SLEEP DOC

Real-Time Vital Signs Detection using BM502 mmWave Sensor

AoA Estimation and GUI + Touch based KIOSK

Intern Name: SAI CHIRAG R
Institution: NIT ANDHRA PRADESH

Guide: Prof. A. Routray, IIT Kharagpur

Duration: May 2025 – July 2025

Introduction

- mmWave radar for contactless health monitoring
- Objective: Detect breathing and heart rate of people
- Visualize on GUI and send data to Arduino LCD / KIOSK

Hardware Setup

BM502 mmWave Sensor

Raspberry Pi 3B+ for signal processing and GUI

UART connection to Arduino

LCD 3.5" TFT display via Arduino

Raspberry pi Touch Display

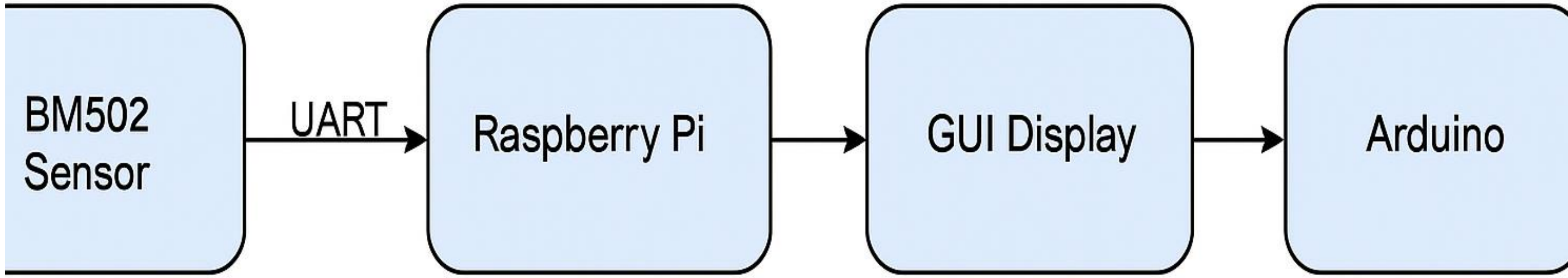
WS2812 LED ring

Speaker

Hardware Setup

BM502 and Arduino are connected to Raspberry Pi through USB.

System Block Diagram



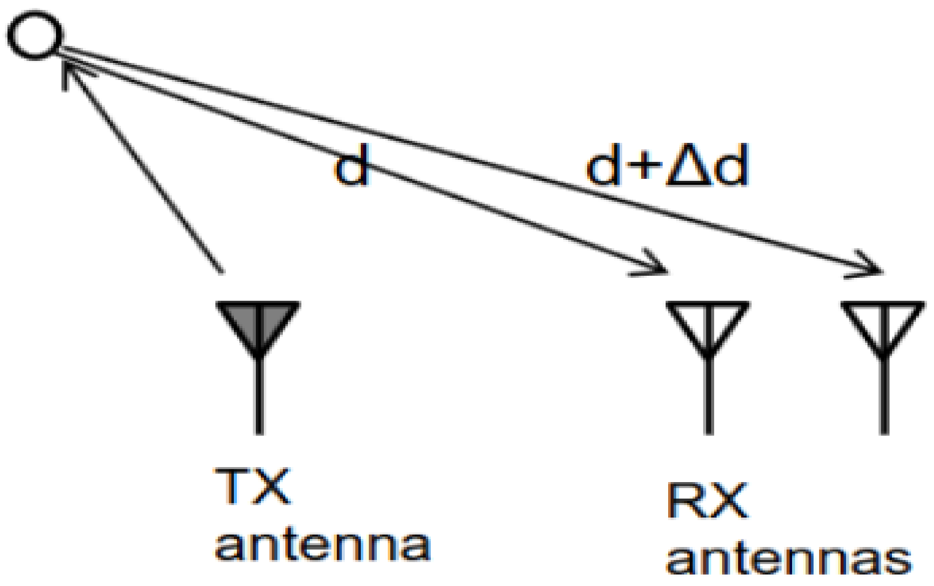
Objective

Locate the position of subjects using Angle of Arrival.

Extract vitals at the estimated angles.

Angle of Arrival

A Radar uses a Transmitting (Tx) antenna to transmit signal. It reflects of a distant object and the reflection is incident on Receiving (Rx) antennas.



Angle of Arrival (AoA)

Signal Delay in ULA (Uniform Linear Array)

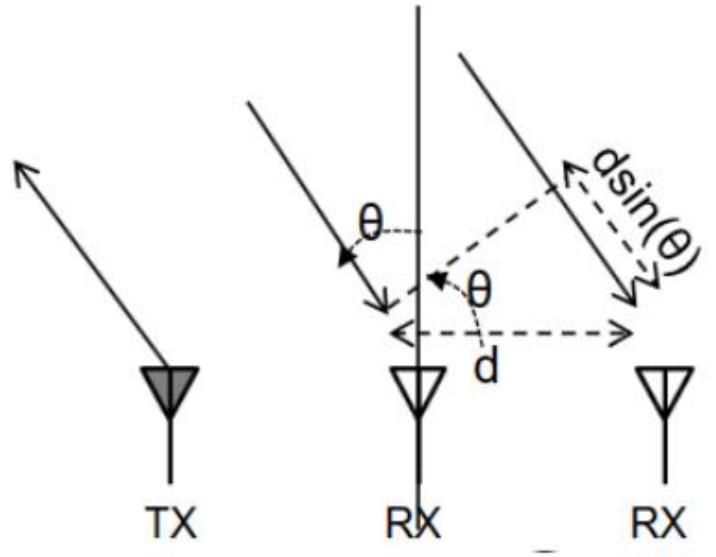
For an incoming wave with wavelength λ, angle of arrival θ, and velocity v, the delay between successive antenna elements (for the *i*th source) is:

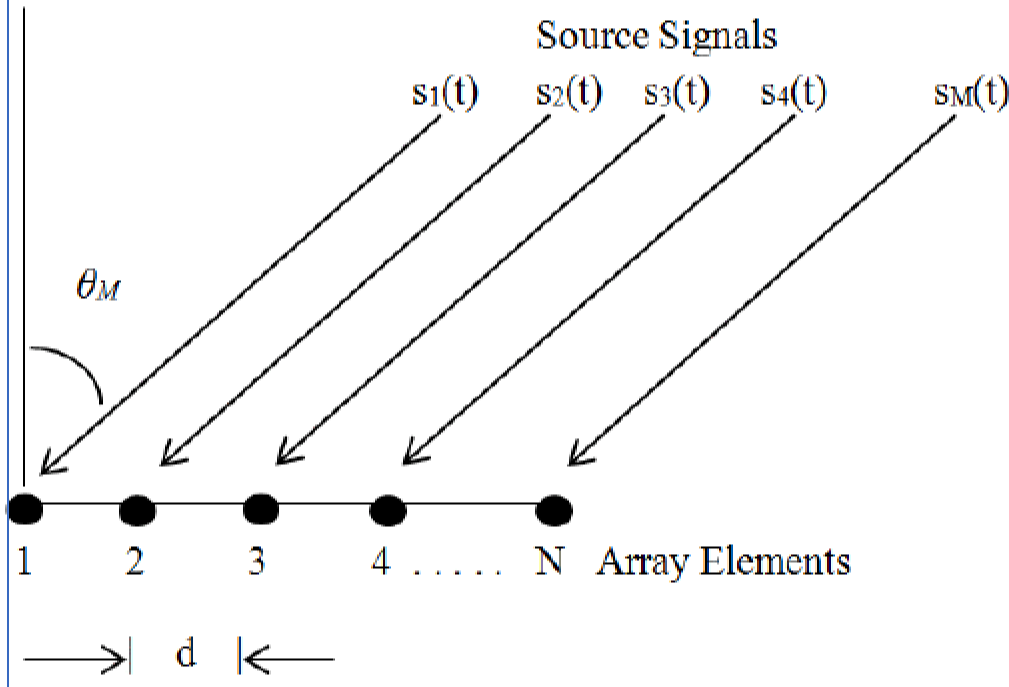
$$\tau = d \sin\theta / v$$

- d: spacing between elements

If the received signal at element 1 is x1(t), then received signal at the element k is delayed by

$$\tau_k = (k-1)\tau = (k-1)(d\sin\theta_k)/c$$





Then the received signal with delay at element k is

$$x_k(t) = e^{-j\omega\tau_k} x_1(t) = e^{-j\frac{2\pi}{\lambda}d(k-1)\sin\theta} x_1(t)$$

Assuming that there are M sources and some noise arriving at each of the elements simultaneously, the total received signal at the kth element is given by

$$x_k(t) = \Sigma_{i=1}^{M} s_i(t) e^{-j\frac{2\pi}{\lambda}d\sin\theta_i} + n_i(t)$$

Where θi represents the direction of arrival of the ith source. Putting received signals from all N elements together

$$\boldsymbol{x}(t) = [x_1\ x_2\ \ldots\ x_N]^T = [1\ e^{-j\psi}\ \ldots\ e^{-j(N-1)\psi}]^T\ \boldsymbol{s}(t) \\ = \boldsymbol{a}(\theta)\boldsymbol{s}(t)$$

Steering Vector

- $a(\theta)$ is called the steering vector and $\psi = 2\pi d \sin\theta_i/\lambda$ is called the spatial frequency of the signal associated with the ith source.
- The M signal sources received by the array arriving at angles $\theta_1, \theta_2, \ldots, \theta_M$ can be modeled as

$$\boxed{\boldsymbol{x}(t) = \boldsymbol{A}\,\boldsymbol{s}(t)}$$

- where A is array steering matrix having Vandermonde structure given by

$$A = [a(\theta_1), a(\theta_2), \ldots. a(\theta_M)]$$

BM502 sensor

01

Number of receiving antennas = 4

02

Number of sources = 1

03

Frame rate = 20

Data Processing Flow

- Raw I/Q Data → MUSIC Algorithm → AoA Estimation

- AoA selection → Signal projection → FFT analysis

- Bandpass filtering → Extract Heart/Breath Rates

Raw Data Extraction

- Step 1: Receiving Raw Binary Data from UART

Raw TLV data (248 bytes): 060000008000000000000d00007abb4b202400000b00210082552844a864623fa6c0c73e00d8bd420058e3428b
2e5a4200b09a4200c0a841c459f94000803b416524a8407e150541023a6c3ee1e1423e18bf6640918c605179c861440000803f00404e4100b0654200
000000000000000000000000000000000000000000000000000000000000000020000005c0000007b036702e0f35afbfc08240f080908fde100a8ecfc
effa074c05e8053d04c9fc64fa90fe4403a50478ff76fbd3febf035f02d9fe270034ff62feb8fea7ff0102be014eff87fdbcfed1006103c201b0fd43
fe16006600fc002f0142ff0f0f0f0f0f0f0f0f0f0f0f0f

- Step 2: Convert Raw Bytes to Integers

iq_raw = np.frombuffer(raw_data_bytes, dtype=np.int16)

raw_data_bytes: binary byte stream from UART

np.int16: because each I and Q sample is 2 bytes (16-bit signed integer)

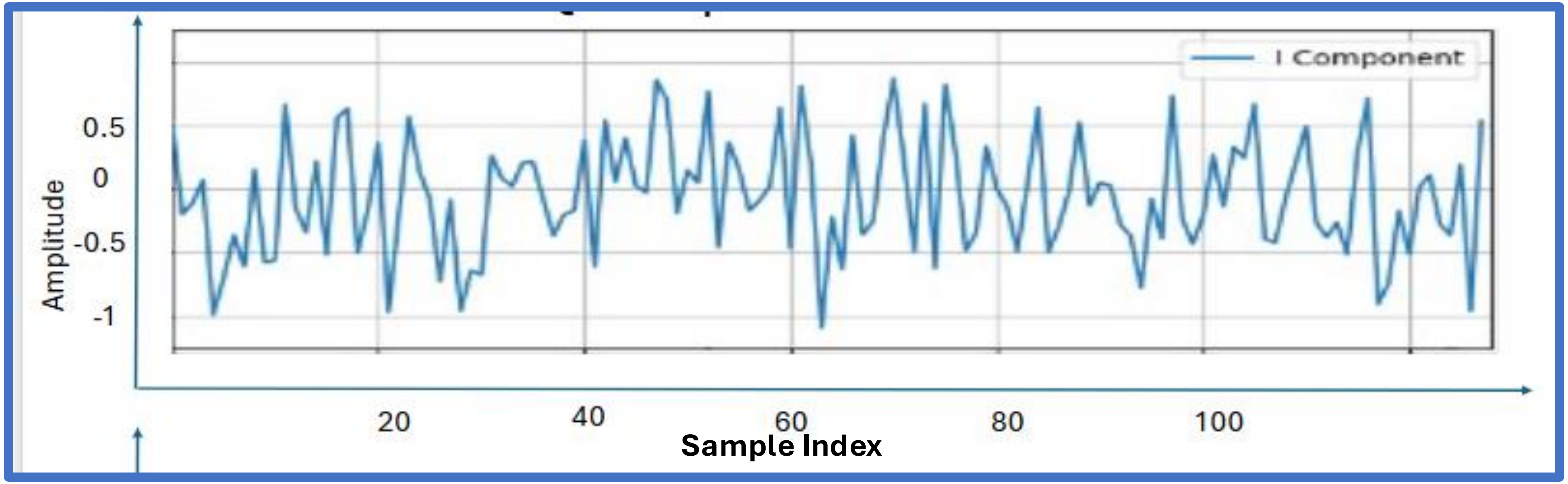
Step 3: Reshape into Antenna × Sample × 2 Format

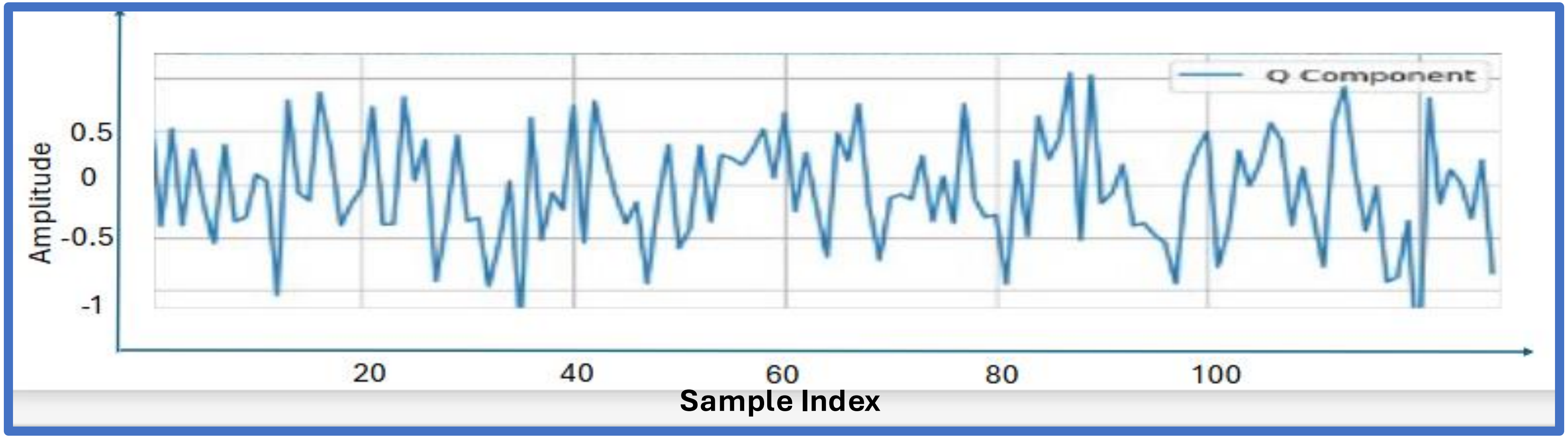
- **iq_raw = iq_raw.reshape((num_antennas, samples_per_frame, 2))**
  - num_antennas = 4
  - samples_per_frame = 128
  - Each antenna has 128 (I, Q) pairs → 256 int16s.
    4 antennas → 1024 values total → flat array of length 1024.

