

ESCUELA POLITECNICA NACIONAL

ESCUELA DE FORMACION DE TECNOLOGOS

CONSTRUCCIÓN DE UN SISTEMA DE CONTROL DE ILUMINACIÓN Y SISTEMA DE SEGURIDAD DE PUERTAS Y VENTANAS DE UNA CASA PROTOTIPO MEDIANTE MENSAJES DE TEXTO SMS UTILIZANDO EL MICROCONTROLADOR 16F877.

PROYECTO PREVIO A LA OBTENCION DEL TITULO DE TECNOLOGO EN ELECTRONICA Y TELECOMUNICACIONES

VICTOR GABRIEL GONZALEZ MORENO

`gaboman85@hotmail.com`

DIRECTOR: ING.PATRICIO CARRASCO MEDINA

`patricio.carrasco@epn.edu.ec`

Quito, agosto 2012

DECLARACIÓN

Yo, GONZÁLEZ MORENO VÍCTOR GABRIEL, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

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

**GONZALEZ MORENO
VICTOR GABRIEL**

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por GONZALEZ MORENO VICTOR GABRIEL, bajo mi supervisión.

ING. PATRICIO CARRASCO

DIRECTOR DE PROYECTO

DEDICATORIA

Dedico este trabajo a mi madre, gracias por todo, y al Ing. Patricio Carrasco, por sus sabios conocimientos.

CONTENIDO

Resumen	i
Antecedentes	iii
CAPITULO I	1
1. FUNDAMENTACIÓN TEÓRICA.	1
1.1 Servicio de mensajes cortos de texto (SMS)	1
1.1.1 Introducción.	1
1.1.2 Evolución de la telefonía celular.	2
1.1.3 Arquitectura de la red GSM	4
1.1.4 Tipos de SMS	7
1.1.5 Beneficios	8
1.2 Protocolos de comunicación en los módems GSM	9
1.2.1 Los comandos AT	9
1.2.2 Códigos de control ascci.	12
1.2.3 Comunicación en modo texto y en modo PDU	12
1.2.4 Comunicación serie	13
1.2.5 Protocolo MBUS	16
1.2.6 Protocolo FBUS	16
1.3 El microcontrolador 16F877	16
1.3.1 Arquitectura	16
1.3.2 Descripción	17
1.3.3 Características	18
1.3.4 Ventajas	21
1.3.5 Desarrollo del software	22
1.3.5.1 El lenguaje microcode	23

1.3.5.2 Características principales de microcode	23
1.3.5.3 La pantalla de microcode	23
1.3.5.4 El software programador	25
1.4 Otros dispositivos electrónicos.	28
1.4.1 El circuito integrado máx. 232.	28
1.4.2 El LCD.	30
1.4.3 El Teclado.	32
1.4.4 Cables y conectores seriales.	33
1.5 Descripción de cada bloque y funcionamiento.	34
1.5.1 Modulo de transmisión y recepción.	34
1.5.2 Circuito de control	36
1.5.3 Fuente de poder.	43
1.5.4 Circuito de energía de respaldo	44
Capitulo II	45
2. Construcción de la interfaz de control	45
2.1 Desarrollo del hardware	45
2.1.1 Desarrollo del diagrama circuital en proteus.	46
2.1.2 Realización de la placan impresa.	48
2.1.3 Ensamblaje de la interfaz de control.	49
2.2 Programación del software de control	51
2.2.1 El lenguaje Visual Basic 6.0	51
2.2.2 Las pantallas de Visual Basic	52
2.2.3 Comunicación serie desde Visual Basic.	55
2.2.4 Crear proyectos ejecutables desde Visual Basic.	57
2.5 Ensamblaje de todo el proyecto	58

2.5.1 Instalación de luces y sensores en la maqueta.	58
2.5.2 Conexión entre la maqueta y la interfaz de control	59
2.6 Pruebas y calibración.	59
2.6.1 Tiempo de respuesta ante una alarma.	60
2.6.2 Análisis de resultados.	61
Conclusiones	62
Recomendaciones.	63
ANEXO A	v
TABLA DE CODIGOS DE CONTROL ASCCI	
ANEXO B	viii
PROGRAMAS DE LOS MICROCONTROLADORES	
ANEXO C	xIv
DATASHEET DEL 4N25	
ANEXO D	xIvii
DIAGRAMAS CIRUITALES DE LAS PLACAS DEL SISTEMA	
ANEXO E	I
MANUAL DE USUARIO.	
Bibliografía y Web grafía.	Ixi

RESUMEN

En este documento se explica la teoría aplicada para la construcción del equipo, el cual consiste en un sistema que permite controlar las luces de una casa prototipo y monitorear la apertura de la puerta o ventanas, utilizando el servicio de mensajería corta, (SMS siglas en Inglés de Short Message Service).

El prototipo utiliza un sistema basado en un microcontrolador conectado a un teléfono celular. Si un evento ocurre en la casa prototipo mientras el sistema se encuentra en modo alerta, un mensaje será enviado a través de la red GSM a los celulares de dos usuarios para informarles de las novedades; este circuito también permite extraer los caracteres de un nuevo SMS recibido y de acuerdo a esta información activa uno de los 3 relés que encienden o apagan las luces de la casa prototipo.

Además, se puede modificar los números de celular de los usuarios que controlan el sistema, así como la clave para poner al sistema en modo alerta; utilizando un programa para enviar los nuevos datos a través de una conexión serie entre el computador y el equipo.

El desarrollo de esta tesis se ha dividido en dos capítulos para su mejor comprensión.

En el Capítulo 1, se presenta la introducción y evolución de la telefonía celular, junto con la arquitectura para el servicio de mensajería corta (SMS); seguido a esto se muestran los protocolos de comunicación que utilizan los teléfonos celulares con dispositivos exteriores, después en este mismo capítulo están los fundamentos teóricos de los dispositivos electrónicos que van a ser utilizados en el desarrollo de este proyecto; y finalmente la descripción de cada una de las etapas de funcionamiento del equipo.

El Capítulo 2, contiene el diseño del prototipo y su ensamblaje. Se presenta la programación del software de control y su compresión en un solo archivo ejecutable, además se realiza la construcción de la casa prototipo y las respectivas conexiones entre la maqueta y el equipo. Con el prototipo terminado se presentan las pruebas y calibración, por ultimo este capitulo finaliza con las conclusiones y recomendaciones del proyecto.

A continuación del capítulo 2, se incluyen los anexos; los cuales contienen las tablas de los códigos de control ASCCI, se presenta además el código fuente de los microcontroladores, los diagramas que componen el circuito, las hojas de datos de los dispositivos electrónicos utilizados, y finalmente se presenta el manual de usuario de la aplicación.

ANTECEDENTES

La inseguridad en la ciudad es una realidad preocupante en la actualidad, por el aumento de la delincuencia y la falta de cooperación entre los ciudadanos.

Por esta razón este proyecto está dirigido al área de seguridad, con la finalidad de desarrollar un sistema que pueda ser controlado remotamente.

En este proyecto se aplicó la tecnología para crear un sistema de seguridad y control de luces; con el cual se puede mantener los domicilios protegidos de la delincuencia sin importar la distancia a la que se encuentren sus habitantes.

La cualidad de este proyecto es aprovechar el teléfono celular que en estos días está al alcance económico de casi todas las personas, para que puedan recibir una señal de alerta si ocurre algún intento de robo en sus domicilios. A diferencia de otros sistemas de seguridad, este equipo siempre contacta a los usuarios, ya que su alcance es cualquier lugar donde exista la cobertura GSM, de esa manera las personas que reciben el mensaje de alerta en sus celulares, pueden realizar el procedimiento necesario como llamar a la policía o a los vecinos, para detener a los delincuentes si se ha producido una infiltración no autorizada al domicilio.

Además, gracias a la tecnología GSM es posible el control de cualquier dispositivo eléctrico dentro del hogar. En este proyecto se eligió la activación y desactivación de las luces en el domicilio, esta acción simula que alguien se encuentra presente, y disminuye los intentos de robo a los hogares.

En este proyecto se aplican conocimientos tecnológicos en electrónica.

CAPITULO 1

1 FUNDAMENTACIÓN TEÓRICA.

1.1SERVICIO DE MENSAJES CORTOS DE TEXTO (SMS)

El uso de Celulares desde hace algunos años dejo de ser objeto de lujo o status social, en nuestro país este dispositivo se ha convertido en un instrumento indispensable para la comunicación .Se hace evidente que cada vez es mas necesaria la comunicación, y por consiguiente la integración de tecnología y diversas aplicaciones que van haciendo del teléfono celular el computador del futuro.

Dentro de los servicios conocidos están los mensajes de texto (SMS), los cuales son una forma de comunicación que se ha extendido muy ampliamente, llegando ha convertirse en una herramienta comercial muy potente y de amplia gama de servicios.

Es por esto que hay que aprovechar esta tecnología disponible y crear aplicaciones que satisfagan necesidades cotidianas.

1.1.1 INTRODUCCIÓN¹.

El servicio de mensajería corta consiste en el intercambio de mensajes de texto que pueden incluir letras, números, y otros caracteres a través de la red GSM. SMS apareció en escena en 1991 en Europa, donde la tecnología inalámbrica digital echo raíces.

El servicio de transmisión de datos (SMS) permite transferir un mensaje de texto entre una estación móvil (MS) y otra entidad (SME), a través de un centro de servicio (SC).El servicio final ofrecido es una comunicación extremo a extremo entre la estación móvil (MS) y la entidad (SME). La entidad puede ser otra estación móvil o puede estar situado en una red fija. En la figura 1.1 se aprecia lo mencionado.

¹ Flores Christian,Rivadeneira José Luis , Tesis Escuela Politécnica Nacional, Diseño e Implementación de un Sistema Microprocesado Para Adquisición de Datos en Forma Remota de un Medidor Digital de Consumo de Energía Electrica Tipo Industrial Mediante Telefonía Celular,pag 24,2006.

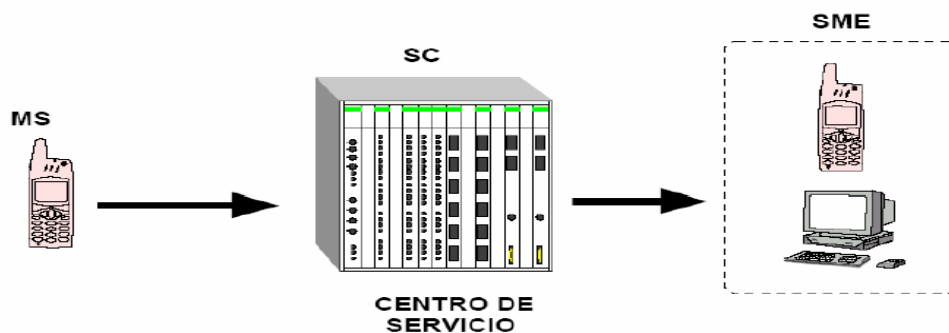


FIG 1.1: servicio sms¹

SMS también garantiza la entrega de los mensajes cortos por la red. Errores temporales son identificados y el mensaje es guardado en la red hasta que el destino este disponible.

1.1.2 EVOLUCIÓN DE LA TELEFONÍA CELULAR².

La telefonía móvil ha evolucionado a partir de las nuevas necesidades del usuario, su evolución se la puede explicar en cuatro generaciones que se describen a continuación.

Primera generación.

Aparece alrededor de 1979. Se caracterizó por utilizar multiplexación por división en frecuencia. Fue un sistema analógico, donde se prestaba servicio de voz. La tecnología más conocida de primera generación es AMPS (Advanced Mobile Phone Service). AMPS empezó a ser utilizado en Estados Unidos desde 1981. Estaba basado en tecnología FDMA (Frecuency Division Multiple Access). Sólo podía tener a un usuario por canal de frecuencia. Tuvo problemas de capacidad y de seguridad.

Segunda generación.

Aparece en el año de 1990. La generación 2G constituye un cambio de telefonía móvil analógica a digital.

²Estrada Juan, Reinos Diego, Tesis Escuela Politécnica Nacional, Diseño E Implementación De Un Prototipo De Monitoreo Remoto De Contaminación Ambiental Utilizando Tecnología GSM, ,2009,pag 2.

Con estos cambios se solucionaron problemas de calidad y seguridad de la comunicación, mejorando la capacidad.

Se permite la integración de nuevos servicios como SMS (Short Message Service), fax y otros.

Las tecnologías más comercializadas en esta generación fueron: GSM(Global System for Mobile Communications), CDMA (Code Division Multiple Access), DAMPS(Digital Advanced Mobile Phone System) entre otros.

CDMA ofrece ventajas como un incremento de alrededor 8 veces la capacidad de sistemas analógicos. Posee mejoras en calidad, cobertura y seguridad.

CDMA utiliza una técnica de espectro ensanchado (spread spectrum), permitiendo una mayor privacidad en la comunicación; por todos estos atributos esta tecnología es la mas costosa.

En 1992 aparece la tecnología GSM como un estándar abierto europeo. La principal funcionalidad que ofrece es la transmisión de voz, aunque también se pueden transmitir datos a bajas velocidades. Características como llamadas prepago, roaming Internacional, entre otras, hicieron que esta tecnología se convierta en una de las más populares. Presenta nuevos servicios como voice mail, llamada en espera y otros, a costos más bajos.

Tercera generación.

La tercera generación fue desarrollada para ofrecer al usuario una mayor velocidad y conexión multimedia, soportar altos niveles de tráfico desde 144 kbps hasta mayores a 2Mbps.

Permite servicios para transferencia de voz y datos, como también aplicaciones de descarga de programas, correo electrónico, mensajería instantánea entre otros.

Entre las ventajas de 3G está una mayor velocidad de acceso, con la ayuda de IP se logra la implementación de servicios multimedia y nuevas aplicaciones de banda ancha, como video-telefonía y video-conferencia.

Cuarta generación.

Los sistemas 4G se enfocan a servicios de video de alta calidad, con tasas de transferencia de datos de alrededor de 100 Mbps en una estación móvil, y 1 Gbps en una estación fija. 4G reensambla y converge algunas tecnologías existentes (GPRS, EDGE, WLAN, HIPERLAN).

LTE (Long Term Evolution) es una tecnología en estado de prueba, esta podría ser considerada como el estándar 4G. LTE alcanza tasas entre 16 y 154 Mbps.

1.1.3 ARQUITECTURA DE LA RED GSM³

Todas las redes GSM se pueden dividir en cuatro partes fundamentales y bien diferenciadas:

La Estación Móvil o Mobile Station (MS): Consta a su vez de dos elementos básicos que debemos conocer; por un lado el terminal o equipo móvil, y por otro lado el SIM o Subscriber Identity Module. Con respecto a los terminales poco tenemos que decir ya que los hay para todos los gustos, lo que si tenemos que comentar es que la diferencia entre unos y otros radica fundamentalmente en la potencia que tienen, esta va desde los 20 watos (generalmente instalados en vehículos), hasta los 2 watos de nuestros terminales.

El SIM es una pequeña tarjeta inteligente que sirve para identificar las características de nuestro terminal. Esta tarjeta se inserta en el interior del celular y permite al usuario acceder a todos los servicios que tenga disponibles por su operador, sin la tarjeta SIM el terminal no nos sirve de nada por que no podemos hacer uso de la red.

El SIM esta protegido por un número de cuatro dígitos que recibe el nombre de PIN o Personal Identification Number.

La mayor ventaja de las tarjetas SIM es que proporcionan movilidad al usuario ya que puede cambiar de terminal y llevarse consigo el SIM. Una vez que se introduce el PIN en el terminal, éste va a ponerse a buscar redes GSM que estén disponibles, y va a tratar de validarse en ellas. Una vez que la red

³ Estrada Juan, Reinos Diego, Tesis Escuela Politécnica Nacional, Diseño E Implementación De Un Prototipo De Monitoreo Remoto De Contaminación Ambiental Utilizando Tecnología GSM, pag 6, 2009

(generalmente la que tenemos contratada) ha validado nuestro terminal, el teléfono queda registrado en la célula que lo ha validado.

La Estación Base o Base Station Subsystem (BSS): Sirve para conectar a las estaciones móviles con los NSS que veremos a continuación, además de ser los encargados de la transmisión y recepción. Como los (MS), los (BSS) también constan de dos elementos diferenciados: La Base Transceiver Station (BTS) y la Base Station Controller (BSC). La BTS consta de transceivers y antenas usadas en cada célula de la red, y suelen estar situadas en el centro de la célula, generalmente su potencia de transmisión determinan el tamaño de la célula.

Los BSC se utilizan como controladores de los BTS, y tienen como funciones principales las de estar al cargo de los handovers, los frequency hopping, y los controles de las frecuencias de radio de los BTS.

El Subsistema de Conmutación y Red o Network and Switching Subsystem (NSS): Este sistema se encarga de administrar las comunicaciones que se realizan entre los diferentes usuarios de la red; para poder hacer este trabajo la NSS se divide en siete sistemas diferentes, cada uno con una misión dentro de la red.

- Mobile Services Switching Center (MSC): Es el componente central del NSS, y se encarga de realizar las labores de conmutación dentro de la red, así como de proporcionar conexión con otras redes.

- Gateway Mobile Services Switching Center (GMSC): Un gateway es un dispositivo traductor (puede ser software o hardware), el cual se encarga de interconectar dos redes, haciendo que los protocolos de comunicaciones que existen en ambas redes se entiendan. La misión del GMSC es servir de mediador entre las redes de telefonía fijas y la red GSM

- Home Location Register (HLR): El HLR es una base de datos que contiene información sobre los usuarios conectados a un determinado MSC. Entre la información que almacena el HLR tenemos fundamentalmente la localización del usuario y los servicios a los que tiene acceso. El HLR funciona en unión con el VLR que vemos a continuación.

- Visitor Location Register (VLR): Contiene toda la información necesaria para que un usuario acceda a los servicios de red. Forma parte del HLR con quien comparte funcionalidad.
- Authentication Center (AuC): Proporciona los parámetros necesarios para la autenticación de usuarios dentro de la red; también se encarga de soportar funciones de encriptación.
- Equipment Identity Register (EIR): Se utiliza para proporcionar seguridad en las redes GSM pero a nivel de equipos válidos. La EIR contiene una base de datos con todos los terminales que son válidos para ser usados en la red. Esta base de datos contiene los (International Mobile Equipment Identify) o IMEI de cada terminal, de manera que si un determinado móvil trata de hacer uso de la red y su IMEI no se encuentra localizado en la base de datos del EIR, no puede hacer uso de la red.
- GSM Interworking Unit (GIWU): sirve como interfaz de comunicación entre diferentes redes para comunicación de datos.

Los Subsistemas de soporte y Operación u Operation and Support Subsystem (OSS): Los OSS se conectan a diferentes NSS y BSC, para controlar y monitorizar toda la red GSM.

Un esquema de red GSM se puede apreciar en la figura 1.2.

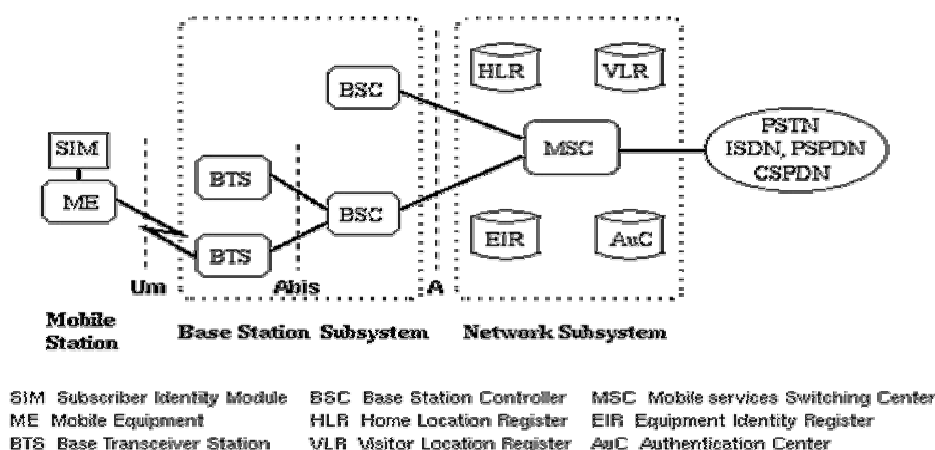


FIG 1.2: arquitectura de la red gsm³

1.1.4 TIPOS DE SMS⁴.

Los SMS pueden clasificarse según el número de destinatarios en: Mensajes punto a punto y punto multipunto.

Punto a punto

En este tipo de mensajes el destinatario es único, y se pueden clasificar según la dirección de envío en: Mobile Originated y Mobile Terminated.

Mobile Originated

Son los mensajes que se originan en el SME. El mensaje es transportado desde el SME hasta el SMSC (*SMS-SUBMIT*), el destino puede ser otro usuario móvil o una aplicación. Se puede configurar el SME para que el SMSC envíe un reporte de confirmación del mensaje recibido (*SMS-SUBMIT-REPORT*). El mensaje puede ser enviado a un número corto (Ej. 5039), que previamente ha sido contratado a las operadoras móviles por parte de las empresas que prestan servicios utilizando SMS.

Este tipo de mensajes son los que se emplean para participación en concursos, votaciones, petición de alertas o de recepción de información en el móvil.

En la figura 1.3 se muestra un esquema de este tipo de SMS.



FIG 1.3: mo (mobile originated)⁴

Mobile Terminated

Son los mensajes que se envían a un SME. El mensaje es enviado desde el SMSC hasta el terminal móvil (*SMS-DELIVER*), la fuente puede ser otro usuario móvil o una aplicación. Una vez que el mensaje llega al terminal móvil un reporte confirma al SMSC que la entrega fue completada (*SMS-DELIVER-REPORT*). En la figura 1.4 se puede observar un esquema de este tipo de SMS.

⁴Estrada Juan, Reinos Diego, Tesis Escuela Politécnica Nacional, Diseño E Implementación De Un Prototipo De Monitoreo Remoto De Contaminación Ambiental Utilizando Tecnología GSM, pag 19,2009

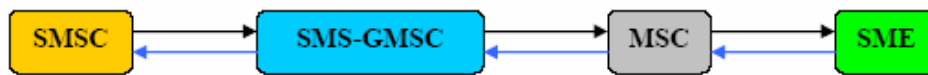


FIGURA 1.4: mt (mobile terminated)⁴

Punto multipunto

En este tipo, el mensaje es enviado a un conjunto de usuarios. A este tipo corresponde Cell broadcast. Permite el envío simultáneo de mensajes a múltiples usuarios en un área geográfica específica. El destino del mensaje está descrito en términos de identificadores de celda utilizados por la BSC para enrutar el contenido del mensaje a los usuarios de la BTS.

1.1.5 BENEFICIOS

Los beneficios que traen los SMS son:

Servicios de notificación. Permite el envío de mensajes a un grupo de usuarios, los cuales constan en una base de datos específica, por ejemplo: clientes de compañías de televisión, clubs deportivos, supermercados y otros minoristas, aerolíneas y bancos. Estos mensajes pueden ser publicitarios, de notificación, entre otros.

Servicios de localización.

Aplicado a la localización de vehículos, integra un sistema de posicionamiento global (GPS). Los datos de longitud y latitud son transferidos a un terminal móvil. El terminal por medio de un SMS envía estos datos a un servidor, en donde se procesan para indicar la localización actual del vehículo en un mapa geográfico.

Supervisión remota.

El servicio de mensajería corta puede usarse para gestionar máquinas en ambientes de supervisión remota. Esta aplicación proporciona valiosa información sobre el estado o el suceso de algún evento, ocurrido sobre la máquina que el usuario precisa saber.

Comercio electrónico.

Se pueden llevar a cabo transacciones financieras a través del terminal móvil, para la cual será necesario tener convenios con algunas instituciones bancarias.

1.2 PROTOCOLOS DE COMUNICACIÓN EN LOS MÓDEMS GSM

1.2.1 LOS COMANDOS AT.

Los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un terminal módem. La figura 1.5 muestra un esquema de la comunicación entre un celular y un computador, utilizando comandos AT.



FIGURA 1.5: Esquema de comunicación entre un computador y un celular⁵

Los comandos AT se denominan así por la abreviatura de attention. Aunque la finalidad principal de los comandos AT es la comunicación con módems, la telefonía móvil GSM también ha adoptado como estándar este lenguaje para poder comunicarse con sus terminales.

De esta forma, todos los teléfonos móviles GSM poseen un juego de comandos AT específico, que sirve de interfaz para configurar y proporcionar instrucciones a los terminales. Este juego de instrucciones puede encontrarse en la documentación técnica de los terminales GSM, y permite acciones tales como realizar llamadas de datos o de voz, leer y escribir en la agenda de contactos, y enviar mensajes SMS; además de muchas otras opciones de configuración del Terminal⁵.

La notación para la ejecución de cualquier comando AT es la siguiente:

AT + Comando <CR>= DATO

⁵ Figura y texto tomados de: www.zonabot.com

El prefijo AT debe ser adicionado al empezar cada línea de comando, luego viene el tipo de comando que se va a utilizar, y un retorno de carro al final (CR = enter).

Cuando la sintaxis del comando es incorrecta retorna la palabra ERROR.

Si la sintaxis del comando es correcta pero con algún parámetro incorrecto retorna el texto (+CMS ERROR).

Cuando la línea de comando ha sido desarrollada exitosamente retorna la palabra OK.

Comandos de Configuración.

