SparkFun ESP8266 Thing
Development Workshop

Agus Kurniawan

Copyright

SparkFun ESP8266 Thing Development Workshop

Agus Kurniawan

1st Edition, 2015

Table of Contents

Preface

This book was written to help anyone want to get started with SparkFun ESP8266 Thing board development. It describes the basic elements of SparkFun ESP8266 Thing development using Arduino software.

Agus Kurniawan

Depok, August 2015

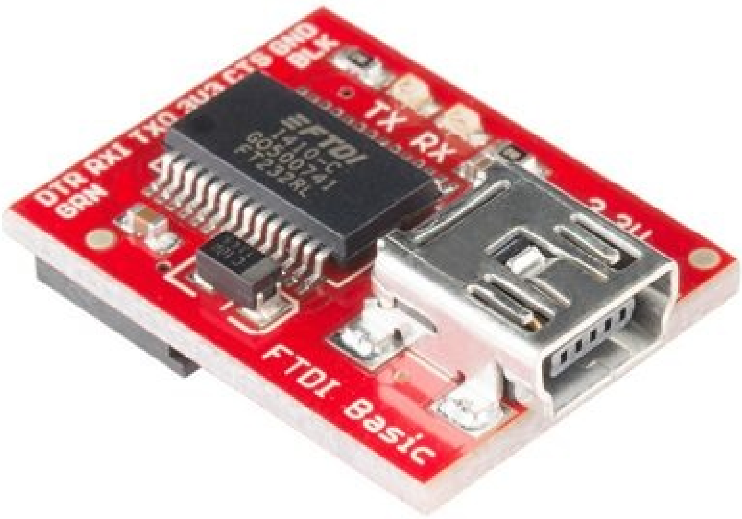
1. Preparing Development Environment

1.1 SparkFun ESP8266 Thing

The SparkFun ESP8266 Thing is a breakout and development board for the ESP8266 WiFi SoC – a leading platform for Internet of Things (IoT) or WiFi-related projects, and designed by SparkFun. The following is a form of The SparkFun ESP8266 Thing.

Officially you can buy this board on https://www.sparkfun.com/products/13231. You also can buy this product on your local electronic store.

To develop a program for SpakFun ESP8266 Thing target, you need UART/Serial USB with UART power output 3.3V. I recommended to use FTDI USB, for instance, https://www.sparkfun.com/products/9873 . In this book, I used Foca serial adapter from ITEAD Studio, http://imall.itead.cc/foca.html .



1.2 Electronics Components

We need electronic components to build our testing, for instance, Resistor, LED, sensor devices and etc. I recommend you can buy electronic component kit. We can use electronics kit from Arduino to be developed on SparkFun ESP8266 Thing. The following is a list of electronics kit which can be used in our case.

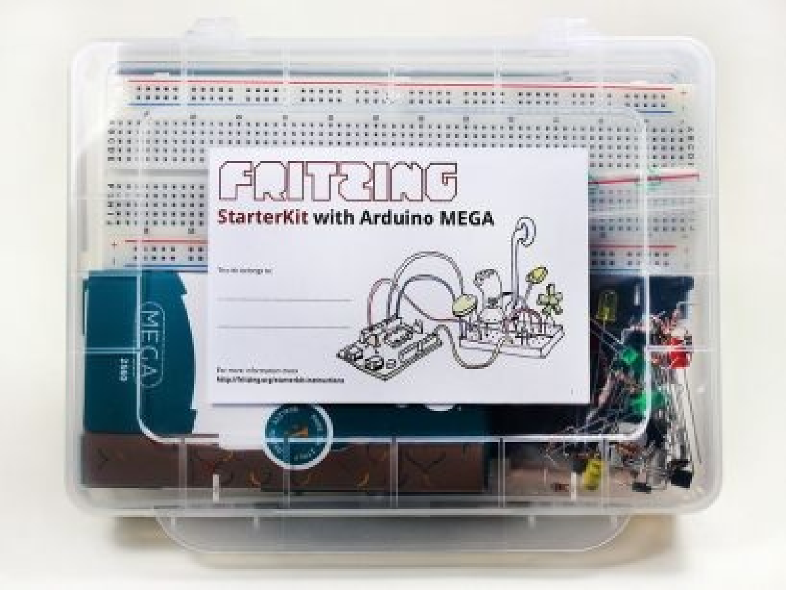
1.2.1 Arduino Starter Kit

Store website: http://arduino.cc/en/Main/ArduinoStarterKit

1.2.2 Fritzing

Store website: http://shop.fritzing.org/ .

You can buy Fritzing Starter Kit with Arduino UNO or Fritzing Starter Kit with Arduino Mega.



1.2.3 Cooking-Hacks: Arduino Starter Kit

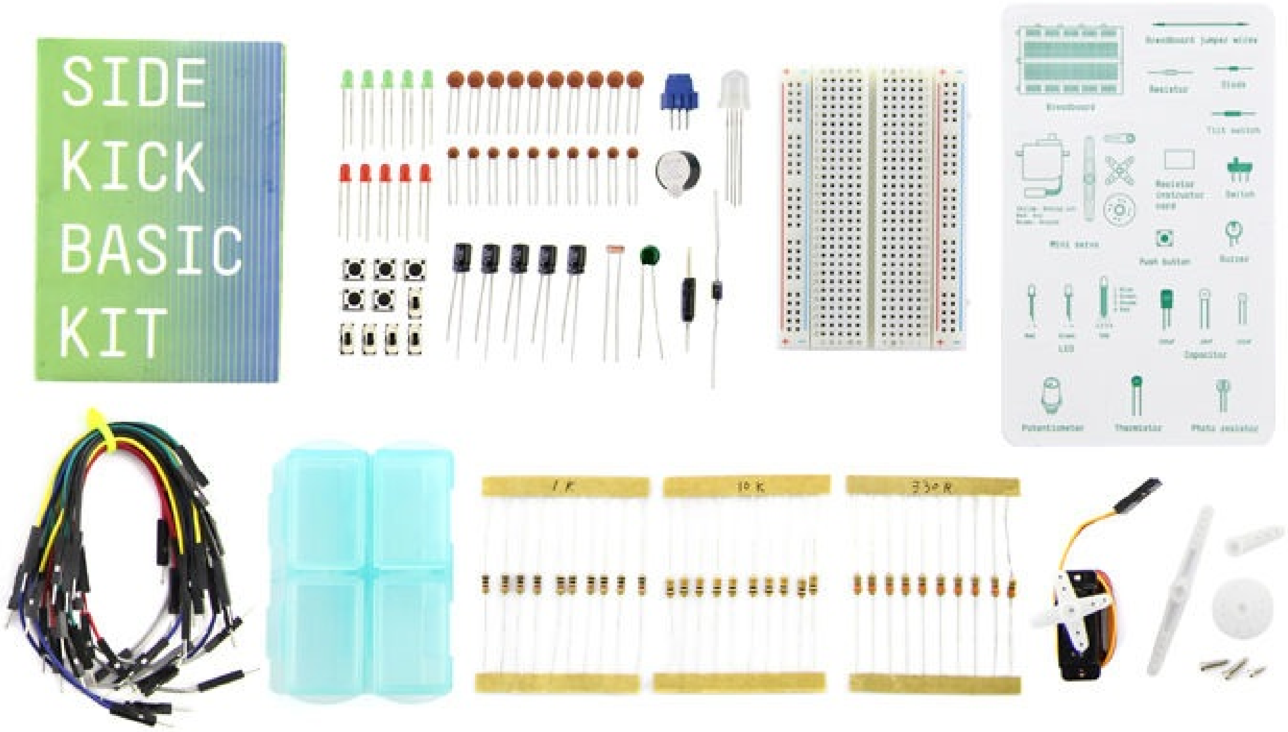
Store website: http://www.cooking-hacks.com/index.php/shop/arduino/starter-kits/arduino-starter-kit.html

1.2.4 Arduino Sidekick Basic kit v2

Store website: http://www.seeedstudio.com/depot/Sidekick-Basic-Kit-for-Arduino-V2-p-1858.html

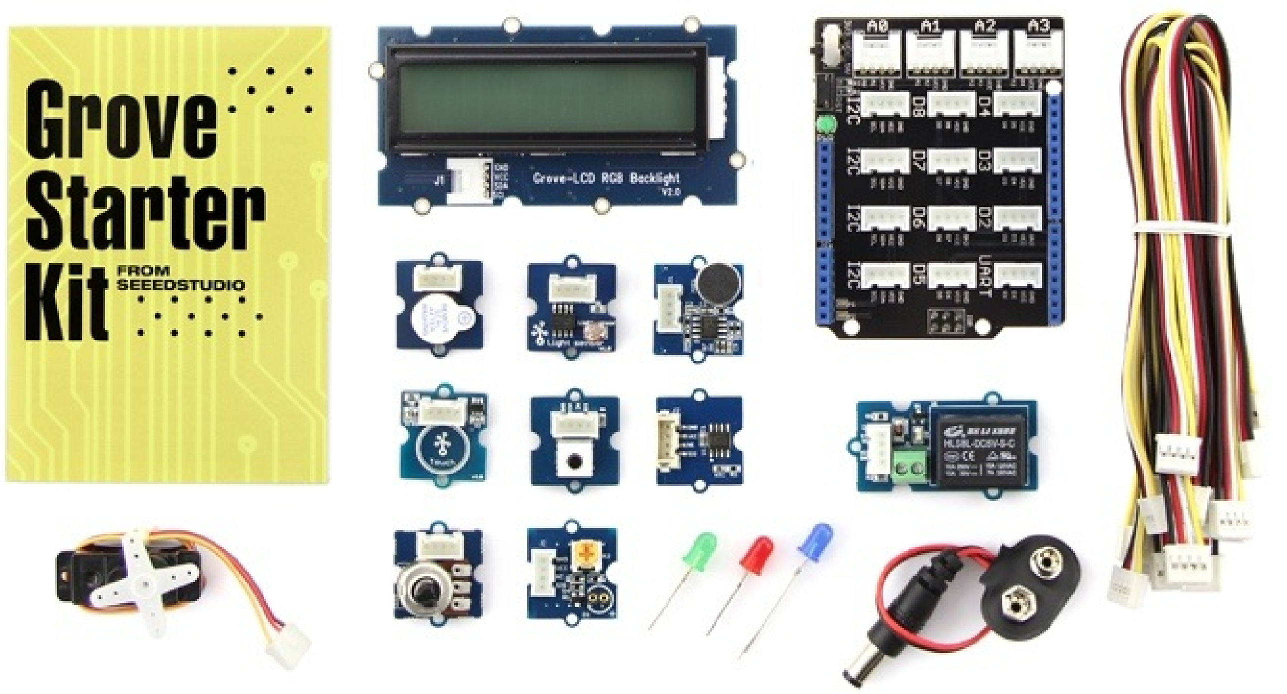
You also can find this kit on this online store.

http://www.exp-tech.de/seeed-studio-sidekick-basic-kit-for-arduino-v2



1.2.5 Grove - Starter Kit for Arduino

Another option, you can buy this kit on Seeedstudio, http://www.seeedstudio.com/depot/Grove-Starter-Kit-for-Arduino-p-1855.html .



1.2.6 DFRobot - Arduino Kit for Beginner v3

DFRobot provides Arduino kit too. You can buy it on the following website.

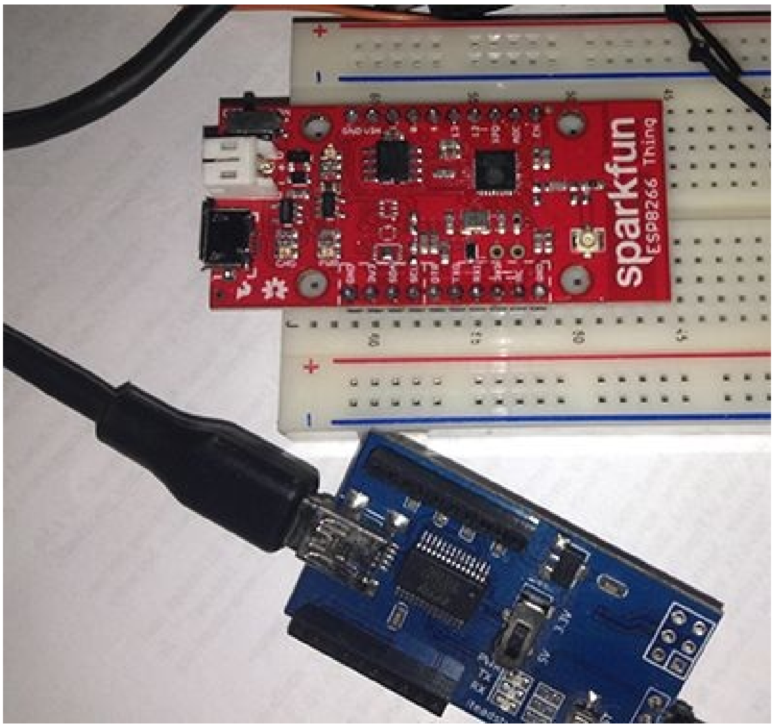
http://www.dfrobot.com/index.php?route=product/product&path=35_49&product_id=345

1.3 Development Tools

To develop app with SparkFun ESP8266 Thing target, I use Arduino IDE for text editor. You can learn how to install it on chapter 2.

1.4 Testing

For testing, I used SparkFun ESP8266 Thing on Windows 10, OS X and Ubuntu.



I also used Arduino Sidekick Basic kit for electronic components.

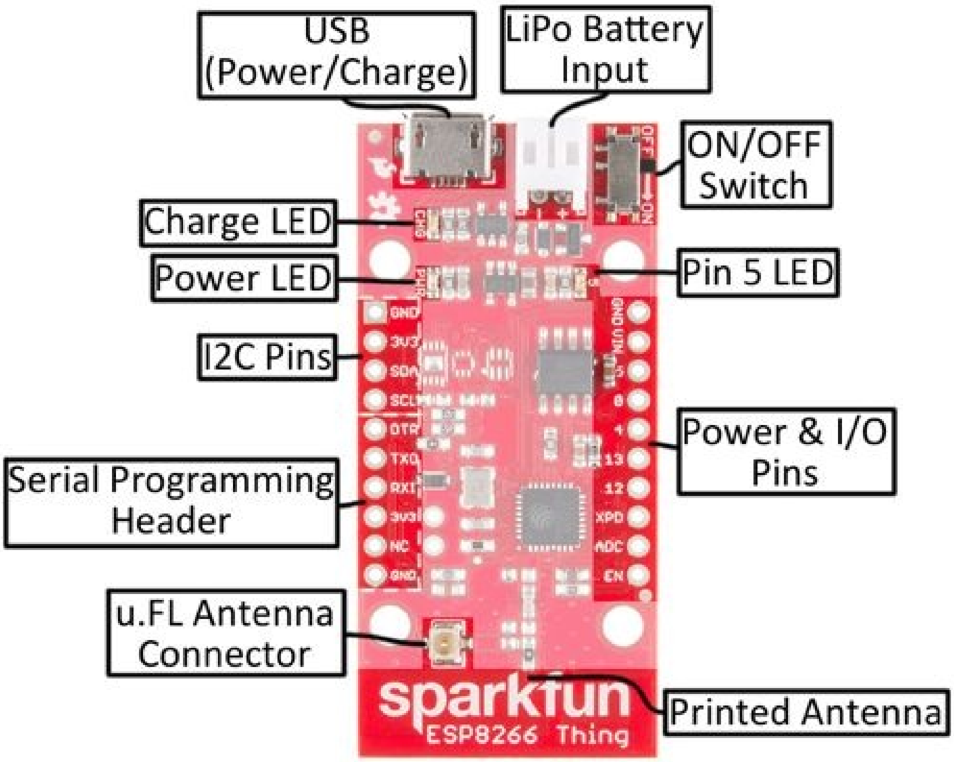
2. Setting Up SparkFun ESP8266 Thing

This chapter explains how to work on setting up SpakFun ESP8266 Thing board.

2.1 Getting Started

In this chapter, we learn how to get started with SpakFun ESP8266 Thing board. We try to build a simple app, Blink, on SpakFun ESP8266 Thing board. This chapter is based on https://learn.sparkfun.com/tutorials/esp8266-thing-hookup-guide/all .

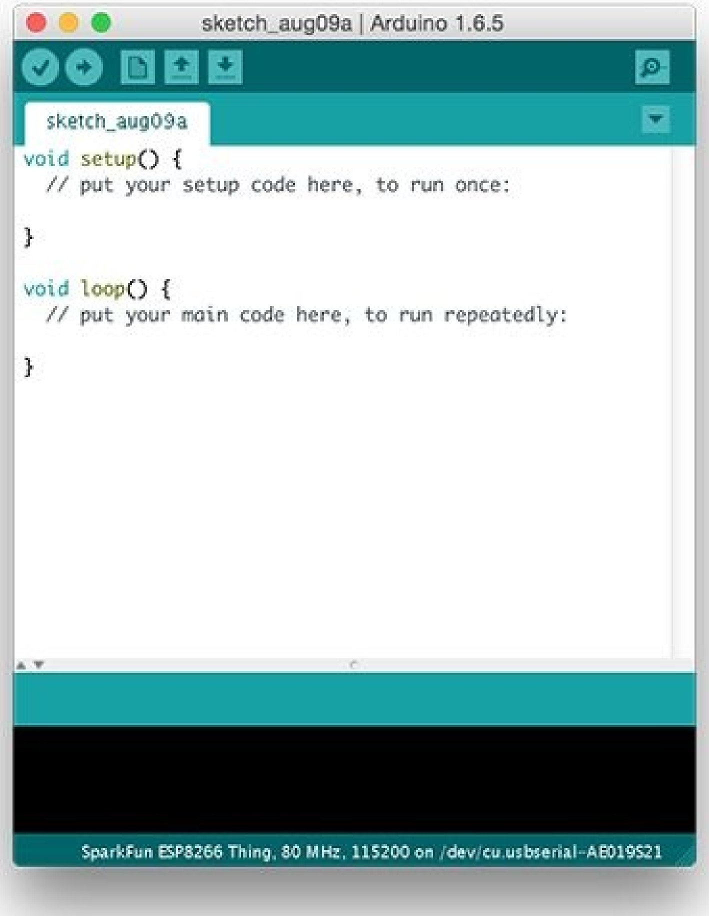
The following is a scheme of SpakFun ESP8266 Thing board.



