CprE 1850 – Lab #1

Maxwell Miller

1.1: MITS Altair 8800

Min RAM: 0.256K; 256 bytes; or 2048 bits

Max RAM: 64K; 64,000 bytes; 512,000 bits

CPU: Intel 8080 (2.0 MHz)

1.2: MOS KIM-1

RAM: 1.024K; 1024 bytes; or 8192 bits

CPU: MOS 6502 (1.0 MHz)

1.3: Apple 1

Min RAM: 4K; 4,000 bytes; or 32,000 bits

Max RAM: 64K; 64,000 bytes; 512,000 bits

CPU: MOS 6502 (1.0 MHz)

1.4: IBM Personal Computer (PC) 5150

Min RAM: 16K; 16,000 bytes; or 128,000 bits

Max RAM: 256K; 256,000 bytes; 2,048,000 bits

CPU: Intel 8088 (4.77 MHz)

1.5: Apple Macintosh

Min RAM: 128K; 128,000 bytes; or 1,024,000 bits

Max RAM: 512K; 512,000 bytes; 4,096,000 bits

CPU: Motorola 68000 (7.83 MHz)

CPPE Lab #1

Decimal	Binary	Octal	hex
1	1	1	1
10	1010	12	A
42	101010	52	2A
255	11111111	377	FF
15	1111	17	F
223	11011111	337	DF
129	10000001	201	81
4	0100	4	04
147	10010011	223	93
63	111111	77	3F

$8^1 \; 8^0$

$10 = 2^3 + 2^1 = 1 + 2 = A$

$42 = 2^5 + 2^3 + 2^1$

$42 = 5 \cdot 8^1 + 2 \cdot 8^0$

$42 = 2 \cdot 16^1 + 10 \cdot 16^0$

$255_{10} = 7$

$11111111 \rightarrow 377$ $\qquad$ $11111111 \rightarrow FF$

$11011111 \rightarrow 337$

$DF = 13 \cdot 16^1 + 15 \cdot 16^0 =$

	Binary						
	0	0	0	0	0	0	0
Decimal	64	32	16	8	4	2	1
	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Octal					
0	0	0	0	0	0
8^5	8^4	512	64	8	1

