Sleep Tracking Alarm

ECE 445 Design Document

Team 43: Elliot Salvaggio, Kishan Surti, Rutu Patel

TA: Shuai Tang

4/17/2020

1 Introduction

1.1 Objective

For some people, waking up in the morning is not a difficult task. They are ready to hop out of bed and start their day with little difficulty. For everybody else, being motivated to get out of bed at a good time without feeling groggy is not always easy [1]. Every night, people set their alarms on their phones to wake them up the next morning. Whether it's one single alarm or ten different alarms every ten minutes, the basic phone alarm is often not enough to wake up and to get out of bed. After a few nights it becomes a routine to turn off the alarm without even noticing and falling right back to sleep. It is clear that for some the basic alarm is not enough.

Our solution is a new type of alarm system that can get the user out of bed as well as make them wake up feeling energized. Our design is a wearable that goes on a person's wrist, paired with a camera system. The wearable system includes sensors to track sleep patterns which are used to determine best time to set off the alarm. It has an LED screen with buttons used to set a prefered wake up time interval. This is paired with the computer vision camera system. Once the alarm goes off, the camera system waits until it can confirm the person is standing in an upright position before turning off the alarm. This forces the user to get out of bed without turning off the alarm and falling back asleep. The wearable detects the user's body movement while they are sleeping using various sensors to determine whether they are in deep sleep (little to no movement) or in light sleep. We use this information to wake them up while they are in their lightest sleep during the preset wake up time interval, which is anywhere from 15 minutes to 60 minutes, in 15 minute intervals. Let's say a person wants to wake up by 8 am to be at work by 9. If they set their alarm interval to 30 minutes, the system will wake them up at 7:30 am if it

finds they are in very light sleep, before they fall back into deeper sleep. Although they get a little less sleep, they will actually wake up feeling more refreshed by waking up at the end of a sleep cycle. This further motivates them to get up out of bed and stand up straight when the alarm starts ringing.

1.2 Background

According to a study done in the United States, 58% of people reported they spend more than five minutes in bed after turning off their alarm in the morning. Additionally, 57% say they still feel tired after waking up and only 33% describe their experience waking up as good [1]. Clearly, waking up is difficult for many people, and many wake up feeling tired or groggy. Turning off a phone alarm on the phone or hitting snooze is so easy. This does not force a person to get up and out of bed, it merely wakes them up for a small amount of time.

Many people wake up feeling tired due to waking up in the middle of deeper sleep cycles. Humans go through different sleep cycles while they sleep which range from very light sleep to deep sleep. Sleep cycles last around 90 minutes, as we drift from light sleep, into deep sleep, and back to light sleep. During lighter sleep people are relaxed but still restless, as they fall deeper into REM sleep, the body moves very little and heart rate is mostly slow [2]. Waking up in the middle of deeper sleep is difficult and disorienting [3]. The goal is to wake up the person by finding an optimal time; that is when the person is in light sleep according to their sleep trends. If we can wake up people when we know they are at the end of a sleep cycle, rather than at a set time, we believe people will wake up feeling more motivated to start their day.

1.3 Visual Aid

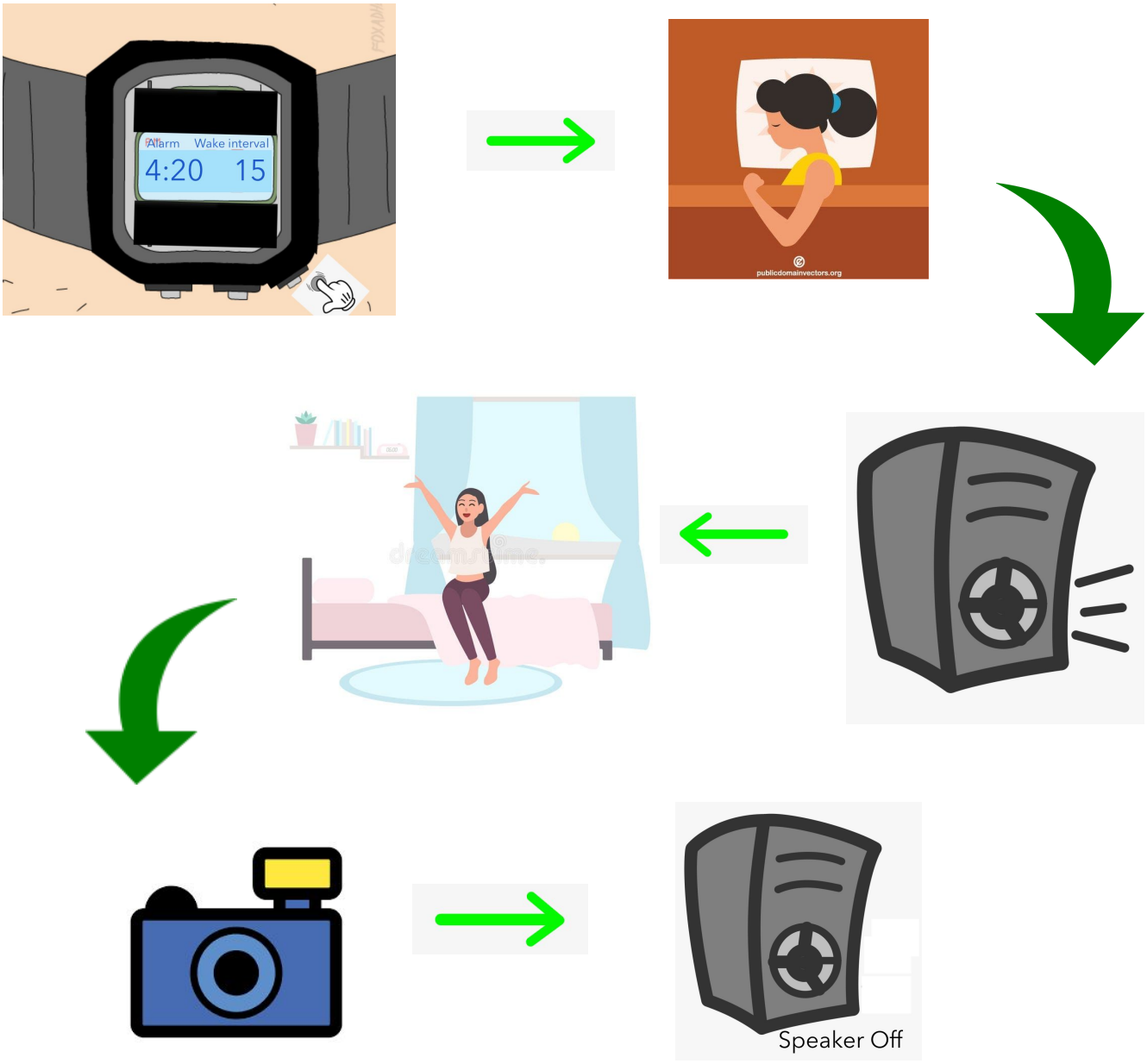


Figure 1: Visual Representation of a use case scenario

1.4 High-Level Requirements

- Able to detect a heart rate between 50 - 120 bpm (0.83 - 2 Hz) +/- 5 bpm with Pulse Oximeter sensor, +/- 3g of force in the x, y, and z dimensions through the accelerometer, as well as recognize sounds within the range of 50-65 dB with the microphone.
- Able to take the user's set alarm time and wake up interval from the wearable device, and find an optimal time to wake up the user based on sleep trends, with exact precision of hour and minute.
- Able to function for at least 8 hours after the battery is charged fully.

