Moodmusic Report

By: Anthony Lam and Sally Kim
Date: December 20, 2019

Abstract

This project will consist of a wearable hardware device that can read a user's mood and in turn select and play music that fits the current mood expressed by the user. This device would resemble a pair of over-the-head wired headphones with some additional electrical components that would act as the "mood" measurer. In this device, the mood will be determined by having sensors to detect changes in the person's environment and biometrics in which the data is categorized into a range of moods a person can exhibit.

Project Concept

Moodmusic Headphones resemble the traditional over-the-head headphones that exist today. However, they offer an additional feature that automatically selects a playlist of music for the user based on their "mood". The headphones will have added sensors to the typical structure that allow the device to monitor and determine what the mood of the user will be. The device's measurements will consist of both biometrics and the external environment that the user is currently in. The biometric measurements include the user's heart rate, surface level temperature of their skin, and their range of motion. External environmental factors that are taken into account are the surrounding temperatures and the surrounding light intensity. From these measurements that are read by the device, it will parse the information so that it can determine a "mood" that the user is feeling based on an algorithm.

Some pre-existing ideas operating in a similar or partial manner of our project include: Mood Music Player by FriQDevelop, this is an application a user can download to play music based on a pregenerated mood option; Habu Music by GravityMobile, similarly the user can choose a pre-generated mood option based on the user's inputted songs; and Mood Agent by Moodagent, a slightly different application in which the user chooses their mood range the music is directly streamed to their device matching that mood. Currently, the options for mood-based music is very limited to either the user entering their own mood that they may be feeling at the moment or rely on a pre-generated playlist which is based on data collection and analysis. This shows that there is an opening for a device that is capable of accurately reading and defining a person's mood and from that picking a playlist that is pre-generated. Although the playlist is still a pre-generated one created by the user, the mood analysis portion of the device is all done automatically rather than a user picking a playlist based on the mood that the playlist describes or the user describing their own mood for the application to find music for them.

Part Selection

There are four key parts of this device revolving around the use of sensors. The device will need a part that measures light intensity, a sensor to evaluate the movement of the user, a sensor to determine the heart rate of the user, and some temperature sensors to read surface skin of the user and the surrounding temperature. Additional parts needed would be a Raspberry Pi Zero to control the sensors and a part to assist in outputting sound.

Motion/Movement Sensor:

Ideal specifications of the sensor and what it should cover:

The speed of the person should go from 0 to 10 mph (~4.5m/s) at normal speeds as it should be able to measure when the person is still and when the person is either biking, running, or walking. The range for the accelerometer should be around 0-5gs (9.8*2 m/s max) because the speed a person is traveling should not be very high and the average max speed of biking is around 5 m/s and the minimum should be measurable as to when a person is still. However, it should also be able to adjust for higher speeds for when a person is traveling in a vehicle like a car or a bus. So having options available for 8 or 16 gs would be preferable. Additionally it would preferable for the sensor to be less than 10Hz of frequency (Low Vibration). It should also be a 3-axis accelerometer to detect the force of gravity/motion in all 3 directions.

The types of accelerometers that exist include Capacitive Micro-Electro Mechanical Systems (MEMS) which run on changes to the capacitor when movement is measured, perfect for low vibration, motion, and steady state acceleration. The limits of MEMS include a 200g limit, poor signal strength, and limited bandwidth. Piezoresistive Accelerometers provide a signal when the resistance changes due to motion. The pros of this type is the large range, the high bandwidth (for high frequency, short time, shock events), and highly accurate values. The downsides include, inaccurate vibration measurements, low sensitivity, high temperature sensitivity, and high cost. Piezoelectric Accelerometer provide an electric signal when acceleration is detected. These are generally good for most vibration measurements due to their wide frequency response, good sensitivity, and easy installation.However, these do not measure static vibration and require a charge amplifier. Other devices can measure motion, such as IR, but these are not suitable for measuring the speed of the user. It is also under the assumption that these accelerometers are all under breakout boards.

The most suitable types of accelerometers that fit the Pi device would consist of capacitive MEMS rather than Piezoresistive or Piezoelectric ones. Although they all are capable of measuring general human motion, CMEMS is more suitable. An example of one is the ADXL345 3AXIS, 17.5$, from Digikey, runs on 3.3V-5V, a variable sensing range from 2-16gs. A digital version of the breakout board would also be ideal for easier use.

