VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI - 590 018, KARNATAKA

A Project Report

On

Stress Detection And Intervention

Submitted in partial fulfilment of the requirements for the degree of

Bachelor of Engineering

In

ELECTRONICS AND COMMUNICATION ENGINEERING

(VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI)

BY

Sheline k v	**USN: 4SO21EC094**
Sneha Paikat	**USN: 4SO21EC106**
Trisha Steeven S	**USN: 4SO21EC114**
Veeksha K V	**USN: 4SO21EC119**

Under the Guidance of

Ms. Preetha D'Souza, B.E, M.Tech, M.I.S.T.E

Assistant Professsor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

(UG PROGRAMME ACCREDITED BY NATIONAL BOARD OF ACCREDITATION, NEW DELHI)

ST JOSEPH ENGINEERING COLLEGE

AN AUTONOMOUS INSTITUTION

Accredited By NAAC with A+ Grade

Vamanjoor, Mangaluru - 575028, India

May 2025

ST JOSEPH ENGINEERING COLLEGE

(Affiliated to Visvesvaraya Technological University, Belagavi)

Vamanjoor, Mangaluru – 575028

Department of Electronics and Communication Engineering

(UG Programme Accredited By National Board of Accreditation, New Delhi)

CERTIFICATE

Certified that the project work entitled **"Stress Detection and Intervention"** is carried out by

Sheline k v	**USN: 4SO21EC094**
Sneha Paikat	**USN: 4SO21EC106**
Trisha Steeven S	**USN: 4SO21EC114**
Veeksha K V	**USN: 4SO21EC119**

bonafide students of VIII semester, B.E, Electronics and Communication Engineering Department of St Joseph Engineering College, Vamanjoor, Mangaluru-575028, in partial fulfilment for the award of Bachelor of Engineering in Electronics and Communication Engineering of the Visvesvaraya Technological University, Belagavi during the year 2024-25. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report and deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

17/5/2025	17/5/25	
Ms Preetha D'Souza	**Dr Dayakshini**	**Dr Rio D'Souza**
Project Guide	**Head of Department**	**Principal**
Dept. of E&C Engineering	**Dept. of E&C Engineering**	

EXTERNAL VIVA VOCE EXAMINATION

1. **Name:** Dr Dayakshini
   Designation: Professor & HoD
   27/5/25
   Signature with date
2. **Name:** Dr. Jaylin
   Designation: Assoc Prof
   27/5
   Signature with date

ACKNOWLEDGEMENT

We express our deep gratitude to Rev. Fr Wilfred Prakash D'Souza, Director and Rev. Fr Kenneth Rayner Crasta, Assistant Director for providing all the facilities for carrying out our project work.

We are indebted to our Principal, Dr Rio D'Souza and the Management of St Joseph Engineering College, Vamanjoor, Mangaluru for having provided all the facilities that helped us in timely completion of the project.

We are grateful to Dr Dayakshini, Head of the Department, Electronics and Communication Engineering and the Project Co-ordinators Ms Preetha D Souza and Mr Glenson Toney for their support and encouragement.

We would like to offer our earnest gratitude to our project guide, Ms Preetha D Souza, Assistant Professor, Department of Electronics & Communication Engineering, SJEC, Mangaluru. This work would not have been possible without the encouragement and able guidance of her.

We wish to express our sincere gratitude to all the Faculty and Technical Staff of the Department of Electronics & Communication Engineering, SJEC Mangaluru, for their valuable guidance, help and support.

Sheline k v

Sneha Paikat

Trisha Steeven S

Veeksha K V

ABSTRACT

This project presents a real-time stress detection and management system that aims to promote mental wellness.

Existing research in stress detection often relies on isolated physiological signals or facial analysis. However, the combination of both has shown potential for more reliable results.

Our system integrates facial emotion detection using Convolutional Neural Networks (CNNs) with Electrodermal Activity (EDA) sensors, interfaced through an ESP32. The data is sent to Firebase and visualized via a React-based web application offering remedies like music and relaxation exercises.

The results confirm improved stress identification through multimodal input.

Future work includes wearable integration and expanded therapeutic interventions.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Sl.No	Abbreviations	Keywords
1	GSR	Galvanic Skin Response
2	ML	Machine Learning
3	KNN	K Nearest Neighbours
4	IOT	Internet Of Things
5	EDA	Electrodermal Activity
6	CNN	Convolutional Neural Network
7	FCN	Fully Convolutional Network
8	SVM	Support vector Machine
9	RF	Random Forest

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

CHAPTER - 1

INTRODUCTION

Stress is now a common occurrence in daily life. Understanding and managing it is more important than ever to preserve your happiness and well-being. Long-term stress can lead to serious problems like heart disease, low energy, and anxiety. People can manage stress before it has a detrimental effect on their health and well-being when they recognize it early.

There are several methods to detect stress, including tracking physiological changes like heart rate and skin reactivity or analyzing feelings and behaviors. Machine learning has opened up new avenues for stress identification by analyzing facial expressions and body language.

This system assesses emotions from both live and previously captured facial pictures using algorithms such as Random Forest, K-Nearest Neighbors, Support Vector Machines, and Convolutional Neural Networks. By examining facial expression patterns to identify whether a person is worried, these models offer accurate insights into their emotional state.

To further improve its reliability, the system also incorporates data from skin conductance, electrodermal activity, and heart rate. EDA measures the amount of perspiration that occurs when a person's skin reacts to stress. An ESP32 microcontroller with built-in sensors is used to collect this data in real-time and combine it with the emotion-based results. These inputs work together to create a clearer and more accurate picture of the stress level.later Create an app for individualized stress management by combining emotions and GSR data to identify stress levels.

This approach employs machine learning to provide users with a simple and efficient method of identifying and managing stress by combining insights from physiological signals and facial emotions. It gives them the tools to live a more balanced and healthful life by assisting them in understanding their stress levels and pointing them in the direction of easy ways to feel better.

1.1 Physiological Monitoring:

Stress immediately alters physiological signals, and these changes are tracked using a variety of techniques. Typical techniques include:

- Skin Conductance (also known as Galvanic Skin Response, or GSR): Stress can alter skin conductance by increasing the activity of sweat glands. This method is frequently used to measure arousal levels.

- Heart Rate Monitoring: An elevated heart rate is frequently the result of stress. Photoplethysmography (PPG) and electrocardiography (ECG) are used by devices like wearable fitness trackers and chest straps to measure changes in heart rate and heart rate variability (HRV), which are important markers of stress levels.
- Monitoring Respiratory Rate: Stress can change how people breathe, usually making them shallower and faster. To track these changes, monitoring devices employ wearable patches, spirometers, or chest straps.
- Blood Pressure: Stress frequently results in brief increases in blood pressure. Stress-related changes can be detected with the use of wearable cuffs or cuffless devices that use pulse transit time (PTT).
- Skin Temperature: Vasoconstriction during stress can cause the peripheral skin temperature to drop. Wearable temperature sensors or infrared thermometers can track these variations.
- Pupil Dilatation: When the autonomic nervous system is activated in response to stress, the size of the pupil may change. These changes can be detected by specialized cameras or eye-tracking devices.
- Muscle Activity (Electromyography, EMG): EMG sensors applied to particular muscle groups, like the forehead or shoulders, can measure muscle tension, which is exacerbated by stress.

- Electroencephalography (EEG) of the Brain Activity: Stress can change the patterns of brainwaves. These changes are measured by EEG devices, which offer a comprehensive understanding of emotional and cognitive reactions.
- Salivary Cortisol: Salivary cortisol levels can be obtained non-invasively and analyzed as a trustworthy biomarker of stress.

CHAPTER 2
LITERATURE REVIEW

CHAPTER - 2

LITERATURE REVIEW

Table 2.1 An overview of the literature survey to identify the gaps

Sl.No.	Paper Title	Author	Year	Key Points
1	UBFC-Phys:AMultimodal Database For Psychophysiological Studies of Social Stress ,IEEE	Rita Meziati Sabour, Yannick Benezeth, PierreDe Oliveira, Julien Chappé, and Fan Yang.	2021	Deals with social stress, along with physiological signals like heart rate, GSR, and EEG. It's designed to help develop machine learning models for detecting stress and recognizing emotions.
2	A Review on Mental Stress Detection Using Wearable Sensors and Machine Learning Techniques. IEEE.	Bakker J, Pechenizkiy, M,Sidorova, N.	2021	Wearable sensors and machine learning for stress detection focus on heart rate, skin conductance, and brain activity, tackling dataset and feature extraction challenges.
3	Portable and wearable real-time stress monitoring: A critical review," Sensors and Actuators Reports,	O. Parlak	2021	Portable and wearable devices for real-time stress monitoring focus on sensors and techniques for

				continuous stress detection.
4	A New intelligent Approach for Automatic Stress Level Assessment Based on Multiple Physiological Parameters Monitoring	G. Ribeiro, O. Postolache, and F. F. Martín	2024	The system uses ESP32 microcontrollers to gather data from PPG, GSR, and temperature sensors, estimating key health metrics like heart rate, heart rate variability, respiratory rate, and SpO2. This data can be stored either locally or remotely on Firebase using Wi-Fi.
5	Emotion and Stress and Utilizing Galvanic Skin Response and Wearable Technology: A Real-time Approach for Mental Health Care,	Hosseini, R. Fang,R.Zhang, S.Rafatirad, andH.Homayoun,	2023	Develops a real-time emotion recognition system using GSR signals, processing and classifying emotions with machine learning.
6	A Review on Mental Stress Detection Using Wearable Sensors and Machine Learning Techniques	S. Gedam ,S Paul	2021	Wearable sensors (ECG, EEG, GSR) combined with machine learning, sensor fusion, and deep learning enable real-time stress detection with

				improved accuracy across various environments.
7	Review of Stress Detection Methods using wearable sensors	G.Taskas Applidis, D. Fotiadis, and P. D. Bamidis		Focusing on wearable technologies and biological signals to assess stress through autonomic nervous system and HPA axis responses**.**
8	Real-Time Stress Detection and analysis using facial emotion recognition	H P Chandika ,B Soumya ,B N E Reddy and B M S S Manideep	2024	Real-time stress detection is achieved through facial emotion recognition, analyzing facial expressions as non-invasive indicators for assessing psychological states.
9	Stress Detection with Machine Learning and Deep Learning using Multimodal Physiological Data	P Bobade ,M Vani	2020	Combines machine learning and deep learning to detect stress using multimodal physiological data (ECG, GSR, respiration rate), improving accuracy through data integration.
10	Extraction of facial features as indicators of stress and anxiety	M Pediaditis et al	2015	A method to assess stress and anxiety levels by analyzing facial expressions

				through computer vision, using facial cues as non-invasive indicators of psychological states.
11	Using Deep Convolutional Neural Network for Emotion Detection on a Physiological Signals Dataset	L.Santamaria-Granados, M. Munoz-Orgaero,G.Ramirez-González, E. Abdulhay, and N. Arunkumar	2019	Deep learning to detect emotions from body signals like ECG and skin response. It uses the AMIGOS dataset and applies a CNN model to predict how a person feels. The results show better accuracy than older methods, making it useful for personalized apps.

CHAPTER 3
METHODOLOGY

CHAPTER - 3

METHODOLOGY

3.1 Software Methodology

A facial emotion recognition model was developed using the CK+ dataset, which contained a total of 989 images—735 for training and 246 for testing. These images were used to detect seven different emotions: happiness, fear, sadness, contempt, anger, surprise, and disgust.

Multiple algorithms were evaluated for emotion detection, including Convolutional Neural Networks (CNN), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest. Among these, CNN demonstrated the highest accuracy due to its ability to extract both local and global features through convolutional, pooling, and dense layers.

The model was implemented in Google Colab using TensorFlow and Keras. To ensure optimal performance, the dataset was pre-processed before training. Grayscale conversion was applied to simplify the images by removing color information. All images were resized to a standard dimension of 48x48 pixels to maintain uniformity. Normalization was performed to scale pixel values to the range [0, 1]. Label mapping was conducted to group the emotions into broader categories such as positive, negative, and neutral. The dataset was then split into training and testing sets in a 75:25 ratio.

Overfitting and underfitting were examined for each algorithm to ensure the generalization capability of the models and avoid performance degradation on unseen data.

3.1.1 Algorithms

3.1.1.1 Convolutional Neural Network (CNN)

Images were loaded from subfolders, each representing a specific emotion. After undergoing preprocessing, the images were label-encoded into categories: positive, negative, and neutral.

The CNN architecture consisted of an input layer that accepted 48x48 grayscale images, followed by four convolutional layers with increasing filter sizes (32, 64, 128, 156) to extract facial features. ReLU activation was applied to introduce non-linearity and enable the recognition of complex patterns. Pooling layers were used to reduce spatial dimensions. The 2D feature maps were then flattened into a 1D vector and passed through dense layers. Dropout regularization was applied to prevent overfitting.

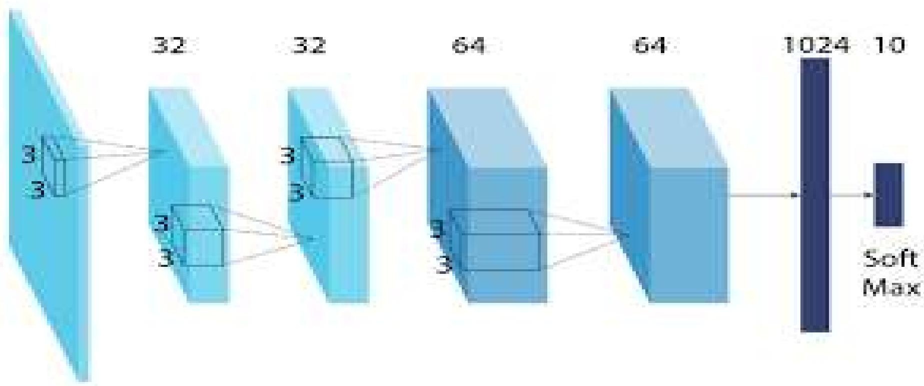


Fig 3.1 CNN model Architecture with four layers

Overall the accuracy of CNN was 0.99.

3.1.1.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) maps input data into a higher-dimensional space where classes can be linearly separated. Images were loaded from subfolders based on their respective emotions and underwent preprocessing, including grayscale conversion, resizing to 48x48, and normalization. Label mapping was done to categorize emotions as positive, negative, or neutral.

