

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

### **IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE SEGURIDAD INALÁMBRICO PARA VIVIENDAS, VÍA TELEFONÍA CELULAR CON TECNOLOGÍA GSM**

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

**ALLAICA PACHACAMA CARMEN PILAR**  
carmen.pilar\_88@hotmail.com

**CAJAMARCA ACHIG VERÓNICA FERNANDA**  
fer\_rony@hotmail.com

**DIRECTOR: ING. COSTALES ALCÍVAR**  
eduardo.costales@epn.edu.ec

**Quito, Abril del 2013**

## **DECLARACIÓN**

Nosotras, Allaica Pachacama Carmen Pilar y Cajamarca Achig Verónica Fernanda, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros 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.

---

**Allaica Pachacama Carmen  
Pilar**

---

**Cajamarca Achig Verónica  
Fernanda**

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por Allaica Pachacama Carmen Pilar y Cajamarca Achig Verónica Fernanda, bajo mi supervisión.

---

**Ing. ALCÍVAR COSTALES**  
**DIRECTOR DE PROYECTO**

## **AGRADECIMIENTO**

Agradezco primeramente a DIOS, a mi madre y mis hermanos por todo el apoyo incondicional tanto en lo moral como en lo económico, que me ayudo a alcanzar uno de mis objetivos en mi vida profesional.

Además, quiero dar un profundo agradecimiento a mi novio Daniel que es una persona muy especial en mi vida, quien me brinda ayuda, apoyo y cariño en los buenos y malos en los momentos en toda la carrera universitaria.

Por otro lado agradezco a nuestro director de tesis el ING. Alcívar Costales, por la disposición y la ayuda que nos brindó durante la culminación de nuestro proyecto.

*Carmen Allaica*

## DEDICATORIA

Este proyecto es la culminación de una de mis metas de mi vida profesional, por lo que dedico en especial a mi madre, quien me infundió el valor de la honestidad, verdad y el trabajo, y que con su sacrificio supo darme la mejor herencia en la vida que es una carrera universitaria.

A mi padre que desde el cielo, está mirando culminada una etapa de mi vida y siempre está cuidándome y guiando mi camino.

*Carmen Allaica*

## **AGRADECIMIENTO**

Quiero plasmar aquí el más profundo agradecimiento a Dios por haberme dado salud y fuerza para concluir esta etapa de mi vida.

A mi madre que siempre con su amor y sus palabras de aliento ha sabido guiar mi camino y apoyarme incondicionalmente, y, a mi familia por brindarme calor de hogar.

Un agradecimiento especial al Ing. Alcívar Costales, Director de Tesis de Grado, quien en el desarrollo de este proyecto nos brindó su colaboración y conocimientos.

*Verónica Cajamarca*

## DEDICATORIA

A mi querida madre, quien ha sido mi mayor fortaleza para seguir adelante, gracias por tu sacrificio, por el ejemplo de honestidad e integridad que me has infundido a lo largo de mi vida, a mi padre que aunque ya no esté aquí conmigo se que siempre me da sus bendiciones desde el cielo.

A Dios por haberme dado siempre sabiduría de tomar las mejores decisiones, y a la luz que está en mi corazón por haberme dado sostén, felicidad y apoyo absoluto para culminar este proyecto.

*Verónica Cajamarca*

## Índice

ÍNDICE .....	VIII
ÍNDICE DE FIGURAS .....	XII
ÍNDICE DE TABLAS .....	XIV
PRESENTACIÓN .....	XV
CAPITULO I.....	1
<b>1 FUNDAMENTO TEÓRICO .....</b>	<b>1</b>
1.1 MICROCONTROLADORES .....	1
1.1.1 INTRODUCCIÓN .....	1
1.1.2 DEFINICIÓN.....	1
1.1.3 FUNCIONAMIENTO .....	1
1.1.4 APLICACIONES.....	2
1.1.5 ARQUITECTURA BÁSICA DE UN MICROCONTROLADOR .....	3
1.1.6 UNIDAD DE PROCESAMIENTO CENTRAL (CPU).....	3
1.1.7 MEMORIA .....	4
1.1.7.1 Memoria ROM <sup>4</sup> .....	4
1.1.7.2 Memoria RAM <sup>4</sup> .....	5
1.1.8 LÍNEAS DE ENTRADA Y SALIDA <sup>4</sup> .....	5
1.2 RECURSOS ESPECIALES.....	6
1.2.1 TEMPORIZADORES O "TIMERS" <sup>5</sup> .....	6
1.2.2 PERRO GUARDIÁN O "WATCHDOG" <sup>5</sup> .....	7
1.2.3 PROTECCIÓN ANTE FALLO DE ALIMENTACIÓN O "BROWNOUT" <sup>5</sup> .....	7
1.2.4 ESTADO DE REPOSO O DE BAJO CONSUMO <sup>5</sup> .....	7
1.2.5 CONVERTOR A/D (CAD) <sup>5</sup> .....	8
1.2.6 CONVERTOR D/A (CDA) <sup>5</sup> .....	8
1.2.7 COMPARADOR ANALÓGICO <sup>5</sup> .....	8
1.2.8 MODULADOR DE ANCHURA DE IMPULSOS O PWM <sup>5</sup> .....	8
1.2.9 PUERTOS DE E/S DIGITALES <sup>5</sup> .....	9
1.2.10 PUERTOS DE COMUNICACIÓN <sup>5</sup> .....	9
1.3 SENSORES.....	10
1.3.1 Sensor inalámbrico de movimiento (PIR).....	11
1.3.1.1 Características <sup>7</sup> .....	11
1.3.1.2 Funcionamiento.....	12
1.3.2 Sensor inalámbrico de apertura de puerta .....	12
1.3.2.1 Características.....	13
1.3.2.2 Funcionamiento.....	13
1.4 OTROS DISPOSITIVOS .....	13
1.4.1 LCD.....	14
1.4.1.1 Características <sup>10</sup> .....	14
1.4.1.2 Funcionamiento <sup>10</sup> .....	14
1.4.2 TECLADO .....	15
1.4.3 RECEPTOR RLP434 .....	17
1.4.4 MAX 232 .....	18



1.4.4.1	Funcionamiento.....	19
1.4.4.2	Niveles de voltaje.....	20
1.4.5	<i>MÓDULO GSM ZTE</i> .....	20
1.4.6	<i>Circuito Integrado ULN2003A</i> .....	22
1.4.6.1	Características generales <sup>14</sup> .....	22
1.4.6.2	Descripción del ULN2003A <sup>14</sup> .....	22
1.5	SISTEMAS DE SEGURIDAD.....	24
1.5.1	<i>COMPONENTES DE UN SISTEMA DE SEGURIDAD</i> .....	25
1.5.2	<i>ALARMA</i> .....	26
1.5.3	<i>TIPOS DE ALARMAS</i> .....	26
1.5.3.1	Con conexión a la central.....	27
1.5.3.2	Sin conexión a la central.....	27
1.5.4	<i>SISTEMA DE ALARMA</i> .....	27
1.5.4.1	Partes principales de un sistema de alarma.....	27
1.5.5	<i>SISTEMA DE ALARMA INALÁMBRICO CON MÓDULO GSM</i> .....	28
1.6	TECNOLOGÍA GSM.....	29
1.6.1	<i>TECNOLOGÍA GSM COMO UN ESTÁNDAR</i> .....	30
1.6.2	<i>ARQUITECTURA DE LA RED GSM</i> .....	30
1.6.3	<i>BANDAS DE FRECUENCIAS DE OPERACIÓN GSM</i> .....	32
1.6.4	<i>FRECUENCIAS QUE OPERAN MOVISTAR, CLARO Y ALEGRO</i> .....	32
1.6.5	<i>MENSAJES DE TEXTO SMS</i> .....	33
1.6.5.1	Arquitectura de la red SMS <sup>18</sup> .....	34
1.6.5.2	Pasos de envío y recepción de SMS <sup>18</sup> .....	35
1.6.5.3	Aplicaciones <sup>18</sup> .....	36
1.6.6	<i>TECNOLOGÍA GSM APLICADA A LOS SISTEMAS DE SEGURIDAD</i> .....	37
	<b>CAPÍTULO II.....</b>	<b>38</b>
<b>2</b>	<b>DISEÑO DE LOS CIRCUITOS DEL SISTEMA DE CONTROL DE SEGURIDAD.....</b>	<b>38</b>
2.1	DESCRIPCIÓN DE LOS DISPOSITIVOS.....	38
2.1.1	<i>MICROCONTROLADOR ATMEGA 164P</i> .....	38
2.1.1.1	Características generales <sup>20</sup> .....	38
2.1.2	<i>PUERTOS DEL ATMEGA164P<sup>20</sup></i> .....	40
2.2	DESARROLLO DEL CIRCUITO DEL SISTEMA DE SEGURIDAD.....	42
2.2.1	<i>DESCRIPCIÓN DE LAS ETAPAS DEL HARDWARE</i> .....	42
2.2.1.1	Fase de control (A).....	44
2.2.1.2	Bloque de la Fuente Regulada de Voltaje (B).....	47
2.2.1.3	Teclado de 5 dígitos para controlar el sistema de seguridad y LCD para visualizar (C).....	48
2.2.1.4	Control de Iluminación para la vivienda (C).....	51
2.2.1.5	Detección de movimiento en el domicilio (D).....	51
2.3	PLACA IMPRESA DEL SISTEMA DE SEGURIDAD.....	52
2.4	DISEÑO DE LA MAQUETA PARA SIMULACIÓN DE UN DOMICILIO.....	54
	<b>CAPITULO III.....</b>	<b>55</b>
<b>3</b>	<b>DESARROLLO DEL SOFTWARE PARA EL SISTEMA DE SEGURIDAD.....</b>	<b>55</b>
3.1	LENGUAJE DE PROGRAMACIÓN BASCOM AVR.....	55
3.1.1	<i>INICIO<sup>21</sup></i> .....	56

3.1.2	COMPILADOR <sup>21</sup> .....	56
3.1.3	SIMULADOR <sup>21</sup> .....	57
3.1.4	EMULADOR SERIAL <sup>21</sup> .....	58
3.2	COMANDOS AT.....	59
3.2.1	NOTACIÓN DE LOS COMANDOS AT <sup>22</sup> .....	60
3.2.2	COMANDOS AT MÁS UTILIZADOS <sup>22</sup> .....	61
3.2.3	COMANDOS GENERALES <sup>22</sup> .....	62
3.2.4	COMANDOS DEL SERVICIO DE RED <sup>22</sup> .....	62
3.2.5	COMANDOS DE SEGURIDAD <sup>22</sup> .....	62
3.2.6	COMANDOS PARA LA AGENDA DE TELÉFONOS <sup>22</sup> .....	62
3.2.7	COMANDOS PARA SMS <sup>22</sup> .....	62
3.3	GRABADOR DEL MICROCONTROLADOR.....	63
3.3.1	PROGISP 172.....	63
3.4	PRINCIPALES SENTENCIAS UTILIZADAS EN EL PROYECTO .....	65
3.4.1	INSTRUCCIONES INICIALES <sup>21</sup> .....	65
3.4.1.1	\$regfile .....	65
3.4.1.2	\$crystal .....	66
3.4.1.3	\$Baud .....	66
3.4.2	CONFIGURACIONES INICIALES <sup>21</sup> .....	66
3.4.2.1	Config .....	66
3.4.2.2	DDRx, PORTx.....	66
3.4.2.3	Alias .....	67
3.4.3	TIPO DE DATOS <sup>21</sup> .....	67
3.4.3.1	On Urxc.....	68
3.4.3.2	Dim .....	68
3.4.4	ADMINISTRACIÓN DE BITS <sup>21</sup> .....	68
3.4.4.1	Toggle .....	68
3.4.5	ADMINISTRACIÓN DE STRINGS <sup>21</sup> .....	69
3.4.5.1	INSTR .....	69
3.4.5.2	Mid.....	69
3.4.6	INSTRUCCIONES DE USO GENERAL <sup>21</sup> .....	69
3.4.6.1	Wait .....	69
3.4.6.2	Incr .....	70
3.4.6.3	Estructura de las sentencias .....	70
3.4.6.4	Do – Loop .....	70
3.4.6.5	For-Next .....	70
3.4.6.6	If-Then-Else .....	71
3.4.6.7	Gosub .....	71
3.4.7	CONFIGURACIÓN MEDIANTE COMANDOS <sup>21</sup> .....	71
3.4.7.1	Config Com1, Com2.....	71
3.4.7.2	Config Lcdpin .....	72
3.5	PROGRAMA PRINCIPAL PARA EL SISTEMA DE SEGURIDAD.....	72
3.6	DIAGRAMA DE FLUJO DEL SISTEMA DE SEGURIDAD .....	92
3.7	PROGRAMA “EAGLE” PARA EL DISEÑO DEL CIRCUITO.....	94
3.7.1	EDITOR DE ESQUEMAS .....	94
3.7.2	EDITOR DE LÍNEAS DE CONEXIÓN.....	95
3.7.3	EDITOR DE LIBRERÍA .....	96

<b>CAPITULO IV.....</b>	<b>97</b>
<b>4 PRUEBAS Y RESULTADOS.....</b>	<b>97</b>
4.1 PRUEBAS.....	97
4.1.1 PRUEBA 1: ACTIVACIÓN Y DESACTIVACIÓN DE LA ALARMA.....	97
4.1.2 PRUEBA 2: INGRESO DE 5 DÍGITOS PARA CONTROL DEL SISTEMA.....	98
4.1.3 PRUEBA 3: DETECTANDO MOVIMIENTO.....	99
4.1.4 PRUEBA 4: CONTROL DE ILUMINACIÓN.....	102
4.2 RESULTADOS:.....	103
4.2.1 RESULTADO DE CAMBIO DE CLAVE:.....	104
4.2.2 DURANTE ROBO:.....	105
4.2.3 ACTIVACIÓN Y DESACTIVACIÓN DEL SISTEMA DE SEGURIDAD:.....	106
4.2.4 CONTROL DE ILUMINACIÓN:.....	108
4.2.5 BOTÓN DE PÁNICO:.....	109
<b>CAPITULO V.....</b>	<b>112</b>
<b>5 CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>112</b>
5.1 CONCLUSIONES.....	112
5.2 RECOMENDACIONES.....	114
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>115</b>
<b>GLOSARIO.....</b>	<b>117</b>
<b>ANEXOS.....</b>	<b>120</b>
<b>ANEXO A.....</b>	<b>121</b>
<b>ANEXO B.....</b>	<b>147</b>
<b>ANEXO C.....</b>	<b>149</b>
<b>ANEXO D.....</b>	<b>155</b>
<b>ANEXO E.....</b>	<b>164</b>

## ÍNDICE DE FIGURAS

Figura 1. 1: Estructura básica de un microcontrolador .....	3
Figura 1. 2: Arquitectura Neumann .....	3
Figura 1. 3: Arquitectura Harvard .....	4
Figura 1. 4: Ejemplo de alarmas de seguridad para viviendas.....	10
Figura 1. 5: Sensor de movimiento PIR.....	11
Figura 1. 6: Sensor magnético .....	12
Figura 1. 7: Bus de datos LCD de 4 bits.....	14
Figura 1. 8: Diagrama de conexión tipo bus de 4 bits .....	16
Figura 1. 9: Receptor RLP 434.....	17
Figura 1. 10: Circuito MAX232 .....	19
Figura 1. 11: Ejemplo de Módulo GSM .....	21
Figura 1. 12: Circuito Integrado ULN2003A .....	22
Figura 1. 13: Distribución de pines del ULN2003A <sup>14</sup> .....	23
Figura 1. 14: Características técnicas del ULN2003A <sup>14</sup> .....	23
Figura 1. 15: Ejemplo de sistema de seguridad. ....	25
Figura 1. 16: Ejemplo de los componentes de un Sistema de Seguridad .....	26
Figura 1. 17: Partes principales de un sistema de alarma.....	28
Figura 1. 18: Tecnología GSM. ....	30
Figura 1. 19: Arquitectura GSM.....	31
Figura 1. 20: Bandas de frecuencias asignadas a GSM. ....	32
Figura 1. 21: Elemento de la red SMS .....	34
Figura 2. 1: Ejemplo Atmega164P.....	38
Figura 2. 2: Distribución de pines del Atmega164P.....	40
Figura 2. 3: Diagrama de Bloques del sistema de seguridad .....	43
Figura 2. 4: Distribución de pines del micro controlador ATmega164P .....	45
Figura 2. 5: Diagrama Circuital de la sirena .....	46
Figura 2. 6: Configuración del Microcontrolador ATmega 164P .....	47
Figura 2. 7: Diagrama circuital fuente de poder.....	48
Figura 2. 8: Diagrama circuital del LCD y teclado .....	50
Figura 2. 9: Diagrama circuital del control de Iluminación .....	51
Figura 2. 10: Alcance del sensor PIR .....	52
Figura 2. 11: Diseño de la placa del sistema de seguridad en EAGLE v5.0 .....	53
Figura 2. 12: Diseño de la placa en baquelita .....	53
Figura 2. 13: Plano en AutoCAD de un domicilio .....	54
Figura 3. 1: Diagrama de flujo de programación estructurada .....	55
Figura 3. 2: Área de trabajo del BASCOM AVR .....	56
Figura 3. 3: Cuadro donde se muestra guardar nuevo archivo .....	57

Figura 3. 4: Cuadro de confirmación de compilación .....	57
Figura 3. 5: Icono del simulador del programa .....	57
Figura 3. 6: Cuadro de simulación del BASCOM AVR .....	58
Figura 3. 7: Icono emulador serial del BASCOM AVR .....	58
Figura 3. 8: Emulador de comunicación serial.....	58
Figura 3. 9: Diagrama circuital del programador USB y Progisp 172 .....	64
Figura 3. 10: Pantalla de grabación del programador USB .....	65
Figura 3. 11: Ejemplo de la condición de repetición FOR-NEXT <sup>20</sup> .....	70
Figura 3. 12: Ejemplo de la ccondición Lógica IF-ELSE.....	71
Figura 3. 13: Ejemplo del entorno del editor de esquemas .....	95
Figura 3. 14: Ejemplo de entorno de líneas de conexión .....	95
Figura 3. 15: Panel de control .....	96
Figura 4. 1: Simulación del circuito del sistema de seguridad en protoboard.....	97
Figura 4. 2: Inicialización del módulo GSM para el circuito .....	98
Figura 4. 3: Inicialización del sistema de seguridad al activarlo .....	99
Figura 4. 4: Conexión del ATmega164P y el módulo Rx.....	100
Figura 4. 5: Conexión serial para visualizar señales en la PC.....	100
Figura 4. 6: Diferentes señales emitidas por el aire .....	101
Figura 4. 7: Señales emitida por el sensor inalámbrico.....	101
Figura 4. 8: Ubicación de los jumpers para mandar señales F0.....	102
Figura 4. 9: Prueba encendido de Foco 1 .....	102
Figura 4. 10: Prueba apagado Foco 1 .....	103
Figura 4. 11: Cerebro del sistema de seguridad.....	103
Figura 4. 12: Cerebro del sistema de seguridad con la conexión del teclado .....	103
Figura 4. 13: Nueva clave de 5 dígitos .....	104
Figura 4. 14: Mensaje antes de ingresar 5 dígitos nuevos. ....	104
Figura 4. 15: Mensaje luego de ingresar los 5 dígitos nuevos.....	105
Figura 4. 16: Ubicación del teclado en la maqueta.....	105
Figura 4. 17: Mensaje enviado al usuario en caso de robo .....	106
Figura 4. 18: Sirena activada en caso de robo .....	106
Figura 4. 19: Activación del Sistema de seguridad.....	107
Figura 4. 20: Ingresar 5 dígitos por teclado de forma incorrecta .....	107
Figura 4. 21: Mensaje cuando el sistema está listo.....	108
Figura 4. 22: Verificación del encendido del foco 1 .....	108
Figura 4. 23: Verificación del encendido del foco 2 .....	109
Figura 4. 24: Verificación del encendido del foco 3 .....	109
Figura 4. 25: Verificación el apagado de iluminación .....	109
Figura 4. 26: Ubicación del botón de pánico para la simulación.....	110
Figura 4. 27: Mensaje en el LCD al pulsar el botón de pánico .....	110

## ÍNDICE DE TABLAS

Tabla 1. 1: Descripción de pines tipo bus de 4 bits <sup>10</sup> .....	15
Tabla 1. 2: Características principales del TLP434 <sup>12</sup> .....	18
Tabla 1. 3: Funciones de cada puerto del MAX232.....	19
Tabla 1. 4: Niveles de voltaje del RS232.....	20
Tabla 1. 5: Interfaces para el ULN2003A .....	23
Tabla 2. 1: Tabla de funciones del sistema d seguridad .....	43
Tabla 2. 2: Distribución de pines y puertos para el teclado .....	49
Tabla 2. 3: Distribución de pines del LCD .....	49
Tabla 3. 1: Tipos de variables .....	68
Tabla 3. 2: Archivos de Eagle.....	94
Tabla 4. 1: Tabla de Comandos del teclado para el sistema de seguridad .....	99
Tabla 4. 2: Mensajes de texto para el módulo GSM.....	99
Tabla 4. 3: Resultados generales .....	110
Tabla 4. 4: Mensajes de texto utilizados para el Sistema de Seguridad .....	111
Tabla 4. 5: Mensajes de texto que el módulo GSM envía al usuario.....	111

## PRESENTACIÓN

El desarrollo de la electrónica digital y la comunicación móvil, hace posible a la reducción del hardware que se utiliza, tomando como una alternativa altamente eficiente, rentable y segura, la implementación de dispositivos programables como son los microcontroladores los cuales hacen posible la ejecución de aplicaciones de una forma más sencilla.

Otros de los adelantos tecnológicos que han permitido lograr este objetivo ha sido la posibilidad de comunicación entre dispositivos electrónicos programables con módulos de comunicaciones de telefonía móvil la combinación de estos dispositivos es posible activar o desactivar diferentes funciones controladas por un microcontrolador. Además por su facilidad de uso se le puede dar una cantidad de aplicaciones.

El desarrollo de este prototipo de sistema de seguridad va dirigido a la utilización didáctica para mostrar el funcionamiento, comunicación y las aplicaciones que se puede lograr con la integración de la electrónica digital y módulos de comunicación móviles.

## **CAPITULO I**

### **1 FUNDAMENTO TEÓRICO**

#### **1.1 MICROCONTROLADORES**

##### **1.1.1 INTRODUCCIÓN<sup>1</sup>**

Los microcontroladores son computadores digitales integrados en un chip que cuentan con un microprocesador o unidad de procesamiento central (CPU), una memoria para almacenar el programa, una memoria para almacenar datos y puertos de entrada salida. A diferencia de los microprocesadores de propósito general, como los que se usan en los computadores PC, los microcontroladores son unidades autosuficientes y más económicas.

##### **1.1.2 DEFINICIÓN**

Los microcontroladores son computadores digitales integrados en un chip; en su memoria solo reside un programa destinado a gobernar una aplicación determinada, sus líneas de entrada/salida soportan el conexionado de los sensores y actuadores del dispositivo a controlar y todos los recursos complementarios disponibles, tienen como única finalidad atender sus requerimientos. Una vez programado y configurado el microcontrolador solamente sirve para gobernar la tarea asignada.

##### **1.1.3 FUNCIONAMIENTO**

El funcionamiento de los microcontroladores está determinado por el programa almacenado en su memoria a diferencia de los microprocesadores que necesitan de una memoria y de interfaces para recibir y enviar datos. Es decir un microprocesador es el corazón de una computadora.

---

<sup>1</sup>[http://web.ing.puc.cl/~mtorrest/downloads/pic/tutorial\\_pic.pdf](http://web.ing.puc.cl/~mtorrest/downloads/pic/tutorial_pic.pdf)



En un diseño tradicional, una tarea tan sencilla como modificar un tiempo de respuesta o agregar alguna pequeña función no contemplada en el diseño original, obliga a construir un nuevo circuito impreso, agregar o quitar integrados, entre otros; los proyectos que están contruidos en torno a un microcontrolador, en pocos minutos pueden realizar un cambio en el software.

La mayoría de los microcontroladores actuales pueden reprogramarse repetidas veces, quizás esta sea una de sus mayores ventajas al realizar modificaciones en el comportamiento de nuestro proyecto.

#### **1.1.4 APLICACIONES<sup>2</sup>**

Cada vez existen más productos que incorporan un microcontrolador con el fin de aumentar sustancialmente sus prestaciones, reducir su tamaño y coste, mejorar su fiabilidad y disminuir el consumo.

Algunos fabricantes de microcontroladores superan el millón de unidades de un modelo determinado producidas en una semana. Este dato puede dar una idea de la masiva utilización de estos componentes

A continuación se presenta las distintas aplicaciones que se tiene:

- ❖ Aparatos electrodomésticos: microondas, refrigeradores, hornos, TV, reproductores y grabadores de CD y DVD, equipos de sonido.
- ❖ Equipo de cómputo: impresoras, módems, unidades de disco, ratones, teclados.
- ❖ Automóviles: mando de sistemas del automóvil.
- ❖ Domótica: sistemas antirrobo, vigilancia y alarma en los edificios para optimizar el rendimiento de ascensores, calefacción, entre otros.
- ❖ Instrumentación: Equipos de medida.

---

<sup>2</sup>[http://galia.fc.uaslp.mx/~cantocar/microcontroladores/SLIDES\\_8051\\_PDF/3\\_APLICA.PDF](http://galia.fc.uaslp.mx/~cantocar/microcontroladores/SLIDES_8051_PDF/3_APLICA.PDF)

### 1.1.5 ARQUITECTURA BÁSICA DE UN MICROCONTROLADOR

Un microcontrolador es un dispositivo complejo, formado por otros más sencillos. A continuación se analizan los más importantes.

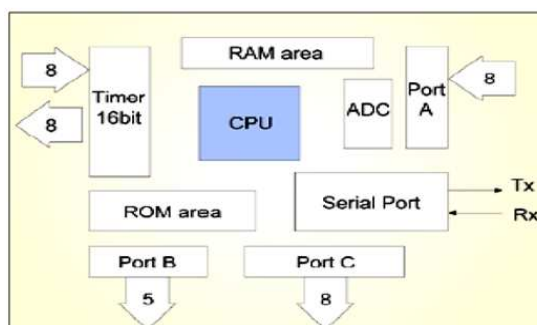


Figura 1. 1: Estructura básica de un microcontrolador

### 1.1.6 UNIDAD DE PROCESAMIENTO CENTRAL (CPU)

Controla el funcionamiento de todos los elementos del sistema, en función de las instrucciones del programa.

**Von Neumann:** Es una arquitectura básica que se caracteriza por tener una sola memoria que se comunica a través de un sistema de buses único.



Figura 1. 2: Arquitectura Neumann

**Arquitectura Harvard:** Tiene dos tipos de memorias de anchura diferente por lo cual es posible acceder a ambas memorias al mismo tiempo aumentando la velocidad.



**Figura 1. 3:** Arquitectura Harvard<sup>3</sup>

El procesador de los modernos microcontroladores responde a la arquitectura RISC (Computadores de juego de instrucciones reducido), que se identifica por poseer un repertorio de instrucciones máquina pequeño y simple, de forma que la mayor parte de las instrucciones se ejecutan en un ciclo de instrucción.

Otra aportación frecuente que aumenta el rendimiento del computador es el fomento del paralelismo implícito, que consiste en la segmentación del procesador (pipe-line), descomponiéndolo en etapas para poder procesar una instrucción diferente en cada una de ellas y trabajar con varias a la vez.

### 1.1.7 MEMORIA<sup>4</sup>

En los microcontroladores la memoria de instrucciones y datos está integrada en el propio chip. Una parte debe ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación. Otra parte de memoria será tipo RAM, volátil, y se destina a guardar las variables y los datos.

#### 1.1.7.1 Memoria ROM<sup>4</sup>

- ❖ Son memorias de solo lectura.
- ❖ No pierden la información aunque se interrumpa la alimentación.
- ❖ Las memorias ROM se programan en fábrica y no es posible modificar su contenido

<sup>3</sup><http://perso.wanadoo.es/pictob/microcr.htm>

<sup>4</sup>[http://www.dte.us.es/tec\\_inf/itig/etc2/material/ETC2\\_0304\\_memorias.pdf](http://www.dte.us.es/tec_inf/itig/etc2/material/ETC2_0304_memorias.pdf)

#### 1.1.7.1.1 Tipos de Memorias ROM<sup>4</sup>

- ❖ **PROM:** Contiene una matriz de fusibles que es programable por el usuario una sola vez.
- ❖ **EPROM:** Permiten su reprogramación después de someter al chip a un proceso de borrado por radiación ultravioleta.
- ❖ **EEPROM:** Permite su reprogramación eléctrica.
- ❖ **EEPROM FLASH:** Son más rápidas por permitir borrado y escritura por bloques aunque tienen el inconveniente del desgaste (10 000 – 90 000 borrados)

#### 1.1.7.2 Memoria RAM<sup>4</sup>

Son memorias de lectura y escritura.

##### 1.1.7.2.1 Tipos de Memorias RAM<sup>4</sup>

- ❖ **SRAM:** Celda básica basada en un biestable (6 transistores) y la información no se pierde mientras no exista alimentación.
- ❖ **DRAM:** celda básica: 1 transistor + condensador, necesita refresco, mayor capacidad en comparación con la SRAM y menor velocidad de acceso.

#### 1.1.8 LÍNEAS DE ENTRADA Y SALIDA<sup>4</sup>

A excepción de dos pines destinadas a recibir la alimentación, otros dos pines para el cristal de cuarzo regulador de la frecuencia de trabajo y una más para el Reset, los restantes pines de un microcontrolador sirven para soportar su comunicación con los periféricos externos que controla.

Según los controladores de periféricos que posea cada modelo de microcontrolador, las líneas de E/S se destinan a proporcionar el soporte a las señales de entrada, salida y control.

## 1.2 RECURSOS ESPECIALES<sup>5</sup>

Cada fabricante oferta numerosas versiones de una arquitectura básica de microcontrolador. En algunas amplía las capacidades de las memorias, en otras incorpora nuevos recursos, en otras reduce las prestaciones al mínimo para aplicaciones muy simples, etc. La labor del diseñador es encontrar el modelo mínimo que satisfaga todos los requerimientos de su aplicación. De esta forma, minimizará el coste, el hardware y el software.

Los principales recursos específicos que incorporan los microcontroladores son:

- ❖ Temporizadores o "Timers".
- ❖ Perro guardián o "Watchdog".
- ❖ Protección ante fallo de alimentación o "Brownout".
- ❖ Estado de reposo o de bajo consumo.
- ❖ Conversor A/D.
- ❖ Conversor D/A.
- ❖ Comparador analógico.
- ❖ Modulador de anchura de impulsos o PWM.
- ❖ Puertos de E/S digitales.
- ❖ Puertos de comunicación.

### 1.2.1 TEMPORIZADORES O "TIMERS"<sup>5</sup>

Se emplean para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores).

Para la medida de tiempos se carga un registro con el valor adecuado y a continuación dicho valor se va incrementando o decrementando al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a 0, momento en el que se produce un aviso.

---

<sup>5</sup> <http://bibdigital.epn.edu.ec/bitstream/15000/2349/1/CD-3097.pdf>

Cuando se desean contar acontecimientos que se materializan por cambios de nivel o flancos en alguna de las patitas del microcontrolador, el mencionado registro se va incrementando o decrementando al ritmo de dichos impulsos.

### **1.2.2 PERRO GUARDIÁN O "WATCHDOG"<sup>5</sup>**

Cuando el computador personal se bloquea por un fallo del software u otra causa, se pulsa el botón del reset y se reinicializa el sistema. Pero un microcontrolador funciona sin el control de un supervisor y de forma continuada las 24 horas del día.

El perro guardián consiste en un temporizador que, cuando se desborda y pasa por 0, provoca un reset automáticamente en el sistema.

Se debe diseñar el programa de trabajo que controla la tarea de forma que refresque o inicialice al perro guardián antes de que provoque el reset. Si falla el programa o se bloquea, no se refrescará al perro guardián y, al completar su temporización, "ladrará y ladrará" hasta provocar el reset.

### **1.2.3 PROTECCIÓN ANTE FALLO DE ALIMENTACIÓN O "BROWNOUT"<sup>5</sup>**

Se trata de un circuito que resetea al microcontrolador cuando el voltaje de alimentación (VDD) es inferior a un voltaje mínimo ("Brownout"). Mientras el voltaje de alimentación sea inferior al de brownout el dispositivo se mantiene reseteado, comenzando a funcionar normalmente cuando sobrepasa dicho valor.

### **1.2.4 ESTADO DE REPOSO O DE BAJO CONSUMO<sup>5</sup>**

Son abundantes las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, a que se produzca algún acontecimiento externo que le ponga de nuevo en funcionamiento. Para ahorrar energía, (factor clave en los aparatos portátiles), los microcontroladores disponen de una instrucción especial (SLEEP en los PIC), que les pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos. En dicho estado se detiene el reloj principal y se "congelan" sus circuitos asociados, quedando sumido en un profundo

"sueño" el microcontrolador. Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo.

### **1.2.5 CONVERTOR A/D (CAD)<sup>5</sup>**

Los microcontroladores que incorporan un Conversor A/D (Analógico/Digital) pueden procesar señales analógicas, tan abundantes en las aplicaciones. Suelen disponer de un multiplexor que permite aplicar a la entrada del CAD diversas señales analógicas desde las patitas del circuito integrado.

### **1.2.6 CONVERTOR D/A (CDA)<sup>5</sup>**

Transforma los datos digitales obtenidos del procesamiento del computador en su correspondiente señal analógica que saca al exterior por una de las patitas de la cápsula. Existen muchos efectores que trabajan con señales analógicas.

### **1.2.7 COMPARADOR ANALÓGICO<sup>5</sup>**

Algunos modelos de microcontroladores disponen internamente de un Amplificador Operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por una de las patitas de la cápsula. La salida del comparador proporciona un nivel lógico 1 ó 0 según una señal sea mayor o menor que la otra.

También hay modelos de microcontroladores con un módulo de tensión de referencia que proporciona diversas tensiones de referencia que se pueden aplicar en los comparadores.

### **1.2.8 MODULADOR DE ANCHURA DE IMPULSOS O PWM<sup>5</sup>**

Son circuitos que proporcionan en su salida impulsos de anchura variable, que se ofrecen al exterior a través de las patitas del encapsulado.

### 1.2.9 PUERTOS DE E/S DIGITALES<sup>5</sup>

Todos los microcontroladores destinan algunas de sus patitas a soportar líneas de E/S (entrada / salidas) digitales. Por lo general, estas líneas se agrupan de ocho en ocho formando Puertos.

Las líneas digitales de los Puertos pueden configurarse como entrada o como salida cargando un 1 ó un 0 en el bit correspondiente de un registro destinado a su configuración.

### 1.2.10 PUERTOS DE COMUNICACIÓN<sup>5</sup>

Con objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos.

Algunos modelos disponen de recursos que permiten directamente esta tarea, entre los que destacan:

- ❖ **UART:** adaptador de comunicación serie asíncrona.
- ❖ **USART:** adaptador de comunicación serie síncrona y asíncrona

Puerta paralela esclava para poder conectarse con los buses de otros microprocesadores.

USB (Universal Serial Bus), que es un moderno bus serie para los PC.

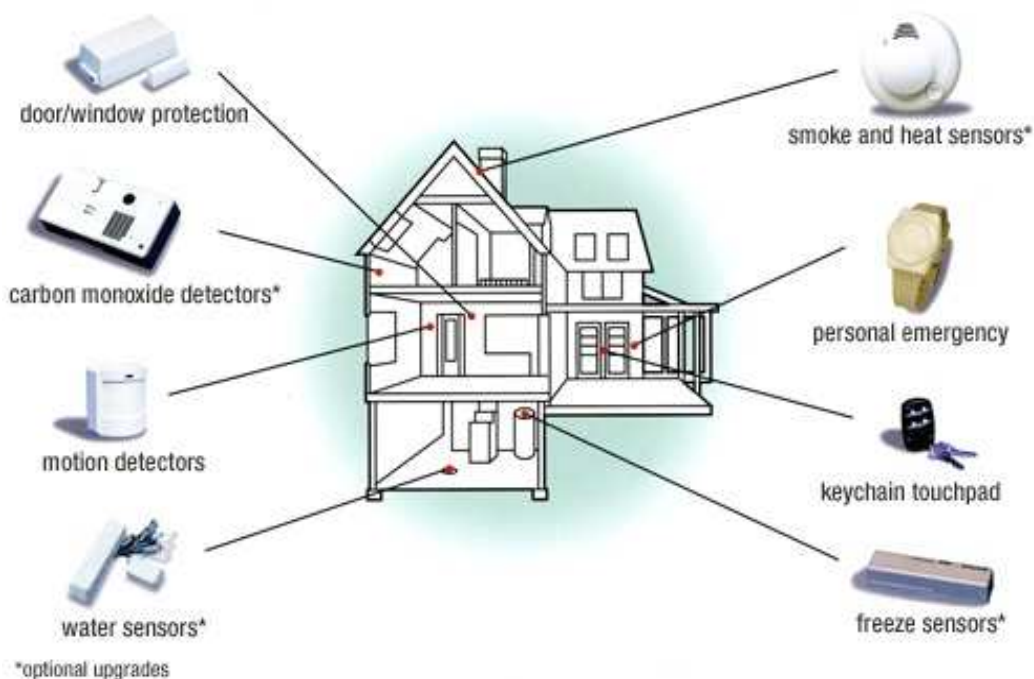
Bus I2C, que es un interfaz serie de dos hilos desarrollado por Philips.

Uno de los factores que más importancia tiene a la hora de seleccionar un microcontrolador entre todos los demás es el soporte tanto software como hardware de que dispone. Un buen conjunto de herramientas de desarrollo puede ser decisivo en la elección, ya que pueden suponer una ayuda inestimable en el desarrollo del proyecto.



