Peaceful Pillow

Team 6

Jacob Knaup, Kevin Lam, Pamela Lombardi, Anthony Spencer

Arizona State University

EGR 304

Dr. Shawn Jordan

Problem Definition

Final Problem Statement:
The purpose of our project is to create a comfortable device to help educate sleepers of their sleeping habits and gently wake them. The device will enable its users to gain valuable information about their sleeping patterns while asleep.

Final Impact Statement:
Our device can help those who have troubles falling asleep or staying asleep by giving them more information about their sleeping patterns. With this information, they can go to a medical professional and seek advice on what actions they should take to solve their sleeping problems.

Criteria:

Criteria	**Threshold** ***(Halt & Review)***	**Target Goal** ***(Plan of Record)***	**Stretch Goal** ***(Design Stop)***	**Rationale**
Cr-1: Variable audio alarm	>40db, <100db	80 db +- 30	80db +-60	Awake user gently (goldilocks)
Cr-2: Measure sleeping	Detect presence/absence of head Sensitivity: +/-10lbs	Detect rolling Sensitivity: +/-5lbs	Detect all movements +/- 0.5lbs	Measure user's soundness of sleeping based on head movement
Cr-3: Battery Life	12 hr battery	24 hr battery	186 hr battery	Stay powered through the night with no cords
Cr-4: Number of pressure sensor collection samples	2 samples	10 samples	100 samples	Track user's sleep cycle
Cr-5: Accuracy of keeping track of time	+/- 1 min	+/- 1 second	+/.1 second	Know how long the user is sleeping and what time it is

Cr-6: Size of entire unit	8” x 10” x 4”	6” x 8” x 3”	6” x 4” x 2”	We don’t want the box to be too big and make the user’s sleep uncomfortable

Constraints

Constraints	**Threshold *(Halt & Review)***	**Target Goal *(Plan of Record)***	**Stretch Goal *(Design Stop)***	**Rationale**
Co-1: Stay cool (components)	<110 degrees F	<90 degrees F	<80 degrees F	Project specific
Co-2: Material, (flame retardant)	<50 degrees F	>/= 100 degrees F	>= 200 Degrees F	Project specific
Co-3: Pillow Size	28”L x 16”W x 6”H	22”L x 14”W x 4”H	18”L x 12”W x 4”H	Project specific
Co-4: Prototype budget	<$160	$80	$40	Defined by professors
Co-5: Power supply (AC adapter, battery, or solar panel connected to voltage regulator. No USB power packs may be used)	>3.3V	5V	12V	Defined by professors
Co-6: Microcontroller (Cypress Bluetooth®	>/= 1	1	2	Defined by professors

Low Energy (BLE) Pioneer Kit (CY8CKIT-042-BLE-A), PSoC® 4 Pioneer Kit (CY8CKIT-042), and/or Prototyping Kit (CY8CKIT-049-42xx))				
Co-7: Sensor(s) read by a microcontroller	6	2	1	Defined by professors
Co-8: Actuator(s) controlled by a microcontroller	1	2	4	Defined by professors
Co-9: Bluetooth Low Energy communications to a phone, computer, or another device using a Cypress BLE Pioneer Kit	1-way communication with max range 5 ft	1-way communication with max range of 10 ft	2-way communication with max range of 10 ft	Defined by professors
Co-10: Custom printed circuit board • Must be created in Cadence • No commercial	1	2	3	Defined by professors

boards				
Co-11: Surface mount components • Size 0805 or larger recommended	0	6	100	Defined by professors
Co-12: Programmed in C or C++	Programmed in C	Programmed in C	Programmed in C	Defined by professors

<u>Project deliverables:</u>

- Problem definition
- Block diagram
- Major component selection rationale
- Proof of major parts
- Purchase ICs, actuators, and sensors from suppliers
- Gantt Chart
- Hardware design
- Manufacture 4 subsystem PCBs
- Software design
- Mechanical design
- Design review
- Proof of all parts
- Full PCB Layout
- Manufacture final PCB
- Bluetooth communication
- Demonstrate functionality of final PCB
- Code for final board
- Demonstrate functionality of PSoC program
- Innovation showcase and poster
- Final report

Project management overview

Individual team member responsibilities and accomplishments:

Jacob Knaup:

I was responsible for the design of the sensor subsystem. I selected a sensor type that would be appropriate for monitoring sleep activity. Flex sensors, which are passive resistive sensors that respond to bending, were selected. Then an appropriate voltage divider and impedance buffer were designed to provide an analog voltage signal to the PSoC. This signal indicated how much the flex sensors were bent, allowing the microcontroller to monitor the pressure from a person on the pillow. Ultimately I selected flex sensors sold by Adafruit and a TI quadruple operational amplifier sold by digikey.

With these components selected, I added them to our team's power budget, so that the team member in charge of our power supply could ensure that it met our requirements. I also wrote our Gantt Chart entries for September 7 through October 5.

Using the design described above, a PCB was manufactured and tested by me to demonstrate the functionality of the sensor subsystem. This involved, placing components and routing traces in Cadence, performing a DFM check, submitting the artwork files to the PERALTA lab, soldering components on the board after it had been milled, and finally, connecting the board to a benchtop power supply and oscilloscope to test functionality. Additionally, I verified the functionality of the final PCB in preparation for hardware demonstration, which involved replacing faulty 5V regulator and soldering connections to off-board sensors and actuators.

I lead the software design and programming by planning the PSoC software in the form of a state diagram. Then, I transferred the planned software design into PSoC C code written in PSoC Creator. Specifically, this involved setting up the PSoC bluetooth module, a 2-channel digital to analog converter, a PWM block, 2 analog input pins, and 3 digital output pins. Then I wrote the appropriate source code to operate the pillow's alarm, monitor the user's sleep, and communicate over bluetooth with the CySmart app running a smartphone. I debugged this software, tuned the frequency and volume of our buzzer to produce a pleasant alarm, and calibrated the code according to the expected sensor values during various states of use (e.g. sleeping vs. awake and restless vs. deep sleep).

Kevin Lam:

Responsible for the design of the power supply subsystem. Used a 12V DC primary power source which was supplied from a wall outlet, and a 7.4V LiPo rechargeable battery as a secondary power source. In our final product, we ended up only using the 12V wall adapter power source. This is because we did not want to risk destroying the battery if we ended up charging it incorrectly. As the components in our project ran on 5V, we used a 5V, 1A LDO regulator (MC7805) in our power supply to bring the 12V down to 5V.

Contributed to the problem definition, explaining how our project can be used to help users by getting information they can give to medical professionals. Picked parts for major component selection rationale, choosing a battery, recharge circuit (which later was excluded from the final design), and primary power supply (wall adapter). Proof of major parts/all parts, ordered all DigiKey parts (buzzers, wall adapter, and amplifiers). Worked on Gantt chart and mechanical design. Helped with the installation of our PCB container into our pillow's foam layer. Also designed and submitted the Innovation Showcase poster. Lastly, worked on the final report.

Pamela Lombardi:

I was responsible for the design of the buzzer subsystem and integration of the circuits. I used a surface mount buzzer, CMT-8540S-SMT-TR, and the frequency was adjusted to assist in the gentle waking of users. The circuit was a simple one that consisted of a resistor, transistor, diode and buzzer.

Integrating the individual subsystems took a reasonable amount of time to join correctly. The subsystems were created in Design Entry CIS, also referred to as Cadence, With the guidance of tutors and available resources, my teams' subsystems were integrated with minimal need for changes. Once the circuit schematic was finalized I began working with the Printed Circuit Board Editor to make our projects final board. After testing the boards continuity, I soldered on the surface mount components as well as the through-hole components.

Anthony Spencer:

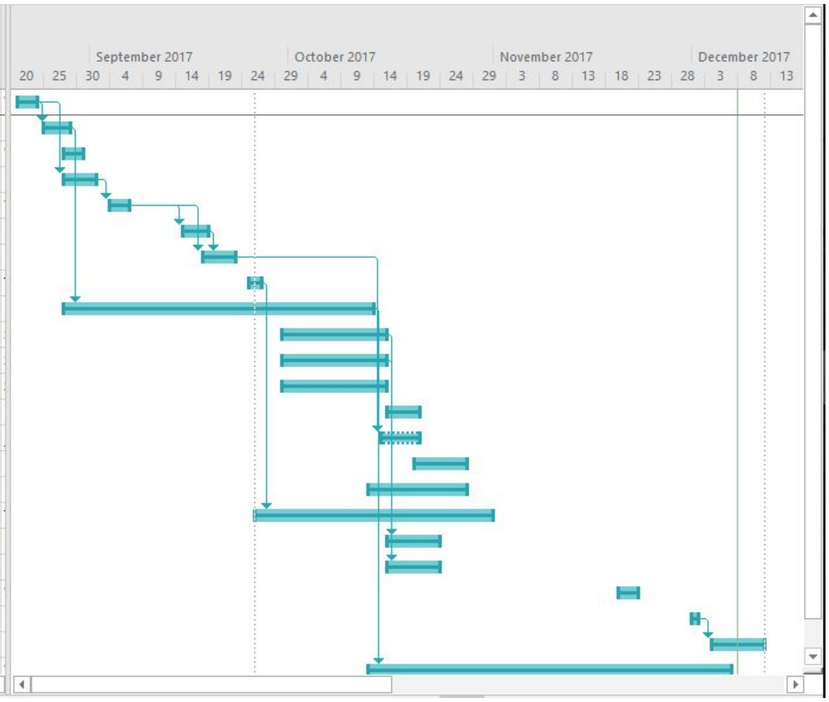
I was responsible for the design of the buzzer subsystem. After carefully researching many different suitable options I came across a mini disc vibration motor. After finding an appropriate transistor to amplify the current from the PSoC to the motor and connecting a parallel diode to protect the PSoC I submitted the buzzer's details to the team's power budget so the correct power supply could be selected.

