



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE
ADQUISICIÓN DE ESTADOS DE ALARMAS CON
TECNOLOGIA GSM SOBRE LA BASE DE UN
MICROCONTROLADOR**

T E S I S
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO ELÉCTRICO – ELECTRÓNICO
P R E S E N T A :

SALVADOR OROZCO GUTIÉRREZ

DIRECTOR DE TESIS:

ING. ALEJANDRO SOSA FUENTES





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

A **Dios** por permitirme alcanzar una de las muchas metas que espero tener en mi vida.

A mi **papá Salvador** y a mi **mamá Teresa** los amo mucho, sin su amor y apoyo no hubiera logrado terminar esto, desde pequeño me enseñaron que todo esfuerzo tiene una recompensa y ahora lo veo más claro.

A la **Facultad de Ingeniería de la U.N.A.M.**, por brindarme la oportunidad de ser egresado del mejor lugar para estudiar Ingeniería de México.

A mis **profesores** porque me brindaron sus conocimientos y enseñanza a lo largo de mi vida escolar.

A mis **abuelos paternos Salvador y Consuelo**, a mis **abuelos maternos Enrique y Micaela**, siempre me apoyaron, me quisieron y me hubiera gustado mucho que estuvieran presentes para que vieran cumplida mi promesa, pero no importa porque los llevo en el corazón, siempre los amaré.

A mis **hermanos Andrés y Daniel** por su apoyo, como dice el dicho “los últimos siempre serán los primeros”.

A mis **tíos paternos Rosa, Fernando, María, Alejandro, Guadalupe, Luis, Martha** y a sus respectivos hijos, que siempre me apoyaron desde chico y aún de grande lo siguen haciendo, los quiero mucho.

A mis **tíos maternos María, Beatriz, José, Celia, Jaime, Irene, Salomón, Joaquín, Lourdes, Felipe, Lucila, Antonio, Rocío, Enrique** y a sus respectivos hijos, por su amor y apoyo, los quiero mucho.

A mis **amigos** desde la primaria hasta la Facultad por su apoyo y amistad.

ÍNDICE

INTRODUCCIÓN	4
CAPÍTULO 1	5
SISTEMAS DE ALARMAS EXISTENTES	5
1.1 Introducción	5
1.2 Tipos de alarmas automotrices	6
1.2.1 Alarmas con chapa bancaria	6
1.2.2 Alarmas a control remoto	7
1.2.3 Alarmas con protección vía satélite	9
1.3 Alarmas de casa habitación	10
1.3.1 Elementos de una alarma de casa habitación	11
1.3.2 Sensores	12
CAPÍTULO 2	19
EL SERVICIO DE MENSAJES CORTOS DEL SISTEMA GLOBAL DE COMUNICACIONES MÓVILES	19
2.1 Introducción	19
2.2 Arquitectura de red GSM	20
2.3 Arquitectura celular del Sistema GSM	22
2.4 Interfaz Radio (Um)	23
2.5 Servicio SMS	24
2.6 Estructura de red para el Servicio SMS	26
2.7 Nivel SM-TL y protocolo SM-TP	27
2.7.1 SMS-SUBMIT (Mensaje corto originado desde el móvil)	28
2.7.2 SMS-DELIVER (Mensaje corto entregado al móvil)	32
2.7.3 Un ejemplo de trama SMS-SUBMIT	33
2.8 Acceso a los Servicios SMS	34
2.9 Protocolo de comunicación RS232	35
2.9.1 Señales del Protocolo RS232	38
2.10 Interfaz con MODEMS: comandos AT	39
2.11 Interfaz con Módems GSM	40
2.11.1 Comandos AT+	40
2.11.2 Algunos ejemplos	41
2.12 Selección del dispositivo GSM	45

CAPÍTULO 3.....47

SELECCIÓN DEL MICROCONTROLADOR..... 47

3.1	Introducción	47
3.2	Controlador y microcontrolador	47
3.3	Diferencia entre microprocesador y microcontrolador	48
3.4	Aplicaciones de los microcontroladores	49
3.5	El mercado de los microcontroladores	50
3.6	Recursos comunes a todos los microcontroladores	51
3.6.1	Arquitectura básica	51
3.6.2	El microprocesador o CPU	52
3.6.3	Memoria	52
3.6.4	Puertos de Entrada y Salida	54
3.6.5	Reloj principal	55
3.7	Recursos Especiales	55
3.7.1	Temporizadores o “Timers”	56
3.7.2	Perro guardián o “Watchdog”	56
3.7.3	Protección ante fallo de alimentación o “Brownout”	56
3.7.4	Estado de reposo ó de bajo consumo	56
3.7.5	Convertor A/D (CAD)	57
3.7.6	Convertor D/A (CDA)	57
3.7.7	Comparador analógico	57
3.7.8	Modulador de anchura de pulsos o PWM	57
3.7.9	Puertos de E/S digitales paralelo	57
3.7.10	Puertos de comunicación serie	58
3.8	Herramientas para el desarrollo de aplicaciones	58
3.9	Selección del microcontrolador	59
3.9.1	La familia de los PIC como elección	61
3.9.2	Arquitectura del PIC16F84.....	64
3.9.3	Segmentación del PIC16F84	65
3.9.4	Formato de las instrucciones del PIC16F84	65
3.9.5	Juego de instrucciones del PIC16F84	65
3.9.6	Herramientas de soporte potentes y económicas	65

CAPÍTULO 4 66

DESCRIPCIÓN DEL SISTEMA PROPUESTO 66

4.1	Introducción	66
4.2	Diagrama a bloques del sistema de adquisición de estados de alarmas	66
4.2.1	Alimentación	66
4.2.2	Dispositivo GSM (Teléfono celular GSM)	68
4.2.3	Acondicionador del protocolo RS232	68
4.2.4	Microcontrolador (PIC16F84A)	69
4.2.5	Acondicionador de Voltajes	70
4.2.6	Sensores de alarma	70
4.3	Diagrama a bloques del sistema de pruebas con la PC	71

4.3.1	Alimentación	71
4.3.2	Dispositivo para el envío y recepción de la comunicación serial asíncrona	72
4.3.3	Computadora personal (PC)	72
CAPÍTULO 5		73
SOFTWARE DEL SISTEMA DE ADQUISICIÓN DE DATOS		73
5.1	Introducción.	73
5.2	Desarrollo de programa en Visual Basic para depuración	73
5.3	Desarrollo del programa para el microcontrolador	75
5.3.1	Descripción del programa elaborado en PICBasic PRO.	76
5.3.2	Diagramas de flujo de la comunicación serial asíncrona	80
CAPÍTULO 6		83
HARDWARE DEL SISTEMA DE ADQUISICIÓN DE DATOS.....		83
6.1	Introducción	83
6.2	Alimentación	83
6.3	Microcontrolador	84
6.4	Sensores	85
6.5	Dispositivo Rx/Tx Teléfono GSM	86
6.6	Acondicionador RS232	87
6.7	Indicador de estados del PIC	89
6.8	Diagrama electrónico del sistema de adquisición de datos	90
CONCLUSIONES		91
APÉNDICES		92
APÉNDICE A		92
	Código de 7 bits utilizado en la conversión de texto a PDU	92
APÉNDICE B		93
	Código Visual Basic del programa de comunicación	93
	Módulo de conversión Hexadecimal a Decimal	93
	Módulo de formatos y limitaciones de texto	95
	Módulo de conversión de texto a PDU	96
	Forma principal del programa	98
APÉNDICE C		102
	Código en PICBasic Pro para el microcontrolador PIC16F84	102
BIBLIOGRAFÍA		106

INTRODUCCIÓN

Este proyecto va enfocado al desarrollo de un sistema de comunicación, el cual nos permita saber en forma personal el estado en que se encuentra la alarma del vehículo ya que como muchos se habrán dado cuenta en ocasiones están las alarmas activadas con sus sirenas sonando y el dueño no se ve por ningún lado o se aparece cuando ocurren dos cosas; o se acabó la batería o ya no hay coche.

Es por eso la importancia que tiene el desarrollar el sistema de comunicación GSM por medio del cual nos permita conocer personalmente el estado en que se encuentra nuestro vehículo, por ejemplo si esta sonando la alarma, nos manda un tipo de aviso, si se abrió una puerta también, o si están tratando de arrancar a este.

Por otra parte dejamos nuestra casa o negocio y no sabemos que sucede hasta que llegamos a los mismos, mientras que con el sistema propuesto podemos saber si alguien entró a la casa, rompieron algún vidrio, a que hora abrieron el negocio así como la hora a la que cerraron y todo mediante mensajes de texto enviados por un teléfono celular GSM, a otro u otros según programemos el sistema de adquisición de estados GSM.

CAPÍTULO 1

SISTEMAS DE ALARMAS EXISTENTES

1.1 Introducción

A lo largo de la historia, siempre han existido toda clase de problemas sociales, uno de estos problemas es la delincuencia, la cual ha causado estragos sobre todo en las grandes ciudades como lo es el Distrito Federal.

Están por todas partes. Las alarmas, las cámaras, los vigilantes inundan nuestra vida cotidiana. En la casa de un vecino, en el banco, en el trabajo, en la tienda, en el metro... Las medidas de seguridad han proliferado en los últimos años en México impulsadas por un fantasma que corre como la pólvora: el miedo.

Si algo caracteriza a las sociedades desarrolladas modernas, según los sociólogos y psicólogos sociales, es justamente eso: el miedo, el miedo a perder lo que uno tiene (incluida la vida), lo que ha reunido a lo largo de su historia. Y esa contagiosa sensación, agudizada a escala mundial por amenazas como el terrorismo internacional y fenómenos como la inmigración.

Dentro de los principales delitos que se cometen todos los días no solo en nuestro país sino en el ámbito mundial, es el robo de autos y una de las formas que hay para evitar este delito, es el instalar dispositivos conocidos como alarmas.

Existen una gran variedad de alarmas, desde las que se activan con chapa bancaria hasta las que se activan vía satélite, claro que dependiendo del sistema que escogemos es la cantidad que vamos a pagar por él.

Ahora bien con el desarrollo de estas alarmas han surgido otros problemas como es el caso de la contaminación ambiental por ruido, y esto debido a falsos disparos de las alarmas, causados por una mala calibración de los sensores o simplemente por sensores de mala calidad.

Otro problema es el hecho que día a día se desarrollan mejores dispositivos de seguridad en su contraparte, los delincuentes, también van desarrollando dispositivos o técnicas para su desactivación.

Además cada vez existen más vehículos con dispositivos de seguridad, esto dificulta el poder distinguir por ejemplo en estacionamientos, si es nuestro vehículo el que esta sufriendo algún intento de robo.

De ahí la importancia que tiene nuestro proyecto el cual cumple con dos características muy importantes, una es el hecho de que la alarma nos mande señales de estados a un celular, para de esa forma saber que efectivamente es nuestro vehículo el que esta siendo afectado por algún golpe, que se están abriendo alguna puerta, cofre o cajuela, que lo están tratando de encender, en fin el limite lo ponemos nosotros.

Y otra característica es el de solicitar por el mismo dispositivo un reporte de estado de la alarma, la capacidad de activar o desactivar algún dispositivo como; prender luces de una casa, apagar un automóvil, en fin depende del ingenio del instalador.

1.2 Tipos de alarmas automotrices

Todos sabemos que ha aumentado el índice de inseguridad en la capital en forma alarmante, como dato, durante 1998 fueron robados 200 vehículos diarios y en la mayoría de los casos estos ocurrieron con violencia.

Debido a este creciente índice de la delincuencia no solo en nuestro país, sino en el ámbito mundial, existe una gran variedad tanto de tipos de alarmas como de fabricantes, siempre en busca de algo novedoso para atraer el mayor mercado posible o solo con el fin de no quedarse atrás con respecto de sus competidores.

A pesar de que en los últimos años la gran mayoría de los vehículos nuevos ya cuentan con algún sistema de alarma, este no es suficientemente seguro o no cumple con las expectativas de los usuarios, por lo que se ven en la necesidad de instalar otra alarma.

Esto provoca que los fabricantes desarrollen mejoras a sus alarmas, en busca de mayores alcances, sensores con mejores características, más funciones, etc...

Una clasificación de las alarmas sería por su forma de activarlas o desactivarlas. Dentro esta clasificación se encuentran las que utilizan una llave llamada chapa bancaria, que son las que se instalaban incluso de agencia hasta hace unos años; las que utilizan un control remoto o transmisor, que de hecho son las más utilizadas, y por último las controladas vía satélite.

1.2.1 Alarmas con chapa bancaria

Estas alarmas como se dijo anteriormente eran las que instalaban en los vehículos en un inicio y constaban de un interruptor para poder activar o desactivar la alarma, el cual comúnmente era una chapa bancaria, por su complejidad de poder falsificar la combinación de esta llave.

También tenían sirenas en algunas ocasiones, o simplemente se conectaban al claxon existente, y solo se disparaba la alarma cuando se abría alguna puerta, cofre o cajuela, en el mejor de los casos tenían algún cortacorriente de tal manera que cuando se activa la alarma no se podía encender el vehículo.

El mayor problema que tenían estas alarmas era que, si por alguna razón se disparaba la alarma esta no dejaba de sonar hasta que se desactivara por medio de la chapa bancaria.

Posteriormente se agregaron mejores sensores, como acústicos o ultrasónicos, para que se activara la alarma en caso de que se rompiera algún cristal.

Características del sistema con chapa bancaria.

- Activación y desactivación vía chapa bancaria.
- Aviso vía luces intermitentes y claxon.
- Cortacorriente(opcional)

Costos de un sistema con chapa bancaria.

Dispositivo alarma	\$ 150.00
Instalación	\$ 150.00
TOTAL	\$ 300.00

Debido a la ineficiencia de estas alarmas así como del crecimiento de la tecnología, fue que surgieron las primeras alarmas a control remoto.

1.2.2 Alarmas a control remoto

Estas alarmas como se dijo reemplazaron a las alarmas de chapa bancaria, por la facilidad de activar o desactivar al sistema por medio de un control remoto además de poseer mayor lógica en sus circuitos, lo que nos resulta en mayores funciones. En la actualidad existe un gran número de fabricantes de alarmas de este tipo.

Pero a pesar de esa gran variedad de alarmas, no son muchas las diferencias entre ellas ya que parten de características similares, para entender un poco más sobre las alarmas de control remoto se señala lo siguiente.

Características del sistema a control remoto.

- **Control Remoto de Dos Botones**

Diseñados para obtener el mayor alcance para poder controlar su sistema de seguridad.

- **Armado Activo**

Modalidad para proporcionar al usuario la opción de armar el sistema desde el control cuando así sea necesario. Es una opción personal, tal como el apagador de luz de casa, le permite encenderlo o apagarlo cuando sea deseado.

- **Armado Pasivo**

Cuando se programa en este modo, el sistema de seguridad se activará 30 segundos después de que salga del vehículo. Inclusive si el usuario olvida activar el sistema, armado pasivo protegerá el vehículo de posibles ladrones.

- **Pánico / Localizador de Vehículo**

La alarma completa se activa cuando el botón de pánico es presionado fijamente, llamando la atención al vehículo, ya que suena la sirena y empiezan a destellar las luces. El usuario puede utilizarlo como un agente disuasivo. Esto incrementa la seguridad personal del usuario / dueño del vehículo. También sirve como una manera fácil y rápida de localizar el vehículo en un estacionamiento.

- **LED de Estado**

El LED de estado destella para ser utilizado como un agente disuasivo y para reportar modos importantes de seguridad, como saber si la alarma esta activada o desactivada, reportar cualquier violación que ocurrió mientras el sistema estaba armado (activado), saber si el sistema esta en modo de valet.

- **Modo Silencioso**

Silencie su sistema desde su control cuando así lo desee. El modo silencioso es ideal para armado y desarmado tarde en la noche. Simplemente presione el botón de modo silencioso por un segundo antes de activar o desactivar, y el mensaje de confirmación será eliminado para esa operación únicamente. El modo silencioso elimina todos los mensajes y sonido.

- **Sirena de Seis Tonos**

Aunque todas cuentan con esta sirena lo que varia es su potencia y los diferentes tonos que contienen.

- **Cambio de códigos**

Otra función muy importante que solo en las alarmas más caras puede realizarse es el cambio de códigos, esta es la facultad de un sistema de seguridad del automóvil al cambiar el código transmitido toda vez que la alarma es activada o desactivada. El receptor sabe que cada vez que esto ocurre tiene que esperar un código diferente.

Esta función evita que un ladrón equipado con un instrumento de grabación de frecuencias de radio, acechando en las cercanías de su auto, grabe el código de su alarma cuando la active o desactive. Utilizando dicho instrumento y reproduciendo el código después de que usted ha activado su alarma y se haya marchado, el ladrón podría, teóricamente desactivar la alarma tan fácilmente como usted.

Estas son solo algunas funciones que tienen en común la mayoría de las alarmas a control remoto, partiendo de estas es que los fabricantes ponen mayor empeño en mejorarlas o en algunos modelos de incorporar nuevas funciones como es el caso del antiasalto, o sensores de diferentes tipos como radares o detectores de movimiento en zona interna del vehículo o externa.

Para hacer un sistema de seguridad más confiable y fácil de utilizar, los fabricantes han empezado a incorporar la tecnología de radio frecuencia la cual puede viajar a través de objetos sólidos. También incrementa las distancias o alcances por la cual se puede operar un sistema. La mayoría de los sistemas de seguridad de la actualidad utilizan esta tecnología y todos operan dentro de una banda de baja frecuencia que es de los 300 MHz.

Pero esto es un problema toda vez que entre más productos electrónicos estén operando cerca del sistema de seguridad dentro de esa banda, mayor es la interferencia y menor es el alcance que se obtendrá. Por lo que una solución es operar en una frecuencia más alta como es por ejemplo el rango medio de los 400 MHz, que son los llamados sistemas HF.

Costos del sistema a control remoto

Dispositivo alarma económica	\$ 500.00
Instalación o reinstalación	\$ 300.00

TOTAL	\$ 800.00
--------------	------------------

Dispositivo alarma promedio	\$1,000.00
Instalación o reinstalación	\$ 600.00

TOTAL	\$1,600.00
--------------	-------------------

Cabe mencionar que los precios se incrementan de acuerdo a dos factores, el primero es el lugar de la instalación; es decir no cuesta lo mismo ir al taller de la esquina que ir a un taller de especialistas en donde el precio se puede hasta triplicar. Y el segundo factor es el de solicitar se incluyan más sensores o dispositivos como lo son sensores radares o de presencia, actuadores para puertas o seguros eléctricos, solenoides o abridores de cajuela, etc...

1.2.3 Alarmas con protección vía satélite

Este es un servicio que están empezando a ofrecer algunas empresas de telefonía celular, aquí en México la primera que esta haciéndolo es IUSACELL, ofreciendo su novedoso servicio llamado "STOP CAR".

STOP CAR es un nuevo servicio de IUSACELL Digital que ofrece gran protección al permitir el bloqueo de cualquier tipo de vehículo (automóvil, trailer, motocicleta, flotillas) inmediatamente después del robo.

Un pequeño dispositivo electrónico de seguridad oculto en la unidad se activa en forma remota mediante una llamada telefónica, bloqueando de inmediato los sistemas eléctrico, electrónico y de suministro de combustible. En cuestión de segundos el vehículo se detiene y no hay forma de encenderlo, aun cuando encuentren el dispositivo.

El vehículo robado no podrá circular mas allá de unas cuantas cuadras y con otra llamada telefónica se desbloquea electrónicamente el sistema de seguridad de STOP CAR para recuperarlo.

Características del sistema vía satélite STOP CAR

- Se acciona desde el celular o cualquier teléfono de tonos.
- No se puede desactivar manualmente.
- El cliente tiene una clave y un código de seguridad personal que puede ser cambiado las veces que el usuario requiera.

- El usuario desconoce la ubicación física del dispositivo dentro de su automóvil para seguridad propia.
- Se puede activar o desactivar las veces que se deseé las 24 horas del día 365 días del año.
- Se puede controlar flotillas, desde la empresa.
- El sistema también funciona a través de líneas telefónicas convencionales.
- Sin llaves, switches ni control remoto manual.

El único inconveniente del sistema es que actualmente tiene una cobertura que aunque dicen que es nacional, tan solo cubre unos cuantos estados como son: Estado de México, D.F., Cuernavaca, Pachuca, Puebla, Monterrey y Morelia.

Costos del sistema vía satélite STOP CAR

Activación	\$ 0.00
Dispositivo STOP CAR	\$2,295.00
Instalación o reinstalación	\$ 300.00
Servicio anual paging	\$ 755.00
TOTAL	\$3,350.00

1.3 Alarmas de casa habitación

Los robos en domicilios no dejan de aumentar, el aumento de casas así como de viviendas de interés social, hacen que los amigos de lo ajeno no necesiten escalar por ventanas ni arriesgarse a atacar pisos residenciales en los que los ladrones pueden ser sorprendidos con más facilidad.

Es por todo lo anterior por lo que la contratación de sistemas de seguridad se esta incrementando en nuestro país. Dependiendo de zonas, de empresas, incluso de los distintos servicios que incorporen, el precio de una alarma completa, esto es, un sistema viable, puede oscilar por los 9,000 o 10,000 pesos , a esto hay que añadir una cuota anual por la central receptora de alarmas, algo completamente necesario, que supone unos 900 pesos más al año.

Estos precios serian los básicos y se pueden elevar más en función de las necesidades específicas del cliente, tales como diferentes zonas de control, más detectores, o detectores de otro tipo, así como la asistencia de vigilantes al lugar, o la custodia de llaves alquileres de cajas fuertes, recogida de valores... etc.

La instalación de los diferentes equipos no suele ser muy complicada, y va incluida en el precio, en uno o dos días estará lista y operativa.

Otra alternativa para adquirir sistemas de seguridad, y como parte del Subprograma de Prevención al Delito, que se lleva a cabo en el Distrito Federal, se han entregado miles de alarmas vecinales a ciudadanos de las diferentes demarcaciones. Entre el año 2004 y en lo que va del año 2005, se han distribuido aproximadamente 50 mil aparatos de este moderno sistema, el cual se conecta a la Central de Monitoreo de Seguridad Pública con el fin de ofrecer una respuesta inmediata ante cualquier llamado de emergencia por parte de la población, el único costo es de la llamada telefónica hacia la central de la Secretaría de Seguridad Pública.

El inconveniente de esta alarma vecinal es que para que funcione adecuadamente debe de haber coordinación por parte de los vecinos y alguien debe darse cuenta de alguna situación anormal para que se active esta.

1.3.1 Elementos de una alarma de casa habitación

Una alarma puede tener desde unos componentes básicos hasta una gran cantidad de componentes, de esto depende el costo de la misma. A continuación se describen algunos de ellos. Además que más adelante existe un apartado completo dedicado a lo que sería la base de las alarmas y que son los sensores.

- Teclado, desde el que se conecta y desconecta la alarma, suele admitir varias claves para saber quién ha desconectado la alarma, también dispone de teclas auxiliares, como la tecla de pánico que dispara la alarma de forma inmediata.
- Central, es donde van todas las conexiones, suele estar próxima al teclado, y es una caja blindada con sensores de apertura, en caso de ser abierta o atacada dispara la alarma. Dentro aloja el cerebro de la alarma.
- Mando remoto con tecla de pánico, consiste en un pequeño mando a distancia, mas pequeño que el que abre las puertas de los garajes, cuya misión es disparar la alarma en caso de necesidad, es usado normalmente por personas que se encuentran solas, y en prevención de ataques personales. También sirven como mandos remotos de la alarma en bancos comercios etc.
- Sirena exterior, suele tener una luz para indicar qué alarma suena, el sonido suele andar por los 110 decibelios, algo realmente molesto y que si en el exterior son muy molestos, en el interior lo son aun más, puede ir acompañada de una luz de destellos.
- Detectores en cristales, disparan la alarma cuando un cristal es violentado, o roto, muy frecuente en cristales de comercios, actúan de forma preventiva al disparar la alarma antes de que entren los ladrones, conviene ajustar la sensibilidad para evitar disparos de la alarma innecesarios.
- Detectores sísmicos, de humo, fotoeléctricos, de calor, de movimiento, de humedad, de presencia.... son detectores que se colocan en determinadas zonas o para prevenir determinados riesgos, aparte de los normales, se suelen instalar en función de la necesidad del cliente.

- Detectores, los hay de varios tipos, infrarrojos, de movimientos, volumétricos, de sonido, y la conexión con la central puede ser por cable, por radio, normalmente los detectores incorporan varias tecnologías a la vez para evitar disparos de la alarma injustificados.
- Mantenimiento, suele ser mediante contrato, pactando una revisión anual, o semestral de los diferentes equipos adquiridos, para comprobar que los mismos siguen en perfecto estado, comprueban las diferentes baterías, cambiando aquellas que no mantiene un nivel aceptable de carga.
- Receptora de alarmas, la falta de sensibilidad de la gente hacia las sirenas de las alarmas ha hecho verdaderamente necesario este servicio, además en determinadas comunidades autónomas, se sanciona con fuertes multas los disparos accidentales de alarmas si no se tiene una receptora de alarmas que la desconecte en caso de falsa alarma. Estas receptoras, suelen llamar al domicilio o comercio protegido, y solicitan una clave, en caso de no contestar o clave incorrecta comunican directamente con la policía o envían a sus propios vigilantes. Es bastante útil, y el pago, a diferencia del resto de elementos es anual.

1.3.2 Sensores

Son los encargados de controlar e informar de las variaciones que se producen en su campo de acción y determinar si éstas están dentro del margen de tolerancia o constituyen una indicación exacta de que se han modificado las condiciones de trabajo normales.

Se dividen en dos grupos: Normal Abierto (NA) ó Normal Cerrado (NC). Esto se refiere al estado que tendrán sus contactos en condiciones de vigilancia. Un sensor normal cerrado, presentará sus contactos cerrados durante todo el tiempo que no detecte anomalía alguna, y los abrirá cuando detecte una modificación en sus condiciones naturales de trabajo. Un ejemplo sería el caso de una barrera infrarroja, que en condiciones normales tiene sus contactos de salida cerrados, en el momento de interrumpirse el haz infrarrojo, los contactos de su salida se abrirán hasta que termine la interrupción.

Entre los sensores más utilizados en los equipos comerciales figuran:

a) Detectores magnéticos de apertura.

También llamado microswitch o magnético; basa su funcionamiento en un microcontacto, generalmente ubicado dentro de una ampollita de vidrio, éste microswitch se activa cuando se le acerca un campo magnético; por eso este sensor consta también de un imán; recuerde que la mayoría de los sistemas de alarma funciona con sensores NC (normalmente cerrado), es decir que el modo de reposo NC del microswitch se mantiene siempre y cuando el imán se encuentre a su lado; en el momento que el imán se aleja, el contacto del microswitch se abre activando la zona de alarma.

El montaje correcto para estos sensores será entonces: el microswitch (que posee dos cables o bornes), deberá ser fijado al marco de la puerta o ventana; y el imán deberá ser montado

sobre la parte móvil de éstas, (hoja de la puerta o ventana), de manera tal que se acerquen lo suficiente al cerrarse las aberturas figura 1.1.

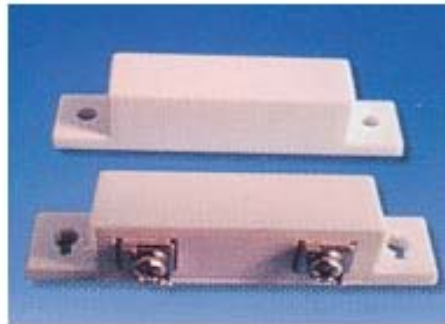


Figura 1.1. Detector magnético de apertura para interior.

Otra alternativa son los magnético de aplicación exterior figura 1.2, cuya fijación por medio de tornillos o autoadhesivo. Conexión por chicote de cables, o bornes.



Figura 1.2. Detector magnético de apertura para exterior.

b) Barreras infrarrojas.

Este sensor llamado barrera infrarroja, está compuesto por dos elementos: el emisor, que emite un haz de luz infrarroja en línea recta; éste además posee el borne de conexión de alimentación; el segundo elemento es el receptor de dicho haz que para funcionar correctamente debe estar perfectamente alineado y enfrentado con el emisor, cerrando de esta forma el circuito a NC; el receptor tiene los bornes correspondientes a la alimentación y a los contactos NC (normalmente cerrado).

Al igual que con los otros sensores hay diferentes marcas y modelos con diversas características técnicas, aquí resumiremos con algunos ejemplos:

- 1) Barrera infrarroja para interior, simple haz, alcance de unos 10 metros.(Figura 1.3)



Figura 1.3. Barrera de simple haz.

2) Barrera infrarroja para interior de embutir, doble haz sincronizado, alcance 10 metros. (Figura 1.4)



Figura 1.4. Barrera de doble haz.

3) Barrera infrarroja: alcance hasta 50 mts., uso interior y exterior (con ciertos reparos), para montaje en poste o en pared, doble haz sincronizado, inmunidad a interferencia infrarroja proveniente de la luz solar u otras fuentes. Óptica de alineación incorporada. Son muy costosos (Figura 1.5)



Figura 1.5. Barrera de doble haz, alcance hasta 50 mts.

4) Barrera para exterior de triple haz, inmune a luz solar o reflejos, Alcance 25 mts., altura 33 cms. Se adapta a cualquier clima, los fabricantes ofrecen variados modelos con alcance hasta 50 mts. y de 1.8 mts de altura. (Fig 1.6)



Figura 1.6. Barrera de triple haz.

c) Detectores de movimiento o infrarrojo pasivos.

El "pasivo" es un sensor de movimiento, basado en rayos infrarrojos. Éste detecta movimientos de elementos que irradian cierta temperatura (calor); para ello despliega un haz de rayos infrarrojos en forma de abanico, que según el modelo puede variar tanto en el largo del haz (alcance), como en los ángulos horizontales y verticales. (Fig 1.7)

Existe gran variedad de modelos según características técnicas, tales como diferentes ángulos de incidencia, filtros de luz, termo compensaciones para evitar la detección de mascotas, microondas, doble sistema óptico, para funcionar en la intemperie, etc., etc.



Figura 1.7. Sensor de movimiento.

d) Sensores de rotura de vidrios.

Hay básicamente dos modelos de sensor que detectan la rotura de vidrios (por ejemplo una ventana), uno consta de un micrófono conectado a un circuito electrónico calibrado a la frecuencia de ruido que genera un vidrio rompiéndose, requiere alimentación; el segundo modelo suele ser un dispositivo mecánico sensible a las vibraciones, es regulable, generalmente va adherido directamente al vidrio y no necesita alimentación, (éste puede detectar vibración o golpes, aunque el vidrio no se haya roto), ambos tienen un contacto NC.

Ahora veremos (figuras 1.8 y 1.9) qué forma presentan algunos modelos de sensores de rotura de vidrios con micrófono incorporado



Figura 1.8. Tipo 1 de sensor de rotura de vidrios.



