Libraries Guide

Content Guide

V1.0 – 12th February 2025

Quanser Consulting Inc.   info@quanser.com
119 Spy Court   Phone : 19059403575
Markham, Ontario   Fax   : 19059403576
L3R 5H6, Canada   printed in Markham, Ontario.

Background

To support a wide range of application development, Quanser has developed a series of libraries designed to help you accelerate your academic goals.

A visual representation of the libraries folder structure:
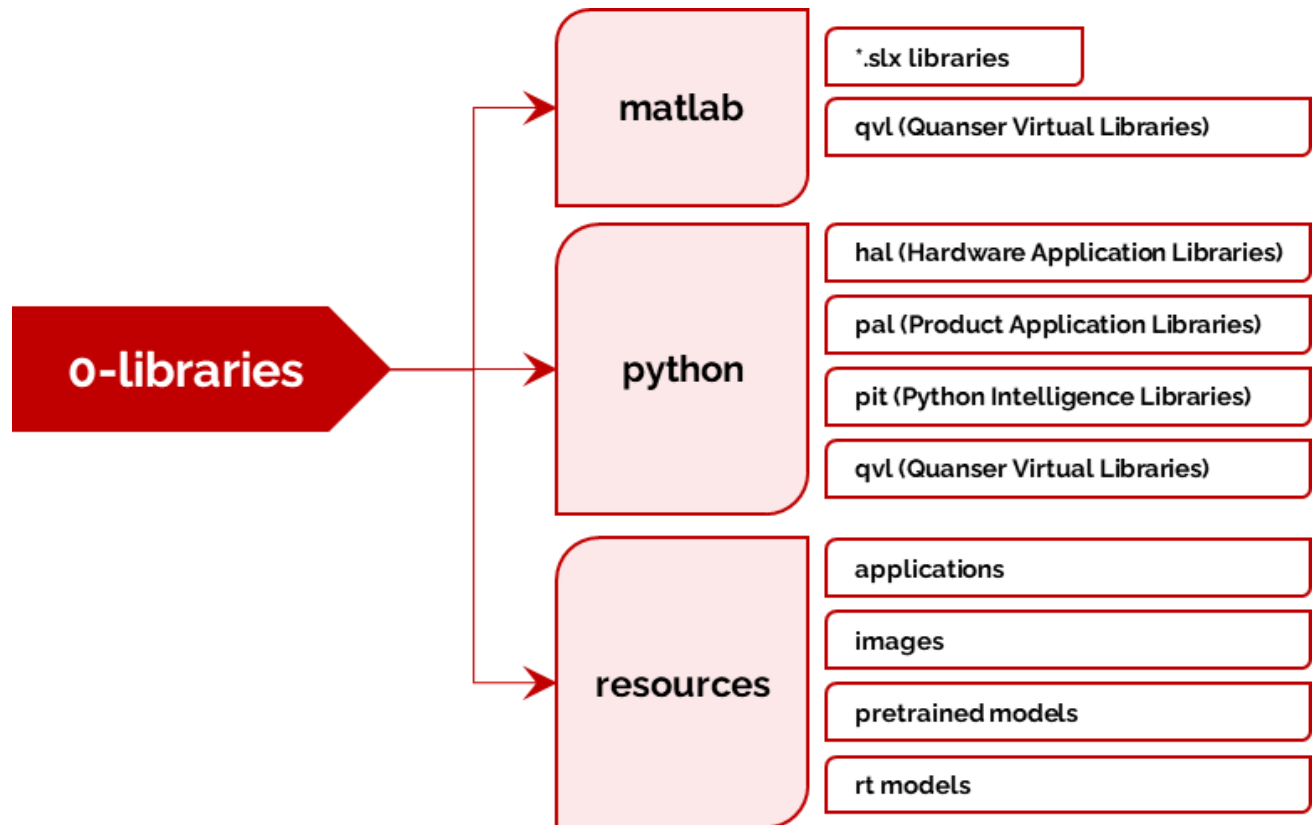
matlab

To support continual development using Quanser devices, the matlab library was generated to help you get started. Along side the series of auxiliary .slx libraries, Quanser developed a qvl (Quanser Virtual Libraries) folder to allow development using Quanser Interactive Labs.

