

# Sensornetwerk Proof of Concept v0.2

Groep 5

September 2019

# 1 Inleiding

Dit document beschrijft de software implementaties van groep 5 voor het vak “Sensornetwerk Ontwerp” aan de Hogeschool van Amsterdam.

Het doel van de opdracht is om een dynamisch netwerk van sensornodes te maken, waarbij in dit document de nadruk ligt op de algemene netwerkstructuur en de implementatie in MCU van de sensornodes. Hiervoor geldt over het algemeen dat het stuk software wat in dit document beschreven wordt gelijk is voor alle sensornodes. Tijdens de ontwikkeling van de individuele sensorimplementaties (vak “Sensormodule Ontwerp”) kan de code afhankelijk per node worden aangepast.

Het doel van dit document is om een ‘proof of concept’ te geven van de algemene netwerkstructuur waar de sensornodes tijdens het vak “Sensormodule Ontwerp” gebruik van zullen maken.

## 2 Algemene Node Programma

Zoals besproken in de inleiding van dit document het grootste deel van de programma code voor de node MCU’s gelijk. In dit hoofdstuk wordt deze code besproken.

### 2.1 Routing

Het is belangrijk dat het sensornetwerk dynamisch is. Dit wil zeggen dat het netwerk waarover de sensordata verstuurd wordt op ieder willekeurig moment kan veranderen. In de praktijk gebeurt dit wanneer de draadloze sensornodes van plaats veranderen of om welke reden dan ook stopt met communiceren.

Om voortdurend valide informatie te hebben over het netwerk, broadcasten de nodes periodiek de informatie die ze op dat moment hebben over het netwerk. De informatie bestaat uit het aantal hops dat een node denkt dat andere nodes verwijderd zijn. Uit deze informatie kunnen aanliggende nodes opmaken welke nodes te bereiken zijn via de node die net zijn netwerkinformatie broadcast.

Wanneer een node de netwerkinformatie van een buur ontvangt zal deze informatie worden opgeslagen in een tweedimensionaal array. Een voorbeeld situatie is te zien in figuur 1 en de bijbehorende routingtable in tabel 1.

In tabel 1 staat in rij N de routinginformatie die node N bezit, in kolom M staat de afstand tussen node N en node M.

Wanneer node Z (zender) een bericht wil sturen naar node O (Ontvanger), kijkt node Z en kolom O van zijn routingtabel. Node Z checkt in de rij

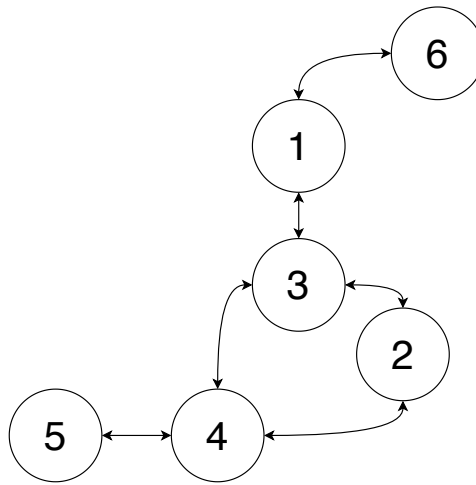


Figure 1: Voorbeeld netwerk.

Routingdata van:	Afstand Node 1:	Afstand Node 2:	Afstand Node 3:	Afstand Node 4:	Afstand Node 5:	Afstand Node 6:
Node 1	0	2	1	2	3	1
Node 2	2	0	1	1	2	3
Node 3	1	1	0	1	2	2
Node 4	2	1	1	0	1	3
Node 5	3	2	2	1	0	4
Node 6	1	3	2	3	4	0

Table 1: Routingtable van de nodes in het netwerk in figuur 1.

van elke directe buur om de laagste hopcount te vinden. Wanneer deze is gevonden word het bericht naar de buur met de laagste hopcount verstuurd die dit proces herhaalt tot het bericht bij de ontvanger is.

## 2.2 Flowchart

Om de algemene node-code inzichtelijk te maken is gebruik gemaakt van een flowchart. In figuur 2 is deze flowchart te zien.

