Stress Detection and Management System

By

Yong Ho Kwon

Udara Cabraal

Hong Lee

Final Report for ECE 445

TA: Igor Federov

12 Dec 2012

Team #30

Table of contents

Abstract

Details in this document are accomplished work, and in progress with future work regards to the Stress Detection and Management System. It includes design, test procedures, results, and future works.

1. Introduction

1.1 Project Overview

1.1.1 Purpose

Certain amount of stress is necessary for our lives, but too much stress brings negative consequences such as decreases in level of concentration, mental health issues such as anxiety and depression as well as ineffective ways of coping, such as substance abuse. Most people do not know when and what situations they get stress from. Therefore, we are designing a system that can daily record a person's stress level and time and help the user with regulated breathing as a way of reducing their momentary stress. On the market, there are smart phone's apps where people can hold a finger to the camera, which will then detect slight changes in color related to blood flow. With these apps, users can purchase other apps that direct the user towards stress reduction techniques. These solutions, however, do not continuously record stress level. If a person able to recognize when they get stress and what they get stress from, it will be helpful for them to find ways to relieve it. There are also continuous stress detection devices, such as stress dots but they don't provide the user with a way of relieving there momentary stress. It is our intention to address these gaps in the market and create a system that will be of benefits to a great many patients and health care practitioners by better assisting them by taking control of an elevated physiological response that has many negative health consequence.

1.1.2 Objectives

Our goal is to create and combine a continuous monitoring device and stress management device into one system. Our continuous monitoring device will be responsible for monitoring the user's stress level, so that the user will be able to concentrate on his/her tasks throughout the day and be assured that stress levels are accounted for. We will also help the user regulate his breathing to relieve any stress that is detected.

1.1.3 System overview

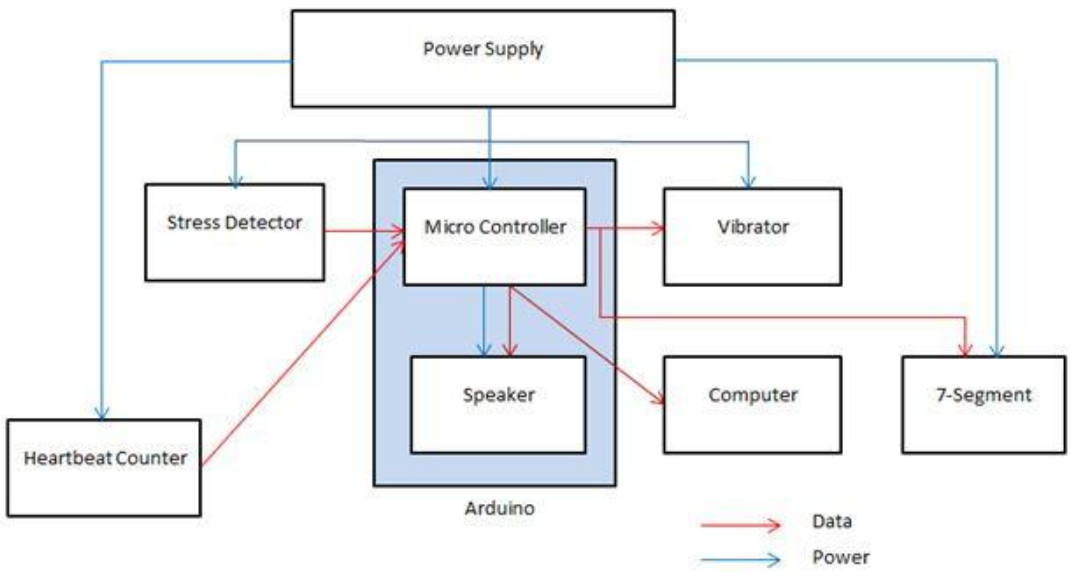


Figure 1: System overview block diagram

We are making two different devices for detecting stress and one controlling device. The first device is detecting changed color from stress dots and second device is detecting user's heartbeat rate. The stress dots use liquid crystals to detect any change in skin temperature. A decrease in blood flow from the extremities is a result from a stressed state. This decreased blood flow causes a temperature drop. Since stress dots, however, can get affected by room conditions, such as temperature, and may change the color, we built a heartbeat detecting device to give more accurate results to users. This is a small and portable device, so user can check their stress anytime and anywhere.

The controlling device, our Arduino, decides whether user is on stress level or not. If a stressed state is detected by the photodetecting circuit the vibrator will turn on, alerting the user of a stressed state. The second device detects heartbeat and it will be applied by the user to confirm that a stressed state is apparent, if the heartbeat is above 90 beats per minute the metronome will turn on, providing the user with means to regulate his breathing. When user is in the stress level, it will record the time that the stressed state is detected. Also, it sends heartbeat rate output to 7-segments and shows on the screen.

On the first device, the photodetector, when a user gets stressed the stress dot will change color and the LED light will magnify the changing of color of stress dots. The resulting color change will be detected by the photodiode. Each color has different voltage, so photodiode will send an output voltage dependent on the color to the microcontroller (Arduino). The microcontroller will send an output voltage to the vibrator to let the user know that he/she is stressed. Also, it will record a specific time and stress level inside the memory in microcontroller.

For more accurate results, we will built a second device, which detects stress by heartbeat rate. When the user's heartbeat is over certain amount number (90 beats per minute), it will let user know by rhythmic fixed interval metronome sounds. A microcontroller(Arduino) will activate the assisted stress management device, when elevated levels of stress are detected. The user can check their heartbeat rate anytime they want even though they are not on the stressed level. The microcontroller has USB connections so it can transport recorded data from device to user's computer.

2. Design

2.1 Design Procedure

2.1.1 Stress Detector

We use a photodiode and LED light to magnify the color of stress dots and detect the color. Stress dots change color when a user is stressed. We will use LED light to magnify the color and photodiode will detect the changed color. Since we use 9V battery, we will use voltage regulator to regulate to the 5v, to power the LED diode. When the photodiode detects different voltage levels depending on the color, it will send the analog output to the microcontroller.

