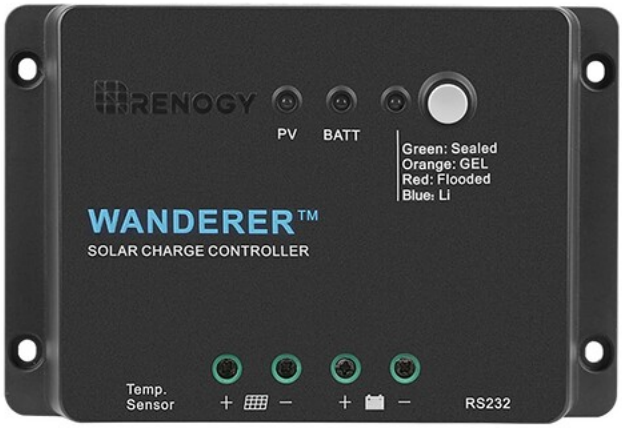
Trying to read registers from a solar charge controller over RS232 (solved!)

wrybread 1 October 9, 2022, 3:03am

I have the Renogy Wanderer 30amp solar charge controller, which has an RS232 port.



I'm trying to read the battery voltage and other data via it's RS232 port from an ESP32 or an Arduino. I haven't found any projects that do that, but I did find one that uses a Pi and NodeJS:

GitHub - mickwheelz/NodeRenogy: Utility to retrieve data from Renogy solar...

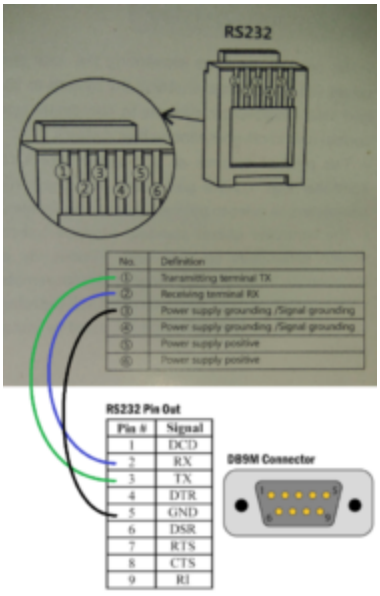
Utility to retrieve data from Renogy solar controllers and publish it to MQTT, written in NodeJS - GitHub - mickwheelz/NodeRenogy: Utility to retrieve data from Renogy solar controllers and publish...

It describes how to build the RS232 cable, which has a somewhat non-standard pinout:

RJ12 Pin	DB9 Pin	Function
1	2	Serial TX > RX
2	3	Serial RX > TX
3	5	Ground
4		Ground
5		VCC (+15V)
6		VCC (+15V)

RJ12 and DB9 Pins are counted right to left, with the contacts facing you.

That's the same pinout as **this project**, which also interfaces the Renology to a Pi but doesn't give much documentation:



I'm using **this RS232 to TTL Serial converter**. On the TTL side I have ground connected to an ESP32 ground, VCC to ESP32 3.3v, the adaptor's RXD to ESP32 pin TX2, the adaptor's TXD to ESP32 pin RX2.

I'm currently using this sketch:

```
#define RXD2 16
#define TXD2 17
```

```
String incoming;

void setup() {
  Serial.begin(115200);
  Serial.println("Started!");
  Serial2.begin(9600, SERIAL_8N1, RXD2, TXD2);
}

void loop() {
  while (Serial2.available() >= 1) {
    //Serial.println("checking...");
    Serial.print(char(Serial2.read()));
    //incoming = Serial2.readStringUntil('\n');
    //Serial.println(incoming);
  }
}
```

No data reaches the console other than boot data, but I think that's because I'm not checking the charge controller's registers. Here are the registers are described at **that NodeJS project**:

The below is a list of supported registers for device information:

0x00A	Controller voltage rating	Volts
0x00A	Controller current rating	Amps
0x00B	Controller discharge current rating	Amps
0x00B	Controller type	
0x00C - 0x013	Controller model name	
0x014 - 0x015	Controller software version	
0x016 - 0x017	Controller hardware version	
0x018 - 0x019	Controller serial number	
0x01A	Controller MODBUS address

The below is a list of supported registers for state data:

0x100	Battery Capacity	Percent
0x101	Battery Voltage	Volts
0x102	Battery Charge Current	Amps
0x103	Battery Temperature	Celcius
0x103	Controller Temperature	Celcius
0x104	Load Voltage	Volts
0x105	Load Current	Amps
0x106	Load Power	Watts
0x107	Solar Panel (PV) Voltage	Volts

