Introduction to the DS4 and Functions

LAB 3

SECTION 3

Jack Morrison

SUBMISSION DATE:

02/17/2023

Problem

The first part of the lab has us running the provided ds4rd.exe executable to collect some data from the dual shock controller. The executable has some flags that it takes to perform different tasks depending on what flags are set. For instance the flag “-t” gives time in milliseconds, and “-a” gives acceleration data. You also have to set whether or not you are using the controller in Bluetooth mode or USB mode.

The next problem has us write some functions namely, magnitude, minutes, seconds, and milliseconds. The purpose of magnitude is to calculate the magnitude of a vector which is described by the following formula $\sqrt{\left(x^2+y^2+z^2\right)}$. Then minutes, seconds, and milliseconds are meant to calculate the number of minutes, then remaining seconds, then remaining milliseconds from a given input in milliseconds.

Problem three has us using the buttons on the controller. It asks us to write a function that returns the number of buttons being pressed at any give moment.

Analysis

For problem 1 there isn’t much to analyze you pretty much just follow instructions on how to use the command. I however would like to mention that the use of an external executable writing input to your c code via the stdin via pipes is a little odd. Why not provide a libds4rd.lib file to compile our C code with, and provide an interface for acquiring data from the controller within C code.

For problem 2 magnitude is a fairly straight forward problem, it only requires a few math concepts which can be almost directly translated into C code. However, the minutes, seconds, and milliseconds functions require a little more thought. You have to design each function to take the full milliseconds input and only return what is left after the larger time blocks, so the design requires extra care to be taken when designing it.

For problem 3 the function to count the number of buttons pressed is fairly straightforward. Each button is either 0 or 1, 1 if pressed so just add up each individual buttons value and voila.

Design

No design for problem 1 just data gathering into files.

For problem 2 magnitude uses the math.h header file and the sqrt, and pow functions to calculate the correct magnitude of a vector. For minutes the calculation is easy its just the milliseconds converted into minutes dropping any fractional leftovers. The seconds function is a little harder you convert the milliseconds into seconds, and then subtract the amount of time calculated in minutes converted back into seconds. You then follow the same process for milliseconds by subtracting the seconds and minutes converted back into milliseconds.

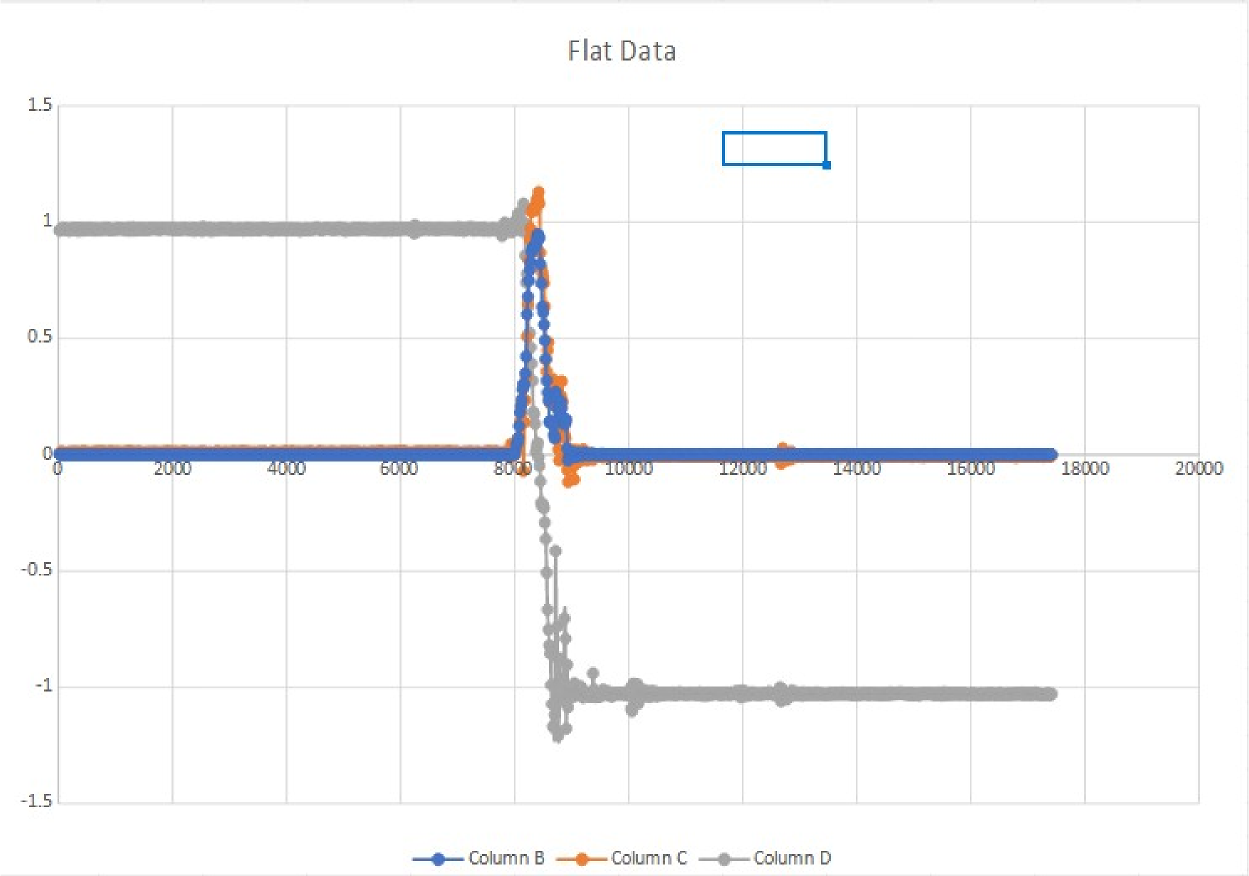
For problem 3 the design is as simple as stated take 4 individual button inputs and sum them all together to get then number of buttons pushed.

