UNIVERSITY OF VOCATIONAL TECHNOLOGY

Faculty of Engineering Technology

Department of Electro-Mechanical Technology

EE402040 Internet of Things

(IoT)

Project Proposal

Water Heater Temperature Control and Monitoring System by using ESP32-IoT

Group Members

Name : H.M.Janith odithya bamunugedara (MEC/22/B1/22)

: N.M.A.M.Shehan maduranga (MAN/22/B1/09)

Program : B. tech in Mechatronics Technology

Submission Date : 31st June,2025

Instructed by: Mr. Janith Kasun

1. Introduction

The Internet of Things (IoT) has revolutionized how devices interact, making home automation more intelligent, responsive, and energy-efficient. This project presents an ESP32-based water heater temperature control and monitoring system that allows users to monitor the water temperature in real-time, control the heater relay remotely via Blynk app and web dashboard, and receive safety alerts. The project focuses on safety, efficiency, and remote accessibility, using cost-effective hardware and open-source platforms.

Problem Statement

Conventional water heaters typically operate without intelligent control or remote monitoring capabilities. This leads to several challenges:

- **Energy Waste:** Heaters often run longer than needed, resulting in unnecessary electricity consumption.
- **Inconsistent Temperature:** Manual operation can cause water to be too hot or too cold, impacting comfort and safety.
- **Safety Risks:** Overheating can lead to scalding or even fire hazards, especially when heaters are left unattended.
- **Lack of Convenience:** Users must be physically present to monitor or adjust the heater, which is inconvenient in modern, fast-paced lifestyles.

Applications

The proposed system is highly versatile and can be applied in:

- **Residential homes** for smart water heating and energy savings.
- **Hotels and commercial buildings** for centralized, remote management of multiple heaters.
- **Industrial settings** where precise temperature control is crucial for processes.

- **Healthcare and elderly care facilities** to maintain safe water temperatures and prevent accidents.
- **Remote or off-grid locations** where efficient energy use and remote monitoring are critical.

2. Objectives

- To monitor water temperature in real-time using an IoT-based system.
- To automatically switch ON/OFF the water heater based on set temperature thresholds.
- To allow manual control through a mobile app (Blynk) and a local web dashboard.
- To alert users of overheat conditions through MQTT.
- To design an efficient, low-cost solution using ESP32 and OLED display.
- Components Used

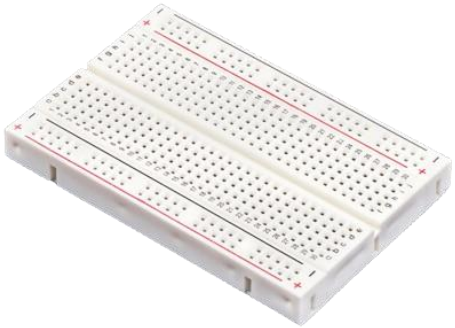
3. Project Structure & Components

3.1. Hardware Structure

- ESP32 Sensor Node
  - DS18B20 waterproof temperature sensor
  - Publishes temperature data via MQTT
- ESP32 Control Node
  - Receives temperature data via MQTT
  - Controls relay connected to water heater
  - OLED display for temperature, mode, relay status
  - Hosts web server for local control
- Relay Module
  - 5V relay to switch water heater power

