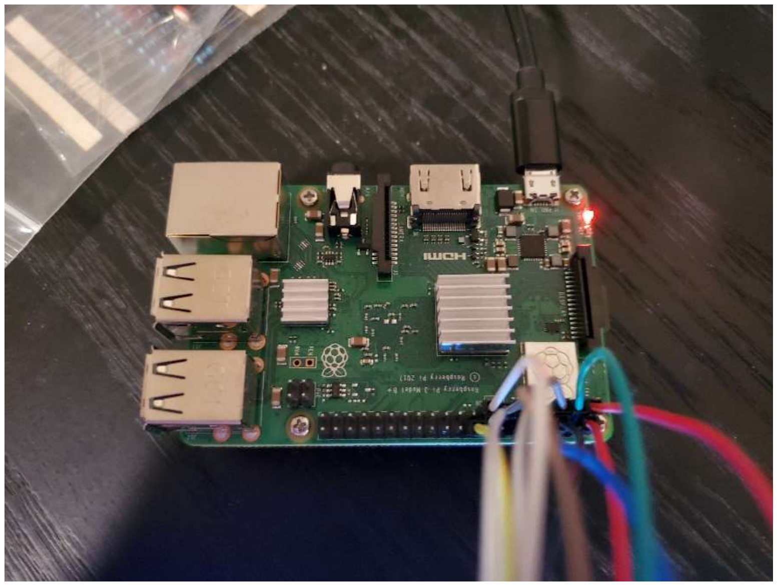
Jordan Ditzler

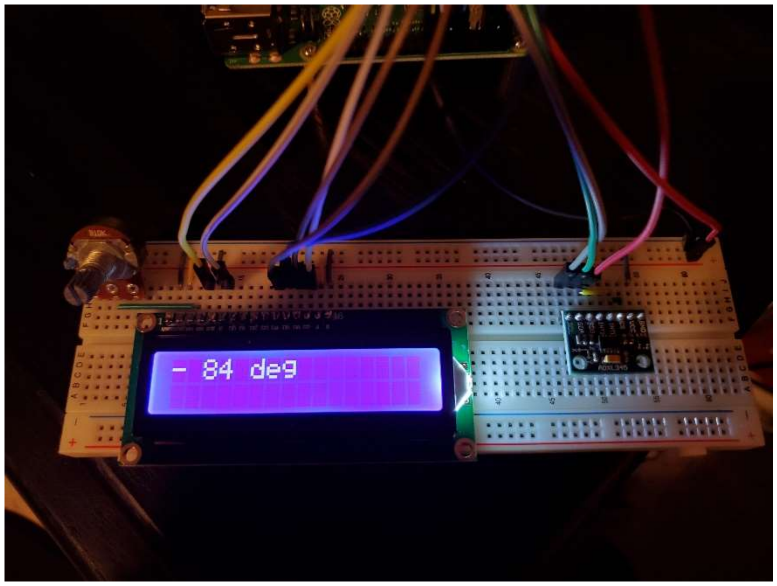
G00967092

Lab 6 Report

November 8, 2020

Photo of RPi & Board





LCD:

- RS: GPIO24
- E: GPIO23
- D4: GPIO22
- D5: GPIO27
- D6: GPIO18
- D7: GPIO17

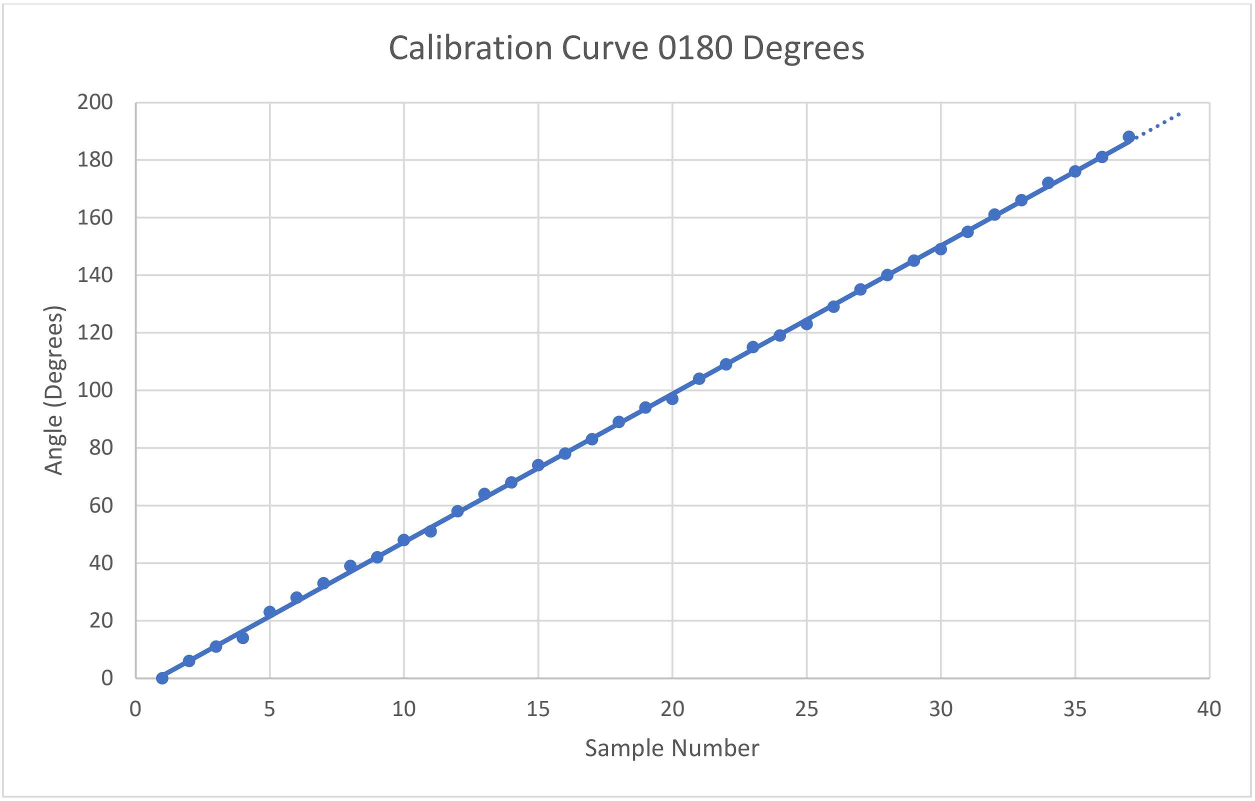
ADXL:

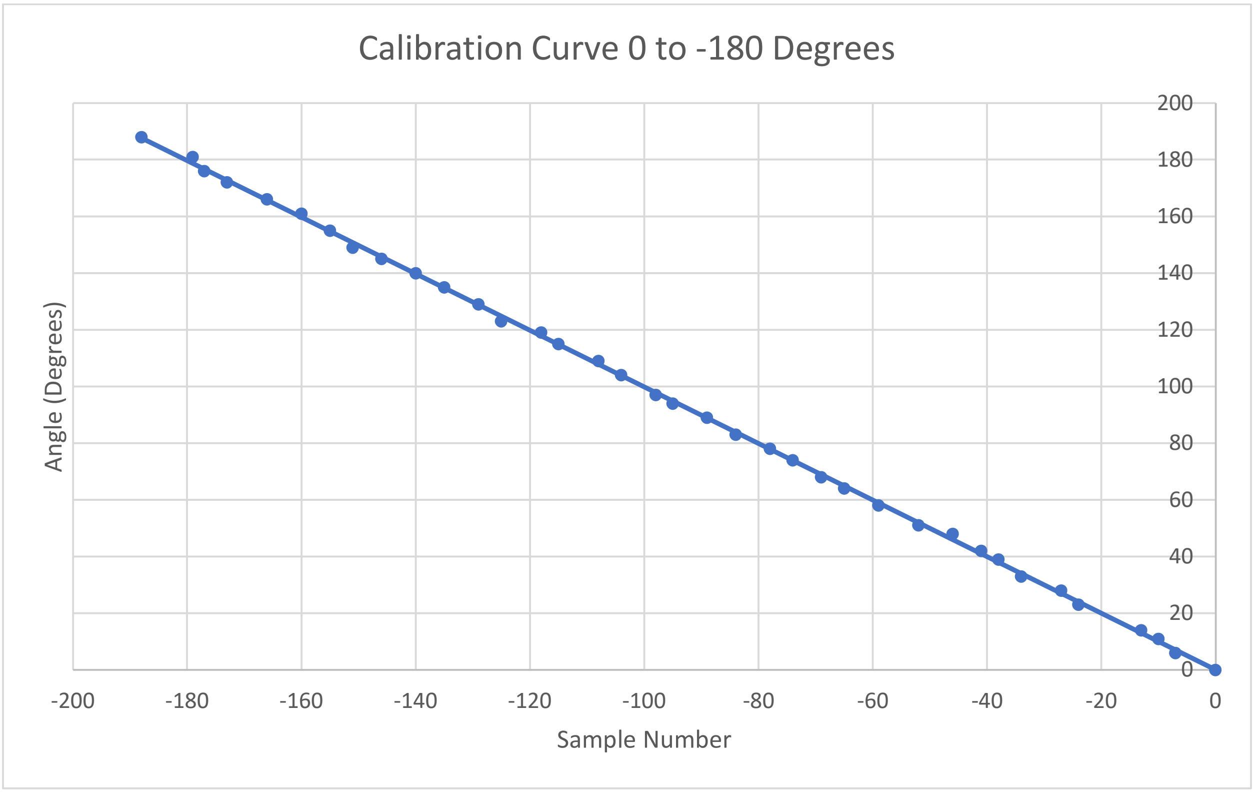
- SCL: GPIO3
- SDA: GPIO2

Tables

Expected (°)	Actual (°)
0	0
5	6
10	11
15	14
20	23
25	28
30	33
35	39
40	42
45	48
50	51
55	58
60	64
65	68
70	74
75	78
80	83
85	89
90	94
95	97
100	104
105	109
110	115
115	119
120	123
125	129
130	135
135	140
140	145
145	149
150	155
155	161
160	166
165	172
170	176
175	181
180	188

Expected (°)	Actual (°)
0	0
-5	-7
-10	-10
-15	-13
-20	-24
-25	-27
-30	-34
-35	-38
-40	-41
-45	-46
-50	-52
-55	-59
-60	-65
-65	-69
-70	-74
-75	-78
-80	-84
-85	-89
-90	-95
-95	-98
-100	-104
-105	-108
-110	-115
-115	-118
-120	-125
-125	-129
-130	-135
-135	-140
-140	-146
-145	-151
-150	-155
-155	-160
-160	-166
-165	-173
-170	-177
-175	-179
-180	-188


Calibration Curve 0180 Degrees
Angle (Degrees)
Sample Number
0
20
40
60
80
100
120
140
160
180
200
0
5
10
15
20
25
30
35
40


Calibration Curve 0 to -180 Degrees
Angle (Degrees)
Sample Number
0
20
40
60
80
100
120
140
160
180
200
-200
-180
-160
-140
-120
-100
-80
-60
-40
-20
0

