IOT-BASED SMART WOMEN SAFETY SYSTEM

A PROJECT REPORT

Submitted by

NISHA M [20L234]

PRIYANKAA R [20L239]

REVATHI M [20L243]

SANGAVI S [20L244]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

KARPAGAM COLLEGE OF ENGINEERING

COIMBATORE – 641 032

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2024

KARPAGAM COLLEGE OF ENGINEERING

COIMBATORE – 641 032

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"IOT-BASED SMART WOMEN SAFETY SYSTEM"** is the bonafide work of **NISHA M [20L234], PRIYANKAA R [20L239], REVATHI M [20L243], SANGAVI S [20L244]** who carried out the project work under my supervision.

SIGNATURE
Dr.R.Sarankumar., M.E., Ph.D.,
HEAD OF THE DEPARTMENT
Electronics and Communication Engineering
Karpagam College of Engineering
Coimbatore – 641 032

SIGNATURE
Mr.G.Rajarathinam., M.E.,
SUPERVISOR
Associate Professor
Electronics and Communication Engineering
Karpagam College of Engineering
Coimbatore – 641 032

Certified that the candidate was examined during the viva voce examinations held on ____________________

Signature of the Internal Examiner with date

Signature of External Examiner with date

DECLARATION

We hereby declare that this Project report entitled "**IOT-BASED SMART WOMEN SAFETY SYSTEM**" submitted by us for the degree of **BE in Electronics and Communication Engineering at Karpagam College of Engineering, Coimbatore** is the record of original work done by us under the guidance and supervision of **Mr.G.RAJARATHINAM.,M.E., Associate Professor** at the Department of Electronics and Communication Engineering, Karpagam College of Engineering, Coimbatore – 641032 and has not formed the basis for the award of any degree, or diploma or titles in this institution or any other Institution of higher learning.

Date:

Place: Coimbatore

Name and Signature of the Candidates

NISHA M	[20L234]
PRIYANKAA R	[20L239]
REVATHI M	[20L243]
SANGAVI S	[20L244]

ACKNOWLEDGEMENT

First and foremost, praises and thanks to the almighty for her showers and blessings throughout our project work to complete it successfully.

We extend our gratitude to the Management of Karpagam College of Engineering, Coimbatore for the excellent infrastructure and support facilities to undergo the project work.

We a are very grateful to **Dr. V. KUMAR CHINNAIYAN**, the Principal and **Dr.R. SARANKUMAR**, the HoD, **Electronics and Communication Engineering** for provided the facilities, support, and permission to carried out our project work at our esteemed institution.

We record my sincere gratitude to our Project Coordinator **Dr.S.SOBANA** for giving inputs, encouragement for the continuous improvement during the progress and to complete this project work.

We would like to express our sincere gratitude to our Supervisor **Mr.G.RAJARATHINAM** for the continuous support for our UG study, for his motivation and adequate guidance which helped us to achieve success in all our accomplishments and to complete this project work.

We also thank all the teaching faculty members and non-teaching Staff members of the department of **Electronics and Communication Engineering**, Karpagam College of Engineering, Coimbatore for their kindness and support.

I would like to thank **our parents, family members and friends** who sacrificed their time and energy to complete the project work successfully.

NISHA M	**[20L234]**
PRIYANKAA R	**[20L239]**
REVATHI M	**[20L243]**
SANGAVI S	**[20L244]**

ABSTRACT

The "IoT Based Smart Women Safety System" is an innovative and comprehensive project that addresses the pressing issue of women's safety in today's world. It incorporates advanced technologies to create a holistic safety solution for women, ensuring their well-being in various situations. By harnessing the capabilities of the Internet of Things (IoT), this system offers a multi-faceted approach to women's security, providing real-time monitoring, instant alerts, and timely assistance. The core components of this project include wearable devices that are discreet, comfortable, and equipped with a range of sensors. These sensors track vital parameters such as heart rate, location, and movement patterns. These wearables connect seamlessly to a centralized IoT platform, where data is processed and analyzed in real-time. One of the standout features of this system is its ability to automatically trigger alerts when it detects anomalies or signs of distress. Machine learning algorithms are employed to continuously refine the system's ability to recognize patterns indicative of a dangerous situation. Moreover, users have the option to manually trigger distress signals, allowing for immediate response from authorities or pre-designated contacts. The IoT network that connects these devices to the central platform is designed with security in mind, ensuring that data is transmitted in an encrypted and protected manner. The central platform itself is cloud-based, enabling scalable and efficient data processing. Additionally, a user-friendly mobile application serves as the interface for users to interact with the system, check their safety status, and request assistance if needed. In summary, the "IoT Based Smart Women Safety System" is a groundbreaking project that marries technology with a social cause, aiming to ensure the safety and security of women while fostering a more inclusive and equal society.

TABLE OF CONTENTS

APPENDIX I

APPENDIX II

APPENDIX III

APPENDIX IV

LIST OF FIGURES

LIST OF ABBREVIATIONS

S NO	ABBREVIATIONS	EXPANSIONS
1	GSM	Global System for Mobile Communication
2	LCD	Liquid Crystal Display
3	IOT	Internet Of Things
4	GPS	Global Positioning System
5	SOS	Save Our Ship
6	SMS	Short Message Service
7	UART	Universal Asynchronous Receiver Transmitter

CHAPTER -1

INTRODUCTION

The "IoT Based Smart Women Safety System" represents a visionary response to the pressing issue of women's safety in an increasingly complex and interconnected world. Women's security and well-being have been a longstanding concern, and in this era of technological advancements, it is only fitting that we harness the capabilities of the Internet of Things (IoT) to create a robust and comprehensive solution. This project embodies a commitment to empowerment, security, and equality, addressing the multifaceted challenges faced by women in various scenarios. In a world where personal safety is a paramount concern, this project seeks to provide women with a reliable and adaptable safety net. The Internet of Things, characterized by the interconnection of devices and the exchange of data, offers an ideal platform to build a system that can actively monitor, analyze, and respond to women's safety needs in real-time.

Fig 1.1 Women Harrassment

At its core, the "IoT Based Smart Women Safety System" revolves around wearable devices that blend seamlessly into a woman's daily life. These wearables are designed to be discreet, comfortable, and unobtrusive, making them an integral part of a woman's attire. Embedded with an array of sensors, these devices track vital parameters, such as heart rate, location, and movement patterns. Equally important, they possess the ability to detect and respond to unusual activities or signs of distress. The data collected from these wearables is transmitted securely to a centralized IoT platform, where it is processed and analyzed. This platform is a crucible of innovation, employing machine learning algorithms to continuously enhance its capacity to recognize patterns of distress or emergency. The result is a system that becomes more astute and responsive with time.

Moreover, the "IoT Based Smart Women Safety System" offers two modes of operation. In the first, it autonomously triggers alerts when it identifies anomalies or threats to a woman's safety. This proactive approach ensures that women are safeguarded even in situations where they might not be able to call for help themselves. In the second, users have the ability to manually trigger distress signals, thus enabling swift responses from authorities or pre-designated contacts. The project also places a significant emphasis on data security, ensuring that all information is transmitted in an encrypted and protected manner. The central platform is hosted in the cloud, allowing for scalability and efficient data processing. A user-friendly mobile application serves as the interface, providing users with a simple yet powerful tool to interact with the system, check their safety status, and request assistance when needed.

In conclusion, the "IoT Based Smart Women Safety System" is not just a technological marvel; it is a testament to our collective commitment to the safety, empowerment, and equality of women. It stands as a beacon of hope in a world where women often face uncertainty, fear, and danger. This project symbolizes the fusion of technology with a profound social cause, aiming to give women the confidence to navigate their lives without trepidation and to provide a rapid, coordinated response in times of crisis. It is a bold step towards a more secure, inclusive, and equal world for all.

CHAPTER-2

LITERATURE SURVEY

In Paper **[1]** such device is designed which is a portable one which can be activated as per the requirement of the individual which will locate the victim using GPS and with the help of GSM emergency messages can be sent to the respective locations as per the design. The gadget provides an alarm system, call for help, and electric shock to get rid of the attacker.

In Paper **[2]** suggests a new perspective to use technology to protect women. The system contains a normal belt which when gets activated, tracks the location of the victim using GPS (Global Positioning System) and sends messages.

In Paper **[3]** describes a GPS and GSM based vehicle tracking and women employee security system that provides the combination of GPS device and specialized software to track the location of the vehicle as well as provide messages and alerts with an emergency button trigger. The information of vehicle position provided by the device can be viewed on Google maps app.

In Paper **[4]** proposed system with the push of one button, people can alert selected contacts that the person is in danger and share the location. With this personal safety app, women never walk alone. The personal safety application needs the name and number of the person who is to be contacted in times of emergency.

In Paper **[5]** the information of vehicle position provided by the device can be viewed on Google maps app. Emergency messages can be sent to the respective locations as per the design. The gadget provides an alarm system, call for help, and electric shock to get rid of the attacker.

In Paper **[6]** system provides slight contribution. It can be inserted or fitted in the jacket also. It tracks the location of the victim using GPS (Global Positioning System) and sends messages.

In Paper **[7]** by pressing the switch, the entire system will be activated to track the location of the vehicle as well as provide messages and alerts with an emergency button trigger. The information of vehicle position provided by the device can be viewed on Google maps app.

In Paper **[8]** the idea to develop a smart system for women is completely comfortable. With this personal safety app, women never walk alone. The personal safety application needs the name and number of the person who is to be contacted in times of emergency.

In Paper **[9]** it can send messages to SOS with instant location. The gadget provides an alarm system, call for help, and electric shock to get rid of the attacker.

In Paper **[10]** it provides smart alert system by pressing the panic button, immediately it sends alert to SOS / family members contacts.

In Paper **[11]** proposed system with the push of one button, people can alert selected contacts that the person is in danger and share the location. With this personal safety app, women never walk alone. The personal safety application needs the name and number of the person who is to be contacted in times of emergency.

In Paper **[12]** which will locate the victim using GPS and with the help of GSM emergency messages can be sent to the respective locations as per the design. The gadget provides an alarm system, call for help, and electric shock to get rid of the attacker.

CHAPTER-3

EXISTING SYSTEM

The most of the existing systems are implemented using GSM, and GPS. But the main disadvantage of those systems is that ladies, girls should have to carry mobile phone with them while travelling. Hence there is need of a novel system which can overcome the above limitations. There is thus, a need of simpler safety solution that can be activated as simply as by pressing a button and can immediately send out alerts to the near ones of the sufferer. This project focuses on a security system that is designed exclusively to serve the purpose of providing security and safety to women so that they never feel weak while facing such social challenges. The GSM and GPS Module is interfaced with this raspberry pi pico to track the location and to send SMS alert to SOS/selected contact. A security solution that creates a sense of safety among women needs to be developed. In instances of attack, it is largely reported that women's are immobilized. If the panic button is pressed, the GPS and GSM will activate and send messages to SOS/selected contact. This new system is interfaced with panic button, GPS module and GSM module is there which is interfaced with the raspberry pi.

CHAPTER-4

PROPOSED SYSTEM

The Main aim of the project is a new system that is built using raspberry pi pico, GSM & GPS Module and IoT. This device provides quick response to those who are under threat. In order to overcome various crimes against women this system provides significant contribution. During emergency, she can press the panic button in which the current location of the victim will be sent to the number that are already given in the system. If the panic button is pressed, the GPS and GSM will activate and send alert messages to SOS/selected contact and also send mail alerts to selected mail id. This new system is interfaced with panic button, GPS & GSM module and IoT.

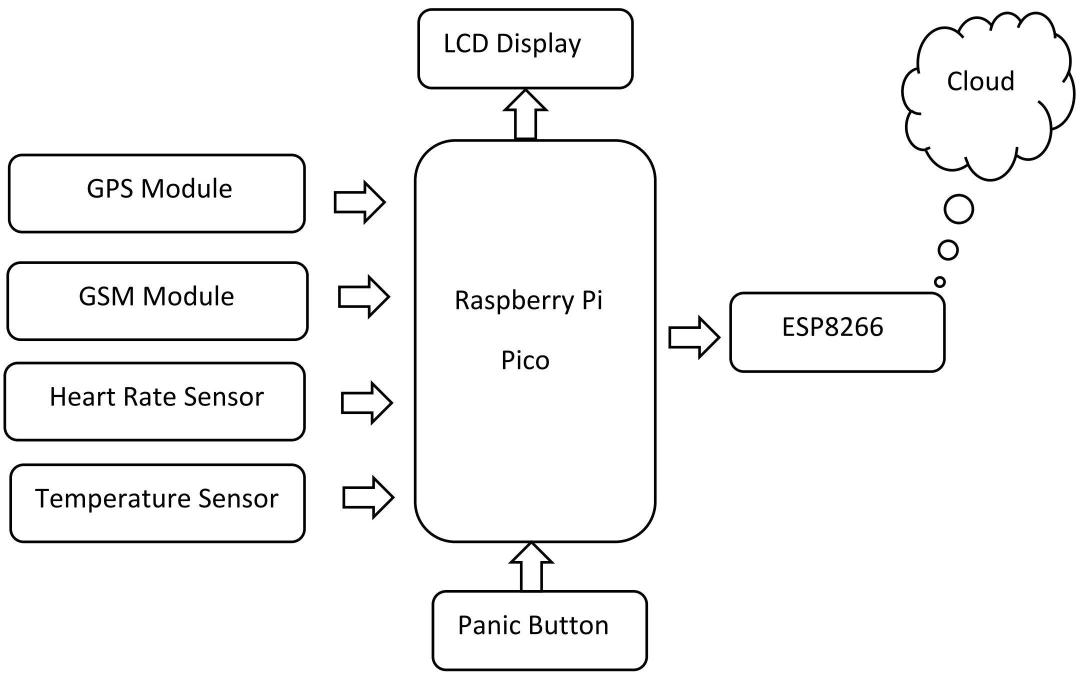

Fig 4.1 Block diagram of the Proposed method

SOFTWARE: Thonny IDE, Arduino IDE, Ubidots STEM

HARDWARE: Raspberry Pi Pico, Heart Rate Sensor, Dallas Temperature Sensor, Panic button, GSM Module, GPS Module, ESP8266 WiFi Module and LCD.

