

Alflex Products B.V.

LRN-010-x-C

Payload specificatie

Versie: 1.0.3

Datum vrijgave: 29-1-2020

Inhoud

Introductie	2
1 Bericht types.....	3
1.1 Report bericht.....	3
1.2 Configuratie berichten.....	3
2 Report bericht	4
2.1 Report bericht data	4
2.1.1 Report header	4
2.1.2 Systeem blok (Blok 0)	4
2.1.3 Event blok (Blok 1).....	5
2.1.4 Accelero blok (Blok 2).....	5
2.1.5 Temperatuur blok (Blok 3)	6
2.1.6 Luchtdruk blok (Blok 4)	6
2.1.7 Licht blok (Block 5)	6
2.1.8 Analooq blok (Block 6).....	6
2.1.9 Luchtvochtigheid blok (Blok 7)	7
2.2 Voorbeeld	7
2.2.1 The Things Network parser	7
3 Report configuratie	8
3.1 format.....	8
3.2 Voorbeeld	9
4 Systeem configuratie bericht	10
4.1 Format downlink.....	10
4.2 Format uplink	10
4.3 Velden.....	10
4.3.1 Measurement interval.....	10
4.3.2 Ack interval.....	10
4.3.3 Retries	11
4.3.4 Version	11
5 Accelerometer configuratie bericht	12
5.1 Format downlink.....	12
5.2 Format uplink	12
5.3 Velden.....	12
5.3.1 Sample rate	12
5.3.2 Axis	13
5.3.3 Threshold.....	13
5.3.4 OR/AND	13
5.3.5 Duration	13
5.3.6 Reference	13
6 Event configuratie bericht	14
6.1 Format downlink.....	14
6.2 Format uplink	14
6.3 Velden.....	14
6.3.1 Report content	14
6.3.2 Event trigger	14
6.3.3 Timeout	15
Bijlage I: The Things Network payload parser	16

Introductie

Voor het lezen van dit document is wat basis kennis vereist van de termen die gebruikt worden bij het LoRaWAN netwerk.

Dit document geeft inzicht in de mogelijkheden van de LoRaNode wat voor data de node stuurt en hoe deze geconfigureerd kan worden.

1 Bericht types

De LoRaNode heeft de mogelijkheid om verschillende berichttypes te kunnen verzenden, de verschillende bericht types worden hieronder beschreven.

1.1 Report bericht

Het report is een uplink bericht dat bestaat uit meetwaardes van de verschillende sensoren. Afhankelijk van hoe de node is ingesteld wordt een report bericht verzonden op basis van tijd of een event.

Standaard zal de node elk uur een bericht verzenden met alle gemeten waardes. Daarnaast zal de node een report bericht verzenden wanneer de node wordt geschud of wanneer een magneet bij de node wordt gehouden of wordt verwijderd.

1.2 Configuratie berichten

Configuratie berichten worden als uplink gebruikt om op de server op de hoogte te houden van de instellingen van de node. Wanneer de node via OTAA joined op het LoRaWAN netwerk wordt een bericht verstuurd met daarin de systeem configuratie van de node. Het is ook mogelijk om de configuratie op te vragen vanaf het LoRaWAN netwerk d.m.v. een downlink bericht.

Verder is het mogelijk om de configuratie van de node aan te passen d.m.v. een downlink bericht. Let hierbij op dat een downlink door het LoRaWAN netwerk alleen verstuurd kan worden nadat de node een uplink bericht heeft verzonden. Dit betekent dat als alle events worden uitgezet en het meetinterval wordt ingesteld op een hoge waarde, het vervolgens lang kan duren voordat de configuratie van de node d.m.v. een downlink kan worden aangepast.

2 Report bericht

Voor het verzenden van meetgegevens via het LoRaWAN netwerk wordt een report bericht door de node samengesteld. Het report bericht zal voorlopig standaard op poort 1 verzonden worden. Het report bericht bestaat uit meerdere blokken die, afhankelijk van de configuratie en de hardware mogelijkheden van de node in-/uitgeschakeld kunnen worden. Dit geeft de mogelijkheid om zelf het bericht te kunnen samenstellen afhankelijk van de gewenste meetwaarden.

