**Charging Station PCB Documentation**

* **Revision Number:**  
  Rev. 2.0
* **Date of Creation or Update:**  
  14/12/2024
* **Prepared By:**

Muhammad Sabahat Akhter  
Muhammad .Sulaiman Shahid  
Maria Tabassum

* **Reviewed By:**
* **Approved By**

**Table of Contents**

1. Introduction
2. Sensors Work Flow Diagrams

* Lock operation
* Flame sensor
* Temperature Sensor
* RGB led

1. Micro-controller cumulative Work flow with Sensors
2. Charging Station single panel circuit working Details.

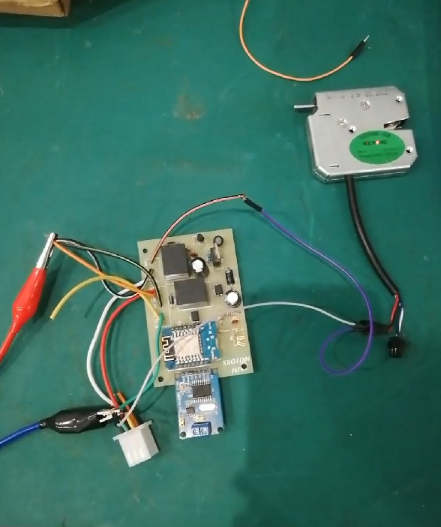
* Power supply
* Data acquisition and processing
* Data Communication

1. Charging Station PCB Flow Diagram
2. Master- Slave flow
3. Operation on local Server
4. Operation on Cloud Server
5. Comparison with previous revision
6. Troubleshooting the Charging Station Circuit

**1. Introduction**

This document outlines the comprehensive details of the integration and working of the charging station circuit. It highlights the functionality of connected sensors, microcontrollers, and communication modules for effective operation. It also covers troubleshooting methods to resolve common issues encountered during operation.

