

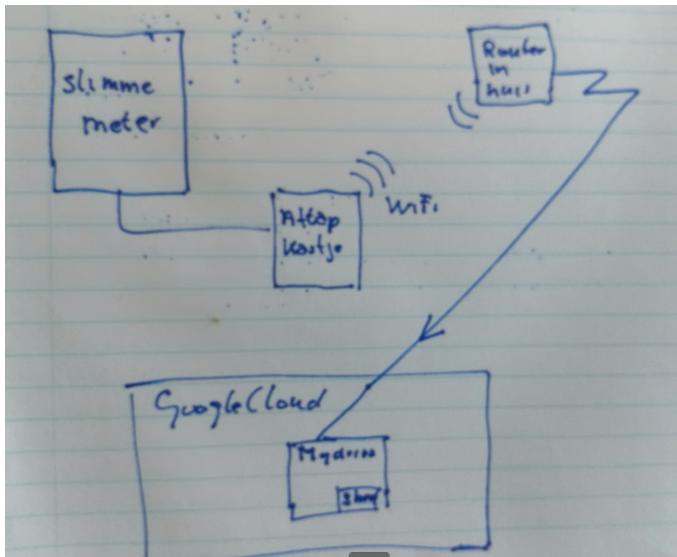
Table of Contents

Slimmemeter intro.....	2
Het aftapkastje aansluiten op het eigen netwerk.....	2
Nieuwe software downloaden.....	3
Hardware.....	4
Connector overzicht:.....	5
SoftwareSerial.....	6
Aanpak update Google sheet via URL.....	6
json suggestie:.....	6
Leesstof.....	6
Github software.....	6
Fingerprint.....	6
(Spread) Sheet op Google MyDrive.....	7
Javascript voor deze sheet.....	7
Sheet.....	10
Problemen en to do.....	10
Programma (versie 4) in de esp8266.....	11
Netwerk details (hoe knoop je “alles”aan elkaar).....	15
De ESP8266.....	16
Google web pagina.....	16
De Google sheet.....	16
Google heeft een complete ontwikkelomgeving obv Javascript om de cellen in het spreadsheet te bewerken.....	17
CRC 16 Theorie.....	17
CRC generators.....	18
Hoe werkt CRC in het esp8266 programma.....	18
Problemen CRC.....	19

Slimmemeter intro

Deze slimmemeter aftap zorgt er voor dat de gemeten slimmemeter standen gezet worden in een Google spreadsheet in the Google Drive (Cloud). In huis is er niet veel meer nodig dan een aftapkastje met verbinding met het internet.

Het aftapkastje neemt zelf contact op met de Google spreadsheet. Er is dus geen toegang vanuit Google naar het netwerk of het aftapkastje in huis. Kortom VEILIG.



Het Google account is:

laag.straat22@gmail.com

Het paswoord heeft te maken met de wandelclub op woensdag.

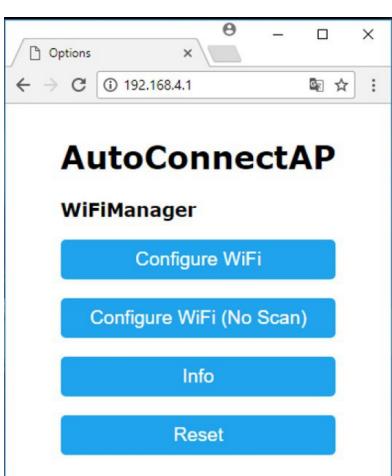
Een uitleg met meer informatie staat in het hoofdstuk: Netwerkdetails.

Het aftapkastje aansluiten op het eigen netwerk.

De WiFi manager is opensource en te vinden op het internet:

<https://github.com/tzapu/WiFiManager>

Deze wifi manager toont onderstaand scherm wanneer het niet lukt om verbinding te maken met je eigen wifi netwerk.



Het aftapkastje moet net als een pc bekend gemaakt worden bij het WiFi netwerk.

Als het kastje opstart wil het, in WiFi station mode, contact zoeken met het locale WiFi netwerk. Maar dat gaat niet lukken omdat hij, initieel, niet bekend is met dit netwerk.

Na enige tijd geeft het kastje het op en zet zich zelf in accesspoint mode met SSID SlimmePeter.

Stap 1: verbind een laptop met het WiFi netwerk SlimmePeter.

Stap 2: in een browser type in 192.168.4.1 (zie plaatje hiernaast)

Stap 3: Klik Configure WiFi

Stap 4: Kies je eigen WiFi netwerk

Stap 5: Vul je eigen WiFi paswoord in.

Stap 6: Wacht en huiver... als het goed is start het afstapkastje nu netjes op. Kijk naar de signaalsterkte (RSSI) het bereik tussen de -40 en – 70 is werkzaam.

Als je het netwerkkastje de volgende keer aanzet zal het automatisch op je eigen netwerk opstarten.

Nieuwe software downloaden

Nieuwe software (als die er al is) kan met een druk (5 seconden of meer) op de UpdateButton gedownload worden.

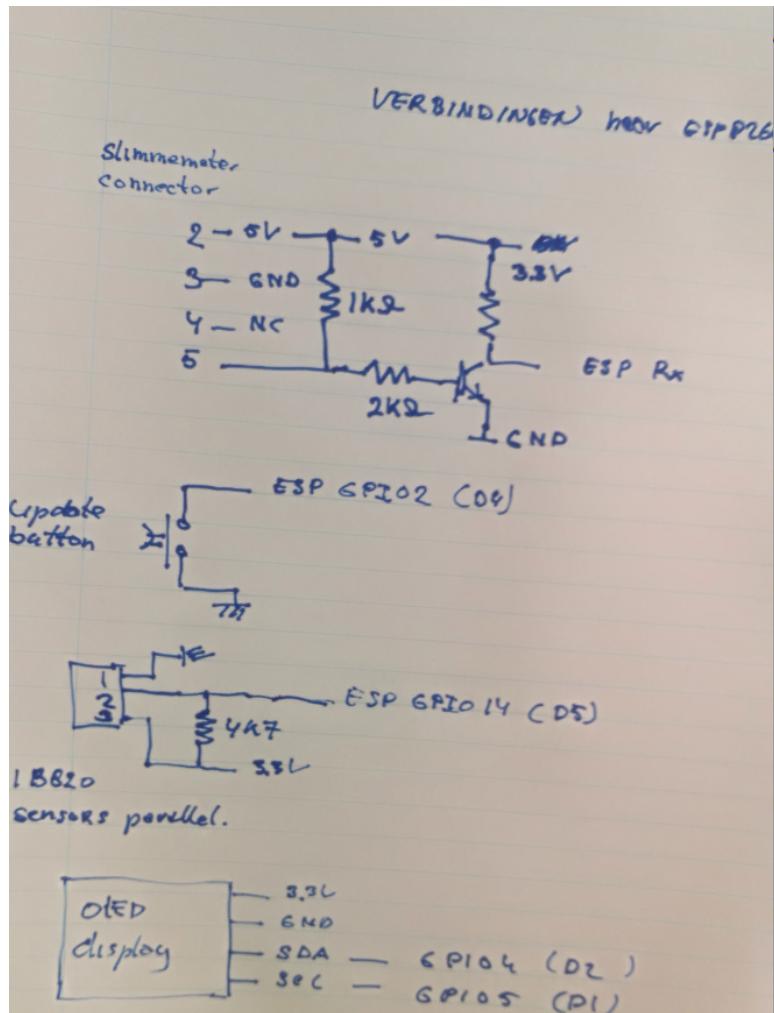
De nieuwe software staat op mijn nas.

http://sagittaweb.synology.me/ESP_bin/slommeter_piet.ino.bin

Na het downloaden start het kastje zelf weer op. Heb enkele minuten geduld.

