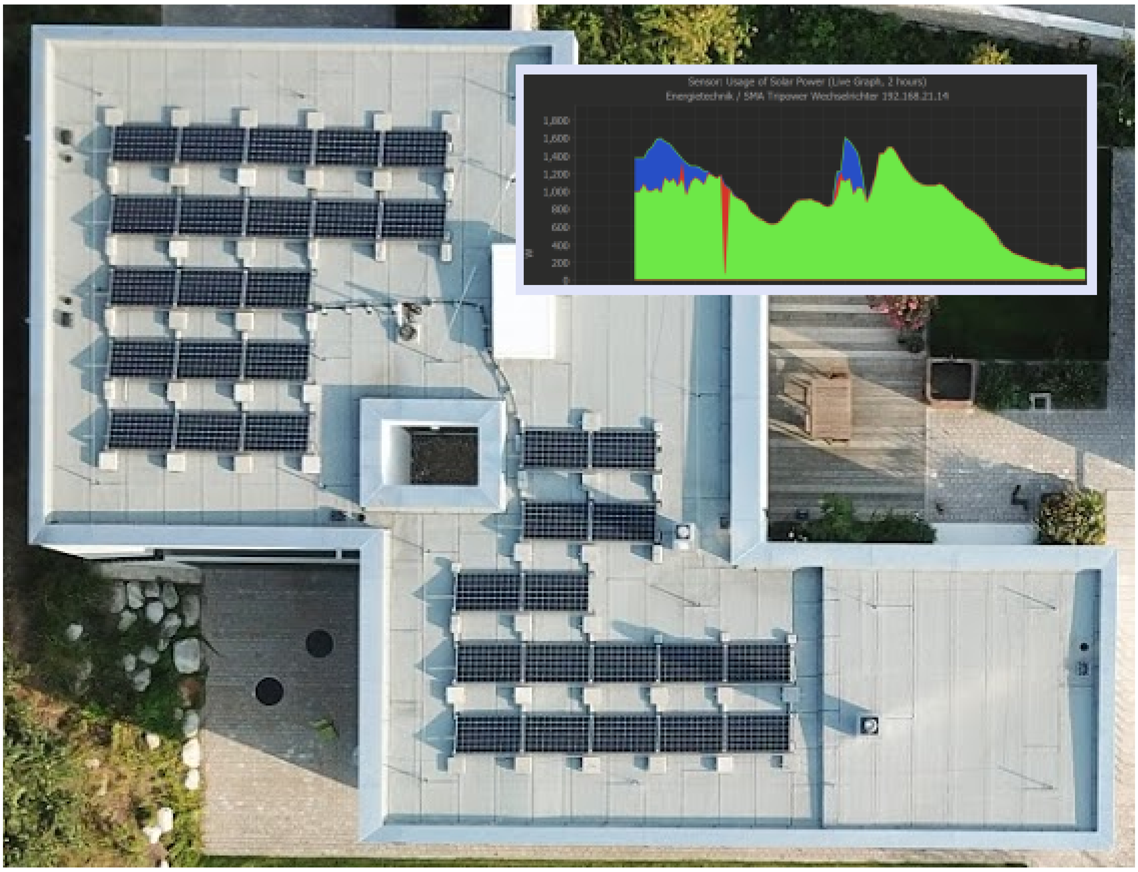
Dirk Paessler's Personal Blog

Stuff that seems to be interesting

Monitoring solar panels and our home’s energy usage/mix via Modbus using PRTG (ft. SMA Sunny Island and SMA Sunny Tripower)

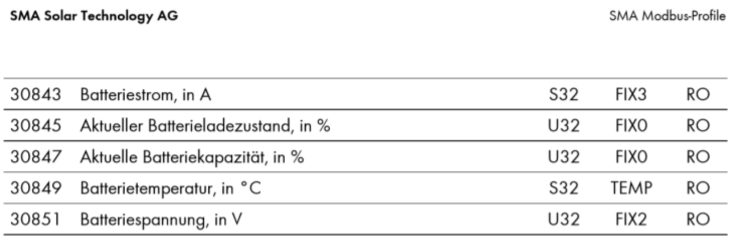

Sensor: Usage of Solar Power (Live Graph, 2 hours)
Energietechnik / SMA Tripower Wechselrichter 192.168.21.14
1,800
1,600
1,400
1,200
1,000
800
600
400
200
0
W