2 Design

2.1 Block Diagram

There are two main parts to the design of our system. The most important being the wrist-wearable device, consisting of various sensors, an LCD screen, push buttons, and the microcontroller. Secondly, we have a computer vision module to study the user's posture when the alarm is triggered to confirm that the user is out of the bed.

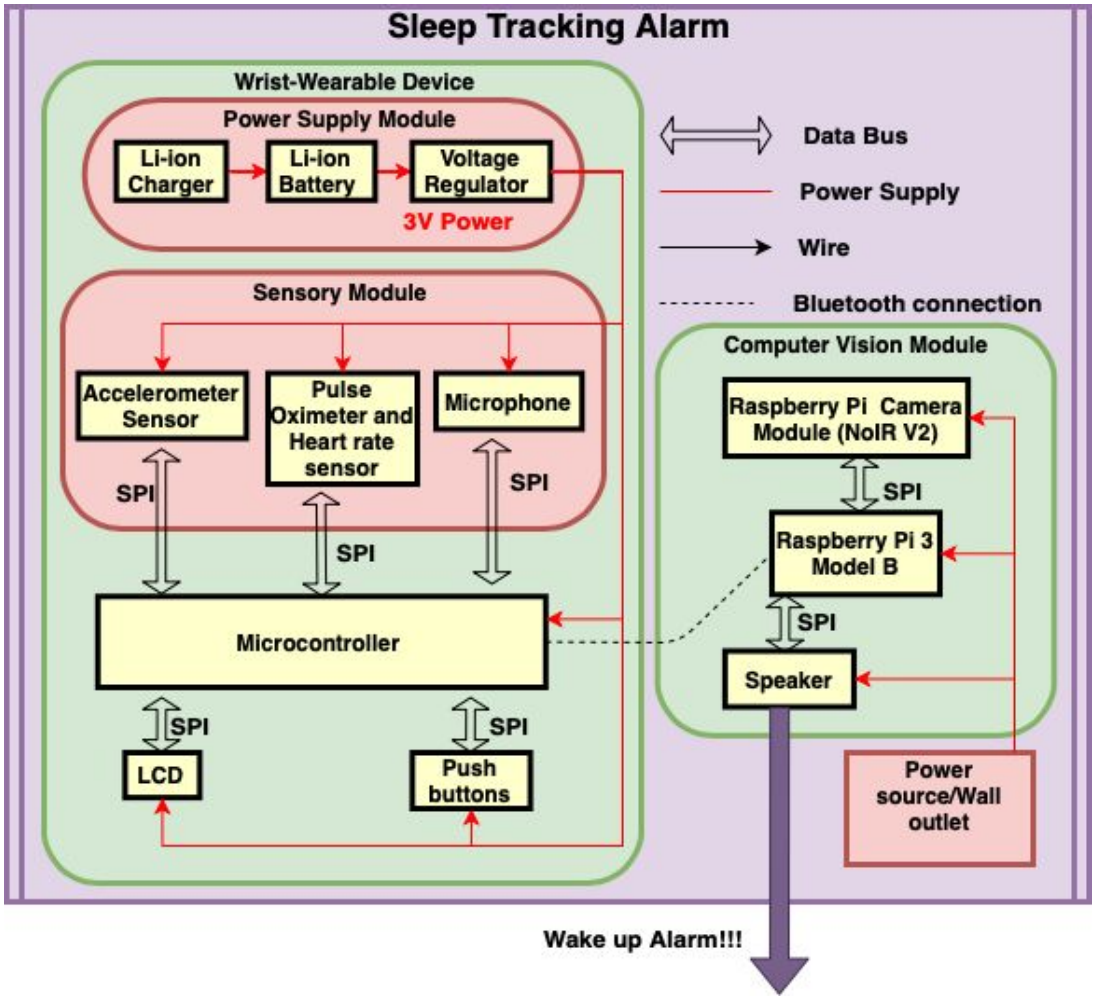


Figure 2. Block Diagram of Sleep Tracking Alarm, showing component level interfaces.

2.2 Physical Design

Figure 3 shows the physical design of the proposed sleep tracking alarm, consisting of the wrist-wearable watch and computer vision module. It shows an ideal setup of how the system is intended to be used. For example, the computer vision module is about ~5 feet away from the bed, so the camera can study the user's whole body posture. The wearable device should be worn by the user while sleeping so the sensors can analyze movements and track the user's sleep.

