



# IROS 2012

## Handling ROS tutorial

*Vilamoura, October 7th 2012*

<http://mrl.isr.uc.pt/events/iros2012tutorial/>

**Shadow Robot Company**

**Control of the Shadow Dexterous Hand using ROS**

Toni Oliver - [toni@shadowrobot.com](mailto:toni@shadowrobot.com)



# Summary

Explain the ROS based control architecture and demonstrate it controlling a real Shadow hand



 ROS



# Outline

Hardware architecture

Host side controller architecture

Control demo

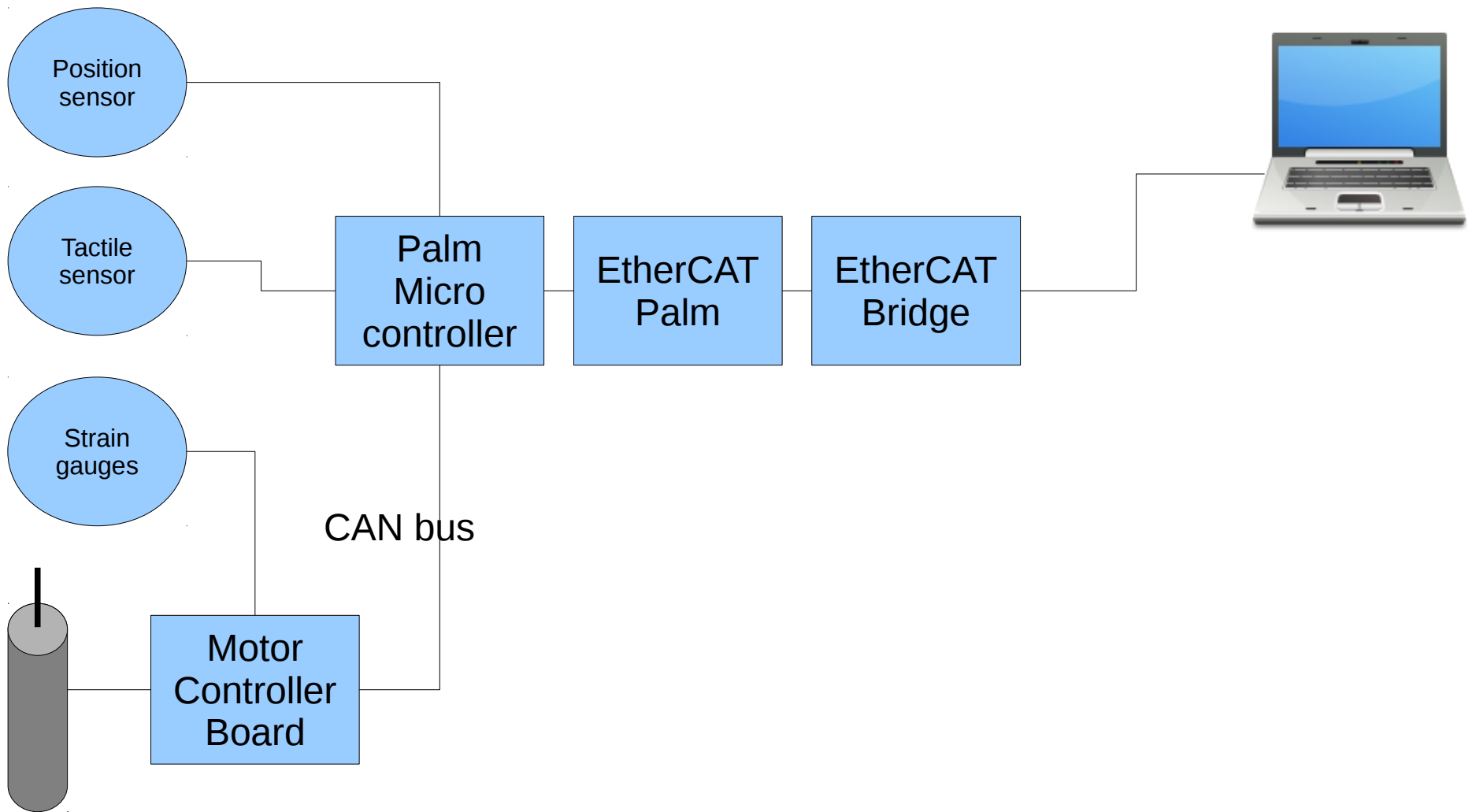


# Hardware

- 24 joints
- 20 actuators
- Sensors:
  - Hall effect position sensor in every joint
  - Tactile sensor in every fingertip
  - 2 strain gauges per motor (measured on tendon)
  - Motor temperature
  - Motor supply voltage
  - Motor current

