BABY MONITOR FOR DEAF PARENTS

Capstone Project

B.Sc. Electrical and Electronic Engineering

Sanuratu Koroma

ID:23442020

2020

ASHESI UNIVERSITY

AN AUTOMATED BABY MONITOR FOR DEAF PARENTS

CAPSTONE PROJECT

This capstone project was submitted to the Engineering Department at Ashesi University as a final requirement for a Bachelor of Science Degree in electrical and electronic engineering.

Sanuratu Koroma

2020

DECLARATION

I hereby declare that this Capstone Project is my original work, and no part of it has been presented or submitted for another degree in this university or elsewhere. Candidate's Name:

Candidate's Signature:

SANURATU.KOROMA

....

Date: 29th/05/2020

....

I hereby declare that the preparation and presentation of this capstone were supervised per the guidelines on the supervision of a capstone laid down by Ashesi University.

Supervisor's Signature:

....

Supervisor's Name:

....

Date:

....

ACKNOWLEDGMENTS

Firstly, I thank the Almighty God for his grace and favor in my life every day. I would like to express my thanks and appreciation to my supervisor Mr. Richard A. Akparibo, for his kind support and guidance towards this project. I will also like to extend my sincere gratitude to Dr. Nathan Amanquah for also supporting me in my project. My profound thanks and appreciation to all the Lecturers in the Engineering department, your feedback helped in this project. Special thanks and appreciation to my parents for their support and encouragement through this project. Finally, I will like to extend my gratitude to Emmanuel Yaw Manu Annor for motivating me and helping me to finalize this entire project.

ABSTRACT

An automated baby monitor for financially challenged deaf parents in Africa. This system is operating on a 433MHz RF module to aid wireless communication. Its input is from a pulse/heartbeat sensor, which checks the beat per minute (BPM) of the baby when it is higher than 145BPM, indicating that the baby is crying. The Microcontroller used is ATMega328P. The focus of this system is to help a deaf parent recognize that their baby is crying when the parent is not in plain sight of the baby or the parent is sleeping. Hence, this system has two devices, one of which will be worn by the baby and another which will be worn by the parent, and communication is wireless. This system is made from affordable electronics components to ensure that the final product is cheaper as compared to its competitions.

TABLE OF CONTENTS

Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1: General Introduction

1.1 Background

Technology advancement has covered an impressive milestone over the years. The most notable developments are those that have contributed to people's relationship with society and their environments [1]. Such developments are key, especially to physically challenged individuals. One of the most common types of disability is deafness. About 5% of the world's population are deaf, which accounts for three hundred and ninety million people of the entire world's population [1]. Out of this 5%, those found in sub-Saharan Africa find it difficult to interact in society and have independent lives. It is often difficult for a deaf parent to relate to their babies without getting most of their help from other people. Newborn babies cannot walk or call out to their parents; therefore, they often rely on their cries to get their parent's attention. If the deaf parents are poor, they cannot afford the very few and expensive options on the market. Finding a cheap option has been the challenge for many years now. Only very few researches exist in this area. Hence, this project seeks to research and develop a cheap solution to this problem. This device will bridge the gap of emotional trauma that hearing babies and deaf parents face in the first few years of the baby's life. This system will enable the parents to attend to their babies with little help from other people, thereby giving them control over their baby.

1.2 Problem Definition

This project focuses on developing a device that helps poor deaf parents recognize that their baby is crying. This development will be beneficial, especially when the parents are not close to their baby or the parent(s) are sleeping. This device will serve to strengthen the bond between a deaf parent and their baby by giving the parents a sense of independence. With this device, the parents will be able to take care of their babies with little help from other people, such as family members and friends.

Surveys show that physically challenged deaf parents, feel the most helpless when they cannot take care of their babies without help from friends [3]. A device that bridges this gap will strengthen the bond between the parents and their babies. Certain ailments that would have otherwise occurred because the baby cries too much could be avoided. The babies will spend less time crying because their parents can find out easily. Hence, there is a great need to embark on this project.

1.3 Project Objectives

The following are the objectives of the project;

1. A cheap and portable baby monitor for an average income deaf parent(s).
2. A baby monitor that works over a minimum distance of 100m.
3. A baby monitor with very low power consumption.
4. A baby monitor that is easy to use.

1.4 Expected Outcomes of the Project

The final product must satisfy all the following:

1. The device should be light enough to be worn by a baby of 3 months to 3 years old.
2. The device should last a year without changing the battery.
3. The device should operate for at least 100m.

1.5 The Motivation for the Baby Monitor Project

Living in one of the most underdeveloped parts of the world has many limitations and complications, even for a person with no physical challenge. Thinking of the social, economic, and emotional issues poor deaf parents faces in countries like Ghana, it seems almost impossible to believe. Meanwhile, this problem could be solved using electrical and electronic engineering.

1.6 Research Methodology Used

The following steps are the methods used in performing this project.

1. The initiation stage of this project focuses on literature reviews of other approaches to this problem.
2. Designing and prototyping of the baby monitor hardware.
3. Software development and programming using C code in Arduino and ATMega328P.

1.7 Facilities Used for the Research

The following tools and facilities aid in this project:

1. Computer, internet, and library facilities at Ashesi.
2. The electrical and electronic laboratory at Ashesi.
3. Programming languages such as Arduino

1.8 Scope of work

This project is limited to the design of a crying baby detector using a heartbeat sensor, microcontrollers, a vibrating motor, and LEDs.

1.9 Project Organization

This project consists of five chapters.

Chapter one: entails a summary of the aims and objectives, motivation for carrying out this project, expected outcomes, scope of works, and a list of the facilities that will be needed to accomplish this objective.

Chapter two: consists of a detailed literature review of similar papers. It discusses the aims, objectives, and other solutions of other authors that had simulated or built a similar device. It involves knowledge of statistical graphs, microcontrollers, voice detection sensors that were tested and yield relevant results.

Chapter three: focuses on the design methodology used to accomplish the baby monitor project. The hardware and software implementations of the designs are discussed in this chapter.

Chapter four: describes the results of the design and explains what each result means.

Chapter five: summarizes the project in conclusions, limitations, and future works of the baby monitor project.

Chapter 2: Literature Review

2.1 Introduction

Research has shown that the development of the bond between parents and their children is dependent on the parent's ability to be responsive and sensitive to their child's physical and emotional needs [3]. This project explores a different approach to solving the problem of deaf parents being unable to know when their baby is crying. The implementation of this project makes use of the rate at which the heartbeat of the baby changes from a rest state to a crying state. The very few competitions out there makes use of the voice of the child as an input. As stated earlier, this project will use the beat per minute (BPM) of babies between the ages of three months to thirty-six-months and use their crying range as an input to the system. Studies showed that the heart rate of an infant increases considerably from pre-crying and peak at crying. As indicated in Table 2.1, the input to this system will be 145BPM [5].

Table 2.1: Relationship between the heart rate of a baby and cry.

The time before crying(s)	Heart rate (BPM)
10	135
6	137
2	140
Cry	145

Table 2.2: Relationship between the age of a baby and their resting heartbeat.

Age	Resting Heartbeats per minute
0-3 months old	100-150
3-6 months old	90-120
6-12 months old	80-120
1-10 years	70-130

2.2 Overview of the Crying Baby Detection System

2.2.1 Method 1: A Baby Monitor Using an Algorithm/App.

According to Arianna Anderson et al. [2], the use of algorithms such as an app could help one know when a baby cry. The cries of over 2000 babies were sampled and used to build an app that can categorize the cry of a baby into pain, fussiness, and hunger. They achieved this by looking at the different frequencies in the cries and different patterns of cries and silence. The duration of silence while the baby cries was used to analyze the possible reasons why the baby might be crying. The app was used on iPhone and Android devices. The app works such that five seconds of the baby's cry is recorded and uploaded to a database. The cry properties such as frequency are compared with other rates to determine why the baby is crying. The result is displayed in the bar-graphical form [2]. A major setback of this design is that the deaf parents must be aware of their baby crying before they could use the app to tell why the baby is crying. Another setback is that sophisticated phones and knowledge about graphs are needed to operate the system. A financially challenged or illiterate deaf parent cannot be able to use the app because of its complexity.

2.2.2 Method 2: A Wireless Baby Monitor with The Use of a Voice Recorder.

This system was embedded in a toy. It is used to monitor whether the baby is crying or not. The cry detector consists of a voice recorder, an XBee module for communication, and an Arduino. When in operation, a notification from the toy module is sent to a watch module that the parent wears. The watch module consists of a screen that displays the data of the time the baby is at rest, and when he/she is crying. It also includes a vibrating motor that vibrates when the baby is crying [3].

Chapter 3: Design Methodology

3.1 Introduction

The baby monitor project consists of hardware and software development. This chapter highlights the concepts and procedures of this project.

3.2 Requirements

The Pugh chart in Table 3.1 of the appendix shows a comparison between the WIFI module, Bluetooth module, GSM module, 433MHz RF module, and LoRaWAN module against Infrared as the baseline. After making the comparison, the 433MHz RF module as the best fit for the wireless communication of this project.

Also, in Table 3.2 of the appendix, the baseline for making the comparison is the Raspberry Pi. After the comparison, the ATMega328P chip emerges as the most suitable one for this project.

The Requirements of the baby monitor project include:

1. Comfortable and detachable material to be worn by the baby to encourage washing and provide comfort for the baby.
2. Waterproof material for the device worn by parents.
3. A power supply system that is easy to manage.
4. A fast and responsive system
5. A system that can be used in every environment.
6. A system that is cheap to manufacture and repair.

3.3 Hardware Design

The components used for this project are Heart rate sensor, 2 ATMega328P, 433MHz RF, LEDs, Resistors, capacitors, op-amp, vibrating motor, photodiode, IR LEDs, transistors, Jumper wires, two 9v Batteries, Breadboards, and circuit packages.

The transmitter circuit consists of a heart rate sensor that connects to the Microcontroller (ATmega328P). The ATmega328P is triggered to send an electrical input to the LEDs and vibrating motor once the heart rate sensor detects a heartbeat > 145 BPM. The Atmega328P processes the information and sends the data to the 433 MHz RF transmitter module to be transmitted. The data is transmitted wirelessly as an Electromagnetic signal to the 433MHz RF receiver module. The 433MHz receiver module receives the data from the transmitter and sends it to the other Microcontroller (ATmega328P) for processing on the receiving end. The Atmega328P sends an electrical signal to the LEDs and vibrating motor to go HIGH. When the vibrating motor goes off, and the LEDs blink, it serves as an indication to the deaf parent that the baby is crying. The block diagram in figure 3.1 and 3.2 illustrates the transmitting and receiving process.

