INTRODUCTION TO THE DS4 AND FUNCTIONS

LAB 03

SECTION 5

ARYAN RAO

DATE-09/14/2021

SUBMISSION DATE-09/21/2021

Problem 1

Use the DualShock 4 controller and collect data as we move the controller around.

Analysis

The ds4rd.exe file is already included. We need to run that and give variations to the movement of the controller so that we get different values.

Design

- Run the command `/ds4rd.exe -d 054c:05c4 -D DS4_BT -t -g`
- Place the DS4 flat and the rotate it each for 10 seconds.
- Hold the DS4 pointing upward and then turn it around.
- Choose a simple movement for the DS4(Sideways).

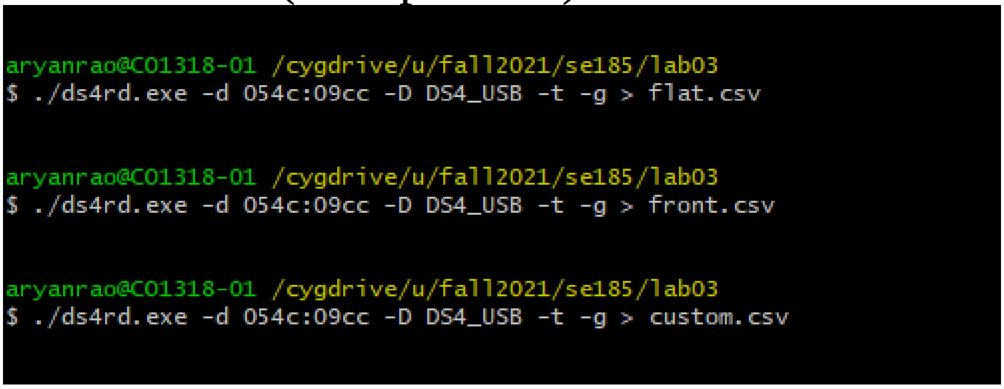
Testing

The first graph should have a bump in the middle and the second graph should be pointing downwards. Compare them with the results.

Comments

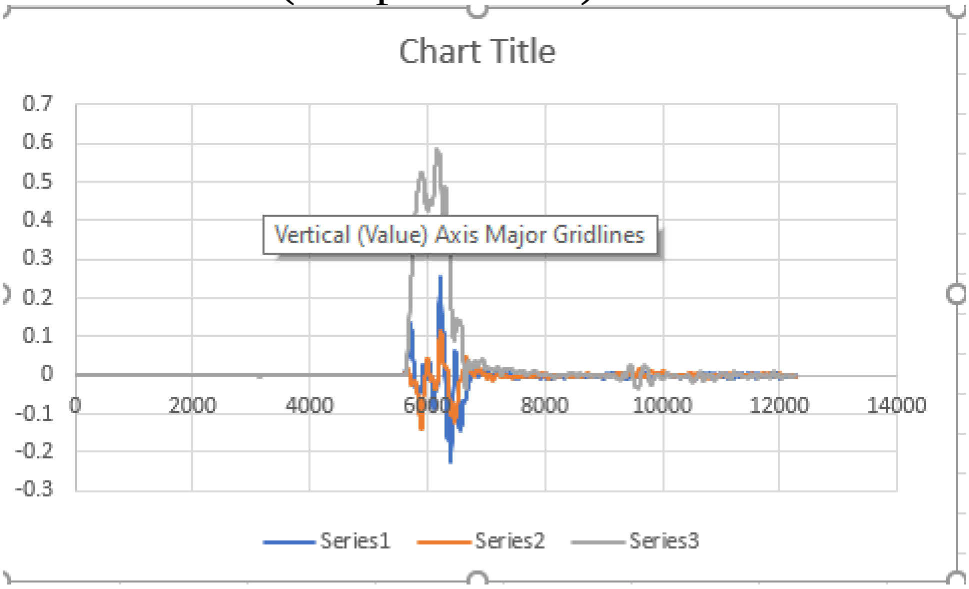
Make sure to use DS4_BT if using Bluetooth and DS4_USB if the controller is connected to the cable.