I was also in charge of developing and updating the team's Gantt chart. The team as a whole created separate sections of the Gantt chart and I used this data to create one massive chart to use as the team's guidelines.

I also created a 3D printed container for the PSoC, PCB, and wiring to sit in before installing the assembly into the foam used to create the pillow.

<u>Final Gantt Chart:</u>

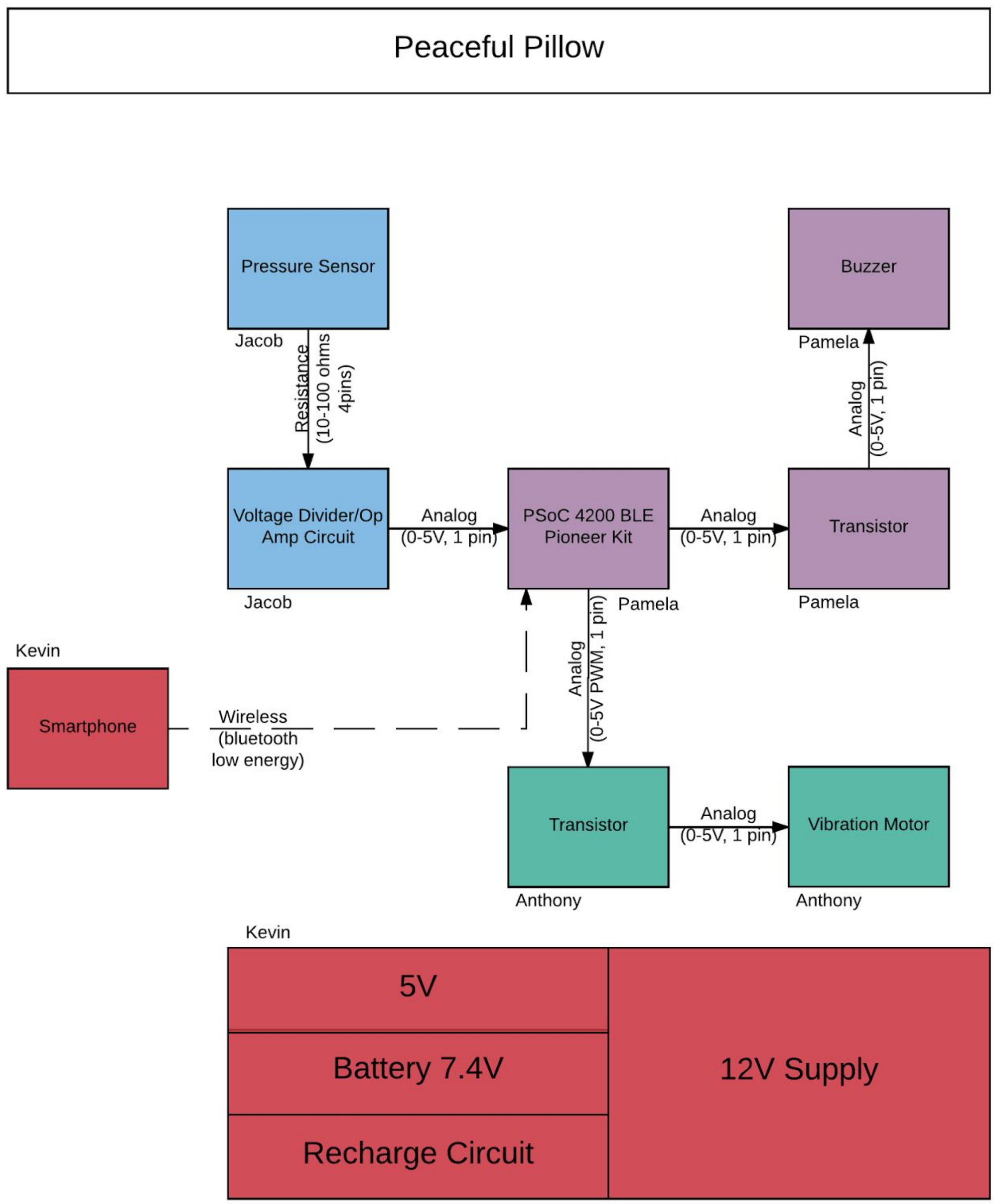
	Task Name	Duration	Start
1	Invention Pitch	3 days	Mon 8/21/17
2	Beginning-of-Semester Project Repository	2 days	Fri 8/25/17
3	Team Charter	3 days	Mon 8/28/17
4	Problem Definition	5 days	Mon 8/28/17
5	Block Diagram	3 days	Mon 9/4/17
6	Major Component Selection Rationale	2 days	Fri 9/15/17
7	Proof of Major Parts	5 days	Mon 9/18/17
8	Initial Gantt Chart	2 days	Mon 9/25/17
9	Middle-of-Semester Project Repository	35 days	Mon 8/28/17
10	Hardware Design	12 days	Sat 9/30/17
11	Software Design	12 days	Sat 9/30/17
12	Mechanical Design	12 days	Sat 9/30/17
13	Design Review	5 days	Mon 10/16/17
14	Proof of All Parts	6 days	Sun 10/15/17
15	Full PCB Layout	6 days	Fri 10/20/17
16	Homework 6: Bluetooth Communication	11 days	Fri 10/13/17
17	Revised Gantt Chart	26 days	Tue 9/26/17
18	Unit Testing Demonstration - Hardware	6 days	Mon 10/16/17
19	Unit Testing Demonstration - Software	6 days	Mon 10/16/17
20	Weeks 13 - 14 Status Report	3 days	Mon 11/20/17
21	Innovation Showcase	1 day	Fri 12/1/17
22	Final Report	6 days	Mon 12/4/17
23	End-of-Semester Repository	39 days	Fri 10/13/17



GANTT CHART

Hardware design narrative

Final Electrical Block Diagram:



Detailed hardware description:

Power Supply: VEL18US120 wall adapter supplies 12V DC and 1.5A through a DC barrel jack which is mounted on the PCB. The jack leads to a 1N4007 diode and into a 5V, 1A LDO regulator which will supply the rest of the components (the pressure/flex sensors, buzzers, vibration motors, and PSoC).

PSoC 4200 BLE: The PSoC reads the flex sensors using 2 analog pins and a 2-channel ADC. It also controls the actuators using three digital output pins and a PWM block. Finally, it sends messages to a smartphone over bluetooth.

Voltage divider/Op Amp Circuit: The voltage divider circuit allows the changing resistance of the passive flex sensors to be measurable. It consists of a second resistor in series with each flex sensor so that as the resistance of a flex sensor changes, the voltage drop across it and its corresponding resistor can also be measured. This voltage is then fed into the op amp, which is set up in the follower configuration to act as an impedance buffer. Although not strictly necessary, the op amp prevents the current drawn off by the PSoC while reading the analog voltage from affecting the measurement. It is a good precaution and is a common practice in industry. The output of each of the op amps channels (one for each pressure sensor) is then connected to an analog input on the PSoC.

Buzzer: The buzzer is connected to a 1N4007 diode, a TIP31C transistor and a 180 Ohm resistor.

Pressure sensor: The flex sensors detect pressure when the user places their head on the pillow. This pressure, bends the flex sensors, causes the resistance of the sensors to change. As a result of then voltage divider circuit, the voltage drop across the flex sensors therefore also changes. The pressure sensors are connected to the LM324 op amp and PSoC.

Vibration Motors: Each one is connected to a 2N2222 transistor and 1N4007 diode

Smartphone: The smartphone uses the CySmart App to connect to the PSoC via bluetooth.

