

## Contents

Learn to use the Linux command line, C++, and Python .....	3
Things you need to know .....	3
Two Flavors/Distros we will be using.....	3
Things you need to do.....	4
Lab 1 C++ program on taz (20 points).....	4
Lab 2 Python Program on taz (20 points).....	5
Raspberry Pi - Getting Started .....	5
Things you need .....	5
Things to do.....	5
Lab 3 - Setup Raspberry Pi (20 points) .....	5
Raspberry Pi - X Window System .....	8
Things you need to know .....	8
Things you need to do.....	8
Lab 4 – X Window System (5 points).....	8
Raspberry Pi - First Programs.....	8
Lab 5 – C++ program on the Raspberry Pi (15 points) .....	8
Lab 6 – Python program on the Raspberry Pi (15 points).....	9
Raspberry Pi - First GPIO Programs.....	10
Things you need to know .....	10
Things you need to do.....	11
Lab 7 – GPIO LED Lab (30 points).....	11
Lab 8 – GPIO PWM & LED (10 points).....	13
Lab 9 – GPIO PWM using RPIO (10 points) .....	14
Raspberry Pi - GPIO Input .....	14
Things you need to know .....	14
Things you need to do.....	16

Lab 10 – Morse Code reader .....	16
Final Project Proposal .....	18
Other Useful Links.....	19

# Learn to use the Linux command line, C++, and Python

## Things you need to know

### Two Flavors/Distros we will be using.

1. CentOS release 6.4 (Final) on Taz.harding.edu
2. Raspbian 3.10.25 - a modification of Debian

### Command line programs

- ls or ls
- cp – copy a file to another location
- mv – rename a file
- cd – change directory/folder
- mkdir – create a directory/folder
- rm – remove file (rm -r )
- rmdir – remove directory
- pwd – print working directory
- cat – shows contents of a text file
- man/info – to get help on a program
- clear – clears the screen
- nano – text editor
- sudo – run a program as the root (super user)
- vi – text editor
- command line history using up and down arrows
- ./ to use a program in the current directory
- c++

- command line arguments

- argv
- argc

- g++ example.cpp -o example

- check to see if a file exists

```
bool fileExists(char filename[])
{
    bool result = true;
    ifstream fin(filename);
    if(fin.fail())
    {
        result = false;
    }
    fin.close();
    return result;
}
```

- python

- python example.py - to run the program... no need to compile
- #comment
- no semicolons and no {}
- if else syntax
- use tabbing to indicate lines of code that belong to control structures like if
- sys.argv[n], len(sys.argv)
- infile = open('filename') - file to read from
- outfile = open('outfilename', 'w') file to write to
- infile.close(), outfile.close() - close them when you are done
- print 'text to print'
- os.path.isfile(filename) – to see if a file exists

## Things you need to do

### Lab 1 C++ program on taz (20 points)

Write a c++ program that will work similarly to the cp program.

#### Details

- Create a directory in your home (~) named comp170
- Create a directory in your comp170 directory named filecopy
- Inside the filecopy directory make a file named filecopy.cpp (5 points)
- Your program should take two additional command line arguments besides the program name (remember to use argc and argv for this) (5 points)
- Note: have an "int main" instead of "void main" to make the compiler happy
- Use the fstream to copy a file designated by argv[1] into a file designated by argv[2] (5 points)
- Display a "usage" statement as follows when invalid number of arguments are entered or if the from file does not exist (5 points)
  - "source file does not exist"
  - "useage: filecopy source destination"
- Compile your program: g++ filecopy.cpp -o filecopy
- Run your program to test it: ./filecopy filecopy.cpp copy.cpp
- Make sure copy.cpp has the right contents
- Make sure it displays the usage statement for the proper cases.
- Make a 2 minute or less YouTube video that contains the following
  - A view of all of your code. It should fit on one screen.
  - An example of running the program successfully copying a text file
    - Use ls to show that it worked
  - An example of running the program with an invalid number of arguments to get the usage statement to display
  - An example of running the program with an invalid source file (one that does not exist)