Omdat de meetwaarden die de node naar het netwerk verstuurt per node uniek kan zijn worden aan het begin van het pakket twee bytes gebruikt. Deze twee bytes worden als header van het bericht gebruikt om aan te geven welke blokken er in het bericht aanwezig zijn. Wanneer een bit in de eerste twee bytes op '1' staat betekent dat het corresponderende blok zich aan het bericht is toegevoegd. De data van elk blok wordt toegevoegd op volgorde van de bit nummering in de eerste twee bytes. Om het laatste blok te kunnen decoderen moet men ten minste het aantal bytes weten van alle voorgaande velden die aan het bericht zijn toegevoegd.

Een blok kan een of meerdere meetwaarden bevatten. Hoe de blokken zijn opgebouwd wordt in dit hoofdstuk verder uitgewerkt.

2.1 Report bericht data

Het report bericht wordt verzonden op poort 1 met de volgende payload.

2.1.1 Report header

De report header geeft aan welke blokken er actief aan het report bericht zijn toegevoegd.

Byte	MSB	Bit						LSB
	7	6	5	4	3	2	1	0
0	Blok 7	Blok 6	Blok 5	Blok 4	Blok 3	Blok 2	Blok 1	Blok 0
1	Blok 15	Blok 14	Blok 13	Blok 12	Blok 11	Blok 10	Blok 9	Blok 8

2.1.2 Systeem blok (Blok 0)

In het systeem blok wordt de batterij spanning en de temperatuur van de microcontroller verzonden. Zo kan de status van de node gemonitord worden. De meetwaarden hebben een beperkte meetbereik en nauwkeurigheid. Voordat de meetwaarden gebruikt kunnen worden moet er gebruik gemaakt worden van een formule.

Byte	MSB	Bit						LSB
	7	6	5	4	3	2	1	0
0	Batterij spanning							
1	Interne temperatuur							

$$V_{batterij} = \frac{byte[0]}{200} + 2,4$$

$$T_{intern} = \frac{byte[1]}{2} - 30$$

2.1.3 Event blok (Blok 1)

Het event blok wordt standaard alleen in het report bericht gezet als de node een event heeft geregistreerd. Een event kan door meerdere sensoren gegenereerd worden zoals bijvoorbeeld een accelerometer of een wisseling van toestand op de externe input.

Het Event blok bevat informatie met betrekking tot het gegenereerde event.

Byte	MSB	Bit						LSB
	7	6	5	4	3	2	1	0
0				IO2 Event	IO1 event*	NFC event	Accelerometer event	HAL event
1		IO2 status	IO1 status*	HAL status	Event counter[8:11]			
2	Event counter[0:7]							

*Hardware afhankelijk, niet elke node ondersteunt dit type event

Byte [0] wordt hier gebruikt om aan te geven welk event door de node is geregistreerd. Omdat bij LoRaWAN afhankelijk van de netwerkqualiteit het enige tijd kan duren voordat een nieuw event bericht verzonden kan worden, wordt er gebruikt gemaakt van een eventcounter in byte[1] en byte[2]. Het veld 'Event counter' telt het aantal events dat door de node zijn geregistreerd.

In sommige gevallen is alleen het registreren van een event niet genoeg daarom is dit blok uitgebreid met een status van sommige geregistreerde events. Dit kan bijvoorbeeld gebruikt worden om te controleren of een contact gesloten of open is.

2.1.4 Accelerometer blok (Blok 2)

In het accelerometer blok worden de waarden van de accelerometer verzonden deze waarde is een signed 8 bit waarde voor elke as. Wanneer de node stil ligt kan hieruit de oriëntatie van de node worden bepaald.

Byte	MSB	Bit						LSB
	7	6	5	4	3	2	1	0
0	X-as							
1	Y-as							
2	Z-as							

De accelerometer kan een acceleratie van maximaal 2G meten. Hierdoor kan de meetwaarde omgerekend worden naar G-krachten zodat men een indicatie krijgt van wat er met de meetwaarde bedoeld wordt.

$$G_{kracht} = meetwaarde \cdot \frac{2}{128}$$

2.1.5 Temperatuur blok (Blok 3)

In dit blok wordt de temperatuur meetwaarde verzonden. Dit is een signed 16-bit getal.