3.2. Software Components

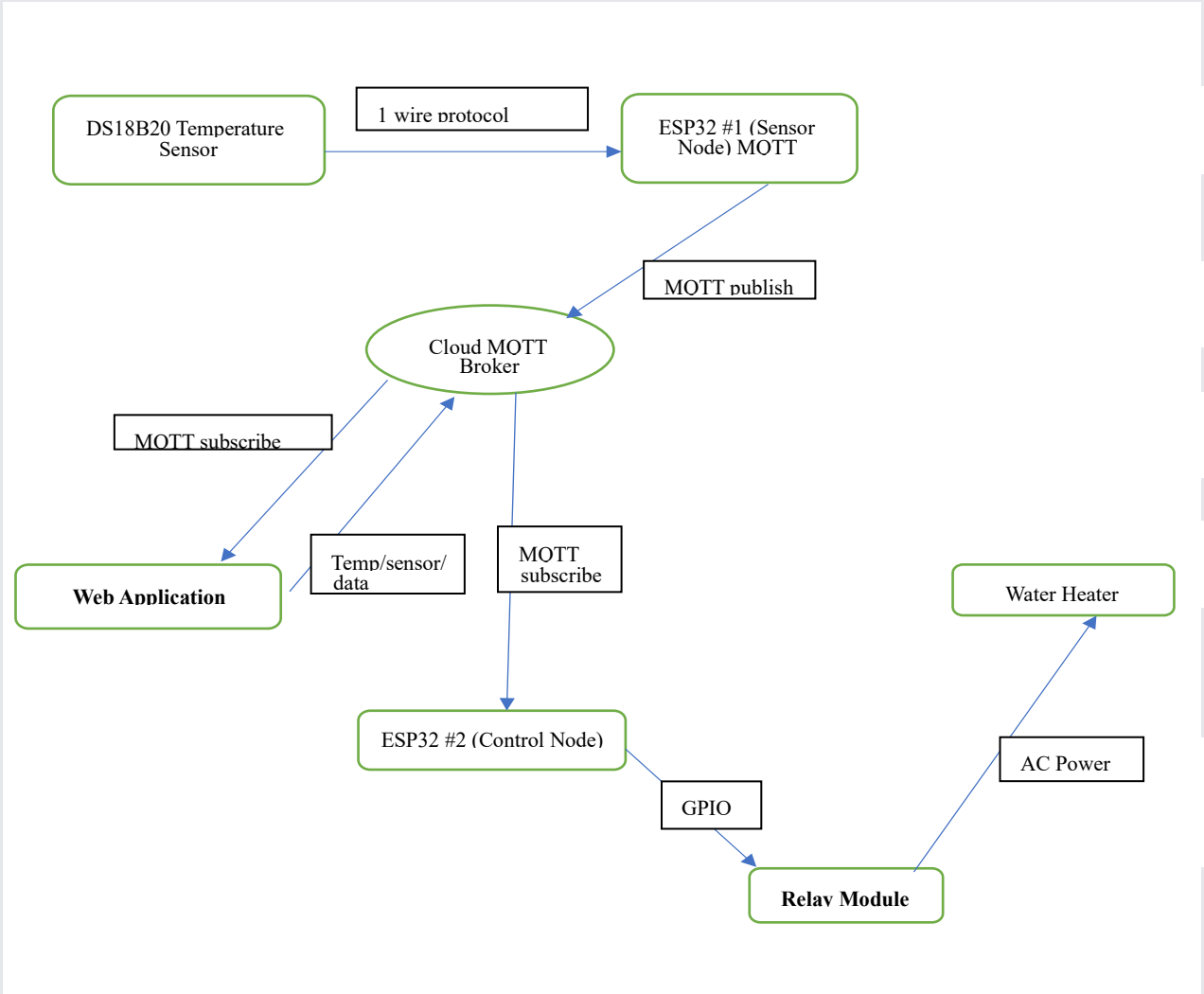
- Arduino IDE for ESP32 programming
- MQTT protocol (broker: broker.hivemq.com)
- Blynk IoT App integration
    - V0: Display Temperature
    - V1: Relay Control (Manual Mode)
    - V2: Relay Status Label
    - V3: Mode Switch (Auto/Manual)
- Web Dashboard (hosted by ESP32 control node)
    - Displays temperature, relay status, and mode
    - Allows switching relay and mode (updated in latest version)
- OLED Display
    - Local feedback for temperature, relay, and mode



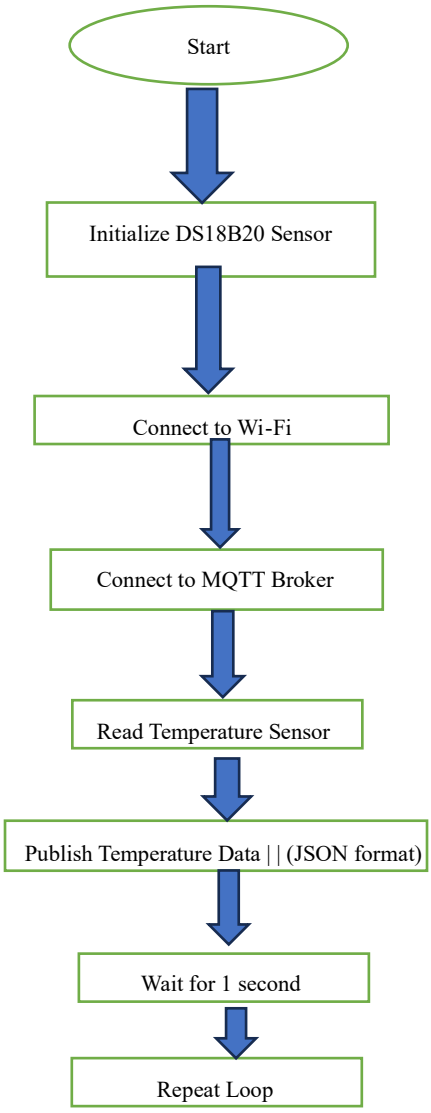
4. System Workflow

- The sensor node reads water temperature and publishes it over MQTT.
- The control node subscribes to this data and applies control logic:
    - In Auto Mode: It toggles the heater based on preset temperatures (e.g., ON if <50°C, OFF if ≥50°C)
    - In Manual Mode: It accepts user commands from Blynk or web dashboard to control the relay
- All status updates are reflected in:
    - OLED display (on ESP32 control node)
    - Blynk mobile app
    - Local web dashboard

