

# **Beep Base handleiding V1.1**

## **ID190222-01**

Datum:	4-11-2019
Versie:	1.1
Titel:	Beep Base handleiding

## 1 Documentrevisie en verdeling

Documentrevisie:

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

## 2 Inhoudsopgave

<b>1</b>	<b>DOCUMENTREVISIE EN VERDELING</b>	<b>2</b>
<b>2</b>	<b>INHOUDSOPGAVE</b>	<b>3</b>
<b>3</b>	<b>INLEIDING</b>	<b>5</b>
<b>4</b>	<b>HARDWARE</b>	<b>5</b>
4.1	nRF52840	5
4.2	AA batterijen	5
4.3	DS18B20 temperatuur sensor	5
4.4	HX711 rekstrook sensor	6
4.5	Reedswitch	6
4.6	TPS boostconverter en supply switch	6
4.7	Buzzer	6
4.8	Flash storage	6
4.9	RFM95	6
4.10	ATECC608A	6
4.11	BME280	6
4.12	TLV320ADC3100	6
4.13	Logging	7
<b>5</b>	<b>SOFTWARE</b>	<b>7</b>
5.1	<b>Beep Protocol</b>	<b>7</b>
5.1.1	0d/0x00 – RESPONSE	8
5.1.2	1d/0x01 - READ_FIRMWARE_VERSION	9
5.1.3	2d/0x02 - READ_HARDWARE_VERSION	9
5.1.4	3d/0x03 - READ_DS18B20_STATE	10
5.1.5	131d/0x83 - WRITE_DS18B20_STATE	10
5.1.6	4d/0x04 - READ_DS18B20_CONVERSION	11
5.1.7	132d/0x84 - WRITE_DS18B20_CONVERSION	11
5.1.8	9d/0x09 - READ_HX711_STATE	12
5.1.9	137d/0x89 - WRITE_HX711_STATE	13
5.1.10	10d/0x0A - READ_HX711_CONVERSION	14
5.1.11	138d/0x8A - WRITE_HX711_CONVERSION	14
5.1.12	20d/0x14 - READ_LORAWAN_STATE	16
5.1.13	145d/0x91 - WRITE_BUZZER_DEFAULT_TUNE	16
5.1.14	146d/0x92 - WRITE_BUZZER_CUSTOM_TUNE	17

5.1.15	148d/0x94 - WRITE_LORAWAN_STATE	17
5.1.16	21d/0x15 - READ_LORAWAN_DEVEUI	18
5.1.17	149d/0x95 - WRITE_LORAWAN_DEVEUI	18
5.1.18	22d/0x16 - READ_LORAWAN_APPEUI	18
5.1.19	150d/0x96 - WRITE_LORAWAN_APPEUI	19
5.1.20	23d/0x17 - READ_LORAWAN_APPKEY	19
5.1.21	151d/0x97 - WRITE_LORAWAN_APPKEY	19
5.1.22	136d/0x98 - WRITE_LORAWAN_TRANSMIT	19
5.1.23	27d/0x1B - READ_nRF_ADC_CONVERSION	20
5.1.24	155d/0x9B - WRITE_nRF_ADC_CONVERSION	20
5.1.25	29d/0x1D - READ_APPLICATION_CONFIG	21
5.1.26	157d/0x9D - WRITE_APPLICATION_CONFIG	21
5.1.27	30d/0x1E - READ_PINCODE	22
5.1.28	158d/0x9E - WRITE_PINCODE	22
<b>5.2</b>	<b>Bluetooth Low Energy</b>	<b>23</b>
5.2.1	Pin code	23
5.2.2	Device information service	23
5.2.3	Battery service	23
5.2.4	DFU	24
5.2.5	Beep service	24
<b>5.3</b>	<b>LoRaWAN</b>	<b>25</b>
5.3.1	Standaard bericht types	25
<b>6</b>	<b>PROGRAMMEREN</b>	<b>26</b>
<b>6.1</b>	<b>BEEPBASE</b>	<b>26</b>
<b>6.2</b>	<b>Programeer script.</b>	<b>26</b>
<b>6.3</b>	<b>nRFutil</b>	<b>27</b>
<b>6.4</b>	<b>Segger Embedded Studio</b>	<b>27</b>
<b>6.5</b>	<b>Applicatie debuggen</b>	<b>27</b>
<b>6.6</b>	<b>Compilatie scripts</b>	<b>27</b>
<b>7</b>	<b>ELEKTRISCH</b>	<b>28</b>

