Meetup TTN Community Enschede
Woensdag 30 Mei 201
Ariensplein 1
Enschede

TTN LoRa stofsensor

t.b.v.

Apeldoornindata

Jaap Bruijn, TTN Community Apeldoorn

Inhoud:

1. Inleiding (TTN, Vuurwerk, Apeldoornindata)
2. Hardware (Arduino, LoRa, Sensor, Voeding)
3. Software (Arduino IDE, LMIC, dustsensor_01)
4. Meetresultaten (Stofconcentratie, meetduur/periode)
5. Conclusies (Nauwkeurigheid, behuizing, plaats)
6. Aanbevelingen (……, ……., ……., …….?)

Apeldoorn , 26 mei 2018

TTN The Things Netwerk

TTN Community Apeldoorn

How to connect Utrecht E&A exhibition KISS LoRa to TTN?

End Devices (Nodes)

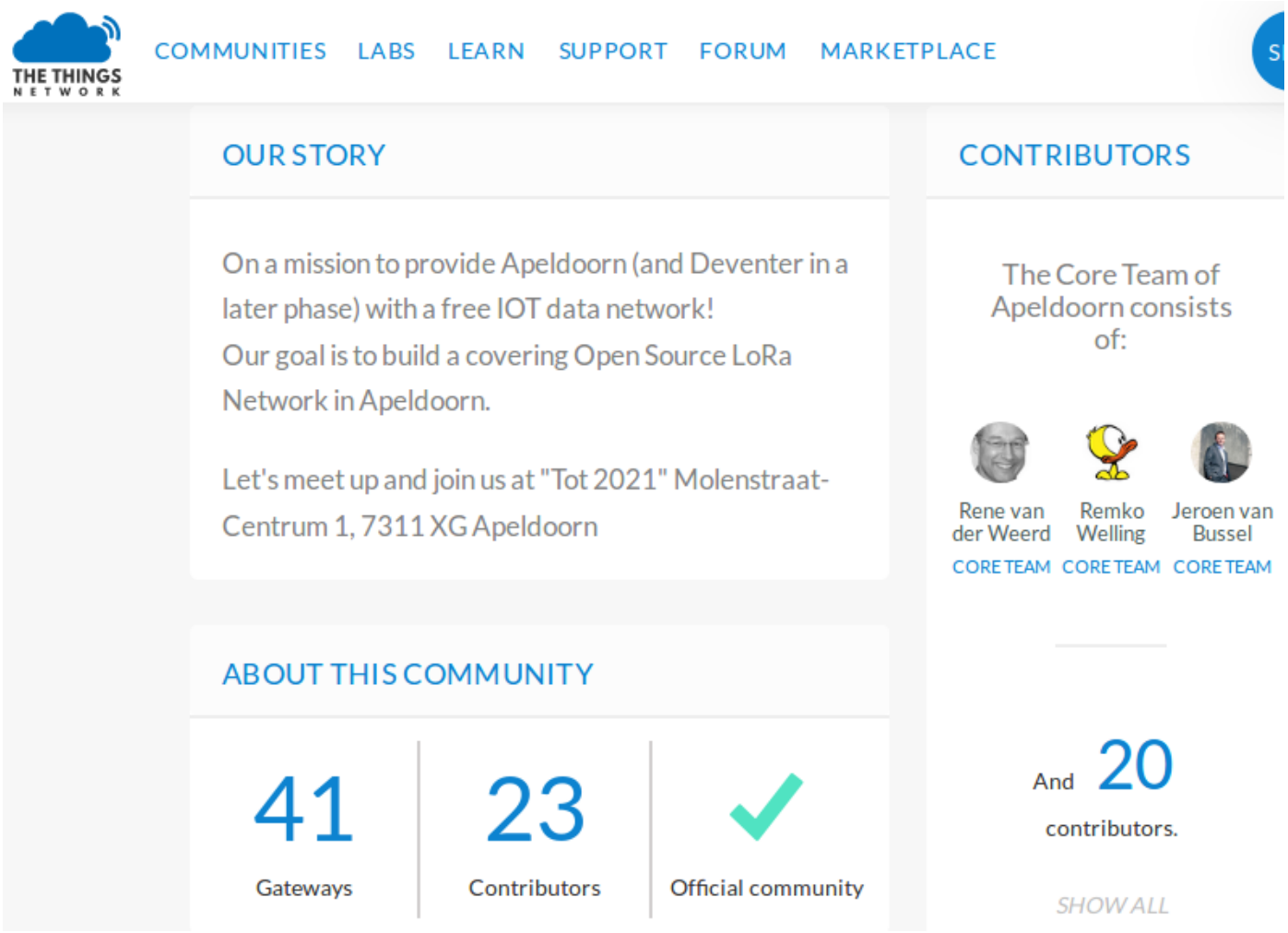
pe1mew Remko — 1 Jun '17

@Jaapbr
I looked up your position in the json you provided.
The area you live will not have highest level of coverage. Especially not for SF7.
I propose to lower to SF9. that will increase probability of coverage I expect.

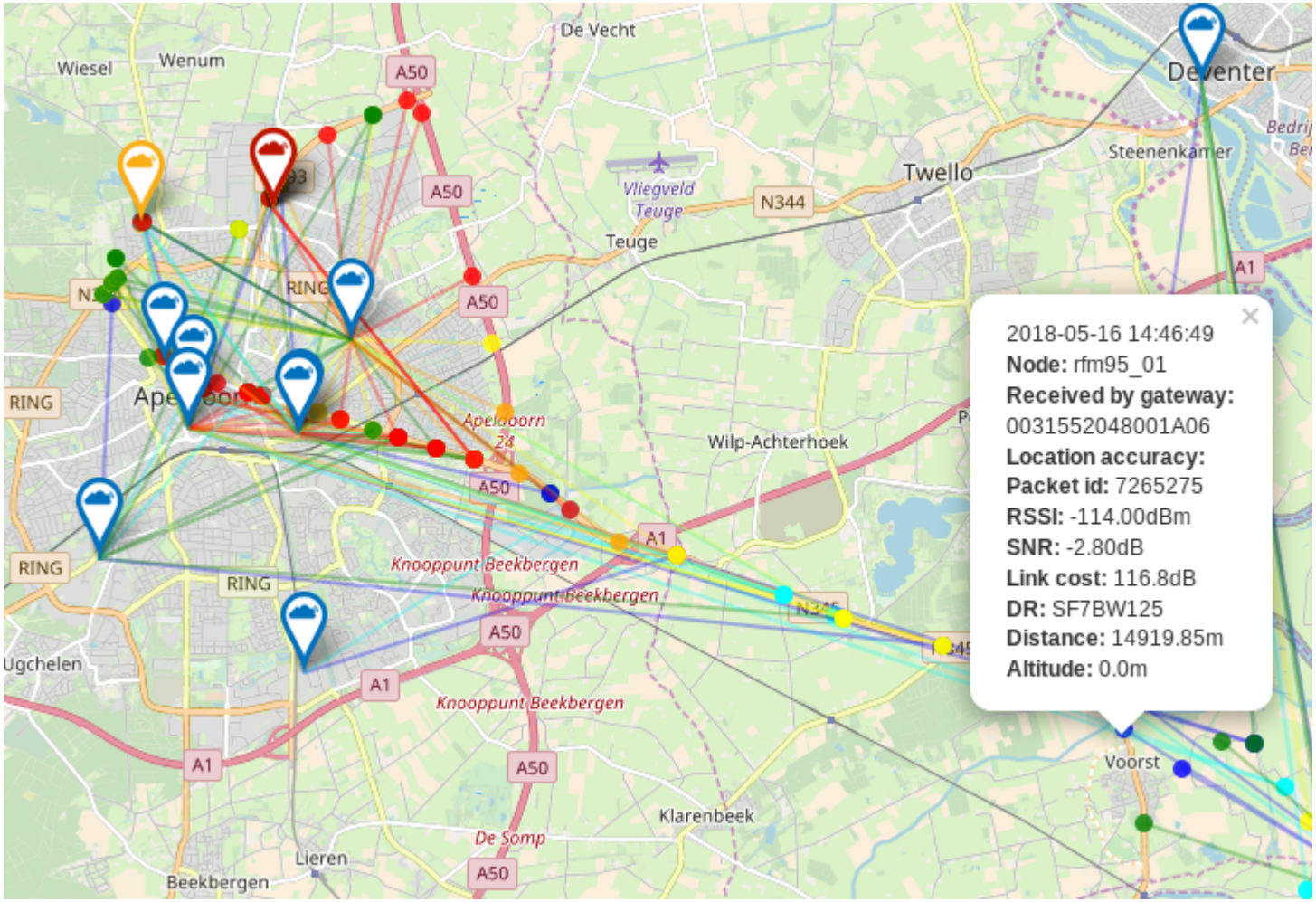
PS: There are currently no plans from the community of Apeldoorn to build aditional gateways in that area.
Unfortunately for you the current focus is on the south.

2 Replies

TTN Community Apeldoorn



TTN Community Apeldoorn



TTN Community Apeldoorn

13
DEC.

woensdag 13 december 2017, 19:30

Bouwavond - vuurwerk + hapje drankje?!

