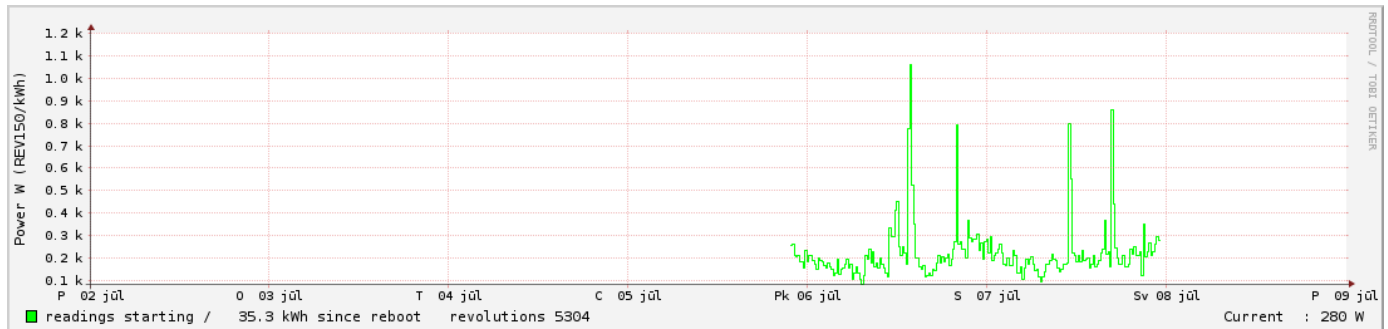


# Electricity meter optical readout for home automation Arduino as counter and graphing with Linux

This will guide you how to generate such graph and table:



```
UnixDate Date kWh_since_reboot kWh_daily
15526 2012_07_05 10.1 10.1
15527 2012_07_06 20.3 10.2
15528 2012_07_07 32.3 12.0
```

## Overview of existing solutions

**Commercial.** Wattcher by ELV.de

[http://www.elv.de/Funk-Sensor-Wechselstromz%C3%A4hler-ESA-1000-WZ/x.aspx/cid\\_726/detail\\_31320](http://www.elv.de/Funk-Sensor-Wechselstromz%C3%A4hler-ESA-1000-WZ/x.aspx/cid_726/detail_31320)

Wireless. It not battery operated and needs a power supply.

## IR LED and a photodiode with transistor amplifier

This was first idea that came to mind. Google found schematics example under name:

**Make your own IR obstacle detection sensor. Lets's make robots.**

It is not esthetic when attached to electric meter as there are many extra components on the circuit board. Needs some kind of a box.

## Commercial obstacle detectors using phototransistors

Connecting your electricity meter to Pachube:

<http://community.cosm.com/node/65>

Sensor: Focused reflective object sensor, **OPB702** from Optek sold by RS components

## CNY70

<http://www.strippenstrolch.de/1-2-12-der-reflexkoppler-cny70.html>

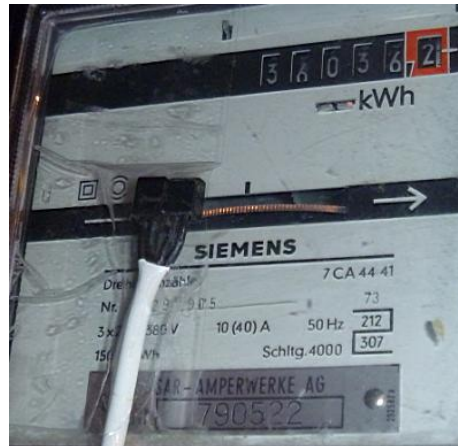
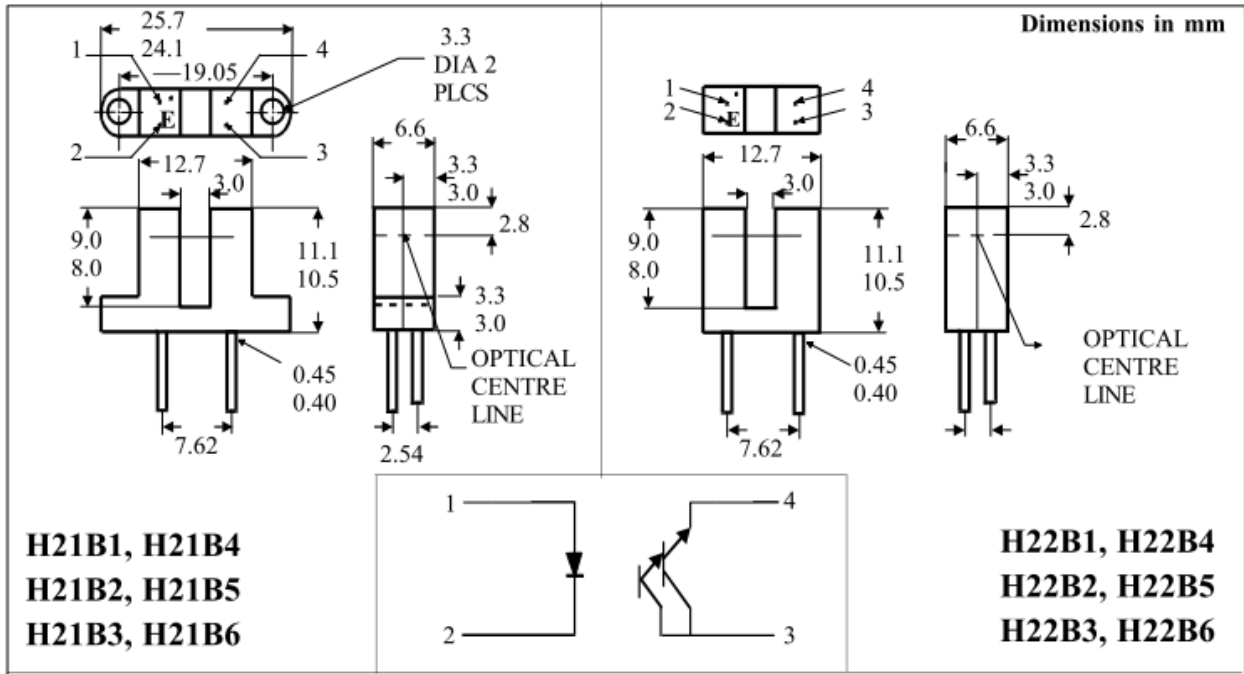
<http://nota-lan.de/wiki/index.php/Stromz%C3%A4hler-Sensor>

