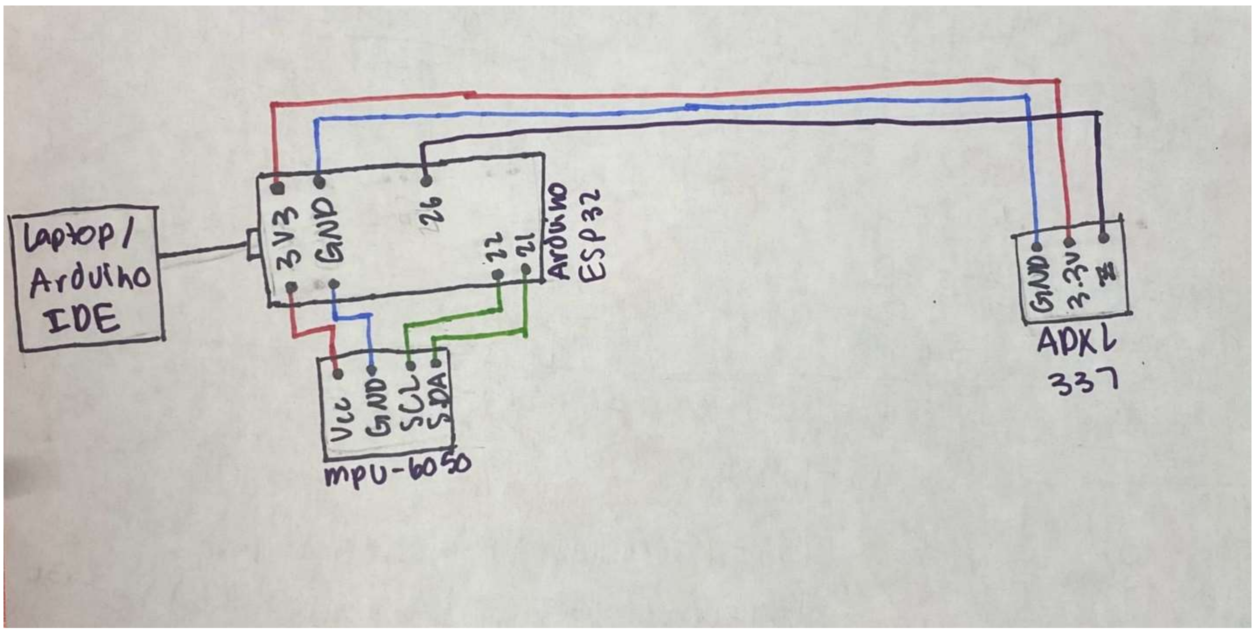
Measuring Involuntary Breathing Movements with Prolonged Breath Holding

Lillian Tucker

EECE 244

February 2023

This paper is a report of an independently constructed device to measure IBM (involuntary breathing movements) with breath holding for extended periods of time. Provided electronics included a programmable circuit board and various electrical components as well as materials to manipulate these components. The goal was to create a more accurate reading for IBM.



(Figure 1: Diagram of ESP32, ADXL337 and MPU6050)

<u>Digital Accelerometer (ADXL 337)</u>

The component used to measure the IBM is an ADXL337 digital accelerometer. This sensor is attached to the corresponding GND pin, the 3V3 pin of the ESP32 for power. For the input from the accelerometer to the Arduino, it is connected to pin 26 which is a general GPIO pin.

It is placed flat on the middle of the sternum to measure the IBM using the Z axis of the accelerometer. When watching the video of Jordan performing breath holding to demonstrate the IBM, I noticed that with any IBM, there was consistently movement in the sternum area. Another reason for this is the sternum area had minimal movement when Jordan was adjusting his body for comfort.

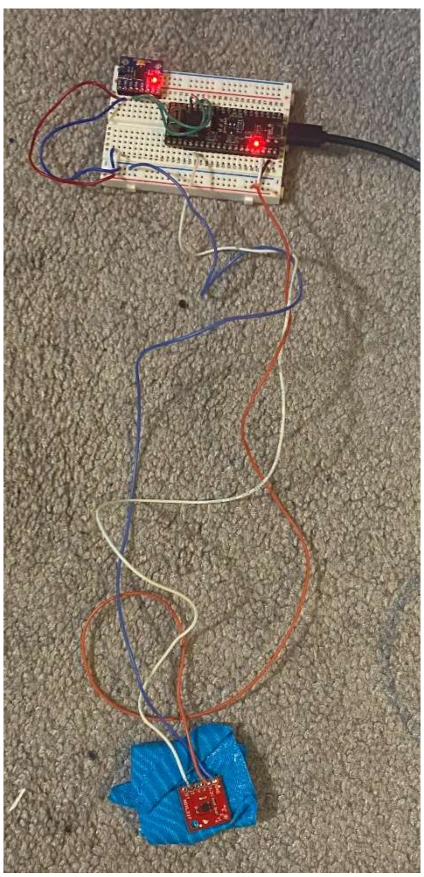
The raw input of the accelerometer is the voltage. Using an example code provided by Dr. M, the function mapf() can be used to take this input and process it to output the acceleration of the accelerometer. This can then be printed to the serial monitor or serial plotter using Serial.println(). I scaled the acceleration data by a multiple of 10 to be able to clearly see changes in readings.

