para el programador universal BK PRECISION [URL 21].

Al abrir el AVR Studio se tiene que seleccionar el modo de programación: C o Ensamblador, y el elemento al que se le pretende incluir el código después de ser compilado. Para este caso el modo de programación es C y el elemento un microcontrolador ATMEGA16. La figura C.1, es una imagen del entorno AVR Studio, se encuentra abierto el programa *tesis.c* y para compilarlo basta con ir a la pestaña Build-Build o bien pulsar la tecla F7. Si no existen errores se crea en la ruta actual un archivo titulado *tesis.hex*, en la parte inferior de la pantalla muestra la cantidad y ubicación de los errores si es que existen, o bien, el porcentaje de memoria que ocupó el programa en cada una de las memorias del microcontrolador. En este caso se ocupó el 91.2% de la memoria de Programa y el 37.5% de la memoria de Datos del ATMEGA16.

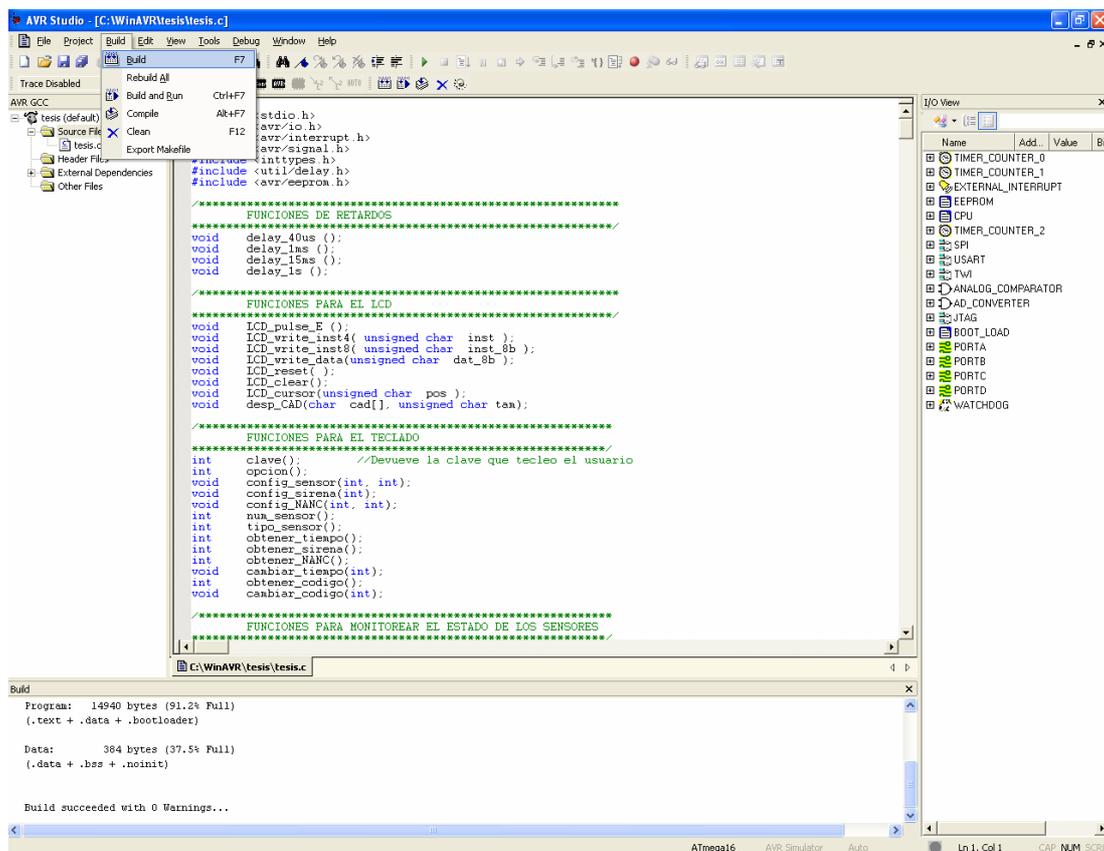


Figura C.1.- Programación en AVR Studio de *tesis.c*.

Ya que se generó el archivo *tesis.hex* por medio del AVR Studio, la forma de descargarlo al microcontrolador se muestra en la figura C.2 y en la figura C.3 se observa la manera de configurar el tipo de programación que utilizara el PG4UW para el ATMEGA16, esta pantalla es importante ya que si no se configura de forma correcta, el microcontrolador no guardaría datos en su memoria EEPROM o no le haría caso a las interrupciones. Tómese en cuenta que el programador universal debe estar correctamente conectado al puerto de la computadora y que al abrir el programa que lo manipula, PG4UW, se debe especificar el elemento a programar, para este caso es el ATMEGA16.

