<u>Rooster Band</u>
The Wristband That Keeps You Awake

By

Junfei Gu,
Bum “Jun” Kim
Jaime Ontiveros

Final Report for ECE 445, Senior Design, Fall 2019
TA: Enliang Li

11 December 2019
Project No. 11

Abstract - **For our project we decided to make a wristband that will help people stay awake. It can be used by students in class, people at work, and even drivers that feel fatigued. It will use a pulse sensor and an accelerometer to take in data from the user and send it to the microcontroller which will have an algorithm that will decide whether the user is falling asleep or not. The wristband will then activate a vibration motor that will wake the user and let them continue their activity. This report will summarize our design process and results for the product known as roosterband.**

Table of Contents:

1. Introduction

1.1 Objective:

Everyday life has become fast paced and with so much work to do and events, getting a good night's sleep can be hard to come by. Some people also have sleeping disorders which make them want to sleep at random times. People being drowsy can lead to sleeping in class, in meetings, behind the wheel, or on transportation. This will lead to them missing their stop, information, or even fatal accidents.

With the wristband we are creating, people will be kept awake within five minutes of entering the first stage of sleep. It will use an accelerometer and pulse sensor that will collaborate to accurately tell if the user has entered the first stage of sleep. Once the sensors send a positive single that the user is in stage one of sleep, then the wristband will use a small vibration motor to get the user back to being alert.

1.2 Background:

A large sleep census conducted by Sealy and the Loughborough University in 2016 showed that in the five countries where the census was conducted, an average of around 23% percent of people suffer from insomnia, and only 24% of people get the recommended 8+ hours of sleep. Clearly many people around the world have sleep problems and can’t get enough sleep. The study further shows that around 47% of people take naps during the day which disrupts their work and school day.[1] People are clearly having trouble staying awake and our device should be a simple solution that requires minimum effort for the user to take advantage of.

At first this project may just seem to be as simple as adding a vibrating function to one of the popular smartwatches or wearable sleep monitors but there are clear differences. The aforementioned devices’ main goal is to track a person's sleep, using an accelerometer to record the movements of the wearer and determine the wearer’s sleep status by the stillness of his/her body.[2] The problem with this is twofold. Firstly, the movement of a body during sleep will only completely stop after stage two of NREM sleep, meaning that by the time the user has been detected sleeping, he/she has already been asleep for 10-20 minutes and the damage will have already been done. The second problem is that it won't be able to detect people falling asleep in other settings besides laying down on a bed making it useless for drivers and the like.

1.3 Functionality

Pulse sensor is the main sleep detection system, it tracks the pulse drop the user has during the day, the accelerometer records movement at the time. The algorithm decides whether the user is sleep or not. Then the vibration motor runs to alert the user. The vibration will turn off with a very strong shake from the user.

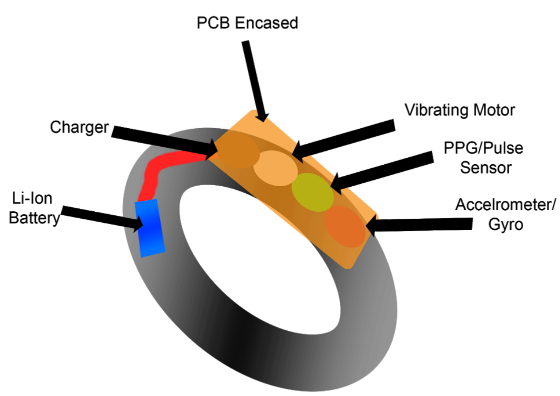
1.4 Visual Aid:

Fig 1.1: Visual of the device in use

1.5 High-Level Requirements List:

1. **Assure accurate sensor reading within 10% of error while user is active in sedentary environment.**
2. **Once sleep is detected, alert the subject to wake within 1 minute.**
3. **Minimum of 30 mins of continuous runtime(long enough to chase off sleep) while activating all waking peripherals. 6 hours of continuous silent runtime.**

1.6 Physical Design:



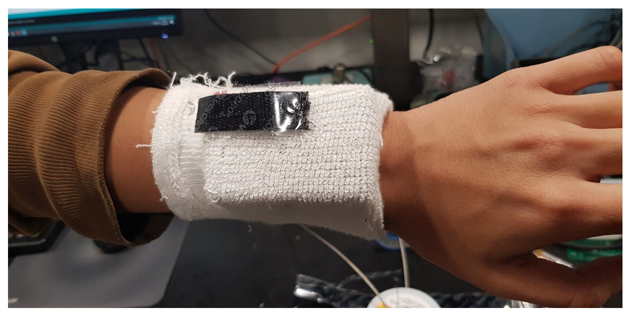


Fig 1.2(Left): Basic physical diagram of the wristband

Fig 1.3(Right): Final Product on use

Our ideal form for this wearable device is that of a wristband. Wristbands are commonly used wearables that are convenient and compact. They are casual enough to wear in any situation making it very accessible. Figure 1.2 is a physical design prototype is similar to a watch where instead of a watch face, we will have a solid box that will hold the pcb and most of the internals including the pulse sensor and accelerometer. This box is represented by the orange rectangle in the diagram and the two dots underneath are the pulse sensor and accelerometer. The pulse sensor will be placed directly above the wrist in a way to effectively read the user's pulse. The battery was placed in the box along with the PCB. The material for the band is a stretchable cloth used for tennis sports bands. Using a sports band was ideal because it could fit various wrist sizes while also being able to stay in place well enough for the accelerometer to function properly.

1.7 Block Diagram

There are a total of 4 subsystems that we implemented for this project. Each one of these subsystems will fulfill or help fulfill one or more of our high-level requirements. The sleep detecting subsystem is comprised of the two sensors we will be using to detect sleep within 3-5 minutes of stage one non-rem sleep. Every sensor will be able to send information to the control subsystem which will process everything through a programmable microcontroller. The accelerometer will use SPI(Serial Peripheral Interface) and the pulse sensor will use I2C(Inter-Integrated Circuit). The sensors will be continuously sending data through the data line into the microcontroller in the control subsystem at a rate of 32 Hz. When the microcontroller gets positive signals from the sleep detecting subsystem, it will send signals to the wake up subsystem which sends signals to the vibration motor to activate and wake up the user. The user can then press a button to stop the vibration and reset the device. The same button will also be power button to turn the device on and off. Everything will be powered by our power subsystem which will have a 3.7 V lithium ion battery that goes through a voltage regulator to maintain a steady 3.3 V voltage that will power all of our individual parts. This battery will have a charger allowing the device the be rechargeable.

