Biosensor Bracelet

Microphone Sound Sensor

Eshan Salwan

N01422232

Hardware Production Technology - CENG-317-0NA

August 18, 2023

Abstract:

In today's healthcare landscape, non-invasive monitoring of vital signs remains critical for the well-being of individuals. This project introduces a novel approach to monitoring sound-related data using easily accessible components, including the LM393 microphone sound sensor, an Raspberry Pi, passive components like resistors and transistor, a custom-designed circuit board, and LED indicators. The proposed system offers a convenient and efficient method for capturing sound data without causing disruption or discomfort.

At the heart of the system lies the LM393 microphone sound sensor, a compact and high-performance sensor capable of accurately capturing sound frequency. When coupled with an Raspberry Pi, the sensor's data collection capabilities enable real-time sound monitoring. The inclusion of passive components such as resistors and transistors aids in signal conditioning, amplification, and noise reduction, enhancing the accuracy of sound measurements.

A custom-designed circuit board serves as the foundation for assembling the components into a functional circuit. LED indicators provide a user-friendly interface, assisting users in positioning the sensor optimally for precise sound data collection. When sound is detected, the system records and processes the signals, allowing for the analysis of sound patterns and levels.

The software component of this project involves the development of data processing algorithms tailored to the LM393 microphone sound sensor. Signal processing techniques are employed to captured audio signals, transforming them into usable sound data. This data is then presented through a user-friendly interface, making it accessible to caregivers and healthcare professionals for real-time sound monitoring.

In conclusion, this project introduces an innovative system for sound monitoring, utilizing the LM393 microphone sound sensor, Raspberry Pi capabilities, and well-constructed circuitry. The potential implications for healthcare and various applications are substantial, as the system provides timely and critical sound data that can be invaluable in ensuring the well-being of infants.

Table of Contents:

Introduction..4

Integration with Software APP...5

Project Proposal Revisited...6

Hardware Components..7

Circuit Design and Functionality..8

Functionality of MAX30102...8.1

Functionality of sensor pins..8.2

Functionality of PCB Board...8.3

Schematic design and visual view of sensor..8.4

Usage Process..9

Testing and Validation...10

GitHub Repository...11

Testing Results and Outputs..12

Troubleshooting and Learning...13

Discussion and Insights..14

Conclusion..15

Code Appendix..16

4. Introduction:

In the ever-evolving intersection of technology and healthcare, this project introduces the Sound Health Monitor, a fusion of innovation and sound-related data monitoring. This initiative aims to capture and interpret sound patterns using the LM393 microphone sound sensor, an Raspberry Pi, passive components like resistors and transistor, a custom-designed circuit board, and LED indicators.

The Sound Health Monitor empowers caregivers and healthcare professionals with a non-invasive approach to sound monitoring. The LM393 microphone sound sensor excels at capturing sound data, and when combined with the Raspberry Pi, it facilitates real-time sound analysis. This synergy between hardware and software, with data processing conducted by the Raspberry Pi, exemplifies the transformative impact of technology in enhancing healthcare practices.

5. Integration with Software APP:

Baby Health Monitor integrates with the software created in CENG 322 to fill the gap between hardware and software. The sensor's output is effortlessly transported to the Raspberry Pi device, where the APP will use the Firebase Database to analyze and display the data. Through this, consumers are given accurate and current health measurements, enabling well-informed decision-making and prompt action.

6. Project Proposal Revisited:

Modern hardware and cutting-edge technologies have been included into the original Baby Health Monitor idea to provide a more effective design for monitoring a baby's health. The primary goal is still to create a system to monitor the frequency and sound patterns of newborns. However, the strategy and its parts have been improved for efficient application.

The goals include simple component integration, reliable communication, and a user-friendly interface. Thorough testing validates measurement accuracy and enhances the relationship between hardware and software. The updated plan maintains the original objective while creating a state-of-the-art baby health monitor thanks to technology and ingenuity.

7. Hardware Components:

The core of Baby Health Monitor includes:

LM393 sensor: designed to detect sound levels and convert them into electrical signals that can be processed.

Raspberry Pi: As a hub, it handles data processing, shows instant results, and promotes user involvement.

NPN Transistor: a crucial switch that turns on the LED and guarantees reliable signal transfer.

Resistors: A 2.2 k resistor enhances the precision of the NPN transistor, while a 220 resistor safeguards the LED.

PCB: the foundation for consistent component interaction, optimising connections for efficiency.

LED Light: a visible signal that highlights effective metrics and improves user transparency.

These elements work together to enable the infant health monitor's ability to provide accurate monitoring.

8. Circuit Design and Functionality:

The LM393 sensor, Raspberry Pi, NPN transistor, resistors, circuit board, and LED light are expertly integrated into the infant health monitor's circuit design to form a single device that flawlessly catches and records sound patterns and frequency.

The LM393 microphone sound sensor functions as the primary data source in this system. When activated, it captures ambient sound by utilizing its sensitive microphone component. Sound waves in the environment cause fluctuations in the sensor's input, which are then processed to extract sound-related data.

The signal gathered by the LM393 microphone sound sensor is directed towards the core of computation, the Raspberry Pi. Within this central processing unit, intricate algorithms are employed to interpret the incoming sound input. These algorithms analyze the sound patterns and characteristics, allowing for the extraction and subsequent display of relevant sound-related information on an interface. NPN transistors and resistors orchestrate precisely. A transistor responding to the Raspberry Pi signal controls the LED lighting. Its brightness ensures a successful measurement.

