Een Simpele RF Data Link

Tekst en uitleg voor een succesvolle draadloze verbinding

Samenvatting

Data of informatie versturen over een draadloze verbinding is slechts zeer beperkt te vergelijken met het sturen van signalen door bijvoorbeeld een koperkabel of een glasvezel. Dat komt in hoofdzaak door het verschil in eigenschappen van de fysieke laag.

Het draadloos transporteren van data in een ruwe vorm (‘1’-en en ‘0’-en) is vrijwel onmogelijk. Om een draadloze verbinding zo optimaal mogelijk te kunnen laten werken zult u met een veelvoud aan eigenschappen rekening moeten houden. Dit document beschrijft het gedrag van de EM fysieke laag en behandeld een aantal verbeteringen voor de draadloze verbinding die in de microcontroller kunnen worden opgelost. Hiermee kunnen teleurstellende resultaten worden voorkomen.

De in dit document voorgestelde gegevens zijn slechts een voorbeeld. Het dient ter informatie waarop een werkend systeem kan worden ontworpen.

Eigenschappen

Vrijwel alle draadloze signalen worden opgewekt door gebruik te maken van elektromagnetische straling. Radiosignalen maar ook licht zijn vormen van EM straling. Dit type fysieke laag gedraagt zich anders dan een koper- of glasvezelkabel.

Een signaalkabel bestaat uit meestal 2 parallelle geleiders die van elkaar zijn gescheiden door de isolatie rond de geleiders. Door een circuit te maken en stroom door de geleiders te sturen, kunnen signalen worden verzonden. De kabel kan worden afgeschermd en dit geheel vormt zo een bijna perfecte verbinding. Er zijn vrij weinig invloeden van buiten die het signaal door de kabel kunnen beïnvloeden en er is bijna een 100% garantie te geven dat signalen kunnen worden getransporteerd. Een koperkabel heeft dus een zeer hoge beschikbaarheid. Bij een glasvezelkabel is die beschikbaarheid nog veel hoger.

Bij een draadloze verbinding is er geen fysieke verbinding tussen de zender en de ontvanger en we kunnen dus ook geen stroom rondsturen. Het medium bestaat uit de ons omgevende lucht, maar ook in vacuüm kan EM straling worden verstuurd. De enige selectiviteit is de frequentie waarop het kanaal wordt gevormd. Externe bronnen zorgen voor storingen en het medium is zeer slecht in het transporteren van de EM straling. Afhankelijk van de frequentie is de demping soms zo hoog dat een verbinding helemaal niet tot stand kan komen. Het signaal wordt onderweg tussen de zender en de ontvanger teveel verzwakt. De afstand is te groot of het uitgezonden signaal is te zwak.

Maar daar blijft het niet bij. Het kanaal wordt ook door andere apparaten gebruikt. Een veelgebruikte frequentie is 433.92 MHz. Dat is een licentievrij verzamelkanaal waar bijvoorbeeld veel deuropeners voor auto’s op werken. Elke keer dat uzelf of uw buurman de deuren van de auto opent, wordt op die frequentie een signaal verstuurd. En er zijn nog meer storende factoren. Veel draadloze audioapparatuur werkt ook in die frequentieband en dit kan dus een vrijwel constante storingsbron vormen. Een beschikbaarheid van 95% op deze frequentie is zeer goed te noemen. Meestal heeft een draadloze verbinding op een verzamelkanaal een nog lagere beschikbaarheid.

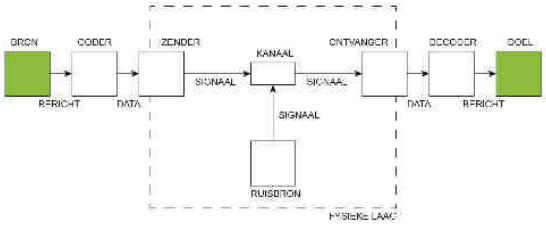
Helaas kan er aan deze vorm van storingen niet zoveel gedaan worden. Het medium is immers niet of nauwelijks te beschermen. Maar dit is dan ook gelijk het grootste voordeel. Het is niet nodig om een kabel door het huis te trekken of een kabel in te graven.

Om een betrouwbare draadloze verbinding te maken moeten we de eigenschappen van het medium en onze apparatuur kennen. De grootste valkuil is wel de opgave van de fabrikant over het bereik van de zender/ ontvanger. Ga er vanuit dat deze opgave slechts onder ideale omstandigheden haalbaar is.

