Welcome!

Thank you for purchasing our *AZ-Delivery Bluetooth Wireless RF Transceiver Module*. On the following pages, you will be introduced to how to use and set up this handy device.

Have fun!

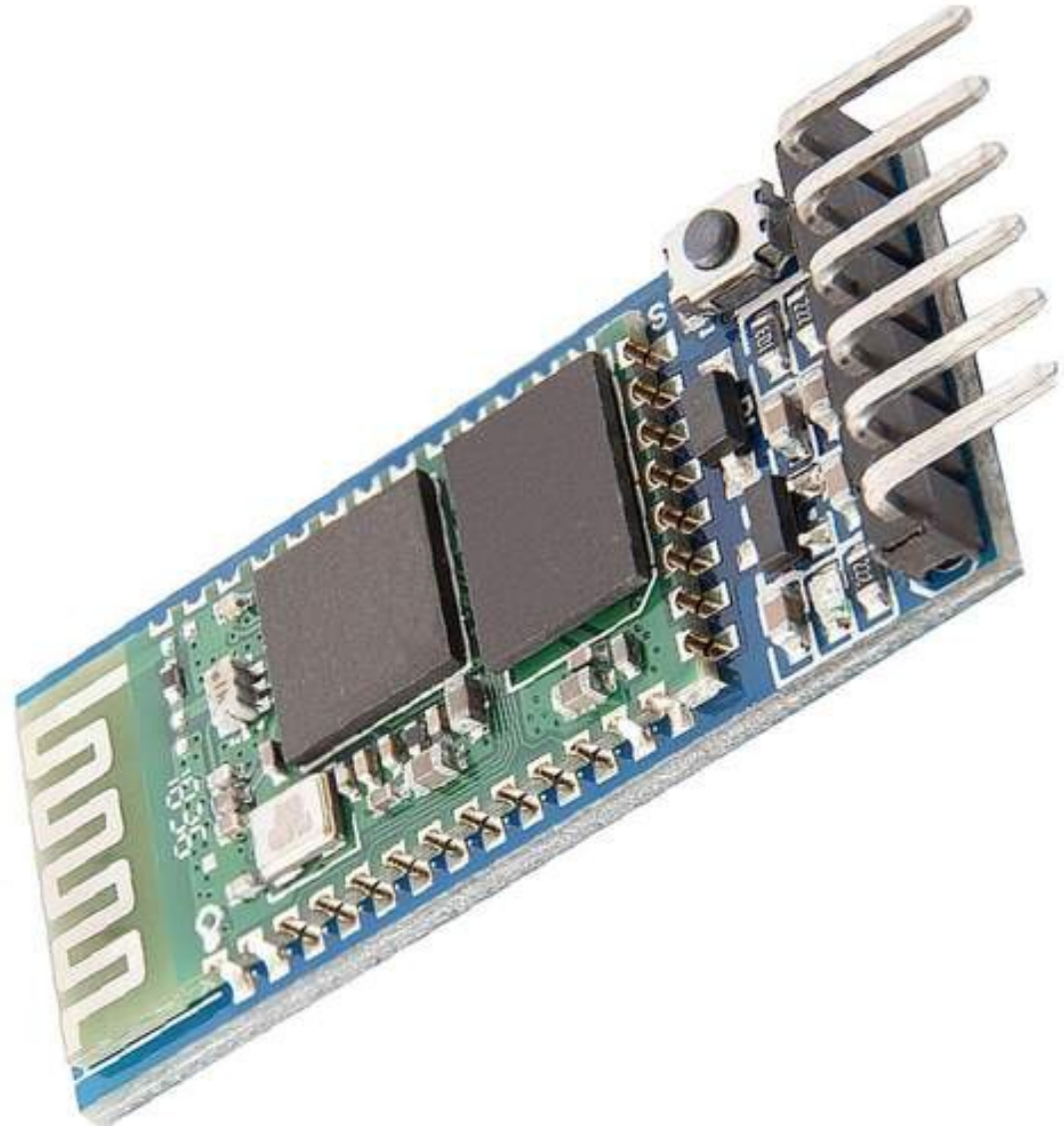


Table of Contents

Introduction

Bluetooth is a wireless technology standard which is used for transferring data between fixed and mobile devices over short ranges using UHF (Ultra High Frequency) radio waves. It is a conventional wire-replacement communication protocol initially created for lower power consumption, based on low-cost transceiver microchips in each device. As the devices use a radio communication system, they do not have to be in visual line of sight of each other, but a quasi-optical wireless path must be viable. The actual range depends on the qualities of the devices, as well as the air conditions and other factors. The effective range varies depending on propagation conditions, material coverage, production sample variations, antenna configurations and battery conditions.