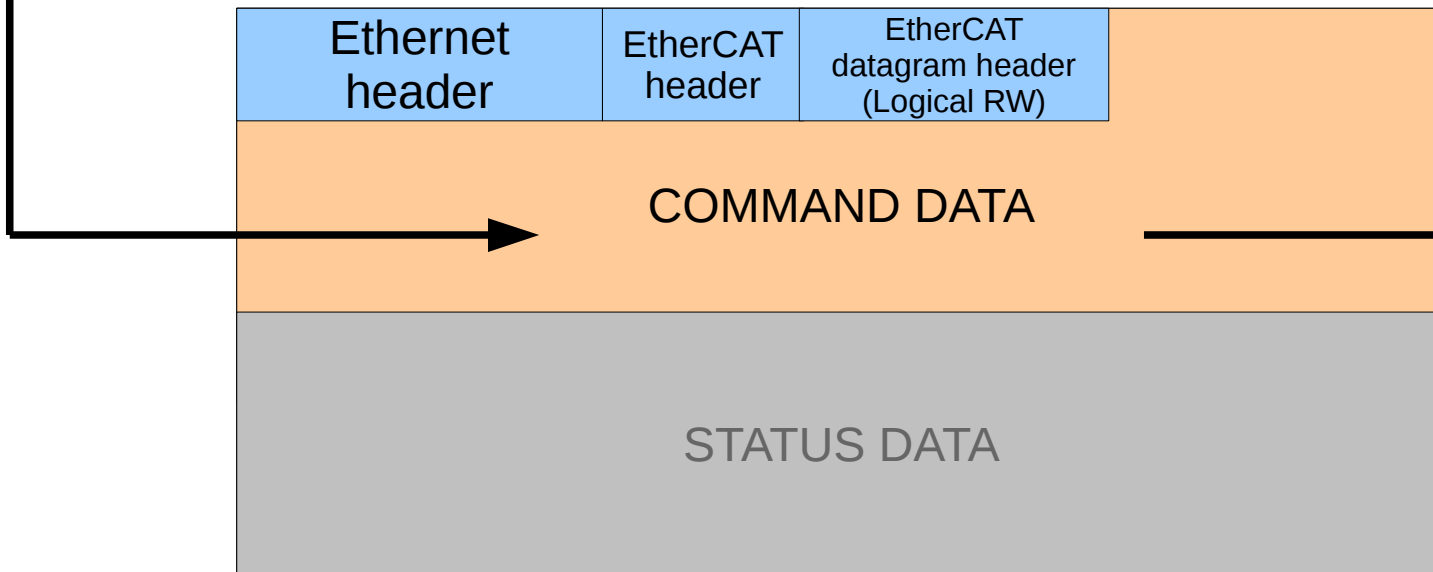


# Hardware



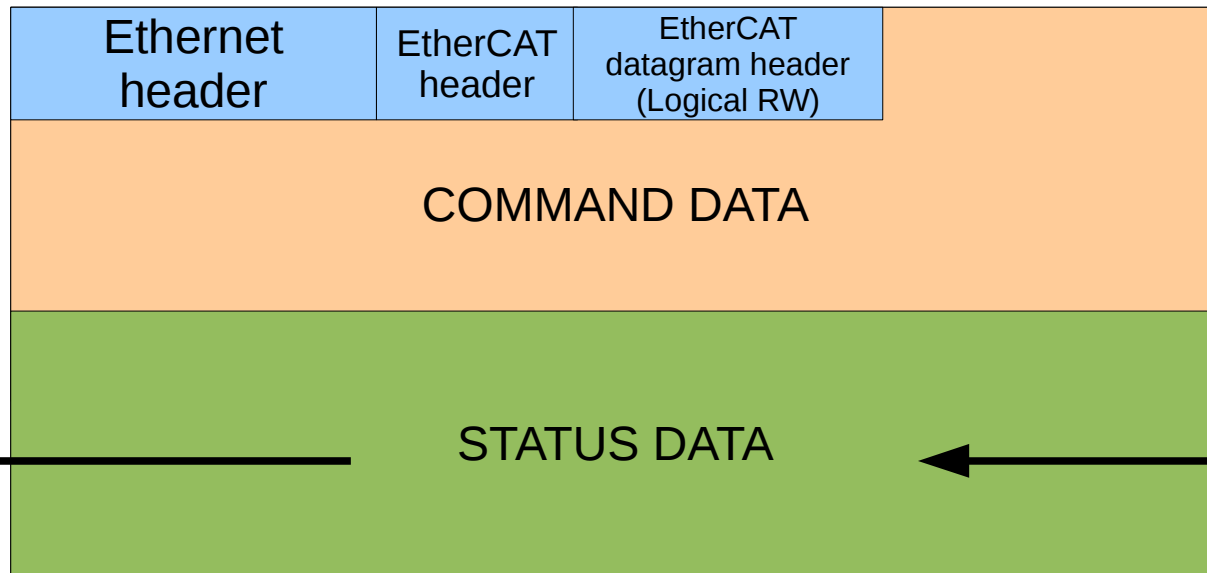


# EtherCAT frame



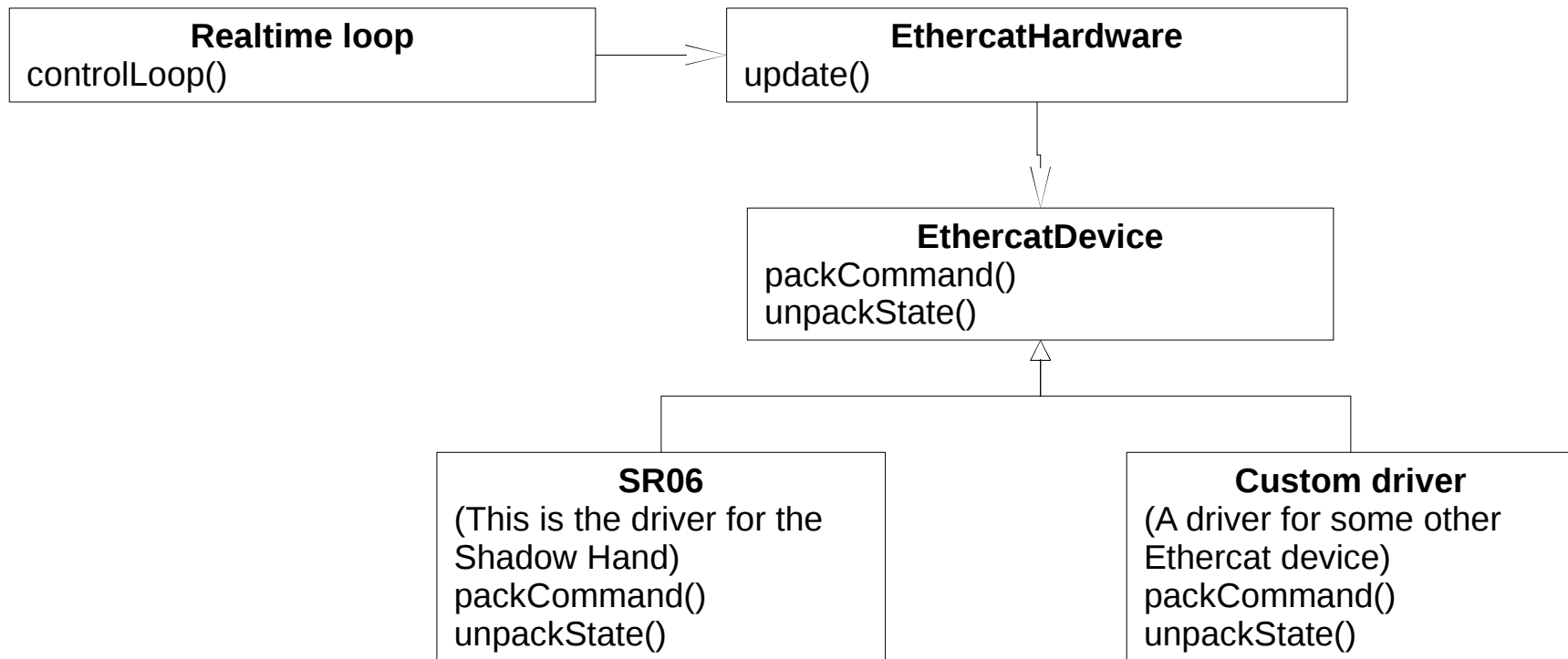


# EtherCAT frame





# Hardware drivers







# Hardware drivers

- Inherit from EthercatDevice:
  - packCommand: send data for this device to the etherCAT bus
  - unpackState: get the data coming from this device
- Drivers are loaded as plugins by EthercatHardware:
  - `PLUGINLIB_REGISTER_CLASS(6, SR06, EthercatDevice);`
  - Xml declaration of the plugin:

```
<library path="lib/libsr_edc_ethercat_drivers" >  
  <class name="6" type="SR06" base_class_type="EthercatDevice">  
    <description>  
      Shadow C6M2 etherCAT dual CAN motor Hand driver  
    </description>  
  </class>  
</library>
```
- Class name must match the Product ID of the ethercat device we want to control



