Care-O-bot Manual

Contents

Chapter 1

Introduction

This manual is divided into two main parts. The first part (chapter 2) adresses robot administrators and covers topics like setting up the pcs, configuring network and add new user accounts. The second part (chapter 3) is intended for all users, it shows how to startup the robot, login and execute simple commands on the robot.

Chapter 2

Administrator manual

2.1 Setup robot pcs

On all Care-O-bots there are at least two pcs. Some Care-O-bots have an optional third pc, which is not coveres by this manual. Within this section we will guide you through setting up new pcs. When nothing otherwise is mentioned the following instructions are for both pc1 and pc2, please do the same steps on both pcs.

2.1.1 Install operating system

The first step is to install the operating system for each pc, which means pc1 and pc2 (optionally pc3). We are using Ubuntu as the main operating system for the robot. We recommend to install the **Ubuntu 10.4 LTS (long term stable) 64-bit** version because this version is well tested to work with the hardware.

For the first installating please install Ubuntu (english version) creating a normal swap partition. Please choose *robot* as an admin account with a really safe password which should only be known to the local robot administrator. The hostname of the pc should be *cob3-X-pc1* and *cob3-X-pc2*.

2.1.2 Install basic tools

Next we have to install some basic tools for the further setup of the pcs. In order to install the packages a internet connection is needed.

```
sudo apt-get update
sudo apt-get install vim tree openssh-server gitg meld curl
```

To facilitate the further setup we created a setup repository with some helpfull scripts. To checkout the setup repository use:

```
mkdir ~/git
cd ~/git
git clone git@github.com:ipa320/setup.git
```

2.1.3 Setup internal robot network

Inside the robot there's a router which connects the pcs and acts as gateway to the building network. Setup the router with the following configuration.

The ip adress of the router should be **192.168.0.1** and for the internal network dhcp should be activated. Use **cob3-X** as hostname for the router. Register the MAC adresses of pc1 and pc2 so that they get a fixed ip adress over dhcp. Use **192.168.0.101 for pc1** and **192.168.0.102 for pc2**. Enable **port-forwarding** for port 2201 to 192.168.0.101 and for port 2202 to 192.168.0.102.

After ensuring that the network configuration of the router is setup correctly, we can configure the pcs. All pcs should have two ethernet ports. The upper one should be connected to the internal router. Sometimes the graphical network manager causes troubles, so it is best to remove it

```
sudo apt-get remove network-manager
```

After removing the network manager we will have to edit */etc/network/interfaces* manually.

Network configuration on pc1

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.0.101 # internal ip adress of pc1 for router
netmask 255.255.255.0 # netmask

auto eth1
iface eth1 inet static
address 192.168.42.1 # ip adress for controller network
netmask 255.255.255.0 # netmask
```

Network configuration on pc2

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.0.102 # internal ip adress of pc2
netmask 255.255.255.0 # netmask

auto eth1
iface eth1 inet static
address 192.168.21.99 # ip adress for camera network
netmask 255.255.255.0 # netmask
```

2.1.4 Install NFS

After the network is configured properly we can setup a NFS netween the robot pcs. pc2 will act as the NFS server and pc1 as NFS client.

NFS configuration on pc2 (server)

Install the NFS server package and create the NFS directory

```
sudo apt-get install nfs-kernel-server
sudo mkdir /u
```

Add the following following line to */etc/fstab*:

```
/home /u none bind 0 0
```

Now we can mount the drive

```
sudo mount /u
```

Activate IDMAPD in */etc/default/nfs-common* by changing the NEED_IDMAPD to yes

```
NEED_IDMAPD=yes
```

Copy the file *~/git/setup/nfs_setup/server/exports* to */etc/exports*

```
cp ~/git/setup/nfs_setup/server/exports /etc/exports
```

Change the home directory of the *robot* user from */home/username* to */u/username* in the */etc/passwd* file.