Figura 1.9. Tipo 2 de sensor de rotura de vidrios.

Los modelos que no tienen micrófonos están compuestos por una cinta metálica que se adhiere al vidrio en todo su contorno (este sistema es antiguo y está perdiendo la preferencia de los instaladores).

e) Interruptores de mercurio.

Estos sensores están formados, básicamente, por una pequeña ampolla de vidrio, que contiene una pequeña cantidad de mercurio (metal líquido). En uno de los extremos de la ampolla se alojan dos contactos internos con su correspondiente terminal exterior.

La ampolla de vidrio suele ser de forma alargada y cilíndrica con sus puntas redondeadas, con ésta forma el mercurio puede desplazarse de un extremo al otro, libremente; logrando así cerrar o no el circuito según en qué posición se encuentre la ampolla.(Figura 1.10)

Es de uso muy común en las alarmas para vehículos que se disparan al elevarse el auto con una grúa

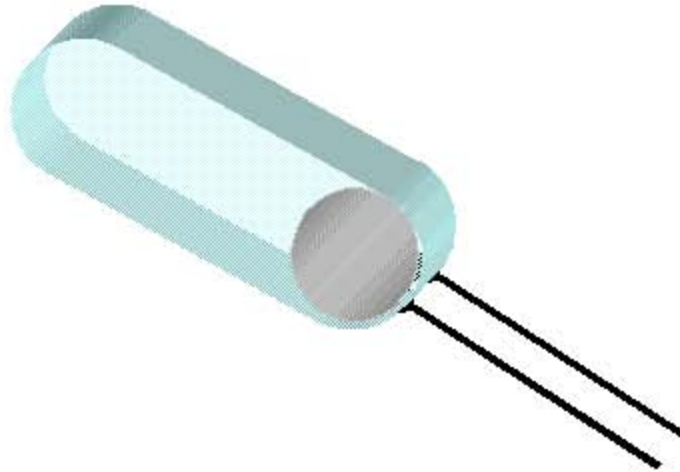


Figura 1.10. Interruptor de mercurio.

f) Detectores de humo.

Este sensor está diseñado para detectar un aumento considerable de humo en el aire circundante al mismo, para ello hay diferentes técnicas: desde cambios fotoeléctricos hasta variaciones en la ionización del aire, esto en algunos casos se logra con la utilización de algún material radioactivo (americio), en el circuito electrónico. (son los más comunes)

Al igual que con los otros sensores hay diferentes marcas y modelos con diversas características técnicas, aquí resumiremos con dos ejemplos.

En la figura 1.11 un detector fotoeléctrico de humo (detecta por variación de luz) y en la figura 1.12 un detector iónico de humo, (generalmente son inalámbricos y por lo tanto mucho más costosos que los cableados).



Figura 1.11. Detector fotoeléctrico de humo.



Figura 1.12. Detector iónico de humo.

Hasta aquí hemos desarrollado una muy rápida (pero completa) visión de los principales sensores que se utilizan en las alarmas domiciliarias.

CAPÍTULO 2

EL SERVICIO DE MENSAJES CORTOS DEL SISTEMA GLOBAL DE COMUNICACIONES MÓVILES

2.1. Introducción

El sistema GSM es el sistema de comunicación móvil digital de 2ª generación basado en células de radio. Apareció para dar respuestas a los problemas de los sistemas analógicos. Fue diseñado para la transmisión de voz por lo que se basa en la conmutación de circuitos.

El servicio de mensajes cortos, SMS, tiene gran auge en nuestros días, siendo mucho más utilizado de lo que inicialmente se estimó. En este tema se describe el servicio SMS desde un punto de vista práctico, haciendo hincapié en cómo es posible realizar aplicaciones que utilicen este servicio.

Se parte de una breve descripción teórica de la red GSM en general y la arquitectura de red subyacente[1]. Después se describe el servicio SMS así como la arquitectura de red y los protocolos empleados, centrándose en el protocolo SM-TP, el cual utiliza el nivel de aplicación para enviar/recibir mensajes cortos esto para familiarizarse con la terminología y tener una idea básica de lo que está pasando. Más información se puede encontrar en [2]

Se muestra cómo es la interfaz entre las aplicaciones y el servicio SMS utilizándose un módem GSM para tener acceso a ellas, y cómo es posible controlar este módem mediante los comandos AT y AT+ [3, 4, 5].

2.2. Arquitectura de red GSM

El objetivo de este capítulo es describir el Servicio de Mensajes Cortos (**SMS**, **Short Message Service**) a nivel de aplicación, sin entrar en demasiados detalles de la red subyacente.

El Sistema Global de Comunicaciones Móviles (**GSM**, **Global System for Mobile Communication**) es un sistema que ofrece los siguientes servicios:

- Transmisión/recepción de voz
- Transmisión/recepción de datos
- Transmisión/recepción de mensajes cortos **SMS**

Todas las redes GSM se pueden dividir en tres partes fundamentales y bien diferenciadas como lo podemos observar en la figura 2.1.

1.- Subsistema Radio (RSS, Radio SubSystem). Cubre la comunicación entre las estaciones móviles (**MS**, **Mobile Station**) y las estaciones base (**BS**). La interfaz radio entre ellas se denomina **Um**. La estación móvil consta a su vez de dos elementos básicos que debemos conocer, por un lado la terminal o equipo móvil (**ME**, **Mobile Equipment**) y por otro lado el módulo de identidad del suscriptor (**SIM**, **Subscriber Identity Module**). Con respecto a las terminales poco tenemos que decir ya que las hay para todos los gustos, lo que si tenemos que comentar es que la diferencia entre unas y otras radica fundamentalmente en la potencia que tienen que va desde los 20 vatios (generalmente instalados en vehículos) hasta los 2 vatios de nuestras terminales.

El SIM es una pequeña tarjeta inteligente que sirve para identificar las características de nuestra terminal. Esta tarjeta se inserta en el interior del móvil y permite al usuario acceder a todos los servicios que haya disponibles por su operador, sin la tarjeta SIM la terminal no nos sirve de nada porque no podemos hacer uso de la red. El SIM está protegido por un número de cuatro dígitos que recibe el nombre de **PIN** o **Personal Identification Number**.

La mayor ventaja de las tarjetas SIM es que proporcionan movilidad al usuario ya que puede cambiar de terminal y llevarse consigo el SIM aunque todos sabemos que esto en la práctica en muchas ocasiones no resulta tan sencillo. Una vez que se introduce el PIN en la terminal, la terminal va a ponerse a buscar redes GSM que estén disponibles y va a tratar de validarse en ellas, una vez que la red (generalmente la que tenemos contratada) ha validado nuestra terminal el teléfono queda registrado en la célula que lo ha validado.

2.- Subsistema de estaciones base (BSS, Base Station Subsystem), incluido dentro de la parte Radio, sirve para conectar a las estaciones móviles con los Subsistemas de Conmutación y Red, además de ser los encargados de la transmisión y recepción. Como los MS también constan de dos elementos diferenciados:

- a) Estación base emisora/receptora (**BTS**, **Base Transceiver Station**): Procesa los canales radio (Interfaz Um). Consta de emisor, receptor y antena, usadas en cada célula de la red y que suelen estar situadas en el centro de la célula, generalmente su potencia de transmisión determinan el tamaño de la célula.

- b) Controlador de Estación Base (**BSC, Base Station Controller**). Los BSC se utilizan como controladores de los BTS y tienen como funciones principales las de estar al cargo de los controles de las frecuencias de radio de los BTS, mapeo de canales radio sobre los canales terrestres. Por un lado se comunica con las BTS a través de un interfaz con canales de 16kbts/s (**Abis**), y por otro lado se comunica con los MSC a través de un interfaz con canales de 64kbts/s(**A**).

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

- Centro de conmutación de servicios móviles (**MSC, Mobile Services Switching Center**): Es el componente central del NSS y se encarga de realizar las labores de conmutación dentro de la red, así como de proporcionar conexión con otras redes.
- Centro de conmutación e interconectado de servicios móviles (**GMSC, Gateway Mobile Services Switching Center**): Un gateway es un dispositivo traductor. Puede ser software o hardware que se encarga de interconectar dos redes haciendo que los protocolos de comunicaciones que existen en ambas redes se entiendan. Bien, la misión del GMSC es esta misma, es decir, servir de mediador entre las redes de telefonía fijas y la red GSM
- Registro de locación primario (**HLR, Home Location Register**): El HLR es una base de datos que contiene información sobre los usuarios conectados a un determinado MSC. Entre la información que almacena el HLR tenemos fundamentalmente la localización del usuario y los servicios a los que tiene acceso. El HLR funciona en unión con en VLR que vemos a continuación.
- Registro de locación de visitante (**VLR, Visitor Location Register**): contiene toda la información sobre un usuario necesaria para que dicho usuario acceda a los servicios de red. Forma parte del HLR con quien comparte funcionalidad.
- Centro de autenticación (**AuC, Authentication Center**): Proporciona los parámetros necesarios para la autenticación de usuarios dentro de la red; también se encarga de soportar funciones de encriptación.
- Registro de identidad de equipo (**EIR, Equipment Identity Register**): También se utiliza para proporcionar seguridad en las redes GSM pero a nivel de equipos válidos. La EIR contiene una base de datos con todas las terminales que son válidos para ser usados en la red. Esta base de datos contiene las identidades internacionales de equipos móviles (**IMEI, International Mobile Equipment Identity**) de cada terminal, de manera que si un determinado móvil trata de hacer uso de la red y su IMEI no se encuentra localizado en la base de datos del EIR no puede hacer uso de la red.

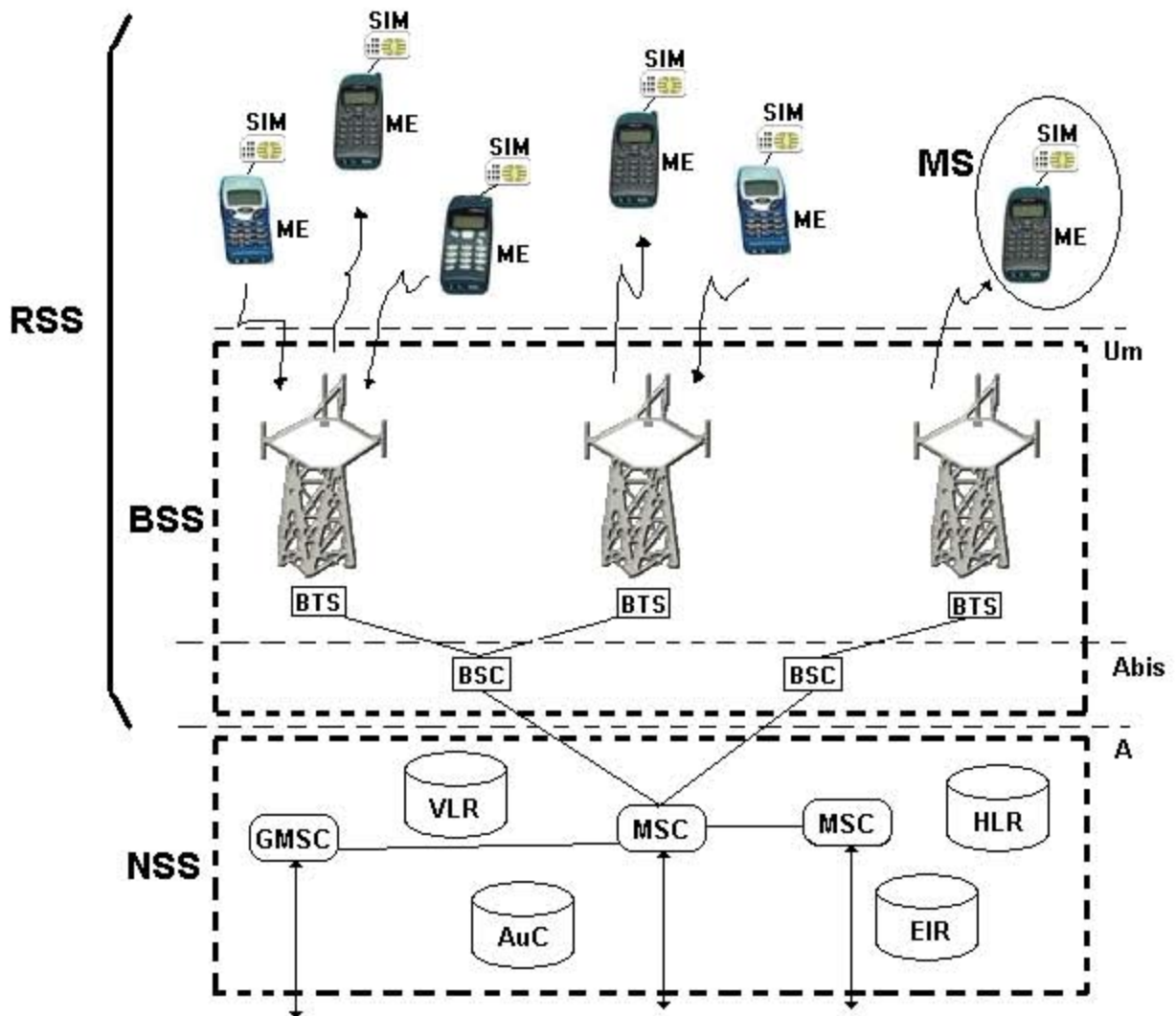


Figura 2.1. Arquitectura de red GSM.

2.3 Arquitectura celular del Sistema GSM

El sistema GSM está basado en una arquitectura celular que permite una mejor utilización del espectro a base de replicar unas unidades fundamentales o celdas, con las que se teje el área que se quiere cubrir. El tamaño de las celdas es un parámetro de diseño que se calcula en base al número medio de usuarios (tráfico medio generado) y el porcentaje de utilización de la estación base. Cuanto menor sea el radio de las celdas, mayor cantidad de usuarios puede soportar el sistema, sin embargo mayor es el coste en infraestructura.

En las ciudades el radio es pequeño (100m-1km) mientras que en las zonas menos pobladas el radio es mayor, cubriéndose con una única estación base una mayor superficie (Hasta 30km por celda).

En la figura 2.2 se muestra un ejemplo de un trozo de un sistema celular, en el que las células tienen el mismo tamaño. Todas las estaciones base (**BS**) están interconectadas a través del subsistema de red (**NSS**).

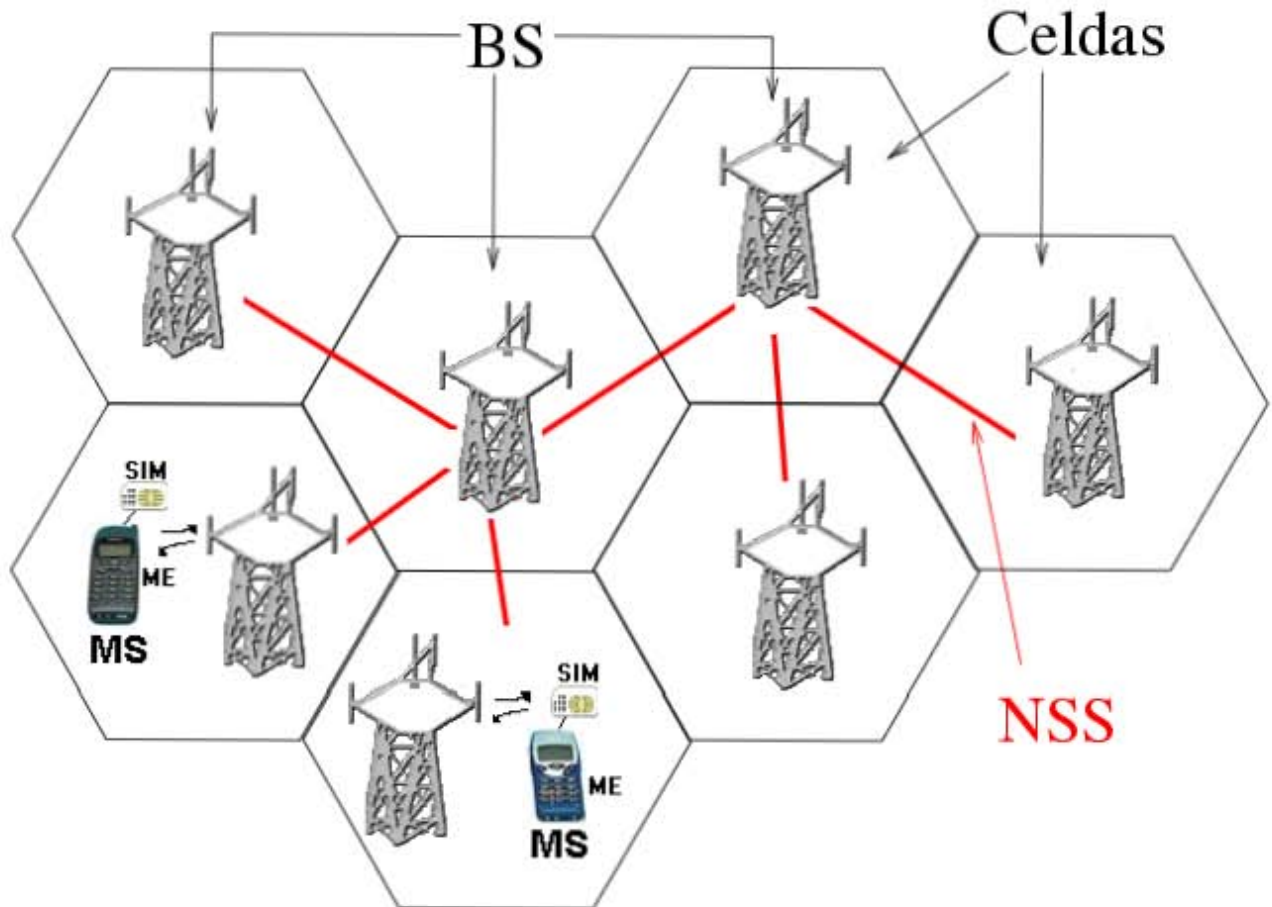


Figura 2.2. Sistema Celular.

2.4 Interfaz Radio (Um)

Para la transmisión de "bits" entre la estación base y una estación móvil se utilizan canales físicos, caracterizados por un número de ranura y una portadora. Dentro de cada portadora se multiplexan en el tiempo 8 ranuras, formando una trama de acceso múltiple por división de tiempo (**TDMA, Time Division Multiple Access**). (Figura 2.3)

A un nivel superior, los **canales físicos** se dividen en :

- **Canales de tráfico:** Llevan la voz y/o los datos
- **Canales de Control:** señalización y señales de control.

Los canales de tráfico pueden ser de 2.4, 4.8 ó 9.6kb/s. **Para el servicio SMS se utilizan canales de control.**

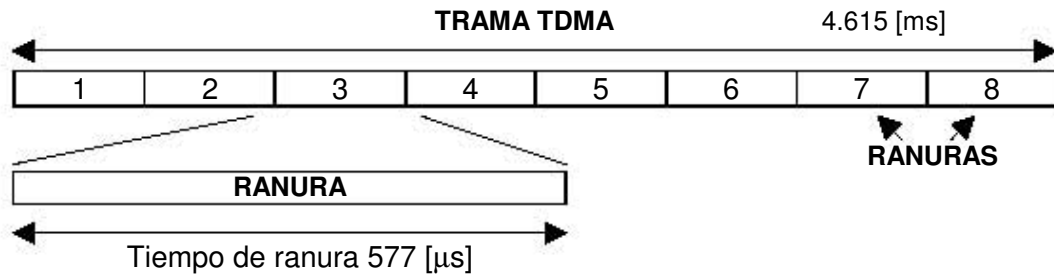


Figura 2.3. Trama TDMA y RANURAS.

2.5 Servicio SMS

El **servicio SMS** permite transferir un mensaje de texto entre una estación móvil (**MS**) y otra entidad de mensajes cortos (**SME, Short Message Entity**) a través de un centro de servicio (**SC, Service Center**) (Figura 2.4)

El servicio final ofrecido es una **comunicación extremo-extremo entre la estación móvil (MS) y la entidad (SME)**. La entidad puede ser otra estación móvil o puede estar situado en una red fija. En el caso de envío de un mensaje entre dos móviles, ambas partes son estaciones móviles. Cuando se envía un mensaje para solicitar algún tipo de servicio (o realizar alguna votación, sobre todo en los concursos de la TV, que ahora están tan de moda), un extremo es una estación móvil y la otra es un servidor que atiende las peticiones (o anota los votos).

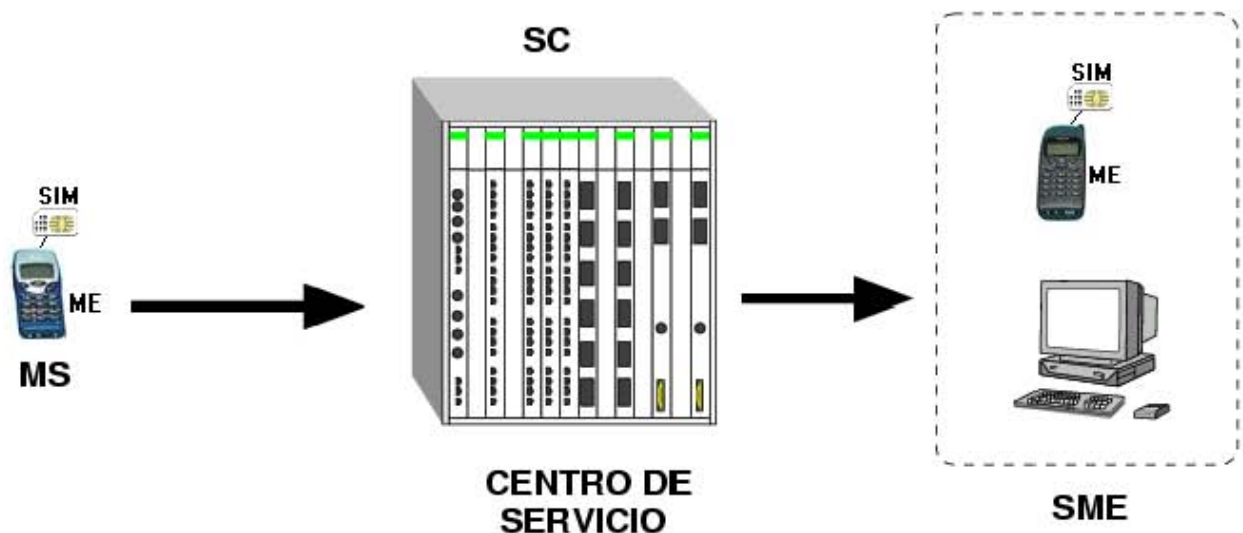


Figura 2.4. Servicio SMS.

En la norma **GSM** sólo se especifica la parte de comunicaciones entre las estaciones móviles (**MS**) y el Centro de servicio. La comunicación entre el Centro de Servicio y las entidades fijas queda fuera del ámbito de esta norma (Figura 2.5).

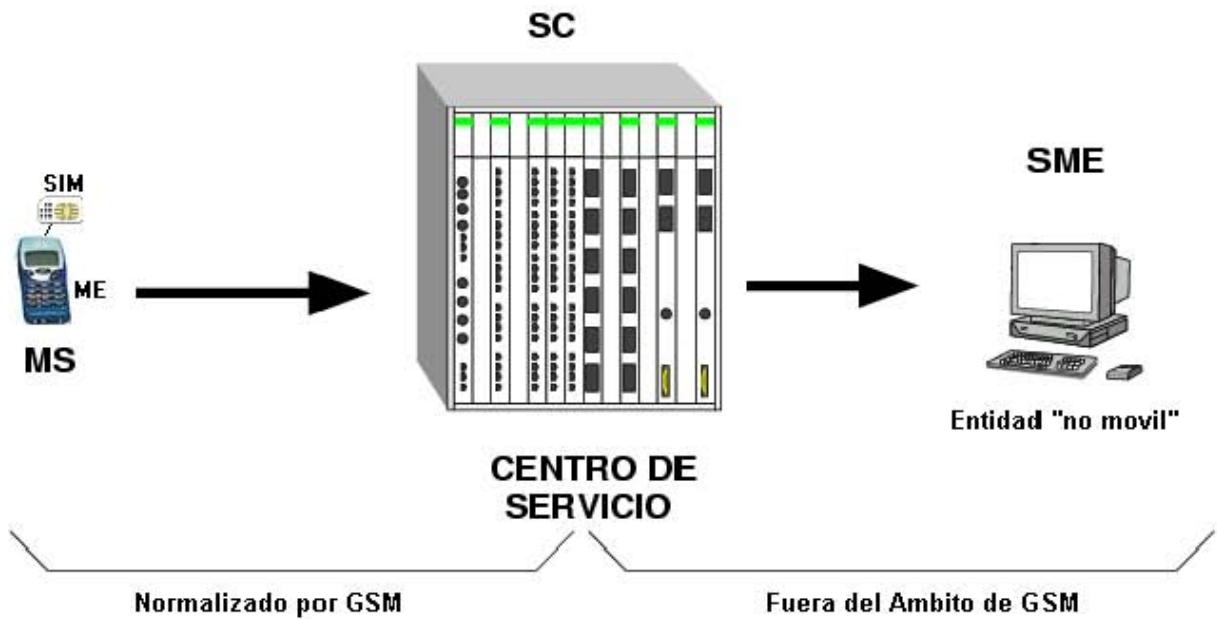


Figura 2.5. Envío de un SMS entre una MS y una entidad fija.

El servicio SMS se divide en dos servicios Básicos:

1. **Mensaje corto enviado desde el equipo móvil (SM MO, Short Message Mobile Originated Point-to-Point)**. Servicio de entrega de un mensaje desde una MS hasta un SC, obteniéndose un informe sobre lo ocurrido.(Figura 2.6)



Figura 2.6. Servicio básico SM MO.

2. **Mensaje corto entregado al equipo móvil (SM MT, Short Message Mobile Terminated Point-to-Point)**. Servicio de entrega de un mensaje desde el SC hasta una MS, obteniéndose un informe sobre lo ocurrido.(Figura 2.7)

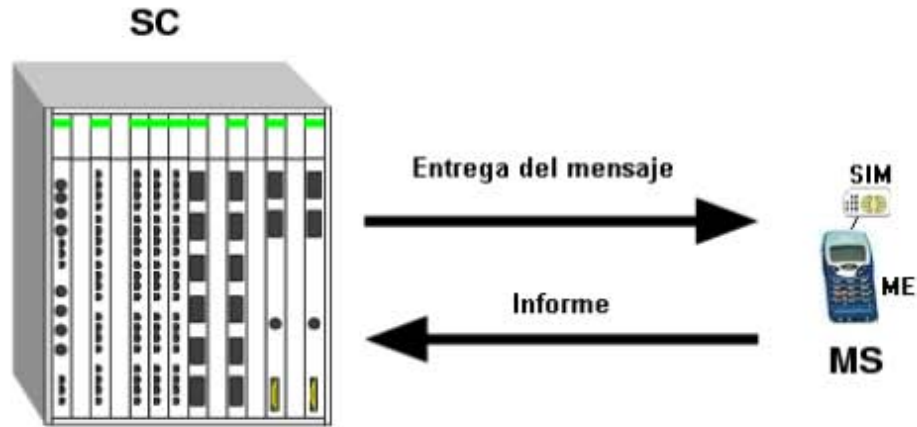


Figura 2.7. Servicio básico SM MT.

2.6 Estructura de red para el Servicio SMS

La estructura básica de la red para el servicio SMS se muestra en la figura 2.8. Las entidades involucradas son las siguientes:

- **MS:** Estación móvil.
- **MSC:** Centro de conmutación.
- **SMS-GMSC:** Centro de conmutación e interconectado de servicios móviles, pasarela para el servicio de mensajes cortos (Servicio SM MT).
- **SMS-IWMSC (Interworking Mobile Services Switching Center):** Centro de conmutación e interconexión entre PLMN y el SC (Servicio SM MO).
- **SC:** Centro de Servicio.
- **HLR, VLR** Bases de datos.

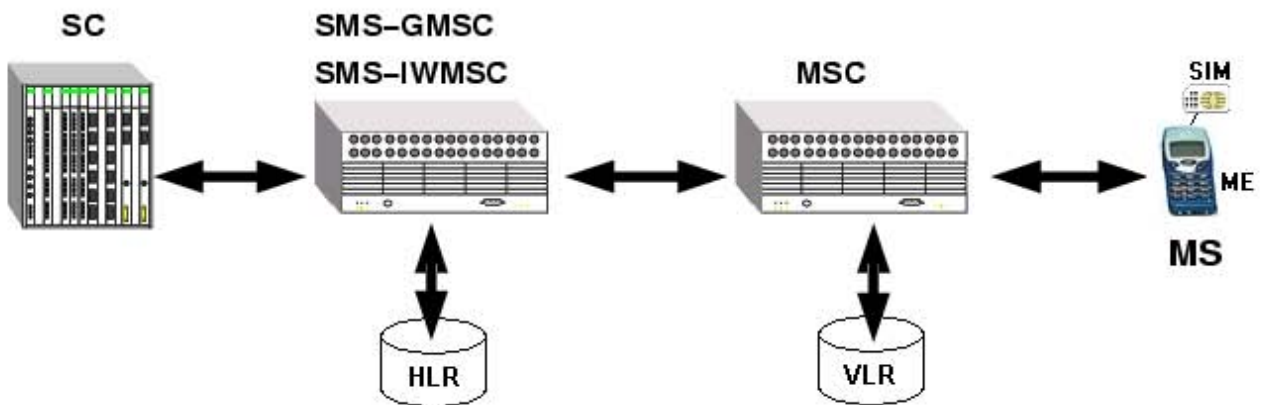


Figura 2.8. Estructura básica de la red para la transferencia de mensajes cortos.

Para la descripción detallada de la arquitectura, se utiliza un modelo de capas, en el que cada capa o nivel proporciona un servicio a la capa superior, y este servicio se implementa mediante el protocolo correspondiente. La arquitectura se divide en 4 capas(Figura 2.9):

- **SM-AL (Short Message Application Layer): Nivel de aplicación.** Utiliza la interfaz hombre maquina (**MMI, Man Machine Interface**) basada en los comandos AT que veremos mas adelante.
- **SM-TL (Short Message Transfer Layer): Nivel de transferencia.** Servicio de transferencia de un mensaje corto entre una MS y un SC (en ambos sentidos) y obtención de los correspondientes informes sobre el resultado de la transmisión. Este servicio hace abstracción de los detalles internos de la red, permitiendo que el nivel de aplicación pueda intercambiar mensajes.
- **SM-RL (Short Message Relay Layer): Nivel de repetición.** Proporciona un servicio al nivel de transferencia que le permite enviar las Unidades de Datos del Protocolo de Transporte (**TPDU Transport Protocol Data Units**) a su entidad gemela.
- **SM-LL (Short Message Lower Layers): Niveles inferiores.**

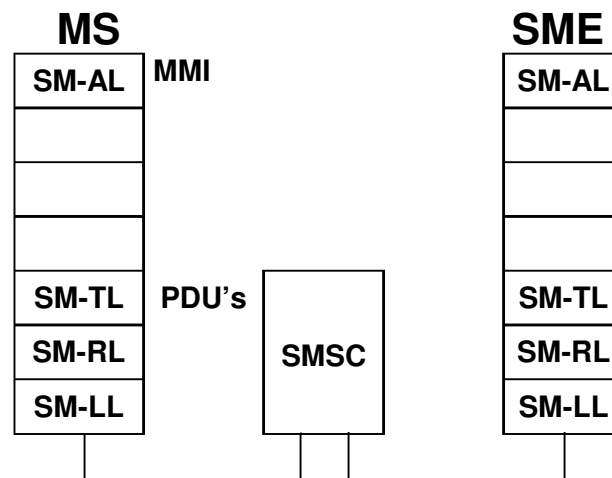


Figura 2.9. Niveles y servicios para el envío de mensajes cortos.