$10000001 \rightarrow 201 \rightarrow 8 \cdot 16^1 + 1$

$04 \rightarrow 0100 \rightarrow$

$10010011 \rightarrow 93 \rightarrow 223 \rightarrow 9 \cdot 16 + 3$

$00111111 \rightarrow 3F$

$111111 = 77$

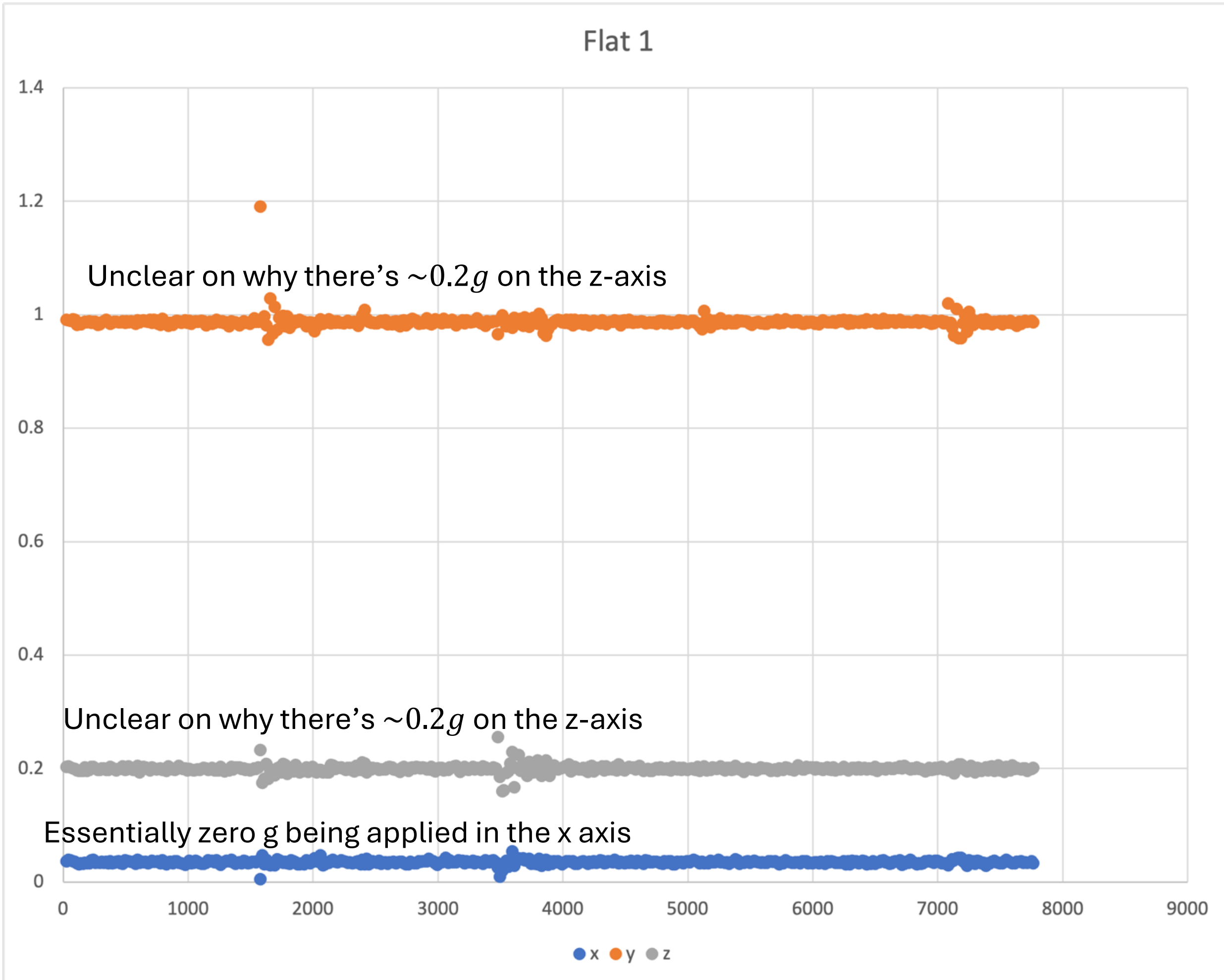
$111111 \rightarrow 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 63$

