# STM32-Based Photovoltaic Power Generation Monitoring and Protection System

**Final Report**

**ECE 445 Senior Design**

**Project:** STM32-Based Photovoltaic Power Generation Monitoring and Protection System

**Prepared by:** Guangjun Xu, Xu Li, and Sunhao Zhang

**Date:** May 2026

# Abstract

This report presents the completed design of an STM32-based photovoltaic (PV) power generation monitoring and protection system. The project was developed to improve the visibility, safety, and controllability of a small PV generation setup used in a household or small commercial demonstration environment. The final prototype uses an STM32F103C8T6 microcontroller as the main controller, a voltage-sensing channel for the PV-side electrical input, a DS18B20 digital temperature sensor for local thermal monitoring, an organic light-emitting diode (OLED) display for local data presentation, push buttons for local user control, an ESP8266-01S WiFi module for communication with the Gizwits mobile application, and relay, light-emitting diode (LED), and buzzer outputs for protection and alarm functions.

The completed prototype verifies the core embedded-system functions required by the final project scope: real-time monitoring of voltage and temperature, calculation and display of current and power indicators, threshold-based automatic protection, manual control, local threshold adjustment, remote data observation, and remote command input through the mobile application. The earliest project description used the term charger, but the implemented hardware does not include a battery charging circuit or battery energy-storage subsystem. In the final design, the relay is used as a controllable output and protection device for the PV/load path. Block-level and system-level tests confirmed that the prototype initializes correctly, displays sensor data, responds to local and remote user settings, enters an alarm state when threshold conditions are violated, disconnects the relay output in automatic mode, and communicates with the mobile application through WiFi.

**Keywords:** STM32F103C8T6, photovoltaic monitoring, ESP8266, OLED display, threshold protection, relay control

## Contents

# 1 Introduction

## 1.1 Motivation and Problem Statement

Small PV systems are increasingly used in household and small commercial scenarios because they are clean, compact, and relatively easy to install. However, a basic PV panel and load connection does not automatically provide clear information about operating status. Without a monitoring and control layer, users may not know whether the PV output is within a safe range, whether the local temperature is abnormal, or whether the output path should be disconnected to protect equipment. These limitations motivated the project.

The design goal was to build a low-cost embedded monitoring and protection system for a small PV generation setup. The system collects key operating data, displays the data locally, supports local configuration, provides automatic alarm behavior, and allows remote observation and control through a mobile application. The project targets a demonstrable embedded-control prototype rather than a grid-connected inverter, a commercial protection relay, or a certified battery charger.

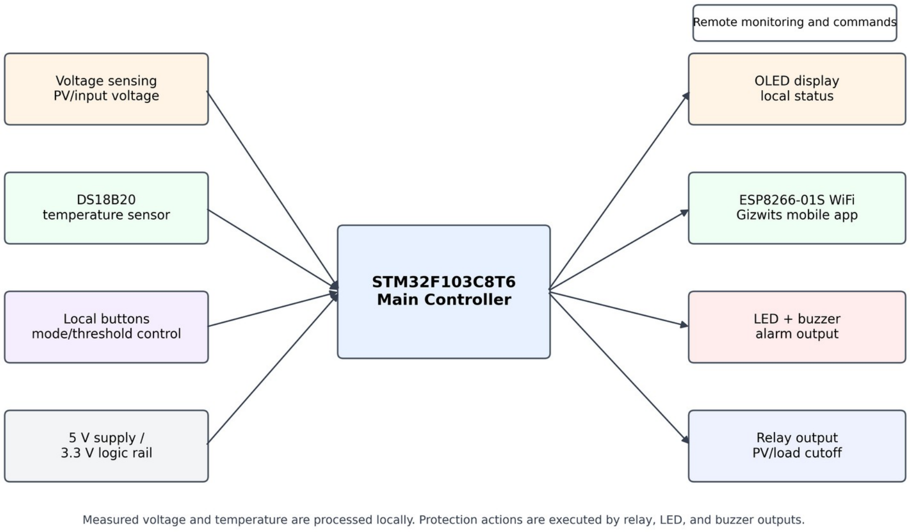
## 1.2 Final Scope and Design Boundary

The completed project is a PV generation monitoring and control system. It does not include a battery charging circuit, a battery management system, or an energy-storage pack. This final boundary is important because some early project wording referred to a PV charger. During implementation, the team narrowed the scope to the embedded monitoring, alarm, relay-protection, and remote-control functions. This choice made the prototype more reliable and easier to verify within the project schedule while preserving the main engineering value of the design: safe and intelligent supervision of a small PV output.