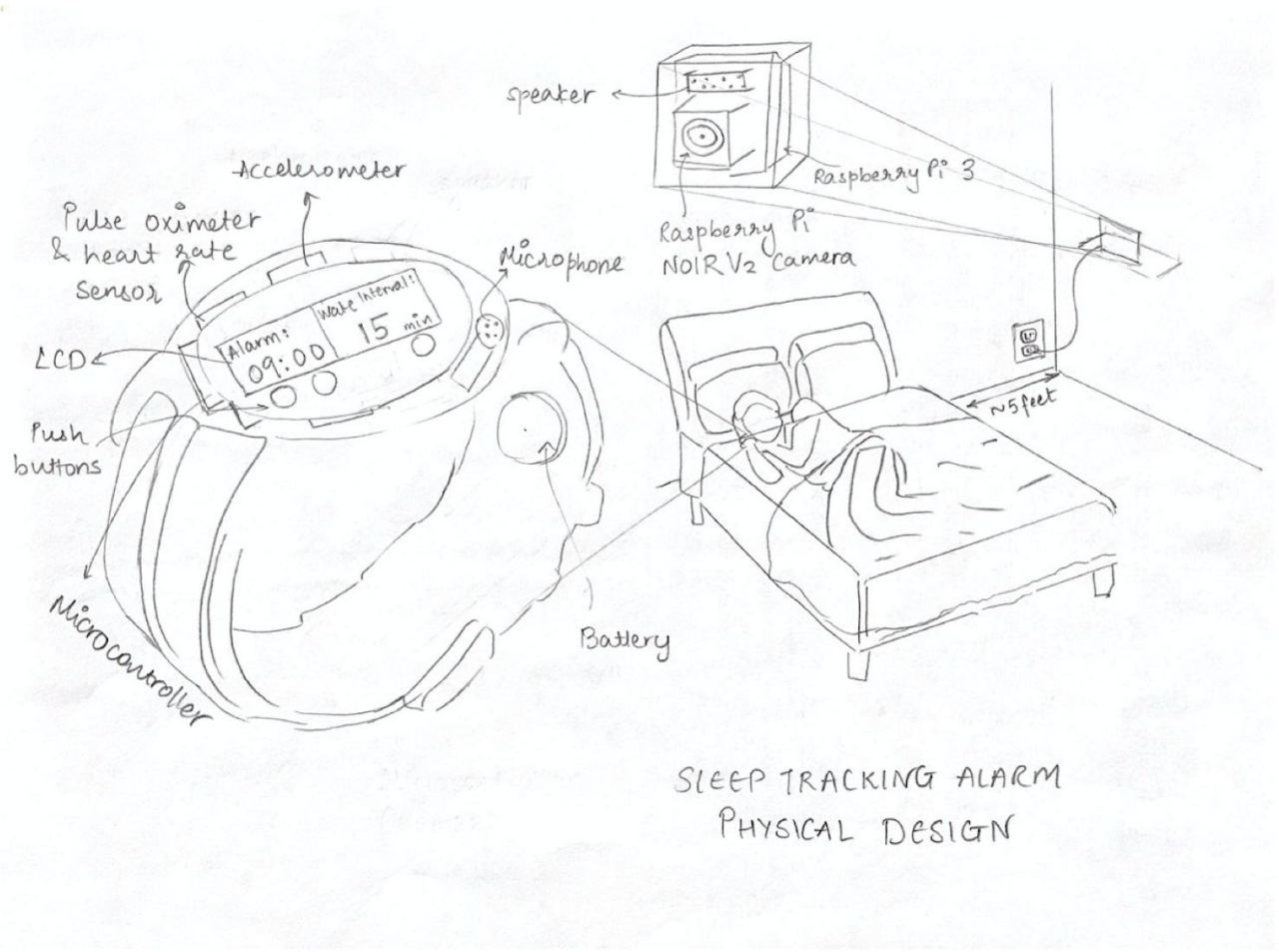


Figure 3: Physical Design of Sleep Tracking Alarm; highlighting the Wearable device and Computer Vision module.

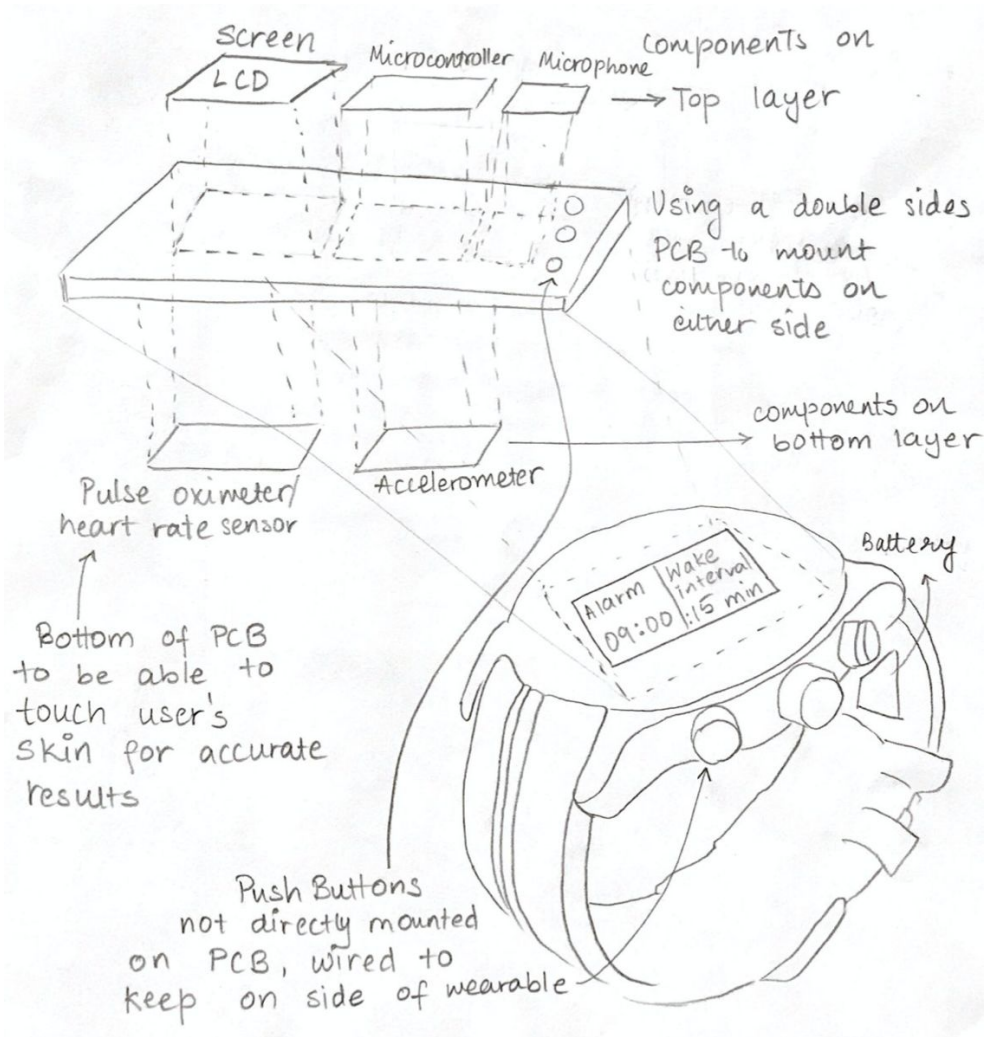


Figure 4: Layered representation of wearable to see components arrangement from cross-sectional view

Components vs Dimension	Length	Width
Pulse Oximeter/Heart rate sensor	25.4 mm	12.7 mm
Accelerometer	3 mm	3 mm
Microphone	9.7 mm diameter	4.5 mm height
LCD screen	60.7 mm	33.85 mm

Considering double sided PCB, minimum dimension : Length = 63.7 mm, Width = 36.85 mm

2.3 Wrist-Wearable Device

The Wrist-wearable Device is a system consisting of multiple hardware components. It includes two sensors; a Pulse Oximeter/Heart rate sensor and an Accelerometer, a microphone, a speaker, an LCD display, two pushbuttons, a microcontroller, and lastly, the power supply for each component. This device is intended to be worn by the user while sleeping to collect sensory sleep data, which is used to decide an optimal time to wake up the user based on their set interval, that is, during a period of light sleep. It will also communicate with the Computer Vision module (Raspberry Pi) to disable the alarm. Just like breathing, heart rate and blood pressure vary during sleep. During a non-REM sleep, heart rate and blood pressure go down and are steadier. However, during a REM sleep, they rise higher. Various sensory inputs are used in the wrist-wearable device to develop a more accurate picture of a user's sleep cycle.

2.3.1 Power Supply

2.3.1.1 Li-ion Battery

We will have one rechargeable 3V Li-ion battery that is used to supply power to each component of the project. Li-ion batteries are a great option for wearables because they are extremely light weight and slim so it would be ideal for our project. We have chosen to go with a 1000 mAh battery to supply the wearable device components.

Requirements	Verifications
The Li-ion battery must be able to power the system at 3-3.5V.	● Connect a fully charged Li-ion Battery to a constant-current test circuit.

	● Use a voltmeter across the battery to ensure that the voltage is within 3V-3.5V range.

2.3.1.2 Voltage Regulator

We will use a voltage regulator to regulate the voltage that is sent to different components such as pulse oximeter/heart rate sensor, accelerometer, microphone, speaker, and microcontroller.

Requirements	Verifications
Voltage Regulator must be able to provide 2.2V-3.6V +/- 5% for ESP32 microcontroller.	● Connect a voltage regulator in between the battery and the microcontroller. ● Use an oscilloscope to measure the open-circuit voltage where we would place the microcontroller. ● Ensure that the voltage and current gets in the correct range 2.2V-3.6V +/- 5% by inserting a resistor with the proper resistance value.

