Robot op afstand bedienen met bluetooth

**Internet of things 2**

## Beschrijving project

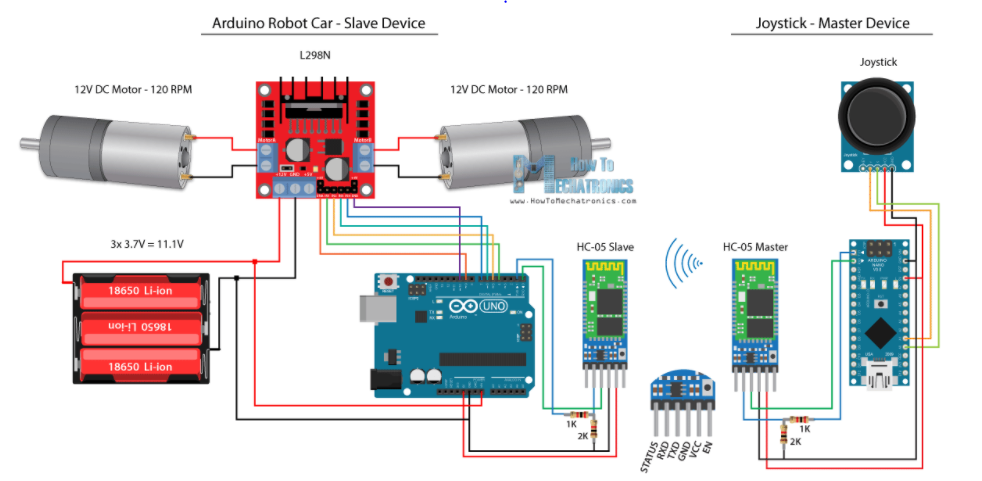
In dit project zal een point to point verbinding worden opgesteld tussen twee HC-05 modules die elk verbonden zijn aan hun eigen CY8KIT59 controller.

De eerste controller noemen we het control board, de bluetooth module die hier is op aangesloten zal functioneren als de master. Met het control board is het dus mogelijk om met de joystick de robot zich te laten verplaatsen.

De tweede module is aangesloten aan de robot die voor het vak project al reeds werd uitgewerkt. De robot zal functioneren als slave.

Om communicatie mogelijk te maken moeten eerst beide modules geconfigureerd worden, dit zullen we doen met behulp van putty. Ten tweede moet er een systeem uitgewerkt worden die het mogelijk maakt om data uit te wisselen.

In onderstaand schema is onze projectuitwerking voorgesteld. Het enige verschil is dat wij gebruik maakten van CY8KIT59 in plaats van Arduino. Dit leek ons een grotere uitdaging aangezien dit volledig nieuw is voor ons en we met Arduino al redelijk wat ervaring hadden.

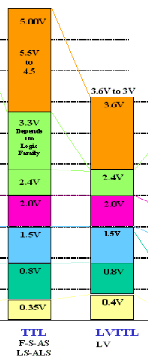


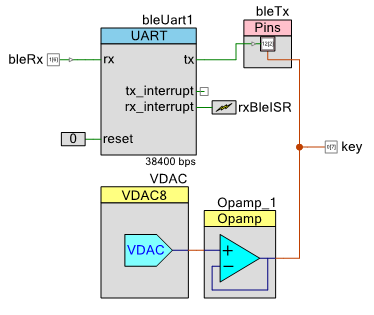
## Interfacing in PSOC

De modules krijgen een voeding van 5 V maar de data pinnen verwachten inputssignalen van maximaal 3.3 V en geven output signalen van 3.3 V dus hiermee zullen we rekening moeten houden.

Op Figuur 1 is te zien hoe dit mogelijk is gemaakt. De ‘bleRx’ input pin moet ingesteld worden op LVTTL zodat hij de signalen die de module stuurt kan lezen. De output pin ‘bleTx’ zijn drive level moet worden ingesteld op ‘Vref’, hierdoor kan je referentie spanning aansluiten aan de pin. Indien de pin een digitale 1 zal sturen zal het signaal dus even groot zijn als de aangesloten referentie spanning.