- Add the URL of your uploaded YouTube video to a cppFileCopy.txt file and upload it to Easel. Include your full name in the txt file.
- Example YouTube video (with code view missing because that is your job)



[http://youtu.be/akr\\_C9Tbf6k](http://youtu.be/akr_C9Tbf6k)

### Lab 2 Python Program on taz (20 points)

Redo lab 1 using Python (20 points), same requirements as lab 1

- Add the URL of your uploaded YouTube video to a pyFileCopy.txt file and upload it to Easel. Include your full name in the txt file.
- Example YouTube video (with code view missing because that is your job)



<http://youtu.be/Ne7X4fygkQU>

## Raspberry Pi - Getting Started

### Things you need

- Raspberry Pi Box
- Keyboard
- Mouse
- Monitor (not supplied)
- An Ethernet internet connection
- A wireless network to connect to

### Things to do

#### Lab 3 - Setup Raspberry Pi (20 points)

- Open up the protective case
  - Remove two small black tabs that will give access to the Ethernet and USB ports.
- Remove the Raspberry Pi from the small white box with "CanaKit" label on the front.
- Remove ribbon cable from the clear plastic package and push it onto the Raspberry Pi board. The pink wire on the ribbon should be furthest from the yellow VGA port on the board and the ribbon should lay flat out of the case as shown in the photo. If it will not lay flat like this you have the wrong end of the ribbon.



- Place Pi in the case
  - You should not have to push hard to close it
  - Screw in the 4 screws provided, do not over tighten
- Remove SD card from package and insert it into the Raspberry Pi
  - Insert the SD card into port near the usb power port
  - The Raspberry logo should be face down
- Plug your keyboard into one of the USB ports
- Plug your mouse into the USP port on the keyboard
- Plug your HDMI cable into the Pi and your monitor or TV
- Remove the USB power supply from the package and use it to power the Pi
- When it boots
  - Select “Raspbian [Recommended]” by pressing enter. A warning will appear to indicate that all data on the SD card will be overridden. Choose “Yes” to override.
  - Read all of the messages that appear while the Operating System is installing. Define the following in a text file named RaspbianInstall.txt. You will upload this file to Easel. (10 points)
    - For example the default username and password
    - What is an Operating system
    - raspi-config
      - startx
    - Command Line
  - Click ok when it is done. It will boot.
  - A configuration screen will appear. Press the tab key and choose “Finish”
  - Change the keyboard layout to us
    - \$sudo nano /etc/default/keyboard
    - change XKBLAYOUT=”gb” to XKBLAYOUT=”us”
    - save and exit
    - \$sudo reboot -n           to reboot
  - Update the Raspbian OS (with a Ethernet cable plugged in)
    - \$sudo apt-get update
    - \$sudo apt-get upgrade

- select y to continue
  - Install a utility to choose network settings (with a Ethernet cable plugged in)
    - sudo apt-get install wicd-curses
      - select y to continue
    - shutdown the Pi ... sudo shutdown -h -t 0 now
    - plugin the wireless adapter
    - reboot - unplug the power and plug it back in
    - login
    - select a wlan
      - \$sudo wicd-curses
      - If wireless networks do not appear you may need to do the following inside of wicd-curses
        - Select P for preferences
        - Type wlan0 into the wireless interface preference
        - Press F10 to accept
        - Press R to Refresh
      - choose a network with a good signal strength
        - use the down arrow to highlight a network
        - press enter and wait to select the network
        - press Q to quit
      - \$ifconfig
        - Verify that the wlan0 entry has an ip address listed
- Make a 2 minute or less YouTube video that contains the following (10).
  - A view of your assembled Raspberry Pi running and a command prompt showing on the monitor
  - A view of the results of the following programs run from the command prompt
    - \$cat /etc/default/keyboard
    - \$cat /etc/\*-release
    - \$ifconfig
- Add the URL of your uploaded video to the RaspbianInstall.txt file you will upload to Easel. Include your full name and your partner's full name in the txt file.
- Example YouTube video



<http://youtu.be/7NxdrlIBtLc>

## Raspberry Pi - X Window System

### Things you need to know

- What is an X Window System – see [http://en.wikipedia.org/wiki/X\\_Window\\_System](http://en.wikipedia.org/wiki/X_Window_System)
- The x Window system is started using the startx program from the command prompt.
- Start programs in the x Windowing system like you do in MS windows
- Multiple desktops are available to organize your work. Toggle between them using the utility in the tray at the bottom or use the scroll wheel.