Mediante este tipo de comandos se puede cambiar la configuración interna del Modem del teléfono celular; enviar un mensaje de texto, leer el estado de la batería, etc.

Comando de Atención.

Este comando de Atención, tiene la función de garantizar una buena conexión en el canal de comunicación, cuando existe una buena conexión el teléfono responde OK..La tabla 1.1 indica el comando y su respuesta.

Comando	Respuesta del teléfono celular	Especificaciones del teléfono
AT	OK	Comando de atención

TABLA 1.1: Respuesta al comando atención⁶

Comandos para leer mensajes.

Cuando un nuevo SMS llega al celular este se almacena en la bandeja de entrada, y ocupa parte de la memoria del teléfono o de la tarjeta SIM. En la tabla 1.2 se aprecia el conjunto de comandos para leer un SMS , y sus respectivas descripciones.

⁶ Figura tomada de : www.zonabot.com

Comando AT	Descripción de la instrucción
AT+CMGL="ALL"	Mediante este comando se puede leer todos los mensajes almacenados en la memoria del teléfono celular.
AT+CMGL="REC UNREAD"	Se accede a los mensajes recibidos pero no leídos.
AT+CMGL="REC READ"	Se accede a los mensajes recibidos y leídos.
AT+CMGL="UNSEND"	Son los mensajes escritos y almacenados pero no enviados.
AT+CMGL="SENT"	Mensajes enviados.
AT+CMGR=1,2,3	Mediante este comando se lee los mensajes de acuerdo a la posición en la que llegaron al teléfono celular.

TABLA 1.2: Tabla de comandos para lectura de un sms⁶

Comandos para Enviar mensajes.

Para el envío de mensajes, el teléfono celular debe estar configurado en modo texto, esto lo veremos mas adelante. Los comandos para enviar un SMS se aprecian en la tabla 1.3.

Comando AT	Respuesta del teléfono celular	Descripción de la instrucción
AT+CMGS= "# "	> Aquí se escribe el SMS	Se envía un mensaje teniendo en cuenta que se debe tener saldo.
AT+CMSS=1,2,..	> Aquí se escribe el SMS	Enviar mensaje almacenado en la dirección que se encuentre, de igual manera se debe tener saldo.

TABLA 1.3: Tabla de comandos para enviar un sms⁶

Para utilizar los comandos de la tabla 1.3 de forma correcta, se debe seguir el siguiente orden:

- AT+CMGS [ENTER<CR>]
- ESCRIBIR MENSAJE
- CTRL+Z

Comando para lectura del estado de la batería.

Es posible obtener un dato que muestre la cantidad de carga que tiene la batería del celular en un momento determinado. La tabla 1.4 indica la información que despliega este comando.

Comando AT	Respuesta del teléfono celular	Descripción de la instrucción
AT+CBC	> 0,100	El dato 100 muestra la carga total de la batería este dato puede variar entre 75 , 50 y 25 dependiendo de la carga.

TABLA 1.4: Comando de lectura del estado de batería del celular.

Mediante los comandos AT el teléfono celular puede comunicarse con la PC u otros dispositivos, por ejemplo un microcontrolador, esta ventaja es aprovechada para realizar este proyecto.

1.2.2 CÓDIGOS DE CONTROL ASCII⁷.

El código ASCII reserva los primeros 32 códigos (numerados del 0 al 31 en decimal) para caracteres de control (códigos no pensados originalmente para representar información imprimible). Estos se utilizaban para controlar dispositivos como impresoras que usaban ASCII. Por ejemplo, el carácter 10 representa la función "nueva línea", el cual hace a una impresora avanzar el papel, y el carácter 27 representa la tecla "escape", que a menudo se encuentra en la esquina superior izquierda de los teclados comunes. Para el desarrollo de este proyecto se utilizarán principalmente dos códigos ASCII: (CARRIAGE RETURN=ENTER) y (SUBSTITUTE=CTRL+Z). En el anexo A se encuentra las tablas con los códigos de control ASCII.

1.2.3 COMUNICACIÓN EN MODO TEXTO Y EN MODO PDU.

Existe dos formas para la comunicación con un celular utilizando comandos AT.

-modo texto

-modo PDU (Protocol Description Unit)

En el modo texto los datos serán interpretados como una secuencia de caracteres decimales, que se envía al teléfono al momento de intercambiar datos. El modo PDU procesa la información como una cadena de caracteres en octetos hexadecimales, o semioctetos decimales.

⁷ Texto tomado de www.wikipedia.com/codigosascii.html

Para seleccionar el tipo de comunicación con el que se va a trabajar utilizamos el comando `AT+CMGF=?`. Este comando permite configurar el modo de interpretación de los datos por parte del teléfono, si la equivalencia de la “?” es igual a “1”, los datos serán interpretados en modo texto. Si la equivalencia de “?” es igual a “0”, los datos son interpretados en modo PDU. La tabla 1.5 muestra la forma de selección de estas comunicaciones.

Comando AT	Respuesta del teléfono celular	Descripción de la instrucción
AT+CMGF=0	OK	Modo =0 indica el formato de mensajes en modo PDU.
AT+CMGF=1	OK	Modo =1 indica el formato de mensajes en modo TEXTO.

TABLA 1.5: Tabla de comandos para elegir el tipo de comunicación con el celular.

Para el desarrollo de este proyecto utilizamos la opción “AT+CMGF=1”, debido a que la comunicación entre el microcontrolador y el teléfono celular se realiza en modo de texto.

1.2.4 COMUNICACIÓN SERIE⁸.

Es un protocolo de comunicación entre dispositivos, y toma este nombre debido a que los bits se reciben uno detrás de otro o en serie.

El nivel lógico "1", representa un estado de tensión o corriente denominado marca, el nivel lógico "0", representa un estado de tensión o corriente denominado espacio. En la figura 1.6 se aprecia un ejemplo de esta forma de comunicación.

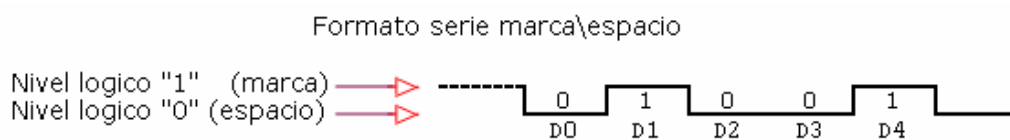


FIG 1.6: Formato de la comunicación serie⁸

Existen varios estándares que usan el formato marca/espacio, el más conocido es el estándar RS232.

⁸ Texto e imagen tomados de :www.edudevices.com.ar

La norma RS232.

La norma RS232 es una de las más populares que se utilizan en la comunicación serie, y es la que se utiliza en los PC's; este tipo de comunicación se establece entre el **DTE** (Equipo Terminal de Datos), por ejemplo un PC y el **DCE** (Equipo para la comunicación de datos), por ejemplo un ratón:

Características eléctricas de la señal: Se establece que la longitud máxima entre el DTE y el DCE no debe ser superior a los 15 metros, y la velocidad máxima de transmisión es de 20.000 bps. Los niveles lógicos no son compatibles con los TTL. En la tabla 1.6 se aprecia las diferencias entre los niveles de voltaje TTL y el formato RS232.

FORMATO	Nivel Lógico "1" (Marca)	Nivel Lógico "0" (Espacio)
TTL	5V	0
RS 232	-3V a -15V	+3V a +15V

TABLA 1.6: Tabla de los niveles de voltaje para cada formato.

Características mecánicas de los conectores: Se utiliza un conector de 9 patillas, DB 9. La figura 1.7 muestra a los conectores DB9 y su distribución de pines

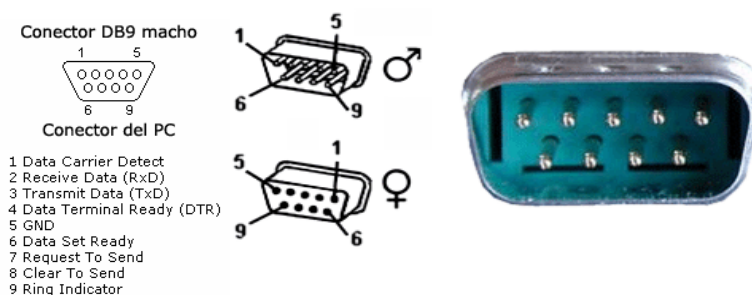


FIG 1.7: Conectores seriales y su descripción de pines⁸

Velocidad: La velocidad está estandarizada según la norma RS 232 en baudios:

- a. 300
- b. 600
- c. 1200
- d. 2400
- e. 4800
- f. 9600
- g. 19200

RS232 en el PC.

El puerto serie de un computador trabaja en modo asincrónico. El puerto serie recibe y envía información fuera del computador mediante un determinado software de comunicación. La información se envía al puerto carácter por carácter.

Por tanto en un PC se utilizan conectores DB9 macho, de 9 patillas, por los que se conectan los dispositivos al puerto serie. Los conectores hembra que se enchufan tienen una colocación de patillas diferente, de manera que se conectan la patilla 1 del macho con la patilla 1 del hembra, la patilla 2 con el 2, etc. Normalmente, las comunicaciones serie en el PC tienen los siguientes parámetros: 9.600 baudios, 1 bit de arranque, 8 bits de Datos, 1 bit de parada y sin paridad. En la figura 1.8 se puede ver un ejemplo de la transmisión en TTL del dato binario 01011001. La línea en reposo está a nivel lógico alto (+5 voltios).

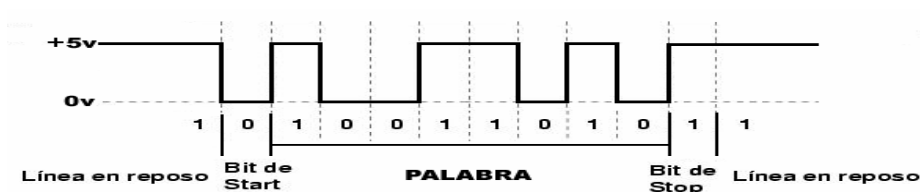


FIG 1.8: Transmisión serie con niveles TTL.

En la figura 1.9 se puede ver un ejemplo de la transmisión en RS232 del dato binario 01011001. La línea en reposo está a nivel lógico alto (-15 voltios).

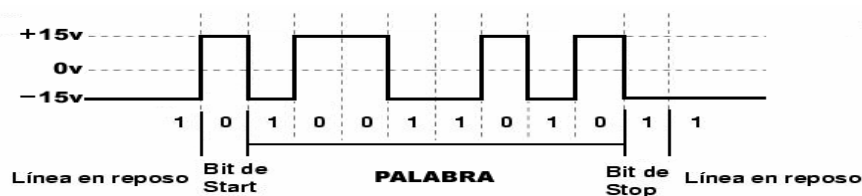


FIG 1.9: Transmisión serie con niveles RS232

La comunicación serial es un protocolo estándar de comunicaciones incluso presente en los teléfonos celulares, es por eso que en este proyecto se utiliza este tipo de comunicación, que permite comandar un dispositivo celular utilizando un microcontrolador y los comandos AT.

Además de la comunicación serial, existen otros protocolos para el intercambio de información con un teléfono celular, por ejemplo:

1.2.5 PROTOCOLO MBUS.

MBUS utiliza un solo pin para la transmisión y recepción. La transmisión es *Halfduplex*. Son utilizados dos pines *DATA* y *GND*. La comunicación con el teléfono se da a 9600 bps, 8 bits de datos, paridad impar, y un bit de parada.

1.2.6 PROTOCOLO FBUS.

Utiliza un pin de transmisión de datos, un pin de recepción, y un pin de tierra. Se parece a un puerto de transmisión serial estándar. Trabaja a 115200 bps, 8 bits de datos, sin paridad, y un bit de parada.

Tanto la comunicación MBUS como FBUS, se dan mediante un formato exclusivo de los teléfonos marca Nokia.

1.3 EL MICROCONTROLADOR 16F877⁹.

1.3.1 ARQUITECTURA.

Inicialmente todos los microcontroladores adoptaron la arquitectura clásica de Von Neumann, pero actualmente se impone la arquitectura Harvard.

⁹ www.microchip.com.

La arquitectura conocida como Harvard, consiste en un esquema en el que la CPU esta conectada a dos memorias, por intermedio de dos buses separados. Una de las memorias contiene solamente las instrucciones del programa, y es llamada memoria de programa; la otra memoria solo almacena los datos, y es llamada memoria de datos. Ambos buses son totalmente independientes y pueden ser de distintos anchos. La figura 1.10 muestra un esquema de esta arquitectura.

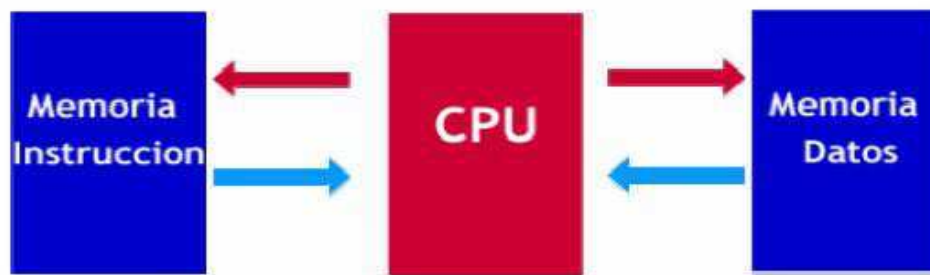


FIG 1.10: Arquitectura Harvard⁹

Las ventajas de esta arquitectura son:

- a. El tamaño de las instrucciones no esta relacionado con el de los datos, por lo que permite que cada instrucción solo ocupe una posición de memoria de programa.
- b. La velocidad de acceso se aumenta, al poseer acceso de instrucciones separado del acceso a los datos.

1.3.2 DESCRIPCIÓN.

El PIC 16F877 es un circuito integrado, en cuyo interior posee toda la arquitectura de un computador, CPU, memoria RAM, memoria EPROM y circuitos de entrada y salida. La figura 1.11 muestra un esquema de dispositivos externos conectados al microcontrolador.



FIG 1.11: Conexión de dispositivos externos al PIC⁹

El microcontrolador 16F877 de Microchip, pertenece a una gran familia de microcontroladores de 8 bits (bus de datos), este integrado tiene las siguientes características generales que los distinguen de otras familias:

- Arquitectura Harvard
- Tecnología RISC (Computador de juego de instrucciones reducido).
- Tecnología CMOS

Estos atributos se juntan para lograr un dispositivo altamente eficiente en el uso de la memoria de datos y programa; y por lo tanto en la velocidad de ejecución. En la figura 1.12 se aprecia al microcontrolador 16F877.



FIG 1.12: PIC 16F877⁹

Se utilizó este microcontrolador ya que posee las características necesarias para la implementación de este proyecto.

1.3.3 CARACTERÍSTICAS

El número de pines que posee este PIC son 40. La figura 1.13 tiene su descripción de pines.

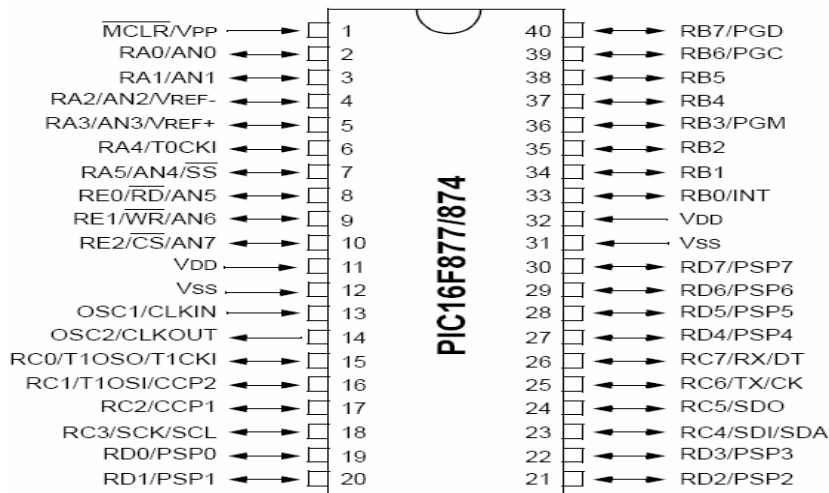


FIG 1.13: Distribución de pines del PIC 16F877⁹

CPU:

- Tecnología RISC
- Sólo 35 instrucciones que aprender
- Todas las instrucciones se ejecutan en un ciclo de reloj, excepto los saltos que requieren dos
- Frecuencia de operación de 0 a 20 MHz (200 nseg de ciclo de instrucción)
- Opciones de selección del oscilador

Memoria:

- Hasta 8k x 14 bits de memoria Flash de programa.
- Hasta 368 bytes de memoria de datos (RAM).
- Hasta 256 bytes de memoria de datos EEPROM.
- Lectura/escritura de la CPU a la memoria flash de programa.
- Protección programable de código.

Reset e interrupciones:

- Hasta 14 fuentes de interrupción
- Reset de encendido (POR)
- Timer de encendido (PWRT)
- Timer de arranque del oscilador (OST)
- Sistema de vigilancia Watchdog timer.

Otros:

- Modo SLEEP de bajo consumo de energía.
- Rango de voltaje de operación de 2.0 a 5.5 volts
- Alta disipación de corriente de la fuente: 25mA.
- Rangos de temperatura: Comercial, Industrial y Extendido.
- Bajo consumo de potencia.

Organización de la memoria del PIC 16F877

Existen tres bloques de memoria dentro de un PIC 16F877. La memoria de programa y la memoria de datos tienen buses separados, por lo que es posible el acceso a cada una de ellas en forma concurrente. El tercer bloque es la memoria de datos EEPROM.

RP1:RP0	Bank
00	0
01	1
10	2
11	3

TABLA 1.7: Bits de selección del banco de memoria

Organización de la memoria de programa.

Los dispositivos PIC16F877 tienen 8K x 14 palabras de FLASH program memory. El vector de RESET (por donde comienza a ejecutar el PIC) está en la dirección 0000h, y el vector de interrupciones se encuentra en la dirección 0004h.

Organización de la memoria de datos.

La memoria de datos está particionada en múltiples bancos que contienen los registros de propósito general, y los registros de funciones especiales. El PIC16F877 posee una RAM de datos de 368 x 8, dividido en 4 bancos de 128 bytes cada uno. La tabla 1.7 indica el equivalente en binario de cada banco de memoria.

Banco de registros y memoria de datos.

Los registros son de 8 bits y están formados por cuatro bancos.

En las primeras posiciones de dichos bancos se encuentran los registros de

El reloj.

Para que el PIC pueda procesar las instrucciones, es necesario un reloj, cuya frecuencia es parámetro fundamental en el momento de establecer la velocidad de ejecución de las instrucciones, y en el consumo de energía.

El tiempo en que tarda en ejecutarse una instrucción se llama ciclo de instrucción, en los PIC's un ciclo de instrucción emplea cuatro periodos de reloj. Todas las instrucciones del PIC se realizan en un ciclo de instrucción, salvo las de salto que tardan el doble.

1.3.4 VENTAJAS

-El PIC 16F877 de 40 pines usa tecnología flash (EEPROM) que permite un borrado rápido y reprogramación para acelerar la depuración de programas.

-Disponen de memoria de datos no volátil que puede ser usada para archivar variables y otros parámetros.

-Admite interrupciones, posee comparadores de magnitudes analógicas, convertidores A/D, puertos serie y diversos temporizadores.

-Sus distintos modelos, contienen todos los recursos que se precisan en las aplicaciones de los microcontroladores de 8 bits.

-El TMR1 que hay en esta gama, tiene un circuito oscilador que puede trabajar asincrónicamente, y que puede incrementarse aunque el microcontrolador se halle en el modo de reposo (sleep), posibilitando la implementación de un reloj en tiempo real.

1.3.5 DESARROLLO DEL SOFTWARE

Para hacer funcionar un PIC hay que primero programarlo, pero no en lenguaje común, es decir que para que el PIC realice un proceso determinado es necesario aprender su lenguaje. Existen diferentes lenguajes:

Lenguaje de alto nivel.- Permite que los algoritmos se expresen en un nivel y estilo de escritura fácilmente legible y comprensible por el hombre. En la actualidad se trata de lenguajes de tipo visual.

Lenguaje de bajo nivel.- El usuario se acerca un poco mas al lenguaje de maquina. Permiten un acceso más amplio al control físico de la maquina (hardware).

Lenguaje ensamblador.- Se considera al lenguaje de más alto nivel. El usuario escribe código en el mismo idioma del procesador.

Se tiene control total del sistema. Es necesario un conocimiento de la arquitectura mecánica del procesador para realizar una programación efectiva. Este lenguaje genera un código en ensamblador para luego dárselo al PIC, pero el PIC tampoco sabe leer éstas instrucciones, por lo tanto se necesita un compilador, que transforma lo escrito, en un archivo “.hex”.

1.3.5.1 El lenguaje microcode¹⁰

Microcode Studio es una interfase utilizada para la programación de microcontroladores utilizando el lenguaje Basic. Cuenta con un entorno de gran alcance visual de desarrollo integrado (IDE). En este programa se puede escribir el código del microcontrolador, y al mismo tiempo nos muestra una corrección de errores de sintaxis, otro de los beneficios es que ordena las subrutinas. En Microcode al finalizar el programa, se debe compilar y tendremos generado el archivo .Hex, los programas deben ser guardados en formato .Bas.Los errores de compilación pueden ser fácilmente identificados y corregidos, mediante la ventana de error de los resultados; simplemente con un clic en un error de compilación y Microcode Studio automáticamente le llevará a la línea de error.

1.3.5.2 Características principales de microcode.

- Coloración total del código fuente.
- Puede identificar y saltar rápidamente a las sentencias de include, symbols, defines, variables y etiquetas usando la ventana de explorador de códigos.
- Identifica y permite corregir los errores de compilación de código.
- Soporta compilación de código para una gran cantidad de microcontroladores.

1.3.5.3 La pantalla de microcode.

La pantalla principal de este programa se asemeja a la aplicación de Windows en la que se tiene dos barras.

La primera es la barra de herramientas ubicada en la parte superior, contiene opciones para guardar archivos, copiar, borrar, elegir el tipo de PIC que se va a utilizar y el puerto de comunicación por el que se descarga el programa.

Luego se tiene la barra de estado localizada en la parte inferior, muestra el número de líneas que contiene el programa y los errores que existen cuando se realiza la compilación. La figura 1.15 muestra la ventana principal del programa MICROCODE.

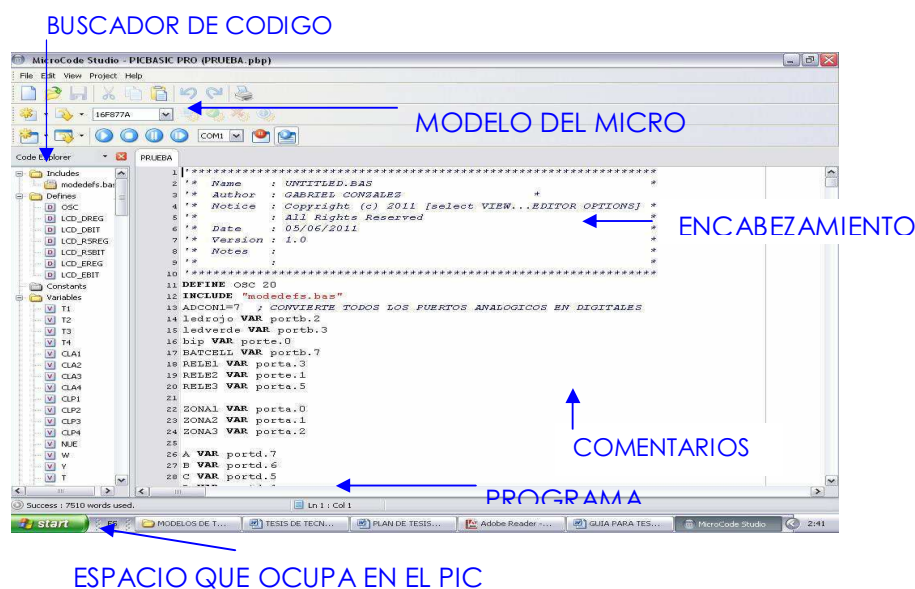


FIG 1.15: Ventana principal de Microcode Studio

1. Modelo del MicroPIC.- Esto es lo primero que se debe escoger antes de empezar a programar, seleccionar de acuerdo al modelo del PIC que se va a programar.

2. Buscador de código.- Aquí se van adicionando las variables que creamos, sirve para saber que componentes incluyen en el programa, y también

¹⁰ Texto tomado de : www.frino.com

como buscador de líneas, para esto basta con dar un clic en el nombre de la línea que se desea encontrar y automáticamente indica.

3. Espacio que ocupa en el PIC. - Aparece una vez que se compila el programa, se debe fijar si alcanza en el PIC que se dispone o debemos reemplazar por otro de mayor capacidad.

4. Programa del microcontrolador. - En esta parte escribimos nuestro programa.

5. Encabezado del programa.- Se utiliza para comentarios en los que se puede incluir nombre, fecha, autor y una explicación en breves palabras de como y para que sirve el programa.

6. Comentarios.- Es recomendable usar comentarios todo el tiempo, dentro de un lapso determinado no recordara para que servia tal instrucción, en ese momento los comentarios son de mucha ayuda.

Para realizar un programa en MICROCODE primero se crea el archivo fuente, cuyo nombre termina con la extensión “.bas”, luego se escribe el programa utilizando el editor de texto con la cantidad de líneas necesarias que cumplan con las funciones requeridas.