In dit geval hebben we aan die pin een VDAC aangesloten, deze kan je zodanig instellen dat hij als een spanningsbron werkt die 3.3 V levert. Aan de uitgang van de VDAC is een OPAMP aangesloten die geschakeld is als spanningsvolger. Deze heeft als doel om ervoor te zorgen dat we geen spanningsdaling zouden krijgen indien de kring belast wordt.





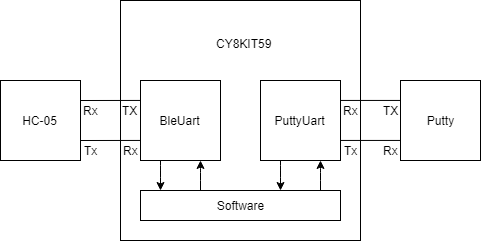
Figuur 1: Interfacing HC-05

Figuur 2: TTL vs LVTTL

Zoals reed eerder vermeld maakt de HC-05 gebruik van LVTTL signalen. Hierbij wordt een signaal als hoog gezien bij een minimale spanning van 2,4V. Het hoge signaal moet dus gelegen zijn tussen de 2,4V en de 3,6V. Bij een TTL niveau moet een hoog signaal gelegen zijn tussen de 3,3V en de 5,5V.

Verder hebben we de ‘key’ pin die we nodig zullen hebben om de bluetooth module in programmeer modus te zetten, deze is ook aangesloten aan de buffer opamp dus op de pin staat een constante spanning van 3.3 V. De ‘UART’ module heeft standaard instellingen behalve de baudrate, die is aangepast naar 38400 bps aangezien dat dit de standaard baudrate is van een HC-05 module. Als laatste hebben we een interupt aangesloten aan ‘rx\_interrupt’ deze interupt zal getriggerd worden indien er data binnen komt.

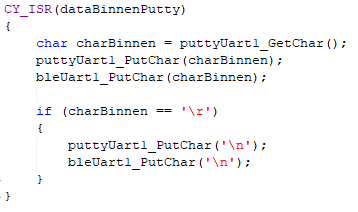
Op figuur 3 is het gebruikte blokschema afgebeeld. Zo is te zien dat de Rx steeds is aangesloten aan de Tx en omgekeerd. De 2 Uart blokken zijn via software met elkaar gelinkt zodat het mogelijk is om vanuit Putty de HC-05 te programmeren en om data die in de HC-05 binnenkomt zichtbaar te maken in Putty voor eventueel te debuggen.



Figuur 3: Blokschema

## Configuratie modules

Om de HC-05 modules te programmeren moeten we eerst en vooral aan de key pin van module een hoog signaal aanleggen. Hierdoor zal het ledje met een bepaald tijdsinterval knipperen. Als we nu AT commando’s sturen naar de module zal de module oneindig lang terug blijven antwoorden. Dit komt omdat putty enkel een ‘\r’ karakter stuurt na een druk op de enter toets. We moeten dus via software een ‘\n’ karakter toevoegen na elk ‘\r’ karakter. De code hiervoor is te zien op Figuur 3.



Figuur 4: Interrupt puttyUart Rx

De volgende commando’s moet je ingeven om een connectie op te stellen tussen de 2 HC-05 modules.

### Slave module

1. ‘AT’: hiermee kan je testen als je commando succesvol toe komt, de module antwoord met OK.
2. ‘AT+NAME=HC05\_SLAVE’: Hiermee stel je de naam in van de module.
3. ‘AT+PSWD=1234’: Als de module naar een paswoord vraagt geef je dit commando in.
4. ‘AT+ROLE=0’: Met dit commando stel je de module in als slave.
5. ‘AT+ADDR?’: Als je dit ingeeft krijg je het addres van de slave module te zien, noteer dit.