Het opstart scherm toont de versie van de nieuwe software zodat je zeker weet dat de nieuwe software in hetkastje “zit”.

Hardware

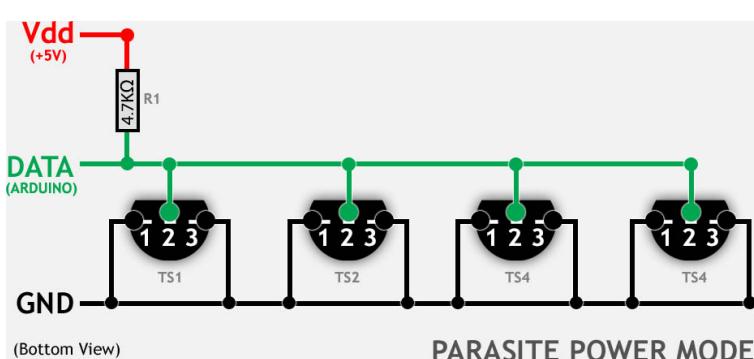


De slimmemeter connector is 6 polig. De 4 middelste aansluitingen (2,3 en 5) worden gebruikt. De NPN transistor inverseert het signaal van de slimmemeter. Als alternatief is een 74ls7404 inverter ingezet met op de ingang een pullup van 470 Ohm.

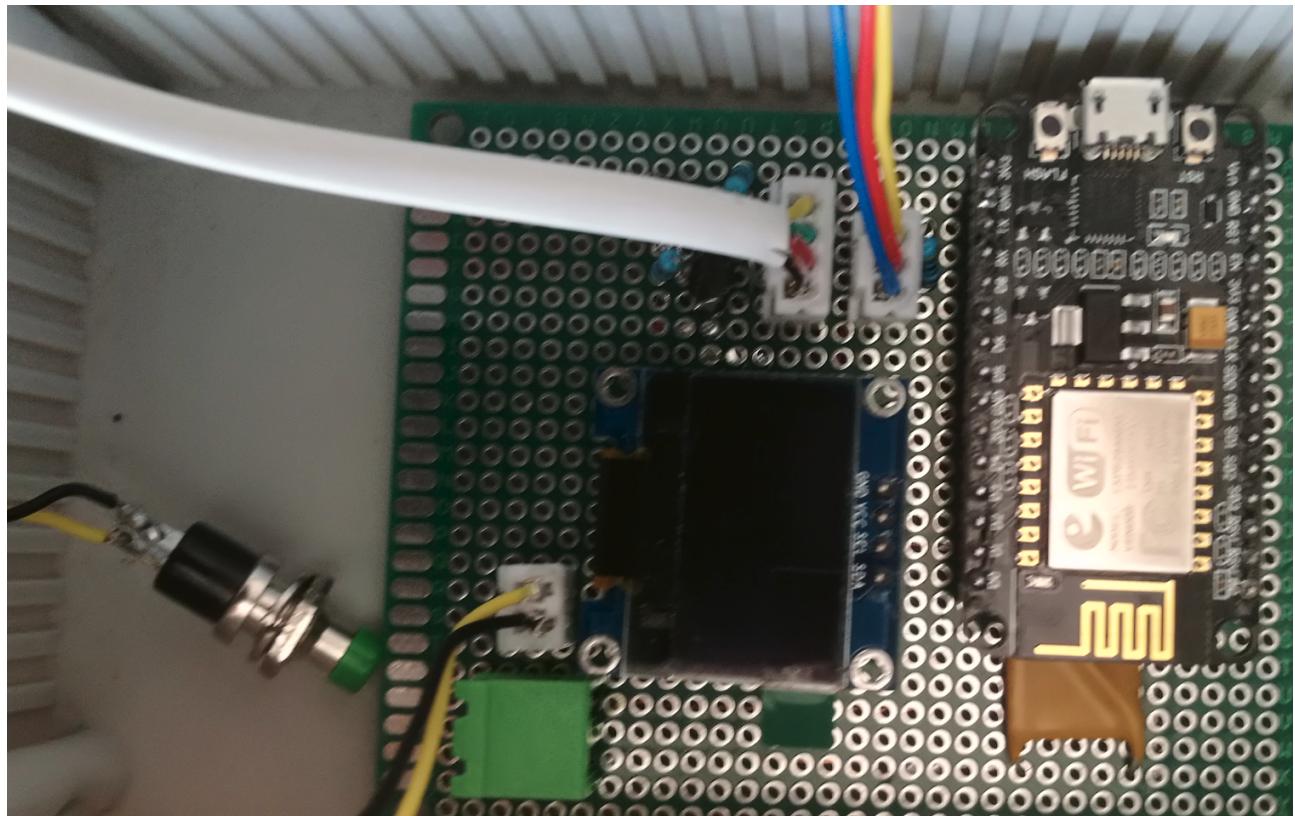
Deze knop dient om een nieuw image (software) vanaf de NAS die bij Jan Jansen staat te downloaden.

Voor de temperatuur sensors (1B820 van Dallas) zie details hieronder.

Het mini display maakt gebruik van de i2c bus.



De temperatuursensors worden parallel aangesloten. In de software wordt onderscheid gemaakt. Ik gebruik er twee met index 0 en 1.



Connector overzicht:

Middenboven links: de slimmemeter connector

Middenboven rechts de temperatuursensot connector

Linksonder boven: de Update Button

Linksonder onder: aansluiting van een stabiele DC bron tussen de 6 en 20V. De voeding is tegen verkeerom aansluiten (diode) beveiligd en er is een thermozekering aanwezig die uitgeschakelt bij overbelasting.

Onder de esp8266 zit een 24V naar 5V convertor. (NOOIT aan de instelschroef draaien.)

SoftwareSerial

De laatste versie van de esp8266 software draait met Softwareserial. Niet het allerbetrouwbaarste product maar met CRC er op om de transmissie te controleren werkt het, tot nu toe, betrouwbaar,

Aanpak update Google sheet via URL

<http://embedded-lab.com/blog/post-data-google-sheets-using-esp8266/>

json suggestie:

Sorry about the delay hengis. If you post your data in JSON format from ESP then it will be a lot easier to parse in GAS. So construct your ESP data in JSON format like this: {"tag1":"10", "tag2":"9", "tag3":"11"} then in GAS side you could something like this:

```
var espData = JSON.parse('{"tag1":"10", "tag2":"9", "tag3":"11"}')  
then to get the tag1 value, espData.tag1.
```

Leesstof

<http://domoticx.com/p1-poort-slimme-meter-hardware/>

Bovenstaande link toont een overzicht van de slimme meters op de Nederlandse markt.

Github software

<https://github.com/jantenhove/P1-Meter-ESP8266>

De Smartmeter software maakt gebruik van een programma op Github. Er zijn verschillende veranderingen aangebracht. Met name de Telegramgrootte was veel te klein en er waren ook problemen met CRC.

Fingerprint

De Google sites zijn allen https. ER is een fingerprint key nodig. Die kan op onderstaande manier achterhaald worden. Helaas verloopt deze key. Momenteel blijft de app werken maar op het display laat zien of de fingerprint verlopen is.

