Content:

Setup: Matlab, Git, Visual Code, Clone Repo

Intro to Matlab and Simulink:

- Linear Algebra examples
- Laplace Transform Example

Calibrate the IMU

Introduction:

In this lab, you will set up our individual programming environment and get familiar with MATLAB. Finally, your group will be assigned a robot kit, and you will calibrate your robot’s IMU.

SETUP:

In order to program the robot, you must download a few software programs.

- Matlab R2018b or later with **Instrument Control Toolbox**
- Visual Studio Code with **PlatformIO IDE Extension**
- Git (with Bash) (instructions)

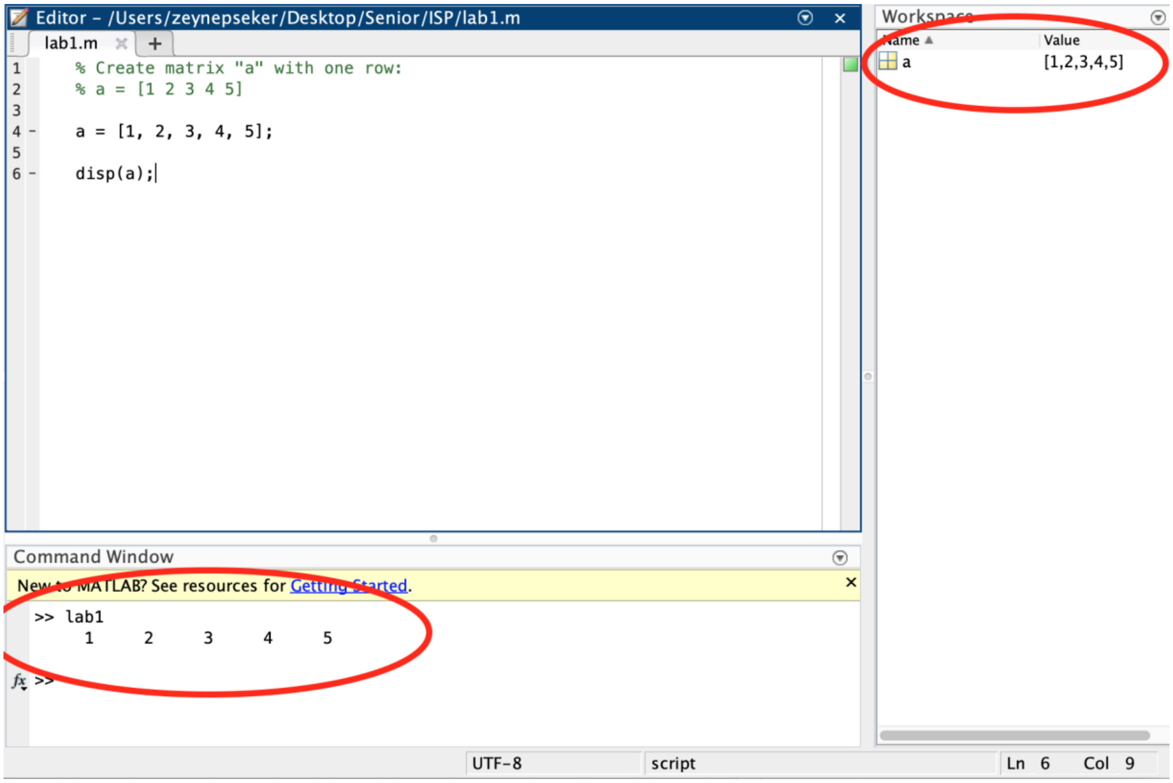
MATLAB (WPI Courses):

1) Create matrix "a" with one row and print it in the console:

   a = [1 2 3 4 5]

Paste your code here:

```
% Create matrix "a" with one row:
% a = [1 2 3 4 5]

a = [1, 2, 3, 4, 5];
disp(a);
```


Editor – /Users/zeynepseker/Desktop/Senior/ISP/lab1.m
lab1.m
Workspace
Name
Value
a
[1,2,3,4,5]
Command Window
New to MATLAB? See resources for Getting Started.
UTF-8
script
Ln 6
Col 9

```
% Create matrix "a" with one row:
% a = [1 2 3 4 5]

a = [1, 2, 3, 4, 5];

disp(a);
```

```
>> lab1
     1     2     3     4     5

>>
```

***

MATLAB Tip: If you type *clc* in the console, it will clear the previous prints.