2.1.2 Heartbeat Detector

We use TCRT5000L package to detect user's heartbeat. It has phototransistor and IR emitter together. The user should hold finger's tip on the top of TCRT5000L. When IR light hits the finger, a varying amount is reradiated depending on the presence of blood or a heartbeat. Reflected light from the finger is detected by the phototransistor. The current pulse from the phototransistor will be amplified and sent as output to the microcontroller. Every time a heartbeat is detected, a red LED will blink at the output of our circuit.

2.1.3 Controlling Device

Our controlling device contains a microcontroller (Arduino), RTC (Real Time Clock), 7-Segments, vibrator, and speaker. Microcontroller will be fully programmed with RTC and 7-Segments. Two of digital outputs will connect with vibrator and speaker.

2.2 Design Details

Power Supply

For our design, we are going to use an alkaline battery, which supplies 9V. Since the microcontroller's input voltage range is 7~12V, we decided that 9V alkaline battery is the most reasonable choice. Other hardware components in our circuit require 5V, so we stepped down the 9V battery with an LM7805 voltage regulator to 5V.

2.2.1 Stress Detector

The stress detecting circuit includes stress dots, an LED, and photodiode. When the user gets stressed, the stress dot will change color (green = relaxed, red = stressed). We will magnify the intensity of color by LED, so photodiode can detect changing color. Red has wavelength about 650nm with 0.225 V across our 1.2Mohm resistor and green has about 450nm with 0.164V (Results from color detection test). The photodiode has a breakdown voltage 120V, which is much larger than any voltage we will be providing. The photodiode analog output sends a signal to microcontroller's analog input. Since each analog input of Arduino has 1024 steps from 0 to 5V, both color voltages can be handled. The amount of voltage needed to operate the photodiode is 5V. However, our power supply provides 9V. Therefore, we are going to use a +5V voltage regulator (LM7805) to regulate the voltage.

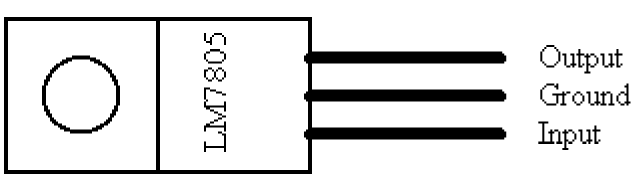


Figure 2:LM7805

2.2.2 Heartbeat Detector

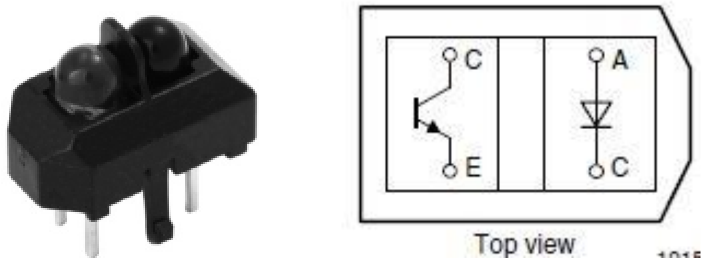


Figure 3: TCRT5000L

TCRT5000L is in a black leaded package which blocks all the visible lights and only can detects IR lights. The IR emitter and phototransistor required less than 5V to produce IR lights. The phototransistor's output should go to amplifier and amplify the signal. A Blood flow rate is directly dependent on the heartbeat rate. Finger skin is the thinnest skin in human's body. When the IR light is reflected from the finger, it has changed signal. It is very small signal to detect by microcontroller, so we amplified it twice. We used MCP602, which is dual op-amp. It required 5V to operate and pin 4 is going to input voltage and pin 8 is going to the ground. Both TCRT5000L and MCP602 get power from power supply. Since they only require 5V, the voltage regulator (LM 7805) would be required.

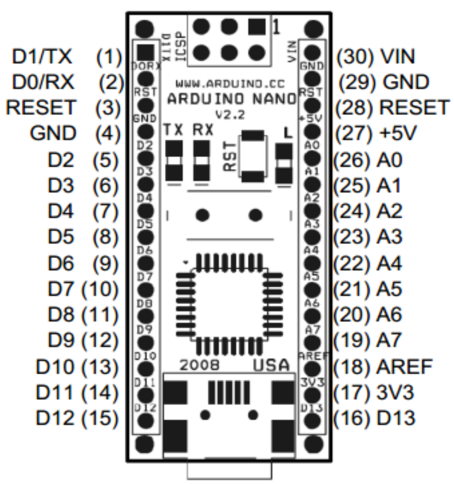
Figure 4: MCP 602

2.2.3 Controlling Device

a) Microcontroller

We are going to use "Arduino nano 3.0" for the microcontroller. It has operation voltage of 5V and input voltage 7~12V. There are total 14 digital input and output pins and six of them provide PWM (pulse width modulation) output. It has 8 analog input pins, but do not have any analog output pins. It has 40mA of DC current per input and output pins. It also has 32KB of flash memory with 2KB of SRAM and 1KB EEPROM, which we can read and write with the EEPROM library. It has clock speed of 16MHZ. When photodiode detects color change of stress dots, each color has different wavelength and voltage. According to our color detection test, red has about 0.225V and green has 0.164V. When it gets analog signals from photodiode, we need PWM digital outputs send signal to BJT (bipolar junction transistor) Since Arduino does not have analog outputs. At the same time, micro controller receives a voltage signal from the stress detector block and outputs data to the EEPROM. For the time, we are going to use RTC (real time clock), so microcontroller can record correct time on the memory. Microcontroller is the most important component in the device. It decides whether user got stress or not. When user got stressed, it turns on metronome and vibrator. It also sends heartbeat rate to the 7-segments and showing the average heartbeat rate to the user. Microcontroller can get power from power supply or from the computer through USB cable.

Arduino Nano Pin Layout



Pin No.	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A7-A0	Input	Analog input channel 0 to 7
27	+5V	Output or Input	+5V output (from on-board regulator) or +5V (input from external power supply)
30	VIN	PWR	Supply voltage

Figure 5: Microcontroller pin lay out

Metronome and vibrator needed analog output to turn on. Since Arduino only provides digital output, we used PWM (Pulse Width Modulation) technique to get analog result with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off. As it shown on Appendix C, AnalogWrite() is on a scale of 0-255, such that analogWrite(255) requests a 100% duty cycle always on, and analogwrite(127) is a 50% duty cycle , and analogWrite(0) is a 0% duty cycle.

