#### Dev Shah

#### WEEK 15:

**Date:**4/22/22  
**Total hours:** 13  
**Description of design efforts:**  
This week my work consisted of completing all the remaining software and combining them all into one coherent program. I started this by finishing the free fall detection algorithm for the accelerometer. The accelerometer is now able to detect free falls when experiencing less than 156 mg for roughly 75 ms. We determined that if a worker were to be hit by something noticable that would require attention, their head would move the same way as though they were in free fall. Because of this, I stopped working on the tap detection after implementing the fall detection. Image 15.1 shows what is outputted to the terminal when the hard hat detects a free fall. In Video 15.1, the hard hat shines an LED when it detects a free fall.

Text

Description automatically generated

Image 15.1

<https://engineering.purdue.edu/477grp19/Team/progress/img/shah15.1.mp4>

Video 15.1

After getting the free fall detection to work, I began combining all the separate tasks we had code for. The first segment of code I combined were the accelerometer code and the LoRa code and set it so the worker's LoRa would send a signal to the supervisor's LoRa when a free fall was noticed. Once that was working, I moved on to programming the emergency button. This button is configured to so that the supervisor system displays "Emergency" when pressed.  
  
After the completion of this task, I went back to working on obtaining an accurate BPM from the pulse sensor. To test the code that was created, I had the software print "Pulse" to the terminal everytime it read a pulse value. After different attempts, we determined that a pulse value would be counted if the software noticed peaks in the data. An issue this software faced was that it would sometimes read a pulse more times than there actually was one. To combat this, I, along with the help of Ryan, changed the code so that a pulse value would be detected if the voltage from the sensor was 400 mV greater than the wearer's average pulse level. The average pulse level is updated every 50 ms and a BPM value is calculated after every 15 seconds. This code is now set to beep the buzzer everytime it detects the wearer’s pulse for the first minute of use. This allows the wearer to determine if the pulse is being read correctly. After the first minute, the buzzer stops beeping and the software is able to calculate an accurate BPM. This was tested by comparing the BPM values of the sensor to the BPM values read by an Apple Watch. If a BPM higher than 180 or lower than 40 is calculated, the hard hat sends a signal to the supervisor.  
  
The final sensor that was implemented was the gas sensor. Because the code was already created, we simply had to test it to make sure the readings were accurate and a signal would be sent to the supervisor when appropriate. In order to do so, we went to a parking garage and placed the hard hat close to the exhaust of a team member's car. When a significant amount of carbon monoxide was detected, resulting in a reading greater than 1000 mV, a signal was sent to the supervisor.  
  
Once this code was correct, the final software for the hard hat was completed. The hard hat is set so that for the first minute of being powered on, the pulse sensor would beep when it detects the wearer's pulse. After this minute, the hard hat will send a signal to the supervisor depending on what event occurred. Every 15 seconds, a BPM will be calculated and if it is over 180, the hard hat will send "High BPM". If the BPM is lower than 40, it will send "Low BPM". If a free fall is detected, "User Fell" will be sent, and if a significant amount of carbon monoxide is measured, "Gas Found" is sent. "Emergency" is sent to the supervisor if the wearer presses the button. The latest event that occurs gets delivered to the supervisor every 5 seconds. If the hard hat's battery is low, its buzzer beeps 5 times every 10 seconds.