



**Universidad del Azuay**

Facultad de Ciencia y Tecnología

Escuela de Ingeniería Electrónica

Diseño, Construcción y Configuración de un Sistema de Control  
mediante SMS aplicado a Domótica.

Trabajo de graduación previo a la obtención del título de  
Ingeniero Electrónico.

Autor

Juan Andrés Alvarez Pineda

Director

Leopoldo Carlos Vázquez Rodríguez

Cuenca, Ecuador  
2013

## DEDICATORIA

Dedico este trabajo a mis padres por brindarme su apoyo incondicional a lo largo de todos los años de estudio.

## **AGRADECIMEINTOS**

Agradezco principalmente a Dios por concederme la vida y sabiduría para poder culminar este proyecto.

A mis padres por la confianza que han puesto en mí, el apoyo moral y económico que siempre he recibido de ellos con el cual he logrado terminar esta carrera.

A mi primo Marco, por el conocimiento impartido, su generosidad y dedicarme su valioso tiempo.

A mis profesores por compartir además de sus conocimientos profesionales, valores que me han ayudado a ser cada día una persona mejor.

*03/06/13*

## RESUMEN

### Diseño, Construcción y Configuración de un Sistema de Control mediante SMS aplicado a Domótica.

Para configurar la comunicación entre el módulo GSM y microcontrolador PIC, y para diseñar la central de mando que hará el control del sistema de iluminación, alarma y climatización, se utiliza la programación basada en Basic conjuntamente con el software ProtonIDE el cual sirve como programador y compilador.

La comunicación entre los dos dispositivos se hace de forma serial asíncrona, la comunicación permite el envío y recepción de caracteres ASCII, estos son leídos y filtrados por el microcontrolador PIC para hacer actuar a los diferentes sistemas.

Cada sistema podrá ser controlado mediante comandos representados por mensajes de texto (SMS) enviados desde cualquier teléfono móvil a un número celular específico. El envío de mensajes de texto se puede realizar desde cualquier teléfono móvil de forma manual o utilizando una aplicación para dispositivos que soporten el sistema operativo android y red GSM.

Palabras clave: SIM9000, DS18B20, EUSART, 18F4550, comandos AT, 16F628A.

DIRECTOR DE ESCUELA  
INGENIERÍA ELECTRÓNICA

Ing. Francisco Vásquez

DIRECTOR DEL TRABAJO DE  
GRADO

Ing. Leopoldo Vásquez

AUTOR

Juan Andrés Alvarez Pineda

## ABSTRACT

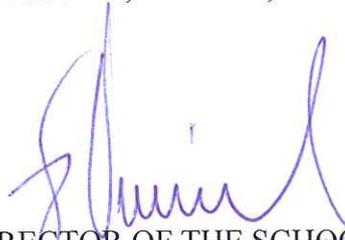
### Design, Construction, and Configuration of a Control System through SMS applied to Domotica

In order to configure the communication system between the GSM module and the PIC microcontroller, and to design the central command that will control the illumination, alarm, and air-conditioning systems, we employed programming based on Basic and ProtonIDE software, which serves as a programmer and a compiler.

The communication between the two devices is carried out in an asynchronous serial manner, it allows sending and receiving ASCII characters, which are then read and filtered by the PIC microcontroller in order to make the different systems operate.

Each system can be controlled through commands represented by text messages (SMS) that can be sent from any mobile phone to a specific cellular phone number. It is possible to send the text message from any mobile phone manually or by using an application for devices that have android operative system and GSM network.

**Key Words:** SIM9000, DS18B20, EUSART, 18F4550, AT commands, 16F628A



DIRECTOR OF THE SCHOOL  
OF ELECTRONIC ENGINEERING  
Ing. Francisco Vasquez



DIRECTOR OF THE GRADUATION  
WORK

Ing. Leopoldo Vasquez

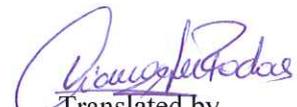


AUTHOR

Juan Andres Alvarez Pineda



UNIVERSIDAD DEL  
AZUAY  
DPTO. IDIOMAS



Translated by,  
Diana Lee Rodas

## INDICE DE CONTENIDOS

Dedicatoria.....	ii
Agradecimiento.....	iii
Resumen.....	iv
Abstract.....	v
Índice de Contenidos.....	vi
Índice de Tablas.....	xi
Índice de Ilustraciones.....	xii

### **CAPÍTULO 1: MÓDULO GSM/GPRS SIM900**

1.1. Definición.....	2
1.2. Características generales.....	2
1.3. Fuente de Alimentación.....	4
1.4. Encender/Apagar el módulo SIM900.....	5
1.4.1. Pin PWRKEY.....	5
1.4.2. Apagado del módulo SIM900 usando comando AT.....	6
1.5. Pin STATUS.....	6
1.6. Interfaz de antena.....	7
1.7. Interfaz serial.....	7
1.7.1. Definición de pines de la interfaz serial (conector DB9).....	7
1.7.2. Puerto serial.....	8
1.8. Comandos AT.....	9
1.8.1. Sintaxis de los comandos AT.....	10
1.9. Indicador del estado de la red.....	12
1.10. Pruebas de comunicación con el modem SIM900.....	13
1.10.1. Sincronización entre dispositivos.....	13
1.10.2. Configuración del puerto serial.....	14
1.10.3. Lectura de mensajes de texto.....	16
1.10.4. Borrar la memoria de mensajes.....	16

### **CAPÍTULO 2: MICROCONTROLADOR PIC**

2.1. Introducción.....	18
2.1.1. Tipos de arquitecturas de los microcontroladores PIC.....	18
2.1.2. Arquitectura Von Neumann.....	18
2.1.3. Arquitectura Harvard.....	19
2.2. Elementos de los microcontroladores.....	20

2.2.1.	El procesador.....	20
2.2.2.	Memoria.....	21
2.2.3.	Puertas de entrada y salida.....	24
2.2.4.	Reloj principal.....	24
2.3.	Recursos principales.....	24
2.3.1.	Temporizadores o timers.....	25
2.3.2.	Perro guardián o Watchdog.....	25
2.3.3.	Protección ante fallo de alimentación o Brownout.....	26
2.3.4.	Estado de reposo o de bajo consumo.....	26
2.3.5.	Convertor A/D (CAD).....	26
2.3.6.	Convertor D/A (CDA).....	26
2.3.7.	Comparador analógico.....	27
2.3.8.	Modulador de anchura de impulsos o PWM.....	27
2.3.9.	Puertas digitales de entrada y salida E/S.....	27
2.3.10.	Puertas de comunicación.....	27
2.4.	Familia PIC18F y PIC 16F.....	28
2.4.1.	Características fundamentales PIC18F.....	28
2.4.2.	Características fundamentales PIC16F.....	30
2.5.	Proceso de desarrollo.....	30
2.5.1.	Desarrollo del programa.....	31
2.5.2.	Programación del microcontrolador.....	36
2.5.3.	Prueba y verificación.....	38

### **CAPÍTULO 3. COMUNICACIÓN ENTRE MICROCONTROLADOR PIC Y MÓDULO GSM**

3.1.	Protocolo RS-232.....	39
3.1.1.	Comunicación serial síncrona.....	40
3.1.2.	Comunicación serial asíncrona.....	40
3.1.3.	La norma RS-232.....	41
3.1.4.	El circuito MAX-232.....	42
3.2.	Encendido del módulo SIM900.....	45
3.3.	Comunicación entre modem SIM900 y PIC 18F4550.....	47
3.3.1.	Sincronización entre microcontrolador y módulo SIM900....	53
3.3.2.	Borrado de SMS del módulo SIM900.....	54
3.3.3.	Lectura de SMS del módulo SIM900.....	55
3.3.4.	Filtrado de SMS.....	55
3.4.	Comunicación entre PIC 16F628A y PIC18F4550.....	57

3.4.1. Comunicación serial entre PIC16F628A y PIC18F4550.....	58
3.4.2. Comunicación para el sistema de iluminación.....	60
3.4.3. Comunicación para el sistema de alarma.....	63
3.4.4. Comunicación para el sistema de climatización.....	66
3.4.5. Comunicación para reportes de Iluminación, alarma o climatización.....	68

## **CAPÍTULO 4. PERIFÉRICOS DE ENTRADA Y SALIDA**

4.1. Componentes electrónicos activos.....	71
4.1.1. Baterías.....	72
4.1.2. Diodo.....	73
4.1.3. Transistor.....	76
4.2. Componentes electrónicos pasivos.....	88
4.2.1. Resistor.....	88
4.2.2. Condensador.....	93
4.2.3. Diodo LED.....	98
4.2.4. Relés.....	99
4.2.5. Conectores.....	102
4.2.6. Interruptores.....	103
4.3. Teclado matricial.....	105
4.4. LCD matricial 4x20.....	106
4.5. Lámparas LED.....	108
4.6. Detector de movimiento.....	109
4.6.1. Sensor infrarrojo pasivo (PIR).....	109
4.6.2. Limitaciones en el empleo de detectores de movimiento PIR.....	110
4.7. Detector de apertura.....	110
4.7.1. Reed switch.....	111
4.7.2. Aplicaciones.....	112
4.8. Indicadores de alarma.....	112
4.8.1. Sirenas electrónicas.....	112
4.8.2. Luces estroboscópicas.....	113
4.9. Conexión y configuración del teclado matricial.....	114
4.10. Conexión y configuración del LCD matricial 4x20.....	116
4.11. Control del sistema de alarma.....	119
4.11.1. Conexión de sensores al sistema de alarma .....	120
4.11.2. Conexión de sirena al sistema de alarma.....	121

4.11.3. Activación modo presente.....	121
4.11.4. Activación modo ausente.....	123
4.11.5. Desactivación de alarma.....	126
4.11.6. Envío de reporte de alarma.....	127
4.11.7. Envío de reporte de estado de alarma.....	129
4.12. Control del sistema de iluminación.....	130
4.12.1. PWM – Frecuencia y Duty Cycle.....	130
4.12.2. Control de iluminación PWM con ProtonIDE.....	132
4.12.3. Conexión de lámparas LED al sistema de iluminación.....	135
4.12.4. Envío de reporte del estado de iluminación.....	135
4.13. Control del sistema de climatización.....	136
4.13.1. Sensor de temperatura DS18B20.....	136
4.13.2. Lectura de temperatura con PIC 16F628A.....	138
4.13.3. Control de temperatura.....	140
4.14. Fuente de poder.....	142
4.14.1. Transformador.....	142
4.14.2. Rectificador.....	142
4.14.3. Filtro.....	142
4.14.4. Regulador de Tensión.....	143

## **CAPÍTULO 5. DISEÑO Y CONSTRUCCIÓN DE TARJETAS DE CIRCUITO IMPRESO**

5.1. Circuito esquemático.....	144
5.1.1. Altium Designer.....	144
5.1.2. Circuito esquemático en Altium Designer.....	145
5.1.3. Dibujando el circuito esquemático.....	147
5.1.4. Cableado del circuito esquemático.....	150
5.1.5. Net y Net Labels (Etiquetas de conexión).....	151
5.2. Tarjetas de circuito impreso.....	154
5.2.1. Ruteo de pistas para PCB.....	154
5.2.2. Impresión de pistas para PCB.....	158
5.2.3. Preparación de la Placa (Baquelita o Fibra de Vidrio).....	159
5.2.4. Transferencia térmica del papel hacia la lámina de cobre...	159
5.2.5. Proceso de atacado (reducción) del cobre.....	160
5.2.6. Limpieza de la placa.....	162
5.2.7. Perforación de la placa.....	163
5.2.8. Soldadura de elementos.....	163

5.3.	Chasis o caja para el proyecto.....	165
5.3.1.	Instalación de elementos.....	166
<b>CAPÍTULO 6. APLICACIÓN PARA TELÉFONO MÓVIL</b>		
6.1.	Google App Inventor.....	168
6.1.1.	Características.....	168
6.1.2.	Configuración MIT App Inventor.....	170
6.2.	Aplicación móvil para sistema de control de domótica.....	170
	<b>CONCLUSIONES.....</b>	<b>177</b>
	<b>RECOMENDACIONES.....</b>	<b>178</b>
	<b>BIBLIOGRAFÍA.....</b>	<b>179</b>

## ÍNDICE DE TABLAS

Tabla 1.1: Asignación de Pines del Conector DB9.....	7
Tabla 1.2: Estado del pin NETLIGHT.....	12
Tabla 1.3: Configuración del puerto Serial (Paridad, bits de parada, datos).....	15
Tabla 2.1: Características de los PIC de la familia 18F.....	29
Tabla 2.2: Características de los PIC de la familia 16F.....	30
Tabla 3.1: Niveles de Voltaje para el CI MAX232.....	45
Tabla 3.2: Formula de Baudios.....	48
Tabla 3.3: Registros asociados en la comunicación serial.....	48
Tabla 3.4: Registros asociados con la transmisión y recepción serial.....	52
Tabla 3.5: Comandos para el sistema de Iluminación.....	56
Tabla 3.6: Comandos para el sistema de Alarma.....	56
Tabla 3.7: Comandos para el sistema de Climatización.....	57
Tabla 4.1: Componentes electrónicos activos.....	71
Tabla 4.2: Código de colores.....	90
Tabla 4.3: Descripción de señales empleadas por el modulo .....	107
Tabla 4.4: Pines del sensor DS18B20.....	137
Tabla 4.5: Relación de temperatura.....	138

## ÍNDICE DE FIGURAS

Figura 1.1 SIM900 de SIMCom.....	3
Figura 1.2. Asignación de Pines del Módulo SIM900.....	3
Figura 1.3. Diagrama en Bloques del Módulo.....	4
Figura 1.4. Fuente de Alimentación externa.....	4
Figura 1.5: Pulso para encender/apagar el módulo SIM900.....	5
Figura 1.6: Pulso para encender/apagar el módulo SIM900 mediante un pulsante	6
Figura 1.7: Conexión entre Modulo y Cliente.....	8
Figura 1.8. Respuesta del Modem SIM900 al comando AT “AT”.....	13
Figura 1.9. Identificación del Fabricante.....	14
Figura 1.10. Configuración de Registros ICF y IPR.....	15
Figura 1.11. Lectura de un mensaje de texto.....	16
Figura 1.12. Borrado de memoria de mensajes.....	17
Figura 2.1. Arquitectura Von Neumann.....	19
Figura 2.2. Arquitectura Harvard.....	20
Figura 2.3. Etapas del desarrollo del software.....	32
Figura 2.4. Programa Proton IDE.....	33
Figura 2.5. Programación del Microcontrolador.....	36
Figura 2.6. Software Programador PICKit2 v2.61.....	37
Figura 2.7. Hardware programador PICKit2.....	37
Figura 2.8. Microcontrolador PIC Programado.....	39
Figura 3.1. Transmisión Serial Síncrona.....	40
Figura 3.2. Transmisión de un byte en forma asíncrona.....	40
Figura 3.3. Conector RS-232 tipo hembra.....	42
Figura 3.4. Circuito Integrado MAX232.....	43
Figura 3.5. Diagrama de Conexiones del CI MAX 232.....	44
Figura 3.6. Encendido del Modem SIM900.....	46
Figura 3.7. Comunicación serial entre PIC18F4550 y Modem SIM900.....	47
Figura 4.1. Batería recargable de 12V 4Ah.....	72
Figura 4.2 Diodo 1N4004.....	73
Figura 4.3. Símbolo del Diodo.....	74
Figura 4.4. Polarización directa de un diodo.....	74
Figura 4.5. Polarización inversa de un diodo.....	75
Figura 4.6. Símbolos del transistor.....	77
Figura 4.7. Corrientes en el transistor.....	78
Figura 4.8. Voltajes en el transistor.....	78

Figura 4.9. Estructura de un transistor bipolar.....	79
Figura 4.10. Estructura real de un transistor.....	79
Figura 4.11. Dimensiones de un transistor.....	80
Figura 4.12. Terminales del transistor.....	80
Figura 4.13. Ejemplo de funcionamiento del transistor.....	81
Figura 4.14. Efecto Transistor. ....	82
Figura 4.15. Polarización.....	82
Figura 4.16. Corrientes en un transistor.....	83
Figura 4.17. Polarización Zona Activa .....	87
Figura 4.18. Polarización Zona de Saturación. ....	87
Figura 4.19. Símbolo de un resistor. ....	88
Figura 4.20. Ejemplo para código de colores. ....	91
Figura 4.21. Resistor Variable.....	92
Figura 4.22. Condensador.....	93
Figura 4.23. Tipos de Condensadores.....	94
Figura 4.24. Funcionamiento del Condensador.....	95
Figura 4.25. Resistencia limitadora conectada al condensador.....	95
Figura 4.26 Carga y descarga del condensador.....	96
Figura 4.27 Símbolo del diodo led.....	98
Figura 4.28 Bobina.....	99
Figura 4.29 Relé tipo interruptor.....	100
Figura 4.30. Simbología de un relé.....	100
Figura 4.31 Relé tipo conmutador. ....	101
Figura 4.32. Símbolo relé tipo conmutador. ....	101
Figura 4.33 Símbolo de un relé de 2 polos. ....	101
Figura 4.34 Símbolo de un relé de 4 polos. ....	102
Figura 4.35 Simbología de un interruptor. ....	103
Figura 4.36 Simbología de un pulsador. ....	103
Figura 4.37 Simbología de un interruptor de doble polo. ....	104
Figura 4.38 Simbología de un interruptor de doble vía. ....	104
Figura 4.39 Simbología de un interruptor de doble polo y doble vía.....	105
Figura 4.40 Pulsadores del Teclado Matricial. ....	105
Figura 4.41 Conexión de Teclado Matricial a un Microcontrolador. ....	106
Figura 4.42 Diagrama de conexiones del LCD.....	107
Figura 4.43 Lámparas LED. ....	109
Figura 4.44 Sensor Infrarrojo pasivo (PIR) .....	110
Figura 4.45 Detector de Apertura con reed switch. ....	111

Figura 4.46. Sirena Electrónica.....	113
Figura 4.47. Luz Estroboscópica. ....	113
Figura 4.48. Conexión parcial del teclado matricial al PIC18F4550.....	114
Figura 4.49. Conexión del LCD al uC 18F4550.....	117
Figura 4.50. Conexión del Contacto del Sensor al PIC16F628A.....	120
Figura 4.51. Conexión de sirena al sistema de alarma. ....	121
Figura 4.52 Conexión de lámparas led al PIC 16F628A.....	135
Figura 4.53 Sensor de Temperatura DS18B20. ....	136
Figura 4.54 Pines del sensor DS18B20.....	137
Figura 4.55 Conexión de módulo DS18B20 al microcontrolador.....	140
Figura 4.56. Interfaz de potencia para el control de un relé.....	140
Figura 4.57. Diagrama de Fuente Básica.....	143
Figura 4.58. Fuente de Poder 12V 3A y transformador de 110VCA a 16,8VCA. ....	143
Figura 5.1 Nuevo circuito esquemático. ....	145
Figura 5.2. Hoja para dibujar el circuito esquemático. ....	146
Figura 5.3. Tamaño de la hoja de archivo esquemático.....	146
Figura 5.4. Circuito electrónico (protoboard).....	147
Figura 5.5. Librería en Altium Designer.....	148
Figura 5.6. Instalación de nuevas librerías.....	148
5.7. Opción Libraries Search.....	149
Figura 5.8 Propiedades principales de un elemento.....	149
Figura 5.9. Activación de la opción Wire.....	150
Figura 5.10. Activación de la opción <i>Bus</i> .....	151
Figura 5.11. Cableado para lamparas led.....	151
Figura 5.12 Propiedades de un Net Label.....	152
Figura 5.13. Circuito esquemático del Sistema de Control Mediante SMS.....	153
Figura 5.14. Componentes para el circuito PCB.....	154
Figura 5.15. Herramienta Track.....	155
Figura 5.16. Propiedades de la Pista.....	155
Figura 5.17. Ruteo de pistas a cada componente.....	156
Figura 5.18. Herramienta Circuir Board.....	156
Figura 5.19. Herramienta Cooper Label.....	157
Figura 5.20. Herramienta Cooper Area.....	157
Figura 5.21. Circuito PCB para el Sistema de Control mediante SMS.....	158
Figura 5.22. Placa y hoja para circuito impreso.....	159
Figura 5.23. Transferencia térmica.....	160
Figura 5.24 Reducción del Cobre.....	162

Figura 5.25. Limpieza de la Placa.....	162
Figura 5.26. Perforación de la Placa.....	163
Figura 5.27. Materiales, Pasta para soldar, estaño, caufín eléctrico.....	164
Figura 5.28. Soldadura de elementos.....	164
Figura 5.29. Cajas plásticas para proyectos electrónicos.....	165
Figura 5.30. Caja Metálica de Central de Alarma.....	165
Figura 5.31. Display LCD y Teclado Matricial a ser instalados.....	166
Figura 5.32. Parte exterior de la caja metálica.....	166
Figura 5.33. Conexiones del Sistema de Control mediante SMS.....	167
Figura 6.1. Diagrama en bloques para programación en App Inventor.....	169
Figura 6.2. Pantalla para un Nuevo Proyecto.....	170
Figura 6.3 Pagina Web denominada Diseñador.....	171
Figura 6.4 Abrir el Editor de Bloques.....	171
Figura 6.5. Editor de Bloques de App Inventor.....	172
Figura 6.6. Pantallas de la Aplicación Móvil. ....	173
Figura 6.7. Programación de la Pantalla de INICIO (Clave).....	173
Figura 6.8. Programación para abrir diferentes pantallas.....	174
Figura 6.9. Envío de Mensajes de texto.....	175
Figura 6.10. Paquete para el teléfono móvil.....	176

Alvarez Pineda Juan Andrés  
Trabajo de graduación  
Ing. Leopoldo Vázquez  
Junio 2013

## **DISEÑO, CONSTRUCCIÓN Y CONFIGURACION DE UNA SISTEMA DE CONTROL MEDIANTE SMS APLICADO A DOMÓTICA**

### **INTRODUCCIÓN**

Debido al incremento de inseguridad en nuestro medio es necesario un sistema de control y monitoreo de los eventos realizados en un domicilio o local comercial mediante un sistema de alarma. Además de la inseguridad existe un desperdicio de energía eléctrica por lo que es necesario un sistema que controle la iluminación y climatización de una vivienda de forma local o remota.

Es necesario construir un sistema de control y monitoreo para alarma, iluminación y climatización para un domicilio o local comercial, que permita reducir el alto consumo de energía eléctrica y estar siempre informado de los eventos ocurridos en el inmueble mediante envío y recepción de mensajes de texto.

Estos sistemas serán controlados de forma local mediante un teclado y de forma remota con mensajes cortos de texto (SMS). La comunicación entre un microcontrolador y un modem GSM hará posible la recepción de comandos, los cuales harán actuar a cada sistema, así como el envío de mensajes de texto con la información de los eventos ocurridos.

La metodología utilizada para el desarrollo del proyecto se basa en una investigación experimental, porque considera aspectos como: planteamiento del problema, manipulación de variables independientes, medida de variables dependientes, procedimientos, análisis de datos, anticipación de resultados posibles.

## CAPÍTULO 1

### MÓDULO GSM/GPRS SIM900

#### 1.1. Definición

El SIM900 es un modem GSM / GPRS ultra compacto del tipo “Cuatri Banda” en montaje superficial y diseñado con un “Core” AMR926EJ-S, un procesador “single - chip” muy poderoso que permite mayores prestaciones que el promedio de los módulos existentes en el mercado, con mayor velocidad de procesamiento y un significativo ahorro de energía en el modo “sleep” (figura1.1 – figura 1.3).

Con una interfaz estándar, el SIM900 entrega una alta prestación en los modos GSM / GPRS en las bandas de 850/900/1800/1900 MHZ para voz, SMS, Datos, y Fax. El módulo GSM está diseñado con una técnica de ahorro de energía que en modo “*sleep*” ocupa una corriente baja (1,5mA).

En el presente capítulo, todo lo que trata a los pines del módulo SIM900, referirse a la figura 1.2.

#### 1.2. Características Generales

- Dimensiones: 24mm x 24mm x 3mm
- Cuatri Banda 850/ 900/ 1800/ 1900 MHz
- Peso: 3.4gramos
- Control vía comandos AT (GSM 07.07 ,07.05 y comandos AT SIMCOM)
- Rango de Alimentación: 3.1... 4.8VDC.
- Bajo consumo de Energía: 1.5mA (sleep mode)
- Temperatura de operación: -40°C a +85 °C
- Interfaz a SIM externa de 3V/1.8V
- Interfaz de Audio Analógico.
- RTC backup
- Interfaz SPI (opcional)
- Interfaz Serial
- Pad de Antena
- Interfaz Celular de comandos AT



Figura 1.1 SIM900 de SIMCom  
Fuente: SIMCOM. SIM900\_Hardware\_Design\_V2.00. 2010. Pagina 54.

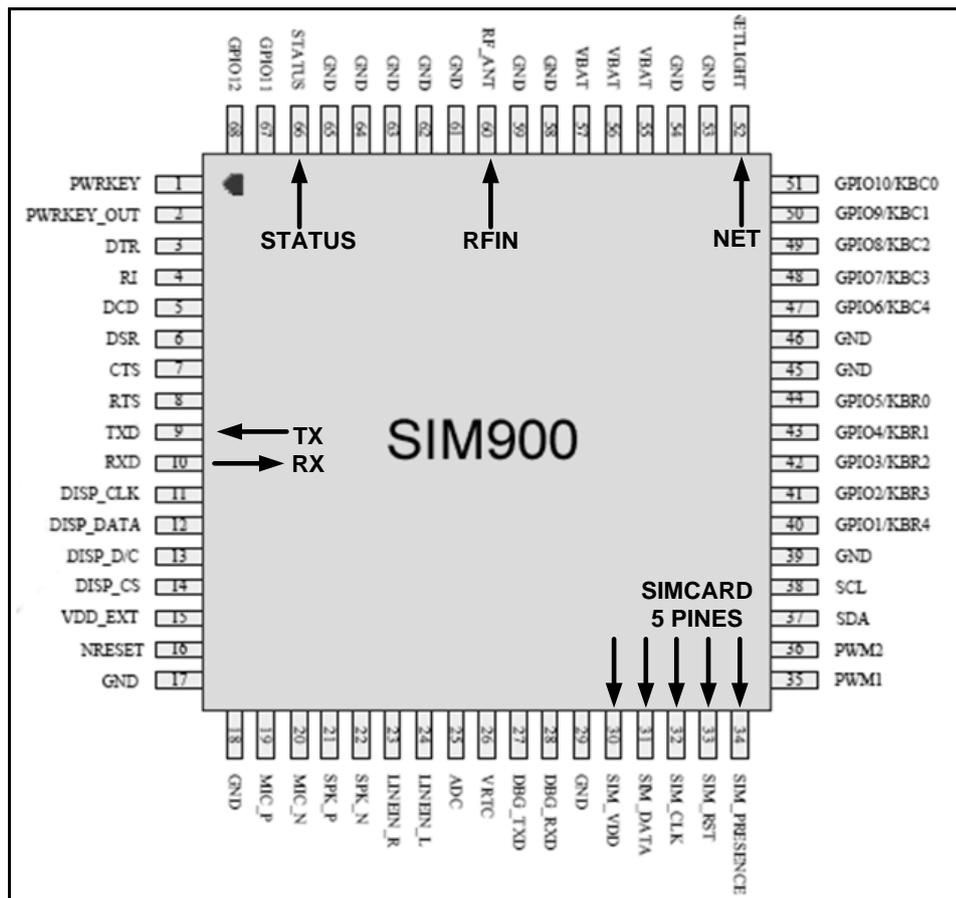


Figura 1.2. Asignación de Pines del Módulo SIM900  
Fuente: SIMCOM. SIM900\_Hardware\_Design\_V2.00. 2010. Pagina 55.

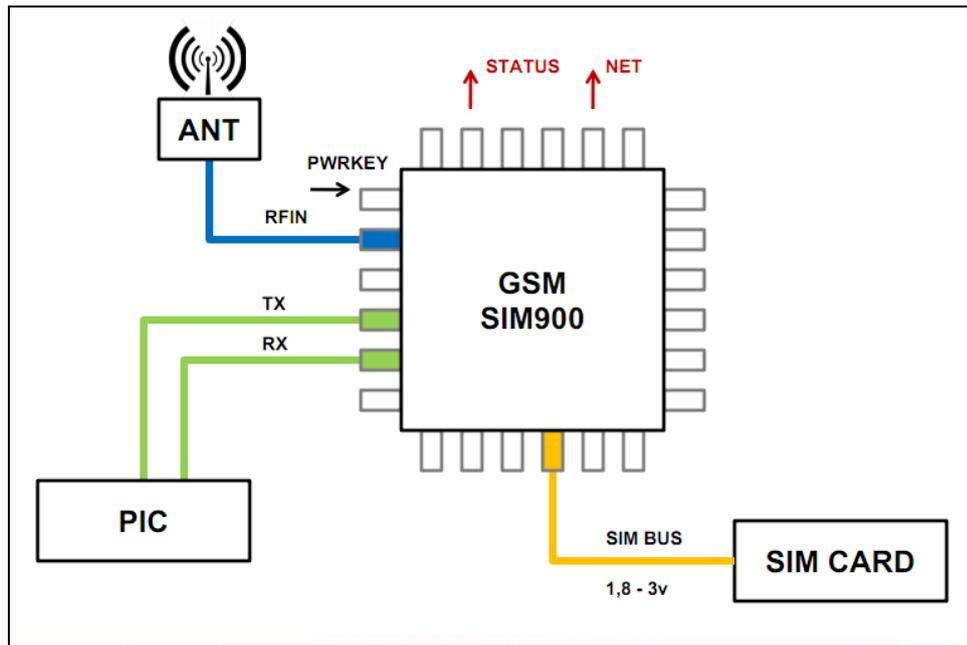


Figura 1.3. Diagrama en Bloques del Módulo  
 Fuente: McElectronics.Sim900. [http://www.mcelectronics.com.ar/trackme/clase\\_tracking\\_esp.pdf](http://www.mcelectronics.com.ar/trackme/clase_tracking_esp.pdf).  
 [Consulta: 2 de Agosto de 2012]

### 1.3. Fuente de alimentación:

La fuente de alimentación (figura 1.4) está construida con un regulador de voltaje de baja caída lineal, que permite conectar una gran variedad de fuentes de alimentación no regulada (4,2V – 14V).

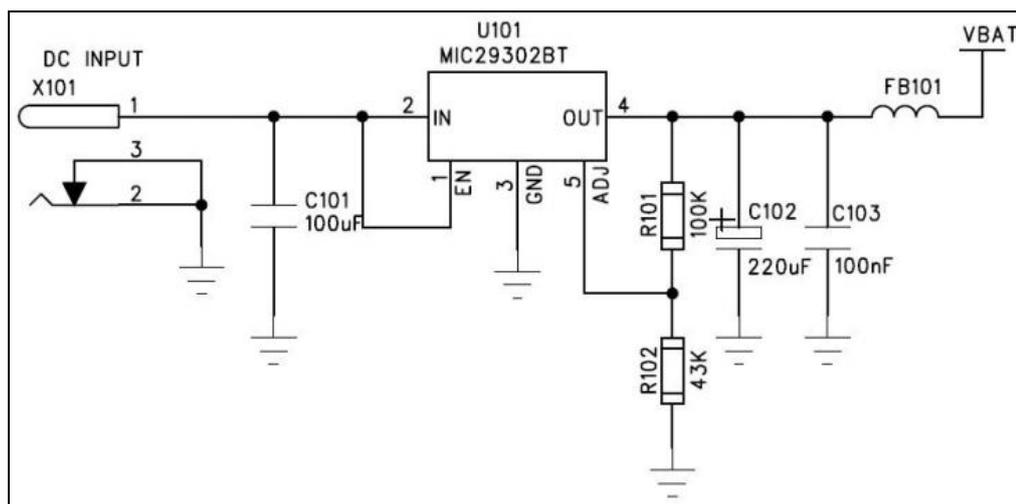


Figura 1.4. Fuente de Alimentación externa.  
 Fuente: SIMCOM. SIM900\_Hardware\_Design\_V2.00. 2010. Página 22.

La fuente de alimentación para el módulo SIM900 es una sola de VBAT = 3,4V - 4,5V, en algunos casos en la transmisión de datos la corriente aumenta a picos típicos de 2A, así que la fuente de alimentación debe ser capaz de proporcionar suficiente corriente hasta de 2A.

#### 1.4. Encender/Apagar el Modulo SIM900

##### 1.4.1 Pin PWRKEY

Se puede encender/apagar el módulo SIM900 conectando el pin PWRKEY a un voltaje de bajo nivel con una resistencia limitadora de corriente (recomendado 1K) en serie, por un tiempo de 2 segundos y luego desconectarlo. Se puede utilizar un circuito de interfaz de transistor (figura 1.5) o simplemente con un pulsante (figura 1.6).

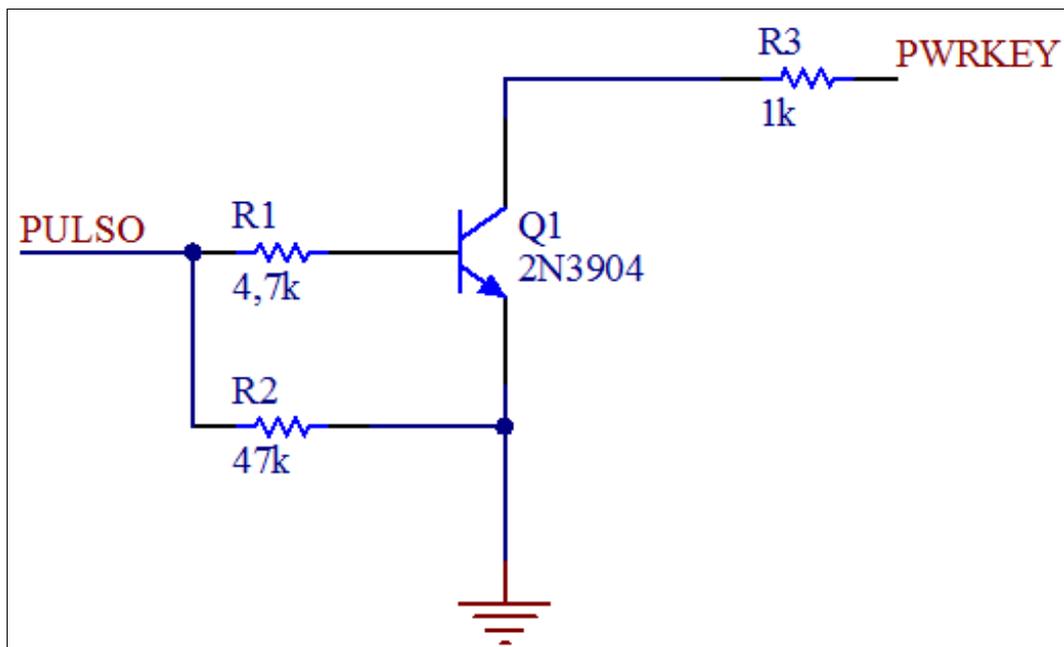


Figura 1.5. Pulso para encender/apagar el módulo SIM900

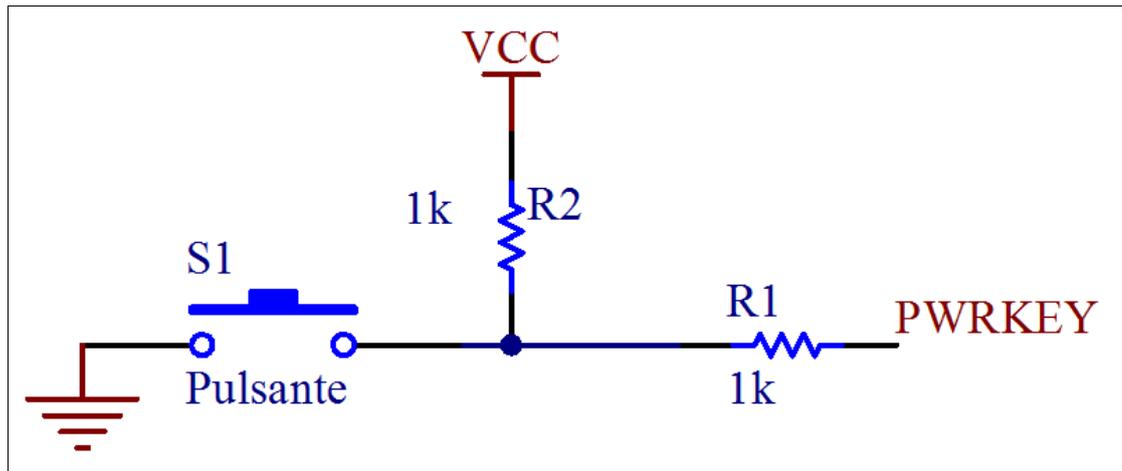


Figura 1.6. Pulso para encender/apagar el módulo SIM900 mediante un pulsante

#### 1.4.2 Apagado del módulo SIM900 usando comando AT

Los comandos AT son el lenguaje de comunicación entre el usuario y el módulo SIM900.

Puede utilizar el comando "AT+CPOWD=1" para desactivar el módulo. Este comando permite que el modulo cierre la sesión con la red y entre en un estado seguro de apagado, guardando los datos antes de ser desconectado completamente de la alimentación.

Antes de la finalización del cambio de estado, el módulo enviará el código del resultado: "*NORMAL POWER DOWN*".

Después de este momento, los comandos AT no podrán ser ejecutados. El modulo entra en modo apagado.

#### 1.5. Pin STATUS

El pin *STATUS* nos indica el estado de trabajo del módulo, el pin permanece en un nivel de voltaje bajo (0V) cuando está apagado y en nivel alto (3,8V) cuando el modulo está encendido.

## 1.6. Interfaz de antena

El modem SIM900 proporciona una interfaz de antena de RF (Radio Frecuencia), la antena debe estar colocada en el tablero principal del módulo, el pin correspondiente en el Modulo GSM es el número 60.

## 1.7. Interfaz serial

El módulo está diseñado con un circuito RS232 convertidor de nivel, que permite conectarse directamente a un puerto serial, la velocidad de transmisión se puede configurar desde los 1200 hasta 115200 baudios.

### 1.7.1: Definición de pines de la interfaz serial (conector DB9)

Nombre	Pin	Función
DTR	3	Terminal de Datos Listo
RI	4	Ring Indicador
DCD	5	Datos de detección de Portadora
DSR	6	Datos Listos
CTS	7	Borrar para enviar
RTS	8	Solicitud de Envío
TXD	9	Transmisión de Datos
RXD	10	Recepción de Datos

Tabla 1.1. Asignación de Pines del Conector DB9

El módulo GSM está diseñado con un DCE (Equipo de Comunicación de Datos), siguiendo con el tradicional DTE (Equipo Terminal de Datos). El módulo y el cliente (DTE) están conectados a través de la siguiente señal (figura 1.7).

El Puerto serie TXD: envía datos a la línea RXD del DTE.

El Puerto RXD: recibe datos de la línea TXD del DTE.

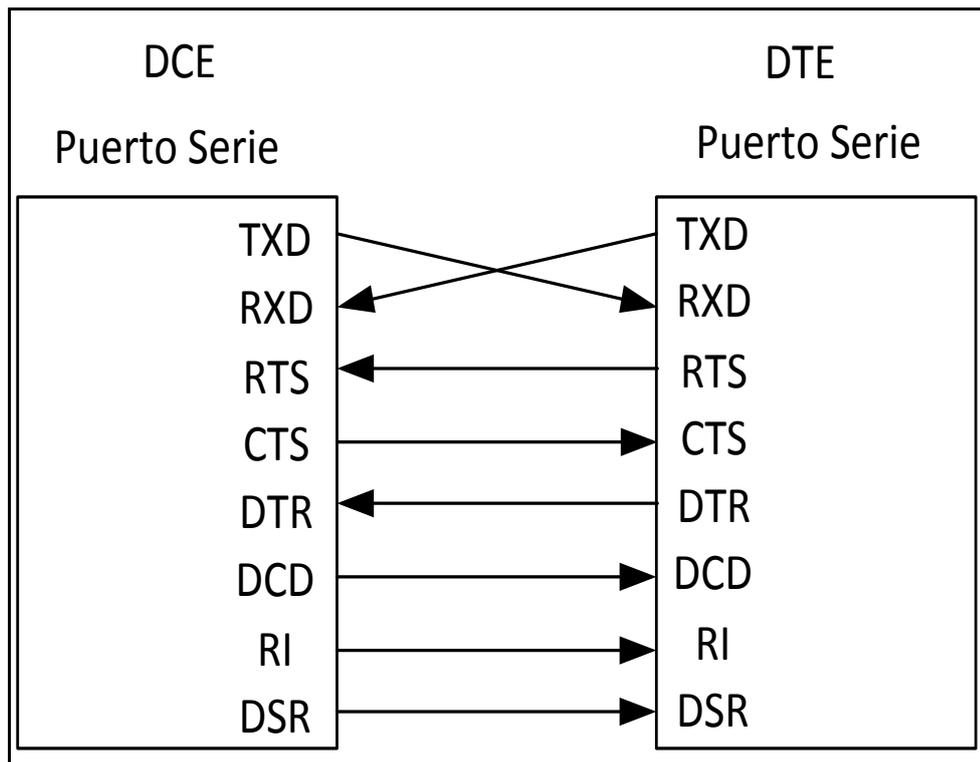


Figura 1.7. Conexión entre Modulo y Cliente

### 1.7.2. Puerto serial

Contiene líneas de datos TXD y RXD, líneas de estado RTS y CTS, líneas de control DTR, DCD, DSR y RI.

El puerto serial puede ocuparse para CDS FAX, servicio GPRS, función de multiplexado, enviar comandos AT.

El puerto serial soporta velocidades de comunicación tales como: 1200, 2400, 4800, 9600, 38400, 57600, 115200 baudios.

Para la comunicación el modulo solo necesita 3 hilos de conexión (TX, RX, GND).

Usando este modem se podrá enviar y recibir mensajes de texto (SMS), utilizando comandos AT. Los comandos AT para configurar el puerto serial son: "AT+IPR" y "AT+IFC".

## 1.8. Comandos AT

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

Los comandos AT aparecen en el año de 1977, desarrollados por Dennis Hayes como una interfaz de comunicación con un modem para así poder configurarlo y proporcionarle instrucciones, como marcar un número de teléfono. Con el avance del budio, fueron las compañías Microcomm 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 de 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 SMS, además de muchas otras opciones de configuración del terminal.

Existen los modem GSM propiamente, estos equipos ya no tienen la parte visual de manejo, es decir, no tienen pantalla ni teclado como los teléfonos celulares, sin embargo tienen la posibilidad de gestionar su base de datos, contactos telefónicos, enviar SMS, realizar llamadas, realizar configuraciones. Estos equipos para ser configurados necesitan obligatoriamente de comandos AT.

Es claro que la implementación de 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 serial, canal de infrarrojos, Bluetooth, etc.

La configuración entre módems GSM, para pruebas o conexión, se realiza mediante un ordenador o microcontrolador por lo que no dispone de teclado y pantalla como se indicó anteriormente, para dicho fin los módems GSM pueden tener puertos RS232 o USB.

### 1.8.1. Sintaxis de los comandos AT

Los comandos AT están compuestos por cadenas de caracteres ASCII que para su ejecución se debe anteponer la palabra "AT" a excepción de los comandos de pause y de repetición de comando anterior, en los que no se requiere.

Para enviar comandos AT (configuración) a un modem GSM se debe seguir la siguiente estructura:

AT+	CMGF=1	<CR>
PREFIJO	COMANDO	SUFIJO

El prefijo de los comandos AT es la cadena de caracteres AT seguido del signo +.

El sufijo de los comandos AT es <CR> carrieje return (retorno de carro) que es equivalente a un ENTER y su valor ASCII es 0Dh.

El comando es la instrucción que se dará al módulo SIM900, cuando se coloca el signo igual (=) al comando, se está configurando un parámetro, cuando se coloca el signo de interrogación (?) se está pidiendo información, cuando se combina el signo igual-interrogación se obtiene el rango de opciones posibles que se puede configurar en el módulo.

La respuesta del modem ante un comando tiene la siguiente estructura:

<CR><LF>	OK	<CR><LF>
PREFIJO	CODIGO	SUFIJO

CODIGO es la respuesta del módulo, en este caso OK corresponde a una operación exitosa, en caso de una operación fallida se retorna ERROR

El prefijo y sufijo siempre constan de <CR> y <LF> line feed (salto de línea).

Los comandos que hacen que el módulo realice alguna acción se escriben de la siguiente manera:

AT	<X>
SUFIJO	ACCIÓN

En donde <X> será la acción que realizará el módulo.

A continuación se detallan algunos comandos AT.

Los siguientes comandos pueden ser probados en cualquier modem GSM o teléfono celular que acepte comandos AT:

Acción realizada por el modulo

- ATH: Descuelga el teléfono
- ATI : Revisa la memoria ROM del modulo
- ATM: Conexión/Desconexión del altavoz

Comandos para información del equipo

- AT+CGMI: Identificación del fabricante.
- AT+CGSN: Obtener número de serie.
- AT+CIMI: Obtener el IMSI (Identificación de la Estación Móvil

Internacional.

- AT+CPAS: Leer estado del modem.