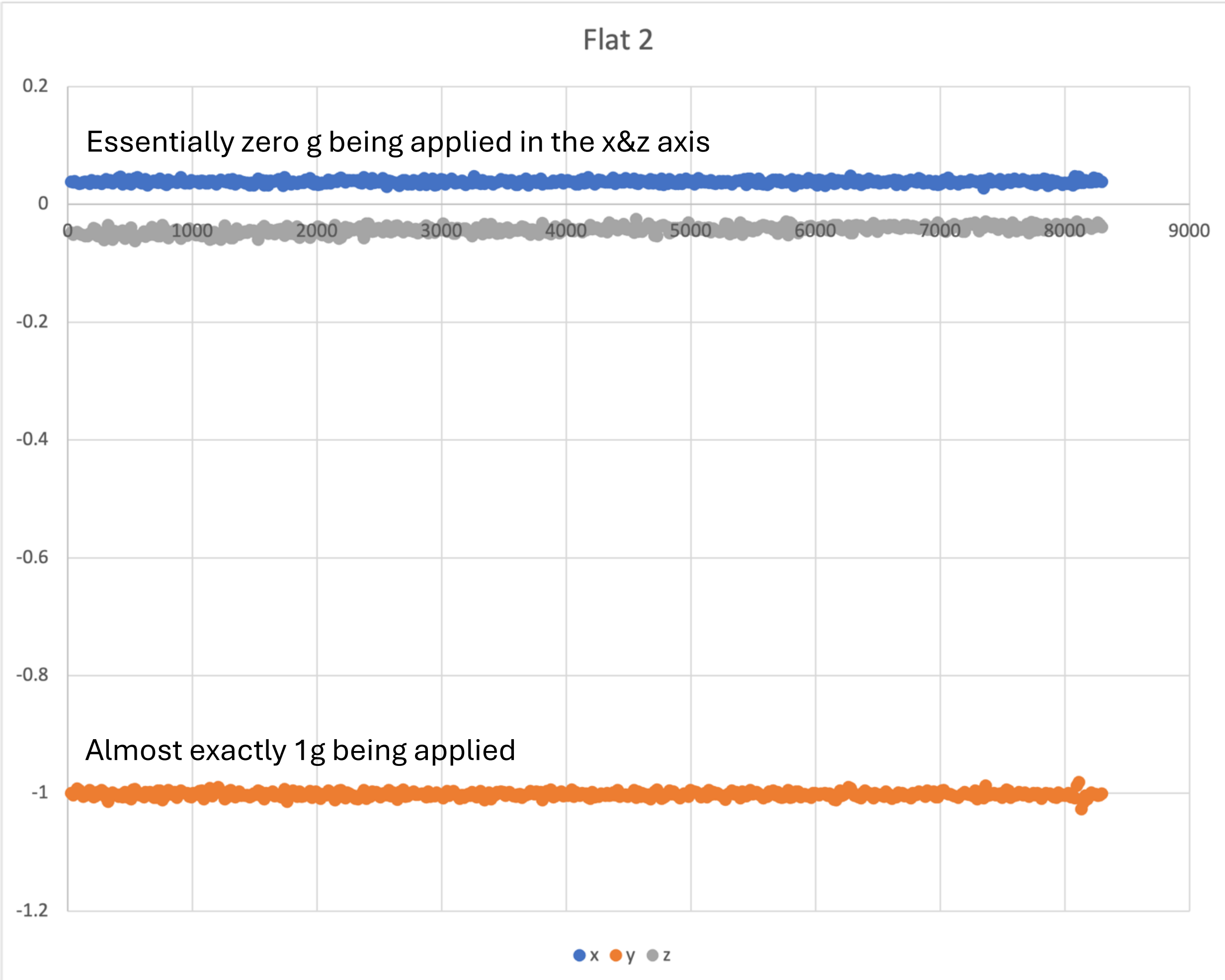
hex

hex	0–9	A–F
deci	0–9	10–15

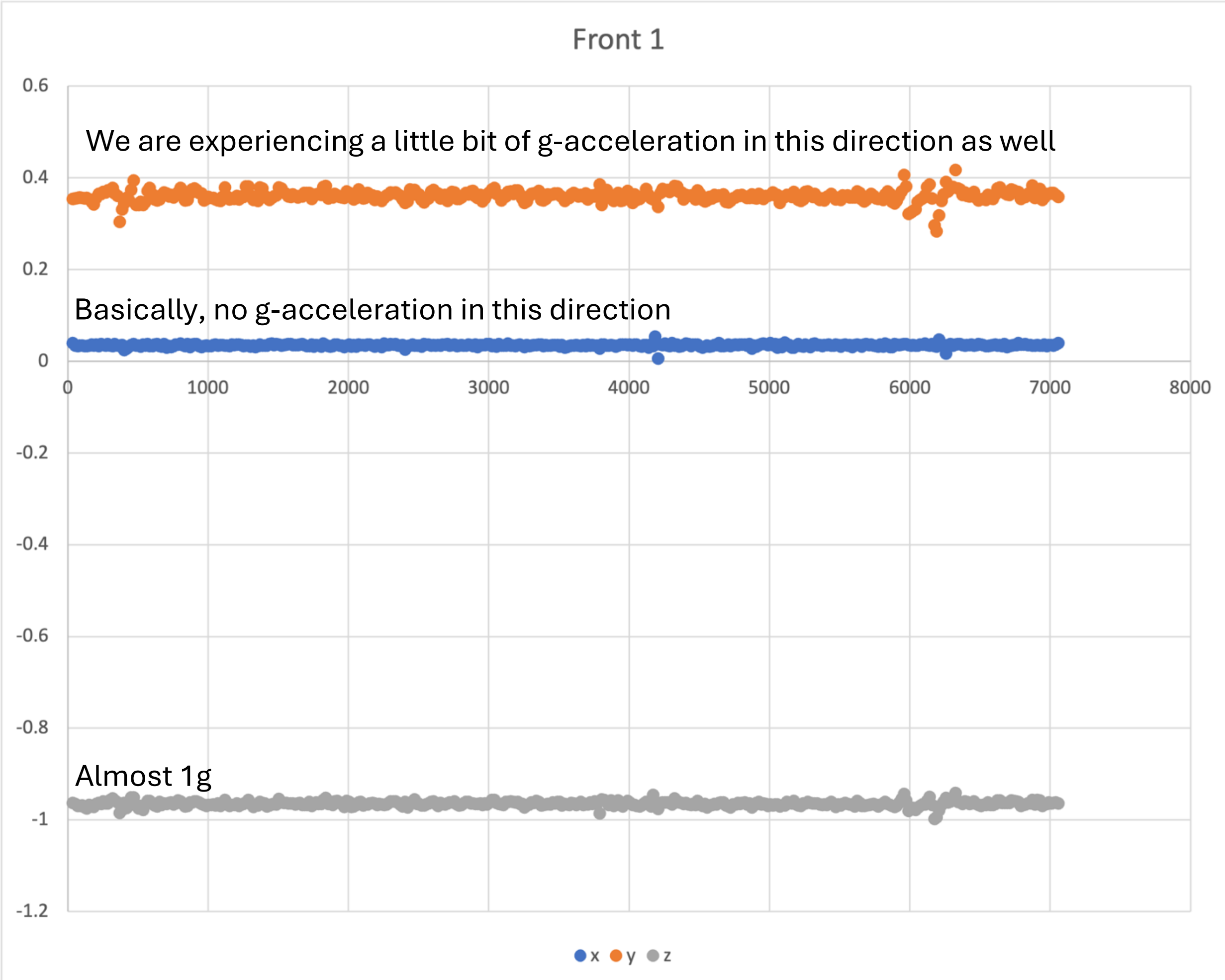
n_{10}	n_8	$3 \times n_2$
0	0	000
1	1	001
2	2	010
3	3	011
4	4	100
5	5	101
6	6	110
7	7	111