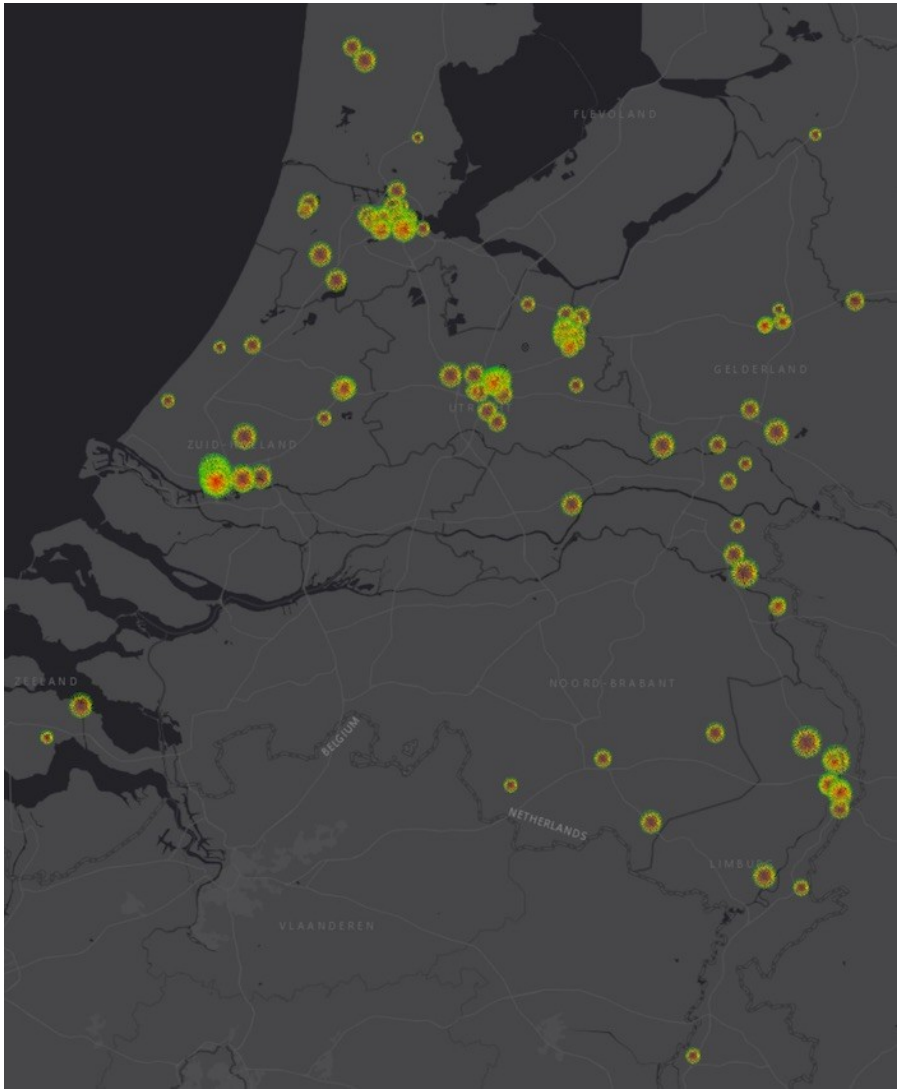
Gehost door Jeroen van B. en 2 anderen

Nu we volledige dekking in Apeldoorn hebben is het tijd om weer eens apps te bouwen en toepassingen te verzinnen. Dus we gaan vanavond bouwen! Rond oud en nieuw wordt er veel vuurwerk de lucht in geschoten. Onze bedoeling is om een LoRa device te bouwen die fijnstof gaat meten. Alle metingen worden verzameld en op een dashboard

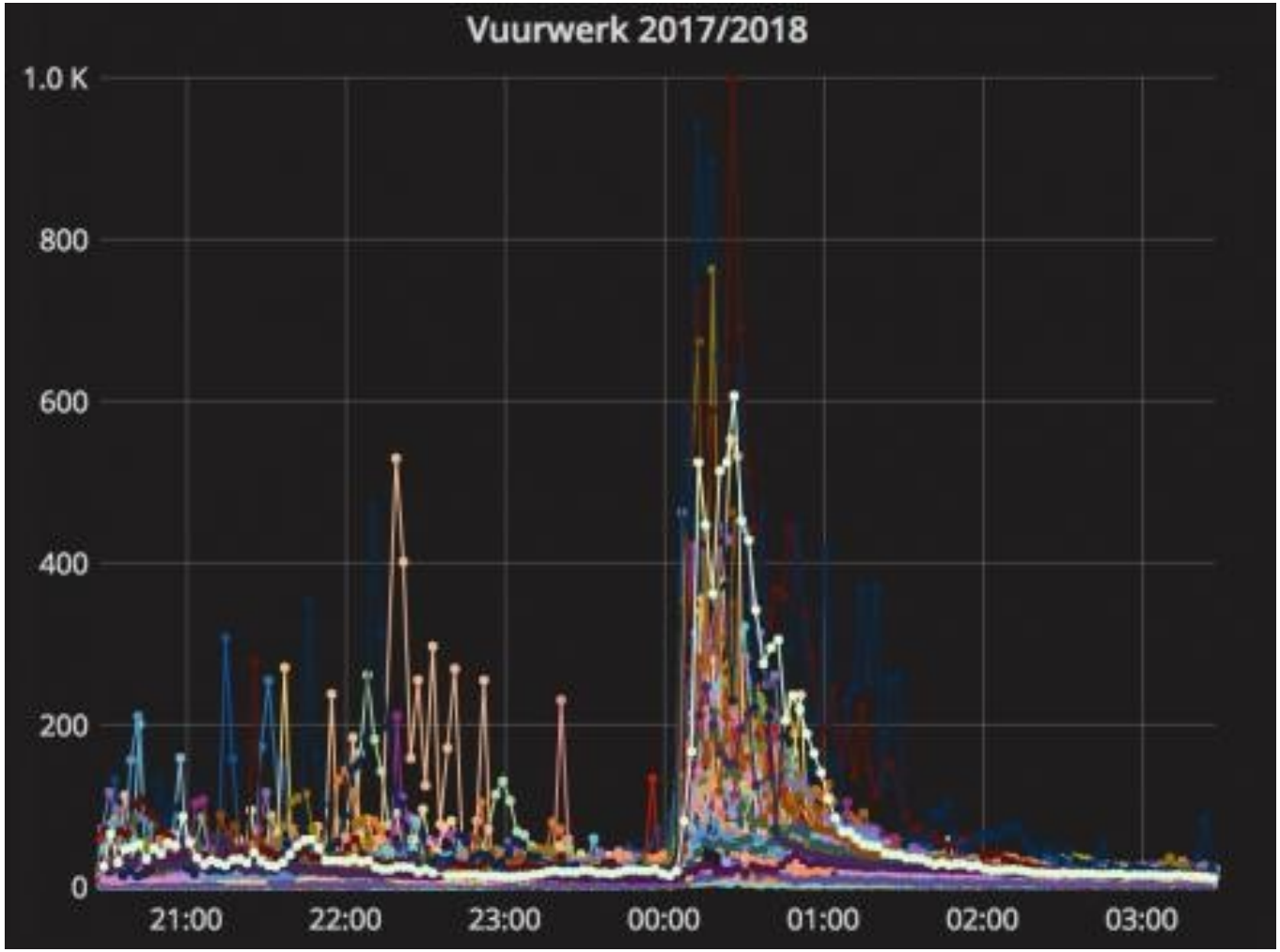
Tot 2021
Molenstraat-Centrum 1 · Apeldoorn

29 reacties

RIVM Vuurwerkexperiment 2016/17



Apeldoornse Vuurwerkdata 2017/18



ApeldoornInData

ApeldoornInData

Arduino boards

This table shows a quick comparison between the characteristics of all the Arduino and Genuino boards.

Name	Processor	Operating/Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [kB]	SRAM [kB]	Flash [kB]	USB	UART
Nano	ATmega168 ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	0.512 1	1 2	16 32	Mini	1
Pro Mini	ATmega328P	3.3 V / 3.35-12 V 5 V / 5-12 V	8 MHz 16 MHz	6/0	14/6	1	2	32	-	1
Uno	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/6	1	2	32	Regular	1

LoRa Circuits/Shields

Hope RFM95:

– Semtech SX 1276 LoRa radio ontvanger/zender
– LoRaWan software op eigen microcontroller

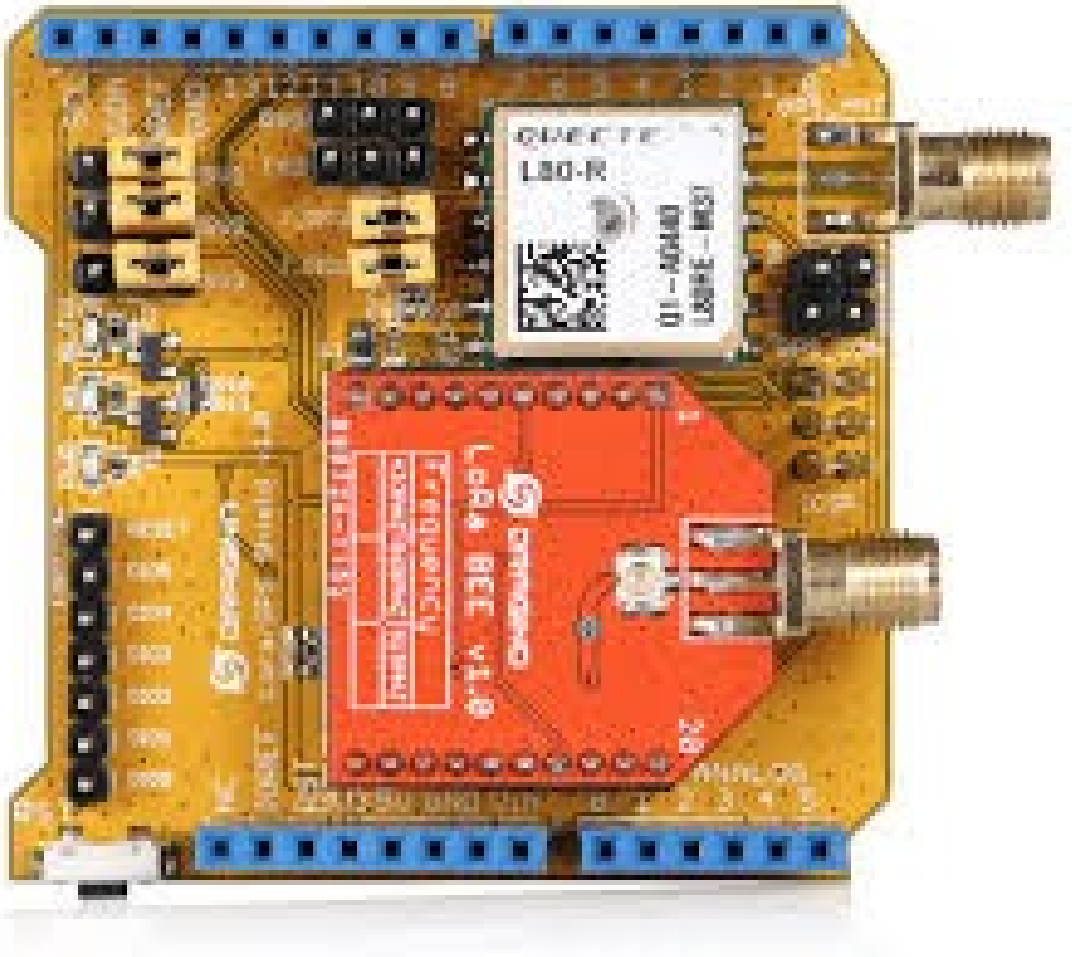
>>>Hope RFM95W circuit (Jaap Fijnstof)
>>>Dragino Arduino Shield (Atranco Fijnstof2)

Microchip RN2483:

– Semtech SX 1276 LoRa radio ontvanger/zender
– Microcontroller met LoRaWan-protocol software

>>>TTN LoRa node

RFM95 Circuits/Shields



Stofsensoren

Shunyei PPD42NJ:

– “pulsjes tellen” ?, onnauwkeurig
– ”stookweerstand”, 90 mA, niet geschikt voor batterijvoeding
– goedkoop Euro 5,-

Sharp GP2Y1010AU0F: !!!!!!!!!!!!!!!