Table 1
Accelerometer

Parts	Operating Voltage	Sensing Range	Price	Type of measurement
ADXL345	3.3-5V	2-16gs (2,4,6,8,16)	17.5	MEMS
ADXL345BCCZ	2-3.6V	2-16gs (2,4,6,8,16)	7.33	MEMS
KX023-1025-FR	1.8-3.6V	2-8gs (2,4,6,8)	2.4	MEMS
ADXL335	1.8-3.6V	3gs	9.95	MEMS

This accelerometer breakout board was chosen because it fulfilled the requirements that were needed. The sensing range was large enough to cover both still and moving moments and be able to determine the difference. Additionally, rather than an embedded device that would need to be soldered onto a board, the ADXL345 is already a breakout board, allowing for easier use and testing.

Temperature Sensor

Ideal specifications of the sensor and what it should cover:

The purpose of temperature sensor for our specific project is to detect and measure both the ambient temperature and the body temperature of the user. In order for the sensor to accurately measure the temperature of the user's surroundings, it needs to be exposed to the same environment as the user's. And depending on where the user brings our device to use it, the temperature sensor must be able to measure a wide range of temperature. Also, in order to measure the user's body temperature, it needs to be in contact with the skin. For the sensor to accurately measure the body temperature without irritating the skin of the user too much, it must also be flat and small.

The temperature sensor, along with other sensors, will determine whether the user is sitting down calmly or involved in heavy physical activities or exposed in a cold/hot environment because the values that the sensor will output will fluctuate depending on their movements and also their environments in which they're in.

Table 2
Temperature sensor

Parts	Operating Voltage	Temperature Range	Accuracy of measurement	Price
LilyPad temperature sensor	2.3-5.5V	-40 - 125C	+/-2 (0- 70C)	3.50
MAX31820	3.0	-55 - 125C	+/-0.5 (10 - 45C)	1.95
DS18B20	3.0-5V	-55 - 125C	+/-0.5 (-10 - 85C)	3.95

DS18B20 was chosen over other sensors primarily because it was easy to install and measure a wide range of temperature. As it is shown on the table, it measures from -55 to 125C and unlike MAX31820, in which its accuracy significantly drops (.5-2C) when it goes over 40C, DS18B20 doesn't. Also, Lilypad temperature sensor is not chosen because it is a breakout board which is more difficult to install.

Light Sensor

Ideal specifications of the sensor and what it should cover:

The purpose of light sensor for our specific project is to detect and measure external environment luminosity. These measurements will allow us to know and determine how bright or dark the user's surrounding is. This sensor must be exposed to the same environment as the user as well to accurately collect data. It should not be enclosed in a case or covered up by any material as possible.

This sensor will determine not only it is day or night, but also whether the user's in a fairly bright or dark environment. Since light has a significant effect on the user's emotions, this is a critical measurement that needs to be collected along others.

Table 3

Parts	Operating Voltage	Sensitivity Wavelength Range	Rise/Fall	Price
LilyPad light sensor	2.5-5.5V	390-700 nm	110/ 220 ns	3.50
VEML6030	2.5-3.6V	400-700 nm	300/1000 ns	4.95
photoresistor	Up to 100V	400-700 nm	55000/20000 ns	0.95

The photoresistor was chosen over other sensors mainly due to its cheap price and a decent function it can do as other sensors can do also. Also, compared to Lilypad and VEML6030, it is not a breakout board and is very simple to install and wire it (only two pins).

Heart Rate monitor

Ideal Specifications of the sensor and what it should cover:

Ideally this device would run on a 3.3V or 5V voltage due to the power capabilities of the Raspberry Pi itself. It should be able to detect heart rate in a place near or on the ear or somewhere on the head because the design plans on putting the sensor near the ear. The sensor also needs to be capable of measuring resting and moving heart rate and differentiating them because motion and resting periods are crucial in the algorithm to determine mood.

There are two main types of sensors that exist to measure a person's heart rate. The first one is one where the user wraps their finger in a band or probe that collects the data and send it to a device. The other is a plate sensor that is placed on the person's body; preferably on a finger, earlobe, or on the side of the throat. The first choice is not optimal because its not practical to have a wire connecting the device that must be attached to a finger of the user to the head where the computations and measurements are done. Therefore the ideal sensor is one where it consists of a plate that is able to detect heart beats of the user. Additionally looking for non-overlapping ports required to setup the heart rate sensor.

