



**Escola Politècnica Superior
de Castelldefels**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL DE FI DE CARRERA

TÍTOL DEL TFC: Proyecto ICARUS. Sistema de comunicación GPRS.

**TITULACIÓ: Enginyeria Tècnica de Telecomunicació, especialitat
Sistemes de Telecomunicació**

**AUTORS: Esunly Medina Medina
Natalia Revuelto Lorda**

DIRECTOR: Enric Pastor Llorens

DATA: 20 de febrer de 2006

Título: Proyecto ICARUS. Sistema de comunicación GPRS

Autores: Esunly Medina Medina
Natalia Revuelto Lorda

Director: Enric Pastor Llorens

Fecha: 20 de febrer de 2006

Resumen

El presente proyecto desarrolla el sistema de comunicación de un UAV (Unmanned Aerial Vehicle). Los UAVs son aviones no tripulados con aplicaciones tanto en el ámbito militar como civil.

El objetivo que pretendemos alcanzar es el desarrollo de un sistema para la transferencia de datos desde el UAV a través de la red GPRS, el cual se situará dentro del avión junto con los dispositivos de control del mismo.

Estudiamos la infraestructura de la red GPRS y los diferentes protocolos de software que nos permitirán establecer la comunicación. Además se diseña el hardware necesario para tal fin.

Este trabajo forma parte del proyecto ICARUS, en el cual se desarrollan diversos sistemas que se integrarán dentro de un UAV.

Title: ICARUS Project. GPRS communication system.

Authors: Esunly Medina Medina
Natalia Revuelto Lorda

Director: Enric Pastor Llorens

Date: February, 20th 2006

Overview

This project develops the communication system of a UAV (Unmanned Aerial Vehicle). UAVs are radio controlled devices and they can be applied at military environments as well as civil ones.

The main goal of this project is the development of a data transfer system for the UAV using the GPRS network. This system will be placed inside the plane together with the control unit.

GPRS network architecture and several software protocols needed to establish communication have been studied. There has also been designed the hardware interface.

This document is part of the ICARUS project which develops different systems to be set in a UAV.

“Porque yo también quisiera, con tiempo bueno o malo, poder ofrecer siempre algo agradable”

Bertold Brecht

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. INTRODUCCIÓN A LOS UAV Y SU SISTEMA DE COMUNICACIÓN.....	2
1.1. Aplicaciones de los UAV	2
1.2. Elementos que componen un UAV.....	3
1.3. Sistema de transmisión/recepción de datos	5
1.4. Organización de los capítulos sobre el sistema de comunicación.....	5
CAPÍTULO 2. GPRS	7
2.1. Introducción al capítulo	7
2.2. Tecnología GPRS.....	7
2.3. Transmisión de datos con GPRS.....	8
2.3.1. Conmutación de paquetes.....	8
2.3.2. Velocidad de transmisión	9
2.3.3. Conexión permanente	9
2.4. GPRS en Internet	10
2.5. La red GPRS.....	11
2.5.1. Infraestructura GSM	11
2.5.2. Arquitectura de la red GPRS	14
2.6. La estación móvil GPRS	16
2.6.1. Clasificación de los terminales	16
2.6.2. Partes del terminal móvil	16
2.6.3. Tarjeta SIM	17
2.6.4. Estados de funcionamiento	17
2.7. Velocidad de GPRS. Esquemas de codificación.....	19
CAPÍTULO 3. SOFTWARE.....	21
4.1. Introducción al capítulo	21
4.2. Comunicación entre DTE y DCE	22
4.2.1. Comandos AT.....	22
4.3. Comunicación entre DCE y GGSN.....	41
4.3.1. El protocolo PPP	42
4.3.2. Protocolos TCP/IP	47
CAPÍTULO 4. MO200	52
4.1. Introducción al MO200	52

4.2. Descripción de las funciones del MO200.....	54
4.2.1. LEDs	54
4.2.2. Audio.....	55
4.2.3. GPS	56
4.2.4. Transmisión de datos	56
4.2.5. Alimentación	57
4.2.6. Control de la energía	58
4.2.7. SIM	60
4.2.8. Enlace V24	62
4.2.9. Multiplexación DATA/COMMAND	63
4.2.10. Clocks	64
4.3. PINOUT	64
CAPÍTULO 5. DISEÑO DE LA PLACA	67
5.1. Introducción.....	67
5.2. Componentes.....	69
5.2.1. Conversores de nivel.....	69
5.2.2. Regulador de tensión	70
5.2.3. UART	73
5.2.4. Controlador Puerto Serie.....	75
5.2.5. Conector SIM.....	77
5.2.6. Puerta OR.....	79
5.2.7. Decodificador.....	80
5.2.8. Conectores	81
5.2.9. Módulo A9M9750	82
5.3. Esquemáticos de la placa.....	85
5.3.1. Conector AVX 20 5604 120 222 829 + Vbat	85
5.3.2. Conversor de nivel ADG3247	86
5.3.3. Conversor de nivel ADG3246.....	87
5.3.4. Dual UART.....	87
5.3.5. RS232 MAX3237E	89
5.3.6. DB9.....	90
5.3.7. Regulador	90
5.3.8. SIM	91
5.3.9. AMP 177984_5 (X1 y X2).....	91
5.3.10. Puerta OR.....	94
5.3.11. Decodificador 2-to-4	94
CONCLUSIONES Y LÍNEAS FUTURAS	95
BIBLIOGRAFÍA	96
ANEXOS	99
ANEXO 1: COMANDOS AT PARA GPRS.....	100
ANEXO 2: CÓDIGOS DE RESULTADO Y MENSAJES NO SOLICITADOS	110
ANEXO 3: CÓDIGOS ERROR DE GPRS	111
ANEXO 4: EJEMPLOS PRÁCTICOS DEL USO DE COMANDOS AT	112
ANEXO 5: PROCEDIMIENTOS DE ATTACH Y ACTIVACIÓN DE CONTEXTO PDP	116

INTRODUCCIÓN

Este trabajo se enmarca dentro de la línea de investigación del proyecto ICARUS, en el cual se desarrollan los sistemas que se embarcan dentro de un UAV.

Un UAV es un avión no tripulado que combina diferentes mecanismos encargados de pilotarlo. El UAV tiene diversas aplicaciones, normalmente en el ámbito militar; pero cada vez más se están desarrollando nuevas aplicaciones en el civil, dentro de las que se encuentra en proyecto ICARUS.

El presente documento desarrolla uno de los sistemas que componen el UAV de ICARUS. Se trata de un sistema de comunicación que emplea la tecnología GPRS para transmitir datos desde y hacia el UAV.

GPRS resulta una solución eficaz para implementar este sistema, debido a las ventajas que ofrece para la transmisión de datos y al coste razonable que esto supone.

El objetivo que pretendemos conseguir es el desarrollo del software y hardware necesarios para poder establecer la comunicación desde el UAV y se puedan enviar los datos utilizando la red GPRS.

Este proyecto se estructura en cinco capítulos que abarcan los conceptos sobre el UAV y sobre el sistema de comunicación del mismo que necesitaremos conocer. El capítulo 1, se presenta como una introducción a la tecnología de los UAV, sus aplicaciones y explica el contenido de los demás capítulos; los cuales se centrarán en el estudio del sistema de comunicación del UAV. En el capítulo 2, se explica el sistema GPRS y sus funcionalidades, en el capítulo 3, se desarrolla la parte del software de comunicación necesario y en los dos últimos se realiza el diseño de la interfaz hardware.

CAPÍTULO 1. INTRODUCCIÓN A LOS UAV Y SU SISTEMA DE COMUNICACIÓN

1.1. Aplicaciones de los UAV

Un **Unmanned Aerial Vehicle** (Fig. 1.1) es la expresión inglesa empleada para identificar aviones capaces de volar sin necesidad de piloto. Es decir, un sistema informático que combina sensores, GPS, servomecanismos y una CPU; todo ello encargado de pilotar el avión sin que sea necesario disponer de un piloto en la cabina. Los UAV se emplean principalmente en misiones de reconocimiento militar y el principal país que los emplea es los EEUU, que dispone de diferentes modelos de UAV de reconocimiento.

En el ámbito civil, los helicópteros y aviones tripulados tienen dos debilidades muy serias en aquellas aplicaciones que se clasifican como “*D-cube*” (Aburridas (Dull), Sucias (Dirty) o Peligrosas (Dangerous)): el alto coste de operación y que deben ser tripulados por seres humanos con las consecuencias y riesgos que eso conlleva en aplicaciones peligrosas. Estos riesgos se pueden minimizar utilizando los UAV’s, puesto que no necesitan tripulación alguna y su coste de construcción, operación y mantenimiento es reducido comparativamente.

Desde los años 90 hasta la actualidad los UAVs han desarrollado un papel importante en el campo militar como observadores de las fuerzas enemigas, pero también han ido apareciendo aplicaciones civiles, como puede ser la monitorización del medioambiente, substituyendo en algunos casos a la utilización de fotos de satélite.



Fig. 1.1 Un UAV en vuelo

Actualmente tan solo EEUU, Japón y Australia apuestan por esta tecnología no solo en un ámbito militar. EEUU utiliza variantes de los distintos UAVs disponibles por la NASA en distintos aspectos de la investigación científica.

Japón utiliza helicópteros-UAV para fumigar y sembrar campos de forma generalizada. Finalmente, Australia dispone de diversos UAV de uso civil que están siendo usados en diferentes programas de evaluación medio-ambiental.

Países como Suecia, Alemania, Israel y Holanda hace unos años que han comenzado la investigación dentro del campo de los UAVs puesto que supone uno de los pilares fundamentales del futuro de la aeronáutica. Tan sólo USA, Japón y Australia apuestan por esta tecnología en el ámbito civil, por ejemplo en USA el gasto público destinado a UAVs se incrementa más de un 30 % cada año, desde más de 300 millones de dólares en 2001 hasta 1761 millones en 2004.

Sin lugar a dudas los UAVs son una de las mayores líneas de investigación dentro del campo de la aeronáutica; tanto en lo que se refiere al mundo civil como al militar.

En el campo civil se plantean diversos ámbitos de aplicación. Por un lado se emplea para el control de fronteras en zonas remotas o en líneas de costa; principalmente para combatir el contrabando de drogas o de inmigrantes ilegales o soporte a servicios SAR (Search and Rescue). También se utiliza para el desarrollo de nuevos sistemas de comunicaciones, utilizando UAVs como estaciones repetidoras de largo alcance. Finalmente, para la supervisión medioambiental de zonas remotas, monitorización de cosechas extensivas, detección y control de incendios y el soporte al control/prevención de catástrofes.

Un ejemplo de aplicación del UAV en el ámbito civil, se encuentra en el proyecto ICARUS, actualmente desarrollado en el seno de un grupo de investigación. El objetivo de ICARUS es diseñar un sistema embarcado en un UAV, dedicado a la detección, control y análisis de incendios forestales. La información en tiempo real adquirida por la plataforma aérea, junto con las correspondientes herramientas de análisis permitiría tomar las decisiones estratégicas pertinentes para optimizar la lucha contra el fuego por parte de las autoridades competentes en su extinción. Este trabajo forma parte de dicho proyecto y se encargará del diseño de uno de los sistemas que estarán situados dentro del UAV.

1.2. Elementos que componen un UAV

Un UAV es un sistema complejo que debe incorporar toda una serie de elementos, que operando de forma conjunta, lo convierten en una plataforma de gran capacidad funcional. Los elementos clave de la plataforma UAV (Fig. 1.2) son:

- El fuselaje y motor del UAV.
- El sistema de guiado automático o "Flight Computer System" (FCS).
- El sistema de control del "payload".
- Plataformas de observación fijas o ventrales giro-estabilizadas.

- Sistemas sensor requeridos para mediciones atmosféricas y medioambientales.
- Sistemas de transmisión/recepción de datos.

El **fuselaje y motor del UAV** debe ser suficientemente grande como para albergar todos los sistemas y tener capacidad eléctrica para alimentarlos durante el tiempo preciso; suficientemente veloz para desplazarse a las zonas de operación de forma aceptable; y capaz de permanecer en operación por un margen de tiempo satisfactorio.

El **sistema de guiado** (FCS) es el corazón de los UAV. Estos sistemas son capaces de controlar los UAV de forma totalmente autónoma una vez se ha definido un plan de vuelo. Así mismo, el FCS integrado con un sensor **GPS** ofrecerá la precisión necesaria para poder localizar con exactitud la información adquirida durante el vuelo.

El **sistema de control del payload** debe ser capaz de controlar la operación de los distintos sensores (cámaras, instrumentos, etc.) que compongan el payload. Entre otros elementos se debe gestionar la direccionalidad de las cámaras, el almacenamiento de la información hasta su transmisión a la estación de procesado de datos, el encendido/apagado de los sensores, etc. Este sistema precisa trabajar con el **modulo de comunicaciones** del UAV, así como con el **FCS** para conocer en cada momento el comportamiento del UAV y operar los sensores en concordancia.

La **plataforma de observación giro-estabilizada** es el soporte sobre el cual situar la mayor parte de los sensores embarcados en el UAV. Esta plataforma debe permitir orientar los sensores hacia la zona de interés independientemente de la operativa del UAV, garantizando una correcta focalización de las imágenes.

Los **sensores de observación** son los que en realidad permiten la transmisión de imágenes en tiempo real. Éstos, incluirán la evaluación de cámaras de video CCD o CMOS, cámaras IR, escáneres termales, etc.

El **sistema de transmisión/recepción de datos** tiene dos funciones principales: mantener el contacto operativo entre la estación base y la plataforma UAV; y permitir la transferencia de datos/imágenes desde el UAV a la estación de procesado de datos. Relacionado directamente con el modulo de control del payload, su objetivo es permitir la transmisión en tiempo real de la información científica adquirida durante el vuelo.

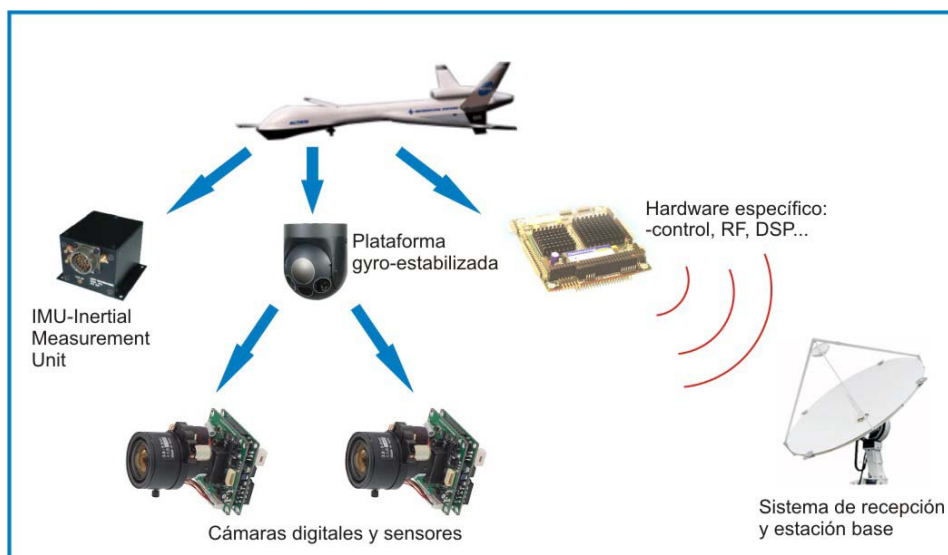


Fig. 1.2 Esquema de los componentes en un UAV

1.3. Sistema de transmisión/recepción de datos

Este proyecto se centra en el estudio y desarrollo del sistema de comunicaciones del UAV utilizando la tecnología GPRS. La justificación del empleo de GPRS, es que se trata de una alternativa aceptable para la transmisión de datos; porque a pesar de sus limitaciones, permite cumplir con los requisitos de manera asequible. Existen otras opciones que resultarían más eficientes para esta aplicación, como por ejemplo las comunicaciones vía satélite; pero debido al elevado costo de las mismas, no se trata de una alternativa viable.

El UAV normalmente vuela a una altura superior a la situación de las estaciones base GPRS, por lo que no recibirá la señal de las mismas. Al no tener cobertura a la altura normal de vuelo, el avión tendrá que descender para poder conectarse a la red GPRS.

Para establecer la comunicación se utiliza un módulo GPRS, que será controlado por un microprocesador. Se ha realizado el diseño de una interfaz hardware y se han estudiado los protocolos software necesarios para poder realizar las comunicaciones a través de dicho módulo.

1.4. Organización de los capítulos sobre el sistema de comunicación

En los capítulos siguientes se especifican las características del hardware y se estudian los aspectos software del sistema de comunicación GPRS. A continuación detallamos los aspectos que se estudiarán en cada uno de ellos.

En el capítulo 2 se resumen los aspectos principales de la tecnología GPRS. Se explica la arquitectura de red, las características de los terminales, la técnica empleada para la transmisión de los datos y las funcionalidades en general que hacen que GPRS sea apropiado para la conexión a Internet.

En el tercer capítulo se detallan todas las características del software de protocolos que intervienen en el proceso de comunicación entre el módulo y la web. Se diferencia entre la parte de comunicación ente el procesador y el módulo y la parte entre este último y el elemento que actúa como enlace con la red.

En el capítulo 4 se estudia en profundidad el módulo GPRS y se detallan todas sus funcionalidades; aunque algunas no se utilicen para implementar el sistema de comunicación.

Por último, en el Capítulo 5 se diseña la placa que conectará al microprocesador con el módulo GPRS y se estudian todos los dispositivos que la componen.

CAPÍTULO 2. GPRS

2.1. Introducción al capítulo

Este capítulo se presenta como una introducción a la tecnología de comunicaciones móviles que utilizaremos para la transmisión de información desde nuestro UAV, el GPRS. Tiene como objetivo principal intentar situarnos dentro del contexto en el que se desarrolla nuestro proyecto.

Pretendemos presentar una visión global del GPRS para que el lector que no esté familiarizado con el mismo, pueda tener los conocimientos básicos que le permitan entender mejor ciertos conceptos que presentaremos más adelante.

Además, mediante este capítulo intentamos justificar los motivos de elección del GPRS como alternativa para el desarrollo de la aplicación en la que se basa este trabajo.

2.2. Tecnología GPRS

GPRS (General Packet Radio Service) [1] [6] [7] [9] es un sistema de comunicaciones inalámbricas basado en la conmutación de paquetes. Pertenece a la denominada generación 2.5 (2.5G), por ser el resultado de la evolución de GSM (2G) y representar un paso hacia los sistemas 3G.

Es una tecnología superpuesta a la infraestructura GSM existente (Fig. 2.1), es decir, sólo requiere unas modificaciones sobre la red GSM (añadir nuevos elementos que funcionen como pasarelas específicas para la conexión de datos); por lo que continúa soportando los servicios de dicha tecnología y añade uno nuevo que permite la transmisión de paquetes vía radio, utilizando el protocolo IP. Esta característica ha hecho posible que GPRS permita la conexión a Internet.



Fig. 2.1 Tecnología GPRS

2.3. Transmisión de datos con GPRS

Aunque utilizando GSM ya es posible la transmisión de datos vía radio, veremos cómo ciertas características de esta tecnología la hacen ineficiente para tal fin y cómo GPRS se presenta como una alternativa más eficaz y económica.

2.3.1. Conmutación de paquetes

GSM utiliza la técnica de conmutación de circuitos [6] [7], que consiste en seleccionar en base a los dígitos marcados un camino de comunicación hasta el destino (como una llamada telefónica regular); por lo que los recursos quedan ocupados por un único usuario durante todo el tiempo que dura la comunicación. Debido a este modo de funcionamiento, la facturación se realiza en base al tiempo transcurrido.

GSM resulta adecuado para la transmisión de voz; debido a que cuando hablamos emitimos un flujo casi constante de sonidos y el canal de comunicación se utiliza casi en su totalidad. En cambio, los datos se transmiten a ráfagas, es decir, una gran cantidad de datos seguidos por periodos de silencio; por lo que si utilizamos conmutación de circuitos, el canal por el que se establece la comunicación estaría siendo utilizado de manera ineficaz, por debajo de lo que su capacidad permite.

La técnica de conmutación de paquetes [6] [7] utilizada en GPRS consiste en encaminar los paquetes hacia el destino a través de la red en función del tráfico existente en cada ruta. Los paquetes son el resultado de fragmentar las ráfagas de datos en trozos de longitud fija y añadirles una cabecera, que contiene las direcciones de origen y destino, entre otras cosas.

La ventaja de esta técnica es que los circuitos sólo se ocupan cuando se envían paquetes; por lo que el canal puede ser compartido por varios usuarios, a diferencia de la conmutación de circuitos, donde los abonados utilizan exclusivamente los recursos durante todo el tiempo que permanezca la comunicación. Por este motivo la tarificación se realiza en función de la cantidad de información enviada o recibida y no por el tiempo de conexión.

En la figura 2.2 se observa cómo mediante conmutación de circuitos se asigna un radiocanal (time slot) por usuario durante toda la comunicación; ya sea de voz o de datos. En el caso de la sesión de datos, también se utilizará siempre un time slot por usuario durante todo el tiempo que dure la sesión, sin tener en cuenta la cantidad de datos transmitidos. En cambio, dos usuarios de datos GPRS comparten los recursos radio (los time slots 5 y 6, aunque podrían haber compartido uno solo) y no los ocupan si no se están transfiriendo datos; por lo que otros usuarios que tengan las mismas características de uso, también podrían compartir los mismos slots.

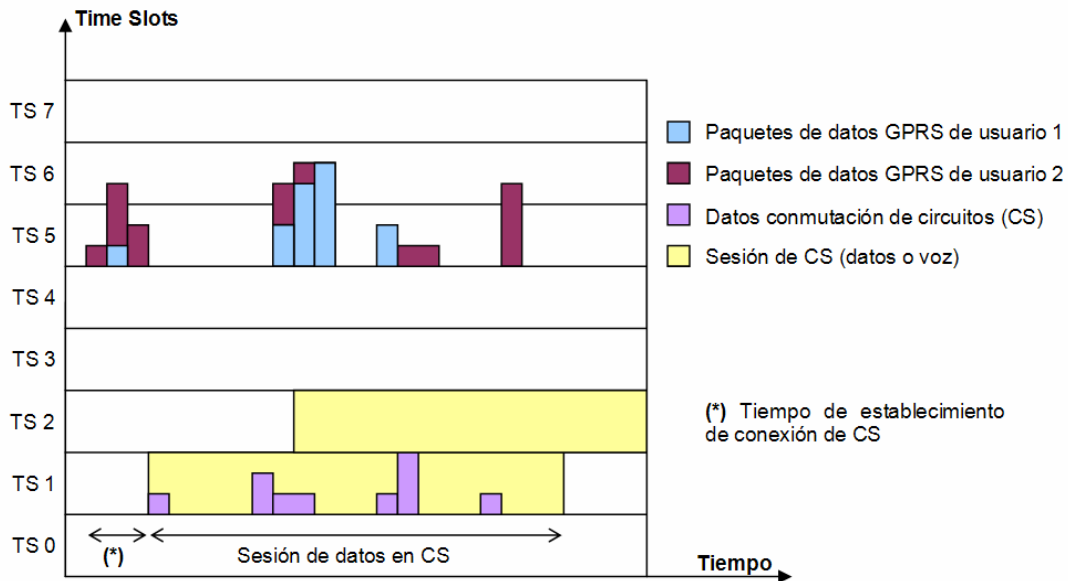


Fig. 2. 2 Conmutación de circuitos y conmutación de paquetes

2.3.2. Velocidad de transmisión

La conmutación de paquetes, las nuevas codificaciones de canal y el uso de multislot introducidos por GPRS permiten obtener mayores tasas de transmisión que con GSM. Con GPRS teóricamente podemos obtener una velocidad máxima de 171 Kbps, cuando con GSM era de 9,6 Kbps.

En GSM sólo se puede tener un canal asignado (timeslot), sin embargo, en GPRS, se pueden tener varios, tanto en el sentido de transmisión de móvil a estación base (enlace ascendente o uplink) como de la estación base al móvil (enlace descendente o downlink). La velocidad de transmisión aumentará con el número de canales asignados. Además, GPRS permite el uso de esquemas de codificación de datos que permiten una velocidad de transferencia de datos mayor que en GSM.

2.3.3. Conexión permanente

Una de las aportaciones más importantes de GPRS es permitir la posibilidad de que un terminal permanezca siempre conectado a la red. Propiedad que se conoce como "Always on" [6] [7].

Esta característica es posible, por un lado, a que ya que el terminal no hace uso de recursos de red mientras no esté recibiendo ni transmitiendo datos, no tiene que pagar por minuto; por lo que puede estar conectado todo el tiempo que desee.

Además, mientras que para GSM (y para todas las redes de conmutación de circuitos) es necesario un proceso de marcado dial-up para poder establecer el circuito de comunicación, para los usuarios GPRS la conexión siempre está disponible, no se necesita un procedimiento de establecimiento de conexión. Mientras el usuario GPRS puede iniciar una sesión e instantáneamente enviar y recibir datos, los usuarios GSM tienen que esperar un tiempo mientras se inicia la conexión. No está claro cuando empieza o termina una sesión GPRS porque normalmente dura todo el tiempo que el terminal permanece encendido.

2.4. GPRS en Internet

Actualmente tanto el número de usuarios de telefonía móvil como de Internet continúa aumentando cada día más. Además, la creciente necesidad de movilidad ha hecho que ambas tecnologías hayan tomado caminos paralelos y que tiendan a integrarse. Por esto, ha surgido el concepto conocido como "Internet móvil" [3] [6], el cual tiene como factor clave para su desarrollo el poder acceder a redes de conmutación de paquetes.

De hecho, una de las grandes ventajas de GPRS es que unifica el mundo de la telefonía móvil con la tecnología IP (Internet Protocol), como se observa en la figura 2.3; creándose una red superpuesta a la red GSM orientada a la transmisión de datos y que permite acceder directamente a los proveedores de contenidos de Internet (por eso también se le conoce como GSM-IP).

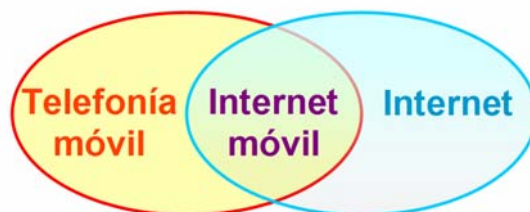


Fig. 2.3 Concepto Internet Móvil

Como hemos visto en los apartados anteriores, las características de GPRS lo hacen mucho más apropiado que GSM para acceder a Internet. No sólo el hecho de que se trate de una red de paquetes; sino también el aumento de la velocidad de transmisión, la característica de poder permanecer siempre conectado y el modo de tarificación.

Las redes de conmutación de circuitos no son apropiadas para sesiones de datos, donde es necesario garantizar cierta velocidad de transferencia y donde la cantidad de información enviada y recibida varía bastante (transmisiones intermitentes en forma de ráfagas, frecuentes de pequeño volumen o infrecuentes de volumen elevado).

No se puede navegar satisfactoriamente por Internet a una velocidad de máxima de 9,6 Kbps. Además, con GSM el usuario tiene que establecer una nueva conexión cuando quiere obtener o enviar datos y el coste resulta alto tanto para el usuario como para el operador de la red, porque los recursos radio son utilizados ineficientemente y se paga por tiempo de conexión. Todas estas deficiencias de GSM para la conexión a Internet fueron solucionadas con GPRS.

El hecho de que GPRS permita acceder en movilidad a todas las facilidades de Internet, usando al terminal como si fuera un módem, lo hace idóneo para la aplicación que describimos en este proyecto. Nuestro UAV podría enviar las imágenes captadas a través de Internet a donde fuera necesario. Sólo bastaría con acercarse a una antena de la red GPRS y descargar sus datos, que llegarán rápidamente a la persona que necesite utilizarlos.

2.5. La red GPRS

En este apartado describiremos brevemente la arquitectura de la red GPRS y explicaremos la función principal de cada uno de los elementos que la componen; pero para ello comenzaremos con la red GSM [1] [2] [9], al ser esta la base de GPRS.

2.5.1. Infraestructura GSM

A continuación se muestra un esquema de los elementos que componen la red GSM (Fig. 2.4). Todos ellos están presentes en la red GPRS; pero con ciertas modificaciones, consistentes en la actualización del software existente y en la implantación de nuevos elementos.

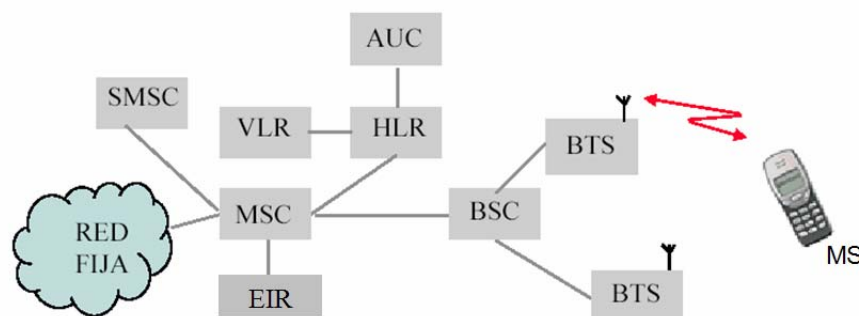


Fig. 2.4 Red GSM

2.5.1.1. Subsistema estación base

Al conjunto formado por un BSC y varios BTS (Fig. 2.5) se conoce como subsistema estación base (BSS: Base Station Subsystem) y agrupa la infraestructura específica de la parte radio.

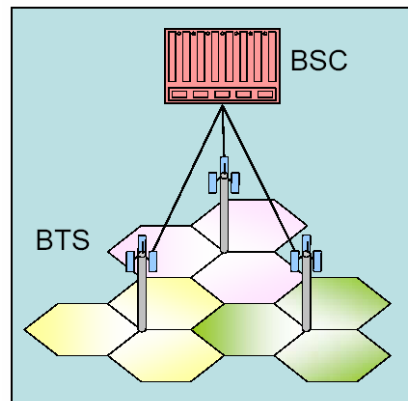


Fig. 2.5 Estructura del BSS

El controlador de estación base (**BSC**: Base Station Controller), como su nombre indica, es el encargado del control de los procesos que realizan en las estaciones base (BTS: Base Transceiver Station), es decir, de la administración de la interfaz radio y la gestión de sus recursos. Se encarga, entre otras cosas, de controlar las conexiones radio de los usuarios móviles, de la asignación y liberación de radiocanales, etc.

La **BTS** dispone de los dispositivos necesarios para la transmisión y recepción vía radio (radiocanales). Puede haber hasta un máximo de 16 radiocanales por estación base. La BTS es la encargada de tomar las medidas necesarias para asegurar la calidad de la comunicación.