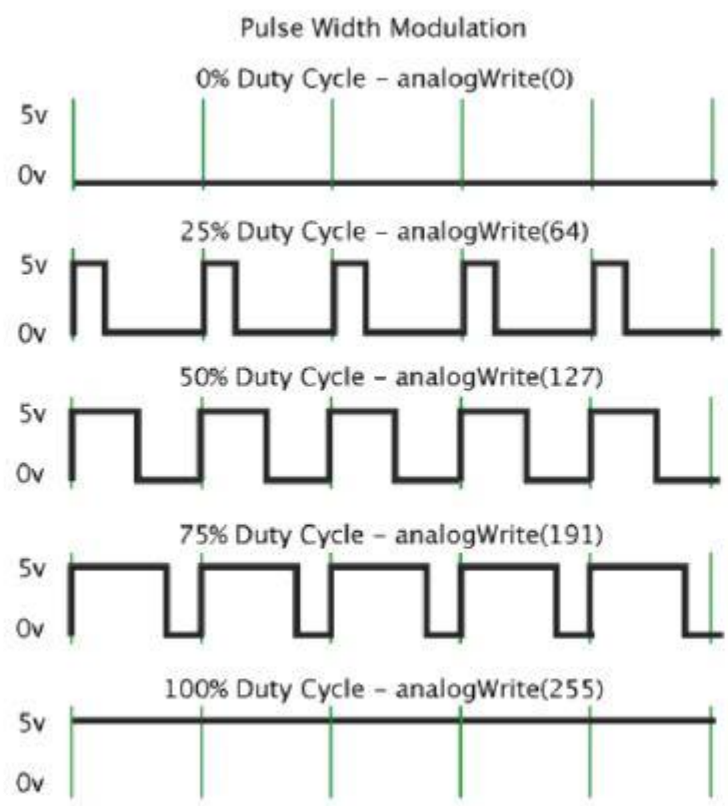


Figure 6: Pulse Width Modulation

b) Metronome

Metronome will be a program, which is a part of Arduino. We will output audio signal to speaker. If we use pulse wave, it will be a long sound with different amplitude. Therefore, we are going to use square waves to make 60bits/minutes, so it can have one bit per seconds. When user's heartbeat rate is on the stress level, it will turn on. Metronome gets power from microcontroller.

c) Vibrator

Vibrator motor needs operation input voltage 2.5~3.8V. Vibrator is on when first device detects the stress. Vibrator gets power from microcontroller.

3. Design Verification

Power Supply

The power supply was tested to make sure it supplied a regulated 5V from a 9V battery at varying loads. The 9V battery supplied the input power to the LM7805 voltage regulator. The output pin of the regulator was connected to a 150 ohm resistor. We varied the resistance in 150 ohm intervals up to a max of 1k ohms and we used a Digital Multimeter (DMM) to record the voltage across each new resistor. The voltage stayed at around 5V while the current varied from about 5mA to 50mA.

Output (V)	Resistance (ohms)	Current (mA)
5.06	150	33.7
5.06	300	16.9
5.07	450	11.3
5.07	600	8.45
5.07	750	6.76
5.07	900	5.63
5.07	1000	5.07

Photo-detector Circuit

The white LED was first tested. We supplied 5V to the anode of the LED and observed that sufficient intensity white light was being radiated with forward current. The complete circuit was then tested to make sure it provided voltages for Green and Red detection. We tested in a dark room and covered the photodiode sides, so that most of the light detected would be the light reradiated from the object above. The objects were placed 1mm above for each trial. The photodiode has a 1.2 Mohm resistor in series with it so that photo-current rise when a specific color is detected causes a higher voltage drop across the resistor. A DMM was used to observe the voltages across the 1.2 Mohm resistors. The average difference between the voltages of the two colors detected was 0.0618V, which is well within the 0.004V resolution of the Arduino inputs. Test results are included on Appendix D.

Heartbeat Detector and LCD

For the heartbeat detecting circuit we sent the output from pin 7 of the op-amp to the Arduino. Any time a pulse was detected by the LED in the circuit the Arduino also detected it. The LCD was tested by sending the beats-per-minute from the Arduino to the display. We found that the same beats-per-minute shown in the serial monitor of the computer was displayed by the LCD. The actual heartbeat was checked by counting the number of pulses on our wrist for sixty seconds. The counter was blind to the LCD display and we found that the number of counted pulses and that shown on the display only differed by an average of 0.43 beats-per-minute. Test results are included on Appendix D

Arduino

The Arduino was tested for correct functionality using a grounded input as a stressed state and voltage high (5V) as a relaxed state input. We wrote a program on the Arduino to store the time when the user is stressed, and in this measure the RTC was also tested to make sure it outputs the correct time to the Arduino. When we grounded the input, the time stored in the EEPROM was written to the serial monitor, showing that the Arduino was getting the correct time from the RTC and the correct data was being written and read. When we sent the input high, to simulate a relaxed state, nothing was being written into memory, proven by a blank serial monitor.

Next we tested the Arduino to make sure it provides the correct output to the vibrator and speaker. Again ground and 5V were used as inputs to simulate a stressed state and relaxed state respectively. We observed that as written in our program, the vibrator buzzed for 1sec with a wait time of 2 sec in between buzzes and our speaker made 10 beeps every second (to manage a fixed ratio of 4:1:5 sec in:hold:out breaths). This happened only when the input voltage went low, to simulate a stressed state.

Next we did the same tests, but used the output from our photodetector circuit as input to test the resolution of the Arduino. Changing the color in a dark room, showed the correct output, namely when the stressed state was detected the vibrator and speaker turned on behaving correctly and the correct time was read from memory and written to the serial monitor.

These tests were done with the Arduino powered by our 5V regulated voltage. Once the tests were run through the Arduino was again connected to the laptop through a mini USB to check for correct time storage.

