java need to be installed

su -c yum search java
yum install java-1.6.0-openjdk.i686

External Control — ExtCtrl

1 Content

1 Content 1
2 History 2
3 To Do / Notes 3
4 General description of the setup 5
5 Description of ExtCtrl 6
5.1 Architecture 6
5.1.1 Signals 6
5.1.2 Provided Simulink blocks 9
5.2 Communication between ExtCtrl-Level and RAPID-Level 9
5.2.1 RAPID Example 9
5.2.2 ExtCtrl Example 11
5.2.2.1 MATLAB/Simulink 11
5.2.2.2 Scilab/XCos 11
5.2.2.3 Plain C 11
5.3 Connection via Ethernet and RS232 11
5.4 Side effects 11
6 Installation of ExtCtrl 13
6.1 ExtCtrl installation on the PC-Side 13
6.1.1 Installation of Fedora based ExtCtrl from Live-USB 13
6.1.2 Finish Fedora based ExtCtrl installation during reboot 18
6.1.3 Further setup of ExtCtrl-PC after reboot 18
6.1.3.1 Check for right kernel 18
6.1.3.2 Login as root 18
6.1.3.3 Check network configuration 18
6.1.3.4 Run orca_setup 19
6.1.3.5 Saving the current configuration 20
6.1.3.6 Check for real time ethernet card 20
6.2 Wiring 21
6.2.1 RS232 serial port 21
6.2.1.1 Rights of /dev/tty* 21
6.2.2 Real time Ethernet 21
6.3 Installing correct RobotWare-system 21
6.3.1 Making copy/backup of an already running system 22
6.3.2 Installing the robot system 22
6.4 Installation Robot-Side 23
6.4.1 Set write access on the IRC5 23
6.4.1.1 Terminal with MS Windows 23
6.4.1.2 Terminal with Linux 24
6.4.1.3 VxWorks commands 25
6.4.2 Exchange mc and mc.sym 26

6.4.3 Change install files....27
6.4.3.1 Identify robot type....27
6.4.3.2 Activate ExtCtrl....27
!!! You need to have activated with -use_ilc "ext_ctrl" before you continue. .28
Choose feedforward mode for better tracking with extctrl....28
6.4.3.4 Brake timer....29
6.4.3.5 Configuration-file for extctrl, ffwmode and brake_timer....30
6.4.4 Perform I-Start....30
6.4.5 Copy ORCA files to the robot....31
7 Build a controller for ExtCtrl....33
7.1 MATLAB/Simulink....33
7.1.1 Installation and setup....33
7.1.2 Build a controller from MATLAB/Simulink....34
7.2 Scilab/XCos....34
7.2.1 Installation and setup....34
7.2.2 Build a controller from Scilab/XCos....34
7.3 Plain C-Code....34
7.3.1 Installation and setup....34
7.3.2 Build a controller from plain C-Code....34
8 Run ExtCtrl (Test ExtCtrl installation)....35
8.1 Booting the system....35
8.2 Run Opcom....35
8.3 Logging....36
8.4 Switch between ExtCtrl and normal operation....36
9 Appendix....38
9.1 Some useful Linux commands....38
9.2 Some useful VxWorks commands....38
9.3 How to create a LiveUSB-Stick....38
9.3.1 On Windows....38
9.3.2 On Linux....41
9.3.2.1 Fedora....41
9.3.2.2 Any other Linux or Unix distribution....42
9.4 Some Experiments....42
9.4.1 Gear ratios IRB140 Type C....42
9.4.2 Gear ratios IRB140 Type A....43
9.4.3 Fast movements with and without having ExtCtrl enabled....44

2 History

V 0.1	07.03.2011	Denis Störkle	denis.stoerkle@de.abb.com
V 0.11	06/10/11	Anders Robertsson	Anders.Robertsson@control.lth.se
V 0.12	15/10/11	Björn Olofsson	Bjorn.Olofsson@control.lth.se
V 0.13	19/10/11	Anders Robertsson 2_B_UPDATED!!	Anders.Robertsson@control.lth.se