Comandos del servicio de red

- 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.

Comandos de seguridad

- AT+CPIN: Introducir el PIN.
- AT+CPINC: Obtener el número de reintentos que quedan.
- AT+CPWD: Cambiar contraseña.

Comandos para la agenda de teléfonos

- AT+CPBR: Leer todas las entradas.
- AT+CPBF: Encontrar una entrada.
- AT+CPBW: Almacenar una entrada.
- AT+CPBS: Buscar una entrada.

### Comandos para SMS

- AT+CPMS: Seleccionar lugar de almacenamiento de los SMS.
- AT+CMGF: Seleccionar formato de los SMS.
- AT+CMGR: Leer un SMS almacenado.
- AT+CMGL: Listar los mensajes almacenados.
- AT+CMGS: Enviar 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.

### Comandos utilizados:

- ATE: Eliminar el ECO, retornan los datos sin en el comando recibido.
- AT: Atención
- ATH: Cuelga Llamada
- AT+CMGL: Lista los Mensajes Almacenados
- AT+CMGS: Enviar mensaje SMS.
- AT+CMGD: Borra mensaje SMS.

### 1.9. Indicador del estado de la red

El pin NETLIGHT se puede utilizar para saber el estado de la red mediante un indicador luminoso, el estado de este pin se enumera en la siguiente tabla:

Estado	Función de SIM900
Apagado	SIM900 no está trabajando
64ms encendido/ 800ms apagado	SIM900 no encuentra la red
64ms encendido / 3000ms apagado	SIM900 encontró la red
64ms encendido / 300ms apagado	Comunicación GPRS

Tabla 1.2. Estado del pin NETLIGHT

### 1.10. Pruebas de comunicación con el modem SIM900

Para realizar pruebas de comunicación con el modem SIM900 utilizaremos un computador (PC) para enviar comandos AT mediante el *hyperterminal* o algún programa que soporte envío y recepción de datos en forma serial.

El modem SIM900 por defecto viene programado con la opción *auto-bauding* activada, que ajustará automáticamente la velocidad de comunicación entre el PC y el módulo, una vez que los dispositivos han sido conectados entre sí. Enlazado el modem con el PC el usuario puede configura el módulo SIM900 como sea necesario.

Para estas pruebas utilizaremos el comunicador serial de ProtonIDE, este software también será ocupado en los capítulos siguientes para la programación del microcontrolador PIC.

#### 1.10.1 Sincronización entre dispositivos

Para probar que el modulo está correctamente enlazado con el PC enviamos el comando "AT", el modem responderá de la siguiente manera:

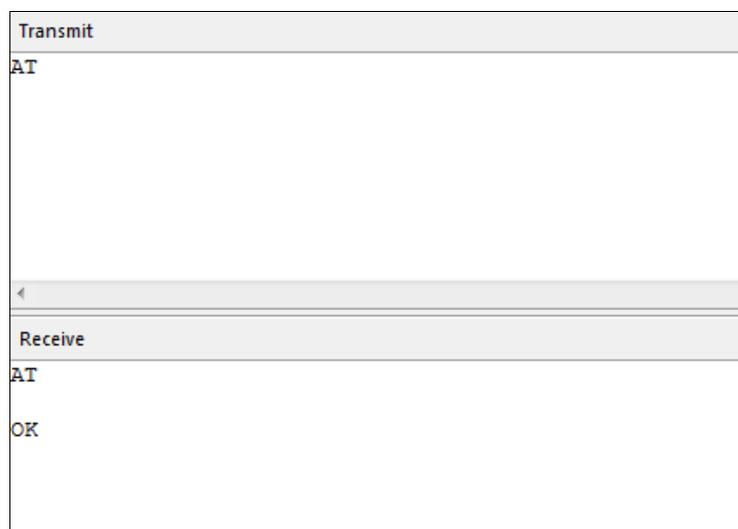


Figura 1.8. Respuesta del Modem SIM900 al comando "AT"

El modulo responde el mismo comando recibido "AT" (opción *ECO* activada) y la palabra "OK" que indica una operación exitosa o en este caso que los dos dispositivos están enlazados correctamente.

Ahora como ejemplo, preguntaremos al módulo la identificación del fabricante con el comando “*AT+CGMI*”, en la siguiente figura se muestra el ejemplo:

Transmit
AT+CGMI
←
Receive
AT+CGMI
SIMCOM_Ltd
OK

Figura 1.9. Identificación del Fabricante

Como se ve en la figura la primera línea de respuesta siempre es el mismo comando que enviamos desde el PC, esto se debe a que la opción de *ECO* está activada; la segunda línea corresponde a la información del fabricante, en este caso *SIMCOM\_Ltd*; la tercera línea es la palabra *OK* que indica una operación exitosa.

Para el control del sistema de domótica es necesario filtrar toda la respuesta del modem SIM900 y la primera línea está demás, por lo que desactivamos la opción de *ECO* del módulo. El comando para activar/desactivar esta opción es: *ATE1* y *ATE0* respectivamente.

### 1.10.2 Configuración del puerto serial

Para configurar el puerto serial utilizamos los comandos “*AT+ICF*” y “*AT+IPR*” e ingresamos los siguientes datos: baudrate = 9600, paridad = 0, tamaño del Byte = 8, Bits de parada = 1.

La tabla 1.3 muestra los parámetros a configurar en el registro *ICF*. El comando que tenemos que enviar, está estructurado de la siguiente manera: *AT+ICF=<format>,<parity>*; para el caso ocupamos 8 datos por cada Byte, sin paridad y 1 bit de parada, el comando quedaría de la siguiente manera: *AT+ICF=3,3*.

La velocidad de transmisión se configura con la opción *auto-bauding* activada. El comando es el siguiente: AT+IPR=<rate>. Rate = 0 para activar la opción auto-bauding, según el manual SIM900\_AT Command Manual\_V1.03.

Parámetros		
<format>	1	8 data 0 parity 2 stop
	2	8 data 1 parity 1 stop
	3	8 data 0 parity 1 stop
	4	7 data 0 parity 2 stop
	5	7 data 1 parity 1 stop
	6	7 data 0 parity 1 stop
<parity>	0	Odd
	1	Even
	3	Space (0)

Tabla 1.3. Configuración del Puerto Serial. (Paridad, bits de parada, datos)

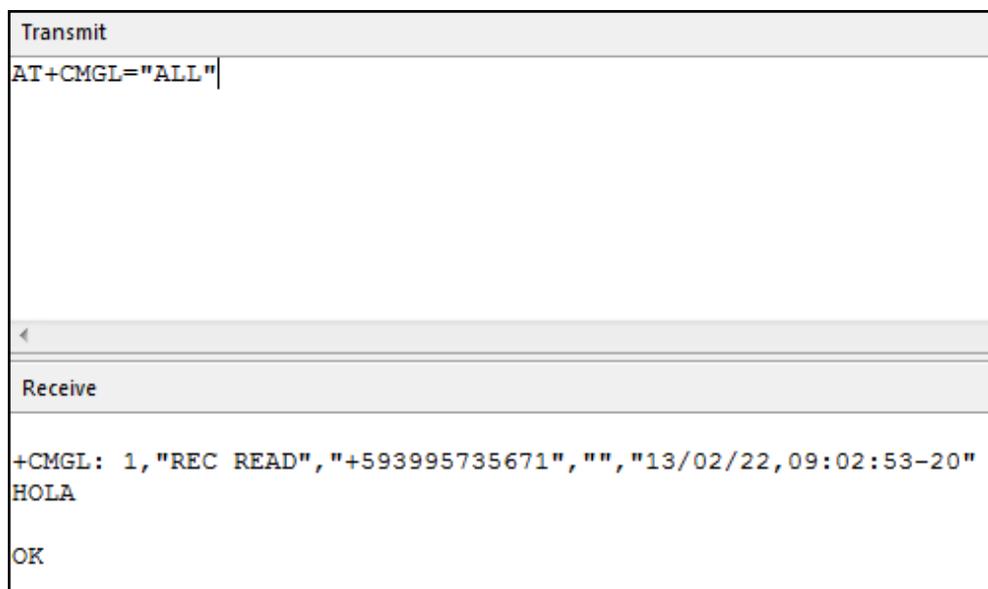
Transmit	AT+ICF=3,3	AT+IPR=0
Receive	OK	OK

Figura 1.10. Configuración de Registros ICF y IPR

### 1.10.3. Lectura de mensajes de texto

Para leer todos los mensajes de texto, enviamos el siguiente comando: `AT+CMGL="ALL"`; el módulo SIM900 responde una cadena de texto con la siguiente información: número de teléfono, fecha, hora y mensaje de texto. En caso que la memoria del modem SIM900 este vacía, la respuesta será la palabra OK. En la figura 1.11 se muestra un ejemplo.

Para esta prueba enviamos un mensaje de texto con la palabra *HOLA* desde cualquier número de teléfono al modem SIM900.



```
Transmit
AT+CMGL="ALL"

Receive
+CMGL: 1,"REC READ","+593995735671","", "13/02/22,09:02:53-20"
HOLA
OK
```

Figura 1.11. Lectura de un mensaje de texto

En la figura se muestra la cadena de caracteres que responde el modulo, este mensaje tiene que ser filtrado para obtener solamente el mensaje de texto (HOLA) y el número de teléfono (+593995735671 en este caso).

### 1.10.4 Borrar la memoria de mensajes

Una vez leído el SMS se puede borrar la memoria de mensajes del módulo con el siguiente comando: `AT+CMGD=<index>,<delflag>`; el comando para borrar todos los mensajes es `AT+CMGD=0,4` (Según el manual SIM900\_AT Command Manual\_V1.03).

Transmit
AT+CMGD=0,4
<
Receive
OK

Figura 1.12. Borrado de memoria de mensajes

Estos son algunos de los comandos que ocuparemos para el control de los diferentes sistemas, estos comandos serán enviados y receptados por el microcontrolador PIC.

## **CAPÍTULO 2**

### **MICROCONTROLADOR PIC**

#### **2.1 Introducción**

Un microcontrolador es un computador digital encapsulado en un chip, que cuenta con un microprocesador o unidad central de proceso (CPU), una memoria para almacenar el código del 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 autosuficientes y económicos.

El comportamiento de los microcontroladores está definido por el programa almacenado en su memoria. Este puede escribirse en distintos lenguajes de programación. La mayoría de los microcontroladores actuales pueden reprogramarse varias veces.

Por las características indicadas y su alta flexibilidad, los microcontroladores son ampliamente utilizados como la parte principal de una gran variedad de sistemas que controlan máquinas, componentes de sistemas complejos, aplicaciones industriales de automatización, robótica, domótica, equipos médicos, sistemas aeroespaciales, e incluso dispositivos de la vida diaria como automóviles, hornos de microondas, teléfonos, televisores, etc.

Comúnmente se emplea la notación uC o las siglas MCU (por microcontroller unit) para referirse a los microcontroladores. De ahora en adelante, los microcontroladores serán referidos en este documento por uC.

#### **2.2. Tipos de arquitecturas de los microcontroladores PIC**

##### **2.2.1. Arquitectura Von Neumann**

La construcción tradicional de computadoras y microprocesadores está basada en la arquitectura Von Neumann, en la cual la unidad central de proceso, está conectada a una memoria única donde se guardan las instrucciones del programa y los datos (figura 2.1). El tamaño de la unidad de datos o instrucciones está fijado

por el ancho del bus que comunica la memoria con la CPU. Así un microprocesador de 8 bits, tendrá que manejar datos e instrucciones de una o más unidades de 8 bits (1 bytes) de longitud. Si tiene que acceder a una instrucción o dato de más de un byte de longitud, tendrá que realizar más de un acceso a la memoria. Y el tener un único bus hace que el microprocesador sea más lento en su respuesta, ya que no puede buscar en la memoria una nueva instrucción mientras no finalicen las transferencias de datos de la instrucción anterior. Las principales limitaciones que nos encontramos con la arquitectura Von Neumann son:

1. La limitación de la longitud de las instrucciones por el bus de datos, que hace que el microprocesador tenga que realizar varios accesos a memoria para buscar instrucciones complejas.
2. La limitación de la velocidad de trabajo a causa del bus único para datos e instrucciones que no deja acceder simultáneamente a datos e instrucciones, lo cual impide superponer ambos tiempos de acceso.

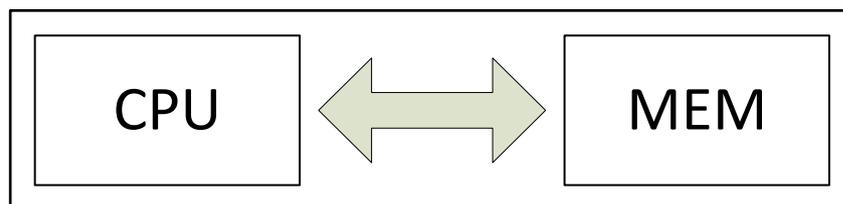


Figura 2.1. Arquitectura Von Neumann

### 2.2.2. Arquitectura Harvard

En la arquitectura Harvard la unidad central de proceso está conectada a dos memorias (instrucciones y datos) por medio de dos buses diferentes (figura 2.2). Una de las memorias contiene solamente las instrucciones del programa (Memoria de Programa), y la otra sólo almacena datos (Memoria de Datos). Cada bus es independiente del otro y pueden ser de distintos anchos. Para un procesador de Set de Instrucciones Reducido, o RISC (Reduced Instrucción Set Computer), el set de instrucciones y el bus de memoria de programa pueden diseñarse de manera que todas las instrucciones tengan una sola posición de memoria. Además, al ser los buses independientes, la CPU puede acceder a los datos para completar la

ejecución de una instrucción, y al mismo tiempo leer la siguiente instrucción a ejecutar. Ventajas de esta arquitectura:

1. El tamaño de las instrucciones no está relacionado con el de los datos, y por lo tanto puede ser optimizado para que cualquier instrucción ocupe una sola posición de memoria de programa, logrando así mayor velocidad y menor longitud de programa.
2. El tiempo de acceso a las instrucciones puede superponerse con el de los datos, logrando una mayor velocidad en cada operación.

Una pequeña desventaja de los procesadores con arquitectura Harvard, es que deben poseer instrucciones especiales para acceder a tablas de valores constantes que pueda ser necesario incluir en los programas, ya que estas tablas se encontraran físicamente en la memoria de programa (por ejemplo en la EPROM de un microprocesador)

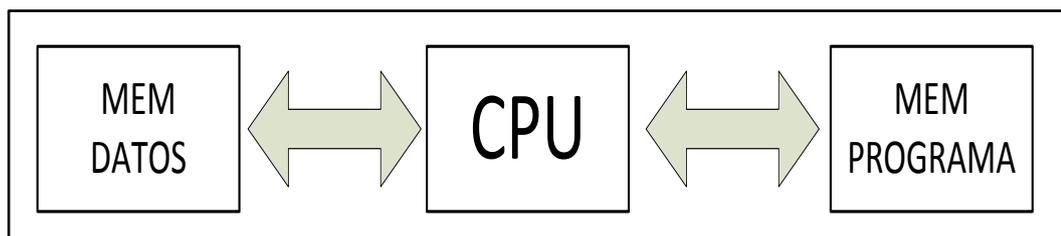


Figura 2.2. Arquitectura Harvard

### 2.3. Elementos de los microcontroladores

Describiremos los elementos más comunes en todo tipo de microcontroladores.

#### 2.3.1 El procesador

Es el elemento más importante del microcontrolador y establece sus principales características, tanto a nivel de hardware como de software. Se encarga de direccionar la memoria de instrucciones, decodifica y ejecuta la operación que implica la instrucción, busca de los operandos y el almacenamiento del resultado. Existen tres orientaciones en cuanto a la arquitectura y funcionalidad de los procesadores actuales.

- CISC

Un gran número de procesadores utilizados en los microcontroladores están basados en la filosofía CISC (Computadores de Juego de Instrucciones Complejo). Disponen de más de 80 instrucciones de máquina en su repertorio, algunas de las cuales son muy sofisticadas y potentes, requiriendo muchos ciclos para su ejecución. Una ventaja de los procesadores CISC es que ofrecen al programador instrucciones complejas que actúan como macros.

- RISC

Tanto la industria de los computadores comerciales como la de los microcontroladores están decantándose hacia la filosofía RISC (Computadores de Juego de Instrucciones Reducido). En estos procesadores el repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y, generalmente, se ejecutan en un ciclo.

La sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del procesador.

- SISC

En los microcontroladores destinados a aplicaciones muy concretas, el juego de instrucciones, además de ser reducido, es específico, o sea, las instrucciones se adaptan a las necesidades de la aplicación prevista. Esta filosofía se ha bautizado con el nombre de SISC (Computadores de Juego de Instrucciones Específico).

### **2.3.2 Memoria**

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

Hay dos propiedades que diferencian a los microcontroladores de los computadores personales:

No tienen sistemas de almacenamiento masivo como disco duro o disquetes. Como el microcontrolador sólo se destina a una tarea en la memoria ROM, sólo hay que almacenar un único programa de trabajo.

La RAM en estos dispositivos es de poca capacidad pues sólo debe almacenar las variables y los cambios de información que se produzcan en el transcurso del programa. Por otra parte, como sólo existe un programa activo, no se requiere guardar una copia del mismo en la RAM pues se ejecuta directamente desde la ROM.

Los usuarios de computadores personales están familiarizados a manejar Megabytes de memoria, pero, los diseñadores con microcontroladores trabajan con capacidades de ROM comprendidas entre 512 bytes y 8k bytes y de RAM comprendidas entre 20 y 512 bytes.

Según el tipo de memoria ROM que dispongan los microcontroladores, la aplicación y utilización de los mismos es diferente. Se detallan las cinco versiones de memoria no volátil que se pueden encontrar en los microcontroladores del mercado.

- ROM con máscara

Es una memoria no volátil sólo de lectura, su contenido se graba durante la fabricación del chip. El nombre se lo da por la fabricación, en la cual se ocupan unas máscaras con agujeros que permiten el paso de luz con lo cual se impregnan las capas de silicio y oxido de silicio. El elevado costo del diseño de la máscara sólo hace aconsejable el empleo de los microcontroladores con este tipo de memoria cuando se precisan cantidades superiores a varios miles de unidades.

- OTP

El microcontrolador contiene una memoria no volátil de sólo lectura programable una sola vez por el usuario. OTP (One Time Programmable). Es el usuario quien escribe el programa en el chip mediante un grabador controlado por un programa desde un computador personal. La versión OTP es recomendable cuando el ciclo de diseño del producto es pequeño, o bien, en la construcción de prototipos y series muy pequeñas. Tanto en este tipo de memoria como en la EPROM, se suele usar la encriptación mediante fusibles para proteger el código contenido.

- EPROM

Los microcontroladores que disponen de memoria EPROM (Erasable Programmable Read Only Memory) pueden borrarse y grabarse varias veces. La grabación se realiza, como en el caso de los OTP, con un grabador controlado desde un PC. Si, posteriormente, se desea borrar el contenido, disponen de una ventana de cristal en su superficie por la que se somete a la EPROM a rayos ultravioleta durante varios minutos. Las cápsulas son de material cerámico y son más caros que los microcontroladores con memoria OTP que están hechos con material plástico.

- EEPROM, E2PROM o E2PROM

Se trata de memorias de sólo lectura, programables y borrables eléctricamente EEPROM (Electrical Erasable Programmable Read Only Memory). Tanto la programación como el borrado, se realizan eléctricamente desde el propio grabador y bajo el control programado de un PC. Es muy cómoda y rápida la operación de grabado y de borrado. No disponen de ventana de cristal en la superficie. Los microcontroladores con memoria EEPROM una vez instalados en el circuito, pueden grabarse y borrarse cuantas veces se quiera sin ser retirados de dicho circuito. Para ello se usan "grabadores en circuito" que dan una gran flexibilidad y rapidez a la hora de realizar modificaciones en el programa de trabajo. El número de veces que puede grabarse y borrarse una memoria EEPROM es limitado, por lo que no es recomendable una reprogramación continua. Los fabricantes tienden a incluir una pequeña zona de memoria EEPROM en los circuitos programables para guardar y modificar los parámetros que adecuan el dispositivo a las condiciones del entorno. Este tipo de memoria es relativamente lenta.

- FLASH

Se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar. Funciona como una ROM y una RAM, consume menos y es más pequeña. A diferencia de la ROM, la memoria FLASH es programable en el circuito. Es más rápida y de mayor densidad que la EEPROM. La opción de memoria FLASH está recomendada frente a la EEPROM cuando se necesita una gran cantidad de memoria de programa no volátil. Es más veloz y permite más ciclos de escritura/borrado. Las memorias EEPROM y FLASH son muy

útiles al permitir que los microcontroladores que las incorporan puedan ser reprogramados en el circuito, es decir, sin tener que sacar el circuito integrado de la tarjeta. Así, un dispositivo con este tipo de memoria incorporado al control de cualquier tipo de sistema permite modificar el programa durante la rutina de mantenimiento periódico. La reprogramación del microcontrolador puede convertirse en una labor rutinaria dentro de la puesta a punto.

### **2.3.3 Puertas de entrada y salida**

Las puertas de Entrada y Salida (E/S) comunican al procesador con cualquier tipo de periféricos, a través de interfaces. Estas puertas, también llamadas puertos, son la principal utilidad de los pines de un microprocesador. Según el tipo de microcontrolador estos pines son destinados para señales de entrada, salida y control.

### **2.3.4 Reloj principal**

Todos los microcontroladores poseen un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema. Esta señal del reloj es el motor del sistema y la que hace que el programa y los contadores avancen.

La señal de reloj está incorporada en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la frecuencia de trabajo. Estos componentes consisten en un cristal de cuarzo junto a elementos pasivos o bien un resonador cerámico o una red R-C.

Aumentar la frecuencia de reloj supone disminuir el tiempo en que se ejecutan las instrucciones pero lleva aparejado un incremento del consumo de energía y de calor generado.

## **2.4. Recursos principales**

Cada fabricante ofrece numerosas versiones de una arquitectura básica de microcontrolador. En algunas aumenta 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 costo, 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 (Sleep mode).
- Conversor A/D (Analógico ->Digital).
- Conversor D/A (Digital ->Analógico).
- Comparador analógico.
- Modulador de anchura de impulsos o PWM (Pulse Wide Modulation).
- Puertas de E/S digitales.
- Puertas de comunicación.

A continuación pasamos a ver con un poco más de detalle cada uno de ellos.

#### **2.4.1 Temporizadores o timers**

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

#### **2.4.2. Perro guardián o Watchdog**

Cuando el computador personal se bloquea por un fallo del software u otra causa, se pulsa el botón del reset y se reinicia 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 contador que, cuando llega al máximo, provoca un reset automáticamente en el sistema.

Se debe diseñar el programa de trabajo que controla la tarea, de forma que resetee al Perro Guardián de vez en cuando antes de que provoque el reset. Si falla el programa o se bloquea (si cae en bucle infinito), no se refrescará al Perro Guardián y, al completar su temporización, provocará el reset del sistema.

#### **2.4.3 Protección ante fallo de alimentación o Brownout**

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. Esto es muy útil para evitar datos erróneos por transiciones y ruidos en la línea de alimentación.

#### **2.4.4. Estado de reposo o de bajo consumo**

Son muchas las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, hasta 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. Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo.

#### **2.4.5. Conversor A/D (CAD)**

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

#### **2.4.6. Conversor D/A (CDA)**

Transforma los datos digitales obtenidos del procesamiento del CPU en su correspondiente señal analógica que saca al exterior por una de las patillas del chip. Existen muchos circuitos que trabajan con señales analógicas.

### **2.4.7. Comparador analógico**

Algunos microcontroladores tienen internamente de un amplificador operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por uno de los pines 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.

### **2.4.8. Modulador de anchura de impulsos o PWM**

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

### **2.4.9. Puertos digitales de entrada y salida E/S**

Todos los microcontroladores destinan parte de sus pines a líneas de E/S 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.

### **2.4.10. Puertas de comunicación**

Son pines destinados para darle al microcontrolador 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. (Ej: Puerto Serie)

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.

CAN (Controller Area Network), para permitir la adaptación con redes de conexasión multiplexado, desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles.

## 2.5. Familia PIC18F y PIC 16F

### 2.5.1 Características Fundamentales PIC18F

- Arquitectura RISC avanzada Harvard.
- 77 instrucciones.
- Desde 18 a 80 pines
- Hasta 64K bytes de programa (hasta 2 Mbyte en ROMless)
- Hasta 3968bytes de RAM y 1 Kbyte de EEPROM
- Frecuencia máxima de Reloj 48MHz
- Pila de 32 Niveles
- Múltiples fuentes de interrupción
- Periféricos de comunicación avanzados
- 

Microcontroladores PIC18F2455, PIC18F2450, PIC18F4455, PIC18F4550

CARACTERISTICAS	PIC18F2455	PIC18F2450	PIC18F4455	PIC18F4550
Frecuencia de Operación	Hasta 48MHz	Hasta 48MHz	Hasta 48MHz	Hasta 48MHz
Memoria de Programa (bytes)	24576	32768	24576	32768
Memoria RAM de Datos (bytes)	2048	2048	2048	2048
Memoria EEPROM Datos (bytes)	256	256	256	256
Interrupciones	19	19	20	20
Líneas de Entrada/Salida	24	24	35	35
Temporizadores	4	4	4	4

Continuación:

Módulos de Comparación/Captura/PWM (CCP)	2	2	1	1
Módulos de Comparación/Captura/PWM (CCP) Mejorado (ECCP)	0	0	1	1
Canales de Comunicación Serie	MSSP,EUSART	MSSP,EUSART	MSSP,EUSART	MSSP,EUSART
Canal USB	1	1	1	1
Puerto paralelo de transmisión de Datos	0	0	1	1
Canales de Conversión de A/D de 10 bits	10 Canales	10 Canales	13 Canales	13 Canales
Comparadores analógicos	2	2	2	2
Juego de Instrucciones	75	75	75	75
Encapsulados	PDIP 28 PINES SOIC 28 PINES	PDIP 28 PINES SOIC 28 PINES	PDIP 40PINES QFN 40 PINES TQFP 40 PINES	PDIP 40PINES QFN 40 PINES TQFP 40 PINES

Tabla 2.1. Características de los PIC de la Familia 18F. (Para el proyecto se utiliza el PIC18F4550).

### 2.5.2 Características Fundamentales PIC16F:

CARACTERÍSTICAS	PIC16 F616	PIC16F 628A	PIC16 F684	PIC16 F688	PIC16 F690	PIC16F 73	PIC16 F84A	PIC16 F88
Frecuencia de Operación	Hasta 48MHz	Hasta 48MHz	Hasta 48MHz	Hasta 48MHz	Hasta 20MHz	Hasta 20MHz	Hasta 4MHz	Hasta 8MHz
Memoria de Programa (bytes)	2000	2000	2000	4000	4000	4000	1000	7000
Memoria RAM Datos (bytes)	128	224	256	256	256	192	68	368
Memoria EEPROM Datos (bytes)	128	128	256	256	256	128	64	256
Líneas de E/S	12	16	12	12	18	22	13	16
Temporizadores	3	3	3	2	2	3	1	3

Tabla 2.2. Características de los PIC de la Familia 16F. (Para el proyecto se utiliza el PIC16F628A)

### 2.6. Proceso de Desarrollo

La alta integración de subsistemas que componen un uC reduce el número de circuitos integrados, la cantidad de pistas y espacio que se requiere en una placa de circuito impreso, en comparación a un sistema equivalente usando chips separados.

Una parte muy importante para el desarrollador de circuitos basados en microcontroladores son las interfaces de entrada/salida (E/S). A través de los pines del chip asociados a las interfaces de E/S, el uC puede interactuar con otros circuitos externos enviándoles señales de comando o recibiendo datos correspondientes a variables externas. Por lo general varios pines de datos son bidireccionales, es decir pueden configurarse como entradas o salidas.

Los pines de entrada, pueden adquirir datos interpretando el valor de voltaje como un valor lógico 0 o 1, mientras que cuando son salidas pueden entregar una señal binaria de voltaje cuya magnitud dependerá del valor lógico 0 o 1. Monitoreando el valor de las entradas, el microcontrolador puede responder a eventos externos y realizar una cierta acción, como variar las señales de salida de acuerdo al valor en las entradas. Para responder a eventos externos, los uCs cuentan con un recurso conocido como interrupciones. Una vez ejecutada la subrutina de interrupción la ejecución del programa continúa en el punto en que se encontraba antes de generarse la interrupción. Un ejemplo es el de un botón pulsador conectado a un pin de entrada. Una vez pulsado, se genera una señal de interrupción que iniciaría la ejecución de la subrutina de interrupción, que por ejemplo podría activar un pin de salida para encender un led.

No todas las interrupciones necesariamente están asociadas al cambio del estado de los pines de entrada. También hay interrupciones que pueden estar asociadas al valor de una entrada AD, o al cumplimiento de un periodo de tiempo fijado por un timer o temporizador. Estas características dependerán del modelo de uC empleado.

El proceso de desarrollo de una aplicación basada en microcontroladores se compone de las siguientes etapas principales, las cuales se explican en más detalle en las siguientes subsecciones.

### **2.6.1. Desarrollo del programa**

En esta etapa consiste en escribir y compilar/ensamblar el programa que determinará las acciones del uC y su funcionamiento. Existen distintas maneras de desarrollar el programa, dependiendo del lenguaje inicial que se utiliza para escribir el programa.

- Lenguaje Assembly - Lenguaje de Máquina/Código Objeto  
(.asm) → Ensamblador → (.hex, .o, .bin, .coff)
- Lenguaje de Alto Nivel - Lenguaje Assembly - Lenguaje de Máquina/Código Objeto (.c, .cpp) → Compilador → (.asm) → ensamblador → (.hex, .o, .bin, .coff)

En la figura 2.3 se muestran las dos alternativas típicas que tiene el desarrollador para generar el código de máquina que es entendido por el microcontrolador.

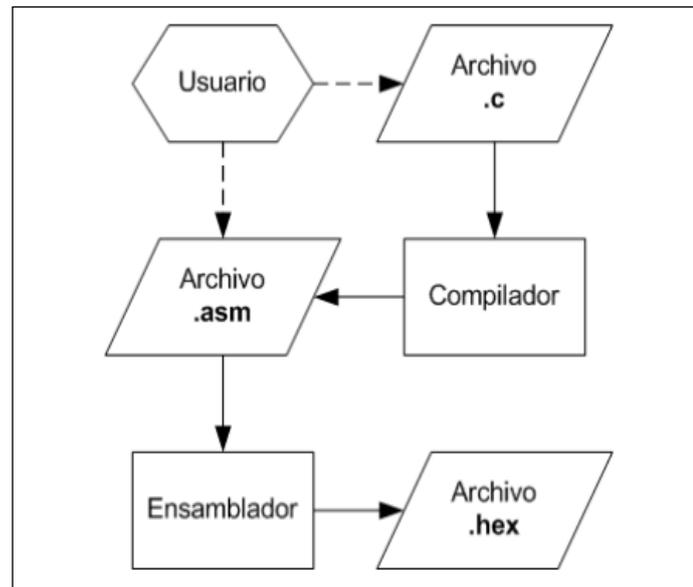


Figura 2.3. Etapas del desarrollo del software

El método básico es escribir el programa en lenguaje Assembly (lenguaje de Ensamblador) en un archivo de texto con extensión `.asm` y luego utilizar una programa ensamblador (Assembler) para generar un archivo en lenguaje de máquina, también denominado código de máquina o código objeto (object code), compuesto por instrucciones en código binario que son directamente entendidas por la CPU del microcontrolador.

El ensamblador normalmente genera un archivo con extensión `.hex` (por hexadecimal), `.o` (por objeto), `.bin` (por binario), o `.coff` (common object file format) dependiendo del ensamblador. El lenguaje Assembly se compone de instrucciones mnemónicas de bajo nivel, es decir que están ligadas a las características del microcontrolador y con un número mínimo o nulo de abstracciones. Al carecer de abstracciones, el lenguaje Assembly es más difícil de emplear, requiere experiencia y un mayor tiempo de desarrollo.

La ventaja es que el código de máquina generado a partir de un programa escrito en lenguaje de máquina es por lo general más eficiente, ya que el programa se desarrolla en un nivel cercano a las características del hardware.

Otra alternativa es emplear un lenguaje de alto nivel con una mayor cantidad de abstracciones, la cuales son más fáciles de usar y reducen los tiempos de desarrollo. Tal vez los lenguajes de alto nivel más comunes para la programación de controladores es el C y C++, pero también existen otros lenguajes variantes del

BASIC y el Pascal. Una vez escrito el programa en el lenguaje de alto nivel, será necesario emplear un compilador para traducirlo, ya sea a lenguaje de ensamblador o directamente a lenguaje de máquina.

Para el desarrollo del proyecto se utilizará el software Proton IDE que emplea un lenguaje de alto nivel.

Proton IDE es un poderoso y profesional entorno de desarrollo integrado (IDE), que ha sido diseñado específicamente para el compilador Proton Plus.

Proton IDE acelera el desarrollo de productos en un entorno de usuario confortable sin comprometer el rendimiento, la flexibilidad o control.

Utilizando el software Proton IDE (compilador/ensamblador) figura 2.4, se procede de la siguiente manera:

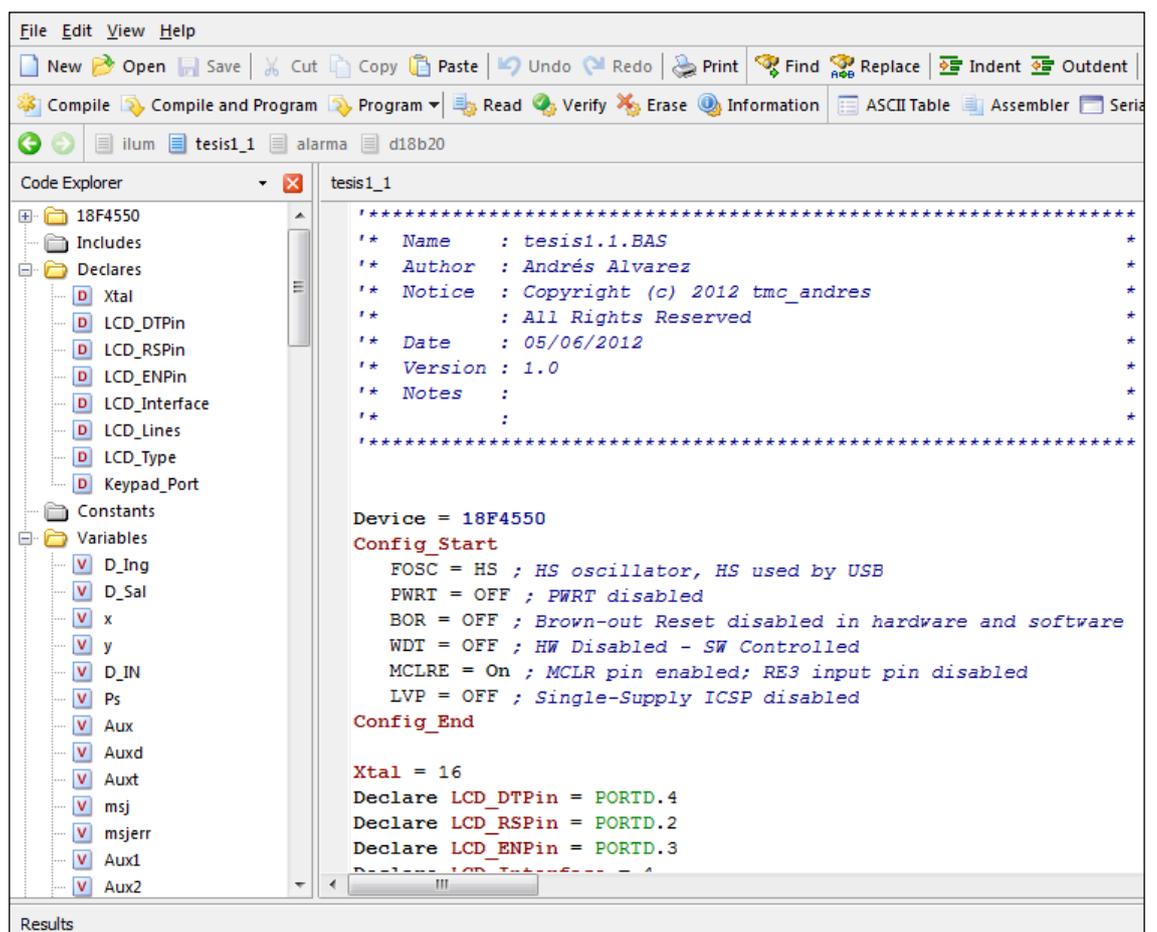


Figura 2.4. Programa Proton IDE.

- Se define el dispositivo que se va a programar:  
Para el PIC18F4550: **Device** = **18F4550**.  
Para el PIC16F628A: **Device** = **18F628A**  
Sentencia para el dispositivo a ser utilizado
- Configuración de bits para cada tipo de PIC y según el uso que se quiera dar al mismo, se muestra un ejemplo para el PIC18F4550:

**Config\_Start**

```

FOSC = HS ; HS oscillator, HS used by USB
PWRT = OFF ; PWRT disabled
BOR = OFF ; Brown-out Reset disabled in
hardware and software
WDT = OFF ; HW Disabled - SW Controlled
MCLRE = On ; MCLR pin enabled; RE3 input
pin disabled
LVP = OFF ; Single-Supply ICSP disabled
Config_End

```

Configuración de bits (PIC18F4550).

- Configuración del Oscilador, según el tipo de cristal con el que se vaya a emplear, ejemplo 16MHz.

```
Xtal = 16.
```

Configuración del Oscilador para 16MHz

- En caso de utilizar teclados, LCDs u otros elementos, tendrán que ser declarados en el programa.

```

Declare LCD_DTPin = PORTB.4
Declare LCD_RSPin = PORTB.3
Declare LCD_ENPin = PORTB.2
Declare LCD_Interface = 4
Declare LCD_Lines = 2
Declare LCD_Type = 0

```

Declaración para LCD de dos filas conectado a la parte alta del PuertoB.

- Escribir el código que será descargado al microcontrolador PIC.

```

dat_ing:
SerIn PORTA.2,16468,[D_Ing1]
SerIn PORTA.2,16468,[D_Ing2]

    If D_Ing1 = "1" And D_Ing2 = "0" Then
GoTo desact
    If D_Ing1 = "2" Then
        DelayMS 50
        SerOut PORTA.3,16468,[x]
        DelayMS 20
        SerOut PORTA.3,16468,[y]
        DelayMS 20
    End If
Return

```

Subrutina para leer los datos ingresados en forma serial.

## 2.6.2. Programación del microcontrolador

Una vez escrito todo el programa, el software Proton IDE funciona como compilador y ensamblador ya que nos genera los archivos .hex y .asm, mediante un programador (PicKit2) nos permite grabar el archivo .hex al microcontrolador.

Este proceso corresponde a utilizar un programa en el PC (figura 2.6) que toma el código ensamblado (.hex, .o, .bin, .coff) para el uC específico, y lo envía mediante algún puerto (serial, paralelo, USB, etc.) a un dispositivo (figura 2.7) que lo escribe en la memoria del uC. El software programador a veces recibe también el nombre de downloader, ya que su propósito es descargar o transferir desde el PC al uC el código ensamblado.

En la figura 2.5 se muestran los componentes involucrados en el proceso de programación del uC. Es importante mencionar que no deben confundirse los términos desarrollo o programación del software y programación del uC, el primero se refiere a escribir el programa, mientras que el segundo se refiere a transferir el código de máquina a la memoria del uC.

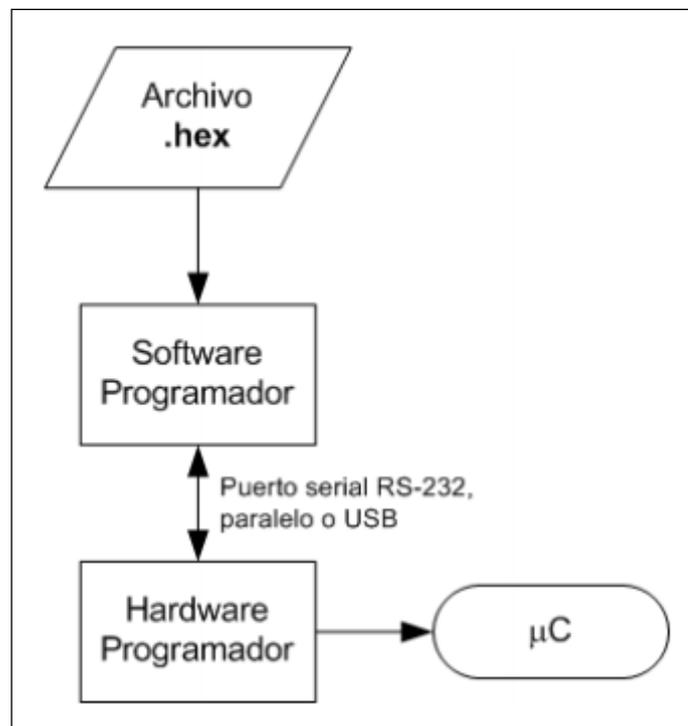


Figura 2.5. Programación del Microcontrolador

Para grabar el programa en el uC utilizaremos el programa PICKit2v 2.61 (figura 2.5).

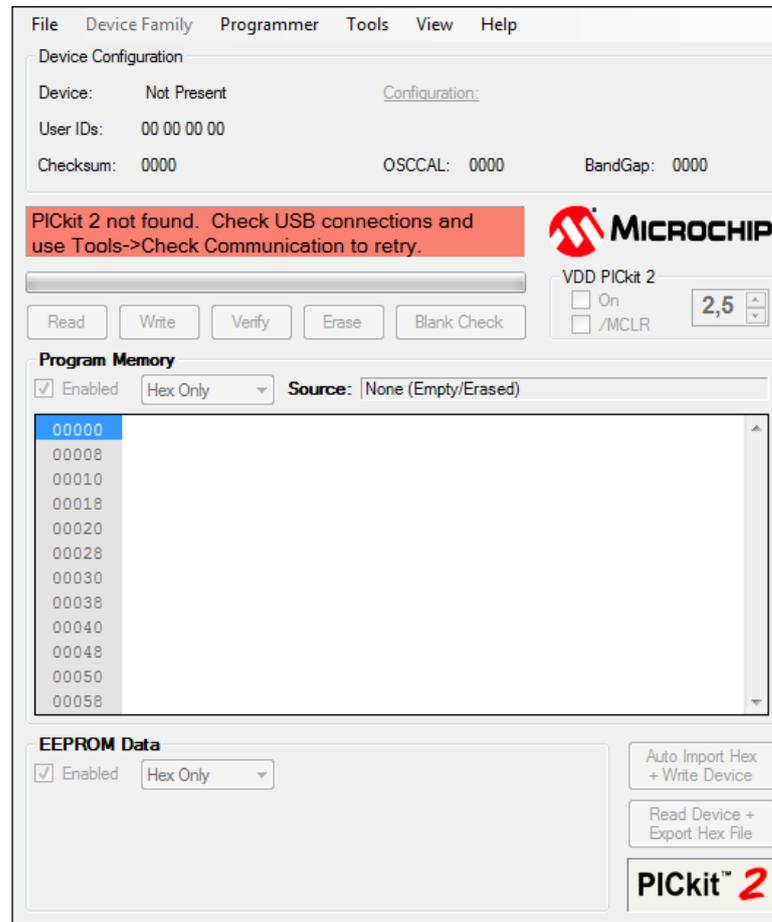


Figura 2.6. Software Programador PICKit2 v2.61



Figura 2.7. Hardware programador PICKit2

Fuente: MICROCHIP. <http://ww1.microchip.com/downloads/en/devicedoc/51553e.pdf>. [Consulta: 7 de Octubre de 2012]

El grabador se conecta a un PC mediante un cable USB, y será reconocido por el programa automáticamente. Luego se conecta el microcontrolador al grabador mediante los pines correspondientes.

El programa PICKit2 reconocerá automáticamente el uC conectado, se carga el archivo .hex generado por el software Proton IDE y se escribe los datos en el uC, de esta manera el uC estará listo para su funcionamiento en el circuito respectivo (figura 7).

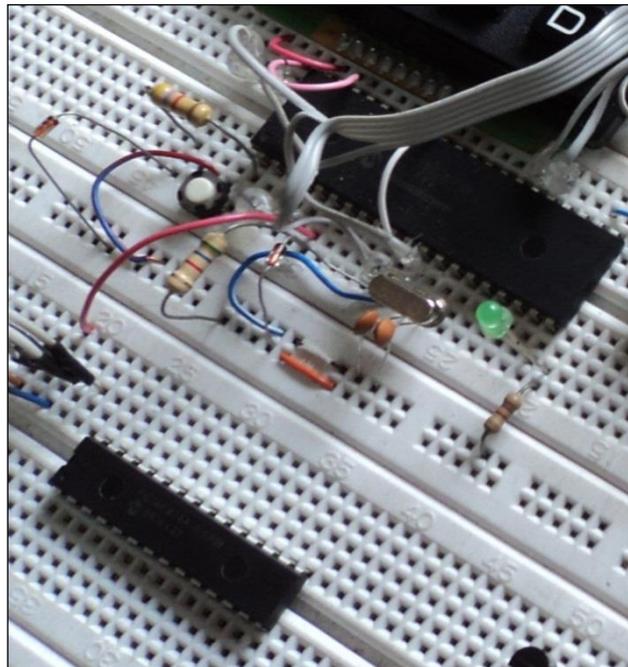


Figura 2.8. Microcontrolador PIC Programado

### 2.6.3. Prueba y verificación

Una vez programado el uC, se puede instalar en el circuito final para comprobar su adecuado funcionamiento. Existen herramientas de software que permiten simular el comportamiento de un uC, muy útiles cuando el programa alcanza cierta complejidad. Para resolver problemas en un circuito real, el instrumento más utilizado es el analizador lógico.