2) Create matrix "b" with one row and two columns, and print it in the console:

b = [1 2 3 4 5]
[6 7 8 9 10]

Paste your code here:

```
% Create matrix "a" with one row:
% a = [1 2 3 4 5]

b = [1, 2,3, 4, 5; 6, 7, 8, 9, 10];
disp(b);
```

Connecting to the Robot:

Software configuration of the robot involves programming the Bluetooth module, calibrating the IMU, and testing the teleoperation code.

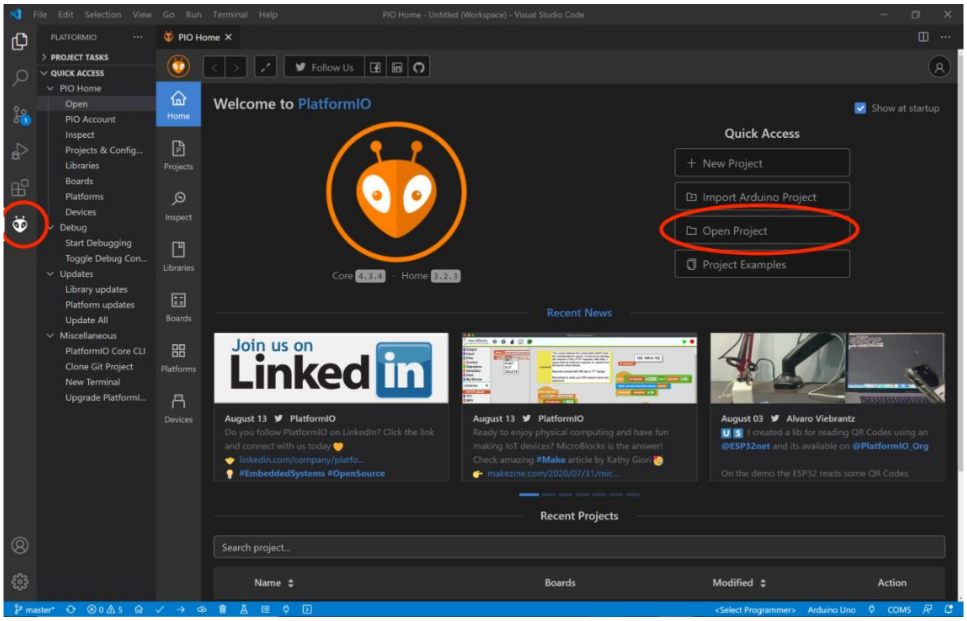
This requires first cloning the github repository of the balancing robot software. Navigate to the directory where you want the code copied (this can be done through the command prompt with the command **cd)**, run Git Bash or enter the following command:

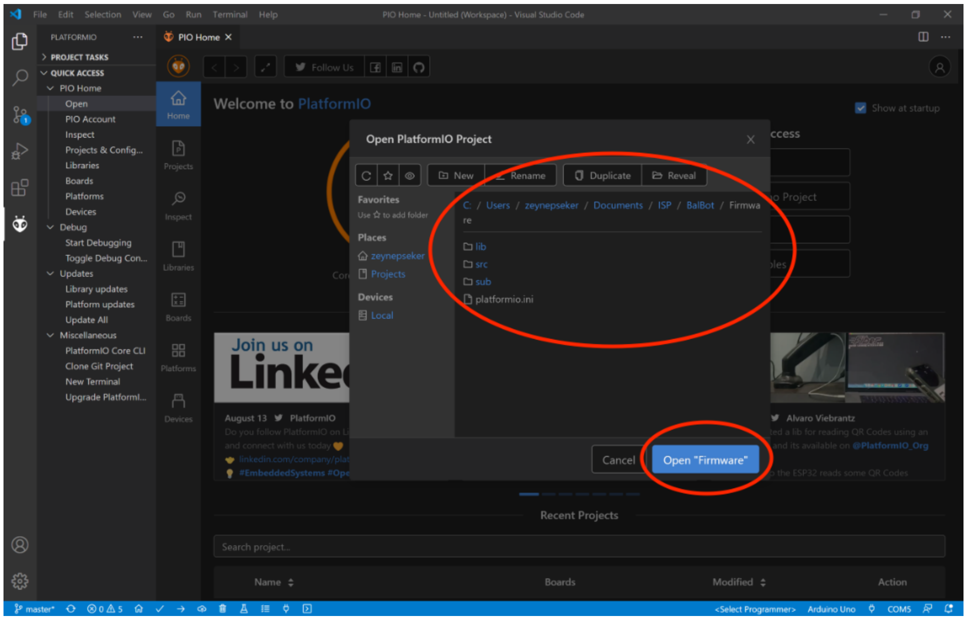
git clone --recursive https://github.com/doates625/BalBot.git

This should create a directory called "BalBot" in your current directory with all of the required software.