Table 4:

Heart Rate Monitor

Parts	Operating Voltage	Type of Return	Communication Protocol	Price
MAX30100	1.8-3V	Digital (Board)	I2C	10-12
Gravity Heart Rate Monitor	3.3-6V	Analog + Digital	SPI	16
MAXREFDES98	2-5.5V	Analog	I2C	16
SEN-11574	3-5.5V	Analog	SPI	25

This part was selected because of the simplicity of the setup because all of the choices were plate based sensors that allows the user to rest the device is varying locations for pulse measurements. Additionally, it was preferred that the sensor run through an SPI communication protocol because of the possible conflict for the I2C bus.

Part Evaluation

From the testing of our procured equipment, it was shown that the sensors were able to provide the use intended and needed. The accelerometer was required to read the movements of the user from a sitting or motionless state with some small variance in movement from the user, as well as the states in which the user is moving, either walking or moving and calibrate a mode after a period of time of data collection. The accelerometer was accurately able to detect small movements of the device as well as the larger one, so application wise, it completed the necessary benchmarks.

The photoresistor's main requirement was sensibility of a suitable range of light intensities. In a normal case scenario, the photoresistor would need to sense when the user was in a light void location versus when the user is outside in the bright sunlight. The photoresistor was capable of returning decimal values in which determined the level of light intensity, ranging from 10-40 in an indoor setting with the lights on, 0-15 in a dark environment where the lights were off or the sensor was covered, and above a 100 rating when bright light was shone on it.

The heart rate sensor's main requirement was to be able to detect a heartbeat through the skin of a user. After hooking up the sensor to the users hand, the pulse was detected, however it's difficult to read continuously, especially when the device will be placed in a location where the pulse is not the strongest. The pulse is also not going to be pressed onto the skin of the user at all times as the user may be moving or the headphone might not be placed on the ideal way. This means the data being read by the sensor may be a bit skewed, but the testing for reading a BPM was successful.

The temperature sensor's main requirement was detecting the temperature around it. It was able to accurately determine ambient room temperature as well as detect the increase and decrease of the temperature on the head of the sensor after placing and removing our fingers from the device. The device was also able to be configured to the Pi and return decimal values back to the terminal.

Design

(Same content as presentation, write it out)

The main proof of concept is a wearable device that imitates the traditional over-the-head headphones but with the additional of automation. The device will be able to read the user's "mood" based on biometrics and the surrounding environment. The sensors used in this device include: a heart rate monitor; two temperature sensors, one for ambient temperature and one for the user's; an accelerometer to measure the movements; and a photoresistor to measure light intensity. Additional components include a two-channel ADC and a stereo decoder.

The kiCAD circuit design, shown in Appendix: Figure 1, is composed of our sensors and additional components. It shows the connections between all of the components to the Raspberry Pi Zero. The figure shows the connections of the two temperature sensors, Figure 4, the stereo decoder, Figure 6 , and the accelerometer, Figure 3, as digital inputs into the Pi, directly connecting them. The accelerometer uses the I2C communication protocol, connecting the SDA/SCL lines on the pi to the part. The stereo decoder and the temperature sensors both use the regular GPIO pins on the Pi, where the decoder uses GPIO pins 18,19, and 21 which is based on the tutorial setup and the temperature sensor uses 17, which was selected over the default pin of 4. For the analog parts of the schematic, Figure 2 depicts the ADC setup through an SPI communication protocol with the Pi and the heart rate monitor, Figure 5, and photoresistor, Figure 8, having channel connections to the ADC of 0 and 1, respectively.

The software library shows the code for 5 components of the device. The first component is the MCP3002 class. This allows for the component to be instanced when another sensor requires the use of the ADC module. This class is referenced from heart rate monitor's source code, but additionally references Interfacing Electronics Lab 4 and Input and Output Lab 2 for use and setup of the SPI settings and the photoresistor. The photoresistor uses this class by the READ function provided by the class, taking in the channel that the photoresistor is using as a parameter and returning decimal values. The heart rate monitor has its own class instancing, but also references the MCP3002 class for initialization. The code here is taken directly from the GitHub repository stated in Attributions. This was because it was just easier to use already working and compiled code that someone else used instead of trying to figure out how the sensor is working and the data that is being transmitted. The temperature sensor referencing code snippets from a blog post on how to setup multiple 1-wire buses to a single gpio. The code used essentially activates the temperature modules and then extracts the information directly from the file location and parses it into a file reader, returning decimal values to the device. The speaker is mainly setup through an installation feature that Adafruit provides and the software portion is mainly for testing whether the speaker works (outputs white noise).