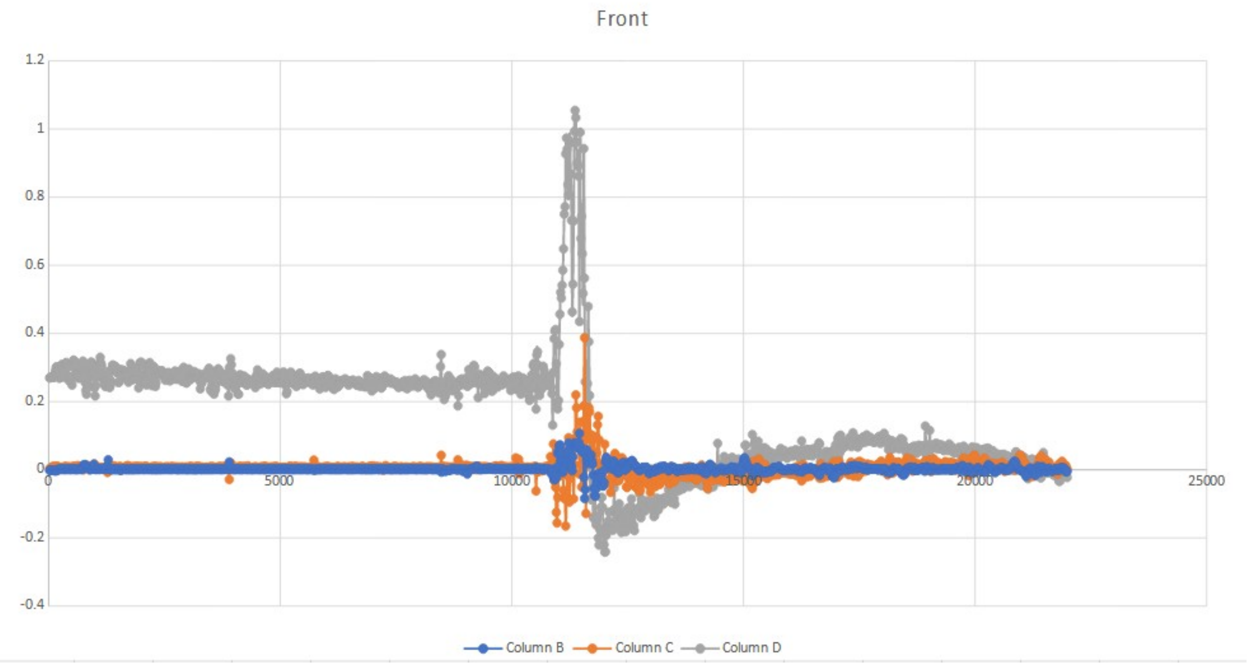
Testing

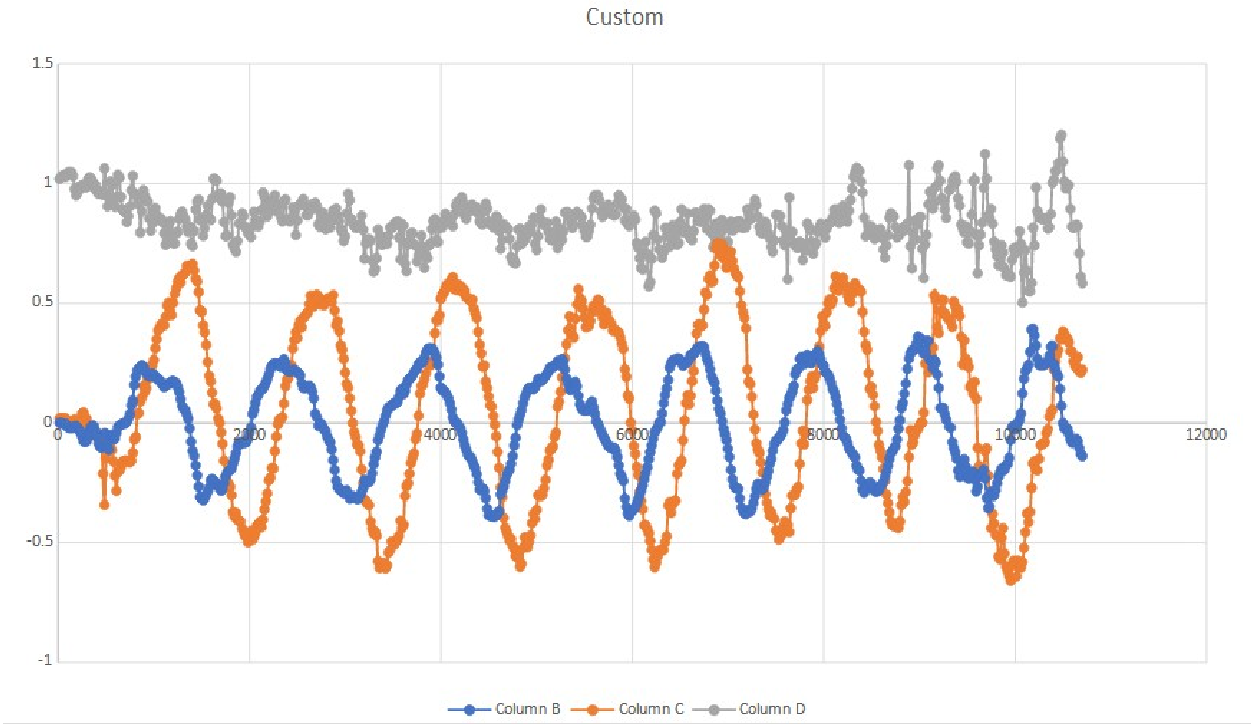
To test problem 2 I started with magnitude, and input some known values and compared them to an online calculator to make sure the result was correct. For minutes, seconds, and milliseconds I ran the ds4rd.exe and piped it to the input and watched to make sure the behavior was correct. And I found a couple of times where I had incorrect behavior. The biggest one was when I first wrote it I forgot to subtract minutes from the milliseconds so after a minute passed the milliseconds value had an extra 60,000 in front of it.

