

Test report WiWire

Gemaakte keuzes

Door dat dit project op minstens twee due's moet draaien, tijdsgebonden is en het functioneren van de ontvangers niet deterministisch is ga ik dit project niet testen met Catch2 testcases. In plaats daarvan zijn er meerdere methode van testen zijn geleverd in de github (<https://github.com/ottovis/WiWire>) in de map Test. Hier staan verschillende main_*.cpp bestanden in die hernoemd, gebouwd en geflashed kunnen worden om zekere functionaliteiten van de library te testen

Testcase 1

Door middel van main_send.cpp en main_recieve.cpp. Deze twee bestanden zijn twee simpele programma's die respectievelijk in de zender due en in de ontvanger due geladen kunnen worden om betrouwbaarheid van overdracht te testen. De ontvanger ziet op de seriële poort het bericht "hello world" en de lengte van het bericht en de zender ziet op de seriële poort het aantal pogingen dat nodig was om de informatie over te brengen. De timing van het versturen van dit Hello World bericht is zo ingesteld dat de ontvanger compleet terug is naar zijn hoogste versterkings niveau en dus pure ruis produceert. Dit heb ik zo gedaan om de verificatie en detectie van data te testen.

Testcase 2

Met de programma's main_broadcast.cpp en main_recieve.cpp is het mogelijk om de broadcast functionaliteit van de library te testen. Deze functioneert grotendeels hetzelfde als testcase 1, echter zonder de targeting en acknowledge functionaliteiten van testcase 1. Het is mogelijk om meerdere due's met main_recieve.cpp te laden, deze zouden allemaal hetzelfde Hello World bericht moeten sturen.

Testcase 3

Met het main_sendfail.cpp programma kan je testen of het herhaaldelijk sturen en falen na x aantal pogingen werkt. Deze test kan gedraaid worden met of zonder aangesloten transmitter, om beide een betrouwbare baseline te geven voor het aantal retries goed werkt met jou opstelling te vinde en om te kijken hoe de library zich gedraagt als het geen acknowledges krijgt.

Overige testmogelijkheden

Verdere testmogelijkheden betrekken meerdere due pins die aangestuurd worden op verschillende momenten, zo zijn er logicHigh en logicLow die pulseren op het moment dat er een logische hoog of logische laag wordt gedetecteerd door de receiver. Dit betekend dat de library makkelijk te controleren is in werking met behulp van een logic analyzer of een oscilloscoop.