2.5.1.2. Subsistema de red

El **HLR** o registro de localización de abonado (Home Location Register) es una base de datos central que proporciona la información de suscripción de los usuarios y los datos necesarios para poder establecer una llamada dirigida hacia un terminal móvil. Cada usuario debe estar registrado en un único HLR.

En la figura 2.6 se muestra la información almacenada en el HLR por cada uno de los usuarios.

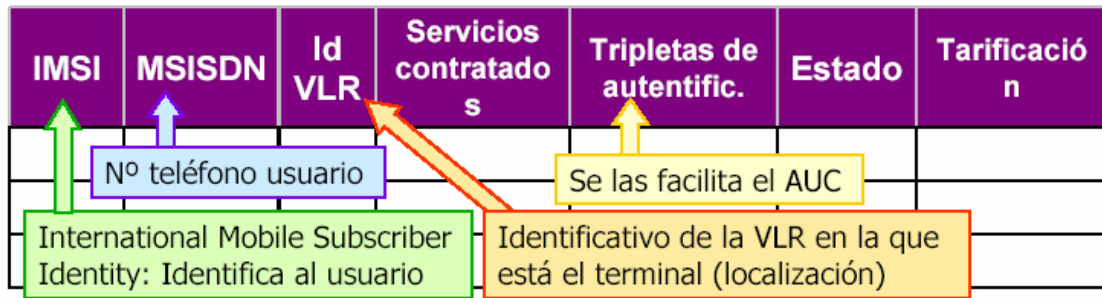


Fig. 2.6 Información por usuario contenida en el HLR

Existe otra base de datos controlada por el MSC, el registro de localización visitado (VLR: Visited Location Register); que contiene información relevante sobre los MT que están en el área de localización del VLR. Es decir, el VLR es una base de datos de los usuarios de una zona. La figura 2.7, muestra los datos de usuario que se encuentran en el VLR.

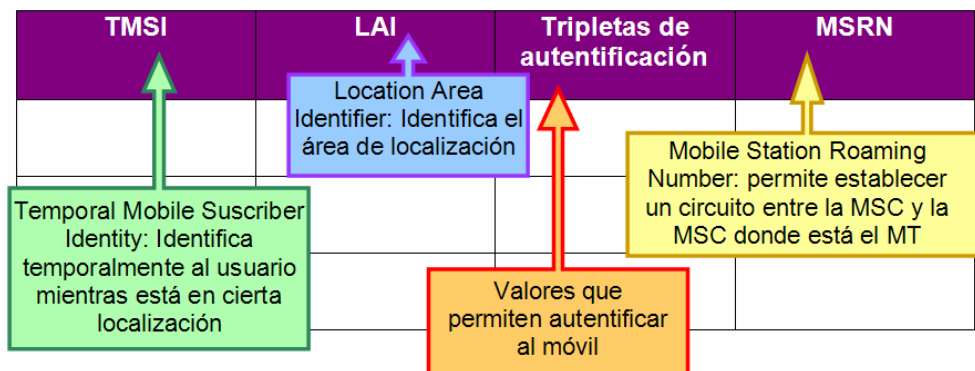


Fig. 2.7 Información contenida en el VLR

Otro elemento de la red GSM es el centro de conmutación o **MSC** (Mobile services Switching Center) que se encarga de realizar labores de conmutación dentro de la red y funciona como interfaz con otras redes de telecomunicaciones. El MSC agrupa a varios BSS y realiza funciones de gestión de movilidad de los MT, entre otras. Suele estar asociado a un VLR dando soporte a los procesos de llamada y autenticación.

Cuando un terminal entra a la red, el par MSC/VLR pregunta al HLR los datos del usuario.

Si el MSC actúa como interfaz con la red telefónica conmutada (RTC) se habla de GMSC (Gateway MSC). Un gateway se encarga de interconectar dos redes haciendo que los protocolos de comunicaciones que existen en ambas redes se entiendan.

El centro de autenticación o **AUC** gestiona los datos de seguridad y autenticación de los abonados. Es el encargado de entregar al HLR las tripletas de autenticación.

El registro de identificación de equipos (**EIR**) es una base de datos mundial que tiene la función de validar a los equipos. Utiliza el IMEI (identificador internacional de equipo móvil) para asegurar que un terminal es un equipo válido. Al recibir el IMEI, el EIR consulta tres listas donde aparecen los equipos ubicados en los diferentes países participantes en GSM, los equipos que se encuentran restringidos y los que se encuentran dañados o no aprobados.

El **SMSC** (centro de servicio de mensajería) permite enviar y recibir mensajes cortos entre móviles y puede permitir interconexión con otros sistemas de correo electrónico. Es independiente de la red GSM y puede estar conectado a varias redes GSM simultáneamente.

Como podemos observar en la figura 2.8, al conjunto formado por el SMSC, el EIR, el par MSC/VLR, el HLR y el AUC se le conoce como subsistema de red (NSS).

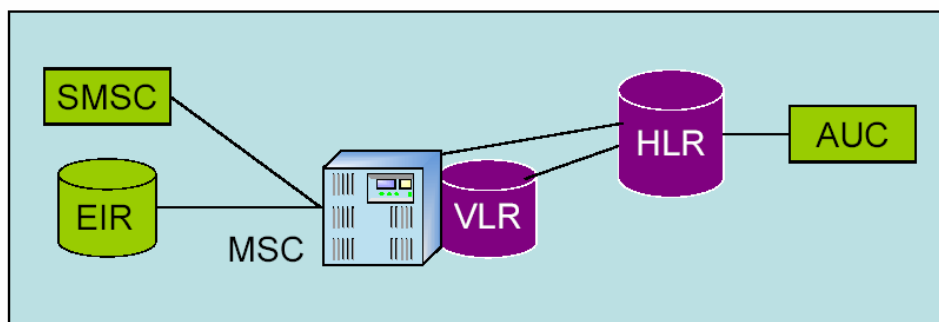


Fig. 2.8 Subsistema de Red

2.5.2. Arquitectura de la red GPRS

Como comentamos en párrafos anteriores, la red GPRS [1] [2] [9] utiliza la infraestructura de la red GSM existente pero introduce nuevos elementos y realiza ciertas modificaciones en los existentes.

Tanto a las bases de datos existentes como a las BTS se realiza una actualización del software para poder soportar los nuevos modelos de llamada y las nuevas funcionalidades.

La infraestructura de la BSS sigue siendo la misma pero a nivel de BSC se introduce el **PCU** (Packet Control Unit). La adición de este elemento viene como consecuencia directa del hecho de que GPRS sea una estructura superpuesta a la red GSM; ya que se tienen que separar los diferentes flujos de

datos y dirigirlos hacia la red correcta. Es decir, el PCU es el encargado de encaminar los diferentes tipos de tráfico. Si se trata de voz, lo encaminará hacia la MSC (red GSM) y si son datos los envía hacia la red de conmutación de paquetes específica de GPRS.

GPRS introduce dos nuevos nodos (el equivalente a las centrales de conmutación) a la red: el **GGSN** (Gateway GPRS Support Node) y el **SGSN** (Serving GPRS Support Node).

El GGSN actúa como interfaz lógico entre la red GPRS y las redes de paquetes de datos externas. Realiza funciones de encaminamiento de los paquetes de usuario, mantenimiento de sus datos, de tarificación y de seguridad, al permitir la introducción de elementos que protejan al sistema GPRS.

El SGSN es el responsable de la entrega de paquetes al terminal en su área de localización. Básicamente es un conmutador de paquetes GPRS que proporciona funciones parecidas a las realizadas por el MSC. Realiza funciones de autenticación, cifrado, control de acceso a la red, gestión de movilidad, etc.

También está encargado de establecer el proceso de "Tunneling" que consiste en encaminar correctamente los paquetes de datos de usuario entre SGSN y en GGSN.

En la figura 2.9 se muestra la arquitectura de la red GPRS y se destacan los cambios introducidos respecto a la estructura GSM.

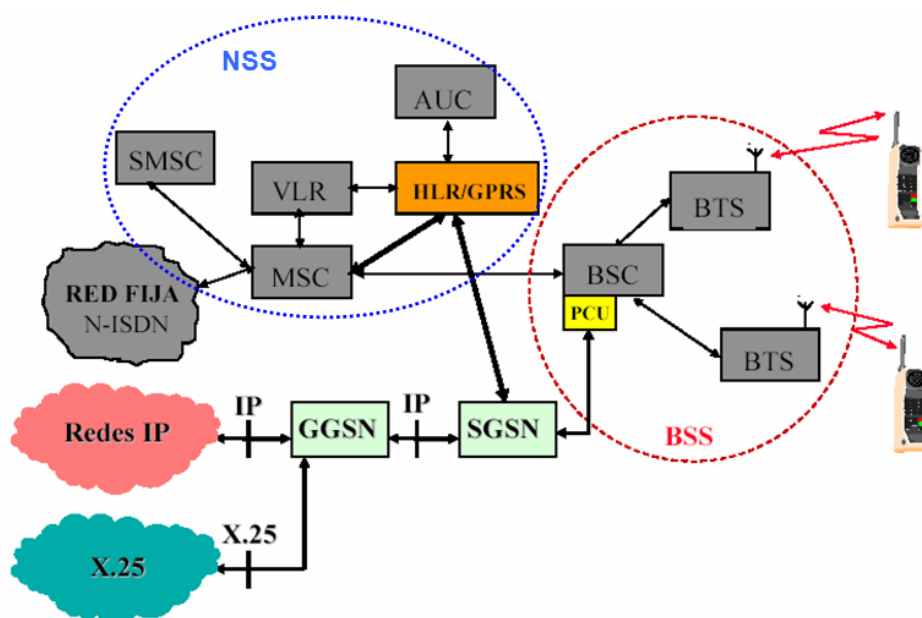


Fig. 2.9 Red GPRS

2.6. La estación móvil GPRS

En GPRS la estación móvil (MS: Mobile Station), también conocida como terminal móvil, es totalmente distinta que en GSM para poder soportar los nuevos servicios que se ofrecen. Los terminales GPRS pueden ser de tres tipos en función de si soportan simultáneamente servicios GSM y GPRS o no.

2.6.1. Clasificación de los terminales

Los terminales [6] [7] de **Clase A** permiten el uso simultáneo de GSM y GPRS. Los equipos permiten enviar y/o recibir datos y voz al mismo tiempo sin que haya degradación de la calidad de ninguno de los dos servicios. Por este motivo estos terminales resultan caros de fabricar.

Los de **Clase B** por su parte, monitorizan los canales GSM y GPRS a la vez pero no pueden establecer una comunicación simultánea. Es decir, registran tanto GSM como GPRS pero uno de los dos permanece en suspensión mientras el otro está activo. GSM tiene la prioridad, por lo que la calidad del servicio GPRS se ve reducida (degradación de QoS: Quality of Service).

Finalmente, en los de **Clase C** el usuario debe seleccionar el tipo de servicio al que quiere conectarse. En ningún caso puede hacerse uso simultáneo de GPRS y GSM.

2.6.2. Partes del terminal móvil

Para nuestro proyecto resulta imprescindible distinguir entre dos partes del terminal: El terminal móvil (MT: Mobile Terminal) y el equipo terminal (TE: Terminal Equipment) [7] [18].

El **TE** es la parte que funciona como plataforma de aplicación, es el dispositivo donde se ejecutan los servicios que se ofrecen al usuario. A lo largo de este documento también se le llama **DTE** (Data Terminal Equipment).

El **MT** contiene la parte “módem” de la estación, que le permite conectarse a Internet. Es la parte que se comunica con la red.

El terminal móvil y el equipo terminal están físicamente separados y debe establecerse algún tipo de conexión entre ellos. En nuestro caso será una conexión física RS232. En el capítulo “Software” veremos con mayor detalle estos conceptos.

2.6.3. Tarjeta SIM

Un elemento importante de los terminales es la SIM [9] (Subscriber Identity Module). Es una tarjeta que permite personalizar al terminal. En ella se encuentran los parámetros identificativos del usuario y los tipos de servicios contratados. Esta tarjeta se inserta en el interior del móvil, permitiendo al usuario el acceso a la red.

Contiene los números de seguridad para evitar el uso fraudulento del terminal móvil. Dichos números se detallan en la figura 2.10.

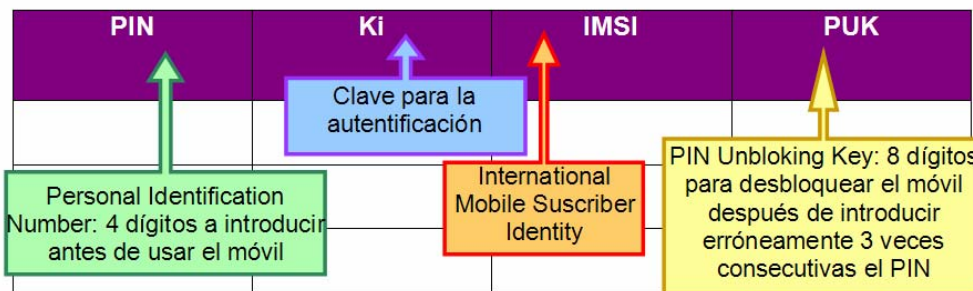


Fig. 2.10 Números contenidos en la SIM

2.6.4. Estados de funcionamiento

Los requerimientos de nuestro proyecto nos hacen tener muy en cuenta el aspecto de la movilidad. El UAV se estará desplazando continuamente, mientras recopila información, y luego para descargarla en alguna antena; por lo que una de las mayores limitaciones que se le pueden presentar es el tiempo de duración de la batería.

Para que nuestra aplicación funcione de manera satisfactoria debemos asegurar un ciclo de descarga elevado y una calidad permanente de la energía suministrada mientras haya carga. De no ser así, nuestro avión se quedaría en poco tiempo sin la posibilidad de enviar los datos almacenados; por lo que realmente no tendría ningún sentido utilizarla para este fin.

Además, los sistemas que transmiten datos consumen más batería que los de voz. Especialmente la transmisión de imágenes, que exige mucho proceso en el terminal; por lo que incrementa notablemente el consumo de batería.

El problema se encuentra en todas las funcionalidades de la red que exijan mucho proceso en el móvil, especialmente para aquellos que están conectados permanentemente a la red (característica "Always on" de GPRS). Por esto se deben buscar procedimientos que reduzcan dicho consumo.

Los terminales GPRS tienen la cualidad de tener un consumo de batería bajo a pesar de ofrecer una alta funcionalidad. Esto se debe a que pueden trabajar en tres estados o modos de actividad, conmutando entre ellos, lo que les permite ahorrar energía. Los estados son: “Inactivo”, “Listo” y “En espera” (Idle, Ready y Standby) [6] [7] [18]; tal como podemos observar en la figura 2.11.

Cuando el terminal se conecta a la red, cambia su estado de Inactivo a Listo. En este estado, está preparado para enviar y recibir paquetes. Si no se transfieren paquetes durante un periodo de tiempo, expira un temporizador que hace al terminal situarse en Standby (En espera). El dispositivo permanece en este modo hasta que envíe o reciba paquetes o hasta que expire otro temporizador que lo hace volver a Inactivo.

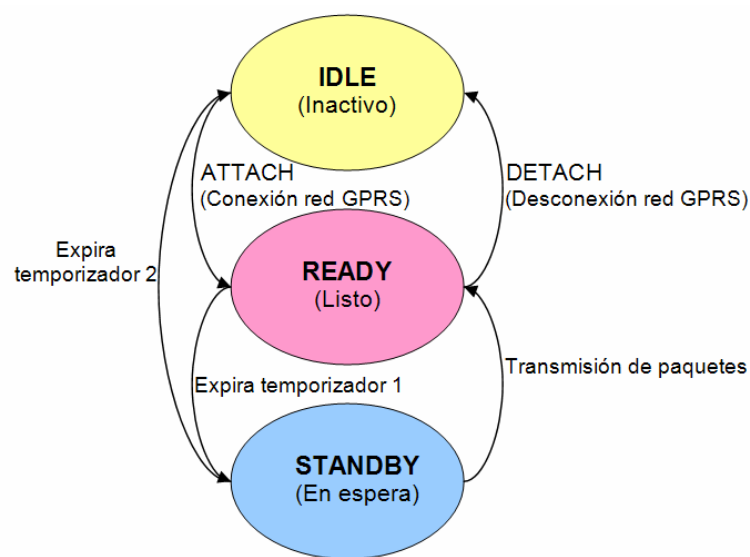


Fig. 2.11 Estados de funcionamiento de terminales GPRS

La importancia de estos diferentes estados es que permiten a la estación móvil mantener un consumo de energía bajo mientras no se está comunicando activamente. Esta propiedad es de gran importancia para el caso del servicio que queremos desarrollar; ya que permite que el avión pueda mantener la comunicación por más tiempo y así enviar un mayor número de imágenes, al minimizar la comunicación con el sistema en la medida de lo posible.

Si el terminal se encuentra en modo Listo, lo que significa que está enviando o recibiendo datos, debe mantener continuamente la comunicación con el SGSN para poder saber el camino que deben seguir los paquetes. Esto significa el envío de mensajes al SGSN indicándole en cada momento cual es su localización, lo que requiere procesado por parte del móvil y por tanto consumo importante de batería.

Cuando se encuentra en Standby, el móvil no está continuamente enviando o recibiendo información y ha estado inactivo por algún tiempo. El SGSN no tiene que saber exactamente donde se encuentra el terminal; pero sí tener una idea

de dónde está. El móvil no necesita enviar ninguna actualización de posición si se mueve dentro de una determinada área (routing area), lo que minimiza el consumo de carga.

2.7. Velocidad de GPRS. Esquemas de codificación

Un factor importante a considerar cuando se trata de servicios de datos, es la velocidad de transmisión de los mismos. En GPRS los principales factores que determinan dicha velocidad son: el número de usuarios existentes en la celda en la que se realiza la conexión, el número de timeslots que soporta el terminal y el proceso de codificación empleado.

En cuanto a la codificación, podemos decir que en GPRS existen cuatro esquemas de codificación diferentes, que afectarán significativamente a la tasa de transferencia de los datos y que añaden integridad a los mismos.

Los códigos hacen posible la recuperación de paquetes de datos incluso si se pierden bits en la interfaz radio. En función del código empleado habrá una mayor o menor posibilidad de recuperar la información, es decir, proporcionan diferentes grados de robustez frente a la pérdida de bits.

Por otro lado, hay que considerar que existe una relación inversa entre la velocidad y la seguridad de integridad de los datos; por tanto, para poder garantizar una robustez elevada, será necesario añadir mayor cantidad de información de protección; por lo que la velocidad de transferencia será menor. Debemos tener en cuenta que ante diversas condiciones de la interfaz radio se requerirá una determinada fortaleza del código para garantizar la recuperación de los datos. Es decir, debemos usar códigos más robustos cuando la calidad de la señal es baja que cuando tenemos mejores condiciones de la interfaz radio.

Los diferentes esquemas de codificación de canal [1] [7] [9] definidos en los estándares GPRS proporcionan, por tanto, diferentes niveles de integridad y tasas de transmisión variables. En la tabla 2.1, mostrada a continuación, se encuentran los cuatro tipos de códigos GPRS ordenados de los que proporcionan mayor robustez y velocidades menores a los que son menos robustos pero tienen tasas mayores.

Tabla 2.1 Esquemas de codificación GPRS

Código	Bits útiles	Tasa de datos (Kbps)
CS-1	181	9,05
CS-2	268	13,4
CS-3	312	15,6
CS-4	428	21,4

Como podemos comprobar en la tabla, el código CS-1 es el más fuerte, el que tiene menor número de bits útiles y el de menor velocidad; en cambio, el CS-4 es menos robusto, proporciona más velocidad y tiene mayor cantidad de carga útil. Podemos concluir que hay una relación inversa entre la cantidad de información útil de los bloques de datos y el grado de protección de la integridad de los mismos.

CAPÍTULO 3. SOFTWARE

4.1. Introducción al capítulo

En este capítulo estudiaremos los fundamentos del software que harán posible establecer la comunicación entre el DTE e Internet. Nosotros sólo nos centraremos en los mecanismos de comunicación existentes entre el DTE y el GGSN, que es el elemento encargado de conectar la red GPRS con Internet.

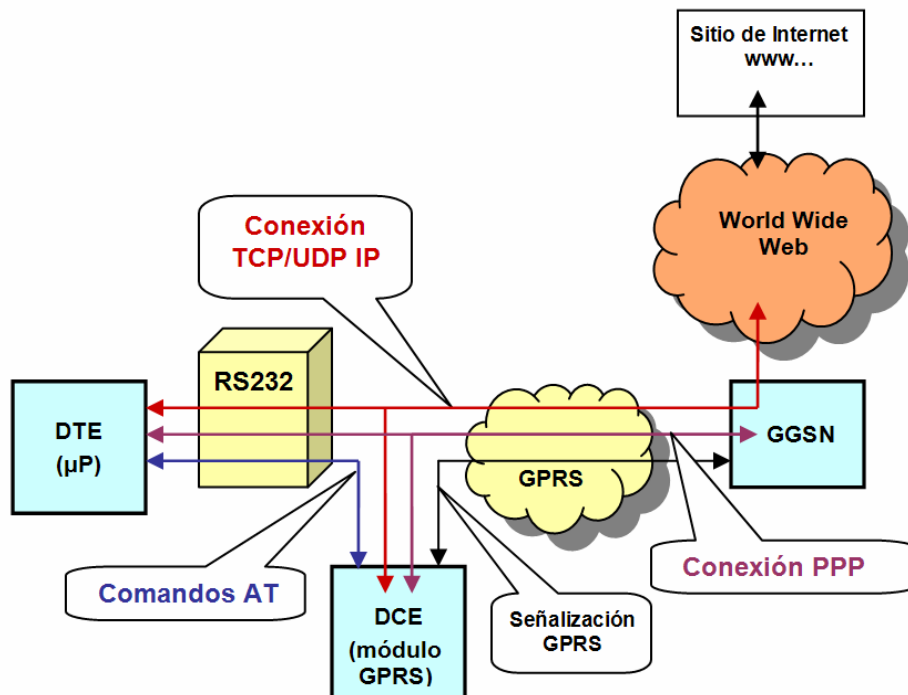


Fig. 3.1 Comunicación entre DTE e Internet

La figura 3.1 muestra una visión general de los enlaces y protocolos que permiten la conexión a Internet [7] [13]. Podemos distinguir entre tres tipos de enlaces: físicos, punto a punto y TCP/UDP IP.

En el caso de los enlaces físicos, observamos que el terminal se conecta al DCE usando una conexión física RS232 y que éste se conecta al GGSN usando un enlace radio GPRS.

Respecto a los enlaces punto a punto, vemos que para transferir datos entre el terminal y el DCE se utiliza el protocolo de comandos AT; además el DCE se conecta al GGSN usando el protocolo PPP.

Por último, el DCE puede transferir datos con la web usando tanto el protocolo TCP/IP como UDP/IP.

4.2. Comunicación entre DTE y DCE

Como habíamos explicado en el capítulo anterior, las estaciones móviles constan de dos partes bien diferenciadas: el TE y el MT. Recordemos que el equipo terminal (TE) es la parte donde corre la aplicación, es decir, el procesador donde se ejecuta el software de comunicación. Y el terminal móvil (MT) es la parte módem, es decir, la que adapta los datos para poder comunicarse con la red. Nosotros utilizaremos como TE un microprocesador de NetSilicon, el NS9750 y como MT el módulo GPRS MO200, de SAGEM.

Un aspecto clave del presente proyecto es el estudio de la interfaz existente entre ambas partes; ya que lo que pretendemos es conseguir que nuestro módulo GPRS se comunique con la red inalámbrica y para ello debemos conseguir establecer una comunicación entre él y el micro. Por tanto, debemos lograr que el módulo entienda y ejecute las instrucciones que reciba de parte del procesador.

Para conseguir establecer esta comunicación el TE y el MT deben conectarse físicamente a través de un enlace serie que en nuestro caso será una interfaz V.24 o RS232 [18]. Además es necesario que haya un intercambio de mensajes a través de este medio que permita un entendimiento entre los dispositivos, esto se realizará con el protocolo de comandos AT. En la figura 3.2 se muestran los elementos que intervienen en el proceso de comunicación entre el DTE y el DCE.

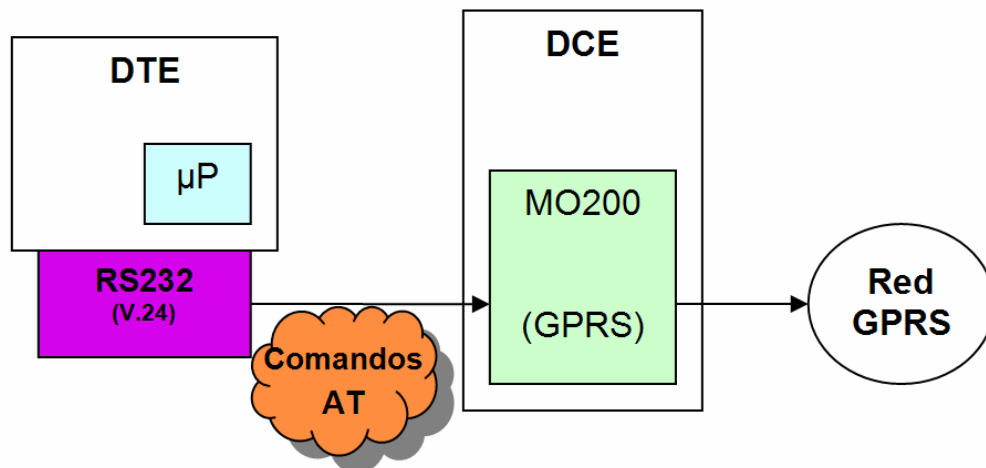


Fig. 3.2 Comunicación entre el DTE y el DCE a través del RS232 utilizando comandos AT

4.2.1. Comandos AT

Para poder acceder al hardware del terminal y a las propiedades de red, los estándares GPRS definen ciertas instrucciones, denominadas “comandos de atención” (AT) [7] [10] [13] [14] [15]. Los comandos AT son un grupo de

órdenes enviadas por el microprocesador del TE, que crean un lenguaje de mando que permite controlar al MT.

El estándar de la **ETSI GSM 07.07** [16] describe los comandos AT (también conocidos como “instrucciones Hayes”) establecidos para terminales GSM, incluyendo GPRS.

En las especificaciones, se divide al MT en adaptador de terminal (TA) y equipo móvil (ME), tal como se observa en la figura 3.3 [7]. El TA es el elemento del MT que recibe e interpreta los comandos AT y el ME es la parte que se conecta directamente con la red.

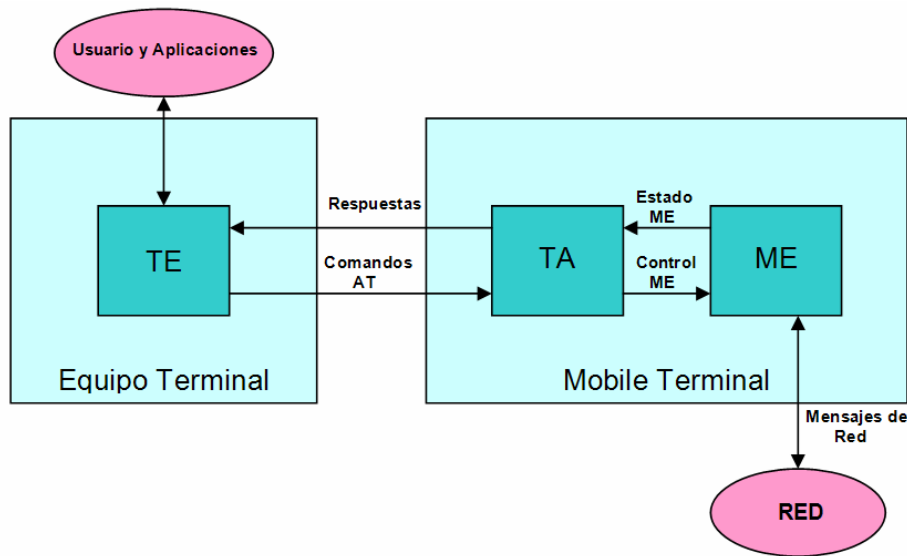


Fig. 3.3 Acceso al MT mediante comandos AT

En las hojas de características del fabricante del módulo GPRS [18] [19], al TE le llama **DTE** (Data Terminal Equipment) y al módulo en sí, se le denomina **DCE** (Data Communication Equipment).

3.2.1.1. Multiplexado entre datos y comandos: Modos de operación

A través de la interfaz que conecta al DTE y al DCE es necesario enviar dos tipos diferentes de información: Comandos AT y los propios paquetes de datos PPP [7] [12] [18]; es decir, los datos captados por nuestro avión, procesados en el DTE y que deben ser enviados a la red.

Es importante separar estos dos flujos de información. Además, no se pueden mezclar; debido a que no disponemos de suficiente ancho de banda para transmitirlos simultáneamente por el mismo canal. Por este motivo, el enlace serie debe funcionar en dos diferentes estados o modos de operación [18].

Se dice que el enlace opera en **modo comando** cuando se reserva para la transmisión de comandos AT y que lo hace en **modo datos** cuando por él se transmite exclusivamente el flujo de datos. Cuando nos encontramos en modo comando, el DCE, es decir nuestro módulo, interpreta cualquier información que recibe del DTE como comandos y no como datos.

Es importante definir un procedimiento para conmutar entre estos dos estados de funcionamiento; porque puede darse el caso de que durante la conexión de datos, se necesite enviar ciertos comandos AT para notificar al otro extremo un evento importante.

La primera alternativa, provista por el estándar GSM, es usar los comandos “+++” y “ATO”, tal como se refleja la figura 3.4. Ésta es una solución muy simple; pero tiene la contrapartida de permitir sólo al DTE controlar el cambio entre los dos modos; por lo que normalmente se usa sólo para finalizar una conexión de datos.

Si el módulo está en una conexión de datos, la orden “+++” lo hace cambiar a modo comando. De esta forma se pueden enviar comandos AT sin necesidad de terminar dicha conexión; a la que se puede volver nuevamente, escribiendo la instrucción “ATO”.

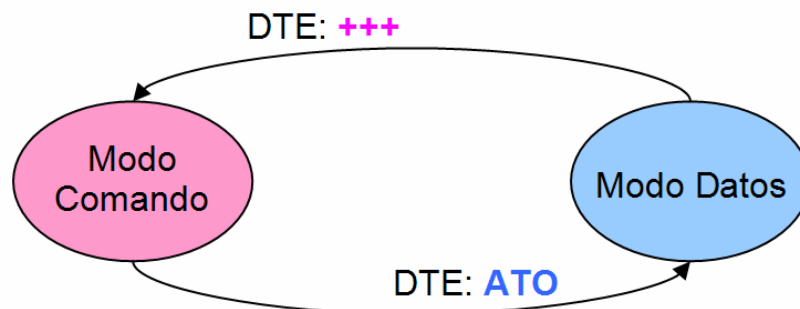


Fig. 3.4 Conmutación entre modo comandos y modo datos

Una segunda opción consiste en implementar el estándar **GSM 07.10** [17], que proporciona una interfaz de multiplexado; pero implica el desarrollo de un software para ejecutar en el DTE (driver). El fabricante del módulo que utilizaremos recomienda esta opción.