Al finalizar la escritura del programa, éste debe ser guardado utilizando el icono de la barra de herramientas de MICROCODE para obtener el archivo “.bas”, Para compilar se utiliza la opción en la barra de herramientas, el compilador despliega un mensaje de inicialización y procesa el archivo, al aceptar esta acción se crea un archivo de código fuente ensamblado con extensión “.asm” , y automáticamente se invoca al ensamblador para completar la tarea.

Si todo se encuentra bien, se crea un archivo de código con extensión “.hex “, caso contrario se emite un listado de los errores en la barra de estado, éstos deben ser corregidos en el archivo fuente antes de ser compilados nuevamente.

En el anexo B se encuentra los códigos fuente de los PICs 16F877 y 16F628A, realizado en microcode Studio.

1.3.5.4 El software programador.

Para transferir el código de un computador al PIC normalmente se usa un dispositivo llamado programador.

Para este proyecto se utilizara el programador Pickit 2, el cual es un equipo compuesto por hardware y software que trabajan en conjunto, y soportan una amplia gama de microcontroladores.

El grabador Pick-2 se conecta directamente con uno de los puertos USB disponibles en el computador, mediante el correspondiente cable de interfase incluido con el equipo. Inmediatamente es reconocido por Windows como un dispositivo de interfase de usuario (HID), que se puede ver ingresando a Panel de Control >Administrador de Dispositivos. En la figura 1.16 se aprecia el grabador de PICs USB.



FIG 1.16: Grabador PICKIT-USB¹¹

La ejecución del software se puede realizar desde el icono de acceso directo ubicado en el escritorio, éste se crea después de la instalación.

El software programador muestra una ventana de estado, la cual informa que el Pick-USB ha sido localizado y conectado "PICKit 2 found and connected". Si además hubiera algún PIC insertado en el zócalo, éste será detectado "PIC Device Found. La figura 1.17 muestra la ventana principal del software para grabar el PIC.

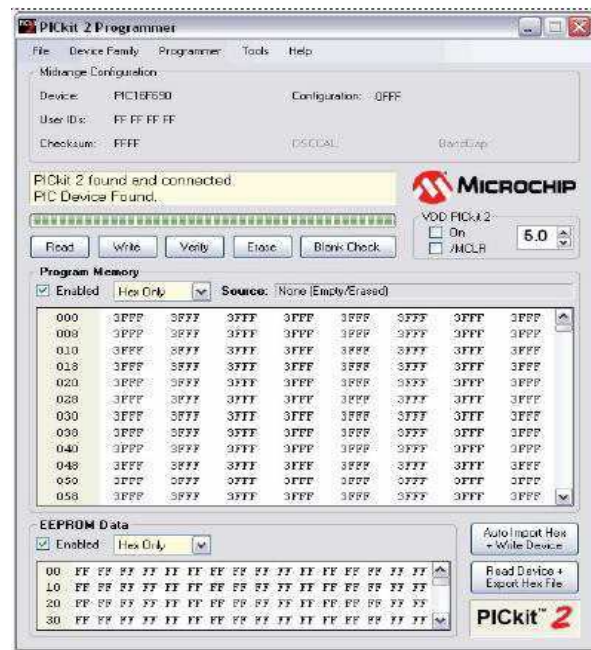


FIG 1.17: Ventana principal del software PICKit 2

Para Importar un fichero HEX debemos dar un clic en File Import Hex, y se abre la clásica caja de dialogo que nos permitirá localizar el fichero HEX deseado. Una vez seleccionado, cargamos su contenido sobre los buffers de memoria del software PICKit-2, para su posterior grabación sobre la memoria física del controlador.

Después que un archivo fue abierto con el software, en las ventanas correspondientes se aprecia que tanto el buffer de la memoria de programa como el de la EEPROM de datos, se llenan con el contenido del fichero HEX recién leído o importado. El contenido de dichos buffers se puede visualizar tanto en hexadecimal como en ASCII y también se puede modificar manualmente. En la figura 1.18 se aprecia el programa que se grabará en hexadecimal, y en la figura 1.19, el programa que se almacenará en la memoria EEPROM del PIC.

¹¹ Texto tomado de : www.frino.com

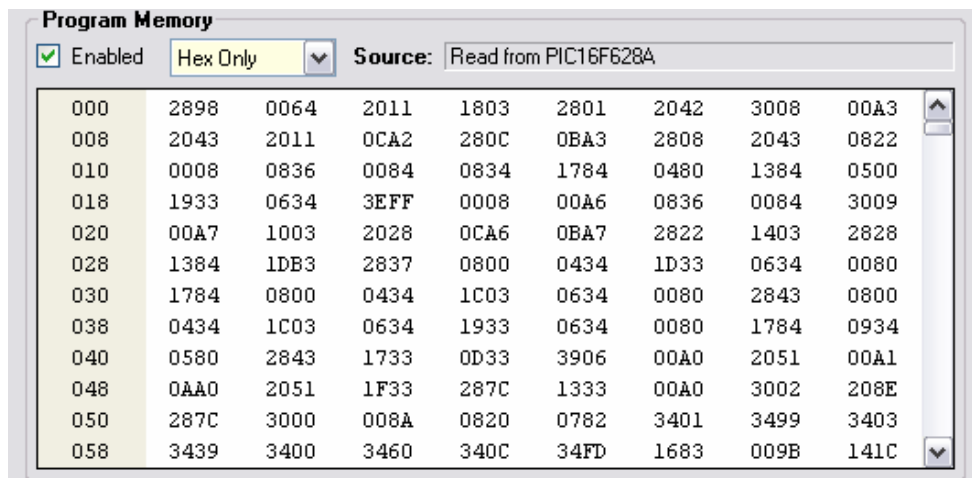


FIG 1.18: Datos almacenados en un PIC en hexadecimal.

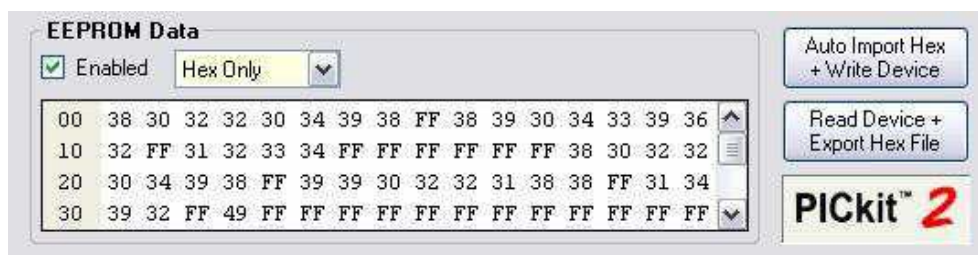


FIG 1.19: Datos almacenados en la memoria no volátil de un PIC

También tenemos acceso a la memoria de configuración haciendo clic sobre "Configuration". Aquí se muestra el valor actual, en binario, de la palabra o palabras de configuración del PIC que se encuentra insertado en el grabador. La figura 1.20 permite apreciar la ventana de los bits de configuración del PIC.

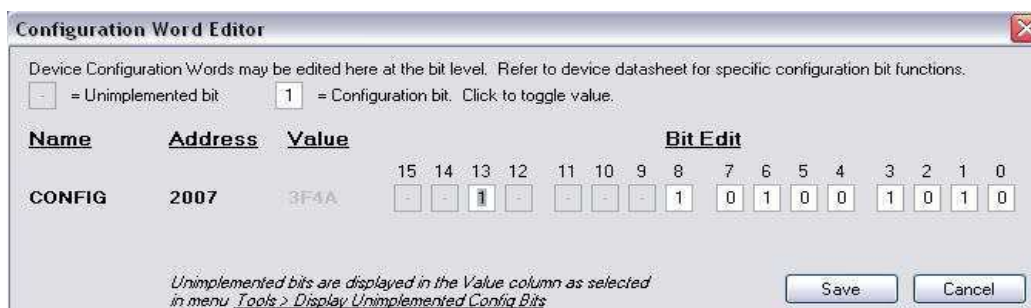


FIG 1.20: Ventana de configuración de los bits de función del PIC.

Una vez insertado el PIC en el zócalo , seleccionado la familia y modelo del mismo, e importado el fichero HEX, podemos proceder a grabarlo. Las tareas básicas están disponibles en los botones de la figura 1.21.



FIG 1.21: Botones para las tareas principales en el PICKIT-2

1.4 OTROS DISPOSITIVOS ELECTRÓNICOS.

Para que el sistema de seguridad y control de luces pueda mostrar y recibir comandos, necesitamos los siguientes dispositivos electrónicos que sirvan como interfase de usuario.

1.4.1 EL CIRCUITO INTEGRADO MÁX. 232¹².

Este integrado es usado para comunicar un microcontrolador o sistema digital, con un PC o sistema basado en el estándar RS232, esto lo hace adaptando los niveles de voltaje que maneja el computador, los cuales van desde (-15V a +15V), a voltajes TTL (0V a 5V) con los que trabaja el microcontrolador. La figura 1.22 indica la conexión eléctrica entre un PIC, el max 232, y un computador.

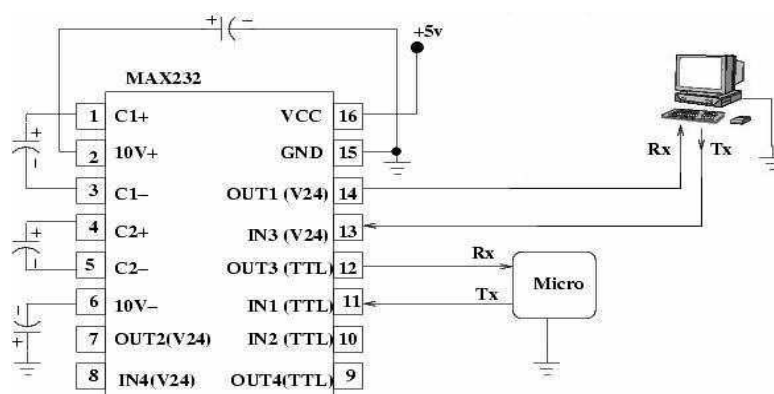


FIG 1.22: Esquema de conexiones entre un pc y un PIC utilizando el MAX 232¹².

¹²Texto y figura tomados de: circuitos electronicos.com.

Este circuito integrado lleva internamente 2 convertidores de nivel de TTL a RS232, y otros 2 de RS232 a TTL, con lo que en total podremos manejar 4 señales del puerto serie del PC.

En caso de que la distancia de comunicación sea corta (2 metros o menor) no es necesario utilizar el max 232, pero es necesario limitar la corriente que viene del PC, ya que el microcontrolador soporta una corriente de 25 mA máximo en cualquiera de sus pines. El calculo de la corriente que ingresa al PIC es el siguiente

$$I = \frac{10 \text{ V}}{22000 \Omega} = 0.5 \text{ mA}$$

Con 0.5 mA en la entrada del PIC no existe riesgo de que este se dañe.

Para la transmisión desde el PIC hacia el PC se coloca una resistencia de 220 Ω , para regular la corriente de entrada al puerto serie. El calculo de esta corriente se presenta a continuación.

$$I = \frac{5 \text{ V}}{220 \Omega} = 22 \text{ mA}$$

Una corriente de 22mA, regulada con la resistencia de 220 Ω , es el mismo valor de amperaje que se obtiene utilizando el MAX232.

En la figura 1.23 se aprecia la conexión eléctrica entre el microcontrolador, y el conector DB9 que tiene incorporado la placa del circuito.

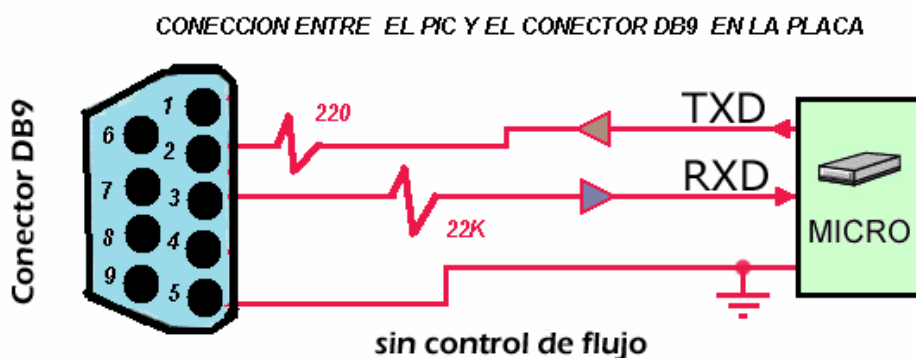


FIG 1.23: Conexión entre el PIC y el puerto serie sin utilizar el MAX 232.

1.4.2 EL LCD¹³.

Es un dispositivo visualizador, utilizado para mostrar mensajes que indican al operario el estado de la maquina, o para dar instrucciones de manejo. El LCD permite la comunicación entre las maquinas y los humanos, este puede mostrar cualquier carácter ASCII, y consume mucho menos que los displays de 7 segmentos, el mas popular es el 2X16, 2 líneas de 16 caracteres cada una. En la a figura 1.24 se aprecia la LCD.

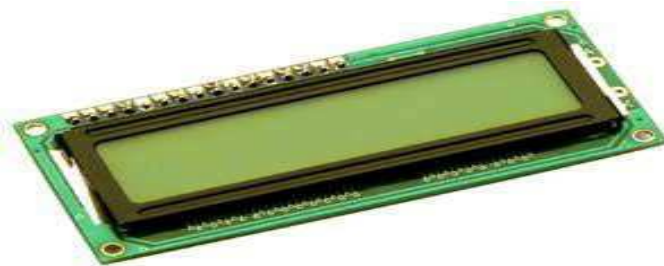


FIG 1.24: Pantalla LCD 2X16¹³.

Está constituido por un circuito impreso en el que están integrados los controladores del display, y los pines para la su conexión.

Sobre el circuito impreso se encuentra el LCD en sí, rodeado por una estructura metálica que lo protege. En total se pueden visualizar 2 líneas de 16 caracteres cada una, es decir, $2 \times 16 = 32$ caracteres.

La tensión nominal de alimentación es de 5V, con un consumo menor de 5mA.

El LCD consta de 16 pines para su conexión con un microcontrolador, cada uno tiene una función definida. Además de la polarización este dispositivo necesita un potenciómetro de 10K para la regulación del contraste del cristal. En la figura 1.25 se muestra la conexión del potenciómetro.



FIG 1.25. Distribución de pines del LCD¹³

¹³Texto y grafico tomados de www.X-ROBOTICS.com

En la TABLA 1.8 se encuentra la descripción de cada pin del LCD.

PIN	SIMBOLO	DESCRIPCION
1	V _{ss}	Tierra de alimentación GND
2	V _{cc}	Alimentación de +5 Vcc
3	V _o	Ajuste del contraste del cristal líquido (oa +5V)
4	R _s	Selección del Registro controlados RS=0 reg. Control RS=1
5	R/W	Lectura / Escritura en LCD R/W=0 escritura R/W=1
6	E	Habilitación
7	D0	Bit menos significativo (bus de datos bidireccional)
8	D1	
9	D2	
10	D3	
11	D4	
12	D5	
13	D6	
14	D7	Bit más significativo (bus de datos bidireccional)

TABLA 1.8: Funciones de cada pin del LCD.

En la figura 1.26 aparecen las señales necesarias para el funcionamiento y control del display.

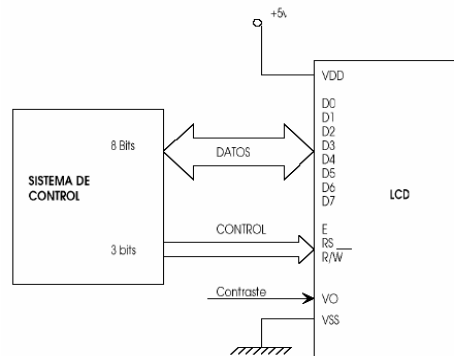


FIG 1.26: Conexión entre un PIC y un LCD.

Los datos se transmiten por un bus de datos de 8 bits de anchura, aunque el display también ofrece la posibilidad de trabajar con este bus multiplexado en dos grupos de 4 bits. Para el control del display son necesarios 3 bits: una señal de enable (E), una para indicar lectura/escritura (R/W), y otra para seleccionar

uno de los dos registros internos (RS). Por lo tanto se puede controlar el display con $8+3=11$ bits.

1.4.3 EL TECLADO¹⁴.

Es un dispositivo de entrada de datos, consta de 16 teclas o pulsadores, dispuestos e interconectados en filas y columnas. Dispone de un conector SIL (Single In Line) de 8 pines, que corresponden con las 4 filas y las cuatro columnas de las que dispone. La figura 1.27 muestra el teclado utilizado en el equipo.



FIG 1.27: Teclado matricial adhesivo.

En la figura 1.28, vemos el esquema de conexionado interno del teclado matricial, cuando se presiona un pulsador se conecta una fila con una columna, teniendo en cuenta este hecho es muy fácil averiguar que tecla fue pulsada.

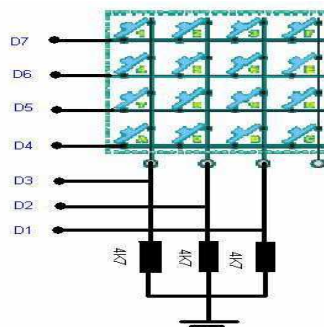


FIG 1.28: Conexión del teclado al puerto D del PIC.

¹⁴Texto y grafico tomados de www.X-ROBOTICS.com

1.4.3 CABLES Y CONECTORES SERIALES.

Para realizar la conexión entre el PC y el sistema de seguridad y control de luces, podemos utilizar un cable serie que tenga conectores db9 tipo hembra en cada extremo. En la figura 1.29 se aprecia un cable para la comunicación serie.



FIG 1.29: Cable serie¹⁵.

La conexión interna del cable debe ser directa: En la figura 1.30, aparecen las conexiones entre los pines de cada uno de los conectores DB9.

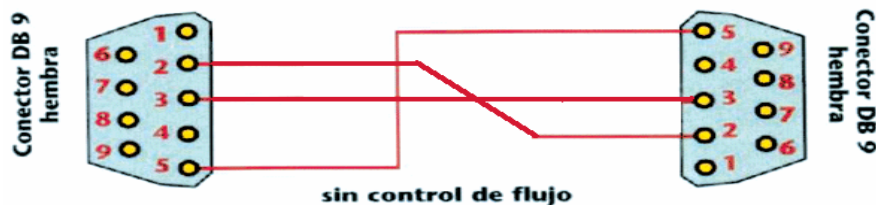


FIG 1.30: Conexiones del cable utilizado para la comunicación entre el PIC y el pc.

1.5 DESCRIPCIÓN DE CADA BLOQUE Y FUNCIONAMIENTO.

La figura 1.31 muestra las etapas que conforman el hardware del equipo, y la interacción con el usuario.

¹⁵Figura tomada de ; www.carlosvolt.com.ar

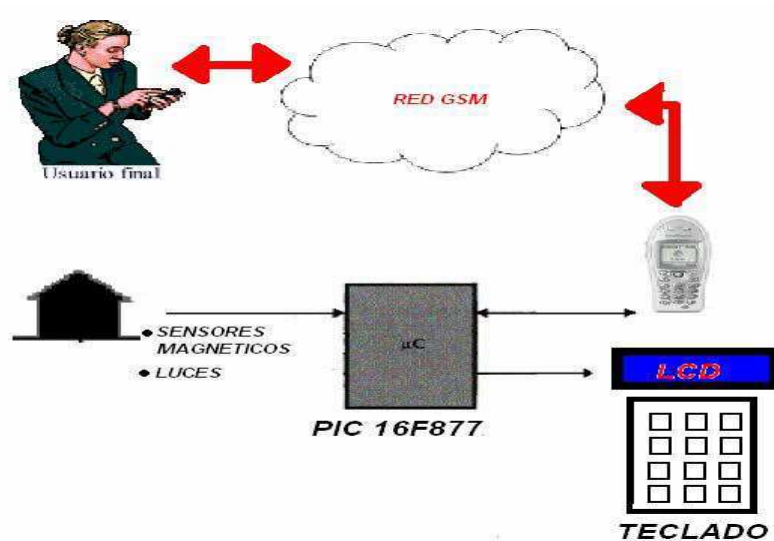


FIG 1.31: Esquema de funcionamiento del sistema de alarma y control de luces

1.5.1 MODULO DE TRANSMISIÓN Y RECEPCIÓN.

El modulo de transmisión y recepción es un teléfono celular, este dispositivo enviara la señal de alarma atreves de la red GSM, en forma de un SMS. Para su elección se debe considerar los siguientes requisitos:

- Tecnología GSM
- Capacidad para enviar y recibir mensajes
- Puerto de comunicación de datos
- Manejo de comandos AT

El modulo de transmisión y recepción que será utilizado en el equipo, es el módem incorporado en el teléfono celular de tecnología GSM modelo Motorola C115.

Dentro del mercado se pueden encontrar fácilmente terminales Motorota, y debido a que existe una mayor cantidad de soporte para aplicaciones con este tipo de terminales, se eligió a un teléfono de esta marca para el desarrollo del proyecto. En la figura 1.32 se aprecia el celular MOTOROLLA C115.



FIG 1.32: Teléfono Motorola C115¹⁶

Características del Teléfono.

Entre las características generales del teléfono se pueden citar las siguientes:

Antena: Interna

Dimensiones (Largo x Ancho x Profundidad, en mm.): 101 x 47,8 x 21,9

Peso: 80 gr.

Pantalla a color

Características de rendimiento.

Tiempo de duración de la Batería (en espera) valor aproximado: Hasta 12 días

Tiempo de duración de la Batería (conversación) valor aproximado: Hasta 8 horas.

SMS : Envío y recepción de mensajes de texto

Para la comunicación entre el celular y el PIC, se requiere de tres pines; Rx, Tx,GND, el puerto de comunicaciones del celular esta comprendido de 3 pines, transmisión, recepción, y tierra digital, conectados a un mini jack stereo, éste se puede observar en la figura 1.33.



FIG 1.33: Distribución de pines del conector (Jack) a través del cual intercambian datos el PIC y el celular¹⁶.

¹⁶Figura tomada de ; www.carlosvolt.com.ar

Para la transmisión y recepción, el teléfono maneja voltajes de 0V para el caso de 0L y 3.3 V para el caso de 1L, por lo que es necesario adaptar los voltajes TTL (0-5 voltios) del microcontrolador con un par de diodos zener de 3.3 voltios, tanto para la transmisión como para recepción. La figura 1.34 muestra la conexión entre el mini jack stereo y el PIC.

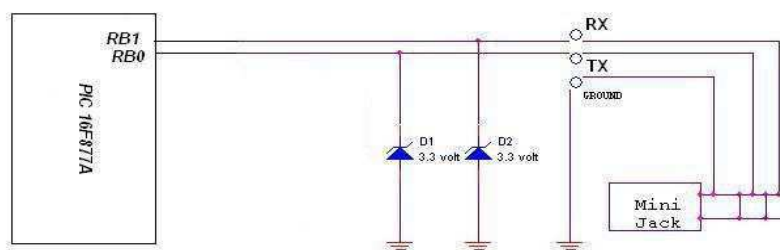


FIG 1.34: Conexión eléctrica entre el mini jack y el PIC¹⁷.

1.5.2 CIRCUITO DE CONTROL.

El circuito de control esta compuesto de algunas etapas, las cuales se muestran en la figura 1.35.

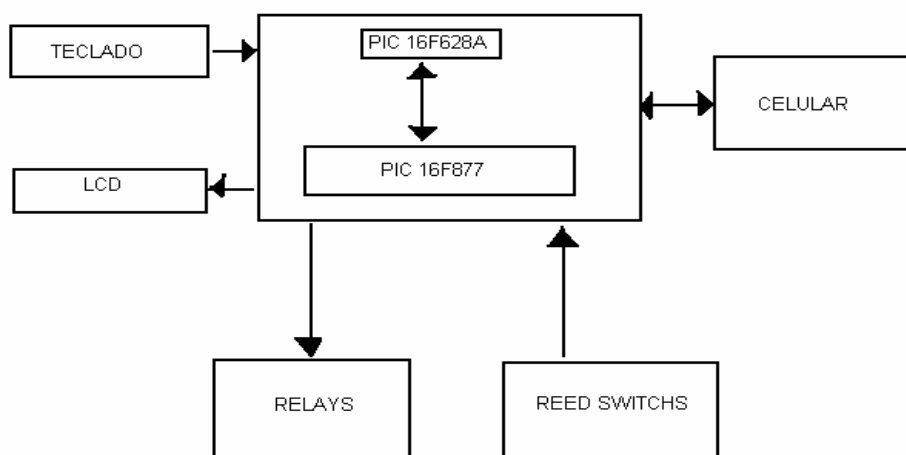


FIG 1.35: Diagrama de bloques del sistema de alarma y control de luces.

La unidad central de proceso esta compuesta por los PICs 16F877 y 16F628A, el teclado permite al usuario ingresar una clave para poner al sistema en modo alerta, el LCD muestra la información necesaria para la interacción con el usuario; el celular recibe comandos AT desde la unidad central de proceso y devuelve información para decidir si alguno de los relays es activado o

¹⁷Figura tomada de : www.carlosvolt.com.ar

desactivado. Los reed switches dan la señal para que la unidad central de proceso pueda saber si la seguridad fue violada, en este caso se ordena al celular enviar un SMS de alerta a los números de celular almacenados en la memoria del PIC 16F628A.

