Sleep Cycle-Triggered Lighting Wake up

By
Group 21
Melech Lapson
Han Chen
Weipeng Wang

Final Report for ECE 445, Senior Design, Spring 2021
TA: Dean Biskup

06 May 2021
Project No.21

Abstract

Many people have a hard time waking up in the morning and are really upset when they are awakened by an alarm. Our project is trying to make this process easier for people. Our project is a smart wristband that can detect sleep cycles by sets of datas and use the data to determine the right time to wake the user up. Our project managed to successfully wake a person up at the right time, which in our finding is the last light sleep cycle. In this report, we will have our product's background information and design process.

1. Introduction

1.1 Objective and Background

Many people have a hard time waking up in the morning. There are so many of us who just press the snooze button on our alarms even though we are going to be late for work. According to a survey of almost 20,000 people done by the French tech firm Withings, more than 57% of the Americans are snoozers [1]. Many people reported feeling tired or fatigued even when they slept for 7 to 8 hours, which is the suggested amount by experts [2]. Therefore, we are trying to find a way to help people wake up easier and feel more energetic.

The most common way people wake up is by setting an alarm, using either a cellphone or a clock. The problem with this is that people wake up too abruptly from the loud sound of the alarm clock. According to a survey conducted by Niall McCarthy, only 1 in 7 Americans who wake up by an alarm feel refreshed every day. Most of those people's days are ruined by the noisy alarm clock [3].

Some new technologies are attempting to track a user's sleep cycle to know the optimal time to wake them up [4]. According to an article by MedicalNewsToday, the best time to wake up is at the end of a sleep cycle, when the sleep is the lightest [5]. Other studies also show that using light can help your body prepare to wake up [6]. Our project combines these technologies to create a device that will track your sleep cycle and then trigger a lightbulb to turn on. This will allow us to leverage the benefits of both technologies to help people wake up in the morning.

In general, our team is trying to make a device that fits on a band or watch so that a user can wear it during his/her sleep. The device will collect three sets of data (pulse, sound and movement) from the users during their sleep. The data will be utilized to determine which sleep cycle the user is currently in. When it detects one is in a sleep cycle where it is suitable to wake the user up, it will send a signal using a bluetooth module to the light bulb, turning it on. This would require the users to buy the specific light bulbs along with the band.

1.2 High Level Requirements

1. The device should have two modes for the user to select. The first is manual mode where the device will wake up the user according to the time the user set. The second is smart mode, where the alarm will wake the user up according to the sleep cycle of the user and the range of time the user gave to the device.
2. The band system should be able to gather data using microphone, pulse sensor and accelerometer, and calculate the sleep cycle of the user using the data and transmit a signal to the lightbulb system.
3. The light system should use light of 2000 lumens to wake the user up.

1.3 Software Workflow and Block Diagram

There are two subsystems in our project: the band subsystem and lightbulb subsystem. Each of these subsystems has its own microcontroller (See Figure 4). The band subsystem's microcontroller processes the data passed to it by the three sensors in the subsystem: the pulse sensor, microphone, and accelerometer. Using heart rate, sound, and movements, we can track the sleeping cycle of the user [7].

We accomplished this by combining these sets of data and analyzing them by comparing the datas with the thresholds that we found based on our research (see Figures 2 and 3) [8][9].

In the beginning, the light in the light subsystem is off, and the pulse sensor, microphone and accelerometer is constantly measuring the user's heart rate, sound and movement respectfully, and in the meantime, the band subsystem is using the data to see if the user is in light sleep stage or not, if so, the band subsystem will send a signal to the light subsystem, once the light subsystem receives the signal, the light will be turned on, and the user will be waked up. (see Figure 1)

