Report 4/12/2019:

William Kerr

# The Good:

I found out how to harvest data. I found a free, open-source library called pymodbus. Using this library, I can collect this type of data:

Battery Voltage (V)

Array Voltage (V)

Battery Current (A)

Array Current (A)

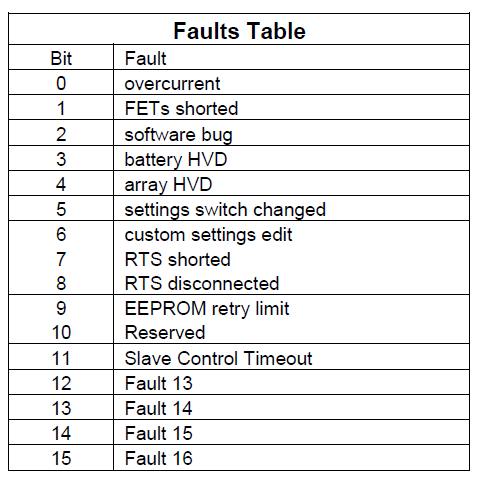
Heatsink Temperature (degrees C)

RTS temperature (degrees C)

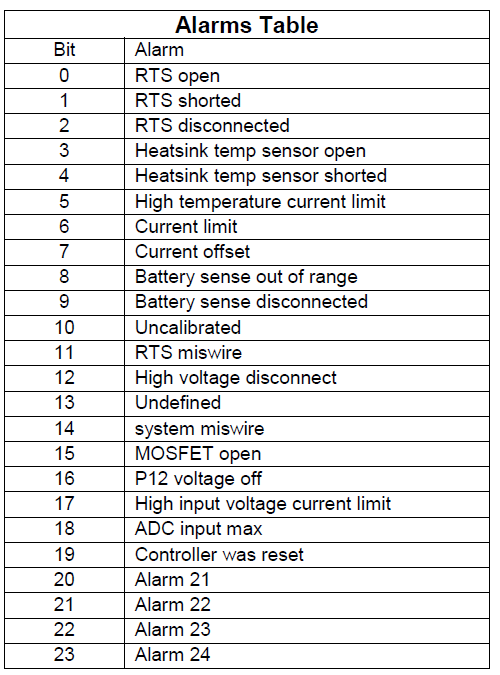
Slow Voltage and Current (for use in threshold alarms)

Hours in operation

Faults (contained in a bitfield. 16-bit value.)

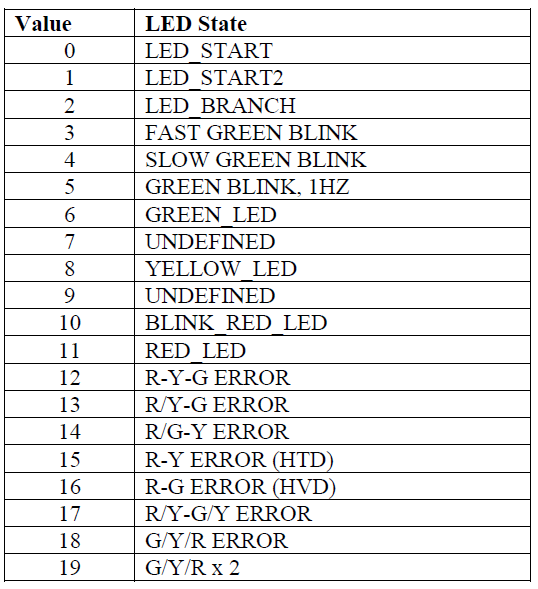


Alarms: Contained in 2 16-bit bitfields. Here’s the table for this.

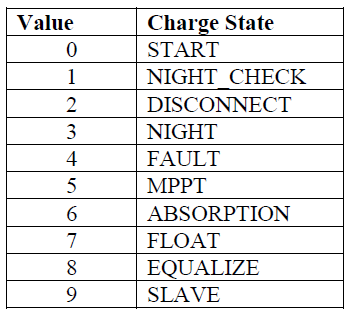


DIP switch settings: also contained in a bitfield. Bit 0 is switch 1, bit 1 is switch 2, and so on.

LED state: contained in a value. Value table:



Charger State: contained in a value. Value table:



Logging values:

Can log daily maximum and minimum voltage, current, and temperature values, total amp and watt hours, daily flags, faults, and alarms raised (but only for that day), daily power output, how much time was spent absorbing power for each day, reference voltages,

The Tristar MPPT contains an EEPROM to hold custom charge settings. They can only be set using a PC using MSView. Download it from Morningstar’s website.

The EEPROM contains things like this:

TCP network settings: HTTP port number, MODBUS port number, BIT0, etc.

The Charge Settings: (absorption voltage and time, float voltage, voltage that cancels float, float timer, days between equalization cycles, equalize time limit above Vreg, battery high voltage disconnect, and so on.

LED Settings: What voltages will the Tristar MPPT’s LED respond to in the battery?

MODBUS and Meterbus Slave addresses.

Serial Number, Model number, hardware version.

If you try to write to the EEPROM using a microcontroller, there will be an EEPROM CHANGED fault. The control must be reset to clear this fault.

The voltages are sent in Tristar’s proprietary format. You must scale it by dividing it by the scaling factor listed in the MODBUS document. Some of the values are signed, some of them are not. That is also listed in the document.

There is a feature on the Tristar MPPT that allows you to write single coil at address 0x0003 to force the controller into a disconnect state, in case anything goes wrong. I will put in the manual: “If there is an emergency, SSH into the raspberry pi, run python, connect to the charge controller, and execute client.write\_coil(0x0003, true). This will internally disconnect the charge controller from the batteries.”

There’s other things you can do with these coils, including triggering an equalize, clear Ah resettable ad total, clear kWh resettable and total, reset battery service calendar, clear faults, clear alarms, force EEPROM update, clear battery max and min voltages (Vb\_min and Vb\_max), test a single phase, send test notifications 1-4, and reset communications server.

# The Bad:

I tried using pymodbus3, but it doesn’t work at all. The Tristar MPPT kept sending bad messages to this library for some odd reason. I ditched this library. MinimalModbus doesn’t work either.

I found a supposedly faster library on the internet used by professionals to speed up their PLC programming. The problem is that it’s very expensive. It assumes that the person using their products is a working professional, which I am not; I’m a student. It costs $200. I’m not sure if I can justify spending $200 on a C++/Java/.NET MODBUS library. I can use the 30-day trial period, but I cannot implement a free trial for a long-term project like this.

You MUST program the Tristar MPPT on a PC using MSView. There is no other way to do this without breaking it.

# The Ugly:

When I poll the Tristar MPPT for data, it only works 15% of the time. The fix for this bug: make the program poll 1000 times before it quits trying to read data.

I don’t know what’s causing this bug. Possible causes:

My laptop is slow. It’s 5 years old now, and sometimes it doesn’t receive things. Unlikely though.

The library I’m using is slow. Python is inherently slower than C or C++. I just wanted to see if it is possible to poll the Tristar MPPT for data using Python.

The Tristar MPPT I’m using to test my program on is broken/unreliable. This could be the case. Sometimes it works, sometimes it doesn’t. Sometimes it takes 900 tries, but it works.