

Multiflexmeter

specificatie software

Project :P22296

Product :P4748 / P4750 / P4758

Klant nr :-



ELECTRONICS
SOFTWARE
MECHATRONICS
MANUFACTURING

Dekimo Goes B.V.

Klant :Provincie Zeeland

Auteur :Dekimo Goes

Bestand :P22296-10-SPEC-2.0 software.docx

Historie

Datum	Version	Opmerkingen / Wijzigingen	Handtekening
05-04-2023	1,0		
13-07-2023	1,1	Verwerking van opmerkingen klant (vrijgegeven door Tim 15-08-2023)	
30-11-2023	1,2	<ul style="list-style-type: none"> - Indeling backup registers gewijzigd - Aanvulling command Set+Test=6 (SD kaart), Set+Test=7 (RTC), Set+Test=8 (batterij monitor), Nieuw commando tbv Set+Test=10 (FRAM), Set+Test=11 (Dataflash), Set+Test=12 (system checks) - I2C I/O expander adressen geupdate nalv implementatie - Sensor module types definitie RS485/one-wire - Uplink informatie aangevuld - Druk sensor meetdata definitie toegevoegd. - Definitie Lora data packet aangevuld - Lora versie software naar V1.0.4 - Correctie MSB/LSB in schrijf meettijd 	
19-12-2023	1,3	<ul style="list-style-type: none"> - Aanvulling flash indeling - Fram indeling toegevoegd - I2C adressen aangevuld met RTC en batterij monitor - Wijziging test commando's RTC / SD kaart - Wijziging pagina indeling dataflash - Rejoin at next interval - Gebruik van backup registers 5, 6, 7, 8 - Lora protocol <message type> definitie toegevoegd 	
15-02-2024	1,4	<ul style="list-style-type: none"> - Standaard OK response vervangen voor response bericht (H3.8) - Correctie typefouten + eenduidig LoRa / LoRaWAN - Endianes en data volgorde verduidelijkt. - Opspliten verplichte commando's en test commando's sensor module. - Toevoegen ontbrekende DeviceID H0 - Test commando's voor Sleep en meting (Set+Test=7,7 en 7,8) - Nieuwe seriële commando's Get+Vcc, Set+Reboot, Set+InitSensor, Set+Rejoin. - Nieuwe sensor module commando (I2C) tbv sensor initialisatie - Aanvulling commando's met Set+save - Wijziging commando Set+MeasureTime -> Set+Samples - Beschrijving reset knop met geforceerde rejoin - Aanvulling MFM data, tijdsynchronisatie, confirmed messages 	
11-03-2024	1,5	<ul style="list-style-type: none"> - Aanpassing meetinterval per sensor -> per cycle - Correctie in "status LEDs" - SensorModule Slot ID in communicatie van 0-5 -> 1-6 nalv FAT - Protocol sensormodule onewire naar ruwe waarden - Nieuwe commando's: Set+Eos, Get+Nonce, Set+Nonce 	
02-04-2024	1,6	<ul style="list-style-type: none"> - Set/Get+Nonce aangepast voor individuele counters - Downlink commando rejoin uitgebreid met optioneel reset counter - Downlink commando voor interval toegevoegd - Commando voor configureren nieuwe batterij. 	
15-07-2024	1,7	<ul style="list-style-type: none"> - Sensor commando's tbv productie test toegevoegd H4.3.6 en H4.3.7 - I2C restart gewijzigd in start nalv implementatie. - I2C CRC berekening verduidelijkt 	
29-01-2025	2,0	Kopieren verklaring verwijderd.	

Inhoudsopgave

1	Inleiding	5
2	Functionele beschrijving	5
3	MFM basis module	6
3.1	Data opslag.....	6
3.1.1	Definitie data	6
3.1.2	Flash controller.....	6
3.1.3	Backup registers	6
3.1.4	Data flash (NOR flash).....	7
3.1.5	SD kaart	7
3.1.6	Fram.....	7
3.2	Wake-up.....	7
3.3	Reset schakelaar	7
3.4	Status LEDs	8
3.5	Status monitoring	8
3.6	I2C bus I/O expander en Sensormodules	8
3.6.1	I/O expander MFM basis module.....	8
3.6.2	I/O expander expansion board.....	9
3.6.3	Sensormodules	9
3.6.4	Gereserveerde adressen	9
3.7	I2C systeem bus MFM basis module systeem	10
3.8	Seriële poort (via USB)	10
3.8.1	Commando Get+ModuleInfo.....	11
3.8.2	Commando Get+SensorInfo	11
3.8.3	Commando Set+JoinID / Get+JoinID	11
3.8.4	Commando Set+DeviceID / Get+DeviceID	11
3.8.5	Commando Set+DeviceId	12
3.8.6	Commando Set+AppKey / Get+AppKey.....	12
3.8.7	Commando Set+Sensor / Get+Sensor	12
3.8.8	Commando Set+LoraInterval / Get+LoraInteval.....	12
3.8.9	Commando Set+Samples / Get+Samples	13
3.8.10	Commando Get+DataDump	13
3.8.11	Commando Set+AlwaysOn / Get+AlwaysOn	13
3.8.12	Commando Set+Erase.....	14
3.8.13	Commando Set+Save.....	14
3.8.14	Commando Set+Test.....	14
3.8.15	Commando Get+Bat	18
3.8.16	Commando Get+Vbus	18
3.8.17	Commando Get+Vcc.....	18
3.8.18	Commando Set+Reboot	18
3.8.19	Commando Set+InitSensor.....	18
3.8.20	Commando Set+Rejoin.....	19
3.8.21	Commando Set+Eos.....	19
3.8.22	Commando Get+Nonce / Set+Nonce	20
3.9	Configuratie.....	21
3.9.1	Parameters.....	21
3.9.2	Opslag locatie	21
3.10	Meetdata	21
3.10.1	Blok definitie dataflash	21
3.10.2	Wis tijden.....	21
3.10.3	Indeling meetdata pagina.....	21
3.10.4	Administratie meetdata	22
3.10.5	Schrijftijd meting.....	22
3.10.6	Wissen meetdata	22
3.10.7	Koude start.....	23
3.11	Indeling backup registers	23
3.12	Broncode.....	23
4	Communicatie MFM basis module <-> MFM sensor module.....	24
4.1	Bericht definitie	24

4.1.1	Lees commando's	24
4.1.2	Schrijf commando's	24
4.2	Sensor commando's	24
4.2.1	Lees sensor firmware versie	25
4.2.2	Lees protocol versie	25
4.2.3	Lees sensor type	25
4.2.4	Start initialisatie module	25
4.2.5	Status initialisatie module	25
4.2.6	Schrijf start / stop meting	25
4.2.7	Lees status meting	26
4.2.8	Lees meettijd	26
4.2.9	Lees meetdata	26
4.3	Test commando's	26
4.3.1	Lees aantal sensors	26
4.3.2	Schrijf selectie van sensor	26
4.3.3	Schrijf type meetwaarde	27
4.3.4	Schrijf / lees meetsamples per meting	27
4.3.5	Lees sensor waarde van geselecteerde sensor	27
4.3.6	Schrijf uitgang slot GPIO/INT	27
4.3.7	Schrijf ingang/uitgang voor slot GPIO	27
4.4	Sensor types	27
4.5	Type meetwaarde	28
4.6	Error code	29
5	MFM sensor module : Drukmodule	30
5.1	Meetdata RS485	31
5.2	Meetdata onewire	31
5.3	Variant	32
5.3.1	Drukmodule RS485	32
5.3.2	Drukmodule One-wire	32
5.4	Broncode	32
6	LoRaWAN communicatie	33
6.1	MFM data bericht	33
6.1.1	MFM data velden	34
6.1.2	MFM diagnostische status bits	34
6.2	Join	34
6.3	Tijd synchronisatie	34
6.4	Uplink bericht	35
7	Tool en licenties	36
7.1	Licentie STMicroelectronics	36
7.2	Licentie STM32WLxx_Nucleo	37
7.3	STM32WLxx_HAL_Driver	37
7.4	Licentie LoRaWan driver	37
7.5	Licentie SubGhz_Phy (radio) driver	37
8	Configuratie software MFM basis module	39
8.1	Ontwikkeling	39
8.2	Functioneren (gebruikers interactie)	39
8.3	Broncode	40
9	Drivers USB VCP	41
10	Diagrammen software structuur	42
10.1	Activiteit diagram MFM basismodule	42
10.2	Activiteit diagram MFM sensormodule	43

1 Inleiding

Dit document vormt de specificatie van de software voor de Multiflexmeter (MFM).

2 Functionele beschrijving

De MFM bestaat uit een MFM basis module en een MFM sensor module.

De MFM basis module is te gebruiken met 1 t/m 6 MFM sensor modules. Bij meer dan 2 tot 6 MFM sensor modules dient er een MFM expansion board toegepast te worden.

De MFM basis module software is het hart van het systeem en verzorgt oa:

- LoRa communicatie
- Lokaal instellen van parameters (configuratie).
- Lokaal uitlezen van parameters (configuratie).
- Lokaal uitlezen en wissen van geheugen
- Opslag van meetgegevens van aangesloten MFM sensormodules.
- Communicatie met aangesloten MFM sensormodules.
- In en uitschakelen van aangesloten MFM sensormodules en MFM basis module
- Meten van batterij status.

De MFM basis module is de master en zorgt ervoor dat iedere actief geconfigureerde sensormodule op het ingestelde interval één keer gemeten wordt.

Er wordt altijd maar één sensormodule slot tegelijkertijd gemeten en ingeschakeld.

Een meting bestaat uit de volgende stappen: Inschakelen van de slot spanning, korte wachttijd ivm opstarten, opvragen status, opvragen meettijd, starten van meting sensormodule, wachten verstrijken meettijd, opvragen status, opvragen meetdata, uitschakelen sensormodule slot.

Na het starten van een meting wordt gewacht tot het verstrijken van de “meettijd”, waarna de MFM basismodule de meetstatus op gaat vragen obv polling. Indien de status “gereed” is, kan de meetdata opgevraagd worden. Hierna zal sensormodule weer uitgeschakeld worden.

De MFM basismodule zal de meetdata opslaan in het dataflash geheugen, waarna de nieuwe data verstuurd wordt via LoRa.

Na één zendpoging wordt de MFM basismodule in slaap¹ gebracht, totdat de volgende slaapinterval verstreken is. Waarnaar bovenstaande herhaald wordt.

Indien er meer dan één sensormodules actief geconfigureerd zijn, zal in de volgende minuut de volgende sensor actief worden. Deze cycli wordt herhaald totdat alle actief geconfigureerde sensormodules ingelezen zijn en verstuurd.

Nadat deze hele cyclus van 1 tot 6 sensormodules afgewerkt is gaat het systeem in slaap voor “meetinterval – (actieve sensoren * 1 minuut)”².

