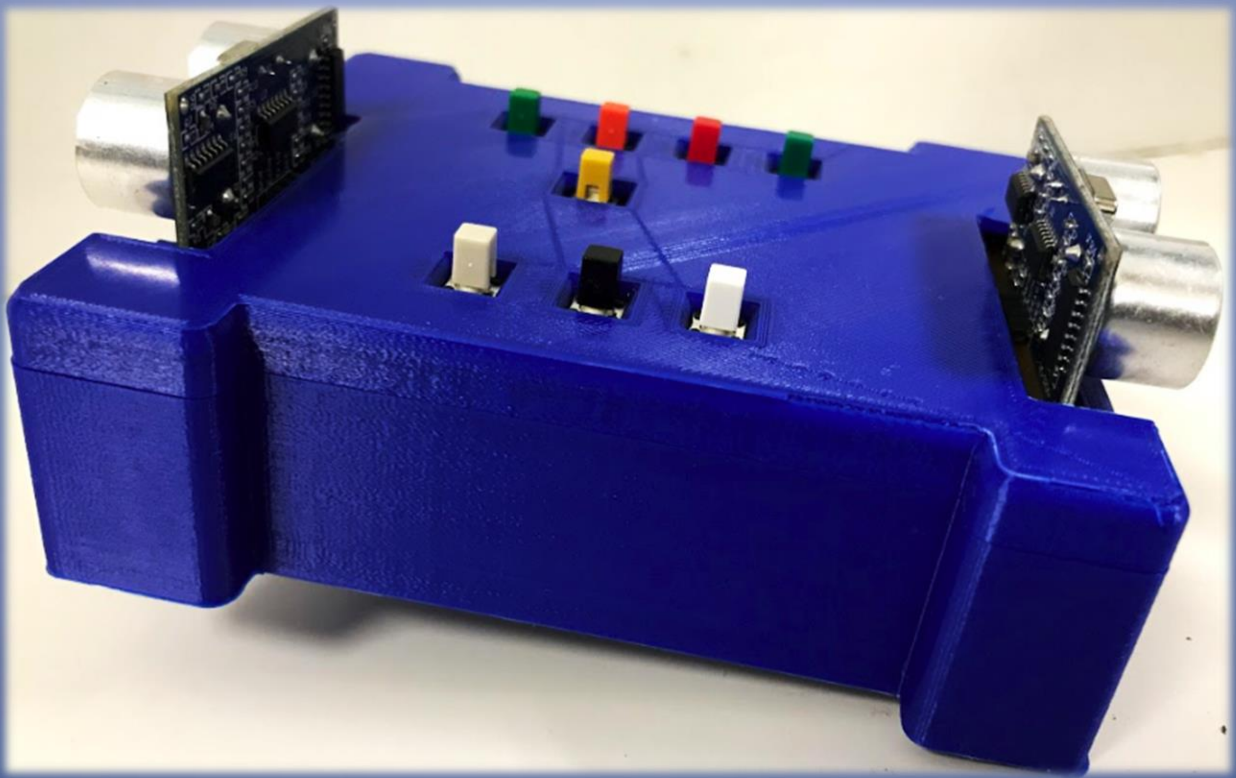
Motion Synthesizer



Amandeep Singh,
Bikramjit Singh,
Zachary Legg

Recipients: Mel Dundas, Joe Benge

Applied Research Project (ECET 290)

12/04/2017

Memorandum

To: Mel Dundas, Wayne Mayes

CC: Joe Benge

From: Amandeep Singh, Bikramjit Singh, Zachary Legg

Date: December 04, 2017

Subject: Motion Synthesizer Final Report

BiramanZac group is pleased to submit the final technical report on the Motion Synthesizer Project as it has been completed successfully. Our team has been working hard for the past 3 months on the project, and the final product reflects our efforts.

Our product meets the major specifications that we described in the requirements document, and the device's cost is lower than the budget allotted to us by the college. We implemented an Enclosure, GUI interface, PCB, and Color Calibration feature for our device. We found the relevant courses like *Design for Manufacturing, Project Management, Networking & Operating Systems, Object Oriented Programming, and CAD for Electronics* very helpful throughout our development. We are very proud of our final product that uses some interesting features of Electronics & Computer Technology.

Our team is very thankful to Mel Dundas, Joe Benge, Todd Rayson, Andrew James, Backroom Staff, and all the members of Electronics Department who guided us throughout this project in various situations. We are very pleased with the technical knowledge, documentation skills, and design experience we gained by this project.

Enclosed: Final Report (1 copy)

EXECUTIVE SUMMARY

Our team developed an interesting device which allows people of any age to play and enjoy it. It's a perfect toy for children, and a tool for demonstration of new technologies and artificial intelligence. The Motion Synthesizer allows the user to control musical effects with some unique methods and features. Our device:

- Allows the user to control music with back & forth hands' movement.
- Supports HDMI screens, 3.5mm jack headphones/speakers.
- Works for tracking objects of any color and control music with that.
- Gives the option of 20 different sound effects in various modes.
- Displays a GUI interface for an easy access of different options.
- Guides the user by showing some instructions on the GUI window.
- Runs on plug in to the power.
- Requires very less technical expertise to set it up and play.

We were able to meet all the requirements, and get a finished product within the budget of $250. However, we could have added more features and modes if we had more time. Also, we would have liked to develop our own training file for the OpenCV library to track a particular object. But we were not able to implement this method because of the limited time.

Overall, our team is glad that we learned a lot about modern artificial intelligence and machine learning in this project, and developed the product successfully. This experience of project management, manufacturing and designing, software research, and time management is definitely an asset to our careers.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION/CONCEPT

The Motion Synthesizer is a fun device for creating and playing music. It's an electronic instrument that allows anyone to control different sound effects. The user can control the pitch of the sounds with motion. We created a simple Human to Machine Interface for easy access to the different options available in our device.

