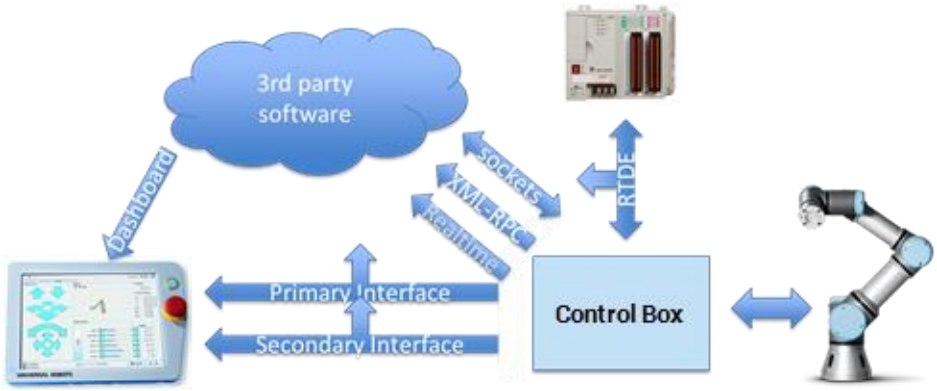
UR Robot Control

Overview:

Connecting to URControl:

URControl is the low-level robot controller running on the Mini-ITX PC in the Control Box. When the PC boots up, the URControl starts up as a daemon (i.e., a service) and the PolyScope connects as a client using a local TCP/IP connection.

UR robot can interact with external devices by different types:



Primary/Secondary Interfaces: The primary interface transmits robot state data and additional messages. The secondary interface transmits robot state data only. The data is mainly used for communication between GUI and controller. Both accept URScript commands with 10 Hz update rate

Real-time Interfaces (also known as the Matlab interface**):** similar with primary/secondary interfaces. The main difference is update rate.

Dashboard Server: Main functions of the server are to load, play, pause, and stop a robot program.

Socket Communication: communicate with outside equipment through TCP/IP protocol, robot acts as client and other device play a role as server.

XML-RPC: the UR controller can call methods/functions (with parameters) on a remote program/server and get back structured data.

RTDE (Real-Time Data Exchange): replacement for the real-time interface.

More details about the ports and how to remotely control the UR robot are shown in the How-to guide provided by Universal robot, it can be found in that link:

https://www.universal-robots.com/how-tos-and-faqs/how-to/ur-how-tos/overview-of-client-interfaces-21744/

Control Universal Robot through TCP/IP

UR controller provides servers to send robot state data and receive URScript commands

Programming a robot at the Script Level is done by writing a client application (running at another PC) and connecting to URControl using a TCP/IP socket.

Ethernet TCP/IP socket communication via URscript is demonstrated in that like

https://www.universal-robots.com/how-tos-and-faqs/how-to/ur-how-tos/ethernet-socket-communication-via-urscript-15678/

with example and code source available to download

When a connection has been established URScript programs or commands are sent in clear text on the socket. Also another tutorial to illustrate that is shown in that link:

http://www.zacobria.com/universal-robots-zacobria-forum-hints-tips-how-to/script-via-socket-connection/

Also, there is a Python library to help control UR,

https://github.com/SintefManufacturing/python-urx

How to use the XML-RPC port to call methods and functions from third party remotely is explained in that link,

https://www.universal-robots.com/how-tos-and-faqs/how-to/ur-how-tos/xml-rpc-communication-16326/

one example using C++ and one using python and the code source are available to download in the link.

ROS Driver :

The new driver ROS release for universal robot was forked from the ur_modern_driver, the goal of new driver is to provide a stable and sustainable interface between UR robots and ROS. For using that driver with a remote PC we need the URcap external control to enable external control of the robot, this URcap must be installed on the UR robot and the ur_rtde_driver must be installed on the remote PC with real-time kernel.

The minimal PolyScope versions are 3.7 CB3 and 5.1 e-series

Details about how to setup the remote PC and run that driver can be found in that link

https://github.com/UniversalRobots/Universal_Robots_ROS_Driver

MATLAB Toolbox for UR Manipulators:

Developed by Michael Kutzer at United States Naval Academy. This toolbox interfaces Universal Robots with MATLAB. The toolbox allows users to interface manipulators leveraging the **URX Python library**. This toolbox also includes visualization tools,

forward and inverse kinematics functions, Jacobians, and a URScript control approach allowing users to smoothly command UR manipulators along desired trajectories.

The Toolbox with how to install:

https://www.usna.edu/Users/weaprcon/kutzer/_Code-Development/UR_Toolbox.php

Also Matlab API interface with examples showing how to control the robot can be found in that repo:

https://github.com/ZakariaChekakta/Universal-Robots-Matlab-interface