### Things you need to do

#### Lab 4 – X Window System (5 points)

- \$startx
  - X Windows should start
- Start a couple of programs on each desktop
- Toggle between the desktops
- Make a very brief YouTube video that shows that you have completed this lab
  - xWindows already running
  - Programs like terminal windows, web browsers, and file managers should be started
- Add the URL of your YouTube video to an xWindow.txt file and upload to Easel. Include your full name and your partner's full name in the txt file.
- Example YouTube video



<http://youtu.be/cu42qqvQFic>

## Raspberry Pi - First Programs

#### Lab 5 – C++ program on the Raspberry Pi (15 points)

You are going to write a program that will allow you to record a note from the command line. We will save the notes to a text file named notes.txt that will be saved on the desktop (/home/pi/Desktop).

- With xWindows running start a LXTerminal
- Create a file named notes.txt on the desktop by right clicking on the desktop and selecting “Create New”, “Blank File”. Name it notes.txt.
- Create a directory in your home (~) named comp170
- Create a directory in the comp170 directory named notes
- Create your source code file named notes.cpp within the comp170 folder
- Write a c++ program within notes.cpp that will accept a command line argument as a note.



- The current date and time and note should be appended it to the end of the file at /home/pi/Desktop/notes.txt.
- To get the current date time as a c-style string include `ctime` and `string.h` and do the following:
 

```
time_t now = time(NULL);
char timeString[50];
strcpy(timeString, ctime(&now));
timeString[strlen(timeString)-1] = '\\0'; //remove the \\n from end
```
- Each note entry should be on a single line
- If the user does not pass in a note the program should report:
  - "Useage: notes \"your note\\\"\\n\";
- Compile the program to notes
  - `g++ notes.cpp -o notes`
- Test the program
  - `./notes "This is an example note. If this were a real note it would be important."`
- Example notes.txt contents:
 

```
Wed Mar 12 21:59:46 2014: This is a test2
Wed Mar 12 21:59:49 2014: This is a test3
```
- Make a 2 minute or less YouTube video that shows the following:
  - A quick view of your notes.cpp source code (5 points)
  - An example execution of your notes program (5 points)
  - A view of notes.txt file to show that the note was added (5 points)
- Add the URL of your YouTube video to a cppNotes.txt file and upload to Easel. Include your full name and your partner's full name in the txt file.
- Example YouTube video (without code view that you should have)



<http://youtu.be/29HFA3zGHQI>

## Lab 6 – Python program on the Raspberry Pi (15 points)

Redo lab 5 using Python

- Example notes.txt contents:
 

```
Wed Mar 12 21:59:46 2014: This is a test2
Wed Mar 12 21:59:49 2014: This is a test3
Thu Mar 13 00:38:15 2014: example py note
Thu Mar 13 00:38:38 2014: example py note 2
```
- To append to the file open it as follows:
 

```
outfile = open('/home/pi/Desktop/notes.txt', 'a')
```
- To get the current date time as a string import time and do the following:
 

```
outfile.write(time.strftime("%c"))
```
- Change the useage statement to:
 

```
print 'useage: python notes.py "example note\\n'
```
- Make a 2 minute or less YouTube video that shows the following:
  - A quick view of your notes.py source code (5 points)

- An example execution of your `notes.py` program (5 points)
- A view of `notes.txt` file to show that the note was added (5 points)
- Add the URL of your YouTube video to a `pyNotes.txt` file and upload to Easel. Include your full name and your partner's full name in the txt file.
- Example YouTube video (without code view that you should have)