The images were reshaped into 1D arrays and label-encoded into numeric values. A 75:25 train-test split was applied. SVM, primarily used for classification, employed the Radial Basis Function (RBF) kernel to create non-linear decision boundaries by measuring data point similarity.

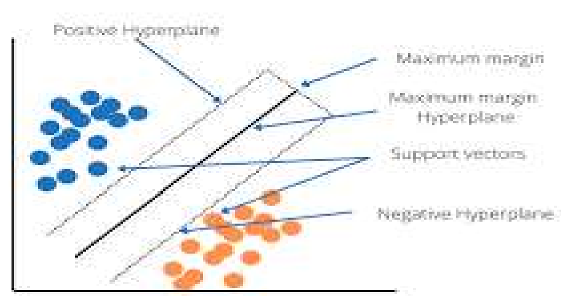


Fig 3.2 SVM model

Key hyperparameters included C, which controls the trade-off between margin size and classification error, and gamma, which determines the influence of individual data points. After prediction, numeric labels were decoded back to their corresponding emotion categories for evaluation.

The SVM model showed an accuracy of 0.94. It performed but it slightly lagged compared to CNN.

To evaluate the performance of the stress detection model, standard classification metrics such as Accuracy, Recall, and F1 Score are used.. The formulas are given below:

A. Recall: How many of the actual labels are correctly predicted.

$$Recall = \frac{True\ Positives(TP)}{False\ Negatives(FN) + True\ Positives(TP)} \quad (1)$$

B. F1 score: A balance between precision and recall.

$$F1 - Score = 2\frac{Precision \text{ x } Recall}{Precision + Recall} \quad (2)$$

C. Accuracy: The overall percentage of correct predictions

$$Accuracy = \frac{True\ Positives(TP) + True\ Negatives(TN)}{Total\ Samples} \quad (3)$$

D. Precision: How many of the predicted labels are correct.

$$Precision = \frac{True\ Positives(TP)}{False\ Positives(FP) + True\ Positives(TP)} \quad (4)$$

3.1.1.3 Random Forest

Random Forest is an ensemble learning algorithm that combines predictions from multiple decision trees. Each tree is trained on a randomly sampled subset containing approximately 63.2% of the original data. At each node split, a random subset of features is selected. If *M* is the total number of features, then only a subset of *M* is used for each split.

$$m = \sqrt{M} (for\ classification\ tasks) \quad (5)$$

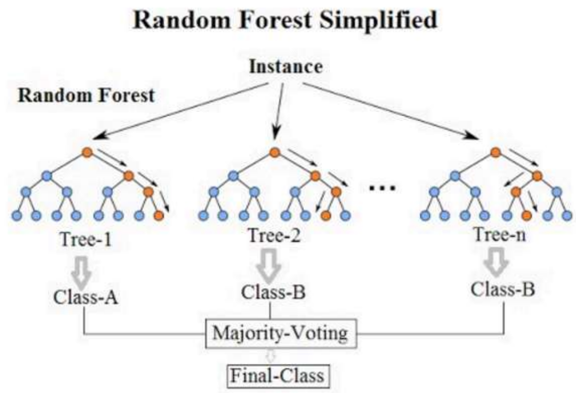


Figure 3.3 Random Forest model

After feature extraction, each decision tree recursively partitions the data by selecting splits that maximize the reduction in impurity. Gini Impurity is used as the splitting criterion and is defined as:

$$Gini(t) = 1 - \sum_{i-1}^{c} p_i^2 \tag{6}$$

where *pi* is the proportion of samples belonging to class *i* at node *t*, and *C* is the number of classes.

For final classification, a majority voting mechanism is applied across all trees:

$$\hat{y} = arg\ max_c \sum_{j=1}^{T} I(y_j = c) \tag{7}$$

Random forest gave an accuracy of 0.99. Random forest and CNN are showing good accuracy but extracting facial features is done more precisely by CNN .

3.1.1.4 K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) classifies images based on their similarity to nearby samples in the training data using Euclidean distance. Preprocessing steps such as grayscale conversion, resizing, normalization, and label encoding were applied. Images were flattened into 1D arrays, and the dataset was split into training and testing sets in a 75:25 ratio. Label encoding was consistent across all algorithms: Positive = 0, Negative = 1, Neutral = 2.

The model stores data points and, during prediction, compares new samples with stored ones. Here, $k = 3$ was chosen, meaning the prediction was based on the three nearest neighbors. The distance between data points was calculated using the Euclidean distance formula:

$$Distance(Xi, Xj) = \sqrt{\sum_{k=1}^{n} (X_{ik} - X_{jk})^2} \quad (8)$$

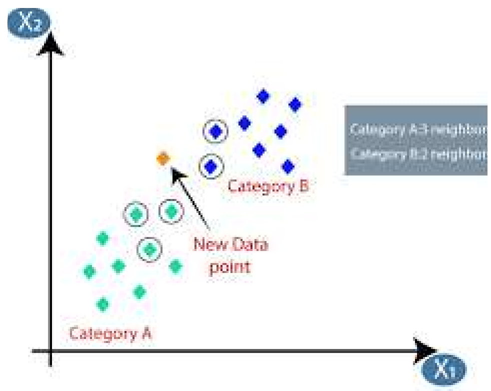


Figure 3.4 KNN model

KNN achieved an overall accuracy of 0.89, but showed lower recall values, indicating a limitation in correctly identifying all true positives.

3.1.2 Overall

Among all evaluated algorithms—CNN, SVM, Random Forest, and KNN—CNN outperformed the others due to its ability to accurately extract facial features through its layered architecture. This makes CNN the most suitable model for facial emotion recognition. Using this CNN model, emotions can be identified effectively, aiding in stress level assessment based on detected emotional states.

3.2 Hardware And Software Implementation

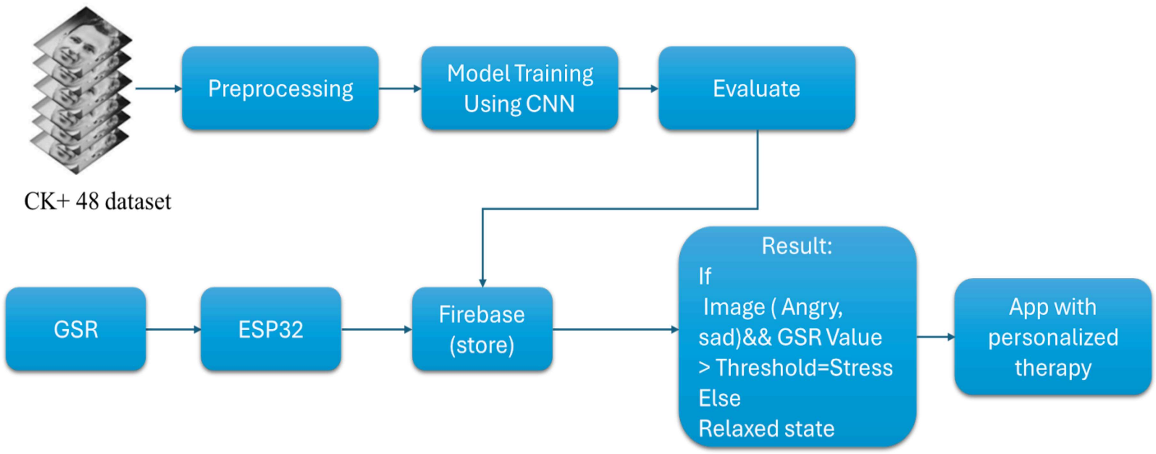


Figure 3.5: Block diagram

3.2.1 Emotion Recognition Using Facial Features[Software]

3.2.1.1 Environment Setup

- Install Necessary Libraries
  - To set up the Python environment:
  - Installed the required libraries using pip
- Set Up VS Code
  - Open Visual Studio Code (VS Code).
  - Created a new Python file
  - Provided code into the file.
  - Provided the Python interpreter in VS Code to the version where the required libraries are installed.
- Hardware Requirements
  - The system has a functional webcam for live capture.
  - Tested the webcam.

3.2.1.2 Image Upload and Stress Classification

- Run the Script
  - Execute the script in the VS Code terminal
- Upload Image
  - When prompted, a file dialog will appear to let you upload an image.

- Selected an image containing a face for emotion analysis.
- Emotion Classification
    - The script:
        - Detects faces using the Haar cascade classifier.
        - Analyzes emotions using DeepFace.
        - Maps emotions:
            - ❖ angry, sad→ classified as stress.
            - ❖ happy, surprise → classified as themselves.
            - ❖ fear, disgust → classified as themselves.
- Result Visualization
    - The processed image is displayed in a new window showing:
        - A rectangle around the detected face.
        - A label with the predicted emotion or stress classification.

3.2.1.3 Stress Detection Using Webcam

- Modified the Code for Live Capture
    - Added a function for webcam input
- Run Live Detection
    - Replaced the upload_image() call with live_capture_and_predict() in the main() function for detection.
    - Executed the script to display the live webcam feed.

3.2.1.4 Combining and Testing

- Switch Between Modes
    - Added an option for users to choose between image upload or live capture

- Run the Updated Script
    - Execute the script and choose the desired mode:
        - Option 1: Upload an image for analysis.
        - Option 2: Use live capture for detection.

3.2.1.5 Validation and Debugging

- Validation
    - Tested the script with various images and live feeds to ensure:
        - Correct stress classification for emotions (angry, sad, fear, disgust).
        - Accurate identification of non-stress emotions (happy, surprise).
- Debugging

- Common issues to address:
  - No Face Detected: Ensure proper lighting and visible faces in the image or live feed.
  - Incorrect Classification: Reviewing preprocessing steps and considering improving the DeepFace model configuration.

3.2.2 Stress Detection Using EDA sensor

- ESP32:
  The ESP32-WROOM-32 is a compact and versatile microcontroller designed for IoT applications. With its dual-core processor, built-in Wi-Fi, and Bluetooth capabilities, it efficiently handles data processing and communication in real time. In this project, the ESP32 serves as the central hub, collecting physiological data from sensors, running a lightweight stress detection model, and transmitting results to the mobile app. Its small size, low power consumption, and flexibility make it ideal for wearable stress monitoring and intervention systems.

- EDA sensor:
  The GSR sensor measures changes in the skin's electrical conductivity, which occur due to sweat gland activity influenced by stress levels. When stress increases, the sensor detects the subtle rise in conductance and provides an analog signal for analysis. In this project, the GSR sensor plays a crucial role in monitoring physiological stress responses in real time. Its integration with the ESP32 ensures accurate and timely stress detection, forming the backbone of the system's real-time stress monitoring capabilities.

3.2.2.1 System Setup and Initialization

- Hardware Components:
  - Using an EDA sensor connected to an ESP32 microcontroller to measure electrodermal activity.
  - The GSR values from the EDA sensor indicate skin conductivity, which correlates with stress levels.
- Wi-Fi Configuration:
  - Connect the ESP32 to a Wi-Fi network by specifying the WIFI_SSID and WIFI_PASSWORD.
  - Ensure the ESP32 has a stable Wi-Fi connection .
- Firebase Setup:
  - Configure Firebase Database to store GSR data.

    - Use the provided Firebase credentials (FIREBASE_HOST and FIREBASE_AUTH) to authenticate and initialize Firebase communication.
- EDA Sensor Pin:
    - Connect the EDA sensor to the specified analog pin (GSR_PIN 34).

3.2.2.2 Data Collection and Processing

- Data Reading:
    - Collect 20 consecutive readings from the EDA sensor using analogRead(GSR_PIN).
- Averaging GSR Values:
    - Compute the average of the collected GSR values to reduce noise and fluctuations in the readings.
- Threshold Determination:
    - Define a threshold value for classifying stress.
    - Compare the average GSR value to this threshold:
        - Above Threshold: Stress detected.
        - Below Threshold: Relaxed.

3.2.2.3 Uploading Data to Firebase

- Real-Time Storage:
    - Use Firebase to store the averaged GSR values along with a unique timestamp-based path.
- Error Handling:
    - Implement robust error checking for Firebase operations to ensure data reliability.
- Testing:
    - Validate the successful upload of GSR values by monitoring Firebase entries.

3.2.2.4 Stress Classification

- Threshold-Based Classification:
    - The threshold logic for stress detection is currently implemented on the ESP32.
- Integration with Software:
    - The classification logic for combining GSR data and emotion detection will be handled in the software.
    - This integration is under development and will involve syncing EDA sensor data with emotion detection models.

3.3 Web Application Methodology

The web application serves as the main interface for users to interact with the stress detection and management system. It is built using React.js, a popular JavaScript framework known for its simplicity and ability to create dynamic, interactive user interfaces. The goal of this application is to fetch real-time data from a Firebase database, assess the user's stress level, and provide meaningful feedback and resources to help them relax and feel better.

3.3.1 Setting Up the Environment

To start, the app connects to Firebase, which acts as a cloud backend. We initialize Firebase using the provided configuration and use it to pull live data such as GSR (Galvanic Skin Response) readings and facial emotion classification which is uploaded by the ESP32 microcontroller and an emotion detection system.

Firebase allows the app to listen for changes as they happen, meaning any updates (like a sudden spike in stress) immediately reflect on the user interface without needing to refresh the page.

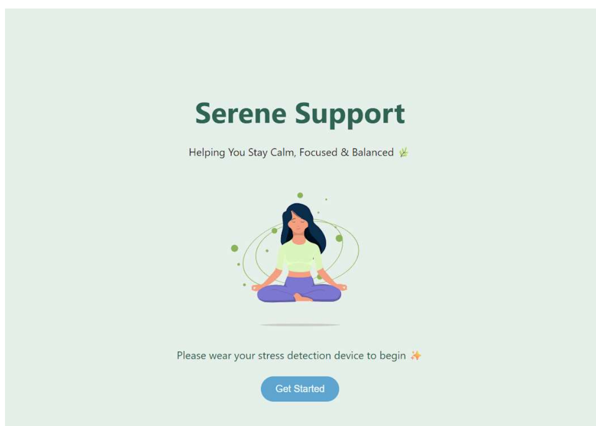


Figure 3.6 : Home screen

3.3.2 How It Reads and Uses the Data

- The app keeps track of key pieces of information:
    - The user's GSR value, which reflects their stress level.
    - Their emotion label, such as "Relaxed" or "Stressed".

- Basic user details like name and age.

When the app detects a high GSR value and a stressed facial expression, it responds by showing calming suggestions and resources. If the user seems calm, it provides positive reinforcement to keep them motivated.

