**Beep Base handleiding V1.1**

**ID190222-01**

Datum: 4-11-2019

Versie: 1.1

Titel: Beep Base handleiding

# Documentrevisie en verdeling

Documentrevisie:

|  |  |  |
| --- | --- | --- |
| **Versie** | **Datum** | **Wijzigingen** |
| 1.0 | 31-10-2019 | 1e uitgave: Inleiding, Hardware, Software toegevoegd. |
| 1.1 | 4-11-10 | HX711 opdracht aangepast, HX711 meet configuratie toegevoegd, buzzer opdrachten toegevoegd, Programmeren hoofdstuk toegevoegd. |

# Inhoudsopgave

[1 Documentrevisie en verdeling 2](#_Toc23779681)

[2 Inhoudsopgave 3](#_Toc23779682)

[3 Inleiding 5](#_Toc23779683)

[4 Hardware 5](#_Toc23779684)

[4.1 nRF52840 5](#_Toc23779685)

[4.2 AA batterijen 5](#_Toc23779686)

[4.3 DS18B20 temperatuur sensor 5](#_Toc23779687)

[4.4 HX711 rekstrook sensor 6](#_Toc23779688)

[4.5 Reedswitch 6](#_Toc23779689)

[4.6 TPS boostconverter en supply switch 6](#_Toc23779690)

[4.7 Buzzer 6](#_Toc23779691)

[4.8 Flash storage 6](#_Toc23779692)

[4.9 RFM95 6](#_Toc23779693)

[4.10 ATECC608A 6](#_Toc23779694)

[4.11 BME280 6](#_Toc23779695)

[4.12 TLV320ADC3100 6](#_Toc23779696)

[4.13 Logging 7](#_Toc23779697)

[5 Software 7](#_Toc23779698)

[5.1 Beep Protocol 7](#_Toc23779699)

[5.1.1 0d/0x00 – RESPONSE 8](#_Toc23779700)

[5.1.2 1d/0x01 - READ\_FIRMWARE\_VERSION 9](#_Toc23779701)

[5.1.3 2d/0x02 - READ\_HARDWARE\_VERSION 9](#_Toc23779702)

[5.1.4 3d/0x03 - READ\_DS18B20\_STATE 10](#_Toc23779703)

[5.1.5 131d/0x83 - WRITE\_DS18B20\_STATE 10](#_Toc23779704)

[5.1.6 4d/0x04 - READ\_DS18B20\_CONVERSION 11](#_Toc23779705)

[5.1.7 132d/0x84 - WRITE\_DS18B20\_CONVERSION 11](#_Toc23779706)

[5.1.8 9d/0x09 - READ\_HX711\_STATE 12](#_Toc23779707)

[5.1.9 137d/0x89 - WRITE\_HX711\_STATE 13](#_Toc23779708)

[5.1.10 10d/0x0A - READ\_HX711\_CONVERSION 14](#_Toc23779709)

[5.1.11 138d/0x8A - WRITE\_HX711\_CONVERSION 14](#_Toc23779710)

[5.1.12 20d/0x14 - READ\_LORAWAN\_STATE 16](#_Toc23779711)

[5.1.13 145d/0x91 - WRITE\_BUZZER\_DEFAULT\_TUNE 16](#_Toc23779712)

[5.1.14 146d/0x92 - WRITE\_BUZZER\_CUSTOM\_TUNE 17](#_Toc23779713)

[5.1.15 148d/0x94 - WRITE\_LORAWAN\_STATE 17](#_Toc23779714)

[5.1.16 21d/0x15 - READ\_LORAWAN\_DEVEUI 18](#_Toc23779715)

[5.1.17 149d/0x95 - WRITE\_LORAWAN\_DEVEUI 18](#_Toc23779716)

[5.1.18 22d/0x16 - READ\_LORAWAN\_APPEUI 18](#_Toc23779717)

[5.1.19 150d/0x96 - WRITE\_LORAWAN\_APPEUI 19](#_Toc23779718)

[5.1.20 23d/0x17 - READ\_LORAWAN\_APPKEY 19](#_Toc23779719)

[5.1.21 151d/0x97 - WRITE\_LORAWAN\_APPKEY 19](#_Toc23779720)

[5.1.22 136d/0x98 - WRITE\_LORAWAN\_TRANSMIT 19](#_Toc23779721)

[5.1.23 27d/0x1B - READ\_nRF\_ADC\_CONVERSION 20](#_Toc23779722)

[5.1.24 155d/0x9B - WRITE\_nRF\_ADC\_CONVERSION 20](#_Toc23779723)

[5.1.25 29d/0x1D - READ\_APPLICATION\_CONFIG 21](#_Toc23779724)

[5.1.26 157d/0x9D - WRITE\_APPLICATION\_CONFIG 21](#_Toc23779725)

[5.1.27 30d/0x1E - READ\_PINCODE 22](#_Toc23779726)

[5.1.28 158d/0x9E - WRITE\_PINCODE 22](#_Toc23779727)

[5.2 Bluetooth Low Energy 23](#_Toc23779728)

[5.2.1 Pin code 23](#_Toc23779729)

[5.2.2 Device information service 23](#_Toc23779730)

[5.2.3 Battery service 23](#_Toc23779731)

[5.2.4 DFU 24](#_Toc23779732)

[5.2.5 Beep service 24](#_Toc23779733)

[5.3 LoRaWAN 25](#_Toc23779734)

[5.3.1 Standaard bericht types 25](#_Toc23779735)

[6 Programmeren 26](#_Toc23779736)

[6.1 BEEPBASE 26](#_Toc23779737)

[6.2 Programeer script. 26](#_Toc23779738)

[6.3 nRFutil 27](#_Toc23779739)

[6.4 Segger Embedded Studio 27](#_Toc23779740)

[6.5 Applicatie debuggen 27](#_Toc23779741)

[6.6 Compilatie scripts 27](#_Toc23779742)

[7 Elektrisch 28](#_Toc23779743)

# Inleiding

Het Beep meet systeem (Beep Base) is een systeem voor het monitoren van een bijen kast door middel van het gewicht, temperatuur en geluid. Al deze gemeten data wordt gelogd door de Beep Base.

LoRaWAN wordt gebruikt om regelmatig dat te versturen richting het beep back-end, alarmering en om op afstand instellingen te wijzigingen.

Met bluetooth low energy kan de gelogde data worden uitgelezen door de Beep App. De Beep App wordt ook gebruikt voor de initiële configuratie van de Beep Base. Met de App kunnen de instellingen van de sensoren en meet intervallen worden aangepast, maar bijvoorbeeld ook de encryptie sleutels voor LoRaWAN worden aangepast.

# Hardware

Voor Beep is de hardware voor dit project al ontworpen: ID190222. De print heeft de volgende onderdelen:

* nRF52840 BLE low power microcontroller (BMD-340 module).
* 2x AA batterij.
* DS18B20 temperatuur probe sensor met one-wire interface.
* HX711 dubbele weegbrug sensor voor het meten van het gewicht van de bijenkast.
* SQ-SEN-645 Tilt switch: horizontaal en verticaal detecteren.
* Reed switch voor gebruiker activatie
* TPS61292 boost converter.
* TPS22917 supply switch.
* Buzzer voor audio feedback aan de gebruiker.
* Flash voor loggen van meetgegevens.
* RFM95 voor LoRaWAN communicatie.
* ATECC608A encryptie en unieke key voor DEVEUI.
* BME280 temperatuur, luchtvochtigheid en luchtdruk meten.
* TLV320ADC3100 Electret signaal conditioner en recorder voor Fourier analyse.

## nRF52840

De nRF52840 microcontroller van Nordic wordt gebruikt om de functionaliteit van de Beep Base te implementeren. De nRF52840 heeft een radio module die door middel van het SDK van Nordic bluetooth low energy ondersteund.

## AA batterijen

Om de elektronica van energie te voorzien worden er twee lithium AAA batterijen van Energizer in serie gebruikt.

## DS18B20 temperatuur sensor

Om de temperatuur op verschillende plekken in de bijenkast te meten worden er meerdere DS18B20 temperatuur probes gebruikt. Deze sensoren gebruiken een one-Wire protocol om de sensor in te stellen, een temperatuur conversie te starten en het resultaat uit te lezen.

## HX711 rekstrook sensor

De HX711 rekstrook sensor wordt gebruikt om de rekstrook te meten waarop het gewicht van de bijenkast rust. Met het meetresultaat en de gevoeligheid van de rekstrook kan het gewicht van de bijenkast worden berekend.