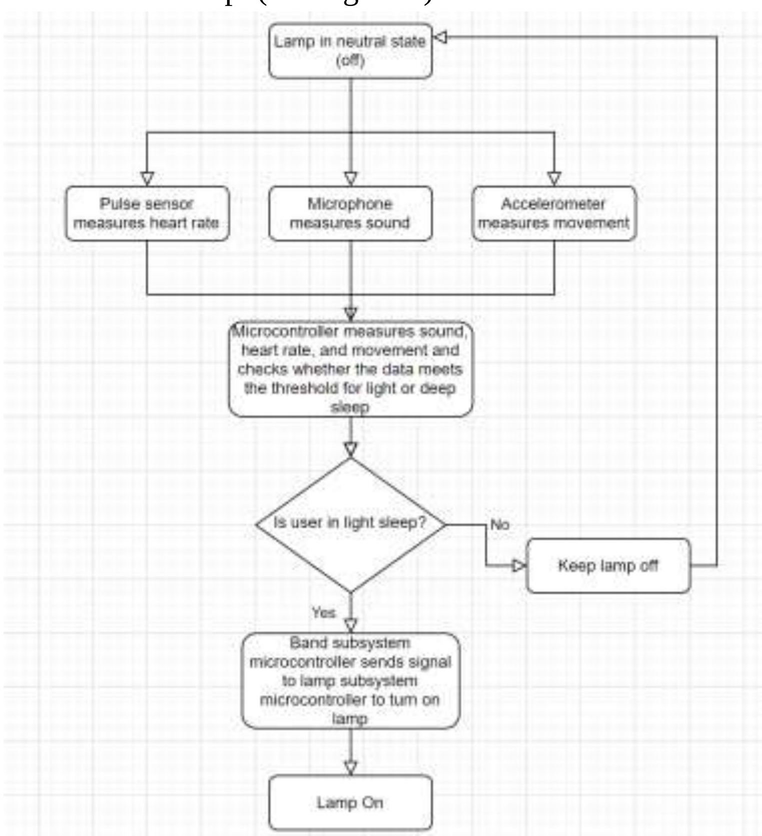


Figure 1: Software flow diagram

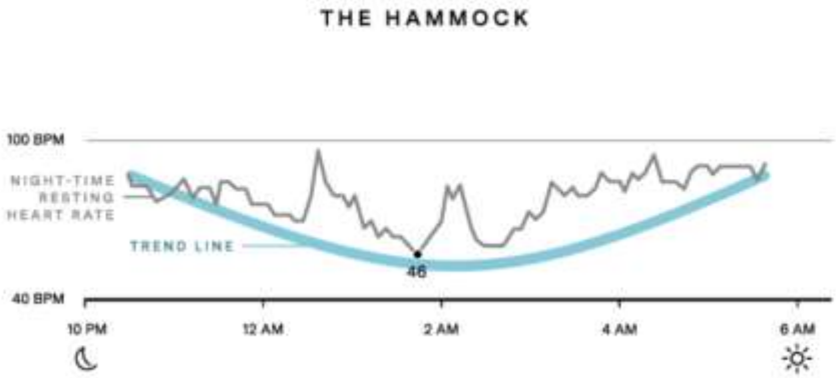


Figure 2: Normal heart rate range during sleep [8]

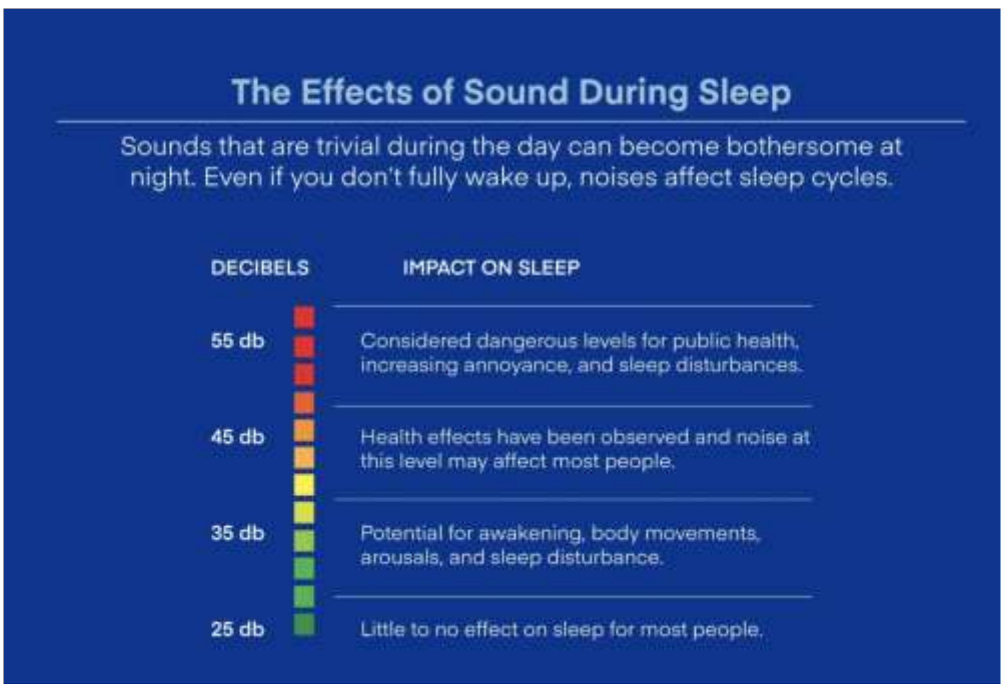


Figure 3: The Effects of Sound During Sleep [9]

The design of our project included two subsystems, the band subsystem and lightbulb subsystem. Figure 4 shows all our components and the connections between them. The band subsystem collects data from the user and uses it to detect a user's sleep cycle and then sends a signal to the lightbulb system. We originally planned to use a clock chip to keep time but ended up relying on the built-in Real-Time Clock on the microcontroller chip.