Existe un tercer método, desarrollado por SAGEM S.A. [18] que consiste en crear un tercer estado para el enlace serie y gestionarlo mediante señales hardware. Como observamos en la figura 3.5, este modo se llama “suspended data mode” (**modo datos suspendido**) y es usado temporalmente para enviar comandos AT.

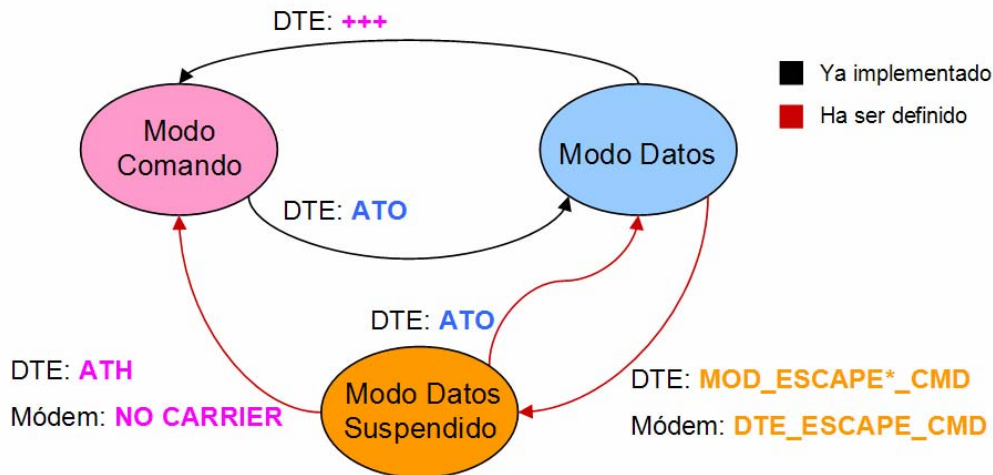


Fig. 3.5 Diagrama de estados del multiplexado entre datos y comandos mediante señales hardware

3.2.1.2. Protocolo de funcionamiento

En este apartado explicaremos brevemente la metodología de intercambio de mensajes entre los dispositivos que queremos comunicar, es decir, el protocolo de comandos AT [13].

El módulo se activa por medio de la recepción de instrucciones AT, cuyo valor subsiguiente modifica la ejecución de las mismas. Los mensajes enviados por el DCE proporcionan información sobre la ejecución de las instrucciones.

La comunicación casi siempre la inicia el DTE; lo que significa que la petición de todos los servicios debe realizarlas el terminal mediante el envío de comandos AT. Cada comando debe ser respondido con un “código de resultados”, que será enviado por el módulo hacia el DTE.

Los mensajes “indicador” son enviados asíncronamente y son generados debido a que algunos comandos inician un modo en el que se generan eventos específicos en el DCE.

Algunos comandos pueden incluir varias peticiones “respuesta”, para enviar datos desde el módulo hacia el terminal. El DCE también puede repetir los comandos recibidos y enviarlos de vuelta hacia el DTE.

En la figura 3.6, se muestra la secuencia normal de mensajes intercambiados entre el DTE y el DCE.

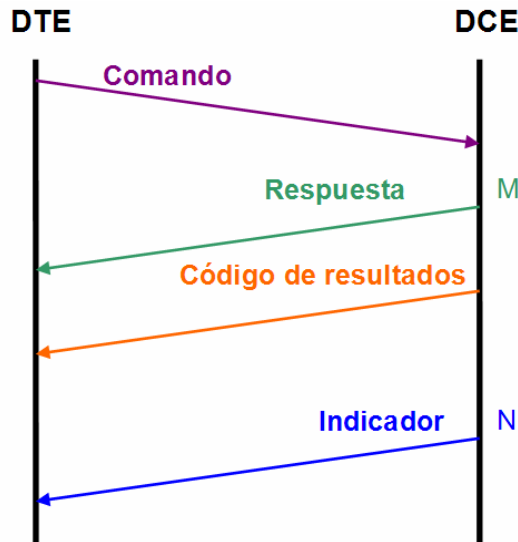


Fig. 3.6 Protocolo de comandos AT

3.2.1.3. Línea de comandos AT

Cada línea de instrucción debe iniciar con el código de atención [10] [13], "AT", y a continuación se escribe la información que el DCE debe interpretar.

Por ejemplo: Si introducimos la orden ATD12345678, el módulo estará recibiendo la orden de marcar el número de teléfono 12345678.

Tanto el código de atención como los comandos pueden ser introducidos en mayúsculas o minúsculas. Además, en cada línea de instrucción se pueden concatenar varios comandos, uno detrás de otro sin necesidad de separadores, siempre que se trate de comandos básicos; pero es necesario escribir punto y coma entre dos comandos, si éstos son extendidos. Los comandos extendidos son los que tienen el prefijo "+".

El hecho de tener varios comandos AT en la misma línea nos evita el tener que escribir el código "AT" antes de cada comando y tener que esperar una respuesta para cada comando individual. Sin embargo, la principal ventaja consiste en que nos permite ahorrar ancho de banda en el enlace entre el módulo y el DTE.

La longitud máxima de una línea de instrucciones es de 391 caracteres y viene determinada por la capacidad de memoria intermedia. Si se sobrepasa este número, no se ejecutará ninguno de los comandos de la línea y el TA devolverá un mensaje de error.

Para poder ejecutar una línea de comandos, se debe añadir al final de la misma los caracteres <CR><LF>. De no ser así, la línea de comandos permanecerá almacenada en la memoria intermedia de instrucciones hasta que reciba la correspondiente secuencia <CR><LF> o hasta que sea cancelada.

Tras la recepción de <CR><LF> el DCE ejecuta las órdenes de la línea y responde con un mensaje adecuado.

La figura 3.7 resume la estructura básica de una línea de comandos AT:

Prefijo: "AT"	Comando 1	Delimitador: ";" para extendidos	Comando 2	Delimitador: ";" para extendidos	...	Comando n	Sufijo: "<CR><LF>"
------------------	------------------	--	------------------	--	-----	------------------	-----------------------

Fig. 3.7 Estructura de una línea de comandos AT

3.2.1.4. Estructura de comandos

Tal como se observa en la figura 3.8, cada comando individual está compuesto de tres partes: formato, modo y argumento [10] [13] [14]. El **formato** indica si se trata de un comando básico, de un registro S o de un comando extendido. El **modo** se nos informa si se trata de un comando de lectura, de escritura, de prueba o de ejecución. El **argumento** puede ser un parámetro o conjunto de parámetros separados por coma “,”.

<u>Formato:</u>	<u>Modo:</u>	<u>Argumento:</u>
-Básico -Registro S -Extendido	-Lectura -Escritura -Prueba -Ejecución	-Parámetro -Conjunto de Parámetros

Fig. 3.8 Estructura de un comando AT

Formato

El formato de un **comando básico**, excepto para los comandos D y S, es: "COMANDO[<número>]". El comando puede ser tanto un caracter individual como el caracter "&" seguido de un caracter individual (que será un caracter alfabético). El número representa un valor entero decimal y puede ser una cadena de uno o más caracteres de "0" a "9".

Los **registros S** son localizaciones de memoria en el DCE donde se almacena la configuración activa. Para modificar y comprobar determinados valores de éstos registros, se emplean un grupo especial de comandos, que empiezan con la letra "S". Estos se diferencian de otros comandos en que el número que se

encuentra a continuación de la “S” indica el número de registro que está siendo referenciado. A continuación de este número puede aparecer un “=”, que se usa para darle un nuevo valor al parámetro S; o un “?”, que sirve para leer el valor actual del registro indicado.

Tal como habíamos comentado, los **comandos extendidos** son los que comienzan con el carácter “+”.

Modo

Existen cuatro tipos de comandos AT extendidos:

Los **comandos de lectura**, tienen la estructura “AT+COMANDO?” Y retornan el valor actual de los parámetros. Para establecer los valores de los parámetros relacionados con el usuario se usan los **comandos de escritura**, “AT+COMANDO=<...>”.

Por su parte, los **comandos de prueba**, “AT+COMANDO=?”, hacen que el equipo devuelva la lista de parámetros y rangos de valores establecidos con el correspondiente comando de escritura o por procesos internos. Por último, los **comandos de ejecución**, “AT+COMANDO”, leen los parámetros no variables afectados por procesos internos del equipo.

Argumento

Un **parámetro** es un valor adicional que se debe introducir junto con el comando y que puede ser tanto una constante numérica, como una secuencia de caracteres constante (debe iniciar y finalizar con comillas (“ ”)).

Como explicamos con anterioridad, el argumento de un comando AT puede ser un parámetro o un conjunto de parámetros separados por coma:

<parámetro>

<parámetro1>, <parámetro2>, ..., <parámetroN>

A lo largo de este documento los parámetros opcionales aparecerán entre corchetes “[]” y un nombre que se encuentre entre “< >” será un elemento de sintaxis; por lo que dichos caracteres no deben aparecer en la línea de comandos.

Si un parámetro es una cadena de caracteres debe ir entre comillas (“ ”). Los parámetros opcionales o subparámetros pueden ser omitidos a no ser que tengan otros parámetros a continuación. Un parámetro puede ser sustituido por una coma si se encuentra en medio de una cadena de caracteres.

3.2.1.5. Estructura de códigos de resultados

Cuando el terminal emite un comando, el módulo responde con un mensaje llamado “código de resultados” [10] [13] [14], el cual proporciona información sobre el estado de ejecución de las instrucciones (es decir, el resultado del comando AT solicitado) o sobre el estado de una conexión. Estos códigos se pueden presentar tanto en formato de código numérico como en respuestas verbales y su estructura se resume en la figura 3.9.

Prefijo: <CL><RF>	Código: Números o palabras	Sufijo: <CL><RF>
---	---	--

Fig. 3.9 Estructura del código de resultados

3.2.1.6. Estructura de respuestas e indicadores

Tanto la estructura de las respuestas como la de los indicadores [10] [13] [14] (Fig. 3.10) es muy similar a la de los comandos; pero se diferencia en que entre el formato y el argumento hay un carácter que actúa como separador, “.”.

<u>Formato:</u> -Básico -Registro S -Extendido	<u>Separador:</u> “.”	<u>Argumento:</u> -Parámetro -Conjunto de Parámetros
--	---------------------------------	--

Fig. 3.10 Estructura de respuestas e indicadores

3.2.1.7. Comandos AT para GPRS

Existe una gran variedad de comandos AT, en función de las diversas funciones que se quieran desarrollar en el módulo o DCE. Nosotros nos centraremos en aquellas que necesitamos para poder llevar a cabo la aplicación que nos interesa.

Por tanto, estudiaremos los comandos AT que el DTE puede usar para controlar al terminal móvil, definidos en el estándar **GSM 07.07**; pero específicamente aquellos que se refieren al servicio GPRS.

Dicho estándar proporciona una interfaz no multiplexada, lo que da lugar a ciertas limitaciones en la funcionalidad de la misma. Así por ejemplo, no es posible para el MT enviar información de control hacia el TE o para el TE enviar comandos mientras la interfaz está en una conexión de datos; por lo que tendríamos que emplear uno de los mecanismos explicados en el epígrafe 2.2.1 para conmutar entre modos de funcionamiento.

En los siguientes apartados explicaremos los comandos GPRS soportados por los módulos SAGEM, estos comandos pueden ser consultados en [10].

Comando +CGATT: attach o detach GPRS

Este comando permite al MT realizar los procesos de attach o detach del servicio GPRS. Si el MT ya se encuentra en el estado solicitado, este comando es ignorado y se devuelve la respuesta "OK". Si la petición no puede ser realizada, se devuelve una respuesta "ERROR". Todo contexto PDP activo se desactivará automáticamente cuando el estado cambie de attach a detach.

- Comando de prueba: **AT+CGATT=?**

Se usa para pedir información sobre el estado de los servicios GPRS soportados.

Respuesta: **+CGATT:** (lista de estados (<state>) soportados)
OK

- Comando de lectura: **AT+CGATT?**

Devuelve el estado actual del servicio GPRS.

Respuesta: **+CGATT: <state>**
OK

- Comando de escritura: **AT+CGATT= <state>**

Este comando realiza el attach o detach del MT del servicio GPRS.

Respuesta: **OK**

Parámetros: **<state>**: Indica el estado del proceso de attach GPRS (**0**:detached; **1**:attached).

Comando +CGACT: Activación o desactivación del contexto PDP

Este comando se usa para activar o desactivar el contexto PDP (Packet Data Protocol).

- Comando de prueba: **AT+CGACT=?**

Respuesta: **+CGACT:** (lista de estados (<state>) soportados)
OK

- Comando de lectura: **AT+CGACT?**

Respuesta: **+CGATT: <cid>, <state>**
OK

- Comando de escritura: **AT+CGACT= <state> [, <cid>]**

Respuesta: **OK**

Parámetros:

<state>: Indica el estado de activación del contexto PDP (**0**:desactivado; **1**:activado). Otros valores resultarán en una respuesta de "ERROR" ante el comando de ejecución.

<cid>: Se conoce como identificador del contexto PDP y es un parámetro numérico que especifica una definición particular del contexto PDP. Es un parámetro local a la interfaz entre el TE y el MT y se usa en otros comandos relacionados con el contexto PDP.

Comando +CGCLASS: Clase de estación móvil GPRS

Este comando se usa para lograr que el módulo opere de acuerdo a la clase de terminal GPRS especificada. Si la clase pedida no es soportada se devolverá una respuesta de "ERROR". El módulo debe ser reiniciado para que este comando se haga efectivo.

- Comando de prueba: **AT+CGCLASS=?**

El comando de prueba se utiliza para pedir información de las clases soportadas.

Respuesta: **+CGCLASS:** (lista de clases (<class>) soportadas)
OK

- Comando de lectura: **AT+CGCLASS?**

Devuelve la clase GPRS actual.

Respuesta: **+CGCLASS: <class>**
OK

- Comando de escritura: **AT+CGCLASS= <class>**

Por medio de este comando se puede establecer la clase en la que funcionará el módulo.

Respuesta: **OK**

Parámetros:

<class>: Este parámetro es una cadena de caracteres que indica la clase de terminal móvil GPRS (en orden descendiente de funcionalidad). (**B**: Clase B; **CG**: clase C sólo en modo GPRS; **CC**: clase C sólo en modo de conmutación de circuitos). Debemos tener en cuenta que nuestro módulo no soporta la clase A.

Comando +CGDCONT: Definir contexto PDP

- Comando de prueba: **AT+CGDCONT=?**

Este comando devuelve todos los valores soportados. Si por ejemplo, el MT soporta diferentes tipos PDP, <PDP_type>, este comando devolverá en líneas separadas los rangos de valores para cada <PDP_type>.

El número de contextos PDP que pueden estar en un estado definido al mismo tiempo lo da el rango devuelto por el comando de prueba.

Respuesta:

+CGDCONT: (rango de <cid> soportados), <PDP_type>,,,(lista de <d_comp> soportados), (lista de <h_com> soportados)[, (lista de <pd1> soportados) [...],[, (lista de <pdN> soportados)]]][...]]
[+CGDCONT:(rango de <cid> soportados), <PDP_type>,,,(lista de <d_comp> soportados), (lista de <h_com> soportados)[, (lista de <pd1> soportados) [...],[, (lista de <pdN> soportados)]]][...]]] **OK**

- Comando de lectura: **AT+CGDCONT?**

El comando de lectura devuelve el escenario actual para cada contexto definido.

Respuesta:

+CGDCONT:<cid>,<PDP_type>,<APN>,<PDP_addr>,<d_comp>,<h_comp>[,<pd1> [...],[,pdN]]]
[+CGDCONT:<cid>,<PDP_type>,<APN>,<PDP_addr>,<data_comp>,<head_comp> [,<pd1>[,...],[,pdN]]] **OK**

- Comando de escritura:

AT+CGDCONT=<cid>,<PDP_type>,<APN>,<PDP_addr>,<d_comp>,<h_comp>

Este comando especifica los valores de parámetros del contexto PDP para un contexto PDP identificado por el parámetro de identificación de contexto (local), <cid>. Una forma especial de este comando, +CGDCONT=<cid> provoca que los valores para el número de contexto <cid> se vuelvan indefinidos.

Respuesta: **OK**

Parámetros:

<PDP_type>: Packet Data Protocol type. Es una cadena de caracteres que especifica el tipo de protocolo de paquete de datos. Debemos tener en cuenta que el módulo sólo soporta el protocolo IP.

<APN>: Acces Point Name. Este parámetro es un nombre lógico (cadena de caracteres) que se usa para seleccionar al GGSN o la red de paquetes de datos externa.

<PDP_addr>: Cadena de caracteres que identifica al MT en el espacio de direcciones aplicable al PDP. Como actualmente sólo se soporta IP, este parámetro debe ser una dirección IP. Si el valor es nulo ("0.0.0" ó "0"), entonces el TE puede proveer uno durante el procedimiento de arranque del PDP o si esto falla, se puede solicitar una dirección IP dinámica. El comando de lectura continuará devolviendo el valor nulo, aun cuando se haya asignado una dirección durante el proceso de inicio del PDP. La dirección asignada se puede leer usando el comando +CGPADDR.

<d_comp>: Parámetro numérico que controla la compresión de datos PDP (**0**: off, viene por defecto y es el único valor soportado).

<h_comp>: Parámetro numérico que controla la compresión de cabecera PDP (**0**: off, viene por defecto y es el único valor soportado).

<pd1>,...<pdN>: Cadena de 0 a N caracteres cuyos significados son específicos del <PDP_type>

Comando +CGDATA: Entrar en estado de datos

- Comando de prueba: **AT+CGDATA=?**

Respuesta: **+CGDATA:** (lista de <L2P> soportados)
OK

- Comando de escritura: **AT+CGDATA= <L2P>,<cid>**

Respuesta: **CONNECT**

Parámetros:

<L2P>: Cadena de caracteres que indica el protocolo de capa 2 que va a ser usado entre el TE y el MT. Actualmente sólo está permitido el protocolo PPP.

<cid>: Explicado anteriormente.

Comando +CGEREP: Informe de evento GPRS

- Comando de prueba: **AT+CGEREP=?**

Respuesta: **+ CGEREP:** (lista de <mode> soportados), (lista de <bfr>)
OK

- Comando de lectura: **AT+ CGEREP?**

Respuesta: **+ CGEREP:** <mode>,<bfr>
OK

- Comando de escritura: **AT+ CGEREP = [<mode>,<bfr>]]**

Respuesta: **OK**

Parámetros:

<mode>: Puede tomar los valores 0, 1 ó 2 (**0**: almacenar en memoria los códigos de resultado no solicitados en el MT. Si la memoria está llena, los más antiguos pueden ser descartados. Los códigos no se reenvían al TE. **1**: descartar códigos de resultados no solicitados cuando el enlace entre MT y TE está reservado; por ejemplo en modo de conexión de datos, si no reenviarlos directamente al TE. **2**: almacenar en memoria los códigos de resultado no solicitados en el MT cuando el enlace entre MT y TE está reservado y enviarlos al TE cuando el enlace vuelva a estar disponible).

<bfr>:(**0**: la memoria de códigos de resultados no solicitados del MT, definida en este comando, se borra cuando se introducen los valores 1 ó 2 para el parámetro <mode>. **1**: los códigos de resultados no solicitados que se encuentran en la memoria del MT son enviados hacia el TE cuando se establece el parámetro <mode> con los valores 1 ó 2. Se deben enviar respuestas de OK después de enviar los códigos).

Comando +CGPADDR: Mostrar dirección PDP

Este comando devuelve una lista de direcciones PDP para los identificadores de contexto especificados.

- Comando de prueba: **AT+CGPADDR=?**

Respuesta: **+ CGPADDR:** (lista de <cid> soportados)
OK

- Comando de escritura: **AT+ CGPADDR =<cid>,<cid>,[...]]**

Respuesta:

+ CGPADDR: <cid>,<PDP_addr>
[+ CGPADDR: <cid>,<PDP_addr>[...]]

OK

Parámetros: **<PDP_addr>**: Cadena de caracteres que identifica al MT en el espacio de direcciones aplicable al PDP. La dirección puede ser estática o dinámica. La dirección estática será la establecida por el comando +CGDCONT cuando fue definido el contexto. La dirección dinámica será la asignada durante la última activación del contexto PDP que utilizó la definición de contexto referenciada por el parámetro <cid>. < PDP_addr > se omite si no hay ninguna dirección disponible.

Comando +CGQMIN: Perfil de calidad de servicio mínimo aceptable

Este comando habilita al terminal a especificar el mínimo perfil aceptable, que es comprobado por el MT comparándolo con el perfil negociado, que se encuentra en el mensaje devuelto al aceptar la activación del contexto PDP.

- Comando de prueba: **AT+CGQMIN=?**

Devuelve los rangos de valores para cada parámetro <PDP_type>

Respuesta:

+CGQMIN: <PDP_type>, (lista de **<precedence>** soportados), (lista de **<delay>** soportados),(lista de **<reliability>** soportados),(lista de **<peak>** soportados),(lista de **<mean>** soportados)
[+ CGQMIN:...]
OK

- Comando de lectura: **AT+CGQMIN?**

Devuelve los valores actuales para cada contexto PDP definido.

Respuesta:

+CGQMIN:<cid>,<precedence>,<delay>,<reliability>,<peak>,<mean>
[+CGQMIN:...]
OK

- Comando de escritura:

AT+CGQMIN=
[<cid>,<precedence>,<delay>,<reliability>,<peak>,<mean>]]]]]]]

Este parámetro especifica el perfil para el contexto identificado por el parámetro <cid>. Como este parámetro es el mismo que se usa en el comando +CGDCONT, el comando +CGQMIN es una extensión del anterior. El perfil de calidad de servicio (QoS) consta de un número de parámetros, cada uno de los cuales puede ser establecido a un valor diferente.

Respuesta: **OK**

Parámetros:

<precedence>: Parámetro numérico para la clase de precedencia (**0**: valor suscrito a la red; **1**: servicio de alta prioridad; **2**: servicio de prioridad normal; **3**: baja prioridad).

<delay>: Parámetro numérico para la clase de retardo.

<reliability>: Parámetro numérico para la clase de fiabilidad (**0**: valor suscrito a la red; **1**: tráfico no a tiempo real, aplicación sensible a errores que no puede hacer frente a la pérdida de datos; **2**: tráfico no a tiempo real, aplicación sensible a errores que puede hacer frente a pérdidas infrecuentes de datos; **3**: tráfico no a tiempo real, aplicación sensible a errores que puede hacer frente a pérdidas de datos; **4**: tráfico a tiempo real, aplicación sensible a errores que puede hacer frente a pérdidas de datos; **5**: tráfico a tiempo real, aplicación no sensible a errores que puede hacer frente a pérdidas de datos)

<peak>: Parámetro numérico para la clase de throughput máximo (**0**: valor suscrito a la red; **1**: 8 Kbps; **2**: 16 Kbps;...; **9**: 2048 Kbps).

<mean> Parámetro numérico para la clase de throughput medio (**0**: valor suscrito a la red; **1**: 0.22 bps aproximadamente; **2**: 0.44 bps aproximadamente;...; **18**: 111 Kbps aproximadamente; **31**: best effort).

Comando +CGQREQ: Petición de perfil de calidad de servicio

Este comando permite al TE especificar un perfil de QoS que es usado cuando el MT envía un mensaje de petición de activación del contexto PDP.

- Comando de prueba: **AT+CGQREQ=?**

Respuesta:

+ CGQREQ: <PDP_type>, (lista de **<precedence>** soportados), (lista de **<delay>** soportados),(lista de **<reliability>** soportados),(lista de **<peak>** soportados),(lista de **<mean>** soportados)

[+CGQREQ:...]

OK

- Comando de lectura: **AT+ CGQREQ?**

Respuesta:

+CGQREQ:<cid>,<precedence>,<delay>,<reliability>,<peak>,<mean>

[+CGQREQ:<cid>,<precedence>,<delay>,<reliability>,<peak>,<mean>

[...]

OK

Comando de escritura:

AT+ CGQREQ =
[<cid>,<precedence>,<delay>,<reliability>,<peak>,<mean>]]]]]]

Respuesta: **OK**

Parámetros: **<cid>,<precedence>,<delay>,<reliability>,<peak>,<mean>**

Comando +CGREG: Estado de registro de red GPRS

- Comando de prueba: **AT+CGREG=?**

Respuesta: **+CGREG:** (lista de <n> soportados)
OK

- Comando de lectura: **AT+CGREG?**

Respuesta: **+CGREG: <n>,<stat>[<lac>,<ci>]**
OK

- Comando de escritura: **AT+CGREG = [<n>]**

Respuesta: **OK**

Parámetros:

<n>: (**0:** deshabilitar código de resultado de registro de red no solicitado; **1:** habilitar el código de resultado de registro de red no solicitado +CGREG:<stat>; **2:** habilitar el códigos de resultado de registro de red y de información de localización no solicitado +CGREG:<stat>[<lac>,<ci>])

<stat>:(**0:** no registrado, el equipo móvil no está actualmente buscando un operador para registrarse. El servicio GPRS está deshabilitado, se le permite a la MS realizar el proceso de attach si es pedido por parte del usuario; **1:** registrado, red local; **2:** no registrado, pero el equipo móvil está actualmente intentando realizar el attach o buscando un operador para registrarse; **3:** registro denegado; **4:** desconocido; **5:** registrado)

<lac>: parámetro tipo cadena de caracteres. Código hexadecimal de dos bytes que indica el área de localización del MT.

<ci>: Cadena de caracteres que indica la identificación de la celda. Dos bytes en formato hexadecimal.

Este comando controla la presentación del código de resultados no solicitado *CGREG:<stat> cuando <n>=1 y hay un cambio en el estado de registro de red GPRS del MT; o del código *CGREG:<stat>[,<lac>,<ci>] cuando <n>=2 y se produce un cambio en la celda de red.

Comando +KMCLASS: Cambio de clase multislot GPRS

- Comando de prueba: **AT+KMCLASS=?**

Respuesta: **+KMCLASS:** (lista de <mclass> soportadas)
OK

- Comando de lectura: **AT+KMCLASS?**

Respuesta: **+KMCLASS:** <mclass>
OK

- Comando de escritura: **AT+KMCLASS = <mclass>**

Respuesta: **OK**

Parámetros: <mclass>: (1: 1+1; 2: 2+1; 4: 3+1; 8: 4+1; 10: 4+2).

Este comando necesita reiniciar el módulo para hacerse efectivo.

3.2.1.8. *Ejemplos prácticos del uso de comandos AT*

Activación de un contexto IP/PDP

Ejemplo 1:

```
AT +CGDCONT=1, "IP", "internet"; +GCDCONT=2, "IP", "abc.com"
OK
```

```
ATD*99***1#          /*con este comando comienza la conexión PPP (TCP/IP) */
CONNECT
```

Ejemplo 2:

```
AT +CGCLASS="CG"
OK
```

```
+CGREG: 1
```

```
AT +CGDCONT=1, "IP", "internet"
OK
```

```
AT +CGQREQ=1,1,4,5,2,14
OK
```

```
AT +CGQMIN=1,1,4,5,2,14
OK
```

```
AT +CGATT=1
```

OK

AT +CGACT=1,1

OK

*/*Nota sobre +CGDATA: El objetivo de este comando es el mismo que ATD*99*** así que, es mejor usar solamente ATD*99*** */*

AT +CGDATA=1

CONNECT

.....

Nota: Estos ejemplos han sido tomados del documento referenciado en [15].

Ejemplo de conexión a Internet en modo GPRS

at+cgatt? */* comprueba si el módulo está en la fase de attach GPRS*/*
+CGATT:0

OK

at+cgatt=1 */* empieza el procedimiento de attach GPRS*/*

OK

at+cgatt? */* comprueba si el módulo está en la fase de attach GPRS*/*
+CGATT:1

OK

at+cgdcont=1,"IP","APN",0,0,0 */* define el contexto PDP*/*

OK

/ at+cgqreq et at+cgqmin si la QoS necesita ser detallada*/*

atd*99***1# */* inicia la conexión GPRS (abre el contexto PDP)*/*

CONNECT

~ }#À!;!} } }2!}\$}%Ü"}&} }*} } }#}\$À#kZ~ +++ */* desconectar al DTR para colgar*/*

OK

ath */* termina la conexión GPRS (cierra el contexto PDP)*/*

OK

Recordar que +++ es el comando para escapar del modo datos y volver al modo comando.

Nota: Este ejemplo ha sido tomado del documento referenciado en [8].

Ejemplos de comandos GPRS

Ejemplo 1: +CGCLASS, clase de estación móvil GPRS

```
AT+CGCLASS=?  
+CGCLASS: (B)  
OK
```

Ejemplo 2: +CGDCONT, definir contexto PDP

```
AT+CGDCONT=?  
+CGDCONT: (1-3),("IP"),,,(0,1),(0,1)  
OK
```

```
AT+CGDCONT?  
+CGDCONT: 1,"IP",,"", "0.0.0.0",0,0  
+CGDCONT: 2,"IP",,"", "0.0.0.0",0,0  
+CGDCONT: 3,"IP",,"", "0.0.0.0",0,0  
OK
```

```
at+cgdcont= 1,"IP","internet","0.0.0.0",0,0  
OK
```

```
at+cgdcont?  
+CGDCONT: 1,"IP","internet","0.0.0.0",0,0  
+CGDCONT: 2,"IP",,"", "0.0.0.0",0,0  
+CGDCONT: 3,"IP",,"", "0.0.0.0",0,0  
OK
```

```
at+cgdcont= 1,"IP","internet","0.0.0.0",0,0  
OK
```

```
at+cgdcont=2,"IP","internet","0.0.0.0",1,1  
OK
```

Ejemplo 3: +CGQMIN, perfil de QoS mínimo aceptable

```
AT+CGQMIN=?  
+CGQMIN: (1-3),(0-3),(0-4),(0-5),(0-9),(0-18,31)  
OK
```

```
AT+CGQMIN?  
+CGQMIN: 1,2,4,3,9,10  
+CGQMIN: 2,2,4,3,9,10  
+CGQMIN: 3,2,4,3,9,10  
OK
```


Ejemplo 4: +CGQREQ, petición de perfil de QoS

```
AT+CGQREQ=?  
+CGQREQ: (1-3),(0-3),(0-4),(0-5),(0-9),(0-18,31)  
OK
```

```
AT+CGQREQ?  
+CGQREQ: 1,2,4,3,9,10  
+CGQREQ: 2,2,4,3,9,10  
+CGQREQ: 3,2,4,3,9,10  
OK
```

```
AT+CGQREQ=1,0,,0,0,0  
OK
```

```
AT+CGQREQ?  
+CGQREQ: 1,0,4,0,0,0  
+CGQREQ: 2,2,4,3,9,10  
+CGQREQ: 3,2,4,3,9,10  
OK
```

Ejemplo 5: +CGATT: Attach o Detach GPRS

```
AT+CGATT=?  
+CGATT: (0,1)  
OK
```

```
AT+CGATT?  
+CGATT: 0  
OK
```

```
AT+CGATT=0  
OK
```

Nota: Estos ejemplos han sido tomados del documento referenciado en [13].