### 1.3 SENSORES

Los sensores son componentes de hardware que pueden proporcionar al equipo información acerca de la ubicación del mismo, sus alrededores y otros datos.



**Figura 1. 4:** Ejemplo de alarmas de seguridad para viviendas.

Existen dos tipos de sensores:

- ❖ Sensores integrados en el equipo
- ❖ Sensores que están conectados al equipo mediante una conexión con cable o inalámbrica.

Dentro de este tema solo hablaremos de los sensores con una conexión inalámbrica que son los que utilizaremos para este proyecto.

### 1.3.1 Sensor inalámbrico de movimiento (PIR)<sup>6</sup>



**Figura 1. 5:** Sensor de movimiento PIR.

Es un sistema más flexible que los tradicionales porque los nodos se despliegan sin necesidad de que exista una infraestructura previa, tienen capacidad de auto organización, la comunicación es inalámbrica. Esto les da una fortaleza, es por ejemplo en la industria, dónde existen sitios muy difíciles para cablear por sus condiciones ambientales o por imposibilidad de acceso.

Se puede dotar a la red de cierta inteligencia según los requerimientos, sin necesidad de que el usuario conozca la tecnología o la esté operando en ese momento. El caso que más muestran los medios es el de domótica o casas inteligentes, que activan sistemas de riego, alarma, abren cortinas, encienden la calefacción, todo en forma autónoma del usuario, detectando simplemente condiciones ambientales.

Los sensores inalámbricos tienen varias aplicaciones en el monitoreo de animales y todas las variables ambientales, además es bastante versátil en cámaras y sistemas de seguridad para viviendas, por lo que se toma como un recurso importante dentro de nuestro diseño por su facilidad de instalación y comodidad del usuario que se muestra en la Figura N°1.5.

#### 1.3.1.1 Características<sup>7</sup>

---

<sup>6</sup> [http://convergenciadigital.com/eshop/product\\_info.php?products\\_id=831](http://convergenciadigital.com/eshop/product_info.php?products_id=831)

- ❖ Controlado por un microprocesador con un algoritmo avanzado procesador de señal digital adaptable.
- ❖ Montado en superficies y esquinas remueve la protección manipulada.
- ❖ Señales de supervisión aleatorias para verificar la integridad del sistema está libre de problemas.
- ❖ Detección de batería baja.
- ❖ Luz blanca superior de rechazo de ruido.
- ❖ Rango de detección: 12 metros en un ángulo de 90°.
- ❖ Rechazo superior de interferencias de radio de hasta 300 m en frecuencias de 100KHz a 1GHz.
- ❖ Mecanismo automático de ahorro de energía.
- ❖ A prueba de insectos.
- ❖ LED: Indicador de modo de prueba y falla.

### 1.3.1.2 Funcionamiento

El principio de funcionamiento de los sensores de movimiento PIR, se basa en que el dispositivo detecta la diferencia entre la temperatura ambiente y la temperatura del cuerpo humano. Cuando la temperatura del cuerpo humano difiere con la del ambiente, la radiación generada por un intruso hace que se active el sensor el cual envía un aviso de alerta al microcontrolador, y este ejecuta la instrucción de encendido del sistema de seguridad.

### 1.3.2 Sensor inalámbrico de apertura de puerta



**Figura 1. 6:** Sensor magnético

Los sensores magnéticos inalámbricos de apertura de puerta que se muestra en la Figura N°1.6, son aquellos que detectan una variación en el campo magnético en respuesta a la variación de alguna magnitud física.<sup>7</sup>

### 1.3.2.1 Características

Se caracterizan principalmente por ser dispositivos de estado sólido, no tener partes móviles, compatibilidad con otros circuitos analógicos y digitales.<sup>8</sup>

- ❖ Energía: 12VDC (batería de 12V 23A)
- ❖ Frecuencia de trabajo: 433MHz, 315MHz
- ❖ Gap para ser accionada: > 10m m
- ❖ Corriente espera: Menos que 1uA
- ❖ Corriente que transmite: Menos que 20Ma
- ❖ Energía que transmite: 15mW
- ❖ Peso bruto: 55g

### 1.3.2.2 Funcionamiento

Un sensor magnético inalámbrico de apertura de puerta consta de dos partes: la parte emisora de la señal (donde está la antena) y la parte pequeña que es un magneto. Su funcionamiento consiste en adherir el emisor en la puerta y el magneto va pegado en el marco de la puerta. Cuando el sistema de seguridad se encuentra activado y las dos partes del sensor están en contacto el circuito está cerrado, pero cuando se detecta una separación de las partes del sensor, el led del emisor se enciende, enviando un aviso de alerta al microcontrolador y este a la vez enciende el sistema de alerta.

## 1.4 OTROS DISPOSITIVOS

---

<sup>7</sup> <http://www.uaeh.edu.mx/docencia/Tesis/icbi/licenciatura/documentos/Sensores%20magneticos.pdf>

<sup>8</sup> [http://es.made-in-china.com/co\\_youngcheer/product\\_Wireless-Door-Magnetic-Sensor-YCF200MC-\\_hshhennnn.html](http://es.made-in-china.com/co_youngcheer/product_Wireless-Door-Magnetic-Sensor-YCF200MC-_hshhennnn.html)

### 1.4.1 LCD<sup>9</sup>

La pantalla de cristal líquido o LCD (Liquid Crystal Display) es un dispositivo Controlado de visualización gráfico para la presentación de caracteres, símbolos o incluso dibujos (en algunos modelos), es este caso dispone de 2 filas de 16 caracteres cada una y cada carácter dispone de una matriz de 5x7 puntos (píxeles), aunque los hay de otro número de filas y caracteres. Este dispositivo está gobernado internamente por un microcontrolador y regula todos los parámetros de presentación, este modelo es el más comúnmente usado y esta información se basará en el manejo de este u otro LCD compatible.

#### 1.4.1.1 Características<sup>10</sup>

- ❖ Pantalla de caracteres ASCII.
- ❖ Desplazamiento de los caracteres hacia la izquierda o la derecha.
- ❖ Proporciona la dirección de la posición absoluta o relativa del carácter.
- ❖ Memoria de 40 caracteres por línea de pantalla, movimiento del cursor y cambio de su aspecto
- ❖ Permite que el usuario pueda programar 8 caracteres.
- ❖ Conexión a un procesador usando un interfaz de 4 u 8 bits.

#### 1.4.1.2 Funcionamiento<sup>10</sup>

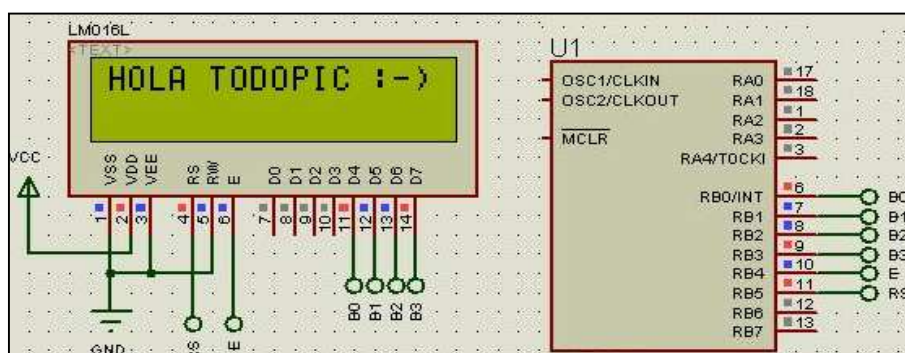


Figura 1. 7: Bus de datos LCD de 4 bits

<sup>9</sup> <http://es.scribd.com/doc/44252680/LCD-16X2>

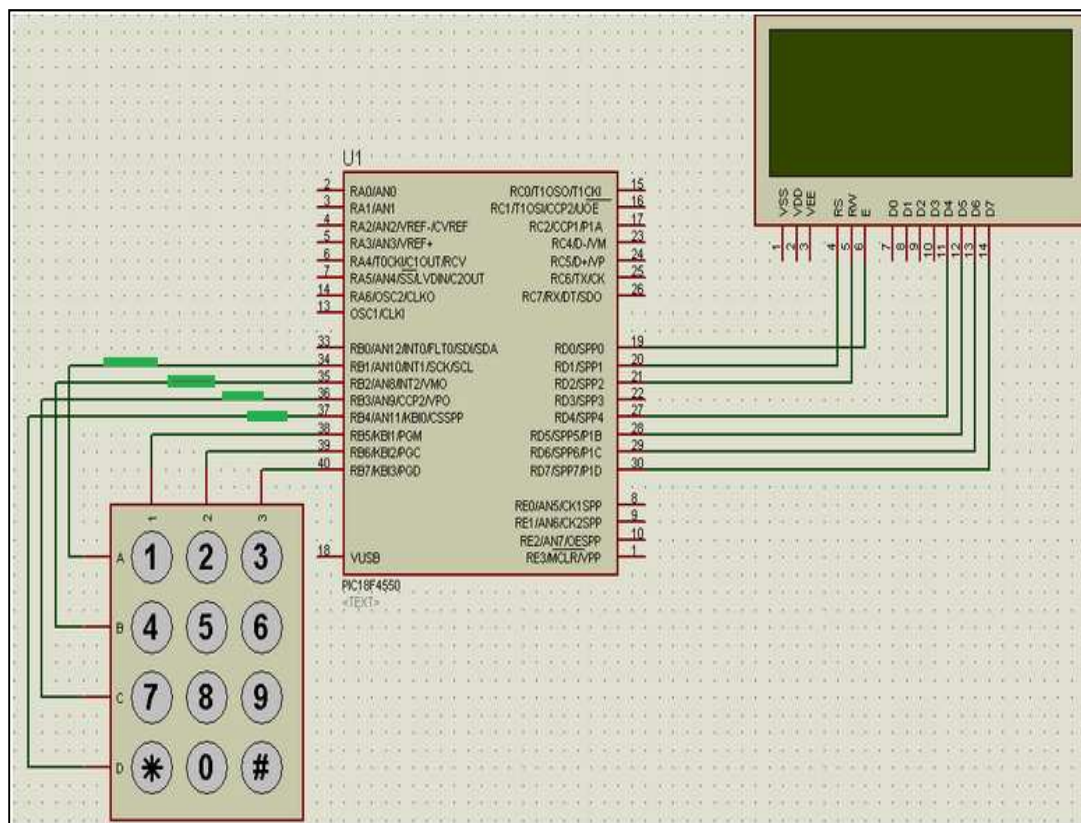
Para comunicarse con la pantalla LCD podemos hacerlo por medio de sus pines de entrada de dos maneras posibles, con bus de 8 bits o con bus de 4 bits, este último es el que explicaré y la rutina también será para este.

**Tabla 1. 1:** Descripción de pines tipo bus de 4 bits<sup>10</sup>

<b>N° DE PIN</b>	<b>SÍMBOLO</b>	<b>DESCRIPCIÓN</b>
1	Vss	Tierra de alimentación GND
2	Vdd	Alimentación de 5 Vcc
3	Vo	Contraste del cristal líquido (0v a 5v)
4	RS	Selección del registro de control/registro de datos: RS0: Selección registro de control RS1: Selección registro de datos
5	R/W	Señal de lectura/escritura R/W =0 Escritura (Write) R/W=1 Lectura (Read)
6	E	Habilitación del módulo: E=0 Módulo desconectado E=1 Módulo conectado
7-14	D0-D7	Bus de datos bidireccional

#### 1.4.2 TECLADO

Un teclado matricial es un arreglo de pulsadores conectados en filas y columnas de modo que puede leer varios botones con el mínimo número de pines requeridos, logrando optimización de espacio como se muestra en la Figura N° 1.7



**Figura 1. 8:** Diagrama de conexión tipo bus de 4 bits<sup>10</sup>

Como se muestra en la Figura N°1.8 las filas del teclado se conectan a las líneas RB1, RB2, RB3, RB4 configuradas como salidas, mientras que las columnas se conectan a las líneas RB5, RB6, RB7 las cuales son configuradas como entradas.

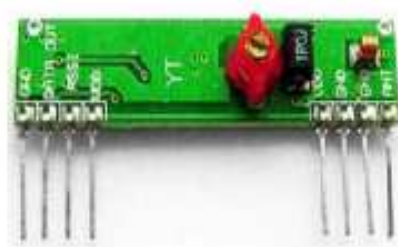
El teclado matricial de 4x3 ocupa solo 4 líneas de un puerto para las filas y 3 líneas para las columnas y de esta manera se puede leer 12 teclas utilizando solo 7 líneas de un microcontrolador.

Para detectar si hay una tecla pulsada se realiza un barrido por cada una de las filas, inicialmente están en alto (1L), la pulsación de una tecla se la detecta al poner en cada fila en un nivel bajo (0L) y chequear cada columna hasta encontrar un cero, si en ninguna de las columnas está en un nivel bajo, entonces el 0L de las filas se recorre hacia las siguientes, y así de una manera sucesiva.

<sup>10</sup> <http://www.forosdeelectronica.com/f24/manejar-teclado-matricial-mostrar-lcd-14348/>

### 1.4.3 RECEPTOR RLP434<sup>11</sup>

El circuito mostrado en la Figura N° 1.9, permite demodular los datos que se encuentran en el medio, de forma que estos pueden ser utilizados por el microcontrolador. Las características de este módulo de recepción de radiofrecuencia son:



**Figura 1. 9:** Receptor RLP 434

- ❖ Opera en el rango de frecuencias de 433.92 MHz
- ❖ Modulación ASK
- ❖ Teledetección
- ❖ Tasa de transmisión de datos de 4.8 Kbps
- ❖ Su sensibilidad es de  $-106$  dBm, este parámetro hace referencia a la potencia de recepción mínima que es capaz de detectar el módulo.
- ❖ Espaciamiento entre canales de  $\pm 500$  KHz
- ❖ Alimentación de 5 V
- ❖ Utilizado en la industria de sistema de seguridad

<sup>11</sup> <http://www.alldatasheet.com/view.jsp?Searchword=TLP434> (Anexo en el datasheet)



**Tabla 1. 2:** Características principales del RLP434<sup>12</sup>

Símbolo	Parámetros	Condiciones	Min	Typ	Max	Unid
Vcc	Operating supply voltage		3.3	5.0V	6.0	V
I <sub>tot</sub>	Operating Current		-	4.5		mA
V <sub>data</sub>	Data Out	I <sub>data</sub> = +200 uA ( High )	V <sub>cc</sub> -0.5	-	V <sub>cc</sub>	V
		I <sub>data</sub> = -10 uA ( Low )	-	-	0.3	V
Electrical Characteristics						
Characteristics	SYM	Min	Typ	Max	Unit	
Operation Radio Frequency	FC	315, 418 and 433.92			MHz	
Sensitivity	Pref		-110		dBm	
Channel Width			+500		Khz	
Noise Equivalent BW			4		Khz	
Receiver Turn On Time			5		ms	
Operation Temperature	Top	-20	-	80	C	
Baseboard Data Rate			4.8		KHz	

#### 1.4.4 MAX 232

El MAX232 es un circuito integrado que convierte las señales de un puerto serie RS232 a señales adecuadas para los circuitos TTL compatible con la lógica digital. Es un controlador de doble receptor que convierte todas las señales RX, TX, CTS y RTS.

Este circuito integrado MAX232 tiene salidas que maneja niveles de voltaje del RS-232 (aprox.  $\pm 7.5$  V) que las produce a partir de un voltaje de alimentación de + 5 V utilizando multiplicadores de voltaje, está constituido internamente de condensadores externos. Esto es de mucha utilidad para la implementación de puertos serie RS-232 en dispositivos que tengan una alimentación simple de + 5 V.

La versión MAX232A es compatible con la original MAX232, y tiene la mejora de trabajar con mayores velocidades de transferencia de información en Baudios, lo que reduce el tamaño de los condensadores externos utilizados por el multiplicador de voltaje, -0.1  $\mu$ F en lugar del 1.0  $\mu$ F usado en el dispositivo original. Como se muestra Figura N°1.11 y sus características en la Tabla 1.3



**Figura 1. 10:** Circuito Integrado MAX232

#### 1.4.4.1 Funcionamiento

La principal función del MAX 232, es acoplar los niveles de voltaje seriales, entre el microcontrolador que tiene una comunicación serial TTL que varía de 0-5 V y el módulo GSM que se comunica mediante una salida RS232 (aprox. +12V y -12V).

Además se debe poner condensadores externos de 10uf, para llegar a tener velocidades de hasta 120 Kbps como se indica en el datasheet.

**Tabla 1. 3:** Características de los puertos del MAX232

PUERTO	FUNCIÓN
C1+	Conexión positiva del condensador C1 del doblador de voltaje de +5V a +10V.
C1-	Conexión negativa del condensador C1 del doblador de voltaje de +5V a +10V.
C2+	Conexión positiva del condensador C2 del inversor de voltaje de +10V a -10V.
C2-	Conexión negativa del condensador C2 del inversor de voltaje de +10V a -10V.
V-	Conexión de salida del voltaje de -10V.
V+	Conexión de salida del voltaje de +10V.
T1in, T2in,R1out,R2out	Conexiones a niveles de voltaje de TTL o CMOS.

T1out, T2out,R1in,R2in	Conexiones a niveles de voltaje del protocolo RS-232.
VCC	Alimentación positiva del MAX232
GND	Alimentación negativa del MAX232

#### 1.4.4.2 Niveles de voltaje

Cuando un circuito integrado MAX232 recibe un nivel lógico lo convierte a un nivel lógico de 0 a un nivel comprendido entre +3 y +15 V, y cambia un nivel lógico 1 a un nivel comprendido entre -3 a -15 V, y viceversa.

**Tabla 1. 4:** Niveles de voltaje del RS232

TIPO DE LÍNEA RS232 Y NIVEL LÓGICO	VOLTAJE RS232	VOLTAJE AL O DESDE EL MAX232
Transmisión de datos (Rx/Tx) Nivel lógico 0	+3 V a +15 V	0 V
Transmisión de datos (Rx/Tx) Nivel lógico 1	-3 V a -15 V	5 V
Señales de control (RTS/CTS/DTR/DSR) Nivel lógico 0	-3 V a -15 V	5V
Señales de control (RTS/CTS/DTR/DSR) Nivel lógico 1	+3 V a +15 V	0V

#### 1.4.5 MÓDULO GSM ZTE<sup>12</sup>

El módulo GPRS/GSM ZTE está provisto por la corporación ZTE y se programa mediante los comandos AT, mediante los cuales el módulo puede comunicarse con dispositivos externos como se observa en la Figura N° 1.12.

<sup>12</sup>[bibdigital.epn.edu.ec/bitstream/15000/4391/1/CD-4004.pdf](http://bibdigital.epn.edu.ec/bitstream/15000/4391/1/CD-4004.pdf)

Los comandos AT se utilizan de acuerdo a las aplicaciones del estándar GSM de voz y de mensajes cortos.

Los módems GSM se comportan de una manera similar a un modem normal, permitiendo el intercambio de datos a otro modem mediante comandos AT originales. Estos son como pequeños teléfonos móviles, que contienen su propia tarjeta SIM para su funcionamiento y permite tener acceso a una base de datos de teléfonos, envío y recepción de mensajes SMS, además de configurar diversos parámetros.

El estándar utilizado para manipular los módems se basa en los comandos AT, el módem antes de realizar una conexión con otro, se encuentra en modo comando. En este modo podemos configurar y controlar al modem mediante los comandos AT para llegar al modo conexión y de esta manera llega la información al modem por el puerto serial. Una vez establecida la conexión con un modem remoto, se va de modo comando a modo conexión, ya que la información que llega al modem por el puerto serial no es interpretada como comandos AT sino como información a transmitir, pero una vez terminada la conexión el módem vuelve a modo comando.

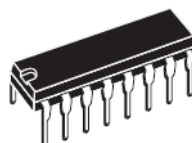
Los comandos AT son cadenas de ASCII que comienzan por caracteres AT y terminan con un remoto. Cuando el módem recibe un comando lo procesa y devuelve un resultado, lo que normalmente es una cadena de ASCII.



**Figura 1. 11:** Ejemplo de Módulo GSM

### 1.4.6 Circuito Integrado ULN2003A<sup>13</sup>

En la Figura N° 1.13 se puede observar que dentro del ULN 2003, se encuentran 7 transistores NPN Darlington, ideal para ser empleado como interfaz entre salidas de cualquier integrado de la familia TTL o CMOS, además de dispositivos que requieran una corriente más elevada para funcionar, como un relé.



DIP16

**Figura 1. 12:** Circuito Integrado ULN2003A

#### 1.4.6.1 Características generales<sup>14</sup>

- ❖ Arreglo de siete darlingtons
- ❖ Corriente de salida 500mA.
- ❖ Salida de voltaje 50V.
- ❖ Diodos integrados para supresión inductiva.
- ❖ Las salidas se pueden conectar en paralelo para mayor corriente
- ❖ Tiene entradas TTL / CMOS / PMOS / DTL COMPATIBLES con las salidas para simplificar la distribución.

#### 1.4.6.2 Descripción del ULN2003A<sup>14</sup>

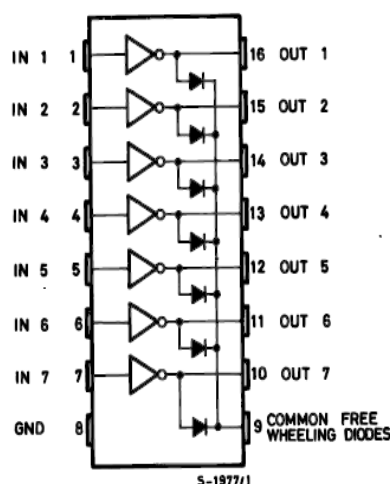
El ULN2003A tienen un alto voltaje y alta corriente en el arreglo de matrices con darlingtons, cada uno de estos contiene siete pares de colectores abiertos darlingtons con emisores comunes.

---

<sup>13</sup> [http://www.datasheetcatalog.net/es/datasheets\\_pdf/U/L/N/2/ULN2003.shtml](http://www.datasheetcatalog.net/es/datasheets_pdf/U/L/N/2/ULN2003.shtml)

Contiene un canal nominal de 500 mA y puede soportar corrientes de pico de 600 mA. Tiene diodos de supresión que sirven para conducir carga inductiva, entradas y salidas opuestas para simplificar la distribución de la tarjeta.

La distribución de pines del ULN2003A se muestra en la Figura N° 1.14.



**Figura 1. 13:** Distribución de pines del ULN2003A<sup>14</sup>

Las interfaces para el ULN2003A son como se muestra una en la Tabla 1.5

**Tabla 1. 5:** Interfaces para el ULN2003A

ULN2003A	5 V TTL,CMOS
----------	--------------

Debido a que este dispositivo es versátil tiene diferentes aplicaciones como: manejar los relés de motores DC, displays de LED y buffers de alto nivel de energía.

Symbol	Parameter	Value	Unit
$V_o$	Output Voltage	50	V
$V_{in}$	Input Voltage (for ULN2002A/D - 2003A/D - 2004A/D)	30	V
$I_c$	Continuous Collector Current	500	mA
$I_b$	Continuous Base Current	25	mA
$T_{amb}$	Operating Ambient Temperature Range	- 20 to 85	°C
$T_{sig}$	Storage Temperature Range	- 55 to 150	°C
$T_j$	Junction Temperature	150	°C

**Figura 1. 14:** Características técnicas del ULN2003A<sup>14</sup>

## **1.5 SISTEMAS DE SEGURIDAD**

En un sistema de seguridad que consta de componentes de software, hardware, dispositivos periféricos y equipo de control son controlados por un operador de seguridad. Los diseñadores deben determinar el software y el hardware, los cuales se ajusten a las necesidades del cliente. Permitiendo un sistema de seguridad que garantice al usuario no solo tenga confianza en el sistema sino que también se sienta cómodo con el servicio.

Todo este equipo físico está controlado por programas de aplicación, estos programas pueden incluir la implementación de una base de datos, la acción de límites de administración de los operadores, generando un control determinado para los elementos de seguridad instalados dentro de la casa, los cuales son necesarios para proporcionar a las personas y bienes materiales una protección frente agresiones, robos incendios etc.

Así, en un siniestro, en un principio se lo detecta, luego se inicia acciones para determinar las causas que activaron el sistema de alarma y posteriormente establecer una solución para extinguir los efectos.

Los sistemas de seguridad pueden variar, según las necesidades del lugar a proteger y del presupuesto que se tenga para este.

Algunas de sus aplicaciones principales de los sistemas de seguridad son en los hogares de clase media, oficinas de las ciudades.



**Figura 1. 15:** Ejemplo de sistema de seguridad.<sup>14</sup>

Para el diseño de un sistema de alarma es necesaria la investigación de algunos términos o definiciones.

### 1.5.1 COMPONENTES DE UN SISTEMA DE SEGURIDAD

La mayoría de los sistemas de seguridad puede realizar tan solo una función así como también un amplio rango de funcionalidades. Los diferentes componentes que puede contener un sistema de seguridad son:

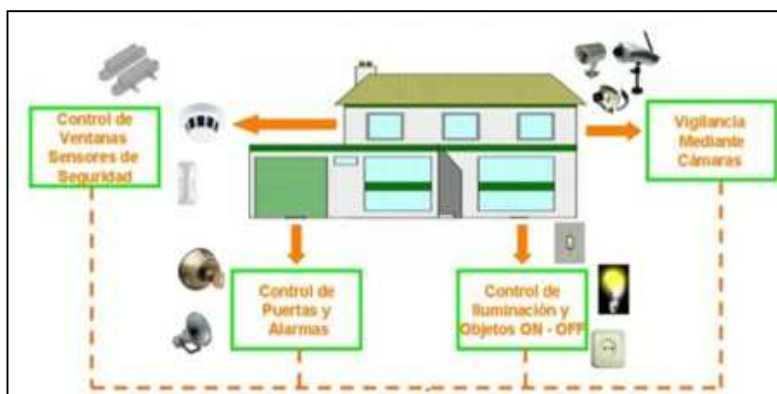
- ❖ **Central de Seguridad:** Este dispositivo controla el sistema según su programación y la información que recibe. Otro de sus propósitos también es ser responsable de la comunicación del sistema de seguridad con el exterior, así como el aviso a una central de alarma.
- ❖ **Detector:** Es un sensor que monitoriza todo a su alrededor, detectando anomalías como son movimientos, presencia, temperatura, apertura de puertas y ventanas, presencia de gas y de humo.
- ❖ **Medio de Transmisión:** Es la infraestructura que conduce la información e interactúa con los distintos dispositivos del sistema de seguridad por medio de un cableado (red eléctrica, telefónica o de datos) o en forma inalámbrica.
- ❖ **Interfaces:** Estas interfaces se refiere a los dispositivos y sus formatos que muestran información del sistema para los usuarios y además de los cuales se

<sup>14</sup> <http://protecciontotal.blogspot.com/>



puede interactuar con los dispositivos del sistema como son el teclado, voz, web, móvil, etc.

- ❖ **Sirenas:** Son dispositivos que generan un sonido alto, alternando con avisos luminosos. Uno de sus objetivos es de dar alerta a los propietarios de la casa y a las personas de su alrededor, así como el asustar a los posibles intrusos.
- ❖ **Cámaras y Grabadoras de Video:** Son componentes de los sistemas de seguridad que captan información, graban imágenes, sonidos dentro y fuera del inmueble para poder ser revisados posteriormente.



**Figura 1. 16:** Ejemplo de los componentes de un Sistema de Seguridad

Para el diseño de un sistema de alarma es necesaria la investigación de algunos términos o definiciones.

### 1.5.2 ALARMA

Es un dispositivo que detecta la presencia de una acción anormal por medio de un cambio visible o de una señal audible, en ocasiones puede tratarse de las dos señales al mismo tiempo, las cuales tienen un fin de atraer la atención.

### 1.5.3 TIPOS DE ALARMAS

Básicamente se distinguen dos tipos de alarmas de acuerdo al grado de seguridad que necesitemos:

### 1.5.3.1 Con conexión a la central

Este funciona detectando una anomalía, informara a la empresa de seguridad que a la vez avisará a la policía o acudirán ellos mismos al lugar, este sistema de seguridad tiene un gran éxito en lugares que se encuentran ubicados en zonas muy alejadas o simplemente no están habitados.

### 1.5.3.2 Sin conexión a la central

Siendo el más básico cumple con todas las necesidades en los lugares habitados en zonas urbanas de manera continua.

El funcionamiento de este tipo de alarmas es sencillo y consiste en emitir un sonido cuando se intenta ingresar a un lugar. Con esto se pretende ahuyentar al intruso y llamar la atención de las personas que se encuentren a su alrededor así como de la policía.

## 1.5.4 SISTEMA DE ALARMA

Un sistema de alarma es un elemento de seguridad pasiva lo cual significa que esto no evitara una situación anormal, pero si tienen la capacidad de advertir de alguna situación de peligro, cumpliendo así su función de advertencia de posibles problemas.

### 1.5.4.1 Partes principales de un sistema de alarma

- ❖ **Central procesadora:** Es una CPU del sistema, en el que se encuentra la tarjeta madre, la fuente y la memoria principal. Aquí se recibe las diferentes señales que los sensores emiten y los mismos trabajan disparando una alarma de alerta, comunicándose con la central por medio de un modem.
- ❖ **Teclado:** Es un teclado numérico de tipo telefónico, el cual permite a los usuarios autorizados armar, activar y desarmar mediante códigos preestablecidos. Además puede tener botones con funciones de Emergencias, Médica y Fuego.

- ❖ **Detectores de Movimiento (PIR):** Son sensores que detectan movimiento y temperatura, pero a la vez excluyen los movimientos de las mascotas.
- ❖ **Sensores inerciales o sísmicos:** Son aquellos que detectan golpes sobre una base. Se colocan especialmente en cajas fuertes, también en puertas, paredes y ventanas, detectando un intento de forzar su apertura.
- ❖ **Batería y cargador:** Estos elementos sirven para proveer de alimentación eléctrica interrumpida, de esta manera ante una falta de suministro eléctrico de la red provocado por un intruso, el sistema de alarma actúe normalmente



**Figura 1. 17:** Partes principales de un sistema de alarma

### 1.5.5 SISTEMA DE ALARMA INALÁMBRICO CON MÓDULO GSM

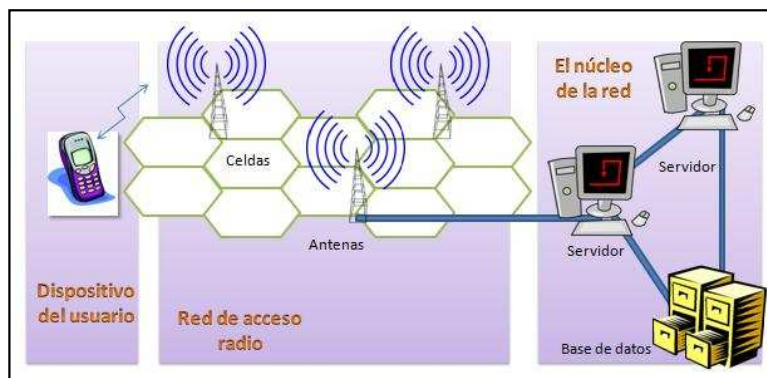
A diferencia de las alarmas normales un sistema de alarma inalámbrico con módulo GSM, puede comunicarse mediante una red GSM celular sin tener que utilizar necesariamente una línea de teléfono convencional. A esta alarma se le puede configurar vía manual o través de su teléfono móvil activando o desactivando mediante un mensaje de texto. También se puede acceder a otras configuraciones vía SMS entre estas el cambio de clave, que por seguridad se lo debe realizar cada 3 a 4 meses.

Un sistema inalámbrico tiene algunas ventajas frente a un sistema normal entre estos se puede nombrar que un sistema de seguridad inalámbrico no tiene cables, facilitando la instalación de una manera más rápida en la casa o en la oficina que requieran un servicio de seguridad.

La mayoría de las alarmas normales no tienen la posibilidad de que cuando se registra un robo, se pueda alertar de manera inmediata a su propietario. Todo lo contrario de un sistema de alarma inalámbrico con GSM, en el cual por ejemplo se logra almacenar 5 número telefónicos: 3 números de los propietarios y un número que esta enlazado a la policía para que ellos puedan actuar de manera inmediata frente a una posible alerta de robo.

## **1.6 TECNOLOGÍA GSM**

**GSM** es un sistema digital de comunicaciones móviles, que se basa en el concepto de celdas, es decir zonas circulares que se superponen para cubrir una área geográfica, la cual es utilizada para transmitir voz y datos, en su proceso digitaliza y comprime la información y la trasmite mediante la asignación a cada usuario de una celda llamada una ranura de tiempo. Permitiendo que múltiples llamadas compartan un mismo canal simultáneamente sin tener interferencias con las demás. Este sistema a diferencia de la primera generación de telefonía, este tiene hilos de tecnología digital. Mediante una tecnología TDMA (Acceso múltiple con división en el tiempo), codifica la voz digitalmente con un único vocoder simulando de esta manera algunas de las características de la voz humana, permitiendo una eficiente velocidad de datos e información.



**Figura 1. 18:** Tecnología GSM.

### 1.6.1 TECNOLOGÍA GSM COMO UN ESTÁNDAR

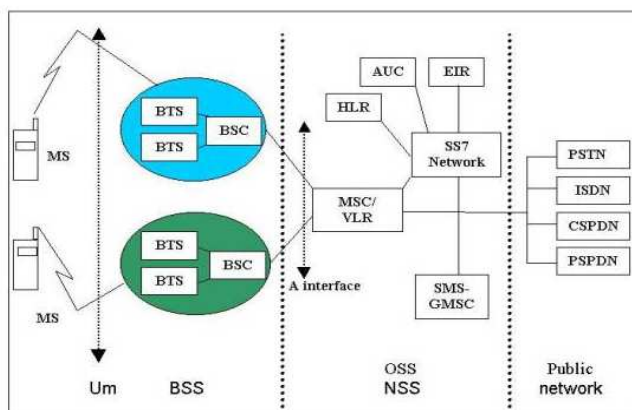
La red GSM tiene sus inicios en 1982, en la conferencia de telecomunicaciones CEPT de ese año, cuya tarea era desarrollar un estándar europeo de telefonía móvil digital. En 1990 se finalizaron las especificaciones para el primer estándar GSM-900, al que siguió DCS-1800 un año más tarde. En 1991 fueron presentados los primeros equipos de telefonía GSM como prototipos. En 1992 las primeras redes europeas de GSM-900 iniciaron su actividad. En los años siguientes, el GSM es considerado como el estándar más usado en Europa, compartiendo con otros estándares digitales, pero se terminó imponiendo también en América Latina y Asia. En 2000 el GSM, se denomina estándar "De Segunda Generación" (2G) porque, a diferencia de la primera generación de teléfonos portátiles, las comunicaciones se producen de un modo completamente digital.

El estándar GSM permite un rendimiento máximo de 9,6 Kbps, que permite transmisiones de voz y de datos digitales, como mensajes de texto SMS.

### 1.6.2 ARQUITECTURA DE LA RED GSM

La arquitectura de un sistema GSM se constituye como una unidad de celdas radioeléctricas contiguas, con una cobertura del área de servicio.

Una red GSM está compuesta de subsistemas principales y varias etapas dentro de estas, como se indica en la Figura N° 1.20.



**Figura 1. 19:** Arquitectura GSM<sup>15</sup>

GSM consiste en 4 subsistemas, los cuales son:

- ❖ **Estación móvil (MS):** Consta de equipos que presentan algunos servicios o soportan una conexión externa o terminales externos, esto puede ser una interfaz de una PC o un Fax. En una Estación Móvil presenta equipos móviles además de un módulo de identificación suscriptor (SIM). La SIM guarda toda la información del suscriptor, una vez que se inserta la SIM el equipo pasa a formar una estación móvil que pertenece al suscriptor, y las llamadas llegan a esta estación. El equipo móvil no tiene registrado ningún número telefónico, ya que la que tiene esa información es la SIM de esta manera el usuario puede utilizar cualquier equipo, solo tiene que insertar la SIM con la información.
- ❖ **Sistema de Estación Base (BSS):** La BSS se comunica con la Estación Móvil mediante señales de radio, y a la vez se conecta a la NSS. La BSS consta de una estación transceptora base (BTS), la cual consta de equipos transmisores/receptores de radio en una banda de uso (900/1800 MHz), con el cual se enlazan con el usuario realiza o recibe llamadas y mensajes.
- ❖ **Subsistema de red y conmutación (NSS):** Este subsistema utiliza redes inteligentes y administra las comunicaciones entre los usuarios GSM y otros usuarios de telecomunicaciones. La NSS administra algunos subconjuntos como son: Centro de Conmutación de servicio móvil (MSC), Registro de

<sup>15</sup> <http://seguridadengprs.galeon.com/tema2.htm>

localización Local (HLR), Registro de posiciones de visitante (VLR) y Gateway MSC (GMSC).

### 1.6.3 BANDAS DE FRECUENCIAS DE OPERACIÓN GSM.

El espectro de frecuencias de operación del Sistema Global de Comunicaciones Móviles usan bandas de frecuencia de 900 MHz y 1800 MHzsin embargo, en los Estados Unidos se usa la banda de frecuencia de 1900 MHz por esa razón, los teléfonos portátiles que funcionan tanto en Europa como en los Estados Unidos se llaman TRIBANDA y aquellos que funcionan sólo en Europa se denominan BIBANDA.

Banda	Nombre	Canales	Uplink (MHz)	Downlink (MHz)	Notas
<b>GSM 850</b>	GSM 850	128 - 251	824,0 - 849,0	869,0 - 894,0	Usada en los EE.UU., Sudamérica y Asia.
	P-GSM 900	0-124	890,0 - 915,0	935,0 - 960,0	La banda con que nació GSM en Europa y la más extendida
<b>GSM 900</b>	E-GSM 900	974 - 1023	880,0 - 890,0	925,0 - 935,0	E-GSM, extensión de GSM 900
	R-GSM 900	n/a	876,0 - 880,0	921,0 - 925,0	GSM ferroviario (GSM-R).
<b>GSM1800</b>	GSM 1800	512 - 885	1710,0 - 1785,0	1805,0 - 1880,0	
<b>GSM1900</b>	GSM 1900	512 - 810	1850,0 - 1910,0	1930,0 - 1990,0	Usada en Norteamérica, incompatible con GSM-1800 por solapamiento de bandas.