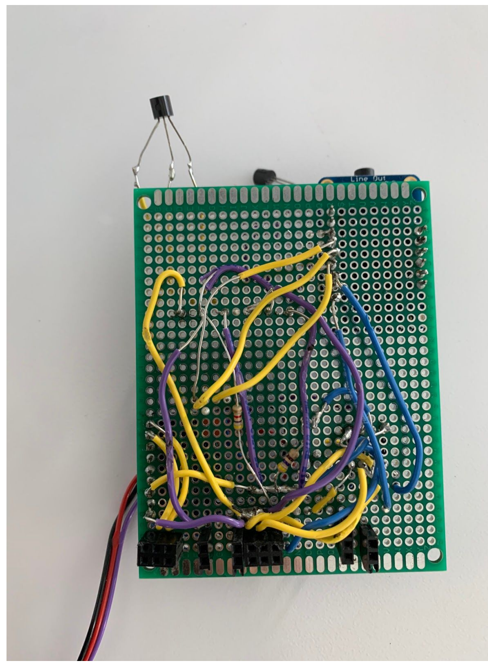
The PCB design (files can be found in the "docs" folder) was created first by loading the Netlists from the kicad schematic. Then, all the appropriate wires were connected together (sensor to itself and sensor to pi) including 3.3V power, GND, and data lines (GPIO pins). All the GND wires were connected on the back side of the PCB where it has a uniform ground plane.

Also, it was made sure that the wires don’t overlap each other and also not have long wiring as possible since long traces may pick up interference and add noise.