In the final design, the relay output connects or disconnects the controlled PV/load path. In automatic mode, the relay opens when voltage, current, or temperature values exceed preset limits. In manual mode, the relay and alarm outputs can be controlled through local buttons or the mobile application. This design centers the project on sensing, embedded decision logic, human-machine interaction, and wireless control rather than electrochemical charging behavior.

## 1.3 System Overview and Performance Requirements

Figure 1 shows the final system organization. The STM32F103C8T6 receives information from the voltage sensor, DS18B20 temperature sensor, and local buttons. It processes the data, updates the OLED display, communicates with the ESP8266-01S WiFi module, and drives the LED, buzzer, and relay outputs. The project requirements used for final verification are summarized in Table 1.



*Final system block diagram.*

Table 1. Final system performance requirements

| Requirement | Target |
|---|---|
| Power rails | 5 V input and 3.3 V logic rail are available and stable after power-on. |
| Data display | OLED shows voltage, calculated current, calculated power, temperature, operating mode, and threshold values. |
| Temperature sensing | DS18B20 updates temperature data during operation. |
| Voltage sensing | ADC-based voltage measurement changes consistently with the PV/input voltage. |
| Automatic protection | Over-threshold voltage, current, or temperature causes local alarm and relay cutoff. |
| Manual control | Buttons can switch modes, adjust thresholds, and control relay/buzzer functions. |
| Remote communication | Gizwits app can receive system data and send control or threshold commands. |
| Final scope | No battery charging or storage function is required for final verification. |

# 2 Design

## 2.1 Design Procedure and Major Decisions

The system was divided into input, processing, output, and communication blocks. The input blocks include the voltage-sensing channel, DS18B20 temperature sensor, local buttons, and regulated power input. The processing block is the STM32F103C8T6 microcontroller. The output blocks include the

OLED display, LED, buzzer, and relay. The communication block is the ESP8266-01S WiFi module connected to the Gizwits mobile application.

Several design choices were made to keep the system practical for a course prototype. The STM32F103C8T6 was selected instead of a simpler 8051-type controller because it provides general-purpose input/output (GPIO), analog-to-digital converter (ADC) capability, serial communication, and enough processing resources for the sensing, display, and communication tasks [2], [3]. The OLED display was selected instead of a character liquid-crystal display because it can show several measured and threshold values in a compact area. The DS18B20 sensor was selected because it provides digital temperature data and avoids an analog temperature-conditioning circuit [4]. The ESP8266-01S was selected because it provides a low-cost WiFi interface suitable for a mobile-app demonstration [5].

The original idea of including a battery subsystem was removed from the final system. A safe battery charger would require charge-current regulation, battery chemistry selection, overcharge and overdischarge protection, thermal safety design, and a separate verification process. Those requirements were outside the completed prototype. The final system instead demonstrates a monitoring and relay-protection layer that could be placed in front of a future charger or controlled load interface.

## 2.2 Hardware Design

### 2.2.1 Main Controller

The STM32F103C8T6 minimum system board is the center of the prototype. It handles sensor acquisition, display refresh, key scanning, alarm decisions, relay control, and serial communication with the WiFi module. The board provides the GPIO pins required for buttons and outputs, an ADC input for the voltage-sensing channel, and a universal asynchronous receiver-transmitter (UART) interface for the ESP8266-01S.