**Figura 1. 20:** Bandas de frecuencias asignadas a GSM.<sup>16</sup>

### 1.6.4 FRECUENCIAS QUE OPERAN MOVISTAR, CLARO Y ALEGRO.

En el Ecuador existen 3 conocidas operadoras como son: Claro, Movistar, y Alegro.

En 1993 la operadora Porta inicia sus actividades en nuestro país, para el 2011 la empresa realiza una alianza con Telmex y su nombre fue cambiado a CLARO (EcuadorTelecom) el cual ofrece servicios Telefonía Fija, Telefonía Móvil, Internet. Esta operadora trabaja en la banda GSM-850, además de otro servicio como es 3G (Tercera Generación), la cual trabaja en la banda de 3G850, lo cual le permite dar servicio a sus 10 millones de usuarios en el Ecuador con una cobertura más extensa del país, con antenas en gran cantidad de ciudades, pueblos y aéreas urbanas de esta manera llegando a todos las provincias.

<sup>16</sup>Fuente:[http://es.wikipedia.org/wiki/Sistema\\_global\\_para\\_las\\_comunicaciones\\_m%C3%B3viles#Arquitectura\\_de\\_red](http://es.wikipedia.org/wiki/Sistema_global_para_las_comunicaciones_m%C3%B3viles#Arquitectura_de_red)

Otra empresa telefónica que aparece en 1998 fue Bellsouth, para después en el año de 2004 cambio su nombre a MOVISTAR (Otecel), con 4 millones de usuarios en el Ecuador, trabaja con una tecnología CDMA y una banda GSM 850 MHz brindando servicio de telefonía Móvil e Internet. Su cobertura llega a muchas ciudades y pueblos, pero todavía tiene ciertos lugares en los cuales su cobertura no alcanza, debido a que la tecnología CDMA que utiliza solo llega a los sitios más importantes del país.

Por último está la operadora ALEGRO es una empresa Ecuatoriana que aparece en el año 2003. Actualmente mediante un decreto presidencial es parte de la CNT (Corporación Nacional de Telecomunicaciones), la cual trabaja brindando su servicio de telefonía móvil e internet en una frecuencia GSM 1900 MHz, funciona solo en las ciudades más importantes de la costa y de sierra con 400 mil clientes y cada vez con tendencia a la baja.

#### **1.6.5 MENSAJES DE TEXTO SMS<sup>17</sup>**

El servicio de mensajes SMS (Short messages services) es una red digital que permite a los usuarios de teléfonos celulares enviar y recibir mensajes de texto. Un mensaje de texto SMS puede ser enviado desde un teléfono celular, un modem o desde una dirección IP, cada mensaje tiene una longitud de hasta 160 caracteres. Estos 160 caracteres pueden ser palabras, números o una combinación alfanumérica y no contiene imágenes o gráficos.

Para utilizar el servicio de mensajes cortos los usuarios necesitan la suscripción y el hardware específico, determinados por los siguientes puntos:

- ❖ Una suscripción a una red de telefonía móvil que soporte SMS.
- ❖ Un teléfono móvil que soporte SMS.
- ❖ Un destino para enviar o recibir el mensaje, un PC, un terminal móvil o un buzón de e-mail.

---

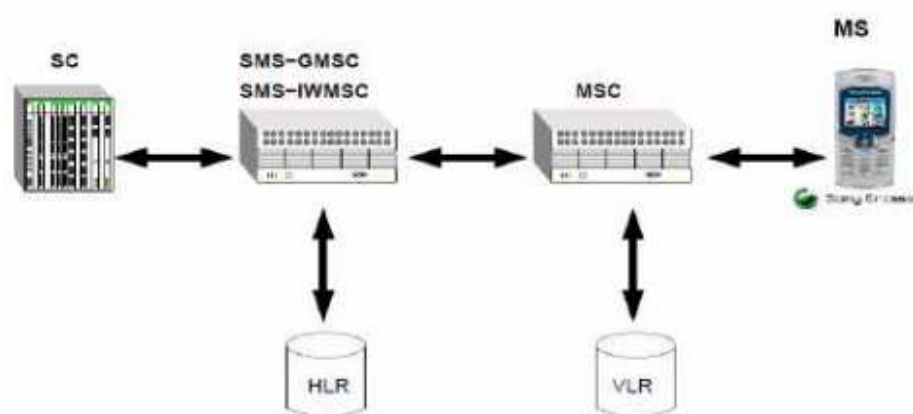
<sup>17</sup> <http://dspace.espace.edu.ec/bitstream/123456789/331/1/38T00174.pdf>



Los mensajes SMS pueden ser enviados desde teléfonos TDMA, CDMA, GSM, bajo la red móvil celular y son transferidos entre teléfonos por medio del Centro de servicios de mensajes cortos (SMSC). El SMSC es un software de la operadora de red que se encarga de manejar, procesar y facturar los mensajes. El despacho de los mensajes se realiza en colas de espera de tal forma que el mensaje tarda un tiempo en llegar al usuario destino el cual depende de la cantidad de mensajes y de la velocidad del software de la operadora.

### 1.6.5.1 Arquitectura de la red SMS<sup>18</sup>

La figura N° 1.21 se muestra la estructura básica de la red SMS, con la que funciona la arquitectura de una red SMS.



**Figura 1. 21:** Elementos de la red SMS

#### Entidad de envío de mensajes cortos SME

SME (Short messaging entity) entidad que puede enviar o recibir mensajes cortos pudiendo estar localizados en la red fija, una estación móvil, u otro centro de servicio.

#### Centro de servicio de mensajes SMSC

SMSC (Short Message Service Center) es el responsable de la transmisión y almacenamiento de un mensaje corto, entre el SME y una estación móvil.

### **Centro de conmutación móvil MSC**

MSC (Mobile switching center) lleva a cabo funciones de conmutación del sistema, el control de llamadas y desde otro teléfono y sistema de datos.

### **SMS-Gateway/interworking MSC (SMS-GMSC)**

Es un MSC capaz de recibir un mensaje corto de un SMSC, interrogando al HLR (Home location register) sobre la información de encaminamiento y enviando el mensaje corto al MSC.

#### **1.6.5.2 Pasos de envío y recepción de SMS<sup>18</sup>**

Pasos para el envío:

- ❖ El mensaje corto es enviado del SME al SMSC.
- ❖ Después de completar su proceso interno, el SMSC pregunta al HLR y recibe la información de enrutamiento del usuario móvil.
- ❖ El SMSC envía el mensaje corto hacia el MSC.
- ❖ El MSC extrae la información del usuario del VLR. Esta operación que se está llevando a cabo.
- ❖ Si lo solicita el SME, el SMSC retorna un informe indicando la salida del mensaje corto.

Pasos para la recepción:

- ❖ La MS transfiere el mensaje corto al MSC.
- ❖ El MSC interroga al VLR para verificar que el mensaje transferido no viola los servicios suplementarios para las restricciones impuestas.
- ❖ El MSC envía el mensaje corto al SMSC usando el mecanismo ForwardShortMessage.
- ❖ El SMSC entrega el mensaje corto al SME.
- ❖ El SMSC reconoce al MSC el éxito del envío
- ❖ El MSC devuelve a la MS el resultado de la operación de envío.

### 1.6.5.3 Aplicaciones<sup>18</sup>

Las principales aplicaciones basadas en SMS son:

#### ❖ Mensajes de persona a persona

Los usuarios de telefonía móvil utilizan comúnmente el servicio de mensajería corto para comunicarse con otro usuario móvil de su misma operadora e incluso de una operadora diferente.

#### ❖ Alertas de E-mail

Los SMS permiten notificar al usuario que tiene un nuevo e-mail. Este mensaje usualmente contiene la dirección de quien envía, el título y unas pocas palabras del inicio de E-mail.

#### ❖ Servicios de notificación

Permite el envío de mensajes a ciertos usuarios que constan en una base de datos específica tales como: clientes de compañías de televisión, clubs deportivos, supermercados y otros minoristas.

#### ❖ Servicio de información

Permite enviar al terminal móvil mensajes con pequeños contenidos de información periódica, de un amplio rango como reporte del clima, reportes financieros, información deportiva.

#### ❖ 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.

### **1.6.6 TECNOLOGÍA GSM APLICADA A LOS SISTEMAS DE SEGURIDAD<sup>18</sup>**

En este proyecto se aplica la tecnología GSM para crear un sistema de seguridad y control de luces, mediante el cual se mantendrá los domicilios protegidos en la ausencia de los propietarios. El principal objetivo de este proyecto es utilizar un teléfono celular que en estos días pueda estar al alcance económico de casi todas las personas, recibiendo una señal de alerta si ocurre un intento de robo en sus domicilios. A diferencia de otros sistemas de seguridad este tiene un alcance, en cualquier lugar donde exista la cobertura GSM. Gracias a la tecnología GSM es posible el control de cualquier dispositivo electrónico dentro del hogar. En este proyecto se eligió la activación y desactivación en el domicilio, para simular que alguien se encuentra presente en el mismo, disminuyendo así los intentos de robo a los hogares.

---

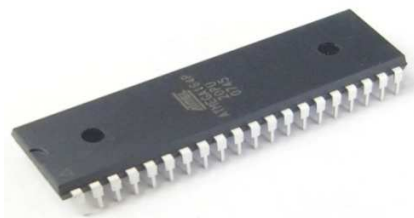
<sup>18</sup>[bibdigital.epn.edu.ec/bitstream/15000/4869/1/CD-4463.pdf](http://bibdigital.epn.edu.ec/bitstream/15000/4869/1/CD-4463.pdf)

## CAPÍTULO II

### 2 DISEÑO DE LOS CIRCUITOS DEL SISTEMA DE CONTROL DE SEGURIDAD

#### 2.1 DESCRIPCIÓN DE LOS DISPOSITIVOS

##### 2.1.1 MICROCONTROLADOR ATMEGA 164P<sup>19</sup>



**Figura 2. 1:** Ejemplo Atmega164P

El ATmega164P es un microcontrolador CMOS de 8 bits de bajo consumo fabricado por ATMEL y perteneciente a la familia Atmega de 40 pines, basado en la arquitectura RISC mejorada. Sus instrucciones se ejecutan en un ciclo de máquina, el ATmega164P consigue transferencias de información alrededor de 1 MIPS por MHz admitido por el sistema, permitiendo al diseñador del sistema optimizar el consumo de energía versus la velocidad de procesamiento.

##### 2.1.1.1 Características generales<sup>20</sup>

###### ❖ Arquitectura avanzada RISC.

- 32 registros de trabajo de 8 bits.
- Capacidad de procesamiento de unos 20 MIPS a 20 MHz
- 131 instrucciones de un solo ciclo de reloj en su mayoría.

---

<sup>19</sup><http://www.atmel.com/Images/doc7674.pdf>

### ❖ Interfaz JTAG

- Soporte Extendido Debug dentro del chip.
- Programación de EEPROM, Flash, fusibles y bits de bloqueo a través de la interface JTAG.

### ❖ Características de los periféricos.

- Dos Timer/Contadores de 8 bits con prescalamiento separado y modo comparación.
- Un Timer/Contador de 16 bits con prescalamiento separado, modo comparación y modo de captura.
- Contador en Tiempo Real con Oscilador separado.
- 6 Canales para PWM.
- ADC de 10 bits y 8 canales.
- Dos puertos Seriales USART Programables.
- Interfaz Serial SPI maestro-esclavo.
- Watchdog Timer programable con oscilador independiente, dentro del mismo chip.
- Comparador Analógico dentro del mismo Chip.
- Interrupt and Wake-up on Pin Change.

### ❖ Características especiales del microcontrolador

- Oscilador RC interno calibrado.
- Fuentes de interrupción externas e internas.
- 6 modos de descanso: Idle, reducción de ruido ADC, Standby extendido, Power-save, Power-down y standby.

### ❖ Encapsulados para Entradas/Salidas (E/S)

- 32 líneas de E/S programables.

- PDIP de 40 pines.

#### ❖ Voltajes de Operación

- 1.8 -2.7– 5.5V para el ATMEGA 164P.

#### ❖ Velocidad de Funcionamiento

- 0 – 4MHz: 1.8– 5.5V y 10MHz: 2.7 – 5.5V.

#### ❖ Memorias de programa de datos no volátiles de alta duración.

- 16.32Kbytes de FLASH auto programable en el sistema.
- 512 Kbytes de EEPROM.
- 1Kbyte de RAM.
- Ciclos de escritura y borrado: 10.000 en Flash y 100.000 en EEPROM respectivamente.
- Operación de lectura durante la escritura.
- Bloqueo programable para la seguridad del software.

### 2.1.2 PUERTOS DEL ATMEGA164P<sup>20</sup>

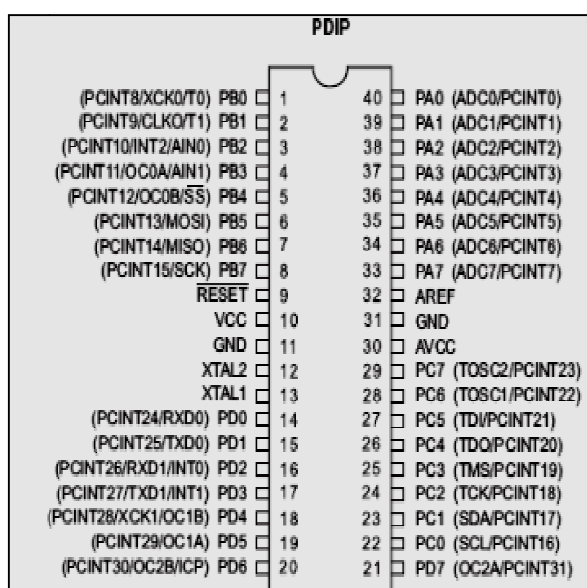


Figura 2. 2: Distribución de pines del Atmega164P

- ❖ **VCC:** Alimentación de Voltaje Digital.
- ❖ **GND:** Tierra.
- ❖ **Puerta A (PA7-PA0):** El puerto A sirve como entradas analógicas para el conversor Análogo Digital. El puerto A también sirve como un puerto bidireccional de 8 bits con resistencias internas de pull-up (seleccionables para cada bit). Los pines del puerto A están en tri-estado cuando las condiciones de reset están activadas o cuando el reloj no esté corriendo.
- ❖ **Puerta B (PB7-PB0):** Es un puerto bidireccional de 8 bits de E/S con resistencias internas de pull up. Las salidas de los buffers del puerto B tienen características simétricas controladas con fuentes de alta capacidad. Los pines del puerto B están en tri-estado cuando las condiciones de reset están activadas o cuando el reloj no esté corriendo.
- ❖ **Puerta C (PC7-PC0):** El puerto C es un puerto bidireccional de 8 bits de E/S con resistencias internas de pull-up (seleccionadas por cada bit). Las salidas de los buffers del puerto C tienen características simétricas controladas con fuentes de alta capacidad. Los pines del puerto C están en tri-estado cuando las condiciones de reset están activadas siempre y cuando el reloj no esté corriendo. El puerto C también sirve para las funciones de Interfaz del JTAG.
- ❖ **Puerta D (PD7-PD0):** El Puerto D es un puerto bidireccional de entradas y salidas con resistencias internas de pull-up (seleccionadas por cada bit). Las salidas de los buffers del puerto D tienen características simétricas controladas con sumideros de fuentes de alta capacidad. Los pines del Puerto D están en tri-estado cuando llega una condición de reset activa, siempre y cuando el reloj no esté corriendo.
- ❖ **Reset:** Un pulso de nivel bajo en este pin por períodos de pulso mínimo genera un reset, siempre y cuando el reloj no esté corriendo.  
La longitud del pulso mínimo está especificada en las características y Sistemas de Reset. Pulsos cortos no son garantizados para generar un reset.
- ❖ **XTAL1:** Entrada para el amplificador del oscilador invertido y entrada para el circuito de operación del reloj interno.
- ❖ **XTAL2:** Salida del Oscilador amplificador de salida.



- ❖ **AVCC:** Es la alimentación de voltaje para el pin del Puerto F y el Conversor Análogo a Digital. Este debe ser conectado externamente a Vcc, siempre y cuando el ADC no sea usado. Si el ADC es usado, este deberá ser conectado a Vcc a través de un filtro paso bajo.

## **2.2 DESARROLLO DEL CIRCUITO DEL SISTEMA DE SEGURIDAD**

Para nuestro proyecto utilizamos un sistema de seguridad inalámbrica, en una maqueta para la simulación de viviendas de clase media- alta, con sensores en puertas y ventanas, además un sistema de alerta mediante la tecnología GSM de mensajes cortos a sus propietarios.

### **2.2.1 DESCRIPCIÓN DELAS ETAPAS DEL HARDWARE**

Los principales componentes del sistema de seguridad son:

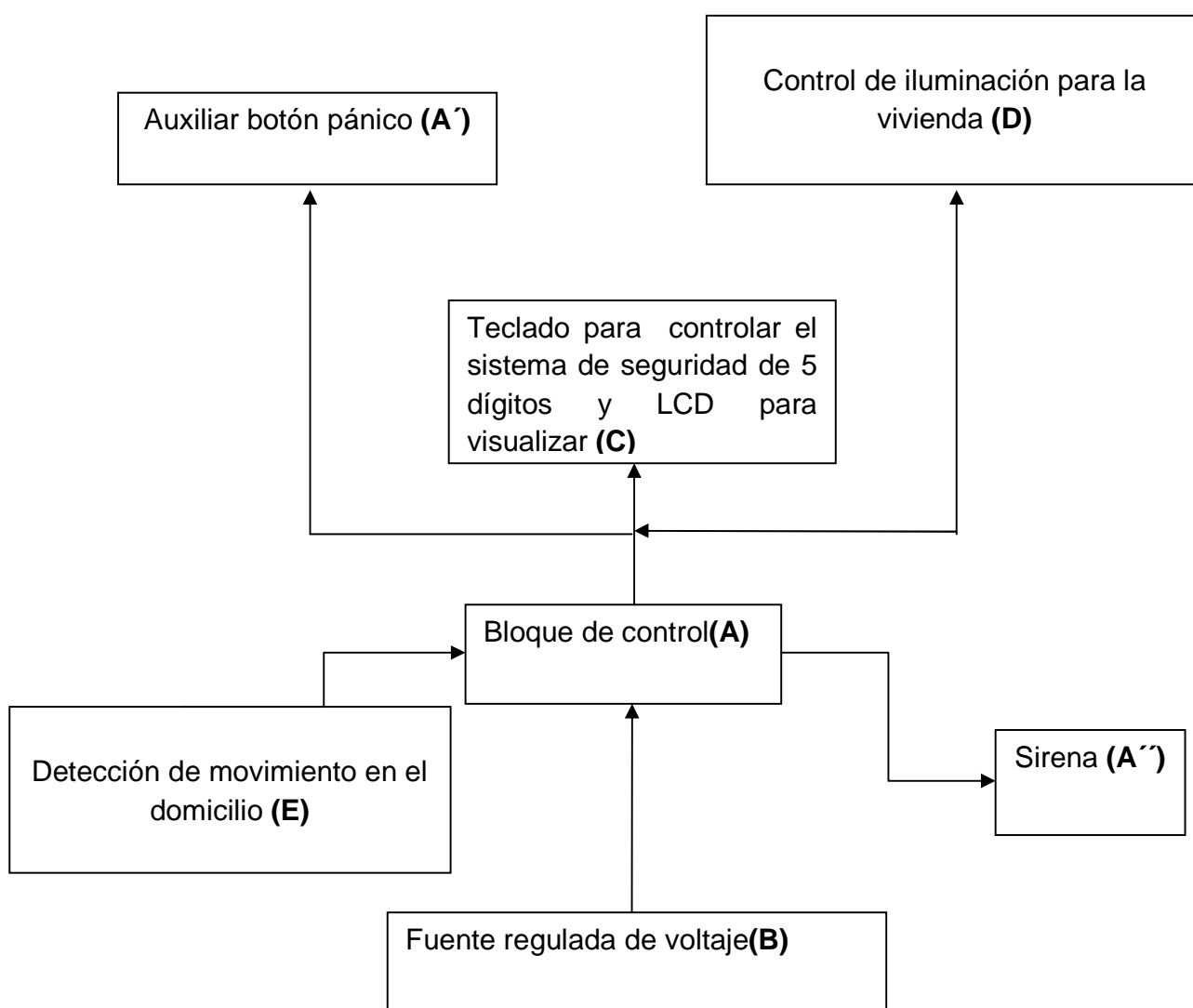
- ❖ Módulo GSM
- ❖ Sistema de control de seguridad
- ❖ Sensores inalámbricos de movimiento y magnéticos.
- ❖ Fuente Regulada de voltaje

Nuestro sistema de alarma tiene un teclado para activar, desactivar y cambiar contraseña. Cuando sus propietarios se encuentran fuera del hogar, el sistema de seguridad cumple las funciones que se muestran en la Tabla 2.1:

Para el diseño del hardware es necesario tener en cuenta como se encuentra estructurado su funcionamiento en un diagrama de bloques como se muestra en la Figura N°2.3.

**Tabla 2. 1:** Tabla de funciones del sistema d seguridad

<b>FUNCIÓN</b>	<b>DESCRIPCIÓN</b>
Encendido y apagado de luces	Mensajes vía teléfono móvil
Activación y desactivación de Alarma	Mensajes vía teléfono móvil y teclado
Botón de pánico	Pulsación del usuario dentro del inmueble
Cambio de contraseña	Mediante teclado
Sirena	Violación de la seguridad de la casa
5 números telefónicos	Contactos para aviso de alerta
Alerta en caso de robos	Mensajes de texto a 5 números telefónicos.

**Figura 2. 3:** Diagrama de Bloques del sistema de seguridad

### 2.2.1.1 Fase de control (A)

En esta etapa se describe el funcionamiento del cerebro del sistema de seguridad, la parte primordial es el microcontrolador ATmega164P conformado por 40 pines, los cuales están distribuidos como se explicó anteriormente.

Algunas de las exigencias del software del microcontrolador para el diseño de nuestro circuito son:

- ❖ Controlar un LCD 16x2 para imprimir los caracteres deseados.
- ❖ Procesar la información que se ingresa en el teclado.
- ❖ Capacidad de comunicación con una PC y el módulo GSM mediante el MAX 232, permitiendo una transmisión serial.

A continuación se describe la conexión de cada pin en el microcontrolador Atmega164P.

#### ❖ Puerto A

**PA0 (pin 40):** Sirena.

**PA1 (pin 39):** Relé que controla Foco 1

**PA2 (pin 38):** Relé que controla Foco 2

**PA3 (pin 37):** Relé que controla Foco 3.

**PA4 (pin 36):** Relé que controla Foco 4

#### ❖ Puerto B

**PB0 (pin 1):** Este terminal se conecta a la fila 1 del teclado matricial.

**PB1 (pin 2):** Este terminal se conecta a la fila 2 del teclado matricial.

**PB2 (pin 3):** Este terminal se conecta a la fila 3 del teclado matricial

**PB3 (pin 4):** Este terminal se conecta a la fila 4 del teclado matricial

**PB4 (pin 5):** Este terminal se conecta la columna 1 del teclado matricial.

**PB5 (pin 6):** Este terminal se conecta la columna 2 del teclado matricial.

**PB6 (pin 7):** Este terminal se conecta la columna 3 del teclado matricial.

**PB7 (pin 8):**Pulsador de pánico

#### ❖ Puerto C

**PC2 (pin 24):** DB7 LCD

**PC3 (pin 25):** DB6 LCD

**PC4 (pin 26):** DB5 LCD

**PC5 (pin 27):** DB4 LCD

**PC6 (pin 28):**Enable (E) del LCD

**PC7 (pin 29):**Pin utilizado para el reseteado del LCD (RS).

#### ❖ Puerto D

**PD0 (pin 14):** Se utilizó para la conexión Rx del módulo RX TX.

**PD2 (pin 16):** Se utilizó para la recepción de las señales del módulo GSM.

**PD3 (pin 17):** Se utilizó para la transmisión de las señales del módulo GSM

**PD7 (pin 21):** Se utilizó para el terminal de la chicharra

# PIN	PIN	FUNCIÓN	DESCRIPCIÓN
1	B0	TECLADO	FILA1
2	B1		FILA2
3	B2		FILA3
4	B3		FILA4
5	B4		COLUMNA 1
6	B5		COLUMNA 2
7	B6		COLUMNA 3
8	B7		Pulsador Panico
9	RS		
10	VCC		
11	GND		
12	XT1		OSCILADOR
13	XT2		
14	D0		RX DE MODULO RXTX
16	D2		RX DE MODULO GSM
17	D3		TX DE MODULO GSM
21	D7		CHICHARRA
24	C2	LCD	D7
25	C3		D6
26	C4		D5
27	C5		D4
28	C6		E
29	C7		RS
30	AVCC		
31	GND		
32	AREF		
36	A4	CONTROL DE ILUNINACIÓN	Rele q controla Foco 4
37	A3		Rele q controla Foco 3
38	A2		Rele q controla Foco 2
39	A1		Rele q controla Foco 1
40	A0		SIRENA

**Figura 2. 4:** Distribución de pines del micro controlador ATmega164P

A continuación se describe los pines principales del bloque de control (A):

### Sirena

Este terminal está conectado al microcontrolador mediante el ULN2003A para amplificar la señal de voltaje al relé y su activación se realiza en paralelo con la activación del sistema como se observa en la Figura N°2.5.

Sirena PA0 pin 40

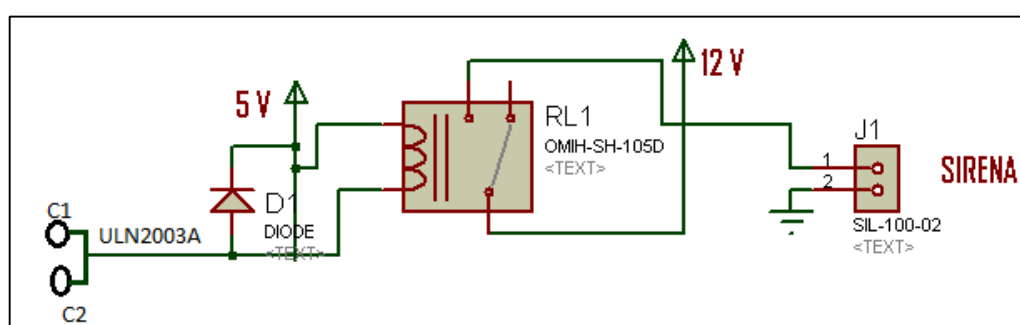


Figura 2. 5: Diagrama circuital de la sirena

### Chicharra

Este terminal se utiliza para la chicharra, la cual funciona cuando se activa o se desactiva manualmente con el teclado la alarma para dar aviso al usuario de que el sistema está funcionando correctamente.

Chicharra PD7 Pin 21

### Botón de Pánico

Esta salida se utiliza como un complemento a nuestro sistema de seguridad, el cual se encuentra en el interior de la casa, siendo su ubicación solo conocida por sus propietarios. El botón de pánico está ubicado al lado de la puerta principal de la maqueta, para la simulación.

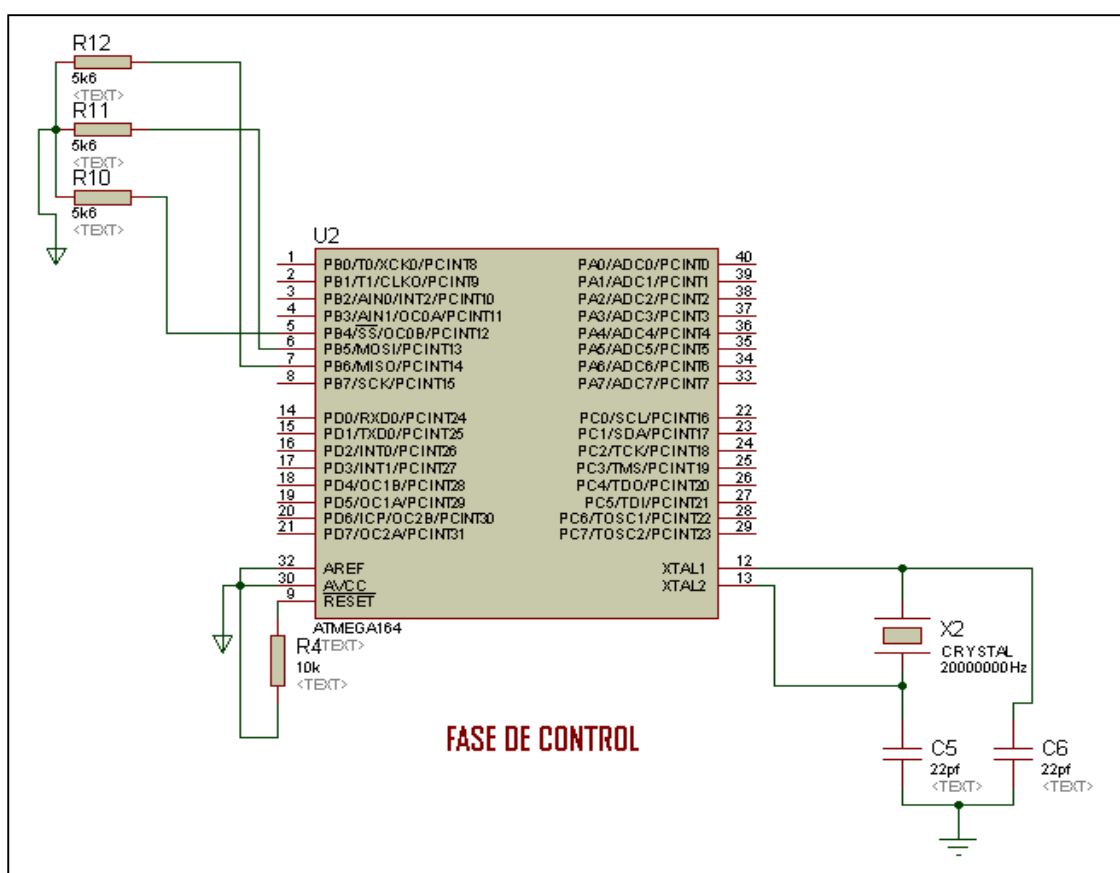
B.pánico PB7 Pin 8

## Oscilador

Este elemento es importante ya que es el motor del microcontrolador que no debe faltar. El oscilador se conecta mediante un cristal de cuarzo de 20 MHz y condensadores que se recomienda en la datasheet.

XTAL 1 Pin 13

XTAL 2 Pin 12



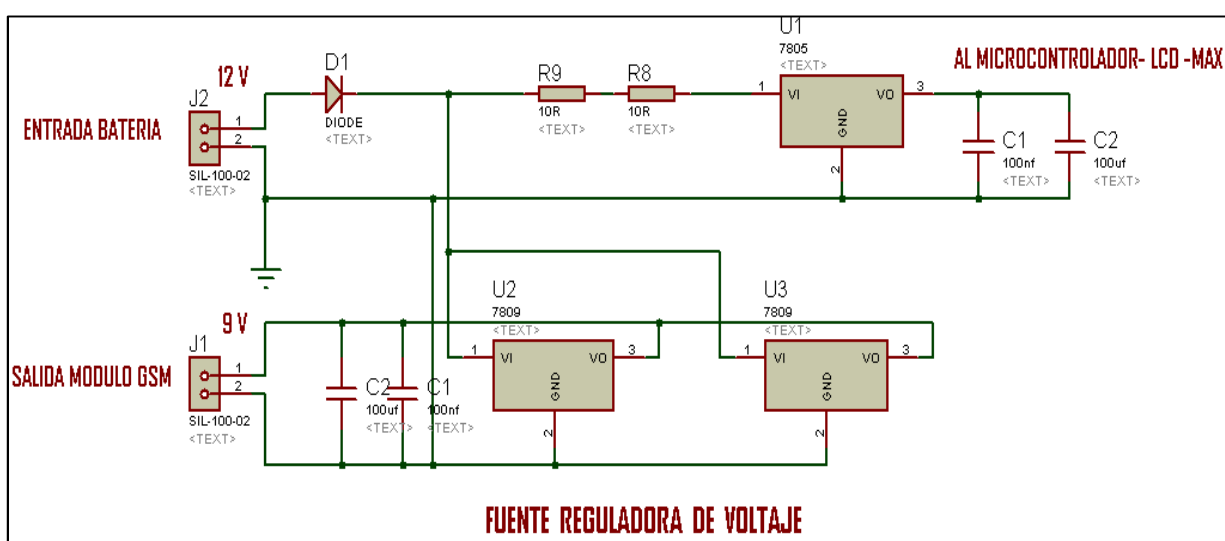
**Figura 2. 6:** Configuración del microcontrolador ATmega 164P

### 2.2.1.2 Bloque de la Fuente Regulada de Voltaje (B)

El sistema de seguridad está diseñado para que funcione con una fuente regulada de 12 V. La entrada de 12 V pasa por un diodo que se polariza de tal forma que circule solo en un solo sentido, para a continuación pasar por resistencias de protección para ingresar a un regulador de voltaje de 5 V fijos y 100mA (78LM05) que con

ayuda de los condensadores obtenemos una señal continua de 5V, que energiza al microcontrolador ATmega 164P, LCD y MAX 232.

Para el funcionamiento del módulo GSM 3006 ZTE se necesita una fuente fija de 9 V, para la cual se obtiene colocando a la salida del diodo un regulador de voltaje de 9V fijos y 5mA (LM 7809) que pasan por condensadores de 100µf obteniendo el voltaje deseado.



**Figura 2. 7:** Diagrama circuital fuente de poder

### 2.2.1.3 Teclado de 5 dígitos para controlar el sistema de seguridad y LCD para visualizar (C).

En este bloque se describe el diagrama circuital del microcontrolador ATmega164P y elementos como el LCD, teclado, MAX232 para el diseño del sistema de seguridad.

Las 4 filas y 3 columnas del teclado están conectadas al microcontrolador ATmega164P mediante 7 líneas de entrada y salida distribuidas de la siguiente forma:

**Tabla 2. 2:** Distribución de pines y puertos para el teclado

Fila 1	Pin 1	PB0
Fila 2	Pin 2	PB1
Fila 3	Pin 3	PB2
Fila 4	Pin 4	PB3
Columna 1	Pin 5	PB4
Columna 2	Pin 6	PB5
Columna 3	Pin 7	PB6

El LCD 16x2 está conectado al microcontrolador a los pines de salida [24-27] mediante un bus de datos de 8 bits, un pin de habilitación E, y un pin de selección que indica que si lo que se está enviado por el bus es un dato o una instrucción (RS), además de un pin que indica si se va a leer o escribir en el módulo. A continuación se describe una tabla a detalle de los pines mencionados.

**Tabla 2. 3:** Distribución de pines del LCD

<b>Pines</b>	<b>Nombre</b>	<b>Función</b>
1	VSS	GND (Tierra 0V)
2	VDD	Alimentación +5 V
3	VEE	Ajuste de contraste del LCD
4	RS	Selección Dato/Control
5	RW	Escritura o lectura del LCD
6	E	Habilitación del LCD
7	D0	D0 Bit menos significativo
8	D1	D1
9	D2	D2
10	D3	D3
11	D4	D4
12	D5	D5
13	D6	D6
14	D7	D7 Bit más significativo
15	A	Ánodo del LED (Luz de fondo)
16	K	Cátodo del LED (Luz de fondo)



En esta etapa se utilizó el MAX 232, que soluciona la conexión necesaria para lograr la comunicación entre el puerto serie de una PC y módulo GSM al microcontrolador ATmega164P en base a señales de nivel TTL/CMOS.

Los pines 16,17 de salida del microcontrolador están conectadas al MAX 232 a los pines de entrada de Tx y Rx, para que me permitan obtener niveles lógicos para poder comunicarse con el Módulo GSM.

Mientras que el módulo RX va conectado al pin 14 del microcontrolador únicamente para recepción de señales como se muestra en la Figura N° 2.8.

