

Se denomina Bluetooth al protocolo de comunicaciones diseñado especialmente para dispositivos de bajo consumo, que requieren corto alcance de emisión y basados en transceptores de bajo costo.

Los dispositivos que incorporan este protocolo pueden comunicarse entre sí cuando se encuentran dentro de su alcance. Las comunicaciones se realizan por radiofrecuencia de forma que los dispositivos no tienen que estar alineados y pueden incluso estar en habitaciones separadas si la potencia de transmisión es suficiente. Estos dispositivos se clasifican como “Clase 1”, “Clase 2” o “Clase 3” en referencia a su potencia de transmisión.

La especificación de Bluetooth define un canal de comunicación a un máximo 720 kbit/s (1 Mbit/s de capacidad bruta) con rango óptimo de 10 m (opcionalmente 100 m con repetidores). Opera en la frecuencia de radio de 2,4 a 2,48 GHz con amplio espectro y saltos de frecuencia con posibilidad de transmitir en Full Duplex(los módulos de Arduino no) con un máximo de 1600 saltos por segundo. Los saltos de frecuencia se dan entre un total de 79 frecuencias con intervalos de 1 MHz; esto permite dar seguridad y robustez.

El hardware que compone el dispositivo Bluetooth está compuesto por dos partes:

- un dispositivo de radio, encargado de modular y transmitir la señal.**
- un controlador digital, compuesto por una CPU, un procesador de señales digitales (DSP – Digital Signal Processor) llamado Link Controller (o controlador de Enlace) y de las interfaces con el dispositivo anfitrión.**

Los dispositivos BlueTooth pueden actuar como Masters o como Slaves. 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 la inmediaciones.

Para manejar los módulos Bluetooth usamos comandos Hayes o AT:

Comandos AT básicos

•AT+VERSION, versión del Firmware

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

•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

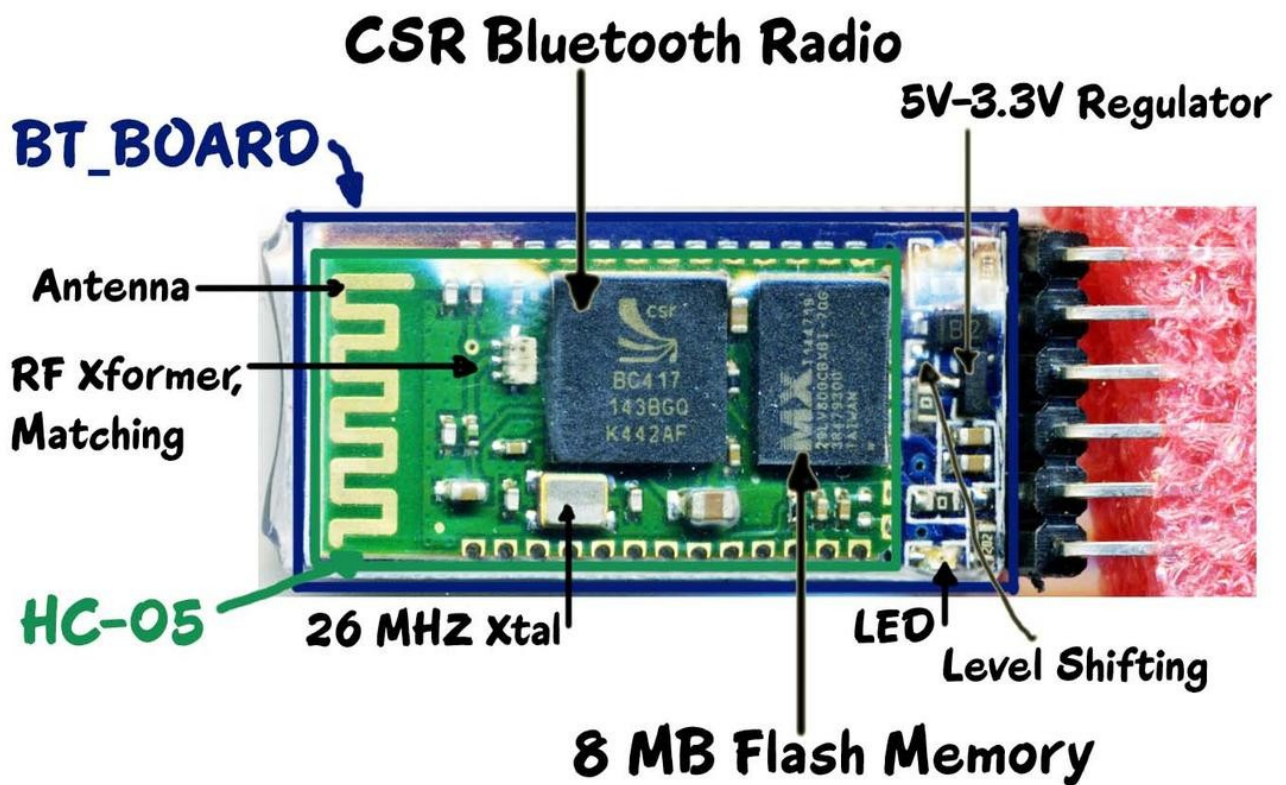
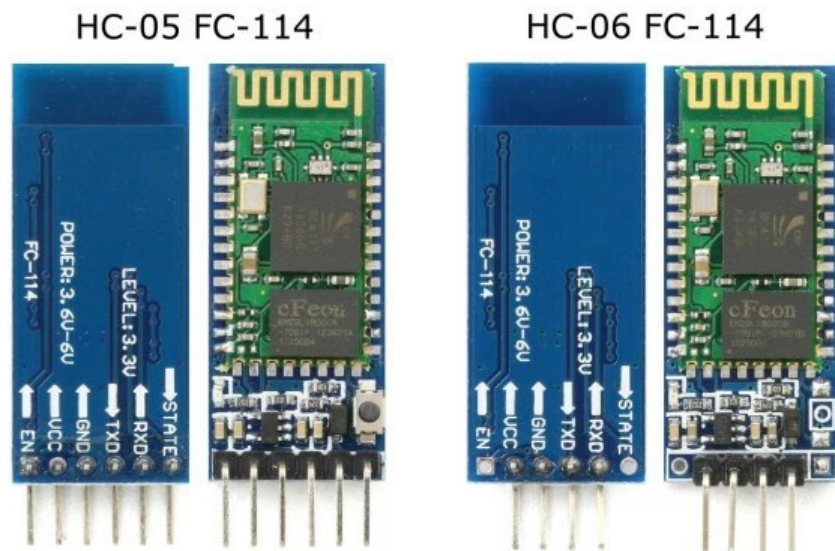
•AT+PINXXXX, configura el número de identificación personal, que se requerirá para establecer la vinculación

•AT+ROLE Nos informa de si está configurado como Maestro 1, o como esclavo 0.