**2. Sensors Workflow Diagrams**

**2.1 Electromagnetic Lock Operation**

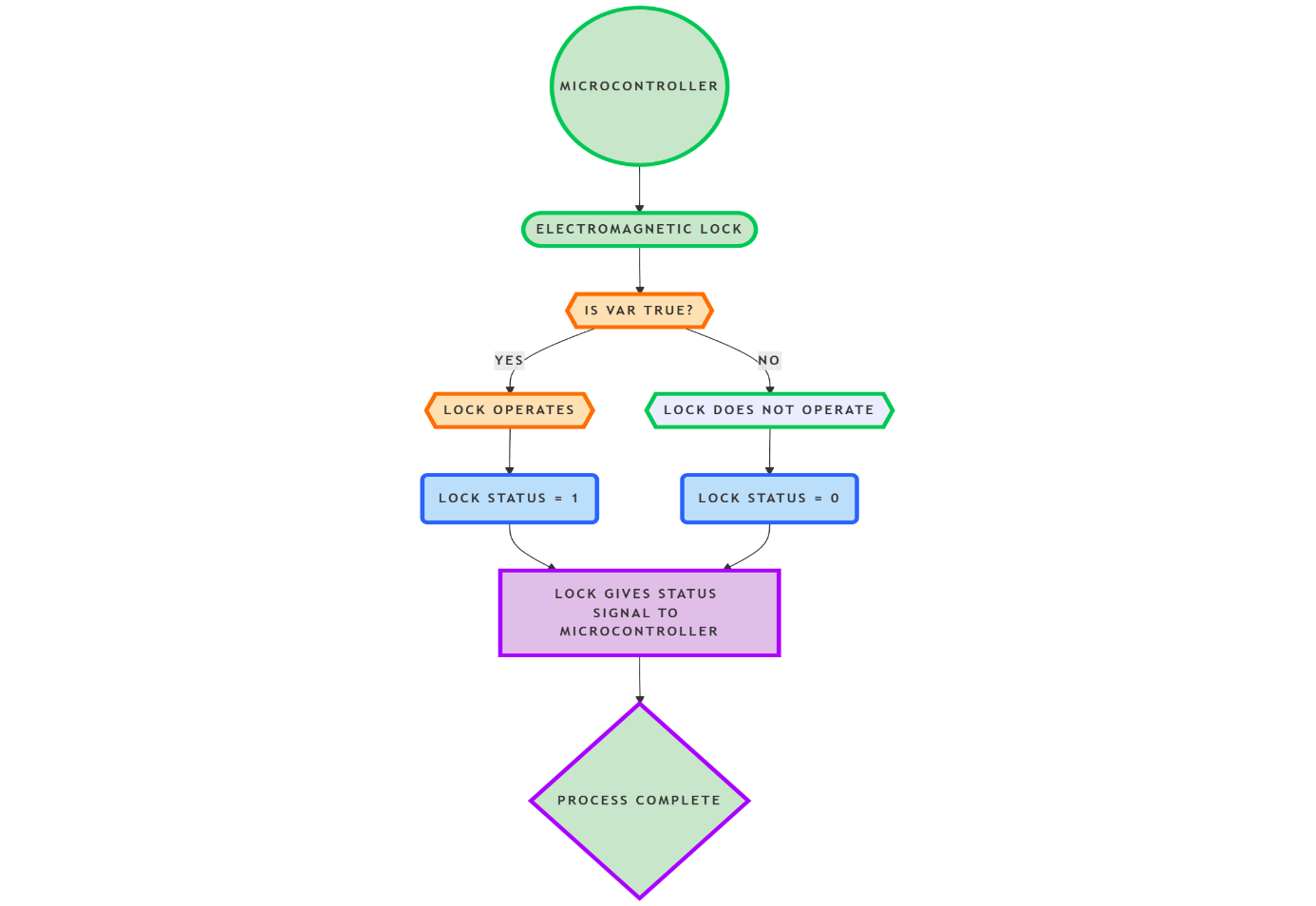
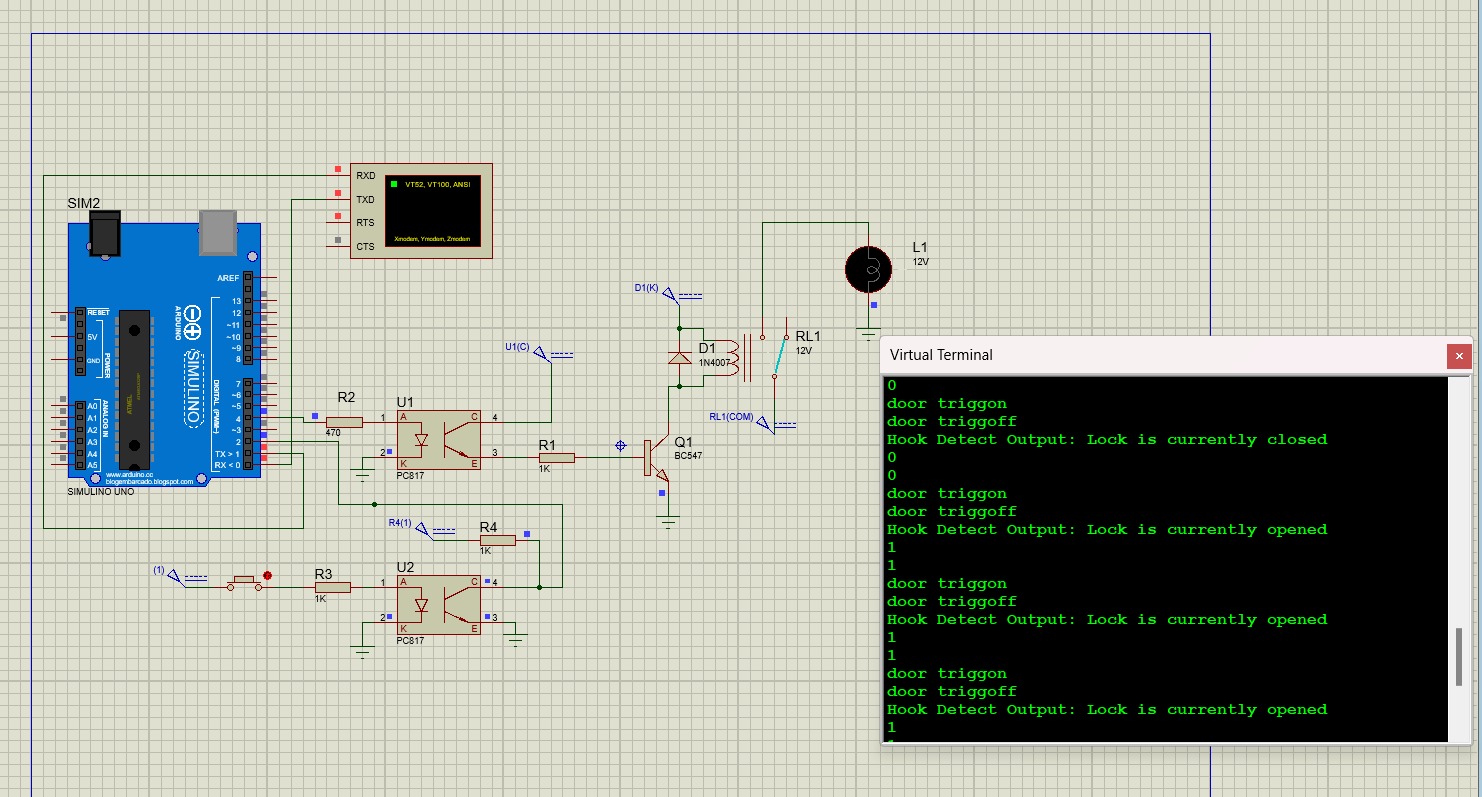
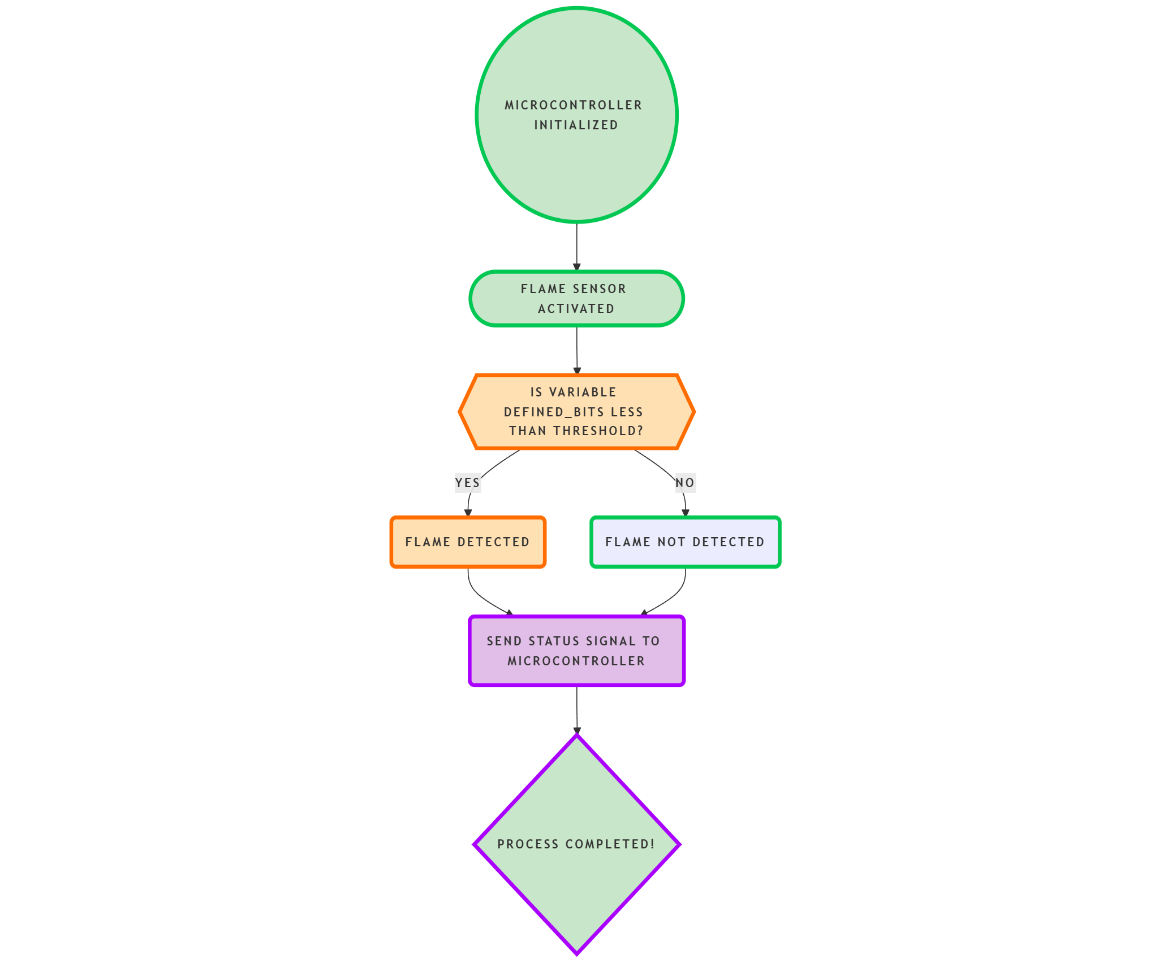
* Electromagnetic Lock operates whenever there is an operating command from micro-controller.
* The electromagnetic lock also triggers a signal to the microcontroller for its state monitoring.
* Figure 3 shows the simulation a lock is being operate through optocoupler and relay mechanism. Micro-controller gives signals to optocoupler which then triggers the relay to actuate the lock, Whereas, the electromagnetic lock gives back its status signal through its status pin which gives input to the micro-controller through optocoupler.

Fig1:lock circuit

Figure 2: Electromagnetic lock flow chart



 Figure 3: Electromagnetic lock function simulation

**2.2 Flame Sensor**

* Monitors potential fire hazards in the charging area.
* Outputs an alert signal to the microcontroller when a threshold temperature is detected.