## 2.3 Statemachine

Een groot gedeelte van de tijd is de MCU is aan het wachten op berichten om deze vervolgens te verwerken. Daarom is er voor gekozen om een statemachine te gebruiken. Deze is afgebeeld in figuur 3.

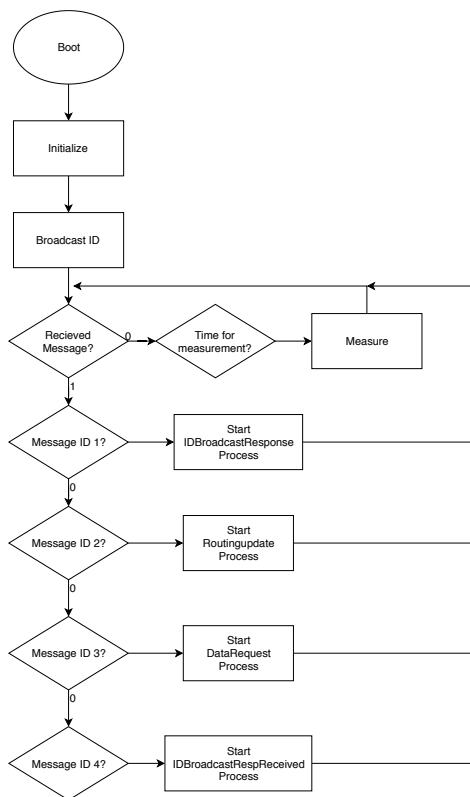


Figure 2: Nadat de node opgestart is (Boot), wordt de MCU geïntialiseerd in het blok [Initialize].

**Figuur dient in deze versie als voorbeeld en zal geüpdatet worden wanneer het basisonwerp af is.**

### 3 Basisstation

Het basisstation heeft als doel het weergeven en opslaan van sensor metingen en het tonen van de luchtkwaliteit. Voor het basisstation wordt er gebruik gemaakt van een Raspberry Pi 3B+ in combinatie met een multi-touch display. Aan het basisstation is, via een USB verbinding, een sensormodule aangesloten om zo het basisstation op het nemen in het netwerk. Om informatie te tonen op het display wordt er gebruik gemaakt van de programmeer library *Ncurses*. Hiermee is een op tekst gebaseerde grafische gebruikers interface (GUI) gemaakt.

#### 3.1 Flowchart

In de flowchart 4 is te zien dat na het opstarten er steeds gekeken wordt of er data binnenkomt. Als er data binnenkomt kan dit vallen in een van de vol-

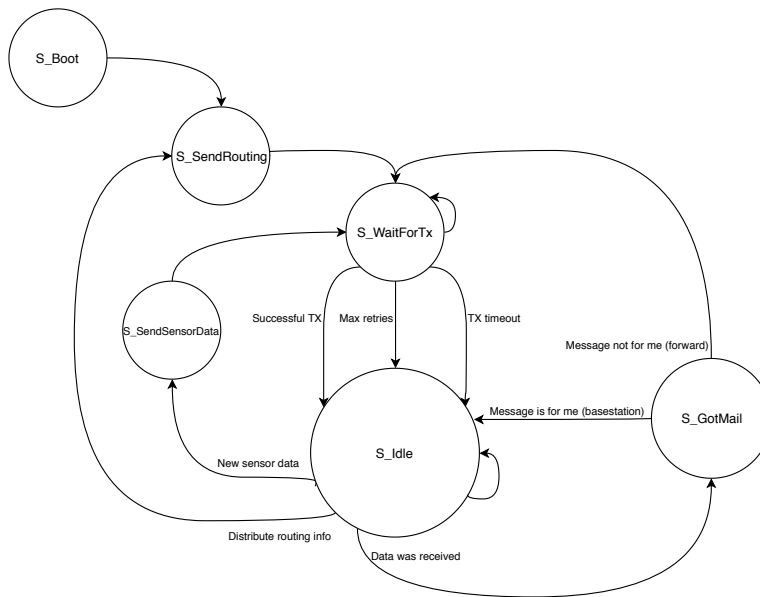


Figure 3: Statemachine die de verschillende states laat zien waar de Xmega in komt vanaf het opstarten.

**Figuur dient in deze versie als voorbeeld en zal geupdate worden wanneer het basis ontwerp af is.**

gende drie categorieën; sensor data, routing table of debug data. Afhankelijk van het type data wordt deze op het scherm getoont en eventueel in een logbestand opgeslagen.