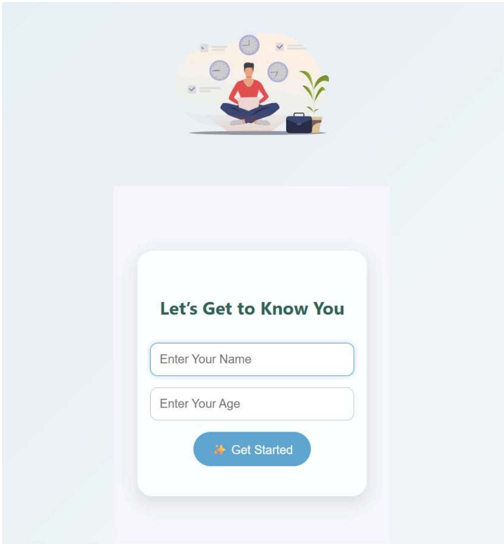


Figure 3.7 : User information page

3.3.3 Real-Time Stress Detection Logic

We set a threshold for GSR (e.g., values above 1800). If the GSR is high and the emotion is tagged as "stress," the app assumes the user is feeling overwhelmed. It then switches the view to show various relaxation methods, including yoga, meditation, soothing music, and helpful lifestyle suggestions like connecting with nature or pets.

3.3.4. Engaging and Helpful UI

To make the app more engaging, we use Lottie animations lightweight, visually appealing animations that play in real-time. These animations show peaceful scenes, like someone meditating or walking outdoors, and they enhance the overall user experience.

The app also greets users personally (e.g., “Hi, Alice!”) and guides them gently through the process, from entering their name and age to choosing an activity that suits their mood.

3.3.5. Curated Wellness Content

If the system detects stress, the app presents a carefully chosen list of YouTube videos related to:

- Yoga sessions
- Guided meditations
- Relaxing music

These links are displayed in a simple, easy-to-navigate list format, making it effortless for users to try them out immediately.

3.3.6. Clean Design and Smooth Interaction

We've kept the interface light, soothing, and distraction-free. The CSS ensures that:

- The layout adapts to different screen sizes
- Input forms are user-friendly
- Colors and transitions are gentle, in line with the calming theme

Buttons are clearly labeled, and forms are designed to make users feel at ease while using the app.

3.3.7. Built to Scale

The app uses reusable components, meaning new features can be added easily without starting from scratch. For example, more stress-relief methods, user tracking, or even chatbots could be added in the future.

The code is also modular and clean, following best practices for React development. This makes it maintainable and ready for upgrades as the system grows.

CHAPTER 4
RESULTS AND DISCUSSION

CHAPTER - 4

RESULTS AND DISCUSSION

The results include the performance comparison of various machine learning models for facial emotion recognition, along with the implementation of a real-time stress detection system. The project combines CNN-based emotion classification with GSR sensor data to identify stress levels. Both hardware and software components are integrated through Firebase, and a web application provides users with stress feedback and relaxation suggestions.

4.1 Result for Training algorithms on a facial emotion dataset

Table 4.1 Comparison of Different Algorithms with their Performance Metrics

Algorithm	Metric	Negative	Positive	Neutral
CNN	Precision	**1.00**	0.97	**1.00**
	Recall	**1.00**	**1.00**	**0.99**
	F1 score	**1.00**	0.99	**0.99**
KNN	Precision	0.81	0.95	0.87
	Recall	0.66	0.78	0.67
	F1 score	0.83	0.94	0.88
SVM	Precision	0.98	0.96	0.88
	Recall	0.83	0.99	0.94
	F1 score	0.90	0.97	0.99
Random Forest	Precision	0.98	**1.00**	**1.00**
	Recall	**1.00**	**1.00**	0.92

	F1 score	0.99	**1.00**	**0.99**
FCN	**Precision**	0.00	0.96	0.53
	Recall	0.00	0.97	0.95
	F1 score	0.00	0.96	0.68

CNN and Random Forest are the top-performing algorithms.

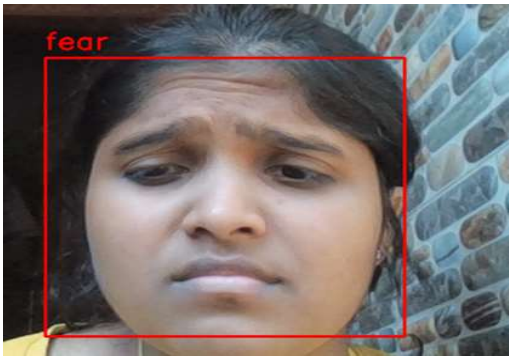
- SVM performs well but slightly lags behind.
- KNN shows moderate performance, struggling with recall for certain classes.
- FCN fails significantly in classifying Negative and Neutral categories.
- CNN or Random Forest are recommended for deployment.

4.2 Results for Hardware and Software Implementation

4.2.1 Software Implementation

A trained model along with the DeepFace module was used to detect faces and identify emotions. Emotions such as 'angry' and 'sad' were classified as stress indicators, while other emotions remained in their original categories.

Fig 4.1 Image with angry and sad emotion classified as stress



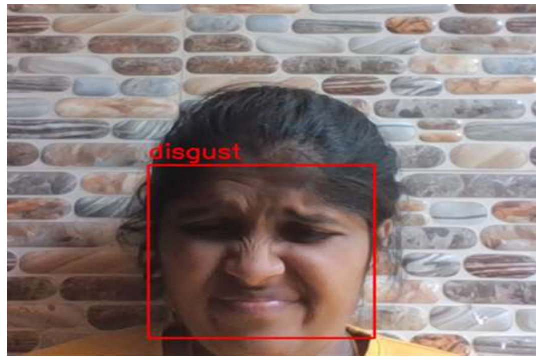


Fig 4.2 Images with fear and disgust emotions

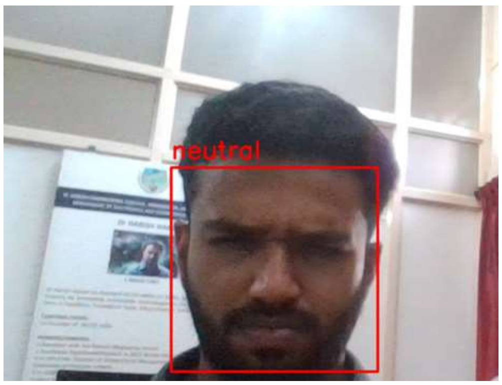


Fig 4.3 Images with neutral and surprise emotions

Fig 4.4 Image with happy emotion

4.2.2 Hardware Implementation

The GSR-based stress detection system successfully monitored and processed skin conductance data to identify stress levels. The average GSR values were calculated over 20 readings, and the system classified the user as either "Relaxed" or "Stressed" based on a predefined threshold of 1200.

- Relaxed State: GSR Average below 1200.
- Stressed State: GSR Average equal to or above 1200.

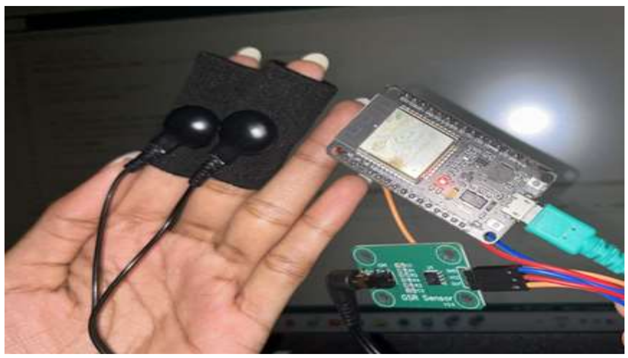


Fig 4.5 Hardware implementation

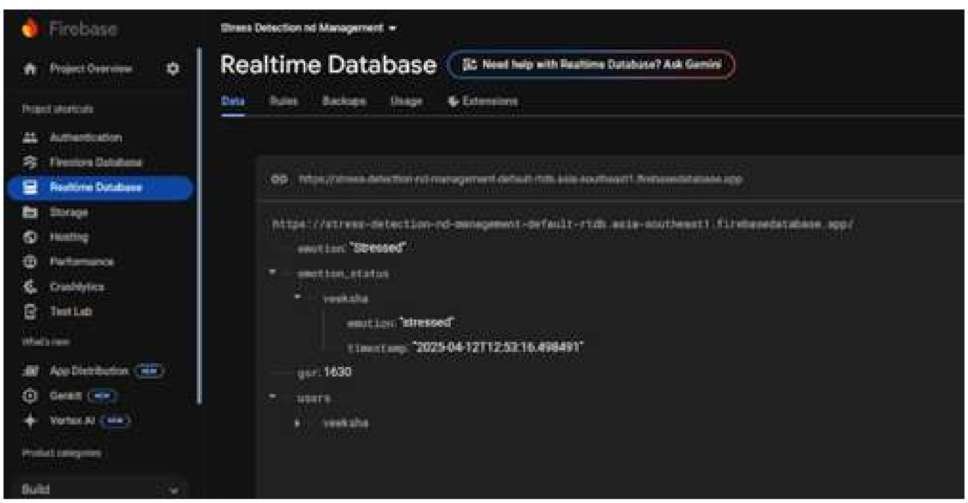


Fig 4.6   GSR Data Visualization in Firebase Realtime Database

4.2.3 Emotion and GSR-Based Stress Feedback System

The Emotion detection outputs with the EDA sensor data through firebase is synced and Based on the analysis.

The below mentioned is the overview of the Web application:

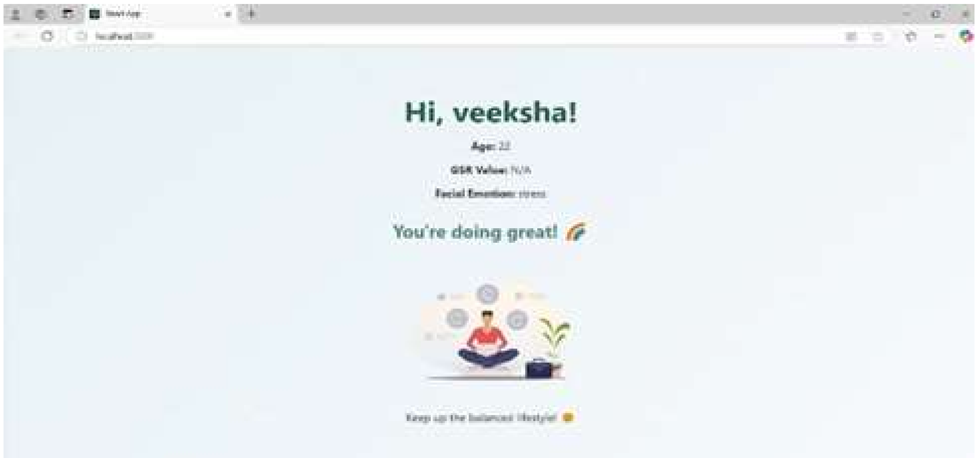


Fig 4.7 Home Page view of the Web Application

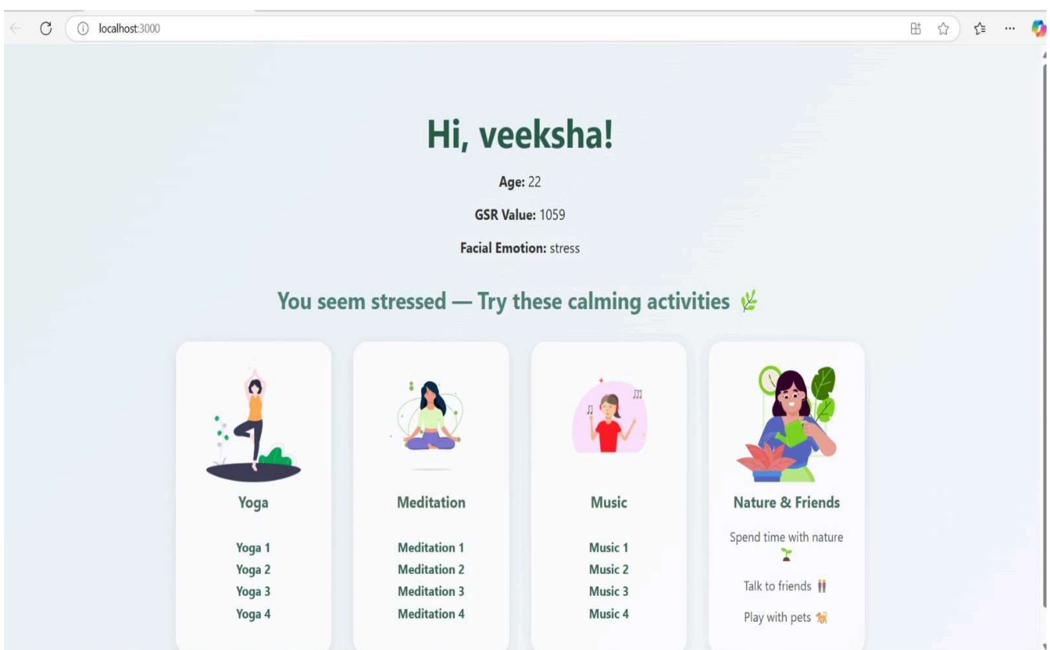


Figure 4.8 Image demonstrating the working of App

The application detects stress by combining facial emotion analysis and GSR sensor values. When stress is identified, the application displays calming activity suggestions like yoga, meditation, music, and spending time with nature or friends.

CHAPTER 5
CONCLUSION

CHAPTER - 5

CONCLUSION

This project presents a practical approach to stress detection and personalized intervention by combining facial emotion recognition with physiological signals. By integrating data from facial expressions and EDA measured via a GSR sensor, the system shows a more complete understanding of a person's emotional and physical state. Using machine learning models, especially Convolutional Neural Networks (CNN) the system can accurately identify stress-related emotions such as anger and sadness, while the ESP32-based GSR sensor provides real-time physiological insights.

What sets this work apart is the integration of emotion detection, sensor data, and cloud storage via Firebase to offer timely and personalized stress-relief suggestions through a web application. Whether it's recommending yoga, meditation, or calming music, the system ensures that the user receives support that suits their individual needs in that moment.

Stress is increasingly common thus this project shows how technology can be used thoughtfully not just to detect stress, but to respond with care. It shows accessible, user-friendly tools that improve individuals to take control of their mental well-being, offering both early detection and immediate support in one integrated platform.

CHAPTER 6
FUTURE SCOPE AND IMPLICATIONS

CHAPTER - 6

FUTURE SCOPE AND IMPLICATIONS

The project system has a lot of room to grow in the future. Using more sophisticated machine learning algorithms and incorporating other data modalities, like voice analysis and physical activity levels, can improve its accuracy. These enhancements might offer a more thorough comprehension of stressors.