2.3.2 Sensory Module

Heart rate values during sleep vary from person to person, and can range from normally from 40-100 bpm. Many other factors can affect a person's sleeping habits as well. For the sake of this project, we will use normal and healthy baseline values to look for in determining sleep cycles. Based on an existing sleep tracking application, Oura [4], the most common nighttime heart rate is around 55 bpm. The lowest values during the night can be seen within the range of 35-84 bpm as well [4]. The table below describes the trends we are typically looking for in the different sleep stages.

Sensors Vs Sleep stages	Stage 1: Light Sleep	Stage 2: Normal sleep (similar to light)	Stage 3: Deep Sleep	Stage 4: REM Sleep (Dream state)
Accelerometer	Between +/- 3g of force (some usually movement, maybe turning sides or adjustment)	Between +/- 2.5g of force (some usually movement, maybe turning sides or adjustment)	Between +/- 0.5 g of force (little to no movement)	Between +/- 2 g of force (mostly no movement, can move due to dreams, however)
Pulse Oximeter	50-90 bpm (mostly in the middle, less than a resting heartbeat)	50-70 bpm (heart rate begins to lower a bit)	40-70 bpm (mostly on lower end)	50-100 bpm (heart rate varies most here because of dreaming)
Microphone	Little to no sound	Little to no sound	Potential sound (snoring)	Little to no sound

Table 1: Shows Sensory reading vs Sleep stages

2.3.2.1 Pulse Oximeter/Heart Rate Monitor

Many heart rate detection devices today involve the use of photoplethysmography, which is the measurement of blood volume changes in the microvascular bed of tissue [5]. Pulse Oximeters are an integral part of this and are widely used to determine how much light is absorbed by the skin, giving a good indicator how much blood is being pumped. In other words, it employs LED light through the blood measuring the amount of oxygen saturation levels, and thereby producing a graph for determining heart rate. The Apple Watch and many other wearable devices use the same technology today. To implement this, we plan to use a low power Sparkfun Pulse Oximeter and Heart Rate Sensor (MAX30101 & MAX32664). Once the Pulse Oximeter has collected data from its sensor, it is then sent to the Heart Rate Sensor, which performs algorithms to determine blood oxygen saturation and, of course, heart rate. It works best for our project because not only is the component relatively small in size, but it also provides important information regarding the sensor's confidence percentage in its reporting, which makes it more reliable.

Requirements	Verifications
Be able to detect heart rate between 50 - 120 bpm (0.83 - 2 Hz) +/- 5 bpm	● Measure resting heart rate manually through pulse and keep this data. ● Write code in Arduino to obtain output data and print out heart rate and confidence percentage values. ● Hook up the chip to an Arduino board and place finger on the sensor firmly (can use a rubber band to hold in place)

	● Compare output values to the heart rate measured manually and determine if it successfully read within 50-120 bpm.

2.3.2.2 Accelerometer

An accelerometer is the most common component in not only wrist-worn applications, but also cell phones and many other devices today. It is able to measure changes in gravity, and outputs an AC signal for each dimension for even more precision. For the scope of our project, it will be used to analyze movements in the user's sleep. We plan to use the Triple Axis Accelerometer Breakout ADXL337 from Sparkfun for our wrist-wearable device. It does not draw a ton of power since it has a typical current of 300μA, and its size is less than a U.S. quarter, ideal for a wearable.

Requirements	Verifications
Be able to detect changes in forces from +/- 3g in the x, y and z-dimensions.	● Hook the chip up to a breadboard, supply it with 3.3 V and connect each of the output pins to a DMM one at a time. ● Begin to move the chip around the air so it can detect a change in gravity. ● Check the DMM to ensure that it has output values of [1.6 V, 3.3 V] or [-3.3V, -1.6V]

2.3.2.3 Microphone

The microphone is used as another one of the sensory inputs to track a user's sleep. The purpose of including this is to determine whether the user is producing any sounds during their sleep, mostly to detect snoring. We've decided to go with the SparkFun Electret Microphone Breakout, which has the capability to amplify the sounds of voices, claps, door knocks, and for our case, snoring. We would utilize the ADC converter on ESP32 microcontroller to make meaningful use of the output that comes from this microphone.

Requirements	Verifications
Be able to detect sounds at a range of at least 50-65 dB when supplied with specified voltage of 2.7V-5.5V.	● Connect microphone to an Arduino board with a supplied 3.3V, and its output to an LED. ● Produce normal sounds within a 50-65 dB range (such as clapping, knocking, snoring, or normal talking level) and observe if the LED lights up

2.3.3 Microcontroller

Microcontroller acts as a logical component; the brain of our project. We plan to use an ESP32 microcontroller. It is responsible for taking all the sensory inputs such as Pulse Oximeter sensor, Accelerometer, and microphone, processing inputs to get an average for higher accuracy, and finding an optimal wake up time. It is responsible to enable the alarm; trigger the speaker to

wake up the user. Additionally, it communicates with the computer vision module, more specifically Raspberry Pi via inbuilt Bluetooth to enable the alarm. The computer module later would study the user's body posture and disable the alarm when the user is out of bed standing. A main aspect of the project is to be able to find an optimal time to wake up users in their lightest sleep. For that detail, we rely on these sensors to study the user's sleep trend (Table 1 describes the ground truths of each sensory reading with its associated sleep stage). Microcontroller is responsible for taking in data from all these sensors, and we will program it to combine each sensory input to give us the user's current sleep stage. Continuous monitoring will allow us to see increase or decrease in user's sleep stages giving us a graph of sleep trends, which would in turn allow us to find the next cycle and wake up time.

Requirements	**Verifications**
Microcontroller must be able to take sensory inputs and trigger alarm at an optimal time with time precision of hour and minute.	● Connect the microcontroller to all the sensors; accelerometer, pulse oximeter, and microphone . ● Program the microcontroller to take an average of readings from the ground truth shown in *Table 1 for Sensory inputs vs Sleep stage*. ● Program the microcontroller to find the next sleep cycle of 90 minute close to wake up time and in wake up interval ● Ensure manually if the microcontroller logic found the correct optimal time as

	desired.
Microcontroller must be able to communicate with the Computer Vision Module (precisely Raspberry Pi) using wireless Bluetooth communication protocol in less than 20 seconds.	• Send any dummy data to the Pi, from microcontroller via Bluetooth • Check the data received on the Pi to see if the data matched with sent information. • Ensure that the dummy data is verified as expected in the time range.

2.3.4 Push buttons

Push buttons are a way for letting users add their desired wake up time on the wearable device itself. We plan to use three Mini Pushbutton Switches COM-00097. It's a miniature single pole single throw switch, which is good for clicks on a wearable. Two switches are used to set up alarm time, specifically, one for hour precision and other for minute precision. Third switch is used for wake up interval time, with 15 minute increment, considering the 90 minute sleep cycle.

Requirements	**Verifications**
Can be momentarily clicked, incrementing the interval time by 15 minutes when rated up to 50mA current.	• On a breadboard, connect the push button to an LED circuit, with one pin to specified power supply, second to ground and third to a digital I/O pin of Raspberry Pi (pin 7). • Examine from the third pin, while button open (unpressed), the pin is connected to power, so we read High. • Similarly, when the button is closed

	(pressed), the pin is connected to ground, so we read Low. ● Program microcontroller for a logic where each button press/unpress increments the minute timer by 15 minutes, and resets to 0, reaching 90 minutes.