4.1 CIRCUIT DIAGRAM

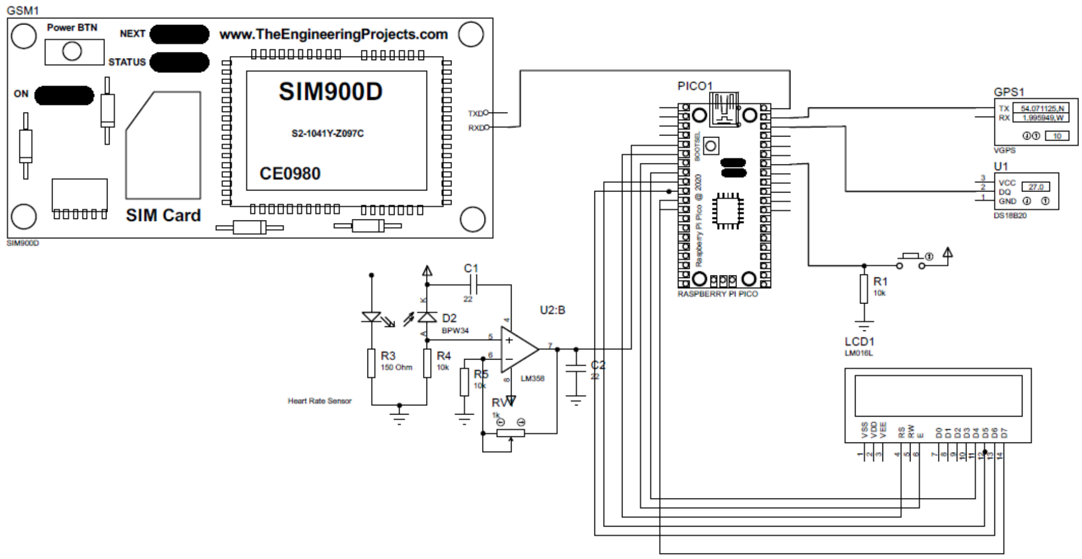

Fig 4.2 Circuit diagram

CHAPTER 5

COMPONENTS USED

5.1 RASPBERRY PI PICO

The Raspberry Pi Pico, developed by the Raspberry Pi Foundation, is a cost-effective and versatile microcontroller board tailored for embedded and microcontroller applications. At its core is the powerful Raspberry Pi RP2040 microcontroller, boasting a dual-core ARM Cortex-M0+ processor clocked at 133MHz. This microcontroller comes equipped with 264KB of RAM and 2MB of on-board flash memory, offering ample resources for a variety of projects. One of the standout features of the Pico is its 26 multi-function GPIO pins, which enable a wide range of tasks, including digital input/output, PWM output, SPI, I2C, UART, and more. While it lacks built-in wireless connectivity like Wi-Fi or Bluetooth, the Pico's compact form factor and low price point make it an excellent choice for cost-sensitive projects.

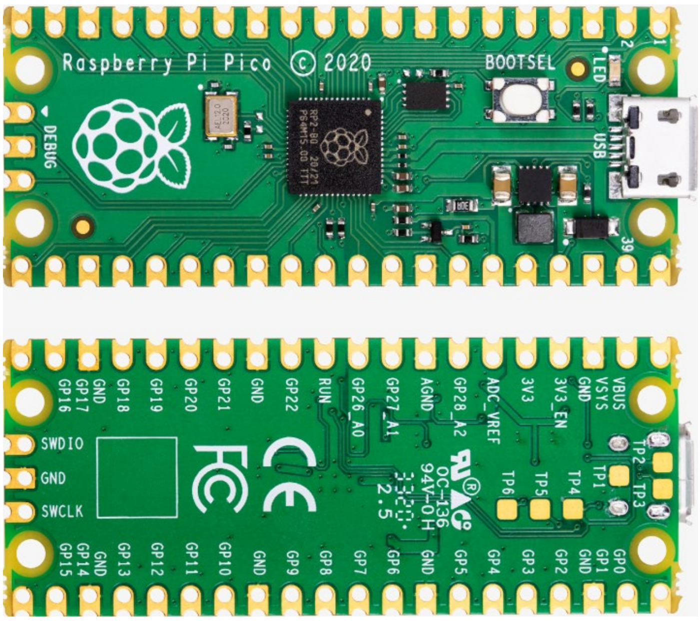

Fig 5.1 Raspberry Pi Pico

Programming the Pico is accessible with support for popular languages like MicroPython, C/C++, and CircuitPython. Moreover, the Raspberry Pi Foundation provides extensive documentation and resources, and a vibrant online community offers tutorials and projects to assist users in harnessing the Pico's capabilities. Whether for robotics, home automation, IoT devices, or sensor-based projects, the Raspberry Pi Pico offers an affordable and powerful platform for makers and developers alike.

PIN DESCRIPTION

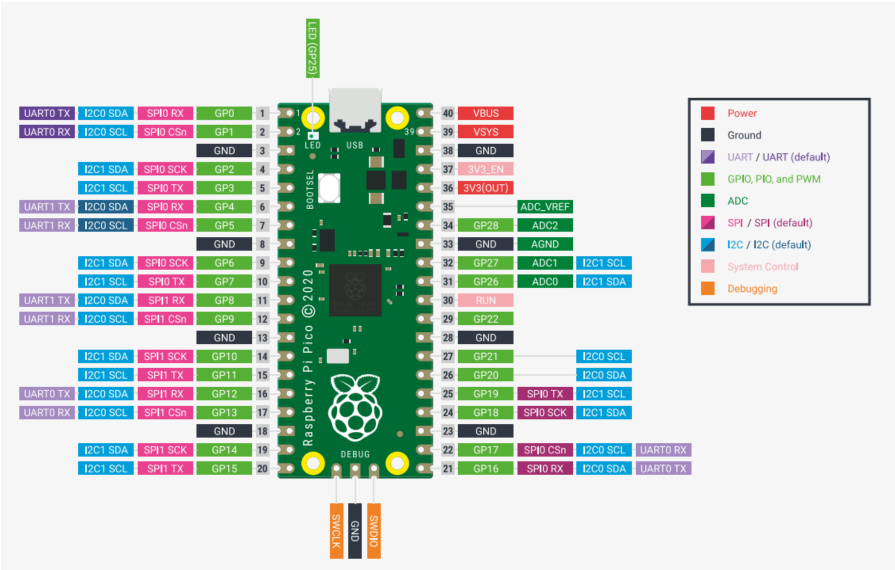

Fig 5.2 Pin Description

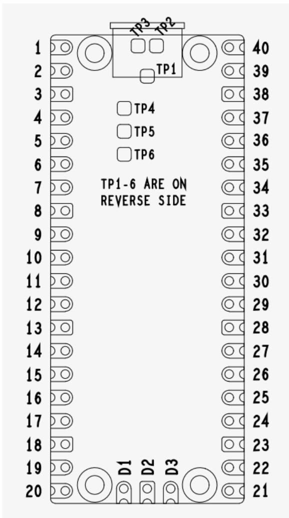

Fig 5.3 Pinout for Raspberry Pi Pico

GPIO0 (GP0, GP28, ADC0): General-purpose I/O pin. It can also be used as an ADC (Analog to Digital Converter) input labeled as ADC0.

GPIO1 (GP1, GP27, ADC1): General-purpose I/O pin. It can also be used as an ADC input labeled as ADC1.

GPIO2 (GP2, GP26, ADC2): General-purpose I/O pin. It can also be used as an ADC input labeled as ADC2.

GPIO3 (GP3, GP25, ADC3): General-purpose I/O pin. It is connected to the onboard user LED.

GPIO4 (GP4, GP24): General-purpose I/O pin. It is used for VBUS sensing (high if VBUS is present, else low).

GPIO5 (GP5, GP23): General-purpose I/O pin. It controls the on-board SMPS Power Save pin.

GPIO6 (GP6, GP22): General-purpose I/O pin.

GPIO7 (GP7, GP21): General-purpose I/O pin.

GPIO8 (GP8, GP20): General-purpose I/O pin.

GPIO9 (GP9, GP19): General-purpose I/O pin.

GPIO10 (GP10, GP18): General-purpose I/O pin.

GPIO11 (GP11, GP17): General-purpose I/O pin.

GPIO12 (GP12, GP16): General-purpose I/O pin.

GPIO13 (GP13, GP15): General-purpose I/O pin.

GPIO14 (GP14, GP14): General-purpose I/O pin.

GPIO15 (GP15, GP13): General-purpose I/O pin.

GPIO16 (GP16, GP12): General-purpose I/O pin.

GPIO17 (GP17, GP11): General-purpose I/O pin.

GPIO18 (GP18, GP10): General-purpose I/O pin.

GPIO19 (GP19, GP9): General-purpose I/O pin.

GPIO20 (GP20, GP8): General-purpose I/O pin.

GPIO21 (GP21, GP7): General-purpose I/O pin.

GPIO22 (GP22, GP6): General-purpose I/O pin.

GPIO26 (GP26, GP2, ADC4): General-purpose I/O pin. It can also be used as an ADC input labeled as ADC4.

GPIO27 (GP27, GP1, ADC5): General-purpose I/O pin. It can also be used as an ADC input labeled as ADC5.

GPIO28 (GP28, GP0, ADC6): General-purpose I/O pin. It can also be used as an ADC input labeled as ADC6.

GPIO29 (GP29, GP3, ADC7): General-purpose I/O pin. It can also be used as an ADC input labeled as ADC7 to measure VSYS/3.

GPIO30 (RUN): RP2040 enable pin with an internal pull-up resistor (~50kΩ). Short this pin low to reset RP2040.

GPIO31 (BOOTSEL): Used for boot mode selection.

3V3_EN: Connects to the onboard SMPS enable pin and is pulled high (to VSYS) via a 100kΩ resistor. Short this pin low to disable the 3.3V supply.

3V3: The main 3.3V supply to RP2040 and its I/O, generated by the onboard SMPS. Can be used to power external circuitry (maximum output current depends on RP2040 load and VSYS voltage).

ADC_VREF: ADC power supply and reference voltage, generated on Pico by filtering the 3.3V supply. Can be used with an external reference if better ADC performance is required.

AGND: Ground reference for GPIO26-29. It is used when the ADC is in use or when ADC performance is critical.

VSYS: The main system input voltage, ranging from 1.8V to 5.5V, used by the onboard SMPS to generate the 3.3V supply for the RP2040 and its GPIO.

VBUS: Micro-USB input voltage, typically 5V when connected to a powered USB source.

I2C Pins (I2C0):

GPIO0 (GP0, GP28, ADC0): General-purpose I/O pin. It can function as the SDA (Serial Data) line for I2C communication.

GPIO1 (GP1, GP27, ADC1): General-purpose I/O pin. It can function as the SCL (Serial Clock) line for I2C communication.

UART Pins (UART0):

GPIO4 (GP4, GP24): General-purpose I/O pin. It can be used as a UART0 TX (Transmit) pin.

GPIO5 (GP5, GP23): General-purpose I/O pin. It can be used as a UART0 RX (Receive) pin.

SPI Pins (SPI0):

GPIO2 (GP2, GP26, ADC2): General-purpose I/O pin. It can function as the SPI0 SCK (Serial Clock) pin.

GPIO3 (GP3, GP25, ADC3): General-purpose I/O pin. It can function as the SPI0 MOSI (Master Out Slave In) pin.

GPIO7 (GP7, GP21): General-purpose I/O pin. It can function as the SPI0 MISO (Master In Slave Out) pin.

GPIO8 (GP8, GP20): General-purpose I/O pin. It can be used as a user-defined SPI0 CS (Chip Select) pin.

RESET PIN (RUN):

GPIO30 (RUN): The RUN pin is the RP2040 enable pin with an internal pull-up resistor of approximately ~50kΩ. To reset the RP2040, you can short this pin low by connecting it to GND (ground).

ADDITIONAL PINS ON THE MAIN 40-PIN INTERFACE:

- **VBUS:** Micro-USB input voltage, typically 5V when connected to a powered USB source.
- **VSYS:** The main system input voltage, which can range from 1.8V to 5.5V. It's used by the onboard SMPS to generate the 3.3V supply for the RP2040 and its GPIO.
- **3V3_EN:** This pin connects to the onboard SMPS enable pin. It's pulled high (towards VSYS) via a 100kΩ resistor. To disable the 3.3V supply (which also depowers the RP2040), short this pin to ground.
- **3V3:** The main 3.3V supply to the RP2040 and its I/O, generated by the onboard SMPS. This pin can also be used to power external circuitry, although the maximum output current will depend on the RP2040 load and VSYS voltage. It's recommended to keep the load on this pin below 300mA.
- **ADC_VREF:** This pin is the ADC power supply (and reference) voltage. It's generated on the Pico by filtering the 3.3V supply. If better ADC performance is required, an external reference can be used with this pin.