Calibrating the IMU:

The IMU on every robot is slightly different, so each robot needs to be individually calibrated before the first time you use it and again if needed. Begin by running *Visual Studio Code.* In the *PlatformIO IDE Home* click the Home icon in the blue bar at the bottom of the window. From there, click "Open Project", navigate to <BalBot/Firmware> through your directory and click "Open". Visuals for this process can be seen below.

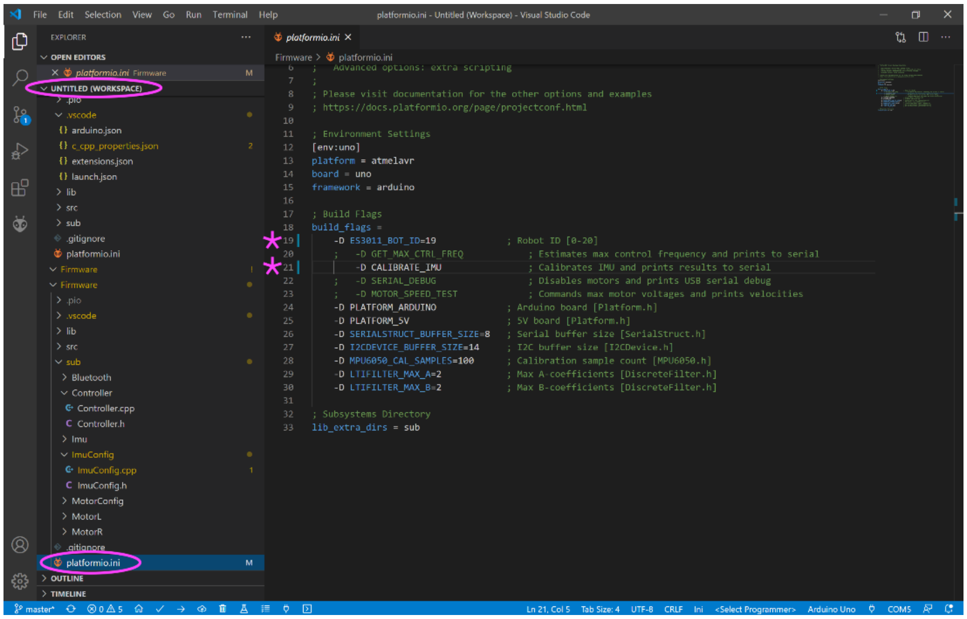




C: / Users / zeynepseker / Documents / ISP / BalBot / Firmware

Next, open **<platformio.ini>**. In the ‘build_flags’ section, set “ES_3011_BOT_ID=0” on line 19 to your robot number (do not leave a space between the number and the equal sign), and uncomment “-D CALIBRATE_IMU” on line 21.

Tip: While this line is uncommented, the robot will not balance itself. Make sure you comment this line after you are done with calibration.


platformio.ini - Untitled (Workspace) - Visual Studio Code
EXPLORER
OPEN EDITORS
platformio.ini Firmware
UNTITLED (WORKSPACE)
Firmware > platformio.ini

```
 6  ;   Advanced options: extra scripting
 7  ;
 8  ; Please visit documentation for the other options and examples
 9  ; https://docs.platformio.org/page/projectconf.html
10
11  ; Environment Settings
12  [env:uno]
13  platform = atmelavr
14  board = uno
15  framework = arduino
16
17  ; Build Flags
18  build_flags =
19      -D ES3011_BOT_ID=19             ; Robot ID [0-20]
20      ;   -D GET_MAX_CTRL_FREQ            ; Estimates max control frequency and prints to serial
21          -D CALIBRATE_IMU                ; Calibrates IMU and prints results to serial
22      ;   -D SERIAL_DEBUG                 ; Disables motors and prints USB serial debug
23      ;   -D MOTOR_SPEED_TEST             ; Commands max motor voltages and prints velocities
24      -D PLATFORM_ARDUINO             ; Arduino board [Platform.h]
25      -D PLATFORM_5V                  ; 5V board [Platform.h]
26      -D SERIALSTRUCT_BUFFER_SIZE=8   ; Serial buffer size [SerialStruct.h]
27      -D I2CDEVICE_BUFFER_SIZE=14     ; I2C buffer size [I2CDevice.h]
28      -D MPU6050_CAL_SAMPLES=100      ; Calibration sample count [MPU6050.h]
29      -D LTIFILTER_MAX_A=2            ; Max A-coefficients [DiscreteFilter.h]
30      -D LTIFILTER_MAX_B=2            ; Max B-coefficients [DiscreteFilter.h]
31
32  ; Subsystems Directory
33  lib_extra_dirs = sub
```