Eventueel niet verstuurd data wordt niet opnieuw gestuurd en blijft behouden in het dataflash voor lokale uitlezing. Zie meer details H3.10 Meetdata.

¹ De controller wordt volledig uit gezet en wordt gewekt door het RealTime Clock (RTC) IC.

² Bij “meetinterval” = 5 minuten en 6 actieve sensoren zal het effectieve interval 10 minuten zijn ipv 5 minuten.

3 MFM basis module

De applicatie wordt ontwikkeld voor een STM32WL55CCU6 controller en is gebaseerd op de "LoRaWan_End_Node" (single core) uit de STM32Cube MCU packages voor STM32WL serie met LoRa protocol versie 1.0.4³ klasse A.

De voordelen bij een single core applicatie zijn een eenvoudigere software structuur en er is de volledige hoeveelheid flash en RAM beschikbaar voor de applicatie.

3.1 Data opslag

3.1.1 Definitie data

Alle data definities in code zijn standaard "little-endian".

Voor de leesbaarheid in documentatie zijn alle hexadecimale getalen in dit document standaard "big-endian". Wanneer er gesproken wordt over MSB / LSB byte dan wordt dit ook vanuit het "big-endian" perspectief bedoeld.

Voorbeeld:

Decimaal getal: 6000

Hexadecimaal in big-endian: 0x1770

MSB = 0x17

LSB = 0x70.

Bij het versturen van data wordt het LSB byte van een hexadecimaal getal in "big-endian" eerst verstuurd.

3.1.2 Flash controller

Gebruikt voor opslag van programma (optioneel: bootloader⁴) en configuratie (zie H0).

256 kbyte flash

De fabrikant garandeert minimum van 10k cycli bij T-omgeving van -40 tot +105 °C.

De maximale omgevingstemperatuur heeft invloed op de data retentie. De fabrikant garandeert een minimum bij 10k cycli van 30 jaar bij +55 °C, 15 jaar bij +85 °C en 10 jaar bij +105 °C.

Zie uitgebreide informatie in datasheet STM32WL55xx (DS13293).

Voor de configuratie is een apart geheugen segment gedefinieerd.

Indeling is als volgt:

Flash voor applicatie: 0x08000000 – 0x0803CFFF (244kbytes)

Gereserveerd voor virtueel EEPROM: 0x0803D000 – 0x0803EFFF (8kbytes)

Gereserveerd voor NVMDATA LoRaWAN driver: 0x0803F000 – 0x0803FFFF (2kbytes)

3.1.3 Backup registers

De controller STM32WL55CCU6 heeft 20x 32bit registers welke behouden blijven indien de VBat pin van voeding is voorzien.

Deze registers worden gebruikt voor de administratie van de meetdata.

Zie ook H3.11 (Indeling backup registers).

³ In driver om te zetten naar 1.0.3, de gereserveerde ruimte in FRAM is onvoldoende voor de lora NVM settings versie 1.0.3.

⁴ Niet in gebruik

3.1.4 Data flash (NOR flash)

Gebruikt voor opslag van meetdata.

Levensduur afhankelijk van type:

Voor AT25QF641B: 100k programmeer en wis cycli en data retentie van 20 jaar.

Het geheugen wordt als ringbuffer gebruikt waarbij één sensormodule meting per pagina wordt opgeslagen.

Zie uitgebreide beschrijving van gebruik in H3.10 Meetdata.

3.1.5 SD kaart

Verwisselbaar geheugen, voor toekomstig gebruik icm FOTA (Firmware over the air) en uitgebreide log mogelijkheid.

3.1.6 Fram

Non-volatile 16 kbit ferroelectric random access memory (F-RAM) behoud data betrouwbaar zonder voedingsspanning voor meerdere jaren.

16kbit = 2048 bytes.

Indeling:

0x0000 – 0x01FF: applicatie instellingen

0x0200 – 0x07FF: LoRaWAN NVM instellingen (initieel: 1472 bytes)

3.2 Wake-up

Het merendeel van het de tijd is het systeem in slaapmode. Hierbij is de controller spanningsloos. Om de controller te activeren zijn meerdere wake-up bronnen, zie hardware specificatie P22296-1-SPEC-x.x

Door oa het aansluiten van de USB wordt de controller actief, waarna het systeem uit te lezen is.

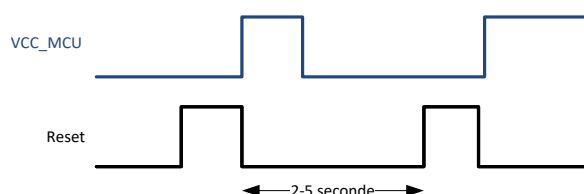
3.3 Reset schakelaar

De reset schakelaar heeft een dubbele functie.

1. Reset van processor indien deze actief is.⁵
2. Wakker maken van de processor indien deze in slaap is (VCC_MCU is uit).

Met behulp van deze schakelaar is een REJOIN te forceren door onderstaande sequentie uit te voeren vanuit slaap.

Ook wordt hierdoor de sensormodule initialisatie geactiveerd bij eerstvolgende sensor module slot opvraag met het commando "Start initialisatie module" zie h4.2.4.



Figuur 1: timing geforceerde rejoin

⁵ Een reset tijdens het uitvoeren van het programma (schrijven van data) kent risico's op corrupte data.

3.4 Status LEDs

Op de MFM basis module zitten twee status LEDs voor test doeleinden.

In ontwikkelfase in te schakelen met soldeerjumper.
In definitieve versie standaard uitgeschakeld.

- LED1 knippert indien MFM basis module actief.
- LED2 knippert indien LoRa actief.

3.5 Status monitoring

De MFM basismodule monitort met de interne ADC van de microcontroller de 3V3(bat) spanning en de VBUS spanning.

Deze spanningen zijn uit te lezen via commando op de seriële poort.

De batterij monitor wordt in de End Of Service (EOS) mode gebruikt. Het EOS algoritme maakt gebruik van spanning, stroom en temperatuurgegevens om de weerstand en de mate van weerstandsverandering van de batterij te bepalen. De weerstandsgegevens worden vervolgens gebruikt om de diepte van de ontlading te bepalen. Zie voor uitgebreidere beschrijving de technische handleiding van BQ35100 van Texas Instruments (document SLUUBH1C).

De EOS meting wordt 1x per 24 uur uitgevoerd rond 12:00 uur.

De waarde wordt opgeslagen in de backup registers van de controller (zie H3.11 Indeling backup registers) en verstuurd naar de portal via LoRaWAN (zie H6.1 MFM data bericht).

3.6 I2C bus I/O expander en Sensormodules

De I2C bus is een gemeenschappelijke bus voor de I/O expander aanwezig op de MFM basismodule, voor de sensormodule slot 1 en slot 2 op de MFM basismodule en voor de MFM expansion board connector waarop uiteindelijk 1x I/O expander en 4x sensormodules aangesloten kunnen worden.

De I/O expander op de MFM basis module is altijd aanwezig.

De overige modules kunnen aanwezig zijn.

3.6.1 I/O expander MFM basis module

De volgende I/O wordt bedient met de I/O expander MFM basis module.

1. Load switch sensor module 1
2. Load switch sensor module 2
3. Sensor 1 I/O 1 ⁶
4. Sensor 1 I/O 2 ⁶
5. Sensor 1 I/O 3 ⁶
6. Sensor 2 I/O 1 ⁶
7. Sensor 2 I/O 2 ⁶
8. Sensor 2 I/O 3 ⁶
9. Ready signal sensor module 1 ⁶
10. Ready signal sensor module 2 ⁶
11. Ready signal sensor module 3 ⁶
12. Ready signal sensor module 4 ⁶
13. Ready signal sensor module 5 ⁶
14. Ready signal sensor module 6 ⁶
15. Geen functie
16. Geen functie

Zie tabel H3.6.4 Gereserveerde adressen voor het adres van de I/O expander.

⁶ Alleen voorbereid, heeft geen functie in de software.
aBestand: P22296-10-SPEC-2.0 software.docx
specificatie software Multiflexmeter

3.6.2 I/O expander expansion board

De volgende I/O wordt bedient met de I/O expander MFM expansion board.

1. Load switch sensor module 3
2. Load switch sensor module 4
3. Load switch sensor module 5
4. Load switch sensor module 6
5. Sensor 3 I/O 1⁶
6. Sensor 3 I/O 2⁶
7. Sensor 3 I/O 3⁶
8. Sensor 4 I/O 1⁶
9. Sensor 4 I/O 2⁶
10. Sensor 4 I/O 3⁶
11. Sensor 5 I/O 1⁶
12. Sensor 5 I/O 2⁶
13. Sensor 5 I/O 3⁶
14. Sensor 6 I/O 1⁶
15. Sensor 7 I/O 2⁶
16. Sensor 8 I/O 3⁶

Zie tabel H3.6.4 Gereserveerde adressen voor het adres van de I/O expander.

3.6.3 Sensormodules

De communicatie tussen de MFM basis module en MFM sensor modules verloopt via I2C bus. Hierbij is de MFM basis module de master en zijn de (tot 6 stuks) MFM sensor modules de slaves.

De adressen van MFM sensor modules worden bepaald door het sensor slot in hardware (zie tabel H3.6.4 Gereserveerde adressen).

De MFM basis module (master) stuurt een verzoek (of commando) naar een register van de MFM sensor module (slave). Afhankelijk van het bericht volgt nog een "restart" om databytes uit te lezen, zie H4 "Communicatie MFM basis module <-> MFM sensor module".

Iedere MFM sensor module zal een standaard set aan registers ondersteunen. Per type MFM sensor kunnen nog specifieke registers gedefinieerd zijn.

Meet algoritme.

1. De MFM basis module schakelt de MFM sensor module in.
2. De MFM basis module stuurt een start commando naar de MFM sensor module.
3. De MFM basis module vraagt periodiek de meet status op van de MFM sensor module.
 - a. Indien status " bezig " dan wacht en herhaal 2.
 - b. Indien status " gereed " dan stap 3.
4. De MFM basis module vraagt de meetdata grootte op van de MFM sensor module.
5. De MFM basis module vraagt de meetdata op van de MFM sensor module.
6. De MFM basis module schakelt de MFM sensor module uit.

Zie beschrijving commando's H4.2 "Sensor commando's".

3.6.4 Gereserveerde adressen

Voor I2C zijn adressen 0x00 t/m 0x7F beschikbaar.

De gebruikte adressen voor sensor 1 t/m sensor 6 zijn respectievelijk 0x11 t/m 0x16. De gereserveerde adressen voor de I/O expanders zijn 0x20 t/m 0x27.