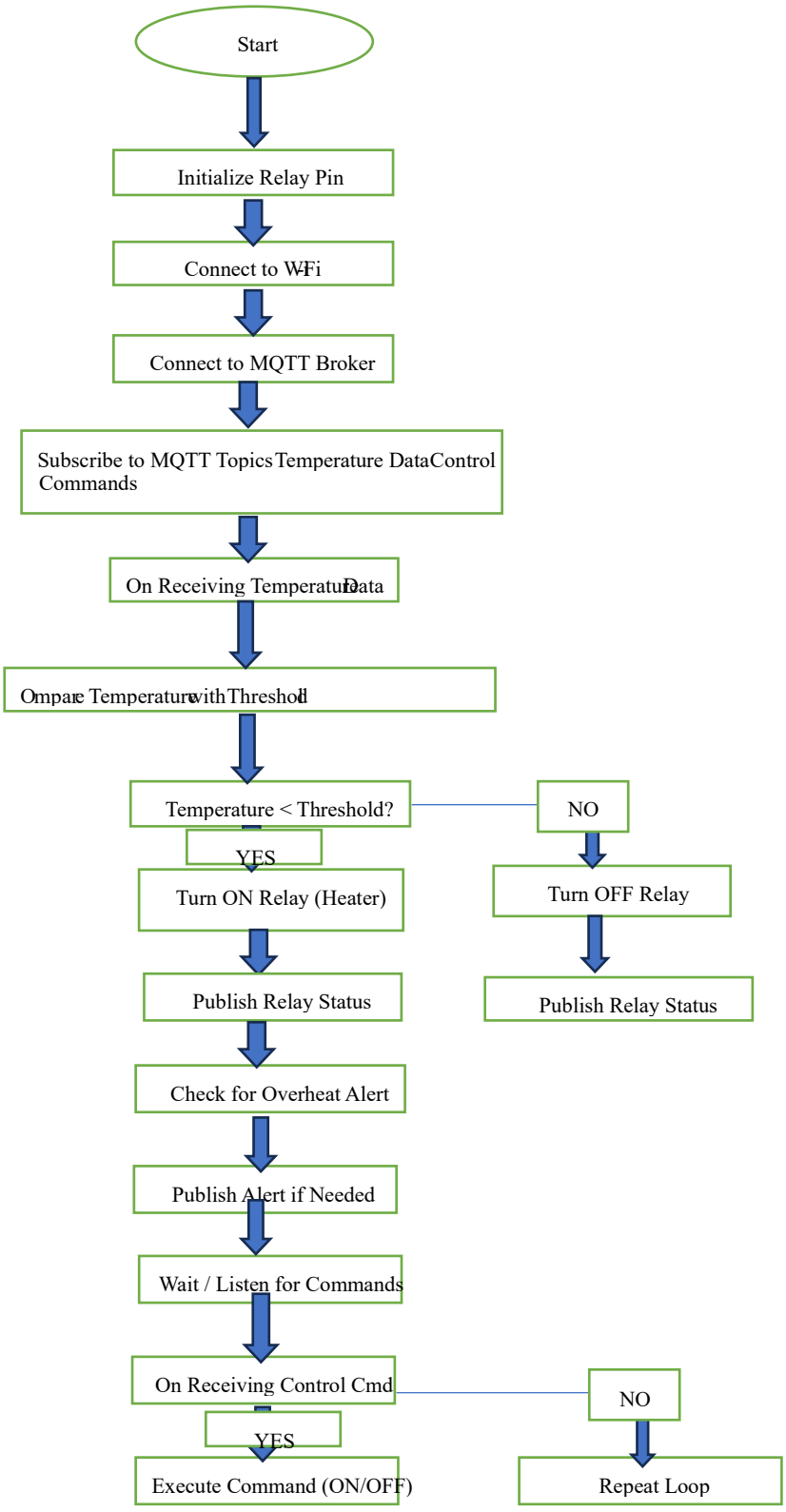
5. High-Level Block Diagram



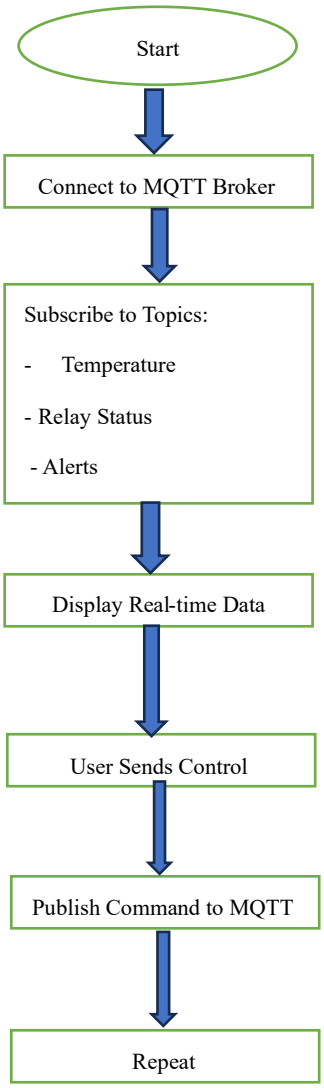
6. Flowchart 1: Sensor ESP32 - Temperature Sensing and Publishing



7. Flowchart 2: Controller ESP32 - Relay Control and Monitoring



8. Flowchart 3: Web Application Interaction



9. Expected Outcomes

- Improved energy efficiency by automatic control
- User safety through overheat detection and alerts
- Convenience through mobile and web control
- Expandability for future features (e.g., scheduling, notifications)

10.Future Enhancements

- Add cloud database (Firebase or ThingsBoard) for logging temperature history
- Add scheduling feature via app or dashboard
- Add notification alerts (email, SMS, push) when abnormal events occur
- Integrate voice assistant or AI for smarter control

11.Bill of Materials

Component	Quantity	Price	Description
ESP32 Dev Board	2	Rs2200.00	Wi-Fi-enabled microcontroller
DS18B20 Temp Sensor	1	Rs900.00	Digital waterproof temperature sensor
Relay Module (5V)	1	Rs.300.00	Controls heater ON/OFF state
OLED Display	1	Rs600.00	Shows temperature and relay status
Jumper Wires	—	Rs200.00	Connections
USB Cable / Adapter	1	Rs300.00	Power source

12.Challenges Faced

- Ensuring accurate real-time temperature readings.
- Maintaining reliable Wi-Fi and Blynk connections.
- Switching between Auto and Manual modes without conflicts.
- Debouncing the relay logic for stable switching.