Maar hoe dan wel?

Na dit ietwat negatieve relaas is de tijd voor een positievere benadering en het geven van wat oplossingen. Er zijn gelukkig genoeg mogelijkheden om een betrouwbare verbinding tot stand te brengen en om de onverwachte storingen te verwerken.

Als voorbeeld gebruiken we een systeem dat tot doel heeft om 4 bytes draadloos te transporteren in een simplex transmissiesysteem. We gaan er in het voorbeeld vanuit dat er voldoende hoogfrequent signaal door de ontvanger kan worden verwerkt voor een redelijk betrouwbare verbinding (~95%).



Figuur 1; Simplex transmissiesysteem

In figuur 1 wordt een algemeen simplex transmissiesysteem voorgesteld. De informatie wordt verstuurd van links naar rechts.

Omwille van de eenvoud worden in dit document het opwekken van de EM straling, de modulatie, het ontvangen van de EM straling en de demodulatie (reconstructie) niet behandeld. We gaan er vanuit dat bij een gevormd kanaal (= werkende zender en ontvanger op dezelfde frequentie) er informatie getransporteerd kan worden. Zie voor meer informatie over modulatie en het opwekken van EM straling de uitgaven in de bibliografie.

In het kort komt het hierop neer: De zender vervormt de informatie tot een wisselspanning met een zeer hoge frequentie en met bepaalde eigenschappen (modulatie) waardoor de informatie in de ontvanger eenvoudig kan worden gereconstrueerd. Bij een werkende verbinding zal deze wisselspanning door de antenne van de zender worden uitgestraald in de vorm van EM straling. Een klein gedeelte van deze straling (energie) wordt door de antenne bij de ontvanger opgevangen en omgezet in een wisselspanning. De ontvanger is in staat om hieruit de informatie te reconstrueren.

Bij het draadloze transport van informatie is het van belang te weten dat de ontvanger niet kan differentiëren tussen een gewenst signaal en een ongewenst signaal. De fysieke laag is immers onbeschermd. We moeten zelf een identificatie aanbrengen waardoor de ontvanger kan differentiëren tussen een gewenst en een ongewenst signaal. Hiervoor moeten we enkele eigenschappen toevoegen aan de fysieke laag. Door het toevoegen van wat slimme eigenschappen kan de verbinding beter worden benut.

Selectiviteit vergroten

Eerder is al opgemerkt dat de enige selectiviteit de frequentie van het hoogfrequente signaal is. Maar daar hebben we niet veel aan als iedereen het kanaal kan gebruiken.

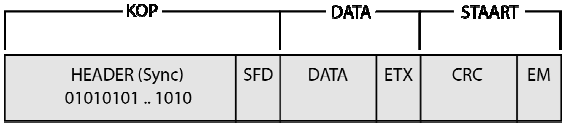
Een andere mogelijkheid voor het vermijden van storingen op de verbinding is het verspreiden van de informatie in de tijd. De zender is niet altijd ingeschakeld. Sterker nog, de voorwaarde voor het gebruik van dit kanaal (433.92 MHz) is dat de zender zo kort mogelijk ingeschakeld mag zijn. De zender wordt dus alleen ingeschakeld zolang als het duurt om de data te verzenden. En dat doen alle andere (stoor)zenders dus ook.

Maar het geheel vermijden van storingen met een simplex verbinding (1 zender en 1 ontvanger) is niet eenvoudig. Om gebruik te maken van *Collision Avoidance* moet de zenderkant in staat zijn het kanaal te beoordelen. Hiervoor is weer een ontvanger nodig. Met de informatie uit de ontvanger kan het proces wachten met zenden tot het kanaal vrij is.

Maar het is wel altijd mogelijk om informatie te verzenden. We doen dit door gewoon direct te starten met zenden, ongeacht de situatie op het kanaal. We noemen dit de *Brute Force* methode. Door de data niet eenmaal, maar tweemaal of zelfs driemaal te verzenden is het mogelijk om een zeer hoge beschikbaarheid te krijgen. Als de informatie de eerste keer niet aankomt, dan de tweede of derde keer wel. Hiermee accepteren we dat informatie mogelijk niet aankomt en dat we storingen veroorzaken voor andere gebruikers van het kanaal. Het is aan de gebruiker, u dus, om zeker te gaan dat de informatie is aangekomen. Door de verbinding zo kort mogelijk te laten bestaan is de kans het grootst dat de informatie in één geheel wordt ontvangen en dat er zo min mogelijk storingen worden veroorzaakt.

