

Bluetooth Connection between Arduino Uno and Android Phone Using HC-05 Bluetooth Module

What is Bluetooth?

Bluetooth is the most popular WPAN (Wireless Personal Area Network) technology.

Bluetooth is a standardized protocol for sending and receiving data via a 2.4GHz wireless link.

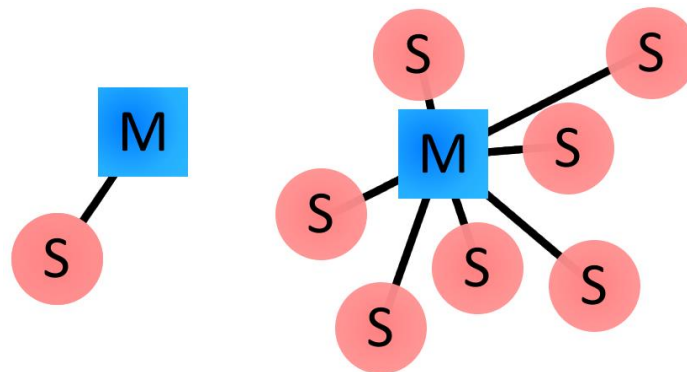
It's a secure protocol, and it's perfect for short-range, low-power, low-cost, wireless transmissions between electronic devices

Application:

Bluetooth serves as an excellent protocol for wirelessly transmitting relatively small amounts of data over a short range (<100m). It's perfectly suited as a wireless replacement for serial communication interfaces. You'll find Bluetooth embedded into a great variety of consumer products, like headsets

How Bluetooth Works?

Bluetooth devices all have a unique address, which is usually presented as a hexadecimal value. The Bluetooth protocol operates at 2.4GHz



Bluetooth networks (commonly referred to as **piconets**)

Bluetooth Addresses: Every single Bluetooth device has a unique address

Command of **"AT+ADDR?"** will get module Bluetooth address

Bluetooth Names: Bluetooth devices can also have user-friendly names given to them.

These are usually presented to the user, in place of the address

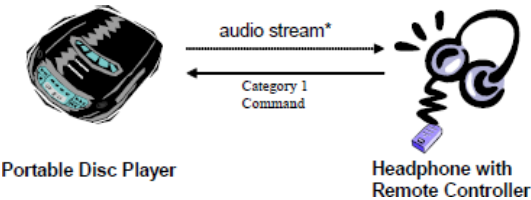
Command of **AT+NAME=<Param>** will set device's name

Bluetooth Profiles: Bluetooth profiles are additional protocols that build upon the basic

For two Bluetooth devices to be compatible, they must support the same profiles.

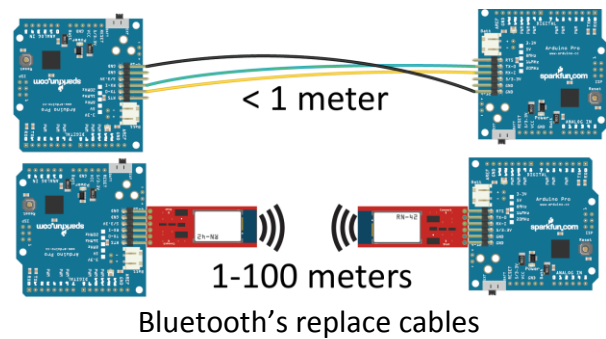
Example 1

The audio/video remote control profile (**AVRCP**) allows for remote controlling of a Bluetooth device.



Example 2

Serial Port Profile (**SPP**) sends bursts of data between two devices. It's one of the more fundamental Bluetooth profiles. Using SPP, each connected device can send and receive data just as if there were RX and TX lines connected between them. Two Arduinos, for example, could converse with each other from across rooms, instead of from across the desk.



Question: Can Serial Port Profile (**SPP**) Bluetooth device works with the audio/video remote control profile (**AVRCP**) Bluetooth device?

