3. Smart meter

Due to Dutch regulations, I was forced to have a smart meter installed. The smart meter has a P1 port that allows the user to read the values. I bought a Homewizard Wi-Fi P1 Meter, which is a quite affordable one.

3.1 Activating the P1 meter

To be able to use the P1 meter, you need to install the Homewizard Energy app from the playstore or appstore. The app will be used the connect the device to the WiFi network.

The device uses DHCP to get an IP address. We need to access the device with `curl` so it is fairly important to assign a specific IP address. In my `dhcpd.conf` I have added:

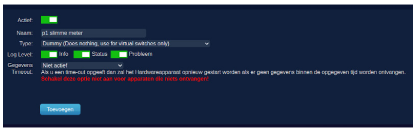
```
host p1 {
hardware ethernet 3c:39:e7:25:ff:ff;
fixed-address 192.168.xxx.xxx;
}
```

On my DNS server, I have added the host `p1` with that address.

3.2 Connecting to Domoticz

It took quite a bit of puzzling before I could get the values of the p1 meter into Domoticz. I created a script that is launched from the crontab and updates 3 virtual sensors.

First, I added a new hardware.



Actief:

Naam: p1 slimme meter

Type: Dummy (Does nothing, use for virtual switches only)

Log Level: Info Status Probleem

Gegevens Timeout: Niet actief

Als u een time-out opgeeft dan zal het Hardwareapparaat opnieuw gestart worden als er geen gegevens binnen de opgegeven tijd worden ontvangen.
Schakel deze optie niet aan voor apparaten die niets ontvangen!

For the hardware, I made three virtual sensors:

- slimme_meter_stroom type "Slimme meter elektra"
- slimme_meter_gas type "Gas"
- slimme_meter_active_power

