Capítulo 4. PROGRAMACIÓN Y PROCESADO DE DATOS

1. INTRODUCCCIÓN

1.1 Antecedentes

Cuando existen sistemas que se encuentran localizados a larga distancia, es necesario un método que permita obtener la información de las variables de manera rápida para poder llevar a cabo una acción, que en determinados casos puede ser de vital importancia para la entidad.

En el caso convencional en el que una persona tenga que realizar las acciones de obtención de variables en lugares ubicados remotamente, se puede encontrar con varias dificultades como son: la ubicación topográfica del lugar, el costo que implica obtener estas variables, es por ello que la telemetría es una solución a estas dificultades, tomando gran importancia el correcto funcionamiento del sistema.

A través del control y de la medición a distancia se minimiza el tiempo en el que se recogen los datos o variables y se puede realizar con mayor rapidez los ajustes respectivos tomando en cuenta las condiciones del caso, también se minimizan los accidentes de trabajos y se obtiene un ahorro significativo de dinero al emplear menor cantidad de personal y/o transporte.

1.2. Telemetría

La telemetría consiste en la adquisición de datos de cualquier índole a distancia mediante sensores o transductores ya sean éstos analógicos o digitales y enviarlos a una estación de control a través de un sistema de telecomunicaciones donde estos datos son administrados, procesados y visualizados.

Esta comunicación se realiza por medio de módems GSM que acondicionan las señales de información de acuerdo al medio en el que se realiza la comunicación.

1.3. Aplicaciones de Telemetría

Las aplicaciones de telemetría pueden relacionarse a varios tipos de sistemas con fines científicos como por ejemplo la industria aeroespacial, en el control de naves no tripuladas (aviones de reconocimiento), robots submarinos en la investigación submarina a donde el ser humano no puede ir, también en las agencias espaciales como la Nasa para operar naves espaciales y satélites, en la perforación de pozos petrolíferos donde se obtienen datos telemáticos a través del lodo de perforación.

Las aplicaciones también pueden tener fines no científicos como por ejemplo en la Fórmula 1 o moto GP donde se deben leer datos sobre el estado de los vehículos y otra aplicación muy importante se da en las empresas que brindan servicios básicos como luz, agua y teléfono.

La eficacia de la telemetría depende en gran medida del sistema de comunicaciones que se utiliza, siendo este el responsable de la velocidad con que lleguen los datos de información enviados desde las estaciones recolectoras de datos.

2. PROGRAMACIÓN DE ARDUINO

2.1 Introducción

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares. Se creó para diseñadores, artistas, aficionados y cualquier interesado en crear entornos y objetos interactivos.

El microcontrolador de la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring), con el propio entorno de desarrollo Arduino (basado en Processing) y un cargador de arranque (boot loader) que corre en la placa. Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectarse a un ordenador, también tienen la posibilidad de hacerlo y comunicarse con diferentes tipos de software (por ejemplo Flash, Processing, MaxMSP).

Al ser open-hardware, tanto su diseño como su distribución son libres. Es decir, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia. El entorno de desarrollo integrado libre se puede descargar gratuitamente de la página oficial de Arduino.

La plataforma Arduino se programa mediante el uso de su propio lenguaje, sin embargo, es posible utilizar otros lenguajes de programación y aplicaciones populares en Arduino, por ejemplo Java, C#, C++, Python, Visual Basic .NET, VBScript.

Esto es posible debido a que Arduino se comunica mediante la transmisión de datos en formato serie que es algo que la mayoría de los lenguajes anteriormente citados soportan. Para los que no soportan el formato serie de forma nativa, es posible utilizar software intermediario que traduzca los mensajes enviados por ambas partes para permitir una comunicación fluida. Es bastante interesante tener la posibilidad de interactuar con Arduino mediante esta gran variedad de sistemas y lenguajes puestos que dependiendo de cuales sean las necesidades del problema que vamos a resolver podremos aprovecharnos de la gran compatibilidad de comunicación que ofrece.

Arduino es una herramienta a tener en cuenta por su versatilidad y bajo coste en uso industrial. Es muy útil en aquellas situaciones en las que se necesita controlar un sistema o producto del que se van a fabricar un pequeño número de unidades. En esta situación el ingeniero no necesita emplearse en el diseño electrónico de la tarjeta de control del microcontrolador a utilizar, pues ya viene diseñada y lista para cargar tu programa.