The durability of the LED is increased by a 220 resistor, which guards against high current. The NPN transistor's performance is enhanced by a 2.2 k resistor, resulting in cautious switching. Together, these components work in harmony to form a whole system. The Baby Health Monitor is a user-friendly tool that delivers crucial information, made possible by the LM393 sensor and Raspberry Pi capabilities.

8.1 Functionality of LM393:

The LM393 microphone sound sensor serves as a fundamental component for capturing and analyzing sound in various applications. Its functionality relies on its highly sensitive microphone, which detects sound waves in the surrounding environment. When sound occurs, the microphone converts these acoustic signals into electrical variations. The LM393, often coupled with an Raspberry Pi or similar microcontroller, then processes these electrical signals. It interprets the sound patterns, amplitude, and frequency, making it capable of detecting events like claps, knocks, or changes in ambient noise levels. This functionality is invaluable in applications ranging from sound-activated switches to environmental monitoring systems, where real-time sound data is needed for decision-making or automation processes.

8.2 Functionality of sensor pins:

The sensor pins of the LM393 play a role in data transmission and acquisition.

GND (Ground): This pin provides the reference ground for the sensor and provides a common electrical ground between the sensor and the microcontroller.

VIN (Voltage Input): The VIN pin provides power to the sensor. It receives the necessary supply voltage (typically 3.3-5V) that the sensor needs to function properly.

AO (Analog Output): The AO pin provides an analog representation of the sound detected by the microphone. It outputs a continuous voltage signal that varies in proportion to the intensity of the sound. As the surrounding sound level changes, the voltage output on the AO pin fluctuates accordingly. This analog signal can be connected to an analog-to-digital converter (ADC) on a microcontroller or other hardware to obtain precise sound level measurements. Using the AO pin, you can capture the sound's intensity in real-time, which is useful for applications like sound level meters or audio recording systems.

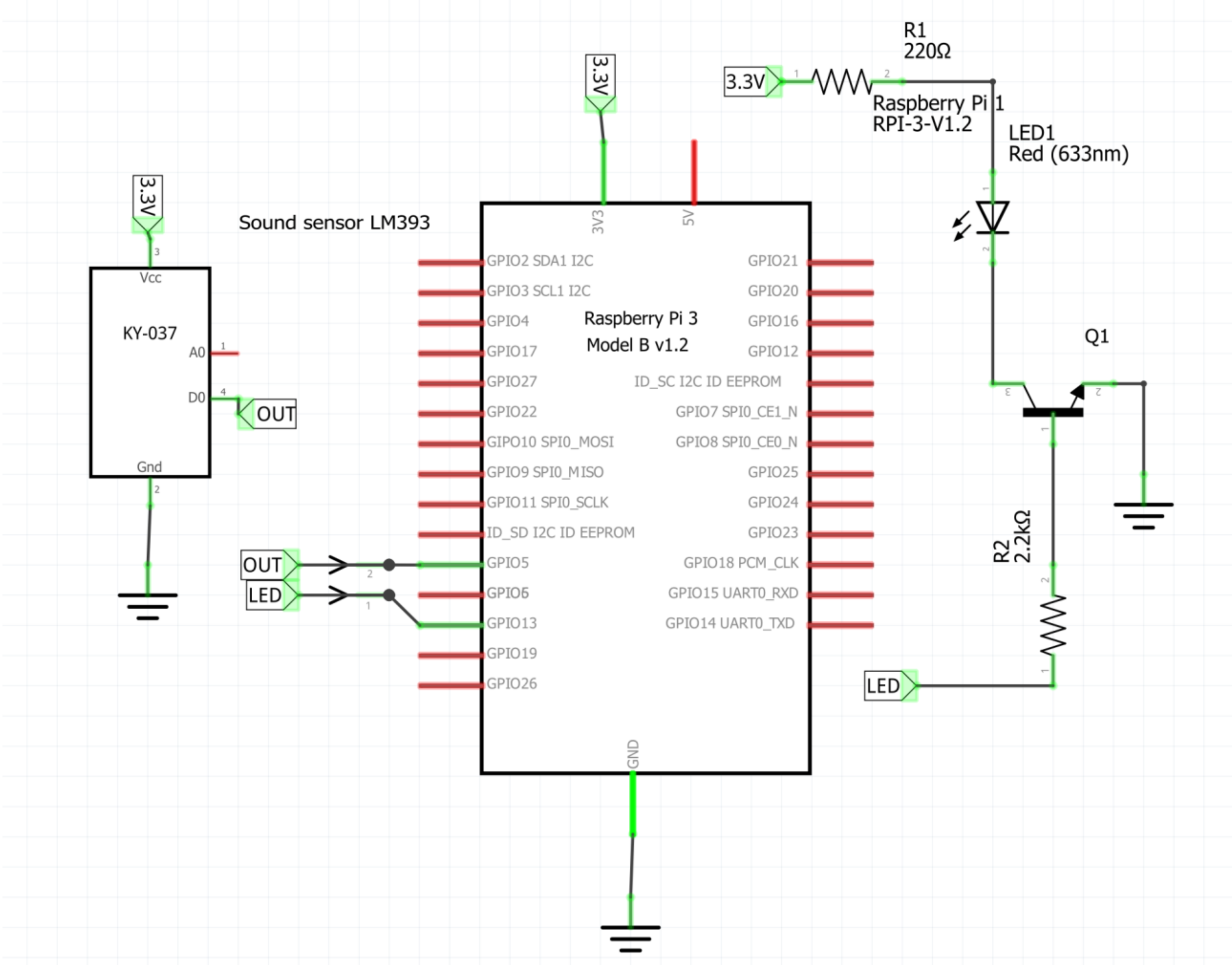
DO (Digital Output): The DO pin provides a digital output that indicates the presence or absence of a sound event that exceeds a predefined threshold level. When the sound level surpasses the set threshold, the DO pin goes HIGH (usually at the supply voltage level), indicating that a sound event has been detected. When the sound level falls below the threshold, the DO pin goes LOW (usually at ground level). The DO pin is often used in sound-activated switch applications or for triggering events when a specific sound intensity or pattern is detected. You can configure the threshold level using an onboard potentiometer or resistor, making it adaptable to different sound level requirements.