## CAPÍTULO 3

### COMUNICACIÓN ENTRE MICROCONTROLADOR PIC Y MÓDULO GSM.

El control del sistema de Iluminación, Alarma y Climatización se hace mediante mensajes cortos de texto (SMS) de forma remota y con un teclado de manera local. Los SMS serán enviados a través de una aplicación para teléfono móvil lo que hará más amigable la interfaz entre el usuario y el Sistema de Control de Domótica.

Los mensajes de texto receptados por el modulo serán filtrados por el uC PIC18F4550 y este será capaz de hacer actuar a los diferentes sistemas controlados cada uno por un microcontrolador PIC16F628A. Ejemplo de SMS enviado hacia el módulo para encender una luz al nivel máximo de iluminación: I1A, para activar el Sistema de Alarma en Modo Ausente: AAA. Más adelante se explicará cómo filtrar los mensajes.

La comunicación entre el módulo SIM900 y el uC se hará de forma serial asíncrona, utilizando comandos AT como se explicó en el Capítulo 1 sección 1.9. Pruebas con el modem SIM900.

#### 3.1. Protocolo RS-232

El puerto serial es conocido como puerto RS-232, este nos permite las comunicaciones entre otros dispositivos, para nuestro caso con el módulo SIM900.

Existen dos formas de intercambiar información binaria: la paralela y la serial.

La comunicación paralela transmite todos los bits de un dato de manera simultánea, por lo tanto la velocidad de transferencia es rápida, sin embargo tiene la desventaja de utilizar una gran cantidad de líneas, por lo tanto se vuelve más costoso y tiene las desventaja de atenuarse a grandes distancias, por la capacitancia entre conductores así como sus parámetros distribuidos.

Tipos de Comunicaciones Seriales:

Existen dos tipos de comunicaciones seriales: la síncrona y asíncrona.

### 3.1.1. Comunicación serial síncrona

En la comunicación serial síncrona, se necesitan 2 líneas, una línea sobre la cual se transmitirán los datos y otra la cual contendrá los pulsos de reloj que indicaran cuando un dato es válido.

Ejemplos de este tipo de comunicación son los protocolos:

- I2C (Inter Integrated Circuit)
- SPI (Serial Peripheral Interfaz)

En la siguiente figura se muestra la estructura de la comunicación serial síncrona.

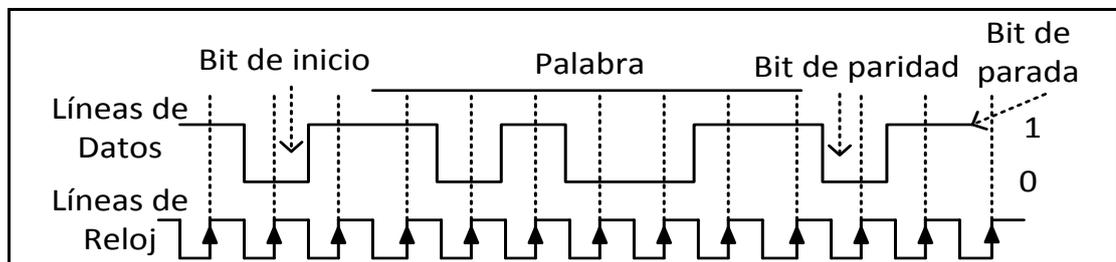


Figura 3.1. Transmisión Serial Síncrona

### 3.1.2. Comunicación serial asíncrona

En la comunicación serial asíncrona, no son necesarios los pulsos de reloj. La duración de cada bit está determinada por la velocidad con la cual se realiza la transferencia de datos.

La siguiente figura muestra la estructura de un carácter que se trasmite en forma serial asíncrona.

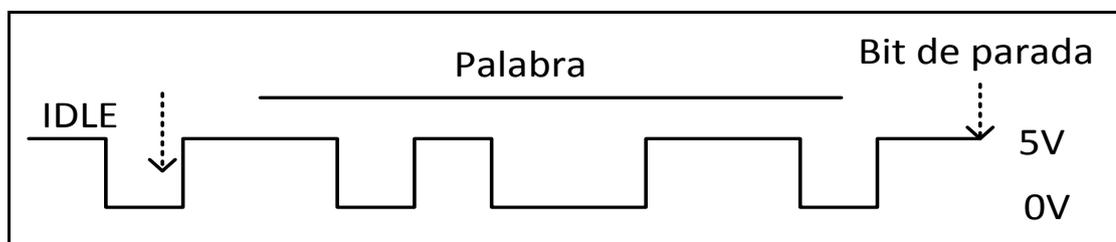


Figura 3.2. Transmisión de un byte en forma asíncrona

Normalmente cuando no se realiza ninguna transferencia de datos, la línea del transmisor se encuentra en estado de IDLE, esto quiere decir en un estado alto.

Para iniciar la transmisión de datos, el transmisor coloca esta línea en nivel bajo durante un determinado tiempo, a lo cual se le conoce como bit de arranque (start bit) y a continuación empieza a transmitir en un intervalo de tiempo fijo, los bits correspondientes al dato, empezando siempre por el BIT menos significativo (LSB) y terminando con el BIT más significativo.

Si el receptor no está sincronizado con el transmisor, este desconoce cuándo se van a recibir los datos, por lo tanto el transmisor y el receptor deberán tener los mismos parámetros de velocidad, paridad, número de bits del dato transmitido y de bit de parada.

En circuitos digitales, cuyas distancias son relativamente cortas, se puede manejar transmisiones en niveles lógicos TTL (0-5V), pero cuando las distancias aumentan, estas señales tienden a distorsionarse debido al efecto capacitivo de los conductores y su resistencia eléctrica. El efecto se incrementa a medida que se incrementa la velocidad de la transmisión.

Todo esto origina que los datos recibidos no sean igual a los datos transmitidos, por lo que no se puede dar la transferencia de datos.

Una de las soluciones más lógicas, es aumentar los márgenes de voltaje con que se transmiten los datos, de tal manera que las perturbaciones por causa de la línea se puedan corregir.

### **3.1.3. La Norma RS-232**

Ante la gran variedad de equipos, sistemas y protocolos que existen surgió la necesidad de un acuerdo que permitiera a los equipos de varios fabricantes comunicarse entre sí. La EIA (Electronics Industry Association) elaboró la norma RS-232, la cual define la interfaz mecánica, los pines, las señales y los protocolos que debe cumplir la comunicación serial.

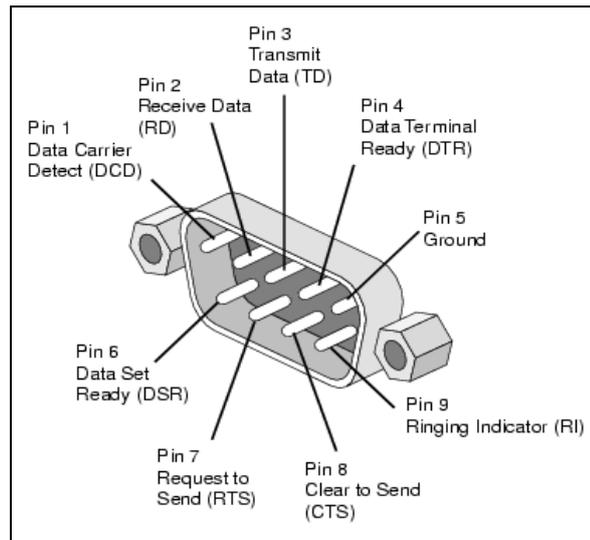


Figura 3.3. Conector RS-232 tipo hembra.

Fuente: DB9-PINOUT. <http://www.db9-pinout.com/db9-pinout/db9-pinout.gif>. [Consulta: 27 de Octubre de 2012]

Todas las normas RS-232 cumplen con los siguientes niveles de voltaje:

- Un "1" lógico es un voltaje comprendido entre  $-5V$  y  $-15V$  en el transmisor y entre  $-3V$  y  $-25V$  en el receptor.
- Un "0" lógico es un voltaje comprendido entre  $+5V$  y  $+15V$  en el transmisor y entre  $+3V$  y  $+25V$  en el receptor.

El envío de niveles lógicos (bits) a través de cables o líneas de transmisión necesita la conversión a voltajes apropiados. En los microcontroladores para representar un '0 lógico' se trabaja con voltajes inferiores a  $0.8V$  y para un '1 lógico' con voltajes mayores a  $2V$ . En general cuando se trabaja con familias TTL y CMOS se asume que un "0" lógico es igual a cero Volts y un "1" lógico es igual a 5 Voltios.

### 3.1.2. El Circuito MAX-232

Este circuito soluciona los problemas de niveles de voltaje cuando se requiere enviar señales digitales sobre una línea RS-232. Este chip se utiliza en aquellas aplicaciones donde no se dispone de fuentes dobles de  $+12V$  y  $-12V$ . El MAX 232 necesita solamente una fuente de  $+5V$  para su operación, internamente tiene un elevador de voltaje que convierte el voltaje de  $+5V$  al de doble polaridad de  $+12V$  y  $-12V$ .

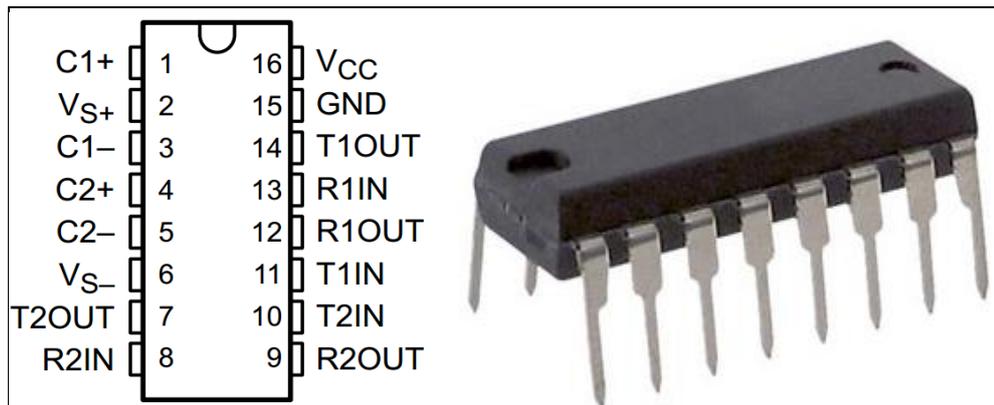


Figura 3.4. Circuito Integrado MAX232

Fuente: ROBODACTA. Max232.

<http://www.robodacta.mx/index.php?dispatch=products.view&product>. [Consulta: 27 de Octubre de 2012]

#### **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.

#### **Vs-**

Conexión de salida del voltaje de -10V.

#### **Vs+**

Conexión de salida del voltaje de +10V.

**T1<sub>in</sub>, T2<sub>in</sub>, R1<sub>out</sub>, R2<sub>out</sub>**

Conexiones a niveles de voltaje de TTL o CMOS.

**T1<sub>out</sub>, T2<sub>out</sub>, R1<sub>in</sub>, R2<sub>in</sub>**

Conexiones a niveles de voltaje del protocolo RS-232.

**VCC**

Alimentación positiva del MAX232

**GND**

Alimentación negativa del MAX232

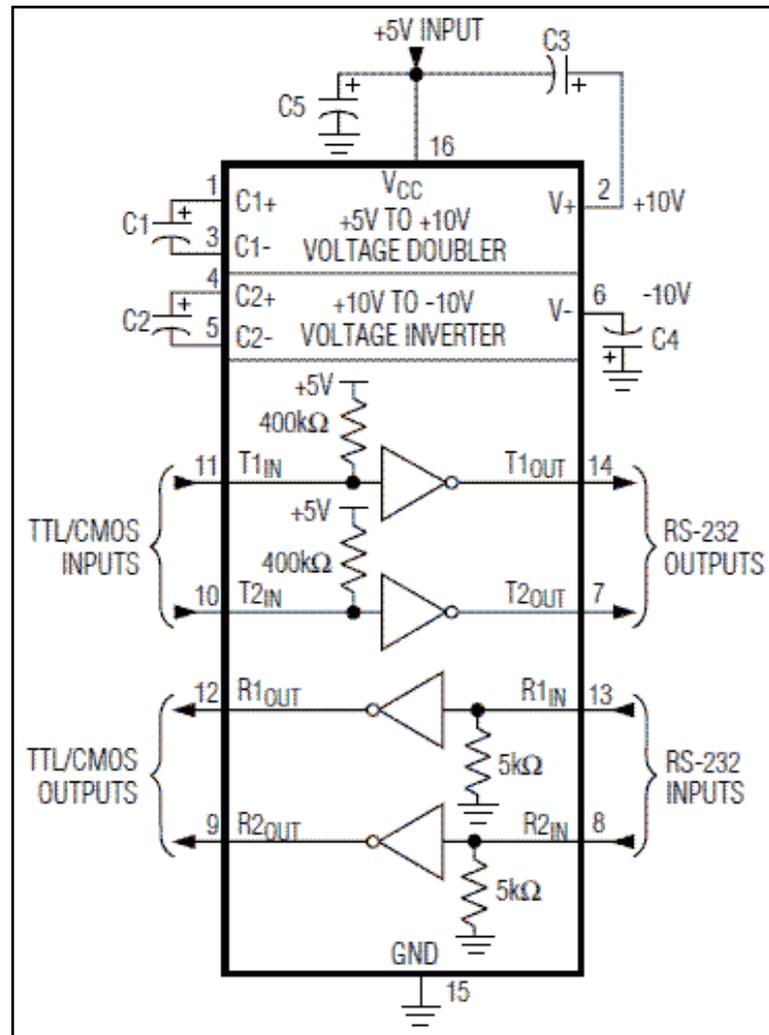


Figura 3.5. Diagrama de Conexiones del CI MAX 232

Fuente: SIONGBOON. Max232. [http://www.siongboon.com/projects/2006-03-06\\_serial\\_communication/max232.gif](http://www.siongboon.com/projects/2006-03-06_serial_communication/max232.gif). [Consulta: 27 de Octubre de 2012]

### Niveles de Voltaje

Cuando un circuito integrado MAX232 recibe un nivel TTL lo convierte, cambia un nivel lógico TTL de 0 a un nivel comprendido entre +3 y +15 V, y cambia un nivel lógico TTL 1 a un nivel comprendido entre -3 a -15 V, y viceversa, para convertir niveles de RS232 a TTL.

Tipo de línea RS232 y Nivel lógico	Voltaje RS232	Voltaje TTL hacia 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	5 V
Señales de control (RTS/CTS/DTR/DSR) Nivel lógico 1	+3 V a +15 V	0 V

Tabla 3.1. Niveles de Voltaje para el CI MAX232

### 3.2. Encendido de Modulo SIM900

Al arrancar todo el sistema tanto el modulo como el uC están apagados, el pin STATUS del SIM900 se encuentra en un nivel bajo (0V) como se indicó en el (capitulo 1 sección 1.4.1), el modulo necesita un pulso de bajo nivel para ser encendido, para este fin se conecta el uC al módulo mediante un transistor el cual generará un pulso de bajo nivel durante 3 segundos. O hasta que el pin STATUS (Capitulo 1 sección 1.4.5) del módulo SIM900 pase al estado de alto nivel (5V), esto nos indica que el modulo está listo para funcionar y ser sincronizado con el uC.

Código en Proton IDE.

Para evitar confusiones al escribir el código, se cambia los nombres de los pines del microcontrolador. *Symbol* es la sentencia utilizada en ProtonIDE para cambiar de nombres a los puertos del uC.

En este caso se cambia de nombre al pin 3 de puerto D por *pwrkey*, que representa al pin de encendido del modem.

```

Symbol pwrkey = PORTD.3
encendido:
pwrkey = 1
DelayMS 3000
pwrkey = 0
DelayMS 4000
Goto inicio

```

Subrutina para encender el módulo SIM900

Por el pin 3 del puerto D del uC, se activa un transistor 2N3904, este engancha a un voltaje de 0V al pin PWRKEY del modem Sim900 durante 3 segundos, luego pasa al estado normal y espera 4 segundos para regresar al inicio del programa con el modulo encendido. En la figura 3.1 se muestra la conexión de encendido del modem SIM900.

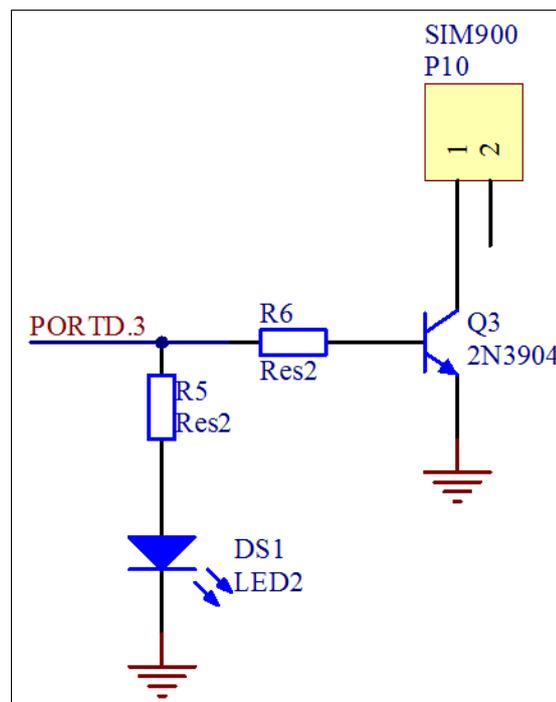


Figura 3.6. Encendido del Modem SIM900

### 3.3. Comunicación entre modem SIM900 y PIC18F4550.

El PIC18F4550 posee internamente una EUART que se puede programar para la transmisión y recepción de datos en forma serial. Este tipo de comunicación serial es la que se utiliza con la interfaz RS-232.

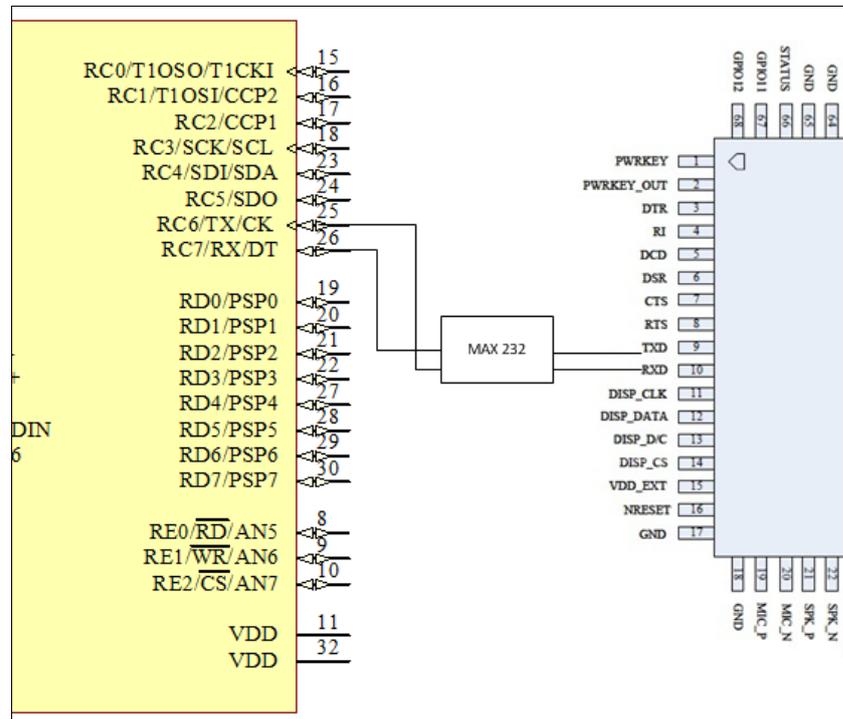


Figura 3.7. Comunicación serial entre PIC18F4550 y Modem SIM900

La programación de la interfaz serie del PIC es muy sencilla, basta con poner los valores adecuados en los registros SPBRG, TXSTA, RCSTA y BAUDCON, además hay que programar el pin RC6 como salida y el RC7 como entrada. Para ello TRISC debe ser: `TRISC= 0b10000000; // RC6 salida, RC7 entrada.`

Para generar la velocidad de transmisión existe un temporizador dedicado BRG, que se incrementa por impulsos de la señal de reloj interna Fosc. Puede funcionar en modo 8 bits (bit BRG16=0) o 16 bits (bit BRG16=1). El bit BRG16 está en el registro BAUDCON y se puede acceder a él para elegir el modo de transmisión.

El valor de cuenta se almacena en SPBRH:SPBRG. Existen un predivisor por 4, 16 o 64, que se selecciona mediante los bits BRG16 y BRGH. La velocidad en baudios se controla mediante el registro SPBRH:SPBRG y las siguientes fórmulas:

Configuración de Bits			Modo BRG/EUSART	Formula Baudios
SYNC	BRG16	BRG		
0	0	0	8-bit/Asíncrono	$F_{OSC}/[64(n+1)]$
0	0	1	8-bit/Asíncrono	$F_{OSC}/[64(n+1)]$
0	1	0	16-bit/Asíncrono	
0	1	1	16-bit/Asíncrono	$F_{OSC}/[64(n+1)]$
1	0	x	8-bit/Síncrono	
1	1	x	16-bit/Síncrono	

Tabla 3.2. Formula de Baudios

Por ejemplo, para una velocidad de 9600 baudios con receptor en modo asíncrono (SYNC=0) con alta velocidad (BRGH=1) y recarga de 8 bits solamente (BRG16=0), la fórmula para el cálculo del valor de recarga de SPBRG es:  $F_{osc}/[16(SPBRG+1)]$ . Despejando el valor de SPBRG para  $F_{osc}=4$  MHz (frecuencia del oscilador)  $SPBRG=25$ . El resumen de los registros asociados con el cálculo del valor de recarga es:

Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
<b>TXSTA</b>	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
<b>RCSTA</b>	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
<b>BAUDCON</b>	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	-	WUE	ABDEN
<b>SPBRGH</b>	Generador de velocidad de baudios. Registro Byte alto							
<b>SPBRG</b>	Generador de velocidad de baudios. Registro Byte bajo							

Tabla 3.3. Registros asociados en la comunicación serial.

Además hay que programar el funcionamiento del puerto como: asíncrono, de 8 bits, con recepción y transmisión continua. Estas características se programan sobre los registros TXSTA y RCSTA:

### Registro TXSTA

	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
TXSTA	<b>CSRC</b>	<b>TX9</b>	<b>TXEN</b>	<b>SYNC</b>	<b>SENDB</b>	<b>BRGH</b>	<b>TRMT</b>	<b>TX9D</b>
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

**Bit 7 CSRC:** Selección de la fuente de reloj

Modo asíncrono:

No importa.

Modo síncrono:

1 = Modo maestro (reloj generado internamente desde BRG)

0 = Modo esclavo (reloj de fuente externa).

**Bit 6 TX9:** Transmisión de 9 bits

1 = Selecciona 9-bits de transmisión

0 = Selecciona 8-bits de transmisión

**Bit 5 TXEN:** Habilitación de Transmisión

1 = Habilitado

0 = Deshabilitado

**Bit 4 SYNC:** Modo de EUSART

1 = Modo síncrono

0 = Modo asíncrono

**Bit 3 SENDB:** Enviar carácter de break

Modo asíncrono:

1 = Enviar carácter de break

0 = Se ha completado el envío.

**Bit 2 BRGH:** Selección de alta frecuencia.

Modo asíncrono:

1 = alta velocidad

0 = Baja velocidad

Modo síncrono:

No se utiliza en este modo.

**Bit 1 TRMT:** habilitación de la transmisión

1 = TSR vacío

0 = TSR lleno

**Bit 0 TX9D:** 9no bit de datos

### Registro RCSTA

	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

**Bit 7 SPEN:** Bit de habilitación del puerto serie

1 = activado el puerto serie (configura RX / DT y TX / CK pines como pines del puerto serie)

0 = deshabilitado puerto serie

**Bit 6 RX9:** Recepción de 9-bits

1 = Habilitada recepción de 9-bits

0 = Habilitada recepción de 8-bits

**Bit 5 SREN:** Habilita la recepción de un solo dato

Modo asíncrono:

No importa.

Modo síncrono - Master:

1 = Habilita la recepción única

0 = Se ha completado la recepción

Este bit se borra después de finalizada la recepción.

Modo síncrono - Esclavo:

No importa.

**Bit 4 CREN:** Bit de habilitación de la recepción

Modo asíncrono:

1 = Habilita

0 = Deshabilita

El modo síncrono:

1 = Habilita

0 = Deshabilita

**Bit 3 ADDEN:** Detección de la dirección

Modo asíncrono 9-bit (RX9 = 1):

1 = Habilita la detección del noveno bit a "1" solo si el dato tiene el noveno bit a "1" lo recibe.

0 = Deshabilita la detección del noveno bit, todos los bytes son recibidos y noveno bit se puede utilizar como bit de paridad.

Modo asíncrono de 8-bits (RX9 = 0):

No importa.

**Bit 2 FERR:** Error de trama

1 = error de trama

0 = No error

**Bit 1 OERR:** Error de sobre escritura

1 = error de desbordamiento

0 = Sin error de desbordamiento

**Bit 0 RX9D:** noveno bit de los datos recibidos

Los valores de estos dos registros deben ser:

TXSTA=0b00100100      Asíncrono, alta velocidad, 8 bits, transmisión habilitada.

RCSTA=0b10010000      Habilita el puerto y la recepción de 8 bits.

Una vez configurado la interfaz serie ya se pueden enviar y recibir datos de 8 bits. El envío de un byte se realiza simplemente escribiendo en el registro TXREG y la lectura se realiza leyendo el registro RCREG.

Es muy importante sincronizar correctamente estas señales, es decir, si se quieren enviar varios caracteres seguidos no se pueden escribir en TXREG a la velocidad del procesador, ya que entonces se perderían la mayoría de los datos (el procesador es mucho más rápido que la transmisión serie). Para estar seguros de que el último byte que se escribió en TXREG sea pasado al registro de desplazamiento TSR, hay que mirar el bit 4 (TXIF) del registro PIR1 para ver si el buffer de transmisión está lleno o vacío; si está lleno (valor 0) no podemos escribir un nuevo dato en TXREG, si está vacío (valor 1) si podemos. Este mismo bit sirve para que un manejador de interrupción sepa qué periférico produjo la interrupción, ya que cada vez que se vacía el buffer TXREG se produce un.

Con la recepción pasa algo parecido. Cuando se recibe datos por el puerto serie (en RCREG) se produce una interrupción y se activa el bit 5 (RCIF) del registro PIR1, indicando que hay algo en RCREG para leer.

Las interrupciones del puerto serie se activan con los bits 4 (TXIE) y 5 (RCIE) del registro PIE1 además de los bits PEIE y GIE del INTCON (habilitación de las interrupciones en general y de periféricos en particular). El resumen de los registros asociados son:

Nombre	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
<b>INTCON</b>	GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
<b>PIR1</b>	SPPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
<b>PIE1</b>	SPPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
<b>IPR1</b>	SPPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP
<b>TMR1L</b>	parte baja del registro TIMER1							
<b>TMR1H</b>	parte alta del registro TIMER1							
<b>T1CON</b>	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON

Tabla 3.4. Registros asociados con la transmisión y recepción serial

Al utilizar el software ProtonID (Capítulo 2 2.6.1. Desarrollo del programa), todos estos registros son configurados automáticamente al escribir la sentencia para la transmisión serial.

```
SerOut tx_sim900 ,16468, ["AT",13]
```

Sentencia para transmitir datos.

```
SerIn rx_sim900 ,16468, [D_Ing[x]]
```

Sentencia para recibir datos.

Con estas sentencias se transmite y recibe a una velocidad de 9600 baudios, modo asíncrono, 8 bits de datos.

### 3.3.1. Sincronización entre microcontrolador y módulo SIM900

Se cambia los nombres de los pines RC6 y RC7 a *rx\_sim900* y *tx\_sim900* respectivamente.

```
Symbol rx_sim900 = PORTC.7
Symbol tx_sim900 = PORTC.6
```

Sentencia para cambiar de nombre a los pines.

Sincronizamos el módulo sim900 con el uC, por el puerto RC6 se envía en forma serial la palabra "AT" (*atención*):

```
SerOut tx_sim900 ,16468, ["AT",13]
```

Sentencia para enviar en forma serial el comando "AT"

El módulo debe responder la palabra "OK" la cual indica que el módulo y uC están sincronizados, el uC lee los datos enviados por el módulo esperando la palabra "OK".

```

Repeat
    SerIn rx_sim900 ,16468, [D_Ing[x]]
    Inc x
    Inc z
    If z = 20 Then GoTo inicio
Until D_Ing[x-1]="K"

```

Lectura de datos desde el módulo SIM900

El uC lee todos los datos hasta que el último sea la letra "K" y en caso de que el modulo no responda, el uC volverá a enviar el comando AT de sincronización hasta recibir la palabra "OK".

Una vez ya sincronizados los dos dispositivos, el modulo está listo para recibir cualquier comando AT enviado desde el uC, por ejemplo para eliminar el ECO (capitulo 1 sección 1.7.1).

```

SerOut tx_sim900 ,16468, ["ATE0",13]

```

Comando para eliminar el ECO

### 3.3.2. Borrado de SMS del Módulo SIM900

En el módulo SIM900 se van a almacenar SMS erróneos enviados por otras personas ya sea por equivocación en el discado del número, mensajes de promociones, mensajes enviados por la misma operadora, etc.

Estos mensajes no contienen ningún tipo de comando lo que puede generar errores al momento de querer controlar cualquier tipo de sistema, ya sea iluminación, alarma o climatización. Por lo que es necesario borrar los SMS cada vez que se realice alguna acción en el módulo.

A continuación se muestra el comando enviado para borrar los mensajes de texto del módulo SIM900:

```

SerOut tx_sim900 ,16468, ["AT+CMGD=0,4",13]

```

Comando para borra todos los mensajes del módulo.

### 3.3.3. Lectura de SMS del módulo SIM900

Cada vez que el módulo sim900 reciba un mensaje de texto tendrá que ser filtrado y leído por el uC para saber si es un comando que hará actuar a algún sistema o solamente es un mensaje erróneo que tendrá que ser ignorado por el microcontrolador.

```
SerOut tx_sim900 ,16468, ["AT+CMGL=", 34, "ALL", 34, 13]
```

Comando para leer los mensajes del modem Sim900.

### 3.3.4 Filtrado del SMS

Este mensaje es leído por el uC, el cual toma toda la información del SMS como es fecha, hora, número de teléfono, etc. El uC debe filtrar el mensaje para obtener el comando enviado por el usuario.

```
Repeat

    If D_Ing[x] > 45 And D_Ing[x] < 91 Then
    Aux[Ps]=D_Ing[x]
    Inc Ps
    End If
    Inc x

Until D_Ing[x-1]="."
```

Subrutina para filtrar solo el comando enviado por el usuario.

De esta manera se lee el comando o mensaje recibido para hacer actuar los diferentes sistemas.

## Comandos para el control de cada sistema

Iluminación:

Comando	
ITA	Enciende todas la luces
IT0	Apaga todas las luces
I1A	Enciende la luz de la zona 1 al nivel Máximo
I1B	Enciende la luz de la zona 1 al nivel Alto
I1C	Enciende la luz de la zona 1 al nivel Medio
I1D	Enciende la luz de la zona 1 al nivel Bajo
I10	Apaga la luz de la zona 1
I2A	Enciende la luz de la zona 2 al nivel Máximo
I2B	Enciende la luz de la zona 2 al nivel Alto
I2C	Enciende la luz de la zona 2 al nivel Medio
I2D	Enciende la luz de la zona 2 al nivel Mínimo
I20	Apaga la luz de la zona 2
RI	Envía un reporte del sistema de Iluminación.

Tabla 3.5. Comandos para el sistema de Iluminación

Alarma:

Comando	
AAA	Activa la alarma en modo Ausente
AAP	Activa la alarma en modo Presente
ADD	Desactiva la alarma
RA	Envía un reporte del sistema de alarma

Tabla 3.6. Comandos para el sistema de Alarma

Climatización:

Comando	
T1XX	Fija la temperatura de la zona 1 al nivel de XX
T2XX	Fija la temperatura de la zona 1 al nivel de XX
RT	Envía un reporte del sistema de Climatización

Tabla 3.7. Comandos para el sistema de Climatización

XX es un valor entre 15 y 30, que significa el nivel de temperatura que se quiere dar a la habitación de una zona determinada, el nivel de temperatura viene dado en °C (grados Celsius).

Ejemplo: para fijar la temperatura de la zona 1 en 20°C el comando a enviar es T120.

### 3.4. Comunicación entre PIC16F628A y PIC18F4550

El sistema de iluminación, alarma y climatización es controlado cada uno por un uC PIC16F628A (uC16) el que hace actuar a los diferentes dispositivos de entrada y salida. El uC16 no se comunica directamente con el modem GSM, sino que todos los datos son enviados hasta el microcontrolador PIC18F4550 (uC18), que está conectado directamente con el módulo sim900.

El uC18 después de haber filtrado el SMS es capaz de identificar a que sistema debe hacer actuar, alertando solamente a uno de los tres uC16, manteniendo el funcionamiento sin interrupciones a los dos uC restantes.

El SMS filtrado es enviado de forma serial hasta el uC16, el cual identifica el tipo de orden dada, para así, hacer actuar a los periféricos requeridos.

```

Case "I"
    datoc1 = 0
    datoc2 = 1
    Select D_Ing[2]
        Select D_Ing[3]
            Case "A"
                d1tx = "1"
                d2tx = "A"
            GoSub envioia

```

Subrutina en uC18 para encender la luz de la zona 1 al máximo.

Los datos datoc1 y datoc2 son cargados con 0 y 1 respectivamente para indicar que se va a controlar el sistema de iluminación y se lee los datos D\_Ing[2] y D\_Ing[3] lo que forma el comando I1A (enciende la luz de la zona 1 al máximo).

### 3.4.1 Comunicación serial entre PIC16F628A y PIC18F4550

El uC18 filtra el mensaje y obtiene por ejemplo: "I1A" el cual tiene que ser transmitido hasta el uC16.

El comando leído por el uC18 se guarda en las variables d1\_tx y d2\_tx para luego ser transmitidos hacia el uC16.

```

Select D_Ing[1]
Case "I"
    datoc1 = 0 ; datoc2 = 1
    Select D_Ing[2]
        Case "1"
            Select D_Ing[3]
                Case "A"
                    d1tx = "1"
                    d2tx = "A"
                GoSub envioia

```

Subrutina para seleccionar el comando ingresado en el PIC18F4550 (comando ingresado: I1A).

La comunicación serial entre los dos microcontroladores (uc16 y uC18) es igual a la comunicación entre el uC18 y el módulo SIM900.

```

envioia:
ctrl1 = datoc1
ctrl2 = datoc2
DelayMS 50
SerOut tx_todos, 16468, [d1tx]
DelayMS 20
SerOut tx_todos, 16468, [d2tx]
DelayMS 20
ctrl1=0
ctrl2=0
Return

```

Subrutina en uC18 para enviar datos de iluminación.

El uC18 transmite la información a todos los uC16 pero solo uno de ellos es el que lee el mensaje dependiendo del valor cargado en ctrl1 y ctrl2 que son pines de alerta para cada uC16.

El uC16 recibe la información y según el comando ingresado hará actuar a los periféricos de cada sistema. A continuación se muestra un ejemplo para el sistema de iluminación:

```

SerIn rx,16468,[D_Ing1]
SerIn rx,16468,[D_Ing2]
DelayMS 20
Select D_Ing1
    Case "1"
        Select D_Ing2
            Case "A"
                duty1=a
            Case "B"
                duty1=b
        End Select
GoTo inicio

```

Subrutina para leer el dato enviado por el PIC18F4550.

En esta subrutina se leen los datos D\_Ing1 y D\_Ing2 para identificar a que zona pertenece y que cantidad de iluminación se desea programar.

### 3.4.2. Comunicación para el sistema de iluminación.

Los comandos que deben ser enviados por el usuario para comandar el sistema de iluminación son los mostrados en la Tabla 3.5. Este comando es leído por el uC18 de la siguiente manera:

```

Select D_Ing[1]
Case "I"
    datoc1 = 0
    datoc2 = 1
    Select D_Ing[2]
    Case "1"
        Select D_Ing[3]
        Case "A"
            d1tx = "1"
            d2tx = "A"
            GoSub envioia
        Case Else
            GoSub com_no_reg
        End Select
    Case "T"
        Select D_Ing[3]
        Case "A"
            d1tx = "T"
            d2tx = "A"
            GoSub envioia
        Case "0"
            d1tx = "T"
            d2tx = "0"
            GoSub envioia
        Case Else
            GoSub com_no_reg
        End Select
    End Select
End Select

```

Subrutina para la lectura de comandos del sistema de Iluminación.

En la variable `D_Ing[1]` se almacena la letra correspondiente al sistema a controlar, para iluminación la letra "I". En `D_Ing[2]` se almacena el número de zona a comandar, número "1" para la Zona 1 y "2" para la Zona 2. Por último en la variable `D_Ing[3]` se almacena el valor para modificar la iluminación en la zona determinada, "A" Máximo, "B" Alto, "C" Medio, "D" Mínimo y "0" Apagado.

El comando leído por el uC18 puede ser cualquiera de los mostrados en la Tabla 3.5. Tomaremos como ejemplo el comando I1A.

Los datos leídos son guardados en las variables `d1tx` y `d2tx` (dato 1 y dato 2 a transmitir) para luego ser transmitidos hacia el uC16 que controla la iluminación.

El uC18 transmite solo dos datos (número de zona y cantidad de iluminación) ya que las variables `datoc1` y `datoc2` son cargadas a los pines `ctrl1` y `ctrl2` que son los que alertan al uC16 para que entre en modo de recepción de datos.

```

Symbol ctrl1 = PORTA.3
Symbol ctrl2 = PORTA.4

envioia:

ctrl1 = datoc1
ctrl2 = datoc2

DelayMS 50

SerOut tx_todos, 16468, [d1tx]
DelayMS 20
SerOut tx_todos, 16468, [d2tx]
DelayMS 20

ctrl1=0
ctrl2=0
Return

```

Subrutina de envío de datos hacia el PIC16F628A

El uC16 recibe toda esa información de la siguiente manera:

```

SerIn rx,16468,[D_Ing1]
SerIn rx,16468,[D_Ing2]
DelayMS 20
Select D_Ing1
    Case "1"
        Select D_Ing2
            Case "A"
                duty1=a
            Case "B"
                duty1=b
        End Select
    GoTo inicio
    Case "T"
        Select D_Ing2
            Case "A"
                duty1=a
                duty2=a
                duty3=a
                duty4=a
            End Select
        Case "R"
            Select duty1
                Case a
                    d1tx = "A"
            End Select
            SerOut tx, 16468, [d1tx]
            DelayMS 20
            SerOut tx, 16468, [d2tx]
            DelayMS 20
        Case Else
            GoTo inicio
    End Select
GoTo inicio

```

Lectura del dato enviado por el PIC 18F4550

El uC16 lee el dato correspondiente al número de zona y cantidad de iluminación, en caso de ser un reporte (comando "RA"), el PIC transmitirá en forma serial los datos de iluminación de cada zona.

### 3.4.3. Comunicación para el sistema de alarma.

El comando es leído por el uC18 de la siguiente manera:

```

Case "A"
    datoc1=1
    datoc2=0
    Select D_Ing[2]
    Case "A"
        Select D_Ing[3]
        Case "A"
            d1tx = "A"
            d2tx = "A"
            GoSub envioia

        Case Else
            GoSub com_no_reg
        End Select
    Case "D"
        If D_Ing[3] = "D" Then
            d1tx = "D"
            d2tx = "D"
            GoSub envioia
        Else
            GoSub com_no_reg
        End If
    Case Else
        GoSub com_no_reg
    End Select

```

Subrutina para la lectura de comandos del sistema de Alarma.

En la variable D\_Ing[1] se almacena la letra correspondiente al sistema a controlar, para alarma la letra "A". En D\_Ing[2] se almacena el dato de control para la alarma, "A" para activar y "D" para desactivar. En la variable D\_Ing[3] se almacena el tipo de activación, "A" activación en modo ausente (Activa todas las zonas) y "P" activación en modo presente (desactiva las zonas interiores). En caso de que sea el comando para desactivar la alarma, el tercer dato no tiene importancia.

Igual que en el sistema de iluminación, las variables datoc1 y datoc2 son las que alertan al uC16 para que entre en modo de recepción de datos. Tomaremos como ejemplo el comando "AAA" (activación de alarma en modo ausente).

```

envioia:

ctrl1 = datoc1
ctrl2 = datoc2

DelayMS 50

SerOut tx_todos, 16468, [d1tx]
DelayMS 20
SerOut tx_todos, 16468, [d2tx]
DelayMS 20

ctrl1=0
ctrl2=0
Return

```

Envío de datos en forma serial para el PIC 16F628A

El PIC 16F628A recibe todos los datos enviados por el PIC 18F4550 de la siguiente manera:

```

Repeat
Until ctrl1 = 1 And ctrl2 =0
SerIn rx,16468,[D_Ing1]
SerIn rx,16468,[D_Ing2]
DelayMS 20

Select D_Ing1
  Case "A"
    Select D_Ing2
      Case "A"
        GoTo act_aus

      Case "P"
        GoTo act_pre
    End Select
    GoTo inicio
  Case "D"
    If D_Ing2 = "D" Then GoTo desact
    Case "R"
      DelayMS 40
      SerOut tx,16468,[d1tx]
      DelayMS 20
      SerOut tx,16468,[d1tx]
      DelayMS 20
    End Select

```

Lectura de datos enviado por el PIC 18F4550

El uC16 simplemente lee los datos recibidos en D\_Ing1 y D\_Ing2 para activar, desactivar o enviar un reporte de alarma. En caso de ser reporte de alarma el uC16 envía al uC18 todos los datos del sistema de alarma.

### 3.4.4. Comunicación para Sistema de Climatización

En el sistema de climatización se lee las variables D\_Ing[x] (x=1, 2, 3, 4), en el D\_Ing[1] indica el sistema a controlar, "T" para climatización. La variable D\_Ing[2] contiene el número de zona o habitación a controlar, "1" para la zona 1 y "2" para la zona 2. En los valores restantes D\_Ing[3] y D\_Ing[4] se encuentra el valor de temperatura comprendido entre 15 y 30°C, este cálculo se hace en la subrutina *valor\_temp*, la que simplemente toma los dos valores y los transforma a un número decimal.

```

Case "T"
    datoc1=1
    datoc2=1
    Select D_Ing[2]
    Case "1"
        GoSub valor_temp
        If temp_t > 30 Then temp_t = 30
        If temp_t < 15 Then temp_t = 15
        d2tx = "S"
        d1tx = temp_t * 3
        GoSub envioia
            Case "2"
                GoSub valor_temp
                If temp_t > 30 Then temp_t = 30
                If temp_t < 15 Then temp_t = 15
                d2tx = "D"
                d1tx = temp_t * 3
                GoSub envioia
            Case Else
    End Select

```

Subrutina para la lectura de comandos del sistema de Climatización.

Los datos son cargados en las variables correspondientes y son enviadas al uC16 para hacer actuar al sistema de climatización.

```

envioia:
ctrl1 = datoc1
ctrl2 = datoc2
DelayMS 50
SerOut tx_todos, 16468, [d1tx]
DelayMS 20
SerOut tx_todos, 16468, [d2tx]
DelayMS 20
ctrl1=0
ctrl2=0
Return
enviori:
ctrl1 = datoc1
ctrl2 = datoc2

```

Envío de datos de climatización hacia el PIC 16F628A

El PIC 16F628A que controla el sistema de climatización recibe los datos de la siguiente manera:

```

dat_ing:

DelayMS 20
SerIn rx,16468,[D_Ing1]
SerIn rx,16468,[D_Ing2]
DelayMS 20

Select D_Ing2
Case "S"
Tsets = D_Ing1 / 3
Tmaxs = Tsets + 1
Tmins = Tsets - 1

```

Recepción de datos de climatización desde el PIC 18F4550

El uC16 lee los datos y configura la temperatura con los valores establecidos. En caso de ser un reporte el uC16 envía en forma serial todos los datos correspondientes a la temperatura de cada zona.

