Sunways NT4000

The Sunways NT4000 is a solar inverter. It stores data locally, which can be retrieved via eighter a RS485 or a RS232 interface. When using multiple Sunways inverters, you can connect them to each other via the RS485 interface, so that one acts as master and handles the communication.

RS485 Interface

This serial interface is used for communication with the inverters. With the help of for example the HALJIA USB zu RS485 Konverter you can connect the inverters to your computer or Raspberry Pi. For me I could only communicate via RS485 when the inverters where actually receiving solar power.

Settings for the Interface

Baud rate	9600
Data bits	8
Parity	None
Stop bits	1
Handshake	none

Software

There are multiple different software solutions to read and write data via software. The software from the manufacturer is called *Sunways Monitor 2.0*, but it is not available for download anymore.

FHEM

Another software to communicate with the inverters is the house automation software **FHEM**. It support many different modules and Prof. Dr. Peter A. Henning created a module for Sunways NT inverters called NT5000 (seems to be moved to 70_SolarView). Targeted actually at the NT5000, the inverters NT1800, NT2600, NT4000 and NT6000 have the same protocol. Further information can be found in the wiki. To use the module you firstly need to get your FHEM installation up and running, for example as docker container. There you need to mount a volume for the configuration to be persistent and also pass the USB device to the container.

```
docker run -d -p 8083:8083 --device=/dev/serial/by-id/usb-1a86_USB2.0-Serial-if00-port0:/dev/ttyUSB0 -v
/home/pi/fhem:/opt/fhem fhem/fhem
```

After that you need to copy the 70_NT5000.pm (fhem.txt) file from contrib to the FHEM folder so that the module can be loaded. Test if the module can be loaded by running the command **reload NT5000** on the FHEM web interface. If it succeeds, you can then create the solar module object by running **define solar NT5000 /dev/ttyUSB0**. Then you should see the values of the inverter on the website. Sadly I could not find any option to get support for multiple inverters. So the only way to use this software with multiple inverters would be to wire each inverter to the computer on its own. Publishing the received data can be done for example via MQTT.

```
define mosquitto MQTT <IP>:<PORT>
define mqttGeneric MQTT_GENERIC_BRIDGE
attr mqttGeneric IODev mosquitto
attr mqttGeneric globalPublish *:topic={"<MQTTTOPIC>/$device/$reading"}
```

Example config: config.txt

SolarView aka SolarMax-Proxy

Another software to read and write the data is SolarView, a data logger for various solar inverters. While it doesn't support Sunways support out of the gate, a community member of this forum created a proxy for the inverters. The software proxies the Sunways inverter to act like a SolarMax inverter, which the software supports. The solarmax-proxy is available on SourceForge, but no longer maintained by the the original creator. In this thread the evolution and usage of the proxy is discussed and also multiple different binary files where shared. Here are multiple files saved for preservation:

- solarmax-proxy-code.zip
- ssl_7-patch.zip
- smp_rpi single always log message only with parameter debug.zip

- [smp-0.21-mipsel.zip](#)

- [smp_v0.23 code and binary rpi and 7390.zip](#)

- [smp 0.22 all binaries 7170 7270 73xx 7272 7490 x86 rpi.zip](#)

- [smp 0.22 all binaries 7170 7270 73xx 7272 7490 x86 rpi NEW.zip](#)

- [smp 0.22.zip](#)

- [smp compiled to raspberry pi changed for clear code.zip](#)

- [smp compiled for raspberry pi.zip](#)

- [useful_infos.txt](#)

The requirement for smp to work are:

- RS485 from the inverter connected to a serial to network adapter (or connected to the raspberry pi and forwarded via ser2net software via network)
- SMP connected to the adapter
- SolarView configured to use SMP as SolarMax inverter

Example calls:

- ./smp -c 1 -l 11200 -f /var/log/smp.log
- ./smp -c 2 -l 11200
- ./smp -t 5 -c 1 -f smp.log -l 5000
- ./smp.rpi -h 192.168.178.74 -p 6000 -d 2 -c 3 -f smp.log

As all these step dont seem like a stable solution for me, I did not try this route, but the source code gained from this proxy was more than helpful.

Python Script

Which leads to the last software, a little python script written by myself. As it still is a work in progress, here is the current version of the software.

[test-2.py](#)

Backup: [SMP.zip](#) and [SUNWAYS.zip](#)

Example output:

```
 pi@raspberrypi: ~
pi@raspberrypi:~ $ python3 test.py
Data for inverter at address 1.
Serialnumber: 2433A4000099s
Firmware: 11NT8-C3C_0
Month 1: 90.4
Month 2: 77.4
Month 3: 57.4
Month 4: 83.2
Month 5: 394.7
Month 6: 395.2
Month 7: 454.5
Month 8: 527.7
Month 9: 427.1
Month 10: 357.6
Month 11: 228.1
Month 12: 0.0
Data for inverter at address 2.
Serialnumber: []
Firmware: []
Month 1: 563.4
Month 2: 97.1
Month 3: 89.8
Month 4: 70.6
Month 5: 97.3
Month 6: 395.6
Month 7: 396.1
Month 8: 458.8
Month 9: 526.9
Month 10: 427.9
Month 11: 358.1
Month 12: 237.7
Data for inverter at address 3.
Serialnumber: ñ<1433A4002437p
Firmware: []
Month 1: 563.4
Month 2: 89.9
Month 3: 6.6
Month 4: 96.0
Month 5: 311.1
Month 6: 423.6
Month 7: 432.6
Month 8: 237.5
Month 9: 83.6
Month 10: 107.1
Month 11: 244.0
Month 12: 326.9
```

Working Software

- sunways.zip

File collection backup: Multiusb.zip