- **AGND:** AGND is the ground reference for GPIO26-29. There is a separate analog ground plane running under these signals and terminating at this pin. If the ADC is not used or ADC performance is not critical, this pin can be connected to the digital ground.
- **RUN:** RUN is the RP2040 enable pin, and it has an internal (on-chip) pull-up resistor to 3.3V, approximately ~50kΩ. To reset RP2040, you can short this pin to ground.

TEST POINTS (TP1-TP6):

- **TP1:** Ground (close-coupled ground for differential USB signals)
- **TP2:** USB DM (USB Data-)
- **TP3:** USB DP (USB Data+)
- **TP4:** GPIO23/SMPS PS pin (Caution: Do not use)
- **TP5:** GPIO25/LED (Not recommended for use)
- **TP6:** BOOTSEL (Used for boot mode selection)

TP1, TP2 and TP3 can be used to access the USB signals instead of using the micro-USB port. TP6 can be used to drive the system into mass-storage USB programming mode (by shorting it low at power-up). Note that TP4 is not intended to be used externally, and TP5 is not really recommended to be used as it will only swing from 0V to the LED forward voltage (and hence can only really be used as an output with special care).

MECHANICAL SPECIFICATIONS

Dimensions	21mm x 51mm
Form Factor	40-pin DIP-style PCB
Mounting	Can be surface mounted as a module

USB Port	Micro-USB B port for power, data, and reprogramming
GPIO Pins	23 GPIO pins are digital-only 3 GPIO pins are ADC
Debug Port	3-pin ARM Serial Wire Debug (SWD) port
Power Supply	powering via micro-USB, external supplies, or batteries
IO Voltage	Operates at 3.3V (Pico IO voltage is fixed at 3.3V)

ELECTRICAL SPECIFICATIONS

Input Voltage Range	4.5V to 5
Input Current	100mA
Digital Logic Levels	All GPIO pins operate at 3.3V logic levels
Current Sourcing Capability	16mA individually and up to 51mA total
Current Sinking Capability	Sink up to 16mA individually and up to 51mA total.
UART Logic Levels	UART interfaces also operate at 3.3V logic levels.
I2C Logic Levels	I2C interfaces use 3.3V logic levels.
SPI Logic Levels	SPI interfaces utilize 3.3V logic levels.
SWD Debug Port	3-pin ARM Serial Wire Debug (SWD)
Analog Input Range	The 12-bit ADC supports a voltage range of 0 to 3.3V
Operating Temperature	-20°C to 85°C.
External Flash Compatibility	Compatible with external Quad-SPI Flash

Clock Options	The RP2040 microcontroller's on-chip PLL

WORKING MODES & FUNCTIONS

NORMAL OPERATION MODE:

1. In normal operation mode, the Raspberry Pi Pico executes user-programmed code. This code can be written in programming languages like MicroPython, C/C++, or CircuitPython.

2. The Pico's dual-core ARM Cortex-M0+ processors, running at speeds of up to 133MHz, handle the execution of this code. This processing power allows the Pico to perform a wide range of tasks, from basic GPIO (General Purpose Input/Output) control to more complex applications, including IoT (Internet of Things) projects, robotics, and sensor-based systems.

PROGRAMMING MODE:

1. Programming mode is engaged when you want to load new firmware or programs onto the Raspberry Pi Pico.

2. To enter this mode, you typically hold down the BOOTSEL button (or another designated button) while connecting the Pico to a computer via USB. This action instructs the Pico to enter a state where it appears as a USB mass storage device on the computer.

3. Once in programming mode, you can simply drag and drop firmware files onto the Pico. This ease of programming makes the Pico accessible for both beginners and experienced developers.

DEBUGGING MODE:

1. Debugging mode enables you to inspect and debug code running on the Raspberry Pi Pico.

2. The Pico supports the SWD (Serial Wire Debug) protocol, allowing you to use debuggers like GDB (GNU Debugger) to connect to it.

3. Debugging is a valuable tool for identifying and fixing issues in your code. It allows you to set breakpoints, view variable values, and step through code execution for precise debugging.

LOW-POWER MODES:

1. The Raspberry Pi Pico offers low-power modes, which are crucial for conserving energy in battery-powered applications.

2. By carefully managing its power consumption, the Pico can maximize battery life. It can be put into sleep modes or powered down when it's not actively processing data or performing tasks.

3. These low-power modes are particularly valuable for portable and energy-efficient projects.

GPIO FUNCTIONS:

1. GPIO pins on the Pico are versatile and can be configured for various functions. These functions include digital input and output, PWM (Pulse Width Modulation) output, I2C, SPI, UART (Universal Asynchronous Receiver-Transmitter), and more.

2. GPIO pins can respond to external events through interrupts, making them suitable for tasks that require real-time responsiveness.

3. Additionally, some GPIO pins can be used for ADC (Analog-to-Digital Conversion), allowing the Pico to read analog sensor data with precision.

UART, I2C, and SPI:

1. The Pico features multiple UART, I2C, and SPI interfaces, which are essential for communication with other devices. These interfaces are widely used in

embedded systems to connect sensors, displays, memory devices, and other peripherals.

2. UART provides serial communication, while I2C and SPI are commonly used for interfacing with sensors and other microcontrollers. These interfaces are fundamental for building connected and sensor-based projects.

PWM (PULSE WIDTH MODULATION):

1. Pulse Width Modulation (PWM) is a technique used to generate analog-like signals from digital devices. The Pico provides PWM channels that allow you to control the duty cycle and frequency of the output waveform.
2. PWM is commonly used to control servos, LEDs, motors, and other devices that require variable voltage levels.

TIMERS AND REAL-TIME COUNTER:

1. The Raspberry Pi Pico includes on-board timers and a real-time counter, offering precise timing capabilities.
2. Timers are essential for scheduling tasks and creating accurate delays in your code.
3. The real-time counter can be used for timekeeping and measuring time intervals in applications that require precise timing.

PROGRAMMABLE IO (PIO) BLOCKS:

1. PIO blocks provide a unique and highly flexible way to implement custom high-speed IO functionality on the Pico.
2. Each PIO block includes eight state machines that can be programmed to perform custom logic operations.

3. PIO blocks can emulate various interfaces, such as SD Card and VGA, effectively extending the Pico's capabilities beyond its default hardware features.

APPLICATIONS

- **Embedded Systems Development:** Create custom controllers and systems.
- **IoT Devices:** Build connected, smart gadgets.
- **Robotics Control:** Control motors and sensors in robots.
- **Sensor-Based Projects:** Gather data from environmental sensors.
- **Home Automation:** Automate home lighting and appliances.
- **Data Logging:** Record and store sensor data.
- **Display Controllers:** Drive various display types.
- **Audio Projects:** Design interactive audio applications.
- **Educational Tools:** Ideal for learning electronics.
- **Custom Controllers:** Develop tailored device interfaces.
- **Gaming Projects:** Create simple interactive games.
- **Prototyping:** Rapidly prototype electronic ideas.
- **Art Installations:** Add interactivity to art pieces.
- **Remote Control Systems:** Control machines remotely.
- **Scientific Instruments:** Build research tools and experiments.

5.2 HEART RATE SENSOR

A heart rate sensor combines a green LED (light-emitting diode) to illuminate the skin, a BPW34 pin diode to detect reflected light, and an LM358 operational amplifier to process the signal. This sensor measures subtle changes in blood volume, enabling real-time heart rate monitoring in wearables and medical devices.

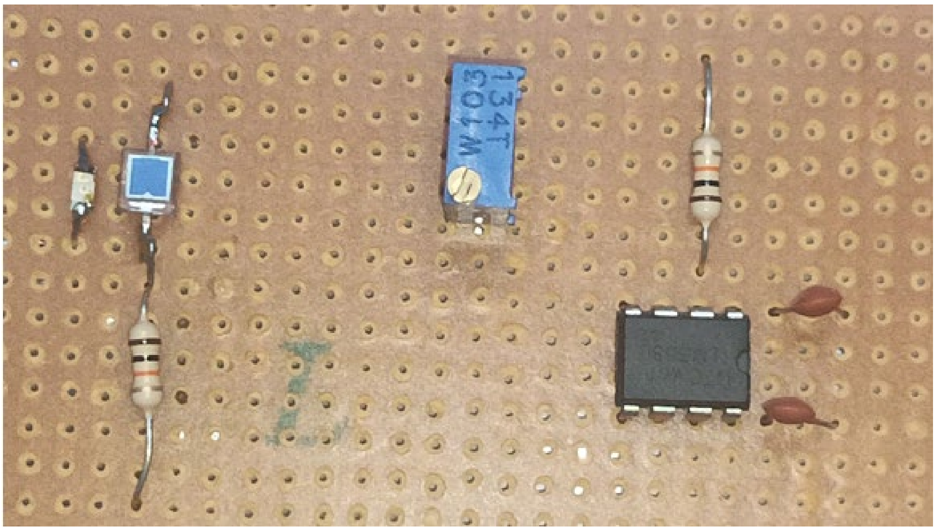

Fig 5.4 Heart Rate Sensor

PINOUT:

VCC: Connect to Ground

GND: Connect to Ground

AOUT: Analog Output

DIN: Digital Input

SPECIFICATIONS

- Sensor Type: Reflective photoplethysmographic (PPG) heart rate sensor.
- LED Type: Green light-emitting diode (LED) for skin illumination.

- Photodiode Type: BPW34 high-speed PIN photodiode for detecting reflected light.
- Signal Amplification: LM358 operational amplifier for signal processing.
- Wavelength: Typically in the range of 500 to 600 nanometers (green spectrum) for optimal blood volume measurement.
- Power Supply Voltage (Vcc): Typically operates at 3.3V or 5V.
- Output Signal: Analog voltage signal proportional to heart rate.
- Sensitivity: High sensitivity to small changes in blood volume.

5.3 DALLAS TEMPERATURE DS18B20

The DS18B20 is a digital temperature sensor manufactured by Dallas Semiconductor, which is now a part of Maxim Integrated. It's a popular choice for measuring temperature in various applications due to its accuracy, simplicity, and ability to communicate over a one-wire interface.

Fig 5.5 DS18B20 Dallas Temperature Sensor

PIN DESCRIPTION

VDD (Pin 3 - Power Supply): This is the power supply pin for the sensor. It requires a voltage supply in the range of 3.0V to 5.5V. In parasite power mode, this pin is not connected to an external power source; instead, the sensor draws power from the data line (DQ pin). In external power mode, you need to provide power to this pin.

DQ (Pin 2 - Data Line): DQ stands for "Data Quality," and it's the data communication pin of the sensor. This is the pin you use to send and receive data to and from the sensor. In parasite power mode, the DQ pin also serves as the power source for the sensor. It uses a one-wire communication protocol.

GND (Pin 1 - Ground): This is the ground reference for the sensor. Connect it to the ground (0V) of your power supply or microcontroller.

DS18B20 sensor can also operate in parasite power mode, where it draws power from the data line (DQ pin). In this mode, you would connect VDD and DQ together, and only use GND as a separate pin.

And Note here DQ pin need pull-up resistor used to maintain the integrity of the signal and ensure reliable data transmission.

SPECIFICATIONS

Temperature Range	-55°C to +125°C (-67°F to +257°F)
Accuracy	±0.5°C
Resolution	9 to 12 bits.
Conversion times	93.75ms (9-bit), 187.5ms (10-bit), 375ms (11-bit), 750ms (12-bit).
Voltage Supply Range (VDD)	3.0V to 5.5V.
Parasitic Power Mode	Yes
Data Interface	One-wire interface
Unique 64-Bit Address	64-bit ROM ID.

Operating Current	1mA
Operating Voltage	3.3 – 5V
Package Size (Dimension)	4.8 x 4.8 x 4 mm

WORKING

The DS18B20 temperature sensor works by measuring the temperature of its surroundings and converting that temperature into a digital value that can be read and interpreted by a microcontroller or other digital device. Here's a basic overview of how the sensor works:

- **TEMPERATURE SENSING**: The DS18B20 contains a temperature-sensing element, which is often a silicon-based sensor called a "band-gap sensor." This sensor's electrical characteristics vary with temperature. As the temperature changes, the electrical properties of the sensor change as well.
- **ANALOG-TO-DIGITAL CONVERSION**: The sensor's internal analog-to-digital converter (ADC) converts the varying analog voltage from the temperature-sensing element into a digital value. This digital value represents the temperature in a coded format.
- **ONE-WIRE COMMUNICATION**: The DS18B20 communicates using a one-wire digital communication protocol. This protocol allows the microcontroller to request temperature readings and receive the digital temperature value from the sensor.
- **UNIQUE ADDRESS**: Each DS18B20 sensor has a unique 64-bit address burned into its memory during manufacturing. This address is used to differentiate multiple sensors on the same one-wire bus.

- **CONVERSION PROCESS**: To measure the temperature, the microcontroller initiates a temperature conversion by sending a command to the DS18B20. The sensor then performs an internal conversion process, which takes a certain amount of time depending on the selected resolution. Once the conversion is complete, the sensor makes the digital temperature value available for the microcontroller to read.
- **READING TEMPERATURE**: The microcontroller reads the digital temperature value from the sensor using the one-wire protocol. This digital value is then processed and converted into an understandable temperature reading. The microcontroller can use this reading for various applications, such as displaying the temperature on a display, logging it, or triggering actions based on temperature thresholds.