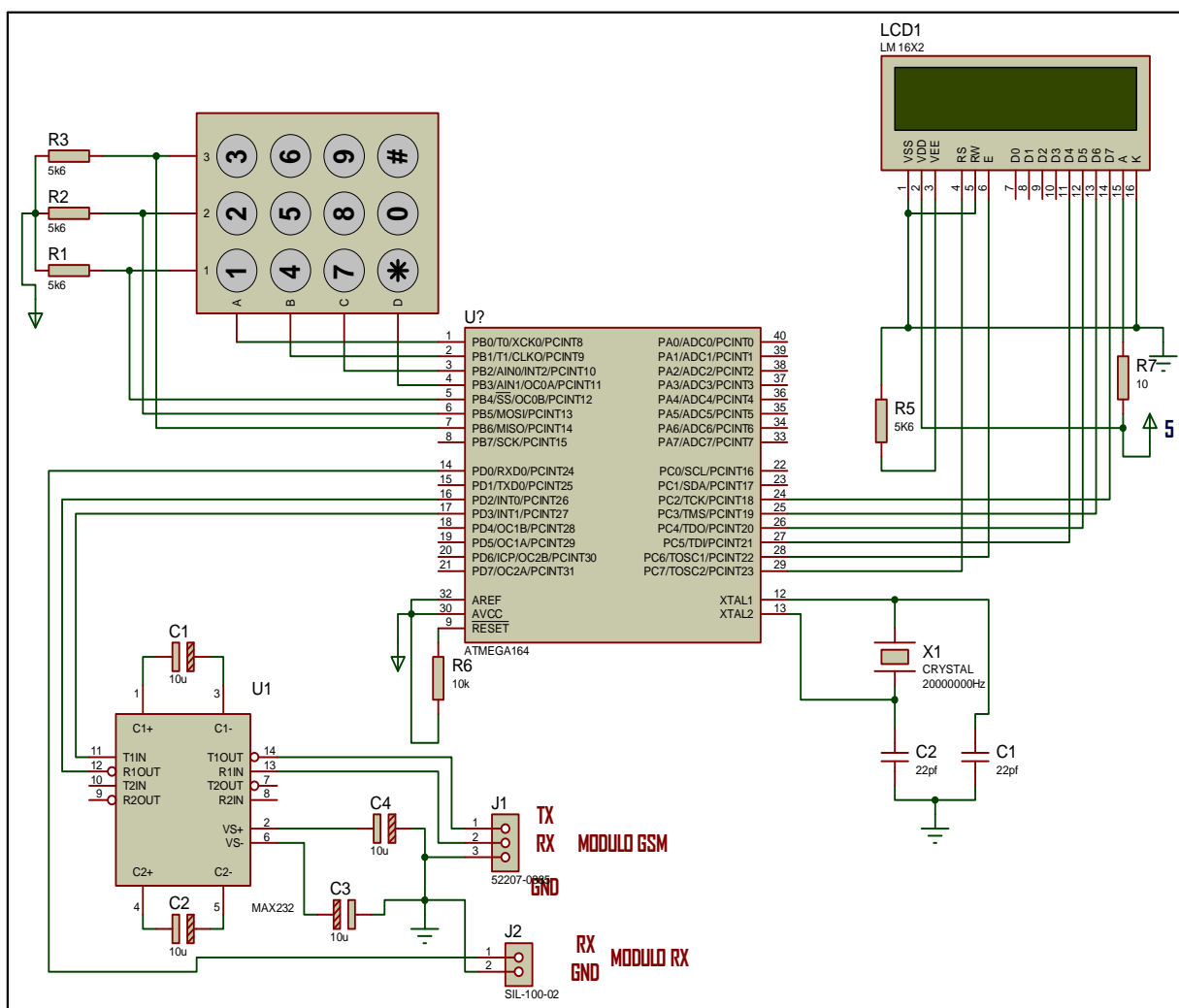


Figura 2. 8: Diagrama circuital del LCD y teclado

### 2.2.1.4 Control de Iluminación para la vivienda (C)

Nuestro sistema de seguridad también cuenta con un control de iluminación, que permite activar y desactivar 4 focos de manera independiente con ayuda del integrado ULN2003A, que mediante sus pines de salida C3, C4, C5, C6, realiza la función de conmutar de manera individual cada relé que está conectado al foco1, foco2, foco3, foco4 respectivamente como se ve en la Figura N°2.9.

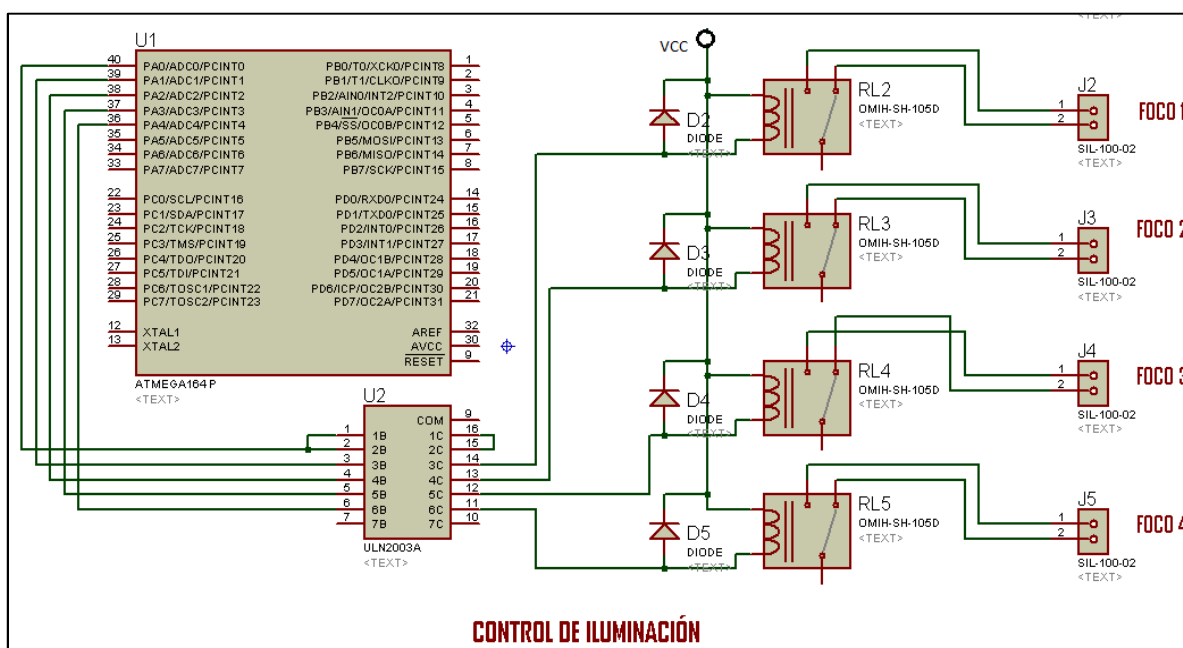
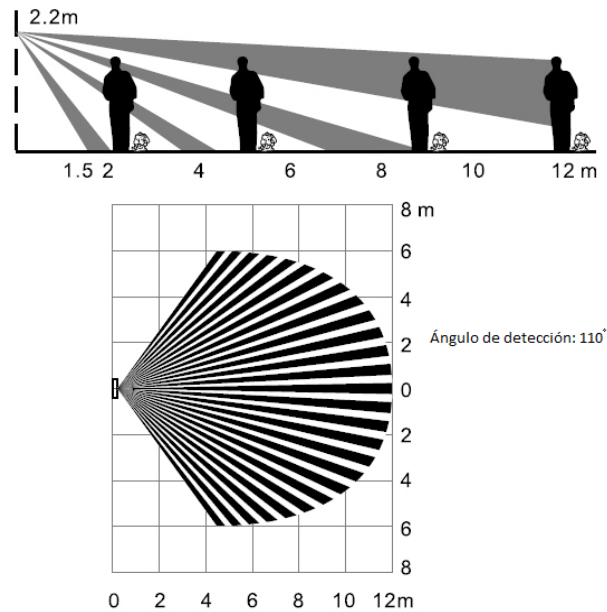


Figura 2. 9: Diagrama circuital del control de Iluminación

### 2.2.1.5 Detección de movimiento en el domicilio (D)

Como se indica en la Figura N° 2.10, en el monitoreo del domicilio utilizamos un sensor inalámbrico de movimiento y uno magnético para apertura, de puertas los que trabajan a una frecuencia de 430 MHz y cuentan con un direccionamiento de jumpers que se encuentran en su interior, los cuales indican la dirección de la posición en la que se va a colocar.

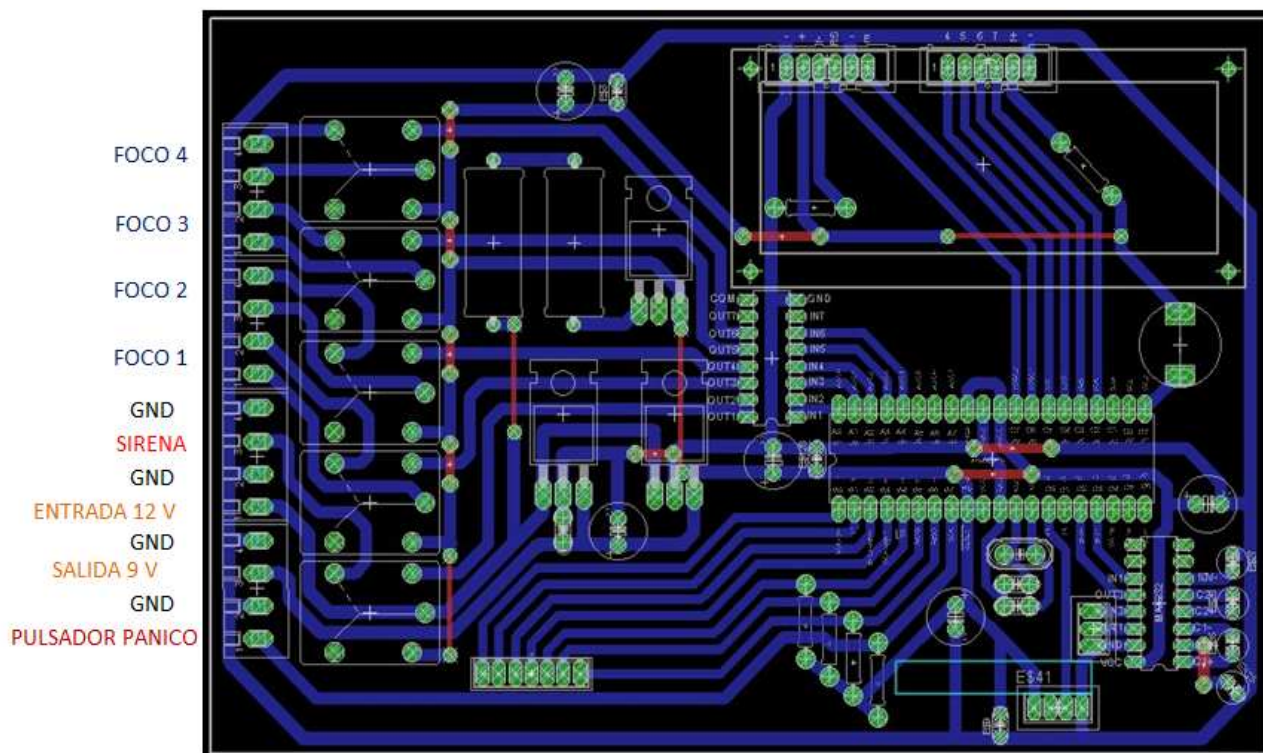


**Figura 2. 10:** Alcance del sensor PIR

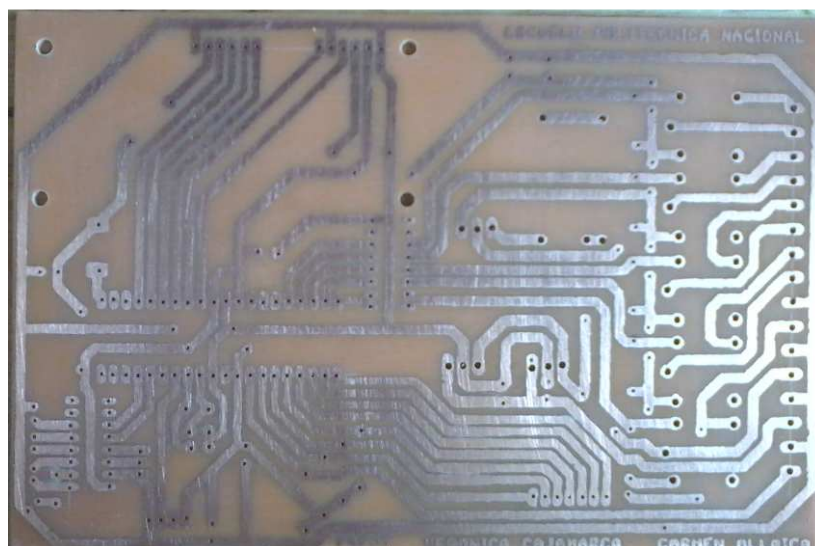
### 2.3 PLACA IMPRESA DEL SISTEMA DE SEGURIDAD

Pasos para la construcción de la placa Figura N°2. 11

- ❖ Realizar el circuito en PROTEUS v7.0 con todos los elementos requeridos.
- ❖ Armado del circuito en protoboard
- ❖ Ruteo de pistas en EAGLE v5.0
- ❖ Impresión del circuito en papel térmico
- ❖ Realizar la transferencia térmica del diseño a la baquelita
- ❖ Colocar la placa en ácido sulfúrico
- ❖ Limpiar la baquelita con diluyente thinner y soldar los elementos.



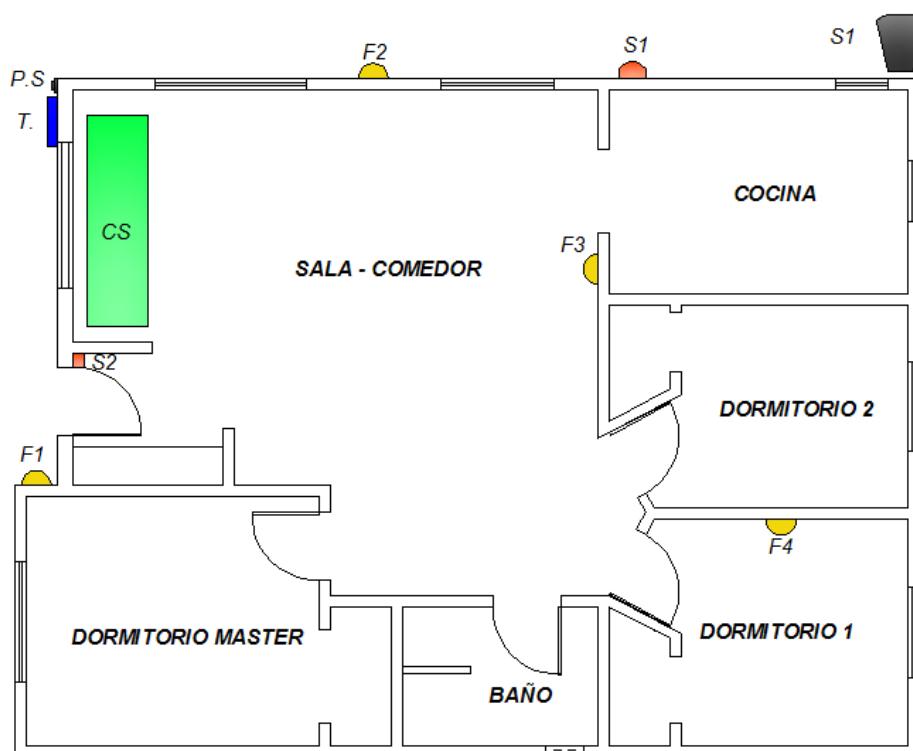
**Figura 2. 11:** Diseño de la placa del sistema de seguridad en EAGLE v5.0



**Figura 2. 12:** Diseño de la placa en baquelita

## 2.4 DISEÑO DE LA MAQUETA PARA SIMULACIÓN DE UN DOMICILIO.

El presente proyecto simula una opción dentro de los sistemas de seguridad en una vivienda; para lo cual se realizó el plano de la maqueta que se muestra en la Figura N°2.12.



SIMBOLOGÍA	
CS	CEREBRO DEL SISTEMA Y LCD
S1	SENSOR INALAMBRICO DE MOVIMIENTO
S2	SENSOR MAGNETICO INALAMBRICO DE APERTURA
T.	TECLADO DEL SISTEMA
S1	SIRENA
P.S.	PULSADOR DE PÁNICO
F4	FOCO 1, FOCO 2, FOCO 3, FOCO 4

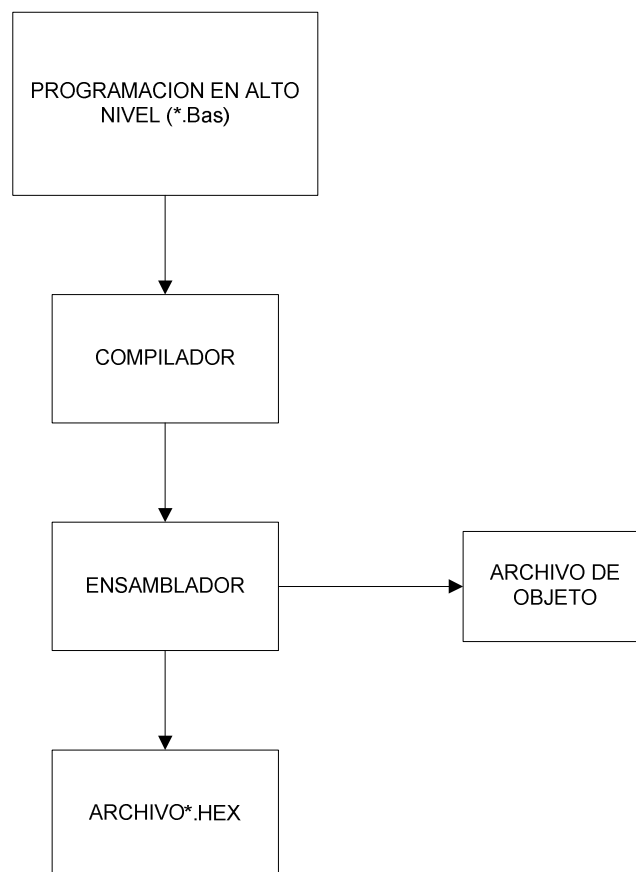
**Figura 2. 13:** Plano de la Vivienda

## CAPITULO III

### 3 DESARROLLO DEL SOFTWARE PARA EL SISTEMA DE SEGURIDAD

#### 3.1 LENGUAJE DE PROGRAMACIÓN BASCOM AVR<sup>20</sup>

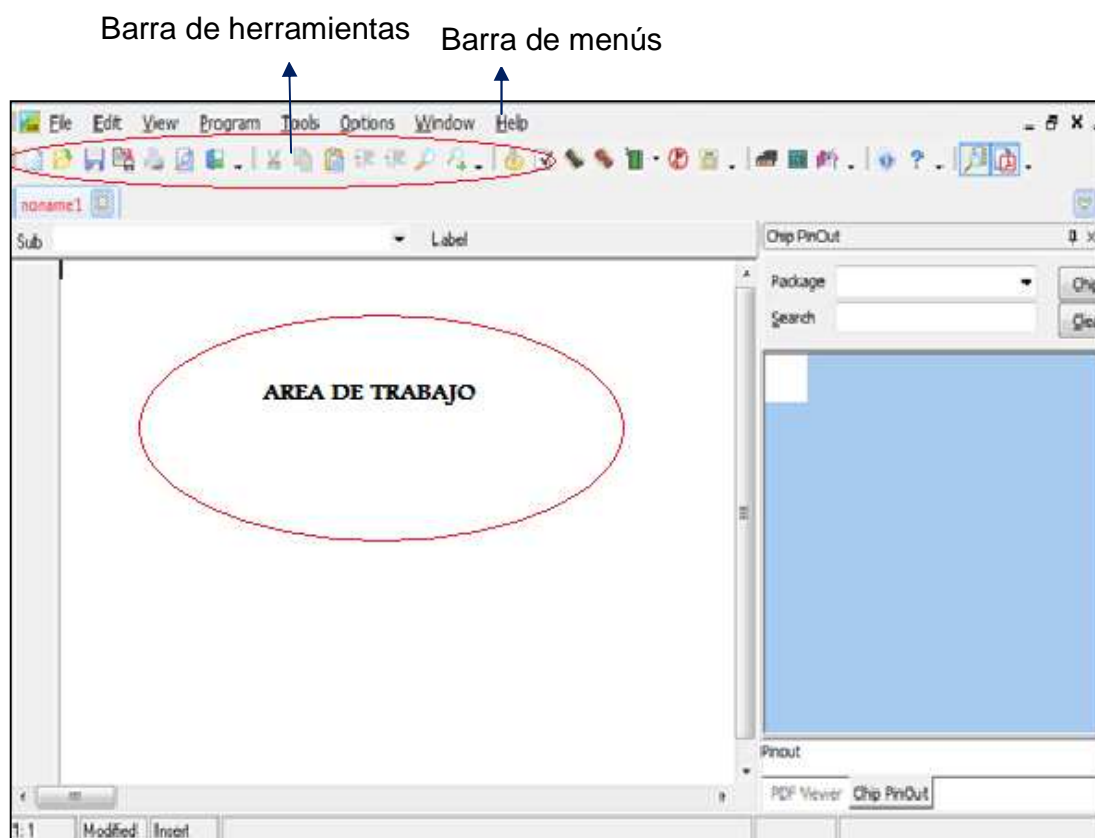
La herramienta BASCOM AVR desarrollada por la empresa MCS Electronics, sirve para realizar programas de alto nivel para microcontroladores AVR, el cual posee un compilador y un ensamblador que traduce las instrucciones estructuradas en lenguaje de máquina.



**Figura 3. 1:** Diagrama de flujo de programación estructurada

<sup>20</sup> <http://bibdigital.epn.edu.ec/bitstream/15000/3835/1/CD-3618.pdf>

Luego de instalar el paquete computacional, el cual se puede conseguir como DEMO en la página del MCS Electronics, podemos apreciar la siguiente pantalla inicial.



**Figura 3. 2:** Área de trabajo del BASCOM AVR

Aquí podemos visualizar una barra de herramientas, un menú y el área de trabajo. A continuación se explicara los iconos o caminos más importantes para manejar la herramienta BASCOM AVR.

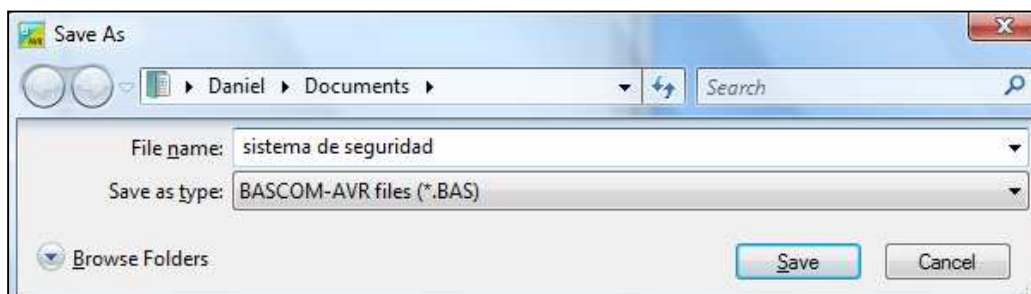
### 3.1.1 INICIO<sup>21</sup>

Presiono NEW, para abrir una nueva hoja de trabajo para el proyecto.

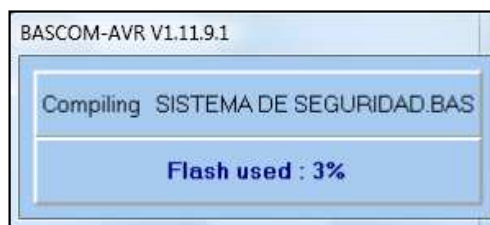
### 3.1.2 COMPILADOR<sup>21</sup>

Presionando el icono de la barra de herramientas o la tecla **F7**, nosotros podemos compilar nuestro proyecto y obtener un archive .HEX, el cual va hacer grabado en el microcontrolador (Figura N° 3.4) y una vez que se ha compilado el proyecto puede

aparecer el siguiente cuadro de confirmación (Figura N° 3.5) e indicarme cual es el porcentaje de memoria en el microcontrolador.



**Figura 3. 3:** Cuadro donde se muestra guardar nuevo archivo



**Figura 3. 4:** Cuadro de confirmación de compilación

### 3.1.3 SIMULADOR<sup>21</sup>

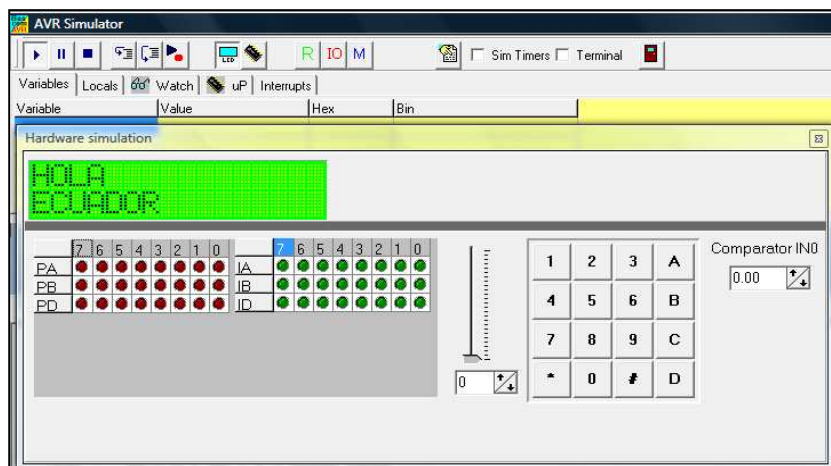
Una vez que se compila un proyecto, se puede simular con ayuda de BASCOM SIM, presionando el icono de simulación de la barra de herramientas o la tecla **F2** como se muestra en la Figura N° 3.5.



**Figura 3. 5:** Icono del simulador del programa

Una vez que se presiona el icono de simulación, aparece una pantalla donde se puede apreciar el programa principal, espacios de memoria, emuladores de comunicación serial, emuladores de LCD, entre otros.



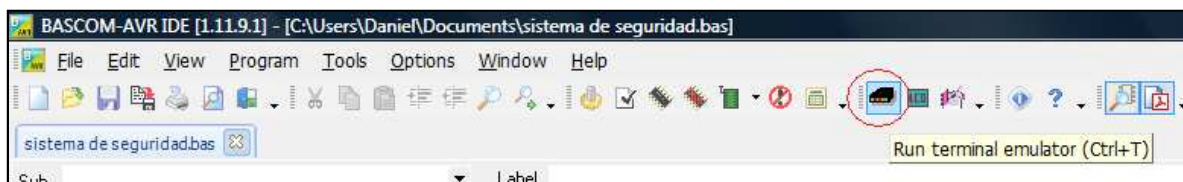


**Figura 3. 6:** Cuadro de simulación del BASCOM AVR

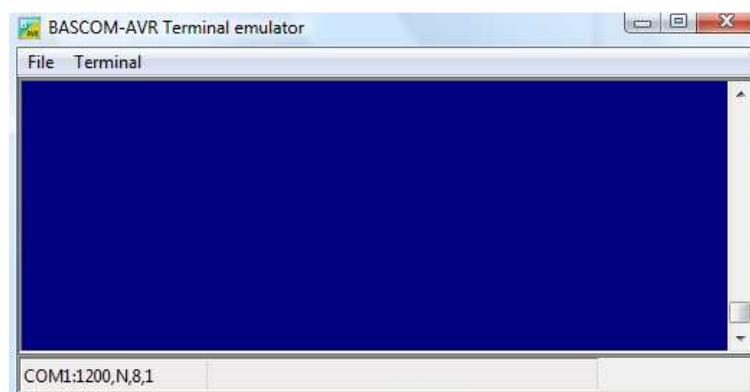
A medida que se va realizando el proceso debe verificarse paso a paso con la tecla **F8** que permite pausar y visualizar el avance de la simulación.

### 3.1.4 EMULADOR SERIAL<sup>21</sup>

Este icono se usó como emulador de comunicación serial entre el microcontrolador ATmega164P y la computadora, como se muestra en la Figura N°3.7



**Figura 3. 7:** Icono emulador serial del BASCOM AVR



**Figura 3. 8:** Emulador de comunicación serial

Una vez seleccionado el icono de emulador serial, se efectúa la simulación del programa. Este emulador es un programa informático que simula el funcionamiento de un terminal de computadora en cualquier dispositivo de visualización en nuestro caso un LCD.

### 3.2 COMANDOS AT<sup>21</sup>

Los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un Terminal MODEM.

Los comandos AT fueron desarrollados en 1977 por Dennis Hayes como un interfaz de comunicación con un MODEM para así poder configurarlo y proporcionarle instrucciones, tales como marcar un número de teléfono. Más adelante, con el avance del baudio, fueron las compañías Microcom y US Robotics las que siguieron desarrollando y expandiendo el juego de comandos hasta universalizarlo.

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, permiten 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 terminal.

---

<sup>21</sup>[http://www.google.com.ec/webhp?source=search\\_app#hl=es&tbo=d&sclient=psy-ab&q=comandos+AT&oq=comandos+AT&gs\\_l=serp.3..0l4.3252.3560.7.4174.2.2.0.0.0.194.194.0j1.1.0...0.0...1c.1.W7wduqyayk&pbx=1&bav=on.2,or.r\\_gc.r\\_pw.r\\_qf.&fp=c97bc84180386383&bpcl=39650382&biw=1236&bih=584](http://www.google.com.ec/webhp?source=search_app#hl=es&tbo=d&sclient=psy-ab&q=comandos+AT&oq=comandos+AT&gs_l=serp.3..0l4.3252.3560.7.4174.2.2.0.0.0.194.194.0j1.1.0...0.0...1c.1.W7wduqyayk&pbx=1&bav=on.2,or.r_gc.r_pw.r_qf.&fp=c97bc84180386383&bpcl=39650382&biw=1236&bih=584)

Es claro que la implementación de los comandos AT corresponde a los dispositivos GSM y no depende del canal de comunicación a través del cual estos comandos sean enviados, ya sea cable de serie, canal Infrarrojos entre otros.

### 3.2.1 NOTACIÓN DE LOS COMANDOS AT<sup>22</sup>

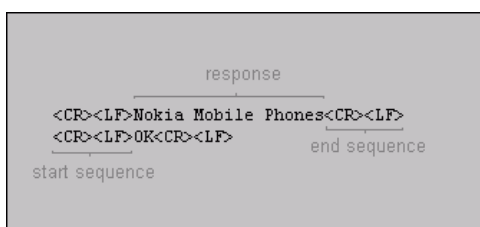
El envío de comandos AT requiere la siguiente estructura:

#### ❖ Petición



<CR> ... Carriage return

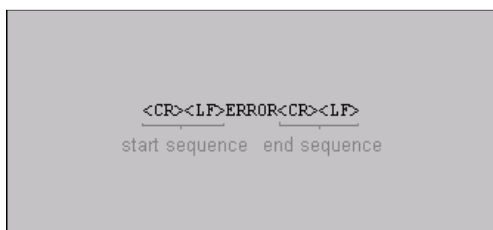
#### ❖ Respuesta correcta



<CR> ... Carriage returns

<LF> ... Line feed

#### ❖ Respuesta incorrecta



<CR> ... Carriage return

<LF> ... Line feed

### 3.2.2 COMANDOS AT MÁS UTILIZADOS<sup>22</sup>

❖ ATE:

- Eco

Los comandos introducidos en el modem vuelven por eco al PC (por defecto).

❖ ATL: Programa el volumen del altavoz

❖ ATM : Programa conexión/desconexión del altavoz

❖ ATO:

- Vuelve a estado on-line desde el estado de comandos.

Permite retomar una conexión ya en marcha.

❖ ATQ: Programa los códigos de resultado a ON/OFF.

❖ ATV: Envía códigos de resultado en palabras o números

❖ ATZ: Reset

❖ AT&K: Programa control de flujo

❖ AT&W: Almacena perfil de configuración del usuario.

❖ AT&Y: Especifica que perfil de configuración usuario de los almacenados se va a utilizar.

### **3.2.3 COMANDOS GENERALES<sup>22</sup>**

AT+CGMI: Identificación del fabricante

AT+CGSN: Obtener número de serie

AT+CIMI: Obtener el IMSI.

AT+CPAS: Leer estado del modem o del módulo GSM.

### **3.2.4 COMANDOS DEL SERVICIO DE RED<sup>22</sup>**

AT+CSQ: Obtener calidad de la señal

AT+COPS: Selección de un operador

AT+CREG: Registrarse en una red

AT+WOPN: Leer nombre del operador

### **3.2.5 COMANDOS DE SEGURIDAD<sup>22</sup>**

AT+CPIN: Introducir el PIN

AT+CPINC: Obtener el número de reintentos que quedan

AT+CPWD: Cambiar password.

### **3.2.6 COMANDOS PARA LA AGENDA DE TELÉFONOS<sup>22</sup>**

AT+CPBR: Leer todas las entradas

AT+CPBF: Encontrar una entrada

AT+CPBW: Almacenar una entrada

AT+CPBS: Buscar una entrada

### **3.2.7 COMANDOS PARA SMS<sup>22</sup>**

AT+CPMS: Seleccionar lugar de almacenamiento de los SMS

AT+CMGF: Seleccionar formato de los mensajes SMS

AT+CMGR: Leer un mensaje SMS almacenado

AT+CMGL: Listar los mensajes almacenados

AT+CMGS: Enviar mensaje SMS

AT+CMGW: Almacenar mensaje en memoria

AT+CMSS: Enviar mensaje almacenado

AT+CSCA: Establecer el Centro de mensajes a usar

AT+WMSC: Modificar el estado de un mensaje

### **3.3 GRABADOR DEL MICROCONTROLADOR**

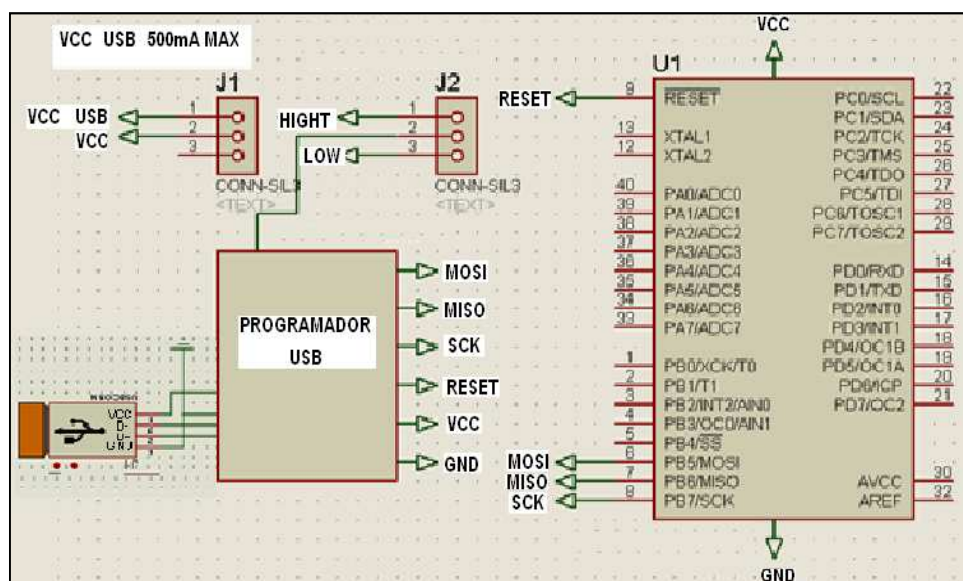
#### **3.3.1 PROGISP 172**

Cuando se compila un programa en Bascom AVR se crea el archivo hexadecimal .hex, con todas las instrucciones que el microcontrolador necesita para funcionar correctamente, este archivo se guarda en el microcontrolador utilizando el grabador Progisp 172.

Características del grabador Progisp 172:

- ❖ Su comunicación con el computador es mediante un puerto USB
- ❖ Posee un jumper que permite alimentar al microcontrolador con el voltaje del computador o con una fuente externa.
- ❖ Tiene un jumper que elige la velocidad de grabación.
- ❖ Para guardar el archivo hexadecimal el grabador posee 6 pines de conexión

En la Figura N° 3.9 se observa los pines de conexión entre grabador y el microcontrolador.



**Figura 3. 9:** Diagrama circuital del programador USB y Progisp172

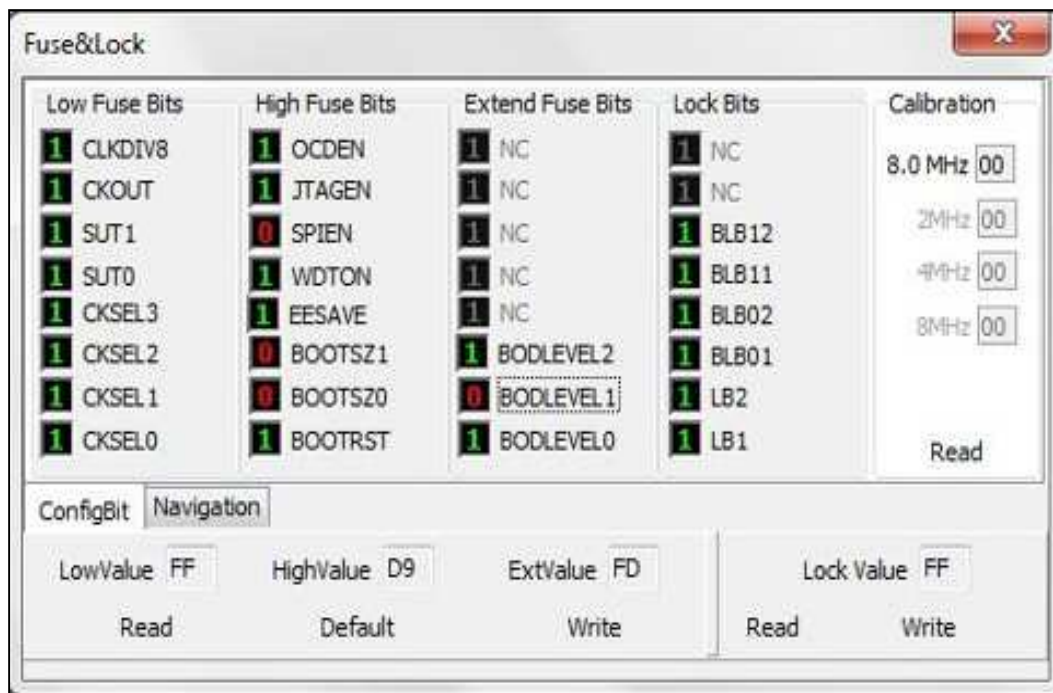
Este grabador posee un software amigable permitiéndonos seleccionar al microcontrolador a grabar, y sus respectivos “fuse bits”, que permiten:

- ❖ Definir si el oscilador es interno o externo
- ❖ El oscilador con el que trabaja el microcontrolador
- ❖ Seleccionar el nivel de voltaje con el que se va a trabajar

Además le da opciones para la programación como:

- ❖ Leer el contenido del microcontrolador
- ❖ Borrar el contenido del microcontrolador
- ❖ Verificar si la grabación se realizó correctamente
- ❖ Cargar automáticamente el archivo hexadecimal
- ❖ Proteger al archivo hexadecimal contra lectura

En la Figura N° 3.10 muestra las configuraciones con el software de los “fuse bits”, sus opciones de programación en un microcontrolador. En la configuración el microcontrolador se le asigna un oscilador externo, mediante la selección de los casilleros CKSELK y se ha protegido el código del chip contra lectura por medio del uso de Lock Chip ubicado en las opciones de programación.