Figure 4: flame sensor flow chart

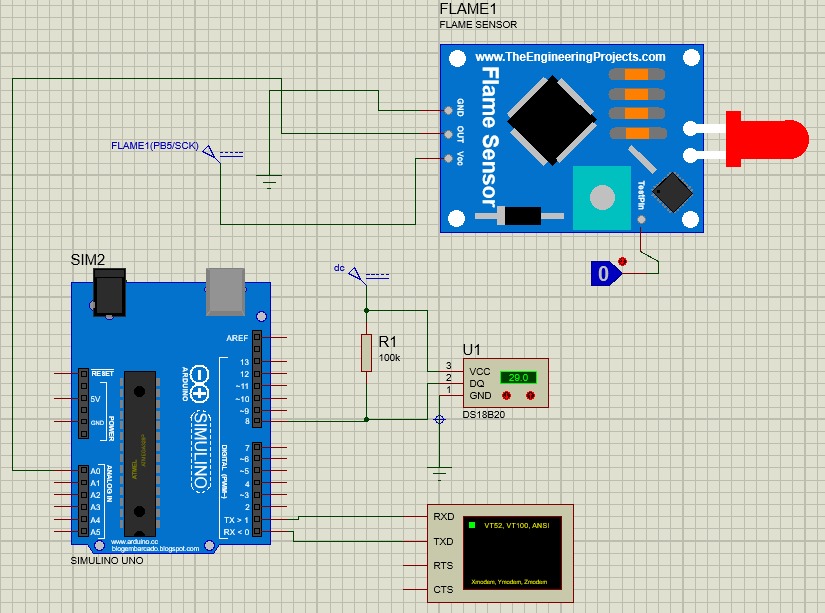
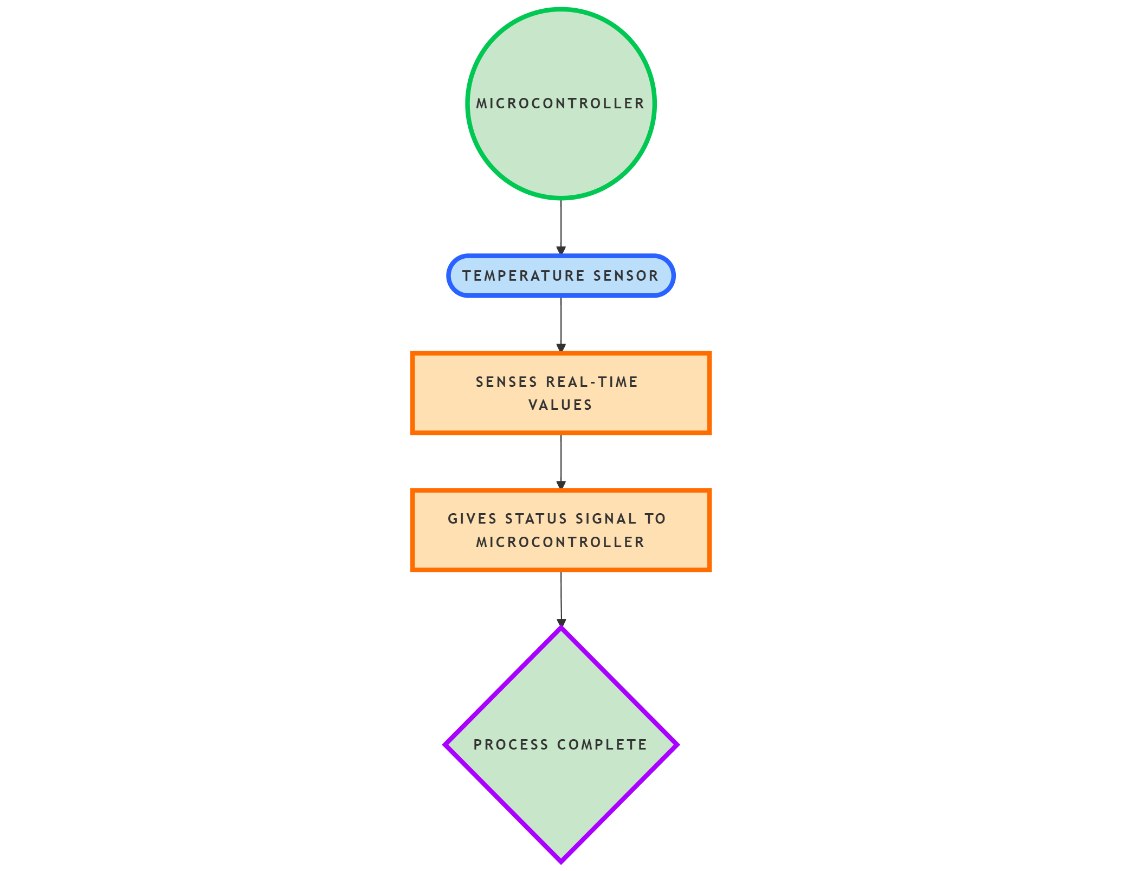


Figure 5: flame and temperature sensor simulation

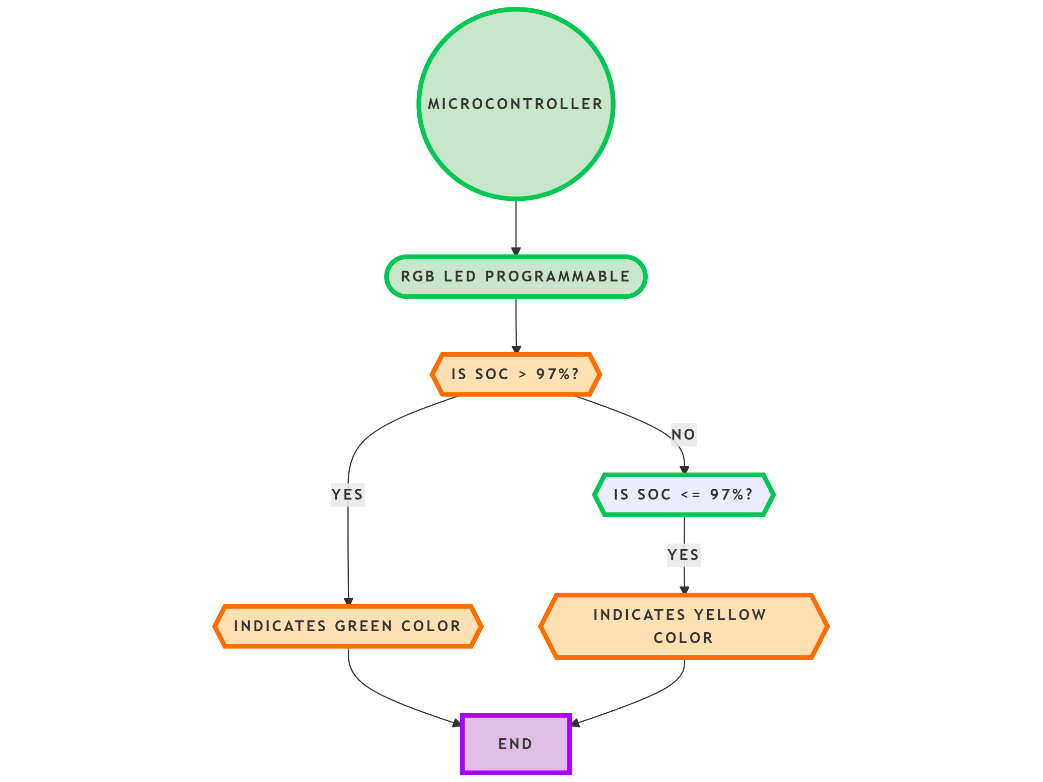


**2.3 Temperature Sensor**

* Continuously measures the ambient temperature to ensure safe operating conditions.
* Provides data to the microcontroller for real-time temperature monitoring.

