

UNIVERSIDAD NACIONAL DE INGENIERIA

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



DISEÑO DE UN SISTEMA DE MONITOREO DE ALERTAS MULTI AREA VIA ETHERNET/GSM/GPRS APLICADO A UNA PLANTA CON VIGILANCIA CENTRALIZADA

INFORME DE SUFICIENCIA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PRESENTADO POR:

ANGEL MARTIN VILCA CHUQUICAJA

**PROMOCIÓN
2009 - II**

**LIMA – PERÚ
2013.**

**DISEÑO DE UN SISTEMA DE MONITOREO DE ALERTAS
MULTI AREA VIA ETHERNET/GSM/GPRS APLICADO A
UNA PLANTA CON VIGILANCIA CENTRALIZADA**

A mi Familia:
Mis padres Angel y Mercedes(in Memoriam),
Mis Hermanos,
Mi esposa Sharon y mi hijo Angel Hans.
Por todo el apoyo y la confianza.

SUMARIO

En el presente trabajo se describe el desarrollo del diseño de un sistema de seguridad electrónica, aplicado a una planta con vigilancia centralizada, constituyéndose en una herramienta útil para el personal encargado de la seguridad corporativa. De esta manera las diferentes áreas de la planta deben estar protegidas por sub-sistemas de alarma independientes, donde los eventos de alarma son monitoreados por una estación central, utilizando como medio de comunicación la infraestructura de red de la planta y adicionalmente de forma inalámbrica aplicando tecnología GPRS/GSM.

Para el diseño de los sistemas de alarma en cada área, se propone una aplicación basado en tecnología de microcontroladores PIC, principalmente por constituirse en una familia de amplia gama de prestaciones, que permite realizar las tareas requeridas, tanto el control de los sensores de alarma, como el manejo de salidas para la conexión a dispositivos de salida audible, dispositivos de visualización LCD y también para comandar las interfaces de comunicación específicas, en este caso las tarjetas Ethernet Shield y Modem GPRS/GSM.

El monitoreo de las señales provenientes de cada área, se realiza mediante la ejecución de un software, que se ejecuta en un computador ubicado en la estación central; al ejecutarse estará siempre en estado de "Escucha", para ante cualquier evento reportado por las unidades instaladas en las áreas, realice las acciones programadas. Estas acciones comprenden: la generación de alertas audibles en el caso de eventos de alarma, el registro ordenado de todos los eventos del sistema, envío de mensajes de texto a una o a todas las áreas, envío de alertas audibles a una o a todas las áreas.

En conjunto se conforma un sistema fácilmente escalable, que complementa la seguridad de una planta, sistematizando el proceso de supervisión de seguridad con la invaluable ayuda de la Ingeniería Electrónica y las TIC (Tecnología de la Información y la Comunicación).

INDICE

INTRODUCCION	1
CAPITULO I	
PLANTEAMIENTO DE INGENIERIA DEL PROBLEMA	2
1.1 Descripción del problema	2
1.2 Objetivo del trabajo.....	3
1.3 Evaluación del problema	3
1.4 Alcance del trabajo	3
1.5 Síntesis del Trabajo.....	4
CAPITULO II	
MARCO TEORICO	5
2.1 Microcontroladores PIC	5
2.2 Lenguaje de Programación C	5
2.3 Software de Simulación	5
2.4 El módulo integrado de comunicación serial asíncrona MSSP	6
2.4.1 Comunicación serial SPI	6
2.5 Protocolo Ethernet.....	7
2.5.1 Códigos.....	7
2.5.2 La trama Ethernet.....	7
2.6 Ethernet Shield W5100.....	7
2.6.1 Aplicaciones	7
2.6.2 Especificaciones Técnicas.....	8
2.7 Sistema global para las comunicaciones móviles GSM	8
2.7.1 Mensajes de texto SMS.....	9
2.7.2 HSCSD Datos por Conmutación de Circuitos de Alta Velocidad.....	9
2.7.3 GPRS (General Packet Radio System)	9
2.7.4 Modem GSM/GPRS	10
2.8 Arquitectura Cliente Servidor (Sockets).....	11
CAPITULO III	
DISEÑO DEL SISTEMA	12
3.1 Componentes del Sistema.....	12
3.2 La Unidad Remota.....	13
3.2.1 Fuente de Alimentación	13
3.2.2 Unidad de control	14
3.3 Software de Monitoreo	28
3.3.1 Archivos de Inicialización.....	29
3.3.2 Comunicación vía Sockets	30
CAPITULO IV	
ANALISIS Y PRESENTACION DE RESULTADOS	35

4.1	Análisis descriptivo	35
4.1.1	Unidad Remota	35
4.1.2	Centro de control	35
4.2	Estimación de costos en Equipos y Componentes	36
4.3	Proyección de Implementación.....	36
CONCLUSIONES Y RECOMENDACIONES		38
ANEXO A		
EI MICROCONTROLADOR PIC 16F877A		39
ANEXO B		
ETHERNET SHIELD W5100		50
ANEXO C		
MODEM GPRS/GSM FASTRACK M1306B		62
BIBLIOGRAFIA		69

INTRODUCCION

Los microcontroladores PIC al englobar un ordenador programable cada vez con mayores niveles de complejidad y prestaciones son una opción viable para múltiples soluciones; en el caso de la seguridad electrónica se puede aplicar para implementaciones de alarmas, para operar diferentes tipos de sensores como detectores de apertura, detectores de movimiento, pulsadores de emergencia.

Agregando conectividad a un sistema de alarmas basado en microcontroladores, se puede estructurar un sistema con unidades remotas múltiples monitoreadas desde un centro de control. La conectividad primaria se realiza a través de la red Ethernet. Para lograr dicha conectividad a través de la red Ethernet se plantea la aplicación conjunta de módulos Ethernet, que servirán de interface al microcontrolador PIC permitiendo la asignación de direcciones IP y direcciones de hardware MAC ADDRESS a cada unidad remota, para finalmente lograr la comunicación con un ordenador central, donde a través de software se realizará el control de los eventos generados por el sistema de alarmas en cada unidad remota.

Con el objetivo de aumentar el nivel de seguridad se plantea la aplicación de dispositivos GSM/GPRS que permiten comunicar eventos del sistema utilizando las redes de telefonía móvil e Internet, para comunicación inalámbrica a cualquier distancia.

CAPITULO I PLANTEAMIENTO DE INGENIERÍA DEL PROBLEMA

En este capítulo se realiza el planteamiento de ingeniería del problema, a partir de la descripción del problema que se desea solucionar, se fijan los objetivos a alcanzar, producto de la evaluación se presentan los alcances del trabajo, finalizando con la presentación de una síntesis del diseño propuesto.

1.1 Descripción del problema

Se parte enfocando el escenario del presente estudio: Una planta que está dividida en varias áreas secundarias, donde cada área cuenta con vigilancia personal que se encarga de controlar los ingresos – egresos personal y vehicular, así como salvaguardar los bienes patrimoniales; la supervisión general está a cargo de un supervisor de seguridad, cuyo centro de operaciones es el centro de control, ubicación donde se centraliza las comunicaciones vía radio, control de personal de seguridad. En la figura 1.1 se muestra un esquema de la planta con vigilancia centralizada.

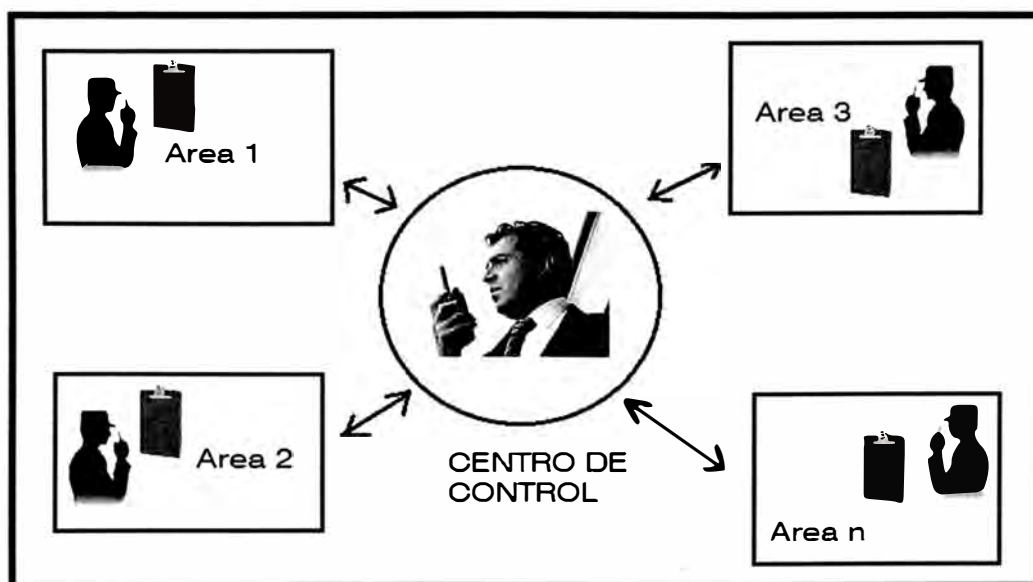


Figura 1.1 Planta con vigilancia centralizada. (Fuente: Diseño propio)

El nivel de seguridad no es la óptima porque se encuentra supeditada al factor humano, que por negligencia o dolo pueden dejar de registrar eventos tales como acceso indebido a áreas restringidas, pérdida de bienes patrimoniales, con el consiguiente perjuicio a la entidad.

1.2 Objetivo del trabajo

Desarrollar una solución de seguridad basada en sistemas electrónicos, que permita tener un registro unívoco de eventos tales como acceso y tránsito relacionados con la seguridad. Mediante la instalación de sensores de apertura para puertas, sensores volumétricos de movimiento, que coadyuven a la mejora del nivel de seguridad de la planta.

El sistema debe ser monitoreado ininterrumpidamente mediante sistemas informáticos desde el centro de control, para tener un registro de eventos relacionados con la seguridad.

1.3 Evaluación del problema

En un área específica se presentan situaciones o eventos que deben ser registradas por el sistema de seguridad, por ejemplo: acceso indebido, alertas de emergencia, apertura de puertas en horario restringido, amago de aniego, rotura de vidrios, sabotaje entre otros.

Para tener un control de todos los eventos que se generen en las distintas áreas, se requiere que los reportes se registren en el centro de control.

La implementación de redes locales informáticas está bien difundida en nuestro medio, y el promedio de las empresas cuenta con una red informática local, por lo que es posible aprovechar toda esta infraestructura para mantener la comunicación del sistema de seguridad propuesto.

La oferta actual en el medio por parte de las empresas que brindan servicios de seguridad electrónica se basa principalmente en sistemas de alarmas con comunicación telefónica, y generalmente ofrecen paquetes de servicios con monitoreo a través de un centro de control propio de las empresas de seguridad. Con un sistema que utilice recursos de comunicación disponibles, además del monitoreo supervisado por personal existente de seguridad se puede lograr el incremento de la seguridad mediante la aplicación de un sistema electrónico.

1.4 Alcance del trabajo

En el presente trabajo se describe el diseño de un sistema de monitoreo centralizado de alarmas.

En la figura 1.2 se muestra el diagrama de la solución propuesta, donde se monitorea múltiples unidades remotas, mediante la implementación de sistemas de alarmas independientes basados en la aplicación de microcontroladores PIC.

La exploración de estado de los sensores se plantea realizar aprovechando la capacidad de integrar convertidores análogo a digital de ciertos microcontroladores, aumentando la capacidad de un sistema convencional de alarmas midiendo el estado del cableado,

permitiendo aumentar la fiabilidad del sistema ante posibles actos de sabotaje como corte del cableado, o para identificar fallos del cableado en sí.

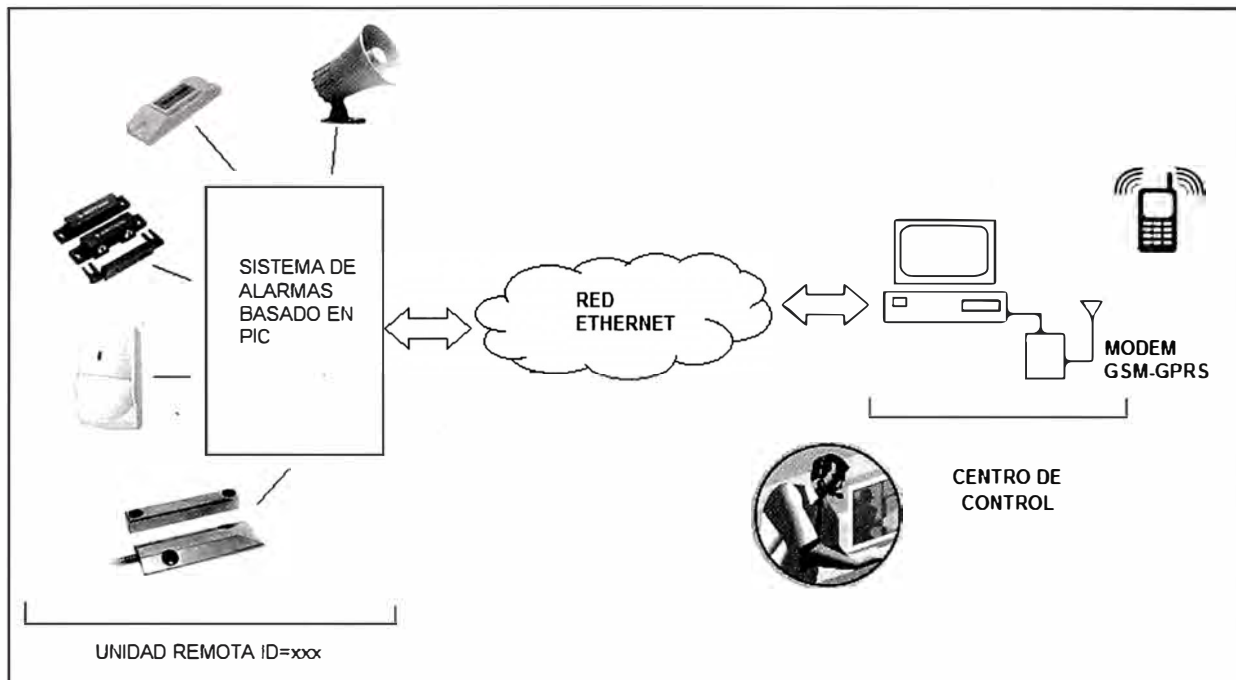


Figura 1.2 Esquema de la solución propuesta (Fuente: Diseño propio)

Se plantea el monitoreo de los sistemas de alarma a través de un computador personal ubicado en el centro de control conectado a la red Ethernet, para ello se requiere que cada sub-sistema basado en microcontrolador se conecte a la red Ethernet; para lograr la conectividad con la red se plantea la utilización de un módulo Ethernet Shield que permita la comunicación con el microcontrolador a través del puerto serial integrado.

Para asegurar que un evento de alarma generado por el sistema siempre sea reconocido por el operador, todo evento necesita de la confirmación por parte del operador de que se ha recibido la señal; en caso contrario luego de un lapso de tiempo se debe enviar automáticamente vía mensajes de texto las alertas de alarma a los números telefónicos pre programados.

1.5 Síntesis del Trabajo

El presente informe consta de dos partes principales: En la primera parte (Marco Teórico) se explica la tecnología involucrada en la solución planteada. Se describen sus partes y sus respectivas funciones. Sensores de alarma, microcontroladores PIC, módulos de comunicación Ethernet Shield – GPRS/GSM. En la segunda parte (Solución del Problema) se explica cómo se realizó el diseño, los criterios tomados del marco teórico, los recursos de hardware y software. Finalmente se muestra los costos y tiempo de implementación.

CAPITULO II MARCO TEORICO

2.1 Microcontroladores PIC

Los microcontroladores PIC son dispositivos programables, que contiene todos los componentes necesarios para un funcionamiento autónomo, constituyéndose en un pequeño ordenador que facilita el desarrollo de soluciones específicas. Sus aplicaciones se pueden dividir en dos grandes campos principalmente, el procesamiento de señales y las comunicaciones entre sistemas.

Principalmente el microcontrolador está concebido para aplicaciones donde el número de tareas sea relativamente pequeño, permitiendo reducir significativamente el coste de la solución. En estas aplicaciones el microprocesador ejecuta un programa almacenado permanentemente en la memoria, interactúa con el medio exterior a través de los puertos de entrada o salida del que dispone.

2.2 Lenguaje de Programación C

Una herramienta para el desarrollo de código para el microcontrolador es el compilador, en este caso se eligió el compilador C de CCS, que ha sido desarrollado específicamente para microcontroladores y es una adaptación del lenguaje C, lenguaje de alto nivel que permite facilitar la creación de programas medianamente complejos con muy pocas líneas de código, utilizando librerías estándar de C, con la posibilidad de incluir librerías especializadas, o definir librerías personalizadas; todo aunado a la potencia del lenguaje C constituye una herramienta de prestaciones invaluable.

En contraste con el lenguaje Assembler, que es un lenguaje nemónico más cercano al lenguaje de máquina; C tiene una sintaxis más familiar, con la ventaja que supone utilizar la programación estructurada, que si bien ocupa mayor espacio en la memoria, justifica su aplicación con los resultados que se obtienen. La utilización de un lenguaje de bajo nivel muestra sus ventajas en aplicaciones medianamente complejas

2.3 Software de Simulación

Una vez que tenemos el código de programa compilado, antes de cualquier implementación física podemos realizar la simulación a través de software especializado, se eligió el software PROTEUS VSM de LABCENTER ELECTRONICS (www.labcenter.co.uk), que ofrece la posibilidad de simular código en un entorno de

diseño electrónico, por su versatilidad y potencia.

2.4 El módulo integrado de comunicación serial síncrona MSSP

El módulo MSSP (Master Synchronous Serial Port) es una interface serial, útil para comunicaciones con otros periféricos u otros microcontroladores. Los dispositivos periféricos pueden ser memorias seriales EEPROMs, registros de desplazamiento, displays, convertidores A/D, etc. El módulo MSSP puede operar en uno de los dos modos siguientes:

- SPI (Serial Peripheral Interface).
- I²C (Inter-Integrated Circuit).

2.4.1 Comunicación serial SPI

La comunicación serial SPI permite la transmisión y recepción simultánea de datos entre dos dispositivos. Dependiendo de la polaridad de la data y la fase del reloj existen 4 modos de configuración. Para lograr la comunicación SPI se requieren básicamente 3 pines.

- SDO (Serial Data Salida)
- SDI (Serial Data Entrada)
- SCK (Reloj Serial)

La transmisión y recepción dependen de la señal SCK, según el modo este se puede realizar una vez finalizado el flanco de subida o de bajada. La comunicación prácticamente se realiza aplicando el desplazamiento de data en registros, en nuestro caso se necesitan 8 ciclos de reloj para completar el envío de un byte. El modo SPI es aplicado para comunicar el PIC con el módulo Ethernet Shield W5100.