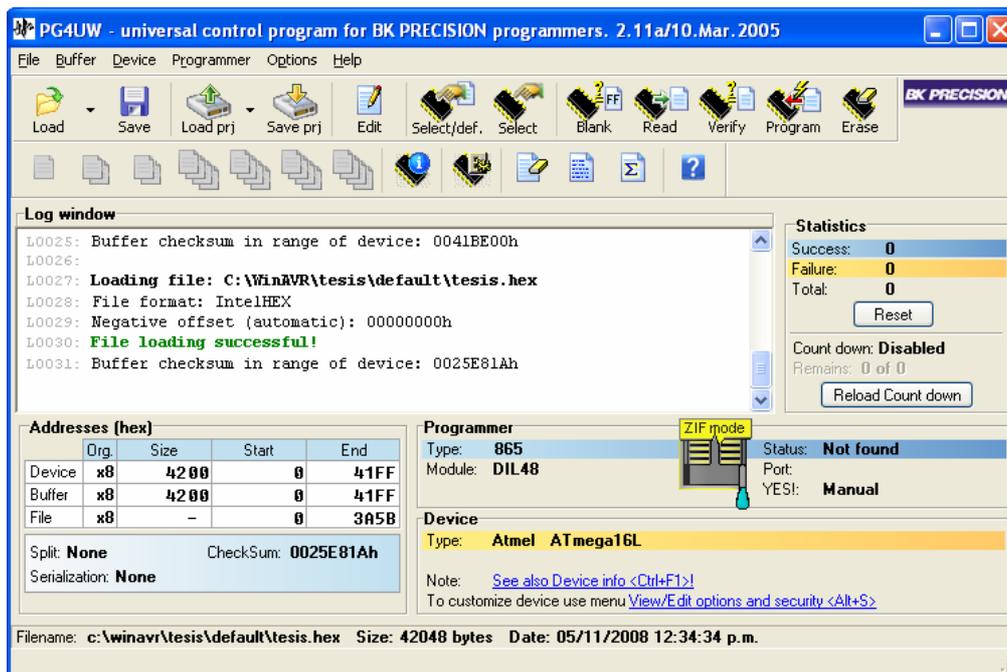


Figura C. 2.- Descarga de *tesis.hex* al ATMEGA16 con PG4UW.

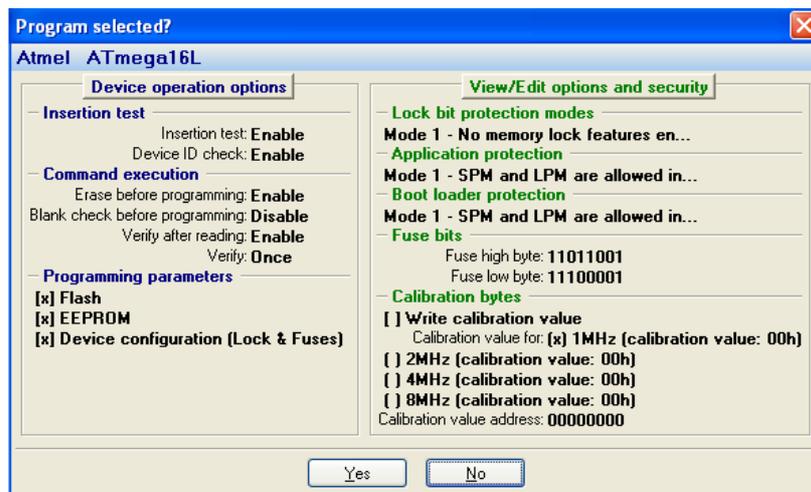
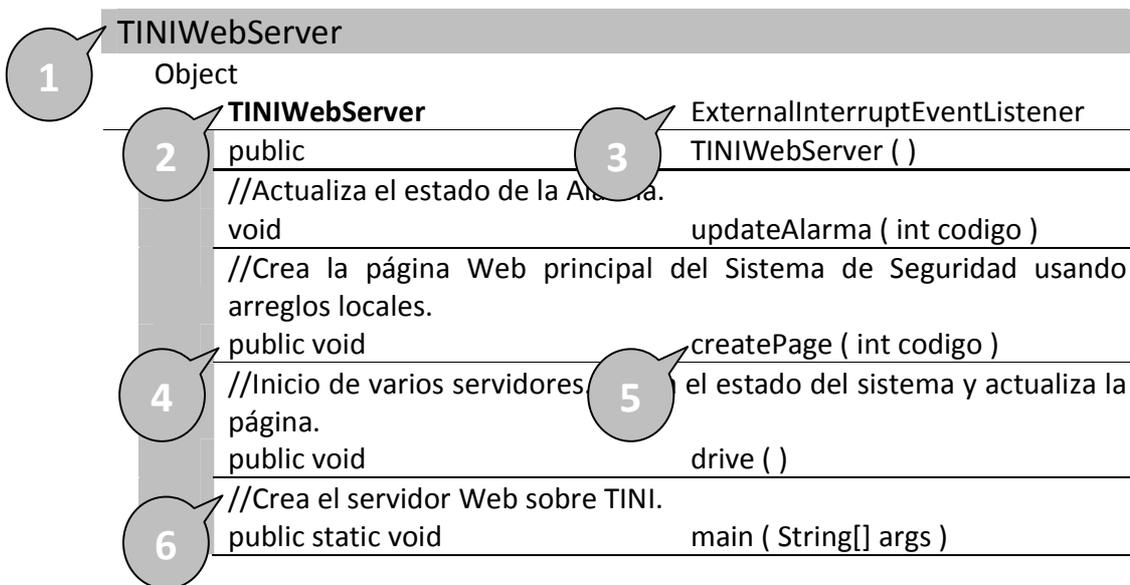


