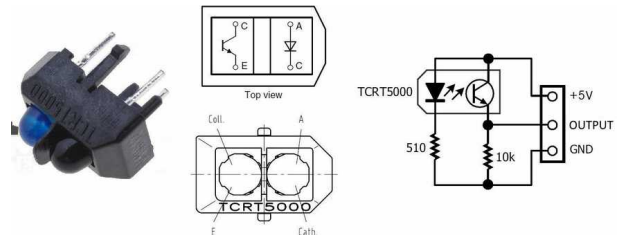


De pullup-weerstand wordt aan de ene zijde met +5VDC verbonden, de andere zijde in serie met de LDR naar GND gevoerd. Er is nu een weerstandspanningsdeler gemaakt, met op het knooppunt van de weerstand en LDR een lichtafhankelijke spanning, welke op één van de analoge ingangen wordt aangesloten. Plak de LDR met een stuk tape vast op de knipperled van de kWh-meter.

IR reflectiesensor. Telpulsen kunnen ook gemaakt worden met een infrarode (reflectie)sensor. Dit kan een oplossing zijn wanneer geen gebruik kan worden gemaakt van een telpuls-led of -uitgang, of de verbruikmeter blijkt voorzien te zijn van een IR-led (komt ook voor).

Hiernaast de infrarode reflectie-zender-opnemer TCRT5000, waarbij de lichte (blauwe) led de **zender (transceiver)** en de donkere 'led' de **ontvanger (receiver)** is. Hetzelfde is van toepassing bij losse leds; de lichte (soms blauwe) transparante diode ofwel led **straalt (emits)** niet zichtbaar IR-licht uit, welke door de donkere IR-photo-transistor(!) ontvangen zal worden. Onder ideale omstandigheden zijn twee weerstanden genoeg om een digitale puls te kunnen maken.



Het principe van de IR-lichtsluis, is dat gebundeld IR-licht naar een object-oppervlak wordt gericht, en het weerkaatste licht wordt gemeten. Afwisselend lichte en donkere vlakken van een bewegend object veroorzaken wel of géén reflectie van het IR-licht, die door een IR-fototransistor als **basisstroom** wordt gebruikt. De IR-led en IR-fototransistor zijn onder een zodanige hoek in de houder gemonteerd, zodat reflectie mogelijk is op een afstand van 0,2...15 mm. De stroom door de IR-led mag maximaal 60 mA zijn. De Uce is maximaal 70VDC bij een Ic van 100 mA bij 25°C(!).

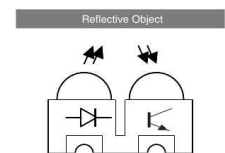
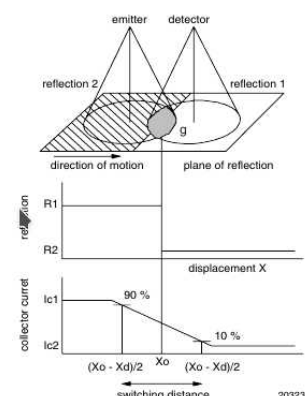


Foto-transistoren verouderen door inwerking van fotonen. De 70% sensorlevensduur bij continu-bedrijf is ca 5 jaar. Optische rookmelders moeten na 10 jaar vervangen worden. Een nieuwe televisie reageert al op de afstandsbediening weerkaatst via een witte muur. Na een aantal jaren lukt dit niet meer en moet de afstandsbediening rechtstreeks op het toestel worden gericht. Anders gezegd; een nieuwe IR-sensor zal in het begin regelmatig bijgesteld moeten worden; maar des te ouder te sensor, hoe minder vaak dit nodig zal zijn.

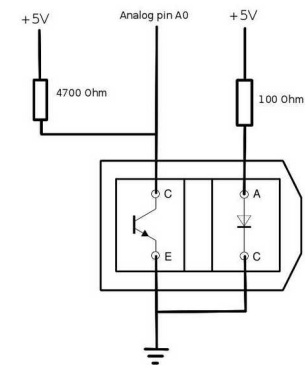
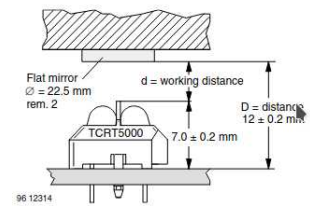
Hiernaast, uit de datasheet van Vishay een afbeelding van hoe de lichtsluis een reflectie-overgang waarneemt, en het bijbehorend schakelgedrag. Elke reflectie-overgang geeft een analoog signaal, terwijl een digitaal signaal gewenst is. InfraRood is warmtestraling, zodat de omgevingstemperatuur grote invloed op de schakeldrempel heeft. Bij 60°C is Uce terug gelopen naar maximaal 20 VDC bij een $I_{c_{max}}$ van 20 mA.

