

Localisatie met Arduino's op basis van vier bakens

Peter van Dijk & Elizabeth Schermerhorn

10 juli 2014

Inhoudsopgave

1	Inleiding	3
2	Probleemstelling	3
3	Gerelateerd werk	3
3.1	Methodologie	4
3.1.1	De hardware	4
3.1.2	De software	4
3.1.3	De instellingen/vastgestelde constanten	6
3.2	Resultaten en analyse	6
3.2.1	Metingen en resultaten	7
3.3	Conclusie	8
4	Betrouwbare end-to-end-communicatie	9
4.1	Inleiding	9
4.2	Probleemstelling	9
4.3	Protocol	9
4.4	Methodologie	10
4.4.1	De software	10
4.4.2	De hardware	10
4.5	Resultaten en analyse	10
4.6	Conclusie	11
A	De radio en de RF24-library	12
A.1	Eigenschappen van de RF24	12
A.2	Energieverbruik	12
A.3	Enhanced ShockBurst	12
A.3.1	Automatische hertransmissie	13
A.4	Adressering	13
A.5	Functies	14
B	Code Sender	14
C	Code Receiver	14
D	Code Hop	14

1 Inleiding

In deze paper zal een localisatie algoritme besproken worden. De localisatie zal gedaan worden op basis van vier bakens die een ultrasoon geluid uitzenden en een arduino met een ultrasoon ontvanger. Het doel van dit experiment is om binnen een 5m bij 5m veld de positie zo nauwkeurig mogelijk proberen te bepalen. Aan de hand van het ultrasone geluid in combinatie met een GPS-bepalings algoritme wordt nauwkeurig bepaald wat de coördinaten zijn.

2 Probleemstelling

De onderzoeksvraag die hier centraal staat: *"Een algoritme opstellen waarmee de exacte GPS-positie te bepalen is."* De hypothese is dat dit gaat lukken. Aangezien het om een statische omgeving gaat waarbinnen alles te controleren is zou dit mogelijk moeten zijn. Door het signaal van de bakens te ontvangen kan de afstand tot de individuele bakens berekend worden. Vervolgens kan met een GPS-algoritme de precieze locatie bepaald worden. Een verwacht obstakel hierbij zal de overflow (Getallen zijn te groot om te representeren) in arduino zijn.

3 Gerelateerd werk

Op dit gebied is er al veel onderzoek verricht. Hierdoor is er veel werk wat gebruikt kan worden bij het onderzoek. Aangezien dit een onderzoek is wat al vele malen is uitgevoerd is er veel gerelateerd werk wat is gebruikt in dit onderzoek. In deze twee papers worden er globale localisatie systemen beschreven en hoe deze gebruik maken van externe factoren. In de volgende twee papers worden er GPS-bepalings algoritmen gepresenteerd waarmee de exacte locatie bepaald kan worden. In de eerste twee papers worden twee manieren toegelicht waarop localisering mogelijk is. In [<http://www.cl.cam.ac.uk/research/dtg/www/publications/public/files/tr.97.10.pdf>] worden twee manieren beschreven, namelijk:

- Active badges
- ParcTab

Active badges is gebaseerd op het periodiek verzenden van informatie naar iedereen. Door -bijvoorbeeld een gebouw - heen zijn er overal sensoren geplaatst die luisteren naar berichten die worden verstuurd door badges. Door vast te stellen welke sensoren het signaal van welke badge hebben ontvangen is het mogelijk een grove schatting te geven van de locatie van de badge.

ParcTab is een PDA die gebruik maakt van infrarood voor het netwerk. Wat betreft de localisatie bepaling lijkt deze heel veel op de badges die hierboven staan beschreven. Er worden berichten verstuurd en afhankelijk van welke sensor

deze ontvangen wordt de locatie bepaald. Een groot verschil is dat de ParcTab is bedoeld voor veel meer en intensiever gebruik dan alleen de localatie berekenen.

