

MÓDULO BLUETOOTH HC-06

(Versión 29-11-17)

Algunos conceptos básicos de Bluetooth

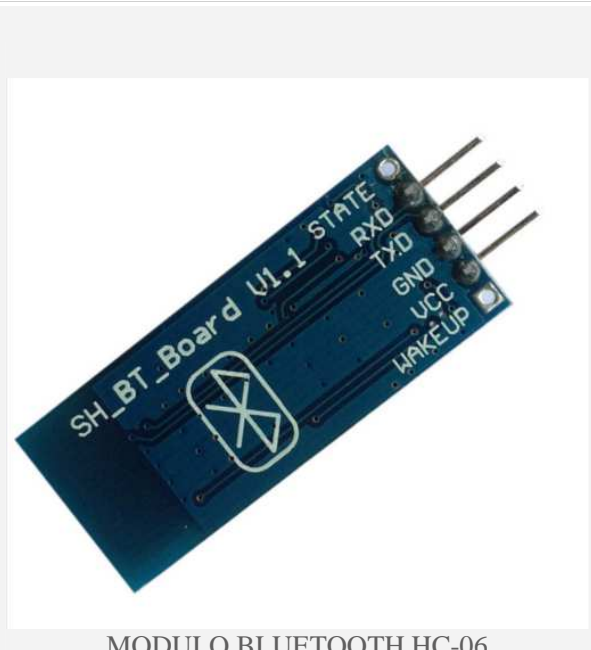
OBJETIVOS

- Presentar los conceptos básicos del Bluetooth.
- Presentar los comandos AT.
- Introducir el modulo Bluetooth HC-06.
- Presentar un programa básico de comunicación, entre un teléfono externo y un Arduino mediante el modulo Bluetooth.

MATERIAL REQUERIDO.



ARDUINO UNO REV 3



MODULO BLUETOOTH HC-06

Pocos de los que estén leyendo estas líneas, no tendrán un teléfono móvil en el bolsillo, y muy probablemente será del tipo Smartphone, que incluirán un sistema operativo como Android o Apple IOS equipados con Wifi y Bluetooth.

Por eso la posibilidad de integrar conexión WIFI o Bluetooth en nuestros proyectos, nos abre unas posibilidades inmensas. Poder controlar nuestros proyectos desde el propio móvil, bien con WIFI o bien con Bluetooth, es algo muy interesante para controlar montajes de diferentes tipos.

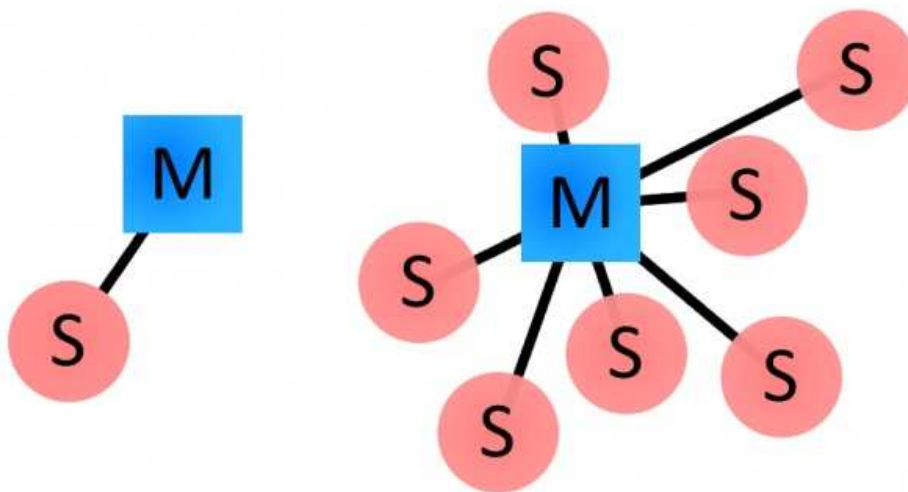
En las sesiones previas, vimos cómo usar el móvil conectado a la red local para gobernar múltiples salidas digitales, y en próximas sesiones veremos cómo usar el WIFI como sistema de control desde nuestros teléfonos.