In verhouding tot de gevoeligheid van de IR-fototransistor is de IR-straling vrij fors. Zelfs de lichtsluishouder voorkomt niet dat er direct IR licht op de transistor valt (overspraak). Een ander probleem is ongewenste omgevingsreflectie, veroorzaakt door glas, spiegelende of licht gekleurde oppervlakken, enz. Zonlicht, daglicht, maar ook lampen en gekleurde leds, stralen IR-licht uit, welke de opnemer kunnen verstoren. Om een ongewenste basisstroom te voorkomen, is een gesloten of lichtafgeschermd opnemer-omgeving eigenlijk een vereiste.



De belangrijkste schakelfactor is de IR-lichtsterkte welke door de fototransistor wordt ontvangen. De IR-fotonen veroorzaken een basisstroom. Wanneer deze te groot is (te veel IR-licht), raakt de transistor verzadigd en zal de schakeltijd van de neergande flank sterk oplopen. Bij verzadiging is het schakelgedrag verstoord, wordt er ongewenste warmte ontwikkeld, en geeft een extra snellere veroudering. Om IR-led-overstraling te voorkomen zou deze regelbaar moeten zijn.

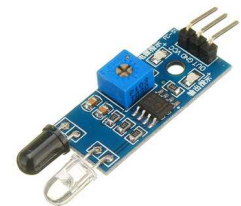
IR-licht is voor het menselijk oog onzichtbaar. De werking van een IR-led kan gecontroleerd worden, door naar de led te kijken via een digitale fotocamera of webcam.



Analoge IR-sensor. Een zeer eenvoudige meetoplossing is hiernaast weergegeven, waarbij de collector van de foto-transistor gebruik maakt van een analoge ingang. Met de pullup-weerstand van 4k7 is de analoge ingang maximaal hoog. De ingang krijgt een lagere waarde op het moment dat de sensor een reflectie waarneemt. Het resultaat is een redelijk stabiele waarde op de ingang, afhankelijk van de gemeten reflectie ergens tussen 0 en 255.

Bij de analoge meetmethode moet de software aangepast worden naar de gemeten analoge waarden, ofwel dit is een typische 'programmeurs-oplossing'.

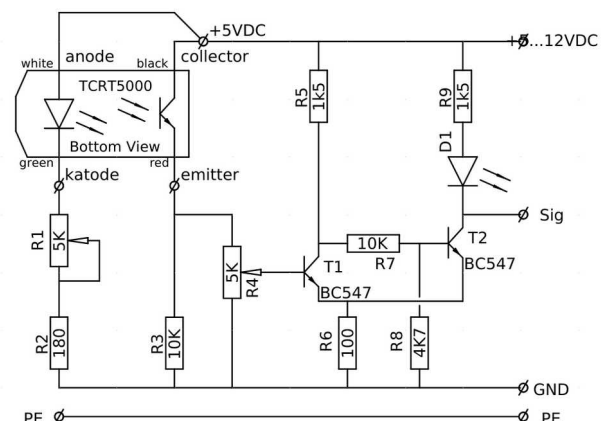
IR-sensor en Opamp. Een standaard IR-lichtsluis-module is meestal opgebouwd rond een opamp (393, 358, 741, oid), en geeft een digitaal signaal af. De IR-led straalt maximaal. Met een instelpotmeter kan de transistor-schakeldrempel worden ingesteld. IR-sensormodules hebben over het algemeen drie aansluitingen; +5V, GND, en de sensoruitgang. Soms is als extra een analoge signaaluitgang beschikbaar.



In veel gevallen (en wanneer zelf de reflectievlakken aangebracht kunnen worden), is dit een goedkope en prima oplossing. De overspraak bij de module hiernaast is zonder afscherming enorm groot, zodat daarvoor een constructieve oplossing moet worden bedacht. Door de niet regelbare IR-straling is de veroudering ook fors, en zal de instelling telkens gaan verlopen.

IR-opnemer en Schmitt-trigger. Om de IR-sensor ook voor de meer kritische telwerk-decimaal-cijfer-detectie te kunnen gebruiken, moet zowel de IR-lichtsterkte als de detectie-schakeldrempel instelbaar zijn, en dat bij een V_{cc} van 5...12 VDC.