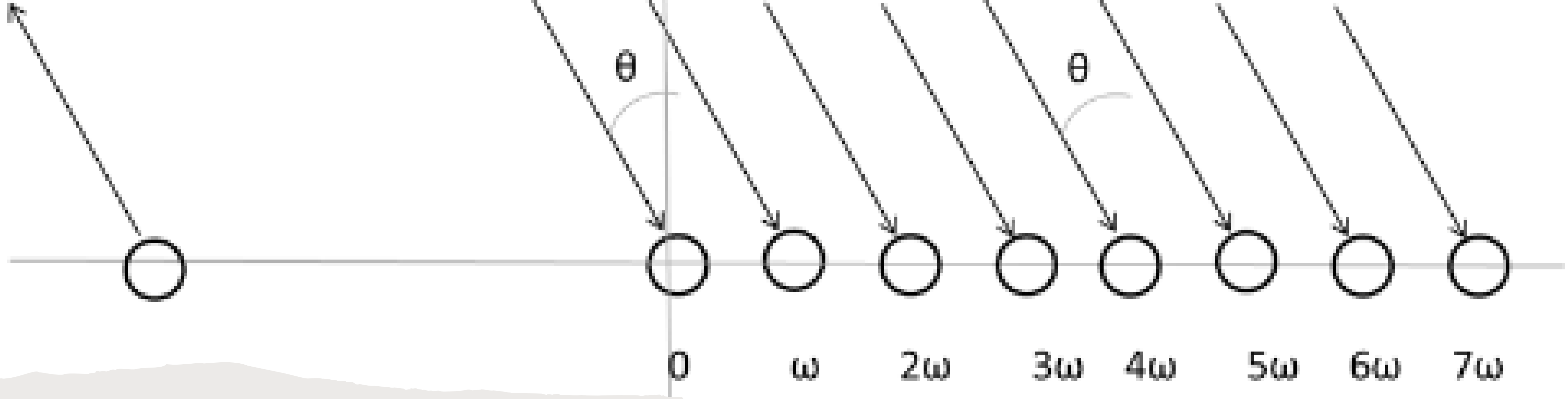
Step 4: Combine I and Q into Complex Numbers

- **iq_data = iq_raw[..., 0] + 1j * iq_raw[..., 1]**
  - After I/Q demodulation:
  - x(t)=I(t)+jQ(t)=Ae^jϕ

I/Q Samples Plot







AoA estimation algorithms.

- A. Beamforming
  1) Bartlett's Conventional Beamforming
  2) Capon's Beamforming
- B. Subspace Based Techniques
  1) Multiple Signal Classification (MUSIC)

MUSIC Algorithm

MUSIC is a high-resolution algorithm for **estimating the direction of arrival (DoA/AoA)** of multiple signal sources using an antenna array.

- Assumptions

Narrowband signals: The signal's bandwidth is much smaller than the carrier frequency.

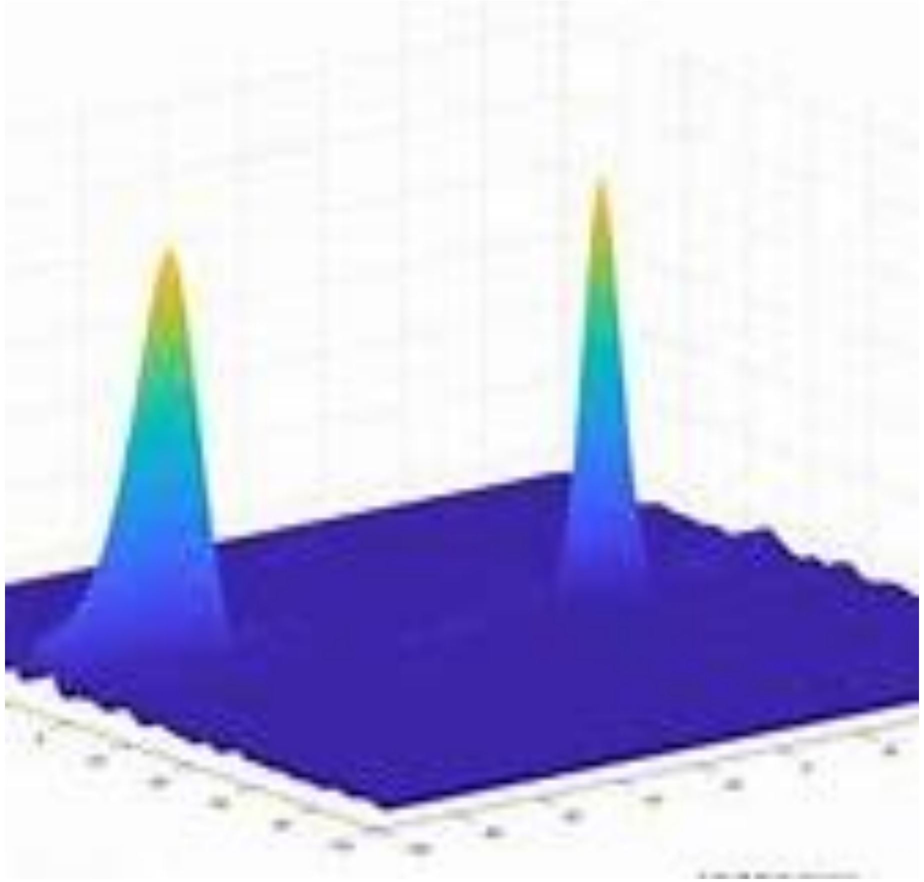
Far-field sources: Waves from sources arrive as planar wavefronts at the antenna array.

Uncorrelated sources: Sources are statistically independent and uncorrelated.

Additive white Gaussian noise (AWGN): The noise is independent of the signal and identically distributed across antennas.

Number of sources is known: 1 Sources.

More antennas than sources: 4 Antennas > 1 Source



Why MUSIC?

- This technique gives better resolution in comparison to Beamforming because it exploits the Eigen decomposition property of the data covariance matrix
- It decomposes the space into signal and noise subspaces to estimate the direction of arrival of signals.
- It further exploits the orthogonality of signal subspace with noise subspace.
- Subspace-based algorithms (like MUSIC, Root-MUSIC) offer superior resolution and accuracy compared to Classical Beamforming, especially in distinguishing closely spaced sources.
- These advanced DOA estimation techniques are more sensitive to low SNR, requiring higher signal-to-noise conditions to maintain performance.

MUSIC has been experimentally tested to be most reliable compared to other algorithms.

Hence the already tested MUSIC algorithm will be used for the estimation of the subjects

- **Time Delay** between adjacent elements:

$$\boxed{\tau = d \sin \theta_0 / c}$$

- **Phase Difference** for a narrowband signal of frequency f:

$$\varphi = \omega \tau$$

$$\varphi = 2\pi f \times \frac{d \sin \theta_0}{c}$$

$$\varphi = 2\pi f \times \frac{d \sin \theta_0}{f \lambda}$$

$$\varphi = 2\pi \frac{d \sin \theta_0}{\lambda}$$

- If the incoming signal is denoted as s(t), the phase delayed version of the incoming signal at the adjacent element is given as

$$\boxed{s(t)e^{j\omega(t-\tau)} = s(t)e^{j(\omega t-\varphi)}}$$

- So, At the nth element:

$$\boxed{s(t)e^{j\omega t}e^{-j(N-1)\varphi}}$$

- Thus taking all the signals received by the ULA and expressing them in matrix form we have the total received signal as

$$\boxed{x(t) = s(t)e^{j\omega t}\begin{bmatrix} 1 \\ e^{-j\varphi} \\ \cdot \\ \cdot \\ \cdot \\ e^{-j(N-1)\varphi} \end{bmatrix}}$$

- Each ULA element receives the signal with a phase shift due to propagation delay. Ignoring the carrier, the received signals form a phase-shifted vector.

$$x(t) = s(t) \begin{bmatrix} 1 \\ e^{-j\frac{2\pi d}{\lambda}\sin\theta_0} \\ \vdots \\ e^{-j(N-1)\frac{2\pi d}{\lambda}\sin\theta_0} \end{bmatrix}$$

$$x(t) = s(t)a(\theta_0)$$

- The **steering vector** represents the phase shifts across ULA elements and depends on the AoA (θ_0), spacing (d), wavelength (λ), and number of elements (N).

- The covariance matrix RXX=E[x(t)xH(t)] is a Hermitian-Toeplitz matrix and after performing eigen-decomposition on this covariance matrix, we have

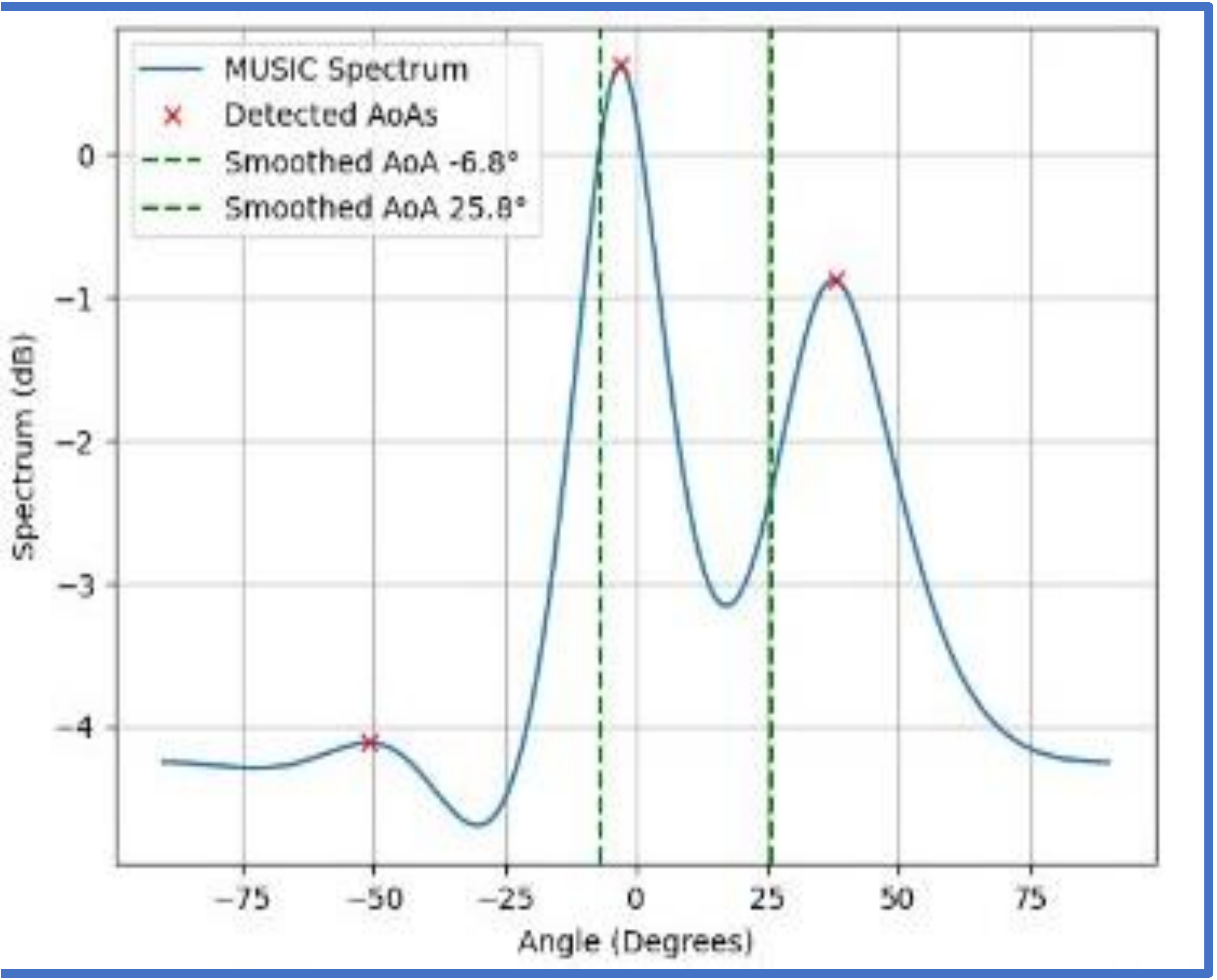
$$\boxed{R_{xx} = A U A^H}$$

- U=diag([Λ1,λ2,….,λN]) is a diagonal matrix and λ1,λ2,…….,λN are called the eigenvalues.
- The covariance matrix R_{xx} can be decomposed into two parts, the signal and noise subspace which are orthogonal to each other separately.

$$\boxed{R_{XX} = A_S U_S A_S^H + A_n U_n A_n^H}$$

- The spatial spectrum function for the DOA estimation for MUSIC algorithm is given by

$$\boxed{P_{MUSIC}(\theta) = \frac{1}{a^H(\theta) A_n A_n^H a(\theta)}}$$



Smooth AoA gives us the average AoAs over a time period to avoid the angle fluctuations every second and get the average position of the subject.

AoA Estimation → Beamforming Pipeline

1. Beamforming After AoA Detection

Once AoAs are estimated (e.g., via MUSIC), apply **beamforming** to isolate signals from each direction.

