Documentatie Embedded systems   
Sprint 5

Lucas van Lippen

T3DB

# PID simulatie

Het idee is nu om een servo feedback systeem te maken dat zichzelf op de juiste snelheid zet aan de hand van de positie input die hij van de feedback pin binnen krijgt. Hiervoor is het handig om eerst een PID simulatie te maken.

Diagram

Description automatically generated

Hiervoor moet ik eerst de overdrachts-functie en overdrachts-vertraging bepalen.

Om de overdrachts-vertraging te bepalen moet je kijken naar de wanneer de servo pas besluit omhoog te gaan in snelheid. Volgens de meetgegevens in de Excel-lijst voor een feedback servo is dat na 0.032 s.  
Om de overdrachts-functie te bepalen moeten we de tau tijdsconstante weten.   
Daar kunnen we achter komen door de amplitude snelheid te bepalen die de servo kan draaien.   
Als we dan weer kijken naar de Excel-lijst dan komen we erachter dat die op 2.32 RMS. Als we die dan vermenigvuldigen met 0.63 dan komen we uit op een waarde van 1.47 RMS. Deze omwentel snelheid word voor het eerst bereikt op 111ms nu kunnen we dus de tau-waarde berekenen door 0.111 – 0.032 en dan komen we dus uit op 0.079 als tau waarde die we voor de overdrachts-functie kunnen gebruiken.

Graphical user interface, text, application, email

Description automatically generated

Nadat we die waardes hebben vastgesteld kunnen we de PID waardes bepalen door ze te tunen. Hierbij gebruik ik een discrete tijdsperiode van 0.020s omdat we met een digitaal signaal werken.

A screenshot of a computer

Description automatically generated with medium confidence

Grafiek die ik als resultaat krijg.

Chart, line chart

Description automatically generated

Zichtbare overdrachts-vertraging.

# Project 1: PID systeem implementatie

## Code

Text

Description automatically generated

Text

Description automatically generated

Hier worden de twee timers voor de servo ingesteld om hem te besturen en de hoek te bepalen.

Text

Description automatically generated

Hier worden alle constanten ingesteld en alle globale waarden die bijgehouden moeten worden.

Text

Description automatically generated

In de GetServoSpeed-functie wordt er op het laatste moment ook nog gecheckt of de waarde ook mogelijk kan zijn in vergelijking met wat de specificaties zeggen. De servo’s die wij gebruiken kunnen niet sneller draaien dan 2.33 omwentelingen per seconden. Dus als de waarde daar buiten 5 omwentelingen valt wordt de oude waarde aangehouden. Dit is niet de beste manier om het te doen omdat ik in essentie een sample weg gooi, dus in de volgende oplevering ben ik van plan dit op te lossen met fatsoenlijke wrap-around code.

Text

Description automatically generated

Hier is de omzettings functie die de snelheid omzet van een snelheid in RPS naar een pulsebreedte die de servo kan begrijpen.

Zoals je zag bij de simulatie heb ik 3 PID waardes gekregen die ik kon gebruiken om ervoor te zorgen dat de servo ook bij weerstand goed zou draaien als er weerstand op zou komen. Die heb ik in de code ingesteld. Ik heb daarna nog wel de kP-waarde aangepast omdat hij daardoor zijn snelheid beter aan paste.

Text

Description automatically generated Text

Description automatically generated

Hierbij heb ik daarna wat PID controller code geschreven die ervoor zorgt dat de servo motor goed loopt.

## Resultaten

Text

Description automatically generated

Hier kun je zien hoe de draaisnelheid omhoog gaat van 0 naar 1.

Graphical user interface

Description automatically generated

Hier is het aanstuursignaal zonder dat ik hem probeer vast te houden.