**Figura 3. 10:** Pantalla de grabación del programador USB

## 3.4 PRINCIPALES SENTENCIAS UTILIZADAS EN EL PROYECTO

### 3.4.1 INSTRUCCIONES INICIALES<sup>21</sup>

Estas instrucciones son utilizadas al inicio de cada programa para determinar de la mejor manera las características del integrado a utilizarse, algunas de las siguientes son mencionadas a continuación:

#### 3.4.1.1 \$regfile

Esta instrucción se encuentra por lo general al inicio de cualquier proyecto, ya que es el encargado de direccionar el respectivo microcontrolador que se va a utilizar.

**Ejemplo:**

ATMEGA 16 → **\$regfile="m16def.dat"**

ATMEGA 8 → **\$regfile="m8def.dat"**



### 3.4.1.2 \$crystal

La siguiente instrucción especifica la frecuencia de oscilación con la que trabaja el microcontrolador.

#### Ejemplo:

\$crystal = 20000000      para 20 MHz

### 3.4.1.3 \$Baud

Esta instrucción permite determinar el valor en baudios con la que está trabajando el microcontrolador.

\$baud=9600  
\$baud1 = 115200+

## 3.4.2 CONFIGURACIONES INICIALES<sup>21</sup>

En esta etapa del programa se inicializa un pin o un grupo de pines para que realicen una función específica.

### 3.4.2.1 Config

Esta instrucción especifica la configuración de un pin, un puerto o dispositivo que se los configura como de entrada o salida.

#### Ejemplo:

**Config** portb = input      Declara todo el puerto B como entrada

**Config** portd.0 = output      Pin D.0 configurada como salida

### 3.4.2.2 DDRx, PORTx

La instrucción se la utiliza para declarar las variables que se utilizan en el programa, además son registros que nos permiten usar un puerto como de entrada salida. También es importante tener en cuenta que cuando se configura un puerto

como de salida se debe ocupar la palabra PORT y si es de salida se utiliza la palabra PIN.A continuación se muestra las combinaciones, que hacen que los pines funcionen en configuración especial:

Ddrb.x = 0 Entrada de alta impedancia

Portb.x = 0

Ddrb.x =0 Entrada pull up

Portb.x = 1

Ddrb.x = 1 Salida a cero (0L) 20 mA

Portb.x = 0

### **Ejemplo:**

Ddrb.0 = 1: Portb.0 = 0: Fila1 Alias Portb.0      puerto definido como salida

Ddrb.1 = 1: Portb.1 = 0: Fila2 Alias Portb.1      puerto definido como entrada

### **3.4.2.3 Alias**

Se utiliza para dar nombre específico a un pin o un puerto en un proyecto, ya que facilita el uso de un in porque es más fácil recordar la función que realiza el puerto, que su nombre.

### **Ejemplo:**

Ddra.4 = 1: Porta.4 = 0: Foco\_4 **Alias** Porta.4      el Porta.4 se llama foco

### **3.4.3 TIPO DE DATOS<sup>21</sup>**

Los datos ingresados en un programa son declarados según el tipo de variable a utilizar, ya que estos nos permiten realizar cálculos con dichas variables, evitando errores de dimensionamiento.

### 3.4.3.1 On Urxc

Esta instrucción permite habilitar la interrupción serial de la subrutina.

On Urxc Recepcion\_rx  
Enable Urxc

### 3.4.3.2 Dim

Instrucción que dimensiona el tipo de variable a utilizar, en la Tabla N° 3.1 se indica los tipos de variable que se utilizan en BASCOM AVR.

**Tabla 3. 1:** Tipos de variables

TIPO	DIMENSIÓN
Bit	0 - 1
Byte	0 a 255
Word	0 a 65535
Long	-2147483648 a 2147483647
Integer	-32768 a 32767
Single	$1.5 \times 10^{-45}$ a $3.4 \times 10^{38}$
String	Cadena de caracteres máximo 254
Double	$5.0 \times 10^{324}$ a $1.7 \times 10^{308}$

**Ejemplo:**

**Dim** Cont\_rxtx **As** Byte

## 3.4.4 ADMINISTRACIÓN DE BITS<sup>21</sup>

### 3.4.4.1 Toggle

El siguiente comando sirve para complementar el estado anterior de alguna variable o pin de algún puerto.

**Ejemplo:**

**Toggle** Band\_on\_off\_alarma

**3.4.5 ADMINISTRACIÓN DE STRINGS<sup>21</sup>****3.4.5.1 INSTR**

Retorna la posición de una substring en una string, en nuestro proyecto el **Instr** realiza la función de almacenamiento de datos en el Buffer en la transferencia de datos y activa la alarma.

**Ejemplo:**

Msj\_activa = **Instr** (1 , Buffer\_telf , "Activa")

**3.4.5.2 Mid**

La función MID retorna parte de una string a una sub string. Cumple la función de MID es la de reemplazar parte de una variable de una string por otra string.

Hay que tener muy en cuenta que la posición inicial de una string siempre se encuentra en 1

**Ejemplo:**

**Mid**(clave\_lcd\_aux , Cont\_clave , 1 ) = " \* "

**3.4.6 INSTRUCCIONES DE USO GENERAL<sup>21</sup>****3.4.6.1 Wait**

Esta instrucción me permite crear un retardo, ya sea este en segundos, milisegundos y microsegundos respectivamente.

**Ejemplo:**

Waitms 1	Espera 1 segundos
Waitms 100	Espera 100milisegundos

**3.4.6.2 Incr**

Incr nos permite incrementa el valor de una variable.

**Ejemplo:**

**Incr** Contador

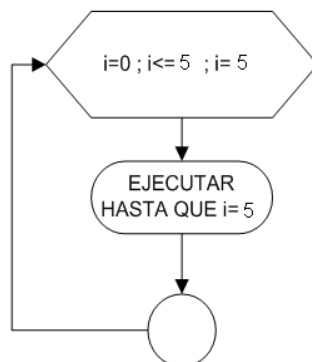
**Incr** Cont\_chicharra

**3.4.6.3 Estructura de las sentencias****3.4.6.4 Do – Loop**

Esta sentencia establece un lazo cerrado, mediante el cual se ejecutan un conjunto de instrucciones de forma indefinida.

**3.4.6.5 For-Next**

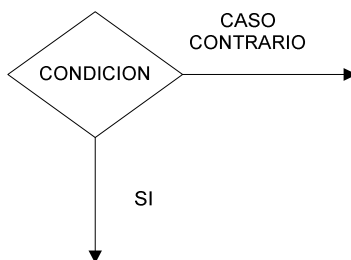
La siguiente sentencia es de repetición, en la cual se ejecutan un grupo de instrucciones hasta cumplir con la condición que finaliza el lazo, está condición de terminación está dada por una variable que se incrementa o se decrementa en pasos previamente establecidos como se observa en la Figura N°3.11.



**Figura 3. 11:** Ejemplo de la condición de repetición FOR-NEXT<sup>20</sup>

### 3.4.6.6 If-Then-Else

Estas sentencias condicionales, permiten condicionar la ejecución de instrucciones, establecidos en la evaluación entre dos o más variables con la ayuda de operadores lógicos como se muestra en la Figura N°3.12



**Figura 3. 12:** Ejemplo de la condiciónlógica IF-ELSE

### 3.4.6.7 Gosub

Esta sentencia obliga al programa a realizar un salto a una subrutina, donde se ejecutan las instrucciones definidas para luego retornar y continuar con el programa.

**Ejemplo:**

**Do**

**Gosub** Teclado

IfNúmero\_tecla<> 12 Then

If Band\_teclado = 1 Then Gosub On\_off\_alarma

If Band\_teclado = 0 Then Gosub Cambiar\_key\_alarma

**End If**

## 3.4.7 CONFIGURACIÓN MEDIANTE COMANDOS<sup>21</sup>

### 3.4.7.1 Config Com1, Com2

La siguiente instrucción nos permite la configuración de una comunicación serial.

**Ejemplo:**

**Config** Com1 = Dummy, Synchron = 0, Parity = None, Stopbits = 1, Databits = 8, Clockpol = 0

### 3.4.7.2 Config Lcdpin

Por medio de esta instrucción nos permite la configuración de todos los puertos del LCD.

#### Ejemplo:

Config Lcdpin = Pin, Db4 = Portc.5, Db5 = Portc.4, Db6 = Portc.3, Db7 = Portc.2, E = Portc.6, Rs = Portc.7

ConfigLcd = 16 \* 2  
Initlcd

## 3.5 PROGRAMA PRINCIPAL PARA EL SISTEMA DE SEGURIDAD

Para iniciar el programa se indica que tipo de microcontrolador se va a utilizar, frecuencia de cristal, se habilita los timers y las interrupciones.

\$regfile = "m164pdef.dat"  
\$crystal = 2000000

\$baud = 9600  
\$baud1 = 115200

Config Com1 = Dummy, Synchron = 0, Parity = None, Stopbits = 1, Databits = 8, Clockpol = 0  
Config Com2 = Dummy, Synchron = 0, Parity = None, Stopbits = 1, Databits = 8, Clockpol = 0

Open "com2:115200,8,n,1" For Binary As #1 **' crear puerto serial virtual**  
Config Lcdpin = Pin , Db4 = Portc.5 , Db5 = Portc.4 , Db6 = Portc.3 , Db7 = Portc.2 , E = Portc.6 , Rs = Portc.7

Config Lcd = 16 \* 2  
Initlcd  
Cls : Cursor Off Noblink

\*\*\*\*\*

## VARIABLES PARA TECLADO

Ddrb.0 = 1 : Portb.0 = 0 : Fila1 Alias Portb.0  
 Ddrb.1 = 1 : Portb.1 = 0 : Fila2 Alias Portb.1  
 Ddrb.2 = 1 : Portb.2 = 0 : Fila3 Alias Portb.2  
 Ddrb.3 = 1 : Portb.3 = 0 : Fila4 Alias Portb.3

Ddrb.4 = 0 : Portb.4 = 1 : Columna1 Alias Pinb.4  
 Ddrb.5 = 0 : Portb.5 = 1 : Columna2 Alias Pinb.5  
 Ddrb.6 = 0 : Portb.6 = 1 : Columna3 Alias Pinb.6

### 'Entradas de teclado

Ddrb.7 = 0 : Portb.7 = 1 : Pul\_panico Alias Pinb.7

### 'Salidas de teclado para sirena y chicharra

Ddrd.7 = 1 : Portd.7 = 1 : Chicharra Alias Portd.7

Ddra.0 = 1 : Porta.0 = 0 : Sirena Alias Porta.0

Ddra.1 = 1 : Porta.1 = 0 : Foco\_1 Alias Porta.1  
 Ddra.2 = 1 : Porta.2 = 0 : Foco\_2 Alias Porta.2  
 Ddra.3 = 1 : Porta.3 = 0 : Foco\_3 Alias Porta.3  
 Ddra.4 = 1 : Porta.4 = 0 : Foco\_4 Alias Porta.4.

\*\*\*\*\*

## VARIABLES EEPROM (ATmega164P)

Dim Null\_0 As Eram Byte  
 Dim Key\_alarma\_guardada As Eram String \* 5

Const Buffer\_size\_telf = 61  
 Const Buffer\_size\_rtx = 20  
 Const Tiempo\_miliseg = 20000  
 Const Zise\_num\_telf = 10  
 Dim Num\_telf As String \* Zise\_num\_telf



Dim Buffer\_txrx As String \* Buffer\_size\_rtx  
 Dim Buffer\_telf As String \* Buffer\_size\_telf

Dim Band\_teclado As Bit  
 Dim Band\_on\_off\_alarma As Bit  
 Dim Band\_chek\_sensores As Bit  
 Dim Band\_cont\_msj As Bit

Dim Cont\_rtx As Byte  
 Dim Cont\_telf As Byte  
 Dim Número\_tecla As Byte  
 Dim Msj\_alarm\_activada As Byte  
 Dim Msj\_activa As Byte  
 Dim Msj\_desactiva As Byte  
 Dim Cont\_clave As Byte  
 Dim Cambiar\_clave As Byte  
 Dim Cont\_for As Byte  
 Dim Cont\_enviar\_msj As Word  
 Dim Cont\_chicharra As Word

\*\*\*\*\*

### **VARIABLES PARA MARCADO DE NÚMEROS**

Dim Conf\_tel1 As Byte  
 Dim Conf\_tel2 As Byte

\*\*\*\*\*

### **VARIABLES PARA INICIALIZAR EL MÓDULO GSM ZTE**

Dim Modulo\_respuesta As Byte  
 Dim Mod\_res\_error As Byte  
 Dim Contador As Byte

\*\*\*\*\*

### **VARIABLES PARA EL ENCENDIDO Y APAGADO DE LA ILUMINACIÓN**

Dim Msj\_foco1\_on As Byte  
 Dim Msj\_foco1\_off As Byte  
 Dim Msj\_foco2\_on As Byte

```
Dim Msg_foco2_off As Byte
Dim Msg_foco3_on As Byte
Dim Msg_foco3_off As Byte
Dim Msg_foco4_on As Byte
Dim Msg_foco4_off As Byte
```

```
*****
```

### VARIABLES PARA TARJETA SIM

```
Dim Key_alarma_leida As String * 5
Dim Key_nueva As String * 5
Dim Selec_sim_num As String * 9
Dim Clave_armar As String * 5
Dim Clave_lcd_aux As String * 5
```

```
*****
```

### VARIABLE PARA EL CÓDIGO QUE ENVÍA EL SENSOR

```
Dim Codigo_sensor As String * 4
Codigo_sensor = "0000"
```

### SUBROUTINAS PARA EL SISTEMA DE SEGURIDAD

A continuación se muestra las subrutinas para las etapas del software del sistema de seguridad.

```
*****
```

Subrutina serial

```
On Urxc Recepcion_rx
Enable Urxc
```

```
On Urxc1 Módulo
Enable Urxc1
```

```
Enable Interrupts
```

```
Disable Urxc
```

**'Tiempo de espera hasta que se inicialice el módulo GSM**

Do

```
If Contador.0 = 0 Then
  Cls
  Locate 1 , 1 : Lcd " INICIALIZANDO "
  Locate 2 , 1 : Lcd " MÓDULO GSM "
End If
```

```
If Contador.0 = 1 Then Cls
```

```
Incr Contador
Waitms 500
Loop Until Contador > 16
Módulo_respuesta = 0 : Cont_telf = 0 : Buffer_telf = "" : Contador = 0
```

#### **'Configurar velocidad de transmisión**

```
Do
  Print #1, "AT+IPR=115200" ; Chr(13)
  Gosub Confirmacion_mod_telf
  Loop Until Módulo_respuesta > 0
  Módulo_respuesta = 0 : Cont_telf = 0 : Buffer_telf = ""
```

#### **'Configurar para enviar mensaje**

```
Do
  Print #1 , "AT+CMGF=1" ; Chr(13)
  Gosub Confirmacion_mod_telf
  Loop Until Módulo_respuesta > 0
  Módulo_respuesta = 0 : Cont_telf = 0 : Buffer_telf = ""
```

#### **'Configurar para recibir mensaje y enviarlo al puerto serial**

Do

```
Print #1 , "AT+CNMI=3,2,0,0,0" ; Chr(13)
Gosub Confirmacion_mod_telf
Loop Until Módulo_respuesta > 0
Módulo_respuesta = 0 : Cont_telf = 0 : Buffer_telf = ""
```

**'Guarda configuraciones realizadas**

```

Do
  Print #1 , "AT&W" ; Chr(13)
  Gosub Confirmacion_mod_telf
  Loop Until Módulo_respuesta > 0

```

**'Suenan chicharra e indica que se ingresa al programa principal**

```

Chicharra = 0 : Waitms 200 : Chicharra = 1 : Waitms 200 :
Chicharra = 0 : Waitms 200 : Chicharra = 1 : Waitms 200 :
Chicharra = 0 : Waitms 200 : Chicharra = 1 : Waitms 200 :

```

```

Cont_rtx = 0 : Buffer_trx = "" : Cont_telf = 0 : Buffer_telf = ""

```

```

Cont_clave = 0

```

```

Band_teclado = 1      'selecciona el tipo de teclado que se verificara

```

```

Band_on_off_alarma = 1 'Alarma inicia activada en caso de corte de energía

```

```

Band_chek_sensores = 1

```

```

Band_cont_msj = 0

```

```

Cont_enviar_msj = 0

```

```

Cont_chicharra = 0

```

**'Mensajes en General para el sistema de seguridad**

```

Gosub Mensaje_general

```

```

Enable Urx

```

```

Do

```

```

  Gosub Teclado

```

```

  If Número_tecla <> 12 Then

```

```

    If Band_teclado = 1 Then Gosub On_off_alarma

```

```

    If Band_teclado = 0 Then Gosub Cambiar_key_alarma

```

```

  End If

```

```
If Buffer_trxx <> "" And Cont_rtxx > 18 And Band_chek_sensores = 1 Then
'Buffer_size_rtxx-2
```

```
    Waitms 1
```

```
Msj_alarm_activada = Instr(1 , Buffer_trxx ,Codigo_sensor )
```

```
If Msj_alarm_activada > 0 Then
```

```
    Band_cont_msj = 1
```

```
    Cont_chicharra = 0
```

```
End If
```

```
    Cont_rtxx = 0 : Buffer_trxx = ""
```

```
End If
```

```
If Band_cont_msj = 1 Then
```

```
    Incr Cont_chicharra
```

```
    Incr Cont_enviar_msj
```

```
    If Cont_chicharra > 1000 Then
```

```
        Toggle Chicharra
```

```
        Cont_chicharra = 0
```

```
    End If
```

```
    Waitms 1
```

```
End If
```

```
If Cont_enviar_msj > Tiempo_miliseg And Band_cont_msj = 1 Then
```

```
Chicharra = 1
```

```
    Sirena = 1
```

```
    Gosub Enviar_msj_robo_casa
```

```
Band_cont_msj = 0 : Cont_enviar_msj = 0
```

```
End If
```

```
If Buffer_telf <> "" And Cont_telf > 40 Then
```

```
'+CMT: "+593985675740", "12/06/20,10:46:29-20"Activa-alarma
```

```
    '+CMT: "+593995747623", "12/06/20,10:46:29-20"Desactiva-alarma
```

```
'+CMT: "+593985675740",,"12/06/20,10:46:29-20"Clave-cambio
'+CMT: "+593995747623",,"12/06/20,10:46:29-20"F1-on-casa
'+CMT: "+593985675740",,"12/06/20,10:46:29-20"F1-off-casa
```

### 'Comandos a enviar

```
'ENVIO AT+CPBR=1 RESPONDE +CPBR:1,"0995747623",129,"cont1"OK
'ENVIO AT+CPBR=2 RESPONDE +CPBR:2,"0985675740",129,"cont2"OK
'ENVIO AT+CPBR=3 RESPONDE +CPBR:3,"0995747623",129,"cont3"OK
'ENVIO AT+CPBR=4 RESPONDE +CPBR:4,"0985675740",129,"cont4"OK
'ENVIO AT+CPBR=3 RESPONDE +CPBR:5,"0995747623",129,"cont5"OK
```

Waitms 100

```
Msj_activa = Instr(1 , Buffer_telf , "Activa") 'Activa-alarma
Msj_desactiva = Instr(1 , Buffer_telf , "Desactiva" ) 'Desactiva-alarma
Cambiar_clave = Instr(1 , Buffer_telf , "Clave" )'Clave-cambio
```

```
Msj_foco1_on = Instr(1 , Buffer_telf , "F1-on" ) 'F1-on-casa
Msj_foco1_off = Instr(1 , Buffer_telf , "F1-off" )'F1-off-casa
```

```
Msj_foco2_on = Instr(1 , Buffer_telf , "F2-on" )
Msj_foco2_off = Instr(1 , Buffer_telf , "F2-off" )
```

```
Msj_foco3_on = Instr(1 , Buffer_telf , "F3-on" )
Msj_foco3_off = Instr(1 , Buffer_telf , "F3-off" )
```

```
Msj_foco4_on = Instr(1 , Buffer_telf , "F4-on" )
Msj_foco4_off = Instr(1 , Buffer_telf , "F4-off" )
```

```
If Msj_activa > 0 And Msj_desactiva = 0 Then 'activa la alarma
Cls : Locate 1 , 1 : Lcd "ALARMA ACTIVADA"
Wait 1
Gosub Mensaje_general
```

```
Band_chek_sensores = 1
Sirena = 0 : Band_cont_msj = 0 : Cont_enviar_msj = 0
Gosub Sonar_sirena
```

End If

If Msj\_desactiva > 0 And Msj\_activa = 0 Then   **'desactiva la alarma**  
 Cls : Locate 1 , 1 : Lcd **"DEACTIVA ALARMA"**  
 Wait 1

    Gosub Mensaje\_general

    Band\_chek\_sensores = 0

    Sirena = 0 : Band\_cont\_msj = 0 : Cont\_enviar\_msj = 0

Gosub Sonar\_sirena

End If

If Cambiar\_clave > 0 Then                   **'permite q se modifique la clave**

Band\_chek\_sensores = 0 : Sirena = 0 : Band\_cont\_msj = 0 : Cont\_enviar\_msj = 0

Cls : Locate 1 , 1 : Lcd **"MODIFICAR CLAVE"**

Band\_teclado = 0

End If

If Msj\_foco1\_on > 0 Then Foco\_1 = 1

If Msj\_foco1\_off > 0 Then Foco\_1 = 0

If Msj\_foco2\_on > 0 Then Foco\_2 = 1

If Msj\_foco2\_off > 0 Then Foco\_2 = 0

If Msj\_foco3\_on > 0 Then Foco\_3 = 1

If Msj\_foco3\_off > 0 Then Foco\_3 = 0

If Msj\_foco4\_on > 0 Then Foco\_4 = 1

If Msj\_foco4\_off > 0 Then Foco\_4 = 0

Cont\_telf = 0 : Buffer\_telf = "   " : Key\_alarma\_leida = "   "

End If

If Pul\_panico = 0 Then

Cls : Locate 1 , 1 : Lcd **"PANICO ACTIVADO"**

```

    Wait 1
    Gosub Mensaje_general
    Gosub Enviar_msj_asalto
End If

Loop

On_off_alarma:
'Número_tecla = 11 => Enter
'Número_tecla = 10 => Limpiar

    If Número_tecla = 10 Then

        Cont_clave = 0 : Clave_armar = "   " : Clave_lcd_aux = "   "
Cls : Locate 1 , 1 : Lcd " CLAVE ALARMA "

    End If

    If Número_tecla <> 10 And Número_tecla <> 11 Then

        Incr Cont_clave
        If Cont_clave > 5 Then Cont_clave = 5

        Número_tecla = Número_tecla + 48
        Mid(clave_armar , Cont_clave , 1 ) = Número_tecla

        Cls : Locate 1 , 1 : Lcd " CLAVE ALARMA "
        'Locate 2 , 6 : Lcd Clave_armar

        Mid(clave_lcd_aux , Cont_clave , 1 ) = "*"
        Locate 2 , 6 : Lcd Clave_lcd_aux

    End If

    If Número_tecla = 11 Then

        Key_alarma_leida = Key_alarma_guardada 'Clave guardada en la eeprom

        If Clave_armar = Key_alarma_leida Then

```



If Band\_on\_off\_alarma = 0 Then

Locate 2 , 1 : Lcd "**ALARMA ACTIVADA**"

Band\_chek\_sensores = 1

Sirena = 0 : Band\_cont\_msj = 0 : Cont\_enviar\_msj = 0

Gosub Sonar\_chicharra

End If

If Band\_on\_off\_alarma = 1 Then

Locate 2 , 1 : Lcd "**DESACTIVA ALARMA**"

Band\_chek\_sensores = 0

Sirena = 0 : Band\_cont\_msj = 0 : Cont\_enviar\_msj = 0

Gosub Sonar\_chicharra

End If

Toggle Band\_on\_off\_alarma

Wait 1

Gosub Mensaje\_general

Cont\_clave = 0 : Clave\_armar = " " : Key\_alarma\_leida = " " :  
Clave\_lcd\_aux = " "

Else

Locate 2 , 1 : Lcd "**CLAVE INCORRECTA**"

Chicharra = 0 : Wait 1 : Chicharra = 1

Cont\_clave = 0 : Clave\_armar = " " : Clave\_lcd\_aux = " "  
Gosub Mensaje\_general

End If

End If

Return

**Cambiar\_key\_alarma:**

'Número\_tecla = 11 => Enter

'Número\_tecla = 10 => Limpiar

If Número\_tecla = 10 Then

Cont\_clave = 0 : Clave\_armar = " "

Cls : Locate 1 , 1 : Lcd "**MODIFICAR CLAVE**"

End If

If Número\_tecla <> 10 And Número\_tecla <> 11 Then

Incr Cont\_clave

If Cont\_clave > 5 Then Cont\_clave = 5

Número\_tecla = Número\_tecla + 48

Mid(clave\_armar , Cont\_clave , 1 ) = Número\_tecla

Cls : Locate 1 , 1 : Lcd "**MODIFICAR CLAVE**"

Locate 2 , 6 : Lcd Clave\_armar

End If

If Número\_tecla = 11 Then

Key\_alarma\_guardada = Clave\_armar

Band\_teclado = 1 **'selecciona el teclado de armado desarmado**

Band\_on\_off\_alarma = 0 **'mantiene la alarma desactivada**

Clave\_armar = " "

Cont\_clave = 0

Cls :

Locate 1 , 1 : Lcd " **MODIFICANDO** "

Locate 2 , 1 : Lcd " **CLAVE** "

Wait 1

Gosub Sonar\_chicharra

Gosub Mensaje\_general

End If

Return

**Sonar\_sirena:**

```
If Band_chek_sensores = 1 Then          'activa  
  For Cont_for = 1 To 4  
    Sirena = 1 : Waitms 400 : Sirena = 0 : Waitms 400  
  Next Cont_for  
End If
```

```
If Band_chek_sensores = 0 Then          'desactiva  
  For Cont_for = 1 To 4  
    Sirena = 1 : Wait 1 : Sirena = 0 : Wait 1  
  Next Cont_for  
End If
```

Return

**Sonar\_chicharra:**

```
If Band_chek_sensores = 1 Then          'activa  
  For Cont_for = 1 To 4  
    Chicharra = 0 : Waitms 400 : Chicharra = 1 : Waitms 400  
  Next Cont_for  
End If
```

```
If Band_chek_sensores = 0 Then          'desactiva  
  For Cont_for = 1 To 4  
    Chicharra = 0 : Wait 1 : Chicharra = 1 : Wait 1  
  Next Cont_for  
End If
```

Return

**Enviar\_msj\_robo\_casa:**

Disable Urxc

```

Buffer_telf = "" : Cont_telf = 0 : Selec_sim_num = "AT+CPBR=1" : Gosub
Mensaje_robo
  Buffer_telf = "" : Cont_telf = 0 : Selec_sim_num = "AT+CPBR=2" : Gosub
Mensaje_robo
  Buffer_telf = "" : Cont_telf = 0 : Selec_sim_num = "AT+CPBR=3" : Gosub
Mensaje_robo
  Buffer_telf = "" : Cont_telf = 0 : Selec_sim_num = "AT+CPBR=4" : Gosub
Mensaje_robo
  Buffer_telf = "" : Cont_telf = 0 : Selec_sim_num = "AT+CPBR=5" : Gosub
Mensaje_robo
  Buffer_telf = "" : Cont_telf = 0
Enable Urx
Return

```

### **Enviar\_msj\_asalto:**

```
Disable Urx
```

```

  Buffer_telf = "" : Cont_telf = 0 : Selec_sim_num = "AT+CPBR=4" : Gosub
Mensaje_asalto
  Buffer_telf = "" : Cont_telf = 0 : Selec_sim_num = "AT+CPBR=5" : Gosub
Mensaje_asalto

```

```
Enable Urx
```

```
Return
```

### **Mensaje\_robo:**

```
Cls : Locate 1 , 1 : Lcd "EnviaMensajeRobo"
```

```
Do
```

```
  Print #1 , Selec_sim_num ; Chr(13) : Waitms 300
```

```

  Conf_tel1 = Instr(1 , Buffer_telf , "0")      'espera
+CPBR:1,"012345678",129,"nombre"OK
  Num_telf = Mid(buffer_telf , Conf_tel1 , Zise_num_telf)
  Conf_tel2 = Instr(1 , Buffer_telf , "OK")

```

```
Loop Until Conf_tel2 > 0
```

```

Print #1 , "AT+CMGS=" ; Chr(34) ; Num_telf ; Chr(34) ; Chr(13)
Waitms 200
Print #1 , "ROBO EN EJECUCION" : Print #1 , Chr(26)

```

```

Wait 1                               ' Print "ATH" : Waitms 200

```

```

Return

```

### **Mensaje\_asalto:**

```

Cls : Locate 1 , 1 : Lcd "EnviaMeAsalto"

```

```

Do

```

```

Print #1 , Selec_sim_num ; Chr(13) : Waitms 300

```

```

    Conf_tel1 = Instr(1 , Buffer_telf , "0")   'espera
+CPBR:1,"012345678",129,"nombre"OK
    Num_telf = Mid(buffer_telf , Conf_tel1 , Zise_num_telf)
    Conf_tel2 = Instr(1 , Buffer_telf , "OK")

```

```

Loop Until Conf_tel2 > 0

```

```

Print #1 , "AT+CMGS=" ; Chr(34) ; Num_telf ; Chr(34) ; Chr(13)
Waitms 200
Print #1 , "AYUDA NOS ESTAN ASALTANDO" : Print #1 , Chr(26)

```

```

Wait 1                               ' Print "ATH" : Waitms 200

```

```

Return

```

```

Confirmacion_mod_telf: 'busca OK dentro del string que envía el módulo

```

```

    Waitms 300

```

```

Módulo_respuesta = Instr(1 , Buffer_telf , "OK" )
Mod_res_error = Instr(1 , Buffer_telf , "ERROR" )

```

```

Return

```

Mensaje\_general:

Cls :

Locate 1 , 1 : Lcd " **ALARMA CASA** "  
 Locate 2 , 1 : Lcd " **MÓDULO GSM** "

Return

Recepcion\_rx:

Incr Cont\_rtx  
 Mid(buffer\_trrx , Cont\_rtx , 1 ) = Inkey()  
 If Cont\_rtx >= Buffer\_size\_rtx Then Cont\_rtx = 0

Return

**Módulo GSM:**

Disable Urx  
 Incr Cont\_telf

Mid(buffer\_telf , Cont\_telf , 1 ) = Inkey(#1)

If Cont\_telf >= Buffer\_size\_telf Then Cont\_telf = 0

Enable Urx  
 Return

**Teclado:**

Número\_tecla = 12  
 Gosub \_delay\_1

Fila1 = 0  
 Gosub \_delay\_1 : If Columna1 = 0 Then Gosub Tecla\_1  
 Gosub \_delay\_1 : If Columna2 = 0 Then Gosub Tecla\_2  
 Gosub \_delay\_1 : If Columna3 = 0 Then Gosub Tecla\_3

Fila1 = 1

Gosub \_delay\_1

```

Fila2 = 0
  Gosub _delay_1 : If Columna1 = 0 Then Gosub Tecla_4
  Gosub _delay_1 : If Columna2 = 0 Then Gosub Tecla_5
  Gosub _delay_1 : If Columna3 = 0 Then Gosub Tecla_6
Fila2 = 1

```

```

  Gosub _delay_1

```

```

Fila3 = 0
  Gosub _delay_1 : If Columna1 = 0 Then Gosub Tecla_7
  Gosub _delay_1 : If Columna2 = 0 Then Gosub Tecla_8
  Gosub _delay_1 : If Columna3 = 0 Then Gosub Tecla_9
Fila3 = 1

```

```

  Gosub _delay_1

```

```

Fila4 = 0 :
  Gosub _delay_1 : If Columna1 = 0 Then Gosub Tecla_10
  Gosub _delay_1 : If Columna2 = 0 Then Gosub Tecla_11
  Gosub _delay_1 : If Columna3 = 0 Then Gosub Tecla_12
Fila4 = 1

```

Return

\_delay\_1:

```

Nop : Nop : Nop : Nop
Nop : Nop : Nop : Nop
Nop : Nop : Nop : Nop
Nop : Nop : Nop : Nop
Nop : Nop : Nop : Nop
Nop : Nop : Nop : Nop
Nop : Nop : Nop : Nop
Nop : Nop : Nop : Nop
Return

```

Tecla\_1:

```

  Waitms 15

```

```

  If Columna1 = 0 Then
    Número_tecla = 1
    Gosub Rebotes
  End If

```

Return

Tecla\_2:

Waitms 15

If Columna2 = 0 Then  
Número\_tecla = 2  
Gosub Rebotes  
End If

Return

Tecla\_3:

Waitms 15

If Columna3 = 0 Then  
Número\_tecla = 3  
Gosub Rebotes  
End If

Return

Tecla\_4:

Waitms 15

If Columna1 = 0 Then  
Número\_tecla = 4  
Gosub Rebotes  
End If

Return

Tecla\_5:

Waitms 15

If Columna2 = 0 Then  
Número\_tecla = 5  
Gosub Rebotes  
End If

Return



Tecla\_6:

Waitms 15

If Columna3 = 0 Then  
Número\_tecla = 6  
Gosub Rebotes  
End If  
Return

Tecla\_7:

Waitms 15

If Columna1 = 0 Then  
Número\_tecla = 7  
Gosub Rebotes  
End If  
Return

Tecla\_8:

Waitms 15

If Columna2 = 0 Then  
Número\_tecla = 8  
Gosub Rebotes  
End If

Return

Tecla\_9:

Waitms 15

If Columna3 = 0 Then  
Número\_tecla = 9  
Gosub Rebotes  
End If  
Return

Tecla\_10:

Waitms 15

If Columna1 = 0 Then

```
Número_tecla = 10
  Gosub Rebotes
End If
Return
```

```
Tecla_11:
  Waitms 15
```

```
  If Columna2 = 0 Then
Número_tecla = 0
  Gosub Rebotes
End If
```

```
Return
```

```
Tecla_12:
  Waitms 15
```

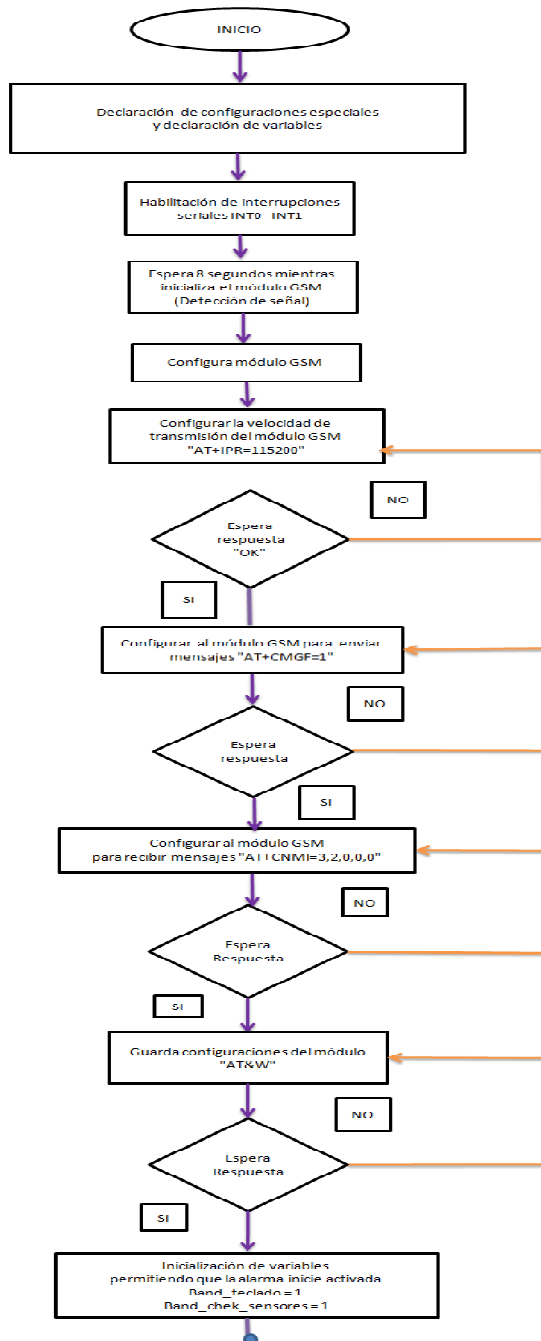
```
  If Columna3 = 0 Then
Número_tecla = 11
Gosub Rebotes
  End If
Return
```

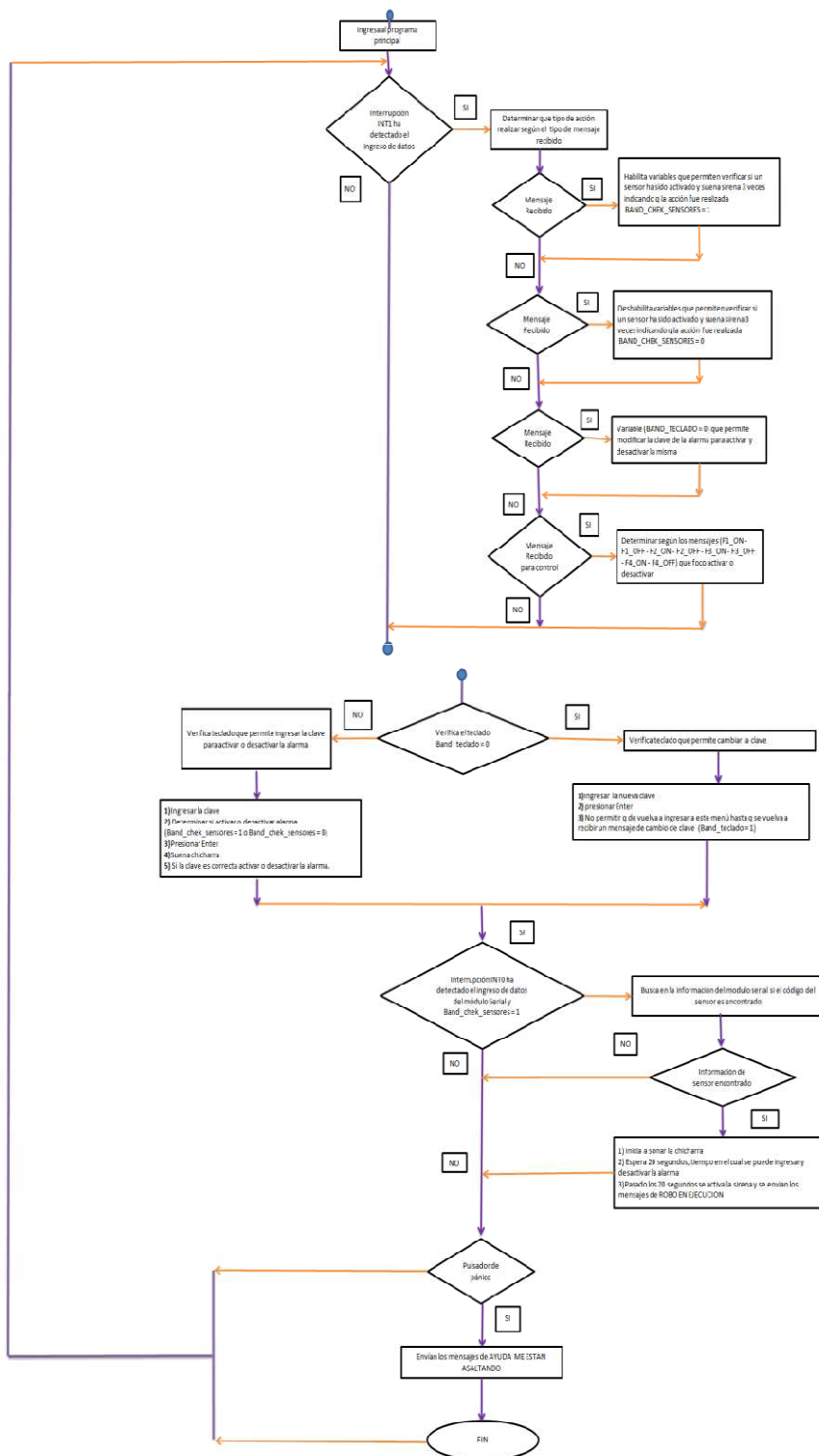
```
Fila1 = 1 : Fila2 = 1 : Fila3 = 1 : Fila4 = 1 : Waitms 300
```

```
Return
```

```
'If Número_tecla = 11 Then
  ' Incr Cont_panico
  ' If Cont_panico > 5 Then
'   Cls : Locate 1 , 1 : Lcd "AYUDA ROBO EN PROGRESO"
'   Wait 1
  '   Cont_panico = 0
  ' End If
'
  'End If
```

### 3.6 DIAGRAMA DE FLUJO DEL SISTEMA DE SEGURIDAD





### 3.7 PROGRAMA “EAGLE” PARA EL DISEÑO DEL CIRCUITO

El Editor Eagle es una poderosa herramienta que permite diseñar circuitos impresos de forma fácil.