Figura C. 3.- Parámetros para programar el ATMEGA16.

ANEXO D. Clases generadas para manipular a distancia el Sistema de Seguridad

Estas son las clases realizadas para controlar a distancia el Sistema de Seguridad.



- 1.- El nombre de la clase
- 2.- La cadena de superclases. Cada clase es una subclase de la que le precede.
- 3.- Los nombres de las interfaces implementadas por cada clase.
- 4.- El tipo de variable que regresa un método o el tipo de declaración de una variable.
- 5.- El nombre de la clase miembro. Si es un método, la lista de parámetros y la cláusula opcional.
- 6.- Descripción de la función que realiza la clase miembro.

Clase: EnviarSPI

Esta clase sirve para comunicar al subsistema de interfaz local con el subsistema de interfaz remota por medio del puerto SPI.



public	EnviarSPI ()
//Ejecuta el hilo de la transferencia SPI.	
public void	run ()

Clase: WebWorker

Para que se tenga acceso al Sistema de Seguridad en forma remota por medio de una página Web, es necesaria la implementación de ésta clase.

WebWorker	
Object	
WebWorker	Runnable
public	WebWorker ()
//Obtiene el directorio raíz del servidor Web.	
string	getWebRoot ()
//Obtiene la página por defecto del servidor Web.	
string	getWebPage ()
//Ejecuta el hilo del servidor.	
void	run ()

Clase: TINIServer

Esta clase genera una página Web y junto con las clases EnviarSPI y WebWorker, habilita un servidor Web y envía datos por el puerto SPI.

TINIServer	
Object	
TINIServer	ExternalInterruptEventListener
public	TINIServer ()
//Actualiza el estado de la Alarma.	
void	updateAlarma (int codigo)
//Crea la página Web principal del Sistema de Seguridad usando arreglos locales.	
public void	createPage (int codigo)
//Inicio de varios servidores. Checa el estado del sistema y actualiza la página.	
public void	drive ()
//Crea el servidor Web sobre TINI.	
public static void	main (String[] args)

ANEXO E. Programa generado para manipular localmente el Sistema de Seguridad

La secuencia de instrucciones dentro del ATMEGA16 para manipular a distancia el Sistema de Seguridad es la siguiente.