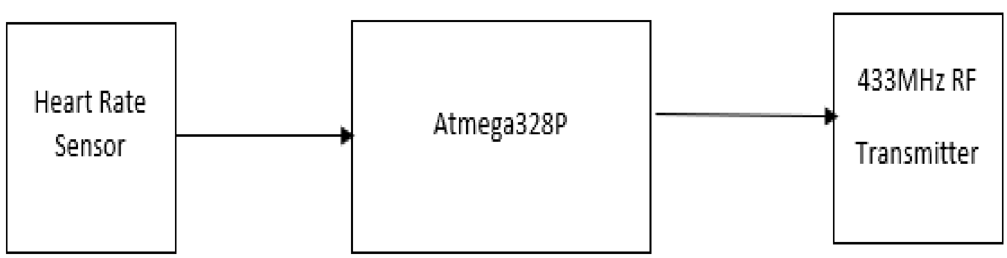

Figure 3.1: Transmitter Block Diagram

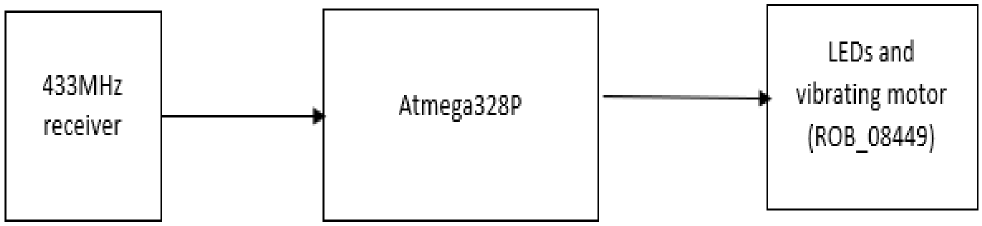

Figure 3.2: Receiver Block Diagram

3.3.1 The 433MHz RF Module

The 433MHz RF module is an ISM (industrial scientific medical) module that operates in the 433MHz RF frequency band. This module consists of a transmitter and a receiver module. Figure 3.3 is the 433MHz transmitter module. It consists of a surface acoustic wave (SAW) resonator whose carrier wave operates at 433MHz and a transistor acting as a switch and an antenna. A logic high across the transistor means data is present and low means that information is not absent. Sending data is done using Amplitude shift keying (ASK)[6]. The electromagnetic data (carrier wave plus data wave) travels through the antenna. As illustrated in figure 3.3, the data pin of the transmitter module goes to the Microcontroller while the GND and VCC pin goes to power supply[7]. In figure 3.4 is the receiving device; it comprises an RF tuner circuit, op-amps for amplifying the received signals, and a phase-locked loop for better-decoded output and noise reduction. Unlike the transmitter module, the receiver module has four legs. GND, VCC, and two center data pins are tied together. Therefore either of them can be used to be connected to the Microcontroller [7]. The 433MHz module is very cheap and requires 5V to operate. It is a low power consumption module; hence it is cheaper to run. Using this module with the Arduino code requires the library Radiohead [6].

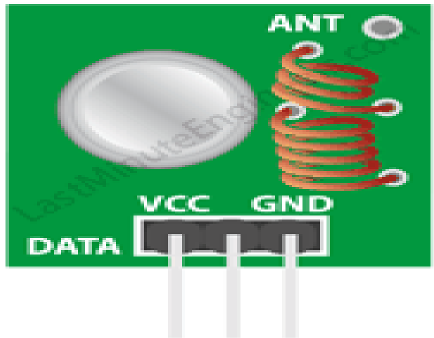

Figure 3.3: 433MHz Transmitter Module

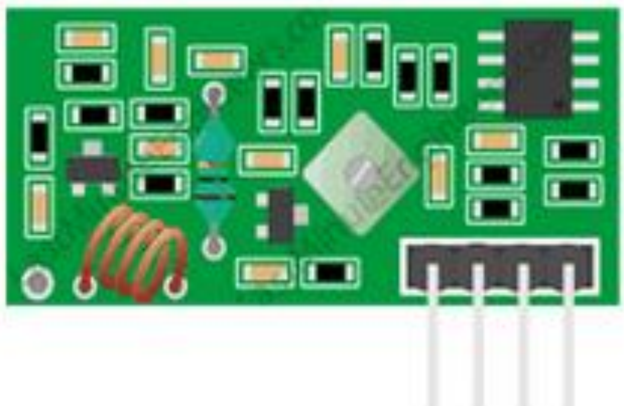

Figure 3.4: 433MHz Receiving module

3.3.2 Pulse Sensor

Figure 3.2 below is a pulse sensor/heartbeat sensor; it is used to check the heartbeat of a person either on their earlobe or index finger. It operates on the photoplethysmography principle [11]; that is, it measures the change in the volume of blood in the organ of a person as their heart beats. A very bright LED or IR LED is used to shone light on a person's finger or earlobe, and a light detector such as a photodiode or an LDR picks the output. The light reflected when the light sensor captures the flow of blood when the heart contracts or relaxes, is different. The difference between the amount of light reflected is measured and converted into an electrical pulse [10] signal. The electrical pulse from the light detector goes through a capacitor that blocks the DC component of the message and then fed into the non- inverting input of the first op-amp. This op-amp amplifies the signal with an amplification factor of 1001[8]. The output of the first amplifier goes to the second amplifier, which acts as a comparator. The output of the comparator or ADC triggers a transistor, which sends the digital signal to the Microcontroller to check the BPM. BPM= 60×f (pulse frequency) [11].

Figure 3.5: Pulse sensor

3.3.3 Vibrating Motor

A vibrating motor is a small miniature motor that vibrates. It can operate in a range of applications such as gamepads and mobile phones. These brushless DC motors are the size of a coin; hence they do not suffer from commutator wear. They consist of an internal IC, which makes it easier for them to integrate. They are also called haptic feedback actuators, and they are known for their remarkable rise and stop times [13].

Figure 3.6: Vibrating motor

3.3.4 ATMega328P Microcontroller

Figure 3.7 is a picture of an ATMega328P. ATMega328P microcontroller belongs to the family of Atmel chips. It is quite affordable and is a low power consumption device. It is an eight-bit AVR (advanced and RISC) microcontroller with 28-pins and requires its 5V supply to work [15]. The ATMega328P is the same microcontroller chip that is on Arduino Uno boards. To be used without the Arduino Uno board, it requires peripherals such as a 16MHZ crystal oscillator, which is responsible for clock pulse. The analog level for this Microcontroller is between 0 to 1023 digital values. This Microcontroller consists of 14 digital input/output pins, 6 PWM (Pulse width Modulation) pins, six analog input pins [14]. It also has a reset pin that is responsible for programs to rerun or start over.

Figure 3.7: ATMega328P

3.3.5 LEDs

Light-emitting diodes or LED for short is an electrical component that produces light by using semiconductors and Electroluminescence. Light-emitting diode has two major broad groups, namely LEDs and OLEDs (organic LED). In this paper, the focus will be on the LEDs because they are used as indicator lights. Another important characteristic of an LED is that it consumes a small amount of current usually between 30 to 60mwatts. LEDs work by having

Electroluminescence in a semiconductor material [17]. According to R.Tiwari[16], Electroluminescence is an optical and electrical phenomenon where a material emits light in response to an electric current or an electric field passing through it. The current in motion causes electrons through the material to fill up electron holes [16]. Figure 3.8 is a well-labeled diagram of an LED.

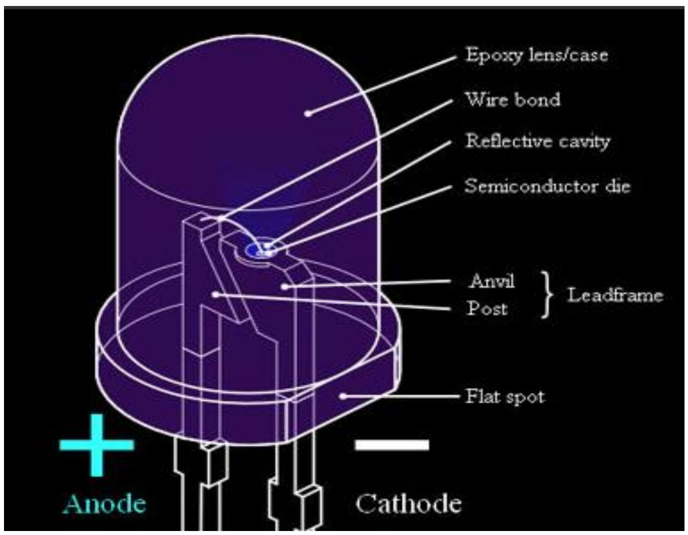

Figure 3.8: light Emitting Diode (LED)

3.3.6 9V Lithium Battery Power Supply

Most of the 9V lithium batteries are primary batteries; this means that they are not rechargeable. These batteries have a very long shelf life and can last between 5 to 10 years in a low power consuming electronics device such as toys, cameras, etc. [18]. 9V lithium batteries, just like other power sources, have two terminals. The cathode or positive terminal of the battery and the anode or negative terminals of the cells. Batteries operate when two chemicals are prevented from mixing by using a barrier. It leads to electron imbalances in the system; hence the electrons (-) flow to the positive terminals causing a voltage difference to occur, leading to current flow [19]. In this project, a 9V lithium battery connects to the input of a 9V to 5V battery converter. The 5V outputted from the battery converter powers the Microcontroller. Figure 3.9 shows a 9V lithium battery power supply.

Figure 3.9: 9V Lithium Battery

3.4 Methods

3.4.1 Circuit Diagrams

In figure 3.3 and 3.4 below, the hardware circuitry for both the transmitter and the Receiver illustrated. In Figures 3.5 and 3.6, the battery and the sensor circuits shown.

Transmitter Circuit

The transmitter makes use of three main components. Figure 3.10 below is the circuit of the transmitter circuit. The ATMega328P Microcontroller has peripherals such as the resonator as the clock source, capacitors, and a resistor. The Microcontroller, together with its peripherals, works just like the Arduino Uno board. The transmitter has the 433MHz RF transmitter module for communication. The pulse sensor built from scratch is also in figure 3.10. Figure 3.11 shows the transmitter circuit, but with an existing pulse sensor, this circuit performs the same function as the one in figure 3.10. The ATmega328P will process the readings of the heart rate sensor and transmit when the heart rate exceeds 145BPM. This project ensures that transmission is possible for at least a 100m.