Beamforming formula:

$$x_\theta(t) = \sum_{n=0}^{N-1} x_n(t) \cdot e^{j\frac{2\pi d}{\lambda} n \sin(\theta)}$$

xn(t): Signal at antenna *n*

θ: Estimated Angle of Arrival

d: Antenna spacing

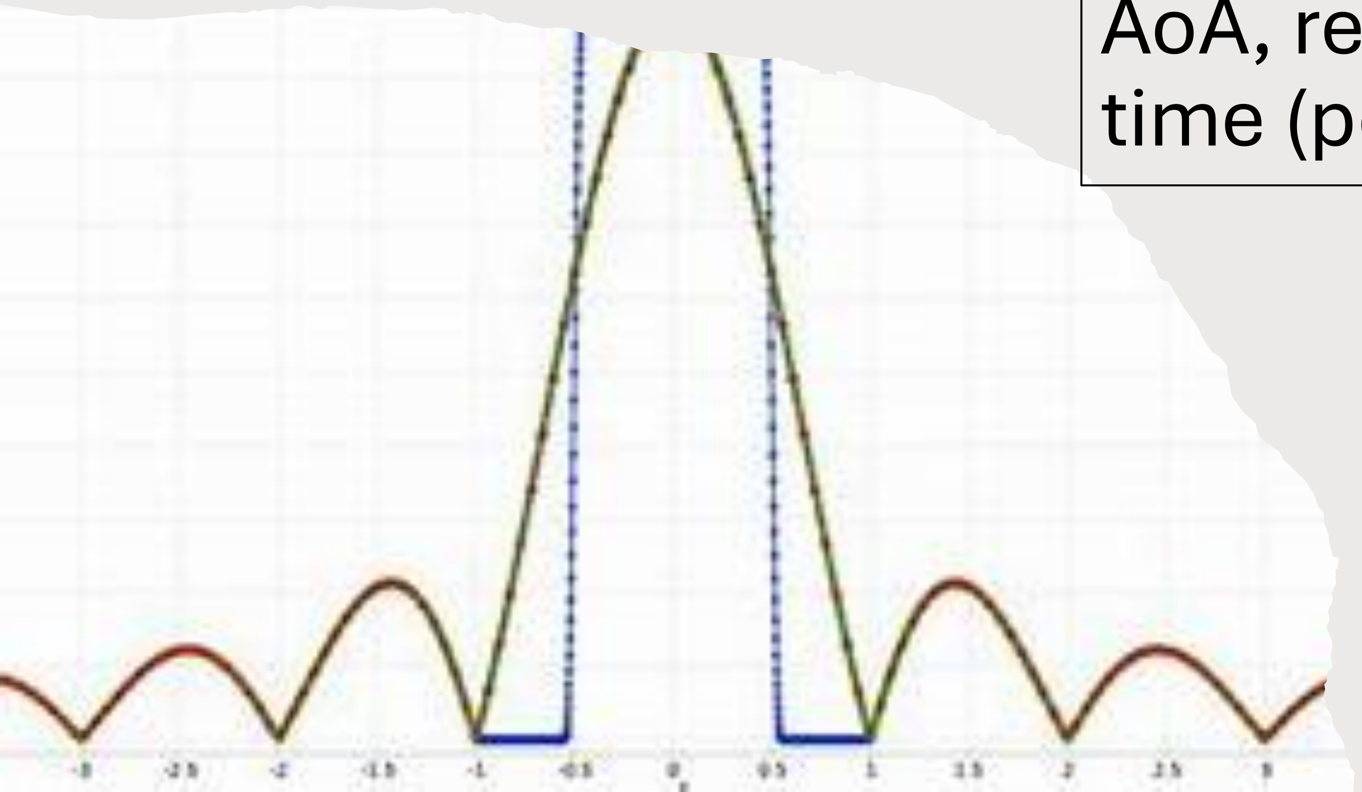
λ: Wavelength

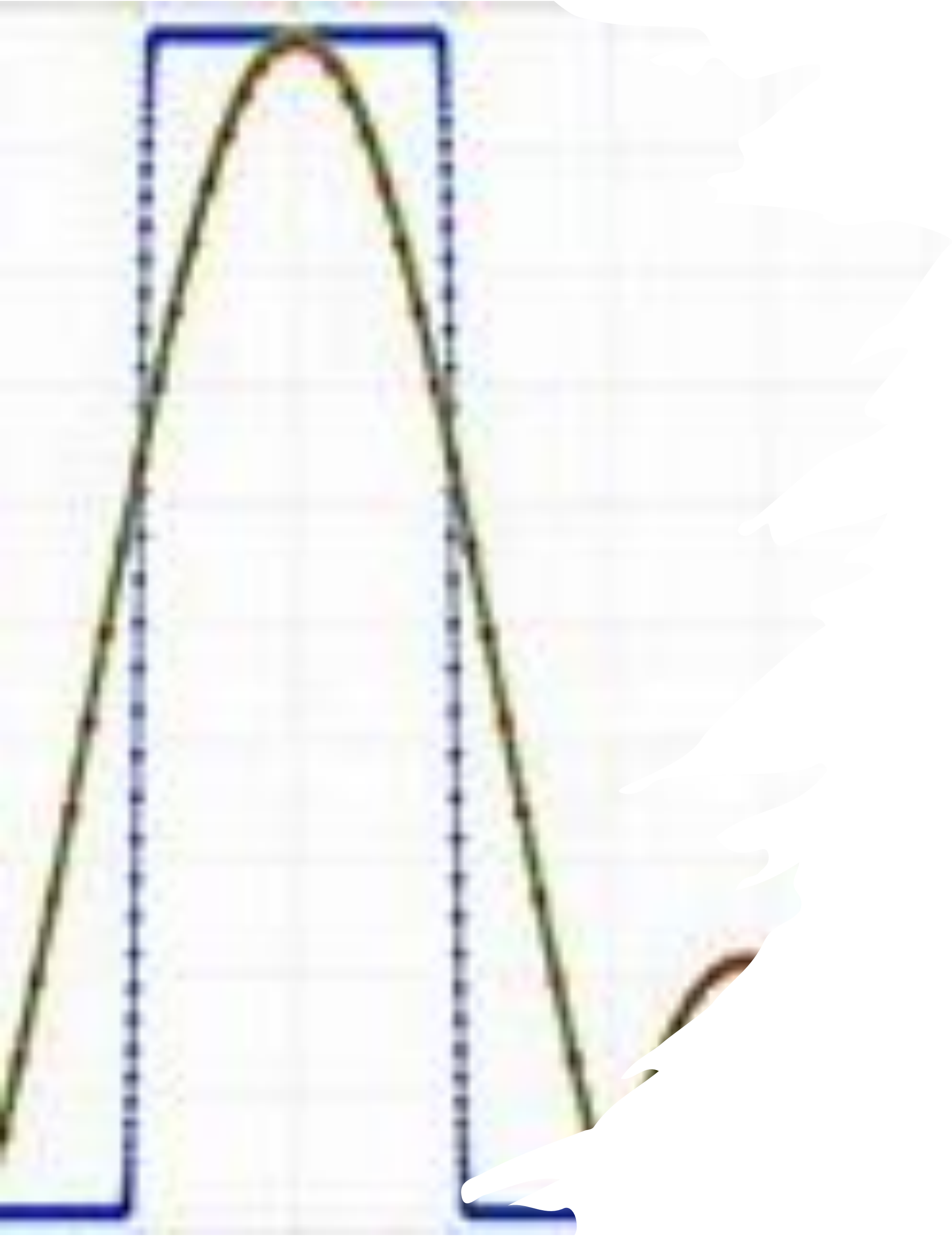
2. Time-Domain Signal Construction

Take **real part** of beamformed signal:

$$s\theta(t)=R\{x\theta(t)\}$$

This gives a 1D waveform for each AoA, reflecting chest motion over time (per person).





Vital Signs Extraction from Beamformed Signals

3. Vital Signs Extraction via FFT

Apply FFT on sθ(t) to get frequency domain signal:

$$S\theta(f)=FFT\{s\theta(t)\}$$

4. Frequency Band Interpretation

Breathing Rate: 0.1–0.5 Hz → 6–30 bpm

Heart Rate: 0.8–2.0 Hz → 48–120 bpm

5. Calculate Rates

Find peak frequency in each band:

Breathing Rate (bpm)= freq(breath)×60

Heart Rate (bpm)= freq(heart)×60

Final output: One heart rate and one breathing rate per AoA (person/source).

Software Algorithm

- Mode 1: Detect : Estimate AoAs.
- Mode 2: Track : Track the AoAs and update vitals at those angles.
- Cycle repeats

Algorithm: Real-Time AoA Estimation and Vital Signs Detection

Initialize: Set serial communication, system constants, and buffers. Start in detect mode.

1

Read Data: Continuously read and decode I/Q data frames from mmWave sensor.

2

Estimate AoA : Scan -90° to +90°, detect AoA peaks.

3

Detect Mode : Use FFT to estimate breathing (0.1–0.5 Hz) and heart rate (0.8–2 Hz).

4

Track Mode : Use AoAs to extract and update vital signs continuously.

5

Display/Output: Show AoAs and vital signs in real time.

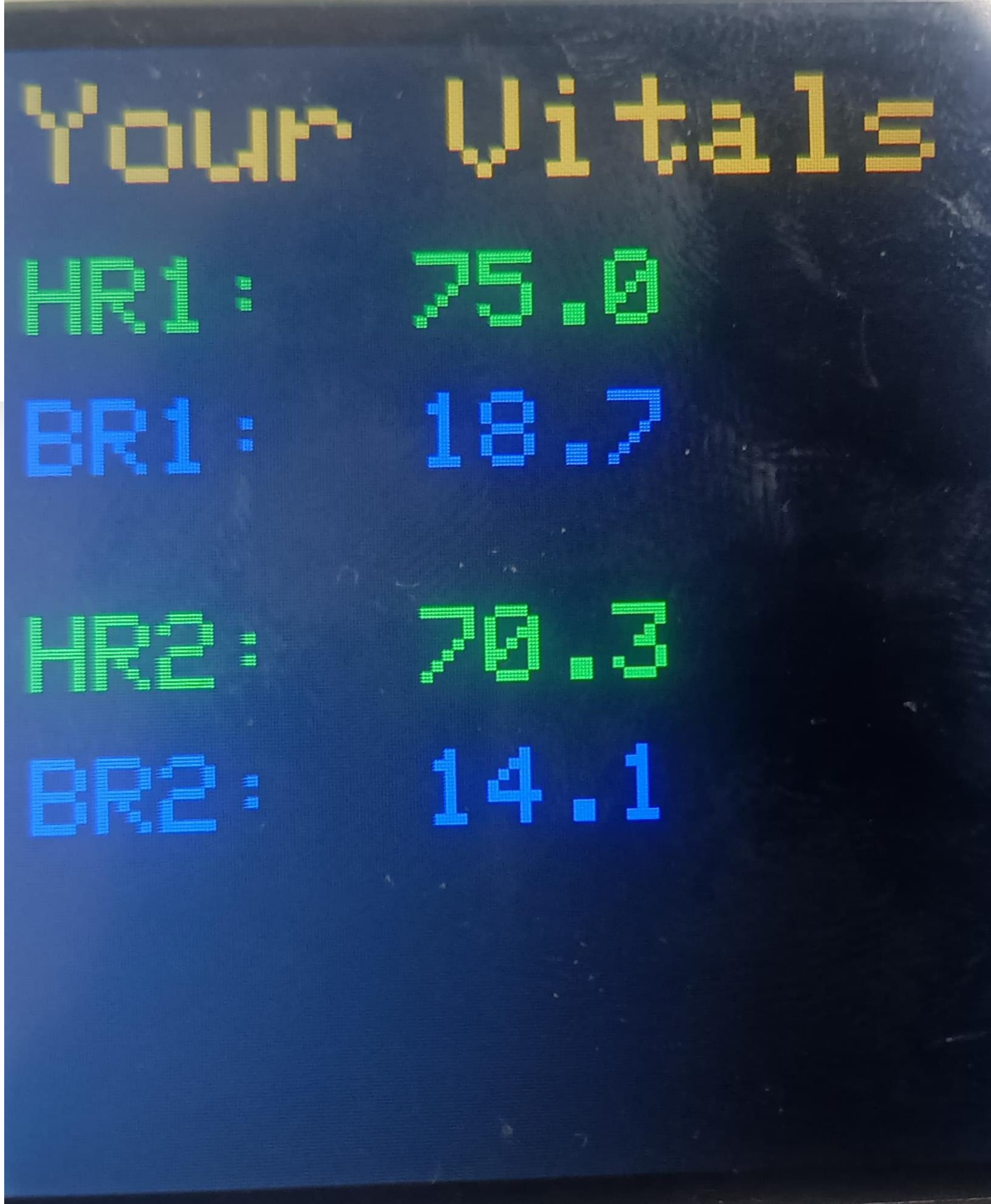
GUI and Serial Communication

- GUI displays real-time vitals
- Serial (UART) data sent to Arduino every 1s
- PyQt/Matplotlib based visualization

Arduino + LCD Integration

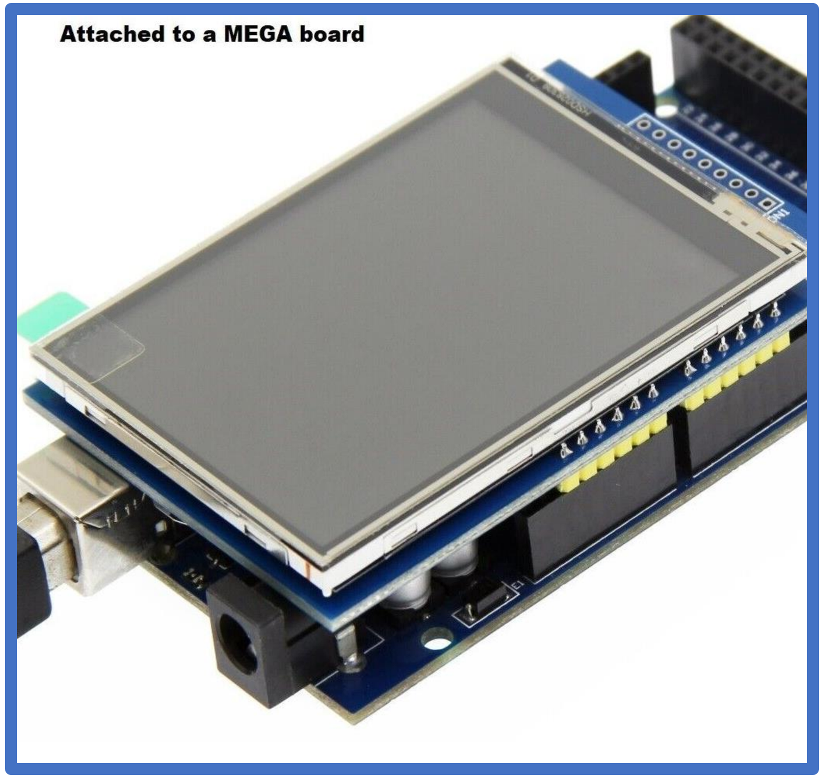
- UART receives heart and breath rate from Raspberry Pi
- Parses and displays on 3.5"TFT LCD Sheild.
- Example: Breath: 18 bpm

  Heart : 74 bpm



Arduino + LCD Integration

Attached to a MEGA board



Phase 2

- Addition of Gesture based sensing.
- Ambient Light integration.
- Sleep inducing sound integration.
- Environmental parameters(Temp.,Humidity, Pressure).
- Building Pi LCD based Touch GUI.

