QUICK START GUIDE FOR SDUMBUSCARD

SOLA HD Modbus Quick Start Guide

There are many Modbus TCP Client PLCs and software applications in the market. The scope of this document is to provide general information and the Modbus register mapping for the SOLA HD UPS so the end user knows which registers to access from their Modbus client. Also, the basic steps for any application will be covered.

Basic set up:

The first step will be to set the IP address of the Modbus TCP module. The user will have to set the IP address of the module using the **IPconfig** tool that is provided on the website and the Ethernet connection from your PC to the SDU AC-A (visit: www.solahd.com)
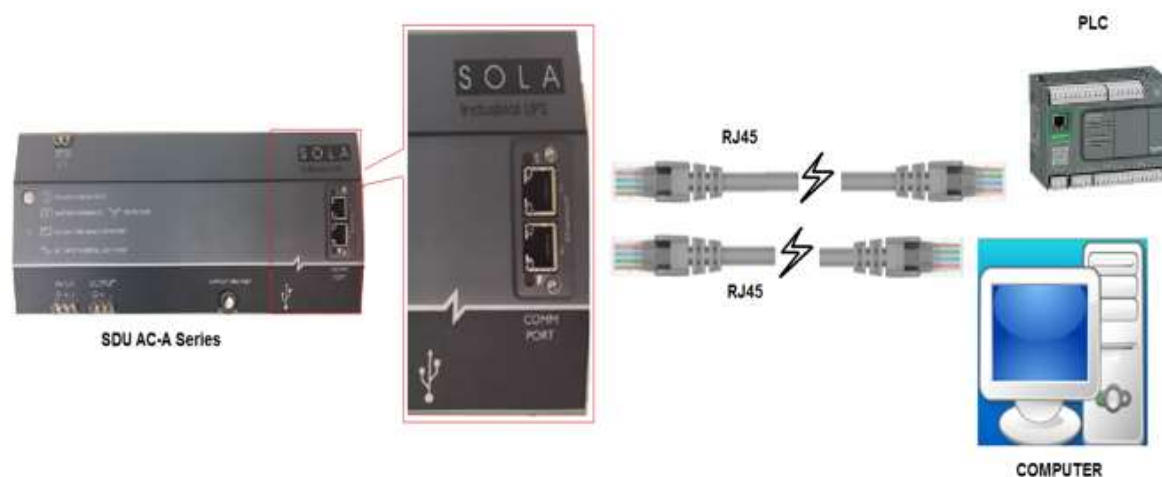

FIG.01 Wiring Connection of SDU AC-A Series UPS to Computer

When you are in Ipconfig, you will need to set the UPS to be on the same subnet range as the Modbus TCP client that you are using. This means the first three octets of the IP address should be the same, and the fourth octet should be unique.
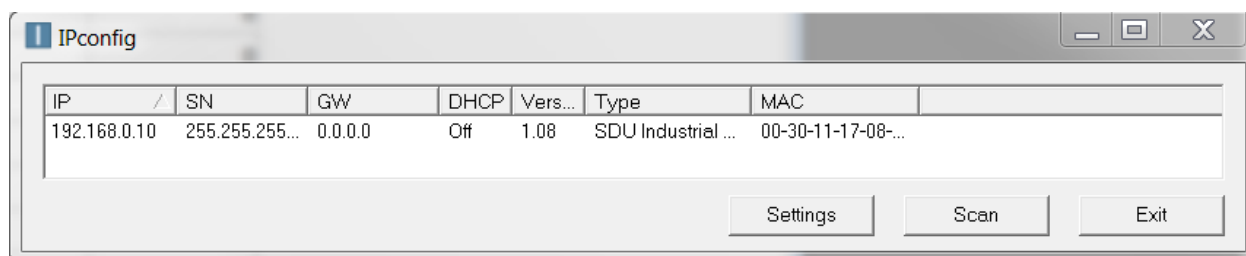

FIG.02 IPconfig Setup

QUICK START GUIDE FOR SDUMBUSCARD

Register Mapping and access:

With Modbus TCP there are a few ways to think about accessing data. Different Modbus TCP clients may use different interpretations or require different information for reading and writing data. Some Clients will require a prefix that tells you which register set you are reading/writing. Some will require the user to choose a function code that will access the data points. Although these seem different, they are equivalent.