Additionally, the system can develop to provide individualized interventions based on a person's stress profile, such as breathing exercises, mindfulness exercises, or relaxation techniques. Optimizing the hardware and software elements could make the solution more affordable, portable, and available to a wider range of users. It could also integrate with wearable technology, such as fitness trackers or smartwatches.

Long-term stress monitoring is another potential use for the system, which would allow users to pinpoint stressors and modify their lifestyles.It could be a useful tool for counselors and therapists, giving them insight into their clients' feelings and enhancing the effectiveness and personalization of therapy.

Our research paper has been accepted for presentation at the 16th International Conference on Recent Engineering and Technology (ICRET 2025), organized by East West College of Engineering, Bangalore, India, in collaboration with Samarkand State University, Uzbekistan. The conference will be held on the 17th and 18th of May 2025 at East West College of Engineering, Bangalore, and will be conducted in hybrid mode, allowing both online and offline participation.

REFERENCES

[1] R. M. Sabour, Y. Benezeth, P. De Oliveira, J. Chappée, and F. Yang, "UBFC-Phys: A Multimodal Database for Psychophysiological Studies of Social Stress," IEEE Transactions on Affective Computing, vol. 14, no. 1, pp. 622–634, Jan.-Mar. 2023, doi: 10.1109/TAFFC.2021.3056960.

[2] J. Bakker, M. Pechenizkiy, and N. Sidorova, "What's Your Current Stress Level? Detection of Stress Patterns from GSR Sensor Data," 2011 IEEE 11th International Conference on Data Mining Workshops, Vancouver, BC, Canada, 2011, pp. 573-580, doi: 10.1109/ICDMW.2011.178.

[3] O. Parlak, "Portable and wearable real-time stress monitoring: A critical review," Sensors and Actuators Reports, vol. 3, p. 100036, 2021, doi: 10.1016/j.snr.2021.100036. Available: https://doi.org/10.1016/j.snr.2021.100036.

[4] G. Ribeiro, O. Postolache and F. F. Martín, "A New Intelligent Approach for Automatic Stress Level Assessment Based on Multiple Physiological Parameters Monitoring," IEEE Transactions on Instrumentation and Measurement, vol. 73, pp. 1-14, 2024, Art no. 4001714, doi: 10.1109/TIM.2023.3342218.

[5] Hosseini, R. Fang, R. Zhang, S. Rafatirad, and H. Homayoun, "Emotion and Stress Recognition Utilizing Galvanic Skin Response and Wearable Technology: A Real-time Approach for Mental Health Care," in 2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Istanbul, Turkiye, 2023, pp. 1125-1131. doi: 10.1109/BIBM58861.2023.10386049.

[6] S. Gedam and S. Paul, "A Review on Mental Stress Detection Using Wearable Sensors Machine Learning Techniques," *IEEE Access*, vol. 9, pp. 84045-84066, 2021, doi: .1109/ACCESS.2021.3085502.

[7] G. Taskasaplidis, D. A. Fotiadis, and P. D. Bamidis, "Review of Stress Detection Methods Using Wearable Sensors," *IEEE Access*, vol. 12, pp. 38219-38246, 2024, doi: 10.1109/ACCESS.2024.3373010.

[8] H. P. Chandika, B. Soumya, B. N. E. Reddy, and B. M. S. S. Manideep, "Real-Time Stress Detection and Analysis using Facial Emotion Recognition," International Journal of Advanced Research in Computer and Communication Engineering, vol. 13, no. 3, pp. 158-172, Mar. 2024, doi: 10.17148/IJARCCE.2024.13324.

[9] P. Bobade and M. Vani, "Stress Detection with Machine Learning and Deep Learning using Multimodal Physiological Data," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 51-57, doi: 10.1109/ICIRCA48905.2020.9183244.

[10] M. Pediaditis et al., "Extraction of facial features as indicators of stress and anxiety," in 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milan, Italy, 2015, pp. 3711-3714. doi: 10.1109/EMBC.2015.731919.

[11] L. Santamaria-Granados, M. Munoz-Organero, G. Ramirez-González, E. Abdulhay, and N. Arunkumar, "Using Deep Convolutional Neural Network for Emotion Detection on a Physiological Signals Dataset (AMIGOS)," IEEE Access, vol. 7, pp. 57-67, 2019. doi: 10.1109/ACCESS.2018.2883213.

APPENDIX

LIST OF COMPONENTS USED FOR PROJECT

SL.NO.	COMPONENT	VALUE	DETAILS
1	GSR Skin Current Sensor	1200.00	Measures skin conductivity to assess stress levels
2	ESP32 wroom-32	489.00	Microcontroller with Wi-Fi and Bluetooth capabilities for IoT applications.
	TOTAL	1689.00	

LIST OF SOFTWARE TOOLS USED FOR THE PROJECT

SL.NO.	SOFTWARE	VERSION/ DETAILS
1	Arduino IDE	2.3.4 / It is used for programming and uploading code to the ESP32.
2	Visual Studio Code	1.93 / Used as a code editor for more complex development tasks.
3	Firebase	A cloud platform used in data storage, authentication and real-time database functionality.

Grove - GSR

Release date： 9/20/2015

Version： 1.0

Wiki: http://www.seeedstudio.com/depot/Grove-GSR-p-1614.html

Bazaar: http://www.seeedstudio.com/depot/Grove-GSR-p-1614.html

Document Revision History

Revision	Date	Author	Description
1.0	Sep 21, 2015	Victor.He	Create file

Contents

Disclaimer

For physical injuries and possessions loss caused by those reasons which are not related to product quality, such as operating without following manual guide, natural disasters or force majeure, we take no responsibility for that.

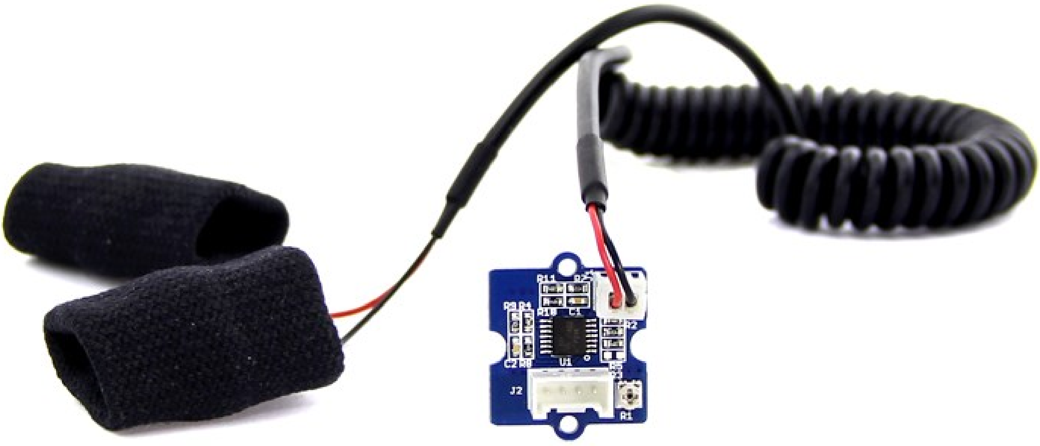
Under the supervision of Seeed Technology Inc., this manual has been compiled and published which covered the latest product description and specification. The content of this manual is subject to change without notice.

Copyright

1. Introduction

GSR, standing for galvanic skin response, is a method of measuring the electrical conductance of the skin. Strong emotion can cause stimulus to your sympathetic nervous system, resulting more sweat being secreted by the sweat glands. Grove – GSR allows you to spot such strong emotions by simple attaching two electrodes to two fingers on one hand, an interesting gear to create emotion related projects, like sleep quality monitor.

SKU: SEN01400P



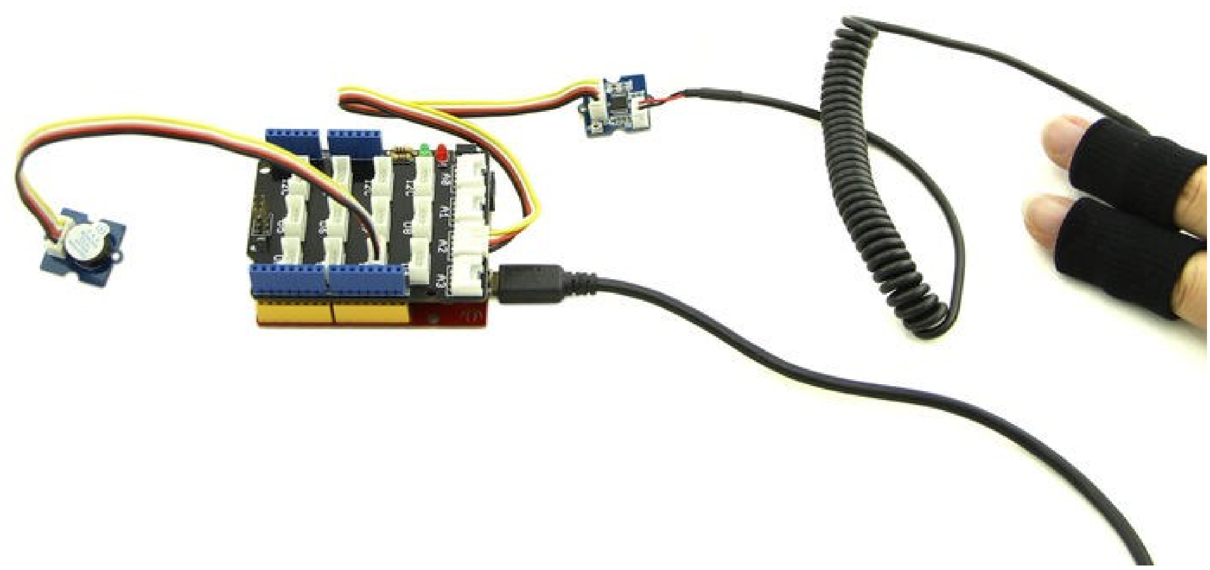
2. Specifications

- Input Voltage: 5V/3.3V
- Sensitivity adjustable via a potentiometer
- External measuring finger cots

3. Demonstration

In the following we are showing you how to use the Grove - GSR.

Connect Grove - GSR Sensor to the analog port A2 of Grove-Basic Shield and Grove - Buzzer to digital port3.

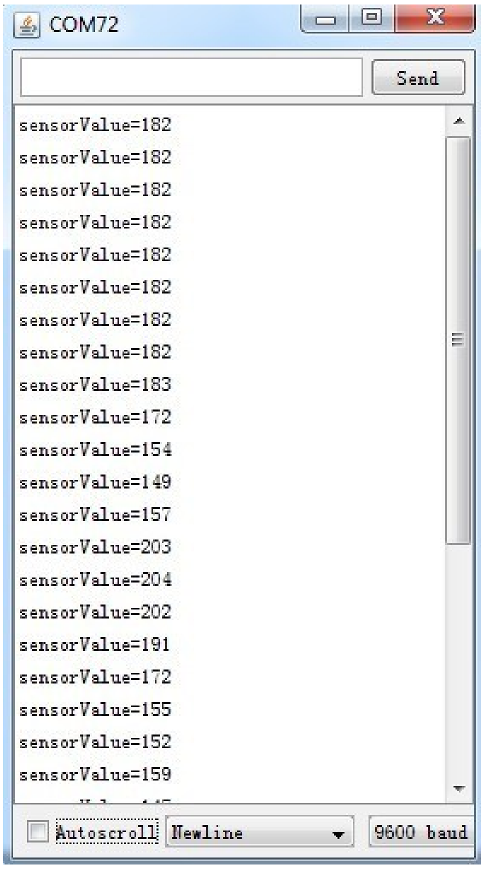


Copy and paste the code below to a new Arduino sketch and upload it to Arduino.

```
const int BUZZER=3;
const int GSR=A2;
int threshold=0;
int sensorValue;

void setup(){
    long sum=0;
    Serial.begin(9600);
    pinMode(BUZZER,OUTPUT);
    digitalWrite(BUZZER,LOW);
    delay(1000);

    for(int i=0;i<500;i++)
    {
        sensorValue=analogRead(GSR);
        sum += sensorValue;
        delay(5);
```

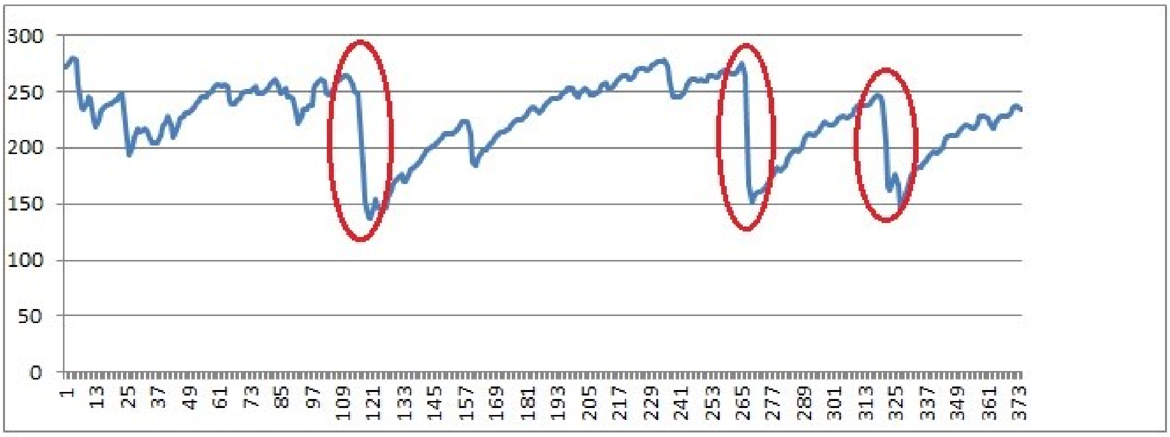
```
    }
    threshold = sum/500;
    Serial.print("threshold =");
    Serial.println(threshold);
}

void loop(){
    int temp;
    sensorValue=analogRead(GSR);
    Serial.print("sensorValue=");
    Serial.println(sensorValue);
    temp = threshold - sensorValue;
    if(abs(temp)>50)
    {
        sensorValue=analogRead(GSR);
        temp = threshold - sensorValue;
        if(abs(temp)>50){
            digitalWrite(BUZZER,HIGH);
            Serial.println("YES!");
            delay(3000);
            digitalWrite(BUZZER,LOW);
            delay(1000);}
    }
}
```

Wear the finger sheath and relax, Now open serial monitor, we can see:



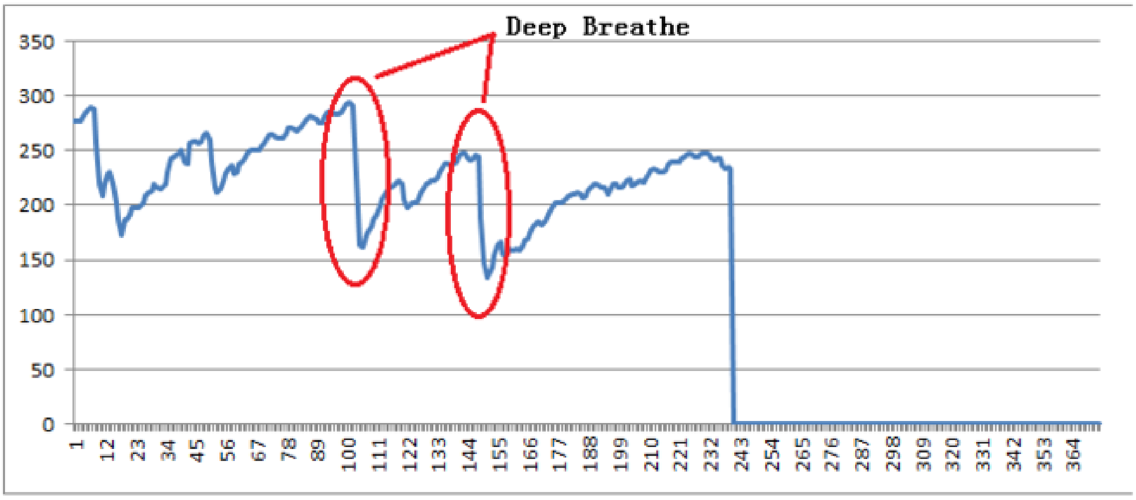
Then take a deep breath. The buzzer should buzz now. And an obvious change in the output value should be observed.

