Ontwerpdocumentatie Robotarm simulatie

Geschreven door: André Slokker

1. Structuur van de packages

De code bestaat uit drie verschillende packages.

Package	**Functie**
Controller	De functie van de controller is het sturen van SSC32U commando's naar de robotarm.
Object	De functie van het object package is het visualiseren van een bekertje. Daarnaast zit er een functie in die ervoor zorgt dat het bekertje kan vallen.
Robotarm_description	De robotarm_description is de belangrijkste package in dit project. Deze package zorgt ervoor dat commando's van de Controller worden verwerkt. Deze commando's worden uitgevoerd op een robotarm die gevisualiseerd wordt in Rviz. Daarnaast zit er ook een Gripper klasse in deze package die kijkt of het bekertje tussen de gripperarmen zit.

Er draait ook nog een extra node namelijk de robot_state_publisher. Deze node zorgt ervoor dat de TF library de juiste transformaties van de joint_states krijgt.

Daarnaast worden er verschillende berichten verstuurd tussen de packages

Topic	**Functie**
SSC32UPosition	Dit topic wordt gebruikt om commando's te versturen van de Controller naar de Robotarm_description. Deze commando's moeten volgens het SSC32U protocol zijn.
visualization_marker	Dit topic wordt gebruikt om de pose van de marker naar de robotarm_description package te sturen en naar Rviz.
gripper_ctr_pos	Dit topic bevat het middelpunt van de gripper zodat de cup weet naar welke positie hij moet gaan (als hij is opgepakt door de gripper).
marker_in_gripper	Dit topic wordt gebruikt om te sturen of de gripper het bekertje vastheeft.
joint_states	Dit topic wordt gebruikt om de joint posities van de robotarm te sturen.

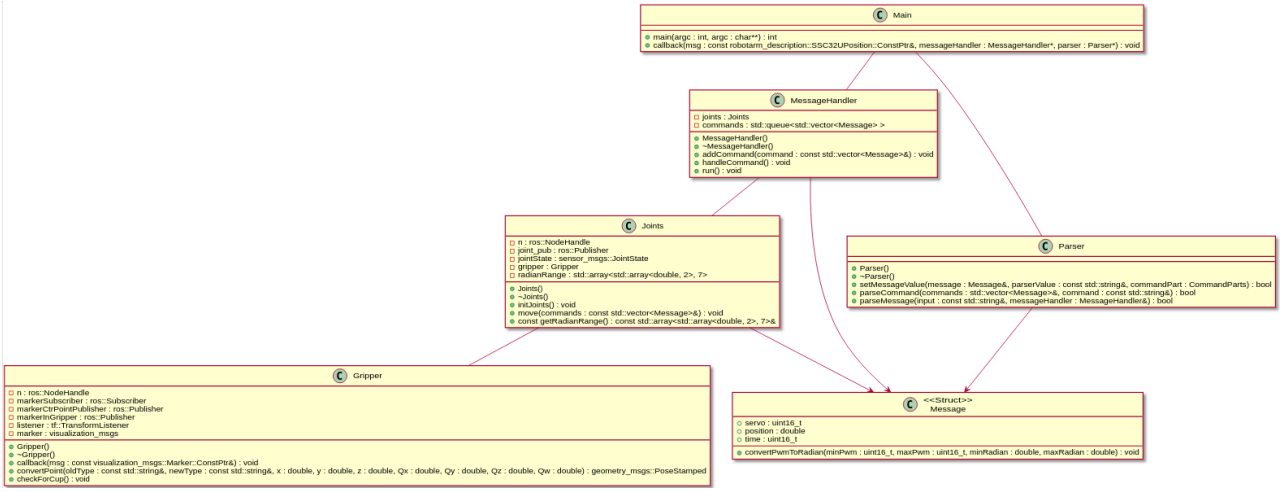
Daarnaast worden er ook verschillende message types gebruikt.

Message type	**Functie**
visualization_msgs/Marker	Dit message type bevat alle informatie over de marker. Zo bevat hij een pose en enkele dingen zoals de kleur en de schaal.

std_msgs/Bool	Dit message type wordt gebruikt bij het topic “marker_in_gripper”. Deze message bestaat uit een boolean waarde.
geometry_msgs/PoseStamped	Dit message type wordt gebruikt om Pose variabelen door te sturen.
robotarm_description/SSC32UPosition	Dit message type wordt gebruikt om commando’s (volgens het SSC32U protocol) over te sturen tussen de Controller en de Robotarm_description.
sensor_msgs/JointState	Dit message type wordt gebruikt om joints door de sturen.

2. Structuur van de code

Klassendiagram robotarm_description:



De package robotarm_description bestaat uit drie componenten. Het eerste component kan berichten ontvangen en parsen. Het tweede component voert de commando’s uit, waardoor de robotarm naar een andere positie gaat, en het derde component kijkt of er een cup tussen de armen van de gripper zit.