13.SoftwareDesign

The program is written in Arduino using libraries like BlynkSimpleEsp32, PubSubClient, and Adafruit SSD1306. It follows a state-based control model that continuously checks the temperature and decides whether the relay should be ON or OFF in Auto mode. Manual mode disables automatic control and allows user input via the app or web switch. Data is also displayed on an OLED for local viewing.

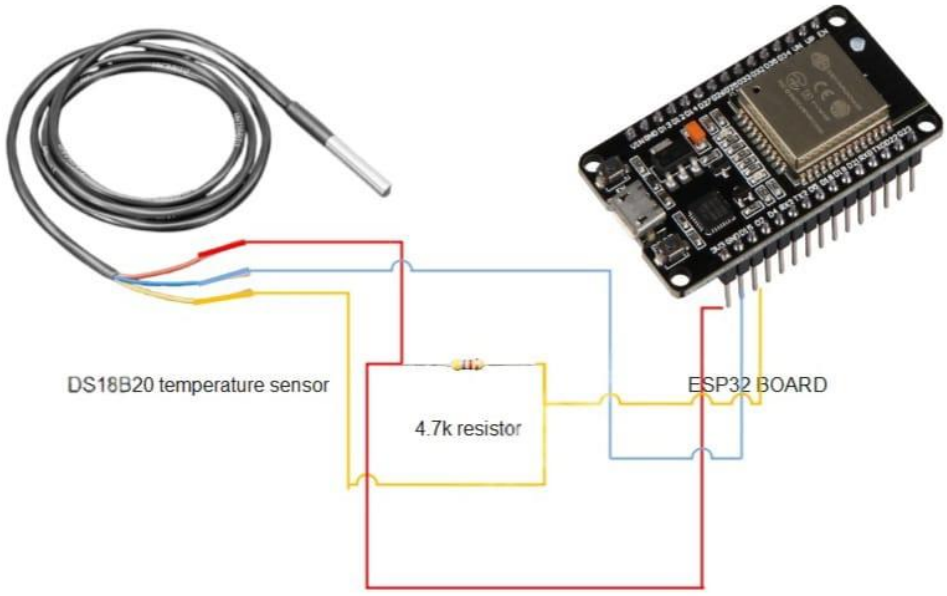
14.Hardware Design

14.1 Wiring Summary

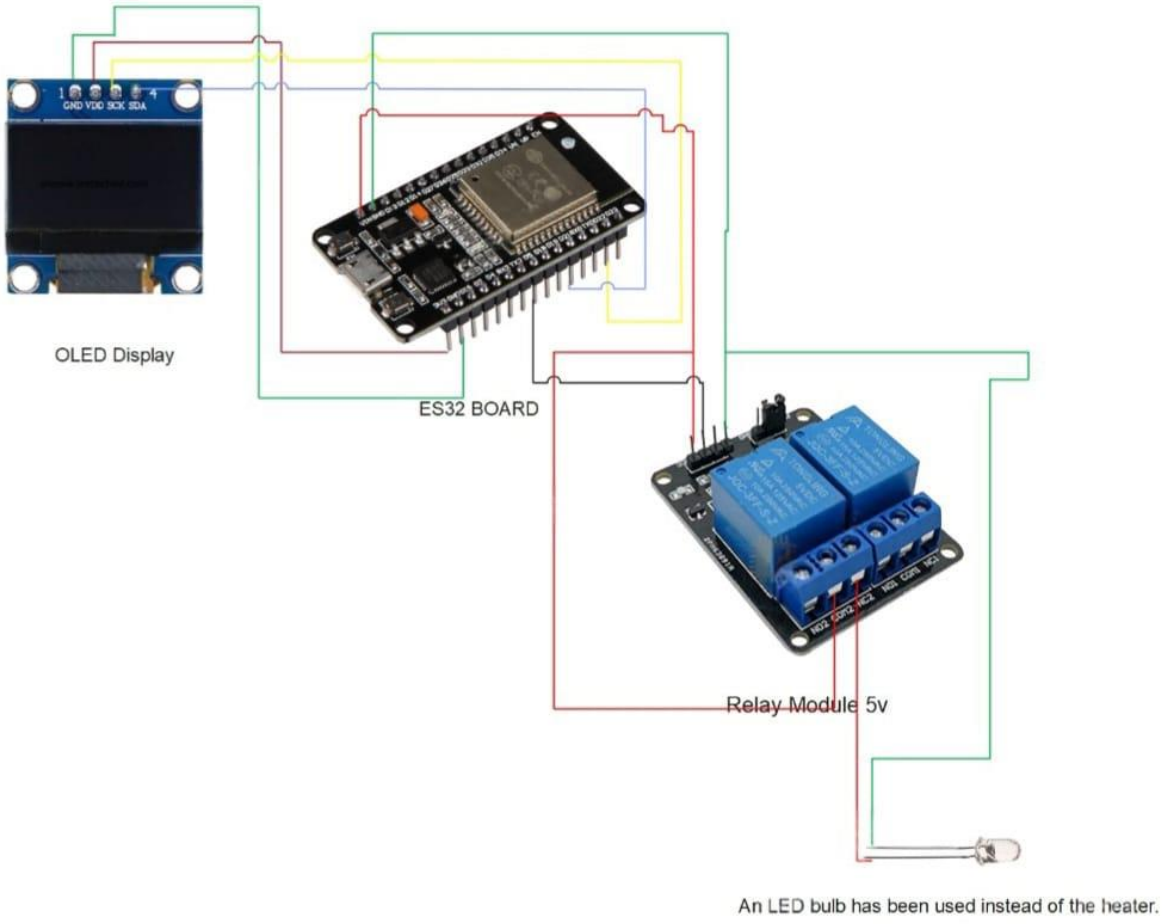
- DS18B20 → GPIO 4 (1-Wire Data)
- Relay Module IN → GPIO 5
- OLED SDA → GPIO 21
- OLED SCL → GPIO 22
- 5V and GND connected to all components