DS18B20's ability to provide accurate temperature readings using a simple one-wire interface and its compact form factor make it a popular choice for various temperature sensing applications, from industrial control systems to consumer electronics.

5.4 GPS Module

A GPS (Global Positioning System) module is a device that allows you to receive signals from GPS satellites and determine your precise location on Earth.

Fig 5.6 GPS Module

SATELLITE COMMUNICATION: GPS satellites orbit the Earth and continuously transmit signals that include information about the satellite's position and the current time. A GPS module receives these signals and uses the information to calculate its own position.

TRIANGULATION: The GPS module receives signals from multiple satellites. By analyzing the time it takes for the signals to travel from the satellites to the module, the module can calculate the distance to each satellite. With distances from at least four satellites, the module can triangulate its precise location using trilateration algorithms.

POSITION AND TIMING: In addition to determining its own latitude, longitude, and altitude, a GPS module can also provide accurate timing information. This timing data is often used in applications that require precise synchronization.

COMMUNICATION INTERFACES: GPS modules typically have communication interfaces that allow them to connect to external devices such as microcontrollers, computers, or smartphones. Common communication protocols include UART (serial communication) and I2C.

NMEA SENTENCES: Like the NEO-6M module mentioned earlier, GPS modules commonly output data in the form of NMEA sentences. These sentences contain information about the GPS fix status, latitude, longitude, altitude, speed, and more. NMEA sentences are ASCII strings that can be parsed easily.

ANTENNA: GPS modules require an external antenna to receive signals from satellites. The quality and placement of the antenna can impact the accuracy of the module's positioning.

PIN DESCRIPTION

VCC: This pin is used to provide power to the GPS module. It's usually connected to a positive supply voltage, such as 3.3V or 5V, depending on the module's specifications.

GND OR GROUND: This pin is connected to the ground or common reference of the power supply.

TX OR TRANSMIT DATA: This pin is used for serial communication and sends data from the GPS module to an external device, like a microcontroller or computer.

RX OR RECEIVE DATA: This pin is also used for serial communication and receives data from an external device.

SPECIFICATIONS

Receiver Type	50-channel u-blox 6 receiver
GPS Bands	L1 C/A code, SBAS, QZSS
Sensitivity	Tracking & Navigation: -161 dBm
	Cold Start: -147 dBm
	Hot Start: -156 dBm
Position Accuracy	Autonomous: < 2.5 meters CEP
	SBAS: < 2.0 meters CEP
Update Rate	Up to 5 Hz (5 times per second)
Protocol Support	NMEA-0183 / u-blox(UBX)
Interface	UART
Antenna	External
Supply Voltage	3.3V

Power Consumption	30mA
Backup Power Consumption	15 µA
Operating Temperature	-40°C to +85°C
Dimensions	22.4 mm x 17 mm
Positioning Modes	Auto 2D/3D/2Donly/3Donly
Dynamics velocity	Up to 4g acceleration, up to 515 m/s^2
Time Pulse	1 pulse per second (PPS) output

WORKING

The working principle of a GPS module, such as the NEO-6M, involves receiving signals from multiple GPS satellites orbiting the Earth and using those signals to calculate the module's precise location. Here's how it works in more detail:

- **SATELLITE SIGNALS:** GPS satellites continuously transmit signals that include information about the satellite's location and the current time. These signals travel at the speed of light.
- **TIME OF FLIGHT:** The GPS module receives these signals, and by measuring the time it takes for the signals to travel from the satellites to the module, it can determine the distance between itself and each satellite. This is possible because the signal travels at a known speed (the speed of light).
- **TRILATERATION:** To determine its position, the GPS module needs signals from at least four satellites. With signals from four or more satellites, the module can use a process called trilateration to calculate

its exact position. Trilateration involves intersecting spheres centered at the satellite positions, and the intersection point is the GPS receiver's location.

- **DATA PROCESSING:** The GPS module processes the received signals to extract information about the satellites' positions, the time the signals were transmitted, and other relevant data.
- **COORDINATE CALCULATION:** Using the time-of-flight measurements and the known positions of the satellites, the GPS module calculates its latitude, longitude, and altitude. This information provides the module's precise position on Earth.
- **NMEA SENTENCES:** The GPS module typically outputs data in the form of NMEA (National Marine Electronics Association) sentences. These sentences contain information about the GPS fix status, latitude, longitude, altitude, speed, and more. The module transmits these sentences over a serial communication interface (UART) to an external device (like a microcontroller) for further processing.
- **AIDING DATA:** Some GPS modules can use aiding data, such as ephemeris and almanac data, to speed up the position calculation process. This data provides information about satellite orbits and positions, helping the module acquire a position fix more quickly.
- **ACCURACY AND FIX TYPES:** The accuracy of the GPS module's position calculation depends on factors like the number of satellites in view, signal quality, and the module's processing capabilities. The module can provide different types of fixes, such as "No Fix" (when it can't determine a position), "2D Fix" (latitude and longitude), and "3D Fix" (latitude, longitude, and altitude).

- **POWER MANAGEMENT:** To optimize power usage, some GPS modules have power-saving modes. For example, they might go into a low-power mode when not actively acquiring satellite signals.

GPS module works by using precise timing measurements from multiple satellites to calculate its position on Earth. The accuracy of the position fix depends on various factors, including the quality of the signals, the number of satellites in view, and the module's processing capabilities.

5.5 GSM Module

A GSM module is a hardware component that allows devices to communicate over cellular networks using the GSM standard. These modules are widely used in applications where remote communication, mobile connectivity, or wireless data transmission is required.

Fig 5.7 GSM Module

COMMUNICATION TECHNOLOGY: GSM modules utilize cellular networks to establish communication. They can send and receive SMS messages, make and receive voice calls, and in some cases, establish GPRS or 2G/3G/4G data connections for internet access.

SERIAL COMMUNICATION: GSM modules usually communicate with external devices, such as microcontrollers, through serial communication interfaces like UART (Universal Asynchronous Receiver-Transmitter) or RS-232. This allows for easy integration into various projects.

AT COMMANDS: Communication with a GSM module is often achieved by sending AT commands. These commands are simple textual instructions that allow you to control various functions of the module, such as sending texts, making calls, checking signal strength, and more.

SIM CARD: A SIM card (Subscriber Identity Module) is required for the GSM module to function. The SIM card contains essential information for identifying and authenticating the device on the cellular network.

ANTENNA: A GSM module requires an antenna for transmitting and receiving signals. The quality and placement of the antenna can significantly affect the module's performance.

GSM vs. GPRS vs. 3G/4G: GSM refers to the 2G cellular network standard. GPRS (General Packet Radio Service) is an extension of GSM that allows for data transmission. 3G and 4G (LTE) are later generations of cellular networks that offer higher data speeds and capabilities compared to 2G.

PIN DESCRIPTION

VCC (Power Supply Voltage): This is the power supply pin. Connect it to a suitable power source (usually 3.3V or 5V) within the specified voltage range of the module.

GND (Ground): Connect this pin to the ground of your power supply.

TXD (Transmit Data): This pin is used to send data from your microcontroller or device to the GSM module. Connect it to the RX pin of your microcontroller.

RXD (Receive Data): This pin is used to receive data from the GSM module. Connect it to the TX pin of your microcontroller.

RST: Reset pin. Optional. Connect to a digital output pin for module reset.

RI: Ring Indicator pin. Indicates an incoming call or SMS message. Connect to a digital input pin.

DTR: Data Terminal Ready pin. Optional. Connect to a digital output pin to indicate readiness for communication.

CTS: Clear To Send pin. Optional. Connect to a digital input pin to check if the module is ready to receive data.

RTS: Request To Send pin. Optional. Connect to a digital output pin to indicate readiness to send data.

GPIOS (General Purpose Input / Output Pins): These pins are used for various purposes like controlling external devices, reading sensor data, or indicating status.

ANTENNA: Connect to an external GSM antenna for signal reception and transmission.

MIC+ and MIC-: Microphone input pins for voice communication.

SPK+ and SPK-: Speaker output pins for voice communication.

SPECIFICATIONS

Frequency Bands (Quad-band)	850/900/1800/1900 MHz
SIM Card Interface	Standard SIM card slot
Power Supply Voltage	3.4V - 4.5V
Serial Interface	UART
AT Command Support	Yes
SMS Support	Yes
Voice Call Support	Yes

Data Transmission	GPRS/EDGE/3G/4G
Antenna	External antenna required
GPIO Pins	Available for external connections
Built-in SIM Toolkit	Yes (if supported)
Network Services	Caller ID, call waiting, call forwarding
Dimensions	2.5cm x 2.5cm GSM
Maximum Transmit Power	Around 2W (33 dBm) for GSM.
Operating Temperature	-20°C to +85°C
SIM Voltage Compatibility	1.8V, 3V (depends on the module).
Integrated TCP/IP Stack	Yes
Network Registration	Automatic.
Roaming Support	Automatic

WORKING

The working of a GSM (Global System for Mobile Communications) module involves several key steps that allow it to connect to a cellular network and provide functionalities like making calls, sending SMS, and transmitting data. Here's a simplified overview of how a GSM module works:

- **POWER ON AND INITIALIZATION:** When the GSM module receives power, it goes through an initialization process. It sets up its internal components, establishes communication interfaces, and prepares for network registration.

- **SIM CARD DETECTION:** The module checks if a SIM card is inserted into the SIM card slot. If a SIM card is detected, it proceeds to the next step.
- **Network Registration:** The GSM module attempts to register with the available cellular network. It searches for nearby network towers and tries to establish a connection with the strongest available network.
- **AT COMMAND COMMUNICATION:** The module communicates with external devices (such as microcontrollers or computers) using AT commands over a serial interface (typically UART). These commands are sent from the external device to the module to instruct it on various tasks.
- **MAKING CALLS:** To make a call, the external device sends an appropriate AT command to the module. The module then dials the specified phone number and establishes a voice connection if the call is successful.
- **SENDING AND RECEIVING SMS:** SMS functionality is also controlled using AT commands. The external device sends commands to the module to create, send, and read SMS messages. The module can store incoming messages in its memory and notify the external device about new messages.
- **DATA TRANSMISSION:** For data communication, the GSM module can use technologies like GPRS, EDGE, 3G, or 4G. It establishes a data connection with the cellular network, and data can be transmitted in the form of packets using protocols like TCP/IP.

- **SIGNAL STRENGTH MONITORING:** The module continually monitors the signal strength of the connected cellular network to ensure a stable connection. This information can be shared with the external device if needed.
- **POWER SAVING MODES:** To conserve power, GSM modules often support different power-saving modes. These modes reduce power consumption when the module is idle or not actively communicating.
- **ERROR HANDLING:** The module has built-in error-handling mechanisms to deal with issues such as network disconnections, failed calls, or transmission errors. It can send error codes or notifications to the external device.
- **SECURITY:** The GSM module uses SIM-based authentication to ensure secure network access. It also supports encryption for data transmission, enhancing the security of communications.
- **SIM TOOLKIT INTERACTION:** If the module supports SIM Toolkit functionalities, it can interact with the SIM card to access services and applications provided by the mobile network operator.

Implementation and commands can vary between different GSM module models and manufacturers. Always refer to the specific module's documentation and AT command guide for detailed information on how to control and use its functionalities.

5.6 TACTILE PUSH BUTTON

A tactile push button is a type of switch that is designed to be activated by applying physical pressure to it, typically by pressing it with a finger or some other object. These buttons are commonly used in various electronic devices and control panels where user input is required. They provide a tactile feedback, which means that when the button is pressed, you can feel a physical "click" or resistance, confirming that the button press has been registered. Tactile push buttons usually consist of a button cap, a plunger or actuator, and a switch mechanism underneath. The switch mechanism is responsible for making or breaking an electrical connection when the button is pressed or released. The tactile feedback is often achieved through the use of a dome switch, which is a small metal dome that collapses and then rebounds when the button is pressed, producing the characteristic tactile sensation and audible click.

Fig 5.8 Tactile Push Button

PIN DESCRIPTION

1. **COMMON TERMINAL (COM)**: This pin is usually the common connection for the switch and is connected to one side of the switch contact.
2. **NORMALLY OPEN TERMINAL (NO)**: When the switch is not pressed, this terminal is not connected to the common terminal. When the switch is pressed, it makes a connection to the common terminal.
3. **NORMALLY CLOSED TERMINAL (NC)**: When the switch is not pressed, this terminal is connected to the common terminal. When the switch is pressed, it breaks the connection to the common terminal.
4. **GROUND (GND)**: Some tactile switches might have a separate pin for grounding purposes, especially if they are used in electronic circuits.