## Reedswitch

De gebruiker kan door middel van de reedswitch de BLE communicatie activeren. Optioneel wordt de reedswitch ook gebruikt om de pincode te resetten.

## TPS boostconverter en supply switch

De TPS61291 boost converter wordt gebruikt om de batterij spanning te verhogen naar 3V als de batterij spanning lager is. De boost converter kan worden uitgeschakeld waarna de batterij spanning direct aan de uitgang wordt doorgegeven.

Omdat niet alle onderdelen werken of zijn gespecificeerd voor onder de 3V is er nog een voedingsschakelaar gebruikt om die onderdelen van de voedingspanning los te koppelen.

## Buzzer

Als er met BLE parameters worden geschreven of de sensor in een nieuwe oriëntatie wordt geplaats wordt de buzzer gebruikt voor feedback aan de gebruiker. De buzzer zal slechts een aantal tonen/melodieën ondersteunen.

## Flash storage

De MX25R6435 flash storage IC wordt gebruikt om de gemeten gegeven op te slaan. Het flash IC heeft een opslag grote van 64Mb. Met BLE kan dit vervolgens worden uitgelezen met de Beep App.

## RFM95

De RFM95 voor de 868MHz EU band wordt gebruikt voor LoRaWAN communicatie. Een antenne kan worden aangesloten door middel van een micro UFL connector.

## ATECC608A

De ATEC608A is een crypto authenticatie IC die verschillende vormen van encryptie, decryptie, hash berekeningen, een 72 bits unieke serienummer en opslag van sleutels of certificaten ondersteund. Voor de Beep Base wordt echter enkel de unieke serienummer gebruikt om de DEVEUI van af te leiden en de Beep Base hardware-matig te identificeren in het back-end.

## BME280

Temperatuur, luchtvochtigheid en luchtdruk sensor van Bosch die op een 2 meter lange kabel wordt gemonteerd, zodat deze in het bijenhok kan worden geplaatst. Wordt via I2C aangestuurd door middel van de nRF52840.

## TLV320ADC3100

De TLV320ADC3100 is een audio ADC die twee electret microfoons kan voeden en kan uitlezen. De Audio meet data wordt door middel van een I2S interface naar de nRF52840 getransporteerd. Met een I2C interface wordt de audio ADC ingesteld op de juiste ingang en filter responses.

In de nRF52840 wordt de gemeten audio data met een FFT omgezet naar amplitudes in een aantal frequentie banden, wat vervolgens wordt gelogd.

## Logging

Het logging protocol is nog niet gespecificeerd. Naar alle waarschijnlijkheid zal dit een ASCII protocol worden. Optioneel een binaire log in het Flash geheugen en bij het uitlezen vertalen naar een ASCII formaat.

# Software

## Beep Protocol

Het beep protocol is opgebouwd uit complementaire lees en schrijf opdrachten die worden geïdentificeerd door een enkele byte waarvan altijd de zevende bit 1 is voor schrijf opdrachten. Bijvoorbeeld het READ\_DS18B20\_CONVERSION commando met de waarde 4d/0x04h heeft een complementaire schrijf commando 132d/0x84h.

Hieronder een kort overzicht van de gedefineerde commando’s:

|  |  |  |  |
| --- | --- | --- | --- |
| **Dec/hex** | | **Naam** | **Omschrijving** |
| 0 | 0x00 | RESPONSE | Antwoord op een schrijf opdracht |
| 1 | 0x01 | READ\_FIRMWARE\_VERSION | Lees de firmware versie uit |
| 2 | 0x02 | READ\_HARDWARE\_VERSION | Lees de hardware versie uit |
| 3 | 0x03 | READ\_DS18B20\_STATE | Lees de temperatuur resolutie en status uit. |
| 131 | 0x83 | WRITE\_DS18B20\_STATE | Beschrijf de temperatuur resolutie en status. |
| 4 | 0x04 | READ\_DS18B20\_CONVERSION | Lees de laatste temperatuur conversie waardes |
| 132 | 0x84 | WRITE\_DS18B20\_CONVERSION | Start een temperatuur conversie |
| 5 | 0x05 | READ\_DS18B20\_CONFIG | N.A. |
| 6 | 0x06 | READ\_BME280\_STATE | N.A. |
| 7 | 0x07 | READ\_BME280\_CONVERSION | N.A. |
| 8 | 0x08 | READ\_BME280\_I2C | N.A. |
| 9 | 0x09 | READ\_HX711\_STATE | N.A. |
| 10 | 0x0A | READ\_HX711\_CONVERSION | Lees het laatste meet resultaat |
| 138 | 0x8A | WRITE\_HX711\_CONVERSION | Start een nieuwe meting op een kanaal |
| 11 | 0x0B | READ\_AUDIO\_ADC\_STATE | N.A. |
| 12 | 0x0C | READ\_AUDIO\_ADC\_CONVERSION | N.A. |
| 13 | 0x0D | READ\_AUDIO\_ADC\_I2C | N.A. |
| 14 | 0x0E | READ\_ATECC\_READ\_ID | N.A. |
| 15 | 0x0F | READ\_ATECC\_I2C | N.A. |
| 16 | 0x10 | READ\_BUZZER\_STATE | N.A. |
| 17 | 0x11 | READ\_BUZZER\_TUNE | N.A. |
| 18 | 0x12 | READ\_BUZZER\_CUSTOM\_TUNE | N.A. |
| 19 | 0x13 | READ\_SQ\_MIN\_STATE | N.A. |
| 20 | 0x14 | READ\_LORAWAN\_STATE | Lees de LoRaWAN status: aan/uit, joined, duty-cycle, Adaptive Data Rate, correcte sleutels |
| 148 | 0x94 | WRITE\_LORAWAN\_STATE | Schrijf de LoRaWAN status: aan/uit, duty-cycle, Adaptive Data Rate |
| 21 | 0x15 | READ\_LORAWAN\_DEVEUI | Lees de DEVEUI, 8 bytes |
| 149 | 0x95 | WRITE\_LORAWAN\_DEVEUI | Schrijf de DEVEUI, 8 bytes |
| 22 | 0x16 | READ\_LORAWAN\_APPEUI | Lees de APPEUI, 8 bytes |
| 150 | 0x96 | WRITE\_LORAWAN\_APPEUI | Schrijf de APPEUI, 8 bytes |
| 23 | 0x17 | READ\_LORAWAN\_APPKEY | Lees de APPKEY, 16 bytes |
| 151 | 0x97 | WRITE\_LORAWAN\_APPKEY | Schrijf de APPKEY, 16 bytes |
| 136 | 0x88 | WRITE\_LORAWAN\_TRANSMIT | Zend een LoRaWAN bericht |
| 25 | 0x19 | READ\_CID\_nRF\_FLASH | N.A. |
| 26 | 0x1A | READ\_nRF\_ADC\_CONFIG | N.A. |
| 27 | 0x1B | READ\_nRF\_ADC\_CONVERSION | Lees de laatste conversie waardes van de batterij, nRF voedingspanning en batterij percentage. |
| 155 | 0x9B | WRITE\_nRF\_ADC\_CONVERSION | Start een ADC conversie |
| 28 | 0x1C | READ\_APPLICATION\_STATE | N.A. |
| 29 | 0x1D | READ\_APPLICATION\_CONFIG | Lees het meet interval en de verhouding tussen meten en versturen uit. |
| 157 | 0x9D | WRITE\_APPLICATION\_CONFIG | Stel het meet interval en de verhouding tussen meten en versturen in. |
| 30 | 0x1E | READ\_PINCODE | Lees de BLE pin code, 7 – 16 getallen: ‘0’ – ‘9’ |
| 158 | 0x9E | WRITE\_PINCODE | Schrijf de BLE pin code, 7 – 16 getallen: ‘0’ – ‘9’ |
| **Tabel:1** | | | |

Alle commando’s en waardes zijn in big endian.

Voor LoRaWAN is het mogelijk om meerdere opdrachten in een enkel bericht te stoppen, bijvoorbeeld:

|  |  |
| --- | --- |
| **Hex** | **Opdrachten** |
| 0102 | READ\_FIRMWARE\_VERSION, READ\_HARDWARE\_VERSION |

Bij BLE wordt er maar 1 opdracht per bericht uitgevoerd en zal per opdracht een bericht moeten worden verzonden en het eventuele antwoord worden afgevangen.

LoRaWAN maximale buffer grote is 52 bytes en het BLE control point heeft een grote van 30 bytes.