2.5 Protocolo Ethernet

Ethernet refiere a la familia de Redes de Área Local LAN (Local Area Network) formado principalmente por tres categorías principales:

- 10 Mbps Ethernet especificado en la norma IEEE 802.3, donde se especifican las redes LAN que operan a 10Mbps sobre cable coaxial.
- 100 Mbps Ethernet que también se le conoce como "FAST ETHERNET", está especificado para redes que operan a velocidades de 100 Mbps sobre par trenzado.
- 1000 Mbps Ethernet o Gigabit Ethernet que está especificada para redes LAN que operan a 1000 Mbps (1Gbps) sobre par trenzado o Fibra óptica.

La evolución de Ethernet está íntimamente ligada con los avances logrados en la tecnología de las comunicaciones, que permitieron construir la red de redes llamada Internet. Estandarización que facilitó la interconexión de muchos equipos que permiten compartir la información de manera fiable a escala cada vez mayor.

2.5.1 Códigos

En Ethernet la transmisión se realiza de manera asincrónica, la señal que sirve de sincronismo se encuentra embebido a los datos mediante el uso de códigos. Estos códigos fueron evolucionando para lograr altas velocidades, cada quien más complejo y por lo tanto de mayor costo, se destacan: Manchester, 8B/6T, PAM5x5, NRZ, 8B/10B.

2.5.2 La trama Ethernet

La trama Ethernet se estructura de la siguiente manera:

Tabla 2.1 Trama Ethernet

PREAMBULO	DIRECCION DESTINO	DIRECCION FUENTE	TIPO/ LONGITUD	DATOS	CRC
8 bytes	6 bytes	6 bytes	2 bytes	46-1500 bytes	4 bytes

La tabla 2.1 muestra la trama para el protocolo Ethernet / 802.3. Donde el preámbulo está formado por una secuencia 10101010 que se repite 7 veces que finaliza con un delimitador 10101011, se especifican las direcciones destino y fuente, la longitud de la data, los datos y una secuencia de comprobación de errores.

2.6 Ethernet Shield W5100

La tarjeta W5100 es una interface Ethernet, diseñado para aplicaciones embebidas con fácil integración, estabilidad, performance, costo asequible. Facilita la implementación de conectividad Internet sin Sistema Operativo.

El W5100 incluye la pila TCP/IP, dirección de hardware MAC. El hardware TCP/IP soporta: TCP, UDP, IPv4, ICMP, ARP, IGMP y PPPoE que han sido probados por muchos años. Contiene un buffer interno de 16Kbytes para transmisión de data. No requiere consideración sobre manejo del controlador Ethernet, en cambio requiere conocimientos básicos de programación de Sockets.

2.6.1 Aplicaciones

El W5100 puede aplicarse a diferentes soluciones embebidas, que pueden incluir:

- Dispositivos domésticos de Red: Set-Top Boxes, Adaptadores Digitales Multimedia, PVRs.
- Interfaces Serial a Ethernet: Controles de Acceso, Matriz de LEDs, Relay AP inalámbrico, etc.
- Interfaces Paralelo a Ethernet: POS / Mini Impresoras, fotocopadoras.
- Interfaces USB a Ethernet: Dispositivos de Almacenamiento, Impresoras en red.
- Interfaces GPIO a Ethernet: Sensores domésticos de red.
- Sistemas de Seguridad: DVRs, Cámaras IP, Alarmas.
- Automatización de fabricación y construcción.
- Monitoreo de equipamiento médico.

- Servidores Embebidos.

2.6.2 Especificaciones Técnicas

- Protocolos TCP/IP soportados por hardware: TCP, UDP, ICMP, IPv4, ARP, IGMP, PPPoE, Ethernet.
- Ethernet PHY 10BaseT/ 100Base TX embebido.
- Soporta auto negociación Full duplex y Semi duplex.
- Soporte Auto MDI/ MDIX.
- Soporte de conexión ADSL.
- Soporta 04 sockets independientes simultáneos.
- No soporta fragmentación IP.
- Buffer Interno 16Kbytes para Transmisión y Recepción.
- Tecnología CMOS 0.18µm
- Voltaje de operación 3.3V con tolerancia a señales de 5V.
- Encapsulado LQFP 80 pines.
- Encapsulado libre de plomo.
- Soporta Interface Serial de Periféricos (SPI MODO 0,3).
- Indicadores LED multifunción (Tx, Rx, Full/Half dúplex, Colisión, Enlace, Velocidad).

2.7 Sistema global para las comunicaciones móviles GSM.

Las comunicaciones móviles inicialmente concebidas para la comunicación de voz a través de los teléfonos han evolucionado gracias al avance tecnológico, permitiendo la transmisión de data con tecnología digital, logrando de esta manera abrir un abanico de infinitas posibilidades de aplicaciones. GSM o Sistema Global para las telecomunicaciones móviles es un sistema estándar definido ante la necesidad de mejorar la eficiencia espectral, asegurar el *roaming* internacional, mejorar la calidad de voz transmitido y la compatibilidad con la red internacional ISDN, que antes se desarrolló en forma separada en diferentes países.

GSM aparece en 1982 como esfuerzo de estandarización promovida por la Conferencia Europea de Administraciones de Correos y Telecomunicaciones (CEPT), se crea para tal fin el Groupe Special Mobile que da origen a las siglas GSM, y desarrollan el estándar que finalizan en 1990.

Su arquitectura se basa en la división del espectro disponible, asignando a cada compañía cierto ancho de banda. Permite el envío de mensajes de texto SMS. GSM se implantó en Europa y en otros países del resto del mundo. TDMA y CDMA en EEUU, mientras que PDC en Japón. Estas tecnologías son consideradas de segunda

generación.

2.7.1 Mensajes de texto SMS

SMS es un servicio que permite el envío y recepción de mensajes de texto a través de teléfonos móviles (o dispositivos GSM conectados a un procesador). El SMS fue desarrollado como parte del estándar GSM fase 1 en 1992.

La longitud de los mensajes de texto está limitada a 160 caracteres, pudiendo ser palabras, números o alguna combinación alfanumérica. Los mensajes se transmiten a través de un centro de SMS, cada red de telefonía móvil que utiliza SMS tiene sus respectivos centros de mensajería.

La transmisión de los mensajes de texto puede realizarse simultáneamente a la comunicación de voz o datos, debido a que los mensajes de texto viajan sobre un canal independiente, dedicado a la señalización.

2.7.2 HSCSD Datos por Conmutación de Circuitos de Alta Velocidad

El HSCSD (High Speed Circuit Switched Data) especificado en la fase 2+ del estándar GSM, homologada por el ETSI(European Standard Institute) en febrero de 1997, es una plataforma multi-canal para la transmisión de datos a alta velocidad.

Mediante circuitos conmutados GSM, permitió la evolución hacia sistemas con mejor eficiencia, incluso mayor que las conexiones dedicadas. Puede alcanzar velocidades de hasta 57.6 kbit/s o superior, cuando se combina la aplicación de compresores y filtros (combinando canales de 9.6 kbit/s o 14.4 kbit/s).

Si bien las velocidades alcanzadas por el HSCSD representan una ventaja importante, aunado a la mínima inversión en su implementación (modificación de software, mínimos cambios en hardware), pueden representar un mayor costo para el usuario final, debido a la utilización de más de un canal de comunicaciones.

2.7.3 GPRS (General Packet Radio System)

El cambio que permitiría a la tecnología GSM dar un paso fundamental para su desarrollo, sin duda fue el establecimiento de un nuevo estándar, que implementaba la transmisión de paquetes vía radio, denominándose GPRS(General Packet Radio System), conocido también como GSM-IP, que permite la integración de los protocolos de comunicación TCP/IP a la red móvil GSM existente.

La tecnología GPRS se basa en la conmutación de paquetes realizando la transmisión sobre la red GSM, que a su vez se basa en la tecnología de conmutación de circuitos. La característica que hace la diferencia a la nueva tecnología GPRS, es que solo utiliza la red cuando existen datos por enviar. Lográndose alcanzar velocidades del orden de 115kbit/s, que lo hacen ideal para la comunicación por internet, como email, transferencia de archivos, navegación web. Otra ventaja es que los costos del servicio dependen de la

cantidad de data transmitida.

2.7.4 Modem GSM/GPRS

El modem GSM es un dispositivo que se conecta a la red GSM para el envío y recepción de data. Para el propósito de nuestro estudio nos enfocamos en el modem GSM/GPRS modelo FASTRACK M1306B de la marca Wavecom, que se controla mediante un puerto serial RS232, pudiendo ser controlado por un ordenador o un sistema basado en microcontrolador.



Figura 2.1 Modem GSM/GPRS Fastrack M1306B (Fuente: Wavecom)

a.- Características

En la tabla siguiente se resume las principales características del modem.

Tabla 2.2 Características del Modem Fastrack.

Características	GSM	DCS
Open AT	Compatibilidad comandos Open AT Operación Autónoma.	
Standard	900 MHz Cumple E-GSM Pot. de Salida: Clase4(2W) Compatibilidad ETSI GSM fase 2 + SMS	1800 MHz Pot. Salida: Clase1 (1W) Compatibilidad ETSI GSM Fase 2 + SMS
GPRS	Clase 10 Soporta PBCCH Esquema codificación: CS1 a CS4. Compatibilidad SMG31 Pila TCP/IP embebida (opcional)	
Interfaces	RS232 (V.24/V.28) Interface serial. • Velocidades (bits/s): 300,600,1200,2400,4800,9600,	

	19200,38400,57600,115200.
SMS	Texto PDU Punto a Punto (MT/MO) Transmisión celular.
Data	Circuito Asíncrono de Datos. Modos Transparente y No Transparente. Hasta 14400 bits/s. Corrección de errores MNP Clase 2. Compresión de data V42.
Audio	Cancelación de eco. Reducción de ruido. Telefonía. Llamadas de Emergencia. Operación (FR/EFR/HR) Full Rate, Enhanced Full Rate, Half Rate. Función Dual Tonos en multifrecuencia DTMF.

2.8 Arquitectura Cliente Servidor (Sockets)

Es la tecnología mediante el cual se tiene acceso transparente a las aplicaciones, datos, servicios o cualquier otro recurso del grupo de trabajo o red, en un sentido más amplio los componentes cliente y servidor estarán situados en más de un computador, todos ellos interconectados por una red de comunicaciones. Soportan un medio ambiente distribuido.

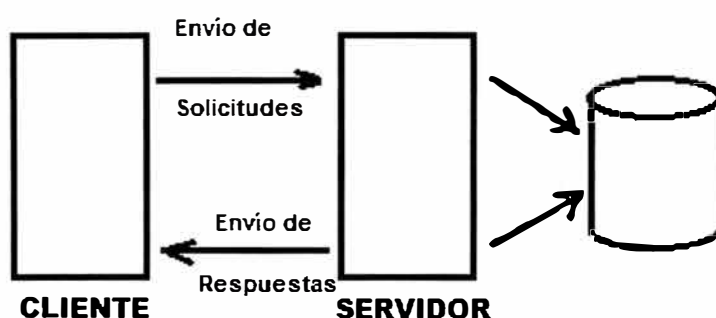


Figura 2.2 Modelo Cliente – Servidor (Fuente: diseño propio)

Para establecer la comunicación entre 2 puntos remotos en este modelo, se requiere que el servidor sea inicializado antes y se encuentre en función escucha, para que cuando un cliente necesite comunicarse envíe la petición de comunicación al servidor y este establezca el intercambio de información.

En la práctica los elementos que permiten la comunicación a nivel de software se denominan sockets, funcionan como punto final de las comunicaciones y constituyen un mecanismo para el intercambio de paquetes de datos.

CAPITULO III DISEÑO DEL SISTEMA

3.1 Componentes del Sistema

En las figuras 3.1 y 3.2 se muestran los componentes modulares del sistema propuesto, tanto la unidad remota como el centro de control.

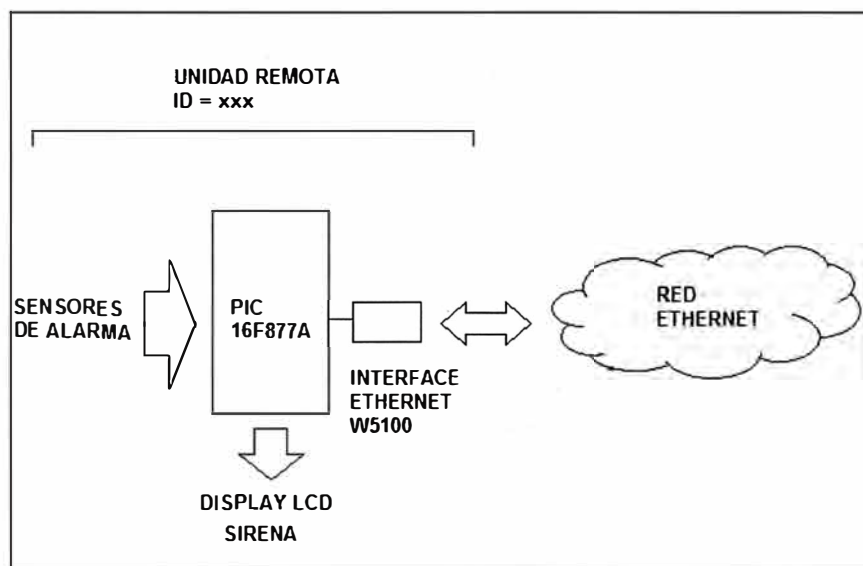


Figura 3.1 Esquema de la unidad remota (Fuente: Diseño propio)

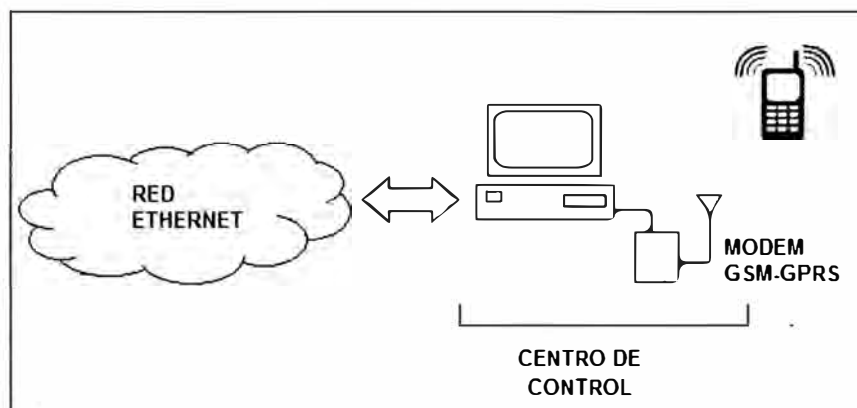


Figura 3.2 Esquema del centro de control (Fuente: Diseño propio)

Donde la unidad remota representa a uno de los sub-sistemas ubicados en un área independiente que se desea supervisar de forma electrónica. En el lado del Centro de

Control se tiene un ordenador donde se realiza la ejecución del programa de monitoreo de manera ininterrumpida, a través del software se identifica los eventos transmitidos desde las unidades remotas mediante un identificador unívoco ID y según la naturaleza del mismo puede generar una alarma visual y audible que requerirá de que se produzca un evento de reconocimiento por parte del operador del sistema, asegurando que se tomen las medidas del caso; en caso de superarse el tiempo pre-programado como la espera máxima del reconocimiento, el programa enviará automáticamente un mensaje de texto a los números telefónicos programados, a través de la red celular de comunicaciones.

3.2 La Unidad Remota

La unidad remota está constituida básicamente por una unidad de control gobernada por un microcontrolador, dicho microcontrolador es el encargado de realizar las acciones de control tanto de los dispositivos de entrada como los de salida.

El microcontrolador a utilizar en el presente trabajo es el PIC 16F877 de Microchip, las características más relevantes para su elección fueron: Integra un convertidor análogo a digital, posee interfaces de comunicación serial RS232 – SPI y toda la flexibilidad posible a través del software de programación en lenguaje C.

3.2.1 Fuente de Alimentación

Para alimentar la unidad de control se plantea la fuente de alimentación cuyo circuito se muestra en la figura 3.3

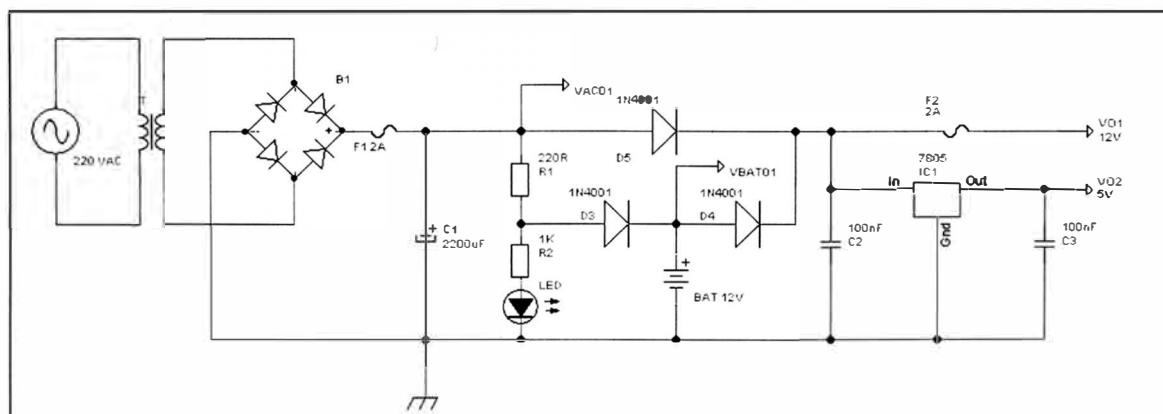


Figura 3.3 Fuente de Alimentación. (Fuente: Captura Tynicad)

La fuente de alimentación propuesta permite entregar 2 tipos de salidas:

- 05VDC permite la alimentación del microcontrolador 16F877A, del display LCD y de la interface Ethernet.
- 12VDC permite la alimentación de los dispositivos de entrada/salida que requieran alimentación, como por ejemplo detectores de movimiento, procesadores de vibración, procesadores de aniego, sirenas.

Para asegurar la correcta operación de nuestra unidad de control, periódicamente se realiza una lectura de las tensiones VAC01 (voltaje de corriente continua asociado a la presencia o ausencia de energía AC) y VBAT01 que representa la tensión en la batería. De tal manera que si estas lecturas caen por debajo de un voltaje de referencia, el sistema generará un evento que es reportado al sistema de monitoreo. Dichos eventos conocidos como “Batería Baja” y “Pérdida AC” producen un aviso sobre la posible inminencia de inoperatividad de la unidad remota.

3.2.2 Unidad de control

La unidad de control se encarga de realizar las principales funciones del sistema de alarma, entre ellas: Exploración del estado de las zonas de la alarma Z01 – Z06, Medición de tensión alterna, Medición de la tensión en la batería, Enviar al módulo LCD la data necesaria para mostrar mensajes alfanuméricos, Comunicación serial SPI con el módulo Ethernet para la interacción remota con el software de monitoreo, a la vez controla la salida audible.