Prefix –

What this would mean for a PLC that requires a prefix is that you read register 300001. The 3 at the very beginning of this number by definition tells the PLC that it is accessing input registers. The 1 at the end tells you which specific register in the input registers to read.

Function code –

For another system that requires a function code, you will choose the code. To read the first two bytes of Status-1 you will choose the function "04 Read input registers". This may look slightly different depending on your system, but the key part is that it is function code #4 which translates to reading the input registers. The user would then define what register to read.

The only function codes that you will need for your UPS are:

- #3 Read Holding Registers
- #4 Read input Registers
- #6 Write Single Register
- #16 Write Multiple Registers

Side notes –

Some systems interpret the first register as 0 and other ones start at 1. This has to do with how different manufacturers interpret the Modbus standard.

You can access the status bytes using functions codes 3 and 4 (reading input and holding registers). This is useful if your Modbus TCP client only supports Holding register access.

Mapping tables (Byte 1 = LSB and Byte 2 = MSB)-

Input register number (3XXXXX)	Description
1	Byte 1 = Load Level, Byte 2 = Battery Level
2	Input Voltage (V)
3	Output Voltage (V)
4	Byte 1 = Input Voltage, Byte 2 = Output Voltage
5	Byte 1 = UPS Status-1, Byte 2 = UPS Status-2
6	Model Number
7	Byte 1 = PRE-SD min Time, Byte 2 = PRE-SD sec Time
8	Byte 1 = PRE-On min time, Byte 2 = UPS FW version

TBL.01 Input Registers

Holding register number (4XXXXX)	Description
1	Command byte 2 and 1
2	Command byte 4 and 3
3	Command byte 6 and 5
4	Command byte 8 and 7
2048	Byte 1 = Load Level, Byte 2 = Battery Level
2049	Input Voltage (V)
2050	Output Voltage (V)
2051	Byte 1 = Input Voltage, Byte 2 = Output Voltage
2052	Byte 1 = UPS Status-1, Byte 2 = UPS Status-2
2053	Model Number
2054	Byte 1 = PRE-SD min Time, Byte 2 = PRE-SD sec Time
2055	Byte 1 = PRE-On min time, Byte 2 = UPS FW version

TBL.02 Holding Registers

Examples:

Reading –

In order to read the first two bytes of Status-1 (see register mapping table) you can read the first input register.

FIG.03 Modbus Poll Setup
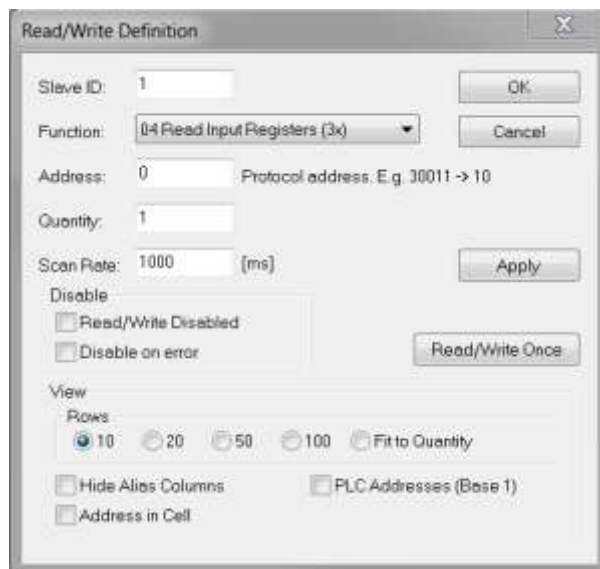
Some systems may even combine the prefix and function code options for clarity (see "Function" in FIG.03). For example in Modbus Poll (Modbus simulator software) this is what it looks like when set it up to read the first two bytes of Status-1.