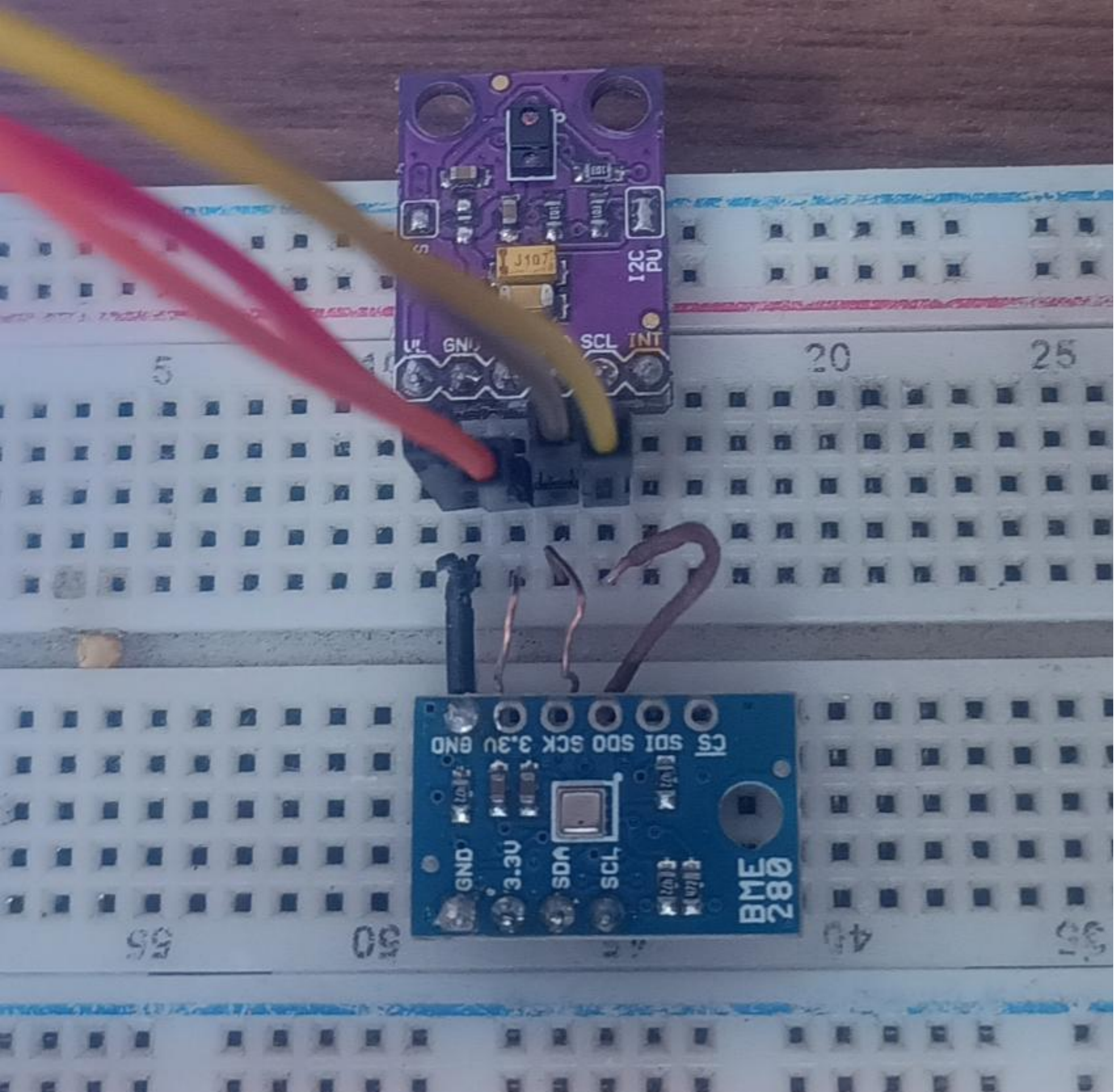
Gesture and Temperature sensor.

- Intergrated Gesture sensor for gesture based operation of the Vitals detection.

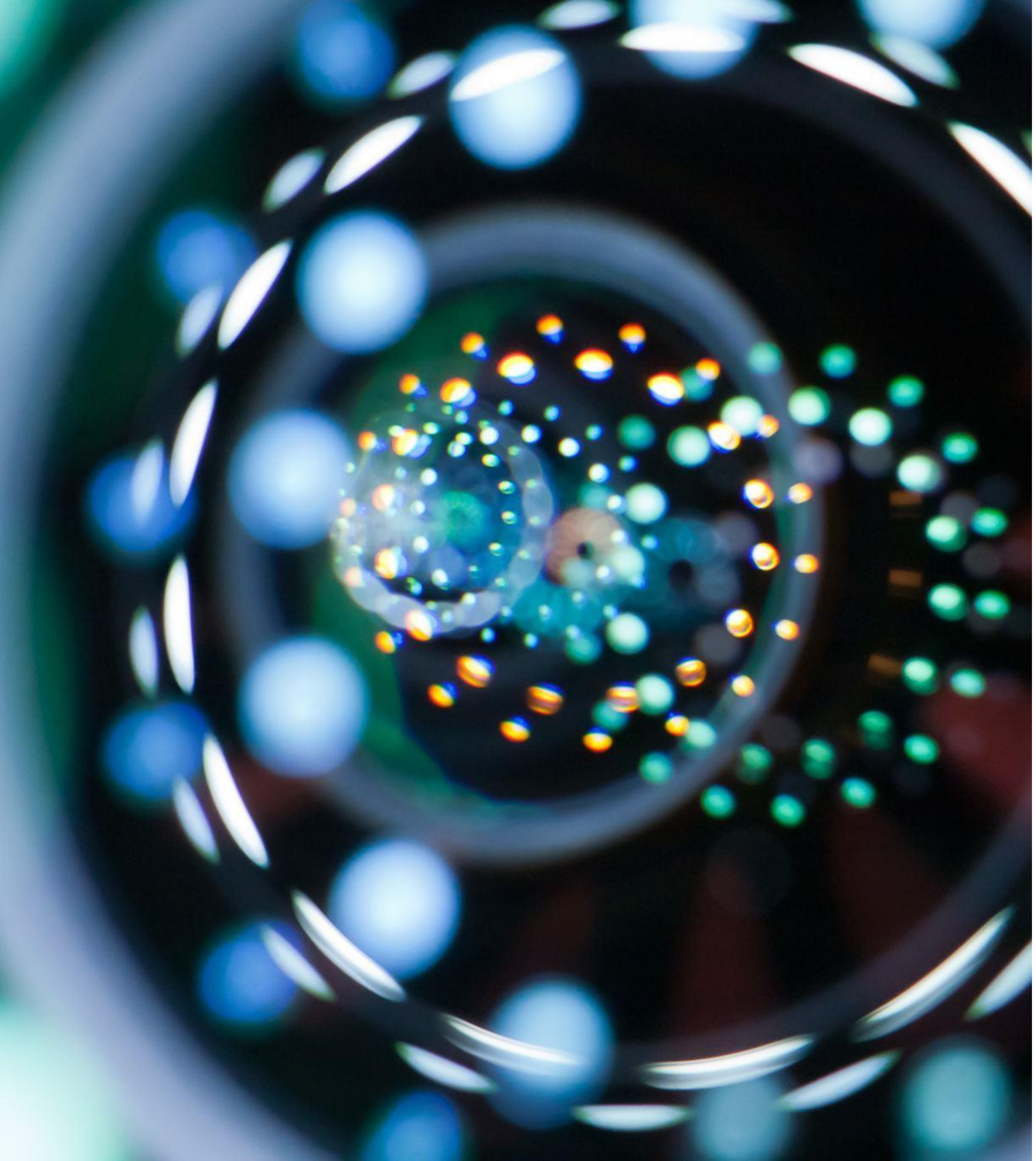
LEFT SWIPE GESTURE – START/CONTINUE

RIGHT SWIPE GESTURE – PAUSE/STOP

- Temperature sensor to measure Temperature, Humidity and pressure of surrounding.



Gesture (APDS9960) and Temperature(BM280) sensor

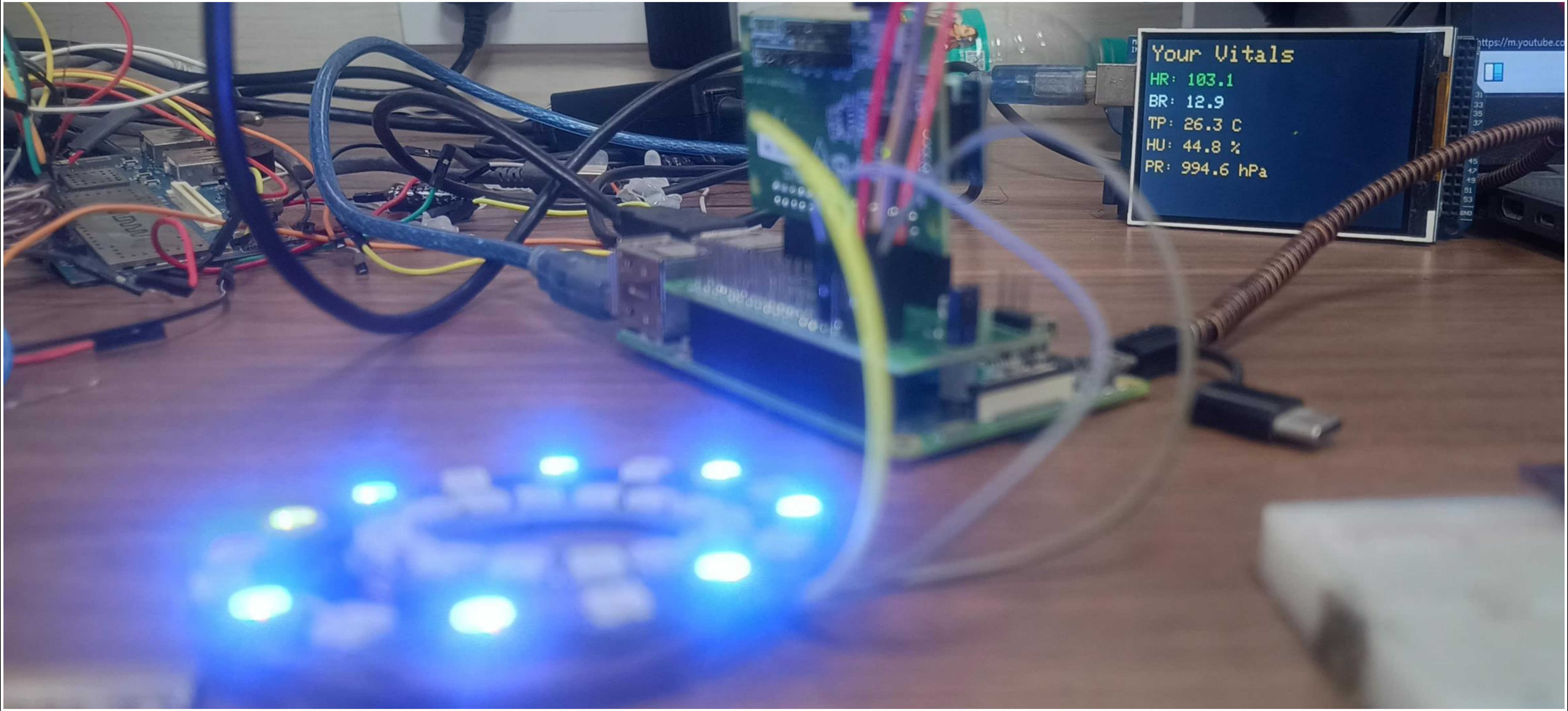


Mood lighting integration.

- Integration of WS2812B-16 Ring LED for the mood light while detecting vitals.

Detection stopped or paused –Loading pattern

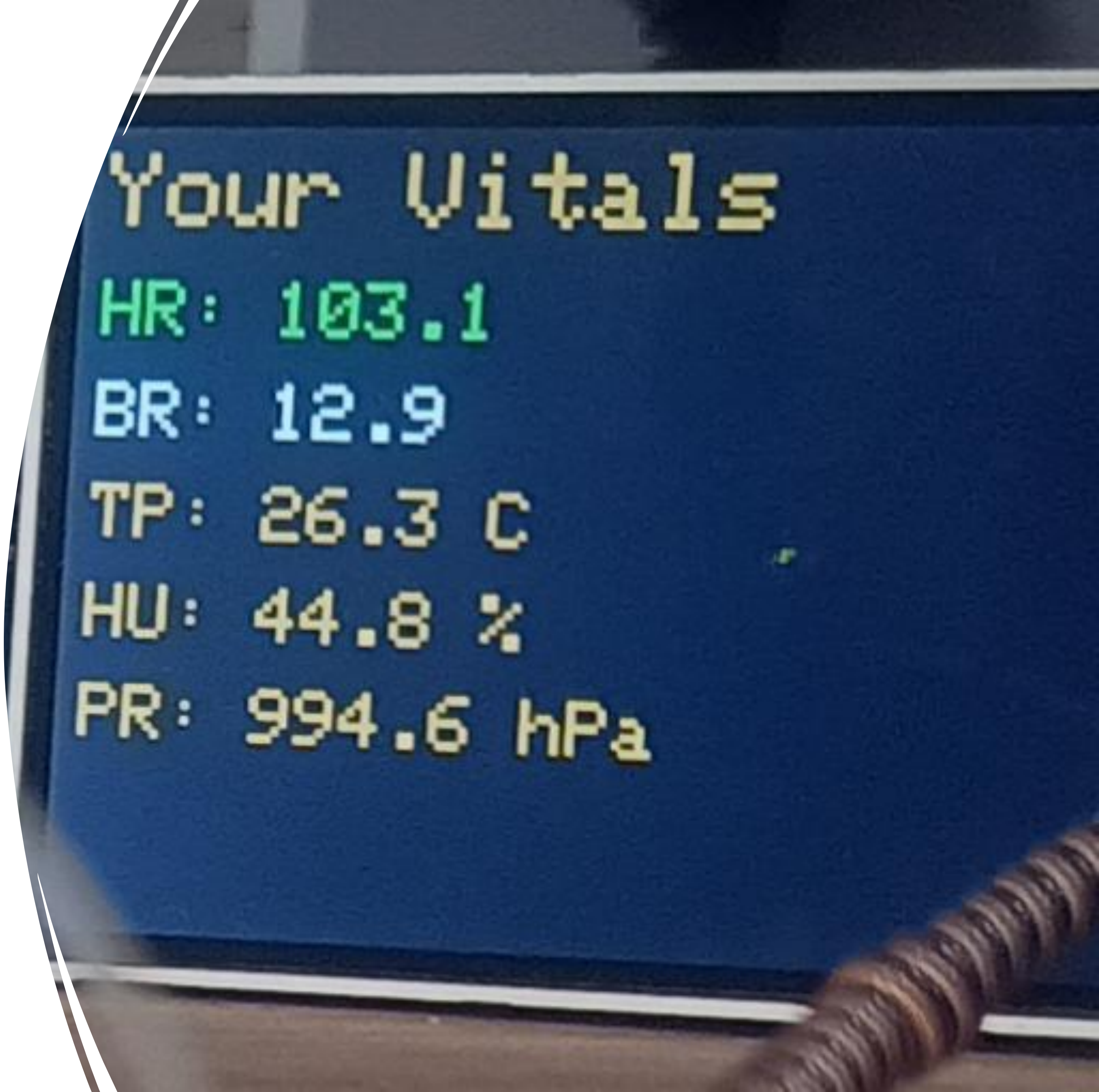
Detection started or continue – Mood light pattern.



