The goal of this assignment is to get used to using general purpose IO (GPIO). **Keep all your work in a github repository.**

## Buttons and LEDs

The goal here is to get some LEDs and buttons wired up and tested before doing the next part.

1. Wire up your breadboard to have 4 buttons. Each is to have one terminal attached to +3.3V and the other to a GPIO port.
2. Also wire up 4 LEDs with current limiting resistors. Tie the *plus* side of the LED to the GPIO port and run the *minus* to the resistor then to ground.
3. Write a simple program that reads the switches and lights a corresponding LED.

## Measuring a gpio pin on an Oscilloscope

Get an oscilloscope so you can measure the output of your gpio pins. Run

bone$ **git clone** [**https://github.com/beagleboard/vsx-examples**](https://github.com/beagleboard/vsx-examples)

bone$ **cd vsx-examples/examples/BeagleBone/Black**

bone$ **./blinkLED.sh**

and answer the following questions about gpio measurements.

1. What's the min and max voltage?
2. What period and frequency is it?
3. Run **htop** and see how much processor you are using.
4. Try different values for the sleep time. What's the shortest period you can get? Make a table of the fastest values you try and the corresponding period and processor usage. Try using markdown tables: <https://www.markdownguide.org/extended-syntax/#tables>
5. How stable is the period?
6. Try launching something like **vi**. How stable is the period?
7. Try cleaning up blinkLED.sh and removing unneeded lines. Does it impact the period?
8. What's the shortest period you can get?

### Python

Modify **blinkLED.py** to toggle a gpio pin as fast as possible.

1. What period and frequency is it?
2. Run **htop** and see how much processor you are using.
3. Present the shell script and Python script results in a table for easy comparison.

### C

Repeat the above using C **(blinkLED.c**)**.** Add your results to the table.

## gpiod

gpiod is the new gpio pins that replaces /sys/class/gpio. There are several gpiod examples in /opt/vsx-examples/examples/BeagleBone/Black/gpiod.

bone$ **cd ~/vsx-examples/examples/BeagleBone/Black/gpiod**

bone$ **ls**

bulk\_blink.py getsetEvent.py toggle1.c toggleLED.c

get.c getset.py toggle1.py toggleLED.py

get.py get.sh toggle1.sh

getset.c Makefile toggle2.c

getsetEvent.c ReadMe.md toggle2.py

Use the toggle1 examples measure how fast you can toggle one gpio bit using c and python. Repeat the exercise using two bits. Add the results to your table.

## getsetEvent.py

Modify **getsetEvent.py** to read your four buttons and turn on the corresponding LED.

## ~~Security (optinal)~~

* + - 1. ~~Change ssh port number from 22 to 2022.~~
      2. ~~Setup iptables to only accept ssh connections from on campus. Show output from i~~**~~ptables -L~~**
      3. ~~Use fail2ban to reject ssh connections for 15 seconds after 2 failed attempts.~~

## Etch-a-sketch

Next write modify your Etch-a-sketch[1] program to be controlled by the pushbuttons using gpiod For now, just print the grid in the terminal window. Next week we’ll interface it to the LED grid.

0 1 2 3 4 5 6 7

0: x x x x

1: x x

2: x x x x

3: x x

4: x x x x

5: x x x x

6: x x

7: x x x x

My example shows an 8 by 8 grid. Make yours able to do any size.

Extras

Here are some other things you can do to make the project more interesting. Feel free to think of your own ideas.

* Add a button to clear the display, or toggle between writing and erasing.

## What to turn in

1. Create a repository on <https://github.com/>

2. List your repository on the Embedded Repos link on Moodle.

3. Make a subdirectory in your github repository called **hw02**.

4. Put all your files in the directory, include a **ReadMe.md** that explains what your homework does and how to use it. Use Mark Down to format ReadMe.md and your tables.

5. Document your code.

[1]<http://en.wikipedia.org/wiki/Etch_A_Sketch>