In de laatste twee papers, [<http://en.wikipedia.org/wiki/Trilateration>] en [Localization and positioning, CH. 9] worden er twee manieren besproken waarop de GPS positie bepaald kan worden.

In hoofdstuk 9 van Localization and positioning wordt er ook gebruik gemaakt van bakens die een signaal uitzenden. Er moet eerst berekend worden wat de afstand is tot de verschillende bakens. Voor dit algoritme is het van belang dat er minimaal tot drie bakens de locatie bekend is. Wanneer de afstand tot drie bakens bekend is wordt er gebruik gemaakt van matrixberekeningen om de locatie te bepalen. Dit is de implementatie die ook is gebruikt in het experiment wat in deze paper is gemaakt.

Een tweede GPS-positie bepaling die gebruikt zou kunnen worden heet triangulation. Ook hier wordt gebruik gemaakt van de afstand tot de bakens die een signaal verzenden. Dit algoritme lijkt veel op het algoritme uit CH9 maar hier wordt gebruik gemaakt van de berekening van cirkels waardoor er geen matrixberekeningen nodig zijn.

3.1 Methodologie

Om deze hypothesen te toetsen moeten er metingen gedaan worden. Hier zijn verschillende zaken van belang.

- De gebruikte hardware
- De gebruikte software
- De instellingen/vastgestelde constanten
- De onderzoeksopstelling

3.1.1 De hardware

De Hardware die gebruikt is bij de metingen is een Arduino UNO en een Nordic nRF24L01 draadloze transceiver in combinatie met een ultrasoonontvanger.

3.1.2 De software

Om de hardware te gebruiken wordt gebruik gemaakt van de opensourcesoftwarebibliotheek voor de Arduino, die te vinden is op:

<http://maniacbug.github.io/RF24/>

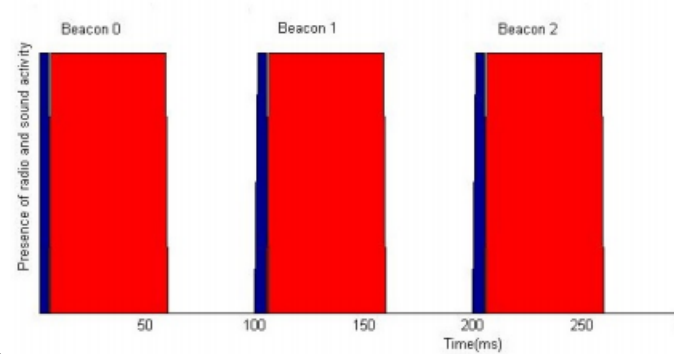
Met deze software kunnen de radio en ontvanger aangestuurd worden en kunnen pakketjes worden verzonden en ontvangen. De radio luistert en verstuurt pakketten over een bepaald kanaal. Het is van belang dat de radio's hetzelfde kanaal gebruiken, zodat ze met elkaar kunnen communiceren. De kanalen hebben een nummer van 0 tot en met 125. De radio heeft de volgende instellingen nodig:

- Kanaal 76
- Automatisch herverzenden Uit
- Transmissiesnelheid: 2 Mbps
- Adres verzendende pipe: 0xdeadbeef1LL
- Payload-grootte 1 byte

De ultrasoon ontvanger moet op een bepaalde manier aangesloten worden zodat

	afkorting	Wijze van aansluiten
deze geluid kan ontvangen:	E	Verbinden met GND van Arduino
	GN	Verbinden met 5V van Arduino
	++	Verbinden met 5V van Arduino
	GND	Verbinden met Ground van Arduino
	S	niet aansluiten