tesis.c

```
//Servicio a la interrupción: Se validan los datos recibidos por el puerto SPI
ISR (SPI_STC_vect)
//Crea el control a distancia del Sistema de Seguridad
int main ( void )
//Inicializa el sistema escribiendo en la posición m de la EEPROM el valor de n
void eeprom_write_byte( m , n )
//Inicializa el ATMEGA16 como esclavo para la comunicación SPI
void slave_init ()
//Empieza ciclo infinito: el sistema está en espera de cualquier actividad realizada
por el usuario, de forma local o remota.
//Empieza ciclo finito: mientras el usuario no pida salir del sistema o bien
mientras no se termine el tiempo de configuración.
//Monitorea el estado de los sensores
void monitorear ( )
//Actualiza el estado de la alarma para enviarla por SPI
void actualizaSPI ()
//Solicita la clave de acceso y retorna su valor
int clave ( )
//Lee de la EEPROM las claves de usuario y verifica si la clave es correcta
int eeprom_read_byte( n )
//Empieza ciclo finito: mientras el usuario se mantenga dentro del sistema
//Retorna la opción del teclado matricial seleccionada por el usuario,
mientras la clave "usuario" se lo permita
int opcion ( usuario )
//Dependiendo de la opción seleccionada por el usuario el programa le
hará caso a la petición
//Si la opción es: MODO. Se pide el numero de sensor: 1, 2, 3, 4, 5 o
6, el modo: NA o NC y se guarda en la EEPROM.
int num_sensor ( )
int obtener_NANC ( )
void config_NANC ( num_sensor, modo )
//Si la opción es: ESTADO. Lee el estado actual del sistema y lo
cambia de reposo-armado, armado-reposo y activado-reposo.
int eeprom_read_byte ( estado )
```

```

void                                eeprom_write_byte ( estado, nuevo )
//Si la opción es: SIRENA. Obtiene el estado actual de la sirena y lo
//modifica de on-off u off-on.
int                                  obtener_sirena ( )
void                                config_sirena ( sirena )
//Si la opción es: CONFIGURACIÓN. Se muestra en pantalla la
//configuración actual de los sensores, sirena y retardo.
void                                configuracion ( )
//Si la opción es: SENSOR. Solicita el numero de sensor: 1, 2, 3, 4, 5,
//6, el tipo: 01, 02, 03 y se guarda en la EEPROM.
int                                  num_sensor ( )
int                                  tipo_sensor ( )
void                                config_sensor ( sensor, tipo )
//Si la opción es: RETARDO. Solicita al usuario el tiempo de retardo
//y se guarda en la EEPROM.
int                                  obtener_tiempo ( )
void                                cambiar_tiempo ( tiempo )
//Si la opción es: CODIGO. Solicita nuevo código y es guardado en la
//EEPROM.
int                                  obtener_codigo ( )
void                                cambiar_codigo ( código , usuario )
//Si la opción es: SALIR. Los dos ciclos finitos se terminan.
//Se terminan las opciones del teclado.
//Termina el segundo ciclo finito.
//Termina el primer ciclo finito.
//Termina el ciclo infinito.

```

ANEXO F. Hipervínculos Web

La página principal del Sistema de Seguridad cuenta con hipervínculos para su CONFIGURACIÓN a distancia, establece los REQUERIMIENTOS necesarios de nuestra computadora para poder manipularlo y de AYUDA para saber cómo contactar con el soporte técnico. Los códigos que generan los hipervínculos son los siguientes:

configuración.html	
Titulo	
Configuración	
	//Se genera una función para abrir la pantalla correspondiente al sensor seleccionado y configurar su modo: 01, 02 o 03. void enviarsensor ()
	//Se establece una función para abrir la pantalla de los tiempos de retardo disponibles para el sistema. void enviartiempo ()
	//Con esta función se abre la pantalla de la sirena para establecer su estado: ON u OFF. void sirena ()
	//Se declara una función para abrir una página en donde se puede escoger el estado del sistema: reposo o armado. void estado ()
	//Si se presiona el botón: TIPO, se despliega le numero de sensor seleccionado con sus posibles estados a configurar. onClick enviarsensor ()
	//Si se presiona el botón: RETARDO, se despliegan en una ventana los posibles valores para el retardo del sistema. onClick enviartiempo ()
	//Si se presiona el botón: SIRENA, se puede cambiar la configuración de la sirena: encendida o apagada. onClick sirena ()
	//Si se presiona el botón: ESTADO, en la pantalla generada se elige la opción deseada. onClick estado ()
	//Si se presiona el botón Regresar, la página vuelve a la pantalla principal.

requerimientos.html

Título

Requerimientos

//Se despliegan las características necesarias de la PC para poder manipular a distancia el Sistema de Seguridad.

//Si se presiona el botón DESCARGAR, se descarga el software: sshfactory.

