

PylonTech Battery Emulator - Function Documentation with RS485

Overview

This document provides a detailed description of the functions in the PylonTech Battery Emulator project.

It covers CAN bus communication with Victron and PylonTech, charge current adjustments, and the newly integrated RS485 communication.

1. **sendDeviceIdentification()** - CAN ID: 0x305

Identifies the BMS to Victron as 'PYLONTEC'. This ensures compatibility with Victron inverters and proper system integration.

The function sends a fixed identification message.

2. **send_module_status()** - CAN ID: 0x381 + Module ID

Transmits the highest and lowest cell voltage per module along with the charging permission status.

Victron uses this data to manage charge balancing between modules.

3. **send_corrected_charge_current()** - CAN ID: 0x386

Calculates and sends the corrected charge current based on module requests.

If a module requests a charge stop, its current is deducted from the total charge current.

4. **RS485 Communication Integration**

The RS485 communication is now used to interface with the Current Monitor.

Only the active master (determined via CAN bus) sends RS485 data, while all slaves remain in listening mode.

5. **setRS485Mode()**

Switches between send and receive modes for RS485.

Ensures that only the master sends data while others remain in listening mode.

6. **sendDataToCurrentMonitor()**

Sends Modbus RTU commands via RS485 to the Current Monitor.

Only the active master is allowed to send, preventing communication conflicts.

7. read_response()

All devices listen for responses from the Current Monitor over RS485.

Ensures that both master and slaves receive the necessary data.

8. loop() - Main Execution Cycle

Runs periodically to update system status and send necessary data over CAN and RS485.

Includes logic to check for master status before sending RS485 messages.

Conclusion

This document outlines the key functions of the PylonTech Battery Emulator, including both CAN bus and RS485 communication.

The improvements ensure smooth charge control and efficient data handling between all system components.