4. Costs

4.1 Parts Cost

Name	Price	Quantity	Model #	Cost
Micro-controller (Arduino Nano 3.0)	$65	1	ATMEGA328	$65
LED (white)	$0.63	1	LED-156	$0.63
Photodiode	$2.14	1	VTP1188SH	$2.14
Stress Dots	$0.16	10	BC208	$1.6
Internal speaker	$20.3	1	PSP 2000	$15
Vibrator	$4.99	1	ROB-08449	$4.99
PCB	$33	1		$33
Op-Amps	$1.10	1	MCP 602	$1.10
Voltage Regulator	$6.8	4	LM7805	$27.2
BJT's	$0.08	1	2N3904G	$0.08
Push Button Switch	$5.99	2	IM206	$11.98
Mini B-USB cable	$2.29	1	GO27005	$2.29
Resistors (100 Ω)	$1.07	1	20J100	$1.07
Resistors (250 Ω)	$1.08	2	20J250E	$1.08
Capacitors	$0.15	6	P5178-ND	$0.9
TCRT 5000L	$1.08	1	SEN-0988	$1.08
LED (Red)	$1.50	1	COM-08862	$1.50
Op-Amp	$0.95	1	LM358	$0.95
RTC (Real Time Clock)	$14.95	1	BOB-00099	$14.95

Total	$163.28			$186.54

4.2 Labor Cost

Name	**Hourly Rate**	**Total work hours**	**Salary**
Yong Ho Kwon	$20	100hr	5,000
Udara Cabraal	$20	100hr	5,000
Hong Lee	$20	100hr	5,000
Total	$60	300hr	15,000

4.3 Total Costs

	Cost
Labor	$15,000
Parts	$186.54
Total	$15,186.54

5. Conclusions

5.1 Wrap-up and future work

We could fulfilled the all the requirements and verifications that we proposed. We could successfully magnify the color changed from stress dots. We could detect accurate heartbeat rate of user. We could send output signals from each device to microcontroller, and it could store the right current time. When stress is detected, metronome and vibrator was turn on. As we designed, device is small and portable size, so user can carry it around to anywhere at any time.

Our future work for this project would be using wireless communication between each device. Instead of using wire, we think it will be much useful to use wireless communication to send output signal to controlling device. To get more accurate results from first device, we should find a better way to isolate the photodiode from LED light, since photodiode also get interfered by other visible lights besides LED.

5.2 Ethical considerations

IEEE Code of Ethics	Considerations
1. To accept responsibility in making decisions consistent with the safety, health and welfare of the public, and to disclose promptly factors that might endanger the public or the environment.	We used 9V battery for power supply. Explosion might be a problem for battery. Therefore, through the test, we found the power for each devices and supply power to devices correctly. Hard case for LED and photodiode will protect user from burning risk.
3. To be honest and realistic in stating claims or estimates based on available data.	In order to get data, we did several tests and our design is based on this data.
5. To improve the understanding of technology, it's appropriate application, and potential consequences.	Our design utilizes several devices with micro-controller. Through the separate tests, we checked that the devices successfully implement with microcontroller and then we combined all devices together and check to accomplish the task
6. To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations.	We do not have experience to create devices ourselves. However, we tried to apply our knowledge that we learned in other ECE courses, and we tried to improve our design through researching and learning from other sources.
7. To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others.	Our group adheres to the IEEE Code of Ethics and in order to get best result, we did feedback and criticism each other through peer review and design review.
8. To treat fairly all persons regardless of such factors as race, religion, gender, disability, age, or national origin.	All team members involved with this project with the same respect and we respect and shared other member's works to improve our project.
10. To assist colleagues and co-workers in their professional development and to support them in following this code of ethics.	Even though each member charged different parts of project, all members understand of the whole project and communicate and share knowledge each other for our professional development.

6. Reference

1) *"*VTP118SH, VTP Process photodiodes", *PerkinElmer Optoelectronics.*

   http://datasheet.octopart.com/VTP1188SH-EG-datasheet-56243.pdf

2) *"*ROB-08449, Vibration motor*", Sparkfun Electronics.*

   https://www.sparkfun.com/products/8449

3) "Stress dots", *Harold Taylor*.

   https://www.taylorintime.com/index.php?page=shop.product_details&flypage=flypage-ask.tpl&product_id=163&category_id=10&option=com_virtuemart&Itemid=100123

4) "Low Pass Filter", All About Circuits.

   http://www.allaboutcircuits.com/vol_2/chpt_8/2.html

5) "TCRT5000", Vishay Semiconductor.

   http://www.vishay.com/docs/83760/tcrt5000.pdf

6) "Arduino Nano", Arduino.

   http://arduino.cc/en/Main/ArduinoBoardNano

7) "5mm Super Bright White 24000mcd", UniqueLEDs.com

   http://www.unique-leds.com/index.php?target=products&product_id=1862

8) "Single Supply CMOS Op Amps, MCP602", Microchip.

   http://ww1.microchip.com/downloads/en/DeviceDoc/21314g.pdf

9) "Positive Voltage Regulator, LM7805", Texas Instruments.