//Si se presiona el botón Regresar, la página vuelve a la pantalla principal.

ayuda.html

Título

Ayuda

//Muestra la información necesaria para contactar con el soporte técnico.

//Si se presiona el botón Regresar, la página vuelve a la pantalla principal.

ANEXO G. Lista de materiales y circuito impreso

La tabla G.1 es la lista de materiales necesarios para armar el Sistema de Seguridad manipulable por Internet:

ABREVIATURA	CANTIDAD	DISTRIBUIDOR No. PARTE	DESCRIPCIÓN	COSTO UNITARIO (MN)
PF1	1	STEREN PC-10X15	Placa fenólica de 10x15.	\$ 16.00
PF2	1	STEREN PC-15X15	Placa fenólica de 15x15.	\$ 23.00
B8	3	STEREN IC08P	Base para circuito integrado de 8 patas.	\$ 3.00
B14	1	STEREN IC14P	Base para circuito integrado de 14 patas.	\$ 3.00
B20	1	STEREN IC20P	Base para circuito integrado de 20 patas.	\$ 4.00
B40	1	STEREN IC40P	Base para circuito integrado de 40 patas.	\$ 6.00
LD	6	STEREN E5/COLOR-C	Led's diferentes colores.	\$ 3.00
R1	8	STEREN R10M 1/2	Resistencias de 10 MΩ.	\$ 1.00
R2	1	STEREN R500 1/2	Resistencia de 500 Ω.	\$ 1.00
C1	1	STEREN C22-500	Capacitor 22 pF.	\$ 3.00
C2	1	STEREN C220-500	Capacitor 220 pF.	\$ 3.00
PB	17	STEREN AU-1012	Micro switch de push, con dos terminales.	\$ 4.00
DT	1	AG ELECTRONICA MM74C923N	Decodificador de teclado	\$ 80.00

CN	1	AG ELECTRONICA DM74LS04N	Compuerta NOT	\$ 7.00
MC	1	AG ELECTRONICA ATMEGA16	Microcontrolador ATMEGA16	\$ 80.00
MR	1	MAXIM KIT-DS80C400	Tarjeta de Evaluación del microcontrolador de red DS80C400.	\$ 1200.00
TT	8	STEREN TRT-03	Terminales chicas con tres tornillos, para circuito impreso.	\$ 11.00
CH	1	STEREN FCC220-16	Conector hembra para cable plano, doble línea y 8 vías.	\$ 5.00
CC	3	STEREN MCC220-16	Conectores para cable plano, 2 líneas y 8 vías.	\$ 5.00
CP	2	STEREN CABLE-16	Metros de cable plano de 16 hilos.	\$ 3.00
CE	1	STEREN CABLE-E	Cable para Ethernet.	\$ 3.00
LCD	1	AG ELECTRONICA JHD162A	Pantalla LCD JHD 162A.	\$ 120.00
SIR	1	SYSCOM R58GS	Sirena de 92 dB y alimentación de 8 Volts.	\$ 200.00
SEN	6	AG ELECTRONICA DC2541WH, 998PI, 4959SN	Sensor magnético, de presencia o de apertura de puertas y ventanas.	\$ 25.00 \$ 300.00 \$ 250.00
VCC	1	MITZU MCA-34 RD	Fuente de alimentación con salida de 6 V. y potencia de 9 W.	\$ 90.00
-	-	-	Otros elementos (puentes, acido, impreso del circuito, etc.)	\$ 200.00

COSTO TOTAL	\$ 3,406.00
--------------------	--------------------

Tabla G. 1.- Lista de materiales empleados en el Sistema de Seguridad.

Los circuitos impresos de las dos placas para la manipulación local y remota, se muestran a continuación, los puentes presentes en cada placa están especificados con una línea punteada color rojo.