### 3 Inleiding

Het Beep meet systeem (Beep Base) is een systeem voor het monitoren van een bijenkast 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.

### 4 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.

#### 4.1 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.

#### 4.2 AA batterijen

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

#### 4.3 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.

### 4.4 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.

### 4.5 Reedswitch

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

### 4.6 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.

### 4.7 Buzzer

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

### 4.8 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.

### 4.9 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.

### 4.10 ATECC608A

De ATECC608A 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.

### 4.11 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.

### 4.12 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.

## 4.13 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.

## 5 Software

### 5.1 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 gedefinieerde commando's:

Dec/hex	Naam	Omschrijving
0	0x00	RESPONSE
1	0x01	READ_FIRMWARE_VERSION
2	0x02	READ_HARDWARE_VERSION
3	0x03	READ_DS18B20_STATE
131	0x83	WRITE_DS18B20_STATE
4	0x04	READ_DS18B20_CONVERSION
132	0x84	WRITE_DS18B20_CONVERSION
5	0x05	READ_DS18B20_CONFIG
6	0x06	READ_BME280_STATE
7	0x07	READ_BME280_CONVERSION
8	0x08	READ_BME280_I2C
9	0x09	READ_HX711_STATE
10	0x0A	READ_HX711_CONVERSION
138	0x8A	WRITE_HX711_CONVERSION
11	0x0B	READ_AUDIO_ADC_STATE
12	0x0C	READ_AUDIO_ADC_CONVERSION
13	0x0D	READ_AUDIO_ADC_I2C
14	0x0E	READ_ATECC_READ_ID
15	0x0F	READ_ATECC_I2C
16	0x10	READ_BUZZER_STATE
17	0x11	READ_BUZZER_TUNE
18	0x12	READ_BUZZER_CUSTOM_TUNE
19	0x13	READ_SQ_MIN_STATE
20	0x14	READ_LORAWAN_STATE
148	0x94	WRITE_LORAWAN_STATE
21	0x15	READ_LORAWAN_DEVEUI
149	0x95	WRITE_LORAWAN_DEVEUI
22	0x16	READ_LORAWAN_APPEUI
150	0x96	WRITE_LORAWAN_APPEUI
23	0x17	READ_LORAWAN_APPKEY
151	0x97	WRITE_LORAWAN_APPKEY
136	0x88	WRITE_LORAWAN_TRANSMIT

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 waarden 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 waarden 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.

## 5.1.1 0d/0x00 – RESPONSE

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

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
<b>Opdracht</b>	0x96	WRITE_LORAWAN_APPEUI
<b>Antwoord</b>	0x009600000009	RESPONSE voor WRITE_LORAWAN_APPEUI, error code: Invalid Length

Als een onbekend commando wordt gestuurd wordt

	Hex bericht	Inhoud
<b>Opdracht</b>	0xFE	Geen gespecificeerde opdracht
<b>Antwoord</b>	0x00FE00000005	RESPONSE voor 0xFE, error code: Not found

## 5.1.2 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
<b>Opdracht</b>	0x01	READ_FIRMWARE_VERSION opdracht
<b>Antwoord</b>	0x01000000000001	READ_FIRMWARE_VERSION antwoord: firmware versie 0.0.1