0x108	Solar Panel (PV) Current	Amps
0x109	Solar Panel (PV) Power	Watts
0x10B	Min Battery Voltage Today	Volts
0x10C	Min Battery Voltage Today	Volts
0x10D	Max Charge Current Today	Amps
0x10E	Max Discharge Current Today	Amps
0x10F	Max Charge Power Today	Watts
0x110	Max Discharge Power Today	Watts
0x111	Charge Amp/Hrs Today	Amp Hours
0x112	Discharge Amp/Hrs Today	Amp Hours
0x113	Charge Watt/Hrs Today	Watt Hours
0x114	Discharge Watt/Hrs Today	Watt Hours
0x115	Controller Uptime	Days
0x116	Total Battery Over-charges	Count
0x117	Total Battery Full Charges	Count

Any guidance on how I would check one of those registers? I think it uses modbus to do so.

Thanks for any help.

---

wrybread 2 October 9, 2022, 3:38am

And aha, I found documentation for the Rover's modbus. It's a bear finding this stuff since their support forum is now defunct. But someone uploaded the file here:

> **Download ROVER MODBUS.DOCX (314.26 KB) now. Fast and easy at workupload.com**
>
> Download ROVER MODBUS.DOCX (314.26 KB) now. Fast and easy at workupload.com

I think it's mostly just enumerating the registers which are posted in my previous post though.

Edit: found it on github too:

> **KyleJamesWalker/renogy_rover/blob/master/reference/ROVER MODBUS.pdf**
>
> This file is binary. **show original**

Railroader 3 October 9, 2022, 3:40am

> wrybread:
>
> I haven't found any projects that do that

That's the normal outcome from searching.

Don't expect helpers to plow through project presentations to find out what You aim at. Select the facts and present them here.

Good, You've found the necessary RS232 to TTL converter.

> wrybread:
>
> On the TTL side I have ground connected to an ESP32 ground, VCC to ESP32 3.3v,

Be aware, know, there are 3.3 volt logic level families but also 5 volt families. Don't connect them "just like that".

Please provide schematics. Without that analysis are just guesses consuming time for no good.

> wrybread:
>
> I think

How would that help forum helpers?

---

wrybread 4 October 9, 2022, 3:48am

> Don't expect helpers to plow through project presentations to find out what You aim at. Select the facts and present them here.

Apologies if I sounded like I was expecting people to do google research. I wasn't, I'm just presenting the facts as I know them as background to my question.

> Be aware, know, there are 3.3 volt logic level families but also 5 volt families. Don't connect them

> "just like that".

Sorry I should have mentioned, the Amazon reviews for the TTL converter say it works with 3.3v.

And confirmed, the Renogy uses modbus. So really where I'm stumped is how to read a modbus register using a TTL converter like this. For example, how would I read register 0x00A.

---

wrybread 6 October 9, 2022, 4:16am

> noiasca:
>
> I can't open the docx. Maybe you convert it to pdf and post it as attachment to this forum.

Good idea, attached.
ROVER MODBUS.pdf (549.9 KB)

Railroader 7 October 9, 2022, 4:19am

Sorry, just gave You the hard reality, what helpers are willing to, and not. There are highly experienced and skilled helpers once working for NASA, the space industry, the military.... They don't like questions lacking facts, like "mama help me".

You better present facts according to the expectations of forum. That's told in the advice in "How to get the best out of this forum".

> wrybread:
>
> TTL converter say it works with 3.3v.

Good! Signals can be sent from 3.3 volt logic to 5 volt logic but not the opposite.

> wrybread:
>
> the Renogy uses modbus

What is Renogy? Some helpers might know and can reply. Other helpers don't reply.....

> wrybread:
>
> how to read a modbus register

Modbus has no registers. If it works handling modbus the question is accessing the item in the other end of the line, the device being accessed.
There is a mix of RS32 and modbus as I feel.
Reading the solar stuff is the aim. Right?
Late here. Tomorrow is another day.

---

noiasca 8 October 9, 2022, 4:27am

> wrybread:
>
> Good idea, attached.

ok, MODBUS, nothing special at first glance. They even have copied/pasted parts of the MODBUS specification in their document.

Install the library and start with one register. For example page 17 chapter 3.5 read the SoC as you have a good example. The Table on page 2 is missing the function code. So better start with that particular example in 3.5

---

markd833 9 October 9, 2022, 6:40am

If your controller is using modbus over RS232 then it might be worth using your PC along with a USB-serial adapter to try and talk to it using one of the free modbus tools out there.