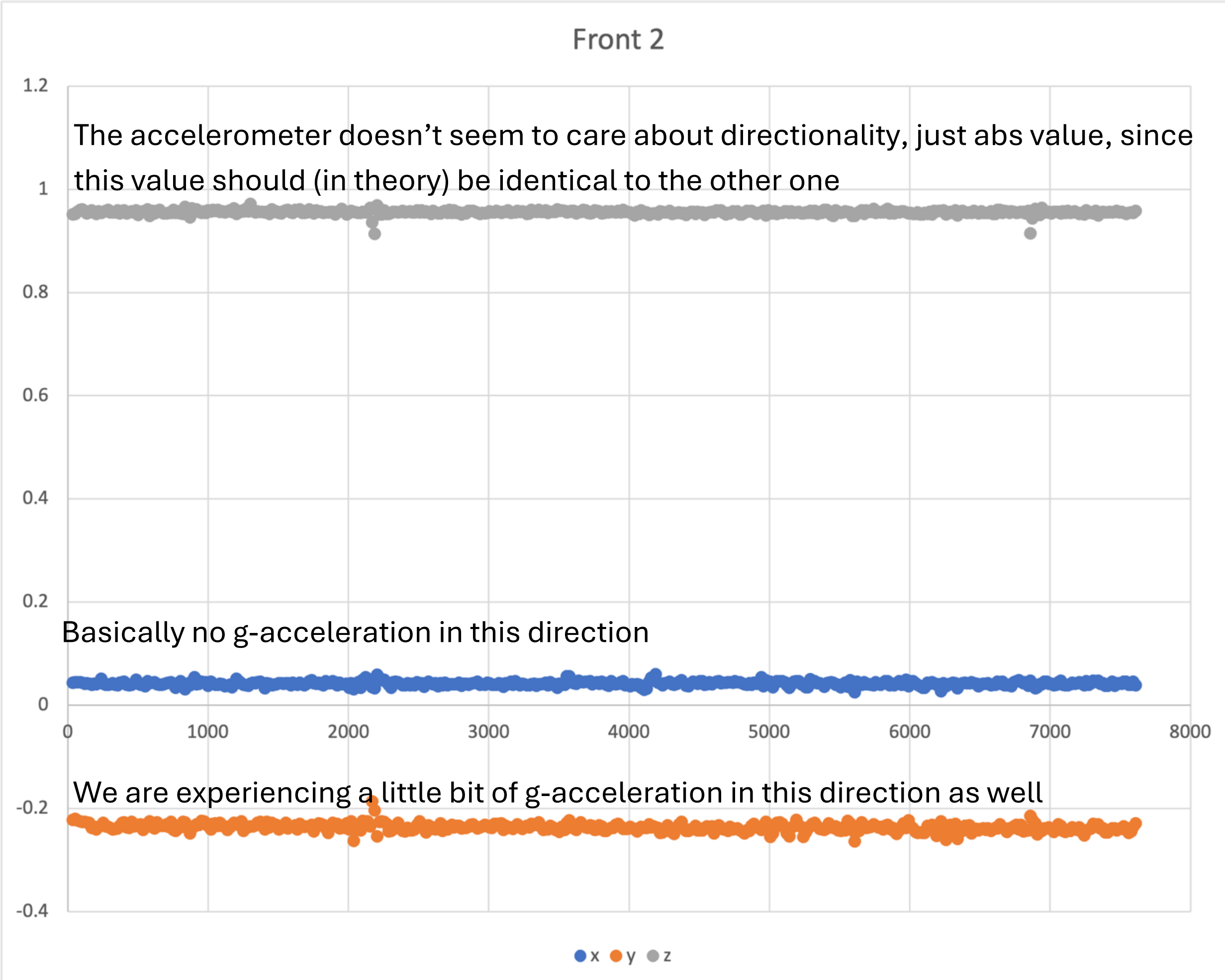
n_{10}	n_{16}	$4 \times n_2$
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