2.3.5 LCD Display

LCD screen is attached to the wrist-wearable device for users to display their set alarm time, and wake up time interval, which the user feeds in through the push buttons. We plan to use a small I2C LCD which is ideal for displaying text and numbers, and is easily paired with ESP32 microcontroller to support the logic from push buttons to be displayed accurately.

Requirement	Verification
Must be able to display numbers precisely when a push button is pressed.	● Connect your laptop to ESP32 with LCD directly to its GPIO pins. ● The I2C address should be displayed on the serial monitor. Find the LCD I2C address. ● With the LCD properly wired to the ESP32, upload the I2C Scanner sketch. ● Select where to display characters on screen, and simply send a "Hello" for testing.

2.4 Computer Vision Module

After triggering the alarm from the wrist-wearable device, we want to make sure the user is actually out of the bed. To accomplish that task, we designed the computer vision module, ideally placed approximately 5 feet from the user's bed to record the user's body posture and confirm the user stood out of the bed.

2.4.1 Raspberry Pi 3 Model B

Raspberry Pi 3 is intended for high performance to run computer vision algorithms on the video captured by its attached camera. Once the alarm is triggered, the microcontroller from the wrist-wearable device communicates with Raspberry Pi 3 using the Bluetooth. Raspberry Pi 3 triggers the attached Raspberry Pi Camera Module NoIR V2 to start recording the user. Recording is used to study the user's body posture to confirm if the user is sleeping, standing or sitting. We conduct the study by using an open source library called 2D Pose Estimation and Action Recognition. Once the user is out of bed, we will use the built-in Bluetooth module of the Raspberry Pi 3 to send that information to the microcontroller in the wearable device, which thereafter would disable the alarm; accomplishing the final task of the device.

Requirements	Verification
Be able to trigger the Raspberry Pi Camera NoIR V2.	● Connect the Raspberry Pi 3 along with the Camera Module NoIR V2 to any laptop.

	● Confirm that Raspberry Pi 3 triggers the Camera Module V2, after keypress from laptop [6] ● Confirm the camera takes a picture in any picture formats stored on the laptop.
Must be able to perform 2D Pose Estimation accurately stating the person is sitting, sleeping or standing.	● Visually confirm if the output from the open source 2D Pose Estimation algorithm accurately judges the person's posture.

2.4.2 Raspberry Pi Camera Module NoIR V2

We plan to attach a Raspberry Pi Camera Module NoIR V2 which gives us an advantage to capture the user's posture even in the dark with infrared lighting. Raspberry Pi 3 triggers the camera to start recording, as well as stop recording. The camera will be attached on the front side of the computer vision module system.

Requirements	Verification
Be able to record surroundings when notified by the Raspberry Pi 3 that is good enough quality of an image that it is properly analyzed by the 2D Pose Estimation software.	● Connect the Raspberry Pi 3 along with the Camera Module NoIR V2 to any laptop using a USB cable. ● Run the Raspberry Pi script for taking a picture via camera module. ● Visually confirm the camera takes a picture of good quality stored on the laptop.

2.4.3 Speaker

Speaker is used for the alarm sound to wake up the user. We plan to use Sparkfun Thin Speaker - 0.5W, 8Ohm as it's only 40mm in diameter, and 4mm thick, and perfect to be placed on with the Raspberry Pi 3.

Requirement	Verification
Be able to output alarm sound when supplied with input voltage of ~1.5V.	• Hook up the speaker to the Arduino board and add it to a 1.5 V power supply. • Add Arduino code to take in an input on the board and have the speaker output a loud sound each time the input is triggered