### 0d/0x00 – RESPONSE

Antwoord op een opdracht met een status indicatie van de fout of succes. Wordt enkel door de BEEPBASE verstuurdt

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| RESPONSE | Uint8\_t | 0x00 | RESPONSE opdracht ID |
| command | Uint8\_t | - | Het opdracht ID waarop een antwoord wordt gestuurd |
| Error code | Uint32\_t | - | Zie de tabel hieronder voor de error code en de omschrijving |

nRF SDK Foutcodes:

|  |  |  |
| --- | --- | --- |
| **Error code** | **#** | **Omschrijving** |
| NRF\_SUCCESS | 0 | Successful command |
| NRF\_ERROR\_SVC\_HANDLER\_MISSING | 1 | SVC handler is missing |
| NRF\_ERROR\_SOFTDEVICE\_NOT\_ENABLED | 2 | SoftDevice has not been enabled |
| NRF\_ERROR\_INTERNAL | 3 | Internal Error |
| NRF\_ERROR\_NO\_MEM | 4 | No Memory for operation |
| NRF\_ERROR\_NOT\_FOUND | 5 | Not found |
| NRF\_ERROR\_NOT\_SUPPORTED | 6 | Not supported |
| NRF\_ERROR\_INVALID\_PARAM | 7 | Invalid Parameter |
| NRF\_ERROR\_INVALID\_STATE | 8 | Invalid state, operation disallowed in this state |
| NRF\_ERROR\_INVALID\_LENGTH | 9 | Invalid Length |
| NRF\_ERROR\_INVALID\_FLAGS | 10 | Invalid Flags |
| NRF\_ERROR\_INVALID\_DATA | 11 | Invalid Data |
| NRF\_ERROR\_DATA\_SIZE | 12 | Invalid Data size |
| NRF\_ERROR\_TIMEOUT | 13 | Operation timed out |
| NRF\_ERROR\_NULL | 14 | Null Pointer |
| NRF\_ERROR\_FORBIDDEN | 15 | Forbidden Operation |
| NRF\_ERROR\_INVALID\_ADDR | 16 | Bad Memory Address |
| NRF\_ERROR\_BUSY | 17 | Busy |
| NRF\_ERROR\_CONN\_COUNT | 18 | Maximum connection count exceeded. |
| NRF\_ERROR\_RESOURCES | 19 | Not enough resources for operation |
| **Tabel: 2** | | |

Als een opdracht wordt gestuurd dat nog een aantal extra bytes verwacht, maar er worden te weinig bytes meegestuurd met de opdracht. Dan wordt de error code “Invalid Length” terug gestuurd.

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x96 | WRITE\_LORAWAN\_APPEUI |
| **Antwoord** | 0x009600000009 | RESPONSE voor WRITE\_LORAWAN\_APPEUI, error code: Invalid Length |

Als een onbekend commando wordt gestuurd wordt

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0xFE | Geen gespecificeerde opdracht |
| **Antwoord** | 0x00FE00000005 | RESPONSE voor 0xFE, error code: Not found |

### 1d/0x01 - READ\_FIRMWARE\_VERSION

Met dit commando wordt de firmware versie uitgelezen.

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_FIRMWARE\_VERSION | Uint8\_t | 0x01 | READ\_FIRMWARE\_VERSION opdracht ID |
| Major | Uint16\_t | - | Firmware major number |
| Minor | Uint16\_t | - | Firmware minor number |
| Sub | Uint16\_t | - | Firmware sub number |

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x01 | READ\_FIRMWARE\_VERSION opdracht |
| **Antwoord** | 0x01000000000001 | READ\_FIRMWARE\_VERSION antwoord: firmware versie 0.0.1 |

### 2d/0x02 - READ\_HARDWARE\_VERSION

Met dit commando wordt de Hardware versie en ID nummer uitgelezen.

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_HARDWARE\_VERSION | Uint8\_t | 0x02 | READ\_HARDWARE\_VERSION opdracht ID |
| Major | Uint16\_t | - | Hardware major number |
| Minor | Uint16\_t | - | Hardware minor number |
| ID | Uint32\_t | - | Hardware ID number |

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x02 | READ\_HARDWARE\_VERSION opdracht |
| **Antwoord** | 0x02000100000002E70E | READ\_HARDWARE\_VERSION antwoord: Hardware versie 1.0, ID= 190222 |

Firmware en hardware uitlezen

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x0102 | READ\_FIRMWARE\_VERSION , READ\_HARDWARE\_VERSION opdracht |
| **Antwoord** | 0100000000000102000100000002E70E | READ\_FIRMWARE\_VERSION= 0.0.01  READ\_HARDWARE\_VERSION=1.0, ID= 190222 |

### 3d/0x03 - READ\_DS18B20\_STATE

Lees de temperatuur resolutie en status uit.

Opdracht:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_DS18B20\_STATE | Uint8\_t | 0x03 | READ\_DS18B20\_STATE opdracht ID |

Antwoord2:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_DS18B20\_STATE | Uint8\_t | 0x03 | READ\_DS18B20\_STATE opdracht ID |
| status | Uint8\_t |  | Bit[0] = Aan/Uit: 0=Uit, 1=Aan  Bit [1:3] = Temperatuur Resolutie  1 = 9Bit resolutie  2 = 10Bit resolutie  3 = 11Bit resolutie  4 = 12Bit resolutie  Bit[4:7]= ongebruikt |

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x03 | READ\_DS18B20\_STATE |
| **Antwoord** | 0x0309 | 0b0000 1001 =Aan/uit=1 en resolutie=4:12bit resolutie |

### 131d/0x83 - WRITE\_DS18B20\_STATE

Zet de temperatuur resolutie en status.

Opdracht:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| WRITE\_DS18B20\_STATE | Uint8\_t | 0x83 | WRITE\_DS18B20\_STATE opdracht ID |
| status | Uint8\_t |  | Bit[0] = Aan/Uit: 0=Uit, 1=Aan  Bit [1:3] = Temperatuur Resolutie  1 = 9Bit resolutie  2 = 10Bit resolutie  3 = 11Bit resolutie  4 = 12Bit resolutie  Bit[4:7]= ongebruikt |

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x8309 | WRITE\_DS18B20\_STATE, aan/uit=1, 12-bit resolutie |
| **Antwoord** | 0x008300000000 |  |

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x8305 | WRITE\_DS18B20\_STATE, aan/uit=1, 10-bit resolutie |
| **Antwoord** | 0x008300000000 |  |

### 4d/0x04 - READ\_DS18B20\_CONVERSION

Lees de laatste temperatuur conversie waardes van de aangesloten DS18B20’s.

Opdracht:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_DS18B20\_CONVERSION | Uint8\_t | 0x04 | READ\_DS18B20\_CONVERSION opdracht ID |

Antwoord:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_DS18B20\_CONVERSION | Uint8\_t | 0x04 | READ\_DS18B20\_CONVERSION |
| N | Uint8\_t | < 10 | Aantal DS18B20 sensoren |
| Temperatuur sensor | N \* int16\_t |  | MSB van int16\_t temperatuur in honderdste graden nauwkeurigheid |

Voorbeeld enkele temperatuur sensor:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x04 | READ\_DS18B20\_CONVERSION |
| **Antwoord** | 0x04010898 | READ\_DS18B20\_CONVERSION, 1 DS18B20, temperatuur[0] = 0x0898/2200d = 22.00°C |

Voorbeeld meerdere temperatuur sensoren:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x04 | READ\_DS18B20\_CONVERSION |
| **Antwoord** | 0x0403089809C4FF9C | READ\_DS18B20\_CONVERSION,  3 DS18B20 sensoren  temperatuur[0] = 0x0898/2200d = 22.00°C  temperatuur[1] = 0x09C4/2500d = 25.00°C  temperatuur[2] = 0xFF9C/-10000d = -100.00°C |

Als een gemeten temperatuur op -100.0C staat betekent dit dat de sensor een communicatie fout had tijdens het starten van de conversie of tijdens het uitlezen. Dit kan voorkomen tot nu toe als het soft device bezig is met een actie met hoge prioriteit, bijvoorbeeld het schrijven of lezen van het flash.

### 132d/0x84 - WRITE\_DS18B20\_CONVERSION

Start een temperatuur conversie voor een enkele DS18B20 met een opgegeven index of voor alle temperatuur sensoren.