### 3.4.5. Comunicación para Reportes de Iluminación, Alarma o Climatización.

Cuando el usuario pide al Sistema de Control algún tipo de reporte, ya sea de iluminación, alarma o climatización, envía los datos correspondientes hacia el uC16 (RA, RI o RT), pero en este caso el uC18 espera a que el uC16 responda con los datos solicitados para generar un mensaje de texto que será enviado al usuario. En la siguiente subrutina se muestran los datos enviados por el sistema de iluminación para generar un SMS de reporte.

```

Case "R"
  Select D_Ing[2]
  Case "I"
    d1tx = "R"
    d2tx = "R"
    GoSub enviiori

    Select d1rx
      Case "A"
        Aux1 = "Maximo"
        Aux1 = "Apagado"
      Case Else
        Aux1 = "Error"
    End Select

Cls
Print At 1,1, "Enviando REPORTE"
Print At 2,1 , Num
SerOut tx_sim900,16468, ["AT+CMGS=", 34, "+", Num, 34, 13]
DelayMS 1000
SerOut tx_sim900,16468, ["Iluminacion", 13, "Zona1:
", Aux1, 13, "Zona2: ", Aux2, 26, 13]

  Repeat
    x=0
  Inc x

```

```
Until D_Ing[x-1]="K"  
Cls  
Print At 1,1,"Mensaje ENVIADO "  
DelayMS 2000  
Cls  
GoSub Borrando
```

Subrutina para enviar reporte de Iluminación.

En el siguiente capítulo se explicará más a fondo las subrutinas mostradas en lo que se refiere a envío de SMS de reporte, iluminación de cada zona, control de climatización, activación/desactivación de alarma y todo lo que no se refiere a la comunicación serial entre los PIC's 16F628A, 18F4550 y Modem SIM900

## CAPÍTULO 4

### PERIFÉRICOS DE ENTRADA Y SALIDA

Un periférico es un dispositivo físico que se conecta o acopla al circuito, pero no forma parte del núcleo básico (microcontrolador, placa principal, alimentación eléctrica) del mismo.

Los periféricos sirven para comunicar el circuito con el exterior (pulsadores, teclados matriciales, LDC matriciales, LCD gráficos, actuadores electrónicos). En general, estos pueden conectarse o desconectarse del sistema, pero el mismo seguirá funcionando pero con menos capacidades.

Los periféricos son parte del hardware del sistema, pero no todo hardware es periférico (por ejemplo: microcontrolador, placa principal, son hardware pero no periféricos). Los periféricos forman parte de los accesorios o complementos del circuito. El término “accesorio del circuito” incluye a los periféricos, pero también a otros componentes como chips, placas principales, sensores, etc. Por lo tanto la palabra accesorios es un término más amplio de los periféricos.

Para la conexión de periféricos y funcionamiento óptimo de todo el circuito es necesario utilizar componentes electrónicos.

#### **Componentes Electrónicos**

Hay que diferenciar entre componentes y elementos. Los componentes son dispositivos físicos, mientras que los elementos son modelos o abstracciones idealizadas que constituyen la base para el estudio teórico de los mencionados componentes. Así, los componentes aparecen en un listado de dispositivos que forman un circuito, mientras que los elementos aparecen en los desarrollos matemáticos de la teoría de circuitos.

Entonces un componente electrónico es un dispositivo que forma parte de un circuito.

Se suele encapsular, generalmente en un material cerámico, metálico o plástico, y terminar en dos o más terminales o patillas metálicas. Se diseñan para ser

conectados entre ellos (normalmente mediante soldadura), a un circuito impreso, para formar el mencionado circuito.

Estos componentes pueden ser: discretos, integrados, semiconductores, activos, pasivos, electromagnéticos, etc.

#### 4.1 Componentes Electrónicos Activos

Los componentes activos son aquellos capaces de excitar los circuitos, realizar ganancias. Fundamentalmente son los generadores eléctricos y algunos componentes semiconductores. Estos últimos, en general, tienen un comportamiento no lineal, esto es, la relación entre la tensión aplicada y la corriente demandada, no es lineal. Los componentes activos semiconductores derivan del diodo de Fleming y del tríodo de Lee de Forest. En una primera generación aparecieron las válvulas que permitieron el desarrollo de aparatos electrónicos como la radio o la televisión. Posteriormente, en una segunda generación, aparecerían los semiconductores que más tarde darían paso a los circuitos integrados (tercera generación), cuya máxima expresión se encuentra en los circuitos programables (microprocesador y microcontrolador) que pueden ser considerados como componentes, aunque en realidad sean circuitos que llevan integrados millones de componentes.

En la actualidad existe un número elevado de componentes activos, siendo usual, que un sistema electrónico se diseñe a partir de uno o varios de estos componentes cuyas características lo condicionará el funcionamiento del circuito. Esto no sucede con los componentes pasivos. En la siguiente tabla se muestran los principales componentes activos utilizados para la construcción del Sistema de Control mediante SMS.

Componente	Función más común
Microcontrolador	Control de sistemas digitales.
Diodo	Rectificación de señales, regulación, multiplicador de tensión.
Batería	Generación de energía eléctrica.
Transistor	Amplificación, conmutación.

Tabla 4.1. Componentes electrónicos activos.

### 4.1.1. Baterías

Una pila o batería recargable (también llamada acumulador), es un grupo de una o más celdas electroquímicas secundarias.

Una celda electroquímica, es un dispositivo capaz de obtener energía eléctrica a partir de reacciones químicas, o bien, de producir reacciones químicas a través de la introducción de energía eléctrica. Un ejemplo común de celda electroquímica es la pila estándar de 1,5 voltios. En realidad, una pila es una celda galvánica simple, mientras una batería consta de varias celdas conectadas en serie.

#### Tipos de celdas electroquímicas

Hay dos tipos fundamentales de celdas y en ambas tiene lugar una reacción redox, y la conversión o transformación de un tipo de energía en otra.

- La celda voltaica, transforma una reacción química espontánea en una corriente eléctrica, como las pilas y baterías. También reciben los nombres de celda galvánica, pila galvánica o pila voltaica (figura 4.1).



Figura 4.1. Batería recargable de 12V 4Ah

- La celda electrolítica, transforma una corriente eléctrica en una reacción química de oxidación-reducción que tiene lugar de modo espontáneo. En muchas de estas reacciones se descompone una sustancia química por lo que dicho proceso recibe el nombre de electrolisis.

También reciben los nombres de celda electrolítica o cuba electrolítica. A diferencia de la celda voltaica, en la célula electrolítica, los dos electrodos no necesitan estar separados, por lo que hay un sólo recipiente en el que tienen lugar las dos semirreacciones.

#### 4.1.2. 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. Este término generalmente se usa para referirse al diodo semiconductor, el más común en la actualidad consta de una pieza de cristal semiconductor conectada a dos terminales eléctricos. Figura 4.2.

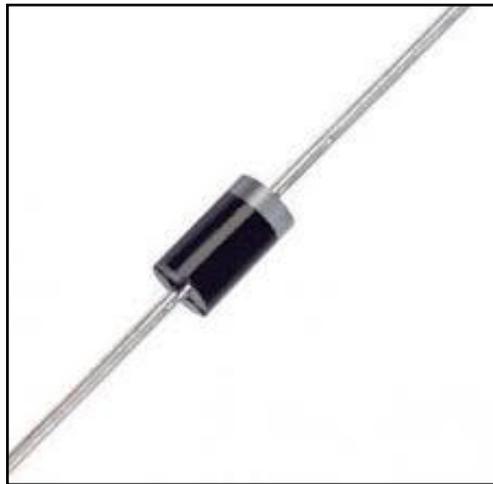


Figura 4.2. Diodo 1N4004

Fuente: DIYAUDIO. Diodo 1N4004. <http://diyaudio.es/es/diodos/929-diodo-1n4002.html>. [Consulta: 18 de Julio de 2012]

De forma simplificada, la curva característica de un diodo (I-V) consta de dos regiones: por debajo de cierta diferencia de potencial, se comporta como un circuito abierto (no conduce), y por encima de ella como un circuito cerrado con una resistencia eléctrica muy pequeña. Debido a este comportamiento, se les suele denominar rectificadores, ya que son dispositivos capaces de suprimir la parte negativa de cualquier señal, como paso inicial para convertir una corriente alterna en corriente continua. Su principio de funcionamiento está basado en los experimentos de Lee De Forest.

## Diodo Semiconductor

Un diodo semiconductor moderno está hecho de cristal semiconductor como el silicio, con impurezas en él para crear una región que contiene portadores de carga negativos (electrones), llamado semiconductor de tipo n, y una región en el otro lado que contiene portadores de carga positiva (huecos), llamado semiconductor tipo p. Las terminales del diodo se unen a cada región. El límite dentro del cristal de estas dos regiones, llamado una unión PN, es donde la importancia del diodo toma su lugar.

Cuando se somete al diodo a una diferencia de tensión externa,  $V$ , por medio de una batería se dice que el diodo está polarizado, pudiendo ser la polarización directa o inversa. En los circuitos el diodo se representa por el símbolo dado por la Figura 4.3. Donde al extremo  $p$  se le denomina ánodo (A), y al  $n$  cátodo (C).



Figura 4.3. Símbolo del Diodo

## Polarización directa

En este caso, la fuente de alimentación disminuye la barrera de potencial de la zona de carga, permitiendo el paso de la corriente de electrones a través de la unión; es decir, el diodo polarizado directamente conduce la electricidad.

Para que un diodo esté polarizado directamente, tenemos que conectar el polo positivo de la batería al ánodo del diodo y el polo negativo al cátodo, como se observa en la Figura 4.4.

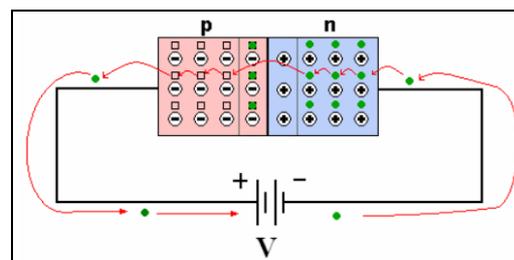


Figura 4.4. Polarización directa de un diodo.

Fuente: ECURED. Diodos. <http://www.ecured.cu/index.php/Transistor>. [Consulta: 20 de Julio de 2012]

En estas condiciones podemos observar que:

El polo negativo de la batería repele los electrones libres del cristal  $n$ , con lo que estos electrones se dirigen hacia la unión  $p-n$ , el polo positivo de la batería atrae a los electrones de valencia del cristal  $p$ , esto es equivalente a decir que empuja a los huecos hacia la unión  $p-n$ .

Cuando la diferencia de potencial entre los bornes de la batería es mayor que la diferencia de potencial en la zona de carga, los electrones libres del cristal  $n$ , adquieren la energía suficiente para saltar a los huecos del cristal  $p$ , los cuales previamente se han desplazado hacia la unión  $p-n$ .

Una vez que un electrón libre de la zona  $n$  salta a la zona  $p$  atravesando la zona de carga, cae en uno de los múltiples huecos de la zona  $p$  convirtiéndose en electrón de valencia. Una vez ocurrido esto el electrón es atraído por el polo positivo de la batería y se desplaza de átomo en átomo hasta llegar al final del cristal  $p$ , desde el cual se introduce en el hilo conductor y llega hasta la batería. De este modo, con la batería cediendo electrones libres a la zona  $n$  y atrayendo electrones de valencia de la zona  $p$ , aparece a través del diodo una corriente eléctrica constante hasta el final.

### Polarización inversa

En este caso, el polo negativo de la fuente se conecta a la zona  $p$  y el polo positivo a la zona  $n$ , como se muestra en la Figura 4.5, lo que hace aumentar la zona de transición,  $W$ , y la tensión en dicha zona hasta que se alcanza el valor de la tensión de la fuente de alimentación, tal y como se explica a continuación:

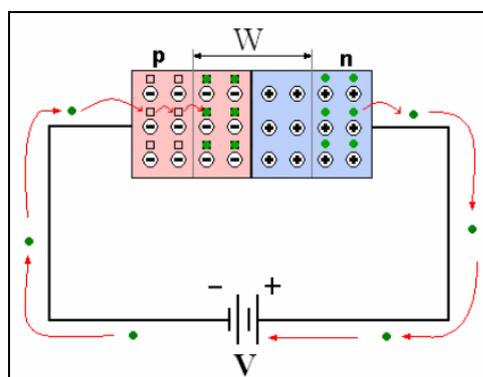


Figura 4.5. Polarización inversa de un diodo.

Fuente: ECURED. Diodos. <http://www.ecured.cu/index.php/Transistor>. [Consulta: 20 de Julio de 2012]

El polo positivo de la batería atrae a los electrones libres de la zona  $n$ , los cuales salen del cristal  $n$  y se introducen en el conductor dentro del cual se desplazan hasta llegar a la fuente. A medida que los electrones libres abandonan la zona  $n$ , los átomos pentavalentes que antes eran neutros, al verse desprendidos de su electrón en el orbital de conducción, adquieren estabilidad convirtiéndose en iones positivos.

El polo negativo de la batería cede electrones libres a los átomos de la zona  $p$ . Estos átomos sólo tienen 3 electrones de valencia, con lo que una vez que han formado los enlaces covalentes con los átomos de silicio, tienen solamente 7 electrones de valencia, siendo el electrón que falta el denominado hueco. El caso es que cuando los electrones libres cedidos por la batería entran en la zona  $p$ , caen dentro de estos huecos con lo que los átomos trivalentes adquieren estabilidad, convirtiéndose así en iones negativos.

Este proceso se repite una y otra vez hasta que la zona de carga espacial adquiere el mismo potencial eléctrico que la batería.

En esta situación, el diodo no debería conducir la corriente; sin embargo, debido al efecto de la temperatura se formarán, por roturas de enlaces, pares electrón huecos a ambos lados de la unión produciendo una pequeña corriente (del orden de  $1 \mu\text{A}$ ) denominada corriente inversa de saturación. Además, existe también una llamada corriente superficial de fugas la cual, conduce una pequeña corriente por la superficie del diodo, ya que en la superficie los átomos de silicio no están rodeados de suficientes átomos para realizar los cuatro enlaces covalentes necesarios para obtener estabilidad. Esto hace que los átomos de la superficie del diodo, tanto de la zona  $n$  como de la  $p$ , tengan huecos en su orbital de valencia con lo que los electrones circulan sin dificultad a través de ellos.

No obstante, al igual que la corriente inversa de saturación, la corriente superficial de fugas es despreciable.

#### **4.1.3. Transistor**

El transistor, inventado en 1951, es el componente electrónico estrella, pues inició una auténtica revolución en la electrónica que ha superado cualquier previsión inicial.

Con el transistor vino la miniaturización de los componentes y se llegó al descubrimiento de los circuitos integrados, en los que se colocan, en pocos

milímetros cuadrados, miles de transistores. Estos circuitos constituyen el origen de los microprocesadores y, por lo tanto, de los ordenadores actuales.

Por otra parte, la sustitución en los montajes electrónicos de las clásicas y antiguas válvulas de vacío por los transistores, reduce al máximo las pérdidas de calor de los equipos.

Un transistor es un componente que tiene, básicamente, dos funciones:

- Deja pasar o corta señales eléctricas a partir de una pequeña señal de mando.
- Funciona como un elemento amplificador de señales.

Hay dos tipos básicos de transistor:

- Transistor bipolar o BJT (Bipolar Junction Transistor)
- Transistor de efecto de campo, FET (Field Effect Transistor) o unipolar

En el presente capítulo se hablará solamente del transistor bipolar que es el utilizado en la construcción del proyecto.

### Transistor Bipolar

El transistor bipolar es un dispositivo de tres terminales -emisor, colector y base-, que, atendiendo a su fabricación, puede ser de dos tipos: NPN y PNP. En la figura 4.6 se encuentran los símbolos de circuito y nomenclatura de sus terminales. La forma de distinguir un transistor de tipo NPN de un PNP es observando la flecha del terminal de emisor. En un NPN esta flecha apunta hacia fuera del transistor; en un PNP la flecha apunta hacia dentro. Además, en funcionamiento normal, dicha flecha indica el sentido de la corriente que circula por el emisor del transistor.

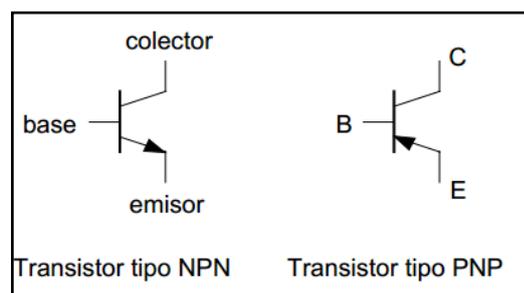


Figura 4.6. Símbolos del transistor

En general se definen una serie de voltajes y corrientes en el transistor, como las que aparecen en las figuras 4.7 y 4.8.

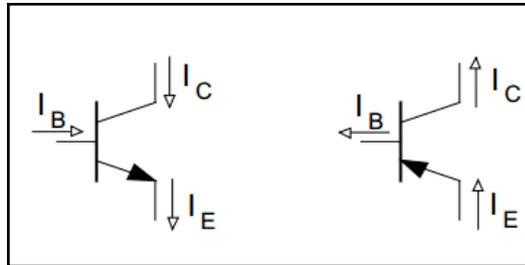


Figura 4.7. Corrientes en el transistor

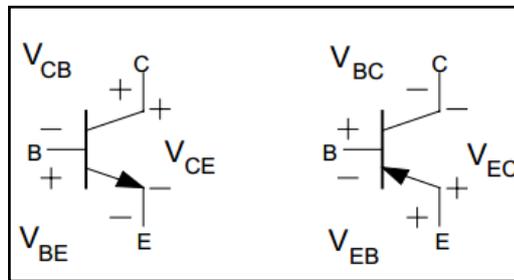


Figura 4.8. Voltajes en el transistor

### Estructura física

El transistor bipolar es un dispositivo formado por tres regiones semiconductoras, entre las cuales se forman unas uniones (uniones PN). En la figura 4.9 observamos el aspecto útil para análisis de un transistor bipolar. Siempre se ha de cumplir que el dopaje de las regiones sea alterno, es decir, si el emisor es tipo P, entonces la base será tipo N y el colector tipo P. Esta estructura da lugar a un transistor bipolar tipo PNP. Si el emisor es tipo N, entonces la base será P y el colector N, dando lugar a un transistor bipolar tipo NPN.

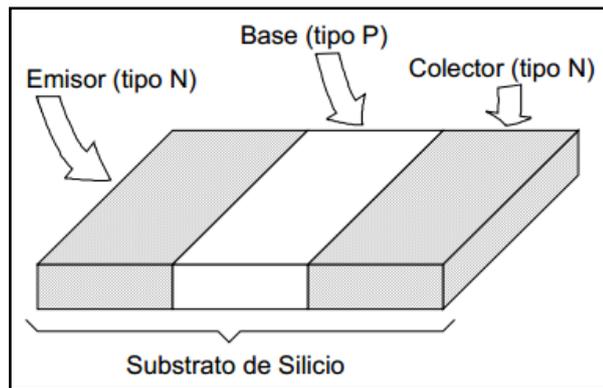


Figura 4.9. Estructura de un transistor bipolar

Fuente: UVIGO. Transistor Bipolar. <http://webs.uvigo.es/mdgomez/DEI/Guias/tema5.pdf>. [Consulta: 20 de Julio de 2012]

El transistor se fabrica sobre un sustrato de silicio, en el cual se difunden impurezas, de forma que se obtengan las tres regiones antes mencionadas. En la figura 4.10 vemos el aspecto típico de un transistor bipolar real, de los que se encuentran en cualquier circuito integrado. Sobre una base  $n$  (sustrato que actúa como colector), se difunden regiones  $p$  y  $n+$ , en las que se ponen los contactos de emisor y base.

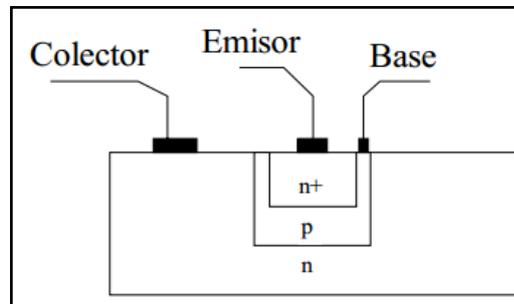


Figura 4.10. Estructura real de un transistor

Fuente: UVIGO. Transistor Bipolar. <http://webs.uvigo.es/mdgomez/DEI/Guias/tema5.pdf>. [Consulta: 20 de Julio de 2012]

Las dimensiones reales del dispositivo son muy importantes para el correcto funcionamiento del mismo.

En la figura 4.11, se pretende dar una idea de las relaciones de tamaño que deben existir entre las tres regiones para que el dispositivo cumpla su misión.

- El emisor es una región muy dopada (de ahí la indicación  $p+$ ). Cuanto más dopaje tenga el emisor, mayor cantidad de portadores podrá aportar a la corriente.

- La base ha de ser muy estrecha y poco dopada, para que tenga lugar poca recombinación en la misma, y prácticamente toda la corriente que proviene de emisor pase a colector. Además, si la base no es estrecha, el dispositivo puede no comportarse como un transistor, y trabajar como si fueran dos diodos en oposición.
- El colector ha de ser una zona menos dopada que el emisor. Las características de esta región tienen que ver con la recombinación de los portadores que provienen del emisor.

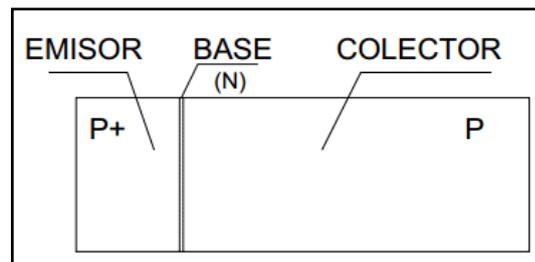


Figura 4.11. Dimensiones de un transistor

Fuente UVIGO. Transistor Bipolar . <http://webs.uvigo.es/mdgomez/DEI/Guias/tema5.pdf>. [Consulta: 20 de Julio de 2012]

Por último, en la figura 4.12 vemos el resto de componentes de un transistor bipolar, que son los contactos metálicos y los terminales (el transistor es un dispositivo de 3 terminales).

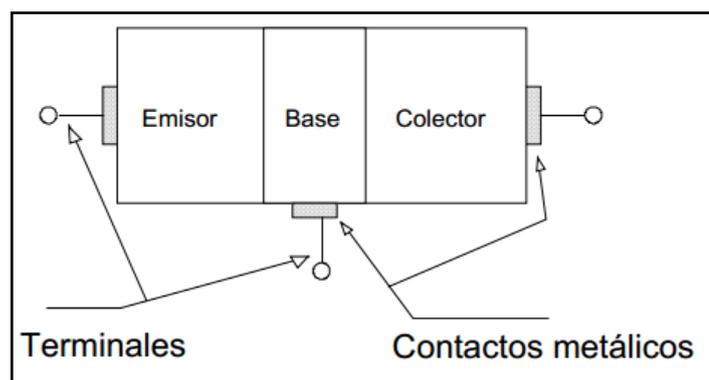


Figura 4.12. Terminales del transistor

Fuente: UVIGO. Transistor Bipolar. <http://webs.uvigo.es/mdgomez/DEI/Guias/tema5.pdf>. [Consulta: 20 de Julio de 2012]

## Funcionamiento del transistor

El transistor bipolar es un dispositivo de tres terminales gracias al cual es posible controlar una gran potencia a partir de una pequeña. En la figura 4.13 se puede ver un ejemplo cualitativo del funcionamiento del mismo. Entre los terminales de colector (C) y emisor (E) se aplica la potencia a regular, y en el terminal de base (B) se aplica la señal de control de potencia. Con pequeñas variaciones de corriente a través del terminal de base, se consiguen grandes variaciones a través de los terminales de colector y emisor. Si se coloca una resistencia se puede convertir esta variación de corriente en variaciones de tensión según sea necesario.

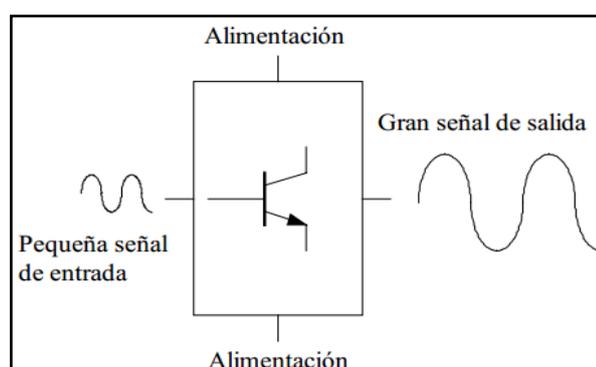


Figura 4.13. Ejemplo de funcionamiento del transistor.

Fuente: UVIGO. Transistor Bipolar. <http://webs.uvigo.es/mdgomez/DEI/Guias/tema5.pdf>. [Consulta: 20 de Julio de 2012]

## Fundamentos físicos del efecto transistor

El transistor bipolar basa su funcionamiento en el control de la corriente que circula entre el emisor y el colector del mismo, mediante la corriente de base. Entonces un transistor se puede considerar como un diodo en directa (unión emisor-base) por el que circula una corriente elevada, y un diodo en inversa (unión base-colector), por el que, en principio, no debería circular corriente, pero que actúa como una estructura que recoge gran parte de la corriente que circula por emisor base.

En la figura 4.14 se puede ver lo que sucede. Se dispone de dos diodos, uno polarizado en directa (diodo A) y otro en inversa (diodo B). Mientras que la corriente por A es elevada ( $I_A$ ), la corriente por B es muy pequeña ( $I_B$ ). Si se unen ambos diodos, y se consigue que la zona de unión (lo que llamaremos base del transistor) sea muy estrecha, entonces toda esa corriente que circulaba por A ( $I_A$ ), va a quedar absorbida por el campo existente en el diodo B. De esta forma entre el

emisor y el colector circula una gran corriente, mientras que por la base una corriente muy pequeña. El control se produce mediante este terminal de base porque, si se corta la corriente por la base ya no existe polarización de un diodo en inversa y otro en directa, y por tanto no circula corriente.

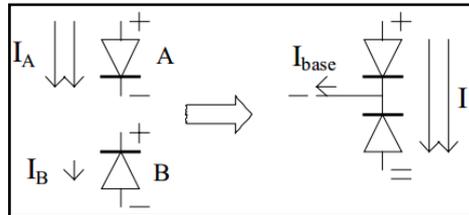


Figura 4.14. Efecto Transistor.

Fuente: UVIGO. Transistor Bipolar. <http://webs.uvigo.es/mdgomez/DEI/Guias/tema5.pdf>. [Consulta: 20 de Julio de 2012]

### Corrientes y tensiones

Para el análisis de las distintas corrientes que aparecen en un transistor vamos a considerar un transistor tipo PNP, que polarizamos tal y como aparece en la figura 4.15.

Este tipo de polarización será el usado cuando el transistor trabaje en región activa. La unión emisor-base queda polarizada como una unión en directa, y la unión colector-base como una unión en inversa.

En la figura 4.16 se muestran las principales corrientes (de electrones y huecos) que aparecen en el transistor tras aplicar la polarización indicada en la figura 4.15. Se puede observar lo siguiente:

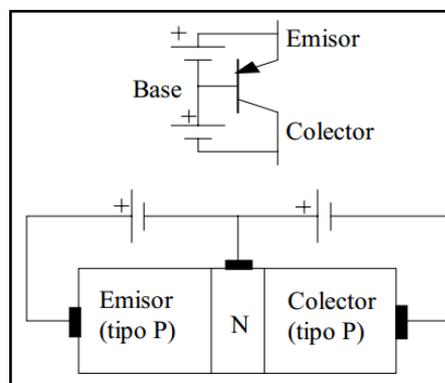


Figura 4.15. Polarización

Fuente: UVIGO. Transistor Bipolar. <http://webs.uvigo.es/mdgomez/DEI/Guias/tema5.pdf>. [Consulta: 20 de Julio de 2012]

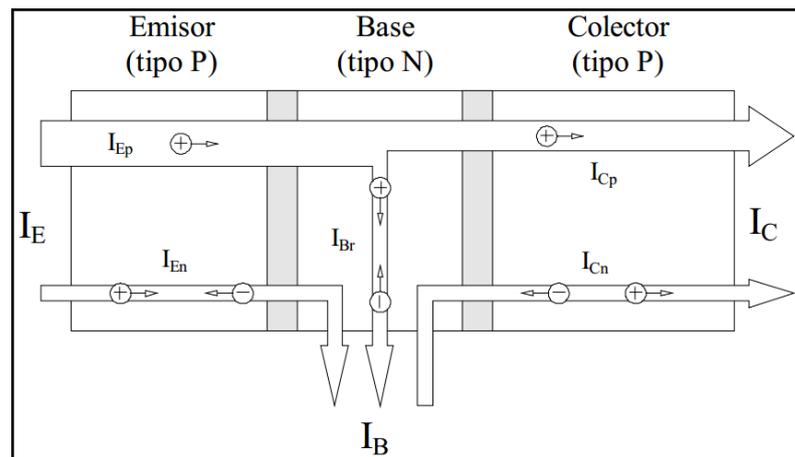


Figura 4.16. Corrientes en un transistor

Fuente: UVIGO. Transistor Bipolar. <http://webs.uvigo.es/mdgomez/DEI/Guias/tema5.pdf>. [Consulta: 20 de Julio de 2012]

Entre el emisor y la base aparece una corriente ( $I_{Ep} + I_{En}$ ) debido a que la unión está en directa. El efecto transistor provoca que la mayor parte de la corriente anterior no circule por la base, sino que siga hacia el emisor ( $I_{Cp}$ ). Entre el colector y la base circula una corriente mínima por estar polarizada en inversa ( $I_{Cn}$  más una parte mínima de  $I_{Cp}$ ).

Por la base realmente circula una pequeña corriente del emisor, más otra de colector, más la corriente de recombinación de base ( $I_{En} + I_{Cn} + I_{Br}$ )

A partir de lo anterior podemos obtener algunas ecuaciones básicas como son las siguientes:

$$I_E + I_B + I_C = 0 \quad (\text{ec4.1})$$

Esta ecuación viene impuesta por la propia estructura del circuito, es decir, el transistor es un nodo con tres entradas o salidas, por tanto la suma de las corrientes que entran o salen al mismo son igual a cero.

Cada una de las corrientes del transistor se puede poner en función de sus componentes de la siguiente forma:

$$I_E = I_{En} + I_{Ep} \quad (\text{ec4.2})$$

$$I_C = I_{Cn} + I_{Cp} \quad (\text{ec4.3})$$

$$I_B = I_{En} + I_{Cn} + I_{Br} \quad (\text{ec4.4})$$

Relaciones más importantes.

Parámetros  $\alpha$  y  $\beta$ :

En un transistor bipolar uno de los aspectos más interesantes para su análisis y uso es el conocer las relaciones existentes entre sus tres corrientes ( $I_E$ ,  $I_B$  e  $I_C$ ). En la ecuación 1 tenemos una primera relación. Otras relaciones se pueden obtener definiendo una serie de parámetros dependientes de la estructura del propio transistor.

Definimos los parámetros  $\alpha$  y  $\beta$  (de continua) como la relación existente entre la corriente de colector y la de emisor, o la de emisor y la de base, es decir:

$$\alpha = \frac{I_C}{I_E} \quad \beta = \frac{I_C}{I_B} \quad (\text{ec4.5})$$

$$\beta = \frac{I_C}{I_B} = \frac{I_C}{I_E - I_C} = \frac{I_C}{I_E(1 - \frac{I_C}{I_E})} = \frac{\alpha}{1 - \alpha} \quad (\text{ec4.6})$$

En general el parámetro  $\alpha$  será muy próximo a la unidad (la corriente de emisor será similar a la de colector) y el parámetro  $\beta$  tendrá un valor elevado (normalmente  $> 100$ ).

A partir de las ecuaciones anteriores se puede obtener una más que es útil cuando se trabaja con pequeñas corrientes de polarización, en las que el efecto de la corriente inversa que circula entre colector y base puede no ser despreciable:

$$I_C = \beta I_B + (\beta + 1)I_{C0} \quad (\text{ec4.7})$$

En esta ecuación se ha denominado  $I_{C0}$  a la corriente inversa de saturación de la unión colector base, la cual, en general se puede aproximar por  $I_{Cn}$ , y corresponde a la corriente que circularía por dicha unión polarizada en inversa si se deja al aire el terminal de emisor.

Regiones de funcionamiento

Corte:

Cuando el transistor se encuentra en corte no circula corriente por sus terminales. Concretamente, y a efectos de cálculo, decimos que el transistor se encuentra en corte cuando se cumple la condición:  $I_E = 0$  ó  $I_E < 0$ .

Para polarizar el transistor en corte basta con no polarizar en directa la unión base-emisor del mismo, es decir, basta con que  $V_{BE}=0$ .

Activa:

La región activa es la normal de funcionamiento del transistor. Existen corrientes en todos sus terminales y se cumple que la unión base-emisor se encuentra polarizada en directa y la colector-base en inversa.

En general, y a efectos de cálculo, se considera lo siguiente:

$$V_{BE} = V_r \quad (\text{ec4.8})$$

$$I_C = \beta I_B \quad (\text{ec4.9})$$

Donde  $V_r$  es la tensión de conducción de la unión base-emisor (en general 0,6 voltios).

## Saturación

En la región de saturación se verifica que tanto la unión base-emisor como la base-colector se encuentran en directa. Se dejan de cumplir las relaciones de activa, y se verifica sólo lo siguiente:

$$V_{BE} = V_{BEsat} \quad (\text{ec4.10})$$

$$V_{CE} = V_{CEsat} \quad (\text{ec4.11})$$

Donde las tensiones base-emisor y colector-emisor de saturación suelen tener valores determinados (0,8 y 0,2 voltios habitualmente).

Es de señalar especialmente que cuando el transistor se encuentra en saturación circula también corriente por sus tres terminales, pero ya no se cumple la relación:

$$I_C = \beta I_B \quad (\text{ec4.12})$$

## Polarización del Transistor

Polarizar un transistor bipolar implica conseguir que las corrientes y tensiones continuas que aparecen en el mismo queden fijadas a unos valores previamente decididos. Se puede polarizar el transistor en zona activa, en saturación o en corte, cambiando las tensiones y componentes del circuito en el que se trabaja.

El transistor bipolar se emplea en numerosas aplicaciones, y en infinidad de circuitos diferentes. Cada uno de ellos lo polariza de forma determinada. Supongamos que se quiere polarizar un transistor bipolar en zona activa. Se ha de conseguir que sus tensiones y corrientes cumplan las condiciones de estar en activa:  $V_{BE} = 0,7V$ ,  $V_{CE} > 0,2V$ . Una primera opción sería usar un circuito como el de la figura 4.17. Podemos ver cómo conseguimos polarizar la unión base-emisor mediante una resistencia (R) conectada a alimentación. Por la base del transistor circulará una corriente igual a  $(V_{CC}-V_{BE})/R$ , y en colector-emisor tendremos  $V_{CE} = V_{CC} > V_{CEsat}$ .

Este circuito tiene como inconveniente por un lado que el transistor nunca se podría polarizar en saturación, pues no se puede conseguir que  $V_{CE} = 0,2V$  siendo  $V_{BE}=0,7V$ . Un circuito un poco más complejo, y con el que se puede conseguir polarizar al transistor en las tres regiones de funcionamiento es el de la figura 4.18.

Vemos que en este caso la tensión colector-emisor depende directamente de la corriente de base ( $V_{CE}=V_{CC}-\beta \cdot I_B \cdot R_C$ ), y dicha corriente se fija actuando sobre la resistencia de base ( $I_B= (V_{CC}-V_{BE})/R_B$ ). Para polarizar el transistor en cada una de las regiones se pueden emplear las dos ecuaciones mencionadas y aplicar las restricciones de cada región.

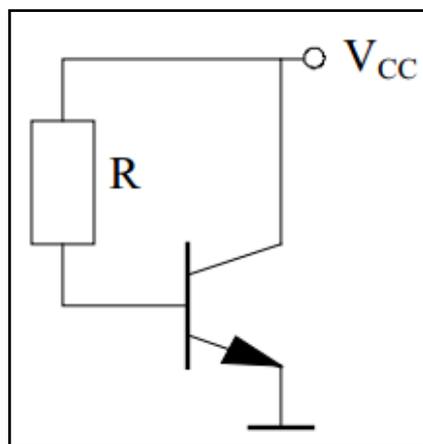


Figura 4.17. Polarización Zona Activa

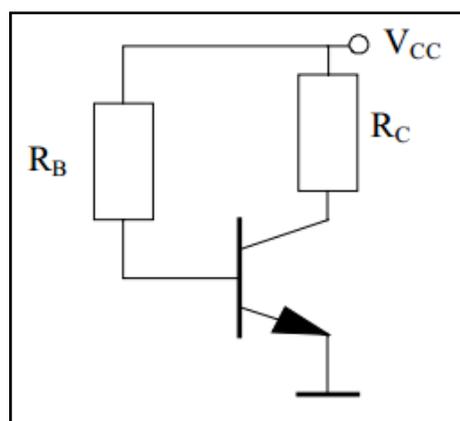


Figura 4.18. Polarización Zona de Saturación.

## 4.2 Componentes Electrónicos Pasivos

Son aquellos que no producen amplificación y que sirven para controlar la electricidad favoreciendo al mejor funcionamiento de los elementos activos. Los componentes pasivos están formados por elementos de diversas clases que tendremos que considerar independientemente, ya que son diferentes sus objetivos, construcción y resultados, Los componentes pasivos se dividen en: componentes pasivos lineales y componentes electromecánicos.

Componentes pasivos lineales:

- Condensador
- Resistor o Resistencia.
- Bobina.

Componentes electromecánicos:

- Interruptores
- Fusibles
- Conectores
- Relés
- Diodos.

### 4.2.1 Resistor

La función de estos componentes en un circuito eléctrico es limitar la cantidad de corriente o dividir el voltaje. La unidad de medida es el ohm ( $\Omega$ ) y su símbolo es como se muestra en la figura 4.19.

Tipo	Símbolo
Fijo	
Variable	

Figura 4.19. Símbolo de un resistor

Las dos principales características de un resistor son su resistencia  $R$  dada en ohms y la disipación de potencia  $W$ , se pueden encontrar en una amplia variedad de valores de  $R$ , desde unos cuantos ohms, hasta varios megaohms. La disipación de potencia es muy importante ya que indica la máxima cantidad de potencia que un resistor puede disipar (generalmente en forma de calor) sin sufrir un calentamiento excesivo. El término disipar significa que la potencia  $I^2R$  (derivada de la ley de ohm) se desperdicia, ya que no se utiliza el calor generado. Un calentamiento excesivo puede ocasionar que el elemento se dañe. Por lo general, como medida de seguridad, la disipación nominal de potencia de un resistor es mayor que la disipación de potencia real.

Se pueden encontrar dos tipos de construcción de resistores como son; de carbón, utilizados en aplicaciones de “baja” potencia igual o menor a 2 W y resistores de alambre que van de potencias igual o superiores a los 5 W, para potencias entre 2 y 5 W se pueden encontrar resistores tanto de carbón como de alambre.

### **Resistores de alambre devanado**

Para construir este tipo de resistores se utiliza un alambre conocido como alambre resistencia, este alambre se enrolla alrededor de un núcleo aislante, la longitud empleada y la resistividad específica determinan la resistencia  $R$  del elemento. Los materiales más comunes para la fabricación del alambre de resistencia son el tungsteno y la mangamina. El material del núcleo aislante generalmente es porcelana, un material fenólico parecido a la baquelita. Se emplea alambre sin recubrimiento, pero el resistor se encuentra, en general, cubierto por material aislante.

Este tipo de resistores son utilizados generalmente en aplicaciones de alta potencia y la resistencia es baja. Los valores de resistencia para este tipo de resistores van desde menos de un ohm hasta varios miles de ohms, estos resistores suelen ser empleados cuando es necesario tener valores exactos y estables de resistencia.

### **Resistores de carbón**

Se construyen con una mezcla de carbón o grafito y material aislante pulverizado que sirve como adhesivo. La cantidad de mezcla es la necesaria para obtener un valor específico de resistencia  $R$ , el resistor tiene un recubrimiento plástico que

sirve como aislante y que, además, le proporciona rigidez mecánica. Unidos a los extremos del elemento de carbón del resistor, se encuentran metalizaciones con terminales estañadas de cobre, una para cada extremo, que sirven para soldar el resistor a cualquier circuito.

Este tipo de terminal es conocida como terminal axial, debido a que su ubicación respecto al resistor coincide con el eje de éste.

Los resistores de carbón se encuentran disponibles con valores de resistencia R desde pocos ohms hasta 20 M $\Omega$ , la disipación nominal de potencia para este tipo de resistores tiene, en general, los siguientes valores: 0,1 – 0,125 – 0,25 – 0,5 – 1 y 2 W.

Una forma común para saber el valor R de un resistor es de acuerdo a el código de colores, la base de este sistema radica en el uso de colores para representar números; la correspondencia entre estos se muestra a continuación.

Color	Valor	Tolerancia
Negro	0	
Café	1	
Rojo	2	
Naranja	3	
Amarillo	4	
Verde	5	
Azul	6	
Lila	7	
Gris	8	
Blanco	9	
Plata	-	+/- 10%
Oro	-	+/- 5%
Ninguno	-	+/- 20%

Tabla 4.2 Código de colores

Este código de colores está estandarizado por la asociación de industrias electrónicas (EIA).

Las bandas de color se encuentran ubicadas en uno de los extremos del cuerpo aislante del resistor. La lectura se lleva a cabo de izquierda a derecha; la primera banda indica el primer dígito del valor de la resistencia R, con la siguiente banda se indica el segundo dígito, la tercera banda señala el valor del multiplicador decimal, el cual proporciona el número de ceros que sigue a los dos primeros dígitos. Como ejemplo obsérvese la figura siguiente:

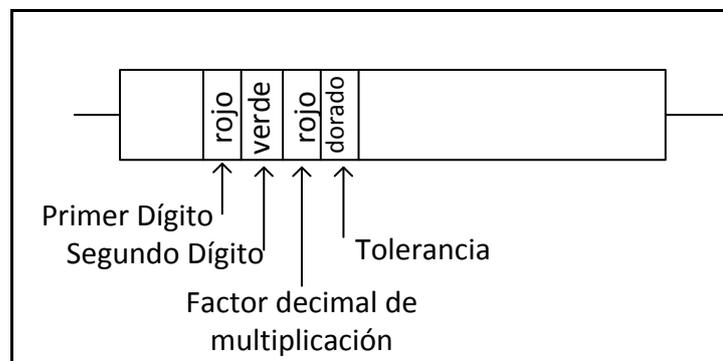


Figura 4.20. Ejemplo para código de colores

La primera banda de color rojo y de acuerdo a la tabla anterior, corresponde el número 2, la siguiente banda, verde, indica el número 5, la tercera banda, rojo, indica nuevamente el número 2 y la cuarta banda. Dorado indica la tolerancia de  $\pm 5\%$ .

Quedaría de la siguiente manera:

2	5	x	100	=	2500 $\Omega$	ó	2,5 k $\Omega$
Con una tolerancia de $\pm 5\%$							

Este valor de 2500  $\Omega$  es un valor relativamente "ideal" o que su valor central es de 2500  $\Omega$  ya que al medirlo con un instrumento muy preciso puede ser que el instrumento indique otro valor, esto es debido principalmente a la tolerancia del mismo resistor, entre otros factores, en este caso la tolerancia es de  $\pm 5\%$ , por lo que su valor bien puede estar en el rango de 2375  $\Omega$  a 2625  $\Omega$ , sin embargo para

efectos de diseño en algún circuito de uso general, esto no representa ningún problema ya que los valores son muy cercanos al valor real  $R$  que se pueden encontrar en el mercado y esto es debido a la tecnología que se utiliza para la fabricación de los componentes.

Sin embargo si el diseño que se está realizando requiere ser preciso en su función, entonces es muy posible que se requiera la utilización de resistores de precisión ya sean fijos y variables, de los conocidos multivuelta, por ejemplo, esto conllevará a un aumento en el costo del circuito ya que estos elementos son relativamente más caros que un resistor común.

Para algunos elementos en particular en donde no se señala el elemento con el código de colores, como es el caso de termistores, varistores o de fotoresistencias, será necesario recurrir a la hoja de especificaciones del fabricante para poder conocer el rango de operación de dichos elementos o en último caso se tendría que hacer la medición directa con la ayuda de un medidor de resistencia y modificar la variable que afecta el valor  $R$  del elemento para poder conocer el rango de operación, sin embargo esto no es deseable ya que será solamente una prueba de ensayo y error para acercarse al posible valor.

### Resistores variables

Estos resistores pueden ser de alambre devanado o de carbón.

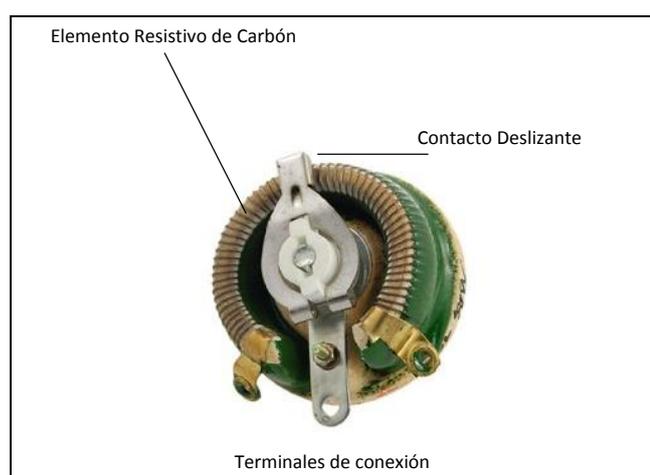


Figura 4.21. Resistor Variable

Fuente: RADIO CONTROL. Resistor. [http://www.e-aeromodelismo.com.ar/Notas/radiocontrol/img/faq\\_pot1.jpg](http://www.e-aeromodelismo.com.ar/Notas/radiocontrol/img/faq_pot1.jpg). [Consulta: 23 de julio de 2012]

En el interior del encapsulado metálico del resistor de la figura anterior el control tiene un disco metálico en el que se encuentra el elemento resistivo de carbón.