El diagrama de la unidad de control se muestra en la figura 3.5, el principal componente es el microcontrolador PIC 16F877A donde irán conectados los dispositivos de entrada y salida según los requerimientos de funcionamiento, luego se realiza la programación donde se establece las acciones de control para todos los eventos que generen el sistema.

a.- Zonas de Alarma

El sistema admite la conexión de 06 zonas de alarma, donde se pueden conectar sensores ya sean normalmente cerrado o normalmente abierto. Para realizar la lectura del estado de cada zona se aprovecha la disponibilidad del convertidor Análogo a Digital ADC integrado al microcontrolador. Para monitorear el cableado del sensor se utilizará resistores de fin de línea, que permite agregar un estado adicional para reportar algún sabotaje o deterioro del cableado.

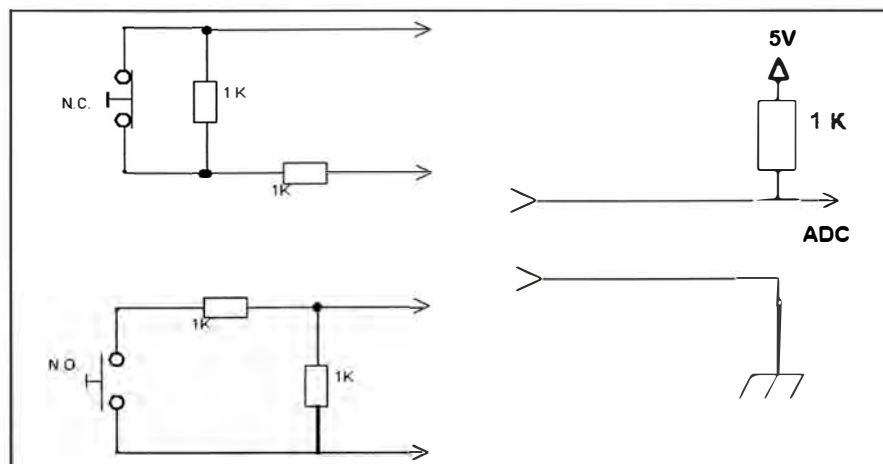


Figura 3.4 Zonas de alarma (Fuente: Diseño propio)

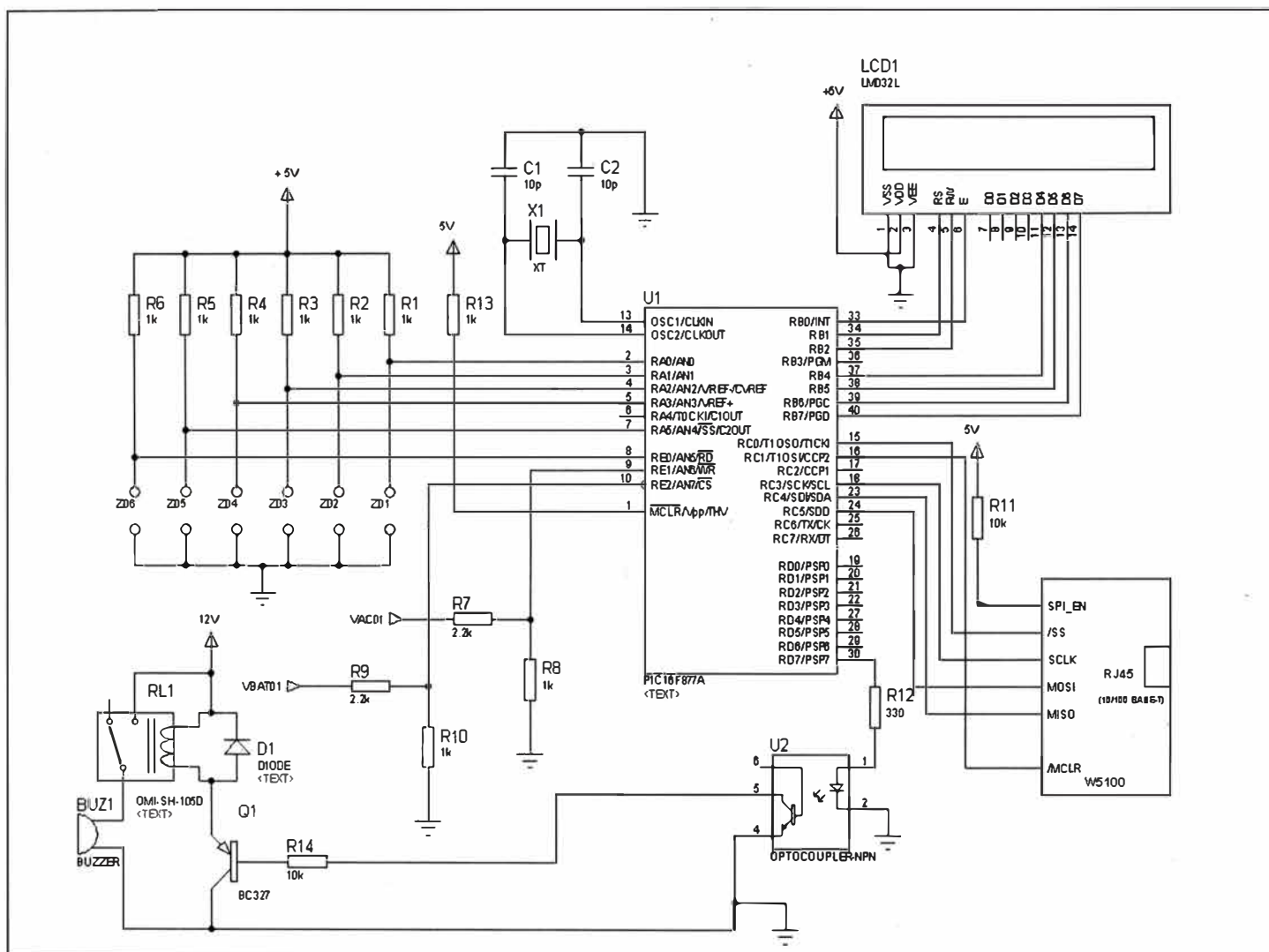


Figura 3.5 Diagrama de la Unidad de Control (Fuente: Captura Proteus)

Para esta disposición de conexión tendremos los siguientes estados y las respectivas mediciones del ADC, mostrados en la tabla 3.1 Estos valores ideales necesitan una tolerancia debido al error en los valores de las resistencias y las impedancias agregadas por el cableado. En aplicaciones reales estos estados adicionales de las zonas de alarma permiten alertar sabotajes tales como el corte de cableado, donde la definición de la tolerancia se establece por código.

Tabla 3.1 Estados de las zonas de alarma

Estado	Normalmente Cerrado	Normalmente Abierto
Cableado cortado (abierto)	5.00 V	5.00 V
Cableado corto circuito	0.00 V	0.00 V
Sensor impedancia infinita	3.33 V	2.50 V
Sensor impedancia cero	2.50 V	1.67 V

b.- Medición de tensión de la Fuente y de la Batería

Para monitorear constantemente la presencia de AC y la tensión de la batería se utilizarán 02 entradas del convertidor análogo – digital a través de un divisor de tensión, debido a que la entrada del convertidor análogo digital tiene su límite superior en 5V. Con el circuito de la figura 3.6 la tensión medida por el ADC está en relación 1/3.3 de la tensión medida, pudiendo de esta manera realizar medidas desde cero voltios hasta 16.5 voltios.

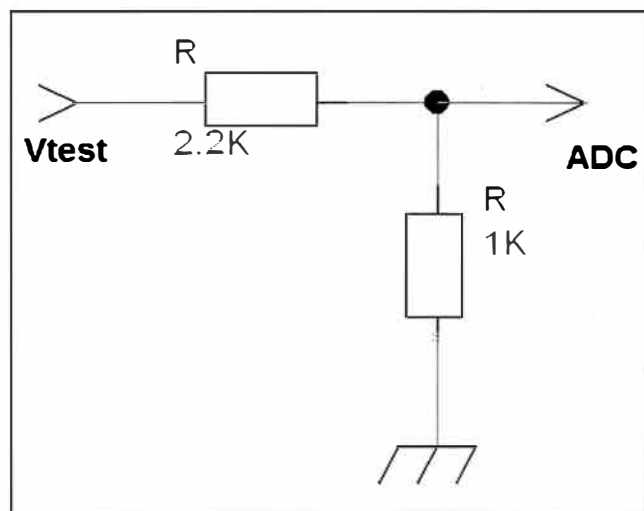


Figura 3.6 Circuito de medición de tensiones (Fuente: Diseño propio)

c.- Conexión del Módulo LCD

Para mostrar mensajes alfanuméricos se conecta al puerto B del microcontrolador un módulo display LCD 32 caracteres LM032L, para utilizar la mínima cantidad de pines del microcontrolador se aplica la forma multiplexada, donde la data se envía a través de 04 pines, adicionalmente se necesita 03 pines de control.

El módulo LCD viene integrado a un propio sistema microprocesado, de tal manera que el microcontrolador PIC solo necesita enviar comandos para mostrar cadenas de caracteres, mas todo el tiempo su funcionamiento es autónomo. En situaciones donde se requiera optimizar el uso de los puertos del microcontrolador, se puede aprovechar los puertos que envían data al módulo LCD para otros fines.

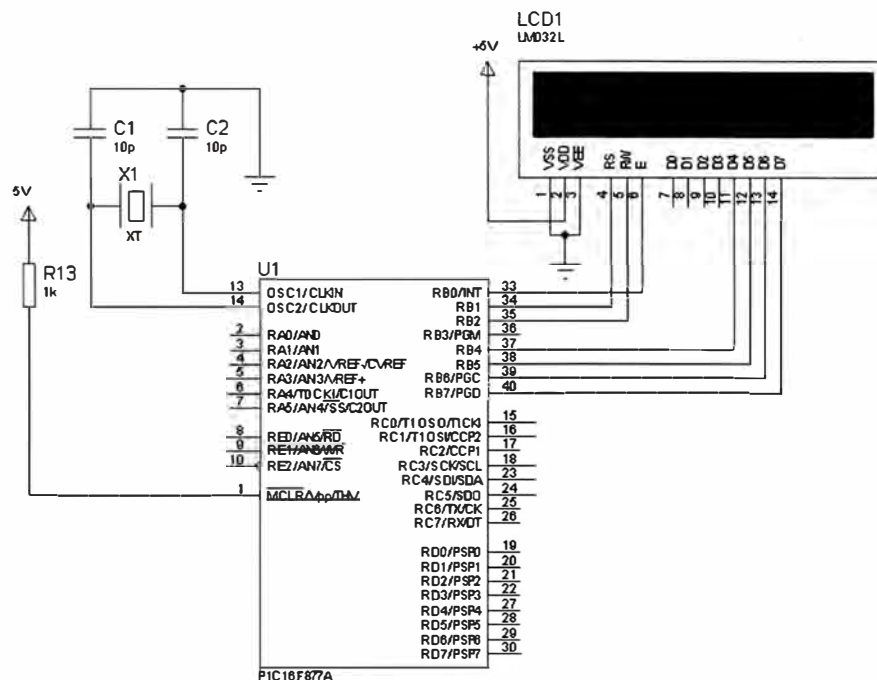


Figura 3.7 Conexión del módulo LCD (Fuente: Captura Proteus)

d.- Manejo de dispositivos de Salida

Para manejar dispositivos de salida, la unidad de control puede activar un relé que permite energizar dispositivos tales como una sirena electrónica. Para evitar cualquier sobrecarga se utiliza un acoplador óptico, cuya carga es similar a un LED. El acoplador óptico controla la conducción del transistor Q1 que energiza la bobina del relé. El circuito se muestra en la figura 3.8.

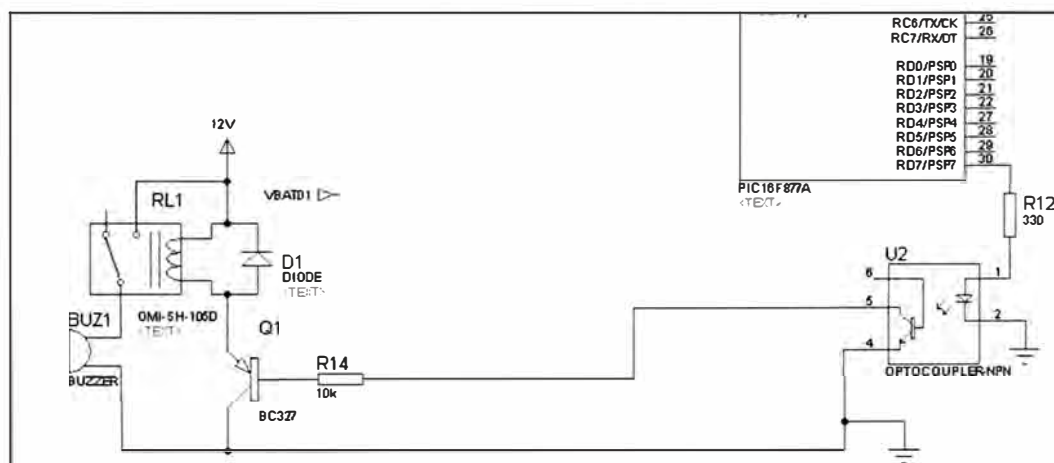


Figura 3.8 Circuito de salida 12V (Fuente: Captura Proteus)

e.- Conexión del Módulo Ethernet Shield

Para la conexión del módulo Ethernet con el microcontrolador se necesitan 03 pines definidos por el fabricante para la comunicación serial SPI (SCLK pin18, MOSI Master Output Slave Input pin24, MISO Master Input Slave Output pin23), además 01 pin I/O para seleccionar el dispositivo (/SS), esto es útil cuando se controlan más de un dispositivo mediante SPI, también 01 pin I/O para controlar el RESET del módulo Ethernet.

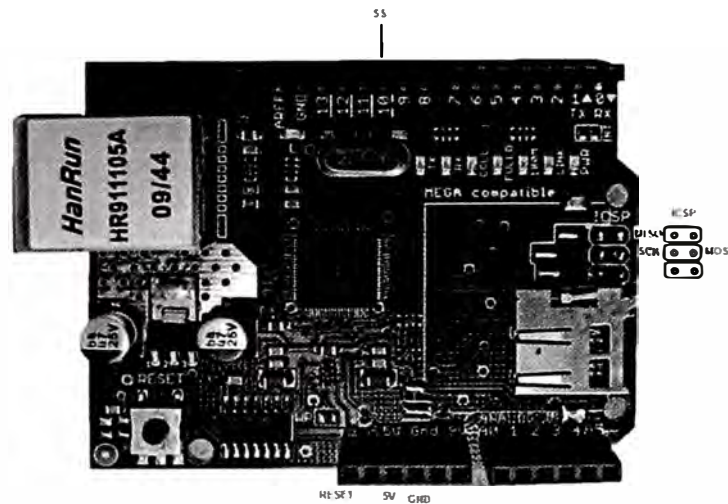


Figura 3.9 Pinout W5100 (Fuente: Wiznet)

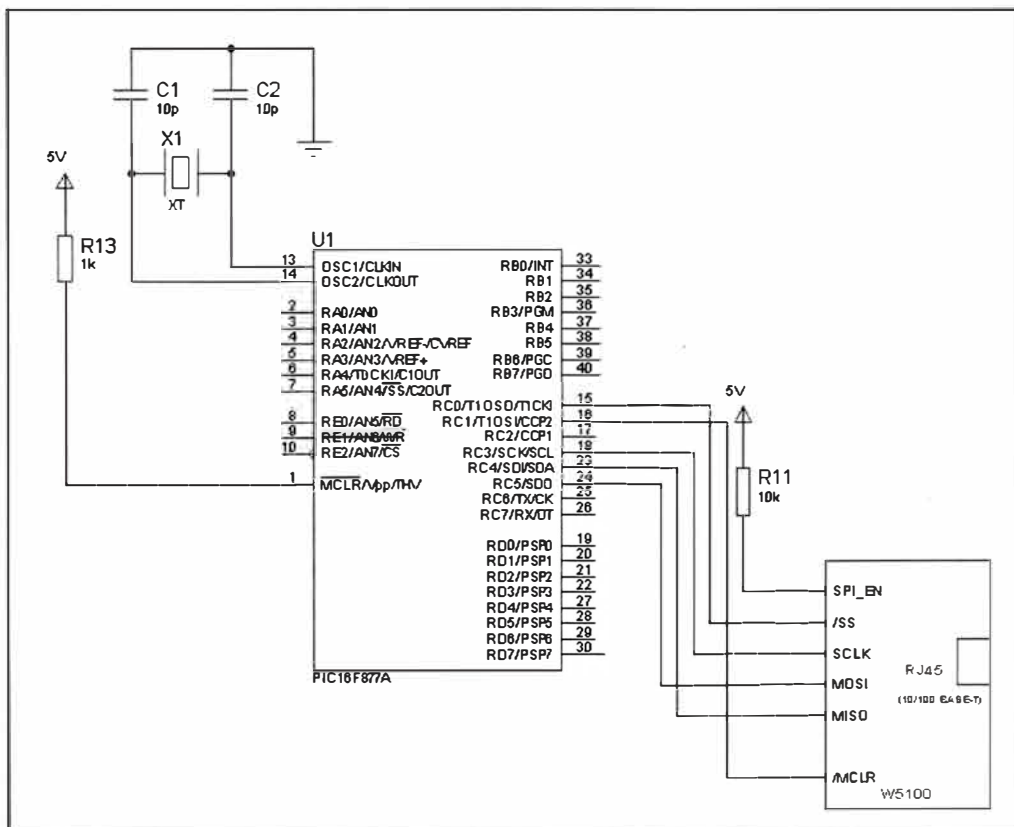


Figura 3.10 Conexión Ethernet Shield W5100 (Fuente: Captura Proteus)

f.- Software de Control para la unidad remota.

Una vez finalizado el diseño a nivel de hardware para nuestro sistema, se procede a desarrollar el componente software, que permitirá al microcontrolador realizar las acciones de control necesarias para el funcionamiento del sistema.