After finishing you need to reboot the pc

```
sudo reboot
```

NFS configuration on pc1 (client)

Install the NFS client package and create the NFS directory

```
sudo apt-get install nfs-kernel-server autofs
sudo mkdir /u
```

Activate IDMAPD in */etc/default/nfs-common* by changing the NEED_IDMAPD to yes

```
NEED_IDMAPD=yes
```

Edit */etc/auto.master* and add

```
/- /etc/auto.direct
```

Copy the file *~/git/setup/nfs_setup/client/auto.direct* to */etc/auto.direct*

```
cp ~/git/setup/nfs_setup/client/auto.direct /etc/auto.direct
```

Activate the NFS

```
sudo update-rc.d autofs defaults
sudo service autofs restart
sudo modprobe nfs
```

Change the home directory of the *robot* user from */home/username* to */u/username* in the */etc/passwd* file.

After finishing you need to reboot the pc

```
sudo reboot
```

2.1.5 Setup NTP time synchronitation

Install the ntp package

```
sudo apt-get install ntp
```

NTP configuration on pc1 (NFS server)

Edit */etc/ntp.conf*, change the server to *cob3-X-pc1* and add the restrict line

```
server 0.pool.ntp.org
restrict 192.168.0.0 mask 255.255.255.0 nomodify notrap
```

NTP configuration on pc2 (NFS client)

```
server server cob3-X-pc1
```

2.1.6 Install ROS and Care-O-bot driver software

For genreal instructions see http://www.ros.org/wiki/Robots/Care-O-bot/electric.

Install additional tools

```
sudo apt-get install openjdk-6-jdk zsh terminator
sudo apt-get install python-setuptools
sudo easy_install -U rosinstall
sudo apt-get install ros-diamondback-care-o-bot ros-diamondback-perception-pcl-addons ros-diamondback-erratic-robot
sudo apt-get install ros-electric-care-o-bot ros-electric-perception-pcl-addons ros-electric-pr2-desktop ros-electric-pr2-robot ros-electric-pr2-apps pr2-power-drivers
```

Setup bash environment

For the three PC's you have to copy cob-bash-bashrc.pcX to /etc/cob-bash-bashrc and for all the users copy user.bashrc to /.bashrc you have these files on the setup folder.

2.2 PC's Overview

In PC1 usually bringup the components of the robot, PC2 is used to run the cameras and visual sensors and PC3 can be used to run extra nodes. When you launch the robot with the bringup file you have these nodes:

- PC1:
  Send the robot_description to the param server
  Start the the robot_state_publisher
  Startup the Hardware , launch the components
  Diagnostics
  Teleop
  Sounds
- PC2:
  Cameras (left, right, kinects)

2.3 Network

2.3.1 Using a route

You can setup a route to the internal network addresses. Please change the robot name and your network device to t fit your settings. E.g. for connecting to:

- cob3-X on eth0

```
sudo route add -net 192.168.0.0 netmask 255.255.0.0 gw cob3-X dev eth0
```

- or cob3-X on wlan0

```
sudo route add -net 192.168.0.0 netmask 255.255.0.0 gw cob3-X dev wlan0
```

You can check the settings with:

```
ping 192.168.0.101
```

2.3.2 Setup name resolution

To satisfy the ROS communication you need a full DNS name lockup for all machines. Therefore add the following addresses to your /etc/hosts. Please change the robot name to fit your settings

```
192.168.0.101 cob3-X-pc1
192.168.0.102 cob3-X-pc2
192.168.0.103 cob3-X-pc3
```

You can check the settings with:

```
ping cob3-X-pc1
```

2.4 Getting an account

On PC2 and with administration rights you can add an user with the following instruction:

```
sudo adduser new_user_name
```

If you want that this new user have also sudo rights you have to add it to the admin group in all PCs.

2.5 Calibration

Now is not working....

2.6 Backup and restoring users

Now is not working...