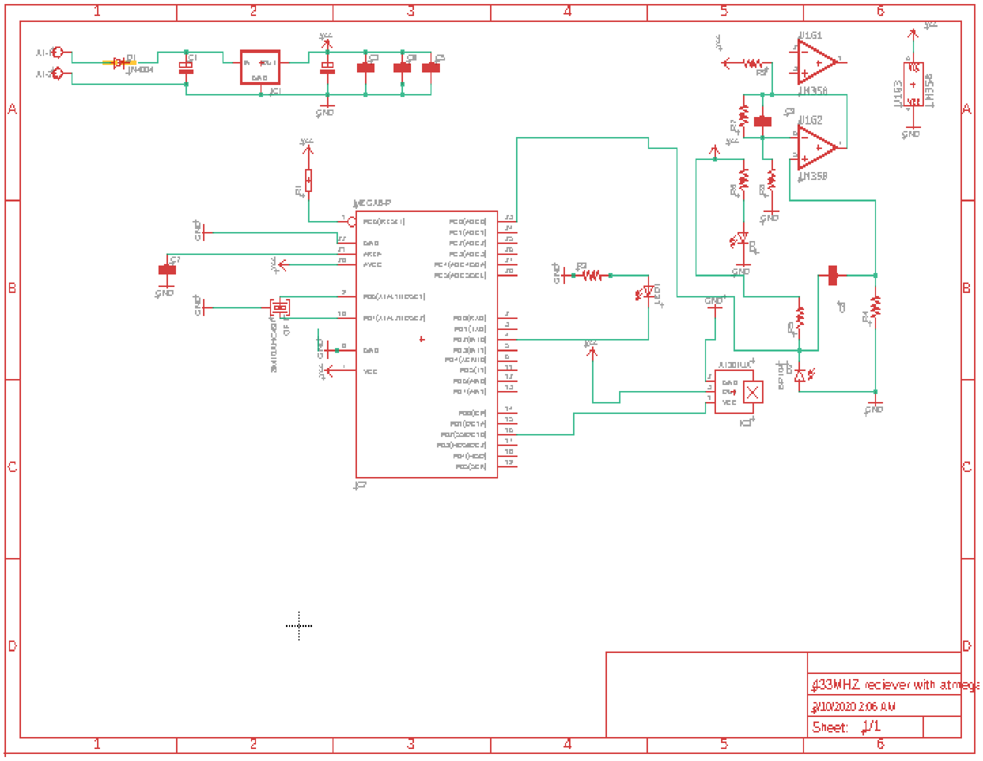
Figure 3.10: Transmitter Circuit with pulse sensor circuit built from scratch

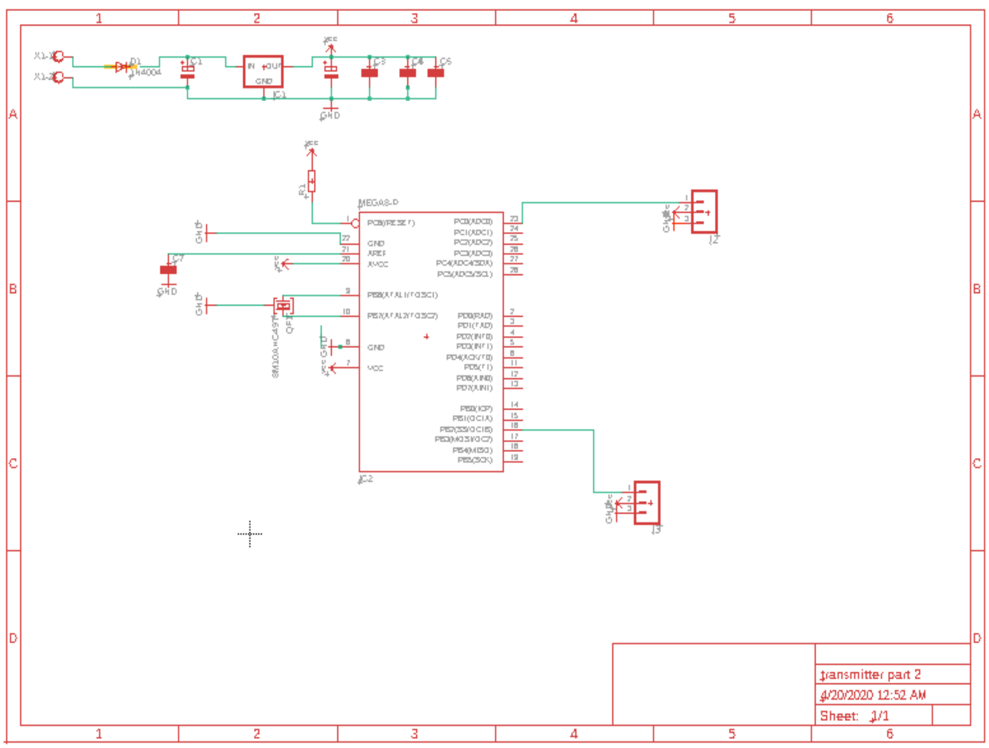
Figure 3.11: Transmitter Circuit with existing pulse sensor

Detailed steps are listed below to aid in understanding the connections being made in figure 3.3, note that the steps listed below are the connection for when the code is on the ATMega328P transmitting microcontroller. Other measures for boot-loading and programming the ATMega328P will be explained in the software design section of this report.

1. Connect one end of a 10k resistor to pin 1 of the ATMega328P and the other end to VCC (5V).
2. Connect a 0.1µF capacitor across pin 7 and 8 of the ATMega328P, also connect pin 7 to VCC and pin 8 to GND.
3. Connect pin 9 and 10 to the leftmost and rightmost pin of the resonator; note that it does not matter which pins connect on either side. Then connect the center pin to ground.
4. Connect pin 18 of the ATMega328P to the data pin of the 433MHz RF transmitter module. Also, compare the center pin of the 433MHz RF transmitter module to VCC and the GND pin to GND of the ATMega328P.
5. Connect a 0.1µF capacitor across pin 20 and 22 of the ATMega328P, also connect pin 20 to VCC and pin 22 to GND.
6. Connect the signal pin of the pulse sensor to pin 23 of the ATMega328P. Also, connect the VCC and GND of the pulse sensor to VCC and GND of the ATMega328P, respectively.

With the Microcontroller successfully coded, the transmitter circuit is ready for operation.

Receiver Circuit

The receiver circuit in figure 3.12 has three main parts, just like the transmitter circuit. The ATmega328P processes the readings of the receiving 433MHz RF module. The processed signal activates the vibrating motor and LEDs when the heart rate sensor reads a beat higher than

145BPM; this indicates a cry. Just below the receiving circuit diagram, some steps describe the connections made in the circuit in figure 3.12.

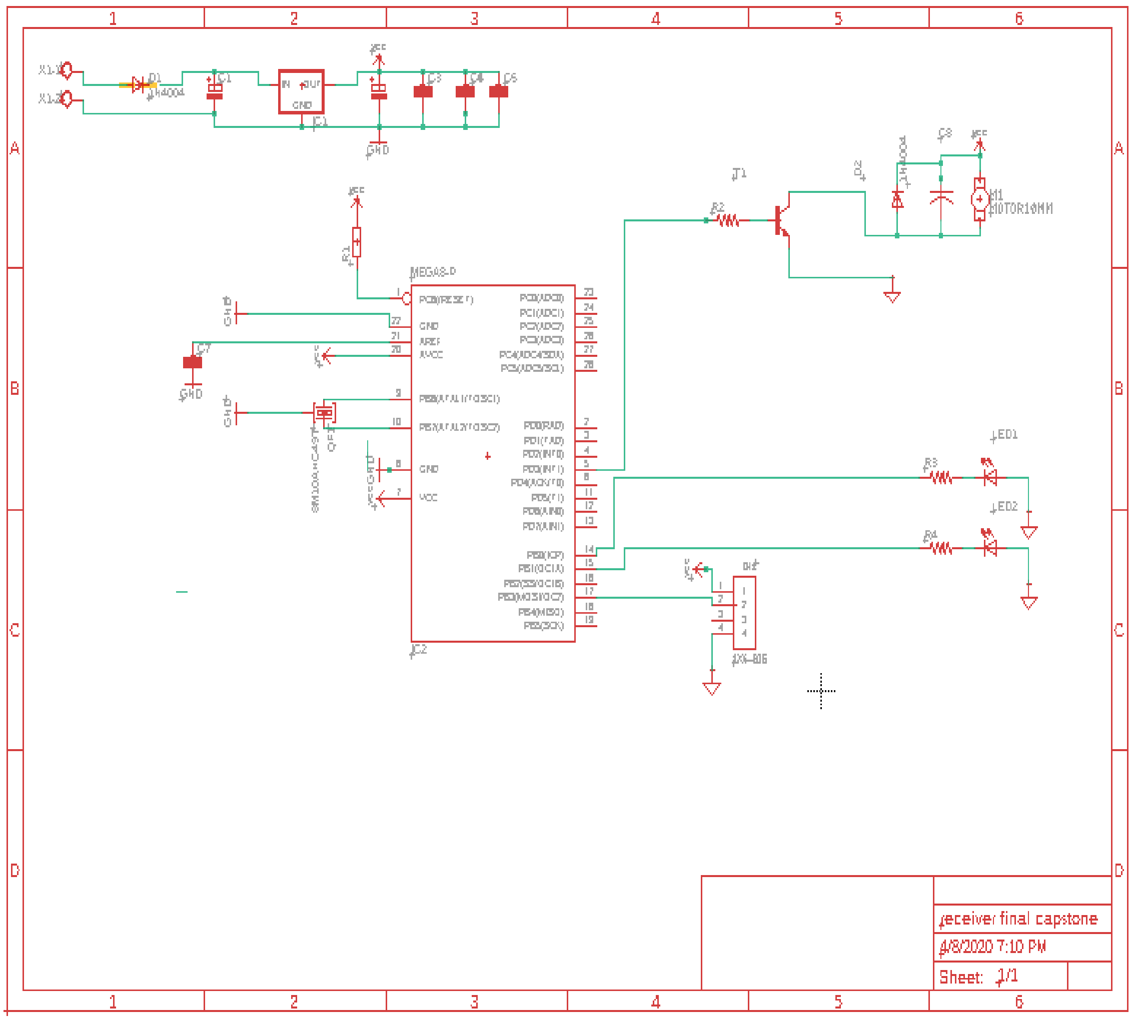

Figure 3.12: Receiver Circuit

1. Connect one end of a 10k resistor to pin 1 of the ATMega328P and the other end to VCC (5V).
2. Connect a 0.1µF capacitor across pin 7 and 8 of the ATMega328P, also connect pin 7 to VCC and pin 8 to GND.

3. Connect pin 9 and 10 to the leftmost and rightmost pin of the resonator; note that it does not matter which pins connect on either side. Then connect the center pin to ground.
4. Connect pin 14 to the positive leg of an LED and connect the negative leg of the LED to one end of a 1kΩ Resistor, and the other end of the resistor to GND.
5. Connect pin 15 to the positive leg of an LED and connect the negative leg of the LED to one end of a 1kΩ Resistor, and the other end of the resistor to GND.
6. Connect pin 17 of the ATMega328P to the data pin of the 433MHz RF receiving module. Also, connect the center pin of the 433MHz RF receiving module to VCC and the GND pin to GND of the ATMega328P.
7. Connect a 0.1µF capacitor across pin 20 and 22 of the ATMega328P, also connect pin 20 to VCC and pin 22 to GND.
8. Connect pin 5 to one end of a 1kΩ resistor, then connect the other end of the resistor to the base of a 2N2222 transistor. Connect the emitter of the transistor to ground, and the collector to the positive terminal of a 1N4001 diode. Next, connect the positive end of the diode to GND. Next, connect the negative end of the diode to one end of a 0.1µF capacitor which also goes to VCC, the other end of the capacitor goes to GND. Finally, connect the positive pin of the vibrating motor (ROB_08449) to VCC and the negative end to GND.

9V to 5V Battery Converter Circuit

The diagram in figure 3.12 illustrates a 9V to 5V battery converter circuit. The importance of this circuit is to convert the 9V that is supplied by the 9V lithium battery to 5V. The 5V is for powering the ATMega328P Microcontroller. The capacitors in the circuit are responsible for smoothening out the voltage to prevent oscillations in the voltage signal; the more Capacitors added, the smoother the electrical signal. Figure 3.13 shows the connections, and just below figure 3.13 are the steps to achieve this circuit.