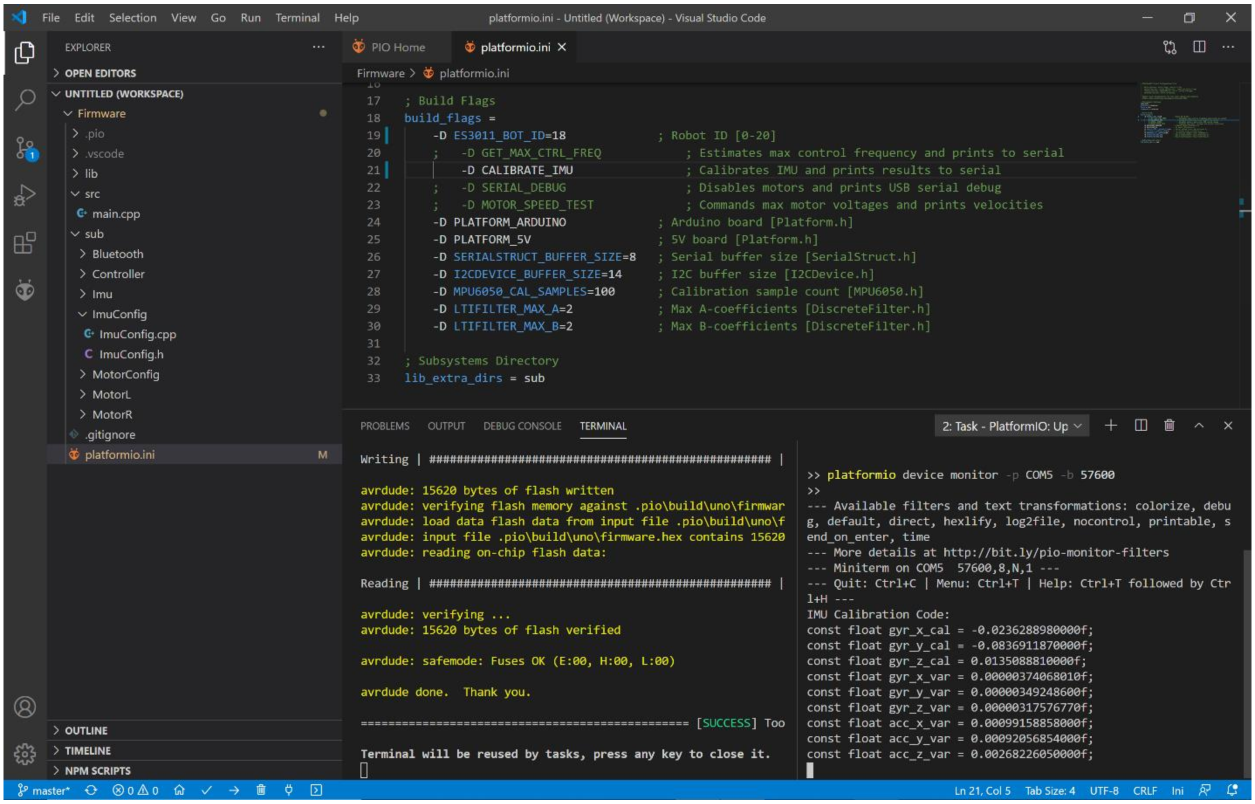
Connect the robot to the computer with the USB cable, verifying that **both** the Power and Bluetooth switches are **off**. The calibration will not work if any of these are turned on! Next, open the Windows Device Manager and find the COM port of the Arduino, which should be named “USB Serial Device” under “Ports (COM & LPT)”, and have the form “COMXX”.

