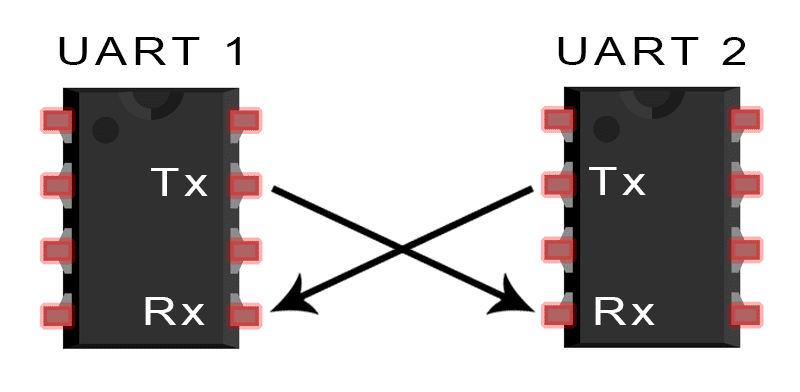
UART Design

Brent Schoenmakers & Cihan Kurt



# Inleiding

Dit is ons design document voor UART. Dit document bevat een state diagram voor het ontvangen van een bericht en een state diagram voor het verzenden van een bericht. De statediagrammen zijn doormiddel van tekst uitgelegd.

Inhoudsopgave

[Inleiding 1](#_Toc526931808)

[Ontvangen 3](#_Toc526931809)

[Verzenden 4](#_Toc526931810)

[Clockcycles 5](#_Toc526931811)

[Bronnen 5](#_Toc526931812)

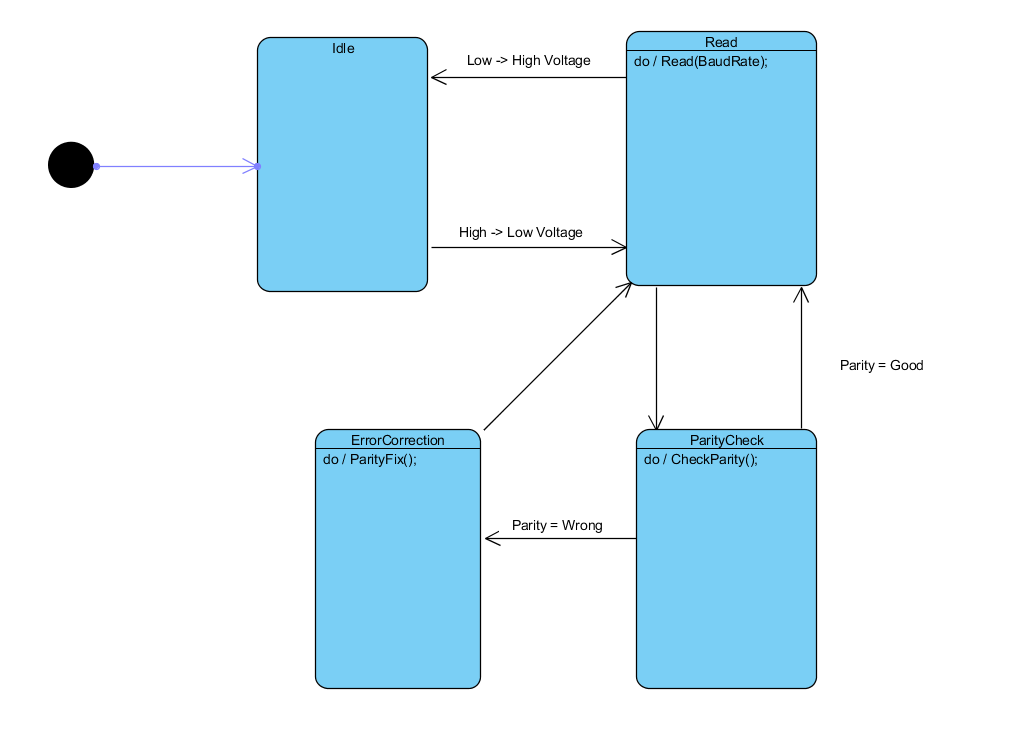
[V2 6](#_Toc526931813)

[Setup foto: 9](#_Toc526931814)

[Demo 10](#_Toc526931815)