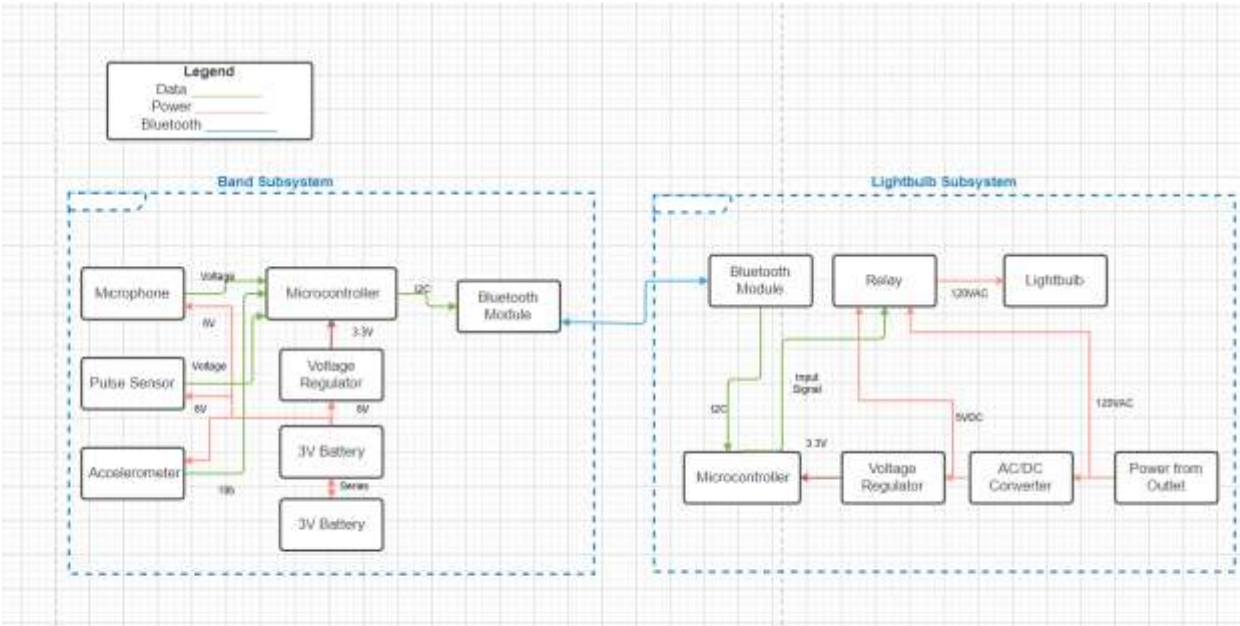


Figure 4: Block Diagram

2. Design

2.1 Component Design

For our design we used two microcontrollers, three sensors, two Bluetooth modules, two voltage regulators, two batteries, an AC/DC adapter and a relay. The individual components are described below.

2.1.1 Microcontrollers

For our microcontroller, we used the 64-pin STM32L552RET6 microcontroller. It has 47 input and output pins and 256KB of SRAM. We needed the I/O pins for attaching all the sensors and Bluetooth modules, and the memory was needed to store the sensors data over time. We originally tested with STM32L552RCT6 chips, but switched over to the STM32L552RET6 chips when the STM32L552RCT6 went out of stock.

2.1.2 Bluetooth Modules

For our Bluetooth modules we used HC-05 modules as they had a large enough range of 10m and had documentation [12] on how to pair two of them together to communicate with each other.

2.1.3 Relay

For our relay we used a 5V single channel HiLEtGo Relay which would allow us to control the flow of 125VAC with a 5V input. We chose this relay because it could be controlled by 5V which we could provide with our AC/DC adapter and it was rated for 120VAC which powered the lightbulb

2.1.4 Voltage Regulators

For our voltage regulators we used AMS1117 SOT223 voltage regulators. This would allow us to convert 5-15V into 3.3V that would power our microcontrollers. They also allowed for up to 1.1A of current which would also help in powering all our components.

2.1.5 AC/DC Adapter

For our AC/DC adapter we used an OSMW2-5 which converted 120VAC to 5VDC this was necessary to bring down the voltage so we could use it to power the relay and input into the voltage regulator to power the microcontroller in the lightbulb subsystem. It also provided 400mA of current which we would need to power the relay and microcontroller.

2.1.6 Sensors

For our sensors we used an ADXL343 accelerometer, a MAXIM9814 microphone, and a pulsesensor.com pulse sensor. They all required about 5V to power and had analog outputs. THe accelerometer was set to a +-2g sensitivity which would be reasonable for movement of a user's arm during sleep. The microphone could measure up to 60dB of sound which is about as loud as a user will make during sleep (See Figure 3). The pulse sensor accurately measured heart rate (See Table 1 in verifications below).

2.1.7 Batteries

For our batteries we used 3V CR2032 Lithium ion batteries. Based on the battery datasheet, the batteries would last at least 1100 hours while in use based on the 40kΩ load resistance of the microcontroller mentioned in the microcontroller's datasheet.

2.2 Printed Circuit Board Components Design

2.2.1 Holes

To test the pins of our microcontroller, we created a PCB component for holes that would be connected individually to each pin of the microcontroller so we can test each pin. Here is the footprint and symbol.

