

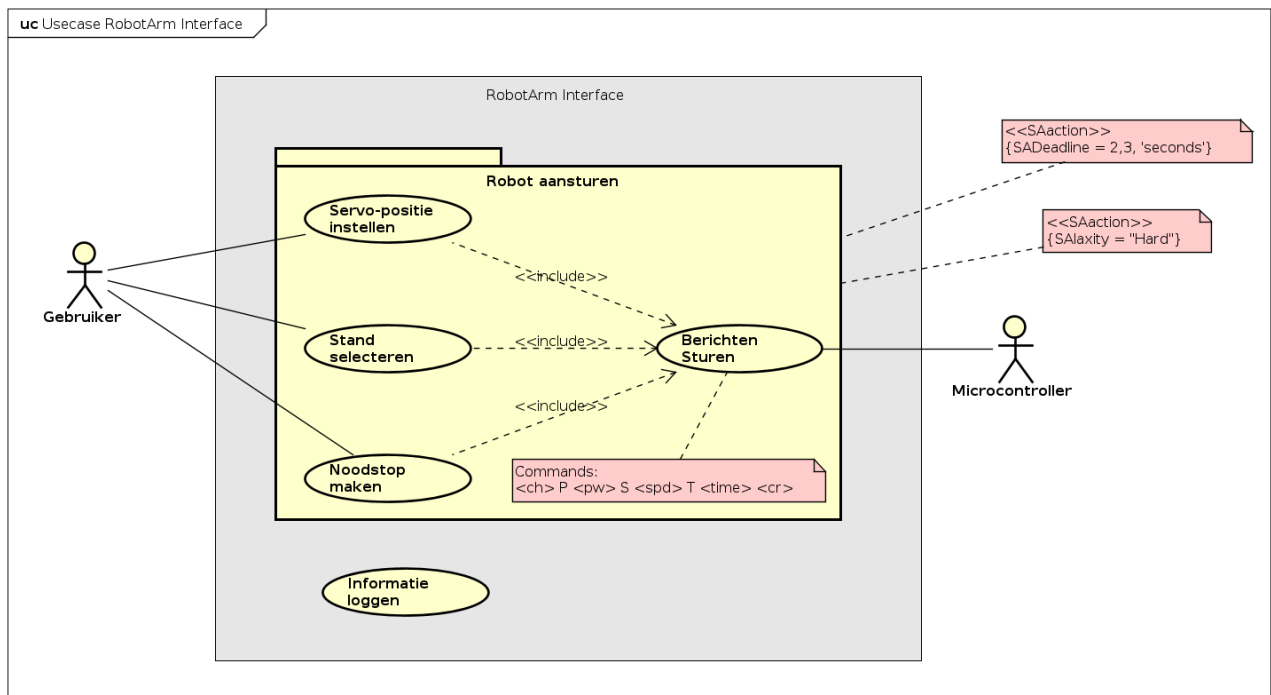
WOR-Interface beroepsproduct

Naam: Kethan Ponnuthurai & Ernesto Tersluijsen

Table of Contents

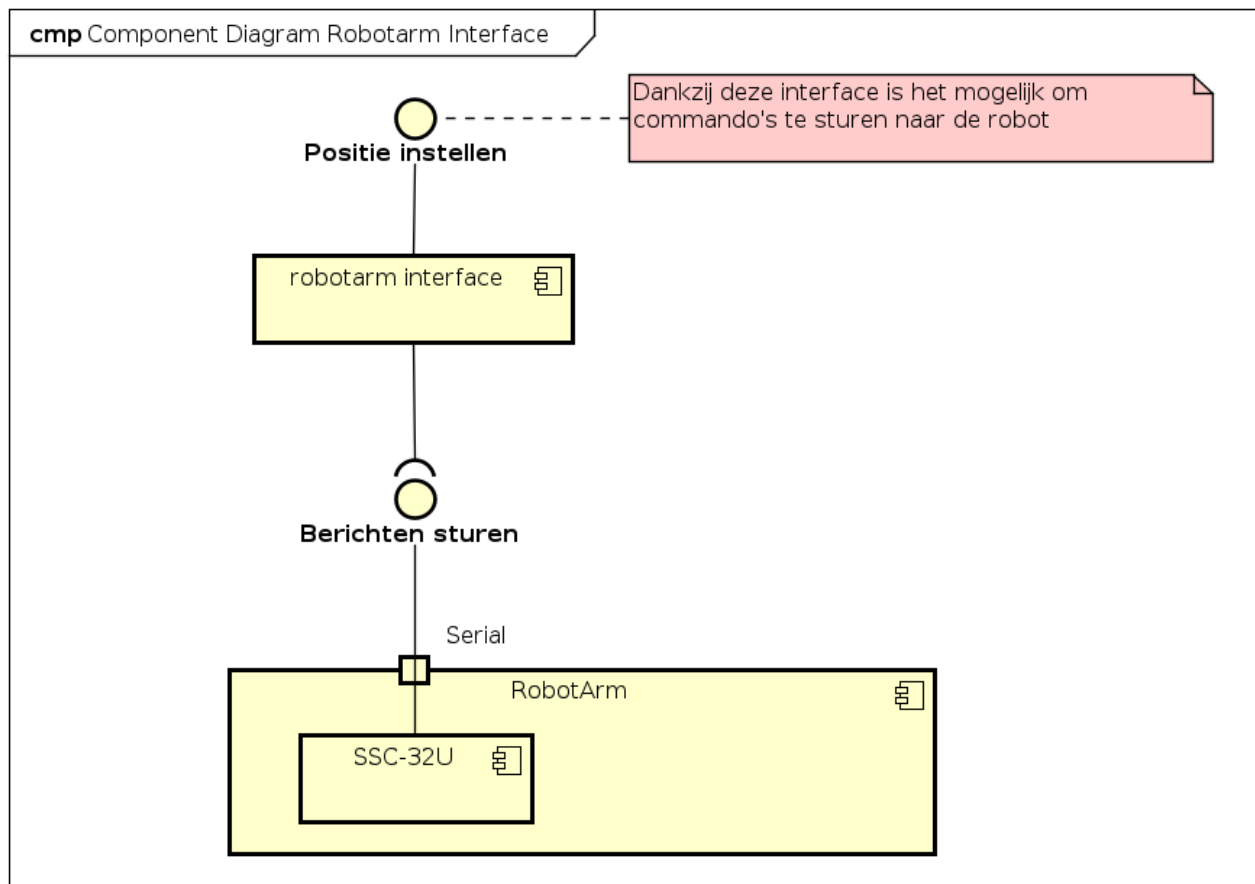
Usecase diagram.....	3
Component diagram.....	4
Protocol State Machine.....	5
Sequence Diagram.....	6
State Diagram.....	7
Class Diagram.....	8
Timing diagram.....	9
Quality of service.....	10
Timeliness.....	10
Safety.....	10