Het Packet

De volgende stap in het verhogen van de selectiviteit is het toevoegen van informatie waarmee de ontvangende kant kan differentiëren dat de informatie alleen afkomstig kan zijn van onze zender. We doen dit door de informatie te verpakken in een *Frame* of *Packet* met unieke eigenschappen.



Figuur 2, Opbouw van het packet

Een packet bestaat uit een kop (header), de data en een staart (footer). De kop van het packet bestaat uit informatie die toegevoegd wordt om de UART in de microcontroller te laten synchroniseren. De kop is van essentieel belang voor de werking van de draadloze verbinding. Zonder deze kop zou de data niet foutloos kunnen worden overgebracht. De UART kan niet goed synchroniseren op de data alleen. De eerste bytes worden vrijwel zeker gemist, en dat is lastig als u er maar 4 bytes wilt verzenden. Door een kop van minimaal 8 bytes aan de data toe te voegen kunt u dit probleem voorkomen.

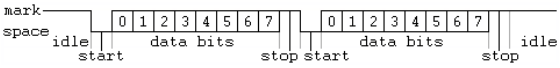
De lengte van de kop is afhankelijk van de tijd die nodig is voor de UART om te synchroniseren. Het sturen van data met een data rate van 2,4 kbit/s kan een kortere kop hebben dan met een data rate van 9,6 kbit/s. In theorie moet de kop voor een data rate van 9,6 kbit/s 4x zo lang zijn als de kop bij een data rate van 2,4 kbit/s. In de praktijk komt het erop neer dat u zelf bepaald hoe lang de kop moet worden. U kiest de juiste lengte van de kop door te beoordelen met welke koplengte de verbinding nog net werkt, en u gebruikt het dubbele aantal bytes. Dus als het met 6 bytes nog werkt, dan gebruikt u 12 bytes voor de kop.

De kop wordt afgesloten met scheidingsteken waarna de data volgt. De data wordt ook afgesloten met een scheidingsteken. Dit scheidingsteken kan vervallen als de data altijd een vaste lengte heeft. De staart bevat een afsluitende reeks bits en eventueel een code; het resultaat van de berekening waarmee de data versleuteld is.

Door aan alle informatie van het packet unieke eigenschappen te geven, is het vrijwel uitgesloten dat een ander signaal door de ontvangende microcontroller wordt ‘herkend’ als een gewenst signaal.

De UART

Het packet wordt verstuurd door serie bytes (8 bits) te verzenden door middel van een UART. Een Universal Asynchronous Receiver/Transmitter stuurt de 8 bits van elke byte, stuk voor stuk naar buiten met een vast ritme. Aan elke groep van 8 bits wordt een start- en een (of twee) stopbit(s) toegevoegd:



Figuur 3, Uitgangssignaal van de UART

Door de keuze van de bytes (karakters) kan het packet worden gevormd. We gebruiken hiervoor de ASCII (American Standard Code for Information Interexchange) code.

Een packet kan er zo uitzien:

SYN-SYN-SYN-SYN-SYN-SYN-SYN-SYN-STX-DATA1-DATA2-DATA3-DATA4-ETX-EM

Wat opvalt, is om 4 bytes (DATA1 t/m DATA4) te transporteren, er maar liefst 15 bytes (150 bits) moeten worden verzonden!

De zender moet ongeveer 50 ms voor de start van de eerste byte worden ingeschakeld. Deze tijd is nodig voor de zender om te stabiliseren op het juiste uitgangsvermogen en op de juiste frequentie. Deze tijd is ook nodig voor de ontvanger om te reageren en om het squelch circuit te laten schakelen. Het vormt een ernstig probleem als deze tijd niet wordt aangehouden. In 50 ms kunnen met een data rate van 9,6 kbit/s maar liefst 480 bits verzonden worden. In die tijd past ons packet dus wel 3x!

Het is ook raadzaam, maar niet verplicht, om de zender nog 5 ms ingeschakeld te laten na het versturen van de laatste byte. Vaak is het in de software niet mogelijk om precies aan te geven wanneer het laatste bit verstuurd is. In dat geval kunt u de software na de laatste byte nog 5 ms laten wachten voordat de zender wordt uitgeschakeld. Op deze manier bent u er zeker van dat de byte geheel wordt verzonden terwijl de zender nog ingeschakeld is.