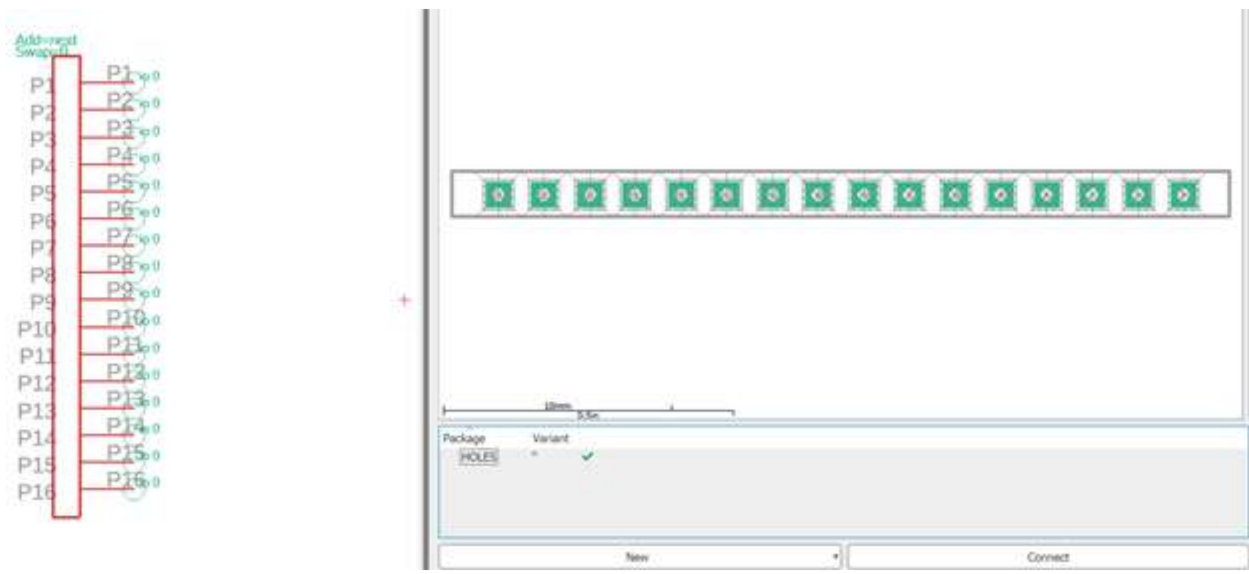


Figure 5 Footprint and symbol for holes for the testing PCB

2.2.2 LEDs

We decided to use LED to keep track of when some of the pins on the STM32 were on or transmitting. On our PCB we currently designed, we left a space to test that the LED works in general. Here is the footprint and symbol.

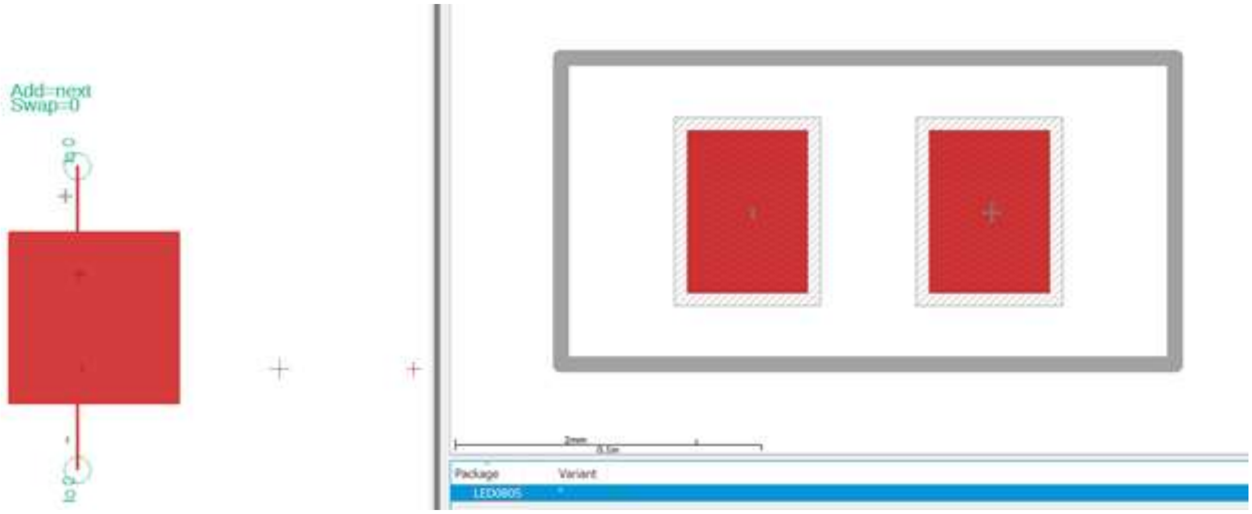


Figure 6 Footprint and symbol for LEDs for the testing PCB

2.2.3 SOT223 Voltage Regulator

In order to maintain the correct voltage for our microcontroller and sensors, we needed a 6 to 3.3V voltage regulator and decided to purchase the SOT223. I had to create the PCB component in Eagle and here is the footprint and symbol.

