Github repo: <https://github.com/UniversalRobots/Universal_Robots_ROS_Driver>

1. Look into folders ur\_callibration and ur\_dashboard\_msgs
2. Start out at *scripts -> test scripts*

#### Notes

In *Universal\_Robots\_ROS\_Driver*

##### ../ur\_calibration

* From **/README**: ur\_calibration is used to extract the factory calibration to be used in future applications where you intend to move robot using ‘this’ driver
  + Not sure what ‘this driver’ is referring to (from **/CHANGELOG.rst** is seems to be referring to ur\_robot\_driver)
* From **/launch**: had .launch file for downloading the calibration data from a specific robot ip and storing in target file
* From **/src**: contains files used in obtaining calibration data and providing interface for user to input robot ip’s and target files
  + Also provides file calibration\_correction which I guess either handles cases when no user input is provided or makes corrections to calibration data.
* From **/test/calibration\_test.cpp**: tests the calibration and calculates the precision

##### ../ur\_dashboard\_msgs

* From **/action**: SetMode.action is used to set robot mode into a running state and change safety modes to non-critical states. Then it can be used to play a loaded program
* From **/msg**: I think these are either templates for the different messages present or they are messages themselves that can be used whenever
* From **/srv**: Seems to be the same deal, more templates for messages that services will use in their processes.
* **CMakeLists.txt**:seems to create different action, service, topic lists.

##### ..**/ur\_robot\_driver/scripts**

* From **/tool\_communication**: helper script to start tool communication, using robot ip and such
* From **/test\_move**:
  + Imports begin with importing the different message formats to be used in all the processes. Notice certain messages coming from .srv files
    - Also imports the action library ‘actionlib’ (see ROS wiki)
  + Creates lists for Joint names, trajectory controllers, cartesian trajectory controllers, etc.

#### ROS2 python things to look into

* rospy
* actionlib

<http://wiki.ros.org/scaled_joint_trajectory_controller>

PID controls

“**Important note** Even when a goal has been aborted, the controller will still attempt to execute the trajectory as best as possible. “

* Test this

Joint trajectory controllers interfaces (at bottom)

Are they messages, what are they

<http://wiki.ros.org/joint_trajectory_controller/UnderstandingTrajectoryReplacement>