### 2.2.2 Voltage Measurement

The PV-side electrical input is measured through a voltage-sensing channel connected to the microcontroller ADC. Because the ADC input range is limited by the microcontroller reference voltage, the external voltage is scaled before entering the ADC. The firmware converts the ADC count back to an estimated PV/input voltage using the selected divider or sensor gain. This measured voltage is also used to estimate output current and power in the demonstration model.

The design should be calibrated before precise numerical reporting. In the completed demonstration, the important verification goal is monotonic and consistent response of the displayed voltage to changes in the input condition. A future revision should add a dedicated current-sensing stage if current and power must be reported as calibrated electrical measurements.

### 2.2.3 Temperature Measurement

The temperature-monitoring block uses a DS18B20 digital temperature sensor. The sensor communicates with the controller through a one-wire digital interface [4]. This choice simplifies the hardware design because no analog signal-conditioning circuit is required for temperature. The measured temperature is displayed on the OLED and compared with the temperature threshold in automatic mode.

### 2.2.4 Local Display and User Input

The OLED display is the main local user interface. It shows the system mode, measured voltage, calculated current and power, temperature, and threshold values. Local push buttons provide direct interaction for mode switching, threshold adjustment, confirmation of parameter settings, and manual control actions. This local interface allows the system to remain usable even when the WiFi connection is unavailable.

### 2.2.5 Alarm and Relay Output

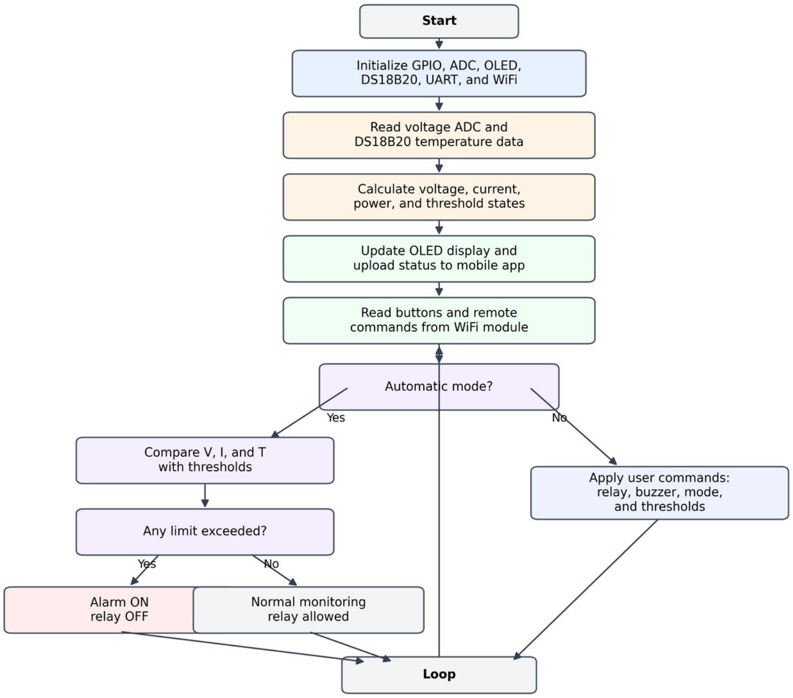
The alarm block consists of an LED indicator and an active buzzer. When a measured value exceeds a preset threshold in automatic mode, the controller turns on the visual and audio alarm. At the same time, the relay opens to disconnect the controlled PV/load path. In manual mode, the relay and buzzer can be controlled by the user. This behavior provides automatic safety response while retaining direct human override for demonstration and debugging.

### 2.2.6 WiFi Communication

The ESP8266-01S WiFi module provides the remote communication channel. The STM32 sends measured data and state information to the module through UART. The mobile application receives uploaded data and can send commands back to the system, including mode switching, relay control, alarm control, and threshold adjustment. This design demonstrates an Internet of Things (IoT) layer for a small PV monitoring system.

