Infant Health Monitor System

An IoT Health Monitoring System tailored for infants

PROJECT REPORT

for ECE 3501 IoT Fundamentals

by

Akshunn Trivedi
19BEC0038

Shubham Sherki
19BEC0044

Nikhil M Jain
19BEC0046

Vanessa CS Pudhucheri
19BEC0051

School of Electronics
Vellore Institute of Technology
Vellore

November 2021

Declaration by Authors

This is to declare that this report has been written by us as part of our coursework. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. We aver that if any part of the report is found to be plagiarized, we shall take full responsibility for it.

Nikhil M Jain
19BEC0046

Shubham Sherki
19BEC0044

Vanessa CS Pudhucheri
19BEC0051

Akshunn Trivedi
19BEC0038

Place: Vellore

Date: 29th November'21

TABLE OF CONTENTS

ABSTRACT

The baby monitor will primarily be used by parents to keep track of their babies from the office or other places. It also provides the nannies or other caretakers with a medium to keep a close eye on the baby. The app that is connected to the baby monitor will be installed on the phone of the parent and/or caretaker. The app will display the basic information and data, as well as receive timely alerts, which will help in preventing accidents, such as SIDS (Sudden Infant Death Syndrome).

INTRODUCTION

Infant Healthcare monitoring is an essential application of wearable technology. Over the years, a number of smart health monitoring systems have been developed. Smart infant health monitoring systems based on wearable technology can replace the traditional ways of health monitoring and free healthcare practitioners from the tedious tasks of self-monitoring patients from time to time.

Continuous health status monitoring of infants is achieved with the development and fusion of wearable sensing technologies, wireless communication techniques and a low energy-consumption microprocessor,thus enabling us to design a product with minimal energy needs.

These devices can help in providing critical medical input to parents and doctors alike, enabling quicker and efficient methods to save lives of infants who are in constant danger of syndromes like SIDS[Sudden Infant Death Syndrome]

This project is our attempt at solving this problem by designing a product from the ground up, and trying to achieve industry level performance.

Literature Survey

S. no	Title, Author, Name of the Journal,Year of publication	Methodology	Pros	Cons
1	IoT-BBMS: Internet of Things-Based Baby Monitoring System for Smart Cradle , WAHEB A. JABBAR , HIEW KUET SHANG, SAIDATUL N. I. S.	The system architecture consists of a baby cradle that will automatically swing using a motor when the baby cries. Parents can also monitor their	It can continuously monitor the infant	It uses Node MCU - it has 1 analog input pin while we require 4.

	HAMID, IEEE ACCESS , 12 July 2019	babies' condition through an external web camera and switch on the lullaby toy located on the baby cradle remotely via the MQTT server to entertain the baby		
2	IOT based Infant Health Monitoring System Shilpa Mandke, Komal Kudave, Rakshanda Labde ,Principal Dr.J. W. Bakal International Research Journal of Engineering and Technology (IRJET) Apr-2018	IOT infant health monitoring has 4 sensors. First one is a temperature sensor, second is Heartbeat sensor and the third one is motion sensor and voice sensor (detects the noise).The microcontroller connects to the Wi-Fi network using a Wi-Fi module. It sends this data to the cloud by sending this data to a particular URL/IP address. Then this action of sending data to IP is repeated after a particular interval of time. On online portal data is logged with a database and those values fetched to show status of health. PHP will be used as the processing language along with HTML to provide Graphical user interface.	Uses various different sensors like PIR,DHT11, Heartbeat sensor Wifi Connection	Doesn't support bluetooth technology Can't detect roll over , no condition for SIDS External Wifi module required Atmega328 is slow compared to esp32
3	Arduino Based Infant Monitoring System , Daing Noor Farhanah Mohamad Ishak , Muhammad Mahadi Abdul Jamil and Radzi Ambar , 2017	A monitoring system which utilizes humidity sensor and pulse rate sensor is developed to measure the humidity in incubator, and baby's heartbeat respectively. The data collected from the sensors can be viewed on a personal computer (PC) and an alarm system will also alert parents if the data readings reach to certain level that can harm babies.\-	It will also allow us to closely monitor infant conditions through local area networks or the internet.	It is quite bulky in size, since the arduino doesn't have a wifi module integrated in it and thus requires an external Wifi module.