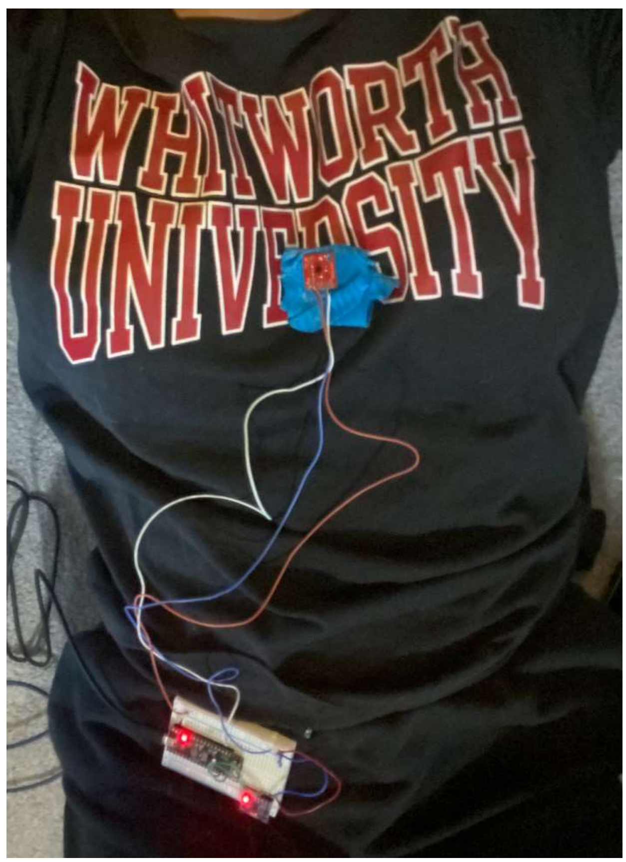
<u>I2C Accelerometer (MPU 6050)</u>

The component used to filter out movements other than IBM is an MPU 6050 I2C accelerometer. It is attached to the blank circuit board that the ESP32 Arduino is wired on. This sensor is attached to the corresponding GND and 3V3 pins of the ESP32 for power. Since this sensor is I2C, an SCL and SDA pin are required to operate. On the ESP32, pin 22 is used for SCL and pin 21 is used for SDA so the sensor is connected accordingly.

This circuit board is placed flat on the hips round the hip bone area. When watching the IBM demonstration of Jordan, there was consistent movement in the hip area when he adjusted his body for comfort, and minimal movement during IBM. Like the ADXL337, the raw input from the MPU 6050 is voltage readings. Using a sample code downloaded from the MPU 6050 library, the voltage readings can be processed to acceleration. One unique component of the MPU 6050, is that in addition to being able to measure acceleration in the XYZ plane, it can also measure acceleration gyroscopically. I used this to measure hip movements in filtering out data. In the Arduino code, I used an if statement to determine when to plot the data from the digital accelerometer. If the gyroscopic data from the I2C accelerometer read less than 0.7, it would plot the data read by the ADXL337. Otherwise, no data would plot and there would be a 500-millisecond delay to allow time between the start of bodily adjustments and the start of measuring steady chest movements.

The calibration of the if-else statement was find using trial and error. The gyroscopic threshold was originally set to < 0 but I found that it was too sensitive and would exclude more drastic chest movements. The delay in the else statement was originally set to 100-milliseconds. I found this delay to be too short as data would resume plotting while bodily adjustments were still active. I increased this delay by 100-milliseconds until I was satisfied by the delay.