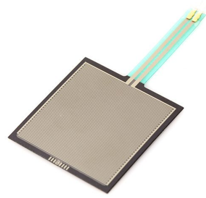
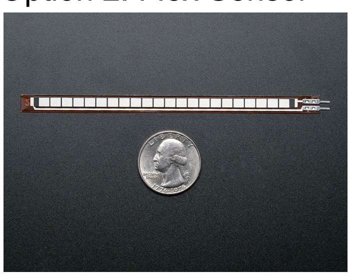
Criteria	Relevant Component	Explanation
Cr-1: Variable audio alarm	Buzzer	The buzzer’s volume can be varied dramatically by adjusting the duty cycle of the PWM square wave sent to it by the PSoC
Cr-2: Measure sleeping	Flex sensors	It was found from testing that the flex sensors are very sensitive to small deflections and have relatively low noise, allowing the user’s sleep activity to be measured in detail
Cr-3: Battery Life	LiPo battery	Battery was able to hold a charge through at least one nights sleep to avoid the need for the pillow to be connected to the outlet.
Cr-4: Number of pressure sensor collection samples	Flex sensors & PSoC	The PSoC’s ADC was able to be configured with an update time on the order of microseconds, providing more than enough data to get an accurate picture of sleep activity
Cr-5: Accuracy of keeping track of time	PSoC & smartphone BLE	The PSoC is able to keep track of time on its own relative to when the user goes to sleep, provided it stays powered on
Cr-6: Size of entire unit	All components	The final size of the pillow is 14” W x 18”L x 2.5” H

Constraints	Explanation
Co-1: Stay cool (components)	Breathable foam material to keep user comfortable and components from overheating
Co-2: Material, (flame retardant)	Flame retardant foam was used because of high heat resistance and should an anomaly of excessive overheating occur, the foam would melt.
Co-3: Pillow Size	Pillow is 14” x 18” x 2.5” as this is an average size which allowed space for

	component placement without interfering the user’s comfort
Co-4: Prototype budget	Actual amount spent was $105.02
Co-5: Power supply (AC adapter, battery, or solar panel connected to voltage regulator. No USB power packs may be used)	12V AC-DC adapter connected to 5V LiPO regulator
Co-6: Microcontroller (Cypress Bluetooth® Low Energy (BLE) Pioneer Kit (CY8CKIT-042-BLE-A), PSoC® 4 Pioneer Kit (CY8CKIT-042), and/or Prototyping Kit (CY8CKIT-049-42xx))	1 Cypress PSoC BLE was used
Co-7: Sensor(s) read by a microcontroller	2 flex sensors were read by the PSoC
Co-8: Actuator(s) controlled by a microcontroller	2 vibration motors and 1 buzzer were controlled by the microcontroller
Co-9: Bluetooth Low Energy communications to a phone, computer, or another device using a Cypress BLE Pioneer Kit	Cypress PSoC BLE communicated over bluetooth with the CySmart app running on a smartphone
Co-10: Custom printed circuit board ● Must be created in Cadence ● No commercial boards	1 PCB was included in the pillow
Co-11: Surface mount components ● Size 0805 or larger recommended	The buzzer as well as all bypass capacitors were surface mount components
Co-12: Programmed in C or C++	PSoC was programmed in C using PSoC Creator

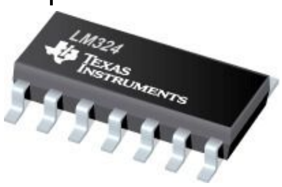
Final Major Component Selection Rationale:

Pressure Sensor

Solution	Pros	Cons
Option 1. Force Sensitive Resistor  https://www.sparkfun.com/products/9376	• Larger measurement area • Works with harder pillows • Cheaper	• Must be placed under head • Less accurate • Requires direct pressure • Less compatible with softer pillows
Option 2. Flex Sensor  https://www.adafruit.com/product/182	• Works with softer pillows • Can be placed off to side • Wide range of supply voltages	• Smaller • Requires more sensors to cover pillow • More expensive

Choice: Option 2: Flex Sensor
Rationale: Flex sensors will be easier to embed in the pillow because they can be distributed throughout our pillow to measure different locations. This will be useful for detecting movement of the user. They will also be more sensitive than Option 1 since they measure deformation rather than force.

Sensor Op Amp Circuit

Solution	Pros	Cons
Option 1. LM358 Dual Op Amp  http://www.ti.com/product/lm358	• Recommended by force sensitive resistor data sheet • Large supply voltage range • Low frequency	• Only 2 channels • Small package size • Low output current
Option 2. LM324 4 channel Op Amp  http://www.ti.com/product/lm324	• Recommended by flex sensor data sheet • 4 channels • Wide supply voltage range	• Medium-small package size • More expensive • Lower frequency band

Choice: Option 2, 4 channel Op Amp
Rationale: The second option is the best because it allows for more advanced customization since it includes 4 operational amplifiers allowing us to use more flex sensors in our pillow. The price difference over the dual op amp is small, so it seems worthwhile to get more channels. Additionally, it was recommended by the flex sensor data sheet, so it will be compatible with our sensor.

Buzzer

Solution	**Pros**	**Cons**
Option 1. Speaker  http://www.jameco.com/z/FR-29PC-100-MG-Electronics-Single-Pole-Speaker-Round-PC-Mount-1-2-Inch-Diameter-100-Ohm-15-20w-1-1kHz-plusmn-20-Mylar_2099817.htm	• Variable sound output • dB rating of 95 +/-3 • Freq 8.3KHz	• Poor Datasheet • Audio files required • More difficult programming
Option 2. Speaker  http://www.jameco.com/z/38MS30008-MG-Electronics-Single-Pole-Speaker-Round-PC-Mount-1-3-Diameter-8-Ohm-0-15-0-20-Watts-700hz-4kHz-Mylar_2095234.html	• Variable sound output • dB 83 • Freq 700-4000Hz	• Audio files required • Poor Datasheet • Higher power requirements
Option 3. Siren  https://www.digikey.com/product-detail/en/cui-inc/CMT-8540S-SMT-TR/102-3746-1-ND/6012447	• Quality datasheet • Simple - no audio file • No Op-Amp required • 4000 Hz freq	• Price per unit • Transistor required • Limited control of volume

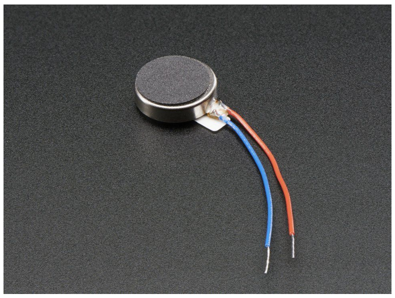
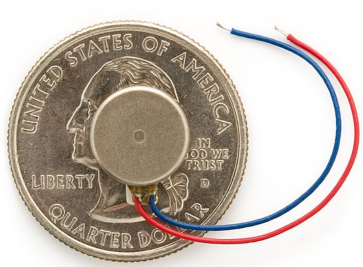
Choice:Option 3 Siren
Rationale: Option 3 is the best choice for sound emission. The siren does not require any audio file which makes it simpler to code. The compact size of the siren allows it to be surface mounted.

Transistor

Solution	Pros	Cons
Option 1.TIP31C TRANS NPN 100V 3A TO-220  https://www.digikey.com/product-detail/en/stmicroelectronics/TIP31C/497-2615-5-ND/603640	• Excellent datasheet • Comparable to recommended part for siren • VCEO - 45V	• Collector cut-off of 0.1 μA • IC 100 mA • Surface mount
Option 2. MPSA06RA  https://www.digikey.com/product-detail/en/fairchild-on-semiconductor/MPSA06RA/MPSA06RACT-ND/1626265	• Excellent Datasheet • VCEO - 80V • IC - 500mA	• Collector cut-off of 0.1 μA • Throughout • More expensive

Choice: Option 1.
Rationale: The first option is the best choice to handle the current required to pass through the transistor to the buzzer/motor. Although option one has a higher per unit cost, the benefits will outweigh the costs and is comparable to the recommended transistor from the buzzers datasheet.

Vibration Motor

Solution	Pros	Cons
Option 1. Vibration Motor https://www.adafruit.com/product/1201 	• 2-5V • Light • Cheap • Explanation for use with transistor	• No official datasheet • No test data • No part number
Option 2. Vibration Motor https://www.sparkfun.com/products/8449 	• Extremely detailed datasheet • Light • Reliable	• 2.3-3.6 V • Datasheet is half in Chinese • Thick(er) • Expensive

Choice: Option 1, Vibration Motor
Rationale: While this motor doesn’t have a detailed datasheet all of the information needed is provided via the adafruit website. This motor also will operate on the 5V power rail that we have decided to use while the other motor listed does not.

Power Supply Selection

12V Power Supply

Solution	Pros	Cons
Option 1. VEL18 Series  http://www.xppower.com/Portals/0/pdfs/SF_VEL18.pdf	• Easy to use • Good voltage output • Can plug into wall	• Limited input voltage • Single output • Output voltage options limited
Option 2. PMT-12V35W1A  http://www.deltapsu.com/products/download/Datasheet/PMT-12V35W1AA	• Protects from short circuiting and overheating • Versatile connection options • Universal AC input range	• Limited output power options (11-14V) • Might have surplus of power • Panel Mount

Choice: Option 1

Rationale: Option 1 has just enough current output and works with our battery charger, also can be plugged into wall, so it is easy to use.