2.5 Schematics

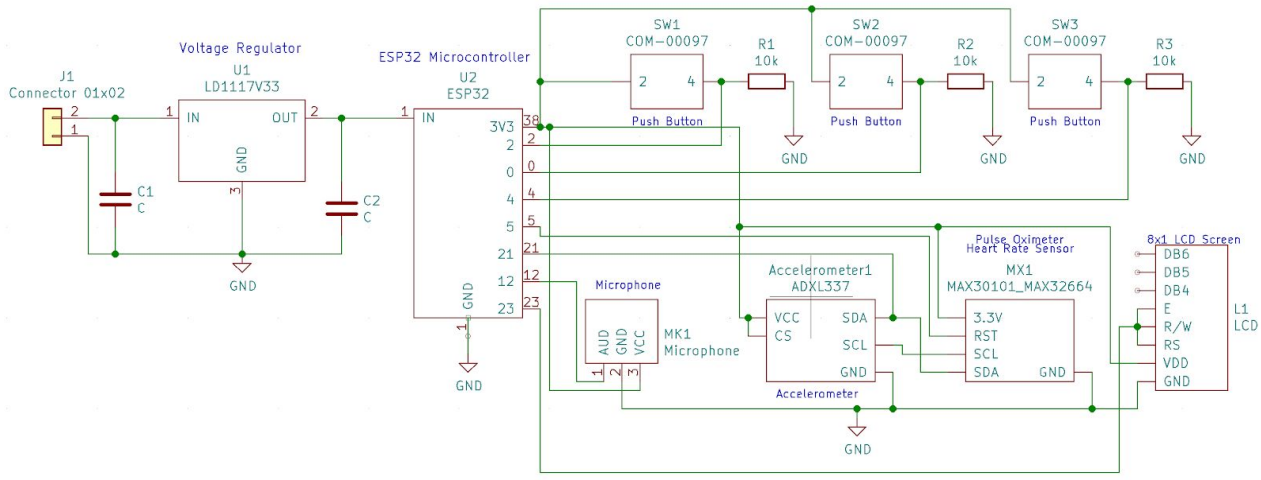


Figure 5: Circuit Diagram of the wearable

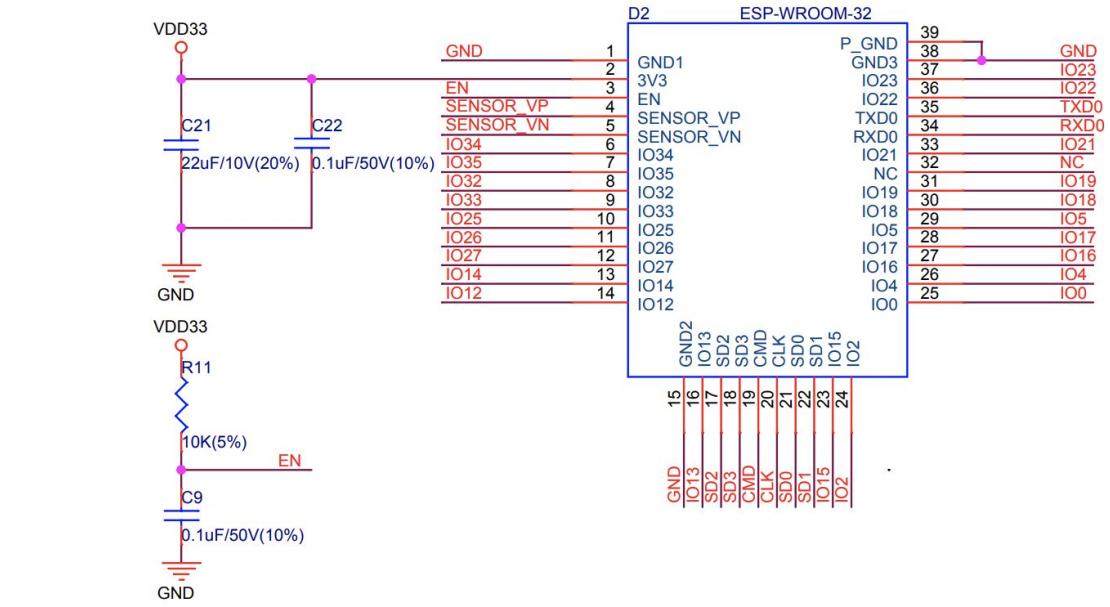


Figure 6: Espressif ESP32 schematic

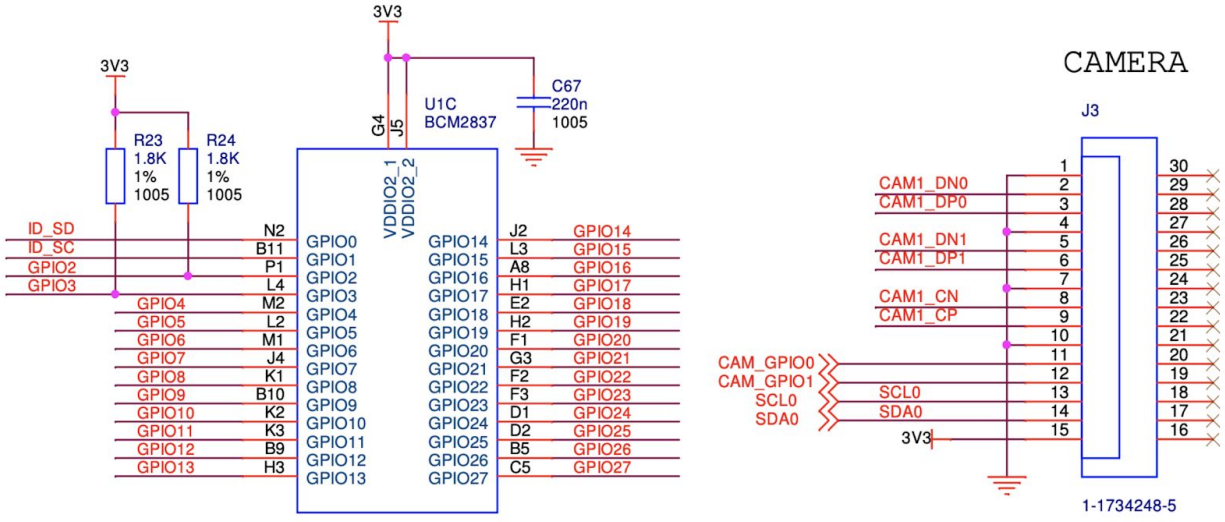


Figure 7: Detailed Raspberry Pi 3 B and Raspberry Pi Camera Module NoIR V2

2.6 Tolerance Analysis

The portion of our project that poses the biggest risk to completion would be our Wearable Device module. This is the most important piece of the project because this is where all the sensory inputs are at, and it will be the one to determine when to wake the user up from their sleep. It also is the most complex, since we will need to take in data from these various inputs and be able to recognize what stage of sleep the user is in. This will provide a higher level of accuracy than normal sleeping tracking devices, since we have additional inputs than the existing devices. There will be several requirements from this module that must be addressed for it to function properly in the way we need it to: 1) The Pulse Oximeter and Heart Rate Monitor must be able to detect a heart rate between 50 - 120 bpm (0.83 - 2 Hz) +/- 5 bpm, 2) the Accelerometer needs to be able to measure +/- 3g of force in the x, y, and z dimensions, 3) the Microphone needs to to pick up sounds within a range of 50-65 dB, 4) the battery must be able supply power to all components for at least 8 hours.

Our first requirement will be taken care of by two chips that are integrated with the Pulse Oximeter: the MAX30101 and MAX32664 chips. The former does all the sensing by using its internal LEDs to bounce light off the skin's arteries and measuring how much light is absorbed from the use of photodetectors. This process is called *photoplethysmography*, which will be

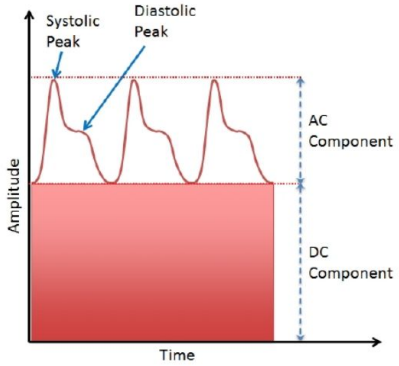


Figure 8. Example photoplethysmograph

explained more down below. The latter of the two chips receives data from the other and applies algorithms to determine both the heart rate and blood oxygen saturation.

As mentioned before, the way in which this particular sensing will work is known as *photoplethysmography*. It creates a photoplethysmograph (PPG) that helps to detect blood volume changes in the microvascular bed of tissue [7]. The figure above shows what a simple PPG signal would look like, and consists of both an AC and DC component. The AC signal is superimposed on top of the DC, and this specifically shows the changes in arterial blood volume, which can then be utilized to determine heart rate [7]. As shown in Figure 8, we want to measure the peak-to-peak interval between two Systolic peaks. Once this value is obtained, we can calculate two different forms of a heart rate: the instantaneous heart rate (HR_{inst}) and the mean heart rate (HR_{med}). For the first equation, $\boldsymbol{t_1}$ represents that peak-to-peak interval time from before. For our second equation, $\boldsymbol{Q_{nn}}$ corresponds to the amount of normal intervals (***NN***) in the time frame of [$\boldsymbol{T_i}$, $\boldsymbol{T_f}$] [8]. Here, we would most likely choose to go with determining a mean heart rate, since this can provide us with a more accurate reading and give us a tolerance of +/- 5 bpm.

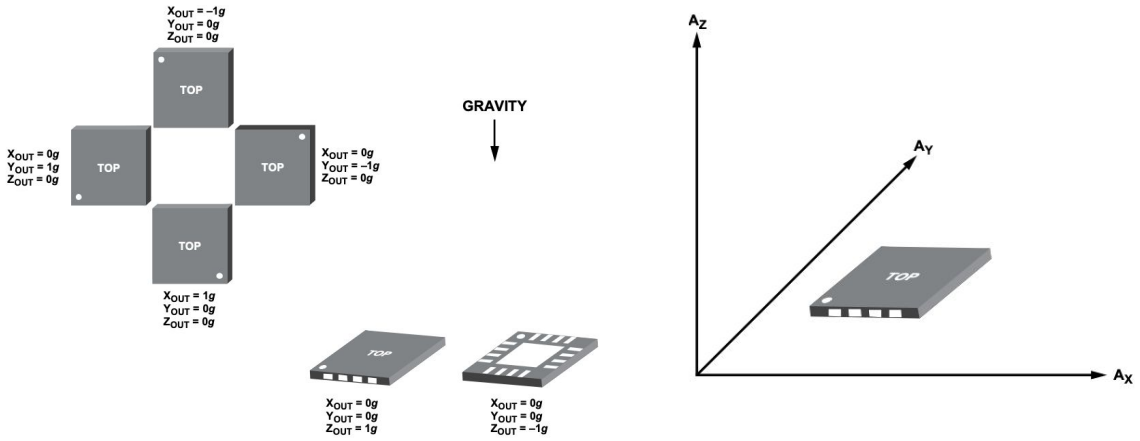
$$HR_{inst} = \frac{60}{t_1}$$

$$HR_{med} = \frac{1}{Q_{nn}} \sum_{k \in [T_i, T_f]} NN\,[k]$$

Figure 9: Heart Rate Value (HRV) Equations

Our second requirement is that the accelerometer needs to measure +/- 3g of force in x, y, and z dimensions. This is important for when we want to recognize any small vibrations or movements when a person is sleeping. For this reason, we have decided to go with the ADXL337 accelerometer, which is excellent for detecting small movements. It is ideally