A transceiver is a device that is able to both transmit and receive information, which is usually accomplished via radio waves.

HC-05 is more capable Bluetooth device, which can be set to either master or slave, while HC-06 can be only set as a slave device. The module has two modes of operation - command response work mode where we can send AT commands to it and automatic connection work mode where it transmits and receives data to another Bluetooth module.

The module can dynamically switch between the roles of Slave and Master.

Specifications

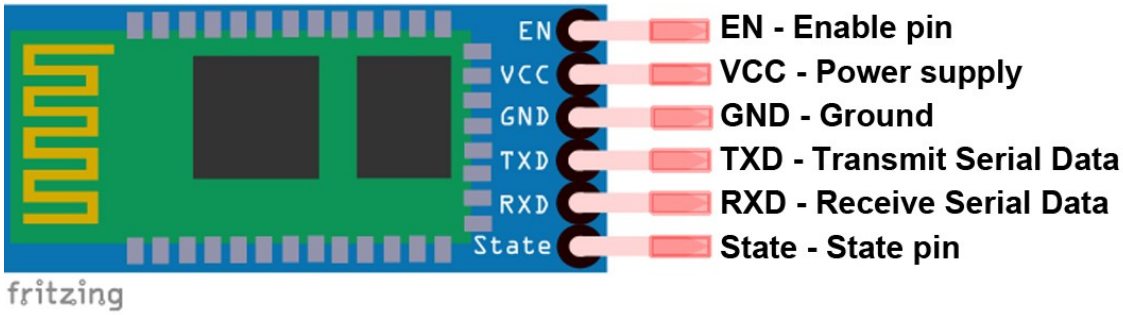
»	Range:	up to 10m (class 2)
»	Frequency:	2.45GHz
»	Power supply:	3.3 to 5V
»	Dimensions:	37mm + 8mm [1.45in + 0.3in], 15mm [0.6in], 2mm [0.07in]
»	Device name:	HC-05
»	Password:	1234
»	Serial UART interface	

The module has a LED for status display installed. When flashing fast, it means the device is ready for connection; when flashing slowly, the connection is established and when flashing every 2 seconds AT mode is on.

The device has on-board 5V to 3.3V regulator and a switch button.