3.3-12V Battery

Solution	Pros	Cons
Option 1. https://www.sparkfun.com/products/11855 7.4V lithium ion battery  https://www.sparkfun.com/products/10473	• Safe from fire and leakage • Easy to find charger • Easy to use	• Needs its own special charger • Expensive • Poor data sheet
Option 2. 6V Rechargeable battery https://na.industrial.panasonic.com/sites/default/pidsa/files/lcseries_datasheets_merged.pdf 	• Small and compact • Low maintenance • Cheap	• Less voltage • Hard to find charger • Shorter battery life

Choice: Option 1

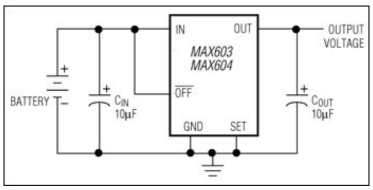
Rationale: Option 1 is the best choice for the battery, because not only is this battery smaller but so is the charger, and there is more voltage.

Recharge Circuit

Solution	**Pros**	**Cons**
Option 1:  https://cdn.sparkfun.com/datasheets/Prototyping/IMAX-B6.pdf	• Safe • Ease of use • Not bulky	• Expensive • Fragile • Lots of settings
Option 2:  http://www.mouser.com/ds/2/331/A-F_Series_Chargers_10_Sept_27-8389.pdf	• Easy to use • Many options • Safe	• Expensive • Multiple parts • Bulky

Choice: Option 1
Rationale: Option 1 is the better choice because using this charger is a simpler process and there are not as many parts to worry about.

5V Regulator

Solution	**Pros**	**Cons**
Option 1: MAX603  https://www.maximintegrated.com/en/products/power/linear-regulators/MAX603.html	• Low dropout • Thermal overload protection • Large input range (2.7-11.5V)	• Low current <500mA • Drops out at 320 mV • More expensive
Option 2: MC7805  http://pdf.datasheetcatalog.com/datasheet2/9/0p4t1g1lw0spoo5ukh2s1uxchzky.pdf	• Low dropout • High current (1.0 A) • 5V output	• Heat sink gets very hot • Lower Temperature Range • Lower Output Range

Choice: Option 2
Rationale: Option 2 provides a higher maximum output current, which will be a useful precaution to ensure we can source enough current to run all of our components at once. Although the heatsink gets hot, the foam we are using is flame resistant.

Final Power Budget:

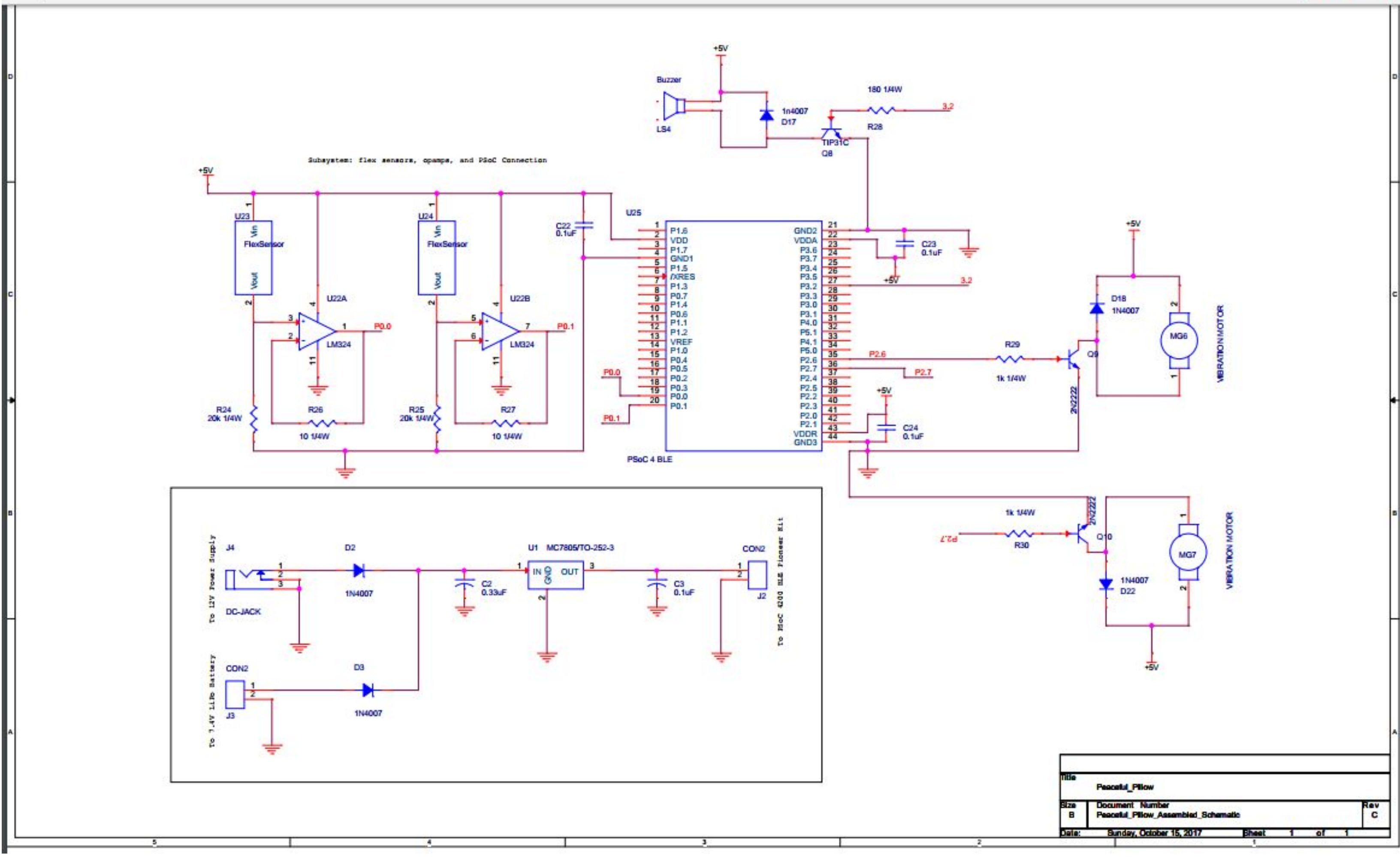
Peaceful Pillow Power Budget						
Team Number: 6	**Team Members: Jacob Knaup, Kevin Lam, Pamela Lombardi, Anthony Spencer**		**Project Name: Peaceful Pillow**		**Version: 1**	
1. List major components						
Component Name	**Part Number**	**Supply Voltage Range**	**Quantity**	**Absolute Maximum Current (mA)**	**Total Current (mA)**	
Op Amp	LM324	3.0-32V	1	60	60	mA
Buzzer	CMT-8540S-SMT-TR	1-6 V	1	150	150	mA
Vibration Motor	1201	2V-5V	3	100	300	mA
PSoC™ 4 BLE Module	CY8CKIT-042-BLE	3.3V/5V	1	200	200	mA
5V Regulator	MC7805	5-18 V	1	2200	1000	mA
2. Assign major components to power rails. Try to minimize the number of power rails necessary. Add additional voltage rails if needed.						
5V Power Rail						
Component Name	**Part Number**	**Supply Voltage Range**	**Quantity**	**Absolute Maximum Current (mA)**	**Total Current (mA)**	
PSoC™ 4 BLE Module	CY8CKIT-042-BLE	3.3V/5V	1	200	200	mA
Op Amp	LM324	3.0-32V	1	60	60	mA
Vibration Motor	1201	2V-5V	3	100	300	mA
Buzzer	CMT-8540S-SMT-TR	1-6 V	1	150	150	mA
				Subtotal	710	mA

				Safety Margin	25%	
5V Regulator				***Total Current Required on 5V Rail***	887.5	mA
5V Regulator	MC7805	5-18 V	1	1000	1000	mA
3. Calculate Power Supply Requirements						
Component Name	**Part Number**	**Supply Voltage Range**	**Quantity**	**Absolute Maximum Current (mA)**	**Total Current (mA)**	
5V Regulator	MC7805	5-18 V	1	1000	1000	mA

Power Supply Design:

Our power supply was designed to produce 5V and 1A of current for our circuit, as each component ran on 5V and the total current required for the system was 1 amp. The 12V power supply coming from the wall adapter produces a maximum of 1.5A and with our low dropout voltage regulator, is brought down to 5V and 1 amp.

Final Schematics:



Software Design Narrative

Software Design Rationale:

At the outset of the project, two functional goals were defined: monitor the user's sleep and gently, yet effectively wake the user. Our software accomplishes those two goals using the state machine defined below.

The sleep tracking is accomplished within the User Sleeping state. Each time the loop runs, the PSoC reads an analog signal from the pressure sensors and determines whether the user's head is or is not on the pillow at that instant by comparing the reading with a threshold value. The number of times the user is found to be off the pillow is recorded and this value is then used to assess how much they moved around while sleeping. The user is then notified about how they slept through their smartphone before the program enters the User Awake state. Although the information the user receives is only a general analysis, it does fulfill the goal of giving the user some indication of how they slept in order to educate them and potentially raise their interest so that they may consult a doctor.