## 2.3 Software Design

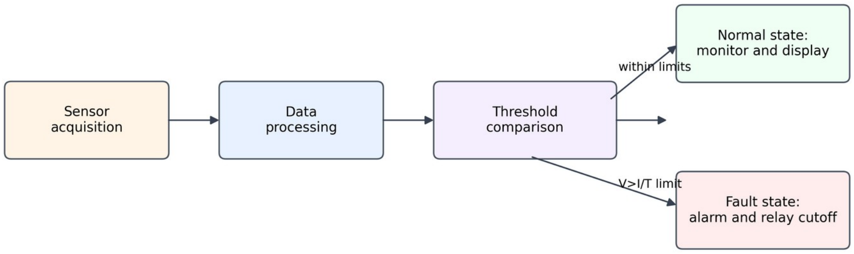
Figure 2 shows the main firmware flow. After power-on, the program initializes GPIO, ADC, OLED display, DS18B20 interface, UART, and WiFi communication. The main loop then reads sensor values, calculates derived values, updates the display, checks local buttons and remote commands, and executes either automatic-protection logic or manual-control logic.



*Main program flow for the completed prototype.*

In automatic mode, the firmware compares measured or calculated values with user-defined thresholds. If the voltage, current, or temperature exceeds its limit, the firmware activates the LED and buzzer and opens the relay. If all values are within range, the system continues monitoring. In manual mode, control priority is given to the user. Button and app commands can directly control the relay and alarm outputs, which is useful for testing and controlled demonstration.

Figure 3 summarizes the protection logic. The protection decision is executed locally on the STM32 so that relay cutoff does not depend on network availability.



The protection decision is executed locally so that relay cutoff does not depend on the wireless link.

*Threshold-based protection logic.*

## 2.4 Design Equations and Data Processing

The ADC conversion uses a proportional relationship between ADC code and the controller reference voltage. Equation (1) gives the ADC input voltage, where $N_{ADC}$ is the sampled ADC count, $V_{REF}$ is the ADC reference voltage, and $N_{MAX}$ is the maximum ADC count. For a 12-bit ADC, $N_{MAX}=4095$.

$$V_{ADC}=\frac{N_{ADC}}{N_{MAX}}V_{REF}$$

If the voltage sensor uses a divider or sensor gain represented by $K_V$, the external PV-side voltage is calculated by Equation (2).

$$V_{PV}=K_V V_{ADC}$$

For the demonstration model, current and power are calculated from the estimated voltage and an equivalent load relationship. If $R_{EQ}$ is the equivalent load resistance, Equations (3) and (4) estimate the current and electrical power.

$$I_{OUT}=\frac{V_{PV}}{R_{EQ}}$$

$$P_{OUT}=V_{PV}I_{OUT}$$

The threshold protection condition is summarized by Equation (5), where $V_{TH}$, $I_{TH}$, and $T_{TH}$ are the voltage, current, and temperature thresholds.

$$\text{Fault}=1\ \text{if}\left(V_{PV}>V_{TH}\right)\vee\left(I_{OUT}>I_{TH}\right)\vee\left(T>T_{TH}\right)$$

If a panel conversion-rate indicator is displayed, it should be interpreted as a relative indicator based on a selected reference power rather than a certified PV efficiency measurement unless the panel area, irradiance, and calibrated output power are measured. Equation (6) gives this relative display indicator.

$$\eta_{display} = \frac{P_{OUT}}{P_{REF}} \times 100\,\%$$

These equations are sufficient for the completed monitoring prototype. A future battery-charging version would require additional equations and hardware for charge current, battery voltage, charge-stage control, and battery safety protection.

# 3 Verification

## 3.1 Verification Method

The project was verified in two stages. First, each block was tested independently to reduce uncertainty during integration. The power rails were checked before connecting sensitive modules. The OLED, button, sensor, alarm, relay, and WiFi functions were then tested one by one. Second, the full system was tested in automatic and manual modes to confirm that the integrated behavior matched the final requirements.

The verification focused on observable engineering behavior: stable power, correct initialization, changing sensor values, display refresh, response to threshold changes, alarm output, relay cutoff, and mobile-app communication. Since the final product does not include a battery, no battery charge-current, charge-voltage, or storage-capacity test was performed.