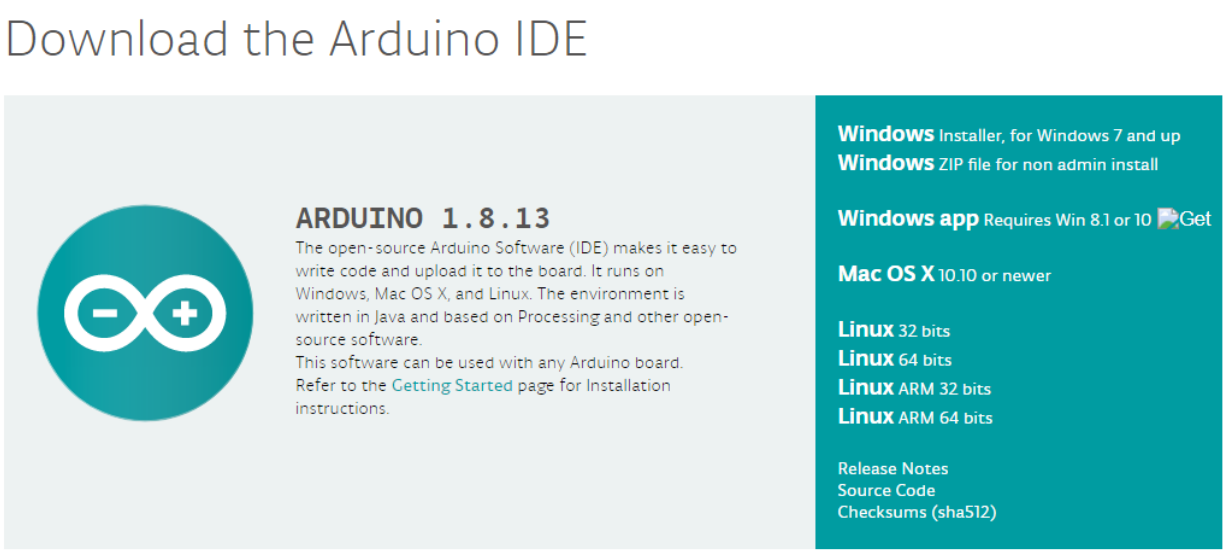
The Pinout

The Bluetooth Wireless RF Transceiver Module has 6 pins. The pinout is shown in the following image:



How to set-up Arduino IDE

If the Arduino IDE is not installed, follow the *link* and download the installation file for the operating system of choice. The Arduino IDE version used for this eBook is **1.8.13.**



For *Windows* users, double click on the downloaded *.exe* file and follow the instructions in the installation window.

For *Linux* users, download a file with the extension *.tar.xz*, which has to be extracted. When it is extracted, go to the extracted directory and open the terminal in that directory. Two *.sh* scripts have to be executed, the first called *arduino-linux-setup.sh* and the second called *install.sh*.

To run the first script in the terminal, open the terminal in the extracted directory and run the following command:

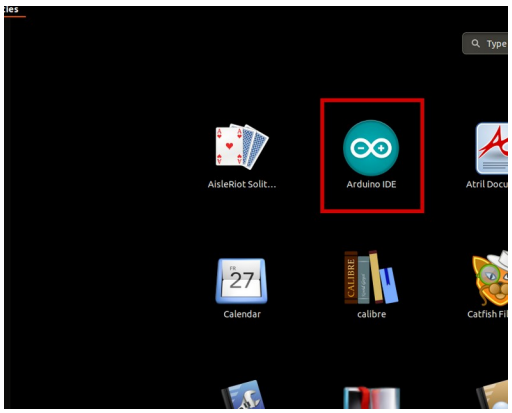
sh arduino-linux-setup.sh user_name

user_name - is the name of a superuser in Linux operating system. A password for the superuser has to be entered when the command is started. Wait for a few minutes for the script to complete everything.

The second script, called *install.sh*, has to be used after the installation of the first script. Run the following command in the terminal (extracted directory):