El avance del desarrollo de software se detallará progresivamente, para finalmente integrar todo a nivel funcional del sistema.

f.1.- Mostrar mensajes alfanuméricos mediante módulo LCD

Para el manejo del módulo LCD el fabricante facilita una librería escrito en c, donde se define entre otros la función `lcd_putc` que permite enviar cadena de caracteres al módulo LCD, los mismos que se mostrarán en el display alfanumérico. Para darle un formato a la cadena de caracteres y se puedan incluir valores numéricos vamos a utilizar el comando estándar de salida `printf`. A continuación se muestra el programa de pruebas del funcionamiento del módulo LCD y la captura correspondiente de la simulación realizada en Proteus.

```
// CODIGO FUENTE EN C DEL PROGRAMA DE PRUEBAS DEL
// MODULO LCD
#include <16F877A.h>
#fuses HS,NOWDT,NOPROTECT,NOLVP
// Definimos la frecuencia del reloj en 20MHz
#use delay(clock=20000000)
// Incluimos la librería para manejo de módulo LCD
// a través del puerto B.
#include <lcdh.c>
void main(){
    // Inicializamos el módulo LCD con su respectivo retardo
    // Sugerido por el fabricante
    lcd_init();
    delay_ms(50);
    // Enviamos cadena de texto utilizando el procedimiento
    //lcd_putc definido en la librería lcdh.c, conjuntamente
    //con el comando de salida printf
    //Se utilizan caracteres de control
    // \n cambio de línea y \f borrado de pantalla
    printf(lcd_putc,"Iniciando test LCD ...\n");
    delay_ms(2000);
    printf(lcd_putc,"UNI FIEE 2013\n");
    delay_ms(4000);
    printf(lcd_putc,"\fFin.");
    delay_ms(2000);
```

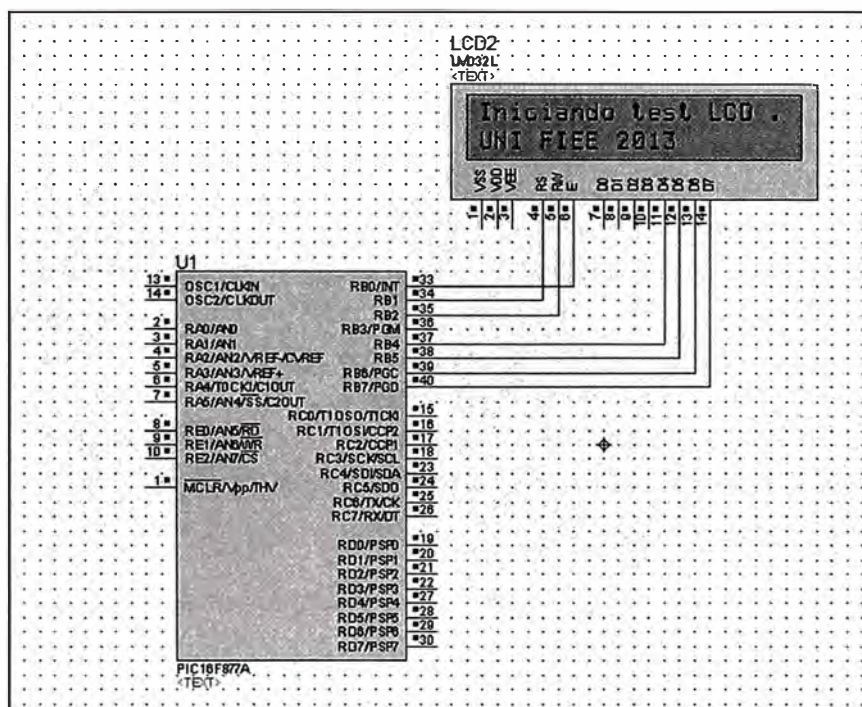


Figura 3.11 Simulación Módulo LCD (Fuente: Captura Proteus)

f.2 Adquisición de data aplicando el convertidor ADC

Se utiliza el módulo convertidor analógico digital integrado al microcontrolador, para realizar mediciones de tensión, tanto para medir el estado de las zonas de alarma como para medir periódicamente la energía de la batería y de la fuente. El ADC tiene 8 entradas, las que se distribuyen de la siguiente manera: 06 entradas corresponderán a las zonas de alarma, 01 entrada a la tensión de la batería de respaldo y 01 entrada a la tensión de la fuente de alimentación.

Se desarrolló un programa de pruebas para realizar la exploración de las zonas de alarma, que a través de un bucle iterativo realiza la lectura secuencial de la tensión en los terminales del convertidor análogo a digital. Luego se realiza la comparación de los valores obtenidos con los valores de la tabla 3.1 que permite concluir el estado de cada zona. En el programa de pruebas, se muestra el mensaje "LISTO" solo si todas las zonas se encuentran cerradas, caso contrario se muestran las zonas abiertas y el mensaje "NO LISTO". A continuación se muestra el programa desarrollado para las pruebas y las capturas de la simulación en Proteus.

```
#include <16F877.h>
#device adc=10;

#FUSES XT,NOWDT
#FUSES

#use delay(clock=20000000)
#include <lcd.h>

int exploracion(int canal){
```

```

int16 q;
float p;
int estado;
set_adc_channel(canal);
delay_us(20);
q=read_adc(); // Lectura canal
p=5.0*q/1024.0; // Conversión de tensión
estado=0;
if(p<0.5) estado=0; // Corto Circuito
if(p>4.5) estado=3; // Cable Abierto
if((p>2)&&(p<3)) estado=1; // Sensor Normal
if((p>3)&&(p<3.6)) estado=2; // Sensor Abierto
return(estado);

void main(){
  int j;
  int estado zona, estado listo;

  setup_adc_ports(ALL_ANALOG);
  setup_adc(ADC_CLOCK_INTERNAL); // Fuente de Reloj RC

  lcd_init();
  do{
    estado_listo=0;
    printf(lcd_putc, "\f");
    for(j=0; j<4; j++){
      estado_zona=exploracion(j);
      if(estado_zona!=1){
        estado_listo++;
        printf(lcd_putc, "Z%d ", j+1);
      }
    }
    if(estado_listo!=0) printf(lcd_putc, "\n... NO LISTO");
    else printf(lcd_putc, "\n... LISTO");
    delay_ms(1000);
  }while(1);
}

```

En esencia la exploración secuencial de tensiones realizado por el ADC, produce un valor de tensión para un determinado instante de tiempo, ese valor obtenido es un número de 10 bits, para obtener el valor de la tensión se necesita realizar una conversión de escala, multiplicando el valor obtenido por ADC por el valor de 5 (tensión de referencia 5V) y dividiendo el resultado entre 1024. Una vez obtenido el valor de la tensión solo se


```

spi_write(W5100_WRITE_CMD);
spi_write(addr >> 8); // msb
spi_write(addr);      // lsb
spi_write(data);      // Envía data al W5100
output_high(W5100_CS);delay_us(50);

// Func lee un byte en la direcc addr

int8 w5100_read_reg(int16 addr){
int8 retval;
output_low(W5100_CS);delay_us(50);
spi_write(W5100_READ_CMD);
spi_write(addr >> 8); // msb
spi_write(addr);      // lsb
spi_write(0x00);
retval = spi_read(); // Lee data desde W5100
output_high(W5100_CS);delay_us(50);
return(retval);

}

void main()

int8 data0, data1, data2, data3;

set_tris_a(0);
set_tris_d(0);
set_tris_c(16);
// Configuración del puerto C para trabajar el spi
setup_spi(SPI_MASTER | SPI_L_TO_H |SPI_XMIT_L_TO_H| SPI_CLK_DIV_16);

lcd_init();// Inicialización LCD
delay_ms(50);

lcd_putc("Iniciando ...");
output_high(W5100_CS);
delay_ms(1000);

lcd_putc("\fSelect W5100");
output_high(SS);
delay_ms(1000);

lcd_putc("\fReset W5100");
delay_ms(1000);
output_low(W5100_RESET);

```

```

delay_ms(500);
output_high(W5100_RESET);
delay_ms(50);

w5100_write_reg(W5100_GAR0_REG, 0xC0); //192 GATEWAY
w5100_write_reg(W5100_GAR1_REG, 0xA8); //168
w5100_write_reg(W5100_GAR2_REG, 0x01); //1
w5100_write_reg(W5100_GAR3_REG, 0x01); //1

w5100_write_reg(W5100_SUBR0_REG, 0xFF); //255 MASCARA DE SUBRED
w5100_write_reg(W5100_SUBR1_REG, 0xFF); //255
w5100_write_reg(W5100_SUBR2_REG, 0xFF); //255
w5100_write_reg(W5100_SUBR3_REG, 0x00); //0

w5100_write_reg(W5100_SHAR0_REG, 0x00); //00 DIRECCION MAC
w5100_write_reg(W5100_SHAR1_REG, 0x00); //00
w5100_write_reg(W5100_SHAR2_REG, 0xD7); //D7
w5100_write_reg(W5100_SHAR3_REG, 0xF3); //F3
w5100_write_reg(W5100_SHAR4_REG, 0xF6); //F6
w5100_write_reg(W5100_SHAR5_REG, 0xFA); //FA

w5100_write_reg(SIPR0, 0xC0); //192 IP DEL MODULO
w5100_write_reg(SIPR1, 0xA8); //168
w5100_write_reg(SIPR2, 0x01); //1
w5100_write_reg(SIPR3, 0x07); //7

data0 = w5100_read_reg(W5100_GAR0_REG);
data1 = w5100_read_reg(W5100_GAR1_REG);
data2 = w5100_read_reg(W5100_GAR2_REG);
data3 = w5100_read_reg(W5100_GAR3_REG);

lcd_putc("\fGateway:\n");
printf(lcd_putc, "%u.%u.%u.%u", data0, data1, data2, data3);
delay_ms(3500);

data0=w5100_read_reg(W5100_SUBR0_REG);
data1=w5100_read_reg(W5100_SUBR1_REG);
data2=w5100_read_reg(W5100_SUBR2_REG);
data3=w5100_read_reg(W5100_SUBR3_REG);

lcd_putc("\fMask:\n");
printf(lcd_putc, "%u.%u.%u.%u", data0, data1, data2, data3);
delay_ms(3500);

```

```

data0=w5100_read_reg(SIPR0);
data1=w5100_read_reg(SIPR1);
data2=w5100_read_reg(SIPR2);
data3=w5100_read_reg(SIPR3);
lcd_putc("\fIP:\n");
printf(lcd_putc,"%u.%u.%u.%u",data0,data1,data2,data3);
delay_ms(3500);

```

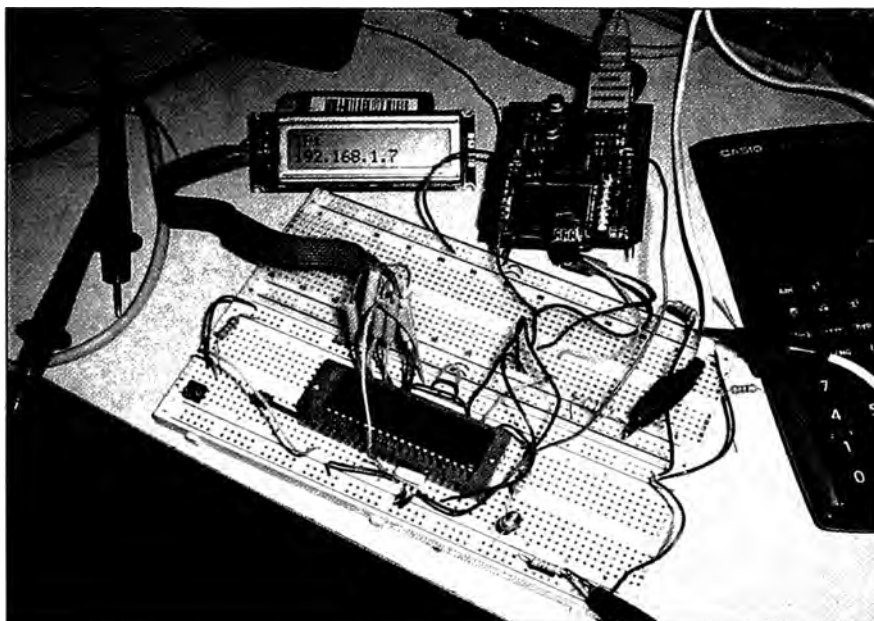


Figura 3.14 Fotografía de las pruebas de conectividad W5100 – PIC 16F877A

Luego de las pruebas correspondientes a la conectividad entre el microcontrolador y el módulo Ethernet Shield, se realizó la comprobación de la comunicación en red con un ordenador conectado al mismo switch de comunicaciones, en la figura 3.15 se muestra el resultado de ejecutar el comando ping desde la ventana de comandos del ordenador.

```

C:\windows\system32\cmd.exe
C:\Users\user>ping 192.168.1.7 -t
Haciendo ping a 192.168.1.7 con 32 bytes de datos:
Respuesta desde 192.168.1.7: bytes=32 tiempo=3ms TTL=128
Respuesta desde 192.168.1.7: bytes=32 tiempo=7ms TTL=128
Respuesta desde 192.168.1.7: bytes=32 tiempo=2ms TTL=128
Respuesta desde 192.168.1.7: bytes=32 tiempo=4ms TTL=128
Respuesta desde 192.168.1.7: bytes=32 tiempo=2ms TTL=128
Respuesta desde 192.168.1.7: bytes=32 tiempo=4ms TTL=128
Respuesta desde 192.168.1.7: bytes=32 tiempo=2ms TTL=128
Respuesta desde 192.168.1.7: bytes=32 tiempo=5ms TTL=128
Respuesta desde 192.168.1.7: bytes=32 tiempo=1ms TTL=128
Respuesta desde 192.168.1.7: bytes=32 tiempo=1ms TTL=128
Respuesta desde 192.168.1.7: bytes=32 tiempo=1ms TTL=128
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.

Estadísticas de ping para 192.168.1.7:
    Paquetes: enviados = 13, recibidos = 11, perdidos = 2
    (15% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 1ms, Máximo = 7ms, Media = 2ms
Control-C
^C
C:\Users\user>

```

Figura 3.15 Pruebas conectividad (Fuente: Captura Ventana Comandos)

f.4.- Comunicación a Estación Central de Monitoreo

Cada unidad remota reporta el estado correspondiente al software de monitoreo central y a la vez estará a la escucha de posibles comandos. La comunicación utiliza la tecnología de “sockets”, que permite la comunicación de dos ordenadores a través de la transferencia de un flujo de datos.

Por cada unidad remota se establecen 02 sockets, uno definido como cliente para enviar los reportes al centro de control y otro definido como servidor para recibir comandos.

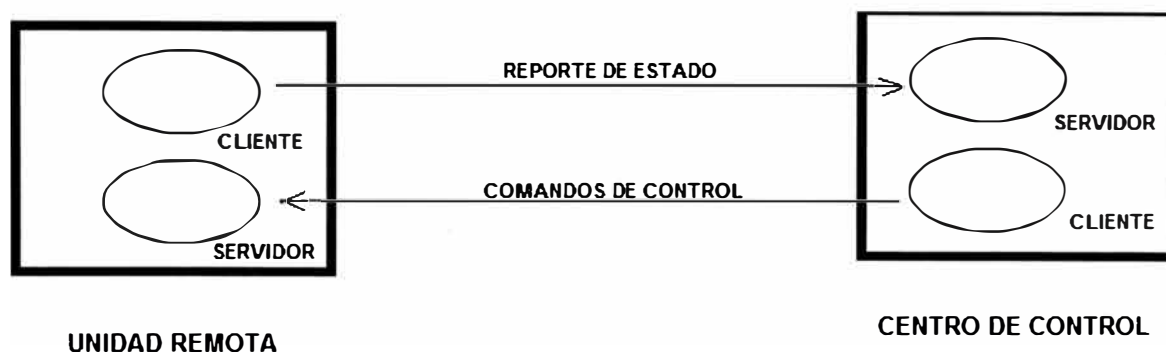


Figura 3.16 Forma de comunicación entre Sockets (Fuente: Diseño propio)

En la figura 3.16 se muestra la forma en la que se realizarán las comunicaciones entre cada unidad remota y el centro de control. Cuando la unidad remota necesite reportar un evento al centro de control, el socket cliente de la unidad remota solicitará al socket servidor del centro de control que se encuentra en estado de escucha(a la espera de comunicaciones) establecer la comunicación, luego el servidor valida la procedencia de la información y se inicia la transferencia de información que se almacena en el respectivo buffer del servidor. En el caso que el centro de control necesite enviar comandos a una unidad remota, se realizará el proceso en sentido inverso al caso anterior.

i.- Formato de señal de reporte

Se establece un formato para la data a transmitir desde cada unidad remota, en principio algún cambio de estado del sistema se transmitirá inmediatamente, en cambio sino sucedieran eventos de cambio en un tiempo predeterminado como 2 minutos por ejemplo, se transmitirá el estado del sistema como una forma de confirmar al centro de control que el monitoreo está activo.

El formato definido para la comunicación del estado de cada unidad remota se muestra en la tabla 3.2.

Tabla 3.2 estructura de reporte

ID UNIDAD REMOTA			Z01	Z02	Z03	Z04	Z05	Z06	BAT	AC
Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10

Cada cadena de caracteres que se transmitirá será del tamaño de 10 bytes, que explayamos a continuación:

Byte0,Byte1,Byte2	:	Identificación de la unidad remota (001 hasta 999)
Byte3	:	Estado de la zona 1 (0,1,2,3)
Byte4	:	Estado de la zona 2 (0,1,2,3)
Byte5	:	Estado de la zona 3 (0,1,2,3)
Byte6	:	Estado de la zona 4 (0,1,2,3)
Byte7	:	Estado de la zona 5 (0,1,2,3)
Byte8	:	Estado de la zona 6 (0,1,2,3)
Byte9	:	Estado Tensión de la Batería (0,1)
Byte10	:	Estado Tensión AC (0,1)

ii.- Formato de Comandos admitidos por el sistema

Se establece el formato para los comandos admitidos por la unidad remota, que en esencia son 02 comandos. El primero SOUT que según su valor activará o desactivará la sirena audible, además se define un parámetro que limita la duración de la sirena activada en un valor estándar de 4 minutos, siendo totalmente programable esta opción. El segundo valor LMSG permitirá mostrar en el display LCD mensajes alfanuméricos, por tal motivo debe tener un argumento tipo cadena de caracteres.

SOUT1	:	Activa la salida de sirena
SOUT0	:	Desactiva la salida de sirena
LMSG "xxxxxxxxxxxxxxxx"		Muestra mensaje alfanumérico en el display LCD.
START	:	Habilita e Inicia la comunicación
STOP	:	Deshabilita la comunicación

3.3 Software de Monitoreo

Uno de los principales componentes de nuestro sistema de monitoreo lo constituye el software de monitoreo, que se encargará de recibir los reportes de estado de las unidades remotas, registrar los eventos de alarma, enviar comandos a las unidades remotas, generar alertas audibles en caso de alarmas y enviar mensajes de texto a los teléfonos móviles pre-establecidos en el caso de que no se produzca el reconocimiento de un evento de alarma en un lapso de tiempo pre-determinado.

El desarrollo se realizó en el entorno de programación visual basic, que cuenta con herramientas especializadas en comunicación. Para la comunicación con las unidades remotas en forma bidireccional se utilizará la tecnología de "Sockets", el cual permite la comunicación entre 02 unidades remotas, cada una identificada con su respectiva IP y número de puerto; dicha forma de comunicación se realiza siguiendo la arquitectura

Cliente – Servidor. El sistema incorpora la aplicación de la tecnología GSM/ GPRS para enviar alertas a terminales móviles, para ello el software de monitoreo enviará los comandos AT necesarios al módulo “MODEM GSM GPRS” vía puerto serial, para que se envíen dichas alertas.