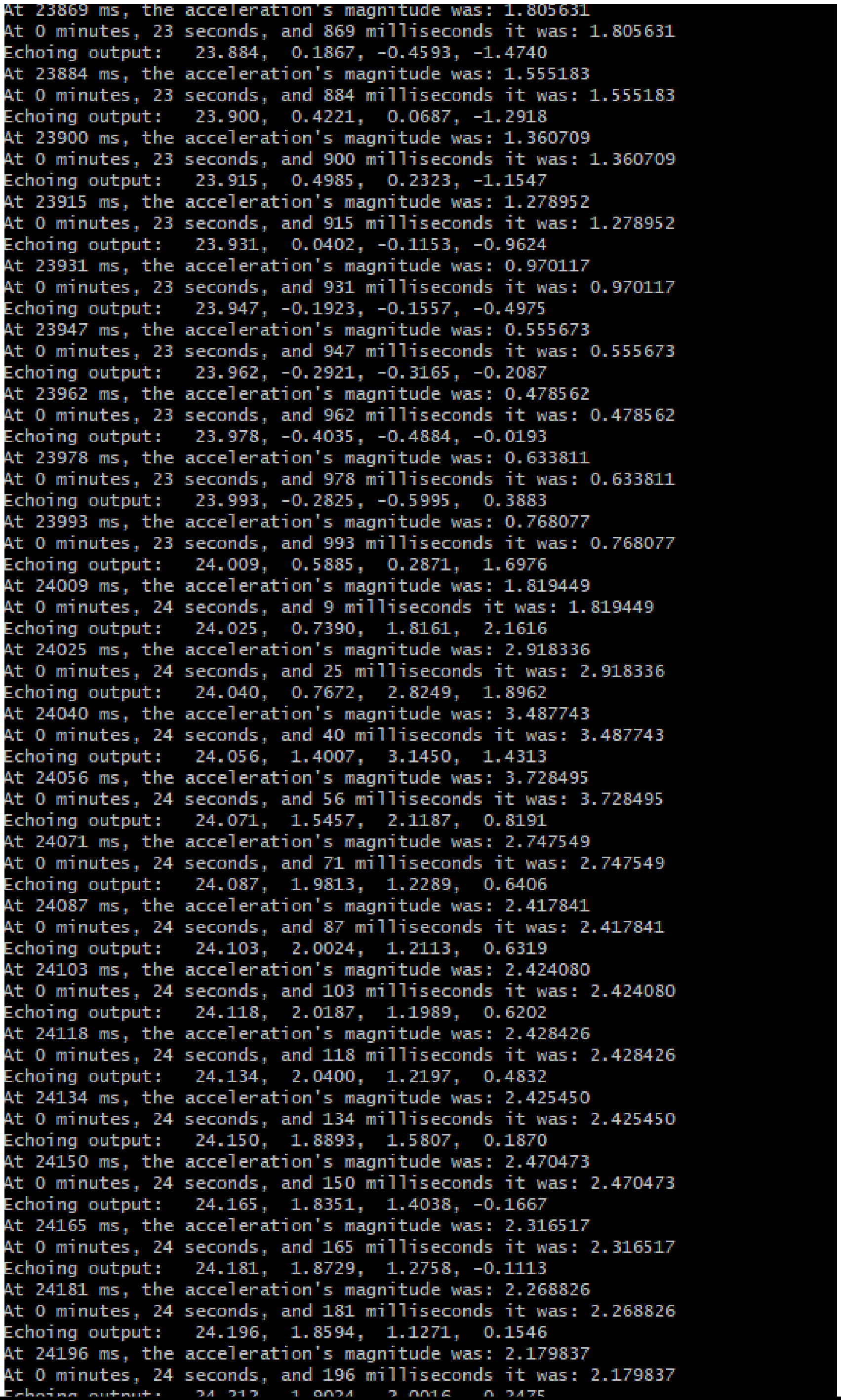
To test problem 3 I just ran ds4rd.exe and made sure the number of buttons I pressed matched the number of buttons that the program displayed.

