Submission for Robothon 2022

Team RoboPig

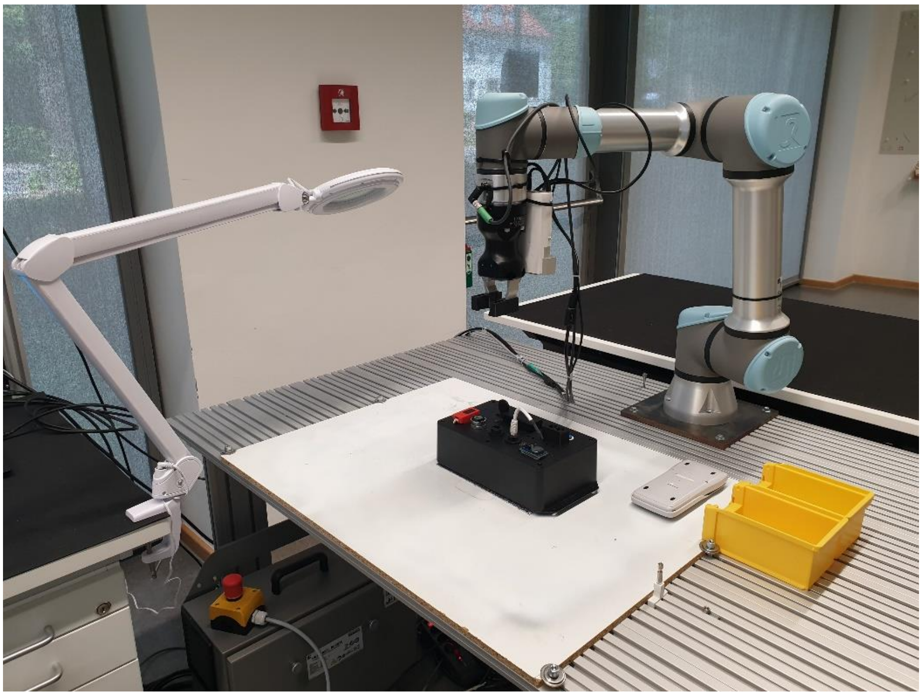
29 May 2022

Executive Summary

This document is a required submission for the Robothon 2022 competition by team RoboPig. We describe the robot platform and all associated software and hardware tools used in the competition, as well as a quick start guide for our setup.

Robot Platform

Our team is using the system shown below consisting of an off-the-shelf UR5e robot with a Robotiq Hand-E gripper, an Azure Kinect camera mounted to the gripper, and an external lighting source. The robot platform itself was not modified. We 3D-printed custom jaws for the Hand-E gripper.