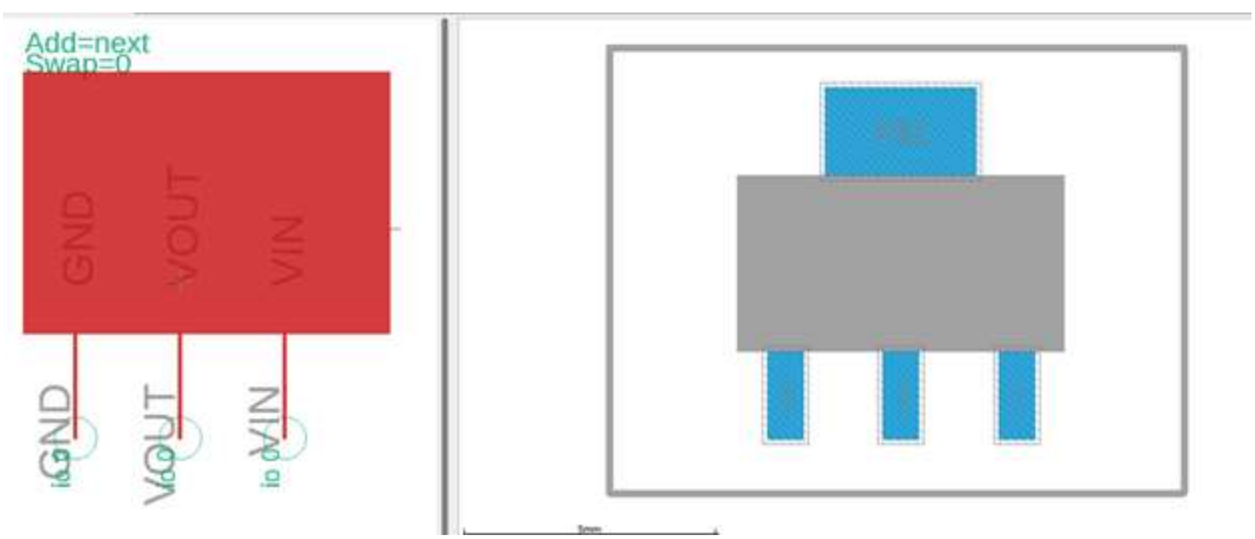


Figure 8 Footprint and symbol for SOT223 for the testing PCB

2.2.4 STM32L552 Microcontroller

Our microcontroller for both of our subsystems is the 64-pin STM32L552RET6 microcontroller. We had to create the PCB component in Eagle for the microcontroller as shown in Figure 9.

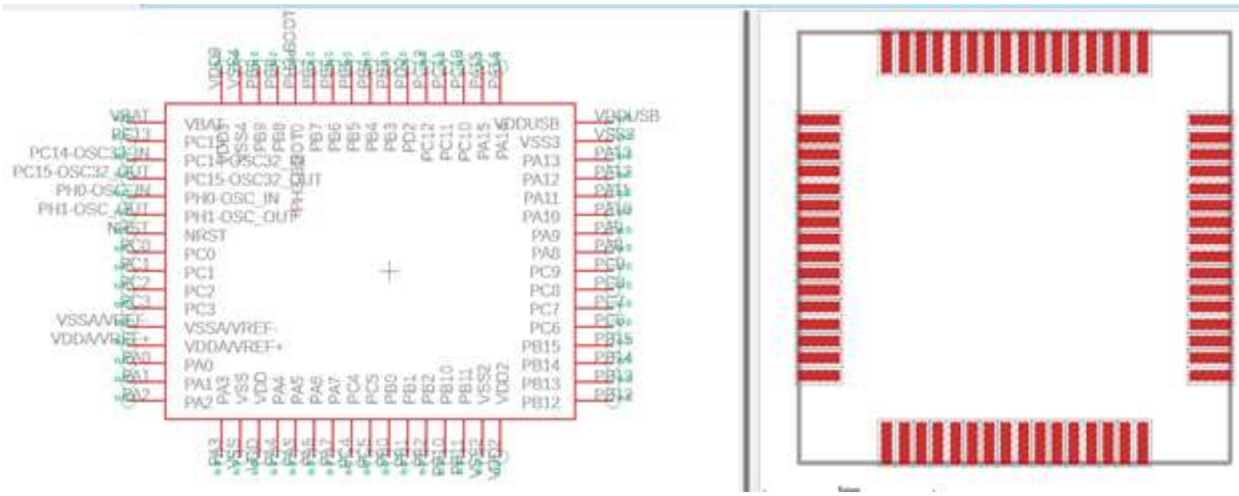


Figure 9 Footprint and symbol for STM32L552 for the testing PCB

2.3 Printed Circuit Board Design

Finally, I had to design the actual PCB itself that we would use to test our SMD parts, especially the voltage regulator and microcontroller (shown in Figures 10 and 11). The PCB contained a spot for the microcontroller and voltage regulator, as well as a couple LEDs for testing.

