Custom PCB Design Scope:

Components on PCB:

1. ESP32 Microcontroller with Bluetooth
2. BNO085 accelerometer/gyroscope/magnetometer
3. USB-C port for flashing program and charging
4. On-board rechargeable battery or battery management circuit for 18350 cell
5. Reads data from CUI Devices AMT103-2048-N4000-S encoder or similar
   5.1. Board requires connection jack for this sensor, it will likely be mounted remotely unless PCB can fit within footprint of sensor
6. Four push button momentary switches for the following functions detailed in program section
7. Toggle power switch
8. Please also include any required small components (ex. Pullup resistors) for correct function of the components

9. BME280 Pressure, Humidity, Temperature Sensor (Optional, mainly looking for cost of adding this to design)

Additional System Components Required:

1. Monochrome OLED display
   1.1. Would like to have two options available
      1.1.1. Option 1: 0.96” SSD1306 128x64 pixel (White text on black background) (https://www.aliexpress.us/item/2255799957966981.html?spm=a2g0o.detail.1000060.1.17ca4922cYD7VB&gps-id=pcDetailBottomMoreThisSeller&scm=1007.13339.291025.0&scm_id=1007.13339.291025.0&scm-url=1007.13339.291025.0&pvid=1fc998bf-2762-413a-b6f6-3e367a4413be&_t=gps-id%3ApcDetailBottomMoreThisSeller%2Cscm-url%3A1007.13339.291025.0%2Cpvid%3A1fc998bf-2762-413a-b6f6-3e367a4413be%2Ctpp_buckets%3A668%232846%238112%231997&pdp_npi=3%40dis%21USD%212.67%211.74%21%21%21%21%21%40210324c816898003960677761ec61e%211000000004384370 46%21rec%21US%21&gatewayAdapt=glo2usa)
         1.1.1.1. If all requested data will not fit on that size and remain readable from approximate 6” away, a larger 128x128 1.5” monochrome display can be used (https://www.aliexpress.us/item/3256805334922670.html?pdp_npi=2%40dis%21USD%21%247.52%21%245.56%21%21%21%21%21%402101c59116898003014452208eb14d%211200003340 1755579%21btf&_t=pvid%3A61efffba-4915-4e3d-bee0-0f737d52cc13&afTraceInfo=1005005521237422__pc__pcBridgePPC_xxxxxx__1689800301&spm=a2g0o.ppclist.product.mainProduct&gatewayAdapt=glo2usa)
      1.1.2. Option 2: Larger 2.42” display with area reserved for text pushed from mobile app

(https://www.aliexpress.us/item/3256803601551716.html?spm=a2g0o.productlist.main.25.1c5c4f302Gy0rx&algo_pvid=7a26f171-ff11-43af-8a47-348b62201f13&aem_p4p_detail=202307191403432499456794026740005677186&algo_exp_id=7a26f171-ff11-43af-8a47-348b62201f13-12&pdp_npi=3%40dis%21USD%2120.88%2115.45%21%21%2120.88%21%21%40211bd3cb16898006232014130d07ee%21112000027168660216%21sea%21US%210&curPageLogUid=5U1enlNEKWJD&search_p4p_id=20230719140343249945679402674000567718 6_13)

2. Two red LEDs and one green LED (will be mounted near OLED display, so if display is not mounted on PCB, LEDs will not be either)

Program Description:

Encoder

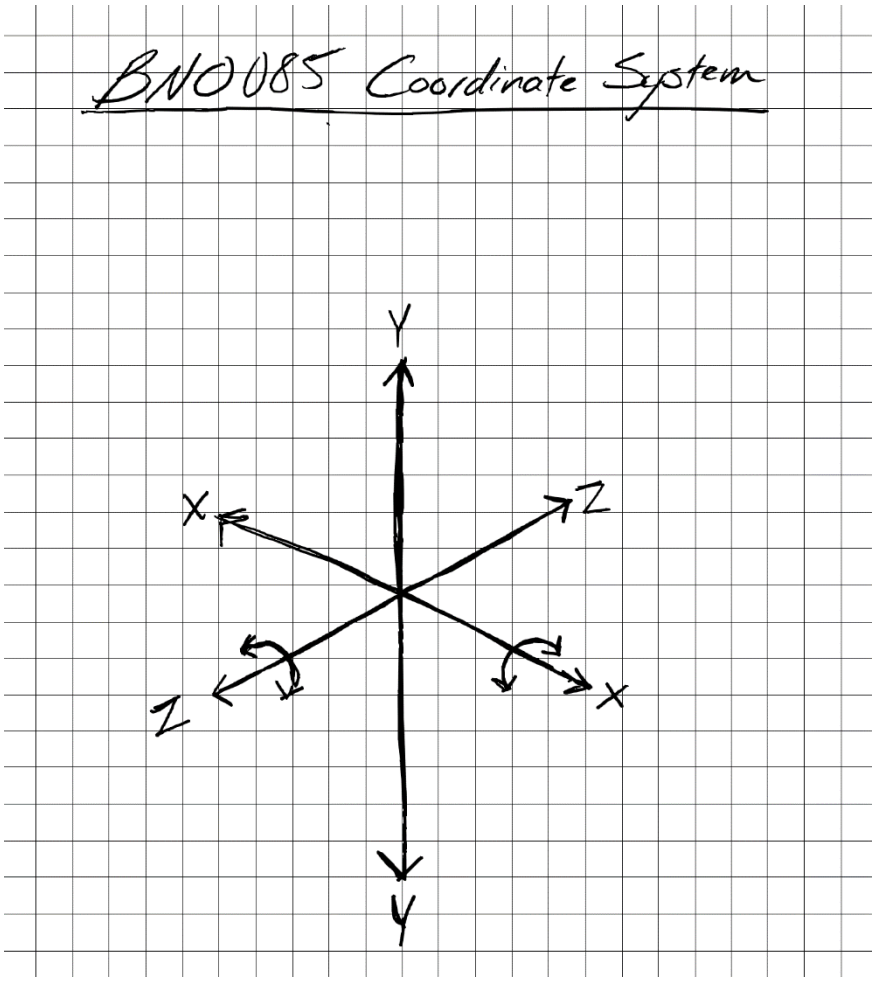
Encoder will be linked to an output shaft that requires precise position data. The output shaft has a number of detents in per revolution (named “CLICKS” going forward), the possible values are 60, 100, and 120 CLICKS per revolution of the output shaft. All of these values will be in the program and the correct value will be selected during setup by the user, using a momentary push button switch (BUTTON 1) to cycle through the options displayed on the screen. I would like the option to name or rename the options easily in the program, while keeping the same values. I would also like to be able to easily add additional options if more are needed in the future (example adding 90 CLICKS per revolution of the output shaft). Because the output shaft can vary in diameter, the ratio of encoder pulses to CLICKS can vary depending on the output shaft size. The program will need a calibration sequence to define the number of encoder pulses in one full revolution of the output shaft. Pressing and holding a momentary push putton switch (BUTTON 1) for longer than 3 seconds will run the calibration process. The calibration process will record the position of the encoder when the calibration starts (position 0) and when it finishes when button is released (position 1). The difference between position 0 and position 1 is the number of encoder pulses in one full revolution of the output shaft, stored in the program as CALIBRATION VALUE. CALIBRATION VALUE must be maintained in non-volatile memory and be maintained when power is lost. The current value of the CLICKS can be rest by pressing and holding a momentary push button switch (BUTTON 2) for 3 seconds.The selection of CLICKS per revolution of the output shaft will also need be stored there. The device should be able to be turned off and not require recalibration when turned back on.