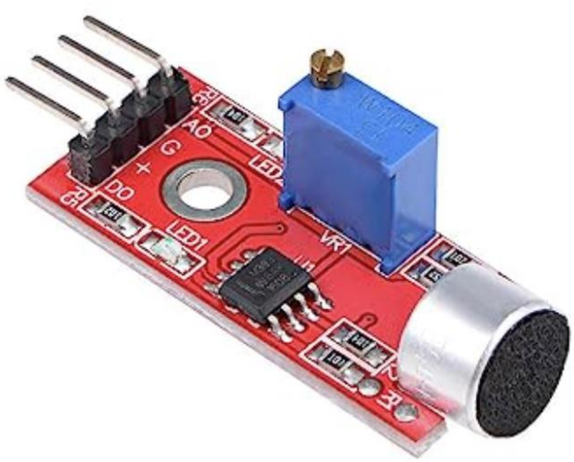
In summary, the LM393 microphone sound sensor's AO pin provides an analog representation of sound intensity, allowing for precise measurement of sound levels. The DO pin, on the other hand, provides a digital output that can be used to detect sound events exceeding a predefined threshold. These dual outputs make the sensor versatile for a wide range of sound-related applications, from basic sound detection to more sophisticated audio analysis tasks.

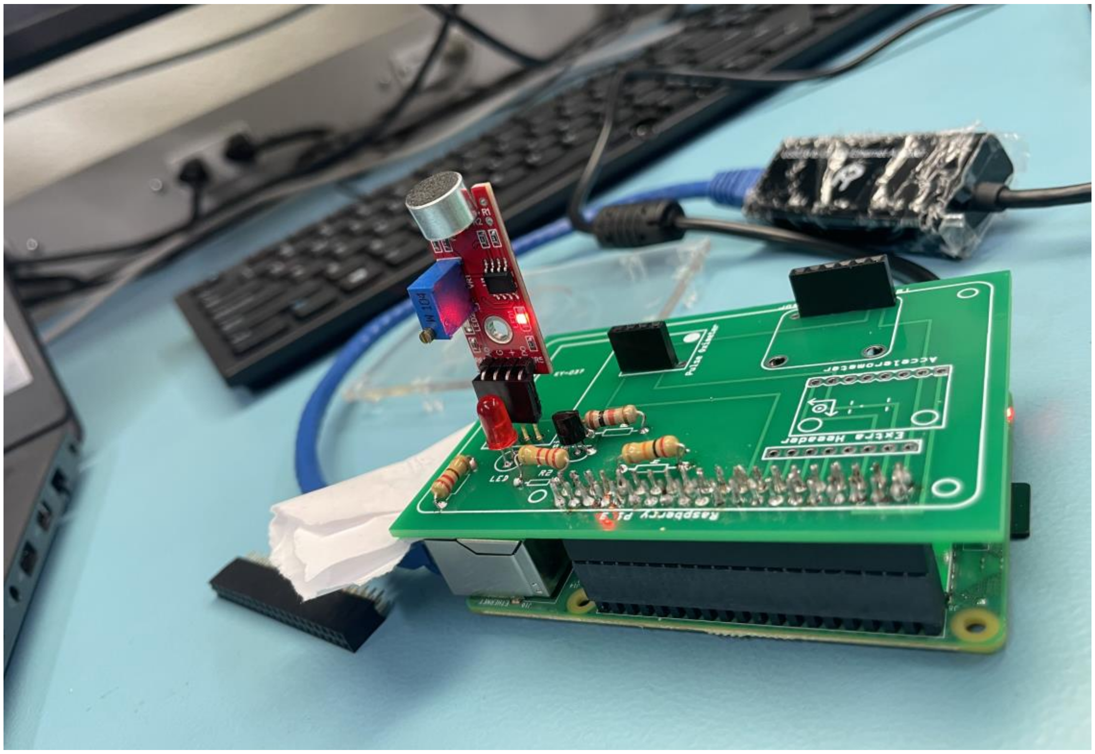
8.3 Functionality of PCB Board:

The PCB serves as a crucial link between the Raspberry Pi and hardware elements like the LM393 sensor. The proper connections between these components are ensured by the printed circuit's well-organized architecture. Circuit traces are used to transfer data from the LM393 sensor to the Raspberry Pi, which has computing capability. Then, using computer algorithms, this data is transformed into comprehensible sound signals. The Baby Health Monitor system's entire operation is improved by the PCB, which guarantees dependable data transfer.