14.2 Suggested Block Diagram:

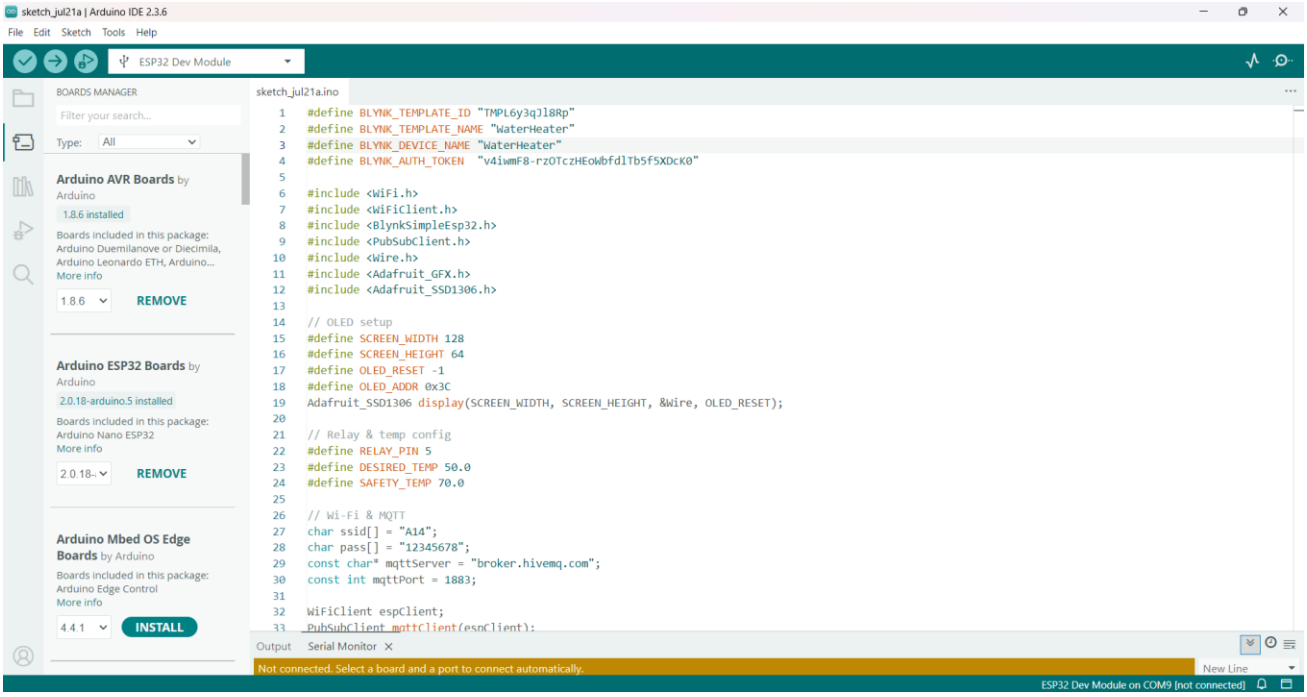
Sensor node wiring block diagram



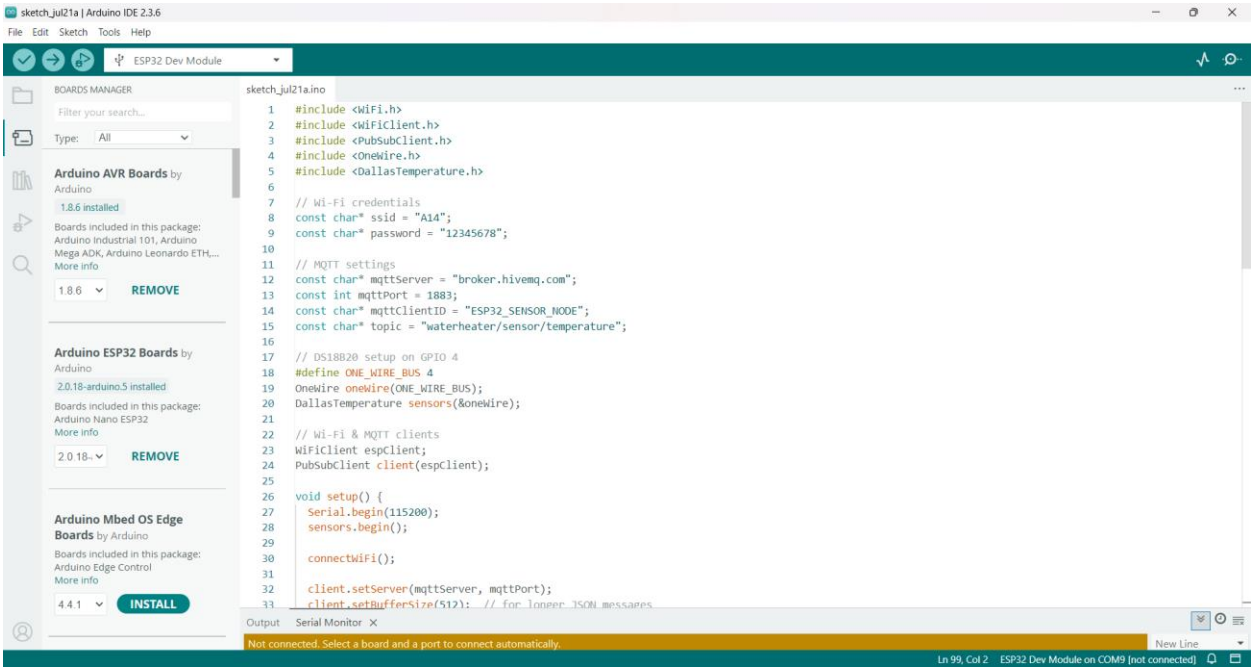
control node wiring block diagram



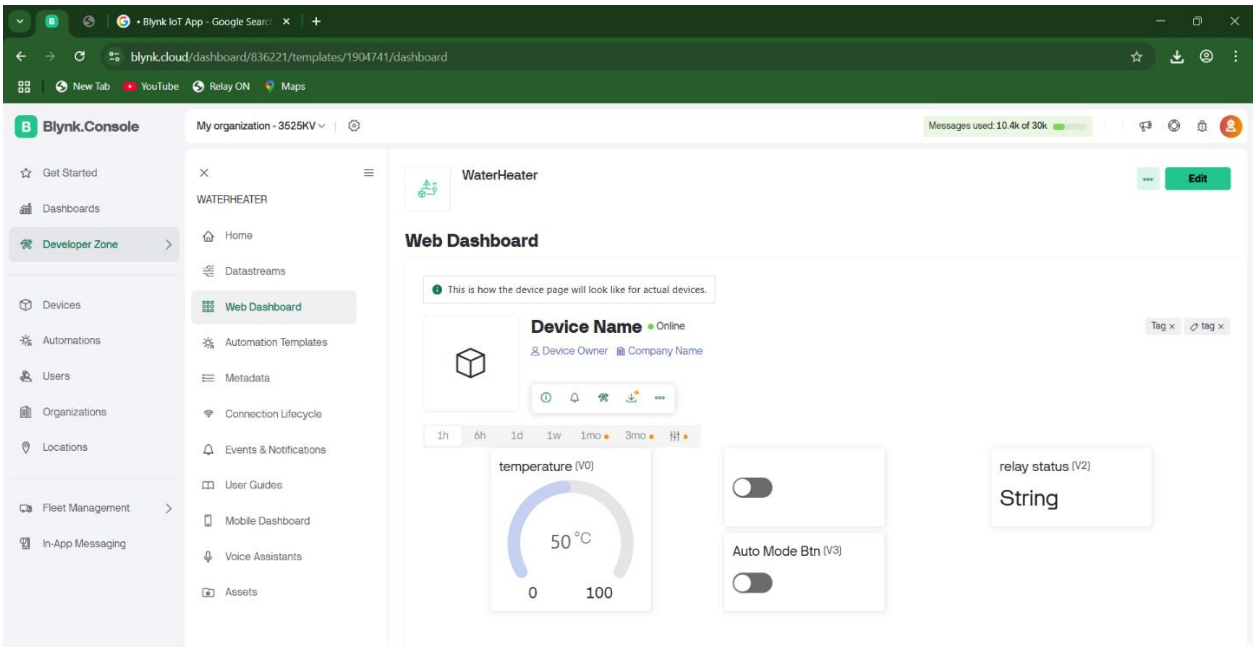
15.Esp32 control node code

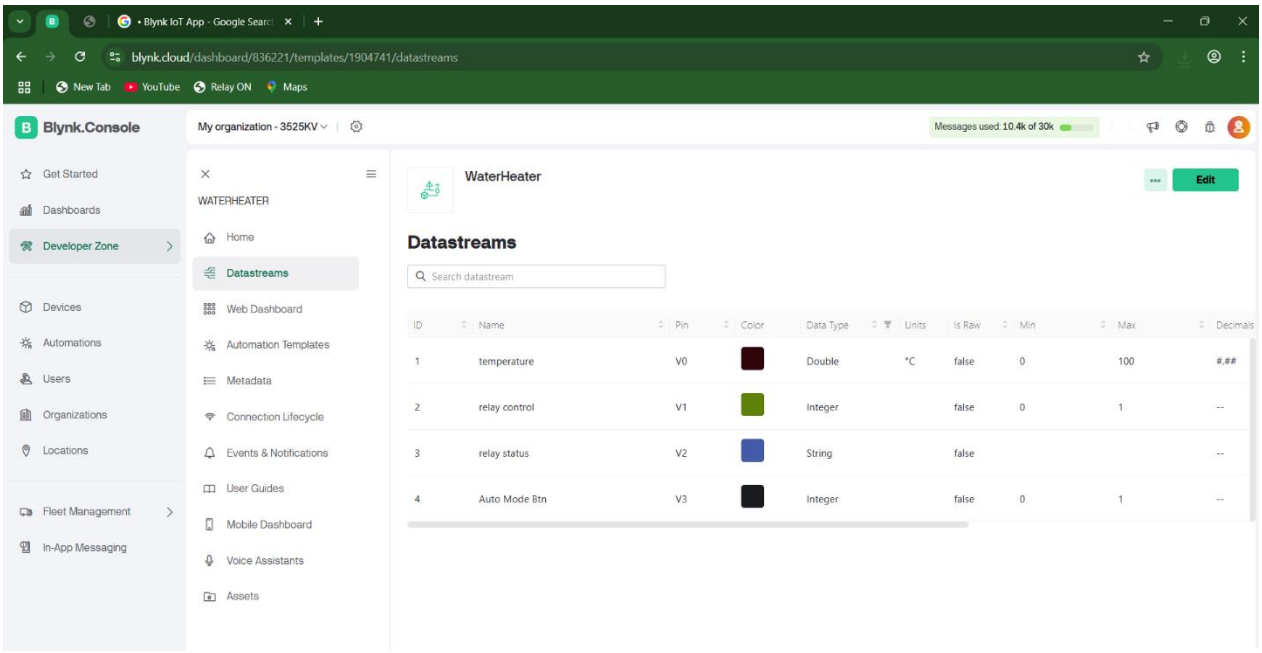


16.Esp32 sensor node code

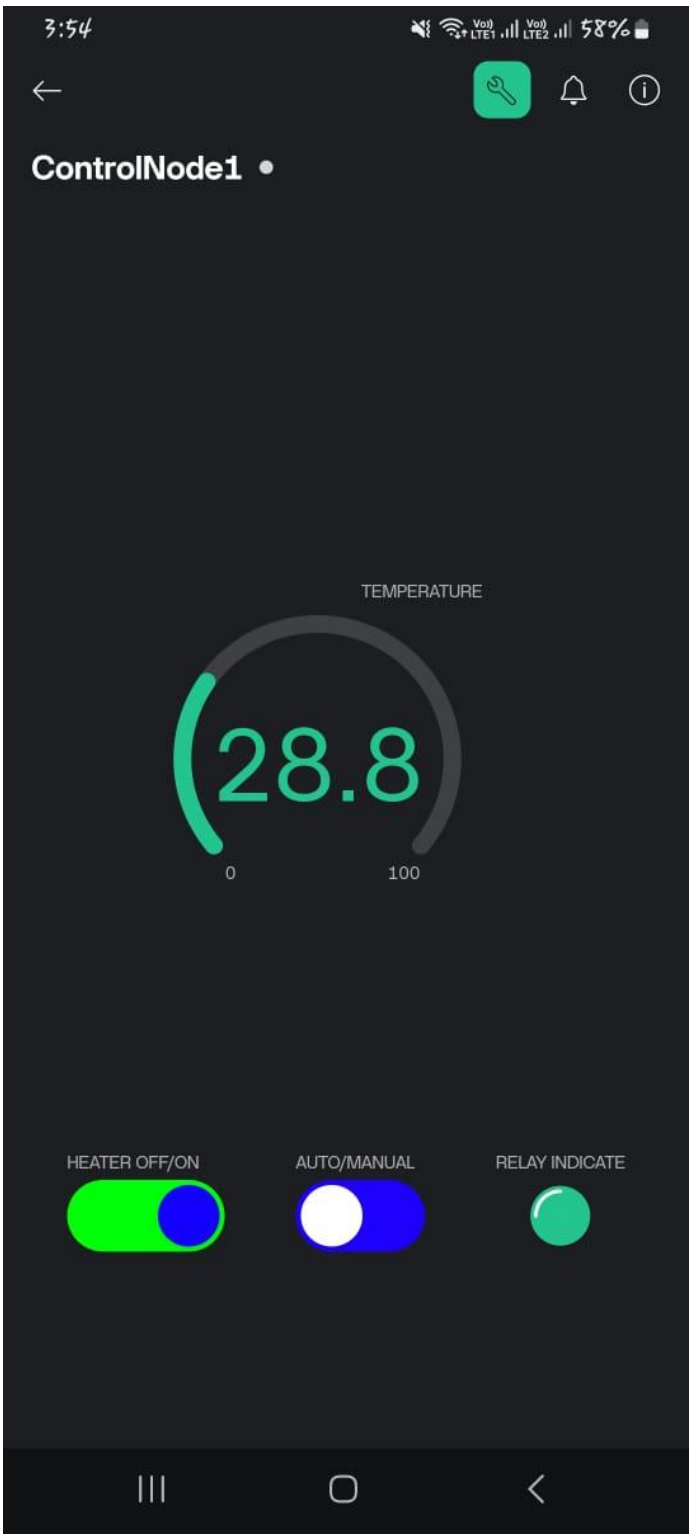


17.Blynk IOT

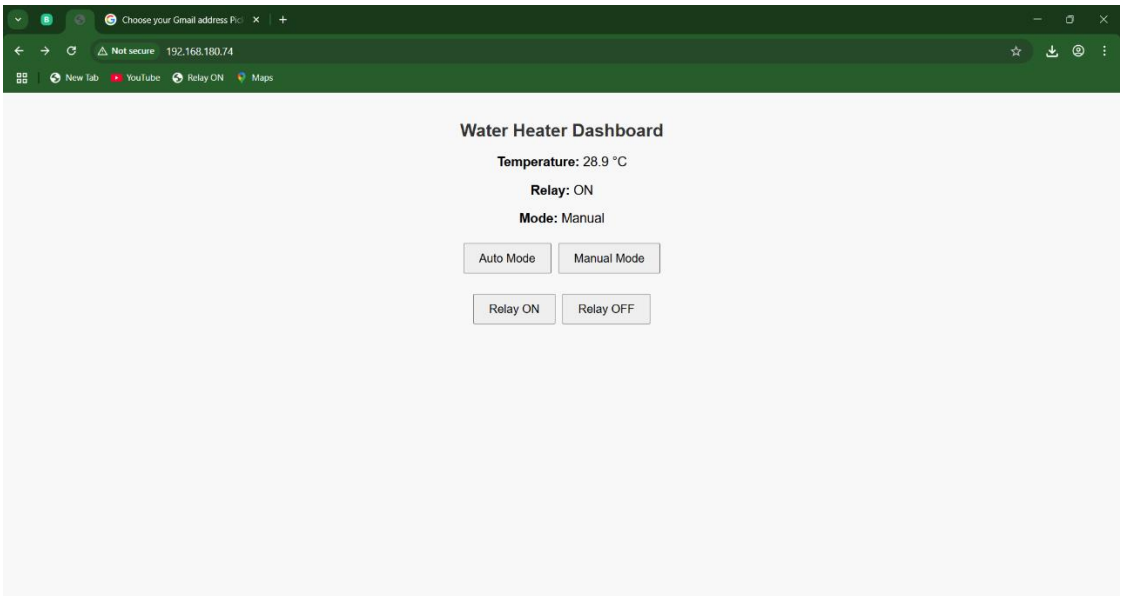

Blynk IoT App - Google Search
blynk.cloud/dashboard/836221/templates/1904741/dashboard
New Tab
YouTube
Relay ON
Maps
Blynk.Console
My organization - 3525KV
Messages used: 10.4k of 30k
Get Started
Dashboards
Developer Zone
Devices
Automations
Users
Organizations
Locations
Fleet Management
In-App Messaging
WATERHEATER
Home
Datastreams
Web Dashboard
Automation Templates
Metadata
Connection Lifecycle
Events & Notifications
User Guides
Mobile Dashboard
Voice Assistants
Assets
WaterHeater
Edit
Web Dashboard