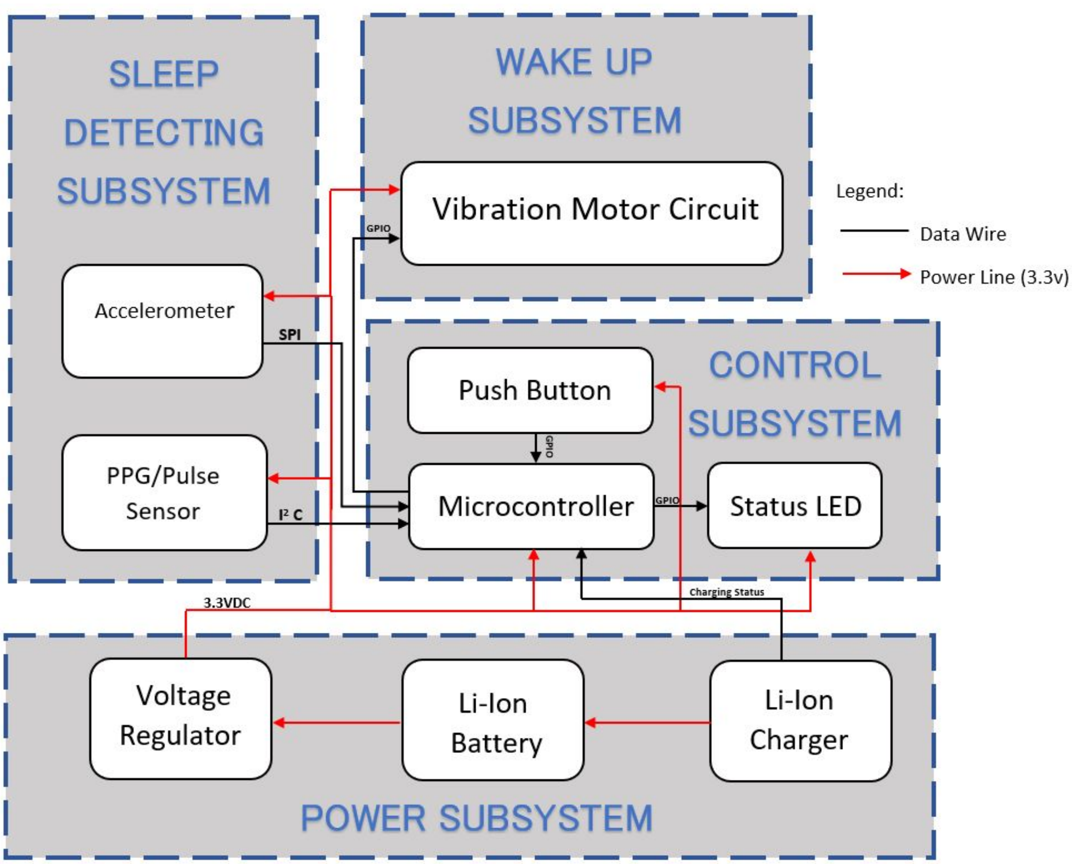


Fig 1.4: Block Diagram with 4 subsystems

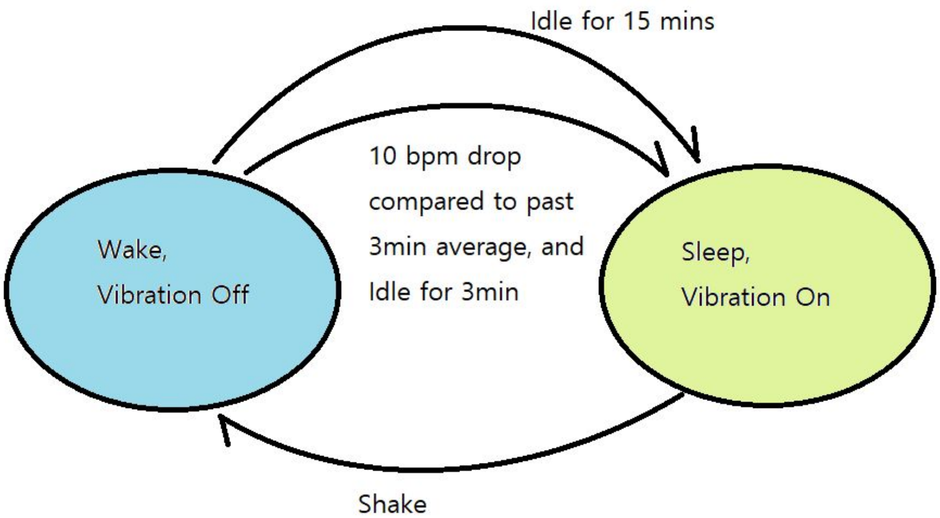


Fig 1.5 State Machine with two states

2. <u>Design</u>

2.1 Sleep Detection Subsystem:

2.1.1: Accelerometer:

The accelerometer measures the movement in your wrist. The sensor should be able to catch the smallest movement in the wrist while typing driving or writing, the most frequent activity that people spend their time that causes sleepiness. Accelerometers generally supports both SPI and I2C but we decided to use SPI specifically because our microcontroller only has one I2C port and SPI is generally faster than I2C giving us better response times.

Accelerometer runs on 3-5 V, and uses the four pins MISO, MOSI, SCLK, CS to communicate with the microcontroller. There is a bypass capacitor since the accelerometer chip can be damaged by a sudden voltage change.