Adres	Functie
0x11	Sensor module 1

0x12	Sensor module 2
0x13	Sensor module 3
0x14	Sensor module 4
0x15	Sensor module 5
0x16	Sensor module 6
0x20	I/O expander 1 (MFM expansion board)
0x21	I/O expander 2 (MFM basis module)
0x22	I/O expander 3 (gereserveerd)
0x23	I/O expander 4 (gereserveerd)
0x24	I/O expander 5 (gereserveerd)
0x25	I/O expander 6 (gereserveerd)
0x26	I/O expander 7 (gereserveerd)
0x27	I/O expander 8 (gereserveerd)

3.7 I2C systeem bus MFM basis module systeem

Aparte I2C bus voor systeem functies.

Adres	Functie
0x21	I/O expander 2 (MFM basis module)
0x55	Batterij monitor BQ35100
0x69	Realtime clock (RTC) AM1805

3.8 Seriële poort (via USB)

Voor lokaal instellen / uitlezen van parameters (configuratie) en het lokaal uitlezen en wissen van het geheugen wordt de seriële poort (via USB) gebruikt.

De volgende baudrate wordt toegepast: 115200, 8, N, 1

Voor communicatie is een set aan commando's gedefinieerd opgesplitst in "set" en "get" commando's. Het betreft ASCII commando's die via een standaard terminal applicatie vanaf een PC te versturen zijn.

Hieronder is beschreven hoe de commando's standaard reageren. Uitzonderingen en specifiek gedrag is beschreven onder het individuele commando.

Het "Set" commando is opgebouwd uit een commando en een instelling op de volgende wijze:
"<Set+commando>=<waarde><CR><LF>"

Als het commando geaccepteerd is wordt er een response bericht teruggestuurd.
Het bericht is beschreven bij de afzonderlijke commando's.
Bij onjuist commando of onjuiste waarde wordt er een ERROR bericht terug gestuurd.
"ERROR <CR><LF>".

Het "Get" commando is opgebouwd uit een commando en een instelling op de volgende wijze:
"<Get+commando><CR><LF>"

Als het commando geaccepteerd is wordt het resultaat gestuurd.
"<Get+commando>=<waarde><CR><LF>"

Bij onjuist commando een ERROR bericht terug gestuurd.
"ERROR <CR><LF>".

De geldige invoer waardes kunnen opgevraagd worden met <COMMANDO>=?<CR><LF>.
Er zal dan gereageerd worden met <COMMANDO>:(x1-y1),(x2-y2),...<CR><LF>

3.8.1 Commando Get+ModuleInfo

Commando om versie informatie van de MFM basis module op te vragen.

Commando: Get+ModuleInfo

Commando antwoord: ModuleInfo:<protocol1>,<protocol2>,<versie>

<protocol1> = protocol versie seriële poort communicatie MFM basis module (0-255).

<protocol2> = protocol versie I2C communicatie MFM basis module <-> MFM sensor module (0-255).

<versie> = firmware versie van MFM basis module.

3.8.2 Commando Get+SensorInfo

Commando om versie informatie van de MFM sensor module op te vragen.

Commando: Get+SensorInfo

Commando antwoord:

SensorInfo:<sensorId>,<protocol>,<versie>

Per aangesloten sensor volgt een regel.

<sensorId> = Sensormodule slot (1 t/m 6)

<protocol> = protocol versie I2C communicatie van MFM sensor module

<versie> = firmware versie van MFM sensor module

3.8.3 Commando Set+JoinID / Get+JoinID

Commando om het LoRaWAN Join ID (JoinEUI) in te stellen.

Invoer in hexadecimale waarde.

Geaccepteerde waarden: 0x0000000000000000 t/m 0xFFFFFFFFFFFFFFFF

Commando: Set+LoraJoinID=0x0000000000000000

Commando om het LoRaWAN Join ID op te vragen.

Command: Get+JoinID

Commando antwoord: JoinID:<waarde>

<waarde> = Hexadecimaal 64bit waarde.

3.8.4 Commando Set+DeviceID / Get+DeviceID

Commando om het DeviceID in te stellen.

Invoer in hexadecimale waarde.

Geaccepteerde waarden: 0x0000000000000000 t/m 0xFFFFFFFFFFFFFFFF

Commando: Set+DeviceID=0x0000000000000000

Indien 0x0000000000000000 dan wordt het unieke nummer van de controller gebruikt.

Uniek nummer van controller. (64-bit UID in overeenstemming met IEEE 802-2001 standaard)

Commando om het LoRaWAN Device ID (DevEUI) op te vragen.

Commando: Get+DeviceID

Commando antwoord: DeviceID:<waarde>
<waarde> = Hexadecimaal 64bit waarde.

3.8.5 Commando Set+DeviceID

Commando om het LoRaWAN Device ID (DevEUI) in te stellen.

3.8.6 Commando Set+AppKey / Get+AppKey

Commando om de LoRaWAN App key in te stellen.

Invoer in hexadecimale waarde (16 bytes = 128bit).

Geaccepteerde waarden: 0x00000000000000000000000000000000 t/m
0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Commando: Set+AppKey=0x00000000000000000000000000000000

Commando om de LoRaWAN App key op te vragen.

Commando: Get+AppKey

Commando antwoord: AppKey:<waarde>
<waarde> = Hexadecimaal 16 bytes (128bit) waarde.

3.8.7 Commando Set+Sensor / Get+Sensor

Commando om voor een sensor nummer de status te configureren.

Commando: Set+Sensor=<id>,<status>

<id> = sensor slot 1 t/m 6

<status> = sensor actief: 0 = inactief en 1 = actief.

Commando om van een sensornummer het sensor type op te vragen.

Commando: Get+Sensor=1

Commando antwoord: Sensor:<id>,<status>,<type>

<id> = sensor nummer 1 t/m 6

<status> = sensor actief: 0 = inactief en 1 = actief.

<type> = sensor module type 1 t/m 999 (zie H4.4 Sensor types)

3.8.8 Commando Set+LoraInterval / Get+LoraInterval

Commando om het LoRaWAN zend interval van te configureren⁷.

Geaccepteerde waarden: 5 t/m 1440 (decimaal)
Eenheid: minuten.

Commando: Set+LoraInterval=<x>

⁷ Het meetinterval is LoraInterval x aantal actieve sensormodules
aBestand: P22296-10-SPEC-2.0 software.docx
specificatie software Multiflexmeter

Commando om het lees interval op te vragen.

Commando: Get+LoraInterval

Commando antwoord: LoraInterval:<waarde>
<waarde> = decimale waarde in minuten

3.8.9 Commando Set+Samples / Get+Samples

Commando om het aantal meet samples van een sensormodule te configureren.
Het aantal samples bepaald indirect het uitlees moment na inschakelen van de sensormodule.

Geaccepteerde waarden: 1 t/m 100

Commando: Set+Samples=<id>,x

<id> = sensor slot 1 t/m 6

Commando om het aantal samples van een sensormodule uit te lezen.

Commando: Get+Samples=<id>

Commando om het aantal samples van een sensormodule op te vragen.

Commando antwoord: Samples:<id>,<waarde>
<waarde> = decimale waarde.

3.8.10 Commando Get+DataDump

Commando om opgeslagen data van interne opslag uit te lezen (tot 10 dagen).

Commando: Get+DataDump

Commando antwoord: DataDump:<waarde>

<waarde>=aantal regels

Hierna volgen de <waarde> aantal dataregels in komma gescheiden sensor data (CSV).

<tijdstip meting>;<sensor id>;<sensor data>;<CR><LF>

<tijdstip meting> is aantal secondes van activatie systeem. (32 bits decimaal getal)

<sensor id> is nummer van 1 t/m 6

<sensor data> (nader te bepalen).

Nadat laatste dataregel "DataDump:OK<CR><LF>" verstuurd.

3.8.11 Commando Set+AlwaysOn / Get+AlwaysOn

Commando om de 3v3 always on spanning in te schakelen.

De spanning is vervolgens op alle sensor module sloten beschikbaar op een separate aansluiting.

Commando: Set+AlwaysOn=<waarde>

<waarde> = 0 : uit of 1 : aan.

Commando om het de 3v3 always on op te vragen.

Commando: Get+AlwaysOn.

Commando antwoord: LoraInterval:<waarde>
<waarde> = 0 : uit of 1 : aan.

3.8.12 Commando Set+Erase

Commando wist opgeslagen data van interne opslag.
Actie kan afhankelijk van geheugen tot +/- 25 seconde duren.

Commando: Set+Erase

Commando antwoord:

- Voortgang: Set+Erase =<voortgang in percentage><CR><LF> (+/- 1x per seconde)
- Gereed: Erase:OK<CR><LF>

3.8.13 Commando Set+Save

Commando om gewijzigde instellingen op te slaan in Flash van de controller.

Commando: Set+Save

Commando antwoord: Save:OK<CR><LF>

3.8.14 Commando Set+Test

Om de MFM basismodule in test mode te zetten.
Functie wordt automatisch na 5 minuten inactiviteit beëindigt.

Commando: Set+Test=<test nummer>,<waarde>

<test> : decimale waarde 0 t/m 999.

<waarde> : optioneel veld afhankelijk van test

Test	Toelichting	Waarde
1	Opvragen software versies	
2	Opvragen interne ADC	0 t/m 21. 4 = CHANNEL_4_ADC = vBUS 18 = CHANNEL_VREFINT_ADC = 3V3 (vref) 19 = CHANNEL_TEMPSENSOR_ADC
3	LED mode	0 = uit <i>LED 1 is uit</i> <i>LED 2 is uit</i> 1 = test functie <i>LED 1 en LED 2 knipperen om en om.</i> 10 = Ontwikkel mode. <i>LED 1 knippert indien MFM basis module actief</i> <i>LED 2 knippert indien LoRa actief</i>
4	Digitale ingangen test	<nummer>,<status> <nummer> = 0 t/m 56 0 = INT_IO_MCU_IRQ_DETECT 1 = INT_IO_DEBUG_LED1 2 = INT_IO_BOX_OPEN 3 = INT_IO_VDD_TCXO_EXT 4 = INT_IO_DEBUG_LED2 5 = INT_IO_EXP_IRQ 6 = INT_IO_3V3_DETECT 7 = INT_IO_RF_SW_CTRL 8 = EXT_IO_FRAM_CS 9 = EXT_IO_FLASH_SD_CS 10 = EXT_IO_FLASH_NOR_CS 11 = EXT_IO_GE_EN 12 = EXT_IO_BOX_OPEN 13 = EXT_IORST

