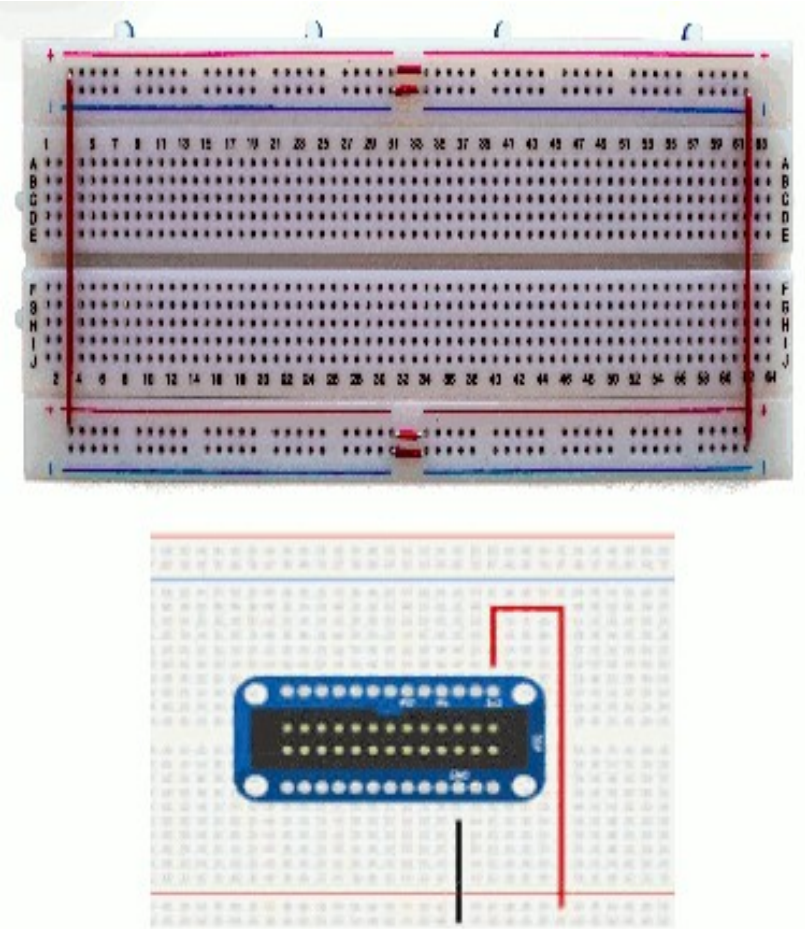
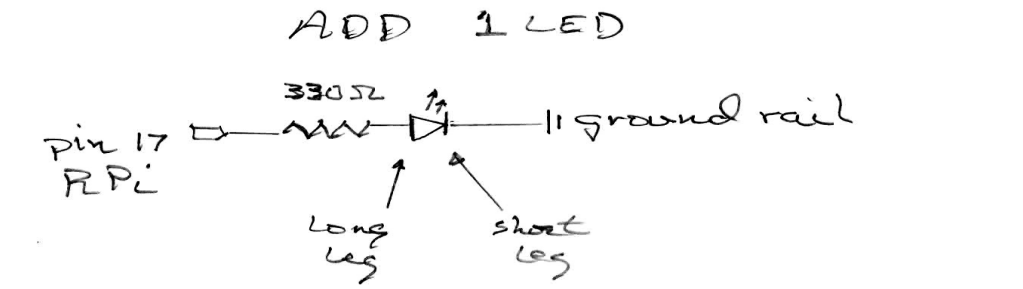
WebIOPi

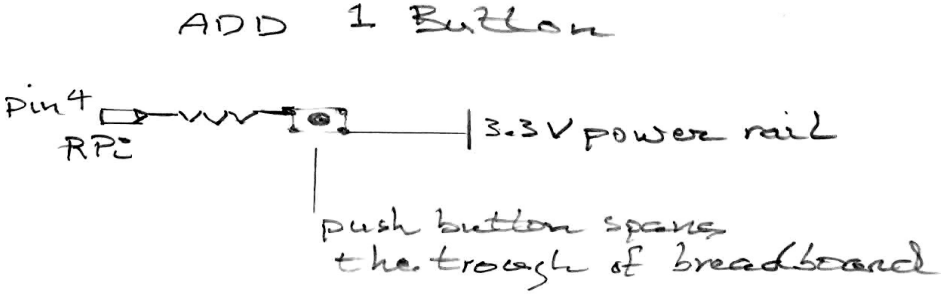
In-class exercise

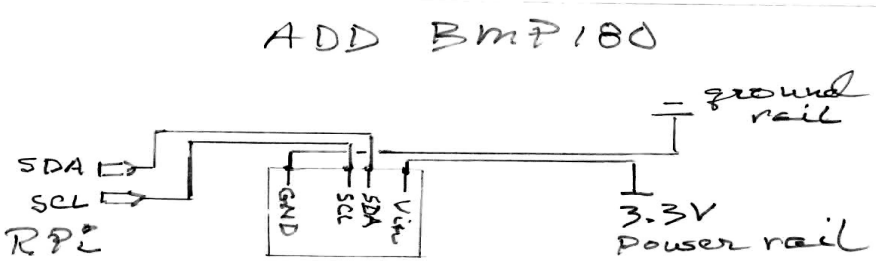
Hardware Setup

- Start with the breadboard configuration used on Monday:









In-class GPIO Exercise

- Modify the index.html in the WebIOIPi-0.7.0/examples/scripts/simple directory to contain one button that drives the LED on pin #17 (toggles between high & low with each button press).
  - Configure the server to test your index.html file
- Configure the server to run the python script and index.html file in the WebIOIPi-0.7.0/examples/scripts/macros directory
  - Make all buttons control the LED on pin 17; make the switch be connected to pin 4
    - Try to understand how each button control works
  - Modify script.py and index.html to configure pin 17 as PWM and have the 0-100% duty cycle slider control pin 17
    - Remove the blink setup in script.py's loop function
    - Play with the slider interface

In-Class Exercise Recap

- When do you need to use a python script?
  - Read pins
  - Work with devices
  - Set pins to PWN mode
  - Complex pin control
- Use macros to introduce new functionality to the API
  - Expedites comunication between client and server

Device Setup

- Make sure that you have WebIOPi installed as described on Monday of last week:
  - Install WebIOPi on your Raspberry Pi
    - Download the tar archive file:

      wget www.cs.unca.edu/~bruce/Fall14/WebIOPi-0.7.0.tar.gz
    - Uncompress:

      tar xvfz WebIOPi-0.7.0.tar.gz
    - Change directory to new WebIO folder:

      cd WebIOPi-0.7.0
    - Run setup shell script:

      sudo ./setup.sh

Devices

- WebIOPi device tutorial

- Another example
  - BMP180 demo presented at the start of semester
  - Download the code:

    wget ***www.cs.unca.edu/~bruce/Fall14/demo1.tar.gz***
  - Move the archive file to your *examples* directory and unpack:

    mv demo1.tar.gz ……

    tar xvfz demo1.tar.gz

Misc (hopefully) helpful info

- How to search the source code:

  find . -type f -exec grep SEARCHSTRING {} \; -print

  or

  find . -name “.js” -exec...*

- Use firebug to debug

- Setting the class of an html element:

  button = webiopi().createButton("m1", "Temp",callMacroTemp);

  content.append(button); // append button to content div

  webiopi().setClass("m1", "TempButt"); //set element class

In-class Device Exercise

- Connect the TSL-2561 Lux sensor to your Pi and develop an WebIOPi-based webpage to display the light readings.
- Show me your work before you leave.