2.7 Nivel SM-TL y protocolo SM-TP

Cada capa proporciona los servicios a la capa superior utilizando un protocolo. Se definen los protocolos **SM-TP** y **SM-RP**, que se corresponden con las capas **SM-RL** y **SM-TL**. El nivel de interés de este trabajo es el **SM-TL**, que es el que se usará para enviar y recibir SMS.

El servicio proporcionado por la **capa SM-TL** permite al nivel de aplicación enviar mensajes a su entidad gemela, recibir mensajes de ella así como obtener informes sobre el estado de transmisiones anteriores.

Se utilizan los siguientes 6 protocolos de unidades de datos (**PDU**) (Figura 2.10):

- **SMS-DELIVER**: Transmitir un mensaje desde el **SC** al **MS**
- **SMS-DELIVER-REPORT**: Error en la entrega (si lo ha habido)
- **SMS-SUBMIT**: Transmitir un mensaje corto desde el **MS** al **SC**
- **SMS-SUBMIT-REPORT**: Error en la transmisión (Si lo ha habido)
- **SMS-STATUS-REPORT**: Transmitir un informe de estado desde el **SC** al **MS**
- **SMS-COMMAND**: Transmitir un comando desde el **MS** al **SC**

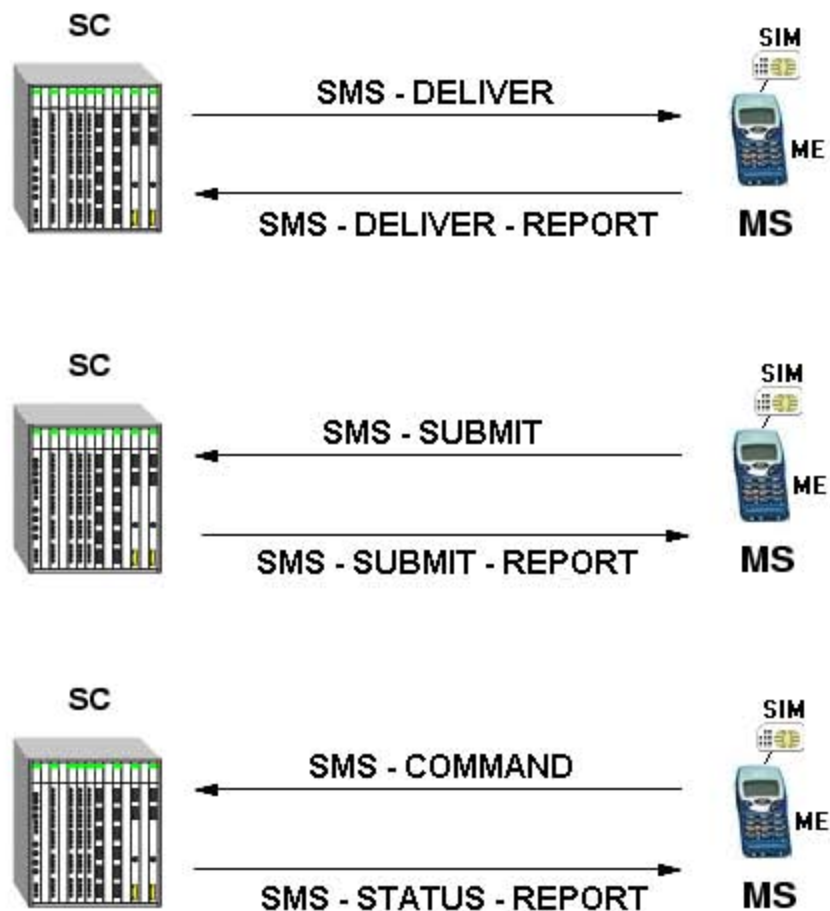


Figura 2.10. Las 6 PDU's del SM-TP.

2.7.1 SMS-SUBMIT (Mensaje corto originado desde el móvil)

La estructura de la PDU **SMS-SUBMIT** se muestra en la figura 2.11. Los campos que la componen son los siguientes:

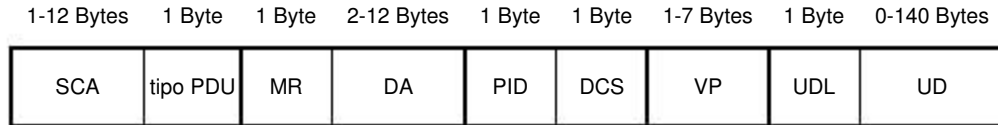


Figura 2.11. Trama SMS-SUBMIT.

- **SCA(Service Center Address):** Número de teléfono del Centro de Servicio (SC). La estructura detallada se muestra en la figura 2.12. Consta de los siguientes campos:

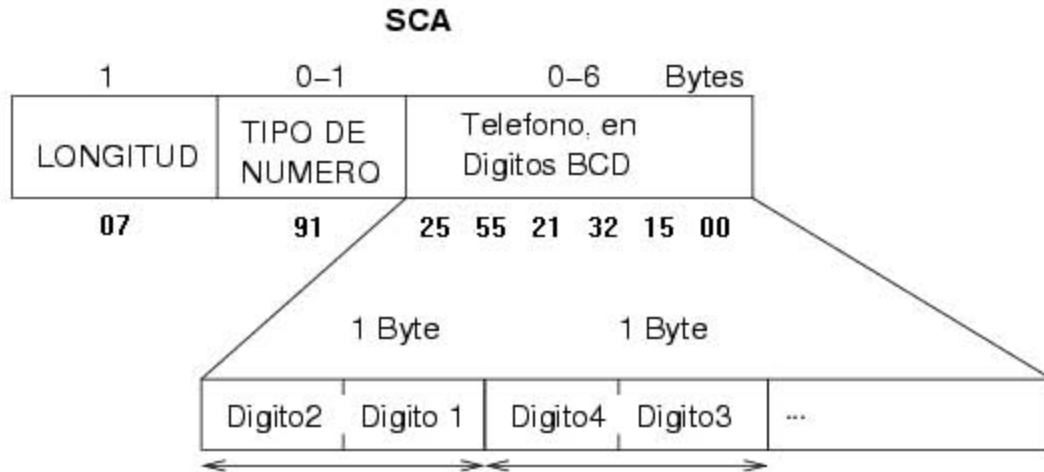


Figura 2.12. Detalle del campo SCA.

- **Longitud:** Número de dígitos del teléfono del SC.
- **Tipo de número:** Indica si se trata de un número nacional o internacional:
 - ♦ 80h: De manera inherente toma el SC que tiene el teléfono GSM
 - ♦ 81h: Nacional
 - ♦ 91h: Internacional
- **Dígitos BCD:** Número de teléfono del SC, en dígitos BCD, si el número contiene un número de dígitos impar, el último dígito deberá ser llenado con "Fh"

Ejemplo:

Si se tiene el número de SC +52 5512235100 tendríamos que escribir lo siguiente para el SCA

0791255521321500

- **PDU-Type(Protocol Data Unit Type):** Contiene información sobre el tipo de PDU se escribe en formato hexadecimal después de configurar los parámetros binarios figura 2.13.

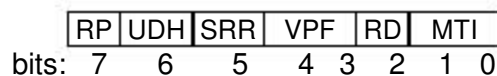


Figura 2.13. Detalle del campo tipo PDU.

- **RP:** Bit 7 indica si existe camino de respuesta. RP=0 si no RP=1
- **UDHI:** Bit 6 indica si el campo datos de usuario UD contiene sólo el mensaje corto (UDHI=0) o si existe una cabecera antes del mensaje corto (UDHI=1)
- **SRR:** Bit 5 informe de estado no solicitado (SRR=0) o sí solicitado (SRR=1)
- **VPF:** Bit 4 y Bit 3 para indicar si el campo **VP** está o no presente

Bit 4	Bit 3	Descripción
0	0	VP no presente
0	1	Reservado
1	0	VP representado valor relativo
1	1	VP representado valor absoluto

- **RD:** Bit 2 indica si se rechazan o no duplicados. RD=0 instruye al SC para aceptar mensajes cortos repetidos, si RD=1 instruye al SC de rechazar mensajes repetidos.
- **MTI:** Bit 1 y Bit 0 indica el tipo de mensaje corto:

Bit 1	Bit 0	Descripción
0	0	SMS-DELIVER
0	0	SMS-DELIVER-REPORT
0	1	SMS-SUBMIT
0	1	SMS-SUBMIT-REPORT
1	0	SMS-STATUS-REPORT
1	0	SMS-COMMAND
1	1	Reservado

- **MR(Message Reference):** Parámetro para la referencia del mensaje este es un número entero de 0 a 255 creado por la estación móvil MS y enviado al SC.
- **DA(Destination Address):** Dirección del SME destino (número de teléfono)fig 2.14

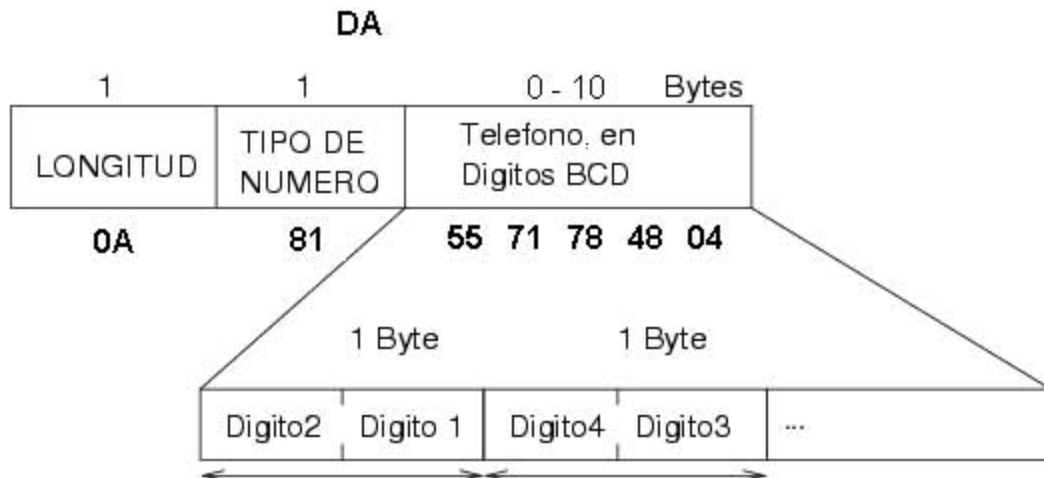


Figura 2.14. Detalle del campo DA.

- **Longitud:** Contiene el número de dígitos BCD.
- **Tipo de número:** Indica si se trata de un número nacional o internacional:
 - ◆ 81h: Nacional
 - ◆ 91h: Internacional
- **Dígitos BCD:** Número de teléfono al que se envía el mensaje corto, en dígitos BCD, si el número contiene un número de dígitos impar, el último dígito deberá ser llenado con “Fh”

Ejemplo:

Si se quiere enviar al número +5517878404 tendríamos que escribir lo siguiente para el DA.

0A815571784804

- **PID(Protocol Identifier):** Identificación del protocolo es la información por medio de la cual el nivel de transferencia SM-TL se refiere a la capa superior de transporte, para indicar como será tratado el PDU, para nuestro caso usamos 00h que indica que será tratado como mensaje corto.
- **DCS(Data Coding Scheme):** Identificación del tipo de codificación dentro de los datos de usuario, el alfabeto por inherencia indica que los datos de usuario están codificados utilizando el alfabeto de 7-bit por lo que DCS =00h, este alfabeto lo podemos ver en el **apéndice A**.

Quando este alfabeto es usado, 8 caracteres del mensaje son empacados en siete octetos, y el mensaje puede consistir en hasta 160 caracteres.

- **VP(Validity Period):** Periodo de validez del mensaje es un elemento de información que proporciona la estación móvil MS al SC para incluir un tiempo específico de validez del mensaje corto.
- **UDL(User Data Length):** La longitud del campo de datos de usuario nos da una representación entera del número de caracteres en el siguiente campo que es el de datos de usuario UD.
- **UD(User Data):** Los datos de usuario son los caracteres del mensaje escrito codificados según vemos en la figura 2.15. Tanto UDL como UD para el mensaje “Hola” nos quedaría de la siguiente forma:

Ejemplo:

Si se quiere enviar mensaje Hola tendríamos que escribir lo siguiente para el UD.

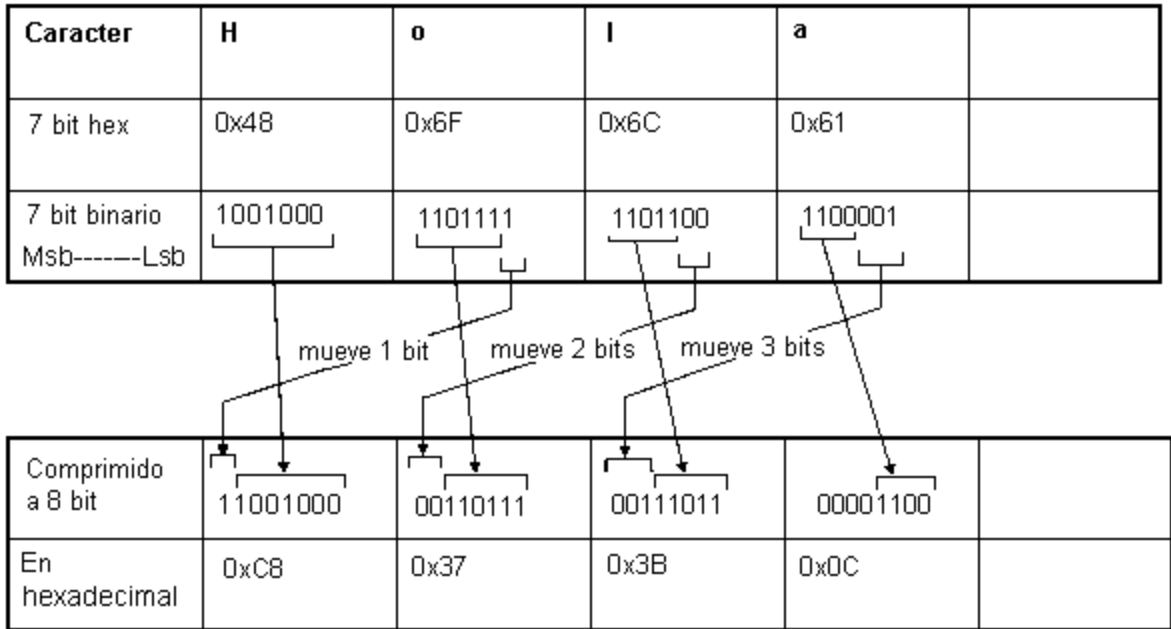


Figura 2.15. Codificación de 7-bit.

2.7.2 SMS-DELIVER (Mensaje corto entregado al móvil)

Esta trama, transmitida desde el **SC** hasta el **MS**, tiene una estructura similar a **SMS-SUBMIT** y se muestra en la figura 2.16.

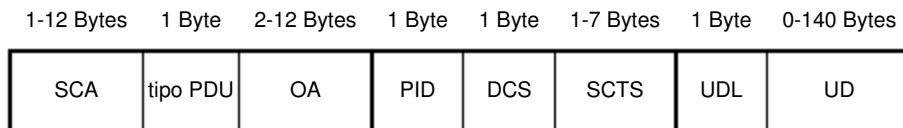


Figura 2.16. Trama SMS-DELIVER.

Los nuevos campos que aparecen son los siguientes:

- **OA(Originator Address):** Dirección del SME que envía el mensaje, se rige por las mismas reglas que el DA .
 - **Longitud:** Contiene el número de dígitos BCD.
 - **Tipo de número:** Indica si se trata de un número nacional o internacional:
 - ◆ 81h: Nacional
 - ◆ 91h: Internacional

- **Dígitos BCD:** Número de teléfono al que se envía el mensaje corto, en dígitos BCD, si el número contiene un número de dígitos impar, el último dígito deberá ser llenado con "Fh"

Ejemplo:

Si se recibe un mensaje del teléfono +5515965150 tendríamos lo siguiente para el OA.

0C91255551691505

- **SCTS(Service Center Time Stamp):** Marca de tiempo de cuando el centro de servicio recibió el mensaje.

AÑO	MES	DIA	HORA	MINUTO	SEGUNDO	ZONA DE TIEMPO
SCTS						

Ejemplo:

Si se recibe un mensaje el 03 de Febrero del 2005 a las 17:31:28 en la misma zona tendríamos lo siguiente para el SCTS.

50 20 30 71 13 82 00

2.7.3 Un ejemplo de trama SMS-SUBMIT

Se quiere enviar el mensaje corto "hola" al teléfono 630672901 utilizando el Centro de mensajes +341710760000.

- **SCA:** 0C91437101670000 (8 bytes)

Longitud	Tipo	Teléfono en BCD
0C	91	43-71-01-67-00-00

- **PDU-TYPE:** 01h. Trama de tipo **SMS-SUBMIT**. Campo de usuario sin cabecera. Informe de estado no solicitado. Campo VP no presente.

7	6	5	4 3	2	1 0
RP	UDHI	SRR	VPF	RD	MTI
0	0	0	00	0	01

- **MR:** 00h. Número de referencia 0.
- **DA:** 0681366027091F (7 bytes). Teléfono destino.

Longitud	Tipo	Teléfono en BCD
09	81	36-60-27-09-F1

- **PID:** 00h (mensaje corto)
- **DCS:** F6h (Codificación de 8 bits, en ASCII)
- **UDL:**04. Longitud de los datos de usuario.
- **UD:** 686F6C61 (4 bytes). Datos de usuario.

h	o	l	a
68	6F	6C	61

La trama final es la mostrada en la figura 2.17, que ocupa 24 bytes.

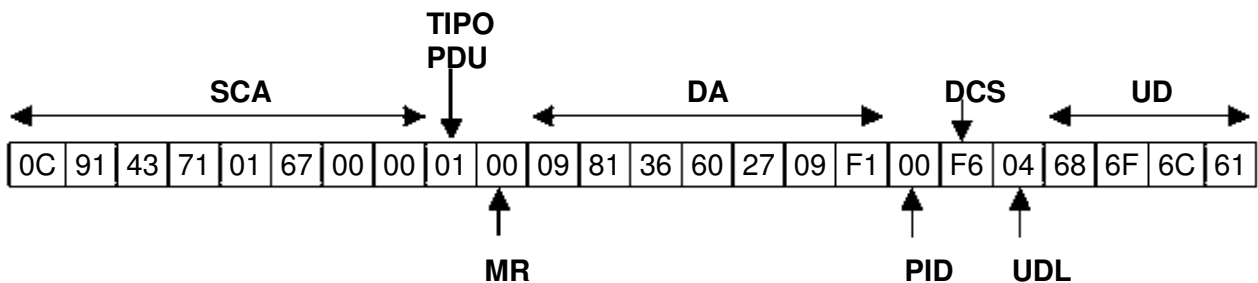


Figura 2.17. Trama SMS-SUBMIT de ejemplo.

2.8 Acceso a los Servicios SMS

Actualmente están apareciendo gran cantidad de servicios basados en mensajes cortos. Además de ser usados para enviar mensajes de texto entre personas, se están ofreciendo otros servicios como son:

- Votaciones mediante SMS
- Suscripción a servicios de información, como por ejemplo el ofrecido durante estos mundiales, que te envían un SMS cada vez que la selección Española marca un gol.
- Informe de averías en ciertos equipos. Por ejemplo, muchos cajeros automáticos envían un SMS al servicio técnico cuando detectan que hay alguna avería o les falta algún recurso: dinero, papel...

Para poder ofrecer estos servicios es necesario diseñar software y hardware que pueda acceder a los servicios SMS. Esto se puede conseguir de varias maneras:

1. Algunos teléfonos se pueden conectar directamente a un PC y mediante un software propietario se puede acceder a los datos de móvil (agenda, tarjeta SIM...), así como enviar y recibir mensajes SMS. El principal problema de esta solución es que no es abierta, y los fabricante no proporcionan suficiente información como para poder realizar aplicaciones con ellos. Es necesario realizar ingeniería inversa.
2. Utilización de un MÓDEM GSM.

2.9 Protocolo de comunicación RS232

La comunicación serial, como su nombre lo indica, realiza la transferencia de información enviando o recibiendo datos descompuestos en bits, los cuales viajan secuencialmente uno tras otro. Está compuesta principalmente de dos elementos básicos, el hardware, que hace referencia a la configuración de los conectores y niveles de voltaje, y el software, con el que se controla la información binaria que se quiere transferir. Todo esto está regido por normas o protocolos donde el utilizado por las computadoras convencionales es el protocolo RS-232.

El PC utiliza la norma RS232 para la comunicación serial, por lo que los niveles de tensión de los pines están comprendidos entre +15 y -15 voltios. El método utilizado en las computadoras para la comunicación serial, consiste en agregar marcadores dentro del flujo de bits para ayudar a seguir cada bit de datos.

Al introducir un bit de inicio que indica el inicio de un corto flujo de datos, la posición de cada bit puede ser determinado cronometrando los bits a intervalos regulares, enviando bits de inicio al comienzo de cada flujo de 8 bits los dos sistemas pueden no estar sincronizados por una señal de reloj, lo único que es importante es que ambos sistemas estén configurados a la misma velocidad.

Cuando el dispositivo receptor de la comunicación recibe el bit de inicio comienza un temporizador de corto tiempo. Al mantener los flujos cortos no hay suficiente tiempo para que el reloj salga de sincronía. Este método es conocido como comunicación asíncrona debido a que las terminales de la comunicación no están sincronizados por una línea de señal.

Se llama comunicación serial asíncrona porque el receptor se resincroniza el mismo con el transmisor usando el bit de inicio de cada trama. Los caracteres se pueden transmitir en cualquier tiempo, con un retraso de tiempo arbitrario entre caracteres. Existen también protocolos de comunicación serial síncrona donde los caracteres se envían en bloques sin una trama de bits circundante. En esta aproximación, el transmisor continuamente transmite señales, con un caracter de sincronización especial que se transmite si no hay datos reales disponibles para transmitir.

Cada flujo de bits es separado en grupos de 5 a 8 bits llamados palabras. Usualmente en el ambiente de la computación se encuentran palabras de 7 y 8 bits, el primero es para acomodar todas las letras mayúsculas y minúsculas del alfabeto en código ASCII, con un total de 127

caracteres. Las palabras de 8 bits son utilizadas para corresponder a un byte. Por convención, el bit menos significativo de la palabra es enviado primero y el más significativo al final. Durante la comunicación, el transmisor codifica cada palabra agregándole un bit de inicio al principio y 1 o 2 bits al final. Algunas ocasiones agregará un bit de paridad entre el último bit de la palabra y el primer bit de paro, esto es utilizado como verificación de integridad de datos. Al paquete de bits que se transmiten usualmente se le llama marco de datos.

Se pueden utilizar 5 tipos diferentes de bits de paridad:

- Bit marcador de paridad) siempre es un 1 lógico.
- Bit de paridad de espacio) siempre es un 0 lógico.
- Bit de paridad par es puesto en 1 lógico si cuando al contar el número de bits en la palabra el resultado es par.
- Bit de paridad non es puesto en 1 lógico si cuando al contar el número de bits en la palabra el resultado es non.
- Marco sin bit de paridad, ó sin paridad, es cuando se elimina el bit de paridad del marco.

Otro factor importante de toda señal serial asíncrona es la velocidad de comunicación (baudios), que es la velocidad a la que los datos son transmitidos. Las velocidades a las que se envía la información comienzan en los 50 bits por segundo (bps) y casi de manera estándar se van duplicando 50, 100, 150, 300, 600, 1200, 2400, 4800, 9600, 19200.... hasta 384,000.

Una conexión RS-232 con velocidad de 9600 baudios tiene la capacidad de enviar 9600 bits de datos en 1 segundo. Si se pueden enviar 9600 bits en un segundo, como máximo, el inverso de 9600 dará como resultado el tiempo de bit (período de un bit).

$$\frac{1}{\text{Velocidad_de_baudios}} = \text{tiempo_de_bit} = \frac{1}{9600} = 104\mu\text{s}$$

Si un receptor y un transmisor se conectan a 9600 baudios, el transmisor enviará bits de datos cada 104µs, y el receptor tomará lectura de los bits de datos cada 104µs.

Según se conforme el marco de datos es como se referencia al marco mismo. Por ejemplo: Palabras de 8 bits, con paridad Non y 1 bit de paro es conocido como 8N1, y la trama a una velocidad de 9600 baudios, codificada sin retorno a cero (NRZ) es como se muestra en la siguiente figura 2.18.

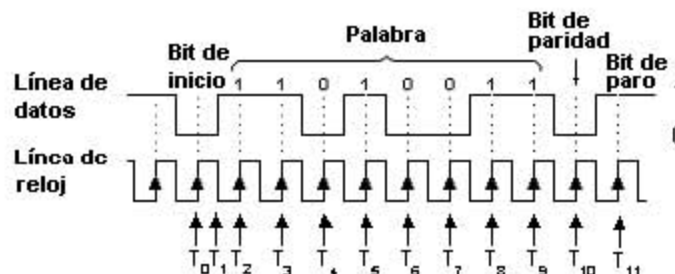


Figura 2.18 Marco de datos de comunicación serial asíncrona.

T₀: La transición del nivel marca a espacio indica al receptor que la transmisión de un nuevo carácter ha comenzado.

T₁: El receptor espera la mitad del tiempo de bit (a 9600 baudios este tiempo es 52 μ s) y toma otra muestra de la línea. Si la línea sigue en el nivel espacio, el bit de inicio es válido. En otro caso, si la línea de recepción regresa al nivel de marca, se trata de un bit de inicio falso que se atribuye a una línea ruidosa.

T₂: El receptor espera un tiempo de bit y muestrea la línea de entrada, el nivel será el correspondiente al bit menos significativo.

T₃-T₉: Se realizan 7 muestreos más, cada 104 μ s (para 9600 baudios), y se obtienen los niveles correspondientes a los bits de datos restantes. Después de T₉ los 8 bits de datos han sido capturados.

T₁₀: Se muestrea el bit de paridad, en este caso cero por la paridad non.

T₁₁: Se muestrea el bit de paro, observe que la línea regresa al nivel de marca.

En este proyecto se emplea la comunicación serial asíncrona, en la cual la transmisión está inactiva en el estado de Marca (1 lógico). La transmisión de cada carácter en una línea de comunicación asíncrona va precedida de un bit de inicio. El bit de inicio es un Espacio (0 lógico) con duración igual al tiempo de bit. En el receptor, cuando la línea cambia de Marca a Espacio se interpreta como el bit de inicio, después de este bit se reciben los bits de datos con un tiempo de bit igual a 104 μ s, si la transmisión es a 9600 baudios.

Después de que el último bit de datos ha sido enviado, el transmisor pasa al nivel de Marca durante un tiempo de bit. Este bit es llamado bit de paro. El bit de paro indica que todos los bits de datos han sido enviados y la transmisión del carácter se ha completado. Si el receptor detecta un bit de inicio, después los bits de datos pero no detecta el nivel de Marca al final, esto indica un error en la transmisión.

Es de tener en cuenta una herramienta muy básica para el manejo y ensayo de la transmisión serial que es el HYPERTERMINAL DE WINDOWS: Este programa es una herramienta útil para trabajar con comunicaciones seriales, ya sea para comunicarnos con otra computadora, con un microcontrolador, con un multímetro, con un módem y en fin, con cualquier dispositivo que esté diseñado para enviar o recibir información a través de este medio.

2.9.1 Señales del Protocolo RS232

La siguiente tabla muestra el conector de 9 pines serie que se encuentra en la mayoría de las PC de hoy en día y sus respectivas señales.

SEÑAL	PIN No.
Carrier Detect	1
Receive Data	2
Transmit Data	3
Data Terminal Ready	4
Signal Ground	5
Data Set Ready	6
Request To Send	7
Clear To Send	8
Ring Indicator	9

- **Request To Send (RTS)** Esta señal se envía de la computadora (DTE) al módem (DCE) para indicar que se quieren transmitir datos. Si el módem decide que esta OK, asiente por la línea CTS. Una vez la computadora prende la señal RTS, esperará que el módem asiente la línea CTS. Cuando la señal CTS es afirmado por el módem, la computadora empezará a transmitir datos.
- **Clear To Send (CTS)** Afirmado por el módem después de recibir la señal de RTS indica que la computadora puede transmitir.
- **Data Terminal Ready (DTR)** Esta línea de señal es afirmada por la computadora, e informa al módem que la computadora está lista para recibir datos.
- **Data Set Ready (DSR)** Esta línea de señal es afirmada por el módem en respuesta a una señal de DTR de la computadora. La computadora supervisa el estado de esta línea después de afirmar DTR para descubrir si el módem esta encendido.
- **Receive Signal Line Detect (RSLD)** Esta línea de control es afirmada por el módem e informa a la computadora que se ha establecido una conexión física con otro módem. A veces se conoce como detector de portadora (CD). sería un error que una computadora transmita información a un módem si esta línea no esta prendida, es decir si la conexión física no funciona.
- **Transmit Data (TD)** es la línea por donde el dato se transmite de un bit a la vez.
- **Receive Data (RD)** es la línea por donde el dato se recibe de un bit a la vez.

2.10 Interfaz con MODEMS: comandos AT

La comunicación con los módems se realiza a través de una **línea serie**, y dependiendo del módem, se pueden usar los niveles definidos por la norma RS232 (Módems para PC) o niveles TTL (Módems para Circuito impreso).

El estándar "de facto" para controlar los módems se basa en los **comandos AT HAYES**, o más comúnmente conocidos como **comandos AT**. El módem, antes de realizar una conexión con otro módem, se encuentra en modo comando. En este modo podemos configurar y controlar el módem utilizando los **comandos AT**. Una vez establecida la conexión con un módem remoto, se pasa del modo comando al modo conexión, por lo que la información que le llega al módem por la línea serie no es interpretada como **comandos AT** sino como información a transmitir. Una vez terminada la conexión el módem vuelve al modo comando.

Los **comandos AT** son cadenas ASCII que comienzan por los caracteres **AT** y terminan con un retorno de carro. Cada vez que el módem recibe un comando, lo procesa y devuelve un resultado, que normalmente es una cadena ASCII salvo que hayamos indicado lo contrario.

