

Perception and Control

Perception and Control

Ros sensor fusion

ros basic knowledge

Install Instructions

ethzasl_xsens_driver

pico_flexx_driver

Usage Instructions

Communication of STM32

Ros sensor fusion

Description: The `sensor_test` ros package is used to process and fuse sensor data. It collects IMU and camera data by the corresponding ros wrappers. It transforms the point cloud in to the world coordinate. It predicts the environment classes and send the result to the control unit (normally a STM32) by serial port.

ros basic knowledge

Go through the [ros tutorial](#) of the *beginner level*.

You should be able to install new packages and understand the basic concepts like *rostopic*, *roscall* and *rospack*

Install Instructions

Two ros packages are needed.

ethzasl_xsens_driver

This is the ros wrapper for the XSens MT/MTi/MTi-G devices.

Can be found [here](#).

Testing: `roslaunch xsens_driver xsens_driver.launch device:=/dev/ttyTHS1`

Note:

- if use a USB, `device:=/dev/ttyUSBx`
- if use a serial port (Tx/Rx), `device:=/dev/ttyTHS1`

pico_flexx_driver

This package is a ROS interface to the CamBoard pico flexx from pmd.

Can be found [here](#).

Testing: `roslaunch pico_flexx_driver pico_flexx_driver.launch`

Usage Instructions

- Step 1: `Ctrl+Alt+T` to open a new terminal and open a rosmaster by `roscore` in this terminal.
- Step 2: Start the camera node. `Ctrl+Shift+T` to open a new tab in the terminal in step one. Start the camera node by `roslaunch pico_flexx_driver pico_flexx_driver.launch`.

- Step 3: Start the IMU node. `Ctrl+Shift+T` to open a new tab in the terminal in step one. Start the IMU node by `roslaunch xsens_driver xsens_driver.launch device:=/dev/ttyTHS1`.
- Step 4: Start the prediction node. `Ctrl+Shift+T` to open a new tab in the terminal in step one. Start the prediction node by `roslaunch sensor_test prediction.py`. I wrote a ros package to run the python script, but it is not necessary.

The main script is `/ROS_sensor_fusion-master/src/sensor_test/scripts/prediction.py`.

Communication of STM32

Description: It contains the instructions to enable the communication of the master computer (PC) and the control unit (STM32) by using ROS.

Find more information [here](#).