Pero como los grandes viajes deben empezar con pequeños pasitos, vamos a dedicar un par de capítulos al Bluetooth, para ver de qué posibilidades disponemos, como configurarlos y demás. De modo que seamos capaces de integrar el Bluetooth con nuestros Arduinos.

Y para ello empezaremos hablando un poco, de que es el Bluetooth y de cómo funciona, así como unos pocos conceptos básicos claves, para poderlo usar con garantías de éxito.

Para empezar diremos que los dispositivos Bluetooth pueden actuar como Masters o como Slaves (Amos o esclavos).

La diferencia es que un Bluetooth Slave solo puede conectarse a un master y a nadie más, en cambio un master Bluetooth, puede conectarse a varios Slaves o permitir que ellos se conecten y recibir y solicitar información de todos ellos, arbitrando las transferencias de información (Hasta un máximo de 7 Slaves)



Cada uno de los dispositivos que se identifican vía Bluetooth presentan una dirección única de 48 bits y además un nombre de dispositivo que nos sirva para identificarlo cómodamente a los humanos. Por eso cuando configuras tu móvil puedes especificar un nombre propio que será el que mostrarás a los demás cuando busquen tu teléfono en las inmediaciones.

La dirección propia también se puede identificar pero lógicamente, es un poco menos cómoda y tiene menos utilidad. Tampoco es raro establecer un protocolo IP sobre transporte Bluetooth, con lo que además de su identificación interna Bluetooth (Equivalente al MAC Ethernet) dispondrá de una dirección IP para conectarse a Internet.

Por eso puedes conectarte vía Bluetooth a tu PC, por ejemplo, y a través de él conectarte a Internet.

Así pues un nodo Bluetooth puede ser Master o Slave y dispone de una dirección única, así como de un nombre para identificarse y muy habitualmente también incluye un PIN de conexión o número de identificación que debe teclearse para ganar acceso al mismo.

Como el Bluetooth lo desarrolló Nokia para conectar teléfonos móviles, a otros dispositivos como auriculares, micrófonos o conexiones al audio del coche, existe un procedimiento definido que se llama Pairing (o emparejamiento) que vincula a dos dispositivos Bluetooth.

Cuando vinculas dos dispositivos BT, se inicia un proceso en el que ellos se identifican por nombre y dirección interna y se solicitan la clave PIN para autorizar la conexión.

Si el emparejamiento se realiza con éxito, ambos nodos suelen guardar la identificación del otro y cuando se encuentran cerca se vuelven a vincular sin necesidad de intervención manual. Por eso el CD de tu coche reconoce el móvil de tu bolsillo en cuanto te subes y puedes reproducir la música que tienes en tu Smartphone.

➤ *Aunque para que tu Bluetooth pueda enviar o recibir música, debe aceptar otra norma posterior llamada Advanced Audio Distribution Profile (A2DP) y que en caso de ser algún sistema antiguo te impedirá la reproducción.*

Naturalmente, a lo largo de los años la norma ha ido variando y existen varias versiones de la misma, con compatibilidad siempre con las versiones anteriores que se diferencian en la distancia que pueden alcanzar (entre 50 y 100 metros, teóricamente y sin obstáculos) además de la velocidad de transferencia.

MÓDULOS BLUETOOTH DISPONIBLES PARA ARDUINO

Hace ya un tiempo que disponemos de módulos Bluetooth sencillos y económicos, que resultan muy prácticos para todo esto, y en esta sesión, vamos a empezar viendo cuáles tenemos disponibles y como trabajar con ellos.