4.3. Comunicación entre DCE y GGSN

Ya sabemos cómo se realiza la comunicación entre el DTE y el DCE; por lo que en este apartado estudiaremos los elementos que hacen posible establecer la conexión hasta el GGSN.

Como habíamos comentado en la introducción al capítulo, el DCE puede comunicarse conectarse a Internet usando una conexión TCP/IP [5] [12] [13]. Veremos que esto es posible gracias al protocolo PPP, que proporciona un mecanismo para que IP y otros protocolos de red puedan funcionar sobre un enlace serie.

Básicamente, explicaremos el funcionamiento de PPP y cómo se relacionan los diferentes protocolos hasta llegar a establecer la conexión TCP con el GGSN. La figura 3.11 muestra los protocolos que intervienen en el proceso de comunicación.

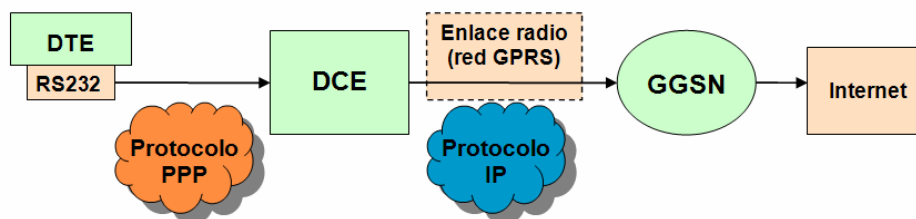


Fig. 3.11 Comunicación entre DCE y GGSN mediante protocolos PPP e IP

4.3.1. El protocolo PPP

PPP (Point to Point protocol) [4] [11] es un protocolo de Internet (IP) que permite enrutar el tráfico de un usuario a un proveedor de servicios de Internet (ISP). Proporciona un método estándar para transportar datagramas multiprotocolo sobre enlaces punto a punto.

Este protocolo consta de tres características principales:

En primer lugar, proporciona un método para el encapsulado de datagramas multiprotocolo.

En segundo lugar, para permitir la portabilidad a una gran variedad de ambientes, PPP posee un protocolo de control para activar, configurar, probar y desactivar la conexión del enlace de datos, el LCP (Link Control Protocol). El LCP se usa para acordar automáticamente las opciones del formato de encapsulado, gestionar diferentes límites en el tamaño de los paquetes, detectar errores de configuración y finalizar el enlace. También proporciona autenticación y determina cuando un enlace está funcionando correctamente o está fallando.

Y en tercer lugar, el PPP provee un mecanismo para negociar las opciones de la capa de red independientemente del protocolo de red utilizado; que consiste en tener un conjunto de protocolos NCP (Network Control Protocol), un NCP distinto para cada protocolo de capa de red soportado. Cada NCP gestiona las necesidades específicas requeridas por sus respectivos protocolos de la capa de red.

3.3.1.1. Funcionamiento de PPP

Para establecer una comunicación sobre un enlace punto a punto, cada extremo debe enviar paquetes LCP para configurar y probar el enlace de datos. Después de establecer el enlace, un extremo puede ser autenticado. A continuación, se envían paquetes NCP para elegir y configurar uno o más protocolos de red y después de configurar cada uno de los protocolos de red elegidos, se pueden enviar sobre el enlace datagramas de cada uno de estos protocolos.

El enlace continuará configurado hasta que determinados paquetes LCP o NCP cierren el enlace o hasta que ocurra algún evento externo.

Para entender mejor el funcionamiento del enlace PPP, veamos un diagrama que muestra el proceso de configuración, mantenimiento y finalización del mismo.

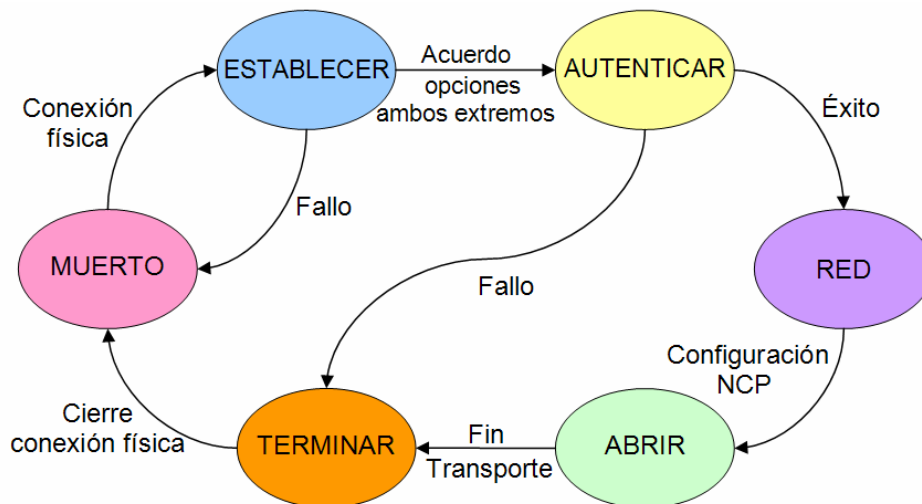


Fig. 3.12 Diagrama de estados de un enlace PPP

La figura 3.12 muestra cómo el enlace PPP pasa a través de diferentes fases. El enlace necesariamente empieza y termina en el estado “Muerto”, que significa que la capa física aún no se encuentra operativa. Si algún evento indica que se ha establecido la conexión en la capa física, el enlace PPP pasará al estado “Establecer”. Típicamente, el enlace puede volver a “Muerto” automáticamente después de la desconexión del módem.

Durante la fase “Establecer” se utiliza LCP para negociar las opciones del protocolo de enlace de datos. Este protocolo se encarga de los mecanismos de negociación y no de las opciones en sí mismas, es decir, proporciona un método para que se haga una propuesta y para que se acepte o rechace total o parcialmente. Además, el LCP permite a los extremos probar la calidad de la línea y comprobar si es aceptable para establecer una comunicación. Es importante considerar que sólo las opciones de configuración que son

independientes de un protocolo de red particular son negociadas por el LCP. Los paquetes recibidos durante esta fase que no sean LCP deben ser descartados.

El siguiente estado es “Autenticar”, en el que los extremos pueden verificar la identidad del otro si así lo desean. Si este proceso ha tenido éxito se pasa a la fase de “Red” y si falla, se procede instantáneamente a la fase “Terminar”.

Al entrar en la fase de “Red” cada protocolo de red debe ser configurado por el correspondiente NCP. Si la configuración ha tenido éxito se pasa a “Abrir” y puede comenzar el transporte de paquetes de la capa de red. Durante esta última fase, el tráfico del enlace es una combinación entre paquetes LCP, NCP o paquetes de protocolos de red.

Al finalizar el transporte, el enlace pasa al estado “Terminar” y al liberarse la conexión física pasa nuevamente a “Muerto”. PPP puede terminar el enlace en cualquier momento, lo que puede suceder debido a un fallo en la autenticación, baja calidad del enlace, pérdida de la conexión física, etc. Cuando el enlace se está cerrando, PPP informa a los protocolos de red para que éstos puedan realizar las acciones pertinentes.

3.3.1.2. Formato de tramas PPP

Como habíamos comentado, PPP es un mecanismo de entramado multiprotocolo. A continuación (Fig. 3.13) [4] veremos cómo es el formato de la trama PPP.

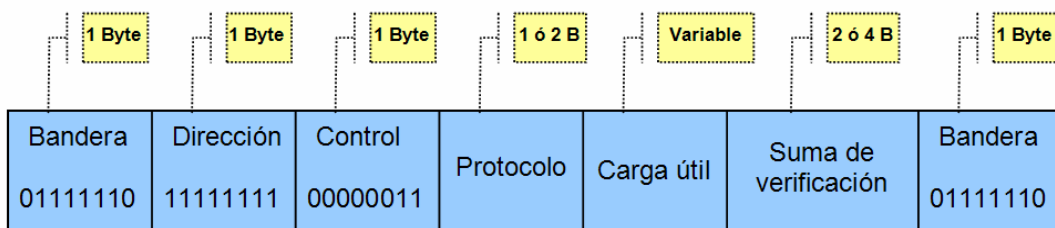


Fig. 3.13 Formato de trama PPP

Las tramas PPP comienzan con el campo “Bandera”, cuyo valor estándar es 01111110 y se rellena si aparece dentro del campo de “Carga útil”.

A continuación aparece el campo “Dirección” que siempre tiene asignado el valor 11111111 para indicar que todos los pares deben aceptar la trama.

En tercer lugar, tenemos el campo “Control”, de 1 byte de longitud, cuyo valor indica si la trama es numerada o no, es decir, si en la transmisión se usan números de secuencia y confirmaciones de recepción. El valor predeterminado

es 00000011, que indica una trama no numerada; lo que significa que PPP no proporciona de manera predeterminada una transmisión confiable.

El cuarto campo de la trama tiene un tamaño predefinido de 2 bytes; pero puede negociarse a 1 byte usando LCP. Es el campo "Protocolo", que indica la clase de paquete contenido en la "Carga útil". Los protocolos que comienzan con el bit "0" son los de la capa de red (IP, por ejemplo) y los que comienzan por "1" se utilizan para negociar otros protocolos (LCP y el conjunto de protocolos NCP).

El campo "Protocolo" va seguido por el de "Carga útil", de longitud variable hasta un máximo negociado. Si durante el proceso de establecimiento del enlace PPP no se negocia dicha longitud, entonces tendrá un valor predeterminado de 1500 bytes. Después de la carga se puede añadir unos bits de relleno si fuera necesario.

El penúltimo campo es el de "Suma de verificación", que normalmente es de 2 bytes; pero puede negociarse a 4. Para finalizar la trama se añade el byte de "Bandera".

El hecho de que PPP sea un mecanismo de entramado multiprotocolo significa que dentro de la trama PPP se pueden encapsular paquetes de diferentes protocolos y debido a que los campos explicados anteriormente se usan para encapsular la información fundamental del protocolo en cuestión; el formato de la trama PPP se podría resumir a los campos mostrados en la figura 3.14 [11].

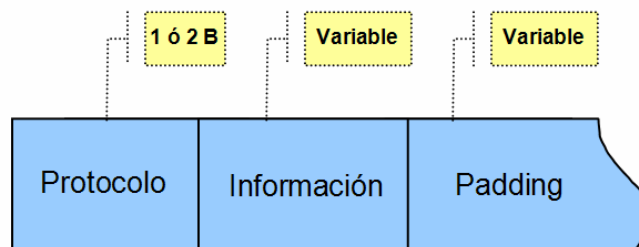


Fig. 3.14 Trama PPP resumida

El campo "Protocolo" son 1 ó 2 octetos y su valor identifica al datagrama encapsulado en el campo de información de la trama. El campo "Información" es de longitud variable y contiene al propio datagrama del protocolo especificado en el campo "Protocolo".

Como comentamos anteriormente, la máxima longitud para este campo, incluyendo el "Padding" pero no el campo "Protocolo", es de 1500 bytes por defecto y viene determinada por la unidad máxima de recepción MRU. Por último el "Padding" también es de longitud variable y contiene bits de relleno para que el campo de "Información" alcance la longitud determinada por el MRU.

3.3.1.3. Tipos de paquetes LCP

Los paquetes LCP [4] [11] se pueden clasificar en tres categorías: Los paquetes de **configuración**, que se usan para establecer y configurar el enlace. Los paquetes de **terminación**, que son los encargados de finalizar la línea y los de **mantenimiento**, que se usan para administrar y depurar el enlace.

Los LCP son encapsulados en el campo “Carga útil” de la trama PPP cuando el campo “Protocolo” de la misma, indica que se trata del protocolo LCP.

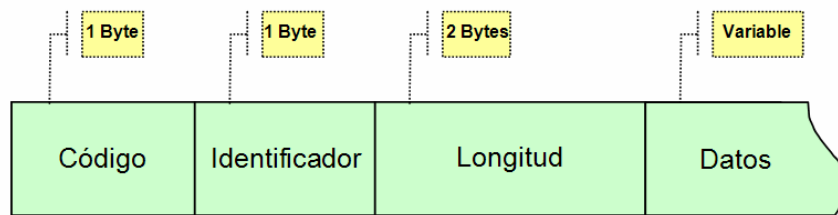


Fig. 3.15 Formato de paquetes LCP

La figura 3.15 muestra el formato de los paquetes LCP. El campo “Código” es un octeto que identifica el tipo de paquete LCP. El campo “identificador”, también consta de 8 bits que ayudan a emparejar peticiones y respuestas. El de “Longitud” son 2 bytes que indican la longitud del paquete LCP, incluyendo “Código”, “Identificador” y el campo “Datos”. Los bits que se encuentren fuera del rango del campo indicado en “Longitud” se consideran relleno y se ignoran en recepción. Por último, el campo “Datos” es de longitud variable, determinada por el valor de “Longitud” y el formato del mismo depende del valor del campo “Código”.

La tabla 3.1 lista los 11 tipos de paquetes LCP: “Configurar solicitud”, “configurar confirmación de recepción” (ACK), “configurar confirmación de recepción negativa” (NACK) y “configurar rechazo” son de la clase de paquetes de configuración; “terminar solicitud” y “terminar confirmación de recepción” son de tipo de terminación de enlace; por último, “código rechazo”, “protocolo rechazo”, “eco solicitud”, “eco respuesta” y “descartar solicitud” son de la categoría de paquetes de administración y depuración.

Los de tipo configuración permiten que el que inicia la negociación (I) proponga valores para una determinada opción y que el que contesta (C) las acepte o rechace. En el caso de que el contestador no acepte la propuesta, puede anunciar al iniciado que no está dispuesto a negociar o también puede realizar una propuesta alternativa.

Si ocurre un error de transmisión no detectado o el iniciador y el contestador utilizan versiones del LCP diferentes; el contestador puede utilizar los mensajes de “código rechazo” y “protocolo rechazo” para indicar que ha recibido algo que no entiende.

Los códigos “eco” sirven para probar la calidad de la línea y el de “descartar solicitud” se utiliza para depurarla.

Tabla 3.1 Tipos de paquetes LCP

Valor campo “Código”	Nombre	Dirección	Descripción
1	Configurar solicitud	I → C	Lista de opciones y valores propuestos
2	Configurar ACK	I ← C	Se aceptan todas las opciones
3	Configurar NACK	I ← C	No se aceptan algunas opciones
4	Configurar rechazo	I ← C	Algunas opciones no son negociables
5	Terminar solicitud	I → C	Solicitud desactivación enlace
6	Terminar ACK	I ← C	Enlace desactivado
7	Código rechazo	I ← C	Se recibió una solicitud desconocida
8	Protocolo rechazo	I ← C	Se solicitó un protocolo desconocido
9	Eco solicitud	I → C	Solicitud de enviar de regreso esta trama
10	Eco respuesta	I ← C	Aquí está la trama de regreso
11	Descartar solicitud	I → C	Descartar esta trama (para pruebas)

4.3.2. Protocolos TCP/IP

Normalmente los sistemas de transmisión de datos no sólo utilizan un protocolo para controlar todas las funciones necesarias para la transmisión; sino que utilizan un conjunto de protocolos que cooperan entre ellos. Este también es nuestro caso; ya que como habíamos comentado, el DCE se conecta al GGSN utilizando el protocolo de red IP; pero en este proceso también intervienen los protocolos de capa física, RS232 y de la capa de enlace, PPP.

En este apartado situaremos los protocolos de comunicación estudiados, dentro de la pila de protocolos TCP/IP y veremos cómo se relacionan entre ellos.

3.3.2.1. Organización de protocolos por capas

El conjunto de protocolos que intervienen en el proceso de comunicación, se organizan conceptualmente en una serie de capas o niveles que hacen que el software del protocolo de Internet sea fácil de entender.

Desde el punto de vista teórico, si queremos enviar un mensaje de una máquina a otra, el mensaje irá pasando por las diferentes capas del software de protocolo de dicha máquina; tal como refleja la figura 3.16. Cada nivel realizará sus funciones sobre el mensaje y lo transmitirá a la capa inferior. El mensaje pasará a través de la red hasta encontrarse en la capa inferior de la máquina receptora, la cual también irá pasando el mensaje de nivel en nivel; pero ahora en sentido ascendente.

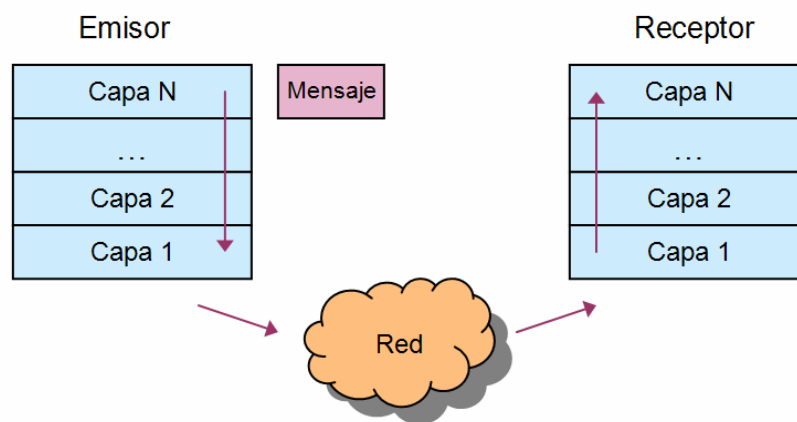


Fig. 3.16 Transferencia de un mensaje a través de las capas del software

3.3.2.2. Pila de protocolos TCP/IP

Hasta el momento hemos hablado de protocolos en general; pero para los protocolos TCP/IP se utiliza una estructura de capas concreta.

El modelo de estratificación por capas de Internet, mostrado en la figura 3.17, se conoce como **modelo de referencia TCP/IP** [5] y tiene cuatro capas conceptuales de software, situadas sobre una capa de hardware.

Los cuatro niveles de software son “Aplicación”, “Transporte”, “Red” (también conocido como Internet) y “Enlace” (o interfaz de red)

Los protocolos que en nuestro caso están asociados a estas capas son los siguientes: TCP para la capa de transporte, IP para la de red y por último PPP para la de enlace de datos.

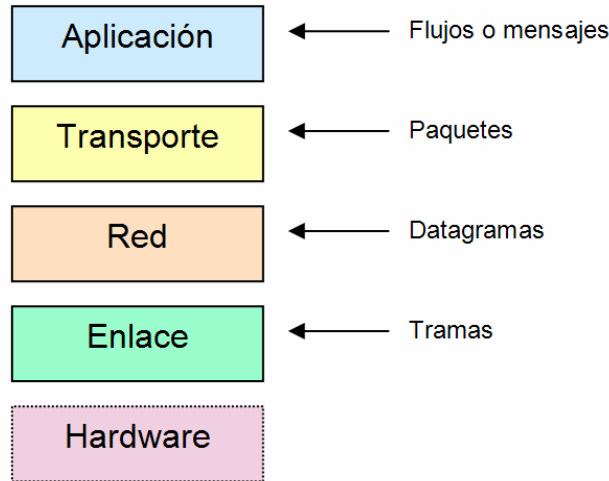


Fig. 3.17 Modelo de referencia TCP/IP para software de protocolo

3.3.2.3. Encapsulado de mensajes

Cada una de las capas de protocolos irá realizando diferentes modificaciones sobre el mensaje que se quiere transmitir.

Como habíamos comentado en el capítulo “GPRS”, los datos se fragmentan en porciones de longitud fija y se les añade una cabecera, dando lugar a los paquetes. Aplicando esto al modelo de capas, los mensajes son los datos del nivel de aplicación; éstos se fragmentan y se les añade una cabecera al pasar a la capa de transporte; los paquetes resultantes, al atravesar el nivel de red se encapsulan en el campo de datos de los datagramas y por último, los datagramas pasan a formar parte del campo de datos de las tramas del nivel de enlace.

Llevando estos conceptos al caso de los protocolos que estamos utilizando y considerando el formato de trama PPP; podemos decir que los paquetes TCP son encapsulados dentro de datagramas IP, y éstos a su vez, en tramas PPP. De forma que la estructura de la trama PPP vendría a ser como la mostrada en la figura 3.18.

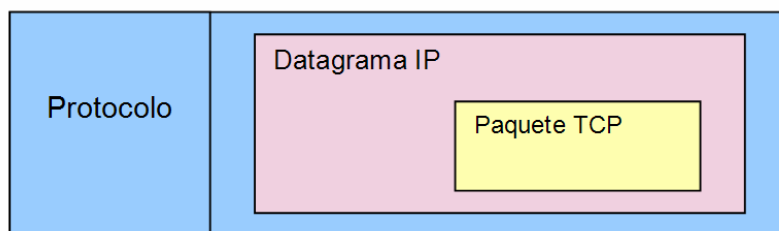


Fig. 3.18 Trama PPP encapsulando a un datagrama IP

3.3.2.4. Resumen de la comunicación completa

Para acceder a Internet, en primer lugar el TE y el MT deben conectarse en la capa física mediante un enlace RS232 y se comunican mediante comandos AT. El siguiente paso es establecer una conexión mediante el protocolo PPP entre los dispositivos, la cual se inicia mediante el comando de llamada (ATD). Este protocolo crea una interfaz de comunicaciones hacia las capas inferiores y habilita al protocolo IP para comunicarse con la red.

El módulo GPRS se comunica a través de dos interfaces físicas; por un lado, un enlace serie RS232 y por el otro, la red inalámbrica GPRS. Estas dos interfaces soportarán una conexión PPP. Pero para conectarse al GGSN, el módulo necesitará establecer una conexión TCP/IP y será precisamente el enlace PPP el que le proporcionará la conexión básica para establecer TCP/IP. Es decir, para poder acceder a la red son necesarias diferentes capas de conexión, cada una de las cuales, proporcionará soporte para las capas superiores.

La figura 3.19 muestra las diferentes capas de conexión que se establecen sobre el enlace físico RS232. Por un lado, estaría toda la pila de protocolos utilizada para poder transferir paquetes de datos PPP entre DTE y DCE y por otro, la conexión de comandos AT para controlar al DCE.

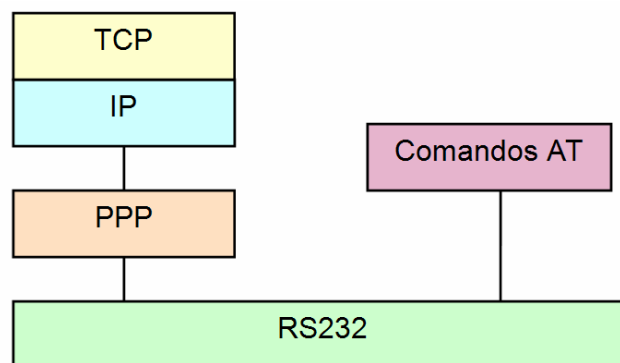


Fig. 3.19 Estructura de capas de protocolos utilizados sobre el RS232

En el caso de la conexión entre DCE y GGSN tendríamos la misma pila de protocolos; pero establecidos sobre un enlace físico GPRS (Fig. 2.20).

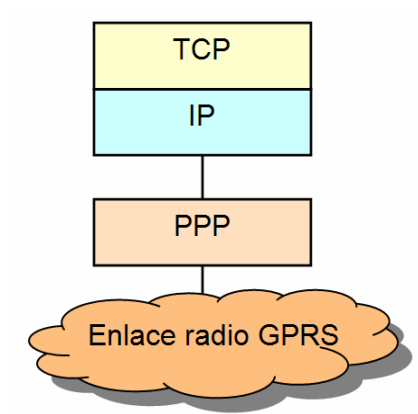


Fig. 3.20. Pila de protocolos de comunicación entre DCE y GGSN

CAPÍTULO 4. MO200

4.1. Introducción al MO200

El MO200 (Fig.4.1) es una versión mejorada del modelo anterior, el MO190 de SAGEM. El MO200 incorpora el XS200, último módem de este fabricante mucho más funcional, y está especialmente destinado a las comunicaciones M2M. El módulo proporciona conexión GSM/GPRS de clase 10 y ofrece la posibilidad de trabajar como módem en un sistema de comunicaciones inalámbrico para así poder acceder a Internet. Debido a su reducido tamaño (54.4 x 34 mm) y a su gran flexibilidad de operación, se puede integrar en casi cualquier producto [19]. Entre sus numerosas aplicaciones encontramos los sistemas de telemetría, alarmas, terminales de cobro con tarjeta de crédito, etc.

El dispositivo incluye una pila TCP/IP embebida y los drivers para utilizar el servicio de localización GPS entre otros. Dispone de comandos AT especialmente mejorados para que un microprocesador se pueda comunicar con el MO200. La comunicación entre los dos dispositivos se realiza a través del puerto serie. También se permiten comunicaciones IrDA. Las bandas de frecuencia en las que trabaja son 900/1800/1900 MHz (Europa y Asia) o 850/900/1900 MHz (América del Norte), garantizando así la operabilidad en todas las redes GSM e integrando los protocolos antes mencionados.

Físicamente el MO200 está compuesto, además del módulo GPRS, por los conectores que permiten conectarlo a otra placa, poner una antena para poder transmitir y tener una batería que alimente el sistema.

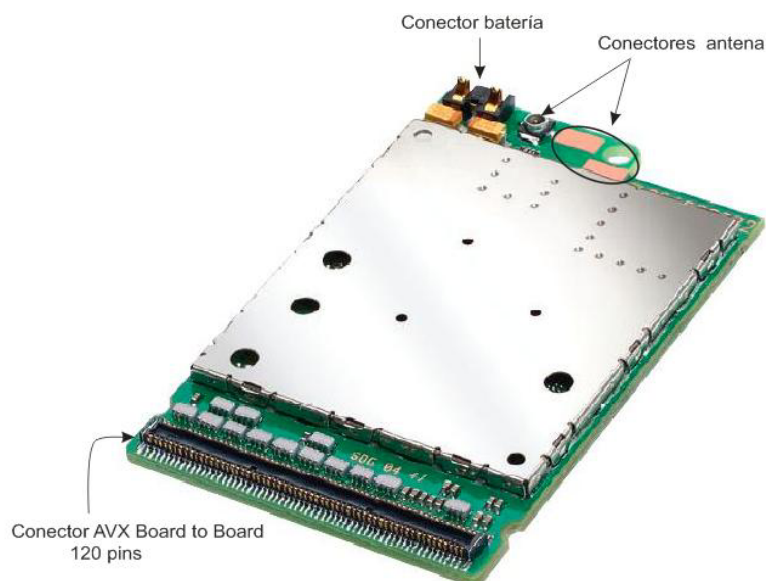


Fig. 4.1 MO200

Entre las funcionalidades del módulo hay algunas como las de audio o GPS que no son útiles para el propósito de este proyecto. En cambio dispone de otras como la conexión de una tarjeta SIM o el enlace serie que son imprescindibles. A continuación tenemos un esquema de todas ellas, tanto las que sirven como las que no:

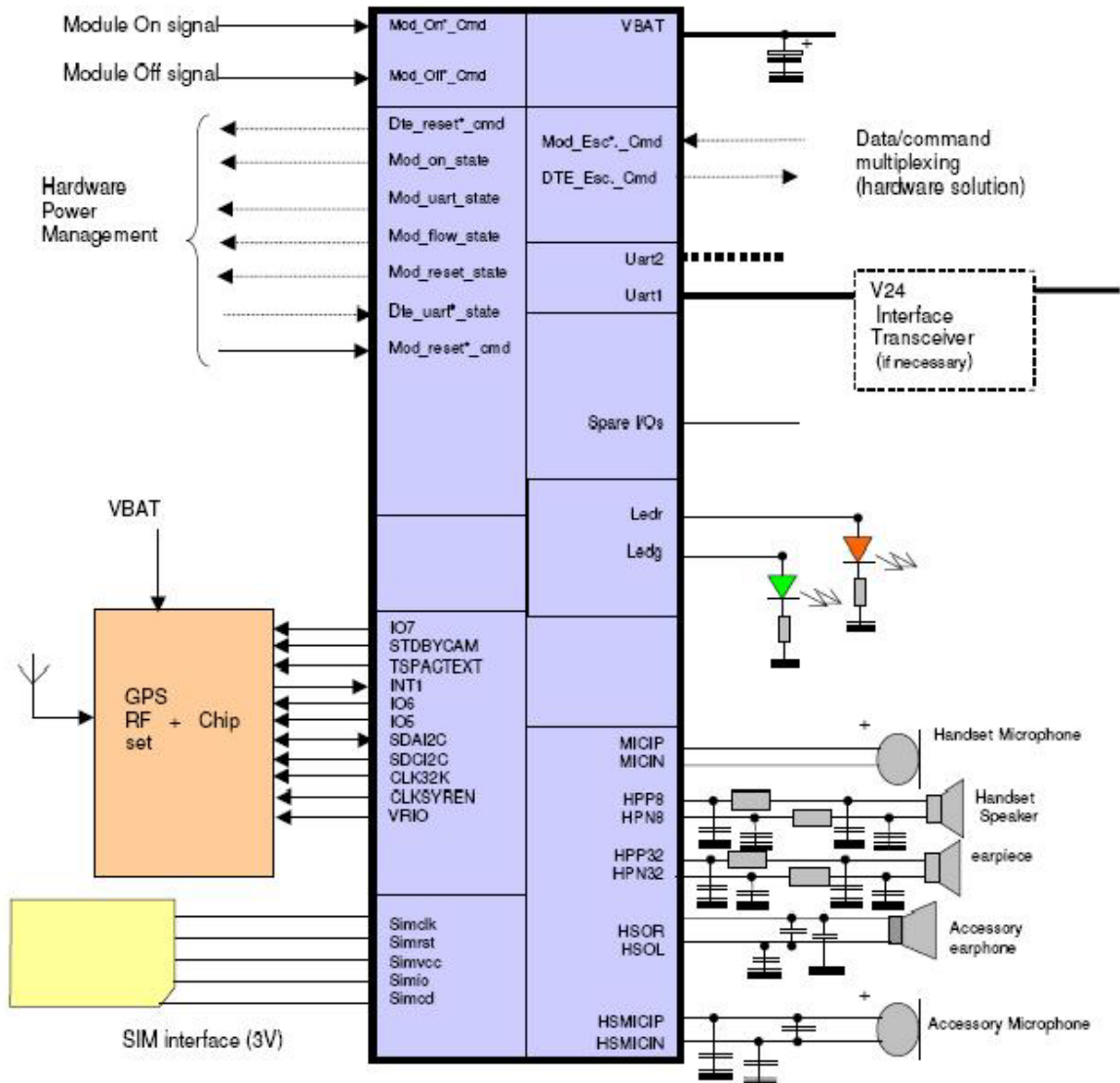


Fig. 4.2 Diagrama de bloques MO200

Observamos en la figura 4.2 que el MO200 dispone de una serie de señales que sirven para controlar la alimentación, la transmisión de datos, etc. Todas ellas se conectan al microprocesador como se indica en la figura siguiente:

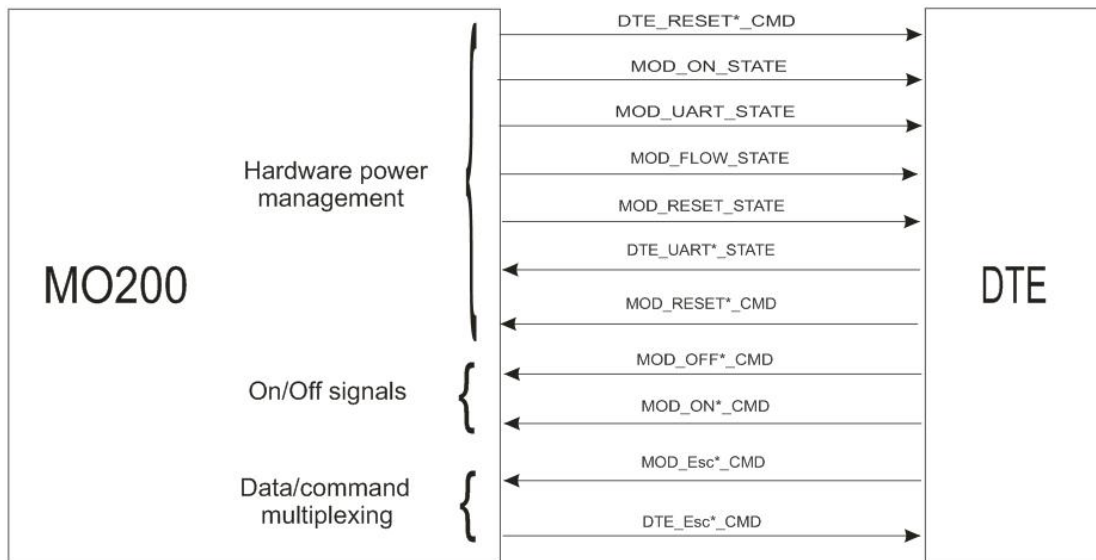


Fig. 4.3 Señales de control del MO200, entrada y salida

4.2. Descripción de las funciones del MO200

El MO200 además del envío de datos permite implementar otras funciones (Fig. 4.2) como por ejemplo un sistema de aviso mediante LEDs, la conexión de un sistema GPS o un sistema de audio. Todas ellas se explican en este apartado.