Screenshot 1(Compilation)

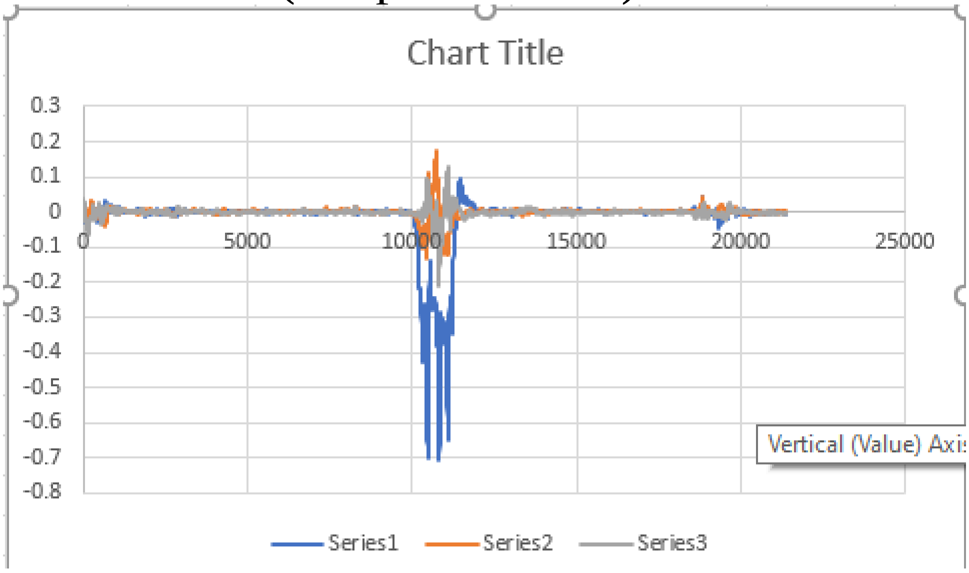


```
aryanrao@C01318-01 /cygdrive/u/fall2021/se185/lab03
$ ./ds4rd.exe -d 054c:09cc -D DS4_USB -t -g > flat.csv

aryanrao@C01318-01 /cygdrive/u/fall2021/se185/lab03
$ ./ds4rd.exe -d 054c:09cc -D DS4_USB -t -g > front.csv

aryanrao@C01318-01 /cygdrive/u/fall2021/se185/lab03
$ ./ds4rd.exe -d 054c:09cc -D DS4_USB -t -g > custom.csv
```

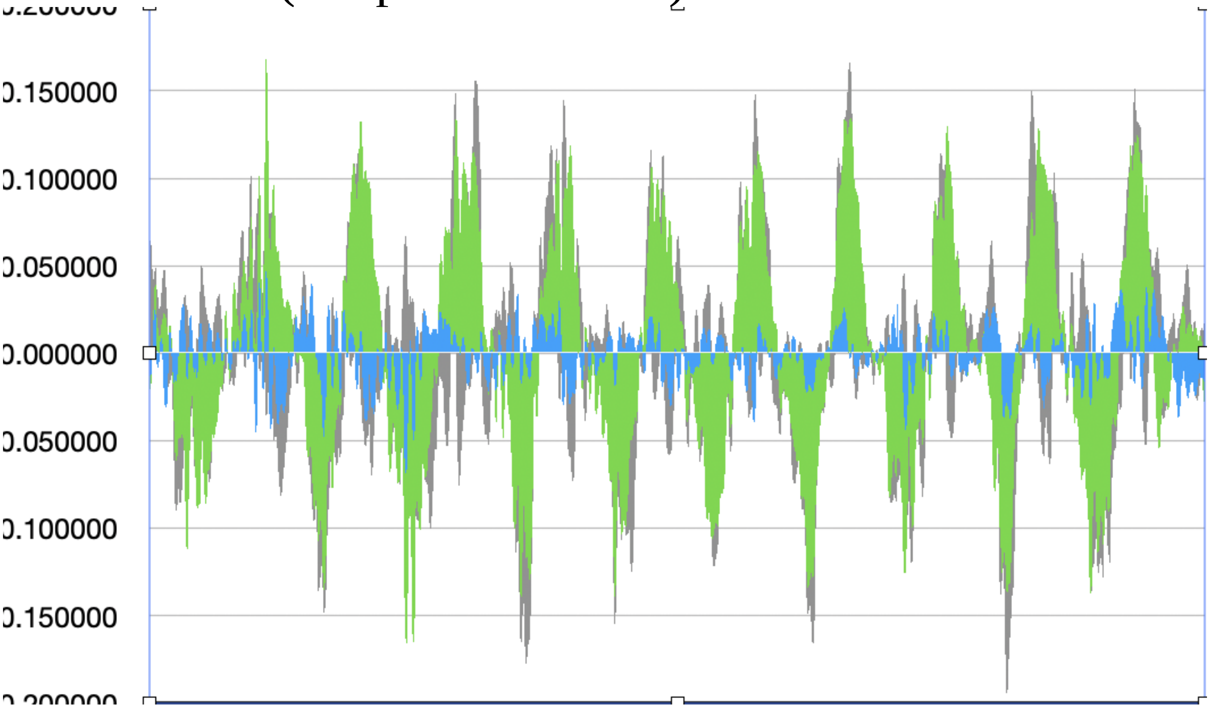
Screenshot 2(Graph-FLAT)