Build and upload the code to the robot by clicking the arrow icon in the blue bar or typing <CTRL+ALT+U>. A terminal window should appear and end with a success message.
At this point, keep the robot still. Look at the Arduino from the side and check that the user LED is off. If the LED is on, this means that the Arduino was unable to communicate with the IMU and the IMU is likely broken and in need of replacing. Open a new terminal by typing <CTRL+SHIFT+5>. In the new terminal, enter the command:

```
platformio device monitor -p COMXX -b 57600
```

Make sure you hold your robot upright to get the closest calibration values.
Where COMXX is the COM port of the Arduino. After a few seconds, the terminal should display something like this:

```
const float gyr_x_cal = -0.0719998850000f;
const float gyr_y_cal = +0.0160055890000f;
const float gyr_z_cal = +0.1958248800000f;
const float gyr_x_var = 0.00000883372190f;
const float gyr_y_var = 0.00000227666920f;
const float gyr_z_var = 0.00000215141040f;
const float acc_x_var = 0.00101798410000f;
const float acc_y_var = 0.00107184890000f;
const float acc_z_var = 0.00253210360000f;
```



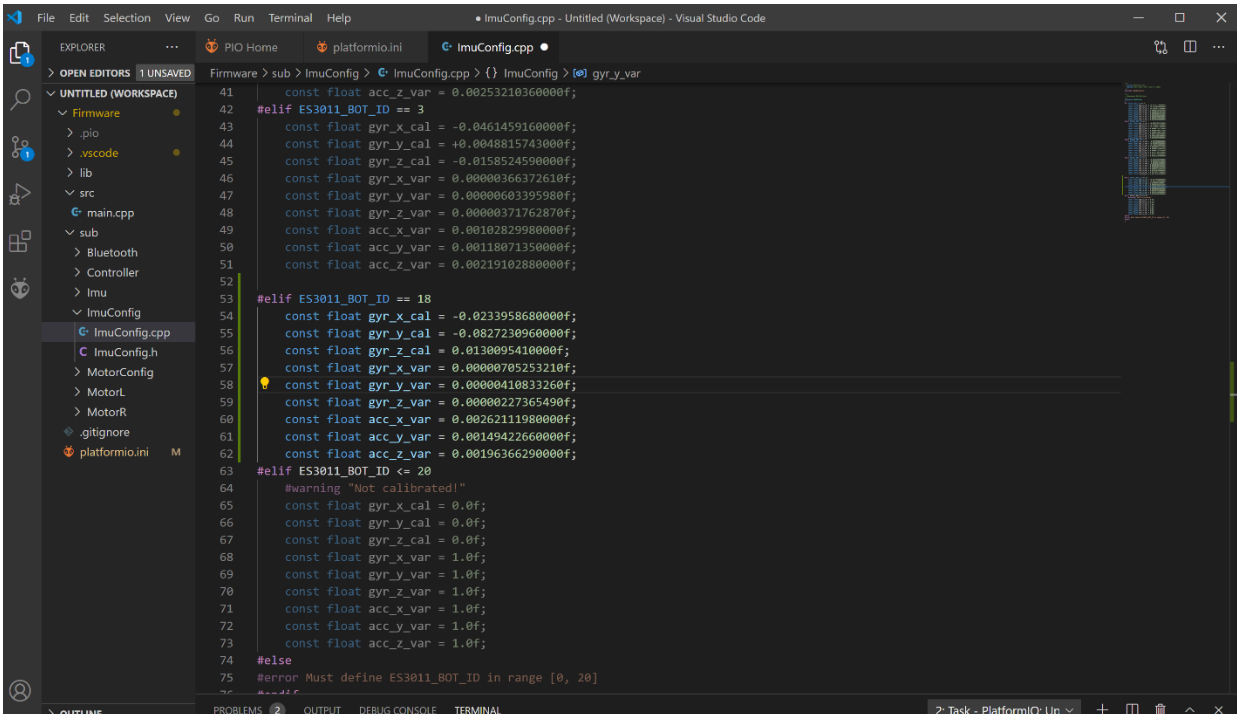
```
; Build Flags
build_flags =
    -D ES3011_BOT_ID=18             ; Robot ID [0-20]
    ;   -D GET_MAX_CTRL_FREQ            ; Estimates max control frequency and prints to serial
        -D CALIBRATE_IMU                ; Calibrates IMU and prints results to serial
    ;   -D SERIAL_DEBUG                 ; Disables motors and prints USB serial debug
    ;   -D MOTOR_SPEED_TEST             ; Commands max motor voltages and prints velocities
    -D PLATFORM_ARDUINO             ; Arduino board [Platform.h]
    -D PLATFORM_5V                  ; 5V board [Platform.h]
    -D SERIALSTRUCT_BUFFER_SIZE=8   ; Serial buffer size [SerialStruct.h]
    -D I2CDEVICE_BUFFER_SIZE=14     ; I2C buffer size [I2CDevice.h]
    -D MPU6050_CAL_SAMPLES=100      ; Calibration sample count [MPU6050.h]
    -D LTIFILTER_MAX_A=2            ; Max A-coefficients [DiscreteFilter.h]
    -D LTIFILTER_MAX_B=2            ; Max B-coefficients [DiscreteFilter.h]

; Subsystems Directory
lib_extra_dirs = sub
```

```
Writing | ################################################## |

avrdude: 15620 bytes of flash written
avrdude: verifying flash memory against .pio\build\uno\firmwar
avrdude: load data flash data from input file .pio\build\uno\f
avrdude: input file .pio\build\uno\firmware.hex contains 15620
avrdude: reading on-chip flash data:

Reading | ################################################## |

avrdude: verifying ...
avrdude: 15620 bytes of flash verified

avrdude: safemode: Fuses OK (E:00, H:00, L:00)

avrdude done.  Thank you.

================================================ [SUCCESS] Too

Terminal will be reused by tasks, press any key to close it.
```

```
>> platformio device monitor -p COM5 -b 57600
>>
--- Available filters and text transformations: colorize, debu
g, default, direct, hexlify, log2file, nocontrol, printable, s
end_on_enter, time
--- More details at http://bit.ly/pio-monitor-filters
--- Miniterm on COM5  57600,8,N,1 ---
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctr
l+H ---
IMU Calibration Code:
const float gyr_x_cal = -0.0236288980000f;
const float gyr_y_cal = -0.0836911870000f;
const float gyr_z_cal = 0.0135088810000f;
const float gyr_x_var = 0.00000374068010f;
const float gyr_y_var = 0.00000349248600f;
const float gyr_z_var = 0.00000317576770f;
const float acc_x_var = 0.00099158858000f;
const float acc_y_var = 0.00092056854000f;
const float acc_z_var = 0.00268226050000f;
```

