

Arduino WiFi and Bluetooth Workshop

Written by Mark Webster and ??

Summary:

Learn to connect WiFi and Bluetooth modules to an Arduino microcontroller.

Introduction

Bluetooth and WiFi are common short range communication technologies. Robots and IoT devices can connect via bluetooth and wifi with a remote controller.

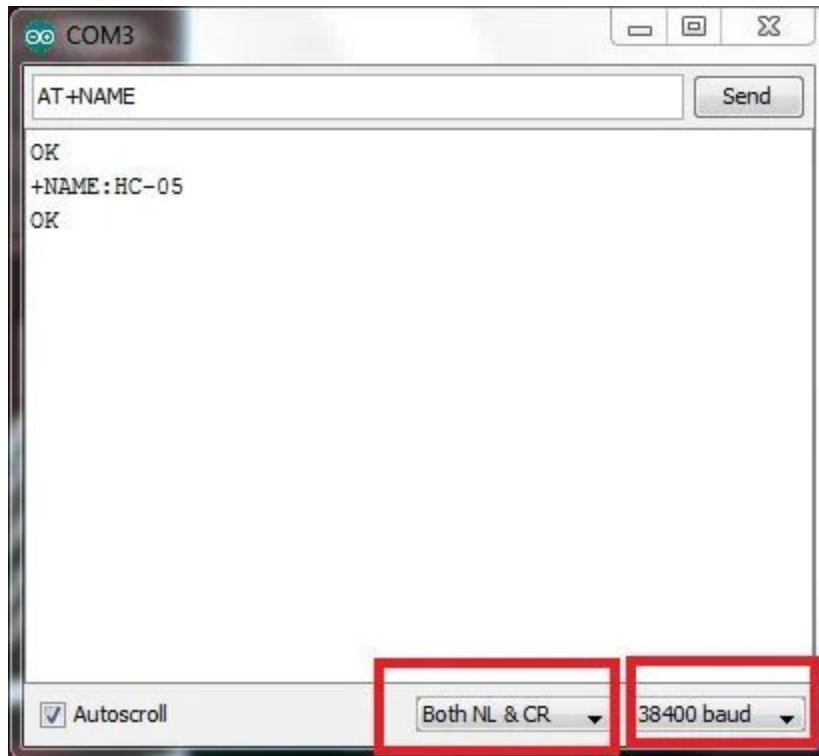
AT Commands

Some technologies never die, they just merge into the innards of new technologies. The AT commands were developed for the Hayes 300 baud Smartmodem in 1981. The AT stands for "Attention ". These commands were used later by GSM phones. Some of the AT commands are also built into bluetooth and WiFi modules used by Arduinos.

There are Command codes and Data codes. Multiple commands can be put on one line separated by the "&" symbol. Typically one device is in master mode and the other device is in slave mode.

[https://www.itead.cc/wiki/Serial_Port_Bluetooth_Module_\(Master/Slave\)_:_HC-05#Download](https://www.itead.cc/wiki/Serial_Port_Bluetooth_Module_(Master/Slave)_:_HC-05#Download)

Set the baud rate to the default 38400 baud and the line feed to NL & CR (new line and carriage return).



1. Test command:

Command	Respond	Parameter
AT	OK	-

2. Reset

Command	Respond	Parameter
AT+RESET	OK	-

3. Get firmware version

Command	Respond	Parameter
AT+VERSION?	+VERSION:<Param> OK	Param : firmware version

Example:

AT+VERSION?\r\n

+VERSION:2.0-20100601

OK

The default state is:

Default state:

Slave mode, pin code :1234, device name: H-C-2010-06-01 ,Baud 38400bits/s.

A few commands are:

AT : Ceck the connection.

AT+NAME : See default name

AT+ADDR : see default address

AT+VERSION : See version

AT+UART : See baudrate

AT+ROLE: See role of bt module(1=master/0=slave)

AT+RESET : Reset and exit AT mode

AT+ORGL : Restore factory settings

AT+PSWD: see default password

Bluetooth

Bluetooth is available on almost all smartphones and can be used to communicate with an Arduino. A bluetooth module or shield is required such as the common HC-05. This is a 3.3V board so ideally a resistor divider is needed going from the Arduino to the HC-05.