## 5.1.3 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
<b>Opdracht</b>	0x02	READ_HARDWARE_VERSION opdracht
<b>Antwoord</b>	0x02000100000002E70E	READ_HARDWARE_VERSION antwoord: Hardware versie 1.0, ID= 190222

Firmware en hardware uitlezen

	Hex bericht	Inhoud
--	-------------	--------

<b>Opdracht</b>	0x0102	READ_FIRMWARE_VERSION , READ_HARDWARE_VERSION opdracht
<b>Antwoord</b>	0100000000000102000100000002E70E	READ_FIRMWARE_VERSION= 0.0.01 READ_HARDWARE_VERSION=1.0, ID= 190222

## 5.1.4 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
<b>Opdracht</b>	0x03	READ_DS18B20_STATE
<b>Antwoord</b>	0x0309	0b0000 1001 =Aan/uit=1 en resolutie=4:12bit resolutie

## 5.1.5 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
<b>Opdracht</b>	0x8309	WRITE_DS18B20_STATE, aan/uit=1, 12-bit resolutie
<b>Antwoord</b>	0x008300000000	

	Hex bericht	Inhoud
<b>Opdracht</b>	0x8305	WRITE_DS18B20_STATE, aan/uit=1, 10-bit resolutie
<b>Antwoord</b>	0x008300000000	

## 5.1.6 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
<b>Opdracht</b>	0x04	READ_DS18B20_CONVERSION
<b>Antwoord</b>	0x04010898	READ_DS18B20_CONVERSION, 1 DS18B20, temperatuur[0] = 0x0898/2200d = 22.00°C

Voorbeeld meerdere temperatuur sensoren:

	Hex bericht	Inhoud
<b>Opdracht</b>	0x04	READ_DS18B20_CONVERSION
<b>Antwoord</b>	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.

## 5.1.7 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
<b>Opdracht</b>	0x8400	WRITE_DS18B20_CONVERSION, start een temperatuur conversie met de temperatuur sensor op index 0.
<b>Antwoord1</b>	0x008400000000	NRF_SUCCESS, conversie wordt gestart. Dit bericht wordt bij de LoRaWAN interface niet terug gestuurd als de error code NRF_SUCCESS is.
<b>Antwoord2</b>	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
<b>Opdracht</b>	0x8408	WRITE_DS18B20_CONVERSION, start een temperatuur conversie met de temperatuur sensor op index 8 met slechts 2 sensoren aangesloten.
<b>Antwoord</b>	0x008400000007	NRF_ERROR_INVALID_PARAM voor opdracht WRITE_DS18B20_CONVERSION

Voorbeeld alle temperatuur sensoren:

	Hex bericht	Inhoud
<b>Opdracht</b>	0x84FF	WRITE_DS18B20_CONVERSION, start een temperatuur conversie met alle aangesloten temperatuur sensor.
<b>Antwoord1</b>	0x008400000000	NRF_SUCCESS, conversie wordt gestart. Dit bericht wordt bij de LoRaWAN interface niet terug gestuurd als de error code NRF_SUCCESS is.
<b>Antwoord2</b>	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 -100.0°C 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.

## 5.1.8 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 berekend. 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
<b>Opdracht</b>	0x09	READ_HX711_STATE
<b>Antwoord</b>	0x090102	0x09=READ_HX711_STATE 0x01= CH_A_GAIN128 0x02= 2 samples per kanaal

Voorbeeld 2:

	Hex bericht	Inhoud
<b>Opdracht</b>	0x09	READ_HX711_STATE
<b>Antwoord</b>	0x09070A	0x09=READ_HX711_STATE 0x07= CH_A_GAIN128, CH_B_GAIN32, CH_A_GAIN64 0x0A= 10 metingen per kanaal