Onderstaand hoe de fingerprint te achterhalen.

```
jan@jan-jul-17:~$ echo | openssl s_client -connect script.google.com:443 |& openssl x509  
-fingerprint -noout  
SHA1 Fingerprint=97:E1:03:DA:DC:42:20:55:BE:8B:DA:F2:D3:B6:52:CE:4A:D1:B3:B8  
jan@jan-jul-17:~$
```

(Spread) Sheet op Google MyDrive

https://docs.google.com/spreadsheets/d/10pEYQDwuxi3Vtk4H5VScbd-Lu_bU6XaK_6FMPw8Ul4w/edit#gid=2146636906

Javascript voor deze sheet

Deze versie bevat de wijzigingen voor een automatisch 7 dagen en jaar overzicht.

```
function doGet(e){
  Logger.log("--- doGet ---");
  console.log ("do get started");

  var Tbinnen = "1", Tbuit = "1", mEVLT = "2", mEVHT = "1", mEOLT = "2", mEOHT = "1", mEAT = "1", mEAV = "2", mGAS = "1", CRC2 = "0", CRC3 = "0";

  try {

    Tbinnen = e.parameters.Tbinnen_val;
    Tbuit = e.parameters.Tbuit_val;
    mEVLT = e.parameters.mEVLT_val;
    mEVHT = e.parameters.mEVHT_val;
    mEOLT = e.parameters.mEOLT_val;
    mEOHT = e.parameters.mEOHT_val;
    mEAT = e.parameters.mEAT_val;
    mEAV = e.parameters.mEAV_val;
    mGAS = e.parameters.mGAS_val;
    CRC2 = e.parameters.CRC2_val;
    CRC3 = e.parameters.CRC3_val;

    console.log (mEVLT); console.log (mEVHT);

    // save the data to spreadsheet
    save_data( Tbinnen, Tbuit, mEVLT, mEVHT,mEOLT, mEOHT, mEAT, mEAV, mGAS, CRC2, CRC3);

    return ContentService.createTextOutput("Wrote:\n Tbinnen: " + Tbinnen + "\n Tbuit: " + Tbuit);

  } catch(error) {
    Logger.log(error);
    return ContentService.createTextOutput("oops.... " + error.message
      + "\n" + new Date()
      + "\ntag: " + Tbinnen +
      + "\nvalue: " + Tbuit);
  }
}

// Method to save given data to a sheet
function save_data(Tbinnen, Tbuit, mEVLT, mEVHT,mEOLT, mEOHT, mEAT, mEAV, mGAS ,CRC2, CRC3 ){
  Logger.log("--- save_data ---");

  try {
    var dateTIme = new Date();
    var hrs = dateTIme.getHours();
    var dag = dateTIme.getDay();

    // Paste the URL of the Google Sheets starting from https thru /edit
    // For e.g.: https://docs.google.com/..../edit
    var ss = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/17Rd1vln4Pz0dMnMCMzQ04Sd1nK82aKunuVM02VWW2m0/edit");
  }
}
```

```

//17Rd1vln4Pz0dMnMCMzQ04Sd1nK82aKunuVM02VWW2m0

var summarySheet = ss.getSheetByName("Summary");
var dataLoggerSheet = ss.getSheetByName("DataLogger");
var longSheet = ss.getSheetByName("Long");
var weekSortSheet = ss.getSheetByName("weekSort");
// Get last edited row from DataLogger sheet
var row = dataLoggerSheet.getLastRow() + 1;
var rowlong = longSheet.getLastRow() + 1;
var rowWeekSort = weekSortSheet.getLastRow() + 1;

// populate longSheet met een waarde per hieronder aangegeven tijd per dag.
if (summarySheet.getRange("B5").getValue() == 0) { // check already one value stored in longSheet
if(hrs == 07 || hrs== 15 || hrs== 23 || hrs== 10) { // hours to be stored
// Start Populating the logdata sheet
longSheet.getRange("A" + rowlong).setValue(rowlong - 1); // ID
longSheet.getRange("B" + rowlong).setValue(dateTime); // date
longSheet.getRange("D" + rowlong).setValue(parseFloat(Tbinnen)); // value
longSheet.getRange("E" + rowlong).setValue(parseFloat(Tbuit)); // value
longSheet.getRange("F" + rowlong).setValue(mEVLT); // value
longSheet.getRange("G" + rowlong).setValue(mEVHT); // value
longSheet.getRange("H" + rowlong).setValue(mEOLT); // value
longSheet.getRange("I" + rowlong).setValue(mEOHT); // value
longSheet.getRange("L" + rowlong).setValue(mGAS); // value
summarySheet.getRange("B5").setValue(1); // indicate "semaphore" stored dwz gedaan
};

};

if(hrs != 07 && hrs!= 15 && hrs!= 23 && hrs!= 10) {
summarySheet.getRange("B5").setValue(0); // free storage of longSheet values
}

//Start populating int weeksortsheet
var index = dag*24 + hrs+2; // bereken plaats in de sheet

weekSortSheet.getRange("B" + index).setValue(dateTime);
weekSortSheet.getRange("A" + index).setValue(index);
weekSortSheet.getRange("D" + index).setValue(parseFloat(Tbinnen));
weekSortSheet.getRange("E" + index).setValue(parseFloat(Tbuit));
weekSortSheet.getRange("F" + index).setValue(mEVLT);
weekSortSheet.getRange("G" + index).setValue(mEVHT);
weekSortSheet.getRange("H" + index).setValue(mEOLT);
weekSortSheet.getRange("I" + index).setValue(mEOHT);
weekSortSheet.getRange("J" + index).setValue(mEAV);
weekSortSheet.getRange("K" + index).setValue(mEAT);
weekSortSheet.getRange("L" + index).setValue(mGAS);

// Start Populating the DataLoggerSheet
if (summarySheet.getRange("B6").getValue() == 1){ // check
dataLoggerSheet.getRange("A" + row).setValue(row - 1); // ID
dataLoggerSheet.getRange("B" + row).setValue(dateTime); // date
dataLoggerSheet.getRange("D" + row).setValue(parseFloat(Tbinnen)); // value
dataLoggerSheet.getRange("E" + row).setValue(parseFloat(Tbuit)); // value
dataLoggerSheet.getRange("F" + row).setValue(mEVLT); // value
dataLoggerSheet.getRange("G" + row).setValue(mEVHT); // value
dataLoggerSheet.getRange("H" + row).setValue(mEOLT); // value
dataLoggerSheet.getRange("I" + row).setValue(mEOHT); // value
dataLoggerSheet.getRange("J" + row).setValue(mEAV); // value
dataLoggerSheet.getRange("K" + row).setValue(mEAT); // value
dataLoggerSheet.getRange("L" + row).setValue(mGAS); // value
dataLoggerSheet.getRange("M" + row).setValue(CRC2);
dataLoggerSheet.getRange("N" + row).setValue(CRC3);
}
// Update summary sheet
summarySheet.getRange("B1").setValue(dateTime); // Last modified date
summarySheet.getRange("B2").setValue(row - 1); // Count
summarySheet.getRange("B4").setValue(rowlong - 1); // Count