8.4 Schematic design and visual view of sensor:

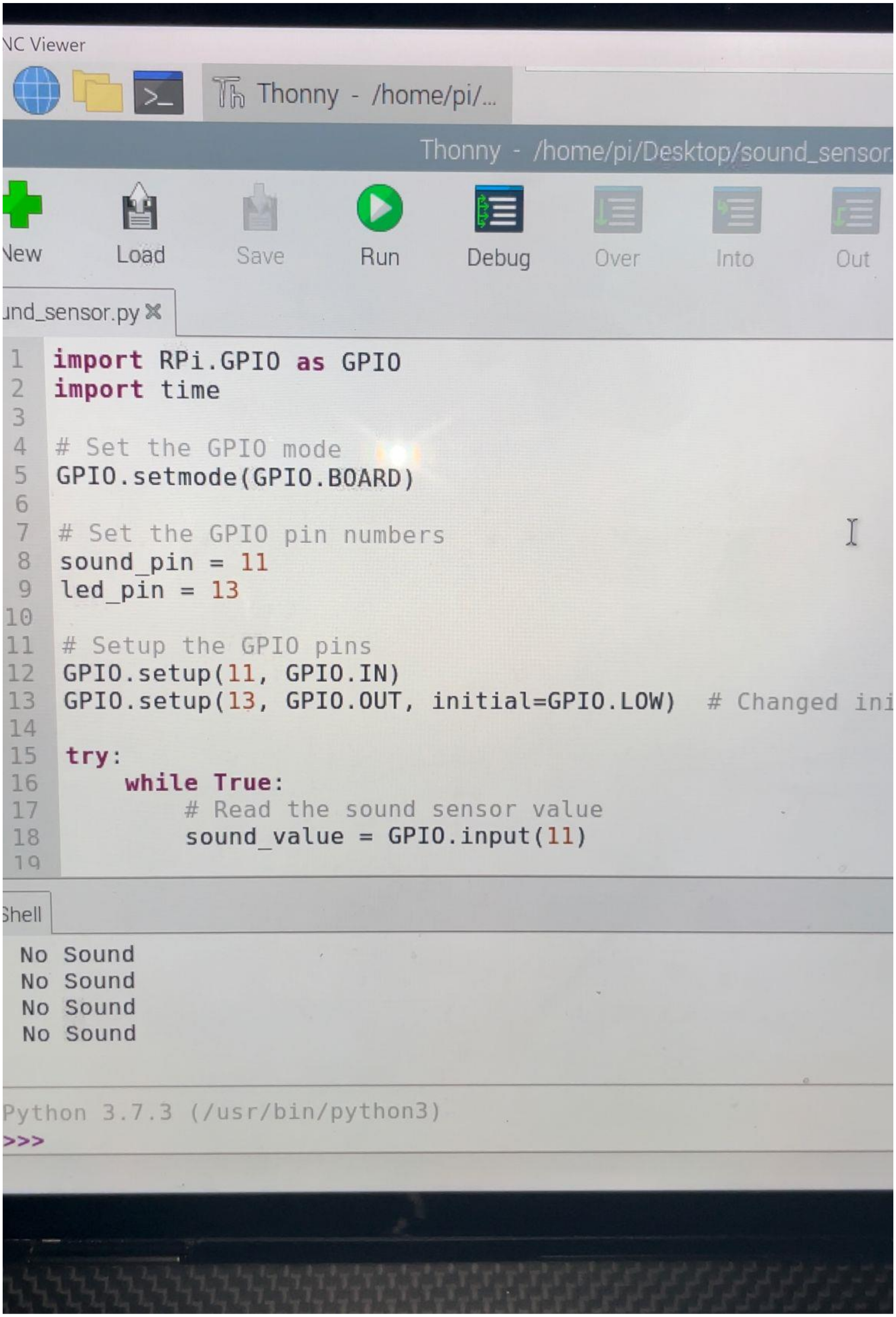

R1
220Ω
3.3V
Raspberry Pi 1
RPI-3-V1.2
LED1
Red (633nm)
3.3V
3.3V
Sound sensor LM393
KY-037
Vcc
A0
D0
Gnd
OUT
3V3
5V
Raspberry Pi 3
Model B v1.2
GPIO2 SDA1 I2C
GPIO3 SCL1 I2C
GPIO4
GPIO17
GPIO27
GPIO22
GIPO10 SPI0_MOSI
GPIO9 SPI0_MISO
GPIO11 SPI0_SCLK
ID_SD I2C ID EEPROM
GPIO5
GPIO6
GPIO13
GPIO19
GPIO26
GPIO21
GPIO20
GPIO16
GPIO12
ID_SC I2C ID EEPROM
GPIO7 SPI0_CE1_N
GPIO8 SPI0_CE0_N
GPIO25
GPIO24
GPIO23
GPIO18 PCM_CLK
GPIO15 UART0_RXD
GPIO14 UART0_TXD
GND
OUT
LED
Q1
R2
2.2kΩ
LED