Screen Shots

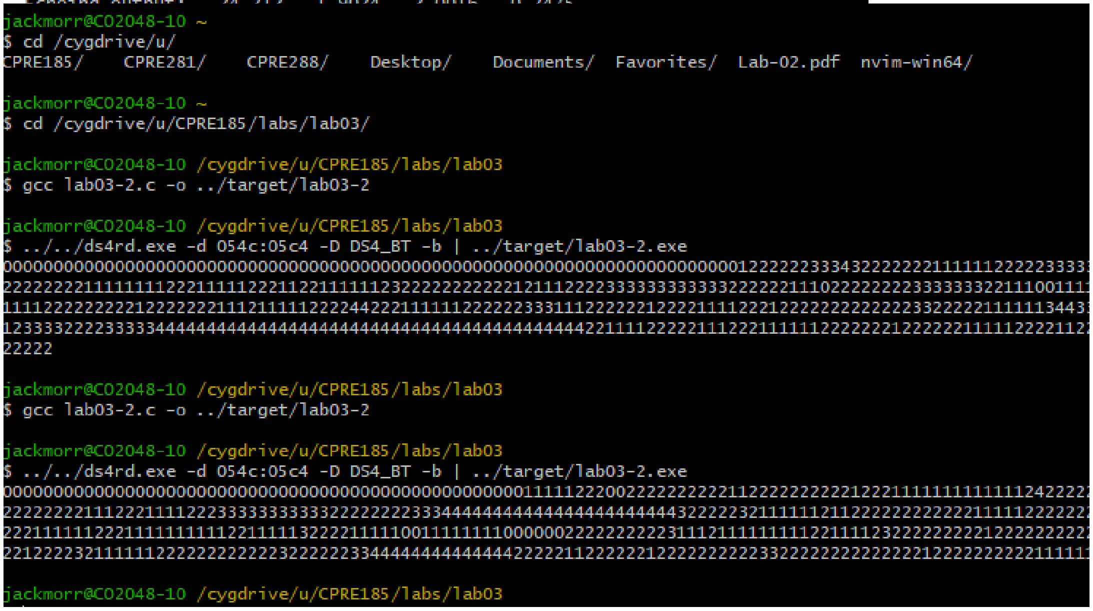



Front
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0
5000
10000
15000
20000
25000
Column B
Column C
Column D