   http://ww1.microchip.com/downloads/en/DeviceDoc/21314g.pdf

Appendix A: System Flow Charts

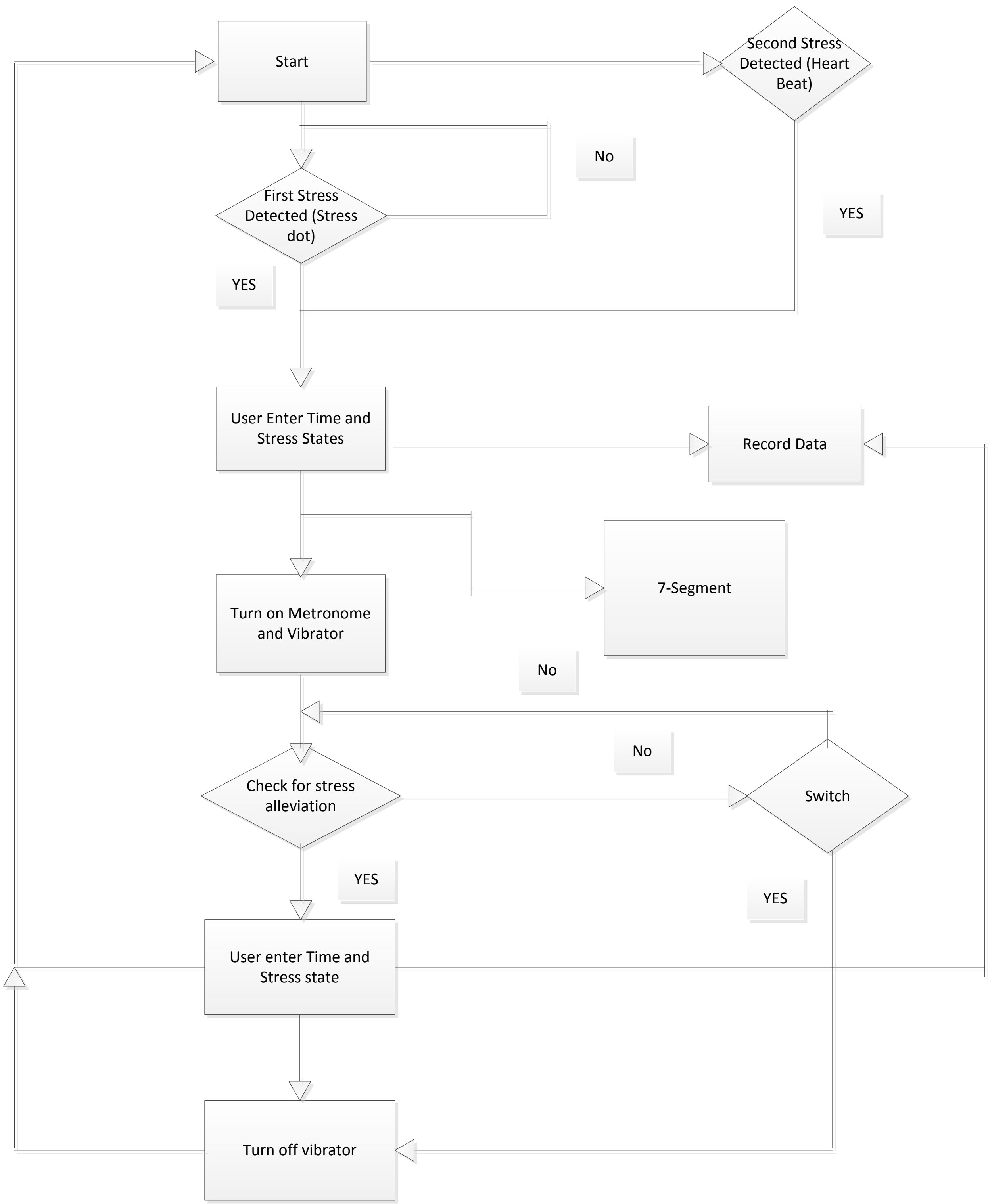


Figure 7: System Flow Charts

Appendix B: System Schematics

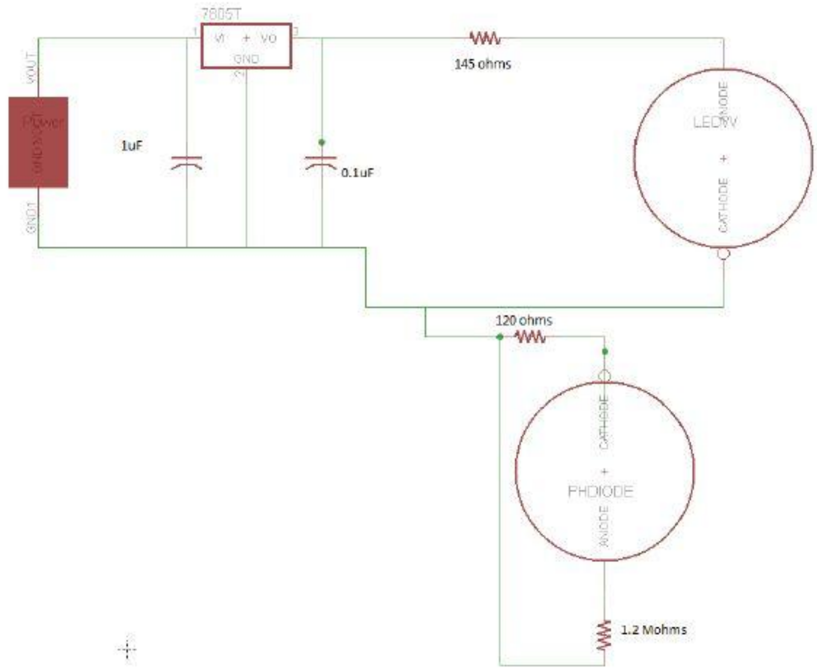


Figure 8: Schematics for First Device (Stress dot)

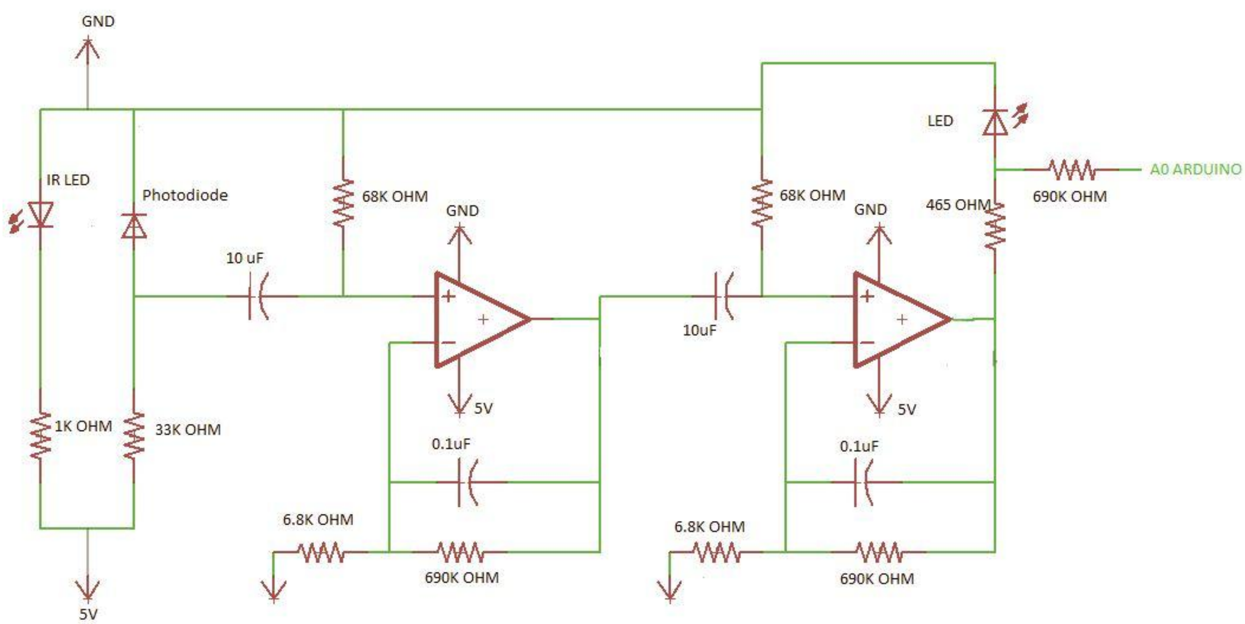


Figure 9: Schematic for Second Device (Heartbeat)