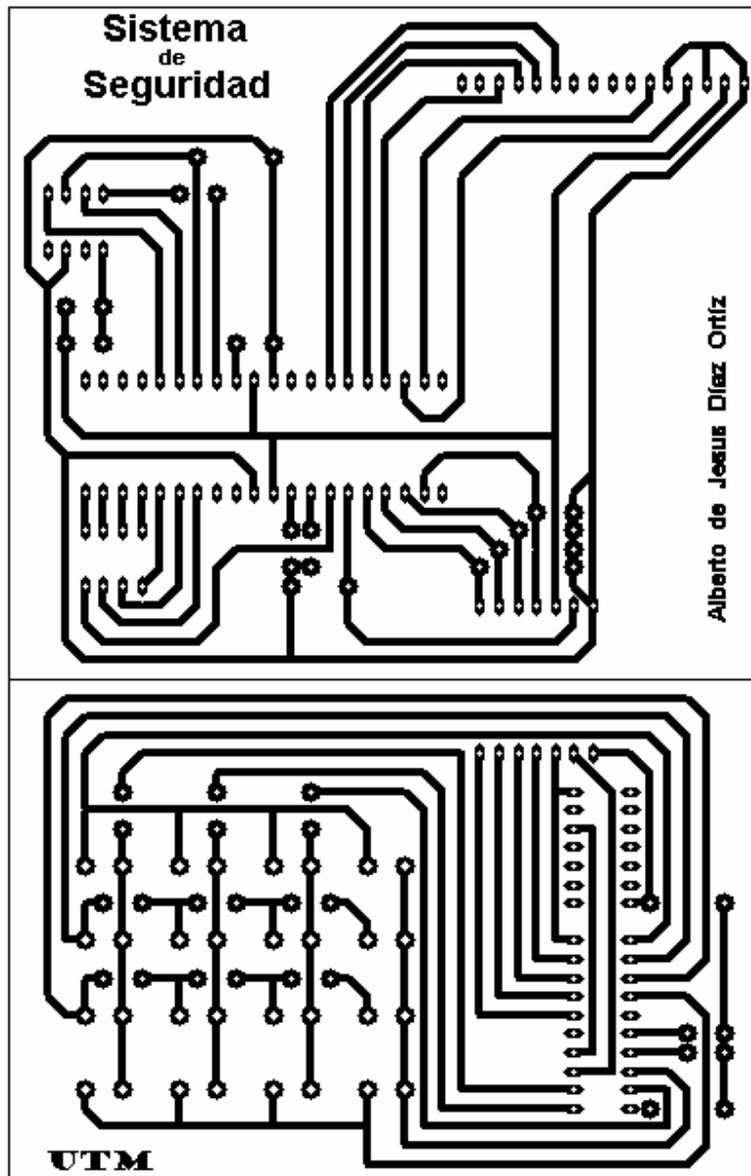


Figura G. 1.- Impreso de la placa para la manipulación local.