```

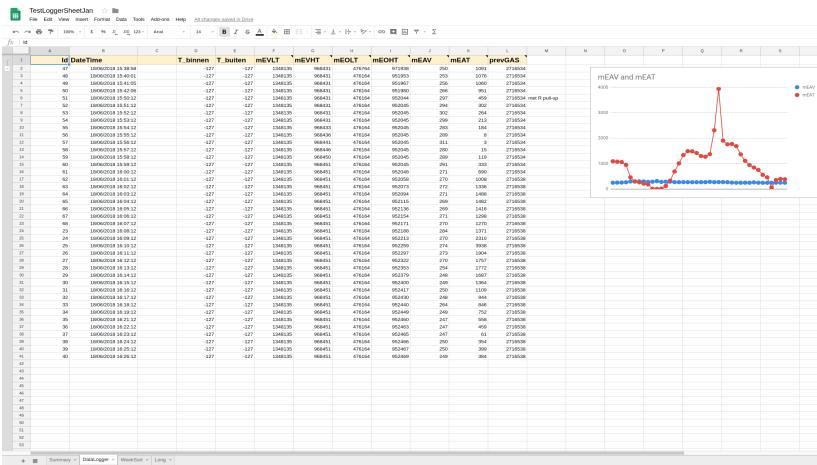
```
var recipient = Session.getActiveUser().getEmail();
var subject = 'A list of files in your Google Drive';
var body = Logger.getLog();
MailApp.sendEmail(recipient, subject, body);

}

catch(error) {
  Logger.log(JSON.stringify(error));
}

Logger.log("--- save_data end---");
}
```

Sheet



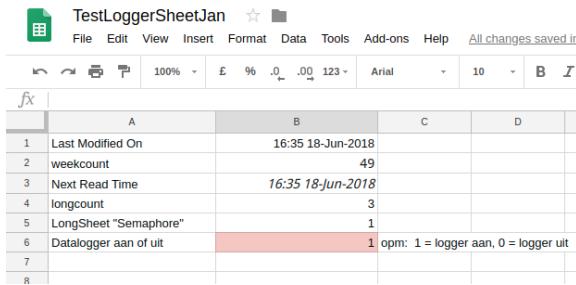
De sheet heeft nu 4 tabbladen: Summary, Datalogger, weeksort en long.

De Datalogger legt alle inkomende informatie vast. Zeker wanneer de informatie maanden lang binnenstroomt kan dit gigantisch worden. De Datalogger kan uitgezet worden op het Summary tabblad.

De weeklogger legt alle informatie op uurbasis vast gedurende 7 dagen. Zondag is dag 0, maandag dag 1 etc.

Long legt informatie op dagbasis vast. De tijdstippen zijn 7, 15 en 23 uur. Momentaan verbruik en teruglevering wordt niet vastgelegd.

Het summary tabblad bevat de boekhouding van bovenstaande spreadsheets.



De datalogger staat aan wanneer je in cel B6 een 1 zet en uit wanneer je er een 0 in zet.

Problemen en to do

- 1) eMVLT wordt niet altijd correct herkent. In het programma in het kastje zit nu een workaround opgenomen. Vermoedelijk zit het probleem in de oplossing met de transistor die het seriele signaal inverseert. Dit probleem is waarschijnlijk opgelost door het verlagen van de pullup naar 470 Ohm.
 - 3) Eerst maar eens een duurproef uitvoeren.
 - 5) Problemen met de Fingerprint. Het werkt ook zonder maar de vraag is hoe lang.
 - 6) Oplossing bedenken voor het invoeren van de Google codes zonder het programma aan te passen. (Heb dit al eens eerder gedaan maar nu wellicht ook met de WiFi manager mogelijk.)
 - 7) Duurproef met de CRC's

Programma (versie 4) in de esp8266

// BY: Jan Jansen
// ON: Junie 2018
// FOR: Piet Swinkels

```
#include <ESP8266WiFi.h>
#include "HTTPSSRedirect.h"
#include <ESP8266httpUpdate.h>
#include <WiFiManager.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include "CRC16.h" // 40-11-2017 slimme meter CRC

// update button
const int updateButton = 2;

// aansturing oled display
#include <Adafruit_GFX.h>
#include <ESP_Adafruit_SSD1306.h>
#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET);

// temperatuur meting met Dallas
#define ONE_WIRE_BUS 14 // gpio14 op de NodeMCU
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
float Tbinnen_val, Tbuit_val;

//Web update via mijn NAS
#define BOOT_AFTER_UPDATE true // zorgt er voor dat de esp boot na remote image laden
//van het te uploaden bestand in SKETCH_BIN Op de NAS staan deze bestanden in de web folder
#define REMOTE_SERVER "http://192.168.0.110"
#define SKETCH_BIN "/ESP_bin/slimmemeter_piet.ino.bin"
String url_upd = "http://sagittaweb.synology.me/ESP_bin/slimmemeter_piet.ino.bin";

int versie = 2;
// The ID below comes from Google Sheets.
// Towards the bottom of this page, it will explain how this can be obtained
const char *GscriptId = "---Your-Google-Script-ID---";

// Push data on this interval
const int dataPostDelay = 1800; // 30 minutes = 30 * 60 * 1000

const char* host = "script.google.com";
const char* googleRedirHost = "script.googleusercontent.com";

const int httpsPort = 443;
HTTPSSRedirect client(httpsPort);

// Prepare the url (without the varying data)
//String url = String("/macros/s/") + "AKfycbwBvLgsWUTdsBe-EcX9jYDtHi9jzSAW1Rtx5pYIX4OjRMOFkJ6Q" + "/exec?";
String url = String("/macros/s/") + "AKfycbwZUkMANjR7cmIk8eGdT5VK3GDDC_AxFEd6BztkFE6QNpIDmM" + "/exec?";
//const char* fingerprint = "8F:BB:13:4B:22:16:D6:C7:1C:F4:E4:43:1A:BD:82:18:29:22:AC:7C"; oude werkte een paar dagen
const char* fingerprint = "97:E1:03:DA:DC:42:20:55:BE:8B:DA:F2:D3:B6:52:CE:4A:D1:B3:B8";
// echo | openssl s_client -connect script.google.com:443 & openssl x509 -fingerprint -noout
// slimme meter 09-11-2017
const bool outputOnSerial = true;
bool check = true;
// Slimme meter variabelen
volatile long mEVLT = 0, EVLT; //Meter reading Electrics - consumption low tariff
volatile long mEVHT = 0, EVHT; //Meter reading Electrics - consumption high tariff
volatile long mEOLT = 0, EOLT; //Meter reading Electrics - return low tariff
volatile long mEOHT = 0, EOHT; //Meter reading Electrics - return high tariff
volatile long mEAV = 0; //Meter reading Electrics - Actual consumption
volatile long mEAT = 0; //Meter reading Electrics - Actual return
volatile long mGAS = 0, GAS; //Meter reading Gas
volatile long prevGAS = 0;
#define MAXLINELENGTH 256 // longest normal line is 47 char (+3 for \r\n\0) Oorspronkelijk 64
char telegram[MAXLINELENGTH];
unsigned int currentCRC = 0;
long now, lastMsg , lastMsg1;
// eo slimme meter

void setup() {
    Serial.begin(115200, SERIAL_8N1 );
    Serial.flush();

    pinMode(updateButton, INPUT_PULLUP);
    WiFiManager wifiManager;
    initdisplay();
    show(" connect WiFi", "probeer");
    wifiManager.autoConnect("SlimmePeter", "Piet222");
    //wifiManager.startConfigPortal("ESP8266", "Piet&Jet");
    initdisplay();

    delay(10000);

    Serial.println(" IP address:");
    Serial.println(WiFi.localIP());
    Serial.print(String("Connecting to "));
    Serial.println(host);
    show("connecting to: ", host);

    bool flag = false;
    for (int i = 0; i < 5; i++) {
        int retval = client.connect(host, httpsPort);
        if (retval == 1) {
            flag = true;
            break;
        }
    }
}
```

```

else
  Serial.println("Connection failed. Retrying...");
  show( "connection failed: ", "opnieuw");
}

// Connection Status, 1 = Connected, 0 is not.
Serial.println("Connection Status: " + String(client.connected()));
Serial.flush();

if (!flag) {
  Serial.print("Could not connect to server: ");
  Serial.println(host);
  Serial.println("Exiting...");
  Serial.flush();
  return;
}

// Data will still be pushed even certification don't match.
if (client.verify(fingerprint, host)) {
  Serial.println("Certificate match.");
} else {
  Serial.println("Certificate mis-match");
  show( "Certificate: ", "mis-match");
}

sensors.begin(); //Dallas temperatuur sensor
show( "Temp sensors: ", "start");
//lees de temperatuur sensor uit
sensors.requestTemperatures(); // Dallas temperatuur sensor
Tbinnen_val = sensors.getTempCByIndex(0);
show( "Temp binnen: ", String(Tbinnen_val));
Tbuit_val = sensors.getTempCByIndex(1);
show( "Temp buiten: ", String(Tbuit_val));
}

// Continue pushing data at a given interval
void loop() {
  if (!digitalRead(updateButton)) {
    show( "Update software?", "druk update button meer dan 5 sec");
    if ( !digitalRead(updateButton)) {
      delay(5000);
    };
    if ( !digitalRead(updateButton)) {
      WebUpdate();
    };
  };
  now = millis();
  if (now - lastMsg1 > 5000) {
    lastMsg1 = now;
    //lees de temperatuur sensor uit
    sensors.requestTemperatures(); // Dallas temperatuur sensor
    Tbinnen_val = sensors.getTempCByIndex(0);
    Tbuit_val = sensors.getTempCByIndex(1);
    showSlim();
  }
}

//lees de slimme meter uit (doe dit zo vaak mogelijk)
readTelegram();
if(mEVLT !=0 && mEVHT !=0 && mEOLT !=0 && mEOHT !=0 && mGAS !=0) {
if (now - lastMsg > 60000) {
  Serial.println("in RT 2");
  Serial.end();
  yield();
}

postData();
lastMsg = now;
Serial.begin(115200, SERIAL_8N1 );
Serial.println("in RT 3");
}
}

// Deze procedure stuurt de data naar de Google sheet .....
void postData() {

if (!client.connected()) {
  Serial.println("Connecting to client again...");
  show( "Connecting: ", "opnieuw");
  client.connect(host, httpsPort);
}

String urlFinal = url + "&Tbinnen_val=" + String(Tbinnen_val) + "&Tbuit=" + Tbuit + "&Tbuit_val=" + String(Tbuit_val) + "&mEAT_val=" + String(mEAT) + "&mEVLT_val=" +
String(mEVLT) + "&mEVHT_val=" + String(mEVHT) + "&mEAV=" + "mEAV" + "&mEAV_val=" + String(mEAV) + "&mEOLT_val=" + String(mEOLT) + "&mEOHT_val=" + String(mEOHT) +
"&mGAS_val=" + String(mGAS);

client.printRedir(urlFinal, host, googleRedirHost);

// Serial.println("end of postdata");
}

//Webupdate van NAS -----
void WebUpdate() {

delay(1000);
Serial.println();
Serial.println();
Serial.println();
Serial.print( "URL: "); Serial.println(url_upd);

for (uint8_t t = 4; t > 0; t--) {
  Serial.printf("[SETUP] WAIT %d..\n", t);
  show( "setup wait", String(t));
  Serial.flush();
}
}