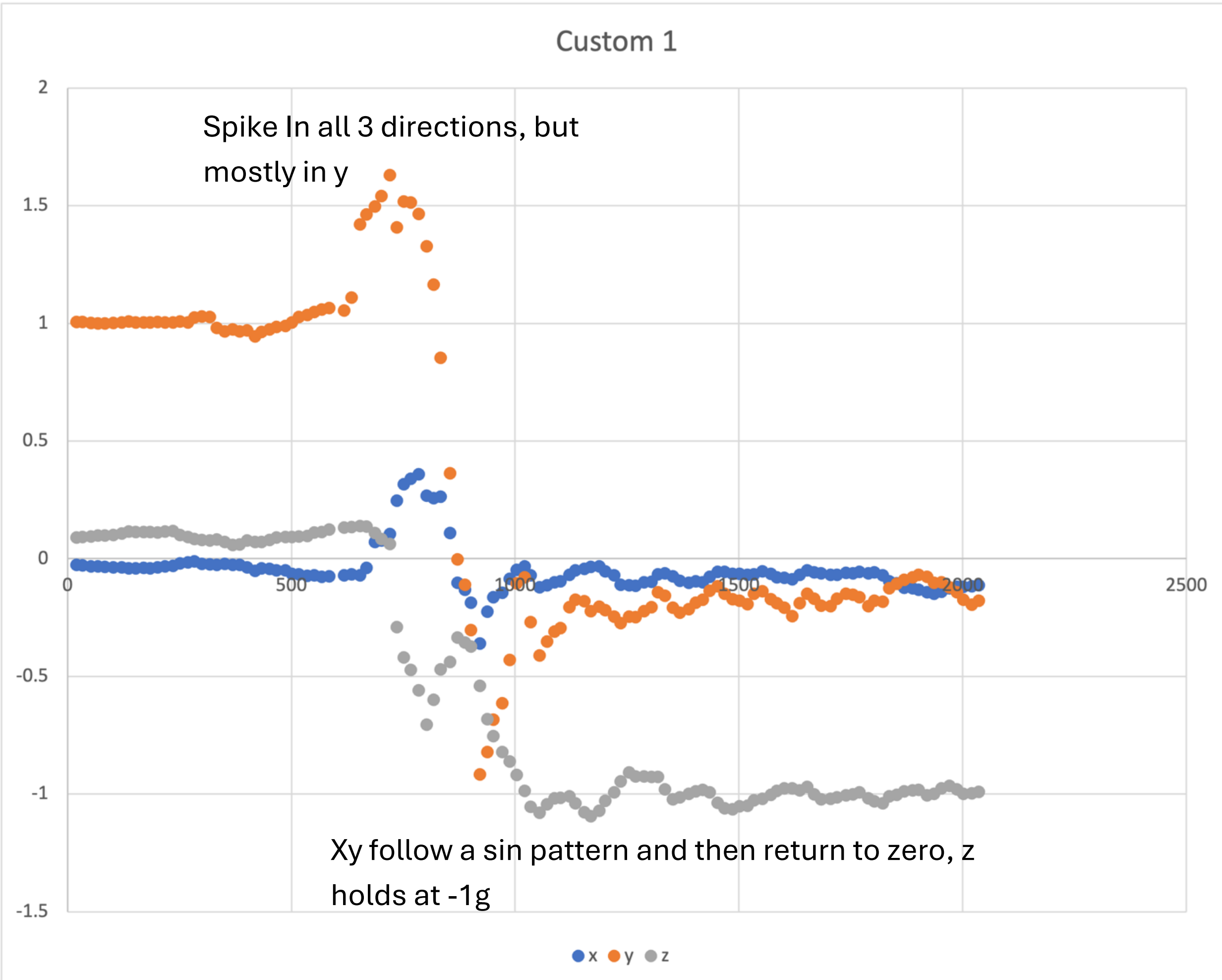

Flat 1
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Unclear on why there's ~$0.2g$ on the z-axis
Unclear on why there's ~$0.2g$ on the z-axis
Essentially zero g being applied in the x axis