4.2.1. LEDs

Dos pins del conector de 120 patas (fig. 5.14) están dedicados a LEDs de alarma, LEDG (pin 90) y LEDR (pin 91), que servirán para avisar del estado de la batería, si hay mensajes recibidos o si se está utilizando el manos-libres. Existe un código de colores para cada caso. Ambos LEDs necesitan una resistencia serie conectada cuyo valor dependerá de las características de los LEDs.

4.2.1.1. Batería

Las indicaciones de los LEDs consisten en flashes de 30ms de duración con periodos de 2s:

- Verde cuando está conectado a la red y la batería está bien.
- Naranja (verde + rojo) cuando está conectado a la red y la batería está baja.
- Rojo cuando no hay red y la batería está baja.

4.2.1.2. Mensajes

La indicación de SMS o mensaje de voz recibido consiste en tres flashes rojos sucesivos de 20ms.

4.2.1.3. Llamada manos libres

Cuando se lleva a cabo una llamada manos libres, los LEDs cambian de verde a rojo sucesivamente con periodos de 1s. La indicación de este tipo de llamada tiene prioridad sobre las demás indicaciones.

Como se va a alimentar directamente todo el sistema desde el exterior, no hace falta saber el estado de la batería. Tampoco se recibirán mensajes ni se realizarán llamadas así que se incluyen los LEDs en el diseño.

4.2.2. Audio

El módulo tiene capacidad para gestionar un set de audio compuesto por un micrófono (MICIP y MICIN) y unos auriculares pequeños (32 Ohms: HPP32 y HPN32; de 6 a 8 Ohms: HPP8 y HPN8). También se puede conectar un micrófono (HSMICIP y HSMICIN) y auriculares estéreo (HSOL y HSOR) en los pins dedicados a los accesorios o bien un kit manos libres.

En la tabla siguiente se muestra un listado de los pins destinados a cada opción mencionada en el párrafo anterior:

Tabla 4.1. Pins del sistema de audio

Nombre	Función	# Pin
MICIP	micrófono pequeño en modo diferencial	63
MICIN		62
HPP32	Auriculares pequeños de 32 Ohms	46
HPN32		47
HPP8	Auriculares pequeños de 6 a 8 Ohms	56
HPN8		57
HSMICIP	Micrófono estéreo accesorios	52
HSMICIN		53
HSOL	Auriculares estéreo accesorios	50
HSOR		49

Para conectar tanto los auriculares como el micrófono se deben incluir resistencias y condensadores cuyos valores se especifican en el datasheet [19]. Estos elementos deben estar lo más cerca posible del MO200. Existe la posibilidad de conectar un sistema de audio no diferencial pero no se recomienda esta solución. Si se desean conectar simultáneamente unos

auriculares y un carkit, se recomienda añadir un switch y configurar los diferentes caminos mediante los comandos AT.

No se va a utilizar este servicio, de momento no sería muy útil en un avión no tripulado.

4.2.3. GPS

Podemos conectar un sistema GPS al MO200 a través de determinados pins del conector. El sistema consta de un receptor de RF y de un procesador.

El receptor se encarga de amplificar, bajar en frecuencia y digitalizar la señal de RF que llega al módulo. Una vez hecho esto la señal se envía al procesador donde se realizan todas las operaciones necesarias para determinar la posición. Es el procesador el que se comunica con el MO200.

A continuación se listan todas las señales del MO200 que intervienen en este sistema:

Tabla 4.2. Pins del sistema GPS

Nombre	# Pin
IO7	107
STDBYCAM	66
TSPACTEXT	67
IO6	65
IO5	80
SDAI2C	93
SCLI2C	14
CLK32K	9
VRIO	2

4.2.4. Transmisión de datos

Los servicios de transmisión de datos ofrecidos por el módulo son:

- GPRS
- CSD
- FAX

La transmisión de datos entre el MO200 y el microprocesador se realiza a través de un enlace V24 completo. Una segunda UART sirve para poder controlar el Debug desde el exterior. Esta UART2 no dispone de señales de control.

4.2.5. Alimentación

La placa tiene un conector para la batería (Vbat, Fig. 4.4) y el MO200 dispone de un sistema de gestión de la misma que se encarga de controlar su carga y descarga. Si el MO200 es el dispositivo SLAVE de un sistema mayor, SAGEM recomienda que esta gestión se realice desde el MASTER. En este proyecto no se dispone de batería sino que el sistema se alimenta desde una fuente exterior y por tanto no será necesaria la gestión por parte de ninguno de los dos dispositivos.

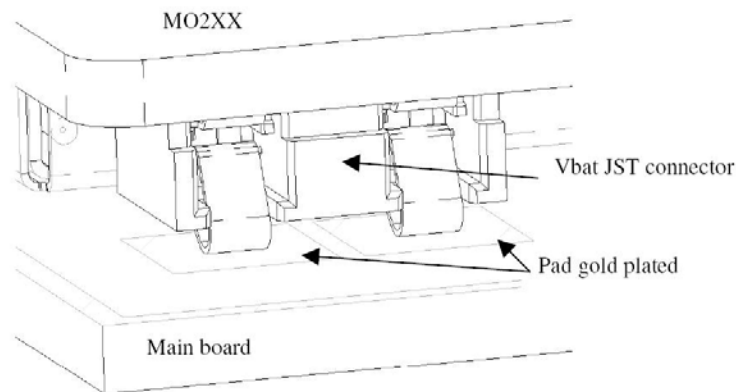


Fig. 4.4 Conector Vbat

En la placa a diseñar se debe incluir un layout tal como muestra la figura 4.5 para proporcionar la tensión y corriente necesarias al MO200:

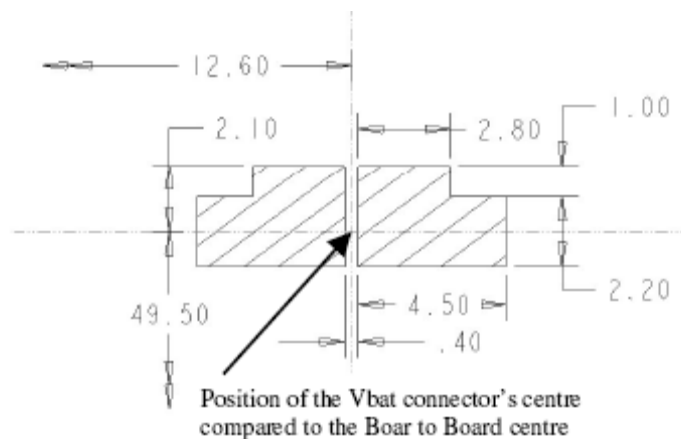


Fig. 4.5 Layout Vbat (medidas en mm)

El nivel de señal debe estar en el rango de 3,45V a 4,5V, con un valor nominal de 3,8V. Nunca debe ser menor que 3,2V porque entonces se corre el riesgo de que se produzca un RESET del sistema. La corriente máxima que va a tener que soportar es de 2 A en picos muy concretos y que se producirán en raras

ocasiones. Este nivel de corriente máximo será determinante a la hora de elegir los componentes de la nueva placa.

El pin VRIO (pin 2) proporciona una tensión de salida de 2,8V con la que se podrían alimentar otros dispositivos (10mA de corriente en modo activo). Los dispositivos elegidos para la placa (capítulo 5) necesitan 3,3V así que no serán alimentados desde este pin.

Vbackup (pin 12) sirve para conectar una batería de reserva por si falla la principal. Si no se utiliza debe conectarse directamente a Vbat (Fig. 5.28).

4.2.6. Control de la energía

El MO200 dispone de un sistema eficiente de control de la energía basado en cuatro estados de funcionamiento. Estos estados permiten que partes del módulo se encuentren desconectadas cuando no se están utilizando, generando así un ahorro de energía considerable.

4.2.6.1. Estados

Visto desde fuera, el módulo tiene 4 estados de funcionamiento:

→ OFF: en este estado no hay ninguna función activa, el enlace serie está desconectado.

→ ON Sleep: en este estado el módulo está dormido y el enlace serie está desconectado. Aunque no esté en funcionamiento, está conectado a la red GSM. Cualquier interrupción lo despierta y activa el enlace serie. La recepción de un dato sirve como interrupción pero este dato se pierde.

→ ON Active: en este estado tanto el módulo como el enlace serie están activos y preparados para recibir datos.

→ RESET: este estado indica que el módulo se está reiniciando y por tanto el enlace serie está desactivado.

El modo en que trabaja el MO200 viene determinado por los niveles lógicos de las señales MOD_ON_STATE, MOD_UART_STATE y MOD_RESET_STATE :

Tabla 4.3. Niveles de señal que determinan el estado del MO200

Estado	MOD_ON_STATE	MOD_UART_STATE	MOD_RESET_STATE
OFF	inactivo	inactivo	inactivo
ON Sleep	activo	inactivo	inactivo
ON Active	activo	activo	inactivo
RESET	activo	cualquiera	activo

Así mismo, la señal DTE_UART_STATE nos indica el estado en que se encuentra el microprocesador:

Tabla 4.4. Niveles de señal que determinan el estado del microprocesador

estado	DTE_UART_STATE
OFF	Inactivo
ON Sleep	Inactivo
On Active	Activo

A continuación se incluye una tabla que indica las transiciones de estados permitidas en el MO200 y quién puede iniciar el cambio:

Tabla 4.5. Transición de estados

	Final	OFF	ON Sleep	ON Active	RESET
Inicial					
OFF		-	Imposible	Microprocesador	Imposible
ON Sleep		Microprocesador	-	Micro./MO200	Micro./MO200
ON Active		Microprocesador	Micro./MO200	-	Micro./MO200
RESET		X	Imposible	X	-

Este control de la energía se puede realizar o bien por software o bien por hardware. El control por software se puede conseguir utilizando el protocolo 07.10 de GSM [17], mientras que para el control hardware se necesitan las señales de la tabla 4.3.

4.2.6.2. Señales implicadas en el control de la energía

Usando el control por hardware se informa a ambos dispositivos del estado del otro mediante las señales de control del MO200 (Fig. 4.3). Estas señales sirven también para que un dispositivo pida al otro que cambie de estado. Esta manera de control ayuda a conseguir un menor consumo de energía por parte del MO200 desactivando sus funciones siempre que se pueda (por ejemplo cuando no hay comunicación alguna).

Hay 4 señales de entrada y 5 de salida para controlar los diferentes estados:

- Entrada: MOD_ON*_CMD, MOD_RESET*_CMD, DTE_UART*_STATE, MOD_OFF*_CMD.
- Salida: MOD_ON_STATE, DTE_RESET*_CMD, MOD_UART_STATE, MOD_FLOW_STATE y MOD_RESET_STATE.

Y cada señal tiene su función aunque algunas de ellas ya se han explicado en el subapartado anterior:

- MOD_ON_STATE (pin 27) permanece activa mientras el MO200 está ON.
- MOD_UART_STATE (pin 25) está desactivada cuando el MO200 está en estado Sleep.
- Cuando MOD_ON_CMD (pin 113) pasa a nivel alto, significa que el microprocesador quiere que el módulo pase a estado ON.
- Cuando MOD_OFF_CMD (pin 76) pasa a nivel alto, significa que el microprocesador quiere que el módulo pase a estado OFF.
- Cuando DTE_UART_STATE (pin 92) pasa a nivel bajo, significa que el microprocesador no tiene nada que enviar de momento.
- Cuando MOD_FLOW_STATE (pin 26) pasa a nivel bajo significa que el módulo no tiene nada que enviar de momento.
- MOD_RESET_STATE (pin 24) es una señal de salida e indica al dispositivo que la recibe que se está produciendo un reset del software. Durante este proceso permanece activa pero no es fiable durante el reset del hardware.
- MOD_RESET_CMD (pin 81) señal de entrada, resetea el módulo a petición del microprocesador.
- DTE_RESET_CMD (pin 106), aunque sea una señal de salida del módulo, en realidad no está controlada por éste. Solamente está ahí por si el microprocesador necesita recibir una señal de reset.

4.2.7. SIM

En el conector de 120 patas se incluye una interfaz para el conector de la SIM compatible con la normativa ISO7816-3 IC para el standard 11.11 de GSM [18]. Esta interfaz incluye la señal SIMCD (SIM Card Detection, pin 114) que como

su nombre indica sirve para que el módulo sepa si hay o no una tarjeta SIM conectada.

El conector de SIM elegido tiene 6 pins por lo que no dispone del hardware necesario para implementar esta función. Se utiliza entonces la configuración sin SIM Card Detection, y por tanto el pin de la señal SIMCD debe conectarse a Vcc. En este modo de operación se supone que siempre hay una SIM conectada.

En cualquier caso, las señales SIMCLK (pin 4), SIMVCC (pin 6), SIMIO (pin 7) y SIMRST (pin 5) necesitan condensadores de desacoplo de 15 pF (fig.4.6), que deben estar lo más próximos posible al conector.

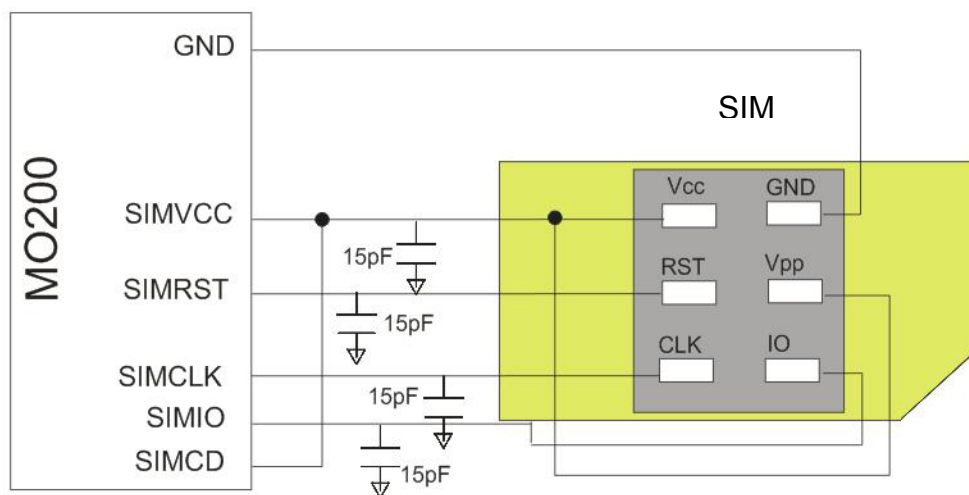


Fig. 4.6 Esquema de conexión de la SIM

SIMVCC proporciona la alimentación a la SIM; SIMIO, SIMRST y SIMCLK son las señales utilizadas para la comunicación.

Cuando se inserta la SIM en el conector, ésta queda girada respecto de la figura 4.6 y por tanto las conexiones quedan al revés (Fig. 4.7). Esto debe tenerse muy en cuenta a la hora de diseñar las conexiones en la placa.

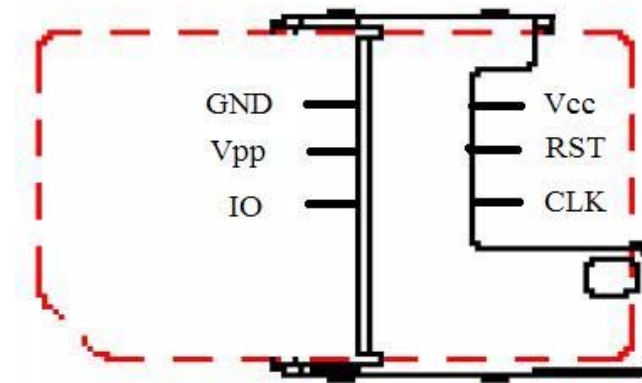


Fig. 4.7 Conexiones SIM en el conector

4.2.8. Enlace V24

En los pins externos se proporciona una interfaz de la conexión V24 que soporta velocidades de hasta 115,2 kbps. El fabricante recomienda que se utilice el enlace completo:

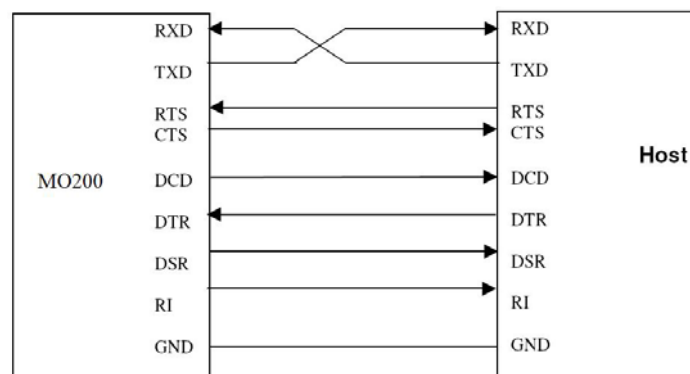


Fig. 4.8 Enlace V24 completo

Para el manejo de las comunicaciones serie, el dispositivo incluye comandos AT y paquetes de datos PPP, ya explicados en el capítulo 3. El enlace puede funcionar en dos modos diferentes:

Command Mode: el enlace está reservado exclusivamente para los comandos AT.

Data Mode: el enlace está reservado para el flujo de datos.

Como el enlace no puede estar ocupado por los dos modos al mismo tiempo, se necesita un procedimiento que cambie de uno a otro dependiendo de lo que

se tenga que transmitir (Fig. 4.10). El MO200 ofrece dos posibilidades, una es utilizar el estándar 7.10 de GSM [17] y la otra es mediante el 'Suspended Data Mode' creado por Sagem y controlado por las señales MOD_ESCAPE*_CMD (pin 18) y DTE_ESCAPE_CMD (pin 23) [18].

4.2.9. Multiplexación DATA/COMMAND

Como ya se ha explicado, hay dos tipos de datos que se pueden enviar por el enlace serie: los comandos AT y los paquetes de datos PPP. Debido a limitaciones en el ancho de banda, el enlace serie no puede ser utilizado por ambos flujos de datos a la vez y por eso existen los dos modos exclusivos de funcionamiento del enlace (DATA mode y COMMAND mode).

Puede ocurrir que durante una conexión de datos, el microprocesador tenga que comunicarle algo urgente al MO200 mediante comandos AT y aquí entra la necesidad de tener un sistema que pueda cambiar de un modo a otro automáticamente.

De entre las opciones nombradas en el apartado anterior se ha elegido utilizar el método creado por SAGEM. Éste utiliza el enlace V24 y las señales MOD_ESCAPE*_CMD (pin 18) y DTE_ESCAPE_CMD (pin 23):

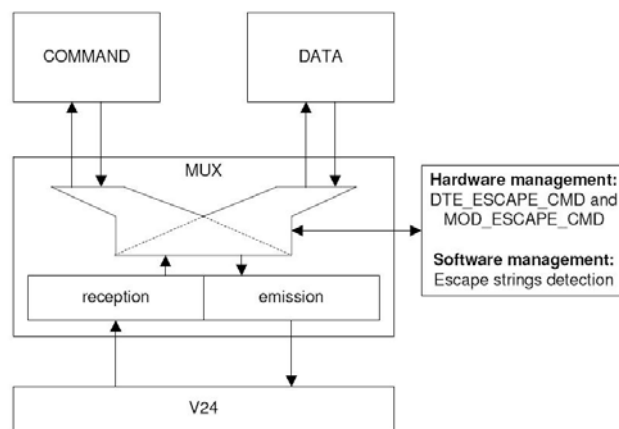


Fig. 4.9 Hardware protocol. Switch entre DATA y COMMAND

Se usa temporalmente para enviar los comandos AT. Durante el switch todos los datos deben ser guardados. La siguiente figura muestra el diagrama de estados del protocolo:

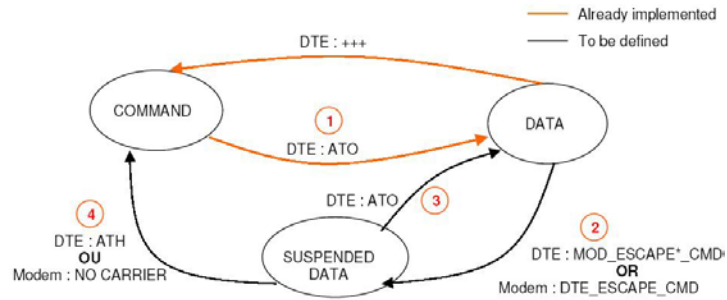


Fig. 4.10 Diagrama de estados

4.2.10. Clocks

El conector de 120 pins proporciona dos salidas de reloj, una de 13 MHz (pin 16) y otra de 32 KHz (pin 9). Estas señales son muy sensibles y deben estar perfectamente aisladas entre ellas. El fabricante recomienda aislar estas señales entre dos planos de masa y separarlas mediante vías que comuniquen los dos planos.

4.3. PINOUT

En este apartado se listan todos los pins que intervienen en los sistemas descritos en el apartado 3.2. (Fig. 4.2):

Tabla 4.6. Pins LEDs

Nombre	# Pin
LEDG	90
LEDR	91

Tabla 4.7. Pins Audio

Nombre	# pin
HPP32	46
HPN32	47
HPP8	56
HPN8	57
HSOR	49
HSOL	50
HSMICIP	52
HSMICIN	53
MICIN	62
MICIP	63

Tabla 4.8. Pins GPS

Nombre	# pin
IO7	107
STDBYCAM	66
TSPACTEXT	67
INT1	111
IO6	65
IO5	80
SDAI2C	93
SCL12C	14
CLK32K	9
VRIO	2

Tabla 4.9. Pins Alimentación

Nombre	# pin
VRIO	2
Backup	12

Tabla 4.10. Control de la energía

Nombre	# pin
MOD_ON_STATE	27
MOD_UART_STATE	25
MOD_ON_CMD	113
MOD_OFF_CMD	76
DTE_UART_STATE	92
MOD_FLOW_STATE	26
MOD_RESET_STATE	24
MOD_RESET_CMD	81
DTE_RESET_CMD	106

Tabla 4.11. Pins SIM

Nombre	# pin
SIMCLK	4
SIMRST	5
SIMVCC	6
SIMIO	7
SIMCD	114

Tabla 4.12. Pins UART1. V24:

Nombre	# pin
CTS	82
RTS	83
DSR	94
DTR	95
TXD1	101
RXD1	105
RI	109
DCD	110

Tabla 4.13. Pins UART2

Nombre	# pin
TXD2	103
RXD2	104

Tabla 4.14. pins DATA/COMMAND

Nombre	# pin
MOD_ESCAPE*_CMD	18
DTE_ESCAPE_CMD	23

Tabla 4.15. Pins CLOCKS

Nombre	# pin
CLK32k	9
CLK13M	16

CAPÍTULO 5. DISEÑO DE LA PLACA

5.1. Introducción

El objetivo de esta placa es permitir la comunicación entre el microprocesador y el MO200; que el microprocesador pueda recibir y procesar los datos que llegan del MO200 y que el MO200 entienda las órdenes y peticiones del microprocesador. Para que esto suceda hay que tener en cuenta que trabajan a niveles distintos de tensión, el microprocesador trabaja a 3,3V [29] y las señales del modulo GPRS van de 2,03V a 3,2V. Además el GPRS necesita una alimentación de 3,8V [18] mientras que todos los demás elementos necesitan 3,3V por tanto se debe incluir en el diseño un regulador que adapte estos niveles.

Se ha decidido poner un conector serie (DB9) en la placa para tener una salida al exterior. Este conector será controlado por un RS232. Además debe haber un dispositivo que controle este puerto serie y también la comunicación serie entre el micro y el GPRS. A continuación, se muestra un cuadro resumen de todos los elementos que incluimos en la placa para el correcto funcionamiento del sistema:

Tabla 5.1. Componentes de la placa

Dispositivo	Part #	Fabricante
Convertor de nivel 16 canales	ADG3247	Analog Devices
Convertor de nivel 10 canales	ADG3246	Analog Devices
Regulador de tensión	LP3966ES	National Semiconductor
Dual UART	TL16C752B	Texas Instruments
Puerta OR	SN74LV32A	Texas Instruments
Controlador Puerto Serie	MAX3237E	Maxim
Conector Board to Board MO200	AVX 20 5604 120 222 829	Kyocera-ELCO
Lector SIM	AMC-ICC-313	Alpha Micro Components
2 conectores Board to Board (μ C)	AMP177984_5	RS Components
Decodificador	SN74LVC1G139	Texas Instruments
Módulo microprocesador	A9M9750	Forth-Systeme GmbH

A continuación se muestra un esquema de bloques de todos los elementos y de como se interconectan entre si:

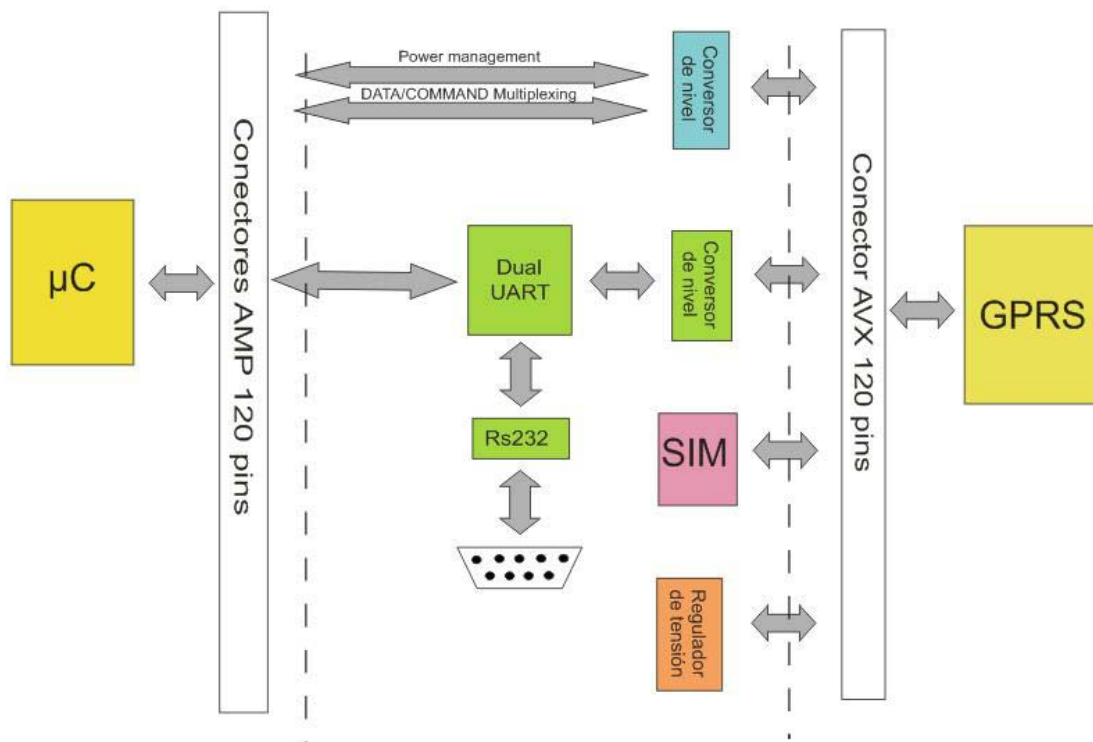


Fig. 5.1 Esquema placa

La conexión entre la DualUART y el conector del microprocesador se ha realizado mediante una puerta OR y un decodificador para utilizar el mínimo número posible de pins del microprocesador.