```

```

delay(1000);
}
show( "software update", "WACHTEN Piet");
delay(1000);
ESPhttpUpdate.rebootOnUpdate(BOOT_AFTER_UPDATE); // true
t_httpUpdate_return ret = ESPhttpUpdate.update(url_upd);
switch (ret) {
case HTTP_UPDATE_FAILED:
    Serial.printf("HTTP_UPDATE_FAILED Error (%d): %s", ESPhttpUpdate.getLastErrorCode(), ESPhttpUpdate.getErrorString());
    show( "http update fout", ESPhttpUpdate.getErrorString());
    break;
}

case HTTP_UPDATE_NO_UPDATES:
    Serial.println("HTTP_UPDATE_NO_UPDATES");
    break;

case HTTP_UPDATE_OK:
    Serial.println("HTTP_UPDATE_OK");
    break;
}

//OLED Display 05-06-2018 -----
void initdisplay() {
    display.begin(SSD1306_SWITCHCAPVCC, 0x78 >> 1);      // init done
    display.clearDisplay();
    display.setTextSize(2);
    display.setTextColor(WHITE);
    display.setCursor(0, 0);
    display.println("Piet & Jet");
    display.setTextSize(1);
    display.setCursor(0, 25); display.print("Versie: "); display.println(versie);
    display.setCursor(0, 55); display.print("RSSI: "); display.println(WiFi.RSSI());
    display.setCursor(0, 35); display.print("SSID: "); display.println(WiFi.SSID());
    display.setCursor(0, 45); display.print("IP: "); display.println(WiFi.localIP());
    display.setCursor(0, 55); display.print("RSSI: "); display.println(WiFi.RSSI());
    display.display();
}

//Show een regel (bericht)
void show(String txt, String msg) {
    display.clearDisplay();
    display.setTextSize(1);
    display.setCursor(0, 0);
    display.print("--BERICHTEN--");
    display.setCursor(0, 35);
    display.print(txt);
    display.setCursor(0, 45);
    display.print(msg);
    display.display();
    delay(1000);
}

// display alle waarden van de slimme meter
void showslim() {
    Serial.println("shows slim");
    display.clearDisplay();
    display.setTextSize(1);
    display.setCursor(0, 0); display.print("EVLT: "); display.println(String(mEVLT));
    display.setCursor(0, 9); display.print("EVHT: "); display.println(String(mEVHT));
    display.setCursor(0, 18); display.print("EOLT: "); display.println(String(mEOLT));
    display.setCursor(0, 27); display.print("EOHT: "); display.println(String(mEOHT));
    display.setCursor(0, 36); display.print("EAV : "); display.println(String(mEAV));
    display.setCursor(0, 45); display.print("EAT : "); display.println(String(mEAT));
    display.setCursor(0, 54); display.print("GAS : "); display.println(String(mGAS));
    display.display();
}

// slimme meter 06-11-2017 -----
void readTelegram() {
    // Serial.println("readTelegram");
    if (Serial.available()) {
        memset(telegram, 0, sizeof(telegram));
        while (Serial.available()) {
            int len = Serial.readBytesUntil('\n', telegram, MAXLINELENGTH);
            telegram[len] = '\n';
            telegram[len + 1] = 0;
            //yield();
            if (decodeTelegram(len + 1)) // alleen na valid CRC versturen
            {
                // Post de gegevens naar de Google sheet
                Serial.println("in RT 1");
            }
        }
    }
    bool isNumber(char* res, int len) {
        for (int i = 0; i < len; i++) {
            if (((res[i] < '0') || (res[i] > '9')) && (res[i] != '.' && res[i] != 0)) {
                return false;
            }
        }
        return true;
    }

    int FindCharInArrayRev(char array[], char c, int len) {
        for (int i = len - 1; i >= 0; i--) {
            if (array[i] == c) {
                return i;
            }
        }
    }
}