Om de tests te kunnen uitvoeren is er gebruik gemaakt van vier bakens die met



een bepaald patroon informatie verzenden. In grafiek

Om de metingen uit te voeren en onze hypothesen te testen is er gebruik gemaakt van één zender en één ontvanger. De zender gebruikt `sender.ino` en de ontvanger gebruikt `receiver.ino`. Deze programma's zijn te vinden in de appendix. De zender zend pakketjes met daarin het pakketnummer. De ontvanger begint met luisteren en onthoudt het nummer van het eerste pakketje dat hij heeft ontvangen. Vervolgens zal de ontvanger blijven luisteren totdat deze 1000 pakketten heeft ontvangen. Uit de inhoud van het laatste pakket kan worden opgemaakt hoeveel pakketten er verloren zijn gegaan voor de duizend die er ontvangen zijn. De PER wordt berekend met de formule:

$$(([\text{laatstepakketnummer}] - [\text{eerstepakketnummer}]) / [\text{aantalontvangenpakketten}])$$

In de software van de Sender en Receiver is gebruik gemaakt van de volgende drie methoden om de verschillende waarden aan te kunnen passen.

- `setChannel(uint8 i)`
met $0 \leq i \leq 125$

- `setPALevel(PALevel level)`
met `level` $\in \{ \text{RF24_PA_MAX}, \text{RF24_PA_MIN}, \text{RF24_PA_LOW} \}$
- `setDataRate(Rate rate)`
met `rate` $\in \{ \text{RF24_1MBPS}, \text{RF24_2MBPS} \}$

Door met deze drie methoden de waarden aan te passen verkrijgen we de gewilde meetresultaten. Deze zullen in sectie 1.4 besproken worden.

3.1.3 De instellingen/vastgestelde constanten

Om deze metingen bruikbaar te houden moeten er een paar constanten gedefinieerd worden.

- De grootte van de payload
- Het aantal te versturen berichten

Het aantal keer dat er een pakket opnieuw verstuurd wordt is niet van belang in deze implementatie omdat het zo geïmplementeerd is dat het aantal ontvangen berichten wordt bijgehouden en het aantal verzonden berichten wordt bijgehouden. Voor de grootte van de payload is er gekozen voor de grootste mogelijke waarde die deze kan aannemen, namelijk 255 bytes. Deze waarde is gekozen omdat de onderzoeksvraag is wanneer is de radio communicatie het meest betrouwbaar en dit moet getest worden voor reële waarden. Het aantal te versturen berichten is op 1000 gezet. Hier is voor gekozen vanwege het idee omdat meer waarden betekent nauwkeurigere resultaten.

De meetopstelling bestaat uit twee radio's welke 270cm uit elkaar geplaatst zijn. Om de resultaten consistent te houden moet bij elke meting deze afstand aangehouden worden. Verder worden berichten niet opnieuw verzonden als er geen acknowledgement volgt, want dit betekent pakket verlies en dat moet gemeten worden.

3.2 Resultaten en analyse

Hier zullen de resultaten weergegeven en geanalyseerd worden.

De constanten zoals ze standaard gebruikt zullen worden. Als ze gewijzigd worden voor de test dan wordt dit aangegeven.

- afstand tussen radio's: 270 cm
- Geen hertransmissie van pakketten
- payload: 255 bytes
- channel: 125

3.2.1 Metingen en resultaten

Tijdens de metingen waren er geen andere radio's aan het sturen, waardoor de radio's geen last van interferentie hadden.

Instellingen	PER	gemiddelde PER
outputpower: MAX	1/1000	2/3000
transmissiesnelheid: 2MBps	1/1000	
delay: 25 ms	0/1000	

Instellingen	PER	gemiddelde PER
outputpower: MAX	0/1000	0/3000
transmissiesnelheid: 2MBps	0/1000	
delay: 50 ms	0/1000	

Instellingen	PER	gemiddelde PER
outputpower: MAX	0/1000	0/3000
transmissiesnelheid: 1MBps	0/1000	
delay: 50 ms	0/1000	

Instellingen	PER	gemiddelde PER
outputpower: MIN	85/1000	1273/3000
transmissiesnelheid: 2MBps	669/1000	
delay: 50 ms	519/1000	

Instellingen	PER	gemiddelde PER
outputpower: LOW	0/1000	8/3000
transmissiesnelheid: 2MBps	2/1000	
delay: 50 ms	6/1000	