We have solar panels on our roof which are able to generate a peak power of 10 kW. An energy controller controls the converter, our heat pump and our battery with the goal that most of the solar energy is used locally. First we use the solar energy in our home, we charge the 230V-battery, and we load the heat storage tank. Only if even more solar power is generated we send the rest into the grid.

To visualize this and I am monitoring this system with PRTG using the Modbus protocol.

The converter is a SMA Sunny Tripower 5000TL-12000TL and the battery system is a SMA Sunny Island 4.4M / 6.0H / 8.0H system. Both support the ModBus protocol and are connected to a Loxone controller and the heatpump via a private Ethernet network (see my previous blog article for details about the multi-homed network setup of the remote probe).

The Modbus documentations for the SMA Sunny Tripower as well as the battery system are identical and are available from the product page on the vendor's website. Both devices have a whole bunch of metrics to choose from:



SMA Solar Technology AG SMA Modbus-Profile

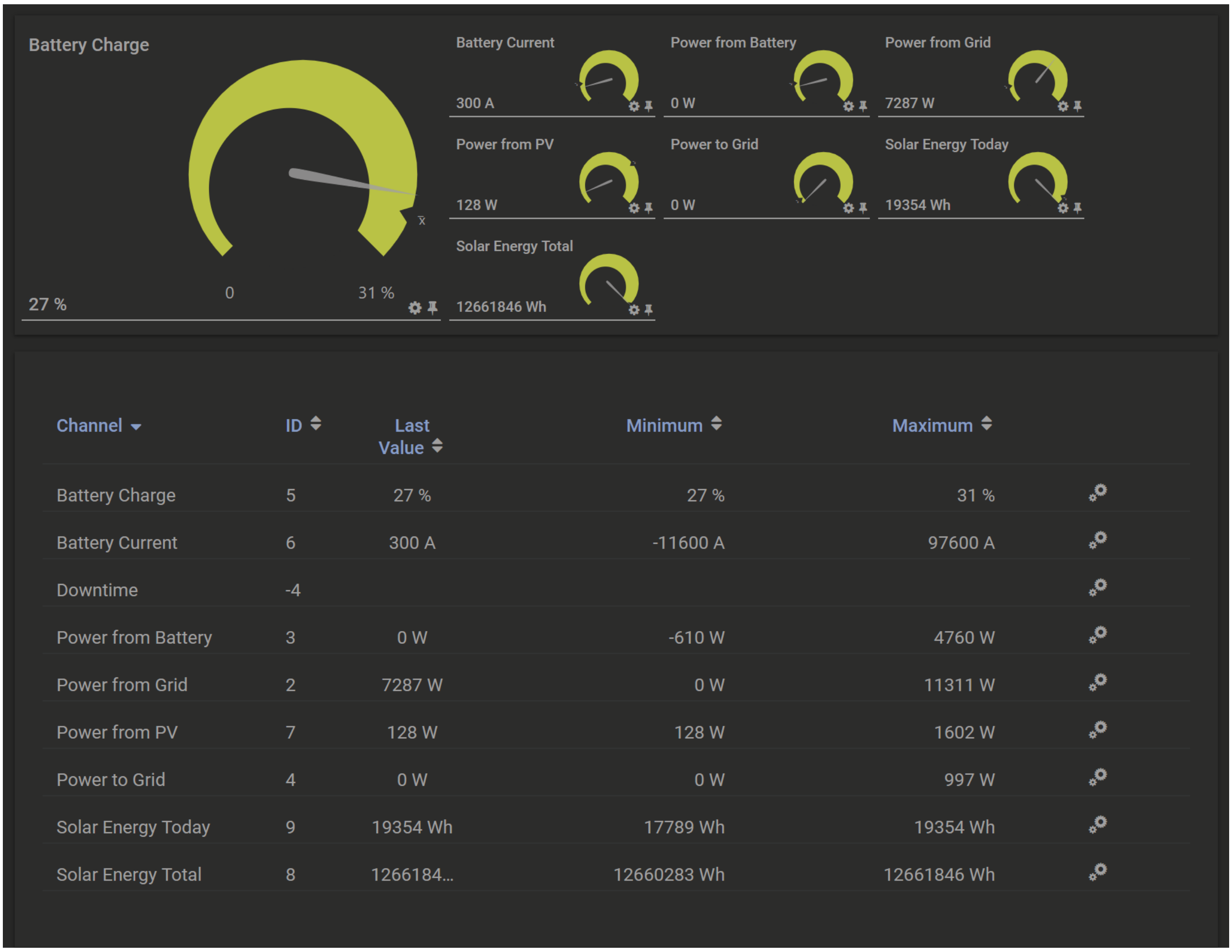
30843	Batteriestrom, in A	S32	FIX3	RO
30845	Aktueller Batterieladezustand, in %	U32	FIX0	RO
30847	Aktuelle Batteriekapazität, in %	U32	FIX0	RO
30849	Batterietemperatur, in °C	S32	TEMP	RO
30851	Batteriespannung, in V	U32	FIX2	RO