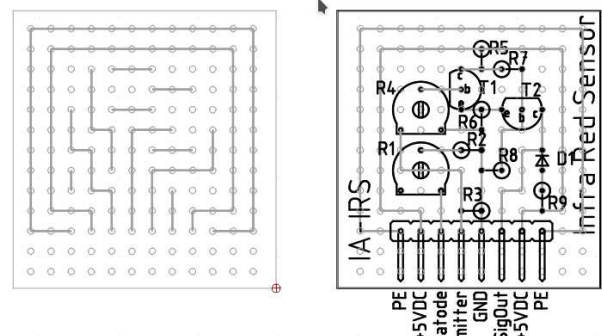
In het schema hiernaast wordt potmeter R1 gebruikt voor het instellen van de IR-lichtsterkte. In feite wordt deze potmeter gebruikt om (via een fotonen-koppeling) de basisschakelstroom voor de foto-transistor in te stellen. R2 voorkomt dat de IR-led met een te grote stroom belast kan worden en doorbrandt. De fototransistor zal (zoals uit de datasheet blijkt) bij een IR-reflectie-overgang met een zekere vertraging langzaam gaan geleiden, danwel langzaam gaan afknijpen.



Via de instelbare spanningsdelers R4 met daarop het foto-transistorsignaal, wordt schakeltransistor T1 aangestuurd. T1 en T2 vormen samen een Schmitt-trigger, zodat de schuine schakelflanken van de foto-transistor netjes omgevormd worden tot een eenduidig digitaal signaal.

LED D1 kan eventueel weggelaten worden (wel R9 met de collector verbinden). In de schakeling wordt ze gebruikt als 'power indicator', die Uit gaat wanneer de ingang Hoog wordt, ofwel de uitgang van de Schmitt-trigger is Laag.

De schakeling kan op een experimenteerprintje als 'meetvoorzet' worden gebouwd.

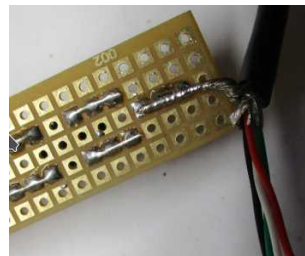


IR-Opnemer. In de praktijk is het uitrusten en lichtdicht monteren van de IR-opnemer een lastige klus. Hoewel Duct Tape handig is, is een meer solide constructie voor de opnemer een betere oplossing. Hoe, en op welke wijze is afhankelijk van de kennis, ervaring en inspiratie van de bouwer.

In dit voorbeeld worden een paar stukjes vlak en recht gevijld experimenteer-printplaatjes gebruikt. De IR-opnemer wordt zo haaks en recht mogelijk midden op de 'bodem' gemonteerd. De aansluitpennen worden aan de koperzijde plat omgebogen en voor de stevigheid over drie soldeereilanden vastgesoldeerd.

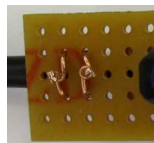
Elke 'zijwand' wordt voorzien van een viertal koperdraadjes, welke gebruikt gaan worden om de 'zijwanden' op de 'bodem' te monteren.

De afscherming van een vierdraads kabeltje (oude USB-printerkabel) wordt aan de bodem vastgesoldeerd. Let op; de kabel-afscherming mag bij de sensor elektrisch nergens mee verbonden worden! Dit wordt 'afisoleren' genoemd.

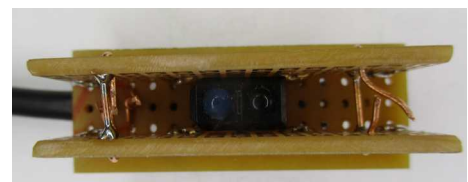
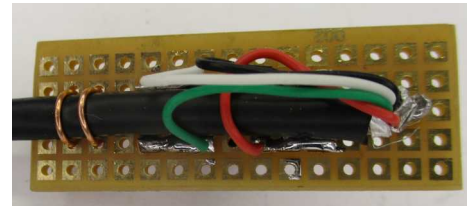
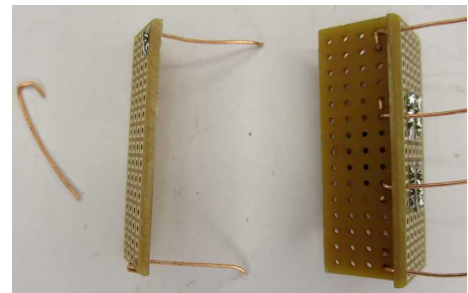
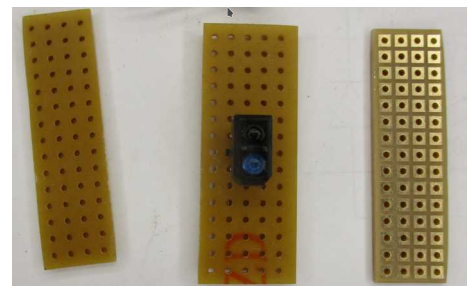


Verbuig de gesoldeerde afscherming en leg de kabel er teruglopend over heen; de gesoldeerde afscherming wordt nu alleen gebruikt als 'trekonlasting'.