Here is an example project that creates an Arduino bluetooth controller.

<https://create.arduino.cc/projecthub/mayooghgirish/arduino-bluetooth-basic-tutorial-d8b737>

There is an Android App which supports controlling the Arduino via the HC-05 bluetooth module. Pair Android App with HC 05/06 Bluetooth module 1) Turn ON HC 05/06 Bluetooth module 2) Scan for available device 3) Pair to HC 05/06 by entering default password 1234 OR 0000.

Example 1: Bluetooth

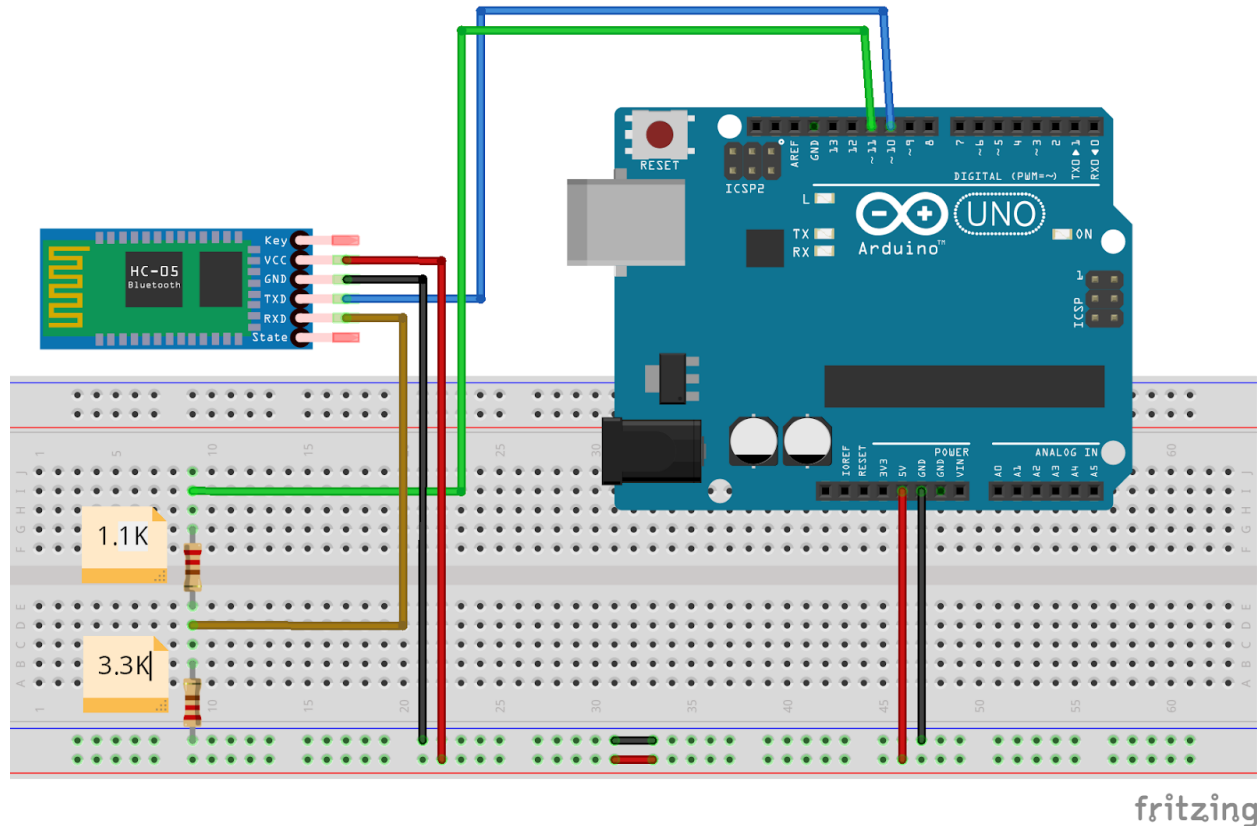
Hardware: HC 05 bluetooth module, 1K and 2K resistors (or 1.1 and 3.3K), wires

Software: Arduino IDE, Android Bluetooth terminal app.

Connect the Arduino to the HC-05 bluetooth module. Install the Android bluetooth terminal app. Turn on and off the LED on the Arduino.

The HC-05 is a Bluetooth SPP board so it communicates using the serial port protocol. This involves AT commands.