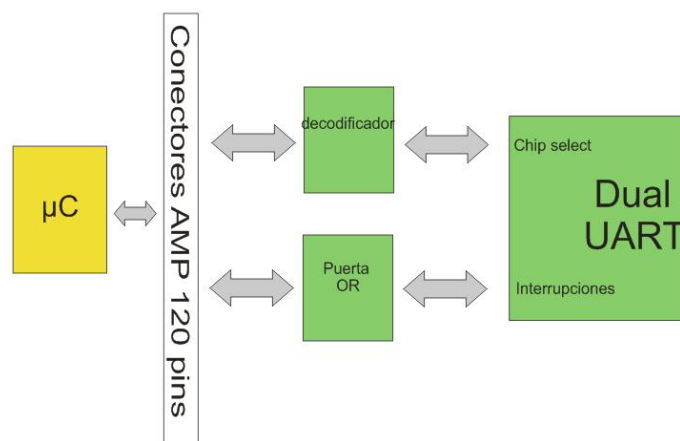


Fig. 5.2 Detalle conexión UART

5.2. Componentes

En esta sección se explica en detalle cada componente incluido en la placa, los aspectos más relevantes de su funcionamiento y de las conexiones que deben realizarse entre ellos.

5.2.1. Conversores de nivel

Tanto el microprocesador como los demás elementos que se han incluido en la placa funcionan a 3,3 V. Para que estos puedan comunicarse con el módulo GPRS se debe realizar la conversión de nivel, de 3,3V a un nivel que entienda el MO200.

En los pins de entrada al módulo V_H tiene un rango de 2,03V a 3,2V y en los pins de salida V_H mínima es 2,32V [18]. Los conversores de Analog Devices ADG3247 y ADG3246 permiten la conversión de 3,3V a 2,5V, que se adecua a las necesidades tanto para las señales de entrada como para las de salida.

5.2.1.1. ADG3247

Este dispositivo dispone de 16 canales. Lo utilizamos para las señales de control del módulo GPRS (Fig. 4.3). Como son solo 11 los canales que se van a convertir quedan los canales del 11 al 15 sin utilizar. El conversor está organizado en 2 bus switches digitales de 8 bits con habilitadores de BUS separados. Así este conversor se puede utilizar o bien como dos bus switches digitales de 8 bits o bien como un bus switch de 16 bits. Estos switches permiten que las señales sean convertidas cuando están en ON. En modo OFF la función de conversión queda bloqueada [20].

Para conseguir la conversión de 3,3V a 2,5V, se alimenta el dispositivo a 3,3V, entonces se supondrá que las señales del puerto A están a 3,3V y las del puerto B a 2,5V (Fig. 5.3).

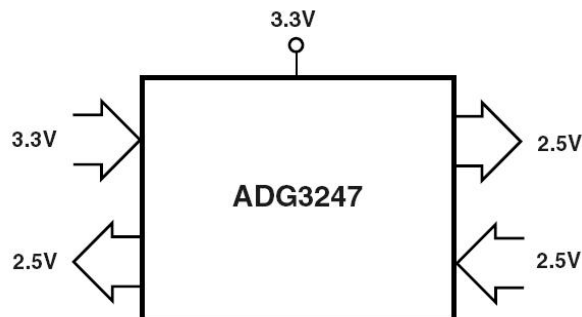


Fig. 5.3 Conversión de nivel

Alimentando a 2,5V, la conversión que se obtiene es de 2,5V a 1,8V. Dispone también del pin SEL, que se activa a nivel bajo para conseguir la conversión de 3,3V a 1,8V. Cuando no se desea esta conversión SEL debe permanecer a nivel alto, conectado a Vcc.

Tabla 5.2. Selección de la conversión

Pin BEx	PIN SEL*	Resultado
L	L	Conversión de 3,3 a 1,8V
L	H	Conversión de 3,3 a 2,5V o de 2,5 a 1,8V
H	X	Desconectado

5.2.1.2. ADG3246

El funcionamiento de este dispositivo es exactamente igual que el anterior (fig. 5.3), la única diferencia entre los dos es que este tiene 10 canales y no tiene dos buses diferenciados. Tiene un solo BE, que permite la conversión cuando está a nivel bajo y desactiva las funciones de conversión cuando está a nivel alto. Lo utilizamos para las señales de la conexión V.24 (Fig. 4.8). Como solo se necesitan 8 canales, habrá 2 de sobra (el 8 y el 9).

5.2.2. Regulador de tensión

Como ya se ha explicado anteriormente, todos los elementos del sistema funcionan al mismo nivel de tensión menos el módulo GPRS. Por eso se necesita un regulador de tensión que adapte la tensión de alimentación del sistema en general a la del módulo. El modelo es de National Semiconductor.

5.2.2.1. LP3966ES_ADJ

Este dispositivo va a transformar la tensión de alimentación en los 3,8V que necesita el módulo GPRS (3,8V es el valor nominal, la tensión de alimentación puede variar entre 3,45V y 4,5V). Se ha elegido este regulador porque puede llegar a proporcionar una corriente máxima de operación de 3A [22], mayor que el máximo de consumo del MO200 que es de 2A (el rango de consumo de corriente va de 0,1 a 2A).

En la tabla siguiente se listan todos los pins del dispositivo:

Tabla 5.3. Pins del regulador

Pin	Nombre	Función
1	SD	Modo Shutdown
2	Vin	Tensión de entrada
3	GND	Ground
4	Vout	Tensión de salida
5	ADJ	Output Adjustable pin

Tanto en Vin como en Vout, el rango de tensiones que acepta va de -0,3V a 7,5V. El pin ADJ servirá para ajustar la tensión exacta que se necesita a la salida, determinando previamente el valor de R2 mediante la fórmula siguiente:

$$R2 = R1 \left(\frac{V_{OUT}}{1,216} - 1 \right) \quad (5.1)$$

R1 vale siempre 10kΩ y Vout = 3,8V

$$R2 = 10k\Omega \cdot \left(\frac{3,8V}{1,216} - 1 \right) \quad \rightarrow \quad R2 = 21,25 \text{ k}\Omega$$

Además de R1 y R2 necesitamos añadir también una serie de condensadores con unas características determinadas:

→ Un condensador de entrada, C_{IN}, para mantener la estabilidad del dispositivo, y cuyo valor mínimo es de 68 μF. En cualquier caso, para determinar un valor aceptable para este condensador podemos utilizar la siguiente fórmula:

$$C_{IN} \text{ ESR (m}\Omega\text{)} / C_{IN} (\mu\text{F}) \leq 1,5 \quad (5.2)$$

donde ESR es la resistencia serie equivalente del condensador (cada condensador tiene la suya propia).

El mejor condensador para C_{IN} es de tipo cerámico. Estos condensadores tienen muy buen comportamiento en AC eliminando el ruido a frecuencias muy altas gracias a su baja ESR. Dentro de los cerámicos, los condensadores que mejor comportamiento presentan en todo el rango de temperaturas y voltajes son los dieléctricos X7R (Fig. 5.4) y X5R.



Fig. 5.4 Condensador cerámico X7R

→ Un condensador de salida, C_{OUT} , para asegurar la estabilidad de lazo. El condensador debe conectarse directamente a masa y a la salida sin que haya corrientes adicionales en su pista. Esta capacidad debe ser mayor que $33 \mu\text{F}$, sin límite superior, aunque hay que tener muy en cuenta el valor de la ESR, que debe cumplir:

$$0,2 \Omega \leq C_{OUT} \text{ ESR} \leq 5 \Omega \quad (5.3)$$

En este caso, el límite inferior de $200 \text{ m}\Omega$, hace inadecuado el uso de un condensador cerámico para C_{OUT} . Además, los condensadores cerámicos no son muy estables en cuanto a variación de temperaturas se refiere.

La mejor opción es elegir un condensador de Tantalio porque su ESR se acerca mucho al valor ideal requerido para la compensación de lazo [22]. Si el condensador es de buena calidad tendrá un buen comportamiento con la variación de temperatura.

→ C_F , que debe tener un valor comprendido entre 68 y 100 pF . Elegimos 68 pF al ser el valor que aparece en el datasheet.

El montaje del regulador con todos sus condensadores queda de la siguiente manera:

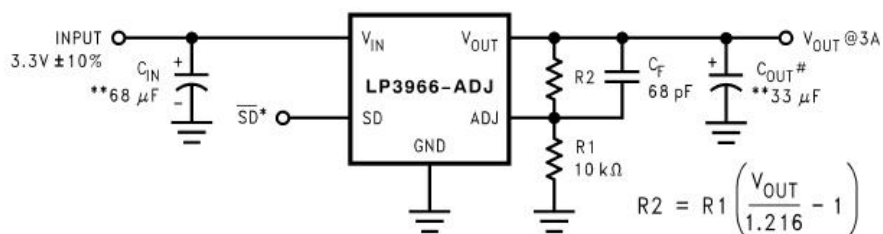


Fig. 5.5 Montaje regulador

Aunque en este esquema el pin SD aparece desconectado, en realidad se conecta al microprocesador quien tendrá que enviar una señal a bajo nivel para desconectar el regulador (modo Shutdown).

La función Shutdown permite encender o apagar el regulador mientras se sigue aplicando tensión en la entrada. Cuando está en modo Shutdown, la corriente que suministra cae a un nivel muy bajo ya que la salida está deshabilitada pero el circuito interno sigue operativo. Cuando se vuelve a encender (este modelo necesita un 1 lógico), el regulador recupera su funcionalidad mucho más rápido que si se hubiera apagado del todo y después se hubiera vuelto a encender [23].

5.2.3. UART

El fabricante del módulo GPRS recomienda utilizar una conexión V.24 completa para la comunicación con el Host. Se ha elegido incluir un dispositivo dual UART de Texas Instruments aun teniendo el del microprocesador ya que permite implementar más de un enlace y así quedan libres los del microprocesador.

5.2.3.1. TL16C752B

Este dispositivo trabaja a 3,3V de tensión igual que el microprocesador, razón por la que se ha elegido este modelo, y por tanto no se necesita modificar la tensión de alimentación mediante reguladores. Implementa en un mismo chip dos enlaces serie. Uno de ellos se destina a la comunicación del GPRS con el microprocesador, el otro servirá para las comunicaciones de la placa con el exterior mediante un conector (DB9) que se incluye en la placa.

La velocidad de transmisión de datos oscila entre 1,5 y 3 Mbps dependiendo de la frecuencia del reloj que se le conecte [24]. Utilizando un cristal (clock a 24 MHz) la velocidad es de 1,5 Mbps mientras que para un oscilador o una fuente de reloj (clock a 48 MHz), la velocidad es de 3 Mbps. En cualquier caso, se puede obtener la velocidad deseada gracias a un divisor programable y junto con la frecuencia de reloj de entrada.

Dispone de dos FIFOs de 64 bytes, una para recepción y otra para transmisión (Fig. 5.6). Los niveles de trigger de ambas FIFOs para DMA y generación de interrupciones son programables. En la FIFO de recepción, además, se pueden programar los niveles de trigger para el control de flujo tanto software como hardware, estos niveles se almacenan en el TCR (Transmission Control Register). El control de flujo automático por software se realiza utilizando los caracteres programables Xon/Xoff. El control por hardware dispone de auto-CTS y auto-RTS.

Utilizando *auto-CTS* el circuito transmisor tiene que comprobar el estado del pin CTS, si está activo entonces puede transmitir. Para que no se envíe el siguiente bit, CTS tiene que estar inactivo antes de la mitad del segundo bit de stop que ya se esté enviando. Este sistema permite a la UART controlar su transmisor y reduce considerablemente las interrupciones al host.

En modo *auto-RTS*, RTS está activo si el nivel de la FIFO receptora está por debajo del nivel de trigger de HALT. Cuando se alcanza este nivel RTS se desactiva automáticamente aunque el dispositivo que envía los datos no se entera hasta que está enviando el siguiente byte. RTS se activa automáticamente una vez que se ha alcanzado el nivel de trigger de RESUME y así el dispositivo que estaba enviando los datos puede continuar con la transmisión.

El dispositivo consta de 20 registros. Las principales funciones que realizan son de control de las FIFOs, indicación y control de los errores ocurridos, del estado de operación, control de la interfaz del MODEM y control de las interrupciones.

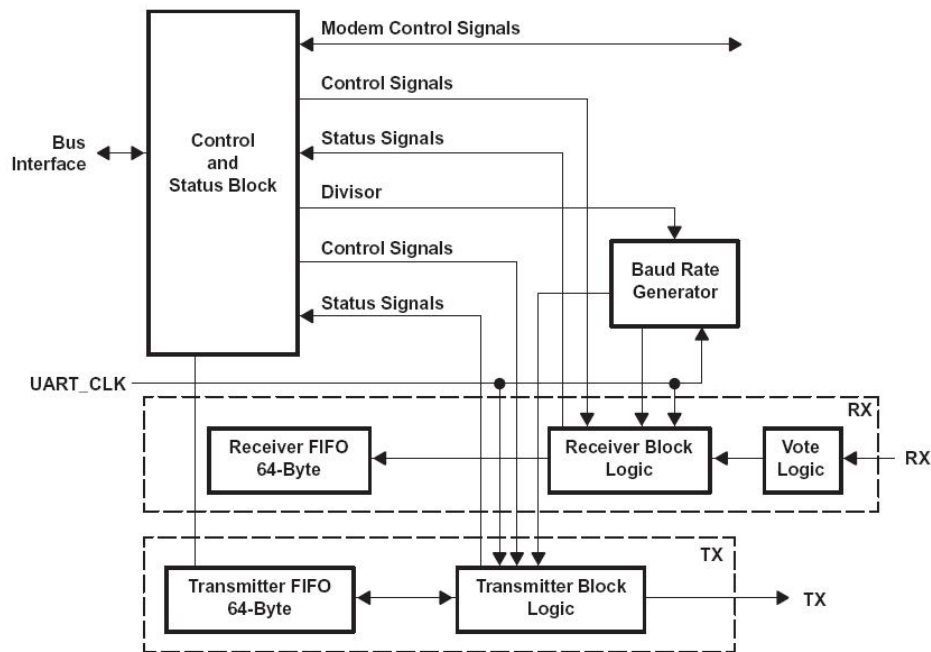


Fig. 5.6 Diagrama de bloques del TL16C752B

Los pins de Chip Select (CSA y CSB) sirven para seleccionar uno de los dos enlaces serie. Para no utilizar más CS de los necesarios en el microprocesador (con uno basta), la selección de puerto se realiza mediante un decodificador. Así la señal CS de microprocesador está siempre activa y será otra de las entradas del decodificador la que seleccione el puerto. La figura 5.7 muestra un ejemplo de cómo se realizaría la conexión:

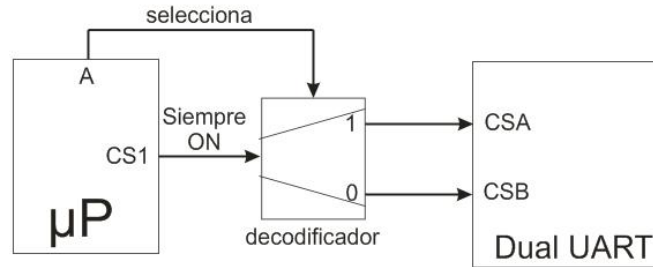


Fig. 5.7 Chip Select mediante decodificador

Cuando la UART tiene que interrumpir al microprocesador utiliza las señales INTA e INTB. Igual que pasaba con el CS, se puede economizar el número de pins del microprocesador que se usan mediante una puerta OR. Si llevamos INTA e INTB a la entrada de la OR, siempre que una de las dos quiera interrumpir el microprocesador se va a enterar, aunque no sepa cual ha sido la que ha interrumpido.

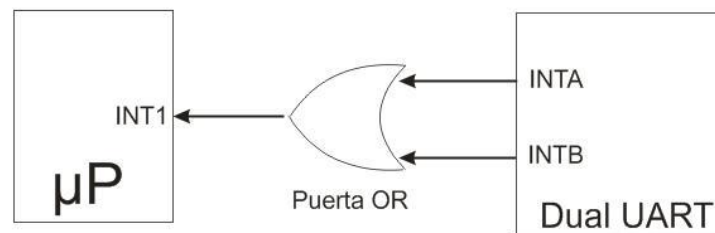


Fig. 5.8 Interrupción mediante una puerta OR

5.2.4. Controlador Puerto Serie

El RS232 va a controlar las señales que entran y salen de la placa a través del conector serie DB9.

5.2.4.1. MAX3237E

Este dispositivo de Maxim trabaja a 3,3V y tiene protección contra descarga electrostática (+/-15 kV) en todos los pins, incluidos los de I/O lógicos [25].

Tiene también la opción de operar en modo MegaBaud. Como su nombre indica, este modo permite una velocidad de transmisión de datos de 1Mbps. En modo normal, la velocidad permitida alcanza los 250 kbps, mayor que los 115,2 kbps que ofrece el módulo GPRS, y por tanto hay de sobra para esta aplicación. En modo SHDN, la corriente inyectada cae hasta 10 nA y se deshabilitan todas las salidas del dispositivo.

También aquí necesitamos añadir condensadores externos para su correcto funcionamiento:

→ C1-C4, el tipo de condensador utilizado no tiene gran repercusión en el funcionamiento del dispositivo. Para una alimentación de 3,3V se deben utilizar capacidades de 0,1 μF .

Tabla 5.4. Condensadores RS232 según V_{in}

V_{cc} (V)	C1 (μF)	C2, C3, C4 (μF)
3,0 – 3,6	0,22	0,22
3,15 – 3,6	0,1	0,1
4,5 – 5,5	0,047	0,33

→ C_{BYPASS} , normalmente un valor adecuado son 0,1 μF . En aplicaciones sensibles al ruido de la fuente de alimentación el valor de este condensador debe ser igual al de C1, que en este caso es también 0,1 μF puesto que se alimenta a 3,3V. Este condensador debe estar situado lo más próximo posible a la alimentación.

La conexión de los condensadores queda de la siguiente manera:

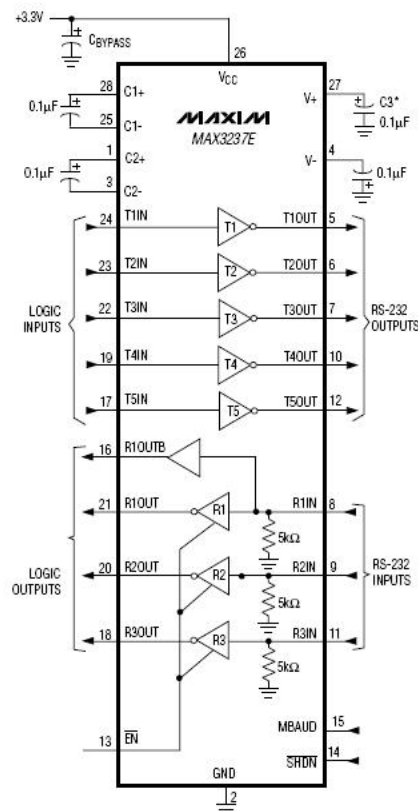


Fig. 5.9 Esquema del dispositivo. Conexión de los condensadores

5.2.5. Conector SIM

El conector que se va a utilizar es de Alpha Micro y consta de 6 pins. Como ya se he explicado en el capítulo anterior, la conexión se realiza sin detección de tarjeta (Fig. 4.6).

5.2.5.1. AMC-ICC-313

La tarjeta SIM es un tipo de Smart Card, que entre otras cosas se caracteriza por tener un Circuito Integrado (IC) con un microprocesador y un espacio de memoria. Este chip puede almacenar datos, modificar los datos almacenados y realizar cálculos más o menos complejos.

La interfaz de conexión de la SIM es compatible con la normativa ISO 7816-3 IC que especifica las características, físicas y eléctricas, que debe tener la tarjeta, las señales de reloj, los protocolos para comunicarse con ella (formato de los comandos, respuestas, etc.) [35].

Contactos:

La tarjeta debe tener 8 contactos de los cuales, de momento, solo se usan 6:

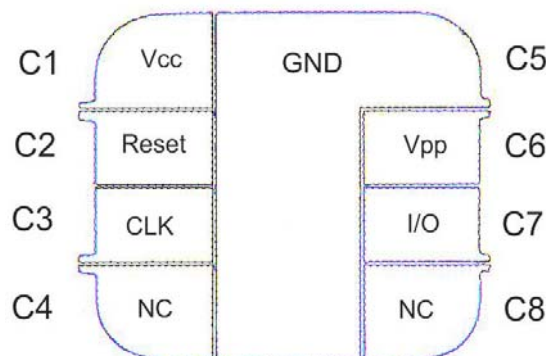


Fig. 5.10 Contactos Smart Card

Cada contacto tiene una función específica, tal como se detalla en la tabla 5.5.

Tabla 5.5. Contactos Smart Card

Contacto	Nombre	función
C1	Vcc	Tensión de alimentación necesaria para que funcione el IC (3V).
C2	Reset	Aplicando una cierta tensión en este contacto indicamos al chip que debe inicializar la tarjeta.
C3	CLK	Señal necesaria para que el circuito interno de la tarjeta funcione correctamente. La frecuencia de esta señal determina la velocidad de comunicación entre la tarjeta y el lector.
C4	NC/RFU	De momento no está conectado y se reserva para usos futuros (RFU)
C5	GND	Conectada a 0V.
C6	Vpp	Este contacto estaba destinado en un principio a conectarse a una segunda alimentación de 5V que permitía a la tarjeta guardar los datos en la memoria. Hoy en día esto se hace con la misma tensión que Vcc así que este contacto se conecta a Vcc.
C7	I/O	Comunicación con el exterior. La comunicación tiene que ser half-duplex porque solo hay un contacto y no se puede enviar y recibir datos al mismo tiempo.
C8	NC/RFU	De momento no está conectado y se reserva para usos futuros (RFU)

5.2.6. Puerta OR

Las señales de interrupción de la UART se activan a nivel alto, igual que la del microprocesador. Una manera muy fácil de simplificar la gestión de estas interrupciones es mediante una puerta OR. Conectando las señales de interrupción a la entrada de la OR y la del microprocesador a la salida solo será necesario utilizar un pin de éste último (Fig. 5.11).



Fig. 5.11 Puerta OR

5.2.6.1. SN74LV32A

Este dispositivo ofrece 4 puertas OR con un rango de operación que va de 2V a 5,5V. El encapsulado elegido es el PW porque es uno de los que ocupan menos espacio (6,6 x 5,1mm). Se utiliza la puerta 1 y las demás quedan libres.

La operación Booleana que realiza una puerta OR es:

$$Y = A + B \quad \text{o} \quad Y = \overline{\overline{A} \cdot \overline{B}} \quad (5.4)$$

y su tabla de verdad:

Tabla 5.6. Tabla de verdad puerta OR

Entradas		Salidas
A	B	Y
H	X	H
X	H	H
L	L	L

Siempre que uno de los dos puertos quiera interrumpir estará en H, y siempre que uno de los dos puertos, o los dos, estén en H, la salida, que es la señal de interrupción que entra al microprocesador, estará también en H. De esta manera el microprocesador sabrá que la UART quiere comunicarse con él aunque no sabrá cuál es el puerto que ha interrumpido.

5.2.7. Decodificador

El decodificador se conecta entre el microprocesador y la UART para implementar el chip select (Fig. 5.7). Tanto el Chip Select del microprocesador como los de la UART se activan por nivel bajo, esto debe tenerse en cuenta a la hora de elegir un decodificador ya que en el puerto seleccionado tiene que aparecer un 0 y no un 1.

5.2.7.1. 74LVC1G139

Este decodificador 2-to-4 de Texas Instruments cumple los requisitos nombrados en el párrafo anterior. Además cuando el dispositivo se desconecta, el circuito I_{off} desactiva las salidas evitando así las corrientes inversas que puedan dañarlo. Trabaja a 3,3V (rango de 1,6 a 5,5V) igual que todos los demás componentes de la placa. El encapsulado elegido ha sido el DCU porque es el más pequeño y no hay que olvidar que el espacio disponible en el UAV es reducido. Dos de las salidas no se conectarán ya que el sistema a implementar sólo necesita dos (CSA y CSB).

Las entradas del decodificador son A y B. En la entrada A se conecta el CS del microprocesador que siempre que esté en funcionamiento estará a nivel bajo. En la entrada B se conecta la señal que decidirá con qué puerto de la UART se va a comunicar. Esta señal es uno de los pins de dirección de memoria del microprocesador. El puerto elegido dependerá de su nivel lógico.

A continuación se muestra la tabla de verdad del decodificador para entender como se va a llevar a cabo el proceso de selección de puerto:

Tabla 5.7. Tabla de verdad del decodificador 2-4

Select Inputs		Outputs			
B	A	Y0	Y1	Y2	Y3
H	L	L	H	H	H
H	H	H	L	H	H
L	L	H	H	L	H
L	H	H	H	H	L

B es la señal de dirección de memoria que envía el microprocesador. Cuando está a nivel alto (H) se selecciona la UART A y cuando está a nivel bajo (L) se selecciona la B. Por otro lado, la entrada A decide si el puerto se selecciona o no. Cuando está a nivel L el puerto está seleccionado en cambio en H no lo está.

Anteriormente se ha dicho que tanto los CS de la UART como el del microprocesador se activan a nivel bajo; es debido a esto que la salida asignada a cada puerto debe tener el mismo nivel de señal que A cuando dicho puerto esté seleccionado por B. Cuando B está a nivel H solo los niveles de Y0

coinciden con los de A y cuando B está a nivel L solo Y2 es igual que A. Por tanto serán estos los canales que se usarán.

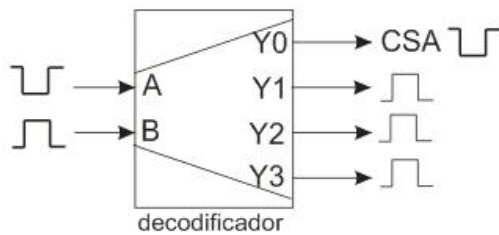


Fig. 4.12 Selección UART A

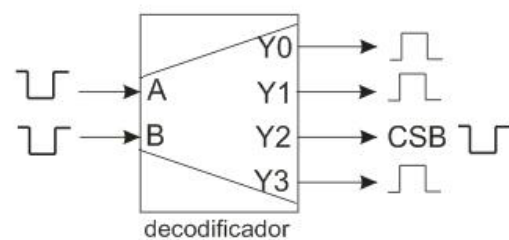


Fig. 4.13 Selección UART B

Las figuras 4.12 y 4.13 muestran gráficamente en qué nivel lógico se encuentran los pins del decodificador cuando se selecciona cada uno de los puertos.

5.2.8. Conectores

Tanto el modulo A9M9750 como el MO200 disponen de unos conectores específicos para poderse conectar a una placa madre. En los apartados siguientes se explican las características de los modelos que utiliza cada dispositivo.

5.2.8.1. AVX 20 5604 120 222 829

Este conector (Board to Board) de Kyocera sirve para conectar a la placa diseñada la del MO200. Dispone de 120 pins; en el capítulo 4 se especifican las señales que ocupan cada pin.



Fig. 5.14 Conector 120 pins AVX 5604

5.2.8.2. AMP 177984_5

Para conectar el módulo A9M9750 a nuestra placa utilizamos dos conectores de este tipo.



Fig. 5.15 Conector 120 pins AMP 177984_5

La distancia que se debe dejar entre ambos viene especificada por el fabricante del módulo [36]:

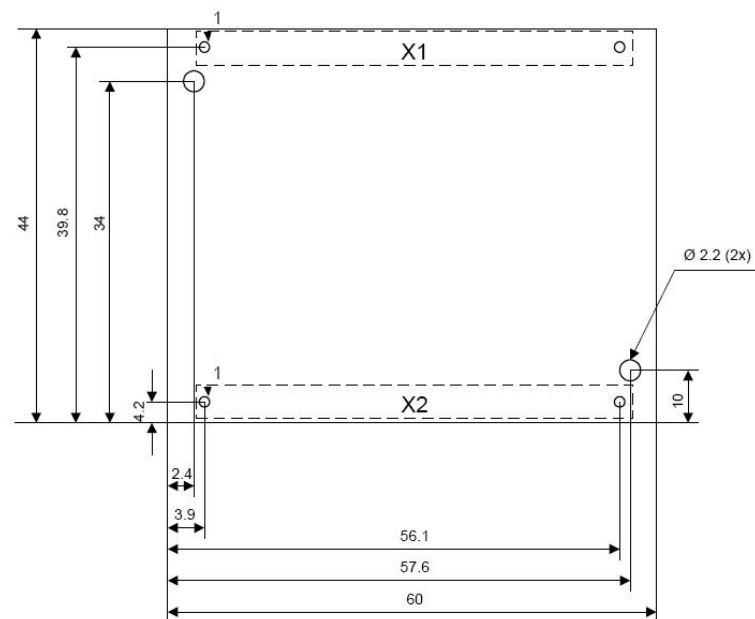


Fig. 5.16 Distancias conectores X1 y X2 del A9M9750

5.2.9. Módulo A9M9750

Éste módulo (Fig.5.17) viene con una placa de desarrollo incluida. En un principio se pensó utilizar todo el conjunto pero después de estudiar la documentación incluida sobre sus funciones y esquemáticos se descartó esta posibilidad; todos los pins necesarios de los conectores del apartado anterior están ocupados por los sistemas de la placa de desarrollo.

5.2.9.1. Módulo A9M9750

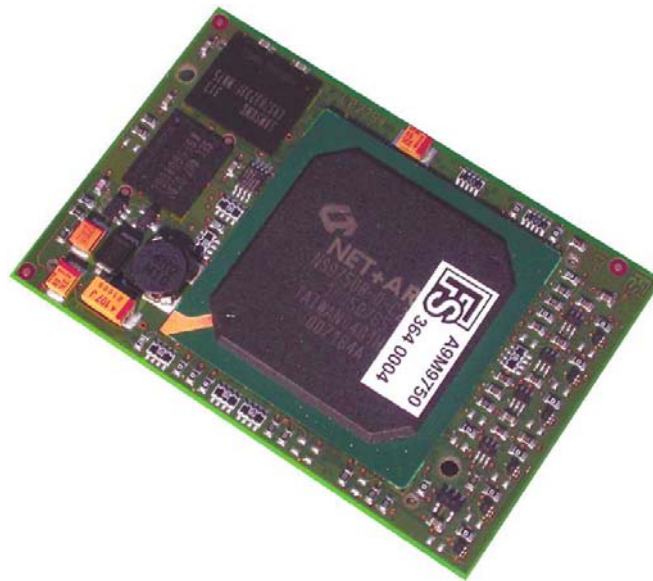


Fig. 5.17 imagen del módulo A9M9750

Características:

- Microcontrolador NS9750 de NetSilicon
- ARM26EJ-S core a 200 MHz
- Controlador LCD
- Puerto USB
- Bus PCI
- Memoria Flash de 32/64/128 MBytes
- Memoria SDRAM de 16/32/64 MBytes
- 4 canales UART
- bus I2C
- bus SPI
- Memoria EEPROM serie para guardar los datos de configuración
- RTC
- Fuente de alimentación a 3,3V
- 10/100 Ethernet MAC y PHY
- 16 timers/contadores programables

Todas las funcionalidades mencionadas en la lista anterior se representan en la figura 5.18.