Su uso se puede extender a la industria para el control de procesos pudiendo adaptar los valores de las entradas y salidas con el uso de etapas de optoacopladoras. A la hora de usar un sensor nos permite su linealización interna dando lugar a la optimización de la recogida de datos y la consecuente reducción de coste en transductores electrónicos.

2.2 Entorno de desarrollo

Para programar la placa es necesario descargar de la página web oficial de Arduino el entorno de desarrollo (IDE). Se dispone de versiones para Windows y MAC, así como las fuentes para compilarlas en Linux. La placa Arduino UNO R3 sólo se puede programar usando la versión 0021 y posteriores del software. En este proyecto se ha utilizado la versión Arduino 1.0.1.

El entorno de desarrollo de Arduino está constituido por un editor de texto para escribir el código, un área de mensajes, una consola de texto, una barra de herramientas con botones para las funciones comunes y una serie de menús. Permite la conexión con el hardware de Arduino para cargar los programas y comunicarse con ellos.

En la figura 36 se puede ver la simplicidad de la interfaz del compilador usado por Arduino, la cual no cambia sea cual sea la versión instalada. En el caso de disponer de una placa USB es necesario instalar los drivers FTDI. Estos drivers vienen incluidos en el paquete de Arduino mencionado anteriormente.

Figura 36. Entorno de desarrollo.

Arduino utiliza para escribir el software lo que denomina "sketch" (*programa*). Estos programas son escritos en el editor de texto, con el editor existe la posibilidad de copiar, cortar, pegar o remplazar el texto. En el área de mensajes se muestra información mientras se cargan los programas y también muestra los errores producidos durante la compilación. La consola presenta el texto de salida para el entorno de Arduino incluyendo los mensajes de error completos y otras informaciones. La barra de herramientas permite la verificación del proceso de carga, creación, apertura y guardado de programas, y la monitorización serie. A continuación se describen los botones de la barra de herramientas.

Verificar

Chequea el código en busca de errores.

Stop

Finaliza la monitorización serie y oculta otros botones.

Nuevo

Crea un nuevo sketch (programa).

 Abrir

Presenta un menú de todos los programas de su “sketchbook”, (librería de sketch). Un clic sobre uno de ellos lo abrirá en la ventana actual.

 Guardar

Salva el programa.

 Cargar

Compila el código y lo vuelca en la placa E/S de Arduino.

 Monitor Serial

Inicia la monitorización serie.

Se pueden encontrar también diferentes comandos en los demás menús disponibles del entorno de desarrollo: Archivo, Editar, Sketch, Herramientas y Ayuda. Los menús son sensibles al contexto en el que nos encontremos, lo que significa que sólo estarán disponibles los elementos relevantes para la tarea que se esté realizando en ese momento. Se van a describir los distintos menús que aparecen y algunos de los comandos más importantes.

-Editar

+ *Copiar para el Foro*
  Copia el código de su *sketch* en el portapapeles con el formato adecuado para publicarlo en un foro, incluyendo la sintaxis coloreada.
+ *Copiar como HTML*
  Copia el código de un programa (*sketch*) al portapapeles en formato HTML, adecuándolo para incrustarlo en una página web.

-Sketch

+ *Verificar*
  Verifica los errores de su programa (*sketch*).
+ *Importar Librería*
  Añade una librería a su programa (*sketch*) insertando la sentencia #include en el código.
+ *Mostrar la Carpeta de Sketch*
  Abre la carpeta de programas (*sketch*) en el escritorio.

+ *Agregar Archivo...*
  Añade un fichero fuente al programa (se incluirá desde su ubicación actual). El fichero aparece en una nueva pestaña en la ventana del programa. Los ficheros pueden ser quitados del programa (*sketch*) utilizando el menú "tab".