3.3.1 Archivos de Inicialización

El sistema requiere almacenar datos de estado y configuración de las unidades remotas, para ello utilizaremos archivos INI (Windows Initialization File) donde se registra los valores de las variables para cada unidad remota. La estructura del archivo consiste en etiquetas encerrado entre llaves [], que definen el nombre de una sección; cada variable correspondiente a una sección es identificada por una etiqueta llamada clave, que viene a ser el nombre de la variable correspondiente, es seguido por el carácter “=” y un valor correspondiente. El modelo del archivo de inicialización para una unidad remota sería por ejemplo:

Contenido del archivo UNIDAD134.ini

```
[UBICACION]
UBIC NOMBRE=AREA PRODUCCION
UBIC ID=134

[RED]
RED IP=192.168.1.45
RED PUERTO1=777
RED PUERTO2=888

[CONFIG_ZONAS]
CONFIG Z01=24HSilencioso
CONFIG Z02=Interior
CONFIG Z03=Interior
CONFIG Z04=Interior
CONFIG Z05=Interior
CONFIG Z06=24HAudible

[ZONIFICACION]
NOMBRE Z01=Pulsador Emergencia
NOMBRE Z02=Puerta Ingreso
NOMBRE Z03=Planta 1
NOMBRE Z04=Planta 2
NOMBRE Z05=Puerta Emergencia
NOMBRE_Z06=Tamper Sirena
```

El acceso a la data contenida en un archivo tipo ini se realizará a través de un comando predefinido en VB que se denomina GetPrivateProfileString. El siguiente código se desarrolló para definir las funciones LeerINI y EscribirINI que permiten manipular datos en archivos de inicialización, estas funciones tienen como argumentos el nombre del archivo,

nombre de la sección y nombre de la variable, devuelve o escribe el valor correspondiente:

```
Declare Function GetPrivateProfileInt Lib "kernel32" Alias "GetPrivateProfileIntA"
(ByVal lpApplicationName As String, ByVal lpKeyName As String, ByVal nDefault As
Long, ByVal lpFileName As String) As Long
```

```
Declare Function GetPrivateProfileString Lib "kernel32" Alias
"GetPrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As
Any, ByVal lpDefault As String, ByVal lpReturnedString As String, ByVal nSize As
Long, ByVal lpFileName As String) As Long
```

```
Declare Function WritePrivateProfileString Lib "kernel32" Alias
"WritePrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As
Any, ByVal lpString As Any, ByVal lpFileName As String) As Long
```

```
Public Sub EscribINI(Seccion As String, Clave As String, miVariable As String,
ArchINI As String)
    'Seccion= Seccion a escribir
    'Clave=Clave a escribir
    'miVariable=Valor a escribir
    'ArchINI=Archivo a leer
    'Escribo finalmente en el archivo INI
    WritePrivateProfileString Seccion, Clave, miVariable, ArchINI
End Sub
```

```
Public Function LeerIni(Seccion As String, Clave As String, ArchivoINI As String) As
String
    'Seccion=Seccion de la cual leer
    'Clave=Clave a leer
    'ArchivoINI=Archivo a leer
    Dim mvar As String
    mvar = Space(128)
    Dim R As Long
    R = GetPrivateProfileString(Seccion, Clave, "ERROR AL LEER INI", mvar, 128,
ArchivoINI)
    LeerIni = Left(mvar, Len(mvar))
End Function
```

3.3.2 Comunicación vía Sockets.

Para establecer comunicación entre las unidades remotas y el software de monitoreo aplicamos la tecnología de sockets, Visual Basic incluye una herramienta llamada Winsock, dicho componente contiene las propiedades, los métodos y eventos para establecer la comunicación entre equipos utilizando el protocolo de comunicaciones de red TCP/IP.

Para nuestra solución definiremos dos sockets, uno como Servidor y otro como Cliente.

En un proceso de comunicación Cliente – Servidor, el servidor debe encontrarse en estado de “escucha” y es el cliente quien inicia la solicitud de inicialización de

comunicación. En la figura 3.17 se muestra los sockets definidos con sus respectivas propiedades principales.

Sockets definidos:

Sockets : Winsock_Cliente y Winsock_Servidor.

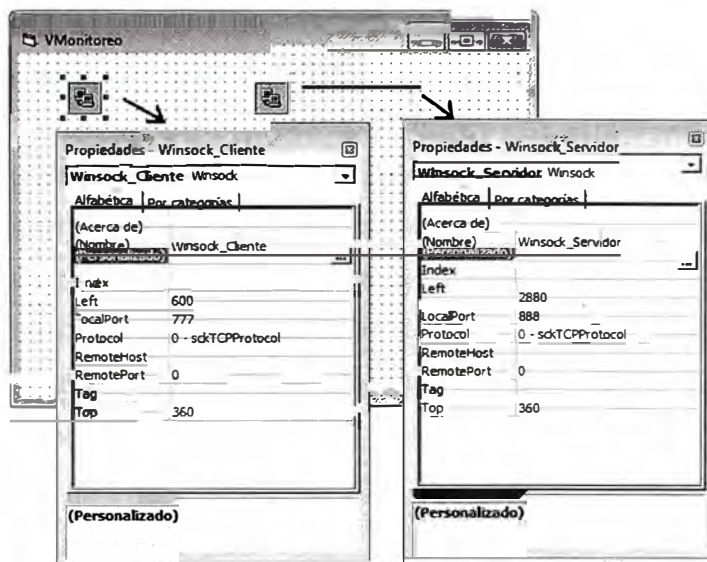


Figura 3.17 Propiedades de los Sockets Servidor – Cliente (Fuente: Captura VB)

a.- Recepción de Data Entrante

En la figura 3.18 se muestra el diagrama de flujo correspondiente al tratamiento de data entrante.

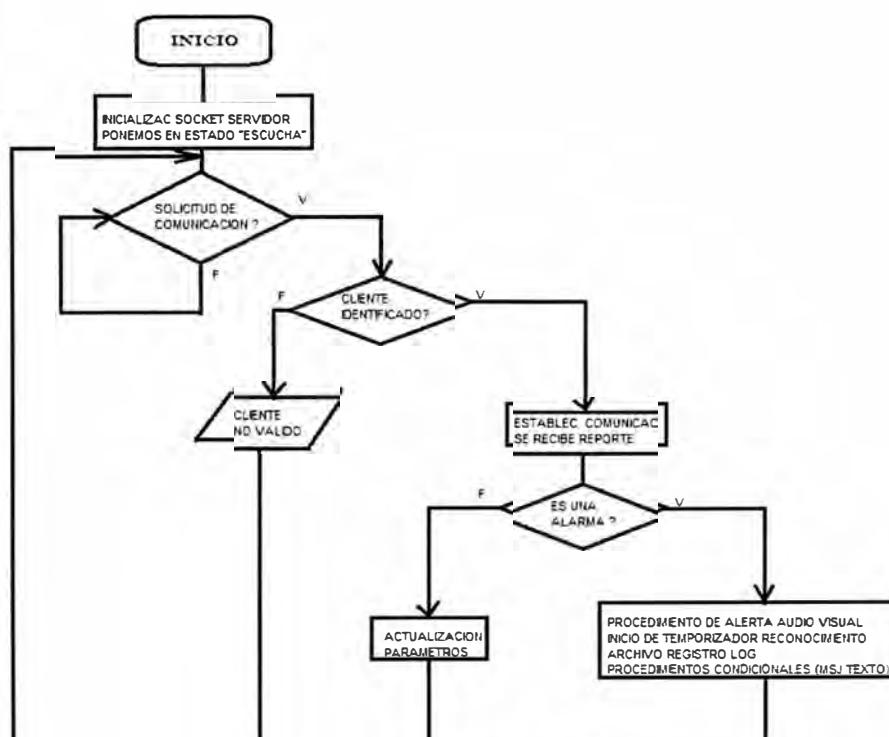


Figura 3.18 Diagrama Flujo Socket Servidor (Fuente: Diseño propio)

Cuando una unidad remota requiera enviar información al centro de control, el respectivo socket cliente realizará una petición al socket servidor del centro de control, que debería estar en estado de “escucha”, luego de las respectivas validaciones se establece la comunicación y la cadena de data será procesada por el programa para seguir su ejecución programada.

En la figura 3.18 se muestra el diagrama de flujo para el proceso de recepción de data entrante.

El socket servidor estará siempre en escucha, cuando se genere el evento “DataArrival” el programa procesará la data recibida, en la mayoría de los casos la data indicará que el sistema está reportando simplemente un estado, lo que nos permite conocer que cierta unidad remota está reportando normalmente. En casos excepcionales y dependiendo del estado del sistema una data entrante puede significar una alarma, si se da el caso de una alarma se llama al procedimiento de alerta, que consiste en una ventana emergente con el reporte respectivo y se reproduce un clip de audio de corta duración correspondiente a una alarma. Seguidamente se inicia un temporizador para controlar que el reconocimiento de parte del operador se realice en un intervalo de tiempo de 2 minutos. Si en el caso de no recibir el reconocimiento en dicho intervalo, el sistema enviará los comandos al módem GSM GPRS para enviar la alerta vía mensaje de texto a los números telefónicos previamente establecidos.

b.- Envío de Comandos a Unidad Remota

El software de control puede enviar a una unidad remota determinada los comandos respectivos para cumplir cierta acción. En la tabla 3.3 se muestra la relación de los comandos establecidos para el envío a la unidad remota. Los mismos que remotamente permite al sistema de alarmas recibir órdenes de parte de la central de monitoreo y ejecutar acciones predefinidas.

Tabla 3.3 Comandos de control.

COMANDO	DESCRIPCION
START	Mensaje que se envía al inicializar el programa de monitoreo, permite inicializar el temporizador de la unidad remota para el reporte periódico y la comunicación instantánea de cambios de estado de la unidad remota.
STOP	Detiene el intercambio de comunicación entre la unidad remota y el centro de control.
LSOUT1	Comando que envía la orden de activar la salida audible en la unidad remota. A la vez en la unidad remota se inicializa el temporizador para controlar el tiempo máximo de duración

	de la emisión de sonido.
LSOUT2	Detiene o Silencia la salida audible en el caso de que esta se encuentre activada.
LMSG "xxxxxx"	Permite reproducir mensajes de texto en el display LCD de una unidad remota

c.- Registro de Eventos

Para registrar los eventos de alarma se crea un archivo tipo "log", donde se agrega la descripción de un evento de alarma adicionalmente la hora y fecha de dicho evento. La hora y fecha del sistema se obtiene mediante la función `Format(Now,"yyyymmddhhmmss")`, que devuelve una cadena tipo 20130901210406 para el evento que ocurra el 01 de setiembre del año 2013 a horas 21:04:06.

Cada vez que se acceda al archivo log se verificará en número de líneas que contiene, fijamos un numero tal como 200 para que sea el máximo número de líneas para el archivo, si se alcanza el valor 200 renombramos el archivo y creamos un nuevo archivo log.

d.- Envío Condicional de Mensajes a Móviles

El software de monitoreo inicializa un contador de tiempo cuando se genere una alarma que requiera el reconocimiento de parte del operador, si este reconocimiento no sucede en el lapso de 2 minutos el sistema iniciará el procedimiento de enviar vía mensaje de texto a terminales móviles, cuyos números se tienen almacenados.

El modem GSM GPRS se comunica con el ordenador del centro de control vía el puerto serial RS232.

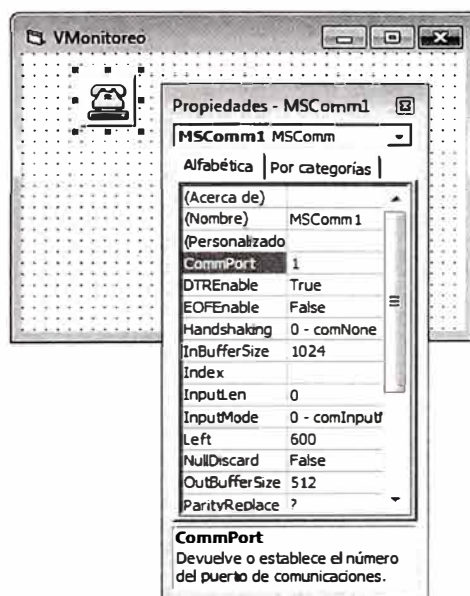


Figura 3.19 Componente MSComm (Fuente: Captura VB)

El módulo GSM/GPRS requiere la incorporación de un chip SIM, que tenga la capacidad de enviar mensajes de texto a través de la red de telefonía y opcionalmente acceso a comunicación por paquetes de datos, que le permitiría además tener la posibilidad del envío de emails.

Para realizar la conectividad, visual basic incorpora la herramienta MSComm que permite realizar comunicación con periféricos vía transmisión serial.

El ordenador comanda las ejecuciones del modem GSM GPRS enviándole comandos AT, a continuación se muestra el fragmento de programa que envía el mensaje de texto a un número telefónico almacenado en la variable NTEL01.

```
Private Sub EnviarMSGTXT(Msg_Txt)
    MSComm1.CommPort=1
    MSComm1.Settings="115200,N,8,1"
    MSComm1.PortOpen=True
    MSComm1.Output="at+cmgs=" & NTEL01 & Chr(13)
    MSComm1.Output=Msg_Txt & Chr(26)
End Sub
```


CAPITULO IV

ANALISIS Y PRESENTACION DE RESULTADOS

En el presente capítulo se presenta un análisis descriptivo del sistema desarrollado, además un estimado de costos que significaría su implementación. Teniendo en cuenta que el desarrollo completo engloba tareas multidisciplinarias, se plantea la proyección de desarrollo especializado en la parte de software.

4.1 Análisis descriptivo

A continuación se detalla las características funcionales del sistema de monitoreo de alarmas materia del presente estudio, dividido según los sub sistemas componentes:

4.1.1 Unidad Remota

En cada unidad remota se tiene un sub sistema de alarmas, cuyo control se realiza completamente desde el centro de control. A cada unidad remota se conectan principalmente los sensores de alarmas, que pueden ser: contactos magnéticos, detectores volumétricos infrarrojos, pulsadores de emergencia, etc. También se conecta a ella el dispositivo de anunciación sonora.

La unidad remota transmite al centro de control la data correspondiente al estado de las variables controladas.

Para minimizar la ocupación del ancho de banda de la red de comunicaciones, la unidad remota solo transmitirá eventos al centro de control si se produce un cambio de estado del sistema, que corresponde principalmente a los estados de las zonas de alarma y de monitoreo de variables internas de energía. Si no se produce eventos que deben ser reportados en un lapso de tiempo (programable), la unidad remota transmitirá su estado al centro de control, siendo una manera de confirmar el estado "en línea" de la unidad remota.

4.1.2 Centro de Control

En el centro de control se tiene un computador, donde se ejecuta el programa de monitoreo. Las actividades que se pueden realizar son:

- Iniciar o Finalizar las comunicaciones con una unidad remota específica.
- Conocer el estado de las zonas de alarma, tensión AC, estado de la batería de respaldo de cualquier unidad remota con comunicación en curso.
- Realizar el Armado o Desarmado de cualquier unidad remota con comunicación

en curso.

- En el caso de un evento de alarma, mostrar mensaje visual y sonoro, adicionalmente enviar los comandos respectivos a la unidad remota en el caso que dicha alarma necesite alerta sonora local. En forma sub siguiente se inicia la temporización en espera de la confirmación manual por parte del operador. En el caso que la confirmación no se produzca en el límite de tiempo programado, se iniciará el envío de mensaje de texto vía red celular, hacia los números telefónicos pre programados
- Enviar mensajes de texto a cualquier unidad remota conectada

4.2 Estimación de costos en Equipos y Componentes

Listamos a continuación la relación de los componentes electrónicos y módulos que conforman el presente trabajo. Inicialmente consideramos 01 unidad remota. El costo promedio de los componentes se relaciona con los obtenidos en el medio local (Lima – Perú).

Tabla 4.1 Estimación de costos en componentes y equipos.

ITEM	CANTIDAD	DESCRIPCION	COSTO S/.
1	1	Microcontrolador PIC 16F877A	30.00
2	1	Módulo Ethernet Shield W5100	100.00
3	1	Modem GSM/GPRS Wavecom Fastrack	400.00
4	1	Transformador para fuente	20.00
5	1	Pantalla LCD 2x16	40.00
6	1	Cristal Resonador 20MHz	5.00
7	6	Diodos	5.00
8	2	Elaboración de Circuito Impreso	60.00
9	1	Componentes Misceláneos	30.00
10	1	Gabinete Metálico para Panel	90.00
11	1	Plan Mensual Comunicaciones Móviles (500MB datos SMS Ilimitados)	129.00
TOTAL S/.			909.00

4.3 Proyección de Implementación

En la tabla 4.2 se muestra la proyección de la implementación del sistema propuesto, en conjunto con las respectivas pruebas que permitirán realizar ajustes necesarios al sistema. Se considera el desarrollo de un prototipo con todas las funcionalidades del sistema propuesto.

CONCLUSIONES Y RECOMENDACIONES

- 1.- Mediante la aplicación de sistemas electrónicos basados en microcontroladores PIC y periféricos de aplicación específica se ha logrado estructurar una solución aplicada a la seguridad electrónica en forma centralizada, cumpliendo el objetivo buscado en el presente trabajo.
- 2.- La interconexión de los múltiples sistemas de alarmas basados en PIC se realiza a través de la red informática Ethernet existente, para poder conectar cada sistema de alarma a la red se realizó el estudio del módulo Ethernet shield W5100, que permite a cada sub-sistema de alarmas la asignación de una dirección IP, haciendo posible la capacidad de poder interactuar con diferentes equipos conectados a la red, en el caso del presente trabajo la comunicación está dirigida principalmente a reportar a una unidad central de monitoreo.
- 3.- Una vez obtenido la interconexión a través de la red Ethernet de los sub-sistemas de alarma, mediante software desarrollado en Visual Basic y la definición de un protocolo propietario de comandos, se comanda los subsistemas de alarma desde el centro de control. Permitiendo al centro de control activar y desactivar el sistema de alarmas, y recibir periódicamente el estado de las zonas de alarma de cada unidad remota.
- 4.- Un punto vital del sistema de monitoreo es el monitoreo remoto, para dar aviso a los responsables de la seguridad. El uso del modem GSM/GPRS cumple todos los requisitos de diseño: alta performance, fiabilidad. Además es bastante flexible para su manejo, lo cual se logró mediante el periférico de comunicación serial RS232 de la computadora del centro de control.
- 5.- Un desarrollo con prestaciones más sofisticadas se puede lograr en un trabajo conjunto de un especialista en desarrollo de software. Pudiendo elaborar una interactividad mejorada, en la gestión de archivos, ventanas, contenido multimedia.