SPECIFICATIONS

Operating Voltage	12V DC
Contact Configuration	Normally Open (NO)
Contact Rating	100mA
Actuation Force	160g
Travel Distance	0.3mm
Life Cycle / Durability	100,000 cycles
Tactile Feedback	Medium
Operating Temperature	-25°C to +70°C
Contact Resistance	< 50 mΩ
Insulation Resistance	> 100 MΩ
Bounce Time	5ms
Mounting Type	Through-hole
Terminal Type	PCB pins
Dimensions	12x12mm

WORKING

The working principle of a tactile switch involves the use of a spring-loaded mechanism to create a temporary electrical connection when the switch is pressed. Tactile switches are designed to provide tactile feedback to the user, allowing them to feel when the switch is actuated. Here's how the working of a typical tactile switch can be explained:

- **INTERNAL COMPONENTS**: A tactile switch consists of several internal components, including a housing, a plunger or button, a spring, and electrical contacts.
- **RESTING STATE**: In the resting state (not pressed), the spring exerts pressure on the plunger, keeping it in an extended position. The plunger is located over the electrical contacts. There are typically two sets of contacts inside the switch: Normally Open (NO) and Normally Closed (NC).
- **USER PRESSES THE SWITCH**: When a user applies force to the button or plunger of the tactile switch, it starts compressing the spring. As the spring is compressed, the plunger moves downward.
- **TACTILE FEEDBACK**: As the plunger moves downward, the user feels a tactile "click" or resistance. This tactile feedback indicates that the switch has been actuated. The tactile feedback is achieved through the mechanical design of the switch, including the shape of the plunger, the spring's tension, and the housing.
- **ELECTRICAL CONNECTION**: As the plunger moves downward, it eventually reaches a point where it pushes the internal electrical contacts together. If the switch is of the Normally Open (NO) type, this action closes the circuit between the common (COM) and NO pins, allowing current to flow through the switch. If the switch is of the Normally Closed (NC) type, pressing the switch opens the circuit between the common (COM) and NC pins, interrupting the current flow.
- **ACTUATED STATE**: The switch remains in the actuated state as long as the user continues to apply pressure on the plunger. The electrical connection remains closed (or open, depending on the switch type) during this time.

- **SWITCH RELEASE**: When the user releases the pressure on the plunger, the compressed spring pushes the plunger back to its extended position. As the plunger moves upward, it separates the electrical contacts. This action returns the switch to its resting state, and the circuit is either opened or closed, depending on the switch type.

A tactile switch uses a combination of mechanical components, such as a spring-loaded plunger and internal contacts, to provide tactile feedback to the user and create a temporary electrical connection when the switch is pressed. The tactile "click" sensation and the electrical behaviour make these switches suitable for various applications where user interaction and momentary switching are required.

5.7 16X2 LIQUID CRYSTAL DISPLAY LCD

A 16x2 LCD (Liquid Crystal Display) is a common type of alphanumeric display module that can display two lines of text, with each line containing up to 16 characters. These displays are widely used in various electronics projects, devices, and applications for displaying information to users. Here are some details about a typical 16x2 LCD module:

DISPLAY SIZE: The LCD screen has 2 lines, and each line can display up to 16 characters (including letters, numbers, symbols, and spaces).

CHARACTER SIZE: The standard character size is typically 5x8 pixels, allowing the display of a variety of characters and symbols.

BACKLIGHT: Many 16x2 LCD modules come with a backlight that can be controlled to improve visibility in different lighting conditions. The backlight can be white, blue, green, or other colors.

COMMUNICATION INTERFACE: 16x2 LCD modules usually use the Hitachi HD44780 or a compatible controller, which is commonly interfaced with microcontrollers using a parallel interface.

CONTRAST CONTROL: Many modules allow you to adjust the contrast of the characters on the screen using a built-in potentiometer.

CONTROLLER COMMANDS: The HD44780 controller supports a set of commands that can be sent from a microcontroller to control the display, cursor position, clearing the display, and more.

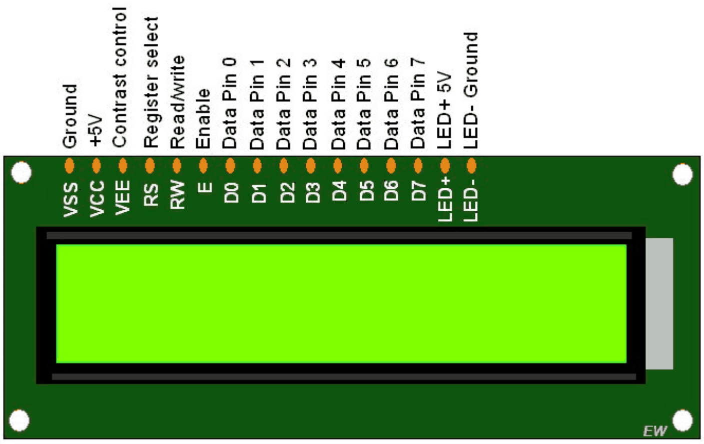

Fig 5.9 16X2 Liquid Crystal Display LCD

PIN DESCRIPTION

VSS (Ground): Connect to the ground of your power supply.

VDD (Power): Connect to the positive supply voltage (usually +5V).

V0 (Contrast): This pin is used to control the contrast of the display. Connect to a variable resistor or a fixed resistor divider to adjust the contrast.

RS (Register Select): This pin selects between data mode (RS = 1) and command mode (RS = 0). In data mode, the module receives character data. In command mode, it receives commands to set display settings.

RW (Read/Write): This pin selects the read (RW = 1) or write (RW = 0) mode. Since most applications only need to write data to the LCD, this pin is often connected to ground (write mode).

E (Enable): The enable pin triggers data/command processing when transitioning from high to low.

D0-D7 (Data Lines): These are the data pins for sending both commands and character data. In 4-bit mode, typically used for minimizing required pins, only the higher 4 data lines (D4-D7) are connected.

Backlight Anode (+): Connect the anode of the backlight to a positive voltage supply (usually +5V) if backlighting is desired.

Backlight Cathode (-): Connect the cathode of the backlight to ground.

Interfacing 16x2 LCD in 4-bit mode:

VSS: GND

VDD: +5V

V0: Connected to a variable resistor for contrast control

RS: Connected to a microcontroller's pin for selecting data or command mode

RW: Connected to GND for write mode

E: Connected to a microcontroller's pin for enabling data/command processing

D4-D7: Connected to microcontroller's pins for data communication

Backlight Anode (+): Connected to +5V

Backlight Cathode (-): Connected to GND

SPECIFICATIONS

Display Size	16 characters x 2 lines.

Character Size	5x8 pixels (5 pixels wide, 8 pixels high).
Controller	HD44780
Operating Voltage	5V
Interface:	4-bit or 8-bit / I2C/SPI
Viewing Area	2.95mm x 5.55mm
Contrast Control	potentiometer or resistor to V0
Temperature Range	0°C to 50°C

WORKING

The working principle of a 16x2 LCD (Liquid Crystal Display) involves manipulating liquid crystals to create characters and graphics on the screen. Here's a step-by-step overview of how a typical 16x2 LCD works:

- **INITIALIZATION:** When the LCD module is powered on, the microcontroller initializes it by sending specific commands. These commands configure the display mode, cursor settings, and other parameters. The initialization process is essential for proper communication between the microcontroller and the LCD.
- **CHARACTER GENERATION:** The LCD has a built-in character generator ROM that contains patterns for standard ASCII characters and some custom symbols. When the microcontroller sends character data to the LCD, the LCD uses this ROM to generate the appropriate character pattern. Each character is formed by a matrix of pixels (dots) that can be turned on or off.
- **DISPLAY DATA RAM (DDRAM):** The LCD has an internal memory called the Display Data RAM (DDRAM). This RAM is organized in a grid that corresponds to the rows and columns of the LCD. When you send

character data to the LCD, it's stored in the DDRAM, and the cursor is automatically advanced to the next position.

- **CURSOR CONTROL:** The microcontroller can control the cursor's position on the screen. By sending cursor-related commands, the microcontroller can move the cursor to a specific location, which determines where the next character will be displayed.
- **INSTRUCTION AND DATA MODES:** The microcontroller communicates with the LCD using the RS (Register Select) pin. When RS is low, the LCD interprets the data on the data pins as commands. When RS is high, the data is treated as character data. This allows the microcontroller to send commands for configuration or character data for display.
- **ENABLE (E) SIGNAL:** The E (Enable) pin is used to trigger the LCD to read the data on its data pins. When the E signal transitions from high to low, the LCD reads the data present on the data pins and processes it as either a command or character data, depending on the RS pin's state.
- **BACKLIGHT CONTROL:** Many LCD modules come with an integrated LED backlight. The backlight can be controlled using a separate connection. Connecting the backlight anode to a positive voltage supply and the cathode to ground activates the backlight.
- **CONTRAST CONTROL:** The V0 pin, also known as the contrast pin, is used to control the contrast of the characters displayed on the LCD. By adjusting the voltage at this pin, you can control the darkness or lightness of the characters.

Overall, the microcontroller sends commands and character data to the LCD module through a parallel or serial interface, and the LCD processes these signals to generate characters and graphics on the screen. It's important to refer to the datasheet of the specific LCD module you're using to understand the details of its operation and communication protocols.

5.8 ESP8266 WIFI MODULE

The ESP8266 is a popular and versatile Wi-Fi module developed by Espressif systems. It gained significant attention for its affordability and capabilities, making it a staple in the maker and IoT (Internet of Things) communities.

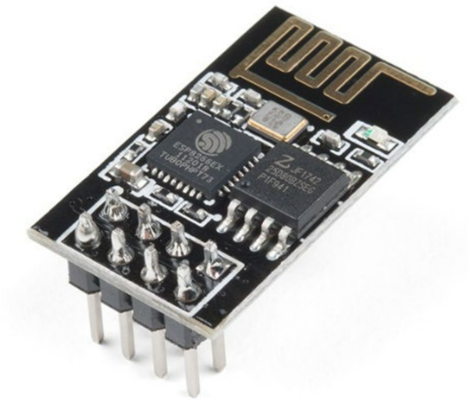

Fig 5.10 Esp8266 WiFi Module

FEATURES: The ESP8266 module integrates a microcontroller, Wi-Fi capabilities, and I/O pins into a single package. It's available in various versions, with differences in terms of flash memory, number of GPIO pins, and other features.

MICROCONTROLLER: The ESP8266 is powered by a Tensilica Xtensa LX106 core, which is a low-power, high-performance microcontroller. It's commonly used for running applications and handling Wi-Fi connectivity.

WI-FI CONNECTIVITY: The module supports 2.4 GHz Wi-Fi networks and can function as both a client (connecting to existing Wi-Fi networks) and an access point (creating its own Wi-Fi network). It also supports various encryption methods for secure communication.

PROGRAMMING: The ESP8266 can be programmed using a variety of programming languages and development environments. The most common language for programming ESP8266 is C/C++, but there are also options for MicroPython and Arduino IDE support.

DEVELOPMENT BOARDS: In addition to the raw module, there are various development boards built around the ESP8266 that make it easier to work with. The most notable is the NodeMCU board.

IOT APPLICATIONS: The ESP8266's affordability, compact size, and Wi-Fi capabilities make it suitable for a wide range of IoT applications, including home automation, sensor networks, smart appliances, and more.

COMMUNITY SUPPORT: The ESP8266 has a large and active community of developers and enthusiasts who contribute to tutorials, libraries, and other resources, making it easier for beginners to get started.

AT COMMANDS: The ESP8266 can be controlled using AT commands, which are simple text-based commands sent over a serial connection. This allows you to interact with the module without extensive programming.

UPGRADED VERSIONS: Since the release of the ESP8266, Espressif Systems has developed more advanced versions, such as the ESP32, which offers even more capabilities, including Bluetooth connectivity and additional GPIO pins.

LICENSING: The ESP8266's software and firmware are released under the Espressif MIT License, allowing for both personal and commercial use.

PIN DESCRIPTION

VCC (3.3V): Power supply pin. The module requires a 3.3V power source. Applying higher voltage can damage the module.

GND: Ground reference pin.

GPIO0: General-purpose digital I/O pin. It's also used during boot-up to determine the boot mode of the module. Depending on its state during boot, the module can enter programming mode, flashing mode, or normal operation.

GPIO2: General-purpose digital I/O pin. This pin can be used for general I/O operations.

TX (Transmit): Serial transmit pin for communicating data from the module to an external device.

RX (Receive): Serial receive pin for receiving data from an external device.

CH_PD (Chip Enable): This is the chip enable pin. It must be held high for the module to function.

RESET: Reset pin. Pulling this pin low momentarily resets the module.

GPIO4: General-purpose digital I/O pin.

GPIO5: General-purpose digital I/O pin.

GPIO12: General-purpose digital I/O pin.

GPIO13: General-purpose digital I/O pin.

GPIO14: General-purpose digital I/O pin.

GPIO15: General-purpose digital I/O pin. Its state during boot determines whether the module boots into UART download mode.

GPIO16: General-purpose digital I/O pin. This pin can also be used to wake the module from deep sleep mode.

ADC (Analog to Digital Converter): This pin can be used to read analog voltages. The ESP8266 has a single ADC input.

SPECIFICATIONS

Operating Voltage	3.3V
Operating Current	70 to 200mA
CPU	Tensilica Xtensa LX106 32Bit
Clock Speed	Normal 80MHz, Overclocked 160MHz
Flash Memory	4MB
RAM	128KB
Wi-Fi Connectivity	Wi-Fi Standards: 802.11 b/g/n (2.4 GHz)
Modes	Station, Soft Access Point, SoftAP
Encryption	WPA, WPA2, WPA/WPA2 Mixed, WEP
Maximum Data Rate	Up to 72.2 Mbps
ADC	10 Bit ADC Single Channel
Interfaces	UART, SPI, I2C
Programming IDE	Arduino IDE
Programming Languages	C/C++, Micro Python
Programming Interfaces	UART, OTA (Over-The-Air)
Dimensions	25 x 15 x 11mm (L x W x H)

WORKING

The ESP8266 Wi-Fi module is a versatile component that enables devices to connect to Wi-Fi networks and communicate over the internet. Here's a high-level overview of how the ESP8266 module typically works:

WI-FI MODES:

- **Station Mode:** The ESP8266 connects to an existing Wi-Fi network as a client, similar to how your smartphone connects to a Wi-Fi network.
- **Access Point Mode:** The ESP8266 creates its own Wi-Fi network to which other devices can connect. This mode is useful for creating standalone networks or configuring the module.