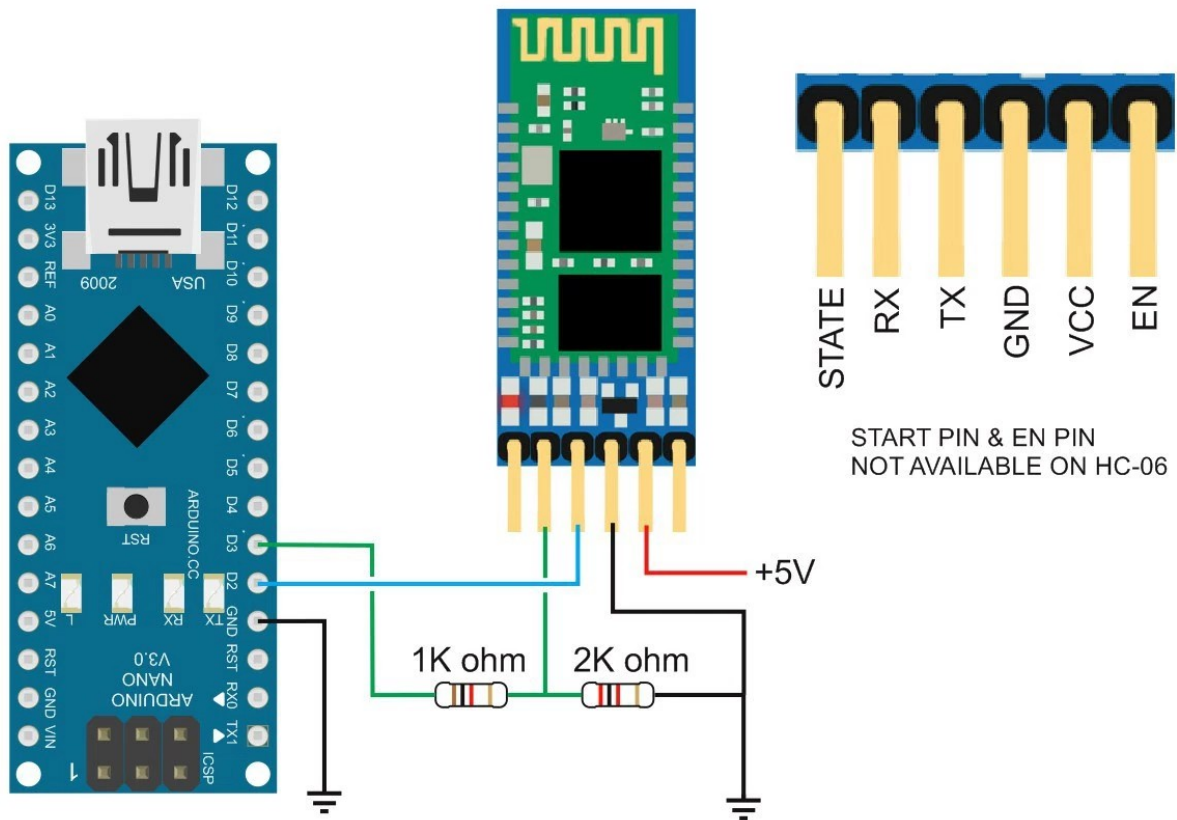
•AT+ROLE1 Configura el modulo como Master.

•AT+ROLE0 Configura el modulo como Slave

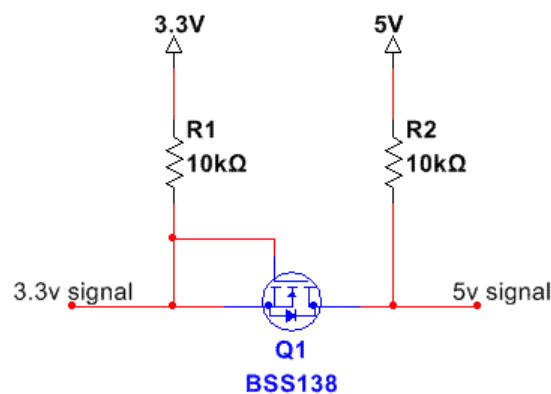
Los módulos más frecuentes en el mercado son los módulos HC-06 y HC-05

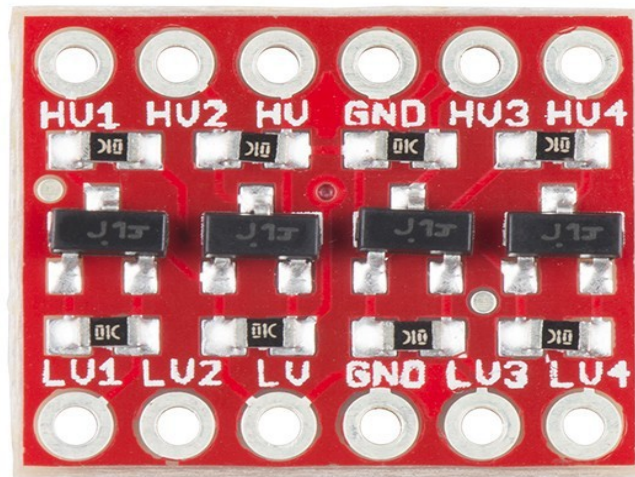


HC-05 FC-114 & HC-06 FC-114 Connections to Arduino



Arduino trabaja a 5V y el módulo bluetooth a 3.3, por lo tanto, utilizaremos un convertidor bidireccional lógico de 3.3 ↔ 5.