-Herramientas

+ *Formato Automático*
  Da formato al código proporcionando estética, por ejemplo realiza tabulaciones entre la apertura y cierre de llaves, y las sentencias que tengan que ser tabuladas lo estarán.
+ *Tarjeta*
  Selecciona la placa que estás usando.
+ *Puerto Serial*
  Este menú contiene todos los dispositivos series (reales o virtuales) de su equipo. Se refrescará automáticamente cada vez que abras el menú herramientas.
+ *Grabar Secuencia de Inicio*
  Este elemento del menú le permite grabar un gestor de arranque (*bootloader*) dentro del microcontrolador de la placa Arduino. Aunque no es un requisito para el normal funcionamiento de la placa, le será útil si compra un nuevo ATmega (el cual viene normalmente sin gestor de arranque). Asegúrese que ha seleccionado la placa correcta en el menú antes de grabar el *bootloader*.

-Librería de Sketch (Sketchbook)

El entorno de Arduino incluye el concepto de "sketchbook", que es el lugar estándar para el almacenamiento de sus programas. Los programas dentro de su "sketchbook" pueden abrirse desde el menú Archivo > Librería o desde el botón de la barra de herramientas Abrir. La primera vez que arranque el software Arduino, se creará un directorio para su "sketchbook".

-Pestañas, Ficheros múltiples y compilación

Permite manejar programas con más de un fichero (cada uno de los cuales aparece en su pestaña). Pueden ser normalmente ficheros de código Arduino (no extensiones), ficheros C (extensiones .c), ficheros C++ (.cpp), o ficheros de cabecera (.h).

-Volcado (Uploading)

Antes de volcar un programa, necesitará seleccionar los elementos correspondientes desde el menú Herramientas > Tarjeta y Herramientas > Puerto Serial.

En los Mac, el puerto serie será probablemente algo como /dev/tty.usbserial-1B1 (para una placa USB), o /dev/tty.USA19QW1b1P1.1 (para una placa serie conectada con un adaptador Keyspan USB-to-Serial).

En Windows, probablemente sea COM1 o COM2 (para una placa serie) o COM4, COM5, COM6, o superior (para una placa USB). Para encontrarlos, debemos buscar los dispositivos serie USB en la sección de puertos del Administrador de Dispositivos de Windows en el panel de control.

En Linux, debería ser /dev/ttyUSB0, /dev/ttyUSB1 o similar.

Una vez seleccionado el puerto serie y la placa, presione el botón de volcado en la barra de herramientas o seleccione Cargar desde el menú Archivo.

Las actuales placas de Arduino se resetean automáticamente y a partir de ahí comienza el volcado. Como las placas antiguas carecen de auto-reset, necesitará presionar el botón de reset en la placa, justo antes de iniciar el volcado. En muchas placas se observará el led RX y TX parpadeando cuando el programa está actualizándose. El entorno de Arduino mostrará un mensaje cuando el volcado esté completado, o mostrará un error en caso de fallo.

Cuando se vuelca un programa, se está utilizando el "bootloader" de Arduino, un pequeño programa que ha sido cargado en el microcontrolador de su placa. Permite el volcado del código sin utilizar hardware adicional. El "bootloader" está activo durante unos segundos cuando la placa es reseteada, después se inicia el programa que más recientemente se hubiera actualizado en el microcontrolador. El "bootloader" produce un parpadeo en el LED de la placa (pin 13) cuando se inicia (por ejemplo cuando las placas son reseteadas).

-Monitor Serial (Serial Monitor)

Muestra los datos enviados desde la placa Arduino (USB o serie). Para enviar datos a la placa, teclee el texto y pulsa el botón "Enviar" o “Enter”. Seleccione la velocidad (baud rate) en el menú desplegable que coincida con el configurado en la función Serial.begin () dentro de su programa.

Advertir que en Mac o Linux, la placa Arduino se resetea cuando conecta con el monitor serie.

-Preferencias (Preferences)

Pueden configurarse otras preferencias en el apartado preferencias (lo podrás encontrar bajo el menú Arduino para los Mac, o en Archivo para Windows y Linux). El resto de opciones puede ser localizado en el fichero de preferencias, que se podrá encontrar dentro del mismo apartado.