Our group BikramanZac consists of Amandeep Singh, Bikramjit Singh, and Zachary Legg. We were all part of important decisions and review meetings regarding our product development. We met all of our specified requirements and tests. We were able to finish our product within the available time and budget allocated by the Electronics Department.

PURPOSE/SCOPE

Our device can be used in the schools, especially by the low incident students who may not be able to play a musical instrument with the band, but still want to be musically creative. It may also help them to express themselves and communicate in a different form. Motion Synthesizer can be also used as an example tool in teachings of modern Machine Learning & Computer Vision Technology.

HOW WE STARTED?

The original idea we had for the project was to control the pitch of the synthesizer's sound by measuring the user's hand movements with two ultrasonic sensors. Upon further discussion and consideration, we decided that we will also use a camera and computer vision to give the user a greater range of motion and creativity to make music. We made a basic block diagram of the functionality of the Motion Synthesizer (see Appendix A).

We researched on different techniques of computer vision to give our product some unique features of tracking objects and human body movements. We spent our initial time to learn basics of *Python* programming language, *Sonic PI* music synthesizer, and configuring the computer vision (OpenCV) library.

WHAT HARDWARE MODULES ARE USED?

The Motion Synthesizer uses a Raspberry Pi 3 to process all incoming and outgoing data. It creates sounds using the *Sonic PI* music software installed on the Raspberry Pi 3. *Sonic PI* is a coding environment that uses a synthesis engine to produce a variety of sound effects. We used two HC-SR04 Ultrasonic sensors for measuring the hand movements, and a USB camera for tracking of objects and body motion. The HC-SR04 is a commonly used module that uses sonar to measure the distances.

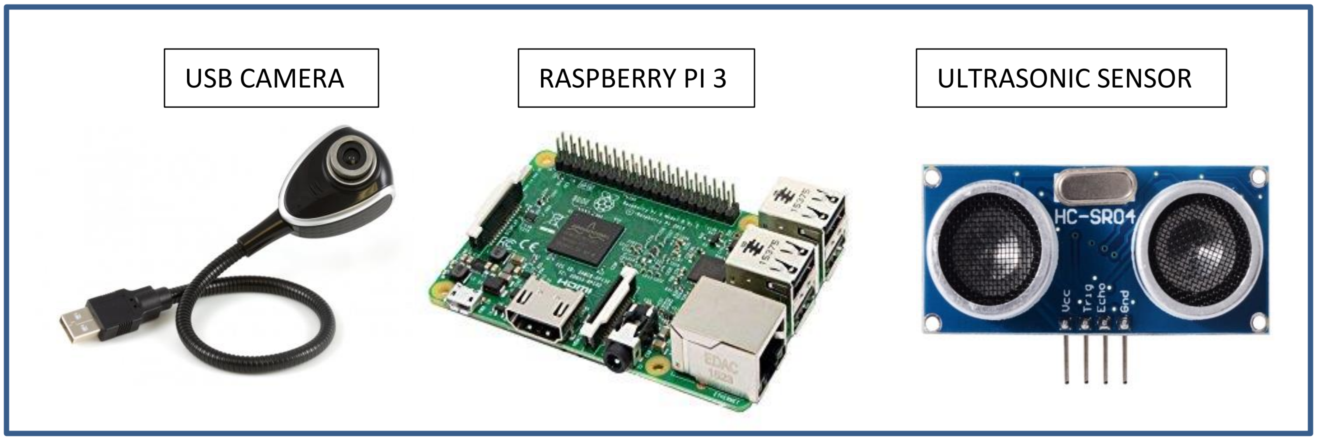


Figure 1: Main Hardware Modules

The ultrasonic sensors are rated for 5V. We used a small voltage divider circuit consisting of two resistors to lower the sensor's output voltage to something our Raspberry Pi can handle (see Appendix E for a schematic).

Motion Synthesizer is compatible with 3.5mm jack headphones or speakers for the audio output. We used the HDMI monitors for the visual output of the synthesizer. The user can use any HDMI compatible screen for the display of the camera output and the GUI interface.

Apart from the above hardware, we used various hardware accessories and components like resistors, HDMI cables, PI Views, Push Buttons, and Raspberry PI 3 header jack for the interface. These accessories are part of our finished product which is ready to go when plugged in.

SOFTWARE DEVELOPMENT

We have mainly used *Python* programming language, *Sonic PI* music software, and *OpenCV* Library for our software development. OpenCV is an open source library available on GitHub that is capable of doing various things related to Machine Learning & Computer Vision. Computer vision is concerned with the automatic extraction and analysis of useful information from a single image or a sequence of images.

We researched and learned about numerous computer vision techniques to track objects and human movements. We control a music synthesizer engine with the data captured by the camera. We tested different methods including Haar Cascade Calssifiers, Frame Difference Method, HSV Color tracking techniques, and inbuilt Tracking Modules of OpenCV library. In our final product, we decided to use Frame Difference technique & HSV Color tracking to make our system work smoothly.

In frame difference method, we capture frames continually with camera, and subtract the previous frame from the latest frame. The difference frame displays the movements made by user in that time period. Based on the difference values, we control the music. HSV color tracking consists of tracking the objects of certain color in a given real-time video stream. We made our own calibration feature also that allows users to calibrate the Motion Synthesizer to track objects of their color choice.

We have ultrasonic sensors interface also that allows user to control music with back & forth hand movements. We have used sub processes & multiprocessing to run different python scripts at same time. We have a main script which launches other scripts based on user's inputs. We used various Python & OpenCV modules in our source code as shown in Figure2. We had to install these modules on Raspberry PI as root user in order to have access to them from everywhere.

```
import numpy as np
import argparse
import cv2
import time
from psonic import *
from tkinter import *
import subprocess
import RPi.GPIO as GPIO
```

Figure 2: Software Modules Used

We also developed a GUI for the user to access all the functions of our device. We used the tkinter module in Python to create things like canvas, background images on screen, displaying different layouts and options to the user.