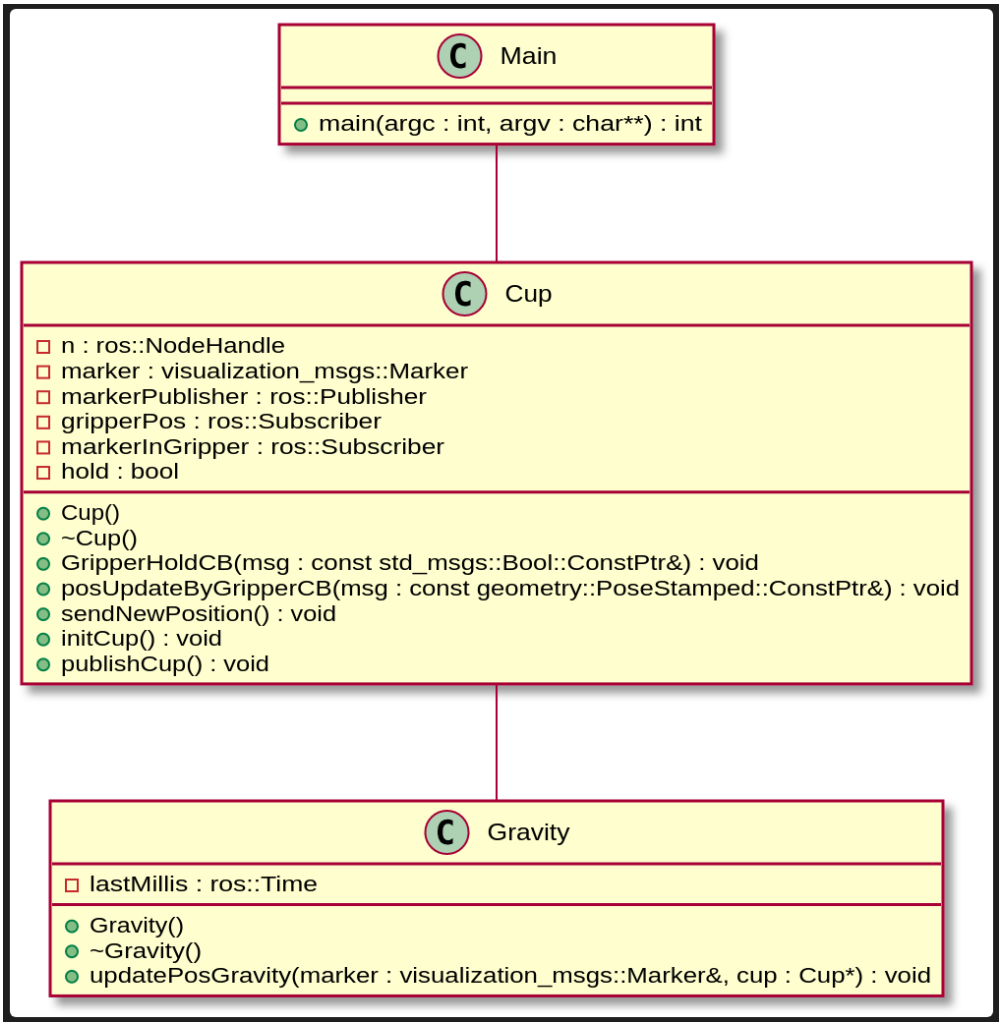
De Parser klasse parst de berichten die binnenkomen vanaf de controller. Deze moeten volgens het juiste protocol zijn (SSC32U). Als dat niet zo is, dan worden de berichten genegeerd. In de andere gevallen wordt het bericht toegevoegd aan een queue. De commando’s in deze queue worden afgehandeld door de MessageHandler.

In de Joints klasse wordt de arm aangestuurd. De gewrichten van de arm worden elk interval geupdated met een waarde die berekend wordt op basis van hoe lang de arm er over moet doen om zijn eindpositie te komen.

Na elke beweging wordt gekeken of de gripper een bekertje tussen zijn armen heeft. Dit gebeurd in de Gripper klasse.

Bij het parsen worden alle berichten in een Message struct gestopt.

Klassendiagram Object:



De Cup klasse bevat enkele functies. De sendNewPosition functie wordt elke keer aangeroepen. In deze functie verstuurd hij zijn huidige Pose. Daarnaast heeft hij twee callback functies die worden uitgevoerd door twee verschillende subscribers. De GripperHoldCB functie wordt uitgevoerd als er een nieuw bericht op het marker_in_gripper topic binnenkomt. Hij kijkt of de marker op dat moment in de gripper zit. Als dat zo is kleurt het bekertje rood, anders groen. Ook kijkt hij wat zijn hoogte is. Als hij ergens in de lucht (de Z waarde is groter dan 0) zit, dan laat hij zich vallen door middel van de zwaartekracht. Daar wordt de Gravity klasse voor gebruikt.

De tweede callback functie (posUpdateByGripperCB) update de positie van de marker als het bekertje op dat moment wordt vastgehouden door de gripper.