Lab 2

Section B

Submitted By:

Kent Mark

Submission Date:

2/7/18

**Problem:**

The purpose of this lab was for us to learn how to: collect gyroscopic/input data from the Sony Dual Shock 4 controller, learn how to write and call functions, and learn how to use a “*while loop*”.

**Analysis:**

The initial lab problem required that a certain command be entered directly into the Cygwin compiler so that access could be gained into the time/gyroscopic output of the Sony Dual Shock 4 controller. We were then asked to place the controller in varying orientations and directions, gather output data for a total of 20 seconds, and store said data into an Excel.csv file where this information would be represented through a scatter-plot graph.

The next lab problem required that same command, “*./ds4rd.exe -d 054c:05c4 -D DS4\_BT –b*”, be entered into the compiler, but in this case instead of simply observing the output we would alter it through the source code that was provided. The alterations included: converting milliseconds to seconds, limiting the character space for seconds to 8 characters and rounding seconds to 3 decimal places, altering the acceleration values to display a number rounded to 4 decimal points within an 8 character area, determining the magnitude of acceleration, and writing separate functions that would allow a certain printf statement to operate.

Finally, the last lab problem required that a program be written that would output the number of buttons being pressed on the controller at a given moment.

**Design:**

For *lab* *problem* *1* I simply ran the code that was provided, collected the resulting data in an excel spreadsheet, rounded the numbers to about 1 decimal place for neatness, highlighted about 20 cells of information and entered them into a three-line scatter plot graph.

For *lab* *problem* *2* I first converted milliseconds to seconds by dividing variable “t” by 1000 and setting both the character limit and the decimal limit with “*8.3lf*”. I then set the character limit and the decimal precision for the acceleration values with “*7.4lf”*. Next the magnitude function was corrected through the use of the “*pow*”, the variable “*acclMag*” was then placed in a “*sqrt*” function. Lastly, I defined three separate functions to determine minutes, seconds, and milliseconds. For minutes I divided my value for min by 6000, for seconds I took the mod of the value for seconds and 6000 and divided this answer by 1000, finally for milliseconds I took the mod of milliseconds and 60000, and then took the mod of the resulting answer and 1000.

For *lab problem 3* I defined a variable for each of the buttons and determined my output by adding together whichever buttons are pressed.

**Testing:**

I tested “*lab problem 2*” by allowing the program to run for a certain amount of time and checking to see if I at least got some sort of output for each of the functions, and I tested “*lab* *problem* *3*” by holding down various buttons and running the code.

**Comments:**

Thanks to this assignment I now have a somewhat clearer understanding on defining functions and I also had fun tinkering with the DS4 controller

Source Code #1

**SE 185 Lab 02**

**- Developed for 185-Rursch by T.Tran and K.Wang**

**- Adapted to cpre 185 - swamy ponpandi**

**- Name: Kent Mark**

**- Section: B**

**- NetID: komark@istate.edu**

**- Date: 1/31/18**

**-----------------------------------------------------------------------------\*/**

**/\*-----------------------------------------------------------------------------**

**- 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 mil);

/\*-----------------------------------------------------------------------------

- 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, %7.4lf, %7.4lf, %7.4lf\n", t/1000.0, ax, ay, az);

/\* CODE SECTION 1 \*/

printf("At %d s, 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) {

double acclMag;

acclMag = pow(x,2)+pow(y,2)+pow(z,2);

//Step 8, uncomment and modify the next line

return sqrt(acclMag);

}

int minutes (int min){

int minute;

minute = min/6000;

return minute;

}

int seconds (int sec){

int seconds;

seconds = (sec%6000)/1000;

return seconds;

}

int millis (int mil){

int millisecond;

mil = (mil%60000)%1000;

return millisecond;