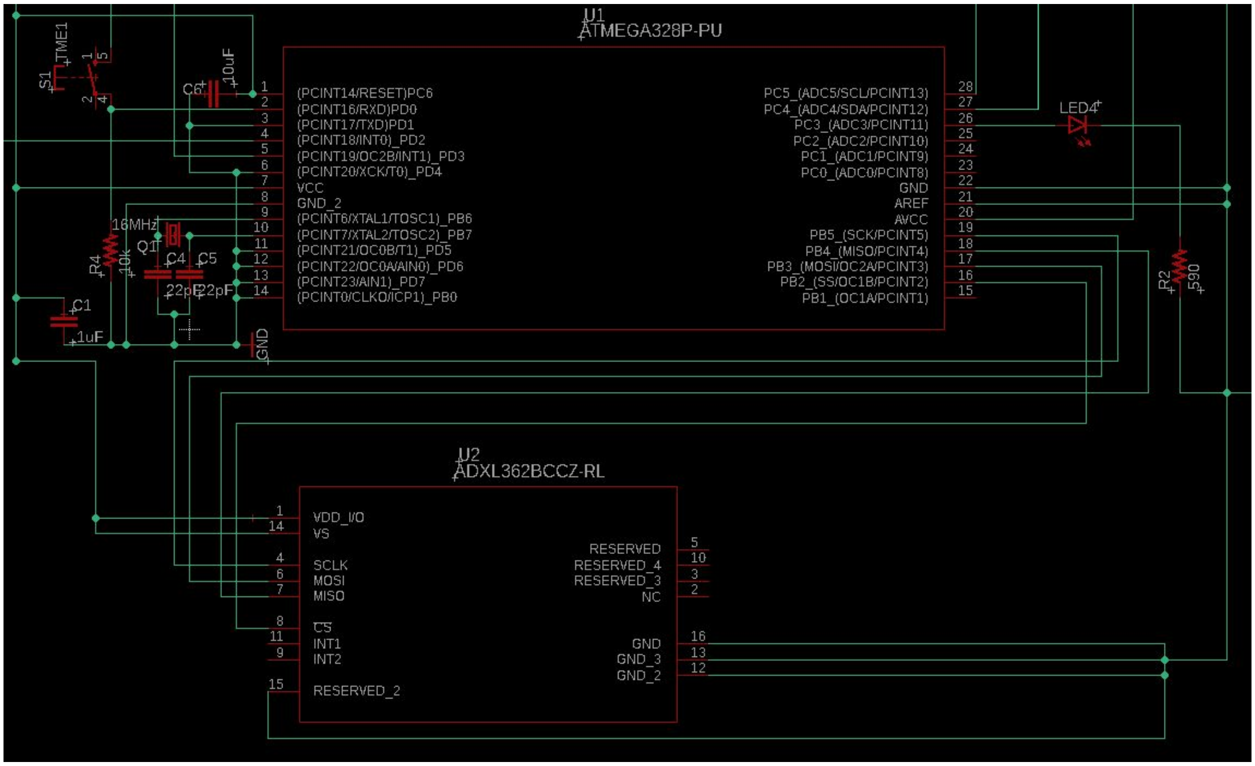


Fig 2.1 Accelerometer schematic. SPI protocol is used.

2.1.2: Pulse Sensor:

A pulse(PPG) sensor is used as out main detection of sleep. Figure 2.5 shows and explains how pulse changes during various stages of sleep. It measures the heart rate by driving LED(s) and one infrared LED onto the skin and using a photodiode to detect the light that has been absorbed by blood cells vs.light that has been reflected by the skin.[3] The sensor will send a blood cell count to the microcontroller via the SDA line and used for processing via code. This blood cell count is then passed through a digital moving average calculator and an infinite impulse response filter to make a pulse wave. As the pulse wave gets continuously updated, equation 2.1 is used to generate a pulse reading in the unit of beats per minute. The peak count is the cycle number the last peak in the pulse wave was found while the current count is the current cycle number. Because we are going to poll the pulse sensor at a rate of 32 Hz, the sampling period will be 1/32 seconds.

$$60/[\,Sampling\ Period * (PeakCount - CurrentCount)]$$ (Eq. 2.1)