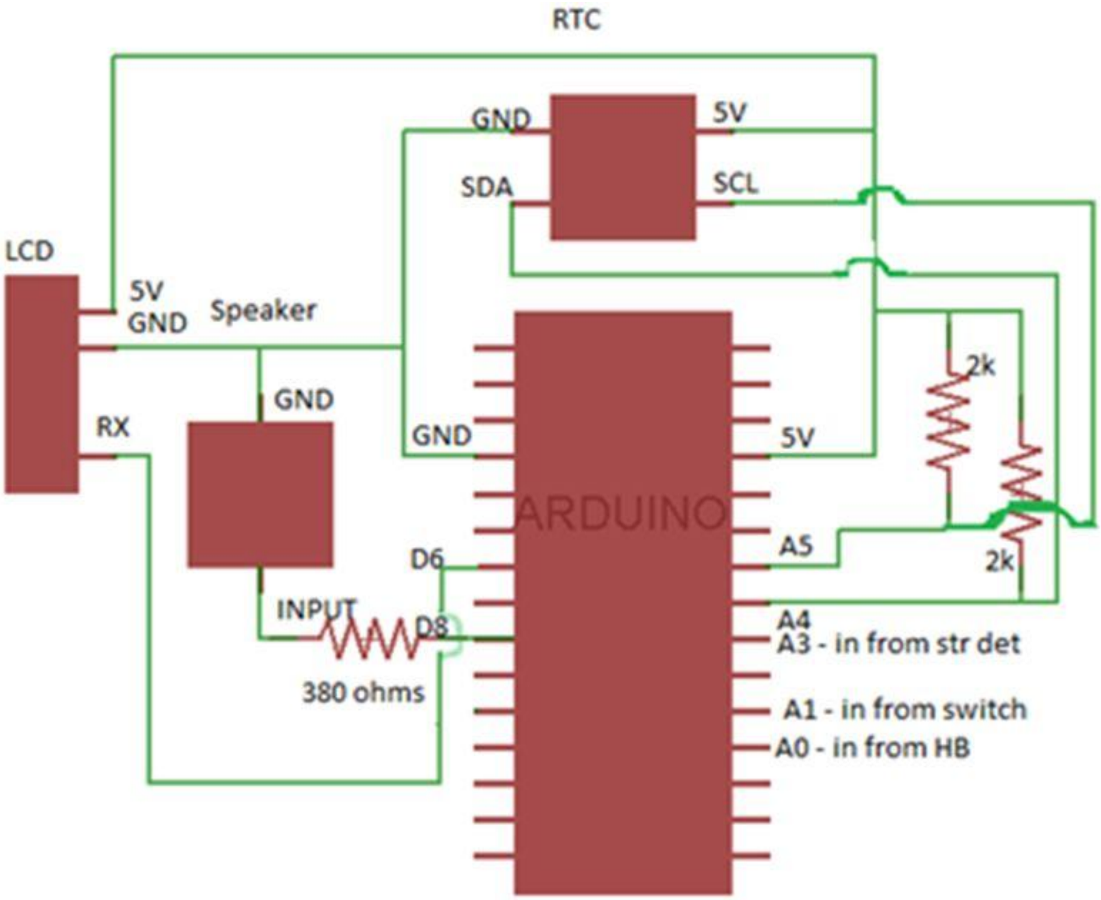


Figure 10: Schematic of Controlling Device

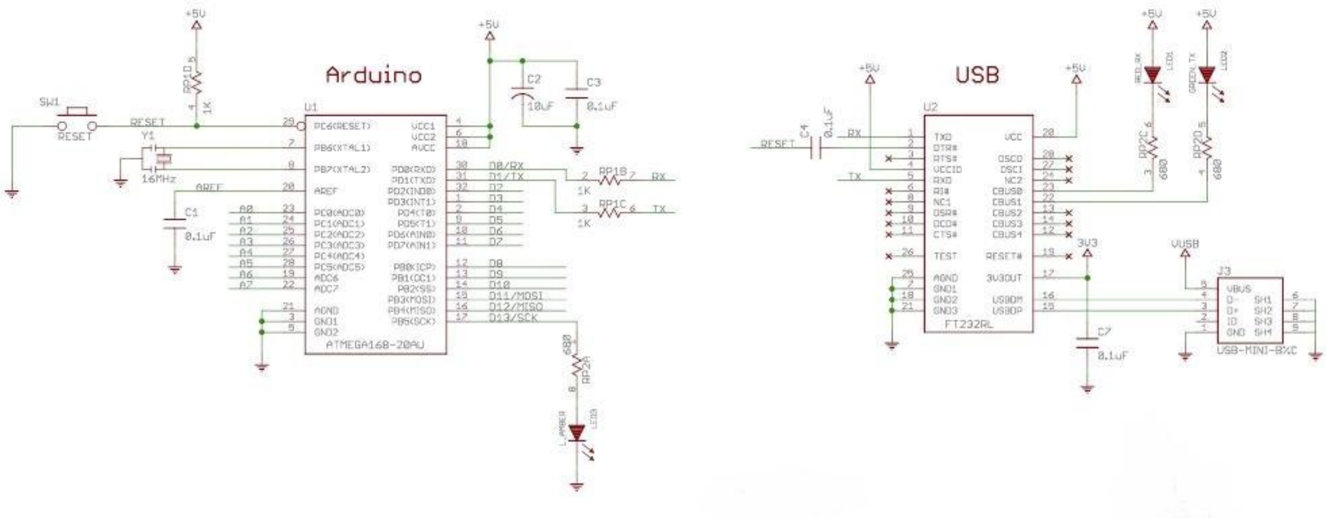


Figure 11: Schematic for Microcontroller

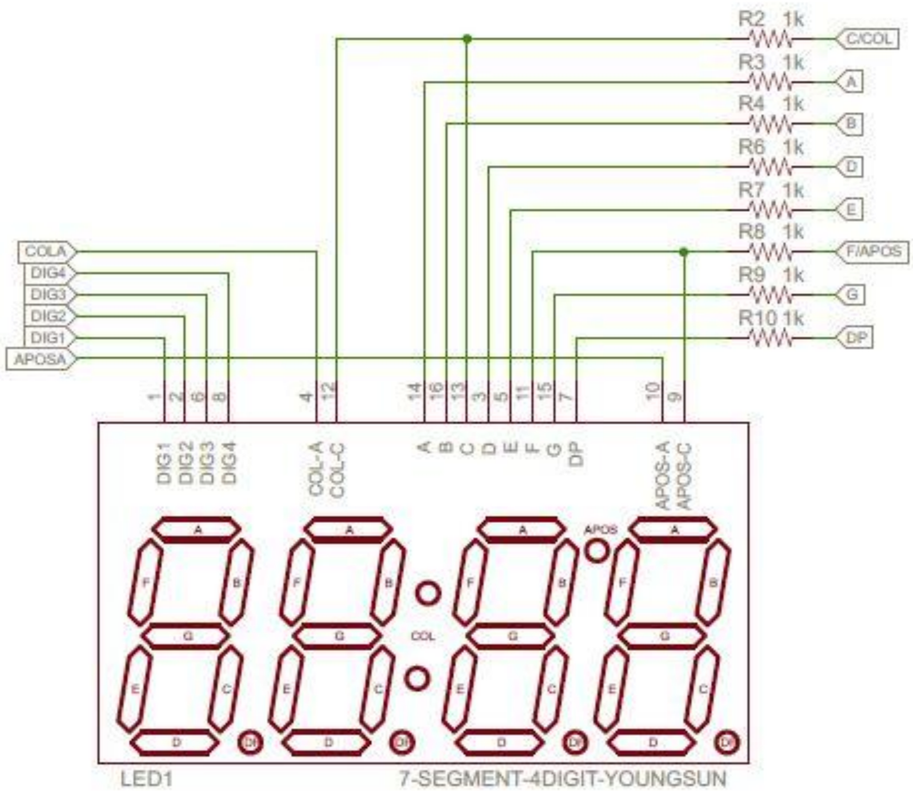


Figure 12: Schematic for 7-Segments

Appendix C: Software Implementation

```
#include "pitches.h"
#include "Wire.h"
#include <EEPROM.h>

#define DS1307_I2C_ADDRESS 0x68

int VibratorPin=9;
byte decToBcd(byte val)
{
return ( (val/10*16) + (val%10) );
}

byte bcdToDec(byte val)
{
return ( (val/16*10) + (val%16) );
}

void stopDs1307()
{
Wire.beginTransmission(DS1307_I2C_ADDRESS);
Wire.send(0);
Wire.send(0x80);
Wire.endTransmission();
}

void setDateDs1307(byte second, // 0-59
byte minute, // 0-59
byte hour, // 1-23
byte dayOfWeek, // 1-7
byte dayOfMonth, // 1-28/29/30/31
byte month, // 1-12
byte year) // 0-99
{
Wire.beginTransmission(DS1307_I2C_ADDRESS);
Wire.write(0);
Wire.write(decToBcd(second));
Wire.write(decToBcd(minute));
Wire.write(decToBcd(hour));
Wire.write(decToBcd(dayOfWeek));
Wire.write(decToBcd(dayOfMonth));
Wire.write(decToBcd(month));
Wire.write(decToBcd(year));
Wire.endTransmission();
}
```

```
void getDateDs1307(byte *second,
byte *minute,
byte *hour,
byte *dayOfWeek,
byte *dayOfMonth,
byte *month,
byte *year)
{
Wire.beginTransmission(DS1307_I2C_ADDRESS);
Wire.write(0);
Wire.endTransmission();
Wire.requestFrom(DS1307_I2C_ADDRESS, 7);

*second = bcdToDec(Wire.read() & 0x7f);
*minute = bcdToDec(Wire.read());
*hour = bcdToDec(Wire.read() & 0x3f); // Need to change this if 12 hour am/pm
*dayOfWeek = bcdToDec(Wire.read());
*dayOfMonth = bcdToDec(Wire.read());
*month = bcdToDec(Wire.read());
*year = bcdToDec(Wire.read());
}
int addr = 0;
byte value;
void setup() {
byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