Figure 6: Temperature sensor flow chart

**2.4 RGB LED**

* Controlled by the microcontroller based on the status of the system.
* It indicate the green color whenever the SOC of the battery will be greater than 97% and will indicate yellow color whenever the given SOC is below 97%.
* The Led can be programmed according to different conditions.

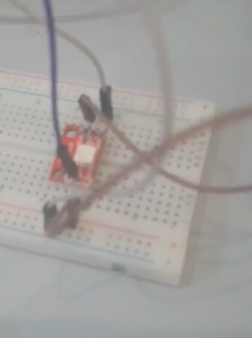
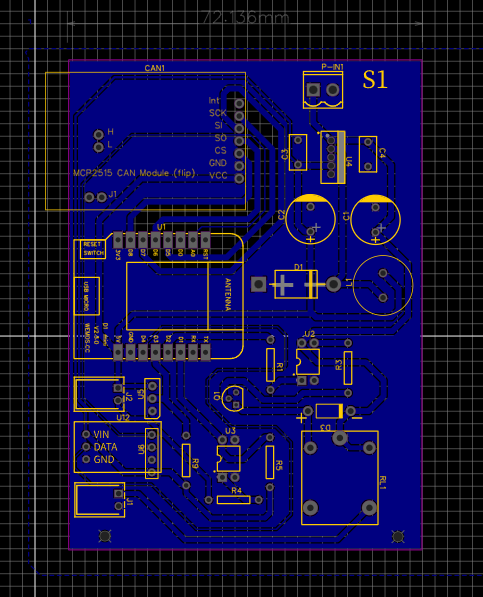


Figure 7: RGB led circuit

Figure 8: RGB led flow chart

**3. Microcontroller Cumulative Workflow with Sensors**

* **Data Input**: Microcontroller collects data from lock, flame, and temperature sensors and CAN module.
* **Data Processing**: Processes sensor data to determine the operational state and identify issues.
* **Data Output**: Controls the RGB LED to display system status and communicates data to its Master micro-controller wirelessly.

**4. Charging Station Single Panel Working Details**

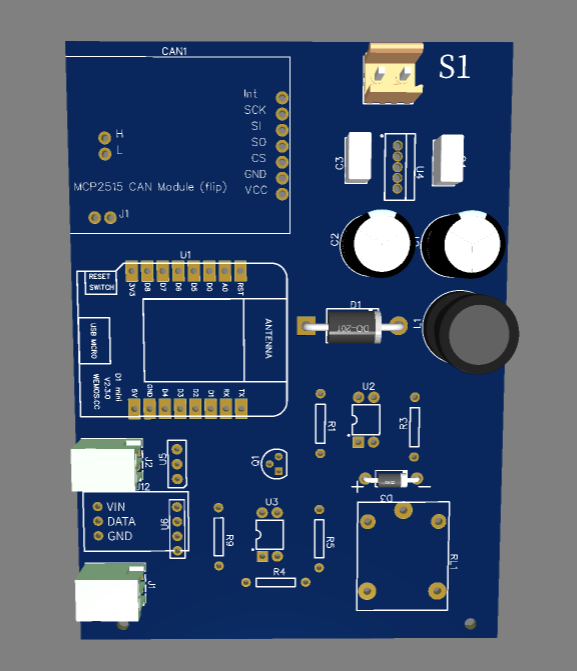
****

Figure 8: Charging Station Panel PCBFigure 9: Charging station Panel Circuit

**4.1 Power Supply**

* Utilizes a 12v supply to provide regulated DC voltage throughout the circuit.
* Step-down voltage regulator LM2576t , capacitors, inductors, and diodes used to drive 5v throughout the circuits where necessary.
* Ensures stable power for all modules, including sensors and communication devices.

**4.2 Data Acquisition and Processing**

* Data from sensors (lock, flame, temperature) is acquired via GPIO pins digitally and uses CAN module to fetch data from Battery packs.
* The microcontroller processes the data for real-time monitoring and system control.

**4.3 Data Communication**

* Utilizes CAN communication for seamless data transmission through battery packs.
* Utilizes ESP-NOW communication to transmit data from slaves to master micro-controller