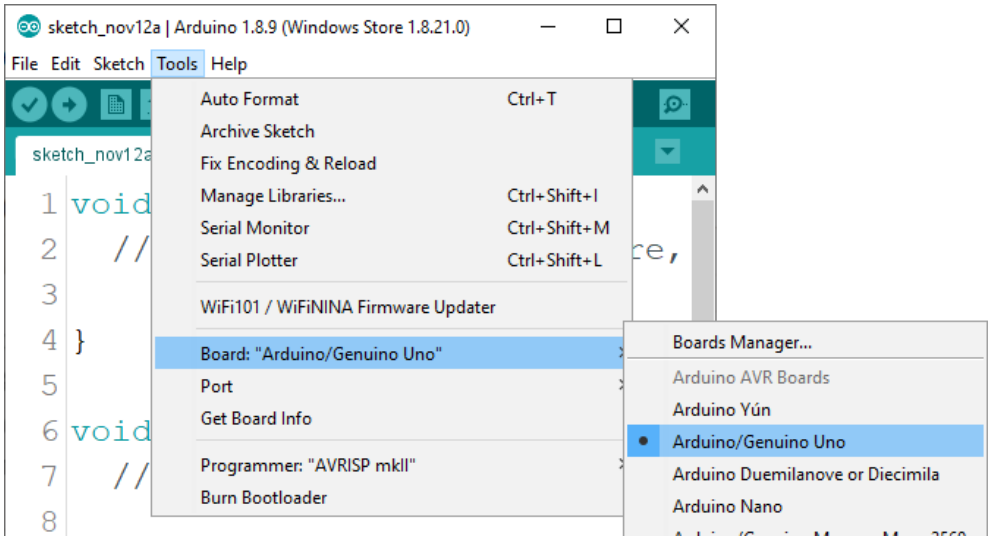
sh install.sh

After the installation of these scripts, go to the *All Apps*, where the *Arduino IDE* is installed.



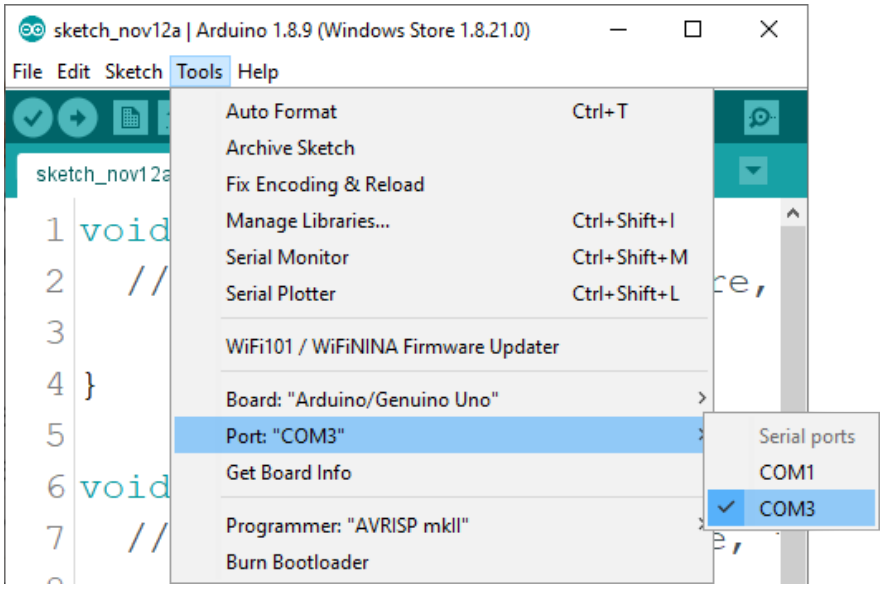
Almost all operating systems come with a text editor preinstalled (for example, *Windows* comes with *Notepad*, *Linux Ubuntu* comes with *Gedit*, *Linux Raspbian* comes with *Leafpad*, etc.). All of these text editors are perfectly fine for the purpose of the eBook.

Next thing is to check if your PC can detect an Arduino board. Open freshly installed Arduino IDE, and go to:
Tools > Board > {your board name here}
{your board name here} should be the *Arduino/Genuino Uno*, as it can be seen on the following image:



The port to which the Arduino board is connected has to be selected. Go to:
Tools > Port > {port name goes here}
and when the Arduino board is connected to the USB port, the port name can be seen in the drop-down menu on the previous image.