Wire.begin();
Serial.begin(9600);
}

void loop() {

int StressD = analogRead(A0);
int SW = analogRead(A1);
float StressD_voltage = StressD * (5.0 / 1023.0);
float SW_voltage = SW*(5.0 / 1023.0);
byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;

getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
while (SW_voltage < 1)//switch off
{
StressD = analogRead(A0);
SW = analogRead(A1);
StressD_voltage = StressD * (5.0 / 1023.0);
SW_voltage = SW*(5.0 / 1023.0);
byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
if (StressD_voltage < 0.23) // check for stress state
{
```

```
EEPROM.write(addr, hour);

value = EEPROM.read(addr);
Serial.print(addr);
Serial.print(":");
Serial.print(value, DEC);
addr = addr + 1;
EEPROM.write(addr, minute);
value = EEPROM.read(addr);
Serial.print(":");
Serial.print(value, DEC);
addr = addr + 1;
EEPROM.write(addr, second);
value = EEPROM.read(addr);
Serial.print(":");
Serial.println(value, DEC);
addr = addr + 1;
if (addr == 24)
addr = 0;
delay(1000);
int VibratorValue= 240;
analogWrite(VibratorPin, VibratorValue); // Metro code
delay(1000);
VibratorValue= 0;
analogWrite(VibratorPin, VibratorValue);
int melody[] = {
NOTE_C4, NOTE_C4,NOTE_C4, NOTE_C4, NOTE_C4,NOTE_C4, NOTE_C4,
NOTE_C4,NOTE_C4,NOTE_C4,NOTE_C4,NOTE_C4,NOTE_C4,NOTE_C4,NOTE_C4,NOTE_C4,NOTE_C4,};
int noteDurations[] = {
4, 4, 4, 4,4,4,4,4,4,4,4,4,4,4,4,4,4,4 };

for (int thisNote = 0; thisNote < 8; thisNote++) {
int noteDuration = 1000/noteDurations[thisNote];
tone(8, melody[thisNote],noteDuration);
int pauseBetweenNotes = noteDuration * 3;
delay(pauseBetweenNotes); }

if (StressD_voltage >0.25)
{ Serial.print("relax");
delay(300);
Serial.print(":");
noTone(8); //metronome}
if (SW_voltage > 1)
{ /*Serial.print("Sw_On");*/
noTone(8); } } }
noTone(8);
}
```