**Single Slave working Flowchart**

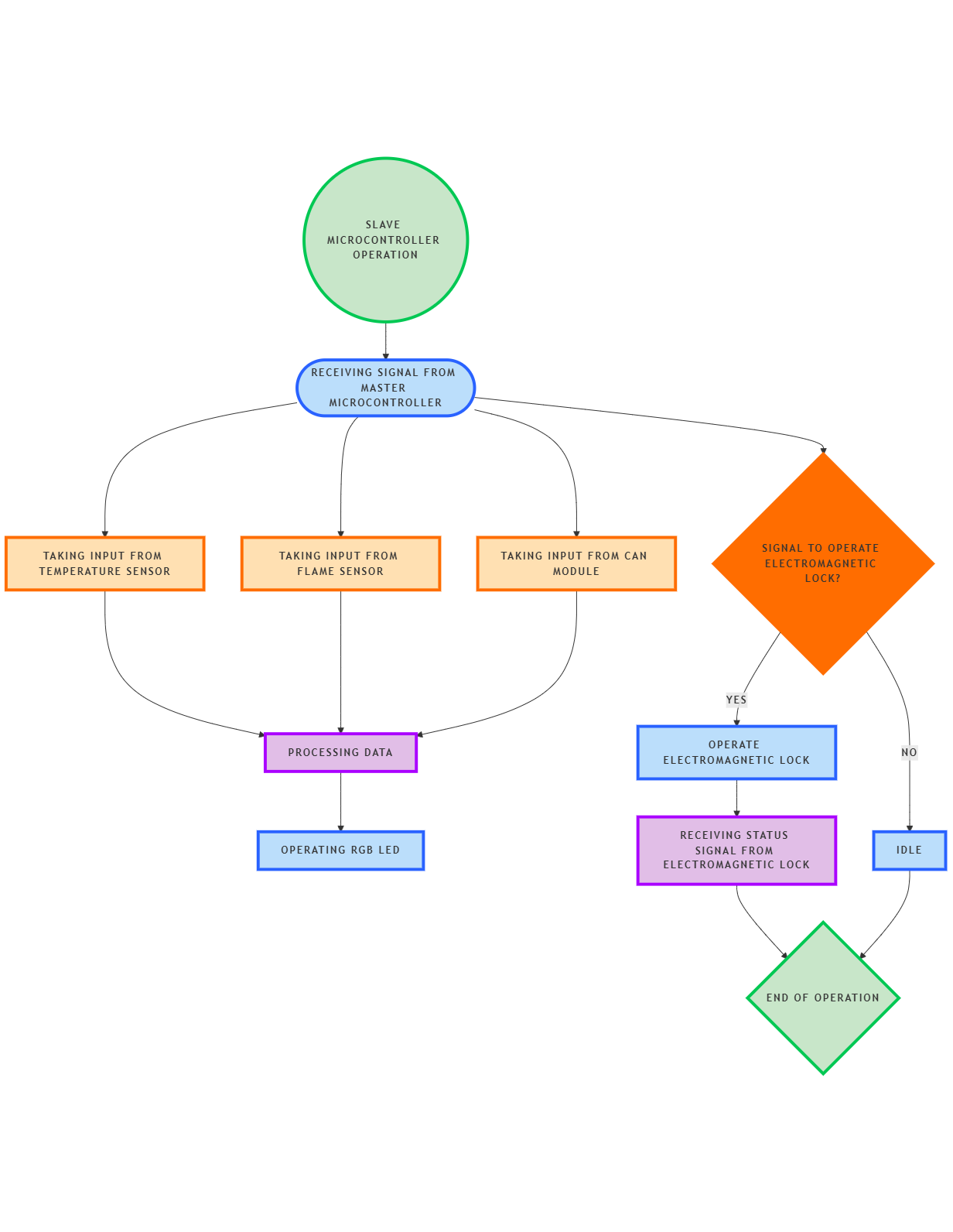


Figure 10: Single Slave working Flowchart

**5. Charging Station PCB Flow Diagram**

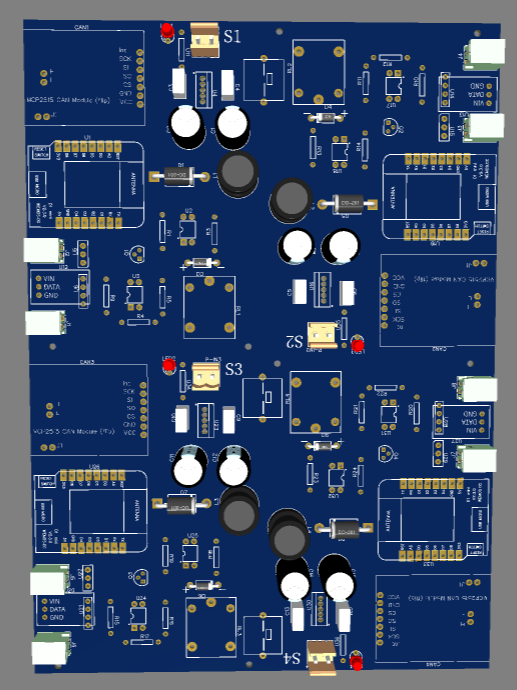
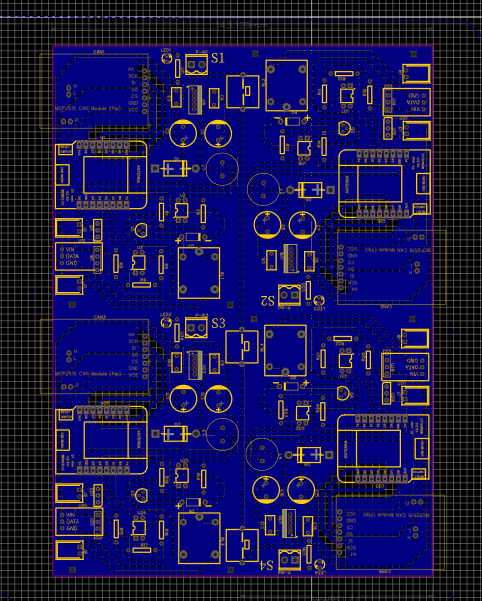
****

Figure 11: Charging Station PCB Figure 12: Charging Station Circuit

Each PCB of charging can handle 4 slave micro-controller which then process the Data and transmit it to Master Micro-controller.

1. **Power Supply Initialization**: Powers all connected components.
2. **Sensor Monitoring**: Sensors send real-time data to the microcontroller.
3. **Data Processing**: Microcontroller analyzes data for decision-making.
4. **Communication**: Transmits processed data via wirelessly to its Master.
5. **System Feedback**: RGB LED displays the operational state of the charging station.

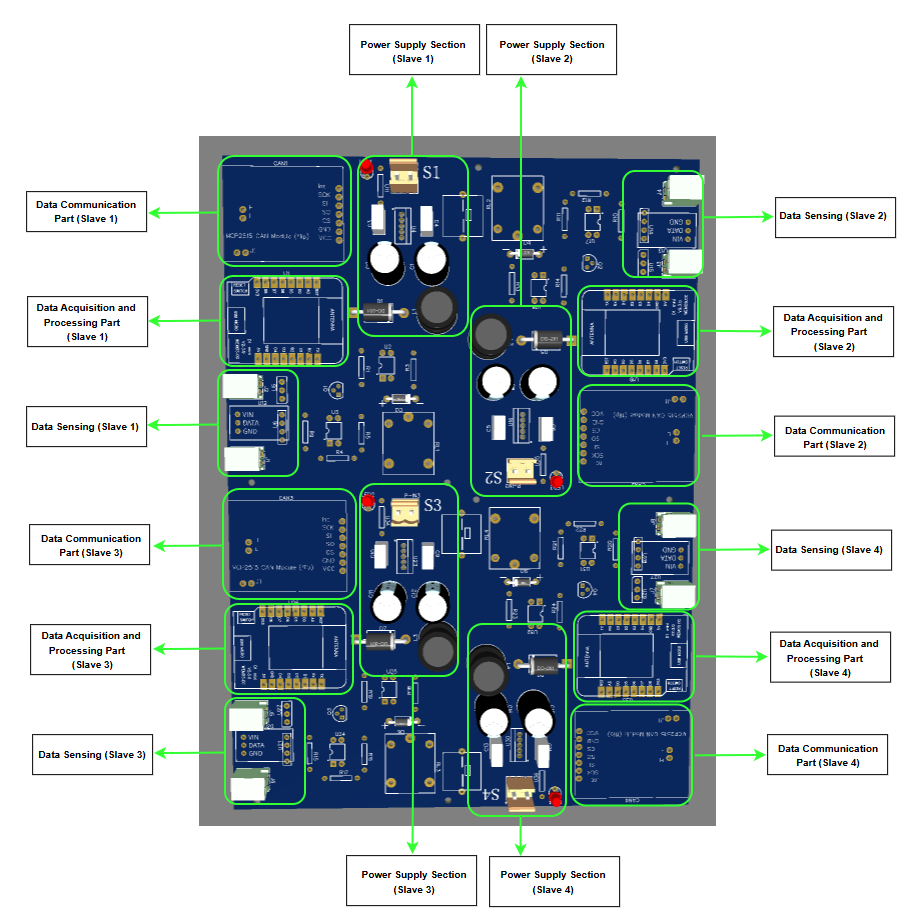


Figure 12: PCB description

**6. Master – Slave Flow**

**GPIOs CONNECTIVITY ON MCU (Single Slave)**

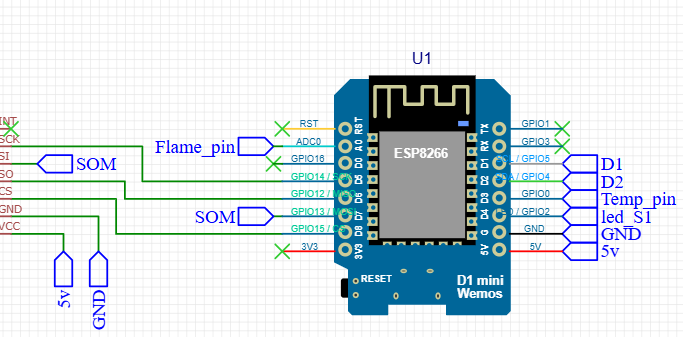
****

Figure 13: esp8266 di mini – sensors connections

|  |  |
| --- | --- |
| **MCU** | **Sensor** |
| GPIO0 | Temperature Sensor (One Wire) |
| GPIO2 | Slot Indication LED (Data Pin) |
| GPIO1 | Cabinet Lamp |
| GPIO4 | Electromagnetic Lock Operating Pin |
| GPIO5 | Electromagnetic Lock Status Pin |
| A0 | Aerosol Fire Extinguisher Pin |
| GPIO12-GPIO15 | MCP CAN Module Connection Pins for Battery Specs |

Total: 10 GPIOs of the MCU are getting utilized.