2.3 Estructura básica de un programa

La estructura básica de programación en Arduino es bastante simple y divide la ejecución en tres partes, declaración de variables, función de inicialización (setup) y función de Loop (programa), tal y como se puede apreciar en la figura 37.

a) *Declaración de variables*

En esta parte como su nombre bien indica sirve para declarar e inicializar las variables que serán utilizadas en el programa, en este espacio también se encuentran las librerías estándar que utilizará Arduino.

b) *setup ()*

La función setup() constituye la preparación del programa y se trata de la primera función que se ejecuta en el programa. Se usa para inicializar las variables, los modos de contactos, comenzar a usar las bibliotecas, etc. La función de configuración sólo se ejecutará una vez, después de cada momento del encendido o reinicio de la placa Arduino.

c) *loop ()*

El Arduino o cualquier tipo de microcontrolador estará la mayoría de tiempo corriendo en un loop (bucle) de programación. Lo que se encuentra en esta función es el programa que controlará la respuesta de la placa, y esto lo hará infinitas veces hasta que se desconecte de la alimentación.

Figura 37. Estructura de un programa Arduino.

3. PROGRAMACIÓN Y PROCESADO DE DATOS

3.1 Programación del nodo sensor

La figura 38 muestra el diagrama de flujo del programa implementado en el microcontrolador del nodo sensor. Como se ha comentado anteriormente, el código se ha realizado en el lenguaje de programación Arduino y se ha utilizado el entorno de desarrollo para alojar el programa en la placa.

En el bloque de declaración de variables se realizan las siguientes tareas:

- Declaración de variables y constantes.
- Inclusión de las librerías utilizadas.

El bloque de inicialización y configuración, bloque setup(), se ejecuta cada vez que se enciende o reinicia el dispositivo. Se encarga de las configuraciones iniciales del microcontrolador. Las tareas son las siguientes:

- Inicialización del puerto serie (UART).
- Configuración del puerto serie.

- Inicialización del reloj RTC.
- Inicialización del lector de tarjeta SD.
- Configuración de los pines digitales y analógicos.
- Configuración de la tarjeta SIM.

En caso de realizar una medida de CO_2, se realizan las siguientes acciones:

- Configuración del pin analógico A0 para recibir la señal procedente del sensor de CO_2.
- Configuración del pin digital 5 como salida, para alimentar el sensor de CO_2.
- Configuración del pin digital 0, como salida para alimentar el circuito de acondicionamiento de señal.
- Espera del tiempo de calentamiento programado para el TGS4161.
- Realización de la conversión analógica digital y guardar el valor.
- Configuración del pin 0 y 5 en nivel bajo para ahorrar energía.
- Envío del SMS o guardar el valor en el fichero de datos.