Instellingen	PER	gemiddelde PER
outputpower: MAX	0/1000	0/3000
transmissiesnelheid: 2MBps	0/1000	
delay: 50 ms	0/1000	
channel: 0		

Instellingen	PER	gemiddelde PER
outputpower: MAX	0/1000	1/3000
transmissiesnelheid: 2MBps	1/1000	
delay: 50 ms	0/1000	
channel: 64		

Verder zijn er nog enkele kleinere tests uitgevoerd op een netwerk met interferentie. Dit leverde de volgende resultaten:

Instellingen	PER	gemiddelde PER
outputpower: MAX	47/300	433/3000
transmissiesnelheid: 2MBps	77/300	
delay: 50 ms	9/300	

Om het resultaat nauwkeuriger en accurater te kunnen weergeven is ervoor gekozen om alle tests in drievoud uit te voeren zodat vreemde gebeurtenissen opgemerkt worden. Een voorbeeld is een error rate > 1 . Tussen deze resultaten is de instelling HIGH voor outputpower niet weergegeven, aangezien deze voor de gebruikte versie van de radio niet een mogelijke waarde was.

3.3 Conclusie

Als er naar de gevonden waarden gekeken wordt, is de PER onacceptabel wanneer de outputpower op MIN staat. Een gemiddelde PER van 1273/3000 pakketten zorgt op den duur voor veel hertransmissies dus veel vertraging op het netwerk. Dit is een situatie die niet gewenst is. Wanneer naar de overige instellingen gekeken wordt en de daarbij behorende gemiddelde PER dan zijn ze perfect of een $PER < 0.01$. Hier moet bij vermeld worden dat op het moment dat er meer interferentie is op het netwerk de PER omhoog schiet op het netwerk. Op dat moment is het kiezen van een ongebruikt frequentiekanaal van belang. Dit experiment heeft plaats gevonden in een interferentievrije ruimte.

De onderzoeksvraag is: *"Voor welke waarden voor de outputpower, transmissiesnelheid en frequentiekanaal is de PER het laagst."* Het antwoord dat hierbij hoort is dat voor elke waarde van de transmissiesnelheid en het frequentiekanaal de PER niet boven de 0.01 uit komt. Dit geldt echter niet voor de outputpower; op het moment dat de outputpower op minimaal gezet wordt bedraagt de gemiddelde PER 1273/3000. Hieruit kan geconcludeerd worden dat de outputpower tenminste op laag of maximaal gezet moet worden om de PER tot een minimum te beperken.

4 Betrouwbare end-to-end-communicatie

4.1 Inleiding

In het vorige experiment is de betrouwbaarheid van een singlehop-netwerk onderzocht. Nu kan de betrouwbaarheid van een multihop-netwerk getoetst worden en kunnen de waarden vergeleken worden om te bepalen welk netwerk betrouwbaarder is.

4.2 Probleemstelling

De probleemstelling waarvan uit zal worden gegaan is *"Is het mogelijk een betrouwbaar end-to-end multihop-connectie op te zetten?"*. De hypothese is dat dit mogelijk is in combinatie met :

- MIN zal het slechtst presteren en MAX het beste(transmissie sterkten)
- alternating bit protocol(ABP) Met deze drie aspecten verwerkt in de implementatie zou het multihop-netwerk moeten werken.

4.3 Protocol

Het protocol wat veelal gebruikt wordt in deze implementatie is het ABP. Dit protocol houdt twee zaken in:

- hertransmissies sturen
- kijken of er iets binnenkomt

Het ABP zorgt ervoor dat wanneer er geen acknowledgement ontvangen wordt er een hertransmissie gestuurd wordt. Door een bit op 0 of 1 te zetten kan er gekeken worden of er nieuwe data ontvangen wordt of niet. Dit is de globale werking van het protocol. Dit protocol is gebruikt in de implementatie van het multihop-netwerk, veelal op dezelfde manier maar dit zal toegelicht worden.