```

```

return -1;
}

long getValue(char* buffer, int maxlen) {
// de waarde staat tussen ( en *
int s = FindCharInArrayRev(buffer, '(', maxlen - 2);
if (s < 8) return 0;
if (s > 32) s = 32;
int l = FindCharInArrayRev(buffer, "*", maxlen - 2) - s - 1;
if (l < 4) return 0;
if (l > 12) return 0;
char res[16];
memset(res, 0, sizeof(res));

if (strcmp(res, buffer + s + 1, l)) {
    if (isNumber(res, l)) {
        return (1000 * atof(res));
    }
}
return 0;
}

bool decodeTelegram(int len) {
//need to check for start
int startChar = FindCharInArrayRev(telegram, '/', len);
int endChar = FindCharInArrayRev(telegram, '!', len);
bool validCRCFound = false;
if(startChar>=0)
{
    //start found. Reset CRC calculation
    currentCRC=CRC16(0x0000,(unsigned char *) telegram+startChar, len-startChar);
    if(outputOnSerial)
    {
        for(int cnt=startChar; cnt<len-startChar;cnt++)
            Serial.print(telegram[cnt]);
    }
    //Serial.println("Start found!");
}

else if(endChar>=0)
{
    //add to crc calc
    currentCRC=CRC16(currentCRC,(unsigned char*)telegram+endChar, 1);
    char messageCRC[5];
    strcpy(messageCRC, telegram + endChar + 1, 4);
    messageCRC[4]=0; //thanks to HarmOtten (issue 5)
    if(outputOnSerial)
    {
        for(int cnt=0; cnt<len;cnt++)
            Serial.print(telegram[cnt]);
    }
    validCRCFound = (strtol(messageCRC, NULL, 16) == currentCRC);
    if(validCRCFound)
        Serial.println("\nVALID CRC FOUND!");
    else
        Serial.println("\n==INVALID CRC FOUND!===");
    currentCRC = 0;
}
else
{
    currentCRC=CRC16(currentCRC, (unsigned char*)telegram, len);
    if(outputOnSerial)
    {
        for(int cnt=0; cnt<len;cnt++)
            Serial.print(telegram[cnt]);
    }
}

// 1-0:1.8.1(000992.992*kWh)
// 1-0:1.8.1 = Elektra verbruik laag tarief (DSMR v4.0)
if (strcmp(telegram, "1-0:1.8.1", strlen("1-0:1.8.1")) == 0)

mEVLT = getValue(telegram, len);

// 1-0:1.8.2(000560.157*kWh)
// 1-0:1.8.2 = Elektra verbruik hoog tarief (DSMR v4.0)
if (strcmp(telegram, "1-0:1.8.2", strlen("1-0:1.8.2")) == 0)

mEVHT = getValue(telegram, len);

// 1-0:2.8.1(000348.890*kWh)
// 1-0:2.8.1 = Elektra opbrengst laag tarief (DSMR v4.0)
if (strcmp(telegram, "1-0:2.8.1", strlen("1-0:2.8.1")) == 0)

mEOLT = getValue(telegram, len);

// 1-0:2.8.2(000859.885*kWh)
// 1-0:2.8.2 = Elektra opbrengst hoog tarief (DSMR v4.0)
if (strcmp(telegram, "1-0:2.8.2", strlen("1-0:2.8.2")) == 0)

mEOHT = getValue(telegram, len);

// 1-0:1.7.0(00.424*kW) Actueel verbruik
// 1-0:2.7.0(00.000*kW) Actuele teruglevering
// 1-0:1.7.x = Electricity consumption actual usage (DSMR v4.0)
if (strcmp(telegram, "1-0:1.7.0", strlen("1-0:1.7.0")) == 0)
    mEAV = getValue(telegram, len);
}

```

```

if (strncmp(telegram, "1-0:2.7.0", strlen("1-0:2.7.0")) == 0)
    mEAT = getValue(telegram, len);

// 0-1:24.2.1(15053120000S)(00811.923*m3)
// 0-1:24.2.1 = Gas (DSMR v4.0) on Kaifa MA105 meter
if (strncmp(telegram, "0-1:24.2.1", strlen("0-1:24.2.1")) == 0)
    mGAS = getValue(telegram, len);

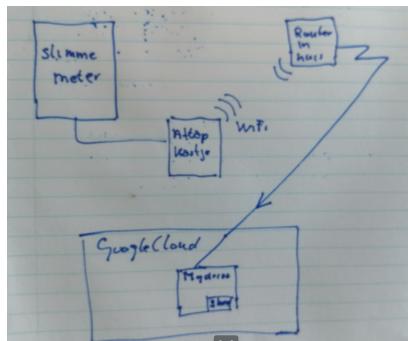
if (mEVLT != 0) EVLT = mEVLT;if (mEVHT != 0) EVHT = mEVHT;if (mGAS != 0) GAS=mGAS;
//if (mEVLT >= (EVLT+2*EVLT/100) ||mEVLT <= (EVLT-2*EVLT/100) || mEVLT >= (EVHT+2*EVHT/100) ||mEVHT <= (EVHT-2*EVHT/100)|| mGAS >= (GAS+2*GAS/100) ||mGAS <= (GAS-2*GAS/100))
// { EVLT=mEVLT; EVHT=mEVHT;validCRCFound = false; Serial.println( "Check 1");return validCRCFound; }

//validCRCFound = true;
return validCRCFound;
}

// eo slimme meter

```

Netwerk details (hoe knoop je “alles” aan elkaar)

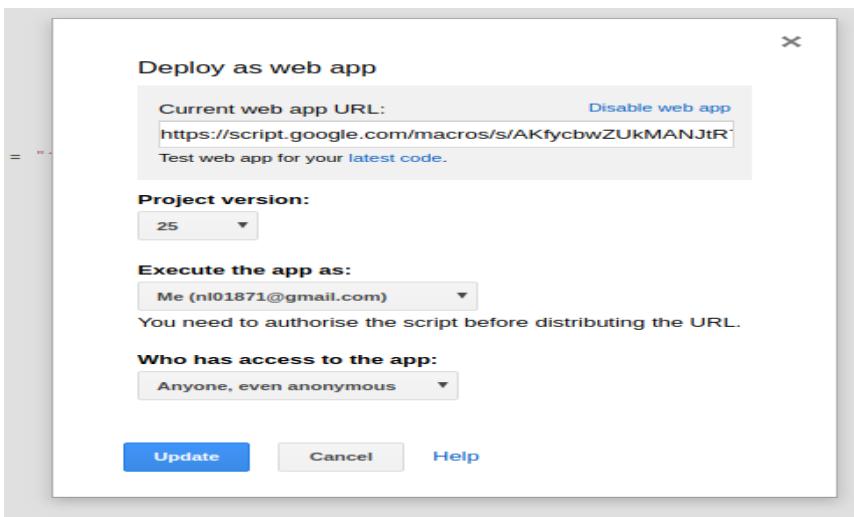


Links het globale plaatje.

Maar hoe knoop je dit netwerk nu eigenlijk aan elkaar?

Google heeft webpagina's aan die je kan koppelen met sheets in de Google Cloud.