BNO085 Sensor

BNO085 Sensor will be used to determine level of the sensor in relation to earth. Program should keep live reading of degrees off level about 2 axes (X and Z axes in cartesian system see Figure 1). Degrees off level about X axis will be shown with positive and negative values on OLED display. Degrees off level about Z axis will show arrows on OLED display, one arrow on each side of the display, pointing in direction of rotation needed to achieve level about Z axis. When degrees off level about Z axis is greater than 2 degrees, red LEDs will light up. When degrees of level is less than 2 degrees, red LEDs will turn off and single green LED will light up. The value of 2 degrees should be able to be adjusted easily in program to fine tune system

functionality. The level reading should be able to be zeroed out by single pressing and holding BUTTON 3 for 3 seconds for the X axis and pressing and holding BUTTON 4 for 3 seconds for the Z axis.

Figure 1



OLED Display

OLED Display should show the following:

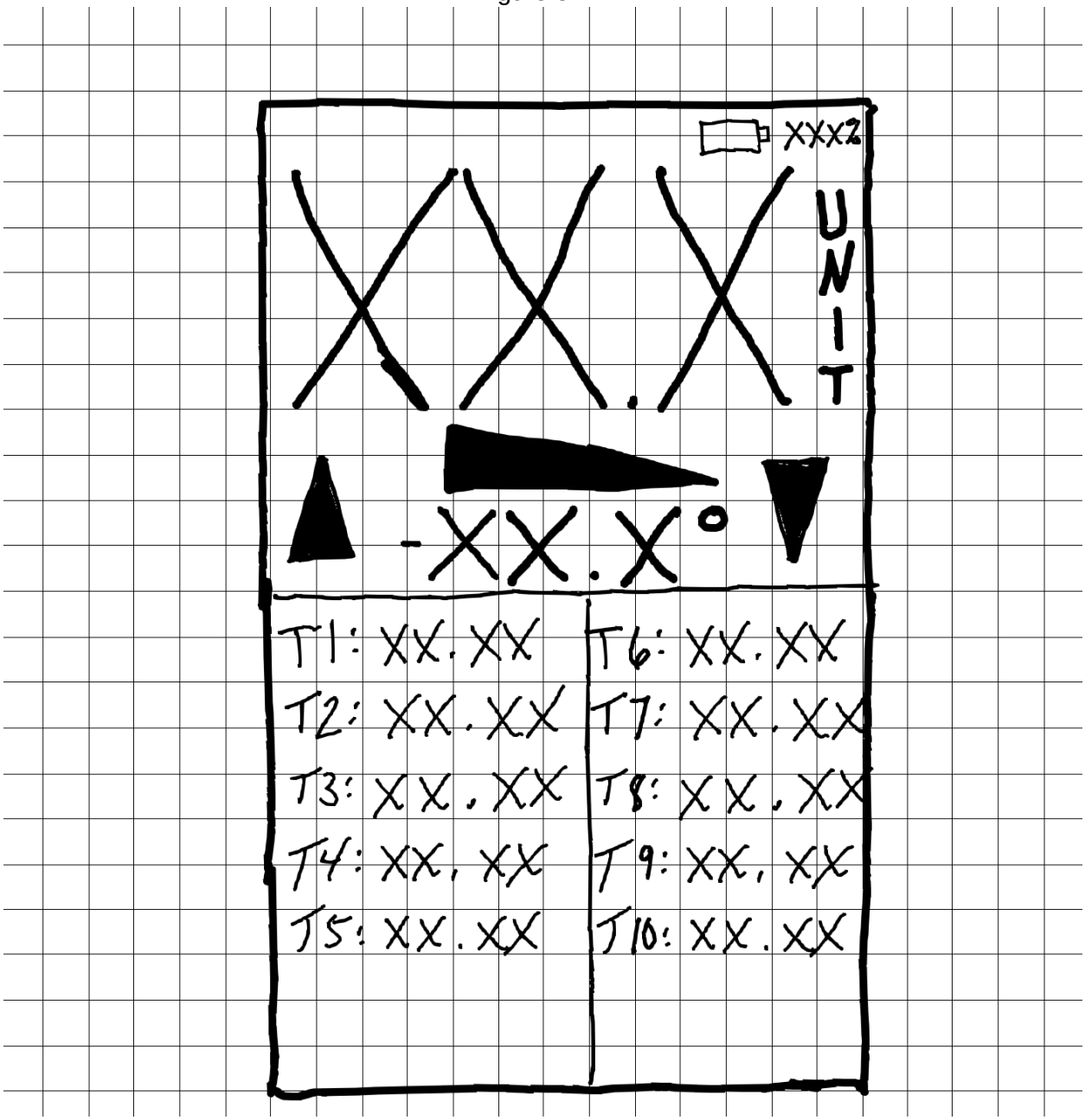
- Current value for number of CLICKS from zero (XX.X in Figure 2)
- Battery symbol or text saying “BAT” or “BATTERY” and the current percentage
- The current setting for the unit
- The live value for the angle from zero about the X axis (-XX.X° in Figure 2)
- Arrows indicating direction of rotation (clockwise or counter clockwise) about the Z axis