Calculations

Expected Angle: 127 °

$$1.146031746 * 127 + 1.028751609 = 131.7974861$$

Expected Angle: 92 °

$$1.146031746 * 92 + 1.028751609 = 95.79117975$$

Expected Angle: 28 °

$$1.146031746 * 92 + 1.028751609 = 29.95107679$$

Expected Angle: 171 °

$$1.146031746 * 92 + 1.028751609 = 177.0625568$$

Expected Angle: 103 °

$$1.146031746 * 92 + 1.028751609 = 107.1074474$$

Code

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <unistd.h>

#include <wiringPiI2C.h>
#include <wiringPi.h>
#include <lcd.h>

#define PI 3.141592654

#define ADX_ADDR    0x1D
#define ADX_DFMT    0x31
#define ADX_PCTL    0x2D

#define ADX_XREG    0x32
#define ADX_YREG    0x34
#define ADX_ZREG    0x36

#define ADX_XOFF    0x1E
#define ADX_YOFF    0x1F
#define ADX_ZOFF    0x20

#define LCD_RS      5
#define LCD_E       4
#define LCD_D1      0
#define LCD_D2      1
#define LCD_D3      2
#define LCD_D4      3

short int readADXAxis(int axis_reg, int ADX_fd)
{
    short int data0, data1;

    // Read both axis registers
    data0 = wiringPiI2CReadReg8(ADX_fd, axis_reg);
    data1 = wiringPiI2CReadReg8(ADX_fd, axis_reg+1);

    // Return data variables in 1 integer
    return data1<<8 | data0;
}

short int calibrateAXD(int axis_reg, int ADX_fd)
{
    int i, val;

    // Get average values of axis
    for (i = 0; i < 100; i++)
    {
        val += readADXAxis(axis_reg, ADX_fd);
    }

    return val / 100;
}
```

```
int main(void)
{
    int ADX_fd, LCD_fd;
    short int offset;
    double datax, datay, degree;
    char sign;

    // Checks for errors
    if (wiringPiSetup() == -1)
    {
        printf("Setup failed!\n");
        return -1;
    }

    // Sets up LCD
    LCD_fd = lcdInit(2,16,4, LCD_RS,LCD_E,
        LCD_D1,LCD_D2,LCD_D3,LCD_D4,0,0,0,0);

    // Sets up I2C connection
    ADX_fd = wiringPiI2CSetup(ADX_ADDR);

    if (LCD_fd == -1 || ADX_fd == -1)
    {
        printf("Setup failed!\n");
        return -1;
    }

    // Clears LCD
    lcdClear(LCD_fd);
    lcdPosition(LCD_fd, 0, 0);

    // Sets data format to max resolution and max range
    wiringPiI2CWriteReg8(ADX_fd, ADX_DFMT, 0x0B);

    // Sets ADXL to measurement mode
    wiringPiI2CWriteReg8(ADX_fd, ADX_PCTL, 0x08);

    // Delete current offset
    wiringPiI2CWriteReg8(ADX_fd, ADX_XOFF, 0);
    wiringPiI2CWriteReg8(ADX_fd, ADX_YOFF, 0);
    wiringPiI2CWriteReg8(ADX_fd, ADX_ZOFF, 0);

    // Calibrate X-axis
    offset = calibrateAXD(ADX_XREG, ADX_fd) + 256;
    wiringPiI2CWriteReg8(ADX_fd, ADX_XOFF, -((offset)/4));

    // Calibrate Y-axis
    offset = calibrateAXD(ADX_YREG, ADX_fd);
    printf("off: %d\n", offset);
    wiringPiI2CWriteReg8(ADX_fd, ADX_YOFF, -((offset)/4));

    // Calibrate Z-axis
    //offset = calibrateAXD(ADX_ZREG, ADX_fd);
    //wiringPiI2CWriteReg8(ADX_fd, ADX_ZOFF, -((offset-256)/4));

    while (1)
    {
        // Clears LCD
        lcdClear(LCD_fd);

        // Reads registers and inverts them
        datax = readADXAxis(ADX_XREG, ADX_fd)*-1;
        datay = readADXAxis(ADX_YREG, ADX_fd)*-1;

        // Gets sign value
        sign = ((datax > 0 && datay < 0) ||
            (datax > 0 && datay > 0)) ? '+' : '-';

        // Prints degree values
        lcdPrintf(LCD_fd,"%c %1.f deg",sign,
            fabs(atan2(datax,datay) / 3 * 180));

        usleep(100000);
    }
}
```