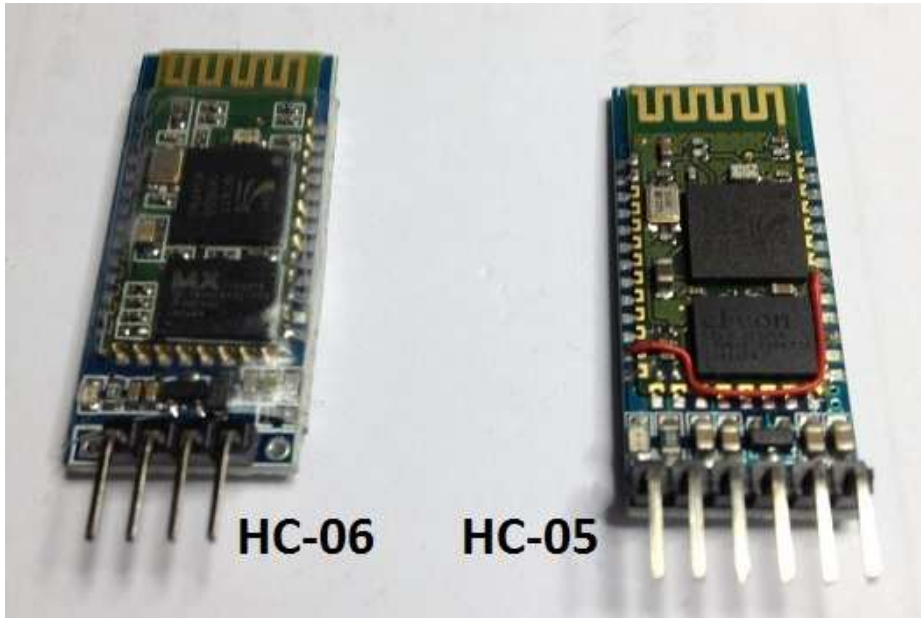
Los más frecuentes en el mercado son los módulos HC-06 y HC-05 que si hacéis una búsqueda por eBay o similares veréis que se consiguen por poco dinero, y están disponibles independientes o en modo SHIELD y para zocalo XBEE (ya hablaremos).

Hay bastante confusión en la red acerca de cómo diferenciar uno de otro y en muchas ocasiones sobre las prestaciones de unos y otros. Vamos a ver si podemos aclarar un poco el tema.

Lo primero es que el hardware de ambos módulos es el mismo. No hay diferencia hardware, pero el software que incorporan es diferente

- *De hecho, parece que es posible reprogramar uno para convertirlo en el otro, aunque el procedimiento es aparentemente bastante pesado.*

Además el módulo de conexión se monta sobre un soporte que a su vez puede presentar diferencias notables según el fabricante y las conexiones que realice, pero por lo que he podido comprobar con un par de módulos de los que dispongo, hay una diferencia obvia, el número de pines del módulo montado.



El modelo HC-06 dispone de 4 pines, en lugar de los 6 que incluye el modelo HC-05, pero hay además importantes diferencias de funcionalidad y de manejo que hace que merezca la pena dedicar una sesión a cada modelo.

Basicamente el modelo HC-06 solo puede actuar como esclavo y además dispone de un juego reducido de instrucciones a las que atiende, mientras que el modelo HC-05 puede actuar como master o como Slave y acepta un número mayor de órdenes de configuración.

En esta sesión vamos a usar el modelo HC-06, pero antes necesitamos hablar de la cuestión de los comandos AT.

LOS COMANDOS AT

En la época heroica, para enviar mensajes (de texto exclusivamente, por supuesto) de un ordenador a otro, usabas las líneas o puertos serie. Muy al estilo de la comunicación que hoy tu Arduino hace con tu PC a través del USB.

El interface no era USB sino RS232, que aunque físicamente diferente, comunicaban vía serie dos puntos próximos hasta un máximo, de digamos, unos 100 metros.

Pero cuando querías enviar información a otro equipo situado en una oficina remota de tu empresa, la única posibilidad era utilizar las líneas telefónicas, mediante un adaptador diabólico, que se llamaba Modem.

La idea básica, era codificar el 0 binario con un tono grave de audio (que se pudiera enviar por la línea de teléfono) y los unos binarios como un tono agudo.