FEATURES & CAPABILITIES

We have two main modes of our device. It has the ultrasonic interface, and the Camera mode that allows object tracking & visual mode of frame difference technique.

➢ Ultrasonic Sensors Mode

We have used two ultrasonic sensors in our design. People can move their hands back & forth in front of the sensors, and it gives them freedom to control the rate of notes, pitch of the sounds. People can also use various push buttons to switch into different advanced modes. There are four push buttons for the ultrasonic interface. Each sensor has one button (Button #1 & Button #4) for changing to different sounds as shown in figure 3 below. There is one button (Button #3) for increasing or decreasing the note rate, and we have one button (Button #2) for a special tone which we created by mixing various effects.

Figure 3: Button Layout of Motion Synthesizer

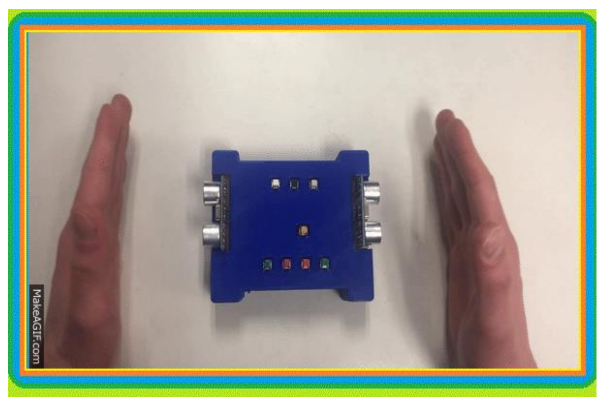


Figure 4: Ultrasonic Mode Playing Sample

There are about 20 different sounds which can be used by the user. Special mode is a unique music effect where one of the sensor controls the main music tone, and the other sensor plays a background effect on that sound. The user can also play different background effects like drums, soft drums etc. People have to wave their hands in this mode as shown in the figure 4 above, and its range is within 50 cm from the sensors

➢ Camera Mode

This is the more advanced and interesting mode we have developed. There are two main sub features in this part:

a) Tracking Mode

People can use our calibration feature to track objects of any color. We have used various balls to play with our device, and it's a lot fun to track anything in this mode. The user gets 10 seconds to place the object in the circle as shown in figure 6, and the calibration feature captures the color and tracks it as shown in figure 5. The user can control the sounds by moving the object over the screen. If the user goes out of the screen, the music restarts when the object will re-enter in the screen again.

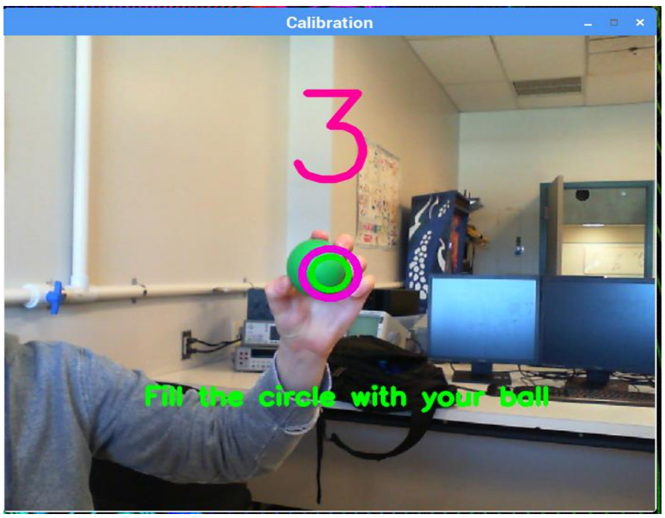


Figure 6: Calibration of the Object

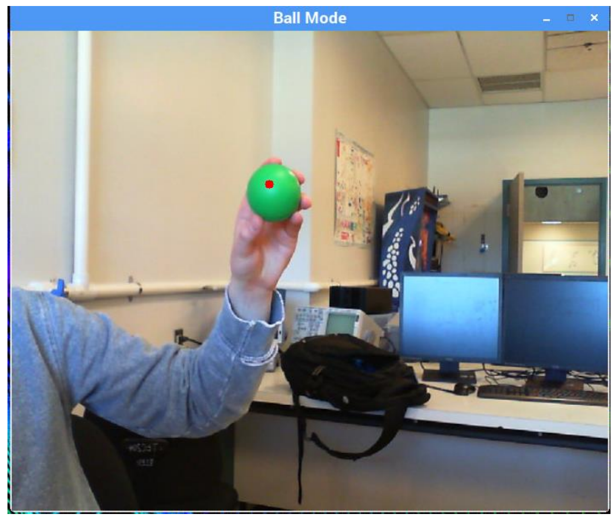


Figure 5: Tracking Mode Example

b) Frame Difference Mode / Dance Mode

This is also a special mode of our device which is based on the Frame Difference Technique we developed using computer vision features as discussed earlier. When anything in the real-time video stream will move, it will be displayed in Gray Scale with unique colors as shown in figure 7. Basically the user has to move/dance or do any kind of movement in front of the screen, and the Motion Synthesizer will start playing sounds. It plays a different song when user restarts the dance. The user has to keep moving in order to keep the music ON, and that's why we named this as the Dance Mode.