## 3.2 Block-Level Verification

Table 2 summarizes the block-level verification. The team recorded pass/fail results for the main functions. Exact multimeter readings and mobile-app screenshots should be retained in the lab notebook or added to an appendix if they are required by the instructor.

Table 2. Block-level verification results

| Block | Test performed | Expected result | Observed result |
|---|---|---|---|
| Power supply | Measured rails before full operation. | 5 V input and 3.3 V logic rail present. | Passed; system powered on and modules initialized. |
| OLED display | Displayed system data after initialization. | Readable values and mode information shown. | Passed; voltage, temperature, mode, and thresholds displayed. |
| Temperature sensor | Read DS18B20 value repeatedly. | Temperature updates without bus failure. | Passed; temperature value updated during operation. |
| Voltage sensor | Changed input level and observed ADC display. | Displayed voltage follows input change. | Passed; voltage display changed consistently. |
| Buttons | Pressed each key in setting/control modes. | Mode and threshold values respond | Passed; local input changed states and |

| Block | Test performed | Expected result | Observed result |
|---|---|---|---|
| | | correctly. | values. |
| Alarm outputs | Forced over-threshold condition. | LED and buzzer activate. | Passed; alarm activated in abnormal condition. |
| Relay | Tested automatic cutoff and manual relay command. | Relay changes state as commanded. | Passed; relay opened during protection state. |
| WiFi/app | Connected ESP8266 to app and sent commands. | Data upload and command downlink work. | Passed; app interaction verified. |

## 3.3 System-Level Verification

The integrated automatic-mode test began with normal sensor readings. The OLED displayed the monitored values and the system remained in the normal state. The threshold was then adjusted so that the current measured or calculated value exceeded the corresponding limit. Under this condition, the controller entered the alarm state, activated the LED and buzzer, and opened the relay. This verified the complete safety path from sensing to decision and output action.

The manual-mode test confirmed that the user could switch out of automatic behavior and directly control the relay and buzzer by local buttons or the mobile application. This mode is useful when the user intentionally wants to connect or disconnect the output path, test the alarm device, or recover after a known abnormal condition. The remote-control test confirmed that the WiFi path could send data to the phone and receive commands from it.

The final-scope change was also verified: the system was tested as a PV monitoring and control prototype rather than as a battery charger. The relay output was treated as the protected PV/load path, not as a battery charge-control output. Therefore, the completed verification matches the actual final hardware.

## 3.4 Requirement and Verification Summary

Table 3 summarizes the final requirement and verification status. A more detailed version is provided in Appendix A.

Table 3. Requirement and verification summary

| Requirement | Verification status | Comment |
|---|---|---|
| Local monitoring | Verified | OLED displays measured and calculated PV information. |
| Automatic protection | Verified | Alarm and relay respond to threshold violations. |
| Manual operation | Verified | Buttons and app support direct user control. |
| Remote monitoring/control | Verified | ESP8266 and Gizwits app demonstrate IoT interaction. |
| Battery charging/storage | Not applicable | Removed from final project scope; not part of final prototype. |

# 4 Costs

The cost estimate includes prototype parts and labor. Component prices in Table 4 are small-quantity retail estimates and should be replaced with exact receipt values if purchase records are available. The design is low-cost because it uses common embedded modules and a minimum STM32 system board rather than custom high-power charging hardware.

Table 4. Estimated bill of materials

| Part | Quantity | Estimated unit cost (USD) | Estimated total (USD) |
|---|---|---|---|
| STM32F103C8T6 minimum system board | 1 | 4.00 | 4.00 |
| DS18B20 temperature sensor module | 1 | 2.00 | 2.00 |
| Voltage sensor module or divider circuit | 1 | 1.50 | 1.50 |
| 0.96 in OLED display module | 1 | 4.50 | 4.50 |
| ESP8266-01S WiFi module | 1 | 3.50 | 3.50 |
| ESP8266 adapter or 3.3 V support board | 1 | 2.00 | 2.00 |
| 5 V relay module | 1 | 2.00 | 2.00 |
| Active buzzer | 1 | 0.75 | 0.75 |
| LEDs, resistors, and buttons | 1 set | 2.50 | 2.50 |
| Prototype board, wires, and connectors | 1 set | 6.00 | 6.00 |
| Small PV panel or adjustable DC source for test | 1 | 15.00 | 15.00 |
| Miscellaneous mounting and replacement parts | 1 set | 5.00 | 5.00 |
| **Total electronics** | | | **48.75** |

