



IOT Lab

Daniel Pivonka



Who am I

- UML Computer Science 2018
 - Redhat Associate Software Engineer
 - Took this class
 - Built personal IOT devices
-
- Email: dpivonka@redhat.com
 - Phone: 978-995-5343
 - Available to meet with you for help if needed I live near campus just ask

What are we doing

- 6 labs to teach basics of the devices and software we will be using for the project
- Devices: Raspberry pi, ESP8266, various sensors



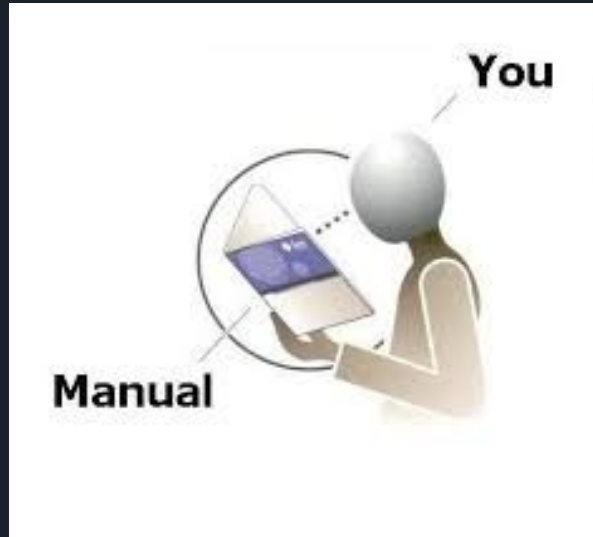
- Software: mqtt, flask, python, C
- Use what you learn in the six labs to make your own IOT device
- <https://github.com/Daniel-Pivonka/iot>



Assignment 1

Install fedora on pi and example on esp8266

Setting up Fedora for Raspberry Pi



https://fedoraproject.org/wiki/Architectures/ARM/Raspberry_Pi

(google "fedora raspberry pi" first link)

Download Fedora and prepare the sd card

Recommended Fedora image: Fedora 31 minimal aarch64

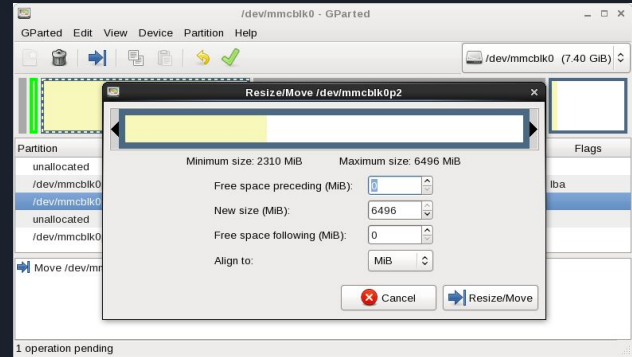
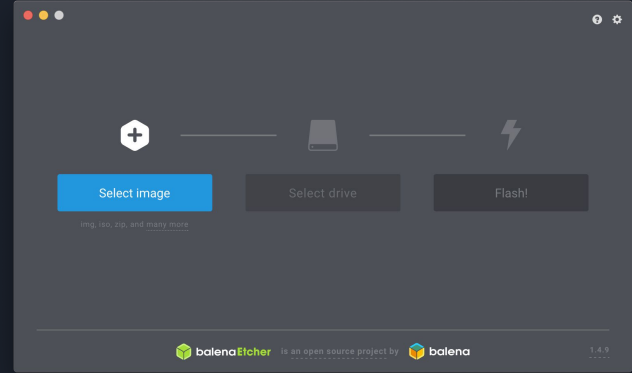
Download image to your laptop, insert sd card and, run this command to flash image to sd card

Preparing the SD card:

<https://www.balena.io/etcher/> (etcher tool to flash sd card)

```
xzcat Fedora-IMAGE-NAME.raw.xz | sudo dd status=progress  
bs=4M of=/dev/XXX (manual command to flash sd card)
```

Resize the root partition: `gparted /dev/XXX`



Setup and boot Pi

Plug in sd card, keyboard, mouse, hdmi, ethernet, and power cables

Follow on screen prompts to set up fedora root password





Run script

We have created a script that will install a desktop and browser and some of the software we will need.

The script is in the class github

Run the provided script to setup a desktop environment

```
Curl -sL http://rpi.pending.name/setup\_script.sh | /bin/bash
```




Setup the esp8266

1. Download the Arduino IDE, the latest version. I suggest doing this on your laptop not the PI (<https://www.arduino.cc/en/Main/Software>)
2. Install the IDE
3. Set up your Arduino IDE as: Go to File->Preferences and copy the URL below to get the ESP board manager extensions:
http://arduino.esp8266.com/stable/package_esp8266com_index.json Placing the http:// before the URL lets the Arduino IDE use it...otherwise it gives you a protocol error.
4. Go to Tools > Board > Board Manager> Type "esp8266" and download the Community esp8266 and install.
5. Set up your chip as:
Tools -> Board -> NodeMCU 1.0 (ESP-12E Module)
Tools -> Flash Size -> 4M (3M SPIFFS)
Tools -> CPU Frequency -> 80 Mhz
Tools -> Upload Speed -> 921600
Tools-->Port--> (whatever it is)

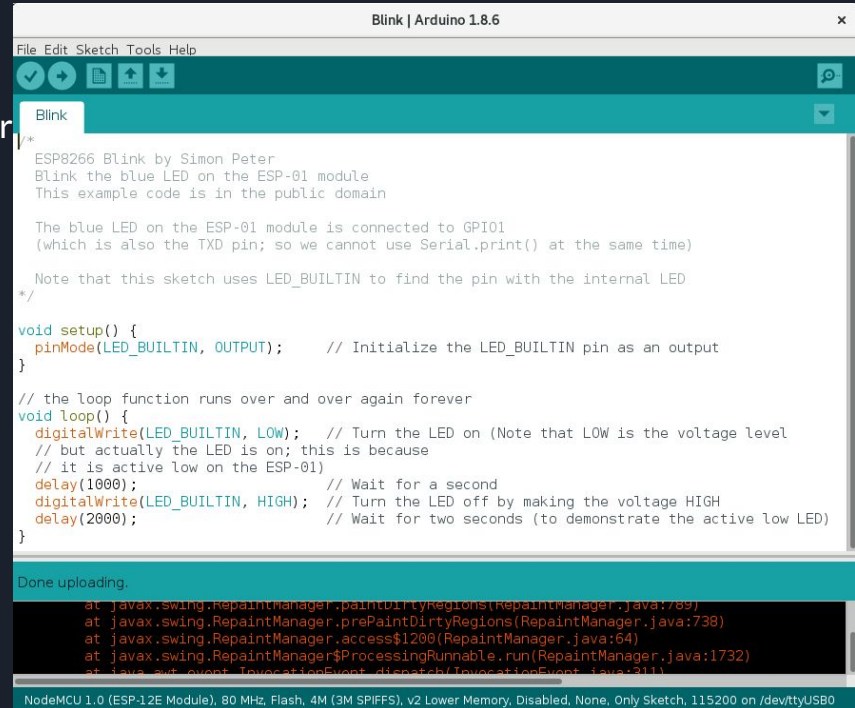
Run an example program

Plug the esp8226 into a usb port using a micro usb cable

File -> Examples -> ESP8266 -> Blink

Upload to board

A red led should blink on the board



The screenshot shows the Arduino IDE interface with the 'Blink' example code loaded. The code is for an ESP8266 module and uses the LED_BUILTIN pin. The IDE window title is 'Blink | Arduino 1.8.6'. The code includes comments explaining the setup and the loop function. The status bar at the bottom indicates the board is 'NodeMCU 1.0 (ESP-12E Module)' and shows the upload progress.

```
File Edit Sketch Tools Help
Blink
*
ESP8266 Blink by Simon Peter
Blink the blue LED on the ESP-01 module
This example code is in the public domain

The blue LED on the ESP-01 module is connected to GPIO1
(which is also the TXD pin; so we cannot use Serial.print() at the same time)

Note that this sketch uses LED_BUILTIN to find the pin with the internal LED
*/

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);    // Initialize the LED_BUILTIN pin as an output
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, LOW);  // Turn the LED on (Note that LOW is the voltage level
  // but actually the LED is on; this is because
  // it is active low on the ESP-01)
  delay(1000);                     // Wait for a second
  digitalWrite(LED_BUILTIN, HIGH); // Turn the LED off by making the voltage HIGH
  delay(2000);                     // Wait for two seconds (to demonstrate the active low LED)
}
```

Done uploading.

at javax.swing.RepaintManager.paintDirtyRegions(RepaintManager.java:789)
at javax.swing.RepaintManager.prePaintDirtyRegions(RepaintManager.java:738)
at javax.swing.RepaintManager.access\$1200(RepaintManager.java:64)
at javax.swing.RepaintManager\$ProcessingRunnable.run(RepaintManager.java:1732)
at java.awt.event.InvocationEvent.dispatch(InvocationEvent.java:311)

NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, 4M (3M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on /dev/ttyUSB0

Make a video

Take a video of both the pi running fedora and the esp8266 blinking

Email it to me before the start of next class for credit

dpivonka@redhat.com



```
dan@localhost:~  
File Edit View Search Terminal Help  
[dan@localhost ~]$ screenfetch  
/usr/bin/screenfetch: line 1341: [: =: unary operator expected  
/:-:-----:\  
:-----:\  
:-----:/shhOHbnp---:\  
/-----omMMMMNNMMMD---:\  
:-----sMMMMNNMMP---:\  
:-----:MMMdP-----:\  
:-----:MMMd-----:\  
:-----:MMMd-----:\  
:-----oNMMMMMMMMMMNho-----:\  
:--.shhhMMMMhhy++-----/  
:-----:MMMd-----:\  
:-----/MMMd-----:\  
:-----/hMMMy-----:\  
:--:dMNdhdNMMNo-----:\  
:--:sdNMMMMNds-----:\  
:-----://:-----:\  
:-----://  
[dan@localhost ~]$
```