2.7 todo

- udev from gitsetupudev_rules01-cob.rules copy to etcudevrules.d on pc1

  to check it: ls -l /dev/ and you should have these lines

  lrwxrwxrwx 1 root root 7 2012-01-17 10:27 ttyRelais -¿ ttyUSB0
  lrwxrwxrwx 1 root root 7 2012-01-17 10:27 ttyScan0 -¿ ttyUSB1
  lrwxrwxrwx 1 root root 7 2012-01-17 10:27 ttyScan1 -¿ ttyUSB2
  lrwxrwxrwx 1 root root 7 2012-01-17 10:27 ttyTact -¿ ttyUSB3

- Camera config you have to change the ip adrrees in the (eth1?) in /etc/network/interfaces it should be on pc2

  auto eth3 iface eth3 inet static address 192.168.21.99 # IP of network adapter to cameras netmask 255.255.255.0 # netmask

  sdh add to dialout group

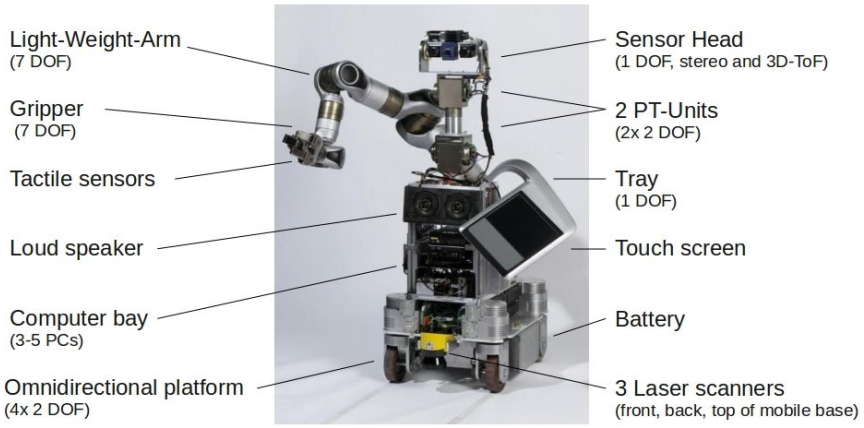
Chapter 3

User Manual

3.1 Hardware overview

You can take a look of the technical data of Care-O-Bot in he official web : `http://www.care-o-bot-research.org/care-o-bot-3/technical-data` and `http://www.care-o-bot-research.org/care-o-bot-3/components` Also you can see the distribution of the different Care-O-Bots in `http://www.ros.org/wiki/Robots/Care-O-bot/distribution`

It is important an overview of the emergency stop localizations and Start/Stop key , the plugged cable of the battery and the scanners:



3.2 Software overview

The bringup level repositories of Care-O-bot are the following:

- cob_extern : The cob_extern stack contains third party libraries needed for operating Care-O-bot. The packages are downloaded from the manu-

facturers website and not changed in any way.

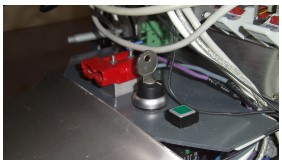
- cob_common : The cob_common stack hosts common packages that are used within the Care-O-bot repository. Also the URDF description of the robot, kinematics and dynamics, 3D models of robot components, information required for gazebo to simulate the COB and utility packages or common message and service definitions
- schunk_modular_robotics : It is the cob_common stack for the components of Schunk in this case lwa and sdh.
- cob_driver : The cob_driver stack includes packages that provide access to the Care-O-bot hardware through ROS messages, services and actions. E.g. for mobile base, arm, camera sensors, laser scanners, etc...
- cob_robots : The cob_robots stack collects Care-O-bot components that are used in bringing up a robot. The user's interface to the cob_robots stack is cob_bringup, where are localize the launch files of the robot.
- cob_environments : This stack provides the parameters of the environments configuration.
- cob_command_tools: This stack provides the source code of the tools that you need to command instructions to the robot: cob_command_gui, cob_dashboard, cob_script_server and cob_teleop.

3.3 Batteries and Power

Care-O-bot provides a Gaia rechargeable Li ion battery (60 Ah 48V) , in order to assure that Care-o-bot has always power it is a recommendable keep the robot plugged when it is not in use. The Power supply has to be set to 56 Volts. Before run the robot be sure that it has power.