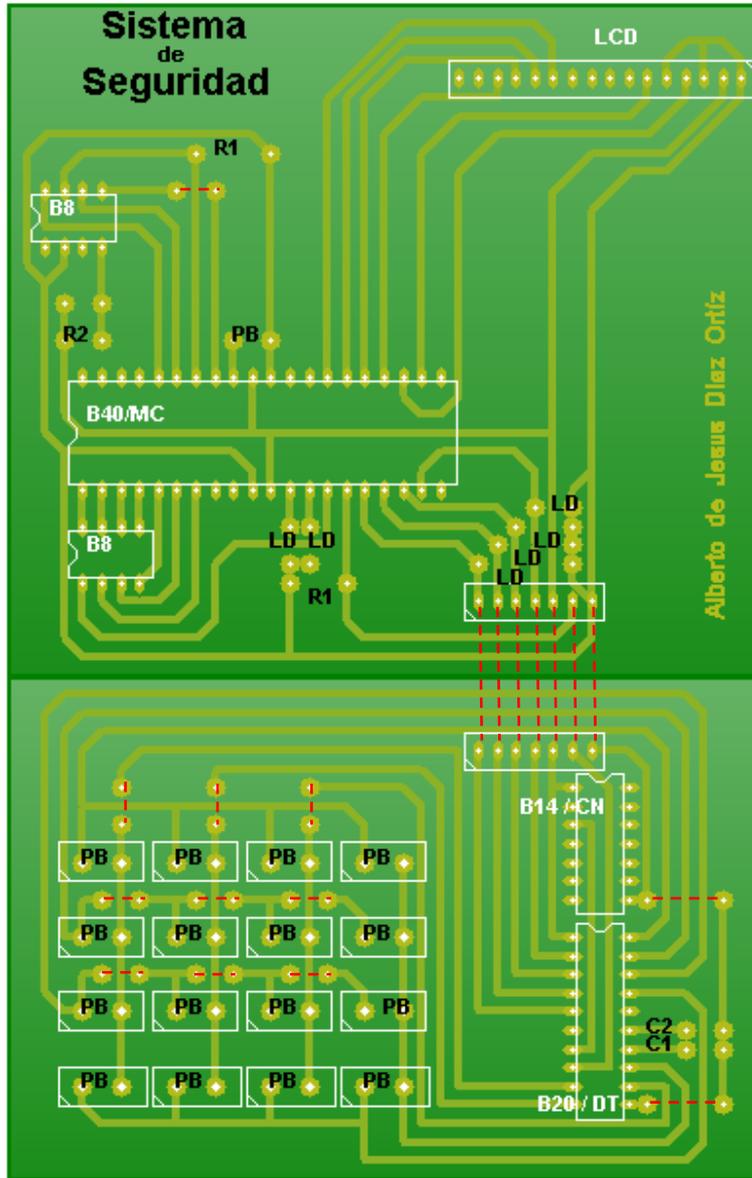


Figura G. 2.- Impreso de la placa para la manipulación local con cada uno de sus elementos.

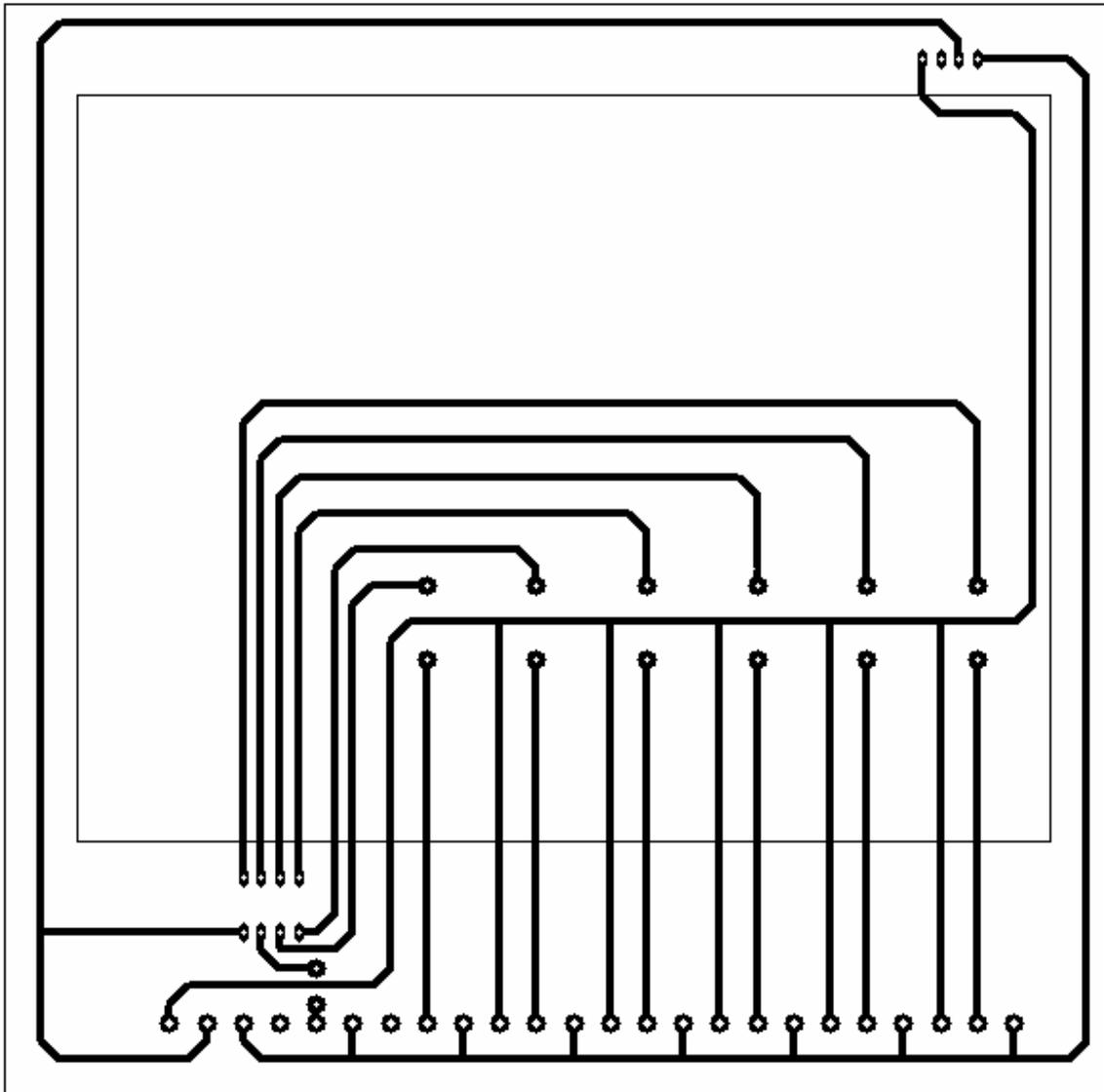


Figura G. 3.- Impreso de la placa para la manipulación remota.

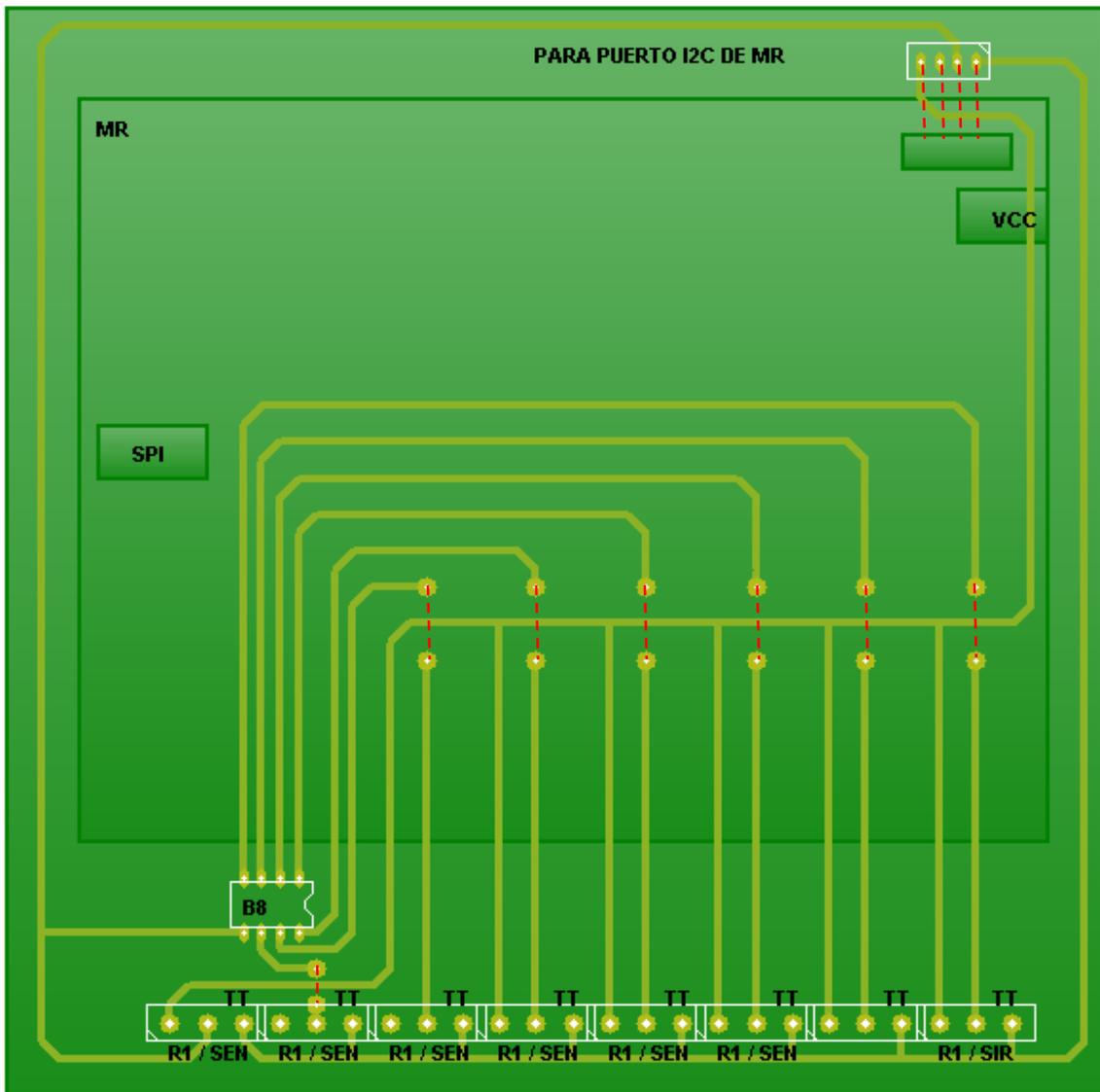


Figura G. 4.- Impreso de la placa para la manipulación remota con cada uno de sus elementos.