Nu kunnen de aders aan de opnemerpennen gesoldeerd worden. Alle vier aansluitpennen worden 'volbedraad', ofwel elk met één ader verbonden. Eventuele doorverbindingen kunnen bij het aansluiten van de kabel worden gemaakt. De kabel wordt met twee koperdraadjes aan de 'bodem' vast 'getwist'.

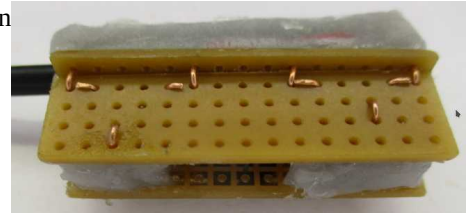


Nu kunnen de 'zijwanden' worden geplaatst en vastgesoldeerd. Als laatste worden met enkele koperdraadjes en soldeer de 'zijwandjes' voor de totale stevigheid onderling met elkaar 'verankerd'. De uitstekende koperdraadjes worden door een ander gaatje teruggebogen.



Nadat de sensor met met **afzetband (masking tape)** afgeplakt is, is de rest van de holle ruimte met donkere siliconenkit opgevuld. (tape-plakzijde tegen de siliconen).

Op de 'bodem-aansluitzijde' zijn de blootliggende soldeerverbindingen met siliconenkit afgedekt en dus elektrisch geïsoleerd. Na droging van de siliconen kan de tape gemakkelijk verwijderd worden.



Omdat IR-opnemers verouderen zullen deze gemakkelijk vervangen moeten kunnen worden. Wanneer de kabel voorzien wordt van een volbedrade vijfpolige stekker, ontstaat een 'universeel opnemer'; het vervangen is dan een kwestie van een nieuwe opnemer inprikken. Een eenvoudige goedkope oplossing is het gebruik van een vijfpolige DIN-stekker.



Opmerking; om storing te voorkomen mag de kabel niet te lang worden gemaakt. Ca 70 cm is toch wel het maximum.

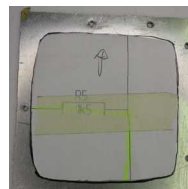
Ferrarismeter. De Ferraris-meter is de ouderswetse draaischijf-kwh-meter. De draaischijf is voorzien van een markeerstreepje, welke als pulsgever gebruikt kan worden.

Op de meter wordt een PulsWaarde genoemd als $P_w = 375$, of als 375 Umdr/1kWh. Dit betekent dat na 375 omwentelingen (of Umdr) 1kWh verbruikt is.



IR-Opnemer op Ferrarismeter. De kWh-meter is van het energiebedrijf en mag niet geopend of aangepast worden. De IR-opnemer moet exact uitgelijnd boven de dunne ferrarischijf worden 'gemonteerd', en zoveel mogelijk tegen omgevingslicht worden afgeschermd.

Van papier is een mal van de glasopening gemaakt, welke iets groter uit een afvalstukje aluminium of pertinax (HPL) is overgetekend. Na het uitzagen moet dit 'meetplaatje' zodanig bijgevijld worden, zodat deze zonder speling (dus bewegingsvrij!) op de glasplaat kan worden gelegd.



Rondom het glas is tijdelijk afzetband aangebracht. Met een lineaal is de horizontale en verticale positie van het meetpunt gemarkeerd. Op dezelfde manier worden eventuele andere openingen gemarkeerd.

Plaats de meetplaat op het glas en gebruik de randmarkeringen om de gewenste openingen aan te geven.

Na voorboren kunnen de gewenste openingen met een figuurzaag worden uitgezaagd.



De achterkant van de meetplaat is matzwart gemaakt om ongewenste reflectie zoveel mogelijk te voorkomen.

Om de IR-opnemer gemakkelijk te kunnen positioneren, is op de voorkant de meetpunt-locatie met een ingekraste lijnen aangegeven.



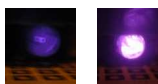
Wanneer de opnemer goed uitgelijnd is, kan deze met een rupsje smeltlijm worden vastgezet.

Als laatste wordt Duct Tape gebruikt om de meetplaat op de Ferraris-meter vast te zetten. De opnemer staat nu exact uitgelijnd voor de ferrarus-schijf en de kWh-stand kunnen we nog steeds uitlezen.