4	Zhu Z, Liu T, Li G, Li T, Inoue Y. Wearable sensor systems for infants. Sensors (Basel). 2015;15(2):3721-3749. Published 2015 Feb 5. doi:10.3390/s150203721	Firstly, the paper reviews some available wearable sensor systems for infants; secondly, introduces the different modules of the framework in the sensor systems; and lastly, the methods and techniques applied in the wearable sensor systems. The latest research and achievements have been highlighted in this paper and the meaningful applications in healthcare and behavior analysis are also presented.	The paper is an exhaustive source of information for the novel wearable tech which can be used to sense the body vitals of infants.	The paper does not have any cons per se,however the focus of the paper has been on the sensor and data analysis part of the project,and less info is provided for the microcontroller design.
5	Wearable technology for infant health monitoring: Saba Feroz Memon , Mohsin Memon, Sania Bhatti IET Circuits, Devices & Systems	The paper discusses the recent advancement in wearable technology and development of wearable infant health monitoring systems (WIHMSs). It discusses systems providing an edge over conventional infant health monitoring systems which are not only bulky and uncomfortable for the infants but are also limited to the clinical settings. It reviews some state-of-the-art WIHMS,mentions the advantages, limitations and the challenges faced by such systems.	The paper is a very good resource of some of the latest WIHMS.	The paper is informative,but still general in nature, and doesn't cover all systems. It just serves as a reference guide.

Overview

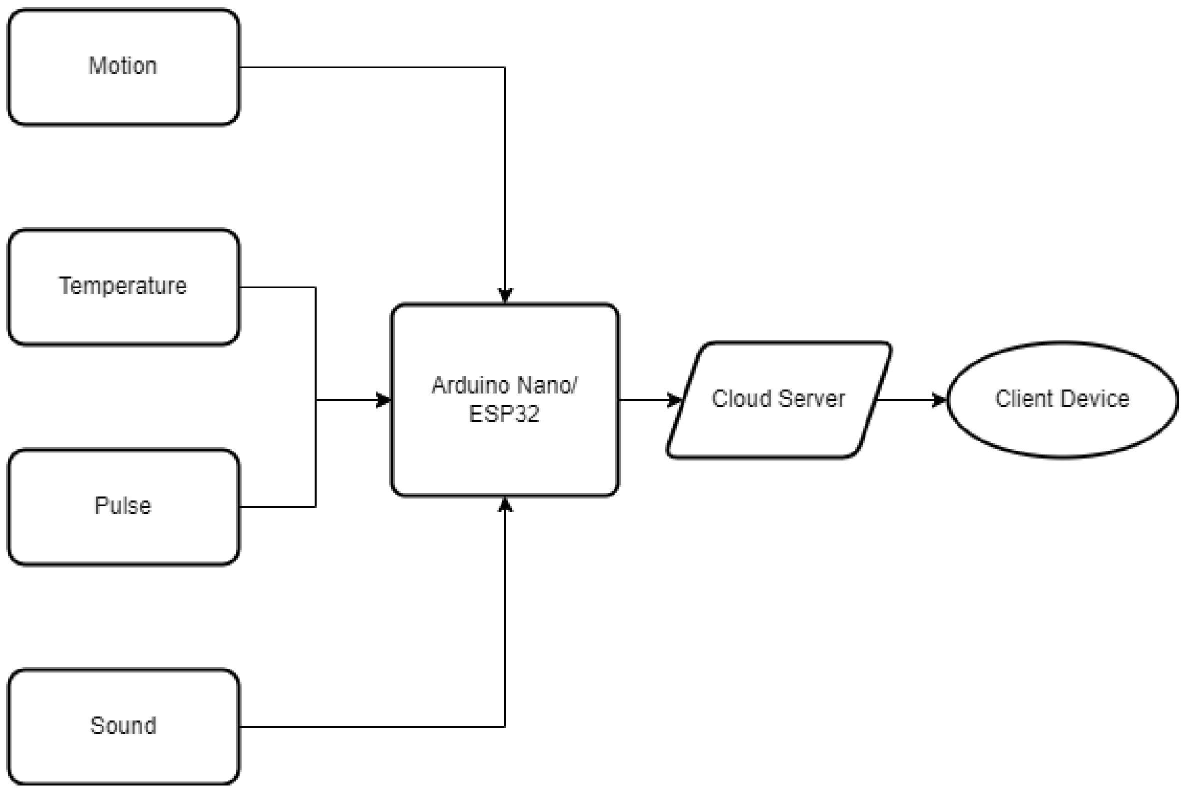
The project had two main components:

1) Acquisition of data concerning the health parameters of the baby and alert based on the same.
2) Subsequent Analysis based on these parameters and SIDS prediction.