Byte	MSB	Bit						LSB
	7	6	5	4	3	2	1	0
0	Temperatuur[8:15]							
1	Temperatuur[0:7]							

Om de temperatuur in graden Celsius te krijgen moet de meetwaarde door 100 gedeeld worden.

2.1.6 Luchtdruk blok (Blok 4)

In dit blok wordt de luchtdruk verzonden.

Byte	MSB	Bit						LSB
	7	6	5	4	3	2	1	0
0	Luchtdruk[8:15]							
1	Luchtdruk[0:7]							

Om de luchtdruk in mBar te krijgen moet de meetwaarde door 10 gedeeld worden.

2.1.7 Licht blok (Block 5)

In dit blok worden twee licht waardes verzonden. De eerste waarde ('Licht') bevat zichtbaar en infrarood licht. De tweede waarde ('IR licht') bevat alleen infrarood licht. Deze waardes geven een indicatie van de lichtsterkte.

Byte	MSB	Bit						LSB
	7	6	5	4	3	2	1	0
0	Licht [8:15]							
1	Licht [0:7]							
2	IR licht [8:15]							
3	IR licht [0:7]							

2.1.8 Analooq blok (Block 6)

In dit blok wordt de waarde verstuurd van de analoge ingang.

Byte	MSB	Bit						LSB
	7	6	5	4	3	2	1	0
0	Analoog [8:11]							
1	Analoog [0:7]							

2.1.9 Luchtvochtigheid blok (Blok 7)

In dit blok wordt een waarde verstuurd met de relatieve luchtvochtigheid.

Byte	MSB	Bit						LSB
	7	6	5	4	3	2	1	0
0	Luchtvochtigheid							

Om de relatieve luchtvochtigheid in % te krijgen moet de meetwaarde door 2 gedeeld worden.

2.2 Voorbeeld

Om de werking van het report bericht te laten zien wordt hieronder een voorbeeld weergegeven. Voor elke byte wordt de waarde in hexadecimaal weergegeven.

Byte	Waarde (hex)	Waarde (decimaal)	Blok	Gedecodeerde waarden
0	0x1D	29	Report header	Bij een waarde van 0x1D zijn bits 0,2,3 en 4 een '1'. Dit betekent dat het bericht de volgende blokken bevat: systeem, accelero, temperatuur en luchtdruk.
1	0x00	0		
2	0xE3	227	Systeem (blok 0)	De batterijspanning is $\frac{227}{200} + 2,4 V = 3,535 V$
3	0x6B	107		De microcontroller temperatuur is $\frac{107}{2} - 30 = 23,5 ^\circ C$
4	0xC2	-62	Accelero (blok 2)	De G kracht op de X-as is $-62 \cdot \frac{2 G}{128} = -0,97 G$
5	0x00	0		De G kracht op de Y-as is $0 \cdot \frac{2 G}{128} = 0 G$
6	0x00	0		De G kracht op de Z-as is $0 \cdot \frac{2 G}{128} = 0 G$
7	0x09	2352	Temperatuur (blok 3)	De temperatuur is $\frac{2352}{100} = 23,52 ^\circ C$
8	0x30			
9	0x27	10107	Luchtdruk (blok 4)	De luchtdruk is $\frac{10107}{10} = 1010,7 mBar$
10	0x7B			

2.2.1 The Things Network parser

Om het decoderen van een report bericht makkelijker te maken wordt een voorbeeld van het decoderen in Javascript weergegeven in de bijlage. Deze zou direct gebruikt kunnen worden in de payload parser van The Things Network.

3 Report configuratie

Zoals eerder genoemd is het mogelijk om de inhoud van een report bericht te configureren via een downlink. Als men bijvoorbeeld alleen geïnteresseerd is in de temperatuur kan de node zo geconfigureerd worden dat alleen de temperatuur in het report bericht wordt geplaatst. Deze configuratie van het report bericht werkt alleen voor het reportbericht dat op een vaste tijdsinterval verzonden wordt. Voor het configureren van de blokken die worden verzonden bij een event moet een andere methode gebruikt worden.