Usecase diagram



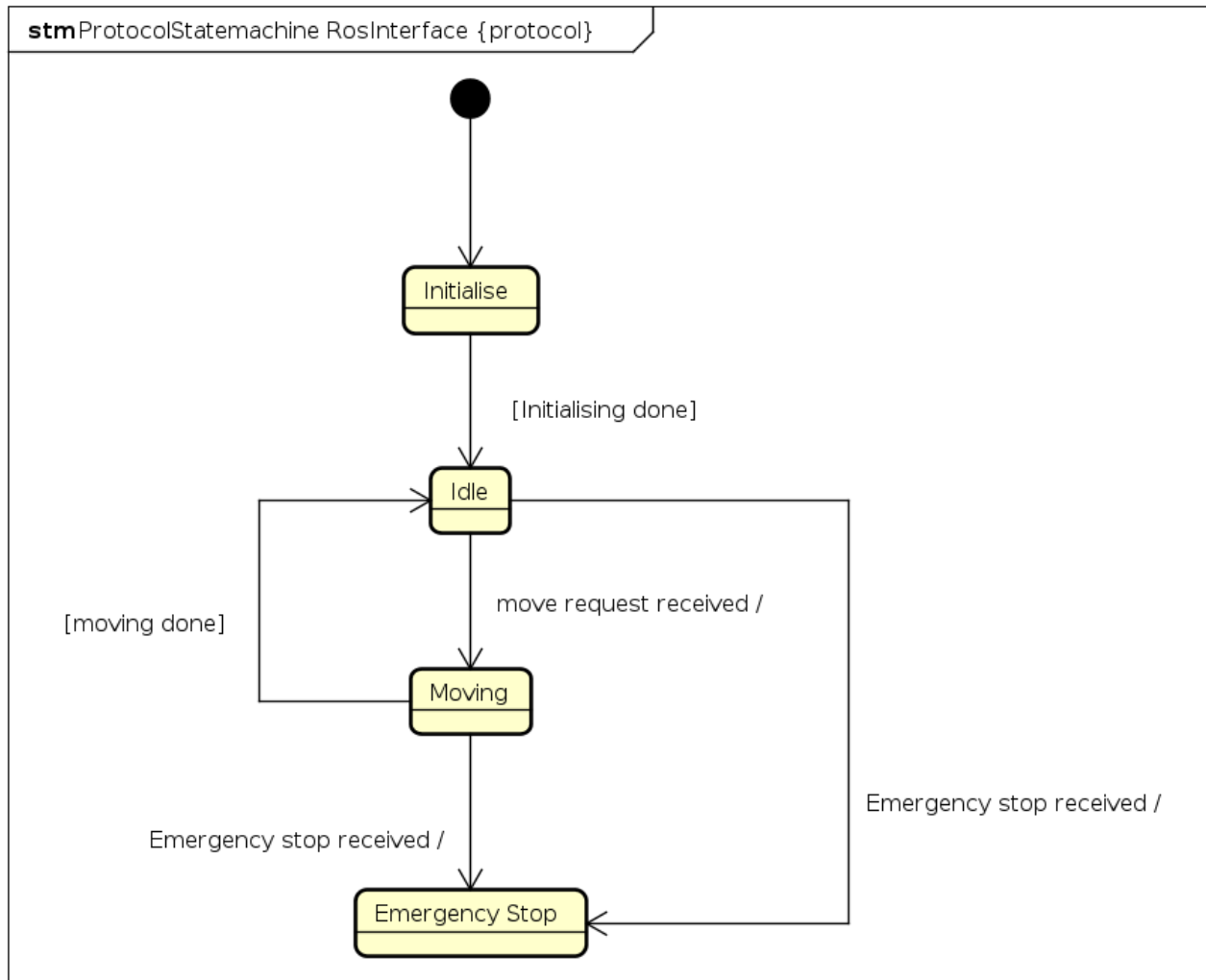
Met de robotarm interface kan de gebruiker 3 acties uitvoeren. De gebruiker kan de robotarm in een specifieke stand zetten, hiervoor moet hij de servo's en hoeken van de servo's opgeven. Daarnaast kan de gebruiker de robotarm in verschillende voorgemaakte standen zetten zoals "Park", "Ready" en "Straight up". Ten slot kan de gebruiker ook een noodstop maken. Al deze use cases sturen berichten naar de microcontroller om die acties uit te voeren.

Component diagram



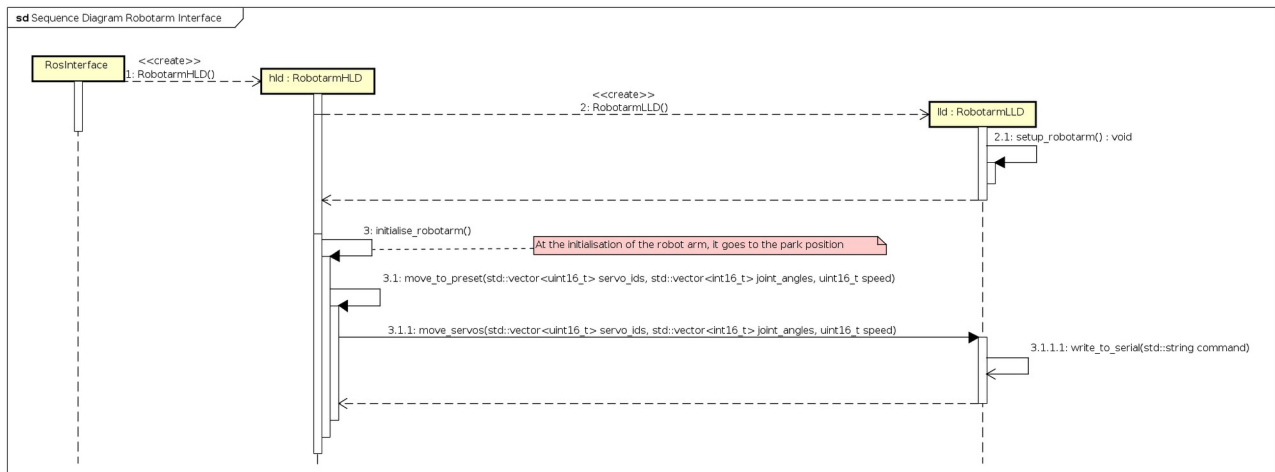
Naam Interface & poorten	Beschrijving functionaliteit
Positie Instellen	In deze interface geef je de positie of de preset door aan de interface (terminal) (Park, Ready or Straight)
Berichten sturen	Door middel van deze interface, kan je de positie van de servo's sturen van de robotarm interface naar de microcontroller van de robotarm. Dit gaat via Serial. De syntax van de commando's is zoals beschreven in de AL5D robotarm documentatie.
Serial	Voor de communicatie met de robotarm wordt gebruik gemaakt van Serial. De Baud rate is ingesteld op 115200.

