

Working with Arduino Bluetooth Module



HC-05 Bluetooth Module

HC-05 PinOut (Right) :

- **KEY:** If brought HIGH before power is applied, forces AT Command Setup Mode. LED blinks slowly (2 seconds)
- **VCC:** +5 Power
- **GND:** System / Arduino Ground
- **TXD:** Transmit Serial Data from HC-05 to Arduino Serial Receive. NOTE: 3.3V HIGH level: OK for Arduino
- **RXD:** Receive Serial Data from Arduino Serial Transmit
- **STATE:** Tells if connected or not

Modes of HC-05 - COMMAND and DATA TRANSFER MODES:

The module has two modes of operation, Command Mode where we can send AT commands to it and Data Mode where it transmits and receives data to another bluetooth module.

The default mode is DATA Mode, and this is the default configuration, that may work fine for many applications:

- Baud Rate: 9600 bps, Data : 8 bits, Stop Bits: 1 bit, Parity : None, Handshake: None
- Passkey: 1234

- Device Name: HC-05

In some cases you may want to change some of the configuration setup values. For that you have to get into Command Mode by following the procedure given below:

Commands are generally sent to the module in UPPERCASE and are terminated with a CR/LF pair.

1. Unplug power from HC-05
2. Upload sketch given down in your Arduino
3. Hold in HC-05 button
4. (Holding the button) Reconnect power to HC-05 (wait until LED blinks slowly)
5. Open Serial Monitor
6. Make sure "Both NL & CR" is selected
7. Type AT commands

Command Mode Commands:

The format of commands is:

- Always starts with "AT"
- Then "+" followed by <ParameterName>
- Then either:
 - o ? (returns current value of parameter)
 - o = (New Value of parameter)

A few examples:

- AT (AT Test command. Should respond with OK)
- AT+VERSION? (show the firmware version)
- AT+UART=9600,0,0 (Set baud rate to 9600, 1 stop bit, no parity)

Programs:

HC-05 at Command prompt:

(For this program to work, HC-05 must be in command mode)

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial BTSerial(10, 11); // RX | TX
```

```
void setup()
```

```

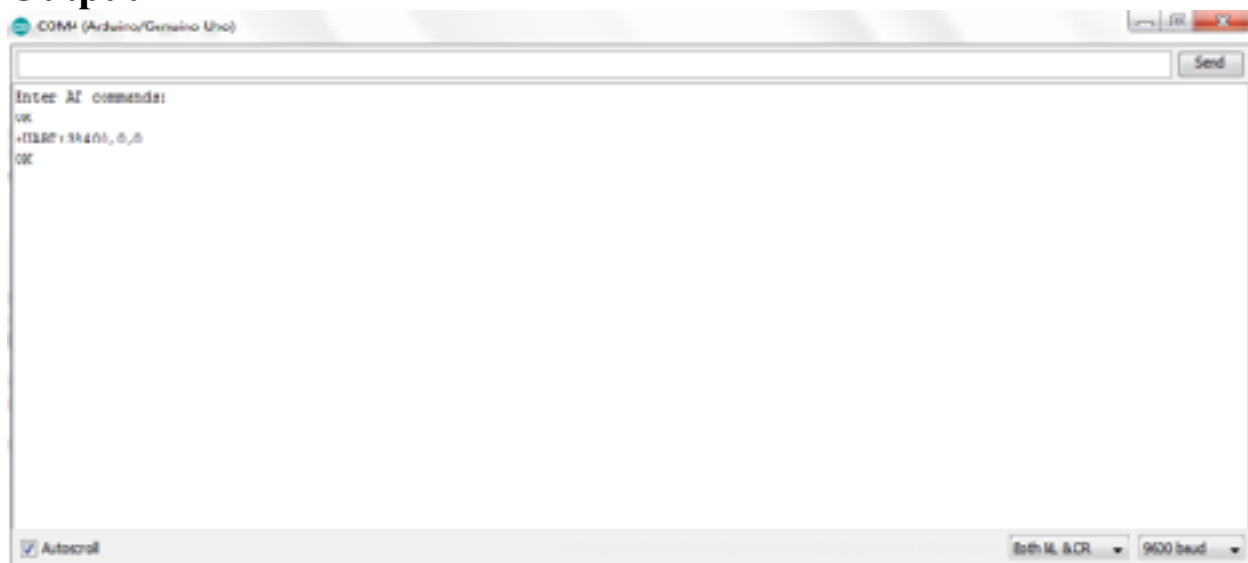
{
    Serial.begin(9600);
    Serial.println("Enter AT commands:");
    BTSerial.begin(38400); // HC-05 default speed in AT command mode
}