### 3.2 Communicatie met het basisstation

De sensornode kan via de UART, over USB, data naar de Raspberry Pi sturen. De sensor en routing table data wordt binair naar de RPi gestuurd. De debug data kan als ASCII worden verstuurd en wordt direct als tekst op het scherm geprint.

De datapakketen worden in het volgende formaat verstuurd; Byte 0 geeft het type pakket aan; Byte 1 geeft aan hoe groot het pakket is; Vanaf byte 2 begint de data.

Syntax pakket: [Type pakket, Pakketlengte, Data 0, Data 1, ...]

Om de verschillende typen datapakketten van elkaar te onderscheiden wordt met de eerste byte aangegeven welk type datapakket er verstuurd wordt, zie tabel 2. Als een datapakket niet begint met een geldig pakket ID

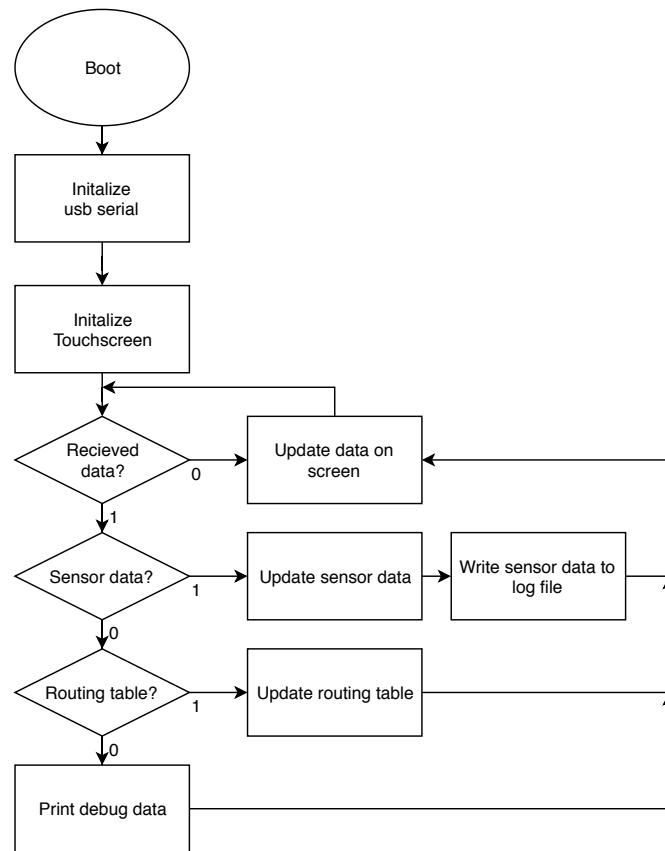


Figure 4: Dataflow basisstation

Pakket ID	Type bericht
0x21	Sensor data
0x63	Routing table

Table 2: Start bytes datapakketten

dan wordt de data aangezien als debug data. Dit is gedaan zodat de debug data ook uit te lezen is met een seriele terminal.

## 4 ISO groep

Om te zorgen dat de nodes van de verschillende groepen elkaars datapakketten kunnen interpreteren en doorsturen zijn er afspraken gemaakt tussen de 5 groepen. Elke groep leverde 1 afgevaardigde, de 5 afgevaardigden vormden de ISO groep. De afspraken die de ISO-groep gemaakt heeft zijn opgesteld

in[1].

Naast afspraken over de algemene NRF-instellingen zijn er ook afspraken gemaakt over de berichttypes en de bijbehorende headers.

Bericht headers staan in de onderstaande tabel.

### Message Types

Mask	Description	Pipe
0x1	ID Broadcast	0
0x2	Routine Routing Table	1
0x3	Receive Port Data	1
0x4	Broadcast Reply	1

## 5 Demonstratie proof of concept

In dit hoofdstuk wordt beschreven hoe de “*proof of concept*” gedemonstreerd wordt. Hierbij zal er onder andere uitleg worden gegeven over de test opstelling, situaties die invloed hebben op het netwerk (zoals het uitvallen van een node) en wat de verwachtingen zijn wanneer dergelijke situaties wel of niet optreden. Vervolgens wordt het netwerk onder deze omstandigheden getest en zal het vastgelegd worden d.m.v. datalogs. Op basis van deze datalogs zal uitgelegd worden of het gedrag van het netwerk overeen kwam met onze verwachtingen.