Figure 2

Variants:

1. I would like a second variant of this program that would add an area on the display where text data could be pushed from a mobile app. This variant would use the larger 2.42” display. Figure 3 shows approximate layout of display.

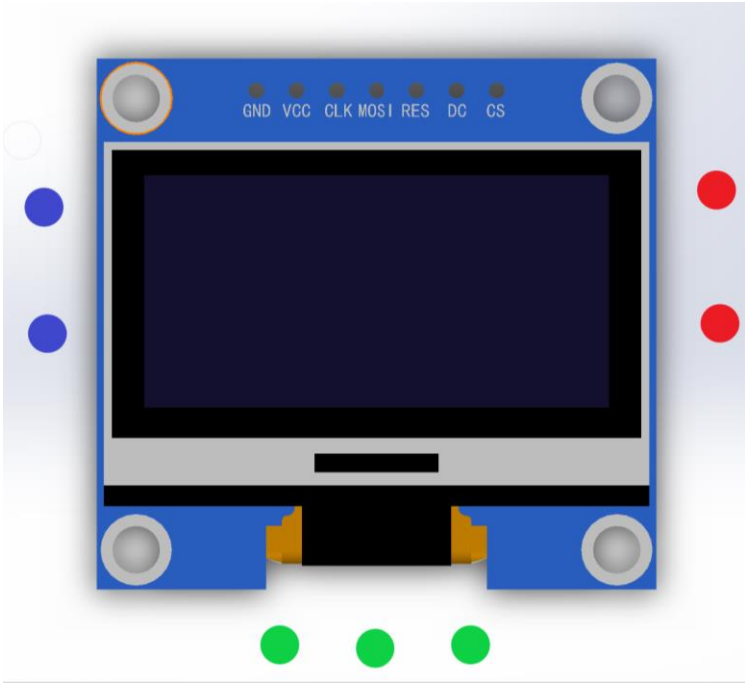
Figure 3



Revision 2:

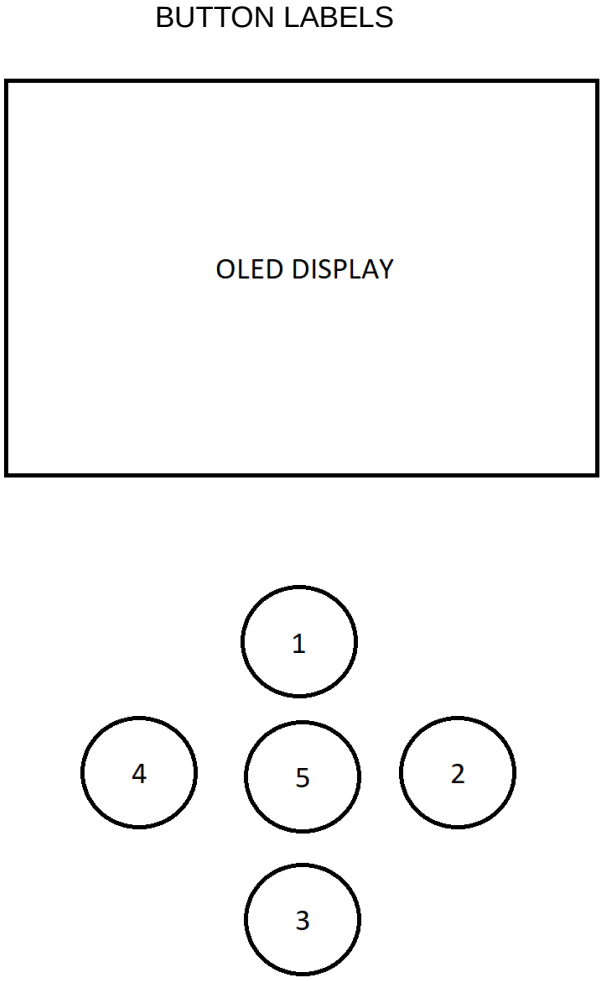
PCB Changes:

1. Rearrange buttons to match or come close to locations on suppled STEP file.
    a. 5 buttons in a directional pad format with center menu button (Buttons 1-5)
2. Add sealed toggle switch to power unit on and off (something similar to this https://www.e-switch.com/product-catalog/500r-series-sealed-slide-switch, does not have to be that switch)
3. Move USB C to location in STEP file
4. Move OLED header socket to location in STEP file
5. Add mounting holes for OLED, so standoff can be used to attach OLED to main board
6. Change header socket connections for encoder and level sensor to a waterproof locking connectors, connected to the PCB by approximately 1” of wire
    a. Encoder and level sensor should be separate connectors
    b. Connectors should be as small as possible, ideally an M5 connector or similar
    c. Connectors can be panel mount or cable mount
7. Move/add LEDs to following configuration
    a. 2 blue LEDs on left of OLED
    b. 2 red LEDs on right of OLED
    c. 3 green LEDs below OLED
    d. LEDs should be at same height as OLED panel so they can be mounted to the enclosure and be visible when looking at the OLED
        i. Having the LEDs on wire leads is also acceptable

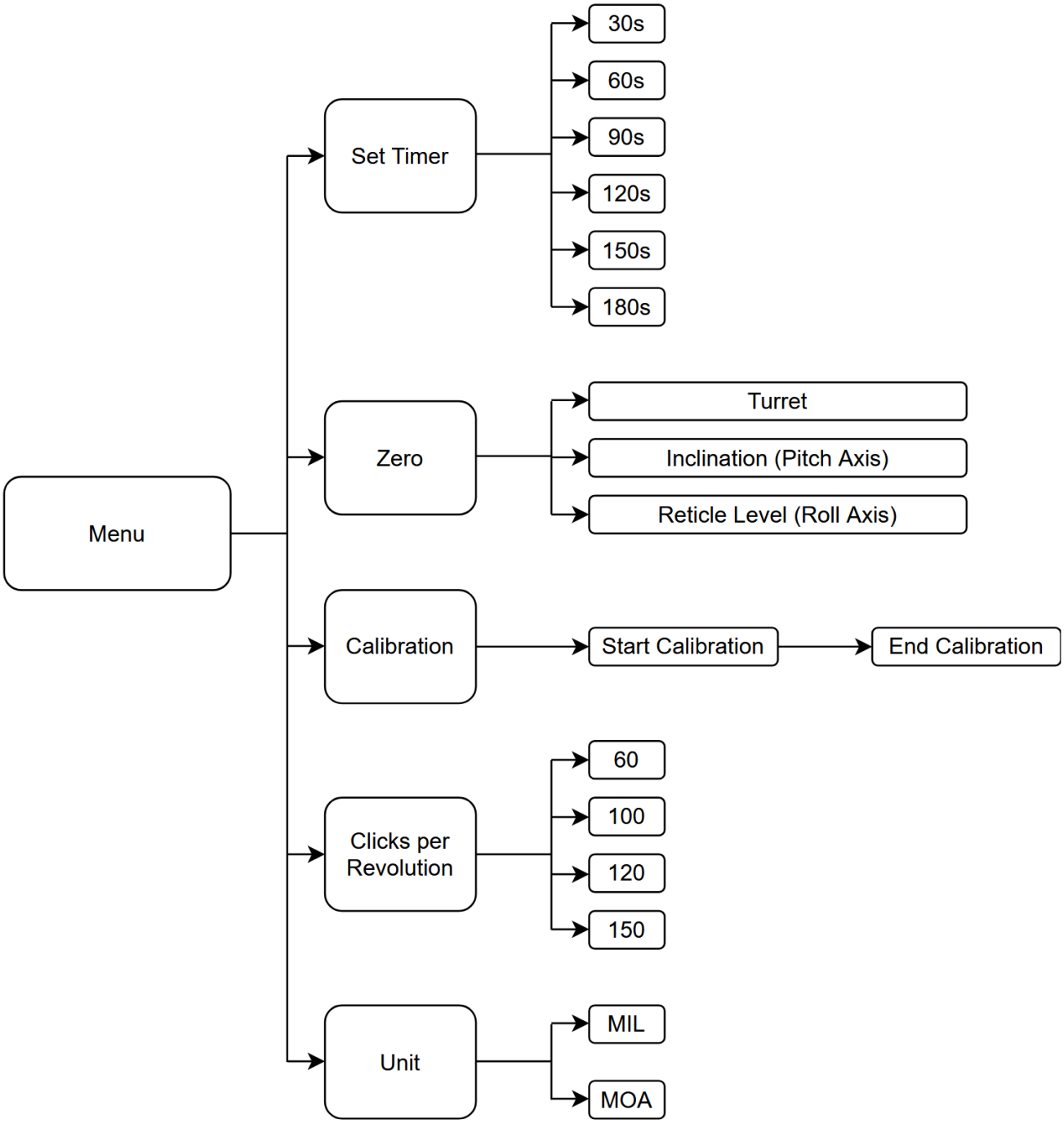


Program Changes:

1. Program boots directly to main information display screen



2. Add timer to main information display screen
    a. Timer value set in menu and counts down to zero by seconds
    b. Timer start/stop done by pressing **button 4** when outside of menus
    c. Timer reset done by pressing and holding **button 1** for 2s
    d. List of timer values selectable in menu should be easily added to or modified