}

Source Code #2

SE 185 Lab 02

- Developed for 185-Rursch by T.Tran and K.Wang

- Adapted to cpre 185 - swamy ponpandi

- Name: Kent Mark

- Section: B

- NetID: Komark@iastate.edu

- Date: 1-31-18

-----------------------------------------------------------------------------\*/

/\*-----------------------------------------------------------------------------

- Includes

-----------------------------------------------------------------------------\*/

#include <stdio.h>

#include <math.h>

/\*-----------------------------------------------------------------------------

- Defines

-----------------------------------------------------------------------------\*/

#define TRUE 1

/\*-----------------------------------------------------------------------------

- Prototypes

-----------------------------------------------------------------------------\*/

/\*-----------------------------------------------------------------------------

- Implementation

-----------------------------------------------------------------------------\*/

int main(void) {

int triangle, o, x, square;

while (TRUE) {

scanf("%d,%d,%d,%d", &triangle, &o, &x, &square);

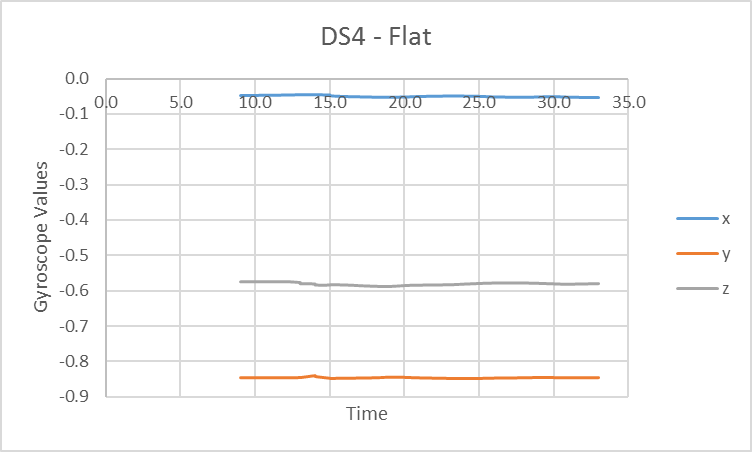
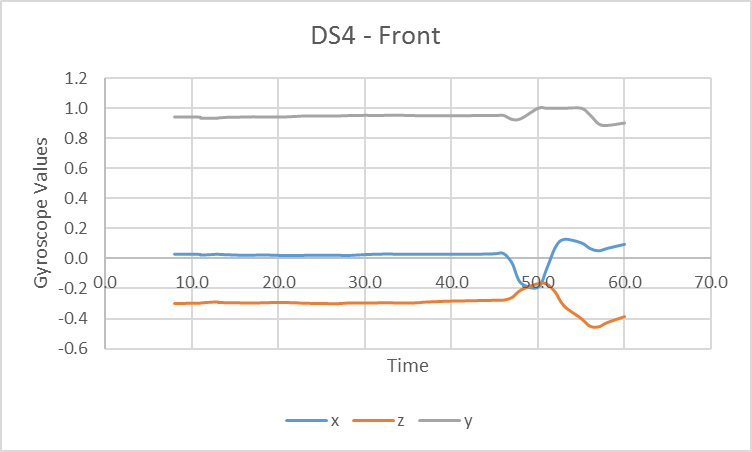
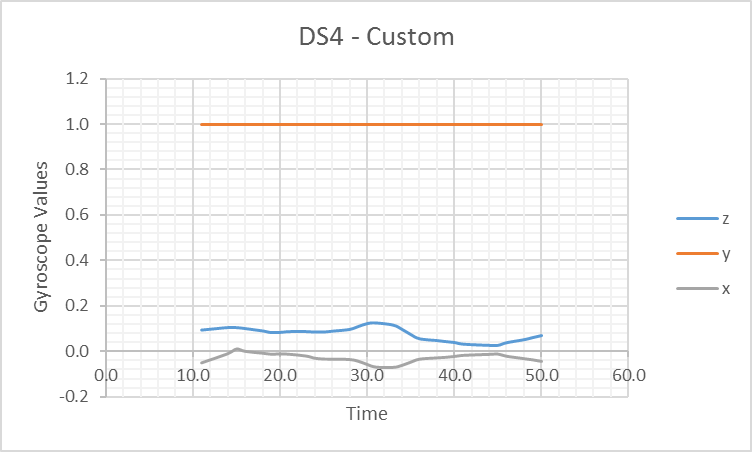
printf("%d", square+x+o+triangle);

fflush(stdout);

}

return 0;

}



*Capture 3*

*Capture 1*

*Capture 2*