Tip: If your screen does not display this message, your IMU might be broken.

Open <sub/ImuConfig/ImuConfig.cpp>. Make a new “#elif” section for the robot ID and paste in the nine lines of code above as printed by the serial terminal.


• ImuConfig.cpp - Untitled (Workspace) - Visual Studio Code
PIO Home
platformio.ini
ImuConfig.cpp
Firmware > sub > ImuConfig > ImuConfig.cpp > {} ImuConfig > gyr_y_var
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
2: Task - PlatformIO: Up

```
    const float acc_z_var = 0.00253210360000f;
#elif ES3011_BOT_ID == 3
    const float gyr_x_cal = -0.0461459160000f;
    const float gyr_y_cal = +0.0048815743000f;
    const float gyr_z_cal = -0.0158524590000f;
    const float gyr_x_var = 0.00000366372610f;
    const float gyr_y_var = 0.00000603395980f;
    const float gyr_z_var = 0.00000371762870f;
    const float acc_x_var = 0.00102829980000f;
    const float acc_y_var = 0.00118071350000f;
    const float acc_z_var = 0.00219102880000f;

#elif ES3011_BOT_ID == 18
    const float gyr_x_cal = -0.0233958680000f;
    const float gyr_y_cal = -0.0827230960000f;
    const float gyr_z_cal = 0.0130095410000f;
    const float gyr_x_var = 0.00000705253210f;
    const float gyr_y_var = 0.00000410833260f;
    const float gyr_z_var = 0.00000227365490f;
    const float acc_x_var = 0.00262111980000f;
    const float acc_y_var = 0.00149422660000f;
    const float acc_z_var = 0.00196366290000f;
#elif ES3011_BOT_ID <= 20
    #warning "Not calibrated!"
    const float gyr_x_cal = 0.0f;
    const float gyr_y_cal = 0.0f;
    const float gyr_z_cal = 0.0f;
    const float gyr_x_var = 1.0f;
    const float gyr_y_var = 1.0f;
    const float gyr_z_var = 1.0f;
    const float acc_x_var = 1.0f;
    const float acc_y_var = 1.0f;
    const float acc_z_var = 1.0f;
#else
#error Must define ES3011_BOT_ID in range [0, 20]
```

Next, open <platformio.ini> and make sure the build flag ES3011_BOT_ID to the robot ID (make sure to leave no spaces around the equal sign).

Finally, re-comment the CALIBRATE_IMU build flag and re-upload the code.

Manually hold the robot upright on a flat surface and flip the power switch on. Wait one second, then press the small red reset button on the Arduino, making sure not to disconnect the Arduino from the shield board.