Custom
1.5
1
0.5
0
-0.5
-1
0
2000
4000
6000
8000
10000
12000
Column B
Column C
Column D



```
At 23869 ms, the acceleration's magnitude was: 1.805631
At 0 minutes, 23 seconds, and 869 milliseconds it was: 1.805631
Echoing output:   23.884,  0.1867, -0.4593, -1.4740
At 23884 ms, the acceleration's magnitude was: 1.555183
At 0 minutes, 23 seconds, and 884 milliseconds it was: 1.555183
Echoing output:   23.900,  0.4221,  0.0687, -1.2918
At 23900 ms, the acceleration's magnitude was: 1.360709
At 0 minutes, 23 seconds, and 900 milliseconds it was: 1.360709
Echoing output:   23.915,  0.4985,  0.2323, -1.1547
At 23915 ms, the acceleration's magnitude was: 1.278952
At 0 minutes, 23 seconds, and 915 milliseconds it was: 1.278952
Echoing output:   23.931,  0.0402, -0.1153, -0.9624
At 23931 ms, the acceleration's magnitude was: 0.970117
At 0 minutes, 23 seconds, and 931 milliseconds it was: 0.970117
Echoing output:   23.947, -0.1923, -0.1557, -0.4975
At 23947 ms, the acceleration's magnitude was: 0.555673
At 0 minutes, 23 seconds, and 947 milliseconds it was: 0.555673
Echoing output:   23.962, -0.2921, -0.3165, -0.2087
At 23962 ms, the acceleration's magnitude was: 0.478562
At 0 minutes, 23 seconds, and 962 milliseconds it was: 0.478562
Echoing output:   23.978, -0.4035, -0.4884, -0.0193
At 23978 ms, the acceleration's magnitude was: 0.633811
At 0 minutes, 23 seconds, and 978 milliseconds it was: 0.633811
Echoing output:   23.993, -0.2825, -0.5995,  0.3883
At 23993 ms, the acceleration's magnitude was: 0.768077
At 0 minutes, 23 seconds, and 993 milliseconds it was: 0.768077
Echoing output:   24.009,  0.5885,  0.2871,  1.6976
At 24009 ms, the acceleration's magnitude was: 1.819449
At 0 minutes, 24 seconds, and 9 milliseconds it was: 1.819449
Echoing output:   24.025,  0.7390,  1.8161,  2.1616
At 24025 ms, the acceleration's magnitude was: 2.918336
At 0 minutes, 24 seconds, and 25 milliseconds it was: 2.918336
Echoing output:   24.040,  0.7672,  2.8249,  1.8962
At 24040 ms, the acceleration's magnitude was: 3.487743
At 0 minutes, 24 seconds, and 40 milliseconds it was: 3.487743
Echoing output:   24.056,  1.4007,  3.1450,  1.4313
At 24056 ms, the acceleration's magnitude was: 3.728495
At 0 minutes, 24 seconds, and 56 milliseconds it was: 3.728495
Echoing output:   24.071,  1.5457,  2.1187,  0.8191
At 24071 ms, the acceleration's magnitude was: 2.747549
At 0 minutes, 24 seconds, and 71 milliseconds it was: 2.747549
Echoing output:   24.087,  1.9813,  1.2289,  0.6406
At 24087 ms, the acceleration's magnitude was: 2.417841
At 0 minutes, 24 seconds, and 87 milliseconds it was: 2.417841
Echoing output:   24.103,  2.0024,  1.2113,  0.6319
At 24103 ms, the acceleration's magnitude was: 2.424080