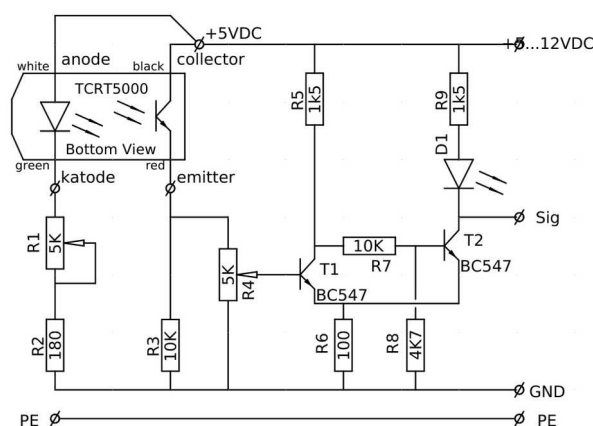


Afregelen IR-sensor. Het afregelen van de Schmitt-trigger vraagt enige geduld en beleid. Sluit een grote stroomverbruiker aan, zoals een stofzuiger, zodat de Ferraris-schijf draait. Tel het aantal seconden tussen twee omwentelingen; dit is de wachttijd tussen twee pulsen.

Begin met het in stappen omhoogdraaien (R1) van de IR-straling naar maximaal. Probeer tussen deze stappen met R4 heen-en-weer draaiend de Schmitt-trigger te laten schakelen.



*De ferrarisschijf is reflecterend en de markering niet. De trigger-ingang is dus normaal Hoog (!) en wordt door de markering Laag (!). De Schmitt-trigger-uitgang is dus normaal **Hoog** en tijdens een puls **Laag**, ofwel de Led is normaal **Uit** en tijdens een puls **Aan**.*



Geduld! Verlaag de IR-led stroom een beetje wanneer de IR-transistor (niet) schakelt, en probeer opnieuw met R4 de IR-transistor te laten schakelen.

Verlaag zo in stapjes naar de laagst mogelijke IR-led stroom totdat de kritische onderwaarde bereikt is waarop de ingang niet meer triggert. Stel R1 dan 10% hoger in, en regel R4 nu definitief af. Het stroomverbruik is nu minimaal met een stabiele schakeldrempel en de snelste flanktijden.

IA2109EC mk1 . Hieronder het basisprogramma voor een verbruikteller. Het grote verschil met een 'normale teller' van de voorgaande sketches, is dat er twee verschillende tellers worden gebruikt, namelijk PulseCounter voor het tellen van de meetpulsen, en EnergyCounter voor het bijhouden van de totaalstand in kWh, ofwel de 'meterstand'.

```
// IA2109EC mk1 Energy_Counter, Harm J Seef, The Netherlands
// 2021 okt 15 HJS - Designed.
// 2021 nov 15 HJS - Altered for use with 2109AK-shield
//
// =====
#include <LiquidCrystal.h> // Add library
const int rs=12, en=11, d4=8, d5=7, d6=5, d7=4; // Set vars for IA2109 interface pinning
LiquidCrystal lcd(rs, en, d4, d5, d6, d7); // Activate LiquidCrystal
//
const int PulseUp = 2; // up pulse at D2, Pinheader 20
int UpState = 0; // normal logic
int PrevState = 0; // reset controlvar
int PulseCounter = 0; // reset pulse counter
char CharValue[17] = "0123456789012345"; // char-array for displaying value as characters
//
// ---meter settings----- //
```

```

long int EnergyCounter = 1234;           // preset energy counter
int EnergyDecimal = 0;                   // number of kWh-decimals
char LcdUnit[] = "kWh";                  // "kWh" or "m3" or "litr" or ....
int PulseCountMax = 4;                   // pulses per kWh

void setup()                             // Initializing...
{ pinMode(PulseUp,INPUT);                // designate counter input
  lcd.begin(16, 2);                       // Init LCD...
  lcd.clear();                            // Reset LCD memory
  lcd.setCursor(0,0);                     // First line first position
  lcd.print(" IA2109EC mk1");             // Show sketch name...
  lcd.setCursor(0,1);                     //
  lcd.print(" Energy Counter  ");         //
  delay(2000);                             //
  lcd.setCursor(0,0);                     //
  lcd.print(" Energy Counter  ");         //
}

void loop()                              // Main program ...
{ UpState = digitalRead(PulseUp);          // If a pulse detected, UpState is HIGH:
  if (UpState == HIGH && PrevState == 0)    // if upcounting pulse detected
  { PulseCounter++;                         // add 1 to pulscounter
    PrevState = 1;                         // set control var
  }
  else                                     // other inputstate
  if(UpState == LOW && PrevState == 1)      // if input low again
  { PrevState = 0; }                      // reset control var
  //else                                  // do nothing (ignore pulse counter)
  // Endif
  if (PulseCounter >= PulseCountMax)       // If counted pulses reached maximum
  { EnergyCounter ++;                     // add 1kWh
    PulseCounter = 0;                     // reset pulse counter
  }
  lcd.setCursor(0,1);                     // Second line first position
  dtostrf(EnergyCounter,12,               // transfoem Value as char-array, 12 positions
    EnergyDecimal,CharValue);             // right alligned, given decimals
  lcd.print(CharValue);                   // Show total value, right alligned,
  lcd.print(" ");                         //
  lcd.print(LcdUnit);                     // show units
  delay(1);                               //
}