Ring LED

Updated LCD

- Additions:
  Temperature
  Humidity
  Pressure



```
2025-06-15 02:26:19, HR: 82.03, BR: 19.92, Temp: 32.60
2025-06-15 02:26:21, HR: 49.22, BR: 14.06, Temp: 32.60
2025-06-15 02:26:23, HR: 62.11, BR: 14.06, Temp: 32.60
2025-06-15 02:26:25, HR: 65.62, BR: 29.30, Temp: 32.62
2025-06-15 02:26:27, HR: 87.89, BR: 29.30, Temp: 32.62
2025-06-15 02:26:29, HR: 83.20, BR: 29.30, Temp: 32.62
2025-06-15 02:26:31, HR: 62.11, BR: 7.03, Temp: 32.62
2025-06-15 02:26:33, HR: 62.11, BR: 26.95, Temp: 32.62
2025-06-15 02:26:35, HR: 64.45, BR: 26.95, Temp: 32.62
2025-06-15 02:26:37, HR: 64.45, BR: 18.75, Temp: 32.62
2025-06-15 02:26:39, HR: 65.62, BR: 18.75, Temp: 32.63
2025-06-15 02:26:41, HR: 65.62, BR: 18.75, Temp: 32.63
2025-06-15 02:26:43, HR: 65.62, BR: 17.58, Temp: 32.63
2025-06-15 02:26:45, HR: 65.62, BR: 17.58, Temp: 32.63
2025-06-15 02:26:47, HR: 65.62, BR: 16.41, Temp: 32.63
2025-06-15 02:26:49, HR: 65.62, BR: 15.23, Temp: 32.62
2025-06-15 02:26:51, HR: 65.62, BR: 16.41, Temp: 32.62
2025-06-15 02:26:53, HR: 65.62, BR: 15.23, Temp: 32.63
2025-06-15 02:26:55, HR: 64.45, BR: 14.06, Temp: 32.63
2025-06-15 02:26:57, HR: 64.45, BR: 14.06, Temp: 32.63
2025-06-15 02:26:59, HR: 64.45, BR: 23.44, Temp: 32.62
```

Data collection

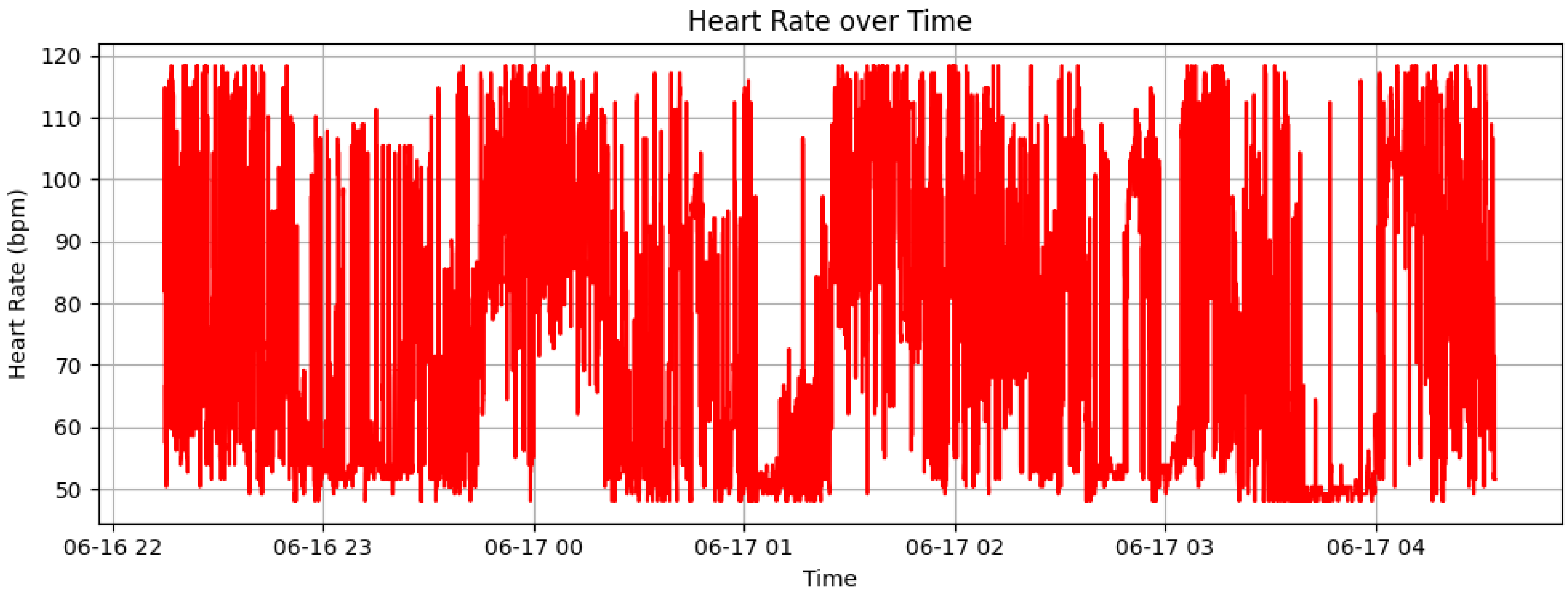
The Heart Rate(HR), Breath Rate(BR) and Temperature(Temp) data had been collected overnight.

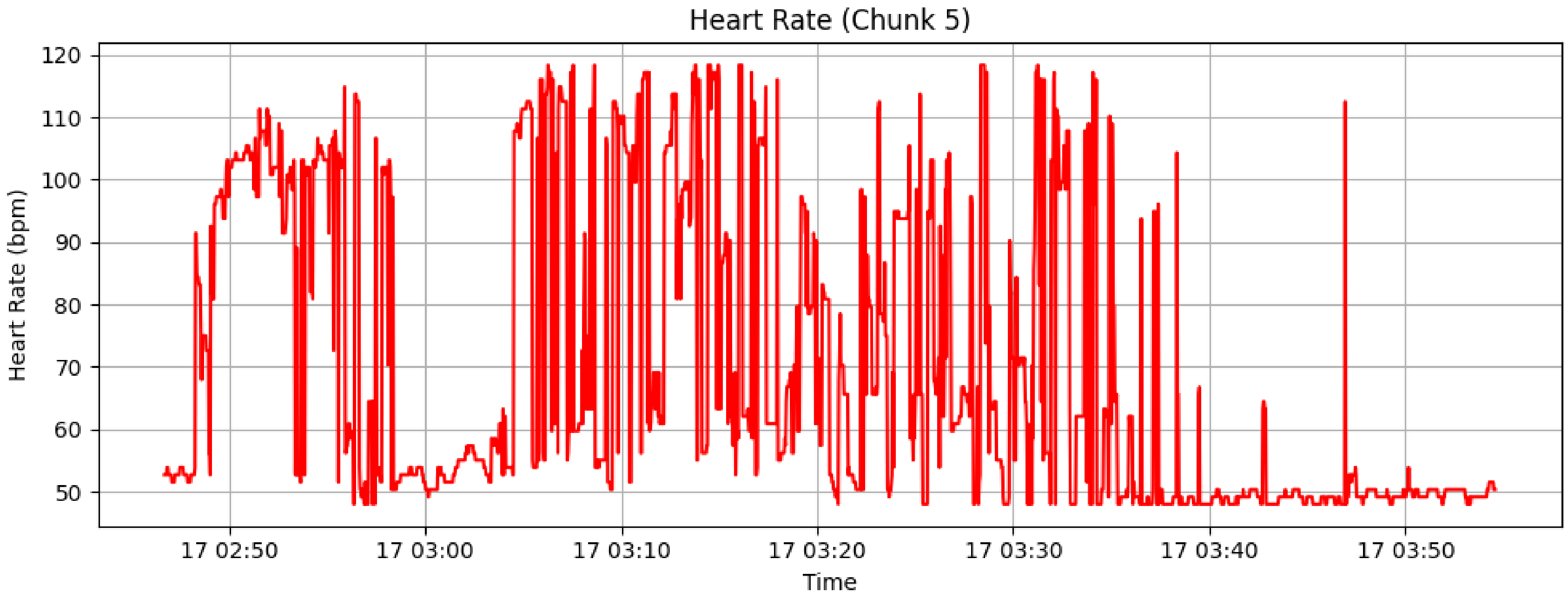
Data collection

Heart Rate overnight.