In de implementatie komt dit protocol terug in alle nodes. Alle nodes moeten weten of er berichten binnen komen. Om dit te registreren is er gekozen voor het ABP.

De zender verstuurt de berichten en wacht totdat er een acknowledgement terugkomt voordat een nieuw bericht wordt verstuurd. Op het moment dat er een time-out plaatsvindt zal er een hertransmissie plaatsvinden. Op het moment dat er een bericht binnenkomt dan zal dit geregistreerd worden en zal dit bericht verder worden afgehandeld.

Figuur 1: Opstelling van het netwerk

4.4 Methodologie

Om de onderzoeksvraag te kunnen beantwoorden is er een meetopstelling nodig. De meetopstelling is weergegeven in Figuur 1 met een Sendernode, een Hopnode en een Receivernode. De namen geven het al aan, de Sendernode stuurt de berichten (en wacht op bevestiging), de Hopnode stuurt de berichten door en de Receivernode ontvangt de berichten (en bevestigt de ontvangst).

4.4.1 De software

De software die gebruikt is voor dit experiment is grotendeels hetzelfde als het voorbeeld pingpair dat wordt meegeleverd bij de RF24 library voor de Arduino. In deze implementatie zijn slechts de adressen van de pipes gewijzigd naar de vier die te zien zijn in Figuur 1 en is de pakketgrootte veranderd naar 255 bytes.

De Hopnode luistert naar de Sendernode en Receivernode en stuurt berichten van de Sendernode door naar de Receivernode en andersom. In de implementatie is ervoor gezorgd dat er hertransmissies gestuurd kunnen worden als er een timeout optreedt. Er is gebruik gemaakt van een aantal constanten, namelijk:

- payload = 255 bytes
- hertransmissies : 15
- delay: 50ms
- transmissiesnelheid: 2MBps

Deze constanten zullen tijdens het testen niet worden veranderd.

4.4.2 De hardware

Deze bestaat uit drie identieke nodes. Deze nodes zijn allemaal voorzien van een radio waarmee ze de berichten kunnen ontvangen en versturen. De nodes zijn in een lijn geplaatst met een meter afstand tussen Sender en Hop en tussen Hop en Receiver. De experimenten zijn uitgevoerd in een netwerk waar nog meer radios aan het sturen zijn.

4.5 Resultaten en analyse

Om de resultaten te verkrijgen die gewenst zijn worden er telkens andere waarden gebruikt voor de transmissie sterkte. De transmissiesterkte geldt voor alle drie de nodes(zender, hop en ontvanger). De drie waarden die hiervoor gebruikt zijn:

- RF24_PA_MIN

- RF24_PA_LOW
- RF24_PA_MAX

De waarden die gevonden zijn bij de volgende instellingen volgen hieronder.

Instellingen	PER	gemiddelde PER
outputpower: MIN	1/1000 17/1000 7/1000	25/3000

Instellingen	PER	gemiddelde PER
outputpower: LOW	0/1000 1/1000 1/1000	2/3000

Instellingen	PER	gemiddelde PER
outputpower: MAX	0/1000 0/1000 1/1000	1/3000

In de bovenstaande drie grafieken staan de PER samen met de transmissie sterkte waarmee de pakketjes verstuurd zijn. Er moet rekening gehouden worden met de omgeving waarin deze experimenten zijn uitgevoerd.

4.6 Conclusie

De hypothese die gesteld werd: *"Is het mogelijk een betrouwbaar end-to-end multihop-connectie op te zetten?"*. Deze hypothese wordt nu gestaafd door de resultaten die verkregen zijn door de experimenten uit te voeren met veranderende transmissie sterktes. Uit de grafieken bij 2.5 kan opgemaakt worden dat een multi-hop verbinding, welke op deze manier geïmplementeerd is, een redelijke implementatie geeft met een lage error rate. Zoals verwacht in de hypothese is zenden op minimum de slechtst scorende, maar zolang je hertransmissies hebt is dit een degelijke implementatie. De hertransmissies komen voort uit het geïmplementeerde alternating bit protocol. Door het ABP en de beste transmissie sterkte te kiezen is er een stabiel netwerk opgezet wat in het dagelijks leven toegepast kan worden.