Screenshot 3(Graph-FRONT)



Screenshot 4(Graph-CUSTOM)



Problem 2

Run the ds4.exe again and convert milliseconds to seconds, acceleration and get the magnitude.

Analysis

The ds4rd.exe file is already included. We need to write function magnitude to convert milliseconds to second, modify acceleration to get values up to 4 digits of precession and make functions minutes, seconds, and milliseconds.

Design

- Run the command `./ds4rd.exe -d 054c:05c4 -D DS4_BT -t -a | ./lab03-1`
- Place the DS4 flat and do some movement to get values.
- Write function magnitude to convert milliseconds to seconds.
- Write functions minutes, seconds, and milliseconds to get subsequent values.

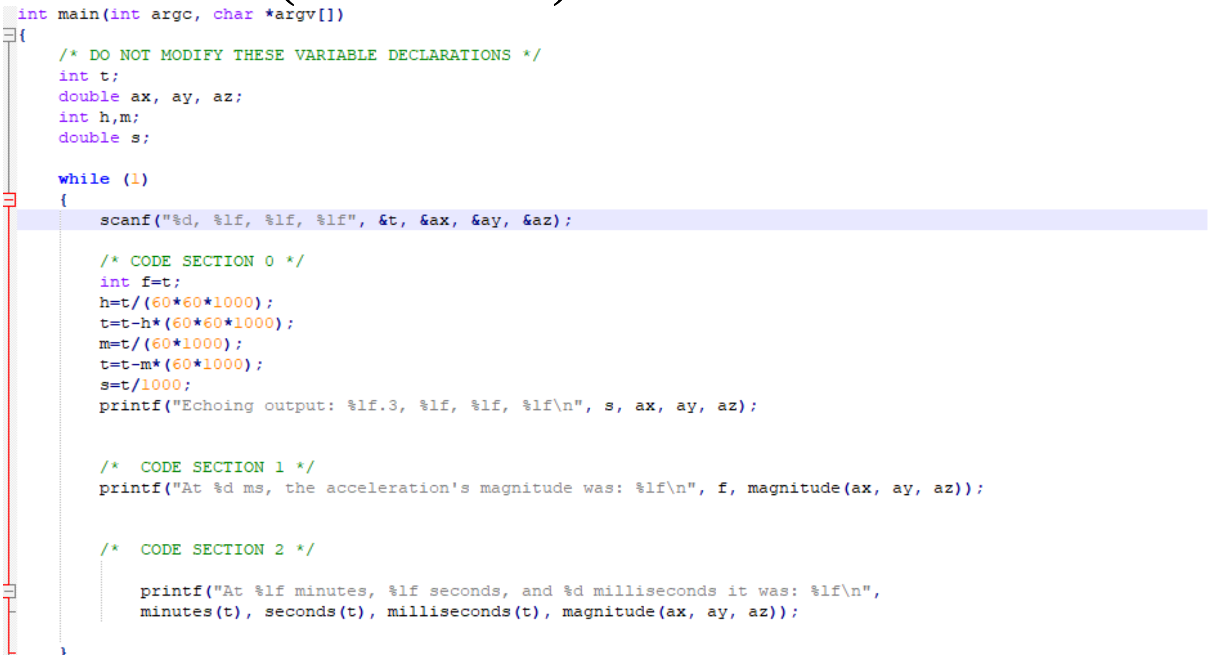
Testing

Calculate the magnitude manually and compare it with the answer you get. Check if function minutes, seconds, and milliseconds are working correctly by comparing them with the real values.

Comments

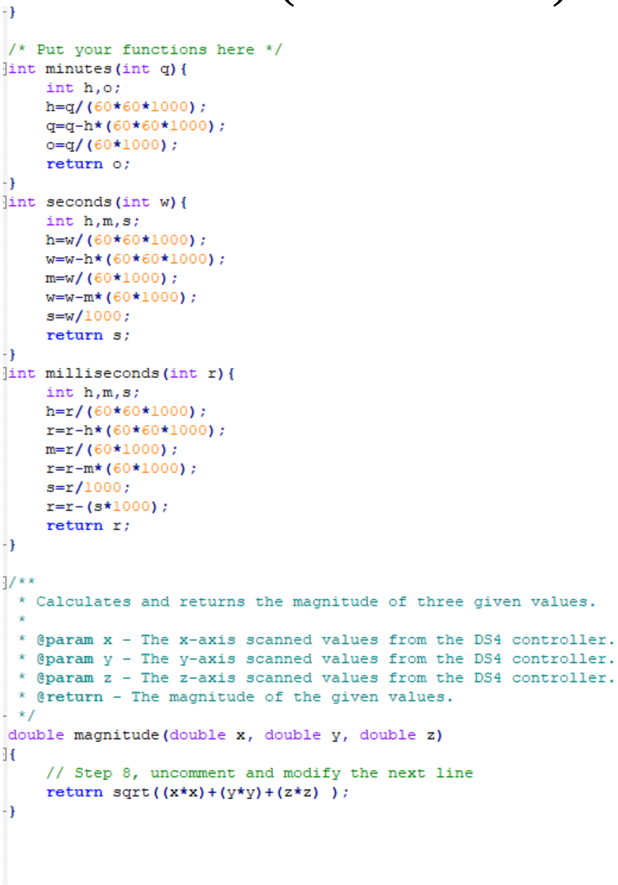
Make sure to prototype the functions beforehand and write their definition afterwards. Make sure to use DS4_BT if using Bluetooth and DS4_USB if the controller is connected to the cable.

Screenshot 1(Code-Part 1)



