

TriStar MPPT System Control

Programmatic Control of Networked TriStar MPPT Controllers

v01



Abstract:

System integrators and project engineers often design systems that require central control over multiple controllers. The TriStar MPPT controller is well suited for this task, providing a simple but powerful software solution. This document covers the information needed to implement system control for a network of TriStar MPPT charge controllers.

Target Audience:

System integrators, designers, and engineers

System Design

- Each controller must have its own solar array.
- All controllers in the system are connected to a common battery bank.
- All controllers are connected to the System Controller through a MODBUS network.

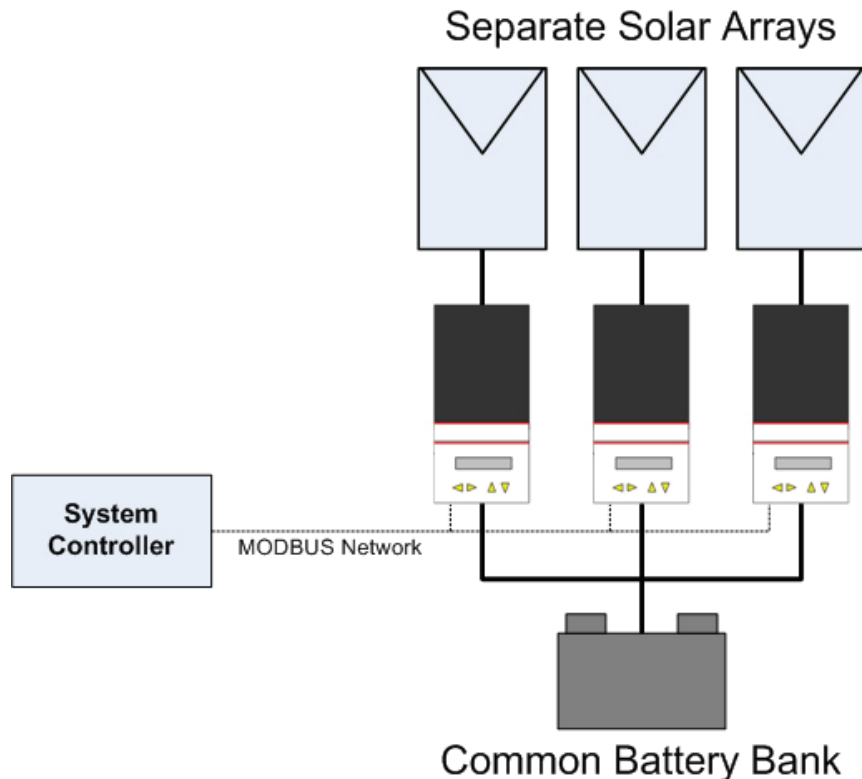


Figure 1. System design overview

The System Controller may be a PC, Linux box, PLC, or other computing device.

The physical MODBUS network may be RS-232, EIA-485 (RS-485), Ethernet, or USB (or a combination of these).

System Control

Note: Download the *TriStar MPPT Modbus Document* from our website to use as a reference for this section.

MODBUS PDU register 0x0059 (Logic Address 90) is the Slave Control register. If a non-zero value is written to this address, the controller will automatically enter SLAVE mode and adjust the battery target voltage to the value written to the register. Note that the scaling value for the Slave Control register is identical to the Battery Voltage value. Once in SLAVE mode, the controller expects the slave register to be updated at least once per minute. If a value is not written to the register within 1 minute, the controller will fault and charging will cease until a new value is written.

The value written to the Slave Control register must be:

- scaled correctly
- the exact regulation value (already temperature compensated if needed)
- updated at least once per minute

In SLAVE mode the controller surrenders the battery charging logic, including battery temperature compensation, to

the system controller. However, all other functions including:

- Maximum power point tracking
- Real time diagnostics reporting (faults & alarms)
- Hardware and software protections
- Day / Night detection and transition
- data logging

are still operational and automatic, requiring no action from the system controller.

Example

Set the target voltage setpoint of all controllers in the system to 14.34 Volts.

Step 1: Read the voltage scaling values

V_PU hi register 0x0000 = 0x00B4 (180 decimal)

V_PU lo register 0x0001 = 0x0000 (0 decimal)

Voltage scaling value = V_PU hi + (V_PU lo / 2¹⁶) = 180.0

Note: Step 1 only needs to be done once. The scaling values stored in the controller may change in future firmware versions, but they are static values that only need to be read once.

Step 2: Compute the scaled value that must be written to the Slave Control register

Scaled Value = (Target Voltage x 32768) / (Voltage Scaling Value)

Scaled Value = (14.34 V x 32768) / 180.0

Scaled Value = 2610.51

Scaled Value = 0x0A32

Step 3. Send the MODBUS write command

Write the scaled value to the Slave Control register of each SLAVE controller on the network. Per the MODBUS specification, a confirmation or error response will be returned for each write command sent.

All controllers will now have a target regulation voltage of 14.34 Volts.