The alarm functionality is accomplished by reading the pressure sensors in the User Awake and Alarm states and by activating the buzzer and motors in the Alarm state. Utilizing the pressure sensors as the means to turn off the alarm ensures that the user must get off the pillow and stay off for several minutes consecutively in order for the alarm to deactivate. If a user refuses to get up and stay up, the alarm will continue to sound. This fulfills the goal of effectively waking the user by preventing someone from turning off their alarm, going back to sleep, and then oversleeping, a common problem we found when talking to fellow college students.

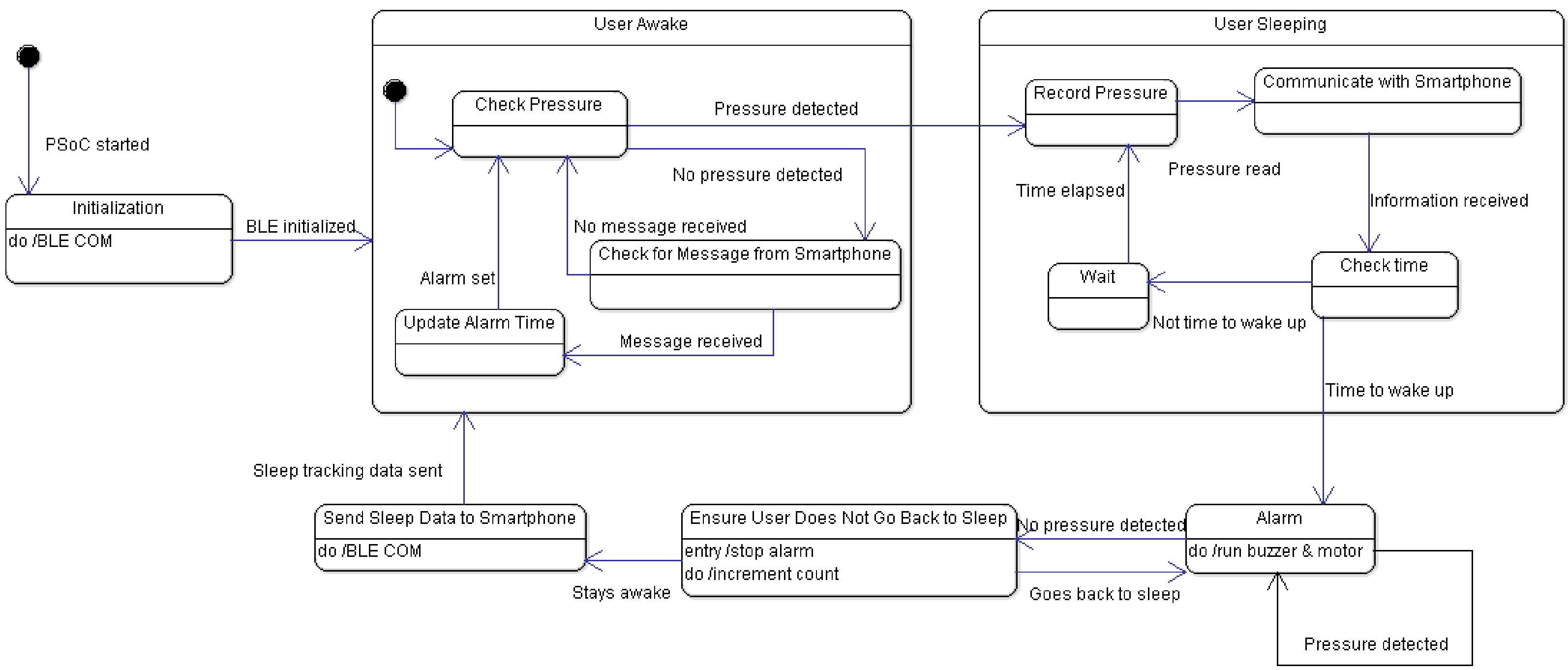
Criteria	**Explanation**
Cr-1: Variable audio alarm	The buzzer's volume can be varied dramatically by adjusting the duty cycle of the PWM square wave sent to it by the PSoC
Cr-2: Measure sleeping	The software tracks how often the user's head is on the pillow vs off the pillow
Cr-3: Battery Life	The software did not need to do anything special related to battery life
Cr-4: Number of pressure sensor collection samples	The PSoC reads the ADC (and therefore the pressure sensors) everytime the "sleeping" loop runs, therefore collecting thousands of samples per second

Cr-5: Accuracy of keeping track of time	The PSoC is able to keep track of time as long as it is powered on by using the CyDelay function and incrementing a variable each time the function is called
Cr-6: Size of entire unit	The software has no relation to this criterion

Constraints	**Explanation**
Co-1: Stay cool (components)	N/A
Co-2: Material, (flame retardant)	N/A
Co-3: Pillow Size	N/A
Co-4: Prototype budget	N/A
Co-5: Power supply (AC adapter, battery, or solar panel connected to voltage regulator. No USB power packs may be used)	N/A
Co-6: Microcontroller (Cypress Bluetooth® Low Energy (BLE) Pioneer Kit (CY8CKIT-042-BLE-A), PSoC® 4 Pioneer Kit (CY8CKIT-042), and/or Prototyping Kit (CY8CKIT-049-42xx))	1 Cypress PSoC BLE was programmed
Co-7: Sensor(s) read by a microcontroller	2 flex sensors were read by the PSoC using 2 analog input pins and a dual channel ADC
Co-8: Actuator(s) controlled by a microcontroller	2 vibration motors and 1 buzzer were controlled by the microcontroller using three digital output pins and a PWM block
Co-9: Bluetooth Low Energy communications to a phone, computer, or another device using a Cypress BLE Pioneer Kit	The PSoC was programmed to send a message to the user's smartphone over BLE telling them how they slept
Co-10: Custom printed circuit board ● Must be created in Cadence ● No commercial boards	N/A

Co-11: Surface mount components • Size 0805 or larger recommended	N/A
Co-12: Programmed in C or C++	PSoC was programmed in C using PSoC Creator

Software Design:



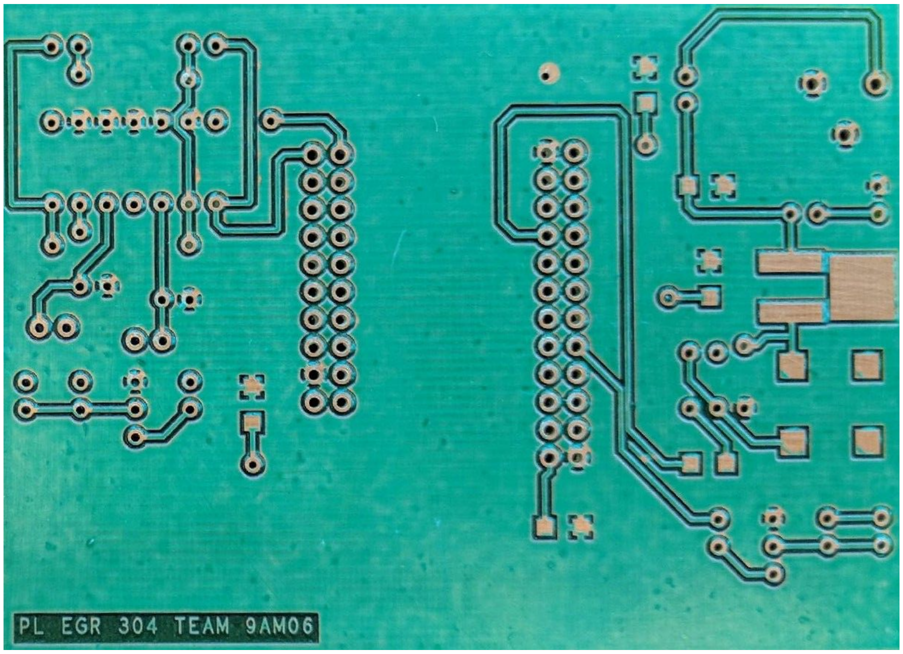
Mechanical Design Narrative

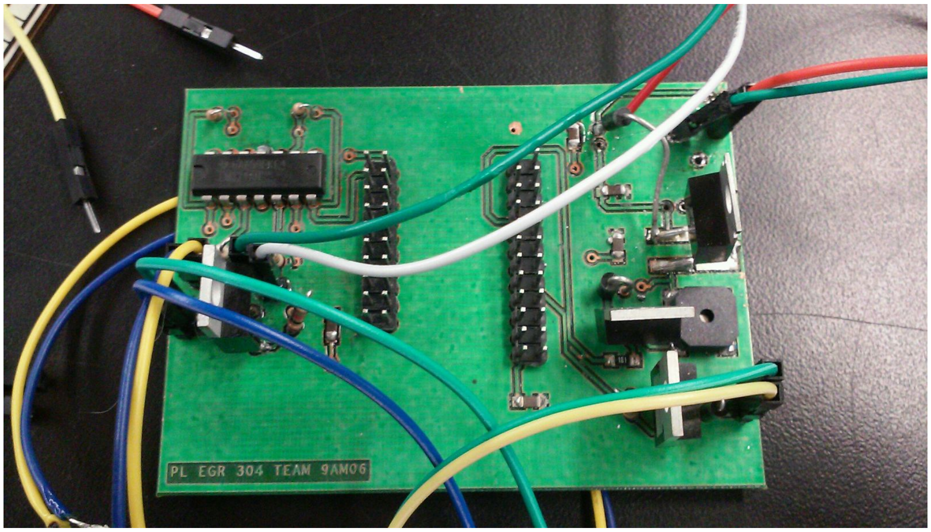
Mechanical Design Rationale:

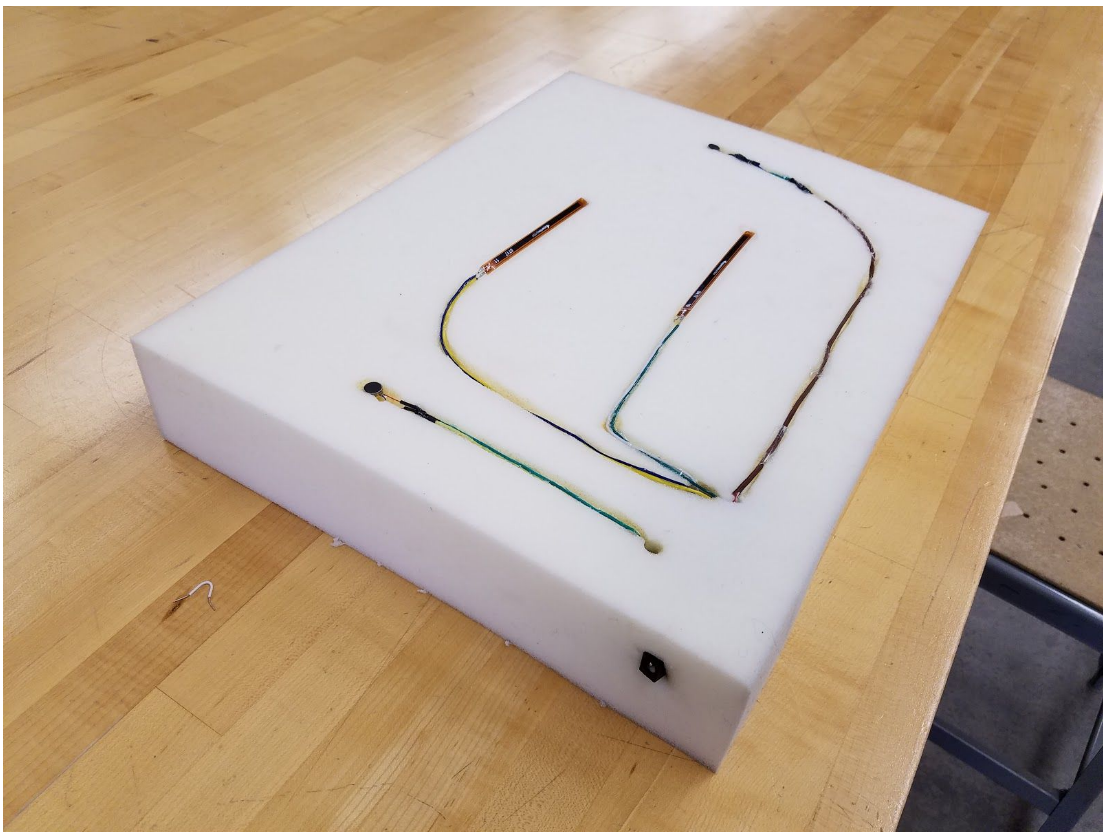
The foam chosen was selected because of its flame retardant properties. The pillow foam meets California standard for flame retardancy, Cal 133 and Cal 117 fire codes. These codes represent the amount of time it takes for the composite to burn with the Cal 133 code as one of the most stringent. The circuit board was placed in a 3D printed box for the protection of the components and the user's comfort. The component box was placed in the top right corner of the pillow since this area is commonly the least used area and allows for easier access to DC charging of the pillow. A hole was cut in the side of the pillow in order to allow the DC charging port to be accessed by the user. A hole was also cut through the pillow to route wires for the sensors and vibrator motors to the surface. Then, grooves were cut in the surface of the pillow in which those components were routed. Finally, they were glued down in their respective grooves on the surface of the pillow. This solution made the pillow comfortable while still leaving the components accessible for repairs and debugging. It also made the vibrations very easy for the user to feel while still keeping the volume of the buzzer pleasant. Finally, it provided the flex sensors with enough resolution to detect the difference in the compression of the foam from when a user's head was on the pillow versus when it was not.

<u>Project in context:</u>

Project Photo(s):




PL EGR 304 TEAM 9AM06



Version 2.0+ and lessons learned

Hardware design:

A smaller rechargeable battery could be used as our primary power supply instead of using a wall adapter or a lithium polymer battery. It would be preferable if the rechargeable battery also has a low risk of overheating, even if our foam is fireproof. Having our pillow be wireless would definitely be more appealing to users, as they wouldn’t have to deal with seeing a long wire stick out from the inside of their pillow. Therefore, using a long-life battery that is safe for use in proximity to humans and flammable substances and including wireless inductive charging would improve our power subsystem.

Another improvement that could be made is using a pressure sensing pad that covers the entire top surface of the pillow foam. This way, regardless of the position of the user’s head on the pillow, their presence will be sensed. Perhaps even the amount of pressure on different areas of the pillow could be observed for more in depth analysis of sleeping behavior. This would allow the pillow to function as more than merely an alarm and as a source of casual information about sleep habits. With better sensing, it could be used for clinical sleep diagnosis and it could be used to screen for or monitor conditions such as sleep apnea or infant sudden death syndrome.

Software design:

For our software, we can implement a program that detects the positioning of the user's head on the pillow (for example: right side or left side, higher or lower). We could also improve the PSoC software to measure how heavily the user's head is resting on the pillow. This would use the full capabilities of our analog input. Currently the analog input is being compared to a threshold value, and then it is essentially turned into a binary value (head is on the pillow or off the pillow).

We could also log the sleep data with timestamps and then send all of the data to the user's smartphone. This would only be practical if we had a custom phone app that was capable of displaying all of this information in graphical form. However, it would be useful for the user as it would allow them to see at what times they slept more soundly and at what times they were more restless. This would be comparable to the information given to users by some of our competitors. Whereas right now we only assess their sleeping as a whole, this representation would give the users a more detailed breakdown in a form that was still easily understandable.

We could also adapt our software to include more functionalities such as playing white noise while the user is sleeping. Or it could give the user feedback to encourage good sleeping practices. The sensing could also be adapted to look for other factors besides movement while the user is sleeping. For example, with very sensitive monitoring, the pillow could detect breathing and the pillow could be used to monitor conditions such as snoring and sleep apnea. With this modification the pillow could be used to screen for these conditions and give the user a warning. With this awareness, the user could then investigate their condition in more detail and consult a physician. An interesting suggestion we received at the innovation showcase was that we turn the pillow into a pad and make the software more precise to watch for infant sudden death syndrome. By detecting breathing, the pillow could send a notification to the parent's phone if the baby stops breathing.

Mechanical design:

The mechanical design could be improved by adding a case to the pillow. Originally, this was our plan, however it was found that the flex sensors were too sensitive and our program mistook the the case for a person's head. The advantage of adding a case would be that it would make our product more visually appealing for the user and would hide the electrical components which can make the pillow look intimidating. It would also improve the pillow by reducing the risk of the components being inadvertently damaged or pulled off the pillow by the user while sleeping. With the current design, the sleeper lies directly on top of the sensors, which is an obvious source of unevenness and hardness in the pillow.

In order to add a case without interfering with the pressure sensing system, we could alter the physical placement of the sensors. Currently the wires, sensors, and vibration motors are routed on top of the pillow in shallow grooves and glued down. A better mechanical solution would be to embed the sensors and vibrator motors in the foam, about an inch beneath the surface. This would prevent the case from affecting the sensors while still allowing them to detect the compression of the foam by the user's head. This would also hide the electronic components so that they are not unightly or frightening to the user and it would reduce the risk of them getting damaged.

Another way we could improve the mechanical design is by using softer foam. The foam we used was specially selected because it was fire retardant. While we believe this is an important safety feature, the foam was firmer than we expected and did not make for a particularly comfortable pillow. Using softer foam would provide for a more comfortable sleeping experience and would be closer to the firmness of the average pillow. Additionally, it would help our sensors detect the user's sleep patterns because the softer foam would compress more under the weight of the user's head than our harder foam. Therefore, the flex sensors would bend more providing a greater difference in the signal when the user is lying down vs when they are up. This greater range in bending would also make it easier to detect smaller motions such as breathing, which would be useful to detect as described in the software section above.