<http://youtu.be/cu42qqvQFjc>

## Raspberry Pi - First GPIO Programs

### Things you need to know

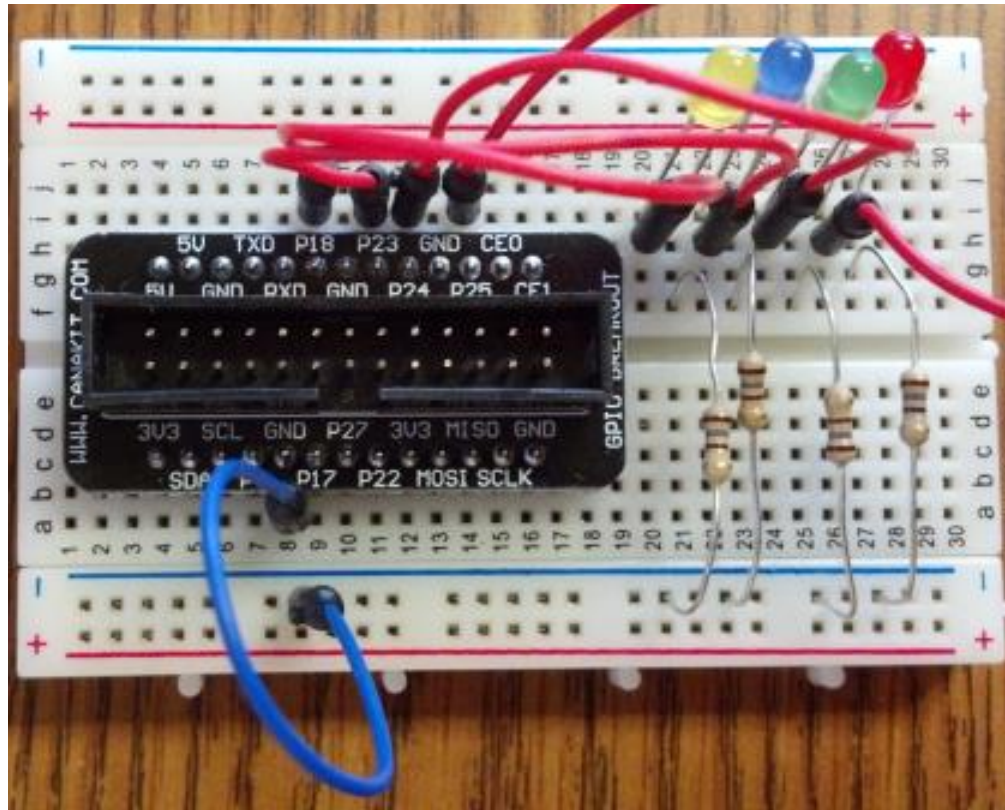
- GPIO Pins – General Purpose Input Output Pins - “GPIO Pins” is a broad term referring to all of the pins on the Raspberry Pi. The pins can really server several purposes.
  - 5V pin
  - 3.3V pin
  - Ground (GND) pin
  - GPIO pins can be turned on and off to control things like LEDs
  - I2C interface pins – <http://en.wikipedia.org/wiki/I%C2%B2C>
  - SPI interface pins – [http://en.wikipedia.org/wiki/Serial\\_Peripheral\\_Interface\\_Bus](http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus)
    - Note: Useful I2C and SPI introduction - <http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/?/article/AA-00255/22/Introduction-to-SPI-and-IC-protocols.html>
  - Serial Rx pin
  - Serial Tx pin
- Adafruit GPIO Breakout – Connects between ribbon from Raspberry Pi and Breadboard
  - Pin layout is changed. Refer to labels on the side on the Adafruit
- Bread board – a prototyping board for circuits
  - On the bread board that comes with the canakit
    - The outer + and – columns are connected
    - Rows are connected in sections a-e
    - Rows are connected in sections f-j
- Python
  - For loop syntax
    - for in in range(1,5):
  - Convert an integer to a string
    - `str(i)`
    - `s = “Red at ”+ str(i) + “%”`
  - Using a GPIO PWM object to create a software based PWM