ANEXO A
MICROCONTROLADOR PIC 16F877A

EL MICROCONTROLADOR 16F877A

La figura A.1 muestra la presentación del PIC 16F877A con el encapsulado de 40 pines, es el modelo comercialmente más común en el medio local. Es la presentación que ofrece mayores facilidades a la hora de realizar pruebas en un protoboard.



Figura A.1 Encapsulado 40 pines (Fuente: Microchip)

La Tabla A.1 muestra las características del Pic 16F877A, obtenido de la dirección web del fabricante [Microchip](http://www.microchip.com):

Tabla A.1 Características PIC 16F877A

Parámetro	Valor
Tipo de Memoria de Programa	Flash
Memoria de Programa (KB)	14
Velocidad CPU (MIPS)	5
RAM Bytes	368
Memoria EEPROM (bytes)	256
Comunicación Digital a Periféricos	1-A/E/USART, 1-MSSP(SPI/I2C)
Periféricos Captura/Comparador/PWM	2 CCP
Temporizadores	2x8-bit, 1x16-bit
ADC Convertidor Análogo a Digital	8 ch, 10-bit
Comparadores	2
Temperatura. Rango de Operación (°C)	-40 a 125
Rango de Voltaje de Operación (V)	2 a 5.5
Número de pines	40

a.- Convertidor Analógico a Digital Integrado

El microcontrolador PIC 16F877A incorpora un módulo de conversión de señal analógica a señal digital, tiene 8 pines que pueden operar como entradas para señales analógicas, el resultado de la conversión es un valor digital de 10 bits. El módulo realiza la conversión de una sola señal a la vez, para utilizar más de una señal, debemos programar la conversión secuencial de las señales a través de código, donde se describa la selección, la orden de adquisición de data, para cada señal. Por lo expuesto queda implícito que dependiendo de la cantidad de señales analógicas a convertir, teniendo en cuenta que el proceso de conversión requiere un intervalo de tiempo (20 uS típico), la conversión no es continua y su aplicación puede ser inadecuada en procesos donde el periodo de muestreo sea crítico, para aplicaciones de seguridad electrónica periodos de lectura y muestreo de 160 microsegundos es aceptable.

El módulo de conversión se caracteriza por los parámetros siguientes:

- Rango de entrada.
- Número de bits.
- Resolución.
- Tensión de fondo de escala.
- Tiempo de conversión.
- Error de conversión

La resolución está determinada por la siguiente relación:

$$\frac{V_{in}}{2^N - 1} \quad A.1$$

Donde V_{in} es la tensión de entrada y N el número de bits del convertidor. Para una tensión máxima de 5V, N igual a 10, la resolución es de 4.8mV por LSB.

a.1.- Registros del módulo A/D

El módulo A/D tiene cuatro registros. Estos registros son:

- A/D ADRESH Bits superiores del resultado de conversión.
- A/D ADRESL Bits inferiores del resultado de conversión.
- A/D ADCON0 Registro de control 0.
- A/D ADCON1 Registro de control 1.

El registro ADCON0 controla las operaciones del módulo A/D. El registro ADCON1 configura las funciones de los pines del puerto. Los pines del puerto pueden ser como configurados como entradas análogas (RA3 puede también configurarse como voltaje de referencia) o como entrada/salida digital I/O.

a.2.- Registro ADCON0 (Dirección 1Fh)

La tabla siguiente muestra la estructura del registro ADCON0.

Tabla A.2 Registro ADCON0

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	--	ADON

bit 7

bit 0

El primer bit activa el módulo ADC cuando su valor es "1" lógico y desactiva en caso contrario. El tercer bit indica el estado de la conversión siempre y cuando el primer bit se establezca a "1", así un valor de "1" lógico indica que el proceso de conversión está en progreso, este bit es puesto a "0" lógico automáticamente cuando finaliza el proceso de conversión. El cuarto, quinto y sexto bit configuran la selección de un canal determinado que estará disponible para el proceso de conversión, con los tres bits se puede seleccionar cualquier canal según el valor binario del número formado por los tres dígitos. El séptimo y octavo bit en conjunción con el séptimo bit del registro ADCON1 permiten seleccionar las características de la frecuencia de muestreo.

a.3.- Registro ADCON1 (Dirección 9Fh)

La tabla siguiente muestra la estructura del registro ADCON1. Los primeros 4 bits correspondientes al registro ADCON1 conforman un cuadro de control, el fabricante entrega una tabla donde según la combinación de estos se configuran el comportamiento de los pines AN0-AN7, así como los voltajes de referencia.

Tabla 3.3 Registro ADCON1

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADSC2			PCFG3	PCFG2	PCFG1	PCFG0

El séptimo bit conforma una tabla de valores de configuración del reloj del módulo A/D. El octavo bit configura el formato de salida del valor obtenido en el proceso de conversión, así un valor de "1" lógico significa una justificación derecha, de esta forma los seis bits superiores del registro ADRESH son leídos como "0" lógico.

a.4.- El proceso de conversión

Para realizar la conversión, el fabricante recomienda seguir los siguientes pasos:

1. Configurar el módulo A/D:
 - a) Configuración de pines analógicos/ tensión de referencia/ E/S digitales con los valores adecuados en el registro ADCON1.
 - b) Selección de la entrada A/D a través de asignar los valores correspondientes en el registro ADCON0.
 - c) Selección de reloj para la conversión A/D (ADCON0).
 - d) Habilitar el módulo A/D (ADCON0<0>).
2. Configurar las interrupciones (opcional):
 - a) ADIF = 0;

- b) $GIE = PEIE = ADIE = 1$.
3. Esperar el tiempo de adquisición.
4. Comenzar la conversión estableciendo a "1" el bit $ADCON0<2>$.
5. Esperar a que termine la conversión, puede ser de dos maneras:
 - a) Mediante lectura continua del bit $GO/DONE$ hasta que sea "0".
 - b) Esperando la interrupción.
6. Leer el registro de conversión $ADRES$ y borrar el flag $ADIF$ si es necesario.
7. Para la siguiente conversión se salta a los puntos 1, 2 ó 3 en función de lo que se necesite.

Existen dos tiempos básicos relacionados con el proceso de conversión analógica a digital: el tiempo de adquisición T_{ACQ} y el tiempo de conversión T_{AD} .

- **Tiempo de adquisición (T_{ACQ}):** Es el tiempo necesario para que se cargue el condensador integrado que actúa como retención de la tensión de entrada. Un factor importante en este proceso viene a ser la impedancia de la fuente de tensión de entrada, el fabricante recomienda que dicha impedancia sea menor a 10 kohm. El tiempo de adquisición típicamente es 20 μs .
- **Tiempo de conversión (T_{AD}):** Es el tiempo necesario para obtener el valor digital correspondiente a la entrada analógica. Este tiempo depende del reloj que se seleccione para la conversión. El fabricante recomienda que el periodo mínimo seleccionado debe asegurar un tiempo mínimo T_{AD} de 1.6 μs .

a.5.- Efecto del modo SLEEP y RESET en el módulo AD

En el modo dormido (mínimo consumo sleep), el módulo AD puede funcionar si se selecciona como reloj para la conversión el modo RC interno ($ADCS1=1$ y $ADCS0=1$). En este caso el módulo espera 1 ciclo de instrucción antes de iniciar la conversión, permitiendo la ejecución de la siguiente instrucción SLEEP. Al finalizar la conversión, el bit $GO/DONE$ es puesto a "0" y el resultado se carga en los registros correspondientes $ADRESH$ y $ADRESL$. Si la interrupción del módulo AD está habilitado ($ADIE=1$ y $PEIE=1$) el dispositivo se despierta, caso contrario el módulo se apaga aunque el bit $ADON$ se encuentre en "1".

Ante un RESET, los registros del módulo AD se inicializan a los valores por defecto, apagándose el módulo AD y se realiza la parada instantánea de la conversión actual.

a.6.- Módulo AD en C

El compilador C permite manejar el convertidor AD mediante las siguientes funciones:

i.- `setup_adc(modos);`

Para la configuración del módulo A/D correspondientes a los bits 7:6 del registro $ADCON0$ C define las siguientes constantes, descritos en la tabla A.4, estas constantes

permiten desactivar el ADC, establecer el reloj de muestreo:

Tabla A.4 Valores para setup_adc.

setup_adc(modo);	ADCON0(1Fh)
ADC_OFF	00h
ADC_CLOCK_INTERNAL	C0h
ADC_CLOCK_DIV_2	00h
ADC_CLOCK_DIV_8	40h
ADC_CLOCK_DIV_32	80h

ii.- setup_adc_ports(valor);

Donde “valor” representa una constante pre-definida que configura la función de los pines de entrada del módulo A/D. El valor utilizado para definir las ocho entradas como entradas analógicas es ALL_ANALOG.

iii.- set_adc_channel(canal);

Donde “canal” representa una constante cuyos valores son: 0,1,...,7 que corresponde a los pines AN0,AN1,...,AN7. Permite seleccionar un canal determinado para el muestreo respectivo. Esta función se aplica principalmente cuando se muestrean múltiples señales analógicas..

iv.- valor=read_adc();

La función read_adc() realiza la lectura del resultado de la conversión, donde valor es un entero de 16 bits según la directiva #DEVICE ADC=n . Dicha directiva trabaja según se describe en la tabla A.5.

Tabla A.5 Valores para la función read_adc.

DEVICE	8 bit	10 bit	11 bit	16 bit
ADC=8	00-FF	00-FF	00-FF	00-FF
ADC=10	X	0-3FF	X	X
ADC=11	X	X	0-7FF	X
ADC=16	0-FF00	0-FFC0	0-FFE0	0-FFFF

a.7.- Módulo USART (Transmisión Recepción Universal Sincrono Asíncrono).

El módulo USART es uno de los módulos de comunicación serial E/S. El USART puede ser configurado como un sistema asíncrono full-duplex que permite comunicaciones con dispositivos periféricos, tales como terminales CRT y computadores personales, o puede ser configurado como un sistema semi-duplex que puede comunicar dispositivos periféricos, como convertidores A/D o D/A, memorias EEPROM seriales, etc.

El USART puede ser configurado en los siguientes modos:

- Asíncrono (full-duplex)

- Síncrono – Maestro (Semi-duplex)
- Síncrono – Esclavo (Semi-duplex)

Básicamente, la transmisión serial se realiza enviando bit a bit a través de una línea común en periodo de tiempos fijos, por ello es importante establecer la velocidad de transmisión o número de bits por segundo (baudios) con el mismo valor tanto en el transmisor como en el receptor. El transmisor y receptor poseen registros de desplazamiento para realizar la comunicación. Los bits pueden codificarse tipo NRZ(nivel alto: 1, nivel bajo: 0) ó NRZI(cambio de nivel:1, constante:0).

El modo más utilizado para comunicación mediante el USART es el modo asíncrono, debido a que permite comunicaciones de mayores distancias relativas. Existen normas de transmisión serie asíncrona, como la RS232, la RS485, entre otros. Los niveles de tensión empleados por estas normas no siempre son las mismas que utiliza el PIC. Motivo por el cual se requieren circuitos externos de adaptación.

a.7.1.- Registros de Control

El módulo MSSP tiene tres registros asociados. Estos incluyen 01 registro de estado (SSPSTAT) y 02 registros de control (SSPCON y SSPCON2).

a.7.2.- Modo SPI

El modo SPI permite la transmisión y recepción simultánea de data de 8 bits en forma síncrona. Los pines usados para la comunicación son 03:

- Salida de data serial (SDO) – RC5/SD0
- Entrada de data serial (SDI) – RC4/SDI/SDA
- Reloj de comunicación serial (SCK) – RC3/SCK/SCL

Si nuestro microcontrolador opera en modo esclavo, se puede usar un cuarto pin:

- Selector Esclavo (SS) – RA5/AN4/SS/C2OUT

i.- Operación

Cuando se inicializa el modo SPI, varias opciones se requiere sean especificadas. Esto se realiza por programación de los bits de control (SSPCON<5:0> y SSPSTAT<7:6>). Estos bits de control permiten especificar:

- Modo Maestro (SCK es la salida de reloj).
- Modo Esclavo (SCK es la entrada de reloj).
- Polaridad del reloj (Estado estacionario del SCK).
- Fase de Muestreo de data de entrada (flanco se subida o bajada).
- Eje de reloj (especifica si la salida se realiza en flanco de subida o bajada).
- Frecuencia de reloj (en modo Maestro solamente).
- Selector esclavo (en modo esclavo solamente).

El módulo MSSP consiste de un registro de transmisión/recepción (SSPSR) y un registro buffer (SSPBUF). El registro SSPSR realiza el desplazamiento de la data de ingreso y salida. Comenzando por el bit de mayor peso (MSb). El registro SSPBUF mantiene la data que será enviado al registro SSPSR hasta que la recepción se complete. Una vez que los ocho bits de la data se han recibido, que el byte completo es movido al SSPBUF. Luego el bit de interrupción SSPSTAT<0>, que indica que el registro SSPBUF ha completado un ciclo de 8 bits, es puesto a "1". Esta operación en dos registros, permite que el siguiente byte inicie la recepción antes de leer la data recibida. Cuando el software de aplicación está a la espera de recibir data, el registro SSPBUF debe leer antes el siguiente byte.

ii.- Habilitación SPI E/S

Para habilitar el puerto serial, el bit "SSP Enable", SSPEN (SSPCON<5>) debe ser establecido a "1" lógico. Para reinicializar o reconfigurar el modo SPI, debemos poner a "0" lógico el bit SSPEN, reinicializar el registro SSPCON y luego establecer a "1" lógico. Esto configura los pines SDI, SDO, SCK y SS como pines del puerto serial. Para que los pines funcionen adecuadamente, algunos bits del registro TRIS deben estar apropiadamente programados. Así:

- SDI es automáticamente controlado por el módulo SPI.
- Para SDO el bit TRISC<5> debe ser "0" lógico.
- Para SCK (En modo Maestro) el bit TRISC<3> debe ser "0" lógico.
- Para SCK (En modo Esclavo) el bit TRISC<3> debe ser "1" lógico.
- Para SS el bit TRISC<4> debe ser "1" lógico.

iii.- Conexión Típica

La figura A.2 muestra una conexión típica entre dos microcontroladores. El maestro (Procesador 1) inicia la transferencia de data enviando la señal SCK. La data es desplazada a través de los dos registros involucrados, estableciéndose la comunicación. Ambos procesadores deben tener los relojes con la misma polaridad (CKP), luego ambos controladores deben enviar y recibir data a la vez.

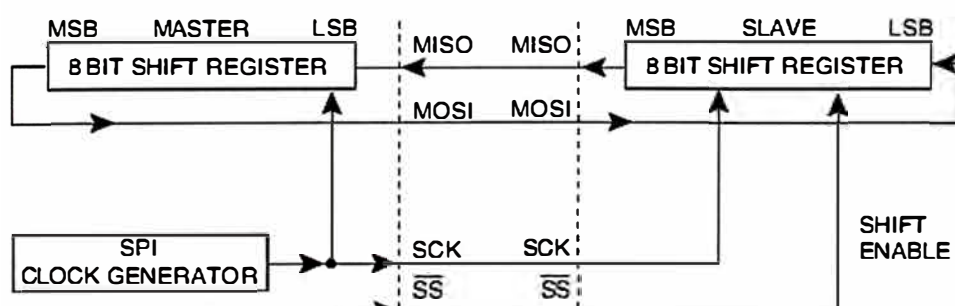


Figura A.2 Conexión Maestro – Esclavo (Fuente: Microchip)

iv.- Modo Maestro

El maestro puede iniciar la transferencia de data en cualquier momento porque es quién controla el reloj SCK. El maestro determina cuando el esclavo puede transmitir data. En modo Maestro, la transmisión/recepción se realiza según se escribe data en el registro SSPBUF.

a.7.2.- El módulo USART en C

La configuración genérica del USART se realiza mediante una directiva específica:

`#USE RS232 (Opciones)`

Esta directiva permite configurar los parámetros del módulo USART, tales como: velocidad de transmisión, número de pines utilizados, etc. Las directivas más importantes se muestran en la tabla A.6:

Tabla A.6 Directivas USART.

BAUD=X	Velocidad en Baudios
XMIT=pin	Pin de transmisión
RCV=pin	Pin de recepción
FORCE_SW	Usa un software UART en lugar de hardware aun cuando se especifican los pines de hardware.
RESTART_WDT	Hace que la función GETC() ponga a cero el WDT mientras se espera un carácter
BRGH10K	Permite velocidades de transmisión bajas en chips que tienen problemas de transmisión.
ENABLE=pin	El pin especificado estará a nivel alto durante la transmisión. Utilizado en transmisión 485.
DEBUGGER	Permite la depuración a través del ICD. El pin por defecto es el B3; usar XMIT y RCV para cambiar el pin (debe ser el mismo en ambos)
RESTART_WDT	Provoca que la función GETC() borre el WDT si espera un carácter.
INVERT	Invierte la polaridad de los pines serie (normalmente no es necesario con el convertidor de nivel, como el MAX232). No puede usarse con el USART interno.
PARITY=X	Donde X es N,E u O
BITS=X	Donde X es 5-9 (no puede usarse 5-7 con el USART interno)
FLOAT_HIGH	Se utiliza para las salidas de colector abierto.
ERRORS	Indica al compilador que guarde los errores recibidos en la

	variable RS232_ERRORS para restablecerlos cuando se producen
SAMPLE_EARLY	No se puede utilizar con USART interno. Provoca que el muestreo del dato a través de la función GETC() se realice al principio de un bit de tiempo.
RETURN=pin	Para FLOAT_HIGH y MULTI_MASTER, este pin se usa para leer la señal de retorno. Por defecto, para FLOAT_HIGH es XMIT y para MULTI_MASTER es RCV.
MULTI_MASTER	Usa el pin de RETURN para determinar si otro master en el bus está transmitiendo al mismo tiempo. Si se detecta una colisión, el bit 6 se pone a 1 en RS232_ERRORS y todos los posibles PUTC() son ignorados hasta que el bit 6 esté a 0. La señal es comprobada al final y al principio de cada bit de tiempo. No se puede utilizar con USART interno.
LONG_DATA	Permite manejar INT16 a las funciones GETC() y PUTC(). En formatos de datos de 9 bits.
DISABLE_INTS	Provoca la des habilitación de interrupciones cuando se ejecuta GETC() y PUTC() evitando distorsiones en los datos.
STOP=x	Numero de bits de parada (por defecto 1)
TIMEOUT=x	Para establecer el tiempo de GETC() espera un carácter (en ms). Si no se recibe carácter en este tiempo, el RS232_ERRORS se pone a 0.
SYNC_SLAVE	Provoca una línea RS232 en modo esclavo síncrono, haciendo la patilla de recepción como entrada de reloj y la de transmisión como entrada/salida de datos.
SYNC_MASTER	Provoca una línea RS232 en modo maestro síncrono, haciendo la patilla de recepción como salida de reloj y la de transmisión como entrada/salida de datos.
SYNC_MASTER_C ONT	Provoca una línea RS232 en modo maestro síncrono en modo continuo, haciendo la patilla de recepción como salida de reloj y la de transmisión como entrada/salida de datos.
UART1	Configurar el XMIT y RCV para el USART1.
UART2	Configurar el XMIT y RCV para el USART2.