## H21B1

### 1mm APERTURE OPTO-ELECTRONIC SLOTTED INTERRUPTER SWITCH WITH DARLINGTON SENSOR

Best solution. It has a Darlington transistor that consists of two transistors after each other so gain is ca 100 x higher compared to CNY70. So that resistor on emitter can be 1k. Collectors pin is connected to 5V.

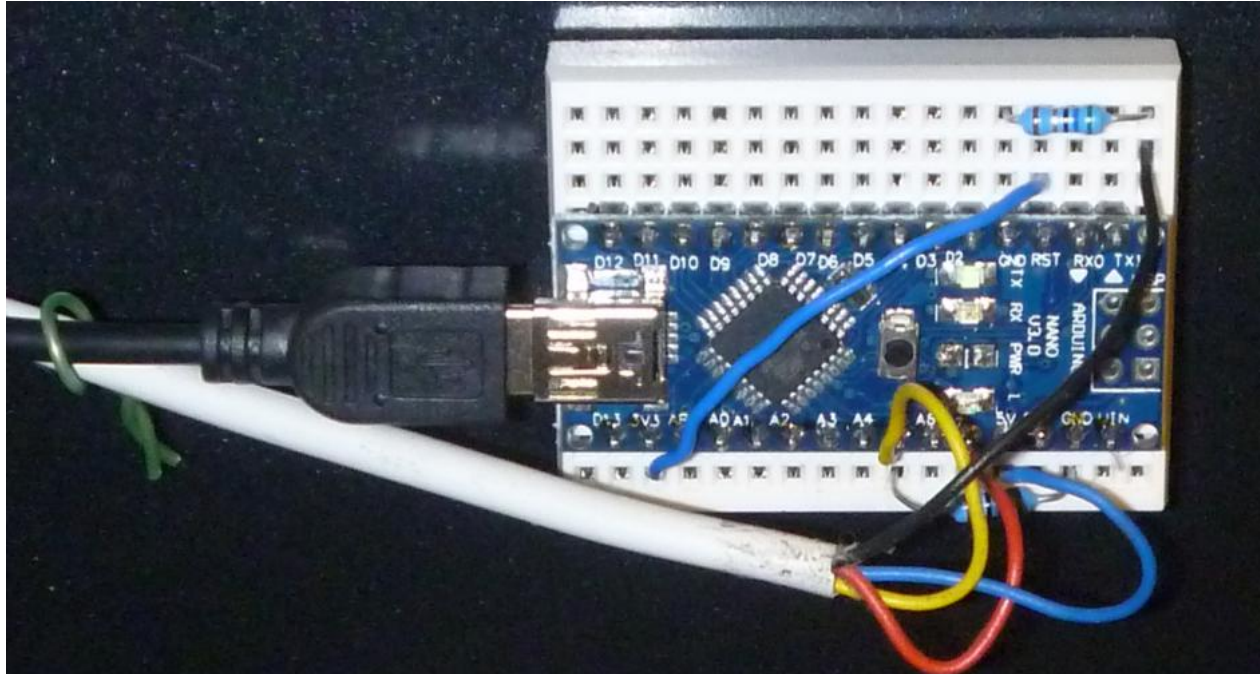
I ended up using H21B1. Original sensor is cut in half and glued together like in photo to direct light towards meter.



150 rev/kWh. Would be nice to use black cable instead of white. Fixed with transparent tape. It should look as simple as possible and nonmagnetic as electricity company might suspect that one tries to cheat the counter.