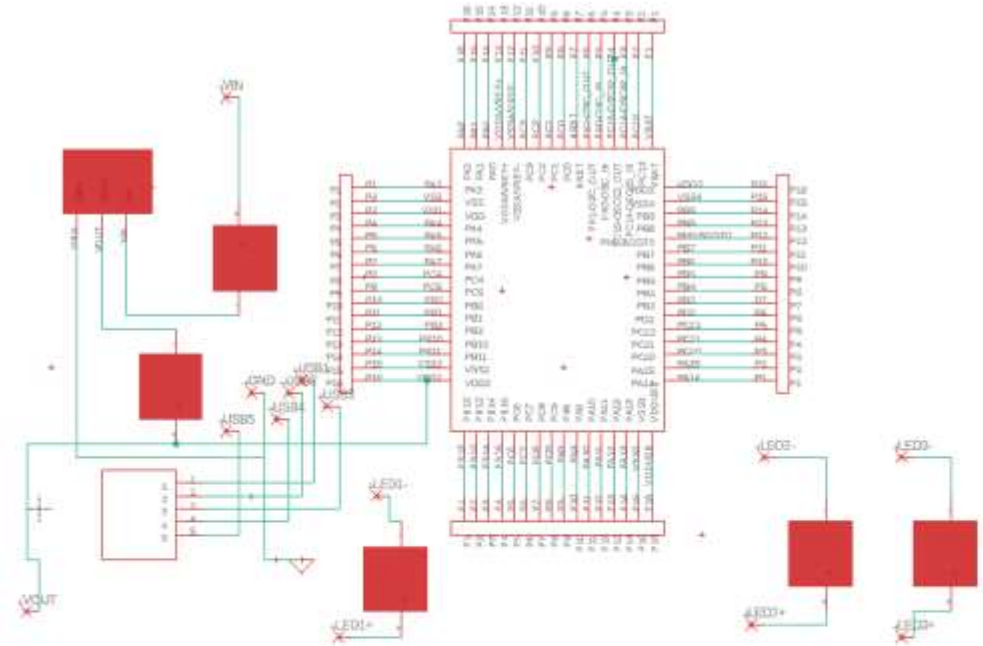


Figure 10 PCB schematic

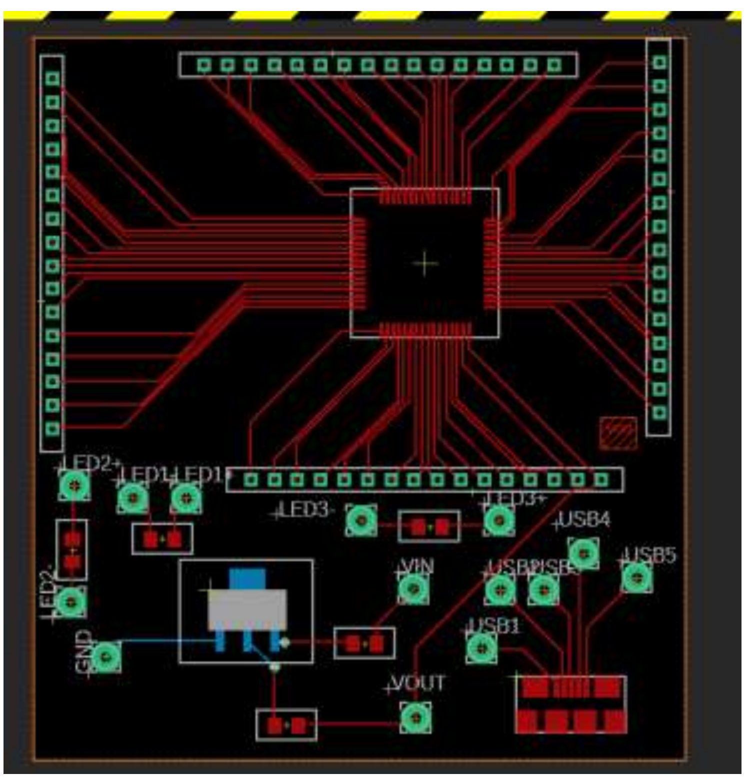


Figure 11 PCB board layout

3. Design Verification

3.1 Hardware Verification

We unti tested all hardware components in both our subsystems and the verifications are detailed in Table 1 below. The results show that all the components met their specifications.

Component	Verification	Result
Voltage Regulator	5 and 6V input	3.303 and 3.27V output respectively
Batteries	Voltage output	Between 3.035-3.165V depending on the battery
Microphone	Output difference between silence and talking	1.97 Volts peak to peak difference.
Lightbulb	120V input	Light turned on
Bluetooth Module	2.5-6V input	LED turned on
Relay	Connected 120V to outputs and alternated between 0 and 5V input	Light turned on when 5V input was given and turned off when input was 0V.
AC/DC Converter	120VAC input	4.86VDC output
Accelerometer	Difference in output when shaken vs when still	498 bits difference in output from still to shaking
Pulse Sensor	Sensor BPM reading vs manual reading from wrist	Measured 78 BPM where as manual check measured 79 BPM

Table 1 Testing results for units tested so far

3.2 Software Verification

3.2.1 Band Subsystem

The band system was able to gather data from the microphone, pulse sensor, and accelerometer. The band system was able to calculate the sleep cycle of the user using the data and then transmit the signal to the lightbulb system. We confirmed that the band system is able to gather data using the three sensors by logging the collected data with the serial monitor program we wrote using python (see Figure 12).

In the band system, data was read from the analog input pin, and was transferred using the USB port, where the data can be captured by the python program. For the pulse data, we used the peak to peak interval of the analog data collected by the pulse sensor (Figure 13) to calculate the

BPM. For the microphone data, we use the high to low distance of the analog data collected by the microphone sensor (Figure 14) and the method described online [10] to calculate the noise level.

```
import serial
import csv
import sys
import os
import time
if os.path.exists("out.txt"):
    os.remove("out.txt")

f = open('out.txt', 'a')
if __name__=='__main__':
    try:
        ser = serial.Serial('/dev/cu.usbmodem141403', 9600, timeout=1)
        print("connected to: " + ser.portstr)
    except serial.SerialException:
        print("serialException")
        pass
    while True:
        # Read a line and convert it from b'xxx\r\n' to xxx
        line = ser.readline().decode('utf-8')[:-1]
        if line:
            print(line)
            f.write(line + ",\n")
f.close()
ser.close()
```