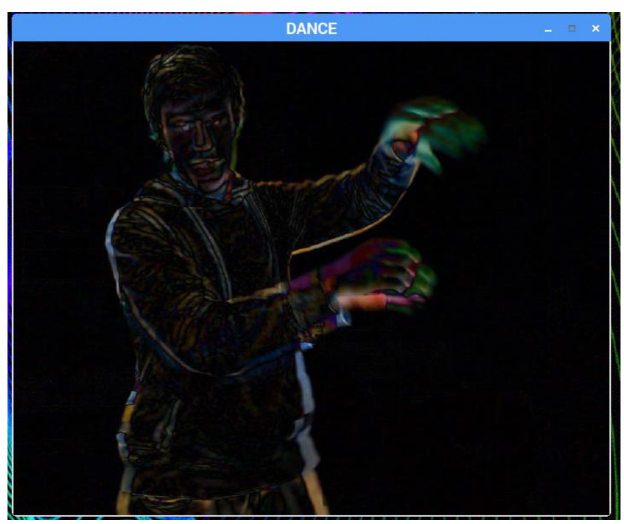


Figure 7: Frame Difference Method's Visual Display

DESIGN & MANUFACTURING

➢ Schematic & PCB Design

We completed the PCB's schematic design in the *Altium Designer* (See Appendix B for the schematic design). The PCB is mounted on top of the Raspberry Pi giving it access to the ultrasonic sensors and eight pushbuttons. Some of the buttons are used for multiple things in different modes. To test the schematic design, we built the circuit on a breadboard and ran the software to ensure that everything works according to the schematic. We consulted the Camosun PCB Manuals to confirm that our PCB layout met all the requirements specified by the Electronics Department (See Appendix C for our PCB layout). We made the PCB single layered, and we were able to get it manufactured in just one week (Figure 9).

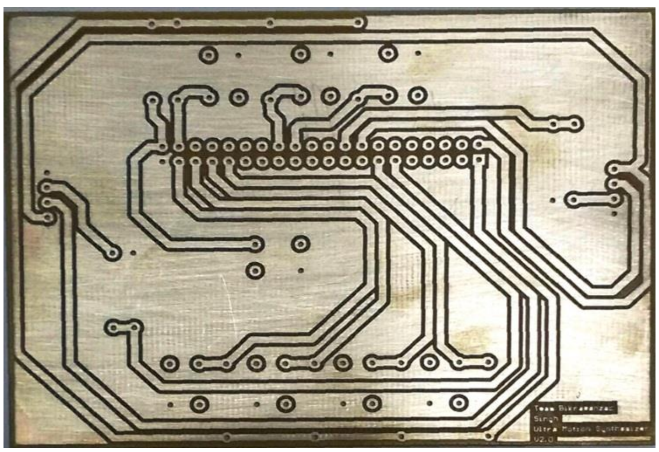


Figure 9: Single Layer PCB Manufactured

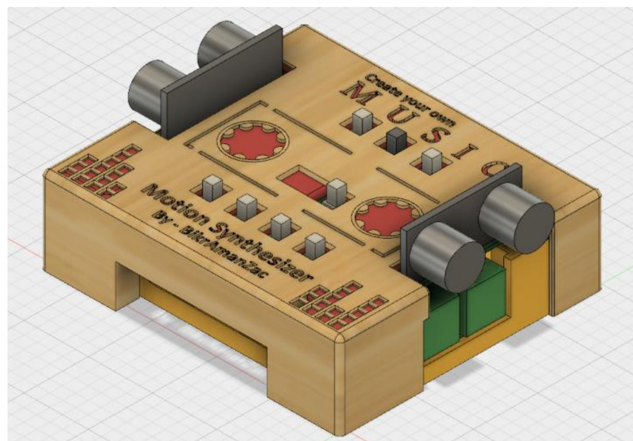


Figure 8: 3D Model of the Enclosure

➢ Enclosure Design & 3D Printing

We designed our enclosure using the *AutoDesk Fusion 360* software. We first created the 3D model of our PCB mounted on the Raspberry PI. We measured the dimensions of every component to make our model accurate. We also designed the Ultrasonic Sensors, and the pushbuttons for the 3D model. We had to print the enclosure 3-4 times to get the perfect fit. We also designed a cover for the enclosure as shown in figure 8. We used the PLA material for the 3D print of our enclosure.

HOW IT WORKS ALL TOGETHER?

Integration was a critical part of our project. We were having the various parts of our project working individually very well. But when we tried to combine everything, we had to change our software a lot. We initially tried to use the threading to run different tasks in parallel. The problem with this method was we could not pause a particular task if needed while running everything together. We used the multiprocessing to handle this issue, and launch different scripts as child processes from the main script. We made an application link for our bash script file so that we can run it on the startup. The main bash script launches the other tasks like camera mode script, sensor's script based on the user's input & some internal arguments. Our final product runs the GUI on the startup, and the user just has to plug the device to play with it.

➢ The Graphical User Interface

We designed the GUI for our device so that the user can access all the features easily. The GUI shows separate screens in each mode to guide the user. When we plug the device in, the GUI starts up and displays the loading message so that everything gets ready in background as shown in figure 10 below. We have used various GIF photos on some of the GUI loading screens.

We also have a pause menu if the user wants to stop the synthesizer. If the user is in a particular mode, the GUI screen tells the user which button is needed to be pressed and how to play in that mode. We have a special feature built in for the calibration of object of any color. The user can press the button #3 while in camera mode to start the calibration window. After 10 seconds, the calibration disappears and the tracking mode starts automatically. Our user manual explains all the details of the interface.

Figure 10: The GUI Interface

PROBLEMS/CHALLENGES

We were having an issue with the camera when we started testing OpenCV. Whenever we tried to capture a picture or video, the camera would unmount from the Raspberry Pi. When we ran our code repeatedly, we kept getting errors because the camera was no longer available. We found out that it takes time for the camera to mount again. We were able to solve this issue by modifying our code so that it waits for the device to connect before trying to access it again.