Al estar la comunicación en ASCII, pondremos utilizar un terminal de comunicaciones desde un ordenador para acceder al módem, bien para configurarlo, bien para hacer pruebas o bien para establecer una comunicación con otro módem. En la figura 2.19 se muestra un ejemplo de comunicación entre un módem y un PC con un terminal de comunicaciones abierto. El usuario teclea el comando "ATZ" seguido de ENTER. El módem interpreta este comando, que es de inicialización y devuelve la cadena "OK" seguida de un retorno de carro para indicar que se ha ejecutado con éxito.

A continuación se listan algunos comandos AT. Se puede encontrar más información en [4, 5]:

- **ATA**: Responder a una llamada entrante
- **ATD**: Llamar a un número de teléfono.
- **ATE**: Eco de comandos ON/OFF
- **ATF**: Seleccionar modo de conexión
- **ATH**: Colgar / descolgar
- **ATI**: Obtener identificadores
- **ATL**: Volumen del altavoz
- **ATM**: Control del altavoz
- **ATS**: Lectura / escritura de registros
- **ATZ**: Reset del módem

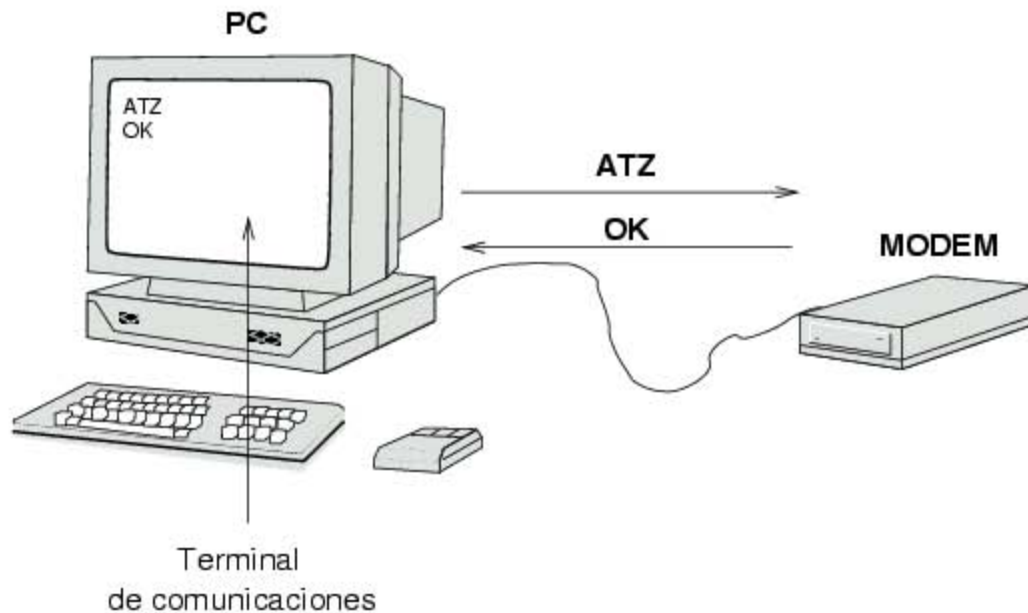


Figura 2.19. Conexión entre un MODEM y una PC.

2.11 Interfaz con Módems GSM

Los módems GSM no sólo se comportan de forma muy parecida a un módem normal, permitiendo el intercambio de datos con otro módem y utilizándose los comandos AT originales, sino que incluyen muchas más características. Son como pequeños teléfonos móviles, que incluyen su propia tarjeta SIM para poder funcionar y por tanto permiten gestionar la base de datos de teléfonos, la lista de los mensajes SMS recibidos, enviar mensajes SMS, configurar diversos parámetros, etc.

Para tener acceso a todos esos servicios, y dado que los comandos AT estaban muy extendidos y muy estandarizados, se ha realizado una ampliación, añadiéndose nuevos comandos. Estos nuevos comandos comienzan por las letras AT+, y se denominan **comandos AT+**.

2.11.1 Comandos AT+

En este apartado se listan algunos de los comandos AT+ implementados en los módems GSM para tener una idea de lo que se puede controlar a través del módem, aunque existen muchos más comandos.

1. Comandos generales

- a) **AT+CGMI**: Identificación del fabricante
- b) **AT+CGSN**: Obtener número de serie

- c) **AT+CIMI**: Obtener el IMSI.
- d) **AT+CPAS**: Leer estado del módem

2. Comandos del servicio de red

- a) **AT+CSQ**: Obtener calidad de la señal
- b) **AT+COPS**: Selección de un operador
- c) **AT+CREG**: Registrarse en una red
- d) **AT+WOPN**: Leer nombre del operador

3. Comandos de seguridad:

- a) **AT+CPIN**: Introducir el PIN
- b) **AT+CPINC**: Obtener el número de reintentos que quedan
- c) **AT+CPWD**: Cambiar password

4. Comandos para la agenda de teléfonos

- a) **AT+CPBR**: Leer todas las entradas
- b) **AT+CPBF**: Encontrar una entrada
- c) **AT+CPBW**: Almacenar una entrada
- d) **AT+CPBS**: Buscar una entrada

5. Comandos para SMS

- a) **AT+CPMS**: Seleccionar lugar de almacenamiento de los SMS
- b) **AT+CMGF**: Seleccionar formato de los mensajes SMS
 - Modo texto
 - Modo PDU
- c) **AT+CMGR**: Leer un mensaje SMS almacenado
- d) **AT+CMGL**: Listar los mensajes almacenados
- e) **AT+CMGS**: Enviar mensaje SMS
- f) **AT+CMGW**: Almacenar mensaje en memoria
- g) **AT+CMSS**: Enviar mensaje almacenado
- h) **AT+CSCA**: Establecer el Centro de mensajes a usar
- i) **AT+WMSC**: Modificar el estado de un mensaje

2.11.2 Algunos ejemplos

A continuación se muestran algunos ejemplos de utilización de los comandos AT+. Para probarlos se ha utilizado un ordenador PC, un módem GSM conectado al puerto serie y una terminal de comunicaciones. El esquema es el mismo que el mostrado en la figura 2.19. Si el sistema operativo empleado es Linux, se puede utilizar el programa minicom. En el caso de Windows, se puede emplear el **HyperTerminal**.

- Lectura de la lista de teléfonos

Para listar todos los teléfonos se utiliza el comando AT+CPBR. Primero comprobamos la capacidad del directorio telefónico:

```
AT+CPBR=?
+CPBR: (1-150),20,14
OK
```

La información devuelta nos indica que el directorio tiene 150 entradas, desde la 1 hasta la 150, y que en cada entrada el teléfono puede ocupar como máximo 20 caracteres y el texto 14. Para listar los teléfonos debemos indicar qué entradas se quieren examinar. Como en este SIM hay 150 como máximo, utilizamos el siguiente comando:

```
AT+CPBR=1,150
+CPBR: 1,"607557556",129,"JAIME"
+CPBR: 2,"915541831",129,"QUINTANAJ"
+CPBR: 3,"914465076",129,"PACO PIO"
+CPBR: 4,"914634436",129,"MARIANO"
+CPBR: 5,"915338584",129,"ANGEL"
+CPBR: 6,"617203473",129,"MOVIL TOR"
+CPBR: 7,"679453250",129,"Virginia"
+CPBR: 8,"629040222",129,"Paloma M"

OK
```

Se listan todos los teléfonos comprendidos entre las entradas 1 y 150. En total hay 8, que ocupan las posiciones 1-8. Para listar las 3 primeras entradas habría que hacer lo siguiente:

```
AT+CPBR=1,3
+CPBR: 1,"607557556",129,"JAIME"
+CPBR: 2,"915541831",129,"QUINTANAJ"
+CPBR: 3,"914465076",129,"PACO PIO"

OK
```

Para una entrada sólo hay que especificar su número:

```
AT+CPBR=7
+CPBR: 7,"679453250",129,"Virginia"
OK
```

- **Listado de mensajes**

Los mensajes cortos se dividen en 5 categorías, cada una identificada por una cadena. Para listar los mensajes se utiliza el comando **AT+CMGL=<categoría>**, donde <categoria> es una cadena de texto que puede valer lo siguiente:

- **"REC UNREAD"**: Mensajes recibidos pero no leídos
- **"REC READ"**: Mensajes recibidos y leídos.
- **"STO UNSEND"**: Mensajes escritos y almacenados pero no enviados.
- **"STO SENT"**: Mensajes enviados
- **"ALL"**: Todos los mensajes

A continuación se leen todos los mensajes:

```
AT+CMGL="ALL"  
+CMGL: 2,"REC READ","609""05/01/27,18:16:51+40"  
Como cliente MoviStar Plus Elección, esta de enhorabuena. Porque desde  
el 18 de enero esta ahorrando un 49 % en sus llamadas de móvil a fijo  
en horario normal  
+CMGL: 3,"REC READ","+34609100609""05/02/01,10:00:16+04"  
Telefónica MoviStar le desea una feliz estancia. Para llamar al CRC  
MoviStar marque +34 609 100 609. Para llamar a su Buzón de Voz marque  
+34 609 123 123  
  
OK
```

Hay dos mensajes recibidos y que han sido leídos.

- **Lectura de un mensaje**

Se utiliza el comando **AT+CMGR=<número>**, donde <número> es el número del mensaje a leer.

```
AT+CMGL=2  
+CMGR: 2,"REC READ","609""05/01/27,18:16:51+40"  
Como cliente MoviStar Plus Elección, esta de enhorabuena. Porque desde  
el 18 de enero esta ahorrando un 49 % en sus llamadas de móvil a fijo  
en horario normal  
  
OK
```

Si se especifica un número de mensaje que no existe se devuelve un mensaje de error:

```
AT+CMGR=1  
ERROR
```

- **Borrar un mensaje**

Se utiliza el comando **AT+CMGD=<número>**, donde <número> hace referencia al número de mensaje a borrar.

```
AT+CMGD=3  
OK
```

Mensaje Borrado. Si ahora se intenta leer:

```
AT+CMGR=3  
ERROR
```

- **Envío de un SMS en modo texto**

Para enviar un mensaje SMS se puede realizar de dos maneras diferentes. Se puede utilizar el **modo texto**, en que sólo hay que indicar el número de teléfono y el contenido del mensaje. Es el módem el que se encarga de generar la trama SMS-SUBMIT correspondiente y enviarla. Este es el modo inherente y el que normalmente se emplea si sólo queremos transmitir un mensaje.

Es posible tener acceso directamente al protocolo **SM-TP**, enviando directamente una trama de tipo SMS-SUBMIT. En este caso se habla de **modo PDU**. Será el nivel de aplicación el que tendrá que generar correctamente la trama SMS-SUBMIT y el módem simplemente la transmitirá.

La configuración del módem para funcionar en uno u otro modo se realiza mediante el comando `AT+CMGF=<modo>`, donde <modo> puede tener los siguientes valores:

- `<modo>=1`: **Modo texto**
- `<modo>=0`: **Modo PDU** (Modo inherente)

Para enviar un mensaje en modo texto, se utiliza el comando **AT+CMGS**. Primero se especifica el número de teléfono, seguido de un carácter retorno carro <CR>. El módem responde enviando el carácter ">" que indica que se puede escribir el mensaje que se quiere enviar. Para delimitar el mensaje hay que enviar el carácter <control-z> (Es el carácter ASCII 26).

Si el mensaje se ha enviado correctamente, devuelve la cadena "+CMGS:<nr>" seguida del OK. El campo <nr> es el número de referencia del mensaje, que se va incrementando, tomando los valores comprendidos entre 0 y 255, cada vez que se envía un sms.

```
AT+CMGS="630672901"<CR>
>Mensaje de prueba <control-z>
+CMGS: 2
OK
```

Puesto que hemos enviado un auto-mensaje (un mensaje SMS con destino el mismo móvil que lo ha originado), al cabo de un cierto tiempo se recibe el mensaje, por lo que aparece en la terminal lo siguiente:

```
+CMTI: "SM",14
```

que indica que se ha recibido un mensaje SMS y se ha almacenado con el número 14. Si ahora leemos el mensaje:

```
AT+CMGR=14
+CMGR: "REC UNREAD","+34630672901""02/06/23,11:57:20+00"
Mensaje de prueba
OK
```

La información que se obtiene es la siguiente. Primero el estado del mensaje, "REC UNREAD", para indicar que es un mensaje nuevo que no se había leído. A continuación el teléfono del remitente, la fecha y la hora en la que se ha recibido y finalmente el mensaje recibido.

Si ahora se vuelve a leer el mensaje, el estado será "REC READ".
En caso de no haber cobertura a la hora de enviar el mensaje, el comando AT+CMGS devuelve la cadena ERROR.

```
AT+CMGS="630672901"<CR>
>Mensaje de prueba <control-z>
ERROR
```

- **Envío de un SMS en modo PDU**

También es posible enviar directamente una trama SMS-SUBMIT. Para ello configuramos el módem para funcionar en modo PDU, con el comando AT+CMGF=0 y después se utiliza el comando AT+CMGS, indicando la longitud de la trama (excluyendo el primer byte)

```
AT+CMGS=16 <CR>
> 000104098136602709F100F604686F6C61 <Control-z>
+CMGS: 8
OK
```

Si el primer byte es 00, no se envía información sobre el centro de mensajes, por lo que el módem toma el que tenga predefinido.

2.12 Selección del dispositivo GSM

Mediante un MÓDEM GSM podemos conectar cualquier sistema digital a la red GSM, no sólo para enviar mensajes SMS sino también para transmitir datos.

Existen dos tipos de MODEMS, según la aplicación que queramos realizar

1. MODEMS para circuito impreso: Son Modems de reducido tamaño y perfectamente apantallados que están preparados para ser incorporados dentro de un circuito impreso y que permiten desarrollar un hardware específico y que no depende de un PC. En la figura 2.20 se muestra la tarjeta DACER, desarrollada para la empresa Pulsar Technologies, que incorpora un módem GSM para circuito impreso. Obsérvese el tamaño del módem en comparación con el disquete.

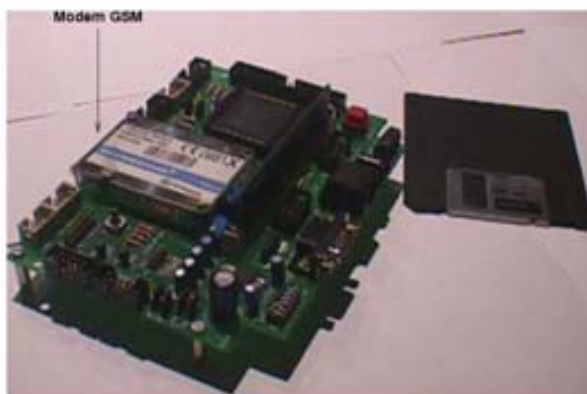


Figura 2.20. Tarjeta con módem GSM integrado.

2. MODEMS para PC. Tienen un tamaño también bastante reducido, y disponen de un conector DB9 hembra para conectarse al PC a través de un cable serie estándar. Son muy útiles para permitir que desde cualquier ordenador de una intranet se puedan enviar mensajes SMS (Ver figura 2.21).



Figura 2.21. Módem GSM para PC.

Pero estos resultan caros así que buscando una mejor opción encontré que casi todos los teléfonos GSM utilizan un MODEM GSM que soporta comandos AT a excepción de la gran mayoría de la marca NOKIA que utilizan otros protocolos de comunicación como el FBUS y el MBUS que utiliza un solo cable para comunicarse. Un teléfono de los más económicos se pueden conseguir desde los \$250.00 pesos, así que resulta conveniente en comparación de los casi \$1000 pesos que se tendrían que pagar por un módulo de los más económicos.

Por lo que al escoger un teléfono GSM solo hay que saber donde están localizados los tres pines necesarios para trabajar con la comunicación asíncrona que son Tx, Rx y la Tierra ver figura 2.22, debido a que por la red existen varias páginas que dicen donde están localizados estos, solo falta ver que teléfono tenemos y en mi caso es un Ericsson modelo A2218z.



Figura 2.22. Conector del Teléfono GSM Ericsson.
De donde el pin 4 es Rx, el pin 5 es Tx y el pin 8 es Tierra.

CAPÍTULO 3

SELECCIÓN DEL MICROCONTROLADOR

3.1 Introducción.

Los microcontroladores están conquistando el mundo. Están presentes en nuestro trabajo, en nuestra casa y en nuestra vida, en general. Se pueden encontrar controlando el funcionamiento de los ratones y teclados de las computadoras, en los teléfonos, en los hornos de microondas y los televisores de nuestro hogar. Pero la invasión acaba de comenzar y el nacimiento del siglo **XXI** será testigo de la conquista masiva de estos diminutos computadores, que gobernarán la mayor parte de los aparatos que fabriquemos y usemos los humanos.

3.2 Controlador y microcontrolador.

Recibe el nombre de **controlador** el dispositivo que se emplea para el gobierno de uno o varios procesos. Por ejemplo, el controlador que regula el funcionamiento de un horno dispone de un sensor que mide constantemente su temperatura interna y, cuando traspasa los límites prefijados, genera las señales adecuadas que accionan los efectores que intentan llevar el valor de la temperatura dentro del rango estipulado.

Aunque el concepto de controlador ha permanecido invariable a través del tiempo, su implementación física ha variado frecuentemente. Hace tres décadas, los controladores se construían exclusivamente con componentes de lógica discreta, posteriormente se emplearon los **microprocesadores**, que se rodeaban con chips de memoria y E/S sobre una tarjeta de circuito impreso. En la actualidad, todos los elementos del controlador se han podido incluir en un chip, el cual recibe el nombre de **microcontrolador**. Realmente consiste en una sencilla pero completa computadora contenida en un circuito integrado.

Un microcontrolador es un circuito integrado de alta escala de integración que incorpora la mayor parte de los elementos que configuran un controlador.

Un microcontrolador dispone normalmente de los siguientes componentes:

- Unidad Central de Proceso o CPU.
- Memoria RAM de usuario.
- Memoria ROM/PROM/EPROM de usuario.
- Líneas de E/S para comunicarse con el exterior.
- Diversos módulos para el control de periféricos (temporizadores, Puertos Serie y Paralelo, CAD: Conversores Analógico/Digital, CDA: Conversores Digital/Analógico, etc.).
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

Los productos que para su regulación incorpora un microcontrolador disponen de las siguientes ventajas:

- Aumento de prestaciones: un mayor control sobre un determinado elemento representa una mejora considerable en el mismo.
- Aumento de la fiabilidad: al reemplazar el microcontrolador por un elevado número de elementos disminuye el riesgo de averías y se precisan menos ajustes.
- Reducción del tamaño en el producto acabado: La integración del microcontrolador en un chip disminuye el volumen, la mano de obra y los stocks.
- Mayor flexibilidad: las características de control están programadas por lo que su modificación sólo necesita cambios en el programa de instrucciones.

El microcontrolador es en definitiva un circuito integrado que incluye todos los componentes de una computadora. Debido a su reducido tamaño es posible montar el microcontrolador en el propio dispositivo al que gobierna. En este caso el microcontrolador recibe el nombre de sistema empotrado (embedded system).

3.3 Diferencia entre microprocesador y microcontrolador.

El microprocesador es un circuito integrado que contiene la Unidad Central de Proceso (CPU), también llamada procesador, de una computadora. El CPU está formado por la Unidad de Control, la Unidad Aritmética y Lógica así como de algunos registros de transferencia.

Los pines de un microprocesador sacan al exterior las líneas de sus buses de direcciones, datos y control, para permitir conectarle con la Memoria y los Módulos de E/S y configurar una computadora implementada por varios circuitos integrados. Se dice que un microprocesador es un sistema abierto porque su configuración es variable de acuerdo con la aplicación a la que se destine. (Figura 3.1.)

MICROPROCESADOR

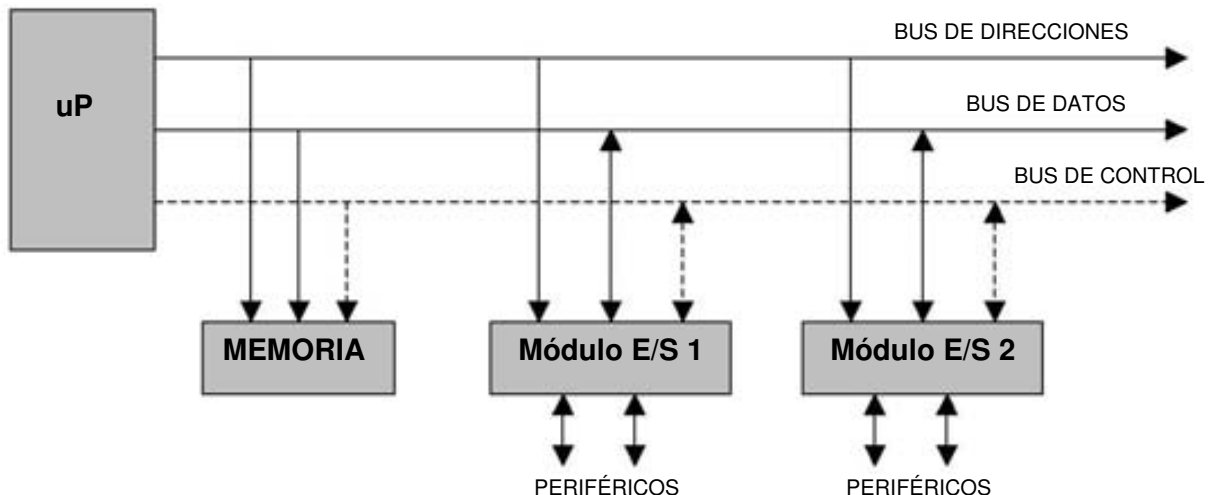


Figura 3.1. Estructura de un sistema abierto basado en un microprocesador. La disponibilidad de los buses en el exterior permite que se configure a la medida de la aplicación.

Si sólo se dispusiera de un modelo de microcontrolador, éste debería tener muy potenciados todos sus recursos para poderse adaptar a las exigencias de las diferentes aplicaciones. Esta potenciación supondría en muchos casos un despilfarro. En la práctica cada fabricante de microcontroladores oferta un elevado número de modelos diferentes, desde los más sencillos hasta los más poderosos. Es posible seleccionar la capacidad de las memorias, el número de líneas de E/S, la cantidad y potencia de los elementos auxiliares, la velocidad de funcionamiento, etc. Por todo ello, un aspecto muy destacado del diseño es la selección del microcontrolador a utilizar. (Figura 3.2.)

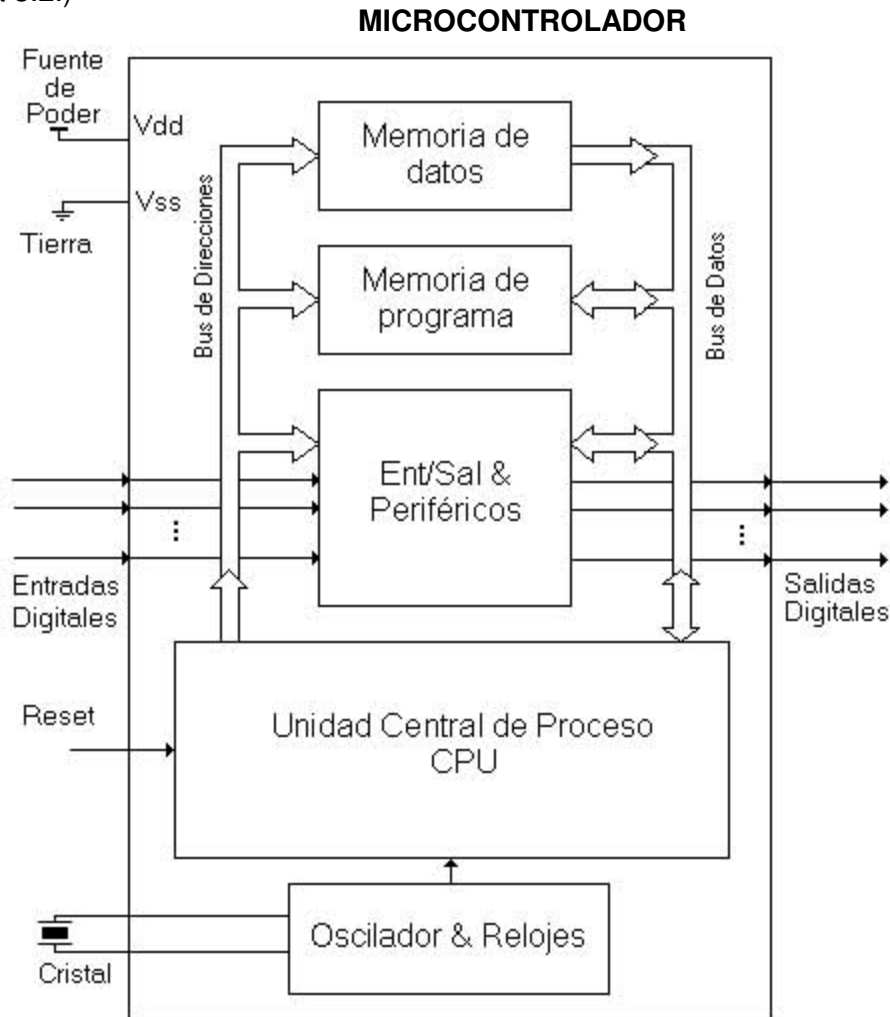


Figura 3.2. El microcontrolador es un sistema cerrado. Todas las partes del computador están contenidas en su interior y sólo salen al exterior las líneas que gobiernan los periféricos.

3.4 Aplicaciones de los microcontroladores.

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

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

Los microcontroladores están siendo empleados en multitud de sistemas presentes en nuestra vida diaria, como pueden ser juguetes, horno de microondas, refrigeradores, televisores, computadoras, impresoras, módems, el sistema de arranque de nuestro coche, etc. Y otras aplicaciones con las que seguramente no estaremos tan familiarizados como instrumentación electrónica, control de sistemas en una nave espacial, etc. Una aplicación típica podría emplear varios microcontroladores para controlar pequeñas partes del sistema. Estos pequeños controladores podrían comunicarse entre ellos y con un procesador central, probablemente más potente, para compartir la información y coordinar sus acciones, como, de hecho, ocurre ya habitualmente en cualquier PC.

3.5 El mercado de los microcontroladores.

Aunque en el mercado de la microinformática la mayor atención la acaparan los desarrollos de los microprocesadores, lo cierto es que se venden cientos de microcontroladores por cada uno de aquellos.

Existe una gran diversidad de microcontroladores. Quizá la clasificación más importante sea entre microcontroladores de 4, 8, 16 ó 32 bits. Aunque las prestaciones de los microcontroladores de 16 y 32 bits son superiores a los de 4 y 8 bits, la realidad es que los microcontroladores de 8 bits dominan el mercado y los de 4 bits se resisten a desaparecer. La razón de esta tendencia es que los microcontroladores de 4 y 8 bits son apropiados para la gran mayoría de las aplicaciones, lo que hace absurdo emplear micros más potentes y consecuentemente más caros. Uno de los sectores que más tira del mercado del microcontrolador es el mercado automovilístico. De hecho, algunas de las familias de microcontroladores actuales se desarrollaron pensando en este sector, siendo modificadas posteriormente para adaptarse a sistemas más genéricos. El mercado del automóvil es además uno de los más exigentes: los componentes electrónicos deben operar bajo condiciones extremas de vibraciones, choques, ruido, etc. y seguir siendo fiables. El fallo de cualquier componente en un automóvil puede ser el origen de un accidente.

En cuanto a las técnicas de fabricación, cabe decir que prácticamente la totalidad de los microcontroladores actuales se fabrican con tecnología CMOS 4 (Complementary Metal Oxide Semiconductor). Esta tecnología supera a las técnicas anteriores por su bajo consumo y alta inmunidad al ruido.

La distribución de las ventas según su aplicación es la siguiente:

- Un **30%** se absorbe en las aplicaciones relacionadas con las computadoras y sus periféricos.
- Un **25%** se utiliza en las aplicaciones de consumo (electrodomésticos, juegos, TV, vídeo, etc.)
- El **16%** de las ventas mundiales se destinó al área de las comunicaciones.
- Otro **16%** fue empleado en aplicaciones industriales.

- El resto de los microcontroladores vendidos en el mundo, aproximadamente un **13%** fueron adquiridos por las industrias de automoción.

También los microcontroladores modernos de 32 bits van afianzando sus posiciones en el mercado, siendo las áreas de más interés el procesamiento de imágenes, las comunicaciones, las aplicaciones militares, los procesos industriales y el control de los dispositivos de almacenamiento masivo de datos.

3.6 Recursos comunes a todos los microcontroladores.

Al estar todos los microcontroladores integrados en un chip, su estructura fundamental y sus características básicas son muy parecidas. Todos deben disponer de los bloques esenciales Procesador, memoria de datos y de instrucciones, líneas de E/S, oscilador de reloj y módulos controladores de periféricos. Sin embargo, cada fabricante intenta enfatizar los recursos más idóneos para las aplicaciones a las que se destinan preferentemente.

En este apartado se hace un recorrido de todos los recursos que se hallan en todos los microcontroladores describiendo las diversas alternativas y opciones que pueden encontrarse según el modelo seleccionado.

3.6.1 Arquitectura básica.

Aunque inicialmente todos los microcontroladores adoptaron la arquitectura clásica de Von Neumann, en el momento presente se impone la arquitectura Harvard. La arquitectura de Von Neumann se caracteriza por disponer de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. A dicha memoria se accede a través de un sistema de buses único (direcciones, datos y control).

La arquitectura Harvard dispone de dos memorias independientes una, que contiene sólo instrucciones y otra, sólo datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias. Figura 3.3.

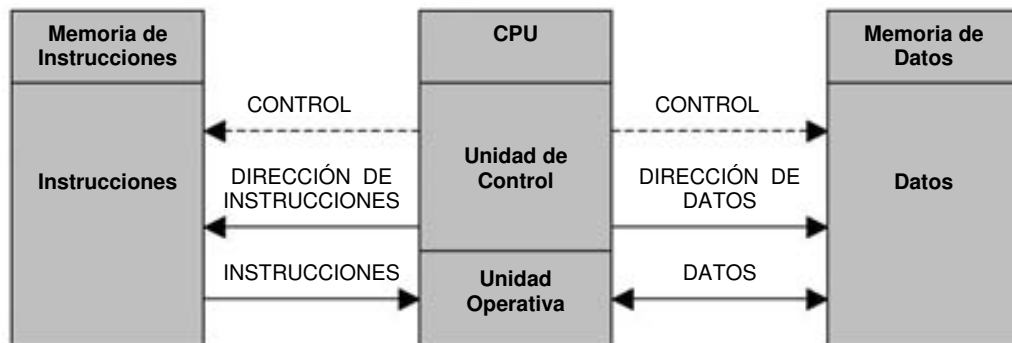


Figura 3.3. La arquitectura Harvard dispone de dos memorias independientes para datos y para instrucciones, permitiendo accesos simultáneos.