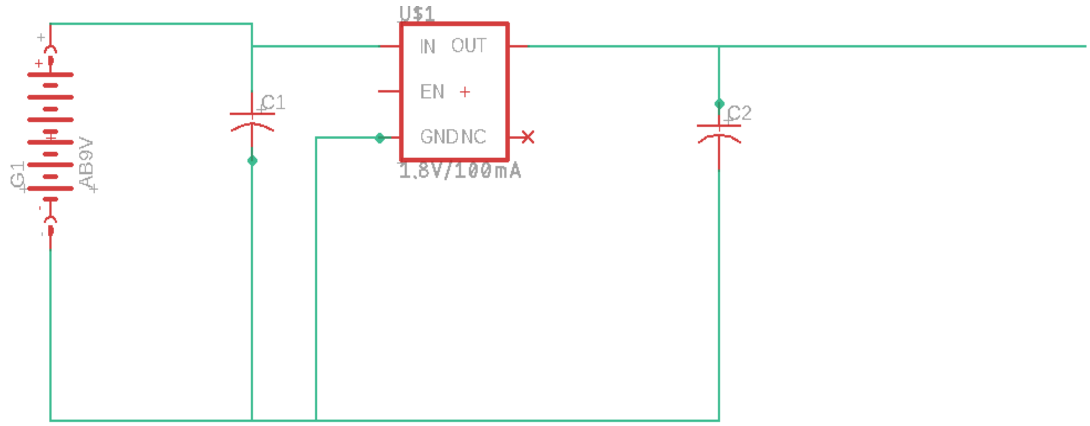

Figure 3.13: Battery Circuit

1. Connect the positive terminal of the 9V battery to one end of a 1.0µF capacitor and then connect this end to the input pin of the L7805 series voltage regulator. Then connect the negative end of the capacitor as well as the negative terminal of the battery to GND. The middle pin of the L7805 also goes to GND.
2. Connect the output pin of the L7805 voltage regulator to a 1.0µF capacitor and this same end to the positive pin (VCC/5V) to the ATMega328P Microcontroller. The other end of the capacitor goes to GND of the ATMega328P Microcontroller.

Sensor Circuit

The sensor circuit in figure 3.6 illustrates a pulse sensor. The pulse sensor consists of an Infrared LED (IR LED), a photo Diode, resistors, capacitors LM358 Op-amp IC, and a BC547 transistor. The IR LED is the transmitter, and the photodiode is the Receiver. When the person puts their finger in between the IR LED and the photodiode, the IR LED shines a light on the person's finger. When the heartbeats, the amount of blood rushing through the veins of a person varies. Hence, the amount of light passing through the finger also varies. To get a value of a resistor that will prevent the IR LED from getting damaged, we use the formula in equation 1 to find the forward current.

$$I_F = \frac{(V_P - V_F)}{R_7} \qquad (3.1)$$

Where V_P is 5V and R_7 is the resistor connected in series with the IR LED, and V_F is the forward voltage drop on the LED. This difference is picked by the photodiode, which then sends its output to a high pass filter. The capacitor in the high pass filter blocks the DC component from the output. The frequency cut is calculated using the formula in equation two below [20].

$$F_C = \frac{1 \times \pi \times R6 \times C}{2} \qquad (3.2)$$

The output from the filter goes to the first op-amp, which is a non-inverting amplifier with a gain of 1001.The output of the first op-amp goes to the second op-amp, which acts as a comparator. The output from this comparator triggers the transistor [8], which then sends the signal to the ATMega328P.

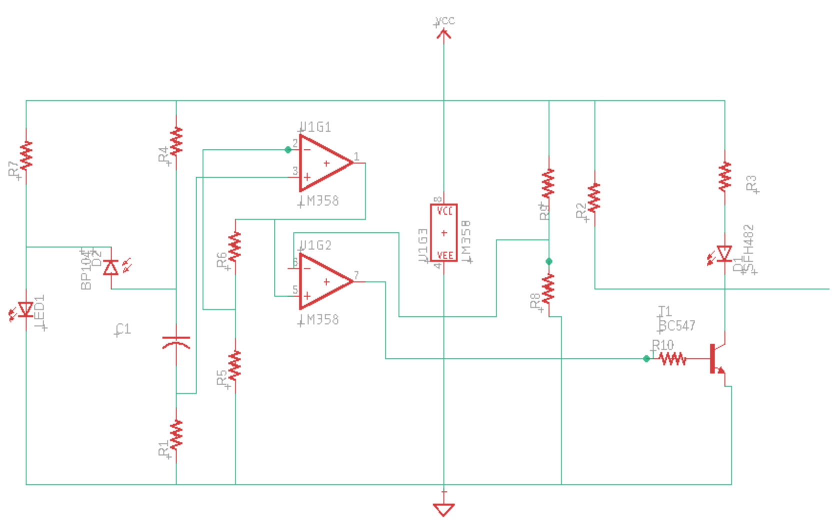

Figure 3.14: Pulse Sensor Circuit

3.4.2 Software Design

The software design consists of first boot-loading the ATMega328P and then programming it with Arduino. Operating systems used include windows Laptop, programming languages to be used include C in Arduino.

The flowchart in figure 3.15 illustrates the operation of the system as follows. When the baby cries, his/her heart rate increases. An increase in the heartbeats beyond 145BPM depicts that the baby is crying. Hence the transmitting signal is sent to the Receiver. The MCU (microcontroller unit) on the receiver side, processes the signal and turn on a vibrating motor and LEDs. Once the baby stops crying, the device will stop vibrating and blinking light.

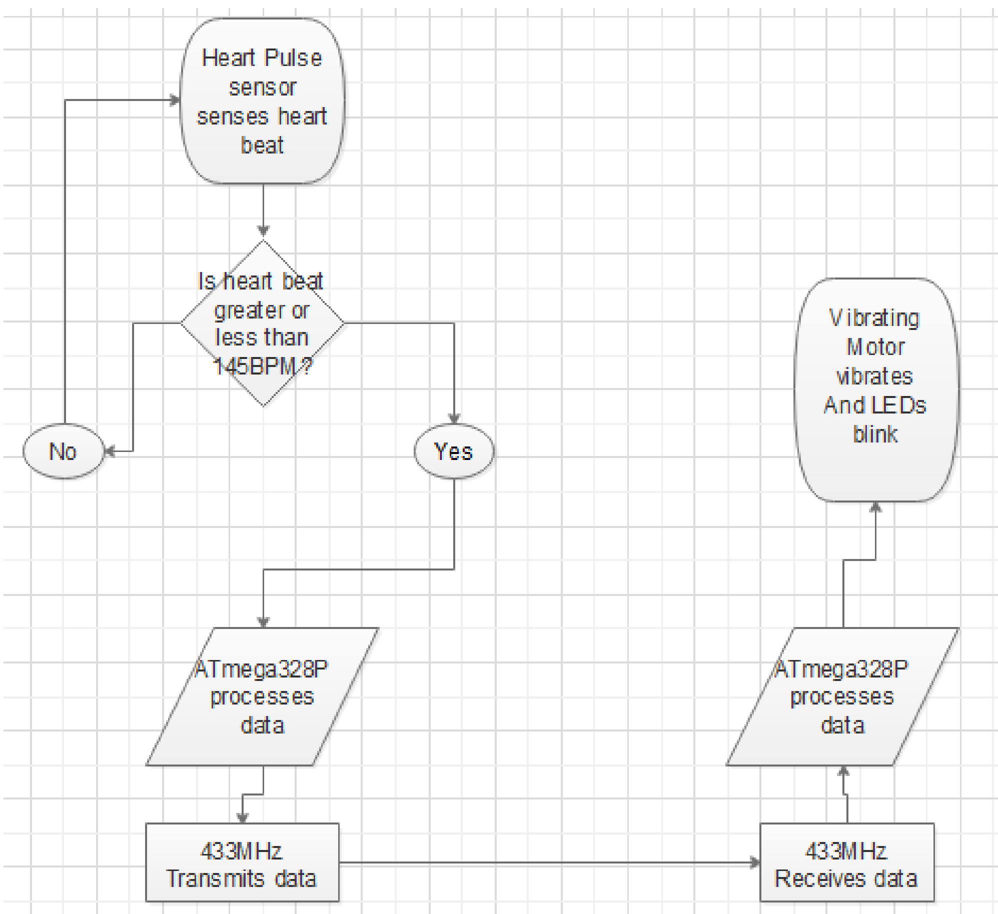

Figure 3.15: Flow Chart

Boot-loading the ATMega328P

Boot loading of the ATMega328P requires the same circuit for the receiver circuit in figure 3.11, except that the 433MHz RF, the vibrating motor, and LEDs are not connected. Then follow these steps:

1. Connect the end of the resistor to pin 1 of the ATMega328P and VCC. Also, connect it to the reset pin of the Arduino.
2. Next connect pins 17, 18, and 19 of the ATMega328P Microcontroller to pins 11, 12, and 13 of the Arduino Uno board, respectively. Ensure that the Arduino Uno board should not have any ATMega328P on it.
3. Then open Arduino software, and navigate through file->examples-> and select Arduino ISP-> then navigate to Tools->Programmer->Arduino as ISP. Then select Tools->Burn Bootloader. Wait for the code to upload and check if there are no error messages. If it was successful, then the device has been successfully bootloader. See the image of the screen after boot-loading in figure 1 of the appendix.

Programming the ATMega328P

The connection for the bootloader is the same except that pin 17,18 and 19 are not connected. Instead, pin two is connected to the RX pin on the Arduino board and pin 3 to the TX pin. The Arduino board should not have any ATMega328P on it. Run the code, and if successful, the Microcontroller is ready to be packaged.

Chapter 4: Results and Discussion

4.1 Introduction

After all the necessary connections and programming, putting the device in a suitable package is the next step. Figures 4.1 and 4.2 show a 3D design of the receiving device that will be worn by the deaf parent. Just as illustrated in the 3D models, the LEDs will show at the top of the box (length=5.5cm, width= 3.5cm and height= 3cm). So that they could be visible. The small box, as shown below, will house the 9V battery and the receiving printed circuit board (PCB) with the vibrating motor on the inside, the LEDs will be the only visible things on the outside of the box.