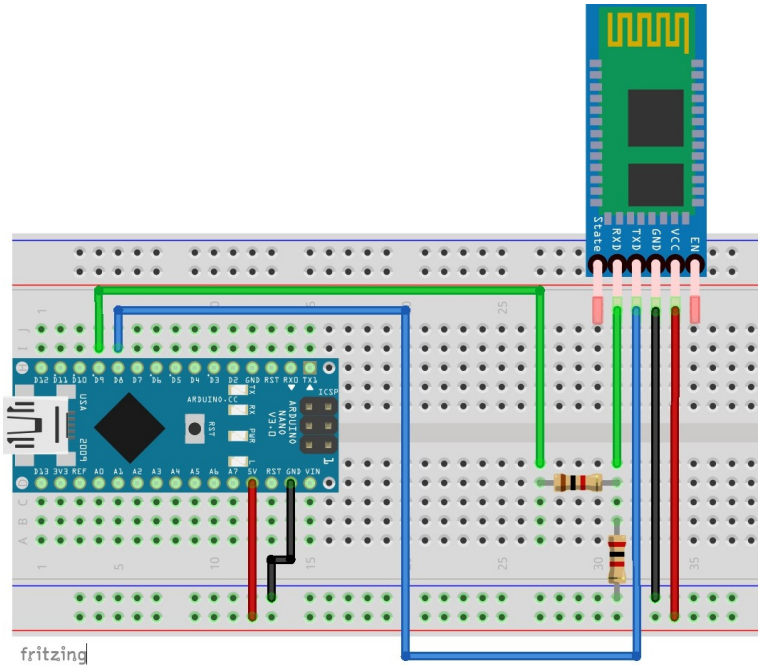
If the Arduino IDE is used on Windows, port names are as follows:



For *Linux* users, for example, port name is */dev/ttyUSBx*, where *x* represents integer number between *0* and *9*.

AT Mode

Connect the module with Nano as shown in the following image:



HC-05 pin	Nano pin	Wiring color
RXD	D9	**Green wire**
TXD	D8	**Blue wire**
GND	GND	**Black wire**
VCC	5V	**Red wire**

When the module is properly connected, the LED flashes continuously (about 5 times per second). In AT command mode, the LED only flashes for 2 seconds every 2 seconds.

To put the HC-05 in the AT command mode, it is needed to disconnect the module from the power, and hold the small gray push-button on the board while reconnecting. Alternatively, you can connect the Enable pin to VCC, before connecting the Power Supply.

The Arduino IDE does not require external libraries to operate the HC-05, as we only use the included SoftwareSerial library.

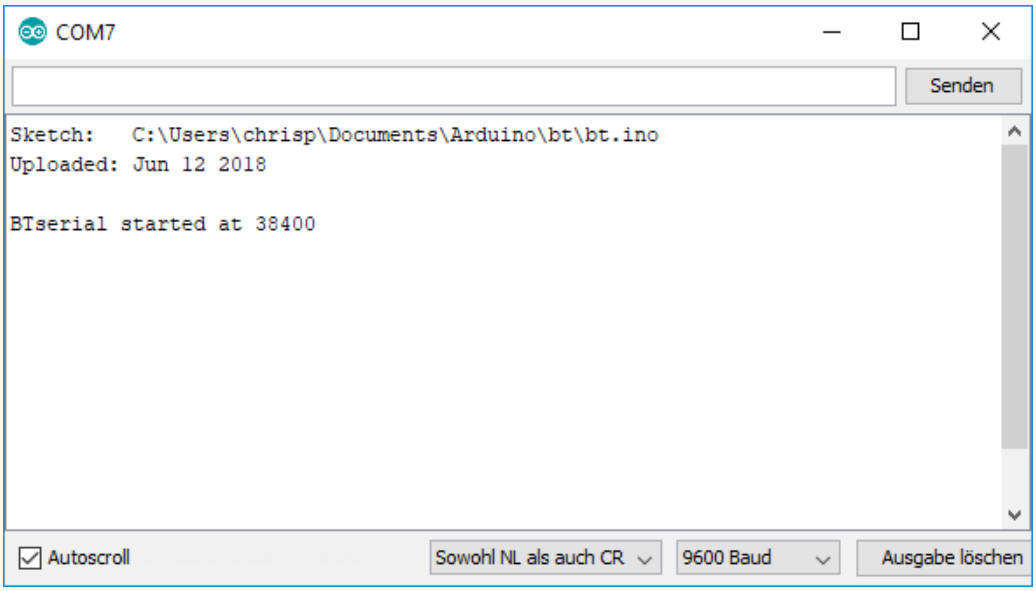
Now your hardware and IDE are ready.