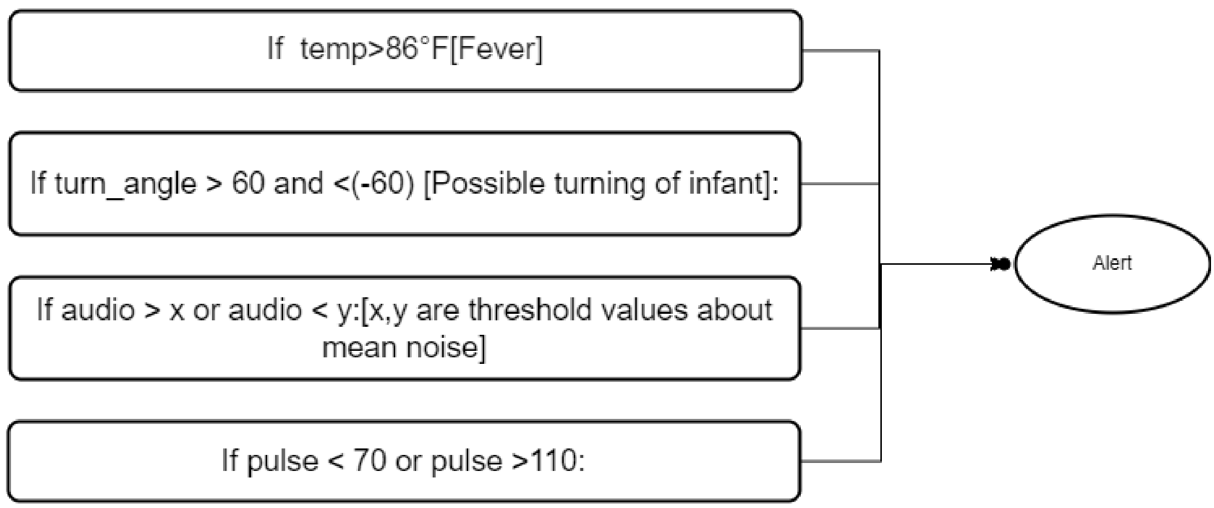
We were successfully able to measure health parameters viz a viz BP,Noise level,Temperature and Angle rotation and sent them to the cloud using Cayenne. These parameters were then analyzed and triggers set. Upon the passing of these parameters below or above the threshold values, an alert was sent to the concerned node [for example Parent or Doctor] in real time.

Subsequently a python script was also developed for the analysis of the raw data secured from our health monitoring system and using the average values of the parameters, likability estimated of SIDS in future.

Flow Chart



Algorithm

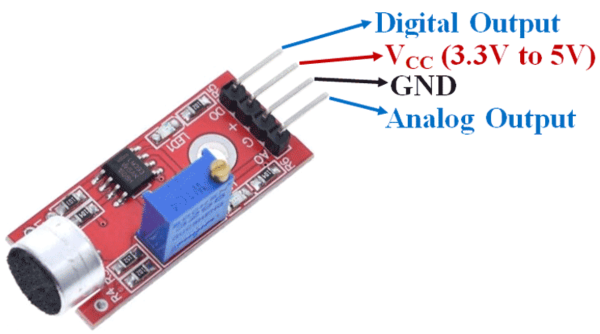


Methodology

Components:

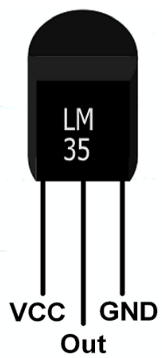
Sensors Used:

1) **LM393 Sound Detection Sensor:** Sound detection sensor module detects the intensity of sound where sound is detected via a microphone and fed into an **LM393 op-amp**. It comprises an onboard potentiometer to adjust the setpoint for sound level.



LM393 Sound Sensor PINOUT

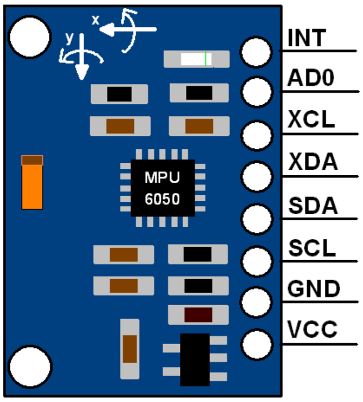
2) **LM35 Temperature Sensor:** A temperature measuring module which gives analog output proportional to temperature and it doesn't require any external calibration circuitry with a sensitivity of 10mV/degree Celius.It is a 3-terminal sensor used to measure surrounding temperature ranging from -55 °C to 150 °C. LM35 gives temperature output which is more precise than thermistor output.



