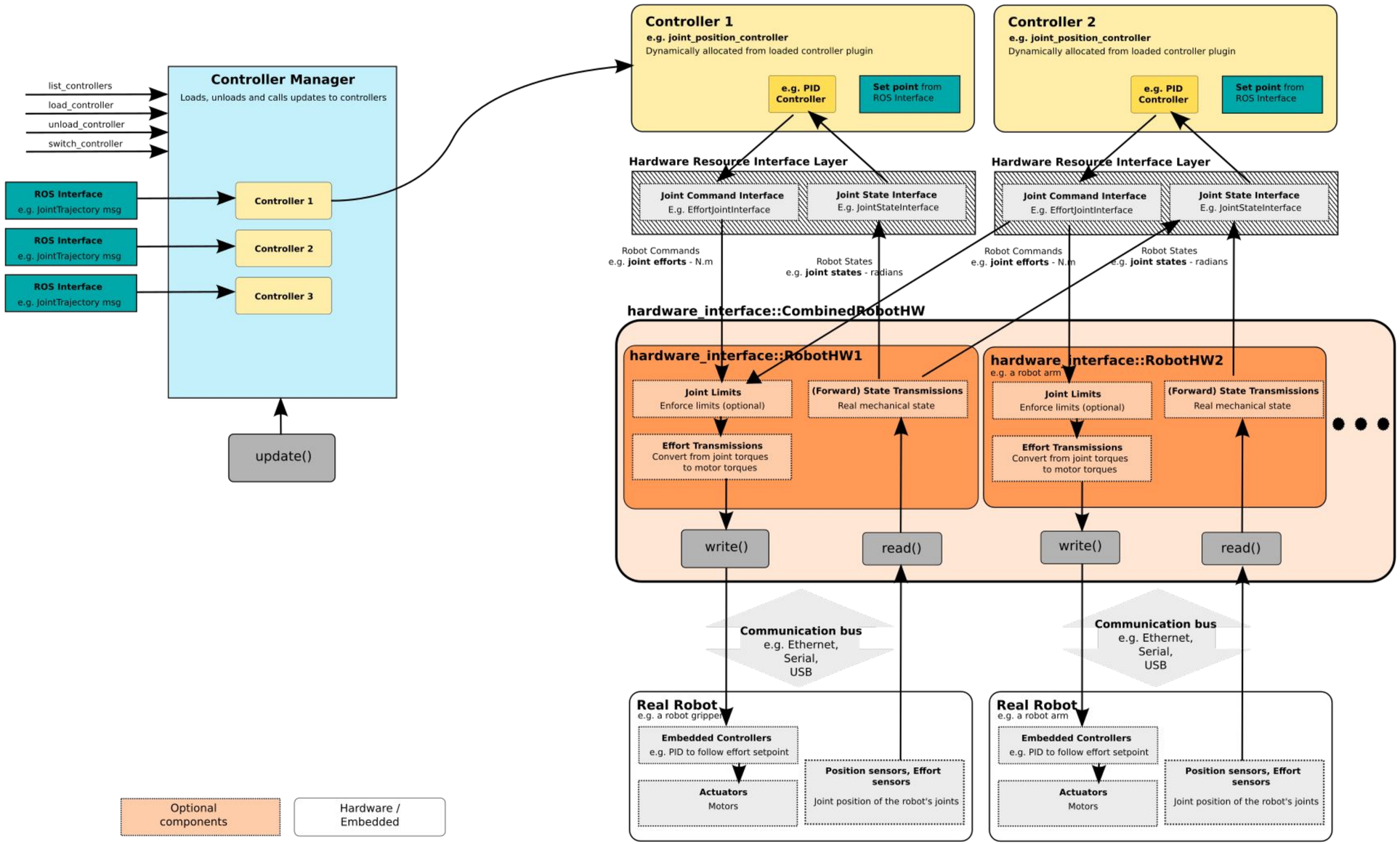