3.4 Run the robot

First you have to connect the power supply to the robot or you can use the battery pressing the green button on the base. To switch on the robot you use the key that it has in the base , you have to move it to the position II and wait few seconds. It is recommendable stop the robot when you are not using it during some time with the emergency stop.



3.5 Logging In

For logging with a remote PC to the robot you have to have a account already create (see the section 1.4) and use a secure shell connection with the PCs of the robot (it is recommended do it with executable rights).

```
ssh -X user_name@cob3-X-pcX
```

3.6 Bringup

The first step to bringup the robot is the roscore, it is necessary to have communication between the nodes. You can run it using this command:

```
roscore
```

If you want to run the robot you have a launch file for launch all the components of the robot, it is localized in the package cob_bringup, you can call this file with the following instruction:

```
roslaunch cob_bringup robot.launch
```

3.7 Dashboard

To have always under control the state of all the components of the robot you can use the tool dashboard , it is in the package cob_bringup:

```
roslaunch cob_bringup dashboard.launch
```

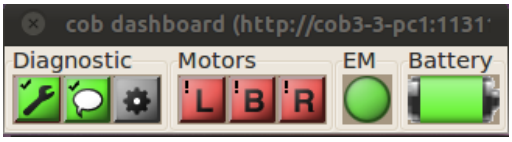
After this launch this file you will see in your display two windows , one is the command_gui , where you can init , recover and move to a predefine position the different components of the robot, in the superior left corner of the command_gui you see the current state of the robot, before move the robot , check that the status is OK. The second window is cob_dashboard, in this windows you can see the state of Diagnostics, Motors , EM(Emergency) and Battery. In the diagnostics you have there buttons Diagnostics, rosout and Motors.

In the case of the Care-O-bot we have disable the buttons for the Motors, you see them always in red.

If you click the first one you will see a new window with three levels: Errors, Warnings and All. There you can see anytime the state of each component.

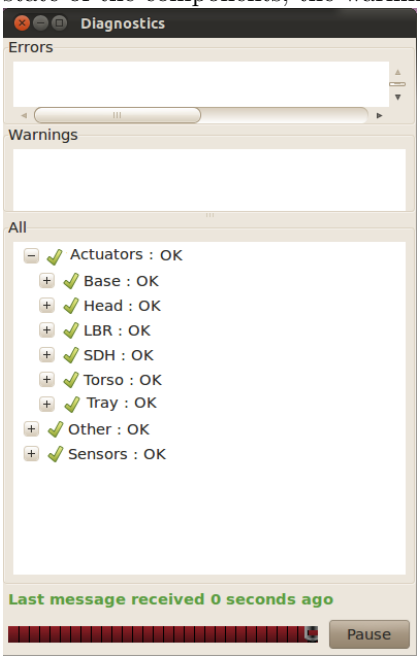
3.7.1 cob_dashboard

The dashboard is a important tool where you can check the the state of the robot, it is recommended that you have it always opened. This is the format:



The different buttons that you have are distributed into Diagnostics, Motors, EM and Battery.

- Diagnosticts
  - Diagnostics: Open the window of Diagnostics , with a list with the state of the components, the warnings and errors.



  - Rosout: you can check if you have communication between your nodes, it has three states OK/Error/Warm , it is determinate for the messages received the last 30 seconds.
  - Motors: it is disable for Care-O-Bot
- Motors : These buttons are disable.
- EM : In this display you can see the state, if it is red the emergency stop is activated if not it is green.

- Battery: you can see the state of the battery , green : full, orange : 40% and red 20%

3.7.2 Diagnostics

3.7.3 cob_command_gui

The standard view of the command_gui is:

In this screenshot you can see different columns: general, base, torso, tray, arm settings, arm pos, arm traj, sdh and eyes.