Designing the PCB was actually tougher than we thought. Routing all the components on one side of the board was a big challenge. We wanted to design a single sided PCB because the manufacturing time is a lot shorter than a double-sided board. To help us route the connections on our PCB, we used a Polygon Pour for the ground plane. After many hours of scrutiny, we finished designing our PCB according to the specifications in Camosun's PCB manuals.

It was challenging to merge the ultrasonic and the computer vision portions of our project together. Both take up a lot of the computer's memory, and it was difficult for the Raspberry Pi to handle all the processes at once. We also ran into the problem of deleting the whole image of the operating system from the Raspberry PI. We had to reinstall all the modules, libraries, and settings for our system. We spent a lot of time to solve the problems in integration using parallel threads. After a lot of effort, we found that there are some limitations of threading modules of *Python3.6.* We solved the issues in integration by using multiprocessing, sub processes, and a bash script to launch the applications on the startup.

Apart from the technical challenges, finishing everything in time was also a separate challenge. The 3D printing of the enclosure takes a lot of time as the different printers have different tolerances, problems, and print settings that make the printing different. We kept the PCB single layer in order to manufacture it quickly. We followed our Gantt chart (See Appendix D for our Gantt chart) and goals strictly so that we can finish everything in the limited time available.

FINANCIAL & TEAM DYNAMICS

We finished our project well within the provided budget of $250 by the college. We only needed to pay for certain things that were also not very expensive. The rest of the materials we used were from the previous years of our program. Overall, we have a device capable of various things with a very low price. A breakdown of all our project costs can be seen below in Table 1.

Table 1 – Materials and Costs

Part	Quantity	Unit Cost	Cost
Raspberry Pi 3	1	$70	$70
HC-SR04 Ultrasonic Sensor	2	$7	$14
USB Camera	1	$30	$30
Speakers	1	$50	$50
PCB Manufacturing	1	$10	$10
Enclosure	1	$10	$10
Pushbuttons	8	$1	$8
Button caps	8	$0.50	$4

Our group has worked together for the last two years in this program. Our good team chemistry helped us to manage our project so that we could achieve our goals within the given constraints. We were all very flexible and hardworking. We solved the problems that came in our way, and always stayed on top of the deadlines. We worked extra time sometimes to finish the tasks ahead of time. We were all able to discuss our problems with each other and the teachers to find the solutions. Amandeep worked mainly on the design & manufacturing, Bikram was in charge of the software research, and Zach was responsible for integration and coordination of the whole project.

CONCLUSION

Our team enjoyed working on development of the Motion Synthesizer. We had to change some of our initial ideas and directions, but in the end we have met all the requirements we stated. We recommend the future students to plan the project's timeline and strictly follow it. Proper documentation is also a key for the success of any project. We also advise people to think of risk management, slack time, engineering hours, and the scope of their project. Deciding about the achievable requirements, tests, and goals was very important to meet all the deadlines. We have learned a lot in various areas throughout this development. Overall, our product is capable of various interesting features. We are very impressed and happy with our team's success. We expect everyone to enjoy playing with the Motion Synthesizer.

CONTACT US

Team Member	Email
Amandeep Singh	amandeepsingh1243@gmail.com
Bikramjit Singh	bikramjit421@gmail.com
Zachary Legg	leggzachary@gmail.com