Het versturen van 4 bytes aan ruwe data duurt dus ongeveer 71 ms met een data rate van 9,6 kbit/s (50 + (150 x 9600^-1) + 5). Met een data rate van 2,4 kbit/s loopt dat op naar ongeveer 118 ms. En dat is kort genoeg. Immers, hoe korter het packet, hoe groter de kans dat het aankomt.

Extra bescherming

Om de ruwe data nog beter te beschermen tegen incidentele fouten kunnen we overtolligheid toevoegen. Net als het herhaald verzenden van het hele packet is het ook mogelijk om alleen de data te herhalen. Hierbij gaan we er vanuit dat de stoorsignalen zeer kort zullen zijn. De kop en de staart van het packet blijven gelijk. Merk op dat hierdoor het totale packet langer wordt en dus ook de verzendtijd van het packet.

Een andere vorm van overtolligheid komt in de vorm van een versleuteling waarmee bitfouten kunnen worden gedetecteerd en gecorrigeerd. Door de simplex verbinding is alleen detectie van fouten niet zo zinvol. Er is immers geen mogelijkheid aan de zender te vertellen dat het packet opnieuw moet worden verzonden.

Een goede maar zeer complexe methode om data te beschermen is door het toepassen van een fout detectie en correctie mechanisme. Denk hierbij aan een FEC of BCH code. Deze vorm van bescherming versleuteld de data en voegt extra bits toe aan het packet.

Het is nog maar de vraag of de moeite om deze methodes in de software te implementeren opweegt tegen de kans dat er een fout optreed of dat de apparatuur wordt misbruikt door derden. In een eenvoudig Domotica systeem waarbij u een lamp wilt schakelen is het wel iets teveel van het goede. In dat geval kunt u kiezen voor een eenvoudigere beschermingsmethode.

Software

Het is aan uw vaardigheid om er iets moois van te maken door het schrijven van de juiste software. Om het systeem zo universeel mogelijk te laten is er een zeer beperkte interface beschikbaar en vrijwel elke microcontroller kan worden gebruikt. Het is daarom onmogelijk om voorbeelden van de code te geven.

Wat we wel kunnen voorstellen is een globale manier van programmeren. We gebruiken de programmering voor de ontvanger als voorbeeld.

De eerste stap in de goede richting is de keuze van de microcontroller. Kies een controller met een hardware UART. De UART moet worden geactiveerd, ingesteld op de juiste BAUDRATE en samenstelling. Deze instelling moet voor beide kanten, zender en ontvanger, gelijk zijn.

De software werkt voor u het gemakkelijkst op interrupts. De UART geeft na ontvangst van een byte een interrupt af. De ISR plaats de byte op een vrije locatie in een lijst en verhoogt een pointer. Als er 5x geen byte is ontvangen (time out), dan zet de ISR een vlag om aan de hoofdlus aan te geven dat een packet is ontvangen. De hoofdlus kan daarna de data decoderen en verwerken.

Maar het kan ook zonder interrupts. De UART plaatst de ontvangen byte op een locatie en zet een ‘byte ontvangen’ vlag op een vaste locatie. De hoofdlus kan zo vrij eenvoudig beoordelen of er een nieuwe byte is ontvangen. De polfrequentie van de hoofdlus moet sneller zijn dan de ingestelde BAUDRATE.

De hoofdlus kan daarna starten met het beoordelen van de serie ontvangen bytes. Door aan de kop van het packet een eigen unieke code te geven, kan de hoofdlus vrij eenvoudig beoordelen of het bericht voor deze ontvanger bestemd is.

Het is niet gebruikelijk om adressering (bij meerdere ontvangende apparaten) in de kop op te lossen. Gebruik hiervoor (een) extra databyte(s).

Vrijwaring

Ideetron kan op geen enkele wijze verantwoordelijk worden gehouden voor storingen en/of schade als gevolg van de gepresenteerde informatie in dit document.

Bibliografie

Een Simpele RF Ontvanger, Data sheet, te downloaden van www.ideetron.nl

Een Simpele RF Zender, Data sheet, te downloaden van www.ideetron.nl

Hoogfrequent Zendontvanger Selectiegids, Data sheet, te downloaden van www.ideetron.nl

Draadloze Communicatietechnologie, Bart Hiddink, 2007, Segment Uitgeverij, ISBN: 9053812032

Computer Networks, Andrew S. Tannebaum, Prentice Hall PTR, 2002, ISBN: 0130661023

Auteur:

Ideetron b.v.
Tel: +31 (0) 343 769 094
e-mail: info@ideetron.nl
www.ideetron.nl