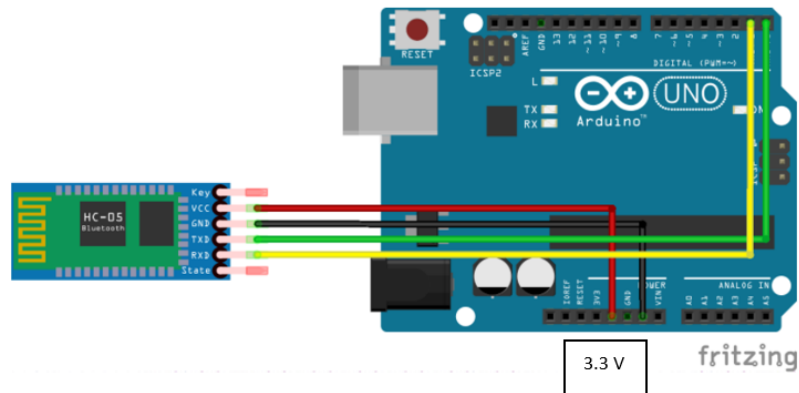
HC-05 Bluetooth Module

	Pin	Description	Function
	VCC	+5V	Connect to +5V
	GND	Ground	Connect to Ground
	TXD	UART_TXD, Bluetooth serial signal sending PIN	Connect with the MCU's (Microcontroller and etc) RXD PIN.
	RXD	UART_RXD, Bluetooth serial signal receiving PIN	Connect with the MCU's (Microcontroller and etc) TXD PIN.
	KEY	Mode switch input	If it is input low level or connect to the air, the mod is at paired or communication mode. If it's input hi level, the module will enter to AT mode.

For today's exercise,

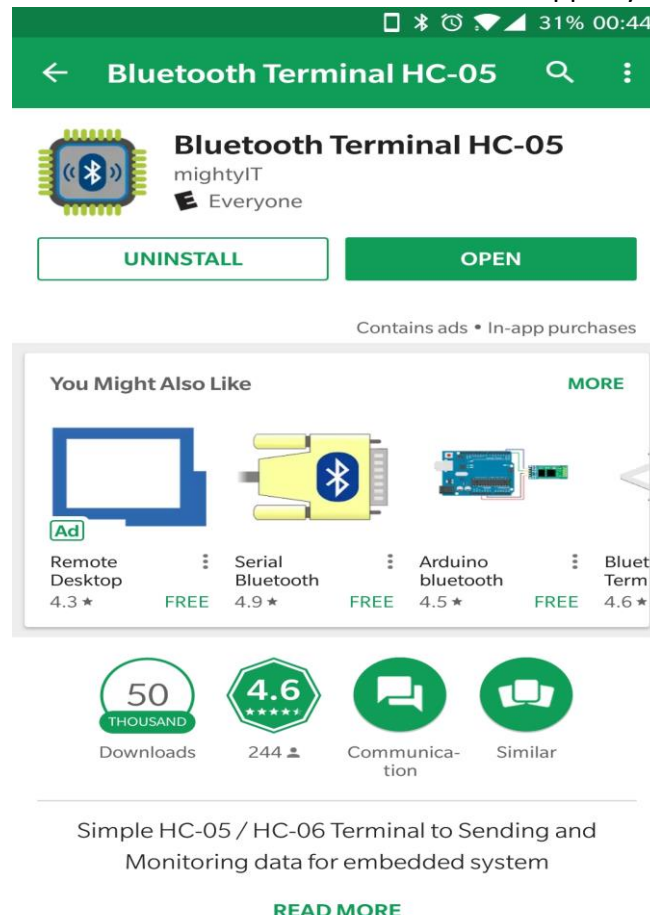
Material: Arduino Uno, LED, 220ohms Resistor, HC-05 Bluetooth Module, Android Phone with Bluetooth, Bluetooth Terminal HC-05 app

Step 1: Setup the Arduino Uno board with HC-05 Bluetooth Module.



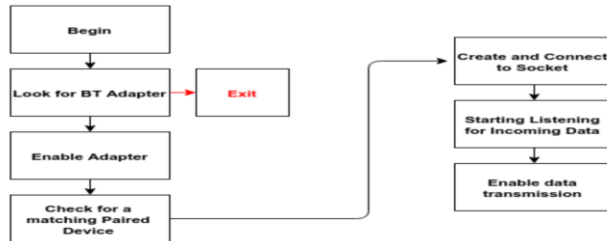
Connecting the HC05 Module

Step 2: Download and install “Bluetooth Terminal HC-05” app in your Android phone.



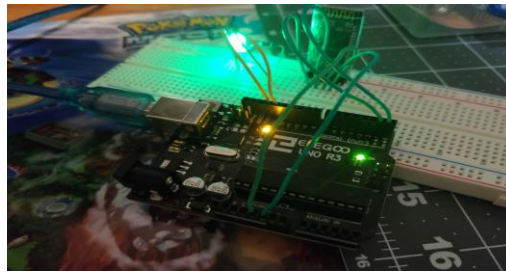
Step 3: Enabled the Bluetooth on the Android phone, and search HC-05 inside the app.