– pulsamplitude, nauwkeurigheid? Calibreren?
– eenvoudig A/D interface
– energie zuinig, 5V, 10 mA
– goedkoop Euro 10,-

(NovaFitness?) SDS011:

– digitale PM 2.5 en PM 10, redelijk nauwkeurig
– ventilator, >100 mA, niet geschikt voot batterijvoeding
– redelijke prijs Euro 30,-

Plantower PMS 7003:

– documentatie ziet er goed uit

Sharp GP2Y1010AU0F

SHARP GP2Y1010AU0F

■ Absolute Maximum Ratings

(T_a=25°C)

Parameter	Symbol	Rating	Unit
Supply voltage	V_{CC}	−0.3 to +7	V
*1 Input terminal voltage	V_{LED}	−0.3 to V_{CC}	V
Operating temperature	T_{opr}	−10 to +65	°C
Soldering temperature	T_{sol}	−20 to +80	°C

*1 Open drain drive input

■Electro-optical Characteristics

(T_a=25°C, V_{CC}=5V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Sensitivity	K	*1 *2 *3	0.35	0.5	0.65	V/(0.1mg/m³)
Output voltage at no dust	V_{OC}	*2 *3	0	0.9	1.5	V
Output voltage range	V_{OH}	*2 *3 R_L=4.7kΩ	3.4	–	–	V
LED terminal current	I_{LED}	*2 LED terminal voltage = 0	–	10	20	mA
Consumption current	I_{CC}	*2 R_L=∞	–	11	20	mA

*1 Sensitivity is specified by the amount of output voltage change when dust density changes by 0.1 mg/m³. And the dust density for detection is a value of the density of cigarette (MILD SEVEN®) smoke measured by the digital dust monitor (P-5L2: manufactured by SHIBATA SCIENTIFIC TECHNOLOGY LTD.).
*2 Input condition is shown in Fig. 1
*3 Output sampling timing is shown in Fig. 2

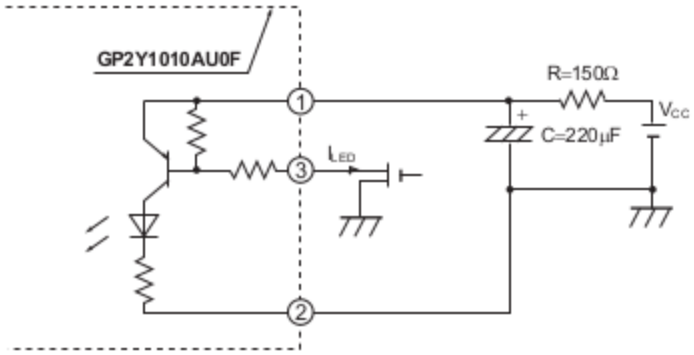
Sharp GP2Y1010AU0F

■**Recommended input condition for LED input terminal**

Parameter	Symbol	Value	Unit
Pulse Cycle	T	10 ± 1	ms
Pulse Width	P_W	0.32 ± 0.02	ms
Operating Supply voltage	V_{CC}	5 ± 0.5	V

SHARP

Fig. 1 Input Condition for LED Input Terminal



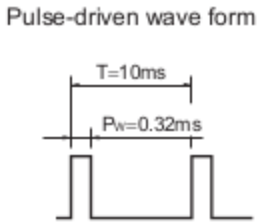
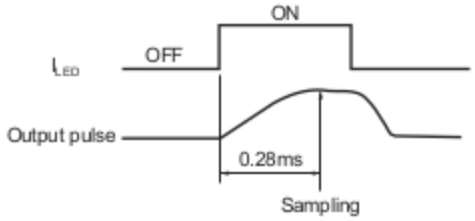
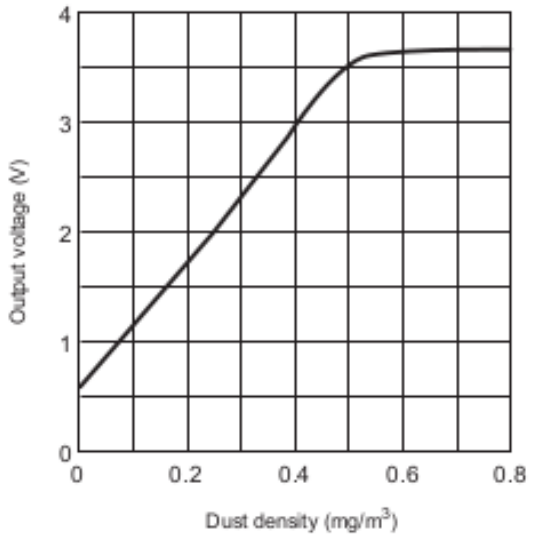


Fig. 2 Sampling Timing of Output Pulse



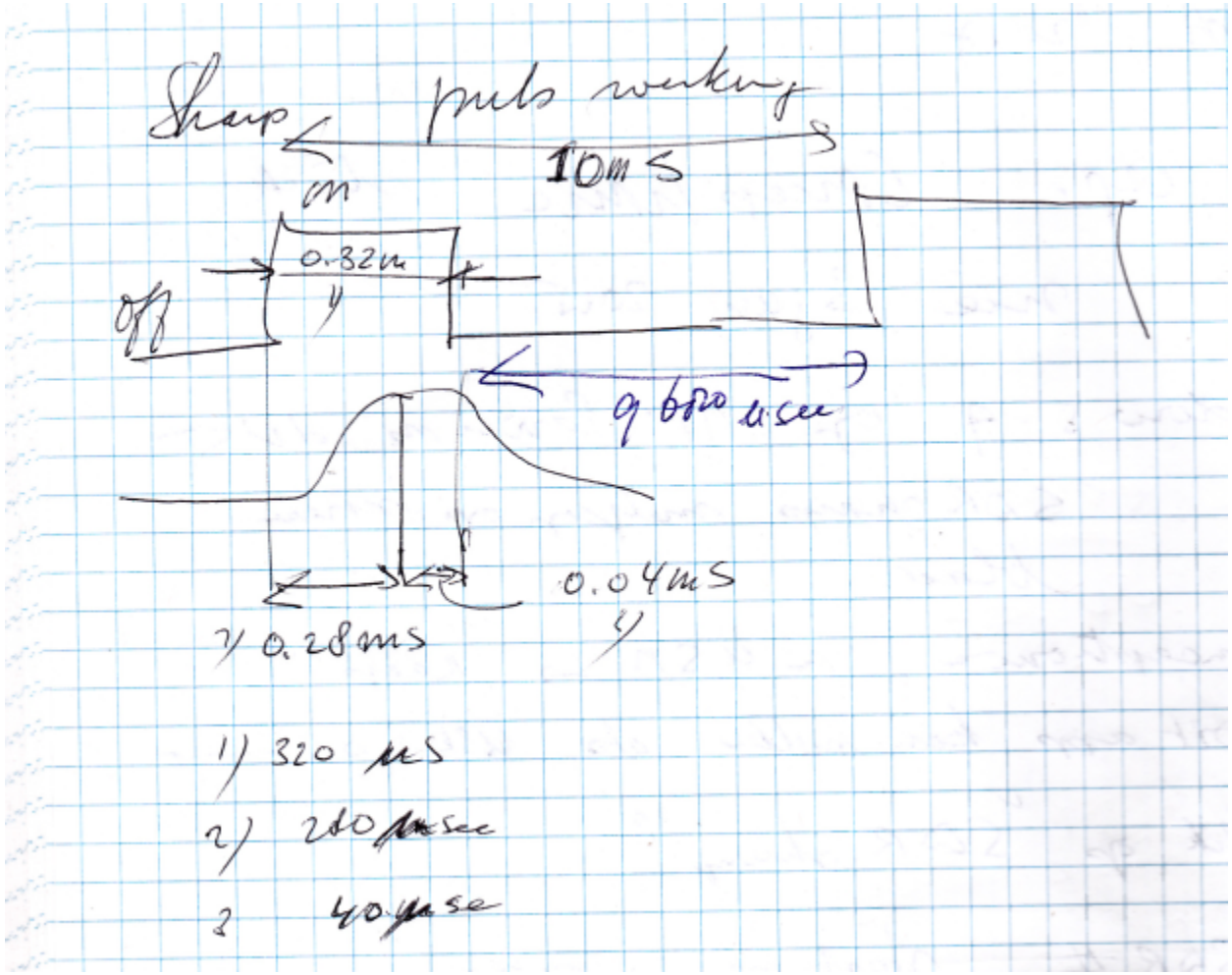
Sharp GP2Y1010AU0F

Fig. 3 Output Voltage vs. Dust Density

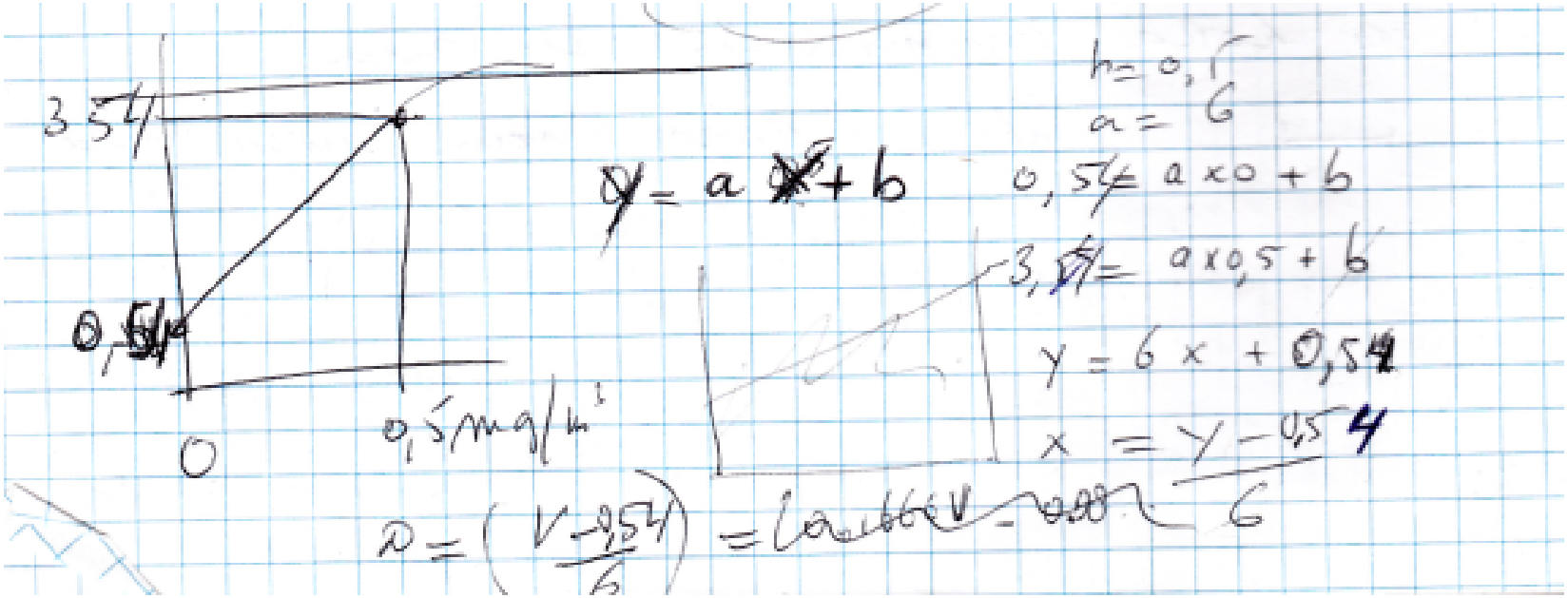


Remarks : Please be aware that all data in the graph are just for reference and are not for guarantee.

