COM I2C debugging

Voor de I2C opdracht moest er een systeem gemaakt worden waarbij allebei de microcontrollers masters moeten kunnen zijn, de masters sturen berichten naar elkaar via de bus wie de display mag gebruiken om meetwaardes van zijn sensor te displayen.

Hierbij ben ik begonnen met het maken van het protocol document waarin ik alle stappen en berichten tussen de 2 microcontrollers en de display heb weergeven, vervolgens ben ik in de code begonnen met een receiveEvent functie die het bericht leest gekregen van de andere controller. Ook heb ik een sendMessage functie gemaakt die de gewenste bericht stuur naar het andere systeem laatste belangrijke deel was de statemachine die bij houd wie de dominante systeem is en welke de slave. Deze statemachine zorgt ervoor dat er duidelijk is wie de bus heeft en wie naar de display mag sturen. Ook heb ik aan het systeem 2 knoppen toegevoegd waarmee je kan aangeven dat de controller de bus wil overnemen van de andere controller.

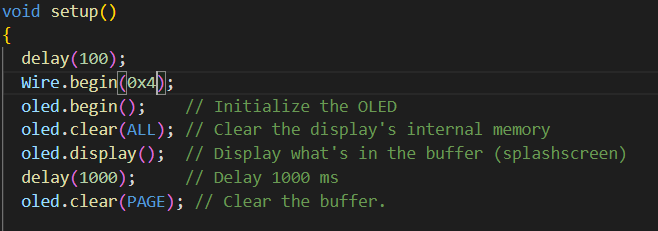
Nadat ik deze onderdelen gedaan had ben ik begonnen met het maken van de main loop, daarin ben ik gaan controleren of het systeem de dominant of de slave is met behulp van de statemachine. Als het systeem de dominant is heeft hij dus controle over de bus en mag hij zijn sensor waarde naar de display sturen. Als het systeem de slave is dan is het enige wat hij kan doen is op een knop drukken dat hij de bus over wil nemen of niet.

Nadat dit allemaal geimplementeerd was kwam ik er achter dat de display niet aan ging, ik ben eerst de bedrading gaan conroleren, dat leek de oorzaak niet te zijn. Vervolgens de code gecontroleerd met de voorbeeld code van de sparkfun site zelf. Ook daar zag ik geen problemen in, heb een extra project gemaakt met alleen de sparkfun code en die geupload naar de redBoard van sparkfun en ook toen werkte hij niet.

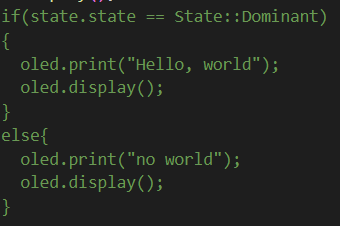
Na een aantal keer voorbeeldcode proberen had ik geen tijd meer om daar verder op te gaan dus had ik bedacht de demo te geven door hetzelfde protocol toe te voegen maar dan op de Serial print van beide bordjes. Ook dat werkte niet en kwamen we er later achter dat het idee van het printen op de I2C display anders is dan communiceren via I2C en dan printen op de Serial.

Na hulp van de docent kwamen we er achter dat de RedBoard kapot is en de display daarom niet werkte, ik heb de display getest door hem op de arduino aan te sluiten op de I2C bus en toen ging de display wel aan. Toen ben ik begonnen met de eerder gevonden basis code van sparkfun zelf op allebei de arduino’s te zetten om te kijken of dat wel goed ging. Dat werkte wel en de display laat het startup logo zien. Ik wilde gaan proberen om deze kennis toe te passen aan de code die ik al had geschreven. Ik heb al mijn eerdere code in comments gezet zodat ik stap voor stap weer kon gaan opbouwen.

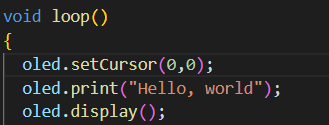
Ik ben begonnen met de basis code erin te zetten en de wire.begin() een address mee te geven, daarna in de nieuwe loop() een simpele if zetten met als dat als hij de dominant is dat hij dan hello world naar de display mag sturen anders zou hij op de serial zeggen dat hij niet de dominant is. Toen ging de display niet meer aan, terwijl van de basis code niks veranderd was. Toen dacht ik dat het misschien kwam doordat er op de verkeerde adressen geschreven werd bij wire.begin() maar dat bleek niet zo te zijn. Ook hebben we gecontroleerd of je het adress van het andere systeem mee moest geven bij display.begin() maar dat was het ook niet. Als laatste hebben we samen met de docent de display alleen aan de dominante systeem gehangen om te kijken of het probleem kwam in de bus wie mag schrijven en wie niet. De conclusie was dat hij daar ook niet aan ging en dat wegens tijd gebrek voor de deadline het verstandiger was om dit document te schrijven waarin we onze stappen beschrijven wat we hebben gedaan om de bugs op te lossen.



Figuur Basis code gegeven door sparkfun die werkt



Figuur Kijken of de state dominant is of niet



Figuur laatste poging om zonder enige verglijking iets te schrijven naar de display