<u>Lessons learned:</u>

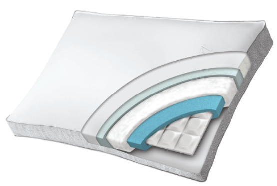
We learned:

1. Check each component, section, and subsystem as you go
2. When debugging, break the problem down and isolate what is not working
3. Soldering neatly can save you a lot of time in troubleshooting
4. Doing the preparation work for a project up front (and doing it right) will save you time later on
5. Scoping a project appropriately to the given criteria and constraints will lead to a better product in the end
6. Testing a product in its exact expected use conditions is important for catching any hidden bugs
7. Creating a design that includes buffer for unexpected circumstances (such as using a higher power power supply than necessary) is a good way to prepare for unforeseen circumstances
8. Choosing components with detailed data sheets is important in order to design around the component's specifications

9. Design reviews and running a design by others is a good way to catch problems that you may not have thought of
10. Software should be planned out before it is programmed

<u>Appendices</u>

Benchmarking Related Solutions:

Solution	**Description**	**Pros**	**Cons**
iComfort Hybrid Triple Effects Pillow ($109) 	The iComfort Pillow solves the problem of poor sleep by providing pressure point relief where needed, supports proper body alignment despite your sleeping angle, and regulates body temperature.	• Adaptive • Simple • Comfortable	• Expensive • Doesn’t give feedback • Works best with the same brand mattress
Oittm Smart Pillow Mat ($25.99)  http://www.oittm.com/Oittm-Smart-Pillow-Mat	The Oittm Smart Pillow Mat solves the problem of poor sleeping by playing music to aid sleeping from the user’s smartphone, tracking the user’s sleep patterns, and coordinating an alarm with the user’s sleep patterns to awake them at the ideal time.	• Numerous features (alarm, music, and tracking) • Integrated sensors and speakers	• Wired connection to smartphone • Must be placed under existing pillow (separate purchase)

Somnox (499 Euro, 597.73 US) 	Somnox simulates human breathing, is able to identify sleep cycles, play music and gently wake the user.	• Eases anxiety • Bluetooth enabled • data collection	• Expensive • Limited quantity release
Sleep Cycle alarm clock App ($29.99 for 1 year premium)	This app has your phone track your sleep states by using sound analysis to track movements during sleep. It will wake you up slowly during your lightest sleep phase.	• Low risk of harm to the user (no physical contact between phone and user). • Being slowly woken during lightest sleep phase means there is no sudden disturbance.	• Does not have in depth analysis of sleeping behaviors. • Does not have a timer for waking up at specific times.
Sleep as Android (free app) http://sleep.urbandroid.org	Sleep as Android solves the problem of waking up in the morning by playing an alarm while the user is sleeping lightly and then verifies the user has woken up by making the user complete a short puzzle to disable to alarm.	• Free • Verifies wakeup • Smart alarm goes off at optimal time	• Requires placing phone on bed • Additional purchases

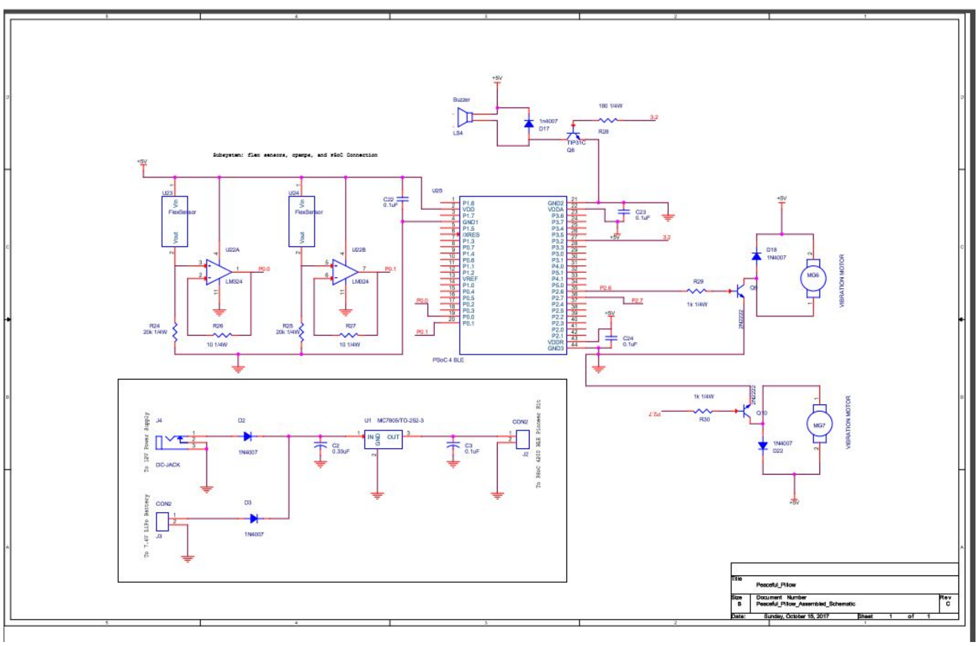
Final Bill of Materials:

Part Name/Description	Unit Quantity	Unit Prototype Cost	Total Prototype Cost	Unit Production Cost	Total Production Cost	Manufacturer	Manufacturer Part #	Supplier	Supplier Part #	# Ordered	Date Ordered	# Received	Surplus	Schematic Reference Designators
VEL18US120 - 12V Power Supply	1	$12.75	$12.75	$12.75	$12.75	XP Power	VEL18US120-US-JA	Digi-Key	1470-3157-ND	1	9/22/2017	1	0	
Lithium Ion Battery - 2200 mAh 7.4V	1	$15.95	$15.95	$15.95	$15.95	King Max	RC02269	Sparkfun	PRT-11856	1	9/21/2017	1	0	
Li-ion/Polymer Battery Charger/Balancer	1	$32.95	$32.95	$32.95	$32.95	iMax	U1B600A113	Sparkfun	PRT-10473	1	9/21/2017	1	0	
Flex Sensor	2	$12.95	$25.90	$10.36	$20.72	Spectra Symbol	FSL0095103ST	Adafruit	182	3	9/22/2017	3	1	U23,U24
Vibrating Mini Motor Disc	2	$1.95	$3.90	$1.56	$3.12	Adafruit	1201	Adafruit	1201	4	9/22/2017	4	2	MG6,MG7
Quadruple Operational Amplifier	1	$0.40	$0.40	$0.16	$0.16	Texas Instruments	LM324AN	Digi-Key	296-9542-5-ND	4	9/22/2017	4	3	U22
AUDIO MAGNETIC XDCR 1-6V SMD	3	$4.59	$13.77	$2.37	$4.74	CUI Inc	CMT-8540S-SMT-TR	Digi-Key	102-3746-1-ND	2	9/22/2017	2	-1	LS4
PSoC 4 BLE	1	$0.0	$0.	$14	$14	Cypr	CY8C	PR	PRLT	2	9/2	2	1	U25

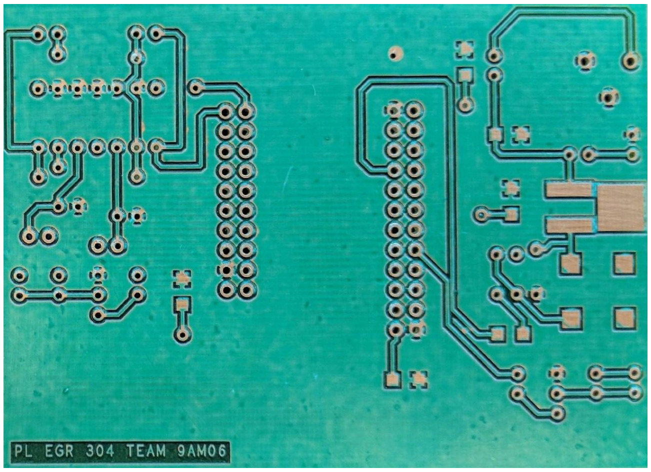
Module		0	00	.96	.96	ess	KIT-143A	LTA Lab	A Lab		0/2017			
Two Pin Header	4	$0.01	$0.04	$0.01	$0.01	PRLTA Lab	PRLTA Lab	PRLTA Lab	Header	4	9/21/2017	4	0	J2,J3,U23,U24
DC Jack	1	$0.01	$0.01	$0.01	$0.01	PRLTA Lab	PRLTA Lab	PRLTA Lab	PJ-102A	1		1	0	J4
Resitor 2k Ohm	2	$0.01	$0.02	$0.01	$0.01	PRLTA Lab	PRLTA Lab	PRLTA Lab	2k Ohm	2	9/21/2017	2	0	R24,R25
Resitor 10 Ohm	2	$0.01	$0.02	$0.01	$0.01	PRLTA Lab	PRLTA Lab	PRLTA Lab	10k Ohm	2	9/21/2017	2	0	R26,R27
Resitor 1k Ohm	2	$0.01	$0.01	$0.01	$0.01	PRLTA Lab	PRLTA Lab	PRLTA Lab	1k Ohm	2	9/21/2017	2	0	R28
Resitor 180 Ohm	1	$0.01	$0.01	$0.01	$0.01	PRLTA Lab	PRLTA Lab	PRLTA Lab	180 Ohm	1	9/21/2017	1	0	R29,R30
Capacitor 0.33uF	1	$0.01	$0.01	$0.01	$0.01	PRLTA Lab	PRLTA Lab	PRLTA Lab	0.33uF	1	9/21/2017	1	0	C2
Capacitor 0.1uF	4	$0.01	$0.04	$0.01	$0.01	PRLTA Lab	PRLTA Lab	PRLTA Lab	0.1uF	4	9/21/2017	4	0	C3,C22,C23,C24
Diode	5	$0.01	$0.05	$0.01	$0.01	PRLTA Lab	PRLTA Lab	PRLTA	1N4007	5	9/21/2017	5	0	D2,D3,D17,D18,D22