Figure 10: Accelerometer layouts



powered with a voltage supply of 3 V, or it can be powered with a minimum of 1.8 V or a maximum of 3.6 V. There are outputs for each of the axes, and these are measured with an analog-to-digital converter, which correlates to the acceleration in that given axis. After reviewing its specifications, it has been noted that the output is also ratiometric, meaning that its sensitivity varies directly to the supply voltage itself. At a $V_S = 3$ V, the output sensitivity is approximately 305 mV/g.

In addition to this, capacitors must be placed at the output supply pins in order to implement low-pass filtering for antialiasing and noise reduction. This will help with the accuracy of measuring those small movements. Down below are the potential capacitor choices and the corresponding bandwidth. The selected bandwidth will decide the measurable

Bandwidth (Hz)	Capacitor (µF)
1	4.7
10	0.47
50	0.10
100	0.05
200	0.027
500	0.01

Figure 11. Capacitor selections for corresponding bandwidth

resolutions, or the smallest detectable acceleration, but the output usually has a typical bandwidth of greater than 500 Hz. For this project, a 0.1 µF capacitor should suffice.

The third requirement is that the microphone needs to be able to pick up sound within a range of 50-65 dB, which is typically what snoring falls under. The reason why this is important is that we differentiate our product through the use of a microphone, and determine whether a person is snoring or not actually helps to determine their sleep stage. Nonetheless, the one we have chosen to implement is the SparkFun Electret Microphone Breakout. This operates within a frequency range of 100 - 10kHz and has a sensitivity of about -46 +/- 2 dB. This is pretty good for an average speaker, and since it is integrated into the Wearable Module, it should have no issue picking up normal sounds in our desired range.

Lastly, the battery must be able to supply enough power to all the components in the wearable for at least 8 hours after a full charge. This is a big challenge because the battery must fit compactly inside the wearable, which means it needs to be smaller and will result in less capacity. For our design, we were able to go with a 1000 mAh rechargeable battery to power the whole wearable. The component that arguably draws the most power would be the ESP32. When the total functionality of the microcontroller is active, including WiFi, Bluetooth, and radio, it can draw current anywhere from 200-750 mA depending on how much processing it is doing. If

we kept it running like this, there would be no way that it can last 8 hours. However, in order to save battery life, we will put the ESP32 into “Modem Sleep” overnight, which will turn off the extra functionality like WiFi and Bluetooth, and turn these back on in the morning in order to transmit a signal to the Computer Vision module. This would result in the ESP32 only drawing a current of 3-20 mA. Taking this into account, along with all the other components in the wearable, we can now calculate the expected battery life:

$$\underline{\textit{Battery Life} = \textit{Battery Capacity (mAh)} / \textit{Load Current (mA)}}$$

$$\textit{Battery Capacity} = 1000\ mAh$$

$$\textit{Load Current} = 5\ mA + 12\ mA + 300\ \mu A + 0.5\ mA + 12\ mA * 3 + 23\ mA + 40\ mA + 600\ \mu A$$

$$\textit{Battery Life} = 1000\ mAh / 117.4\ mA \approx \mathbf{8.52\ hours}$$

After this calculation, we can see that the battery is expected to last at least 8 hours, which satisfies our above requirement.

3 Project Differences

3.1 Overview

The original project from Fall 2019 was a problem-of-the-day based alarm. Though their solution was much different from ours, the problem they were focusing on fixing was, at its core, the same as ours: waking up on time in the morning can be very difficult and there must be a better solution than a basic alarm. Their solution was to quiz the user by using preset complex questions that the user had to answer correctly in order to turn off the alarm system. This system consisted of a speaker, a real time clock module, an LCD display, physical buttons, two microcontrollers, a bluetooth module, and an Android phone application. To use the alarm, the user must input several multiple choice questions through their mobile app that they will have to answer correctly when they wake up.

Our solution is similar in that we also have an alarm that is more complex than a basic alarm to wake the user in the morning, and that the user cannot turn the alarm off until they meet certain criteria that confirms they are awake. In the case of the problem of the day based alarm, they must answer multiple choice questions. In our solution, the user must stand up straight facing the camera system to verify the user is actually out of bed to turn off the alarm. Our system is an improvement over the original group's design for multiple reasons. For the main feature, our solution studies the user's sleep through various sensors to wake the user at a time when they are in light sleep, which means, the need for activating the brain through asking questions is avoided. Even more, with our design the user does not need to manually enter new questions every day to change what is being asked. Our system works every morning the same way requiring no manual input from the user. Additionally, with the original solution, if the user

becomes used to answering the same questions every morning, turning off the alarm becomes significantly easier. The user also already knows the answers to these questions, as they put the questions in themself. It is likely not very difficult to answer questions you know the answers to. Our solution requires confirming they are actually out of their bed and ready to start their day.

3.2 Analysis

A major difference between the two solutions is in how we both wake the user. The core problem we are trying to solve is waking a very tired person enough so they are motivated to get up and out of bed. Both solutions attempt to solve this problem by slowly waking up the user in ways more than loud sound coming from an alarm. In the case of the problem based alarm, their solution slowly wakes the user by stimulating brain activity through answering difficult questions. In our solution, we wake the user while they are in their lightest sleep. Questions can help wake a person significantly. Studies show that “a one minute talk to a stranger wakes up the brain with a 99 percent guarantee” and that when “someone asks you a question in the morning your brain has to wake up to answer” [9]. The POTD based solution tries to replicate this effect by asking the user questions as soon as they wake up. The problem with their solution is that the questions are submitted by the user themselves. A better solution would be to have questions generated from a source from the internet daily to randomize the questions automatically for the user. Our solution focuses on the root of the problem, which is that it is difficult to wake up during deep sleep. The POTD solution does nothing to solve this problem. If a user is waking up in deep REM sleep they will still feel exhausted even after answering questions. Our solution wakes the user during light sleep, which is how the body wakes up naturally. The change from

wakefulness to sleep and sleep to wakefulness occurs during stage 1 non-REM sleep [10]. Our solution wakes the user during this stage 1 sleep, giving the user the feeling of naturally woken up without an alarm. The POTD solution may wake up users during deep sleep making it very difficult to get out of bed, even after reluctantly answering multiple choice questions they wrote for themself the night before. Our solution is an improvement toward fixing this problem.