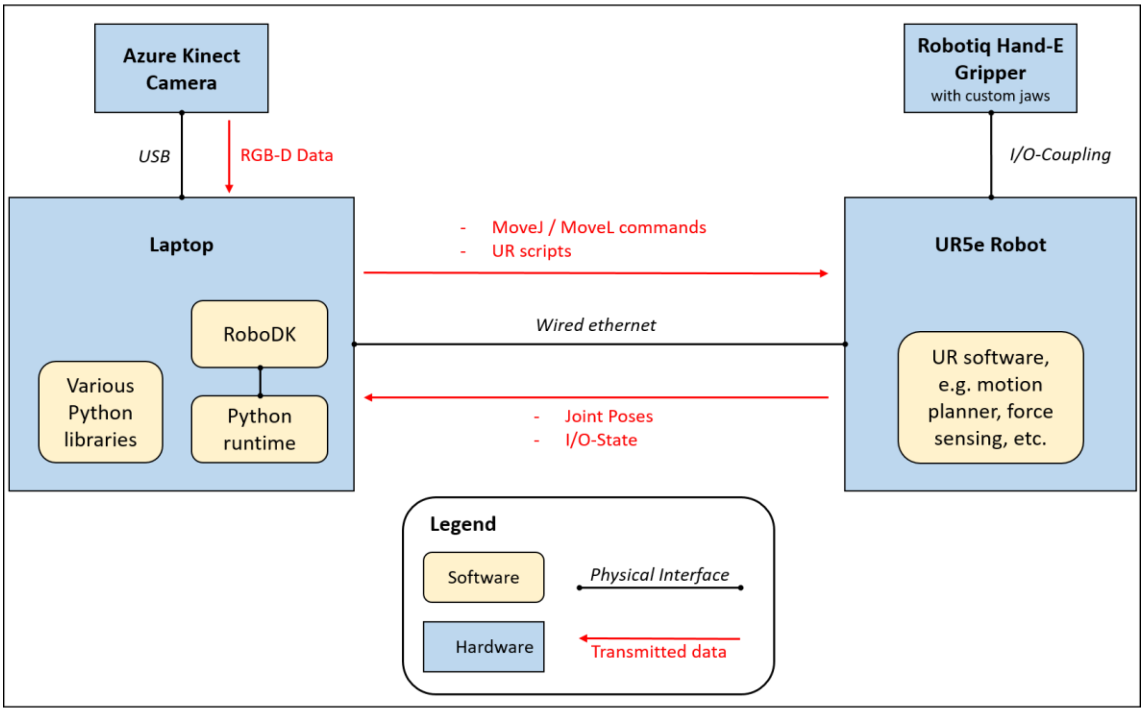


Equipment Used List

Picture	Equipment	Function
	UR 5e Cobot	Utilisation of built-in safety functions, motion planning and tactile/force sensing. UR scripts can be developed using the teach panel and called by Python from the laptop.
	Robotiq Hand E Gripper	Connected to the UR5e via IO-Cuppling. Pick all sorts of objects in the tasks.
	3D-printed Gripper Jaws	Custom jaws for picking the batteries and gripping the bit-tools.
	Lenovo Think Pad E14 Gen 3 Laptop	Base Station for Python runtime and the RoboDK programming environment.
	Azure Kinect Camera	Sends colour and depth data to the laptop via USB. Data is used to localise the task board and for object detection in the transfer task.
	Bit-tool with bit-tool-holder	Additional tools to manipulate different battery cases in the transferability task.
	Boxes	For sorting different types of batteries in the transferability task.
	Additional lights	For constant light settings

Hardware & Software Dependency

The figure below shows the hardware and software architecture used by our team.



Each component is further described below (anticlockwise from top left):

- Azure Kinect Camera
  - Sends colour and depth data to the laptop via USB.
- Laptop
  - Acts as a base station running a Python runtime and the RoboDK programming environment
  - RoboDK fulfils the following functions:
    - Simulation environment for fast and agile development and debugging, for both robot movement and image processing.
    - Setting up targets relative to a fixed coordinate system located on the task board (task board frame).
    - Real-time visualisation of the robot motion and control of the robot.
  - Python:
    - The following libraries are used: OpenCV, pyK4a, ur_rtde, RoboDK, matplotlib and numpy.
    - The task board frame relative to the robot base frame is computed using OpenCV tools. The result is sent to RoboDK, which contains targets relative to the task board frame (see above).
    - The calculator and the battery in the transferability task is localised with the OpenCV library.
  - Communication with the UR5e
    - A socket connection to the UR5e via wired ethernet is set up
    - Laptop -> UR5e
      - MoveJ (targets in joint space) and MoveL (targets in Cartesian space)
      - UR scripts, e.g. an “advanced spiral search”
    - UR5e -> laptop

        - Joint poses
        - I/O States
- UR5e Robot and Control Box
    - Utilisation of built-in safety functions, motion planning, and tactile/force sensing
    - UR scripts can be developed using the teach panel and called by Python from the laptop
- Robotiq Hand-E Gripper
    - Connected to the UR5e via I/O-Coupling
    - 3D-printed custom jaws, e.g. to pick and center the batteries, or grip the additional tools

Quick Start Guide

1. Start RoboDK.
    - The software will load the target positions for the robot and setup the socket communication.
    - Setup robot communication.
2. Start Python Script for the task board task.
3. Choose the order of the tasks by inserting the task numbers in a desired order.
4. Start the Script.
    - Rough detection of the board using camera
        - Image processing libraries detect the task board in the image and extract corner points in image coordinates.
        - Calculates the transformation matrices from the actual TCP pose and hand eye calibration.
        - Gets a camera to task board transformation with the OpenCV perspective-n-point (pnp) pose algorithm.
        - Calculates complete transformation matrix from submatrices.
        - Transforms task board coordinate system to global base coordinate system.
    - Refine localisation by touching the task board on three points.
    - Robot will start the trial by pushing the blue button.
    - Tasks are executed in the desired order.
    - Robot will stop the trial by pushing the red button.