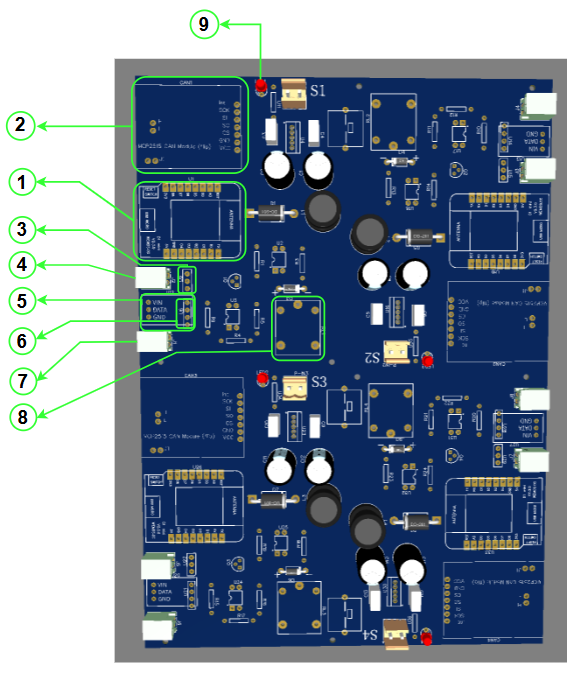


Figure 14: Main Components of pcb

**ESP8266 Wemos D1 mini (Component 1):** The ESP8266 D1 mini is a versatile and cost-effective Wi-Fi module designed to enable seamless internet connectivity for a wide range of devices. It serves as a bridge between the physical world and the digital realm, facilitating data exchange and remote control.

**Working Principle:** At its core, the ESP8266 operates as a System on a Chip (SoC), integrating a 32-bit microcontroller with built-in Wi-Fi capabilities. This allows it to both process data locally and communicate wirelessly over Wi-Fi networks. The module supports the 802.11 b/g/n Wi-Fi standards, ensuring compatibility with a variety of network configurations.

**Functionality:** It performs the functionality of data acquisition, processing and transmission. It fetches data from the sensors (Temperature, flame and CAN) and operates the modules (LED and Lock) attached. After acquiring data from the sensors, it processes the received data, performs calculations, and make decisions based on it. For example, after receiving data from the CAN module, it performs data encoding and computation.

After fetching data from the sensors and modules, ESP8266 transmits the processed data to the “*master”* ESP8266 via “*ESP-NOW”*. It is a protocol developed by Espressif, which enables multiple devices to communicate with one another without using Wi-Fi. The protocol is similar to the low-power 2.4GHz wireless connectivity. The pairing between devices is needed prior to their communication. After the pairing is done, the connection is safe and peer-to-peer, with no handshake being required*.*

**MCP2515- (Component 2):** The MCP2515 is a high-speed CAN (Controller Area Network) transceiver that enables reliable and efficient communication between electronic control units (ECUs) in automotive and industrial applications. It offers a cost-effective and flexible solution for implementing CAN bus networks.

**Working Principle:** The MCP2515 operates on the CAN bus protocol, a robust and reliable communication protocol that allows multiple devices to communicate with each other without a central controller. It employs a carrier-sense multiple access with collision detection (CSMA/CD) mechanism to ensure reliable data transmission.

**Functionality:** When the system is powered on, the MCP2515 initiates communication with the BMS over the CAN bus. This high-speed communication protocol allows for efficient and reliable data exchange between the two devices.

The BMS, in turn, collects real-time data from various sensors within the battery pack. These sensors monitor crucial parameters such as:

* **Battery Voltage:** The overall voltage of the battery pack.
* **Battery Current:** The current flowing into or out of the battery.
* **State of Charge (SoC):** The percentage of remaining charge in the battery.
* **Temperature:** The temperature of individual cells within the battery pack.
* **Battery Capacity:** The total energy storage capacity of the battery.
* **Individual Cell Voltages:** The voltage of each individual cell within the battery pack.

Once the BMS collects this data, it transmits it to the MCP2515 via the CAN bus. The MCP2515 then receives and processes this information, making it available for further use by other components in the system, such as the ESP8266.

**DS18B20- (Component 3):** A versatile and waterproof temperature sensor that operates reliably in temperatures ranging from -55°C to +125°C.

**Working Principle:** It works on the principle of 1-Wire Interface which requires only one data line (and ground) for connection to the microcontroller, making it easy to integrate.

**Functionality:** Attached inside cabinet, it measures temperature of each cabinet.

**Solenoid Lock- (Component 4):** A solenoid is inside every cabinet used to open/close the cabinet door via dashboard.

**WS2812B- (Component 5):** A type of individually addressable RGB LED that can be controlled independently, allowing for precise control over colour and brightness. It has been used to show battery status inside the cabinet (battery fully charged, battery charging, fault indication).

**KY-026- (Component 6):** A small, inexpensive, and easy-to-use flame sensor module that can detect infrared (IR) light emitted by flames. It's particularly sensitive to wavelengths between 760nm and 1100nm, which are commonly emitted by flames.

**Solenoid Lock State Detection- (Component 7):** Used to detect current state of the solenoid lock.

**Relay- (Component 8):** Used to operate a lock by directing power to the lock's actuators after getting control signal from the MCU.