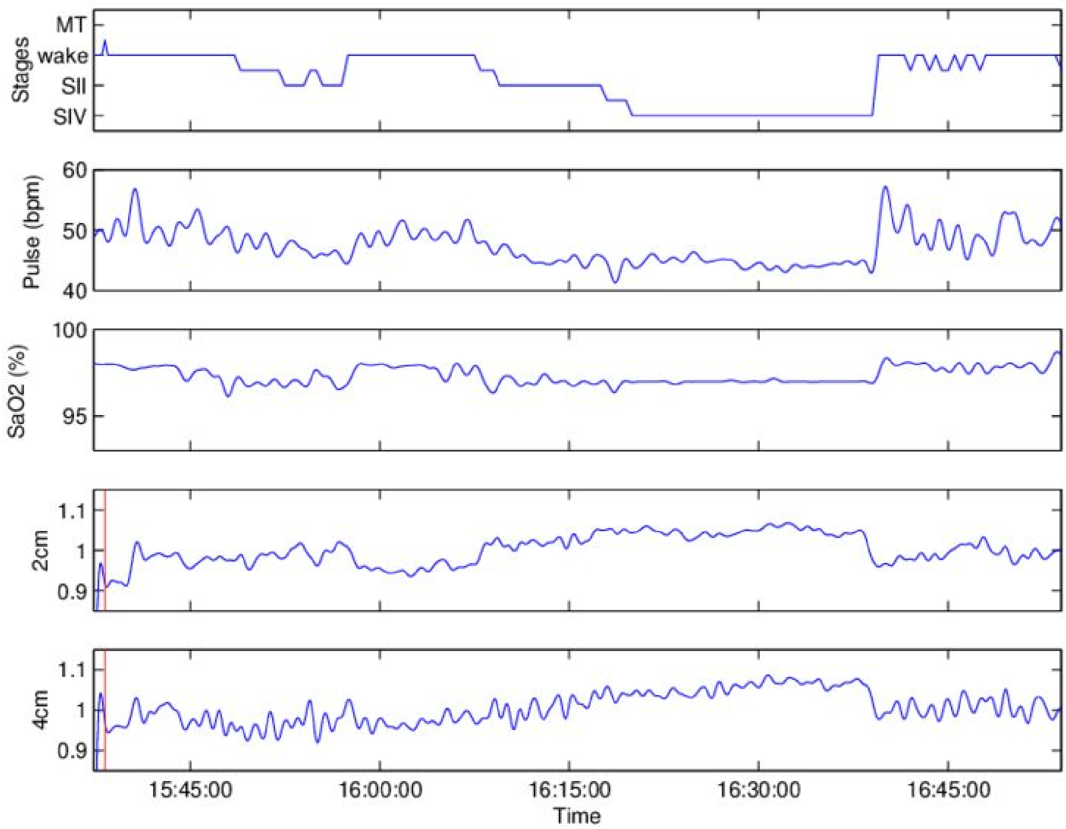


Fig 2.2 (Noponen, 2003) above image illustrates the pulse change with the corresponding sleep state.[4] If we can accurately measure and determine which sleep state the user is in, the accuracy of the device will increase. While other sensors sense the activity of the user, pulse sensor has the most direct way of detecting sleep.

When picking a pulse sensor we had two polarizing options. The first was to use a pulse sensor board sold by Sparkfun. This is a plug and play sensor designed for Arduino that had the circuitry and program done for us and would be a sure-fire way to get a working pulse sensor integrated into our design quickly. Priced at a fair $24.95, it ran at 3 V or 5 V, required no soldering, and was built to be used with the arduino software, making it the most obvious solution for our pulse sensor design. However this sensor had several problems that conflicted with the design philosophy of our device. The first problem was that this sensor was on its own PCB(Printed Circuit Board). This was a problem because we are designing a wristband that had to fit on a human wrist. Ideally, we want everything on a single PCB that goes above the wrist and need to make the overall design as compact as possible. Having multiple PCBs meant that we would have to either expand the container that holds the PCB(s) or that we would have to have two separate containers connected by wires. Both of these cases are undesirable because the first case would significantly enlarge the size of the design while the second will lower the structural integrity of the design due to having to use external wires on a wristband which will be subjected to heavy movement. The second problem was that this sensor was designed to be used on a finger or earlobe and used only one red LED and one infrared LED. Both its hardware and software capabilities were inadequate for reading pulse measurements from the wrist and as a simple hobby device that used an ambient light photo sensor, it's accuracy was debatable.

This lead us to go with the second option, a bare chip that was designed for reading pulse data at a commercial level. The chip we decided on was the BH1792GLC manufactured and developed by ROHM corporation. It runs on an operating voltage in the range of 2.5 V to 3.6 V making it compatible with our power subsystem and could drive two green LEDs and an infrared LED, increasing it's coverage and reach as well as accuracy by having more light waves to read. Another advantage of the BH1792GLC was its small form factor which allowed us to design a smaller PCB accordingly. There were however two challenges that we faced with this decision. This first was the lack of software support, while example code did exist for the sensor, it was made to be used with a sensor shield that was sold by the company separately. To make it compatible with our system, we had to modify the example code and change the values of parameters to suit our needs. Furthermore, the example code only gave us the blood cell count readings and no the pulse readings we wanted. We were able to solve this by using the libraries from an example code from an older model of the pulse sensor(BH1790GLC) and rewriting some more parameters and functions to make the libraries compatible with our pulse sensor blood cell count code. The second problem was getting the sensor soldered onto the board. Not only was the sensor incredibly small, but it also had closed pads that made it difficult to even view the solder connections, making the act of applying solder to the pads under the sensor an arduous task. To help alleviate this problem a little, we extended the pads of the sensor on our PCB. We lost two pulse sensors in the process, but eventually we were able to solder the sensor to the PCB by using solder paste.

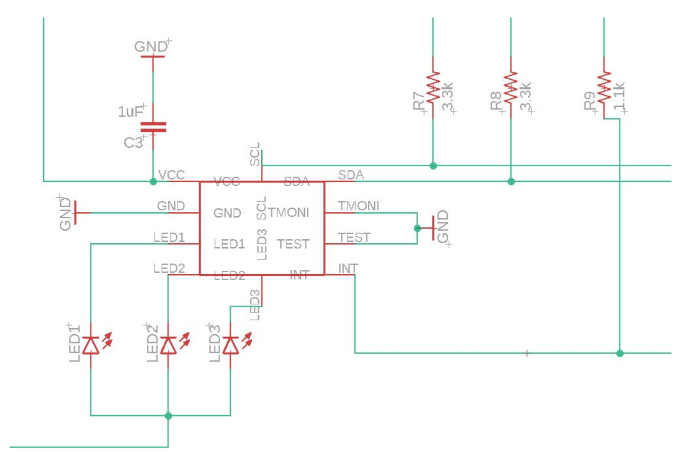


Fig 2.3: BH1792GLC Schematic with I2C connections

The schematic for the BH1792GLC is shown in Fig 2.3. The BH1792GLC has LED drivers built into the chip, so we simply had to connect all three of the LEDs to our 3.3 V output and the corresponding LED pins on the sensor. The three pins that will be communicating with the microcontroller are the SCL, SDA, and INT pins. The SCL line serves as the clock for the I2C connection while the SDA line is the data bus where information will be relayed between the sensor and microcontroller. These pins are active low pins and will need pull up resistors which we can calculate for using the electronic characteristics from the datasheet. Because we need a 10μA current to keep the pins high and our input voltage is 3.3V, we can use Ohm's law to find the resistance needed to keep the SDA and SCL lines high.

$$R = V/I,\ R = 3.3V/0.00001A,\ R = 3,300\Omega$$

The INT pin will be used by the sensor to send interrupts so the microcontroller will know when it can read the data off of the SDA line. It will also be active low and need a pull up resistor. The low output voltage for the INT pin has a max of 0.4V when the input current is 3mA. The microcontroller's pin takes a maximum of .99V and a minimum of -0.5V as a low voltage input. At the given specifications, the output from the pulse sensor will not go over the maximum voltage allowed into the microcontroller. Therefore the safe bet was to follow the datasheet and have the incoming current from our 3.3V source be 3mA. We can once again use Ohm's law to find the resistance needed.

$$R = V/I,\ R = 3.3V/0.003A,\ R = 1,100\Omega$$

2.2 Control Subsystem : Microcontroller

Main control will be handled by the Atmega328P microcontroller, communicating with both of the sensors using the I2C and SPI protocols.[5] It will then use that information to determine whether or not the user is sleeping. If the user is detected sleeping then it will send signals to the wake up subsystem and activate the motor. The button in this subsystem will have two functions, the first will be a basic on/off button for the device. The second function will be a button to stop the vibrating motor after it is pressed. The Atmega supports The Atmega328P was chosen because of its' convenience during coding. We can test and code each subsystem separately with an Arduino UNO before integrating the parts into a PCB and when we have everything working, we can simply load the program into the microcontroller, remove it from the arduino and implement it into our PCB circuit. An LED is used to indicate whether or not the device is operating and is also used as a status indicator when charging.