3 To Do / Notes

- Influnece on large robot far out in workspace when releasing brakes withous trq_fwd!!
- Update /usr/robot/matlab/irb/include/robotdata.h consistently with new robots
- Edit for copy/paste from documentation (now inconsistencies with " " etc)
- Klink / NIKON
    - orca_update - -noyum - -install /orca/klink-k600-new etc (kan ta lång tid)
    - lägg till länk till rätt Mac-adress/nätverkskort
        - ls -l /etc/NetworkManager/dispatcher.d
        - ...
        - lrwxrwxrwx 1 root root 36 4 feb 18.15 10-klink-00:0a:cd:20:78:ae -> /usr/local/sbin/networkmanager-klink*
    - add correct vsg-file and xml for frames (see description from Björn)
    - change in network setting for KLINK-ethernet card to be
        - IP 192.168.254.254
        - network mask 255.255.255.255
        - empty gateway
        - NIKON-PC allowed to have 192.168.1.2 (defined in /usr/local/sbin/klink-start)reglerinmgen

- Config-files in HOME/SYSPAR/ or in RW.../robots/irbcfg/ ??
- problem of updating from newer system keys ....!! + ABB-document to be signed
-
- Makefile in /opt/robot/matlab/irb/mex does not recognize changes in robotdata.h (change dependencies)
- 2012-06: Makefile does not work (sed ??)

Need to have matlab/mex installed/linked to /usr/local/bin/mex...

- New robotdata.h-file to distribution (with IRB140_6 and IRB6660_205_1_9) and no offsets of DH-parameters (should be individual for each robot – need to be changed !!)
- Multiple kinematics except (embedded m-functions work-a-round)
- more firewall-comments, how to work with external sensors...

- Baumer camera and Linux
- _
- Clarify confusion about "–use_ilc"!!
    - [Q?] Is parameter loading sufficient or do I need to change entry in in EXT*.cfg?
    - [A:] It is enough to have e.g. two files with only the entries for "ROBOT:"

where one contains -use_ilc for activation
and the other without for deactivation of extctrl-functionality

- Verify functionality of ExtCtrl in combination with existing ABB features
  - Safe Move
  - MultiMove
  - External Axis
- Detailed description of the ExtCtrl:
  - Premade blocks in Simulink
  - How to change robot type
  - [Q?] Motor angles / gear ratios
    [A:] An approximate printout of these are given by a function call in VxWorks when the new mc has been installed:
    - > print_extctrl_info
  - Networking
- FH Darmstadt
  - Rethink network structure to have only few changes between ExtCtrl and Lab stuff
- Are there any real alternatives to Scilab/XCos?
- Question for
  - Anders Blomdell / Anders Robertsson
    - How is the real time connection between PC and IRC5 established?
    - Latency of ExtCtrl
    - What needs to be changed for different robot types?
    - Gear ratios
      (A: See print_extctrl_info above)
  - Christian Goy
    - Are there any size limitations for the IRC5 flash disc? Industrial norm!!!
      (partial answer on robotlab-wiki
      https://www.control.lth.se/internal/RobotLab#FlashDrive)
    - Latency measured values? Difference between filtered and unfiltered?

4 General description of the setup

This documentation was created from experience of several installations, originally of two Ethernet based ExtCtrl systems with respect to the ExtCtrl system running at LTH. One ExtCtrl systems was installed at ABB Robotics in Friedberg, the other one at the University Of Applied Sciences Darmstadt.

- DA:
  - ExtCtrl-PC:
    - Transtec
    - CPU: Intel Core 2 Duo CPU – E7400 @ 2.80GHz
    - Memory:4GB
    - OS: Fedora 14
    - Kernel: 2.6.35.7_xenomai-2.5.5.2_rtnet-39f7fcf
    - RT Ethernet card: Realtek TRL-8139/8139C/8139C+
  - Robot System:
    - Robot: IRB140 Type C – 140T-6/0.8 Type C [IRB140T-YH6]
    - Controller: IRC5
    - RobotWare version: 5.13_01.01.1037
- FB:
  - ExtCtrl-PC:
    - Lenovo ThinkCentre M Series
    - CPU: Intel Pentium CPU – G6950 @ 2.80GHz
    - Memory: 2GB
    - OS: Fedora 14
    - Kernel: 2.6.35.7_xenomai-2.5.5.2_rtnet-39f7fcf
    - RT Ethernet card: Intel PRO/1000 GT Desktop Adapter [Product Code: PWLA8391GT]
  - Robot System:
    - Robot: IRB140 Type A – 140-5/0.8 Type A [IRB140]
    - Controller: IRC5
    - RobotWare version: 5.13_01.01.1037