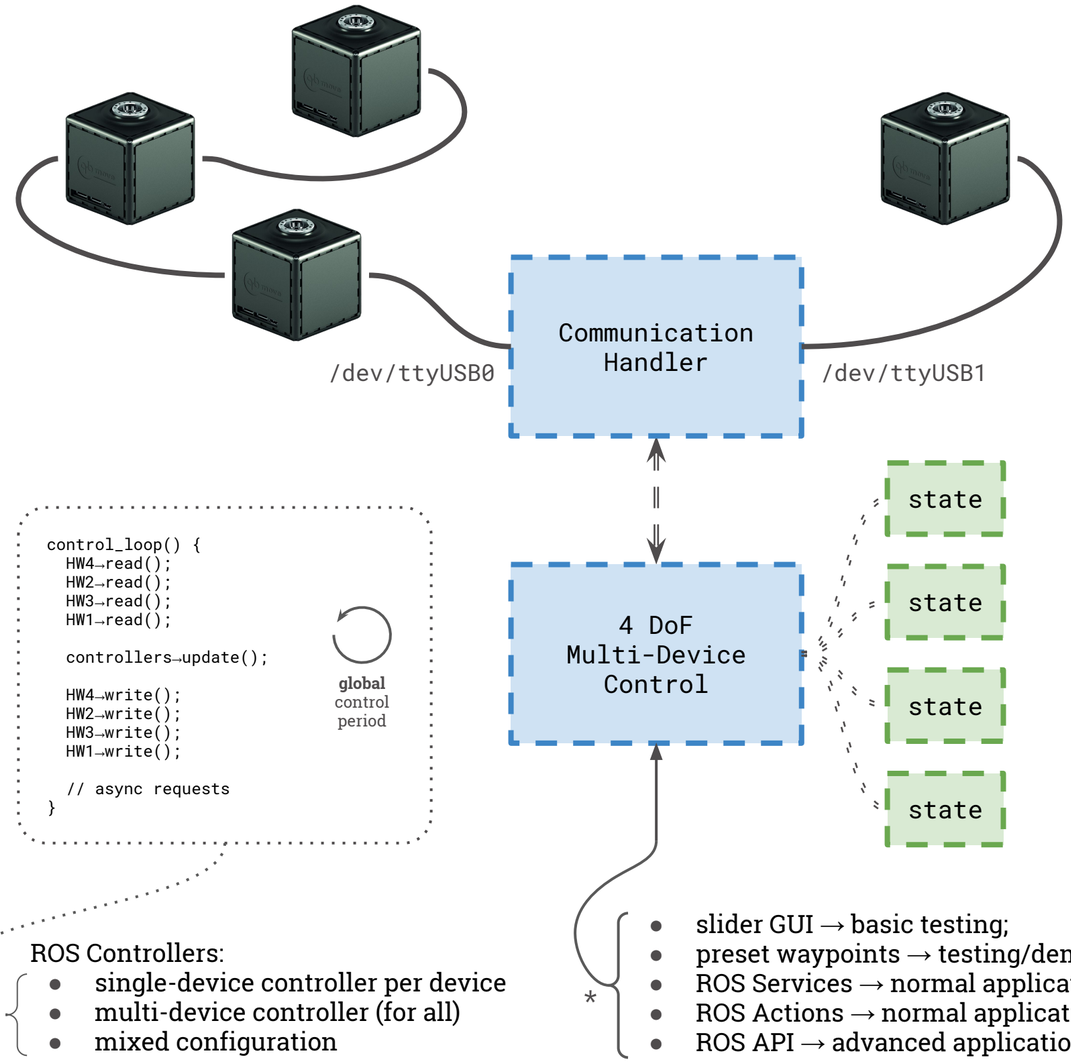
ROS packages