Sharp GP2Y1010AU0F



Sharp GP2Y1010AU0F



Stofsensor + Arduino

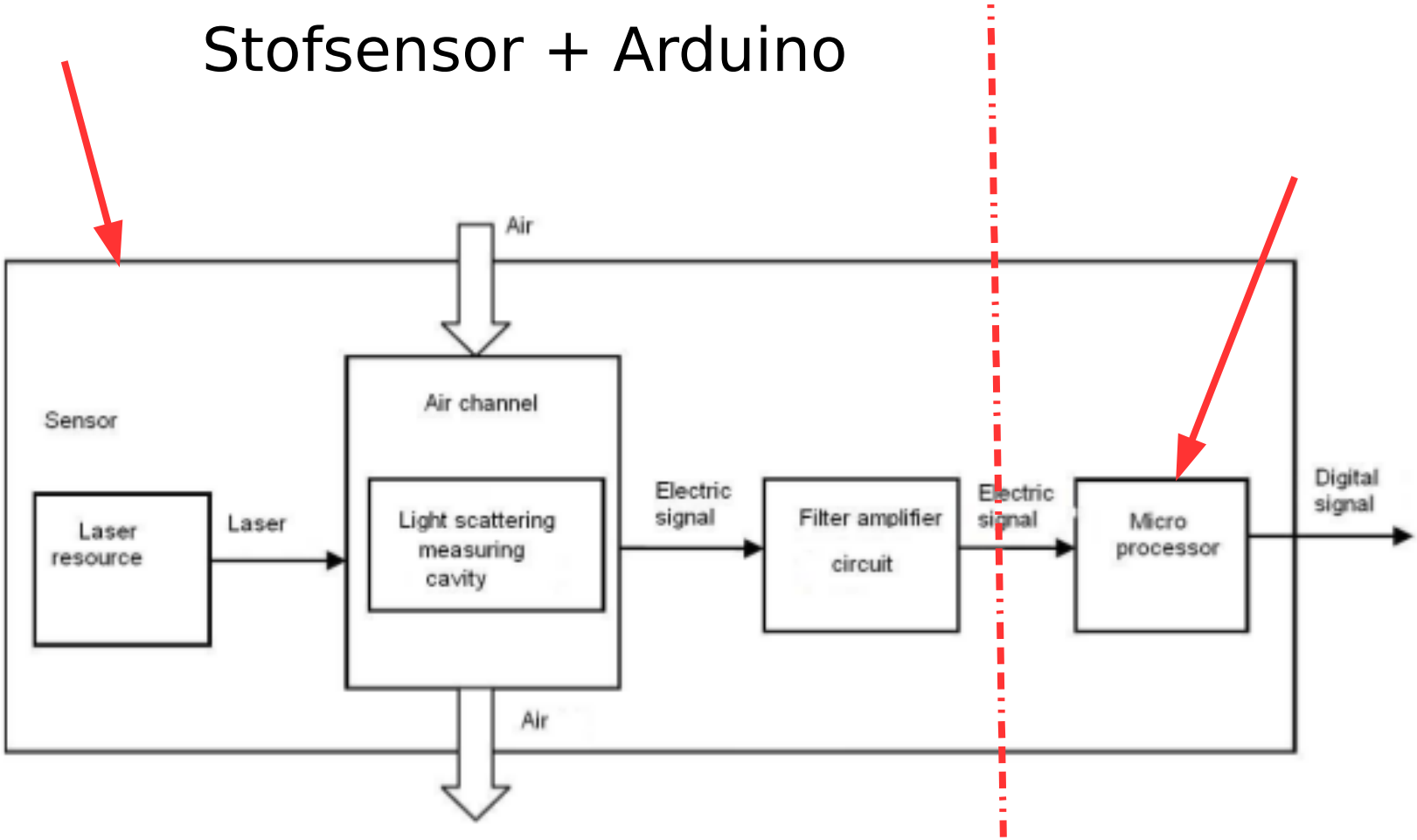


Figure 1 Functional block diagram of sensor

Voeding

USB power 5 V dc :
- direct geschikt voor Arduino Nano en Sharpsensor
- uit PC/laptop tijdens ontwikkeling
- uit Powerbank

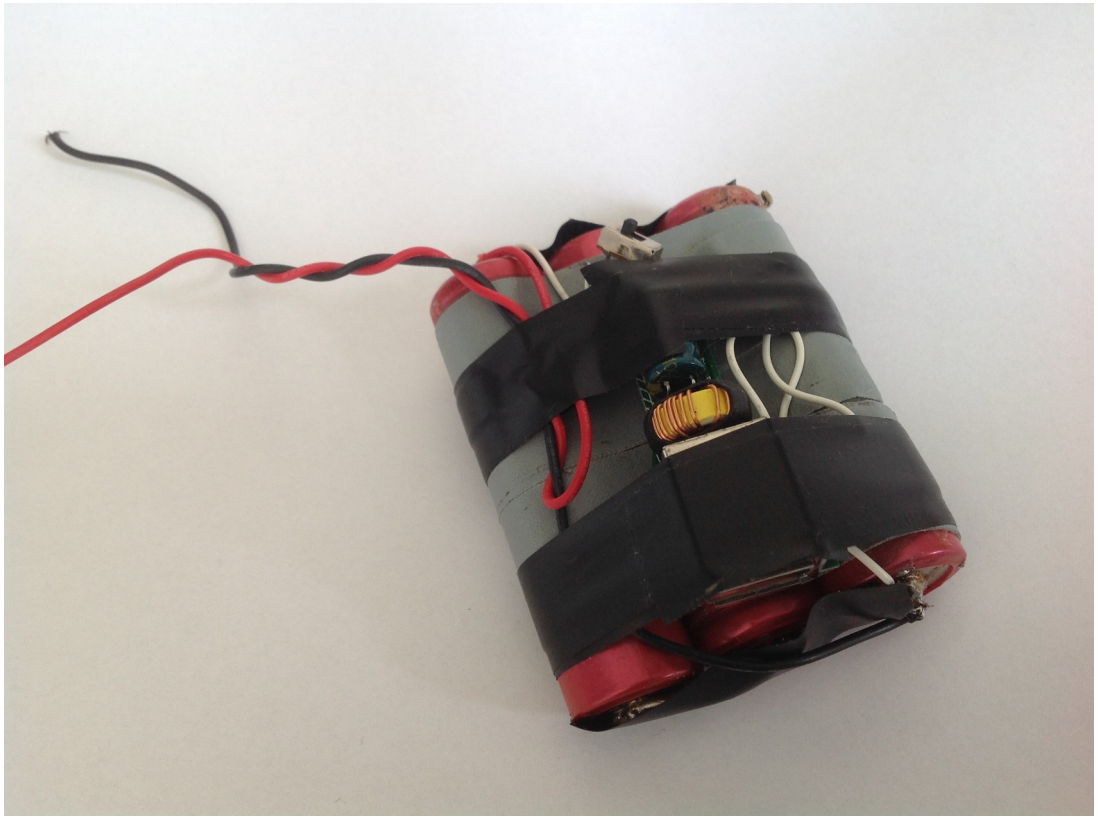
Net 220 V ac :
- omzetten naar USB 5 V dc d.m.v. USB lader

Accu 12 V dc :
- omzetten naar USB 5 V dc d.m.v. USB Car Charger
- bv. Labr31 Duo: 12-24 V in, 5 V /2.4 A uit

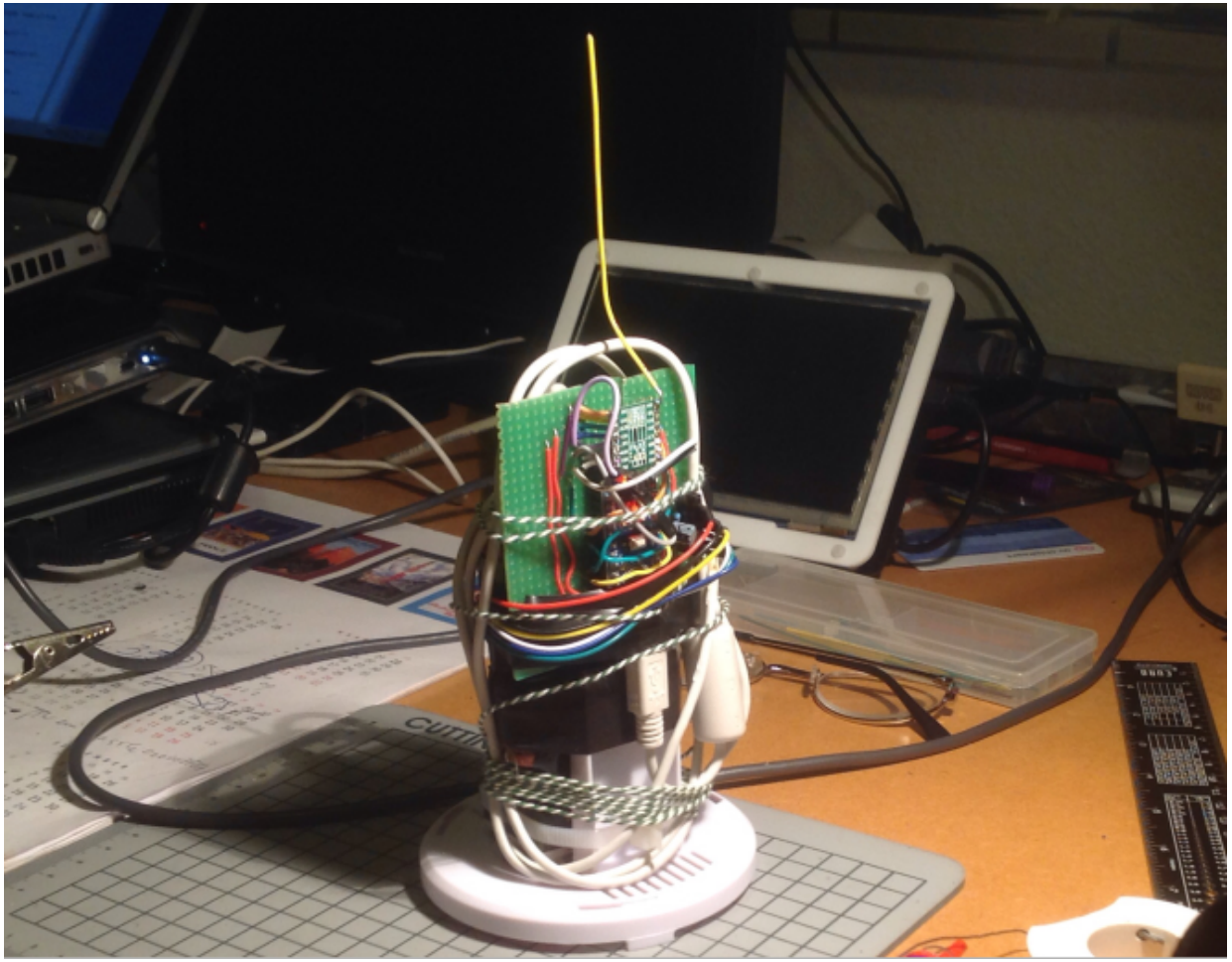
(Oplaadbare) Batterijen x V dc:!!!!!!!!!!!!!!!!!!!
- 3 x oude laptop Lipo's = 11.8 V
- omzetten naar USB 5 V dc d.m.v. USB Car Charger