Opdracht:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| WRITE\_DS18B20\_CONVERSION | Uint8\_t | 0x84 | WRITE\_DS18B20\_CONVERSION opdracht ID |
| Index | Uint8\_t | < 10 | DS18B20 index. Voor waardes onder de 10 wordt er enkel een specifieke sensor gemeten. Voor waardes boven de 10 worden alle sensoren gemeten en uitgelezen. |

Antwoord:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_DS18B20\_CONVERSION/ WRITE\_DS18B20\_CONVERSION | Uint8\_t | 0x04 | READ\_DS18B20\_CONVERSION voor alle temperatuur sensoren of WRITE\_DS18B20\_CONVERSION voor een enkele specifieke sensor |
| Index | Uint8\_t | < 10 of 0xFF | Voor WRITE\_DS18B20\_CONVERSION geeft dit de specifieke sensor waarde aan.  Voor READ\_DS18B20\_CONVERSION is dit het aantal temperatuur sensoren |
| Temperatuur sensor | N \* int16\_t |  | MSB van int16\_t temperatuur in honderdste graden nauwkeurigheid |

Voorbeeld enkele temperatuur sensor:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x8400 | WRITE\_DS18B20\_CONVERSION, start een temperatuur conversie met de temperatuur sensor op index 0. |
| **Antwoord1** | 0x008400000000 | NRF\_SUCCESS, conversie wordt gestart. Dit bericht wordt bij de LoRaWAN interface niet terug gestuurd als de error code NRF\_SUCCES is. |
| **Antwoord2** | 0x04000898 | WRITE\_DS18B20\_CONVERSION, DS18B20 temperatuur sensor op index 0, temperatuur[0] = 0x0898/2200d = 22.00°C |

Voorbeeld temperatuur index = 8, met maar 2 sensoren aangesloten

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x8408 | WRITE\_DS18B20\_CONVERSION, start een temperatuur conversie met de temperatuur sensor op index 8 met slechts 2 sensoren aangesloten. |
| **Antwoord** | 0x008400000007 | NRF\_ERROR\_INVALID\_PARAM voor opdracht WRITE\_DS18B20\_CONVERSION |

Voorbeeld alle temperatuur sensoren:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x84FF | WRITE\_DS18B20\_CONVERSION, start een temperatuur conversie met alle aangesloten temperatuur sensor. |
| **Antwoord1** | 0x008400000000 | NRF\_SUCCESS, conversie wordt gestart. Dit bericht wordt bij de LoRaWAN interface niet terug gestuurd als de error code NRF\_SUCCES is. |
| **Antwoord2** | 0x0403089809C4FF9C | READ\_DS18B20\_CONVERSION,  3 DS18B20 sensoren  temperatuur[0] = 0x0898/2200d = 22.00°C  temperatuur[1] = 0x09C4/2500d = 25.00°C  temperatuur[2] = 0xFF9C/-10000d = -100.00°C |

Als een gemeten temperatuur op -100.0C staat betekent dit dat de sensor een communicatie fout had tijdens het starten van de conversie of tijdens het uitlezen. Dit kan voorkomen tot nu toe als het soft device bezig is met een actie met hoge prioriteit, bijvoorbeeld het schrijven of lezen van het flash.

### 9d/0x09 - READ\_HX711\_STATE

Met de READ\_HX711\_STATE opdracht worden de HX711 kanalen en het aantal samples waarover een gemiddelde wordt berekend uitgelezen vanuit het flash geheugen van de nRF52840. Bij elke meeting op het meet interval worden deze instellingen gebruikt voor de HX711.

Er kunnen meerdere kanalen worden ingesteld waarover een gemiddelde wordt berekent. Voor elk kanaal worden wordt het ingestelde aantal samples gemeten en daarover het gemiddelde berekend.

|  |  |
| --- | --- |
| **Meet kanaal** | **Waarde** |
| CH\_A\_GAIN128 | 0x01 |
| CH\_B\_GAIN32 | 0x02 |
| CH\_A\_GAIN64 | 0x04 |

Bericht opbouw:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_HX711\_STATE | Uint8\_t | 0x09 | READ\_HX711\_STATE opdracht ID |

Antwoord:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_HX711\_STATE | Uint8\_t | 0x89 | WRITE\_HX711\_CONVERSION opdracht ID |
| Meet kanalen | Uint8\_t | 1 - 7 | HX711 meet kanalen, zie bovenstaande tabel |
| Aantal samples | Uint8\_t | >0 | Aantal samples waarover het gemiddelde wordt berekend. |

**Voorbeeld 1:**

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x09 | READ\_HX711\_STATE |
| **Antwoord** | 0x090102 | 0x09=READ\_HX711\_STATE  0x01= CH\_A\_GAIN128  0x02= 2 samples per kanaal |

**Voorbeeld 2:**

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x09 | READ\_HX711\_STATE |
| **Antwoord** | 0x09070A | 0x09=READ\_HX711\_STATE  0x07= CH\_A\_GAIN128, CH\_B\_GAIN32, CH\_A\_GAIN64  0x0A= 10 metingen per kanaal |

### 137d/0x89 - WRITE\_HX711\_STATE

Met de WRITE\_HX711\_STATE opdracht worden de HX711 kanalen en het aantal samples waarover een gemiddelde wordt berekend ingesteld. Bij elke meeting op het meet interval worden deze instellingen gebruikt voor de HX711. Deze gegevens worden opgeslagen in het flash van de nRF52840.

Bericht opbouw:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| WRITE \_HX711\_STATE | Uint8\_t | 0x89 | WRITE\_HX711\_CONVERSION opdracht ID |
| Meet kanalen | Uint8\_t | 1 - 7 | HX711 meet kanalen, zie bovenstaande tabel |
| Aantal samples | Uint8\_t | >0 | Aantal samples waarover het gemiddelde wordt berekend. |

**Voorbeeld:**

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x89070A | 0x89= WRITE\_HX711\_STATE  0x07= CH\_A\_GAIN128, CH\_B\_GAIN32, CH\_A\_GAIN64  0x0A= 10 metingen per kanaal |
| **Antwoord** | 0x008900000000 | 0x00=RESPONSE\_COMMAND  0x89=WRITE\_HX711\_STATE  0x00000000=NRF\_SUCCESS |

### 10d/0x0A - READ\_HX711\_CONVERSION

Lees het laatste meet resultaat met de HX711. Vanaf 1.1 ondersteund de HX711 statemachine het meten van de verschillende kanalen achter een volgend. In plaats van het aantal klok pulsen wordt nu de kanalen door gegeven waarom gemeten is of gemeten moet worden en volgen er meerdere meetresultaten in een enkel bericht. Als een kanaal niet wordt gemeten wordt er geen meetresultaat of 0 waarde mee gestuurd in het resultaat bericht. In de onderstaande tabel zijn de meet kanalen te vinden met de bit waarde voor elk kanaal.

|  |  |
| --- | --- |
| **Meet kanaal** | **Waarde** |
| CH\_A\_GAIN128 | 0x01 |
| CH\_B\_GAIN32 | 0x02 |
| CH\_A\_GAIN64 | 0x04 |

Bericht opbouw:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_HX711\_CONVERSION | Uint8\_t | 0x0A | READ\_HX711\_CONVERSION opdracht ID |

Antwoord:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_HX711\_CONVERSION | Uint8\_t | 0x0A | READ\_HX711\_CONVERSION opdracht ID |
| Meet Kanalen | Uint8\_t | 1 - 7 | HX711 meet kanalen, zie bovenstaande tabel |
| Meet resultaat | Int24\_t | - | Signed meet resultaat CH\_A\_GAIN128, CH\_B\_GAIN32 of CH\_A\_GAIN64 |
| Meet resultaat (optional) | Int24\_t | - | Signed meet resultaat CH\_B\_GAIN32 of CH\_A\_GAIN64 |
| Meet resultaat (optional) | Int24\_t | - | Signed meet resultaat CH\_A\_GAIN64 |

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x0A | READ\_HX711\_CONVERSION |
| **Antwoord** | 0x0A010182E6 | READ\_HX711\_CONVERSION, kanaal 0x01:CH\_A\_GAIN128:, resultaat: 99046/0x0182e6 |

### 138d/0x8A - WRITE\_HX711\_CONVERSION

Start een nieuwe gemidelde meting op de opgegeven kanalen. Als de conversei wordt gestart volgt er eerst een bevestiging van de opdracht of een fout code indien een parameter fout is. Als het meten is voltooid worden de resultaten terug gestuurd.

|  |  |
| --- | --- |
| **Meet kanaal** | **Waarde** |
| CH\_A\_GAIN128 | 0x01 |
| CH\_B\_GAIN32 | 0x02 |
| CH\_A\_GAIN64 | 0x04 |