APPENDICES

APPENDIX A: Drawing of the Motion Synthesizer's Functionality

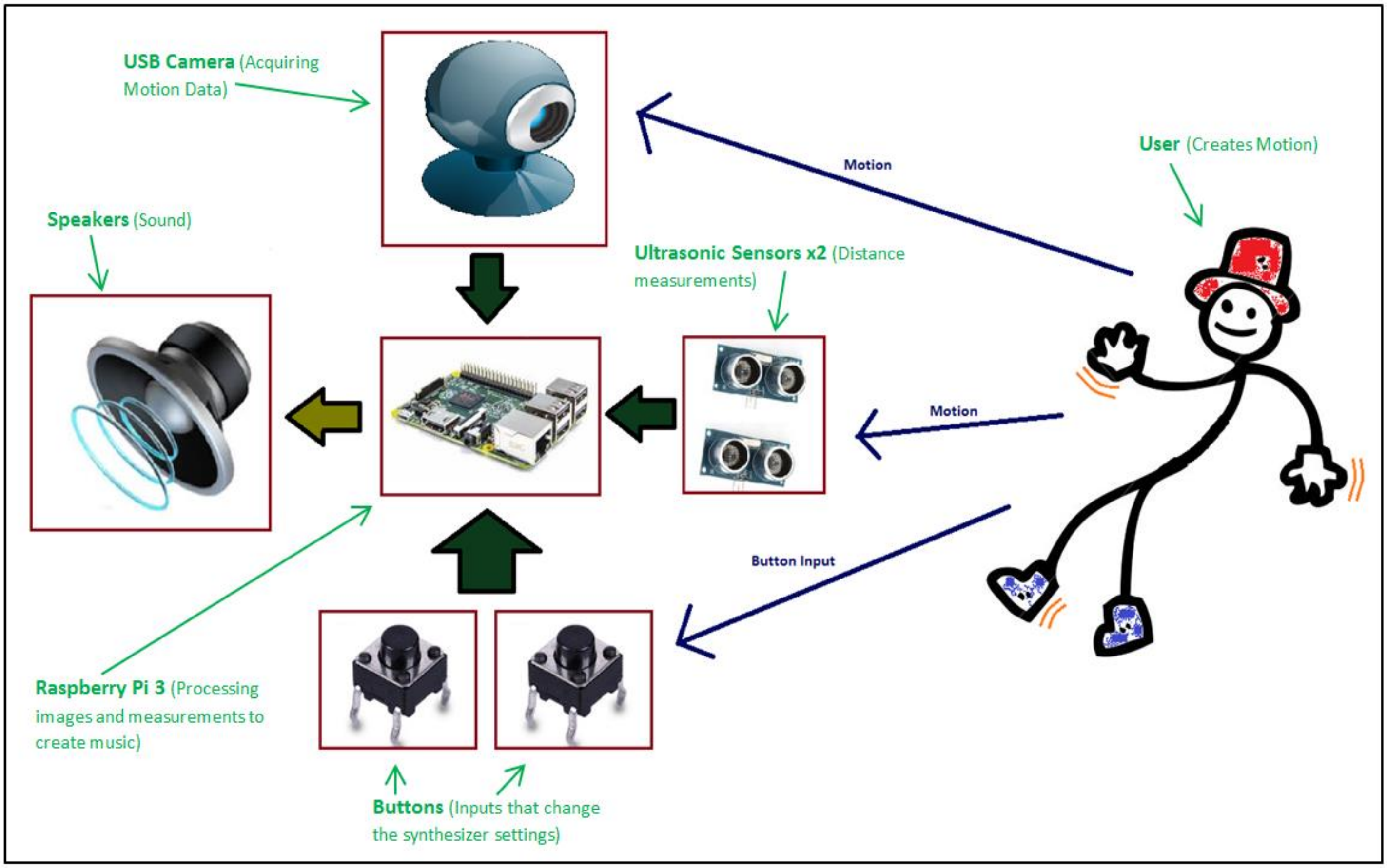


Figure 11: Basic Block Diagram of Motion Synthesizer

APPENDIX B: Circuit Schematic