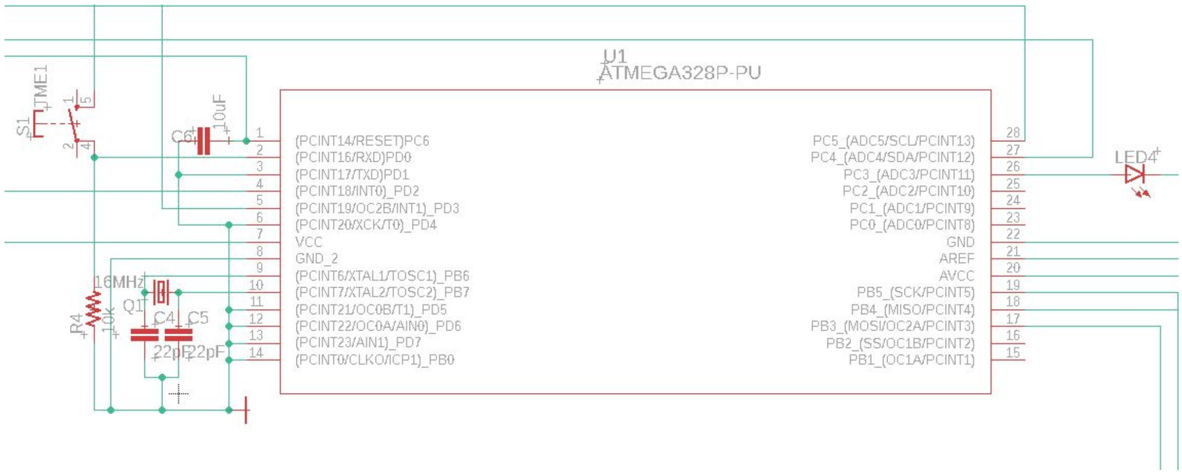


Fig 2.4: Microcontroller Schematic

2.3 Wake up Subsystem : Vibration Motor

Vibration motor will be embedded on the side of the wristband, the activation of the motor will be handled by sending a signal from a GPIO from the microcontroller to put power into the motor. The type of vibration motor that will be used is a coin type ERM(Eccentric Rotating Mass Vibration) motor. This was chosen over the pages type due to its smaller form factor and lack of moving parts which would potentially damage other parts of the circuit or the pcb itself. The vibration motor was chosen due to the same RPM with modern smartphones, which we knew can wake people up.

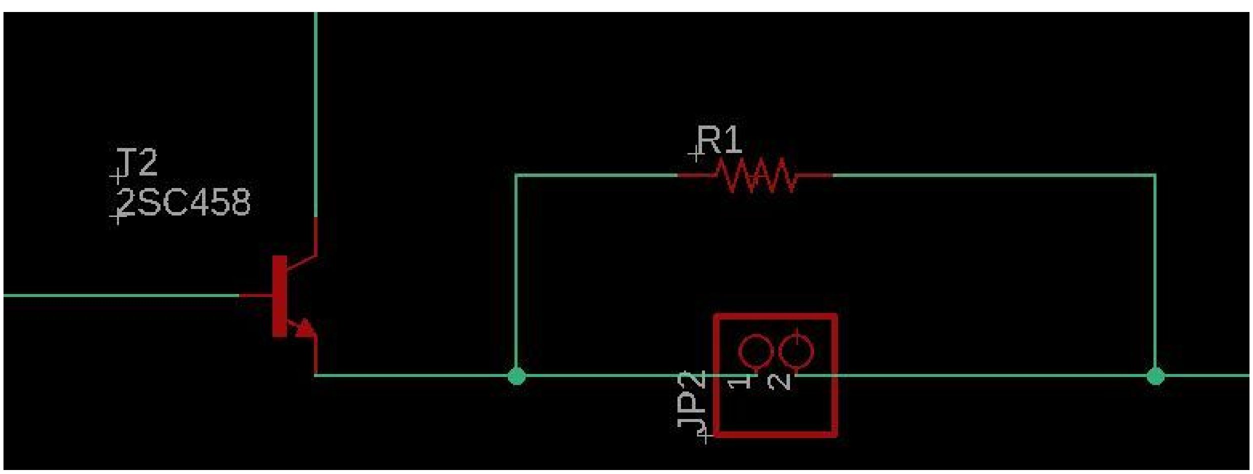


Fig 2.5: Vibration Motor Circuit with Transistor

2.4 Power Subsystem

The main power comes from a 3.7 V, 400 mAh Lithium ion polymer battery, and the battery should be able to power the circuit with all sensors and motors active for at least 30 minutes and run silent for 6 hours straight. The battery will also be able to be recharged using a mini usb port. The battery is connected via linear voltage regulator to other subsystems to ensure constant 3.3V.

$$P = I * V \quad \text{(Eq. 2.2)}$$

Using equation 2.2 for each component, the power consumed can be calculated. The wristband being in silent mode means that both the sensors are running and taking in data, but the motor does not activate and when the wristband is active then the sensors are taking in data with the motor running as well. The power consumed in silent mode was calculated to be 0.0958 W. If we want it to run for 6 hours in silent mode then it will consume a total of 0.5748W/hr and having a battery voltage of 3.7 V would mean we need a 155 mAh battery. To have the wristband running in active mode would change the power consumption to 0.39583 W because the vibration motor is set for 3v and 100mA. If we had wanted the wristband to run for 6 hours in active mode, then the battery would have to be 641 mAh. A 400 mAh was chosen because the user will not have to charge the battery daily and the size of the battery fit well for the wristband and lithium ion polymer is also rechargeable. The L4931 TO-92 linear voltage regulator was chosen because the voltage drop is low and all the components are set to function with 3.3 V. For the charging circuit we used the MCP73831 chip that will control it and had a red LED indicating that the battery is charging and a green LED indicating that the battery is fully charged. This will keep the user from constantly having to switch batteries.