Opdracht:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_HX711\_CONVERSION | Uint8\_t | 0x8A | WRITE\_HX711\_CONVERSION opdracht ID |
| Meet kanalen | Uint8\_t | 1 - 7 | HX711 meet kanalen, zie bovenstaande tabel |
| Aantal samples | Uint8\_t | >0 | Aantal samples waarover het gemiddelde wordt berekend. |

Antwoord:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_HX711\_CONVERSION | Uint8\_t | 0x8A | WRITE\_HX711\_CONVERSION opdracht ID |
| Meet Kanalen | Uint8\_t | 1 - 7 | HX711 meet kanalen, zie bovenstaande tabel |
| Meet resultaat | Int24\_t | - | Signed meet resultaat CH\_A\_GAIN128, CH\_B\_GAIN32 of CH\_A\_GAIN64 |
| Meet resultaat (optional) | Int24\_t | - | Signed meet resultaat CH\_B\_GAIN32 of CH\_A\_GAIN64 |
| Meet resultaat (optional) | Int24\_t | - | Signed meet resultaat CH\_A\_GAIN64 |

**Voorbeeld 1:**

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x8A010A | WRITE\_HX711\_CONVERSION, CH\_A\_GAIN128, 10 samples |
| **Antwoord1** | 0x008A00000000 | WRITE\_HX711\_CONVERSION, NRF\_SUCCESS. Wordt niet terug gestuurd bij een opdracht vanuit de LoRaWAN interface als de error code gelijk is aan NRF\_SUCCESS. |
| **Antwoord2** | 0x8A010183A4 | 0x8A = WRITE\_HX711\_CONVERSION  0x01 = CH\_A\_GAIN128  0x0183A4 = 99.236decimaal |

**Voorbeeld 2:**

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x8A070A | WRITE\_HX711\_CONVERSION, CH\_A\_GAIN128, CH\_B\_GAIN32, CH\_A\_GAIN64 , 10 samples |
| **Antwoord1** | 0x008A00000000 | WRITE\_HX711\_CONVERSION, NRF\_SUCCESS. Wordt niet terug gestuurd bij een opdracht vanuit de LoRaWAN interface als de error code gelijk is aan NRF\_SUCCESS. |
| **Antwoord2** | 8A030183B90058D1 | 0x8A = WRITE\_HX711\_CONVERSION  0x03 = CH\_A\_GAIN128, CH\_B\_GAIN32  CH\_A\_GAIN128: 99257/0x0183b9  CH\_B\_GAIN32: 22737/0x0058d1 |

**Voorbeeld 3:**

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x8A010A | WRITE\_HX711\_CONVERSION, CH\_A\_GAIN128, CH\_B\_GAIN32, CH\_A\_GAIN64 , 10 samples |
| **Antwoord1** | 0x008A00000000 | WRITE\_HX711\_CONVERSION, NRF\_SUCCESS. Wordt niet terug gestuurd bij een opdracht vanuit de LoRaWAN interface als de error code gelijk is aan NRF\_SUCCESS. |
| **Antwoord2** | 0x8A070183A700564100C1FE | 0x8A = WRITE\_HX711\_CONVERSION  0x03 = CH\_A\_GAIN128, CH\_B\_GAIN32  CH\_A\_GAIN128: 99257/0x0183b9  CH\_B\_GAIN32: 22737/0x0058d1  CH\_A\_GAIN64 : 49662/0x00c1fe |

### 20d/0x14 - READ\_LORAWAN\_STATE

Lees de LoRaWAN status: aan/uit, joined, duty-cycle, Adaptive Data Rate, correcte sleutels.

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_LORAWAN\_STATE | Uint8\_t | 0x14 | READ\_LORAWAN\_STATE opdracht ID |
| Status | Uint8\_t | - | Zie onderstaande status table voor bit waardes. |

|  |  |  |
| --- | --- | --- |
| **Bit** | **Functie** | **Bit waarde** |
| 0 | Aan/uit | 0=LoRaWAN uit, 1=LoRaWAN aan |
| 1 | Joined | 0=Nog niet gejoined, 1=netwerk gejoined |
| 2 | Duty-cycle restrictie | 0=Duty cycle limitatie uit, 1= DutyCycle limitatie aan |
| 3 | Adaptive Datarate | 0=ADR uit, 1= ADR aan. |
| 4 | Sleutels correct | 0=Incorrecte sleutels, 1=Correcte sleutels, |
| 5:7 | Ongebruikt | Altijd 0 |

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x14 | READ\_LORAWAN\_STATE |
| **Antwoord** | 0x141F | 0x1F = 0001 1111b: LoRaWAN aan, netwerk gejoined, DutyCycle limitatie aan, ADR aan, Correcte sleutels |

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x14 | READ\_LORAWAN\_STATE |
| **Antwoord** | 0x141B | 0x1F = 0001 1011b: LoRaWAN aan, netwerk gejoined, DutyCycle limitatie aan, ADR uit, Correcte sleutels |

### 145d/0x91 - WRITE\_BUZZER\_DEFAULT\_TUNE

Met de WRITE\_BUZZER\_DEFAULT\_TUNE opdracht wordt er een standard ingestelde pwm patroon afgespeeld op de BEEPBASE. Tot op heden zijn er maar 2 patronenen, maar dat kan door de klant worden uitgebreid.

Bericht opbouw:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| WRITE\_BUZZER\_DEFAULT\_TUNE | Uint8\_t | 0x91 | WRITE\_BUZZER\_DEFAULT\_TUNE opdracht ID |
| Patroon index | Uint8\_t | 0-1 |  |

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x9101 | 0x91=WRITE\_BUZZER\_DEFAULT\_TUNE  0x01=PWM patroon 1 |
| **Antwoord** | 0x009100000000 | 0x00=RESPONSE\_COMMAND  0x91= WRITE\_BUZZER\_DEFAULT\_TUNE  0x00000000=NRF\_SUCCESS |

### 146d/0x92 - WRITE\_BUZZER\_CUSTOM\_TUNE

Met de WRITE\_BUZZER\_CUSTOM\_TUNE opdracht wordt er een pwm patroon afgespeeld volgens de meegegeven parameters.

Bericht opbouw:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| WRITE\_BUZZER\_CUSTOM\_TUNE | Uint8\_t | 0x92 | WRITE\_BUZZER\_CUSTOM\_TUNE opdracht ID |
| Dutycycle | Uint8\_t | 0-100 | Dutycycle in procenten  1d=1%  100d=100% |
| Frequentie in /100 Hz | Uint8\_t | 1-255 | 1kHz = 10d  2kHz = 20d  2.8kHz = 28d |
| Uit-tijd | Uint16\_t | >0 | Tijd dat de PWM uit is in milliseconden |
| Aan-tijd | Uint16\_t | >0 | Tijd dat de PWM aan is in milliseconden |
| Herhalingen | iint16\_t | >0 | Aantal keer dat de aan-uit cyclus wordt herhaald. |

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x92321C03E801F403 | 0x92= WRITE\_BUZZER\_CUSTOM\_TUNE  0x32=Dutycycle: 50%  0x1C=Frequentie:2.8kHz  0x03E8=Uit-tijd:1000ms  0x01F4=Aan-tijd:500ms  0x03=Herhalingen:3 |
| **Antwoord** | 0x009200000000 | 0x00=RESPONSE\_COMMAND  0x92= WRITE\_BUZZER\_CUSTOM\_TUNE  0x00000000=NRF\_SUCCESS |

### 148d/0x94 - WRITE\_LORAWAN\_STATE

Schrijf de LoRaWAN status: aan/uit, duty-cycle, Adaptive Data Rate.