Para que el sistema realice estas funciones, se requiere que a cada puerto del PIC se le asigne una función específica, por lo tanto los pines que tiene el PIC16F877A serán utilizados de la siguiente manera:

- Los pines RB0 y RB1, para la comunicación con el celular.
- El pino C asignado como puerto de salida, para el control del LCD.
- El pino D asignado como puerto de entrada, para el control del teclado.
- Los pines RA0, RA1 y RA2, como entradas para la conexión de los sensores magnéticos en la casa prototipo.
- Los pines RA3, RA5 y RE1, son salidas para la activación de las luces en la casa prototipo.
- Los pines RB5 y RB4 para la comunicación con el PIC 16F628A.
- El pino RB7 para la conexión del cargador del celular, solo si el PIC ha leído un estado bajo en la carga de la batería.
- El pino RE0 se utiliza para la activación de una chicharra.
- Los pines RB3 y RB2 activan unos leds indicativos.

Cada vez que el equipo se enciende, el PIC 16F877 solicita información a un PIC más pequeño (16F628A), que es el encargado de almacenar los números de celular que pueden activar las luces en la casa prototipo, y también guarda la clave de 4 dígitos que pone en alerta al sistema de seguridad.

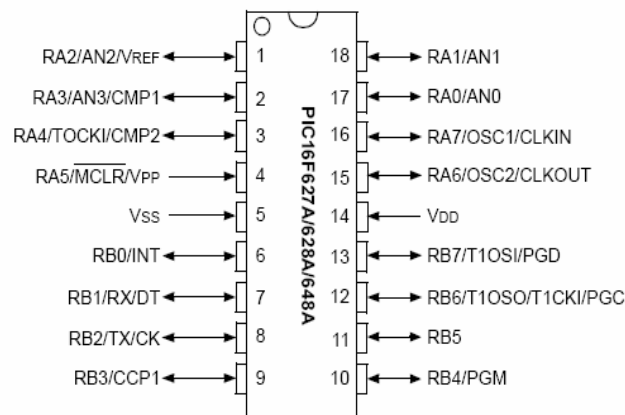
El PIC 16F628A¹⁸.

Es un microcontrolador que maneja 8 bits en el bus de datos, este pertenece a la misma familia que el PIC 16F877, por lo que sus características son casi similares. En la figura 1.36 se aprecia al PIC 16F628A.

¹⁸Texto y gráficos tomados de: www.microchip.com.

FIG 1.36: PIC 16F628A¹⁸

La figura 1.37 muestra la distribución de pines y su descripción.

FIG 1.37 Distribución de pines del PIC 16F628A¹⁸

PORTA: RA0-RA7:

- Los pines RA0-RA4 y RA6-RA7 son bidireccionales y manejan señales TTL .
- El pin RA5 es una entrada Schmitt Trigger que sirve también para entrar en el modo de programación cuando se aplica una tensión igual a V_{pp} (13,4V mínimo) .
- El terminal RA4 puede configurarse como reloj de entrada para el contador TMR0 .
- Los pines RA0-RA3 sirven de entrada para el comparador analógico.

PORTB: RB0-RB7:

- Los pines RB0-RB7 son bidireccionales y manejan señales TTL

- Por software se pueden activar las resistencias de pull-up internas, que evitan el uso de resistencias externas en caso de que los terminales se utilicen como entrada (permite, en algunos casos, reducir el número de componentes externos)
- El pin RB0 se puede utilizar como entrada de pulsos para provocar una interrupción externa
- Los pines RB4-RB7 están diseñados para detectar una interrupción por cambio de estado.
Esta interrupción puede utilizarse para controlar un teclado matricial, por poner un ejemplo

Otros pines

- VDD: Pin de alimentación positiva. De 2 a 5,5 Vcc .
- VSS: Pin de alimentación negativa. Se conecta a tierra o a 0 Vcc .
- MCLR: Master Clear (Reset). Si el nivel lógico de este terminal es bajo (0 Vcc), el microcontrolador permanece inactivo. Este Reset se controla mediante la palabra de configuración del PIC.
- OSC1/CLKIN: Entrada de oscilador externo.
- OSC2/CLKOUT: Salida del oscilador. El PIC 16F628 dependiendo de cómo se configure puede proporcionar una salida de reloj por medio de este pin.

Para este proyecto los pines de este PIC tienen las siguientes funciones:

Pin RB2 recepción de datos seriales desde el computador.

Pin RB3 transmisión de datos seriales hacia el computador.

Pin RB5 encendido de un led indicativo.

Pin RB4 conexión y desconexión de la transmisión de datos desde el celular al PIC 16F877.

El microcontrolador 16F877 es el que comanda todas las funciones que realiza el sistema de seguridad, se podría decir que este es el PIC principal, el microcontrolador 16F628A trabaja como esclavo del PIC principal, sus funciones son:

La comunicación serie con el computador, toma los datos enviados por el usuario y los almacena en la memoria EEPROM, luego de esto el PIC principal (16F877) solicita esa información y la almacena en su memoria RAM, luego realiza las comparaciones entre los datos obtenidos del celular, y los datos almacenados en la RAM del 16F877, basado en esa comparación el PIC ejecuta o no una acción.

-La segunda función es; recibir un dato (carácter ASCII) desde el PIC principal y activar un rele, para permitir el paso de los datos desde el celular hacia el PIC 16F877, luego de 2 segundos desactiva el rele y el PIC 16F877 detecta esta acción como un cambio de estado de 3.3 voltios a cero voltios, de esa manera el PIC principal puede continuar ejecutando las siguientes instrucciones.

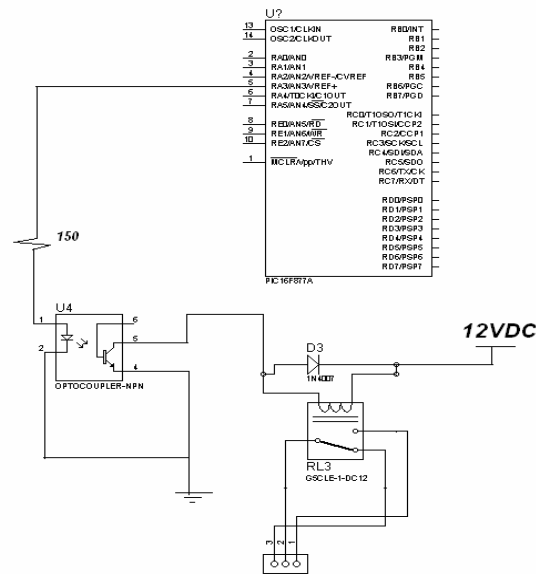
El PIC 16F628A almacena toda programación en su memoria eeprom, por lo tanto; no existe peligro que los datos se borren en caso de que el equipo se quede sin energía totalmente.

El PIC 16F877 activará cualquiera de los pines (RA3, RA5 o RE1), con el propósito de activar el circuito que controla las luces de la casa prototipo, sí el microcontrolador a leído un nuevo mensaje de texto con el código de activación correcto.

Siempre que se quiera aislar eléctricamente las entradas del PIC, se debe utilizar un opto acoplador como el 4N25 o similares, las características de este dispositivo están en el anexo C.

Este dispositivo es un circuito integrado en cuyo interior hay un diodo emisor de luz y un fototransistor, cuando se aplica una señal al diodo, éste emite luz y pone al transistor en modo de saturación, y por tanto el transistor deja pasar corriente al igual que un switch cerrado.

Las luces de la casa prototipo se encuentran aisladas mediante opto acopladores, los mismos que controlan relees de 12 voltios para el encendido o apagado de las luces. La figura 1.38 muestra el circuito de activación de las luces de la casa prototipo.



.FIG 1.38: Diagrama de conexiones entre el PIC , el opto acoplador y un relé.

De acuerdo a la curva característica del opto acoplador, se ve que para una corriente de 30mA, se tiene una caída de voltaje sobre el diodo de 1.25V, entonces:

$$R = \frac{5V - 1.25V}{30 \text{ mA}} = 125 \Omega$$

Valor estándar para proteger el diodo del 4N25= 150Ω.

Para la detección de una alarma se utiliza sensores magnéticos (REED SWITCH), que permiten el paso de 5 voltios hacia los pines (RA0, RA1, RA2), seleccionados como entradas del microcontrolador. La figura 1.39 muestra un reed switch comúnmente utilizado en alarmas.

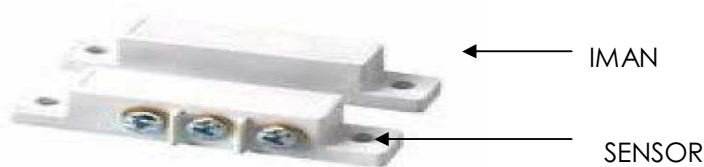


FIG 1.39: Reed switch¹⁹

¹⁹Figura tomada de : www.ico-Ecuador.com

Son dispositivos utilizados muy ampliamente en los sistemas de seguridad, internamente contienen un contacto que se abre o cierra dependiendo de cuán cerca está el imán. La figura 1.40 indica la conexión entre un reed switch y el microcontrolador.

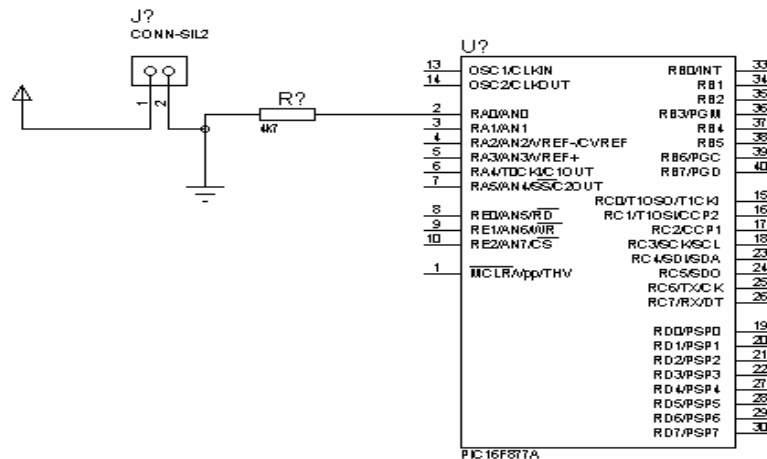


FIG 1.40: Conexión entre el PIC y el reed switch

La corriente de entrada que soporta cada pin del PIC es de 25 mA , por lo tanto:

$$R = \frac{5V}{0.025 A} = 200 \Omega \approx 220 \Omega$$

Esto quiere decir que la resistencia mínima a colocarse es de 220 Ω para estar al límite de la capacidad que soporta el PIC, pero no es aconsejable trabajar con los límites, por lo que se recomienda utilizar una resistencia de 4.7k para limitar la corriente de entrada al PIC. La corriente obtenida es la siguiente:

$$I = \frac{5V}{4700 \Omega} = 1 \text{ mA}$$

Con un miliamperio ingresando al PIC no hay problema, considerando que este soporta 25 mA como máximo en sus entradas.

El PIC 16F877 siempre está solicitando el estado de la batería al celular, si se detecta un estado igual al 50 por ciento de carga, el PIC satura un transistor y así activa un rele que conecta el cargador al celular.

De manera inversa cuando el PIC detecta un estado de carga en la batería del 100 por ciento, el PIC pone en corte al transistor y así desconecta el cargador del celular.

Se seleccionó al transistor 2N3904, ya que este soporta una corriente de colector aproximada de 200mA, suficiente para activar un rele.

De acuerdo a las especificaciones este transistor soporta una corriente pico de base de 100 mA, por lo que se necesita una resistencia para limitar el amperaje en la base. El calculo de la corriente es el siguiente.

$$I = \frac{5V}{4700 \Omega} = 1 \text{ mA}$$

Con 1 miliamperio la base del 2N3904 este dispositivo se satura normalmente. En la figura 1.41 se aprecia el circuito para la conexión del cargador del celular.

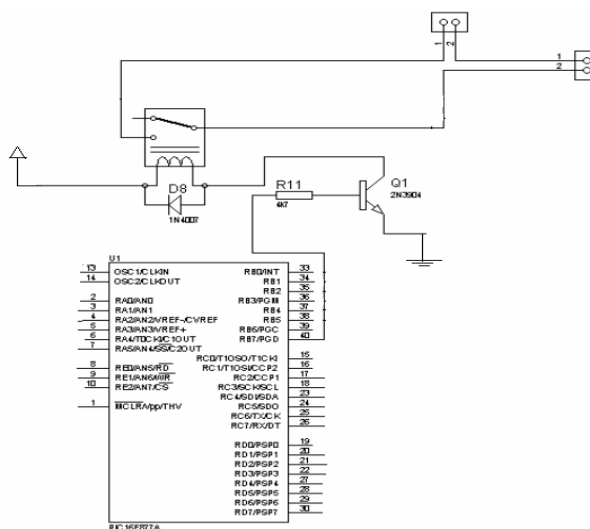


FIG 1.41: Diagrama de para la conexión del cargador del celular

1.5.3 FUENTE DE PODER.

Para este proyecto se construyó una fuente de 12 voltios DC, en una placa de circuito impreso aparte de la placa de circuito de control. Esta fuente es no regulada ya que el regulador LM7805 se encuentra en la placa principal. En la figura 1.42 aparece el diagrama circuital de la fuente de poder.

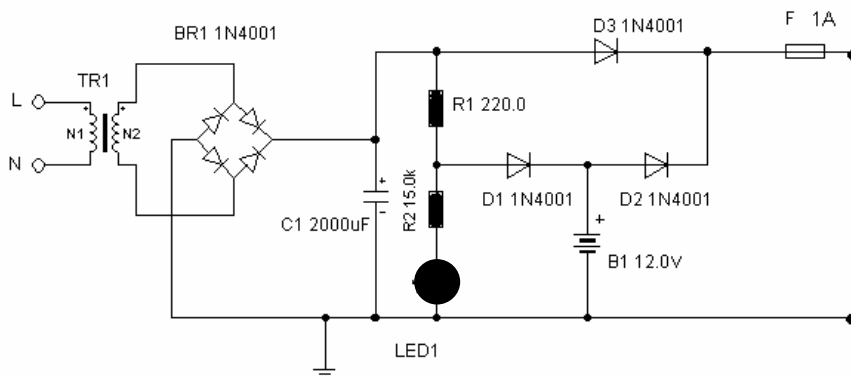


FIG 1.42: Diagrama de la fuente del sistema²⁰

El transformador TR1 tiene su primario conectado a la red eléctrica, que es 120 voltios en Ecuador.

El bobinado secundario es de 12 voltios y 1 amperio.

El puente de diodos 1n4001 rectifica la onda completamente, y luego esta es filtrada por C1.

El led 1 se encenderá solamente cuando la corriente de la red eléctrica está presente, con un apagón el led no lucirá, y el voltaje de salida se mantendrá por la batería (B1).

1.5.4 CIRCUITO DE ENERGÍA DE RESPALDO.

Al producirse el corte de la línea de red (fallo de CA), la batería B1 asume el control.

La resistencia R1 y D1 es el camino de carga para la batería B1, D1 y D3 previenen que el LED1 se ilumine bajo condiciones de fallo de la red.

El circuito carga la batería por goteo, con una corriente de carga definida como:

$$I(\text{carga}) = (VP1 - 0.6) / R1$$

Donde VP1 es el voltaje de la fuente no regulada igual a 13 voltios ,y 0.6V corresponde al diodo D1.

²⁰Diagrama tomado de : circuitoselectronicos.com.

D2 se debe incluir en el circuito, sin D2 la batería se cargaría directamente del voltaje de fuente sin el límite que le proporciona R1, lo que le causaría daño, incluso el recalentamiento en algunas baterías recargables. La figura 1.43 indica el trayecto de la corriente en casa de falla de corriente alterna.

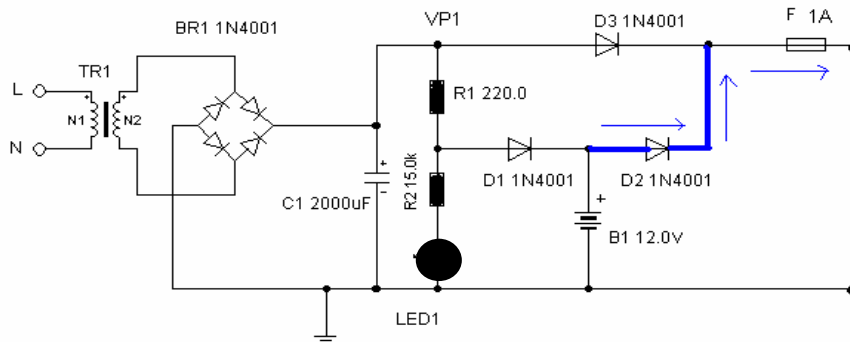


FIG 1.43: Trayecto de la corriente de la batería marcado en azul²⁰.

La capacidad de duración de la batería sin red eléctrica, depende de la cantidad de corriente que utiliza el circuito, y también de la capacidad en amperaje de la batería.

Si utilizara una batería de 12 voltios 1A/h y la carga fuera de 12 voltios 0.5 amperios, entonces la batería de respaldo se mantendría alrededor de 2 horas.

CAPITULO 2

1.-CONSTRUCCIÓN DE LA INTERFAZ DE CONTROL.

En este capítulo se detalla el ensamblaje del proyecto, además se muestra la programación del software realizado en visual Basic 6.0, que permite la comunicación entre un computador y el sistema de seguridad y control de luces, con el fin de modificar los números de celular almacenados en el PIC 16F628A si fuera necesario.

2.1 DESARROLLO DEL HARDWARE.

Se implemento en proto board el circuito de control, con el fin de realizar

mediciones en el tiempo de respuesta ante una alarma, y corregir las fallas que se presenten.

En la figura 2.1 se observa el circuito armado previamente en el protoboard.

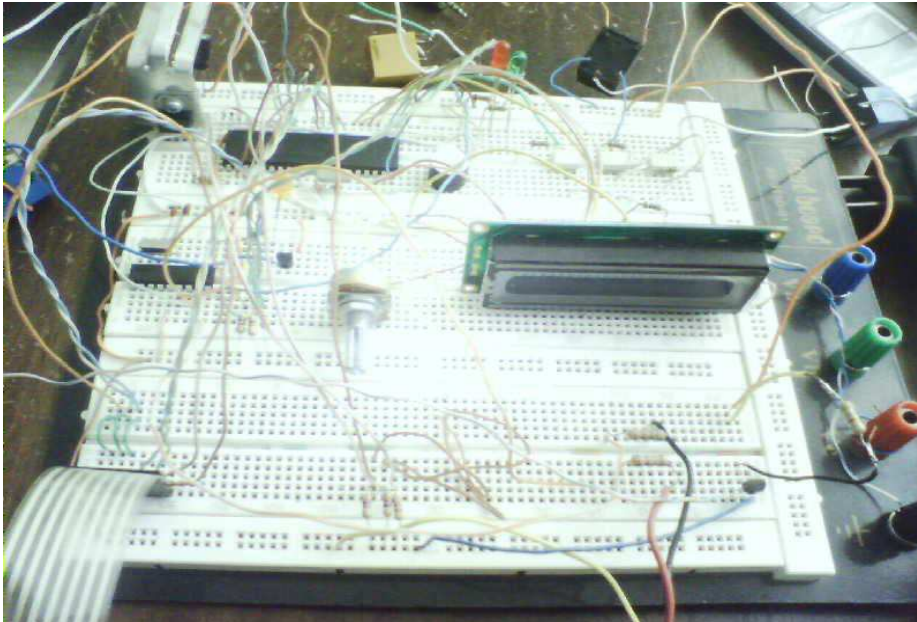


FIG 2.1: Pruebas en el protoboard.

Una vez que el circuito paso las pruebas de funcionamiento en el protoboard, procedemos al diseño del circuito impreso.

A continuación se detalla uno de los programas más útiles para el desarrollo de proyectos electrónicos, este software es PROTEUS de la empresa Labcenter Electronics Ltd , el mismo que fue utilizado para la realización del equipo.

2.1.1 DESARROLLO DEL DIAGRAMA CIRCUITAL EN PROTEUS²¹.

PROTEUS es un entorno de desarrollo, el cual integra diversas herramientas de software para las tareas más comunes, en el desarrollo de proyectos electrónicos tales como: captura de esquemáticos, fabricación de circuitos impresos y simulaciones. La figura 2.2 muestra la ventana principal del software proteus.

²¹Texto y grafico tomados de : www.labcenter.com



FIG 2.2: Ventana principal del programa proteus²¹.

La descripción de la función de cada uno de los componentes del software mostrados en la figura anterior se describe a continuación.

Barra de título: Situada en la parte superior de la pantalla, en ella se muestra el icono del programa, el nombre del fichero abierto, la leyenda ISIS Professional, y en ocasiones mensajes de que el programa ha entrado en un modo particular de funcionamiento (por ejemplo *Animating* cuando se simula).

Barra de menús: Permite el acceso a la mayor parte de las opciones del programa, sin embargo algunas opciones sólo están disponibles en los iconos de las barras de herramientas.

Barras de herramientas: Son varias y presentan las opciones para manejar los elementos del esquemático, tales como colocación de dispositivos, manejo de librerías, visualización, rotación de componentes, operaciones sobre bloques de dispositivos; así como las opciones de animación del diseño, entre otras.

Zona de trabajo: Es donde se colocará el diseño a realizar para posteriormente simularlo.

Ventana de vista completa/Zoom/Mapa del diseño: Esta ventana nos muestra una visión global del diseño, y mediante el puntero podemos seleccionar que zona del diseño estará visible en la ventana de edición, si no fuese posible visualizar todo sobre dicha ventana. La zona visible se encuentra encuadrada dentro de dicha ventana, mediante un recuadro verde.

Lista de componentes: En esta ventana aparecerán todos los componentes, terminales, pines, generadores, etc. que se quieran introducir en el diagrama esquemático, esta ventana dispone de 2 botones, los cuales nos permiten acceder a las librerías de componentes incluidas en ISIS.

Barra de estado: Situada en la parte inferior de la pantalla, en ella se muestran mensajes informativos acerca de las opciones del menú, de los componentes, y de las simulaciones, a la derecha se indican las coordenadas de la posición del cursor, las unidades son en milésimas de pulgada.

El diagrama del equipo fue realizado en proteus, por la flexibilidad que posee en la simulación, diseño, y construcción de placas de circuito impreso.

En el anexo D se encuentran los diagramas que componen el equipo.

2.1.2 REALIZACIÓN DE LA PLACA IMPRESA.

A continuación tenemos las figuras 2.3, 2.4, y 2.5, donde se muestran las placa que conforman todo el equipo.

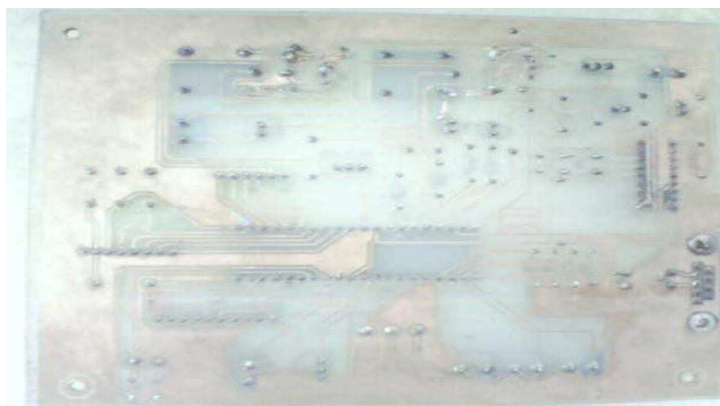


FIG 2.3: Placa impresa del circuito del equipo.

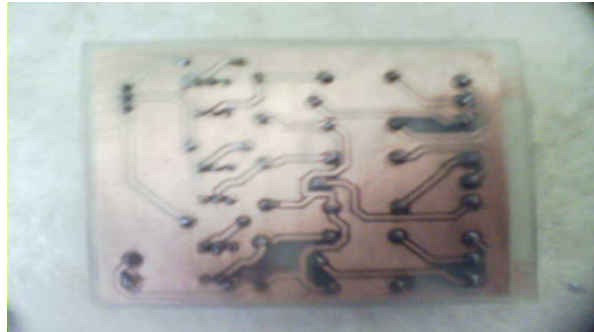


FIG 2.4: Placa impresa del circuito de relés.



FIG 2.5: Placa impresa del circuito de la fuente.

2.1.3 ENSAMBLAJE DE LA INTERFAZ DE CONTROL.

Después de realizar las pruebas en el protoboard y probar el funcionamiento del sistema de seguridad, se procede al ensamblaje del equipo. Aquí podemos apreciar el lado de los elementos ya soldados en la fuente, los cables rojo y negro son para la conexión de la batería. Desde la figura 2.6 hasta la figura 2.11 se puede apreciar las placas y el equipo terminado.



FIG 2.6: Fuente con sus elementos terminada

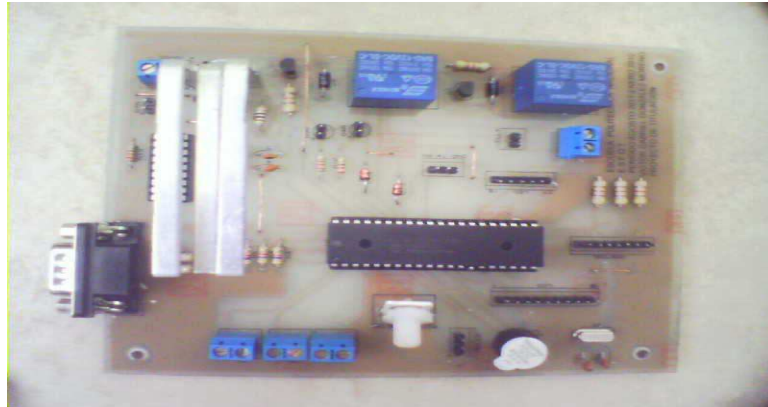


FIG 2.7: Placa madre del sistema con sus elementos terminada.