Wanneer men voornamelijk geïnteresseerd is in de temperatuur of luchtdruk maar ook eens per dag de oriëntatie van de node wil controleren is dit ook mogelijk. Er kan per blok namelijk een zendinterval ingesteld worden. Het zendinterval van een blok bepaald het aantal meetintervallen waarna het blok aan het report bericht wordt toegevoegd. Als bijvoorbeeld het zendinterval op 1 staat ingesteld wordt het blok op elk meetinterval toegevoegd aan het bericht. Wanneer het zendinterval op 2 wordt gezet wordt het blok elke 2 meetintervallen toegevoegd aan het bericht. Door de instellingen van elk blok te configureren kan de werking van de node op maat gemaakt worden.

Het kan een voordeel geven om meetwaardes waar men niet in is geïnteresseerd uit te zetten door het zendinterval van het blok waarin de meetwaarde in zit op 0 in te stellen dit is met uitzondering van het systeem blok. Het uitschakelen van blokken heeft als gevolg dat het report bericht kleiner wordt. Dit betekent dat de node sneller het bericht via LoRa kan versturen en dus zuiniger kan zijn.

Na het ontvangen van een report configuratie bericht zal de node reageren met een uplink bericht op dezelfde poort. Dit bericht bevat de instellingen van alle blokken die ondersteund worden door de node.

Het systeem blok heeft een uitzondering op het maximale instelbare zendinterval. De maximaal instelbare waarde is afhankelijk van het meetinterval. De node moet zijn systeemstatus eens in de 120 minuten bijwerken. Als het meetinterval staat ingesteld op 120 minuten zal het maximale instelbare zendinterval van het systeemblok '1' zijn. Bij een meetinterval van 60 minuten zal dit '2' zijn. Wanneer het zendinterval van het systeem blok op '0' gezet wordt zal de node automatisch het maximale zendinterval toepassen.

3.1 format

De report configuratie downlink werkt op dezelfde manier als een report uplink. De eerste twee bytes in het bericht worden gebruikt om aan te geven voor welke blokken de zendintervallen bedoeld zijn.

Het report configuratie bericht moet als downlink verzonden worden op poort 2.

Byte	MSB	Bit						LSB
	7	6	5	4	3	2	1	0
0	Blok 7	Blok 6	Blok 5	Blok 4	Blok 3	Blok 2	Blok 1	Blok 0
1	Blok 15	Blok 14	Blok 13	Blok 12	Blok 11	Blok 10	Blok 9	Blok 8
2		Zendinterval van blok x						

3.2 Voorbeeld

In dit voorbeeld wordt het zendinterval van het accelero blok op 0 gezet waardoor de meetwaarde van de accelerometer niet meer verzonden wordt in het report bericht. Ook wordt er met dit bericht ingesteld dat het luchtdruk blok elke 3 meetintervallen verzonden wordt.

Byte	Waarde (hex)	Waarde (decimaal)	Blok	Gedecodeerde waarden
0	0x14	20	Report header	Bij een waarde van 0x14 zijn bits 2 en 4 een '1'.
1	0x00	0		
2	0x00	0	Zendinterval blok 2	Zendinterval is 0, accelerometer waardes worden niet verzonden.
3	0x03	3	Zendinterval blok 4	Zendinterval is 3, luchtdruk waardes worden elke 3 meetintervallen verzonden.

4 Systeem configuratie bericht

De systeem configuratie van de node kan door middel van NFC of een LoRa downlink bericht worden aangepast. Als reactie op een nieuwe configuratie zal de node een uplink bericht versturen met de nieuwe configuratie. Voor het systeem configuratiebericht wordt poort 128 gebruikt.

4.1 Format downlink

Om de systeem configuratie te kunnen wijzigen via een LoRa downlink bericht moet het volgende format gebruikt worden. In de eerste byte wordt met een paar bits aangegeven welke velden van het configuratie bericht door de node toegepast moeten worden. Als bijvoorbeeld alleen het bit 'Schrijf meetinterval' een '1' betekent dit dat de node alleen de waarde van het meetinterval zal wijzigen. Voor het downlink bericht moeten echter wel alle velden gevuld zijn wat het bericht 4 bytes groot maakt.