Figure 12: program that read serial data from band system

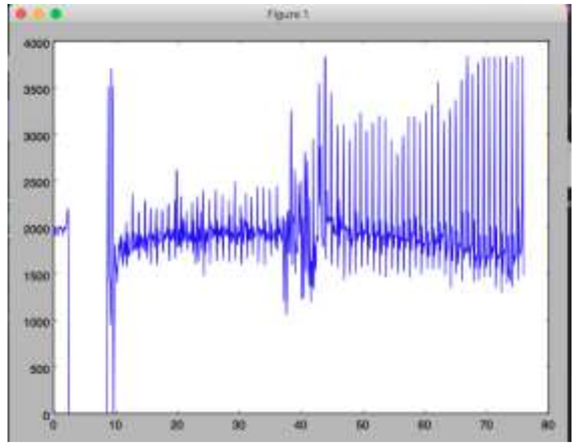


Figure 13: Analog data collected by the pulse sensor

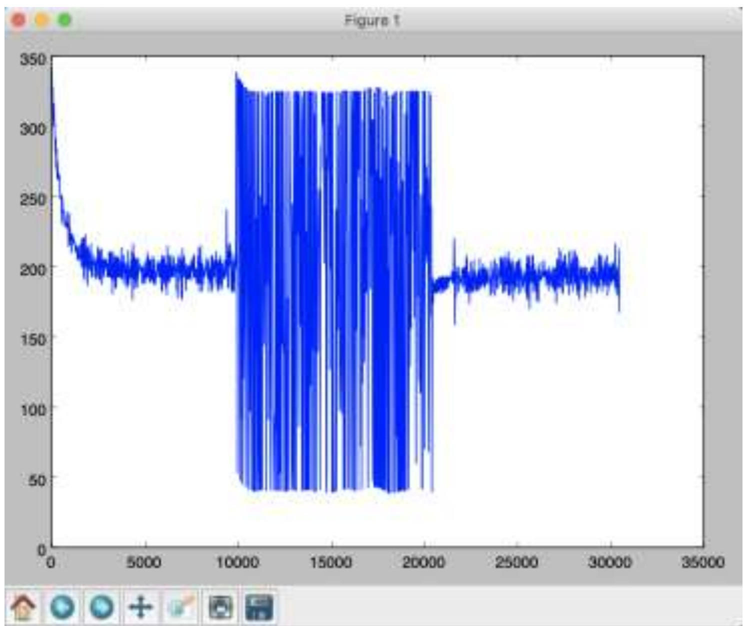
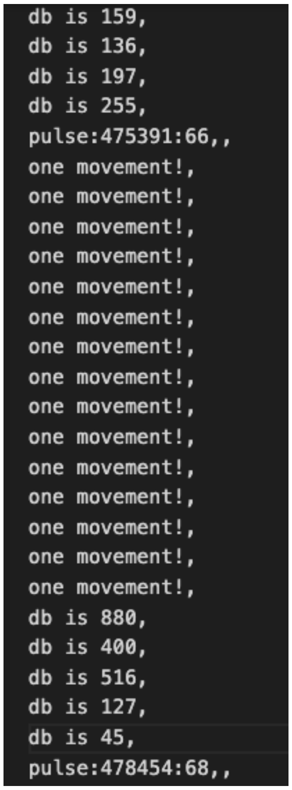


Figure 14: Analog data collected by the microphone

Finally, we detected the movement by contracting the recorded acceleration with the initial acceleration, if the difference is larger than the threshold we set, we say we detected a movement. In the data snippet (Figure 15), the pulse data is sent using the format "pulse: *timestamp*: *data*", the microphone data is sent using the format "db is *data*", and when a movement is detected, the band system will send "one movement!" to the program. Thus, we can see that the band system is gathering data from microphone, pulse sensor and accelerometer.

We verified that the band system could calculate the sleep cycle by having one of the team members wearing it for a night, and as shown below in the screenshot of the video taken while the user is sleep (Figure 16), it successfully send a signal to the light system and the light is on, indicating that it used our algorithm to detect that the user is in the light sleep stage.



```
db is 159,
db is 136,
db is 197,
db is 255,
pulse:475391:66,,
one movement!,
one movement!,
one movement!,
one movement!,
one movement!,
one movement!,
one movement!,
one movement!,
one movement!,
one movement!,
one movement!,
one movement!,
one movement!,
one movement!,
one movement!,
db is 880,
db is 400,
db is 516,
db is 127,
db is 45,
pulse:478454:68,,
```

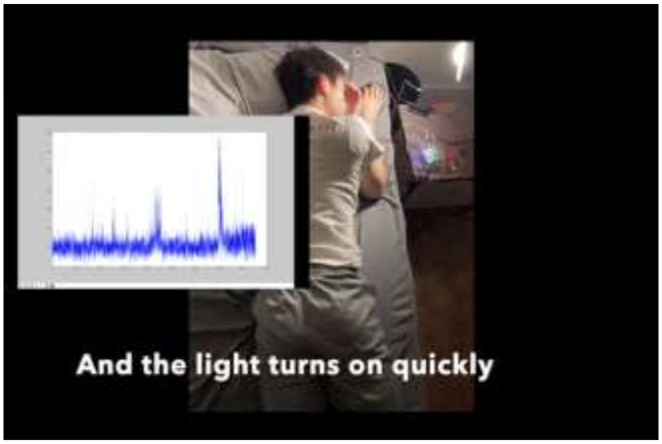


Figure 15: Data snippet collected using the python program

Figure 16: Screen shot from the video [11]

3.2.2 Light Subsystem

We verified that the light system was able to use the light to wake the user up when it received the signal from the band system by having the band system to print a message indicating it has sent the signal. In addition, we could see that the light turned on.

