Getting to Know Your DualShock 4

Lab 1
Section 1

Submitted By:

Abhaya Neupane

Submission Date:

09/08/2023

1 Old Computers

1.1 MITS Altair 8800
Input Device= front panel switches
Output Device= front panel LEDs
Minimum RAM=256 bytes/2048bits
Maximum RAM=64 kilobytes/65536bytes/524288bits
CPU= Intel 8080, 2.0 MHz

1.2 MOS KIM-1
Input Device= on-board hexadecimal keypad
Output Device= 6 digital LED display
RAM=1024 bytes/8192bits
CPU= MOS 6502, 1MHz

1.3 Apple 1
Input Device= keyboard interface port
Output Device= monochrome 280 X 192, 40 X 24 text
Minimum RAM=4 kilobytes/4096 bytes/32768 bits
Maximum RAM=64 kilobytes/65536bytes/524288bits
CPU= MOS 6502, 1MHz

1.4 IBM Personal Computer (PC) 5150
Input Device= keyboard
Output Device= 80X24 text
Minimum RAM=16 kilobytes/16384 Bytes/131072 bits
Maximum RAM=640 kilobytes/655360 Bytes/5242880bits
CPU= Intel 8088, 4.77 MHz

1.5 Apple Macintosh
Input Device= keyboard, mouse
Output Device= 9-inch monochrome screen 512x342 pixels
Minimum RAM=128 kilobytes/131072 bytes/1048576bits
Maximum RAM=512 kilobytes/524288 bytes/4194304bits
CPU= Motorola 68000, 7.83 MHz

2 Base Conversion

Decimal 1, 10, 42, 255

- Binary

1 =

8	4	2	1
0	0	0	1

= 0001

10 =

8	4	2	1
1	0	1	0

= 1010

42 =

32	16	8	4	2	1
1	0	1	0	1	0

= 101010

255 =

128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1

= 11111111

Octal

1 =

	Quotient	Remainder
1/8	0	1

$= (1)_8$

10 =

10/8	1	2
1/8	0	1

$= (12)_8$

42 =

42/8	5	2
5/8	0	5

$= (52)_8$

	Quotient	Remainder
255 = 255/8	31	7
31/8	3	7
3/8	0	3

$= (377)_8$

Hexadecimal:

$1 = (1)_{16}$

	Quotient	Remainder
1/16	0	1

$10 = (A)_{16}$

10/16	0	10 = A

$42 = (2A)_{16}$

42/16	2	10 = A
2/16	0	2

$255 = (FF)_{16}$

255/16	15	15 = F
15/16	0	15 = F

Hexadecimal F, DF, 81, 04

Binary:

F = 1111

DF = 1101 1111
= 11011111

81 = 8 1
= 1000 0001
= 10000001

04 = 00000100

Hex	Binary	Hex	Binary
0	0000	9	1001
1	0001	A	1010
2	0010	B	1011
3	0011	C	1100
4	0100	D	1101
5	0101	E	1110
6	0110	F	1111
7	0111		
8	1000		

Decimal (Base$_{10}$)

$(F)_{16} = (15 \times 16^0) = (15)_{10}$

$(DF)_{16} = (13 \times 16^1) + (15 \times 16^0) = (223)_{10}$

$(81)_{16} = (8 \times 16^1) + (1 \times 16^0) = (129)_{10}$

$04 = (0 \times 16^1) + (4 \times 16^0) = (4)_{10}$

Octal ((Convert to binary then to Octal)

F = 1111
= 1 111
= 1 7

DF = 1101 1111
= 11 011 111
= 3 3 7

Binary	Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

81 = 1000 0001
= 10 000 001
= 2 0 1

04 = 0000 0100
= 00 000 100
= 0 0 4
= 4

Binary: 10010011, 111111

Decimal:

$$10010011 = (1 \times 2^7) + (0 \times 2^6) + (0 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) = 147$$

$$111111 = (1 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) = 63$$

Octal:

10010011 = 10 010 011
= 2 2 3
= 223

111111 = 111 111
= 7 7

Binary	Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Hexadecimal: (Referred to the Previous Table)