Este último puede estar formado por un recubrimiento delgado sobre papel comprimido, o por un disco moldeado de carbón. Unidas a cada uno de los extremos, se encuentran las terminales, la terminal ubicada en la parte media está conectada al brazo de posición variable, que entra en contacto con el elemento resistivo a través de una escobilla formada por un resorte metálico, conforme el vástago gira el brazo rotatorio mueve el contacto deslizante por lo que entra en contacto con el elemento resistivo en diferentes puntos. Cuando el contacto se desplaza hacia uno de los extremos, el valor de la resistencia  $R$  disminuye en esta terminal y la que corresponde al brazo rotatorio, sin embargo, entre los dos extremos, la resistencia  $R$  no es variable y siempre tiene el valor máximo de la resistencia.

Es común que el valor de la disipación nominal de potencia en este tipo de resistores se encuentre entre 0,5 y 2 W.

En ocasiones es posible encontrar a estos resistores variables en combinación con interruptores de cierta potencia, con lo que se logra abrir o cerrar un interruptor al girar el vástago y posteriormente variar la resistencia, este tipo de elementos era muy frecuente encontrarlos en radios, al girar la perilla del volumen era posible encender también el aparato.

#### 4.2.2 Condensador

Un condensador (figura 4.22) es un componente formado por dos conductores próximos separados por un dieléctrico (aire, aceite, papel) y cuya función es almacenar cargas eléctricas para posteriormente utilizarlas cuando necesitemos.

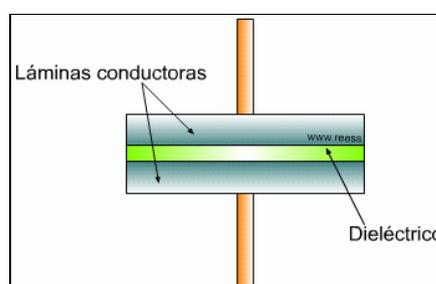


Figura 4.22. Condensador

Fuente: WANDOO. Condensador.

<http://perso.wanadoo.es/jmarti51/componentes/imagenes/condesa1.gif>. [Consulta 23 de Julio de 2012]

Un condensador almacena gran cantidad de electricidad con diferencias de potencial pequeñas.

Definimos capacidad de un condensador como la relación entre la carga y la diferencia de potencial y lo expresamos como:

$$C = \frac{Q}{U} \quad (\text{ec4.13})$$

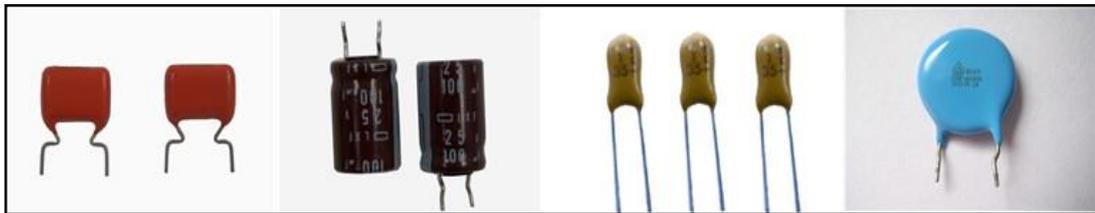


Figura 4.23. Tipos de Condensadores  
Fuente: PLANETA ELECTRONICO. Condensador.

<http://www.planetaelectronico.com/cursillo/tema2/tema2.3.htm>. [Consulta: 23 de Julio de 2012]

En el Sistema Internacional la capacidad se mide en faradios. El faradio es una unidad demasiado grande para el posible almacenaje de cargas eléctricas, por lo que se utilizan unidades derivadas más pequeñas.

Milifaradio:  $1\text{mF} = 10^{-3}F$

Microfaradio:  $1\ \mu\text{F} = 10^{-6}F$

Nanofaradio:  $1\ \text{nF} = 10^{-9}F$

Picofaradio:  $1\ \text{pF} = 10^{-12}F$

Por su forma tenemos los siguientes tipos de condensadores:

- Condensador plano: Está constituido por dos láminas planas separadas por un dieléctrico.
- Condensador cilíndrico: Está constituido por dos cilindros conductores concéntricos.
- Condensador esférico: Está constituido por dos conductores esféricos concéntricos

Funcionamiento:

Cuando se aplica una diferencia de potencial a sus armaduras (Figura 4.24), los electrones que llegan a una de ellas no pueden atravesar el dieléctrico pero la carga eléctrica negativa repele a los electrones de la otra armadura creando un potencial positivo, generando en el condensador una diferencia de potencial igual que la tensión aplicada.

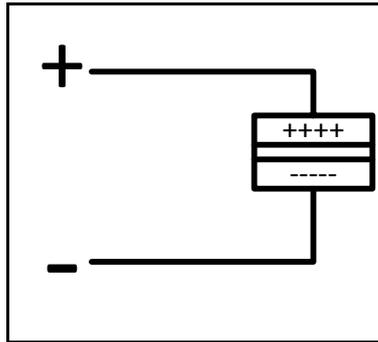


Figura 4.24. Funcionamiento del Condensador

La relación entre la carga eléctrica almacenada medida en culombios y el potencial eléctrico medido en voltios se conoce como la capacidad del condensador y su unidad de medida es el Faradio (F).

Un Faradio es la capacidad de un condensador en el que, sometidas sus armaduras a una diferencia de potencial de un voltio, éstas adquieren una carga eléctrica de 1 culombio.

Para evitar daños al condensador, la tensión de carga se le aplica en serie con una resistencia que limita la corriente eléctrica como se muestra en la figura siguiente.

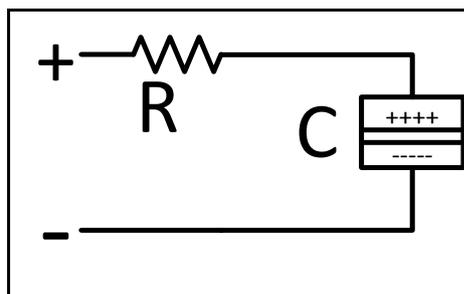


Figura 4.25. Resistencia limitadora conectada al condensador

Al aplicar una tensión continua al condensador a través de la resistencia, el condensador irá modificando su carga hasta que la tensión entre sus armaduras sea igual que la del generador de tensión.

Si la tensión aplicada es mayor a la del condensador, diremos que el condensador se carga, en caso de que sea menor, el condensador se descarga.

Si llamamos:

$V_i$ , a la tensión inicial que tiene el condensador.

$V_f$ , a la tensión final a la que se cargará el condensador.

$R$ , al valor expresado en ohmios de la resistencia limitadora.

$C$ , a la capacidad del condensador expresada en faradios.

El valor de la tensión instantánea durante la carga en función del tiempo viene dada por la siguiente expresión:

$$V_c = V_f + (V_i - V_f)e^{\frac{-t}{rc}} \quad (\text{ec4.14})$$

Y la representación gráfica de la carga y descarga del condensador:

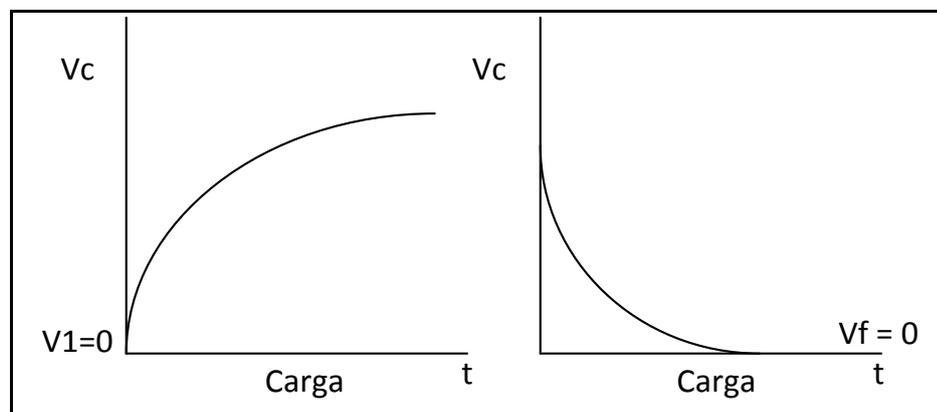


Figura 4.26. Carga y descarga del condensador

Según la ecuación, el tiempo necesario para que el condensador alcance la tensión final, es decir se cargue completamente será:

$$V_c = V_f \quad (\text{ec4.15})$$

$$(V_i - V_f)e^{\frac{-t}{r.c}} \quad (\text{ec4.16})$$

$$t = \textit{infinito} \quad (\text{ec4.17})$$

Se considera que el condensador ha llegado al valor final de la tensión aplicada cuando alcanza el 90% de dicha tensión, es decir cuando ha alcanzado una tensión de:

$$V_c = 0,9 * V_f \quad (\text{ec4.18})$$

Si partimos del condensador descargado,  $V_i = 0$ , el tiempo necesario para la carga del condensador será:

$$0,9V_f = V_f + (0 - V_f)e^{\frac{-t}{r.c}} \quad (\text{ec4.19})$$

Despejamos el tiempo:

$$t = \textit{Ln}10 r.c \quad (\text{ec4.20})$$

El valor de la resistencia por la capacidad,  $r.c$  se conoce como la constante de tiempo del circuito. Podemos comprobar que la carga del condensador no depende de la tensión que se le aplica sino que depende exclusivamente de la capacidad del condensador y la resistencia.

$\textit{Ln}10$  nos aproxima a 2,3 pero como norma general se acepta que el tiempo de carga de un condensador es igual a:

$$t = 3.r.c \quad (\text{ec4.21})$$

### 4.2.3 Diodo LED.

El LED (*Light-Emitting Diode*: Diodo Emisor de Luz), es un dispositivo semiconductor que emite luz cuando se polariza de forma directa la unión PN en la cual circula por él una corriente eléctrica. Este fenómeno es una forma de electroluminiscencia, el LED es un tipo especial de diodo que trabaja como un diodo común, pero que al ser atravesado por la corriente eléctrica, emite luz. Este dispositivo semiconductor está comúnmente encapsulado en una cubierta de plástico de mayor resistencia que las de vidrio que usualmente se emplean en las lámparas incandescentes. Aunque el plástico puede estar coloreado, es sólo por razones estéticas, ya que ello no influye en el color de la luz emitida. Usualmente un LED es una fuente de luz compuesta con diferentes partes, razón por la cual el patrón de intensidad de la luz emitida puede ser bastante complejo.

Para obtener una buena intensidad luminosa debe escogerse bien la corriente que atraviesa el LED y evitar que este se pueda dañar; para ello, hay que tener en cuenta que el voltaje de operación va desde 1,8 hasta 3,8 voltios aproximadamente (lo que está relacionado con el material de fabricación y el color de la luz que emite) y la gama de intensidades que debe circular por él varía según su aplicación. Los valores típicos de corriente directa de polarización de un LED están comprendidos entre los 10 y 20 miliamperios (mA) en los diodos de color rojo y de entre los 20 y 40 (mA) para los otros LED. Los diodos LED tienen enormes ventajas sobre las lámparas indicadoras comunes, como su bajo consumo de energía, su mantenimiento casi nulo y con una vida aproximada de 100,000 horas.

En general, los LED suelen tener mejor eficiencia cuanto menor es la corriente que circula por ellos, con lo cual, en su operación de forma optimizada, se suele buscar un compromiso entre la intensidad luminosa que producen (mayor cuanto más grande es la intensidad que circula por ellos) y la eficiencia (mayor cuanto menor es la intensidad que circula por ellos).

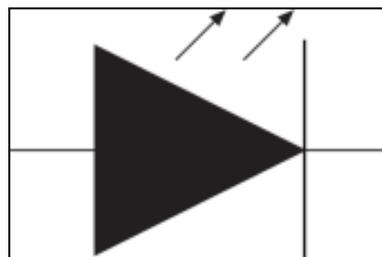


Figura 4.27. Símbolo del diodo led

#### 4.2.4. Relés.

Un relé es un interruptor accionado por un electroimán. Un electroimán está formado por una barra de hierro dulce, llamada núcleo, rodeada por una bobina de hilo de cobre (Figura 4.28). Al pasar una corriente eléctrica por la bobina, el núcleo de hierro se magnetiza por efecto del campo magnético producido por la bobina, convirtiéndose en un imán tanto más potente cuanto mayor sea la intensidad de la corriente y el número de vueltas de la bobina. Al abrir de nuevo el interruptor y dejar de pasar corriente por la bobina, desaparece el campo magnético y el núcleo deja de ser un imán.

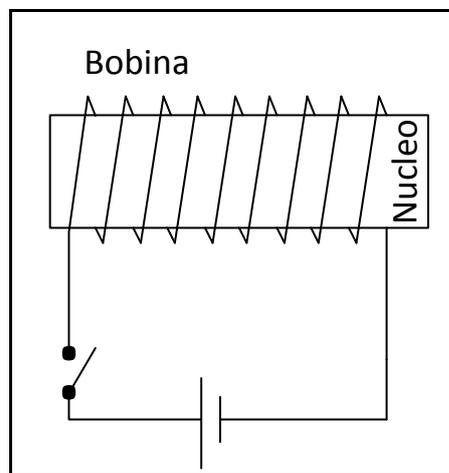


Figura 4.28. Bobina

El relé más sencillo está formado por un electroimán como el descrito anteriormente y un interruptor de contactos (Figura 4.29). Al pasar una pequeña corriente por la bobina, el núcleo se imanta y atrae al inducido por uno de sus extremos, empujando a uno de los contactos hasta que se juntan, permitiendo el paso de la corriente a través de ellos. Esta corriente es, normalmente, mucho mayor que la que pasa por la bobina.

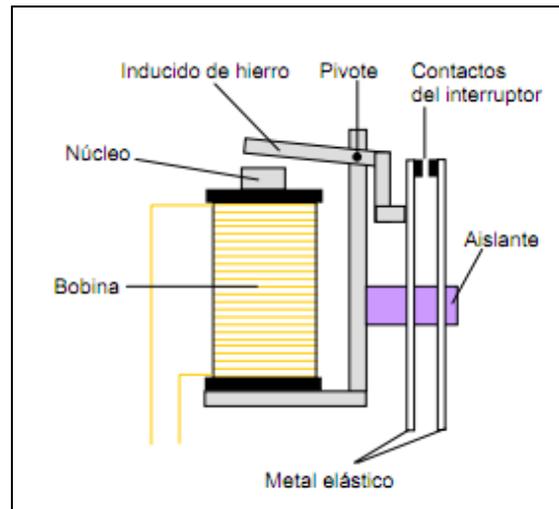


Figura 4.29. Relé tipo interruptor

Fuente: PLANETAELECTRONICO. Relé.

<http://platea.pntic.mec.es/~pcastela/tecno/documentos/apuntes/rele.pdf>. [Consulta 23 de Julio de 2012]

El símbolo del relé es el que puede verse en la Figura 4.30. La bobina se representa por un rectángulo alargado con una línea a 45° que lo atraviesa en su parte central. El interruptor de contactos se representa como un interruptor normal. Entre la bobina y el interruptor se establece un vínculo mediante una línea de trazos, para dar a entender que el interruptor se cierra por efecto de la bobina.

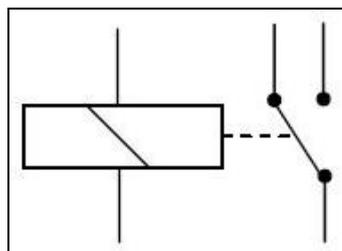


Figura 4.30. Simbología de un relé

### Tipos de Relés.

El relé que hemos visto hasta ahora funciona como un interruptor. Está formado por un contacto móvil o polo y un contacto fijo. Pero también hay relés que funcionan como un conmutador, porque disponen de un polo (contacto móvil) y dos contactos fijos (figura 4.31).

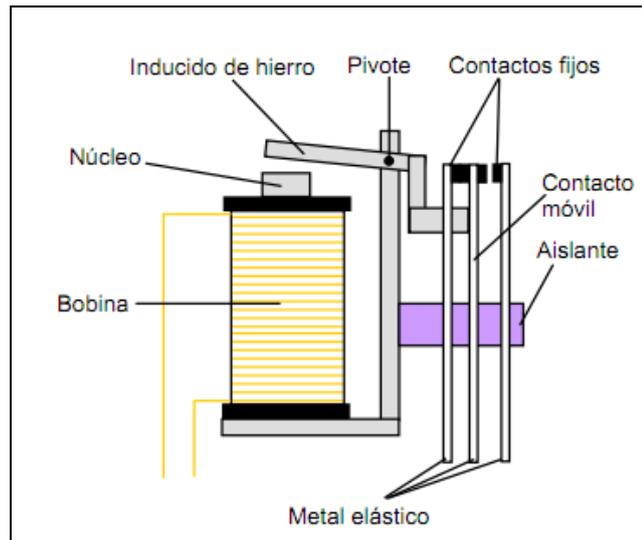


Figura 4.31. Relé tipo conmutador

Fuente: PLANETA ELECTRONICO. Relé.

<http://platea.pntic.mec.es/~pcastela/tecno/documentos/apuntes/rele.pdf>. [Consulta 23 de Julio de 2012]

Cuando no pasa corriente por la bobina el contacto móvil está tocando a uno de los contactos fijos. En el momento que pasa corriente por la bobina, el núcleo atrae al inducido, el cual empuja al contacto móvil hasta que toca al otro contacto fijo. Por tanto, funciona como un conmutador. En la Figura 4.32 puede verse el símbolo de este tipo de relé.

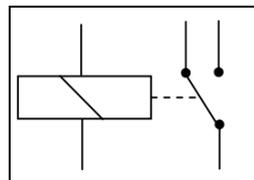


Figura 4.32. Símbolo relé tipo conmutador

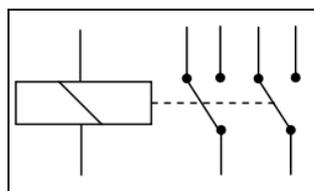


Figura 4.33. Símbolo de un relé de 2 polos

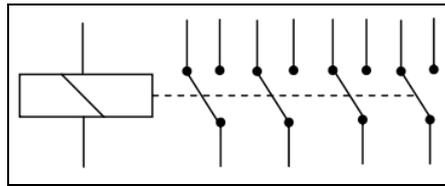


Figura 4.34. Símbolo de un relé de 4 polos

También existen relés con más de un polo (contacto móvil) siendo muy interesantes para los proyectos de tecnología, los relés conmutadores de dos polos y los de cuatro polos se muestran en: (figura 4.33), (figura 4.34).

#### 4.2.5. Conectores

Un conector eléctrico es un dispositivo para unir circuitos eléctricos. Están compuestos generalmente de un enchufe (macho) y una base (hembra).

Los conectores eléctricos se caracterizan por su patillaje y construcción física, tamaño, resistencia de contacto, aislamiento entre los pines, robustez y resistencia a la vibración, resistencia a la entrada de agua u otros contaminantes, resistencia a la presión, fiabilidad, tiempo de vida (número de conexiones/desconexiones antes de que falle), y facilidad de conexión y desconexión.

Pueden estar hechos para impedir que se conecten de manera incorrecta, conectando los pines equivocados donde van otros, y tener mecanismos de bloqueo para asegurar que están completamente conectados y no puedan soltarse o salirse. Algunos conectores están diseñados de tal manera que ciertos pines hagan contacto antes que otros hayan sido insertados, evitando así el rompimiento durante la desconexión; de esta manera se protegen los circuitos que suelen tener conectores de alimentación.

Por lo general, es conveniente un conector que sea fácil de identificar visualmente y de ensamblar, que sólo requiera de herramientas sencillas, y sea económico. En algunos casos el fabricante de equipos puede optar por un conector específico debido a que no es compatible con otros conectores, lo que permite el control de lo que puede ser conectado. Ningún conector tiene todas las propiedades ideales; la variada gama de conectores es un reflejo de los diferentes requisitos.

#### 4.2.6. Interruptores

Un interruptor eléctrico es un dispositivo que permite desviar o interrumpir el curso de una corriente eléctrica. En el mundo moderno sus tipos y aplicaciones son innumerables, van desde un simple interruptor que apaga o enciende un bombillo, hasta un complicado selector de transferencia automático de múltiples capas controlado por computadora.

Su expresión más sencilla consiste en dos contactos de metal inoxidable y el actuante. Los contactos, normalmente separados, se unen mediante un actuante para permitir que la corriente circule. El actuante es la parte móvil que en una de sus posiciones hace presión sobre los contactos para mantenerlos unidos.

#### Clasificación de los Interruptores

##### Actuantes

Los interruptores actuantes pueden ser normalmente abiertos, en cuyo caso al accionarlos se cierra el circuito o normalmente cerrados en cuyo caso al accionarlos se abre el circuito.

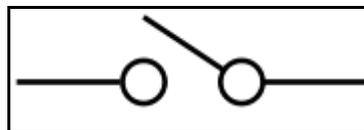


Figura 4.35. Simbología de un interruptor.

##### Pulsadores

También llamados interruptores momentáneos. Este tipo de interruptor requiere que el operador mantenga la presión sobre el actuante para que los contactos estén unidos. Un ejemplo de su uso lo podemos encontrar en los botones de un teclado.

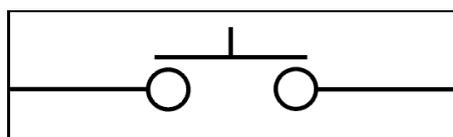


Figura 4.36. Simbología de un pulsador.

### Cantidad de polos

Son la cantidad de circuitos individuales que controla el interruptor. Un interruptor de un solo polo como el que usamos para encender una lámpara. Los hay de 2 o más polos. Por ejemplo si queremos encender un motor de 220 voltios y a la vez un indicador luminoso de 12 voltios necesitaremos un interruptor de 2 polos, un polo para el circuito de 220 voltios y otro para el de 12 voltios.

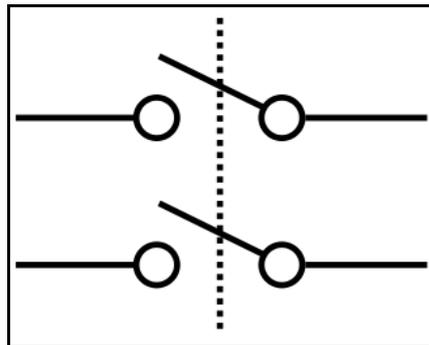


Figura 4.37. Simbología de un interruptor de doble polo.

### Cantidad de vías (tiros)

Es la cantidad de posiciones que tiene un interruptor. Tomando como ejemplo el encendido de una lámpara, en una posición enciende la lámpara mientras que en la otra se apaga.

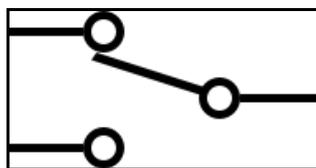


Figura 4.38. Simbología de un interruptor de doble vía

### Combinaciones

Se pueden combinar las tres clases anteriores para crear diferentes tipos de interruptores. En el gráfico inferior podemos ver un ejemplo de un interruptor DPDT.

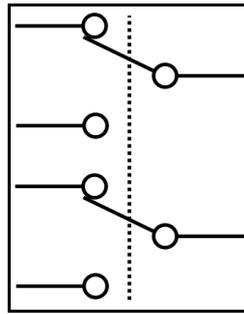


Figura 4.39. Simbología de un interruptor de doble polo y doble vía

Conociendo el funcionamiento de los componentes electrónicos activos y pasivos podemos explicar el funcionamiento de los periféricos o accesorios de entrada/salida que serán conectados al sistema de control.

El manejo de este sistema se hará localmente con un teclado matricial 4x4 (cuatro filas y 4 columnas) y un LCD matricial de 4x20 (cuatro filas y 20 columnas). Y de forma remota con un dispositivo móvil que soporte el uso de red GSM.

### 4.3 Teclado Matricial

Es un dispositivo de entrada que consiste en un arreglo matricial de 16 teclas (pulsadores). Dispone de un conector SIL (Single In Line) de 8 pines: 4 filas y 4 columnas. La interconexión de los pulsadores es como se muestra en la Figura siguiente:

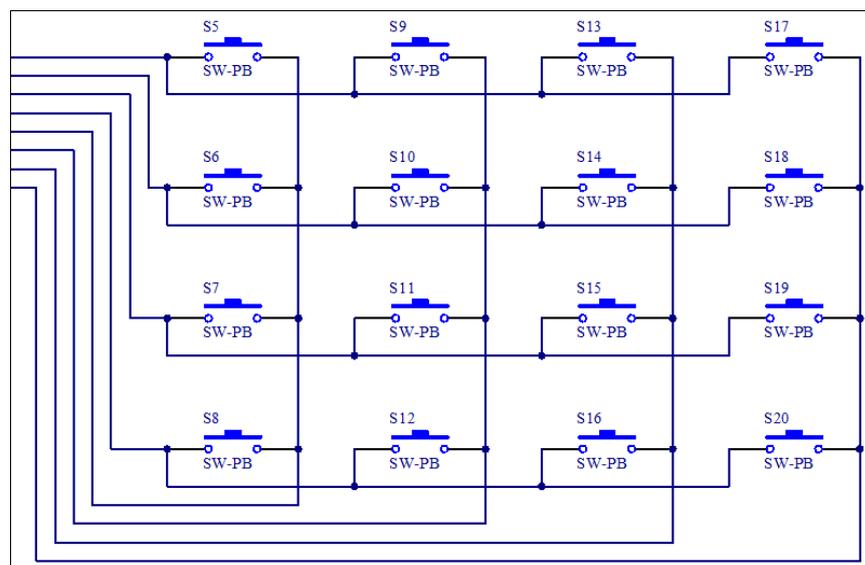


Figura 4.40. Pulsadores del Teclado Matricial

Cuando se presiona una tecla se conecta una fila con una columna. La manera de averiguar la tecla presionada consiste en enviar un nivel lógico a una de las filas diferente al nivel de las filas restantes y buscar la columna por la que se recibe el mismo nivel lógico. De esta manera se obtiene la fila y columna al igual que en una matriz. Figura 4.4.

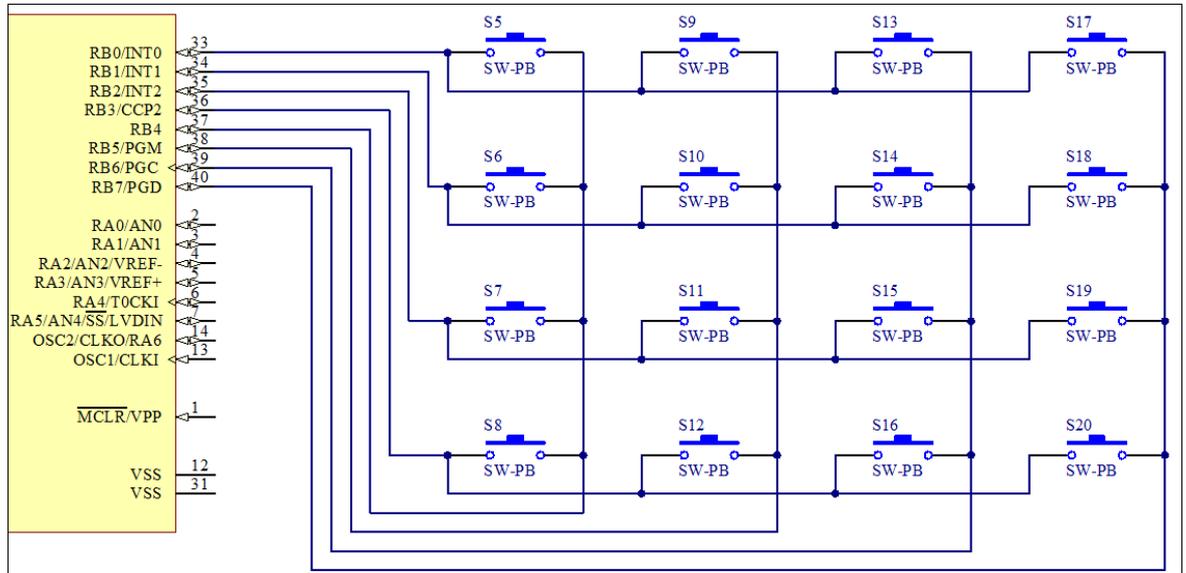


Figura 4.41. Conexión de Teclado Matricial a un Microcontrolador.

#### 4.4. LCD Matricial 4x20.

La pantalla de cristal líquido o LCD (Liquid Crystal Display) es un dispositivo microcontrolado de visualización gráfico para la presentación de caracteres, símbolos o incluso dibujos (en algunos modelos), en este caso dispone de 4 filas de 20 caracteres cada una y cada carácter dispone de una matriz de 5x7 puntos (pixels), aunque los hay de otro número de filas y caracteres. Este dispositivo está gobernado internamente por un microcontrolador Hitachi 44780 y regula todos los parámetros de presentación, este modelo es el más común.

Los displays LCD son mayoritariamente estándar y se controlan de formas muy parecidas, incluso cuando no coincide el número de caracteres. En la figura 4.42 podemos ver los elementos básicos de un display LCD estándar.

Por un lado se tiene el bus de datos D0..D7 que será conectado a un puerto del uC PIC. Este bus de datos puede ser de entrada al LCD (para escribir caracteres y

enviar instrucciones) o puede ser de salida del LCD (para poder leer el estado por ejemplo). El LCD tiene tres señales de control: E (enable) sirve para habilitar el LCD, la señal R/W sirve para indicar operación de lectura o escritura; por último se la señal RS es la de sincronismo de datos e instrucciones. Además el LCD tiene señales para alimentación y una señal (VEE) que sirve para controlar el contraste de la pantalla.

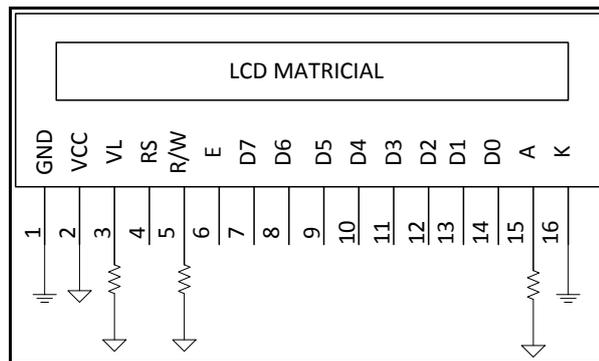


Figura 4.42. Diagrama de conexiones del LCD

Pin	Símbolo	Descripción
1	GND	Pin de conexión a tierra
2	VCC	Pin de alimentación 5V
3	VL	Pin de contraste. Normalmente se conecta a un potenciómetro a través del cual se aplica una tensión variable de 0 a 5V que permite controlar el contraste del LCD
4	RS	Selección del registro de control/registro de datos: RS=0 Selección de registro de control RS=1 Selección de registro de datos
5	R/W	Señal de lectura/escritura R/W=0 El modulo LCD es escrito R/W=1 El modulo LCD es leído
6	E	Señal de activación del módulo LCD E=0 Modulo desconectado E= 1 Modulo conectado
7...14	D0-D7	Bus de datos bidireccional, a través de estas líneas se realiza la transferencia de información entre el modulo LCD y el sistema informático que lo gestiona.

Tabla 4.3. Descripción de senales empleadas por el modulo

En la tabla 3.3 se muestra las señales que utilizan los LCD y los pines con sus respectivos nombres.

Para los diferentes sistemas a controlar se utilizan los siguientes accesorios o periféricos. Para el sistema de iluminación: lámparas led. Para el sistema de alarma: sensores infrarrojos, sensores magnéticos, sirenas, luces indicadoras, fuentes de poder, baterías. Para el sistema de calefacción: ventiladores, niquelinas.

#### **4.5. Lámparas LED**

Una lámpara de led es una lámpara de estado sólido que usa leds (Light-Emitting Diode, Diodos Emisores de Luz) como fuente luminosa. Debido a que la luz capaz de emitir un led no es muy intensa, para alcanzar la intensidad luminosa similar a las otras lámparas existentes como las incandescentes o las fluorescentes compactas, las lámparas LED están compuestas por agrupaciones de leds, en mayor o menor número, según la intensidad luminosa deseada (figura 3.43).

Actualmente las lámparas led se pueden usar para cualquier aplicación comercial, desde el alumbrado decorativo hasta el de viales y jardines, presentando ciertas ventajas, entre las que destacan su considerable ahorro energético, arranque instantáneo, aguante a los encendidos y apagados continuos y su mayor vida útil, pero también con ciertos inconvenientes como su elevado costo inicial.

Los diodos funcionan con energía eléctrica de corriente continua (CC), de modo que las lámparas de LED deben incluir circuitos internos para operar desde el voltaje CA estándar. Los leds se dañan a altas temperaturas, por lo que las lámparas de LED tienen elementos de gestión del calor, tales como disipadores y aletas de refrigeración. Las lámparas de LED tienen una vida útil larga y una gran eficiencia energética, pero los costos iniciales son más altos que los de las lámparas fluorescentes.



Figura 4.43. Lámparas LED

Fuente: LEDBOX. Led. <http://blog.ledbox.es/wp-content/themes/striking/cache/images>. [Consulta: 23 de Julio de 2012]

#### 4.6. Detector de movimiento

Un detector de movimiento es un dispositivo electrónico equipado de sensores que responden un movimiento físico. Se encuentran, generalmente, en sistemas de seguridad o en circuitos cerrados de televisión.

##### 4.6.1. Sensor infrarrojo pasivo (PIR)

Estos sensores son empleados mayormente en alarmas, puertas automáticas y para el ahorro de energía. Los sensores de movimientos PIR no emiten ningún rayo, por eso se les denomina sensor pasivo. Figura 4.44.

Cuando un objeto tiene una temperatura diferente a la del medio y se encuentra dentro del campo de captación del sensor, la radiación calórica del objeto será captada por los lentes del elemento sensorial ocasionando el accionamiento del sensor. Todo cuerpo irradia calor siempre y cuando su temperatura sea superior al cero absoluto (-273o C). La intensidad y la radiación espectral de esta radiación electromagnética dependen de la temperatura y del llamado Grado de Emisión del Cuerpo.

Un cuerpo caliente irradia un espectro de diferentes longitudes de onda, el cual caracteriza al cuerpo en sí. Un cuerpo negro no irradia ni tampoco deja atravesar rayos electromagnéticos

Los rayos infrarrojos provenientes de un ser humano son detectados por un sensor de movimiento PIR debido a un sistema de lentes Fresnel o mediante un espejo que contiene cristales piroeléctricos.



Figura 4.44. Sensor Infrarrojo pasivo (PIR)

Fuente: MLSTATIC. Sensor Infrarrojo. [http://img2.mlstatic.com/sensor-de-movimiento-infrarrojo-antimascota-rokonet-12mts\\_MLA-O-3645920651\\_012013.jpg](http://img2.mlstatic.com/sensor-de-movimiento-infrarrojo-antimascota-rokonet-12mts_MLA-O-3645920651_012013.jpg). [Consulta: 2 de Agosto de 2012]

#### 4.6.2. Limitaciones en el Empleo de Detectores de Movimiento PIR

Las condiciones para un buen funcionamiento de los sensores PIR es que la diferencia de temperatura entre el ambiente y el objeto sea mayor de 2°C, aunque ya existen versiones que detectan una diferencia menor de temperatura.

Además, el objeto que irradia debe tener un tamaño mínimo para poder asegurar una buena detección. El vidrio atenúa bastante los rayos infrarrojos de manera que el sensor PIR no llega a detectar los objetos que están detrás de puertas o ventanas de vidrio. Algunos focos o lámpara halógenas emiten rayos infrarrojos de manera que pueden confundir al sensor PIR.

#### 4.7. Detector de Apertura

Basan su principio de funcionamiento en que, al acercarse un imán, el sensor detecta. Internamente, poseen un reed switch, que es el que provoca la detección. Son extremadamente económicos, pero poseen una vida más limitada que cualquier otro tipo de sensor (poseen una lámina metálica que tiene movimiento mecánico, con el tiempo se daña), pero rinde muchísimas más operaciones que un

microswitch mecánico standard. Los hay cilíndricos en varios diámetros y rectangulares.

#### 4.7.1 Reed Switch

Es un interruptor eléctrico activado por un campo magnético. Cuando los contactos están normalmente abiertos se cierran en la presencia de un campo magnético; cuando están normalmente cerrados se abren en presencia de un campo magnético.

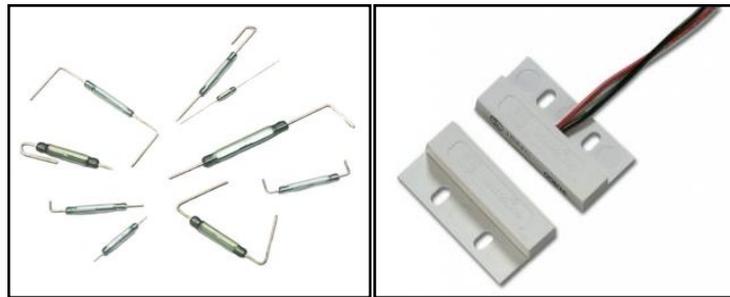


Figura 4.45. Detector de Apertura con reed switch

Fuente: REEDSWITCH. Sensor de Apertura. <http://www.reed-switch-info.com/resources/Custom+Reed+Switch.png>. [Consulta: 2 de Agosto de 2012]

El reed switch consiste en un par de contactos ferrosos encerrados al vacío dentro un tubo de vidrio. Cada contacto está sellado en los extremos opuestos del tubo de vidrio. El tubo de vidrio puede tener unos 10 mm de largo por 3 mm de diámetro. Al acercarse a un campo magnético, los contactos se unen cerrando un circuito eléctrico. La rigidez de los contactos hará que se separen al desaparecer el campo magnético. Para asegurar la durabilidad, la punta de los contactos tiene un baño de un metal precioso. El campo magnético puede estar generado por un imán permanente o por una bobina. Como los contactos están sellados, los reed switch son empleados en lugares con atmósferas explosivas, donde otros interruptores se consideran peligrosos. Esto se debe a que la chispa que se produce al abrir o cerrar sus contactos queda contenida dentro del tubo de vidrio. Los reed switch se diseñan en base al tamaño del campo magnético frente al que deben actuar. La sensibilidad de sus contactos se cambia al variar la aleación con que se fabrican, modificando su rigidez y su coeficiente magnético.

### **4.7.2 Aplicaciones**

Los reed switch son utilizados ampliamente en el mundo moderno como partes de circuitos eléctricos. Un uso muy extendido se puede encontrar en los sensores de las puertas y ventanas de las alarmas antirrobo, el imán va unido a la puerta y el reed switch al marco. En los sensores de velocidad de las bicicletas el imán está en uno de los radios de la rueda, mientras que el reed switch va colocado en la horquilla. Algunos teclados de computadoras son diseñados colocando imanes en cada una de las teclas y los reed switch en el fondo de la placa, cuando una tecla es presionada el imán se acerca y activa sus reed switches. Actualmente esta solución es obsoleta, usándose interruptores capacitivos que varían la condición de un circuito resonante.

Los reed switch también tienen desventajas, por ejemplo sus contactos son muy pequeños y delicados por lo cual no puede manejar grandes valores de tensión o corriente lo que provoca chispas en su interior que afectan su vida útil. Además, grandes valores de corriente pueden fundir los contactos y el campo magnético que se genera puede llegar a desmagnetizar los contactos.

## **4.8 Indicadores de Alarma**

Cuando el sistema de alarma ha sido activado por el cambio de estado de un algún sensor, tendrá que emitir una señal de alarma que puede ser audible o simplemente visual. Para este fin se ocupan sirenas electrónicas, bocinas, luces, etc.

### **4.8.1 Sirenas electrónicas**

La sirena electrónica se compone de una unidad de control que ha almacenado en el interior de la secuencia de tonos, y uno o dos altavoces conectados a esta unidad. El uso de sirenas electrónica está muy extendido, siendo especialmente adecuadas para su funcionamiento continuo, también tienen un bajo consumo eléctrico y no requieren mantenimiento.



Figura 4.46. Sirena Electrónica

Fuente: INTERTRONIC. Sirena Alarma. <http://www.intertronic.com.ve/wp-content/uploads/2011/08/sirena-alarma-110dB.jpg>. [Consulta: 2 de Agosto de 2012]

La sirena electrónica de última generación utiliza altavoces muy potentes que permiten una mayor audibilidad y, por tanto, una mayor eficacia. Algunos incluso han llegado a 200 vatios de potencia cada uno.

#### 4.8.2. Luces estroboscópicas

La luz estroboscópica es una fuente luminosa que emite una serie de destellos muy breves en rápida sucesión.

Los destellos de iluminación normalmente son producidos mediante una lámpara de descarga gaseosa como, por ejemplo, una lámpara fluorescente, aunque generalmente se emplean lámparas de flash por su mayor intensidad luminosa.



Figura 4.47. Luz Estroboscópica

Fuente: MLSTATIC. Luz Estroboscópica. [http://img1.mlstatic.com/s\\_MEC\\_v\\_T\\_f\\_8129951\\_9685.jpg](http://img1.mlstatic.com/s_MEC_v_T_f_8129951_9685.jpg). [Consulta: 2 de Agosto de 2012]

### 4.9. Conexión y configuración del Teclado Matricial 4x4

Para controlar el teclado, los puertos del uC correspondientes a las filas se programan como salidas y los conectados a las columnas del teclado se programan como entradas.

De tal forma que el objetivo principal del algoritmo para decodificar el teclado consiste en determinar la fila y columna que corresponde a la tecla que se presionó. Lo anterior se logra rotando un valor lógico (ya sea 1 o 0) en cada una de las líneas configuradas como salidas (filas en este caso) e inmediatamente después leer el estado lógico de las líneas conectadas como entrada (columnas).

Cuando el valor lógico que se rota es un 1, al algoritmo se le denomina walking ones y walking zeros cuando se trata de un 0.

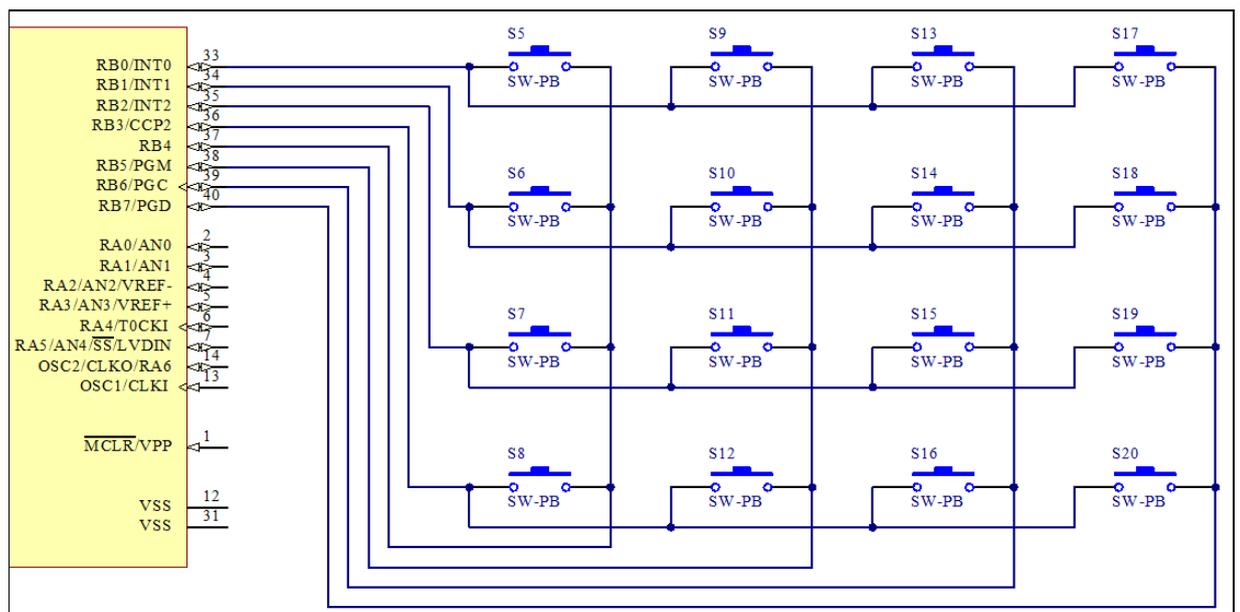


Figura 4.48. Conexión parcial del teclado matricial al PIC18F4550

Configurar los pines RB0, RB1, RB2 y RB3 como salidas, y los pines RB4, RB5, RB6 y RB7 como entradas, esta configuración será utilizada para manipular el funcionamiento del teclado, para conseguir esto modificamos el registro TRISB.

Cuando utilizamos el software ProtonIDE, el programador no tiene que modificar el registro TRISB, Proton IDE lo hace de manera automática al digitar el siguiente comando:

```
Declare Keypad_Port PORTB
```

El comando *INKEY* escanea el teclado en busca de la tecla pulsada y el valor lo guarda en una variable. *INKEY* devuelve un valor entre 0 y 16, si no se pulsa ninguna tecla el valor devuelto es 16. En la siguiente subrutina se muestra la lectura del teclado y el almacenamiento de los valores en las variables correspondientes.