```

kWh of Watt Bij toepassing voor huishoudelijk gebruik is het wel zo handig dat de kWh-meter ook de decimalen en het actuele energieverbruik in Watt weergeeft. Bedenk daarbij wel dat 1 kWh totaal iets anders is dan 1000 Watt, immers 1 kilo-watt-*uur* is de hoeveelheid energie (Joules) die een 1000 Watt belasting in één uur verbruikt. De belasting ofwel actueel energieverbruik is in **Watt per seconde**, en het afrekenen met het energiebedrijf gaat in zogeheten **kilowatt-uren**; de ‘kilo’ als ‘duizend-teleenheid’ voor Uren en niet voor Watts!

Voorbeeld; een lamp van 100 Watt verbruikt 100 Joule per seconde, ofwel 6000 Joule per minuut, ofwel 360.000 Joule per uur, ofwel 3.600.000 Joule per 10 uur, wat hetzelfde is als 1 kWh. Samengevat; 1 kWh = 3.600.000 Joule.



Watt-meter Het is zeer verleidelijk om aan de hand van de gewijzigde tellerstand het huidige actueel verbruik terug te rekenen.

In formulevorm:

$$1kWh = n * Pw \quad // Pw = \text{Pulswaarde; aantal telpusen voor 1 kWh}$$

Het huidige momentele energieverbruik kan terugberekend worden als;

$$kW = \frac{kWh}{h} \quad // \text{ in Kilo Watts per seconde}$$

Het huidige momentele energieverbruik in Watt (VoltAmpere) kan herleid worden als;

$$W = \frac{\frac{1}{Pw}}{\left(\frac{Pw}{360000}\right) * 1000} \quad // \text{ in Watt (VA) per seconde}$$

Hetzelfde maar nu in een kortere formule;

$$W = \frac{3600000}{(n \times P_w) \times 1000}$$

Kleinste meetwaarde (resolutie) Een kWh-teller is beslist géén Watt-meter. De eerste hindernis is het enorme verschil in 'kleinste meeteenheid'. De 'meetbaarheid' van het actuele energieverbruik is afhankelijk van de meter-puls waarde. Bij een $P_w=375$ is de kleinste meetbare verbruikwaarde $1/375$ kWh, ofwel 0,0026667 kWh. Bij terugrekenen van Kilo- naar Watt-uur is dit dus 0,000002667 Wh. De Watt-uren moeten nog gedeeld worden door 3600 seconden, zodat we voor de teleenheid voor het Watt-verbruik in de wetenschappelijke notaties terecht komen. Dit mogen we de Arduino niet aan doen.

Grootste meettijd (meetsnelheid en Time-out) In de huiselijke omgeving is 1000 Watt al een behoorlijke belasting. Bij een belasting van 1000 Watt en $P_w=375$ wordt er elke 0,0026667 kWh een meetbare puls opgewekt.

Wanneer we de meter-puls waarde kennen, kan het actuele verbruik herleid worden uit de periodetijd tussen twee pulsen (PulseTime), ofwel twee opgaande flanken. Hiervoor wordt de sketch uitgebreid met een periodestarttijd (PrevTime) en een periodestoptijd (CurrTime).

```
IF OpgaandeFlankGevonden          // wanneer opgaande puls gevonden
  CurrTime = millis();              // noteer de tijd
  PulseTime = CurrTime - PrevTime; // bereken de pulstijd ofwel periodeduur
  PrevTime = CurrTime;             // update tijdsnotatie voor volgende meting
  Watts = (3600000 / (Pw * PulseTime)) * 1000; // bereken het huidig verbruik in Watt
```

Bij 1000 Watt is de meettijd tussen twee pulsen dan in seconden. Bij een belasting van 100 Watt zal de meettijd 10 maal langer moeten zijn, voordat er een puls 'gemeten' zal worden. Bij een belasting van 10 Watt zal de meettijd 100 maal langer moeten zijn, voordat er een puls 'gemeten' zal worden.

Daar bovenop komt een praktisch probleem. Stel dat de meter een belasting meet van 1000 Watt die even later uitgeschakeld wordt. De laatste meting was 1000 Watt verbruik, maar omdat na het uitschakelen er geen nieuwe puls gemeten gaat worden, blijft de 'meting van huidig verbruik' op 1000 Watt staan, terwijl het in werkelijkheid dus nul Watt is.

