

Probleem

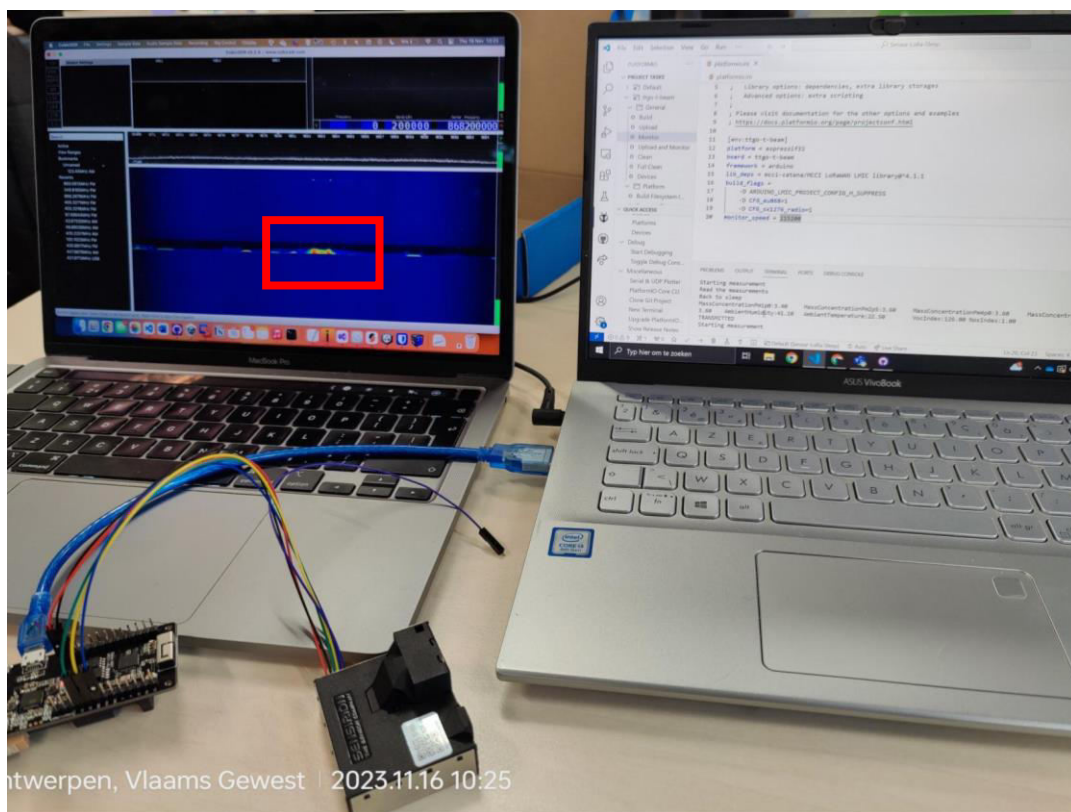
We merken dat bij het testen van de code, de sensor maar 1 keer zijn waardes via LoraWAN doorstuurt, en door een onverklaarbare reden doet hij dit nadien niet meer (bij de volgende uitlezing van de sensor) ... Hiervoor hebben we de volgende troubleshoot manieren gevolgd. Zodat we dit probleem succesvol kunnen oplossen.

Met SDR troubleshooten

Aangezien hij volgens de code wel een lora signaal zou versturen maar deze komt niet aan (in thethingsnetwork) wilde we na gaan of de LoraWAN chip wel werkelijk iets verstuurd als we zenden. Hiervoor hebben we gebruik gemaakt van een SDR met CubeSDR software, zo kunnen we nagaan of het LoraWAN signaal wel in het spectrum verstuurd wordt.

Resultaten van troubleshoot

Bij het uploaden (opstarten) van de sensor krijgen wij 1 pakket binnen die wij ook kunnen zien via een SDR.



Hierna gaat de sensor in slaap en bij het ontwaken doet deze weer een meting en verzend deze niet meer.



TTN

Bij TTN krijgen wij soms het eerste pakketje binnen maar dit resulteert in 00 00 waarde.