Labor cost was estimated using the ECE 445 formula in Equation (7) [1]. The ideal hourly rate was assumed to be $40.00 per hour for undergraduate engineering design work. The hours are estimates based on design, assembly, firmware development, integration, testing, and report preparation.

$$\text{Labor cost} = \text{hourly rate} \times \text{hours spent} \times 2.5$$

Table 5. Estimated labor cost

| Team member | Main responsibility | Estimated hours | Rate (USD/h) | Multiplier | Labor cost (USD) |
|---|---|---|---|---|---|
| Guangjun Xu | System integration, sensing, verification, documentation | 42 | 40.00 | 2.5 | 4,200.00 |
| Xu Li | Hardware wiring, display/alarm/relay debugging, testing | 40 | 40.00 | 2.5 | 4,000.00 |
| Sunhao Zhang | Firmware logic, WiFi/app communication, threshold control | 38 | 40.00 | 2.5 | 3,800.00 |
| **Total labor** | | **120** | | | **12,000.00** |

Table 6. Total estimated project cost

| Category | Estimated cost (USD) |
|---|---|
| Electronics and prototype materials | 48.75 |
| Labor | 12,000.00 |
| Machine shop or enclosure fabrication | 0.00 |
| **Total estimated project cost** | **12,048.75** |

The direct hardware cost is only $48.75 for the prototype because the system uses commodity modules. For a small production batch, the electronics cost could be reduced by replacing development modules with an integrated printed circuit board, but certification, enclosure, surge protection, and reliability testing would increase the true product cost.

# 5 Conclusions

## 5.1 Accomplishments

The completed prototype demonstrates a working STM32-based PV monitoring and protection system. It measures PV-side voltage data and local temperature, calculates operating indicators, displays information locally, supports threshold setting, provides automatic protection, allows manual control, and communicates with a mobile app through the ESP8266-01S WiFi module. The system meets the final project scope and provides a practical embedded-system layer for improving safety and observability in a small PV setup.

The main accomplishment is the integration of sensing, decision logic, user interaction, and remote communication into one functioning prototype. The final design is also clear about its boundary: it is not a battery charger. This distinction prevents the report from claiming an unverified storage or charging function and keeps the final verification consistent with the actual hardware.

## 5.2 Limitations and Future Work

The most important limitation is that the final system does not perform closed-loop maximum power point tracking, DC-DC power conversion, or battery charging. The current and power values are calculated for monitoring and demonstration rather than measured through a precision current-sensing stage. A future version could add a dedicated current sensor, a DC-DC converter, maximum power point tracking, and a safe battery-charging controller. If battery storage is added, the design must include battery chemistry selection, charge-stage control, overcharge protection, overdischarge protection, temperature protection, and independent safety verification.

The WiFi function is suitable for demonstration, but a field-deployed PV controller would also need stronger communication fault handling, enclosure design, surge protection, electromagnetic compatibility testing, and long-term reliability testing. Future work could also improve the mobile interface by adding historical data plots and clearer warning messages.

## 5.3 Ethics and Broader Impact

The design supports safer and more convenient use of small PV systems by giving users visibility into operating conditions and by disconnecting the output path when values exceed limits. This aligns with the ethical responsibility to design systems that reduce risk to users and equipment [7]. At the same time, the system should not be presented as a certified commercial protection device or a battery charger without additional safety testing. Clear documentation of final scope is therefore an ethical requirement.

The broader impact of the project is positive because it supports renewable-energy monitoring at small scale. Low-cost PV supervision can help users understand local energy production and react to abnormal operation. However, any real deployment must consider electrical safety, calibration accuracy, environmental protection, and responsible handling of networked-device data.