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| WRITE\_LORAWAN\_STATE | Uint8\_t | 0x94 | WRITE\_LORAWAN\_STATE opdracht ID |
| Status | Uint8\_t | - | Zie onderstaande status table voor bit waardes. |

|  |  |  |
| --- | --- | --- |
| **Bit** | **Functie** | **Bit waarde** |
| 0 | Aan/uit | 0=LoRaWAN uit, 1=LoRaWAN aan/reset  Als deze bit 1 is wordt de LoRaWAN stack gereset en worden de sleutels en instellingen opnieuw geladen vanuit het flash. Voor LORaWAN communicatie betekend dit dat de sensor eerst opnieuw aanmeld op het LoRaWAN netwerk en er geen antwoord komt via de LoRaWAN interface. |
| 1 | Ongebruikt | Wordt genegeerd |
| 2 | Duty-cycle restrictie | 0=Duty cycle limitatie uit, 1= DutyCycle limitatie aan |
| 3 | Adaptive Datarate | 0=ADR uit, 1= ADR aan. |
| 4:7 | Ongebruikt | Altijd 0. Wordt genegeerd |

Voorbeeld om de LoRaWAN stack te resetten:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 940D | 0x0F = 0000 1101b: LoRaWAN aan, DutyCycle limitatie aan, ADR aan, |
| **Antwoord** | 009400000000 | NRF\_SUCCESS voor WRITE\_LORAWAN\_STATE |

Voorbeeld om de LoRaWAN testen zonder Duty-cycle limitatie:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 9409 | 0x0F = 0000 1001b: LoRaWAN aan, DutyCycle limitatie uit, ADR aan, |
| **Antwoord** | 009400000000 | NRF\_SUCCESS voor WRITE\_LORAWAN\_STATE |

### 21d/0x15 - READ\_LORAWAN\_DEVEUI

Lees de DEVEUI, 8 bytes

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_LORAWAN\_DEVEUI | Uint8\_t | 0x15 | READ\_LORAWAN\_DEVEUI opdracht ID |

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x15 | READ\_LORAWAN\_DEVEUI opdracht |
| **Antwoord** | 0x150001020304050607 | READ\_LORAWAN\_DEVEUI antwoord, DEVEUI: 01020304050607 |

### 149d/0x95 - WRITE\_LORAWAN\_DEVEUI

Schrijf de DEVEUI, 8 bytes

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| WRITE\_LORAWAN\_DEVEUI | Uint8\_t | 0x17 | WRITE\_LORAWAN\_DEVEUI opdracht ID |
| DEVEUI | 8 x uint8\_t | - |  |

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x950001020304050607 | WRITE\_LORAWAN\_DEVEUI opdracht |
| **Antwoord** | 0x009500000000 | WRITE\_LORAWAN\_DEVEUI succesvol. |

### 22d/0x16 - READ\_LORAWAN\_APPEUI

Lees de APPEUI, 8 bytes

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_LORAWAN\_APPEUI | Uint8\_t | 0x16 | READ\_LORAWAN\_APPKEY opdracht ID |

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x16 | READ\_LORAWAN\_APPEUI opdracht |
| **Antwoord** | 0x160001020304050607 | READ\_LORAWAN\_APPEUI succesvol. |

### 150d/0x96 - WRITE\_LORAWAN\_APPEUI

Met deze opdracht wordt de APPEUI ingesteld. De APPEUImoet 8 bytes lang zijn

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| WRITE\_LORAWAN\_APPEUI | Uint8\_t | 0x96 | WRITE\_LORAWAN\_APPEUI opdracht ID |
| AppEUI | 8 x Uint8\_t |  |  |

Als het schrijven succesvol is, wordt dit door middel van een response met NRF\_SUCCESS weer gegeven.

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x960001020304050607 | WRITE\_LORAWAN\_APPEUI opdracht |
| **Antwoord** | 0x009600000000 | WRITE\_LORAWAN\_APPEUI succesvol. |

### 23d/0x17 - READ\_LORAWAN\_APPKEY

Lees de APPKEY, 16 bytes

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_LORAWAN\_APPKEY | Uint8\_t | 0x17 | READ\_LORAWAN\_APPKEY opdracht ID |

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x17 | READ\_LORAWAN\_APPKEY opdracht |
| **Antwoord** | 0x17000102030405060708090A0B0C0D0E0F | READ\_LORAWAN\_APPKEY succesvol. |

### 151d/0x97 - WRITE\_LORAWAN\_APPKEY

Schrijf de APPKEY, 16 bytes

Met deze opdracht wordt de APPKEY ingesteld. De APPKEY moet 16 bytes lang zijn

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| WRITE\_LORAWAN\_APPKEY | Uint8\_t | 0x97 | WRITE\_LORAWAN\_APPKEY opdracht ID |
| Appkey | 16 x Uint8\_t | 0 - 9 |  |

Als het schrijven succesvol is, wordt dit door middel van een response met NRF\_SUCCESS weer gegeven.

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x97000102030405060708090A0B0C0D0E0F | WRITE\_LORAWAN\_APPKEY opdracht |
| **Antwoord** | 0x009700000000 | WRITE\_LORAWAN\_APPKEY succesvol. |

### 136d/0x98 - WRITE\_LORAWAN\_TRANSMIT

Zend een LoRaWAN bericht met de gegeven payload op fport 5.

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| WRITE\_LORAWAN\_APPKEY | Uint8\_t | 0x97 | WRITE\_LORAWAN\_APPKEY opdracht ID |
| Lengte | Uint8\_t | < 28 | Maximale grote van 28 bytes |
| Payload | N x uint8\_t |  | Maximum payload van 28 Bytes, aangezien de control point een maximum grote heeft van 30 bytes. |

Als het schrijven succesvol is, wordt dit door middel van een response met NRF\_SUCCESS weer gegeven.

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x9810000102030405060708090A0B0C0D0E0F | WRITE\_LORAWAN\_APPKEY opdracht met een lengte van 16 bytes. |
| **Antwoord** | 0x009800000000 | WRITE\_LORAWAN\_APPKEY succesvol. Wordt enkel gestuurd wanneer de communicatie interface BLE is. |

### 27d/0x1B - READ\_nRF\_ADC\_CONVERSION

Lees de laatste conversie waardes van de batterij, nRF voedingspanning en batterij percentage.

Opdracht:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_nRF\_ADC\_CONVERSION | Uint8\_t | 0x1B | READ\_nRF\_ADC\_CONVERSION opdracht ID |

Antwoord:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_nRF\_ADC\_CONVERSION | Uint8\_t | 0x1B | READ\_nRF\_ADC\_CONVERSION opdracht ID |
| Batterij spanning | Uint16\_t |  | Batterij spanning in millivolts |
| Voedingspanning | Uint16\_t |  | nRF52840 voeding spanning in millivolts |
| Batterij percentage | Uint8\_t | 0-100% | Batterij percentage, zie batterij spanning hoofdstuk voor de berekening. |

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 1B |  |
| **Antwoord** | 1B0B530BB85C | Batterij spanning 0x0B53=2899mV, nRF spanning=3000mV, Batterij percentage 92% |

### 155d/0x9B - WRITE\_nRF\_ADC\_CONVERSION

Start een ADC conversie.

Opdracht:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| WRITE\_nRF\_ADC\_CONVERSION | Uint8\_t | 0x9B | WRITE\_nRF\_ADC\_CONVERSION opdracht ID |

Antwoord:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| WRITE\_nRF\_ADC\_CONVERSION | Uint8\_t | 0x9B | WRITE\_nRF\_ADC\_CONVERSION opdracht ID |
| Batterij spanning | Uint16\_t |  | Batterij spanning in millivolts |
| Voedingspanning | Uint16\_t |  | nRF52840 voeding spanning in millivolts |
| Batterij percentage | Uint8\_t | 0-100% | Batterij percentage, zie batterij spanning hoofdstuk voor de berekening. |

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x9B | WRITE\_nRF\_ADC\_CONVERSION, start conversie |
| **Antwoord** | 0x9B0B530BB85C | Batterij spanning 0x0B53=2899mV, nRF spanning=3000mV, Batterij percentage 92% |

### 29d/0x1D - READ\_APPLICATION\_CONFIG

Lees het meet interval in minuten en de verhouding tussen het aantal sensor metingen en LoRaWAN berichten uit.

Opdracht:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_APPLICATION\_CONFIG | Uint8\_t | 0x1D | READ\_APPLICATION\_CONFIG opdracht ID |

Antwoord:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| READ\_APPLICATION\_CONFIG | Uint8\_t | 0x1B | READ\_APPLICATION\_CONFIG opdracht ID |
| Ratio meetingen-zenden | Uint8\_t | - | De verhouding tussen het aantal metingen en de LoRaWAN berichten die worden verzonden. Als dit getal drie is, wordt een op de drie metingen met LoRaWAN verstuurt, mits de duty-cycle dit niet beperkt. |
| Interval | Uint16\_t | >0 | Het meetinterval in minuten. |

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x1D | READ\_APPLICATION\_CONFIG |
| **Antwoord** | 0x1D03000A | Ratio van 1:3 voor meten en verzenden. 1 op de drie metingen wordt met LoRaWAN verstuurdt. Meet interval is 0x000A/10d minuten. |

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x1D | READ\_APPLICATION\_CONFIG |
| **Antwoord** | 0x1D000001 | Alle metingingen worden verzonden.. Meet interval is 0x0001/1d minuut. |

### 157d/0x9D - WRITE\_APPLICATION\_CONFIG

Stel het meet interval en de verhouding tussen meten en versturen in.