- GPIO.setup(PIN#, GPIO.OUT)
- pin = GPIO.PWM(PIN#, Frequency in Hertz )
- pin.start(%Duty cycle)
  - parameter is the percentage of the cycle the signal is on
  - pin.ChangeDutyCycle(%Duty cycle)
  - pin.stop() #turn off signal
- Using servo object from RPIO library
  - from RPIO import PWM # add this to the top of your program
  - Servo = PWM.Servo() #create servo object
  - servo.set\_servo(pin# ,cycle time in mico seconds)
    - Cycle times 0 - 2000 worked well when using LEDs

## Things you need to do

### Lab 7 - GPIO LED Lab (30 points)

- Setup board (15 points)
  - Connect the Adafruit Pi Cobbler to the bread board
    - Push the pins straight down into the bread board. You will need to press firmly on each side so that the pins go all the way in.
    - In the image below the left most pin on the Pi Cobbler is on row 4 of the bread board
  - Connect a wire from the GRD pin on the Pi Cobbler to the Ground/Blue (-) column of the bread board
  - Connect wires to pins 18, 23, 24, and 25 of the Pi Cobbler to rows 20, 22, 24 and 26 on the bread board. I used column h.
  - Connect the LEDs
    - If you connect the LEDs backward they will burn out. **Avoid this.**
    - Yellow rows 20 and 21 on column i, long lead on row 20 (from the Cobbler)
    - Blue rows 22 and 23 on column i, long lead on row 22 (from the Cobbler)
    - Green rows 24 and 25 on column i, long lead on row 24 (from the Cobbler)
    - Red rows 26 and 27 on column i, long lead on row 26 (from the Cobbler)
  - Connect the resisters (brown, blue, brown, gold) 160 ohms 5% tolerance
    - The resisters should be connected from rows 21, 23, 25, and 27 on the LED side of the board to any spot on the blue ground column on the lower side of the board.
    - Be sure the resisters do not touch
    - Direction of resister does not matter



- Write a program to turn on the 4 LEDs a half sec each, one at a time, 5 times. They should be turned on in the order yellow, blue, green, red (repeated 5 times). Name your source file fourleds.py . (15 points)
  - Start with this code that turns on 1 LED for 3 seconds

```
import RPi.GPIO as GPIO
import time

GPIO.cleanup()

GPIO.setmode(GPIO.BCM)

YELLOW = 18

GPIO.setup(YELLOW, GPIO.OUT)

GPIO.output(YELLOW, 1)
time.sleep(3)
GPIO.output(YELLOW, 0)

GPIO.cleanup()
```

- Add a loop that iterates 5 times
- Change the sleep from 3 sec to .5
- Add 3 more variables for the other three pins

- Add GPIO output statements for the other three pins
- Run your program
- Make a 2 minute or less YouTube video that shows the following:
  - A view of your completed circuit board
  - A view of your `fourleds.py` source code
  - An example execution of your `fourleds.py` program
- Add the URL of your YouTube video to a `pyGPIO.txt` file and upload to Easel. Include your full name and your partner's full name in the txt file.
- Example YouTube run of example program a 4 LED program



<http://youtu.be/6M2T7UU6Ct4>

### Lab 8 – GPIO PWM & LED (10 points)

In this lab you will make the red LED start dim and brighten over a series of steps.