								Lab						
Transistor TIP31C	1	$0.01	$0.01	$0.01	$0.01	PRLTA Lab	PRLTA Lab	PRLTA Lab	TIP31C	1	9/21/2017	1	0	Q8
Transistor 2N2222	1	$6.00	$6.00	$6.00	$6.00	Adafruit	PN2222	Adafruit	2n2222	1	10/12/2017	1	0	Q9,Q10
Voltage Regulator	1	$0.01	$0.01	$0.01	$0.01	PRLTA Lab	MC7805	PRLTA Lab	LM7805	1	10/12/2017	1	0	U1

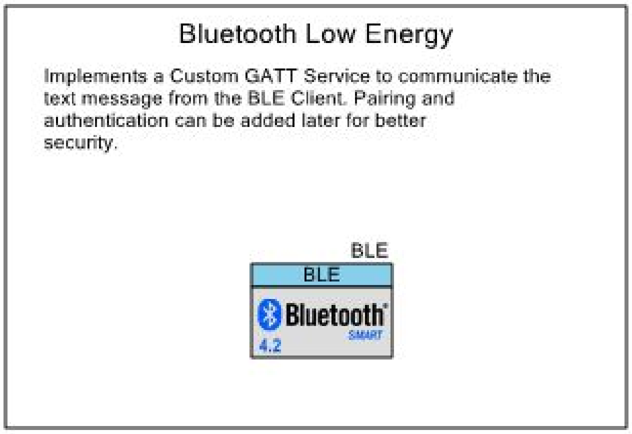
Final printed circuit board schematic:

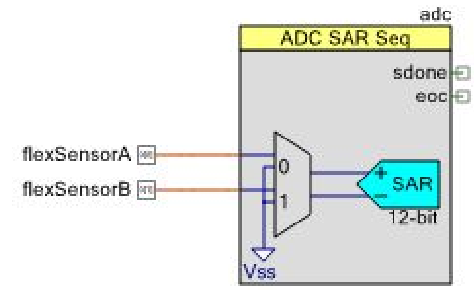


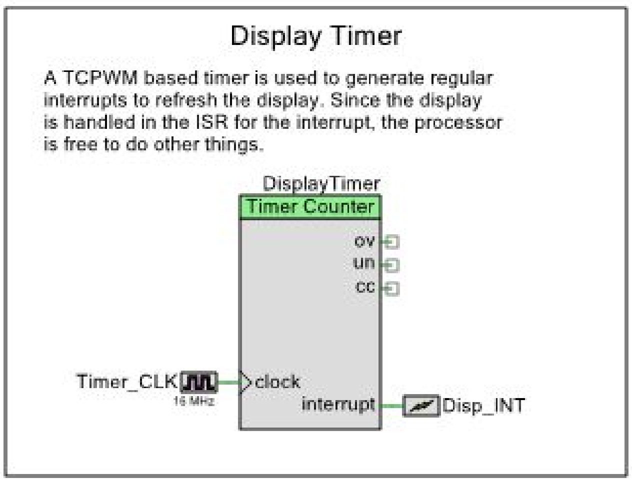
Manufactured PCB:

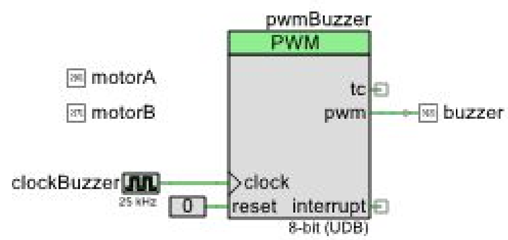

PL EGR 304 TEAM 9AM06

PSoC Creator Top Design:









PSoC Creator Source Code

```
/* ========================================
 *
 * Copyright Jacob Knaup & Team 6, 2017
 * All Rights Reserved
 * UNPUBLISHED, LICENSED SOFTWARE.
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION
 * WHICH IS THE PROPERTY OF Jacob Knaup & Team 6.
 *
 * ========================================
*/
#include <main.h>

char buffer[180];
```

```
void DisplayMessage(char *message, uint8 length)
{
        uint8 i;

        //Timer_CLK_Stop();
        //pos = 0;

        stpncpy(buffer, message, length);
        for (i = length; i<180; i++)
        {
        buffer[i] = 0;
        }

        //Timer_CLK_Start();
}
void testFlex(){
        int analog1, analog2;
        char buffer[] = "Flex Sensor 1: 2016\n Flex Sensor 2: 2017";
        adc_Start();
        adc_StartConvert();
        UART_Start();
        for(int i=0;i<100;i++){
        adc_IsEndConversion(adc_WAIT_FOR_RESULT);
        analog1 = adc_GetResult16(0);
        analog2 = adc_GetResult16(1);
        sprintf(buffer, "Flex Sensor 1: %d Flex Sensor 2: %d\n", analog1, analog2);
        //print
      CyDelay(100);
        }
}

void testMotor(){
        motorA_Write(0);
        motorB_Write(1);
        CyDelay(2000);
        motorA_Write(1);
        motorB_Write(0);
        CyDelay(2000);
```

```
	motorA_Write(0);
}

void testBuzzer(){
	pwmBuzzer_WriteCompare(50);
	CyDelay(3000);
	pwmBuzzer_WriteCompare(0);
	CyDelay(3000);
}

void initializeSubsystems(){
	pwmBuzzer_Start();
	pwmBuzzer_WriteCompare(0);
	adc_Start();
	adc_StartConvert();
	uart2_Start();
	CyBle_Start(StackEventHandler);
}

int flexGet(int num){
	adc_IsEndConversion(adc_WAIT_FOR_RESULT);
	return adc_GetResult16(num);
}
void buzzerSet(int compare){
	pwmBuzzer_WriteCompare(compare);
}

int flexA, flexB;
void updateFlexSensors(){
	flexA = flexGet(0)-500;
	flexB = flexGet(1)-650;
}
void activateAlarm(int on){
	motorA_Write(on);
	motorB_Write(on);
	pwmBuzzer_WriteCompare(50*on);
}

int main()
```

```
{
	/* Place your initialization/startup code here (e.g. MyInst_Start()) */
	//StandardDisplayInit();

	CyGlobalIntEnable;
	char text[32];
	int threshold = 500;

	initializeSubsystems();
	sprintf(text, "Waiting          ");
	DisplayMessage(text, sizeof(text));

	while(1)
	{
	for(int i=0; i< 100100; i++){
	CyBle_ProcessEvents();
	}
	//waiting for sleeper
	updateFlexSensors();
	while(flexA > threshold && flexB > threshold){
	CyDelay(100);
	updateFlexSensors();
	//sprintf(text, "Waiting ");
	DisplayMessage(text, sizeof(text));
	for(int i=0; i< 100100; i++){
		CyBle_ProcessEvents();
	}
	}
	sprintf(text, "Sleeping        ");
	DisplayMessage(text, sizeof(text));
	CyBle_ProcessEvents();
	//sleeping
	int k=0;
	for(int i=0; i < 10000; i+=100){
	updateFlexSensors();
	if(flexA > 500 && flexB > 500)
		k++;
	CyBle_ProcessEvents();
	CyDelay(100);
```

```
        }
          activateAlarm(1);
          updateFlexSensors();
          sprintf(text, "Wakeup!                   ");
          DisplayMessage(text, sizeof(text));
          CyBle_ProcessEvents();
          //waiting to wakeup
          int j =0;
          while(j<50){
                updateFlexSensors();
          if(flexA > 500 && flexB > 500){
                  activateAlarm(0);
                  j++;
          }
          else{
                  activateAlarm(1);
                  j=0;
          }
          CyBle_ProcessEvents();
          CyDelay(100);
          }
          activateAlarm(0);

          if(k<3)
          sprintf(text, "You slept peacefully");
          else
          sprintf(text, "You slept restlessly");
          DisplayMessage(text, sizeof(text));

          for(int i=0; i< 500; i++){
          CyBle_ProcessEvents();
          CyDelay(10);
          }
          }

}

/* [] END OF FILE */
```

User’s Manual:

Before going to sleep, set a time for when you want to wake up. When the set time has been reached, the pillow will begin to vibrate and its buzzer will go off, waking you up. To have the alarm turn off, your head must be off the pillow for 5 or more seconds. The pillow’s flex sensors will detect how many times your head leaves the pillow’s surface. If you have left the pillow 3 or more times, you will be considered unrestful.