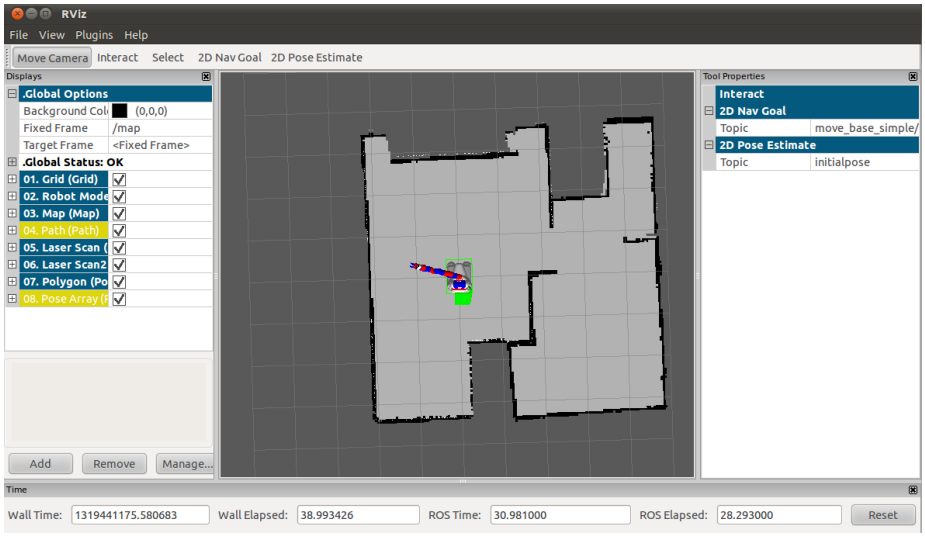
The first column is very important , when you run the robot, if you want to move it, the first that you have to do is click "init all" you will see in the window dashboard the initialization of each component, after a Emergency stop you have to press the button "recover all", the columns of the components have different predefine positions , you can change it in the configuration when you want , also each component has a "stop", "init" and "recover" button , when you see that you have an error in only one of these components you can press only "recover" for this one.

3.8 Rviz

RVIZ is a program that visualizes additional views to the robot e.g. the original images from the camera, the path where the Care-O-bot is moving to and many more. You can add your own items to RVIZ to visualize topics. You can see more information in `http://www.ros.org/wiki/rviz` You can communicate the robot with your local PC and execute RVIZ in it:

```
export ROS_MASTER_URI=http://cob3-3-pc1:11311
rosrun rviz rviz
```

You will see a screen like this:



If you see that the robot is not in the same position in the real environment than in Rviz you have to localize it using the buttons of Rviz 2D Nav Goal and 2D Pose Estimate

3.9 Joystick

To be able to use the joystick the deadman_button has to be pressed all the time, as soon as the button is released a stop will be send to all hardware components.

- For moving the base: Hold the deadman button and use the base rotation and translation axis to move the base.
- For moving the torso: Hold the deadman button and the upper or lower neck button, then use the up_down or left_right axis to move the torso.
- For moving the tray: Hold the deadman button and the tray button, then use the up_down axis to move the tray.
- For moving the arm: Hold the deadman button and one of the arm buttons, then use the up_down or left_right axis to move the selected arm joints.

Have a look at the following image to see which buttons command which components.

3.10 Emergency stop

The user has two possibilities to activate the EM Stop, on the robot you have two red buttons on the laterals you press it when you see that the Care-O-Bot will have an accident. The second possibility is with the Remote, you have to hold this remote control on your hand always that you move Care-O-Bot.

3.10.1 Emergency stop remote control

You can press the red button to stop the robot, after a emergency stop you have to restart the computers, to do it you have to be sure that you have already "up" all the red-emergency stop buttons, the press the green button of the Remote control and then move again the key of the robot to the II position. You will see on your remote PC a message "Emergency stop released!", the you have to press "recover all" in the on the command_gui , open Diagnostics and check that all components are green.

3.11 Putting away

You have to logout in the local PC (Ctrl+D) and press the emergency stop, then you can shut down the PCs, moving the key to the position I (left).

3.12 Support

If you have doubts , please use our Mainlist : `http://www.care-o-bot-research.org/contributing/mailing-lists`

3.13 Packing-Shipping