Lab 3 – Dual shock 4 Remotes

LAB 2

SECTION C

James Mechikoff

SUBMISSION DATE: 9/18/2018

9/17/2018

Problem

The problem presented in this lab involved the dual shock 4 controllers and reading the data from them. It started off giving the lines of code for Cygwin that need to be ran to be ablmajor e to get data from them over the Bluetooth or USB connection. After explaining and having all the output statements be tested like the -b or -t. The next step was outputting the data recorded from time and gyroscopic data into a spreadsheet document to be translated into graphs. After this it required the acceleration output, -a, to be taken and plugged into a function for magnitude. The issue was the given magnitude function was needing to be finished. Then it had the time recorded converted to minutes, seconds, and milliseconds. After this, the button inputs were tested through the output -b. This had button outputs being recorded and combined into larger numbers based on multiple buttons being clicked.

Analysis

Most of the problem was simply translating given inputs and outputs through graphs or functions. It was straight forward and simple from that stand point. The first part was just learning how the dual shock runs, the second part was just outputs to spreadsheets to graphs. The third part was just as simple as input to output to magnitude function that was given and programmed in. The fourth part was as simple as taking inputs from buttons, to the output from the controller and taking that output and plugging it into a function defined to add controller inputs.

Design

The first part did not take any kind of code design past the .exe files being ran and watching the output. The second part didn't take any code yet again, it just required piping that was explained in an earlier lab. The third part was as simple as taking the given code, reading and understanding what it is doing and then editing it as desired. Changing output formats from %d to %lf if needed or changing the magnitude function to work instead of just repeating the output with out calculations. The last part was just writing another basic function based on the number on inputs being received.

Testing

The biggest problem with testing this week was getting the dual shock to cooperate. At first it was getting it to connect with Bluetooth, then it was getting it to read right, then it was finding out why the dual shocks would not work outside of the lab room. After all those issues, some other issues with doing the magnitude function and the time functions had to be tested. Remembering the material from the text about how to get remainders was the biggest hold up on time, but after remembering that things came a lot easier. Working with other classmates helped out a ton to see if things were being done right.

Comments

I loved this lab because it gave me a chance to see how a dual shock 4 controller works. I have used them and seen them a lot in my life so learning how they output to computers just made me so excited. I am glad that their outputs are so easy to work with in the program and I found it interesting to think about all the outputs and their applications. It explains why some of the games I played when I am younger were a bit frustrating with the Gyroscope.

Source Code

3.1

```
/*-----------------------------------------------------------------------------
--
-                                        SE 185 Lab 03
-              Developed for 185-Rursch by T.Tran and K.Wang
-      Name:        James Mechikoff
-      Section:     C
-      NetID:       726219551
-      Date:        9/11/2018
-----------------------------------------------------------------------------
*/

/*-----------------------------------------------------------------------------
--
-                                  Includes
-----------------------------------------------------------------------------
*/
#include <stdio.h>
#include <math.h>


/*-----------------------------------------------------------------------------
--
-                                  Defines
-----------------------------------------------------------------------------
*/
#define TRUE 1


/*-----------------------------------------------------------------------------
--
-                                  Prototypes
-----------------------------------------------------------------------------
*/
/* Put your function prototypes here */
double mag(double x, double y, double z);
int minutes(int min);
int seconds(int sec);
int millis(int ms);


/*-----------------------------------------------------------------------------
--
-                                          Implementation
-----------------------------------------------------------------------------
*/
int main(void) {
    /* DO NOT MODIFY THESE VARIABLE DECLARATIONS */
    int t;
    double  ax, ay, az;


    while (TRUE) {
        scanf("%d,%lf,%lf,%lf", &t, &ax, &ay, &az);
/* CODE SECTION 0 */

        printf("Echoing output: %8.3lf, %.4lf, %.4lf, %.4lf\n",
(double)t/1000, ax, ay, az);

/*    CODE SECTION 1 */
        printf("At %d ms, the acceleration's magnitude was: %lf\n", t,
mag(ax, ay, az));

/*    CODE SECTION 2 */

    printf("At %d minutes, %d seconds, and %d milliseconds it was: %lf\n",
        minutes(t), seconds(t), millis(t), mag(ax,ay,az));


    }

  return 0;
}

/* Put your functions here */
double mag(double x, double y, double z) {
    //Step 8, uncomment and modify the next line
    return sqrt(pow(x,2) + pow(y,2) + pow(z,2));
}

int minutes(int min) {

 return (min/1000/60);

}

int seconds(int sec) {
      if (sec/1000 > 60){

            return ((sec/1000) % 60);

      }else {

            return (sec/1000);

      }
}

int millis(int ms) {

      return (ms % 1000);

}
```