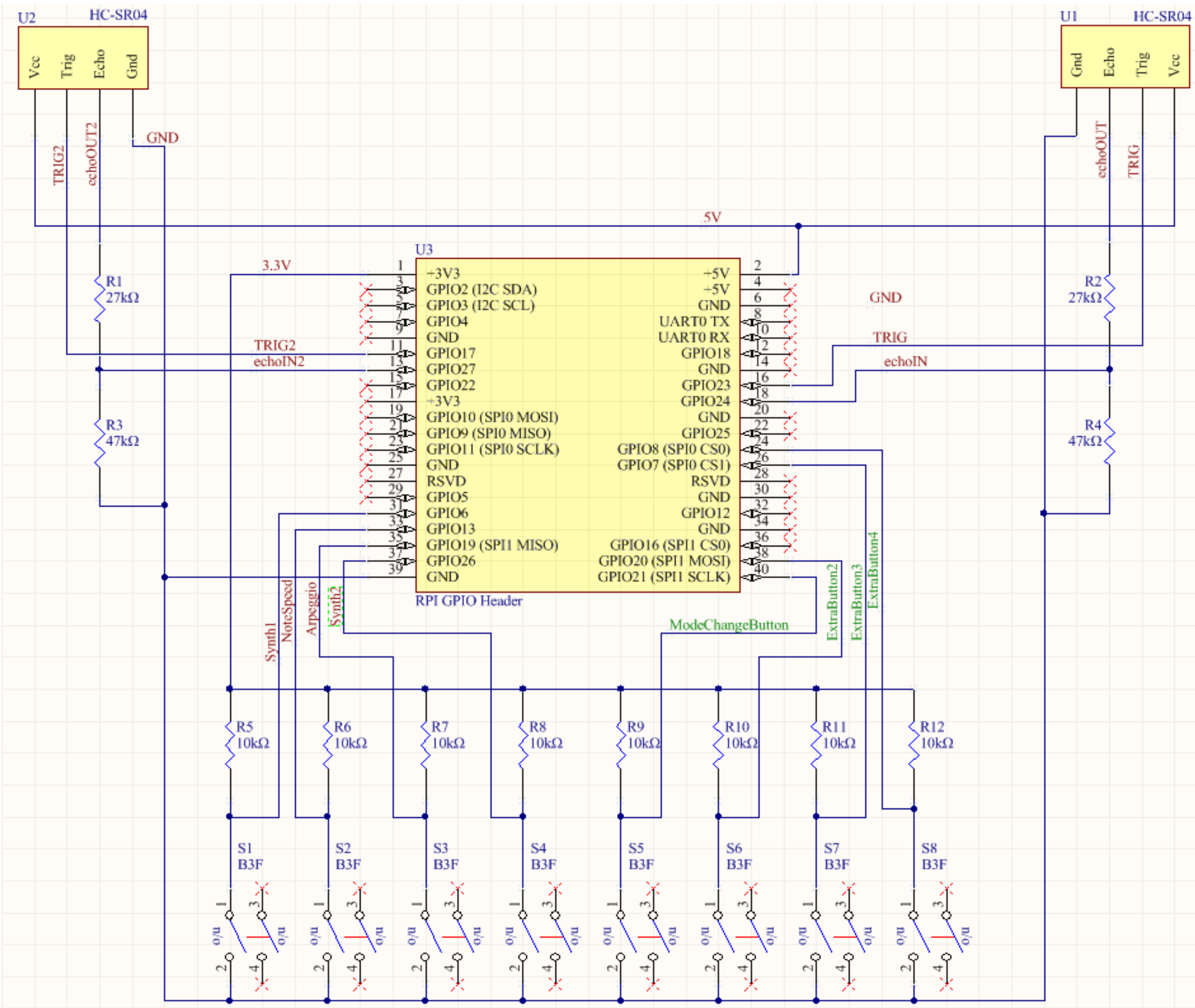


Figure 12: Schematic of Circuit Layout

APPENDIX C: PCB Layout

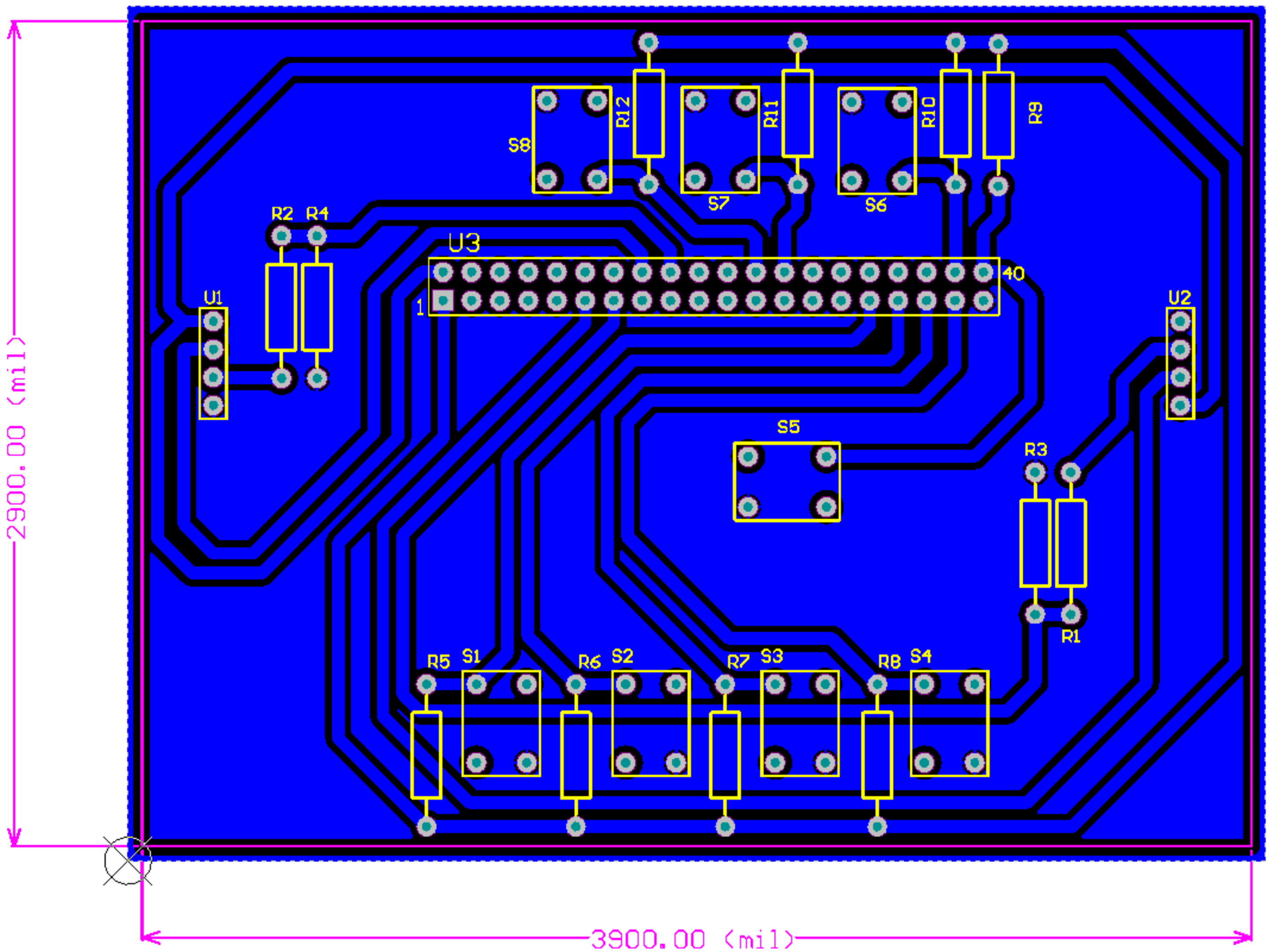


Figure 13: PCB Design in *Altium Designer*

APPENDIX D: Gantt Chart