// configuracion bluetoothHC-05, (con pulsador)

const int LED=13;

const int BTPWR=12;

char nombreBT[10]="Ardu01";

char pin[5]="1234";

char velocidad='4'; //9600

char modo='0';// 0 escl. 1 maest

void setup() {

pinMode(LED,OUTPUT);

```
pinMode(BTPWR,OUTPUT);  
Serial.begin(38400); // Velocidad configuración  
digitalWrite(BTPWR,LOW);  
digitalWrite(LED,HIGH); //pulsamos pulsador HC-05  
delay(4000);  
digitalWrite(LED,LOW);  
digitalWrite(BTPWR,HIGH); //pulsamos pulsador HC-05  
delay(8000);
```

```
Serial.print("AT\r\t"); //AT + salto de linea
```

```
Serial.print("AT+NAME:");// : o = si no funciona // con las tres opciones "  
'=' o ':'
```

```
Serial.print(nombreBT);  
Serial.print("\\r\\t");
```

```
Serial.print("AT+NAME=");// : o = si no funciona  
Serial.print(nombreBT);  
Serial.print("\\r\\t");
```

```
Serial.print("AT+NAME");// : o = si no funciona  
Serial.print(nombreBT);  
Serial.print("\\r\\t");
```

```
Serial.print("AT+PIN");// : o = si no funciona, o AT+PSWD  
Serial.print(pin);  
Serial.print("\\r\\t");  
Serial.print("AT+PIN:");// : o = si no funciona, o AT+PSWD  
Serial.print(pin);  
Serial.print("\\r\\t");  
Serial.print("AT+PIN=");// : o = si no funciona, o AT+PSWD  
Serial.print(pin);  
Serial.print("\\r\\t");
```

```
Serial.print("AT+BAUD:");// : o = si no funciona  
Serial.print(velocidad);  
Serial.print("\\r\\t");
```



```
Serial.print("AT+BAUD");// : o = si no funciona
Serial.print(velocidad);
Serial.print("\r\t");
Serial.print("AT+BAUD=");// : o = si no funciona
Serial.print(velocidad);
Serial.print("\r\t");
```

```
Serial.print("AT+MODE");// : o = si no funciona
Serial.print(modos);
Serial.print("\r\t");
Serial.print("AT+MODE:");// : o = si no funciona
Serial.print(modos);
Serial.print("\r\t");
Serial.print("AT+MODE=");// : o = si no funciona
Serial.print(modos);
Serial.print("\r\t");
```

```
digitalWrite(LED,HIGH); //Terminado
}
```

```
void loop() {
// put your main code here, to run repeatedly:

}
```

```
// configuracion bluetoothHC-06, (con pulsador)
```

```
const int LED=13;
const int BTPWR=12;
```

```
char nombreBT[10]="ArduHC_06";  
char pin[5]="1234";  
char velocidad='4'; //9600
```

```
void setup() {  
  pinMode(LED,OUTPUT);  
  pinMode(BTPWR,OUTPUT);  
  Serial.begin(9600); // Velocidad configuración  
  delay(5000); // espera para ejecución
```

```
  digitalWrite(LED,LOW);  
  digitalWrite(BTPWR,HIGH);
```

```
  Serial.print("AT"); //AT  
  delay(2000); // espera para ejecución
```

```
  Serial.print("AT+NAME");  
  Serial.print(nombreBT);  
  delay(2000); // espera para ejecución
```

```
  Serial.print("AT+PIN");  
  Serial.print(pin);  
  delay(2000); // espera para ejecución
```

```
  Serial.print("AT+BAUD");  
  Serial.print(velocidad);  
  delay(2000); // espera para ejecución
```

```
  digitalWrite(LED,HIGH); //Terminado  
}
```

```
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```

Ejemplo lectura Arduino

```
#include <SoftwareSerial.h>  
  
SoftwareSerial BT1(8, 9); // RX | TX  
const int MESSAGE_MOVE = 0x03;  
void setup()  
{  
  delay (500) ; // Espera antes de encender el  
  modulo  
  Serial.begin(9600);  
  BT1.begin(9600);  
}  
  
void loop()  
{ if (BT1.available())  
  Serial.write(BT1.read());  
  if (Serial.available())  
    BT1.write(Serial.read());  
}
```