Así con la sucesión más o menos rápida de tonos graves y agudos por la línea telefónica se podían enviar un mensaje binario codificado en frecuencia de audio, de un punto a otro.

- *La palabra Modem, que tal vez hayáis oído, deriva del apocope de MODulador DEModulador, y si veis pelis de los años 80 y primeros 90 eran esos chismes que hacían los ruiditos típicos de audio, asociados a la informática.*

Naturalmente, las líneas telefónicas eran analógicas, lo que implicaba una cantidad indecente de ruido térmico, que aumentaba exponencialmente con la distancia, obligando a disminuir la velocidad en función del ruido que te encontraras y a retransmitir una y otra vez el mensaje hasta que conseguías que llegara correctamente (algo que rara vez pasaba a la primera)

- La velocidad de los módems empezaron sobre los 300 baudios o bits por segundos y fueron aumentando con la revolución tecnológica hasta 1.200, 2.400, 9.600 y que yo sepa los últimos que se comercializaron con el ese nombre fueron de 56.000 baudios. (Compara esos 56k de máximo con las redes de cable modernas de 100 Mbps, casi 1.800 veces más rápidas)

Por eso, los módems necesitaban una especie de comandos, que nos permitiera modificar la velocidad según necesitáramos, léase bajarla, cuando el ruido impedía una comunicación fiable y ya de paso modificar algún que otro parámetro sobre la marcha.

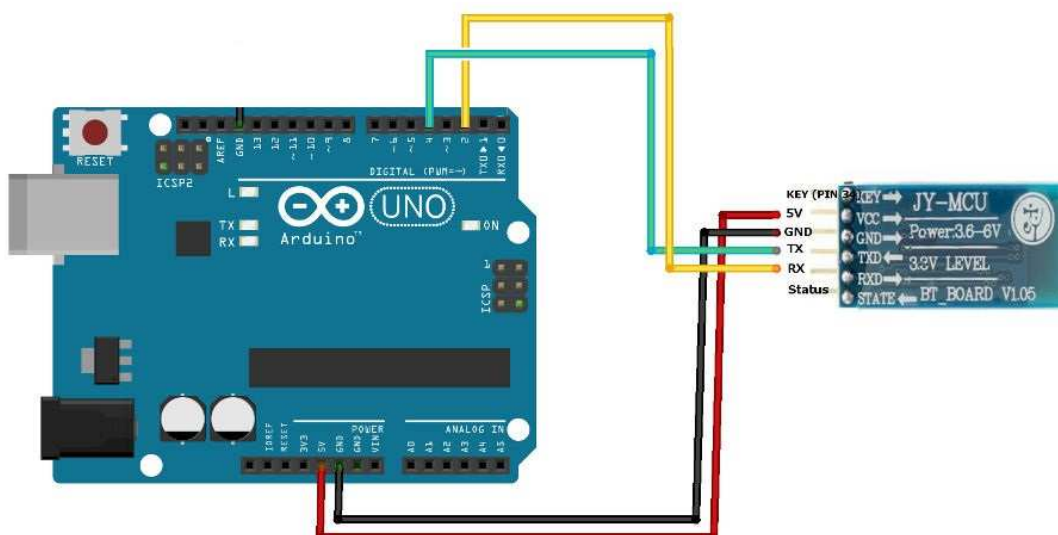
Y como el RS232 solo disponía de dos hilos de comunicaciones, no había más remedio que incorporar una orden de atención que significara que a continuación venía una orden de programación del modem, que no debía ser transmitida al otro extremo.

Es decir que las ordenes eran del tipo "AT+Orden", donde AT era el comando especificado de atención. En un santiamén todos los módems y demás máquinas de comunicación serie empezaron a aceptar este tipo de órdenes, y al conjunto de ellas se llamó comandos AT, por extensión.

Con el tiempo, mucho del hardware sencillo que se comunica con otros equipos vía una humilde puerta serie, siguen aceptando ordenes AT para configurarlos y este es el caso de los módulos Bluetooth HC-05, HC-06 y otros pequeños dispositivos que veremos en el futuro.