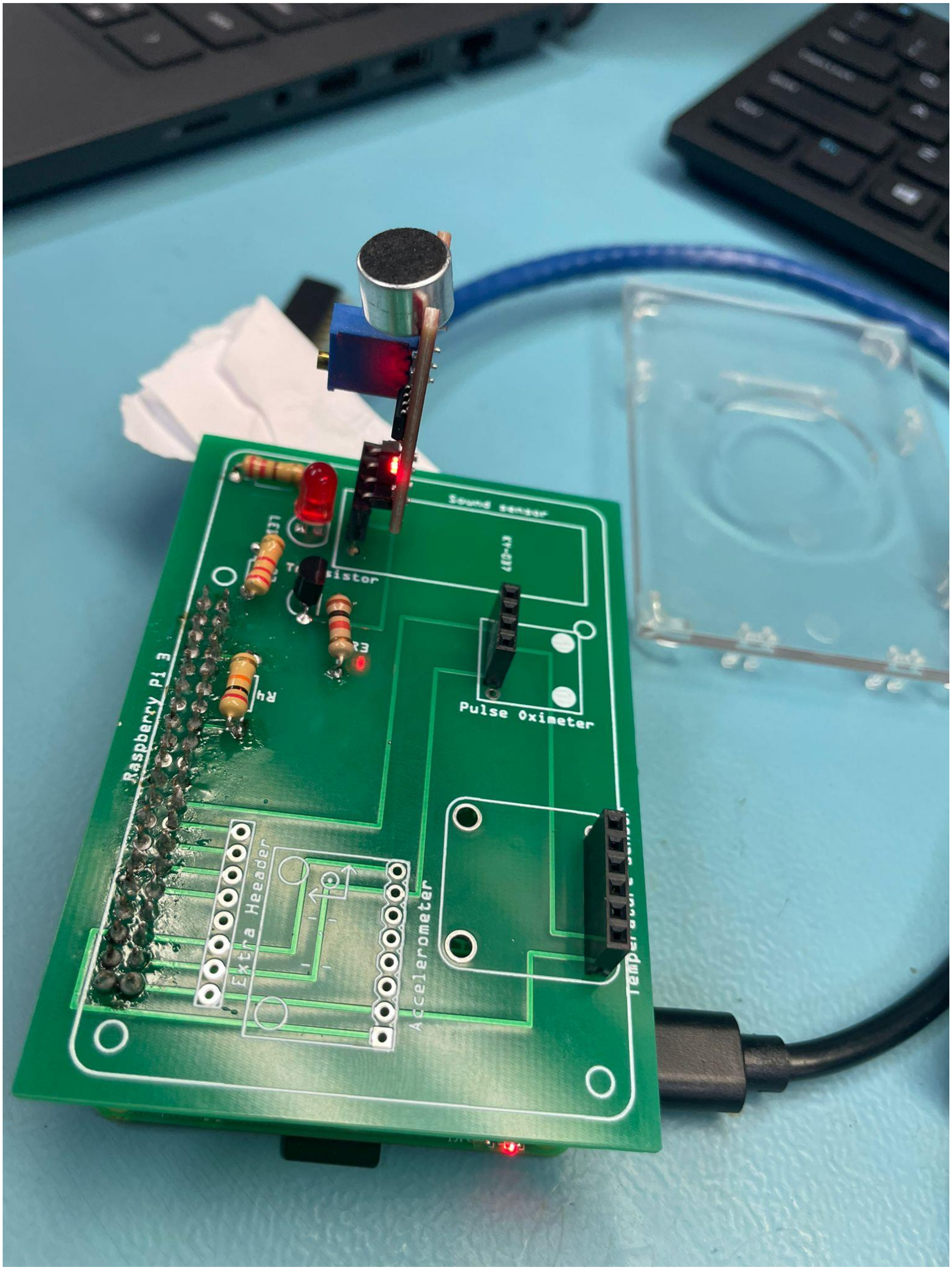




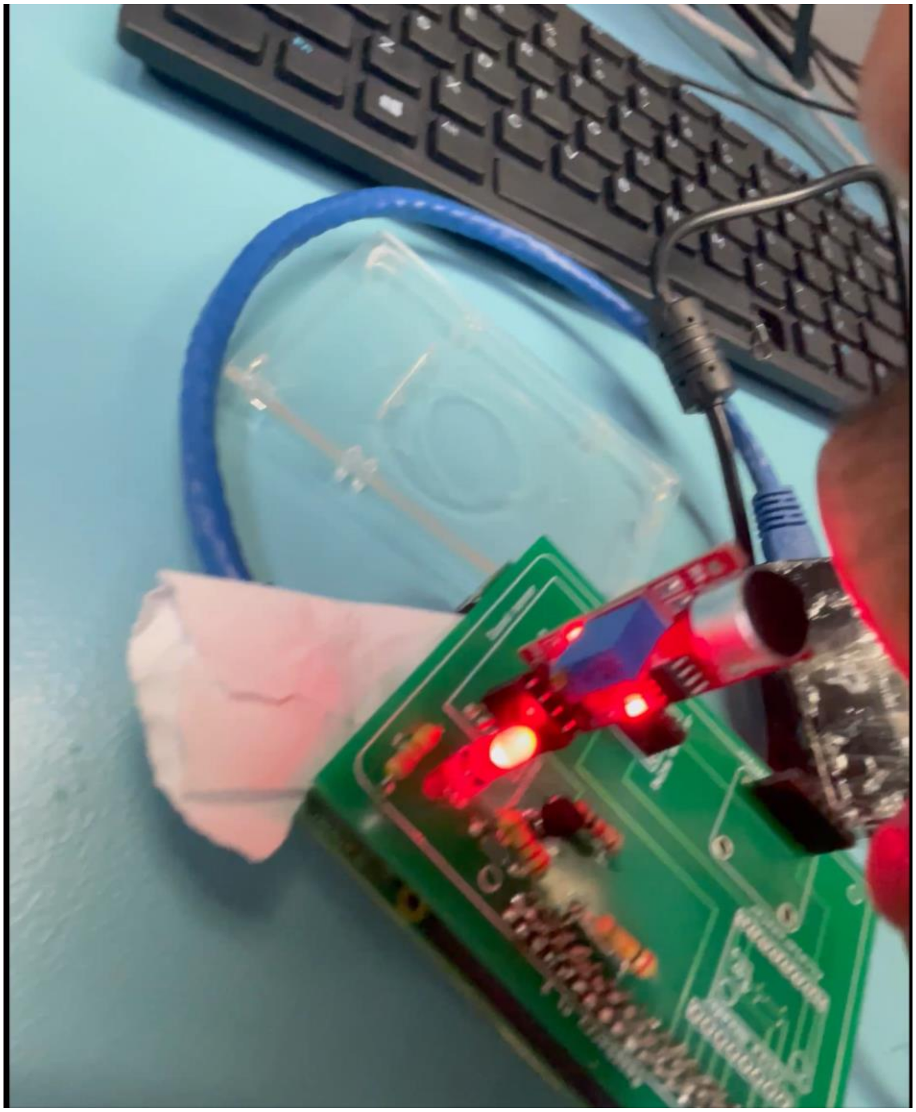
9. Usage Process:

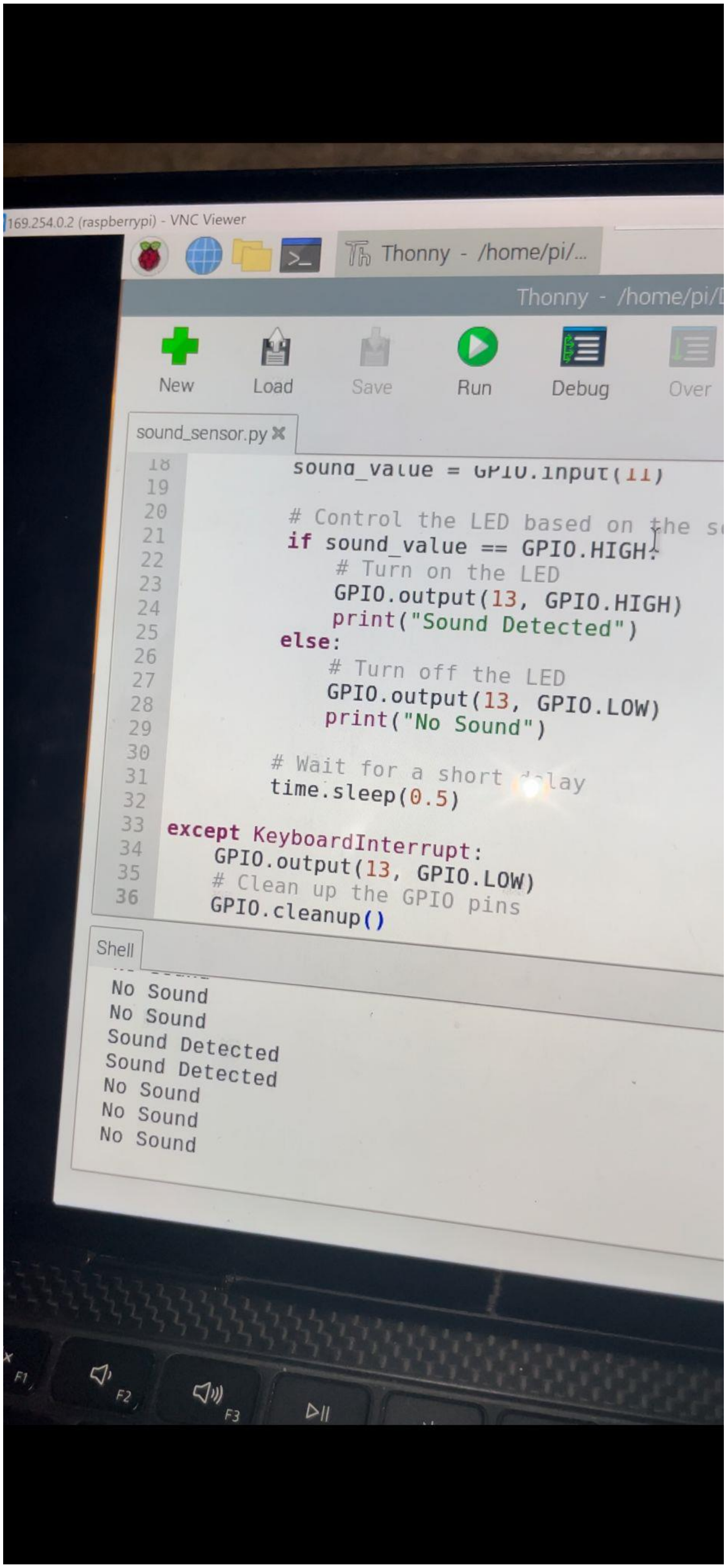
With the baby's health meter, the guardian can view the baby's quality of sleep. The following will show how sound is accurately measure and displayed:


NC Viewer
Thonny - /home/pi/...
Thonny - /home/pi/Desktop/sound_sensor
New
Load
Save
Run
Debug
Over
Into
Out
und_sensor.py
import RPi.GPIO as GPIO
import time
# Set the GPIO mode
GPIO.setmode(GPIO.BOARD)
# Set the GPIO pin numbers
sound_pin = 11
led_pin = 13
# Setup the GPIO pins
GPIO.setup(11, GPIO.IN)
GPIO.setup(13, GPIO.OUT, initial=GPIO.LOW) # Changed ini
try:
while True:
# Read the sound sensor value
sound_value = GPIO.input(11)
Shell
No Sound
No Sound
No Sound
No Sound
Python 3.7.3 (/usr/bin/python3)
>>>


Sound sensor
Pulse Oximeter
Raspberry Pi 3
Extra Heeader
Accelerometer
R3
R4

- Make noise near the sensor.




169.254.0.2 (raspberrypi) - VNC Viewer
Thonny - /home/pi/...
New
Load
Save
Run
Debug
Over
sound_sensor.py
# Control the LED based on the
if sound_value == GPIO.HIGH:
# Turn on the LED
GPIO.output(13, GPIO.HIGH)
print("Sound Detected")
else:
# Turn off the LED
GPIO.output(13, GPIO.LOW)
print("No Sound")
time.sleep(0.5)
except KeyboardInterrupt:
GPIO.output(13, GPIO.LOW)
# Clean up the GPIO pins
GPIO.cleanup()
Shell
No Sound
No Sound
Sound Detected
Sound Detected
No Sound
No Sound
No Sound

- As shown in the pictures above when no noise is being detected the LED is not ON but when sound is received by sensor the LED is ON and sends reading to Raspberry Pi.

10. Testing and Validation:

Raspberry Pi results:

- A reading is sent back displaying that the sound was detected.

Review conclusion:

- After receiving the results, restore the sleep status of the child. Repeat measurements to monitor your child's sleep over time.

11. GitHub repository:

The project code and files for the software App are in a dedicated GitHub repository for easy access and collaboration. You can find the archive at the following link:

https://github.com/ZoyebaMahbub5837/InfantHealthMonitor

12. Testing Results and Outputs:

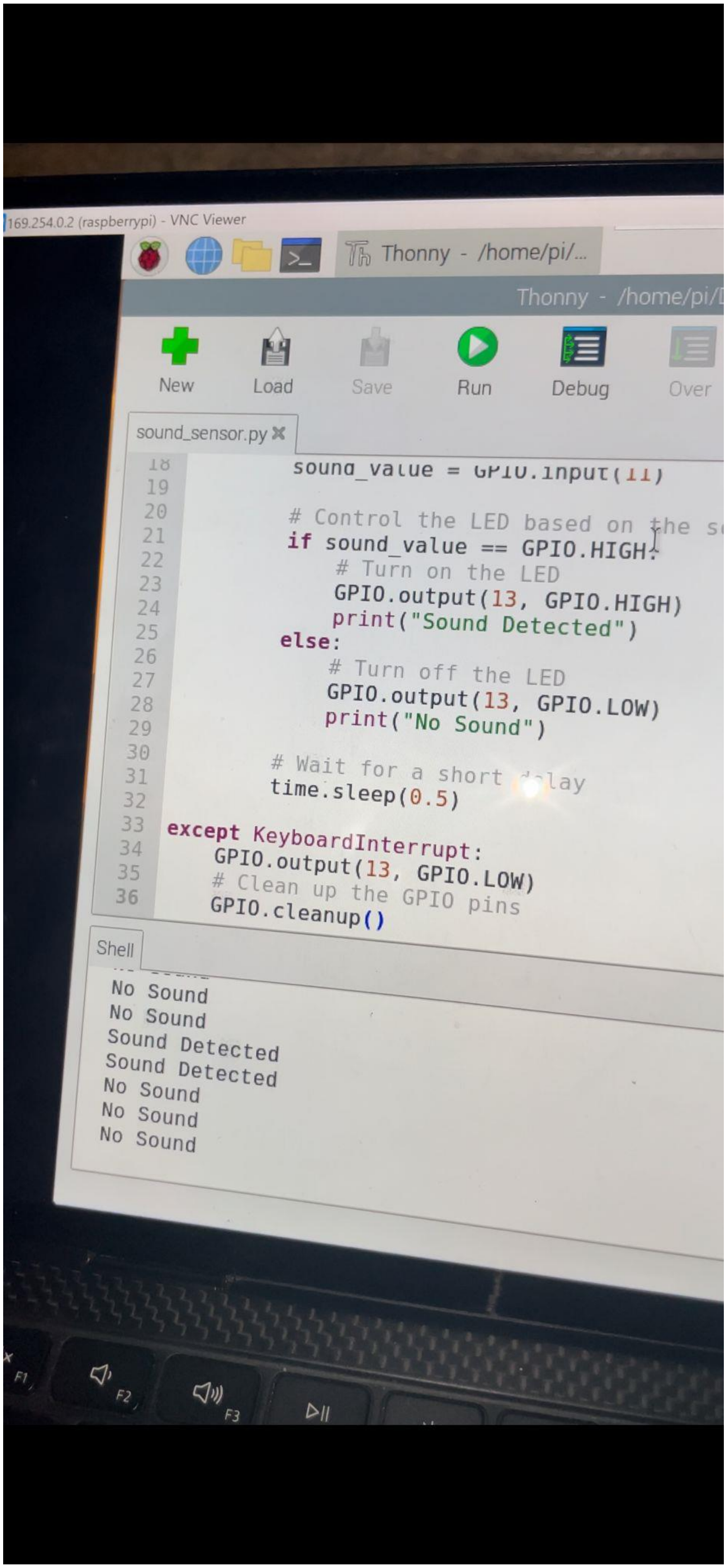
Rigorous testing has confirmed the accuracy and functionality of the baby health monitor.