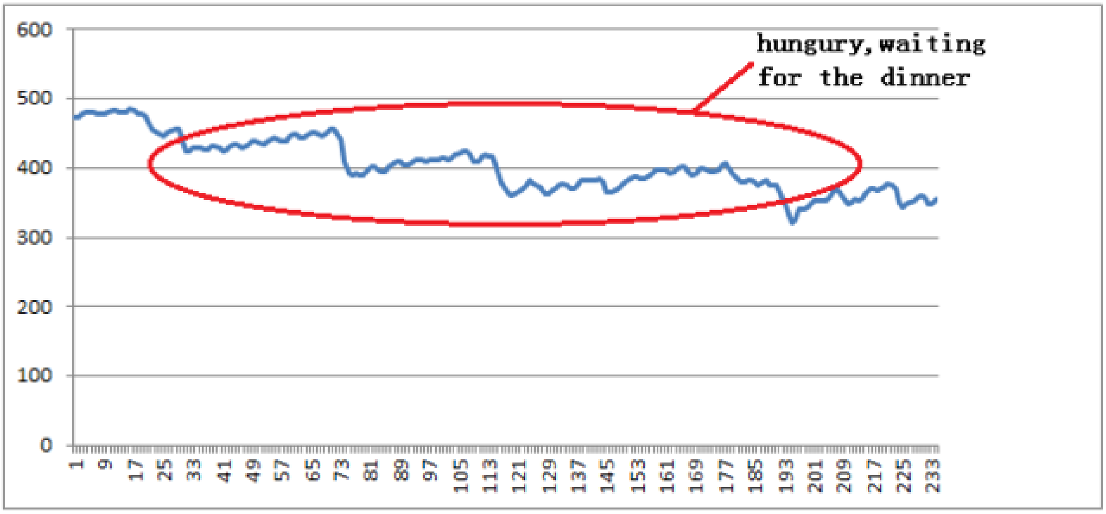
The below is a graphs which is created in Excel using the data above. X axis represents time. and Y axis GSR data.

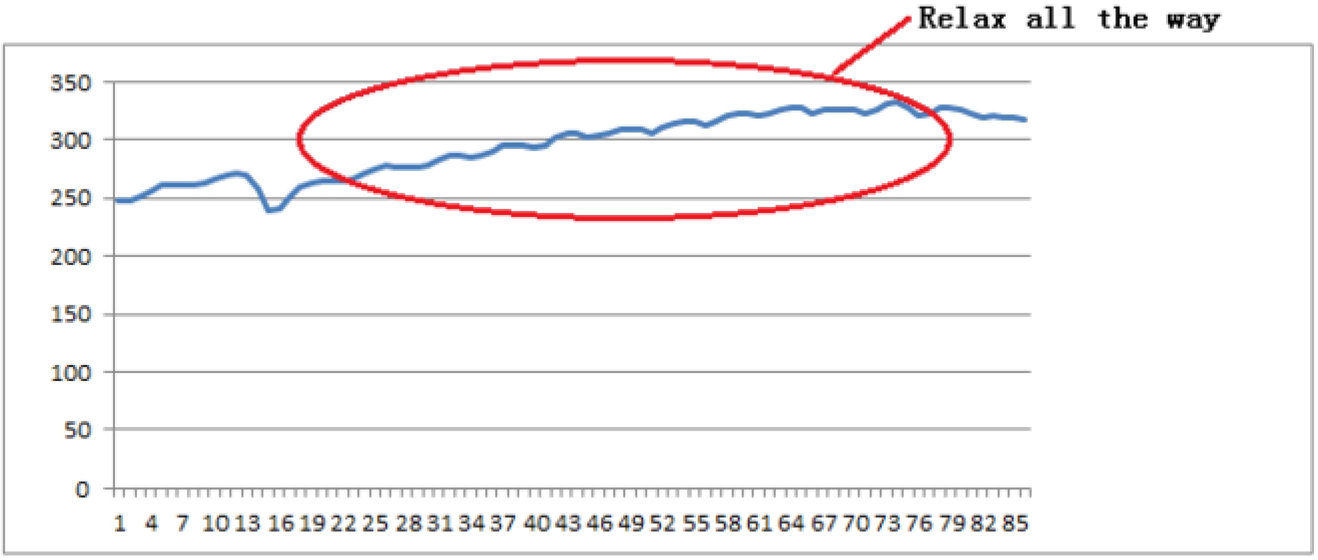

300
250
200
150
100
50
0
1
13
25
37
49
61
73
85
97
109
121
133
145
157
169
181
193
205
217
229
241
253
265
277
289
301
313
325
337
349
361
373

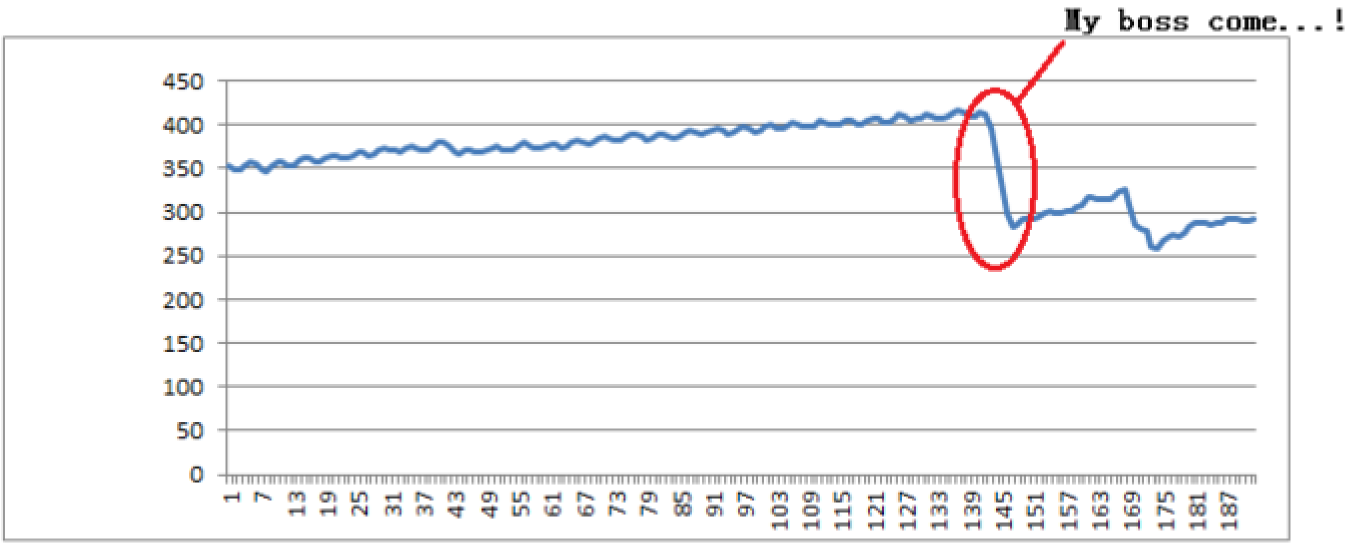
4. Reference

There are several graphs which are created in excel using GSR data.You can open the File:GSR sensor data.xls to see the detail data.






Relax all the way
350
300
250
200
150
100
50
0
1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70 73 76 79 82 85


My boss come...!
450
400
350
300
250
200
150
100
50
0
1 7 13 19 25 31 37 43 49 55 61 67 73 79 85 91 97 103 109 115 121 127 133 139 145 151 157 163 169 175 181 187

5. Resources

[Grove - GSR Eagle File]

[LM324 datasheet]

[File:GSR sensor data.xls]

ESP-WROOM-32 Datasheet

Espressif Systems

September 8, 2017

About This Guide

This document provides the specifications for the ESP-WROOM-32 module.

The document structure is as follows:

Chapter	Title	Subject
Chapter 1	Preface	An overview of ESP-WROOM-32
Chapter 2	Pin Definitions	Device pinout and pin descriptions
Chapter 3	Functional Description	Description of major functional modules
Chapter 4	Peripherals and Sensors	Description of peripherals
Chapter 5	Electrical Characteristics	Electrical characteristics and specifications of ESP-WROOM-32
Chapter 6	Peripheral Schematics	The peripheral schematics of ESP-WROOM-32
Chapter 7	Schematics	The schematics of ESP-WROOM-32
Chapter 8	Dimensions	The physical dimensions of ESP-WROOM-32
Chapter 9	Learning Resources	ESP32-related must-read materials and must-have resources

Release Notes

Date	Version	Release notes
2016.08	V1.0	First release.
2016.11	V1.1	Updated Chapter 6 Schematics.
2016.11	V1.2	Added Figure 7 Peripheral Schematics.
2016.12	V1.3	Updated Section 2.1 Pin Layout.
2017.03	V1.4	Updated Chapter 1 Preface; Updated Chapter 2 Pin Definitions; Updated Chapter 3 Functional Description; Updated Table Recommended Operating Conditions; Updated Table 9 Wi-Fi Radio Characteristics; Updated Section 5.4 Reflow Profile; Added Chapter 9 Learning Resources.
2017.03	V1.5	Updated Section 2.2 Pin Description; Updated Section 3.2 External Flash and SRAM; Updated Section 4.1 Peripherals and Sensors Description.
2017.04	V1.6	Added Figure 2 Reflow Profile.
2017.04	V1.7	Added the module's dimensional tolerance; Changed the input impedance value of 50Ω in Table 9 Wi-Fi Radio Characteristics to output impedance value of 30+j10 Ω.
2017.05	V1.8	Updated Figure 1 Top and Side View of ESP-WROOM-32.
2017.06	V1.9	Added a note to Section 2.1 Pin Layout; Updated Section 3.3 Crystal Oscillators; Updated Figure 3 ESP-WROOM-32 Schematics; Added Documentation Change Notification.

Date	Version	Release notes
2017.08	V2.0	Changed the sensitivity of NZIF receiver to -97 dBm in Table 2; Updated the dimensions of the module; Updated Table 6 Power Consumption by Power Modes, and added two notes to it; Updated Table 8, 9, 10, 11; Added Chapter 8; Added the link to certification download.
2017.09	V2.1	Updated operating voltage/power supply range updated to 2.7 ~ 3.6V; Updated Chapter 7.

Documentation Change Notification

Espressif provides email notifications to keep customers updated on changes to technical documentation. Please subscribe here.

Certificates

Download certificates for Espressif products from here.

Contents

List of Tables

List of Figures

1. Overview

ESP-WROOM-32 is a powerful, generic Wi-Fi+BT+BLE MCU module that targets a wide variety of applications, ranging from low-power sensor networks to the most demanding tasks, such as voice encoding, music streaming and MP3 decoding.

At the core of this module is the ESP32-D0WDQ6 chip*. The chip embedded is designed to be scalable and adaptive. There are two CPU cores that can be individually controlled, and the clock frequency is adjustable from 80 MHz to 240 MHz. The user may also power off the CPU and make use of the low-power co-processor to constantly monitor the peripherals for changes or crossing of thresholds. ESP32 integrates a rich set of peripherals, ranging from capacitive touch sensors, Hall sensors, low-noise sense amplifiers, SD card interface, Ethernet, high-speed SPI, UART, I2S and I2C.

Note:
* For details on the part number of the ESP32 series, please refer to the document *ESP32 Datasheet*.

The integration of Bluetooth, Bluetooth LE and Wi-Fi ensures that a wide range of applications can be targeted, and that the module is future proof: using Wi-Fi allows a large physical range and direct connection to the internet through a Wi-Fi router, while using Bluetooth allows the user to conveniently connect to the phone or broadcast low energy beacons for its detection. The sleep current of the ESP32 chip is less than 5 μA, making it suitable for battery powered and wearable electronics applications. ESP32 supports a data rate of up to 150 Mbps, and 20.5 dBm output power at the antenna to ensure the widest physical range. As such the chip does offer industry-leading specifications and the best performance for electronic integration, range, power consumption, and connectivity.

The operating system chosen for ESP32 is freeRTOS with LwIP; TLS 1.2 with hardware acceleration is built in as well. Secure (encrypted) over the air (OTA) upgrade is also supported, so that developers can continually upgrade their products even after their release.

Table 2 provides the specifications of ESP-WROOM-32.