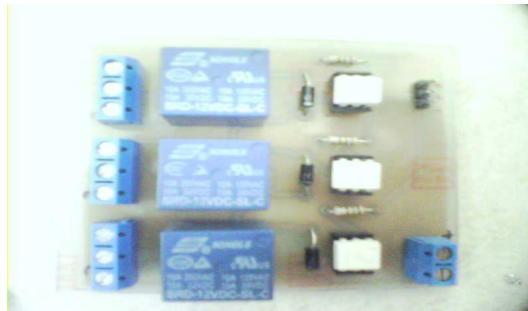


FIG 2.8: Placa de relés que encienden las luces en la casa prototipo.



FIG 2.9: Placas dentro de la caja que contienen todo el proyecto.



FIG 2.10: Transformador y batería de respaldo dentro de la caja.



FIG 2.11: LCD, teclado y leds indicativos en la parte frontal de la caja.

2.2 PROGRAMACIÓN DEL SOFTWARE DE CONTROL.

El software de comunicación entre el sistema de seguridad y el computador ha sido diseñado en el lenguaje de programación Visual BASIC 6.0. Se ha escogido este lenguaje debido a que es una herramienta de diseño de aplicaciones para Windows.

2.2.1 EL LENGUAJE VISUAL BASIC 6.0²².

Visual Basic 6.0 es un lenguaje de programación visual, también llamado lenguaje de 4^a generación.

²²Texto y gráficos tomados de: Curso Completo de Visual Basic 6.0, Escuela Superior de Ingenieros Industriales, UNIVERSIDAD DE NAVARRA, Javier García de Jalón · José Ignacio Rodríguez

Esto quiere decir que un gran número de tareas se realizan sin escribir código, simplemente con operaciones gráficas realizadas con el ratón sobre la pantalla. Es una excelente herramienta de programación que permite crear aplicaciones propias (programas) para Windows 95/98/ XP/ o Windows Vista.

2.2.2 LAS PANTALLAS DE VISUAL BASIC.

Cuando se arranca Visual Basic 6.0 aparece una pantalla similar a la de la figura 2.12.

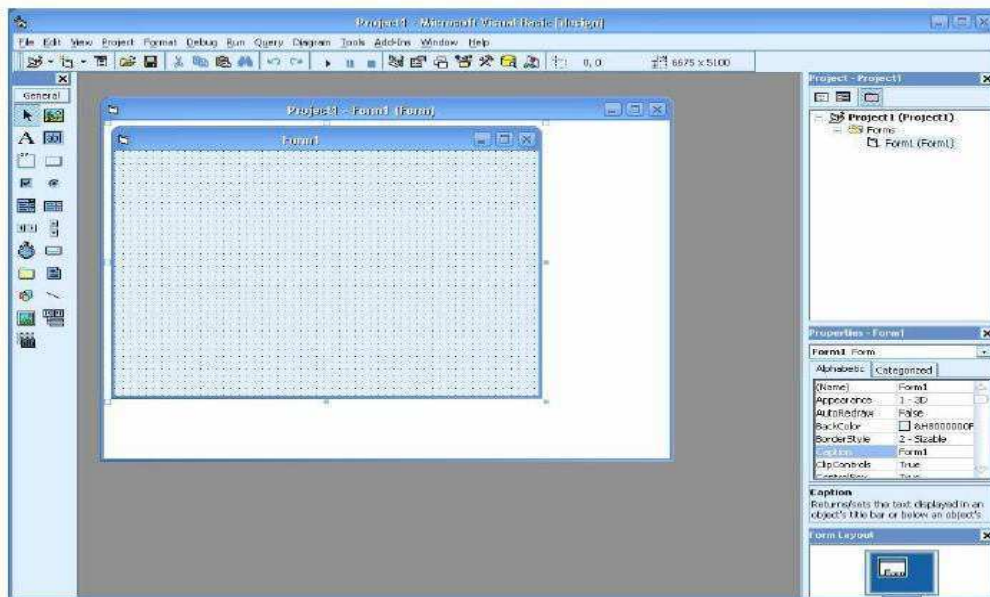


FIG 2.12: Pantalla principal de visual Basic 6.0²²

En ella se pueden distinguir los siguientes elementos:

La barra de menús y la barra de herramientas estándar.

La barra de menús de Visual Basic 6.0 resulta similar a la de cualquier otra aplicación de Windows. En la figura 2.13 se aprecia la barra de menú.

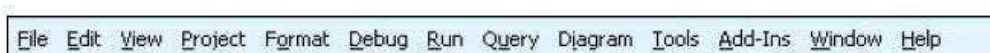


FIG 2.13: Barra de menú de Visual Basic 6.0²²

La Barra de Herramientas Estándar aparece debajo de la barra de menús, que permite acceder a las opciones más importantes de los menús. En Visual Basic

6.0 existen cuatro barras de herramientas: Debug, Edit, FormEditor y Estándar, por defecto sólo aparece la barra de herramientas estándar.

Haciendo clic con el botón derecho del mouse sobre cualquier parte de la barra de herramientas, aparece un menú contextual con el que se puede hacer aparecer y ocultar cualquiera de las barras. La barra de herramientas se aprecia en la figura 2.14.

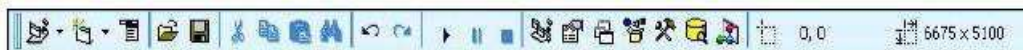


FIG 2.14: Barra de herramientas de Visual Basic 6.0²²

La barra de herramientas no estándar (Toolbox).



Esta incluye los controles con los que se puede diseñar la pantalla de la aplicación. Estos controles son por ejemplo, botones, etiquetas, cajas de texto, zonas gráficas, etc.

Para introducir un control en el formulario simplemente hay que hacer clic con el botón izquierdo del mouse sobre el control deseado, y colocarlo en el formulario con la posición y el tamaño deseado. Haciendo doble clic sobre el control es también otra forma de colocar el control en el formulario, quedando este ubicado en el centro del formulario. En la figura 2.15 aparece el TOOLBOX .

FIG 2.15: Tool Box de Visual Basic 6.0²²

Formularios (Forms).

Los formularios son las zonas de la pantalla sobre las que se diseña el programa, y sobre los que sitúan los controles o herramientas del Toolbox. Al ejecutar el programa, el Form se convertirá en la ventana de la aplicación donde aparecerán los botones, las cajas de texto, los gráficos, etc. En la figura 2.16 se puede apreciar el form de visual Basic 6.0.

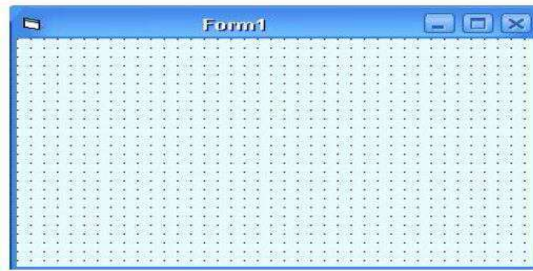


FIG 2.16: Formulario de diseño²².

El conjunto de puntos que aparecen sobre el formulario se llama malla o (grid), ésta permite alinear los controles manualmente de una forma precisa, evitando tener que introducir coordenadas continuamente. Esta malla sólo será visible en el proceso de diseño del programa; al ejecutarlo la malla automáticamente desaparece.

La ventana de propiedades (Properties).

En la ventana de propiedades se muestran todas las características de un control seleccionado en la aplicación. Mediante esta ventana es posible cambiar los valores de las propiedades de cada uno de los controles, indicando así sus características antes de ejecutar la aplicación.



A medida que se seleccionan los controles en la aplicación, se puede apreciar que las propiedades que antes mostraba la ventana desaparecen, y muestran las propiedades del control que ha sido seleccionado actualmente.

La cantidad de propiedades que aparecen en la ventana, depende de las características propias de ese control. La figura 2.17 muestra la ventana de propiedades del objeto.

FIG 2.17: Ventana de propiedades del objeto²².

El Code Editor.

El Editor de Código de Visual Basic 6.0 es la ventana en la cual se escriben las sentencias del programa.

Esta ventana presenta algunas características muy interesantes que conviene conocer, para sacar el máximo partido de la aplicación. Para abrir la ventana del editor de código se elige Code en el menú View. También se abre haciendo clic en el botón View Code de la ventana de proyecto (Project), o haciendo doble clic en el formulario o cualquiera de sus controles. En la figura 2.18 se aprecia la ventana de código.

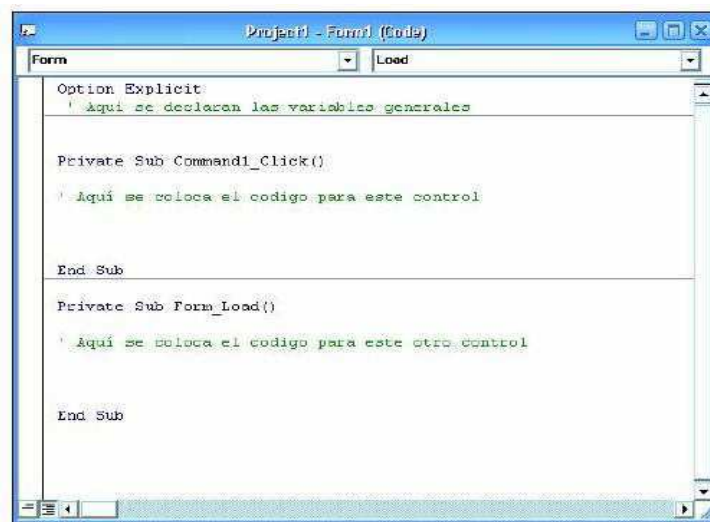


FIG 2.18: Ventana para editar el código del programa²².

2.2.3 COMUNICACIÓN SERIE DESDE VISUAL BASIC.

La herramienta MSComm está presente en Visual Basic 6.0, y permite la comunicación de una aplicación con el puerto serie. Mediante este control se consigue crear aplicaciones que envíen y reciban datos a través del puerto serial con el estándar RS232.

Para la habilitación de la comunicación serial, damos un clic con el botón derecho sobre la barra de herramientas, y escogemos la opción componentes. La figura 2.19 muestra el menú con la opción componentes de visual Basic.

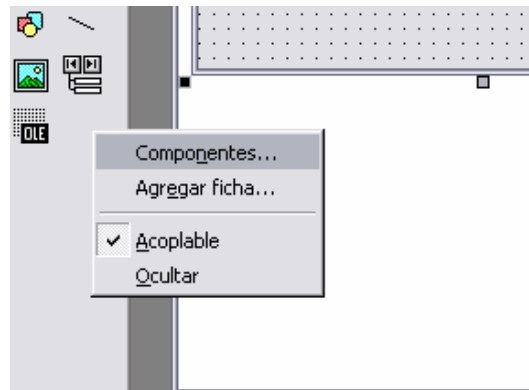


FIG 2.19: Menú escondido en el Tool Box.

Luego saldrá una pantalla con una lista de componentes y buscamos Microsoft Com Control 6.0, seleccionamos y damos un clic en aceptar. Esto se aprecia en la figura 2.20

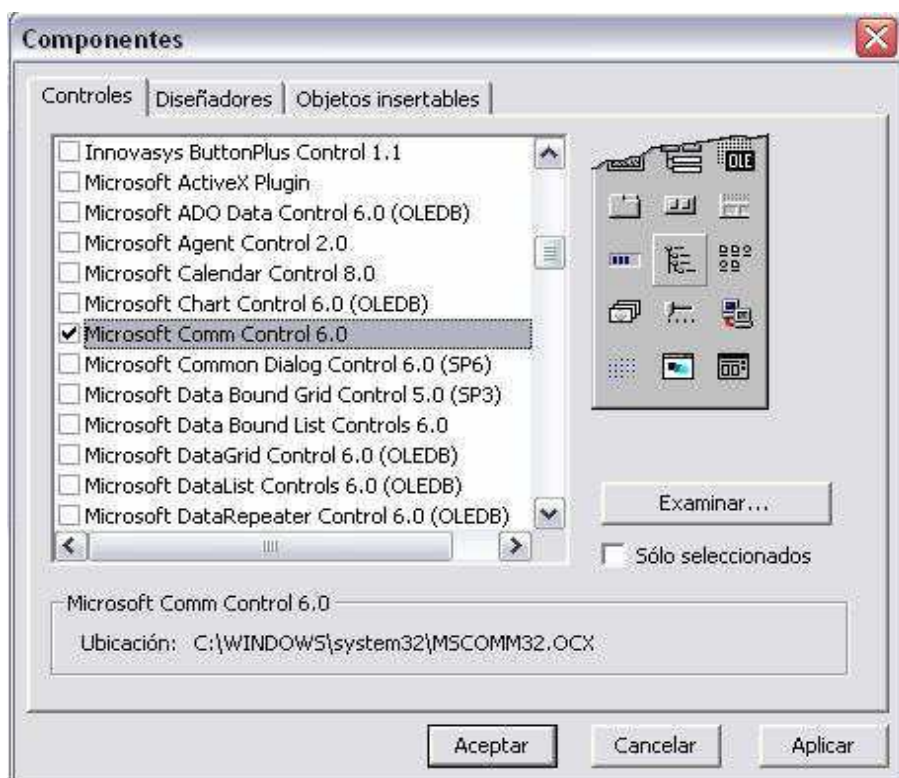



FIG 2.20: Ventana para seleccionar la Comm Control

Notaremos que ahora aparece un icono de un teléfono,  colocamos este teléfono en el form, y en las propiedades colocamos los valores indicados en la tabla 2.1.

PROPIEDAD	VALOR
Bits por Segundo	9600
Bits de Datos	8
Paridad	Ninguno
Bits de Parada	1
Control de Flujo	Ninguno

TABLA 2.1: Configuración utilizada para la comunicación serie

Luego de esto nuestro proyecto esta habilitado para transmitir en forma bidireccional datos por el puerto serie.

2.4.4 CREAR PROYECTOS EJECUTABLES DESDE VISUAL BASIC.

Una vez finalizada la programación de la nueva aplicación, la siguiente tarea es comprimir el programa en un solo archivo ejecutable, para su distribución e instalación en cuantos computadores se desee, incluso aunque en ellos no este instalado Visual Basic 6.0. Para crear un programa ejecutable se utiliza la opción “ Generar proyecto .exe” , que se encuentra en las opciones del menú archivo. En la figura 2.21 se aprecia la opción para generar archivos ejecutables.



FIG 2.21: Opción para generar archivos ejecutables.

Cuando se selecciona dicha opción se despliega la ventana de dialogo mostrada en la figura 2.22, desde la cual se indica la ubicación y el nombre del nuevo archivo ejecutable.

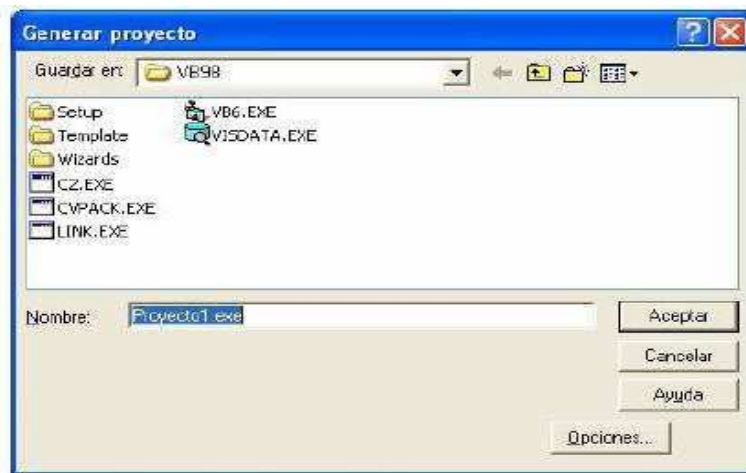


FIG 2.22: Ventana para guardar un proyecto con extensión .exe.

En el anexo E, que corresponde al manual de usuario, se puede visualizar como quedo terminado el software de control, y su funcionamiento.

2.5 ENSAMBLAJE DE TODO EL PROYECTO.

Para observar el funcionamiento del equipo es necesario contar con una maqueta o casa prototipo, donde se pueda simular los eventos que ocurran en una casa normal.

2.5.1 INSTALACIÓN DE LUCES Y SENSORES EN LA MAQUETA.

La casa prototipo donde se van ha realizar todas las pruebas se muestra en la figura 2.23.



FIG 2.23: Casa prototipo para hacer pruebas.

La casa prototipo tiene instaladas luces que funcionan con 110 VAC en las ventanas y el techo.

La señal de apertura de la puerta o una de las ventanas es dada por los reedswitch instalados en la maqueta. En la figura 2.24 se aprecia un reed switch colocado en la puerta de la casa prototipo.



FIG 2.24: Reed switch instalado en la casa prototipo.

2.5.2 CONEXIÓN ENTRE LA MAQUETA Y LA INTERFAZ DE CONTROL

La conexión entre el sistema de alarma y control de luces con la casa prototipo, se la hace a través de un total de 6 cables de par trenzado, de los cuales 3 controlan las luces, y el resto corresponden a las 3 zonas que tiene la placa madre, y dan la señal de alarma al sistema. En la figura 2.25 aparece las conexiones entre la casa prototipo y el equipo.

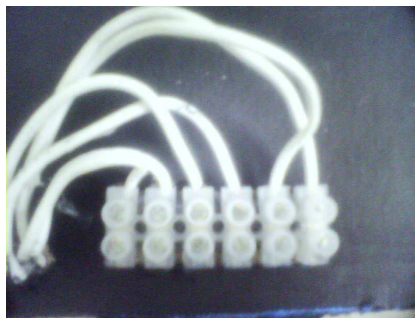


FIG 2.25: Borneras con las conexiones entre la casa prototipo y el sistema de alarma.

2.6 PRUEBAS Y CALIBRACIÓN.

Con el equipo terminado y en funcionamiento, se procede a realizar las respectivas pruebas de hardware y software.

En las pruebas de software, el programa para cambios de usuario, muestra un mensaje de error cuando se selecciona un puerto "com" que no existe en el computador. En la figura 2.26 se aprecia este error.



FIG 2.26: Mensaje de error en el software para cambios de usuario.

Aparte de este error el programa pasó las pruebas sin más novedades.

El hardware cumplió todas las funciones programadas sin problemas, tanto con corriente alterna, como con la batería de respaldo sin presentar novedades.

2.6.1 TIEMPO DE RESPUESTA ANTE UNA ALARMA.

Para este análisis se debe tomar en cuenta que este sistema de seguridad trabaja en base a la red GSM, por lo tanto el tiempo de respuesta ante una alarma, depende mayormente de la cantidad de señal celular que exista en el lugar donde este instalado.

Una vez que tenemos claro esto, empezamos revisando cuales son los procesos que el microcontrolador ejecuta constantemente.

Cuando el sistema de alarma se encuentra en modo alerta este realiza internamente la ejecución de los siguientes procesos:

- a) La lectura del estado de batería que demora 1.5 segundos.
- b) La revisión en el celular de un nuevo mensaje de texto, este proceso tarda 1.5 segundos

Entre cada uno de estos procesos existe un tiempo de 2 segundos, durante el cual, el microcontrolador realiza una revisión de cada una de las zonas para detectar la apertura de alguna de ellas; este proceso también ocurre cuando el microcontrolador revisa el estado de la batería del celular, y la llegada de un nuevo SMS, así que, después que el usuario ingresa la clave correcta para armar el sistema, el microcontrolador permanece ejecutando un ciclo infinito, el cual se compone de la revisión de las zonas, el estado de la batería del

celular, y la llegada de un SMS, esto se aprecia en la figura 2.27.

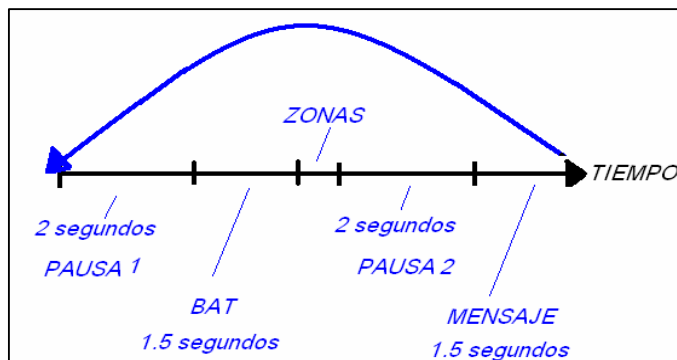


FIG 2.27: Ciclo de revisión del PIC.

Después que el equipo detecta una zona abierta, durante 6 segundos enciende y apaga las luces de la casa prototipo; luego el microcontrolador tarda 6.4

segundos en enviar las señales necesarias hacia el celular, para que éste envíe el primer mensaje de alerta; luego hace una pausa de 5 segundos, y después tarda otros 6.4 segundos en enviar el segundo mensaje de alerta. Por lo tanto el tiempo que transcurre desde que una zona es abierta hasta que el SMS de alerta llega al primer usuario es aproximadamente 9 segundos, y para el segundo usuario el tiempo será aproximadamente 19 segundos.

El análisis descrito anteriormente es válido para las zonas 2 y 3, ya que la zona 1 tiene un tiempo de 10 segundos de espera, para que el usuario desarme el sistema sin que ocurra una alarma, por lo tanto, si llega a ocurrir una alarma en la zona 1, el tiempo que demoran en llegar los SMS a los usuarios es 10 segundos más largo; 19 segundos para el primer usuario y 29 segundos para el segundo usuario.

2.6.2 ANÁLISIS DE RESULTADOS.

Se realizaron pruebas en un lugar donde la cobertura GSM es baja para verificar el comportamiento del equipo en el peor de los casos; se obtuvieron los siguientes resultados:

Los SMS enviados desde el equipo hacia los usuarios, tardan más tiempo (aproximadamente 1 minuto más tarde) en llegar a sus destino; además si el

usuario envía mas de un SMS para activar las luces de la casa prototipo , el equipo no responde , ya que los SMS se encuentran retenidos en la red GSM hasta que el dispositivo este listo, y cuando la red despacha los SMS retenidos, estos llegan casi al mismo tiempo, y el microcontrolador no alcanza a leer la información necesaria para activar cualquiera de las luces.

También se observo que si la batería del celular tiene una carga menor al 50%, los comandos enviados por el microcontrolador hacia el MODEM no funcionan en su totalidad, solo un 10% de ellos funcionó correctamente.

Las observaciones descritas anteriormente, indican que este equipo tiene retardos en la transmisión como en la recepción de información, pero esta es una falla ajena al equipo.

CONCLUSIONES

Las conclusiones más importantes acerca de este proyecto son:

El momento en que exista una caída en la red GSM este equipo queda inhabilitado para la transmisión de una alarma.

Este sistema es una solución de bajo costo, por lo que puede competir en el mercado con otros dispositivos similares.

El equipo pasó todas las pruebas de funcionamiento y configuración, por lo tanto se concluye que el sistema satisface las necesidades para la que fue diseñado.

Un plan de mensajes mensual es suficiente, para que el sistema pueda enviar los SMS que sean necesarios, si ocurre una alarma cualquier momento durante el mes.

El celular no puede llegar a un estado de batería baja, ya que el PIC siempre está monitoreando el nivel de carga.

El programa para cambiar los usuarios y la clave del sistema, no funciona si en la conexión se utiliza un cable con una longitud superior a 2 metros

El celular motorola C115 utilizado en este proyecto, soporta un amplio conjunto de comandos AT, que son suficientes como para implementar aplicaciones más complejas.

RECOMENDACIONES.

Las recomendaciones más importantes acerca de este proyecto son:

Realizar un borrado manual de los mensajes almacenados en el celular una vez al mes, así no existe riesgo que la bandeja de entrada se llene, y la memoria del celular no permita el ingreso de más mensajes.

Utilizar el circuito integrado MAX 232, para comunicaciones seriales con un computador, en distancias superiores a los 2 metros e inferiores a 15 metros.

Realizar un chequeo de la batería, después de que se haya producido un corte de luz, ya que si ésta se encuentra baja, no podrá respaldar al equipo en caso de otro corte de energía.

Instalar este equipo en lugares donde exista una buena cobertura celular, para no tener problemas en la transmisión de alarmas.

Hacer pruebas de funcionamiento frecuentemente, ya que todo sistema de seguridad debe funcionar correctamente los 365 días del año.

El mantenimiento del equipo lo debe hacer una persona con conocimiento en electrónica.

Se recomienda no utilizar los comandos AT en celulares marca NOKIA, ya que estos tienen sus propios protocolos de comunicación, los cuales son MBUS y FBUS.