Test 1: Make loud noise far from sensor

Test 2: Make quiet noise close to sensor

Test 3: Make a crying sound from median distance

Screenshot:


169.254.0.2 (raspberrypi) - VNC Viewer
Thonny - /home/pi/...
New
Load
Save
Run
Debug
Over
sound_sensor.py
sound_value = GPIO.input(11)
# Control the LED based on the s
if sound_value == GPIO.HIGH:
# Turn on the LED
GPIO.output(13, GPIO.HIGH)
print("Sound Detected")
else:
# Turn off the LED
GPIO.output(13, GPIO.LOW)
print("No Sound")
# Wait for a short delay
time.sleep(0.5)
except KeyboardInterrupt:
GPIO.output(13, GPIO.LOW)
# Clean up the GPIO pins
GPIO.cleanup()
Shell
No Sound
No Sound
Sound Detected
Sound Detected
No Sound
No Sound
No Sound

These results confirm accurate measurement and representation of important metrics. Approximate alignment with expected values confirms the reliability of the display. Real-time Raspberry Pi imaging promotes rapid decision-making for the baby's well-being.

13. Troubleshooting and Learning:

Challenge 1: Wrong sensor data

Wrong data caused false sensor readings. Resolution: signal filtering and fix connection. Lesson learned: proper connection ensures reliable results.

Challenge 2: Miscommunication

LM393-Raspberry Pi inconsistent communication. Solution: Optimized protocol and interrupted handling. Lesson Learned: Solid communication ensures smooth communication.

Challenge 3: LED indicator

The LED not lighting when sensor active, confusing users. Solution: Fixed transistor and circuit problems.

Lesson: Small components make a difference in performance.

14. Discussion and Insights:

Real-Life Impact:

The Baby Health Monitor redefines child health care and offers potential in different scenarios:

Home Health: Allows parents to monitor sleep quality at home, allowing for timely intervention.

Neonatal wards: Nurses can monitor sleep without giving having to monitor themselves, which improves newborn care.

Remote Monitoring: Critical information can reach remote healthcare workers, helping in areas with limited access.

Preventive maintenance: Regular monitoring facilitates early problem detection and preventive measures.

Hardware Software Integration Overview: Hardware and software integration is revolutionizing healthcare.

Accuracy and reliability: The synergy of hardware and software improves the accuracy of health measurements.

User-centric design: Intuitive user interfaces provide caregivers with user-friendly experiences.
Real-time tracking: Fast data processing helps with timeliness.

Scalability: Adapting to changing needs and technological developments.

Innovations in healthcare: The project is an example of interdisciplinary innovation. The Baby Health Monitor hardware and software alliance is a pioneering force in healthcare reform that reflects the transformative power of technology to improve children's well-being.

15. Conclusion:

The Biosensor Bracelet project is the pinnacle of an innovation and technical integration demonstration in the paediatric medical industry. We were able to make a practical and dependable infant health monitor by bringing together the LM393 sensor, Raspberry Pi, NPN transistor, resistors, PCB board, and LED light in a harmonic way. For carers, parents, and healthcare professionals, this monitor offers measurement and real-time visualization of sound patterns and frequency.

16. Code Appendix:

Sound.py:

```
import RPi.GPIO as GPIO
import time
import board
import busio
import threading


# Set the GPIO mode
GPIO.setmode(GPIO.BCM)

# Set the GPIO pin numbers
sound_pin = 13
led_pin = 21

# Setup the GPIO pins
GPIO.setup(13, GPIO.IN)
GPIO.setup(21, GPIO.OUT, initial=GPIO.LOW)  # Changed initial value to LOW

def blink_led():
    while True:
        GPIO.output(led_pin, GPIO.HIGH)
        print("LED ON")
        time.sleep(0.5)
        GPIO.output(led_pin, GPIO.LOW)
        print("LED OFF")
        time.sleep(0.5)

def sound_sense():
    while True:
        # Read the sound sensor value
        sound_value = GPIO.input(13)
```

```
        # Control the LED based on the sound sensor value
        if sound_value == GPIO.HIGH:
            # Turn on the LED
            print("Sound Detected")
        else:
            # Turn off the LED
            print("No Sound")

        # Wait for a short delay
        time.sleep(0.5)

try:
    led_thread = threading.Thread(target=blink_led)
    led_thread.start()

    # Create and start the accelerometer thread
    sound_thread = threading.Thread(target=sound_sense)
    sound_thread.start()

    # Wait for both threads to finish (which will not happen in this case)
    led_thread.join()
    sound_thread.join()


except KeyboardInterrupt:
    GPIO.output(21, GPIO.LOW)
    # Clean up the GPIO pins
    GPIO.cleanup()
```