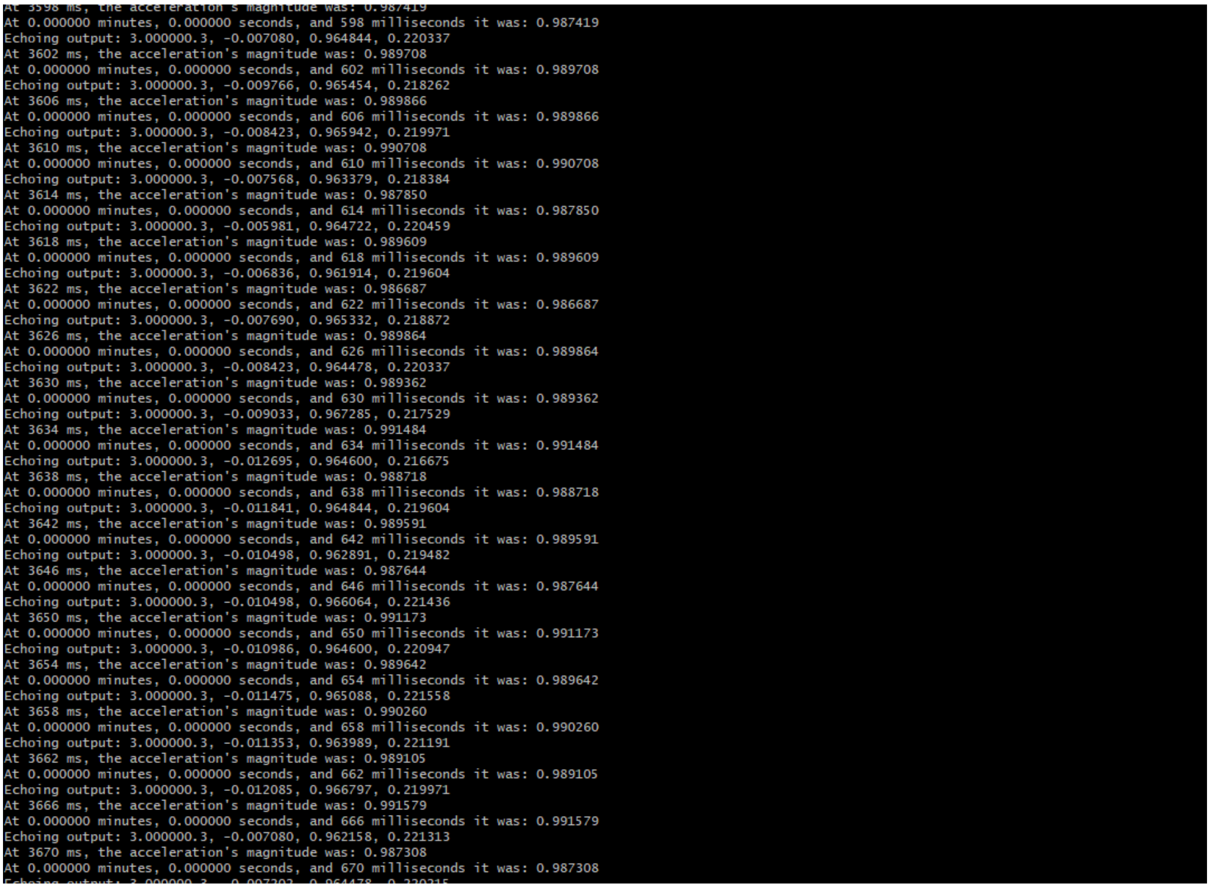
```
int main(int argc, char *argv[])
{
    /* DO NOT MODIFY THESE VARIABLE DECLARATIONS */
    int t;
    double ax, ay, az;
    int h,m;
    double s;

    while (1)
    {
        scanf("%d, %lf, %lf, %lf", &t, &ax, &ay, &az);

        /* CODE SECTION 0 */
        int f=t;
        h=t/(60*60*1000);
        t=t-h*(60*60*1000);
        m=t/(60*1000);
        t=t-m*(60*1000);
        s=t/1000;
        printf("Echoing output: %lf.3, %lf, %lf, %lf\n", s, ax, ay, az);


        /*  CODE SECTION 1 */
        printf("At %d ms, the acceleration's magnitude was: %lf\n", f, magnitude(ax, ay, az));


        /*  CODE SECTION 2 */

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

    }
```

Screenshot 2(Code-Part 2)



```
}

/* Put your functions here */
int minutes(int q){
    int h,o;
    h=q/(60*60*1000);
    q=q-h*(60*60*1000);
    o=q/(60*1000);
    return o;
}
int seconds(int w){
    int h,m,s;
    h=w/(60*60*1000);
    w=w-h*(60*60*1000);
    m=w/(60*1000);
    w=w-m*(60*1000);
    s=w/1000;
    return s;
}
int milliseconds(int r){
    int h,m,s;
    h=r/(60*60*1000);
    r=r-h*(60*60*1000);
    m=r/(60*1000);
    r=r-m*(60*1000);
    s=r/1000;
    r=r-(s*1000);
    return r;
}

/**
 * Calculates and returns the magnitude of three given values.
 *
 * @param x - The x-axis scanned values from the DS4 controller.
 * @param y - The y-axis scanned values from the DS4 controller.
 * @param z - The z-axis scanned values from the DS4 controller.
 * @return - The magnitude of the given values.
 */
double magnitude(double x, double y, double z)
{
    // Step 8, uncomment and modify the next line
    return sqrt((x*x)+(y*y)+(z*z) );
}
```