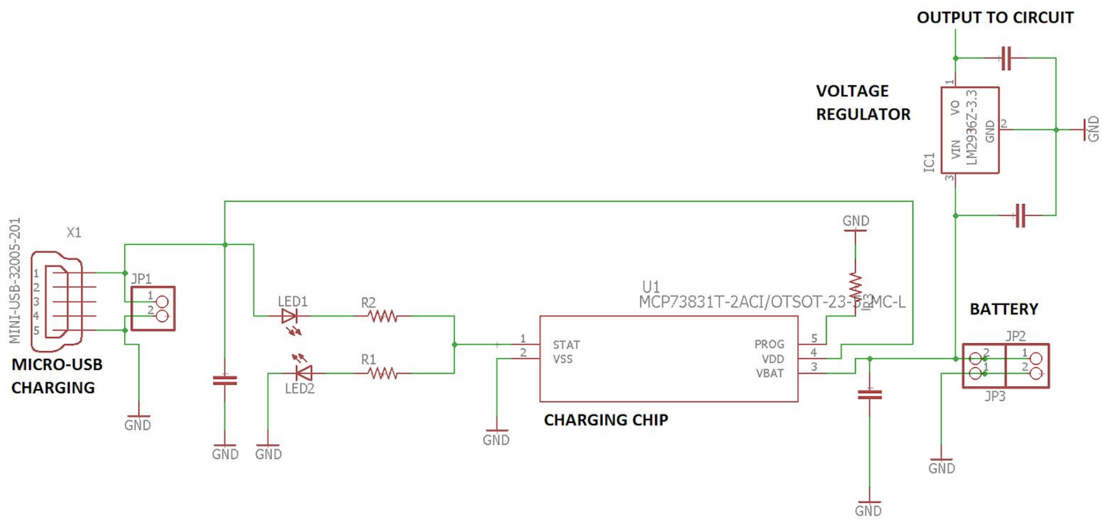


Figure 2.5: Power schematic used to make the PCB.

2.5 Tolerance Analysis:

Our project's main goal is to detect sleep and react quickly enough to activate the waking up mechanisms. This means that the most crucial point of our device is the sensitivity of the sensors and the response time for the sensors to send information to the microcontroller. The accelerometer has a sensitivity tolerance of 0.06 gs at a test condition of +/- 2gs, so around a 3% tolerance between the population of manufactured sensors.[6] We will define the minimum amount of movement an arm must have for a person to be considered awake as a change in position of at a velocity of 10 cm/s in any direction within a span of 1 second. This is minimal speed at which we can say for certain that the user is awake.[7] Setting the arm to start at rest,we find the acceleration for this movement to be:

$$a = \Delta v/t = 0.1/1 = 0.1\ m/s^2 \quad \text{(Eq 2.3)}$$

The average mass of human forearm is 1.72% of his/her body weight and the average body weight of an adult human is 62kgs making the average mass of a human forearm 1.07kgs.[8,9] We can now find the force necessary for this movement with Newton's second law:

$$F = m * a = 1.07 \bullet 0.1 = 0.107\ kg \bullet m/s^2 \quad \text{(Eq 2.4)}$$

3% of this is 0.00321 newtons, therefore we want to avoid having to measure the range between 0.10379 to 0.11021 newtons as the tolerance bias may skew our results and determine that the user is awake when he/she is actually closer to being asleep. We can mitigate this bias by using multiple readings over a short time frame rather than a single reading at an instantaneous time. The sensor can output data at a max rate of 400 Hz meaning we can have 400 readings within a second. We can then determine whether or not the majority of the readings is greater than the minimal .107 newtons of force to get a more accurate reading that will be affected less by the tolerance bias.

Our pulse/biometric sensor is a cornerstone of our design and needs to function appropriately with low bias and high sensitivity. Figure 2.4 shows the sensitivity ratio of the pulse sensor graphed over the wavelength of light used for the LED it drives.[10] There is a near 100% sensitivity ratio at around a wavelength of 540nm meaning that at that wavelength the measurement will be near 100% accurate. However the light read by the sensor is first reflected off of the skin and the skin has a different index of refraction from air possibly changing the wavelength. Furthermore, this index of refraction will change based on the color of the skin. Because we are trying to make this device usable for anyone, we must account the range of indexes of refraction between different skin colors. A study showed that the index of refraction of human skin ranged from 1.2573 to 1.3696 for people of varying skin color. Plugging these values, we can get the wavelength of a 540nm light wave through the skin as shown in Eq. 2.5.

$$\lambda = \lambda_0 / n = 540/1.2573 = 429 \quad \text{(Eq 2.5)}$$

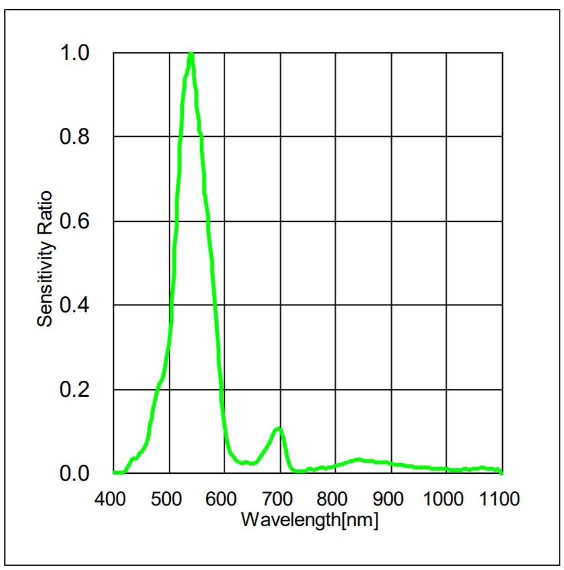


Fig 2.6 Sensitivity Ratio Vs. Wavelength for Pulse Sensor from the sensor's datasheet[10]