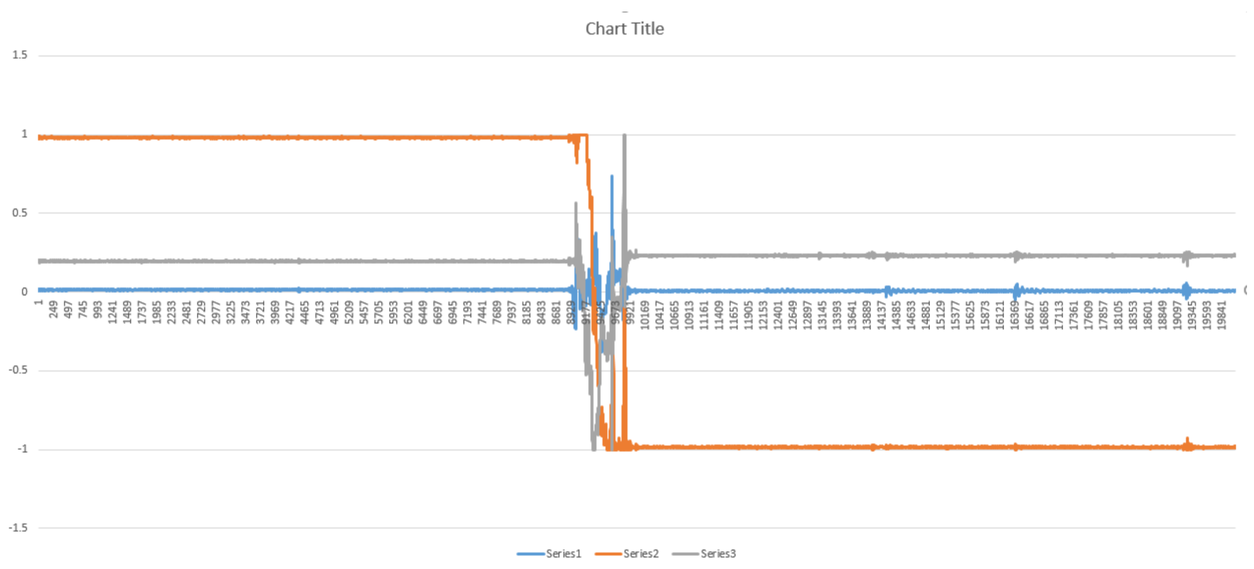
3.2

```
/*-----------------------------------------------------------------------------
--
-                                       SE 185 Lab 03
-              Developed for 185-Rursch by T.Tran and K.Wang
-     Name:       James Mechikoff
-     Section:    C
-     NetID:            726219551
-     Date:       9/17/2018
-----------------------------------------------------------------------------
*/

/*---------------------------------------------------------------------------
--
-                                 Includes
-----------------------------------------------------------------------------
*/
#include <stdio.h>
#include <math.h>


/*---------------------------------------------------------------------------
--
-                                 Defines
-----------------------------------------------------------------------------
*/
#define TRUE 1


/*---------------------------------------------------------------------------
--
-                                 Prototypes
-----------------------------------------------------------------------------
*/
int numButtons(int x, int y, int z, int a) {

    int numPressed;

    numPressed = x + y + z + a;

    return numPressed;

}


/*---------------------------------------------------------------------------
--
-                                           Implementation
-----------------------------------------------------------------------------
*/
int main(void) {

      int btri, bo, bx, bsq;

    while (TRUE) {

            scanf("%d,%d,%d,%d", &btri, &bo, &bx, &bsq);
        printf("Number of buttons pressed: %d\n", numButtons(btri, bo, bx,
bsq));
        fflush(stdout);

    }
```