WI-FI CONFIGURATION:

- **SSID and Password:** In station mode, the module needs the SSID and password of the target network for authentication.
- **IP Configuration:** The module can use DHCP to automatically obtain an IP address or be assigned a static IP.

TCP/IP COMMUNICATION:

- **TCP (Transmission Control Protocol):** Provides reliable, ordered, and error-checked data transmission between devices. It's suitable for applications where data integrity is crucial, like file transfers.
- **UDP (User Datagram Protocol):** Offers faster but less reliable data transmission without guaranteed delivery. It's useful for real-time applications like streaming or gaming.

WEB SERVER AND CLIENT:

- **Web Server:** The ESP8266 can serve web pages to clients, displaying dynamic or static content. This is used in IoT dashboards, remote control interfaces, and data visualization.

- **Web Client:** As a web client, the module can send HTTP requests to web servers, retrieve data, and perform actions based on the server's responses.

DOMAIN NAME SYSTEM (DNS):

- The module can resolve domain names to IP addresses using DNS. This is essential for connecting to servers on the internet using human-readable domain names.

SECURE COMMUNICATION:

- **TLS/SSL:** The ESP8266 can establish encrypted connections (HTTPS) for secure data transmission. This is vital when dealing with sensitive information over the network.

MQTT (Message Queuing Telemetry Transport):

- MQTT is a lightweight messaging protocol suitable for IoT applications. The ESP8266 can act as an MQTT client to publish and subscribe to topics on an MQTT broker.

OTA UPDATES:

- The module supports firmware updates over Wi-Fi. It can download and install new firmware versions, enhancing functionality and fixing bugs without requiring physical access.

GPIO and PWM:

- The module's GPIO pins can be configured for digital input or output operations. Some pins can also generate PWM signals, useful for controlling analog-like devices such as LEDs or motors.

ANALOG-TO-DIGITAL CONVERSION (ADC):

- The ESP8266 has built-in ADCs that can read analog sensor values and convert them into digital values that the microcontroller can process.

DEEP SLEEP MODE:

- The module can be put into a low-power deep sleep mode to conserve energy. It's commonly used in battery-powered applications to extend the device's battery life.

REAL-TIME CLOCK (RTC): The ESP8266 has an internal RTC that can track time and wake up the device from deep sleep at specific intervals.

SPI, I2C, UART COMMUNICATION:

- The module can communicate with other devices using serial protocols like SPI, I2C, and UART. This allows integration with various sensors, displays, and other peripherals.

MEMORY MANAGEMENT:

- The ESP8266 has a limited amount of RAM and Flash memory. Efficient memory management is crucial for stable operation and preventing memory-related issues.

SDKs AND PROGRAMMING LANGUAGES:

- The ESP8266 can be programmed using various programming languages and SDKs, including the Arduino IDE, ESP8266 SDK,MicroPython, and more.

GPIO Interrupts:

- The module can be configured to trigger interrupts on specific GPIO pin state changes. This is useful for responding to external events in real-time.

WIFI MANAGER LIBRARY:

- The WiFi Manager library simplifies Wi-Fi configuration by providing a web interface for connecting to Wi-Fi networks without reprogramming the module.

COEXISTENCE WITH OTHER PROTOCOLS:

- The ESP8266's Wi-Fi communication can coexist with other protocols running on the same microcontroller, enabling complex applications with multiple communication channels.

OTA UPDATES (Optional):

- The module can be programmed to support Over-The-Air (OTA) updates. This means that you can remotely update the firmware on the ESP8266 without physically connecting to it. OTA updates are valuable for maintaining and improving devices in the field.

The firmware is designed to handle various error scenarios, such as network disconnections, failed Wi-Fi connections, and data transmission errors. It often includes mechanisms for reconnection and error recovery to ensure the module's continued operation.

WORKING

1. HARDWARE SETUP:

- Raspberry Pi Pico: The Raspberry Pi Pico serves as the central controller of the system. It connects to and manages all other hardware components.
- LCD Display: The LCD provides a user interface for displaying

information, including location, heart rate, and temperature.

- ESP8266: The ESP8266 module provides internet connectivity, allowing the system to send and receive data via Wi-Fi.
- GPS Module: The GPS module continuously tracks the user's location, providing latitude and longitude coordinates.
- GSM Module: The GSM module facilitates communication through text messages and phone calls to designated contacts in case of an emergency.
- Heart Rate Sensor: Monitors the user's heart rate continuously.
- Temperature Sensor: Measures the ambient temperature.
- Push Button: This acts as an emergency trigger for the user to send distress signals.

2. DATA COLLECTION:

- The GPS module continuously collects location data, which is relayed to the Raspberry Pi Pico.
- The Heart Rate Sensor and Temperature Sensor monitor vital signs and environmental conditions, sending data to the Raspberry Pi Pico.

3. USER INTERACTION:

- The LCD Display shows the user their real-time location, heart rate, and temperature.
- Users can trigger an emergency alert by pressing the Push Button in case of distress or danger.

4. EMERGENCY ALERT:

- When the Push Button is pressed, the Raspberry Pi Pico activates the GSM Module to send distress alerts to designated contacts.
- These alerts may include the user's location obtained from the GPS module.

5. INTERNET CONNECTIVITY:

- The ESP8266 module connects to a central server or cloud platform over Wi-Fi, sending location data for real-time tracking and monitoring.

6. CENTRAL SERVER OR CLOUD PLATFORM:

- This platform receives and processes data from the IoT devices.
- It stores user information, including GPS coordinates, in a secure database.
- It sends alerts to emergency contacts when a distress signal is received.

7. DATA ANALYSIS:

- The system continuously monitors user data, including heart rate and temperature.
- If unusual spikes or drops are detected in these data, the system can automatically trigger alerts or notifications.

8. USER CONFIRMATION:

- In case of an alert, the system may request confirmation from the user through the LCD Display or mobile app to ensure it's not a false alarm.

9. PRIVACY AND SECURITY:

- The system ensures data privacy by encrypting information sent over the network and implementing access control measures.

CHAPTER-6

RESULTS AND DISCUSSION

Temperature, Heart rate and Location Monitoring:

If a woman is in danger, she will press the panic button. In turn, Heart Rate Sensor and Temperature Sensor sense the heart rate and temperature of a woman and the GPS track the location of a woman. The LCD will display the heart rate, temperature and location of a woman. Then the GSM will send the alert message to SOS/ selected contact and also the alert mail is send to the selected mail id using IoT with the help of WiFi module. We have used an IoT software tool called “Ubidots STEM” which will create real-time dashboards to analyse data collected from sensors installed in IoT devices and transfers that data through an IoT gateway. In our project, we can able to monitor the live location, temperature and heart rate of a woman using this IoT software tool.