It will give you a feel for the messaging without having to upload numerous sketches to your board.

---

wrybread 10 October 9, 2022, 6:41am

> For example page 17 chapter 3.5 read the SoC as you have a good example.

Thanks. That's:

> 3.5 To read battery capacity SOC, and the PDU address is known to be 0100H
> To send: 01 03 0100 0002 C5F7
> To receive: 01 03 02 0064 B9AF
> Parsing: (the highest byte OOH is not used) the battery capacity SOC is 64H% （decimal 100%)

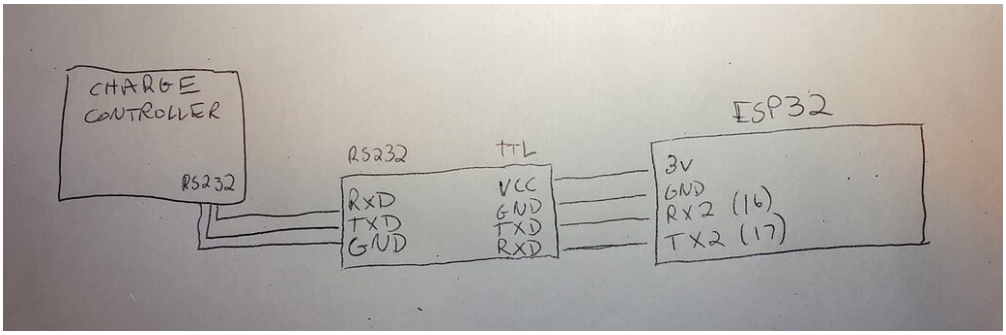
I'm not sure how I plug in those values. Here's a starting point I'm working on based on example code **here** (using "**Modbus Master**" Library from Doc Walker):

```
#include <ModbusMaster.h>

// instantiate ModbusMaster object
ModbusMaster node;

#define RXD2 16
#define TXD2 17

void setup()
{
  Serial.begin(115200);
  Serial.println("Started!");

  // create a second serial interface for modbus
  Serial2.begin(9600, SERIAL_8N1, RXD2, TXD2); // is SERIAL_8N1 corre

  node.begin(0, Serial2); // I think the addy is 255. With anything e

}


void loop()
{
```

Note that I'm trying to use modbus on a second serial port so I can still print output to the console.

In theory that should read 6 holding registers starting with 0x101 (battery voltage), but so far no joy. The above script just outputs "E2" every iteration (for "Serial.println(result)"), which is a timeout. So maybe I'm just not connecting to the controller correctly.

Here's a schematic of my wiring:



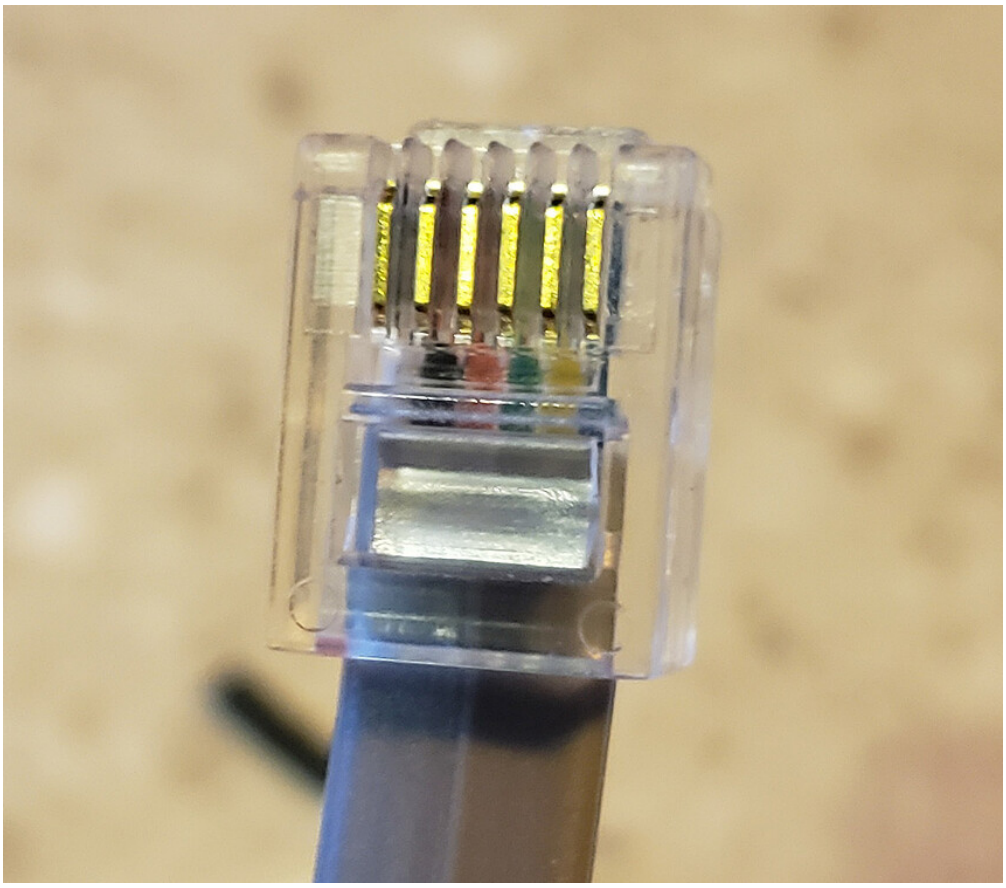
---

wrybread 11 October 9, 2022, 8:41am

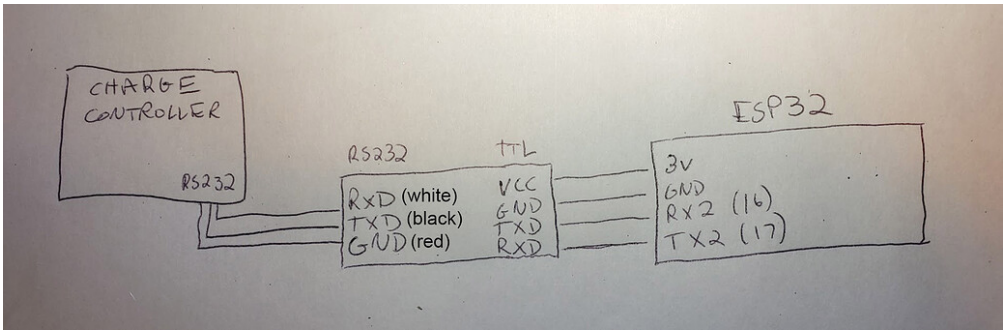
Phew, got it working. Thank you so much for all the help. The "Modbus Master" Library from Doc Walker put me on the right path.

For anyone else who comes this way: the modbus address of the Renogy Wander is 255! If that's in the docs I can't find it. Also make sure your RJ12 cable is making contact with all the pins in the RS232 port, it's possible mine wasn't making full contact.

And for making the cable, here's a picture of my RJ12 cable's jack showing the wire colors. You use the first 3 wires starting from the left (on my cable that's white, black, red).



And here's an updated version of my wiring diagram, showing the wire colors coming from the RJ12 cable:



Here's my current code, this gets all 30 data registers (not all are used on my cheapie controller) and 17 info registers. I haven't processed or formatted the data yet, but there's lots of processing and explanation in that nodejs project above.

```
// Pins used by ESP32 for the 2nd serial port. Not sure if this is po
#define RXD2 16
#define TXD2 17

// Number of registers to check. I think all Renogy controlls have 30
// data registers (not all of which are used) and 17 info registers.
int num_data_registers = 30;
```

```
int num_info_registers = 17;




// https://github.com/syvic/ModbusMaster
#include <ModbusMaster.h>
ModbusMaster node;


void setup()
{
  Serial.begin(115200);
  Serial.println("Started!");

  // create a second serial interface for modbus
```

On my controller with no solar panel connected that outputs:

```
Successfully read the data registers!
120
0
4121
0
0
0
0
0
0
0
120
120
0
0
0
0
0
0
0
0
3
0
```

Sure is going to be nice having access to the data from these charge controllers without having to use a

Raspberry Pi! Life is too short for long bootup times, corruptible SD cards and excessive power overhead, especially in a solar environment.

Some difference between ESP32 Rover and ESP32 Wroom when using second serial port?

How to decode these two bytes? (solved!)

system Closed 12 April 7, 2023, 8:41am

This topic was automatically closed 180 days after the last reply. New replies are no longer allowed.

Related Topics

Topic	Replies	Activity
RJ12 cable question	5	April 2, 2023
Some difference between ESP32 Rover and ESP32 Wroom when using second serial port?	10	April 11, 2023
CRLF RS232 in Arduino	11	May 5, 2021
Interfacing the a Renogy Wanderer Controller	2	April 30, 2022
Help using an Arduino to read UART messages?	9	May 5, 2021