Los microcontroladores PIC responden a la arquitectura Harvard.

3.6.2 El microprocesador o CPU.

Es el elemento más importante del microcontrolador y determina sus principales características, tanto a nivel hardware como software.

Se encarga de direccionar la memoria de instrucciones, recibir el código OP de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda 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 usados en los microcontroladores están basados en la filosofía CISC (Computadoras de Juego de Instrucciones Complejo). Disponen de más de 80 instrucciones 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 las computadoras comerciales como la de los microcontroladores están decantándose hacia la filosofía RISC (Computadoras 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 (Computadoras de Juego de Instrucciones Específico).

3.6.3 Memoria.

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

Hay dos peculiaridades que diferencian a los microcontroladores de las computadoras personales:

1. No existen sistemas de almacenamiento masivo como disco duro o disquetes.

2. 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 contener 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 computadoras personales están habituados a manejar Megabytes de memoria, pero, los diseñadores con microcontroladores trabajan con capacidades de ROM comprendidas entre 512 bytes y 8 kbytes 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 describen las cinco versiones de memoria no volátil que se pueden encontrar en los microcontroladores del mercado.

1ª. ROM con máscara

Es una memoria no volátil de sólo lectura cuyo contenido se graba durante la fabricación del chip. El elevado coste 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.

2ª. 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 puede escribir el programa en el chip mediante un sencillo grabador controlado por un programa desde un PC.

La versión OTP es recomendable cuando es muy corto el ciclo de diseño del producto, 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.

3ª EPROM

Los microcontroladores que disponen de memoria EPROM (Erasable Programmable Read Only Memory) pueden borrarse y grabarse muchas veces, pero este número de veces es finito, por lo que no es recomendable una reprogramación continua. La grabación se realiza, como en el caso de los OTP, con un grabador gobernado 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.

4ª EEPROM

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 la de borrado. No disponen de ventana de cristal en la superficie.

Los microcontroladores dotados de 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 confieren 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 finito, por lo que no es recomendable una reprogramación continua. Son muy idóneos para la enseñanza y la Ingeniería de diseño.

Se va extendiendo en los fabricantes la tendencia de incluir una pequeña zona de memoria EEPROM en los circuitos programables para guardar y modificar cómodamente una serie de parámetros que adecuan el dispositivo a las condiciones del entorno.

Este tipo de memoria es relativamente lenta.

5ª FLASH

Se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar. Funciona como una ROM y una RAM pero 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 alternativa FLASH está recomendada frente a la EEPROM cuando se precisa gran cantidad de memoria de programa no volátil. Es más veloz y tolera 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 circuito”, es decir, sin tener que sacar el circuito integrado de la tarjeta. Así, un dispositivo con este tipo de memoria incorporado al control del motor de un automóvil permite que pueda modificarse el programa durante la rutina de mantenimiento periódico, compensando los desgastes y otros factores tales como la compresión, la instalación de nuevas piezas, etc. La reprogramación del microcontrolador puede convertirse en una labor rutinaria dentro de la puesta a punto.

3.6.4 Puertos de Entrada y Salida

La principal utilidad de los pines que posee la cápsula que contiene un microcontrolador es soportar las líneas de E/S que comunican a la computadora interna con los periféricos exteriores.

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

3.6.5 Reloj principal

Todos los microcontroladores disponen de 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.

Generalmente, el circuito de reloj está incorporado en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la frecuencia de trabajo. Dichos componentes suelen consistir 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.

3.7 Recursos Especiales

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

Los principales recursos específicos que incorporan los microcontroladores son:

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

3.7.1 Temporizadores o “Timers”

Se emplean para controlar períodos 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 algunos de los pines del microcontrolador, el mencionado registro se va incrementando o decrementando al ritmo de dichos impulsos.

3.7.2 Perro guardián o “Watchdog”

Cuando la computadora personal se bloquea por un fallo del software u otra causa, se pulsa el botón del reset y se reinicializa el sistema. Pero un microcontrolador funciona sin el control de un supervisor y de forma continuada las 24 horas del día. El Perro guardián consiste en un temporizador que, cuando se desborda y pasa por 0, provoca un reset automáticamente en el sistema.

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

3.7.3 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.

3.7.4 Estado de reposo ó de bajo consumo

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

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

3.7.5 Conversor A/D (CAD)

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

3.7.6 Conversor D/A (CDA)

Transforma los datos digitales obtenidos del procesamiento de la computadora en su correspondiente señal analógica que saca al exterior por uno de los pines de la cápsula. Existen muchos dispositivos que trabajan con señales analógicas.

3.7.7 Comparador analógico

Algunos modelos de microcontroladores disponen internamente de un Amplificador Operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por una de 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.

3.7.8 Modulador de anchura de pulsos o PWM

Son circuitos que proporcionan en su salida pulsos de anchura variable, que se ofrecen al exterior a través de los pines del encapsulado.

3.7.9 Puertos de E/S digitales paralelo

Todos los microcontroladores destinan algunos de sus pines a soportar 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.

3.7.10 Puertos de comunicación serie

Con objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos, algunos modelos disponen de recursos que permiten directamente esta tarea, entre los que destacan:

- UART, adaptador de comunicación serie asíncrona.
- USART, adaptador de comunicación serie sincrónica 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 conmutación multiplexada desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles. En EE.UU. se usa el J1850.

3.8 Herramientas para el desarrollo de aplicaciones.

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

Las principales herramientas de ayuda al desarrollo de sistemas basados en microcontroladores son:

- Desarrollo del software:

Ensamblador. La programación en lenguaje ensamblador puede resultar un tanto ardua para el principiante, pero permite desarrollar programas muy eficientes, ya que otorga al programador el dominio absoluto del sistema. Los fabricantes suelen proporcionar el programa ensamblador de forma gratuita y en cualquier caso siempre se puede encontrar una versión gratuita para los microcontroladores más populares.

Compilador. La programación en un lenguaje de alto nivel (como el C) permite disminuir el tiempo de desarrollo de un producto. No obstante, si no se programa con cuidado, el código

resultante puede ser mucho más ineficiente que el programado en ensamblador. Las versiones más potentes suelen ser muy caras, aunque para los microcontroladores más populares pueden encontrarse versiones demo limitadas e incluso compiladores gratuitos.

- **Depuración:** debido a que los microcontroladores van a controlar dispositivos físicos, los desarrolladores necesitan herramientas que les permitan comprobar el buen funcionamiento del microcontrolador cuando es conectado al resto de circuitos.

Simulador. Son capaces de ejecutar en un PC programas realizados para el microcontrolador. Los simuladores permiten tener un control absoluto sobre la ejecución de un programa, siendo ideales para la depuración de los mismos. Su gran inconveniente es que es difícil simular la entrada y salida de datos del microcontrolador. Tampoco cuentan con los posibles ruidos en las entradas, pero, al menos, permiten el paso físico de la implementación de un modo más seguro y menos costoso, puesto que ahorraremos en grabaciones de chips para la prueba in-situ.

Tarjetas de evaluación. Se trata de pequeños sistemas con un microcontrolador ya montado y que suelen conectarse a un PC desde el que se cargan los programas que se ejecutan en el microcontrolador. Las tarjetas suelen incluir visualizadores LCD, teclados, LEDs, fácil acceso a los pines de E/S, etc. El sistema operativo de la tarjeta recibe el nombre de programa monitor. El programa monitor de algunas placas de evaluación, aparte de permitir cargar programas y datos en la memoria del microcontrolador, puede permitir en cualquier momento realizar ejecución paso a paso, vigilar el estado del microcontrolador o modificar los valores almacenados en los registros o en la memoria.

Emuladores en circuito. Se trata de un instrumento que se coloca entre el PC anfitrión y el zócalo de la tarjeta de circuito impreso donde se alojará el microcontrolador definitivo. El programa es ejecutado desde el PC, pero para la tarjeta de aplicación es como si lo hiciese el mismo microcontrolador que luego irá en el zócalo. Presenta en pantalla toda la información tal y como luego sucederá cuando se coloque la cápsula.

3.9 Selección del microcontrolador

A la hora de escoger el microcontrolador a emplear en un diseño concreto hay que tener en cuenta multitud de factores, como la documentación y herramientas de desarrollo disponibles y su precio, la cantidad de fabricantes que lo producen y por supuesto las características del microcontrolador (tipo de memoria de programa, número de temporizadores, interrupciones, etc.):

Costes. Como es lógico, los fabricantes de microcontroladores compiten duramente para vender sus productos. Y no les va demasiado mal ya que sin hacer demasiado ruido venden 10 veces más microcontroladores que microprocesadores.

Para que nos hagamos una idea, para el fabricante que usa el microcontrolador en su producto una diferencia de precio en el microcontrolador de algunos dólares es importante (el consumidor deberá pagar además el coste del empaquetado, el de los otros componentes, el diseño del hardware y el desarrollo del software). Si el fabricante desea reducir costes debe tener en cuenta las herramientas de apoyo con que va a contar: emuladores, simuladores, ensambladores, compiladores, etc. Es habitual que muchos de ellos siempre se decanten por microcontroladores pertenecientes a una única familia.

Aplicación. Antes de seleccionar un microcontrolador es imprescindible analizar los requisitos de la aplicación:

- *Procesamiento de datos:* puede ser necesario que el microcontrolador realice cálculos críticos en un tiempo limitado. En ese caso debemos asegurarnos de seleccionar un dispositivo suficientemente rápido para ello. Por otro lado, habrá que tener en cuenta la precisión de los datos a manejar: si no es suficiente con un microcontrolador de 8 bits, puede ser necesario acudir a microcontroladores de 16 ó 32 bits, o incluso a hardware de coma flotante. Una alternativa más barata y quizá suficiente es usar librerías para manejar los datos de alta precisión.
- *Entrada Salida:* para determinar las necesidades de Entrada/Salida del sistema es conveniente dibujar un diagrama de bloques del mismo, de tal forma que sea sencillo identificar la cantidad y tipo de señales a controlar. Una vez realizado este análisis puede ser necesario añadir periféricos hardware externos o cambiar a otro microcontrolador más adecuado a ese sistema.
- *Consumo:* algunos productos que incorporan microcontroladores están alimentados con baterías y su funcionamiento puede ser tan vital como activar una alarma antirrobo. Lo más conveniente en un caso como éste puede ser que el microcontrolador esté en estado de bajo consumo pero que despierte ante la activación de una señal (una interrupción) y ejecute el programa adecuado para procesarla.
- *Memoria:* para detectar las necesidades de memoria de nuestra aplicación debemos separarla en memoria volátil (RAM), memoria no volátil (ROM, EPROM, etc.) y memoria no volátil modificable (EEPROM). Este último tipo de memoria puede ser útil para incluir información específica de la aplicación como un número de serie o parámetros de calibración.

El tipo de memoria a emplear vendrá determinado por el volumen de ventas previsto del producto: de menor a mayor volumen será conveniente emplear EPROM, OTP y ROM. En cuanto a la cantidad de memoria necesaria puede ser imprescindible realizar una versión preliminar, aunque sea en pseudo-código, de la aplicación y a partir de ella hacer una estimación de cuánta memoria volátil y no volátil es necesaria y si es conveniente disponer de memoria no volátil modificable.

- *Ancho de palabra:* el criterio de diseño debe ser seleccionar el microcontrolador de menor ancho de palabra que satisfaga los requerimientos de la aplicación. Usar un microcontrolador de 4 bits supondrá una reducción en los costes importante, mientras que uno de 8 bits puede ser el más adecuado si el ancho de los datos es de un byte. Los microcontroladores de 16 y 32 bits, debido a su elevado coste, deben reservarse para aplicaciones que requieran sus altas prestaciones (Entrada/Salida potente o espacio de direccionamiento muy elevado).
- *Diseño de la placa:* la selección de un microcontrolador concreto condicionará el diseño de la placa de circuitos. Debe tenerse en cuenta que quizá usar un microcontrolador barato encarezca el resto de componentes del diseño.

Los microcontroladores más populares se encuentran, sin duda, entre las mejores elecciones:

8048 (Intel). Es el padre de los microcontroladores actuales, el primero de todos. Su precio, disponibilidad y herramientas de desarrollo hacen que todavía sea muy popular.

8051 (Intel y otros). Es sin duda el microcontrolador más popular. Fácil de programar, pero potente. Está bien documentado y posee cientos de variantes e incontables herramientas de desarrollo.

80186, 80188 y 80386 EX (Intel). Versiones en microcontrolador de los populares microprocesadores 8086 y 8088. Su principal ventaja es que permiten aprovechar las herramientas de desarrollo para PC.

68HC11 (Motorola y Toshiba). Es un microcontrolador de 8 bits potente y popular con gran cantidad de variantes.

683xx (Motorola). Surgido a partir de la popular familia 68k, a la que se incorporan algunos periféricos. Son microcontroladores de altísimas prestaciones.

PIC (MicroChip). Familia de microcontroladores que gana popularidad día a día. Fueron los primeros microcontroladores RISC.

Es preciso resaltar en este punto que existen innumerables familias de microcontroladores, cada una de las cuales posee un gran número de variantes.

3.9.1 La familia de los PIC como elección

Para las aplicaciones más habituales (casi un 90%) la elección de una versión adecuada de PIC es la mejor solución; sin embargo, dado su carácter general, otras familias de microcontroladores son más eficaces en aplicaciones específicas, especialmente si en ellas predomina una característica concreta, que puede estar muy desarrollada en otra familia.

El microcontrolador PIC16F84, es un microcontrolador de 18 pines, de 8 bits con tecnología CMOS. La razón porque he elegido este tipo de microcontrolador se debe al tipo de memoria de programa que posee. Se trata de una memoria EEPROM de 1K palabras de 14 bits cada una, tiene la misma capacidad de memoria de instrucciones, pero de tipo flash. Dispone de 64 bytes de EEPROM como memoria de datos auxiliar y opcional.

La memoria EEPROM y la flash son eléctricamente gravables, lo que permite escribir y borrar el programa bajo prueba manteniendo el microcontrolador en el mismo zócalo y usando el mismo dispositivo para grabar y borrar. Esta característica supone una gran ventaja con la mayoría de los microcontroladores, que tienen como memoria de programa reescribible una tipo EPROM. Estas se graban eléctricamente, pero para borrarlas hay que someterlas durante cierto tiempo a rayos ultravioleta, lo que implica sacar del zócalo el circuito integrado y colocarlo en un borrador de EPROM. El hecho de utilizar una memoria flash es porque tiene mayores posibilidades de aumentar su capacidad con relación a la EEPROM. También por su mayor velocidad y menor consumo. La memoria EEPROM es capaz de soportar 1,000,000 de ciclos de escritura / borrado, frente a los 1,000 de la flash.

Los detalles más importantes de la familia PIC y en particular el **PIC16F84** son las siguientes:

Características del PIC 16F84

- Sencillez de manejo: Solo se necesitan 35 instrucciones de una sola palabra para usarlo.
- Todas las instrucciones son de un solo ciclo a excepción de las ramificaciones del programa que son de dos ciclos.
- La velocidad de funcionamiento es de DC a 20MHz en la entrada del reloj y de DC a 200ns en el ciclo de instrucciones.
- La memoria del programa tiene capacidad para 1024 palabras (1kB).
- La memoria RAM de datos es de 68 bytes.
- La memoria de datos EEPROM es de 64 bytes.
- PILA (STACK): De 8 niveles.
- Las palabras de instrucciones tienen una extensión de 14 bits.
- Los bytes de datos tienen una extensión de 8 bits.
- Posee 15 registros de hardware de funciones especiales.
- Procesa modos de acceso directo, indirecto y relativo.
- Tiene cuatro fuentes de interrupciones:
 - Pata externa RB0/INT
 - Exceso del temporizador TMR0
 - Interrupción al cambio PORTB<7:4>
 - Al completar la escritura del EEPROM de datos
- Buena información, fácil de conseguir y económica.
- Precio: Su coste es comparativamente inferior al de sus competidores.
- Poseen una elevada velocidad de funcionamiento. Buen promedio de parámetros: velocidad, consumo, tamaño, alimentación, código compacto, etc.
- Herramientas de desarrollo fáciles y baratas. Muchas herramientas software se pueden recoger libremente a través de Internet desde Microchip (<http://www.microchip.com>).
- Existe una gran variedad de herramientas hardware que permiten grabar, depurar, borrar y comprobar el comportamiento de los PIC.
- Diseño rápido.
- La gran variedad de modelos de PIC permite elegir el que mejor responde a los requerimientos de la aplicación.

Características Periféricas

- 13 pines I/O con control individual de dirección.
- Fuente de alta corriente para excitación directa de LED's.
 - 25 mA de ánodo por pata
 - 25 mA de cátodo por pata
- Temporizador/contador de 8 bits TMR0 con divisor de frecuencia programable de 8 bits.

Características Especiales del Microcontrolador

- Memoria de programa de 1,000 ciclos de escritura/borrado de flash reforzado.
- Memoria de datos EEPROM de 1,000,000 de ciclos de borrado/escritura típica.
- Retención de datos de EEPROM mayor a 40 años.
- Programación en serie dentro del circuito(ICSP = In Circuit Serial Programming) mediante dos pines.
- Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST).
- Watchdog Timer (WDT) con su propio oscilador RC incorporado en el chip para un funcionamiento confiable.
- Protección de códigos.
- Modo SLEEP para economizar consumo.
- Opciones seleccionables de oscilador.
- Tecnología CMOS con Flash reforzado y EEPROM.
- Tecnología de baja potencia y alta velocidad.
- Diseño estático completo.
- Rango amplio de tensiones de funcionamiento:
 - Comercial: 2.0 Volts a 5.5 Volts
 - Industrial: 2.0 Volts a 5.5 Volts
- Consumo muy bajo:
 - <2mA típico a 5 Volts, 4MHz
 - 15uA típico a 2 Volts, 32kHz
 - <0.5uA típico de corriente en reposo a 2 Volts

Una de las razones del éxito de los PIC se basa en su utilización. Cuando se aprende a manejar uno de ellos, conociendo su arquitectura y su repertorio de instrucciones, es muy fácil emplear otro modelo.

3.9.2 Arquitectura del PIC16F84

La arquitectura del procesador sigue el modelo Harvard. En esta arquitectura, la CPU se conecta de forma independiente y con buses distintos con la memoria de instrucciones y con la de datos.

La arquitectura Harvard permite a la CPU acceder simultáneamente a las dos memorias. Además, propicia numerosas ventajas al funcionamiento del sistema como se irán describiendo.

También su arquitectura se basa en un banco de registros, esto significa que todos los objetos del sistema (puertas de E/S, temporizadores, posiciones de memoria, etc.) están implementados físicamente como registros.

En la siguiente figura 3.4. podemos observar por diagramas de bloques la arquitectura del PIC16F84.

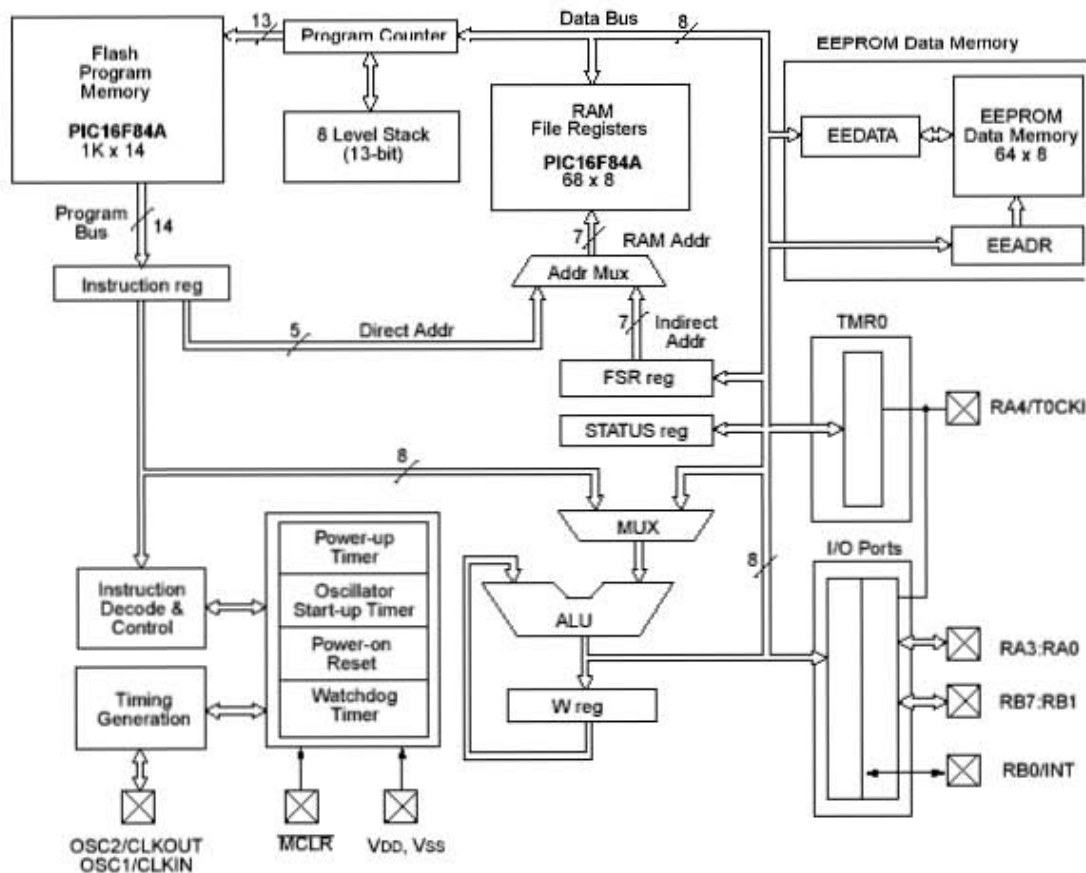


Figura 3.4. Diagrama a bloques del PIC16F84.

3.9.3 Segmentación del PIC16F84

Se aplica la técnica de segmentación (“pipe-line”) en la ejecución de las instrucciones.

La segmentación permite al procesador realizar al mismo tiempo la ejecución de una instrucción y la búsqueda del código de la siguiente. De esta forma se puede ejecutar cada instrucción en un ciclo (un ciclo de instrucción equivale a cuatro ciclos de reloj).

Las instrucciones de salto ocupan dos ciclos al no conocer la dirección de la siguiente instrucción hasta que no se haya completado la de bifurcación.

3.9.4 Formato de las instrucciones del PIC16F84

El formato de todas las instrucciones es de la misma longitud. Todas las instrucciones de los microcontroladores de la gama baja tienen una longitud de 12 bits. Las de la gama media como el PIC16F84 tienen 14 bits y más las de la gama alta. Esta característica es muy ventajosa en la optimización de la memoria de instrucciones y facilita enormemente la construcción de ensambladores y compiladores.

3.9.5 Juego de instrucciones del PIC16F84

Procesador RISC (Computadora de Juego de Instrucciones Reducido).

Los modelos de la gama baja disponen de un repertorio de 33 instrucciones, 35 los de la gama media y casi 60 los de la alta.

Todas las instrucciones son ortogonales. Cualquier instrucción puede manejar cualquier elemento de la arquitectura como fuente o como destino.

3.9.6 Herramientas de soporte potentes y económicas

La empresa Microchip y otras que utilizan los PIC ponen a disposición de los usuarios numerosas herramientas para desarrollar hardware y software. Son muy abundantes los programadores, los simuladores software, los emuladores en tiempo real, Ensambladores, Compiladores C, Intérpretes y Compiladores BASIC, etc.

La arquitectura Harvard y la técnica de segmentación son los principales recursos en los que se apoya el elevado rendimiento que caracteriza estos dispositivos programables, mejorando dos características esenciales:

1. Velocidad de ejecución.
2. Eficiencia en la compactación del código.

CAPÍTULO 4

DESCRIPCIÓN DEL SISTEMA PROPUESTO

4.1 Introducción.

Sabiendo las deficiencias que existen en los sistemas de seguridad de la actualidad, es que surge la idea de desarrollar un sistema que sea adaptable a los existentes y que los complementa solventando de esa manera sus deficiencias.

De esta manera el sistema propuesto cumple con esas exigencias, ya que se puede utilizar tanto para sistemas de seguridad vehicular así como en sistemas de seguridad en casa habitación. Es escalable aunque por el momento solo será desarrollado para cubrir unas pocas necesidades. Puede ser utilizado para capturar estados de avisos de cualquier alarma o sistema, los cuales se harán por medio de los sensores.

4.2 Diagrama a bloques del sistema de adquisición de estados de alarmas.

Diagrama a bloques del sistema de adquisición de estados de alarmas con tecnología GSM sobre la base de un microcontrolador, lo podemos observar en la figura 4.1.

4.2.1 Alimentación.

Tanto el microcontrolador como el dispositivo GSM utilizan niveles de tensión TTL es decir del rango entre 0 a 5 [volts]. Por lo que para poder funcionar adecuadamente se requiere de una fuente de alimentación que proporcione este voltaje, si utilizamos el sistema en un vehículo bastará con un divisor de voltaje o convertidor de CD a CD que disminuya el voltaje ya que las baterías de los vehículos oscilan entre los 12 y 15 [volts] por otro lado si se utiliza en una casa habitación no sería tampoco necesario construir una fuente de este voltaje, ya que con un eliminador de corriente que proporcione este voltaje bastará, de ahí la facilidad de uso del sistema.

Cabe mencionar que si se utiliza el dispositivo GSM este cuenta con fuente independiente, figura 4.2, pero al conectarlo a la alimentación se utiliza para cargar su pila interna.

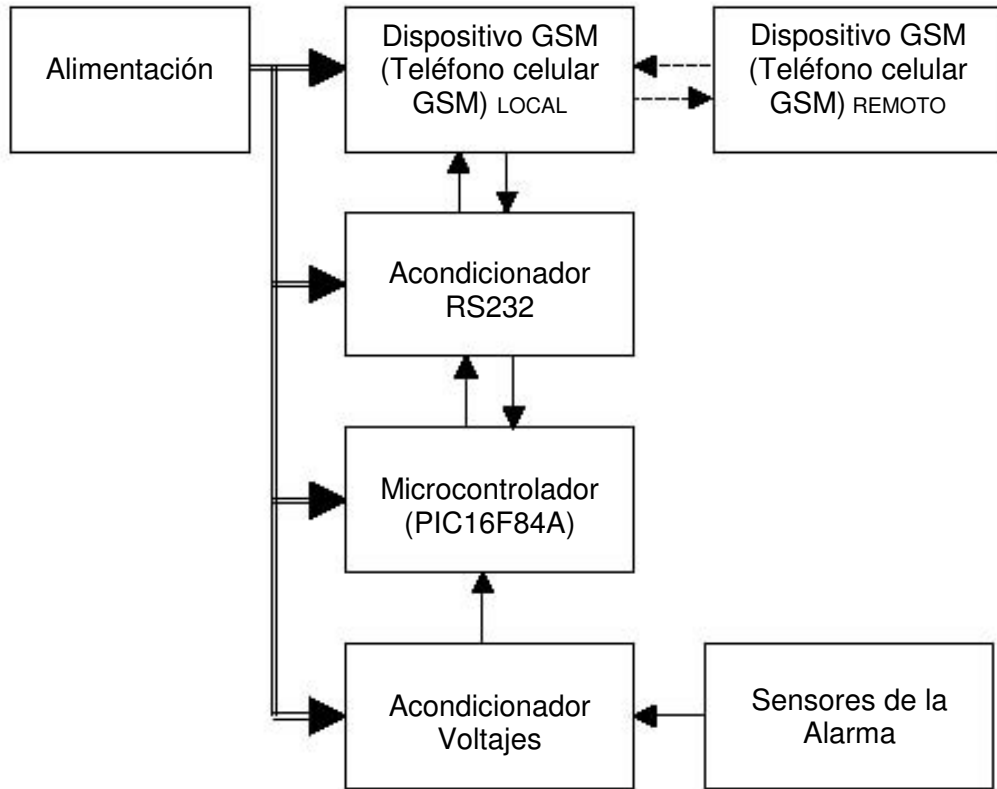


Figura 4.1. Diagrama a bloques del sistema de adquisición de estados de alarmas.



Figura 4.2. Pila interna del Teléfono GSM Ericsson.

4.2.2 Dispositivo GSM (Teléfono celular GSM).



Figura 4.3. Teléfono GSM Ericsson modelo A2218z.

Usamos el dispositivo GSM para hacer las pruebas de comunicación serial y saber que comandos AT son aceptados y como funcionan. Para esto como se explicó en la selección del dispositivo GSM se optó por el teléfono GSM Ericsson por lo que se construyó el cable para la comunicación en configuración para MODEM nulo, ver figura 4.4.

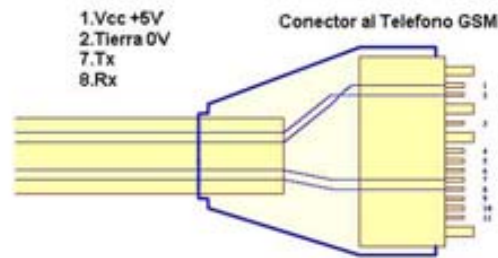


Figura 4.4. Conector del Teléfono GSM Ericsson.

Teniendo así tres terminales importantes para la comunicación serial asíncrona que son Tierra, Tx (Transmisión de datos) y Rx (Recepción de datos) los cuales tienen un nivel TTL de 5[volts].

4.2.3 Acondicionador del protocolo RS232.

Como vimos anteriormente llegamos a tener tres terminales Tierra, Tx y Rx pero a nivel TTL así que si se quiere conectar esto a una computadora que utiliza una norma conocida como RS232 con voltajes de +12V a -12v necesitamos utilizar un acondicionador para poder manejar esta diferencia de voltajes.

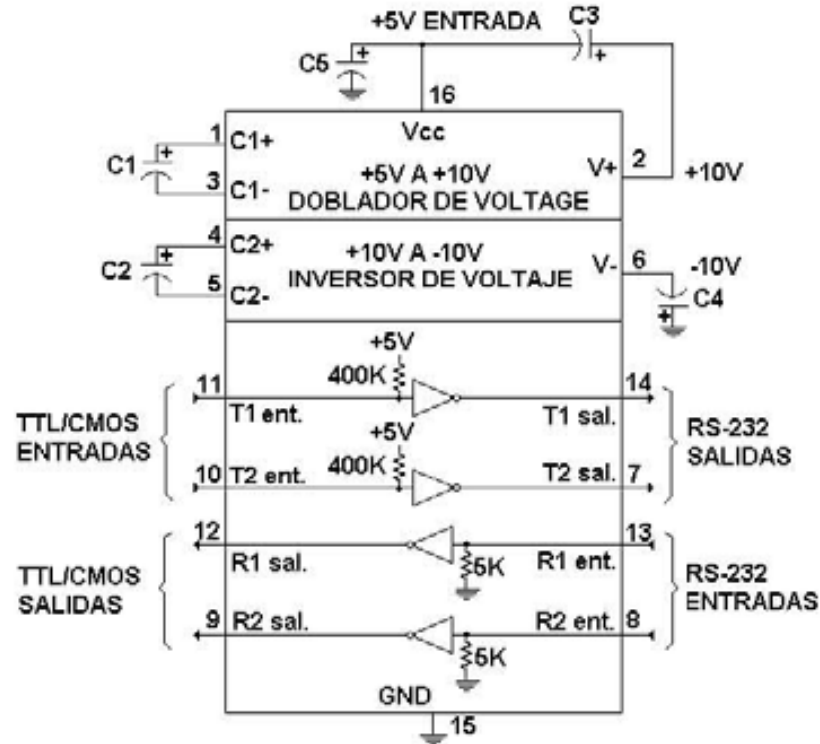


Figura 4.5. Diagrama del MAX 232.