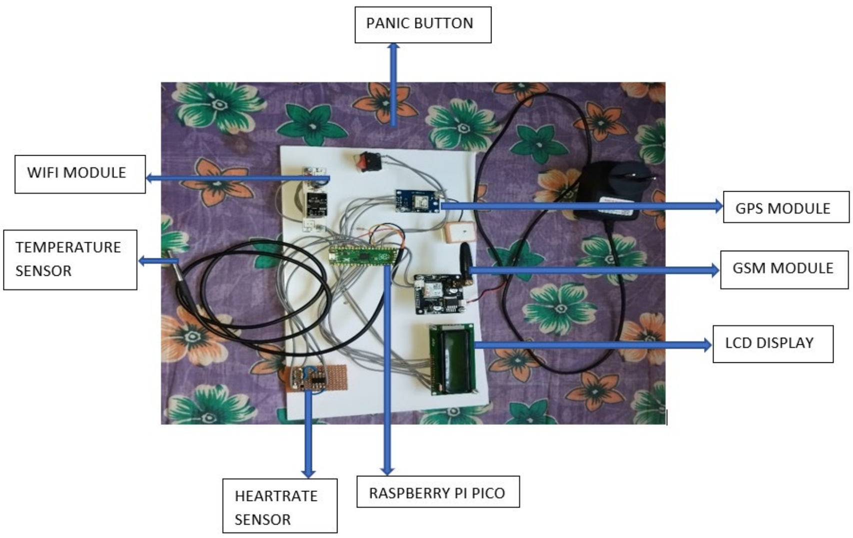

Fig 6.1 Hardware Setup

Fig 6.2 LCD Heart Rate Result

Fig 6.3 LCD Temperature Result

Fig 6.4 LCD Location Result

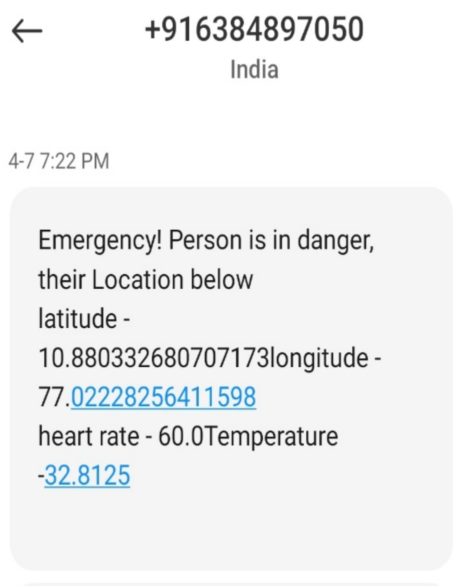

Fig 6.5 Alert Message

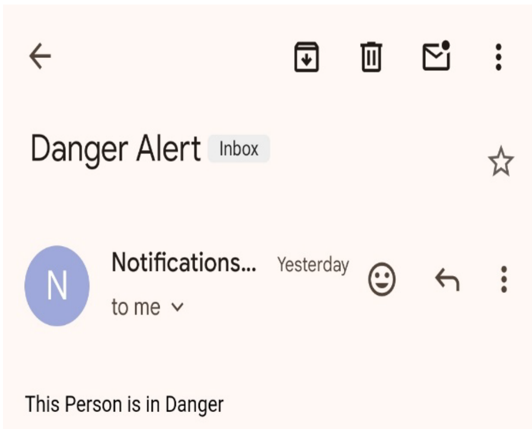

Fig 6.6 Email Alert

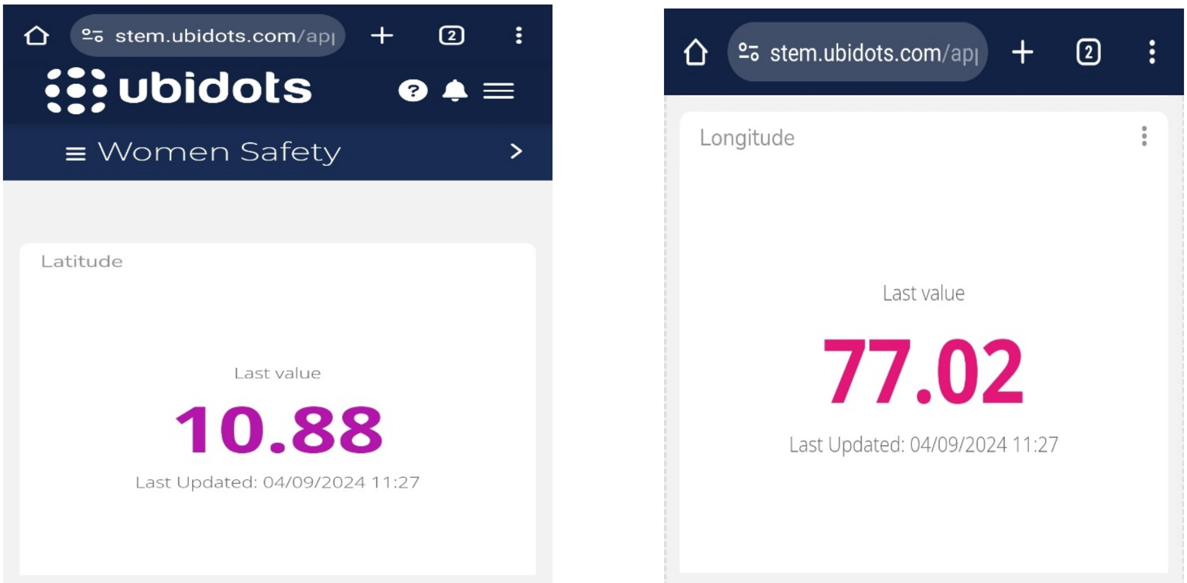

Fig 6.7 Software Location Result

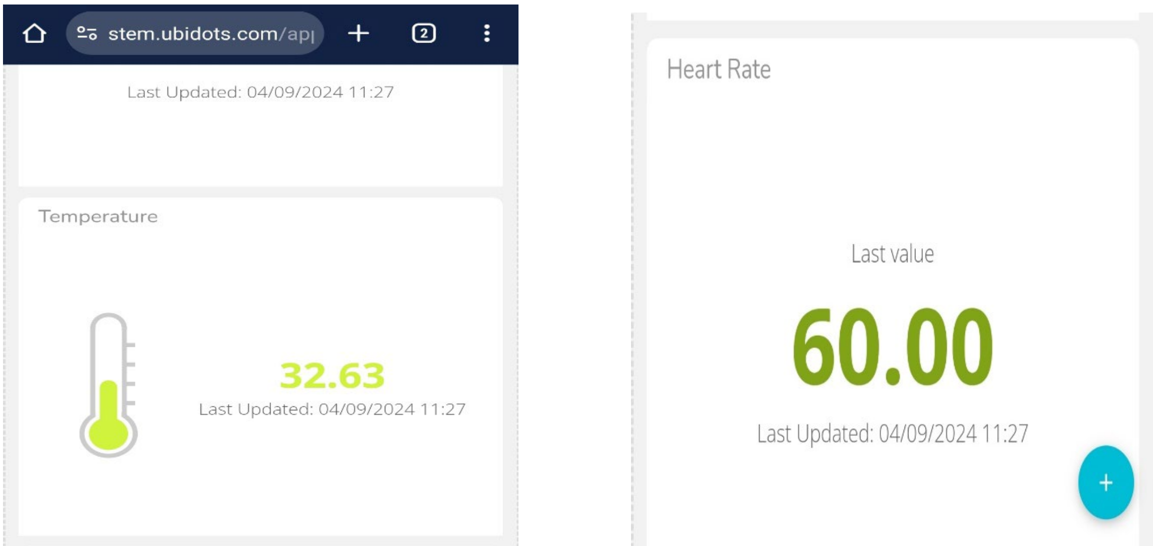

Fig 6.8 Software Temperature and Heart Rate Result

CHAPTER-7

CONCLUSION

The "IoT Based Smart Women Safety System" is a ground breaking project that combines cutting-edge technology with a deep commitment to women's safety. By integrating the Raspberry Pi Pico, LCD, ESP8266 WiFi Module, GPS module, GSM module, Heart Rate Sensor, Temperature Sensor, and the Panic Button, it provides a comprehensive solution for continuous monitoring and instant alerts in distress situations. Through real-time location tracking, vital sign monitoring, and easy user interaction via the LCD Display and mobile app, it empowers women with a sense of security and confidence in their daily lives. The project prioritizes data privacy and security, ensuring confidential user information. While challenges were met with innovative solutions, future enhancements, such as AI integration and 5G networks, hold the promise of a smarter and more responsive system. In conclusion, the "IoT Based Smart Women Safety System" represents a fusion of technology and social responsibility, offering women a safer, more inclusive, and equal world, where they can live without fear and with the assurance that help is always within reach.

CHAPTER-8

FUTURE SCOPE

The IoT Based smart Women safety system using Raspberry Pi Pico can give many more protection for everyone at any situation . A recent estimate shows that, the usage of women safety device will increase in future. The future scope of the project is to implement this prototype model into a complete device with more compact and an effective solution for women safety. It is possible to upgrade the system and can be adaptable to desired environment. Because, it is based on object oriented design, any further changes can be easily adaptable.

REFERENCES

[1] Ahir, S., Kapadia, S., Chauhan, J., &Sanghavi, N. (2018, January). The Personal Stun-A Smart Device For Women's Safety. In 2018 International Conference on Smart City and Emerging Technology (ICSCET) (pp. 1-3). IEEE

[2] Anusha, D.C., Padaganur, S., Purohit, S., Deshmukh, M., Kanamadi, M., Dixit, U. (2022). Design and Implementation of Women Safety Smart Gadget Using IoT. In: Goar, V., Kuri, M., Kumar, R., Senjyu, T. (eds) Advances in Information Communication Technology and Computing. Lecture Notes in Networks and Systems, vol 392. Springer, Singapore.

[3] AshwiniJadhav, N., ReshmaJadhav, S., ShwetaPawar, S., and ShrenikSarade, S.(2017) ‘Review of One Touch System for Women Safety’, International Journal for Research in Applied Science & Engineering Technology, ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887 Volume 5.

[4] Chatterjee, S., &Mandal, S. (2022). Design and Implementation of Women Security System Based on GSM and GPS Technology. In ICT Systems and Sustainability (pp. 217-223). Springer, Singapore.

[5] Chon, D. S., & Clifford, J. E. (2021). The impacts of international rape laws upon official rape rates. International journal of offender therapy and comparative criminology, 65(2-3), 244-261.

[6] Gayathri, I., Sandhya, K., Sushmitha, P., Suresh, N.,Usha Kiran Reddy, A.(2017) ‘Self Defense System for Women Safety with Location Tracking and SMS Alerting’, International Journal of Innovative Technologies, ISSN 2321-8665, Vol.05, Issue.04, Pages:0590-0592.

[7] Huang, Y., Tan, G., Gou, F., Li, M. C., Lee, S. L., & Wu, S. T. (2019). Prospects and challenges of mini- LED and micro- LED displays. Journal of the Society for Information Display, 27(7), 387-401.

[8] Hyndavi, V., Nikhita, N. S., &Rakesh, S. (2020, June). Smart wearable device for women safety using IoT.In 2020 5th International Conference on Communication and Electronics Systems (ICCES) (pp. 459-463).IEEE.

[9] K. Ashok, A. B. Gurulakshmi, M. B. Prakash, R. Poornima, N. S. Sneha and V. Gowtham, "A Survey on Design and Application Approaches in Women-Safety Systems," 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), 2022, pp. 101-110, doi: 10.1109/ICACCS54159.2022.9784981.

[10] Kurian, M.Z., Shambhavi, S. and Nagaraja, M.(2016) ‘Smart Electronic System for Women Safety’, International Journal Of Innovative Research In Electrical, Electronics, Instrumentation And Control Engineering, Vol. 4, Issue 3.

[11] Prakash, N., Udayakumar, E., Kumareshan, N., Gowrishankar, R. (2021). GSM-Based Design and Implementation of Women Safety Device Using Internet of Things. In: Peter, J., Fernandes, S., Alavi, A. (eds) Intelligence in Big Data Technologies—Beyond the Hype. Advances in Intelligent Systems and Computing, vol 1167.Springer, Singapore.

[12] Rajesh Kannan, M., Jyothsna, K., Aparna, T.S., Anjali, T., Meera, M., Amrutha, S.D. (2020).IoT-Based Women Security System. In: Ranganathan, G., Chen, J., Rocha, Á. (eds) Inventive Communication and Computational Technologies. Lecture Notes in Networks and Systems,vol 89. Springer, Singapore

[13] Saleem, S. I., Zeebaree, S., Zeebaree, D. Q., &Abdulazeez, A. M. (2020). Building smart cities applications based on IoT technologies: A review. Technology Reports of Kansai University, 62(3), 1083-1092.

[14] Sasikumar, S., Prabha, S., Saketh, K. S., &Raghav, C. (2022). Design and Implementation of Women's Safety System in Any Problematic Places. Available at SSRN 4121024.

[15] Seelam, K., &Prasanti, K. (2018, January). A novel approach to provide protection for women by using smart security device.In 2018 2nd International Conference on Inventive Systems and Control (ICISC) (pp. 351-357).IEEE.

[16] Singh, V., &Kharat, V. (2017, September). A proposed system for security in campuses using IoT platform: A case study of a women‘s university. In 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC) (pp. 305-310). IEEE.

[17] Sogi, N. R., Chatterjee, P., Nethra, U., & Suma, V. (2018, July). SMARISA: A Raspberry Pi Based Smart Ring for Women Safety Using IoT. In 2018 International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 451-454). IEEE

[18] Toney G, Jaban F, Puneeth S. et al. Design and implementation of safety arm band for women and children using ARM7. 2015 International Conference on Power and Advanced Control Engineering (ICPACE); Bangalore. 2015 Aug 12-14. p. 300–3.

[19] Vigneshwari S, Aramudhan M. Social information retrieval based on semantic annotation and hashing upon the multiple ontologies. Indian Journal of Science and Technology. 2015 Jan; 8(2).

APPENDIX I

Main Code, Raspberry pi pico:

```
import machine
button = machine.Pin(19, machine.Pin.IN, machine.Pin.PULL_UP)
from machine import Pin, UART
import utime, time
gpsModule = UART(0, baudrate=9600, tx=Pin(0), rx=Pin(1))
iot= UART(0, baudrate=9600, tx=machine.Pin(12),
rx=machine.Pin(13),timeout=2000)
uart= UART(1, baudrate=9600, tx=machine.Pin(4),
rx=machine.Pin(9),timeout=2000)
print(gpsModule)
buff = bytearray(255)
TIMEOUT = False
FIX_STATUS = False
latitude = ""
longitude = ""
satellites = ""
GPStime = ""
panic=0
from machine import ADC, Pin
import machine, onewire, ds18x20
ds_pin = machine.Pin(21)
ds_sensor = ds18x20.DS18X20(onewire.OneWire(ds_pin))
roms = ds_sensor.scan()
adc = ADC(Pin(26))
import time #library for Time delay
from lcd_lib import LCD # import Header file LCD
led = machine.Pin(22, machine.Pin.OUT)
# Define the pin numbers for your LCD
rs_pin = 3
e_pin = 5
d4_pin = 6
d5_pin = 7
d6_pin = 8
d7_pin = 9
lcd = LCD(rs_pin, e_pin, d4_pin, d5_pin, d6_pin, d7_pin)
heart=0;
heart_rate=0
def getGPS(gpsModule):
    global FIX_STATUS, TIMEOUT, latitude, longitude, satellites, GPStime
```

```
    timeout = time.time() + 8
    while True:
        gpsModule.readline()
        buff = str(gpsModule.readline())
        parts = buff.split(',')

        if (parts[0] == "b'$GPGGA" and len(parts) == 15):
            if(parts[1] and parts[2] and parts[3] and parts[4] and parts[5] and
parts[6] and parts[7]):
                print(buff)

                latitude = convertToDegree(parts[2])
                if (parts[3] == 'S'):
                    latitude = -latitude
                longitude = convertToDegree(parts[4])
                if (parts[5] == 'W'):
                    longitude = -longitude
                satellites = parts[7]
                GPStime = parts[1][0:2] + ":" + parts[1][2:4] + ":" + parts[1][4:6]
                FIX_STATUS = True
                break

        if (time.time() > timeout):
            TIMEOUT = True
            break
        utime.sleep_ms(500)

def convertToDegree(RawDegrees):

    RawAsFloat = float(RawDegrees)
    firstdigits = int(RawAsFloat/100)
    nexttwodigits = RawAsFloat - float(firstdigits*100)

    Converted = float(firstdigits + nexttwodigits/60.0)
    Converted = '{0:.6f}'.format(Converted)
    return str(Converted)
def map_range(value, from_low, from_high, to_low, to_high):

    # Ensure the value is within the original range
    value = max(min(value, from_high), from_low)
```

```
    # Map the value to the target range
    mapped_value = (value - from_low) * (to_high - to_low) / (from_high -
from_low) + to_low

    return mapped_value
def send_sms():
    uart.write(str.encode('AT'+'\r\n'))
    time.sleep(1)
    print("welcome")
    uart.write(str.encode('ATE0'+'\r\n')) # Disable the Echo
    time.sleep(2)

    uart.write(str.encode('AT+CMGF=1'+'\r\n')) # Select Message format as Text
mode
    time.sleep(2)

    uart.write(str.encode('AT+CNMI=2,1,0,0,0'+'\r\n')) # New SMS Message
Indications
    time.sleep(2)


    uart.write(str.encode('AT+CMGS="8946003304"'+'\r\n')) #mobile number
receiver
    time.sleep(2)

    uart.write(str.encode('Emergency! Person is in danger, their Location below'
+ '\n'+ 'latitude - '+ str(latitude) + 'longitude - '+ str(longitude) + '\n'+ 'heart rate
- '+str(heart_rate) + 'Temperature -'+str(tempC) +'\n')) # Message
    time.sleep(2)

    uart.write(str.encode("\x1A")) # Enable to send SMS
    time.sleep(2)

    print ("scceass" )

while True:
    lcd.clear() #clear LCD
    lcd.write_message(" Smart Women ", 0, 0) # print String Here, Replace
Double Quotes for String , After Comma set Cursor (Row,Column)
    #print interger Here, if need to print float lcd.write_float()
    lcd.write_message(" Safety System ",1, 0)
    time.sleep(3)#delay 3 seconds
```

```
    for i in range(10):
        lcd.clear() #clear LCD
        lcd.write_message(" Please Place ", 0, 0) # print String Here, Replace
Double Quotes for String , After Comma set Cursor (Row,Column)
        #print interger Here, if need to print float lcd.write_float()
        lcd.write_message(" Your Finger ",1, 0)
        led.on()
        time.sleep(0.5)
        heart=adc.read_u16()
        led.off()
        time.sleep(0.5)
        print(heart)
        if heart>700:
            for i in range(10):
                lcd.clear() #clear LCD
                lcd.write_message(" Heart rate ", 0, 0) # print String Here, Replace
Double Quotes for String , After Comma set Cursor (Row,Column)
                #print interger Here, if need to print float lcd.write_float()
                lcd.write_message(" Calibrating ",1, 0)
                led.on()
                time.sleep(0.1)
                heart=adc.read_u16()
                heart_rate=map_range(heart,800,2000,60,120)
                led.off()
                time.sleep(0.1)
        else:
            heart_rate=0;
        if heart_rate > 0:
            break
    print(heart_rate)
    lcd.clear() #clear LCD
    lcd.write_message(" Heart Rate ", 0, 0) # print String Here, Replace Double
Quotes for String , After Comma set Cursor (Row,Column)
    #print interger Here, if need to print float lcd.write_float()
    lcd.write_number(heart_rate,1, 4)
    lcd.write_message("BPM",1,9);
    time.sleep(3.0)
    ds_sensor.convert_temp()
    time.sleep_ms(750)
    for rom in roms:
        print(rom)
        tempC = ds_sensor.read_temp(rom)
```

```
        tempF = tempC * (9/5) +32
        print('temperature (ºC):', "{:.2f}".format(tempC))
        lcd.write_message(" Temperature ",0,0)
        lcd.write_float(tempC,1,4);
        lcd.write_message(" *C",1,8)
    time.sleep(2.0)
    for i in range(4):
        getGPS(gpsModule)
        print("Printing GPS data...")
        print(" ")
        print("Latitude: "+latitude)
        print("Longitude: "+longitude)
        if latitude or longitude:
            break
        else:
            #10.880332680707173, 77.0222825641159811
            latitude="10.880332680707173"
            longitude="77.02228256411598"
    lcd.clear()
    lcd.write_message("Lat - ",0,0)
    lcd.write_message(latitude,0,5)
    lcd.write_message("Lon - ",1,0)
    lcd.write_message(longitude,1,5)
    time.sleep(2.0)
    if button.value()==0: #condition for button pressed or not
        print("Button pressed")
        panic=1
        send_sms()
    else:
        print("Not pressed")
        panic=0
    final=str(heart_rate)+':'+str(tempC)+':'+str(panic)+':'+latitude+':'+longitude+'\
n'
    iot.write(final)
    print(final)
```