Table 2: ESP-WROOM-32 Specifications

Categories	Items	Specifications
Wi-Fi	RF certification	FCC/CE/IC/TELEC/KCC/SRRC/NCC
	Protocols	802.11 b/g/n/e/i (802.11n up to 150 Mbps)
		A-MPDU and A-MSDU aggregation and 0.4 μs guard interval support
	Frequency range	2.4 ~ 2.5 GHz
Bluetooth	Protocols	Bluetooth v4.2 BR/EDR and BLE specification
	Radio	NZIF receiver with -97 dBm sensitivity
		Class-1, class-2 and class-3 transmitter
		AFH
	Audio	CVSD and SBC

Categories	Items	Specifications
Hardware	Module interface	SD card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S, IR
		GPIO, capacitive touch sensor, ADC, DAC, LNA pre-amplifier
	On-chip sensor	Hall sensor, temperature sensor
	On-board clock	40 MHz crystal
	Operating voltage/Power supply	2.7 ~ 3.6V
	Operating current	Average: 80 mA
	Minimum current delivered by power supply	500 mA
	Operating temperature range	-40°C ~ +85°C
	Ambient temperature range	Normal temperature
	Package size	18±0.2 mm x 25.5±0.2 mm x 3.1±0.15 mm
Software	Wi-Fi mode	Station/SoftAP/SoftAP+Station/P2P
	Wi-Fi Security	WPA/WPA2/WPA2-Enterprise/WPS
	Encryption	AES/RSA/ECC/SHA
	Firmware upgrade	UART Download / OTA (download and write firmware via network or host)
	Software development	Supports Cloud Server Development / SDK for custom firmware development
	Network protocols	IPv4, IPv6, SSL, TCP/UDP/HTTP/FTP/MQTT
	User configuration	AT instruction set, cloud server, Android/iOS app

2. Pin Definitions

2.1 Pin Layout

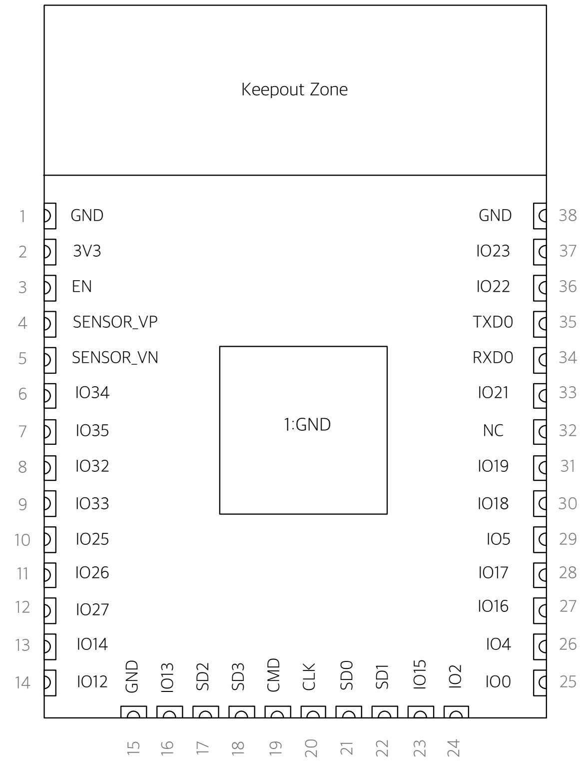


Figure 1: ESP-WROOM-32 Pin layout

2.2 Pin Description

ESP-WROOM-32 has 38 pins. See pin definitions in Table 3.

Table 3: Pin Definitions

Name	No.	Type	Function
GND	1	P	Ground
3V3	2	P	Power supply.
EN	3	I	Chip-enable signal. Active high.
SENSOR_VP	4	I	GPIO36, SENSOR_VP, ADC_H, ADC1_CH0, RTC_GPIO0
SENSOR_VN	5	I	GPIO39, SENSOR_VN, ADC1_CH3, ADC_H, RTC_GPIO3
IO34	6	I	GPIO34, ADC1_CH6, RTC_GPIO4
IO35	7	I	GPIO35, ADC1_CH7, RTC_GPIO5
IO32	8	I/O	GPIO32, XTAL_32K_P (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
IO33	9	I/O	GPIO33, XTAL_32K_N (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8
IO25	10	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
IO26	11	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
IO27	12	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV

Name	No.	Type	Function
IO14	13	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
IO12	14	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
GND	15	P	Ground
IO13	16	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
SHD/SD2*	17	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
SWP/SD3*	18	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
SCS/CMD*	19	I/O	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
SCK/CLK*	20	I/O	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
SDO/SD0*	21	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
SDI/SD1*	22	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
IO15	23	I/O	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
IO2	24	I/O	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
IO0	25	I/O	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
IO4	26	I/O	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
IO16	27	I/O	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
IO17	28	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
IO5	29	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
IO18	30	I/O	GPIO18, VSPICLK, HS1_DATA7
IO19	31	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
NC	32	-	-
IO21	33	I/O	GPIO21, VSPIHD, EMAC_TX_EN
RXD0	34	I/O	GPIO3, U0RXD, CLK_OUT2
TXD0	35	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
IO22	36	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
IO23	37	I/O	GPIO23, VSPID, HS1_STROBE
GND	38	P	Ground

Note:
* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.

2.3 Strapping Pins

Please refer to ESP-WROOM-32 schematics.

ESP32 has five strapping pins, which can be seen in Section 6 Schematics:

- MTDI

- GPIO0
- GPIO2
- MTDO
- GPIO5

Software can read the value of these five bits from the register "GPIO_STRAPPING".

During the chip power-on reset, the latches of the strapping pins sample the voltage level as strapping bits of "0" or "1", and hold these bits until the chip is powered down or shut down. The strapping bits configure the device boot mode, the operating voltage of VDD_SDIO and other system initial settings.

Each strapping pin is connected with its internal pull-up/pull-down during the chip reset. Consequently, if a strapping pin is unconnected or the connected external circuit is high-impendence, the internal weak pull-up/pull-down will determine the default input level of the strapping pins.

To change the strapping bit values, users can apply the external pull-down/pull-up resistances, or apply the host MCU's GPIOs to control the voltage level of these pins when powering on ESP32.

After reset, the strapping pins work as the normal functions pins.

Refer to Table 4 for detailed boot modes' configuration by strapping pins.

Table 4: Strapping Pins

<table>
<tr><th colspan="6">Voltage of Internal LDO (VDD_SDIO)</th></tr>
<tr><td>Pin</td><td>Default</td><td colspan="2">3.3V</td><td colspan="2">1.8V</td></tr>
<tr><td>MTDI</td><td>Pull-down</td><td colspan="2">0</td><td colspan="2">1</td></tr>
<tr><th colspan="6">Booting Mode</th></tr>
<tr><td>Pin</td><td>Default</td><td colspan="2">SPI Boot</td><td colspan="2">Download Boot</td></tr>
<tr><td>GPIO0</td><td>Pull-up</td><td colspan="2">1</td><td colspan="2">0</td></tr>
<tr><td>GPIO2</td><td>Pull-down</td><td colspan="2">Don't-care</td><td colspan="2">0</td></tr>
<tr><th colspan="6">Debugging Log on U0TXD During Booting</th></tr>
<tr><td>Pin</td><td>Default</td><td colspan="2">U0TXD Toggling</td><td colspan="2">U0TXD Silent</td></tr>
<tr><td>MTDO</td><td>Pull-up</td><td colspan="2">1</td><td colspan="2">0</td></tr>
<tr><th colspan="6">Timing of SDIO Slave</th></tr>
<tr><td>Pin</td><td>Default</td><td>Falling-edge Input
Falling-edge Output</td><td>Falling-edge Input
Rising-edge Output</td><td>Rising-edge Input
Falling-edge Output</td><td>Rising-edge Input
Rising-edge Output</td></tr>
<tr><td>MTDO</td><td>Pull-up</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>GPIO5</td><td>Pull-up</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
</table>

Note:
Firmware can configure register bits to change the settings of "Voltage of Internal LDO (VDD_SDIO)" and "Timing of SDIO Slave" after booting.

3. Functional Description

This chapter describes the modules and functions integrated in ESP-WROOM-32.

3.1 CPU and Internal Memory

ESP32-D0WDQ6 contains two low-power Xtensa® 32-bit LX6 microprocessors. The internal memory includes:

- 448 KB of ROM for booting and core functions.
- 520 KB (8 KB RTC FAST Memory included) of on-chip SRAM for data and instruction.
  - 8 KB of SRAM in RTC, which is called RTC FAST Memory and can be used for data storage; it is accessed by the main CPU during RTC Boot from the Deep-sleep mode.
- 8 KB of SRAM in RTC, which is called RTC SLOW Memory and can be accessed by the co-processor during the Deep-sleep mode.
- 1 kbit of eFuse, of which 256 bits are used for the system (MAC address and chip configuration) and the remaining 768 bits are reserved for customer applications, including Flash-Encryption and Chip-ID.

3.2 External Flash and SRAM

ESP32 supports up to four 16-MB of external QSPI flash and SRAM with hardware encryption based on AES to protect developers' programs and data.

ESP32 can access the external QSPI flash and SRAM through high-speed caches.

- Up to 16 MB of external flash are memory-mapped onto the CPU code space, supporting 8, 16 and 32-bit access. Code execution is supported.
- Up to 8 MB of external flash/SRAM are memory-mapped onto the CPU data space, supporting 8, 16 and 32-bit access. Data-read is supported on the flash and SRAM. Data-write is supported on the SRAM.

ESP-WROOM-32 integrates 4 MB of external SPI flash. The 4-MB SPI flash can be memory-mapped onto the CPU code space, supporting 8, 16 and 32-bit access. Code execution is supported. The integrated SPI flash is connected to GPIO6, GPIO7, GPIO8, GPIO9, GPIO10 and GPIO11. These six pins cannot be used as regular GPIO.

3.3 Crystal Oscillators

The ESP32 Wi-Fi/BT firmware can only support 40 MHz crystal oscillator for now.

3.4 RTC and Low-Power Management

With the use of advanced power management technologies, ESP32 can switch between different power modes (see Table 5).

- Power modes
  - Active mode: The chip radio is powered on. The chip can receive, transmit, or listen.
  - Modem-sleep mode: The CPU is operational and the clock is configurable. The Wi-Fi/Bluetooth baseband and radio are disabled.
  - Light-sleep mode: The CPU is paused. The RTC memory and RTC peripherals, as well as the ULP co-processor are running. Any wake-up events (MAC, host, RTC timer, or external interrupts) will wake up the chip.
  - Deep-sleep mode: Only the RTC memory and RTC peripherals are powered on. Wi-Fi and Bluetooth connection data are stored in the RTC memory. The ULP co-processor can work.
  - Hibernation mode: The internal 8-MHz oscillator and ULP co-processor are disabled. The RTC recovery memory is powered down. Only one RTC timer on the slow clock and some RTC GPIOs are active. The RTC timer or the RTC GPIOs can wake up the chip from the Hibernation mode.
- Sleep Patterns
  - Association sleep pattern: The power mode switches between the Active mode, Modem- and Light-sleep mode during this sleep pattern. The CPU, Wi-Fi, Bluetooth, and radio are woken up at predetermined intervals to keep Wi-Fi/BT connections alive.
  - ULP sensor-monitored pattern: The main CPU is in the Deep-sleep mode. The ULP co-processor takes sensor measurements and wakes up the main system, based on the data collected from sensors.

Table 5: Functionalities Depending on the Power Modes

Power mode	Active	Modem-sleep	Light-sleep	Deep-sleep	Hibernation
Sleep pattern	Association sleep pattern			ULP sensor-monitored pattern	-
CPU	ON	ON	PAUSE	OFF	OFF
Wi-Fi/BT baseband and radio	ON	OFF	OFF	OFF	OFF
RTC memory and RTC peripherals	ON	ON	ON	ON	OFF
ULP co-processor	ON	ON	ON	ON/OFF	OFF

The power consumption varies with different power modes/sleep patterns and work statuses of functional modules. Please see Table 6 for details.

Table 6: Power Consumption by Power Modes

Power mode	Description	Power consumption
Active (RF working)	Wi-Fi Tx packet 14 dBm ~ 19.5 dBm	Please refer to ESP32 Datasheet.
	Wi-Fi / BT Tx packet 0 dBm	
	Wi-Fi / BT Rx and listening	
	Association sleep pattern (by Light-sleep)	1 mA ~ 4 mA @DTIM3

Power mode	Description	Power consumption
Modem-sleep	The CPU is powered on.	Max speed 240 MHz: 30 mA ~ 50 mA
		Normal speed 80 MHz: 20 mA ~ 25 mA
		Slow speed 2 MHz: 2 mA ~ 4 mA
Light-sleep	-	0.8 mA
Deep-sleep	The ULP co-processor is powered on.	150 μA
	ULP sensor-monitored pattern	100 μA @1% duty
	RTC timer + RTC memory	10 μA
Hibernation	RTC timer only	5 μA
Power off	CHIP_PU is set to low level, the chip is powered off	0.1 μA

Note:

- During Deep-sleep, when the ULP co-processor is powered on, peripherals such as GPIO and I2C are able to work.
- When the system works in the ULP sensor-monitored pattern, the ULP co-processor works with the ULP sensor periodically; ADC works with a duty cycle of 1%, so the power consumption is 100 μA.