- Keep the same break board configuration as Lab 7
- Copy your code from Lab 7 to start with
  - `cp fourleds.py fourledspwm.py`
- Set pin 25 (the RED LED) to use PWM (see code in “Things you need to know”)
- Remove references to other pins
- Start the PCM on the red pin to a duty cycle of 0
- Change the loop to iterate 20 times
- Sleep for .5 sec each time through the loop
- Change the duty cycle of the red pin to 1.75 times your looping index each time through the loop
- The console output of the program should look like this:
 

```
Red at 1.75%
Red at 3.5%
... Continue sequence here
Red at 31.5%
Red at 33.25%
Red at 35.0%
```
- Make a 1 minute or less YouTube video that shows the following:
  - A view of your `ledpwm.py` source code
  - An example execution of your `ledpwm.py` program
- Add the URL of your YouTube video to a `pyPWM.txt` file and upload to Easel. Include your full name and your partner's full name in the txt file.
- Example YouTube run of example program a PWM LED program. It is difficult to tell that the red led gets brighter in the video, but it does.



<http://youtu.be/-P8p-0VBtu8>

## Lab 9 – GPIO PWM using RPIO (10 points)

- Install RPIO - We will use this for its simple PWM tools
  - `$sudo apt-get install python-setuptools`
    - Choose y when prompted
    - This will take a minute
  - `$sudo easy_install -U RPIO`
- Keep the same bread board configuration as Lab 7 & 8
- Create a source file named `fourledRPIO.py`
- Have a loop that iterates 10 times
  - Use a single servo object to set the power to the 4 LEDs to random values each time through the loop. See notes in “Things you need to know” to learn how to use the servo object
  - sleeping 1 second each time through the loop
- Make a 1 minute or less YouTube video that shows the following:
  - A view of your `fourledRPIO.py` source code
  - An example execution of your `fourledRPIO.py` program
- Add the URL of your YouTube video to a `pyRPIO.txt` file and upload to Easel. Include your full name and your partner’s full name in the txt file.
- Example YouTube run of example program the RPIO 4 LED program



<http://youtu.be/cP2UPNnOU2s>

## Raspberry Pi - GPIO Input

### Things you need to know

Example of a button push program using the Raspberry Pi.

```
# Link to circuit Diagram
#
https://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/robot/buttons\_and\_switches/
import RPi.GPIO as GPIO
import time

GPIO.cleanup()

GPIO.setmode(GPIO.BCM)

GPIO.setup(25, GPIO.IN)

buttonDown = False

while True:
    time.sleep(.1)
    input = GPIO.input(25)
    if input:
        if not buttonDown:
            buttonDown = True
            print("button pushed")
            startTime = time.time()
    else:
        if buttonDown:
            elapsedTime = time.time() - startTime
            print(elapsedTime)
            buttonDown = False

GPIO.cleanup()
```

The Python dict type will be useful for the Morse Code assignment. See, <https://docs.python.org/2/library/stdtypes.html#dict> for more information.

```
states =
{'AL': 'Alabama', 'AK': 'Alaska', 'AZ': 'Arizona', 'AR': 'Arkansas', 'CA': 'California', 'CO': 'Colorado', 'CT': 'Connecticut', 'DE': 'Delaware', 'DC': 'District of Columbia', 'FL': 'Florida', 'GA': 'Georgia', 'HI': 'Hawaii', 'ID': 'Idaho', 'IL': 'Illinois', 'IN': 'Indiana', 'IA': 'Iowa', 'KS': 'Kansas', 'KY': 'Kentucky', 'LA': 'Louisiana', 'ME': 'Maine', 'MT': 'Montana', 'NE': 'Nebraska', 'NV': 'Nevada', 'NH': 'New Hampshire', 'NJ': 'New Jersey', 'NM': 'New Mexico', 'NY': 'New York', 'NC': 'North Carolina', 'ND': 'North Dakota', 'OH': 'Ohio', 'OK': 'Oklahoma', 'OR': 'Oregon', 'MD': 'Maryland', 'MA': 'Massachusetts', 'MI': 'Michigan', 'MN': 'Minnesota', 'MS': 'Mississippi', 'MO': 'Missouri', 'PA': 'Pennsylvania', 'RI': 'Rhode Island', 'SC': 'South Carolina', 'SD': 'South Dakota', 'TN': 'Tennessee', 'TX': 'Texas', 'UT': 'Utah', 'VT': 'Vermont', 'VA': 'Virginia', 'WA': 'Washington', 'WV': 'West Virginia', 'WI': 'Wisconsin', 'WY': 'Wyoming'}

state_key = raw_input("enter a state abbreviation: ")
if states.has_key(state_key):
    print (states[state_key])
else:
    print (state_key + "was not found")
```

## Things you need to do

### Lab 10 – Morse Code reader

This lab will require more design and planning on your part. Requirements are provided rather than step-by-step instructions.