**LED- (Component 9):** Used to indicate which slave is operational on PCB.

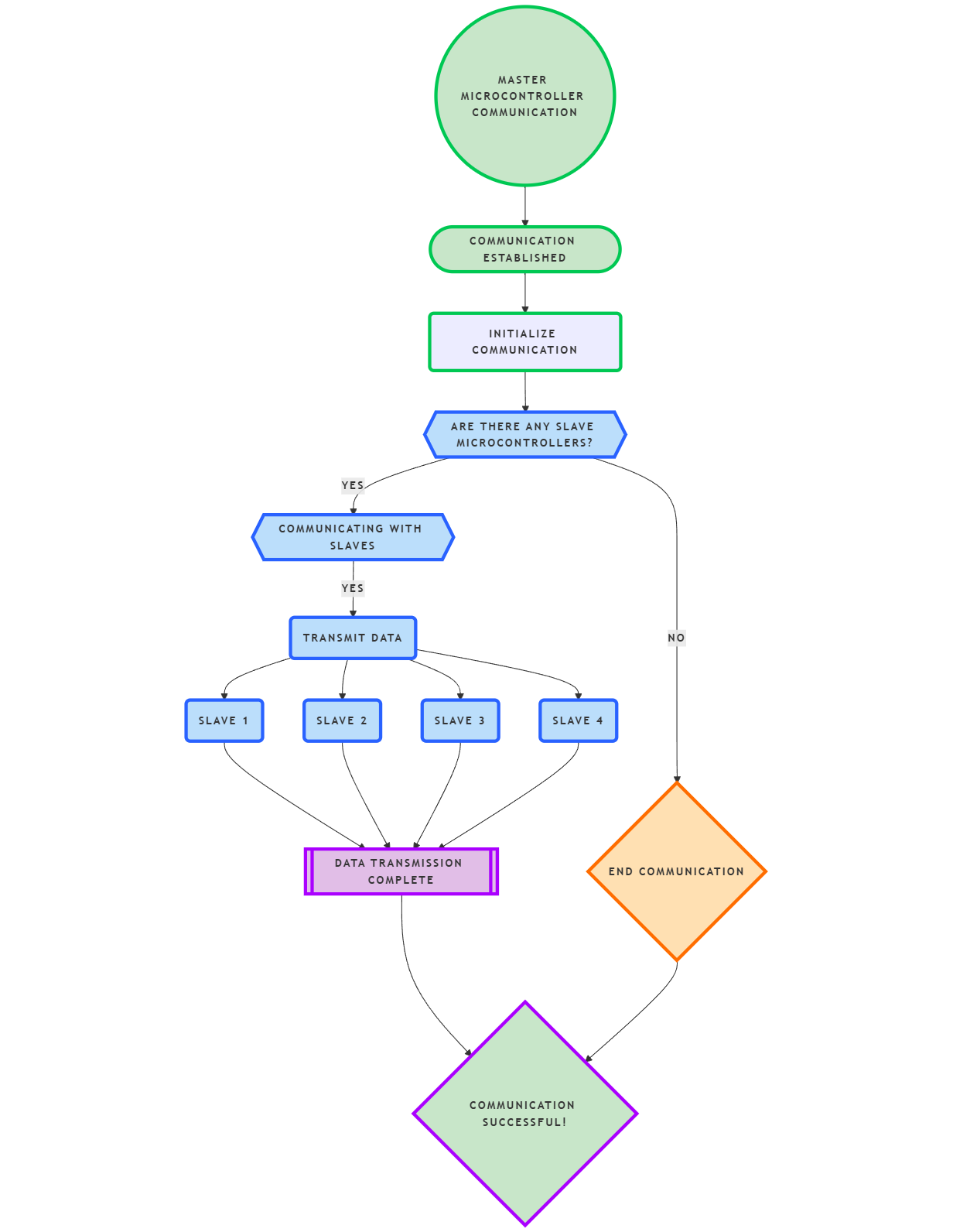


Figure 15: Master and Slave communication flow

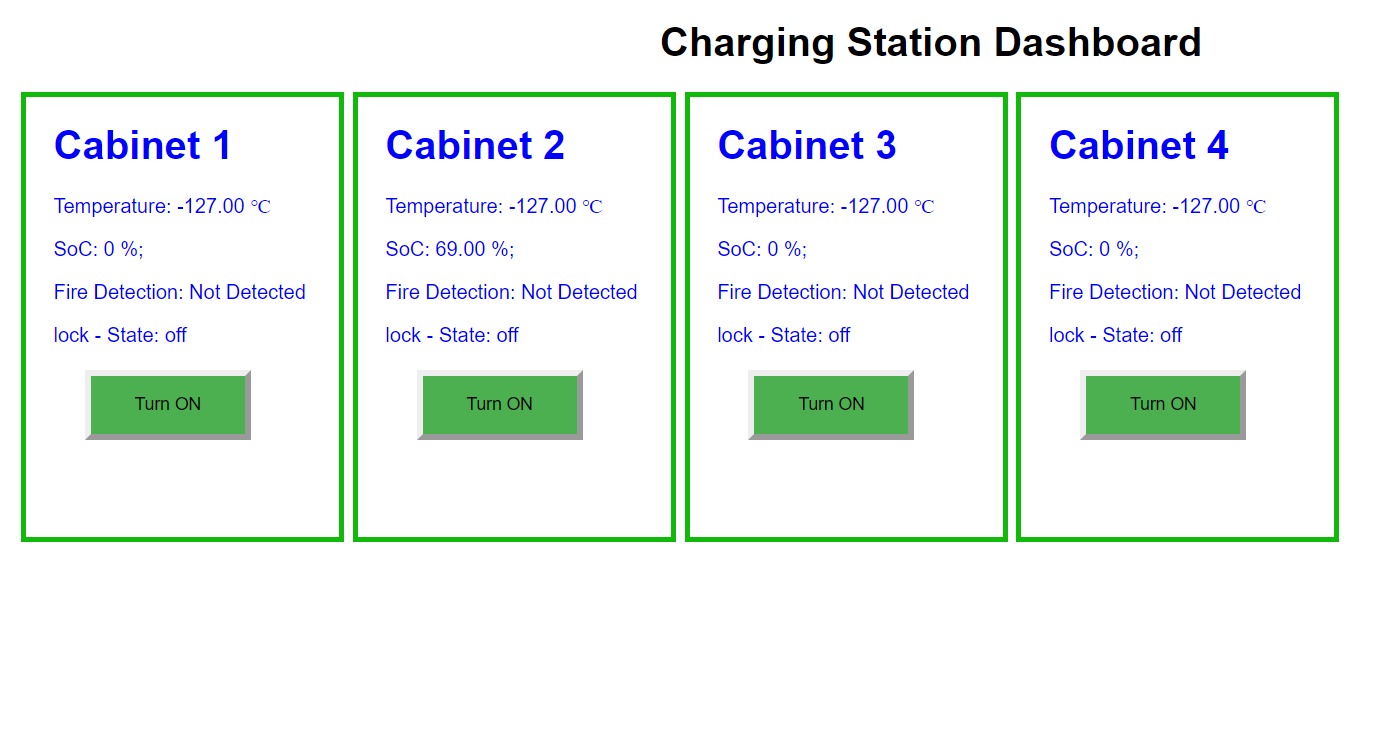
**7. Operation on Local Server**

Figure 16: Esp local Server Dashboard.

The master microcontroller communicates with a local server dashboard hosted on an ESP8266, displaying critical data such as slot temperature, state of charge (SoC), fire detection status, and lock status. This information is received from slave devices via the ESP-NOW protocol, enabling efficient wireless communication and real-time monitoring of the system's key parameters.