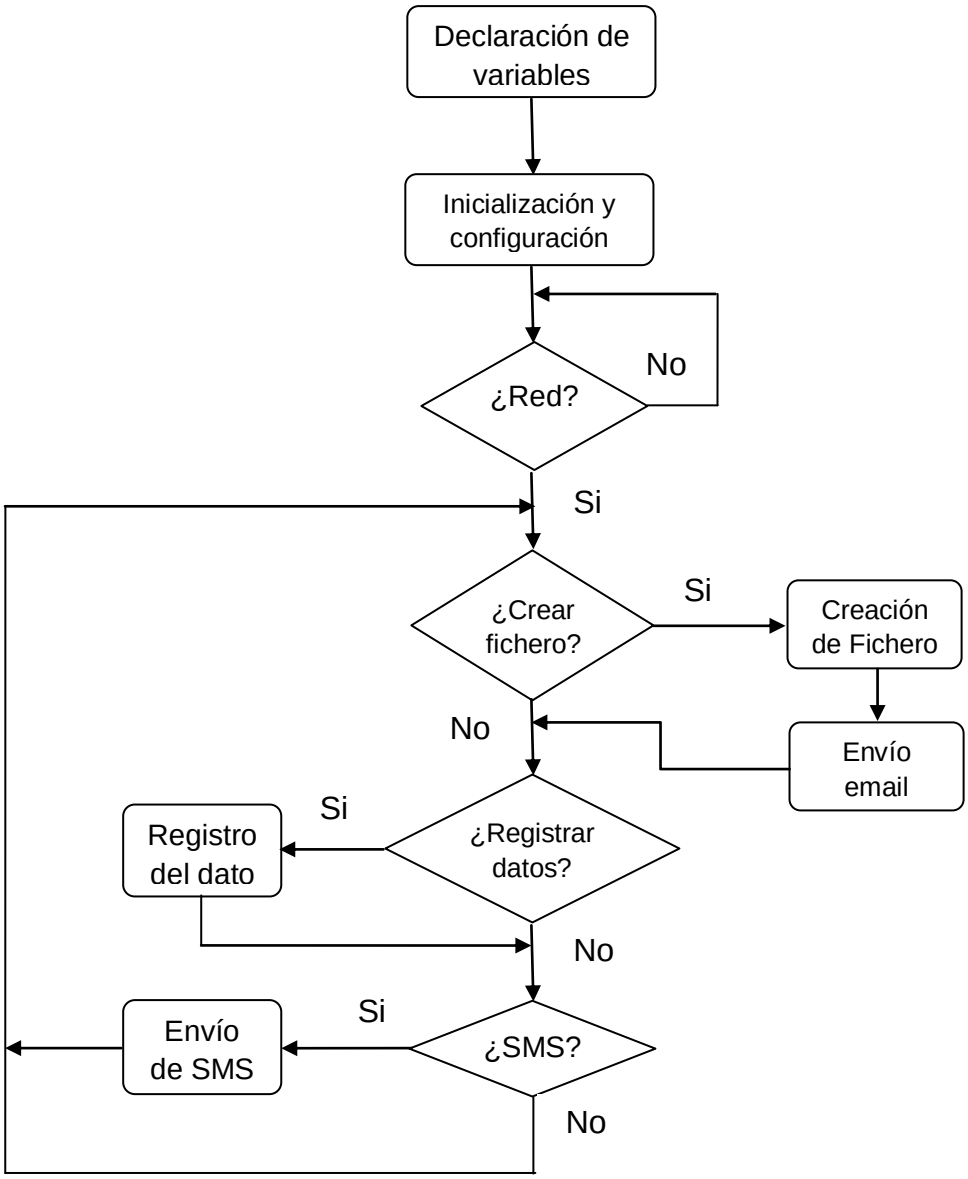

Figura 38. Diagrama de flujo de la programación del Arduino.

En caso de realizar un envío de SMS, se realizan las siguientes tareas:

- Comprobación del estado de la memoria del módem, por si hay algún SMS con petición de medida.
- Realización de una medida de CO_2.
- Preparación del texto del SMS en función del valor medido por el sensor de CO_2.
- Envío del SMS al número de teléfono que realizó la solicitud.

3.2 Procesado de datos

Este sistema se ha diseñado y programado para que realice medidas de concentración de CO_2 existentes en el ambiente. Dichas medidas se realizan de una manera periódica, cada hora, y son almacenadas en ficheros que crea el propio programa de forma diaria.

El procesado en sí de los datos no se realiza en el nodo sensor diseñado, este sistema solamente está preparado para la adquisición y envío de los datos, los cuales son de vital importancia para el control de las variables analizadas, en este caso la concentración de CO_2 en la ubicación donde se haya colocado el datalogger.

El sistema también permite el envío de los datos en tiempo real, lo que llamamos telemetría, cualquier persona que esté conectada a la red GSM mundial podrá tener acceso a los datos que se están midiendo en un determinado instante de tiempo, todo esto se puede realizar a través de un SMS. El nodo sensor permanece todo el tiempo a la espera de recibir un mensaje de texto con la solicitud de reenvió de la información solicitada, una vez recibido el SMS, se efectúa la medición de la variable y posterior envío del resultado vía SMS.

El procesado de datos sería realizado por un servidor de datos que se dedique a la gestión y explotación de la información, monitorizando y manteniendo las estaciones de medidas, todo esto lo realizaría con los datos que le llegan de nuestro nodo sensor.

1. COMANDOS AT

4.1 Introducción

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

Los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un terminal módem, se denominan así por la abreviatura de *attention*.

Aunque la finalidad principal de los comandos AT es la comunicación con módems, la telefonía móvil GSM también ha adoptado como estándar este lenguaje para poder comunicarse con sus terminales. De esta forma, todos los teléfonos móviles GSM poseen un juego de comandos AT específico que sirve de interfaz para configurar y proporcionar instrucciones a los terminales, permiten acciones tales como realizar llamadas de datos o de voz, leer y escribir en la agenda de contactos y enviar mensajes SMS, además de muchas otras opciones de configuración del terminal.