I) Whole night

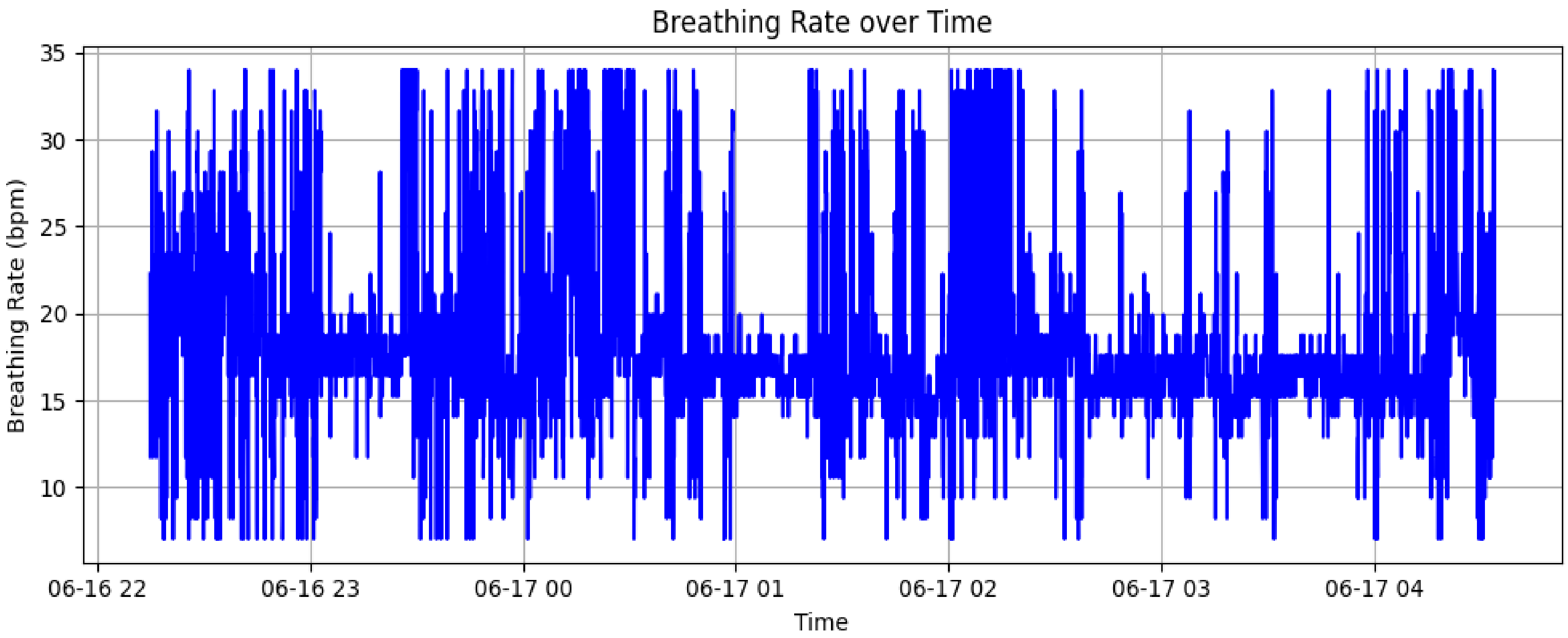
II) Midnight

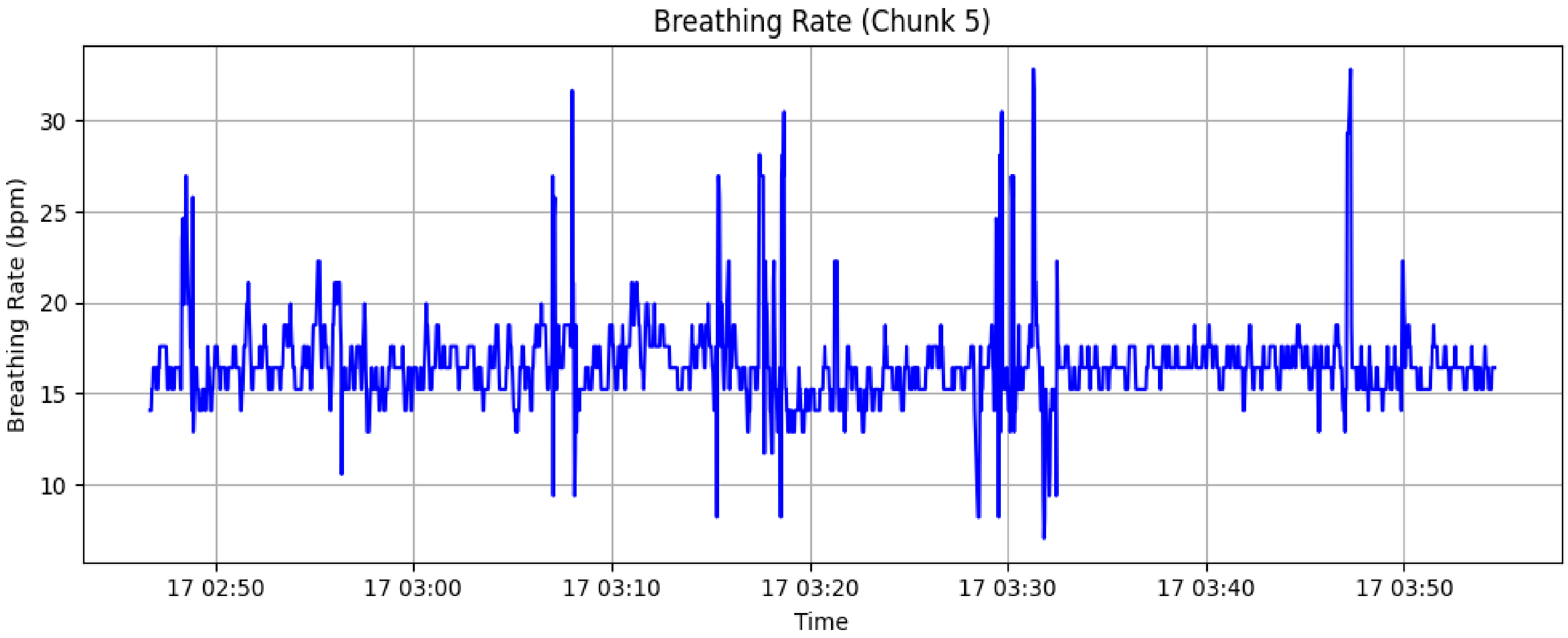




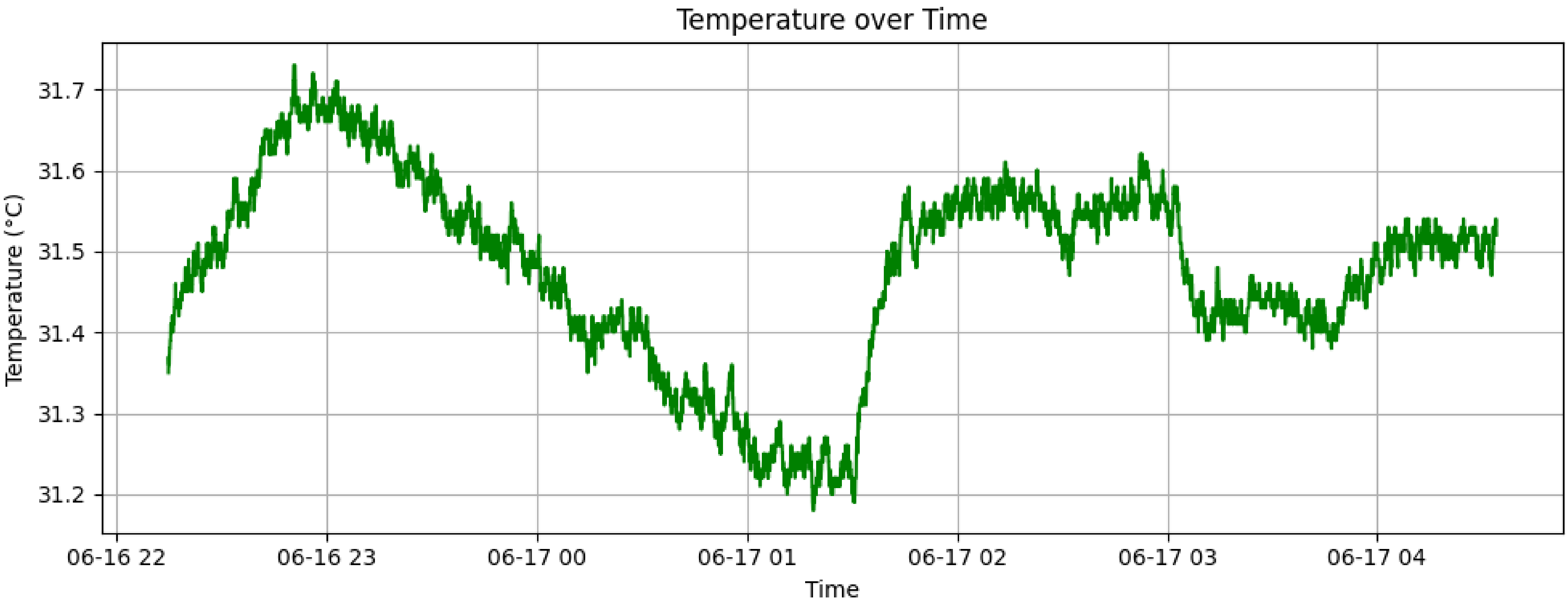
Breath Rate

Breath Rates overnight and at midnight





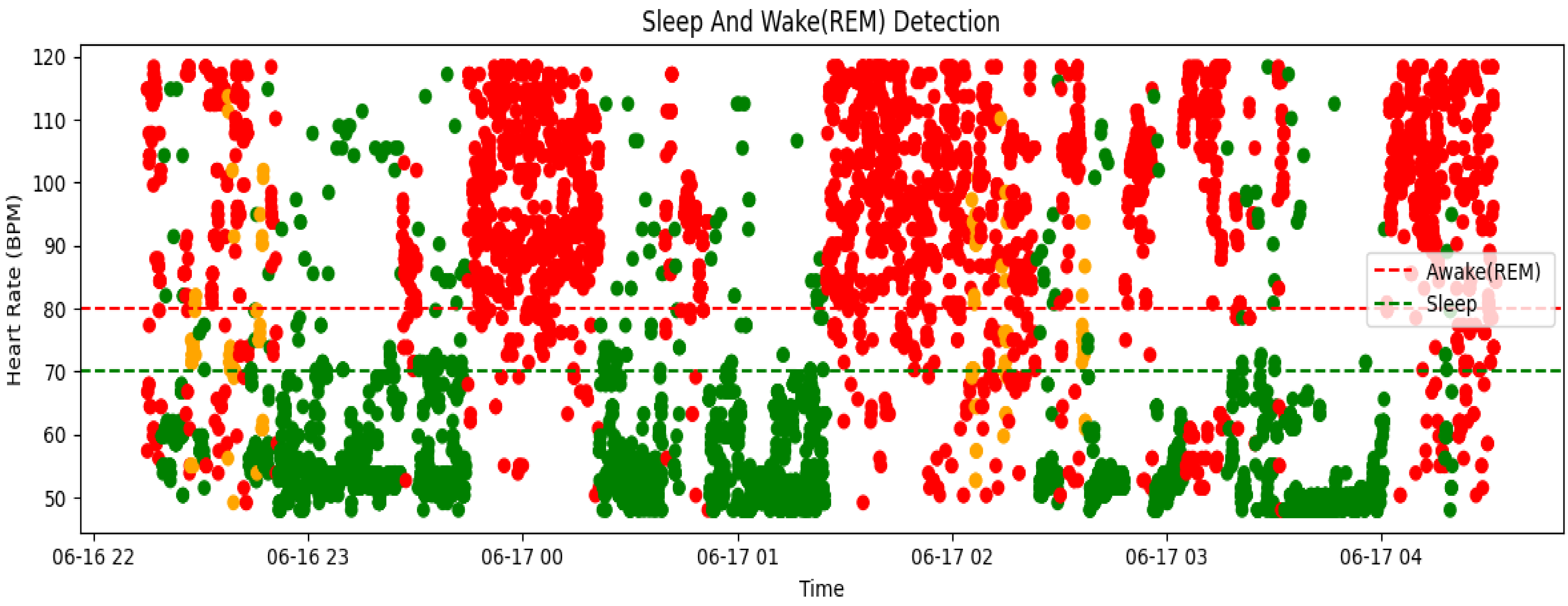
Temperature overnight



🛏️ Stages of Sleep & Heart Rate Patterns

Sleep Stage	Description	Typical Heart Rate Behavior
Awake (Before Sleep)	Fully conscious, may be lying in bed	Elevated or normal (60–100 bpm), varies with activity or stress
N1 (Light Sleep)	Transition from wakefulness to sleep	HR begins to **slow down** gradually
N2 (Light Sleep)	Stable sleep with lowered awareness	HR continues to **drop**, body temp lowers
N3 (Deep Sleep / Slow Wave Sleep)	Restorative stage, physical repair begins	**Lowest HR** (~40–60 bpm), very stable
REM (Rapid Eye Movement)	Dreaming stage, brain active	HR becomes **irregular**, may **increase** slightly

Sleep Stage Detection.



RED – Awake/REM

GREEN – N1/N2/N3

🌙 Light Colour Selection for Better Sleep

- 🟥 **Warm Colors** like **Red, Orange, and Amber** are found to be more effective in promoting sleep.
- Studies show **warm hues** reduce **melatonin suppression** compared to cool or white light.
- These colors **mimic sunset**, signaling the brain that it's time to rest.
- 🔵 In contrast, **cool colors** like **Blue and White** can disrupt sleep cycles by **suppressing melatonin**, the sleep hormone.

🌙 AC Temperature for Sleep

- Cool temperatures help the **body lower core temperature**, aiding **sleep onset**.
- Helps maintain **deep and REM sleep**.
- Too **hot** (>24 °C) → Restlessness, sweating
- Too **cold** (<16 °C) → Wakefulness, shivering
- Recommended option(for India): Set AC to **24 °C + fan** or use **Sleep Mode**

Group	Optimal AC Temperature
Adults	**18–20 °C** (65–68 °F)
Older Adults	**20–22 °C** (68–72 °F)
Infants/Toddlers	**22–24 °C** (72–75 °F)

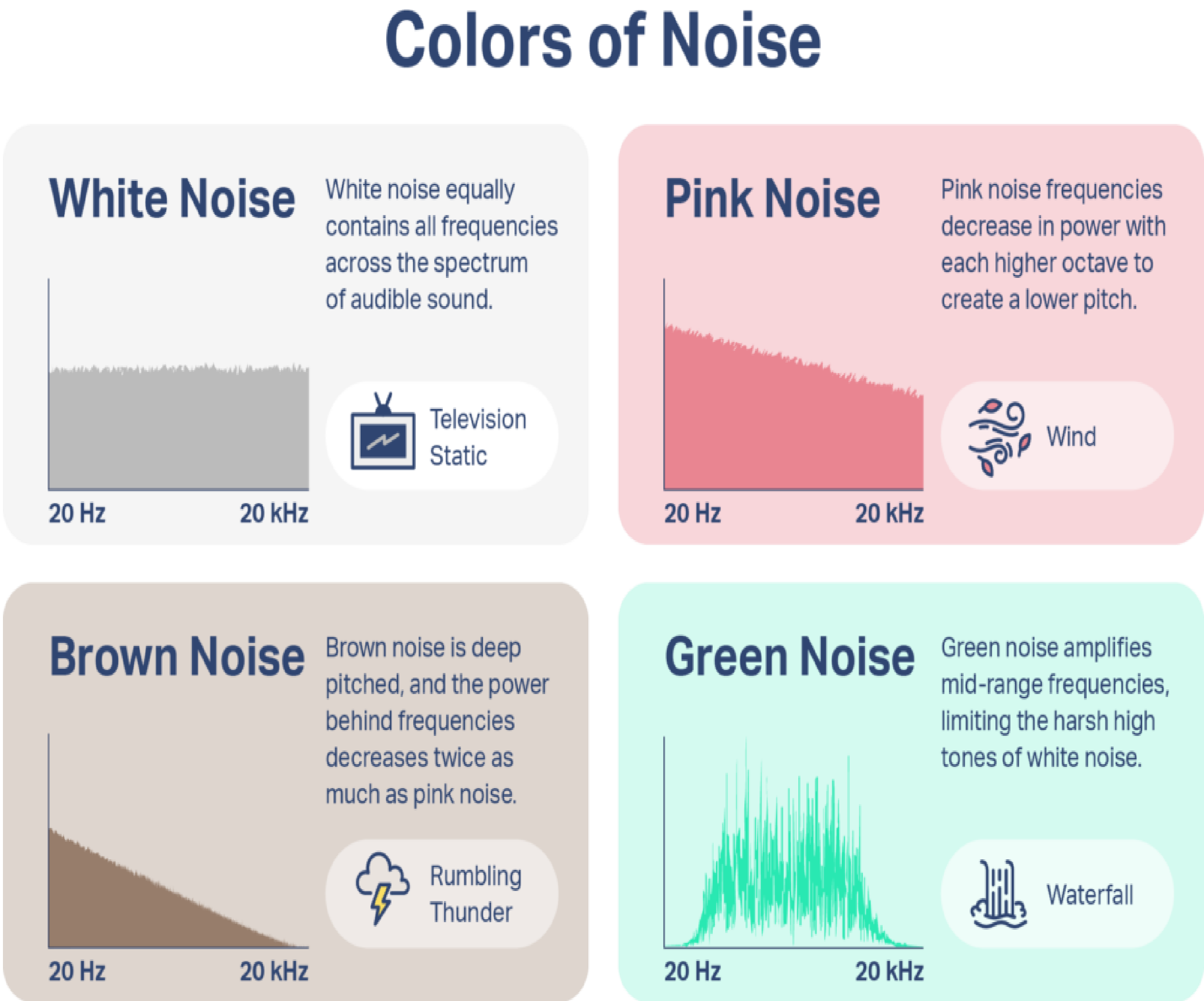
Sleep-Inducing Sounds for Sleep Doc

1. Pink Noise 🌧

- Enhances **deep (N3) sleep** and **memory consolidation**
- Reduces sleep latency (time to fall asleep)
- Mimics natural sounds like rain or wind
- Scientifically shown to **stabilize sleep cycles**

2. Ambient Music (Delta/Theta Range) 🎵

- Reduces anxiety and slows heart rate
- Works even with mild environmental noise
- Supports emotional calm during light sleep phases
- Customizable tones, harmonics, and tempo.



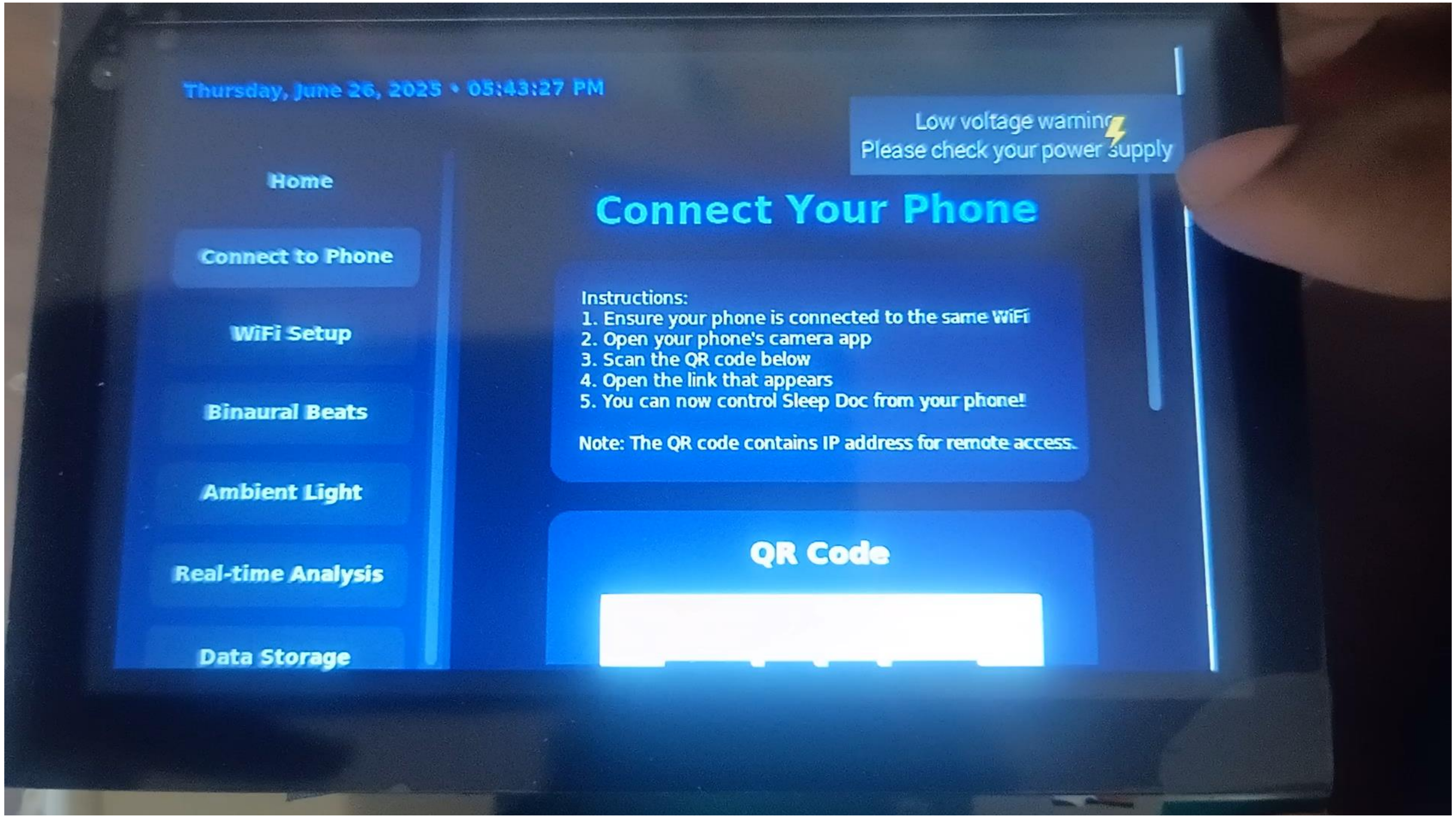
Why Others Rank Lower:

Sound Type	Limitation
White Noise	Harsh, less natural-sounding
Brown Noise	Too low-pitched for some users
Binaural Beats	Needs headphones to work (not speaker-friendly)
Nature Sounds	Less consistent effect without pink noise base