Ejemplos de aplicación:

```
#use delay (clock=2000000);
```

```
#use rs232 (BAUD=9600, XMIT=PIN_C6, RCV=PIN_C7, BITS=8)
```

#use rs232 (BAUD=9600, XMIT=PIN_A2, RCV=PIN_A3)

i.- setup_uart(baud)

Donde "baud" es una constante que establece la velocidad. Al establecerlo en 1 lógico enciende el USART y un 0 lógico lo apaga. Cualquier valor de velocidad hace que el USART se encienda. En dispositivos que utilizan AUSART se admiten los siguientes parámetros:

UART_ADDRESS UART: Solo acepta datos con el noveno bit a 1.

UART_DATA UART: acepta todos los datos.

ii.- set_uart_speed(baud)

Donde baud es una constante que establece la velocidad. Al establecerlo en 1 lógico enciende el USART y un 0 lógico lo apaga. Cualquier valor de velocidad hace que el USART se encienda. En dispositivos que utilizan AUSART se admiten los siguientes parámetros:

UART_ADDRESS UART: Solo acepta datos con el noveno bit a 1.

UART_DATA UART: acepta todos los datos.

En el siguiente ejemplo establecemos la velocidad de transmisión mediante la combinación de los pines B0 y B1, para ello realizamos un AND lógico con 3 que en binario permite obtener los 2 primeros bits de un valor:

```
switch( input_b() & 3 ){
  case 0 : set_uart_speed(2400);break;
  case 1: set_uart_speed(4800);break;
  case 2: set_uart_speed(9600);break;
  case 3: set_uart_speed(19200);break;
```

ANEXO B

INTERFACE ETHERNET SHIELD W5100

ETHERNET SHIELD W5100

En la figura B.1 se muestra el diagrama de bloques del módulo Ethernet Shield W5100.

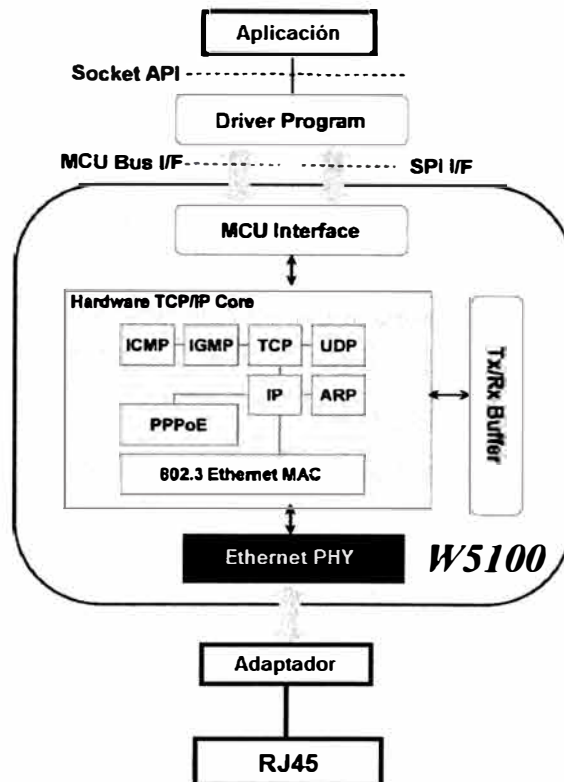


Figura B.1 Diagrama de Bloques W5100 (Fuente: Wiznet)

a.- Señales de Interface MCU

Tabla B.1 Señales MCU

Símbolo	Tipo	Pin N°	Descripción
/RESET	I	59	RESET Se pone a nivel bajo para inicializar o reinicializar el W5100 Se requiere 2µs mínimo para operación.
ADDR 14-0	I	38-42, 45-54	DIRECCIONES Seleccionan un registro o dirección de memoria.

DATA 7-0	I/O	19-26	DATA Usados para lectura y escritura de registros o memoria de data.
/CS	I	55	CHIP SELECT Selecciona el acceso a registros internos o memoria. /WR y /RD seleccionan la dirección de la transferencia de data.
/INT	O	56	INTERRUPCION Indica que el W5100 requiere atención después de conexión, desconexión, recepción de data o tiempo máximo de espera. Está activo en nivel bajo. Es enmascarable.
/WR	I	57	HABILITAR ESCRITURA Habilita la escritura en un registro interno o memoria. Está activo en nivel bajo.
/RD	I	58	HABILITAR LECTURA Habilita lectura desde un registro interno o memoria. Está activo en nivel bajo.
SEN	I	31	HABILITAR SPI Habilita/ Deshabilita el modo SPI Nivel Bajo = Modo SPI deshabilitado Nivel Alto = Modo SPI habilitado
SCLK	I	30	RELOJ SPI Señal de reloj en modo SPI
/SCS	I	29	SELECTOR ESCLAVO SPI
MOSI	I	28	SPI Maestro Salida – Esclavo Entrada
MISO	O	27	SPI Maestro Entrada – Esclavo Salida

b.- Mapa de la Memoria

El W5100 está compuesto por Registros comunes, Registros Socket, Memoria de Transmisión TX, y memoria de Recepción RX. La documentación acerca de la estructuración de la data y los valores admitidos permite obtener un funcionamiento

buscado.

Los registros comunes contienen la data de configuración del sistema, como dirección IP, Mac Address o dirección de hardware, y otros valores de configuración que configuran el W5100.

Los registros sockets contienen la data referente a los mismos, tales como protocolos de comunicación, valores de estado.

Los registros de memoria contienen la data del sistema, que se utilizan en transmisión y recepción.

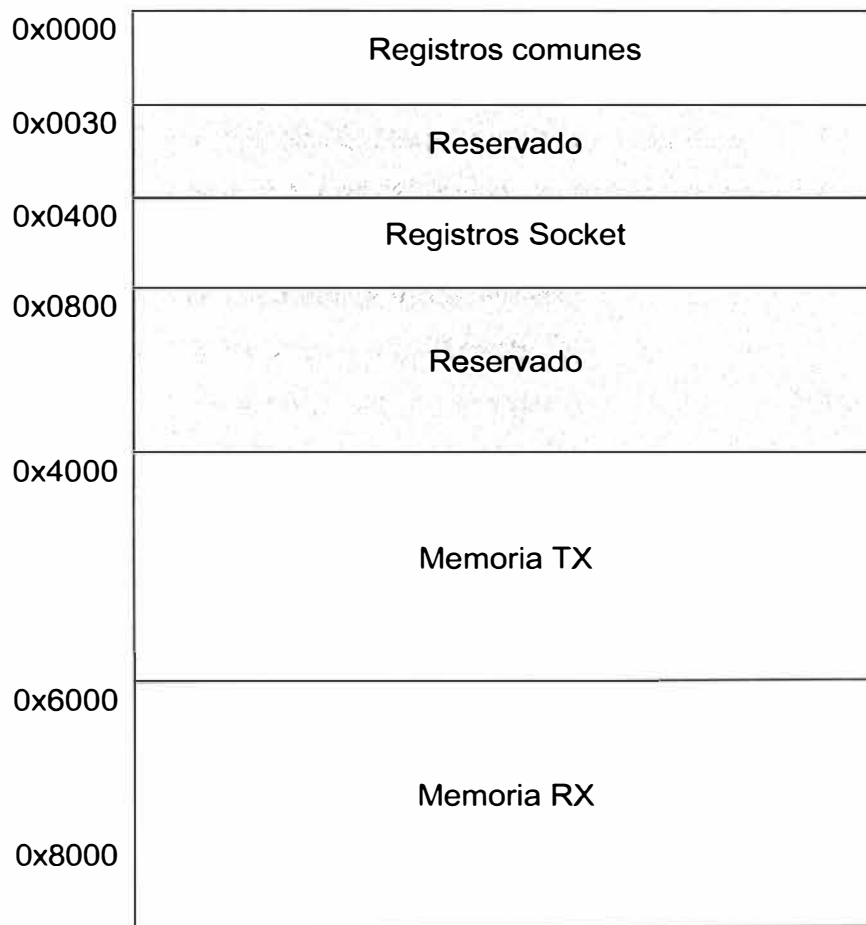


Figura B.2 Mapa de Memoria (Fuente: Wiznet)

c.- Descripción de Registros del W5100

i.- MR (Registro Modo) [R/W] [0x0000][0x00]

Según el valor lógico de los bits correspondientes a este registro se controla: la re inicialización, modo test de memoria, modo bloqueo ping, modo PPPoE y modo de direccionamiento indirecto.

7	6	5	4	3	2	1	0
RST			PB	PPPoE		AI	IND

Tabla B.2 Contenido registro MR

Bit	Símbolo	Descripción
7	RST	Reset Si este bit se pone a “1” lógico los registros internos son reinicializados.
6	Reservado	
5	Reservado	
4	PB	Modo Bloque Ping 0: Deshabilita el bloqueo de la función Ping 1: Habilita el bloqueo de la función Ping Si el valor es “1” lógico, no habrá respuesta a un PING
3	PPPoE	Modo PPPoE 0: Deshabilita modo PPPoE 1: Habilita modo PPPoE Protocolo de comunicación punto a punto sobre Ethernet.
2	No Usado	
1	AI	Auto Incremento Dirección en Bus Indirecto I/F 0: Deshabilita auto-incremento 1: Habilita auto-incremento
0	IND	Modo Bus Indirecto I/F 0: Deshabilita modo bus indirecto I/F 1: Habilita modo bus indirecto I/F

ii.- GWR (Registro Dirección IP Enlace) [R/W][0x0001-0x0004][0x00]

Este registro establece la dirección por defecto del enlace.

Ejemplo: en caso de ser “192.168.1.1”

0x0001	0x0002	0x0003	0x0004
192(0xC0)	168(0xA8)	1(0x01)	1(0x01)

iii.- SUBR (Registro Máscara de Subred)[R/W][0x0005-0x0008][0x00]

Este registro establece la máscara de subred.

Ejemplo: en caso de ser “255.255.255.0”

0x0005	0x0006	0x0007	0x0008
255(0xFF)	255(0xFF)	255(0xFF)	0(0x00)

iv.- (Registro Dirección Hardware MAC)[R/W][0x0009-0x000E][0x00]

Este registro establece la dirección de hardware MAC.

Ejemplo: en caso de ser "00.08.DC.01.02.03"

0x0009	0x000A	0x000B	0x000C	0x000D	0x000E
0x00	0x08	0xDC	0x01	0x02	0x03

v.- SIPR (Registro dirección IP)[R/W][0x000F-0x0012][0x00]

Este registro establece la dirección IP, de acuerdo al segmento utilizado en el sistema para la comunicación.

Ejemplo: en caso de ser "192.168.1.7"

0x000F	0x0010	0x0011	0x0012
192(0xC0)	168(0xA8)	1(0x01)	7(0x07)

vi.- IR (Registro Interrupción)[R][0x0015][0x00]

Este registro es accedido por el procesador para conocer la causa de una interrupción.

Cualquier interrupción puede ser enmascarada a través del registro IMR (máscara de interrupciones). La señal /INT se mantiene en nivel bajo en tanto su bit correspondiente esté establecido a "1" lógico.

7	6	5	4	3	2	1	0
CONFLICTO	INALCANZABLE	PPPoE		S3_INT	S2_INT	S1_INT	S0_INT

Tabla B.3 Contenido Registro IR

Bit	Símbolo	Descripción
7	CONFLICTO	Conflicto IP
6	INALCANZABLE	En modo UDP indica destino inalcanzable.
5	PPPoE	Cierre de conexión punto a punto.
4		
3	S3_INT	Indica ocurrencia en socket respectivo.
2	S2_INT	
1	S1_INT	
0	S0_INT	

vii.- IMR (Registro Máscara de Interrupciones)[R/W][0x0016][0x00]

Este registro es usado para enmascarar interrupciones. Cada bit corresponde a un bit en el registro IR (registro de interrupciones). Una interrupción ocurrirá siempre en cuando el bit correspondiente en IMR sea "1" lógico.

7	6	5	4	3	2	1	0
IM_IR7	IM_IR6	IM_IR5		IM_IR3	IM_IR2	IM_IR1	IM_IR0

Tabla B.4 Registro IMR

Bit	Símbolo	Descripción
7	IM_IR7	Habilitar Indicador de Conflicto IP
6	IM_IR6	Habilitar Indicación Destino Inalcanzable
5	IM_IR5	Habilitar Indicación de cierre PPPoE
4		
3	IM_IR3	Habilitar indicación de Interrupción Socket correspondiente.
2	IM_IR2	
1	IM_IR1	
0	IM_IR0	

viii.- RTR (Registro Tiempo de Espera) [R/W] [0x0017-0x0018][0x07D0]

Este registro establece el periodo de tiempo de espera. Un valor 1 equivale 100us. El valor inicial es 2000(0x07D0). Que es 200ms.

Ejemplo: Para configurar un valor de 400ms, 4000(0x0FA0).

0x0017 0x0018

0x0F	0xA0
------	------

Una retransmisión ocurrirá si no hay respuesta desde el punto remoto a los comandos de Conexión, Desconexión, Cierre, Enviar, o si la respuesta está tardando.

ix.- RCR (Registro Contador de Retransmisiones) [R/W] [0x0019][0x08]

Este registro establece el número de retransmisiones. Si ocurren mas retransmisiones que el número especificado por RCR, una interrupción TIMEOUT ocurrirá (siempre y cuando el bit correspondiente en el registro Sn_IR esté establecido en "1" lógico).

x.- RMSR (Registro Tamaño de Memoria RX)[R/W][0x001A][0x55]

Este registro asigna el total de 8K de memoria RX a cada socket.

7 6 5 4 3 2 1 0

Socket 3		Socket 2		Socket 1		Socket 0	
S1	S0	S1	S0	S1	S0	S1	S0

La combinación S1 S0 permite la asignación de la memoria según los valores siguientes:

Tabla B.5 Registro RMSR

S1	S0	Tamaño de Memoria
0	0	1 KB
0	1	2 KB
1	0	4 KB
1	1	8 KB

De acuerdo a los valores de S1 y S0, la memoria es asignada a los sockets desde el socket 0 dentro del rango de 8KB. Si no hubiese memoria suficiente para ser asignada, el socket no debe utilizarse. El valor por defecto es 0x55, asignando 2KB a cada uno de los sockets.

xi.- TMSR (Registro tamaño de Memoria TX)[R/W][0x001B][0x55]

Este registro es usado para asignar memoria TX a los sockets. La metodología es idéntica a la usada por el registro RMSR.

xii.- PATR (Registro Tipo de Autenticación en modo PPPoE)[R][0x001C-0x001D][0x0000]

Este registro notifica que método de autenticación acordado al establecerse una conexión con el servidor PPPoE. El W5100 soporta dos tipos de métodos de autenticación – PAP y CHAP.

Tabla B.6 Registro tipo Autenticación

Valor	Tipo de Autenticación
0xC023	PAP
0xC223	CHAP

xiii.- PTIMER (Registro Timer Petición LCP en PPP)[R/W][0x0028][0x28]

Este registro indica la duración de la petición LCP. Un valor de 1 corresponde a 25ms.

xiv.- PMAGIC (Registro Magic number LCP en PPP)[R/W][0x0029][0x00]

El registro es usado para la opción “Magic number” durante la negociación LCP. En conexiones ADSL.

xv.- UIPR (Registro Dirección IP destino inalcanzable)[R][0x002A-0x002D][0x00]

En el caso que se transmita usando UDP, si se produce la transmisión a una dirección IP inexistente, la dirección IP y el número de puerto serán almacenados en los registros UIPR y UPORT respectivamente.

xvi.- Sn_MR (Registro Modo Socket n)[R/W] [0x0400,0x0500,0x0600,0x0700] [0x00]

Este registro establece la opción de socket o tipo de protocolo para cada socket.

7	6	5	4	3	2	1	0
MULTI		ND/MC		P3	P2	P1	P0

Tabla B.7 Contenido registro Sn_MR

Bit	Símbolo	Descripción
7	MULTI	Multicasting 0: Deshabilitar Multicasting.

		1: Habilitar Multicasting. Es aplicable solo en caso de comunicación UDP.																																								
6																																										
5	ND/MC	No usar reconocimiento ACK retardado 0: Deshabilita opción ACK sin retardo 1: Habilita opción ACK sin retardo Es aplicable solo en modo TCP. Al habilitar el ACK se transmite cada vez que recibe paquetes de datos por parte del interlocutor. Deshabilitado el ACK es transmitido de acuerdo al mecanismo interno utilizando Timeout. Multicast 0: usa IGMP versión 2 ,1: usa IGMP versión 1																																								
4																																										
3	P3	Protocolo Establece el modo de comunicación TCP,UDP o IP RAW <table><tr><td>P3</td><td>P2</td><td>P1</td><td>P0</td><td>Modo</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Cerrado</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>TCP</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>UDP</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>IPRAW</td></tr></table> En caso de Socket 0 existen 2 modos adicionales <table><tr><td>P3</td><td>P2</td><td>P1</td><td>P0</td><td>Modo</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>MACRAW</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>PPPoE</td></tr></table>	P3	P2	P1	P0	Modo	0	0	0	0	Cerrado	0	0	0	1	TCP	0	0	1	0	UDP	0	0	1	1	IPRAW	P3	P2	P1	P0	Modo	0	1	0	0	MACRAW	0	1	0	1	PPPoE
P3	P2		P1	P0	Modo																																					
0	0		0	0	Cerrado																																					
0	0		0	1	TCP																																					
0	0		1	0	UDP																																					
0	0	1	1	IPRAW																																						
P3	P2	P1	P0	Modo																																						
0	1	0	0	MACRAW																																						
0	1	0	1	PPPoE																																						
2	P2																																									
1	P1																																									
0	P0																																									

xvii.- Sn_CR(Registro Comando Socket n) [R/W] [0x0401,0x0501,0x0601,0x0701] [0x00]

Estos registros son utilizados para ejecutar comandos en los sockets. Después de ejecutar los comandos dichos registros automáticamente son establecidos con valor 0x00.

Tabla B.8 Contenido Registro Sn_CR

Valor	Símbolo	Descripción
0x01	OPEN	Abrir Es usado para inicializar el socket.
0x02	LISTEN	Escucha

		Solo es usado en modo TCP. Permite poner el socket en modo de espera por un requerimiento de conexión realizado por un cliente.
0x04	CONNECT	Conectar Es usado solo en modo TCP. Envía un requerimiento de conexión a un punto remoto (Servidor TCP). Si la conexión falla, ocurre una interrupción Timeout.
0x08	DISCON	Desconectar Solo es usado en modo TCP. Envía un requerimiento de finalización de conexión.
0x10	CLOSE	Cerrar Es usado para cerrar el socket.
0x20	SEND	Enviar Transmite la data.
0x21	SEND_MAC	Es usado en modo UDP. Envía data según la MAC del destinatario.
0x22	SEND_KEEP	Solo es usado en modo TCP. Permite probar el estado de la conexión enviando 1 byte de data.
0x40	RECV	Se realiza la recepción de data.

xviii.- Sn_IR (Registro Interrupción Socket n)[R] [0x0402,0x0502,0x0602,0x0702] [0x00]

Este registro es utilizado para notificar el establecimiento y finalización de una conexión, recepción de data y notificación de tiempo máximo de espera.