## 5.1 Testopstelling

In figuur 5 is de testopstelling te zien. Hierbij vertegenwoordigen 5 Xmega's de sensormodules en 1 Xmega het basisstation van het netwerk. De Xmega die het basisstation vertegenwoordigt is verbonden met een Raspberry Pi 3b+ waar een Raspberry 7 inch touchscreen display aan verbonden zit. Op dit display wordt alle ontvangen sensordata per sensormodule getoond en een routingtable waarin valt af te lezen welke nodes er zijn gezien in het netwerk en hoeveel hops deze verwijderd zijn van het basisstation.

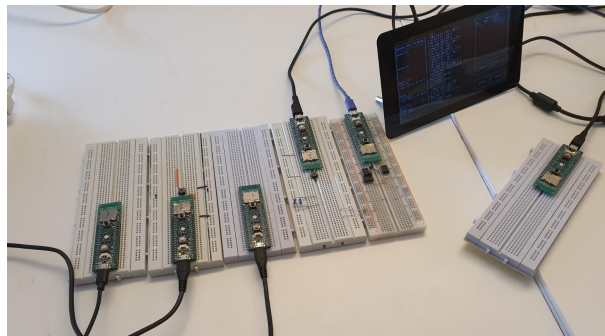


Figure 5: Testopstelling voor de demonstratie van de proof of concept



## 5.2 Situaties die het netwerk veranderen

In tabel 3 is een overzicht te zien van alle situaties die een belangrijke invloed kunnen hebben op een node. Ook zal er in deze tabel worden omschreven wat de verwachte reacties van het netwerk zijn op deze situaties.

Table 3: invloedrijke situaties voor het netwerk en de verwachte reacties

Situatie:	Verwachting:
Sensormodule verplaatst zich richting een nieuwe buur	De verplaatste sensormodule zal aan de burenlust worden toegevoegd van de buur waar hij zich naartoe verplaatst heeft
Sensormodule verplaatst zich van een buur af	De verplaatste sensormodule zal uit de burenlust worden verwijderd van de buur waar hij zich vandaan verplaatst heeft en kan eventueel (afhankelijk van de richting en afstand) geen verbinding meer maken met andere sensormodules
Sensormodule valt opeens uit	De sensormodule zal na een bepaalde tijd verwijderd worden uit de burenlusten en zal in de routingtable bij het basis station als inactief worden gesteld
Sensormodule stuurt onzin data over het netwerk	Aangezien het deze berichten waarschijnlijk niet aan de ISO-standaarden voldoen zullen deze niet worden doorgestuurd of opgenomen door het basisstation
Signalen van een sensormodule worden opeens geblokkeerd	in principe hetzelfde als wanneer een sensormodule uitvalt. Verder zal de sensormodule na een bepaalde tijd alle burenen uit de burenlust verwijderen aangezien het ook geen berichten meer ontvangt
Sensormodule is niet direct verbonden met het basis station maar wel met andere sensormodules	De sensormodule zal proberen via andere sensormodules zijn data bij het basis station terecht te krijgen.

## 6 Demonstratie test

In dit hoofdstuk worden de resultaten van de demonstratie test voor het proof of concept besproken. De demonstratie test is uitgevoerd volgens de testopstelling die wordt besproken in paragraaf 5.1 waarbij de diverse situaties die in paragraaf 5.2 getest worden. in tabel 3 is een overzicht te vinden van alle situaties die getest worden en wat de resultaten zijn die verwacht worden. De situaties zullen op volgorde van de tabel besproken worden.