# Ontvangen

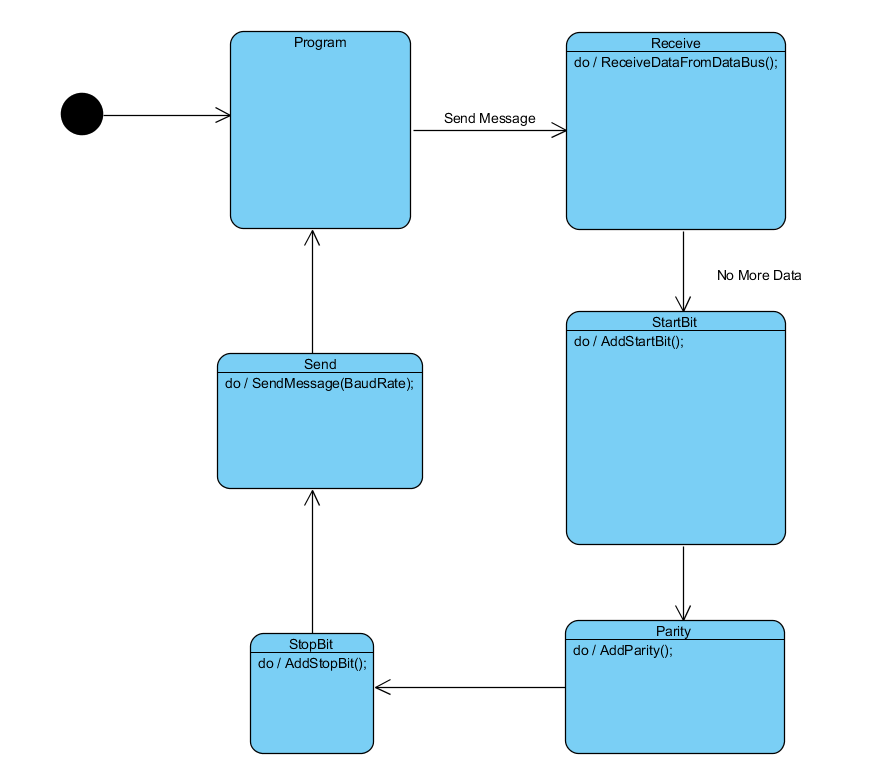
De statemachine van de ontvanger van het bericht ziet er als volgt uit voor ons:



Wij gaan ervanuit dat de ontvanger op het moment niks aan het doen is. Mocht deze toch acties moeten uitvoeren dan kan de state Idle verandert worden naar Program.   
Zodra er een switch van HIGH naar LOW wordt gemaakt gaat de ontvanger de rest van het bericht lezen met de gewenste baudrate. Op dit bericht wordt dan een errorcorrectie uitgevoerd. Zodra alles goed is wordt de stopbit gedetecteerd. Het ontvangen van het bericht stopt op dit moment.

# Verzenden

De statemachine van de zender van het bericht ziet er als volgt uit:



Gezien dit een zender is, moeten er ook bepaalde acties uitgevoerd worden om een bericht te willen verzenden. Hierdoor hebben wij gekozen voor Program en geen Idle als beginstate. Zodra het programma een bericht wilt verzenden wordt de data voor dit bericht opgehaald uit de databus.  
Zodra er geen data meer op te halen is wordt aan deze data een start bit toegevoegd.   
Hier hoeft geen check op te komen. De parity bits worden meteen erna toegevoegd en vervolgd door een stop bit. Vervolgens zal het bericht verzonden worden met de gewenste baudrate.   
Er hoeft niet gechecked worden op het succesvol verzenden van het bericht, door de parity. De zender kan meteen terug naar de rest van het programma.

# Clockcycles

De opdracht was het maken van een design voor UART. UART staat voor Universal Asynchronous Receiver/Transmitter. Dit betekent dat de ontvanger en zender niks weten van elkaars clockcycle. Ze maken gebruik van de zelfde baudrate om alles goed te laten verlopen.  
  
Het gebruiken van de clockcycles zal ervoor zorgen dat dit geen UART maar USART (Universal Synchronous/Asynchronous Receiver/Transmitter) wordt.

<https://www.mikroe.com/blog/uart-serial-communication>

Dit hebben we hier gelezen.