Na het ontvangen van een downlink zal de node reageren met een uplink bericht wat de huidige configuratie van de node bevat.

Byte	MSB	Bit						LSB
	7	6	5	4	3	2	1	0
0						Write retries	Write ack interval	Write measurement interval
1	Measurement interval							
2	Ack interval							
3						Retries		

4.2 Format uplink

Byte	MSB	Bit						LSB
	7	6	5	4	3	2	1	0
1	Measurement interval (60/0x3C)							
2	Ack interval (6)							
3	Version					Retries (2)		

4.3 Velden

4.3.1 Measurement interval

Het meetinterval is de tijd in minuten tussen metingen, standaard staat deze ingesteld op 60 minuten. Dit betekent dat de node elk uur wakker zal worden om een report bericht te verzenden. De maximale waarde van het meetinterval is 255 minuten, gecombineerd met het minimum zendinterval van het systeem blok zal de node dus elke 255 minuten een bericht moeten sturen.

4.3.2 Ack interval

Het ack interval staat voor het aantal report berichten die op basis van het meetinterval verzonden worden voordat de node een confirmed uplink zal sturen. De standaard waarde is 6 wat betekent dat elke 6 report berichten een confirmed uplink stuurt. Bij een meetinterval van 60 minuten betekent dit dat de node 4 keer per dag een confirmed uplink stuurt.

Bij een confirmed uplink verwacht de node antwoord van het netwerk waarmee wordt aangegeven dat het bericht daadwerkelijk door het netwerk is ontvangen. Let op, als de node 28 confirmed berichten heeft verzonden maar geen reactie heeft gekregen van het netwerk is dit voor de node een indicatie dat het netwerk mogelijk niet naar behoren functioneert. De node zal vervolgens een poging doen om het netwerk opnieuw te joinen d.m.v. OTAA. De node is op dit moment niet meer in staat om downlinks te ontvangen of meetwaardes te verzenden.

4.3.3 Retries

Het aantal pogingen is een instelling die gebruikt wordt bij het verzenden van een confirmed uplink. Wanneer een node niet direct antwoord krijgt op een confirmed uplink zal de node hetzelfde bericht voor de waarde van het aantal pogingen opnieuw verzenden. De minimale waarde die ingesteld kan worden is 2 en de maximale waarde is 7.

4.3.4 Version

Ter indicatie van firmwareversie.

5 Accelerometer configuratie bericht

De accelerometer configuratie van de node kan door middel van een LoRa downlink bericht worden aangepast. Als reactie op een nieuwe configuratie zal de node een uplink bericht versturen met de nieuwe configuratie. Voor het accelerometer configuratiebericht wordt poort 129 gebruikt.

5.1 Format downlink

Om de Accelerometer configuratie te kunnen wijzigen via een LoRa downlink bericht moet het volgende format gebruikt worden. In de eerste byte wordt met een paar bits aangegeven welke velden van het configuratie bericht door de node toegepast moeten worden.

Na het ontvangen van een downlink zal de node reageren met een uplink bericht wat de huidige configuratie van de node bevat.

Byte	MSB	Bit						LSB
	7	6	5	4	3	2	1	0
0			Write OR/AND	Write reference	Write duration	Write threshold	Write axis	Write sample rate
1								Sample rate
2								Axis
3	OR/AND							Threshold
4								Duration
5								Reference

5.2 Format uplink

Byte	MSB	Bit						LSB
	7	6	5	4	3	2	1	0
1								Sample rate (3 = 25Hz)
2								Axis (42 = all axis high)
3	OR/AND(0)							Threshold (96 = 1,5G)
4								Duration (0)
5								Reference (0)

5.3 Velden

5.3.1 Sample rate

Met het sample rate veld kan de snelheid waarmee de accelerometer metingen doet worden aangepast. De accelerometer kan hiermee ook worden uitgezet. Door de sample rate te verhogen kunnen korte snelle schokken door de node gedetecteerd worden, hiermee gaat ook het stroomverbruik omhoog waardoor de levensduur van de batterij wordt verminderd.