## 5.1.9 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
<b>Opdracht</b>	0x89070A	0x89= WRITE_HX711_STATE 0x07= CH_A_GAIN128, CH_B_GAIN32, CH_A_GAIN64 0x0A= 10 metingen per kanaal
<b>Antwoord</b>	0x008900000000	0x00=RESPONSE_COMMAND 0x89=WRITE_HX711_STATE 0x00000000=NRF_SUCCESS

## 5.1.10 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
<b>Opdracht</b>	0x0A	READ_HX711_CONVERSION
<b>Antwoord</b>	0x0A010182E6	READ_HX711_CONVERSION, kanaal 0x01:CH_A_GAIN128:, resultaat: 99046/0x0182e6

## 5.1.11 138d/0x8A - WRITE\_HX711\_CONVERSION

Start een nieuwe gemiddelde meting op de opgegeven kanalen. Als de conversie 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
<b>Opdracht</b>	0x8A010A	WRITE_HX711_CONVERSION, CH_A_GAIN128, 10 samples
<b>Antwoord1</b>	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.
<b>Antwoord2</b>	0x8A010183A4	0x8A = WRITE_HX711_CONVERSION 0x01 = CH_A_GAIN128 0x0183A4 = 99.236decimaal

Voorbeeld 2:

	Hex bericht	Inhoud
<b>Opdracht</b>	0x8A070A	WRITE_HX711_CONVERSION, CH_A_GAIN128, CH_B_GAIN32, CH_A_GAIN64, 10 samples
<b>Antwoord1</b>	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.
<b>Antwoord2</b>	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
--	-------------	--------



<b>Opdracht</b>	0x8A010A	WRITE_HX711_CONVERSION, CH_A_GAIN128, CH_B_GAIN32, CH_A_GAIN64, 10 samples
<b>Antwoord1</b>	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.
<b>Antwoord2</b>	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

## 5.1.12 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 waarden.

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
<b>Opdracht</b>	0x14	READ_LORAWAN_STATE
<b>Antwoord</b>	0x141F	0x1F = 0001 1111b: LoRaWAN aan, netwerk gejoined, DutyCycle limitatie aan, ADR aan, Correcte sleutels

	Hex bericht	Inhoud
<b>Opdracht</b>	0x14	READ_LORAWAN_STATE
<b>Antwoord</b>	0x141B	0x1F = 0001 1011b: LoRaWAN aan, netwerk gejoined, DutyCycle limitatie aan, ADR uit, Correcte sleutels

## 5.1.13 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 patronen, 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
<b>Opdracht</b>	0x9101	0x91=WRITE_BUZZER_DEFAULT_TUNE 0x01=PWM patroon 1
<b>Antwoord</b>	0x009100000000	0x00=RESPONSE_COMMAND 0x91= WRITE_BUZZER_DEFAULT_TUNE 0x00000000=NRF_SUCCESS

## 5.1.14 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	uint16_t	>0	Aantal keer dat de aan-uit cyclus wordt herhaald.

Voorbeeld:

	Hex bericht	Inhoud
<b>Opdracht</b>	0x92321C03E801F403	0x92= WRITE_BUZZER_CUSTOM_TUNE 0x32=Dutycycle: 50% 0x1C=Frequentie:2.8kHz 0x03E8=Uit-tijd:1000ms 0x01F4=Aan-tijd:500ms 0x03=Herhalingen:3
<b>Antwoord</b>	0x009200000000	0x00=RESPONSE_COMMAND 0x92= WRITE_BUZZER_CUSTOM_TUNE 0x00000000=NRF_SUCCESS

## 5.1.15 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 waarden.

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
<b>Opdracht</b>	940D	0x0F = 0000 1101b: LoRaWAN aan, DutyCycle limitatie aan, ADR aan,
<b>Antwoord</b>	009400000000	NRF_SUCCESS voor WRITE_LORAWAN_STATE

Voorbeeld om de LoRaWAN testen zonder Duty-cycle limitatie:

	Hex bericht	Inhoud
<b>Opdracht</b>	9409	0x0F = 0000 1001b: LoRaWAN aan, DutyCycle limitatie uit, ADR aan,
<b>Antwoord</b>	009400000000	NRF_SUCCESS voor WRITE_LORAWAN_STATE

## 5.1.16 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
<b>Opdracht</b>	0x15	READ_LORAWAN_DEVEUI opdracht
<b>Antwoord</b>	0x150001020304050607	READ_LORAWAN_DEVEUI antwoord, DEVEUI: 01020304050607

## 5.1.17 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
<b>Opdracht</b>	0x950001020304050607	WRITE_LORAWAN_DEVEUI opdracht
<b>Antwoord</b>	0x009500000000	WRITE_LORAWAN_DEVEUI succesvol.

## 5.1.18 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
<b>Opdracht</b>	0x16	READ_LORAWAN_APPEUI opdracht
<b>Antwoord</b>	0x160001020304050607	READ_LORAWAN_APPEUI succesvol.

## 5.1.19 150d/0x96 - WRITE\_LORAWAN\_APPEUI

Met deze opdracht wordt de APPEUI ingesteld. De APPEUI moet 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
<b>Opdracht</b>	0x960001020304050607	WRITE_LORAWAN_APPEUI opdracht
<b>Antwoord</b>	0x009600000000	WRITE_LORAWAN_APPEUI succesvol.

## 5.1.20 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
<b>Opdracht</b>	0x17	READ_LORAWAN_APPKEY opdracht
<b>Antwoord</b>	0x17000102030405060708090A0B0C0D0E0F	READ_LORAWAN_APPKEY succesvol.

## 5.1.21 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
<b>Opdracht</b>	0x97000102030405060708090A0B0C0D0E0F	WRITE_LORAWAN_APPKEY opdracht
<b>Antwoord</b>	0x009700000000	WRITE_LORAWAN_APPKEY succesvol.

## 5.1.22 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
<b>Opdracht</b>	0x9810000102030405060708090A0B0C0D0E0F	WRITE_LORAWAN_APPKEY opdracht met een lengte van 16 bytes.
<b>Antwoord</b>	0x009800000000	WRITE_LORAWAN_APPKEY succesvol. Wordt enkel gestuurd wanneer de communicatie interface BLE is.

## 5.1.23 27d/0x1B - READ\_nRF\_ADC\_CONVERSION

Lees de laatste conversie waarden 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
<b>Opdracht</b>	1B	
<b>Antwoord</b>	1B0B530BB85C	Batterij spanning 0x0B53=2899mV, nRF spanning=3000mV, Batterij percentage 92%

## 5.1.24 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
<b>Opdracht</b>	0x9B	WRITE_nRF_ADC_CONVERSION, start conversie
<b>Antwoord</b>	0x9B0B530BB85C	Batterij spanning 0x0B53=2899mV, nRF spanning=3000mV, Batterij percentage 92%

## 5.1.25 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 metingen-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 verstuurd, mits de duty-cycle dit niet beperkt.
Interval	Uint16_t	>0	Het meetinterval in minuten.

Voorbeeld:

	Hex bericht	Inhoud
<b>Opdracht</b>	0x1D	READ_APPLICATION_CONFIG
<b>Antwoord</b>	0x1D03000A	Ratio van 1:3 voor meten en verzenden. 1 op de drie metingen wordt met LoRaWAN verstuurd. Meet interval is 0x000A/10d minuten.

	Hex bericht	Inhoud
<b>Opdracht</b>	0x1D	READ_APPLICATION_CONFIG
<b>Antwoord</b>	0x1D000001	Alle metingen worden verzonden.. Meet interval is 0x0001/1d minuut.