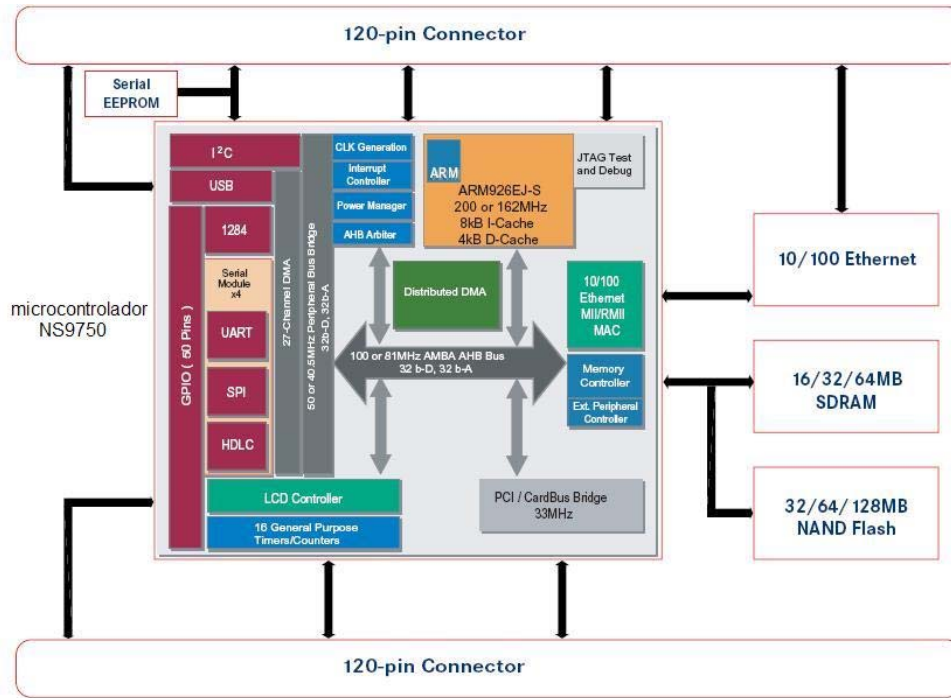


Fig. 5.18 esquema módulo A9M9750

Éste módulo tiene 2 conectores de 120 pins cada uno que sirven para poder conectarlo a otra placa (Fig. 5.15 y Fig. 5.16). Las 4 UARTs de que dispone el módulo están multiplexadas con otras señales en estos conectores por eso se ha decidido no utilizarlas.

Además de las UARTs, tampoco se usarán la pantalla LCD y por tanto muchos de los pins que ocupa su controlador podrán ser utilizados como pins de propósito general (GPIOs).

Su reducido tamaño, 60x44 mm, lo hace ideal para insertarlo en un espacio reducido como es el caso de un UAV.

5.2.9.2. Microprocesador NS9750



Fig. 5.19 microprocesador NS9750

El NS9750 está basado en un procesador RISC ARM926EJ-S de 32 bits, puede operar a 200 MHz de frecuencia y se alimenta a 3,3V de tensión. Está construido sobre el ARM9 más potente de ARM [29]. Este microprocesador y los de su familia están concebidos exclusivamente para funcionar en equipos electrónicos embebidos de alto rendimiento.

Aporta un alto nivel de integración de periféricos que proporciona las interfaces necesarias para conectar un gran número de dispositivos comerciales, los cuatro puertos serie entre otros. Tiene 50 pins de propósito general I/O, multiplexados en el módulo A9M9750 con otras señales.

El microprocesador tiene un sistema de control de energía muy eficiente ya que estando operativo desconecta los módulos que no se están utilizando; en modo Sleep, solamente quedan habilitados los módulos que permiten realizar un wakeup del sistema [28].

Dispone de pila TCP/IP, indispensable para las comunicaciones vía Internet del MO200. También dos memorias cache de 8 y 4 kB para el almacenamiento de instrucciones y datos respectivamente.

5.3. Esquemáticos de la placa

Para diseñar las conexiones entre los diferentes componentes de la placa, se ha utilizado el programa P-CAD. En este programa, primero se dibujan los símbolos de los componentes sin tener en cuenta la forma exacta de los mismos y después se hace un esquema con las medidas reales. El símbolo y el esquema con las medidas reales se juntan para formar el componente. En los esquemáticos que aparecen a continuación se muestran los símbolos.

5.3.1. Conector AVX 20 5604 120 222 829 + Vbat

En el esquemático del conector AVX 5604 del MO200 se ha incluido también el layout de Vbat (Fig. 5.5) para poder medir correctamente las distancias que se deben dejar entre ambos. En este esquemático se incluyen también los condensadores y resistencias que necesitan determinadas señales.

Las señales de la SIM van directamente al conector de ésta (apartado 5.3.8), sin convertir el nivel de tensión. Las señales de control se conectan al convertor de 16 canales (apartado 5.3.2) y las del enlace V24 al convertor de 10 canales (apartado 5.3.3).

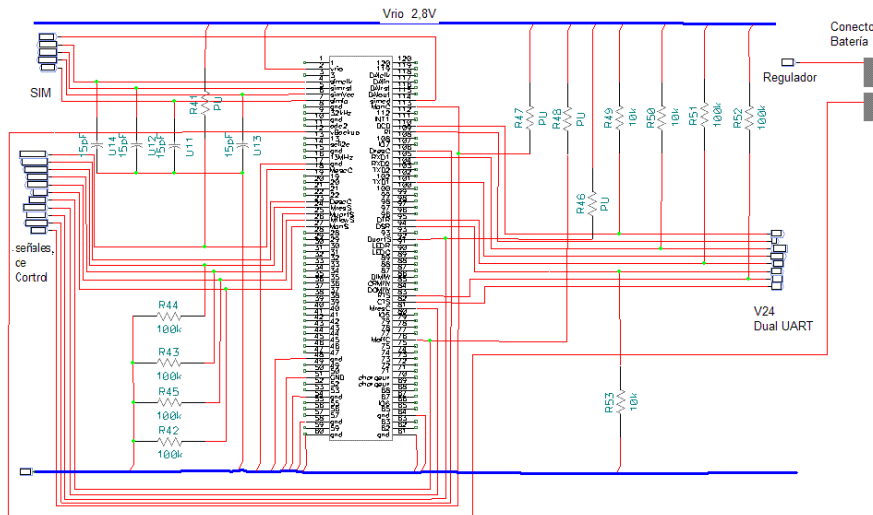


Fig. 5.20 Esquemático conexiones al MO200

5.3.2. Conversor de nivel ADG3247

En la figura 5.21 se pueden observar las señales que vienen o van al microprocesador desde el MO200 y pasando por el convertor de nivel de 16 canales. Los BE se han conectado a GPIOs del A9M9750 quien tendrá que mantener esos pins a nivel bajo lógico para habilitar la conversión. Las señales en la parte izquierda de la imagen se conectan al microprocesador y están a 3,3V de tensión; las que están a la derecha van al MO200 y están a 2,5V (Fig. 5.3). Los canales ocupados son del 0 al 10 y quedan libres del 11 al 15. El convertor se alimenta a 3,3V para que la conversión sea de 3,3 a 2,5V, como ya se explicó en el apartado 5.2.2. Las resistencias que necesitan algunas de estas señales ya se incluyen en el esquemático del conector AVX 5604 (Fig. 5.20).

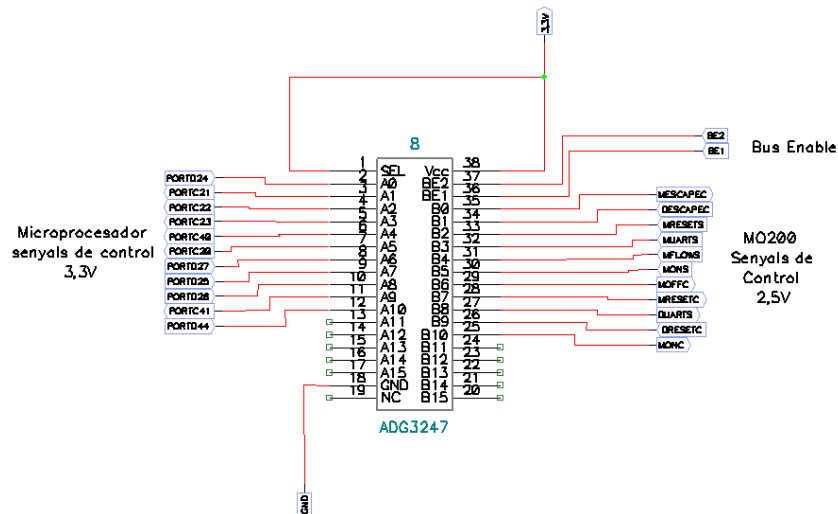


Fig. 5.21 Esquemático convertor ADG3247

5.3.3. Conversor de nivel ADG3246

El ADG3246 se encarga de convertir las señales del enlace V24. Las de la parte de la izquierda de la figura 5.22 son las que van al puerto A de la DualUART y están a 3,3V de tensión; las de la parte de la derecha son las que se conectan al MO200 a 2,5V. En este conversor han quedado libres los canales 8 y 9. Se alimenta a 3,3V por la misma razón que el ADG3247. El pin de BE se ha conectado a un pin de propósito general del A9M9750 igual que el ADG3247.

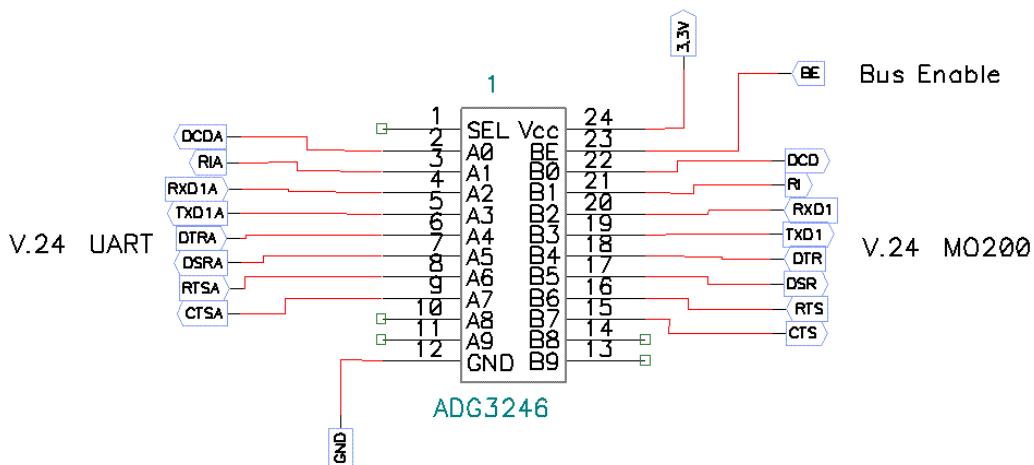


Fig. 5.22 Esquemático conversor ADG3246

5.3.4. Dual UART

La alimentación de este dispositivo es a 3,3V y tiene 7 grupos de señales que se han utilizado:

En primer lugar se encuentran 3 pines de direcciones (A0 pin 29, A1 pin 28 y A2 pin 27). Estos pines sirven para acceder a los diferentes registros internos de la UART. Los 8 bits de datos se conectan directamente al microprocesador (pins 1-3 y 44-48).

A continuación las señales INTA (pin 31) e INTB (pin 30) que interrumpen al microprocesador cuando uno de los dos puertos tiene algo que decirle. Estas interrupciones no se conectan directamente al microprocesador sino que primero pasan por una puerta OR (Fig. 5.2). Ésta convierte las dos señales en una sola hacia el microprocesador de manera que la señal se active independientemente de quien interrumpa.

Las señales que seleccionan un puerto u otro son los Chip Selects: CSA (pin 10) y CSB (pin 11). Igual que las de interrupciones, estas señales no se

conectan directamente al microprocesador sino que primero tienen que pasar por un decodificador, así se ahorran pins (Fig. 5.2).

IOR (pin 19) permite descargar en D0-D7 los datos que se encuentran en el registro interno indicado por A0-A2. IOW (pin 15) realiza la función contraria descargando en el registro interno indicado por A0-A2 y CSA y CSB los datos que se encuentran en D0-D7. Estas dos señales se conectan directamente al microprocesador.

Las señales de las conexiones serie dependen del puerto al que pertenezcan, las del puerto A se conectan al convertor de nivel ADG3246 y las del puerto B al RS232.

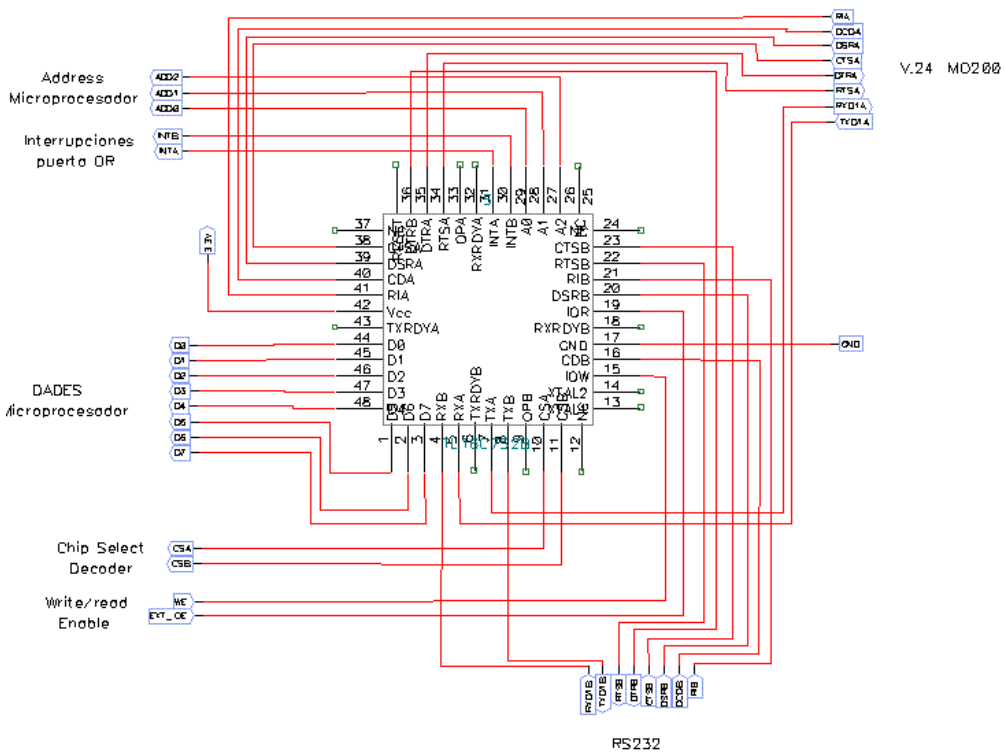


Fig. 5.23 Esquemático DualUART

Tabla 5.8. pins utilizados en la conexión de la DualUART

Señal		# pin	
		Puerto A	Puerto B
RS232	RXD	5	4
	TXD	7	8
	DSR	39	20
	CTS	38	23
	DTR	34	35
	RTS	33	22
	RI	41	21
	DCD	40	16
Address	A0	28	
	A1	27	
	A2	26	
Datos	D0-D4	44-48	
	D5-D7	1-3	
W/R enable	IOW	15	
	IOR	19	
Chip Select	CSA_B	10	11
Interrupciones	INTA_B	30	29

5.3.5. RS232 MAX3237E

Este dispositivo tiene dos grupos de señales. A la izquierda de la figura 5.24 se encuentran las señales que se conectan al puerto B de la DualUART; en la parte de la derecha están las señales que van al conector serie (DB9). Internamente lo que hace este dispositivo no es más que dirigir las señales en el sentido que les toca. Cada pin ya tiene definido su sentido (IN o OUT) y a la hora de conectar las diferentes señales solo se debe tener en cuenta este sentido (Fig. 5.9). Por supuesto debe respetarse su continuidad por lo que las señales homólogas de la DualUART y del conector DB9 deben utilizar el mismo camino (T1, T2, T3, T4, T5, R1, R2 o R3).

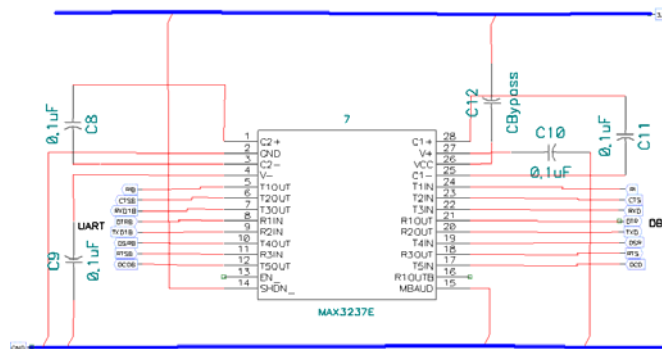


Fig. 5.24 Esquemático MAX3237E

5.3.6. DB9

Éste es el elemento que permite que la placa se conecte con el exterior. A él llegan las señales del puerto B de la DualUART que han pasado por el controlador MAX3237E. Las señales de puerto serie se conectan igual en todos los DB9s, según el estándar (Fig. 5.25).

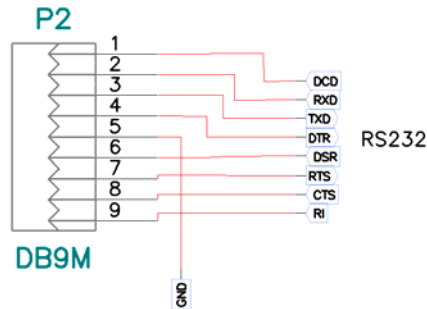


Fig. 5.25 Esquemático conector DB9

Tabla 4.9. Estándar de conexión DB9

Señal	# pin
RXD	2
TXD	3
DSR	6
CTS	8
DTR	4
RTS	7
RI	9
DCD	1

5.3.7. Regulador

Este regulador dispone del pin ADJ que permite adaptar de forma más precisa la tensión de salida a través de la resistencia R2 (Fig. 5.5). La potencia máxima que es capaz de entregar es de 3A, de sobra para los 2A como máximo que consume el MO200. La señal de salida se conecta directamente al layout de Vbat (Fig. 4.5) y la señal de entrada a la alimentación del sistema. Para implementar el modo Shutdown, se conecta el pin SD directamente al microprocesador que enviará una señal cuyo nivel lógico dependerá de si se quiere 'dormir' o 'despertar' al regulador.

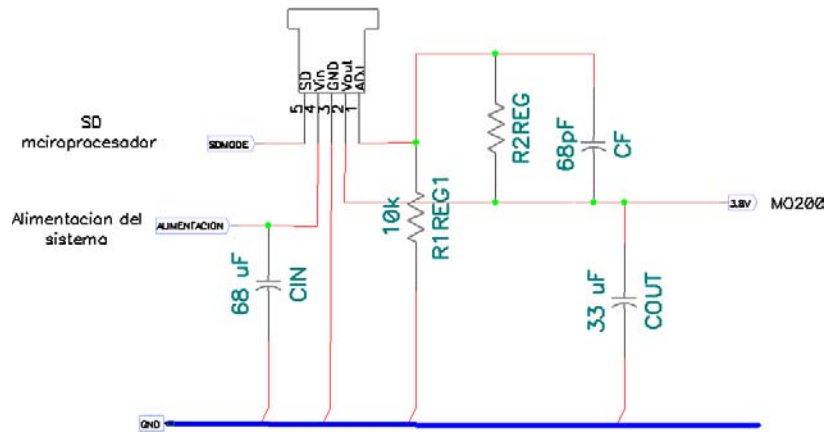


Fig. 5.26 Esquemático regulador LP3966-ADJ

5.3.8. SIM

Las señales de la SIM se conectan directamente al MO200 (Fig. 4.6), quien la alimenta a través del pin SIMVCC (pin 9). Es por eso que no ha necesitado pasar por un convertor de nivel. Los pins que quedan sin conectar (4, 5 y 6) sirven para fijar mejor el conector a la placa. Los condensadores que necesitan estas señales se incluyen en el esquemático del conector AVX 5604 del MO200 (Fig.5.20). El fabricante recomienda que se coloquen lo más cerca posible del MO200.

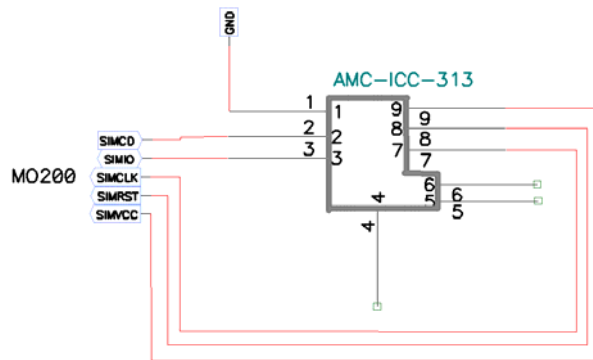


Fig. 5.27 Esquemático SIM

5.3.9. AMP 177984_5 (X1 y X2)

En el esquemático de la documentación del microprocesador, la numeración de los pins de estos conectores no sigue el mismo camino que en este esquemático sino que los pins pares se encuentran en la parte de la derecha del conector y los impares en la izquierda [33]. Teniendo esto en cuenta, las

conexiones no se han hecho basándose en el número de pin sino en la posición que ocupan en el conector.

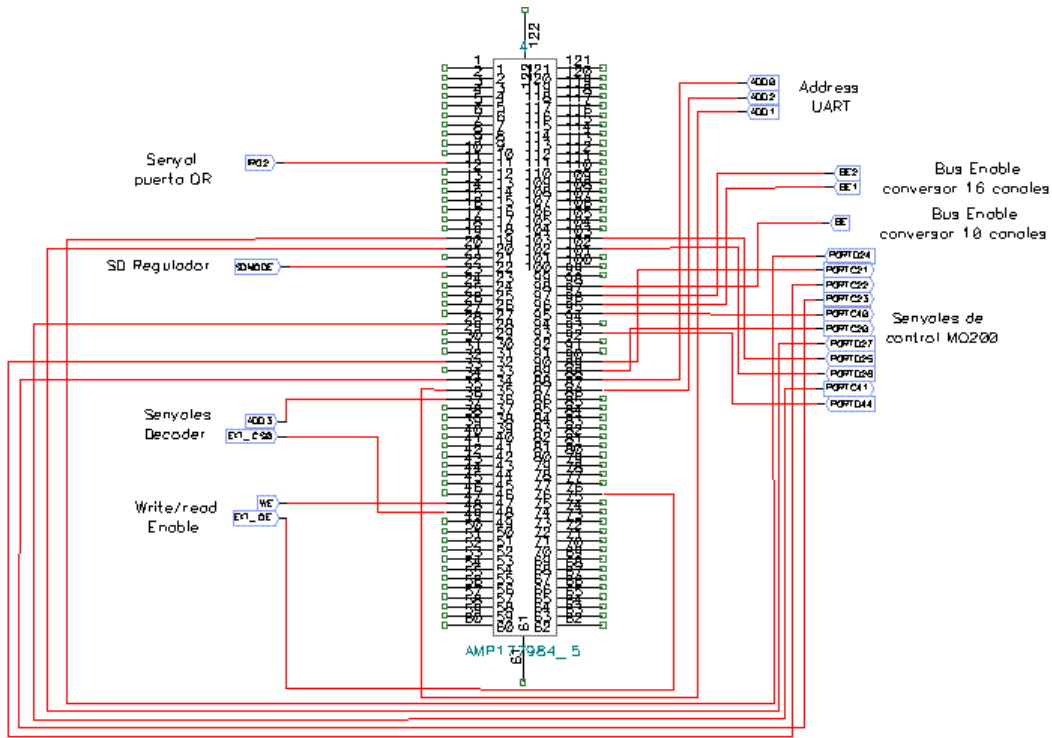


Fig. 5.28 Conector X1 del microprocesador.

A este conector vienen todas las señales que se conectan al microprocesador excepto las de datos. La señal IRQ2 es la unión de las señales de interrupciones internas de las UARTs del microprocesador [33]. Viene de la puerta OR y se activa a nivel alto.

SD de regulador se ha conectado a un pin de propósito general del microprocesador (GPIO29, pin 43 de X1 y pin 22 del esquemático de la placa) [34]. Para implementar el Chip Select mediante el decodificador se utilizan una de las señales CS del microprocesador y una de direcciones.

Para las señales de habilitación de lectura y escritura de la DualUART se han utilizado las señales EXT_OE y EXT_WE respectivamente y para las de dirección de memoria (A0-A2) se han usado las equivalentes del microprocesador (A0-A2).

Para las señales de control del MO200, se han utilizado pins de propósito general (GPIOs) [34]. Estas señales están multiplexadas con las de las UARTs del microprocesador. Dependiendo de si las señales son de entrada o de salida se han elegido las señales de la UART C o D respectivamente:

Tabla 5.10. Pins señales de control

	Señal	GPIO #	# pin
IN	MOD_ON_STATE	20	66
	MOD_UART_STATE	23	67
	MOD_FLOW_STATE	40	54
	MOD_RESET_STATE	22	63
	DTE_RESET_CMD	41	55
	DTE_ESCAPE_CMD	21	64
OUT	MOD_ON_CMD	44	58
	MOD_OFF_CMD	27	41
	DTE_UART_STATE	26	40
	MOD_RESET_CMD	25	38
	MOD_ESCAPE* CMD	24	37

En el conector X2, la numeración de los pins sigue con la del X1 por imposibilidad de nombrar dos pins igual en el mismo esquemático. X1 y X2 deben estar juntos para poder dimensionar correctamente la distancia entre ambos. Así, el pin que tendría que ser el 1 es el 123, etc.

Las señales utilizadas para los datos (D0-D7) de la Dual UART son las mismas en el microprocesador (D0-D7).

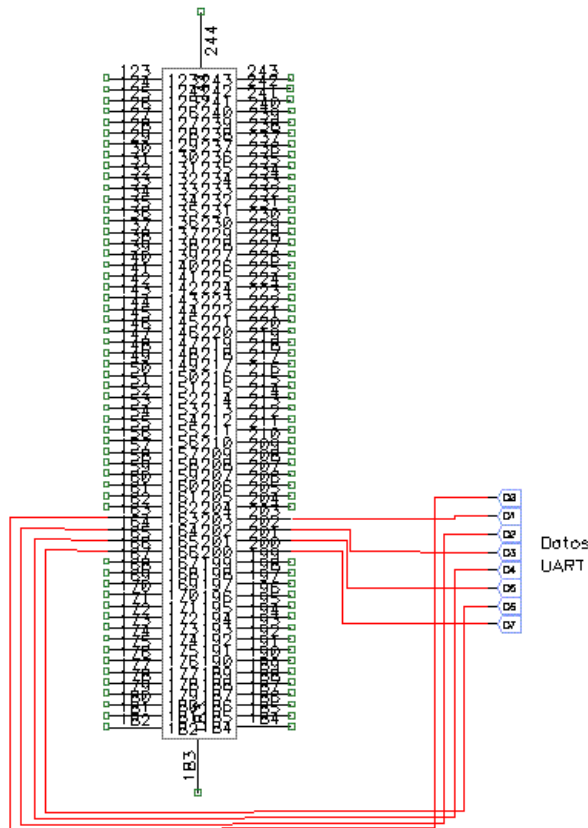


Fig. 5.29 Conector X2

5.3.10. Puerta OR

Este chip implementa 4 puertas OR. En este caso se ha utilizado la puerta 1 donde 1A y 1B son las entradas conectadas a su vez a las señales de interrupción de la UART y 1Y es la salida conectada al pin de IRQ2 del microprocesador. El dispositivo se alimenta a 3,3V. Las puertas no utilizadas se dejan sin conectar.

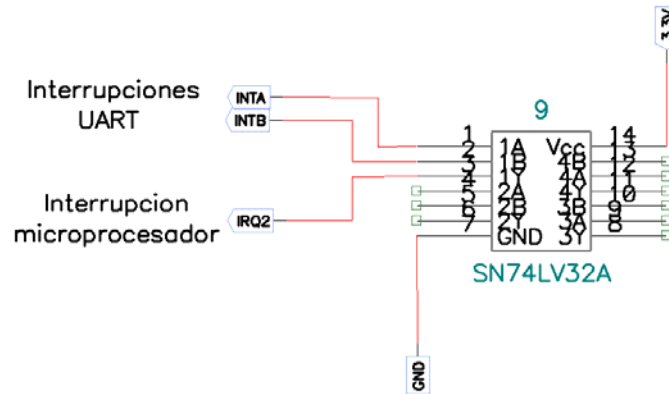


Fig. 5.30 Puerta OR

5.3.11. Decodificador 2-to-4

En las entradas A y B del chip se conectan EXT_CS0 que es uno de los Chip Selects del microprocesador y ADD3, que es la señal de dirección de memoria siguiente a las usadas en la UART. CSA y CSB se han conectado a dos de las salidas del decodificador, la elección de estas salidas se explica en el apartado 5.2.7. Las salidas no utilizadas se dejan sin conectar (Fig. 5.12 y Fig. 5.13). El dispositivo se alimenta a 3,3V.

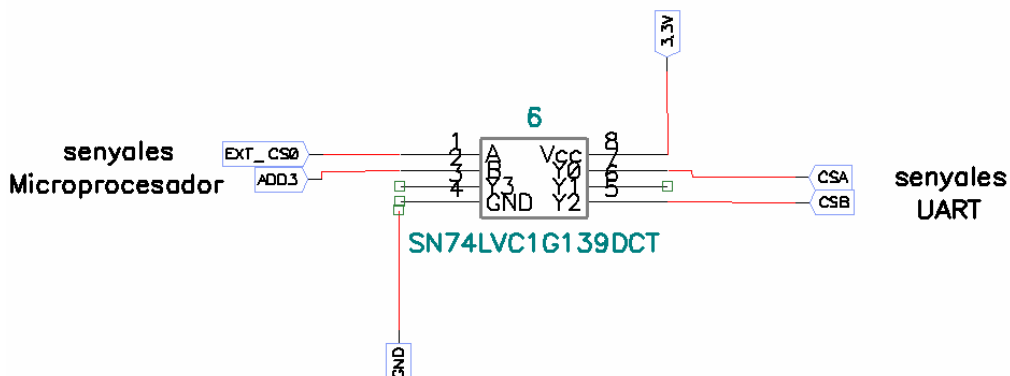


Fig. 5.31 Decodificador 2-to-4

CONCLUSIONES Y LÍNEAS FUTURAS

Conclusiones

A través del presente trabajo hemos sentado las bases para la futura implementación del sistema de comunicaciones del UAV desarrollado en el ámbito del proyecto de investigación ICARUS.

En este proyecto se realizó el diseño de la placa que permite la conexión entre el microprocesador y el módulo GPRS y se estudió la estructura del software necesaria para que el módulo se pueda comunicar con la red GPRS.