At 0 minutes, 24 seconds, and 103 milliseconds it was: 2.424080
Echoing output:   24.118,  2.0187,  1.1989,  0.6202
At 24118 ms, the acceleration's magnitude was: 2.428426
At 0 minutes, 24 seconds, and 118 milliseconds it was: 2.428426
Echoing output:   24.134,  2.0400,  1.2197,  0.4832
At 24134 ms, the acceleration's magnitude was: 2.425450
At 0 minutes, 24 seconds, and 134 milliseconds it was: 2.425450
Echoing output:   24.150,  1.8893,  1.5807,  0.1870
At 24150 ms, the acceleration's magnitude was: 2.470473
At 0 minutes, 24 seconds, and 150 milliseconds it was: 2.470473
Echoing output:   24.165,  1.8351,  1.4038, -0.1667
At 24165 ms, the acceleration's magnitude was: 2.316517
At 0 minutes, 24 seconds, and 165 milliseconds it was: 2.316517
Echoing output:   24.181,  1.8729,  1.2758, -0.1113
At 24181 ms, the acceleration's magnitude was: 2.268826
At 0 minutes, 24 seconds, and 181 milliseconds it was: 2.268826
Echoing output:   24.196,  1.8594,  1.1271,  0.1546
At 24196 ms, the acceleration's magnitude was: 2.179837
At 0 minutes, 24 seconds, and 196 milliseconds it was: 2.179837
Echoing output:   24.212,  1.9024,  2.0016,  0.2475
```



```
jackmorr@C02048-10 ~
$ cd /cygdrive/u/
CPRE185/    CPRE281/    CPRE288/    Desktop/    Documents/  Favorites/  Lab-02.pdf  nvim-win64/

jackmorr@C02048-10 ~
$ cd /cygdrive/u/CPRE185/labs/lab03/

jackmorr@C02048-10 /cygdrive/u/CPRE185/labs/lab03
$ gcc lab03-2.c -o ../target/lab03-2

jackmorr@C02048-10 /cygdrive/u/CPRE185/labs/lab03
$ ../../ds4rd.exe -d 054c:05c4 -D DS4_BT -b | ../target/lab03-2.exe
0000000000000000000000000000000000000000000000000000000000000000000000001222222333432222222111111222223333
2222222211111111222111112221122111111232222222222222121112222333333333333322222211102222222223333333322111001111
1111222222222212222222111211111222244222111111222222333111222222122221111222122222222222222233222221111113443
12333322223333344444444444444444444444444444444444444444221111222221112221111112222222122222211111222211
22222

jackmorr@C02048-10 /cygdrive/u/CPRE185/labs/lab03
$ gcc lab03-2.c -o ../target/lab03-2

jackmorr@C02048-10 /cygdrive/u/CPRE185/labs/lab03
$ ../../ds4rd.exe -d 054c:05c4 -D DS4_BT -b | ../target/lab03-2.exe
000000000000000000000000000000000000000000000000000111112220022222222222112222222222212221111111111111242222
222222221112221111222333333333333222222222333444444444444444444444443222223211111121122222222222221111122222
2221111112221111111111221111132222111110011111111000000222222222231112111111111122111123222222222221222222222
22122223211111112222222222222322222223344444444444444222221122222212222222222233222222222222221222222222211111

jackmorr@C02048-10 /cygdrive/u/CPRE185/labs/lab03
```