LiPo Batterijen

– 3 x oude laptop Lipo’s = 11.8 V
– omzetten naar USB 5 V dc d.m.v. USB Car Charger
– opladen met Lab Power Supply: Vmax:12.4 V, Imax:0.4 A



Jaap Fijnstof Sensor



IDE = Integrated Development Environment

IDE:
- editor, programmeertalen, compiler/linker
- microcontrollers, boards
- downloader, debugger, monitor
- Windows, MacOS, Linux
- Webbased Online/Offline
- populariteit, Open Source, gratis

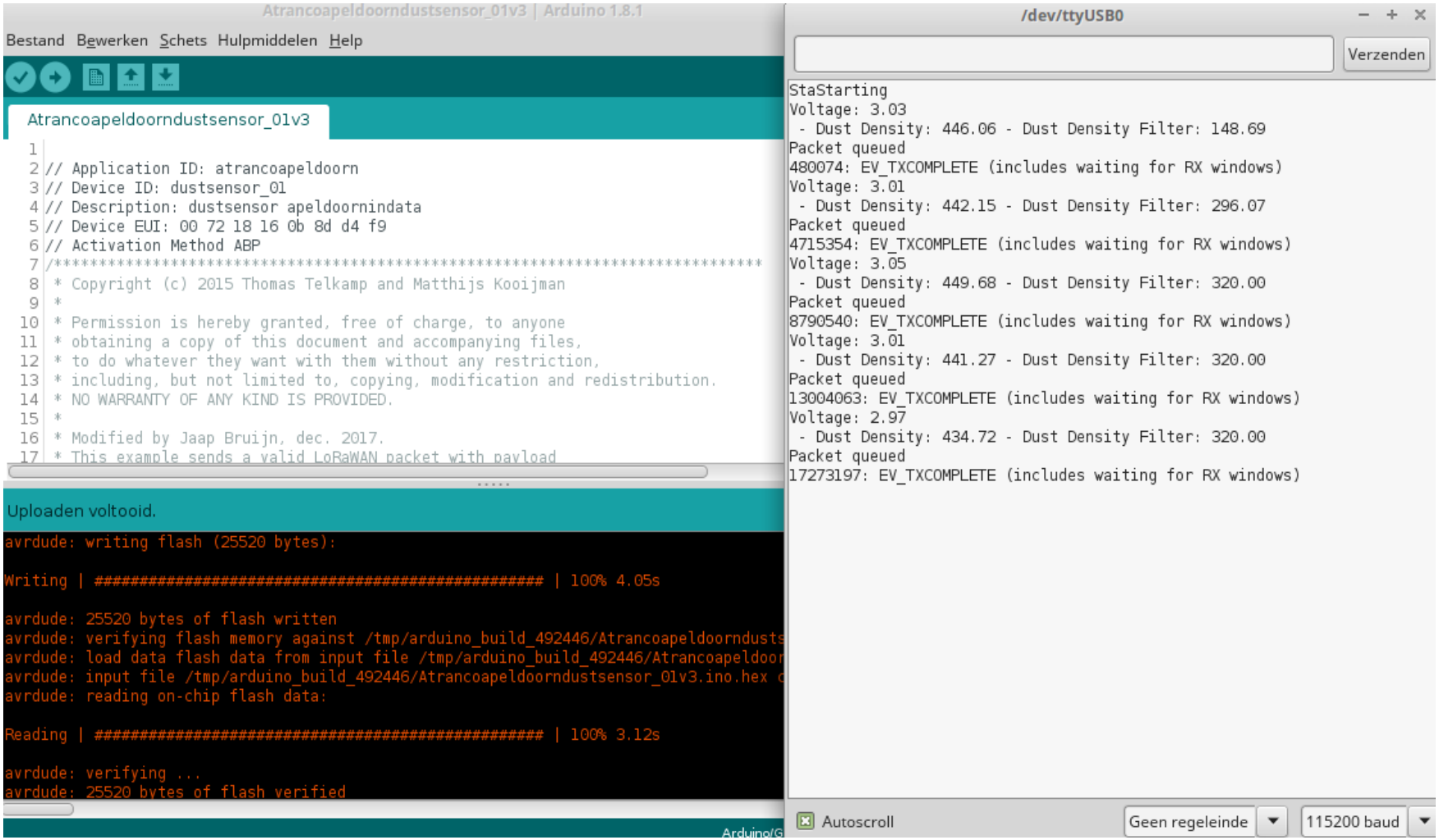
Arduino 1.8.1 : !!!!!!!!!!!!!!!!!
- Uno, Nano
- Linux Mint 18.1 ‘Serena’ Xfce 64-bit
- Offline

Eclipse:
- !@#%%^^**(&^%%%

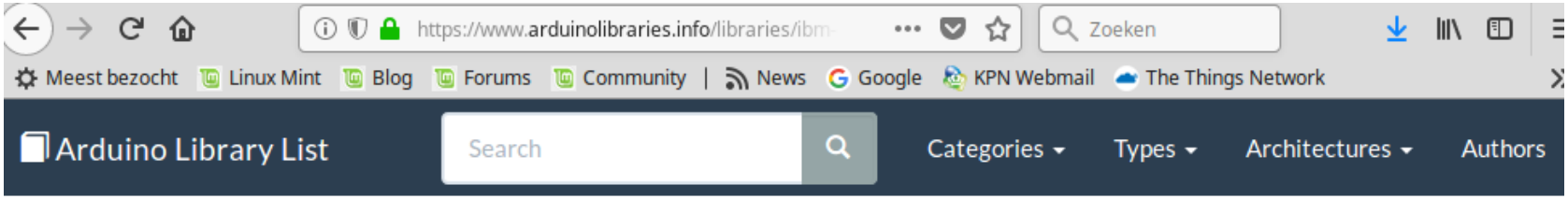
Atom + Platform I/O IDE:
- LoPy, Pymakr, Arduino, ESP32, Mbed

TTN LoRa stofsensor: Software 3.1.2

Arduino 1.8.1



Arduino-limic-v1.5



IBM LMIC framework

Arduino port of the LMIC (LoraWAN-in-C, formerly LoraMAC-in-C) framework provided by IBM.

Author	IBM
Maintainer	Matthijs Kooijman
Website	http://www.research.ibm.com/labs/zurich/ics/lrsc/lmic.html
Github	https://github.com/matthijskooijman/arduino-lmic-v1.5
Category	Communication
License	Unknown
Library Type	Contributed
Architectures	Any

Supports SX1272/SX1276 and HopeRF RFM92/RFM95 tranceivers

Downloads

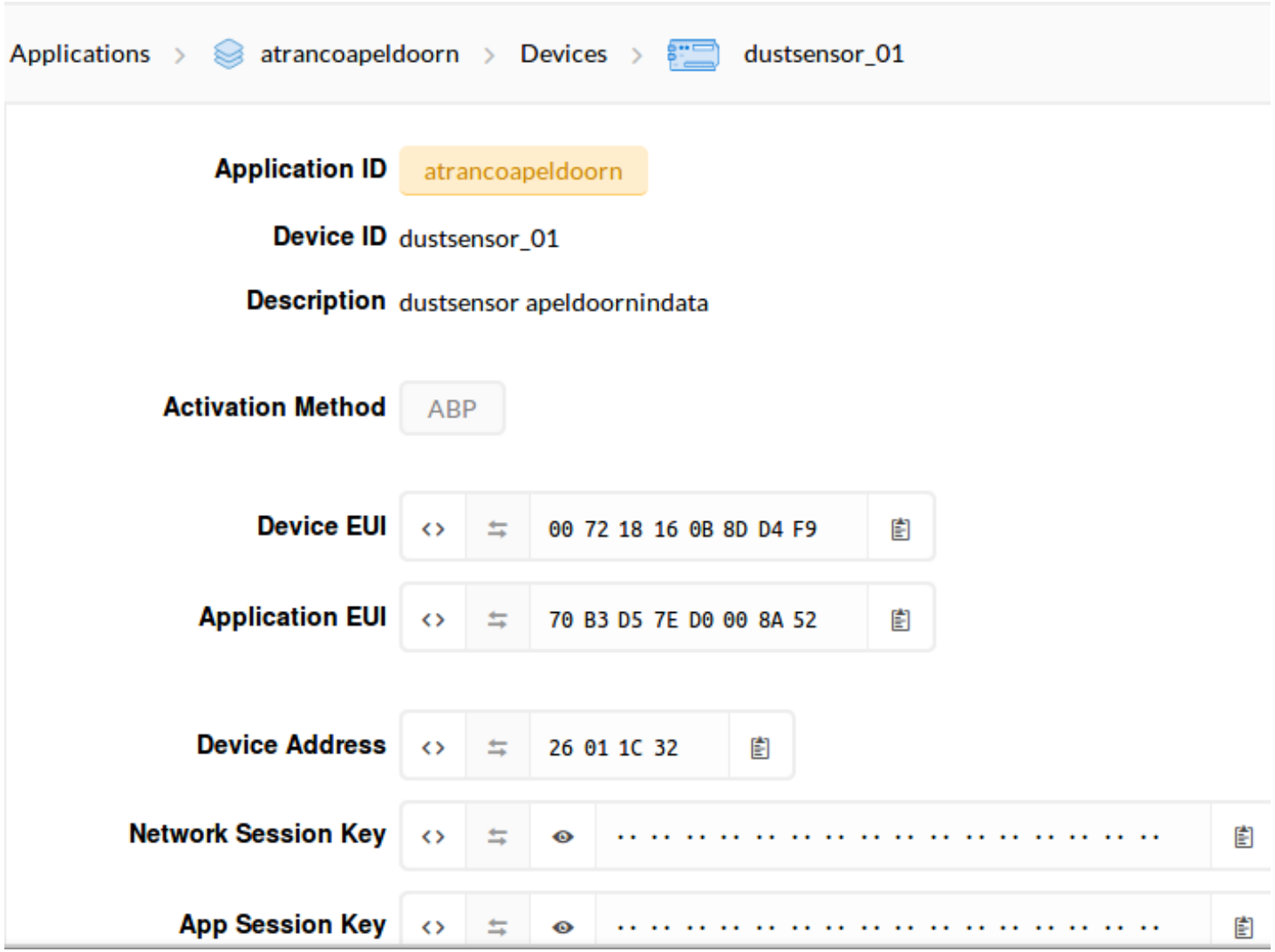
Filename	Release Date	File Size
IBM_LMIC_framework-1.5.0+arduino-2.zip		824.64 KiB
IBM_LMIC_framework-1.5.0+arduino-1.zip		813.70 KiB