Hemos podido comprobar las ventajas de GPRS para la aplicación que queremos desarrollar; debido a que es una tecnología apropiada para la transmisión de datos desde el UAV ya que presenta una serie de facilidades para realizar la conexión a Internet, además de sus funcionalidades para el ahorro de batería de los terminales, lo que favorece al aspecto de la movilidad requerida por el UAV.

Líneas futuras

Respecto a la parte hardware, se plantea para un futuro el diseño del sistema que permita realizar la alimentación del microprocesador para que la placa diseñada pueda funcionar correctamente.

En lo que se refiere al software, se pretende implementar un programa en lenguaje C utilizando el sistema operativo Linux. Dicho programa debe controlar la comunicación entre el módulo y la red GPRS, además de permitir el intercambio de mensajes entre el microprocesador y el módulo a través de los comandos AT.

BIBLIOGRAFÍA

- [1] Roldán Martínez, D., “Telefonía móvil”, cap.3, pp. 50-96, *Comunicaciones inalámbricas*, Ra-Ma, Madrid (2004).
- [2] Arroyo Galán, L., “Segunda generación de telefonía celular”, cap. 3, pp. 50-68, *Tecnología móvil: Aplicaciones GSM, GPRS, UMTS y Wi-Fi*, Anaya, Madrid (2003).
- [3] Andersson, C., “The mobile evolution”, cap. 2, pp. 13-28, *GPRS and 3G wireless applications*, John Wiley & Sons, Inc., New York (2001).
- [4] Tanenbaum, A. S., “La capa de enlace de datos”, cap. 3, pp. 183-342, *Redes de computadoras*, Prentice Hall Inc. (Pearson Educación), México (2003).
- [5] Comer, D. E., “Estratificación de protocolos por capas”, cap. 11, pp. 161-191, *Redes globales de información con Internet y TCP/IP: Principios básicos, protocolos y arquitectura*, Prentice Hall Hispanoamericana S. A., México (1996).
- [6] Arroyo Galán, L., “La transición”, cap. 4, pp. 69-84, *Tecnología móvil: Aplicaciones GSM, GPRS, UMTS y Wi-Fi*, Anaya, Madrid (2003).
- [7] Andersson, C., “GPRS-Wireless packet data”, cap. 3, pp. 29-54, *GPRS and 3G wireless applications*, John Wiley & Sons, Inc., New York (2001).
- [8] Letellier, B., SAGEM S.A., *AT commands in basic GSM/GPRS applications*, 2004.
<http://www.alphamicro.net/resources/Getting%20started%20with%20AT%20commands%20-%20RevB.pdf>
- [9] Grupo de comunicaciones radio, Departamento TSC, UPC, *Tema 4. Sistema GSM/GPRS* [Apuntes Sistemas de comunicaciones móviles], 2005.
- [10] SAGEM S.A., *AT commands set for SAGEM Modules*, 2005.
<http://www.alphamicro.net/components/product~id~248~line~12.asp>
- [11] Network Working Group, Simpson, W., *RFC 1661: The Point-to-Point Protocol (PPP)*, 1994.
<http://www.faqs.org/ftp/rfc/pdf/rfc1661.txt.pdf>
- [12] Turtiainen, E., Arkko, J., *Linux GPRS HOWTO*, 2001.
<http://turtiainen.dna.fi/GPRS-HOWTO>
- [13] MOTOROLA, *Developer's guide: Motorola g20 AT commands*, 2004.

http://www.macro.sk/wwwmacro/newsadmin/upload2/g20_at_commands_12_july04.pdf

[14] Multi-Tech Systems Inc., *AT Commands for GSM/GPRS wireless modems*, 2004.

<http://www.multitech.com/DOCUMENTS/Collateral/manuals/S000293B.pdf>

[15] Lesrel, S., WAVECOM, *AT Commands for GPRS*, 2002.

http://www.picpoint.com/articles/00013/gprs_at_cmd.pdf

[16] ETSI GSM 07.07: Digital cellular telecommunications system (Phase 2+), *AT command set for GSM Mobile Equipment (ME)*, (GSM 07.07 version 7.0.0), 1998.

<http://www.3gpp.org>, <http://etsi.org>

[17] ETSI GSM 07.10: 3rd Generation Partnership Project, Technical Specification Group Terminals, *Terminal Equipment to Mobile Station (TE-MS) multiplexer protocol* (3G TS 27.010 version 2.0.0), 1999.

<http://www.3gpp.org>, <http://etsi.org>

[18] SAGEM S.A., *MO2XX module preliminary specification*, 2004.

<http://www.sagem.com>

[19] SAGEM S.A., *MO2XX module application note*, 2004.

<http://www.sagem.com>

[20] Analog Devices, *ADG3247 2-Port level translator*, 2003.

http://www.analog.com/en/prod/0,,768_835_ADG3247%2C00.html

[21] Analog Devices, *ADG3246 2-Port level translator*, 2003.

http://www.analog.com/en/prod/0,,768_835_ADG3246%2C00.html

[22] National Semiconductors, *LP3963/LP3966 3A fast ultra Low Dropout Linear Regulators*, 2005.

<http://www.national.com/pf/LP/LP3966.html>

[23] National Semiconductors, *Linear regulators: theory of Operation and Compensation*, 2002.

<http://www.national.com/an/AN/AN-1148.pdf>

[24] Texas Instruments, *TL16C752B dual UART*, 1999.

<http://focus.ti.com/docs/prod/folders/print/tl16c752b.html>

[25] Maxim, *MAX3222E-MAX3246E serial controllers*, 2005.

http://www.maxim-ic.com/quick_view2.cfm/gv_pk/1780

[26] AMP, *Plug assembly SH type*, 1993.

<http://www.rs-components.com.au/electronic-components-au/08214-08-pitch-amp-fh-series-08mm-pitch-amp-fh-series-smt-header-socke.html>

- [27] Texas Instruments, *sn54lv32a, sn74lv32a quadruple 2-Inputs positive-OR gates*, 2005.
<http://focus.ti.com/docs/prod/folders/print/sn74lv32a.html>
- [28] NetSilicon, *NET+ARM processors*, 2003-2004.
http://www.netsilicon.com/pdf/prd_nap_ns9750.pdf
- [29] NetSilicon, *9750b-a1 datasheet*, 2005.
<http://www.netsilicon.com>
- [30] Texas Instruments, *sn74lvc1g139 2-to-4 Line decoder*, 2004-2005.
<http://focus.ti.com/docs/prod/folders/print/sn74lvc1g139.html>
- [31] Alpha Micro, *Smart card reader with switch*, 2003.
<http://www.alphamicro.net/components/product~id~341~line~15.asp>
- [32] Kyocera, *Super micro connectors series 5604*.
<http://www.kyocera-elco.com/prdct/type/board/5604.html>
- [33] NetSilicon, *a9m9750dev_1a_V01 developers board standard*, 2005.
- [34] NetSilicon, *Pin description of A9M9750*, 2005.
- [35] PC/SC Workgroup, *Interoperability Specification for ICCs and Personal Computer Systems*, 2005
http://www.pcscworkgroup.com/specifications/files/pcsc2_v2.01.01.pdf
- [36] FS Forth-Systeme GmbH, *Module A9M9750_2 Users manual*, 2005



**Escola Politècnica Superior
de Castelldefels**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

ANEXOS

TÍTULO DEL TFC/PFC: Proyecto ICARUS. Sistema de comunicación GPRS

TITULACIÓN: Ingeniería Técnica de Telecomunicación, especialidad Sistemas de Telecomunicación

**AUTORES: Esunly Medina Medina
Natalia Revuelto Lorda**

DIRECTOR: Enric Pastor Llorens

FECHA: 20 de febrero de 2006

ANEXO 1: COMANDOS AT PARA GPRS

1.1 Comando AT+CGATT

AT+CGATT	
Descripción: GPRS attach o detach	
<u>Comando de prueba:</u>	<u>Respuesta:</u>
AT+CGATT=?	+CGATT: (lista de estados <state> soportados) OK
<u>Comando de lectura:</u>	<u>Respuesta:</u>
AT+CGATT?	+CGATT:<state> OK
<u>Comando de escritura:</u>	<u>Respuesta:</u>
AT+CGATT=<state>	+CGATT:<state> OK
	<u>Parámetros:</u>
	<state>: indica el estado del proceso de attach GPRS. 0:detached. 1:attached.

1.2 Comando AT+CGACT

AT+CGACT	
Descripción: Activar o desactivar contexto PDP	
<u>Comando de prueba:</u>	<u>Respuesta:</u>
AT+CGACT=?	+CGACT: (lista de estados (<state>) soportados) OK
<u>Comando de lectura:</u>	<u>Respuesta:</u>
AT+CGACT?	+CGACT: <cid>,<state> OK

<u>Comando de escritura:</u> AT+CGACT= <state> [,<cid>]	<u>Respuesta:</u> OK <u>Parámetros:</u> <state> : Indica el estado de activación del contexto PDP. 0 : desactivado. 1 : activado. Otros valores resultarán en una respuesta de "ERROR" ante el comando de ejecución. <cid> : Se conoce como identificador del contexto PDP y es un parámetro numérico que especifica una definición particular del contexto PDP. Es un parámetro local a la interfaz entre el TE y el MT y se usa en otros comandos relacionados con el contexto PDP.
---	---

1.3 Comando AT+CGCLASS

AT+CGCLASS	
Descripción: Clase de estación móvil GPRS	
<u>Comando de prueba:</u> AT+CGCLASS=?	<u>Respuesta:</u> +CGCLASS: (lista de clases <class> soportadas) OK
<u>Comando de lectura:</u> AT+CGCLASS?	<u>Respuesta:</u> +CGCLASS: <class> OK
<u>Comando de escritura:</u> AT+CGCLASS= <class>	<u>Respuesta:</u> OK <u>Parámetros:</u> <class> : Este parámetro es una cadena de caracteres que indica la clase de terminal móvil GPRS (en orden descendiente de funcionalidad). B : Clase B. CG : clase C sólo en modo GPRS. CC : clase C sólo en modo de conmutación de circuitos). Debemos tener en cuenta que nuestro módulo no soporta la clase A.

1.4 Comando AT+CGDCONT

AT+CGDCONT	
Descripción: Definir contexto PDP	
<u>Comando de prueba:</u> AT+CGDCONT=?	<u>Respuesta:</u> +CGDCONT: (rango de <cid> soportados), <PDP_type>,,,(lista de <d_comp> soportados), (lista de <h_com> soportados)[, (lista de <pd1> soportados) [...[, (lista de <pdN> soportados)]]][...]] [+CGDCONT: (rango de <cid> soportados), <PDP_type>,,,(lista de <d_comp> soportados), (lista de <h_com> soportados)[, (lista de <pd1> soportados) [...[, (lista de <pdN> soportados)]]][...]]] OK
<u>Comando de lectura:</u> AT+CGDCONT?	<u>Respuesta:</u> +CGDCONT: <cid>,<PDP_type>,<APN>,<PDP_addr>,<d_comp>,<h_comp>[,<pd1>[,...[,<pdN>]]] [+CGDCONT: <cid>,<PDP_type>,<APN>,<PDP_addr>,<data_comp>,<head_comp>[,<pd1>[,...[,<pdN>]]] OK
<u>Comando de escritura:</u> AT+CGDCONT=<cid>,<PDP_type>,<APN>,<PDP_addr>,<d_comp>,<h_comp>	<u>Respuesta:</u> OK <u>Parámetros:</u> <PDP_type>: Packet Data Protocol type. Es una cadena de caracteres que especifica el tipo de protocolo de paquete de datos. Debemos tener en cuenta que el módulo sólo soporta el protocolo IP. <APN>: Acces Point Name. Este parámetro es un nombre lógico (cadena de caracteres) que se usa para seleccionar al GGSN o la red de paquetes de datos externa. <PDP_addr>: Cadena de caracteres que identifica al MT en el espacio de direcciones aplicable al PDP. Como actualmente sólo se soporta IP, este parámetro debe ser una dirección IP. Si el valor es nulo ("0.0.0" o 0), entonces el TE puede proveer uno durante el procedimiento de arranque del PDP o si esto falla, se puede solicitar una dirección IP dinámica. El comando de lectura continuará devolviendo el valor nulo, aun cuando se haya

	<p>asignado una dirección durante el proceso de inicio del PDP. La dirección asignada se puede leer usando el comando +CGPADDR.</p> <p><d_comp>: Parámetro numérico que controla la compresión de datos PDP (0: off, viene por defecto y es el único valor soportado).</p> <p><h_comp>: Parámetro numérico que controla la compresión de cabecera PDP (0: off, viene por defecto y es el único valor soportado).</p> <p><pd1>,...<pdN>: Cadena de 0 a N caracteres cuyos significados son específicos del <PDP_type></p>
--	---

1.5 Comando AT+CGDATA

AT+CGDATA	
Descripción: Entrar en estado de datos	
<u>Comando de prueba:</u>	<u>Respuesta:</u>
AT+CGDATA=?	+CGDATA: (lista de <L2P> soportados) OK
<u>Comando de escritura:</u>	<u>Respuesta:</u> CONNECT
AT+CGDATA= <L2P>,<cid>	<u>Parámetros:</u> <L2P> : Cadena de caracteres que indica el protocolo de capa 2 que va a ser usado entre el TE y el MT. Actualmente sólo está permitido el protocolo PPP. <cid> : Se conoce como identificador del contexto PDP y es un parámetro numérico que especifica una definición particular del contexto PDP. Es un parámetro local a la interfaz entre el TE y el MT y se usa en otros comandos relacionados con el contexto PDP.

1.6 Comando AT+CGEREP

AT+CGEREP	
Descripción: Informe de evento GPRS	
<u>Comando de prueba:</u> AT+CGEREP=?	<u>Respuesta:</u> +CGEREP: (lista de <mode> soportados), (lista de <bfr>) OK
<u>Comando de lectura:</u> AT+CGEREP?	<u>Respuesta:</u> +CGEREP: <mode>,<bfr> OK
<u>Comando de escritura:</u> AT+CGEREP = [<mode>,<bfr>]]	<u>Respuesta:</u> OK <u>Parámetros:</u> <mode> : Puede tomar los valores 0, 1 ó 2. 0 : almacenar en memoria los códigos de resultado no solicitados en el MT. Si la memoria está llena, los más antiguos pueden ser descartados. Los códigos no se reenvían al TE. 1 : descartar códigos de resultados no solicitados cuando el enlace entre MT y TE está reservado; por ejemplo en modo de conexión de datos, si no reenviarlos directamente al TE. 2 : almacenar en memoria los códigos de resultado no solicitados en el MT cuando el enlace entre MT y TE está reservado y enviarlos al TE cuando el enlace vuelva estar disponible. <bfr> : 0 : la memoria de códigos de resultados no solicitados del MT, definida en este comando, se borra cuando se introducen los valores 1 ó 2 para el parámetro <mode>. 1 : los códigos de resultados no solicitados que se encuentran en la memoria del MT son enviados hacia el TE cuando se establece el parámetro <mode> con los valores 1 ó 2. Se deben enviar respuestas de OK después de enviar los códigos.

1.7 Comando AT+CGPADDR

AT+CGPADDR	
Descripción: Definir contexto PDP	
<u>Comando de prueba:</u> AT+CGPADDR=?	<u>Respuesta:</u> +CGPADDR: (lista de <cid> soportados) OK
<u>Comando de escritura:</u> AT+CGPADDR = <cid>[,<cid>,[...]]	<u>Respuesta:</u> +CGPADDR: <cid>,<PDP_addr> [+ CGPADDR: <cid>,<PDP_addr>[...]] OK <u>Parámetros:</u> <PDP_addr>: Cadena de caracteres que identifica al MT en el espacio de direcciones aplicable al PDP. La dirección puede ser estática o dinámica. La dirección estática será la establecida por el comando +CGDCONT cuando fue definido el contexto. La dirección dinámica será la asignada durante la última activación del contexto PDP que utilizó la definición de contexto referenciada por el parámetro <cid>. < PDP_addr > se omite si no hay ninguna dirección disponible.

1.8 Comando AT+CGQMIN

AT+CGQMIN	
Descripción: Perfil de calidad de servicio mínimo aceptable	
<u>Comando de prueba:</u> AT+CGQMIN =?	<u>Respuesta:</u> +CGQMIN: <PDP_type>, (lista de <precedence> soportados), (lista de <delay> soportados),(lista de <reliability> soportados),(lista de <peak> soportados),(lista de <mean> soportados) [+ CGQMIN:...] OK
<u>Comando de lectura:</u> AT+CGQMIN?	<u>Respuesta:</u> +CGQMIN: <cid>,<precedence>,<delay>,<reliabil

	ity>,<peak>,<mean> [+ CGQMIN:...] OK
<u>Comando de escritura:</u> AT+CGQMIN= [<cid>,<precedence>,< delay>,<reliability>,<pe ak>,<mean>]]]]]	<u>Respuesta:</u> OK <u>Parámetros:</u> <precedence> : Parámetro numérico para la clase de precedencia. 0 : valor suscrito a la red. 1 : servicio de alta prioridad. 2 : servicio de prioridad normal. 3 : baja prioridad. <delay> : Parámetro numérico para la clase de retardo. <reliability> : Parámetro numérico para la clase de fiabilidad 0 : valor suscrito a la red. 1 : tráfico no a tiempo real, aplicación sensible a errores que no puede hacer frente a la pérdida de datos. 2 : tráfico no a tiempo real, aplicación sensible a errores que puede hacer frente a pérdidas infrecuentes de datos 3 : tráfico no a tiempo real, aplicación sensible a errores que puede hacer frente a pérdidas de datos. 4 : tráfico a tiempo real, aplicación sensible a errores que puede hacer frente a pérdidas de datos. 5 : tráfico a tiempo real, aplicación no sensible a errores que puede hacer frente a pérdidas de datos). <peak> : Parámetro numérico para la clase de throughput máximo. 0 : valor suscrito a la red. 1 : 8 Kbps. ... 9 : 2048 Kbps. <mean> Parámetro numérico para la clase de throughput medio 0 : valor suscrito a la red. 1 : 0.22 bps aproximadamente. ... 18 : 111 Kbps aproximadamente. 31 : best effort.

1.9 Comando AT+CGQREQ

AT+CGQREQ	
Descripción: Petición de perfil de calidad de servicio	
<u>Comando de prueba:</u> AT+CGQREQ =?	<u>Respuesta:</u> +CGQREQ: <PDP_type> , (lista de <precedence> soportados), (lista de <delay> soportados), (lista de <reliability> soportados), (lista de <peak> soportados), (lista de <mean> soportados) [+CGQREQ:...] OK
<u>Comando de lectura:</u> AT+CGQREQ?	<u>Respuesta:</u> +CGQREQ:<cid>,<precedence>,<delay>,<reliability>,<peak>,<mean> [+CGQREQ:<cid>,<precedence>,<delay>,<reliability>,<peak>,<mean> [...]] OK
<u>Comando de escritura:</u> AT+CGQREQ = [<cid>,<precedence>,<delay>,<reliability>,<peak>,<mean>]]]]]]	<u>Respuesta:</u> OK <u>Parámetros:</u> <cid>,<precedence>,<delay>,<reliability>,<peak>,<mean>

1.10 Comando AT+CGREG

AT+CGREG	
Descripción: Estado de registro de red GPRS	
<u>Comando de prueba:</u> AT+CGREG =?	<u>Respuesta:</u> +CGREG: (lista de <n> soportados) OK
<u>Comando de lectura:</u> AT+CGREG?	<u>Respuesta:</u> +CGREG: <n>,<stat>[<lac>,<ci>] OK
<u>Comando de escritura:</u> AT+CGREG = [<n>]	<u>Respuesta:</u> OK <u>Parámetros:</u> <n>: <ul style="list-style-type: none"> 0: deshabilitar código de resultado de registro de red no solicitado. 1: habilitar el código de resultado de registro de red no solicitado +CGREG:<stat>. 2: habilitar el códigos de resultado de registro de red y de información de localización no solicitado +CGREG:<stat>[<lac>,<ci>]). <stat>: <ul style="list-style-type: none"> 0: no registrado, el equipo móvil no está actualmente buscando un operador para registrarse. El servicio GPRS está deshabilitado, se le permite a la MS realizar el proceso de attach si es pedido por parte del usuario. 1: registrado, red local. 2: no registrado, pero el equipo móvil está actualmente intentando realizar el attach o buscando un operador para registrarse. 3: registro denegado. 4: desconocido. 5: registrado. <lac>: parámetro tipo cadena de caracteres. Código hexadecimal de dos bytes que indica el área de localización del MT. <ci>: Cadena de caracteres que indica la identificación de la celda. Dos bytes en formato hexadecimal.

1.11 Comando AT+KMCLASS

AT+ KMCLASS	
Descripción: Cambio de clase multislots GPRS	
<u>Comando de prueba:</u> AT+KMCLASS =?	<u>Respuesta:</u> +KMCLASS: (lista de <mclass> soportadas) OK
<u>Comando de lectura:</u> AT+KMCLASS?	<u>Respuesta:</u> +KMCLASS: <mclass> OK
<u>Comando de escritura:</u> AT+KMCLASS = <mclass>	<u>Respuesta:</u> OK <u>Parámetros:</u> <mclass>: 1: 1+1 2: 2+1 4: 3+1 8: 4+1 10: 4+2

ANEXO 2: CÓDIGOS DE RESULTADO Y MENSAJES NO SOLICITADOS

Formato Verbal	Numérico	Tipo	Descripción
+CCCM:<ccm>	Como verbal	No solicitado	--
+CMTI	Como verbal	No solicitado	--
CONNECT	1	Intermedio	La conexión ha sido establecida
CONNECT<text>	Específico del fabricante	Intermedio	Igual a CONNECT pero el <text> específico del fabricante da información adicional (ejemplo: velocidad de conexión)
ERROR	4	Final	Comando no aceptado
NO ANSWER	7	Final	Expira temporizador de conexión
NO CARRIER	3	Final	Conexión terminada
NO DIALTONE	5	Final	No se detecta tono de llamada
OK	0	Final	Ejecución de reconocimiento de una línea de comando
RING	2	No solicitado	Señal de llamada de red entrante
+CRING:<type>	Como verbal	No solicitado	--
+CME ERROR:<err>	Como verbal	Final	--
+CBM	Como verbal	No solicitado	--
+CDS	Como verbal	No solicitado	--
+CR: <type>	Como verbal	Intermedio	--
BUSY	6	Final	--

ANEXO 3: CÓDIGOS ERROR DE GPRS

Código de <err>	Significado
Errores relacionados con un fallo para realizar un Attach	
103	MS ilegal (#3)
106	ME ilegal (#6)
107	Servicios GPRS no permitidos (#7)
111	PLMN no permitida (#11)
112	Área de localización no permitida (#12)
113	Itinerancia no permitida en esta área de localización (#13)
Errores relacionados con un fallo para activar un Contexto	
132	Opción de servicio no soportada (#32)
133	Opción de servicio solicitud no suscrita (#33)
134	Opción de servicio temporalmente fuera de servicio (#34)
Otros errores GPRS	
149	Fallo de autenticación PDP
148	Error GPRS no especificado
150	Clase móvil inválida

ANEXO 4: EJEMPLOS PRÁCTICOS DEL USO DE COMANDOS AT

4.1 Activación de un contexto IP/PDP

```
AT +CGDCONT=1, "IP", "internet"; +GDCONT=2, "IP", "abc.com"  
OK
```

```
ATD*99***1#  
CONNECT
```

Otro ejemplo:

```
AT +CGCLASS="CG"  
OK
```

```
+CGREG: 1
```

```
AT +CGDCONT=1, "IP", "internet"  
OK
```

```
AT +CGQREQ=1,1,4,5,2,14  
OK  
AT +CGQMIN=1,1,4,5,2,14  
OK
```

```
AT +CGATT=1  
OK
```

```
AT +CGACT=1,1  
OK
```

//Nota sobre +CGDATA: El objetivo de este comando es el mismo que ATD*99*** así que, es mejor usar solamente //ATD*99***

```
AT +CGDATA=1  
CONNECT
```

.....

Nota: Estos ejemplos han sido tomados del documento referenciado en [15].

4.2 Ejemplo de conexión a Internet en modo GPRS

```
at+cgatt?          /* comprueba si el módulo está en la fase de attach GPRS*/  
+CGATT:0
```

OK

```
at+cgatt=1        /* empieza el procedimiento de attach GPRS*/  
OK
```

```
at+cgatt?          /* comprueba si el módulo está en la fase de attach GPRS*/  
+CGATT:1
```

OK

```
at+cgdcont=1,"IP","APN",0,0,0      /* define el contexto PDP*/  
OK
```

/ at+cgreq et at+cgqmin si la QoS necesita ser detallada*/*

```
atd*99***1#        /* inicia la conexión GPRS (abre el contexto PDP)*/
```

CONNECT

```
~}#À!;!} } }2}!}$}%Ü"}&} }*} } }#}$À#kZ~ +++ /* desconectar al DTR para colgar*/  
OK
```

```
ath                /* termina la conexión GPRS (cierra el contexto PDP)*/  
OK
```

Recordar que +++ es el comando para escapar del modo datos y volver al modo comando.

Nota: Este ejemplo ha sido tomado del documento referenciado en [8].

4.3 Ejemplos de comandos GPRS

4.3.1 +CGCLASS: Clase de estación móvil GPRS

```
AT+CGCLASS=?  
+CGCLASS: (B)  
OK
```

4.3.2 +CGDCONT: Definir contexto PDP

```
AT+CGDCONT=?  
+CGDCONT: (1-3),("IP"),,,(0,1),(0,1)  
OK
```

```
AT+CGDCONT?  
+CGDCONT: 1,"IP",,"", "0.0.0.0",0,0  
+CGDCONT: 2,"IP",,"", "0.0.0.0",0,0  
+CGDCONT: 3,"IP",,"", "0.0.0.0",0,0  
OK
```

```
at+cgdcont= 1,"IP","internet","0.0.0.0",0,0  
OK
```

```
at+cgdcont?  
+CGDCONT: 1,"IP","internet","0.0.0.0",0,0  
+CGDCONT: 2,"IP",,"", "0.0.0.0",0,0  
+CGDCONT: 3,"IP",,"", "0.0.0.0",0,0  
OK
```

```
at+cgdcont= 1,"IP","internet","0.0.0.0",0,0  
OK
```

```
at+cgdcont=2,"IP","internet","0.0.0.0",1,1  
OK
```

4.3.3 +CGQMIN: Perfil de QoS mínimo aceptable

```
AT+CGQMIN=?  
+CGQMIN: (1-3),(0-3),(0-4),(0-5),(0-9),(0-18,31)  
OK
```

```
AT+CGQMIN?
```

```
+CGQMIN: 1,2,4,3,9,10
+CGQMIN: 2,2,4,3,9,10
+CGQMIN: 3,2,4,3,9,10
OK
```

4.3.4 +CGQREQ: *Petición de perfil de QoS*

```
AT+CGQREQ=?
+CGQREQ: (1-3),(0-3),(0-4),(0-5),(0-9),(0-18,31)
OK
```

```
AT+CGQREQ?
+CGQREQ: 1,2,4,3,9,10
+CGQREQ: 2,2,4,3,9,10
+CGQREQ: 3,2,4,3,9,10
OK
```

```
AT+CGQREQ=1,0,,0,0,0
OK
```

```
AT+CGQREQ?
+CGQREQ: 1,0,4,0,0,0
+CGQREQ: 2,2,4,3,9,10
+CGQREQ: 3,2,4,3,9,10
OK
```

4.3.5 +CGATT: *Attach o Detach GPRS*

```
AT+CGATT=?
+CGATT: (0,1)
OK
```

```
AT+CGATT?
+CGATT: 0
OK
```

```
AT+CGATT=0
OK
```

Nota: Estos ejemplos han sido tomados del documento referenciado en [13].

ANEXO 5: PROCEDIMIENTOS DE ATTACH Y ACTIVACIÓN DE CONTEXTO PDP

En nuestro caso, el terminal es clase B; por lo que necesita decirle a la red que él puede recibir tanto conexiones GSM como GPRS. Este procedimiento se conoce con el nombre de **Attach**.

Al realizarse un attach GPRS se crea un enlace lógico entre el SGSN y el MS. Este proceso, que se activa cuando se enciende el terminal, sigue los siguientes pasos:

1. El MS envía un mensaje de petición de Attach al SGSN.
2. El SGSN comprueba si reconoce al MS y trata de encontrar su número de identificación único (IMSI). Si no se reconoce al MS, este pide al SGSN antiguo el IMSI y los tripletes de autenticación.
3. Si el SGSN antiguo no conoce al MS, envía un mensaje de error. El nuevo SGSN pregunta al MS por su IMSI.
4. EL SGSN realiza una autenticación del MS.
5. Si el MS es encontrado en una nueva área de servicio, se actualiza el HLR.
6. Si el MS está actualmente en una nueva área de localización, se actualiza el MSC/VLR.
7. El SGSN le dice al MS su TLLI asignado (Temporary Location Link Identifier). El TLLI se usa a través de la sesión GPRS como un identificador para el enlace lógico MS-SGSN.

El proceso inverso al attach es el **detach**, mediante el cual se desconecta al terminal de la red GPRS y se produce normalmente al apagar el equipo.

Una vez que está establecido el enlace MS-SGSN, el móvil necesita obtener una dirección IP y otros parámetros de conexión. Esto se hace a través de la activación del **contexto PDP** (Packet Data Protocol).

El contexto PDP puede ser visto como un software grabado que contiene parámetros que son relevantes para una determinada conexión. Incluye información sobre los protocolos que se usan, la dirección IP, el perfil de QoS, etc. Los parámetros del contexto PDP deseados pueden ser establecidos por la aplicación usando comandos AT. La activación del contexto PDP hace que el móvil sea visible para la correspondiente GGSN, la cual hace posible las conexiones externas.

A continuación se muestran los pasos para el procedimiento de activación del contexto PDP (si el GPRS Attach ha sido realizado con anterioridad):

1. El MS envía una petición de contexto PDP al SGSN.
2. Se pueden ejecutar funciones de seguridad entre el MS y el SGSN, el cual valida la petición.
3. El SGSN:
 - * Comprueba la suscripción.
 - * Comprueba la QoS, que afecta a la tarificación del servicio.
 - * Envía información al GGSN sobre cómo localizar al MS.
 - * Configura un enlace lógico hacia el GGSN mediante la creación de un "tunel".
4. El GGSN contacta un servidor RADIUS (Remote Access Dial-In User Service) dentro de la red operadora y obtiene una dirección IP para el MS.
5. La dirección IP es enviada de vuelta hacia el MS.

Ahora el usuario GPRS está listo para enviar y recibir paquetes.

Nota: Esta información ha sido tomada del documento referenciado en [7].