5 Description of ExtCtrl

5.1 Architecture

5.1.1 Signals

In the following schematics, the output signals of the main computer (MC) and the feedback control of a single joint running on the axis computer (AXC) are shown:

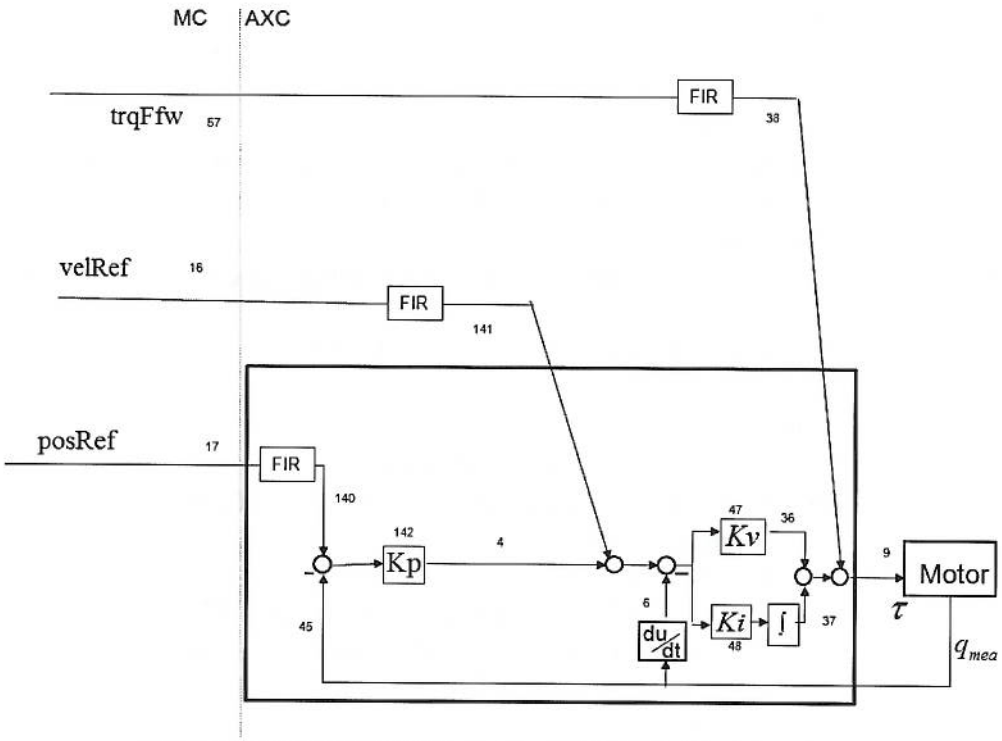


Figure 1: Robot control schematics with ExtCtrl test signals.

All input signals with the according output signals (if there is one) that are currently used in ExtCtrl, are listed and described shortly in the following table.
All the input signals of ExtCtrl are output signals of the robot controller's main computer and have the prefix "irb2ext".
All output signals of ExtCtrl are input signals of the robot controller's axis computer and have the prefix "ext2irb".

Table 1: In- and output signals of ExtCtrl.

Input	Output	Description
Motion specific signals		

irb2ext. joint[i].parKp	ext2irb. joint[i].parKp	Position control proportional gain
irb2ext. joint[i].parKv	ext2irb. joint[i].parKv	Velocity control proportional gain
irb2ext. joint[i].parKi	ext2irb. joint[i].parKi	Velocity control integration gain
irb2ext. joint[i].parTrqMin	No output signal	Minimum torque (control output) allowed
irb2ext. joint[i].parTrqMax	No output signal	Maximum torque (control output) allowed
irb2ext. joint[i].posRawAbs	No output signal	Measured unfiltered motor angle. [abs]
irb2ext. joint[i].posRawFb	No output signal	Measured unfiltered motor angle. [feedback]
irb2ext. joint[i].posFlt	No output signal	Filtered motor angle
irb2ext. joint[i].velRaw	No output signal	Unfiltered angular velocity
irb2ext. joint[i].velFlt	No output signal	Filtered angular velocity, also for motor shaft
irb2ext. joint[i].velOut	No output signal	Filtered angular velocity according to actual velocity control ref.
irb2ext. joint[i].trqRaw	No output signal	Measured (or internal) unfiltered torque
irb2ext. joint[i].trqRefFlt	No output signal	Filtered Torque ref
irb2ext. joint[i].posRef	ext2irb. joint[i].posRef	Nominal position reference
irb2ext. joint[i].velRef	ext2irb. joint[i].velRef	Nominal velocity reference/feedforward