Arduino Limic config.h

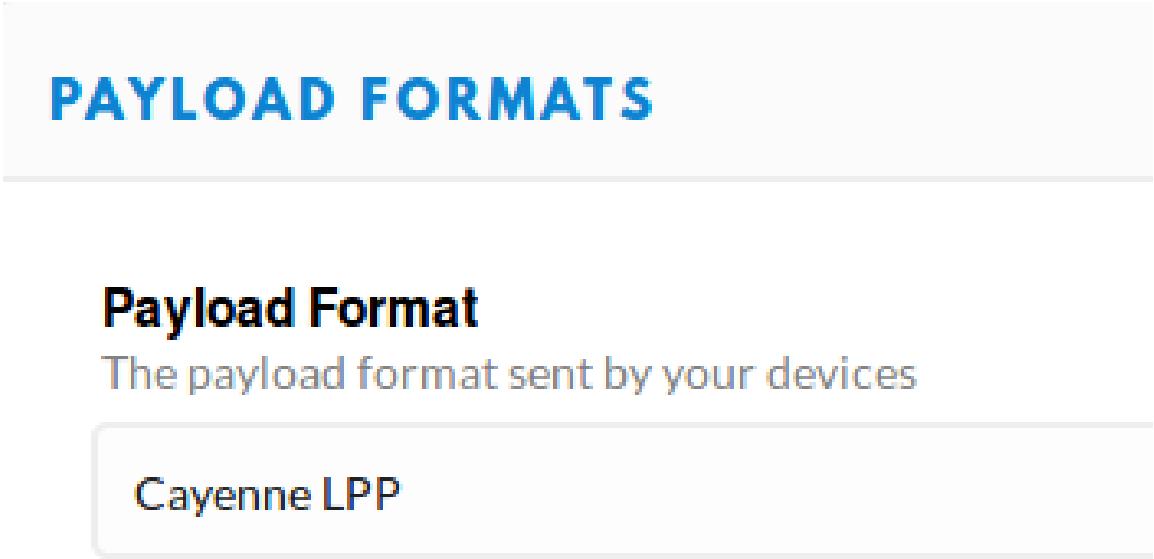
```
// Any runtime assertion failures are printed to this serial port (or
// any other Print object). If this is unset, any failures just silently
// halt execution.
#define LMIC_FAILURE_TO Serial

// Uncomment this to disable all code related to joining
#define DISABLE_JOIN
// Uncomment this to disable all code related to ping
#define DISABLE_PING
// Uncomment this to disable all code related to beacon tracking.
// Requires ping to be disabled too
#define DISABLE_BEACONS

// Uncomment these to disable the corresponding MAC commands.
// Class A
//#define DISABLE_MCMD_DCAP_REQ // duty cycle cap
//#define DISABLE_MCMD_DN2P_SET // 2nd DN window param
//#define DISABLE_MCMD_SNCH_REQ // set new channel
// Class B
#define DISABLE_MCMD_PING_SET // set ping freq, automatically disabled by DISABLE_PING
#define DISABLE_MCMD_BCNI_ANS // next beacon start, automatical disabled by DISABLE_BEACON
```

>>>>>> Code configuren: kleiner maken

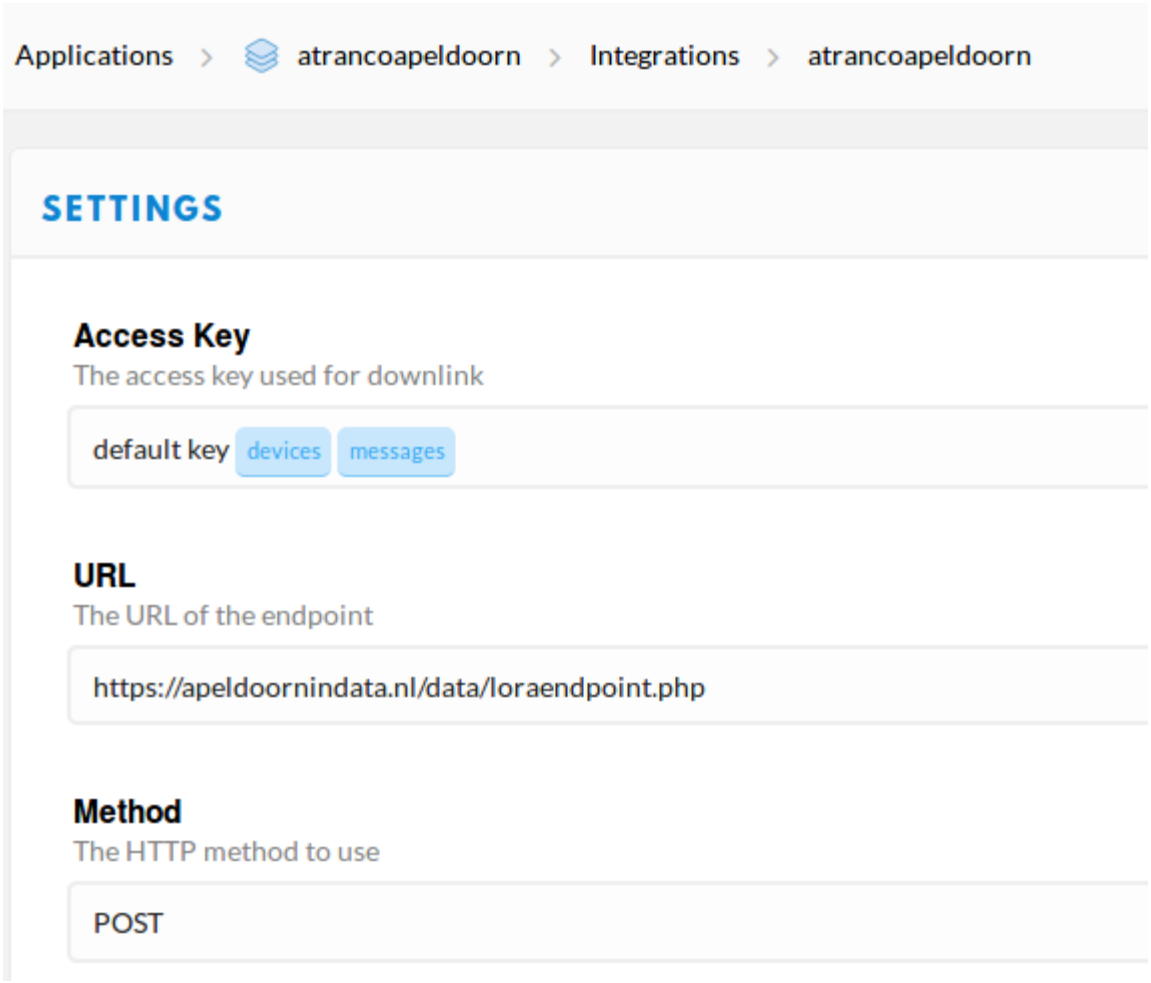
dustsensor_01



dustsensor_01



dustsensor_01>>> apeldoorindata



dustsensor_01v3 TTN keys

AtrancoapeldoorndustsensorV01v3

```
31  *************************************************************************/
32
33 #include <lmic.h>
34 #include <hal/hal.h>
35 #include <SPI.h>
36 #include <CayenneLPP.h>         // include for Cayenne library
37
38 // LoRaWAN NwkSKey, network session key
39 // This is the default Semtech key, which is used by the prototype TTN
40 // network initially.
41 // NB.Nieuwe key te verkrijgen via de TTN Console.
42   static const PROGMEM u1_t NWKSKEY[16] = { 0xE0, 0xCF, 0xCE, 0xB4, 0xDB, 0x9C, 0xC1, 0x35, 0x48
43 //static const PROGMEM u1_t NWKSKEY[16] = { 0xB2, 0x83, 0x95, 0x2D, 0x69, 0xE5, 0x09, 0x28, 0xEF
44
45 // LoRaWAN AppSKey, application session key
46 // This is the default Semtech key, which is used by the prototype TTN
47 // network initially.
48 // NB.Nieuwe key te verkrijgen via de TTN Console.
49   static const u1_t PROGMEM APPSKEY[16] = { 0xC1, 0xD7, 0x3B, 0xE2, 0x3C, 0xAE, 0x95, 0x88, 0x4E
50 //static const u1_t PROGMEM APPSKEY[16] = { 0xBF, 0xCB, 0xAC, 0xA5, 0xF0, 0x83, 0xFC, 0xC1, 0xBC
51
52 // LoRaWAN end-device address (DevAddr)
53 // See http://thethingsnetwork.org/wiki/AddressSpace
54 // Copied from TTN console
55 static const u4_t DEVADDR = 0x26011C32; // <-- Change this address for every node!
56 //static const u4_t DEVADDR = 0x26011390;
57 // These callbacks are only used in over-the-air activation, so they are
58 // left empty here (we cannot leave them out completely unless
59 // DISABLE_JOIN is set in config.h, otherwise the linker will complain).
60 // NB. Alleen ABP activation blijkt goed te werken!
61 void os_getArtEui (u1_t* buf) { }
```

dustsensor_01v3 sensor set+pins

```
/****************** Setup Sharp Dust Sensor ******************
 * NB LMIC uses pins: 10, 9, 2, 6, 7
 * Sharp pin 1 (V-LED)     => 5V (connected through 150ohm resister)
 * Sharp pin 2 (LED-GND)   => Arduino GND pin
 * Sharp pin 3 (LED)       => Arduino pin 3
 * Sharp pin 4 (S-GND)     => Arduino GND pin
 * Sharp pin 5 (Vo)        => Arduino A0 pin
 * Sharp pin 6 (Vcc)       => 5V
 ************************************************************/
//Pin mapping etc.
int measurePin = 0;
int ledPower = 3;
//Timing dustsensor
int samplingTime = 280;
int deltaTime = 40;
int sleepTime = 9680;
//Meetwaarden
float voMsom = 0;
float voMeasured = 0;
float calcVoltage = 0;
float dustDensity = 0;
```

