
Robotik Projekt

- Aufgabenblatt 2 -

Git

Im Rahmen der Lehrveranstaltung erwarten wir die aktive Nutzung von git zur Versionsverwaltung der von Ihnen erstellten Software.

Teil der praktischen Prüfungsleistung ist ein gepflegtes Git-Repo mit aussagekräftiger Readme im main-Branch. Folgende Punkte werden mindestens erwartet:

- kurze Funktionsbeschreibung der Software
- welche Abhängigkeiten werden zusätzlich zu ROS benötigt
- welche Vorbereitungen zum Ausführen der Software sind notwendig
- wie wird die Software ausgeführt
- welche Parameter gibt es, jeweils mit kurzer Beschreibung und Standardwert
- ein Diagramm oder kurze Beschreibung, wie die einzelnen Komponenten / Nodes zusammenhängen

Es wird erwartet, dass sich die komplette Software für die Prüfungsleistung im main-Branch befindet. Sollte davon abgewichen werden, ist in der Readme im main-Branch entsprechend zu beschreiben, wo die entsprechenden Teile zu finden sind und warum eine andere Struktur gewählt wurde.

ROS Vorbereitung

workspace erstellen

Erstellen Sie einen ROS-workspace. In diesem Ordner sollte Sie im Rahmen der Übung sämtlicher Code befinden, den Sie selbst schreiben (oder wenn Sie fremden Code selbst bauen). Führen Sie dafür vom Tutorial die Schritte 1 und 2 aus:

<https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Creating-A-Workspace/Creating-A-Workspace.html>

Ergänzen Sie die .bashrc ganz am Ende um den source-Befehl (Schritt 6) mit einem absoluten Pfad zum ROS-workspace, z.B. "source /home/ubuntu/ros2_ws/install/local_setup.bash" (wenn der Nutzer ubuntu heißt und der ROS-workspace direkt im home-Verzeichnis erstellt wurde)

package erstellen

Erstellen Sie ein ROS-package mit dem Namen "timing_tubaf_py". Folgen Sie dazu dem Tutorial und wandeln die Schritte entsprechend ab:

<https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Creating-Your-First-ROS2-Package.html>

ROS publisher/subscriber

Schreiben Sie einen publisher in Python, welcher auf dem Topic "number" einen Integer-Wert published, welcher sich nach jedem Senden um 1 erhöht.

Schreiben Sie einen subscriber in Python, welcher auf das Topic "number" subscribed und die zeitliche Differenz zwischen der aktuellen und der letzten Nachricht bildet. Diese Differenz soll auf dem Topic "diff" gepublished werden.

Orientieren Sie sich dazu an den entsprechenden Tutorials:

<https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Writing-A-Simple-Cpp-Publisher-And-Subscriber.html>

Nutzen Sie bei der Entwicklung git. Benennen Sie die einzelnen commits sinnvoll. Kommentieren Sie den Code. Erstellen Sie eine Readme mit entsprechender Doku.