Tkinter GUI

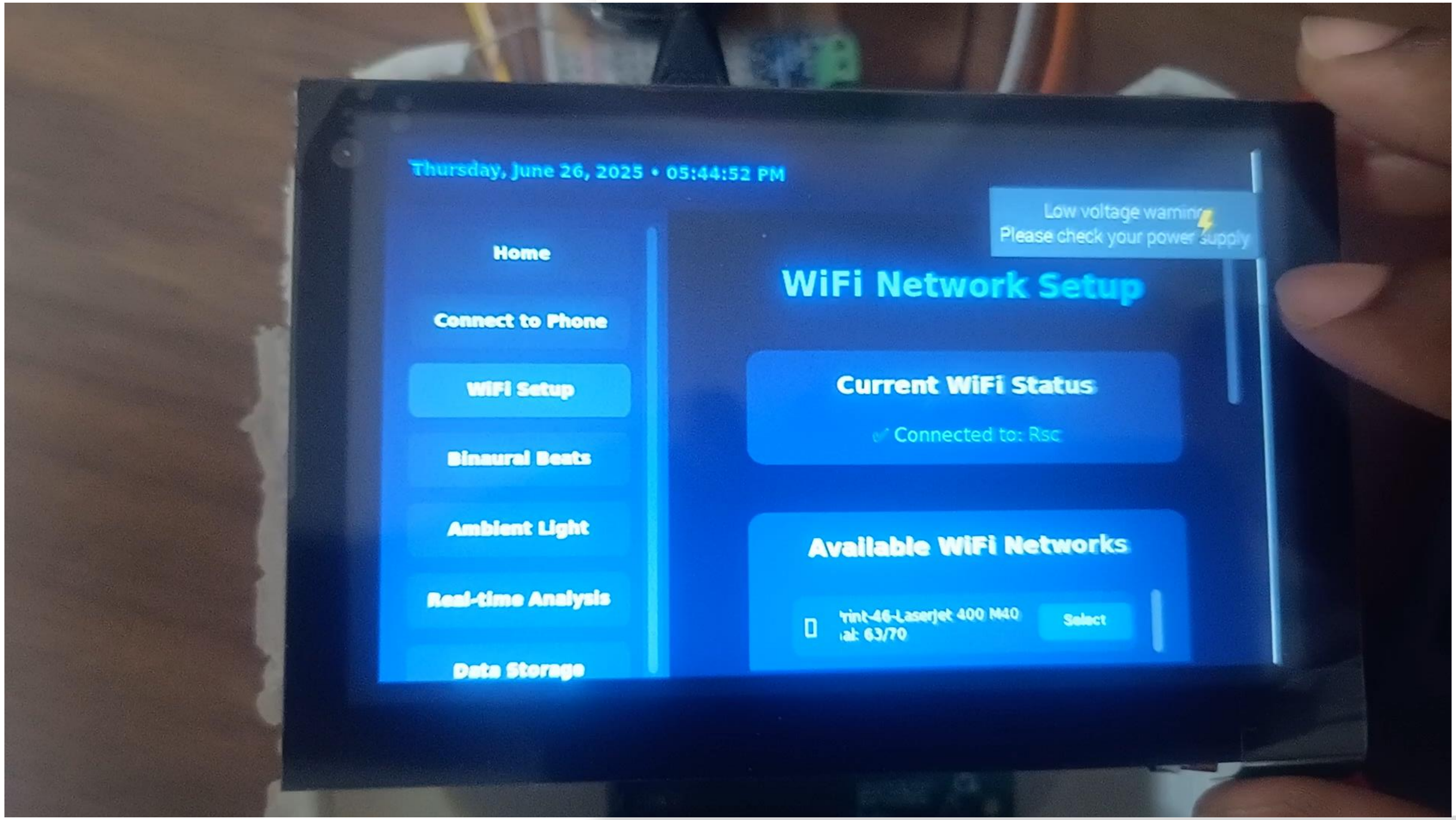
Splash screen

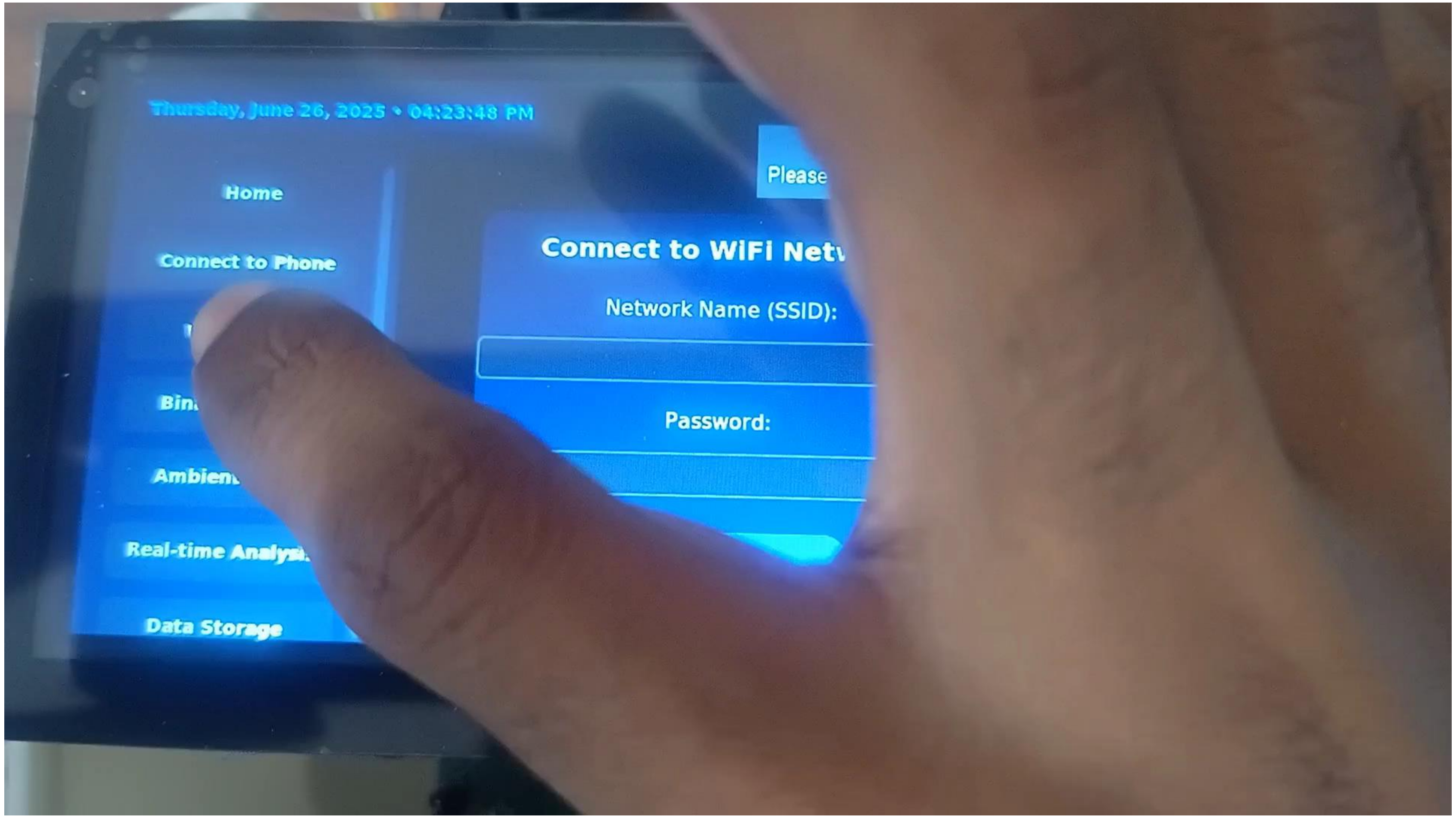
HOME SCREEN



Connect Sleep Doc to Mobile APP via WiFi

WiFi connection setup

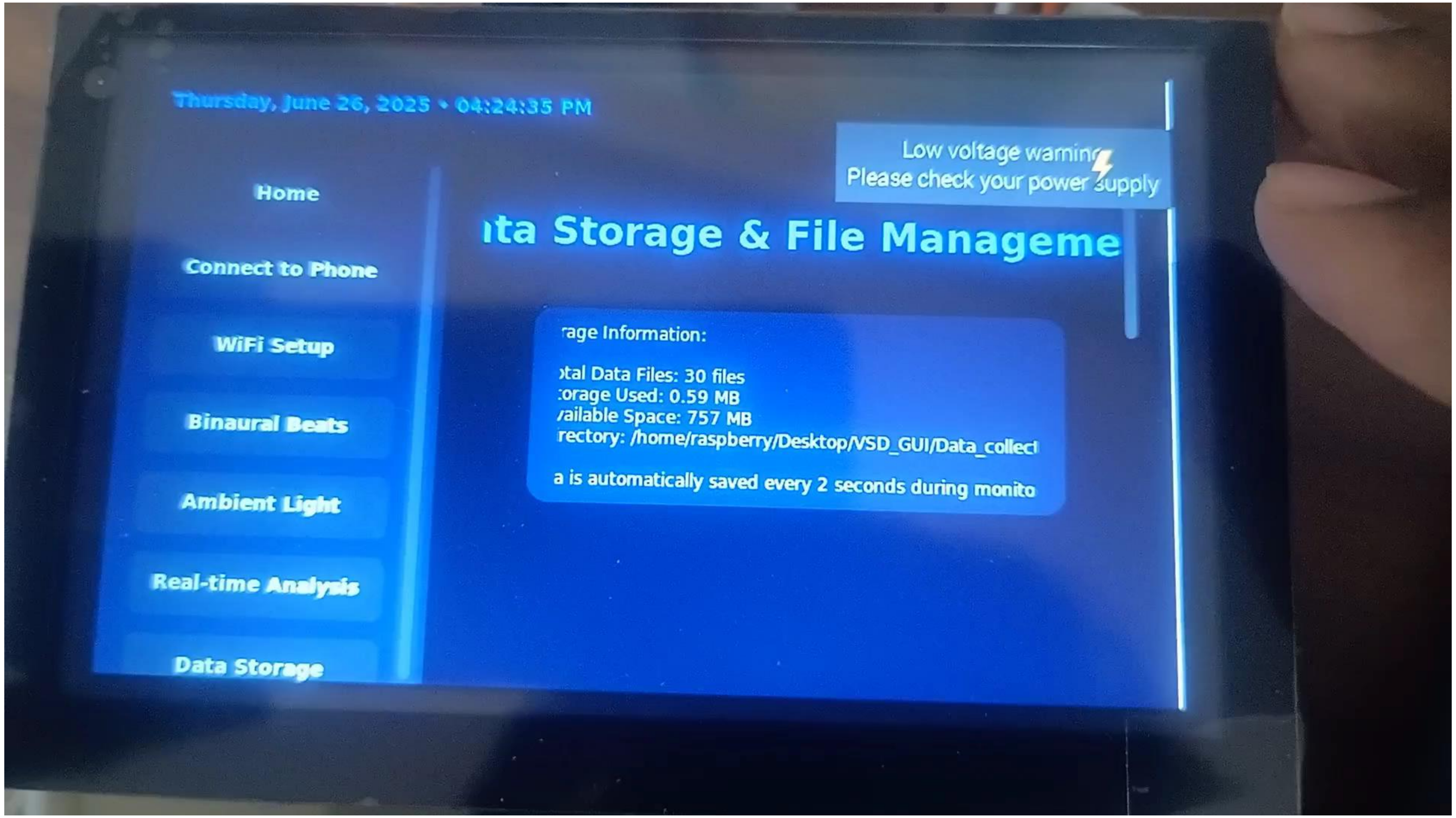




Beats and Audio selection window

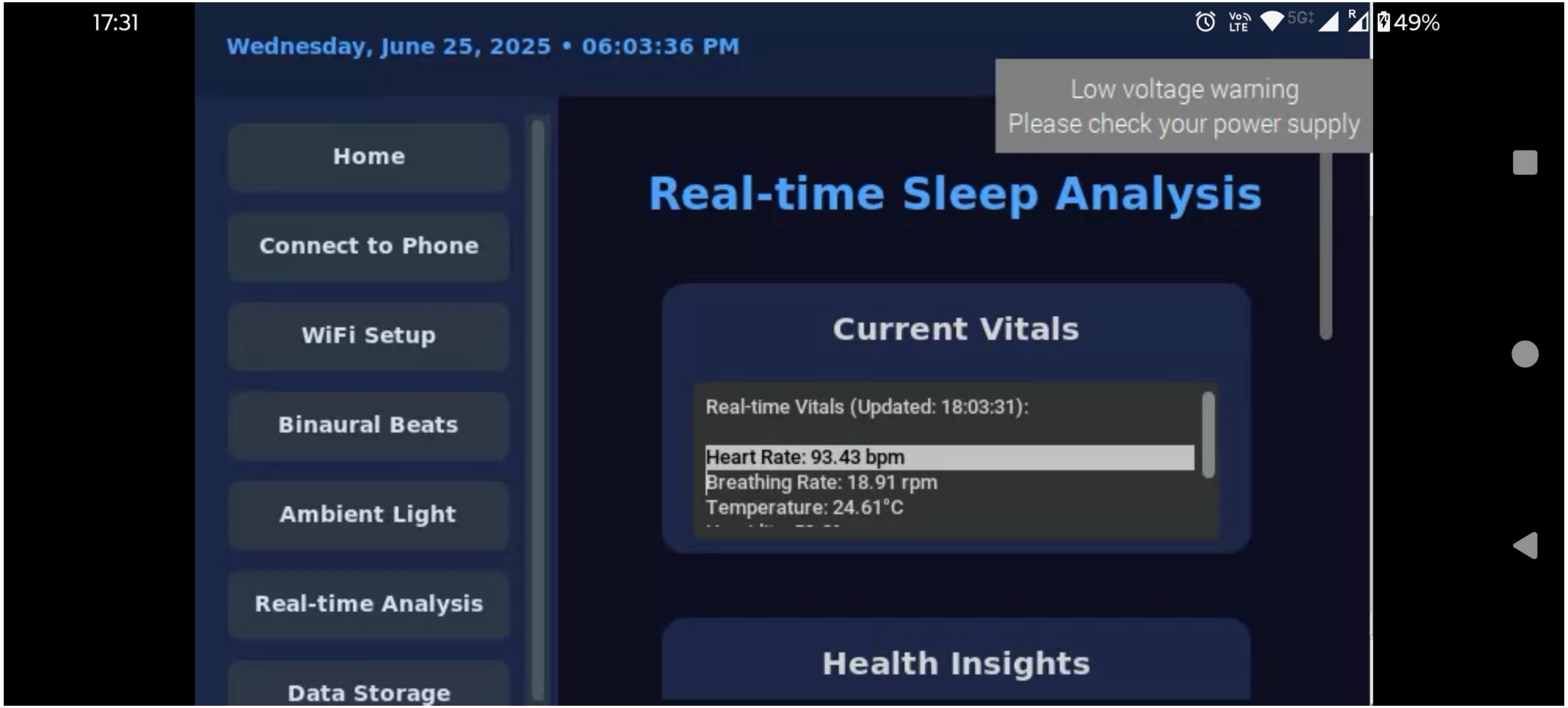
Light selection window

Lighting control from gui



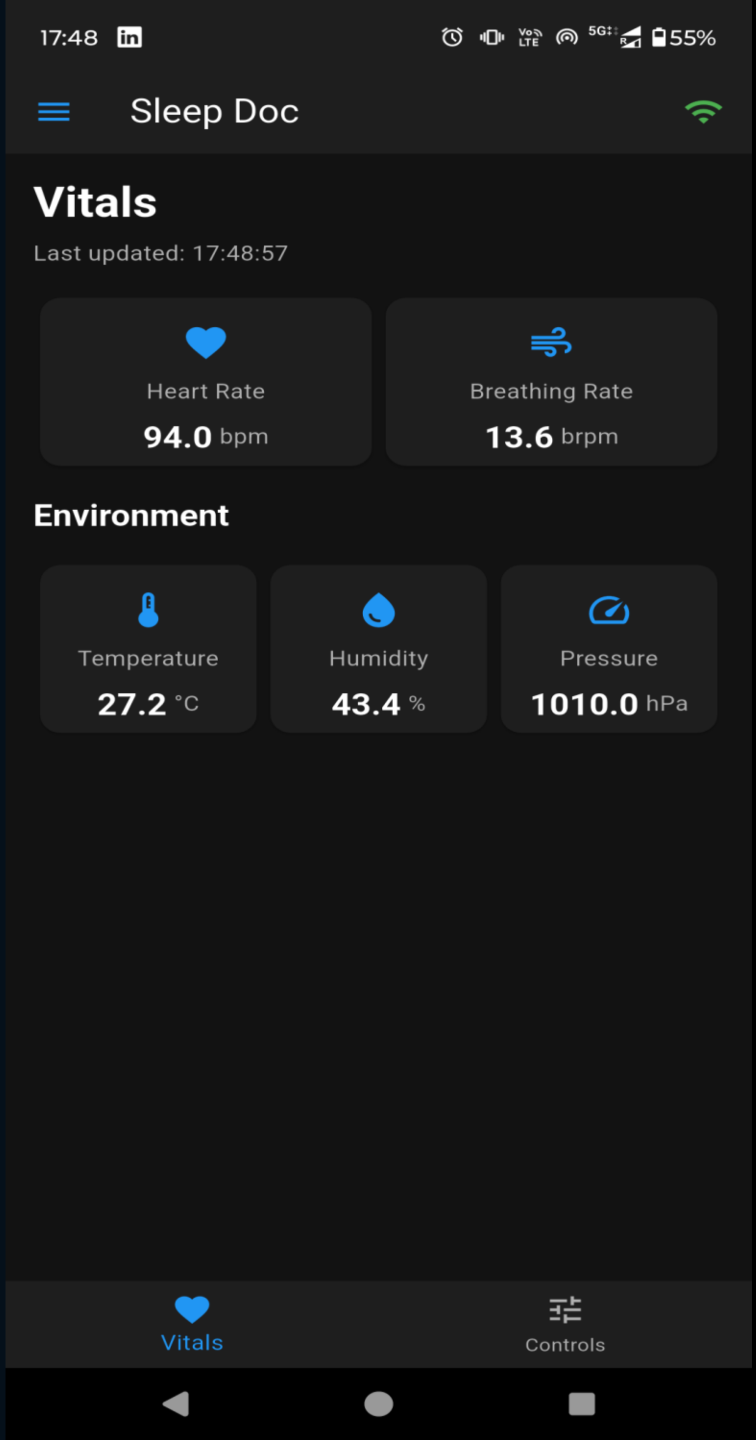
Data Storage and File management

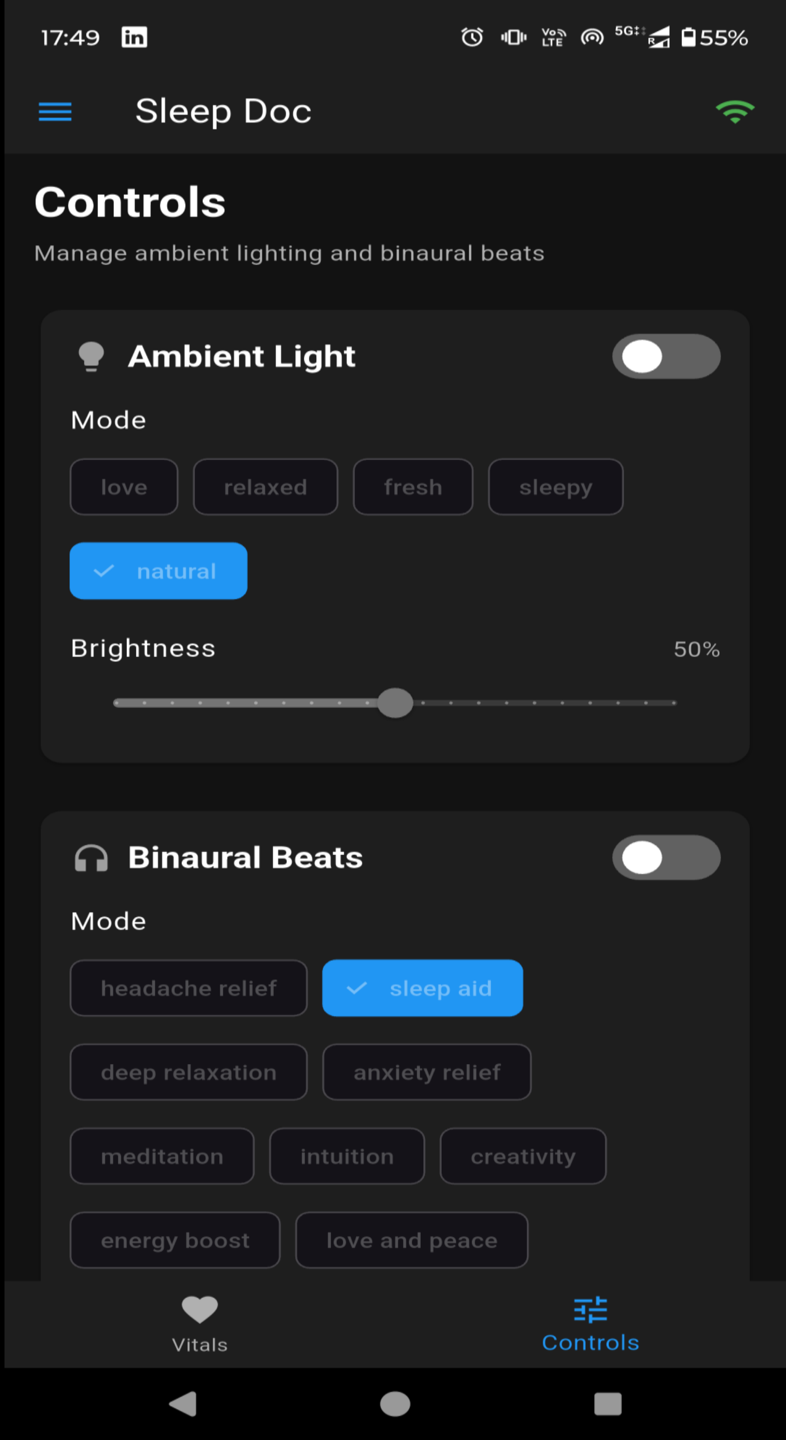
Real Time Analysis window



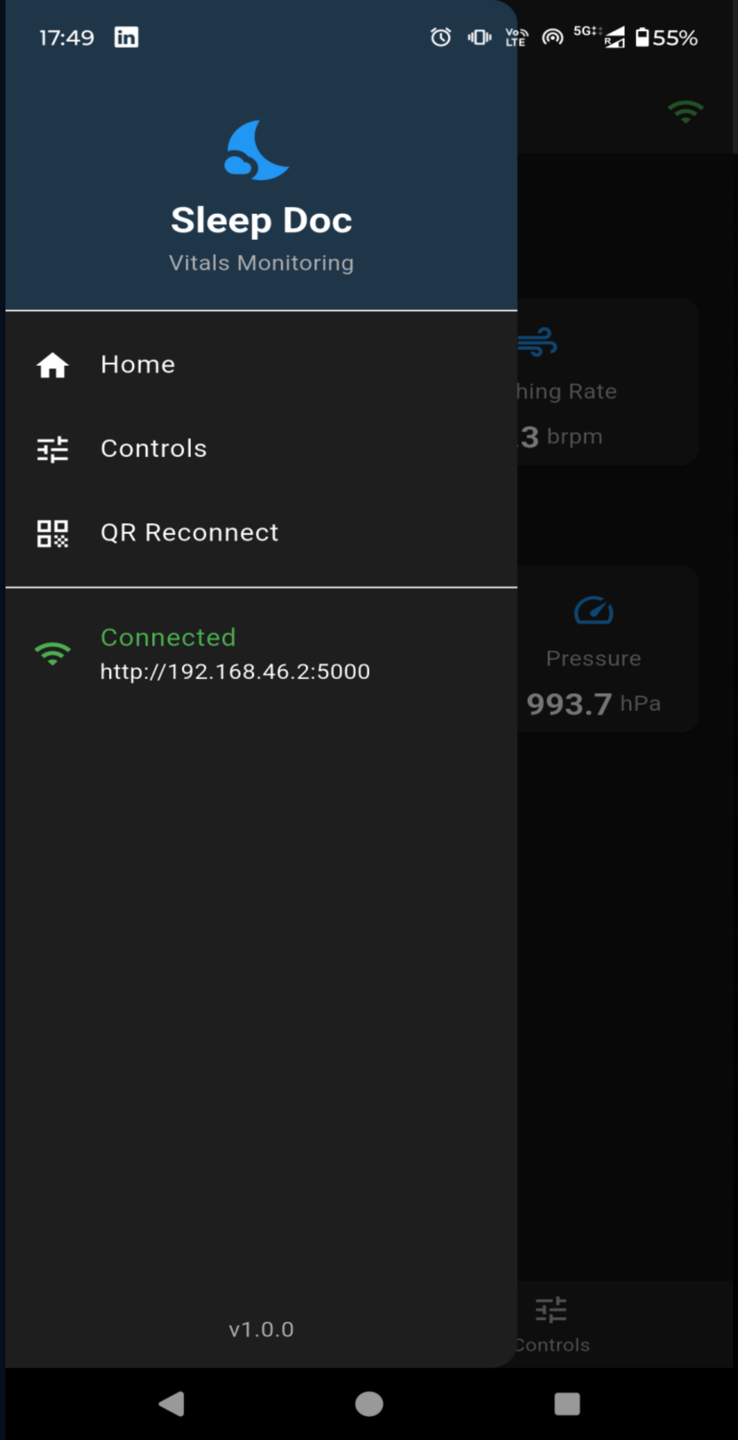
Sleep Doc Mobile APP v1.0.0

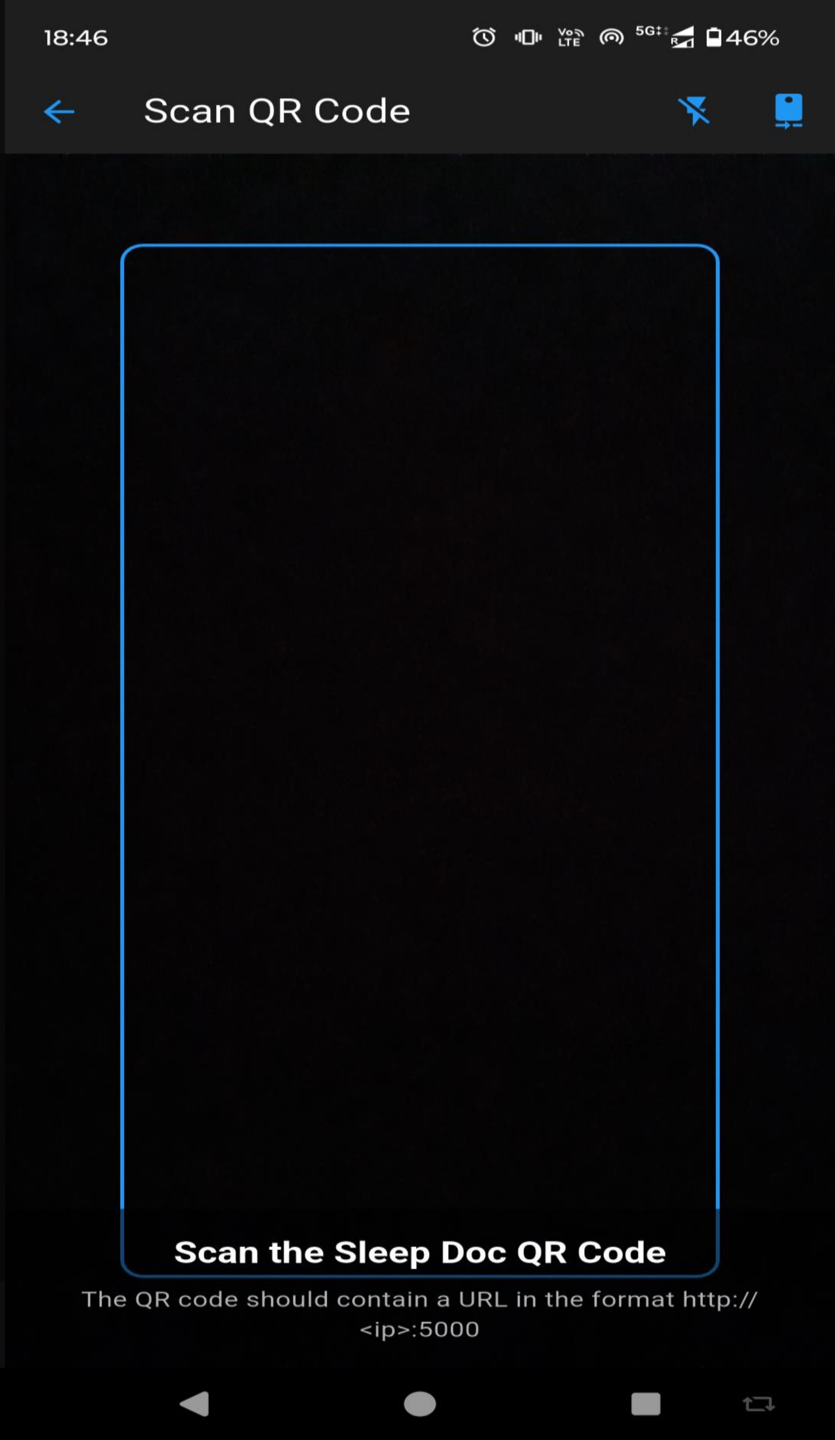
- Sleep Doc intergrated with mobile application.
- APP is developed using Flutter.
- Home screen displays HR, BR, Temp., Humidity, Pressure.
- Light and audio controls from APP.





- Menu sidebar has options Home, Controls, QR Reconnect and connection status.
- The QR code can be scanned from the touch display gui and connect your phone to Sleep Doc via WiFi.





Results & Observations

- Sound and Lighting setting based on user
- Breath & heart rates accurate within ±5 bpm
- Can track people within 1m range with accurate reading
- Full environmental and Vitals analysis of person.
- Touchscreen based GUI and Mobile APP integration

Challenges

RADAR placement for accuracy