Slave micro controllers

ESP 8266 local IP dashboard

**8. Operation On Cloud Based Server**

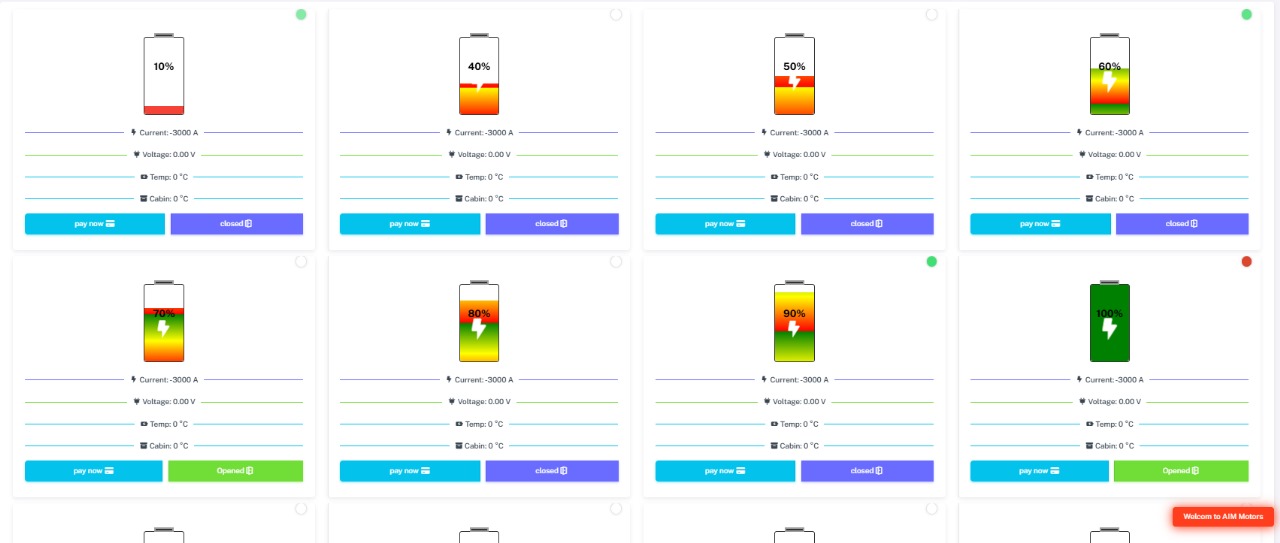


Figure 17: Cloud Based Dashboard

The master microcontroller communicates with a cloud-based server using websockets, transmitting critical data such as slot temperature, state of charge (SoC), lock status, current, voltage, battery temperature, and cabinet temperature. This information is received from slave devices via the ESP-NOW protocol and forwarded to the cloud server. Additionally, the cloud dashboard provides an option to operate the electromagnetic lock remotely, enabling efficient wireless communication, real-time monitoring, and remote control of the system's key parameters from anywhere with internet access.

**9. PCB Comparison with Previous version**

Key difference are listed below;

|  |  |
| --- | --- |
| **VERSION 1 (NED) VERSION 2(AIM)** | |
| **No lock operations** | **Local operation through local server** |
| **No flame sensing** | **Flame sensor integrated** |
| **No temperature sensing** | **Temperature sensor for cabinets integrated** |
| **Master microcontroller using serial communication with PC Dashboard and needs to be placed within 5 meters range of Charging Station** | **Master Micro controller can be operated using same WIFI connection.** |
| **Slaves powering up directly through 5v adopters** | **Slave PCB have 4 options to get power up linking with pushbuttons.** |
| **No RGB led for indications** | **Have RGB led for indication purposes** |
| **No Led power indication** | **Have Led power indication** |
|  |  |

**10 . Troubleshooting the Charging Station PCB System**

This section provides a comprehensive guide to diagnosing and resolving common issues encountered in the Charging Station PCB system, including its components, communication protocols, and power management.

#### ****1. General Troubleshooting Steps****

1. **Visual Inspection**:
   * Inspect for loose wires, soldering issues, or visible damage on the PCB.
   * Check for burnt components, discoloration, or physical cracks on sensors and modules.
2. **Power Supply Verification**:
   * Use a multimeter to check voltage at the power input terminals (should be 12V).
   * Verify the output of the LM2576T voltage regulator for 5V DC supply.
   * Ensure capacitors, inductors, and diodes are properly connected and functional.
3. **Connectivity Testing**:
   * Ensure all connections between the components and microcontroller GPIOs are intact.
   * Verify that CAN bus and ESP-NOW communication pins are correctly wired.
4. **Microcontroller Reset**:
   * Perform a hard reset of the ESP8266 to clear temporary glitches.
   * Reflash the firmware if needed to ensure the correct program is loaded.

#### ****2. Component-Specific Troubleshooting****

**Electromagnetic Lock**

* **Issue**: Lock not operating or providing incorrect status.
* **Possible Causes**:
  + Faulty relay or optocoupler.
  + Incorrect wiring of the lock’s status pin.
* **Solution**:
  + Test the relay and optocoupler using a multimeter.
  + Verify connections between the lock, relay, and microcontroller.
  + Replace defective components.

**Flame Sensor (KY-026)**

* **Issue**: Flame detection not functioning or false alerts.
* **Possible Causes**:
  + Sensor misalignment or dirt on the lens.
  + Incorrect threshold setting in the microcontroller.
* **Solution**:
  + Clean the sensor lens and realign it towards the target area.
  + Update the microcontroller code to adjust detection sensitivity.

**Temperature Sensor (DS18B20)**

* **Issue**: Temperature readings are inaccurate or not received.
* **Possible Causes**:
  + Incorrect wiring of the 1-wire interface.
  + Faulty sensor or improper placement in the cabinet.
* **Solution**:
  + Verify wiring connections (data pin, VCC, and ground).
  + Replace the sensor if it shows incorrect readings after testing.

**RGB LED (WS2812B)**

* **Issue**: LED not changing colors or displaying incorrect status.
* **Possible Causes**:
  + Faulty data pin connection or insufficient power supply.
  + Errors in the microcontroller program controlling the LED.
* **Solution**:
  + Check the LED’s data line and supply voltage.
  + Debug the microcontroller code to ensure proper signal generation.

**CAN Communication (MCP2515)**

* **Issue**: No communication between BMS and microcontroller.
* **Possible Causes**:
  + Incorrect CAN baud rate or faulty termination resistors.
  + Loose connections in the CAN-H and CAN-L lines.
* **Solution**:
  + Reconfigure the CAN baud rate to match the BMS settings.
  + Verify the presence and placement of 120-ohm termination resistors.
  + Tighten loose connections.

**ESP-NOW Communication**

* **Issue**: No data transmission between master and slave.
* **Possible Causes**:
  + Incorrect pairing of ESP8266 devices.
  + Signal interference or incorrect firmware.
* **Solution**:
  + Re-pair the devices using correct MAC addresses.
  + Reduce nearby signal interference by changing the channel.
  + Reflash the firmware if issues persist.

#### ****3. Power-Related Issues****

**No Power to PCB**

* **Possible Causes**:
  + Faulty 12V adapter or connections.
  + Overloaded components causing a shutdown.
* **Solution**:
  + Replace the adapter and inspect the input connections.
  + Check for shorts or overdrawn current in the circuit.

**Overheating Components**

* **Possible Causes**:
  + Excessive current load on the LM2576T voltage regulator.
  + Faulty capacitors or inductor causing thermal stress.
* **Solution**:
  + Reduce the load by disconnecting unnecessary modules.
  + Inspect and replace damaged components.

#### ****LED Indication for Debugging****

|  |  |  |
| --- | --- | --- |
| **LED Color** | **Indication** | **Action** |
| Green | System operational (SOC > 97%) | No action needed. |
| Yellow | SOC below 97% | Check battery charge status. |
| No Light | Power failure or disconnection | Verify power connections and RGB LED wiring. |

#### ****5. Communication Debugging****

* **UART**:
  + Use a serial monitor to verify data transmission.
  + Check baud rate settings and RX/TX connections.
* **CAN**:
  + Utilize a CAN analyzer to inspect data packets on the bus.
  + Ensure all devices on the bus share the same configuration.
* **ESP-NOW**:
  + Use debug logs to confirm data transmission and reception.
  + Test wireless range and reduce interference if signal drops occur.

#### ****6. Advanced Debugging Tools****

* **Multimeter**: For checking voltage levels and continuity.
* **Oscilloscope**: To inspect data signals on communication lines.
* **CAN Analyzer**: For troubleshooting CAN protocol issues.
* **Serial Monitor**: For UART debugging.

By following this troubleshooting guide, most issues in the Charging Station PCB system can be identified and resolved effectively. For unresolved problems, contact the engineering support team.