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
x
y
z



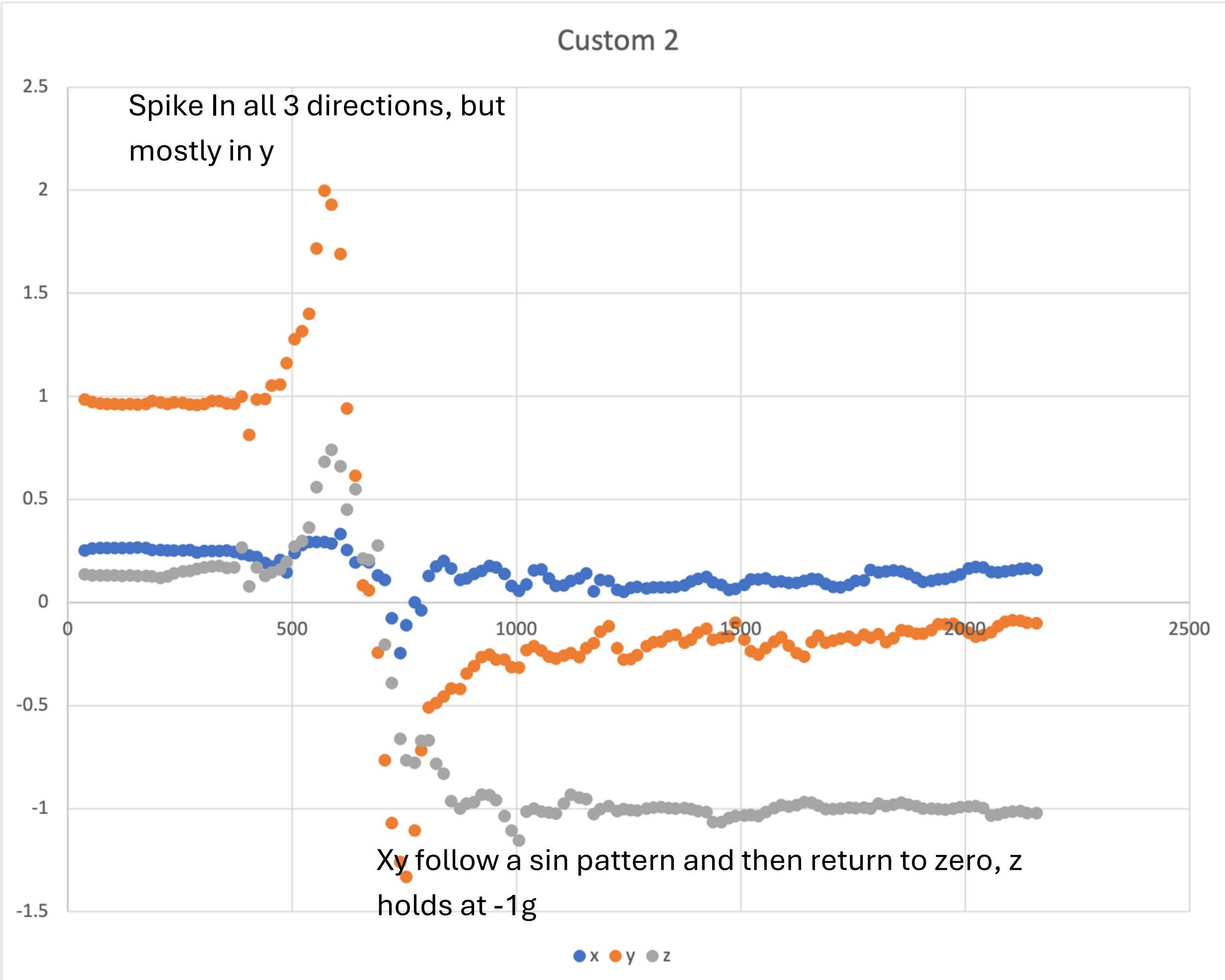
Flat 2
Essentially zero g being applied in the x&z axis
Almost exactly 1g being applied
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
x
y
z