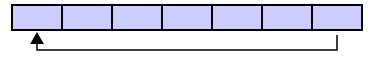
### Master module

1. ‘AT+RMAAD’: Door dit in te geven delete je de vorige connectie instellingen.
2. ‘AT+PSWD=1234’ : Als de module naar een paswoord vraagt geef je dit commando in.
3. ‘AT+ROLE=1’ : Met dit commando stel je de module in als master.
4. ‘AT+LINK=14,2,110007’: Geef nu dit commando in maar met je eigen slave adres.

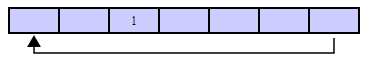
## Communicatie

Data die binnenkomt in de master of de slave wordt opgeslagen in een circulaire buffer. De werking van een circulaire buffer is als volgt:

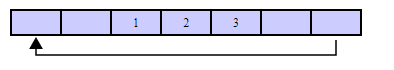
1. Een circulaire buffer is leeg in het begin en heeft een vooraf gedefinieerde grootte. Bv 7-element buffer.



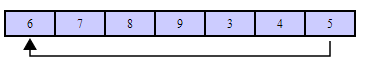
1. De startlocatie is niet zo belangrijk bij een circulaire buffer. Hieronder ziet u de start dat het getal 1 wordt opgeslagen.



1. Stel dat nadien cijfers 2 en 3 ook worden opgeslagen dan gebeurd dit na de 1.



1. Zo gaat het proces door tot de buffer volledig vol loopt. Nadien wordt deze gewoon weer overschreven en dit gebeurd steeds opnieuw.



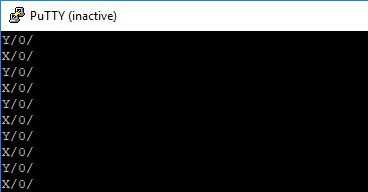
Bij een circulaire buffer zullen dus steeds de variabelen die al voor de langste tijd opgeslagen zijn verwijderd worden.

Een geldig commando die de slave ontvangt ziet er als volgt uit ‘M/FF/’. De M is het commando karakter, als je dit wijzigt kan je bepaalde dingen aansturen. in Tabel 1 staan de gebruikte commando’s voorgesteld. Het deel tussen de ‘/ ‘ karakters is de data. De data wordt hexadecimaal doorgestuurd en zal een waarde representeren tussen 0 en 255

Tabel 1: Geïmplementeerde commando's

|  |  |  |  |
| --- | --- | --- | --- |
| Commando karakter | Beschrijving | Range | Richting |
| I | Stuur de infrarood data van de robot door naar het control board | 0-255 | Robot -> control board |
| T | Stuur een testgetal door om zichtbaar te maken op de lcd van het control board | 0-255 | Robot -> control board |
| X | Stuur de X waarde van de joystick door naar de robot | 0-255 | Control board -> robot |
| Y | Stuur de Y waarde van de joystick door naar de robot | 0-255 | Control board -> robot |
| M | Schakel de motoren aan/uit | 0-1 | Control board -> robot |

Een praktisch voorbeeld die we nodig hadden om onze joystick te ontdekken is als volgt:



We lieten in putty zien wat de x-waarde en de y-waarde is van de joystick. Het commando bevat eerst en vooral dus X of Y. nadien tussen de 2 backslashes is de data terug te vinden. De data kan variëren tussen 0 en 255 maar dan hexadecimaal voorgesteld. 127 zowel voor X als voor Y is de status waarbij de joystick in rusttoestand is. Indien we de robot vooruit willen laten gaan moeten we de Y waarde dus naar 255 brengen wat dus volgende commando creëert: **Y/FF/ .**

De FF wijst dus op 255 wat wil zeggen dat we vooruit willen gaan. Indien de waarde 0 is dan gaan we achteruit.

Naar rechts gaan geeft volgend commando: **X/FF/.** Indien we naar links willen gaan dan wordt dit: **X/00/.**

Met het commando karakter M is het mogelijk om de motoren aan of uit te schakelen. Bijvoorbeeld: **M/01/.** Dit commando zal de motoren activeren.