void loop()
{
    // Keep reading from HC-05 and send to Arduino Serial Monitor
    if (BTSerial.available())
        Serial.write(BTSerial.read());

    // Keep reading from Arduino Serial Monitor and send to HC-05
    if (Serial.available())
        BTSerial.write(Serial.read());
}

```

Output



HC-05 Controlled by mobile:

(For this code to work, HC-05 must be in DATA mode and Arduino Bluetooth App)

```

#define ledPin 13
int state = 0;
void setup() {
    pinMode(ledPin, OUTPUT);
    digitalWrite(ledPin, LOW);
    Serial.begin(38400);
    // Default communication rate of the Bluetooth module
}

```

```

void loop() {
  if(Serial.available() > 0){
    // Checks whether data is coming from the serial port
    state = Serial.read(); // Reads the data from the serial port
  }
  if (state == '0') {
    digitalWrite(ledPin, LOW); // Turn LED OFF
    Serial.println("LED: OFF");
    // Send back, to the phone, the String "LED: ON"
    state = 0;
  }
  else if (state == '1') {
    digitalWrite(ledPin, HIGH);
    Serial.println("LED: ON");
    state = 0;
  }
}

```

BT-Master Slave

Slave Mode:

The HC-05 bluetooth module can also act as a slave. There are fewer commands to set this up:

AT+ORGL Reset to defaults

AT+RMAAD Clear any paired devices

AT+ROLE=0 Set mode to SLAVE

AT+ADDR Display SLAVE address

Master Mode:

To configure the module as Bluetooth Master and to pair with another bluetooth module follow these steps. First we need to put the module into command mode

Enter these commands in order:

AT+RMAAD Clear any paired devices

AT+ADCN

AT+ROLE=1 Set mode to Master

AT+CMODE=0 Allow master to ONLY connect to bound address (slave). This allows the master to automatically connect to the slave when switched on
AT+PSWD=1234 Set PIN. Should be same as slave device

AT+BIND=<address> Set bind address to the slave address
AT+LINK=<address> Connect to slave.
AT+INIT

BT-Slave Program:

```
#include <SoftwareSerial.h>

SoftwareSerial BTSerial(10, 11); // RX | TX

void setup() {
  Serial.begin(9600);
  BTSerial.begin(38400); // HC-05 default speed in AT command mode
}

void loop() {
  // Reading the button
  if(Serial.available())
  {
    String message = Serial.readString();
    Serial.println (message);
    BTSerial.write(message.c_str());
  }
}
```

BT-Master Program:

```
#include <SoftwareSerial.h>

SoftwareSerial BTSerial(10, 11); // RX | TX

#define ledPin 9

String message;
int potValue = 0;

void setup() {
  pinMode(ledPin, OUTPUT);
```

```

digitalWrite(ledPin, LOW);
Serial.begin(9600);
BTSerial.begin(38400); // HC-05 default speed in AT command mode
}

void loop() {
  if(BTSerial.available() > 0){
    // Checks whether data is coming from the serial port
    // Reads the data from the serial port
    message = BTSerial.readString();
    // Controlling the LED
    if(message.indexOf("SWITCH ON")>=0)
    {
      digitalWrite(ledPin, HIGH); // LED ON
    }
    else if(message.indexOf("SWITCH OFF")>=0)
    {
      digitalWrite(ledPin, LOW); // LED OFF
    }
    else
    {
      Serial.println("Nothing to do");
    }
    delay(100);
  }

  delay(10);
}

```