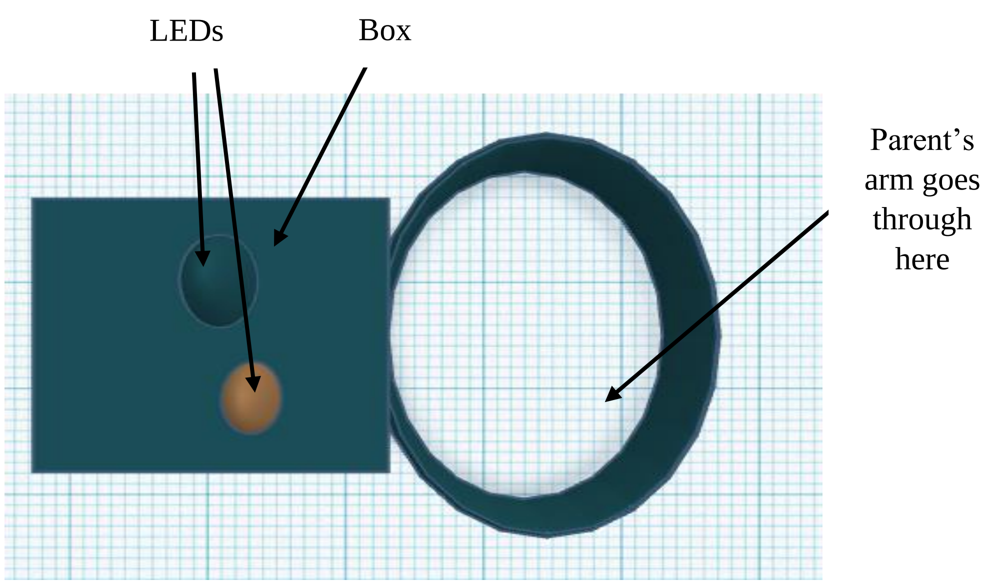

Figure 4.1: Receiver Product Design

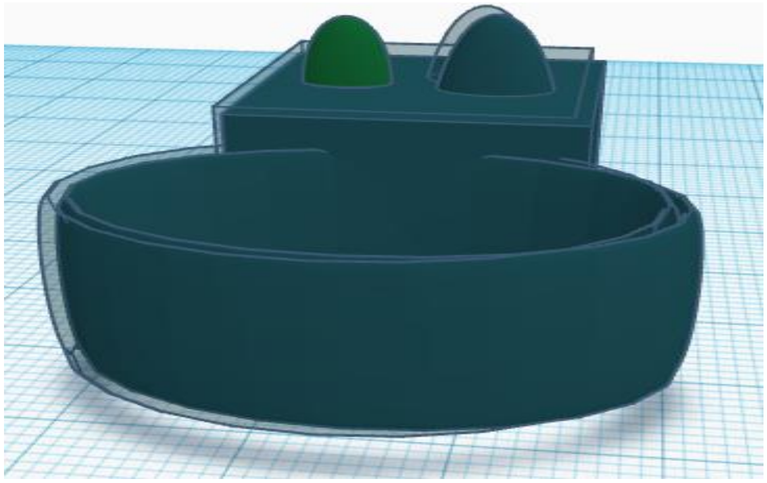

Figure 4.2: Front view of Receiver Product Design

Figure 4.3 shows the 3D front view of the transmitter circuit to be worn by the baby and figure 4.4. shows the same design but from the top perspective. The idea of the design is a bracelet with a ring. This design will be made with a soft cotton cloth to ensure the baby is very comfortable.

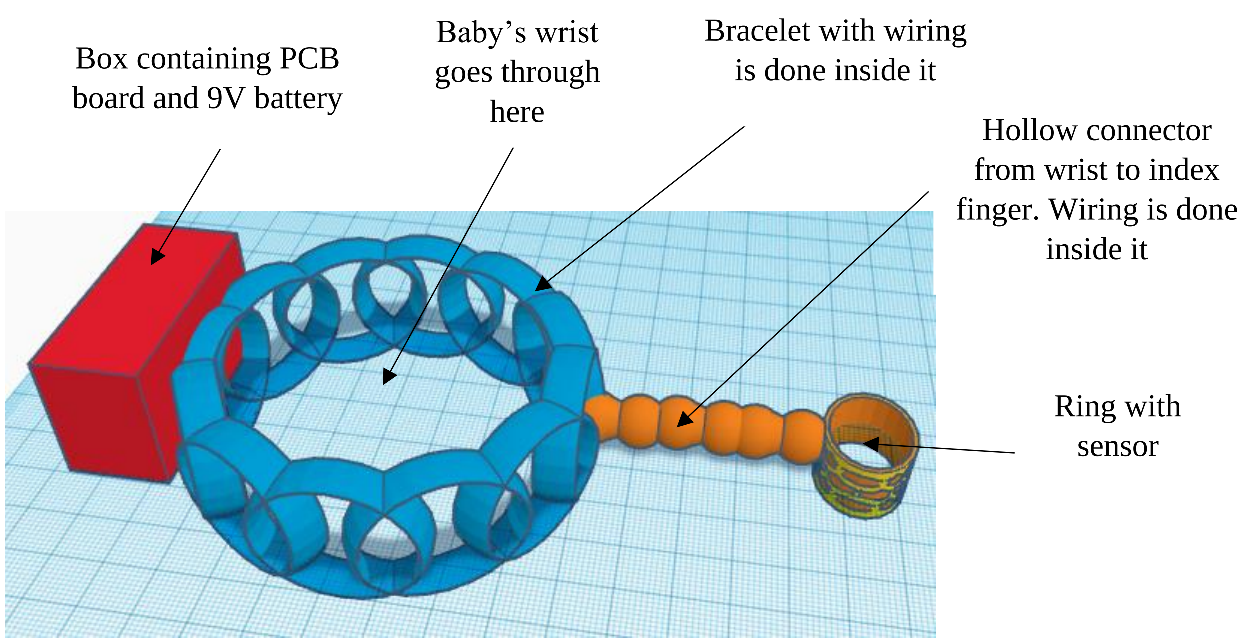

Figure 4.3: Front view of the 3D model worn by a baby

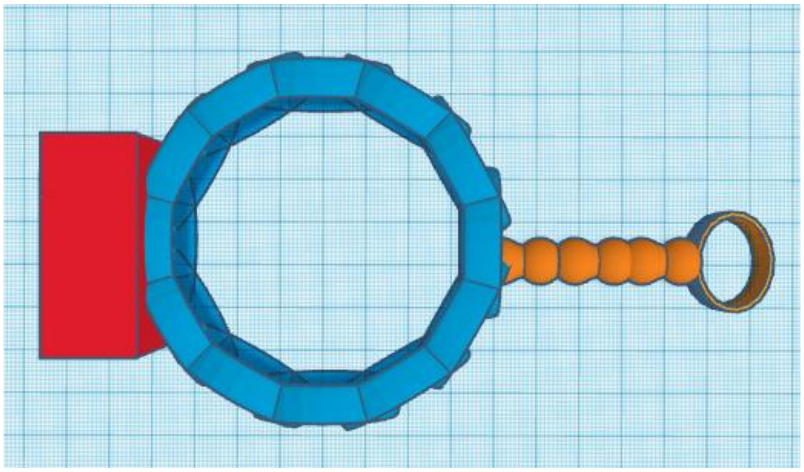

Figure 4.4: Top view of the 3D model worn by a baby

As shown in the image, the box contains the PCB and the 9V battery. The sensor wires come out through a small hole. The wiring is done within the bracelet and the hollow connector such that no wires will be seen from the outside or left hanging. The device will be in soft cotton cover for the absolute comfort of the baby.

4.2 Results from hardware design

The hardware design consists of the heart pulse sensor that reads the physical heartbeat of a baby every 10 seconds and converts it into electrical signals that the ATmega328P can process. After processing the signal, the ATmega328P sends it to the 433MHz RF transmitter module that converts the electrical signals into electromagnetic signals. The electromagnetic signal is transmitted wirelessly to the receiving 433MHz RF receiving module, which gets the data via an antenna. The module converts the electromagnetic signals to an electrical signal to be processed by the ATMega328P. After processing the signal, the ATMega328P sends an electrical voltage signal to the vibrating motor and LEDs to trigger vibration and blinking, respectively. Each of the ATMega328P receives 5V from a 9V source battery, which is regulated, L7805 voltage regulator.

With the connections correctly done and the microcontrollers successfully boot-loaded, followed by uploading the codes, the beat per minute of the baby sums up. If the baby is crying, then the data is transmitted wirelessly, and the LEDs and the vibrating motor will come on.

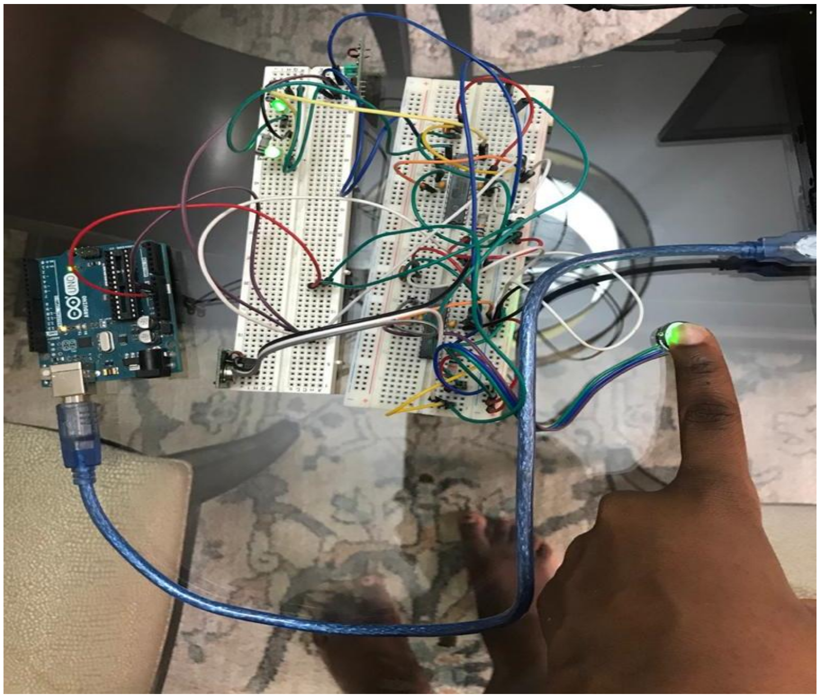

Figure 4.5: Image showing hardware implementation when the pulse sensor is touched.

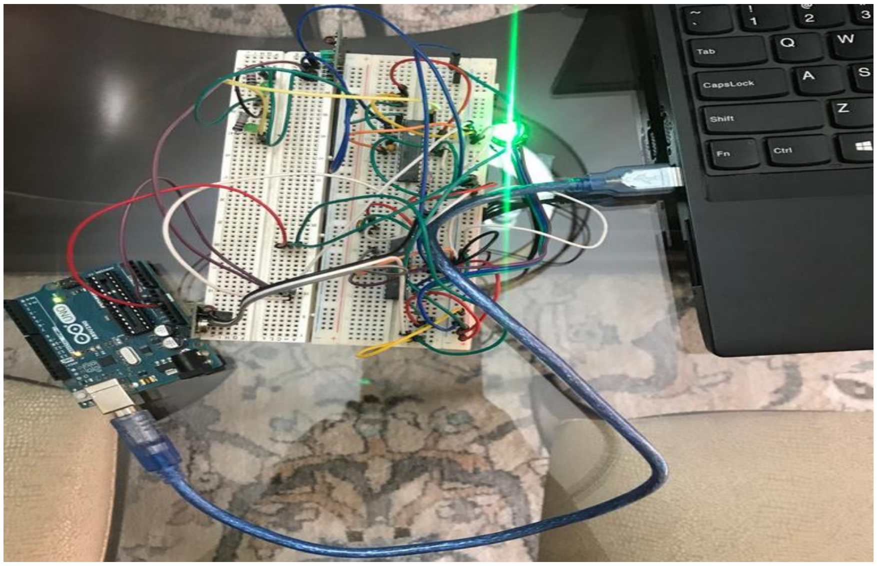

Figure 4.6: Image showing hardware implementation when the pulse sensor is not touched.

For the sake of testing, the beat per minute is at zero. Hence once the sensor detects a heartbeat, it sends signals to the Microcontroller, which then triggers the 433MHz to transmit. Once the Receiver receives data, the LEDs blink, as shown in figure 4.1, and figure 4.2 shows the circuit when no heartbeat read.

4.2.1 9V to 5V battery converter Circuit Implementation

Figure 4.3 shows the connected circuit for the 9v to 5v battery converter circuit.