Step 4: Connect HC-05 and the app, enter passcode 1234 as default (you can change the name and the passcode later if you want to)



Step 4: Write the program and upload the code in Arduino IDE

In this exercise, you will control a LED using serial terminal by input “O” for turn on the LED and “F” for turn off the LED.



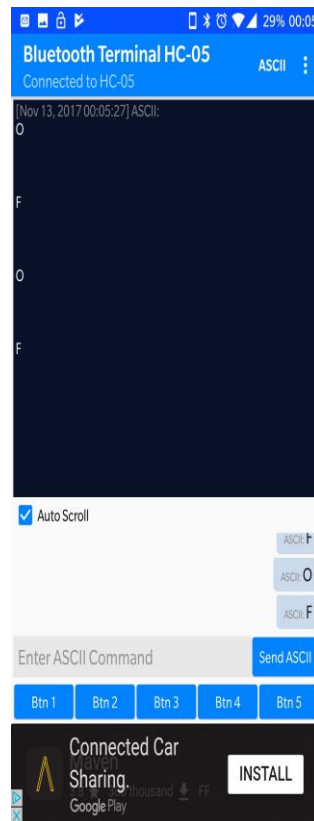
```

char LED;
String inputString="";

void setup()                    // run once, when the sketch starts
{
  Serial.begin(9600);           // set the baud rate to 9600, same should be of your Serial Monitor
  pinMode(13, OUTPUT);
}

void loop()
{
  if(Serial.available()){
    while(Serial.available())
    {
      char inChar = (char)Serial.read(); //read the input
      inputString += inChar;             //make a string of the characters coming on serial
    }
    Serial.println(inputString);
    while (Serial.available() > 0)
    { LED = Serial.read() ; }           // clear the serial buffer
    if(inputString == "O"){              //in case of 'O' turn the LED on
      digitalWrite(13, HIGH);
    }else if(inputString == "F"){        //incase of 'F' turn the LED off
      digitalWrite(13, LOW);
    }
    inputString = "";
  }
}
  
```

Step 5: After successfully pair HC-05 with the app, we can now enter the ASCII command to control the program. In this case by setting the Btn1 as “O” and Btn2 as “F”, we can either turn on the LED or turn it off.



Basic AT commands

Command	Return	Parameter	Description
AT	OK	None	Test
AT+VERSION?	+VERSION:<Param> OK	Param: Version number	Get the soft version
AT+ORGL	OK	None	Restore default status
AT+ADDR?	+ADDR: <Param> OK	Param: Bluetooth address	Get module Bluetooth address
AT+NAME=<Param>	OK	Param: Bluetooth device name	Set device's name
AT+NAME?	+NAME:<Param> OK	Param: Bluetooth device name	Inquire device's name
AT+ROLE=<Param>	OK	Param:0=Slave role; 1=Master role; 2=Slave-Loop role	Set module role
AT+ROLE?	+ROLE:<Param>	Param:0=Slave role; 1=Master role; 2=Slave-Loop role	Inquire module role
AT+UART=<Param>,<Param2>,<Param3>	OK	Param1: baud rate(bits/s); Param2: stop bit; Param3: parity bit	Set serial parameter
AT+ UART?	+UART=<Param>,<Param2>,<Param3> OK	Param1: baud rate(bits/s); Param2: stop bit; Param3: parity bit	Inquire serial parameter

Reference:

<http://microcontrollerslab.com/hc-05-bluetooth-module-interfacing-arduino/>

<http://www.instructables.com/id/How-to-Configure-HC-05-Bluetooth-Module-As-Master-/>

<https://www.bing.com/videos/search?q=how+to+configure+bluetooth+hc+05+with+arduino&view=detail&mid=245337404BE2DB9FFA72245337404BE2DB9FFA72&&FORM=VDRVRV>

Starting to use:

1. Connect the Arduino to the protoboard or the HC-05 .
2. Connect the Arduino to the USB port of your computer.
3. Open the serial monitor, configuring its speed to 9600 baud
4. Follow the instructions as they appear on the serial monitor, entering the parameters in the input box on the upper part of the serial monitor.
 - To configure the speed of the Bluetooth module: AT+UART=57600,0,0
 - To configure the name of the Bluetooth module: AT+NAME=ARDUINO**