CONEXIÓN DEL BLUETOOTH HC-06 AL ARDUINO

Pues es cantidad de fácil. Fíjate que el módulo tiene rotulado el nombre de los pines, y simplemente conectar tensión y GND al módulo y después conectamos Txd y Rxd a los pines digitales 2 y 4 respectivamente de tu Arduino. Eso es todo



- Como el módulo HC-06 funciona a 3,3 V hay bastante controversia en internet respecto a si debemos o no, poner divisores de tensión para acceder a los pines del módulo con Arduino. Por las pruebas que he hecho no parece necesario y no veo ninguna razón para que a largo plazo deban surgir problemas, por lo que yo conecto directamente los pines al Arduino.

Fijaros que al hacer esta conexión, el LED del módulo HC-06, parpadea continuamente. Esto indica que no está pareado o vinculado.

Cuando conectes algo al módulo, esta luz se quedará fija y es la forma de saber si hay conexión o no.

EL PROGRAMA DE CONTROL

Como el modulo Bluetooth es básicamente un nodo BT conectado a un interface serie, podríamos en principio conectar los pines RX y Tx a los equivalentes de Arduino en los pines 0 y 1 digitales, sin más que cruzarlos (BT Tx a Arduino Rx y BT Rx a Aduano Tx) y de hecho muchos ejemplos en Internet utilizan este esquema y se comunican con el BT mediante las familiares instrucciones de Serial.print ().

Sin embargo, prefiero desaconsejar hacerlo así, porque los pines 0 y 1 se utilizan en la comunicación serie de Arduino con el PC a través del USB y por tanto, si los usamos para comunicar con el modulo BT, perderíamos la conexión con el PC.

Por ello prefiero destinar otro par de pines cualesquiera a la transmisión, aunque para ello tenemos que importar una librería que habilite la comunicación serie con otros pines como es la **librería SoftwareSerial**.

Para ello importamos la librería que viene de serie en el IDE y creamos un nuevo objeto serie llamado BT1 conectado a los pines 4 y 2:

```
#include <SoftwareSerial.h>
SoftwareSerial BT1(4,2); // RX, TX
```

Y después, podemos usar BT1 exactamente igual a como usamos Serial.

Vamos a escribir un programa para comunicarnos con el modulo BT que simplemente lee lo que se escribe en la puerta BT1 y si hay algo lo manda a la consola Serial.

A su vez si hay algo en la consola, lee una línea con GetLine() y finaliza con intro antes de enviar la línea completa al BT1

- *El motivo es, que el modulo HC-06, a diferencia del HC-05, no espera un terminador de línea como \n, si no que salta por tiempo, lo que nos fuerza a escribir muy deprisa los comandos largos y resulta incómodo..*
- *Si no escribimos el comando completo a tiempo, simplemente lo ignorará y ni siquiera dará error. Por eso prefiero mandar comandos completos directamente y no carácter a carácter.*

El programa queda algo así:

```
#include <SoftwareSerial.h>
SoftwareSerial BT1(4,2); // RX, TX recorder que se cruzan

void setup()
{
  Serial.begin(9600);
  Serial.println("Enter AT commands:");
  BT1.begin(9600);
}

void loop()
{
  if (BT1.available())
    Serial.write(BT1.read());

  if (Serial.available())
  { String S = GetLine();
    BT1.print(S);
    Serial.println("---> " + S);
  }
}

String GetLine()
{ String S = "" ;
  if (Serial.available())
  { char c = Serial.read(); ;
```

```

while ( c != '\n') //Hasta que el caracter sea intro
{
    S = S + c ;
    delay(25) ;
    c = Serial.read();
}
return( S + '\n' ) ;
}
}

```

Según el manual el modulo HC-06 viene de serie configurado a 9600 y por eso usamos

```
BT1.begin(9600);
```

Si por cualquier motivo, no consigues establecer comunicación con el modulo BT, vete cambiando la velocidad hasta que veas que hay conexión.