LM35 PINOUT

3) **MPU6050 Gyroscope Module:** It is a 6 axis motion tracking device,3-axis Gyroscope 3-axis Accelerometer and digital motion processor embedded with an additional feature of on chip temperature sensor , which communicates with

microcontrollers through I2C protocol. It also has an auxiliary I2C bus to communicate with 3-axis Magnetometer, Pressure sensor etc. If a 3-axis Magnetometer is connected to the auxiliary I2C bus, then MPU6050 can provide complete 9-axis Motion Fusion output.



MPU6050 Gyroscope sensor

4) **Pulse Sensor:** It is a plug & play sensor mainly designed for Arduino boards which can be used by makers, students, developers, artists who can utilize the heartbeat information into their projects. This sensor uses an easy optical pulse sensor along with amplification & cancellation of noise to make a circuit. By using this circuit, we can get fast and reliable heartbeat readings. This circuit can be operated with 4mA current and 5V voltage to use in mobile applications.

Pulse Sensor PINOUT

IoT Platform

Cayenne: Cayenne is a drag-and-drop IoT platform, developed by myDevices, that empowers users to quickly prototype and share their connected IoT solutions.

Working:

- **Prerequisites :**

  User account on Cayenne

- **SetUp:**

  Log in to the Cayenne platform

  Choose your board to start a project

  Set up your Arduino

  Add Cayenne MQTT Library to your Arduino IDE

  The Cayenne MQTT Library is available directly from the Arduino IDE Libraries list. To install the library, select **Sketch** > **Include library** > **Manage Libraries**.

- **Configure Arduino IDE**

  In order to successfully program your Arduino board, you will need to verify that the appropriate Board and Port are selected in the Arduino IDE.

  First, verify that the correct Board is selected in the **Tools** > **Board** menu. Be sure to select the board type that you will be programming.

  Then, verify that you have the correct **Port** selected for communicating with your Arduino. Pick the correct port based upon how you are connecting your Arduino to your PC/Mac.

- **Step 3: Connect your Arduino**

  After setting up your PC/Mac computer with the Arduino IDE and the Cayenne Library, you are ready to install Cayenne onto your device. Continue to Step 3 of the Arduino installation, where you will connect your board with Cayenne.

  On this screen, when selecting your Arduino board, a list of connections appears below the board name. Select the Arduino Uno, and select Ethernet shield W5100.

  Click on the Sketch button next to "Ethernet Shield W5100", you will get a pop-up window which has Arduino Sketch code as in the following picture.

Copy & paste the sketch code into Arduino IDE and select **Sketch** > **Upload** to compile and upload the sketch into Arduino UNO board.As soon as your Arduino device comes online and connects to Cayenne, you will see your Arduino Board in the online dashboard.

Code With Comments

Data Acquisition and Communication(Using Arduino Nano and Cayenne):

```
#define USE_ARDUINO_INTERRUPTS true
#include <CayenneMQTTSerial.h>
#include "DHT.h"
#define DHTPIN 4
#define DHTTYPE DHT11
#include "Wire.h"
#include <MPU6050_light.h>

MPU6050 mpu(Wire);
unsigned long timer ;
int sound;
int soundPin = A0;
DHT dht(DHTPIN, DHTTYPE);
char username[] = "<Enter Your MQTT Username>";
char password[] = "<Enter Your MQTT Password>";
char clientID[] = "<Enter Your Client ID>";
   // Set-up low-level interrupts for most accurate
 BPM math.
#include <PulseSensorPlayground.h>     // Includes the PulseSensorPlayground Library.

//  Variables
const int PulseWire = A3;       // PulseSensor PURPLE WIRE connected to ANALOG PIN 0
const int LED13 = 13;          // The on-board Arduino LED, close to PIN 13.
int Threshold = 550;
int myBPM;
float h,X;
float t,room_sound;
int d;
bool humidity = false;
bool Temperature = false;
PulseSensorPlayground pulseSensor;
void setup() {
 Cayenne.begin(username, password, clientID);
 dht.begin();
  humidity = false;
  Temperature = false;
  Wire.begin();
  byte status = mpu.begin();
   pulseSensor.analogInput(PulseWire);
  pulseSensor.blinkOnPulse(LED13);       //auto-magically blink Arduino LED with heartbeat.
  pulseSensor.setThreshold(Threshold);
  while(status!=0){ }
```

```
  mpu.calcOffsets();
  calculate_roomsound();
  if (pulseSensor.begin()) {
    digitalWrite(13,HIGH); //This prints one time at Arduino power-up,  or on Arduino reset.
  }
  delay(1000);
}
void loop() {
  mpu.update();
  Cayenne.loop();

}
CAYENNE_OUT(V0){

  //Check if read failed and try until success
  do {
    //Read humidity (percent)
   sound=analogRead(soundPin);;
  } while  (isnan(h));

  //Set Humidity to true so we know when to sleep
  humidity = true;

  Cayenne.virtualWrite(V0, sound-room_sound);
}

CAYENNE_OUT(V1){

  do {
    t = dht.readTemperature();
  } while  (isnan(t));
  Cayenne.virtualWrite(V1, t,"temp","c");
}
CAYENNE_OUT(V2){

  do {
    X =mpu.getAngleX();
    timer=millis();
  } while  (millis()-timer>0);
  Cayenne.virtualWrite(V2, X);
}
CAYENNE_OUT(V3){
   int myBPM = pulseSensor.getBeatsPerMinute();  // Calls function on our pulseSensor object that returns BPM
as an "int"
  Cayenne.virtualWrite(V3,myBPM);
}
void calculate_roomsound(){
  while(d<200){
    room_sound=room_sound+analogRead(soundPin);
    d++;
  }
  room_sound=room_sound/200;
```

```
  d=0;
}
```