Screenshot 3(Output)



```
At 3598 ms, the acceleration's magnitude was: 0.987419
At 0.000000 minutes, 0.000000 seconds, and 598 milliseconds it was: 0.987419
Echoing output: 3.000000.3, -0.007080, 0.964844, 0.220337
At 3602 ms, the acceleration's magnitude was: 0.989708
At 0.000000 minutes, 0.000000 seconds, and 602 milliseconds it was: 0.989708
Echoing output: 3.000000.3, -0.009766, 0.965454, 0.218262
At 3606 ms, the acceleration's magnitude was: 0.989866
At 0.000000 minutes, 0.000000 seconds, and 606 milliseconds it was: 0.989866
Echoing output: 3.000000.3, -0.008423, 0.965942, 0.219971
At 3610 ms, the acceleration's magnitude was: 0.990708
At 0.000000 minutes, 0.000000 seconds, and 610 milliseconds it was: 0.990708
Echoing output: 3.000000.3, -0.007568, 0.963379, 0.218384
At 3614 ms, the acceleration's magnitude was: 0.987850
At 0.000000 minutes, 0.000000 seconds, and 614 milliseconds it was: 0.987850
Echoing output: 3.000000.3, -0.005981, 0.964722, 0.220459
At 3618 ms, the acceleration's magnitude was: 0.989609
At 0.000000 minutes, 0.000000 seconds, and 618 milliseconds it was: 0.989609
Echoing output: 3.000000.3, -0.006836, 0.961914, 0.219604
At 3622 ms, the acceleration's magnitude was: 0.986687
At 0.000000 minutes, 0.000000 seconds, and 622 milliseconds it was: 0.986687
Echoing output: 3.000000.3, -0.007690, 0.965332, 0.218872
At 3626 ms, the acceleration's magnitude was: 0.989864
At 0.000000 minutes, 0.000000 seconds, and 626 milliseconds it was: 0.989864
Echoing output: 3.000000.3, -0.008423, 0.964478, 0.220337
At 3630 ms, the acceleration's magnitude was: 0.989362
At 0.000000 minutes, 0.000000 seconds, and 630 milliseconds it was: 0.989362
Echoing output: 3.000000.3, -0.009033, 0.967285, 0.217529
At 3634 ms, the acceleration's magnitude was: 0.991484
At 0.000000 minutes, 0.000000 seconds, and 634 milliseconds it was: 0.991484
Echoing output: 3.000000.3, -0.012695, 0.964600, 0.216675
At 3638 ms, the acceleration's magnitude was: 0.988718
At 0.000000 minutes, 0.000000 seconds, and 638 milliseconds it was: 0.988718
Echoing output: 3.000000.3, -0.011841, 0.964844, 0.219604
At 3642 ms, the acceleration's magnitude was: 0.989591
At 0.000000 minutes, 0.000000 seconds, and 642 milliseconds it was: 0.989591
Echoing output: 3.000000.3, -0.010498, 0.962891, 0.219482
At 3646 ms, the acceleration's magnitude was: 0.987644
At 0.000000 minutes, 0.000000 seconds, and 646 milliseconds it was: 0.987644
Echoing output: 3.000000.3, -0.010498, 0.966064, 0.221436
At 3650 ms, the acceleration's magnitude was: 0.991173
At 0.000000 minutes, 0.000000 seconds, and 650 milliseconds it was: 0.991173
Echoing output: 3.000000.3, -0.010986, 0.964600, 0.220947
At 3654 ms, the acceleration's magnitude was: 0.989642
At 0.000000 minutes, 0.000000 seconds, and 654 milliseconds it was: 0.989642
Echoing output: 3.000000.3, -0.011475, 0.965088, 0.221558
At 3658 ms, the acceleration's magnitude was: 0.990260
At 0.000000 minutes, 0.000000 seconds, and 658 milliseconds it was: 0.990260
Echoing output: 3.000000.3, -0.011353, 0.963989, 0.221191
At 3662 ms, the acceleration's magnitude was: 0.989105
At 0.000000 minutes, 0.000000 seconds, and 662 milliseconds it was: 0.989105
Echoing output: 3.000000.3, -0.012085, 0.966797, 0.219971
At 3666 ms, the acceleration's magnitude was: 0.991579
At 0.000000 minutes, 0.000000 seconds, and 666 milliseconds it was: 0.991579
Echoing output: 3.000000.3, -0.007080, 0.962158, 0.221313
At 3670 ms, the acceleration's magnitude was: 0.987308
At 0.000000 minutes, 0.000000 seconds, and 670 milliseconds it was: 0.987308
Echoing output: 3.000000.3, -0.007202, 0.964478, 0.220215
```