# Outline

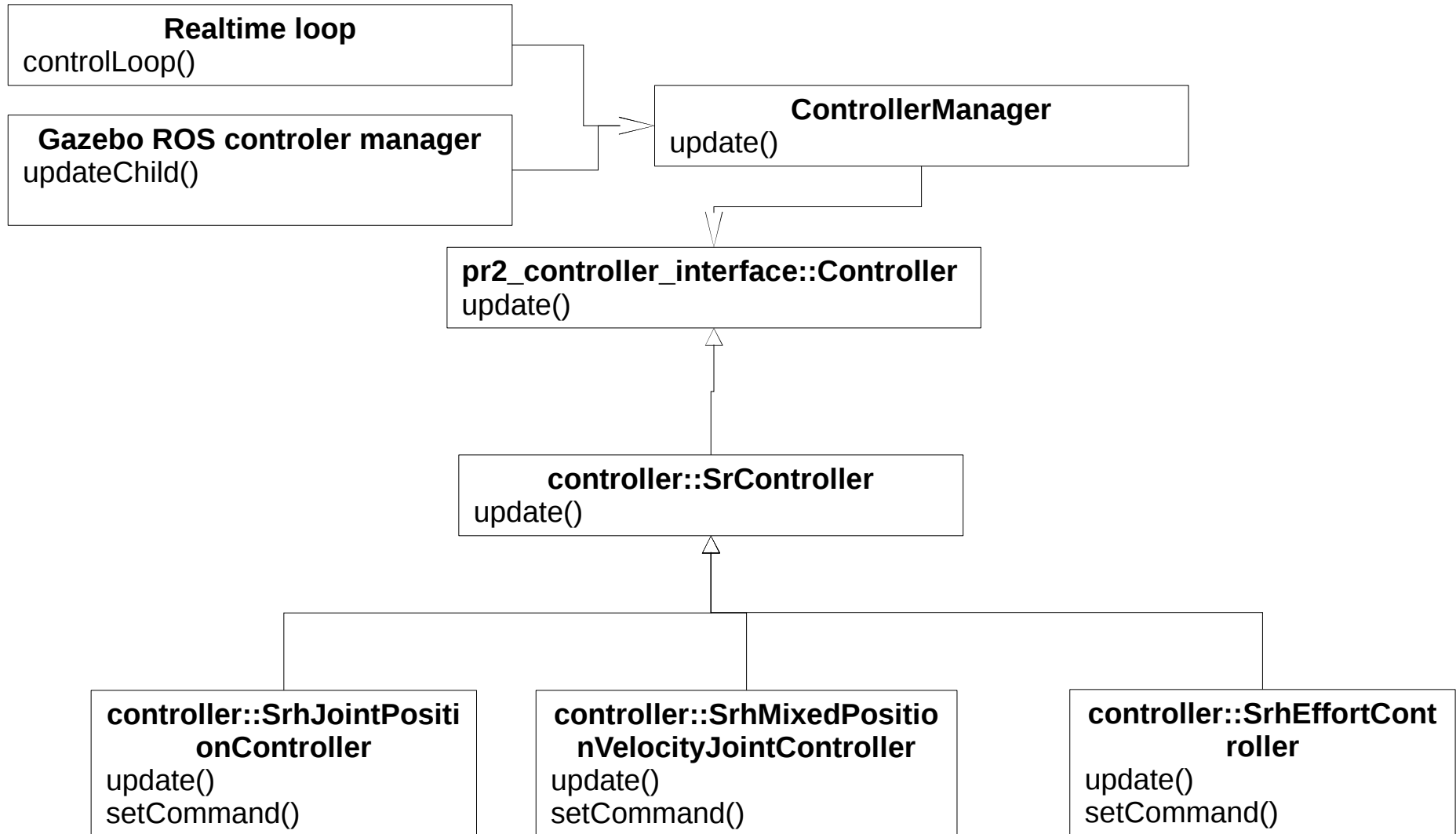
Hardware architecture

Host side controller architecture

Control demo



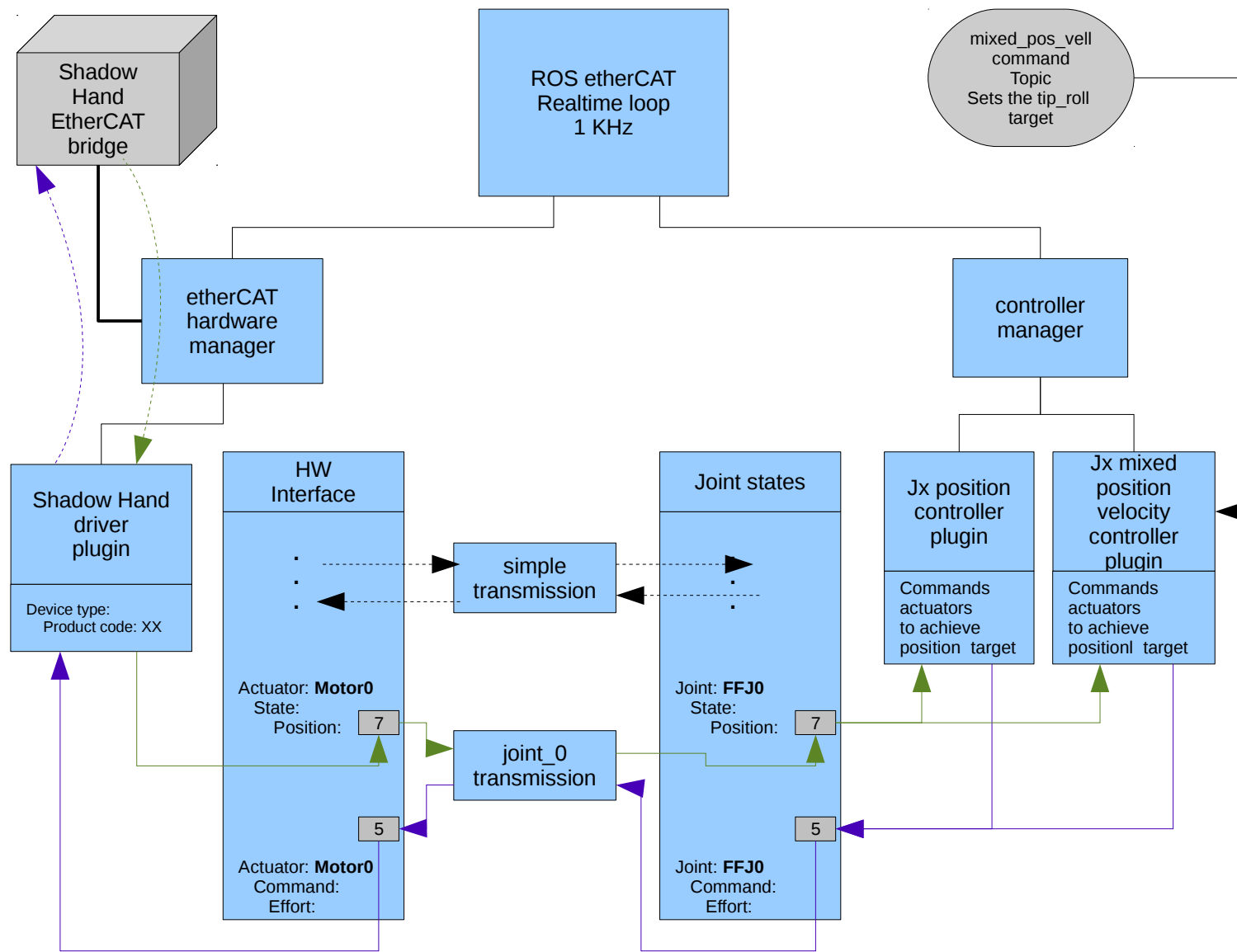
# Controllers





# Controllers

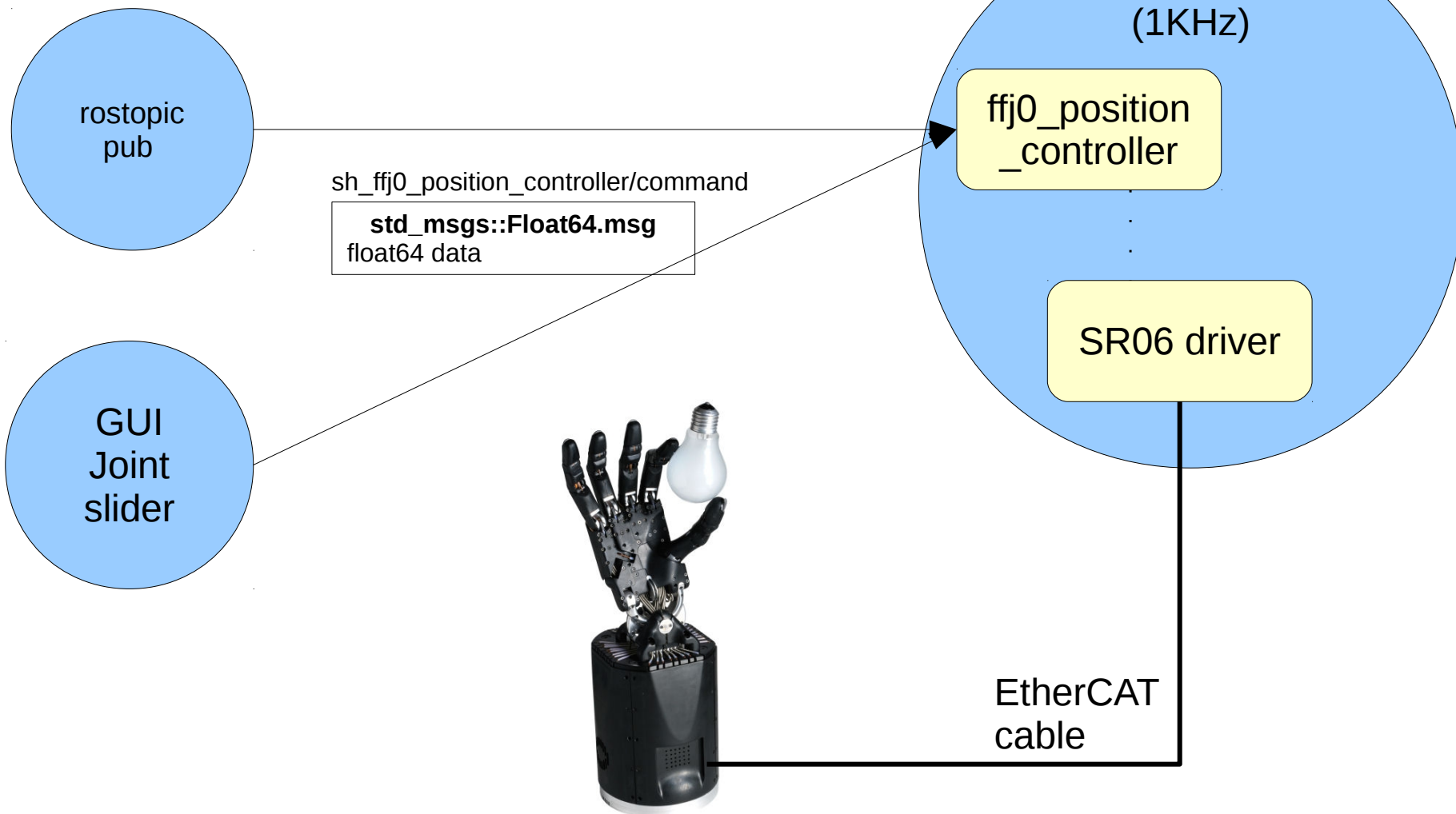
- Methods:
  - **setCommandCB**: the callback of the command input topic
  - **update**: takes last command and calculates and sets the effort value for this joint.
- A PID loop is used in the controller.





# Nodes/Topics

```
> rostopic pub  
sh_ffj0_position_controller/com  
mand std_msgs/Float64 1.5
```





# Outline

Hardware architecture

Host side controller architecture

Control demo



# Topics

- rostopic list

/debug\_etherCAT\_data

/joint\_states

/sh\_ffj0\_mixed\_position\_velocity\_controller/command

/sh\_ffj0\_mixed\_position\_velocity\_controller/state

/tactile

/tf





# Services

- `rosservice list`

`/SimpleMotorFlasher`

`/pr2_controller_manager/list_controllers`

`/pr2_controller_manager/load_controller`

`/pr2_controller_manager/switch_controller`

`/pr2_controller_manager/unload_controller`

`/realtime_loop/change_control_type`

`/realtime_loop/change_force_PID_FFJ0`

`/realtime_loop/reset_motor_FFJ0`

`/sh_ffj0_mixed_position_velocity_controller/set_gains`