Again I decided to write a Powershell script that I can use as an EXE/Script Advanced sensor in PRTG. The metrics are encoded differently this time, so I had to change the heatpump script a lot.

I also need to talk to both systems (converter and battery) to get all the interesting parameters.

```
# Monitoring of SMA Sunny Tripower and SMA Sunny Island
# via Modbus Protokoll

param(
        [string] $remoteHost = "enter IPs below!!",
        [int]$port = 502
)

function modbusread ([string]$remoteHost, [int]$port, [int]$startaddress,[int]$bytecount,[string]$signed) {

    write-host "Reading Register #$startaddress"

    # Build Request Data

        [byte[]]$sendbuffer=00,110  # Transaction Identifier
        $sendbuffer+=00,00                      #Protocol identifier
        $sendbuffer+=00,06                      #Length
        $sendbuffer+=03                         #Unit ID
        $sendbuffer+=04                         #Function Read Input Registers
        $sendbuffer+=[byte]([math]::Truncate(($startaddress)/256)),([system.byte](($startaddress)%256))
        $sendbuffer+=00,($bytecount)

    # Send Request Data

        $tcpclient = new-object System.Net.Sockets.TcpClient($remoteHost, $port)
        $netStream  = $tcpclient.GetStream()
        $netStream.write($sendbuffer,0,$sendbuffer.length)
        start-sleep -milliseconds 50

    # Receive Data

        [byte[]]$recbuffer = new-object System.Byte[] ([int]($bytecount+9))
        $receivedbytes = $netStream.Read($recbuffer, 0, [int]($bytecount+9));
        $netStream.Close()
        $tcpclient.Close()

    # Process Data

        $resultdata = $recbuffer[9..($recbuffer[8]+8)]

    [byte[]] $bytes = $resultdata[3],$resultdata[2],$resultdata[1],$resultdata[0] # need to reverse byte order

    if ($signed -eq "signed")    {
        $result=[bitconverter]::ToInt32($bytes,0);
        write-host  "Received int=" $result
    }
    else     {
        $result=[bitconverter]::ToUInt32($bytes,0);
        write-host  "Received  uint=" $result
    }
    $result
}

# Process One Dataset

function onedataset([string]$name, [int]$theid, [string]$unit, [int]$divider, [string]$type,[string]$signed) {
    write-host "==== $name ===="
    $value=((modbusread $remoteHost $port ($theid) 4 $signed)/$divider)
    if ($value -eq 4294967295 -Or$value -eq 2147483648  -Or $value -eq  -2147483648)
        { $value=0 }
            "    <result>`r`n"
            "        <channel>"+$name+"</channel>`r`n"
            "        <customunit>"+$unit+"</customunit>`r`n"
            "        <value>"+($value/$divider)+"</value>`r`n"
            "        <float>1</float>`r`n"
            "        <mode>"+$type+"</mode><SpeedTime>Hour</SpeedTime>`r`n"
            "    </result>`r`n"
        }


# Main code

write-host "Let's go..."
[string]$prtgresult=""
$prtgresult+="<?xml version=""1.0"" encoding=""Windows-1252"" ?>`r`n"
$prtgresult+="<prtg>`r`n"
[bool]$errorfound = $false


try {

    $remoteHost = "192.168.21.15" # Let's look at the Battery System SMA Sunny Island

    $prtgresult+=onedataset "Power from Grid" 30865 "W" 1 "Absolute" "signed"
    $prtgresult+=onedataset "Power from Battery" 30775 "W" 1 "Absolute" "signed"
    $prtgresult+=onedataset "Power to Grid" 30867 "W" 1 "Absolute" "signed"

    $prtgresult+=onedataset "Battery Charge" 30845 "%" 1 "Absolute"
    $prtgresult+=onedataset "Battery Current" 30843 "A" 1 "Absolute" "signed"

    $status=(modbusread $remoteHost $port (30201) 4)
    switch ($status) {
        "35"    {$prtgresult+="<text>Status: Error</text><error>Status: Fehler</error>`r`n"}
        "303"   {$prtgresult+="<text>Status: Off</text>`r`n"}
        "307"   {$prtgresult+="<text>Status: OK</text>`r`n"}
        "455"   {$prtgresult+="<text>Status: Warning</text>`r`n"}
    }

    $remoteHost = "192.168.21.14" # Let's look at the PV System SMA Sunny Tripower


    $prtgresult+=onedataset "Power from PV" 30775 "W" 1 "Absolute" "signed"
    $prtgresult+=onedataset "Solar Energy Total" 30529 "Wh" 1 "Absolute"
    $prtgresult+=onedataset "Solar Energy Today" 30535 "Wh" 1 "Absolute"

    $prtgresult+="</prtg>"
}
catch {
        write-host "Unable to Connect and retrieve data $_.Exception.Message"
        $prtgresult+="   <error>2</error>`r`n"
        $prtgresult+="   <text>Unable to Connect and retrieve data  "+($_.Exception.Message +" at "+ $_.InvocationInfo.Positi
    $prtgresult+="</prtg>"
        $errorfound = $true
}

if ($errorfound) {
        write-host "Error Found. Ending with EXIT Code" ($prtgresult).prtg.error
}
write-host "Sending PRTGRESULT to STDOUT"
$prtgresult
```