dustsensor_01v3 MA-filter

```
104 // MA-filter, maximaal middelen over 10 meetwaarden = 10 minuten
105 float f1 = 0;
106 float f2 = 0;
107 float f3 = 0;
108 float f4 = 0;
109 float f5 = 0;
110 float f6 = 0;
111 float f7 = 0;
112 float f8 = 0;
113 float f9 = 0;
114 float f10 = 0;
115 float dustDensityFilter = 0;
```

dustsensor_01v3 10 samples=1 meetwaarde

```
194 void do_send(osjob_t* j){
195     // Check if there is not a current TX/RX job running
196     if (LMIC.opmode & OP_TXRXPEND)
197     {
198         Serial.println(F("OP_TXRXPEND, not sending"));
199     }
200     else
201     {
202 /*************Hier de code voor het opvragen van de dustsensor data ********
203 *
204 ****************************************************************************/
205     for (int i = 1; i <= 10; i++)
206     {
207     digitalWrite(ledPower,LOW); // power on the LED
208     delayMicroseconds(samplingTime);
209 
210     voMeasured = analogRead(measurePin); // read the dust value
211 
212     delayMicroseconds(deltaTime);
213     digitalWrite(ledPower,HIGH); // turn the LED off
214     delayMicroseconds(sleepTime);
215 
216     voMsom= voMsom + voMeasured;
217     
218     delay( 90); // looptime=samplingtime: 280+40+9680=10mS + 90mS=0.1 Sec
219     }
220 
221 voMeasured = voMsom/10; // gemiddelde over 10 samples (= 10 x 0.1 = 1 Sec)
222 
223 voMsom =0 ;
```

dustsensor_01v3 convers + lineair

```
226  // 0 - 5 V mapped to 0 - 1023 integer values
227  // recover voltage
228  calcVoltage = voMeasured * (5.0/ 1023);
229
230  // linear equation by
231  // Chris Nafis (c) 2012
232  // dustDensity = 0.17 * calcVoltage - 0.1;
233  // Jaap Bruijn 2017
234  dustDensity = (calcVoltage - 0.8)/5 ; // experimenteel bepalen
235  dustDensity = dustDensity * 1000; // ug/m3
236
237  //Serial.print("Raw Signal Value (0-1023): ");
238  //Serial.print(voMeasured);
239  Serial.print("Voltage: ");
240  Serial.println(calcVoltage);
241
242  Serial.print(" - Dust Density: ");
243  Serial.print(dustDensity);
244
```

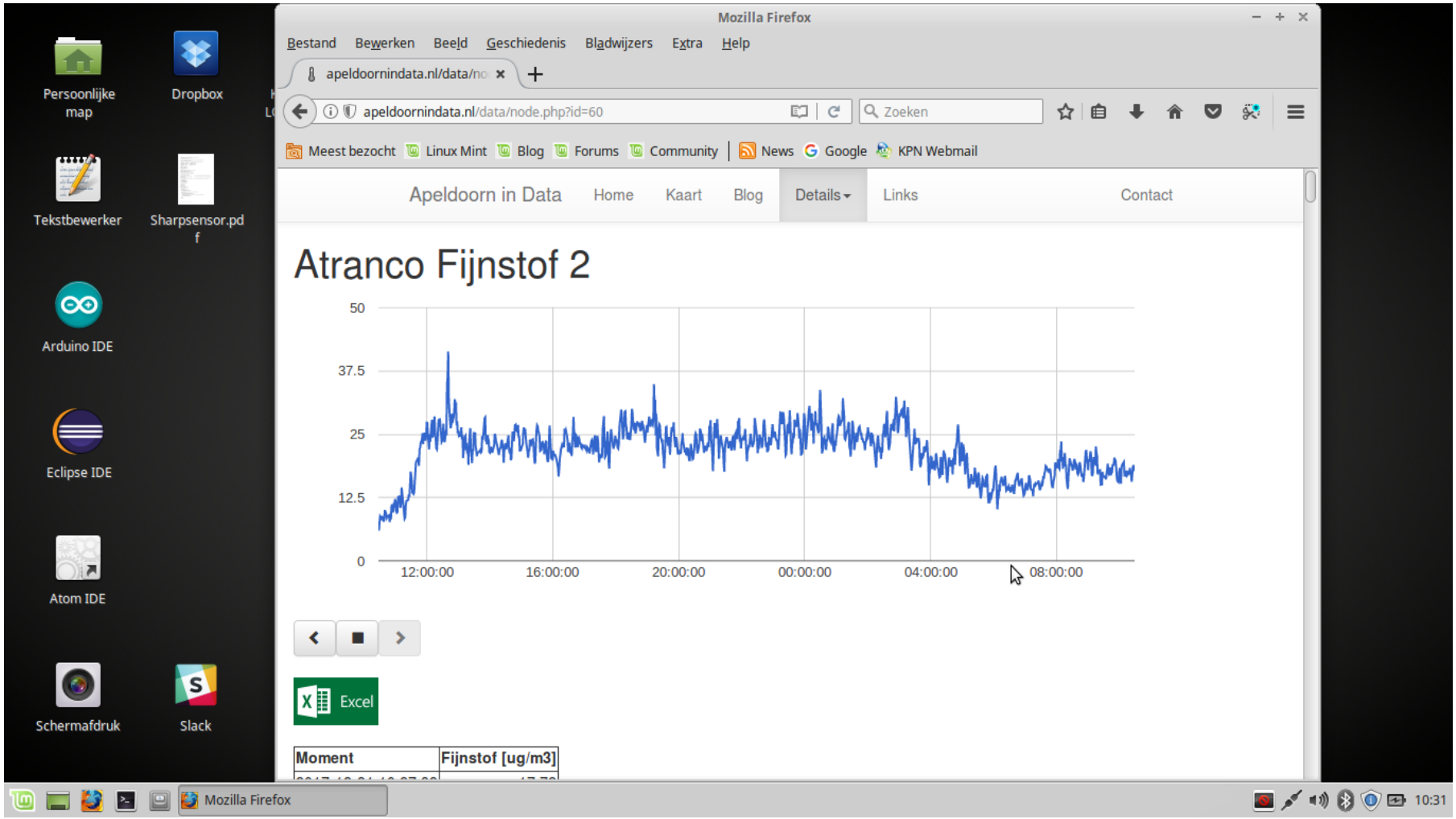
sensor_01v3 dustDensityFilter

```
247  //******* moving average filter
248  //f10 = f9;
249  //f9 = f8;
250  //f8 = f7;
251  //f7 = f6;
252  //f6 = f5;
253  //f5 = f4;
254  //f4 = f3;
255  f3 = f2;
256  f2 = f1;
257  f1 = dustDensity;
258  //dustDensityFilter = (f1+f2+f3+f4+f5+f6+f7+f8+f9+f10)/10; // 10 min gemiddelde
259  //dustDensityFilter = (f1+f2+f3+f4+f5)/5; // 5 min gemidelde
260 dustDensityFilter = (f1+f2+f3)/3; // 3 min gemidelde
261
262 // dustdensity > 0, Cayenne format(16 bit):
263 // -320.00<dust<320.00 (2^15 = 2^5 x 2^10 = 32 k)
264  if (dustDensityFilter > 320) {dustDensityFilter = 320;}
265  if (dustDensityFilter < 0) {dustDensityFilter = 0;}
266
267  Serial.print(" - Dust Density Filter: ");
268  Serial.println(dustDensityFilter);
269 //*****************************************************************************/
```

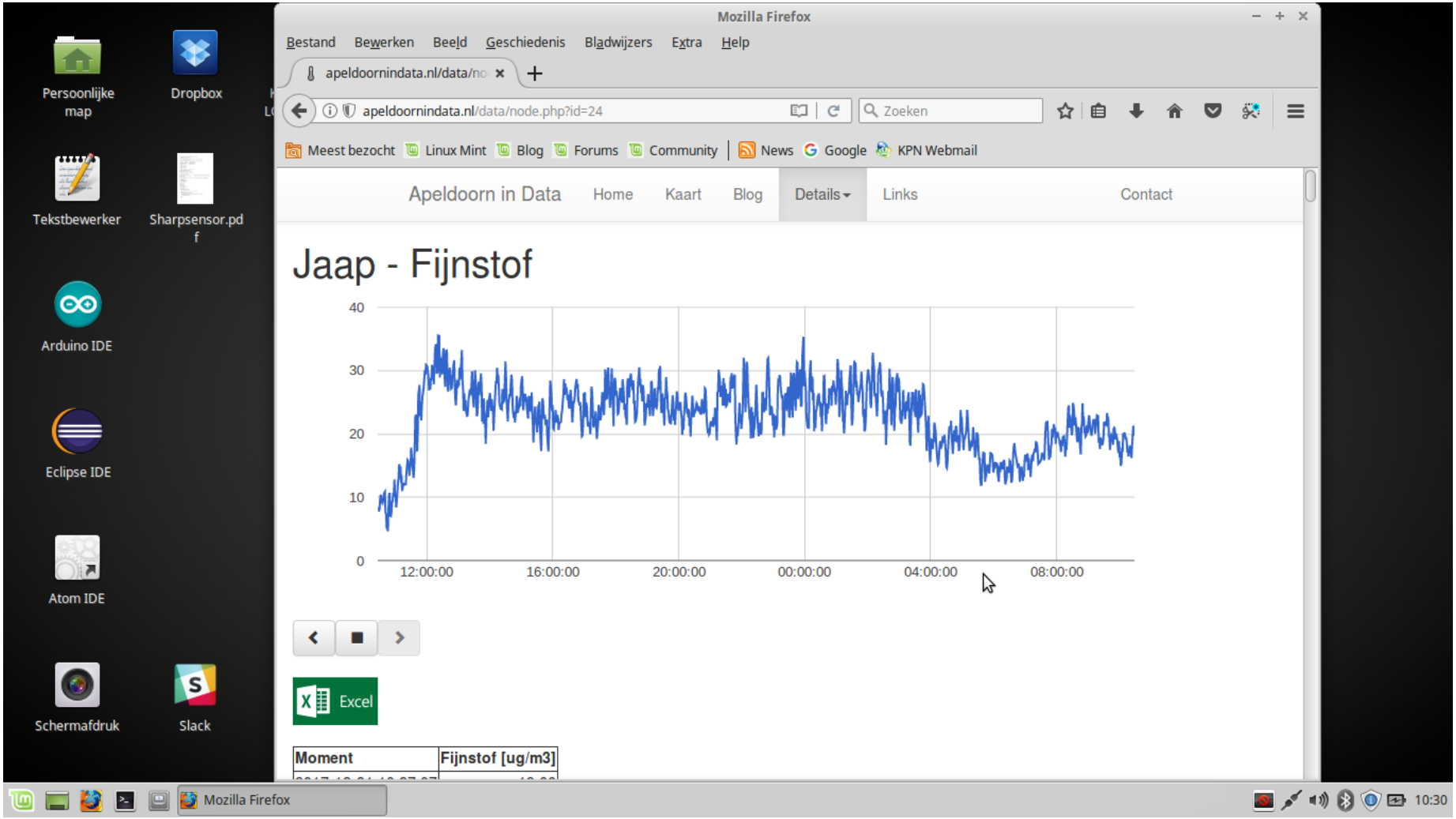
dustsensor_01v3 Cayenne format

```
272       // Compose Cayenne message
273       lpp.reset();    // reset cayenne object
274       lppChannel = 0; // reset channel counter
275
276       // Add sensor values to cayenne data package
277       lpp.addAnalogInput(lppChannel++, dustDensityFilter);
278
279       // The node's GPS location here not required,
280       // but perhaps easier for AiD to place the node at W.P. 17 on the map?)
281       //  lpp.addGPS(lppChannel++,52.23579400,5.95374400, 3);
282
283       // Prepare upstream data transmission at the next possible time.
284       LMIC_setTxData2(APPLICATION_PORT_CAYENNE, lpp.getBuffer(), lpp.getSize(), 0);
285
286       // Float naar character?
287       // dtostrf(calcVoltage, 6, 2, mydata); // mydata for test
288       // Prepare upstream data transmission at the next possible time.
289       // LMIC_setTxData2(1, mydata, sizeof(mydata)-1, 0);
290
291         Serial.println(F("Packet queued"));
292     }
293     // Next TX is scheduled after TX_COMPLETE event.
294 }
295
296 void setup() {
```