ANEXO A
TABLA DE
CÓDIGOS DE
CONTROL ASCCI

	code pos.		Unicode		Description in <i>C0 of ISO 646</i>
	dec.	hex.	abbr.	name	
ctl-H	8	8	BS	BACKSPACE	A format effector which moves the active position one character position backwards on the same line.
ctl-I	9	9	HT	HORIZONTAL TABULATION	A format effector which advances the active position to the next pre-determined character position on the same line.
ctl-J	10	A	LF	LINE FEED	A format effector which advances the active position to the same character position of the next line.
ctl-K	11	B	VT	VERTICAL TABULATION	A format effector which advances the active position to the same character position on the next pre-determined line.
ctl-L	12	C	FF	FORM FEED	A format effector which advances the active position to the same character position on a pre-determined line of the next form or page.
ctl-M	13	D	CR	CARRIAGE RETURN	A format effector which moves the active position to the first character position on the same line.
ctl-N	14	E	SO	SHIFT OUT	A control character which is used in conjunction with SHIFT IN and ESCAPE to extend the graphic character set of the code. It may alter the meaning of octets 33 - 126 (dec.). The effect of this character when using code extension techniques is described in International Standard ISO 2022.
ctl-O	15	F	SI	SHIFT IN	A control character which is used in conjunction with SHIFT OUT and ESCAPE to extend the graphic character set of the code. It may reinstate the standard meanings of the octets which follow it. The effect of this character when using code extension techniques is described in International Standard ISO 2022.
ctl-P	16	10	DLE	DATA LINK ESCAPE	A transmission control character which will change the meaning of a limited number of contiguously following characters. Its is used exclusively to provide supplementary data transmission control functions. Only graphic characters and transmission control characters can be used in DLE sequences.
ctl-Q	17	11	DC1	DEVICE CONTROL ONE	A device control character which is primarily intended for turning on or starting an ancillary device. If it is not required for this purpose, it

	code pos.		Unicode		Description in <i>C0 of ISO 646</i>
	dec.	hex.	abbr.	name	
					between sender and recipient.
ctl-Y	25	19	EM	END OF MEDIUM	A control character that may be used to identify the physical end of a medium, or the end of the used portion of a medium, or the end of the wanted portion of data recorded on a medium. The position of this character does not necessarily correspond to the physical end of the medium.
ctl-Z	26	1A	SUB	SUBSTITUTE	A control character used in the place of a character that has been found to be invalid or in error. SUB is intended to be introduced by automatic means.
ctl-[27	1B	ESC	ESCAPE	A control character which is used to provide additional control functions. It alters the meaning of a limited number of contiguously following bit combinations. The use of this character is specified in International Standard ISO 2022.
ctl-\	28	1C	FS	FILE SEPARATOR	A control character used to separate and qualify data logically; its specific meaning has to be specified for each application. If this character is used in hierarchical order, it delimits a data item called a <i>file</i> .
ctl-]	29	1D	GS	GROUP SEPARATOR	A control character used to separate and qualify data logically; its specific meaning has to be specified for each application. If this character is used in hierarchical order, it delimits a data item called a <i>group</i> .
ctl-^	30	1E	RS	RECORD SEPARATOR	A control character used to separate and qualify data logically; its specific meaning has to be specified for each application. If this character is used in hierarchical order, it delimits a data item called a <i>record</i> .
ctl_	31	1F	US	UNIT SEPARATOR	A control character used to separate and qualify data logically; its specific meaning has to be specified for each application. If this character is used in hierarchical order, it delimits a data item called a <i>unit</i> .
	127	7F	DEL	DELETE	(not defined)

ANEXO B

PROGRAMAS DE LOS

MICROCONTROLADORES

16F877 Y 16F628A

PROGRAMA DEL PIC 16F877

```
define OSC 20
INCLUDE "modedefs.bas"
ADCON1=7 ; CONVIERTE TODOS LOS PUERTOS ANALOGICOS EN
          DIGITALES

ledrojo var portb.2
ledverde var portb.3
bip var porte.0
BATCELL VAR portb.7
RELE1 VAR porta.3
RELE2 VAR porte.1
RELE3 VAR porta.5

ZONA1 VAR porta.0
ZONA2 VAR porta.1
ZONA3 VAR porta.2

A VAR portd.7
B VAR portd.6
C VAR portd.5
D VAR portd.4

UNO VAR portd.3
DOS VAR portd.2
TRES VAR portd.1

T1 VAR BYTE
T2 VAR BYTE
T3 VAR BYTE
T4 VAR BYTE

CLA1 VAR BYTE
```


CLA2 VAR BYTE
CLA3 VAR BYTE
CLA4 VAR BYTE

CLP1 VAR BYTE
CLP2 VAR BYTE
CLP3 VAR BYTE
CLP4 VAR BYTE

CLP1="5"
CLP2="5"
CLP3="5"
CLP4="5"

NUE VAR BYTE
W VAR BYTE
Y VAR BYTE
T VAR BYTE
X VAR BYTE
m var byte
n var byte
o var byte
NUM0 VAR BYTE

W="G"
NUM0="0"
m="D"
N="D"
O="D"

NUMA1 VAR BYTE;numero de usuario1
NUMB1 VAR BYTE
NUMC1 VAR BYTE
NUMD1 VAR BYTE

NUME1 VAR BYTE
NUMF1 VAR BYTE
NUMG1 VAR BYTE
NUMH1 VAR BYTE

NUMA2 VAR BYTE; numero de usuario 2
NUMB2 VAR BYTE
NUMC2 VAR BYTE
NUMD2 VAR BYTE
NUME2 VAR BYTE
NUMF2 VAR BYTE
NUMG2 VAR BYTE
NUMH2 VAR BYTE

A1 VAR BYTE ; VARIABLES QUE ALMACENAN LA INFORMACION DEL
CELULAR

A2 VAR BYTE
A3 VAR BYTE
A4 VAR BYTE
A5 VAR BYTE
A6 VAR BYTE
A7 VAR BYTE
A8 VAR BYTE
A9 VAR BYTE
A0 VAR BYTE
B1 VAR BYTE
B2 VAR BYTE
B3 VAR BYTE
B4 VAR BYTE
B5 VAR BYTE
B6 VAR BYTE
B7 VAR BYTE
B8 VAR BYTE
B9 VAR BYTE

B0 VAR BYTE
C1 VAR BYTE
C2 VAR BYTE
C3 VAR BYTE
C4 VAR BYTE
C5 VAR BYTE
C6 VAR BYTE
C7 VAR BYTE
C8 VAR BYTE
C9 VAR BYTE
C0 VAR BYTE
D1 VAR BYTE
D2 VAR BYTE
D3 VAR BYTE
D4 VAR BYTE
D5 VAR BYTE
D6 VAR BYTE
D7 VAR BYTE
D8 VAR BYTE
D9 VAR BYTE
D0 VAR BYTE
E1 VAR BYTE
E2 VAR BYTE
E3 VAR BYTE
E4 VAR BYTE
E5 VAR BYTE
E6 VAR BYTE
E7 VAR BYTE
E8 VAR BYTE
E9 VAR BYTE
E0 VAR BYTE
F1 VAR BYTE
F2 VAR BYTE
F3 VAR BYTE

F4 VAR BYTE
F5 VAR BYTE
F6 VAR BYTE
F7 VAR BYTE
F8 VAR BYTE
F9 VAR BYTE
F0 VAR BYTE
G1 VAR BYTE
G2 VAR BYTE
G3 VAR BYTE
G4 VAR BYTE
G5 VAR BYTE
G6 VAR BYTE
G7 VAR BYTE
G8 VAR BYTE
G9 VAR BYTE
G0 VAR BYTE
H1 VAR BYTE
H2 VAR BYTE
H3 VAR BYTE
H4 VAR BYTE
H5 VAR BYTE
H6 VAR BYTE
H7 VAR BYTE
H8 VAR BYTE
H9 VAR BYTE
H0 VAR BYTE
I1 VAR BYTE
I2 VAR BYTE
I3 VAR BYTE
I4 VAR BYTE
I5 VAR BYTE
I6 VAR BYTE
I7 VAR BYTE

I8 VAR BYTE
I9 VAR BYTE
I0 VAR BYTE
J1 VAR BYTE
J2 VAR BYTE
J3 VAR BYTE
J4 VAR BYTE
J5 VAR BYTE
J6 VAR BYTE
J7 VAR BYTE
J8 VAR BYTE
J9 VAR BYTE
J0 VAR BYTE
K1 VAR BYTE
K2 VAR BYTE
K3 VAR BYTE
K4 VAR BYTE
K5 VAR BYTE
K6 VAR BYTE
K7 VAR BYTE
K8 VAR BYTE
K9 VAR BYTE

EMPEZAR:
for x=0 to 20
HIGH ledrojo
high ledverde
HIGH bip
pause 50
low ledrojo
low ledverde
low bip
pause 50
next

PAUSE 15000

high ledverde

serout portb.5,N2400,["C"]

SERIN portb.4,N2400,NUMA1;NUMERO 1 DEL CELULAR

SERIN portb.4,N2400,NUMB1

SERIN portb.4,N2400,NUMC1

SERIN portb.4,N2400,NUMD1

SERIN portb.4,N2400,NUME1

SERIN portb.4,N2400,NUMF1

SERIN portb.4,N2400,NUMG1

SERIN portb.4,N2400,NUMH1

SERIN portb.4,N2400,NUMA2;NUMERO 2 DEL CELULAR

SERIN portb.4,N2400,NUMB2

SERIN portb.4,N2400,NUMC2

SERIN portb.4,N2400,NUMD2

SERIN portb.4,N2400,NUME2

SERIN portb.4,N2400,NUMF2

SERIN portb.4,N2400,NUMG2

SERIN portb.4,N2400,NUMH2

SERIN portb.4,N2400,CLA1

SERIN portb.4,N2400,CLA2

SERIN portb.4,N2400,CLA3

SERIN portb.4,N2400,CLA4

LOW ledverde

pause 100

serout2 PORTB.0,84,["AT",13]

PAUSE 2000

serout2 PORTB.0,84,["AT",13]

PAUSE 1000

HIGH ledverde

high bip

```
PAUSE 1000
DEFINE LCD_DREG PORTC
DEFINE LCD_DBIT 4
DEFINE LCD_RSREG PORTC
DEFINE LCD_RSBIT 3
DEFINE LCD_EREG PORTC
DEFINE LCD_EBIT 2
PAUSE 500
LCDOUT $FE,1
PAUSE 200
LCDOUT $FE,$80,"CONECTE EL"
LCDOUT $FE,$C0,"CELULAR"
PAUSE 5000
LOW ledverde
low bip
```

pause 2000

```
LCDOUT $FE,1
LCDOUT
    $FE,$80,NUMA1,NUMB1,NUMC1,NUMD1,NUME1,NUMF1,NUMG1,N
    UMH1
LCDOUT
    $FE,$C0,NUMA2,NUMB2,NUMC2,NUMD2,NUME2,NUMF2,NUMG2,N
    UMH2
PAUSE 15000
LCDOUT $FE,1
PAUSE 200
LCDOUT $FE,$80,CLA1,CLA2,CLA3,CLA4
PAUSE 5000

LCDOUT $FE,1
```

```
PAUSE 200
LCDOUT $FE,$82,"CONTROL-SMS"
```

```
PAUSE 1000
HIGH ledrojo
pause 1000
```

```
HIGH RELE1
PAUSE 1000
LOW RELE1
PAUSE 1000
```

```
HIGH RELE2
PAUSE 1000
LOW RELE2
PAUSE 1000
```

```
HIGH RELE3
PAUSE 1000
LOW RELE3
```

```
PAUSE 1000
HIGH ledverde
serout2 PORTB.0,84,["AT",13]
PAUSE 1000
serout2
    PORTB.0,84,["AT+CPMS=",34,"SM",34,44,34,"SM",34,44,34,"SM",34,1
    3]
pause 2000
serout2
    PORTB.0,84,["AT+CPMS=",34,"ME",34,44,34,"ME",34,44,34,"ME",34,1
    3]
PAUSE 2000
```



```
serout2 PORTB.0,84,["AT+CMGF=1",13];MODO TEXTO
PAUSE 500
serout2
    PORTB.0,84,["AT+CMGS=",34,NUM0,NUMA1,NUMB1,NUMC1,NUMD
    1,NUME1,NUMF1,NUMG1,NUMH1,34,13]
PAUSE 300
SEROUT2 PORTB.0,84,["BIENVENIDO-",m,n,o,26]; ENVIAR
PAUSE 300
serout2
    PORTB.0,84,["AT+CMGS=",34,NUM0,NUMA1,NUMB1,NUMC1,NUMD
    1,NUME1,NUMF1,NUMG1,NUMH1,34,13]
PAUSE 300
SEROUT2 PORTB.0,84,["BIENVENIDO-",m,n,o,26]; ENVIAR

PAUSE 5000
serout2 PORTB.0,84,["AT",13]
PAUSE 1000
serout2
    PORTB.0,84,["AT+CPMS=",34,"SM",34,44,34,"SM",34,44,34,"SM",34,1
    3]
pause 2000
serout2
    PORTB.0,84,["AT+CPMS=",34,"ME",34,44,34,"ME",34,44,34,"ME",34,1
    3]
PAUSE 2000
serout2 PORTB.0,84,["AT+CMGF=1",13];MODO TEXTO
PAUSE 500
serout2
    PORTB.0,84,["AT+CMGS=",34,NUM0,NUMA2,NUMB2,NUMC2,NUMD
    2,NUME2,NUMF2,NUMG2,NUMH2,34,13]
PAUSE 300
SEROUT2 PORTB.0,84,["BIENVENIDO-",m,n,o,26]; ENVIAR
PAUSE 300
serout2
```

```
PORTB.0,84,["AT+CMGS=",34,NUM0,NUMA2,NUMB2,NUMC2,NUMD
2,NUME2,NUMF2,NUMG2,NUMH2,34,13]
PAUSE 300
SEROUT2 PORTB.0,84,["BIENVENIDO",m,n,o,26]; ENVIAR
PAUSE 300
low ledverde

INICIO:
serout2
    PORTB.0,84,["AT+CPMS=",34,"SM",34,44,34,"SM",34,44,34,"SM",34,1
    3]
pause 2000
serout2
    PORTB.0,84,["AT+CPMS=",34,"ME",34,44,34,"ME",34,44,34,"ME",34,1
    3]
PAUSE 2000
serout2 PORTB.0,84,["AT+CMGD=1",13]
PAUSE 500

serout portb.5,N2400,["A"]
PAUSE 100
serout2 PORTB.0,84,["AT+CBC",13]
serIN2 PORTB.1,84,[A1]
serIN2 PORTB.1,84,[A2]
serIN2 PORTB.1,84,[A3]
serIN2 PORTB.1,84,[A4]
serIN2 PORTB.1,84,[A5]
serIN2 PORTB.1,84,[A6]
serIN2 PORTB.1,84,[A7]
serIN2 PORTB.1,84,[A8]
serIN2 PORTB.1,84,[A9]
serIN2 PORTB.1,84,[A0]
serIN2 PORTB.1,84,[B1]
serIN2 PORTB.1,84,[B2]
```

```
serIN2 PORTB.1,84,[B3]
serIN2 PORTB.1,84,[B4]
serIN2 PORTB.1,84,[B5]
serIN2 PORTB.1,84,[B6];DATO PARA LEER
serIN2 PORTB.1,84,[B7]
serIN2 PORTB.1,84,[B8]
serIN2 PORTB.1,84,[B9]
serIN2 PORTB.1,84,[B0]
serIN2 PORTB.1,84,[C1]
serIN2 PORTB.1,84,[C2]
serIN2 PORTB.1,84,[C3]
serIN2 PORTB.1,84,[C4]
serIN2 PORTB.1,84,[C5]
serIN2 PORTB.1,84,[C6]
serIN2 PORTB.1,84,[C7]
serIN2 PORTB.1,84,[C8]
```

```
PAUSE 500
```

```
IF B6="1" THEN LOW BATCELL
IF B6="5" THEN HIGH BATCELL
```

```
IF W="H" THEN GOTO ARMADO
```

```
HIGH D
```

```
IF UNO=1 THEN GOTO TCL
```

```
GOTO MENSAJES
```

```
TCL:
```

```
PAUSE 500
```

```
IF UNO =1 THEN:HIGH bip :pause 3000:LOW bip: GOTO TCL1
```

```
GOTO MENSAJES
```

```
TCL1:
```

```
pause 500
```

Y=1

LCDOUT \$FE,1

PAUSE 200

LCDOUT \$FE,\$80,"INGRESE SU CLAVE"

PROG:

HIGH A

IF UNO=1 THEN T="1" :high bip:pause 500:low bip:GOSUB SIGNUM

IF DOS=1 THEN T="2" :high bip:pause 500:low bip:GOSUB SIGNUM

IF TRES=1 THEN T="3" :high bip:pause 500:low bip:GOSUB SIGNUM

LOW A

HIGH B

IF UNO=1 THEN T="4" :high bip:pause 500:low bip:GOSUB SIGNUM

IF DOS=1 THEN T="5" :high bip:pause 500:low bip:GOSUB SIGNUM

IF TRES=1 THEN T="6" :high bip:pause 500:low bip:GOSUB SIGNUM

LOW B

HIGH C

IF UNO=1 THEN T="7" :high bip:pause 500:low bip:GOSUB SIGNUM

IF DOS=1 THEN T="8" :high bip:pause 500:low bip:GOSUB SIGNUM

IF TRES=1 THEN T="9" :high bip:pause 500:low bip:GOSUB SIGNUM

LOW C

HIGH D

IF UNO=1 THEN :high bip:pause 500:low bip:GOTO ERRORCOD

IF DOS=1 THEN T="0" :high bip:pause 500:low bip:GOSUB SIGNUM

IF TRES=1 THEN :high bip:pause 500:low bip:GOTO COMPROBACION

LOW D

pause 10

GOTO PROG

SIGNUM:

IF Y=1 THEN T1=T:Y=Y+1 :GOTO PROG

IF Y=2 THEN T2=T:Y=Y+1 :GOTO PROG

IF Y=3 THEN T3=T:Y=Y+1 :GOTO PROG

IF Y=4 THEN T4=T:Y=Y+1 :GOTO PROG

IF Y=5 THEN goto ERRORCOD

ERRORCOD:

LCDOUT \$FE,1

PAUSE 200

LCDOUT \$FE,\$80,"CLAVE ERRONEA"

FOR X=0 TO 30

HIGH bip

pause 100

low bip

pause 100

next

LCDOUT \$FE,1

PAUSE 200

LCDOUT \$FE,\$82,"CONTROL-SMS"

goto INICIO

COMPROBACION:

if y=5 then goto COMPROBACION1

goto ERRORCOD

COMPROBACION1:

IF T1=CLP1 THEN PROG1

IF T1=CLA1 THEN CONTINUE1

GOTO ERRORCOD

CONTINUE1:

IF T2=CLA2 THEN CONTINUE2

GOTO ERRORCOD

CONTINUE2:

IF T3=CLA3 THEN CONTINUE3

GOTO ERRORCOD

```
CONTINUE3:
IF T4=CLA4 THEN CONTINUE4
GOTO ERRORCOD

PROG1:
IF T2=CLP2 THEN PROG2
GOTO ERRORCOD

PROG2:
IF T3=CLP3 THEN PROG3
GOTO ERRORCOD

PROG3:
IF T4=CLP4 THEN NUEVAINF
GOTO ERRORCOD

NUEVAINF:
serout portb.5,N2400,["B"]
LCDOUT $FE,1
PAUSE 200
LCDOUT $FE,$82,"CONECCION-PC"
SERIN portb.4,N2400,NUE
IF NUE="L" THEN GOTO TERMINADO
TERMINADO:
LCDOUT $FE,1
PAUSE 200
LCDOUT $FE,$82,"NUEVOS DATOS"
for x=0 to 10
high bip
pause 80
low bip
pause 80
next
pause 2000
```

goto EMPEZAR

ERRORZONA:

LCDOUT \$FE,1

PAUSE 200

LCDOUT \$FE,\$80,"ZONA ABIERTA"

for x=0 to 30

high bip

PAUSE 100

low bip

PAUSE 100

next

LCDOUT \$FE,1

PAUSE 200

LCDOUT \$FE,\$82,"CONTROL-SMS"

goto INICIO

CONTINUE4:

if ZONA1=1 THEN GOTO SIGZONA

GOTO ERRORZONA

SIGZONA:

if ZONA2=1 THEN GOTO SIGZONA1

GOTO ERRORZONA

SIGZONA1:

if ZONA3=1 THEN GOTO SIGZONA2

GOTO ERRORZONA

SIGZONA2:

high bip ;10 SEGUNDOS DE TIEMPO DE SALIDA

LCDOUT \$FE,1

PAUSE 200

LCDOUT \$FE,\$80,"TIEMPO DE SALIDA"

PAUSE 10000

LOW bip

if ZONA1=1 THEN GOTO CONTINUA

```
GOTO ERRORZONA
CONTINUA:
if ZONA2=1 THEN GOTO CONTINUA1
GOTO ERRORZONA
CONTINUA1:
if ZONA3=1 THEN GOTO CONTINUA2
GOTO ERRORZONA
CONTINUA2:
T1="0"
T2="0"
T3="0"
T4="0"
LCDOUT $FE,1
PAUSE 200
LCDOUT $FE,$80,"SISTEMA ARMADO"
GOTO ARMADO
```

```
ARMADO:
W="H"
IF ZONA2=0 THEN GOTO ALARMA
IF ZONA3=0 THEN GOTO ALARMA
IF ZONA1=0 THEN GOTO TIEMPO
GOTO MENSAJES
```

```
ALARMA:
W="G"
FOR X= 0 TO 5
HIGH bip
high RELE1
high RELE2
high RELE3
PAUSE 800
LOW bip
LOW RELE1
```


LOW RELE2

LOW RELE3

PAUSE 800

NEXT

PAUSE 1000

HIGH ledverde

serout2 PORTB.0,84,["AT",13]

PAUSE 1000

serout2

PORTB.0,84,["AT+CPMS=",34,"SM",34,44,34,"SM",34,44,34,"SM",34,1
3]

pause 2000

serout2

PORTB.0,84,["AT+CPMS=",34,"ME",34,44,34,"ME",34,44,34,"ME",34,1
3]

PAUSE 2000

serout2 PORTB.0,84,["AT+CMGF=1",13];MODO TEXTO

PAUSE 500

serout2

PORTB.0,84,["AT+CMGS=",34,NUM0,NUMA1,NUMB1,NUMC1,NUMD
1,NUME1,NUMF1,NUMG1,NUMH1,34,13]

PAUSE 300

SEROUT2 PORTB.0,84,["INGRESO NO AUTORIZADO",26]; ENVIAR

PAUSE 300

serout2

PORTB.0,84,["AT+CMGS=",34,NUM0,NUMA1,NUMB1,NUMC1,NUMD
1,NUME1,NUMF1,NUMG1,NUMH1,34,13]

PAUSE 300

SEROUT2 PORTB.0,84,["INGRESO NO AUTORIZADO",26]; ENVIAR

PAUSE 5000

serout2 PORTB.0,84,["AT",13]

PAUSE 1000

```
serout2
    PORTB.0,84,["AT+CPMS=",34,"SM",34,44,34,"SM",34,44,34,"SM",34,1
    3]
pause 2000
serout2
    PORTB.0,84,["AT+CPMS=",34,"ME",34,44,34,"ME",34,44,34,"ME",34,1
    3]
PAUSE 2000
serout2 PORTB.0,84,["AT+CMGF=1",13];MODO TEXTO
PAUSE 500
serout2
    PORTB.0,84,["AT+CMGS=",34,NUM0,NUMA2,NUMB2,NUMC2,NUMD
    2,NUME2,NUMF2,NUMG2,NUMH2,34,13]
PAUSE 300
SEROUT2 PORTB.0,84,["INGRESO NO AUTORIZADO",26]; ENVIAR
PAUSE 300
serout2
    PORTB.0,84,["AT+CMGS=",34,NUM0,NUMA2,NUMB2,NUMC2,NUMD
    2,NUME2,NUMF2,NUMG2,NUMH2,34,13]
PAUSE 300
SEROUT2 PORTB.0,84,["INGRESO NO AUTORIZADO",26]; ENVIAR
PAUSE 300
low ledverde

LCDOUT $FE,1
PAUSE 200
LCDOUT $FE,$82,"CONTROL-SMS"
GOTO INICIO

TIEMPO:
Y=1
LCDOUT $FE,1
PAUSE 200
LCDOUT $FE,$80,"INGRESE SU CLAVE"
```

```
for x=0 to 10
high bip
pause 50
low bip
pause 50
next
```

```
FOR X=1 TO 401 ;RETARDO PARA DESARMAR DE 5 SEGUNDOS
HIGH A
IF UNO=1 THEN T="1" :high bip:pause 500:low bip:GOSUB SIGNUM1
IF DOS=1 THEN T="2" :high bip:pause 500:low bip:GOSUB SIGNUM1
IF TRES=1 THEN T="3" :high bip:pause 500:low bip:GOSUB SIGNUM1
LOW A
HIGH B
IF UNO=1 THEN T="4" :high bip:pause 500:low bip:GOSUB SIGNUM1
IF DOS=1 THEN T="5" :high bip:pause 500:low bip:GOSUB SIGNUM1
IF TRES=1 THEN T="6" :high bip:pause 500:low bip:GOSUB SIGNUM1
LOW B
HIGH C
IF UNO=1 THEN T="7" :high bip:pause 500:low bip:GOSUB SIGNUM1
IF DOS=1 THEN T="8" :high bip:pause 500:low bip:GOSUB SIGNUM1
IF TRES=1 THEN T="9" :high bip:pause 500:low bip:GOSUB SIGNUM1
LOW C
HIGH D
IF UNO=1 THEN :high bip:pause 500:low bip:GOTO ALARMA
IF DOS=1 THEN T="0" :high bip:pause 500:low bip:GOSUB SIGNUM1
IF TRES=1 THEN :high bip:pause 500:low bip:GOTO DESARMAR
LOW D
IF ZONA2=0 THEN GOTO ALARMA
IF ZONA3=0 THEN GOTO ALARMA
PAUSE 25
NEXT
GOTO ALARMA
```