4. Cost and Schedule

4.1 Cost

Part #	**Manufacturer**	**Description**	**For**	**Price**	**Qty**	**Total**
STM32L552	STM	8 bit controller	Both subsystems	$3.22	2	$17.09
Hc-05	HiLetGo	Bluetooth chip	Both subsystems	$7.99	2	$17.42
Max9814	Adafruit	microphone	band	$14.00	1	$14.00
ADXL343	Analog Devices	Accelerometer	band	$19.22	1	$19.22
	PulseSensor.com	Pulse sensor	band	$33.94	1	$33.94
AMS1117	Advanced Monolithic Systems	Voltage regulator	Both subsystems	$0.12	2	$.24
CR2032	Maxell	Lithium batteries	band	$.69	5	$3.76
2963	HiLetGo	Relay	lightbulb	$3.15	2	$6.31
	HENGRC	Mesh Milanese Loop band	band	$9.45	1	$9.45
OSMW2-5	OSM Inc.	AC/DC Converter	lightbulb	$22.50	1	$22.50
L125A23N25 UNV27K	TCP	Lightbulb	lightbulb	$19.97	1	$19.97
Total						**$161.40**

4.2 Labor Cost

We estimate a grad student at UIUC would make $27/hour and that it will take each of the three partners 2 hours per week for 10 weeks to complete this project. Therefore, our labor cost estimate would be $1,620.

5. Conclusion

5.1 Results

The device was able to record the heart rate, sound, and movements through the sensors. It managed to wake up the user during his light sleep cycle, which means the microcontrollers in the band and lightbulb subsystems both worked. Nonetheless, the algorithm for analyzing these datas still needs to be improved as we only managed to figure out the pattern for a few users. Each individual has different sleeping patterns in their sleep cycles. Even one individual can have multiple sleeping patterns when their living habits or mental state change. The algorithm needs to be more comprehensive to ensure the results are reliable.

5.2 Future Work

There are several ways that the project can be taken to another level. The first thing our team would do is to implement more tests and gather more data. With a larger scale of data, we would try to build a neural network and use the data to train it. The second thing we would work on would be to minimize the room the hardware is using to make it fit in a band. In this way, it will be more convenient for the candidates to implement our tests. In addition, we will try to implement UI interface and other functionalities like WIFI and data transferring.

6. References

[1] Lockett, Eleesha (2020, June 18) Everything to Know About the Five Stages of Sleep. Retrieved May 03, 2021, from https://www.healthline.com/health/healthy-sleep/stages-of-sleep#fun-facts

[2] Roitmann, E. (2017, June 16). To snooze or not to Snooze: The truth about the snooze button. Retrieved May 03, 2021, from https://blog.withings.com/2017/03/16/to-snooze-or-not-to-snooze-the-truth-about-the-snooze-button/

[3] McCarthy, N., & Richter, F. (2015, June 08). Infographic: Americans are tired most of the week. Retrieved May 03, 2021, from https://www.statista.com/chart/3534/americans-are-tired-most-of-the-week/#:~:text=45%20percent%20of%20Americans%20sleeping,or%20more%20days%20per%20week

[4] Roy, S. (2014, August 28). 57% of Americans hit the snooze button. Retrieved May 03, 2021, from https://www.sleepreviewmag.com/sleep-health/sleep-whole-body/heart/americans-snooze-button-withings/

[5] Sleep cycle: Sleep Tracker, monitor & alarm clock (2020, December 18). Retrieved May 03, 2021, from https://www.sleepcycle.com/

[6] Best time to sleep and wake up: Sleep patterns. (n.d.). Retrieved May 03, 2021, from https://www.medicalnewstoday.com/articles/best-time-to-sleep-and-wake-up#sleep-cycle

[7] Brain Basics: Understanding Sleep (2019, August 13). Retrieved May 03, 2021, from https://www.ninds.nih.gov/Disorders/patient-caregiver-education/understanding-sleep

[8] Heart Rate During Sleep: Look for These 3 Patterns (2020, February 11). from https://ouraring.com/blog/heart-rate-during-sleep/

[9] How Sound Impacts Your Sleep Cycle (2021, April 21) from https://amerisleep.com/blog/sound-impacts-sleep-cycle/

[10] Measure Sound/Noise Level in dB with Microphone and Arduino. Circuit Digest. (2019, August 27). https://circuitdigest.com/microcontroller-projects/arduino-sound-level-measurement.

[11] Demo Video (2021, May 4) https://drive.google.com/file/d/1aD3P-R9aUtdVBOHvd-J08iDoB3YYIUnc/view?usp=sharing

[12] Dejan, et al. "How To Configure and Pair Two HC-05 Bluetooth Modules as Master and Slave: AT Commands." HowToMechatronics, 5 Feb. 2021, https://howtomechatronics.com/tutorials/arduino/how-to-configure-pair-two-hc-05-bluetooth-module-master-slave-commands/