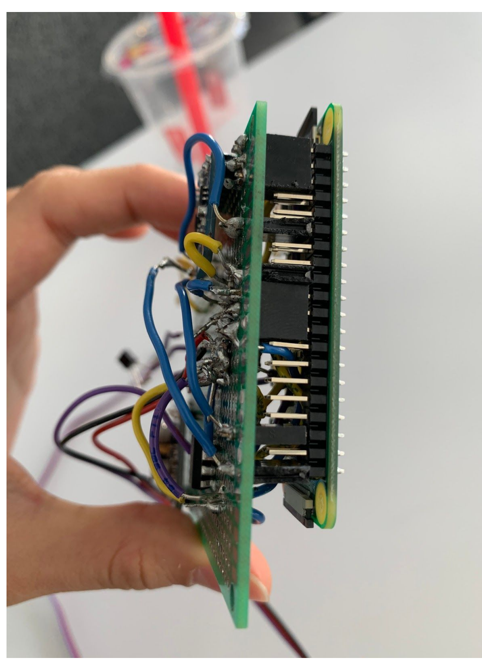
Prototype

Perfboard Connected to the Raspberry Pi Zero

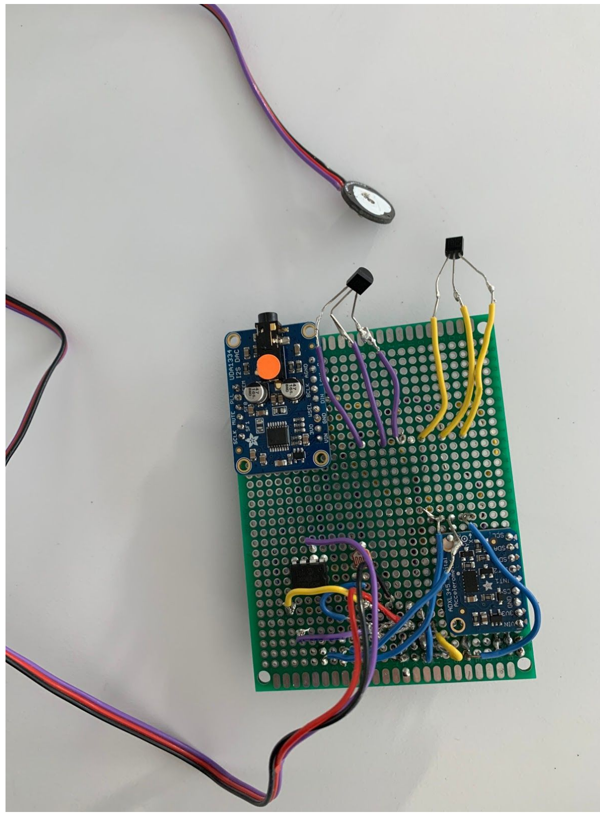
Bottom of the Perfboard



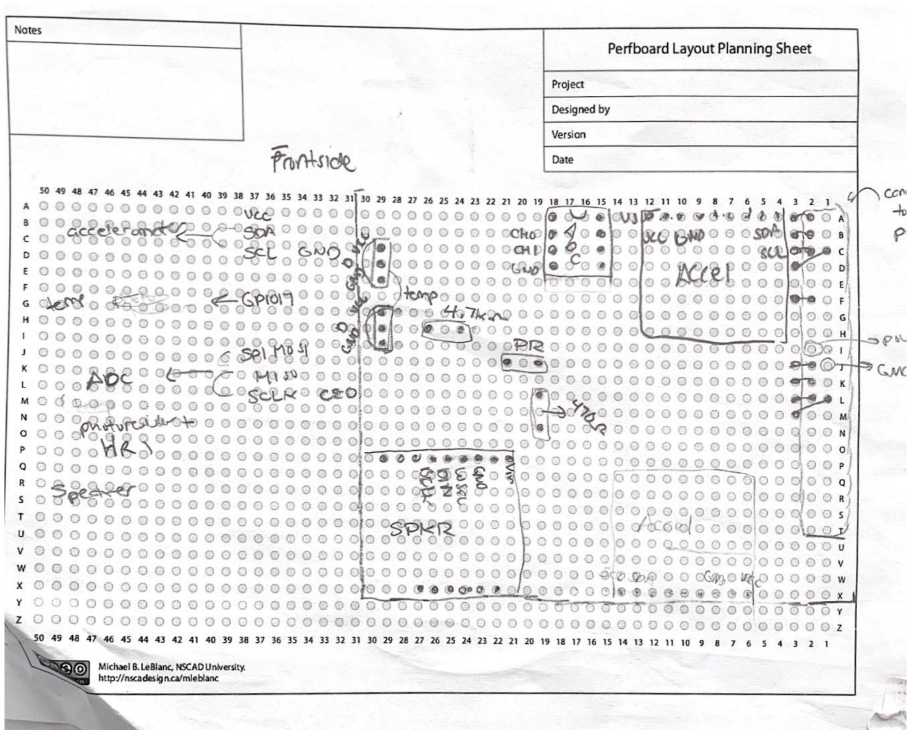
Side view of Perfboard



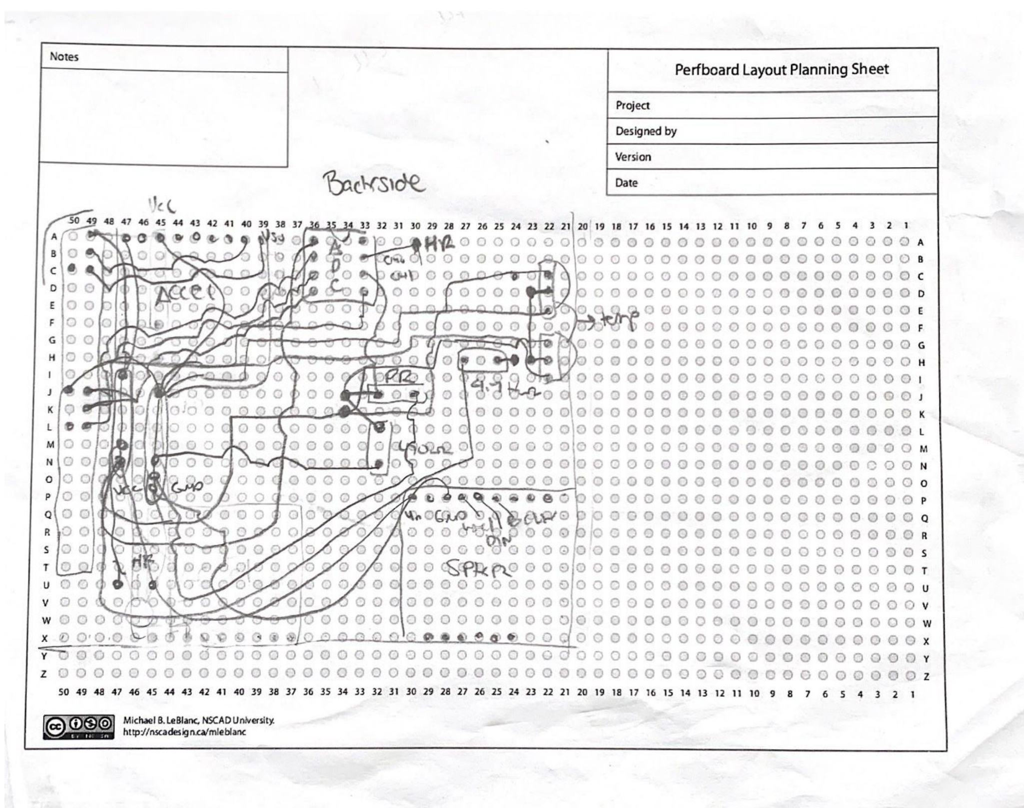
Top view of the Perfboard



Perfboard Layout Design top view



Perfboard Layout Design bottom view



Contribution

Anthony
Design: Came up the possible choices for sensors and measurements so that the device can detect moods based on those values. Performed market research on similar ideas and devices to compare and contrast our product idea to pre-existing ideas. Working on parts research involving the heart rate sensor and the motion sensor of the user. Worked on the software library (everything but pulse sensor), researching and gathering information and snippets of code to use. Created the demo_test file and the setup file. Completed the license and readme txt files. Reviewed the kiCAD schematic design and PCB design layout. Soldered the wire layout of the heart rate sensor, accelerometer, some of the temperature sensors, the adc, and some of the speaker.