Afortunadamente existen en el mercado circuitos integrados que ya hacen este trabajo como el caso del MAX232 de la empresa MAXIM, menciono este ya que lo tengo y solo se necesita agregarle unos cuantos componentes para que funcionen, siendo estos capacitores como se observa en la figura 4.5.

El circuito integrado MAX232 cambia los niveles TTL a los del estándar RS-232 cuando se hace una transmisión, y cambia los niveles RS-232 a TTL cuando se tiene una recepción.

4.2.4 Microcontrolador (PIC16F84A).

Al utilizar este microcontrolador tenemos la ventaja que podemos configurar un puerto como Tx y otro como Rx escogiéndolos dentro de sus 13 puertos, pero también tendríamos niveles TTL de 5[volts]. Se pueden llegar a conectar directamente al teléfono GSM los puertos de transmisión utilizando la configuración de MODEM nulo, pero como se desea que posteriormente el sistema sea adaptable a cualquier teléfono GSM es por eso que se utiliza el acondicionador, porque los cables que venden en el mercado traen ya un acondicionador por lo que para el correcto acoplamiento se necesita este.

De los 11 puertos restantes podemos utilizar los que necesitemos según sean las necesidades para las entradas o salidas, como entradas tendríamos la información de los sensores y como salidas podemos controlar relevadores para cortes de corriente o manejar diferentes voltajes como lo vemos en el siguiente punto.

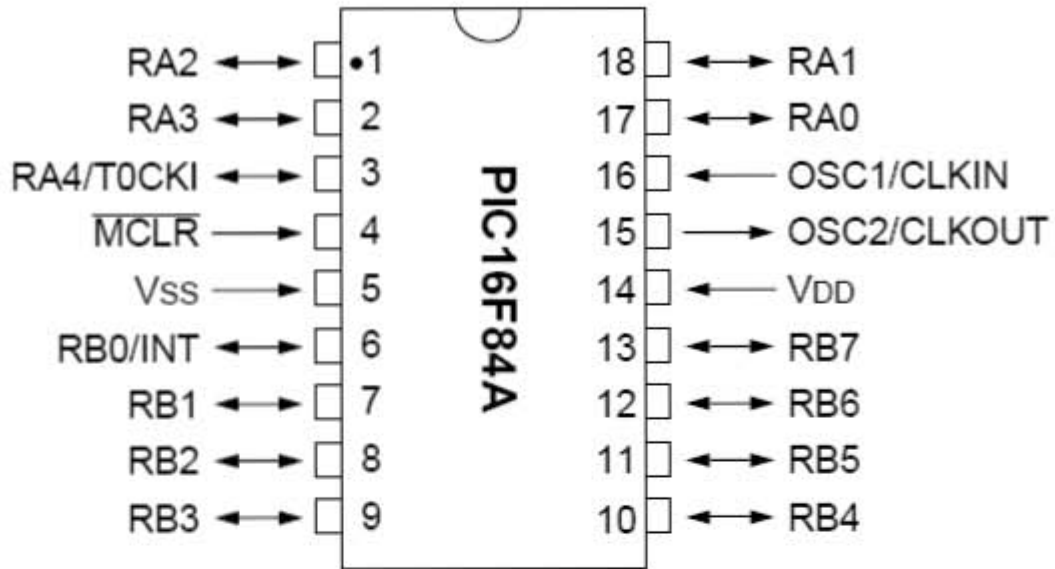


Figura 4.6. Microcontrolador PIC16F84A.

4.2.5 Acondicionador de Voltajes.

Este acondicionador se utiliza debido a las diferencias de voltajes que existen entre la alimentación de la alarma, los sensores de esta y el sistema que como ya lo comente anteriormente utiliza voltajes TTL, por lo que la función de este acondicionador será la de transformar los voltajes a TTL provenientes de la alarma hacia el sistema de adquisición de estados.

Para este cometido se tienen muchas opciones como es el caso de los optoaclopadores, transistores, simples arreglos de resistencias que sirvan como divisores de voltaje, pero ya en el último capítulo se vera cual es la mejor opción para utilizar como acondicionador de voltajes.

4.2.6 Sensores de alarma.

Para poder tener avisos de alarma nos tenemos que apoyar en los sensores de la alarma, que son parte fundamental para el buen funcionamiento del sistema ya que son los encargados de indicarnos el estado de la alarma, estos enviarán su señal a través del sistema acondicionador de voltaje hacia los puertos del microcontrolador, y dependiendo de la programación, actúa de una manera u otra según sea el caso.

4.3 Diagrama a bloques del sistema de pruebas con la PC.

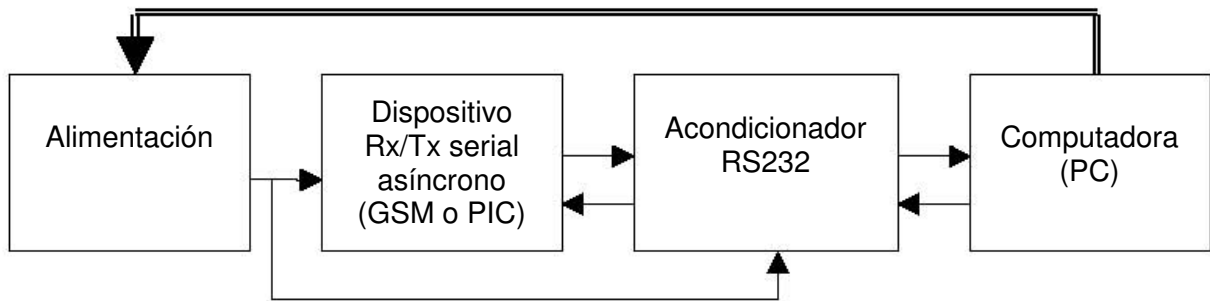


Figura 4.7. Diagrama a bloques del sistema de pruebas.

En la figura 4.7 se observa el diagrama a bloques del sistema utilizado para hacer pruebas de la comunicación serial asíncrona, que será explicado a continuación, con este se puede probar la comunicación serial entre el dispositivo GSM (teléfono celular) y una computadora personal (PC), así como la comunicación de un microcontrolador (PIC16F84A) con esta misma.

Es muy importante esta parte del proyecto ya que de aquí se desprenderán las formas en las que se comunica el dispositivo GSM para después implementar esto con el microcontrolador PIC.

4.3.1 Alimentación.

Tanto el microcontrolador como el dispositivo GSM utilizan niveles de tensión TTL es decir del rango entre 0 a 5 [volts]. Para este sistema de prueba la fuente de alimentación se obtiene de un puerto USB (Universal Serial BUS) ya que estos trabajan a 5[volts] de esta forma tenemos el voltaje sin necesidad de construir una fuente de este voltaje.

Los extremos de un conector USB son los encargados de suministrar este voltaje como lo vemos en la figura 4.8, el voltaje alimenta tanto al microcontrolador o dispositivo GSM según sea lo que se pruebe y al acondicionador RS232.

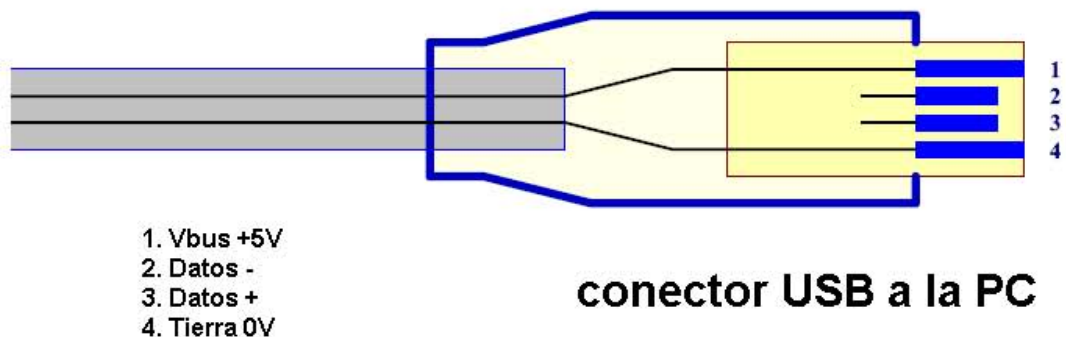


Figura 4.8. Descripción de las terminales del conector USB.

4.3.2 Dispositivo para el envío y recepción de la comunicación serial asíncrona.

Para esta parte como ya lo mencione se puede utilizar un dispositivo GSM como es el caso del teléfono GSM figura 4.3 o un microcontrolador PIC16F84 figura 4.6

Primero usamos el dispositivo GSM para hacer las pruebas de comunicación serial y saber que comandos AT son aceptados y como funcionan. Conociendo los comandos AT ahora solo resta programarlos en el microcontrolador y ver su comunicación serial para posteriormente acoplarlos como se muestra en la figura 4.1.

4.3.3 Computadora personal (PC).

Actualmente todas las computadoras ya sean de escritorio o portátiles cuentan con puertos seriales conocidos como COM 1 y COM 2 los cuales se rigen por la norma RS232, así como conectores USB figura 4.8. La ventaja de usar una computadora para hacer las pruebas es que se puede desarrollar una aplicación para hacer éstas con los puertos o se puede usar una aplicación conocida como hyperterminal para las que cuentan con sistema operativo Windows.

Con esta aplicación cambiamos los parámetros de conexión desde el número de puerto, velocidad de transmisión/recepción , tipo de paridad y número de bits.

De la figura 4.9 observamos los dos puertos que usaremos uno para obtener la fuente de alimentación y otro para conectar el dispositivo a probar, también vemos un comando de los tantos que probaremos para ver la comunicación serial.

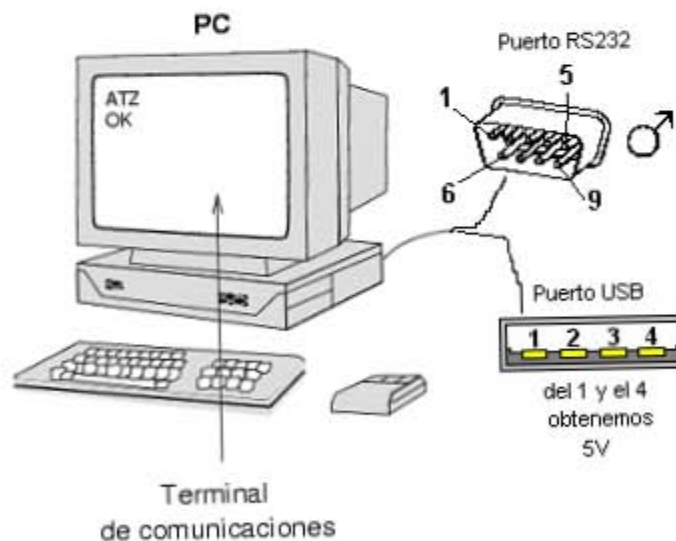


Figura 4.9. Computadora con conectores seriales RS232 y USB.

CAPÍTULO 5

SOFTWARE DEL SISTEMA DE ADQUISICIÓN DE DATOS

5.1 Introducción.

En este capítulo se explica el desarrollo del software que se utilizó, tanto para la depuración y pruebas del sistema en la computadora, como en implementación del mismo en el microcontrolador, es decir su programación en lenguaje ensamblador.

Así mismo se habla de los programas que se utilizaron para la elaboración del sistema.

5.2 Desarrollo de programa en Visual Basic para depuración

Para poder analizar la comunicación serial tanto del microcontrolador PIC como del dispositivo GSM (teléfono celular) fue necesario elaborar un programa que nos proporcionara esto, como el lenguaje de programación que mejor conocemos es el Visual Basic, fue el motivo por el que se hizo así, en la figura 5.1 se muestra la pantalla principal del programa que se conoce como forma dentro del lenguaje de programación Visual Basic. Este programa consta de tres módulos y una forma, el código fuente de este programa se encuentra en el apéndice B.

Este programa es muy básico pero de suma importancia, en primer lugar logra la comunicación por medio del puerto serial de la computadora, esto se lleva a cabo mediante un control de visual basic que se llama MSComm, una vez lograda la comunicación con el dispositivo GSM se procede a probar los diferentes comando AT que comunican al modem GSM con la computadora, algunos comandos no son válidos para todos, incluso algunos son exclusivos de algunas empresas fabricantes de estos.

Como vimos en el capítulo 2 existe todo un procedimiento para el envío de los mensajes SMS así como para el recibo de los mismos, esto también lo programamos para ver como se lleva a cabo la codificación y decodificación de las tramas PDU, esto es la parte principal de este programa ya que una vez comprendido esto solo resta programarlo en el microcontrolador.

En la figura 5.2 observamos ya el programa funcionando con un ejemplo de la conversión de texto a la trama PDU, el cual se utiliza en la programación del microcontrolador. En la caja de texto que dice texto a enviar escribimos el mensaje de texto, para este ejemplo escribimos "CAMBIO OK", este lo usamos para que el sistema nos indique que algún cambio en la configuración fue hecho de manera correcta.

En la caja de texto que dice trama PDU obtenemos el resultado de la conversión esto lo analizamos y tomamos solo lo que necesitamos, esto lo explicaremos mas a detalle en el siguiente punto cuando analicemos el programa para el microcontrolador.

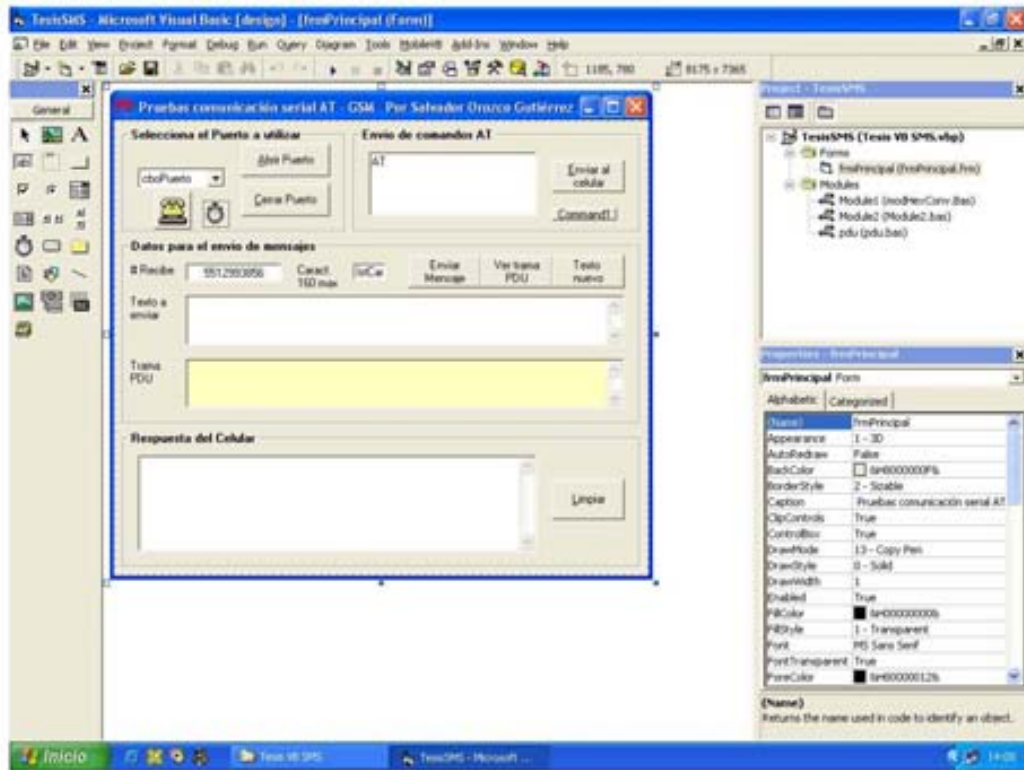


Figura 5.1. Pantalla del programa para pruebas desarrollado en Visual Basic.

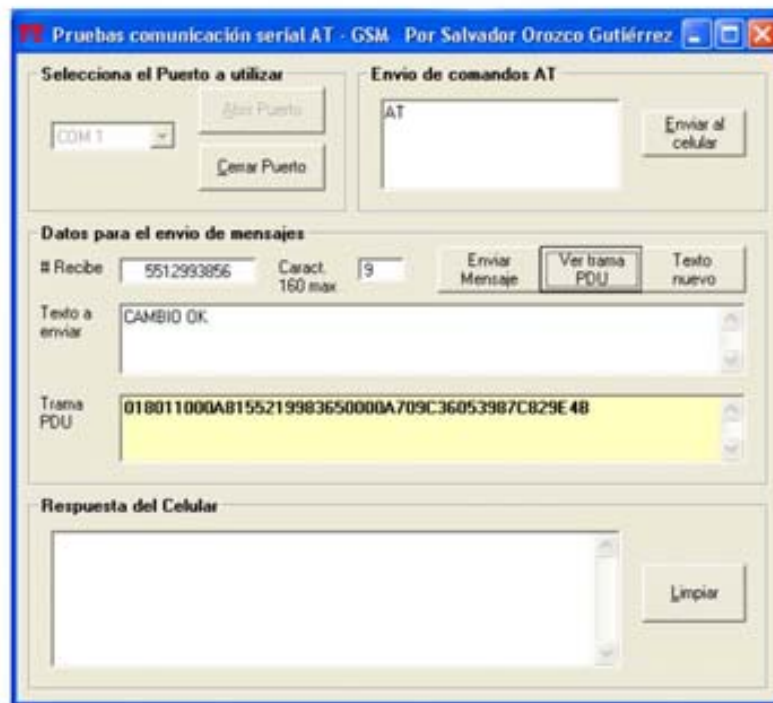


Figura 5.2. Ejemplo de Trama PDU utilizado en el sistema.

5.3 Desarrollo del programa para el microcontrolador

Para poder entrar de lleno en la programación del microcontrolador necesitamos hacer un diagrama de flujo el cual nos ejemplifique el funcionamiento del sistema, una vez teniendo este como lo vemos en la figura 5.3 ahora solo resta ver que herramienta usaremos para la programación de este.

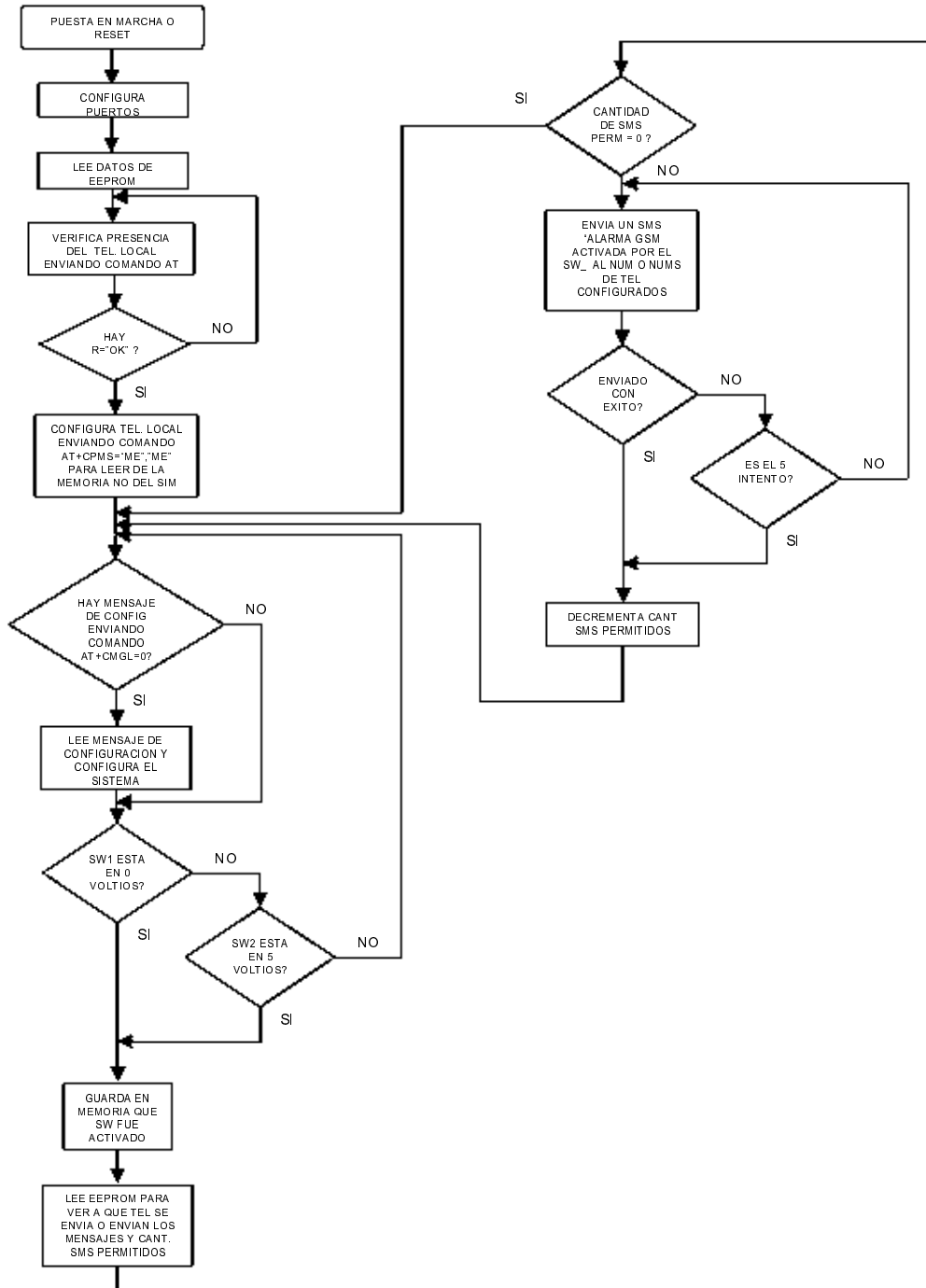


Figura 5.3. Diagrama de flujo del sistema de Adquisición de Estados.

5.3.1 Descripción del programa elaborado en PICBasic PRO.

Como lo explicamos en el punto 3.8 del capítulo 3 existen muchas herramientas para el desarrollo del software para el microcontrolador, debido a la complejidad del sistema elaborarlo en lenguaje ensamblador requeriría de mucho tiempo, por lo que se optó por utilizar un compilador y como el lenguaje que sabemos es el Basic se utilizó el programa llamado “PICBasic PRO”.

El programa completo del sistema se encuentra en el apéndice C a continuación haremos una breve explicación de lo que hacen unas partes del código.

Lo que se encuentra antecedido de un apóstrofe son comentarios por lo que el compilador no hace caso de esto al compilar el programa.

```

'* Tesis      : SISTEMA DE ADQUISICIÓN DE ESTADOS DE ALARMAS CON TECNOLOGIA GSM *
'*           : SOBRE LA BASE DE UN MICROCONTROLADOR. *
'* Autor      : Salvador Orozco Gutiérrez *
'* Fecha      : 22/03/06 *
'* Notas      : El microcontrolador es un PIC16F84A *

```

Iniciamos con la configuración del sistema lo que implica declaración de puertos y de constantes, definimos que puertos utilizaremos y el tipo es decir, si son entradas o salidas así como su valor inicial. Declaramos las variables y definimos los valores iniciales de la memoria EEPROM.

```

' DECLARACIÓN DE PUERTOS Y CONSTANTES.
' IO de la comunicación serial
T9600 CON 84           ' Velocidad transmisión 9600bps
CR   CON 13           ' (CR) Carriage Return - Retorno de carro
LF   CON 10           ' (LF) Line Feed - Alimentación de línea
SYMBOL TX = PORTB.3   ' Puerto PIC Tx PB3
SYMBOL RX = PORTB.0   ' Puerto PIC Rx PB0
' Sensores
SYMBOL HABILITA = PORTB.5 ' Habilita el sistema de adquisición
SYMBOL SENSORBAJO = PORTB.6 ' Sensor activa a 0[V]
SYMBOL SENSORALTO = PORTB.7 ' Sensor activa a 12[V]
' LED para visualizar estados
SYMBOL ESTADO = PORTA.3 ' LED de Estados

' DEFINE ENTRADAS, SALIDAS Y ESTADO INICIAL DE LOS PUERTOS.
INPUT RX           ' Entrada de datos Rx
INPUT HABILITA     ' Entrada para habilitar el sistema
INPUT SENSORBAJO  ' Entrada sensor [0]V
INPUT SENSORALTO  ' Entrada sensor [12]V
OUTPUT TX          ' Salida de datos Tx
OUTPUT ESTADO     ' Salida LED de estados
PORTA =%00000000   ' Estado inicial del Puerto A
PORTB =%00000000   ' Estado inicial del Puerto B

' DECLARACIÓN DE VARIABLES.
TMP1 VAR BYTE      ' Registros temporales
TMP2 VAR BYTE
TMP3 VAR BYTE
TMP4 VAR BYTE
TIEMPO VAR BYTE    ' Para tiempos
NUMCAR VAR BYTE    ' Longitud PDU a enviar
SENSOR VAR BYTE     ' Que sensor se disparó
TEL VAR BYTE[9]     ' Arreglo para núm. teléfono que envía SMS
pduSMS VAR BYTE[10] ' Arreglo para el SMS en formato PDU

```

```
' DATOS DE EEPROM INICIALES.
EEPROM 42, [0]           ' dispara alarma, sensores bajo= 3, alto= 5
EEPROM 43, [5]           ' # de SMS permitidos = 5
EEPROM 44, ["31D98CA602"] ' Contraseña = 1234*
EEPROM 54, ["5591018432"] ' Teléfono que recibe SMS = 5519104823
```

Verificamos presencia del teléfono local enviando el comando **AT** que significa atención y esperamos su respuesta, de no recibirla al cabo de unos segundos entonces nos vamos a la etiqueta VERIFICA para seguir intentándolo.

Los teléfonos GSM pueden utilizar dos tipos de memoria, la interna del teléfono y la que traen en el SIM. Para los propósitos de nuestro sistema, utilizaremos la memoria interna, por lo que si se recibe la respuesta del teléfono ahora configuramos el teléfono con el comando **AT+CPMS** que viene de **Preferred SMS Message Storage** preferencia de almacenaje de los mensajes SMS.

```
'*****
VERIFICA:
'-----
' Verifica presencia del Teléfono Local enviando comando AT
' Esperando respuesta OK de este.
'-----
ESTADO = 1           ' Prende LED de estado
SEROUT TX,T9600, ["AT",CR] ' Verifica presencia del teléfono
SERIN RX,T9600,2000,VERIFICA,[WAIT("OK")]
PAUSE 500           ' Pausa de .5 seg
'-----
' Configura memoria del TEL enviando AT+CPMS="ME", "ME".
' Esperando respuesta OK de este.
'-----
ESTADO = 0           ' Apaga LED de estado
SEROUT TX,T9600, ["AT+CPMS=", 34, "ME", 34, 44, 34, "ME", 34, CR]
SERIN RX,T9600,2000,VERIFICA,[WAIT("OK")]
PAUSE 500           ' Pausa de .5 seg
```

En la rutina DESACTIVA se queda el sistema si no es habilitado por el puerto HABILITA, en la parte de hardware del siguiente capítulo veremos como hacer esto.

```
'*****
DESACTIVA:
'-----
' MIENTRAS NO SE HABILITE EL SISTEMA PERMANECE EN ESPERA.
'-----
WHILE HABILITA = 1
ESTADO = 0           ' Apaga LED de estado
SLEEP 60             ' Duerme el micro por 1 min. aprox.
TIEMPO = 0           ' Inicializa el tiempo
WEND
```

INICIO es la rutina principal del sistema, primero vemos si esta habilitado el sistema, luego esperamos aproximadamente 3 minutos para buscar si llegó mensaje de configuración desde el teléfono remoto, por último verificamos si han sido disparados los sensores, para mandar mensaje SMS y guardar que sensor fue.

```
'*****
INICIO:
IF HABILITA = 1 THEN DESACTIVA ' Se desactiva sistema a 5[V]
IF TIEMPO=180 THEN GOSUB smsCONFIG ' Verifica cada 3 min. si hay smsCONFIG
```



```

ESTADO = 1                                ' Prende LED de estado
PAUSE 500                                  ' Pausa de .5 seg

ESTADO = 0                                ' Apaga LED de estado
PAUSE 500                                  ' Pausa de .5 seg

SENSOR=0
IF SENSORBAJO=0 THEN SENSOR="6" ' Se disparó el sensor a 0[V]
IF SENSORALTO=1 THEN SENSOR="A" ' Se disparó el sensor a 12[V]
IF SENSOR<>0 THEN
  NUMCAR = "2"                             ' Para enviar el SMS indica 22
  WRITE 42,SENSOR                          ' Almacena que sensor disparó la alarma
  READ 43,TMP1                             ' SMS permitidos
  IF TMP1=0 THEN INICIO                   ' TMP1=0 implica ya no enviar SMS
  GOSUB enviaSMS                          ' Va a enviar el mensaje SMS
ENDIF

TIEMPO = TIEMPO + 1                       ' Incrementa la cuenta del tiempo

GOTO INICIO                               ' Regresa y parpadea el LED para siempre

```

La rutina smsCONFIG es la encargada de ver si se ha sido recibido un mensaje de configuración, esto lo hace preguntando al teléfono si hay un mensaje no leído mediante el comando **AT+CMGL=0 List SMS Messages** es decir lista los mensajes, si no hay regresa al programa principal, si existe entonces almacena tanto el número de teléfono que envió el mensaje como el mensaje para una posterior comparación.

Después borra el mensaje que llego para no saturar la memoria mediante el comando **AT+CMGD=1 Delete SMS Message** es decir borra el mensaje, ya que esta reducida según la marca de teléfono, de lo contrario se saturaría y no podríamos seguir recibiendo mensajes hasta borrarlos.

Como lo hemos venido planteando el sistema es escalable y podemos usar muchas formas de configuración, para este caso usamos la modificación del número de teléfono que recibirá los mensajes SMS por eso almacenamos el número que envía el mensaje y verificamos que la contraseña sea la adecuada, una vez pasado esta verificación podemos cambiar el numero en la memoria EEPROM del microcontrolador.