		14 = EXT_IOMUX_CONNECTED 15 = EXT_IOSENSOR_INTX 16 = EXT_IOPAT_ALERT 17 = EXT_IOVSENSOR_EN 18 = EXT_IOVSYS_EN 19 = EXT_IOALWAYS_EN 20 = EXT_IODEBUG_SW1 21 = EXT_IODEBUG_SW2 22 = EXT_IOSC_EN 23 = EXT_IORTC_PWR_EN 24 = EXT_IOLoadSW1 25 = EXT_IOLoadSW2 26 = EXT_IOSLOT1_GPIO0 27 = EXT_IOSLOT1_GPIO1 28 = EXT_IOSLOT1_GPIO2 29 = EXT_IOSLOT2_GPIO0 30 = EXT_IOSLOT2_GPIO1 31 = EXT_IOSLOT2_GPIO2 32 = EXT_IOWINT1 33 = EXT_IOWINT2 34 = EXT_IOWINT3 35 = EXT_IOWINT4 36 = EXT_IOWINT5 37 = EXT_IOWINT6 38 = EXT_IOSPARE1 39 = EXT_IOSPARE2 40 = EXT_IOLoadSW3 41 = EXT_IOLoadSW4 42 = EXT_IOLoadSW5 43 = EXT_IOLoadSW6 44 = EXT_IOSLOT3_GPIO0 45 = EXT_IOSLOT3_GPIO1 46 = EXT_IOSLOT3_GPIO2 47 = EXT_IOSLOT4_GPIO0 48 = EXT_IOSLOT4_GPIO1 49 = EXT_IOSLOT4_GPIO2 50 = EXT_IOSLOT5_GPIO0 51 = EXT_IOSLOT5_GPIO1 52 = EXT_IOSLOT5_GPIO2 53 = EXT_IOSLOT6_GPIO0 54 = EXT_IOSLOT6_GPIO1 55 = EXT_IOSLOT6_GPIO2
5	Digitale uitgangen test	<nummer>, <status> <nummer> 1 : LED 1 2 : LED 2 7: RF_SW (FE_CTRL3) 8: FRAM CS 9: SD card_CS 10: Dataflash CS 11: Battery monitor enable 17: Sensor enable 18: Vsystem enable 20: debug 1 21: debug 2 24: load switch sensor 1

		25: load switch sensor 2 38: spare 1 39: spare 2 40: load switch extern sensor 3 41: load switch extern sensor 4 42: load switch extern sensor 5 43: load switch extern sensor 6 <status> : 0 = uit, 1 = aan
6	Test SD-card	Functie schrijft een code naar een file en leest deze terug. Indien dit overeenkomt volgt response: Test:6,1,<capaciteit>,<vrije capaciteit> Indien dit mislukt is volgt response Test:6,0,0,0
7	Test RTC	<functie>,<datum>,<tijd> <functie> 1: instellen datum + tijd 2: opvragen datum + tijd 3: synchronisatie RTC -> controller 4: synchronisatie controller -> RTC 5: set wakeup IRQ 5 sec 6: read status register 7: forceer in slaap (zelfs met USB). 8: forceer meting -> LoRA <datum> : alleen igv <functie> = 1. "dd-mm-jjjj" <tijd> : alleen igv <functie> = 1 "uu:mm:ss" Response Voorbeeld instellen datum + tijd Set+Test=7,1,<datum>,<tijd> Test:7,1,<datum>,<tijd> Voorbeeld opvragen datum + tijd Set+Test=7,2 Test:7,2,<datum>,<tijd> Voorbeeld test IRQ > Set+Test=7,5 < Test:7,5 Opvragen status IRQ na 1sec > Set+Test=7,6 < Test:7,6,0x00 Opvragen status IRQ na 5sec > Set+Test=7,6 < Test:7,6,0x01<CR><LF>
8	Test Batterij monitor	<functie> 0: enable gauge 1: opvragen spanning in mV 2: opvragen stroom in µA 3: opvragen temperatuur 4: opvragen SOH

		5: opvragen design capacity 6: opvragen ScaledR 7: opvragen MeasuredZ 8: disable gauge 9: enable battery monitor (GE) 10: disable battery monitor (GE) 11: configureren nieuwe batterij. Response Voorbeeld spanning 3210mV Test:8,1,3210 Voorbeeld stroom 1620 μ A Test:8,2,1620 Voorbeeld capaciteit 76.5% Test:8,3,765
9	Test licht sensor	Response Voorbeeld sensor detecteert geen licht: Test:9,0 Voorbeeld sensor detecteert licht: Test:9,1
10	Test FRAM	Functie schrijft een byte naar het FRAM en en leest deze terug. Indien dit overeenkomt volgt response: Test:10,1,<status> Indien dit mislukt is volgt response Test:10,0,<status> <status> = hexadecimal status register of FRAM. Should be 0x00.
11	Test Dataflash	<functie>,<optie> <functie> 1 = test opgegeven block nummer in <optie> van 0 t/m 2047 2 = test random block nummer 3 = test complete flash (LET OP! duurt enkele minuten) Indien geslaagd. Test:11,1,<result>,<block> Bij fout: Test:11,0,<result>,< block > <result> = 0 = okay, empty 1 = okay, not empty -1 = error, verify write failed -2 = error, verify write original data failed < block > = bloknummer
12	System checks	<test>,<status>

		<test> 1 = vSys 2 = vAlwaysOn 3 = enable slot 1 supply <status> : 0 = uit, 1 = aan
--	--	--

3.8.15 Commando Get+Bat

Commando: Get+Bat

Commando vraagt de batterij status op. De batterijspanning wordt standaard gemeten door de interne ADC van de microcontroller.

Commando antwoord: Bat:<spanning>,<percentage>

<spanning> = batterij spanning in mV

<percentage> = resterende batterijcapaciteit in percentage obv End-Of-Service (EOS) mode.

3.8.16 Commando Get+Vbus

Commando vraagt de VBUS spanning op, gemeten door de interne ADC van de microcontroller.

Commando: Get+Vbus

Commando antwoord: Vbus:<spanning>

<spanning> = spanning in mV

3.8.17 Commando Get+Vcc

Commando vraagt de Vcc spanning op, gemeten door de interne ADC van de microcontroller.

Commando: Get+Vcc

Commando antwoord: Vcc:<spanning>

<spanning> = spanning in mV

3.8.18 Commando Set+Reboot

Commando voor een reboot van de microcontroller.

Commando: Set+Reboot

Commando antwoord: Reboot:OK

3.8.19 Commando Set+InitSensor

Commando om de sensoren opnieuw te initialiseren bij eerstvolgende opvraag.

Commando: Set+InitSensor

Commando antwoord: InitSensor:OK

3.8.20 Commando Set+Rejoin

Commando om een rejoin te forceren. Argument is optioneel.

Commando: Set+Rejoin of Set+Rejoin=<delay>

<delay> = wachttijd voor rejoin in secondes.

Commando antwoord: Rejoin:OK

3.8.21 Commando Set+Eos

Commando om batterij EOS data te forceren om bij eerst volgend interval ipv te wachten totdat het eerstvolgende meetmoment (1x 24 uur).

Commando: Set+Eos

Commando antwoord: Eos:OK

3.8.22 Commando Get+Nonce / Set+Nonce

Commando om Nonce counter op te vragen en te wijzigen.

Commando: Get+Nonce of Get+Nonce=0

Commando antwoord: Nonce:0,<DevNonce>,<JoinNonce>,<DnFCnt>,<UpFCnt><CR><LF>

Commando: Get+Nonce=1

Commando antwoord: Nonce:1,<DevNonce><CR><LF>

Commando: Get+Nonce=2

Commando antwoord: Nonce:2,<JoinNonce><CR><LF>

Commando: Get+Nonce=3

Commando antwoord: Nonce:3,< DnFCnt ><CR><LF>

Commando: Get+Nonce=4

Commando antwoord: Nonce:4,< UpFCnt ><CR><LF>

Commando om Nonce counter in te stellen.

Commando: Set+Nonce=0,<DevNonce>,<JoinNonce>,<DnFCnt>,<UpFCnt><CR><LF>

Commando antwoord: Nonce:OK,0,<DevNonce>,<JoinNonce>,<DnFCnt>,<UpFCnt><CR><LF>

Commando: Set+Nonce=1,<DevNonce><CR><LF>

Commando antwoord: Nonce:OK,1,<DevNonce><CR><LF>

Commando: Set+Nonce=2,< JoinNonce ><CR><LF>

Commando antwoord: Nonce:OK,2,< JoinNonce ><CR><LF>

Commando: Set+Nonce=3,< DnFCnt><CR><LF>

Commando antwoord: Nonce:OK,3,< DnFCnt><CR><LF>

Commando: Set+Nonce=4,< UpFCnt ><CR><LF>

Commando antwoord: Nonce:OK,4,< UpFCnt ><CR><LF>

1 = <DevNonce> = Device nonce counter.

2 = <JoinNonce> = Join nonce counter

3 = <DnFCnt> = Download frame counter

4 = <UpFCnt> = Upload frame counter

3.9 Configuratie

3.9.1 Parameters

1. LoRaWAN JoinID. hexadecimaal 16 karakters (64 bit).
2. LoRaWAN DeviceID. hexadecimaal 16 karakters (64 bit).
3. LoRaWAN AppKey. hexadecimaal 32 karakters (128 bit).
4. Meet interval. Uint16 decimaal 5 t/m 1440 (minuten) .
5. AlwaysOn spanning (aan / uit) .
6. Sensor (6x).
 - a. Sensor module type ⁸: uint8
 - b. Sensor module status: uint8 (aan = 0x01 of uit = 0x00)
 - c. Aantal samples: uint8

3.9.2 Opslag locatie

De configuratie parameters worden opgeslagen in het flash van de processor.

3.10 Meetdata

Het dataopslag wordt als ringbuffer gebruikt, waarbij records ter grootte van één pagina. Dit maakt het opslag en wis algoritme eenvoudiger. Ook geeft dit de mogelijkheid om te werken met variabele data grootte voor verschillende modules, als het maar binnen één pagina van 256 bytes past. Wissen wordt uitgevoerd met een 4k blok (16 pagina's).

3.10.1 Blok definitie dataflash

Er wordt uitgegaan van de volgende indeling van de dataflash.

64 Mbit flash met 32768 pagina's van 256 bytes = 8388608 bytes = 67108864 bits.
1 pagina = 256 bytes = 2048 bits.

Het flash kan volledig gewist worden en per block van 4k, 32k en 64k.
4k = 16 pagina's = 4096 bytes = 32768 bits.
32k = 128 pagina's = 32768 bytes = 262144 bits.
64k = 256 pagina's = 65536 bytes = 524288 bits.

3.10.2 Wis tijden

Rekening houden met de volgende wis tijden.
Volledige wissen chip 25 seconden.
Wissen 64 kbyte blok 200ms
Wissen 32 kbyte blok 120ms
Wissen 4 kbyte blok 60ms

3.10.3 Indeling meetdata pagina

De eerste 127 pagina's van 32k zijn gereserveerd voor de meetdata.
De overgebleven pagina van 32k is gereserveerd voor toekomstige functionaliteit.