10010011 = 1001 0011
= 9 3
= 93

111 111 = 11 1111
= 3 F
= 3F

3 Exploration

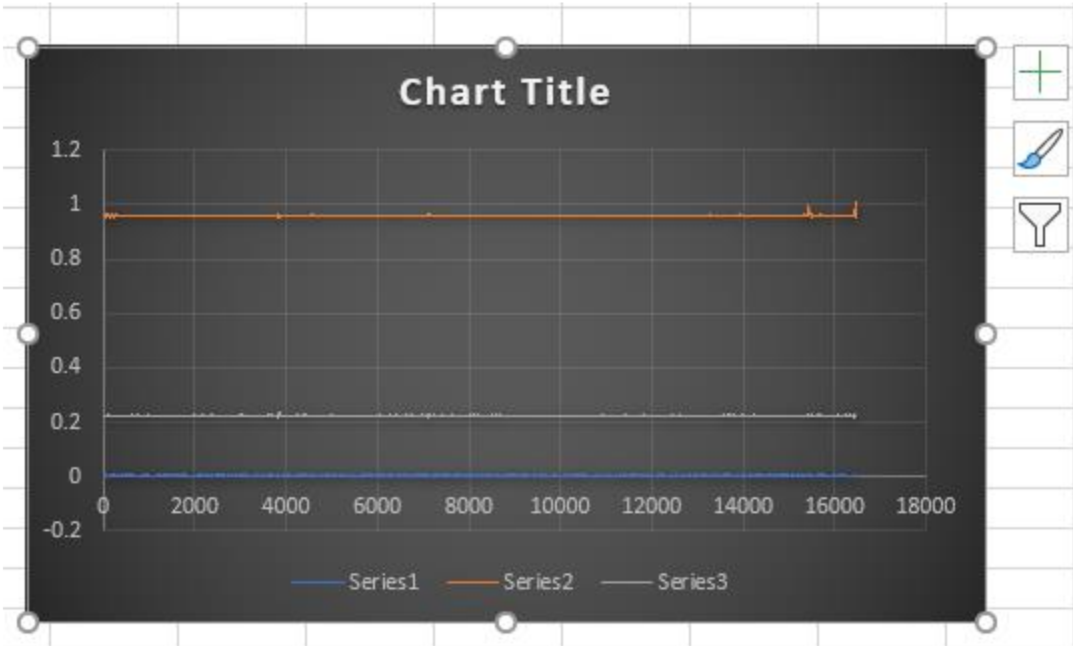

Fig:flat1.csv

Here in this graph we don't see any ups and downs as the controller is lying flat on the surface.