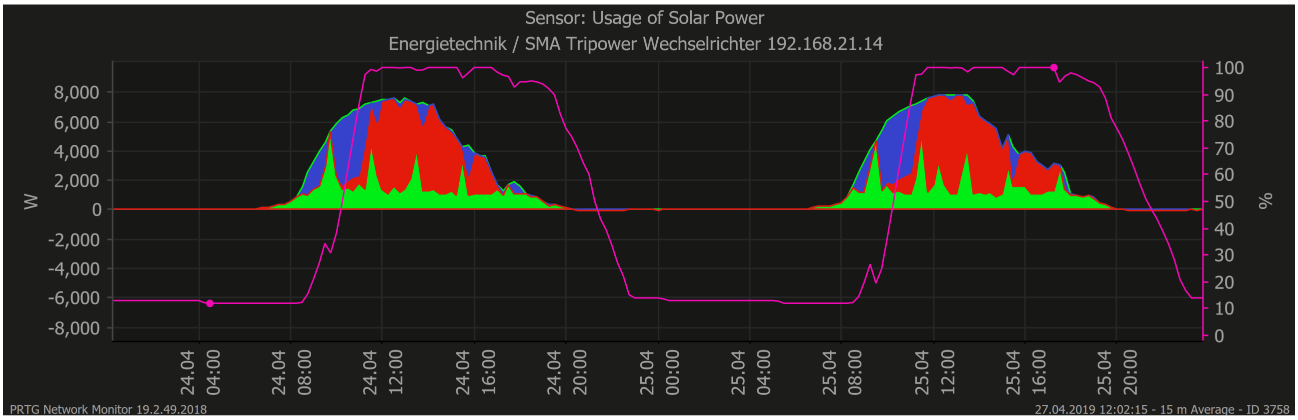
This script gives me the following metrics in PRTG:



Channel	ID	Last Value	Minimum	Maximum
Battery Charge	5	27 %	27 %	31 %
Battery Current	6	300 A	-11600 A	97600 A
Downtime	-4			
Power from Battery	3	0 W	-610 W	4760 W
Power from Grid	2	7287 W	0 W	11311 W
Power from PV	7	128 W	128 W	1602 W
Power to Grid	4	0 W	0 W	997 W
Solar Energy Today	9	19354 Wh	17789 Wh	19354 Wh
Solar Energy Total	8	1266184...	12660283 Wh	12661846 Wh

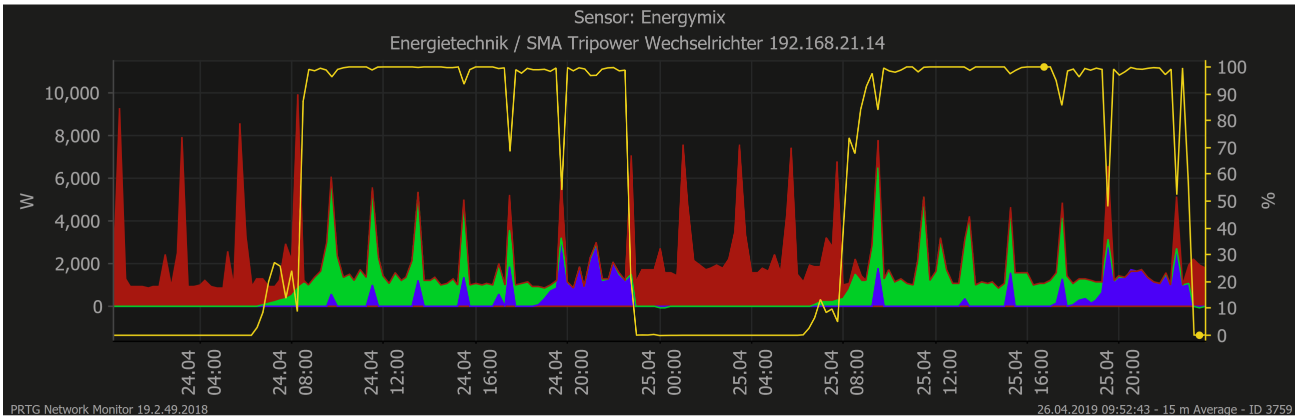
Using a Sensor Factory sensor in PRTG I created two derived sensors that display the data in easy to read graphs:

This first graph shows the usage of solar power (what is the power from the solar panels used for?).



The green part is used in our house at once, blue goes into the battery and red is exported into the grid. The pink line shows the charge status of the solar battery. The daily total is almost 60 kWh on a sunny day in April.

This second graph shows the energy usage of our house (including heating/venting) and where the energy comes from:



- Green: Energy from solar panels
- Blue: Energy from the solar battery
- Red: Energy from grid
- Yellow: Percentage of solar power of total usage

As you can see on a sunny day in April between 8:00 and 22:00 almost all of the energy we use came from the solar panels, either directly or via the battery.

Share this:

Facebook Twitter LinkedIn Reddit Email

Author: Dirk Paessler

Founder and Chairman, Paessler AG View all posts by Dirk Paessler

January 31, 2019 / Smart Home and IoT /

2 thoughts on “Monitoring solar panels and our home’s energy usage/mix via Modbus using PRTG (ft. SMA Sunny Island and SMA Sunny Tripower)”

Pingback: The Basics of Modbus Monitoring with PRTG Network Monitor (With Custom Sensor Script) – Dirk Paessler's Personal Blog

Pingback: Comparison: Seven devices between €69 and €500 for monitoring environmental parameters like temperature and humidity with PRTG – Dirk Paessler's Personal Blog