De webpagina's bevatten JavaScript code (de HTML zien we niet) en deze JavaScript is in staat om een (jouw eigen) sheet in de cloud te modifieren. Die Google webpagina's beginnen met: <https://script.google.com>. Je kan deze webpagina's aanpassen via een Google sheet onder Tools <> Script editor. Zo'n webpagina heeft een uniek internet adres. Dat kan je vinden zoals onderstaand:



Het unieke is het deel van de current Web app URL tussen s/ en /exec.

https://script.google.com/macros/s/AKfycbwZUkMANJtR7cmIk8eGdT5VK3GDDC_AxFEd6BztkFE6QNpIDmM/exec.

Je dient de current web app link in zijn geheel te copiëren zoals hierboven en dan is het unieke deel: [AKfycbwZUkMANJtR7cmIk8eGdT5VK3GDDC_AxFEd6BztkFE6QNpIDmM](https://script.google.com/macros/s/AKfycbwZUkMANJtR7cmIk8eGdT5VK3GDDC_AxFEd6BztkFE6QNpIDmM/exec)

De ESP8266

De esp8266 dient zijn informatie naar dit web adres te forwarden. Je komt het in de esp8266 tegen op ca regel 52: String **url** = String("/macros/s/") +
"[AKfycbwZUkMANJtR7cmIk8eGdT5VK3GDDC_AxFEd6BztkFE6QNpIDmM](https://script.google.com/macros/s/AKfycbwZUkMANJtR7cmIk8eGdT5VK3GDDC_AxFEd6BztkFE6QNpIDmM/exec)" + "/exec?";

In de esp code zie je de "GET" op dit webadres met alle informatie terug:

```
void postData() {  
    if (!client.connected()) {  
        Serial.println("Connecting to client again...");  
        show( "Connecting: ", "opnieuw");  
        client.connect(host, httpsPort);  
    }  
  
    String urlFinal = url + "&Tbinnen_val=" + String(Tbinnen_val) + "&Tbuit=" + String(Tbuit) + "&Tbuit_val=" + String(Tbuit_val) + "&mEAT_val=" + String(mEAT) + "&mEVLT_val=" + String(mEVLT) + "&mEVHT_val=" + String(mEVHT) + "&mEAV=" + "mEAV" + "&mEAV_val=" + String(mEAV) + "&mEOLT_val=" + String(mEOLT) + "&mEOHT_val=" + String(mEOHT) + "&mGAS_val=" + String(mGAS);  
    client.printRedir(urlFinal, host, googleRedirHost);  
}
```

Google web pagina

De "Get" wordt geadresseerd aan de Google web pagina en die ontrekt de slimme meter data aan de "GET".

```
function doGet(e){  
var Tbuit = "1", Tbinnen = "1", mEVLT = "2", mEVHT = "1", mEOLT = "2", mEOHT = "1", mEAT = "1", mEAV = "2", mGAS = "1";  
try {  
    Tbinnen = e.parameters.Tbinnen_val;  
    Tbuit = e.parameters.Tbuit_val;  
    mEVLT = e.parameters.mEVLT_val;  
    mEVHT = e.parameters.mEVHT_val;  
    mEOLT = e.parameters.mEOLT_val;  
    mEOHT = e.parameters.mEOHT_val;  
    mEAT = e.parameters.mEAT_val;  
    mEAV = e.parameters.mEAV_val;  
    mGAS = e.parameters.mGAS_val;
```

In de Google web pagina dient de informatie geschreven te worden naar de sheet himself.

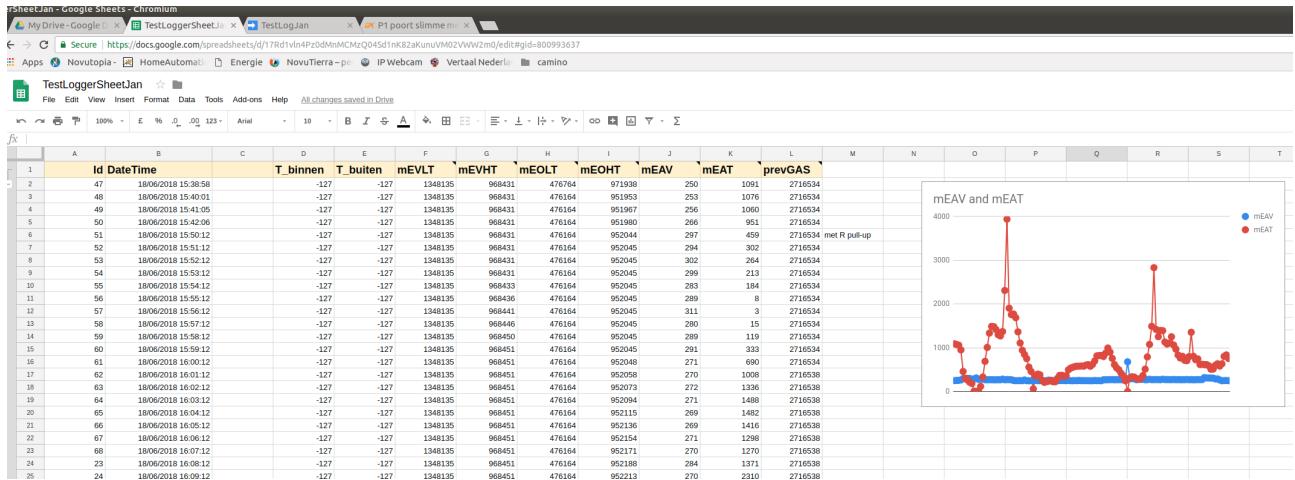
De Google sheet

Ook deze sheet heeft weer een web adres dat nu bekend moet zijn in de Google web pagina.

```
// Paste the URL of the Google Sheets starting from https thru /edit  
// For e.g.: https://docs.google.com/....edit
```

var ss = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/17Rd1vln4Pz0dMnMCMzQ04Sd1nK82aKunuVM02VWW2m0/edit");

De goede lezer zal dit adres tegen komen in het web adres van de Google sheet die we willen bewerken.



Google heeft een complete ontwikkelomgeving obv Javascript om de cellen in het spreadsheet te bewerken.

```

var summarySheet = ss.getSheetByName("Summary");

var dataLoggerSheet = ss.getSheetByName("DataLogger");

var longSheet = ss.getSheetByName("Long");

var weekSortSheet = ss.getSheetByName("weekSort");

// Get last edited row

var row = dataLoggerSheet.getLastRow() + 1;

var rowlong = longSheet.getLastRow() + 1;

var rowWeekSort = weekSortSheet.getLastRow() + 1;

```

CRC 16 Theorie

CRC Generating and Checking

Authors: Thomas Schmidt
Microchip Technology Inc.

INTRODUCTION

This application note describes the Cyclic Redundancy Check (CRC) theory and implementation. The CRC check is used to detect errors in a message. Two implementations are shown:

- Table driven CRC calculation
- Loop driven CRC calculation

This application describes the implementation of the CRC-16 polynomial. However, there are several formats for the implementation of CRC such as CRC-CCITT, CRC-32 or other polynomials.

CRC is a common method for detecting errors in transmitted messages or stored data. The CRC is a very powerful, but easily implemented technique to obtain data reliability.

THEORY OF OPERATION

The theory of a CRC calculation is straight forward. The data is treated by the CRC algorithm as a binary number. This number is divided by another binary number called the polynomial. The rest of the division is the CRC checksum, which is appended to the transmitted message. The receiver divides the message (including the calculated CRC), by the same polynomial the transmitter used. If the result of this division is zero, then the transmission was successful. However, if the result is not equal to zero, an error occurred during the transmission.