Data Analysis(Using Python)

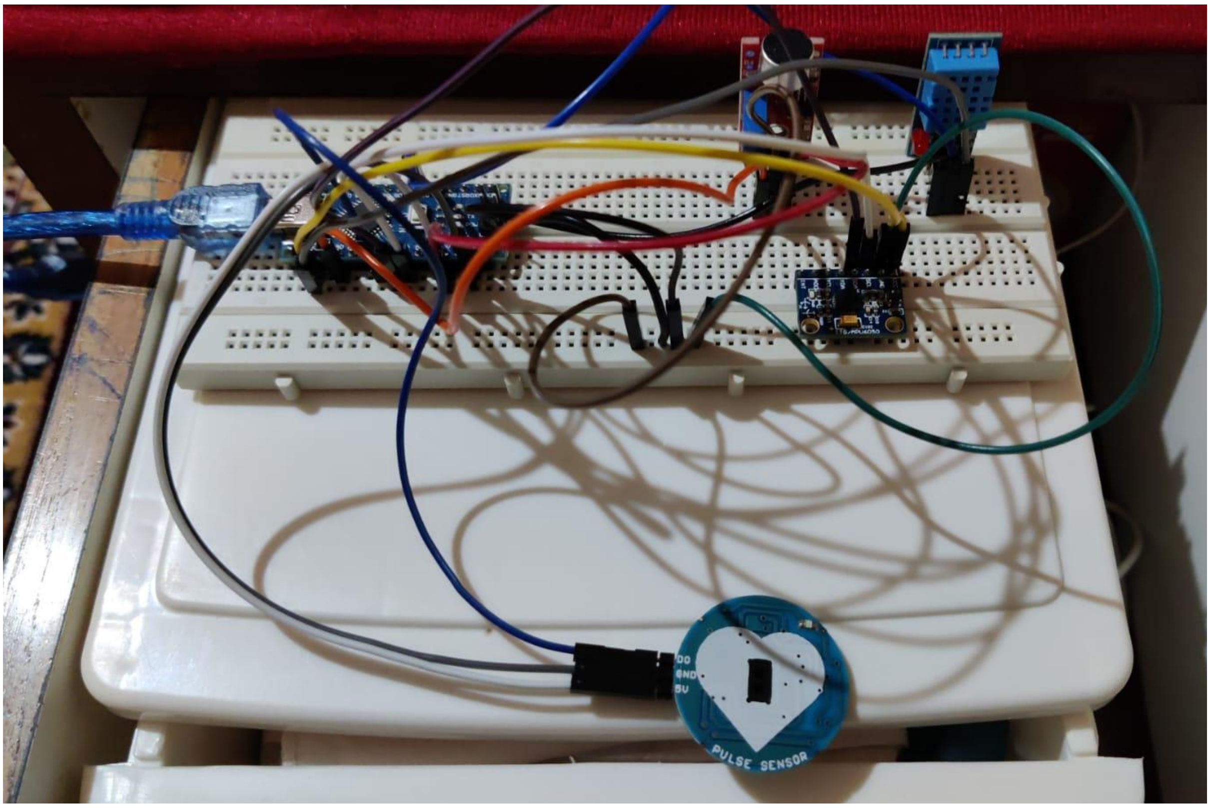
```
import pandas as pd

df=pd.read_csv("/content/IoT Data.csv")

df.head

df=df[['Sensor Name','Value']]

#print(df)

#print(len(df))

#print(df['Sensor Name'][0])

#df["Sound"]=""

#print(df)

for i in range(len(df)):

  if df['Sensor Name'][i]=="bpm" and df['Value'][i]==float("NaN"):

    df['Value'][i]==75

df[["Angle","BPM","Sound","Temp"]]=None

#print(df)

for i in range(len(df)):

  if df['Sensor Name'][i]=="Channel 2":

    df['Angle'][i]=df['Value'][i]

  elif df['Sensor Name'][i]=="bpm":

    df['BPM'][i]=df['Value'][i]

  elif df['Sensor Name'][i]=="Sound":

    df['Sound'][i]=df['Value'][i]

  elif df['Sensor Name'][i]=="Channel 1":

    df['Temp'][i]=df['Value'][i]

#print(df)

df=df.drop(['Sensor Name','Value'],axis=1)

#print(df)

angle_len=0
```