Per pagina (256 bytes) wordt weggeschreven:

- Meetnummer (uint32)
- Timestamp (unix timestamp⁹) (uint32)
- Protocol ID MFM
- Sensor slot ID (1 t/m 6, uint8)
- Type sensormodule (uint8)

⁸ Bepaald door sensormodule

⁹ Unix timestamp uint32 loopt tot +/- jaar 2106.

- Protocol ID sensormodule (uint8)
- Sensor data grootte (uint8)
- Sensor data (36 bytes)¹⁰
- CRC meetdata (2 bytes)¹¹
- MFM base data (9 bytes)
- Spare data (196 bytes)

Rekening houdend met de programmeer en wis cycli is een meetnummer van 4 bytes afdoende.

Het maximaal aantal metingen totdat de 100.000x programmeer wis cycli bereikt wordt is:
 $32768 \text{ pagina's} \times 100.000 = 3.276.800.000$. Dit getal past binnen uint32.

Minimaal 10 dagen opslag bij 1x per 15 minuten een meting.
 $(32768 - 128) / (4 \times 24) = 340 \text{ dagen}$

Doordat het een energie zuinige applicatie betreft, dient er rekening gehouden te worden dat de controller RAM spanningsloos gemaakt wordt.

Dit betekent dat de administratie van metingen niet onthouden kan worden in RAM van de controller. Het opslaan van de administratie in dataflash is niet mogelijk vanwege de maximaal aantal programmeer en wis cycli, wat de levensduur zou verkorten.

De meet administratie zal opgeslagen worden in de backup registers welke behouden blijven zolang VBAT is aangesloten van de hoofdcontroller.

Daarnaast zal een zoekalgoritme (bisectie) ervoor zorg dragen dat bij ontbreken van de administratie, deze bepaald wordt door het dataflash uit te lezen.

3.10.4 Administratie meetdata

De administratie van de meetdata is als volgt gedefinieerd:

Item	Opmerkingen
Oudste meetnummer	Begin bij 1 en wordt indien <nieuwste meetnummer> > 32750 geupdate adhv het wissen en nieuwe data (ivm ringbuffer).
Nieuwste meetnummer	Wordt opgehoogd bij nieuwe meting.

Zie H3.11 (Indeling backup registers) voor de locatie van bovenstaande items.

3.10.5 Schrijftijd meting

De schrijftijd van een meting hangt af of de pagina welke beschreven gaat worden al gewist is of nog data bevat. Als de ringbuffer rond is, dit is het geval indien backup register [0] <nieuw meetnummer> = backup register [1] <eerste meetnummer> + 32.750, dan zal de schrijftijd +/- 60ms langer zijn bij 1 op de 16 metingen. Er moet immers een block van 4k (16 pagina's) gewist worden.

3.10.6 Wissen meetdata

Indien het Commando Set+Erase (H3.8.12) uitgevoerd wordt, zal eerst het backup register[1] <nieuw meetnummer> opgehoogd worden tot een veelvoud van 16 pagina's. Er kan immers alleen gewist worden in blokken van 4k (16 pagina's), 32k (128 pagina's) of 64k (256 pagina's).

Vervolgens worden alle pagina's van <eerste meetnummer> tot <nieuw meetnummer> gewist. Als laatste worden backup register [0] <Eerste meetnummer> en backup register [2] <Verstuurd meetnummer> beide gelijk gemaakt aan backup register [1] <Nieuw meetnummer>.

Opmerking: het wissen van meetdata geeft afhankelijk van het meetnummer voor 0 tot 15 pagina's één extra wis cycli.

¹⁰ Niet gebruikte bytes welke niet geschreven worden blijven 0xFF

¹¹ Berekend over Record sensor slot ID t/m Sensor data[grootte], exclusief niet gebruikte bytes.

3.10.7 Koude start

Indien het systeem volledig spanningsloos is geweest (volledige koude start), dan staan er geen geldige waarden meer in de backup registers.

Het dataflash wordt dan dmv een bisectie zoek algoritme gescand, er wordt gestart op het halve adres (pagina 16384) om de pagina te lezen. Bij het vinden van correcte data of waarde groter dan de vorige wordt het adres vergroot met de helft, zo niet verkleind met de helft. Dit wordt herhaald totdat de laatste registraties is teruggevonden. Hiermee is ook gelijk de laatste timestamp bekend, welke igv alive counter overgenomen kan worden als startwaarde.

Het zoek proces wordt ook uitgevoerd voor het oudste meetnummer.

3.11 Indeling backup registers

Register	Toepassing	Item
0	RTC (timer_if.c)	RTC seconds
1	RTC (timer_if.c)	RTC sub seconds
2	RTC (timer_if.c)	RTC MSB ticks
3	Administratie meetdata	Oudste meetnummer
4	Administratie meetdata	Nieuw meetnummer
5	Tijdstip van laatste tijd synchronisatie portaal	Tijd in secondes van laatste synchronisatie.
6	Rejoin status gezet bij voorafgaand interval na ontvangst commando.	Status vlag tbv geforceerde rejoin bij opstart.
7	Batterij informatie. EOS meet vlag gezet voorafgaand interval.	EOS % waarde (uint8) Spanning mV (uint16) Batterij EOS meet vlag
8	Status bits	Bit 0: testmode actief Bit 1-31: ongebruikt
9	Last wakeup by reset	Seconds: unix timestamp
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		

3.12 Broncode

De broncode van het MFM sensormodule wordt opgeleverd dmv een git repository.

4 Communicatie MFM basis module <-> MFM sensor module

Communicatie tussen MFM basis module en MFM sensor module verloopt via I2C protocol in standard mode op 400khz.¹²

Er is gekozen voor I2C omdat:

- I2C bus een industrie standaard is ondersteund door veel devices.
- Beperkt aantal pinnen vereist. Twee: SDA en SCL.
- Eenvoudig topologie: één master, meerdere slaves. De master bepaald wanneer welke module opgevraagd gaat worden, waardoor de volledige controle in de MFM basis module ligt.
- Eenvoudig uit te breiden met meerdere slaves (tot 128 adressen).

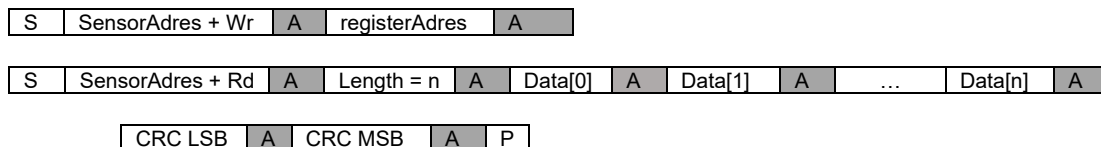
4.1 Bericht definitie

De berichten worden onderverdeeld in lees en schrijf commando's.

S = START
R = RESTART
A = ACK
N = NACK
P = STOP

4.1.1 Lees commando's

uint8 registerAdres
uint8 dataLength¹³
uint8 data[0..255]
uint16 CRC

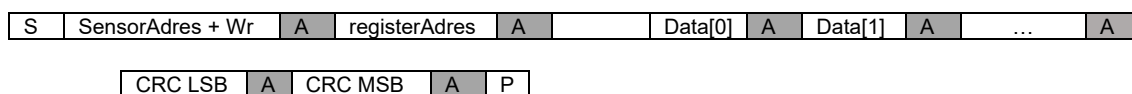


CRC is berekend over lengte + data array.

4.1.2 Schrijf commando's

uint8 registerAdres

uint8 data[0..255]
uint16 CRC



CRC is berekend over registeradres + data array.

Na schrijven wordt het commando terug gelezen, voor verificatie dat de data correct is overgedragen.

4.2 Sensor commando's

Alle commando's gedefinieerd als schrijf commando kunnen ook uitgelezen worden als leescommando. Dit staat niet apart beschreven.

¹² De controller ondersteund "normal mode" (100khz), "fast mode" (400khz) en "fast mode plus" (1000khz), het gebruik van hoge snelheden wordt afgeraden doordat de bus ook van print naar print wordt gebruikt met bekabeling.

¹³ Alleen van toepassing voor commando's met variabele lengte.

4.2.1 Lees sensor firmware versie

Het opvragen van de firmware versie van de sensor module.

< registerAddress > = 0x01
< data[0] > t/m < data[9] > : Versie nummer in ASCII karakters

4.2.2 Lees protocol versie

Het opvragen van protocol versie van de sensor met :

< registerAddress > = 0x02
< data[0] > = 0

4.2.3 Lees sensor type

Het opvragen van type sensor met :

< registerAddress > = 0x03
< data[0] > = sensor module type

(zie H4.4 Sensor types)

4.2.4 Start initialisatie module

Het starten van een initialisatie cycle van de geselecteerde sensor wordt uitgevoerd met :

< registerAdres > = 0x0A
< data[0] > = 1

Hierbij wordt "status" gereset.

Het (voortijdig) beëindigen initialisatie cycle wordt uitgevoerd met :

< registerAdres > = 0x0A
< data[0] > = 0

Opmerking:

- Aantal sensoren uitlezen met commando zie 4.3.1 Lees aantal sensors.
- Sensor selecteren met commando zie H4.3.2 Schrijf selectie van sensor.

4.2.5 Status initialisatie module

Het opvragen van de status (van de laatste) actieve initialisatie cycle wordt uitgevoerd met :

< registerAdres > = 0x0B
< data[0] > = status : 0x00 = geen init, 0x01 = bezig, 0x0A = gereed, 0xFF = niet beschikbaar

Na een power cycle is er geen historische status beschikbaar en is de status 0x00.

4.2.6 Schrijf start / stop meting

Het starten van een meting wordt uitgevoerd met :

< registerAdres > = 0x10
< data[0] > = 1

Het (voortijdig) stoppen van een meting die wordt uitgevoerd met :

< registerAdres > = 0x10
< data[0] > = 0

4.2.7 Lees status meting

Het opvragen van de status van de meting met :

```
< registerAddress > = 0x11  
< data[0] > = status : 0x00 = geen meting, 0x01 = bezig, 0x0A = gereed.
```

4.2.8 Lees meettijd

Het opvragen van de meettijd van een sensormodule, berekend door de sensormodule obv het aantal meet samples, maximaal aantal sensoren en de vaste sensor parameters gedefinieerd in de firmware van de sensor module.

```
< registerAddress > 0x12  
< data[0] > = <meetijd> (LSB)  
< data[1] > = <meetijd> (MSB)
```

<meetijd> : uint16 in milliseconde.