Para saber si fue correcto el procedimiento nos vamos a la rutina de envío de mensajes SMS para que nos mande un mensaje diciendo "CAMBIO OK", y posteriormente regresamos a la rutina principal INICIO.

```

'*****
smsCONFIG:
'-----
' Pregunta si hay mensaje nuevo del TEL enviando AT+CMGL=0
'-----

ESTADO = 1                                ' Prende LED de estado
SEROUT TX,T9600,["AT+CMGL=0",CR]
SERIN RX,T9600,500,noCMGL,[WAIT("+CMGL:")]
SERIN RX,T9600,[WAIT("0891"),SKIP 20, STR TEL\10, SKIP 20, STR pduSMS\10]

' Compara teléfono y SMS
decSMS:
TMP3=0
TMP4=0
FOR TMP1 = 54 TO 63
  READ TMP1,TMP2                          ' Lee el núm. tel. de EEPROM
  IF TMP2<>TEL[TMP1-54] THEN              ' Para saber si es diferente
    TMP3=1

```

```

        ENDIF
        READ TMP1-10,TMP2          ' Lee el SMS en formato PDU de EEPROM
        IF TMP2<>pduSMS[TMP1-54] THEN
            TMP4=1                ' Para saber si es diferente
        ENDIF
    NEXT TMP1

    ' Borra el mensaje recibido, para que no se llene la memoria del teléfono
    SEROUT TX,T9600,["AT+CMGD=1",CR]

    ' Si se cumplen condiciones cambia el número del teléfono que recibe SMS
    IF TMP3=1 AND TMP4=0 THEN cambiaTEL

    ' No hubo mensaje SMS de configuración
noCMGL:
    ESTADO = 0                    ' Apaga LED de estado
RETURN

    ' Cambia el número que recibe el SMS
cambiaTEL:
    ESTADO = 0                    ' Apaga LED de estado

    FOR TMP1 = 0 TO 9             ' Ciclo para sobre escribir el núm. tel.
        WRITE TMP1+54,TEL[TMP1]
    NEXT TMP1

    NUMCAR="1"                    ' Para enviar el SMS indica 21

    GOTO enviaSMS

RETURN

```

La rutina enviaSMS, es la encargada de enviar los mensajes al teléfono remoto y la última del sistema, para enviar los mensajes SMS utilizamos el comando **AT+CMGS Send SMS Messages** seguido de la longitud del mensaje PDU, esperamos respuesta del teléfono y enviamos la trama PDU terminando con CTRL-Z. Terminado esto el programa se detiene aproximadamente por 2 minutos para regresar al programa INICIO.

```

*****
enviaSMS:
'-----
' Envía mensaje enviando AT+CMGS=NUMCAR
' donde NUMCAR es 21 o 22 depende de la longitud del mensaje SMS en PDU
'-----
    ESTADO = 1                    ' Prende LED de estado
    SEROUT TX,T9600,["AT+CMGS=2",NUMCAR,CR]
    SERIN RX,T9600,[WAIT("> ")]
    PAUSE 5000
    SEROUT TX,T9600,["018011000A81"]

    ' Ciclo para leer el número de teléfono al que se envía el mensaje SMS
    FOR TMP2=54 TO 63
        READ TMP2,TMP3
        SEROUT TX,T9600,[TMP3]
    NEXT TMP2
    SEROUT2 TX,T9600,["0000A70"]

    IF NUMCAR ="2" THEN
        ' Mensaje "SW1 ACTIVO" depende del EEPROM 48
        SEROUT TX,T9600,["0AD36B0C141C5293D627",26,CR,LF]
        ' Mensaje "SW2 ACTIVO" depende del EEPROM 48
        SEROUT TX,T9600,["0AD3AB0C141C5293D627",26,CR,LF]
        SEROUT TX,T9600,["AD3",SENSOR,"B0C141C5293D627",26,CR,LF]
    ELSE
        ' Mensaje "CAMBIO OK"
        SEROUT TX,T9600,["09C36053987C829E4B",26,CR,LF]
        SEROUT TX,T9600,["9C36053987C829E4B",26,CR,LF]
    ENDIF

```

```

ENDIF

SERIN RX,T9600,5000,enviaSMS,[WAIT("OK")]

PAUSE 500           ' Pausa de .5 seg
ESTADO = 0           ' Apaga LED de estado

WRITE 43,TMP1-1    ' Decrementa cant. de SMS permitidos

SLEEP 120         ' Duerme el micro por 2 min. aprox.

RETURN

END
    
```

5.3.2 Diagramas de flujo de la comunicación serial asíncrona.

Como parte principal del programa tenemos la comunicación serial asíncrona por lo que será lo único que veremos como funciona en ensamblador a través del comando **SERIN** para la recepción de datos y **SEROUT** para la transmisión de datos, de la figura 5.4 observamos el diagrama de flujo de ambas para posteriormente analizarlo.

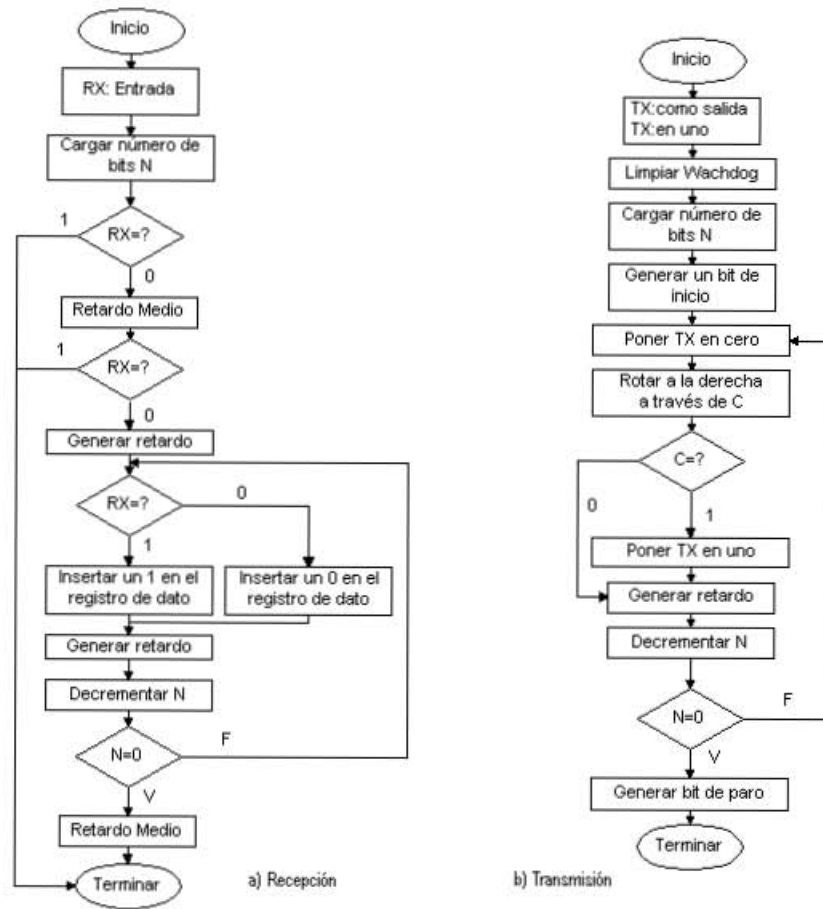


Figura 5.4. Diagrama de flujo de la comunicación serial asíncrona.

Para el comando SEROUT tenemos lo siguiente:

```

SEROUT
MOVWF 0x28 ;envía un byte vía la línea serial
Move 8, 0x26 ;envía 8 bits de datos
BCF IOPort, IOP_TXD ;envía 1 bit de inicio
CALL Wait1Bit ;espera un tiempo por el bit de inicio

SendByte_Loop
RRF 0x28, 1 ;mueve el bit actual en el carry
BTFSC STATUS, C
BSF IOPort, IOP_TXD ;bajo si el bit es igual a 1
BTFSS STATUS, C
BCF IOPort, IOP_TXD ;alto si el bit es igual a 0
CALL Wait1Bit
DECFSZ 0x26, 1 ;se han enviado 8 bits de datos?
GOTO SendByte_Loop ;< 8 bits enviados, envía el próximo bit

BSF IOPort, IOP_TXD ;envia 1 bit de paro
CALL Wait1Bit

RETURN
    
```

Para el comando SERIN tenemos lo siguiente:

```

SERIN
CLRF 0x28 ;limpia el almacén para recibir el byte
Move 8, 0x26 ;recibe 8 bits de datos

RecByte_StartBit
BTFSC IOPort, IOP_RXD
GOTO RecByte_StartBit ;espera el bit de inicio para empezar a recibir
CALL Wait12Bit ;espera medio bit
BTFSC IOPort, IOP_RXD
GOTO RecByte_StartBit ;espera por el bit de inicio una segunda vez

RecByte_Loop
CALL Wait1Bit
BTFSS IOPort, IOP_RXD
BCF STATUS, C ;alto -> 0
BTFSC IOPort, IOP_RXD
BSF STATUS, C ;bajo -> 1

RRF 0x28, f ;rota el bit actual en el carry
DECFSZ 0x26, f ;se han recibido 8 bits de datos?
GOTO RecByte_Loop ;< 8 bits recibidos, recibe el próximo bit

CALL Wait1Bit
MOVF 0x26, W
RETURN
    
```

Tanto para SERIN como para SEROUT utilizamos los siguientes retardos en función de la velocidad de trasmisión que usamos y una macro:

```
;----- carga un registro con el valor de un byte
Move macro value, target
    if value == 0
        CLRF    target
    else
        MOVLW   value
        MOVWF   target
    endif
endmacro

;Espera el tiempo de trasmisión de 1 bit a 9600 bit/s y 4 MHz
Wait1Bit
    MOVLW    D'15'
    MOVWF    0x27
Wait1Bit_Loop
    NOP
    NOP
    NOP
    DECFSZ   0x27, 1
    GOTO     Wait1Bit_Loop
    RETURN

;Espera el tiempo de trasmisión de 1/2 bit a 9600 bit/s y 4 MHz
Wait12Bit
    MOVLW    D'15'
    MOVWF    0x27
Wait12Bit_Loop
    DECFSZ   0x27, f
    GOTO     Wait12Bit_Loop
    RETURN
```

CAPÍTULO 6

HARDWARE DEL SISTEMA DE ADQUISICIÓN DE DATOS

6.1 Introducción.

En este capítulo se describe el hardware del sistema de adquisición de estados, en bloques como ya se mencionó anteriormente. El hardware aquí mencionado es solo el principio básico de lo que puede ser el sistema escalado, ya que es adaptable a cualquier sistema de alarmas además de que puede crecer más agregándole cosas como posicionamiento global satelital (GPS), todo esto, aprovechando los puertos que contiene el microcontrolador, será cuestión de la programación del mismo.

6.2 Alimentación.

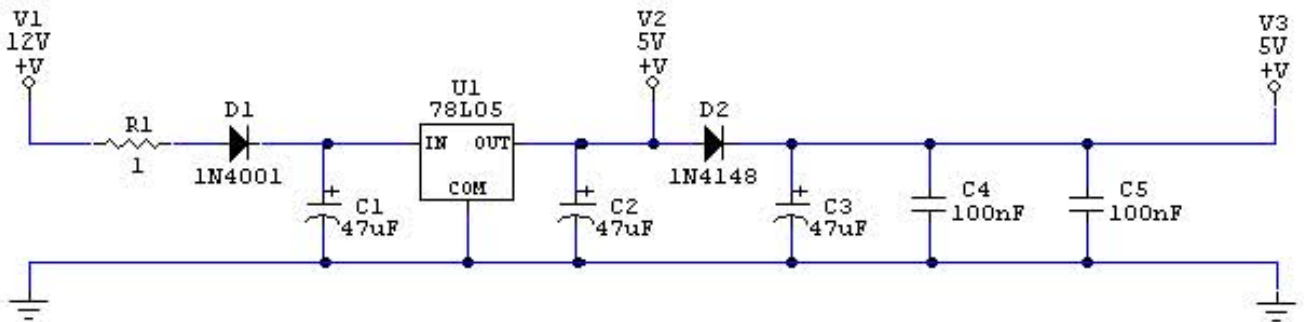


Figura 6.1. Diagrama electrónico de la Alimentación.

De la figura 6.1 vemos tres voltajes V1, V2 y V3; V1 es el voltaje de entrada que si se utiliza el sistema en un vehículo sería de 12[V], si se utiliza el sistema en una casa habitación con un eliminador de corriente tendríamos que generar ese mismo voltaje, V2 es el voltaje de alimentación para el dispositivo GSM que utilizan normalmente 5[V], este voltaje V2 lo obtenemos gracias al U1 78L05, el cual, es un circuito integrado convertidor de CD a CD conocido más como regulador de voltaje, cabe mencionar que si por alguna razón perdiéramos el voltaje V1, ahora el voltaje V2 lo suministraría la pila del dispositivo GSM (teléfono celular GSM) y por último se obtiene el voltaje V3 el cual es el encargado de suministrar el voltaje de 5[V] para alimentar el microcontrolador PIC.

Entre el cable de alimentación de entrada se sitúan dos dispositivos de seguridad una resistencia R1 de 1[Ω] a 1[W] y un diodo D1 1N4001 para la polaridad correcta. El diodo D2 se encargará de aislar la alimentación V3 del PIC de la del dispositivo GSM por último los capacitores C3, C4 y C5 son los encargados de filtrar la alimentación.

6.3 Microcontrolador.

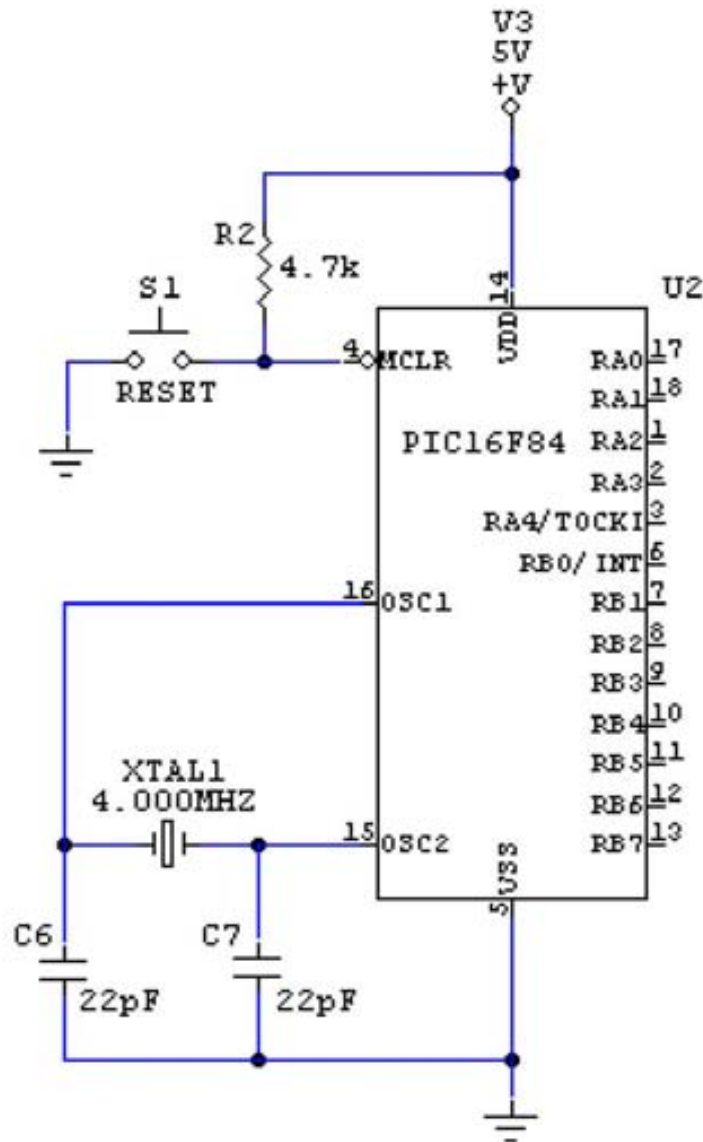


Figura 6.2. Diagrama electrónico del Microcontrolador.

De la figura 6.2 se ve la disposición del PIC, el cristal XTAL1 de 4[MHz] y los dos capacitores C6 y C7 de 22 [pF] que juntos conforman el oscilador de reloj.

La resistencia R2 de 4.7[kΩ] a 0.5[W] se conecta en la línea de reset y a la línea de alimentación para que funcione como de línea alta y no este flotando el pin. El interruptor S1 es utilizado para realizar un reset maestro (MCLR).

6.4 Sensores.

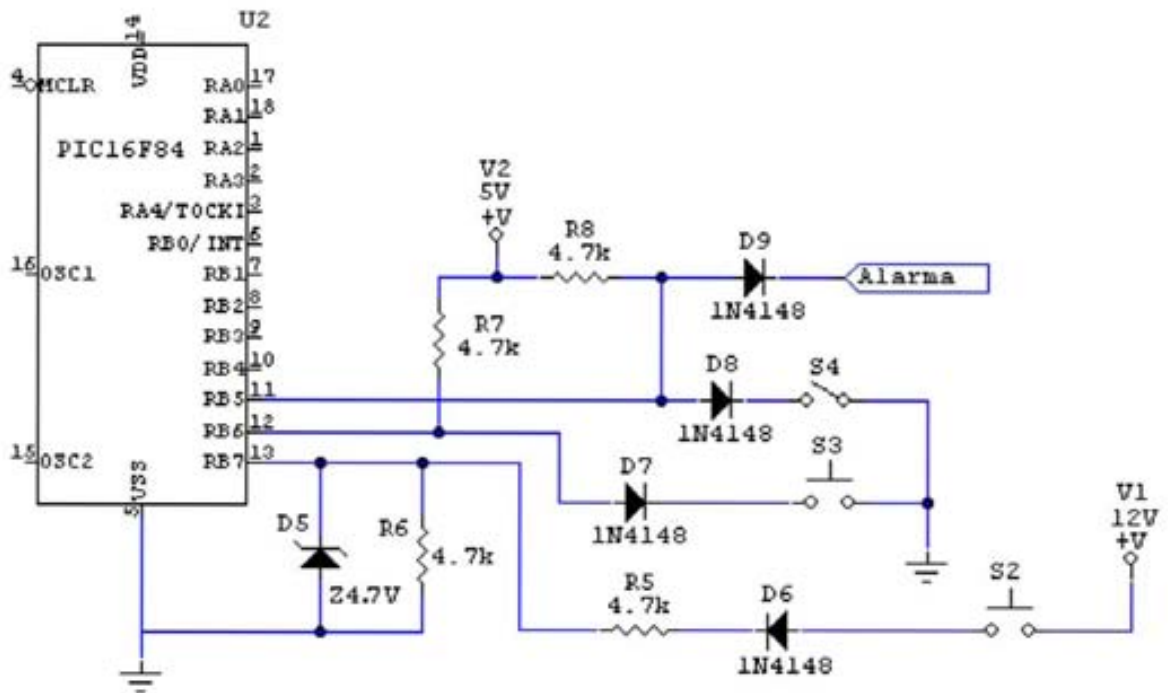


Figura 6.3. Diagrama electrónico de los Sensores.

De la figura 6.3 observamos los puertos que contiene el microcontrolador PIC, de estos utilizamos **RB5**, **RB6** y **RB7** para la conexión de los sensores, aunque cabe mencionar que se pueden utilizar mas e incluso a cada uno de los seleccionados bastara agregar un diodo como se ve en la figura 6.4, ya que estos harán la función de una puerta OR, permitiendo de esta manera conectar varios sensores.

RB5 lo vamos a utilizar para activar y desactivar el sistema, para esto se conecta la resistencia R8 de 4.7[kΩ] a 0.5[W] para forzarlo a 5[V], para que el sistema se active bastará con que por cualquier entrada ya sea por el diodo D8 mediante el interruptor S4 o el diodo D9 mediante la salida de la alarma a la que se conecte el sistema se conecten a 0[V]

Para **RB6** se conecta la resistencia R7 de 4.7[kΩ] a 0.5[W] para forzarlo a 5[V], este sensor S3 disparará la alarma al conectarlo a 0[V]

Para **RB7** el sensor S2 disparará la alarma al conectarlo a 12[V], este se conecta a un divisor de tensión con la resistencia R5 de 4.7[kΩ] a 0.5[W] y la resistencia R6 de 4.7[kΩ] a 0.5[W] y se protege con el diodo zener D5 de 4.7[V].

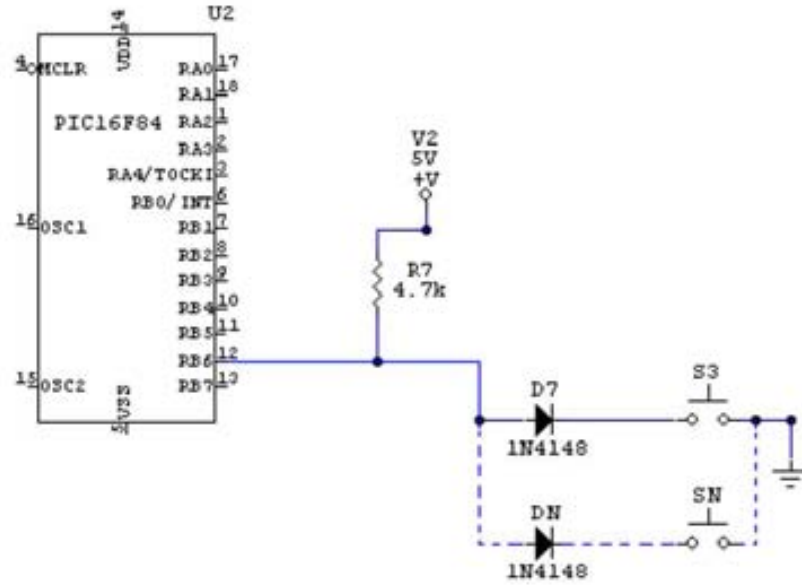


Figura 6.4. Puerta OR para RB6.

6.5 Dispositivo Rx/Tx Teléfono GSM.

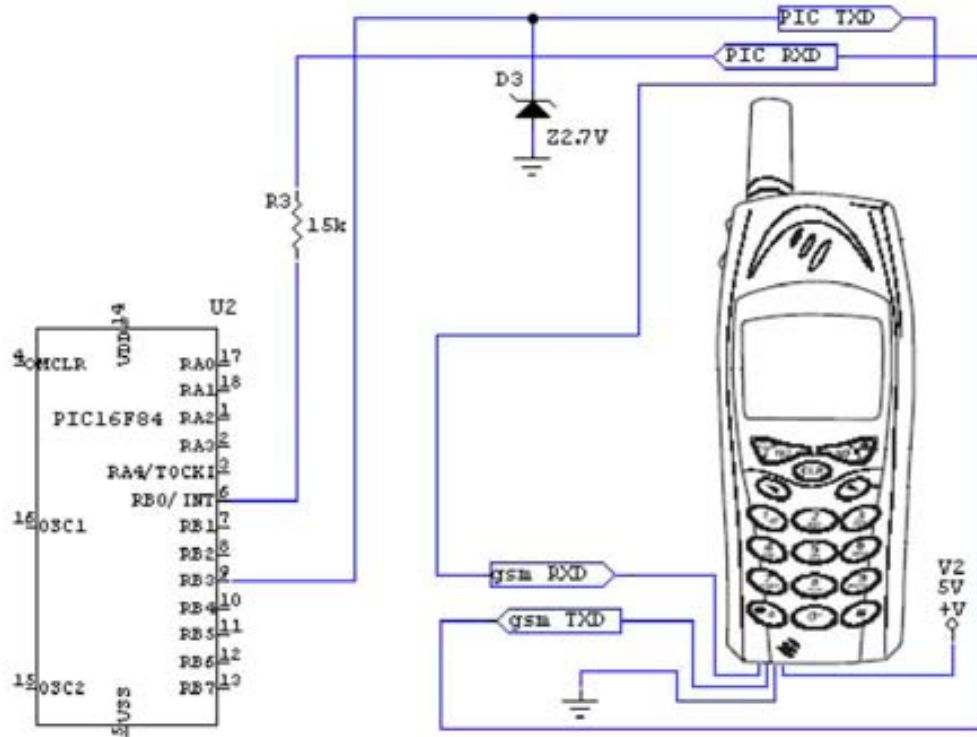


Figura 6.5. Conexión PIC Rx/Tx hacia el Teléfono GSM.

El dispositivo GSM (teléfono celular GSM) se conecta usando un cable de cuatro hilos, dos correspondientes a la alimentación, los 5V los tomamos de V2 y tierra (que es también masa de datos) y otros dos que son la entrada (gsm RXD) y la salida (gsm TXD) de datos serie.

La salida (gsm TXD) va al puerto del PIC **RB0** para recibir los datos del dispositivo GSM (teléfono celular GSM) es decir RXD porque podemos programar este como otra forma de interrupciones según vimos en el capítulo del microcontrolador y la entrada (gsm RXD) se conecta al puerto **RB3** para transmitir datos hacia el dispositivo GSM (teléfono celular GSM) es decir TXD, para este propósito cabe mencionar que se puede utilizar el que se quiera de los puertos sobrantes, solo hay que tener en cuenta el diodo zener D3 que funcionará como protección para el dispositivo GSM (teléfono celular GSM).

Como se puede observar de la figura 6.5 y se deduce del texto anterior, la conexión entre estos va cruzada, es decir que la salida del dispositivo GSM (teléfono celular GSM) va a la entrada en el PIC, y la salida del PIC va a la entrada del dispositivo GSM (teléfono celular GSM), lo que se conoce como configuración de modem nulo.

Después de hacer varias pruebas existen 2 tipos de valores de los dispositivos GSM para el voltaje en el modem por lo que se pueden utilizar diodos zener como D3 = 2.7 [V] o 4.7[V] para mantener el voltaje constante, para nuestro teléfono se utiliza la primera. La resistencia R3 de 15[k Ω] a 0.5[W] se utiliza para limitar la corriente utilizada al transmitir datos el dispositivo GSM.

6.6 Acondicionador RS232.

El acondicionador RS232 lo vamos a utilizar con la finalidad de realizar las pruebas necesarias para ver la comunicación serial del microcontrolador y del dispositivo GSM con la computadora. Primeramente lo hacemos por separado ya después lo hacemos derivando la señal de la conexión entre el microcontrolador y el dispositivo GSM.

Para la parte de la comunicación serial del microcontrolador con la computadora utilizamos el diagrama de la figura 6.6, este fue sacado de la hoja del fabricante utilizando el circuito integrado U3 MAX232, no se necesitan muchos componentes y es muy práctico, tan solo son necesarios 4 capacitores electrolíticos C8, C9, C10 y C11 de 1[μ F]. Se utiliza también un conector macho DB9 el cual va a ir conectado en cualquier puerto serial RS232 de la computadora

Este circuito integrado tiene capacidad para conectar hasta dos conexiones seriales, solo va a ser necesaria una, los pines R1 OUT(salida) y el T1 IN(entrada) como lo observamos en la figura 6.6, los cuales van a ir a los puertos del microcontrolador como se ve en la figura 6.5. De esta conexión vamos a poder observar como se realiza la comunicación serial para poder programar y depurar el programa del microcontrolador PIC

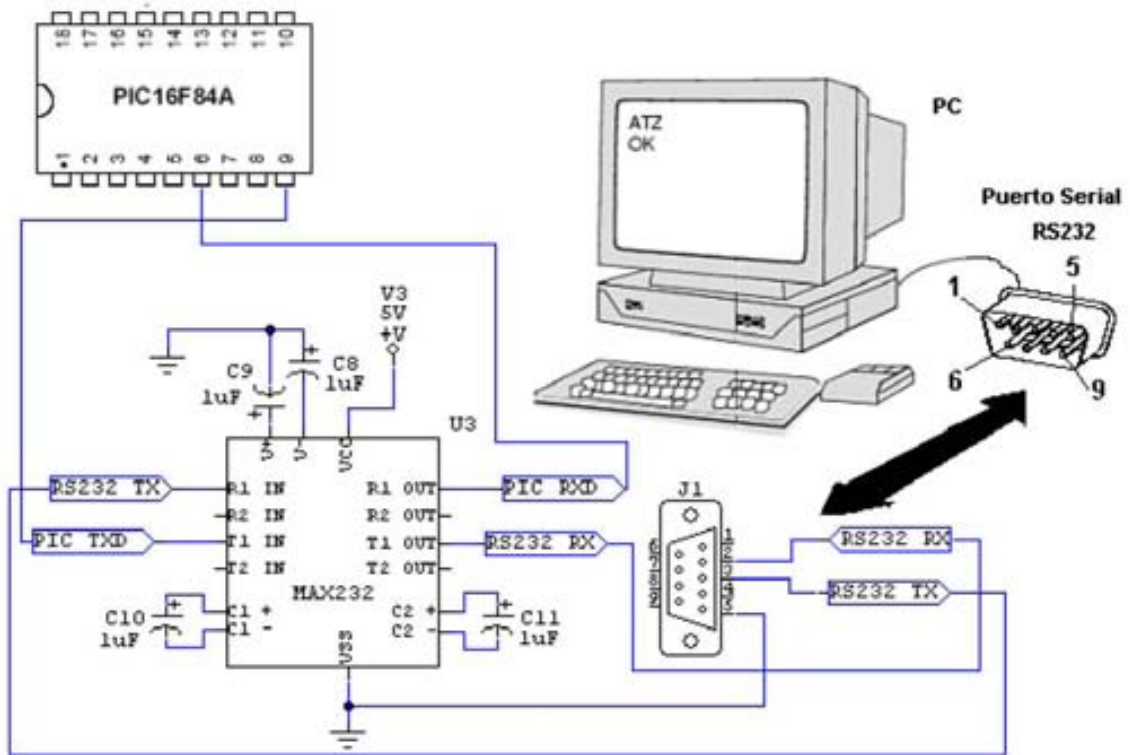


Figura 6.6. Acondicionador RS232 Conexión PIC Rx/Tx hacia la PC.