Note: the HC-05 module is 3.3V and the Arduino digital pins are 5V. The TX pin on the Arduino must use a voltage divider (1K over 2K) to avoid burning out the HC-05 module. The 3.3V output of the HC-05 can be read as a 1 on the Arduino so no level conversion is needed.



(replace)

The first example just sends strings between the Arduino and the cell phone terminal app.

On the Arduino just use the software serial library on pins 10,11. This example uses the Serial Monitor in the IDE to send and receive strings.

```

/* Bluetooth Example in Wifi-Bluetooth workshop
 * Reads text from Serial Monitor and send to Terminal app on
phone
 * and from phone to Serial Monitor
 * First connect phone to HC-05 with password 1234
 * Default COMM mode is 9600, Default AT mode is 38400
 *
 */

#include <SoftwareSerial.h>

```

```

SoftwareSerial mySerial(10, 11); //10= RX, 11=TX

void setup() {

    // set the data rate for the SoftwareSerial port
    mySerial.begin(9600); // Default communication rate of the
Bluetooth module
    Serial.begin(9600);
    Serial.println("Enter text to send to Terminal app");
}

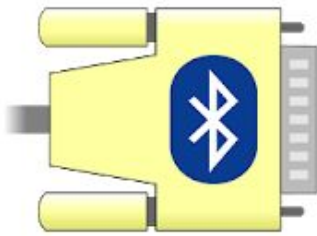
void loop() {
    int state = 0;
    int serialByte = 0;

    // Read byte from bluetooth connection
    if (mySerial.available()) {
        state = mySerial.read();
        Serial.write(state);
    }
    // Read byte from Serial Monitor
    if (Serial.available()) {
        serialByte = Serial.read();
        mySerial.write(serialByte);
    }
}

```

There are many bluetooth apps for Android that can be used with the Arduino. Here is one:

The Android bluetooth serial terminal app
Serial Bluetooth Terminal



Serial Bluetooth Terminal

Kai Morich Tools

★★★★★ 688

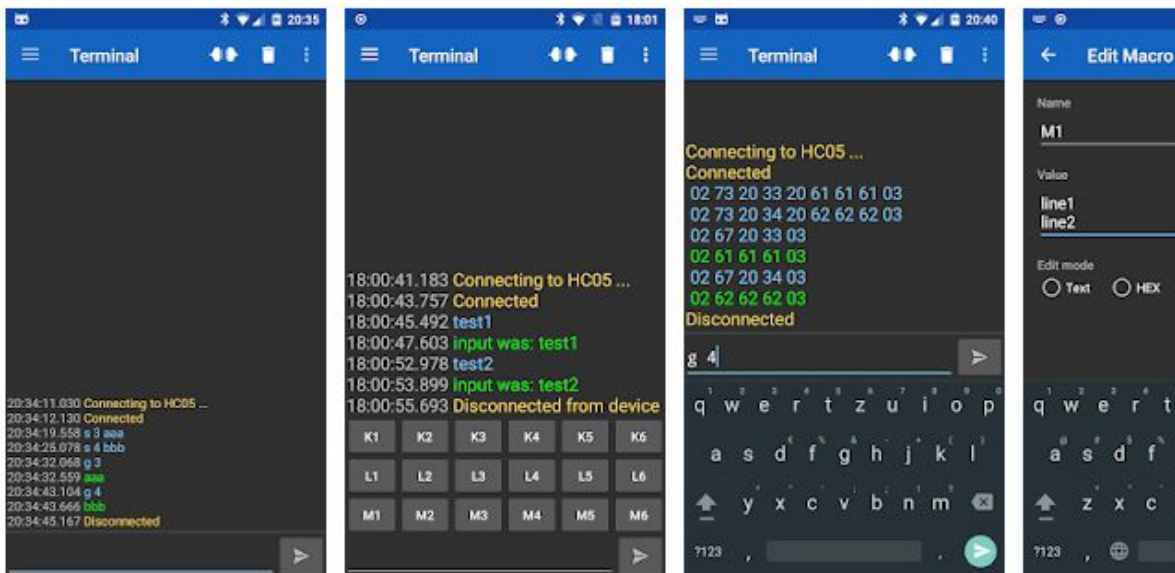
Everyone

Offers in-app purchases

This app is compatible with all of your devices.