recap: ros_control



one synchronous control node



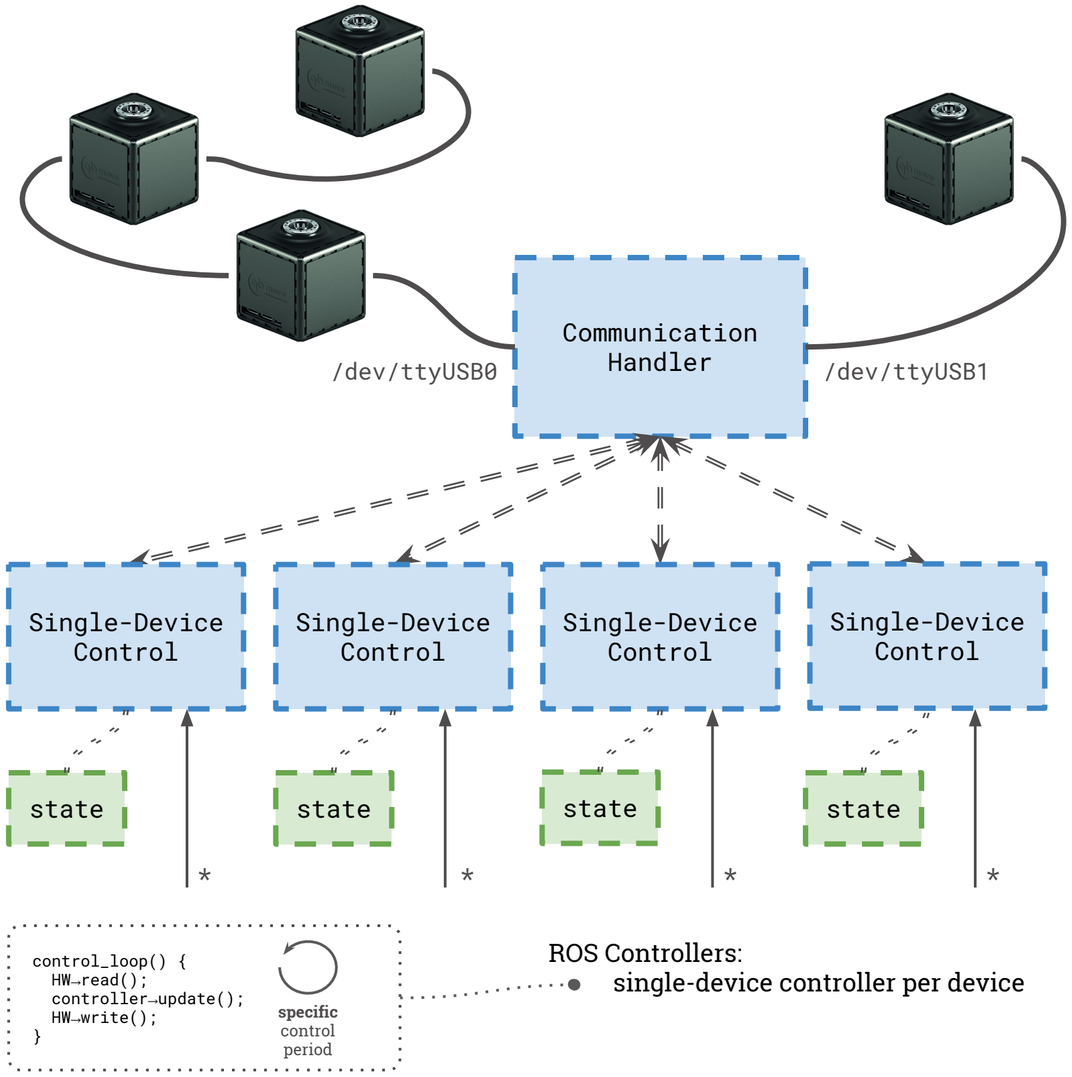
ROS Controllers:

- single-device controller per device
- multi-device controller (for all)
- mixed configuration