Front 1
We are experiencing a little bit of g-acceleration in this direction as well
Basically, no g-acceleration in this direction
Almost 1g
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
0
1000
2000
3000
4000
5000
6000
7000
8000
x
y
z


Front 2
The accelerometer doesn't seem to care about directionality, just abs value, since this value should (in theory) be identical to the other one
Basically no g-acceleration in this direction
We are experiencing a little bit of g-acceleration in this direction as well
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0
1000
2000
3000
4000
5000
6000
7000
8000
x
y
z



Custom 1
Spike In all 3 directions, but
mostly in y
Xy follow a sin pattern and then return to zero, z
holds at -1g
2
1.5
1
0.5
0
-0.5
-1
-1.5
0
500
1000
1500
2000
2500
x
y
z


Custom 2
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
0
500
1000
1500
2000
2500
Spike In all 3 directions, but
mostly in y
Xy follow a sin pattern and then return to zero, z
holds at -1g
x
y
z

Graph Questions

1. I believe that each column is the relative gravitational acceleration excreted on the controller.
2. Most likely "-t" is to record pan/tilt/roll acceleration and "-g" makes it relative to the gravitational content of $g = 9.8\,{}^{m}/_{s^2}$, so when the data is 1, it means the applied acceleration is g.
3. It is measuring in g-acceleration, or relative gravitational acceleration.
4. See graphs above.

4. Joystick Calibration

1. What are you vertical and horizontal joystick equations? Are they similar or not? Why or why not?
    a. Both were: $f(x) = \frac{x}{128}$. It's likely that both directions use the same sensor, resulting in the same 8-bit (1-byte) readout in both directions, resulting in the -128 to 127 range, having the first bit determine if the number is positive or negative, and the following bits representing 2^0 through 2^6
2. What did you find as the center point? Explain why it is or is not 0?
    a. It's not zero (as I explained above in 4.1), it's between 0 and 1. Practically this resulted in the sensor sometimes reading out as 0 and sometimes as 1.
3. What could cause the center to not be 0?
    a. See 4.1 and 4.2
4. What could you change to make the center be 0?
    a. It's never going to be perfectly zero due to physical limitations, along with the way binary works.