The script below is used to get the data from the P1 meter and feed it, via the rest API to the virtual sensors.

```
#!/bin/bash
#REMOTE@ domoticz.home /usr/local/bin/p1_report
############################################################################
# Set or change the hostnames below to match your environment
############################################################################

DOMOTICZ=’domoticz.home:8888’
P1=p1.home

############################################################################
helpme(){
cat <<EOF

NAME:
          p1_report - report smart meter p1 values to Domoticz

SYNOPSIS:
          p1_report

DESCRIPTION:
P1_report reads the Homewizard Wi-Fi P1 meter and feeds the data into two
virtual sensors in Domoticz:
- slimme_meter_gas for the gas readings
- slimme_meter_stroom for the electicity reading
- slimme_meter_active_power
These three virtual sensors must be created manually in Domoticz.

The default URL for Domoticz is ’domoticz.home:8888’ and the default
URL for the Homewizzard P1 meter is ’p1.home’. These can be changed in
the script to match your environment.

P1_report is typicaly started by cron to ensure a regular data feed to
Domoticz.

EOF
}

if [ "$1" = "-h" ] ; then
          helpme
          exit 0
fi

report=/tmp/p1_report.$$
tmp=/tmp/p1_reportmp.$$

date > /tmp/last_p1
now=$(date)

curl --silent  "http://$DOMOTICZ/json.htm?type=command&param=devices_list" |
sed ’s/"//g;s/,//’ |
while read nv col val ; do
          if [ "$nv" = ’name’ ] ; then
                    name="$val"
          elif [ "$nv" = ’value’ ] ; then
                    echo "$val $name"
          fi

done > $tmp

if curl -s $P1/api/v1/data > $report ; then
          :
else
          exit 0
fi

for line in $(cat $report| jq . |sed ’s/^ *"//;s/,$//;s/": /:/p’) ; do
          var=${line%:*}
          val=${line#*:}
          case $var in
                    (total_power_import_t1_kwh)   usage1=$(printf "%8.3f" $val | sed ’s/[\. ]//g’) ;;
                    (total_power_import_t2_kwh)   usage2=$(printf "%8.3f" $val | sed ’s/[\. ]//g’) ;;
                    (total_power_export_t1_kwh)   return1=$(printf "%8.3f" $val | sed ’s/[\. ]//g’) ;;
                    (total_power_export_t2_kwh)   return2=$(printf "%8.3f" $val | sed ’s/[\. ]//g’) ;;
                    (active_power_w)              active_power_w=$val ;;
                    (active_power_l1_w)           active_power_l1_w=$val ;;
                    (active_power_l2_w)           active_power_l2_w=$val ;;
                    (active_power_l3_w)           active_power_l3_w=$val ;;
                    (total_gas_m3)                          total_gas_m3=$(printf "%8.3f" $val | sed ’s/[\. ]//g’) ;;
                    (gas_timestamp)                         gas_timestamp=$val ;;
          esac
done

echo "$now usage1=$usage1 usage2=$usage2 return1=$return1 return2=$return2 active_power_w=$active_power_w total_gas_m3=$total_gas_m3" >> /tmp/p1.l

if [ $active_power_w -lt 0 ] ; then
          prod=$active_power_w
          cons=0
else
          prod=0
          cons=$active_power_w
fi

if [ $usage1 = 0 ] ; then
          if [ $usage2 = 0 ] ; then
                    if [ $return1 = 0 ] ; then
                              if [ $return2 = 0 ] ; then
                                        exit 0
                              fi
                    fi
          fi
fi


meter_idx=$(sed -n ’s/ slimme_meter_stroom//p’ $tmp)
if [ "$meter_idx" = "" ] ; then
          echo "No slimme meter">> /tmp/last_p1
```

```
else
          curl --silent  "http://$DOMOTICZ/json.htm?type=command&param=udevice&idx=$meter_idx&nvalue=0&svalue=$usage1;$usage2;$return1;$return2;$c
          echo "$usage1;$usage2;$return1;$return2;$cons;$prod" >> /tmp/last_p1
fi

meter_idx=$(sed -n ’s/ slimme_meter_active_power//p’ $tmp)
if [ "$meter_idx" = "" ] ; then
          echo "No slimme meter">> /tmp/last_p1
else
          curl --silent  "http://$DOMOTICZ/json.htm?type=command&param=udevice&idx=$meter_idx&nvalue=0&svalue=$active_power_w" > /dev/null
          echo "$active_power_w" >> /tmp/last_p1
fi

meter_idx=$(sed -n ’s/ active_power//p’ $tmp)
if [ "$meter_idx" = "" ] ; then
          echo "No slimme meter">> /tmp/last_p1
else
          curl --silent  "http://$DOMOTICZ/json.htm?type=command&param=udevice&idx=$meter_idx&nvalue=0&svalue=$active_power_w" > /dev/null
          echo "$active_power_w" >> /tmp/last_p1
fi

meter_idx=$(sed -n ’s/ slimme_meter_gas//p’ $tmp)
if [ "$meter_idx" = "" ] ; then
          echo "No gas meter">> /tmp/last_p1
else
          curl --silent  "http://$DOMOTICZ/json.htm?type=command&param=udevice&idx=$meter_idx&nvalue=0&svalue=$total_gas_m3" > /dev/null
          echo "$total_gas_m3" >> /tmp/last_p1
fi

tail -1024 /tmp/last_p1 > $tmp
mv $tmp /tmp/last_p1

rm -f $report $tmp
```

The script runs every minute from `cron` with the following crontab:

```
#USER root@domoticz.home                                                                                #--p1.cron:root
#INSTALLEDFROM verlaine:src/domoticz                                                                    #--p1.cron:root
#  m  h  D   M  dow  cmd                                                                                #--p1.cron:root
   *  *  *   *   *   /bin/bash /usr/local/bin/p1_report > /tmp/last_p1_report.out 2> /tmp/last_p1_report.err  #--p1.cron:root
# end p1                                                                                                #--p1.cron:root
```

Contents