Documentation on qvl libraries is found in
https://qlabs.quanserdocs.com/en/latest/index.html

python

Quanser has developed a series of custom python libraries designed to be integrated into different levels of your teaching and research applications.

hal (Hardware Application Libraries)

hal was designed to give you an insight into how applications can leverage a common python library to build upon your current work. Visually the subdirectories available within the hal library are:

python/hal

content

products

utilities
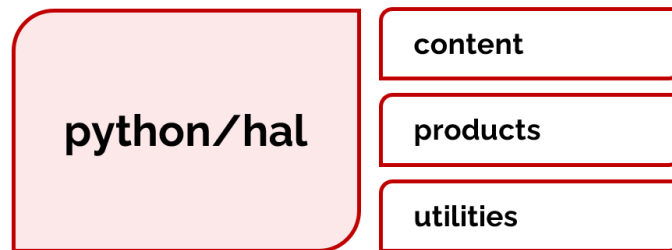
content

For python based curriculum, the content subdirectory is used for students to complete a series of classes/methods used within a skills progression pipeline.

products

Curriculum or custom applications may benefit from a hardware application centric library. In this directory you will notice a series of product centric applications which could be hand developed by students but are designed by Quanser to facilitate the creation and testing of product centric applications.

utilities

Lastly the utilities directory is meant to for device agnostic application examples. In this directory you will see different implementations of control, estimation, image processing. You may choose to use Quanser's implementation of some of these commonly explored algorithms or compare it with your own implementation.

pal (Product Application Libraries)

pal was designed to be the core streamlined experience for getting start with Quanser devices and supported third-party devices. Visually the directory structure of pal is:
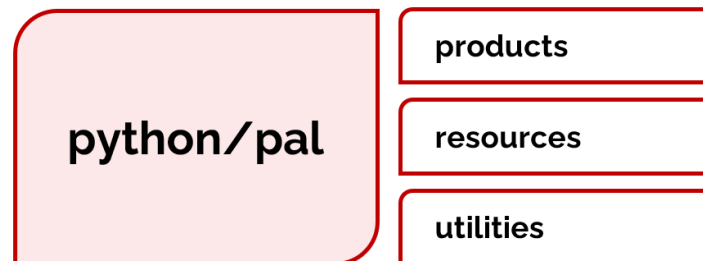
Note: pal was not designed to be modified by students. The core components of pal are designed to be imported into a generic python application and give students streamlined access to devices (hardware or virtual).

products

The products subdirectory holds an easy to use, preconfigured set of classes for a series of Quanser products. They build upon the core Quanser SDK to give you performant access to core Quanser devices.

Note: Please do not modify this directory unless a backup is created. Changes to files under pal/products can cause example applications to work incorrectly.

List of supported products:
- Aero2
- QArm
- QArm Mini
- QBot
- QBot Platform
- QCar 1 and 2
- Qube-Servo 2 and 3
- Traffic Light

resources

An auxiliary library meant to hold any images used in Quanser skills activities. This directory also enables you to launch pre-compiled .rt executables for applications which connect to virtual products.

utilities

A series of software utilities mean to expand the level of information you can interact with when developing a custom python application. Utilities can vary from generic applications such as stream (which enable network communication using Quanser's optimized Stream libraries) to keyboard/gamepad which enable access to a host computer's keyboard for custom applications. The complete list of utilities available:
- gamepad
- keyboard
- lidar
- math
- probe
- scope
- stream
- transfer learning
- vision

pit (python intelligence library)

pit was developed to enhance and streamline applications focused on bridging machine learning applications with actual hardware. Within the pit library Quanser has developed enhance support for different types of Machine Learning network architectures. Visually the folder structure contains:

jetson

Designed to interface with Nvidia devices the jetson class streamlines access to common machine learning architecture optimized for jetson based devices.

LaneNet

Custom for Self-Driving applications, Quanser enabled streamlined access to LanetNet. Useful for research and teaching where the focus is on different methods of vision based lane detection frameworks.

YOLO

One of the most common visual Neural Networks, YOLO provides a fast and retrainable way to classify and detect objects in a scene.

qvl (quanser virtual libraries)

Similar in architecture to the MATLAB implementation. qvl enables you to obtain direct access to modifying, spawning and reconfiguring applications which make use of Quanser Interactive Labs. For a detailed explanation of the python implementation of qvl please refer to the following link: https://qlabs.quanserdocs.com/en/latest/index.html

Resources

This directory acts as a storage point for common resources used across the various MATLAB and python libraries.

Applications

Within the applications subdirectory you will find the source file for compiled. rt executables used across various locations in Quanser's content. Visually the folder structure looks like:
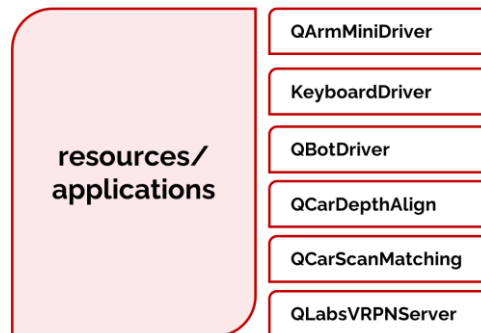
Images

This folder contains images used across QCar curriculum and Simulink examples.

Pretrained Models

Used by the python intelligence libraries (pit), when running examples that need a model downloaded, they get stored in this directory once the application is ran for the first time.

Rt_Models

Specifically developed for Quanser Interactive Labs. This directory contains the compiled .rt models for Quanser products which support multiagent deployment. Visually the list of supported rt models include: