Lab One - Environment Setup and Feather HUZZAH Introduction

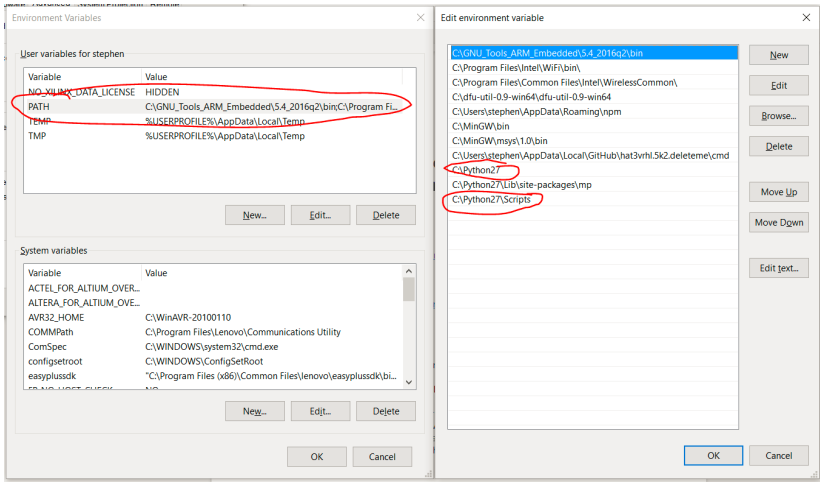
Part One: Python Setup

https://docs.python.org/3/
https://docs.python.org/3/installing/index.html
https://www.python.org/downloads/
https://pypi.python.org/pypi/pyserial
https://pypi.python.org/pypi/colorama/0.3.7
https://pypi.python.org/pypi/websocket-client

Windows:

Step 1: Download and Install Python from https://www.python.org/downloads/. You can download any Python 2 version greater than 2.7 or any Python 3 version greater than 3.4.

Step 2: Make sure your environment variables are properly configured. Navigate to **Control Panel > System and Security > System > Advanced System Settings > Environment Variables**. Check to see if the path to your Python installation folder (where python.exe is located) and the path to the Python scripts folder (where pip.exe or pip3.exe is located) is in the "Path" variable either in the user variables or the system variables. If they are not present, add the paths to the "Path" user variable; create the "Path" user variable if it does not already exist.



Step 3: To use Micropython and set up the ESP8266 Wi-Fi chip on the Feather HUZZAH development board, we need to install a few more libraries for Python: pySerial, colorama, and websocket-client. Open up a command prompt and enter the following commands to finish installation.

```
>pip install pyserial
>pip install colorama
>pip install websocket_client
```

Part Two: HUZZAH Setup

https://docs.micropython.org/en/latest/pyboard/index.html
https://learn.adafruit.com/building-and-running-micropython-on-the-esp8266/overview
http://micropython.org/download/
https://github.com/themadinventor/esptool
https://github.com/wendlers/mpfshell

Windows:

Step 1: Download the esptool through https://github.com/themadinventor/esptool. Download the Micropython firmware for ESP8266 through http://micropython.org/download/.

Step 2: Flash the Micropython firmware onto the HUZZAH board.

Connect the HUZZAH board to your machine and find out which serial port it is connected to. You can either do this by looking under the Ports tab in Device Manager or through the command line by changing directors to **C:\WINDOWS\system32** and using the “mode” command.

Once you know the serial port, navigate to the folder through the command line which you extracted the esptool files to (should have esptool.py inside). Enter the following command to flash the firmware onto the board.

```
>python esptool.py --port SERIAL_PORT --baud 460800 write_flash --flash_size=32m 0
FIRMWARE.bin
```

If there are any issues with the firmware, you can always reprogram the device after erasing the flash with the following command.

```
>esptool.py --port /dev/ttyUSB0 erase_flash
```

Now that the firmware is set up on the HUZZAH, you can now connect to it through serial using a terminal program (mobaXterm, putty, termite, etc.) with the following settings:

```
Baudrate = 115200
Data bits = 8
Stop bits = 1
Parity = None
Flow Control = XON/XOFF
```

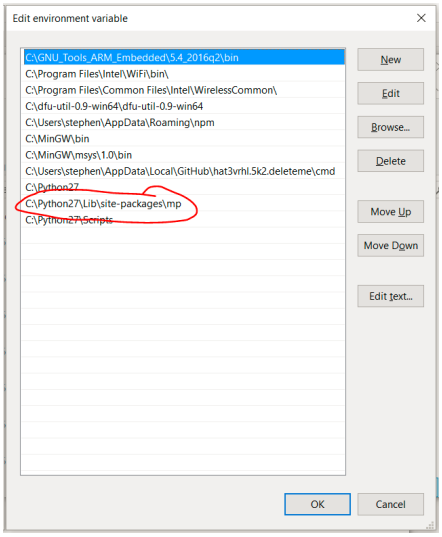
You can now access the python command line on the ESP8266 chip. The Micropython library has some, not all, of the same functionalities as Python 3, as well as some board specific capabilities. More information can be found by looking through the documentation.

Step 3: Install mpfshell, which will be used to access the internal file system on the ESP8266 and transfer files to and from the board.

Download the files through https://github.com/wendlers/mpfshell. Run the following command to install mpfshell.

```
>python setup.py install
```

After installation, make sure to include the path to the “mp” folder (includes mpfshell.py) in the user “Path” variable; This folder should be located within the Python installation folder.



Now you can use the “mpfshell” command in the command line to connect, view, and transfer files to and from the HUZZAH board.

MAC or Linux:

The process for using the Huzzah is relatively easier for mac and Linux as they come with python. We first install pip (Python package manager) to install dependencies and tools. We then download the micropython firmware and flash it on to the board. We also install a small command line tool which makes it easy to upload programs and interact with the Huzzah/

Step 1: If you already have pip installed skip to Step 2. Install pip using this script (get-pip.py). Download the linked file to your computer. Open your terminal app and navigate to the folder where you downloaded it and run

```
python get-pip.py
```

Step 2: Install the esptool (this is used for flashing microptyhon).

```
>sudo pip install esptool
```

Step 3: Download the micropython firmware and flash it on to the Feather Huzzah board. Download the latest version of micropython here. Download the bin file listed under the ESP8266 section and save it. On your terminal navigate to the location of the bin file you downloaded and run the following commands.

The port for the Huzzah is usually /dev/tty.SLAB_USBtoUART on the mac. To find out the port at which the Huzzah is mounted at, open up terminal and type

```
>ls -l /dev/tty.*
```

This lists all the ports to which devices are connected to. Scroll to the top and note the port which says USB/UART etc.

```
>esptool.py --port /dev/tty.SLAB_USBtoUART erase_flash
>esptool.py --port /dev/tty.SLAB_USBtoUART --baud 460800 write_flash --flash_size=32m -fm
dio 0 <name of bin file downlaoded>.bin
```

This flashes micropython to the board and once the process completes, you are all set.

Step 4: To make things easy for programming the board, we are going to use another tool called mpfshell. Download mpfshell utility here. Follow the Installation Instructions on the page to install mpfshell. Please read the documentation on the link to use mpfshell for uploading programs to the Huzzah.

Part Three: Blink LEDs

Now that you have the environment set up to be able to code and run programs on your HUZZAH, we will now write our first few programs. These exercises hopefully can get you familiar with the ESP8266 environment, which will form the basis of the smartwatch. Specifically:

1. Write a program to blink an LED to encode “SOS” in morse code. In morse code, each letter is represented by a series of dots (short tones) and dashes (long tones). You should replicate this on your LED. For dots, the LED should blink at around twice the speed of dashes to represent the string.
2. Write a program to blink three separate LEDs simultaneously (you can use some of the on board LEDs too) at three different rates (e.g. blink LED 1 at 100 ms, LED 2 at 500 ms, LED 3 at 1000 ms). The stipulation is that you are not allowed to use any interrupts or timers, and that the blinking rates should be discernible to the human eye.

Group Members:

EECS 4764 Lab One Checkpoints:

1. Continuously blink a morse code message through the HUZZAH using an LED.

2. Simultaneously blink three separate LEDs at different rates without using interrupts or timers.