SCMU 2018/2019

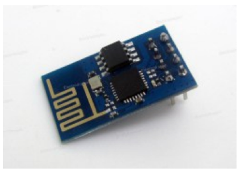
ARDUINO AND WIFI COMMUNICATION (LAB 4)

WiFi - ESP8266

- ESP8266 is a WiFi microcontroller with a full WiFi front-end (both as client and access point) and TCP/IP stack with DNS support as well.
- Very popular.
- Most of the low cost modules are not breadboard friendly, don't have an onboard 500mA 3.3V regulator or level shifting

WiFi – ESP8266 module

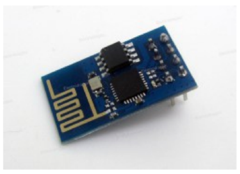
- Computational power onboard.
- Has a full TCP/IP stack protocol.
- Can be easily configured as a web server.



- Accepts commands via simple serial interface, it then responds back with the operation's outcome (assuming everything is running correctly).
- Once the device is connected and is set to accept connections, it will send unsolicited messages whenever a new connection or a new request is issued.

WiFi – ESP8266 module

- Can use this board as a simple WiFi connection board.
- Or can exploit its full power by implementing the logic inside the board itself.
  - Manage sensors and elaborate autonomously their signals and measurements.
  - Hosting simple applications on board can let you make a very compact IoT solution.



WiFi – ESP8266 module

Connections between Arduino and ESP8266:

- ESP-Rx goes to Arduino Tx,
- ESP-Tx goes to Arduino Rx.

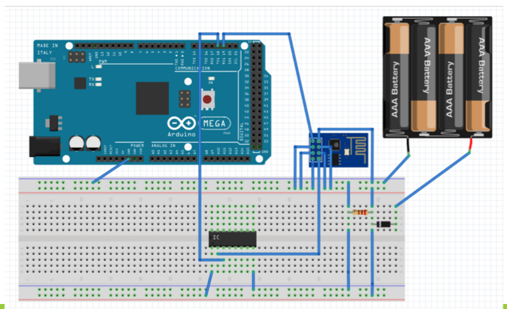
But ESP8266 run on 3.3V, and Arduino pins run on 5V, so before connecting them, you must provide a way to adapt these voltages, or you could damage your ESP8266.

WiFi – ESP8266 module

Use CD4050 to adjust the voltage

CD4050 only adapts Arduino-Tx to ESP-Rx, there is no need to adjust the ESP-Tx to Arduino-Rx, since the Arduino port can handle 3.3V

When using an external power, do not forget of interconnecting both Arduino-GND and power supply GND, to keep the voltage reference. If you don't do that, you'll end up with a lot of transmission errors.



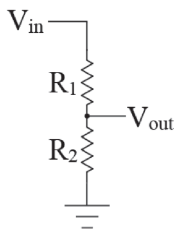
WiFi – ESP8266 module

You can also use a voltage divider circuit to divide 5V source from Arduino to 3.3v source for ESP8266.

Vin = 5V

Vout = 3.3V

Vout = (R2/(R1+R2))* Vin



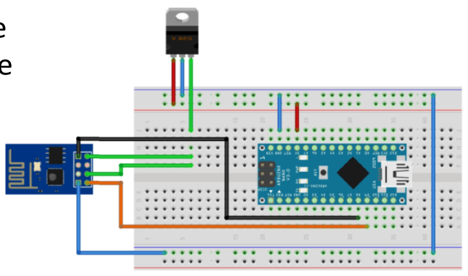
URL for voltage divider calculation: https://www.abelectronics.co.uk/tools/resistor-voltage-divider.aspx

WiFi – ESP8266 module

Use LM3940 Low-Dropout Regulator for 5V to 3.3V

This is because Arduino 3V3 pin may not be able to provide the necessary current to power the ESP8266.

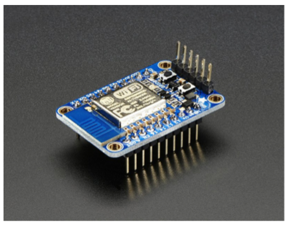
You can also use an external power



WiFi - HUZZAH ESP8266

Certified module with an onboard antenna.
It also has:

- Reset button
- User button that can also put the chip into bootloading mode
- Red LED you can blink
- Level shifting on the UART* and reset pin
- 3.3V out, 500mA regulator
- Two diode-protected power inputs (one for a USB cable, another for a battery)



https://learn.adafruit.com/adafruit-huzzah-esp8266-breakout/overview

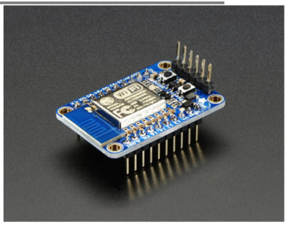
UART - Universal Asynchronous Receiver/Transmitter

- Chip with programming that controls a computer's interface to its attached serial devices.
- It provides a RS-232C interface so that it can "talk" to and exchange data with modems and other serial devices.
- As part of this interface, the UART also:
  - Converts the bytes it receives from the computer along parallel circuits into a single serial bit stream for outbound transmission
  - On inbound transmission, converts the serial bit stream into the bytes that the computer handles
  - Adds a parity bit (if it's been selected) on outbound transmissions and checks the parity of incoming bytes (if selected) and discards the parity bit
  - Adds start and stop delineators on outbound and strips them from inbound transmissions
  - Handles interrupt and device management that require coordinating the computer's speed of operation with device speeds

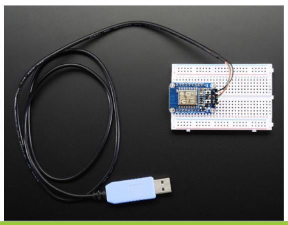
WiFi - HUZZAH ESP8266

Two parallel breakouts on either side give you access to:

- 1 x Analog input (1.8V max)
- 9 x GPIO (3.3V logic)
- 2 x UART pins
- 2 x 3-6V power inputs, reset, enable, LDO-disable, 3.3V output



At the end it has a pinout where you can plug a console cable to upload software and read/write debugging information via the UART.
This enables you to directly program this module at the end remove the cable, and this little module can be embedded into your project box.



WiFi - HUZZAH ESP8266

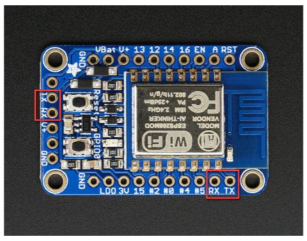
Serial Pins

RX and TX are the serial control and bootloading pins, and are how you will spend most of your time communicating with the ESP module.

The TX pin is the output from the module and is 3.3V logic.

The RX pin is the input into the module and is 5V compliant (there is a level shifter on this pin)

The pins are available in two places, one set is on the right side breakout. The same pins are also at the bottom on the "FTDI/Serial" breakout

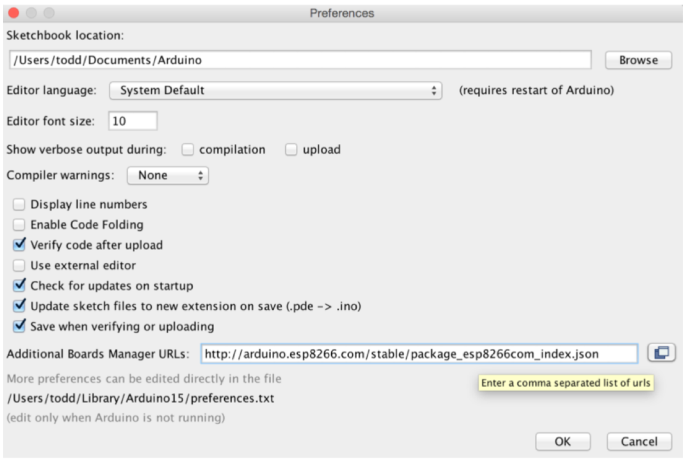


WiFi - HUZZAH ESP8266

Install the ESP8266 Board Package

1. Open Arduino -> Preferences window
2. Enter the following URL into Additional Board Manager field
   http://arduino.esp8266.com/stable/package_esp8266com_index.json
3. Go to Tools -> Board -> Board Manager to install the ESP8266 package.

WiFi - HUZZAH ESP8266



Sketchbook location: /Users/todd/Documents/Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 10

Show verbose output during: ☐ compilation ☐ upload

Compiler warnings: None

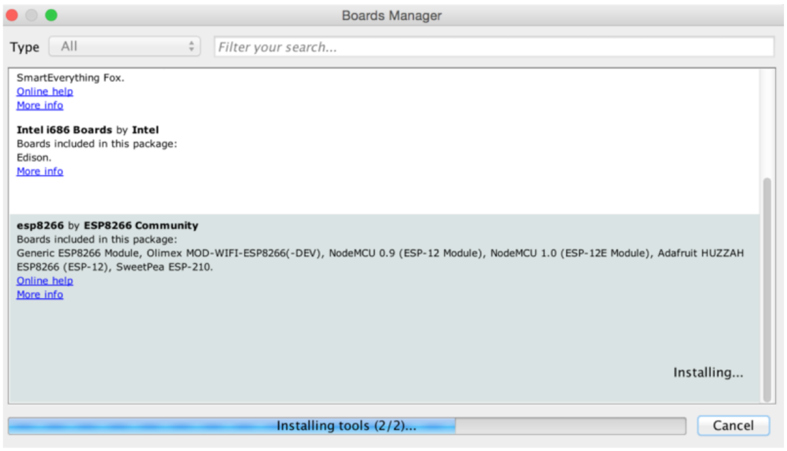
☐ Display line numbers
☐ Enable Code Folding
☑ Verify code after upload
☐ Use external editor
☑ Check for updates on startup
☑ Update sketch files to new extension on save (.pde -> .ino)
☑ Save when verifying or uploading

Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json

Enter a comma separated list of urls

More preferences can be edited directly in the file
/Users/todd/Library/Arduino15/preferences.txt
(edit only when Arduino is not running)

WiFi - HUZZAH ESP8266



SmartEverything Fox.
Online help
More info

Intel i686 Boards by **Intel**
Boards included in this package:
Edison.
More info

esp8266 by **ESP8266 Community**
Boards included in this package:
Generic ESP8266 Module, Olimex MOD-WIFI-ESP8266(-DEV), NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Adafruit HUZZAH ESP8266 (ESP-12), SweetPea ESP-210.
Online help
More info

Installing...

Installing tools (2/2)...

WiFi - HUZZAH ESP8266

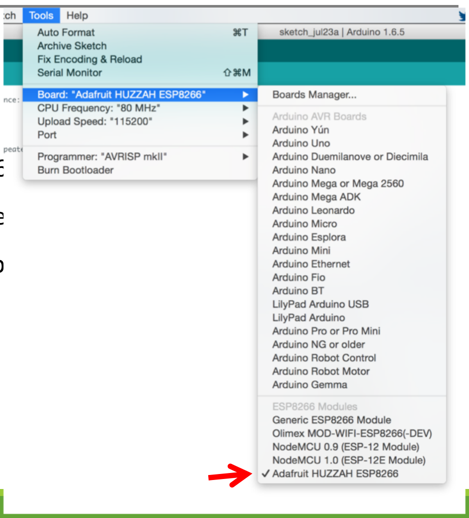
Setup ESP8266 Support

1. Restart the Arduino IDE
2. Select Adafruit HUZZAH ESP8266 from Tools->Board dropdown
3. Select 80 MHz as the CPU frequency
4. Select 115200 as baud upload speed

WiFi - HUZZAH ESP8266

Setup ESP8266 Support

1. Restart the Arduino IDE
2. Select Adafruit HUZZAH ESP826
3. Select 80 MHz as the CPU freque
4. Select 115200 as baud upload sp


sketch_jul23a | Arduino 1.6.5

Board: "Adafruit HUZZAH ESP8266"
CPU Frequency: "80 MHz"
Upload Speed: "115200"
Programmer: "AVRISP mkII"

✓ Adafruit HUZZAH ESP8266

WiFi - HUZZAH ESP8266

Blink test

Sketch code

```
void setup() {
  pinMode(0, OUTPUT);
}

void loop() {
  digitalWrite(0, HIGH);
  delay(500);
  digitalWrite(0, LOW);
  delay(500);
}
```

WiFi - HUZZAH ESP8266

Blink test (cont.)

Put board into bootload mode (have to do this before each upload)

1. Hold down the GPIO0 button, the red LED will be lit
2. While holding down GPIO0, click the RESET button
3. Release RESET, then release GPIO0
4. When you release the RESET button, the red LED will be lit dimly, this means its ready to bootload

When in bootload, upload the “blink” sketch via the IDE

The LED will start blinking

WiFi - HUZZAH ESP8266

Blink test (cont.)

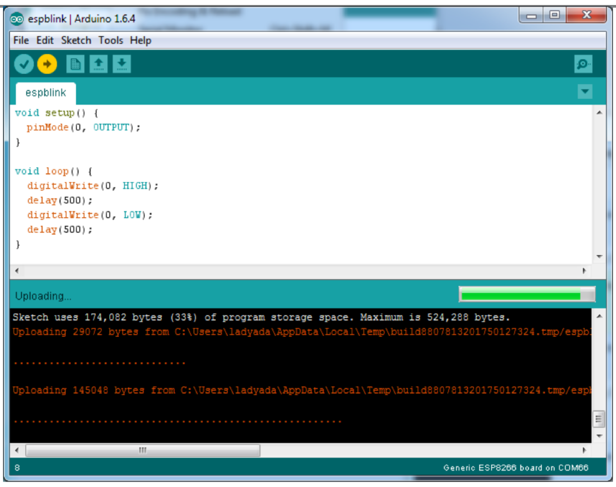
Put board into bootload mode (have to do this before each upload)

1. Hold down the GPIO0 button, the red LED will be lit
2. While holding down GPIO0, click the RESET button
3. Release RESET, then release GPIO0
4. When you release the RESET button, the red LED will be lit dimly, this means its ready to bootload

When in bootload, upload the “blink” sketch via the IDE

The LED will start blinking

WiFi - HUZZAH ESP8266



```
void setup() {
  pinMode(0, OUTPUT);
}

void loop() {
  digitalWrite(0, HIGH);
  delay(500);
  digitalWrite(0, LOW);
  delay(500);
}
```

```
Sketch uses 174,082 bytes (33%) of program storage space. Maximum is 524,288 bytes.
Uploading 29072 bytes from C:\Users\ladyada\AppData\Local\Temp\build8807813201750127324.tmp/espbl

.............................

Uploading 145048 bytes from C:\Users\ladyada\AppData\Local\Temp\build8807813201750127324.tmp/esph

........................................................
```

WiFi - Arduino library

https://www.arduino.cc/en/Reference/WiFi

https://github.com/esp8266/Arduino

http://www.arduinesp.com/wifiwebserver

https://learn.adafruit.com/adafruit-huzzah-esp8266-breakout/overview

Pull-up Resistor

Pull-up resistors are very common when using microcontrollers (MCUs) or any digital logic device.

Why?

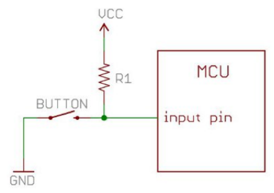
If there is nothing connected to the pin and your program reads the state of the pin, will it be high (pulled to VCC) or low (pulled to ground)?

This phenomena is referred to as *floating*.

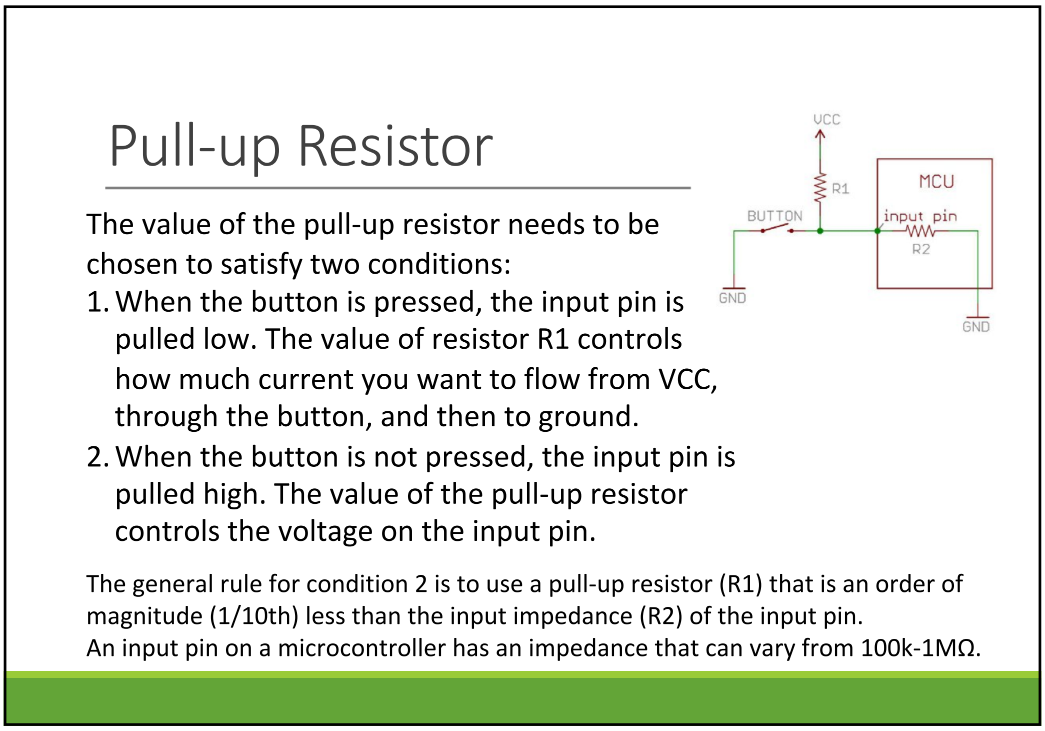
To prevent this unknown state, a pull-up or pull-down resistor will ensure that the pin is in either a high or low state.

Pull-up Resistor

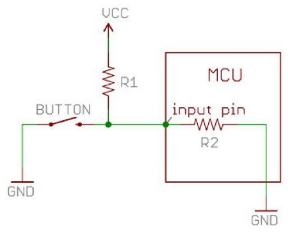
1. With a pull-up resistor, the input pin will read a high state when the button is not pressed. In other words, a small amount of current is flowing between VCC and the input pin (not to ground), thus the input pin reads close to VCC.
2. When the button is pressed, it connects the input pin directly to ground. The current flows through the resistor to ground, thus the input pin reads a low state.



Attention: if the resistor wasn't there, your button would connect VCC to ground, which is very bad and is also known as a short.



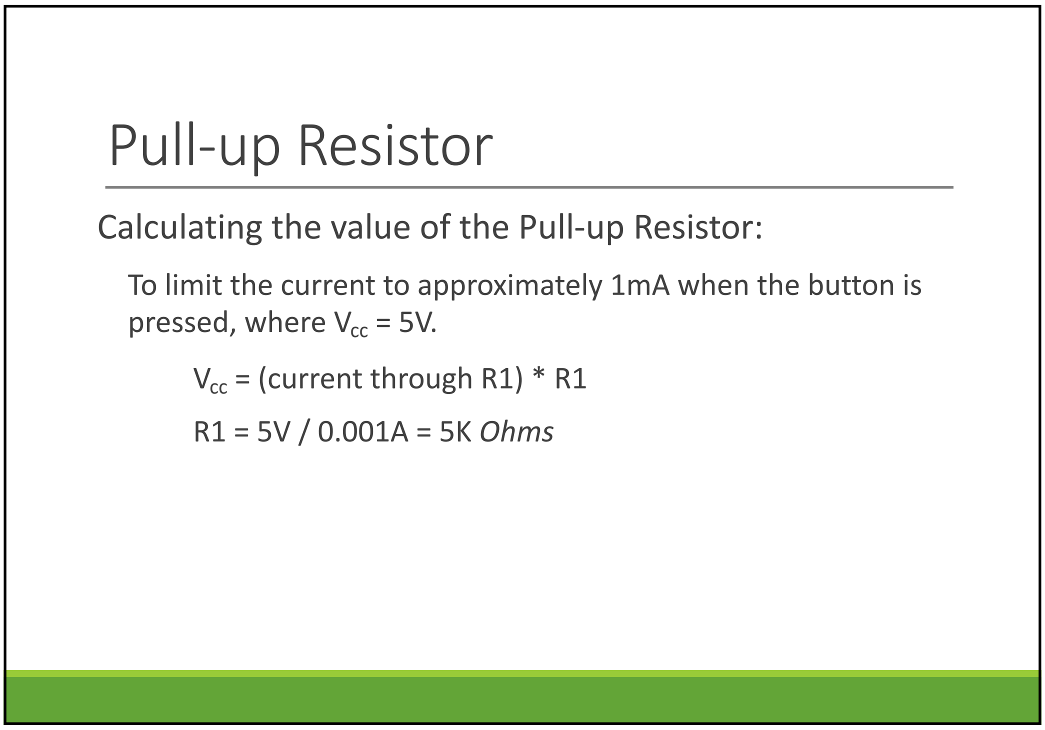
Pull-up Resistor



The value of the pull-up resistor needs to be chosen to satisfy two conditions:

1. When the button is pressed, the input pin is pulled low. The value of resistor R1 controls how much current you want to flow from VCC, through the button, and then to ground.
2. When the button is not pressed, the input pin is pulled high. The value of the pull-up resistor controls the voltage on the input pin.

The general rule for condition 2 is to use a pull-up resistor (R1) that is an order of magnitude (1/10th) less than the input impedance (R2) of the input pin.
An input pin on a microcontroller has an impedance that can vary from 100k-1MΩ.



Pull-up Resistor

Calculating the value of the Pull-up Resistor:

To limit the current to approximately 1mA when the button is pressed, where V_{cc} = 5V.

V_{cc} = (current through R1) * R1

R1 = 5V / 0.001A = 5K *Ohms*