## 5.1.26 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 metingen-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 verstuurd, 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
<b>Opdracht</b>	0x9D03000A	Ratio van 1:3 voor meten en verzenden. 1 op de drie metingen wordt met LoRaWAN verstuurd. Meet interval is 0x000A/10d minuten.
<b>Antwoord</b>	0x009D00000000	NRF_SUCCES voor WRITE_APPLICATION_CONFIG

Voorbeeld met incorrecte sample interval:

	Hex bericht	Inhoud
<b>Opdracht</b>	0x9D03FFFF	Ratio van 1:3 voor meten en verzenden. 1 op de drie metingen wordt met LoRaWAN verstuurd. Meet interval is 0xFFFF/10d minuten.
<b>Antwoord</b>	0x009D00000007	NRF_ERROR_INVALID_PARAM

Testen:

	Hex bericht	Inhoud
<b>Opdracht</b>	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.
<b>Antwoord</b>	0x009D00000000	NRF_SUCCES voor WRITE_APPLICATION_CONFIG

## 5.1.27 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
<b>Opdracht</b>	0x1E	READ_PINCODE opdracht
<b>Antwoord</b>	0x1E083031323334353637	READ_PINCODE antwoord, pincode 8 bytes: "01234567"

## 5.1.28 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_LENGTH
pincode	7-16 x UInt8_t	0 - 9	Byte waarden 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
<b>Opdracht</b>	0x9E083039303930353035	WRITE_PINCODE opdracht: "09090505"
<b>Antwoord</b>	0x009E00000000	WRITE_PINCODE succesvol.

Voorbeeld van fout bericht:

	Hex bericht	Inhoud
<b>Opdracht</b>	0x9E080039303930353035	WRITE_PINCODE opdracht: "/09090505"
<b>Antwoord</b>	0x009E0000000B	WRITE_PINCODE error: NRF_ERROR_INVALID_DATA.

## 5.2 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.

### 5.2.1 Pin code

Is nog niet geactiveerd in firmware.

### 5.2.2 Device information service

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

- Manufacturer Name String: "BEEP"
- Model Number String: "BEEPBASE"
- Serial number String: "TODO:ATTEC"
- Hardware Revision String: "1.0"
- Firmware Revision String: "0.0.1"

### 5.2.3 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.

$$\text{Batterij cutt - off spanning: } mV_{\text{cutt-off}} = 1600 \text{ mV}$$

$$\text{Battery nominaal spanning: } mV_{\text{nom}} = 3000 \text{ mV}$$

$$\text{gemiddelde batterij spanning: } mV_{\text{batt\_gem}}$$

$$mV_{\text{batt\_gem}} = \frac{\sum ADC_{\text{samples}} \times 3600.0}{N_{\text{samples}} \times 4096.0}$$

$$\text{Percentage} = \frac{(mV_{\text{batt\_gem}} - mV_{\text{cutt-off}}) * 100}{mV_{\text{batt\_gem}} - mV_{\text{cutt-off}}}$$

Voorbeeld:

$$mV_{\text{batt\_gem}} = \frac{32993 \times 3600.0}{10 \times 4096.0} = 2899.77 \text{ mV}$$

$$\text{Percentage} = \frac{(2899.77 \text{ mV} - 1600 \text{ mV}) * 100}{3000 \text{ mV} - 1600 \text{ mV}} = 92.84\%$$

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

## 5.2.4 DFU

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

## 5.2.5 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!

### 5.2.5.1 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:  $2200 / 100 = 22.00^{\circ}\text{C}$ .

### 5.2.5.2 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.

### 5.2.5.3 Log uitlezen

Nog niet geïmplementeerd.

## 5.3 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.

### 5.3.1 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.

## 6 Programmeren

### 6.1 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.

### 6.2 Programeer script.

Om de beepbase met de aangeleverde hex file te programmeren 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 geprogrammeerd. Deze waarden 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.

### 6.3 nRFutil

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

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

### 6.4 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.

### 6.5 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 tijdens 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 creëren. 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 zijn in de nRF52840 dan moet die eerst gewist worden om de nieuwe bootloader zonder crc check te programmeren.

### 6.6 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.

## 7 Elektrisch