Sketch example

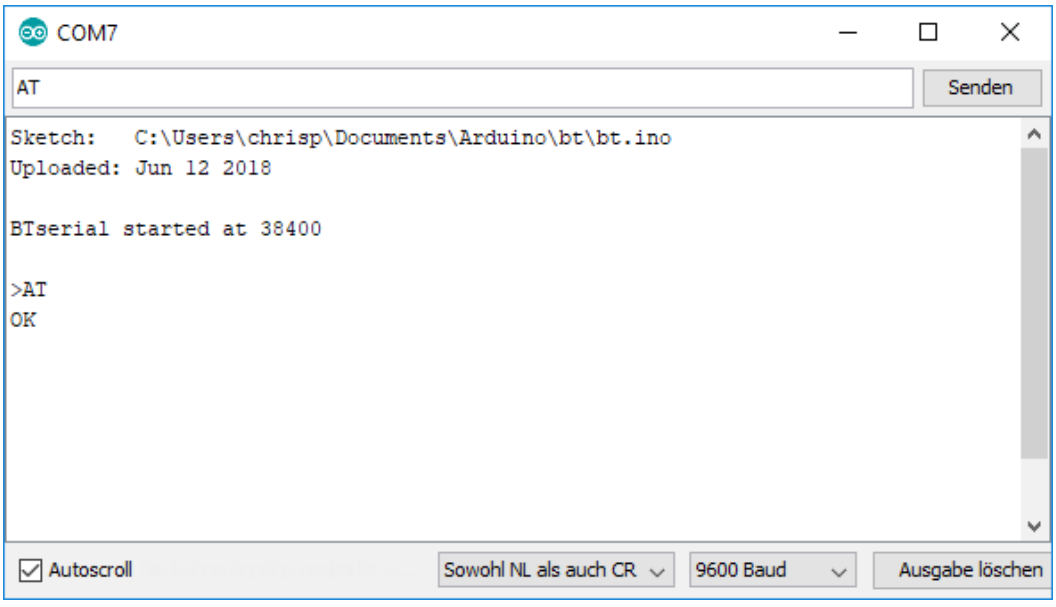
```
// SerialIn_SerialOut_004
// Uses hardware serial to talk to the host computer and Software Serial for communication with the bluetooth module
// What ever is entered in the serial monitor is sent to the connected device
// Anything received from the connected device is copied to the serial monitor
// Pins
// BT VCC to Arduino 5V out.
// BT GND to GND
// Arduino D8 (SS RX) - BT TX no need voltage divider
// Arduino D9 (SS TX) - BT RX through a voltage divider (5v to 3.3v)

#include <SoftwareSerial.h>
SoftwareSerial BTserial(8, 9); // RX, TX

char c=' '; boolean NL = true;


 void setup()
 {
      Serial.begin(9600);
      Serial.print("Sketch:   "); Serial.println(__FILE__); Serial.print("Uploaded:   ");
      Serial.println(__DATE__); Serial.println(" ");


      BTserial.begin(38400); Serial.println("BTserial started at 38400");
      Serial.println(" ");
 }
```

```
 void loop()
{
      // Read from the Bluetooth module and send to the Arduino Serial Monitor
      if (BTserial.available())
{
            c = BTserial.read();
            Serial.write(c);
      }

      // Read from the Serial Monitor and send to the Bluetooth module
      if (Serial.available())
      {
            c = Serial.read(); BTserial.write(c);

          // Echo the user input to the main window. The ">" character indicates
 the user entered text.
            if (NL) { Serial.print(">");         NL = false; }
Serial.write(c);
            if (c==10) { NL = true; }
      }
 }
```

After uploading the sketch to Nano, open the serial monitor and set the speed to 9600 baud, and at the end of the line, "Both NL and CR".



First, check if the module is accessible. Enter the command AT in the serial console and click on "Send" (or confirm with the Enter key). The module answers with "OK".