El nombre Eagle es un acrónimo de “Easily Applicable Graphical Layout Editor”, el programa consiste de tres principales módulos: editor de trazado, editor esquemático y auto ruteado, pudiendo editar con estos módulos los archivos que formarán parte de un circuito impreso.

En la Tabla 3.2<sup>22</sup> se muestra un listado con los tipos de archivos más importantes que pueden ser editados con Eagle.

**Tabla 3. 2:** Archivos de Eagle

<b>Tipo</b>	<b>Ventana</b>	<b>Nombre</b>
Placa	Editor de líneas de conexión	*.bdr
Esquema	Editor de esquemas	*.sch
Librería	Editor de librerías	*.lbr

#### 3.7.1 EDITOR DE ESQUEMAS

En esta ventana se realiza un ejemplo de un diagrama esquemático de un circuito con todas las conexiones que requiere el diseño.

La Figura N°3.13 permite observar la ventana del editor de esquemas con un circuito interno.

<sup>22</sup> [http://picmania.garcia-cuervo.net/eagle\\_tutlbr\\_i\\_library.php](http://picmania.garcia-cuervo.net/eagle_tutlbr_i_library.php)

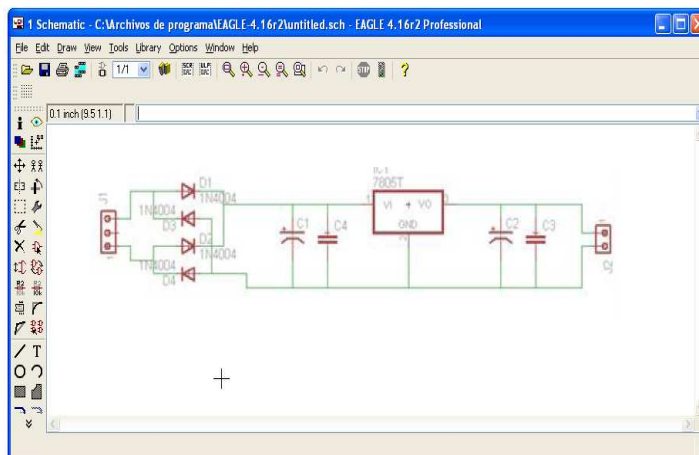


Figura 3. 13: Ejemplo del entorno del editor de esquemas

### 3.7.2 EDITOR DE LÍNEAS DE CONEXIÓN

En este archivo los componentes del circuito, creados en el diagrama esquemático son observados conservando su forma real y ocupando un espacio definido.

El editor de líneas permite crear las pistas de conexión entre los elementos que forman parte del circuito estableciendo un tamaño y forma definida para que finalizado su diseño pueda este ser imprimido, transferido a una baquelita y quemado.

La Figura N° 3.14 muestra un ejemplo del circuito del diagrama esquemático finalizado en el editor de líneas con sus respectivas conexiones.

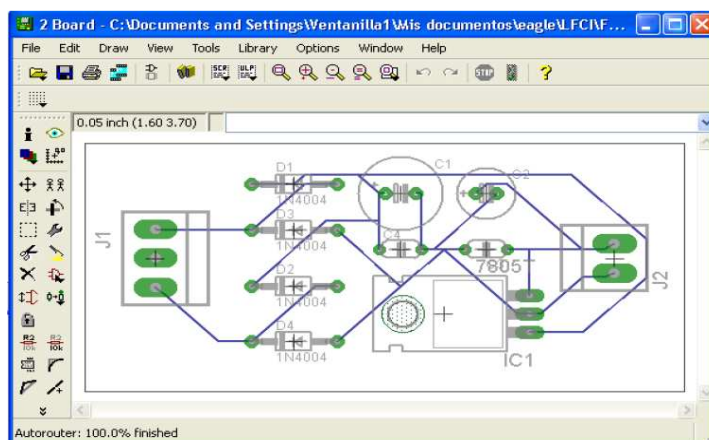


Figura 3. 14: Ejemplo de entorno de líneas de conexión

### 3.7.3 EDITOR DE LIBRERÍA

Eagle facilita el diseño de un circuito, debido a que permite crear librerías propias para elementos que no sean encontrados dentro de sus librerías disponibles o que necesiten ser modificados en alguna de sus características, permitiendo crear un componente con conexiones específicas.

Una librería está compuesta de uno o varios componentes electrónicos. Cada elemento tiene tres archivos, uno a utilizar en el editor esquemático, otro para el editor de líneas y uno que será presentado en el panel de control con sus características.

Todos estos archivos poseen una conexión común entre pines de entrada, salida o alimentación para formar un elemento.

En la Figura N° 3.15 se puede observar el panel de control con la librería de un elemento presentando, su diagrama esquemático y el paquete que será utilizado en el editor de líneas de conexión.

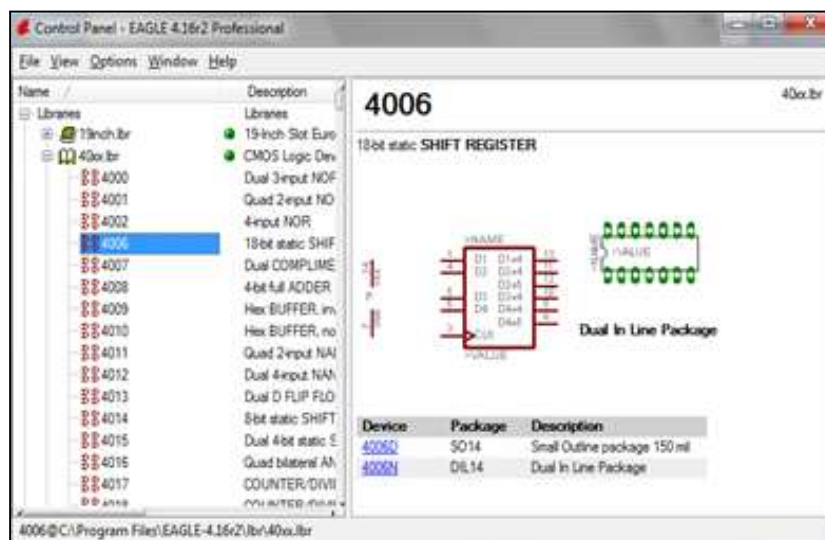


Figura 3. 15: Panel de control

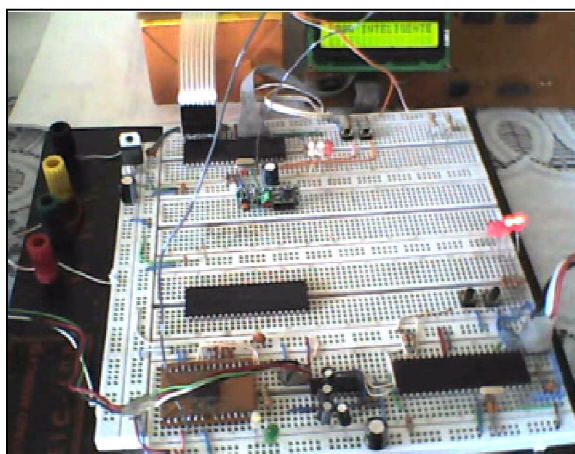
## CAPITULO IV

### 4 PRUEBAS Y RESULTADOS

#### 4.1 PRUEBAS

##### 4.1.1 PRUEBA 1: ACTIVACIÓN Y DESACTIVACIÓN DE LA ALARMA

En esta fase explicaremos la activación y desactivación de la alarma mediante mensajes de texto efectuando pruebas en el protoboard con la simulación del circuito previamente armado como se muestra en la Figura N°4.1.



**Figura 4. 1:** Simulación del circuito del sistema de seguridad en protoboard

La alarma se encuentra por defecto encendida al ser energizada, trabajando a 115200 Baudios de velocidad entre el módulo GSM ZTE y el microcontrolador ATmega164P según estándar.

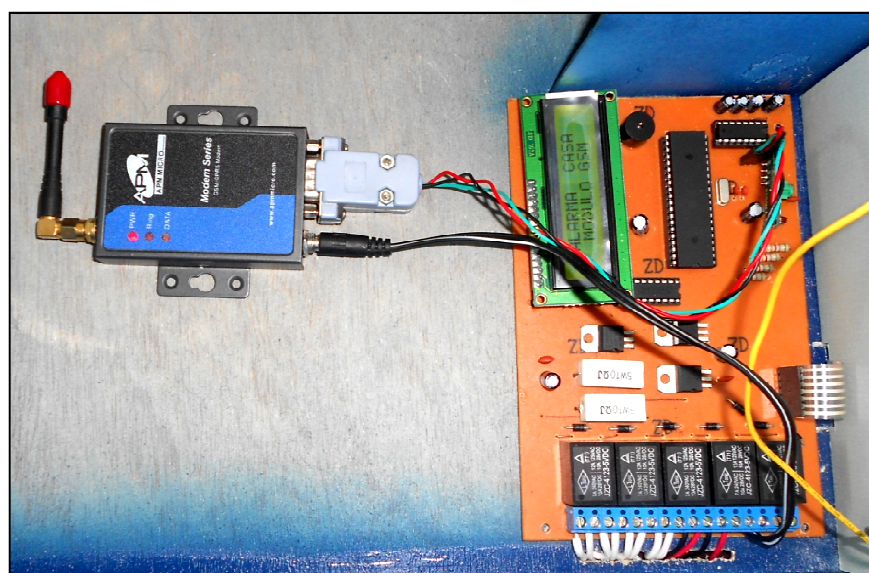
Para la activación de la alarma, se utiliza un celular que manda una señal o código al módulo GSM ZTE solicitando una respuesta, dicha respuesta debe ser efectuada en un máximo de 20 segundos; es decir el usuario tiene este tiempo para introducir por teclado la contraseña que permitirá activar el sistema de seguridad; caso contrario el módulo GSM dará por entendido que se está tratando de vulnerar el sistema de



seguridad y mandará un mensaje de alerta a los 4 contactos previamente programados en el microcontrolador con el mensaje **“ROBO EN EJECUCIÓN”**.

Para la activación y desactivación de la alarma se decidió ingresar la contraseña por teclado, para no confundir al usuario con dos o más códigos y facilitar su operación.

Cuando se activa y desactiva la alarma siempre se inicializa el módulo GSM buscando las frecuencias y de esta manera tener un armando del sistema de seguridad como se muestra en la Figura N°4.2



**Figura 4. 2:** Inicialización del módulo GSM para el circuito

#### 4.1.2 PRUEBA 2: INGRESO DE 5 DÍGITOS PARA CONTROL DEL SISTEMA

El teclado cumple dos funciones importantes dentro del sistema de seguridad y a continuación se hará el ensayo:

##### ❖ **Cambio de clave:**

Para experimentar cambio de clave en el sistema de seguridad se mandó una señal o código GSM con el comando **“Cambio-clave”** y a continuación se ingresó una clave de 5 dígitos tomando en cuenta lo que se muestra en la Tabla 4.1.

**Tabla 4. 1:** Tabla de Comandos del teclado para el sistema de seguridad

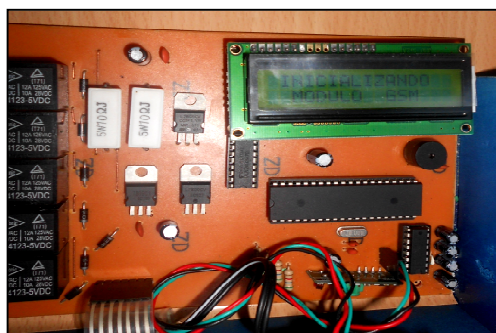
Comandos	Funcionamiento
*	Borrado
#	Enter
Números: [ 0-9]	Clave de 5 dígitos

#### ❖ Activar y Desactivar Alarma

Se envía un mensaje de texto desde el celular al módulo GSM con los comandos que se muestran en la Tabla 4.1 y la Tabla 4.2

**Tabla 4. 2:** Mensajes de texto para el módulo GSM

Mensajes de texto
Activa-alarma
Desactiva-alarma

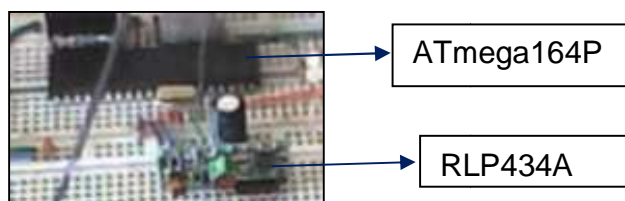
**Figura 4. 3:** Inicialización del sistema de seguridad al activarlo

#### 4.1.3 PRUEBA 3: DETECTANDO MOVIMIENTO

En esta fase se probó la interacción de los sensores inalámbricos con el módulo GSM ZTE y a continuación se explicará los pasos con los que se determinó los códigos y señales para los sensores inalámbricos (no se explicara los sensores magnéticos ya que cumplen el mismo principio).

### PASO 1

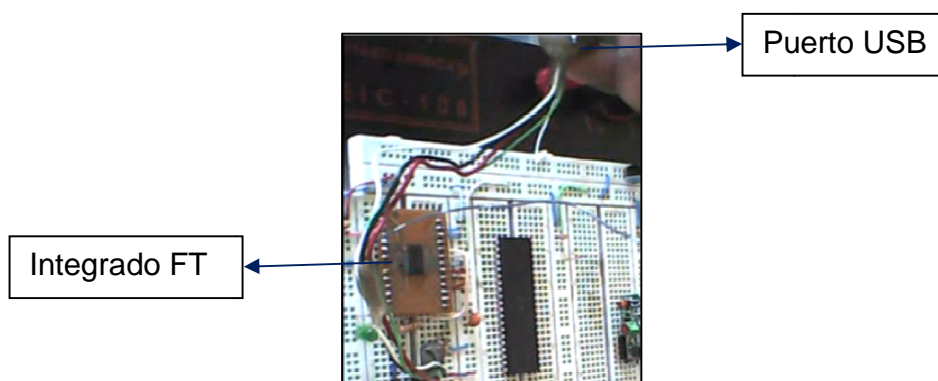
Se utiliza el módulo Rx (RLP434A) conectado al microcontrolador (Atmega164P) como se muestra en la Figura N° 4.4 , para leer todos los códigos que se encuentran en el aire.



**Figura 4. 4:** Conexión del ATmega164P y el módulo Rx

### PASO 2

Una vez que se obtienen todos estos códigos por el módulo Rx estos son enviados al microcontrolador Atmega164P para a su vez ser enviados al integrado FT. El integrado FT nos permitirá conectarnos a la PC mediante puerto USB como se observa en la Figura N° 4.5 y visualizar mediante un emulador (hyperterminal) las señales del ambiente.



**Figura 4. 5:** Conexión serial para visualizar señales en la PC

### PASO 3

Para visualizar las señales o códigos del sensor inalámbrico en el emulador (hyperterminal), primero se comprobó que se esté leyendo las señales del aire (Figura N° 4.6) y a continuación se activa el sensor inalámbrico el cual emite señales de F0 y 00 como se muestra en la Figura N° 4.7.

Para obtener una señal constante del sensor inalámbrico se movió los jumpers para así tener una mayor cantidad de F0 juntos en comparación de 00, facilitando el reconocimiento de una señal de alerta en el microcontrolador como se muestra en la Figura N°4.8. Para la detección del código del sensor se tomó 4F0.

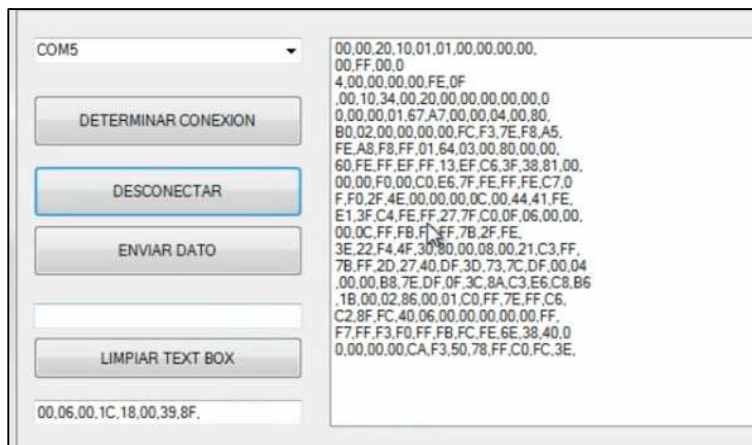


Figura 4. 6: Diferentes señales emitidas por el aire

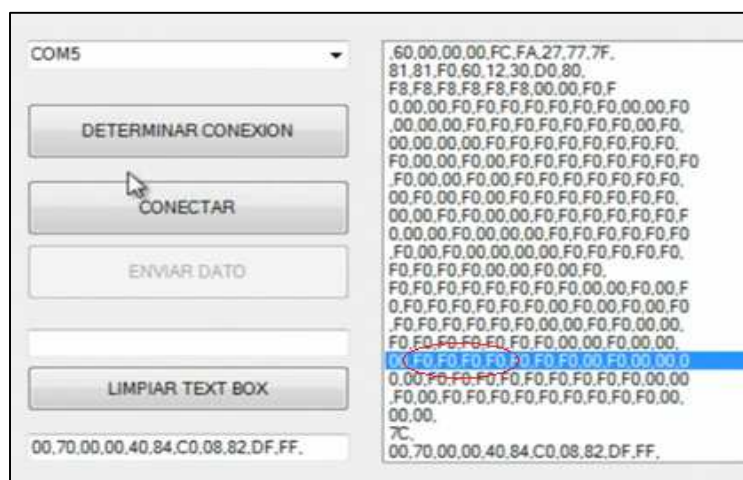
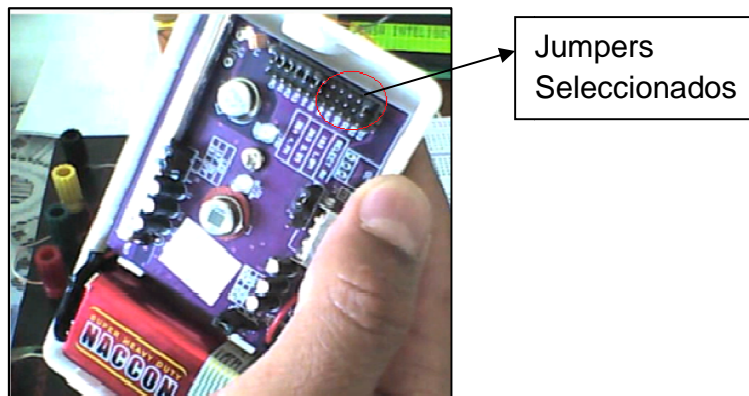


Figura 4. 7: Señales emitida por el sensor inalámbrico



**Figura 4. 8:** Ubicación de los jumpers para mandar señales F0

#### **4.1.4 PRUEBA 4: CONTROL DE ILUMINACIÓN**

El control de iluminación es un complemento importante dentro del sistema de seguridad por lo que se hizo sencillo de manipular para el usuario

En esta fase se realizó las pruebas con una sola bombilla; ya que todos los focos de las áreas de la vivienda tienen el mismo funcionamiento.

Se mandó un mensaje de texto con el comando **“F1-on-casa”** para el módulo GSM y automáticamente se encendió con éxito, de la misma manera procedimos a apagar enviando un mensaje con el comando **“F1-off-casa”**.



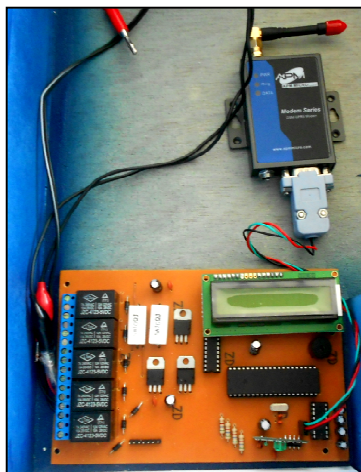
**Figura 4. 9:** Prueba encendido de Foco 1



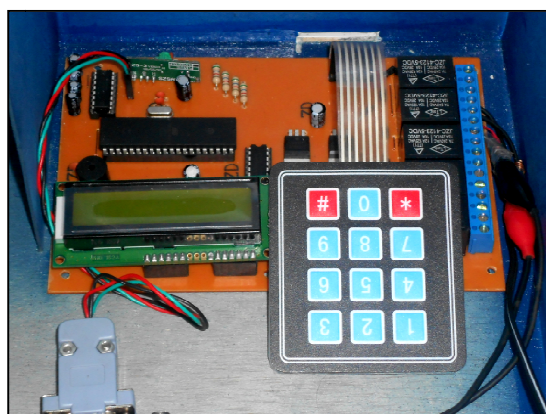
**Figura 4. 10:** Prueba apagado Foco 1

## 4.2 RESULTADOS:

Se acopló el cerebro del sistema de seguridad en la maqueta, junto con la instalación de la iluminación y de esta manera poder simular su funcionamiento obteniendo los siguientes resultados.



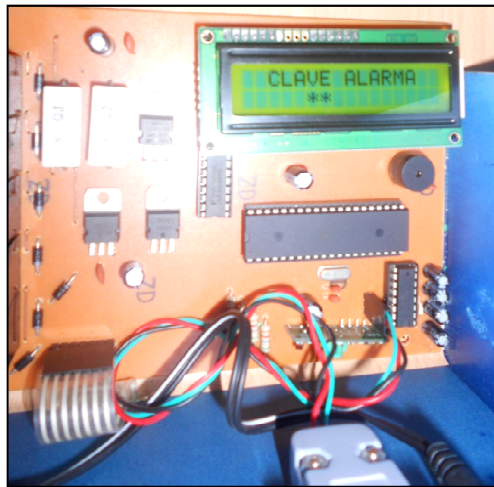
**Figura 4. 11:** Cerebro del sistema de seguridad



**Figura 4. 12:** Cerebro del sistema de seguridad con la conexión del teclado

#### 4.2.1 RESULTADO DE CAMBIO DE CLAVE:

Se procedió a enviar un mensaje de texto con el comando “CAMBIO CLAVE” en este momento se observa que en la pantalla del LCD aparece el mensaje “INGRESE NUEVA CLAVE”, el usuario puede proceder a ingresar dicha clave de 5 dígitos, a continuación se observó que la clave se guardó con éxito y que al ingresar esta nueva clave se activa nuevamente el sistema de seguridad, además de verificar que las teclas funcionen como se las diseñó.



**Figura 4. 13:** Nueva clave de 5 dígitos



**Figura 4. 14:** Mensaje antes de ingresar 5 dígitos nuevos.



**Figura 4. 15:** Mensaje luego de ingresar los 5 dígitos nuevos

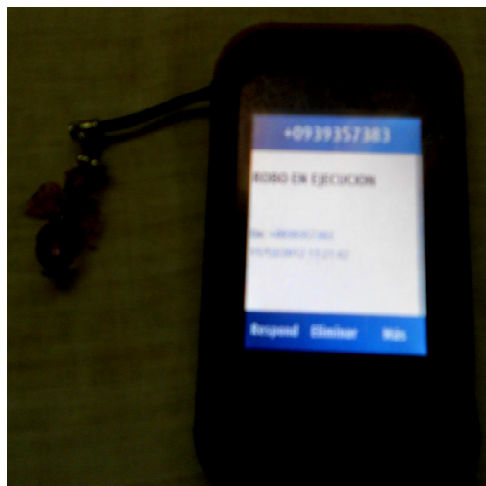


**Figura 4. 16:** Ubicación del teclado en la maqueta.

#### **4.2.2 DURANTE ROBO:**

Se simuló robo activando el sensor inalámbrico de movimiento, este tarda unos segundos en enviar las frecuencias al módulo GSM y este a su vez le toma unos segundos en detectar. Se logra verificar que se envía correctamente los mensajes de texto a los 2 primeros contactos guardados en la tarjeta SIM con el mensaje "ROBO EN EJECUCION" y adicional se escucha que suena la sirena que se encuentra fuera de la maqueta, nos da como resultados una sincronización del microcontrolador y el módulo GSM, con un tiempo de 20 segundos para desactivar la alarma en caso de falsa alarma.





**Figura 4. 17:** Mensaje enviado al usuario en caso de robo



**Figura 4. 18:** Sirena activada en caso de robo

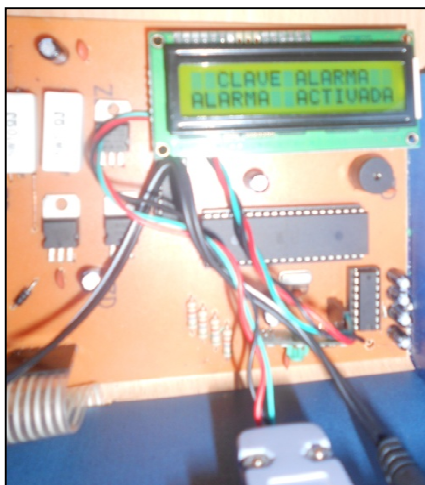
#### **4.2.3 ACTIVACIÓN Y DESACTIVACIÓN DEL SISTEMA DE SEGURIDAD:**

El usuario puede activar y desactivar el sistema de seguridad por medio del celular o por medio del teclado según su necesidad y donde se encuentre, en los dos casos se puede visualizar los mismos mensajes en el LCD.

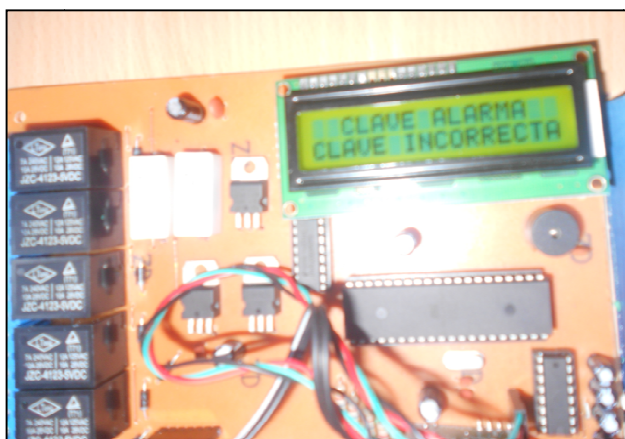
La activación de la alarma se realiza una vez que el usuario ha salido de la vivienda, para la simulación, se manda un mensaje de texto “Activa-alarma” al módulo GSM y unos segundos después se observa en el LCD “ALARMA ACTIVADA” y suena la sirena 3 veces, indicando que el sistema de seguridad empieza el proceso de armado y está listo.

De la misma forma se manda un mensaje de texto “Desactiva-alarma” y se observa que deja de realizar el proceso.

La activación y desactivación del sistema de seguridad normalmente se vuelve útil cuando el usuario ha llegado a su domicilio y se encuentra cerca de la puerta principal y va a ingresar. El usuario deberá ingresar la clave de 5 dígitos por teclado para activar y desactivar el sistema, a continuación se escuchará el sonido de la chichara 3 veces indicando al usuario que se ejecutó la instrucción.



**Figura 4. 19:** Activación del Sistema de seguridad.



**Figura 4. 20:** Ingresar 5 dígitos por teclado de forma incorrecta

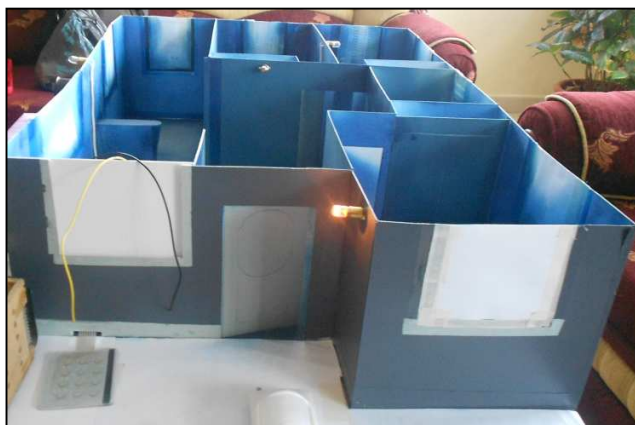


**Figura 4. 21:** Mensaje cuando el sistema está listo

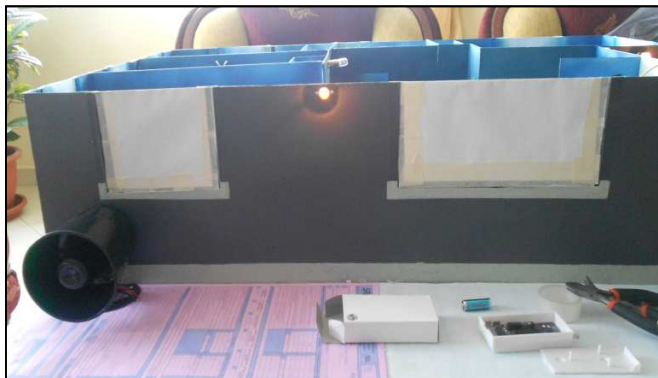
#### **4.2.4 CONTROL DE ILUMINACIÓN:**

El diseño del proyecto fue hecho para controlar cada foco que estará ubicado en 4 puntos elegidos en la maqueta, se mandó un mensaje de texto “f1-on-casa” y después de unos segundos se observó que el foco de la entrada se encendía. De igual forma se mandó un mensaje de texto “f1-off-casa” al módulo GSM para apagarlo.

Puesto que el diseño del proyecto se hizo para controlar cada foco se procedió hacer este mismo procedimiento con los 3 focos restantes para verificar que funcionaban, obteniendo el resultado deseado de verificar que el usuario es capaz de controlar la iluminación por medio de su celular.



**Figura 4. 22:** Verificación del encendido del foco 1



**Figura 4. 23:** Verificación del encendido del foco 2



**Figura 4. 24:** Verificación del encendido del foco 3



**Figura 4. 25:** Verificación el apagado de iluminación

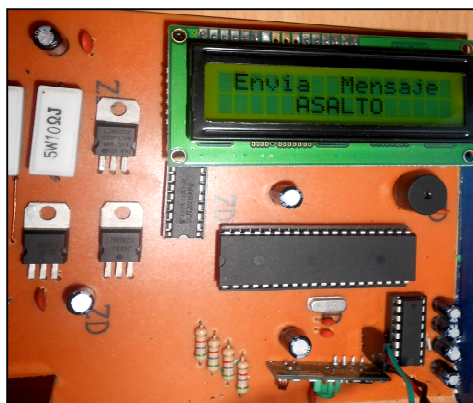
#### **4.2.5 BOTÓN DE PÁNICO:**

El botón de pánico es un auxiliar de nuestro sistema de seguridad inalámbrico, el cual será activado cuando el usuario está siendo amenazado por robo dentro del hogar, al pulsarlo se obtiene como resultados el envío de mensajes de texto a un contacto externo, ya sea un vecino o un familiar lejano, al cual le llega un mensaje

“AYUDA NOS ESTAN ASALTANDO” y en el LCD aparecerá simultáneamente el mensaje: envía mensaje ASALTO.



**Figura 4. 26:** Ubicación del botón de pánico para la simulación



**Figura 4. 27:** Mensaje en el LCD al pulsar el botón de pánico

Con la simulación se logró además tener resultados generales como los que se enuncian a continuación:

**Tabla 4. 3:** Resultados generales

Energía	Se logró medir cuanta corriente se necesita para energizar la maqueta con el sistema de seguridad ya instalado, obteniendo como resultado una fuente regulable de 12 v con una corriente $\geq [1.5- 3]$ A aproximadamente.
Medidas maqueta	Dependiendo de los elementos a utilizar nos dio como resultado una maqueta de 100 x 98 [cm], además que sea fácil de visualizar.

Iluminación	Se obtiene como resultado instalar focos de 12 [v] suficientes para la simulación, si quisiéramos instalar focos de 110 v o 220 v bastaría con acoplar un circuito con un triac o relé.
-------------	---

**Tabla 4. 4:** Mensajes de texto utilizados para el Sistema de Seguridad

1	Activa-alarma
2	Desactiva-alarma
3	Clave- cambio
4	F1-on-casa
5	F1-off-casa
6	F2-on-casa
7	F2-off-casa
8	F3-on-casa
9	F3-off-casa
10	F4-on-casa
11	F4-off-casa

**Tabla 4. 5:** Mensajes de texto que el módulo GSM envía al usuario

1	Botón de pánico	<b>ROBO EN EJECUCION</b>
2	Detección mediante sensores inalámbricos	<b>AYUDA NOSESTAN ASALTANDO</b>

## CAPITULO V

### 5 CONCLUSIONES Y RECOMENDACIONES

#### 5.1 CONCLUSIONES

- ❖ Se logró cumplir con el objetivo principal de la “Implementación de un Prototipo de Sistema de Seguridad Inalámbrico para viviendas, vía telefonía celular con tecnología GSM” en tiempo real que sea útil, confiable, sencillo de instalar, de bajo costo y fácil de manejar para cualquier usuario sin grandes conocimientos en la parte electrónica y de comunicaciones móviles.
- ❖ La tecnología móvil GSM, en la actualidad es utilizada en cualquier ámbito; uno de estos son los sistemas de seguridad que se vinculan con la tecnología GSM, los cuales permiten la utilización de mensajes cortos desde un terminal móvil (celular), para ejecutar alguna instrucción, como por ejemplo el encendido o apagado de los focos de la vivienda y la manipulación del sistema de seguridad en este proyecto.  
También ayuda a informar en caso de allanamiento o robo de la casa, a sus propietarios y varios de sus familiares, mediante un mensaje de alerta a cada uno de los contactos registrados en la tarjeta SIM del módulo GSM.
- ❖ Se utilizó el microcontrolador Atmega164P, que tiene grandes ventajas en su estructura interna como: mayor capacidad de memoria, un oscilador interno, un contador y dos puertos seriales programables que permitieron hacer la conexión con el módulo GSM, siendo esta última característica la que incentiva a utilizar este microcontrolador.
- ❖ La ventaja principal de utilizar el software BASCOM AVR, es que es un lenguaje de programación sencillo para quien lo utiliza, así como también se puede trabajar con comandos AT. Algo tan importante de este software es que permite elegir el microcontrolador a utilizarse mostrando una presentación

física del dispositivo que se está programando, y al colocar el puntero sobre sus pines muestra las características.

- ❖ Los sensores inalámbricos brindan grandes ventajas en sistemas de seguridad domésticos en comparación con sensores que necesitan cableado, entre las principales ventajas tenemos:
  - Facilidad y bajo costo de instalación
  - Mayor eficiencia y confiabilidad
  - Ahorro de energía
  - Excelentes características técnicas.
  
- ❖ En las pruebas realizadas se pudo observar y monitorear el correcto funcionamiento de las 5 etapas que consta el proyecto:
  - Fase de control: Interacción del microcontrolador Atmega164P como cerebro del sistema de seguridad, ya que este ejecuta las instrucciones programadas permitiendo una comunicación con el módulo GSM, focos, teclado, botón de pánico, sirena y sensores inalámbricos.
  - Fuente regulada de voltaje: Permite energizar el sistema de seguridad.
  - Teclado y LCD: Permite introducir instrucciones y visualizar las acciones a ejecutarse respectivamente.
  - Control de iluminación para la vivienda: Accede al encendido de cada foco de la vivienda mediante un mensaje de texto.
  - Detección de movimiento en el domicilio: Mediante sensores inalámbricos se logra monitorear en tiempo real el área de la casa.
  