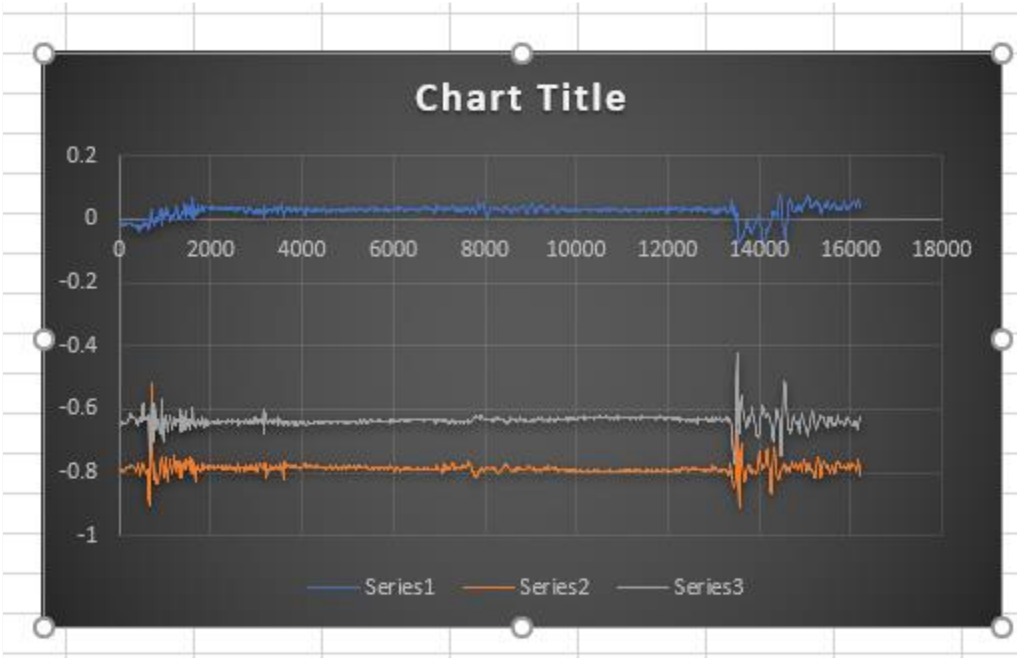

Fig:flat2.csv

Here we see a bit of ups and downs as we are holding the controller in this case and during the time we hold the controller the coordinates have negative values as the controller is above the resting position on the surface of the table

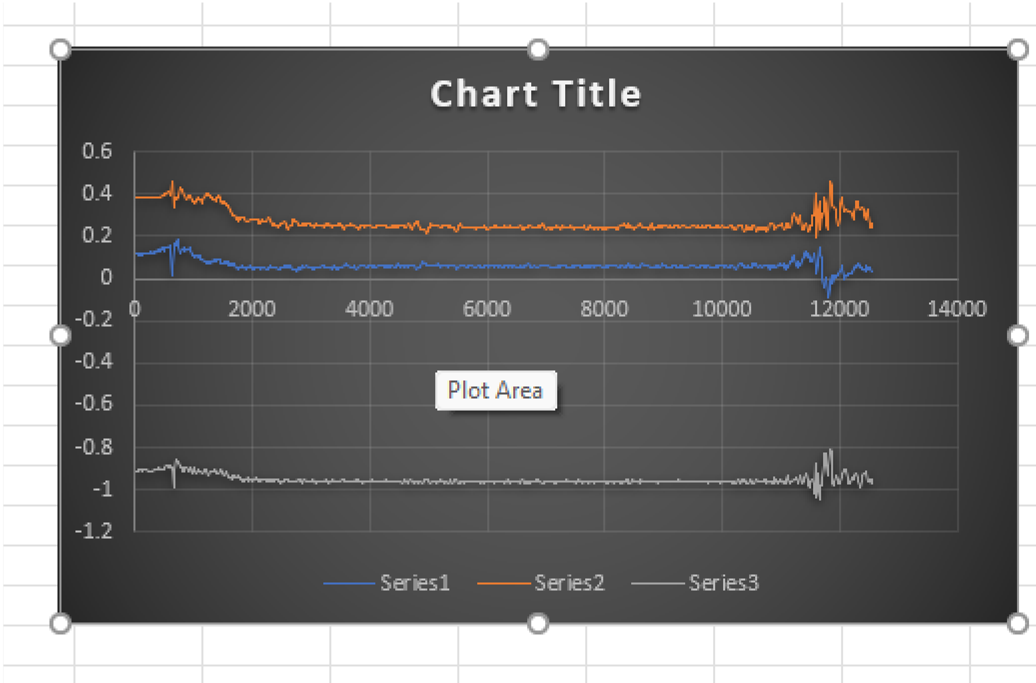

Fig:front1.csv

Here we don't see much ups and downs but one of the value is negative as the controller is pointing upward and is being held by our hand.