Voorbeeld:

```
<meetijd> = 500 -> 0x01F4  
< data[0] > = 0xF4  
< data[1] > = 0x01
```

4.2.9 Lees meetdata

Het opvragen van de meetdata van de meting die is uitgevoerd met :

```
< registerAddress > = 0x20  
< data[0] > = <data grootte>  
< data[1] > = data byte 1  
< data[..] > =  
< data[X+1] > = data byte X
```

Indien status meting (H0) niet "gereed" dan is de opgevraagde data niet betrouwbaar.

4.3 Test commando's

Onderstaande commando's zijn niet noodzakelijk voor goede werking icm de MFM basismodule. Deze commando's zijn wel aan te bevelen om te implementeren zodat de werking van de sensormodule te verifiëren is.

4.3.1 Lees aantal sensors

Een sensormodule kan één of meerdere dezelfde of verschillende sensoren hebben.

Het opvragen van het aantal beschikbare sensors van een sensormodule met:

```
< registerAddress > = 0x30  
< data[0] > = <aantal>.
```

4.3.2 Schrijf selectie van sensor.

Het instellen van een sensor waarvan afzonderlijke (ruwe) meetdata opgevraagd kan worden voor test doeleinden met:

```
< registerAddress > = 0x31  
< data[0] > = sensor nummer (default 0).
```

4.3.3 Schrijf type meetwaarde

Het instellen van type meetwaarde.

< registerAddress > = 0x32

< data[0] > = <type meetwaarde>.

<type meetwaarde> : uint8 (zie ook H4.5 Type meetwaarde)

4.3.4 Schrijf / lees meetsamples per meting

Het opvragen van het aantal meetsamples per meting van de ingestelde sensor met:

< registerAddress > = 0x33

< data[0] > = <aantal samples> (LSB).

< data[1] > = <aantal samples> (MSB)

<aantal samples> : uint16

4.3.5 Lees sensor waarde van geselecteerde sensor

Het opvragen van ruwe sensor waarde van de ingestelde sensor met:

< registerAddress > = 0x38

< data[0] > = <data grootte>

< data[1] > = LSB data byte

< data[..] > =

< data[X] > = MSB data byte

4.3.6 Schrijf uitgang slot GPIO/INT

ALLEEN VOOR TEST DOELEINDEN.

Het schrijven van de status van IO pin, indien deze als uitgang geconfigureerd is:

< registerAddress > = 0x40

< data[0] > =

Bit0 = SLOT GPIO 0

Bit1 = SLOT GPIO 1

Bit2 = SLOT GPIO 2

Bit3 = INT

4.3.7 Schrijf ingang/uitgang voor slot GPIO

ALLEEN VOOR TEST DOELEINDEN.

Het schrijven van type I/O voor slot GPIO: uitgang of ingang.

Default: Ingang

Opmerking: Uitgang alleen te gebruiken in test scenario waarbij geverifieerd dient te worden dat MFM-main zijde als input staat.

< registerAddress > = 0x41

< data[0] > =

Bit0 = SLOT GPIO 0

Bit1 = SLOT GPIO 1

Bit2 = SLOT GPIO 2

4.4 Sensor types

De volgende sensor types zijn gedefinieerd.

0x00 = niet gebruiken
0x01 = MFM Drukmodule RS485
0x02 = MFM Drukmodule One-Wire
....
0xFF = nader te bepalen

4.5 Type meetwaarde

De volgende type meetwaardes zijn gedefinieerd en te gebruiken in de sensormodule.

0x00 = meetwaarde in sensor eenheid (default)
0x10 = gemiddelde meetwaarde in sensor eenheid
0x20 = mediaan meetwaarde in sensor eenheid

Onderstaande alleen als sensor module gebaseerd is ADC meting.

0x01 = ruwe adc waarde in bits
0x11 = gemiddelde adc waarde in bits
0x21 = mediaan adc waarde in bits

0x02 = ruwe adc waarde in mV
0x12 = gemiddelde adc waarde in mV
0x22 = mediaan adc waarde in mV

4.6 Error code

Het opvragen van de error code met:

```
< registerAddress > = 0x51  
< data[0] > = <error code>
```

Gedefinieerde waarden:

0x00 = geen fout

0x01 = CRC fout

0x02 = Register adres fout / onbekend

0x03 = Schrijffout. Register wat geschreven is heeft geen schrijf rechten.

5 MFM sensor module : Drukmodule

De drukmodule is een sensor module met twee digitale ingangen voor stroomsensoren.

Bij inschakelen van de module zal de software de digitale ingangen gaan uitlezen.

De ruwe waardes zijn uit te lezen via onderstaande registers.

Als de module een start commando ontvangen heeft wordt een meting gestart van de ingestelde aantal meetsamples. Als de meting gereed is wordt de meetwaarde opgeslagen in het meetdata register en verandert de meetstatus in status "gereed".

De MFM sensor module blijft de analoge ingangen bemonsteren totdat de module uitschakelt.

Nieuwe meetdata wordt niet automatisch opgeslagen in de meetdata (register 0x21), hiervoor moet een nieuwe meting gestart worden (register 0x10=0x01).

Om te voorkomen dat tijdens lezen van een register de inhoud gewijzigd kan worden, wordt bij uitlezing een schaduw kopie gebruikt welke geblokkeerd is tijdens uitlezing. Dit geldt alleen voor registers waarvan data gewijzigd kan worden en groter zijn dan één byte.

Register	Toelichting	Standaard waarde	Rechten	Data type
0x01	Firmware versie	"x.y"	Lezen	UInt8[10]
0x02	Protocol versie	0x00	Lezen	UInt8
0x03	Sensormodule type	0x01	Lezen	UInt16
0x0A	Start initialisatie	0x00	Lezen / Schrijven	UInt8
0x0B	Status initialisatie	0x00	Lezen	
0x10	Start meten	0x00	Lezen / Schrijven	UInt8
0x11	Status meting	0x00	Lezen	UInt8
0x12	Meettijd	0x0064	Lezen	UInt16
0x20	Meetdata	-	Lezen	UInt8[x]
0x30	Aantal sensoren	0x02	Lezen	UInt8
0x31	Geselecteerde sensor	0x00	Lezen / Schrijven	UInt8
0x32	Type meetwaarde	0x00	Lezen / Schrijven	UInt8
0x33	Aantal meetsamples	0x000A	Lezen / Schrijven	UInt16
0x38	Sensor waarde	-	Lezen	UInt16
0x51	Error code	-	Lezen	UInt8

5.1 Meetdata RS485

Data wordt ingelezen per sensor via modbus adres 0x100 met lengte van 4x uint16 (word).
Waardes worden niet omgerekend, alleen de mediaan wordt bepaald over het ingestelde aantal meetsamples.

Zie documentatie betreffende sensor voor meer informatie.

Byte	Item	Type	value
data[0]	data grootte	uint8	0x10
data[1]	druk sensor 1 in bar	float32	LSB
data[2]			
data[3]			
data[4]			MSB
data[5]	temperatuur sensor 1 in graden Celsius	float32	LSB
data[6]			
data[7]			
data[8]			MSB
data[9]	druk sensor 2 in bar	float32	LSB
data[10]			
data[11]			
data[12]			MSB
data[13]	temperatuur sensor 2 in graden Celsius	float32	LSB
data[14]			
data[15]			
data[16]			MSB
data[17]	CRC	Uint16	LSB
data[18]			MSB

Voorbeeld:

Temperatuur sensor 1 = 20.1 -> 0xCDCCA041

< data[5] > = 0xCD

< data[6] > = 0xCC

< data[7] > = 0xA0

< data[8] > = 0x41

5.2 Meetdata onewire

Data wordt ingelezen in de ontvangen dataframes (ZACwire).
Waardes worden niet omgerekend, alleen de mediaan wordt bepaald over het ingestelde aantal meetsamples.

Zie documentatie betreffende sensor voor meer informatie.

Byte	Item	Type	Toelichting	value
data[0]	data grootte	uint8		0x06
data[1]	druk sensor 1	uint16	3000 = 0% 11000 = 100%	LSB
data[2]				MSB
data[3]	temperatuur sensor 1	uint8	0 = -50°C, 255 = +150°C	
data[4]	druk sensor 2	uint16	3000 = 0% 11000 = 100%	LSB
data[5]				MSB
data[6]	temperatuur sensor 2	uint8	0 = -50°C, 255 = +150°C	

5.3 Variant

5.3.1 Drukmodule RS485

Sensortype: 0x01

Geverifieerd met Keller: PR-26x

Opvraagtijd sensormodule bij 10 meetsamples:

- 1 sensor: +/- 200ms
- 2 sensoren: +/- 300ms

5.3.2 Drukmodule One-wire

Sensortype: 0x02

Geverifieerd met Huba sensor: 713.D100302110

Opvraagtijd sensormodule bij 10 meetsamples:

- 1 sensor: +/- 75ms
- 2 sensoren: +/- 100ms

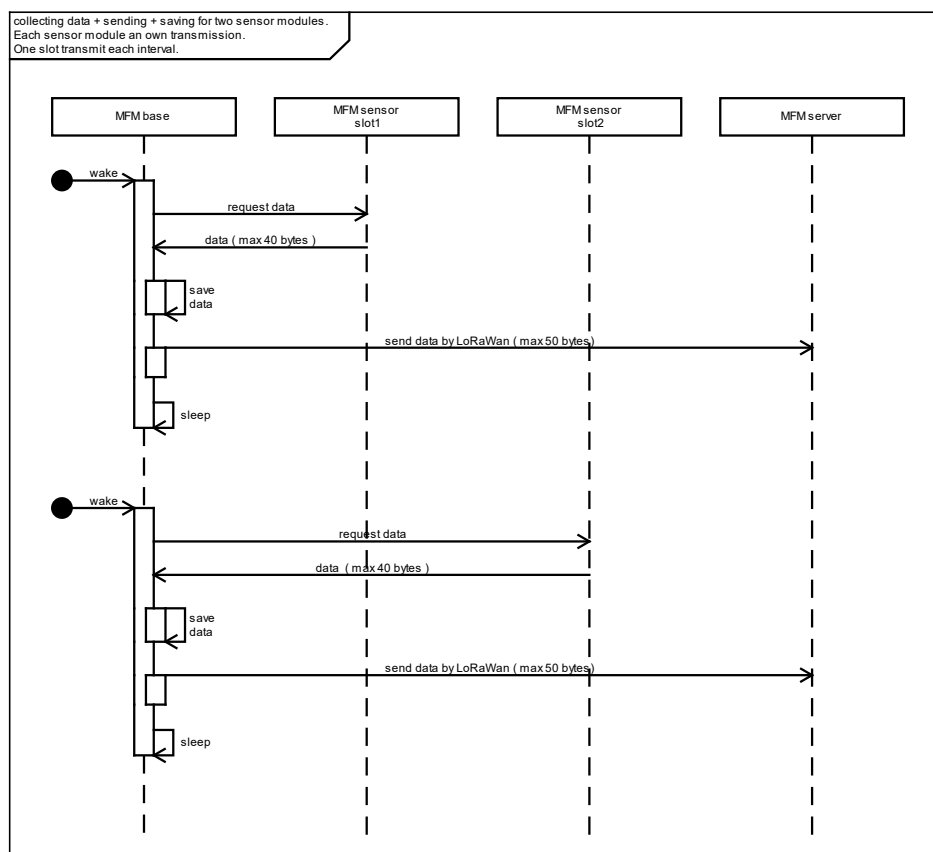
5.4 Broncode

De broncode van het MFM sensormodule wordt opgeleverd dmv een git repository.