### 6.1 Sensormodule verplaats richting nieuwe buur

Om deze situatie te testen was er een sensormodule (met ID 52) op een dusdanige afstand van de andere modules geplaatst dat deze geen burenen heeft, vervolgens is de sensormodule met ID 83 richting sensormodule 52 gelopen. De verwachting was dat sensormodule 83 zal worden toegevoegd aan de burenenlijst van sensormodule 52. In figuur 6 is eerst te zien hoe

```
Lost Neighbor: 83
2 52 51 1 77 1
2 77 51 1 52 1
2 52 51 1 77 1
New Neighbor: 83
2 83 51 2 52 1 77 2
2 77 51 1 52 1
2 52 51 1 77 1 83 1
2 77 51 1 52 1 83 2
2 51 52 1 77 1 83 2
2 52 51 1 77 1 83 1
```

Figure 6: Nieuwe buur 83 wordt toegevoegd aan burenenlijst

sensormodule 83 uit de burenenlijst verwijderd wordt omdat sensormodule 52 met een dusdanige afstand is verplaatst dat ze niet meer met elkaar in verbinding staan. Vervolgens verplaatst sensormodule 83 zich richting sensormodule 52, waarna er vervolgens in figuur 6 te zien is hoe deze weer wordt toegevoegd aan de burenenlijst.

### 6.2 Sensormodule verplaats van buur af

Zoals bij de testsituatie van paragraaf 6.1 wordt er bij deze testsituatie wederom gebruik gemaakt van sensormodules 52 en 83. Bij deze situatie zijn zowel sensormodule 52 als 83 beide op een dusdanige afstand van de andere sensormodules, dat ze alleen elkaar zien. Vervolgens verplaatst sensormodule 83 zich van 52 af. De verwachting is dat sensormodule 83 wordt verwijderd uit de burenenlijst van 52.

```

2 52 51 1 77 1 83 1 105 1
2 51 52 1 77 1 83 1 105 1
2 83 51 1 52 1 77 1 105 1
2 105
2 77 51 1 52 1 83 1 105 1
2 52 51 1 77 1 83 1 105 1
2 105
2 51 52 1 77 1 105 1
2 77 51 1 52 1 105 1
2 52 51 1 77 1 83 1 105 1
2 105
2 51 52 1 77 1 83 2 105 1
Lost Neighbor: 83

```

Figure 7: Buur 83 wordt verwijderd uit burenljst

In figuur 7 is te zien hoe sensormodule 83 wordt verwijderd uit de burenljst van sensormodule 52 nadat deze zo ver is verplaatst dat de verbinding verbroken werd.

### 6.3 Sensormodule valt uit

Voor het testen van deze situatie waren alle nodes verbonden met elkaar. De debug log die is te zien in figuur 8 is gegenereerd vanuit het basisstation. Uit het niets zal sensormodule 52 uitvallen omdat de voeding wordt afgesloten. De verwachting is dat het basisstation sensormodule 52 uit zijn burenljst verwijderd. In figuur 8 is te zien hoe het basisstation van alle 4

```

2 83 51 1 52 1 77 1 105 1 ---
Received Data
Data is for me
0x03 83 105 1 1469
2 51 52 1 53 1 77 1 83 1 105 1
2 53 51 1 52 1 77 1 83 1 105 1
2 105 51 1 52 1 53 1 77 1 83 1
2 77 51 1 52 1 53 1 83 1 105 1
2 51 52 1 53 1 77 1 83 1 105 1
2 53 51 1 77 1 83 1 105 1
Lost Neighbor: 52
2 105 51 1 53 1 77 1 83 1
2 77 51 1 53 1 83 1 105 1
2 83 51 1 52 1 53 1 77 1 105 1
2 53 51 1 77 1 105 1
2 105 51 1 52 2 53 1 77 1 83 1
2 77 52 2 53 1 83 1 105 1
2 83 51 1 53 1 77 1 105 1

```

Figure 8: Buur 52 wordt verwijderd uit burenljst basisstation

andere sensormodules (51, 53, 77, 83). Bij alle ontvangen berichten is de 2e byte het ID van de sensormodule, door te kijken van welk ID alle andere berichten komen is vast te stellen dat alle andere 4 sensormodules actief zijn. Uiteindelijk wordt sensormodule 52 verwijderd uit de burenljst van het basisstation, omdat deze inactief is.

### 6.4 Signalen sensormodule geblokkeerd

Het testen van deze situatie was vergelijkbaar met die van paragraaf 6.3, echter i.p.v. dat sensormodule 52 van de voeding werd afgesloten, werd deze in een metalen kistje gestopt, zie figuur 9.