Problem 3

Run the ds4.exe file and count the different types of directions provided to the joystick.

Analysis

The ds4rd.exe file is already included. We need to provide a logic so that the code counts the different number of times the joystick has been moved.

Design

- Run the command `/ds4rd.exe -d 054c:05c4 -D DS4_BT -b`
- Move the joystick around to get different values.

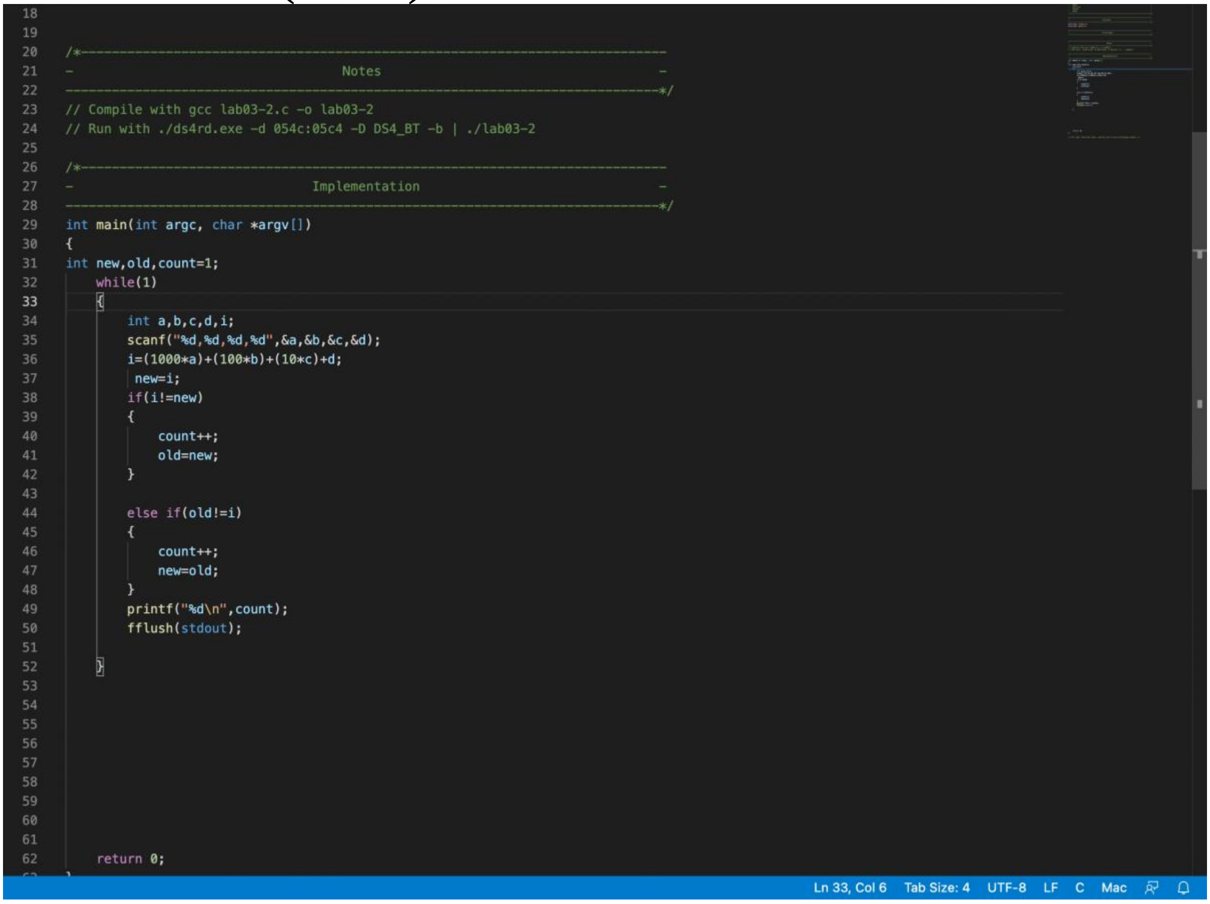
Testing

Manually count the variations given to the joystick and compare that with the values we get.

Comments

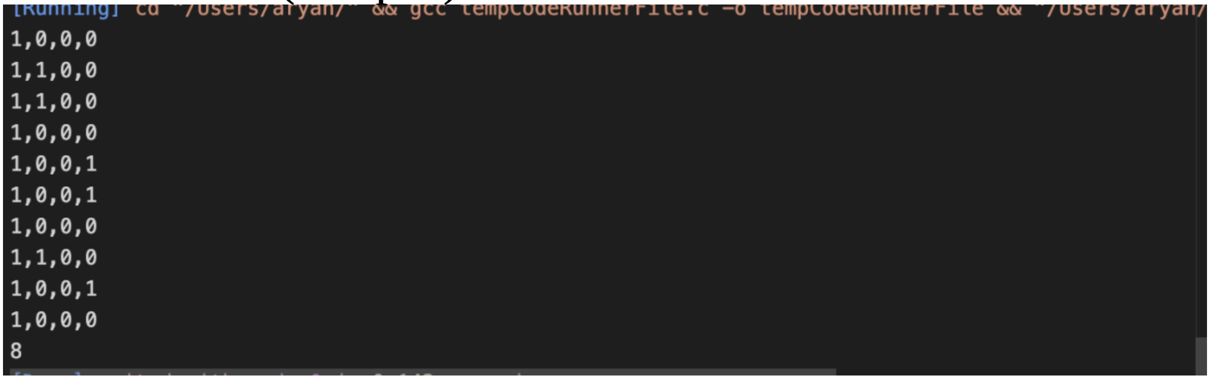
Make sure to include fflush(stdout) statement in your code.

Screenshot 1(Code)



```c


/*----------------------------------------------------------------------------
-                                   Notes                                    -
----------------------------------------------------------------------------*/
// Compile with gcc lab03-2.c -o lab03-2
// Run with ./ds4rd.exe -d 054c:05c4 -D DS4_BT -b | ./lab03-2

/*----------------------------------------------------------------------------
-                               Implementation                               -
----------------------------------------------------------------------------*/
int main(int argc, char *argv[])
{
int new,old,count=1;
    while(1)
    {
        int a,b,c,d,i;
        scanf("%d,%d,%d,%d",&a,&b,&c,&d);
        i=(1000*a)+(100*b)+(10*c)+d;
         new=i;
        if(i!=new)
        {
            count++;
            old=new;
        }

        else if(old!=i)
        {
            count++;
            new=old;
        }
        printf("%d\n",count);
        fflush(stdout);

    }









    return 0;
}
```

Screenshot 2(Output)



```
[Running] cd "/Users/aryan/" && gcc tempCodeRunnerFile.c -o tempCodeRunnerFile && "/Users/aryan/
1,0,0,0
1,1,0,0
1,1,0,0
1,0,0,0
1,0,0,1
1,0,0,1
1,0,0,0
1,1,0,0
1,0,0,1
1,0,0,0
8
```