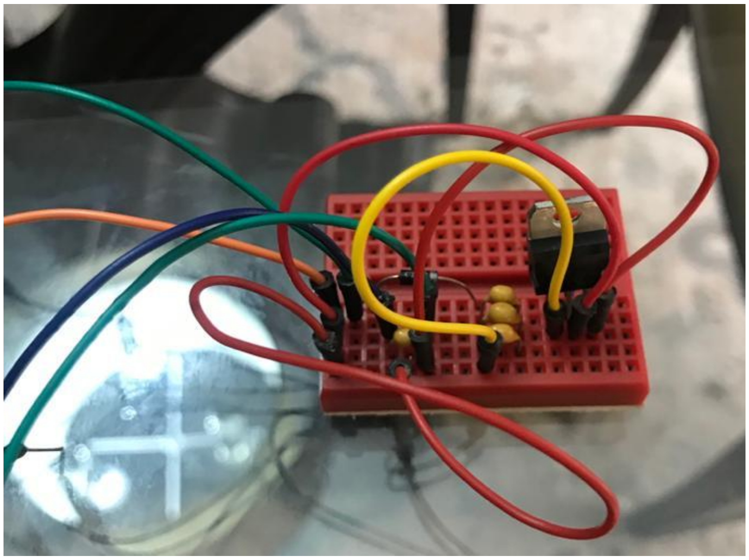

Figure 4.7: 9V to 5V converter circuit

4.2.2 Pulse Sensor Built from Scratch Implementation

Figure 4.4 illustrates the implementation of a pulse sensor built from scratch and the transmitter and receiver circuits.

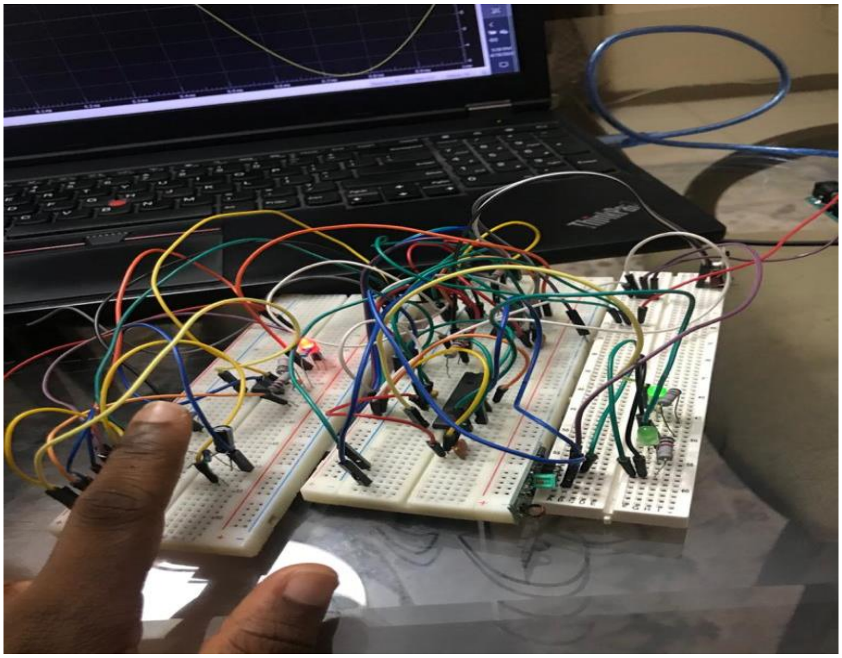

Figure 4.8: Implementation of a pulse sensor built from scratch

4.2.3 PCB Designs

In chapter 3, figure 3.10 and figure 3.11 illustrated as the transmitter and receiver circuit design are updated to a PCB circuit design. The board design in figure 4.3 is the transmitter EAGLE PCB design after successfully routing the board with a 100% accuracy with zero errors. Figure 4.4 shows the receiver EAGLE PCB design after successfully routing the board with a 100% accuracy, 100% accuracy was obtained with zero glitches as well.

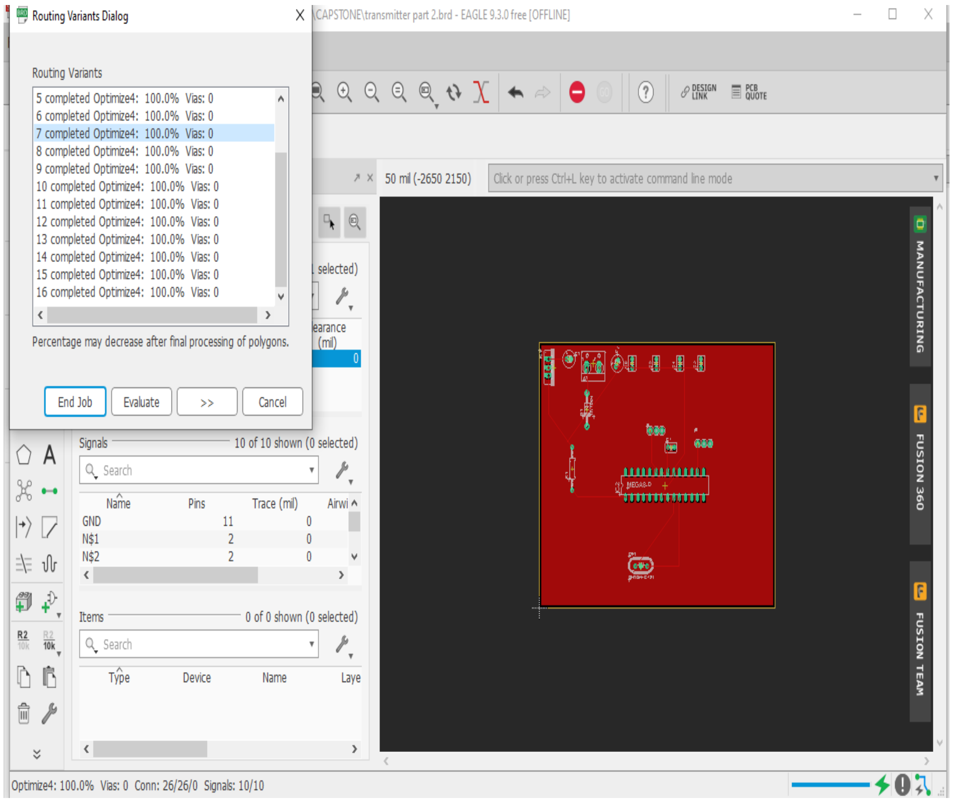

Figure 4.9: A snippet of successful routing of Transmitter PCB on EAGLE

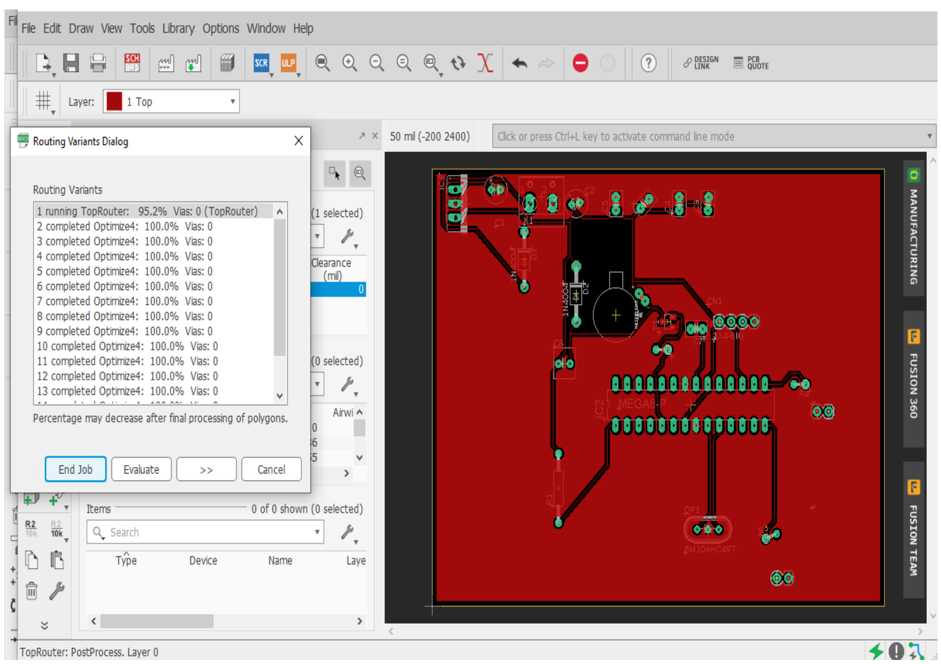

Figure 4.10: A snippet of a successful path of Receiver PCB

4.3 Results from Software Implementation

This section includes the transmitter and receiver codes snippet as implemented on Arduino IDE as well as a simulation of vibrating motor and LED on Tinkercad.

4.3.1 Codes

The transmitter and receiver codes are discussed below:

Transmitter Code

The library Radio-headmaster (RH_ASH.h) must be downloaded and imported into the Arduino IDE. Include this library in the workspace and initialize the driver. Initially, the upper threshold is equal to 594 based on the output from the sensor. The output of the sensor is read at AnalogRead (0) and stored in the integer "reading." The lower threshold is an analog read =511, which indicates the baby is at rest. The lower and upper limits and the values from readings are calibrated, as shown on lines 50 to 53. The calibrations help make sense of the results. The Timer1 and Timer2 Millis in the code are counting to update the BPM and update the serial monitor, respectively. Lines 91 to 96 sends the BPM when it is more significant than 145 beats. Figure 4.9 is a snippet of the transmitter sketch in Arduino, and the full code is in Appendix C.

Figure 4.11: Snippet of transmitter code

Receiver Code

Figure 4.10 shows the receiver code is less complicated since it is missing the vibrating motor code. The complete code is in appendix C. The RF driver is initialized in void setup, while in the void loop, the data is received and used to turn on LEDs that are connected to pin eight and pin nine.

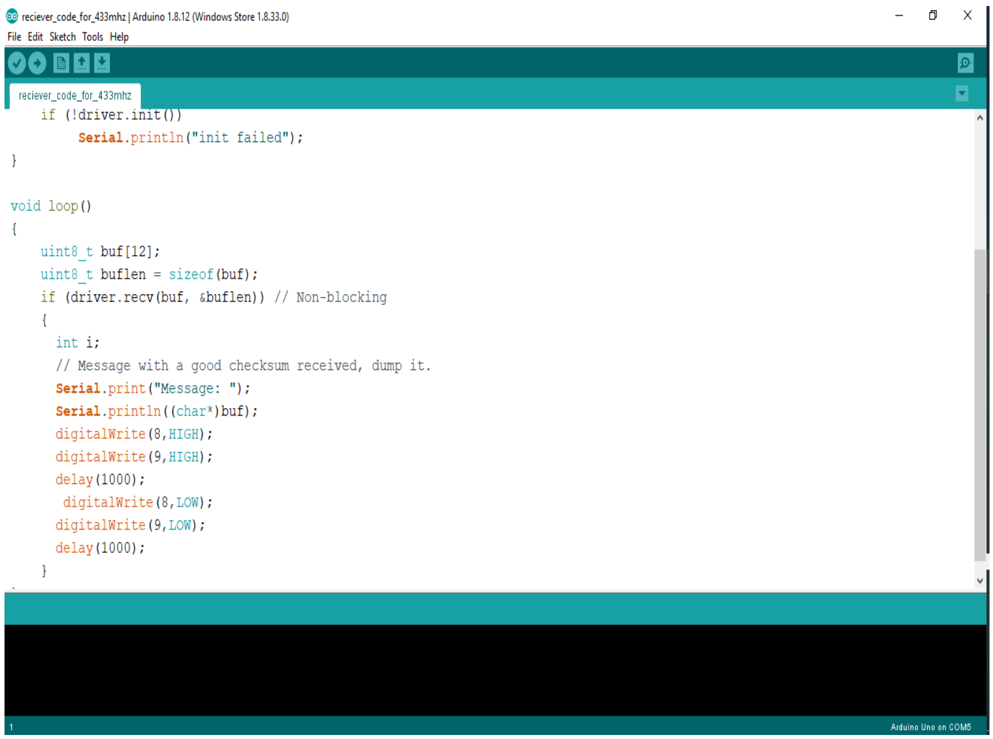

Figure 4.12: Snippet of Receiver Code

4.3.2 Tinkercad Vibrating motor and LED implementation

Due to the 2020 COVID-19 pandemic, which resulted in not acquiring the vibrating motor. A simulation using Tinkercad software to illustrate the operation of the vibrating motor is shown in figure 4.13. Tinkercad is online software that allows real-time simulation of electrical circuits. The circuit was implemented using the Arduino Uno board since Tinkercad does not allow the importation of components. Figure 4.7 shows the circuit and the code as performed on Tinkercad.