3. Pressing and holding **button 3** for 2s zeroes encoder value
4. Create menus that can be accessed by a pressing and holding **button 5** for 3s.
    a. Menus formatted in following layout (use **buttons 1 (up), 2 (forward), 3 (down), 4 (back)** to navigate menus)



5. Modify program to work with 1.3” OLED panel (https://www.aliexpress.us/item/3256805781451407.html?spm=a2g0o.productlist.main.1.3c36bGtRbGtRaQ&algo_pvid=eedb3e83-35ce-4eee-a89c-6967932dbc3d&algo_exp_id=eedb3e83-35ce-4eee-a89c-6967932dbc3d-0&pdp_npi=4%40dis%21USD%213.42%210.99%21%21%2124.66%21%21%402103223417002473637736912e13aa%211200003509488769 3%21sea%21US%210%21AB&curPageLogUid=giltgvqnstSD)
6. Program should output to variables that are displayed using supplied program

    a. If there is a better way to do this please do so, but I would like the output displayed on the main screen as done by the supplied program
7. LEDs light up when level sensor detects the roll axis to be off level
    a. Red and Blue LEDS should activate with the arrows on the display
    b. One red or blue LED (bottom) should be activated if there is between 2 and 5 degrees off level, red if positive value off level, blue if negative value off level
    c. Both LEDs should be activated if there is more than 5 degrees off level red if positive value off level, blue if negative value off level
    d. All green LEDs should activate if roll axis is less than 2 degrees off level
    e. Values that trigger LEDs and arrows should be easily adjustable if needed.

Deliverables:

1. Program
2. CAD files for PCB
3. Bill of Materials
4. Prototype board if possible
5. Quote for production of these boards in quantities of 10, 50, 100, 500, and 1000