You can share this with your family. [Learn more about Family Library](#)

Installed



To control hardware, have the Arduino code test for certain codes like “H” or “1” to turn on an LED, or “L” to turn a robot left.

WiFi

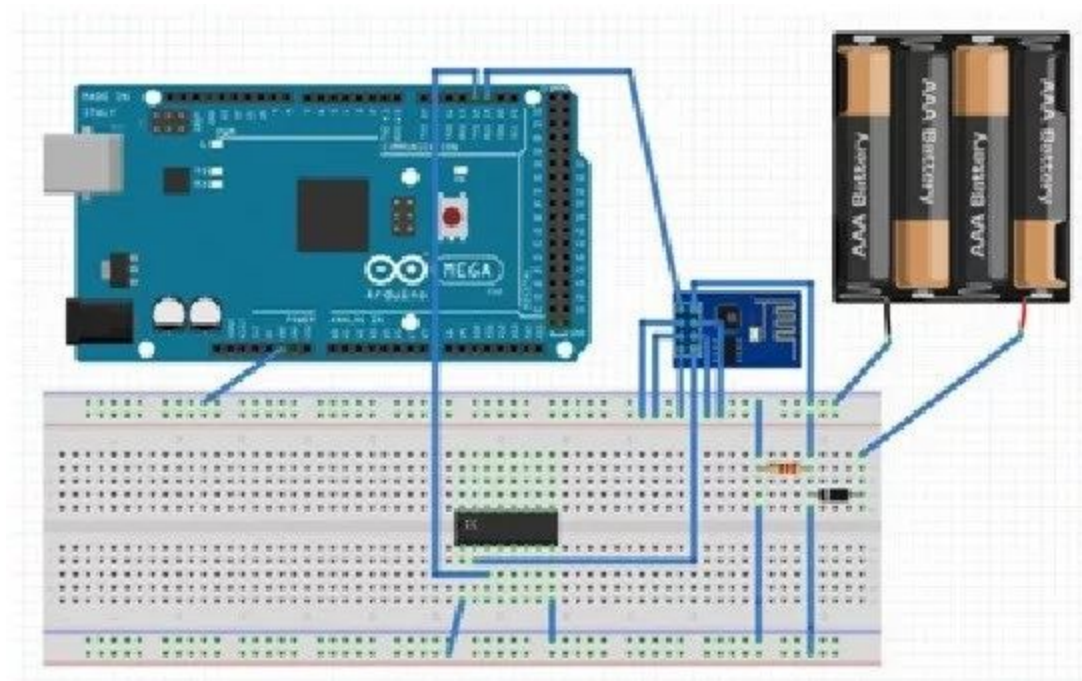
WiFi is a 2.4 ghz protocol (although 5 ghz also exists) that allow devices to connect to an intranet or internet through a WiFi router. A rapidly growing field is to connect microcontrollers to other computers to create an IoT (Internet of Things). This allows sensors to upload data to a distant web server, or web enabled appliances to be controlled remotely.

Most Arduinos do not have onboard Wifi, except the Arduino Yun. There are WiFi shields for the Arduino Uno, but more commonly the way is a ESP8266 WiFi module. It's odd, but the least expensive way for an Arduino to contact the Internet is with another microcontroller chip, the ESP8266.



(replace)

The ESP8266 wifi module communicates with AT commands over a serial connection. Two pins of the Arduino are assigned as serial RX and TX, and the AT command is sent to the wifi module, and a character string response is sent back. An IC is used to convert voltages from 5V to 3.3V. Some people use a resistor divider on the Arduino TX instead of an IC chip.



(replace)

The ESP8266 module can be used directly with a laptop computer via a USB to TTL converter. Then any terminal emulation program (like PuTTY) can be used to talk to the module. Here's some details on the AT commands with the ESP8266 wifi module directly.

<https://alselectro.wordpress.com/2015/05/05/wifi-module-esp8266-1-getting-started-with-at-commands/>

Example 2: ESP8266 Wifi Module

Connect an ESP8266 wifi module to an Arduino Uno. Baud rate is usually 57600. Send an AT command to the module and echo the response (which should be OK) to the Serial Monitor. Enter AT+GMR to see the firmware version.