(Figure 2&3: Real setup (L), Real setup on person(R))

Results

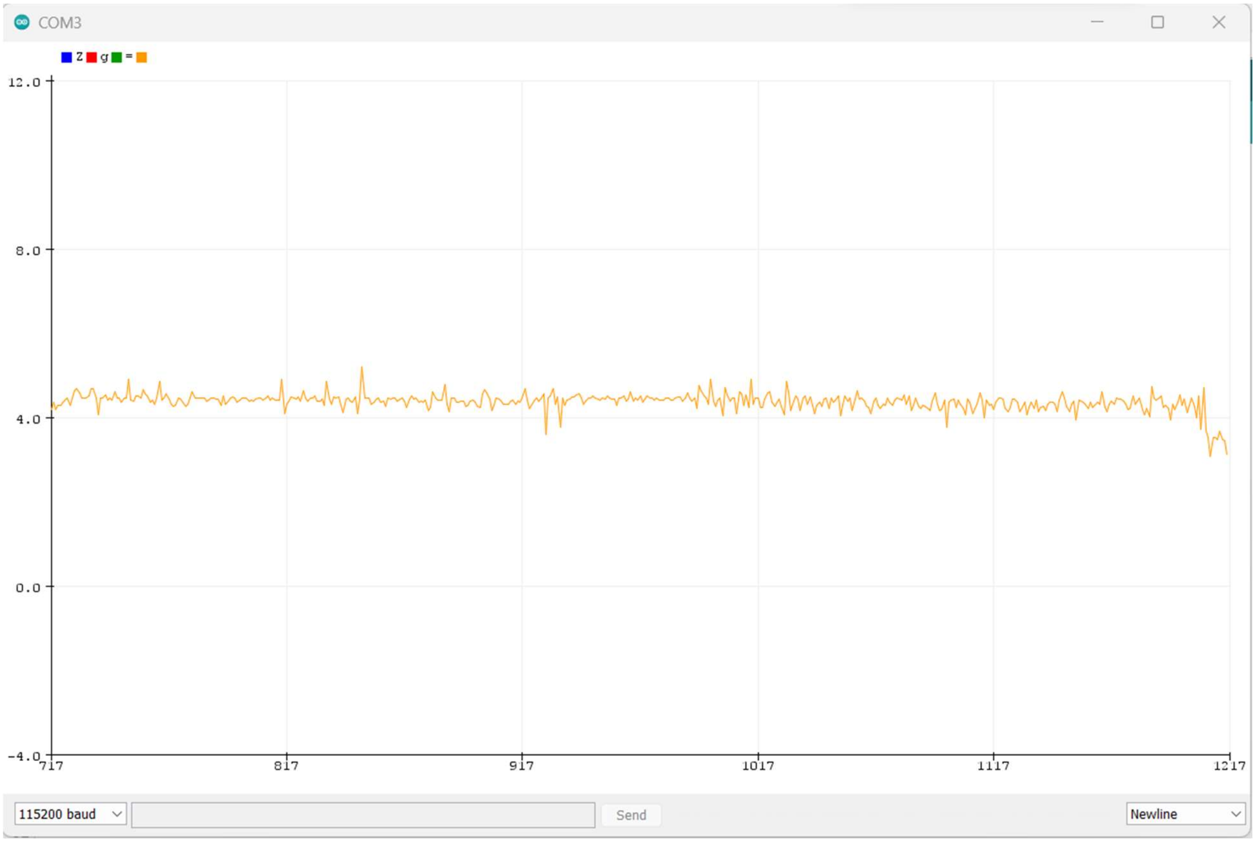


Figure 4: Serial Plot of Data from Arduino (Video includes trial for this data plot)

The numbers seen along the y axis of the plot was the acceleration of the digital accelerometer to a scale of 10, and the x axis is a fixed 500-point plot. This means the numbers on the bottom represent the number of cycles have been executed.

Although not shown on the plot, the plot did stop collecting data when there was noticeable movement measured on the I2C accelerometer. Although there were still some data differences while the chest movements were negligible, the difference in data shown while I performed chest movements and manual breathing was still notable. There were noticeable differences between breathing in(cycle 717-800), steadily holding my breath(cycle 800-815), performing chest movements while holding my breath(cycle 815-920), performing shallow breaths(cycle 1000-1117) and deeper breaths(cycle 1117-1200).

One major difference shown on the plot with the accelerometer is that the data grouping stays consistent rather than gradually increasing. This is because the instantaneous acceleration is independent to each data point rather than pressure plate sensors which can consistently gain pressure as the chest consistently expands as IBM increases.

One adjustment I would like to make in the future is filtering for external electrical components. In a lab for EECE 226, we were asked to see voltage and frequency interference various objects and electronics can produce. One device I decided to observe is my personal insulin which sits on the side of my hip around the hip bone area. When measuring voltage interference of my pumped, I measured 60mVpp interference while the pump was on and touching the oscilloscope. While the oscilloscope was approximately 2 inches away from the insulin pump there was a measured 20 mVpp while the pump was on. In the future I want to keep in mind the various medical devices individuals may have on them day to day.