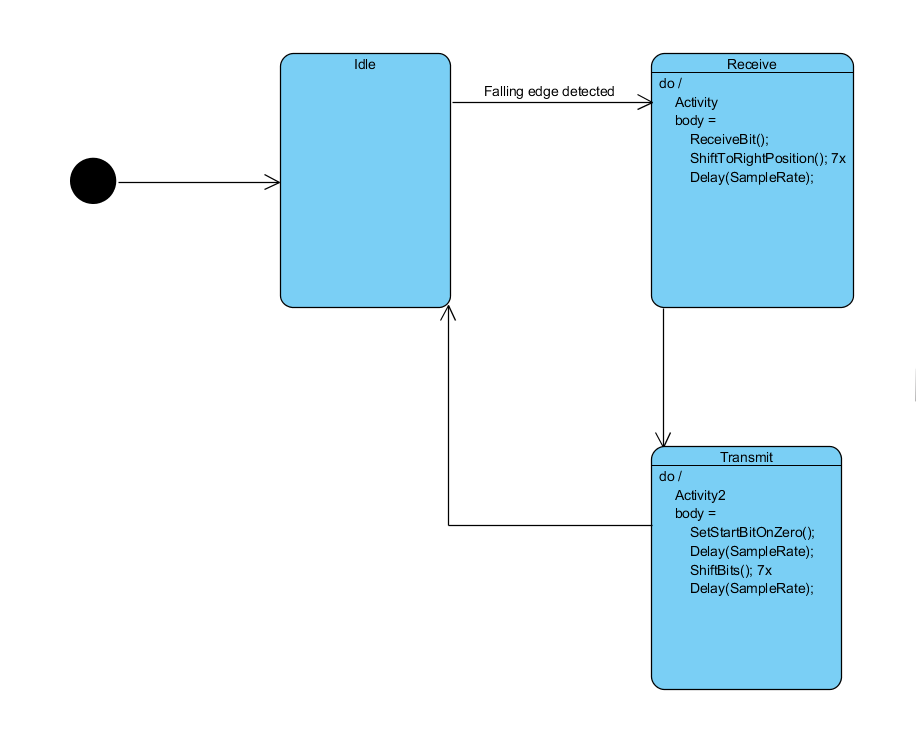
# Bronnen

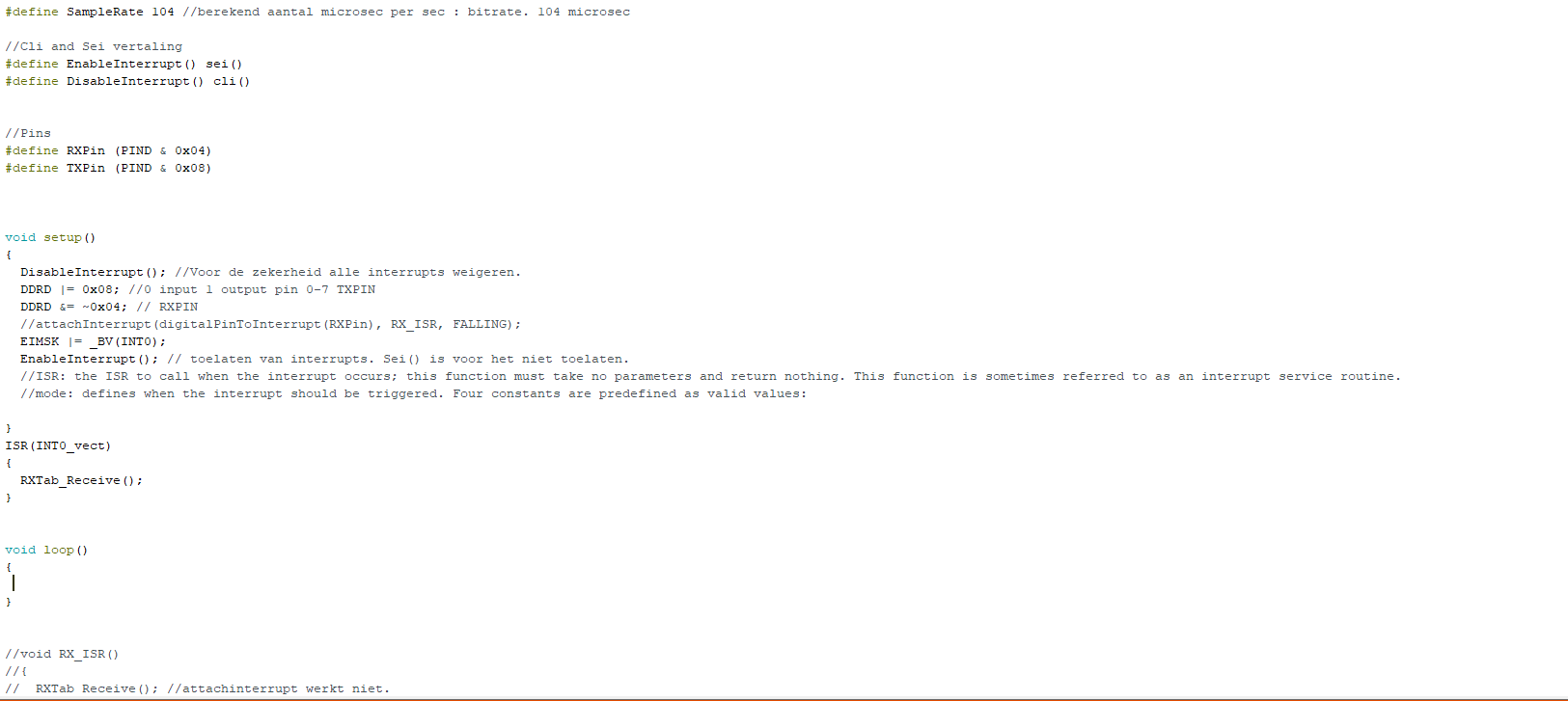
<https://www.mikroe.com/blog/uart-serial-communication>