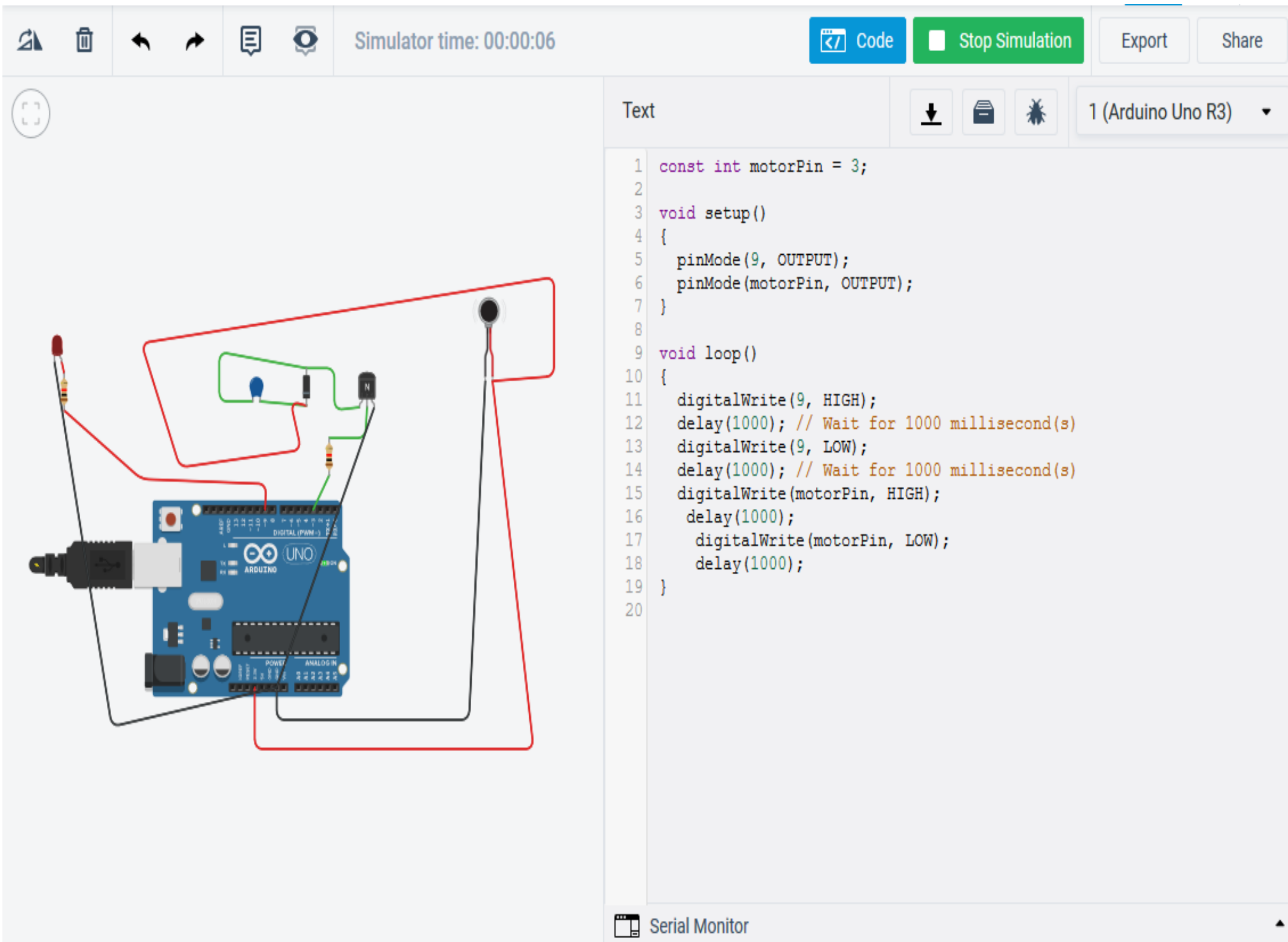

Figure 4.13: Software connections and implementation of vibrating motor and LED using Tinkercad

Chapter 5: Conclusions, Limitations and Future Work

5.1 Conclusion

This project focuses on implementing a baby monitor for financially challenged deaf parents. Implementation includes using the 433MHz RF module, a pulse sensor, a vibrating motor, and a couple of LEDs as the significant components of the system. This device consists of cheap parts that are also reasonable to run; hence it can be afforded by even low-income families. The benefits of this device include 1. An effective way for deaf parents to monitor their babies. 2. A means for deaf parents to feel more connected to their babies, and the same applies to the babies as well. 3. To give the deaf parents an option to be able to live with their baby without another person's supervision. 4. To prevent a problematic situation were a baby gets hurt and cries out for their parents, but they did not attend to the baby on time because they could not hear. Hence this device seeks to curb some if not all the problems deaf parents face when raising a young child between ages 3-30 months old.

5.2 Limitations

This device works because of inputs from the pulse sensor; hence the more accurate the pulse sensor is, the better the device will perform. Another limitation is with the power supply; in this project, a 9V battery is the power source. This 9V was stepped down to 5V to supply the Microcontroller. The voltage regulator circuit and the 9V could have been easily replaced with a 5V battery if they were easily accessible and could last long, thus preventing the need for a 9V battery. The application of this would have made the circuit much less complicated.

5.3 Future Works

This device could be made more efficient with video monitoring. Video monitoring would mean a bigger, more massive, and more complex device, but it will serve the purpose of monitoring a baby more efficiently. Using a rechargeable battery will also result in the lower running costs of the device, therefore adding this feature will be beneficial for future upgrades. Also, introducing a sock version of the device worn by a baby can be another alternative design for this invention. The parents will have an option between the bracelet and ring model and the sock model. Although the sock model will be preferable for night times to prevent it from damaging due to the baby standing on it.

REFERENCES

[1]K. Fullerton,"Deafness in Sub-Saharan Afric", *The Borgen Project*, 2013. [Online]. Available: https://borgenproject.org/deafness-in-sub-saharan-africa/. [Accessed: 11- Oct- 2019].

[2]S. Singer,"New app tells hearing-impaired parents when and why their baby is crying", *University of California*, 2018. [Online]. Available: https://www.universityofcalifornia.edu/news/app-helps-hearing-impaired-parents-know-when-and-why-their-baby-crying. [Accessed: 13- Oct- 2019].

[3]W. Liow, H. Jiang and C. Yen,"Designing a communication device for deaf parents and a hearing infant", *ResearchGate*, 2013. [Online]. Available: https://www.researchgate.net/publication/281719909_Designing_a_communication_device_for_deaf_parents_and_a_hearing_infant. [Accessed: 13- Oct- 2019].

[4]B. Vaughn and A. Sroufe,"The Temporal Relationship between Infant Heart Rate Acceleration and Crying in an Aversive Situation", *jstor.org*. [Online]. Available: https://www.jstor.org/stable/1129435?seq=2#metadata_info_tab_contents. [Accessed: 09- Nov- 2019].

[5] "Pulse | Medical Tests | UCSF BenioffChildren's Hospita", *Ucsfbenioffchildrens.org*, 2012. [Online]. Available: https://www.ucsfbenioffchildrens.org/tests/003399.html. [Accessed: 09- Nov- 2019].

[6]"" Insight Into How 433MHz RF Tx-Rx Modules Work & Interface with Arduin", *Last Minute Engineers*. [Online]. Available: https://lastminuteengineers.com/433mhz-rf-wireless-arduino-tutorial/. [Accessed: 29- Mar- 2020].

[7] "Wireless RF Module | RF Transmitter and Receiver | Latest Application", *ElProCus - Electronic Projects for Engineering Students*. [Online]. Available: https://www.elprocus.com/rf-module-transmitter-receiver/. [Accessed: 29- Mar- 2020].

[8] "Heartbeat Sensor using Arduino (Heart Rate Monitor", *Electronics Hub*, 2017. [Online]. Available: https://www.electronicshub.org/heartbeat-sensor-using-arduino-heart-rate-monitor/. [Accessed: 22- Dec- 2019].

[9] "How to use and control a Vibration Motor using Arduino UN", *Tinkbox*, 2015. [Online]. Available: http://tinkbox.ph/learn/tutorial/how-use-and-control-vibration-motor-using-arduino-uno. [Accessed: 22- Dec- 2019].

[10]P. Pawar,"Heart rate monitoring system using IR base sensor & Arduino Uno - IEEE Conference Publication", *Ieeexplore.ieee.org*, 2015. [Online]. Available: https://ieeexplore.ieee.org/document/7057005/authors#authors. [Accessed: 24- Feb- 2020].

[11] "Heart Beat Sensor - How to Measure Heart Beat: Working and Application", *ElProCus - Electronic Projects for Engineering Students*. [Online]. Available: https://www.elprocus.com/heartbeat-sensor-working-application/. [Accessed: 24- Feb- 2020].

[12]D. Das,"How to Interface RFM69HCW RF Module with Arduino for Wireless Communication", *Circuitdigest.com*, 2019. [Online]. Available:

https://circuitdigest.com/microcontroller-projects/how-to-interface-rfm69hcw-rf-module-with-arduino. [Accessed: 24- Feb- 2020].

[13]"Vibration Motors - Precision Microdrives", *Precisionmicrodrives.com*. [Online]. Available: https://www.precisionmicrodrives.com/vibration-motors/. [Accessed: 24- Feb- 2020].

[14]Q. Dyan, J. Rezalina, M. Mead and R. Maramba,"Wireless Automated Fire Detection System on Utility Posts Using ATmega328", *Ieeexplore.ieee.org.elibraryashesi.remotexs.co*, 2012. [Online]. Available: http://ieeexplore.ieee.org.elibraryashesi.remotexs.co/document/8666329. [Accessed: 07- Apr- 2020].

[15]"Atmega328 Pinout", *Learningaboutelectronics.com*, 2018. [Online]. Available: http://www.learningaboutelectronics.com/Articles/Atmega328-pinout.php. [Accessed: 07- Apr- 2020].

[16]R. Tiwari, V. Dubey and S. Dhoble, *Emerging synthesis techniques for luminescent materials*, 1st ed. Hershey, Pennsylvania: IGI Global, 2018, p. 415.