Hoewel de meter 'nul' zou moeten aangeven wanneer er géén energie wordt verbruikt, is dit in de praktijk niet mogelijk. Als 'oplossing' wordt de meettijd beperkt tot een bepaald tijdvenster (time-out). Wanneer de meting niet betrouwbaar is wordt het verbruik teruggemeld als kleiner dan 1, ofwel als " < 1 Watt".

IA2109EC mk2 als kWh-meter . In de navolgende sketch zijn een kWh-decimaaltelling EnergyRemainder en het huidige energieverbruik WattNow toegevoegd.

Eerst wordt met digitalRead() het huidig niveau van de pulsingang gelezen.

Alleen bij de opgaande pulsflank (Upstate = High) wordt teller PulseCounter met één verhoogd. Tegelijkertijd wordt variabele PrevState als opgaande-flank-merker één gemaakt ten teken dat de opgaande puls geteld is. Bij een volgende programmadoorgang is PrevState geen nul, en zal dezelfde opgaande flank niet weer opnieuw geteld worden.

De kWh-decimaalteller EnergyRemainder wordt bij elke puls met de verbruikte energie verhoogd.

```
void loop() {
  UpState = digitalRead(PulseUp);
  if (UpState == HIGH && PrevState == 0)
  {
    PulseCounter++;
    PrevState = 1;
    EnergyRemainder = EnergyRemainder +
      (1.0 / PulseCountMax);
  }
}
```

Voor ‘nul-watt-verbruik’ wordt controle-variabele WattTimeOut gereset. Voor het herleiden van de puls-periodesduur PulseTime wordt gebruik gemaakt van CurrTime en van PrevTime. Bij elke pulsdetectie ofwel opgaande flank wordt het huidige energieverbruik WattNow berekend.

Bij elke programma-doorgang (cyclus) wordt de verstreken tijd sinds de laatste opgaande flank berekend. Wanneer deze groter is dan de maximaal opgegeven meettijd PulseTimeOut in seconden, wordt controlevariabele WattTimeOut waar, en wordt het energieverbruik in Watts op één Watt gezet.

Alleen wanneer de telpuls weer Laag geworden is (UpState = Low), en de opgaande-flank-merker nog één is, zal de opgaande-flank-merker PrevState weer nul gemaakt worden.

Wanneer pulscounter PulseCounter het maximale aantal pulsen geteld heeft (de meter-pulswaarde Pw, ofwel PulseCountMax), wordt de totaal teller EnergyCounter met één verhoogd, en wordt de pulsteller weer op nul gezet. Omdat de waarde van EnergyRemainder puur cosmetisch is, wordt ook deze genuld. De puls-telcyclus begint nu weer van voren af aan.

Als laatste wordt het display geactualiseerd. Op de eerste regel wordt het huidig verbruik in Watts weergegeven. Wanneer deze meting door gebrek aan meetpulsen uit de maximale meettijd loopt is controle-variabele WattTimeOut waar, en wordt de meetwaarde op het display vooraf gegaan door het ‘is kleiner dan’-symbool.