Report: Responsible for the Abstract, Attribution, Design (all but the PCB portion), Project Concept, Parts Selection (Accelerometer and Heart Rate), Parts Evaluation, Appendices

Sally:
Design: Discussed together to come up with possible choices for sensors and how to implement the design. Worked on the parts research on the light and temperature sensor. Worked on the pulse sensor software library and on demo of it. Reviewed other parts of the software library on other sensors. Worked on wiring of entire Fritzing Diagram (except for UDA wiring). Had completed both kiCAD and PCB schematic design. Soldered the most of the wires for temperature sensor and speaker, along with mounting female headers for the pi rail, and also worked on some parts of back wiring for photoresistor and accelerometer.

Report: Was responsible for Design (PCB portion), Parts Selection (Temperature and Light), Attribution, Appendices

Appendices

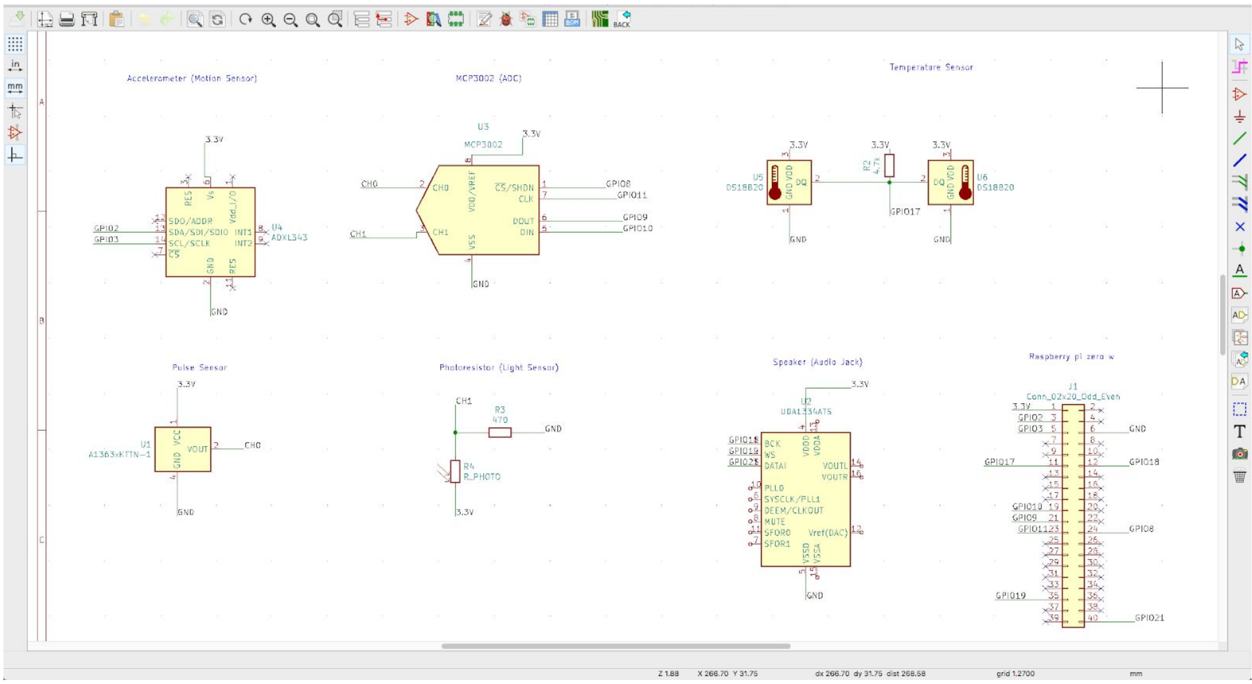


Figure 1: Fritzing Schematic

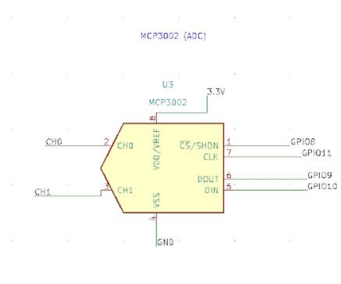


Figure 2- Analog to Digital Convertor (MCP3002)