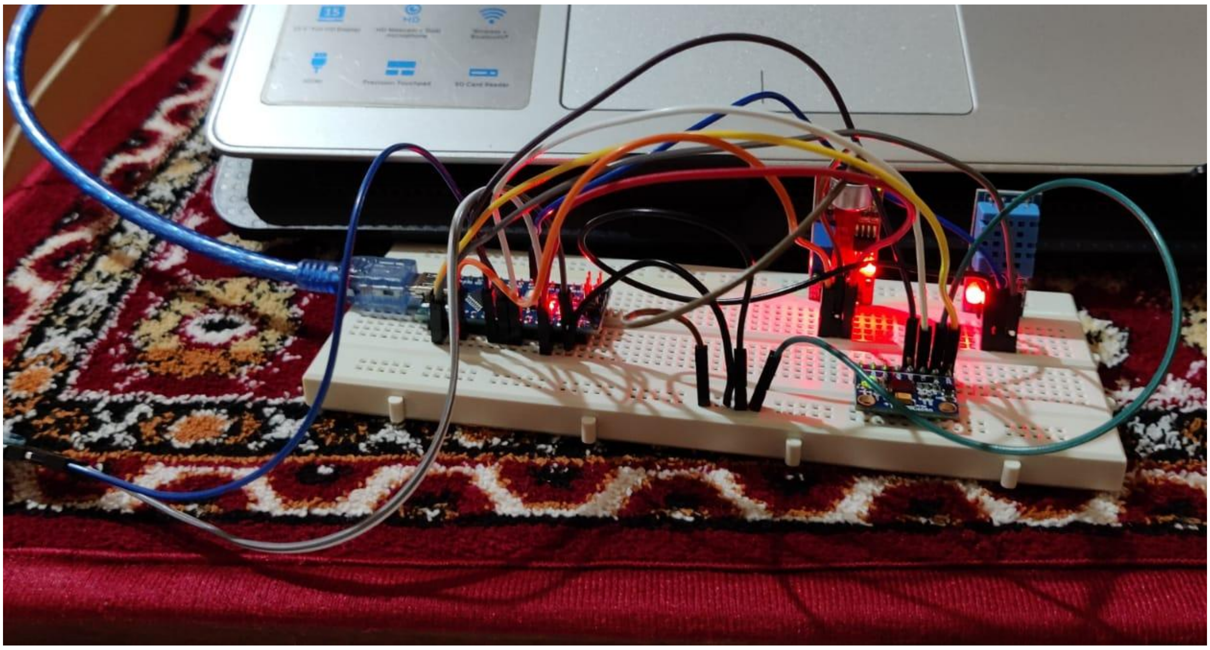
```
BPM_len=0#

Sound_len=0

temp_len=0

for i in range(len(df)):

  if df['Angle'][i]!=None:

    angle_len+=1

  if df['BPM'][i]!=None:

    BPM_len+=1

  if df['Sound'][i]!=None:

    Sound_len+=1

  if df['Temp'][i]!=None:

    temp_len+=1

#print(df['Temp'][0])

avg_angle=df['Angle'].sum()/(angle_len)

avg_BPM=df['BPM'].sum()/(BPM_len)

avg_sound=df['Sound'].sum()/(Sound_len)

avg_temp=df['Temp'].sum()/(temp_len)

angle_min=df['Angle'].min()

angle_max=df['Angle'].max()

angle_range= angle_max-angle_min

sound_max=df['Sound'].max()

pulse_min=df['BPM'].min()

pulse_max=df['BPM'].max()

sound_min=df['Sound'].min()

temp_min=df['Temp'].min()

temp_max=df['Temp'].max()

Body_array=[["Angle","BPM","Sound","Temp"],[avg_angle,avg_BPM,avg_sound,avg_temp],[angle_min,pulse_min,sound_min,temp_min],[angle_max,pulse_max,sound_max,temp_max]]

for i in range(4):

  print("\n")
```

```
  for j in range(4):
    print(Body_array[i][j])


pointer=0
if(angle_range>100):
  pointer+=1
if(avg_temp>30):
  pointer+=1
if(pulse_min<60):
  pointer+=1
print(pointer)
if pointer==0:
  print("No evident danger of SIDS")
if pointer==1 or 2:
  print("Evident danger of SIDS")
if pointer==3:
  print("High danger of SIDS")
```