Op de tweede regel wordt de huidige kWh-tellerstand in de gegeven aantal decimalen weergegeven.

```
// IA2109EC mk2 Energy_Counter, Harm J Seef, The Netherlands
// 2021 okt 15 HJS - Designed.
// 2021 nov 15 HJS - Altered for use with 2109AK-shield
// - kWh-remainder added
// - Watt added
//
// =====
#include <LiquidCrystal.h> // Add library
const int rs=12, en=11, d4=8, d5=7, d6=5, d7=4; // Set vars for IA2109 interface pinning
LiquidCrystal lcd(rs, en, d4, d5, d6, d7); // Activate LiquidCrystal
//
const int PulseUp = 2; // up pulse at D2, Pinheader 20
int UpState = 0; // normal logic
int PrevState = 0; // reset controlvar
// ---meter settings-----
float EnergyCounter = 23157; // preset energy counter
int EnergyDecimal = 2; // number of kWh- / m3decimals
float EnergyRemainder = 0; // reset remainder counter
int PulseCounter = 0; // reset pulse counter
int PulseCountMax = 4; // 100; // pulses per unit (kwh / m3 / see meter specs)
//
unsigned int UseNow = 0; // reset current used energy var
float PulseTime = 0; // reset period time
float CurrTime = 0; // reset time marker
float PrevTime = 0; // ditto
int PulseTimeOut = 20; // measurement time out in seconds
bool UseTimeOut = false; // time out control var
// ---lcd-vars-----
float DisplayValue = 0; // reset temp display value
char CharValue[17] = "0123456789012345"; // char-array for displaying value as characters
char LcdUnit1[] = " kWh"; // kWh m3
char LcdUnit2[] = " W"; // W Lpm
//
void setup() // Initializing...
{ pinMode(PulseUp, INPUT); // designate counter input
  lcd.begin(16, 2); // init LCD...
  lcd.clear(); // reset LCD memory
  lcd.setCursor(0,0); // first line first position
  lcd.print(" IA2109EC mk2"); // show sketch name...
  lcd.setCursor(0,1);
  lcd.print(" Energy Counter ");
  delay(2000);
  lcd.clear();
}

void loop() // Main program ...
{ UpState = digitalRead(PulseUp); // update UpState
  if (UpState == HIGH && PrevState == 0) // if pulse detected
  { PulseCounter++; // add 1 to pulscounter
    PrevState = 1; // set control var
    EnergyRemainder = EnergyRemainder + // add used energy
      (1.0 / PulseCountMax); // as 'unit'/decimal value
    UseTimeOut = false; // reset time out var
    CurrTime = millis(); // update current timestamp
```

```
WattTimeOut = false;
CurrTime = millis();
PulseTime = CurrTime - PrevTime;
PrevTime = CurrTime;
WattNow = (3600000 /
  (PulseCountMax * PulseTime)) * 1000 ;
```

```
else
  if ((millis()-PrevTime) >=
    (PulseTimeOut * 1000))
  {
    WattTimeOut = true;
    WattNow = 1;
  }
  if (UpState == LOW && PrevState == 1)
  {
    PrevState = 0;
  }
// Endif
```

```
if (PulseCounter >= PulseCountMax)
{
  EnergyCounter ++;
  EnergyRemainder = 0;
  PulseCounter = 0;
}
```

```
lcd.setCursor(0,0);
dtostrf(WattNow,12,0,CharValue);
```

```
lcd.print(CharValue);
lcd.print(" W ");
if (WattTimeOut == true)
{
  lcd.setCursor(9,0);
  lcd.print("<");
  delay(1);
}
```

```
lcd.setCursor(0,1);
DisplayValue = EnergyCounter + EnergyRemainder;
dtostrf(DisplayValue,12,
  EnergyDecimal,CharValue);
lcd.print(CharValue);
lcd.print(" ");
lcd.print(LcdUnit);
delay(1);
}
```

```

PulseTime = CurrTime - PrevTime;           // calculate pulse period time
PrevTime = CurrTime;                       // update previous timestamp
if (LcdUnit1 == " kWh")                   // when used as kWh
{
    UseNow = (3600000.0 /                 // calculate used energy, 3.600.000 Joule
        (PulseCountMax * PulseTime) * 1000) ; // 1000 ms per second
}
if (LcdUnit1 == " m3")                   // when used as m3
{
    UseNow = ((1000 / PulseCountMax) /    //
        PulseTime) * 6000 ;              // usage in litres per minute (Lpm)
}
else                                     // check inputstate (blocking pulse counter)
if ((millis()-PrevTime) >=              // if measurement
    (PulseTimeOut * 1000))              // timed out
{
    UseTimeOut = true;                  // set status
    UseNow = 1;                        // set lowest value
}
if (UpState == LOW && PrevState == 1)    // if input low again
{
    PrevState = 0; }                  // reset control var
// Endif
if (PulseCounter >= PulseCountMax)       // if counted pulses reached maximum
{
    EnergyCounter ++;                  // add 1kWh
    EnergyRemainder = 0;               // reset remainder counter
    PulseCounter = 0;                 // reset pulse counter
}
//
lcd.setCursor(0,0);                     // first line first position
dtostrf(UseNow,12,0,CharValue);         // transform Value as char-array, 12 positions
// right aligned, zero decimals
lcd.print(CharValue);                   // show total value, right aligned,
//
if (UseTimeOut == true)                 // if timed out (used energy too low (Not Zero!!))
{
    lcd.setCursor(9,0);                //
    lcd.print("<");                      //
    delay(1);                          //
}
//
lcd.setCursor(0,1);                     // second line first position
DisplayValue = EnergyCounter + EnergyRemainder; // update displayvalue
dtostrf(DisplayValue,12,                // transform Value as char-array, 12 positions
    EnergyDecimal,CharValue);           // right aligned, given decimals
lcd.print(CharValue);                   // show total value, right aligned,
//
lcd.print(LcdUnit1);                    //
delay(1);                               //
}
//

```

Opmerking; Sommige kWh-sketches op het internet berekenen de kwh-stand na elke puls weer opnieuw. Hierboven is dit niet het geval, maar wordt de processortijd zoveel mogelijk gebruikt voor het detecteren en verwerken van de telpuls.

Conclusie. Met de hier gegeven oplossing kan elke verbruikmeter met een puls gemonitort worden. Bedenk wel dat de stroomrichting voor de teller niets uitmaakt. Bij stroom terugleveren zal deze dus ook als verbruik worden geteld. In één van de volgende projecten wordt de stroommeting en diens richting behandeld.