Protocol State Machine



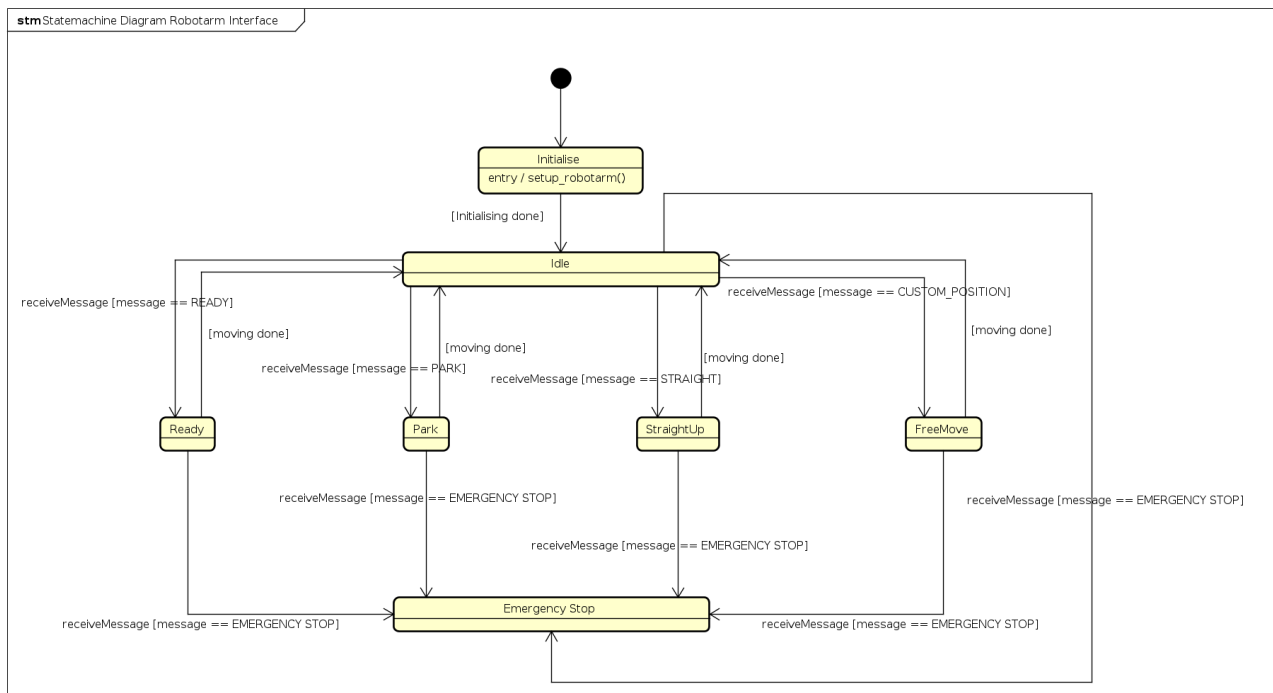
Als het programma optstart belandt het in de Initialise state, hier wordt de robotarm geïnitieerd. Zodra de initialisatie klaar is belandt het programma in de Idle state. Hier kan de gebruiker 2 acties uitvoeren, de gebruiker stuurt een request om de robotarm te bewegen, als de gebruiker dit doet gaat het programma naar de state moving. Terwijl het programma zich in de state moving bevindt kan er ook een emergency stop request gestuurd worden, waarna het programma zich in de emergency stop state bevindt, dit zal zo blijven totdat het programma afgesloten wordt. Als de gebruiker in de Idle state een emergency stop request stuurt zal het programma ook naar de emergency stop state gaan.