*

- slider GUI → basic testing;
- preset waypoints → testing/demo;
- ROS Services → normal application;
- ROS Actions → normal application;
- ROS API → advanced application;

asynchronous control nodes



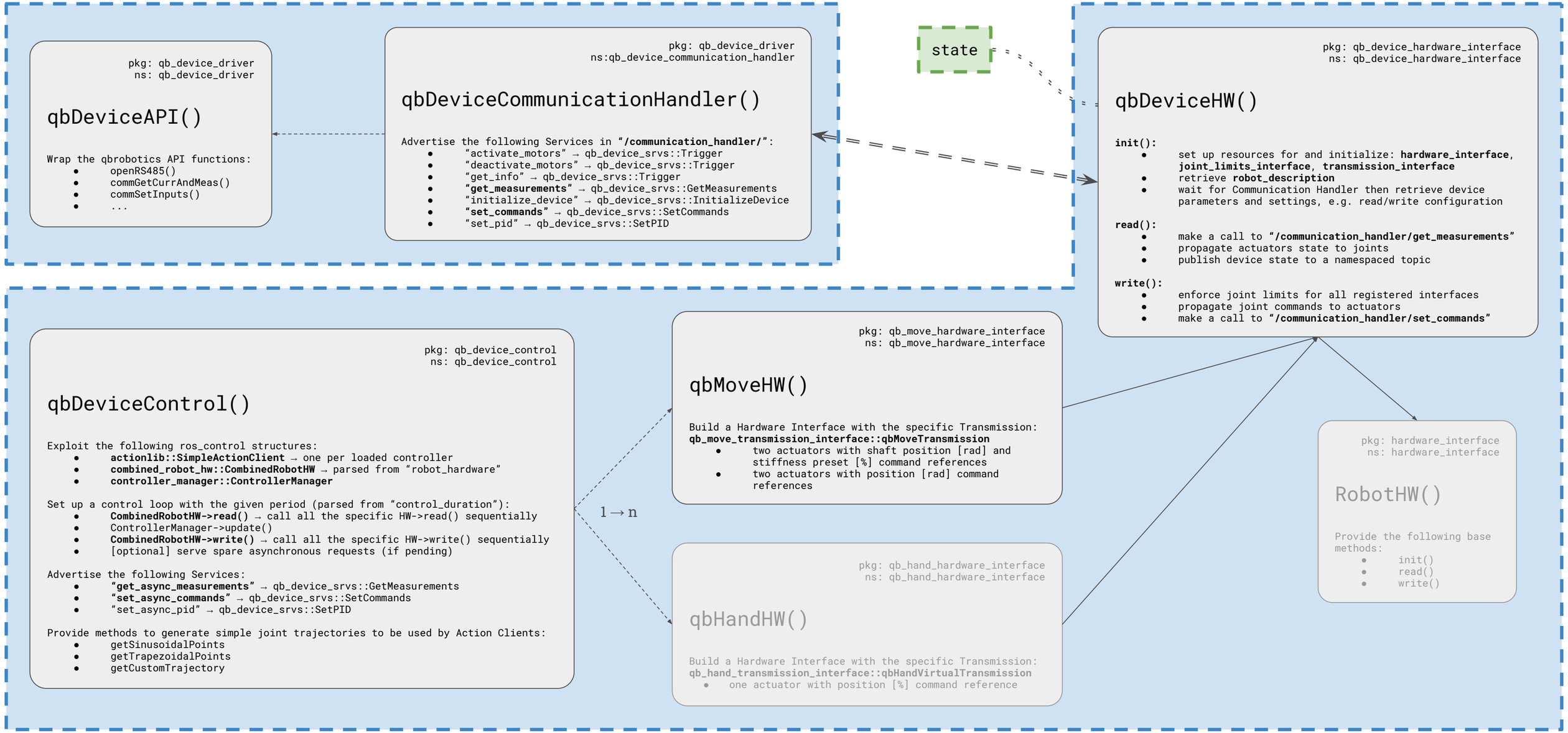
ROS Controllers:

- single-device controller per device

qb robotics®

ROS Topic | ROS Node | wiring | user inputs | ROS Services | ROS Messages

packages overview



C++ Class · ROS Topic · ROS Node · exploit · extend · ROS Services · ROS Messages

multi-device synchronous setup

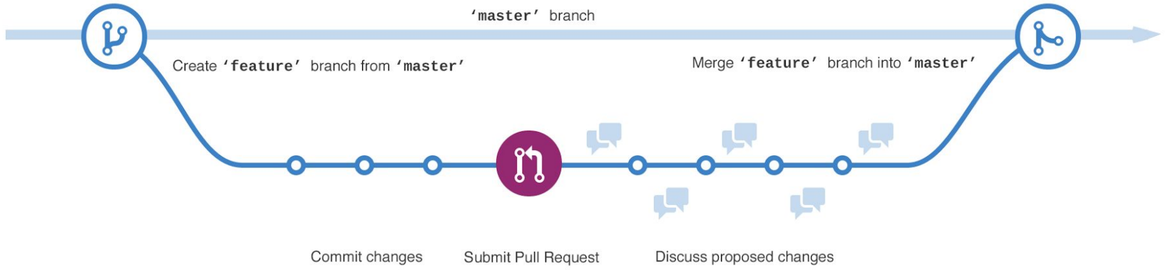
```
<launch>
  <!-- robot settings -->
  <arg name="control_duration" default="0.01" doc="The duration of the control loop [s]."/>
  <arg name="robot_hardware" default="[cube1, cube2, cube3]" doc="The robot hardware interface namespaces, e.g. [device1, device2, ...]."/>
  <arg name="robot_name" default="3dof_chain" doc="The unique robot namespace."/>
  <arg name="robot_namespace" default="$(arg robot_name)" doc="The unique robot namespace."/>
  <arg name="robot_package" default="qb_chain" doc="The base package name prefix for the robot configurations [urdf, rviz, ...]."/>
  <arg name="source_list" default="[control/joint_states]" doc="The joint_states source list for the joint_state_publisher."/>
  <!-- read/write settings -->
  <arg name="get_currents" default="true" doc="Choose whether or not to retrieve current measurements from the device."/>
  <arg name="get_positions" default="true" doc="Choose whether or not to retrieve position measurements from the device."/>
  <arg name="get_distinct_packages" default="false" doc="Choose whether or not to retrieve current and position measurements from the device in two distinct packages."/>
  <arg name="max_repeats" default="3" doc="The maximum number of consecutive repetitions to mark retrieved data as corrupted."/>
  <arg name="set_commands" default="true" doc="Choose whether or not to send command positions to the device."/>
  <arg name="set_commands_async" default="false" doc="Choose whether or not to send commands without waiting for ack."/>
  <!-- initialization settings -->
  <arg name="activate_on_initialization" default="false" doc="Choose whether or not to activate the motors on node startup."/>
  <arg name="rescan_on_initialization" default="false" doc="Choose whether or not to rescan the serial ports on node startup."/>
  <!-- launch settings -->
  <arg name="standalone" default="false" doc="Choose whether or not to start the Communication Handler."/>
  <arg name="use_controller_gui" default="false" doc="Choose whether or not to use the controller GUI."/>
  <arg name="use_rviz" default="true" doc="Choose whether or not to use rviz."/>
  <arg name="use_waypoints" default="false" doc="Choose whether or not to use the waypoint references."/>


  <include file="$(find qb_device_driver)/launch/communication_handler.launch" if="$(arg standalone)"/>

  <include file="$(find qb_device_bringup)/launch/device_bringup.launch" pass_all_args="true">
    <arg name="device_id" value="1"/>
    <arg name="device_name" value="cube1"/>
    <arg name="device_type" value="qbmove"/>
  </include>
  <include file="$(find qb_device_bringup)/launch/device_bringup.launch" pass_all_args="true">
    <arg name="device_id" value="2"/>
    <arg name="device_name" value="cube2"/>
    <arg name="device_type" value="qbmove"/>
  </include>
  <include file="$(find qb_device_bringup)/launch/device_bringup.launch" pass_all_args="true">
    <arg name="device_id" value="3"/>
    <arg name="device_name" value="cube3"/>
    <arg name="device_type" value="qbmove"/>
    <!-- override global settings -->
    <arg name="get_currents" value="false"/>
    <arg name="get_positions" value="true"/>
    <arg name="max_repeats" value="1"/>
    <arg name="set_commands" value="false"/>
  </include>

  <include file="$(find qb_device_bringup)/launch/robot_bringup.launch" pass_all_args="true"/>
</launch>
```