7	6	5	4	3	2	1	0
			SEND_OK	TIMEOUT	RECV	DISCON	CON

xix.-Sn_SR (Registro Estado Socket n) [R] [0x0403,0x0503,0x0603,0x0703] [0x00]

Estos registros contienen un valor asociado al estado del socket n.

Tabla B.9 Registro Sn_SR

Valor	Símbolo	Descripción
0x00	SOCK_CLOSED	Socket cerrado
0x13	SOCK_INIT	Socket inicializado
0x14	SOCK_LISTEN	Socket en escucha
0x17	SOCK_ESTABLISHED	Conexión establecida

0x1C	SOCK_CLOSE_WAIT	Espera cierre
0x22	SOCK_UDP	Socket abierto en modo UDP
0x32	SOCK_IPRAW	Socket abierto en modo IPRAW
0x42	SOCK_MACRAW	Socket abierto en modo MACRAW
0x5F	SOCK_PPPOE	Socket abierto en modo PPPoE

Durante cambio de estados se definen:

Tabla B.10 Cambios de estado y valores asociados.

Valor	Símbolo	Descripción
0x15	SOCK_SYSENT	Luego del comando CONNECT antes de SOCK_ESTABLISH
0x16	SOCK_SYNRECV	En caso de conexiones entrantes antes de SOCK_ESTABLISH
0x18	SOCK_FIN_WAIT	En el proceso de terminación de una conexión muestra uno de estos valores. Si la terminación se realiza satisfactoriamente o se sobrepasa Timeout, cambia a SOCK_CLOSED.
0x1A	SOCK_CLOSING	
0x1B	SOCK_TIME_WAIT	
0x1D	SOCK_LAST_ACK	
0x11 0x21 0x31	SOCK_ARP	Mostrado cuando se realizan requerimientos ARP para solicitar dirección MAC remoto.

xx.- Sn_Port (Registro Puerto Fuente Socket n)[R/W][0x0404-0x0405,0x0504-0x0505, 0x0604-0x0605, 0x0704-0x0705][0x00]

Estos registros establecen el número de los puertos para cada socket cuando se aplican en modo TCP o UDP, y los valores necesarios antes de ejecutar el comando OPEN.

xxi.- Sn_DHAR(Registro Dirección Hardware Destino Socket n)[R/W][0x0406-0x040B, 0x0506-0x050B,0x0606-0x060B,0x0706-0x070B][0xFF]

Estos registros establecen la dirección MAC del destinatario de cada socket.

xxii.- Sn_DIPR (Registro IP destino Socket n)[R/W][0x040C-0x040F, 0x050C-0x050F, 0x060C-0x060F, 0x070C-0x070F][0x00]

Estos registros establecen la dirección IP de los destinatarios para cada socket.

xxiii.- Sn_DPORT (Registro Puerto Destino Socket n) [R/W][0x0410-0x0411, 0x0510-0x0511, 0x0610-0x0611, 0x0710-0x0711][0x00]

Este registro establece el número de puerto del destinatario para cada socket.

xxiv.- Sn_MSS (Registro Tamaño máximo de segmento Socket n) [R/W] [0x0412-0x0413, 0x0512-0x0513, 0x0612-0x0613, 0x0712-0x0713][0x0000]

Estos registros son usados para MSS(máximo tamaño de segmento) de TCP.

Ejemplo: En el caso de Socket 0 MSS=1460(0x05B4) tendremos:

0x0412	0x0413
0x05	0xB4

xxv.- Sn_PROTO (Registro Protocolo IP Socket n) [R/W] [0x0414, 0x0514, 0x0614, 0x0714] [0x00]

Este registro de Protocolo IP es usado para establecer el campo Protocolo de la cabecera IP en modo Layer RAW.

xxvi.- Sn_TX_FSR (Registro Tamaño Data Transmisión TX Socket n) [R] [0x0420-0x0421, 0x0520-0x0521, 0x0620-0x0621, 0x0720-0x0721][0x0800]

Estos registros notifican el tamaño de la data que el usuario puede transmitir. Para transmisión de data, el usuario primero debe checar este valor y controlar el tamaño de la data a transmitir.

xxvii.- Sn_TX_RR (Registro Lectura Puntero TX Socket n) [R] [0x0422-0x0423, 0x0522-0x0523, 0x0622-0x0623, 0x0722-0x0723][0x0000]

Estos registros muestran la dirección en la que la transmisión ha finalizado. Al utilizar el comando SEND para un socket n, este transmite la data desde la posición Sn_TX_RR a Sn_TX_WR y cambia automáticamente después que la transmisión ha finalizado, finalmente Sn_TX_RR y Sn_TX_WR tienen el mismo valor. Cuando se lee estos registros, debemos primero el byte superior (0x0422, 0x0522, 0x0622, 0x0722) luego el byte inferior en (0x0423, 0x0523, 0x0623, 0x0723) para obtener el valor completo.

xxviii.- Sn_TX_WR (Registro Puntero Escritura TX Socket n) [R/W] [0x0424-0x0425, 0x0524-0x0525, 0x0624-0x0625, 0x0724-0x0725] [0x0000]

Estos registros ofrecen información de la dirección donde se ha de escribir data. Los valores correspondientes.

ANEXO C

MODEM GPRS/GSM FASTRACK M1306B

MODEM GPRS/GSM FASTRACK M1306B

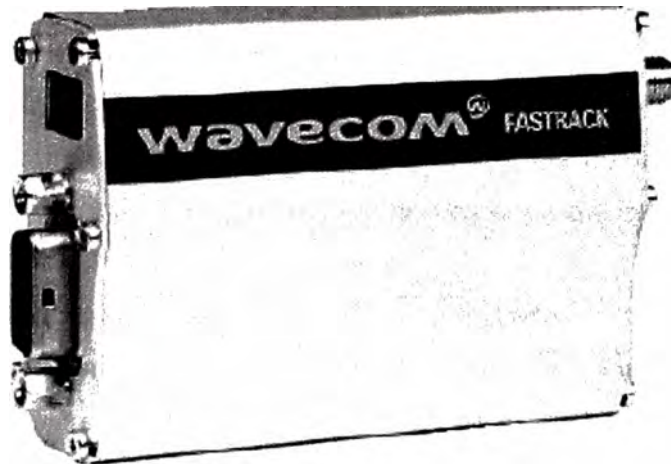


Figura C.1 Modem GSM/GPRS Fastrack M1306B (Fuente: Wavecom)

a.- Configuración del Modem GSM

Para configurar el M1306B se realiza las siguientes operaciones:

- Presionamos el seguro eyector de la bandeja contenedora del chip SIM.
- Insertamos el chip SIM en la bandeja contenedora, luego insertamos la bandeja hacia su posición normal.
- Conectamos la antena al conector SMA.
- Conectamos el cable serial de control.
- Conectamos la fuente de poder a la toma de energía.

El modem está listo para operar, posteriormente trataremos la descripción de los comandos AT para realizar la configuración.

b.- Revisión de la comunicación con el M1306B

Para checar la comunicación con el M1306B, realizaremos las siguientes operaciones:

- Conectaremos el puerto RS232 del modem (DCE) al puerto COM del ordenador (DTE).
- Configuramos la comunicación del puertos RS232 del DTE de la siguiente manera: 115200 bps, 8 bits data, No Paridad, 1 (Bit de parada), Flujo de Control por Hardware.

- Utilizando un programa de comunicación tal como el Hyperterminal de Windows, ingresar el comando AT↵. La respuesta del M1306B debe ser OK mostrada en la ventana del hyperterminal.
- Ejemplos de comandos AT que pueden utilizarse luego de inicializar el M1306B: AT+CGMI cuya respuesta "WAVECOM MODEM" indica que la conexión serial es correcta, AT+CPIN=xxxx para ingresar un código PIN (si está activado), AT+CSQ para verificar la intensidad de la señal recibida, ATD<Número telefónico> para iniciar una llamada, ATH para finalizar una llamada.

c.- Indicador de estado del M1306B

El M1306B cuenta con un indicador de estado, es un LED de color verde localizado en la parte frontal. La tabla siguiente muestra los posibles estados del led y su significado.

Tabla C.1 Indicadores de Estado Modem M1306B

LED de Estado	Actividad del LED	Estado M1306B
ENCENDIDO	LED Encendido permanente	Modem encendido pero no registrado en la red.
	LED Parpadeo Lento	Modem encendido, registrado en la red, no comunicación
	LED Parpadeo Rápido	Modem encendido, registrado en la red y comunicación en curso.
APAGADO	LED APAGADO	M1306B Apagado

d.- Verificación de la intensidad de la señal recibida

El modem M1306B establece una comunicación solamente si la señal recibida es suficientemente fuerte.

Para verificar la intensidad de la señal, debemos realizar las siguientes operaciones:

- Usando un software de comunicación tal como el Hyperterminal, ingrese el comando AT+CSQ.

La respuesta tiene el siguiente formato:

+CSQ: <rss>, <ver>; donde:

- <rss> = indicación de la intensidad de la señal recibida.
- <ber> = tasa de error del canal.

- Verificaremos el valor <rss> usando la tabla siguiente.

Tabla C.2 Valores Indicadores de Señal recibida

Valor <rss>	Significado
0 – 10	Insuficiente

11 – 31	Suficiente
32 – 98	No definido

e.- Verificación del Registro en la Red

Asegúrese que una tarjeta SIM válida esté insertada previamente en el compartimento del M1306B.

Usando un software de comunicación tal como el Hyperterminal, ingresar los comandos AT siguientes:

- AT+CPIN=xxxx para ingresar el PIN xxxx.
- AT+CREG? Para determinar es esto del registro.

El formato de respuesta es: +CREG: <modo>,<estado>

En la tabla siguiente se muestran los valores posibles:

Tabla C.3 Valores de Verificación de Registro

Valor Retornado<modo>,<estado>	Registro en la Red
+CREG: 0,0	No (No registrado)
+CREG: 0,1	Si (Registrado, red principal)
+CREG:0,5	Si (Registrado, roaming)

f.- Principales comandos AT

La tabla siguiente lista los principales comandos AT requeridos para inicializar el equipo inalámbrico.

Tabla C.4 Comandos AT Principales

Descripción	Comando AT	Respuesta Modem	Comentario
Ingresar PIN	AT+CPIN=xxxx (xxxx = PIN)	OK	PIN aceptado
		+CME ERROR: 16	PIN Incorrecto
		+CME ERROR: 3	PIN ya ingresado
Comprobación de registro en la Red.	AT+CREG?	+CREG:0,1	Equipo registrado en la red
		+CREG:0,2	Equipo no registrado en la red, intento de registro.
		+CREG:0,0	Equipo no registrado en la red

Recibiendo una llamada ingresante	ATA	OK	Responder la llamada
Iniciar una llamada	ATD<num. Telef>;	OK	Comunicación establecida
		+CME ERROR:11	Código PIN no ingresado
		+CME ERROR:3	Crédito excedido o una comunicación establecida
Iniciando una llamada de emerg.	ATD112;	OK	Comunicación establecida
Pérdida de comunicación.		NO CARRIER	
Finalizar llamada	ATH	OK	
Almacenar los parámetros en la memoria EEPROM	AT&W	OK	Los parámetros de config. han sido almacenados en la EEPROM

g.- Descripción Funcional

En la figura C.2 se muestra el diagrama de bloques del modem M1306B.

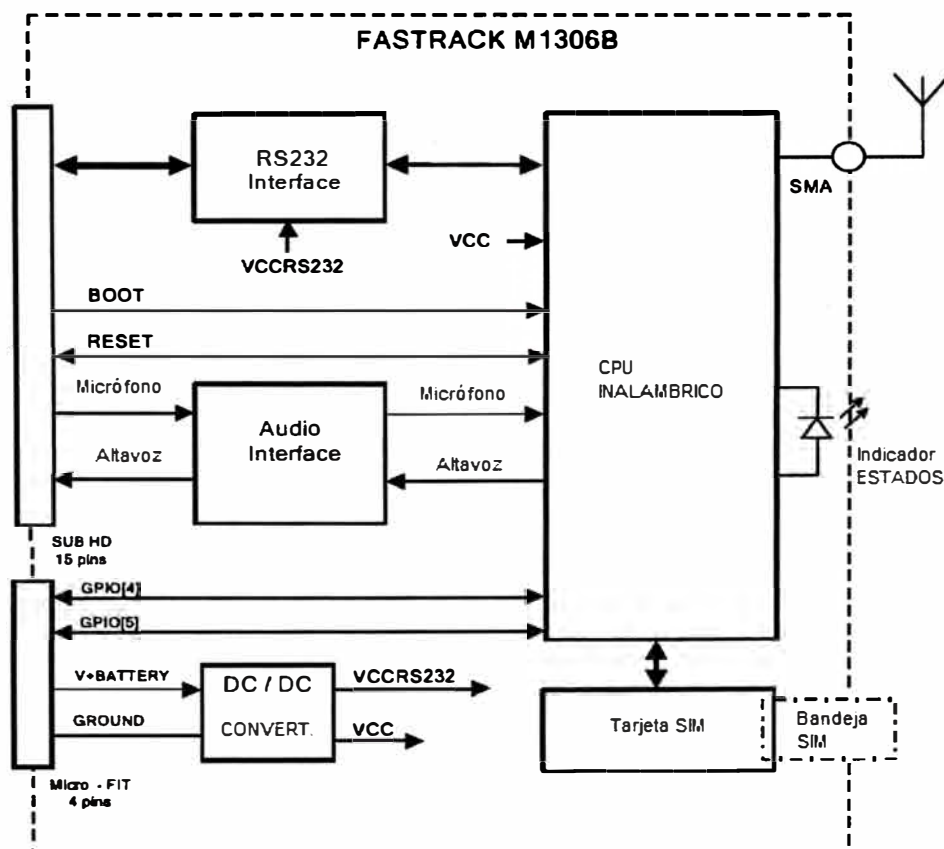


Figura C.2 Diagrama de bloques del modem GSM GPRS (Fuente: Wavecom)

g.1.- Fuente de Energía

El M1306B puede ser alimentado por una fuente externa de corriente continua (V+ BATTERY) desde +5.5V hasta +32V a 2.2A. La regulación es realizada por un convertidor interno DC/DC, que se encarga suministrar la energía necesaria para el funcionamiento del equipo.

El M1306B está protegido por un fusible especificado con un límite de 0.8A/250V.

g.2.- Enlace Serial RS232

La interface RS232 realiza la adaptación del nivel de voltaje ($V_{24}/\text{CMOS} \Leftrightarrow V_{24}/V_{28}$) entre el equipo inalámbrico (DCE) y el mundo externo (DTE).

La Interface RS232 está internamente protegida contra descargas electrostáticas en la línea RS232.

Las señales disponibles en un enlace serial RS232 son:

- TX data (CT 103/TX).
- RX data (CT 104/RX)
- Solicitud para enviar -Request To Send – (CT105/RTS).
- Listo para enviar – Clear To Send (CT106/CTS).
- Terminal Listo – Data Terminal Ready (CT108-2/DTR).
- Data Disponible – Data Set Ready (CT107/DSR).
- Detector de Portadora de Data – Data Carrier Detect (CT109/DCD)
- Indicador Ring (CT125/RI).

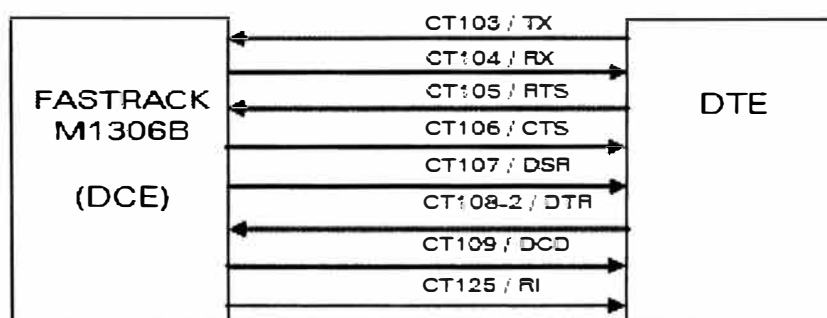


Figura C.3 Líneas de comunicación Modem – Terminal (Fuente: Wavecom)

La interface RS232 ha sido diseñada para permitir una flexibilidad en el uso de las señales correspondientes a la interface serial. Sin embargo, el uso de TX, RX, CTS y RTS son obligatorios, que no es el caso para las señales DTR, DSR, DCD y RI que pueden obviarse.

g.3.- Modo AutoBaudio.

El modo autobaudio permite al M1306B detectar la velocidad usada por el DTE

conectada al enlace serial RS232. De esta manera representa una forma de iniciar la comunicación sin previa configuración de parámetros.

BIBLIOGRAFIA

- [1] MICROCHIP PIC16F87XA Data Sheet 28/40/44-Enhanced Flash Microcontrollers. Edic. 2003. (Manual del Fabricante).
- [2] INTERFACING PIC MICROCONTROLLERS. Martin Bates. ELSEVIER NEWNES. Edic. 2006.
- [3] C Compiler Reference Manual February 2011. Manual CCS.
- [4] MICROCONTROLADORES FUNDAMENTOS Y APLICACIONES CON PIC. Fernando E. Valdés Pérez – Ramón Pallás Areny. ALFAOMEGA MARCOMBO. EDIC. 2007.
- [5] COMPILADOR C CCS Y SIMULADOR PROTEUS PARA MICROCONTROLADORES PIC. Eduardo García Breijo. ALFAOMEGA MARCOMBO. EDIC. 2008.
- [6] MANUAL DE USUARIO DEL COMPILADOR PCW DE CCS. C COMPILER FOR MICROCHIP PIC MICRO MCUS. Andrés Cánovas López.
- [7] W5100 DATASHEET WIZNET VERSION 1.1.2 EDIC. 2006 (Manual del Fabricante)
- [8] AT COMMANDS INTERFACE GUIDE FOR OPEN AT. REVISION 011. WAVECOM.
- [9] FASTTRACK M1306B USER GUIDE REVISION 003 EDICION 2006. WAVECOM.
- [10] GPRS GENERAL PACKET RADIO SERVICE. REGIS J. "BUD" BATES. McGraw-Hill TELECOM Edic 2004 .
- [11] VISUAL BASIC 6. G. Galeano Gil. ANAYA MULTIMEDIA, 1999.