Appendix D: Requirement and Verification

Module Requirements	Verification	Status (Y or N)
Power Supply I. The battery providing power to the system must output 9 V.	A DMM will be used to check the output voltage of the batter to ensure it outputs 9 volts.	Y
Photo-detector Circuit I. LED is on while power is supplied II. Photodiode generates more current when it detects the color Red, from the stress dot, than when it detects Green. III. The photo-diode circuit outputs a voltage for one color that differs from the second color by at least 0.01 V, with a tolerance of 5%.	The photo-detector circuit will be tested with a green object and red object (not the stress dots). In a dark room the color objects will be held at a fixed height of 1cm from the lit LED and photodiode. Verification II explains the measurements that will be recorded. This test will be done for ten trials. We will then do the same test with the stress dots. The stress dots will be attached to a glass containing temperature controlled water. The water temperature will be adjusted by placing the glass on a hotplate. The changing temperature of the water will cause the color of the stress dot to change, and the colors detected will include red and green. Once we see either red or green, verification II will be followed to record data. I. The LED will be provided 3.4 V from the power supply via a voltage divider, with 10mA	Y

	current. II. The photodiode will be placed in series with a 1.2 Mohm resistor. The voltage across the resistor will be measured with a DMM. A higher photo-current generated by the photodiode will correspond to a higher voltage across the resistor. III. The measurements observed in verification II will be analyzed (by calculating the difference between the voltage levels of each trial) to ensure requirement III is met.	
Arduino I. Differentiates between stressed state voltage and relaxed state voltage. II. Stores time and state of user III. Outputs PWM signal to vibrator module and a square audio signal at 1Hz to a speaker when stress is detected. Subsequently turns off	I. Write a test program that gives voltage inputs differentiated by 0.01 V and send the stress state as an output stream to the monitor. II. Use the write() function to write a sample time and stress level to	Y

both signals when user enters sleep mode or relaxed state is detected.	EEPROM and use read() function to read from memory and send the data to the monitor through an output stream. III. Write a test program that provides a voltage level input that corresponds to a stressed state and then a relaxed state voltage level input after ten seconds. The output pins of the Arduino that would be connected to the vibrator and speaker will be connected to an oscilloscope. The requirement will be verified if the correct PWM signal is outputted to the scope from the output pin to the vibrator and a 1Hz square wave from the output pin to the speaker.		
Heart Beat Detector I. TCRT5000L send changed signal to Op-Amp II. Every time when heartbeat is detected, red LED light will be blinks	I. Build a complete circuit with TCRT 5000L with dual –op amp (MCP 602). II. Find a right capacitance value that makes red	Y

III. Successfully send amplified output to controller from	LED lights blinks every time heartbeat is detected. III. Connect with microcontroller and 7-segment to send the output signal. IV. User hold finger on the TCRT5000L to run the heartbeat test.		
7- Segment Heartbeat rate is shown on the screen.	I. Program inside the microcontroller. II. All the wires are labeled correctly	Y	
Vibrator Vibrator is on when stress dot changed the color.	I. Vibrator gets power from microcontroller. II. Program the microcontroller so vibrator get turn on when it is over certain voltage level.	Y	
Metronome Metronome is on when heartbeat rate is on the	I. Metronome gets power from microcontroller. II. Program the microcontroller so it get turn on when heartbeat is over certain level.	Y	

Color Detection Test Results

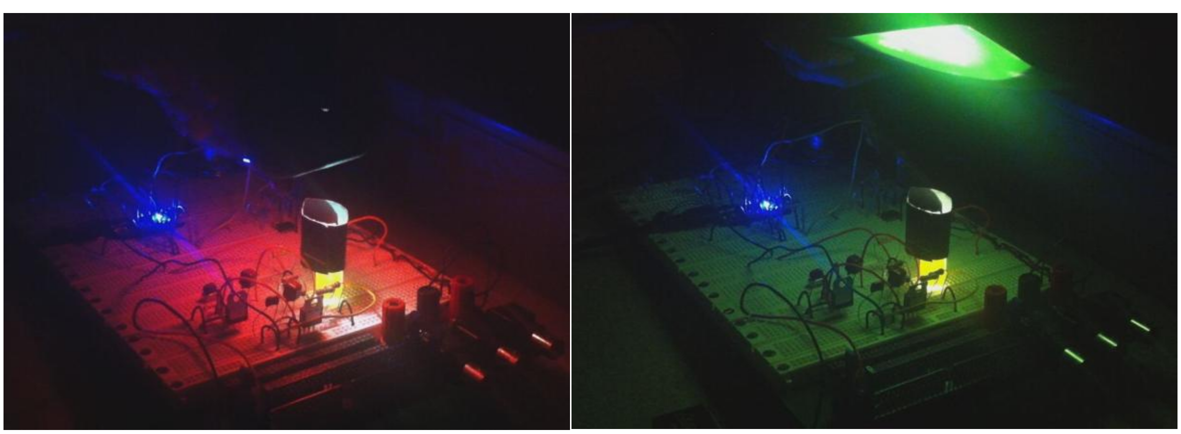


Figure 13: Stressed on Red and Relaxed on Green

Trial	Green	Red	Difference
1	0.161V	0.229V	0.068V
2	0.168V	0.225V	0.057V
3	0.164V	0.226V	0.062V
4	0.165V	0.228V	0.063V
5	0.168V	0.227V	0.059V

Heartbeat Test Results

Trial#	From Detector	Actual Heartbeat
1	65	69
2	63	64
3	59	61
4	69	62
5	72	61
6	62	69
7	66	73
Ave	65.14	65.57