LCD lib code :

```
# lcd_lib.py

import utime
import time
```

```
from machine import Pin

class LCD:
    def __init__(self, rs_pin, e_pin, d4_pin, d5_pin, d6_pin, d7_pin):
        self.rs = Pin(rs_pin, Pin.OUT)
        self.e = Pin(e_pin, Pin.OUT)
        self.d4 = Pin(d4_pin, Pin.OUT)
        self.d5 = Pin(d5_pin, Pin.OUT)
        self.d6 = Pin(d6_pin, Pin.OUT)
        self.d7 = Pin(d7_pin, Pin.OUT)

        self.rs.value(0)
        self.send2LCD4(0b0011) # 8 bit
        self.send2LCD4(0b0011) # 8 bit
        self.send2LCD4(0b0011) # 8 bit
        self.send2LCD4(0b0010) # 4 bit
        self.send2LCD8(0b00101000) # 4 bit, 2 lines, 5x8 dots
        self.send2LCD8(0b00001100) # LCD on, blink off, cursor off
        self.send2LCD8(0b00000110) # Increment cursor, no display shift
        self.send2LCD8(0b00000001) # Clear screen
        utime.sleep_ms(2) # Clear screen needs a long delay

    def pulseE(self):
        self.e.value(1)
        utime.sleep_us(40)
        self.e.value(0)
        utime.sleep_us(40)

    def send2LCD4(self, BinNum):
        self.d4.value((BinNum & 0b00000001) >> 0)
        self.d5.value((BinNum & 0b00000010) >> 1)
        self.d6.value((BinNum & 0b00000100) >> 2)
        self.d7.value((BinNum & 0b00001000) >> 3)
        self.pulseE()

    def send2LCD8(self, BinNum):
        self.d4.value((BinNum & 0b00010000) >> 4)
        self.d5.value((BinNum & 0b00100000) >> 5)
        self.d6.value((BinNum & 0b01000000) >> 6)
        self.d7.value((BinNum & 0b10000000) >> 7)
        self.pulseE()
        self.d4.value((BinNum & 0b00000001) >> 0)
```

```
    self.d5.value((BinNum & 0b00000010) >> 1)
    self.d6.value((BinNum & 0b00000100) >> 2)
    self.d7.value((BinNum & 0b00001000) >> 3)
    self.pulseE()

def lcd_string(self, data):
    for i in range(len(data)):
        self.send2LCD8(ord(data[i]))

def clear(self):
    self.rs.value(0)
    self.send2LCD8(0b00000001)
    utime.sleep_ms(2)

def set_cursor(self, row, col):
    self.rs.value(0)
    if row == 0:
        self.send2LCD8(0x80 + col)
    elif row == 1:
        self.send2LCD8(0xC0 + col)
    utime.sleep_us(100)

def write_message(self, message, row, col):
    self.set_cursor(row, col)
    self.rs.value(1)
    self.lcd_string(message)

def scroll_message(self, message, row, delay=0.5):
    self.set_cursor(row, 0)
    self.rs.value(1)
    self.lcd_string(message)
    time.sleep(delay)
    for i in range(len(message)):
        self.send2LCD8(0b00011000) # Shift the display right
        time.sleep(delay)
def clear_line(self, row):
    self.set_cursor(row, 0)
    self.rs.value(1)
    self.lcd_string(" " * 16) # Clear the entire line

def write_number(self, number, row, col):
    self.clear_line(row)
```

```
        self.set_cursor(row, col)
        self.rs.value(1)
        self.lcd_string(str(number))

    def write_float(self, number, row, col, decimal_places=2):
        self.clear_line(row)
        self.set_cursor(row, col)
        self.rs.value(1)
        format_str = "{:.{}f}".format(number, decimal_places)
        self.lcd_string(format_str)
```

IOT code ESP8266 :

```
#include "Ubidots.h"
   const char* UBIDOTS_TOKEN = "BBUS-
oq95f5UJpBIfWkXwj2YGADtTpTULBV"; // Put here your Ubidots TOKEN
   const char* WIFI_SSID = "blitztech academy_4G"; // Put here your Wi-Fi
SSID
   const char* WIFI_PASS = "Academy@12345"; // Put here your Wi-Fi
password
   Ubidots ubidots(UBIDOTS_TOKEN, UBI_HTTP);
void setup() {
  Serial.begin(9600);
  ubidots.wifiConnect(WIFI_SSID, WIFI_PASS);
}

void loop() {
  if(Serial.available()){
    String myString=Serial.readStringUntil('\n');
    String val1=getValue(myString, ':', 0);
    String val2=getValue(myString, ':', 1);
    String val3=getValue(myString, ':', 2);
    String val4=getValue(myString, ':', 3);
    String val5=getValue(myString, ':', 4);
    ubidots.add("HeartRate", val1.toInt());
    ubidots.add("Temperature", val2.toFloat());
    ubidots.add("panic", val3.toInt());
    ubidots.add("Latitude", val4.toFloat());
    ubidots.add("Longitude", val5.toFloat());
    bool bufferSent = false;
```

```
    bufferSent = ubidots.send(); // Will send data to a device label that matches
the device Id
  }
}
String getValue(String data, char separator, int index)
{
    int found = 0;
    int strIndex[] = { 0, -1 };
    int maxIndex = data.length() - 1;

    for (int i = 0; i <= maxIndex && found <= index; i++) {
        if (data.charAt(i) == separator || i == maxIndex) {
            found++;
            strIndex[0] = strIndex[1] + 1;
            strIndex[1] = (i == maxIndex) ? i+1 : i;
        }
    }
    return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
}
```

APPENDIX II


BANNARI AMMAN INSTITUTE OF TECHNOLOGY

An Autonomous Institution, Affiliated to Anna University, Approved by AICTE, Accredited by NAAC with 'A+' Grade, Sathyamangalam- 638 401, Erode(dt), Tamilnadu, India

Stay Ahead

Department of CSE, CSD, CSBS & AIML

Certificate of Participation

This is to certify that

Prof/Dr/Mr/Ms. NISHA M from KARPAGAM COLLEGE OF ENGINEERING has presented a paper titled IOT-BASED SMART WOMEN SAFETY SYSTEM in **Fourth International Conference on Artificial Intelligence and Smart Computing-2024(ICAISC'24)** held during 14th -16th March 2024 at Bannari Amman Institute of Technology, Sathyamangalam.

Organizer Secretary

Convener

Principal

A1924IC361

BANNARI AMMAN INSTITUTE OF TECHNOLOGY

Stay Ahead

An Autonomous Institution, Affiliated to Anna University, Approved by AICTE, Accredited by NAAC with 'A+' Grade,
Sathyamangalam- 638 401, Erode(dt), Tamilnadu, India

Department of CSE, CSD, CSBS & AIML

Certificate of Participation

This is to certify that

Prof/Dr/Mr/Ms. **PRIYANKAA R** from **KARPAGAM COLLEGE OF ENGINEERING** has presented a paper titled **IOT-BASED SMART WOMEN SAFETY SYSTEM** in **Fourth International Conference on Artificial Intelligence and Smart Computing-2024(ICAISC'24)** held during 14th -16th March 2024 at Bannari Amman Institute of Technology, Sathyamangalam.

Organizer Secretary | Convener | Principal

A1924IC362

BANNARI AMMAN INSTITUTE OF TECHNOLOGY

Stay Ahead

An Autonomous Institution, Affiliated to Anna University, Approved by AICTE, Accredited by NAAC with 'A+' Grade, Sathyamangalam- 638 401, Erode(dt), Tamilnadu, India

Department of CSE, CSD, CSBS & AIML

Certificate of Participation

This is to certify that

Prof/Dr/Mr/Ms. **REVATHI M** from **KARPAGAM COLLEGE OF ENGINEERING** has presented a paper titled **IOT-BASED SMART WOMEN SAFETY SYSTEM** in **Fourth International Conference on Artificial Intelligence and Smart Computing-2024(ICAISC'24)** held during 14^{th} -16^{th} March 2024 at Bannari Amman Institute of Technology, Sathyamangalam.

Organizer Secretary | Convener | Principal

A1924IC363

Stay Ahead

BANNARI AMMAN INSTITUTE OF TECHNOLOGY

An Autonomous Institution, Affiliated to Anna University, Approved by AICTE, Accredited by NAAC with 'A+' Grade,
Sathyamangalam- 638 401, Erode(dt), Tamilnadu, India

Department of CSE, CSD, CSBS & AIML

Certificate of Participation

This is to certify that

Prof/Dr/Mr/Ms. **SANGAVI S** from **KARPAGAM COLLEGE OF ENGINEERING** has presented a paper titled **IOT-BASED SMART WOMEN SAFETY SYSTEM** in **Fourth International Conference on Artificial Intelligence and Smart Computing-2024(ICAISC'24)** held during 14th -16th March 2024 at Bannari Amman Institute of Technology, Sathyamangalam.

Organizer Secretary	Convener	Principal

A1924IC364

APPENDIX III

4/24/24, 10:51 AM Karpagam College of Engineering Mail - Remainder - Session Details for Fourth International Conference on Artificial Intelligence…

KARPAGAM COLLEGE OF ENGINEERING

20L234 NISHA M ECE-B <20l234@kce.ac.in>

Remainder - Session Details for Fourth International Conference on Artificial Intelligence and Smart Computing- Reg

ICAISC24 <icaisc24@gmail.com> Thu, Mar 14, 2024 at 9:04 AM
Bcc: 20l234@kce.ac.in

Dear Participant,

****Gentle Remainder****

Greetings of the Day!
We cordially invite you all on behalf of the Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Sathyamangalam.
The schedule for the "Fourth International Conference on Artificial Intelligence and Smart Computing" is on **14.03.2024 at 09.45 am**. The meet link is given below:

https://meet.google.com/udj-uhap-qev

You are requested to join the meet link at 09.40 AM. without fail.
During presentation,

- The presenter should turn on the camera
- Presentation time: 10 minutes
- Submit the feedback form after completion of your presentation
- Certificate will be provided after feedback has been submitted
- Please verify that your name, paper title, and institution are spelled correctly in the feedback form (The same will be used for Certificate Printing).

4/24/24, 10:53 AM Karpagam College of Engineering Mail - ICAISC'24 - JOURNAL PUBLICATION: IEEE CONFERENCE PROCEEDINGS - REG.

KARPAGAM COLLEGE OF ENGINEERING

20L234 NISHA M ECE-B <20l234@kce.ac.in>

ICAISC'24 - JOURNAL PUBLICATION: IEEE CONFERENCE PROCEEDINGS - REG.

ICAISC24 <icaisc24@gmail.com> Sun, Mar 17, 2024 at 3:58 AM
To: sangeethaasn@bitsathy.ac.in
Bcc: 20l234@kce.ac.in

Dear Authors,
Kindly submit your IEEE formatted paper in the following google form to process for the publication of IEEE Conference Proceedings.
https://forms.gle/yHchyzRvR2NQCF416

Here with We have attached the IEEE Paper Template and Copyrights for your reference.

2 attachments

Copyright form.pdf
106K

Conference-template-A4 (2).doc
63K

APPENDIX IV

Nisha_M.pdf

by Mr Turnitin

Submission date: 09-Apr-2024 12:35PM (UTC+0530)
Submission ID: 2344400169
File name: Nisha_M.pdf (916.76K)
Word count: 14389
Character count: 83497

Nisha_M.pdf

ORIGINALITY REPORT

28%	23%	16%	16%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijetjournal.org Internet Source	2%
2	en.wikipedia.org Internet Source	2%
3	www.nature.com Internet Source	2%
4	A ASUNDI. "MATLAB8 for Photomechanics (PMTOOLBOX)", MATLAB8 for Photomechanics- A Primer, 2002 Publication	1%
5	github.com Internet Source	1%
6	repo.darmajaya.ac.id Internet Source	1%
7	www.coursehero.com Internet Source	1%
8	www.ncbi.nlm.nih.gov Internet Source	1%

Submitted to University of South Australia