Result

Finished Circuit:





Hardware Explanation Video:

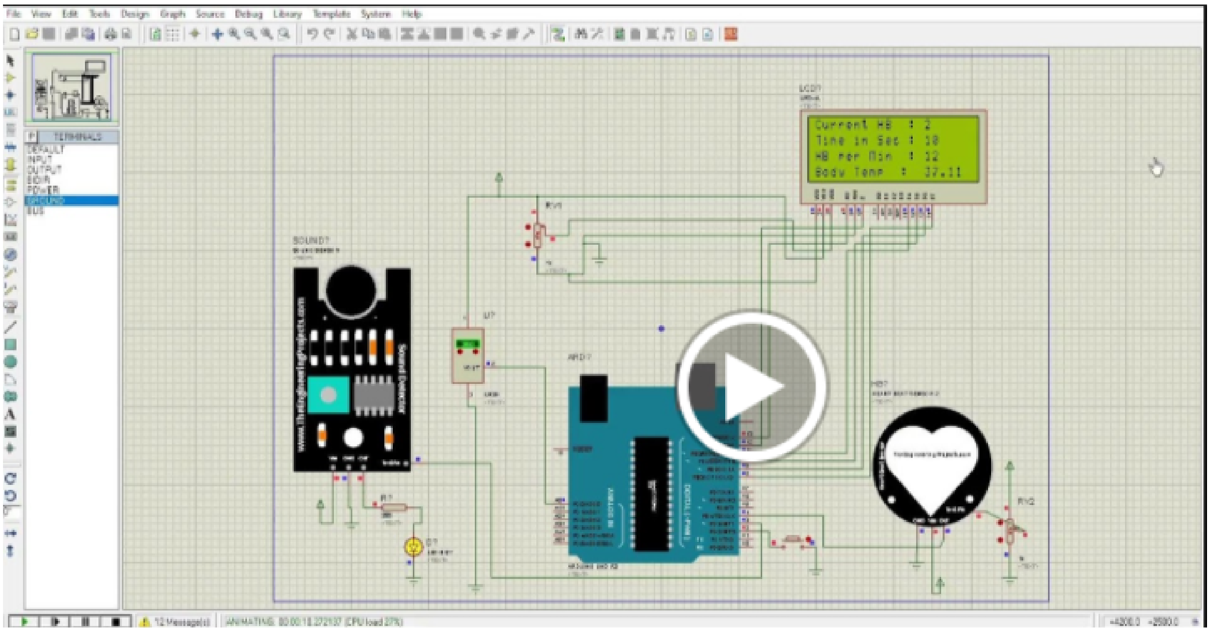
https://drive.google.com/file/d/1d7_2YXIya2mT-oiYssbSNvkjhqIW7SD6/view?usp=sharing

Cayenne Data Exploration and Explanation Video:

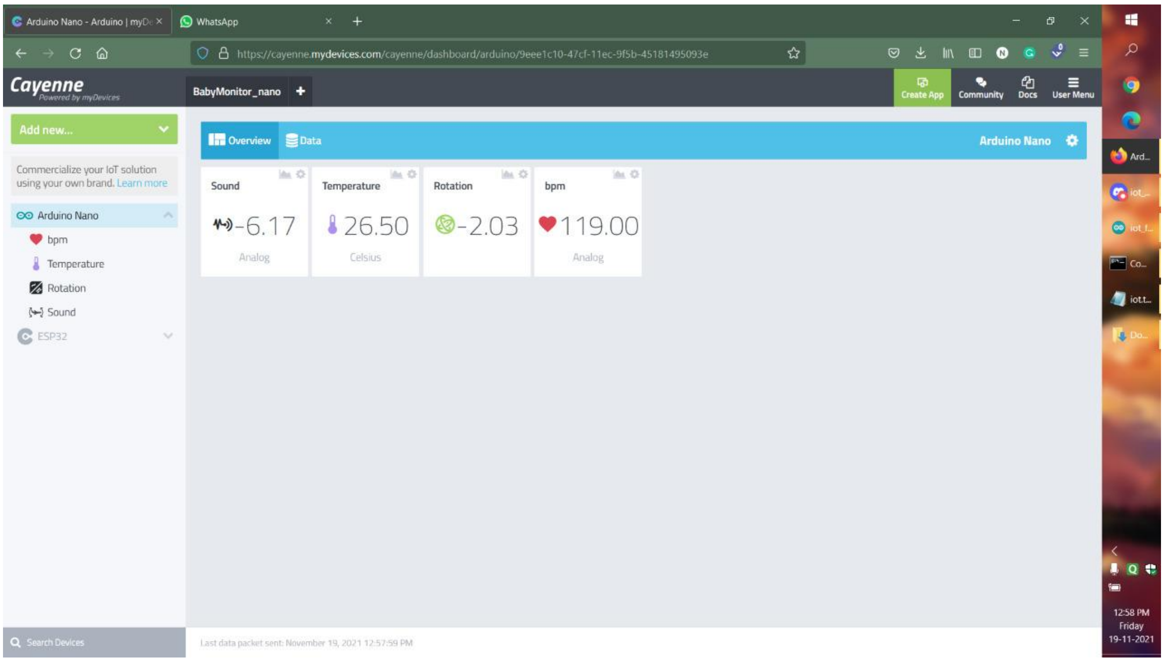
https://drive.google.com/file/d/1MYkfLlkAKfjqyJ6TStMWKG4fWVx6pIdL/view?usp=sharing

Software Simulation Video:

Circuit Simulation Video.mp4 - Google Drive



Cayenne Dashboard:



The dashboard gives us the health monitoring parameters like Sound level, Temperature,Rotation Angle,Heart Rate(BPM)

Data Analysis:

```
Angle
BPM
Sound
Temp

-5.68752639767763O5
101.91071428571429
50.7594674891184
27.065853503292 69

-108.013000499999999
43.0
-62.34500122
26.29999924

160.9850006
231.0
149.0
28.29999924
```

SIDS Prediction Using Raw Data

```
[127] pointer=0
      if(angle_range>100):
        pointer+=1
      if(avg_temp>30):
        pointer+=1
      if(pulse_min<60):
        pointer+=1
      print(pointer)
      if pointer==0:
        print("No evident danger of SIDS")
      if pointer==1 or 2:
        print("Evident danger of SIDS")
      if pointer==3:
        print("High danger of SIDS")

      2
      Evident danger of SIDS
```

According to the average parameters from the raw data, pointers for the SIDS are calculated

Future Work

1. Adding a body warmer - we can add a body warmer in order to regulate temperature around the baby.
2. App development
3. Data encryption
4. Secure connection
5. Better sensor and using once which are for material purposes

Conclusion and recommendations

- We designed a baby monitor using Arduino Nano and respective sensors to measure the noise, rotation angle, heartbeat, and temperature.
- These parameters are used to alert the responsible body for a possible health issue and particularly the chance of occurrence of SIDS. A SIDS prediction program was also developed using the user data
- A software model was also developed using Proteus to simulate the functioning of our system.

- As a closing note, the system is still far from achieving state-of-the-art accuracy and precision but should be improved upon the use of higher-precision sensors and a better connection scheme.

REFERENCES

- https://pimylifeup.com/arduino-cayenne/
- https://lastminuteengineers.com/esp32-dht11-dht22-web-server-tutorial/#:~:text=If%20you%20 didn%27t%20know,analog%2Dto%2d Digital%20converter.
- https://community.mydevices.com/t/double-temperature-rain-humidity-on-lcd-with-esp8266/3520?_ga=2.15416374.1257147753.1634039015-373789862.1634039015
- https://iotdesignpro.com/projects/iot-controlled-led-using-cayenne-and-esp32
- https://www.electronicwings.com/sensors-modules/mpu6050-gyroscope-accelerometer-temperature-sensor-module
- https://www.electronicings.com/components/lm35-temperature-sensor#:~:text=LM35%20is%20a%20temperature%20 measuring,increases%2C%20 output%20 voltage%20also%20 increases