[17]"LED Lights - How it Works - History", *Edisontechcenter.org*, 2020. [Online]. Available: https://edisontechcenter.org/LED.html. [Accessed: 08- Apr- 2020].

[18]T. Escarda and N. Lewis, *Rechargeable alkaline household battery system, Rayovac Corporation, Renewal*, 1st ed. Cincinnati, Ohio: U.S. Environmental Protection Agency, Office of Research and Development, National Risk Management Research Laboratory, 1999, p. 8.

[19]H. Silver, *Do-it-yourself circuitbuilding for dummies*, 1st ed. Hoboken, N.J.: Wiley Pub., Inc., 2008, p. 318.

[20] "Activity: Heart Rate Monitor Circuit [Analog Devices Wiki", *Wiki.analog.com*, 2017. [Online]. Available: https://wiki.analog.com/university/courses/alm1k/alm-lab-heart-rate-mon. [Accessed: 08- Apr- 2020].

[21]M. Hussain,"Getting BPM from the given cod", *Arduino Stack Exchange*, 2018. [Online]. Available: https://arduino.stackexchange.com/questions/43956/getting-bpm-from-the-given-code. [Accessed: 20- Apr- 2020].

APPENDIX

Appendix A

Table 1: PUGH CHART FOR WIRELESS RF MODULE

CRITERIA	BASELINE	WEIGHT				
	INFRARED		WIFI	BLUETOOTH	GSM	433MHz
Distance of 100m	0	5	**5**	**-5**	**5**	**5**
Delay of 10s	0	1	**1**	**1**	**1**	**1**
Cost of purchasing module	0	2	**-2**	**-2**	**-2**	**2**
Portability	0	3	**-3**	**-3**	**-3**	**-3**
Rural area Flexibility	0	4	**-4**	**+4**	**-4**	**+4**
Human friendly	0	7	**+7**	**+7**	**+7**	**+7**
Cost of usage	0	6	**+6**	**+6**	**-6**	**+6**
TOTAL			**10**	**8**	**-2**	**18**

Table 2: PUGH CHART FOR MICROCONTROLLER

CRITERIA	BASELINE	WEIGHT				
	Arduino		Node MCU	Freedom board	Raspberry Pi	ATMega328P chip
Short response time 1s	0	3	**3**	**3**	**3**	**3**
Weight	0	6	**-6**	**-6**	**-6**	**6**
Cost of purchasing	0	1	**1**	**1**	**1**	**-1**
Expertise	0	5	-5	**-1**	**-5**	**5**
Low power consumption	0	4	**4**	**4**	**4**	**4**
Few external components	0	2	**2**	**2**	**2**	**-2**
TOTAL	0		**-1**	**3**	**-1**	**9**

Appendix B

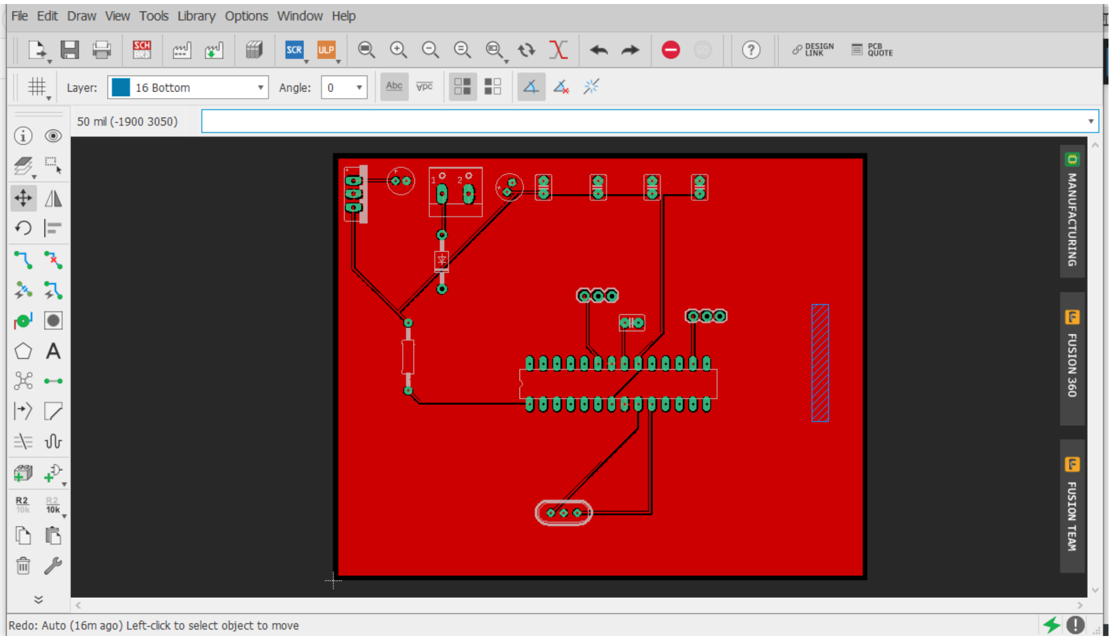

Figure 1: PCB design for transmitter circuit

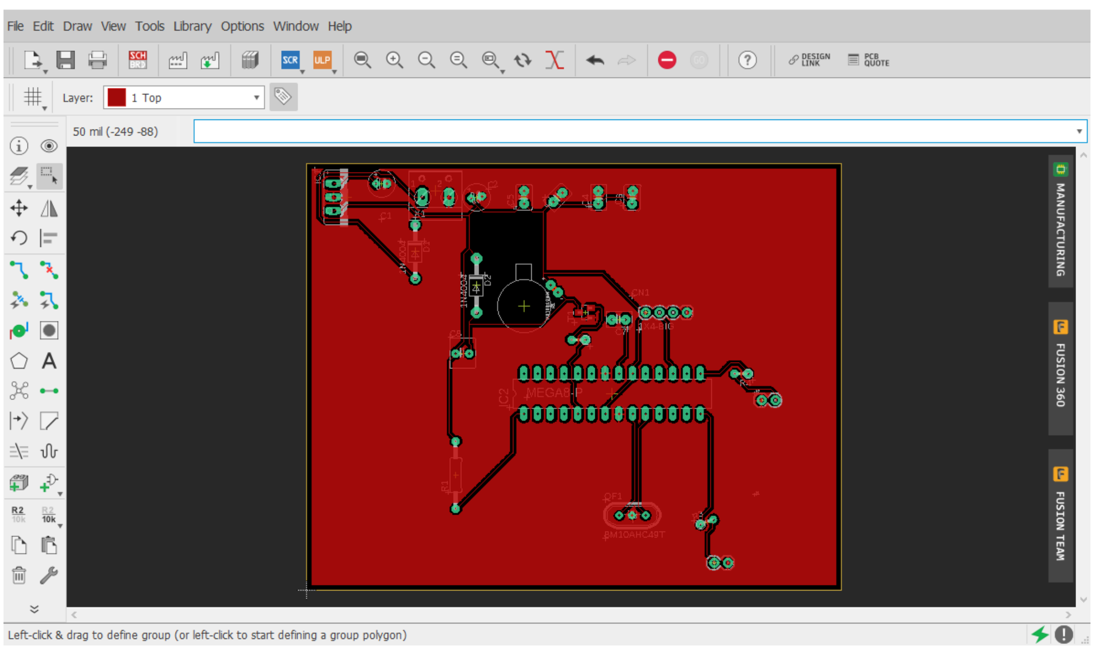

Figure 2: PCB design for receiver circuit

Appendix C
Transmitter Code

```
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#include <RH_ASK.h>
#include <SPI.h> // Not actually used but needed to compile

RH_ASK driver;

//int reading =0;
int UpperThreshold = 594; //518
int LowerThreshold = 511; //490
int reading = 0;
bool IgnoreReading = false;
bool FirstPulseDetected = false;
const unsigned long delayTime = 10;
const unsigned long delayTime2 = 1000;
const unsigned long baudRate = 9600;
unsigned long previousMillis = 0;
unsigned long previousMillis2 = 0;

void setup(){
  Serial.begin(baudRate);
  Serial.begin(9600);  // Debugging only
   if (!driver.init())
   Serial.println("init failed");
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, LOW);
  while (millis()<5000){
   reading = analogRead(0);

   //if(reading>UpperThreshold){
  //  UpperThreshold=reading;
//     }
//    if(reading<LowerThreshold){
//   LowerThreshold=reading;
//      }
    }
}

void loop(){
unsigned long FirstPulseTime = 0;
unsigned long SecondPulseTime = 0;
unsigned long PulseInterval = 0;
float BPM = 0.0;
```

```
// Get current time
unsigned long currentMillis = millis();

// First event
if(myTimer1(delayTime, currentMillis) == 1){
reading = analogRead(0);
reading=map(reading, LowerThreshold, UpperThreshold, 0, 160);
reading=constrain(reading,0,160);
  //reading = analogRead(0);

  // Heart beat leading edge detected.
  if(reading > UpperThreshold && IgnoreReading == false){
    if(FirstPulseDetected == false){

      FirstPulseTime = millis();
      FirstPulseDetected = true;
    }
    else{
      SecondPulseTime = millis();
      PulseInterval = SecondPulseTime - FirstPulseTime;
      FirstPulseTime = SecondPulseTime;
    }
    IgnoreReading = true;
    digitalWrite(LED_BUILTIN, HIGH);
  }

  // Heart beat trailing edge detected.
  if(reading < LowerThreshold && IgnoreReading == true){
    IgnoreReading = false;
    digitalWrite(LED_BUILTIN, LOW);
  }

  // Calculate Beats Per Minute.
  BPM = 10*((1.0/PulseInterval) * 60.0 * 1000);
}

// Second event
if(myTimer2(delayTime2, currentMillis) == 1){
  Serial.print(reading);
  Serial.print("\t");
  Serial.print(PulseInterval);
  Serial.print("\t");
  Serial.print(BPM);
  Serial.println(" BPM");
```

```
    Serial.flush();
  }
  if ((BPM>0)&&(PulseInterval>0)){
    const char *msg = "Hello World!";
    driver.send((uint8_t *)msg, strlen(msg));
    driver.waitPacketSent();
    delay(1000);
         }
}

// First event timer
int myTimer1(long delayTime, long currentMillis){
  if(currentMillis - previousMillis >= delayTime){previousMillis = currentMillis;return 1;}
  else{return 0;}
}

// Second event timer
int myTimer2(long delayTime2, long currentMillis){
  if(currentMillis - previousMillis2 >= delayTime2){previousMillis2 = currentMillis;return 1;}
  else{return 0;}
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

Receiver Code

```
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#include <RH_ASK.h>
#include <SPI.h> // Not actualy used but needed to compile

RH_ASK driver;
void setup()
{
    Serial.begin(9600);  // Debugging only
    if (!driver.init())
         Serial.println("init failed");
}

void loop()
{
    uint8_t buf[12];
    uint8_t buflen = sizeof(buf);
    if (driver.recv(buf, &buflen)) // Non-blocking
    {
      int i;
```

```
      // Message with a good checksum received, dump it.
      Serial.print("Message: ");
      Serial.println((char*)buf);
      digitalWrite(8,HIGH);
      digitalWrite(9,HIGH);
      delay(1000);
       digitalWrite(8,LOW);
      digitalWrite(9,LOW);
      delay(1000);
    }
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```