Opdracht:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| WRITE\_APPLICATION\_CONFIG | Uint8\_t | 0x9D | WRITE\_APPLICATION\_CONFIG opdracht ID |

Antwoord:

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| WRITE\_APPLICATION\_CONFIG | Uint8\_t | 0x9D | WRITE\_APPLICATION\_CONFIG opdracht ID |
| Ratio meetingen-zenden | Uint8\_t | - | De verhouding tussen het aantal metingen en de LoRaWAN berichten die worden verzonden. Als dit getal drie is, wordt een op de drie metingen met LoRaWAN verstuurt, mits de duty-cycle dit niet beperkt. |
| Interval | Uint16\_t | >0 && < 1440 | Het meetinterval in minuten. Moet minimaal 1 zijn en maximaal 1440. Anders errorcode NRF\_ERROR\_INVALID\_PARAM |

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x9D03000A | Ratio van 1:3 voor meten en verzenden. 1 op de drie metingen wordt met LoRaWAN verstuurd. Meet interval is 0x000A/10d minuten. |
| **Antwoord** | 0x009D00000000 | NRF\_SUCCES voor WRITE\_APPLICATION\_CONFIG |

Voorbeeld met incorrecte sample interval:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x9D03FFFF | Ratio van 1:3 voor meten en verzenden. 1 op de drie metingen wordt met LoRaWAN verstuurd. Meet interval is 0xFFFF/10d minuten. |
| **Antwoord** | 0x009D00000007 | NRF\_ERROR\_INVALID\_PARAM |

Testen:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x9D000001 | Ratio van 1:0 voor meten en verzenden. Alle metingen worden met LoRaWAN verstuurd. Meet interval is 0x0001/65535d minuten, wat hoger is dan het maximum interval van 1440 minuten. |
| **Antwoord** | 0x009D00000000 | NRF\_SUCCES voor WRITE\_APPLICATION\_CONFIG |

### 30d/0x1E - READ\_PINCODE

Lees de BLE pin code, 7 – 16 getallen: ‘0’ – ‘9’

Met deze opdracht wordt de BLE pin code uitgelezen. Het antwoord is opgebouwd volgens het WRITE\_PINCODE opdracht, maar dan met de READ\_PINCODE command

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| WRITE\_PINCODE | Uint8\_t | 0x1E | WRITE\_PINCODE opdracht ID |

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x1E | READ\_PINCODE opdracht |
| **Antwoord** | 0x1E083031323334353637 | READ\_PINCODE antwoord, pincode 8 bytes: “01234567” |

### 158d/0x9E - WRITE\_PINCODE

Met deze opdracht wordt de BLE pin code ingesteld. De pin code moet tussen 7 en 16 getallen bevatten met ASCII karakters tussen 0(0x30h) en 9(0x39h).

|  |  |  |  |
| --- | --- | --- | --- |
| **Veld** | **Grote** | **Waarde** | **Omschrijving** |
| WRITE\_PINCODE | Uint8\_t | 0x9E | WRITE\_PINCODE opdracht ID |
| lengte | Uint8\_t | 7 - 16 | Aantal bytes van de pincode moet 7 tot en met 16 zijn, anders errorcode: NRF\_ERROR\_INVALID\_LENGHT |
| pincode | 7-16 x Uint8\_t | 0 - 9 | Byte waardes moeten tussen de 0x30 en 0x39 zijn, anders errorcode: NRF\_ERROR\_INVALID\_DATA |

Als het schrijven succesvol is, wordt dit door middel van een response met NRF\_SUCCESS weer gegeven.

Voorbeeld:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x9E083039303930353035 | WRITE\_PINCODE opdracht: “09090505” |
| **Antwoord** | 0x009E00000000 | WRITE\_PINCODE succesvol. |

Voorbeeld van fout bericht:

|  |  |  |
| --- | --- | --- |
|  | **Hex bericht** | **Inhoud** |
| **Opdracht** | 0x9E080039303930353035 | WRITE\_PINCODE opdracht: “/09090505” |
| **Antwoord** | 0x009E0000000B | WRITE\_PINCODE error: NRF\_ERROR\_INVALID\_DATA. |

## Bluetooth Low Energy

De nRF52840 zal de volgende services ondersteunen:

1. DIS: Device information service.
2. BAS: Battery service.
3. BEEP unieke service
4. Log uitlezen.
5. DFU: Nordic’s firmware update service

Het apparaat zal zich adverteren als BEEPXXXXXXXX (Voorbeeld). Waarbij de laatste 8 karakters de 4 minst belangrijke karakters zijn van het DEVEUI in hexadecimaal. Als het DEVEUI bijvoorbeeld 0x01 23 45 67 89 AB CD EF is, dan is de BLE advertentie naam BEEP89ABCDEF.

Het DEVEUI wordt afgeleid van het unieke ID van de ATECC608A.

### Pin code

Is nog niet geactiveerd in firmware.

### Device information service

De device information service ondersteund de volgende karakteristieken met de volgende waardes:

* Manufacturer Name String: “BEEP”
* Model Number String: “BEEPBASE”
* Serial number String: “TODO:ATTEC”
* Hardware Revision String: “1.0”
* Firmware Revision String: “0.0.1”

### Battery service

De Batterij service (BAS) geeft een grove geschatte batterij percentage aan. Het percentage is slechts een indicatie, aangezien batterijen een zeer sterke temperatuur en stroomverbruik afhankelijkheid hebben.

De batterij percentage wordt berekend over de som van 10 ADC metingen waarmee een gemiddelde batterij spanning in mV wordt berekend. Aan de hand van de nominale batterij spanning en de cutt-off batterij spanning wordt lineair een batterij percentage berekend.

Voorbeeld:

De percentages worden afgerond op hele procenten, dus het batterij percentage wordt dan 92%

### DFU

Nog niet geïmplementeerd, deze vereist altijd een bootloader met SDK15.3.

### Beep service

De beep service heeft de volgende UUID:

|  |  |
| --- | --- |
| UUID: | 1bc3f8c5-ebc6-4050-ad4b-9f71d4a647be |
| Hex UUID | {0x1b, 0xc3, 0xf8, 0xc5, 0xeb, 0xc6, 0x40, 0x50, 0xad, 0x4b, 0x9f, 0x71, 0xd4, 0xa6, 0x47, 0xbe} |

De volgende short UUID worden gebruikt voor de karakteristieken in de service:

|  |  |  |
| --- | --- | --- |
| **Karakteristiek** | **Short UUID** | **Long UUID** |
| Beep Service | 0x68A1 | BE4768A1-719F-4BAD-5040-C6EBC5F8C31B |
| DS18B20 meetresultaat | 0x68A2 | BE4768A2-719F-4BAD-5040-C6EBC5F8C31B |
| Beep Control Point | 0x68B0 | 000068B0-0000-1000-8000-00805F9B34FB |

Let op! NRF Connect geeft de Long UUIDs weer in omgekeerde volgorde!

#### DS18B20 temperatuur resultaat karakteristiek

De DS18B20 temperatuur resultaat karakteristiek geeft de laatste temperatuur meting weer. Net als in het Beep protocol is de eerste byte het aantal sensoren en volgt er daarna een int16\_t voor elke sensor met de temperatuur in honderdste graden nauwkeurigheid. Dus om de temperatuur om te rekenen naar graden Celcius moet het temperatuur getal worden gedeeld door honderd.

|  |  |
| --- | --- |
| Byte | Inhoud |
| 0 | Aantal DS18B20 sensoren |
| (N \* 2) + 0 | MSB van int16\_t temperatuur in honderdste graden nauwkeurigheid |
| (N \* 2) + 1 | LSB van int16\_t temperatuur in honderdste graden nauwkeurigheid |

Bijvoorbeeld:

|  |
| --- |
| 01-08-98 |

Byte 0 is 1, dus maar 1 temperatuur sensoren.

De temperatuur waarde is 0x0898/2200d, maar moet geïnterpreteerd worden als int16\_t voor signedness. Door de waarde van 2200 te delen door honderd worden de temperatuur in graden berekend: 2200 / 100 = 22.00°C.

#### Beep Control point

Het control point ondersteund het beep protocol, maar kan in tegenstelling tot LoRaWAN slechts een commando per keer aan. Als er geen notificaties aan staan worden gestuurde commando’s wel uitgevoerd, maar wordt het antwoord nooit ontvangen door de zender.

#### Log uitlezen

Nog niet geïmplementeerd.

## LoRaWAN

De LoRaWAN stack zal zodra de BEEPBASE in een horizontale positie is beginnen met het initialiseren van de hardware en de LoRaWAN sleutels ophalen. Als LoRaWAN is uitgeschakeld door middel van het BEEP protocol of als een van de sleutels invalide is blijft de LoRaWAN communicatie uitgeschakeld.