Un sistema que implemente el conjunto de comandos Hayes se considera compatible Hayes. Parte del conjunto de comandos Hayes fue incluido por la ITU-T en el protocolo V.25ter, actual V.250. La adopción de este estándar hizo el desarrollo de controladores específicos para distintos módems superfluo.

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

4.2 Comandos AT Y módem GSM

Aunque al principio los comandos AT se utilizaban exclusivamente para la comunicación con módems, éstos se han extendido a la telefonía GSM que también los han adoptado. Existen muchos teléfonos móviles que aceptan este tipo de lenguaje, con comandos AT de carácter general, pudiéndose utilizar estos equipos como módem GSM.

Por otro lado existen los módems GSM propiamente dichos, éstos equipos no tienen la parte visual de manejo, es decir, no tienen ni pantalla ni teclado como los teléfonos móviles, sin embargo tienen la posibilidad de gestionar su base de datos, contactos telefónicos, enviar SMS, realizar llamadas, ajustes de configuración, etc.

Los módems GSM son más específicos en lo que respecta al funcionamiento y a las instrucciones que acepta. Con respecto a su configuración, pruebas o conexión con otros módems se realizan por medio de un ordenador o un microcontrolador ya que no disponen de interfaz gráfica, para su manejo el módem incorpora puerto RS232 o USB.

Dependiendo de la marca o modelo del módem contienen una lista específica de comandos de acuerdo a sus prestaciones y necesidades. Es por ello que al utilizar un módem específico se debe buscar el conjunto de instrucciones AT que se requiera para su manejo.

4.3 Sintaxis de los comandos AT

Los comandos AT están compuestos por cadenas de caracteres ASCII que para su ejecución se debe anteponer el prefijo “AT” a excepción de los comandos de pause y de repetición de comando anterior en los que no se requiere. El prefijo AT deriva de la palabra “ATention” que solicita al módem ponga atención a la solicitud del comando presente.

En la especificación de los comandos AT, se detalla que deberán ser enviados en mayúsculas, aunque actualmente, casi todos los proveedores de módulos GSM admiten comandos en minúsculas.

Para enviar comandos AT a un módem GSM se debe seguir la siguiente estructura:

AT +	CMGF=1	<CR>
PREFIJO	COMANDO	SUFIJO

- El prefijo de los comandos AT debe ser la cadena de caracteres “AT”, el signo “+” también se coloca después de estos caracteres.
- El sufijo de los comandos AT debe ser <CR> Carriage Return (retorno de carro), es el equivalente a ENTER y su valor ASCII es 0Dh.
- Cuando se coloca el signo (=) a un comando, se está configurando un parámetro, cuando se coloca el signo interrogación (?) se está pidiendo información, la expresión (=?) se usa para obtener todo el rango de valores posibles que se pueden configurar.

La respuesta de un módem ante un comando tiene la siguiente estructura:

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

- Los caracteres “OK” corresponden a una operación exitosa, por otro lado y si aparece “ERROR” corresponde a una operación fallida.
- El prefijo y sufijo constan de <CR> Carriage Return (retorno de carro) y <LF> Line feed (final de línea).
- El valor ASCII de LF es 0Ah.

Tipo de comando AT	Sintaxis	Función
Prueba de comandos	AT+CXXX=?	El módem devuelve la lista de parámetros y rangos de los valores establecidos con el comando.
Comando de lectura	AT+CXXX?	Este comando devuelve el valor actual fijado del parámetro o parámetros del módem.
Comando de escritura	AT+CXXX=<…>	Este comando establece los valores de los parámetros del módem.
Comando de ejecución	AT+CXXX	Es el comando de ejecución.

Tabla 11. Tipos de comandos AT

4.4 Listado de comandos AT utilizados

A continuación se presentan un listado con los comandos AT utilizados en este proyecto así como las respuestas que retorna el módem al recibir dichos comandos.

El listado completo de los comandos compatibles con el módem TC65 lo podremos encontrar en el "ATC_Commands_V3", disponible en la web del fabricante.

– Configuración comunicación serie.

Para comprobar que el modem responde y además configurarlo a 19200 baudios, se utiliza el comando AT+IPR.

Se envía AT+IPR=19200 y se recibe un OK.

– Códigos de acceso.