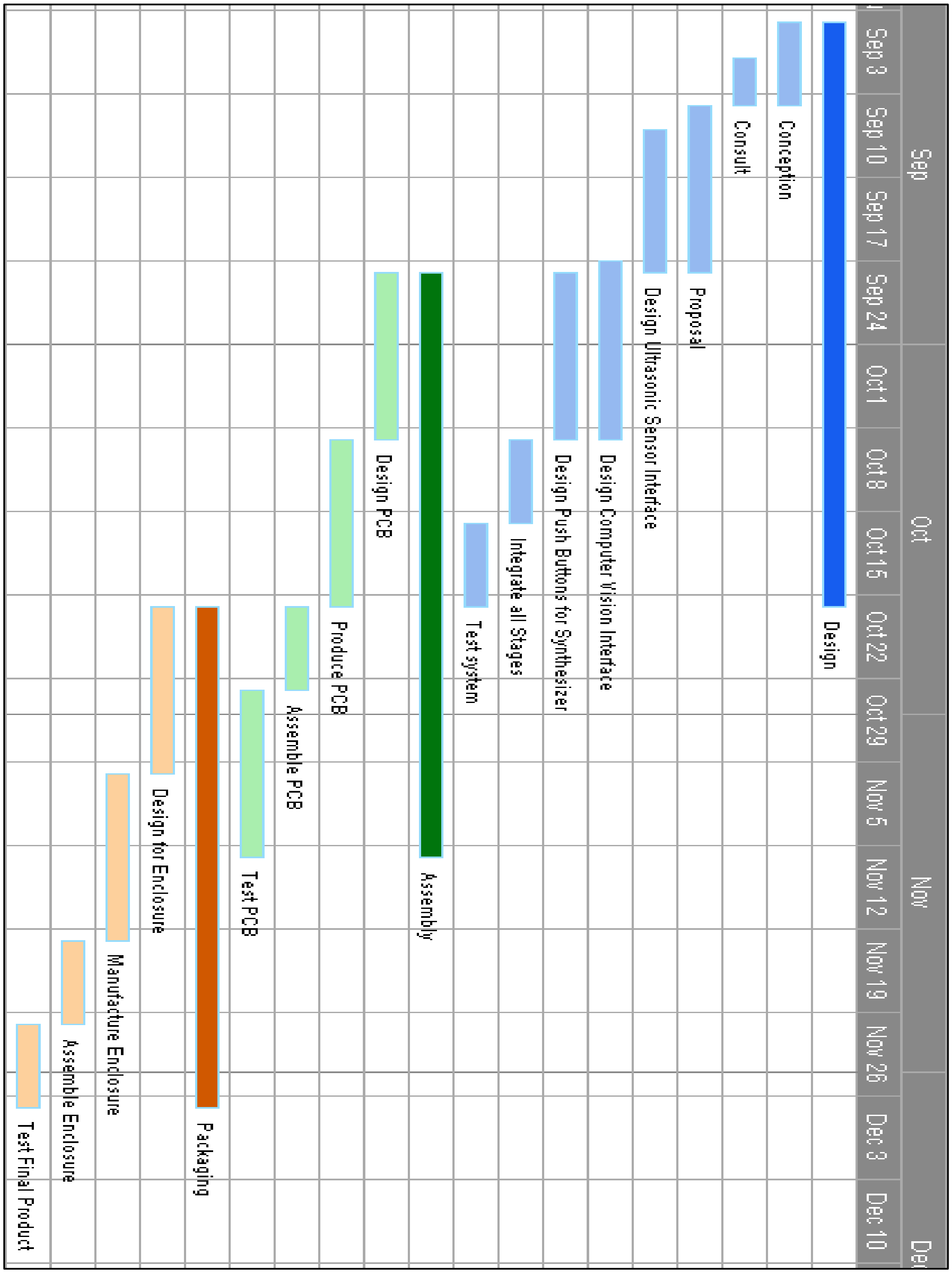


Figure 14: Gantt Chart Illustrating Project's Schedule

APPENDIX E: Voltage Divider Circuit

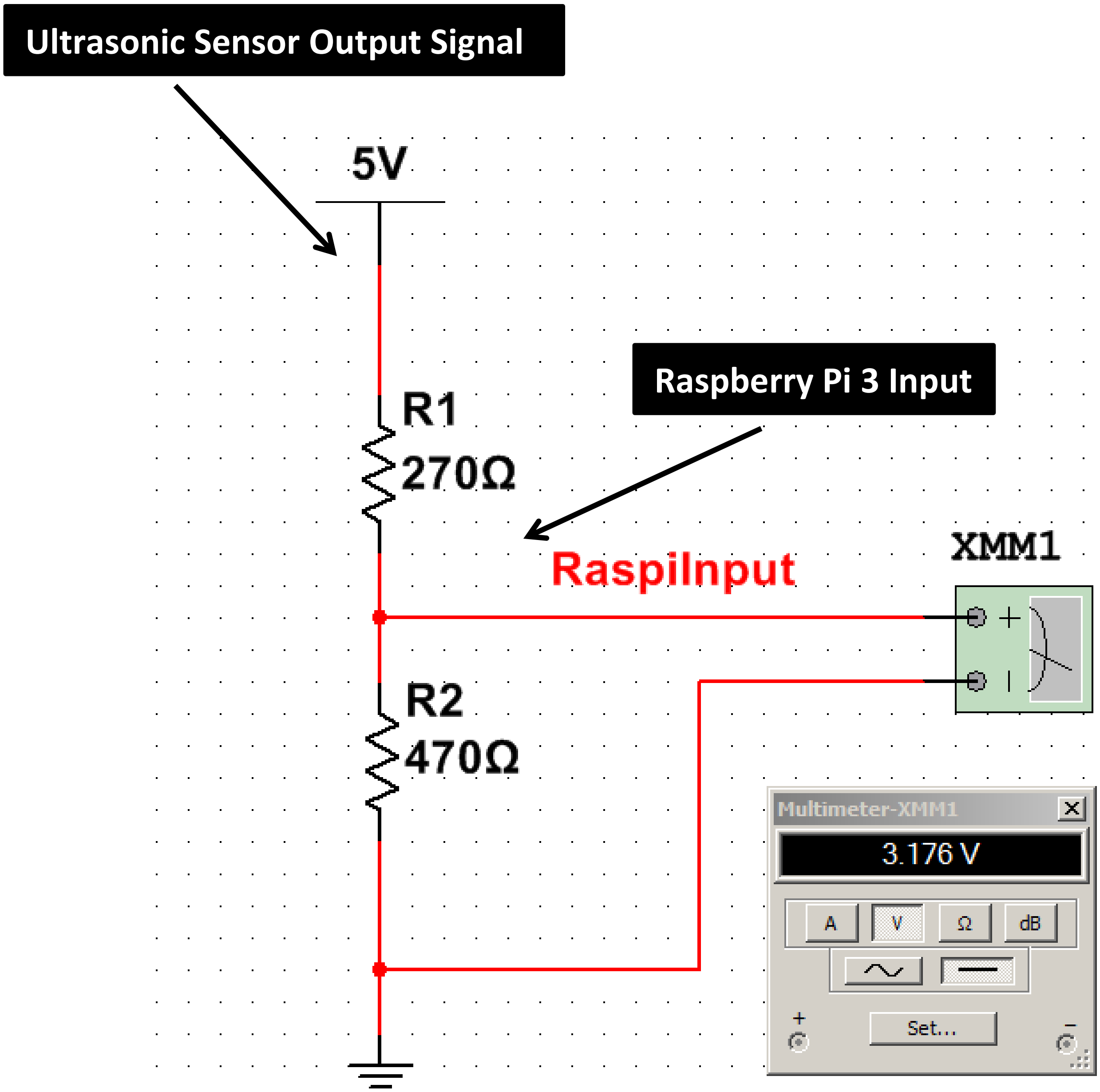


Figure 15: Voltage Divider Circuit for Sensor's Interface