irb2ext. joint[i].trqFfw	ext2irb. joint[i].trqFfw	Model-based torque feedforward
irb2ext. joint[i].trqFfwGrav	No output signal	Total dynamic model based torque

Information about the mode and the state of the robot controller

irb2ext. obtaining	No output signal	Specifies if ExtCtrl is in obtain
irb2ext. manualMode	No output signal	Specifies the mode of the robot controller
irb2ext. controlActive	No output signal	Specifies the state of the robot controller

Communication via MocGenInstr using MocGenData

irb2ext. mocgendata.instruction	ext2irb. mocgendata.instruction	Instruction number of the MocGenData
irb2ext. mocgendata.value1	ext2irb. mocgendata.value1	Value 1 to 6 of the MocGenData that has been sent, which is of type NUM in RAPID and of type ??? in Simulink
irb2ext. mocgendata.value2	ext2irb. mocgendata.value2	
irb2ext. mocgendata.value3	ext2irb. mocgendata.value3	
irb2ext. mocgendata.value4	ext2irb. mocgendata.value4	
irb2ext. mocgendata.value5	ext2irb. mocgendata.value5	
irb2ext. mocgendata.value6	ext2irb. mocgendata.value6	
irb2ext. mocgendata.string1	ext2irb. mocgendata.string1	String 1 to 2 of the MocGenData that has been sent

irb2ext. mocgendata.string2	ext2irb. mocgendata.string2	

5.1.2 Provided Simulink blocks

List of the provided Simulink blocks and a description of them.
Motor2Arm
AngleFix ;-)
Inverse and forward kinematics
WHAT standard libraries to be packaged??

- (extctrl.mdl),
- new model with addpath
- ...

5.2 Communication between ExtCtrl-Level and RAPID-Level

In case, that the RAPID program running on the robot controller and the ExtCtrl are meant to interact, a way of communication is required to synchronize RAPID program execution and ExtCtrl.
Therefore MocGenInstr/MocGenData, with an instruction number **between 88998 and 89600**, is used. Since the MocGenInstr-function call does not indicate the direction of communication, the convention is, that instructions with an **even** instruction number are messages that are sent from **RAPID to ExtCtrl** and instructions with an **odd** instruction number are messages that are sent from **ExtCtrl to RAPID**.

5.2.1 RAPID Example

The following RAPID-Code exemplifies the communication between RAPID and ExtCtrl using MocGenInst/MocGenData. The robot is meant to move to its Home-Position and send a message to ExtCtrl to switch the influence of ExtCtrl on.

```
MODULE MainModule

  ! Declaration of jtHOME jointtarget, defining robot's home-pos
  CONST jointtarget jtHOME      := [[0,0,0,0,0,0],[0,0,0,0,0,0]];

  ! Declaration of mydata mocgendata, used for MocGenInstr
  VAR   mocgendata  mydata      := [0,0,0,0,0,0,"str1","str2"];


  PROC main()

        ! Move the robot to jtHOME
        MoveAbsJ jtHOME, v1000, fine, tool0;

        ! Switch on the influence of ExtCtrl
        switchOnExtCtrl;

  ENDPROC !! main
```

```
PROC switchOnExtCtrl()

      ! Write a message to the FlexPendant
      TPWrite "Switch On The ExtCtrl!";

      ! Execute MocGenInstr with instruction number 89300 and
      ! the parameter mydata
      MocGenInstr 89300, mydata;

      ! Wait 100ms to make sure, that the two MocGenInstr are
      ! sent individually
      WaitTime 0.1;

      ! Execute MocGenInstr with instruction number 89300 and
      ! the parameter mydata
      MocGenInstr 89301, mydata;

      ! Write the received MocGenData to the FelxPendat using
      ! the WriteMocGenData procedure
      WriteMocGenData(mydata);

ENDPROC !! switchOnExtCtrl


PROC WriteMocGenData(mocgendata mydata)

      ! Write value 1 to 6 and string 1 to 2 of MocGenData
      ! mydata to the FlexPendant
      TPWrite "value1= " \Num:=mydata.value1 ;
      TPWrite "value2= " \Num:=mydata.value2 ;
      TPWrite "value3= " \Num:=mydata.value3 ;
      TPWrite "value4= " \Num:=mydata.value4 ;
      TPWrite "value5= " \Num:=mydata.value5 ;
      TPWrite "value6= " \Num:=mydata.value6 ;
      TPWrite "string1= "+mydata.string1;
      TPWrite "string2= "+mydata.string2;

ENDPROC !! WriteMocGenData

ENDMODULE
```