- ❖ Es conveniente hacer una maqueta de una vivienda ya que permite simular en tiempo real el funcionamiento de un sistema de seguridad doméstico en caso de robo.



## 5.2 RECOMENDACIONES

- ❖ Hay que tener precaución con la fuente de alimentación, ya que para un correcto funcionamiento del sistema de seguridad inalámbrico, la corriente que circula por la misma debe ser la necesaria para que todos los dispositivos electrónicos cumplan su función, caso contrario el sistema no se activará y no podrá cumplir la función de alertar mediante una sirena y mensajes cortos a sus propietarios como el caso de este proyecto.
- ❖ Debido a que varios de los elementos utilizados en el cerebro del sistema no trabajan a un mismo voltaje, se necesitó utilizar reguladores de voltaje, para así obtener un solo valor de voltaje a la entrada de todo el sistema.
- ❖ Se debe tener en cuenta que cuando este sistema de seguridad sea implementado en una casa para su funcionamiento, es recomendable que el pulsador de pánico se encuentre en un lugar que solo los propietarios conozcan, y así cuando estén en algún peligro dentro de su hogar al pulsar ese botón se alerte de manera inmediata a otros familiares.
- ❖ Se recomienda escoger cuidadosamente el tipo de microcontrolador de acuerdo a la aplicación y los parámetros técnicos que ayudarán a llevar a cabo algún proyecto.
- ❖ Se recomienda tomar en cuenta los parámetros técnicos de cada uno de los manuales para el diseño del circuito.
- ❖ Se recomienda utilizar el módem ME3006 de la corporación ZTE ya que tiene ventajas en comparación con otros módems. Entre las principales características tenemos el uso de comandos AT que permiten comunicarse con dispositivos externos de forma sencilla, además incluye su propia tarjeta SIM lo que permite gestionar una base de datos de teléfonos.

## REFERENCIAS BIBLIOGRÁFICAS

### MICROCONTROLADOR

<sup>1</sup> [http://web.ing.puc.cl/~mtorrest/downloads/pic/tutorial\\_pic.pdf](http://web.ing.puc.cl/~mtorrest/downloads/pic/tutorial_pic.pdf)

### APLICACIONES DEL MICROCONTROLADOR

<sup>2</sup> [http://galia.fc.uaslp.mx/~cantocar/microcontroladores/SLIDES\\_8051\\_PDF/3\\_APLICA.PDF](http://galia.fc.uaslp.mx/~cantocar/microcontroladores/SLIDES_8051_PDF/3_APLICA.PDF)

### MEMORIA

<sup>3</sup> [http://www.dte.us.es/tec\\_inf/itig/etc2/material/ETC2\\_0304\\_memorias.pdf](http://www.dte.us.es/tec_inf/itig/etc2/material/ETC2_0304_memorias.pdf)

### RECURSOS ESPECIALES DE LOS MICROCONTROLADORES

<sup>5</sup> <http://bibdigital.epn.edu.ec/bitstream/15000/2349/1/CD-3097.pdf>

### LCD

<sup>10</sup> <http://es.scribd.com/doc/44252680/LCD-16X2>

### TLP434

<sup>12</sup> <http://www.alldatasheet.com/view.jsp?Searchword=TLP434>

### MAX232

<sup>12</sup> [http://www.wvshare.com/datasheet/MAXIM\\_PDF/MAX220-MAX249.pdf](http://www.wvshare.com/datasheet/MAXIM_PDF/MAX220-MAX249.pdf)

### MÓDULO GSM ZTE

<sup>13</sup> [bibdigital.epn.edu.ec/bitstream/15000/4391/1/CD-4004.pdf](http://bibdigital.epn.edu.ec/bitstream/15000/4391/1/CD-4004.pdf)

### ULN2003A

<sup>14</sup> [http://www.datasheetcatalog.net/es/datasheets\\_pdf/U/L/N/2/ULN2003.shtml](http://www.datasheetcatalog.net/es/datasheets_pdf/U/L/N/2/ULN2003.shtml)

**MENSAJES DE TEXTO SMS**

<sup>18</sup><http://dspace.esPOCH.edu.ec/bitstream/123456789/331/1/38T00174.pdf>

**TECNOLOGÍA GSM APLICADA A LOS SISTEMAS DE SEGURIDAD**

<sup>19</sup> [bibdigital.epn.edu.ec/bitstream/15000/4869/1/CD-4463.pdf](http://bibdigital.epn.edu.ec/bitstream/15000/4869/1/CD-4463.pdf)

**MICROCONTROLADOR Atmega164P**

<sup>20</sup><http://www.atmel.com/Images/doc7674.pdf>

**LENGUAJE DE PROGRAMACIÓN BASCOM AVR**

<sup>21</sup><http://bibdigital.epn.edu.ec/bitstream/15000/3835/1/CD-3618.pdf>

**COMANDO AT**

<sup>22</sup>[http://www.google.com.ec/webhp?source=search\\_app#hl=es&tbo=d&scient=psy-ab&q=comandos+AT&oq=comandos+AT&gs\\_l=serp.3..0l4.3252.3560.7.4174.2.2.0.0.0.194.194.0j1.1.0...0.0...1c.1.W7-wduqyayk&pbx=1&bav=on.2,or.r\\_gc.r\\_pw.r\\_qf.&fp=c97bc84180386383&bpcl=39650382&biw=1236&bih=584](http://www.google.com.ec/webhp?source=search_app#hl=es&tbo=d&scient=psy-ab&q=comandos+AT&oq=comandos+AT&gs_l=serp.3..0l4.3252.3560.7.4174.2.2.0.0.0.194.194.0j1.1.0...0.0...1c.1.W7-wduqyayk&pbx=1&bav=on.2,or.r_gc.r_pw.r_qf.&fp=c97bc84180386383&bpcl=39650382&biw=1236&bih=584)

**OTRAS REFERENCIAS:****ESPECIFICACIONES TÉCNICAS DEL ZTE**

<http://www.propox.com/download/docs/GSM-ZME3431.pdf>

**MANUAL DE COMANDOS DEL ZTE**

<http://download.maritex.com.pl/pdfs/wi/ME3030.pdf>

**GUÍA DEL USUARIO DELDESARROLLO DEL MÓDULO ZTE**

<http://download.maritex.com.pl/pdfs/wi/ZME-EVK.pdf>

**MANUAL DEL MÓDULO ZTE 3006 APM**

[http://www.forwellwireless.com/download/D1\\_Usermanual\\_Eng.pdf](http://www.forwellwireless.com/download/D1_Usermanual_Eng.pdf)

## **GLOSARIO**

### **BTS**

Estación Transceptora base

### **CDMA**

Acceso Múltiple por división de Código

### **Comandos AT**

Sirven de interfaz para configurar y proporcionar instrucciones a los terminales para tecnología GSM y módems.

### **CPU**

Central Processing Unit (CPU/Unidad Central de Procesamiento) componente principales de dispositivos programables.

### **Darlington**

En electrónica, el transistor Darlington es un dispositivo semiconductor que combina dos transistores bipolares.

### **Diodo**

Un diodo es un componente electrónico de dos terminales que permite la circulación de la corriente eléctrica a través de él en un solo sentido.

### **Domótica**

Se entiende por domótica el conjunto de sistemas capaces de automatizar una vivienda.

### **EcuadorTelecom S.A.**

EcuadorTelecom S.A. (antes también conocida como Ecutel) es la representación jurídica de Claro en Ecuador. Ha tenido una gran expansión desde su ingreso al país prestando servicios de calidad como: televisión digital, internet y telefonía IP.

**GMSC**

Centro de Conmutación de servicio móvil y Gateway

**GSM**

“Sistema Global para las Comunicaciones Móviles”, anteriormente conocida como "Group Special Mobile" (GSM, Grupo Especial Móvil) es un estándar mundial para teléfonos móviles digitales.

**LCD**

Pantalla de Cristal Líquido

**Microcontroladores**

Un microcontrolador (abreviado  $\mu\text{C}$ , UC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria.

**MIPS**

Millones de instrucciones por segundo, forma de medir potencia de los procesadores.

**SMS**

El servicio de mensajes cortos o SMS (Short Message Service) es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos.

**MSC**

Centro de Conmutación de servicio móvil

**Tarjeta SIM**

La Tarjeta SIM (Módulo de Identificación del Suscriptor).

**TDMA**

La multiplexación por división de tiempo (Time División Múltiple Access o TDMA) es una técnica que permite la transmisión de señales digitales.

**Tribanda**

Un teléfono tribanda, conocido también como trimodo, es un dispositivo que soporta las bandas GSM de 900/1800/1900 MHz (usualmente vendidos en Europa Asia y África), o las bandas de 850/1800/1900MHz (usadas en América).

**Pic (microcontrolador)**

Los PIC son una familia de microcontroladores tipo RISC fabricados por Microchip Technology Inc. y derivados del PIC1650.

**Pixeles**

Un píxel o pixel, plural píxeles ("elemento de imagen") es la menor unidad homogénea en color que forma parte de una imagen digital.

**VLR**

Registro de localización Local

**ANEXOS**

**ANEXO A:**DATASHEET ATmega164P

**ANEXO B:** DATASHEET TLP434A&RLP434A

**ANEXO C:** DATASHEET MAX232

**ANEXO D:** MANUAL MÓDULO GSM ZTE

**ANEXO E:**DATASHEET ULN2003A

## **ANEXO A**

DATASHEET ATmega164P



## Features

- High-performance, Low-power AVR<sup>®</sup> 8-bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 × 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 16/32/64K Bytes of In-System Self-Programmable Flash  
Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits  
In-System Programming by On-chip Boot Program  
True Read-While-Write Operation
  - 512B/1K/2K Bytes EEPROM  
Endurance: 100,000 Write/Erase Cycles
  - 1/2/4K Bytes Internal SRAM
  - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel, 10-bit ADC  
Differential mode with selectable gain at 1x, 10x or 200x<sup>(1)</sup>
  - Byte-oriented Two-wire Serial Interface
  - Two Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
  - 32 Programmable I/O Lines
  - 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
  - 2.7 - 5.5V for ATmega164P/324P/644P
- Speed Grades
  - ATmega164P/324P/644P: 0 - 8MHz @ 2.7 - 5.5V, 0 - 16MHz @ 4.5 - 5.5V
- Power Consumption at 8 MHz, 5V, 25°C for ATmega644P
  - Active mode: 8 mA
  - Idle mode: 2.4 mA
  - Power-down Mode: 0.8 µA



**8-bit AVR<sup>®</sup>  
Microcontroller  
with 16/32/64K  
Bytes In-System  
Programmable  
Flash**

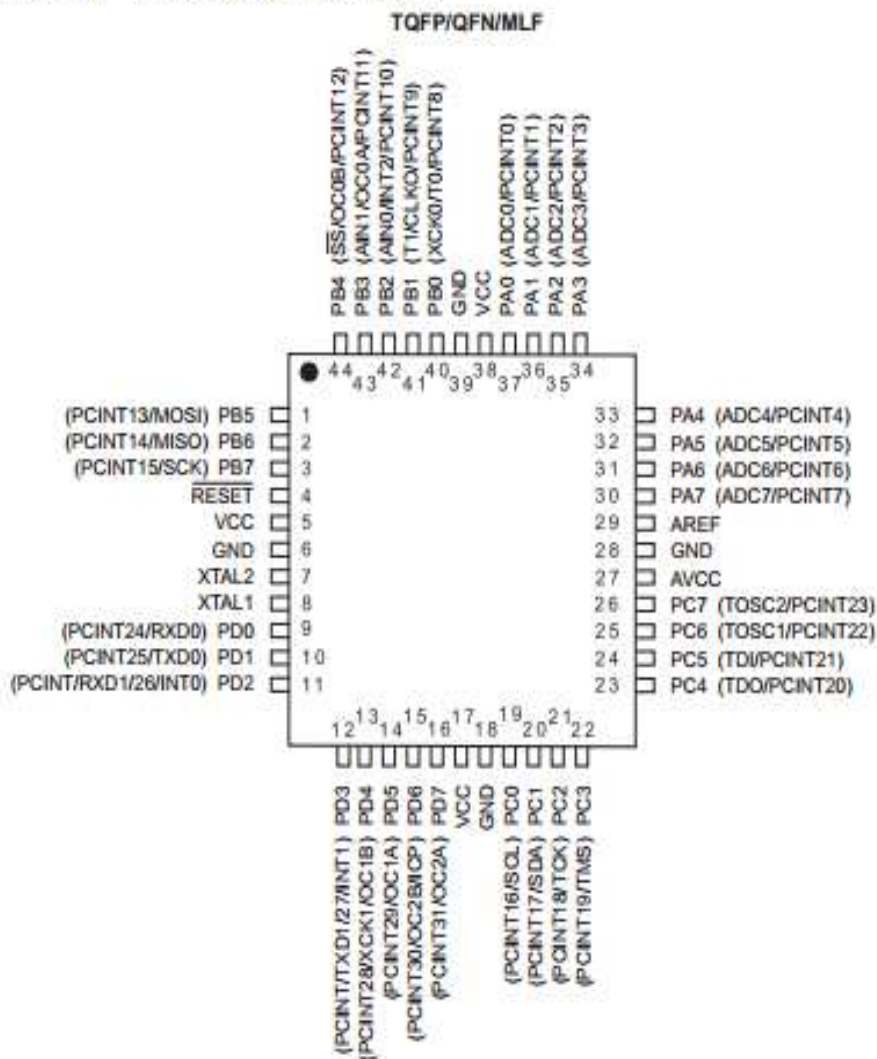
**ATmega164P  
ATmega324P  
ATmega644P**

**Automotive**



## 1. Pin Configurations

Figure 1-1. Pinout ATmega164P/324P/644P



Note: The large center pad underneath the QFN/MLF package should be soldered to ground on the board to ensure good mechanical stability.

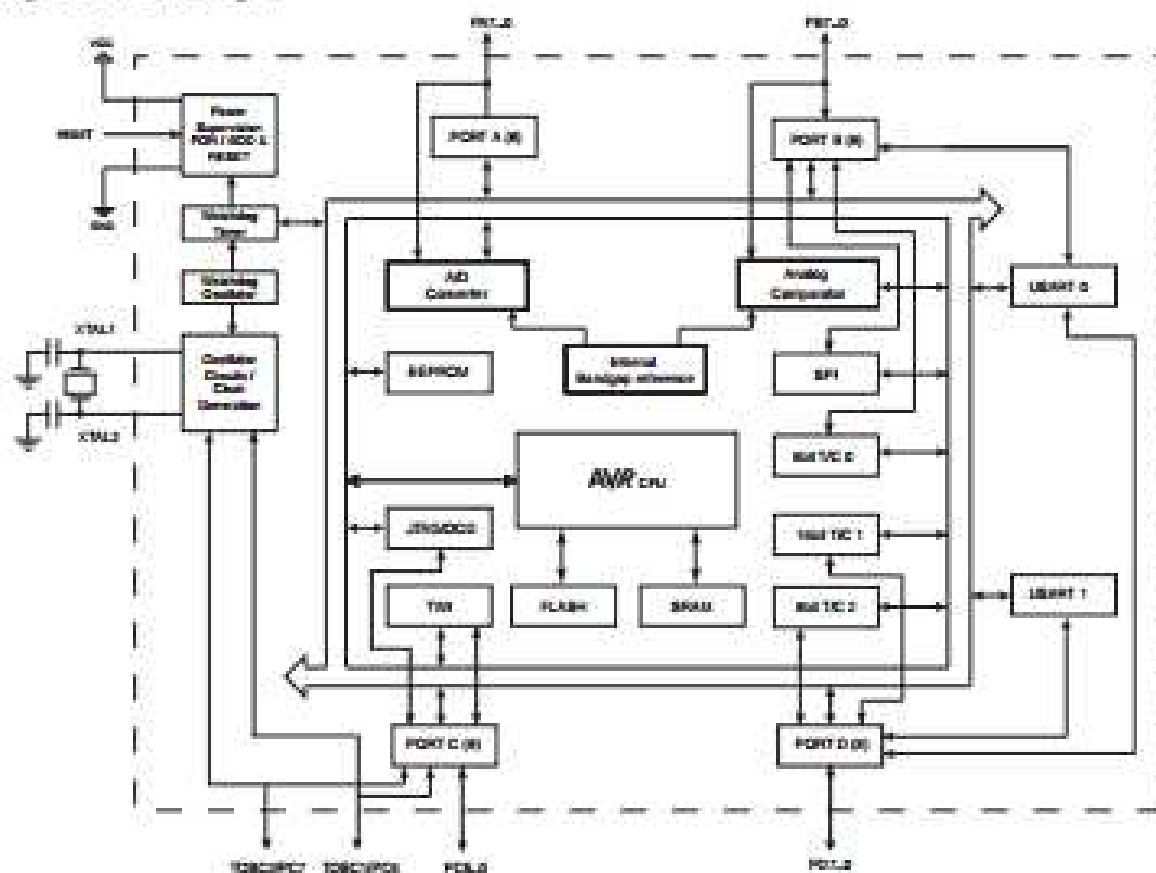
## ATmega164P/324P/644P

### 2. Overview

The ATmega164P/324P/644P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega164P/324P/644P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

#### 2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.





The ATmega164P/324P/644P provides the following features: 16/32/64K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512B/1K/2K bytes EEPROM, 1/2/4K bytes SRAM, 32 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), three flexible Timer/Counters with compare modes and PWM, 2 USARTs, a byte oriented 2-wire Serial Interface, a 8-channel, 10-bit ADC with optional differential input stage with programmable gain, programmable Watchdog Timer with Internal Oscillator, an SPI serial port, IEEE std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a time base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega164P/324P/644P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega164P/324P/644P AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

## 2.2 Comparison Between ATmega164P, ATmega324P and ATmega644P

Table 2-1. Differences between ATmega164P and ATmega644P

Device	Flash	EEPROM	RAM
ATmega164P	16 Kbyte	512 Bytes	1 Kbyte
ATmega324P	32 Kbyte	1 Kbyte	2 Kbyte
ATmega644P	64 Kbyte	2 Kbyte	4 Kbyte

## ATmega164P/324P/644P

### 2.2.1 Automotive Quality Grade

The ATmega164P/324P/644P have been developed and manufactured according to the most stringent requirements of the international standard ISO-TS-16949. This data sheet contains limit values extracted from the results of extensive characterization (Temperature and Voltage). The quality and reliability of the ATmega164P/324P/644P have been verified during regular product qualification as per AEC-Q100 grade 1 (–40°C to +125°C).

Table 2-2. Temperature Grade Identification for Automotive Products

Temperature	Temperature Identifier	Comments
–40 ; +125	Z	Full Automotive Temperature Range

## 2.3 Pin Descriptions

### 2.3.1 VCC

Digital supply voltage.

### 2.3.2 GND

Ground.

### 2.3.3 Port A (PA7:PA0)

Port A serves as analog inputs to the Analog-to-digital Converter.

Port A also serves as an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port A pins that are externally pulled low will source current if the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port A also serves the functions of various special features of the ATmega164P/324P/644P as listed on [page 79](#).

### 2.3.4 Port B (PB7:PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also serves the functions of various special features of the ATmega164P/324P/644P as listed on [page 81](#).





- 2.3.5 Port C (PC7:PC0)**
- Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.
- Port C also serves the functions of the JTAG interface, along with special features of the ATmega164P/324P/644P as listed on [page 84](#).
- 2.3.6 Port D (PD7:PD0)**
- Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.
- Port D also serves the functions of various special features of the ATmega164P/324P/644P as listed on [page 88](#).
- 2.3.7 RESET**
- Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in "System and Reset Characteristics" on [page 332](#). Shorter pulses are not guaranteed to generate a reset.
- 2.3.8 XTAL1**
- Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
- 2.3.9 XTAL2**
- Output from the inverting Oscillator amplifier.
- 2.3.10 AVCC**
- AVCC is the supply voltage pin for Port F and the Analog-to-digital Converter. It should be externally connected to  $V_{CC}$ , even if the ADC is not used. If the ADC is used, it should be connected to  $V_{CC}$  through a low-pass filter.
- 2.3.11 AREF**
- This is the analog reference pin for the Analog-to-digital Converter.

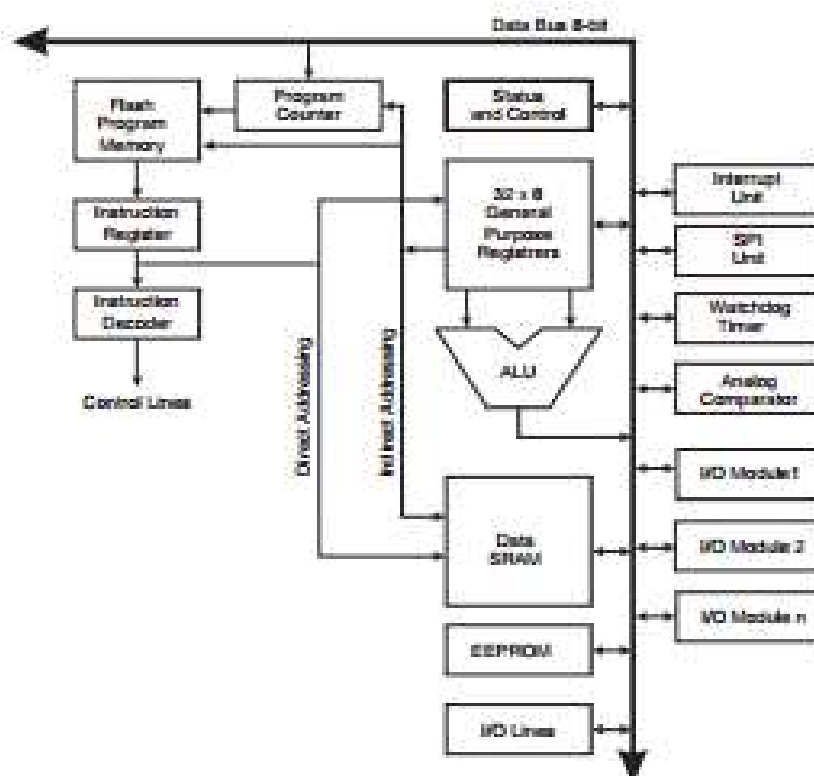
## ATmega164P/324P/644P

### 5. AVR CPU Core

#### 5.1 Overview

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

Figure 5-1. Block Diagram of the AVR Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory.





The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of these address pointers can also be used as an address pointer for look up tables in Flash program memory. These added function registers are the 16-bit X-, Y-, and Z-register, described later in this section.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

Program Flash memory space is divided in two sections, the Boot Program section and the Application Program section. Both sections have dedicated Lock bits for write and read/write protection. The SPM instruction that writes into the Application Flash memory section must reside in the Boot Program section.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM, and consequently the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the Reset routine (before subroutines or interrupts are executed). The Stack Pointer (SP) is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

A flexible interrupt module has its control registers in the I/O space with an additional Global Interrupt Enable bit in the Status Register. All interrupts have a separate Interrupt Vector in the Interrupt Vector table. The interrupts have priority in accordance with their Interrupt Vector position. The lower the Interrupt Vector address, the higher the priority.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File, 0x20 - 0x5F. In addition, the ATmega164P/324P/644P has Extended I/O space from 0x60 - 0xFF in SRAM where only the ST/STS/STD and LD/ LDS/ LDD instructions can be used.

## 5.2 ALU – Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. See the "Instruction Set" section for a detailed description.





## 6. AVR Memories

### 6.1 Overview

This section describes the different memories in the ATmega164P/324P/644P. The AVR architecture has two main memory spaces, the Data Memory and the Program Memory space. In addition, the ATmega164P/324P/644P features an EEPROM Memory for data storage. All three memory spaces are linear and regular.

### 6.2 In-System Reprogrammable Flash Program Memory

The ATmega164P/324P/644P contains 16/32/64K bytes On-chip In-System Reprogrammable Flash memory for program storage. Since all AVR instructions are 16 or 32 bits wide, the Flash is organized as 32/64 x 16. For software security, the Flash Program memory space is divided into two sections, Boot Program section and Application Program section.

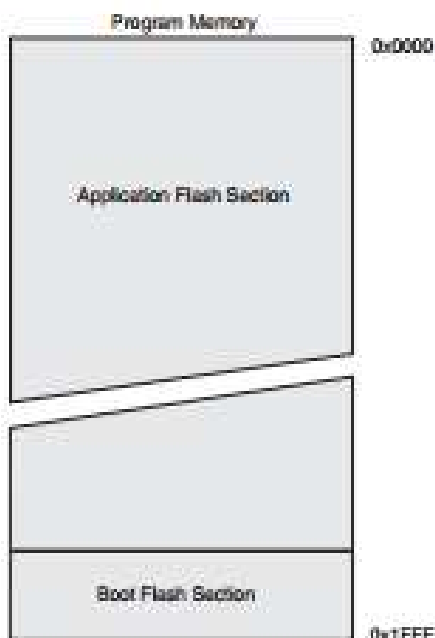
The Flash memory has an endurance of at least 10,000 write/erase cycles. The ATmega164P/324P/644P Program Counter (PC) is 15/16 bits wide, thus addressing the 32/64K program memory locations. The operation of Boot Program section and associated Boot Lock bits for software protection are described in detail in ["Memory Programming" on page 296](#). ["Memory Programming" on page 296](#) contains a detailed description on Flash data serial downloading using the SPI pins or the JTAG interface.

Constant tables can be allocated within the entire program memory address space (see the LPM – Load Program Memory Instruction description).

Timing diagrams for instruction fetch and execution are presented in ["Instruction Execution Timing" on page 14](#).

## ATmega164P/324P/644P

Figure 6-1. Program Memory Map



### 6.3 SRAM Data Memory

Figure 6-2 shows how the ATmega164P/324P/644P SRAM Memory is organized.

The ATmega164P/324P/644P is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in the Opcode for the IN and OUT instructions. For the Extended I/O space from \$060 - \$FF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

The first 4,352 Data Memory locations address both the Register File, the I/O Memory, Extended I/O Memory, and the internal data SRAM. The first 32 locations address the Register file, the next 64 location the standard I/O Memory, then 160 locations of Extended I/O memory and the next 4,096 locations address the internal data SRAM.

The five different addressing modes for the data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-decrement, and Indirect with Post-increment. In the Register file, registers R26 to R31 feature the indirect addressing pointer registers.

The direct addressing reaches the entire data space.

The Indirect with Displacement mode reaches 63 address locations from the base address given by the Y- or Z-register.

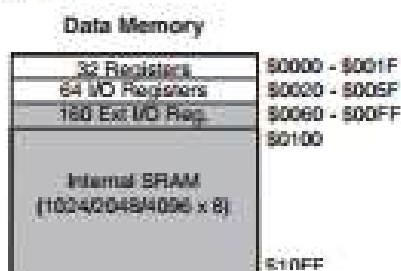
When using register indirect addressing modes with automatic pre-decrement and post-increment, the address registers X, Y, and Z are decremented or incremented.





The 32 general purpose working registers, 64 I/O registers, 160 Extended I/O Registers and the 1024/2048/4096 bytes of internal data SRAM in the ATmega164P/324P/644P are all accessible through all these addressing modes. The Register File is described in "General Purpose Register File" on page 12.

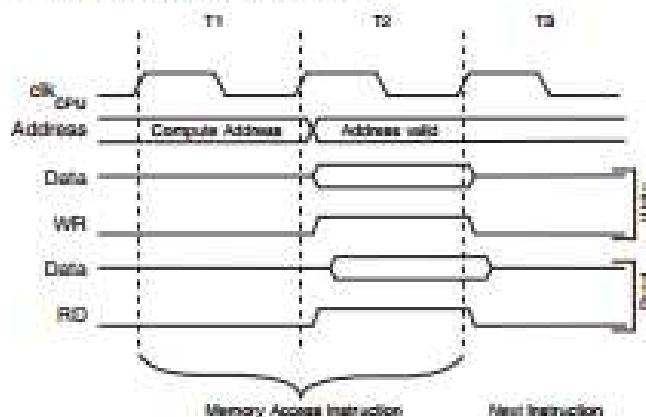
Figure 6-2. Data Memory Map



### 6.3.1 Data Memory Access Times

This section describes the general access timing concepts for internal memory access. The internal data SRAM access is performed in two  $clk_{CPU}$  cycles as described in Figure 6-3.

Figure 6-3. On-chip Data SRAM Access Cycles



## ATmega164P/324P/644P

### 6.4 EEPROM Data Memory

The ATmega164P/324P/644P contains 512B/1K/2K bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described in the following, specifying the EEPROM Address Registers, the EEPROM Data Register, and the EEPROM Control Register.

For a detailed description of SPI, JTAG and Parallel data downloading to the EEPROM, see [page 311](#), [page 315](#), and [page 300](#) respectively.

#### 6.4.1 EEPROM Read/Write Access

The EEPROM Access Registers are accessible in the I/O space. See ["Register Description" on page 23](#) for details.

The write access time for the EEPROM is given in [Table 6-2 on page 25](#). A self-timing function, however, lets the user software detect when the next byte can be written. If the user code contains instructions that write the EEPROM, some precautions must be taken. In heavily filtered power supplies,  $V_{CC}$  is likely to rise or fall slowly on power-up/down. This causes the device for some period of time to run at a voltage lower than specified as minimum for the clock frequency used. See ["Preventing EEPROM Corruption" on page 21](#), for details on how to avoid problems in these situations.

In order to prevent unintentional EEPROM writes, a specific write procedure must be followed. Refer to the description of the EEPROM Control Register for details on this.

When the EEPROM is read, the CPU is halted for four clock cycles before the next instruction is executed. When the EEPROM is written, the CPU is halted for two clock cycles before the next instruction is executed.

#### 6.4.2 Preventing EEPROM Corruption

During periods of low  $V_{CC}$ , the EEPROM data can be corrupted because the supply voltage is too low for the CPU and the EEPROM to operate properly. These issues are the same as for board level systems using EEPROM, and the same design solutions should be applied.

An EEPROM data corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the EEPROM requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions incorrectly, if the supply voltage is too low.

EEPROM data corruption can easily be avoided by following this design recommendation:

Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This can be done by enabling the internal Brown-out Detector (BOD). If the detection level of the internal BOD does not match the needed detection level, an external low  $V_{CC}$  reset Protection circuit can be used. If a reset occurs while a write operation is in progress, the write operation will be completed provided that the power supply voltage is sufficient.





## 6.5 I/O Memory

The I/O space definition of the ATmega164P/324P/644P is shown in [“Register Summary” on page 358](#).

All ATmega164P/324P/644P I/Os and peripherals are placed in the I/O space. All I/O locations may be accessed by the LD/LDS/LDD and ST/STS/STD instructions, transferring data between the 32 general purpose working registers and the I/O space. I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. Refer to the instruction set section for more details. When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATmega164P/324P/644P is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcodes for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

Some of the Status Flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such Status Flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

The I/O and peripherals control registers are explained in later sections.

The ATmega164P/324P/644P contains three General Purpose I/O Registers, see [“Register Description” on page 23](#). These registers can be used for storing any information, and they are particularly useful for storing global variables and Status Flags. General Purpose I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI, CBI, SBIS, and SBIC instructions.

## ATmega164P/324P/644P

### 6.6 Register Description

#### 6.6.1 EEARH and EEARL – The EEPROM Address Register

Bit	15	14	13	12	11	10	9	8	
Out0 (Out0)	–	–	–	–	EEAR11	EEAR10	EEAR9	EEAR8	EEARH
Out1 (Out1)	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
Read/Write	R	R	R	R	R	RW	RW	RW	RW
Initial Value	0	0	0	0	X	X	X	X	
	X	X	X	X	X	X	X	X	

- **Bits 15:12 – Res: Reserved Bits**

These bits are reserved bits in the ATmega164P/324P/644P and will always read as zero.

- **Bits 11:0 – EEAR8:0: EEPROM Address**

The EEPROM Address Registers – EEARH and EEARL specify the EEPROM address in the 4K bytes EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 4096. The initial value of EEAR is undefined. A proper value must be written before the EEPROM may be accessed.

#### 6.6.2 EEDR – The EEPROM Data Register

Bit	7	6	5	4	3	2	1	0	
Out0 (Out0)	MSB							LSB	EEDR
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:0 – EEDR7:0: EEPROM Data**

For the EEPROM write operation, the EEDR Register contains the data to be written to the EEPROM in the address given by the EEAR Register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.

#### 6.6.3 EECR – The EEPROM Control Register

Bit	7	6	5	4	3	2	1	0	
Out0 (Out0)	–	–	EEPM1	EEPM0	EEPE	EEPGS	LOPE	EECR2	EECR
Read/Write	R	R	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	X	1	0	0	1	0	

- **Bits 7:6 – Res: Reserved Bits**

These bits are reserved bits in the ATmega164P/324P/644P and will always read as zero.

- **Bits 5:4 – EEPM1 and EEPM0: EEPROM Programming Mode Bits**

The EEPROM Programming mode bit setting defines which programming action that will be triggered when writing EEPE. It is possible to program data in one atomic operation (erase the old value and program the new value) or to split the Erase and Write operations in two different operations. The Programming times for the different modes are shown in [Table 6-1 on page 24](#).





While EEPE is set, any write to EEP Mn will be ignored. During reset, the EEP Mn bits will be reset to 0b00 unless the EEPROM is busy programming.

Table 6-1. EEPROM Mode Bits

EEPMT	EEPMD	Programming Time	Operation
0	0	3.4 ms	Erase and Write in one operation (Atomic Operation)
0	1	1.8 ms	Erase Only
1	0	1.8 ms	Write Only
1	1	–	Reserved for future use

• **Bit 3 – EERIE: EEPROM Ready Interrupt Enable**

Writing EERIE to one enables the EEPROM Ready Interrupt if the I bit in SREG is set. Writing EERIE to zero disables the interrupt. The EEPROM Ready interrupt generates a constant interrupt when EEPE is cleared.

• **Bit 2 – EEMPE: EEPROM Master Programming Enable**

The EEMPE bit determines whether setting EEPE to one causes the EEPROM to be written. When EEMPE is set, setting EEPE within four clock cycles will write data to the EEPROM at the selected address. If EEMPE is zero, setting EEPE will have no effect. When EEMPE has been written to one by software, hardware clears the bit to zero after four clock cycles. See the description of the EEPE bit for an EEPROM write procedure.

• **Bit 1 – EEPE: EEPROM Programming Enable**

The EEPROM Write Enable Signal EEPE is the write strobe to the EEPROM. When address and data are correctly set up, the EEPE bit must be written to one to write the value into the EEPROM. The EEMPE bit must be written to one before a logical one is written to EEPE, otherwise no EEPROM write takes place. The following procedure should be followed when writing the EEPROM (the order of steps 3 and 4 is not essential):

1. Wait until EEPE becomes zero.
2. Wait until SELFPRGEN in SP\_MCSR becomes zero.
3. Write new EEPROM address to EEAR (optional).
4. Write new EEPROM data to EEDR (optional).
5. Write a logical one to the EEMPE bit while writing a zero to EEPE in EECR.
6. Within four clock cycles after setting EEMPE, write a logical one to EEPE.

The EEPROM can not be programmed during a CPU write to the Flash memory. The software must check that the Flash programming is completed before initiating a new EEPROM write. Step 2 is only relevant if the software contains a Boot Loader allowing the CPU to program the Flash. If the Flash is never being updated by the CPU, step 2 can be omitted. See [“Memory Programming” on page 296](#) for details about Boot programming.

**Caution:** An interrupt between step 5 and step 6 will make the write cycle fail, since the EEPROM Master Write Enable will time-out. If an interrupt routine accessing the EEPROM is interrupting another EEPROM access, the EEAR or EEDR Register will be modified, causing the interrupted EEPROM access to fail. It is recommended to have the Global Interrupt Flag cleared during all the steps to avoid these problems.

## ATmega164P/324P/644P

When the write access time has elapsed, the EEPB bit is cleared by hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEPB has been set, the CPU is halted for two cycles before the next instruction is executed.

- **Bit 0 – EERE: EEPROM Read Enable**

The EEPROM Read Enable Signal EERE is the read strobe to the EEPROM. When the correct address is set up in the EEAR Register, the EERE bit must be written to a logic one to trigger the EEPROM read. The EEPROM read access takes one instruction, and the requested data is available immediately. When the EEPROM is read, the CPU is halted for four cycles before the next instruction is executed.

The user should poll the EEPB bit before starting the read operation. If a write operation is in progress, it is neither possible to read the EEPROM, nor to change the EEAR Register.

The calibrated Oscillator is used to time the EEPROM accesses. [Table 6-2 on page 25](#) lists the typical programming time for EEPROM access from the CPU.

**Table 6-2. EEPROM Programming Time**

Symbol	Number of Calibrated RC Oscillator Cycles	Typ Programming Time
EEPROM write (from CPU)	26,368	3.3 ms







The following code examples show one assembly and one C function for writing to the EEPROM. The examples assume that interrupts are controlled (e.g. by disabling interrupts globally) so that no interrupts will occur during execution of these functions. The examples also assume that no Flash Boot Loader is present in the software. If such code is present, the EEPROM write function must also wait for any ongoing SPM command to finish.

#### Assembly Code Example<sup>1)</sup>

```
EEPROM_write:
    ; Wait for completion of previous write
    sbic EECR,EEPE
    rjmp EEPROM_write
    ; Set up address (r16:r17) in address register
    out EEARH, r16
    out EEARL, r17
    ; Write data (r16) to Data Register
    out EEDR,r16
    ; Write logical one to EEMPE
    sbi EECR,EEMPE
    ; Start eeprom write by setting EEPF
    sbi EECR,EEPF
    ret
```

#### C Code Example<sup>1)</sup>

```
void EEPROM_write(unsigned int uiAddress, unsigned char ucData)
{
    /* Wait for completion of previous write */
    while(EECR & (1<<EEPE))
    {
    }
    /* Set up address and Data Registers */
    EEAR = uiAddress;
    EEDR = ucData;
    /* Write logical one to EEMPE */
    EECR |= (1<<EEMPE);
    /* Start eeprom write by setting EEPF */
    EECR |= (1<<EEPF);
}
```

Note: 1. See "About Code Examples" on page 8.