Fíjate ahora, en el modulo, verás una luz roja parpadeando. Eso significa que el modulo está listo para vincularse a un dispositivo Bluetooth o para aceptar comandos AT, el modulo HC-06 no necesita nada mas para entrar en modo de aceptar comandos AT.

- *De hecho siempre que veas la luz parpadeando significa que el modulo no retransmitirá los comandos AT, sino que los interpretara como ordenes internas.*
- *Pero si realizas una conexión (y la luz roja se queda fija, ignorará los comandos AT y los transferirá sin más.*

Si no te has vinculado aún, y la luz roja parpadea, ya podemos enviar comandos AT. Veamos algún ejemplo.

- Con el **HC-06** es necesario que en el Monitor Serial seleccionemos “**Sin ajuste de línea**”.
- Si usamos el **HC-05**, por el contrario, tendremos que seleccionar “**Ambos NL & CR**”.

El primero y más básico es enviar (siempre en mayúsculas) un simple AT desde la consola. Se supone que debería responder con OK, pero en mi caso y con mi modulo no es así. Y si vuestro caso es el mismo deberéis probar con más comandos.

AT+VERSION, Requiere la versión del Firmware

AT+NAMEXXX, Programa el nombre que queremos presentar cuando alguien nos busque:

AT+NAMEPROMETEC

AT+BAUDX, Fija la velocidad de comunicación entre el modulo y la consola según a la siguiente tabla:

1 configura	1200bps
2 configura	2400bps
3 configura	4800bps
4 configura	9600bps (Default)
5 configura	19200bps
6 configura	38400bps
7 configura	57600bps
8 configura	115200bps

Ejemplo: AT+BAUD7 configura la comunicación a 57600 baudios

AT+PINXXXX, configura el número de identificación personal, que se requerirá para establecer la vinculación

AT+PIN4516, establece 4516 como PIN

En caso de que probando varios de estos comandos, no obtuvierais respuesta, o vierais caracteres extraños en la consola, probad con otras velocidades hasta conseguir una que funcione. Probad diferentes velocidades en la línea:

```
BT1.begin(9600);
```

Y me temo que parece no haber más comandos disponibles para un módulo HC-06 y desde luego no dispone de los comandos de interrogación de los que si dispones su hermano mayor HC-05.

PROBANDO LA CONEXIÓN CON EL MODULO HC-06

Una vez volcado el programa anterior a nuestro Arduino, con una velocidad de comunicación correcta con el modulo vamos a probar a enviar y recibir información desde y hacia, un teléfono móvil.

Para ello voy a utilizar un teléfono Android y un programa Android llamado Bluetooth SPP, pero en realidad servirá cualquier terminal BT.

- *Debo indicar que no he sido capaz de vincular este módulo a mi Iphone 4S, porque este insiste en que no existe nada en los alrededores con lo que vincularse y en cambio mi viejo HTC Android 2.1 lo ve a la primera y sin problemas.*

Lo primero es confirmar que el LED del módulo BT parpadea, indicando que está en modo AT o esperando vinculación con otro dispositivo. Después asegúrate de que el Bluetooth del teléfono está activo y por último, dependiendo del programa que uses como terminal, pídele que busque dispositivos BT en los alrededores y vincúlate al que encuentres.

Si todo va bien ya podrás enviar y recibir mensajes de texto entre tu móvil y Arduino en ambas direcciones.

Aquí os dejo un pequeño mini video con el resultado:

RESUMEN DE LA SESIÓN

- Hemos presentado algunos conceptos básicos del mundo Bluetooth y su relación con Arduino.
- Vimos lo que son los comandos AT y de donde provienen, así como algunos de ellos que sirven para programar el modulo HC-06.
- Empezamos viendo algunas diferencias entre el HC-05 y el HC-06.
- Escribimos un programa básico para comunicar un teléfono Bluetooth con Arduino mediante un programa de terminal Bluetooth.



PROMETEC

www.prometec.net