Fluctuations in Vitals due to noise.

Detecting Multi-person Vitals

Conclusion and Future Works

Successfully estimated Vitals and detected breathing and heart rates using the BM502 mmWave sensor.

Using MUSIC to estimate Multi-person Vitals

To use Improved MUSIC algorithm for sharper and more accurate peaks for estimation.

Intergration of Application and home appliances control.

References

1. A. Anand, M. K. Mukul, *"Comparative Analysis of different Direction of Arrival Estimation Techniques,"* IEEE, 2015. DOI: 10.1109/IACC.2015.82
2. *"High-Resolution DOA Estimation Techniques: A Survey,"* IEEE, 2021. DOI: 10.1109/ACCESS.2021.3058374
3. I. T. Berrios, *"Introduction to Radar Part 3 – Direction of Arrival (DOA) Estimation,"* Medium, 2020. Read Article
4. Brainard et al., *J Biol Rhythms*, 2001 – spectral effects on melatonin pmc.ncbi.nlm.nih.gov+2pubmed.ncbi.nlm.nih.gov+2en.wikipedia.org+2
5. Figueiro & Rea, *Int J Endocrinol*, 2010 – red vs blue light impacts nypost.com+4pmc.ncbi.nlm.nih.gov+4pmc.ncbi.nlm.nih.gov+4
6. Trinder et al., *Hum Neurophysiol*, 1990 – HR changes across sleep stages pubmed.ncbi.nlm.nih.gov+15pubmed.ncbi.nlm.nih.gov+15arxiv.org+15
7. Cassidy et al., *J Appl Physiol*, 2023 – HR & HRV trends in NREM/REM pmc.ncbi.nlm.nih.gov+2journals.physiology.org+2pubmed.ncbi.nlm.nih.gov+2

THANK YOU