Graphical user interface

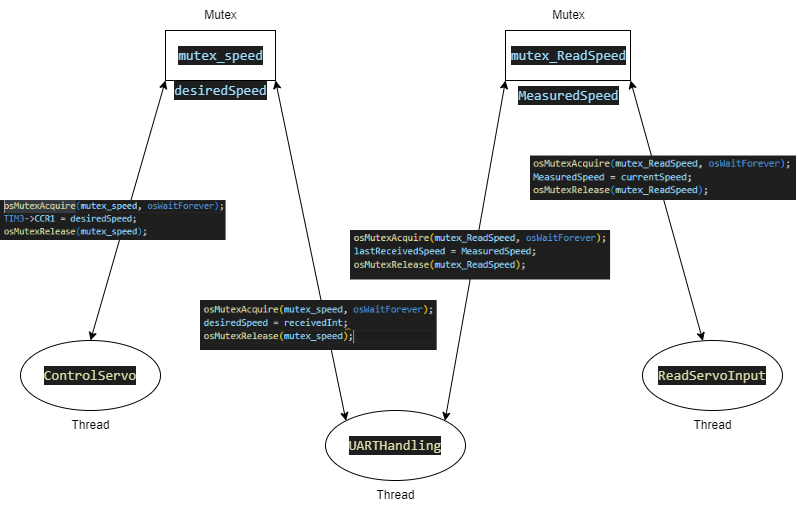
Description automatically generated with medium confidence

Hier is het aanstuursignaal terwijl ik hem probeer vast te houden. Hij is nu net een beetje dunner omdat hij sneller probeert te draaien.

# Project 2: Een input terminal op een andere thread

Het plan is om een systeem te maken waarmee ik via een serieel prompt een servo precies kan aansturen op draai snelheid. Hierbij wil ik een feedback control systeem maken die ervoor kan zorgen dat de servo zo accuraat mogelijk draait. Hiervoor wil ik een feedback servo gebruiken die ook een signaal terugstuurt waarmee je kan zien op welke positie hij staat, hiervoor heb ik dus een pwm input nodig.

Hiervoor ga ik mutexes gebruiken om de data uitwisseling tussen de 3 threads thread safe te maken hiervoor heb ik ook een diagram gemaakt.

  
Text

Description automatically generated

Text

Description automatically generated

Hier worden de twee timers voor de servo ingesteld om hem te besturen en de hoek te bepalen.

## Code

Text

Description automatically generated

Hier initialiseer ik alle threads, mutexes en de shared resources.

Text

Description automatically generated

Dit is de code voor de UART-Thread hier worden alle UI functies afgehandeld. Als er een input binnen komt, dan wordt de desiredSpeed dus aangepast als hij toegang tot de Mutex heeft en daarna wordt de huidige gelezen snelheid uitgeprint om te laten zien hoe snel de servo op dat gegeven moment draait.

A screenshot of a computer

Description automatically generated with medium confidence  
In de ControlServo-thread vraagt de functie elke 20 milliseconden om de desiredSpeed zodat hij de servo snelheid kan veranderen.

Text

Description automatically generated

In de ReadServoInput-thread wordt de puls breedte-verschil bepaald met de vorige met de huidige te vergelijken en daarna om te rekenen naar een waarde in Rotations Per Second. Dit doet hij elke 10 milliseconden. Hier moet nog wel gekeken worden of ik de wraparound kan oplossen. Verder zal ik ook één tijdsinterval moeten gaan gebruiken om ervoor te zorgen dat alles in synchroom gaat lopen als ik hier ook nog een PID systeem aan ga hangen.

A screenshot of a computer

Description automatically generated with medium confidence

Hier worden de 3 threads gestart en worden de registers voor de servo pinnen aangezet met de bovenste 2 functies.

# Plan voor de volgende sprint

Volgende sprint is het mijn doel om deze projecten te combineren en als het mogelijk is ook nog FreeRTOS z’n messagequeues te gebruiken. Als het mij niet lukt om messagequeues erin te verwerken dan zal ik een nieuw project gaan maken om dit weer aan te tonen. Verder wil ik ook nog het wrap-around oplossen in de uiteindelijke code omdat mijn huidige manier in de PID controller wel creatief is maar in essentie gooi ik nog wel gewoon een sample weg en dat wil ik graag oplossen.