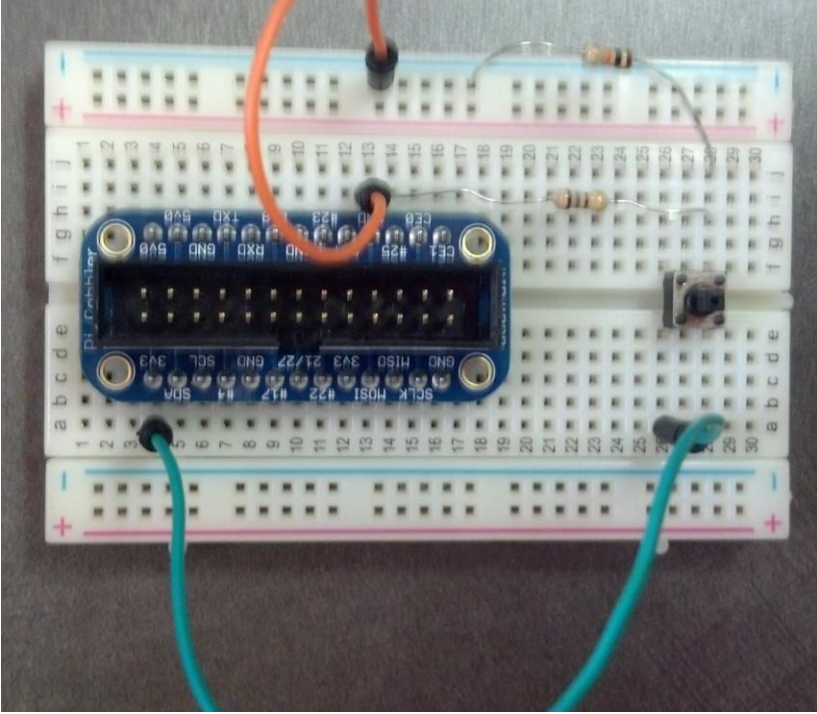
Goal: Interpret input from a GPIO input button on the Raspberry Pi as Morse Code characters. Display the characters on the console as they are recognized. The program should end when the user holds the button down for more than 3 seconds.

Use the International Morse Code listed at:

[http://en.wikipedia.org/wiki/File:International\\_Morse\\_Code.svg](http://en.wikipedia.org/wiki/File:International_Morse_Code.svg)

The exact time period for the dots and the dashes is up to you but they should follow the pattern listed in the International Morse Code standard.





## Final Project Proposal

Write a 1 page project proposal for the raspberry pi.

The proposal should include the following:

1. Project title
2. Team name
3. Team member names
4. Project description
5. A list of the components you would like to interface with the raspberry pi. Be as detailed as possible.

Upload your proposal as a pdf.

## Other Useful Links

[How to control a dc motor](#)

### **Spring 2015**

### **Fall 2014**

[Austin and Jeffrey – Guitar Hero](#)

[Que Wood & Andy Hayes - Maze](#)

[Jared Owens & Corey Shaffer - Sky Cam](#)

[Daryl and Shana - Piano](#)

[Alaina Bridges and Robert Welch - Camera](#)

[Zachary Downum and Kent Gunn - minesweeper](#)

[Lauren Johnson & Anna Crowder - Motion Detector](#)

[John Michael Eaton and Libby VanCuren - Laser Pointer](#)

[Heather Anderson & Hannah Cochran - Reaction Game](#)

[Seth Watkins & Caleb Vance - Stop Light](#)

[Garett Holmes & Evan Robberts - RC Car](#)

[Grant Buehrer & Caleb Russell - LED Game](#)