Sequence Diagram



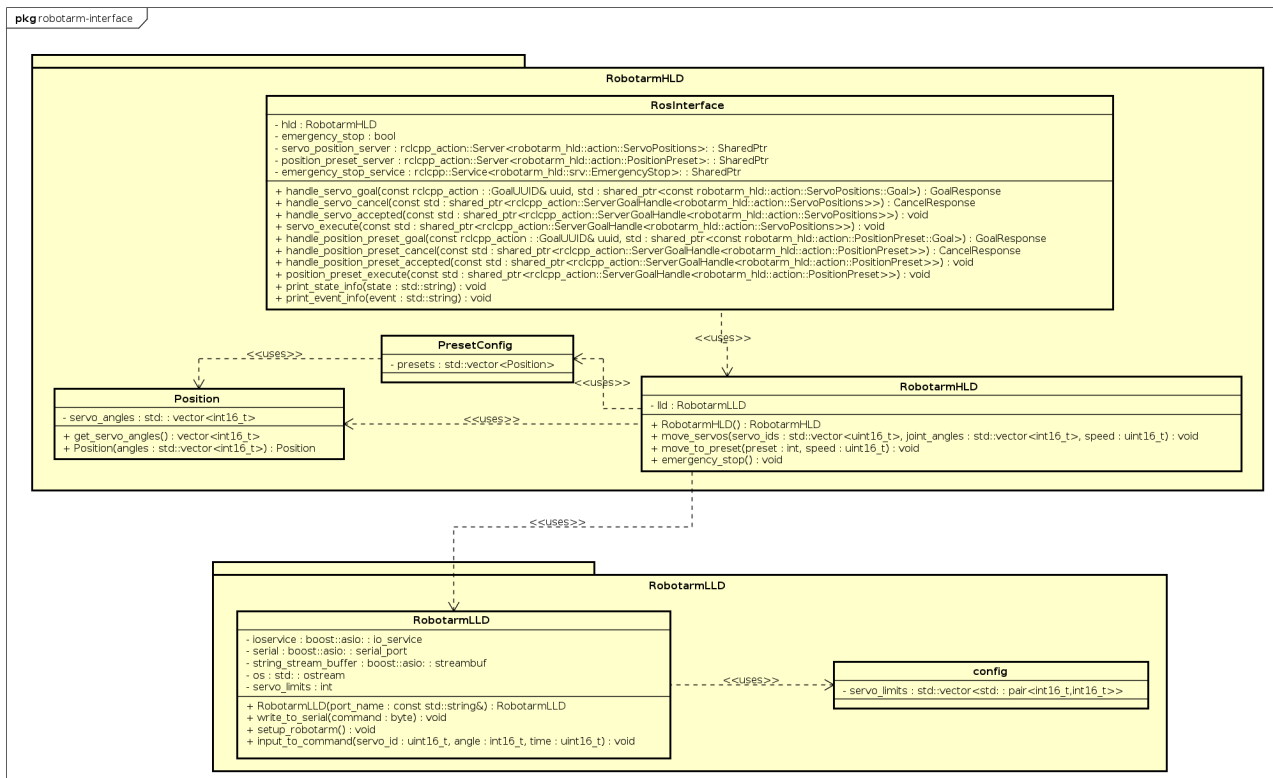
Bij het initialiseren van het programma roept de RosInterface de constructor aan van RobotarmHLD. De RobotarmHLD roept dan de constructor voor de RobotarmLLD aan, deze constructor roept de functie setup_robotarm aan, deze initialiseert de seriele communicatie met de robotarm. Zodra de seriele communicatie geïntialiseerd is gaat de hld bewegen naar de “Park” preset. De functie move_to_preset roept de move_servos functie van lld aan en zodra de lld klaar is met het sturen van de commando's is de initialisatie klaar.

State Diagram



Naam State	Beschrijving functionaliteit
Initialise	Bij deze state wordt de communicatie met de robotarm geïnitieerd.
Idle	In deze state wacht de robot tot zijn volgende commando.
Ready	In deze state verplaatst de robotarm zich van zijn huidige stand naar de “Ready” stand.
Park	In deze state verplaatst de robotarm zich van zijn huidige stand naar de “Park” stand.
StraightUp	In deze state verplaatst de robotarm zich van zijn huidige stand naar de “Straight up” stand.
FreeMove	In deze state verplaatst de robotarm zich naar de aangegeven positie die de gebruiker kan aangeven in het bericht wat naar de robotarm interface gestuurd wordt.
Emergency stop	In deze state wordt de robotarm stop gezet. De robotarm stopt met de huidige beweging en kan geen nieuwe bewegings requests ontvangen.