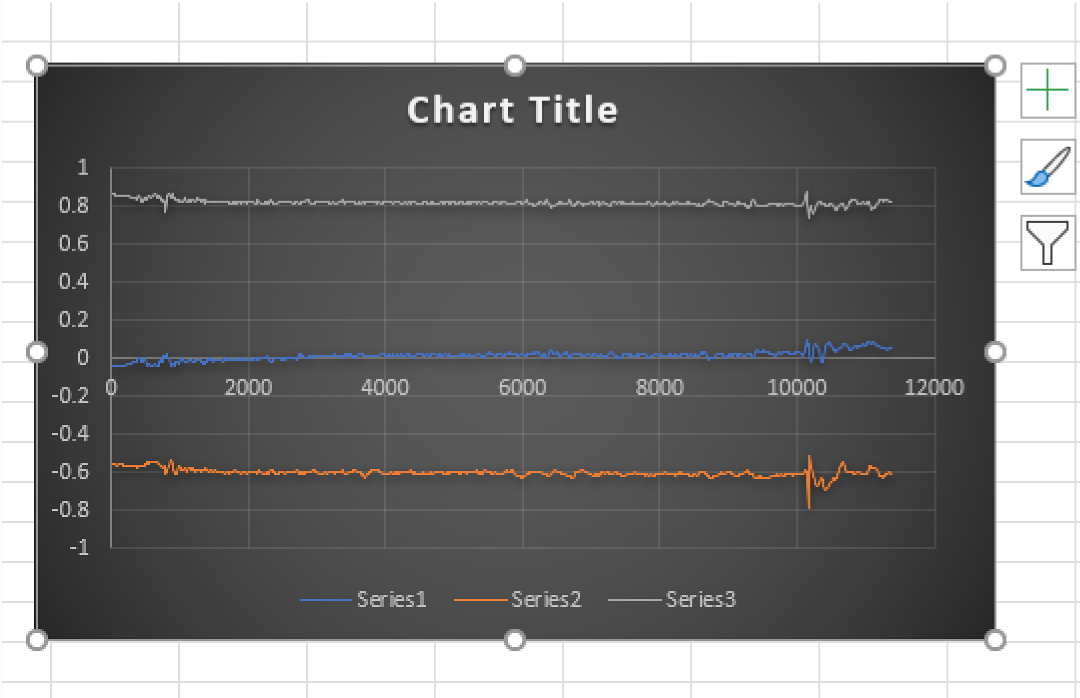

Fig:front2.csv

Here the graph is quiet similar to front1.csv only change is the data that is negative, which is due to the controller being pointed downwards.

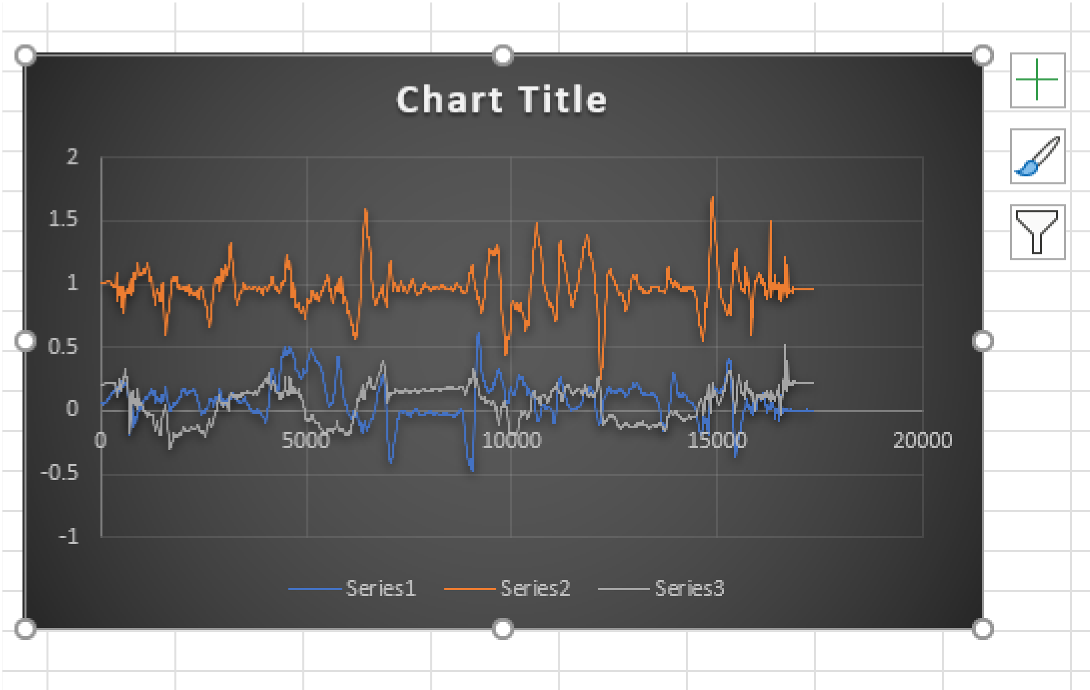

Fig:custom1.csv

In this graph, we see many ups downs which is because the controller was moved up and down in different patterns many times.