A De radio en de RF24-library

A.1 Eigenschappen van de RF24

In dit onderzoek wordt gebruik gemaakt van de rf24 radio. Deze radio heeft de volgende eigenschappen:

- **Frequentieband:** 2.4000-2.4835 GHz
- **Datasnelheid:** 1 of 2 Mb/s
- **Aantal kanalen:** 126 RF-kanalen
- **Modulatietechniek:** Gaussian Frequency Shift Key(GFSK)

A.2 Energieverbruik

De radio het volgende energieverbruik in verschillende modi:

Energiemodus	Energieverbruik in Ampère
Standby-I	22 μ A
Standby-II	320 μ A
Power down	900 nA

Zendmodus	Energieverbruik in Ampère
0 dBm	11.3 mA
-6 dBm	9 mA
-12 dBm	7.5 mA
-8 dBm	7 mA

Ontvangstmodus	Energieverbruik in Ampère
2 Mbps	12.3 mA
1Mbps	11.8 mA

A.3 Enhanced ShockBurst

Een belangrijk aspect van de RF24 radio is Enhanced ShockBurst. Dit is een pakketgerichte datalinklayer protocol. Belangrijke functies van Enhanced ShockBurst zijn:

- Automatische pakket samenvoeging
- 6 data pipe multiceivers
- pakket afhandeling

Het formaat van het datapakket bestaat uit verschillende delen, het adres, pakket identificatie, geen acknowledgement bits en CRC. De velden hebben de volgende functies:

Data veld	Functie
Adres	Het adres van de ontvanger
Pakket identificatie	Om te achterhalen of het pakket een hertransmissie is.
geen acknowledgement bit	Als deze bit gezet is mag het pakket niet automatisch erkend worden.
CRC	Error detectie mechanisme voor het pakket

A.3.1 Automatische hertransmissie

Enhanced Shockburst heeft als eigenschap automatisch pakketvalidatie. In ontvangstmodus zoekt de RF24 constant naar valide adressen welke gegeven zijn in RX_ADDR registers. Op het moment dat een valide adres gedetecteerd wordt, zal Enhanced Shockburst het pakket gelijk valideren. Enhanced Shockburst is voorzien van een ART(Automatic Retransmission) functie. Door deze functie is Enhanced Shockburst in staat een pakket uit zichzelf opnieuw te versturen als er geen acknowledgement wordt ontvangen. Er zijn drie redenen waarom de ART functie stopt met het luisteren na het versturen van een pakket. De ART stopt met luisteren na:

- Auto retransmit delay(ARD) is verlopen
- Geen gematcht adres gevonden binnen $250\mu s$
- Nadat er een pakket ontvangen is

De radio is niet in staat om te broadcasten maar om deze functionaliteit te gebruiken moet er gebruik gemaakt worden van een multiceiver.

A.4 Adressering

De adressen van een multiceiver moeten altijd sterk op elkaar lijken, alleen de least significant byte moet anders zijn. De volgende 4 adressen volgen uit dit principe:

- F0F0A
- F0F0B
- F0F0C
- F0F0D

Doordat de Least significant byte anders is zijn dit vier geldige adressen.

A.5 Functies

Met de volgende functies worden waarden van de RF24 radio ingesteld:

Functie	Methode
Kanaal instellen	<code>void RF24::setChannel (uint8_t channel)</code>
Zendvermogen	<code>void setPALevel (rf24_pa_dbm_e level)</code>
Aantal pogingen van ART instellen	<code>void RF24::setRetries (uint8_t delay, uint8_t count)</code>
Adres van de zender instellen	<code>void openWritingPipe (uint64_t address)</code>
Adres van de ontvanger instellen	<code>void openReadingPipe (uint8_t number, uint64_t address)</code>

B Code Sender

C Code Receiver

D Code Hop