# References

[1] ECE 445 Staff and ECE Editorial Services, *Preparing Your Final Report for ECE 445, Senior Design*, University of Illinois Urbana-Champaign, Sept. 2019.

[2] STMicroelectronics, *STM32F103x8 STM32F103xB Datasheet: Medium-density performance line ARM-based 32-bit MCU*.

[3] STMicroelectronics, *RM0008 Reference Manual: STM32F101xx, STM32F102xx, STM32F103xx, STM32F105xx and STM32F107xx Advanced ARM-based 32-bit MCUs*.

[4] Analog Devices/Maxim Integrated, *DS18B20 Programmable Resolution 1-Wire Digital Thermometer Datasheet*.

[5] Espressif Systems, *ESP8266EX Datasheet and ESP8266 AT Instruction Set*.

[6] Solomon Systech, *SSD1306 OLED/PLED Segment/Common Driver with Controller Datasheet*.

[7] IEEE, *IEEE Code of Ethics*.

[8] J. Tan, “Design and application effect analysis of an embedded photovoltaic automatic tracking control system,” *Technology Innovation and Application*, no. 23, pp. 42-45, 2023.

[9] Z. Zhang, Z. Zhang, J. Wang, et al., “Exploration of an intelligent monitoring and management system for photovoltaic power stations,” *Automation Panorama*, no. 8, pp. 76-81, 2024.

[10] H. Zhang and W. Zhu, “Research and design of a dual-axis solar tracking system based on STM32,” *Instrumentation Technology*, no. 2, pp. 23-26, 2019.

## Appendix A. Detailed Requirement and Verification Table

Table A.1. Detailed requirement and verification table

| No. | Requirement | Verification procedure | Passing criterion | Result |
|---|---|---|---|---|
| 1 | Power supply stability | Measure 5 V input and 3.3 V rail before connecting all modules. | System powers on without reset loop; OLED and MCU initialize. | Pass |
| 2 | Voltage monitoring | Apply or change PV/DC input and observe displayed value. | Displayed voltage changes consistently with input. | Pass |
| 3 | Temperature monitoring | Read DS18B20 repeatedly and warm the sensor slightly by hand. | Displayed temperature changes reasonably and no bus failure occurs. | Pass |
| 4 | OLED interface | Power on and navigate modes. | OLED shows measured values, mode, and thresholds clearly. | Pass |
| 5 | Button interface | Press each button in normal and setting modes. | Mode switch, value adjustment, and confirmation work. | Pass |
| 6 | Automatic alarm | Set threshold below measured value or simulate abnormal input. | LED and buzzer turn on and relay opens. | Pass |
| 7 | Manual control | Switch to | Relay and buzzer | Pass |

| No. | Requirement | Verification procedure | Passing criterion | Result |
|---|---|---|---|---|
| | | manual mode and command relay/buzzer locally. | follow user command. | |
| 8 | Remote app function | Connect ESP8266 to Gizwits app and send command. | App receives data and system receives command. | Pass |
| 9 | Battery function | Check final hardware scope. | No battery charging/storage hardware is included or claimed. | Not applicable |

## Appendix B. Team Contribution Summary

Table B.1. Contribution summary

| Team member | Contribution summary |
|---|---|
| Guangjun Xu | Coordinated system integration, clarified final system scope, supported sensor verification, and revised final documentation. |
| Xu Li | Built and debugged hardware connections, tested the OLED, relay, alarm, and local button functions, and supported system-level verification. |
| Sunhao Zhang | Implemented major firmware logic, supported WiFi/app communication tests, and integrated threshold adjustment and control logic. |

## Appendix C. Notes for Final Submission

The report intentionally avoids claiming unverified battery-charging or energy-storage functions. If the team later adds a real charger stage, the verification section should be updated with battery voltage, charge current, charge termination, overcharge protection, thermal protection, and long-duration safety tests. If exact purchase receipts, oscilloscope screenshots, mobile-app screenshots, or multimeter readings are available, they should be added to Appendix A or to a new measurement appendix before submission.