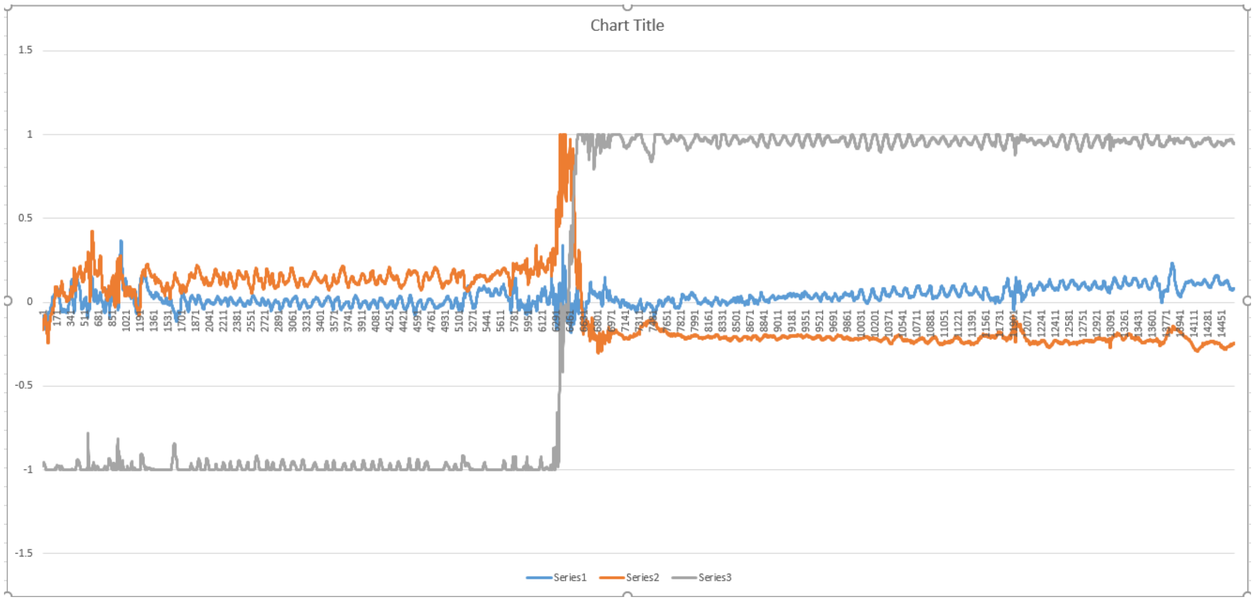
```
  return 0;
}

/* Put your functions here */
```