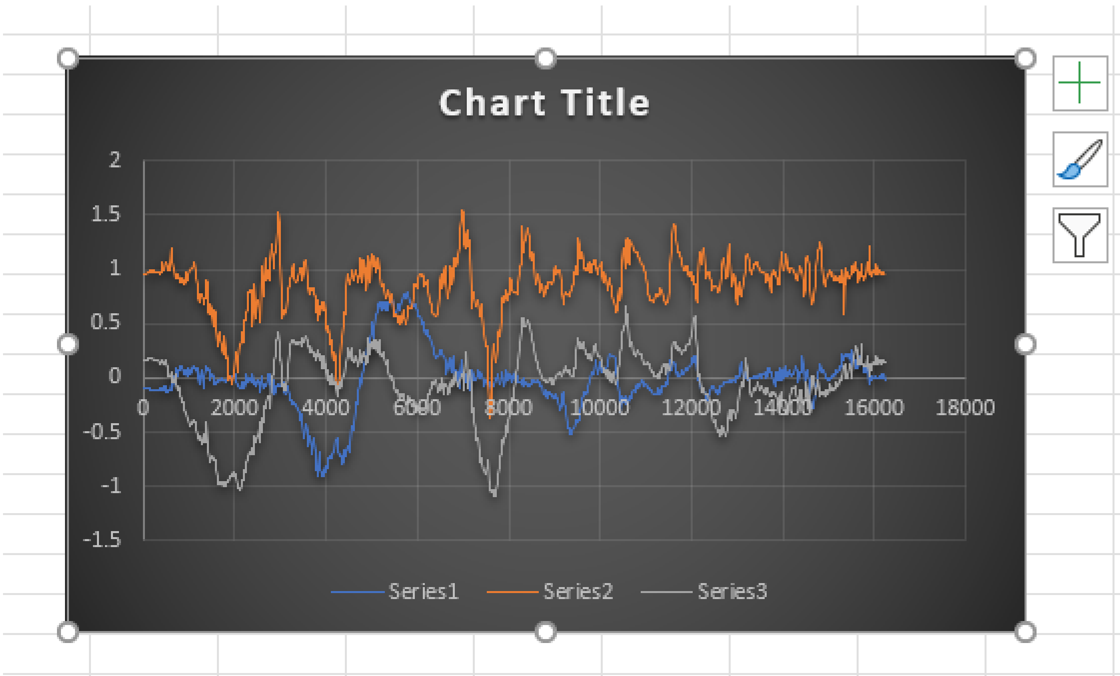

Fig:custom2.csv

In this graph as well, we see a lot of ups and downs as the controller is moved left and right in different patterns.

From, all these graph we can conclude that the data we collected was the real-time position of the controller and the accelerometers present in the controller is providing the location of the controller in coordinates.

4 Joystick Calibration

#4.1

The vertical joystick equation is $f(x) = -x/128$ and the horizontal joystick equation is $f(x') = x'/128$, where x is the value in vertical axis and x' is the value in horizontal axis. The equations aren't similar because as the coordinates move leftward and downward they go towards a more negative value.

#4.2

The center point for the joystick is (-3,2) for the left joystick and (-4,-2) for the right joystick. The center is not 0 because it might not be practically possible to get the center at 0.

#4.3

The joysticks is used a lot and there will be a shift in the center point as we keep on using the joysticks. It might also be because of the way the controller is programmed to not have the center 0.

#4.4

We could change the source code the joystick and recalibrate it, to get the center 0.