This is how the device page will look like for actual devices.
Device Name
Online
Device Owner
Company Name
Tag
tag
1h
6h
1d
1w
1mo
3mo
temperature (V0)
50 °C
0
100
Auto Mode Btn (V3)
relay status (V2)
String


Blynk IoT App - Google Search
blynk.cloud/dashboard/836221/templates/1904741/datastreams
New Tab
YouTube
Relay ON
Maps
Blynk.Console
My organization - 3525KV
Messages used: 10.4k of 30k
Get Started
Dashboards
Developer Zone
Devices
Automations
Users
Organizations
Locations
Fleet Management
In-App Messaging
WATERHEATER
Home
Datastreams
Web Dashboard
Automation Templates
Metadata
Connection Lifecycle
Events & Notifications
User Guides
Mobile Dashboard
Voice Assistants
Assets
WaterHeater
Edit
Datastreams
Search datastream
ID
Name
Pin
Color
Data Type
Units
Is Raw
Min
Max
Decimals
1
temperature
V0
Double
°C
false
0
100
#.##
2
relay control
V1
Integer
false
0
1
--
3
relay status
V2
String
false
--
4
Auto Mode Btn
V3
Integer
false
0
1
--

The Blynk IoT platform is used to remotely monitor and control the ESP32-based water heater system via a mobile application. Blynk provides a user-friendly interface to interact with hardware using virtual pins and widgets, enabling real-time data visualization and remote control.



18.Webdashboad



The Web Dashboard provides a browser-based interface for monitoring and controlling the water heater system in real time. It acts as an alternative to the Blynk mobile app and is especially useful for local access via Wi-Fi without needing cloud connectivity.

19.Youtube video link

https://youtu.be/aLHNA26IkYE

20.github link

https://github.com/Janithbamunugedara/Water-Heater-Temperature-Control-and-Monitoring-System-by-using-ESP32-.git

21.Conclusion

implements an ESP32-based smart water heater controller. It balances automated and manual control through user-friendly interfaces while keeping the setup compact and energy-efficient. Without using overheat alerts, the system focuses on essential functionality: real-time monitoring and simple relay control, making it ideal for small homes or paper and pulp industry setups..

22.References

- Blynk IoT Platform Documentation
- Adafruit SSD1306 Arduino Library
- DS18B20 Temperature Sensor Datasheet
- Arduino IDE Reference
- PubSubClient for MQTT Communication
- Arduino Expert. "Water Heater Temperature Monitoring and Control by Using ESP32 IoT." 2025.
- Instructables. "ESP32 IoT Water Heater." 2024.
- Vishvas Natu, Vedant Hinge, Harsh Shankar. "Design and Implementation of an Automated Geyser Temperature Control System Using ESP32." International Journal of Research in Advent Technology, 2025. DOI: 10.22214/ijraset.2025.72481
- Reddit. "HA ESP32 Dallas Temp Sensor Tado Hot Water Control." /r/homeassistant, 2024.
- GitHub. "Smart Solar Heating Project." UnknownHero99, 2021.
- SSRN. "ESP32 Temperature Control Project With An Emphasis On Energy Efficiency." 2024.