6 LoRaWAN communicatie

Onderstaande afbeelding geeft de een beeld van de communicatie igv 2 sensor modules. De data wordt van één module opgevraagd en daarna via LoRaWAN verstuurd. Het volgende interval wordt de volgende sensor opgevraagd.

Alleen actieve sensoren (obv configuratie) zullen opgevraagd worden.



6.1 MFM data bericht

Standaard unconfirmed berichten met de mogelijkheid om 1x per X berichten een confirmed bericht te sturen.¹⁴

Er kan maximaal 51 bytes verstuurd worden, naast de 13 bytes MFM header. Deze limitatie komt door rekening te houden met worstcase SF12.

Voor de MFM basis module zijn 10 bytes gereserveerd, waarvan 1 byte aan start (verplicht) en 9 bytes optioneel na sensor data.

Voor de MFM sensor module zijn 40 bytes gereserveerd, waarvan 4 bytes sensor header en 36 bytes sensordata.

Het bericht wordt als volgt opgebouwd:

(uint8) <protocol MFM>
(uint8) <sensor module slot ID>
(uint8) <sensor module type ID>
(uint8) <sensor module protocol ID>

¹⁴ Ingesteld op 1x per 24 berichten per sensormodule
aBestand: P22296-10-SPEC-2.0 software.docx
specificatie software Multiflexmeter

(uint8) <sensor data grootte> = N
 (uint8) <sensor data [0]>
 ...
 (uint8) <sensor data [N-1]>
 (uint8) <basis data[0]> : <message type> optioneel
 ...
 (uint8) <basis data[8]> : optioneel

Afhankelijk van <message type> wordt de laatste 8 bytes ingevuld.

Message type	Indeling
0x00	Geen data
0x01	[1]: <uint8> batterij status EOS [2]: <int8> temperatuur batterij monitor [3]: <int8> temperatuur controller [4]: <uint8> diagnostische status bits
0x02	[1]: <int8> temperatuur controller [2]: <uint8> diagnostische status bits
0x03	Gereserveerd voor toekomstig gebruik
....
0xFF	Gereserveerd voor toekomstig gebruik

6.1.1 MFM data velden

Data veld	Bereik	Toelichting
batterij status EOS	0 : 100	End of service of battery percentage
temperatuur batterij monitor	128 : +127	In hele graden Celsius
temperatuur controller	-128 : +127	In hele graden Celsius
diagnostische bits		Bit 0: licht sensor actief Bit 1: USB verbonden Bit 2: batterij laag Bit 3: Sensorboard channel 1 init failed Bit 4: Sensorboard channel 2 init failed

6.1.2 MFM diagnostische status bits

Diagnostische status bits	Bit	Toelichting
Licht sensor actief	0	Ingelezen via I/O "BOX_OPEN"
USB verbonden	1	Ingelezen via I/O "USB_connected"
Batterij laag	2	Ingelezen via I/O "BAT_ALERT"

Diagnostische bits worden bij het wakker worden van de controller ingelezen en tijdelijke opgeslagen. Bij een LoRa transmissie worden de actuele status en tijdelijk opgeslagen status als logische OR verstuurd. Na zenden wordt de tijdelijk opgeslagen waarde gewist.

6.2 Join

Een join wordt procedure zal worden gestart met SF10 (1x). Indien dit faalt wordt er overgegaan op SF11 (1x), als deze ook faalt wordt er herhaaldelijk keer op SF12 verbonden. Per join wordt het totaal 6x herhaald.

6.3 Tijd synchronisatie

Periodiek wordt de tijd opgevraagd van de LoRa server, na ontvangst wordt dit gesynchroniseerd met de extern RTC. Het interval is ingesteld op 1x per 24 berichten per sensormodule. Ook wordt de tijd opgevraagd na detecteren van BAT status bit in status register van de RTC.

6.4 Uplink bericht

Er wordt in voorzien om uplink berichten te ontvangen van de server, hierbij wordt uitgegaan van minimaal twee bytes data naast de <FPort>.

De <FPort> 0 is gereserveerd vanuit de LoRaWan specificatie. Daarnaast is <FPort> 224 is gereserveerd voor MAC compliance testen en <FPort> 225 – 255 zijn gereserveerd voor uitbreidingen vanuit de norm.

Voor het MFM uplink bericht wordt <Fport> 105 (0x69) toegepast

< FPort > = commando activatie byte (= 0x69)

< byte 1 > = commando byte

< byte 2 > = optioneel variabele bij commando

< byte 3 > = optioneel variabele bij commando

<FPort>	Commando byte	Optionele byte	Toelichting
0x69	0x55	<byte>	Rejoin basismodule volgend interval Bit 0: hoog = reset DevNonce Bit 1: hoog = reset JoinNonce Bit 2: hoog = reset DownFrameCntr Bit 3: hoog = reset UpFrameCntr Bit 4: hoog = activeer sensor init ¹⁵ Indien optionele byte niet meegestuurd wordt, worden bits acties niet uitgevoerd.
0x69	0x56	<MSB> <LSB>	Wijzigen zend interval (5 t/m 1440) 2 optionele databytes verplicht
0x03	0x00	-	Instellen Device Class A
0x03	0x01	-	Instellen Device Class B ¹⁶
0x03	0x02	-	Instellen Device Class C ¹⁶
0x02	0x01	-	Debug LED1 blink aan ¹⁷
0x02	0x00	-	Debug LED1 uit

¹⁵ Zie meer info "Start initialisatie module" H4.2.4

¹⁶ Ondersteund door de LoRaWAN driver. Geen onderdeel van project scope, wordt niet meegenomen met testen.

¹⁷ Hardware soldeer jumper bepaald of LED gebruikt kan worden.

7 Tool en licenties

Voor de MFM basis module en sensor module wordt er gebruik gemaakt van STM32CubeIDE.
<https://www.st.com/en/development-tools/stm32cubeide.html>

Standaard wordt er little-endian toegepast.

Onderdelen van de source vallen onder licenties welke hieronder genoemd zijn.

7.1 Licentie STMicroelectronics

Voor code afkomstig van STMicroelectronics geldt onderstaande licentie:

SLA0048 Rev4/March 2018

BY INSTALLING COPYING, DOWNLOADING, ACCESSING OR OTHERWISE USING THIS SOFTWARE PACKAGE OR ANY PART THEREOF (AND THE RELATED DOCUMENTATION) FROM STMICROELECTRONICS INTERNATIONAL N.V, SWISS BRANCH AND/OR ITS AFFILIATED COMPANIES (STMICROELECTRONICS), THE RECIPIENT, ON BEHALF OF HIMSELF OR HERSELF, OR ON BEHALF OF ANY ENTITY BY WHICH SUCH RECIPIENT IS EMPLOYED AND/OR ENGAGED AGREES TO BE BOUND BY THIS SOFTWARE PACKAGE LICENSE AGREEMENT.

Under STMicroelectronics' intellectual property rights and subject to applicable licensing terms for any third-party software incorporated in this software package and applicable Open Source Terms (as defined here below), the redistribution, reproduction and use in source and binary forms of the software package or any part thereof, with or without modification, are permitted provided that the following conditions are met:

1. Redistribution of source code (modified or not) must retain any copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form, except as embedded into microcontroller or microprocessor device manufactured by or for STMicroelectronics or a software update for such device, must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of other contributors to this software package may be used to endorse or promote products derived from this software package or part thereof without specific written permission.
4. This software package or any part thereof, including modifications and/or derivative works of this software package, must be used and execute solely and exclusively on or in combination with a microcontroller or a microprocessor devices manufactured by or for STMicroelectronics.
5. No use, reproduction or redistribution of this software package partially or totally may be done in any manner that would subject this software package to any Open Source Terms (as defined below).
6. Some portion of the software package may contain software subject to Open Source Terms (as defined below) applicable for each such portion ("Open Source Software"), as further specified in the software package. Such Open Source Software is supplied under the applicable Open Source Terms and is not subject to the terms and conditions of license hereunder. "Open Source Terms" shall mean any open source license which requires as part of distribution of software that the source code of such software is distributed therewith or otherwise made available, or open source license that substantially complies with the Open Source definition specified at www.opensource.org and any other comparable open source license such as for example GNU General Public License (GPL), Eclipse Public License (EPL), Apache Software License, BSD license and MIT license.
7. This software package may also include third party software as expressly specified in the software package subject to specific license terms from such third parties. Such third party software is supplied under such specific license terms and is not subject to the terms and conditions of license hereunder. By installing copying, downloading, accessing or otherwise using this software package, the recipient agrees to be bound by such license terms with regard to such third party software.
8. STMicroelectronics has no obligation to provide any maintenance, support or updates for the software package.
9. The software package is and will remain the exclusive property of STMicroelectronics and its licensors. The recipient will not take any action that jeopardizes STMicroelectronics and its licensors' proprietary rights or acquire any rights in the software package, except the limited rights specified hereunder.

10. The recipient shall comply with all applicable laws and regulations affecting the use of the software package or any part thereof including any applicable export control law or regulation.

11. Redistribution and use of this software package partially or any part thereof other than as permitted under this license is void and will automatically terminate your rights under this license.

THIS SOFTWARE PACKAGE IS PROVIDED BY STMICROELECTRONICS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS, IMPLIED OR STATUTORY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS ARE DISCLAIMED TO THE FULLEST EXTENT PERMITTED BY LAW. IN NO EVENT SHALL STMICROELECTRONICS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE PACKAGE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

EXCEPT AS EXPRESSLY PERMITTED HEREUNDER AND SUBJECT TO THE APPLICABLE LICENSING TERMS FOR ANY THIRD-PARTY SOFTWARE INCORPORATED IN THE SOFTWARE PACKAGE AND OPEN SOURCE TERMS AS APPLICABLE, NO LICENSE OR OTHER RIGHTS, WHETHER EXPRESS OR IMPLIED, ARE GRANTED UNDER ANY PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF STMICROELECTRONICS OR ANY THIRD PARTY.

7.2 Licentie STM32WLxx_Nucleo

This software component is provided to you as part of a software package and applicable license terms are in the Package_license file. If you received this software component outside of a package or without applicable license terms, the terms of the BSD-3-Clause license shall apply.