## Counter of revolutions

Arduino nano is more compatible with normal Arduino, but more compact. Costs only 12 USD on Ebay. Breadboard allows to connect more wires. Only two resistors are needed. 51 Ohm for LED current limit and 1 k to GND for Darlington.



```
-----  
// Arduino code.  
int c=0; // c = count revolutions  
int t=0; // t = trigger on edge  
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  if (analogRead(A5)<70) { delay(50); if (analogRead(A5)<70) { if (t==1) {t=0; c++;} } } // perform  
  debounce check if state changed and only then add count  
  
  if (analogRead(A5)>150) { delay(50); if (analogRead(A5)>150) { t=1; } }  
  
  Serial.print("databegin "); // name "databegin" is used by Linux to check that data line received correctly  
  Serial.print(analogRead(A5));  
  Serial.print(" ");  
  Serial.print(t);  
  Serial.print(" ");  
  Serial.print(c);  
  Serial.println(" dataend ");  
}
```

-----

## Ubuntu Linux bash code that reads Arduino via serial and makes graph for 1 week

```
#####3
cd /var/www/dataelectric
p="/dev/ttyUSB1"
ls $p
stty -F $p speed 9600 -echo

if [ ! -f db.rrd ]; then echo "creating rrd"
rrdtool create db.rrd --step 300 \
DS:a:COUNTER:600:U:U \
RRA:AVERAGE:0.5:1:2016 \
RRA:AVERAGE:0.5:12:10000 \
RRA:AVERAGE:0.5:144:3650
fi

while [ 1 ]; do

if [ -c $p ]; then
n=0

while true; do
a=$(head -1 $p)
echo $a
c=$(echo $a | awk -F " " '{ print $1; }')
echo $c

if [ $c = "databegin" ]; then
echo "begin received correctly, continuing"
break
else echo "data received wrong, continuing"
fi

n=$((n+1))
echo "try nr " $n
if [ $n -ge 10 ]; then break; fi
done

a=$(echo $a | awk -F " " '{ print $4; }')
echo $a" turns"
b=$(echo "scale=1; $a/150" | bc)
echo $b" kWh"
echo 150 rev/kWh 2.5rev/min~1 kW or 1 rev/min is 400W 400Wx300s = 12000
rrdtool update db.rrd N:$a;
#sleep 3
#rrdtool fetch db.rrd AVERAGE;

s=$(date -d-last-monday +%Y%m%d)
echo $s
e="start+7days"
echo $e

#--upper-limit 999 --lower-limit 0 --rigid \
rrdtool graph /var/www/tmp/electric.png --start $s --end $e --vertical-label "Power W (REV150/kWh)" \
-w 1000 -h 200 \
--x-grid DAY:1:DAY:1:DAY:1:0:%a" "%d" "%b \
DEF:aaa=db.rrd:a:AVERAGE \
CDEF:bbb=aaa,12000,* \
LINE1:bbb#00FF00:"readings starting "$THIS_YEAR"/"$THIS_MONTH" "$b" kWh since reboot revolutions "$a \
GPRINT:bbb:LAST:"Current \: %3.0lf W"

else echo "arduino not attached"
fi

echo "sleeping"
sleep 300

done
#####3
```

## Script executed daily and calculating how many kWh used per day

---

```
cd /var/www/dataelectric
p="/dev/ttyUSB1"
ls $p

if [ -c $p ]; then
n=0

while true; do
a=$(head -1 $p)
echo $a
c=$(echo $a |awk -F " " '{ print $1; }')
echo $c

if [ $c = "databegin" ]; then
echo "begin received correctly, continuing"
break
else echo "data received wrong, continuing"
fi

n=$((n+1))
echo "try nr " $n
if [ $n -ge 10 ]; then break; fi
done

a=$(echo $a |awk -F " " '{ print $4; }')
echo $a" turns"
b=$(echo "scale=1; $a/150" | bc)
echo $b" kWh"
#echo 150 rev/kWh 2.5rev/min~1 kW or 1 rev/min is 400W

d=$(date +%s)
d=$((d/86400))
e=$(date +%Y_%m_%d)
f=$(tail -1 /var/www/kwh.txt)
echo $f
set -- $f
g=$3
echo $g
h=$(echo "scale=1;$b-$g" | bc -l)

echo $d "$e" "$b" "$h"
echo $d "$e" "$b" "$h" " >> /var/www/kwh.txt

else echo "arduino not attached"
fi
```

---

## php script displaying text file with table

```
UnixDate Date kWh_since_reboot kWh_daily <br><pre>
<?php
$myFile = "kwh.txt";
$fh = fopen($myFile, 'r');
$data = fread($fh,20000);
fclose($fh);
echo $data;
?> <br></pre>
```

---

## **Wireless battery -operated optical sensor**

In future would be good to make battery operated 433 MHz wireless device. Like 432 MHz temperature sensors:

<http://www.instructables.com/id/Wireless-power-outlets-for-home-automation-using-A/>

LED could be on for a short time only to save battery. There is an example for bee counter.

<http://www.discovercircuits.com/H-Corner/beectr.htm>