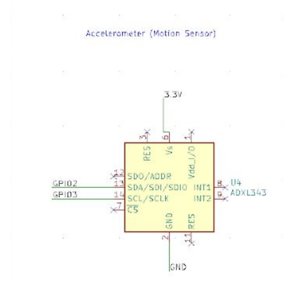


Figure 3- Accelerometer (ADXL345)

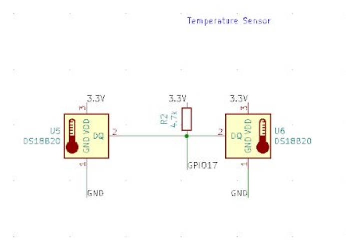


Figure 4- Two Daisy Chained Temperature Sensors (DS18B20)

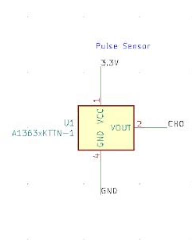


Figure 5- Heart Rate Pulse Sensor

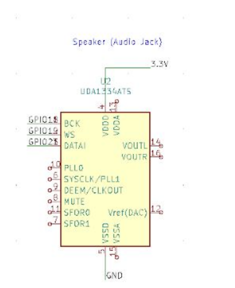


Figure 6- Stereo Decoder (UDA1334A)

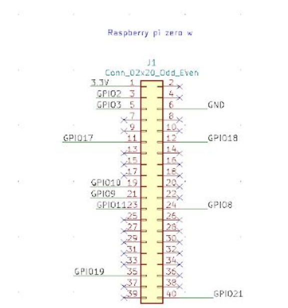


Figure-7 Raspberry Pi Zero Pinout

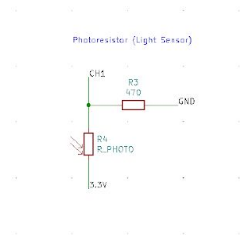


Figure 8-Photoresistor with a resistor of value of 470 ohms

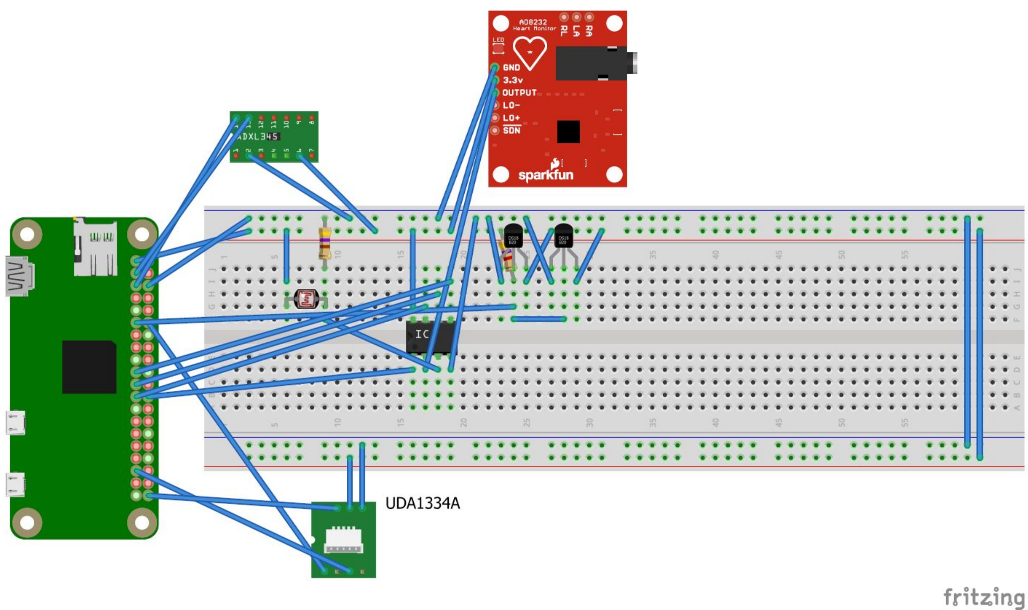


Figure 9-Breadboard Fritzing Diagram

Attribution

Parts Research

Accelerometer types, their definitions, functions, and possible usages for each type.
https://blog.mide.com/accelerometer-selection, Steve Hanly, March 11, 2016