De LoRaWAN sleutels zijn invalide als de gehele DEVEUI, APPKEY of APPEUI 0x00 of 0xFF is. Ook al is LoRaWAN aangezet via het BEEP protocol, als een van de sleutels incorrect is dan gaat de LoRaWAN stack naar een uit stand.

Als er nieuwe LoRaWAN sleutels via het BEEP protocol worden ingesteld moet de LoRaWAN stack gereset worden, zodat de nieuwe sleutels worden geladen. Totdat dit wordt uitgevoerd gebruikt de LoRaWAN stack de oude sleutels.

Als LoRaWAN opstart omdat het via het BEEP protocol is gereset of als de BEEPBASE in een horizontale positie wordt gelegd, dan gaat deze zich aan proberen te melden bij het back-end met de aangeleverde sleutels. De LoRaWAN stack gaat dan Join Request berichten versturen. Als er een gateway binnen het bereik van de BEEPBASE is en de mote is aangemeld bij het back-end, dan krijgt de BEEPBASE een Join Accept bericht terug.

Zodra de BEEPBASE is aangemeld bij het back-end, gaat deze eerst een bericht versturen met de firmware en hardware versie en het unieke ID van de ATECC. Als het back-end dit bericht mist, kan er met een downlink altijd achterhaald worden wat de firmware en hardware versies zijn.

Als er een downlink bericht wordt ontvangen wordt de payload gecontroleerd met het BEEP protocol. Indien er valide commando’s zijn worden deze uitgevoerd. Eventuele antwoorden worden gebufferd door de LoRaWAN stack en verzonden met het eerste volgende uplink bericht.

### Standaard bericht types

De volgende bericht types zijn gedefinieerd en worden aangegeven door de Fport waarde.

|  |  |  |
| --- | --- | --- |
| **Bericht type** | **Fport** | **Bevat Beep Protocol veld** |
| Sensor on | 1 | READ\_FIRMWARE\_VERSION, READ\_FIRMWARE\_VERSION |
| Keep alive | 3 | READ\_nRF\_ADC\_CONVERSION, READ\_HX711\_CONVERSION, READ\_DS18B20\_CONVERSION |
| Alarm | 4 | Nog niet geïmplementeerd |
| Uplink custom | 5 | Uplink payload wordt gespecificeerd door het beep protocol commando. |
| Downlink response | 6 | Bevat het antwoord op een downlink beep commando. |

# Programmeren

## BEEPBASE

De Beepbase moet na assemblage worden geprogrameerd. Vanuit Ideetron worden zip en hex files aangeleverd per release. De Hex file is voor het programmeren met een programmer, bijvoorbeeld een ARM flasher of een nRF528xx development board. De zip bestanden zijn voor firmware update via BLE, bijvoorbeeld de nRF Connect desktop of telefoon App.

Om de zip en hex files te generen wordt een batch file gebruikt die de bootloader, applicatie en bootloader settings onder “Release” configuratie compileert en samenvoegt.

Voor de zip files wordt er ook nog gebruik gemaakt van encryptie door middel van nrfutil.exe, zodat enkel firmware die met dezelfde sleutels is gecompileerd als de bootloader geaccepteerd wordt door de bootloader.

Er worden altijd twee zip files aangeleverd: enkel de applicatie en bootloader, softdevice en applicatie in een. Die laatste is erg handig tijdens ontwikkeling, zodat altijd alle firmware compatible is met elkaar en niet dat een oude bootloader een nieuwere applicatie niet kan laden. Hiervoor is wel de bootloader en applicatie versie nummer controle uitgeschakeld, omdat die anders niet accepteert dat bootloader of applicatie firmware met hetzelfde versie nummer wordt overschreven. Als dit niet is uitgeschakeld weigert de bootloader de complete firmware update.

Voor de uiteindelijk product release is het aan de klant of dat de versie nummer controle weer ingeschakeld moet worden. Het voegt namelijk wel wat extra eisen en controle toe aan firmware releases en de firmware kan niet meer ge-downgrade worden naar een vorige versie.

## Programeer script.

Om de beepbase met de aangeleverde hex file te programeren wordt de volgende batch file gebruikt:

|  |
| --- |
| @ECHO OFF  SET hw\_major=1  SET hw\_minor=0  SET hw\_ID=190222  SET /A hw\_reg\_val=%hw\_major%\*65536 + %hw\_minor%  SET jlink\_id=682613435  ECHO Start programming HW %hw\_major%.%hw\_minor%; reg:%hw\_reg\_val%  start /B /wait nrfjprog --snr %jlink\_id% --eraseall  start /B /wait nrfjprog --snr %jlink\_id% --memwr 0x10001080 --val %hw\_reg\_val%  start /B /wait nrfjprog --snr %jlink\_id% --memwr 0x10001084 --val %hw\_ID%  start /B /wait nrfjprog --snr %jlink\_id% --program Release/Beepbase.hex  start /B /wait nrfjprog --snr %jlink\_id% --reset  ECHO Programming Done  GOTO End  :End  pause |

**Dit batch script zet nog niet de readback protectie aan!**

Met de SET hw\_major, hw\_minor en hw\_ID worden de hardware versie en id nummers in de UUICR geprogrameerd. Deze waardes worden in de firmware gebruikt voor het tonen van de hardware versie in de DIS service en kunnen worden uitgelezen via het BEEP protocol.

Bovenstaande batch file heeft wel een vast Jlink ingesteld met ID 682613435. Dit zal aangepast moeten worden voor de programmer die wordt gebruikt. Het ID kan ook worden weggelaten, dan toont de driver een pop-up met de beschikbare interface bij elk batch commando als er meerdere programmers zijn.

## nRFutil

Om de zip bestanden te creren wordt er gebruik gemaakt van nRFutil.exe, een programman van Nordic dat hex bestanden kan encrypten voor DFU. Zorg er voor dat die inde map Util staat, aangezien deze executable niet in de repository systeem zit vanwege de grote van de applicatie. > 10Mb.

Voor ontwikkeling is nRFUtil 5.2.0 gebruikt. Oudere versies kunnen problemen opleveren die geen duidelijke fout weergeven.

## Segger Embedded Studio

Voor de ontwikkeling van de firmware is er gebruik gemaakt van Segger Embedded Studio, oftewel SES afgekort in de Nordic SDK. Om dit programma te gebruiken is er wel een gratis licentie nodig die wordt gekoppeld aan een hardware ID van de PC.

## Applicatie debuggen

Om de applicatie te debuggen met de DFU service is er een bootloader nodig die de CRC check niet uitvoert en mee wordt geladen tijden het debuggen. Om een bootloader te compileren is er een batch file gemaakt genaamd “Compile\_Bootloader\_SkipCRC” die de bootloader compileert onder een release die de CRC check negeert.

Als de bootloader firmware is aangepast of de source code is gedownload dient dit script eerst een keer te worden uitgevoerd om de hex file te creeren. SES zal bij het laden van alle files voor het debuggen wel een waarschuwing geven als de hex file ontbreekt.

Mocht er al een bootloader aanwezig is in de nRF52840 dan moet die eerst gewist worden om de nieuwe bootloader zonder crc check te programmeren.

## Compilatie scripts

De volgende batch script zijn er in de source code in de map util/Program:

|  |  |
| --- | --- |
| **Batch file naam** | **Functie** |
| Compile\_Beep\_release | Compileert release versies van de applicatie en bootloader. Vervolgens worden die samen gevoegd met het soft device om de hex file te creëren. Vanuit de verschillende gecreëerde batch hex files worden de zip file gecreëerd voor het updaten van de firmware over BLE |
| Compile\_Bootloader\_SkipCRC | Compileert een bootloader die geen CRC check uitvoert. Nodig voor het debuggen van een applicatie in SES. |
| EnableRBP | Zet de readback protectie aan. Hierna moet de BEEPBASE worden gerecovered waarbij de volledig FLASH geheugen van de nRF52840 wordt gewist. |
| erase | Hiermee wordt een microcontroller die aan een programmer gewist. |
| erase\_682613435 | Hiermee wordt een microcontroller die aan een programmer met id 682613435 gewist. |
| FICR\_read | Batch file waarmee enkele hardware parameters van een nRF52840 chip kan worden uitgelezen, zoals silicon version. |
| program\_BeepFirmware | Batch file die een aangesloten BeepBase programeerd met de gecompileerde hex file in de release map. |

# Elektrisch