4 Cost and Schedule

4.1 Cost Analysis

4.1.1 Labor Cost

We are a team of three, all Computer Engineering students. The average starting salary for an Illinois Undergraduate Computer Engineering graduate was $96,518 in 2016. This comes out $96,518/52 weeks = $1,856.11 per week. $1,856.11/40 hours = $46.40/hour. There are about 10 week left of the semester. If we all plan to work 15 hours a week on the project, that puts the total labor cost to (46.40*15*8*3*2.5) = $41,760.

4.1.2 Parts Cost

Part Name	Qty.	Part Description	Cost
Raspberry Pi 3 Model B	1	Computer Vision processing device	$35.00
Raspberry Pi Camera Module NoIR V2	1	Take picture of user's posture	$24.99
Rechargeable Lithium-Ion Battery	1	Power supply for wearable	$9.95
Espressif ESP32	1	Microcontroller	$4.50
SparkFun Triple Axis Accelerometer Breakout - ADXL337	1	Used to sense user's slight body movements	$9.95
SparkFun Electret Microphone Breakout	1	Used to sense user's slight body movements	$6.95
Mini Pushbutton Switches COM-00097	3	Push button used to set the time and interval	$0.35
Newhaven Display NHD-0108HZ-FSW-GBW	1	8x1 LCD display on wearable to display set time and interval	$8.50
SparkFun Pulse Oximeter and Heart Rate Sensor - MAX30101 & MAX32664	1	Pulse Oximeter and Heart Rate sensor used to determine depth of user's sleep	$39.95
Sparkfun Thin Speaker	1	To wake user from sleep	$0.95
Digikey 3.3 V 800 mA Regulator - LD1117-3.3	1	Brings voltage from 5V power source down to 3.3V for ESP32	$0.55

Total: $141.64

4.2 Schedule

Week	Elliot	Kishan	Rutu	Team Goal
Week 1	Buy project parts	Create test cases and check circuit schematic prone of bugs	Make initial conversation with ECE shop to put our system in box	Create PCB design and correct schematics if needed
Week 2	Program microcontroller	Test design	Finalize PCB for Early bird PCB order	Finalize PCB design, order parts and send in design
Week 3	Continue programming microcontroller	Look into 2D key point algorithm for image recognition	Implement CV algorithm to work with Pi and camera	Finish CV module side of project
Week 4	Continue working on data transmission protocols	Continue working on data transmission protocols	Test sensor readings individually	Implement wireless bluetooth communication from Pi to microcontroller
Week 5	Implement sensor reading logic in microcontroller	Put sensors to work together	Put sensors to work together	Connect sensors applications to PCB design
Week 6	Debug control component	Create test cases to find out bugs in sensory readings data	Test end to end flow of the system	Begin debugging overall design, creating test cases
Week 7	Create ways to test overall design	Modify any bugs found	Fix any bugs or errors and finalize product	Finalize and fix bugs/flaw in design, continue testing
Week 8	Prepare for demo	Prepare for demo	Prepare for demo	Final touches on design, prepare for demo, start up final report for project

5 Ethics and Safety

One ethical concern during the development of the project will be to make the wearable both compact and comfortable for the user. To address this, we will need to find the smallest hardware components that we can and try to arrange them in a manner that will keep it from being bulky. There is also a safety concern that comes with a wearable, such as any shocks or electrocutions. To prevent this, we cannot have any possible openings in the hardware that could harm a person while wearing our device. In addition, there may be a possibility of the wearable heating up while it is touching the skin. For this concern, we will have to make sure that we are supplying the proper power to all components, and that certain components will be limited in power when they are not in high use. These safety measures all fall under the IEEE Code of Ethics #1.

Another potential concern that could arise is maintaining a user's privacy. We record a user's sleep cycle and have a computer vision component that looks to see if the user has gotten up and out of their bed. Both of these will accumulate data in some manner, but we do not retain any information on our end. We will not keep the user's sleeping data over time, as it is only used on a nightly basis, and will be overwritten with new data each night. The camera from our Computer Vision module will not be active overnight or during the daytime, and it will include a shutter if the user would like to cover the camera at any point they like. It will only begin detection once it receives a signal from the wearable device that the alarm has been triggered. The user will be informed of these privacy measures, and any images/videos that are taken to detect a person's posture will not be retained either. Both of these cases align with the IEEE

Code of Ethics #1 and #9 about protecting the safety and health of users, as well as avoiding malicious practices.

In regards to any potential breaches, we will need to encrypt data stored in both the Wearable Device and Computer Vision modules. The security of our Computer Vision module will be very crucial since we are examining a person's posture by utilizing a camera. We will not have to worry about breaches from the internet, as the Raspberry Pi will communicate to the wearable via bluetooth and its WiFi chip will not be powered.. As mentioned before, any images that are taken with the camera will be protected and will not be retained in any other way except to detect if a person is up from their bed. We will dispose of this data each time the alarm has been suppressed, since it will no longer be needed.

6 Citations:

[1] S. Roy, “57% of Americans Hit the Snooze Button,” Sleep Review, 28-Aug-2014. [Online]. Available: https://www.sleepreviewmag.com/sleep-health/sleep-whole-body/heart/americans-snooze-button-withings/. [Accessed: 04-Apr-2020].

[2] “3 Things You Should Know About REM Sleep,” Sleep Study, Sleep Clinic | Valley Sleep Center | Arizona, 18-Feb-2020. [Online]. Available: https://valleysleepcenter.com/3-things-you-should-know-about-rem-sleep/. [Accessed: 04-Apr-2020].

[3] S. Davis, “How to Wake Up More Easily,” WebMD, 27-Oct-2009. [Online]. Available: https://www.webmd.com/sleep-disorders/features/trouble-waking-up#1. [Accessed: 04-Apr-2020].

[4] “the most accurate sleep and activity tracker,” Oura Ring. [Online]. Available: https://ouraring.com/heart-rate-while-sleeping. [Accessed: 04-Apr-2020].

[5] J. Allen. Photoplethysmography and its application in clinical physiological measurement. Physiological Measurement, 2007. [Accessed: 18-Apr-2020].

[6] *projects.raspberrypi.org*. [Online]. Available: https://projects.raspberrypi.org/en/projects/getting-started-with-picamera. [Accessed: 18-Apr-2020].

[7] J. L. Moraes, M. X. Rocha, G. G. Vasconcelos, J. E. Vasconcelos Filho, V. H. C. de Albuquerque, and A. R. Alexandria, “Advances in Photopletysmography Signal Analysis for Biomedical Applications,” *Sensors (Basel, Switzerland)*, 09-Jun-2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6022166/. [Accessed: 18-Apr-2020].

[8] K. Shelley and S. Shelley, Pulse Oximeter Waveform: Photoelectric Plethysmography, in Clinical Monitoring, Carol Lake, R. Hines, and C. Blitt, Eds.: W.B. Saunders Company, 2001, pp. 420-428. [Accessed: 18-Apr-2020].

[9] Cotroneo, “How to wake up even the groggiest brain,” *MNN*, 03-Feb-2020. [Online]. Available: https://www.mnn.com/health/fitness-well-being/stories/how-wake-up-brain-morning-groggy. [Accessed: 18-Apr-2020].

[10] “Brain Basics: Understanding Sleep,” *National Institute of Neurological Disorders and Stroke*. [Online]. Available:

https://www.ninds.nih.gov/Disorders/Patient-Caregiver-Education/Understanding-sleep. [Accessed: 18-Apr-2020].