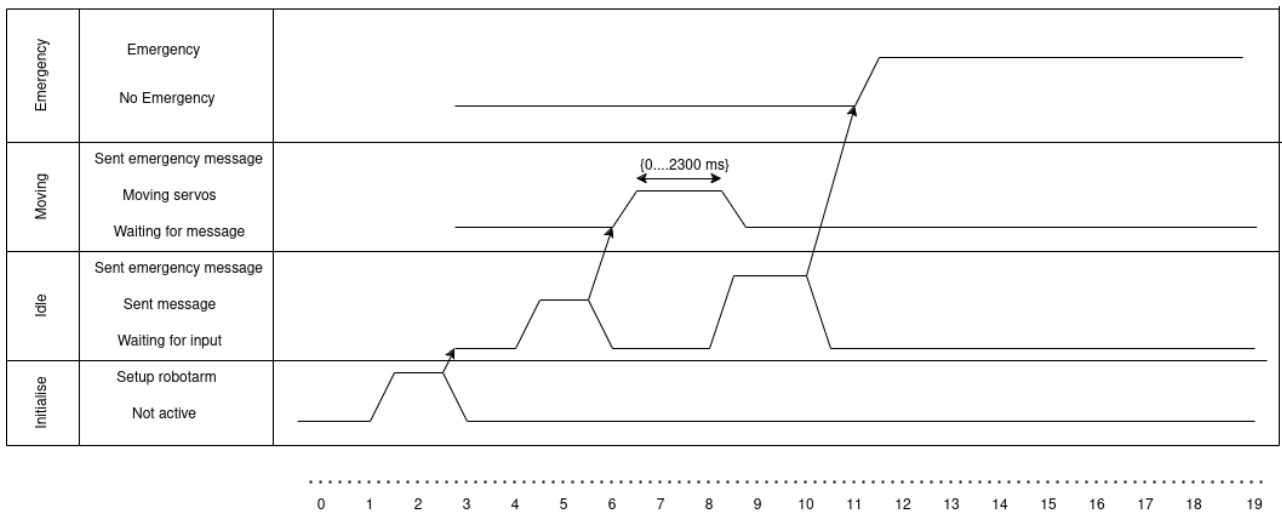
Class Diagram



In de robotarm interface zijn er 2 “packages”, de RobotarmHLD en de RobotarmLLD, waarvan de RobotarmLLD een dynamically linked library is voor de RobotarmHLD. De RobotarmLLD package bevat een RobotarmLLD klasse die verantwoordelijk is voor het omzetten van algemene commands naar commands die specifiek de AL5D robotarm gebruikt. Daarnaast bevat de RobotarmLLD package een bestand met de configuratie voor de limieten op de servo’s.

De RobotarmHLD package is verantwoordelijk voor de ROS2 communicatie. De RosInterface klasse is de ROS node die de action servers en service bevat. Zodra er een bericht op de action server of service komt roept de RosInterface de bijbehorende functie uit de RobotarmHLD aan. De RobotarmHLD klasse is verantwoordelijk voor de preset posities af te handelen en de bijbehorende functies uit de low level driver aan te roepen. De Position klasse bevat de verschillende hoeken de servo’s moeten staan om in een preset positie te staan en de PresetConfig zorgt ervoor dat alle posities makkelijk zijn aan te passen door deze in een bestand samen te voegen.

Timing diagram



In het timing diagram zie je vier staten, waarbij elke staat een event heeft. De Initialisatie State heeft namelijk twee events: het initialiseert de robotarm, en verder doet de staat niet veel. Wanneer de robot geïnitieerd is, bevindt het zich in de idle state, in afwachting van het input event. Als een gebruiker de juiste vorm van het bericht doorgeeft, wordt het systeem naar de moving state gebracht. In deze staat draaien de servo's naar de opgegeven hoek die de gebruiker heeft ingesteld. Daarnaast kan er ook een noodbericht worden verzonden vanuit de idle state of de moving state. In dat geval gaat het systeem naar de noodstate, waarbij de robotarm wordt gestopt en niet meer verder werkt, en er geen berichten meer kunnen worden verzonden.

Quality of service

Timeliness

Bij onze implementatie duurt de beweging van de robot van de ene positie naar de andere maximaal 2300 milliseconden. Dit is zichtbaar in het timingdiagram binnen de moving state.

Safety

Bij onze implementatie kan de robot niet verder gaan dan de geprogrammeerde limieten.

Bijvoorbeeld, wanneer een servo een maximale hoek van 90 graden heeft en er wordt 130 graden verzonden, wordt dit automatisch aangepast naar 90 graden en kan de servo niet verder draaien. Als je naar het timingdiagram gaat, zie je ook bij moving en de idle state het event 'sent emergency state.' Hiermee kan de gebruiker de robotarm op elk moment laten stoppen.