ExFacLab Documentation

SDVUNx: *sdv_un_ros* installation

This ROS Metapackage is a fork of AGV-UN-ROS project from LabFabEx. This project tries to update all SDV software to ROS-Melodic, removing old packages and abandoned projects.

Software Requirements

- Ubuntu 18.04 Server
- ROS Melodic
- Python 2.7
- Python 3.x

Aditional requirements for Intel Realsense Cameras

- OpenCV

Installation

1. On Ubuntu 18.04 install ROS and create a ROS workspace

   Install ROS melodic and create a ROS Workspace

   It's recommended to use *catkin tools*. This python package its a wrapper of *catkin* that contains many useful tools. Visit this link for instructions.

2. Clone the sdv_un_ros project

   ```
   cd ~/catkin_ws/src/
   git clone https://gitlab.com/LabFabEx/sdv_un_ros.git
   ```

3. Install packages for ROS-Melodic: Install ROS packages required with command below:

   ```
   sudo apt install ros-melodic-lms1xx ros-melodic-map-server ros-melodic-move-base ros-mel
   ```

4. Catkin_make: For the first catkin_make is highly recommended execute the following commands on terminal:

   ```
   cd ~/catkin_ws
   catkin_make sdv_serial_generate_messages
   source $HOME/catkin_ws/devel/setup.bash
   catkin_make
   ```

5. Download the SDV laser packages

   - For and SDV with **Sick NAV350** laser: Driver for this laser model needs to be build in catkin workspace.

     1. Clone **sicknav350**: TO-DO
     2. Build catkin workspace:

```
cd ~/catkin_ws
catkin_make
```

- For an SDV with **Sick LMS102** laser: Required drivers are part of ros-melodic-lms1xx and not require extra software.

6. Configure Network Adapters to connect the Sick laser, allowing Internet through WiFi and laser messages through Wired Ethernet

   SDVs works with Ubuntu 18.04 Server. This OS allows to configure the NUC PC whitout a desktop running a GUI. Network connections can be edited via terminal. Lasers communicate with NUC through Ethernet Interface, while Internet is acccesed via WiFi.

   To make network configuration tasks easier, use networkd and netplan. In only one text file, you can set Wifi and Ehternet intefaces.

Next steps describe how to configure Wifi and Ethernet interfaces.

- Open a terminal in SDV-NUC
- Create a file in /etc/netplan folder named “01-netcfg.yaml”. Use next command:

```
sudo vim /etc/netplan/01-netcfg.yaml
```

- Add next content in created file to configure interfaces. Replace **password** value with current password assigned to LabFabEx Wireless Network:

```
network:
version: 2
renderer: networkd
ethernets:
   eno1:
      dhcp4: no
      addresses:
      - 192.168.0.158/24
      optional: true
      routes:
      - to: 192.168.0.0/24
         via: 192.168.0.1
         metric: 100
         on-link: true
wifis:
   wlp2s0:
      dhcp4: yes
      access-points:
      "LabFabEx":
         password: "<password_here>"
      gateway4: 192.168.1.1
```

- Save file. Next, apply configurations to system:

```
sudo netplan apply
```

- Check configuration using ifconfig to view assigned IP Addresses in every interface.

7. Change permissions on serial port

   With this configuration, NUC can communicate with Tiva via USB port. Execute this command line and then, reboot NUC PC.

```
sudo usermod -a -G dialout $USER
```

8. Edit .bashrc file

   Edit .bashrc file to configure ROS every time you run a terminal with bash. For example, for SDV2, add following lines to .bashrc file where 192.168.1.12 is the IP address of SDV2:

   ```
   #SDVUN2
   export ROS_MASTER_URI='http://192.168.1.12:11311'
   export ROS_IP='192.168.1.12'
   export ROS_NAME='192.168.1.12'

   source /opt/ros/melodic/setup.bash
   source $HOME/catkin_ws/devel/setup.bash
   ```

   Make necessary changes to match IP values with SDVUNx robot model.

9. Set **sdv_params.yaml** in **sdv_nav** folder

   All SDV shares software, but an SDV needs to set some ROS parameters required every time we launch sdv_process.launch file or another launch files. To make this work easier, we have to copy a predefined **.yaml** file contained in **params** folder, acording to SDV model.

   - Got to **sdv_nav** package folder:

     ```
     cd $(rospack find sdv_nav)
     ```

   - Link a **.yaml** file stored in **params** folder according to SDV model, and rename it as **sdv_params.yaml**. Example for configuring SDV2:

     ```
     ln -s params/sdv2_params.yaml sdv_params.yaml
     ```

   - Check that **sdv_params.yaml** data is similar to SDV model:

     ```
     cat $(rospack find sdv_nav)/sdv_params.yaml
     ```

10. Add Realsense packages This step is required for SDVs that uses RealSense Depth Cameras. Follow the specific guide for this step (search in ExFacLab documentation).

11. Init PRIA

    PRIA for SDVUNx is installed in **sdv_process** package and requires **Node** to works. PRIA for SDVUNx requires **ros-coms** and **firebase-coms** packages, that belongs to LabFabEx Software Repository as part of PRIA Project. Next steps describe how to add an configure this software in every SDV.

    - Add Node 11.x repository to Ubuntu. Follow instructions in this [link] (https://www.ubuntuupdates.org/ppa/nodejs_11.x).
    - Go to **sdv_process** package:

      ```
      cd $(rospack find sdv_process)/src
      ```

    - Clone **ros-coms** package. You may require permissions from LabFabEx repository admins:

      ```
      git clone git@gitlab.com:LabFabEx/ros-coms
      cd $(rospack find sdv_process)/src/ros-coms
      ```

- Init submodules of **ros-coms** package:

  git submodule init git submodule update

- Go to firebase-coms folder. Then, install Node modules and build Typescript files. After this, two new folders will appear (dist and node_modules):

```
cd $(rospack find sdv_process)/src/ros-coms/firebase-coms
npm install
npx tsc
```

- Now, go to ros-coms folder. Then, repeat the installation of Node modules and build Typescript files:

```
cd $(rospack find sdv_process)/src/ros-coms
npm install
npx tsc
```