4. Peripherals and Sensors

4.1 Peripherals and Sensors Description

Table 7: Description of Peripherals and Sensors

Interface	Signal	Pin	Function
ADC	ADC1_CH0	SENSOR_VP	Two 12-bit SAR ADCs
	ADC1_CH3	SENSOR_VN	
	ADC1_CH4	IO32	
	ADC1_CH5	IO33	
	ADC1_CH6	IO34	
	ADC1_CH7	IO35	
	ADC2_CH0	IO4	
	ADC2_CH1	IO0	
	ADC2_CH2	IO2	
	ADC2_CH3	IO15	
	ADC2_CH4	IO13	
	ADC2_CH5	IO12	
	ADC2_CH6	IO14	
	ADC2_CH7	IO27	
	ADC2_CH8	IO25	
	ADC2_CH9	IO26	
Ultra-Low Noise Analog Pre-Amplifier	SENSOR_VP	IO36	Provides about 60 dB gain by using larger capacitors on PCB
	SENSOR_VN	IO39	
DAC	DAC_1	IO25	Two 8-bit DACs
	DAC_2	IO26	
Touch Sensor	TOUCH0	IO4	Capacitive touch sensors
	TOUCH1	IO0	
	TOUCH2	IO2	
	TOUCH3	IO15	
	TOUCH4	IO13	
	TOUCH5	IO12	
	TOUCH6	IO14	
	TOUCH7	IO27	
	TOUCH8	IO33	
	TOUCH9	IO32	
SD/SDIO/MMC Host Controller	HS2_CLK	MTMS	Supports SD memory card V3.01 standard
	HS2_CMD	MTDO	
	HS2_DATA0	IO2	
	HS2_DATA1	IO4	
	HS2_DATA2	MTDI	
	HS2_DATA3	MTCK	

Interface	Signal	Pin	Function
Motor PWM	PWM0_OUT0~2	Any GPIOs*	Three channels of 16-bit timers generate PWM waveforms. Each channel has a pair of output signals, three fault detection signals, three event-capture signals, and three sync signals.
	PWM1_OUT_IN0~2		
	PWM0_FLT_IN0~2		
	PWM1_FLT_IN0~2		
	PWM0_CAP_IN0~2		
	PWM1_CAP_IN0~2		
	PWM0_SYNC_IN0~2		
	PWM1_SYNC_IN0~2		
LED PWM	ledc_hs_sig_out0~7	Any GPIOs*	16 independent channels @80 MHz clock/RTC CLK. Duty accuracy: 16 bits.
	ledc_ls_sig_out0~7		
UART	U0RXD_in	Any GPIOs*	Two UART devices with hardware flow-control and DMA
	U0CTS_in		
	U0DSR_in		
	U0TXD_out		
	U0RTS_out		
	U0DTR_out		
	U1RXD_in		
	U1CTS_in		
	U1TXD_out		
	U1RTS_out		
	U2RXD_in		
	U2CTS_in		
	U2TXD_out		
	U2RTS_out		
I2C	I2CEXT0_SCL_in	Any GPIOs*	Two I2C devices in slave or master modes
	I2CEXT0_SDA_in		
	I2CEXT1_SCL_in		
	I2CEXT1_SDA_in		
	I2CEXT0_SCL_out		
	I2CEXT0_SDA_out		
	I2CEXT1_SCL_out		
	I2CEXT1_SDA_out		

Interface	Signal	Pin	Function
I2S	I2S0I_DATA_in0~15	Any GPIOs*	Stereo input and output from/to the audio codec, and parallel LCD data output
	I2S0O_BCK_in		
	I2S0O_WS_in		
	I2S0I_BCK_in		
	I2S0I_WS_in		
	I2S0I_H_SYNC		
	I2S0I_V_SYNC		
	I2S0I_H_ENABLE		
	I2S0O_BCK_out		
	I2S0O_WS_out		
	I2S0I_BCK_out		
	I2S0I_WS_out		
	I2S0O_DATA_out0~23		
	I2S1I_DATA_in0~15		
	I2S1O_BCK_in		
	I2S1O_WS_in		
	I2S1I_BCK_in		
	I2S1I_WS_in		
	I2S1I_H_SYNC		
	I2S1I_V_SYNC		
	I2S1I_H_ENABLE		
	I2S1O_BCK_out		
	I2S1O_WS_out		
	I2S1I_BCK_out		
	I2S1I_WS_out		
	I2S1O_DATA_out0~23		
Remote Controller	RMT_SIG_IN0~7	Any GPIOs*	Eight channels of IR transmitter and receiver for various waveforms
	RMT_SIG_OUT0~7		

Interface	Signal	Pin	Function
Parallel QSPI	SPIHD	SHD/SD2	Supports Standard SPI, Dual SPI, and Quad SPI that can be connected to the external flash and SRAM
	SPIWP	SWP/SD3	
	SPICS0	SCS/CMD	
	SPICLK	SCK/CLK	
	SPIQ	SDO/SD0	
	SPID	SDI/SD1	
	HSPICLK	IO14	
	HSPICS0	IO15	
	HSPIQ	IO12	
	HSPID	IO13	
	HSPIHD	IO4	
	HSPIWP	IO2	
	VSPICLK	IO18	
	VSPICS0	IO5	
	VSPIQ	IO19	
	VSPID	IO23	
	VSPIHD	IO21	
	VSPIWP	IO22	
General Purpose SPI	HSPIQ_in/_out	Any GPIOs*	Standard SPI consists of clock, chip-select, MOSI and MISO. These SPIs can be connected to LCD and other external devices. They support the following features: • both master and slave modes; • 4 sub-modes of the SPI format transfer that depend on the clock phase (CPHA) and clock polarity (CPOL) control; • configurable SPI frequency; • up to 64 bytes of FIFO and DMA.
	HSPID_in/_out		
	HSPICLK_in/_out		
	HSPI_CS0_in/_out		
	HSPI_CS1_out		
	HSPI_CS2_out		
	VSPIQ_in/_out		
	VSPID_in/_out		
	VSPICLK_in/_out		
	VSPI_CS0_in/_out		
	VSPI_CS1_out		
	VSPI_CS2_out		
JTAG	MTDI	IO12	JTAG for software debugging
	MTCK	IO13	
	MTMS	IO14	
	MTDO	IO15	

Interface	Signal	Pin	Function
SDIO Slave	SD_CLK	IO6	SDIO interface that conforms to the industry standard SDIO 2.0 card specification.
	SD_CMD	IO11	
	SD_DATA0	IO7	
	SD_DATA1	IO8	
	SD_DATA2	IO9	
	SD_DATA3	IO10	
EMAC	EMAC_TX_CLK	IO0	Ethernet MAC with MII/RMII interface
	EMAC_RX_CLK	IO5	
	EMAC_TX_EN	IO21	
	EMAC_TXD0	IO19	
	EMAC_TXD1	IO22	
	EMAC_TXD2	IO14	
	EMAC_TXD3	IO12	
	EMAC_RX_ER	IO13	
	EMAC_RX_DV	IO27	
	EMAC_RXD0	IO25	
	EMAC_RXD1	IO26	
	EMAC_RXD2	TXD0	
	EMAC_RXD3	IO15	
	EMAC_CLK_OUT	IO16	
	EMAC_CLK_OUT_180	IO17	
	EMAC_TX_ER	IO4	
	EMAC_MDC_out	Any GPIOs*	
	EMAC_MDI_in	Any GPIOs*	
	EMAC_MDO_out	Any GPIOs*	
	EMAC_CRS_out	Any GPIOs*	
	EMAC_COL_out	Any GPIOs*	

Note:

- Functions of Motor PWM, LED PWM, UART, I2C, I2S, general purpose SPI and Remote Controller can be configured to any GPIO except GPIO6, GPIO7, GPIO8, GPIO9, GPIO10 and GPIO11.
- For the items marked with "Any GPIOs*" in the "Pin" column, users should note that GPIO6, GPIO7, GPIO8, GPIO9, GPIO10 and GPIO11 are connected to the integrated SPI flash of ESP-WROOM-32 and are not recommended for other uses.

5. Electrical Characteristics

Note:

The specifications in this chapter have been tested under the following general condition: VDD = 3.3V, T_A = 27°C, unless otherwise specified.

5.1 Absolute Maximum Ratings

Table 8: Absolute Maximum Ratings

Parameter	Symbol	Min	Typ	Max	Unit
Power supply	VDD	2.7	3.3	3.6	V
Minimum current delivered by power supply	I_{VDD}	0.5	-	-	A
Input low voltage	V_{IL}	-0.3	-	$0.25\times V_{IO}$[1]	V
Input high voltage	V_{IH}	$0.75\times V_{IO}$[1]	-	V_{IO}[1]+0.3	V
Input leakage current	I_{IL}	-	-	50	nA
Input pin capacitance	C_{pad}	-	-	2	pF
Output low voltage	V_{OL}	-	-	$0.1\times V_{IO}$[1]	V
Output high voltage	V_{OH}	$0.8\times V_{IO}$[1]	-	-	V
Maximum output drive capability	I_{MAX}	-	-	40	mA
Storage temperature range	T_{STR}	-40	-	85	°C
Operating temperature range	T_{OPR}	-40	-	85	°C

1. V_{IO} is the power supply for a specific pad. More details can be found in the ESP32 Datasheet, Appendix IO_MUX. For example, the power supply for SD_CLK is the VDD_SDIO.

5.2 Wi-Fi Radio

Table 9: Wi-Fi Radio Characteristics

Description	Min	Typical	Max	Unit
Input frequency	2412	-	2484	MHz
Output impedance	-	30+j10	-	Ω
Input reflection	-	-	-10	dB
Tx power				
Output power of PA for 72.2 Mbps	13	14	15	dBm
Output power of PA for 11b mode	19.5	20	20.5	dBm
Sensitivity				
DSSS, 1 Mbps	-	-98	-	dBm
CCK, 11 Mbps	-	-91	-	dBm
OFDM, 6 Mbps	-	-93	-	dBm
OFDM, 54 Mbps	-	-75	-	dBm
HT20, MCS0	-	-93	-	dBm

Description	Min	Typical	Max	Unit
HT20, MCS7	-	-73	-	dBm
HT40, MCS0	-	-90	-	dBm
HT40, MCS7	-	-70	-	dBm
MCS32	-	-89	-	dBm
Adjacent channel rejection				
OFDM, 6 Mbps	-	37	-	dB
OFDM, 54 Mbps	-	21	-	dB
HT20, MCS0	-	37	-	dB
HT20, MCS7	-	20	-	dB

5.3 BLE Radio

5.3.1 Receiver

Table 10: Receiver Characteristics — BLE

Parameter	Conditions	Min	Typ	Max	Unit
Sensitivity @30.8% PER	-	-	-97	-	dBm
Maximum received signal @30.8% PER	-	0	-	-	dBm
Co-channel C/I	-	-	+10	-	dB
Adjacent channel selectivity C/I	F = F0 + 1 MHz	-	-5	-	dB
	F = F0 - 1 MHz	-	-5	-	dB
	F = F0 + 2 MHz	-	-25	-	dB
	F = F0 - 2 MHz	-	-35	-	dB
	F = F0 + 3 MHz	-	-25	-	dB
	F = F0 - 3 MHz	-	-45	-	dB
Out-of-band blocking performance	30 MHz ~ 2000 MHz	-10	-	-	dBm
	2000 MHz ~ 2400 MHz	-27	-	-	dBm
	2500 MHz ~ 3000 MHz	-27	-	-	dBm
	3000 MHz ~ 12.5 GHz	-10	-	-	dBm
Intermodulation	-	-36	-	-	dBm

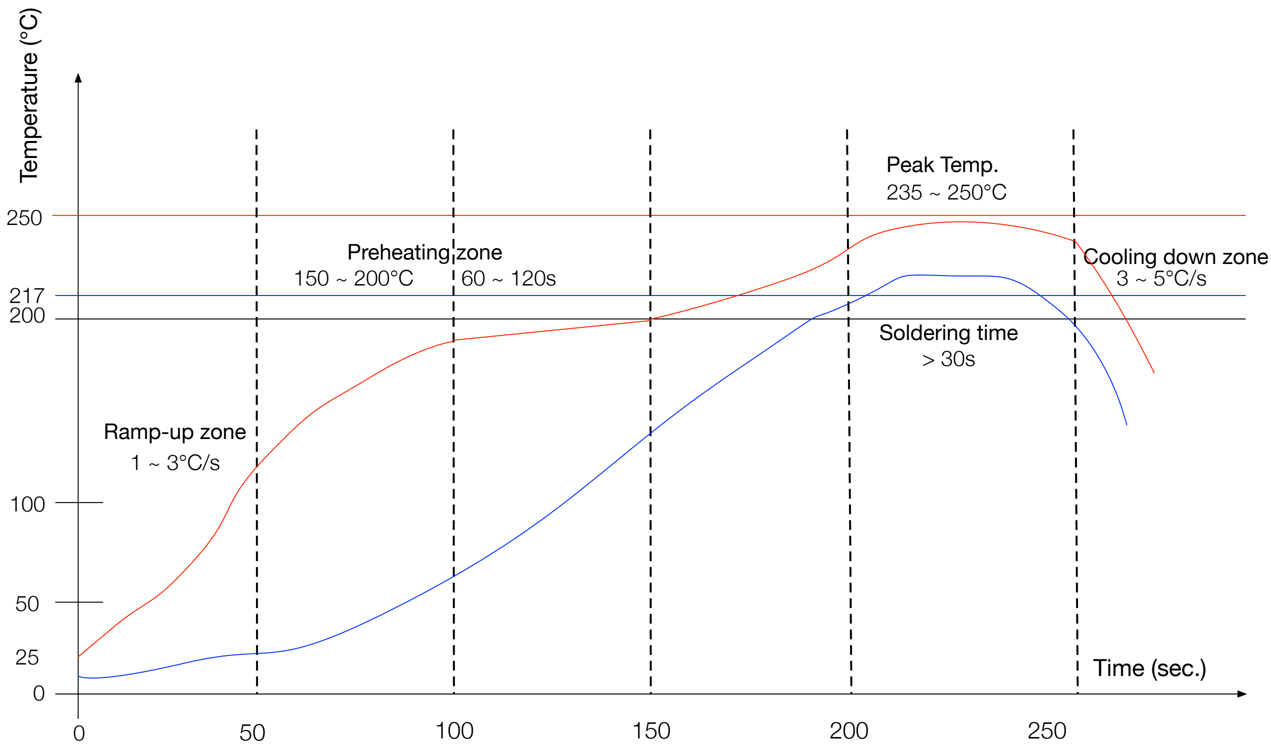
5.3.2 Transmitter

Table 11: Transmitter Characteristics — BLE

Parameter	Conditions	Min	Typ	Max	Unit
RF transmit power	-	-	0	-	dBm
Gain control step	-	-	±3	-	dBm
RF power control range	-	-12	-	+12	dBm

Parameter	Conditions	Min	Typ	Max	Unit
Adjacent channel transmit power	F = F0 + 1 MHz	-	-14.6	-	dBm
	F = F0 - 1 MHz	-	-12.7	-	dBm
	F = F0 + 2 MHz	-	-44.3	-	dBm
	F = F0 - 2 MHz	-	-38.7	-	dBm
	F = F0 + 3 MHz	-	-49.2	-	dBm
	F = F0 - 3 MHz	-	-44.7	-	dBm
	F = F0 + > 3 MHz F = F0 - > 3 MHz	- -	-50 -50	- -	dBm dBm
$\Delta f1_{avg}$	-	-	-	265	kHz
$\Delta f2_{max}$	-	247	-	-	kHz
$\Delta f2_{avg}/\Delta f1_{avg}$	-	-	-0.92	-	-
ICFT	-	-	-10	-	kHz
Drift rate	-	-	0.7	-	kHz/50 μs
Drift	-	-	2	-	kHz