You may obtain a copy of the BSD-3-Clause at:

<https://opensource.org/licenses/BSD-3-Clause>

7.3 STM32WLxx_HAL_Driver

This software component is provided to you as part of a software package and applicable license terms are in the Package_license file. If you received this software component outside of a package or without applicable license terms, the terms of the BSD-3-Clause license shall apply.

You may obtain a copy of the BSD-3-Clause at:

<https://opensource.org/licenses/BSD-3-Clause>

7.4 Licentie LoRaWan driver

This software component is provided to you as part of a software package and applicable license terms are in the Package_license file. If you received this software component outside of a package or without applicable license terms, the terms of the BSD-3-Clause license shall apply.

You may obtain a copy of the BSD-3-Clause at:

<https://opensource.org/licenses/BSD-3-Clause>

7.5 Licentie SubGhz_Phy (radio) driver

Het onderdeel SubGhz_Phy (radio driver) valt onder een BSD licentie.

The Clear BSD License

Copyright Semtech Corporation 2021. All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted (subject to the limitations in the disclaimer below) provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the Semtech corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

NO EXPRESS OR IMPLIED LICENSES TO ANY PARTY'S PATENT RIGHTS ARE GRANTED BY THIS LICENSE. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL SEMTECH CORPORATION BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Licentie CMSIS

CMSIS driver valt onder Apache licentie 2.0 (januari 2004)

Zie: <https://www.apache.org/licenses/LICENSE-2.0>

8 Configuratie software MFM basis module

8.1 Ontwikkeling

Het configuratie programma wordt ontwikkeld voor Windows 10 / Windows 11.
Het programma is geschreven in C# met gebruik van Visual Studio.

Het configuratie programma is volledig in het Engels.

8.2 Functioneren (gebruikers interactie)

De gebruiker kan de MFM basis module verbinden met de beschikbare USB aansluiting van de MFM-basis module naar een PC waarop de configuratie software geïnstalleerd is.

Na het starten van de configuratie software kan de gebruiker op het hoofdscherm een COM-poort selecteren. Hiermee wordt gecommuniceerd met de MFM basis module. Het hangt af van de PC configuratie af welke COM-poort nummer dit is en hoeveel COM-poorten getoond worden.

De gebruiker kan indien gewenst ook de baudrate instellingen van de COM-poort aanpassen. Gezien dat de MFM basis module op een vaste baudrate geconfigureerd is, zullen deze via een sub scherm instelbaar zijn.

Het hoofdscherm bevat velden van configuratie en uitleesbare items.

1. LoRaWAN JoinID. Type : invoer . Formaat : hexadecimaal 16 karakters (64 bit).
 - a. Checkbox voor automatisch 64bit key te genereren.
2. LoRaWAN DeviceID. Type : Formaat : hexadecimaal 16 karakters (64 bit).
 - a. Checkbox voor overschrijven DeviceID. Indien 0x0 dan gebruik van controller DeviceID.
 - b. Checkbox voor automatisch 64bit key te genereren.
3. LoRaWAN AppKey. Type : invoer. Formaat : hexadecimaal 32 karakters (128 bit).
 - a. Checkbox voor automatisch 128bit key te genereren.
4. LoRa zendinterval. Type : invoer. Formaat : decimaal 5 t/m 1440 (minuten) ¹⁸.
5. Sensor (6x). Type : invoer.
 - a. keuzelijst sensortype ¹⁸
 - b. Checkbox voor actief / niet actief.
 - c. Decimaal veld voor aantal samples (1-100).
6. Status ValwaysOn. Type checkbox. Actief = aan, inactief = uit.
7. Module info. Type : alleen weergave.
8. Sensor info. Type : alleen weergave
9. Diagnostische gegevens. Type : alleen weergave. (oa batterij status)

Alle hexadecimale velden worden weergegeven met spaties tussen iedere byte weergegeven.
Invoer zonder spaties wordt geaccepteerd.

De gebruiker kan via een menu balk functie de volgende acties uitvoeren:

1. Open MFM configuratie (uit ASCII bestand).

Velden: JoinID, DeviceID, AppKey, Meet interval, Sensor config.
Er wordt een popup scherm getoond met bestandsverkenner waarin een config bestand geselecteerd kan worden. Bij bevestiging van bestand worden de relevante gegevens uit het bestand geladen in de velden op het scherm.
2. Opslaan MFM configuratie (naar ASCII bestand)
Velden: JoinID, DeviceID, AppKey, Meet interval, Sensor config.
Er komt een popup scherm voor met bestandsverkenner voor opslaan van bestanden. Er wordt automatisch een bestandsnaam voorgesteld obv <DeviceID>_<jjjjmmdd>_<hhmmss>.cfg, deze kan

¹⁸ Waardes bepaald door MFM basismodule software. Configuratie programma haalt de sensormodule naam op adhv de ID uit een txt file.

handmatig aangepast worden alvorens het opslaan te bevestigen.
Alle relevante velden worden opgeslagen in het opgegeven bestand.

3. Download MFM configuratie.
Velden: JoinID, DeviceID, AppKey, Meet interval, Sensor config, Module info, Sensor Info, diagnostische gegevens.
Gegevens van de getoonde velden worden opgevraagd via de geselecteerde COM-poort door de applicatie met behulp van de commando's gedefinieerd in H3.8 Seriële poort (via USB).
4. Upload MFM configuratie.
Velden: JoinID, AppKey, Meet interval, Sensor config.
De gegevens van alle relevante velden worden verstuurd via de geselecteerde COM-poort door de applicatie met behulp van de commando's gedefinieerd in H3.8 Seriële poort (via USB).
De verzonden informatie wordt in CSV formaat met timestamp aan een globaal logbestand toegevoegd.
5. Download MFM meetdata.
Er komt een popup scherm voor met bestandsverkenner voor opslaan van bestanden. Er wordt automatisch een bestandsnaam voorgesteld obv <DeviceID>_<jjjjmmdd>_<hhmmss>.csv, deze kan handmatig aangepast worden alvorens het opslaan te bevestigen.
Alle beschikbare meetdata wordt gedownload via de geselecteerde COM-poort door de applicatie met behulp van het commando gedefinieerd in H3.8.10 (Commando Get+DataDump). De ontvangen data wordt direct in het CSV opgeslagen. Nadat download gereed is komt er een popup scherm om het bestand te openen met het standaard programma gekoppeld aan de extensie (Windows functionaliteit).
6. Wis MFM meetgeheugen.
Popup scherm ter bevestiging van actie. Na deze actie is data niet meer terug te halen.

8.3 Broncode

De broncode van het MFM configuratie tool wordt opgeleverd in dmv een git repository.

9 Drivers USB VCP

Voor het gebruik van de USB aansluiting (virtuele COM port) is een driver nodig voor de PC.

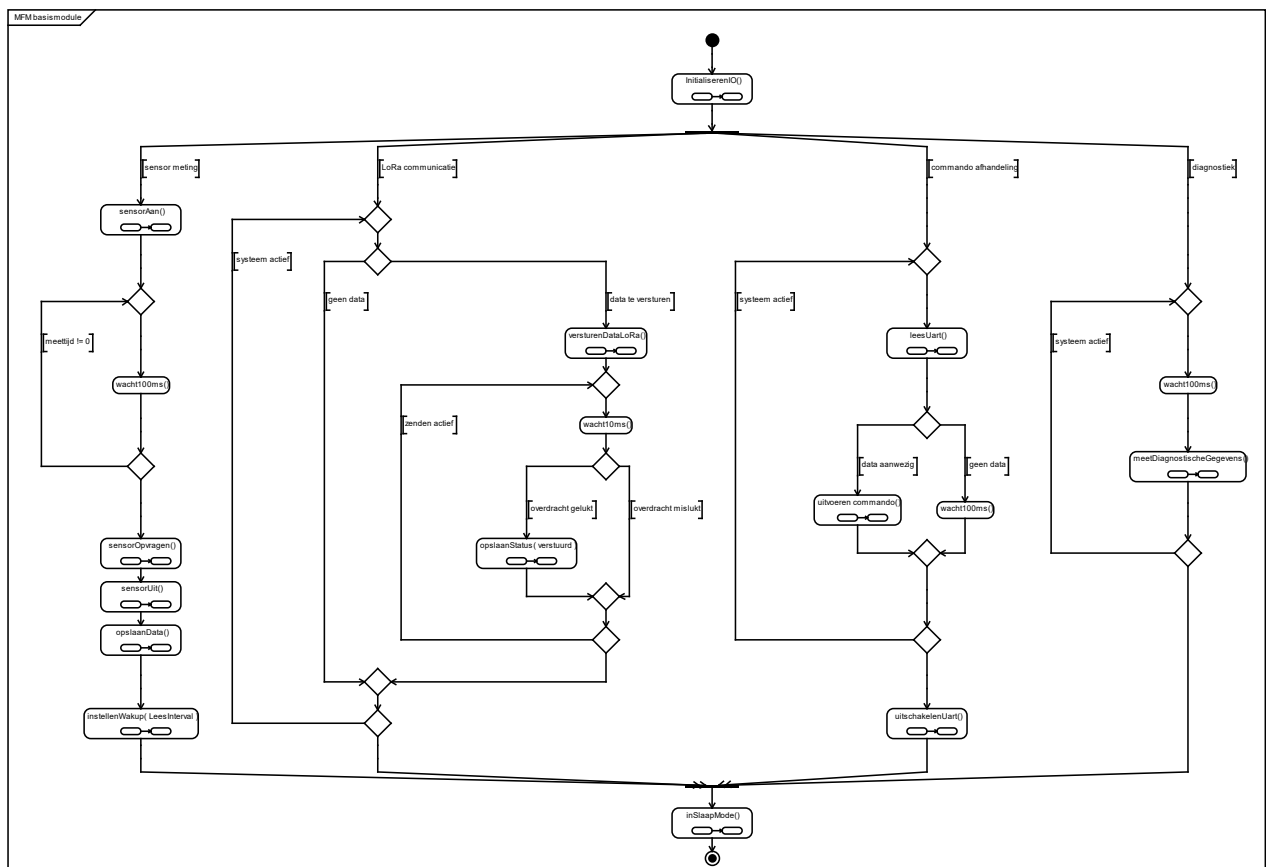
Afhankelijk van het type PC en besturingssysteem zal er een "Virtual COM Port (VCP)"-driver handmatig geïnstalleerd moeten worden.

Deze is te downloaden via de website van de fabrikant van de chip: www.silabs.com.

Over het algemeen gaat dit automatisch.

10 Diagrammen software structuur

10.1 Activiteit diagram MFM basismodule



10.2 Activiteit diagram MFM sensormodule