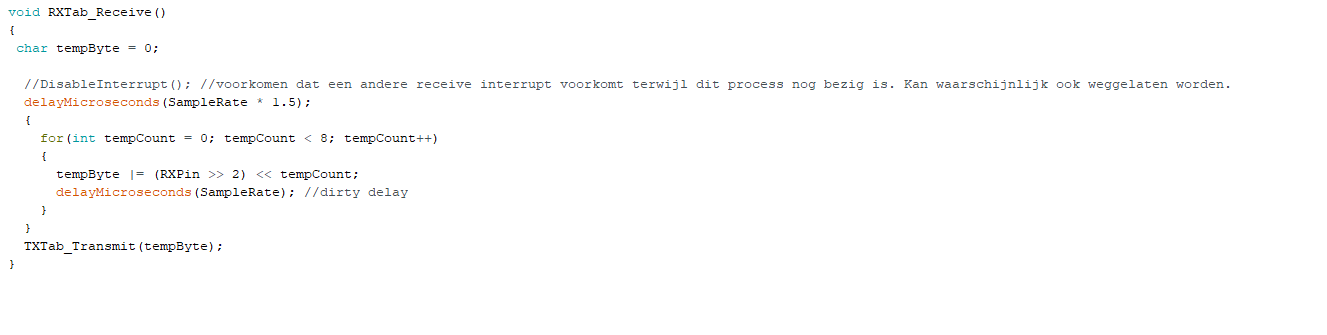
<http://www.circuitbasics.com/basics-uart-communication/>

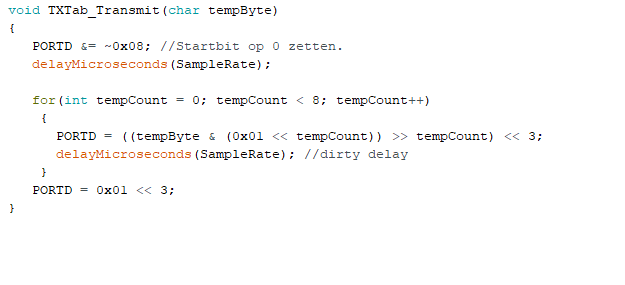
- Canvas Presentatie FHICT.

# V2

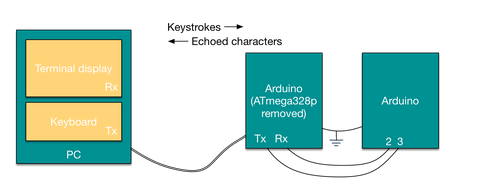
  
Na positieve feedback ontvangen te hebben op onze vorige statediagram hebben we deze toch een beetje aangepast. Dat was ook de tip van Jan Dobbelsteen.  
In plaats van twee aparte statemachines voor het ontvangen en verzenden hebben we er nu een statemachine van gemaakt.   
  
Het ontvangen begint wanneer er een falling edge is gedetecteerd. Vervolgens wachten we heel even en ontvangen we de rest van de bitjes. Deze shiften we meteen naar de juiste positie. Tussen het ontvangen van de bitjes zit er ook een delay van 104 ms (SampleRate).  
Als alle bitjes zijn geshift gaan we meteen door naar het versturen van de ontvangen bitjes. Dit gaat bijna op de zelfde manier. We sturen een startbit, wachten heel even en verzenden dan de rest van de bitjes. Dit verzenden doen we 7x met daartussen weer een delay van 104 ms. Als dit is gedaan gaat het programma automatisch terug naar ‘program/idle’. Dit komt door het gebruik van een interrupt. De code die dit waarmaakt staat op de volgende pagina’s.





  
Een aantal regels in de code, waaronder de interrupts, staan gecomment. Die regels code werkten helaas niet, maar het leek ons wel handig om het erin te laten staan om progressie aan te tonen.

## Setup foto:

Om aan de eisen van de setup te voldoen hebben we gekeken naar het volgende voorbeeld op de canvas pagina.  


In plaats van het verwijderen van de ATMega chip hebben we deze Arduino op resetmode gezet. Hierdoor voert hij geen code uit. Dit hebben we gedaan door de ground aan de reset te verbinden.  
  
Zie foto hieronder voor onze setup:

Afbeelding met elektronica, circuit

Beschrijving is gegenereerd met hoge betrouwbaarheid

## Demo

Het leek ons niet handig om een printscreen te maken van Putty, want hierin kun je niet zien of het programma ook echt doet wat het moet doen. Een demo kunnen wij in de klas geven, mocht dit nodig zijn.