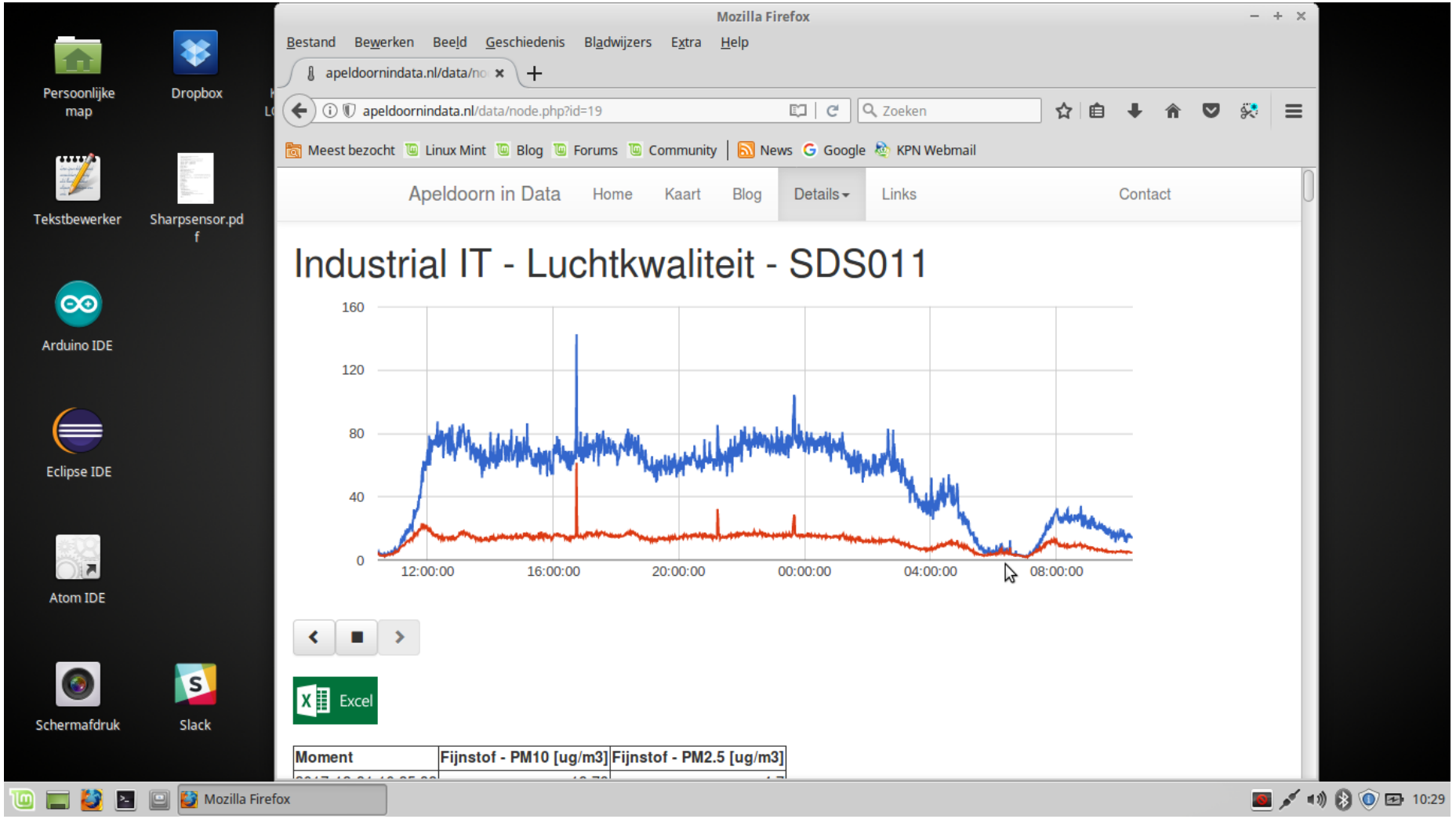
Testmeting december 2017



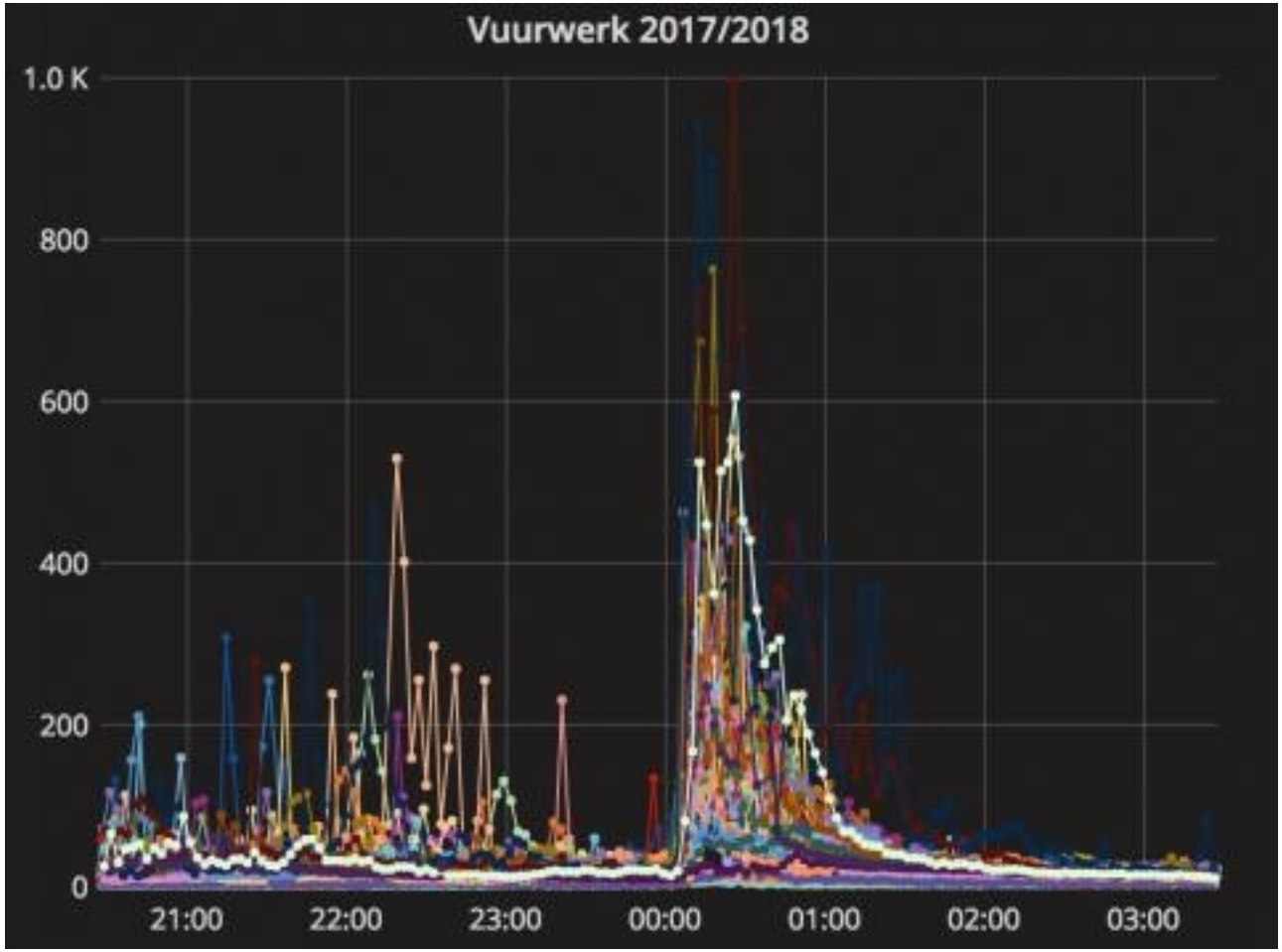
Testmeting december 2017



Testmeting december 2017



Vuurwerpiek Apeldoorn 2017/18



Meetduur

Jaap Stofsensor (3x LiPo’s)

– 2.5 dagen

Meetperiode

Jaap Stofsensor (Netvoeding)

– jan. 2017 tot heden

Nauwkeurigheid

Sharp sensor:

- nauwkerigheid ???
- meet stofcentraties
- laat duidelijk stofpieken zien
- globale kwalitatieve metingen
- correlatie met SDS011
- afhankelijk omgevingstemp en -licht

niet geschikt voor kwantitatieve stofconcentratie-metingen

Behuizing & Plaatsing 1

Behuizing & Plaatsing 2

Behuizing & Plaatsing 3

Behuizing & Plaatsing 4

Behuizing & Plaatsing 5

Aanbevelingen

– geen batterij voeding

– dalende prijzen

– kies betere sensor: SDS 011, Plantower, Honeywell

– andere microcontrollers/boards: ARM Mbed, ESP32, LoPy

Vragen

?