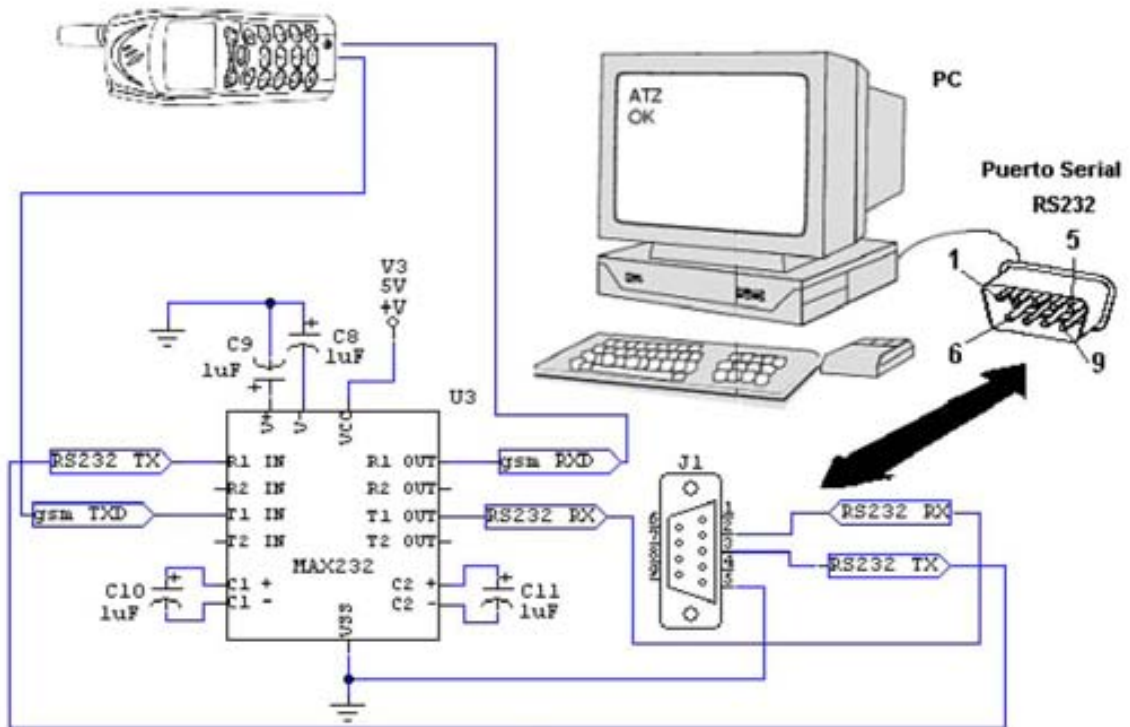


Figura 6.7. Acondicionador RS232 Conexión GSM Rx/Tx hacia la PC.

Para la parte de la comunicación serial del dispositivo GSM con la computadora utilizamos el diagrama de la figura 6.7 el cual es el mismo que la figura 6.6 solo que ahora en lugar de conectar el microcontrolador PIC vamos a conectar nuestro dispositivo GSM o teléfono celular GSM, de la figura 6.5 observamos que tan solo son necesarios tres cables Rx, Tx y Tierra, para ver donde están ubicados estas terminales basta con observar la figura 2.22 para ver la configuración de estos.

De esta conexión vamos a obtener la información mas importante del proyecto, la comunicación serial del dispositivo GSM (teléfono), la obtención de las tramas para poder llevar a cabo el envío y recepción de los mensajes SMS, para este propósito es que fue necesario la elaboración del programa de comunicación entre la PC y el teléfono del capítulo anterior. Ya que una vez visto todo lo anterior solo fue necesario la programación de esto en el microcontrolador PIC para el correcto funcionamiento del proyecto.

Por último teniendo la conexión entre el microcontrolador PIC y el teléfono GSM como se muestra en la figura 6.5 conectamos en forma de derivación el acondicionador RS232 para ver como se lleva a cabo la comunicación entre ellos y en un momento dado realizar una depuración del programa para su mejor funcionamiento.

6.7 Indicador de estados del PIC.

La única finalidad del diagrama de la figura 6.8 es la de poder visualizar de una manera rápida el estado que guarda el sistema; si esta funcionando, si existe comunicación entre el microcontrolador y el dispositivo GSM, si esta bloqueado en fin todo lo que queramos saber lo programamos en el microcontrolador para que por medio del puerto **RA3** nos lo informe mediante un led D4, para limitar la corriente hacia este ponemos una resistencia R4 de 330[Ω] a 0.5[W]

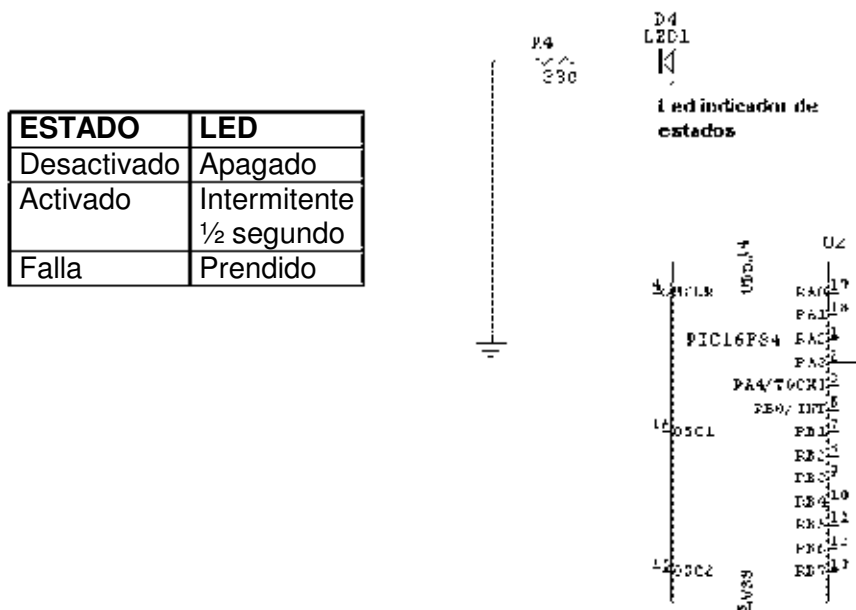


Figura 6.8. Led indicador de estados del Microcontrolador PIC.

CONCLUSIONES

- **El sistema GSM es complejo**, sin embargo los módems GSM nos permite hacer abstracción de la red GSM y utilizar sus servicios desde nuestras aplicaciones.
- La forma de controlar los Módems y de acceder a todos los servicios es mediante los **comandos AT, y AT+**
- El acceso físico al módem es mediante una **línea serie**, con niveles RS232 (módem gsm para PC) o con niveles TTL
- Para realizar aplicaciones en el PC que accedan a los servicios de GSM y en concreto al servicio de envío/recepción de SMS hay que tener información sobre:
 - **Acceso al puerto serie desde el PC**. Esto depende del sistema operativo empleado
 - Descripción y sintaxis de los **comandos AT y AT+**
- La complejidad de la aplicación no depende en sí de los servicios GSM que usa sino de la funcionalidad que se le quiera dar. Los mecanismos de acceso al módem para enviar/recibir SMS son sencillos.
- Conocer más el funcionamiento de los microcontroladores y ver que tienen un gran campo de aplicación todo dependerá que uso les queramos dar. El utilizar las herramientas de desarrollo de estos facilita en gran medida su programación, aunque nunca se comparara a saber manejar el ensamblador ya que este reduce la cantidad de memoria a utilizar además de ser más preciso en tiempos críticos.
- El sistema desarrollado me permitió adquirir experiencia en el desarrollo de aplicaciones para la PC utilizando comunicación serial.
- Los alcances que tiene nuestro sistema son vastos, aquí solo proporcionamos unos ejemplos de lo que es capaz de hacer, todo dependerá de la información que queramos obtener de las alarmas, como de lo que queramos controlar si deseamos hacerlo bidireccional.

APÉNDICES

APÉNDICE A

Código de 7 bits utilizado en la conversión de texto a PDU.

				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	2	0	1	0	1
b4	b3	b2	b1		0	1	2	3	4	5	6	7
0	0	0	0	0	@	Δ	SP	0	-	P	”	p
0	0	0	1	1			!	1	A	Q	a	q
0	0	1	0	2	\$	Φ	”	2	B	R	b	r
0	0	1	1	3		Γ	#	3	C	S	c	s
0	1	0	0	4		Λ		4	D	T	d	t
0	1	0	1	5		Ω	%	5	E	U	e	u
0	1	1	0	6		Π	&	6	F	V	f	v
0	1	1	1	7		Ψ	‘	7	G	W	g	w
1	0	0	0	8		Σ	(8	H	X	h	x
1	0	0	1	9		Θ)	9	I	Y	i	y
1	0	1	0	10	LF	Ξ	*	:	J	Z	j	z
1	0	1	1	11			+	;	K	Ä	k	ä
1	1	0	0	12			,	<	L	Ö	l	ö
1	1	0	1	13	CR		-	=	M		m	
1	1	1	0	14		β	.	>	N	Ü	n	ü
1	1	1	1	15			/	?	O		o	

APÉNDICE B

Código Visual Basic del programa de comunicación.

Módulo de conversión Hexadecimal a Decimal .

```
Dim HexArray(255) As String 'Arreglo de conversión

Public ConvThisText As String

Public Sub ConvToText(Text As String)
'Esto convierte de hexadecimal a texto

    Dim MainTextTemp As String
    Dim B As Long
    Dim X As String

'Prepara el arreglo hex de conversión
    For B = 0 To 255
        HexArray(B) = Hex(B)
    Next B

    MainTextTemp = Text

    Text = ""

    For B = 1 To Len(MainTextTemp) Step 2
        X = Mid(MainTextTemp, B, 2)

        Select Case X 'arregla los digitos solos de numeros
            Case "00"
                X = "0"
            Case "01"
                X = "1"
            Case "02"
                X = "2"
            Case "03"
                X = "3"
            Case "04"
                X = "4"
            Case "05"
                X = "5"
            Case "06"
                X = "6"
            Case "07"
                X = "7"
            Case "08"
                X = "8"
            Case "09"
                X = "9"
            Case "0A"
                X = "A"
            Case "0B"
                X = "B"
            Case "0C"
                X = "C"
            Case "0D"
                X = "D"
            Case "0E"
                X = "E"
            Case "0F"
                X = "F"
        End Select

        Text = Text & Chr(Dec(X))
    Next B
```



```
MainTextTemp = ""
X = ""

'Limpia el arreglo de la memoria
Erase HexArray

End Sub

Public Sub ConvToHex(Text As String)
'Esto convierte de texto a hexadecimal
Dim B As Integer
Dim MainTextTemp As String
Dim X As String

MainTextTemp = Text

Text = ""

For B = 1 To Len(MainTextTemp)
X = Hex(Asc(Mid(MainTextTemp, B, 1)))

'corrige los digitos solos
If Len(X) = 1 Then
X = "0" & X
End If

Text = Text & X
Next B

MainTextTemp = ""
X = ""

End Sub

Public Function Dec(Y)

Dim N As Integer

For N = 0 To 255
If Y = HexArray(N) Then
Dec = N
GoTo Next1
End If
Next N
Next1:

End Function
```

Módulo de formatos y limitaciones de texto.

Option Explicit

Public exitPause As Boolean

Public Function timedPause(secs As Long)

Dim secStart As Variant

Dim secNow As Variant

Dim secDiff As Variant

Dim Temp%

exitPause = False 'para tener una pausa

secStart = Format(Now(), "mm/dd/yyyy hh:nn:ss AM/PM")

Do While secDiff < secs

 If exitPause = True Then Exit Do

 secNow = Format(Now(), "mm/dd/yyyy hh:nn:ss AM/PM")

 secDiff = DateDiff("s", secStart, secNow) 'compara el tiempo de inicio con el actual

 Temp% = DoEvents

Loop

End Function

'Para limitar los textos de numeros

Function LimTexNumero(Source) As String

 'Caracteres permitidos

 Const Numbers\$ = "0123456789"

 'backspace =8

 If Source <> 8 Then

 If InStr(Numbers, Chr(Source)) = 0 Then

 LimTexNumero = 0

 Exit Function

 End If

 End If

 LimTexNumero = Source

End Function

Módulo de conversión de texto a PDU .

```
Public Function txt2pdu(ByVal stxt As String) As String

    For Y = 1 To Len(stxt)
        X = Dec2Bin(Asc(Mid(stxt, Y, 1)))
        'If X = "100000" Then X = "0100000"
        'Debug.Print X
        Text = X & Text
    Next Y
    Debug.Print Text

    l = Len(Text)
    s = l - 7
    out = Bin2Hex(Mid(Text, s, 8))
    Debug.Print Mid(Text, s, 8)
    Debug.Print out

    Do Until s <= 8
        s = s - 8
        out = out & Bin2Hex(Mid(Text, s, 8))
        Debug.Print Mid(Text, s, 8)
        Debug.Print out
    Loop

    out = out & Bin2Hex(Mid(Text, 1, s - 1))
    Debug.Print Mid(Text, 1, s - 1)
    Debug.Print out

    X = Hex(Len(stxt))

    If Len(X) = 1 Then
        X = "0" & X
    End If

    Debug.Print X & out
    txt2pdu = X & out

End Function

Public Function Dec2Bin(ByVal nDec As Integer) As String
    'Esta funcion es la misma de Hex2Bin,
    ' pero ha sido copiada para acelerar el proceso
    Dim i As Integer
    Dim j As Integer
    Dim sHex As String
    Const HexChar As String = "0123456789ABCDEF"

    sHex = Hex(nDec) 'Esta la unica parte que es diferente

    For i = 1 To Len(sHex)
        nDec = InStr(1, HexChar, Mid(sHex, i, 1)) - 1

        For j = 3 To 0 Step -1
            Dec2Bin = Dec2Bin & nDec \ 2 ^ j
            nDec = nDec Mod 2 ^ j
        Next j
    Next i
    'Remueve el primer sin usar 0
    i = InStr(1, Dec2Bin, "1")
    If i <> 0 Then Dec2Bin = Mid(Dec2Bin, i)
    If Len(Dec2Bin) = 6 Then Dec2Bin = "0" & Dec2Bin

End Function
```

```
Public Function Bin2Hex(ByVal sBin As String) As String
    Dim i As Integer
    Dim nDec As Long
    sBin = String(4 - Len(sBin) Mod 4, "0") & sBin 'Agrega un cero para completar el Byte

    For i = 1 To Len(sBin)
        nDec = nDec + CInt(Mid(sBin, Len(sBin) - i + 1, 1)) * 2 ^ (i - 1)
    Next i
    Bin2Hex = Hex(nDec)
    If Len(Bin2Hex) Mod 2 = 1 Then Bin2Hex = "0" & Bin2Hex

End Function
```

Forma principal del programa.

```
Dim numpuerto, msg, pdu As String
Dim longtxt As Integer

Private Sub cboPuerto_Click()

    ' Selecciona el puerto y lo agrega a la variable numpuerto
    numpuerto = cboPuerto.ItemData(cboPuerto.ListIndex)

End Sub

Private Sub cmdAbrePuerto_Click()
    On Error GoTo errAbrir
    ' Configuracion del Puerto
    MSComm1.CommPort = numpuerto
    MSComm1.Settings = "115200,N,8,1"
    MSComm1.PortOpen = True

    ' Evita seleccionar otro puerto
    cboPuerto.Enabled = False
    cmdCierraPuerto.Enabled = True
    fraEntrada.Enabled = True
    fraEnvio.Enabled = True
    fraSalida.Enabled = True
    cmdAbrePuerto.Enabled = False

errAbrir:
    If Err.Number = 8002 Then
        msg = "Error # " & Str(Err.Number) & " generado por " _
            & Err.Source & Chr(13) & "Número de puerto inválido, seleccione otro."
        MsgBox msg, , "Error", Err.HelpFile, Err.HelpContext
    End If

End Sub

Private Sub cmdCierraPuerto_Click()

    ' Habilita seleccionar otro puerto
    cboPuerto.Enabled = True
    cmdAbrePuerto.Enabled = True
    fraEntrada.Enabled = False
    fraSalida.Enabled = False
    fraEnvio.Enabled = False
    cmdCierraPuerto.Enabled = False
    MSComm1.PortOpen = False
    txtSalida.Text = ""

End Sub

Private Sub cmdEnvia_Click()
    On Error GoTo errEnvia

    ' Limpia primero la salida
    txtSalida.Text = ""

    ' Envia comandos al celular
    MSComm1.Output = txtEntrada.Text & vbCr

errEnvia:
    If Err.Number <> 0 Then
        msg = "Error # " & Str(Err.Number) & " generado por " _
            & Err.Source & Chr(13) & Err.Description
        MsgBox msg, , "Error", Err.HelpFile, Err.HelpContext
    End If

End Sub
```

```
Private Sub cmdEnviaMensaje_Click()

    ' Para verificar que exista que convertir
    If txtSMSTexto.Text = "" Then Exit Sub

    ' Limpia primero la salida
    txtSalida.Text = ""

    ' Para crear la trama PDU
    cmdVerPDU_Click

    ' Para obtener la longitud de la trama PDU
    ' Total de la longitud - 4 por quitar SMSC
    ' que es 01 y 80 ddel SCA y entre dos
    tamPDU = CStr((Len(pdu) - 4) / 2)

    ' Prepara el envio y hace una pausa
    MSComml.Output = "AT+CMGS=" & tamPDU + vbCr
    'timedPause 1

    ' chr$(26) significa ctrl-Z necesario para
    ' enviar el mensaje despues de introducir
    ' la trama PDU
    MSComml.Output = txtPDU.Text + Chr$(26)

End Sub

Private Sub cmdLimpiar_Click()
    ' Para limpiar el texto de salida
    txtSalida.Text = ""
End Sub

Private Sub cmdTxtNuevo_Click()
    ' Para limpiar el texto de mensaje
    txtSMSTexto.Text = ""
    txtPDU.Text = ""
End Sub

Private Sub cmdVerPDU_Click()
Dim Na, Nb, Nc, Nd, Ne, Nf, Ng, Nh, Ni, Nj As String
Dim SCAIDMR, DA1, DA2, PIDDCSVP As String

    ' Para verificar que exista que convertir
    If txtSMSTexto.Text = "" Then Exit Sub

    ' Esto se necesita como encabezado del mensaje
    ' SCA = 01 80
    ' ID = 11
    ' MR = 00
    ' Primeros bytes del DA1 = 0A 81
    SCAIDMR = "01801100"
    DA1 = "0A81"

    ' Aqui se convierte el # Cel al que se envia SMS
    ' Ejemplo: 5 5 1 7 8 7 8 4 4 0 -----> 5 5 7 1 7 8 4 8 0 4
    For i = 1 To 10
        Select Case i
            Case 1
                Na = Mid(txtNumCel.Text, 1, 1)
            Case 2
                Nb = Mid(txtNumCel.Text, 2, 1)
            Case 3
                Nc = Mid(txtNumCel.Text, 3, 1)
            Case 4
                Nd = Mid(txtNumCel.Text, 4, 1)
            Case 5
                Ne = Mid(txtNumCel.Text, 5, 1)
            Case 6
                Nf = Mid(txtNumCel.Text, 6, 1)
        End Select
    Next i
End Sub
```

```
        Case 7
            Ng = Mid(txtNumCel.Text, 7, 1)
        Case 8
            Nh = Mid(txtNumCel.Text, 8, 1)
        Case 9
            Ni = Mid(txtNumCel.Text, 9, 1)
        Case 10
            Nj = Mid(txtNumCel.Text, 10, 1)
    End Select
Next

' DA2 segunda parte del numero que recibira SMS
DA2 = Nb & Na & Nd & Nc & Nf & Ne & Nh & Ng & Nj & Ni

' Otra parte del encabezado también
' PID = 00
' DCS = 00
' VP = A7
PIDDCSVP = "0000A7"

' Esto convierte el texto del sms en
' datos PDU de 7 bit
pduTexto = txt2pdu(txtSMSTexto.Text)

' Completa la trama PDU
pdu = SCAIDMR & DA1 & DA2 & PIDDCSVP & pduTexto

' Para ver la trama PDU
txtPDU.Text = pdu

End Sub

Private Sub Command1_Click()
    ' Limpia primero la salida
    txtSalida.Text = ""

    ' Envia comandos al celular
    MSComml.Output = txtEntrada.Text & vbCr
End Sub

Private Sub Form_Load()

    ' Carga el combo con los puertos disponibles
    cboPuerto.AddItem "COM 1"
    cboPuerto.ItemData(cboPuerto.NewIndex) = 1
    cboPuerto.AddItem "COM 2"
    cboPuerto.ItemData(cboPuerto.NewIndex) = 2
    cboPuerto.AddItem "COM 3"
    cboPuerto.ItemData(cboPuerto.NewIndex) = 3
    cboPuerto.AddItem "COM 4"
    cboPuerto.ItemData(cboPuerto.NewIndex) = 4
    cboPuerto.AddItem "COM 5"
    cboPuerto.ItemData(cboPuerto.NewIndex) = 5
    cboPuerto.AddItem "COM 6"
    cboPuerto.ItemData(cboPuerto.NewIndex) = 6
    cboPuerto.AddItem "COM 7"
    cboPuerto.ItemData(cboPuerto.NewIndex) = 7

    ' Selecciona el puerto 2 por default
    ' y lo agrega a la variable numpuerto
    cboPuerto.ListIndex = 1
    numpuerto = cboPuerto.ItemData(cboPuerto.ListIndex)

    ' Máximo de caracteres del mensaje de texto
    txtSMSTexto.MaxLength = 160

End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)

    ' Cierra el Puerto
    If MScmm1.PortOpen = True Then
        MScmm1.PortOpen = False
    End If

End Sub

Private Sub Timer1_Timer()

    ' Si esta abierto el puerto empieza escribir
    If MScmm1.PortOpen = True Then
        Me.txtSalida = Me.txtSalida + MScmm1.Input
        longtxt = Len(txtSMSTexto.Text)
        lstCaracter.Clear
        lstCaracter.AddItem longtxt

    End If

End Sub

Private Sub txtNumCel_KeyPress(KeyAscii As Integer)
    ' Para limitar solo escribir numeros
    KeyAscii = LimTexNumero(KeyAscii)
End Sub
```


APÉNDICE C

Código en PICBasic Pro para el microcontrolador PIC16F84.

```

'*****
'* Tesis      : SISTEMA DE ADQUISICIÓN DE ESTADOS DE ALARMAS CON TECNOLOGIA GSM *
'*           : SOBRE LA BASE DE UN MICROCONTROLADOR.                       *
'* Autor     : Salvador Orozco Gutiérrez                                   *
'* Fecha    : 22/03/06                                                    *
'* Notas    : El microcontrolador es un PIC16F84A                         *
'*****

'-----
' DECLARACIÓN DE PUERTOS Y CONSTANTES.
'-----
' IO de la comunicación serial
T9600 CON 84           ' Velocidad transmisión 9600bps
CR      CON 13         ' (CR) Carrige Return - Retorno de carro
LF      CON 10         ' (LF) Line Feed - Alimentación de línea
SYMBOL TX = PORTB.3   ' Puerto PIC Tx PB3
SYMBOL RX = PORTB.0   ' Puerto PIC Rx PB0

' Sensores
SYMBOL HABILITA      = PORTB.5   ' Habilita el sistema de adquisición
SYMBOL SENSORBAJO   = PORTB.6   ' Sensor activa a 0[V]
SYMBOL SENSORALTO  = PORTB.7   ' Sensor activa a 12[V]

' LED para visualizar estados
SYMBOL ESTADO = PORTA.3         ' LED de Estados

'-----
' DEFINE ENTRADAS, SALIDAS Y ESTADO INICIAL DE LOS PUERTOS.
'-----
INPUT RX           ' Entrada de datos Rx
INPUT HABILITA     ' Entrada para habilitar el sistema
INPUT SENSORBAJO  ' Entrada sensor [0]V
INPUT SENSORALTO  ' Entrada sensor [12]V

OUTPUT TX          ' Salida de datos Tx
OUTPUT ESTADO     ' Salida LED de estados

PORTA =%00000000   ' Estado inicial del Puerto A
PORTB =%00000000   ' Estado inicial del Puerto B

'-----
' DECLARACIÓN DE VARIABLES.
'-----
TMP1  VAR BYTE     ' Registros temporales
TMP2  VAR BYTE
TMP3  VAR BYTE
TMP4  VAR BYTE

TIEMPO VAR BYTE    ' Para tiempos

```

```

NUMCAR VAR BYTE           ' Número de caracteres a enviar
SENSOR VAR BYTE           ' Que sensor se disparó
TEL     VAR BYTE [9]       ' Arreglo para núm. teléfono que envía SMS
pduSMS VAR BYTE [10]      ' Arreglo para el SMS en formato PDU

'-----
' DATOS DE EEPROM INICIALES.
'-----
EEPROM 42, [0]             ' dispara alarma, sensores bajo= 3, alto= 5
EEPROM 43, [5]             ' # de SMS permitidos = 5
EEPROM 44, ["31D98CA602"]  ' Contraseña = 1234*
EEPROM 54, ["5591018432"]  ' Teléfono que recibe SMS = 5519104823

'*****
VERIFICA:
'-----
' Verifica presencia del Teléfono Local enviando comando AT
' Esperando respuesta OK de este.
'-----
ESTADO = 1                 ' Prende LED de estado
SEROUT TX, T9600, ["AT", CR] ' Verifica presencia del teléfono
SERIN  RX, T9600, 2000, VERIFICA, [WAIT ("OK")]
PAUSE 500                  ' Pausa de .5 seg

'-----
' Configura memoria del TEL enviando AT+CPMS="ME", "ME".
' Esperando respuesta OK de este.
'-----
ESTADO = 0                 ' Apaga LED de estado
SEROUT TX, T9600, ["AT+CPMS=", 34, "ME", 34, 44, 34, "ME", 34, CR]
SERIN  RX, T9600, 2000, VERIFICA, [WAIT ("OK")]
PAUSE 500                  ' Pausa de .5 seg

'*****
DESACTIVA:
'-----
' MIENTRAS NO SE HABILITE EL SISTEMA PERMANECE EN ESPERA.
'-----
WHILE HABILITA = 1
  ESTADO = 0               ' Apaga LED de estado
  SLEEP 60                 ' Duerme el micro por 1 min. aprox.
  TIEMPO = 0              ' Inicializa el tiempo
WEND

'*****
INICIO:
IF HABILITA = 1 THEN DESACTIVA ' Se desactiva sistema a 5[V]

IF TIEMPO=180 THEN GOSUB smsCONFIG ' Verifica cada 3 min. si hay smsCONFIG

ESTADO = 1                 ' Prende LED de estado
PAUSE 500                  ' Pausa de .5 seg

```

```

ESTADO = 0           ' Apaga LED de estado
PAUSE 500           ' Pausa de .5 seg

SENSOR=0
IF SENSORBAJO=0 THEN SENSOR="6" ' Se disparó el sensor a 0[V]
IF SENSORALTO=1 THEN SENSOR="A" ' Se disparó el sensor a 12[V]
IF SENSOR<>0 THEN
  NUMCAR = "2"      ' Para enviar el SMS indica 22
  WRITE 42,SENSOR   ' Almacena que sensor disparó la alarma
  READ 43,TMP1      ' SMS permitidos
  IF TMP1=0 THEN INICIO ' TMP1=0 implica ya no enviar SMS
  GOSUB enviaSMS    ' Va a enviar el mensaje SMS
ENDIF

TIEMPO = TIEMPO + 1 ' Incrementa la cuenta del tiempo

GOTO INICIO         ' Regresa y parpadea el LED para siempre

' *****
smsCONFIG:
'-----
' Pregunta si hay mensaje nuevo del TEL enviando AT+CMGL=0
'-----

ESTADO = 1           ' Prende LED de estado
SEROUT TX,T9600,["AT+CMGL=0",CR]
SERIN  RX,T9600,500,noCMGL,[WAIT("+CMGL:")]
SERIN  RX,T9600,[WAIT("0891"),SKIP 20,STR TEL\10,SKIP 20,STR pduSMS\10]

' Compara teléfono y SMS
decSMS:
TMP3=0
TMP4=0
FOR TMP1 = 54 TO 63
  READ TMP1,TMP2      ' Lee el núm. tel. de EEPROM
  IF TMP2<>TEL[TMP1-54] THEN
    TMP3=1           ' Para saber si es diferente
  ENDIF
  READ TMP1-10,TMP2   ' Lee el SMS en formato PDU de EEPROM
  IF TMP2<>pduSMS[TMP1-54] THEN
    TMP4=1           ' Para saber si es diferente
  ENDIF
NEXT TMP1

' Borra el mensaje recibido, para que no se llene la memoria del teléfono
SEROUT TX,T9600,["AT+CMGD=1",CR]

' Si se cumplen condiciones cambia el número del teléfono que recibe SMS
IF TMP3=1 AND TMP4=0 THEN cambiaTEL

' No hubo mensaje SMS de configuración
noCMGL:
ESTADO = 0           ' Apaga LED de estado
RETURN

' Cambia el número que recibe el SMS
cambiaTEL:

```

```

ESTADO = 0                                ' Apaga LED de estado

FOR TMP1 = 0 TO 9                          ' Ciclo para sobre escribir el núm. tel.
  WRITE TMP1+54, TEL,[TMP1]
NEXT TMP1

NUMCAR="1"                                ' Para enviar el SMS indica 21

GOTO enviaSMS

RETURN

'*****
enviaSMS:
'-----
' Envia mensaje enviando AT+CMGS=NUMCAR
' donde NUMCAR es 21 o 22 depende de la cantidad de caracteres del SMS
'-----

ESTADO = 1                                ' Prende LED de estado
SEROUT TX,T9600, ["AT+CMGS=2", NUMCAR, CR]
SERIN RX,T9600, [WAIT("> ")]
PAUSE 5000
SEROUT TX,T9600, ["018011000A81"]

' Ciclo para leer el número de teléfono al que se envía el mensaje SMS
FOR TMP2=54 TO 63
  READ TMP2, TMP3
  SEROUT TX,T9600, [TMP3]
NEXT TMP2
SEROUT2 TX,T9600, ["0000A70"]

IF NUMCAR ="2" THEN
  ' Mensaje "SW1 ACTIVO" depende del EEPROM 48
  ' SEROUT TX,T9600, ["0AD36B0C141C5293D627", 26, CR, LF]
  ' Mensaje "SW2 ACTIVO" depende del EEPROM 48
  ' SEROUT TX,T9600, ["0AD3AB0C141C5293D627", 26, CR, LF]
  SEROUT TX,T9600, ["AD3", SENSOR, "B0C141C5293D627", 26, CR, LF]
ELSE
  ' Mensaje "CAMBIO OK"
  ' SEROUT TX,T9600, ["09C36053987C829E4B", 26, CR, LF]
  SEROUT TX,T9600, ["9C36053987C829E4B", 26, CR, LF]
ENDIF

SERIN2 RX,T9600, 5000, enviaSMS, [WAIT("OK")]

PAUSE 500                                ' Pausa de .5 seg
ESTADO = 0                                ' Apaga LED de estado

WRITE 43, TMP1-1                          ' Decrementa cant. de SMS permitidos

SLEEP 120                                  ' Duerme el micro por 2 min. aprox.

RETURN

END

```

BIBLIOGRAFÍA

JOSÉ ADOLFO GONZÁLEZ V. **Introducción a los microcontroladores** , Editorial McGraw Hill

VALLEJO HORACIO D. **Microcontroladores PIC**, Argentina, 2002
Editorial Quark

MICROCHIP. **Microchip MPLAB IDE User's Guide**,. www.microchip.com

LUIS FRINO. **Traducción al español del Manual Original del Pic Basic Compiler Pro**,
Argentina 2002 ,. www.frino.com.ar

Referencias

[1] Etsi. European Telecommunications Standards Institute. www.etsi.org

[2] "Technical realization of the Short Message Service (SMS) Point-to-Point". GSM-03.40.
Version 5.3.0. ETSI.

[3] Busqueda en google: "AT hayes commands".

[4] Descripción de los comandos AT de Hayes. <http://www.cellular.co.za/hayesat.htm>.

[5] Otra referencias de comandos AT. <http://www.modemhelp.net/basicatcommand.shtml>