Waarde (dec)	Sample rate
0	0 Hz, accelerometer staat uit
1	1 Hz
2	10 Hz
3	25 Hz
4	50 Hz
5	100 Hz

6	200 Hz
7	400 Hz
8	1620 Hz
9	5376 Hz

5.3.2 Axis

Deze instelling maakt het mogelijk om te specificeren op welke as de accelerometer een interrupt moet genereren. Het is mogelijk om de interrupts op de assen te combineren door de decimale waarde op te tellen.

Waarde (dec)	Axis
0	Interrupts uitgeschakeld
1	X low
2	X high
4	Y low
8	Y high
16	Z low
32	Z high

De 'high' staat hier voor een interrupt waarbij de versnelling hoger of lager is dan de ingesteld threshold. De 'low' functie zorgt hier voor een interrupt waarbij de versnelling binnen de threshold valt.

5.3.3 Threshold

Het omslagpunt waarbij een interrupt door de accelerometer zal worden genereerd. In combinatie met de as waarop de interrupt is ingesteld.

5.3.4 OR/AND

Hiermee kan aangegeven worden of er een interrupt gegenereerd moet worden. Wanneer de OR functie aan staat wordt er een interrupt gegenereerd wanneer 1 van de ingestelde assen buiten of binnen de threshold valt. Bij de AND functie wordt er pas een interrupt gegenereerd wanneer alle ingestelde assen buiten of binnen de threshold vallen.

5.3.5 Duration

De duration staat voor het aantal samples dat de waarde binnen of buiten de threshold waarde moet vallen voordat een interrupt gegenereerd wordt.

5.3.6 Reference

De referentie waarde kan gebruikt worden als offset bij een interrupt.

6 Event configuratie bericht

De event configuratie van de node kan door middel van een LoRa downlink bericht worden aangepast. Als reactie op een nieuwe configuratie zal de node een uplink bericht versturen met de nieuwe configuratie. Voor het event configuratiebericht wordt poort 130 gebruikt.

6.1 Format downlink

Om de event configuratie te kunnen wijzigen via een LoRa downlink bericht moet het volgende format gebruikt worden. In de eerste byte wordt met een paar bits aangegeven welke velden van het configuratie bericht door de node toegepast moeten worden.

Na het ontvangen van een downlink zal de node reageren met een uplink bericht wat de huidige configuratie van de node bevat.

Byte	MSB	Bit						LSB
	7	6	5	4	3	2	1	0
0						Write timeout	Write event trigger	Write report content
1	Report content							
2								
3	Event trigger							
4	Timeout							

6.2 Format uplink

Byte	MSB	Bit						LSB
	7	6	5	4	3	2	1	0
0	Report content (0x7F)							
1								
2	Event trigger (0x73)							
3	Timeout (5)							

6.3 Velden

6.3.1 Report content

Het report content veld bepaalt welke informatie via LoRa verzonden wordt wanneer de node een event registreert. Dit werkt op dezelfde manier als het selecteren van de diverse blokken bij een report configuratie downlink. Hier wordt echter geen gebruik gemaakt van een zendinterval. Een blok wordt geselecteerd door het corresponderende bit op '1' te zetten.

Byte	MSB	Bit						LSB
	7	6	5	4	3	2	1	0
1	Blok 7 (0)	Blok 6 (1)	Blok 5 (1)	Blok 4 (1)	Blok 3 (1)	Blok 2 (1)	Blok 1 (1)	Blok 0 (1)
2	Blok 15 (0)	Blok 14 (0)	Blok 13 (0)	Blok 12 (0)	Blok 11 (0)	Blok 10 (0)	Blok 9 (0)	Blok 8 (0)

6.3.2 Event trigger

In het event trigger veld kan er aangegeven welke trigger een event kan genereren waarmee een bericht wordt verzonden.

Byte	MSB	Bit						LSB
	7	6	5	4	3	2	1	0
1	NFC event (0)	Accelero Event (1)	IO2 falling (1)	IO2 rising (1)	(0)	(0)	HAL falling (1)	HAL rising (1)

In dit veld kan voor verschillende events ook gespecificeerd worden op welke toestandsverandering er een event geregistreerd moet worden. Zo kan er bijvoorbeeld gespecificeerd worden dat er alleen een event geregistreerd wordt wanneer een magneet in de buurt van de node komt. Het omgekeerde is ook mogelijk om juist een event te registreren wanneer de magneet uit het bereik van de node gaat. Een combinatie van beide opties is ook mogelijk, deze combinatie staat standaard aan.