Dato=*InKey* guarda un valor entre 0 y 16 en la variable Dato.

```
Select Dato
Case 0
tecla = "D"
valor=13
Case 1
tecla = "C"
valor=12
Case 2
tecla = "B"
valor=11
Case 3
tecla = "A"
valor=10
Case 4
tecla = "#"
valor=255
Case 5
tecla = "9"
valor=9
Case 6
tecla = "6"
valor=6
Case 7
tecla = "3"
```

```
valor=3
Case 8
tecla = "0"
valor=0
Case 9
tecla = "8"
valor=8
Case 10
tecla = "5"
valor=5
Case 11
tecla = "2"
valor=2
Case 12
tecla = "*"
valor=15
End Select
```

Subrutina para guardar el valor DECIMAL y ASCII de cada tecla pulsada.

Con los valores guardados en las variables *tecla* y *valor* se ordena al uC18 realizar la función programada para cada tecla.

#### 4.10. Conexión y configuración del LCD Matricial 4x20

Para comunicarse con la pantalla LCD podemos hacerlo por medio de sus pines de entrada de dos maneras posibles, con bus de 4 bits o con bus de 8 bits. En la siguiente figura vemos la forma de conectar el LCD al uC 18F4550.

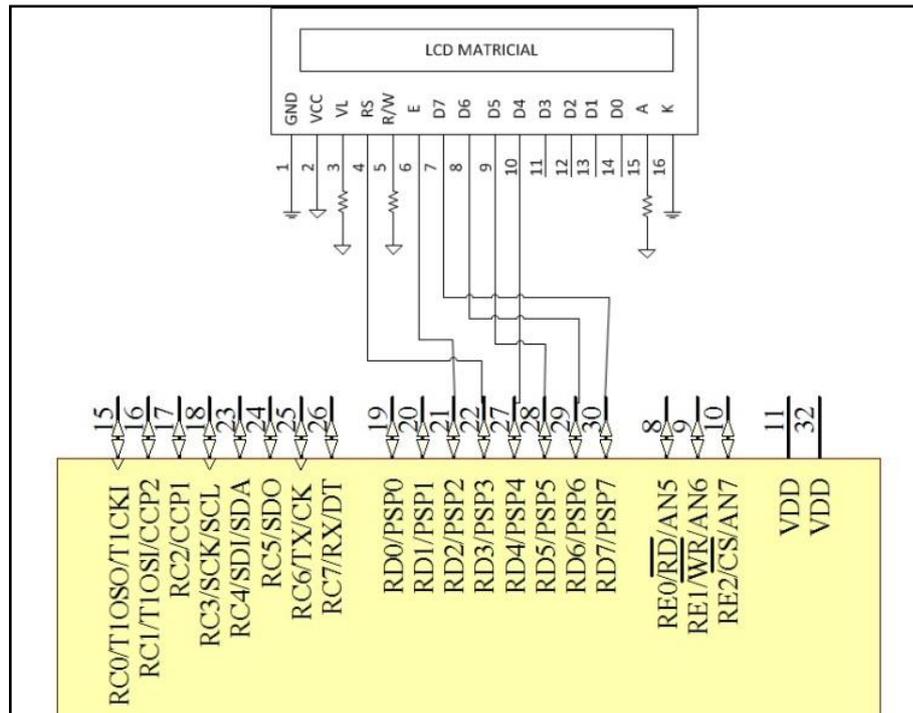


Figura 4.49. Conexión del LCD al uC 18F4550

La secuencia de inicialización para cualquier módulo LCD es imprescindible, y debe ser operada en la pantalla del módulo. La inicialización se basa en una serie de instrucciones introducidas por nosotros y posteriormente procesadas por el módulo LCD para su funcionamiento normal. Las instrucciones que están dentro de la inicialización solamente se ejecutan después que se enciende el modulo LCD y no podrán ser cambiadas posteriormente.

Ejemplos de instrucciones que solo podrán ejecutarse cuando inicializamos el módulo LCD:

- Selección de la longitud del bus de datos (4 Bits / 8 Bits).
- Activar el número de líneas que se visualizaran en el módulo LCD.
- Encender el Modulo LCD.

Con ProtonIDE basta dar las siguientes instrucciones para inicializar el LCD matricial con los parámetros a ocupar.

```

Declare LCD_DTPin = PORTD.4
Declare LCD_RSPin = PORTC.0
Declare LCD_ENPin = PORTC.1
Declare LCD_Interface = 4
Declare LCD_Lines = 4
Declare LCD_Type = 0
    Declare LCD_DTPin = PORTD.4:

```

Instrucción para inicializar el LCD matricial 4x20

La pantalla LCD puede ser conectada al uC utilizando un bus de 4 bits o un bus de 8 bits. Si se utiliza un bus de 8 bits, los 8 bits del puerto deben estar conectados. Si se utiliza un bus de 4 bits, el puerto debe ser conectado a la parte alta o baja de los pines de datos del LCD.

Para el proyecto utilizamos cuatro bits de datos, los datos empiezan desde el pin 4 del puerto D hasta el pin 7 del mismo.

```

Declare LCD_RSPin = PORTC.0

```

Indica la conexión del pin RS del LCD relacionada con el microcontrolador. El pin RS será conectado al pin 0 del puerto C. (Ver Tabla 3.3).

```

Declare LCD_ENPin = PORTC.1

```

Indica la conexión del pin E del LCD relacionada con el microcontrolador. En este caso el pin E será conectado al pin 1 del puerto C. (Ver Tabla 3.3).

```

Declare LCD_Interface = 4:

```

Esta instrucción informa al compilador si se va a utilizar la interfaz de 4 u 8 líneas para los datos. En este caso se configura 4 líneas de datos. Si no se escribe este comando el compilador por defecto configura para 4 líneas de datos.

```
Declare LCD_Lines = 4
```

Este comando informa al compilador el número de líneas que tiene el LCD, en el proyecto se utiliza un LCD de 4 líneas por lo que se configura de esta manera.

```
Declare LCD_Type = 0
```

Indica al compilador el tipo de LCD que será conectado, puede ser un LCD gráfico, alfanumérico o dependerá del fabricante de LCD. En nuestro caso utilizaremos un LCD alfanumérico que se configura con un número 0.

El comando para escribir en el LCD es **-Print AT ypos,xpos "texto"**- ejemplo:

```
Print At 1,1,"Iniciando..."
```

Este comando empieza a escribir en la fila 1 y columna 1 la palabra *Iniciando....*

#### 4.11. Control del Sistema de alarma

El sistema de alarma consta de 4 zonas preprogramadas que entran en funcionamiento cuando el usuario active la alarma en modo presente o ausente.

Tipos de Zona:

Zona 1: Entrada/salida, da un tiempo para que el usuario entre a la vivienda a desactivar el sistema de alarma mediante el teclado de control. En caso que el usuario se demore más del tiempo programado, la zona de entrada/salida generará una alarma, activando la sirena o luz indicadora.

Zona 2: Seguimiento, da tiempo de entrada, siempre y cuando primero se haya activado la zona de entrada/salida. En caso de activarse primero la zona de seguimiento, ésta genera un aviso de alarma instantáneo.

Zona 3: Presente/ausente, entra en funcionamiento solamente cuando el usuario activa la alarma en modo ausente, en caso de que la alarma es activada en modo presente, esta zona es anulada. Funciona también como una zona de seguimiento.

Zona 4: Perimetral, esta dará una alarma instantánea.

El sistema de alarma tiene tres estados: activado modo presente, activado modo ausente y desactivado. Estos estados son programados por el usuario de forma local mediante el teclado matricial o de forma remota mediante mensajes cortos de texto.

Los accesorios conectados al sistema de alarma son: sensores de movimiento, sensores de apertura, sirena, luz indicadora de alarma.

#### 4.11.1 Conexión de Sensores al Sistema de Alarma

Un sensor de movimiento o de apertura tiene un contacto que en su mayoría son normalmente cerrados (NC), estos contactos cambian de estado cuando el sensor detecta movimiento o apertura de puertas o ventanas. En la siguiente figura se muestra el diagrama de conexión de los contactos de los sensores al PIC16F628A que controla el sistema de alarma.

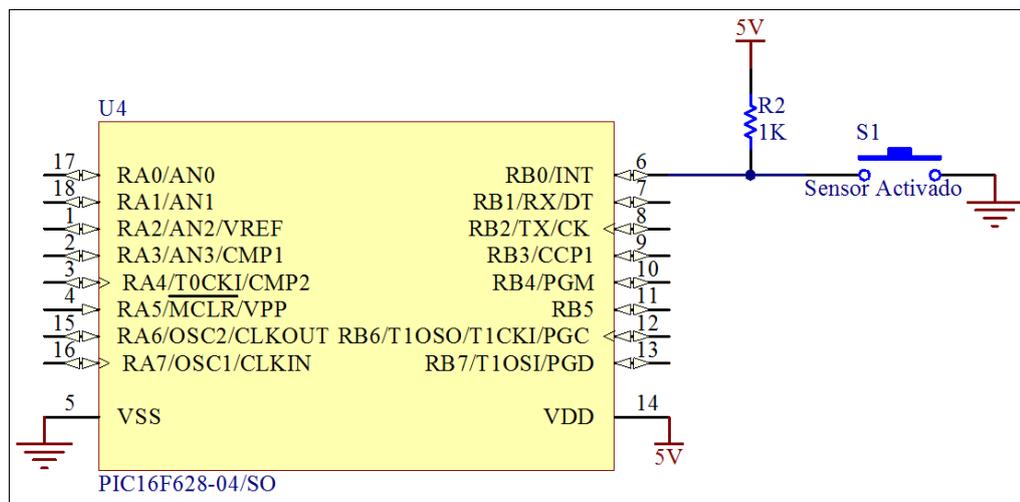


Figura 4.50. Conexión del Contacto del Sensor al PIC16F628A

En la figura 4.50 se muestra la conexión de la Zona 4 que es un sensor de apertura, en este caso el detector está activado por lo que el contacto está abierto ingresando un 1 lógico al PORTB.0. Cuando el sensor pase a su estado normal el contacto S1 se cierra ingresando un 0 lógico al PORTB.0.

Para el resto de zonas la conexión es la misma, solamente cambiando los puertos, para Zona 1: PORTA.4, Zona 2: PORTA.3 y Zona 3: PORTA.2.

### 4.11.2 Conexión de Sirena al Sistema de Alarma

La sirena es un indicador audible que se activa cuando las zonas sean violentadas mientras el sistema de alarma está activado en modo presente o ausente. La conexión de sirena se muestra en la Figura siguiente.

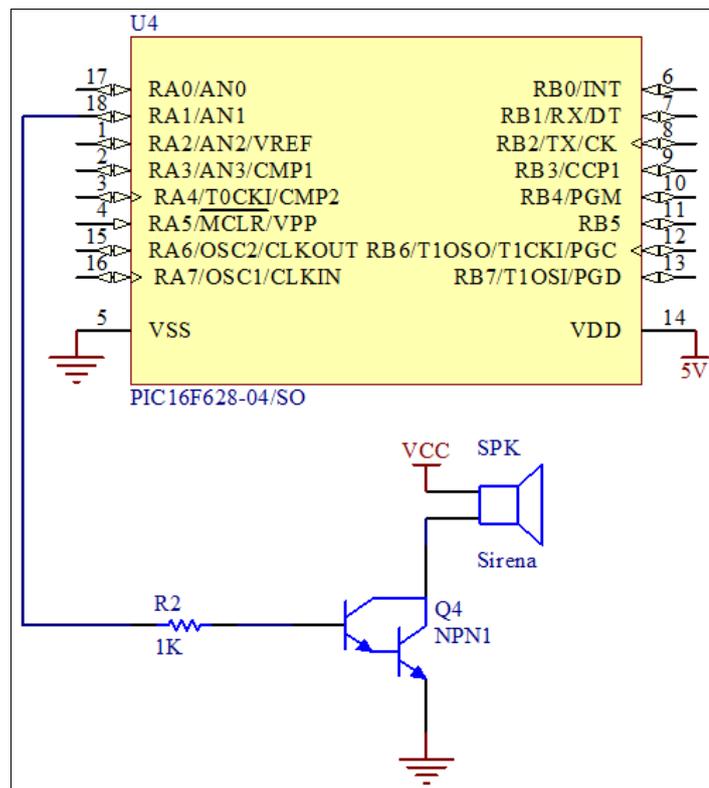


Figura 4.51. Conexión de sirena al sistema de alarma.

### 4.11.3. Activación modo presente

Es un estado de activación, que permite que el usuario se quede dentro de casa luego de activar la alarma, anulando todos los sensores interiores o las zonas programadas como presente/ausente. De esta manera el usuario puede caminar dentro de la casa sin activar la sirena del sistema.

El sistema de alarma puede ser activado en *modo presente* mediante el teclado de control o un SMS. El comando para este tipo de activación es "AAP" (Capítulo3 sección 3.3.4). Este dato es enviado en forma serial desde el uC18 hasta el uC16 el cual lee el comando y activa la alarma en el modo presente.

En el siguiente ejemplo se muestra la activación en modo presente:

```

inicio:
SerIn rx,16468, t, inicio,[D_Ing1]
SerIn rx,16468, t, inicio,[D_Ing2]

```

Lectura de datos en forma serial.

```

Select D_Ing1
  Case "A"
    Select D_Ing2
      Case "A"
        GoTo act_aus
      Case "P"
        GoTo act_pre
    End Select
  GoTo inicio

```

Filtrado de comando AAP.

```

act_pre:
d1tx = "P"
DelayMS 1
act=1
If ctrl1 = 1 And ctrl2 = 0 Then GoSub dat_ing
  If z1=0 Then bz1=1
  If z1=1 And bz1=1 Then GoSub temp_sir
  If z2=0 Then bz2=1
  If z2=1 And bz2=1 Then GoSub temp_sir
  If z4=0 Then bz4=1
  If z4=1 And bz4=1 Then GoSub temp_sir
GoTo act_pre

```

Subrutina de activación en modo presente.

Los datos leídos en forma serial, son comparados para identificar el tipo de activación, en este ejemplo al filtrar el comando ingresado *AAP* nos direcciona a la subrutina *act\_pre* en la cual se testea todas las zonas excepto la zona 3 que es una zona presente/ausente.

En este modo de activación todas las zonas se convierten en zonas instantáneas, con el cambio de estado de cualquier zona se genera una alarma activando la sirena o luz indicadora y enviando un reporte de alarma a un teléfono móvil programado. Para desactivar la alarma el usuario tiene que ingresar la clave de cuatro dígitos programada en el sistema.

#### **4.11.4. Activación Modo Ausente**

Es un modo de activación que pone en funcionamiento todas las zonas del sistema de alarma, se utiliza cuando el usuario sale de la vivienda, el sistema de alarma da un tiempo de salida necesario para que todas las personas desocupen la vivienda después de que se haya dado la orden de activación. Luego de haber transcurrido el tiempo de salida, todas las zonas empiezan a ser testeadas por el uC16 en espera de algún cambio de estado.

En caso que la zona 1 cambie de estado, el programa da un tiempo al usuario para desactivar el sistema de alarma, además las zonas de seguimiento y presente/ausente se convierten en zonas temporizadas, permitiendo llegar hasta el teclado sin generar alarmas hasta que el tiempo de entrada se termine.

Cuando el usuario no desactiva el sistema de alarma en el tiempo programado, el sistema genera una alarma y envía un reporte con todas las zonas que cambiaron de estado.

El usuario puede activar la alarma en *modo ausente* de manera local mediante el teclado y de forma remota mediante un SMS.

El comando para este tipo de activación es "AAA". El uC16 lee el dato recibido y ejecuta la subrutina siguiente:

```

act_aus:
dltx = "A"
cbip = 0

Repeat
If cbip = 250 Then bip = 1
If cbip = 500 Then
bip = 0 cbip = 0
End If

Inc cbip
DelayMS 1
Inc tsal

If ctrl1 = 1 And ctrl2 = 0 Then GoSub dat_ing
Until tsal=5000

bip = 0

act_aus1:
act=1
If ctrl1 = 1 And ctrl2 = 0 Then GoSub
dat_ing
If z1=0 Then bz1=1
If z1=1 And bz1=1 Then GoSub temp_ent

If z2=0 Then bz2=1
If z2=1 And bz2=1 Then GoSub temp_sir

If z3=0 Then bz3=1
If z3=1 And bz3=1 Then GoSub temp_sir

If z4=0 Then bz4=1
If z4=1 And bz4=1 Then GoSub temp_sir
GoTo act_aus1

```

Subrutina para la activación en modo ausente.

En la primera subrutina (act\_aus:) el uC16 solo genera un sonido que indica el tiempo de salida, una vez cumplido este tiempo el programa pasa a la subrutina *act\_aus1* en donde se testea las zonas 2, 3 y 4 esperando un cambio de estado en una de ellas para generar una alarma. Con el cambio de estado en la zona 1 el programa va a la subrutina de tiempo de entrada (temp\_ent).

```
temp_ent:
tent = 0
cbip = 0

Repeat
If z1 = 1 Then z1_1 = 1
If z2 = 1 Then z2_2 = 1
If z3 = 1 Then z3_3 = 1

If cbip = 250 Then bip = 1
    If cbip = 500 Then
        bip = 0
        cbip = 0
    End If
Inc cbip
DelayMS 1

Inc tent

If ctrl1 = 1 And ctrl2 = 0 Then GoSub dat_ing
If z4=1 Then GoSub temp_sir

Until tent = 5000
cbip = 0
GoSub temp_sir
Return
```

Subrutina para el tiempo de entrada

En la subrutina mostrada el programa va guardando en las variables `zx_x` las zonas que cambiaron de estado para que en caso de agotarse el tiempo de entrada sin que el usuario haya ingresado la clave correcta en el sistema pueda enviar un reporte con todas las zonas violentadas.

#### 4.11.5. Desactivación de Alarma

La desactivación de alarma consiste simplemente en no testear las zonas del sistema de alarma, en este caso solo se recibe datos desde el uC18 para posibles reportes o activación de alarma en los diferentes modos: presente o ausente.

```
desact:

z1_1 = 0
z2_2 = 0
z3_3 = 0

bz1=1
bz2=1
bz3=1
bz4=1

dltx = "D"
act=0
sir = 0
bip = 0
GoTo inicio
```

Subrutina de desactivación de alarma.

En la subrutina de desactivación el programa regresa todas las variables a su estado inicial para poder ser ocupadas nuevamente en cualquier evento.

#### 4.11.6. Envío de Reporte de Alarma

Cada vez que el sistema de alarma sea violentado mientras está activado en uno de los dos modos (ausente o presente), genera un reporte de alarma que es enviado a un teléfono móvil programado por el usuario, esto ocurre siempre en la subrutina de tiempo de sirena que es cuando se ha generado una alarma por el cambio de estado de cualquier zona.

```

temp_sir:
tsir = 0
sir = 1
If z1_act = 1 Then aux1 ="A"
If z2_act = 1 Then aux2 ="A"
If z3_act = 1 Then aux3 ="A"
If z4_act = 1 Then aux4 ="A"

        DelayMS 1000
        SerOut tx,16468,[aux1]
        DelayMS 20
        SerOut tx,16468,[aux2]
        DelayMS 20
        SerOut tx,16468,[aux3]
        DelayMS 20
        SerOut tx,16468,[aux4]
        DelayMS 50

Repeat
DelayMS 1
Inc tsir
If ctrl1 = 1 And ctrl2 = 0 Then GoSub dat_ing
aux_z = PORTB & %1111
If aux_z > aux Then GoTo temp_sir
Until tsir = 5000
sir = 0
tent = 0
Return

```

Subrutina de tiempo de sirena

Dependiendo de la zona que se haya activado, el programa carga las variables *aux* con la letra "A" y son enviadas al uC18 que es que genera el mensaje de texto para ser enviado al teléfono móvil programado. En el tiempo de sirena el programa testea también posibles cambios de estado en otras zonas, en este caso el programa vuelve a enviar un reporte nuevo de alarma.

El uC18 es el encargado de generar un mensaje de texto con los datos enviados por el uC16. A continuación se muestra la subrutina en el PIC 18F4550 para el envío de mensajes de texto.

```

envia_rep:

DelayMS 50
SerIn rx_a, 16468, [d1rx]
SerIn rx_a, 16468, [d2rx]
SerIn rx_a, 16468, [d3rx]
SerIn rx_a, 16468, [d4rx]
DelayMS 50

If d1rx = "A" Then Aux1 = "Zona1. "
If d2rx = "A" Then Aux2 = "Zona2. "
If d3rx = "A" Then Aux3 = "Zona3. "
If d4rx = "A" Then Aux4 = "Zona4. "

SerOut tx_sim900,16468,["AT+CMGS=",34,"+593",numr,34,13]
DelayMS 1000
SerOut tx_sim900,16468,["Reporte de Alarma",13,"Alarma
Activada: ",Aux1,Aux2,Aux3,Aux4,26,13]

      Cls
      Print At 1,1,"Mensaje ENVIADO "
      DelayMS 2000
      Cls
GoSub Recibe

```

Subrutina para envío de mensaje de texto

El uC18 recibe los datos en forma serial enviados por el uC16 del sistema de alarma, dependiendo de los datos recibidos se cargan nuevas variables *Aux* y se da la orden al módulo SIM900 mediante el comando **"AT+CMGS="** que envíe un mensaje de texto con los datos cargados en las variables *Aux* al número de reporte guardado en la variable *numr*.

#### 4.11.7 Envío de Reporte de Estado de Alarma.

El usuario puede pedir un reporte del estado de alarma mediante un mensaje de texto, el comando enviado en el SMS es *RA*, esto hace que el sistema de alarma envíe un mensaje con el estado de alarma el cual puede ser: Alarma Activada en Modo Presente, Alarma Activada en Modo Ausente o Alarma Desactivada.

Igual que en el reporte de alarma, el uC16 envía datos con información del estado del sistema de alarma en forma serial al uC18, éste genera un mensaje de texto similar al de Reporte de Alarma.

```

Select d1rx
    Case "A"
        Aux1 = "Activada "
        Aux2 = "Modo "
        Aux3 = "Ausente."
    Case "D"
        Aux1 = "Desact"
        Aux2 = "ivada"
        Aux3 = "."
    End Select
Cls
Print At 1,1, "Enviando REPORTE"
SerOut tx_sim900,16468,["AT+CMGS=",34,"+",Num,34,13
SerOut tx_sim900,16468,["ALARMA",13,"Alarma:
",Aux1,Aux2,Aux3,26,13]
    Cls
    Print At 1,1,"Mensaje ENVIADO "
    DelayMS 2000
    Cls
    GoSub Borrando

```

Subrutina para el envío de un SMS con el Estado de Alarma.

Según el dato recibido en *d1rx* el PIC18F4550 ordena al módulo SIM900 enviar un mensaje de texto con los valores cargados y al número de celular programado.

#### 4.12. Control del Sistema de Iluminación

El sistema de iluminación controlará el dimerizado de lámparas led mediante PWM generado por el uC 16F628A.

##### 4.12.1 PWM: Frecuencia y duty cycle

La técnica PWM (Pulse-Width Modulation) o modulación por anchura del pulso permite generar ondas cuadradas con una frecuencia y ciclo de actividad determinada.

El módulo PWM en el PIC 16f628A es controlado por el registro CCP1CON el que se configura de la siguiente manera:

##### Registro CC1CON

	-	-	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TXST A	-	-	CCP1 X	CCP1 Y	CCP1M 3	CCP1M 2	CCP1M 1	CCP1M 0
	Bit 7	Bit 6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

**Bit 7-6:** No utilizado

**Bit 5-4 CCP1X:CCP1Y:** bits menos significativos del duty cycle del PWM

**Bit 0-35 CCP1M<3:0>:** selección del modo de CCPx

11xx: para modo PWM

Calculo del periodo de PWM:

$$T_{pwm} = [(PR2) + 1] * 4 * T_{osc} * TMR2 \text{ prescale}$$

Calculo del Duty Cycle del PWM o ciclo de trabajo.

$$DC_{pwm} = (CCPR1:CCP1CON < 5:4 >) * T_{osc} * TMR2 \text{ prescale}$$

### Configuración de PWM

Escribir en el PR2 para seleccionar el periodo del PWM.

Escribir en el CCPR1L y CCP1CON<5:4> para obtener el D.C. del PWM.

Configurar la salida CCP1 como pin de salida.

Escribir en el TMR2 el valor del preescaler y habilitar el Timer 2 escribiendo en el T2CON.

### Cálculos:

Frecuencia PIC = 4 MHz ;  $T_{osc} = 2,5 * 10^{-7}$

Frecuencia del PWM = 1 KHz;  $T_{pwm} = 0,001 \text{ seg.}$

$$PR2 = \left( \frac{T_{pwm}}{4 * T_{osc} * TMR2 \text{ prescale}} \right) - 1$$

$$CCPR1L:CCP1CON < 5:4 > = \left( \frac{DC \text{ pwm}}{T_{osc} * TMR2 \text{ prescale}} \right) - 1$$

De esta manera se puede obtener los valores de periodo y ciclo de trabajo para PWM.

#### 4.12.2 Control de Iluminación (PWM) con ProtonIDE

Para generar PWM en ProtonIDE basta con utilizar el comando *PWM*, el cual configurará todos los registros explicados anteriormente, de la siguiente manera.

```
PWM Pin , Duty , Cycles
```

**Pin:** es el pin por donde saldrá la señal de PWM.

**Duty:** es un valor entre 0 y 255 que indica el nivel de voltaje promedio que tendrá la señal de PWM.

**Cycles:** es un valor entre 0 y 255 que indica el número de ciclos que saldrán por el pin seleccionado para PWM. Con un cristal de 4MHz aproximadamente cada ciclo dura 5ms.

Ejemplo de PWM:

```
PWM PORTB.5,255,1
```

Sentencia para generar PWM

Si esto lo hacemos repetidas veces se está generando un PWM por el pin 5 del puerto B con un promedio de 5V.

```
PWM s1,duty1,t
```

Sentencia para generar PWM.

En este ejemplo s1 es el pin de un determinado puerto, el duty1 se puede cambiar haciendo que las lámparas led del sistema de Iluminación sean encendidas con diferentes niveles de voltaje dependiendo de los datos enviados por el uC 18F4550 como se explicó en el capítulo 3 sección 3.4.2. (Comunicación para el sistema de iluminación).

Cada lámpara led está controlada por un puerto (PORTB.4 y PORTB.5), a cada puerto está asociado una variable *duty1* y *duty2* respectivamente. Cada vez que el usuario cambia el nivel de iluminación de una lámpara mediante el uC18 (teclado de control o SMS), lo único que cambia son los valores de las variables *duty1* o *duty2*.

Además el usuario puede cambiar el nivel de luminosidad de cada lámpara mediante pulsadores instalados en cada habitación que se quiera hacer el control y monitoreo, siempre serán dos botones en cada habitación, uno para subir la intensidad hasta el nivel máximo y otro para bajar la intensidad hasta apagar la lámpara.

En el siguiente ejemplo se muestra el código escrito en ProtonIDE para encender las dos lámparas al nivel máximo mediante los pulsadores.

```

Symbol up1 = PORTB.0
duty1 = 0
inicio:
PWM s1,duty1,t
If up1 = 1 Then b_puls = 0
If up1 = 0 And b_puls = 0 Then GoSub subel
Goto inicio

subel:
b_puls=1
Select duty1
  Case a
    duty1 = a
  Case b
    duty1 = a
  Case c
    duty1 = b
  Case d
    duty1 = c
  Case e
    duty1 = d
End Select
GoTo inicio

```

Ejemplo para subir la intensidad de luz mediante pulsadores.

Para el ejemplo, el pulsador llamado *up1* es conectado al pin 0 del puerto B se genera PWM por el pin 5 del mismo puerto. Al accionar el pulsante *up1*, el estado del PORTB.0 cambia a un nivel lógico 0 haciendo que el programa ejecute la subrutina *sube1* y dependiendo del valor con el que anteriormente haya estado cargado *duty1*, la subrutina cambiará a un nivel superior, de esta manera cambia el nivel de luminosidad hasta llegar a un nivel máximo (*duty1* = a, donde a = 255).

En el siguiente ejemplo se muestra cómo apagar la lámpara led número 2 mediante el teclado matricial o mensaje de texto.

Cuando se recibe un SMS con el comando para apagar la lámpara 2 (Capítulo 3 sección 3.3.4 Filtrado del SMS) o mediante el teclado se ordena apagar la lámpara, el uC18 envía en forma serial al uC16 los caracteres "20" que indican el número de lámpara (2) y el nivel de luminosidad (0). El uC16 lee esos datos y cambia los valores del *duty1* o *duty2* dependiendo del caso.

```

SerIn rx,16468, tiempo, inicio,[D_Ing1]
SerIn rx,16468, tiempo, inicio,[D_Ing2]
Select D_Ing1

    Case "2"
        Select D_Ing2
            Case "A"
                duty2=a
            Case "B"
                duty2=b
            Case "C"
                duty2=c
            Case "D"
                duty2=d
            Case "0"
                duty2=e
        End Select
inicio:
PWM s2,duty2,t
Goto inicio

```

Ejemplo para apagar la lámpara 2 mediante teclado o SMS.

Los datos son ingresados en forma serial, estos son comparados para cargar la variable *duty2* con el valor deseado. En el ejemplo el dato leído en forma serial es "20" el cual hace que *duty2* se cargue con el valor de "e" (e=0), apagando la lámpara led número 2.

#### 4.12.3 Conexión de Lámparas Led al Sistema de Iluminación.

Las lámparas led son conectadas a los pines que generan el PWM (PORTB.4 y PORTB.5) como se muestra en la figura 4.52.

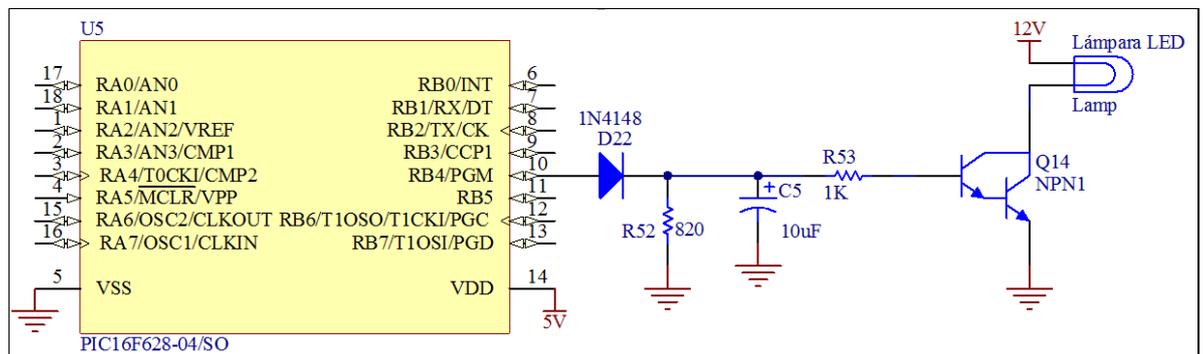


Figura 4.52. Conexión de lámparas led al PIC 16F628A

El PWM pasa a través del diodo rápido 1N4148 cargando al condensador C5 el cual mediante la resistencia de base conectada al transistor Darlington Q14, enciende la lámpara led según el PWM que genere el PORTB.4.

R52 es una resistencia de 820Ω que descargará al condensador en los flancos de bajada del PWM, el diodo 1N4148 conectado en serie impide que ingresen corrientes al PORTB.4 mientras se descarga completamente el condensador C5.

#### 4.12.4 Envío de Reporte del Estado de Iluminación

Igual que en el sistema de alarma el usuario puede pedir un reporte del sistema de iluminación mediante un mensaje de texto y en este caso se puede hacer mediante el teclado de control.

El mensaje de texto que se enviara al módulo SIM900 para que devuelva un SMS con el estado de iluminación es RI, al recibir este comando el uC16 envía datos con la información del estado de iluminación en forma serial al uC18 que es el

encargado de generar el mensaje y dar la orden al módulo SIM900 de enviar el SMS.

La subrutina de envío de reporte es similar a la ocupada para el sistema de alarma. Ver sección 4.11.7.

#### 4.13. Control del Sistema de Climatización

Para el control de temperatura se utiliza el sensor DS18B20 que capta la temperatura del ambiente permitiendo activar niquelinas o ventiladores para subir o disminuir la temperatura en un área determinada. El usuario podrá controlar la temperatura mediante SMS o el teclado de control.

##### 4.13.1. Sensor de temperatura DS18B20



Figura 4.53. Sensor de Temperatura DS18B20.

Fuente: CHINAICMART. DS18B20. <http://www.chinaicmart.com/uploadfile/ic-doc/406-TO-92.jpg>.

[Consulta: 16 de Agosto de 2012]

El termómetro digital DS18B20 reporta grados Celsius con una precisión de 9 a 12 bits, -55 a 125°C (+/- 0,5°C). Este sensor comunica mediante un bus 1-Wire que, por definición, requiere solamente de una línea de datos y dos de alimentación (+5V y GND) para la comunicación con el microcontrolador.

Las aplicaciones que se pueden beneficiar de esta característica incluyen controles ambientales, sistemas de monitorización de la temperatura interior de los edificios, equipos o maquinaria, proceso de vigilancia y sistemas de control.

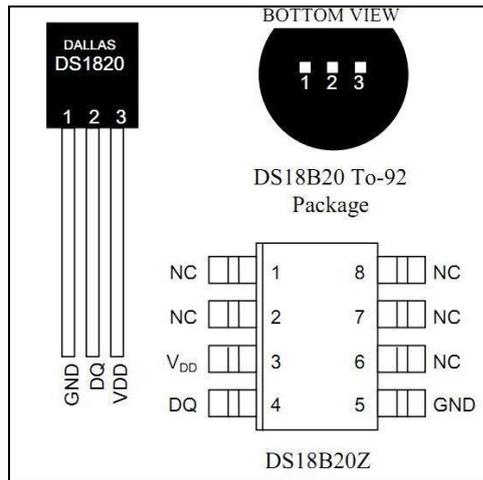


Figura 4.54. Pines del sensor DS18B20

Fuentes: ALIEXPRESS. Sensor 18B20.

[http://i00.i.aliimg.com/img/pb/680/492/429/429492680\\_662.jpg](http://i00.i.aliimg.com/img/pb/680/492/429/429492680_662.jpg).

**Características:**

Para su comunicación requiere una sola línea de conexión al microcontrolador.

No requiere componentes externos.

Rango de la fuente de alimentación 3.0 a 5.5V.

Medidas de temperaturas de -55C a 125C

Resolución seleccionable por el usuario de 9 a 12bits.

Convierte la temperatura (12bits) en 750ms (Max).

**Descripciones de los pines:**

Pin	Nombre	Función
1	GND	Tierra (-)
2	DQ	Entrada/Salida de datos
3	VDD	VDD Alimentación.

Tabla 4.4.Pines del sensor DS18B20

**Relación de temperatura:**

Temperatura	Salida Digital Binario	Salida Digital Hexadecimal
+125	0000 0111 1101 0000	07D0h
+85	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h

Tabla 4.5. Relación de temperatura

**4.13.2. Lectura de Temperatura con PIC 16F628A**

El programa ProtonIDE tiene instalado una librería que controla el protocolo DALLAS 1-WIRE. El protocolo *1-wire* tiene un estándar bien definido para las secuencias de transacciones. Cada secuencia de la transacción se compone de cuatro partes:

Inicialización.

Comando Función ROM

Comando Función de Memoria

Transacción de Datos

El comando función ROM y Memoria son siempre de 8 bits y ocupan los bits menos significativos. La inicialización se hace con un pulso generado por el microcontrolador y cada dispositivo colgado en el sistema responde con un pulso de presencia, el pulso de reset está controlado por los dos últimos bits.

```

captura:
OWrite DQ1, 1, [$CC, $44] ' comando para lectura de
temperatura
Repeat

If ctrl1 = 1 And ctrl2 = 1 Then GoSub dat_ing

DelayMS 25 'espera un tiempo para que la conversion sea
completada
ORead DQ1, 4, [C]          ' lee los datos
Until C <> 0              ' finaliza la lectura del DS18B20.
OWrite DQ1, 1, [$CC, $BE] 'comando indicando lectura de
datos
ORead DQ1, 2, [Temp1.LowByte,Temp1.HighByte] 'graba los
datos de temperature en Temp1 parte baja y parte alta.

```

Subrutina para lectura de temperatura.

En la variable Temp1 esta almacenada la temperatura, los datos son almacenados como se muestra en la tabla de relación de temperatura.

Los 4 primeros bits contienen el valor decimal de temperatura, desde el bit 4 hasta el bit 12 se almacena el valor entero de temperatura y los últimos 4 bits contienen el signo positivo o negativo. Ejemplo:

Temperatura	Variable Temp1	Salida Digital Hexadecimal
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h

El cambio de temperatura por cada bit es de 0.0625°C, para el primer valor mostrado en la tabla (últimos cuatro bits <13:16>) son 0000 que indica un valor positivo de temperatura. Si se lee solamente los bits <4:12> 00011001, transformando a decimal nos da un valor de 25. Los primeros bits <0:3> 0001, transformando a decimal nos da un valor de 1 y multiplicado por 0.0625 nos da el valor decimal de la temperatura. Teniendo como resultado +25.0625°C.

Si se sigue el mismo procedimiento para cada valor almacenado en Temp1 se obtiene la temperatura captada por el módulo DS18B20. En la siguiente figura se muestra la conexión del módulo DS18B20 con el PIC 16F628A.

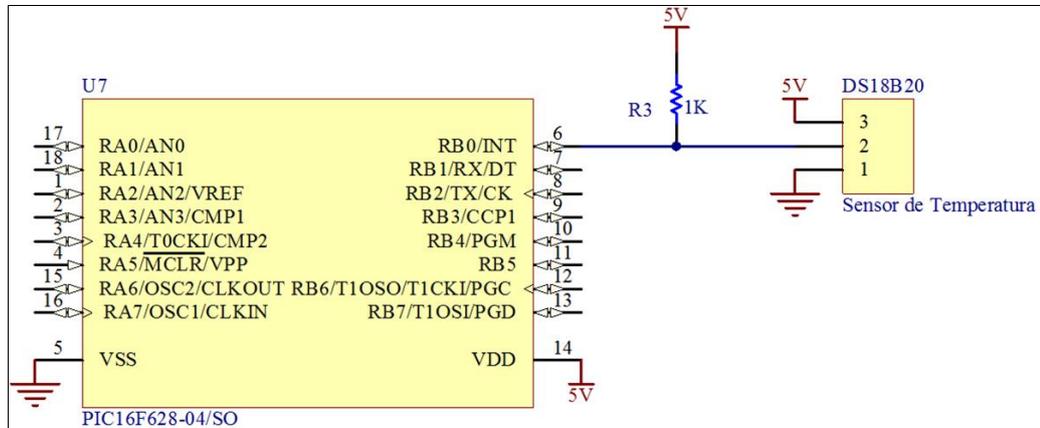


Figura 4.55. Conexión de módulo DS18B20 al microcontrolador

### 4.13.3. Control de Temperatura.

Para demostrar el funcionamiento del control de temperatura, se utilizará niquelinas y ventiladores conectados al microcontrolador mediante una interfaz de potencia, este sistema en realidad permite conectar cualquier tipo de aire acondicionado ya que el PIC16F628A (climatización) controla relés que soportan una corriente hasta de 3 Amperios.

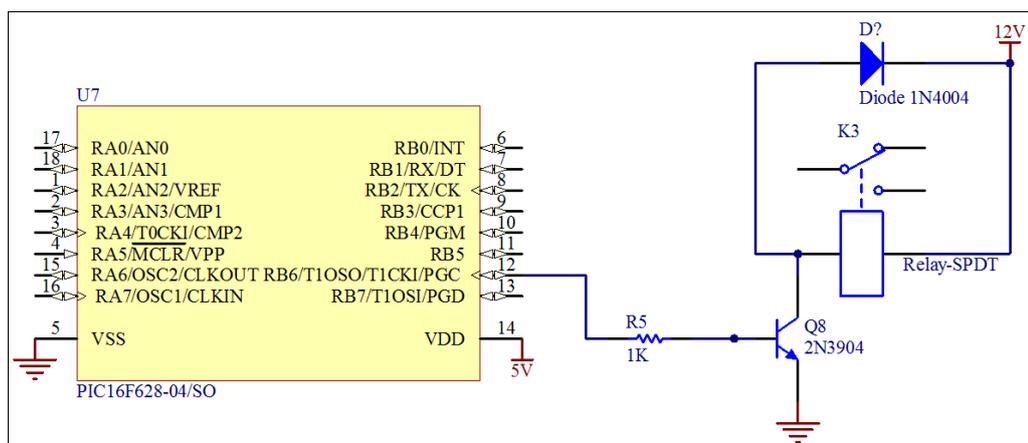


Figura 4.56. Interfaz de potencia para el control de un relé

El sistema de climatización controlará la temperatura de dos zonas por lo que se necesita comandar 4 relés, 2 para ventiladores y 2 para calefactores (niquelinas).

El circuito mostrado en la figura es para el control del relé que actuara cuando se necesite prender el calefactor de la Zona 2. El circuito es el mismo para ventiladores y calefactores.

El resistor R5 conjuntamente con el transistor Q8 son la interfaz de potencia que controla el relé K3, el diodo 1N4004 es de protección al circuito que evita que corrientes parásitas interfieran con el funcionamiento del circuito cuando el relé K3 vuelve a su estado normal.

La temperatura programada por el usuario en una determinada zona se mantiene constante con un error de +/- 1C. La subrutina de control se muestra a continuación:

```

TD1=Temp1/16
If TD1 = Tsets Then
niquels = 0
vents = 0
End If
If TD1 <= Tmins Then
niquels = 1
vents = 0
End If
If TD1 >= Tmaxs Then
vents = 1
niquels = 0
End If

```

Subrutina de control de temperatura.

TD1: temperatura captada por el sensor DS18B20.

Tsets: es la temperatura programada por el usuario

Tmins: temperatura mínima permitida  $Tmins = Tsets - 1$

Tmaxs: temperatura máxima permitida  $Tmaxs = Tsets + 1$

En la subrutina se puede observar que: si la temperatura TD1 es igual a Tsets, la niquelina y el ventilador se apagan, manteniendo la temperatura estable. Si TD1 es

menor o igual que  $T_{\min}$ , se enciende la niquelina o calefactor para calentar la Zona. Cuando  $TD1$  es mayor o igual a  $T_{\max}$  se encienden los ventiladores para bajar la temperatura de la Zona.

#### **4.14. Fuente de Poder.**

Es un dispositivo que convierte la tensión alterna de la red eléctrica a un voltaje de corriente prácticamente continua. Sirve para alimentar todos los circuitos que pertenecen al sistema.

En electrónica las fuentes de alimentación se clasifican en lineales y conmutadas. Una fuente de alimentación lineal consta de varias partes: Transformador, rectificador, filtro y regulador.

##### **4.14.1 Transformador.**

Se encarga de aumentar o disminuir la magnitud de voltaje que aparece en el devanado primario, sin que varíe esencialmente su forma de onda.

##### **4.14.2. Rectificador.**

La forma de onda salida del rectificador muestra que el voltaje es unidireccional pulsante y por tanto, cuenta con un valor promedio de CD y un contenido de armónicos que es necesario eliminar, a fin de obtener el voltaje directo puro.

##### **4.14.3. Filtro.**

La función del filtro es disminuir o eliminar los componentes de CA que existen en el voltaje de salida del rectificador y tal disminución dependerá mucho de las características del rectificador y de la calidad del filtro.

#### 4.14.4. Regulador de tensión.

Está determinado por las posibilidades que tenga el sistema alimentado de admitir las variaciones que experimenta el voltaje directo de salida cuando ocurren cambios en la corriente de la carga, es decir, cambios en el consumo de corriente de los circuitos que son alimentados por la fuente u otras variaciones.

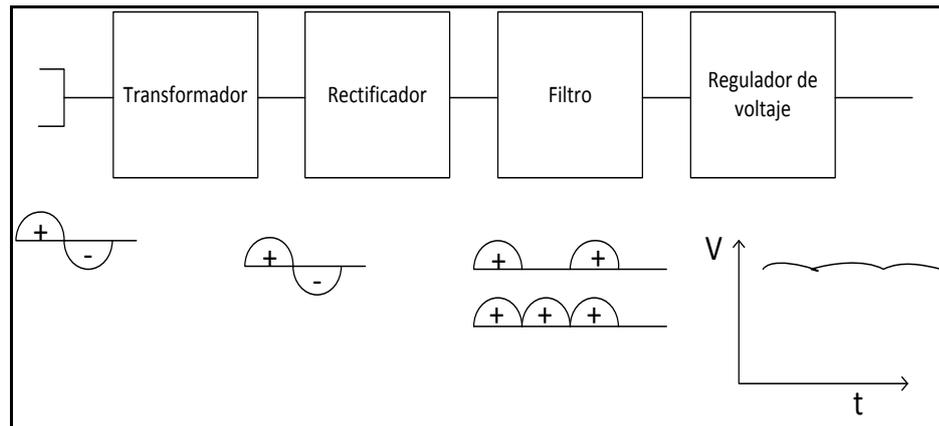


Figura 4.57. Diagrama de Fuente Básica



Figura 4.58. Fuente de Poder 12V 3A y transformador de 110VCA a 16,8VCA

## CAPÍTULO 5

### DISEÑO Y CONSTRUCCIÓN DE TARJETAS DE CIRCUITO IMPRESO

Para la construcción de las tarjetas de circuito impreso o PCB (printed circuit board) se parte de un circuito esquemático que tiene que ser probado para comprobar que todos los componentes que se vayan a soldar en el PCB sean los correctos.

