

1.

```
#!/usr/bin/python

import numpy
from matplotlib import pyplot as plt
from datetime import datetime

# Opens the data file with the formatted data, time, and wind speeds
try:
    f = open("/home/pi/ECE331/HW/HW04/data.txt",'r')
except:
    print "Couldn't open file"
    pass

# Loads the data from the text file then close file
data = numpy.loadtxt(f, dtype=int)

f.close()

# Creates datetimes and puts in array (year, month, day, hour, minute, second)
time = []
for i in range(len(data[:,1])):
    dates = datetime(data[i][0],data[i][1],data[i][2],data[i][3],data[i][4],data[i][5])
    time.append(dates)

# Plot the wind speeds
plt.plot(time,data[:,6],time,data[:,7])
plt.title("Top of Mount Katahdin Wind Speeds")
plt.xlabel("Dates")
plt.ylabel("Wind Speeds in mph")
plt.legend("Min Wind Speeds","Max Wind Speeds")

plt.show()
```

2.

```
***** main.c *****

#include "gpio.h"

#include <stdlib.h>
#include <stdio.h>
#include <stdint.h>
#include <unistd.h>

// Run tests on gpio.c functions
int main(int argc, char *argv[])
{
    // test case for gpio 4
    uint32_t num4 = 4;
    uint32_t iter;

    // turn on and off gpio output five times to blink LED
    GPIO_export(num4);
    GPIO_setdir(num4,"out");
    for (iter = 0; iter < 5; iter++) {
        GPIO_setval(num4,0);
        usleep(500000);
        GPIO_setval(num4,1);
        usleep(500000);
    }
    GPIO_unexport(num4);

    return 0;
}

***** gpio.h *****
```

```
#ifndef GPIO_H
#define GPIO_H

// includes to be used
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <string.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>

// gpio function prototypes
uint32_t GPIO_export(uint32_t gpio);
uint32_t GPIO_unexport(uint32_t gpio);
uint32_t GPIO_setval(uint32_t gpio, uint32_t val);
uint32_t GPIO_setdir(uint32_t gpio, char dir[]);

#endif

***** gpio.c *****

#include "gpio.h"

// Export gpio
uint32_t GPIO_export(uint32_t gpio)
{
    int32_t f; // file
    char string[10]; // write
    sprintf(string, "%u", gpio); // change gpio from int to string

    // open export file
    f = open("/sys/class/gpio/export", O_WRONLY);
    if (f < 0) {
        printf("Could not open export\n");
        return 1;
    }

    // export gpio
    if (write(f, string, strlen(string)) < 0) {
        printf("Could not export\n");
        return 2;
    }

    // close and wait
    close(f);
    usleep(50000);

    return 0;
}

// Unexport gpio
uint32_t GPIO_unexport(uint32_t gpio)
{
    int32_t f; // file
    char string[10]; // write
    sprintf(string, "%u", gpio); // change gpio from int to string

    // open unexport file
    f = open("/sys/class/gpio/unexport", O_WRONLY);
    if (f < 0) {
        printf("Could not open unexport\n");
        return 1;
    }

    // unexport gpio
```

```
    if(write(f,string,strlen(string)) < 0) {
        printf("Could not unexport\n");
        return 2;
    }

    // close and wait
    close(f);
    usleep(50000);

    return 0;
}

// Set value for gpio
uint32_t GPIO_setval(uint32_t gpio, uint32_t val)
{
    int32_t f; // file
    char string[50]; // write
    char string1[10]; // value
    sprintf(string,"/sys/class/gpio/gpio%u/value",gpio); // change gpio from int to string
    sprintf(string1,"%u",val); // change val from int to string

    // open value file
    f = open(string,O_WRONLY);
    if (f < 0) {
        printf("Could not open value\n");
        return 1;
    }

    // write value
    if(write(f,string1,strlen(string1)) < 0) {
        printf("Could not set value\n");
        return 2;
    }

    // close and wait
    close(f);
    usleep(500);

    return 0;
}

// Set direction for gpio
uint32_t GPIO_setdir(uint32_t gpio, char dir[])
{
    int32_t f; // file
    char string[50]; // write
    sprintf(string,"/sys/class/gpio/gpio%u/direction",gpio); // change gpio from int to string

    // open direction file
    f = open(string,O_WRONLY);
    if (f < 0) {
        printf("Could not open direction\n");
        return 1;
    }

    // write value
    if (write(f,dir,strlen(dir)) < 0) {
        printf("Could not set direction\n");
        return 2;
    }

    // close and wait
    close(f);
    usleep(500);

    return 0;
}
```

```
}
```

```
***** Makefile *****
```

```
# Makefile for HW04
# Creates main.o, gpio.o, and the executable gpio
```

```
TARGET=gpio
OBJS=main.o gpio.o
CFLAGS=-Wall -g -O2
```

```
all: $(TARGET)
$(TARGET): $(OBJS)
    $(CC) -o $(TARGET) $(OBJS)
```

```
clean:
    rm -f $(TARGET) $(OBJS) core*
```

```
*****
```

```
3. sudo ln -s /var/lib/systimer/logs/abc /usr/arm/opt/bin/foobar
```

```
4. egrep "^[0-9]+$" [0-9][0-9]
```

```
5. enscript -T 4 --header='$n %E %*|$%|Spencer Goulette' hw04.txt -o - | ps2pdf - output.pdf
```