Ron's Broomstick

This code is set to work on an Arduino Nano connected to an IMU using I2C communication. Thus, we included the “Wire.h” library at the beginning of the code.

A – Setup

We set the built-in LED for the Arduino to OUTPUT mode.

Then we set the IMU register for the accelerator and the gyroscope.

Then, we use the **calibrateImu(int);** function so that the IMU doesn’t read garbage values.

(In this code, we read gyroscope input although we don’t need to).

B – Functions

Interfacing the IMU

- **setImuRegsiter(reg, val)**

This function is needed to set each register in the IMU.

- **readImu2Bytes(reg)**

This function reads 2 bytes from the IMU register. To read two bytes but get one value, we shift the reading of the first byte 8 places then use a bitwise OR operator with the second unshifted byte to get the final number.

- **calibrateImu(n)**

To prevent reading garbage values, we calibrate the IMU, meaning we take a large number of readings while the IMU is stationary. Then, get the average of those readings to find the offset of the measurement.

This is done by a for loop continuously getting readings and adding them to the offset values, then dividing the final values by the number of iterations.

If we want an accurate measurement, we should subtract the offset found from all measured values afterwards.

Reading IMU sensors

- readImuAll()

This function reads 14 bytes from the IMU register, thus getting all values of the acceleration in the direction of all 3 axes as well as gyroscope measurements (unneeded in this task).

C - Loop

Each loop starts by reading all input from IMU.

Then, we calculate true values of the acceleration on three axes (by removing offset, multiplying by the full-scale range, and then dividing by the maximum size of the 2-byte integer).

Afterwards, we calculate both roll and pitch using the following function:

```
pitch = 180 * atan2(accelX, sqrt(accelY*accelY + accelZ*accelZ))/PI;
roll = 180 * atan2(accelY, sqrt(accelX*accelX + accelZ*accelZ))/PI;
```

Then, we check if any of them exceeds 60 degrees. If that condition is true, the built-in led blinks.