Para consultar o introducir el código de acceso, se utiliza el comando AT+CPIN.

Primero debe consultarse el estado del código de acceso, enviamos AT+CPIN? Y podemos recibir las siguientes respuestas:

- +CPIN: SIM PIN: se requiere el código PIN.
- +CPIN: READY: el PIN ya se ha introducido con anterioridad.
- +CPIN: SIM PUK: se requiere el código de desbloqueo PUK.

En todos los casos, se recibe un OK a continuación.

Se envía AT+CPIN=1234 y se recibe un OK o ERROR dependiendo si el código PIN introducido es correcto o no.

- Lectura del número del Centro de Mensajes.

El comando AT+CSCA? solicita al módem el número de teléfono del Centro de Mensajes. Responde con la siguiente trama:

+CSCA: “+34609090909”, 145 seguido de un OK.

Este parámetro no lo necesitamos para el funcionamiento del dispositivo, simplemente lo leemos para almacenarlo en la memoria de datos del microcontrolador.

- Configuración tipo SMS.

El comando AT+CMGF configura el tipo de SMS que se va a manejar. Hay dos opciones modo TEXT y modo PDU. El modo PDU codifica el mensaje recibido, creando una trama ilegible directamente. Por ello, para facilitar la tarea, lo configuramos en Text Mode.

Se envía un AT+CMGF=1 y se recibe un OK.

- Configuración aviso SMS.

Para que el módem envíe un mensaje cuando reciba un nuevo SMS, debemos configurarlo con el comando AT+CNMI.

Se envía AT+CNMI=3,1,0,0 y se recibe un OK.

Donde ‘3’ es el valor que permite la recepción de un aviso sin solicitarlo, el ‘1’ solicita que se envíe además la posición en memoria donde se ha almacenado el mensaje recibido y los dos ‘0’ corresponden a funciones que no se van a manejar.

Al recibir un nuevo SMS, la trama que nos envía el módem es la siguiente:

+CMTI: “SM”, 1, donde “SM”

Indica que el mensaje se ha almacenado en la memoria de la tarjeta SIM y el ‘1’ indica que es la primera posición.

- Borrado de SMS.

Cada vez que se recibe y se trata un SMS, éste es borrado con el fin de asegurar que cualquier SMS recibido tenga sitio en la memoria para poder ser almacenado.

El comando utilizado es AT+CMGD=1 para borrar el mensaje 1 de la memoria y se recibe un OK.

- Envío de SMS.

Para el envío de mensajes SMS el procedimiento a seguir es algo más complejo.

El comando a usar es AT+CMGS, a continuación explicamos su funcionamiento con un ejemplo práctico:

Enviamos AT+CMGS=”+34666123456”,145 donde ”+34666123456” es el número del teléfono móvil destinatario y ‘145’ indica que el número de teléfono lo insertamos en formato internacional, con prefijo.

Una vez se ha enviado el comando, el módem responde con un carácter ‘>’ que nos indica que ya podemos introducir el cuerpo del mensaje. Una vez se haya introducido el texto que se quiere enviar, se debe finalizar el texto con Ctrl-Z, que es el carácter ascii 26.

El módem responderá con la posición donde se ha almacenado el mensaje enviado +CMGS: 1 seguido de un OK.

Todo el proceso quedaría de la siguiente forma:

→ AT+CMGS=”+34666123456”,145

← >

→ mensaje de prueba<ctrl-z>

← +CMGS: 1

← OK

- Lectura de SMS recibido.

Como se ha comentado anteriormente, cuando el módem recibe un nuevo SMS nos envía un aviso como éste:

+CMTI: “SM”, 1, donde “SM”

Indica que el mensaje se ha almacenado en la memoria de la tarjeta SIM y el ‘1’ indica que es la primera posición.

Una vez sabemos la posición en la que se ha almacenado el SMS, en el caso del ejemplo la 1, procederemos a enviar el comando AT de lectura AT+CMGR.

Se envía AT+CMGR=1 y se recibe la siguiente cadena:

+CMGR: “REC UNREAD”, “+34666123456”,,”16/07/2012,20:15:21+08”
Texto del mensaje OK

De la respuesta obtenida haremos uso del número de teléfono para responder a dicho número y el texto del mensaje para ver el tratamiento que se da al SMS recibido.