5.4 Reflow Profile



Ramp-up zone — Temp.: <150°C Time: 60 ~ 90s Ramp-up rate: 1 ~ 3°C/s
Preheating zone — Temp.: 150 ~ 200°C Time: 60 ~ 120s Ramp-up rate: 0.3 ~ 0.8°C/s
Reflow soldering zone — Peak Temp.: 235 ~ 250°C (<245°C recommended) Time: 30 ~ 70s
Cooling down zone — Temp.: 217 ~ 170°C Ramp-down rate: 3 ~ 5°C/s
Solder — Sn&Ag&Cu Lead-free solder (SAC305)

Figure 2: Reflow Profile

6. Schematics

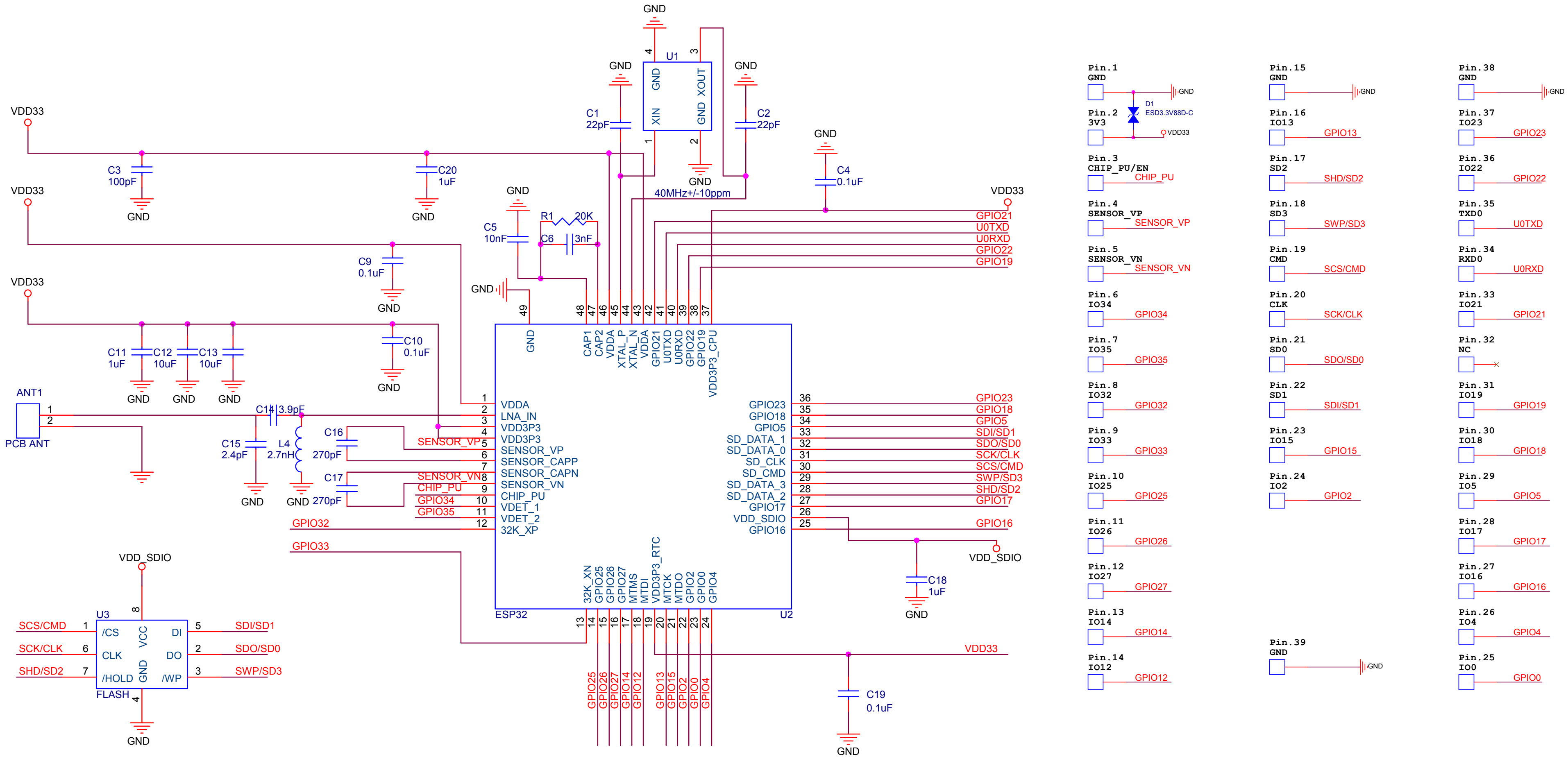
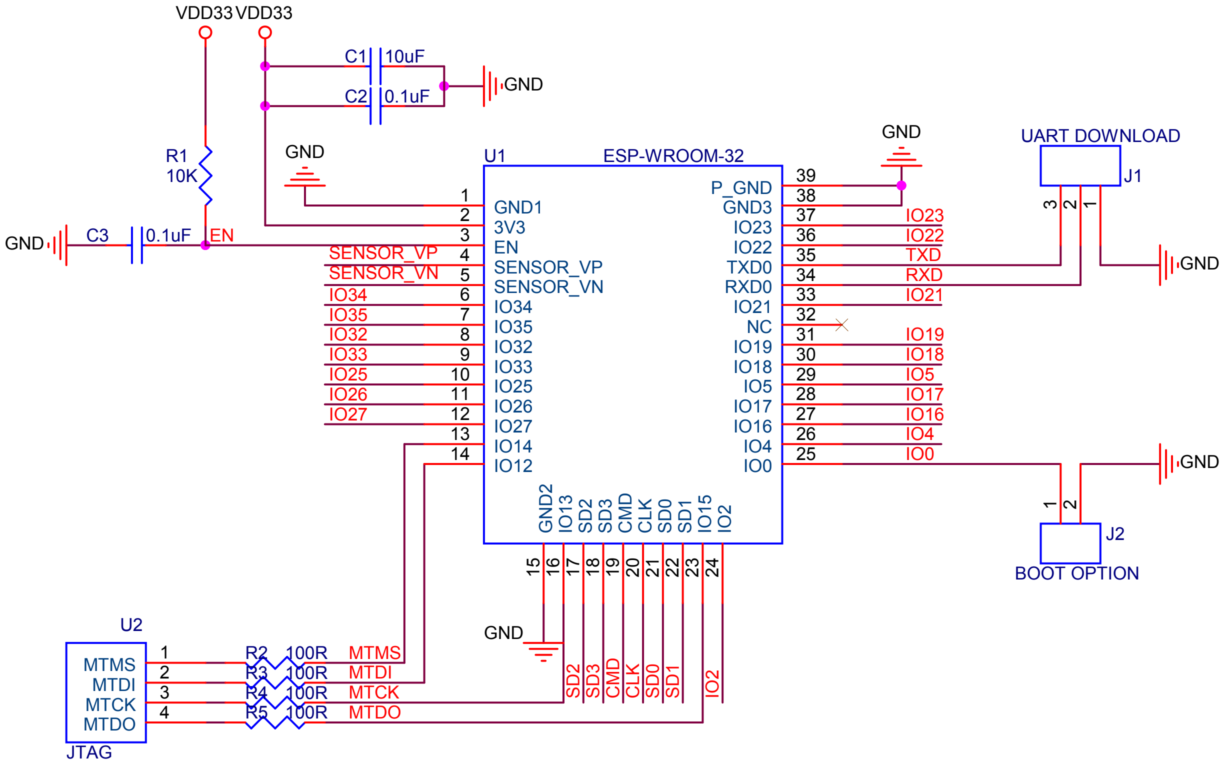


Figure 3: ESP-WROOM-32 Schematics

7. Peripheral Schematics



MTDI should be kept at low electric level when powering up.

Figure 4: ESP-WROOM-32 Peripheral Schematics

> **Note:**
>
> It is recommended that users do not solder Pad 39 to the base board.

8. Dimensions

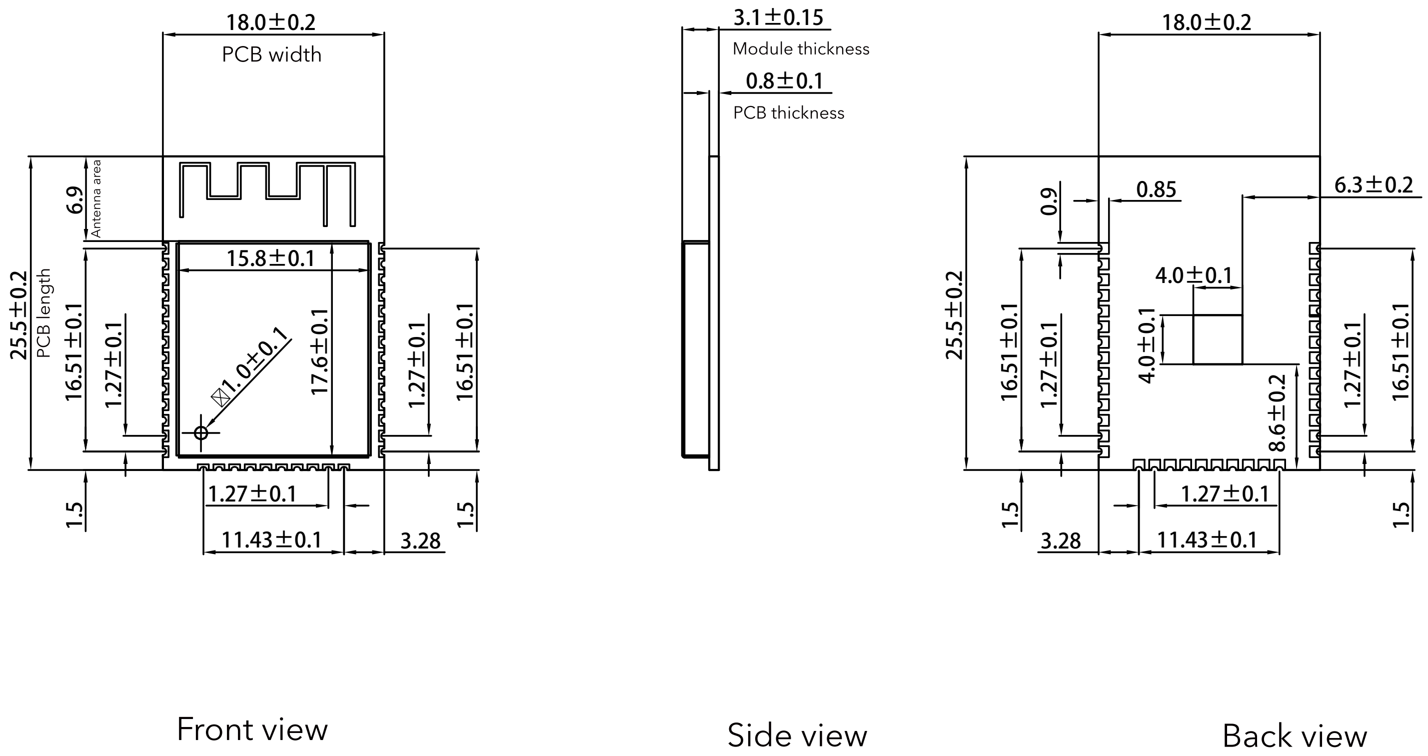


Figure 5: Dimensions of ESP-WROOM-32

> **Note:**
> All dimensions are in millimeters.

9. Learning Resources

9.1 Must-Read Documents

The following link provides documents related to ESP32.

- ESP32 Datasheet
  This document provides an introduction to the specifications of the ESP32 hardware, including overview, pin definitions, functional description, peripheral interface, electrical characteristics, etc.
- ESP32 Technical Reference Manual
  The manual provides detailed information on how to use the ESP32 memory and peripherals.
- ESP32 Hardware Resources
  The zip files include the schematics, PCB layout, Gerber and BOM list of ESP32 modules and development boards.
- ESP32 Hardware Design Guidelines
  The guidelines outline recommended design practices when developing standalone or add-on systems based on the ESP32 series of products, including ESP32, the ESP-WROOM-32 module, and ESP32-DevKitC—the development board.
- ESP32 AT Instruction Set and Examples
  This document introduces the ESP32 AT commands, explains how to use them, and provides examples of several common AT commands.

9.2 Must-Have Resources

Here are the ESP32-related must-have resources.

- ESP32 BBS
  This is an Engineer-to-Engineer (E2E) Community for ESP32 where you can post questions, share knowledge, explore ideas, and help solve problems with fellow engineers.
- ESP32 Github
  ESP32 development projects are freely distributed under Espressif's MIT license on Github. It is established to help developers get started with ESP32 and foster innovation and the growth of general knowledge about the hardware and software surrounding ESP32 devices.
- ESP32 Tools
  This is a webpage where users can download ESP32 Flash Download Tools and the zip file "ESP32 Certification and Test".
- ESP32 IDF
  This webpage links users to the official IoT development framework for ESP32.
- ESP32 Resources
  This webpage provides the links to all available ESP32 documents, SDK and tools.

Ms Preetha D'souza

Preetha_Veeksha

SEE_Major Project_Batch 2025

Plagiarism Report

St Joseph Engineering College, Mangalore

Document Details

Submission ID

trn:oid:::1:3250499897

Submission Date

May 15, 2025, 9:33 AM GMT+5:30

Download Date

May 15, 2025, 10:04 AM GMT+5:30

File Name

final_report_format_.docx.pdf

File Size

3.9 MB

41 Pages

6,165 Words

39,229 Characters

21% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

72 Not Cited or Quoted 20%
Matches with neither in-text citation nor quotation marks

1 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

17% Internet sources
14% Publications
12% Submitted works (Student Papers)

Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author:	Ms Preetha D'souza
Assignment title:	SEE_Major Project_Batch 2025
Submission title:	Preetha_Veeksha
File name:	final_report_format_.docx.pdf
File size:	3.9M
Page count:	41
Word count:	6,165
Character count:	39,229
Submission date:	15-May-2025 09:33AM (UTC+0530)
Submission ID:	2676303398

ABSTRACT

This project presents a real-time stress detection and management system that aims to promote mental wellness.

Existing research in stress detection often relies on isolated physiological signals or facial analysis. However, the combination of both has shown potential for more reliable results.

Our system integrates facial emotion detection using Convolutional Neural Networks (CNNs) with Electrodermal Activity (EDA) sensors, interfaced through an ESP32. The data is sent to Firebase and visualized via a React-based web application offering remedies like music and relaxation exercises.

The results confirm improved stress identification through multimodal input.

Future work includes wearable integration and expanded therapeutic interventions.