Screen Shots

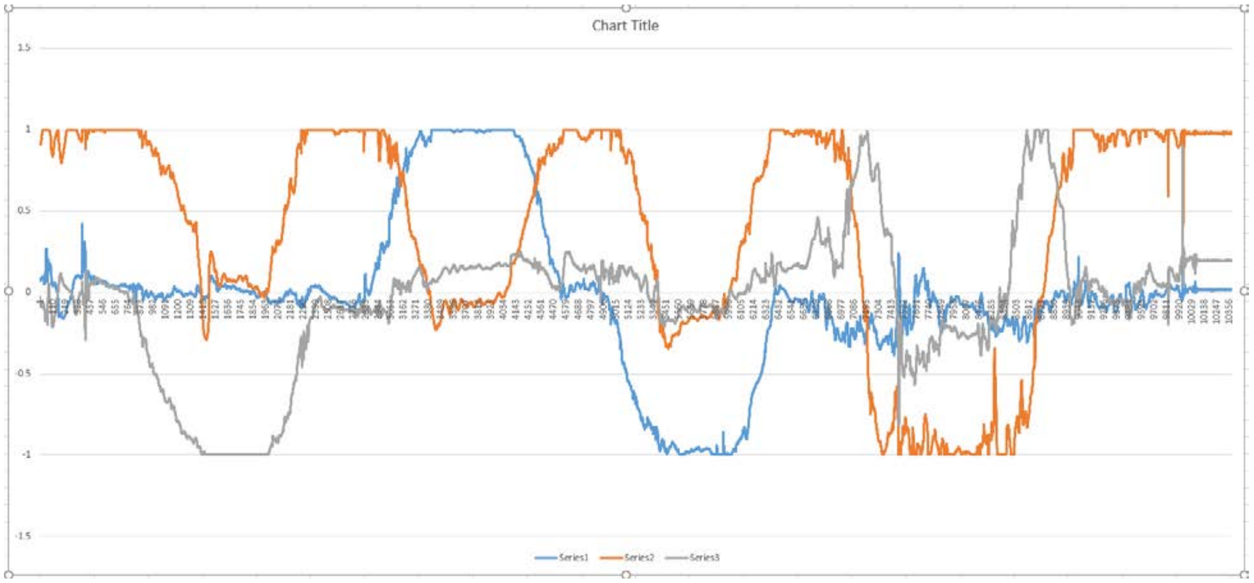
Flat Graph Screenshot



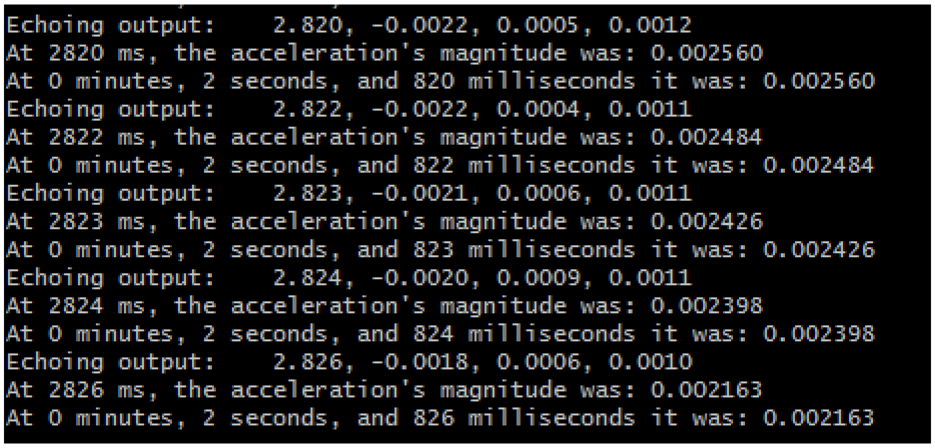
Front Graph.



Custom Graph.

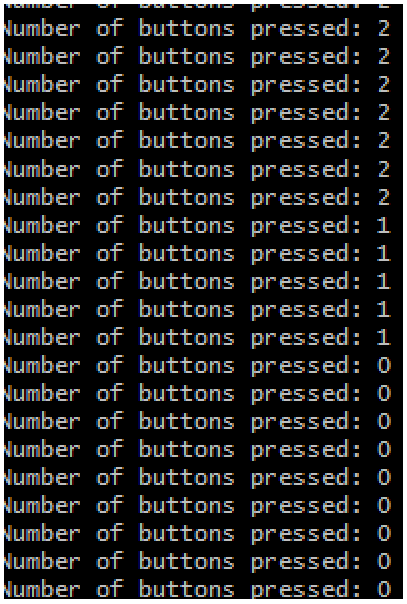


Magnitude

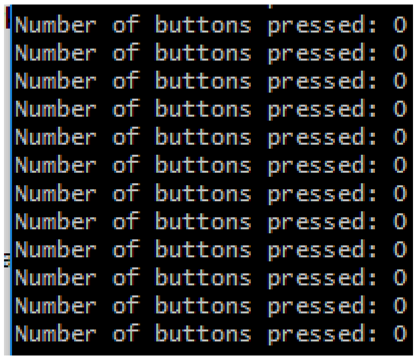


```
Echoing output:    2.820, -0.0022, 0.0005, 0.0012
At 2820 ms, the acceleration's magnitude was: 0.002560
At 0 minutes, 2 seconds, and 820 milliseconds it was: 0.002560
Echoing output:    2.822, -0.0022, 0.0004, 0.0011
At 2822 ms, the acceleration's magnitude was: 0.002484
At 0 minutes, 2 seconds, and 822 milliseconds it was: 0.002484
Echoing output:    2.823, -0.0021, 0.0006, 0.0011
At 2823 ms, the acceleration's magnitude was: 0.002426
At 0 minutes, 2 seconds, and 823 milliseconds it was: 0.002426
Echoing output:    2.824, -0.0020, 0.0009, 0.0011
At 2824 ms, the acceleration's magnitude was: 0.002398
At 0 minutes, 2 seconds, and 824 milliseconds it was: 0.002398
Echoing output:    2.826, -0.0018, 0.0006, 0.0010
At 2826 ms, the acceleration's magnitude was: 0.002163
At 0 minutes, 2 seconds, and 826 milliseconds it was: 0.002163
```

Button Presses



```
Number of buttons pressed: 2
Number of buttons pressed: 2
Number of buttons pressed: 2
Number of buttons pressed: 2
Number of buttons pressed: 2
Number of buttons pressed: 2
Number of buttons pressed: 2
Number of buttons pressed: 1
Number of buttons pressed: 1
Number of buttons pressed: 1
Number of buttons pressed: 1
Number of buttons pressed: 1
Number of buttons pressed: 0
Number of buttons pressed: 0
Number of buttons pressed: 0
Number of buttons pressed: 0
Number of buttons pressed: 0
Number of buttons pressed: 0
Number of buttons pressed: 0
Number of buttons pressed: 0
Number of buttons pressed: 0
```



```
Number of buttons pressed: 0
Number of buttons pressed: 0
Number of buttons pressed: 0
Number of buttons pressed: 0
Number of buttons pressed: 0
Number of buttons pressed: 0
Number of buttons pressed: 0
Number of buttons pressed: 0
Number of buttons pressed: 0
Number of buttons pressed: 0
Number of buttons pressed: 0
Number of buttons pressed: 0
```