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00

↑ 10:40:28 sen55-sensor Forward uplink data message 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ... <> 📄

FPort: 10 Data rate: SF10BW125 SNR: -11 RSSI: -91

Om een random tijd krijgen wij nu 00 00.

↓ 12:28:43	Schedule data downlink for...	DevAddr: 26 08 47 5F	<> 📄	MAC payload: 9C EC 27 8C 07 6F 04 3D 58 09 76 A1 C9 28 59 AC 18 57 F6
↑ 12:28:43	Forward uplink data message	DevAddr: 26 08 47 5F	<> 📄	Payload: { hum: 0, nox: 0, pm10p0: 0, pm1p0: 0, pm2p5: 0, pm4p0
↑ 12:28:43	Successfully processed dat...	DevAddr: 26 08 47 5F	<> 📄	
↓ 12:10:41	Schedule data downlink for...	DevAddr: 26 08 47 5F	<> 📄	MAC payload: 5C 77 64 10 94 4C BE 29 00 A6 C5 C9 E6 C9 91 0E 61 BB EC
↑ 12:10:41	Forward uplink data message	DevAddr: 26 08 47 5F	<> 📄	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ... <> 📄 FPort: 10
↑ 12:10:41	Successfully processed dat...	DevAddr: 26 08 47 5F	<> 📄	
↓ 11:52:38	Schedule data downlink for...	DevAddr: 26 08 47 5F	<> 📄	MAC payload: 22 49 74 D0 FB 27 D1 73 50 B8 25 2F 33 5A 29 8E 50 81 8A
↑ 11:52:38	Forward uplink data message	DevAddr: 26 08 47 5F	<> 📄	Payload: { hum: 0, nox: 0, pm10p0: 0, pm1p0: 0, pm2p5: 0, pm4p0
↑ 11:52:38	Successfully processed dat...	DevAddr: 26 08 47 5F	<> 📄	
↓ 10:18:13	Schedule data downlink for...	DevAddr: 26 08 47 5F	<> 📄	MAC payload: 27 0B 2F 64 AA B8 56 BA 1A 6A E4 EB 0B 01 F3 B5 23 3C 08
↑ 10:18:13	Forward uplink data message	DevAddr: 26 08 47 5F	<> 📄	Payload: { hum: 0, nox: 0, pm10p0: 0, pm1p0: 0, pm2p5: 0, pm4p0

Console

Starting measurement

Read the measurements

Back to sleep

MassConcentrationPm1p0:2.60 MassConcentrationPm2p5:2.80

MassConcentrationPm4p0:2.80	MassConcentrationPm10p0:2.80	AmbientHumidity:32.65
AmbientTemperature:25.47	VocIndex:85.00	NoxIndex:1.00
TRANSMITTED		
Starting measurement		

Oplossing

```
uint32_t pm1p0, pm2p5, pm4p0, pm10p0, hum, temp, voc, nox;
```

```
***
```

```
pm1p0 = massConcentrationPm1p0 * 100;  
pm2p5 = massConcentrationPm2p5 * 100;  
pm4p0 = massConcentrationPm4p0 * 100;  
pm10p0 = massConcentrationPm10p0 * 100;  
hum = ambientHumidity * 100;  
temp = (ambientTemperature + 10) * 100;  
voc = vocIndex * 100;  
nox = noxIndex * 100;
```

```
uint32_t data[8] = { pm1p0, pm2p5, pm4p0, pm10p0, hum, temp, voc, nox };
```

```
for (byte i = 0; i < 8; i++) {  
    //Serial.print(data[i], HEX);  
    txBuffer[0+i*2] = (data[i] >> 8) & 0xFF;  
    txBuffer[1+i*2] = (data[i]) & 0xFF;  
}
```

De lijn waarbij we de waardes invullen in de lijst stond voor dat de waardes erin werden gestoken. Dit resulteerde in lege Txbuffer. Dit is opgelost door de waardes uit te printen en zien tot waar de waardes leeg waren