Figure 9: Metalen kistje voor signaal blokkade

d.m.v. dit metalen kistje worden alle signalen van en naar sensormodule 52 geblokkeerd. De verwachting is dat sensormodule 52 wordt verwijderd uit de burenljsten van alle andere sensormodules. Zoals is te zien in figuur 10

```

New Neighbor: 52
2 52 51 1 77 1 83 2 105 1
2 77 51 1 52 1 83 1 105 1
2 105 51 1 52 1 77 1 83 1
2 51 52 1 77 1 83 1 105 1
2 83 51 1 52 1 77 1 105 1
2 77 51 1 52 1 83 1 105 1
2 105 51 1 52 1 77 1 83 1
2 51 52 1 77 1 83 1 105 1
2 83 51 1 52 1 77 1 105 1
2 77 51 1 52 1 105 1
2 105 51 1 77 1 83 1
Lost Neighbor: 52
2 51 77 1 83 1 105 1

2 51 52 1 77 1 83 1 105 1
2 83 51 1 52 1 77 1 105 1
2 77 51 1 52 1 105 1
2 51 77 1 83 1 105 1
Lost Neighbor: 52

2 105 51 1 52 1 77 1 83 1
2 51 77 1 83 1 105 1
2 83 51 1 52 1 77 1 105 1
2 77 51 1 52 1 105 1
2 105 51 1 77 1 83 1
Lost Neighbor: 52
2 51 77 1 83 1 105 1

```

Figure 10: sensormodule 51, 83 en het basisstation verwijderen 52 uit burenljst

wordt sensormodule 52 verwijderd bij de andere sensormodules. Wederom valt hier, net zoals in paragraaf 6.3, te zien hoe de andere sensormodules nog wel berichten ontvangen van elkaar.

## 6.5 Data doorsturen via andere sensormodule

Om deze situatie te testen is er een sensormodule op een bepaalde afstand van de rest van het netwerk gezet waardoor deze geen verbinding meer heeft met het netwerk. Vervolgens is er een andere sensormodule tussen het netwerk en het de “eenzame” sensormodule in geplaatst. De verwachting is dat de “eenzame” sensormodule verbinding maakt met sensormodule die er tussen is geplaatst, en via hem sensordata stuurt naar het basisstation. De “eenzame” sensormodule is 52, degene die er tussen is geplaatst is 83.

```

2 105 51 1 52 1 77 1 83 1
New Neighbor: 52
2 52 51 1 77 1 83 1 105 2
Received Data
Data is for 105
0x03 52 105 1 878
2 83 51 1 52 1 77 1 105 1

```

Figure 11: sensormodule 83 stuurt sensordata van 52 door

In figuur 11 is te zien hoe sensormodule 52 door 83 gezien wordt. Vervolgens ontvangt sensormodule 83 sensordata van 52, die voor het basisstation (105) bedoeld is (zie blauw bericht wat met 0x03 begint). In figuur 12 is

```

2 83 51 1 52 1 77 1 105 1
2 77 51 1 52 1 83 1 105 1
2 51 77 1 105 1
2 105 51 1 52 1 77 1 83 1
2 52 51 1 77 1 83 1 105 2
Received Data
Data is for me
0x03 52 105 1 878
2 83 51 1 52 1 77 1 105 1
2 77 51 1 52 1 83 1 105 1
2 51 52 1 77 1 83 1 105 1
2 105 51 1 52 1 77 1 83 1
2 52 51 1 77 1 83 1 105 1
2 83 51 1 52 1 77 1 105 1
2 105 51 1 52 1 77 1 83 1
2 83 51 1 52 1 77 1 105 1
2 77 51 1 83 1 105 1
2 51 52 1 77 1 83 1 105 1

```

Figure 12: basisstation ontvangt doorgestuurde data via 83

te zien hoe het basisstation de sensordata van sensormodule 52 ontvangt, terwijl het via sensormodule 83 is doorgestuurd. Zie hiervoor het blauwe bericht wat met 0x03 begint. Zoals bij alle andere berichten is de 2e byte het ID waar het bericht origineel vandaan komt, hieraan is te zien dat het bericht van sensormodule 52 komt.

## References

- [1] F. Hes, “Iso standaarden project sensornetwerken 2019-2020,” Oktober 2019.