2.2 Installing Arduino Software

If you have experience in Arduino development, you can use Arduino IDE to develop SpakFun ESP8266 Thing board too. You can use add-on on your Arduino using this library, https://github.com/esp8266/Arduino .

Open Arduino IDE.

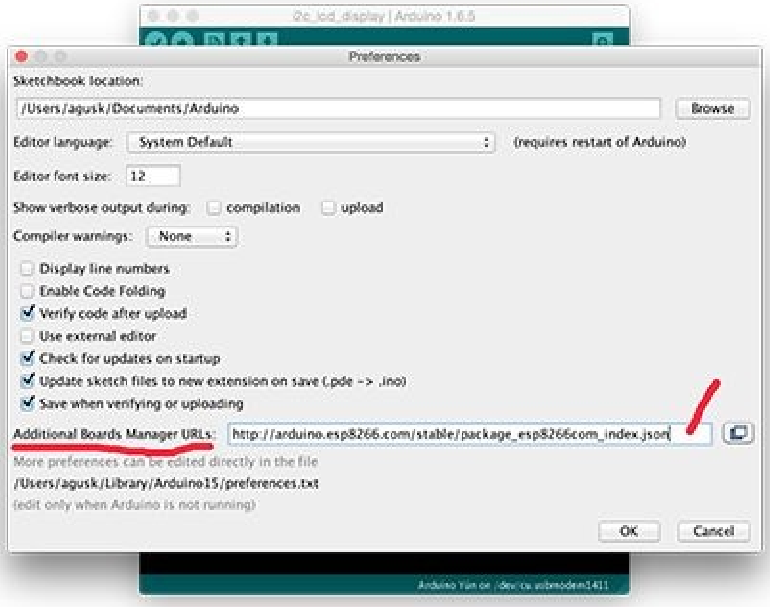


```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

SparkFun ESP8266 Thing, 80 MHz, 115200 on /dev/cu.usbserial-AE019S21

Then, click menu File -> Prefernces (Windows/Linux) or Arduino -> Preferences (OS X) so you should see Preferences dialog.

Add **http://arduino.esp8266.com/stable/package_esp8266com_index.json** on Additional Boards Manager URLs . If you have the list on there, just add ; for new list item,



Sketchbook location: /Users/agusk/Documents/Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

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

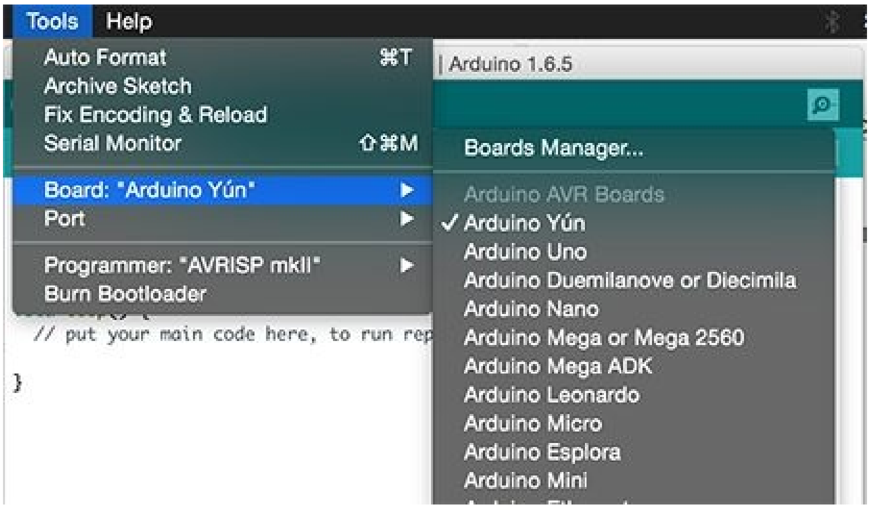
More preferences can be edited directly in the file

/Users/agusk/Library/Arduino15/preferences.txt

(edit only when Arduino is not running)

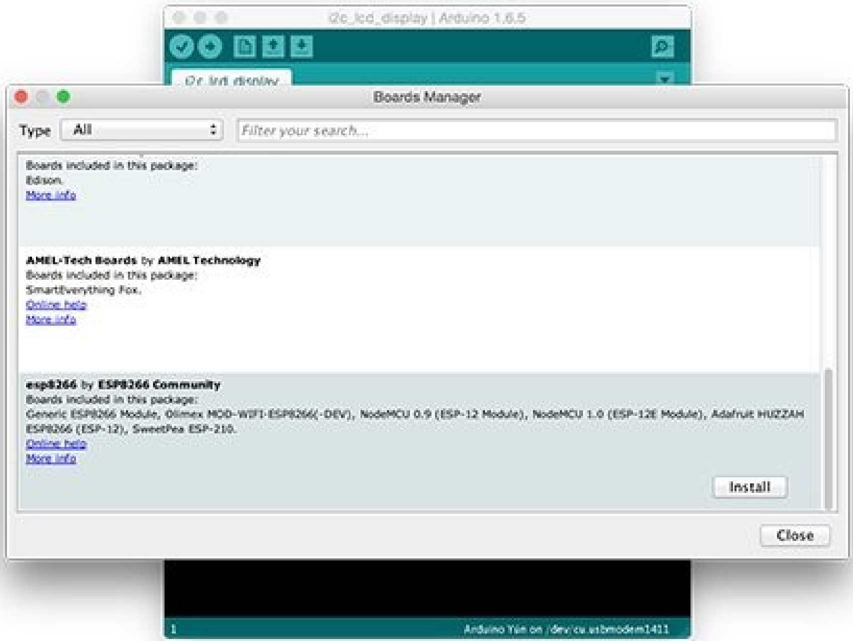
If done, click **OK** button.

Click menu Tools -> Board -> Boards Managers.



```
  // put your main code here, to run rep

}
```

Then, you should see Boards Manager dialog. Find esp8266 board and install it.

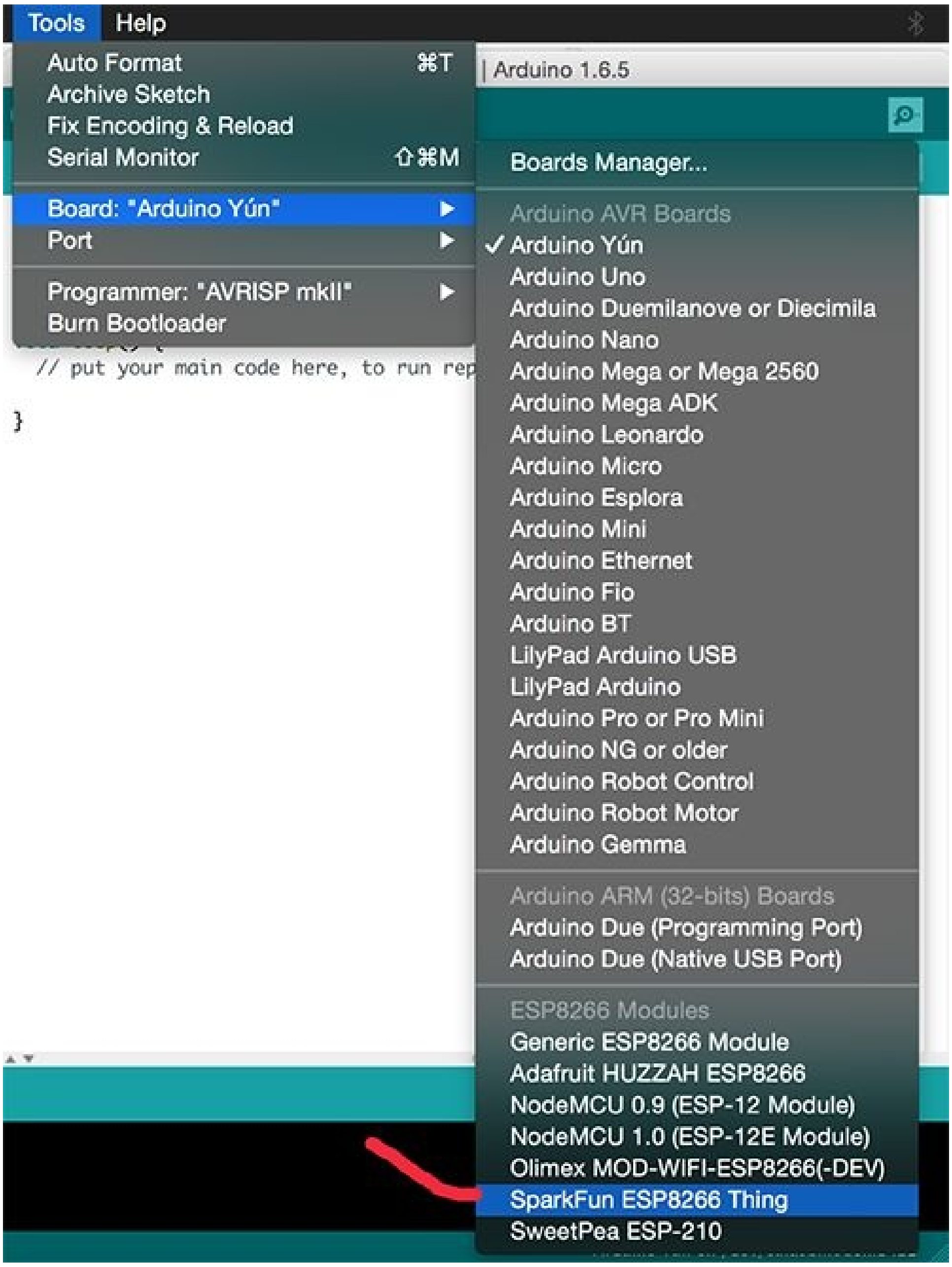


Boards included in this package:
Edison.
More info

AMEL-Tech Boards by **AMEL Technology**
Boards included in this package:
SmartEverything Fox.
Online help
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

After installed, you can see a list of ESP8266 Modules target on Arduino IDE such as SparkFun ESP8266 Thing.



```
  // put your main code here, to run rep
}
```

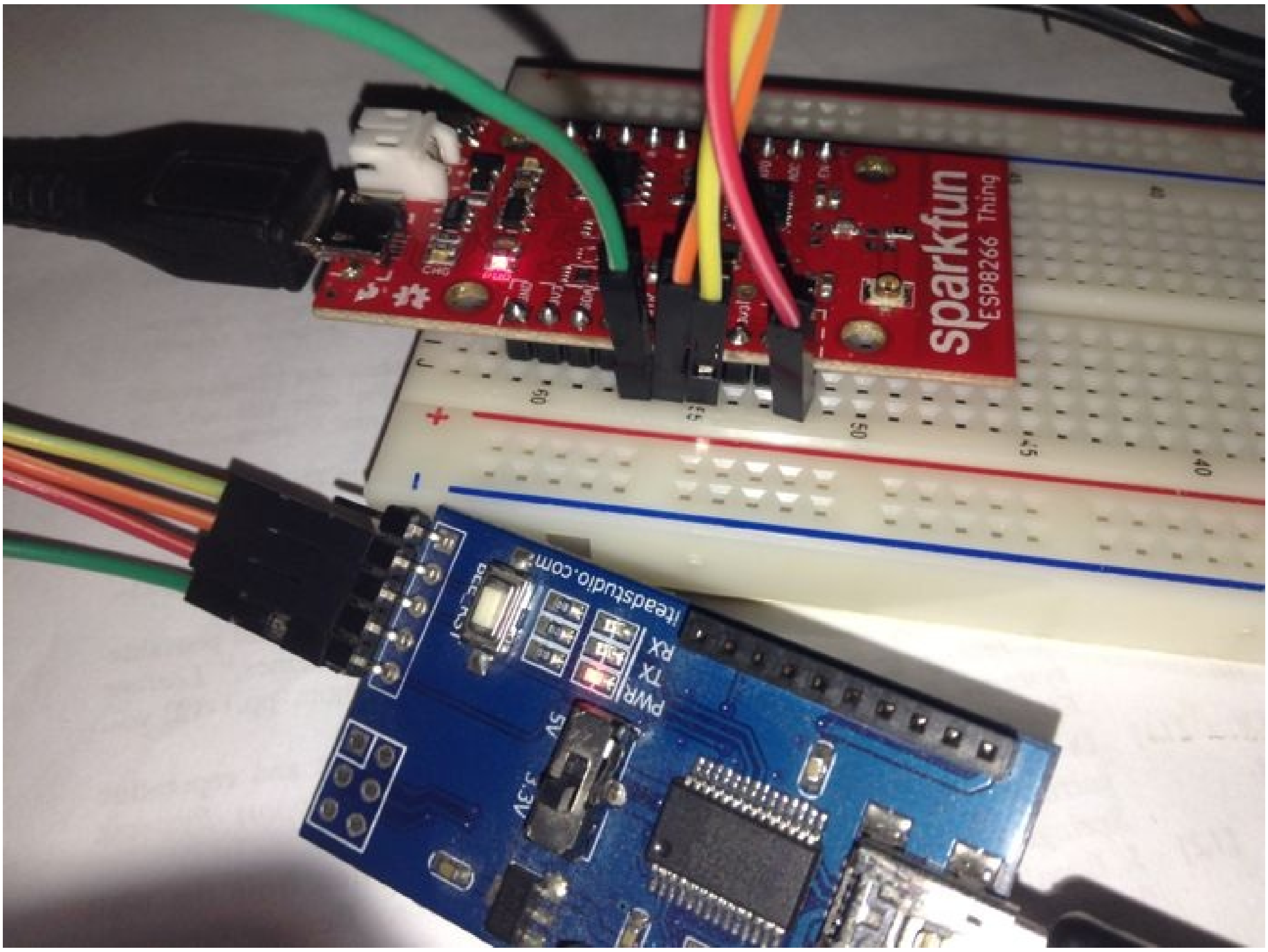
Now you can write the program for SparkFun ESP8266 Thing.

2.3 Connecting SparkFun ESP8266 Thing board to Computer

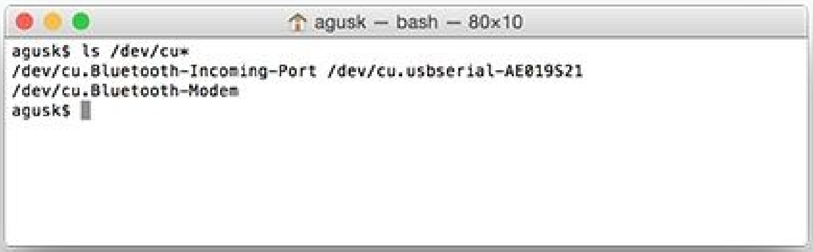
Firstly, we connect Serial adapter to SparkFun ESP8266 Thing board as follows:

- Serial adapter Tx is connected to Rx pin from SparkFun ESP8266 Thing
- Serial adapter Rx is connected to Tx pin from SparkFun ESP8266 Thing
- Serial adapter DTS is connected to DTS pin from SparkFun ESP8266 Thing
- Serial adapter GND is connected to GND pin from SparkFun ESP8266 Thing

Then, connect power adapter (maximum 5V DC) to SparkFun ESP8266 Thing power via microUSB. I recommended to connect it to your computer. Turn on the board. After connected, you may get lighting on blue LED. Now you can connect serial adapter to your computer.



After connected, you can see serial adapter as COMx on Windows, /dev/cu.usbserial* on OSX or /dev/ttyAMA* or /dev/tty* (check it on your Linux platform).



```
agusk$ ls /dev/cu*
/dev/cu.Bluetooth-Incoming-Port /dev/cu.usbserial-AE019521
/dev/cu.Bluetooth-Modem
agusk$
```

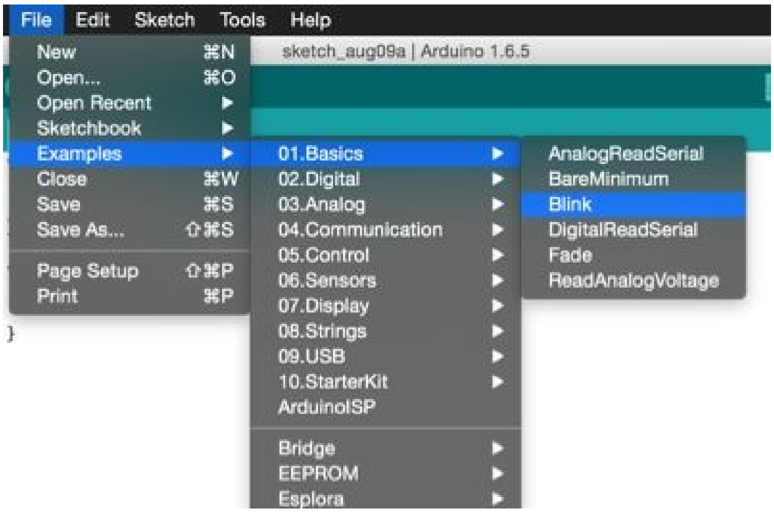
My Foca serial adapter was recognized as /dev/cu.usbserial-AE019521 on my OSX.

2.4 Hello SparkFun ESP8266 Thing: Blinking LED

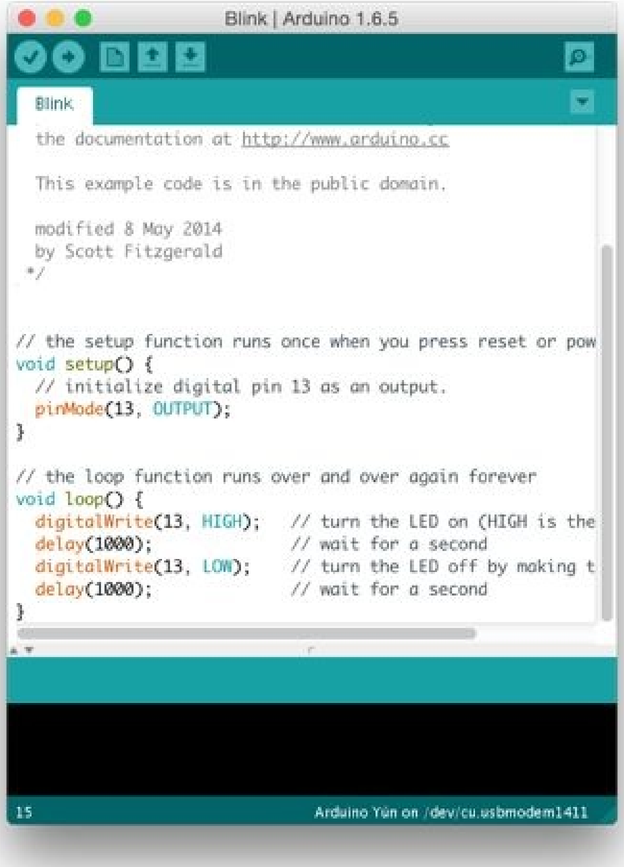
In this section, we build a blinking LED program using Blink program from Arduino. SparkFun ESP8266 Thing board provides onboad LED which is connected on pin 5.

Let’s start to write our Blink program.

Open Arduino. Click menu File -> Examples -> 01.Basics -> Blink.

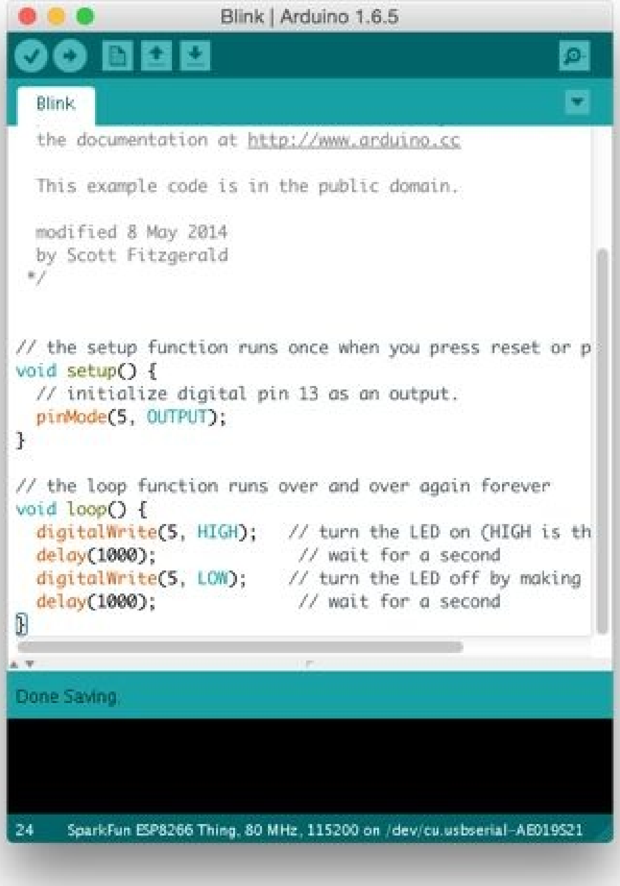


Then, you get Blink code on Arduino.



```
  the documentation at http://www.arduino.cc

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald
 */


// the setup function runs once when you press reset or pow
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // turn the LED off by making t
  delay(1000);              // wait for a second
}
```

Change 13 pin to 5.



```
  the documentation at http://www.arduino.cc

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald
 */


// the setup function runs once when you press reset or p
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(5, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(5, HIGH);   // turn the LED on (HIGH is th
  delay(1000);              // wait for a second
  digitalWrite(5, LOW);    // turn the LED off by making
  delay(1000);              // wait for a second
}
```

Done Saving.

24 SparkFun ESP8266 Thing, 80 MHz, 115200 on /dev/cu.usbserial-AE019S21

Save and compile by pressing Verify icon. To upload program to SparkFun ESP8266 Thing board, you can click Upload icon.

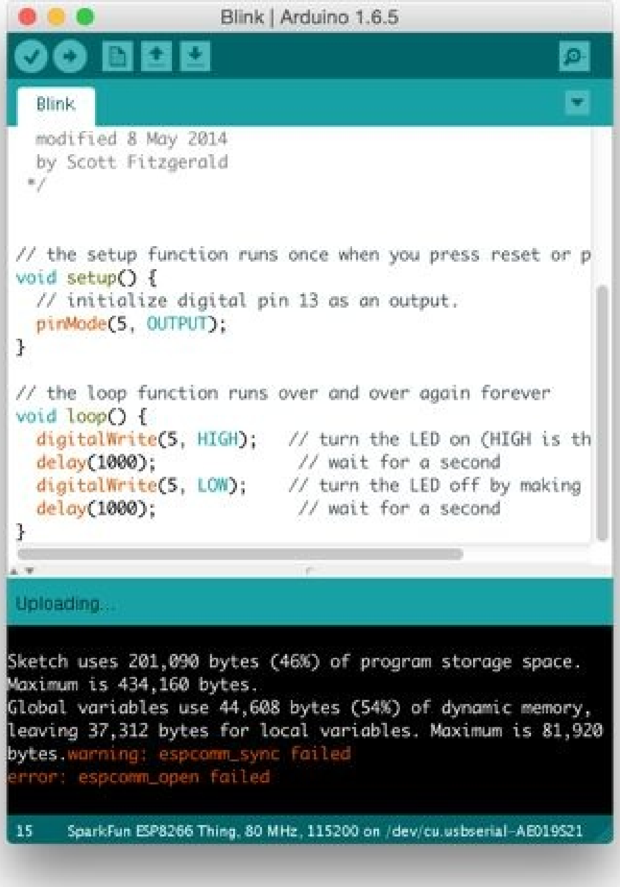
If success, you can see blinking LED on SparkFun ESP8266 Thing board.

If you get error messages as follows,

warning: espcomm_sync failed

error: espcomm_open failed

You can set GPIO0 to GND

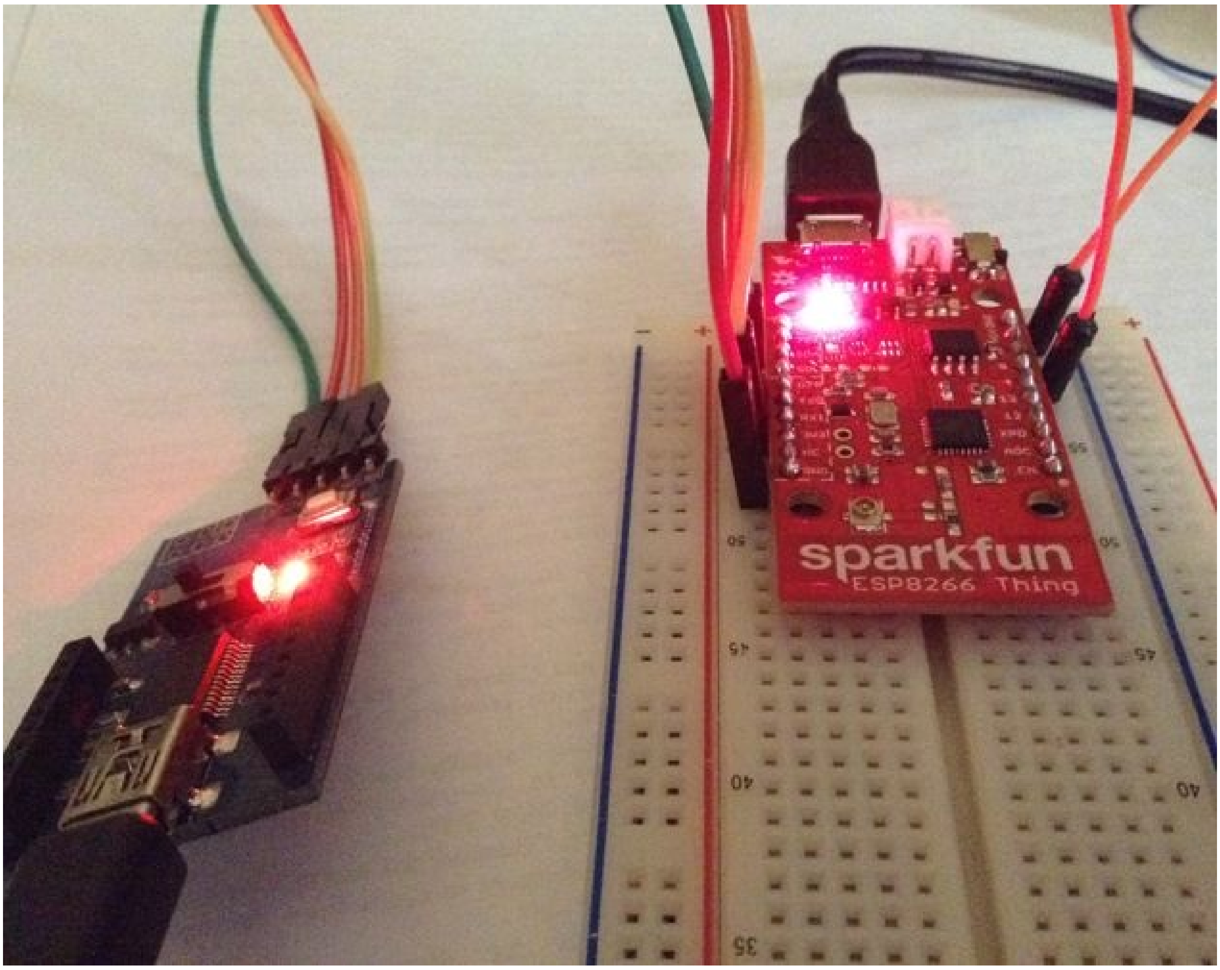


```
  modified 8 May 2014
  by Scott Fitzgerald
 */


// the setup function runs once when you press reset or p
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(5, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(5, HIGH);   // turn the LED on (HIGH is th
  delay(1000);              // wait for a second
  digitalWrite(5, LOW);    // turn the LED off by making
  delay(1000);              // wait for a second
}
```

```
Sketch uses 201,090 bytes (46%) of program storage space.
Maximum is 434,160 bytes.
Global variables use 44,608 bytes (54%) of dynamic memory,
leaving 37,312 bytes for local variables. Maximum is 81,920
bytes.warning: espcomm_sync failed
error: espcomm_open failed
```

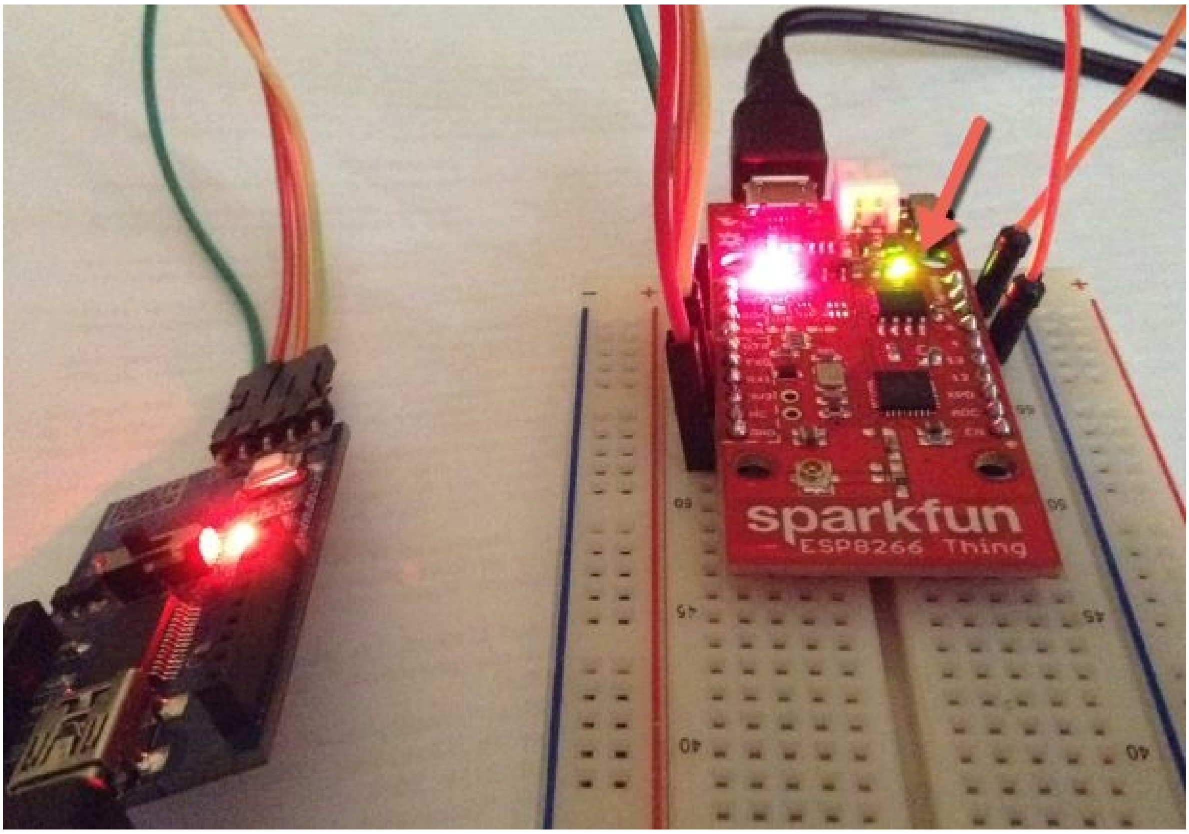
15 SparkFun ESP8266 Thing, 80 MHz, 115200 on /dev/cu.usbserial-AE019S21

Just connect pin 0 to GND.



Then, try again to upload program.

A sample output of blinking program can be seen in Figure below.



2.5 Updating Program

If you have modified the program and want to upload to SparkFun ESP8266 Thing board, you must turn off the board and then turn on. After that, you can upload the program as usual.

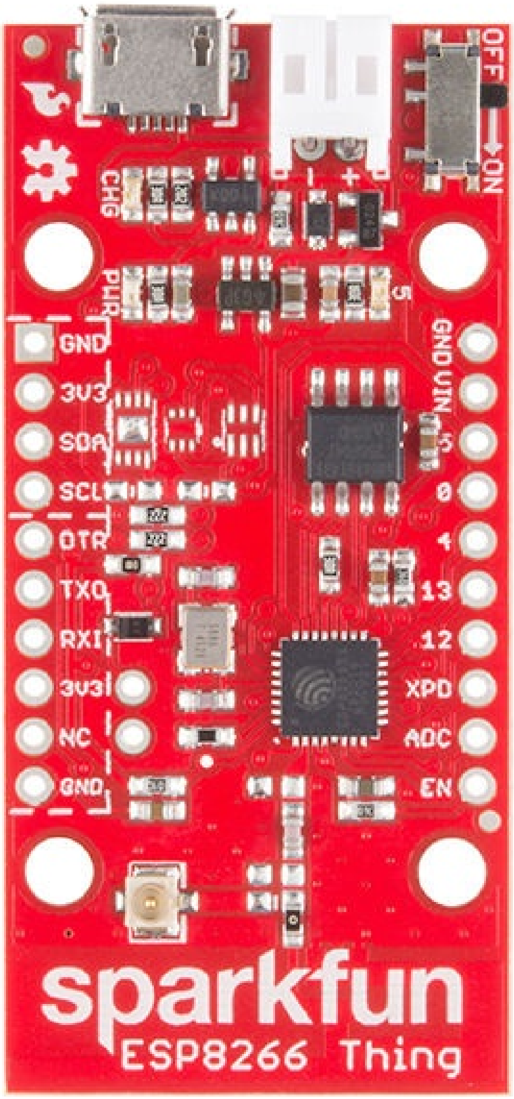
3. GPIO Programming

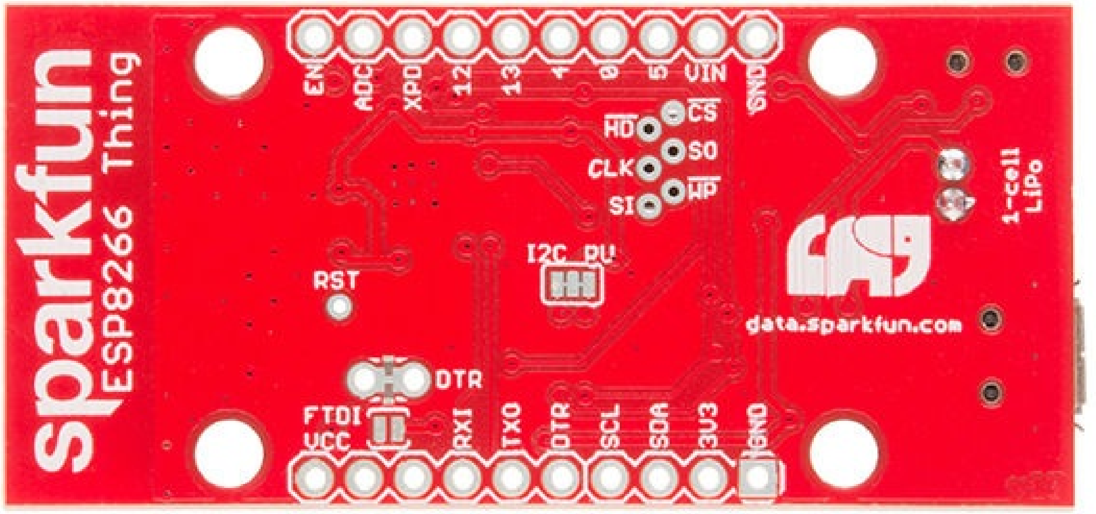
In this chapter I’m going to explain how to work with GPIO on SparkFun ESP8266 Thing and write a program for demo.

3.1 Getting Started

In general, GPIO can be used to control digital I/O on SparkFun ESP8266 Thing. To write data on SparkFun ESP8266 Thing GPIO, we can use digitalWrite() and use digitalRead() to read data from GPIO. To use SparkFun ESP8266 Thing GPIO as digital I/O pins, we must define them using pinMode() with passing OUTPUT or INPUT parameter.

You can see SparkFun ESP8266 Thing GPIO pinout on the following Figure.



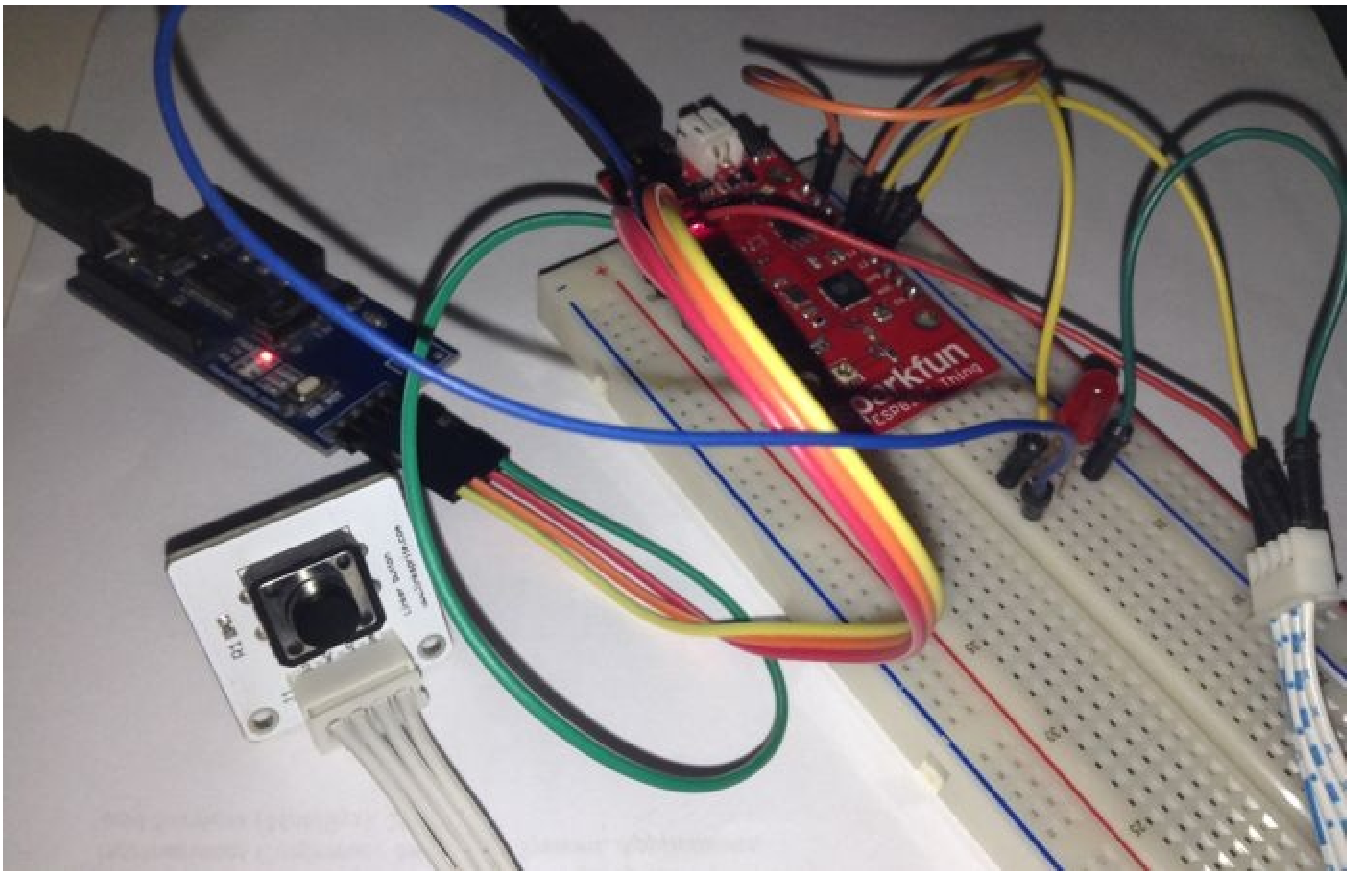


In this chapter, we build a program to illustrate how SparkFun ESP8266 Thing GPIO work. We need a LED and a pushbutton.

Let’s start!.

3.2 Wiring

Connect LED to pin 13 on SparkFun ESP8266 Thing and pushbutton to pin 4 The following is a sample of wiring.



3.3 Writing a Program

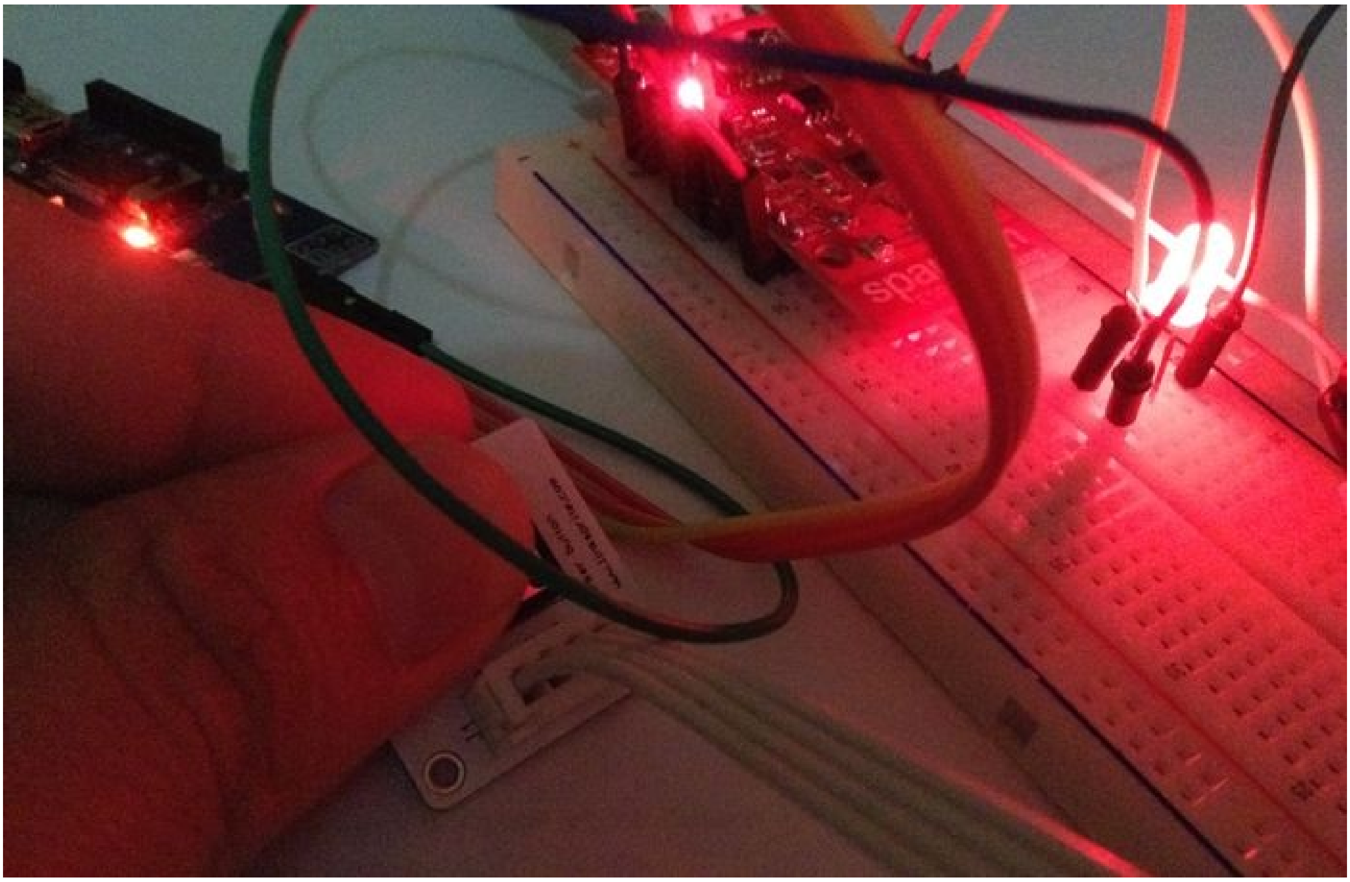
To create a program, we just open Arduino IDE and write this code.

```
int led = 13;
int pushButton = 4;
int state = 0;

void setup() {
  pinMode(led, OUTPUT);
  pinMode(pushButton, INPUT);
}

void loop() {
  state = digitalRead(pushButton);
  digitalWrite(led,state);
  delay(300);
}
```

Save these codes as ButtonLed.

3.4 Testing

Now you can upload and run this program to SparkFun ESP8266 Thing board. For testing, try to press pushbutton. You should see a lighting LED.



4. UART

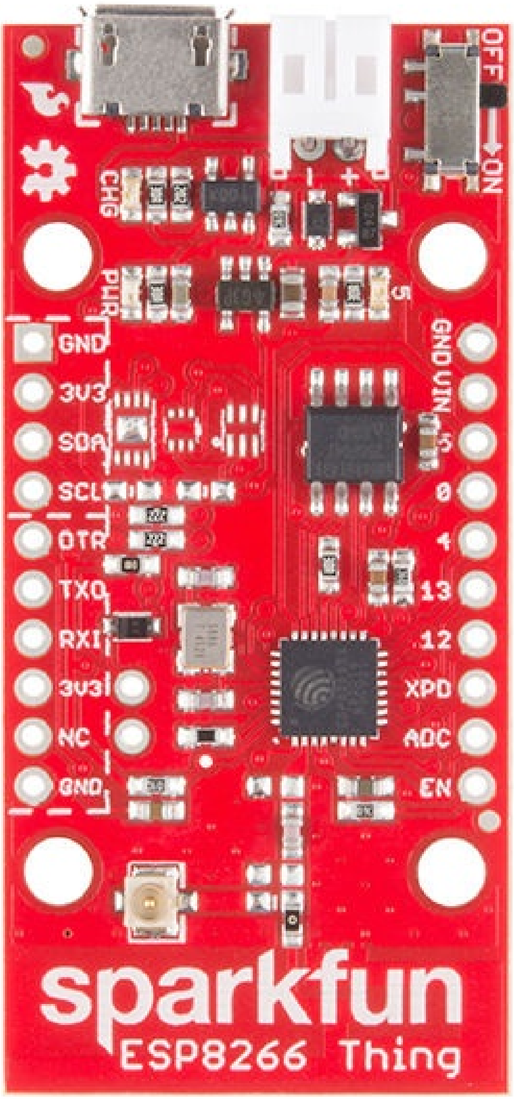
In this chapter I’m going to explain how to access UART on SparkFun ESP8266 Thing board.

4.1 Getting Started

SparkFun ESP8266 Thing provides UART which can be accessed via Serial library or SoftwareSerial. Further information about Serial object, you can read it on https://www.arduino.cc/en/Reference/Serial . We can call Serial.read() to read one byte from UART and Serial.write() to write one byte into UART.

In this chapter, we try to access SparkFun ESP8266 Thing UART via serial adapter which is used to upload a program too.

Let’s start!.



4.2 Wiring

In this scenario, we use the same wiring from chapter 3. We will show pressed state from push button on Serial.

4.3 Writing a Program

To use Serial object, we need to initialize it by calling Serial.begin(baudrate). In this case, we can user baudrate 115200 on SparkFun ESP8266 Thing.

Open Arduino software and write this program.

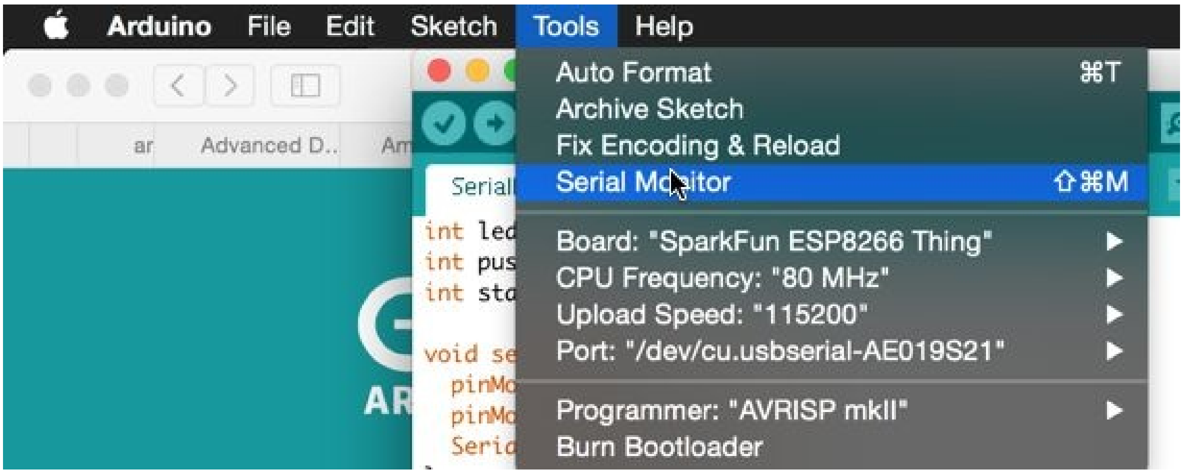
```
int led = 13;
int pushButton = 4;
int state = 0;

void setup() {
  pinMode(led, OUTPUT);
  pinMode(pushButton, INPUT);
  Serial.begin(115200);
}

void loop() {
  state = digitalRead(pushButton);
  digitalWrite(led,state);
  Serial.print("State=");
  Serial.println(state);
  delay(300);
}
```

Save this program as SerialDemo.

4.4 Testing

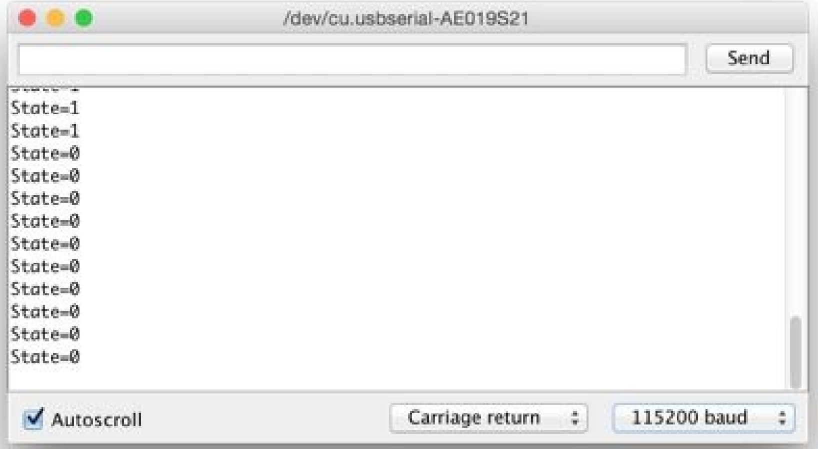
Now you can upload and run program. Don't forget to set board target with SparkFun ESP8266 Thing. Read section 2.4 to upload the program.

To see the UART output, open Serial Monitor tool from Arduino IDE. Set baud 115200 and Carriage return.



Board: "SparkFun ESP8266 Thing"
CPU Frequency: "80 MHz"
Upload Speed: "115200"
Port: "/dev/cu.usbserial-AE019S21"

You should see the UART output. A sample output can seen in Figure below.



```
State=1
State=1
State=0
State=0
State=0
State=0
State=0
State=0
State=0
State=0
State=0
State=0
```

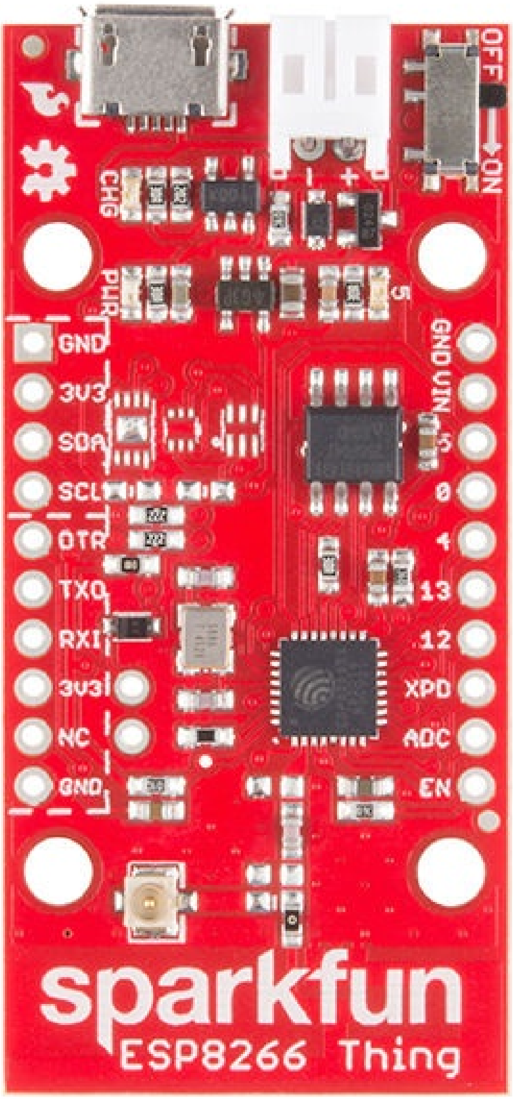
☑ Autoscroll
Carriage return
115200 baud

5. PWM and Analog Input

This chapter explains how to work with SparkFun ESP8266 Thing Analog I/O.

5.1 Getting Started

SparkFun ESP8266 Thing board provides Analog I/O which can be connected to sensor or actuator devices. See the following of SparkFun ESP8266 Thing GPIO.



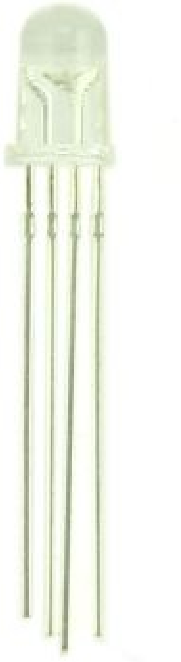
In this chapter, we try to access SparkFun ESP8266 Thing Analog I/O using Arduino software. There are two scenarios for our cases:

- Controlling RGB LED
- Reading Analog input using Potentiometer

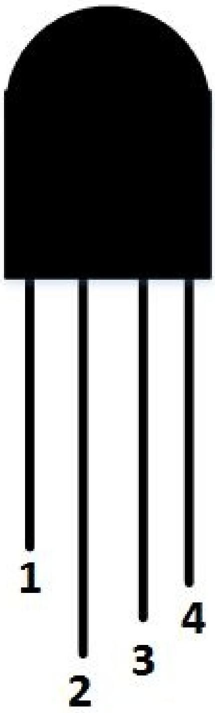
Let’s start.

5.2 Demo Analog Output (PWM) : RGB LED

In this scenario we build a program to control RGB LED color using SparkFun ESP8266 Thing Analog output (PWM). RGB LED has 4 pins that you can see it on Figure below.



To understand these pins, you can see the following Figure.



Note:

- Pin 1: Red
- Pin 2: Common pin

- Pin 3: Green
- Pin 4: Blue

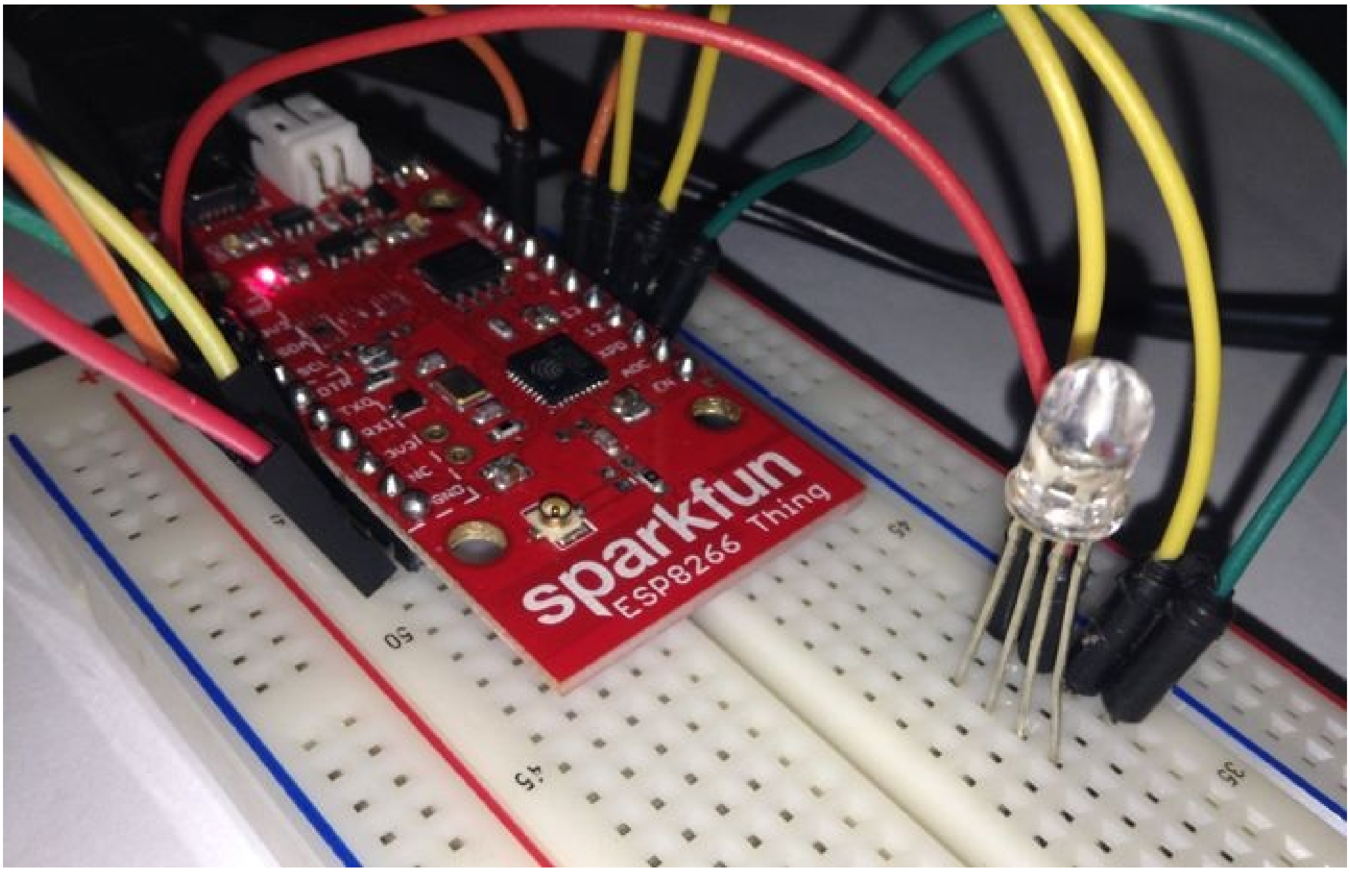
Now we can start to build a program and hardware implementation.

5.2.1 Wiring

For our testing, we configure the following PWM pins.

- RGB LED pin 1 (red) is connected to SparkFun ESP8266 Thing pin 4
- RGB LED pin 2 is connected to SparkFun ESP8266 Thing 3V3 (VCC +3.3V)
- RGB LED pin 3 (green) is connected to SparkFun ESP8266 Thing pin 13
- RGB LED pin 4 (blue) is connected to SparkFun ESP8266 Thing pin 12

Here is a sample implementation with SparkFun ESP8266 Thing and RGB Led.



5.2.2 Writing Program

To display a certain color, we must combine colors from red, green, blue. SparkFun ESP8266 Thing provides API for PWM like Arduino API such as analogWrite() and analogRead() but analog value from 0 to 1023.

Let”s start to build a program. Firstly, open Arduino Software. Then, write these scripts.

```
int redPin = 4;
int greenPin = 13;
int bluePin = 12;


void setup()
{
    pinMode(redPin, OUTPUT);
    pinMode(greenPin, OUTPUT);
    pinMode(bluePin, OUTPUT);
    Serial.begin(115200);
}

void loop()
{
  setColor(0, 1023, 1023);  // red
  Serial.println("red");
  delay(1000);
  setColor(1023, 0, 1023);  // green
  Serial.println("green");
  delay(1000);
  setColor(1023, 1023, 0);  // blue
  Serial.println("blue");
  delay(1000);
  setColor(0, 0, 1023);  // yellow
  Serial.println("yellow");
  delay(1000);
  setColor(700, 1023, 700);  // purple
  Serial.println("purple");
  delay(1000);
  setColor(1023, 0, 0);  // aqua
  Serial.println("aqua");
  delay(1000);
}

void setColor(int red, int green, int blue)
{
  analogWrite(redPin, red);
  analogWrite(greenPin, green);
  analogWrite(bluePin, blue);
}
```

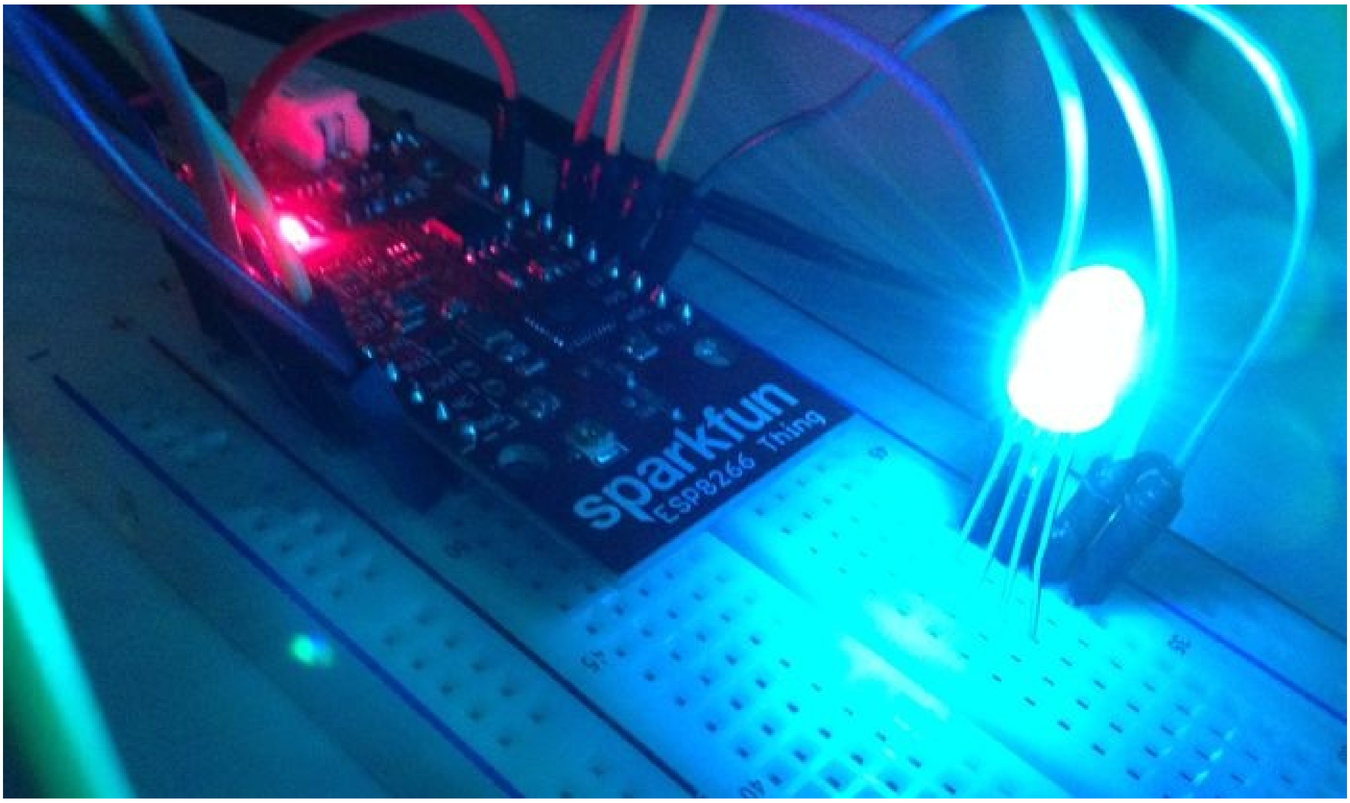
Save this program as PWMDemo.

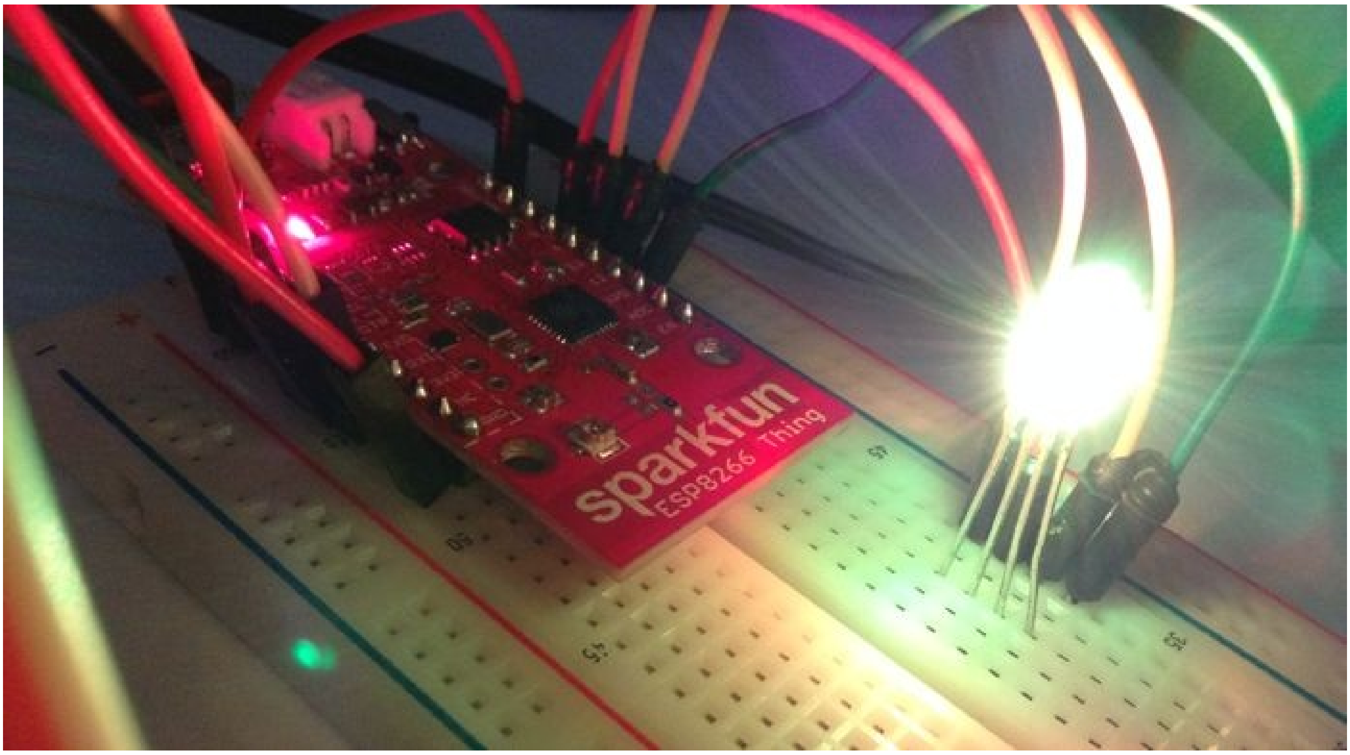
This program will generate six colors: red, green, blue, yellow, purple, and aqua.

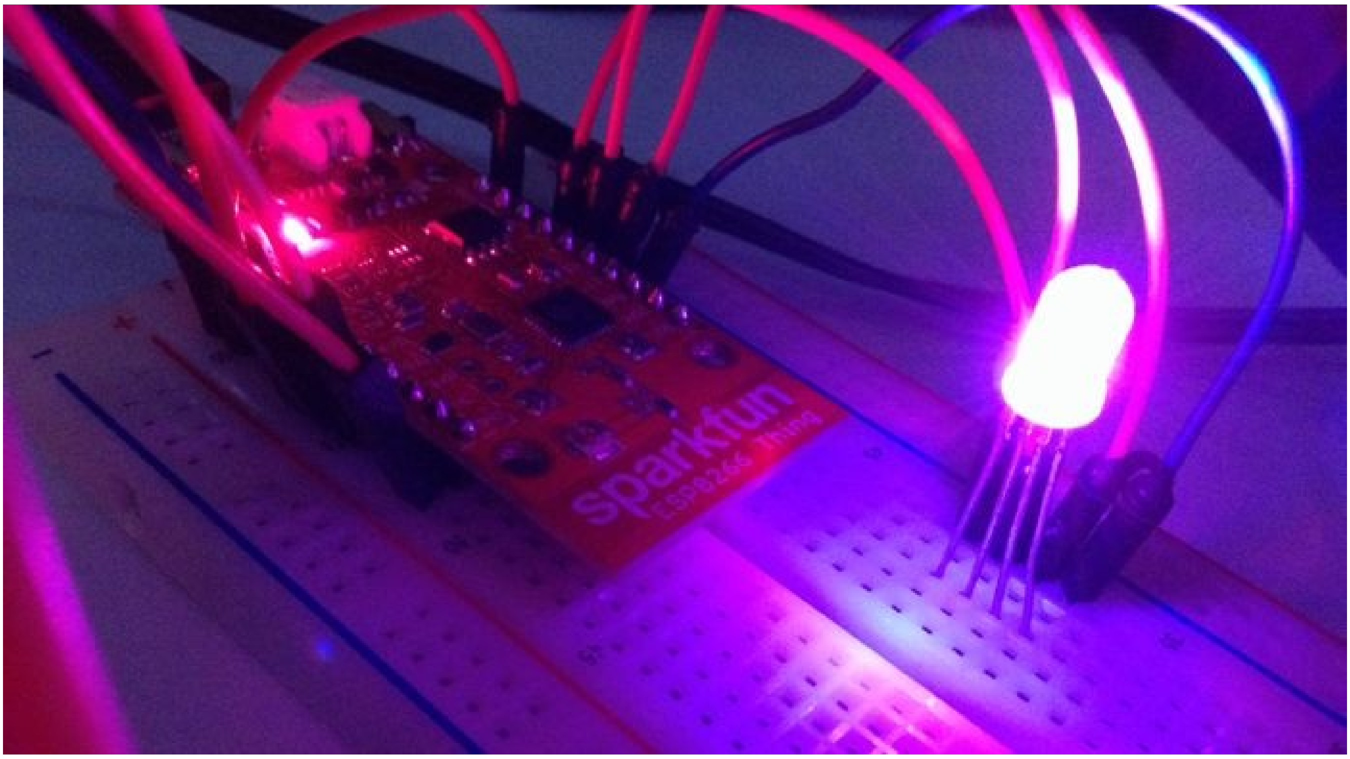
5.2.3 Testing

Upload and run the program. You should see several color on RGB LED.

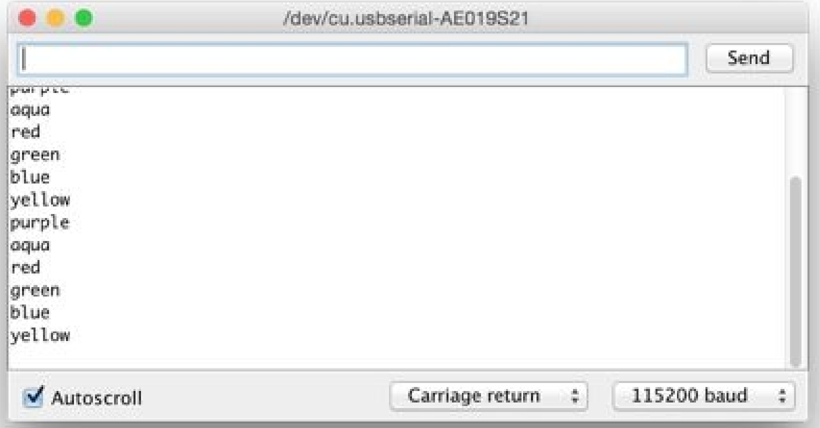
The following is a sample demo on RGB LED.







If you connect to SparkFun ESP8266 Thing board via Serial monitor tool from Arduino, you should get program output, shown in Figure below.



```
purple
aqua
red
green
blue
yellow
purple
aqua
red
green
blue
yellow
```

☑ Autoscroll

Carriage return

115200 baud

5.3 Demo Analog Input: Working with Potentiometer

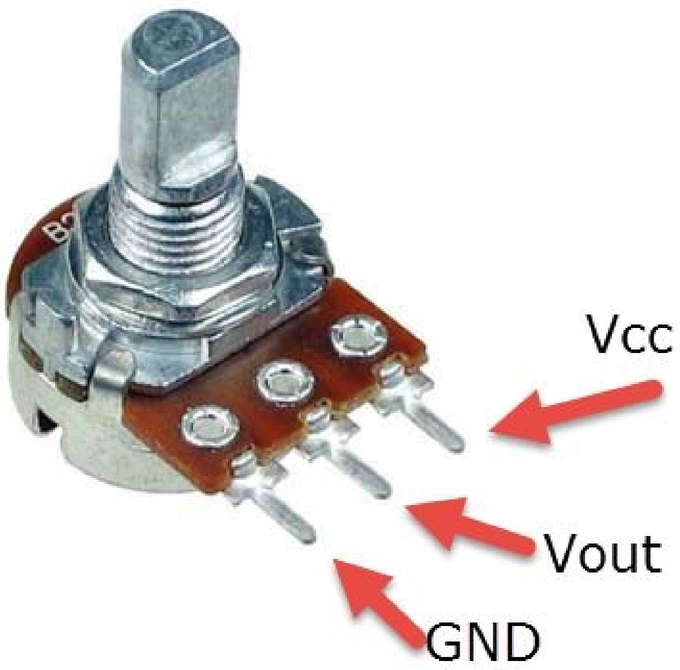
In this section, we learn how to read analog input on SparkFun ESP8266 Thing board. For illustration, I use Potentiometer as analog input source. Our scenario is to read analog value from Potentiometer. Then, display it on Serial Monitor.

SparkFun ESP8266 Thing only has one ADC on A0. FYI, SparkFun ESP8266 Thing ADC can receive maximum 1V so if your analog input 3.3V SparkFun ESP8266 Thing treats it as 1V ADC. If you want to work with many analog input, you must expand it using ICs based ADC. In this section, we are working on SparkFun ESP8266 Thing ADC on A0.

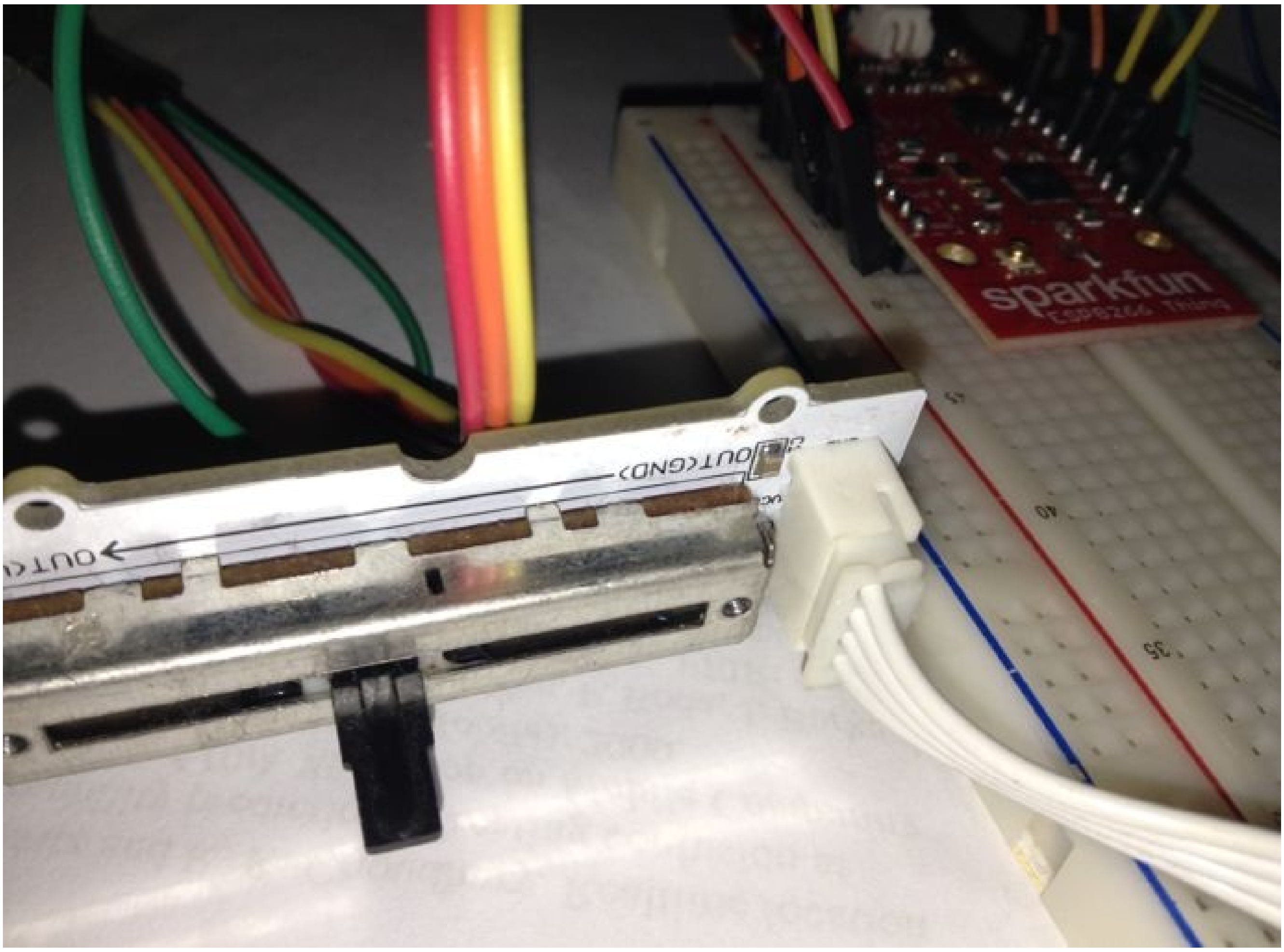
Let’s start!.

5.3.1 Wiring

To understand Potentiometer, you see its scheme in Figure below.



You can connect VCC to SparkFun ESP8266 Thing board on 3V3 pin (VCC +3.3V). Vout to SparkFun ESP8266 Thing board Analog input ADC (A0). In addition, GND to SparkFun ESP8266 Thing board GND. The following is hardware implementation. I use slide potentiometer.



5.3.2 Writing Program

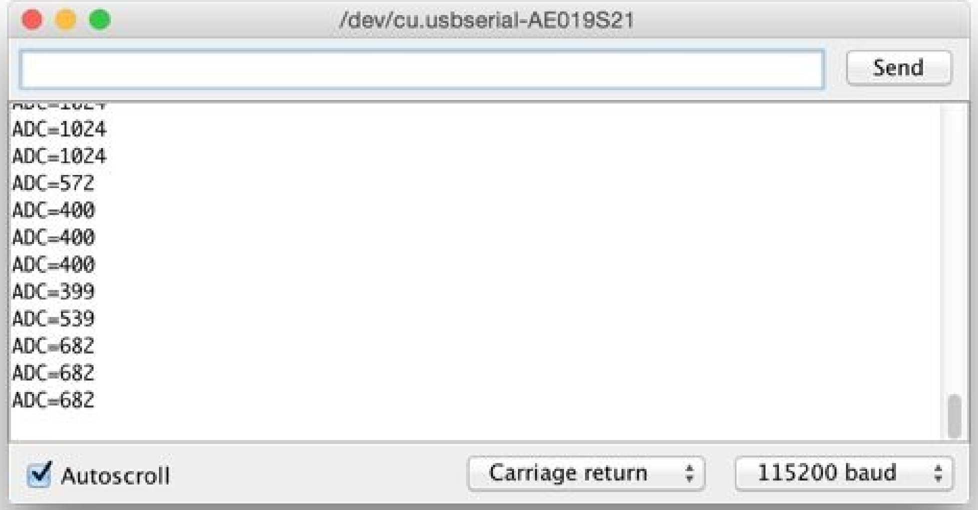
Firstly, create a program using Arduino IDE. To read analog input, we can use analogRead() function. Ok, Let’s write these scripts.

```
int val = 0;

void setup() {
  Serial.begin(115200);
}

void loop() {
  val = analogRead(A0);
  Serial.print("ADC=");
  Serial.println(val);
  delay(300);
}
```

Save this code as ADCDemo.

5.3.3 Testing

Upload and run this program. If success, you can see analog value using Serial Monitor tool from Arduino IDE.



```
ADC=1024
ADC=1024
ADC=572
ADC=400
ADC=400
ADC=400
ADC=399
ADC=539
ADC=682
ADC=682
ADC=682
```

☑ Autoscroll

Carriage return

115200 baud

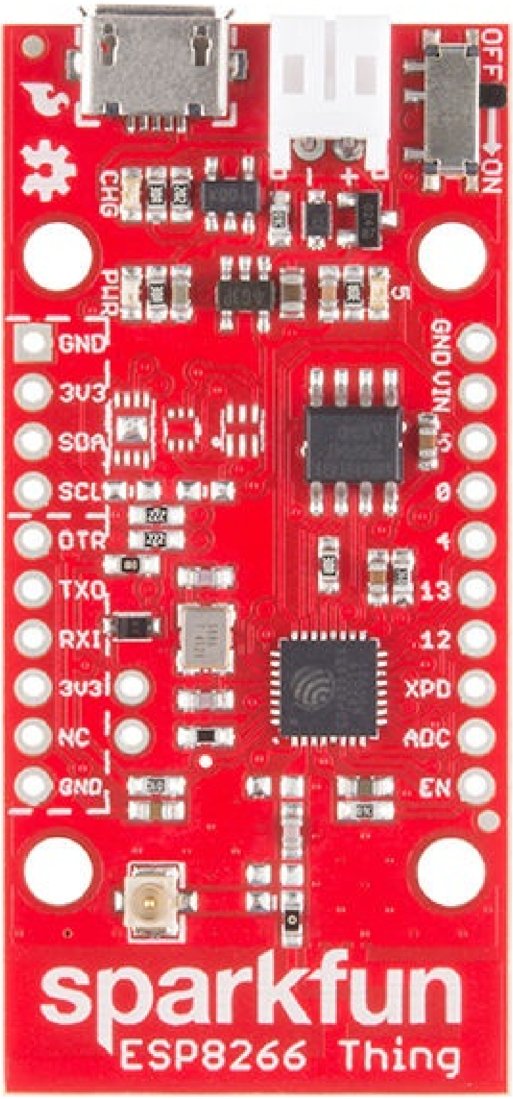
6. Working with I2C

In this chapter we learn how to work with I2C on SparkFun ESP8266 Thing board using Arduino program.

6.1 Getting Started

The I2C (Inter-Integrated Circuit) bus was designed by Philips in the early ’80s to allow easy communication between components which reside on the same circuit board. TWI stands for Two Wire Interface and for most marts this bus is identical to I²C. The name TWI was introduced by Atmel and other companies to avoid conflicts with trademark issues related to I²C.

I2C bus consists of two wires, SDA (Serial Data Line) and SCL (Serial Clock Line). You can see I2C pins on SparkFun ESP8266 Thing board, shown in Figure below.



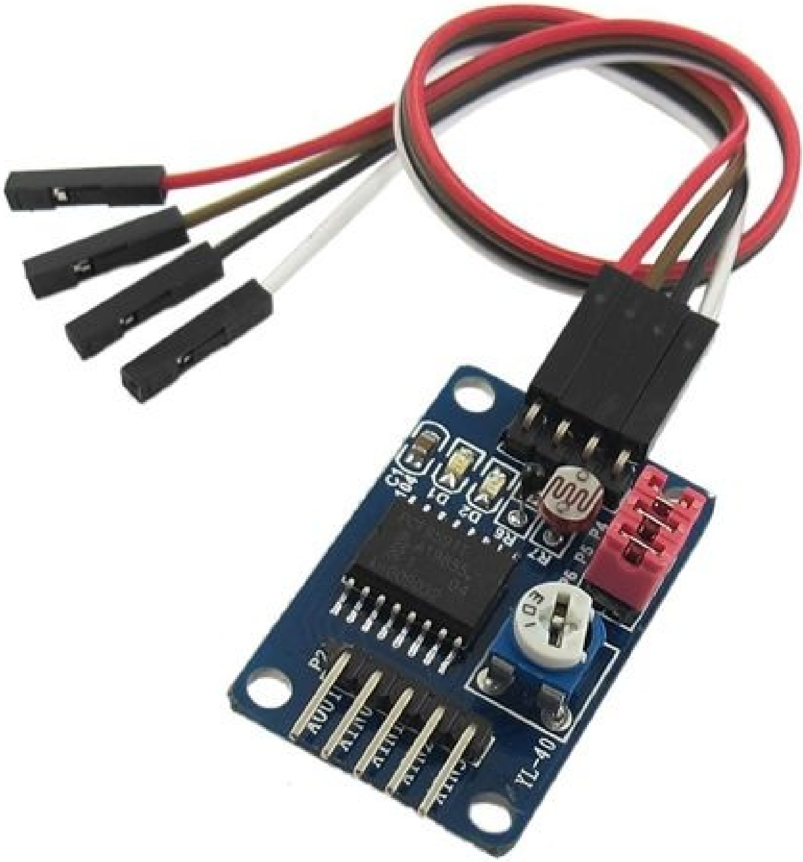
For testing, I used PCF8591 AD/DA Converter module with sensor and actuator devices. You can find it on the following online store:

- Amazon, http://www.amazon.com/PCF8591-Converter-Module-Digital-Conversion/dp/B00BXX4UWC/
- eBay, http://www.ebay.com
- Dealextreme, http://www.dx.com/p/pcf8591-ad-da-analog-to-digital-digital-to-

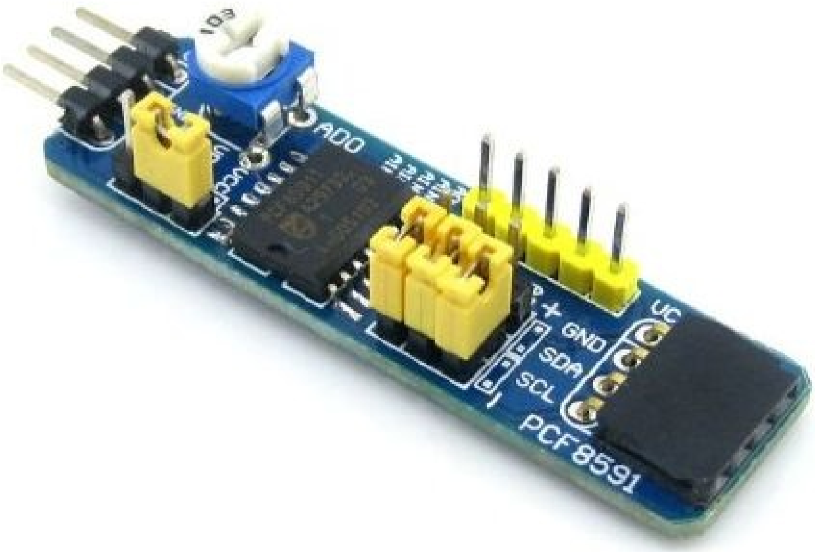
analog-converter-module-w-dupont-cable-deep-blue-336384

- Aliexpress, http://www.aliexpress.com/

In addition, you can find this device on your local electronics store/online store.



This module has mini form model too, for instance, you can find it on Amazon, http://www.amazon.com/WaveShare-PCF8591T-Converter-Evaluation-Development/dp/B00KM6X2OI/ .



This module use PCF8591 IC and you can read the datasheet on the following URLs.

- http://www.electrodragon.com/w/images/e/ed/PCF8591.pdf
- http://www.nxp.com/documents/data_sheet/PCF8591.pdf

In this chapter, we build a program to access sensor via I2C using Arduino software on SparkFun ESP8266 Thing board.

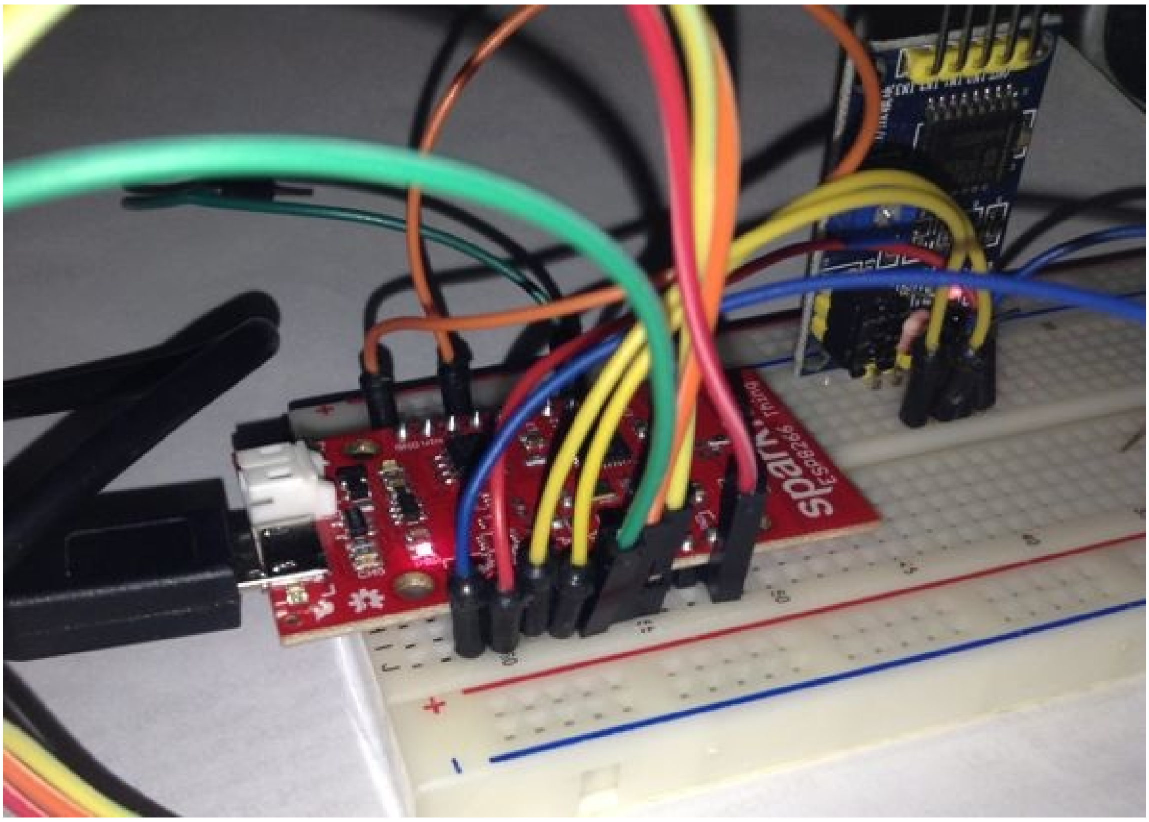
6.2 Writing Program

We use PCF8591 AD/DA Converter as I2C source. You can connect PCF8591 AD/DA Converter to SparkFun ESP8266 Thing board directly.

The following is our wiring lab:

- PCF8591 AD/DA Converter SDA —> SparkFun ESP8266 Thing SDA
- PCF8591 AD/DA Converter SCL —> SparkFun ESP8266 Thing CLK
- PCF8591 AD/DA Converter VCC —> SparkFun ESP8266 Thing VCC 3V3
- PCF8591 AD/DA Converter GND —> SparkFun ESP8266 Thing GND

Hardware implementation can be shown in Figure below.



6.3 Writing Program

We use I2C on SparkFun ESP8266 Thing board using Wire library like Arduino way. PCF8591 AD/DA Converter module has three sensor devices: Thermistor, Photo-voltaic cell and Potentiometer. This module runs on I2C bus with address 0x90. In this case, we read all sensor data.

Open Arduino IDE and write this code.

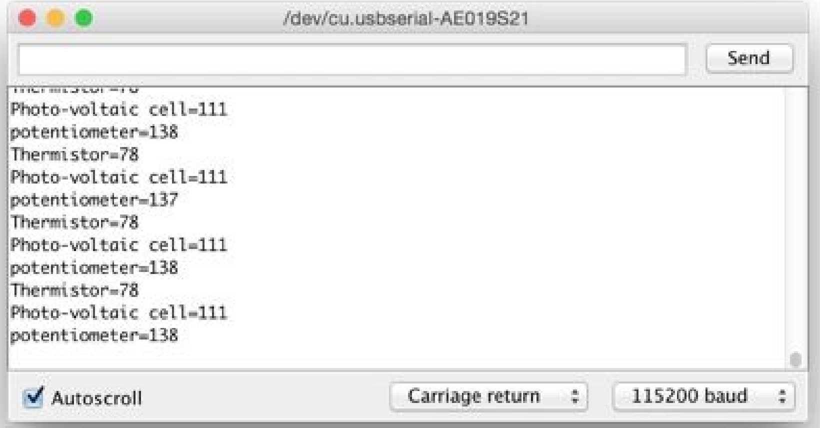
```
#include "Wire.h"
#define PCF8591 (0x90 >> 1) // I2C bus address
#define PCF8591_ADC_CH0 0x00 // thermistor
#define PCF8591_ADC_CH1 0x01 // photo-voltaic cell
#define PCF8591_ADC_CH2 0x02
#define PCF8591_ADC_CH3 0x03 // potentiometer
byte ADC1, ADC2, ADC3;

void setup()
{
  Wire.begin();
  Serial.begin(115200);
}
void loop()
{
  // read thermistor
  Wire.beginTransmission(PCF8591);
  Wire.write(PCF8591_ADC_CH0);
  Wire.endTransmission();
  Wire.requestFrom(PCF8591, 2);
  ADC1=Wire.read();
  ADC1=Wire.read();

  Serial.print("Thermistor=");
  Serial.println(ADC1);

  // read photo-voltaic cell
  Wire.beginTransmission(PCF8591);
  Wire.write(PCF8591_ADC_CH1);
  Wire.endTransmission();
  Wire.requestFrom(PCF8591, 2);
  ADC2=Wire.read();
  ADC2=Wire.read();

  Serial.print("Photo-voltaic cell=");
  Serial.println(ADC2);

  // potentiometer
  Wire.beginTransmission(PCF8591);
  Wire.write(PCF8591_ADC_CH3);
  Wire.endTransmission();
  Wire.requestFrom(PCF8591, 2);
  ADC3=Wire.read();
  ADC3=Wire.read();

  Serial.print("potentiometer=");
  Serial.println(ADC3);

  delay(500);
}
```

Save this code as I2CSensor.

6.4 Testing

Now you can upload and run the program to SparkFun ESP8266 Thing board board.

If success, open Serial monitor tool and connect to SparkFun ESP8266 Thing to see the program output. The following is a sample output.



```
Photo-voltaic cell=111
potentiometer=138
Thermistor=78
Photo-voltaic cell=111
potentiometer=137
Thermistor=78
Photo-voltaic cell=111
potentiometer=138
Thermistor=78
Photo-voltaic cell=111
potentiometer=138
```

☑ Autoscroll

Carriage return

115200 baud

7. SPI

In this chapter I’m going to explain how to work with SPI on SparkFun ESP8266 Thing board.

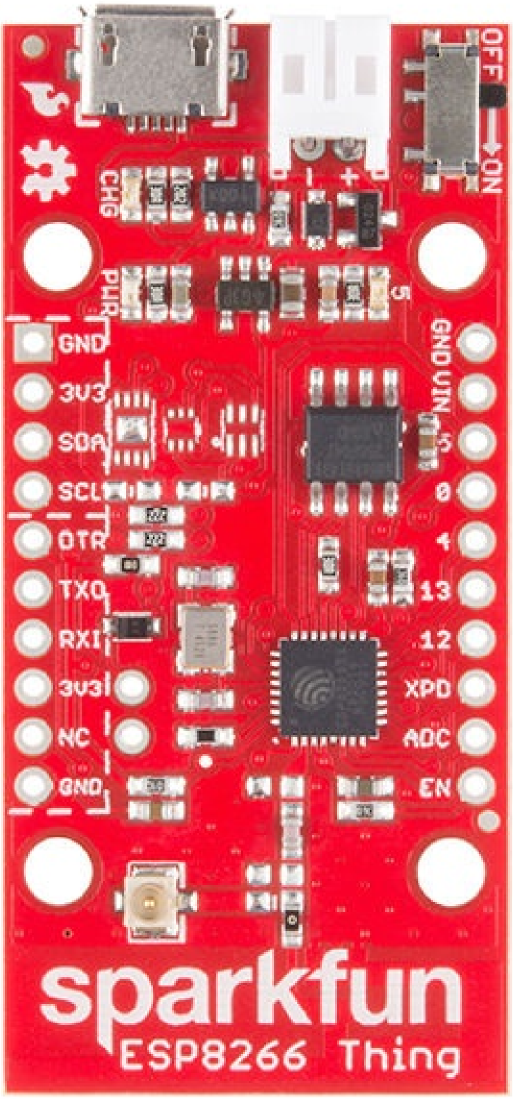
7.1 Getting Started

The Serial Peripheral Interface (SPI) is a communication bus that is used to interface one or more slave peripheral integrated circuits (ICs) to a single master SPI device; usually a microcontroller or microprocessor of some sort.

SPI in SparkFun ESP8266 Thing board can be defined on the following pins:

- MOSI on pin 13
- MISO on pin 12
- SCLK on pin 14 (SCL)

You can see these SPI pins on SparkFun ESP8266 Thing board, shown in Figure below.



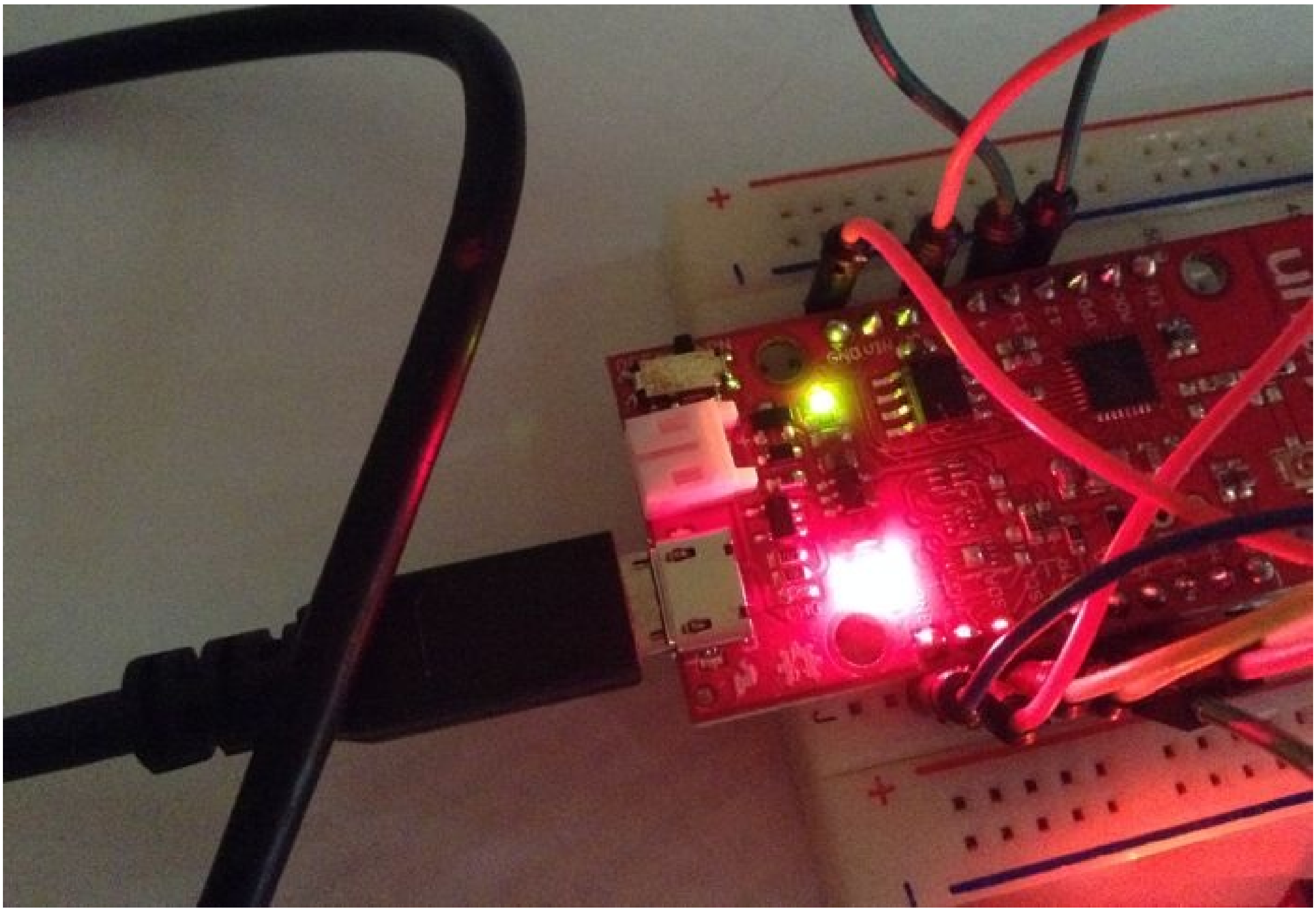
We can only use one SPI on SparkFun ESP8266 Thing board with SPI master mode. We develop program based SPI using SPI library, https://www.arduino.cc/en/Reference/SPI .

In this chapter, we build a SPI Loopback app. Let’s start!.

7.2 Wiring

To develop SPI loopback, we can connect MOSI pin to MISO pin. This means you connect pin 13 to pin 12 using cable.

The following is a sample of wiring.



7.3 Writing a Program

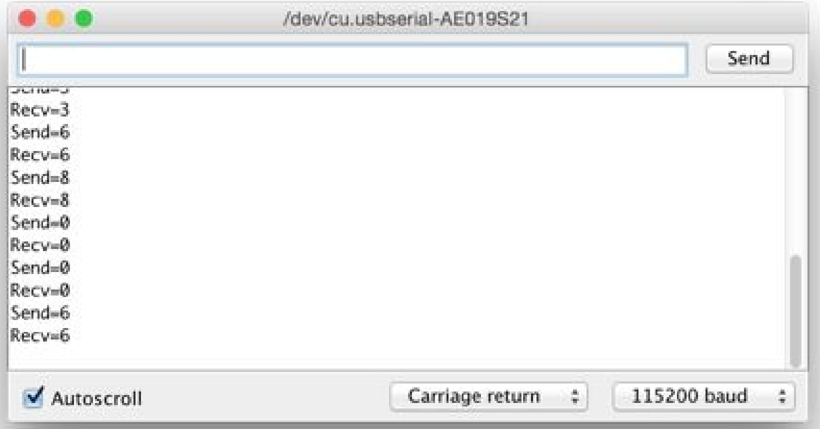
Firstly, we write a program for SparkFun ESP8266 Thing. Write these codes on Arduino IDE.

```
#include <SPI.h>

byte sendData,recvData;
void setup() {
  SPI.begin();
  SPI.setFrequency(1000000);
  Serial.begin(115200);

}

// source:
// http://forum.arduino.cc/index.php?topic=197633.0
byte randomDigit() {
  unsigned long t = micros();
  byte r = (t % 10) + 1;
  for (byte i = 1; i <= 4; i++) {
    t /= 10;
    r *= ((t % 10) + 1);
    r %= 11;
  }
  return (r - 1);
}

void loop() {
  sendData = randomDigit();
  recvData = SPI.transfer(sendData);

  Serial.print("Send=");
  Serial.println(sendData,DEC);
  Serial.print("Recv=");
  Serial.println(recvData,DEC);
  delay(800);
}
```

Save this code.

7.4 Testing

Now you can upload program to SparkFun ESP8266 Thing board. If done, open Serial Monitor tool from Arduino. You should see received data from SPI.



/dev/cu.usbserial-AE019S21

```
Recv=3
Send=6
Recv=6
Send=8
Recv=8
Send=0
Recv=0
Send=0
Recv=0
Send=6
Recv=6
```

☑ Autoscroll

Carriage return

115200 baud

8. Connecting to a Network

In this chapter I’m going to explain how to connect a network from SparkFun ESP8266 Thing.

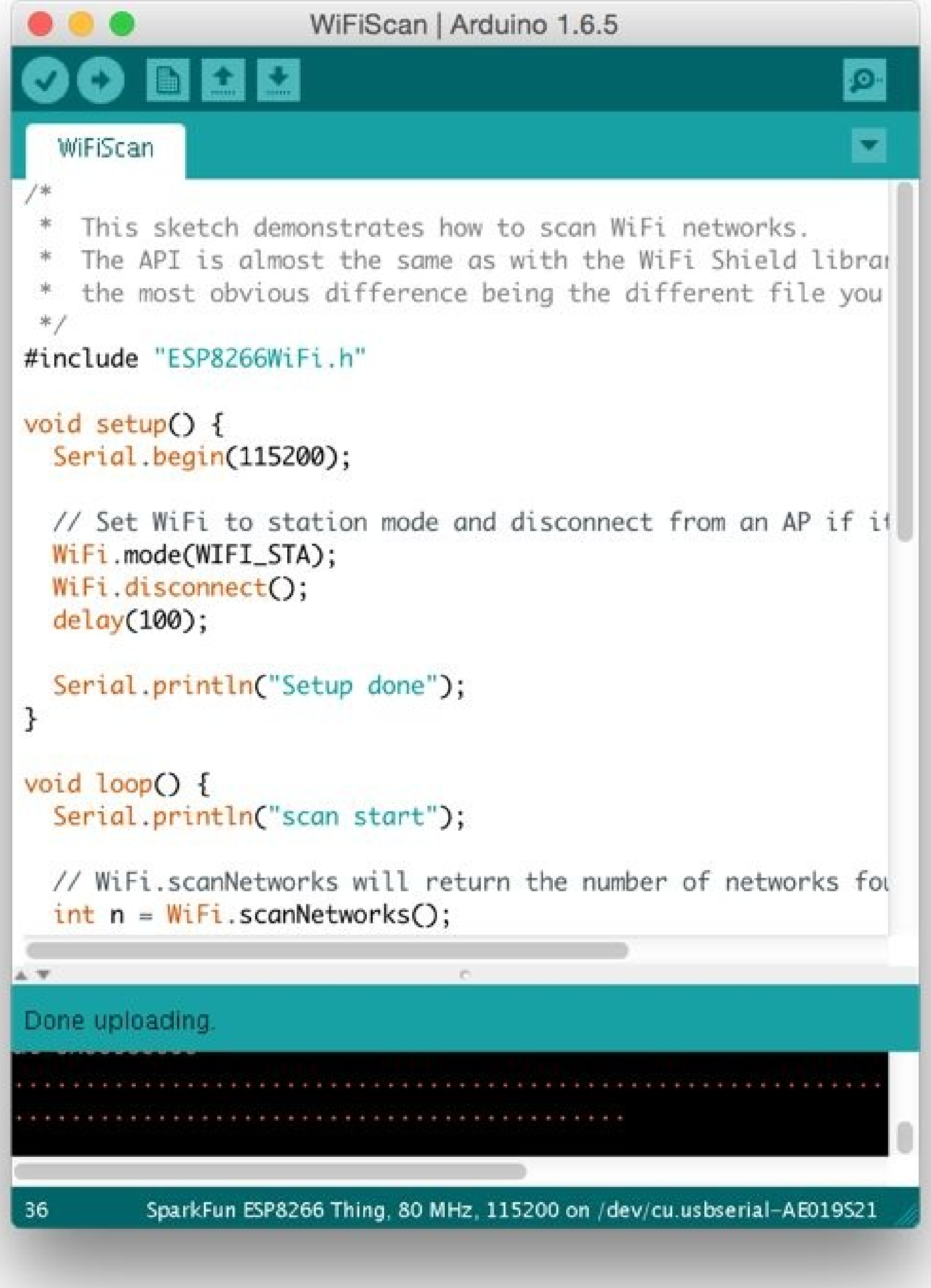
8.1 Getting Started

SparkFun ESP8266 Thing has WiFi Soc, ESP8266. This module can work as AP and WiFi client. To access WiFi module on SparkFun ESP8266 Thing, we can use WIFI library like Arduino. We just change header file, WiFi.h to ESP8266WiFi.h.

In this chapter, we try to scan existing WiFi hotspot and build a simple IoT which turns on/off LEDs remotely via a browser.

8.2 Scanning WiFi Networks

In this section, we can use existing sample program from SparkFun ESP8266 Thing on Arduino IDE. Just select menu File ->Examples -> ESP8266WiFi -> WiFiScan . Then, you get the program.



```
/*
 *  This sketch demonstrates how to scan WiFi networks.
 *  The API is almost the same as with the WiFi Shield librar
 *  the most obvious difference being the different file you
 */
#include "ESP8266WiFi.h"

void setup() {
  Serial.begin(115200);

  // Set WiFi to station mode and disconnect from an AP if it
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);

  Serial.println("Setup done");
}

void loop() {
  Serial.println("scan start");

  // WiFi.scanNetworks will return the number of networks fou
  int n = WiFi.scanNetworks();
```

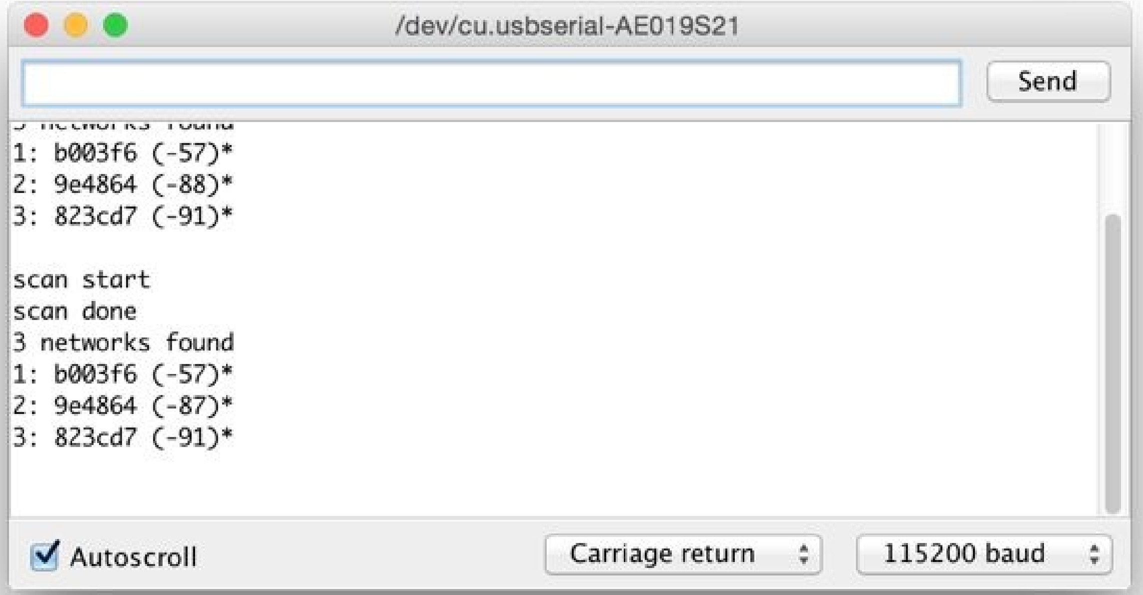
Done uploading.

36   SparkFun ESP8266 Thing, 80 MHz, 115200 on /dev/cu.usbserial-AE019S21

Change baudrate with 115200. Then, compile and upload the program into SparkFun ESP8266 Thing board.

```
#include "ESP8266WiFi.h"

void setup() {
  Serial.begin(115200);

  // Set WiFi to station mode and disconnect from an AP if it was previo
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);

  Serial.println("Setup done");
}

void loop() {
  Serial.println("scan start");

  // WiFi.scanNetworks will return the number of networks found
  int n = WiFi.scanNetworks();
  Serial.println("scan done");
  if (n == 0)
    Serial.println("no networks found");
  else
  {
    Serial.print(n);
    Serial.println(" networks found");
    for (int i = 0; i < n; ++i)
    {
      // Print SSID and RSSI for each network found
      Serial.print(i + 1);
      Serial.print(": ");
      Serial.print(WiFi.SSID(i));
      Serial.print(" (");
      Serial.print(WiFi.RSSI(i));
      Serial.print(")");
      Serial.println((WiFi.encryptionType(i) == ENC_TYPE_NONE)?" ":"*");
      delay(10);
    }
  }
  Serial.println("");

  // Wait a bit before scanning again
  delay(5000);
}
```

After that, open Serial Monitor. You should see a list of WiFi hotspot.



```
3 networks found
1: b003f6 (-57)*
2: 9e4864 (-88)*
3: 823cd7 (-91)*

scan start
scan done
3 networks found
1: b003f6 (-57)*
2: 9e4864 (-87)*
3: 823cd7 (-91)*
```

☑ Autoscroll

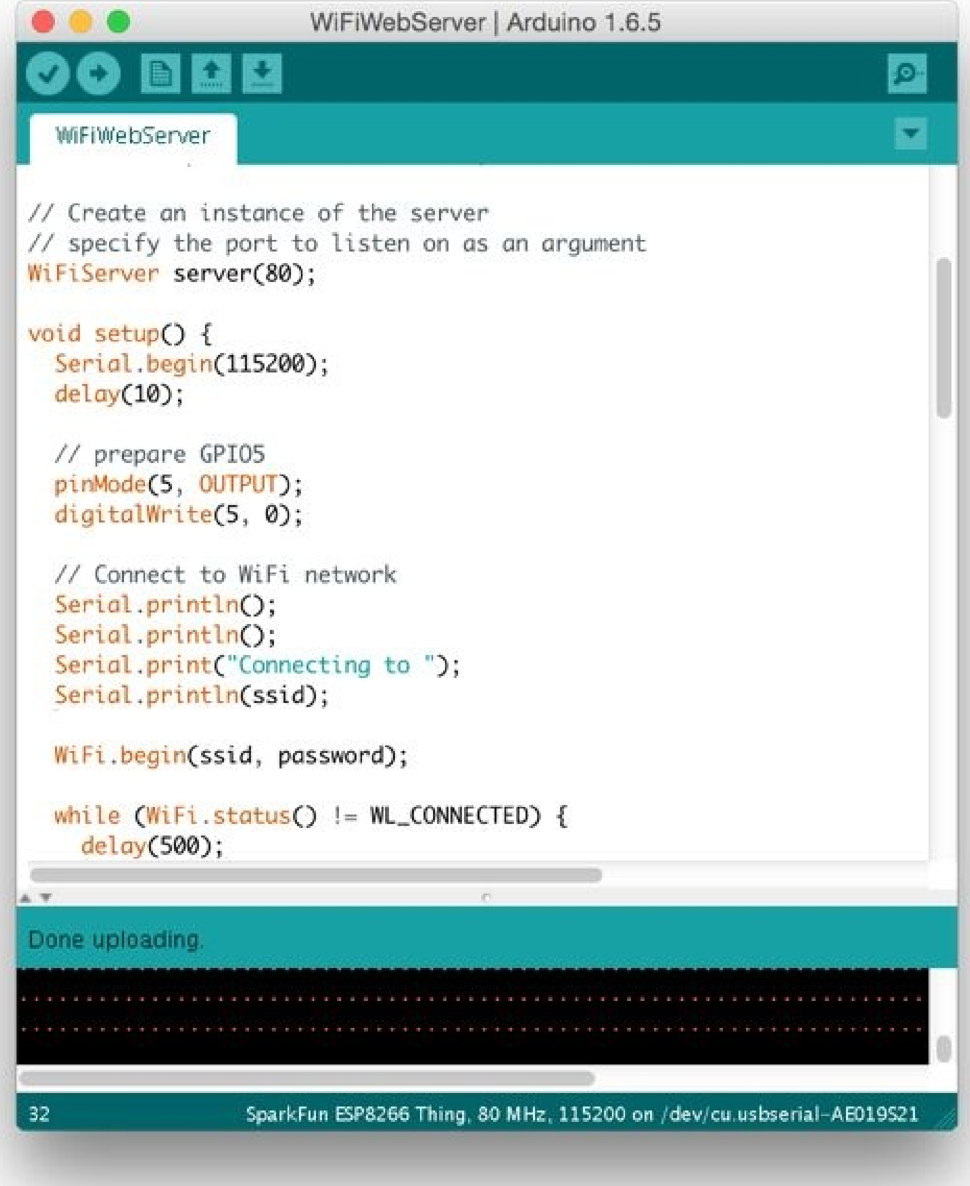
Carriage return

115200 baud

8.3 Building a Simple Internet of Things

In this section, we build a simple Internet of Things program. We will turn on/off on-board LED remotely via a browser. We can use a sample program, WiFiWebServer from ESP8266WiFi.

You should modify SSID, SSID pin and baudrate for Serial.



```
// Create an instance of the server
// specify the port to listen on as an argument
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(10);

  // prepare GPIO5
  pinMode(5, OUTPUT);
  digitalWrite(5, 0);

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
```

Done uploading.

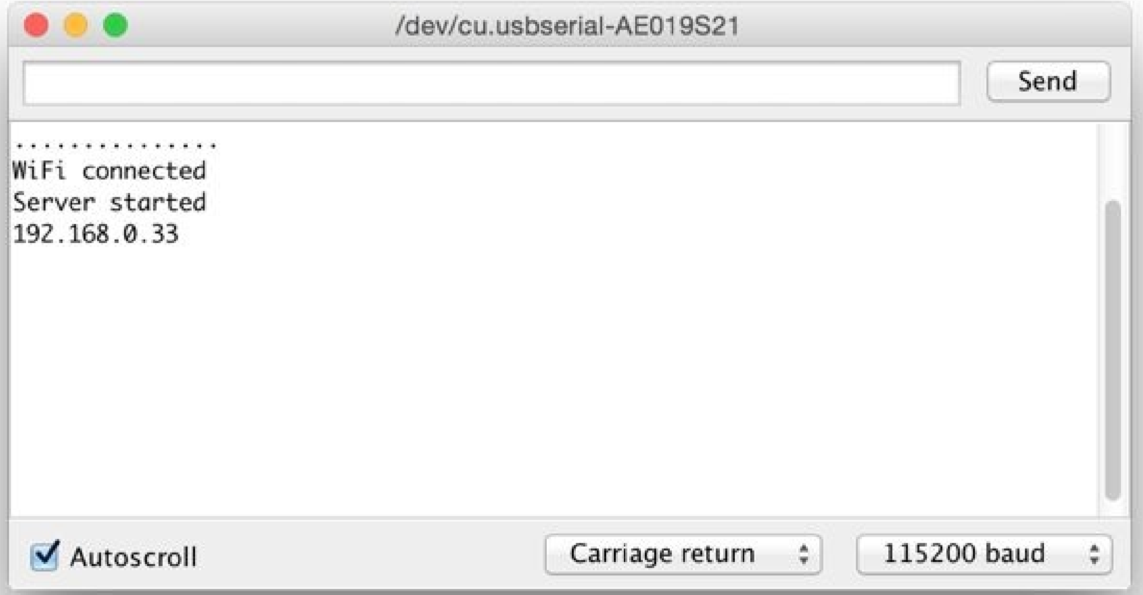
32 SparkFun ESP8266 Thing, 80 MHz, 115200 on /dev/cu.usbserial-AE019S21

The following is the complete code.

```
#include <ESP8266WiFi.h>

const char* ssid = "ssid";
const char* password = "ssid_pin";
```

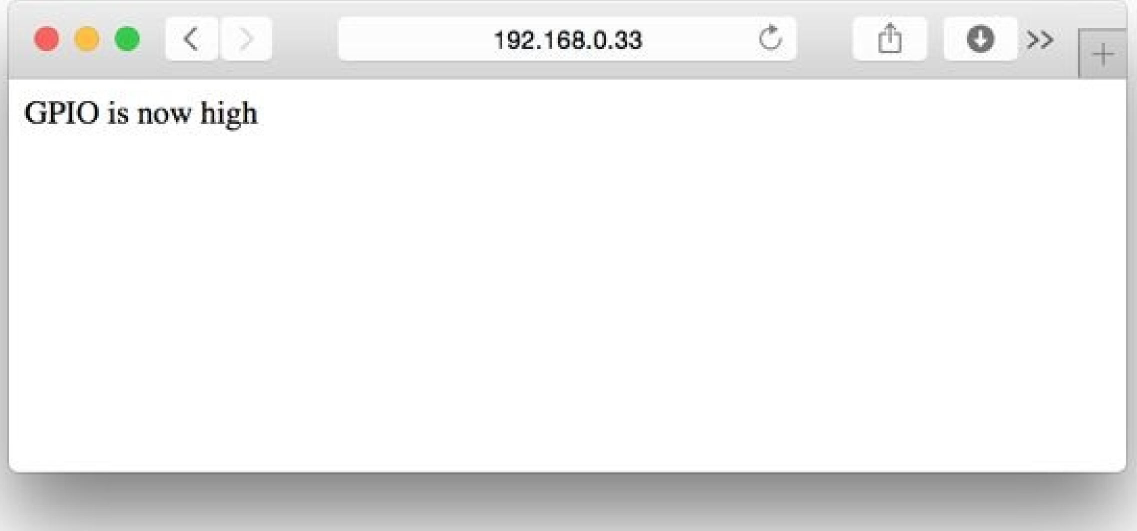
```
// Create an instance of the server
// specify the port to listen on as an argument
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(10);

  // prepare GPIO5
  pinMode(5, OUTPUT);
  digitalWrite(5, 0);

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  // Start the server
  server.begin();
  Serial.println("Server started");

  // Print the IP address
  Serial.println(WiFi.localIP());
}

void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }

  // Wait until the client sends some data
  Serial.println("new client");
  while(!client.available()){
    delay(1);
  }

  // Read the first line of the request
  String req = client.readStringUntil('\r');
  Serial.println(req);
  client.flush();
  // Match the request
  int val;
  if (req.indexOf("/gpio/0") != -1)
    val = 0;
  else if (req.indexOf("/gpio/1") != -1)
    val = 1;
  else {
    Serial.println("invalid request");
    client.stop();
    return;
  }

  // Set GPIO2 according to the request
  digitalWrite(5, val);

  client.flush();

  // Prepare the response
  String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYP
  s += (val)?"high":"low";
  s += "</html>\n";

  // Send the response to the client
  client.print(s);
  delay(1);
  Serial.println("Client disonnected");

  // The client will actually be disconnected
  // when the function returns and 'client' object is detroyed
}
```

Now you compile and upload the program into SparkFun ESP8266 Thing board. Then, open Serial Monitor. You should see IP Address of SparkFun ESP8266 Thing board.



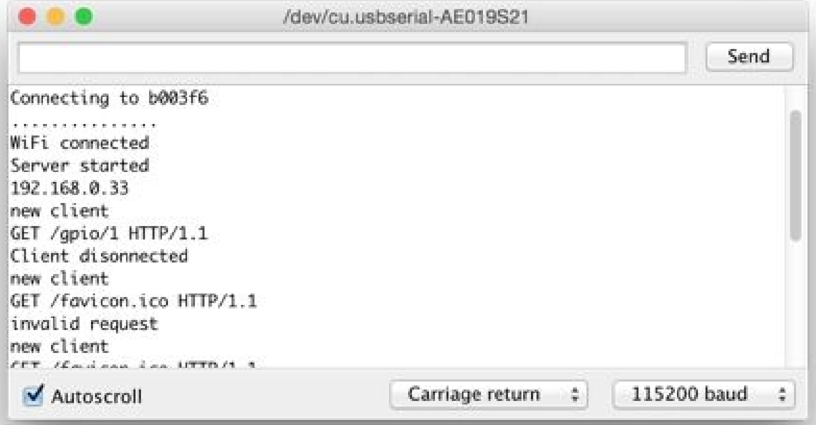
```
...............
WiFi connected
Server started
192.168.0.33
```

To test, you can open a browser and navigate to http://<board_ip>/gpio/1 to turn on LED. You should see a lighting LED on board. Otherwise, you can navigate to http://<board_ip>/gpio/0 to turn off LED.



GPIO is now high

The following is a response from Serial Monitor.



```
Connecting to b003f6
...............
WiFi connected
Server started
192.168.0.33
new client
GET /gpio/1 HTTP/1.1
Client disonnected
new client
GET /favicon.ico HTTP/1.1
invalid request
new client
GET /favicon.ico HTTP/1.1
```

☑ Autoscroll   Carriage return   115200 baud

Source Code

You can download source code on http://www.aguskurniawan.net/book/sparkesp8266_140802.zip .

Contact

If you have question related to this book, please contact me at aguskur@hotmail.com . My blog: http://blog.aguskurniawan.net