

CHARGE CONTROLLER COMMUNICATION USING MATLAB

Jose Chavez-Garcia

Abstract: The MorningStar MPPT-45 charge controller was analyzed to communicate with a computer using a MatLab-Modbus interface. Although a program has been made to send and receive information, it has not been able to decode real values other than temperatures. This device will be connected to a different (Linux) computer to analyze and monitor the status of a solar panel array in the field.

I. Introduction

The purpose of communication between the MatLab and the MS MPPT-45 charge controller (MSCC) is to produce comprehensible values that accurately monitor the status of the array (i.e. temperature, voltage, day/night, etc.) Given that a diesel generator is currently powering the radio antennae array, the purpose of this is to replace it with reliable and sustainable power. Overall, the complications with communication have been with generating a working scaling factor and identifying the output bytes that will compile a real value.

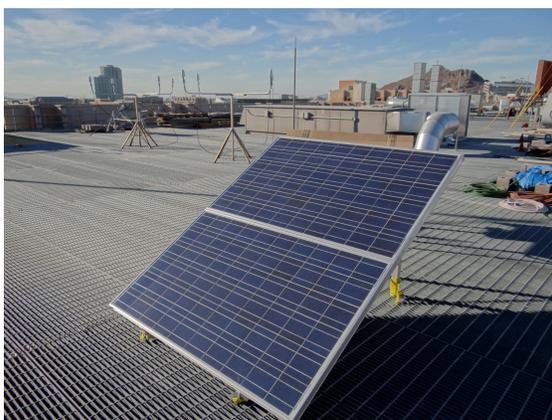


Figure 1 - Solar Panels on the ENGRC roof

II. Procedure

Using the Tri-Star MSCC Modbus Specification Manual [1] provided in MS website, a MatLab code was created. The strategy was to create a command that would take an array using serial communication, via the ModBus protocol, and send it to the charge controller. The program is loosely based on other scripts found on the MathWorks website that also used a MatLab-Modbus interface [2]. The string consisted of four characters including the device ID, function code, logical address / command data, and check sum (error check). This is displayed on lines 47 through 58 on the morningstar.m function file.



Figure 2 – MS MPPT-34 Charge Controller

In return, an array of 7 numbers was returned. This included, the device ID, function code, check sum, and the output that is believed to be the fifth number of the array. The reason for this is because it had accurately measured the temperature. In other variables, usually anything other than temperature, the output was unrealistic. For example, values ranging from 150-160 were received when prompting the MSCC for battery voltage

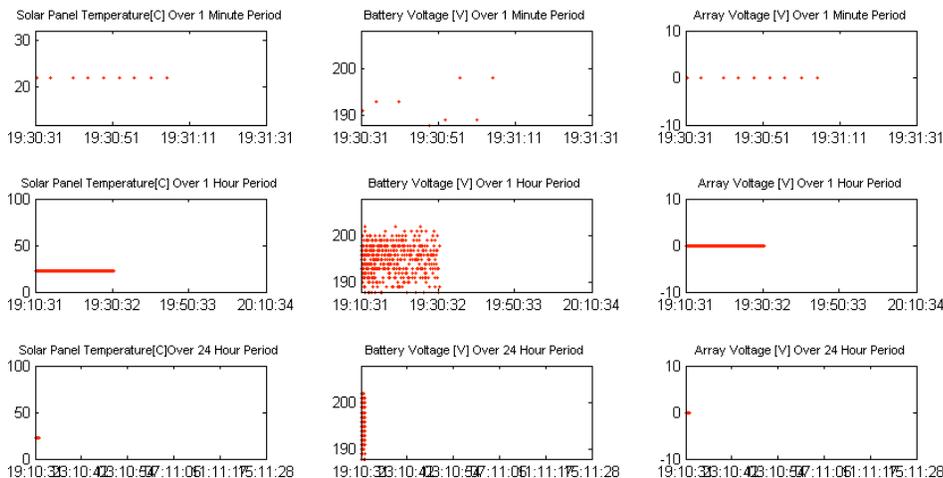


Figure 3

when in reality it was 12V. Similarly, the current was displayed in double digits, which is absurd. The reason for this is because the later two variables needed scaling factors that vary with every command.

Figure 3 shows the plots of three variables that represent the solar panel temperature, battery voltage, and array voltage. The logical addresses are 37, 25, and 36, respectively.

III. Results

Additionally, efforts to decode the values using pg. 25 of the manual have been made. As depicted in the attachment, commands were sent using 1, 2, and 25 as logical addresses representing the whole term, fractional term, and battery voltage. The first two variables will make up the voltage scaling factor (V_PU) of $n * V_{PU} * 2^{-15}$. The returned values were 0, 80, and 130 making the $V_battery$ to be $4.84 * 10^{(-6)}V$. The problem here is that the first variable, scaling whole term, is 0.

MS has been contacted and contributed to this effort. Their technical support team

sent the lab a document explaining the command structure, response message structure, and finding the final output using the scaling factor. Though, they forgot to send the method of finding the scaling factor and have yet to reply.

With MS's response (attachment titled "Modbus message explanation"), the strategy was altered. Instead of just focusing on the 5th byte, the 4th byte was taken into considering as well when calculating n . Rather than n being the output's 5th term, it was the hexadecimal value of the concatenated 4th and 5th terms. The attachment titled "scaling.txt" displays efforts to replicate MS's results. The result, with the lab's own scaling term, was 54.15 instead of the 10 V supplied by the power supply.

III. Conclusion

The MSCC is a good device but proved challenging to communicate with. MorningStar provided some useful information but not enough to decode the final value. The best way to solve this problem is accurately calculate the scaling factor.

References:

- [1] Tri-Star MPPT Modbus Specification
Manuel (V 10.1) *Attached*
- [2]<http://www.mathworks.com/matlabcentral/fileexchange/24696-eurotherm-modbus-rs232-control>