What can be seen is that the first 2 8-bit bytes of Status-1 are together in one value. This is because one register is a 16-bit value. It is easier to split up a register into byte 2 and 1 if you view the value in hex format. Lastly, you will note that Byte 2 is the most significant byte and byte 1 is the least significant byte. These points are much easier to see when you compare the value in the simulator with the data in the web server.
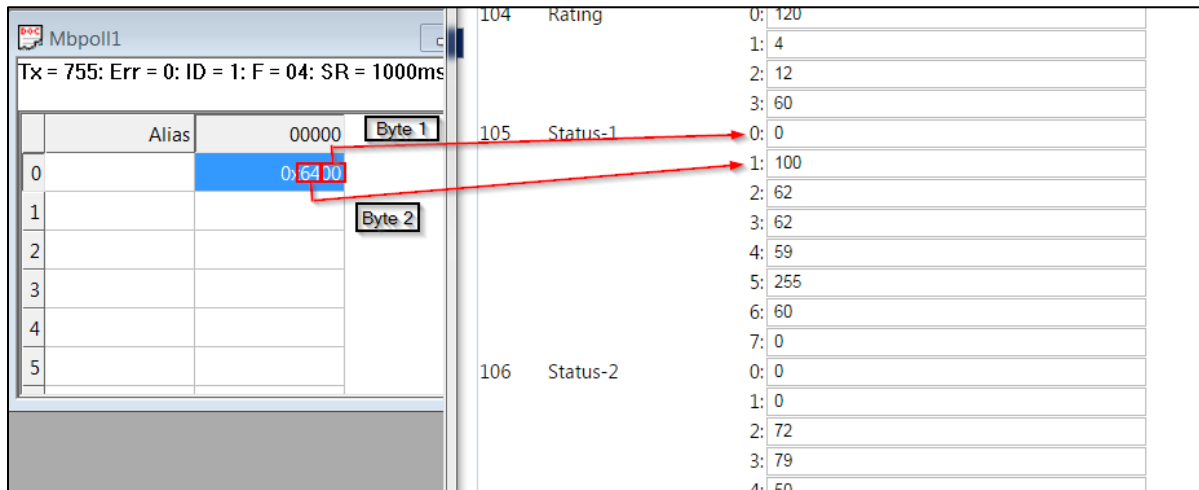
FIG.04 Modbus Simulator Data Compared with Data in the Web Server

As can be seen above, when you convert Byte 2 from 64h you get 100 like in the web server. It is worth mentioning that in many cases you can choose to read multiple registers at once by defining a quantity. This quantity parameter can be seen in FIG.03 above.

Writing –

For the UPS you will likely want to use function code 16 (Write multiple registers) to write a whole command at once, although you could use function code 6 (write single register). You would select either of these commands in the menu seen in FIG.03. For this example we will disable and re-enable the alarm using the 8 command bytes. From command specification in the "SDU A-Series Comm Cards" manual, here are the hex commands.

Disable Alarm:
Command Message Structure: 62h 00h 00h 00h 00h 00h 00h 0Dh
Enable Alarm:
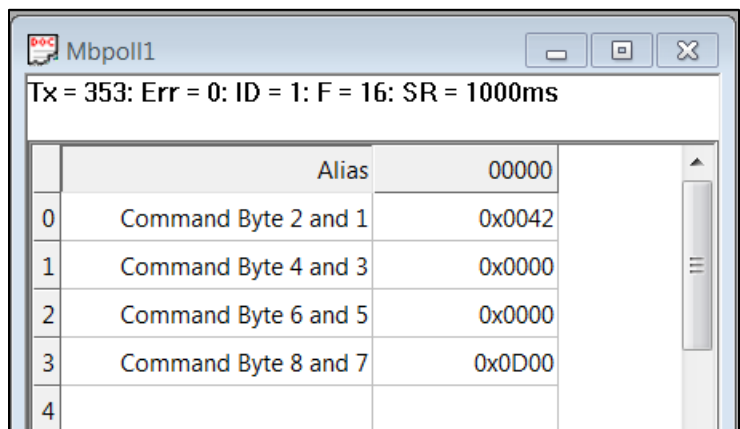Command Message Structure: 42h 00h 00h 00h 00h 00h 00h 0Dh

When you unplug the UPS from the power it should periodically beep, this is the alarm. In order to turn this alarm off you will write the following (as demonstrated in Modbus Poll).