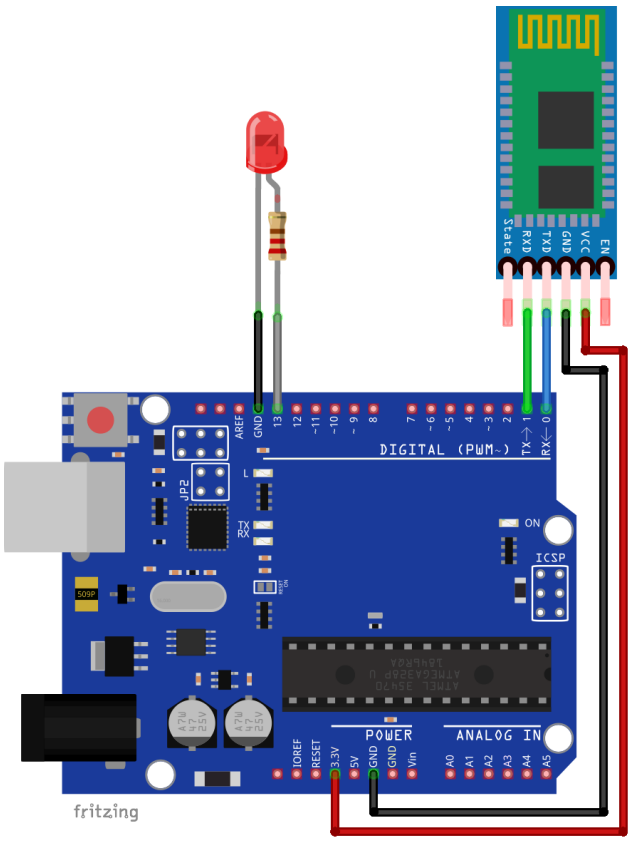
There is a variety of AT commands supported by HC-05. Here is a brief overview of the most important ones:

Command	Description
AT	Test the serial communication. Returns "OK"
AT+VERSION?	Returns the firmware version "VERSION:3.0-20170601"
AT+ORGL	Reset to original configuration
AT+RESET	Restart the module
AT+NAME?	Returns the name of the module
AT+NAME=newName	Rename the module to "newName".
AT+ADDR?	Returns the MAC-Address as Hexadecimal values
AT+STATE?	Returns the current state: INITIALIZED READY PAIRABLE PAIRED INQUIRING CONNECTING CONNECTED DISCONNECTED UNKNOWN
AT+ROLE?	Returns the current ROLE: 0 = slave mode 1 = master mode 2 = slave echo mode

AT+ROLE=x	Switch the current Role: 0 = slave mode 1 = master mode 2 = slave echo mode
AT+CMODE?	Returns the current mode. For CMODE to work, the module has to be in Master Modus (see AT+ROLE command). 0 = Manual connection. The device address must be provided 1 = Auto connect 2 = Slave loop Mode
AT+PSWD?	Returns the current PIN
AT+PSWD=xxxx	Set the PIN-Code for pairing.
AT+PAIR=address,timeout	Pair with other Bluetooth device. the address format is: 1234, 56, abcdef. The timeout - value is in seconds

Connecting Uno with the module for normal operations

Connect the module, resistor and LED diode with the Uno board as shown on the following connection diagram:



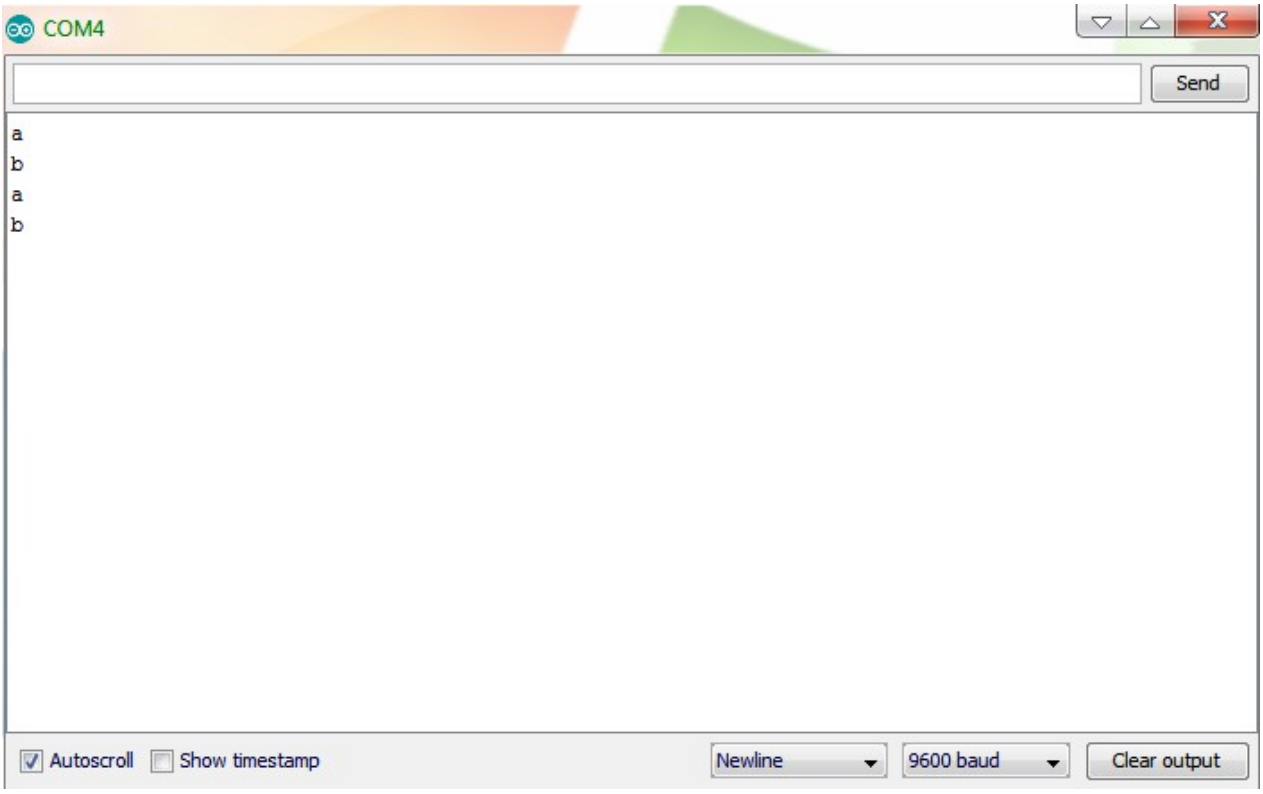
HC-05 pin	Uno pin	Wiring color
GND	GND	**Black wire**
VCC	3.3V	**Red wire**
RXD	TX	**Green wire**
TXD	RX	**Blue wire**
Resistor	**Uno pin**	**Wiring color**
Resistor	13	**Grey wire**
LED diode	**Uno pin**	**Wiring color**
LED diode	GND	**Black wire**

Sketch example

```
char junk;
String inputString="";

void setup() {
  Serial.begin(9600);
  pinMode(13, OUTPUT);
}

void loop() {
  if (Serial.available()) {
    while (Serial.available()) {
          char inChar = (char) Serial.read();
           inputString += inChar;
         }
       Serial.println(inputString);
        while (Serial.available() > 0)  {
            junk = Serial.read() ;
          }
        if (inputString == "a") {
             digitalWrite(13, HIGH);
          }
         else if (inputString == "b") {
              digitalWrite(13, LOW);
          }
         inputString = "";
    }
  }
```

For this project, it is needed to download Bluetooth Terminal on your Smartphone. When the code is uploaded on the Uno, pair your Smartphone and HC-05, the password is '1234'. When it's done, find the module in Bluetooth Terminal and connect it. Then type 'a' in the terminal, and LED turns on, type 'b' and LED turns off.

What is typed in Bluetooth terminal is displayed in the Arduino IDE serial monitor.



Now it is the time to learn and make your own projects. You can do that with the help of many example scripts and other tutorials, which can be found on the internet.

If you are looking for the high quality products for Arduino and Raspberry Pi, AZ-Delivery Vertriebs GmbH is the right company to get them from. You will be provided with numerous application examples, full installation guides, eBooks, libraries and assistance from our technical experts.

https://az-delivery.de

Have Fun!

Impressum

https://az-delivery.de/pages/about-us