## ATmega164P/324P/644P

The next code examples show assembly and C functions for reading the EEPROM. The examples assume that interrupts are controlled so that no interrupts will occur during execution of these functions.

### Assembly Code Example<sup>1)</sup>

```
EEPROM_read:
    ; Wait for completion of previous write
    sbic EECR,EEPE
    rjmp EEPROM_read
    ; Set up address (r16/r17) in address register
    out EEARH, r16
    out EEARL, r17
    ; Start eeprom read by writing EERE
    sbi EECR,EESE
    ; Read data from Data Register
    in r14,EEDR
    ret
```

### C Code Example<sup>1)</sup>

```
unsigned char EEPROM_read(unsigned int uiAddress)
{
    /* Wait for completion of previous write */
    while(EECR & (1<<EEPE))
        ;
    /* Set up address register */
    EEAR = uiAddress;
    /* Start eeprom read by writing EERE */
    EECR |= (1<<EESE);
    /* Return data from Data Register */
    return EEDR;
}
```

Note: 1. See "About Code Examples" on page 8.





#### 6.6.4 GPIOR2 – General Purpose IO Register 2

IO	7	6	5	4	3	2	1	0	GPIOR2
IO/IO (IO/IO)	Input	Input	Input	Input	Input	Input	Input	Input	
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

#### 6.6.5 GPIOR1 – General Purpose IO Register 1

IO	7	6	5	4	3	2	1	0	GPIOR1
IO/IO (IO/IO)	Input	Input	Input	Input	Input	Input	Input	Input	
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

#### 6.6.6 GPIOR0 – General Purpose IO Register 0

IO	7	6	5	4	3	2	1	0	GPIOR0
IO/IO (IO/IO)	Input	Input	Input	Input	Input	Input	Input	Input	
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

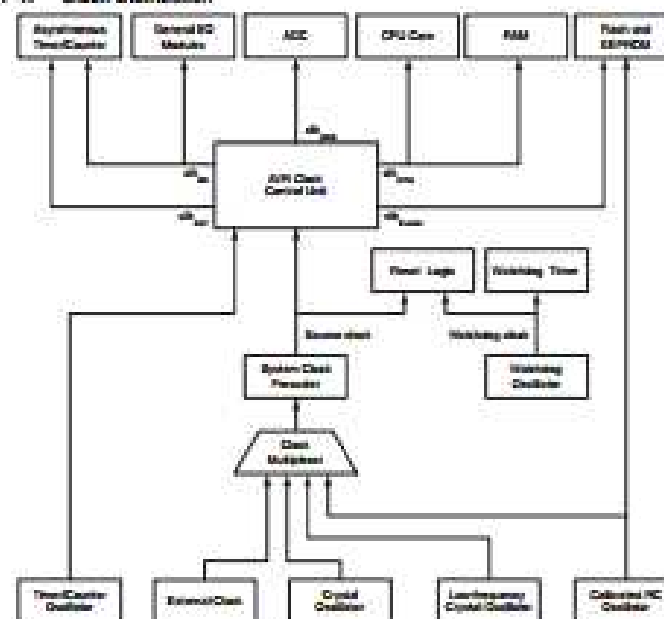
Note: 1. SPWn1 = SPW11 (upper sector) or SPW01 (lower sector), SPWn0 = SPW10 (upper sector) or SPW00 (lower sector). The ALE pulse in period T4 is only present if the next instruction accesses the RAM (internal or external).

## 7. System Clock and Clock Options

### 7.1 Clock Systems and their Distribution

Figure 7-1 presents the principal clock systems in the AVR and their distribution. All of the clocks need not be active at a given time. In order to reduce power consumption, the clocks to modules not being used can be halted by using different sleep modes, as described in "Power Management and Sleep Modes" on page 41. The clock systems are detailed below.

Figure 7-1. Clock Distribution



#### 7.1.1 CPU Clock – $clk_{CPU}$

The CPU clock is routed to parts of the system concerned with operation of the AVR core. Examples of such modules are the General Purpose Register File, the Status Register and the data memory holding the Stack Pointer. Halting the CPU clock inhibits the core from performing general operations and calculations.

#### 7.1.2 I/O Clock – $clk_{IO}$

The I/O clock is used by the majority of the I/O modules, like Time/Counters, SPI, and USART. The I/O clock is also used by the External Interrupt module, but note that some external interrupts are detected by asynchronous logic, allowing such interrupts to be detected even if the I/O clock is halted. Also note that start condition detection in the USI module is carried out asynchronously when  $clk_{IO}$  is halted, TWI address recognition in all sleep modes.





### 7.1.3 Flash Clock – $clk_{FLASH}$

The Flash clock controls operation of the Flash interface. The Flash clock is usually active simultaneously with the CPU clock.

### 7.1.4 Asynchronous Timer Clock – $clk_{ASY}$

The Asynchronous Timer clock allows the Asynchronous Timer/Counter to be clocked directly from an external clock or an external 32 kHz clock crystal. The dedicated clock domain allows using this Timer/Counter as a real-time counter even when the device is in sleep mode.

### 7.1.5 ADC Clock – $clk_{ADC}$

The ADC is provided with a dedicated clock domain. This allows halting the CPU and I/O clocks in order to reduce noise generated by digital circuitry. This gives more accurate ADC conversion results.

## 7.2 Clock Sources

The device has the following clock source options, selectable by Flash Fuse bits as shown below. The clock from the selected source is input to the AVR clock generator, and routed to the appropriate modules.

Table 7-1. Device Clocking Options Select<sup>(1)</sup>

Device Clocking Option	CKSEL3..0
Low Power Crystal Oscillator	1111 - 1000
Full Swing Crystal Oscillator	0111 - 0110
Low Frequency Crystal Oscillator	0101 - 0100
Internal 128 kHz RC Oscillator	0011
Calibrated Internal RC Oscillator	0010
External Clock	0000
Reserved	0001

Note: 1. For all fuses "1" means unprogrammed while "0" means programmed.

### 7.2.1 Default Clock Source

The device is shipped with internal RC oscillator at 8.0MHz and with the fuse CKDIV8 programmed, resulting in 1.0MHz system clock. The startup time is set to maximum and time-out period enabled. (CKSEL = "0010", SUT = "10", CKDIV8 = "0"). The default setting ensures that all users can make their desired clock source setting using any available programming interface.

## ATmega164P/324P/644P

### 7.2.2 Clock Startup Sequence

Any clock source needs a sufficient  $V_{CC}$  to start oscillating and a minimum number of oscillating cycles before it can be considered stable.

To ensure sufficient  $V_{CC}$ , the device issues an internal reset with a time-out delay ( $t_{ROUT}$ ) after the device reset is released by all other reset sources. "On-chip Debug System" on page 45 describes the start conditions for the internal reset. The delay ( $t_{ROUT}$ ) is timed from the Watchdog Oscillator and the number of cycles in the delay is set by the SUTx and CKSELx fuse bits. The selectable delays are shown in Table 7-2. The frequency of the Watchdog Oscillator is voltage dependent as shown in Section 27, "ATmega644P Typical Characteristics" on page 338.

Table 7-2. Number of Watchdog Oscillator Cycles

Typ Time-out ( $V_{CC} = 5.0V$ )	Typ Time-out ( $V_{CC} = 3.0V$ )	Number of Cycles
0 ms	0 ms	0
4.1 ms	4.3 ms	512
65 ms	69 ms	8K (8,192)

Main purpose of the delay is to keep the AVR in reset until it is supplied with minimum  $V_{CC}$ . The delay will not monitor the actual voltage and it will be required to select a delay longer than the  $V_{CC}$  rise time. If this is not possible, an internal or external Brown-Out Detection circuit should be used. A BOD circuit will ensure sufficient  $V_{CC}$  before it releases the reset, and the time-out delay can be disabled. Disabling the time-out delay without utilizing a Brown-Out Detection circuit is not recommended.

The oscillator is required to oscillate for a minimum number of cycles before the clock is considered stable. An internal ripple counter monitors the oscillator output clock, and keeps the internal reset active for a given number of clock cycles. The reset is then released and the device will start to execute. The recommended oscillator start-up time is dependent on the clock type, and varies from 6 cycles for an externally applied clock to 32K cycles for a low frequency crystal.

The start-up sequence for the clock includes both the time-out delay and the start-up time when the device starts up from reset. When starting up from Power-save or Power-down mode,  $V_{CC}$  is assumed to be at a sufficient level and only the start-up time is included.

### 7.2.3 Clock Source Connections

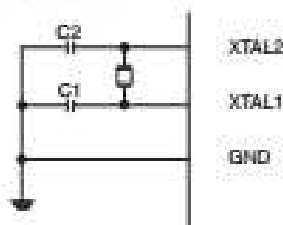
The pins XTAL1 and XTAL2 are input and output, respectively, of an inverting amplifier which can be configured for use as an On-chip Oscillator, as shown in Figure 7-2 on page 32. Either a quartz crystal or a ceramic resonator may be used.

C1 and C2 should always be equal for both crystals and resonators. The optimal value of the capacitors depends on the crystal or resonator in use, the amount of stray capacitance, and the electromagnetic noise of the environment. For ceramic resonators, the capacitor values given by the manufacturer should be used.





Figure 7-2. Crystal Oscillator Connections



### 7.3 Low Power Crystal Oscillator

This Crystal Oscillator is a low power oscillator, with reduced voltage swing on the XTAL2 output. It gives the lowest power consumption, but is not capable of driving other clock inputs, and may be more susceptible to noise in noisy environments. In these cases, refer to the [“Full Swing Crystal Oscillator”](#) on page 33.

Some initial guidelines for choosing capacitors for use with crystals are given in [Table 7-3](#). The crystal should be connected as described in [“Clock Source Connections”](#) on page 31.

The Low Power Oscillator can operate in three different modes, each optimized for a specific frequency range. The operating mode is selected by the fuses CKSEL3..1 as shown in [Table 7-3](#).

Table 7-3. Low Power Crystal Oscillator Operating Modes<sup>1,2</sup>

Frequency Range <sup>3</sup> (MHz)	CKSEL3..1	Recommended Range for Capacitors C1 and C2 (pF)
0.4 - 0.9	100 <sup>2</sup>	—
0.9 - 3.0	101	12 - 22
3.0 - 8.0	110	12 - 22
8.0 - 16.0	111	12 - 22

- Notes:
1. The frequency ranges are preliminary values. Actual values are TBD.
  2. This option should not be used with crystals, only with ceramic resonators.
  3. If 8 MHz frequency exceeds the specification of the device (depends on  $V_{DD}$ ), the CKDIV8 Fuse can be programmed in order to divide the internal frequency by 8. It must be ensured that the resulting divided clock meets the frequency specification of the device.

The CKSEL0 Fuse together with the SUT1..0 Fuses select the start-up times as shown in [Table 7-4](#).

## 17. USART

### 17.1 Features

- Full Duplex Operation (Independent Serial Receive and Transmit Registers)
- Asynchronous or Synchronous Operation
- Master or Slave Clocked Synchronous Operation
- High Resolution Baud Rate Generator
- Supports Serial Frames with 5, 6, 7, 8, or 9 Data Bits and 1 or 2 Stop Bits
- Odd or Even Parity Generation and Parity Check Supported by Hardware
- Data Overflow Detection
- Framing Error Detection
- Noise Filtering Includes False Start Bit Detection and Digital Low Pass Filter
- Three Separate Interrupts on TX Complete, TX Data Register Empty and RX Complete
- Multi-processor Communication Mode
- Double Speed Asynchronous Communication Mode

### 17.2 USART1 and USART0

The ATmega164P/324P/644P has two USART's, USART0 and USART1.

The functionality for all USART's is described below, most register and bit references in this section are written in general form. A lower case "n" replaces the USART number.

USART0 and USART1 have different I/O registers as shown in ["Register Summary" on page 356](#).

### 17.3 Overview

The Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) is a highly flexible serial communication device.

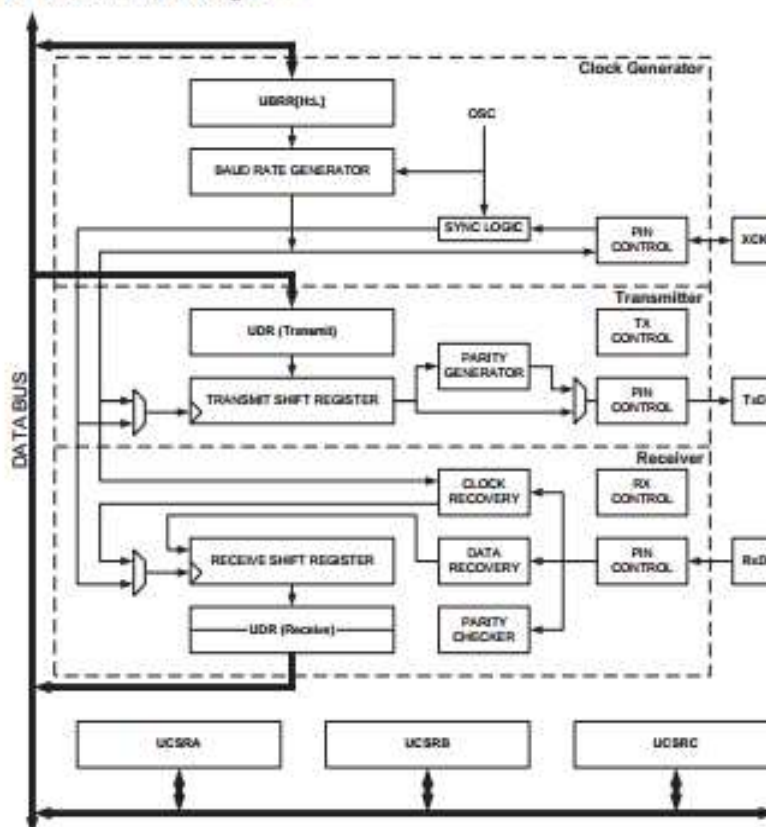
A simplified block diagram of the USART Transmitter is shown in [Figure 17-1 on page 172](#). CPU accessible I/O Registers and I/O pins are shown in bold.

The Power Reduction USART0 bit, PRUSART0, in ["PRR – Power Reduction Register" on page 47](#) must be disabled by writing a logical zero to it.

The Power Reduction USART1 bit, PRUSART1, in ["PRR – Power Reduction Register" on page 47](#) must be disabled by writing a logical zero to it.





Figure 17-1. USART Block Diagram<sup>(1)</sup>

Note: 1. See [Figure 1-1](#) on page 2 and ["Alternate Port Functions"](#) on page 77 for USART pin placement.

The dashed boxes in the block diagram separate the three main parts of the USART (listed from the top): Clock Generator, Transmitter and Receiver. Control Registers are shared by all units. The Clock Generation logic consists of synchronization logic for external clock input used by synchronous slave operation, and the baud rate generator. The XCKn (Transfer Clock) pin is only used by synchronous transfer mode. The Transmitter consists of a single write buffer, a serial Shift Register, Parity Generator and Control logic for handling different serial frame formats. The write buffer allows a continuous transfer of data without any delay between frames. The Receiver is the most complex part of the USART module due to its clock and data recovery units. The recovery units are used for asynchronous data reception. In addition to the recovery units, the Receiver includes a Parity Checker, Control logic, a Shift Register and a two level receive buffer (UDRn). The Receiver supports the same frame formats as the Transmitter, and can detect Frame Error, Data OverRun and Parity Errors.

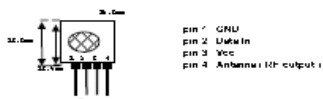
**ANEXO B**

DATASHEET TLP434A&RLP434A

**TLP434A & RLP434A RF ASK Hybrid Modules for Radio Control ( New Version )**

**TLP434A Ultra Small Transmitter**

**Easy-Link  
Wireless**



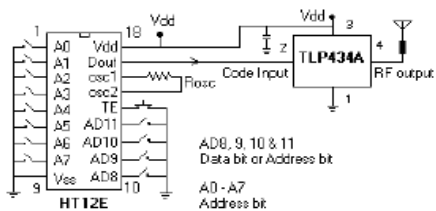
Frequency **315, 418 and 433.92 MHz**

Modulation : ASK  
Operation Voltage : 2 - 12 VDC

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Vcc	Operating supply voltage		2.0	-	12.0	V
Ic 1	Peak Current (2V)		-	-	164	mA
Ic 2	Peak Current (12V)		-	-	19.4	mA
Vh	Input High Voltage	I <sub>data</sub> = 100uA (High)	Vcc-0.5	Vcc	Vcc+0.5	V
Vl	Input Low Voltage	I <sub>data</sub> = 0uA (Low)	-	-	0.3	V
FO	Absolute Frequency	315MHz module	314.8	315	315.2	MHz
PO	RF Output Power- 50ohm	Vcc = 9V-12V	-	16	-	dBm
		Vcc = 5V-6V	-	14	-	dBm
DR	Data Rate	Bisternal Encoding	512	4.8K	200K	bps

Notes : ( Case Temperature = 25°C ± 2°C , Test Load Impedance = 50 ohm )

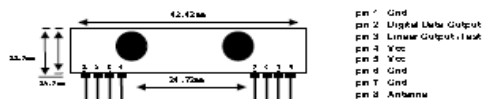
Application Circuit :  
Typical Key-chain Transmitter using HT12E 18DIP, a Binary 12 bit Encoder from  
Hokkai Semiconductor Inc.



**Laipac Technology, Inc.**  
105 West Beaver Creek Rd, Unit 207 Richmond Hill, Ontario L4B 1G5 Canada  
Tel: (905) 762-1228 Fax: (905) 763-1737 e-mail: info@laipac.com



**RLP434A SAW Based Receiver**



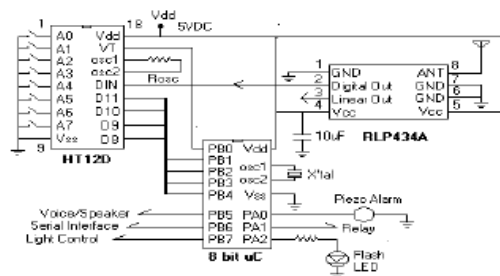
Frequency **315, 418 and 433.92 MHz**

Modulation : ASK  
Supply Voltage : 3.3 - 8.0 VDC  
Output : Digital & Linear

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Vcc	Operating supply voltage		3.3	5.0V	6.0	V
I <sub>ot</sub>	Operating Current		-	4.5	-	mA
V <sub>data</sub>	Data Out	I <sub>data</sub> = +200uA ( High ) I <sub>data</sub> = -10 uA ( Low )	Vcc-0.5	-	Vcc	V

Electrical Characteristics	SYM	Min	Typ	Max	Unit
Operation Radio Frequency	FC		315, 418 and 433.92		MHz
Sensitivity	Ref		-110		dBm
Channel Width			+500		KHz
Noise Equivalent BW			4		KHz
Receiver Turn On Time			5		ms
Operation Temperature	Top	-20	-	80	C
Baseband Data Rate			4.8		KHz

Application Circuit :  
Typical RF Receiver using HT12D-18DIP, a Binary 12 bit Decoder with 8 bit uC HT48RXX from  
Hokkai Semiconductor Inc.



**ANEXO C**

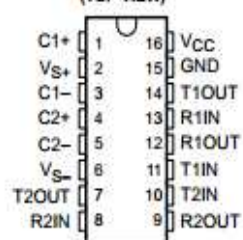
DATASHEET MAX232

## MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLL5047I—FEBRUARY 1989—REVISED OCTOBER 2002

- Meet or Exceed TIA/EIA-232-F and ITU Recommendation V.28
- Operate With Single 5-V Power Supply
- Operate Up to 120 kbit/s
- Two Drivers and Two Receivers
- $\pm 30$ -V Input Levels
- Low Supply Current . . . 8 mA Typical
- Designed to be Interchangeable With Maxim MAX232
- ESD Protection Exceeds JESD 22 – 2000-V Human-Body Model (A114-A)
- Applications
  - TIA/EIA-232-F
  - Battery-Powered Systems
  - Terminals
  - Modems
  - Computers

MAX232 . . . D, DW, N, OR NS PACKAGE  
MAX232I . . . D, DW, OR N PACKAGE  
(TOP VIEW)



### description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept  $\pm 30$ -V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

### ORDERING INFORMATION

T <sub>A</sub>	PACKAGE†	ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube	MAX232N
	SOIC (D)	Tube	MAX232D
		Tape and reel	MAX232DR
	SOIC (DW)	Tube	MAX232DW
		Tape and reel	MAX232DWR
	SOP (NS)	Tape and reel	MAX232NSR
-40°C to 85°C	PDIP (N)	Tube	MAX232IN
	SOIC (D)	Tube	MAX232ID
		Tape and reel	MAX232IDR
	SOIC (DW)	Tube	MAX232IDW
		Tape and reel	MAX232IDWR

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at [www.ti.com/sc/package](http://www.ti.com/sc/package).

## MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS0471 - FEBRUARY 1989 - REVISED OCTOBER 2002

### Function Tables

#### EACH DRIVER

INPUT TIN	OUTPUT TOUT
L	H
H	L

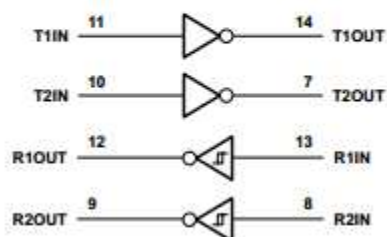
H = high level, L = low level

#### EACH RECEIVER

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

### logic diagram (positive logic)



## MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLL5047I – FEBRUARY 1989 – REVISED OCTOBER 2002

### absolute maximum ratings over operating free-air temperature range (unless otherwise noted)<sup>†</sup>

Input supply voltage range, $V_{CC}$ (see Note 1)	.....	-0.3 V to 6 V
Positive output supply voltage range, $V_{S+}$	.....	$V_{CC} - 0.3$ V to 15 V
Negative output supply voltage range, $V_{S-}$	.....	-0.3 V to -15 V
Input voltage range, $V_I$ : Driver	.....	-0.3 V to $V_{CC} + 0.3$ V
Receiver	.....	$\pm 30$ V
Output voltage range, $V_O$ : T1OUT, T2OUT	.....	$V_{S-} - 0.3$ V to $V_{S+} + 0.3$ V
R1OUT, R2OUT	.....	-0.3 V to $V_{CC} + 0.3$ V
Short-circuit duration: T1OUT, T2OUT	.....	Unlimited
Package thermal impedance, $\theta_{JA}$ (see Note 2): D package	.....	73°C/W
DW package	.....	57°C/W
N package	.....	67°C/W
NS package	.....	64°C/W
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	.....	260°C
Storage temperature range, $T_{stg}$	.....	-65°C to 150°C

<sup>†</sup> Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: All voltage values are with respect to network ground terminal.

2. The package thermal impedance is calculated in accordance with JESD 51-7.

### recommended operating conditions

		MIN	NOM	MAX	UNIT
$V_{CC}$	Supply voltage	4.5	5	5.5	V
$V_{IH}$	High-level input voltage (T1IN, T2IN)	2			V
$V_{IL}$	Low-level input voltage (T1IN, T2IN)			0.8	V
R1IN, R2IN	Receiver input voltage			$\pm 30$	V
$T_A$	Operating free-air temperature	MAX232	0	70	°C
		MAX232I	-40	85	

### electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (see Note 3 and Figure 4)

PARAMETER	TEST CONDITIONS	MIN	TYP <sup>‡</sup>	MAX	UNIT
$I_{CC}$ Supply current	$V_{CC} = 5.5$ V, All outputs open, $T_A = 25^\circ\text{C}$		8	10	mA

<sup>‡</sup> All typical values are at  $V_{CC} = 5$  V and  $T_A = 25^\circ\text{C}$ .

NOTE 3: Test conditions are C1–C4 = 1  $\mu\text{F}$  at  $V_{CC} = 5 \text{ V} \pm 0.5 \text{ V}$ .

## MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047I—FEBRUARY 1989—REVISED OCTOBER 2002

### DRIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 3)

PARAMETER		TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V <sub>OH</sub>	High-level output voltage	T1OUT, T2OUT	R <sub>L</sub> = 3 kΩ to GND	5	7		V
V <sub>OL</sub>	Low-level output voltage‡	T1OUT, T2OUT	R <sub>L</sub> = 3 kΩ to GND		-7	-5	V
r <sub>O</sub>	Output resistance	T1OUT, T2OUT	V <sub>S+</sub> = V <sub>S-</sub> = 0, V <sub>O</sub> = ±2 V	300			Ω
I <sub>OS</sub> §	Short-circuit output current	T1OUT, T2OUT	V <sub>CC</sub> = 5.5 V, V <sub>O</sub> = 0		±10		mA
I <sub>IS</sub>	Short-circuit input current	T1IN, T2IN	V <sub>I</sub> = 0			200	μA

† All typical values are at V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C.

‡ The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

§ Not more than one output should be shorted at a time.

NOTE 3: Test conditions are C1–C4 = 1 μF at V<sub>CC</sub> = 5 V ± 0.5 V.

switching characteristics, V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C (see Note 3)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
SR	Driver slew rate	R <sub>L</sub> = 3 kΩ to 7 kΩ, See Figure 2			30	V/μs
SR(t)	Driver transition region slew rate	See Figure 3		3		V/μs
	Data rate	One TOUT switching		120		kb/s

NOTE 3: Test conditions are C1–C4 = 1 μF at V<sub>CC</sub> = 5 V ± 0.5 V.

### RECEIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 3)

PARAMETER		TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V <sub>OH</sub>	High-level output voltage	R1OUT, R2OUT	I <sub>OH</sub> = -1 mA	3.5			V
V <sub>OL</sub>	Low-level output voltage‡	R1OUT, R2OUT	I <sub>OL</sub> = 3.2 mA			0.4	V
V <sub>IT+</sub>	Receiver positive-going input threshold voltage	R1IN, R2IN	V <sub>CC</sub> = 5 V, T <sub>A</sub> = 25°C		1.7	2.4	V
V <sub>IT-</sub>	Receiver negative-going input threshold voltage	R1IN, R2IN	V <sub>CC</sub> = 5 V, T <sub>A</sub> = 25°C	0.8	1.2		V
V <sub>hys</sub>	Input hysteresis voltage	R1IN, R2IN	V <sub>CC</sub> = 5 V	0.2	0.5	1	V
π	Receiver input resistance	R1IN, R2IN	V <sub>CC</sub> = 5, T <sub>A</sub> = 25°C	3	5	7	kΩ

† All typical values are at V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C.

‡ The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

NOTE 3: Test conditions are C1–C4 = 1 μF at V<sub>CC</sub> = 5 V ± 0.5 V.

switching characteristics, V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C (see Note 3 and Figure 1)

PARAMETER		TYP	UNIT
t <sub>PLH(R)</sub>	Receiver propagation delay time, low- to high-level output	500	ns
t <sub>PHL(R)</sub>	Receiver propagation delay time, high- to low-level output	500	ns

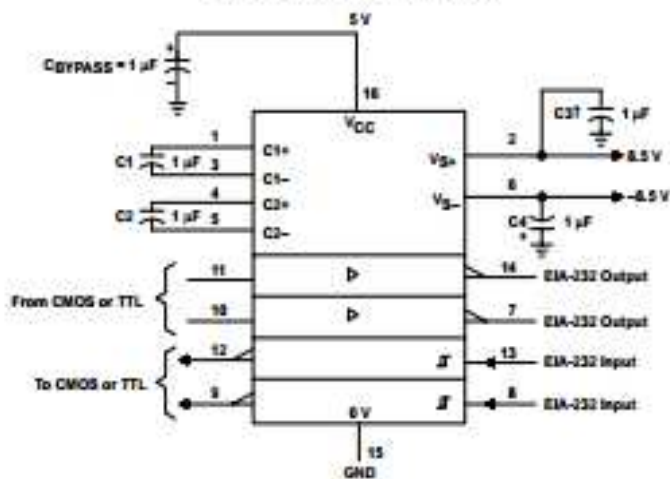
NOTE 3: Test conditions are C1–C4 = 1 μF at V<sub>CC</sub> = 5 V ± 0.5 V.



MAX232, MAX232I  
DUAL EIA-232 DRIVERS/RECEIVERS

SLS067I - FEBRUARY 1989 - REVISED OCTOBER 2002

APPLICATION INFORMATION



† C3 can be connected to V<sub>CC</sub> or GND.

Figure 4. Typical Operating Circuit

**ANEXO D**

MANUALMÓDULO GSM ZTE

**V1.10**

New Exciting Product

D1 Series Cellular DTU



# User Manual

# Chapter 2

## 2 Introduction

### 2.1 Brief

D1 Serial is a GPRS/CDMA DTU with TCP/IP Protocol embedded. It has two comparatively individual parts: IP module with TCP/IP, software interface is AT+H commands; and GPRS/CDMA module, supports all the AT Commands. All the standard AT Commands are transferred to GPRS/CDMA module via the transparent Mode of IP module.

D1 Serial is usually applicable to the Host, which has no TCP/IP but has serial interface, such as SCM Data Collection Transmission System.

### 2.2 Features

- Compact and easy to integrate into your solution;
- Multi-flexible and compact data interface, TTL, 232 and 485, TTL and 232 are reduced to Rx, Tx, GND;
- Supports more IP Protocol families;
- Data transmission via Serial/NET Mode, serial transmission mode when power on;
- Multiple status LED;
- Optimized modularization design, easy to upgrade.

### 2.3 Specification

D12S211&D12Z111&D12H111 Radio Frequency:

采用 GSM phase 2/2+标准

GSM(GSM)900MHz

DCS/GSM 1800MHz

Output power: Class 4 (2 W) at GSM900 Class 1 (1 W) at DCS 1800

D13Z311&D13Z311&D13H221 Radio Frequency:

TIA/EIA-95B, CDMA2000 1X

Band class 0: 800MHz

Band class 1: (USPCS 1900MHz)

Transmitting Frequency Range: 824.64MHz~848.37 MHz

Receiving Frequency Range: 869.94MHz~893.37 MHz.

Sensitivity:-104dB

**D12S211 Power consumption:**

Speech mode: 300mA

Sleep mode: 3.5mA

Power down : 50µA

GPRS Modem average: 360mA

**D12Z111 Power consumption:**

Speech mode: 250mA

Sleep mode: 40mA

Power down: 100mA

**D12H111 Power consumption:**

Speech mode: 250mA

Sleep mode: 3.8mA

Power down: 100mA

**D13Z311 Power consumption:**

Idle mode: 70mA

Data transfer status: 300~400mA

**D13Z311 Power consumption:**

Sleep mode: 70mA

Idle mode: 50mA

Data transfer status: 300~400mA

**D13H221 Power consumption:**

Speech mode: 250mA

Sleep mode: 40mA

Power down: 100mA

**Dimension**

Interface: RS-232/485/TTL DB9

Antenna: 50ohm SMA/Female

Input voltage: 5~25V (9V)

Operating voltage of SIM card: 3V/1.8V

Max speed rate of GPRS: 14.4KBPS

Module reset: AT commands

Voice de code standards(three kinds of rate):

Halfspeed (ETS D6.20)

Fullspeed (ETS D6.10)

Enhanced fullspeed (ETS D6.50/D6.60/D6.80)

Volume: 75\*50\*21.6mm

Weight: 200g

**Environment**

Ambient temperature: -20°C to +60°C

Storage temperature: -30°C ~+85°C.

Humidity: &lt;math&gt;\leq 90\%&lt;/math&gt;

Electromagnetic Compatible

Electrostatic Discharge (ESD): 3 classes

Radiated, radio-frequency, electromagnetic field immunity test: 3 class

## 2.4 Application

- Remote Data Monitor and Control
- Water, gas and oil flow metering
- AMR (automatic meter reading)
- Power station monitoring and control
- Remote POS (point of sale) terminal
- Traffic signals monitor and control
- Fleet management
- Power distribution network supervision
- Central heating system supervision
- Weather station data transmission
- Hydrology data acquisition
- Weighing machine
- Traffic info guidance
- Parking meter and Taxi Monitor
- Telecom equipments supervision (mobile base station, microwave or optical relay station)

## 3 Getting Started

### 3.1 Panel introduction



Note: About Hardware description ,please according to following file  
*M1\_Modem\_DTU\_Hardware\_Description\_V620R.doc*

### 3.2 The LED state

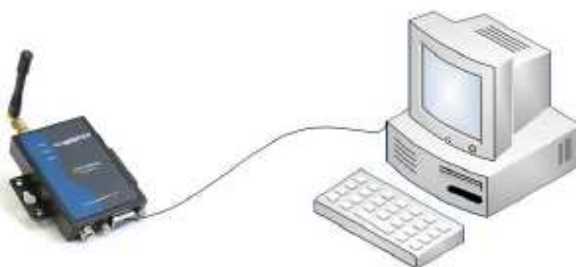
In order to check the module working state. Our product have three Led, pwr LED is power state, Ring LED is Ring state, Data LED is Data state.

	PWR	Ring	Data
Start-up	Lights up 3s, flashing 0.5s,wink 0.5s ,lights up0.5s	wink	Lights up 0.5s
Logon network	flashing	wink	flashing
Sleep state	Lights up 0.5s, wink 0.5s	wink	wink
date Transfer	Lights up 0.5s, wink 0.5s	wink	flashing

No data transfer	Lights up 0.5s, wink 0.5s, Lights up 1s	wink	wink
Voice call	Lights up 0.5s, wink 0.5s	Lights up 1s, wink 4s	wink
reboot	After 5s, wink	wink	wink

### 3.3 Connect to products

Please connect antenna and cable with our products, make sure, the port is COM1 or COM2?



### 3.4 Insert SIM Card

Open the back cover, insert into SIM card as follows:



### 3.5 Note: Hyper Terminal

Open the HyperTerminal and input \*\*\* (any) as follows:



## 5 Comom function

### 5.1 Ping Function

```

AT+IPDS1=220.192.32.103 //setting advanced destination for ping
AT+IPDS2=220.192.0.130 //setting backup destination for ping, when first destination reply
                        time out
AT+IPDS1=www.sina.com //Setting aim top-priority server. send PING package for cycle.
                       ( you can changed address with others )
AT+IPDS2=www.21cn.com //setting backup server. in case top-priority servers have pro blem
                       (you can changed address with others )

AT+IPGT=10000 //setting timeout
AT+IPFR=n //setting frequency to send ping packet
AT+IPARS //parameter save

```

Note: The function is only for Serial NET mode, detect whether online by period sending ping packet. Redial up when be detected offline. In Chinese mainland, China Unicom filter the ping packet to Internet, so the user should set the destination to China Unicom' DNS.

#### 5.1.1 Common China Unicom DNS

```

220.192.32.103
220.192.0.130

```

### 5.2 How to Change Baud Rate

D

#### 5.2.1 Change CDMA Module Baud Rate

```

AT+IMCM //switch to at command mode
AT+IPR? //query current baud rate

```

Shenzhen Forwell Wireless Co., Ltd

AT+IPR=n //setting a new baud rate

Note: n=0(2400)/1(4800)/2(9600)/3(19200)/4(38400)/5(76800)/6(115200) (the factory default value is 9600)

## 5.2.2 Change TCP/IP Module Baud Rate

AT+i //switch to AT+I command mode

AT+IBDRF=n //below AT+I command should take effect after power down and on

AT+IBDRM=n

AT+ENSI="n,S,m,1,0" //m=n,o,e (no parity, odd parity, even parity), the parameters must use low case

AT+IPARS //parameter save

## 5.2.3 The relation with parameter to baud rate

n=3	2400
n=4	4800
n=5	9600
n=6	19200
n=7	38400
n=8	76800
n=9	115200

Note: AT+IPR change the CDMA Module baud rate, AT+IBDRF, AT+IBDRM is TCP/IP Module baud rate for command mode, AT+ENSI is TCP/IP Module baud rate for SerialNET. To change baud rate, you must use the right order: firstly CDMA Module, secondary TCP/IP Module

## 5.3 How to setup APN Or VPDN

### 5.3.1 Setting APN Configuration

AT+IMIS="at+gddcont=1,ip, "" "" //Setting network(APN), fit for D1 Z5211

AT+iUSRN="" "" //user name

AT+iPWD="" "" //password

AT+IPARS //save the parameter

### 5.3.2 Setting VPDN configuration

AT+iUSRN="" "" //user name

AT+iPWD="" "" //password

**ANEXO E**

DATASHEET ULN2003A



## ULN2001A-ULN2002A ULN2003A-ULN2004A

### SEVEN DARLINGTON ARRAYS

- SEVEN DARLINGTONS PER PACKAGE
- OUTPUT CURRENT 500mA PER DRIVER (600mA PEAK)
- OUTPUT VOLTAGE 50V
- INTEGRATED SUPPRESSION DIODES FOR INDUCTIVE LOADS
- OUTPUTS CAN BE PARALLELED FOR HIGHER CURRENT
- TTL/CMOS/PMOS/DTL COMPATIBLE INPUTS
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY LAYOUT

#### DESCRIPTION

The ULN2001A, ULN2002A, ULN2003 and ULN2004A are high voltage, high current darlington arrays each containing seven open collector darlington pairs with common emitters. Each channel rated at 500mA and can withstand peak currents of 600mA. Suppression diodes are included for inductive load driving and the inputs are pinned opposite the outputs to simplify board layout.

The four versions interface to all common logic families :

ULN2001A	General Purpose, DTL, TTL, PMOS, CMOS
ULN2002A	14-25V PMOS
ULN2003A	5V TTL, CMOS
ULN2004A	6-15V CMOS, PMOS

These versatile devices are useful for driving a wide range of loads including solenoids, relays DC motors, LED displays filament lamps, thermal print-heads and high power buffers.

The ULN2001A/2002A/2003A and 2004A are supplied in 16 pin plastic DIP packages with a copper leadframe to reduce thermal resistance. They are available also in small outline package (SO-16) as ULN2001D/2002D/2003D/2004D.



DIP16

ORDERING NUMBERS: ULN2001A/2A/3A/4A



SO16

ORDERING NUMBERS: ULN2001D/2D/3D/4D

#### PIN CONNECTION