single-device setup

```
<launch>
  <!-- device info -->
  <arg name="device_id" default="1" doc="The ID of the device [1, 128]."/>
  <arg name="device_type" value="qbmove" doc="The type of the device [qbhand, qbmove, ...]."/>
  <arg name="device_name" default="$(arg device_type)$(arg device_id)" doc="The unique device name used in the yaml controller configurations (also in the urdf if not already specified there)."/>
  <!-- robot settings -->
  <arg name="control_duration" default="0.01" doc="The duration of the control loop [s]."/>
  <arg name="robot_hardware" default="[$(arg device_name)]" doc="The robot hardware interface namespaces, e.g. [device1, device2, ...]."/>
  <arg name="robot_name" default="$(arg device_type)" doc="The unique robot namespace."/>
  <arg name="robot_namespace" default="$(arg device_name)" doc="The unique robot namespace."/>
  <arg name="robot_package" default="qb_move" doc="The base package name prefix for the robot configurations [urdf, rviz, ...]."/>
  <arg name="source_list" default="[control/joint_states]" doc="The joint_states source list for the joint_state_publisher."/>
  <!-- read/write settings -->
  <arg name="get_currents" default="true" doc="Choose whether or not to retrieve current measurements from the device."/>
  <arg name="get_positions" default="true" doc="Choose whether or not to retrieve position measurements from the device."/>
  <arg name="get_distinct_packages" default="false" doc="Choose whether or not to retrieve current and position measurements from the device in two distinct packages."/>
  <arg name="max_repeats" default="3" doc="The maximum number of consecutive repetitions to mark retrieved data as corrupted."/>
  <arg name="set_commands" default="true" doc="Choose whether or not to send command positions to the device."/>
  <arg name="set_commands_async" default="false" doc="Choose whether or not to send commands without waiting for ack."/>
  <!-- initialization settings -->
  <arg name="activate_on_initialization" default="false" doc="Choose whether or not to activate the motors on node startup."/>
  <arg name="rescan_on_initialization" default="false" doc="Choose whether or not to rescan the serial ports on node startup."/>
  <!-- launch settings -->
  <arg name="standalone" default="false" doc="Choose whether or not to start the Communication Handler."/>
  <arg name="use_controller_gui" default="false" doc="Choose whether or not to use the controller GUI."/>
  <arg name="use_rviz" default="true" doc="Choose whether or not to use rviz."/>
  <arg name="use_waypoints" default="false" doc="Choose whether or not to use the waypoint references."/>


  <include file="$(find qb_device_driver)/launch/communication_handler.launch" if="$(arg standalone)"/>

  <include file="$(find qb_device_bringup)/launch/device_bringup.launch" pass_all_args="true"/>

  <include file="$(find qb_device_bringup)/launch/robot_bringup.launch" pass_all_args="true"/>
</launch>
```

The use of `combined_robot_hw::`**`CombinedRobotHW`** requires the ROS Parameter **`robot_hardware`** to be set and filled with all the device name list.

feature request



Useful links:

- Making a Pull Request
- 5 elements of a perfect pull request
- A successful Git branching model

bug report / need help?