3. Design Verifications:

Requirement	Verification
Wrist band **1.** Ensure sensor readings are accurate while moving **2.** Elastic for sizes 6” to 9”	**1.** (a)user puts on the wristband and fitbit (b)user runs at 3km/h for 3 minutes (c) ensure both readings match with error range of 10% **2.** Make sure the diameter of the band while flat can extend from 3” to 4.5”
Sleep detection **1.** Detect within 2 min whether the user is in sleep or not **2.** System does not incorrectly declare state as “sleeping” when sensor readings get unstable	**1.** (a)Track user with video while in a boring sedentary activity while wearing wristband. (preferably the user is already tired) (b) check if algorithm correctly marks state as sleep when the video shows the user is sleeping **2.** (a) User wears the wristband (b) Try rapid rotational motion of the wrist for 1 min, rotational motion of the whole arm in both directions for 1 min. (c) Ensure the system does not incorrectly categorize state as sleep.
Wake Up System **1.** Vibration motor runs when microcontroller sends an On signal. **2.** User wakes up from sleep when vibration motor runs	**1.** (a) Program wristband to send On signal through GPIO periodically with 5 seconds of off/on time. (b) ensure vibration motor activates and deactivates periodically. **2.** (a) Put wristband on with vibration motor in, go to sleep (b) Set a random time during the sleep to run the motor (c) Ensure user wakes up when motor runs
Power Subsystem **1.** Run without charge in silent	**1.** (a) Calculate the power drain of circuit then construct a mock circuit with same

for 6 hrs. **2.** Run without charge with vibration motor on for 30 mins. **3.** Lithium battery recharges when 3.7V- 4.2V is constantly provided	power drain when the vibration motor is not active. (b) power the circuit with the battery and check if device still runs after 6 hrs. **2.** (a) Modify circuit to match power drain when vibration motor is running. (b) turn device on and check if device still runs after 30 mins. **3.** (a) Drain battery (b) Set up the recharging circuit (c) Test the battery using the mock circuit built above and can operate to standard.
Microcontrol **1.** Can communicate simultaneously with two sensors and one vibration motor.	**1.** (a) connect Accelerometer to SPI, Pulse sensor to I2C, then connect GPIO output through a transistor to run the vibration motor. (b) verify that Microcontroller can receive both data. (c)verify that Microcontroller can send signals to vibration motor and make it vibrate

3.1: Physical Wristband Results

The physical wristband was able to fit all of our physical specifications due to its elastic nature. We tested how stable it was by running the accelerometer code with an LED to detect movement. In all five out of five trials, the LED never went off, meaning that the wristband was stable enough for accurate accelerometer readings. Readings for the pulse sensor was unattainable for reasons that will be explained in the next section.

3.2: Sleep Detect Results:

Due to various events such as losing pulse sensors when soldering and having the break out board break on us, we were unable to complete the device fast enough to get readings from pulse sensor. Because our design had no way to display the numerical pulse numbers, we had to rely on the break out board to get readings from our arduino. However that broke before we were able to complete calibrating the sleep detection subsystem code and had no way to properly verify the sleep detection. What we can say is that the pulse sensor was reading the pulse accurate enough to detect significant drops in BPM.

Figure shows a sample of data that was used to derive the sleep detection algorithm. This data shows the pulse of one of our group members as he falls asleep over the course of several minutes. The difference of pulse readings was used to find a suitable number to compare pulse drops to.

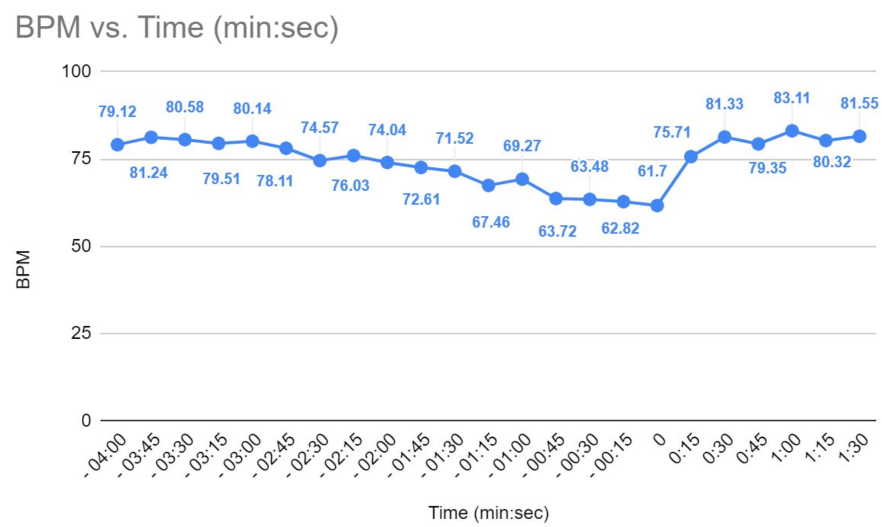


Fig 3.1 Data on pulse readings

3.3 Wake Up Subsystem Results:

After our vibration motor broke on us during the demo, the new we ordered works as intended clearing our verification for having a strong enough vibration through communication with the microcontroller.

3.4: Power Subsystem Results:

The microcontroller was programmed to have the sensors constantly running and had the vibration motor constantly running when power is inputted into the system. The battery was connected and we let the wristband run for 30 minutes continuously. This verified that the wristband can run for 30 minutes without having to be recharged. We were able to measure that the micro-usb charger can charge the battery at 100 mA current rate.

3.5: Microcontroller Results:

We ran multiple variations of code to test communication between the microcontroller and the other components and all of our trials have shown that the microcontroller is communicating properly with every component.

4. Cost

Microcontroller: ATmega328p : $6.99
Accelerometer: ADXL362BCCZ-RL $8.94
PPG ROHN Biometric sensor BH 1792GLC $10.03
Vibration motor C0834B011F $3.28 (digikey)
Battery: LIPO 801735 400mAh 3.7V $6.95
C201 capacitor x2
IXTP01N100D transistor

-Parts total: $36.19
-Labor total: $40/(h x people) x 10h/week x 16week x 3 people x 2.5 = $48,000

Total cost: $48,036.19

5.Schedule

Week	Junfei	Bum "Jun"	Jaime	Group
9/30/19				Order and Finalize components
10/7/19	Sensor and motor Subsystem	Controller Subsystem	Power Subsystem	PCB Design Schematic
10/14/19				PCB Design continued
10/21/19	Code for pulse sensor	Code for accelerometer	Build test circuits	Testing components with breakout board, first PCB order
10/28/19	PCB test with Pulse sensor	PCB test with Accelerometer, found bug in pcb	PCB test with power circuit	
11/4/19	Test systems with existing PCB	Test systems with existing PCB	Test systems with existing PCB	Final PCB Re order PCB Code for testing subsystemsdxc

11/11/19	Debugging Sensors	Debugging Controls	Debugging circuit	Testing code for Subsystems while PCB connected
11/18/19				Continue debugging Start working on Final presentation
11/25/19 (Fall Break)				Working Product for Final Demo
12/2/19	Debug Pulse sensor	Quick Debug fix on broken vibration motor	Demo process organization	Work on Presentation for Final Demo (Monday/Tuesday/Wednesday)
12/9/19	Powerpoint setup	Data collection, diagram	System organization, Presentation micromanagement	Final Presentation due next week (monday/tuesday) Final Report due next week (wednesday)

6. Safety and Ethics

The user should not have to take off the wristband when they go outdoors, so the wristband must be water resistant to keep the device safe from a potential shortage inside the wristband.

Using or charging the device in potentially explosive atmospheres, such as areas of high amount of flammable chemicals, vapors, or particles are in the air. In case of a device malfunction during use or while being charged, a spark of electricity can light an explosion.

The components will have to be stable and have strong connections so that they are not broken when the vibration motor turns on. If they do not then the wristband connections can break and can overload the capacitor causing it to explode or create a short in the wristband. Therefore we will follow code #6 of the IEEE code of ethics and promise “to maintain and improve out technical competence,”[11] and create a complete and safe device with no technical defects that may harm the user.

7. Conclusion

The final product delivery was too late to test on a subject going into sleep. However, we were able to meet with some main requirements such as being able to communicate with both sensors, vibration motor activation is correct. We can show the accelerometer can detect movements over the wanted threshold. Also the pulse sensor properly catches the drop in BPM as shown in the demo by Jaime running up and down the stairs.

However, since we did not have sufficient UI to confirm the readings were accurate, some of the requirements were not met, such as ensure sensor readings are accurate within 10% of error. The original plan was to compare the wristband data with a commercial device FitBit, but since the wristband does not have a UI to communicate the measured data, the verification was not possible.

Citations:

[1] Morgan, Kevin. (2016, Nov) The Sealy Sleep Census. Sealy Corp, TX[Online]. Available: http://www.sleep-census.com/media/PDF/Sealy Sleep Census final report.pdf.

[2](2019, July 27)How Do Sleep Trackers Work? Understand Sleep Tracking Tech. Tuck Sleep.[Online]. Available:
https://www.tuck.com/how-sleep-trackers-work/

[3]Fortine, Giancarlo. Giampà, Valerio. (2010, Feb) PPG-based Methods for Non Invasive and Continuous Blood Pressure Measurement: an Overview and Development Issues in Body Sensor Networks, IEEE Italy[Online]. Available:
https://s3.amazonaws.com/academia.edu.documents/35937749/05480201.pdf?response-content-disposition=inline%3B%20filename%3DPPG-based_Methods_for_Non_Invasive_and_C.pdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWOWYYGZ2Y53UL3A%2F20191004%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20191004T025424Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=180ea9b41a5d68e01b287f02a4cb148c80ffcdcaaca16d2591c60fcabb6810c7

[4] Tommi Noponen(2003, july) Near-infrared measurements of hemodynamic and oxygenation changes on the frontal cortex during breath holding, hyperventilation, and natural sleep, SPIE. [Online] Availabe:
https://www.researchgate.net/publication/228542519_Near-infrared_measurements_of_hemodynamic_an

d_oxygenation_changes_on_the_frontal_cortex_during_breath_holding_hyperventilation_and_natural_sleep

[5] (2015, Nov) ATMEL 8-BIT MICROCONTROLLER WITH 4/8/16/32KBYTES IN-SYSTEM PROGRAMMABLE FLASH. Atmel Corp, CA[Online]. Available: https://www.mouser.com/datasheet/2/268/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48P-1315288.pdf

[6] (2012, Nov) MPU-9250 Product Specification Revision 1.1. InvenSense Inc, CA[Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/TDK%20PDFs/MPU-9250_Rev_1.1.pdf

[7]Y Xu, AV Terekhov, ML Latash, VM Zatsiorsky (2016, May) Forces and moments generated by the human arm: Variability and control . Exp Brain Res[Online]. Available:

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3501382/

[8] Plagenhoef S(1983) Anatomical data for analyzing human emotion, Research Quarterly for Exercise and Sports, CA[Online]. https://www.researchgate.net/publication/261612048_Anatomical_Data_for_Analyzing_Human_Motion

[9] Quilty-Harper Conrad, (2012, Jun) The World's fattest countries: how do you compare?. The Telegraph Corp, UK[Online]. https://www.telegraph.co.uk/news/earth/earthnews/9345086/The-worlds-fattest-countries-how-do-you-compare.html

[10] (2017, Dec) Sensor for Heart Rate Monitor ICs Optical Sensor for Heart Rate Monitor IC. ROHM Corp, UK[Online]. Available:https://www.rohm.com/datasheet/BH1792GLC/bh1792glc-e

[11] Ieee.org. (2006). *IEEE IEEE Code of Ethics*. [online] Available at: http://www.ieee.org/about/corporate/governance/p7-8.html [Accessed 9 Sep.. 2019].