#### 5.1 Circuito Esquemático

Un circuito esquemático o diagrama electrónico es una representación gráfica de un circuito electrónico. Muestra los diferentes componentes del circuito de manera simple y con gráficos uniformes de acuerdo a normas, las conexiones de alimentación y de señal entre los distintos dispositivos. El arreglo de los componentes e interconexiones en el esquema generalmente no corresponde a sus ubicaciones físicas en el dispositivo terminado.

A diferencia de un esquema de diagrama de bloques o disposición, un esquema de circuito muestra la conexión real mediante cables entre los dispositivos. (Aunque el esquema no tiene que corresponder necesariamente a lo que el circuito real aparenta). El tipo de dibujo que sí representa al circuito real se llama negativo (o positivo) de la tablilla de circuito impreso.

El diseño de un circuito esquemático se puede hacer utilizando cualquier tipo de software como: ALTIUM DESIGNER, NI MULTISIM, ORCAD, PROTEUS. En nuestro caso utilizaremos ALTIUM DESIGNER 2004 (DXP).

##### 5.1.1 Altium Designer

Altium Designer es un conjunto de programas para el diseño electrónico en todas sus fases y para todas las disciplinas, ya sean esquemas, simulación, diseño de circuitos impresos, implementación de FPGA, o desarrollo de código para microprocesadores.

No se trata de un conjunto de paquetes sueltos vendidos como una suite y conectados mediante archivos externos (netlist), sino de un programa único

(dxp.exe) que crea un entorno y comunica al usuario con los distintos servidores (por ejemplo, editor de texto, editor de esquemas, editor de PCB).

### 5.1.2. Circuito Esquemático en Altium Designer

1. Para la creación de un circuito esquemático damos click en las siguientes opciones: Add New to Project/Schematic. Figura 5.1.

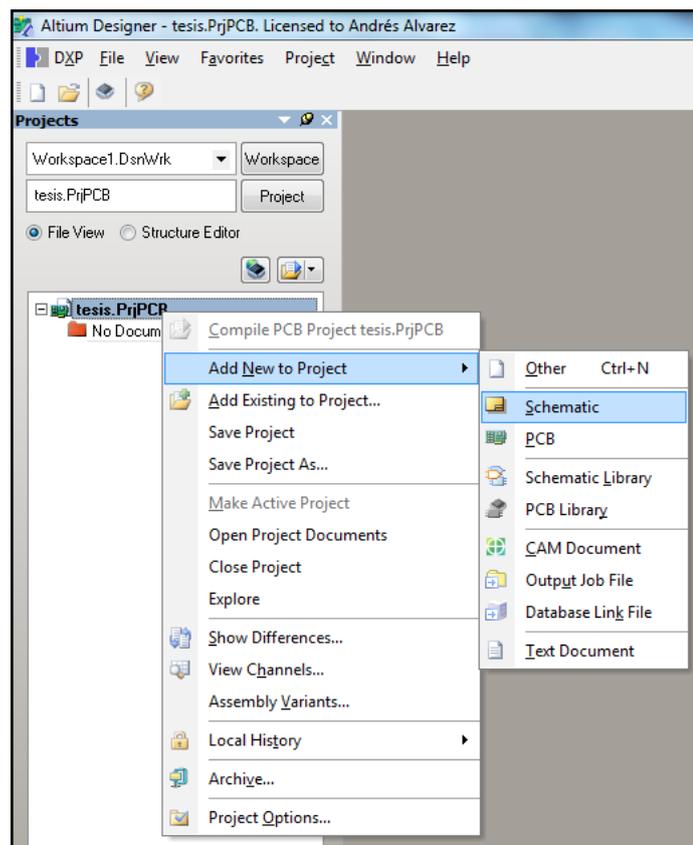


Figura 5.1. Nuevo circuito esquemático

Una hoja para dibujar el circuito esquemático en blanco llamada Sheet1.SchDoc aparece en la ventana de diseño y el documento esquemático es agregado automáticamente al proyecto (figura 5.2).

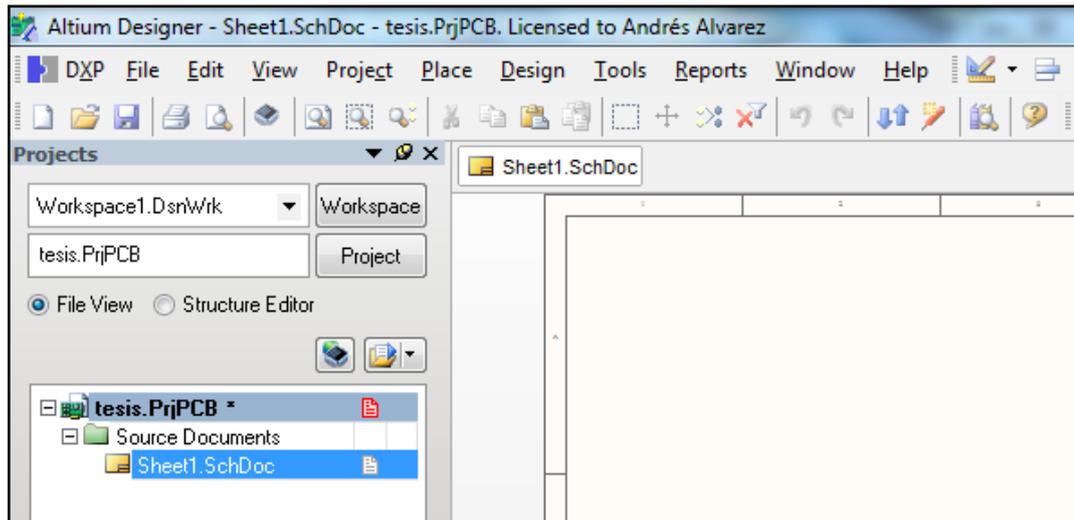


Figura 5.2. Hoja para dibujar el circuito esquemático

2. Cambiamos el nombre del circuito esquemático a tesis.SchDoc abriendo el menú principal (File/Save Project As).
3. Configuración del Archivo esquemático.

Desde el menú principal seleccionar *Design/Documents Options* y dentro del cuadro de dialogo se selecciona la etiqueta *Sheets Options* y la opción *Standard Style* para cambiar el tamaño del archivo esquemático. Figura 5.3.

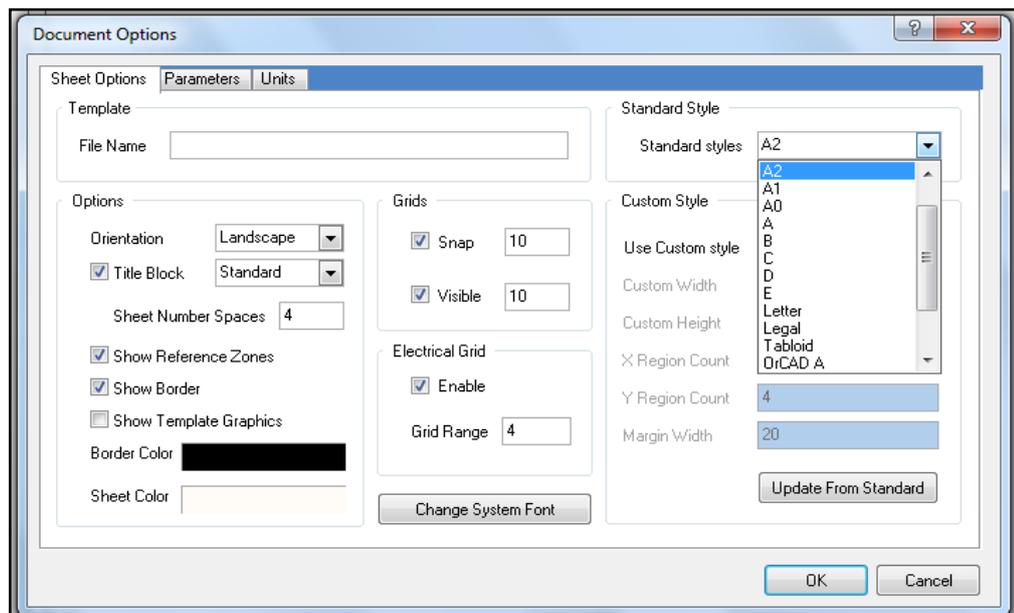


Figura 5.3. Tamaño de la hoja de archivo esquemático

### 5.1.3. Dibujando el circuito esquemático.

Para realizar este dibujo nos basamos en el circuito construido en el protoboard que se muestra en la Figura 5.4. El cual contiene la mayoría de componentes utilizados para el funcionamiento correcto del circuito electrónico.

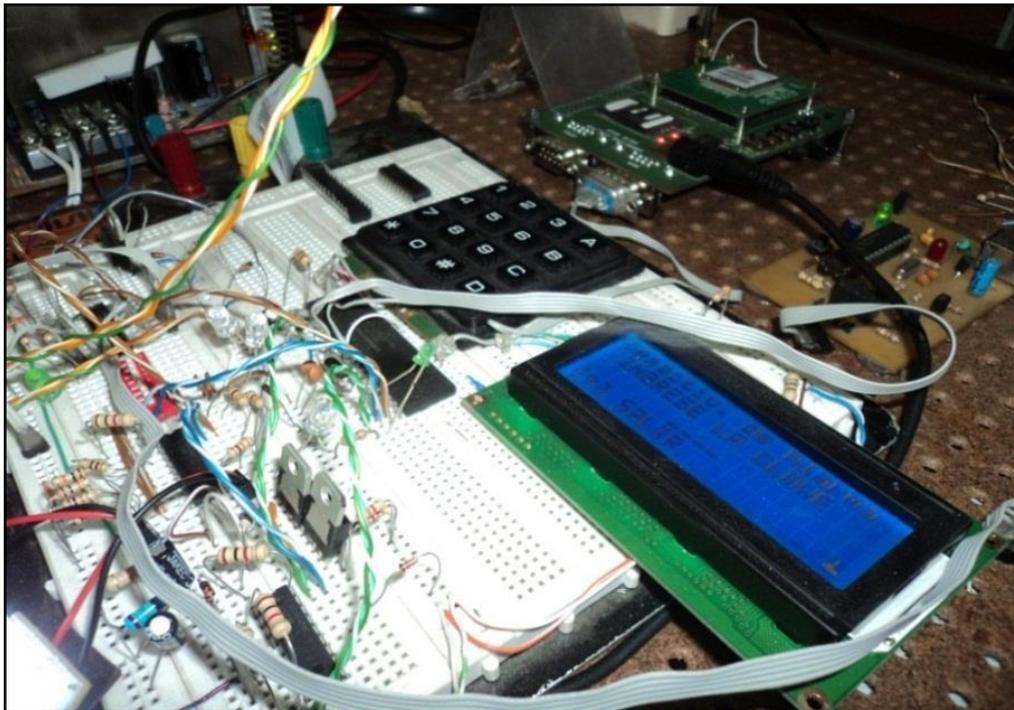


Figura 5.4. Circuito electrónico (protoboard)

1. Clic en la pestaña “Librerías” Figura 5.5. Para elegir los componentes que se van a utilizar como: resistencias, condensadores, diodos, circuitos integrados, transistores, conectores, etc. Algunos elementos que se necesitan no aparecen en las librerías cargadas por defecto en el programa. Instalamos nuevas librerías (Figura 5.6) para elementos especiales como los microcontroladores PIC 18F4550 y 16F628A.

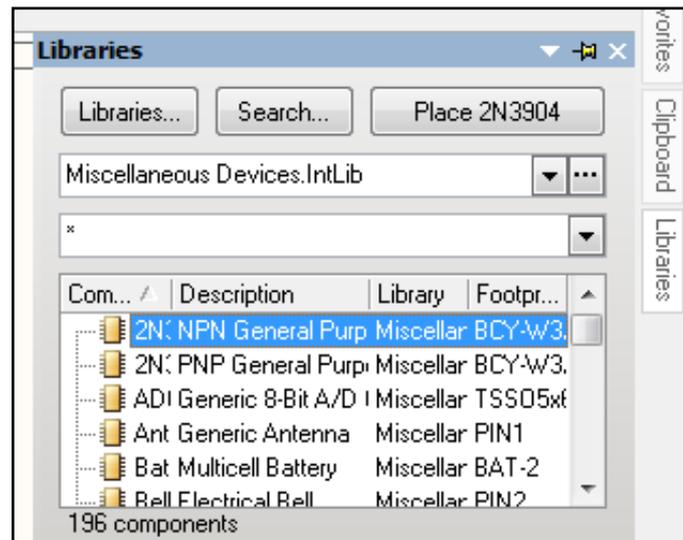


Figura 5.5. Librería en Altium Designer

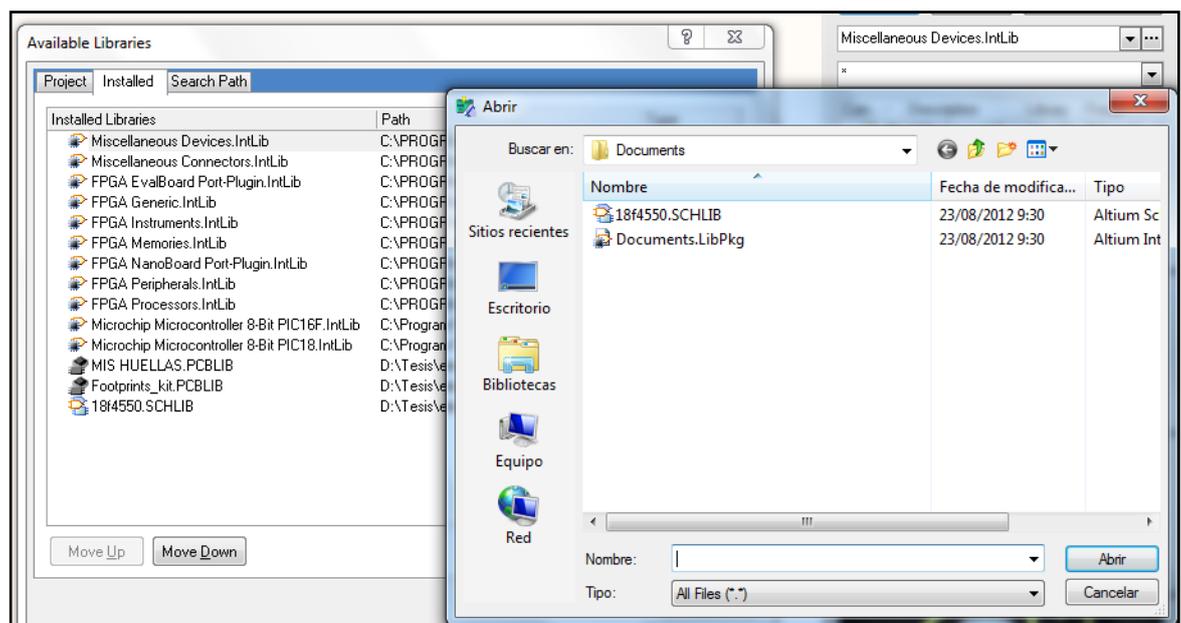
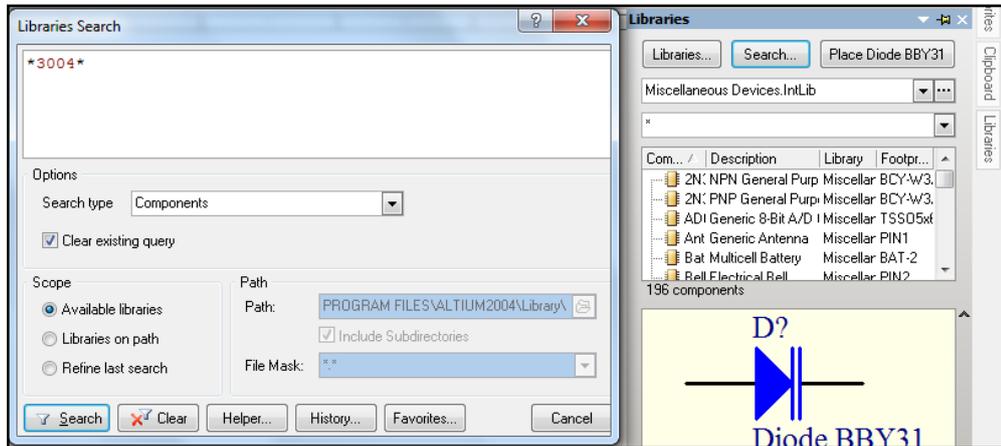


Figura 5.6. Instalación de nuevas librerías

- Para localizar los componentes escribimos una parte o el nombre completo del elemento a buscar en el campo de texto de la opción *Libraries Search* que aparece al dar "clic" en la pestaña *Search* del menú *Libraries* (Figura 5.7). Esta opción nos permite buscar todas las referencias que existan acerca de este componente. El símbolo "\*" indica al sistema que realice una búsqueda con todo lo relacionado con ese número, pasando por alto

marcas, tipos y otras características de los componentes. Si los parámetros de búsqueda fueron introducidos correctamente el sistema encontrará el componente.



5.7. Opción *Libraries Search*.

- Al seleccionar el componente del panel de librerías el cursor aparece con una cruz y el componente, permitiendo colocar el elemento en cualquier parte de la hoja del circuito esquemático. Editamos las propiedades del elemento (doble click en el elemento) para asegurarnos que sus parámetros son los correctos como: footprint (huella para circuito impreso), designator (nombre del elemento) y comentarios. Figura 5.8.

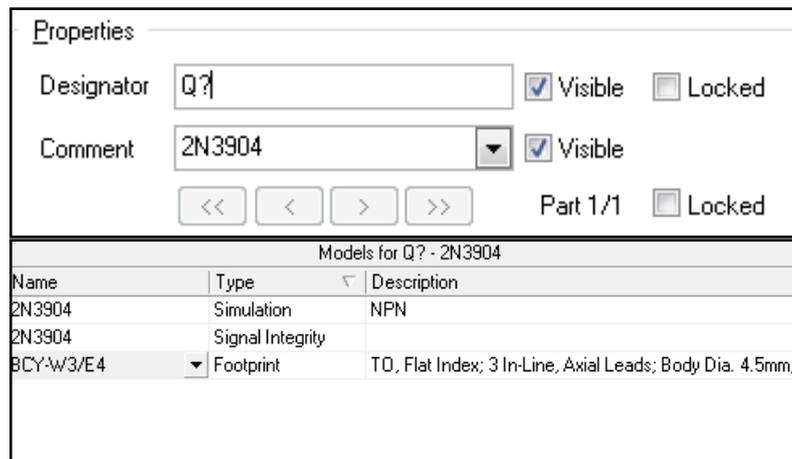


Figura 5.8. Propiedades principales de un elemento

De esta manera se colocan todos los componentes (resistencias, condensadores, conectores, etc.) en la hoja del circuito esquemático.

#### 5.1.4. Cableado del circuito esquemático

En este proceso se designan los puntos de conexión que van a tener los componentes entre si, ya sea para obtener el diagrama esquemático, simulación o circuito PCB.

1. Activamos la opción *Wire* precionando el boton *Place Wire* que se encuentra en la barra de herramientas del cableado o dando click en *Place/Wire*.  
Figura 5.9.

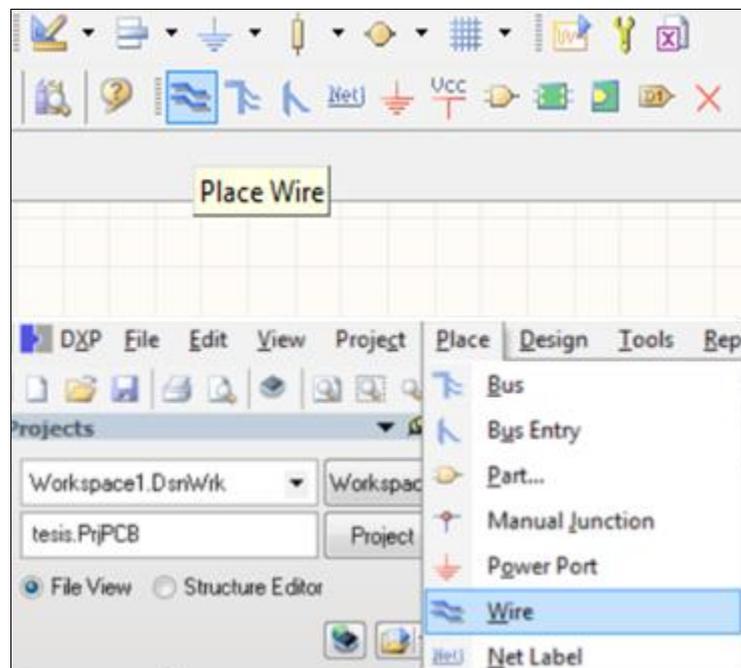


Figura 5.9. Activación de la opción Wire.

Otra manera de hacer el cableado es con la opción *Bus* y *Bus Entry* que crea un bus de datos evitando hacer el cableado línea por línea cuando se necesita cruzar varios cables. Por ejemplo para la conexión del teclado o LCD hasta el microcontrolador. Esta opción se activa directamente desde la barra de herramientas o haciendo clic en *Place/Bus* o *Bus Entry*. Figura 5.10.

2. Con estas herramientas se hace el cableado de todos los componentes.

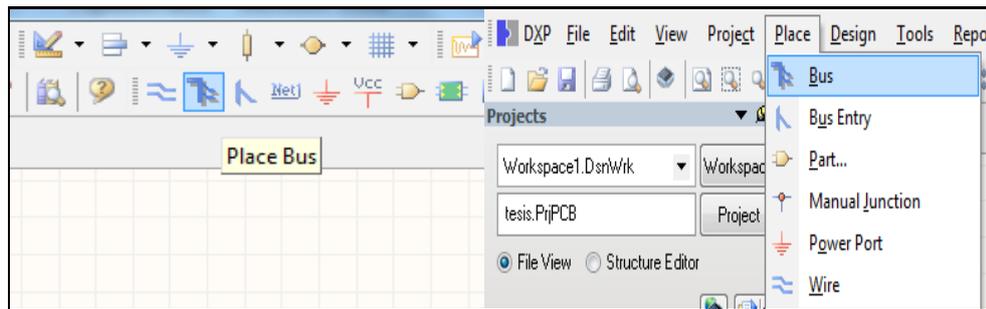


Figura 5.10. Activación de la opción *Bus*.

En la siguiente figura se muestra el cableado del circuito de potencia para las lamparas led.

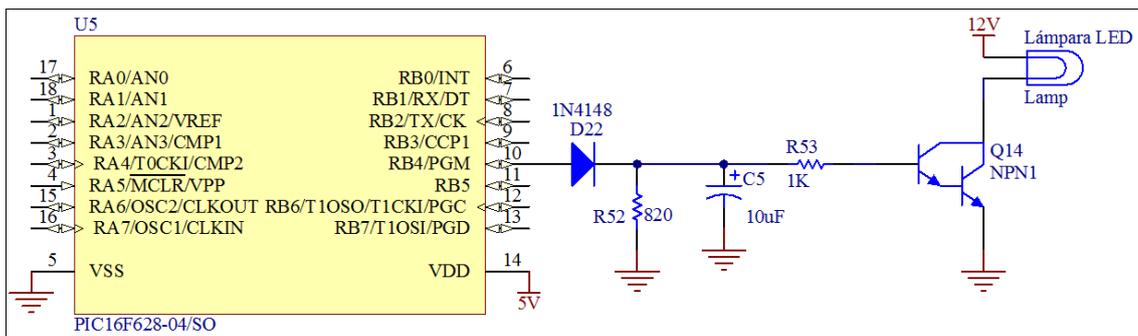


Figura 5.11. Cableado para lamparas led

### 5.1.5 Net y Net Labels (Etiquetas de conexión)

A cada conexión que existe entre dos o más terminales se les conoce como NET, por ejemplo la conexión que existe entre R52, R53 y C5 es común para los tres terminales y tiene un nombre diferente a las demás conexiones existentes en el circuito.

El referenciar a cada conexión con una etiqueta es muy útil al momento de identificar las distintas conexiones, facilita efectuar algunos cambios a una conexión en específico sin alterar ninguna otra parte del circuito. En la figura 5.11 se han agregado las etiquetas 5V y 12V que indican las terminales de las fuentes de poder de 5 y 12 voltios respectivamente.

Las etiquetas se agregan directamente desde la barra herramientas o desde el menú *Place/Net Label*. Las propiedades se modifican dando doble clic en el nombre de la etiqueta. Figura 5.12.

Para posicionar el *Net Label* arrastrar el cursor en forma de cruz hasta que toque la conexión, este curso cambiará a color rojo cuando la conexión sea correcta.

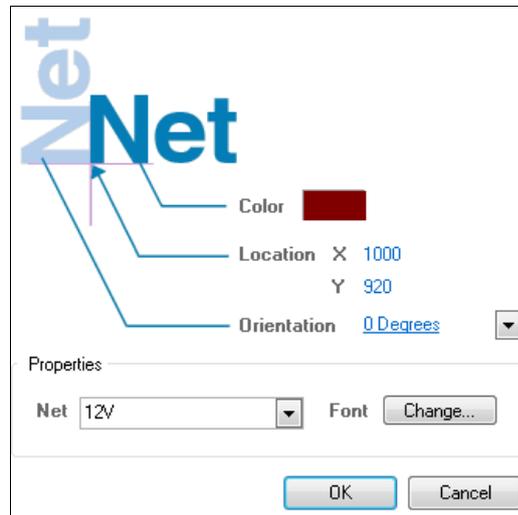


Figura 5.12. Propiedades de un Net Label

Con esto se concluye la elaboración del circuito esquemático.

En la figura siguiente se muestra una parte del circuito esquemático del Sistema de Control Mediante SMS.

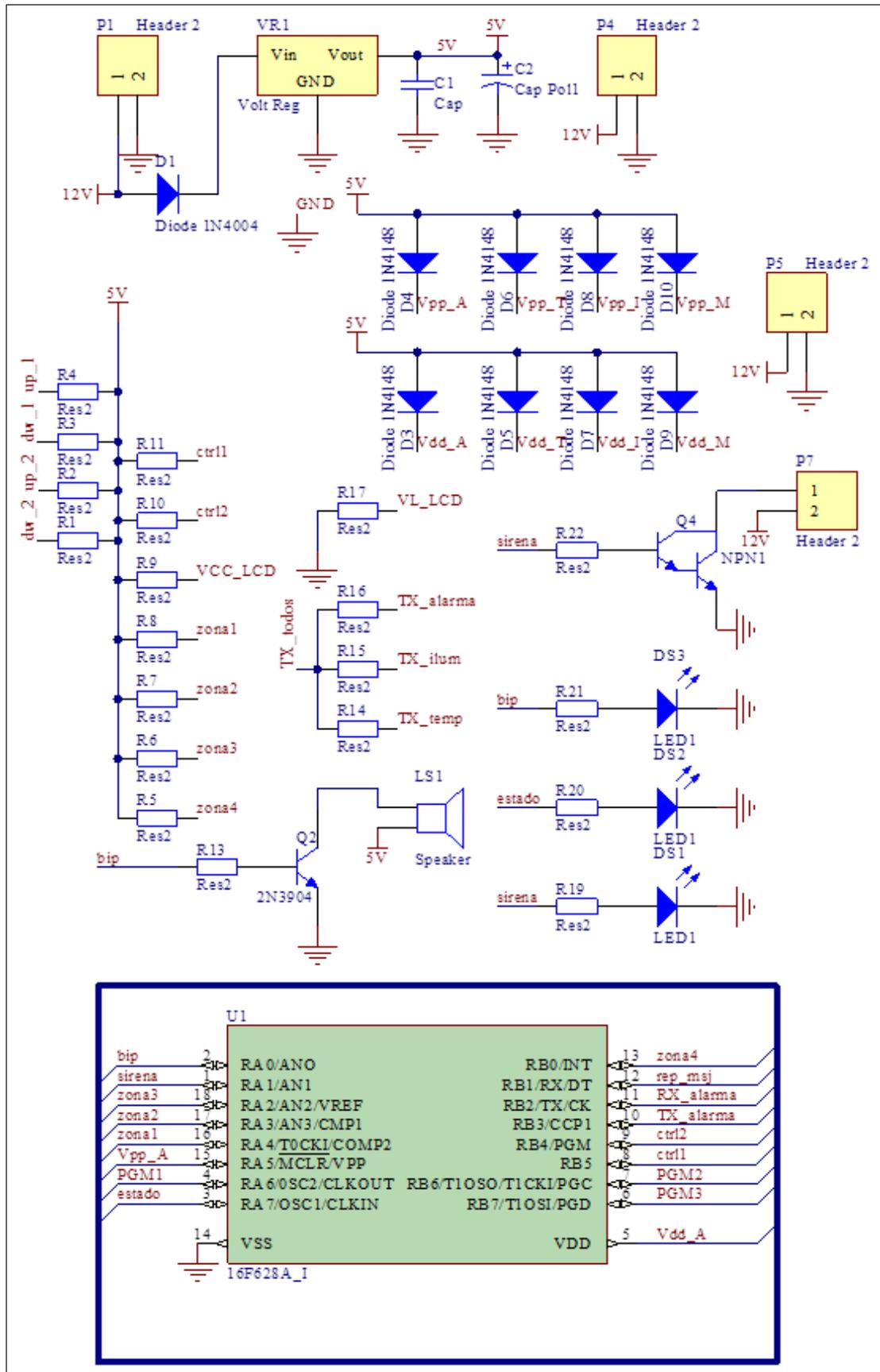


Figura 5.13. Circuito esquemático del Sistema (sección alarma).

## 5.2. Tarjeta de circuito impreso (PCB)

Un circuito impreso es una superficie constituida por caminos o pistas de material conductor laminadas sobre un sustrato no conductor. El circuito impreso se utiliza para conectar eléctricamente por medio del sustrato, un conjunto de componentes electrónicos. Las pistas son generalmente de cobre mientras que el sustrato se fabrica de resinas de fibra de vidrio reforzada (la más conocida FR4), cerámica, plástico, teflón o polímeros como la baquelita.

Una vez probado el circuito esquemático del proyecto, construiremos una placa de circuito impreso.

### 5.2.1 Ruteo de Pistas para PCB

El ruteo de pistas se traza basándose en el circuito esquemático, el ruteo se hará de forma manual para realizar cambios futuros en la placa sin tener el inconveniente de que cada placa modificada tendrá un PCB completamente diferente al anterior.

Para la construcción del PCB utilizaremos el software PCB Wizard que contiene una librería completa para los componentes utilizados en el proyecto.

1. Desde la barra de herramientas escogemos la opción *PCB components* y arrastramos hasta el área de trabajo los componentes necesarios para la construcción del circuito PCB. Figura 5.14.

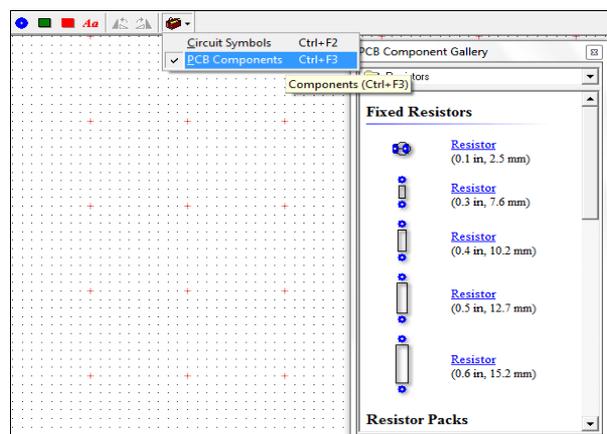


Figura 5.14. Componentes para el circuito PCB

2. Las conexiones entre los componentes se puede hacer mediante pistas (tracks), vías (pad) y cables (wires).

Para adicionar manualmente las pistas escoger cualquiera de las dos opciones: Menú Insert/Track o desde la barra de herramientas hacer clic en el botón Track.

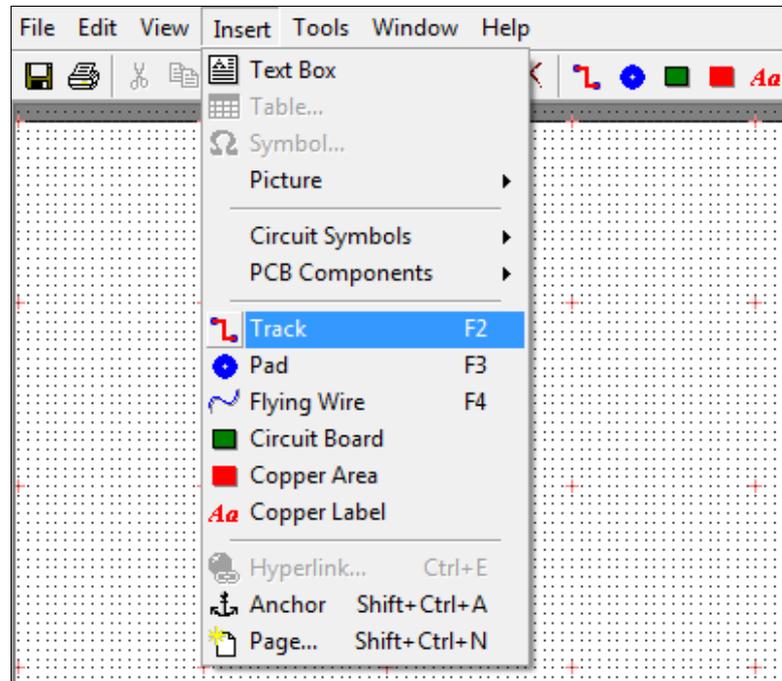


Figura 5.15. Herramienta Track

Doble click en la pista dibujada nos permite cambiar las propiedades (grosor, distancia con el área de cobre, capa a la que pertenece).

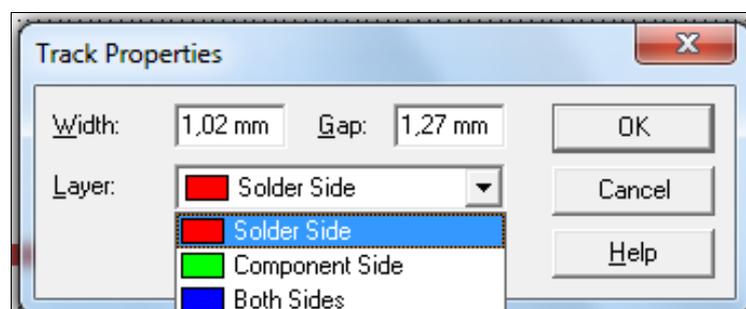


Figura 5.16. Propiedades de la Pista

Para adicionar vías o cables se sigue el mismo procedimiento.

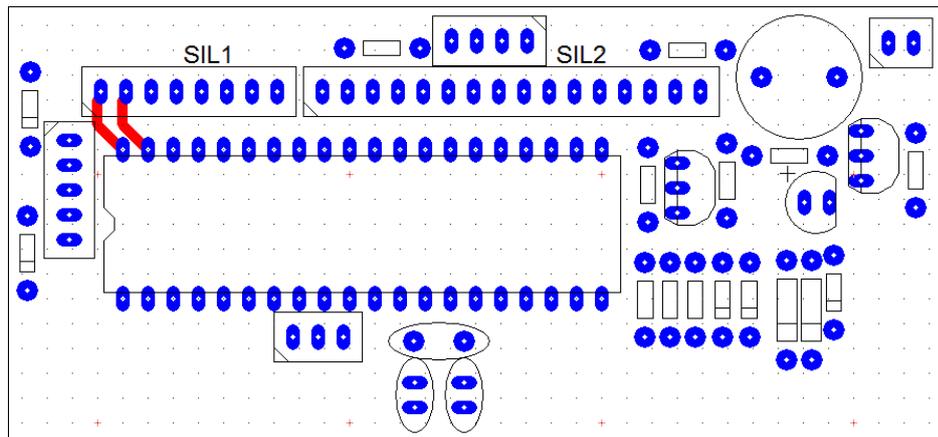


Figura 5.17. Ruteo de pistas a cada componente

- Desde la barra de herramientas escogemos la opción *Circuit Board*, esta herramienta crea un perímetro alrededor del circuito ruteado, este perímetro representa el tamaño físico de la placa de cobre en donde se va a transferir el circuito. Figura 5.18.

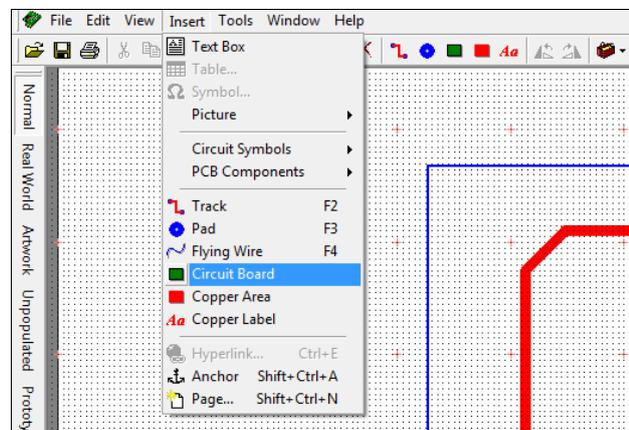


Figura 5.18. Herramienta Circuit Board

- La herramienta Copper Label permite escribir en la parte de cobre, estas etiquetas ayudan a identificar el fabricante del PCB, versión de la tarjeta, fecha de creación, etc. En la siguiente figura se muestra en ejemplo:

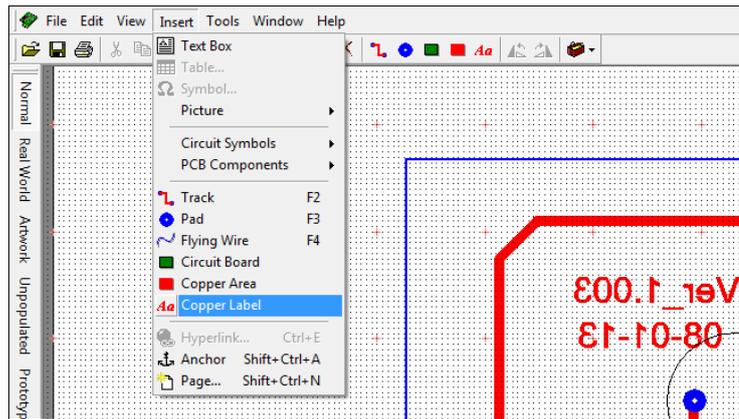


Figura 5.19. Herramienta Cooper Label

5. Cooper Area (Figura 5.20.) es una herramienta que permite crear una área de cobre, sin unirse con las pistas dibujadas, es muy útil al momento de construir el PCB, mientras más cobre existe en la placa más rápida se hace la transferencia del circuito impreso.

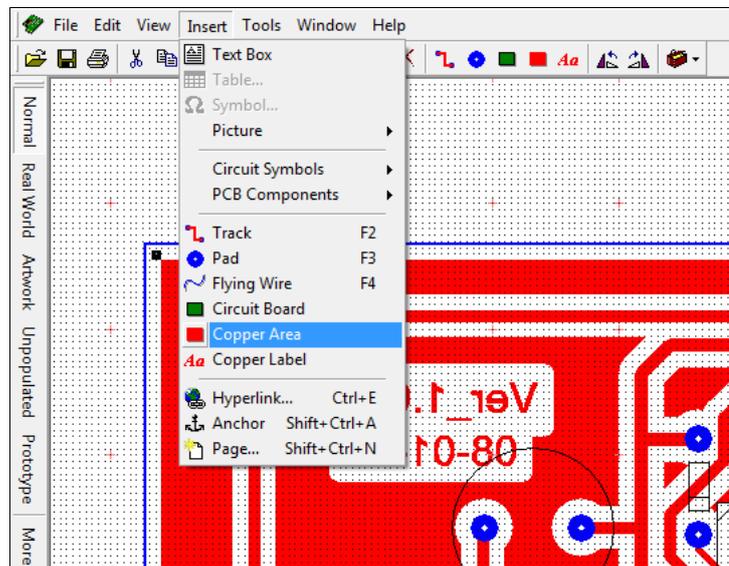


Figura 5.20. Herramienta Cooper Area

Una vez ruteada toda placa utilizando las herramientas principales de PCB Wizard tendremos una placa parecida a la que se muestra en la figura 5.21.

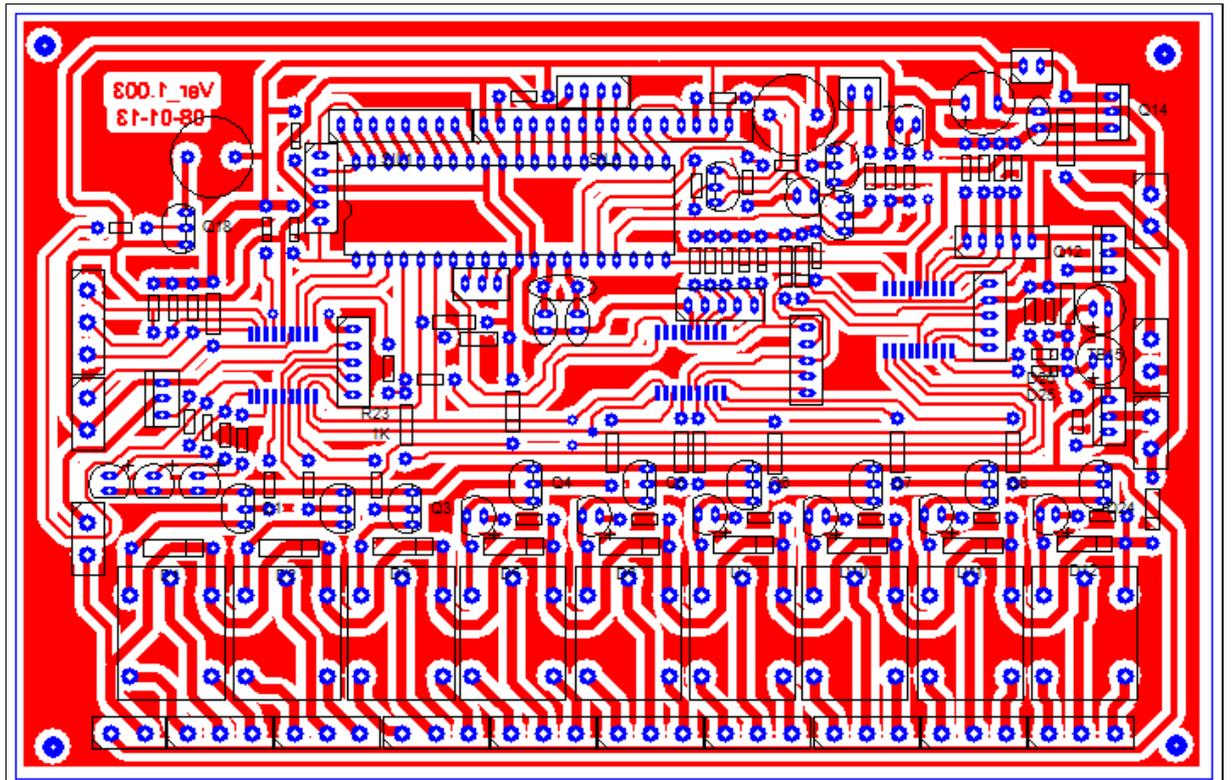


Figura 5.21. Circuito PCB para el Sistema de Control mediante SMS

### 5.2.2 Impresión de pistas para PCB

Una vez terminado el diseño de la placa, tenemos que imprimirlo en una impresora láser, copiadora o cualquier dispositivo que tenga los cartuchos tóner de polvo en color negro. Las hojas que se utiliza son papel de transferencia térmica Press-n-Peel (papel de transferencia PCB).

Otra alternativa es el papel fotográfico tipo Glossy, este papel sirve para imprimir fotografías específicamente en impresoras de inyección de tinta, al imprimirlo en una impresora láser se da un efecto químico al unirse el tóner de la impresora con la capa de barniz que tienen estas hojas (lo mismo ocurre con el papel Press-n-Peel).

El siguiente paso es aplicarle calor por el lado revés de la hoja y sobre las placas, el calor hace que el tóner se derrita y junto al barniz de la hoja se pegan a la lámina de cobre. Para esto iremos explicando paso a paso todos los procedimientos necesarios.

### 5.2.3. Preparación de la Placa (Baquelita o Fibra de Vidrio)

Los materiales utilizados para la preparación de la lámina son:

- 1 placa de baquelita o fibra de vidrio, de una cara de cobre
- 1 lija de metal número 150.
- 1 esponja de acero.

1. Cortamos la placa que puede ser de baquelita o de fibra de vidrio, para las medidas de corte consideramos mínimo 4mm adicionales a cada lado de la placa en relación al del dibujo que vayamos a transferir. Figura 5.22.