The CRC-16 polynomial is shown in Equation 1. The polynomial can be translated into a binary value, because the divisor is viewed as a polynomial with binary coefficients. For example, the CRC-16 polynomial translates to $1000000000000101b$. All coefficients, like x^{16} or x^{15} , are represented by a logical 1 in the binary value.

The division uses the Modulo-2 arithmetic. Modulo-2 calculation is simply realized by XOR'ing two numbers.

EXAMPLE 1: MODULO-2 CALCULATION

$$\begin{array}{r}
 1001100101 \\
 \text{XOR} \quad 0100110111 \\
 = \quad 1101010010
 \end{array}$$

XOR-Function	X ₁	X ₂	Y
	0	0	0
	0	1	1
	1	0	1
	1	1	0

EQUATION 1: THE CRC-16 POLYNOMIAL

$$P(x) = x^{16} + x^{15} + x^2 + 1$$

Example Calculation

In this example calculation, the message is two bytes long. In general, the message can have any length in bytes. Before we can start calculating the CRC value 1, the message has to be augmented by n-bits, where n is the length of the polynomial. The CRC-16 polynomial has a length of 16-bits, therefore, 16-bits have to be augmented to the original message. In this example calculation, the polynomial has a length of 3-bits, therefore, the message has to be extended by three zeros at the end. An example calculation for a CRC is shown in Example 1. The reverse calculation is shown in Example 2.

EXAMPLE 2: CALCULATION FOR GENERATING A CRC

Message = 110101
Polynomial = 101

$$\begin{array}{r}
 11010100 \div 101 = 11101 \\
 \underline{101} \\
 111 \\
 \underline{101} \\
 100 \\
 \underline{101} \\
 110 \\
 \underline{101} \\
 110 \\
 \underline{101} \\
 \underline{\underline{11}}
 \end{array}$$

↑
Quotient (has no function in CRC calculation)

← Remainder = CRC checksum

Message with CRC = 11010111

EXAMPLE 3: CHECKING A MESSAGE FOR A CRC ERROR

Message with CRC = 11010111
Polynomial = 101

$$\begin{array}{r}
 11010111 \div 101 = 11101 \\
 \underline{101} \\
 111 \\
 \underline{101} \\
 100 \\
 \underline{101} \\
 111 \\
 \underline{101} \\
 101 \\
 \underline{101} \\
 \underline{\underline{00}}
 \end{array}$$

↑
Quotient

← Checksum is zero, therefore, no transmission error

CRC generators

http://www.sunshine2k.de/coding/javascript/crc/crc_js.html

Hoe werkt CRC in het esp8266 programma

In de routine decodeTelegram wordt alleerst de CRC van de gehele string berekend:

```
currentCRC=CRC16(0x0000,(unsigned char *) telegram+startChar, len-startChar);
```

0x0000 betekent dat de initieele waarde van de CRC 0 is.

Het lezen van het telegram start bij de “/” en beslaat de rest van het telegram (len-startChar).

Zodra het eind char “! ” gevonden is wordt de rest berekent.

```
currentCRC=CRC16(currentCRC,(unsigned char*)telegram+endChar, 1);
```

Hierbij is het currentCRC het CRC van het gehele telegram.

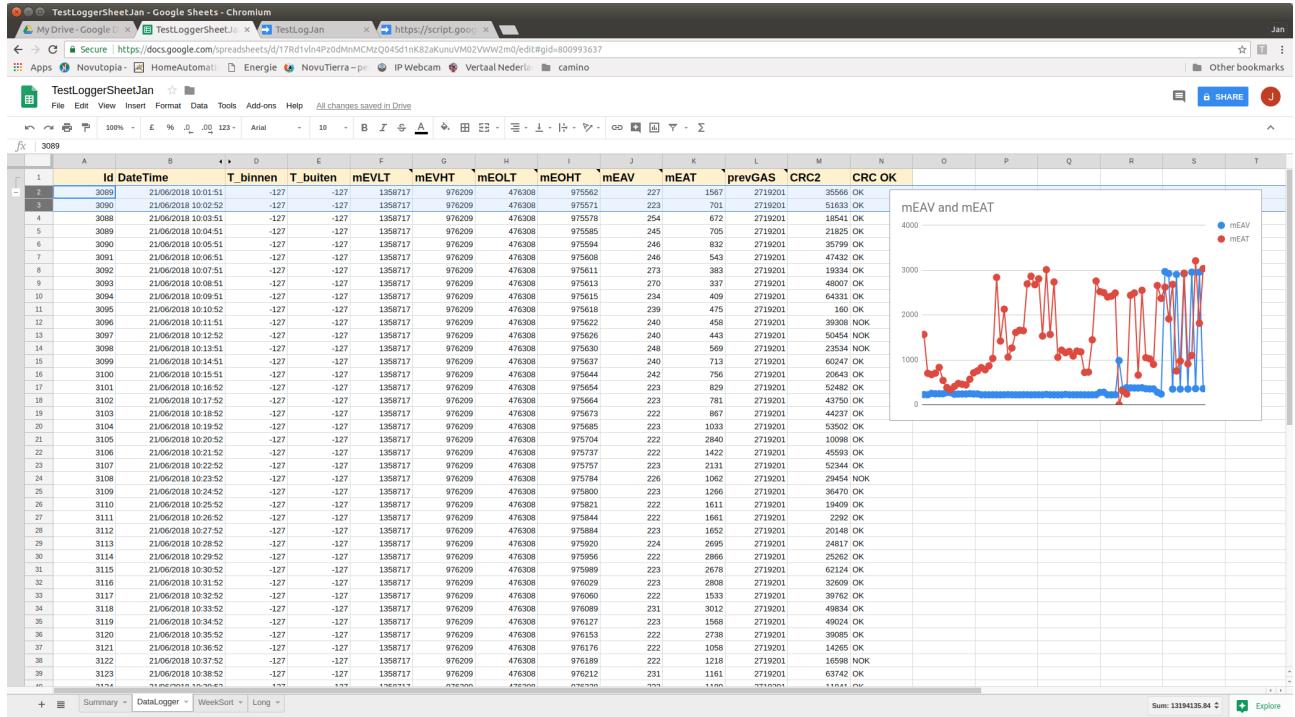
Het CRC van het telegram wordt in messageCRC gelezen:

strncpy(messageCRC, telegram + endChar + 1, 4);

En na enige conversies wordt getest of de CRC's overeenkomen.

validCRCFound = (strtol(messageCRC, NULL, 16) == currentCRC);

validCRCFound wordt doorgestuurd naar de spreadsheet in de Google cloud.



Opvallend is dat de CRC vaak niet ok is (NOK) de waardes in de sheet worden getoond onafhankelijk van de CRC dus is het hopen op overeenkomsten tussen fouten in de gemeten waarden en NOK's.

Problemen CRC

Het CRC mechanisme werkt maar er is een probleem met de lengte van het telegram. De lengte staat standaard op 64:

```
#define MAXLINELENGTH 256 // longest normal line is 47 char (+3 for \r\n\0)
```

Aanvankelijk waren er onbegrepen problemen en omdat ik geen idee had had ik de lengte al vergroot naar 128. Maar elas. Via GitHub contact opgenomen met de ontwikkelaar en die moest bekennen dat het bij hem ook niet meer werkte en is toen gaan testen met een lengte van 120. Ik heb de maxLINELENGTH vergroot naar 256 (ruimte genoeg) en dat bleek voldoende te zijn.

Hij schreef dat de netwerkproviders, binnen de standaard, de firmware veranderen. Dus een punt om op te letten wanneer zich weer problemen voordoen.