Different types of speakers that exist
https://electronics.howstuffworks.com/speaker-drivers.htm

Light sensors
https://www.electronics-tutorials.ws/io/io_4.html

Infrared Temperature Measurement
https://www.omega.com/en-us/resources/infrared-temperature-measurement-theory-application

Market Research

Mood music blog post that shows the top 10 mood music applications and what they are.
https://www.apartmenttherapy.com/mood-music-alternative-audio-advice-based-on-your-feelings-166378

Habu Music application, using the FAQ page to understand how the application works as it is based on pre-generated playlists that have a mood label.
http://habumusic.com/

The app store page that describes what the Mood Music player application by FriQDevelop is about which plays music based on pre-generated mood playlists.
https://play.google.com/store/apps/details?id=com.friq.music&hl=en_US

Another mood music playing app that chooses music based on a range of moods inputted by the user.
https://moodagent.com/

Some information about the current headphone market shares around the world, giving insight on the saturation of the market and whether or not our product will be able to successfully launch
https://www.marketwatch.com/press-release/headphone-market-2019---business-size-share-opportunities-future-trends-top-key-players-market-share-and-global-analysis-by-forecast-to-2024-2019-11-08

Part Setup and Sample Code Usage

DS18B20 Tutorial setup and code
http://www.circuitbasics.com/raspberry-pi-ds18b20-temperature-sensor-tutorial/

Setup for multiple 1-wire buses on the Pi using the Daisy Chaining technique
https://blog.oddbit.com/post/2018-03-27-multiple-1-wire-buses-on-the-/

Adafruit's setup setup, wiring, and testing guide for the stereo decoder. Includes the wiring tutorial, the setup code and library installations and some test code for device
https://learn.adafruit.com/adafruit-i2s-stereo-decoder-uda1334a/raspberry-pi-usage

The Stereo decoder simple datasheet for referencing. It also includes information for setup and example guides.
https://cdn-learn.adafruit.com/downloads/pdf/adafruit-i2s-stereo-decoder-uda1334a.pdf

Photoresistor setup guide with wiring and software tutorial
https://tutorials-raspberrypi.com/photoresistor-brightness-light-sensor-with-raspberry-pi/

Source code for the Heartbeat Pulse Sensor. The repository also includes source code for the analog to digital converter (ADC MCP3002)
https://github.com/tutRPi/Raspberry-Pi-Heartbeat-Pulse-Sensor

Datasheets for Sensors

Photoresistor Datasheet
https://cdn-learn.adafruit.com/assets/assets/000/010/127/original/PDV-P8001.pdf

Accelerometer Datasheet
https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf

Analog to Digital Converter Datasheet (MCP3002)
http://ww1.microchip.com/downloads/en/devicedoc/21294e.pdf

Stereo Decoder Datasheet
https://cdn-shop.adafruit.com/product-files/3678/UDA1334ATS.pdf

Heartbeat Pulse Datasheet
https://media.digikey.com/pdf/Data%20Sheets/Pulse%20Sensor%20PDFs/Pulse_Sensor.pdf

Additional Referenced Sources

Guide and template for making a README text file.
https://www.makeareadme.com/#license-1

The open source licence that our product is going to be licensed under from MIT
https://opensource.org/licenses/MIT

Resources referenced from NYU ECE-4313

Software setup for the SPI communication protocol from Lab 4: Interfacing Electronics
https://newclasses.nyu.edu/access/lessonbuilder/item/29082660/group/3947d0f1-ec3b-401f-86fb-388be6a711b3/Lessons/4:%20Interfacing%20electronics/Lab%20material/interfacing.pdf

Referenced photoresistor setup guide to read light intensity and the Raspberry Pi Zero GPIO pinout from Lab 2: Input and Output
https://newclasses.nyu.edu/access/lessonbuilder/item/29057006/group/3947d0f1-ec3b-401f-86fb-388be6a711b3/Lessons/2:%20Input_output%20with%20pins/Lab%20material/input-output.pdf

Referenced the template layout for setup.py in device from lecture 8 on Software library's presentation: Writing a Software Library for a HAT
https://newclasses.nyu.edu/access/lessonbuilder/item/29117559/group/3947d0f1-ec3b-401f-86fb-388be6a711b3/Lessons/8:%20Software%20library/Software%20library.pdf