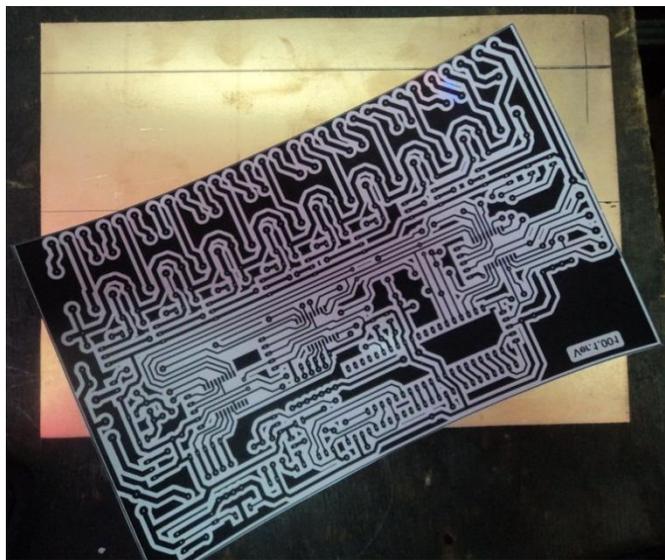


Figura 5.22. Placa y hoja ara circuito impreso

2. Una vez cortada la placa, limpiamos la limallas de cobre que queda en los fillos de la placa con una lija fina de metal, luego limpiamos con una esponja de acero el lado de cobre en donde vamos a transferir las pistas, la lámina de cobre quedará brillante ya que hemos sacado todo el óxido creado en la superficie y los rayones que pudiera tener

### 5.2.4. Transferencia térmica del papel hacia la lámina de cobre

Materiales:

- 1 plancha doméstica o estampadora.
- 2 Pedazos de Papel Bond.
- Placa de cobre íntegramente limpia.
- Papel con el diseño a transferir.

1. Colocamos el papel fotográfico con el lado de la tinta sobre el lado del cobre, introducimos debajo de la hoja de papel bond, todo esto sobre una mesa rígida. Con la plancha a nivel alto de temperatura mantenemos encima de la placa (por la parte de cobre) unos 5 segundos aproximadamente, retiramos la hoja de papel bond y pasamos la plancha por toda la placa uniformemente, este procedimiento dura aproximadamente 5 minutos, por ultimo enfriamos completamente la placa con la finalidad que toda la tinta se pegue a la lámina de cobre y así poder retirar el papel sin que se presente partes cortadas o faltantes. Figura 5.23.



Figura 5.23. Transferencia térmica

La transferencia puede salir defectuosa por algunas razones que se explican a continuación:

Si la plancha no está suficientemente caliente, gran parte de la tinta no se pegará a la placa. Si se utiliza una hoja de papel bond muy gruesa, la temperatura en la placa no se transfiere completamente. Poca presión sobre la plancha, se debe prácticamente apoyarse sobre la plancha y frotarla sobre toda la superficie de la placa.

#### 5.2.5. Proceso de atacado (reducción) del cobre

Materiales:

1 Recipiente de plástico.

Agua.

1 Pinza de Plástico

1 Funda de cloruro férrico (en polvo).

Para reducir el cobre sobrante, es decir el que no está protegido por la tinta y el barniz necesitamos preparar un atacador, existen dos tipos de atacadores: los rápidos y los lentos. Los rápidos por ejemplo es la combinación de 50ml de ácido clórico y 50 ml de agua oxigenada, pueden reducir el cobre no protegido en pocos segundos, pero tiene la desventaja de ser difíciles de conseguir en el mercado. Los lentos en cambio como el Cloruro férrico se lo encuentra en cualquier tienda electrónica pero el proceso de atacado podría durar hasta 1 hora. Sin embargo por ser menos agresivo y porque no emana muchos gases tóxicos, utilizaremos cloruro férrico.

La preparación se hace en un lugar con buena ventilación, al contacto con la ropa o la piel se produce una mancha amarillenta, por lo que hay que tomar las precauciones necesarias.

1. Colocamos el agua en el recipiente de plástico, poco a poco si es posible con una cuchara de plástico, colocamos todo el contenido de la funda de cloruro férrico, es normal que el agua se empiece a calentar (debido a la reacción química), movemos el recipiente hasta disolver todo el cloruro férrico.

La solución preparada puede ser almacenada en un envase de plástico o vidrio para luego ser utilizada varias veces, hasta que el ácido se contamine tanto que ya no sea posible corroer las placas.

2. Introducimos la placa en la solución preparada, el tiempo de corrosión por ser la primera vez, puede variar entre 10 y 20 minutos, por lo que debemos revisar periódicamente para ver si el cobre no protegido ha sido eliminado. Para que el proceso sea más rápido podemos mover el agua de un lado a otro, esto permite que el cobre disuelto, por efecto de la corriente generada por el movimiento, se deposite en el fondo del envase, así permitimos que la lámina tenga contacto directo con el ácido (figura 5.24).



Figura 5.24. Reducción del Cobre

#### 5.2.6. Limpieza de la placa

Una vez que el ácido ha terminado de eliminar el cobre expuesto, retiramos la placa del ácido y lo lavamos con abundante agua (pistas, pads, etiquetas), en esta fase se ven de color negro, esto se debe a que el papel se encuentra remojado, pero cuando se seque volverá a ser de color blanco. Limpiamos todo el residuo de papel y tinta de dos maneras posibles, la primera es utilizando thinner, acetona o cualquier disolvente, la segunda opción es utilizar la misma esponja de acero y un poco de agua.

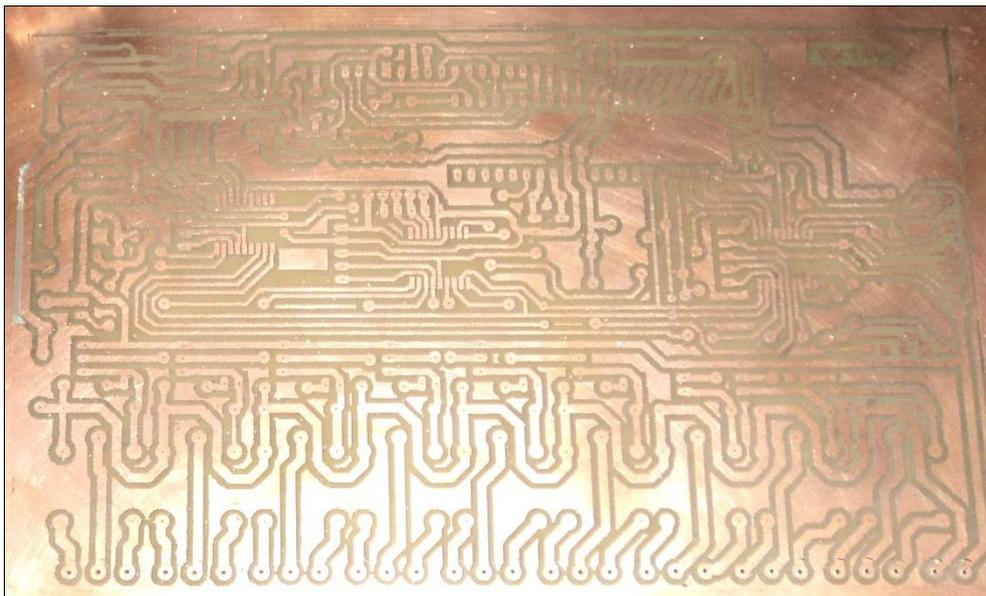


Figura 5.25. Limpieza de la Placa

### 5.2.7. Perforación de la placa

Materiales:

Brocas de metal.

1 taladro o moto-tool (taladro miniatura)

Lo único que nos falta para que la placa esté lista es realizar los respectivos agujeros, una buena herramienta para realizar estos agujeros es un moto-tool o taladro miniatura, estos son de fácil manipulación y permiten colocar brocas de 0,3mm en adelante (figura 5.26).

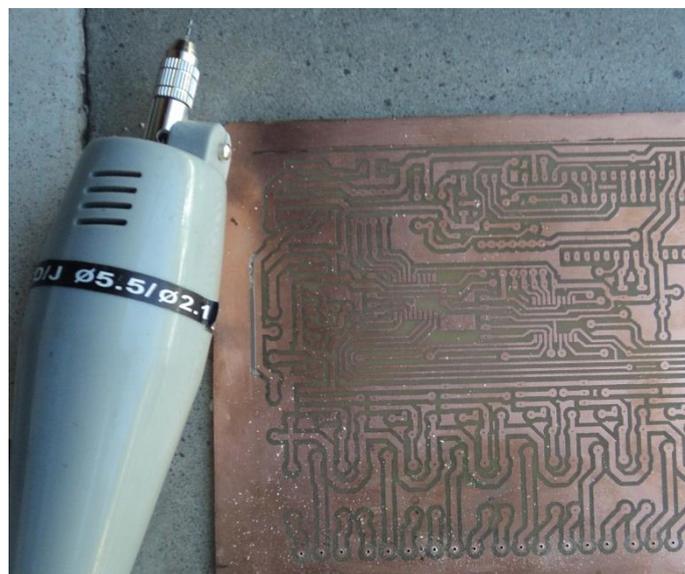


Figura 5.26. Perforación de la Placa

### 5.2.8. Soldadura de elementos

Materiales:

Cautín tipo lápiz de 20W a 30W.

Pasta de Soldar.

Alambre de suelda de estaño.

Componentes del circuito esquemático.



Figura 5.27. Materiales, Pasta para soldar, estaño, cautín eléctrico

Se debe seguir una secuencia de soldadura de los componentes, primero los elementos más bajos y por último los más altos como los capacitores o borneras.

1. El método para soldar los elementos, es calentar un poco el elemento a soldar y luego poner el estaño, movemos la punta del cautín de abajo hacia arriba tocando el alambre de suelda y el elemento, esto permite una rápida adherencia y una buena soldadura.

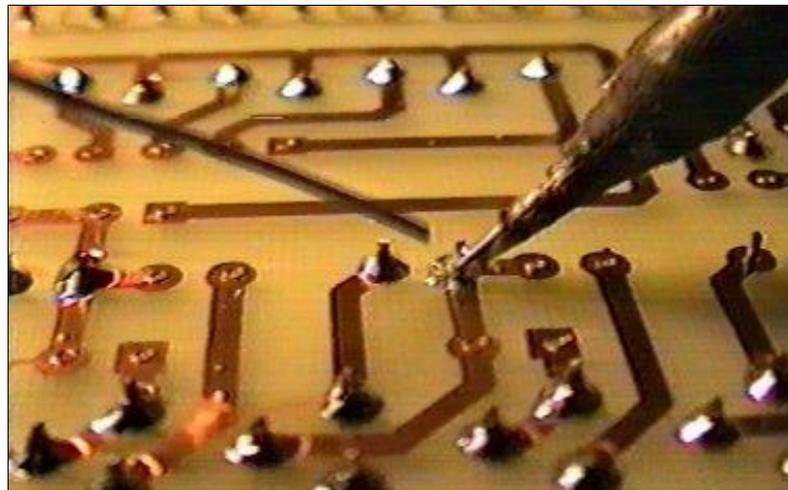


Figura 5.28. Soldadura de elementos

### 5.3. Chasis o caja para el proyecto.

En el mercado podemos encontrar diferentes cajas metálicas o de plástico para proyectos o instalaciones eléctricas. En las siguientes figuras se muestra dos tipos de cajas.



Figura 5.29. Cajas plásticas para proyectos electrónicos

Fuente: CONSTRUNARIO. [http://www.construnario.com/notiweb/noticias\\_imagenes/28000/28243.jpg](http://www.construnario.com/notiweb/noticias_imagenes/28000/28243.jpg).  
[Consulta: 18 de Agosto de 2012]

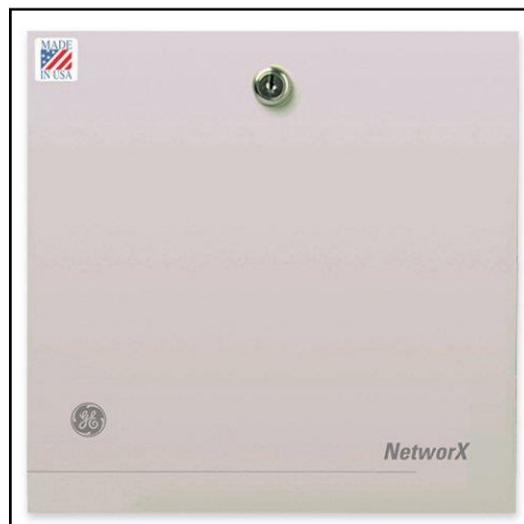


Figura 5.30. Caja Metálica de Central de Alarma

Para el Sistema de Control utilizaremos una caja de central de alarma, las medidas de la caja son 280x285x90mm, suficiente para colocar en el interior la tarjeta central del sistema, fuente electrónica de 12V-3A, batería de respaldo 12V-4Ah, transformador 110VCA a 16,8 VCA 40VA, módulo SIM900.

### 5.3.1. Instalación de elementos

Para la instalación del LCD Matricial tenemos que hacer una ranura de 83x42mm en la tapa de la caja metálica, para el teclado matricial se necesita una ranura de 25x5mm para introducir el conector. En la figura siguiente se muestra los elementos LCD y Teclado a ser instalados.



Figura 5.31. Display LCD y Teclado Matricial a ser instalados

Una vez hechas las ranuras en la caja metálica colocamos los dos elementos (LCD y Teclado) y conectamos a sus respectivos pines en la tarjeta electrónica principal. En la siguiente figura se muestra cómo quedará la parte exterior de la caja metálica.



Figura 5.32. Parte exterior de la caja metálica

El módulo SIM900, fuente, transformador, batería y tarjeta principal van instalados en el interior de la caja, la antena del modem GSM tiene que ser instalada en un lugar donde exista una buena recepción de señal celular de la operadora que se esté ocupando.

Las conexiones de teclado, lcd, fuente, batería, transformador, módulo SIM900, lámparas led, ventiladores, niquelinas, sensores, sirenas, se hacen dentro de la caja.

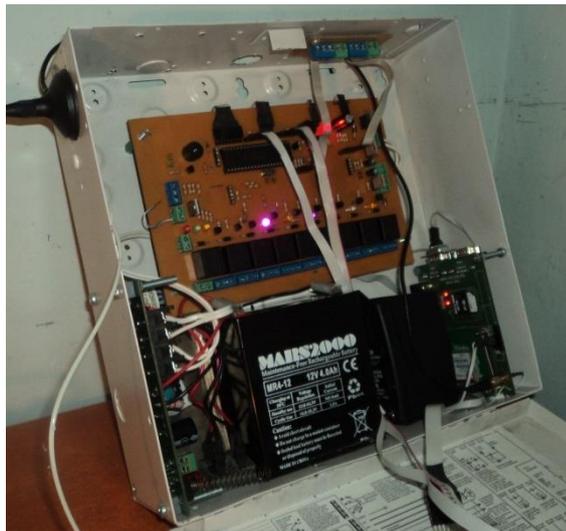


Figura 5.33. Conexiones del Sistema de Control mediante SMS

## CAPÍTULO 6

### APLICACIÓN PARA TELÉFONO MÓVIL

Una aplicación es un programa que se puede descargar a cualquier dispositivo móvil, permiten al usuario participar de juegos, obtener indicaciones de localización, enviar o recibir mensajes de texto, e-mails, acceder a noticias, libros, datos de tiempo y demás.

Para el control del sistema de alarma, iluminación y climatización diseñaremos una aplicación para teléfono móvil que soporte un sistema operativo android, la aplicación será capaz de controlar los tres sistemas de manera remota.

#### 6.1. Google App Inventor

Google App Inventor es una aplicación de Google Labs para crear aplicaciones para el sistema operativo Android. Es un programador visual que a partir de un conjunto de herramientas básicas, el usuario puede ir enlazando una serie de bloques para crear la aplicación. El sistema es gratuito y se puede descargar fácilmente de la web.

Con Google App Inventor, el incremento de aplicaciones para android ha sido grande debido a dos grandes factores: la simplicidad de uso, que facilita la aparición de un gran número de nuevas aplicaciones; y el Android Market, el centro de distribución de aplicaciones para Android donde cualquier usuario puede distribuir sus creaciones libremente.

##### 6.1.1. Características

La creación de una aplicación en App inventor se hace desde el navegador de internet, los servidores de App Inventor guardan los proyectos realizados y ayudan a mantener un registro de todas las aplicaciones.

Las aplicaciones se construyen con: El *Diseñador* de App Inventor, en donde se selecciona los componentes para la aplicación y el *Editor de Bloques* de App Inventor, donde se ensamblan los módulos de programa que especifican como deben comportarse los componentes.

El editor de bloques de la aplicación utiliza la librería *Open Blocks* de Java para crear un lenguaje visual a partir de bloques. Estas librerías están distribuidas por Massachusetts Institute of Technology (MIT) bajo su licencia libre (MIT License). El compilador que traduce el lenguaje visual de los bloques para la aplicación en Android utiliza Kawa como lenguaje de programación, distribuido como parte del sistema operativo GNU de la Free Software Foundation

Cuando se haya terminado de construir el proyecto se puede empaquetar para obtener una aplicación independiente para instalar en cualquier dispositivo que soporte sistema operativo android.

En la figura siguiente se muestra un diagrama en bloques de la programación en App Inventor.

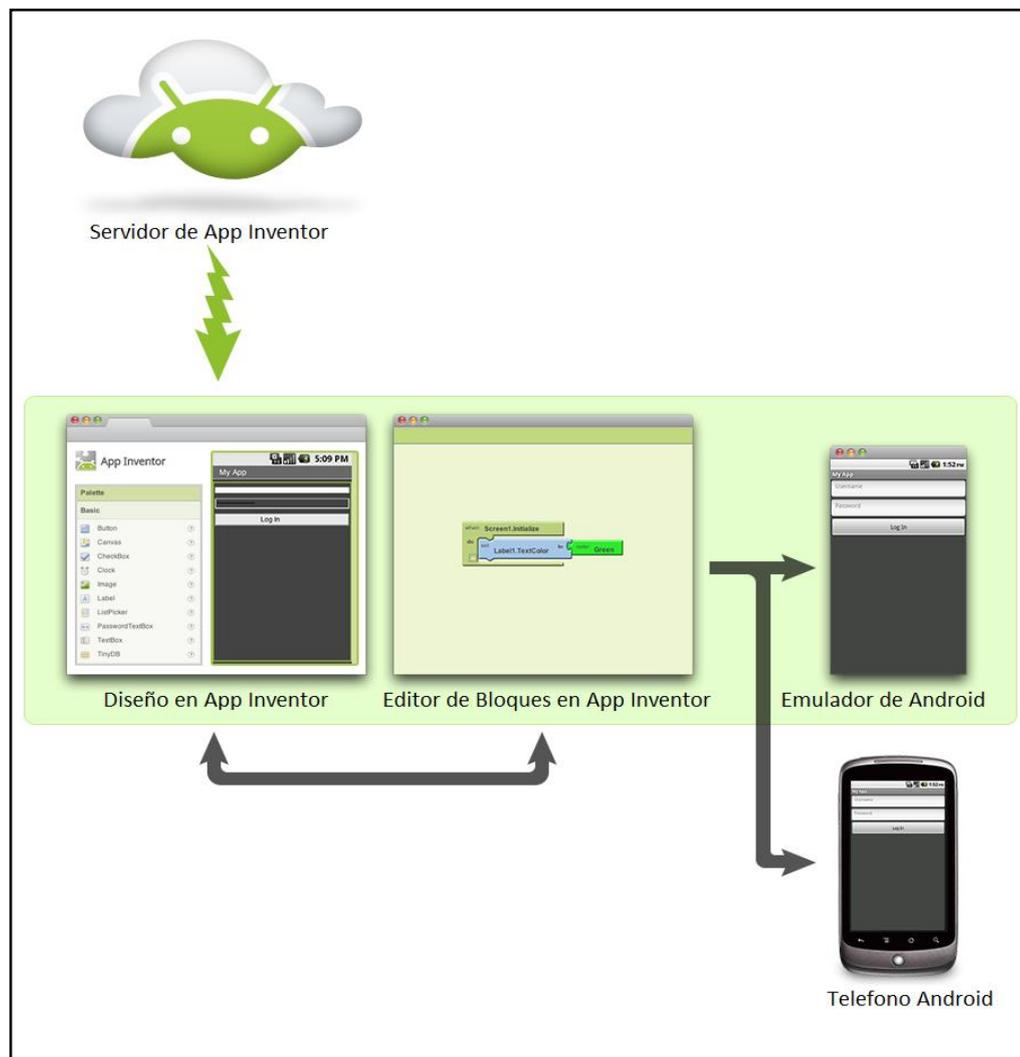


Figura 6.1. Diagrama en bloques para programación en App Inventor

Fuente: APPINVENTOR. <http://appinventor.mit.edu/> [Consulta: 9 de Febrero de 2012]

### 6.1.2. Configuración MIT App Inventor

App Inventor requiere Java por lo que tiene que estar instalado en el sistema operativo del nuestro ordenador, para ejecutar App Inventor es necesario instalar un paquete llamado *Configuración App Inventor*.

Antes de empezar nos aseguramos de tener acceso a: internet y una cuenta de Gmail que es el medio por el cual nos vamos a conectar a App Inventor.

### 6.2. Aplicación móvil para el sistema de control de domótica

Con el explorador web abrimos el siguiente sitio: <http://beta.appinventor.mit.edu/>, es la página que permite crear las aplicaciones y en donde quedarán guardadas.

1. Hacemos “clic” en el botón *New* a lado izquierdo, cerca de la parte superior de la pantalla.
2. Introducimos el nombre del proyecto (Una palabra sin espacios) en el cuadro de dialogo que aparece.

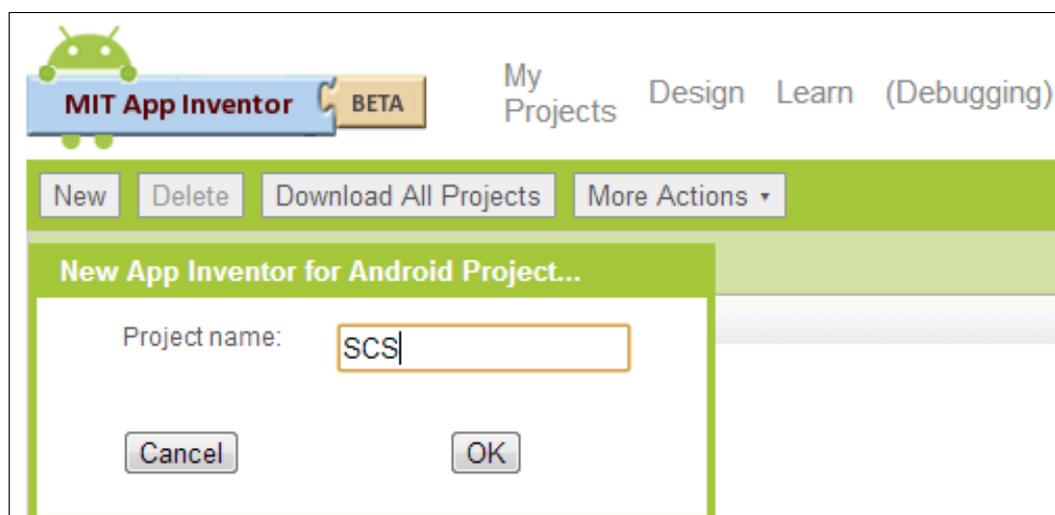


Figura 6.2. Pantalla para un Nuevo Proyecto

El navegador abrirá una página web denominada *Diseñador*, aquí es donde se seleccionan los componentes para la aplicación y el diseño de la interfaz de usuario. Figura 6.3.

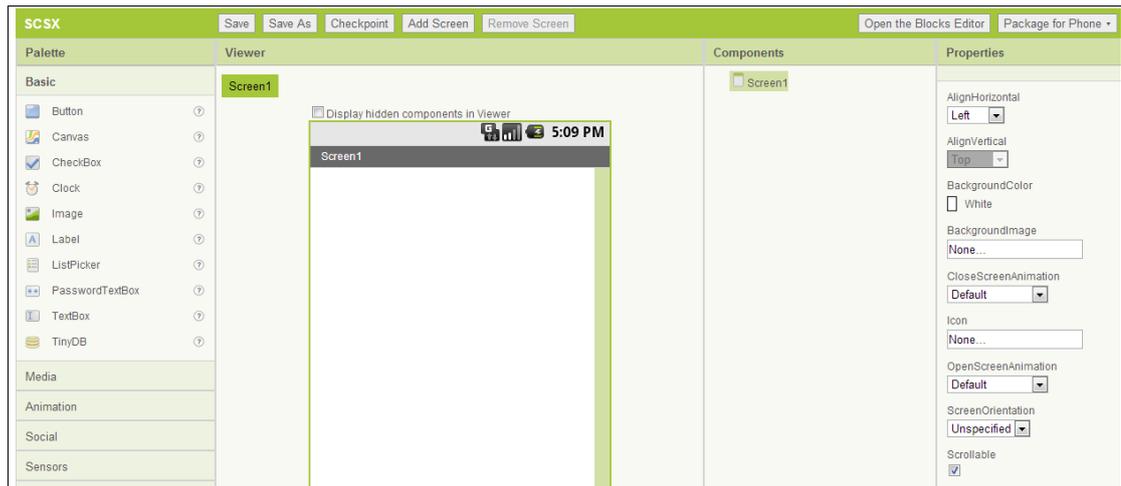


Figura 6.3. Pagina Web denominada Diseñador

Además de la página *Diseñador* es necesario iniciar el editor de bloques, este editor permite ajustar el comportamiento de la aplicación. Se trata de una aplicación independiente con su propia ventana, estas ventanas están relacionadas, así que los cambios realizados en la ventana de diseño se refleja inmediatamente en el editor de bloques.

3. Hacemos “clic” en *Abrir el Editor de Bloques* de la ventana *Diseñador* (parte superior derecha), el editor de bloques es un archivo de programa que debe ser descargado y ejecutado.

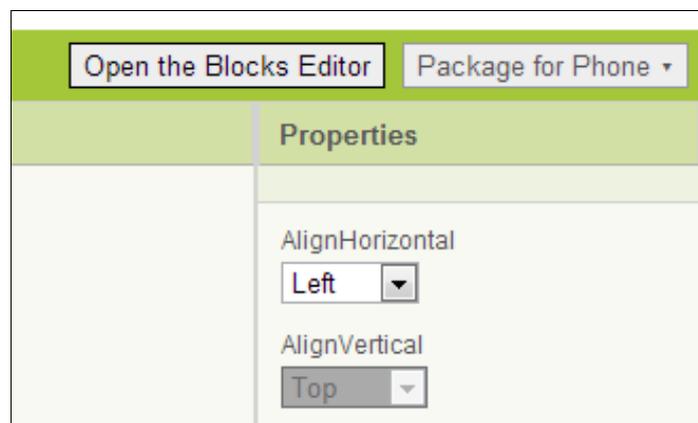


Figura 6.4. Abrir el Editor de Bloques

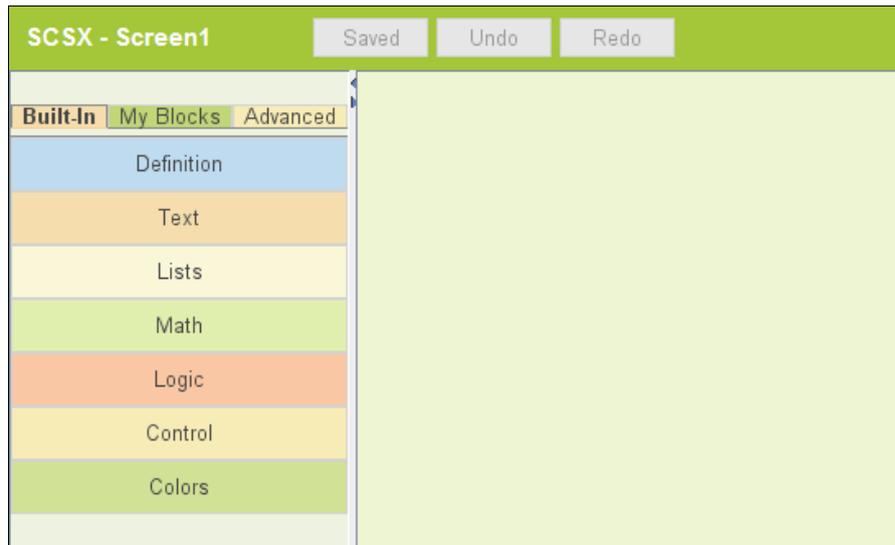


Figura 6.5. Editor de Bloques de App Inventor

El espacio vacío en el lado derecho de la pantalla se conoce como zona de trabajo, en donde se colocan los bloques para armar el programa.

En el lado izquierdo encontramos tres paletas (Built-In, My Blocks, Avanzado) en donde se almacenan los bloques.

*Built-In* contiene un conjunto de bloques para las aplicaciones (texto, listas, operaciones matemáticas, control, etc.). *My Blocks* contiene bloques específicos que están relacionados con el conjunto de componentes que se han elegido para la aplicación. *Avanzado* contiene bloques para construir aplicaciones con una lógica más compleja.

El *Diseñador* se ejecuta desde el navegador y el *Editor de Bloques* se extiende desde java, sin embargo estas ventanas están vinculadas, por lo tanto, incluso si se cierra el editor de bloques, toda la información se almacena en el *Diseñador*.

- Desde el navegador de internet diseñamos la parte grafica de cada pantalla que va a tener la aplicación. Tenemos herramientas Básicas, Multimedia, Animación, Social, Sensores, Arreglos de Pantalla etc.

En cada pantalla colocamos los elementos necesario como: botones, etiquetas, cajas de texto, imágenes, texto etc. y cargamos los archivos a utilizar como: imágenes, audio, etc. Renombramos los elementos para evitar confusiones en el editor de bloques. En la siguiente figura se muestra algunas de las pantallas de la aplicación.



Figura 6.6. Pantallas de la Aplicación Móvil

5. En el editor de bloques damos las instrucciones a cada botón, caja de texto, etiquetas etc. en la siguiente figura se muestra la programación de la pantalla de INICIO.

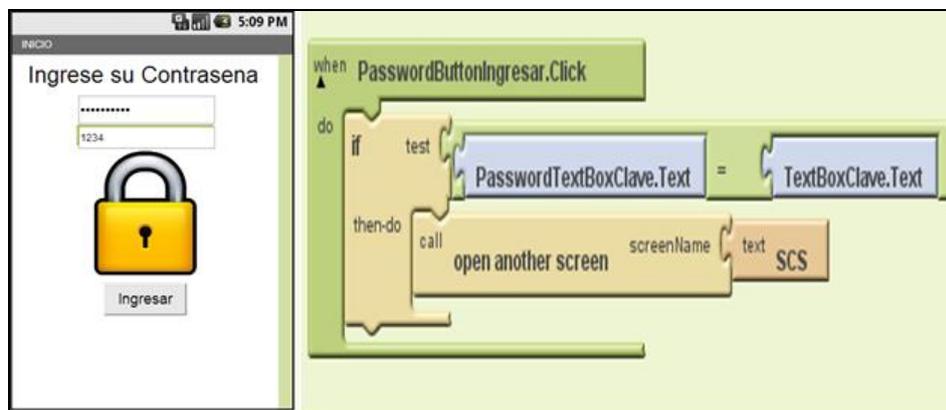


Figura 6.7. Programación de la Pantalla de INICIO (Clave)

En el diagrama de bloques se entiende fácilmente: Cuando presionamos el botón Ingresar, comparamos la clave ingresada en el espacio *PasswordTextBoxClave* con el guardado en *TextBoxClave*, si son iguales abrimos una nueva pantalla, caso contrario no se realiza ninguna acción.

Una vez ingresada la clave correcta la aplicación abre otra pantalla, en la figura siguiente se muestra la programación de la pantalla SCS la cual permite abrir diferentes pantallas al presionar una imagen o simplemente un botón.

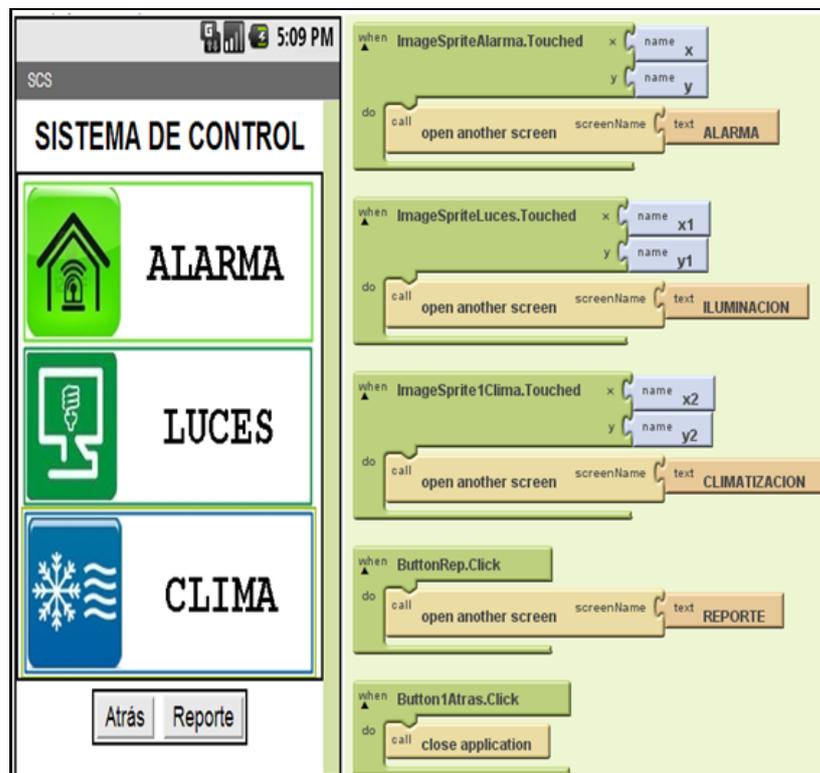


Figura 6.8. Programación para abrir diferentes pantallas

El diagrama de bloques dice: al presionar *ImageSpriteAlarma* abrimos una nueva pantalla llamada *ALARMA*, las pantallas *LUCES* y *CLIMA* se abren de la misma forma.

También se ha programado el botón *ButtonRep* que al ser presionado abrirá la pantalla *REPORTE* y al accionar el botón *Button1Atras* se cierra la aplicación.

Como se explicó en el capítulo 3, el sistema de control se basa en mensajes de texto (SMS), con la pantalla del sistema de alarma se explicará cómo enviar mensajes de texto al número celular instalado en el módulo GSM SIM900.

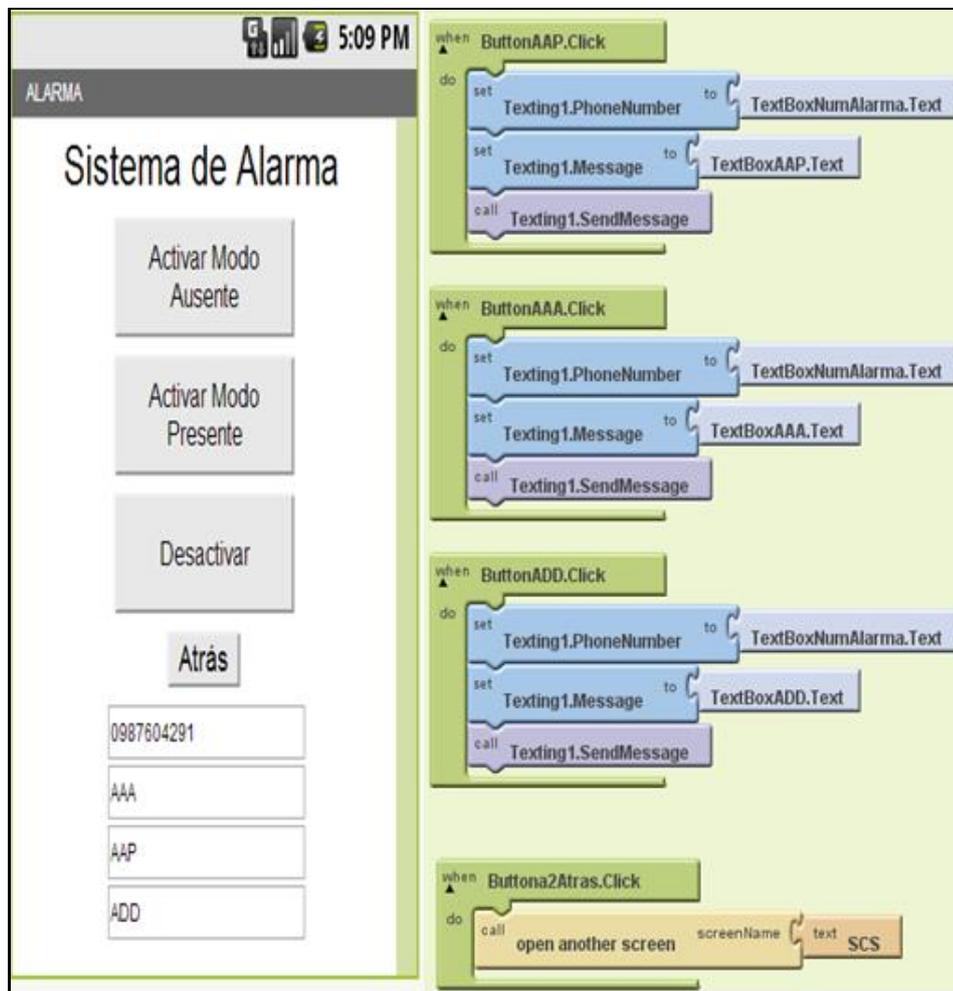


Figura 6.9. Envío de Mensajes de texto

El diagrama de bloques indica que al presionar el botón *ButtonAAP* (Activar Modo Presente) el programa pide un número de teléfono que será siempre el que esté escrito en la caja de texto *TextBoxNumAlarma* (0987xxxxxx) y un mensaje de texto que según el botón presionado puede ser el valor escrito en las cajas de texto *TextBoxADD*, *TextBoxAAA* o *TextBoxAAP*.

El botón *Button2Atras* sirve para abrir la pantalla anterior (Pantalla SCS).

De esta manera se envía mensajes de texto con solo presionar un botón. Para los diferentes sistemas se ocupa el mismo método.

6. Una vez programada toda la aplicación se empaqueta para ser exportada al teléfono móvil directamente o al ordenador para luego ser copiada al dispositivo

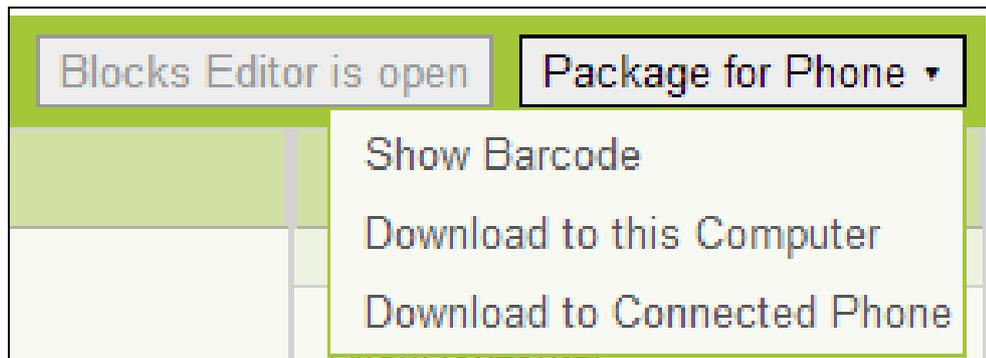


Figura 6.10. Paquete para el teléfono móvil

7. Una vez descarga la aplicación procedemos a instalarla en el dispositivo móvil, hay que tomar en cuenta que para el correcto funcionamiento, la aplicación tendrá que ser instalada en un dispositivo que soporte conexión GSM ya que la aplicación se basa en el envío y recepción de mensajes de texto.

## CONCLUSIONES

El proyecto realizado ha contribuido de manera muy importante para mejorar el nivel de seguridad de una vivienda o local comercial, por la facilidad de controlar el sistema de forma remota, así como el consumo eléctrico por la utilización de lámparas led, las cuales tienen una corriente nominal de funcionamiento de 500mA.

La integración de un modem GSM y un microcontrolador PIC, mediante la comunicación serial y utilización de comandos AT, permite crear aplicaciones tecnológicas de alto nivel por la gran cantidad de señales que pueden generar estos dispositivos, sin tener limitaciones para el control de cualquier tipo de dispositivo.

El control del sistema mediante una aplicación basada en la programación en Java para dispositivos móviles permite controlar el sistema de una forma fácil y su uso se hace óptimo. La aplicación no permite errores a la hora de enviar mensajes, como podría suceder con el control de forma manual, por lo que los SMS son guardados en una base de datos (TinyDB de App Inventor) que es un almacén de datos, los cuales estarán siempre disponibles al ejecutar la aplicación.

El uso de la red GSM para el control del sistema es un aspecto muy importante ya que la mayoría de personas tiene acceso a esta tecnología, comparándola con una conexión a internet que recién ahora se está implementando en domicilios y todavía son pocos los usuarios que ocupan el servicio de internet de sus teléfonos móviles.

El sistema de alarma e iluminación, al ocupar elementos de bajo consumo de corriente como son sensores infrarrojos, sensores magnéticos, lámparas led, seguirán funcionando sin el suministro de red eléctrica ya que el sistema está respaldado por una batería de 12V 4Ah (12 voltios, 4 amperios/hora) que entregará energía suficiente por un tiempo aproximado de 10 horas, para el correcto funcionamiento del sistema, esto no sucede con el sistema de climatización, ya que ocupa elementos que funcionan directamente con la red eléctrica como son las niquelinas y ventiladores.

## RECOMENDACIONES

Al querer realizar el control de algún dispositivo de forma remota, se recomienda el uso de la red GSM, por lo que esta red está disponible y es utilizada por la mayoría de personas en nuestro medio.

Para la recepción de señal GSM, utilice un modem SIM900 por lo que éste se maneja con simples comandos AT mediante la interfaz RS-232 basada en comunicación serial.

Para la comunicación con el modem y control de dispositivos, utilice un microcontrolador PIC ya que, existen varios lenguajes de programación para ponerlos en funcionamiento, se recomienda utilizar el lenguaje de programación en BASIC con el software Proton IDE por la gran cantidad de librerías que por defecto vienen guardadas, estas librerías permiten la configuración directa de elementos (teclados, sensores, lcd, etc) y formas de comunicación (UART, Paralelo, USB, etc).

Para el correcto funcionamiento del sistema se recomienda:

- Instalar la antena de red GSM en un lugar donde se tenga una buena cobertura de señal.

- Ocupar fuentes de voltaje que entreguen una tensión máxima de 13,8 VCD, instalar baterías de valor nominal de 12VCD.

- Conectar solo dispositivos que ocupen corrientes menores a la que entrega la fuente de voltaje o batería.

Para tener un respaldo de energía eléctrica más prolongado:

- Instalar baterías de mayor amperaje, tomando en cuenta el voltaje que suministra cada una de estas.

- Programar el sistema de iluminación para encender la lámparas a un nivel bajo o medio.

- No ocupar accesorios de alto consumo de corriente cuando no se tiene suministro de la red eléctrica.

## BIBLIOGRAFIA

### Referencias Bibliográficas

- GARCIA, F. GONZALES, E. Electricidad-Electrónica Básica. Madrid-España. 2004. 2da Edición.
- GROB. Electrónica Básica. McGraw-Hill. 2003. 5ta Edición.
- Howard, Aiken. Fundamentos de Electricidad. Estados Unidos. 2005. 3ra Edición.
- MARTINEZ, Ignacio. ROMERO, Susana. Microcontroladores PIC. Madrid-España. 2006.
- REYES, Carlos. Aprenda rápidamente a programar microcontroladores PIC. Ecuador. 2004. 193 páginas. 1era Edición.
- REYES, Carlos. Microcontroladores. Ecuador. 2011. 213 páginas. 2da Edición.
- ROSANO, Victor. Electrónica y Microcontroladores PIC. Chile. 2005. 278 páginas.
- SIMCOM. SIM900\_Hardware\_Design\_V2.00. Shanghai-China. 2010. 58 páginas.
- SIMCOM. SIM900\_AT Command Manual\_V1.03. Shanghai-China. 2010. 229 páginas.
- SIMCOM. AT Comands Set. Shanghai-China. 2009. 197 páginas.

### Referencias Electrónicas.

- APPINVENTOR. Aplicaciones para móviles. [en línea]. 2011. [Consulta 10 de Febrero de 2013]. Disponible en: <http://appinventor.mit.edu/>.
- CROWNHILL ASSOCIATES. ProtonIDE. [en línea]. 2013. [Consulta 20 de Agosto de 2012]. Disponible en: <http://www.protonbasic.co.uk/content.php/267>. MIRCROCHIP.
- MICROCHIP. Microcontroladores datasheet. [en línea]. 2009. [Consulta 16 de julio de 2012]. Disponible en: <http://www.microchip.com/TechDoc.aspx?type=datasheet&product=18f4550>.

- MIRCROCHIP. Microcontroladores datasheet. [en línea]. 2009. [Consulta 16 de julio de 2012]. Disponible en:  
<http://www.microchip.com/TechDoc.aspx?type=datasheet&product=16f628a>.
- SCRIBD. Componentes electrónicos activos. España.  
<http://es.scribd.com/doc/86410178/Componentes-electronicos-activos>.  
2007. San Francisco. [Consulta 28 de noviembre de 2012]