FIG.05 Modbus Written Data, Disable Alarm

QUICK START GUIDE FOR SDUMBUSCARD
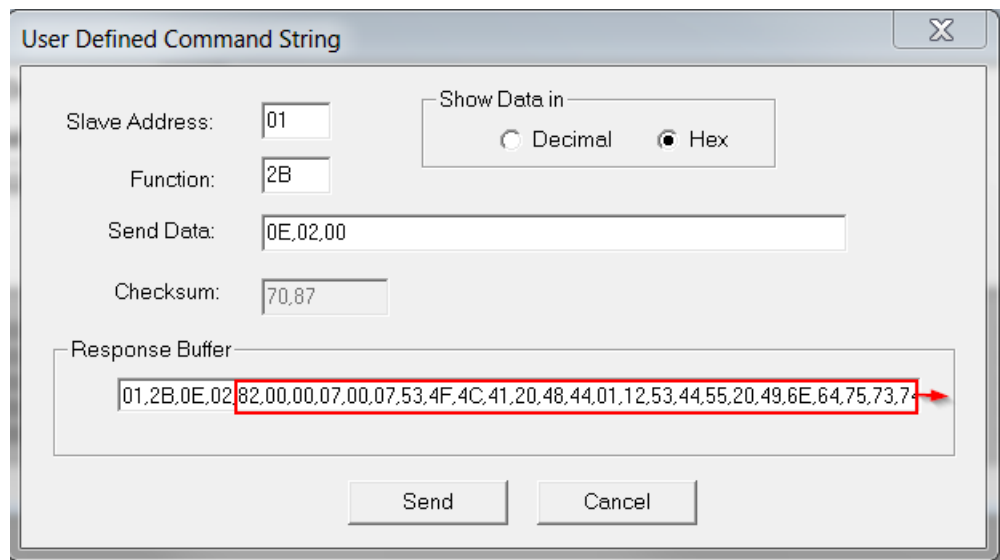
To turn it back on change the value 62h in Byte 1 to 42h.

FIG.06 Modbus Written Data, Enable Alarm

Identity information request –

Some users may not be able to do this request as the ability to send an identity request is not supported in their particular Modbus TCP client. You must use function code 2Bh (43 in decimal) and put in the "Send data" as 0E,02,00 (14,2,0 in decimal). It is better to use hexadecimal for this as it is easier to convert the hex respond into the ASCII characters you would expect. An example of this can be seen in the image below. Modscan64 was used

FIG.07 Modscan64 User Defined Request

QUICK START GUIDE FOR SDUMBUSCARD

If you convert the hex formatted numbers to their ASCII equivalent you will get this list of identity information in order:

Vendor Name - SOLA HD

Product Code - SDU Industrial UPS
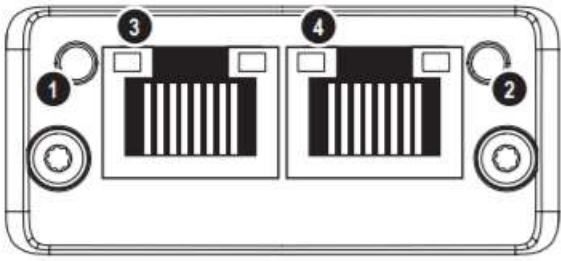
Major Minor Firmware Revision - 1.03

Vendor URL - www.solaHD.com

Product Name - SDU Industrial UPS

Model Name - SDU Industrial UPS

APENDIX:

LEDs –

Front View (Ethernet Connectors)

#	Item	Connector
1	Network Status LED	Ethernet, 45
2	Module Status LED	
3	Link/Activity LED (port 1)	
4	Link/Activity LED (port 2)	

Network Status LED

LED State	Description
Off	No IP address or in state EXCEPTION
Green	At least one Modbus message received
Green, flashing	Waiting for first Modbus message
Red	IP address conflict detected, FATAL ERROR
Red, flashing	Connection timeout. No Modbus message has been received within the configured "process active timeout" time

Module Status LED

LED State	Description
Off	No power
Green	Controlled by a Scanner in Run state
Green, flashing	Not configured, or Scanner in Idle state
Red	Major fault (EXCEPTION-state, FATAL error etc.)
Red, flashing	Recoverable fault(s). Module is configured, but stored parameters differ from currently used parameters.

QUICK START GUIDE FOR SDUMBUSCARD

Simulators –

Sometimes for troubleshooting Modbus TCP simulators can be very helpful. Here are some examples:

ComTest PRO – http://www.baseblock.com/PRODUCTS/demosoftware.htm

Modbus Poll – http://www.modbustools.com/modbus_poll.html

Modscan64 – https://www.win-tech.com/html/demos.htm

END