Robotiq EPick

A ros node for communication with Robotiq EPick suction gripper.
This node uses RTU communication with the suction gripper.
The code for this gripper was mostly copied from: https://github.com/ros-industrial/robotiq

Setup

To use the gripper pymodbus package must be installed.

```
pip install pymodbus
```

Find the device and pass it as the frist argument when launching the file.
Easiest way is by searching:

```
dmesg | grep tty
```

And finding the last connected device.

You also need to add permissions to group to communicate with USB.

```
sudo adduser $USER dialout
```

> Note: Refer to EPick operational manual.

Communication with EPick

Robotic_epick_node listens to **"RobotiqEPickRobotOutput"** topic for **commands** to EPick suction gripper.
The **state** of the suction gripper is updated to **"RobotiqEPickRobotInput"** topic.

To use the gripper include inputMsg and outputMsg to your ros node.
Than communicate with the gripper by posting commands to respective topics.

> Note: Partial commands are not saved. Every time we send a new command we should set all bits correctly not just the one we want to change.

> Note: Refer to end of README to understand the ros MSG.

Using test program

To test the EPick suction gripper run the following command:

```
roslaunch robotiq_epick_control robotiq_EPickSimpleCOntroller_launch.launch
```

By typing the commands on the screen we may use the gripper. If we want to use it in manual mode we may change the parameters directly in the **robotiq_epick_simplecontroller_node.py** file.

Output MSG (Gripper control)

uint8 rACT

0 = clear fault

1 = activate gripper

uint8 rMOD

0 = automatic mode (griper sets vacuum level, timeout, hysteresis)

1 = advanced mode (user sets the settings)

unit8 rGTO

0 = stop vacuum generation

1 = Follow the requested vacuum parameters in real time. When timeout is reached, rGTO must be re-asserted

uint8 rATR

0 = normal operation

1 = Open the valves without any timeout. After an automatic release, rACT must be re-asserted

uint8 rPR

0 = vacume generator always on

22-90 = grip from 100% to 10% of vacum. rPR = 90(Min device vacum: 10%) rPR = 22(Max device vacum: 78%)

100 = Passive relese, relese to ambient pressure

101-255 = Active relese. Release with positive preasure.

uint8 rSP

0-255 = timeout (1 = 100ms)

uint8 rFR

0-99 = object detection preasure in %

Input MSG (Gripper status)

uint8 gACT

The gACT bit is the echo of the rACT bit in the ACTION REQUEST register

uint8 gMOD

The gMOD bits are the echo of the rMOD bits in the ACTION REQUEST register.

uint8 gGTO

The gGTO bit is the echo of the rGTO bit in the ACTION REQUEST register. Valid only if the vacuum/pressure is regulated, otherwise it returns 0x0.

uint8 gSTA

The rSTA bits indicates the status of the gripper activation sequence.

0b00 = Gripper is not activated

0b11 = Gripper is operational.

uint8 gOBJ

0b00 - Unknown object detection. Regulating towards requested vacuum/pressure.

0b01 - Object detected. Minimum vacuum value reached.

0b10 - Object detected. Maximum vacuum value reached.

0b11 - No object detected. Object loss, dropped or gripping timeout reached.

uint8 gVAS

0b00 - Standby. Vacuum generator and valves deasserted (OFF).

0b01 - Gripping. Vacuum generator ON.

0b10 - Passive releasing. Releasing to ambient pressure.

0b11 - Active releasing. Releasing with positive pressure

uint8 gFLT

The gFLT bits indicates priority, minor or major fault codes that are useful for troubleshooting.

uint8 gPR

This register is the echo of the MAXIMUM VACUUM/PRESSURELEVEL REQUEST register.

uint8 gPO

The gPO is the actual vacuum/pressure measured in the suction cups