SIGNUM1:

```
IF Y=1 THEN T1=T:Y=Y+1 :RETURN
IF Y=2 THEN T2=T:Y=Y+1 :RETURN
IF Y=3 THEN T3=T:Y=Y+1 :RETURN
IF Y=4 THEN T4=T:Y=Y+1 :RETURN
IF Y=5 THEN goto ALARMA
```

DESARMAR:

```
IF T1=CLA1 THEN DES1
GOTO ALARMA
```

DES1:

```
IF T2=CLA2 THEN DES2
GOTO ALARMA
```

DES2:

```
IF T3=CLA3 THEN DES3
GOTO ALARMA
```

DES3:

```
IF T4=CLA4 THEN DES4
GOTO ALARMA
```

DES4:

```
W="G"
FOR X=0 TO 2
HIGH bip
pause 50
low bip
pause 50
NEXT
LCDOUT $FE,1
PAUSE 200
LCDOUT $FE,$82,"CONTROL-SMS"
```

```
goto INICIO
MENSAJES:
PAUSE 2000
serout portb.5,N2400,["A"]
PAUSE 100
serout2 PORTB.0,84,["AT+CNMI",13]
serIN2 PORTB.1,84,[A1]
serIN2 PORTB.1,84,[A2]
serIN2 PORTB.1,84,[A3]
serIN2 PORTB.1,84,[A4]
serIN2 PORTB.1,84,[A5]
serIN2 PORTB.1,84,[A6]
serIN2 PORTB.1,84,[A7]
serIN2 PORTB.1,84,[A8]
serIN2 PORTB.1,84,[A9]
serIN2 PORTB.1,84,[A0]
serIN2 PORTB.1,84,[B1]
serIN2 PORTB.1,84,[B2]
serIN2 PORTB.1,84,[B3]
serIN2 PORTB.1,84,[B4]
serIN2 PORTB.1,84,[B5]
serIN2 PORTB.1,84,[B6]
serIN2 PORTB.1,84,[B7]
serIN2 PORTB.1,84,[B8]
serIN2 PORTB.1,84,[B9]
serIN2 PORTB.1,84,[B0]
serIN2 PORTB.1,84,[C1]
serIN2 PORTB.1,84,[C2]
serIN2 PORTB.1,84,[C3]
serIN2 PORTB.1,84,[C4]
serIN2 PORTB.1,84,[C5] ;NUM1
serIN2 PORTB.1,84,[C6] ;NUM2
serIN2 PORTB.1,84,[C7] ;NUM3
serIN2 PORTB.1,84,[C8] ;NUM4
```

serIN2 PORTB.1,84,[C9] ;NUM5
serIN2 PORTB.1,84,[C0] ;NUM6
serIN2 PORTB.1,84,[D1] ;NUM7
serIN2 PORTB.1,84,[D2] ;NUM8
serIN2 PORTB.1,84,[D3]
serIN2 PORTB.1,84,[D4]
serIN2 PORTB.1,84,[D5]
serIN2 PORTB.1,84,[D6]
serIN2 PORTB.1,84,[D7]
serIN2 PORTB.1,84,[D8]
serIN2 PORTB.1,84,[D9]
serIN2 PORTB.1,84,[D0]
serIN2 PORTB.1,84,[E1]
serIN2 PORTB.1,84,[E2]
serIN2 PORTB.1,84,[E3]
serIN2 PORTB.1,84,[E4]
serIN2 PORTB.1,84,[E5]
serIN2 PORTB.1,84,[E6]
serIN2 PORTB.1,84,[E7]
serIN2 PORTB.1,84,[E8]
serIN2 PORTB.1,84,[E9]
serIN2 PORTB.1,84,[E0]
serIN2 PORTB.1,84,[F1]
serIN2 PORTB.1,84,[F2]
serIN2 PORTB.1,84,[F3]
serIN2 PORTB.1,84,[F4]
serIN2 PORTB.1,84,[F5]
serIN2 PORTB.1,84,[F6]
serIN2 PORTB.1,84,[F7]
serIN2 PORTB.1,84,[F8]
serIN2 PORTB.1,84,[F9]
serIN2 PORTB.1,84,[F0]
serIN2 PORTB.1,84,[G1]
serIN2 PORTB.1,84,[G2]

serIN2 PORTB.1,84,[G3]
serIN2 PORTB.1,84,[G4]
serIN2 PORTB.1,84,[G5]
serIN2 PORTB.1,84,[G6]
serIN2 PORTB.1,84,[G7]
serIN2 PORTB.1,84,[G8]
serIN2 PORTB.1,84,[G9]
serIN2 PORTB.1,84,[G0]
serIN2 PORTB.1,84,[H1]
serIN2 PORTB.1,84,[H2]
serIN2 PORTB.1,84,[H3]
serIN2 PORTB.1,84,[H4]
serIN2 PORTB.1,84,[H5]
serIN2 PORTB.1,84,[H6]
serIN2 PORTB.1,84,[H7]
serIN2 PORTB.1,84,[H8]
serIN2 PORTB.1,84,[H9]
serIN2 PORTB.1,84,[H0]
serIN2 PORTB.1,84,[I1]
serIN2 PORTB.1,84,[I2]
serIN2 PORTB.1,84,[I3]
serIN2 PORTB.1,84,[I4]
serIN2 PORTB.1,84,[I5]
serIN2 PORTB.1,84,[I6]
serIN2 PORTB.1,84,[I7]
serIN2 PORTB.1,84,[I8]
serIN2 PORTB.1,84,[I9]
serIN2 PORTB.1,84,[I0]
serIN2 PORTB.1,84,[J1]
serIN2 PORTB.1,84,[J2]
serIN2 PORTB.1,84,[J3]
serIN2 PORTB.1,84,[J4]
serIN2 PORTB.1,84,[J5]
serIN2 PORTB.1,84,[J6]

```
serIN2 PORTB.1,84,[J7]
serIN2 PORTB.1,84,[J8]
serIN2 PORTB.1,84,[J9]
serIN2 PORTB.1,84,[J0]
serIN2 PORTB.1,84,[K1]
serIN2 PORTB.1,84,[K2]
serIN2 PORTB.1,84,[K3] ;COD1
serIN2 PORTB.1,84,[K4] ;COD2
serIN2 PORTB.1,84,[K5] ;COD3
serIN2 PORTB.1,84,[K6]
serIN2 PORTB.1,84,[K7]
serIN2 PORTB.1,84,[K8]
serIN2 PORTB.1,84,[K9]
pause 100
```

```
IF C5=NUMA1 THEN PASO1
IF C5=NUMA2 THEN PASO1
GOTO INICIO
PASO1:
IF C6=NUMB1 THEN PASO2
IF C6=NUMB2 THEN PASO2
GOTO INICIO
PASO2:
IF C7=NUMC1 THEN PASO3
IF C7=NUMC2 THEN PASO3
GOTO INICIO
PASO3:
IF C8=NUMD1 THEN PASO4
IF C8=NUMD2 THEN PASO4
GOTO INICIO
PASO4:
IF C9=NUME1 THEN PASO5
IF C9=NUME2 THEN PASO5
GOTO INICIO
```



```
PASO5:
IF C0=NUMF1 THEN PASO6
IF C0=NUMF2 THEN PASO6
GOTO INICIO
PASO6:
IF D1=NUMG1 THEN PASO7
IF D1=NUMG2 THEN PASO7
GOTO INICIO
PASO7:
IF D2=NUMH1 THEN PASO8
IF D2=NUMH2 THEN PASO8
GOTO INICIO

PASO8:
IF K3="R" THEN RELES
GOTO INICIO

RELES:
IF K4="A" THEN ACTIVAR
IF K4="D" THEN DESACTIVAR
GOTO INICIO
ACTIVAR:
IF K5="1" THEN high RELE1:GOTO INICIO
IF K5="2" THEN high RELE2:GOTO INICIO
IF K5="3" THEN high RELE3:GOTO INICIO
GOTO INICIO
DESACTIVAR:
IF K5="1" THEN LOW RELE1:GOTO INICIO
IF K5="2" THEN LOW RELE2:GOTO INICIO
IF K5="3" THEN LOW RELE3:GOTO INICIO
GOTO INICIO
END
```

PROGRAMA DEL PIC 16F628A

```
define OSC 20
```

```
INCLUDE "modedefs.bas"
```

```
cmcon=7
```

```
ledamarillo VAR portb.5
```

```
RELE VAR portb.4
```

```
SMS VAR BYTE
```

```
SMS1 VAR BYTE
```

```
SMS2 VAR BYTE
```

```
G VAR BYTE
```

```
X VAR BYTE
```

```
NUMA1 VAR BYTE; numero 1 del celular
```

```
NUMB1 VAR BYTE
```

```
NUMC1 VAR BYTE
```

```
NUMD1 VAR BYTE
```

```
NUME1 VAR BYTE
```

```
NUMF1 VAR BYTE
```

```
NUMG1 VAR BYTE
```

```
NUMH1 VAR BYTE; numero 8 del celular
```

```
NUMA2 VAR BYTE; numero 1 del celular
```

```
NUMB2 VAR BYTE
```

```
NUMC2 VAR BYTE
```

```
NUMD2 VAR BYTE
```

```
NUME2 VAR BYTE
```

```
NUMF2 VAR BYTE
```

```
NUMG2 VAR BYTE
```

```
NUMH2 VAR BYTE; numero 8 del celular
```

```
CLA1 VAR BYTE
```

CLA2 VAR BYTE
CLA3 VAR BYTE
CLA4 VAR BYTE

EEPROM 0,["8","0","2","2","0","4","9","8"]
EEPROM 9,["8","9","0","4","3","9","6","2"]
EEPROM 18,["1","2","3","4"]

READ 0,NUMA1
READ 1,NUMB1
READ 2,NUMC1
READ 3,NUMD1
READ 4,NUME1
READ 5,NUMF1
READ 6,NUMG1
READ 7,NUMH1

READ 9,NUMA2
READ 10,NUMB2
READ 11,NUMC2
READ 12,NUMD2
READ 13,NUME2
READ 14,NUMF2
READ 15,NUMG2
READ 16,NUMH2

READ 18,CLA1 ; clave de ingreso a programacion
READ 19,CLA2
READ 20,CLA3
READ 21,CLA4

FOR X=0 TO 10
HIGH ledamarillo
pause 20
low ledamarillo

pause 20

next

PAUSE 1000

INICIO:

serin portb.0,N2400,SMS

IF SMS="C" THEN GOTO COMUNICACION

IF SMS="A" THEN GOTO ATENTO

IF SMS="B" THEN GOTO NUEVOINF

ATENTO:

HIGH ledamarillo

HIGH RELE

PAUSE 1500

LOW ledamarillo

low RELE

GOTO INICIO

COMUNICACION:

HIGH ledamarillo

READ 51,G

IF G="I" THEN GOTO ACTUALIZADO

IF G!="I" THEN GOTO IGUAL

IGUAL:

SEROUT portb.1,N2400,[NUMA1]

PAUSE 100

SEROUT portb.1,N2400,[NUMB1]

PAUSE 100

SEROUT portb.1,N2400,[NUMC1]

PAUSE 100

```
SEROUT portb.1,N2400,[NUMD1]
PAUSE 100
SEROUT portb.1,N2400,[NUME1]
PAUSE 100
SEROUT portb.1,N2400,[NUMF1]
PAUSE 100
SEROUT portb.1,N2400,[NUMG1]
PAUSE 100
SEROUT portb.1,N2400,[NUMH1]
PAUSE 100
SEROUT portb.1,N2400,[NUMA2]
PAUSE 100
SEROUT portb.1,N2400,[NUMB2]
PAUSE 100
SEROUT portb.1,N2400,[NUMC2]
PAUSE 100
SEROUT portb.1,N2400,[NUMD2]
PAUSE 100
SEROUT portb.1,N2400,[NUME2]
PAUSE 100
SEROUT portb.1,N2400,[NUMF2]
PAUSE 100
SEROUT portb.1,N2400,[NUMG2]
PAUSE 100
SEROUT portb.1,N2400,[NUMH2]
PAUSE 100
SEROUT portb.1,N2400,[CLA1]
PAUSE 100
SEROUT portb.1,N2400,[CLA2]
PAUSE 100
SEROUT portb.1,N2400,[CLA3]
PAUSE 100
SEROUT portb.1,N2400,[CLA4]
PAUSE 100
```

low ledamarillo

GOTO INICIO

ACTUALIZADO:

READ 28,NUMA1

READ 29,NUMB1

READ 30,NUMC1

READ 31,NUMD1

READ 32,NUME1

READ 33,NUMF1

READ 34,NUMG1

READ 35,NUMH1

READ 37,NUMA2

READ 38,NUMB2

READ 39,NUMC2

READ 40,NUMD2

READ 41,NUME2

READ 42,NUMF2

READ 43,NUMG2

READ 44,NUMH2

READ 46,CLA1

READ 47,CLA2

READ 48,CLA3

READ 49,CLA4

SEROUT portb.1,N2400,[NUMA1]

PAUSE 500

SEROUT portb.1,N2400,[NUMB1]

PAUSE 500

SEROUT portb.1,N2400,[NUMC1]

PAUSE 500

SEROUT portb.1,N2400,[NUMD1]

```
PAUSE 500
SEROUT portb.1,N2400,[NUME1]
PAUSE 500
SEROUT portb.1,N2400,[NUMF1]
PAUSE 500
SEROUT portb.1,N2400,[NUMG1]
PAUSE 500
SEROUT portb.1,N2400,[NUMH1]
PAUSE 500
SEROUT portb.1,N2400,[NUMA2]
PAUSE 500
SEROUT portb.1,N2400,[NUMB2]
PAUSE 500
SEROUT portb.1,N2400,[NUMC2]
PAUSE 500
SEROUT portb.1,N2400,[NUMD2]
PAUSE 500
SEROUT portb.1,N2400,[NUME2]
PAUSE 500
SEROUT portb.1,N2400,[NUMF2]
PAUSE 500
SEROUT portb.1,N2400,[NUMG2]
PAUSE 500
SEROUT portb.1,N2400,[NUMH2]
PAUSE 500
SEROUT portb.1,N2400,[CLA1]
PAUSE 500
SEROUT portb.1,N2400,[CLA2]
PAUSE 500
SEROUT portb.1,N2400,[CLA3]
PAUSE 500
SEROUT portb.1,N2400,[CLA4]
PAUSE 500
low ledamarillo
```

goto INICIO

NUEVOINF:

serin portb.2,N9600,SMS1

IF SMS1="O" THEN GOTO CELL1

IF SMS1="P" THEN GOTO CELL2

IF SMS1="R" THEN GOTO ALARMA

IF SMS1="Z" THEN GOTO FIN

goto NUEVOINF

CELL1:

HIGH ledamarillo

serin portb.2,N9600,NUMA1

IF NUMA1="T" THEN GOTO CELL1

serin portb.2,N9600,NUMB1

IF NUMB1="T" THEN GOTO CELL1

serin portb.2,N9600,NUMC1

IF NUMC1="T" THEN GOTO CELL1

serin portb.2,N9600,NUMD1

IF NUMD1="T" THEN GOTO CELL1

serin portb.2,N9600,NUME1

IF NUME1="T" THEN GOTO CELL1

serin portb.2,N9600,NUMF1

IF NUMF1="T" THEN GOTO CELL1

serin portb.2,N9600,NUMG1

IF NUMG1="T" THEN GOTO CELL1

serin portb.2,N9600,NUMH1

IF NUMH1="T" THEN GOTO CELL1

serin portb.2,N9600,SMS2

IF SMS2="V" THEN GOTO FINCELL1

IF SMS2="T" THEN GOTO CELL1

FINCELL1:

WRITE 28,NUMA1


```
WRITE 29,NUMB1
WRITE 30,NUMC1
WRITE 31,NUMD1
WRITE 32,NUME1
WRITE 33,NUMF1
WRITE 34,NUMG1
WRITE 35,NUMH1
serOUT portb.3,N9600,["9"]
LOW ledamarillo
GOTO NUEVOINF
```

CELL2:

```
high ledamarillo
serin portb.2,N9600,NUMA2
IF NUMA2="T" THEN GOTO CELL2
serin portb.2,N9600,NUMB2
IF NUMB2="T" THEN GOTO CELL2
serin portb.2,N9600,NUMC2
IF NUMC2="T" THEN GOTO CELL2
serin portb.2,N9600,NUMD2
IF NUMD2="T" THEN GOTO CELL2
serin portb.2,N9600,NUME2
IF NUME2="T" THEN GOTO CELL2
serin portb.2,N9600,NUMF2
IF NUMF2="T" THEN GOTO CELL2
serin portb.2,N9600,NUMG2
IF NUMG2="T" THEN GOTO CELL2
serin portb.2,N9600,NUMH2
IF NUMH2="T" THEN GOTO CELL2
serin portb.2,N9600,SMS2
IF SMS2="T" THEN GOTO CELL2
IF SMS2="V" THEN GOTO FINCELL2
```

FINCELL2:

```
WRITE 37,NUMA2
WRITE 38,NUMB2
WRITE 39,NUMC2
WRITE 40,NUMD2
WRITE 41,NUME2
WRITE 42,NUMF2
WRITE 43,NUMG2
WRITE 44,NUMH2
serOUT portb.3,N9600,["9"]
low ledamarillo
GOTO NUEVOINF
```

ALARMA:

```
high ledamarillo
serin portb.2,N9600,CLA1
IF CLA1="T" THEN GOTO ALARMA
serin portb.2,N9600,CLA2
IF CLA2="T" THEN GOTO ALARMA
serin portb.2,N9600,CLA3
IF CLA3="T" THEN GOTO ALARMA
serin portb.2,N9600,CLA4
IF CLA4="T" THEN GOTO ALARMA
serin portb.2,N9600,SMS2
IF SMS2="V" THEN GOTO FINALARMA
IF SMS2="T" THEN GOTO ALARMA
```

FINALARMA:

```
WRITE 46,CLA1
WRITE 47,CLA2
WRITE 48,CLA3
WRITE 49,CLA4
serOUT portb.3,N9600,["9"]
low ledamarillo
GOTO NUEVOINF
```

```
FIN:  
serOUT portb.3,N9600,["S"]  
SEROUT portb.1,N2400,["L"]  
WRITE 51,"I"  
GOTO INICIO  
END
```

ANEXO C

DATASHEET DEL

OPTOACOPLADOR

4N25

Data sheet 4n25

MOTOROLA SEMICONDUCTOR TECHNICAL DATA

Order this document
by 4N25/D



6-Pin DIP Optoisolators Transistor Output

The 4N25/A, 4N26, 4N27 and 4N28 devices consist of a gallium arsenide infrared emitting diode optically coupled to a monolithic silicon phototransistor detector.

- Most Economical Optoisolator Choice for Medium Speed, Switching Applications
- Meets or Exceeds All JEDEC Registered Specifications
- *To order devices that are tested and marked per VDE 0884 requirements, the suffix "V" must be included at end of part number. VDE 0884 is a test option.*

Applications

- General Purpose Switching Circuits
- Interfacing and coupling systems of different potentials and impedances
- I/O Interfacing
- Solid State Relays

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

Rating	Symbol	Value	Unit
--------	--------	-------	------

INPUT LED

Reverse Voltage	V_R	3	Volts
Forward Current — Continuous	I_F	60	mA
LED Power Dissipation @ $T_A = 25^\circ\text{C}$ with Negligible Power in Output Detector Derate above 25°C	P_D	120	mW
		1.41	mW/ $^\circ\text{C}$

OUTPUT TRANSISTOR

Collector-Emitter Voltage	V_{CEO}	30	Volts
Emitter-Collector Voltage	V_{ECO}	7	Volts
Collector-Base Voltage	V_{CBO}	70	Volts
Collector Current — Continuous	I_C	150	mA
Detector Power Dissipation @ $T_A = 25^\circ\text{C}$ with Negligible Power in Input LED Derate above 25°C	P_D	150	mW
		1.76	mW/ $^\circ\text{C}$

TOTAL DEVICE

Isolation Surge Voltage ⁽¹⁾ (Peak ac Voltage, 60 Hz, 1 sec Duration)	V_{ISO}	7500	Vac(pk)
Total Device Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	P_D	250 2.94	mW mW/ $^\circ\text{C}$
Ambient Operating Temperature Range ⁽²⁾	T_A	-55 to +100	$^\circ\text{C}$
Storage Temperature Range ⁽²⁾	T_{stg}	-55 to +150	$^\circ\text{C}$
Soldering Temperature (10 sec, 1/16" from case)	T_L	260	$^\circ\text{C}$

1. Isolation surge voltage is an internal device dielectric breakdown rating. For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.
2. Refer to Quality and Reliability Section in Opto Data Book for information on test conditions.

Preferred devices are Motorola recommended choices for future use and best overall value.
GlobalOptoisolator is a trademark of Motorola, Inc.

REV 5

© Motorola, Inc. 1995

4N25*
4N25A*
4N26*
[CTR = 20% Min]
4N27
4N28
[CTR = 10% Min]

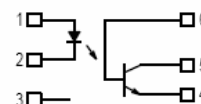
*Motorola Preferred Devices

STYLE 1 PLASTIC



STANDARD THRU HOLE
CASE 730A-04

SCHEMATIC



PIN 1. LED ANODE
2. LED CATHODE
3. N.C.
4. EMITTER
5. COLLECTOR
6. BASE



ANEXO D

DIAGRAMAS DE LAS

PLACAS QUE

COMPONEN EL

SISTEMA

DIAGRAMA DEL SISTEMA DE SEGURIDAD Y CONTROL DE LUCES.

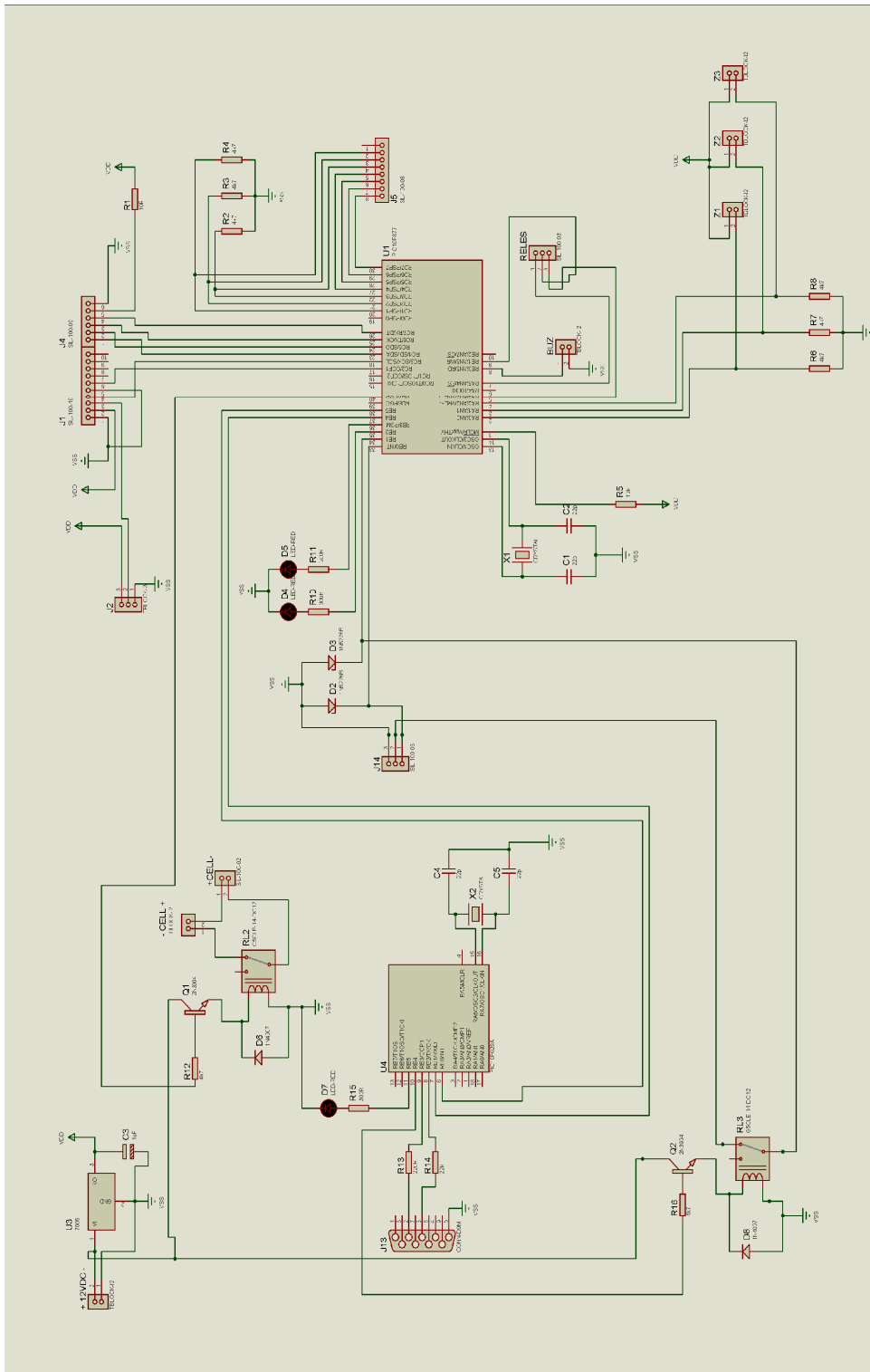


DIAGRAMA DE LA PLACA DE RELES.

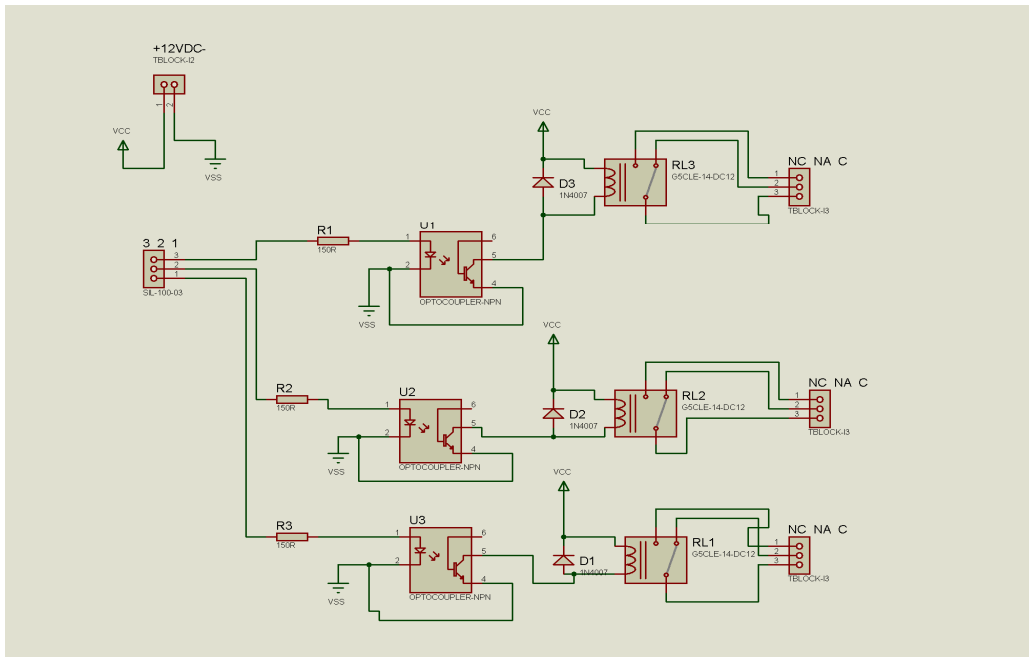
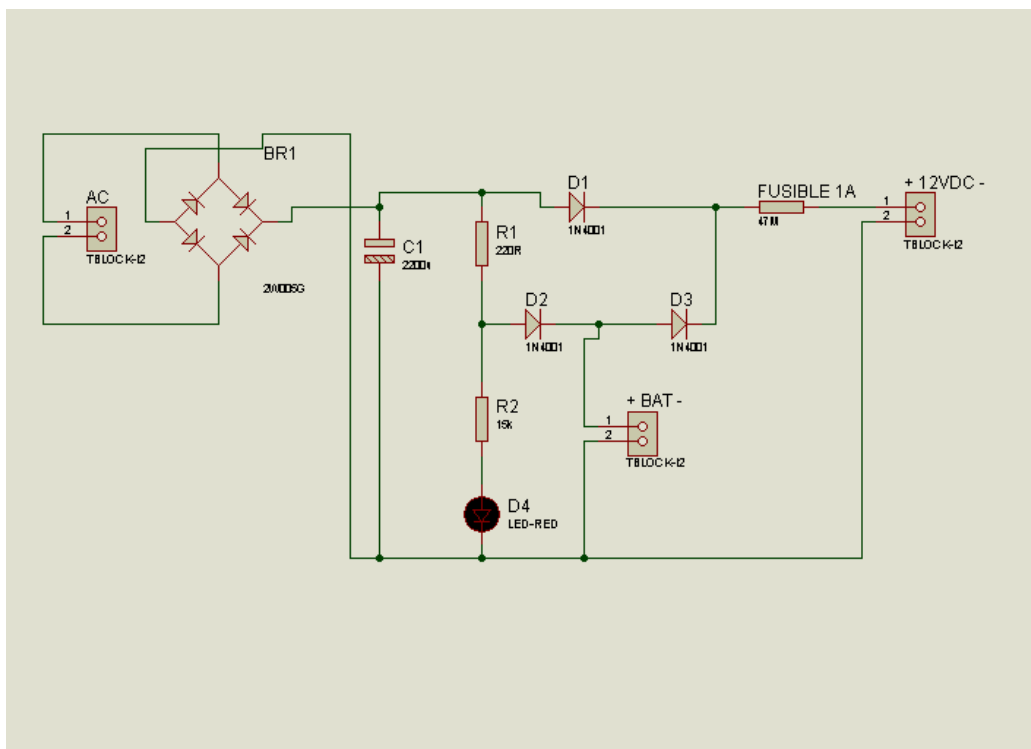


DIAGRAMA DE LA FUENTE



ANEXO E

MANUAL DE USUARIO.

MANUAL DE USUARIO.

El siguiente documento explica los pasos que el usuario debe seguir para manejar el equipo tanto el hardware como el software.

El primer paso es enchufar el equipo a una toma de 110 VAC, luego presionar el botón de encendido que es de color negro y se encuentra al costado derecho de la caja, al encender el equipo este emite varios pitidos durante 2 segundos, luego de esto, los leds verde y amarillo se encenderán al mismo tiempo, esto indica que entre los dos pics se esta intercambiando información.

Cuando los leds amarillo y verde se apagan la pantalla LCD muestra el siguiente mensaje:



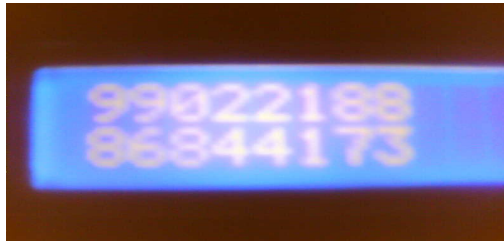
Mensaje que indica conectar el celular a la placa madre del sistema de alarma.

Se debe conectar el jack que tiene la placa madre al puerto de audio del celular, tal como se indica en la siguiente figura.



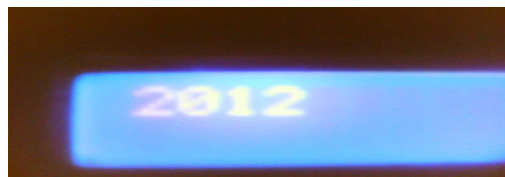
Jack para conectar en el celular

Ahora la Pantalla LCD muestra los números de celular que pueden recibir el mensaje de alarma, y también con un SMS pueden encender las luces de la casa. Esto se aprecia en la siguiente figura.



Números de celular de los usuarios 1 y 2.

Después el LCD indica la clave con la que se pone al sistema en modo alerta, esto se muestra en la siguiente imagen. El modo alerta se refiere a que el sistema está monitoreando la apertura de la puerta o ventanas; y si esto ocurre comienza el proceso de alarma.



CLAVE del sistema mostrada en el LCD

Ahora el equipo prueba el funcionamiento de los 3 relés, encendiendo y apagando las luces de la casa durante 1 segundo cada una. Después el led de color verde se enciende, esto indica que el equipo está enviando los SMS de saludo a los usuarios almacenados en memoria.



El led verde encendido indica que está enviando un SMS.

Con esto termina el proceso de arranque del equipo, de aquí en adelante el sistema se encuentra ejecutando procesos repetitivos, y esperando que el usuario ingrese un comando por el teclado. El LCD mostrara el siguiente mensaje.



Mensaje en la pantalla cuando ha terminado el proceso de arranque de todo el equipo.

Ahora estará el led rojo encendido constantemente, y el led amarillo se encuentra intermitente, cada vez que esta encendido indica que el pic y el celular están intercambiando información.

PASOS PARA ACTIVAR LAS LUCES DE LA CASA.

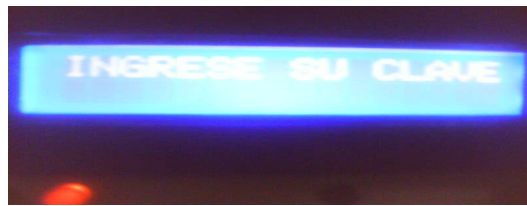
Si cualquiera de los dos usuarios registrados en la memoria del pic quiere encender alguna de las luces de la casa , debe enviar un SMS al numero “087032694”, con uno de los códigos que se muestran a continuación:

TEXTO DEL SMS	DESCRIPCION
RA1	ENCIENDE LAS LUCES DE LA VENTANA IZQUIERDA.
RA2	ENCIENDE LAS LUCES EN EL TECHO.
RA3	ENCIENDE LAS LUCES DE LA VENTANA DERECHA
RD1	APAGA LAS LUCES DE LA VENTANA IZQUIERDA.
RD2	APAGA LAS LUCES EN EL TECHO
RD3	APAGA LAS LUCES DE LA VENTANA DERECHA.

Comandos para prender o apagar las luces.

PASOS PARA ACTIVAR EL SISTEMA DE ALARMA.

Para activar la alarma, el usuario debe mantener pulsado la tecla asterisco en el teclado, hasta que el sistema emita un pitido y la pantalla LCD muestre el mensaje “ingrese su clave”, como se aprecia a continuación:



Mensaje en la pantalla para ingresar la clave de activación del sistema.

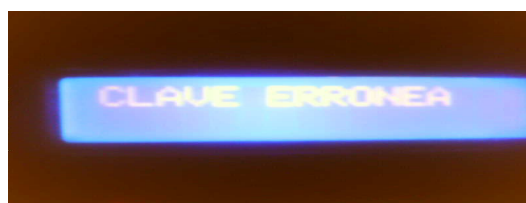
Ahora el usuario debe digitar la clave que apareció en la pantalla cuando el sistema se encendió, luego de digitar la clave se debe finalizar con la tecla numeral (#).

El sistema empieza a contar 10 segundos (ESTE TIEMPO NO ES MODIFICABLE), durante los cuales el usuario debe cerrar todos los accesos para que el sistema pueda armarse, caso contrario el LCD muestra el mensaje "ZONA ABIERTA", esto se puede apreciar a continuación:



Mensaje en la pantalla indicando que una de las zonas esta abierta,

Si todas las zonas están cerradas pero el usuario ingreso mal la clave entonces el LCD muestra el mensaje "CLAVE ERRONEA":



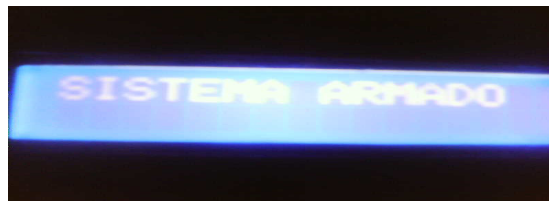
Mensaje en la pantalla indicando que la clave ingresada no es correcta.

Si todas las zonas están cerradas y el usuario ingreso la clave correcta entonces el sistema despliega el siguiente mensaje:



Mensaje en la pantalla indicando que en 10 segundos el sistema se pondrá en modo alerta.

Durante 10 segundos el usuario debe cerrar todos los accesos y pasado este tiempo el sistema se pone en modo alerta, el usuario observara en el LCD el siguiente mensaje:



Mensaje en la pantalla indicando que el sistema esta en modo alerta.

PROCESOS DEL SISTEMA CUANDO OCURRE UNA ALARMA.

La alarma ocurre cuando el sistema esta en modo alerta, y una de las ventanas es abierta, o, si la puerta es abierta y el usuario no ingresa correctamente la clave de desactivación.

Si esto ocurre el equipo enciende y apaga las luces de la casa prototipo durante 6 segundos, luego el led verde se enciende, esto indica que el sistema esta enviando los SMS de alerta a los usuarios que están almacenados en la memoria del PIC.

El SMS de alerta que llega a los usuarios tiene el siguiente texto “ **INGRESO NO AUTORIZADO**”.

PASOS PARA DESACTIVAR EL SISTEMA DE ALARMA.

Para desactivar el sistema, el usuario debe conocer como están distribuidas las zonas, esto se aprecia en la siguiente tabla:

NUMERO DE ZONA	DESCRIPCION
ZONA 1	PUERTA (TEMPORIZADA)
ZONA 2	VENTANA IZQUIERDA
ZONA 3	VENTANA DERECHA

Tabla con el número de zona y su correspondiente acceso.

La zona numero 1 es la única que esta temporizada, si alguna de las ventanas es abierta, la alarma se dispara, sin darle al usuario la posibilidad de ingresar la clave para desarma el sistema.

Para la desactivación de la alarma, el usuario debe abrir la zona 1 y esperar a que el sistema emita un pitido, luego de esto, el usuario tiene 10 segundos para ingresar la misma clave con la que armo el sistema, seguido de la tecla numeral (#); si la clave ingresada es correcta, el sistema vuelve al modo normal, pero si es errónea la alarma se dispara.

NOTA: Cuando el sistema esta en modo alerta el usuario puede controlar las luces con los SMS normalmente.

PASOS PARA CAMBIAR LOS USUARIOS ALMACENADOS EN LA MEMORIA DEL PIC Y LA CLAVE DE ACTIVACION DEL SISTEMA.

Para realizar este proceso primero el usuario debe mantener presionado el botón asterisco en el teclado del equipo, hasta que la pantalla muestre el mensaje "INGRESE SU CLAVE". Para poner al sistema en modo programación el usuario debe ingresar la clave "5555", seguido de la tecla numeral, esta clave no es modificable. Enseguida la pantalla del equipo muestra el siguiente mensaje:



Mensaje que indica modo programación desde el PC.

Ahora se debe conectar el cable serie al sistema de alarma, como se indica a continuación:



Cable serie conectado al puerto de datos del sistema de alarma.

Como este sistema utiliza uno de los puertos serie del computador para la comunicación, es necesario conocer los puertos COM disponibles en el PC.

Para verificar los puertos COM que están disponibles en el computador, hay que ingresar al panel de control, luego a herramientas administrativas, y luego a administrador de dispositivos. En esa ventana se puede observar cuantos puertos serie tiene el computador y su correspondiente número, en la siguiente figura se aprecia lo mencionado.



Pantalla del administrador de dispositivos.

Después de conocer los puertos COM que tiene nuestro computador, verificamos físicamente cual de ellos esta libre (generalmente el COM 1 permanece libre en los computadores), y conectamos el cable serie al puerto COM que se encuentra disponible en el PC.

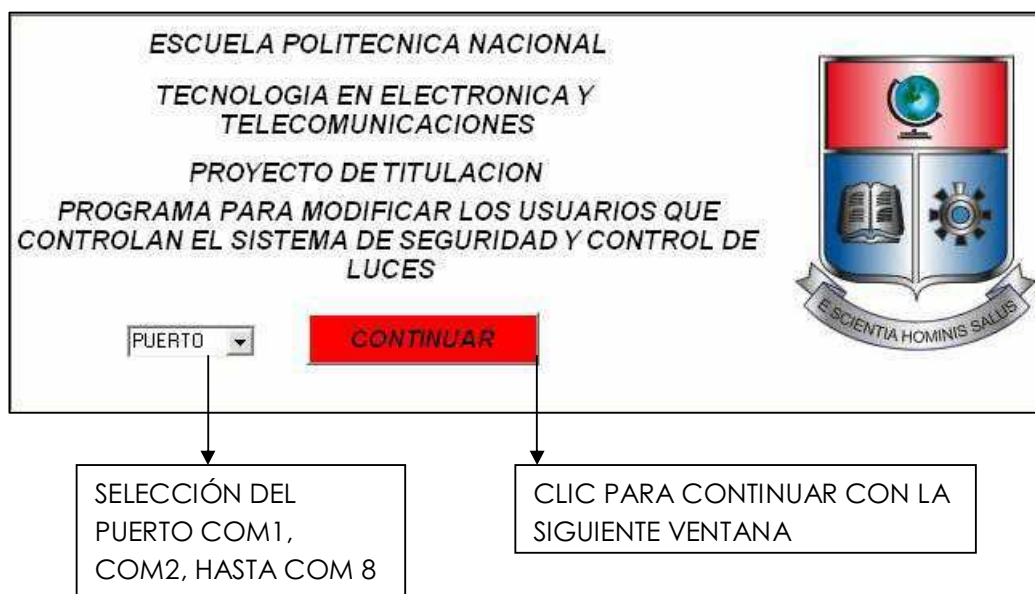
Ahora pasamos a la parte del software:

Para ejecutar el programa se debe ubicar el icono que se muestra a continuación, y dar doble clic sobre este:



Icono del programa CONTROL GSM

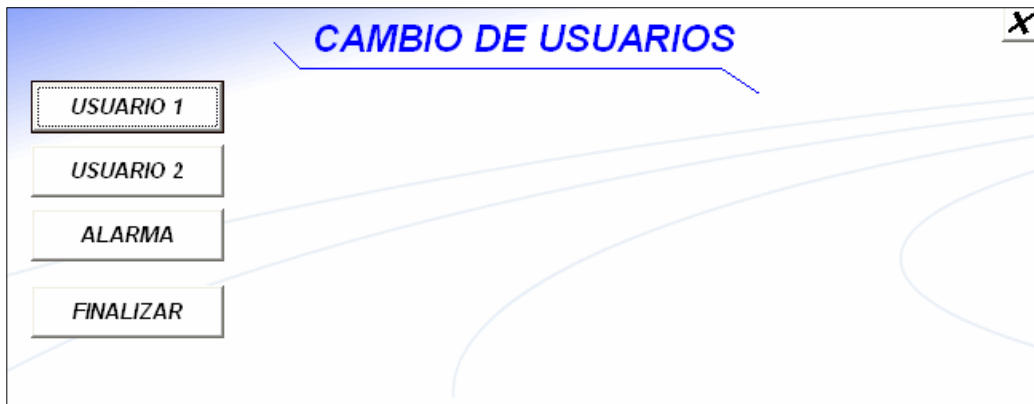
Al dar doble clic en el icono se despliega la pantalla principal del programa:



Pantalla principal del software de control.

En esta ventana primero se debe seleccionar el numero del PUERTO COM, que verificamos en el administrador de dispositivos..

Después que seleccionamos el puerto, damos un clic en el botón continuar, y se despliega siguiente ventana:



Pantalla para el cambio de usuarios y clave del sistema de control GSM.

Para cambiar el número de celular de cualquiera de los usuarios, dar un clic en el botón usuario 1 o usuario 2, y aparecen las siguientes opciones.



Pantalla para ingresar un nuevo número de celular.

Para ingresar un nuevo número, se lo hace a través del teclado que se encuentra en el centro de la pantalla, al dar un clic en cada número, éstos se irán colocando en los casilleros de la derecha, para que el usuario los pueda visualizar.

Nota: El ingreso del nuevo número de celular, se lo debe hacer omitiendo el número cero con el que comienzan todos los números de celular, ya que este número, está colocado en el primer casillero que contiene el nuevo número.

En la siguiente imagen, se aprecia un número de celular ingresado a través del teclado.



Pantalla con un nuevo número escrito.

Si al ingresar el nuevo número hay una equivocación, con un clic en el botón de borrar se puede volver a reingresar el número.

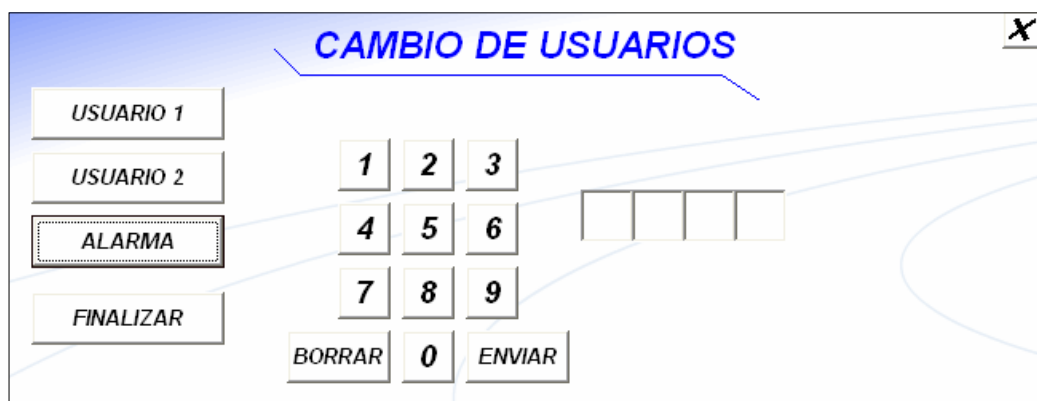
Una vez que el número de celular se encuentra en los casilleros correctamente, dar un clic en el botón enviar, para que el equipo pueda tomar la información, y almacenarla en la memoria EEPROM.

Si todo está correcto en la pantalla aparecerá una ventana con el mensaje "NUMERO GUARDADO".

De esta manera se procede para cambiar cualquiera de los dos usuarios que pueden controlar el sistema de alarma.

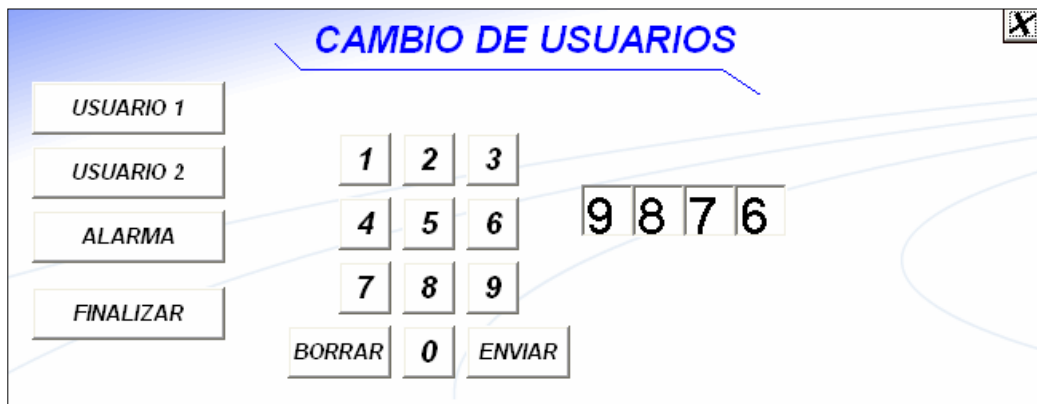
Ahora, para cambiar la clave con la que se pone al sistema en modo alerta, se procede de manera similar.

Primero dar un clic en el botón de alarma, y la ventana mostrará el teclado para ingresar la nueva clave, junto con 4 casilleros al lado derecho.



Ventana donde se puede cambiar la clave del sistema.

Ingresar la nueva clave número por número con un clic en el teclado de la pantalla.



Ventana con un nuevo código escrito.

Para enviar el código escrito en la ventana, hacia el equipo conectado, dar un clic en el botón enviar, entonces aparecerá una ventana con el mensaje “numero guardado”.

El botón finalizar se utiliza para desconectar el equipo del computador, una vez terminada la programación. Al dar un clic sobre este botón, aparece una ventana con el mensaje “AHORA PUEDE DESCONECTAR EL EQUIPO”; seguido a esto, el equipo se reinicia, y en el proceso de arranque el LCD mostrara los nuevos datos que fueron almacenados.

BIBLIOGRAFÍA

- 1 Tecnología GSM, “ Enric Forner”.
- 2 Microcontroladores ,”Carlos A. Reyes”
- 3 CORRALES,SANTIAGO,ELECTRONICA PRACTICA CON MICROCONTROLADORES PIC(PROGRAMACION CON LENGUAJE BASIC,SANTIAGO CORRALES, 2006
- 4 Mobile Menssanging Technologies and Service SMS, EMS and MMS, France, Gwenael Le Bodic, JOHN WILEY & SONS, LTD.
- 5 MSc. Soraya Sinche,Folleto de Comunicaciones Inalámbricas.
- 6 AJAY R. Mishra/ Advanced Cellular Network Planning and Optimization.
- 7 Flores Christian,Rivadeneira José Luis , Tesis Escuela Politécnica Nacional, Diseño e Implementación de un Sistema Microprocesado Para Adquisición de Datos en Forma Remota de un Medidor Digital de Consumo de Energía Eléctrica Tipo Industrial Mediante Telefonía Celular ,2006.
- 8 Estrada Juan, Reinos Diego, Tesis Escuela Politécnica Nacional, Diseño E Implementación De Un Prototipo De Monitoreo Remoto De Contaminación Ambiental Utilizando Tecnología GSM, ,2009.

WEB GRAFÍA.

- 1 www.Microchip.com
- 2 www.mecanique.co.uk
- 3 <http://picbasic.com/>
- 4 http://es.wikipedia.org/w/index.php?title=Reed_switch
- 5 www.mobilesms.com
- 6 www.frino.com
- 7 www.datasheet.com

- 8 www.carlosvolt.com.ar
- 9 www.edudevices.com.ar
- 10 www.microcontroladores.com
- 11 http://www.todobaterias.com/optima_az.html
- 12 <http://www.motorola.com>
- 13 <http://www.gsmserver.com>
- 14 www.X-ROBOTICS.com
- 15 www.labcenter.com
- 16 www.ico-Ecuador.com
- 17 www.wmlclub.com/articulos/fundamentosgsm.html
- 18 www.amena.com/presentacion/particulares/telefonos/gprs/
- 19 www.zonabot.com
- 20 www.wikipedia.com/codigosascii.html