Bij het instellen van het NFC of accelerometer event is echter 1 optie mogelijk. Het NFC event staat standaard uitgeschakeld maar kan eventueel gebruikt worden om te detecteren wanneer een telefoon op de node wordt gelegd.

6.3.3 Timeout

De event timeout kan gebruikt worden wanneer een node te veel event berichten naar het netwerk stuurt. Door de timeout hoger te maken zal de node het eerste event gelijk versturen waarna de timeout in gaat. De node zal vervolgens wel events registreren maar de status update naar de status ervan zal pas na de timeout periode verstuurd worden. Voor de timeout wordt een waarde in secondes gehanteerd.

Bijlage I: The Things Network payload parser

```
function Decoder(bytes, port) {
  // Decode an uplink message from a buffer
  // (array) of bytes to an object of fields.
  var decoded = {};
  var tmp = 0;
  var idx = 0;

  if (port == 128)
  {
    tmp = 0;
    decoded.measurementInterval = bytes[idx++];
    decoded.ackInterval = bytes[idx++];
    tmp = bytes[idx++];
    decoded.version = (tmp & 0xF0) >> 4;
    decoded.retries = (tmp & 0x07);
  }

  if (port == 1)
  {
    var blockField = bytes[idx++] | (bytes[idx++] << 8);

    decoded.blockField = blockField;

    if (blockField & (1<<0)){
      // System block
      decoded.vBat = Math.round((bytes[idx+1]/200 + 2.4) * 1000,4) / 1000;
      decoded.iTemp = bytes[idx+1]/2 - 30;
    }
    if (blockField & (1<<1)){
      // Event block
      // Read events
      tmp = bytes[idx++];
      decoded.eventHal = (tmp & (1<<0)) ? true : false;
      decoded.eventAccelero = (tmp & (1<<1)) ? true : false;
      decoded.eventNfc = (tmp & (1<<2)) ? true : false;
      decoded.eventIo1 = (tmp & (1<<3)) ? true : false;
      decoded.eventIo2 = (tmp & (1<<4)) ? true : false;
      // Read states
      tmp = bytes[idx++];
      decoded.stateHal = (tmp & (1<<4)) ? true : false;
      decoded.stateIo1 = (tmp & (1<<5)) ? true : false;
      decoded.stateIo2 = (tmp & (1<<6)) ? true : false;
      decoded.eventCount = ((tmp & 0x0F) << 8) | bytes[idx++];
    }
    if (blockField & (1<<2)){
      // Accelero block
      if (bytes[idx] < 128) {
        decoded.acceleroXAxis = bytes[idx++];
      } else {
        decoded.acceleroXAxis = (bytes[idx++] - 256);
      }
      if (bytes[idx] < 128) {
        decoded.acceleroYAxis = bytes[idx++];
      }
      else {
        decoded.acceleroYAxis = (bytes[idx++] - 256);
      }
      if (bytes[idx] < 128) {
        decoded.acceleroZAxis = bytes[idx++];
      } else {
        decoded.acceleroZAxis = (bytes[idx++] - 256);
      }
    }
    if (blockField & (1<<3)){
      tmp = ((bytes[idx+1] << 8) | bytes[idx+1]);
      if (tmp >= Math.pow(2, 15)) tmp -= Math.pow(2, 16);
      decoded.extTemperature = tmp/100;
    }
    if (blockField & (1<<4)){
      decoded.pressure = ((bytes[idx+1] << 8) | bytes[idx+1])/10;
    }
    if (blockField & (1<<5)){
      decoded.light = ((bytes[idx+1] << 8) | bytes[idx+1]);
      decoded.irLight = ((bytes[idx+1] << 8) | bytes[idx+1]);
    }
    if (blockField & (1<<6)){
      decoded.analogValue = ((bytes[idx+1] << 8) | bytes[idx+1]) * (12.89 / 4095);
    }
    if (blockField & (1<<7)){
      decoded.extHumidity = bytes[idx+1] / 2;
    }
  }
  return decoded;
}
```