In the "main" procedure, the robot moves to its Home-Position and a message is sent to ExtCtrl to switch the influence of ExtCtrl on.

The procedure "switchOnExtCtrl", executes the procedure MocGenInstr with 89300 and mydata as parameters. With this call, a message is sent to ExtCtrl (due to convention of even and odd numbers). On the ExtCtrl side, this message is one of the input signals and can be used for any kind of task. In this very example, the instruction with the ID 89300 is used to switch on the influence of ExtCtrl, meaning that this needs to be implemented as such on the ExtCtrl side.
After a having waited for 100ms, the answer of ExtCtrl is read back and written to the FlexPendant.
PROC WriteMocGenData:

This procedure prints all values of a variable of the type MocGenData to the FlexPendant.

The procedure WriteMocGenData can be used to print a variable of the type MocGenData to the FlexPendant. It also shows the structure of the data type MocGenData.

5.2.2 ExtCtrl Example

5.2.2.1 MATLAB/Simulink

@TODO

5.2.2.2 Scilab/XCos

@TODO

5.2.2.3 Plain C

@TODO

5.3 Connection via Ethernet and RS232

The ExtCtrl-PC needs to be connected via Ethernet and via RS232 to the IRC5's main computer.

The RS232 connection is used to initiate the

@TODO: How is the real time Ethernet connection established? Overview Picture!?

5.4 Side effects

@TODO: Rewrite
Describe experiment with fast motions

Hej

I did a small experiment to test and visualize the difference between the ExtCtrl-Mode switched on or off (-use_ilc "ext_ctrl" or not). Therefore I mounted a pen as tool and wrote a small RAPID program executing two very fast linear movements to draw a line on the table the robot is mounted on.

I executed the program in automatic mode, to get the fastest possible movement of the TCP to make the influence of the dynamic model visible.

Conclusion:
Robot in ExtCtrl-Mode:
The robot is not able to draw a straight line with full speed, due to the missing dynamic model, the inertias are not compensated. The deviation reached a maximum of 15mm on

the 800mm long path. The line was shaped like an circular arc with the concave side facing towards the robot.

Robot in normal mode:
Even with full speed, the robot still draws a straight line.

In both cases, the start and the end point were reached exactly (by visual judgment). With this experiment, I didn't get any information about the accuracy of velocity and acceleration of the robot's movement. By visual judgment, the motion in normal mode (without -use_ilc "ext_ctrl") looked faster, but I am not really sure about that.

ADD experiment/figures (see mail from Anders)
also good with corresponding code RAPID/simulink modell

6 Installation of ExtCtrl

6.1 ExtCtrl installation on the PC-Side

To start the installation of the Fedora based ExtCtrl, the Orca live image is needed. This live image contains a Fedora distribution with a specific kernel compatible with ExtCtrl. The live image can be used to create a LiveDVD by burning it onto a DVD (due to the size of currently 1.3GB it won't fit a CD) or it can be used to create a LiveUSB-Stick (How to create a LiveUSB-Stick).

6.1.1 Installation of Fedora based ExtCtrl from Live-USB

Power the computer up after having the LiveUSB-Stick/LiveDVD plugged in and choose the booting device accordingly. If the boot device has been selected properly, Fedora should boot automatically after the GRUB-Timeout.
After the boot phase, the user will be logged in automatically and GNOME appears:

Figure 2: Fedora desktop after startup.

By double clicking on the desktop button "Install to Hard Drive", the Fedora based ExtCtrl installation process starts:

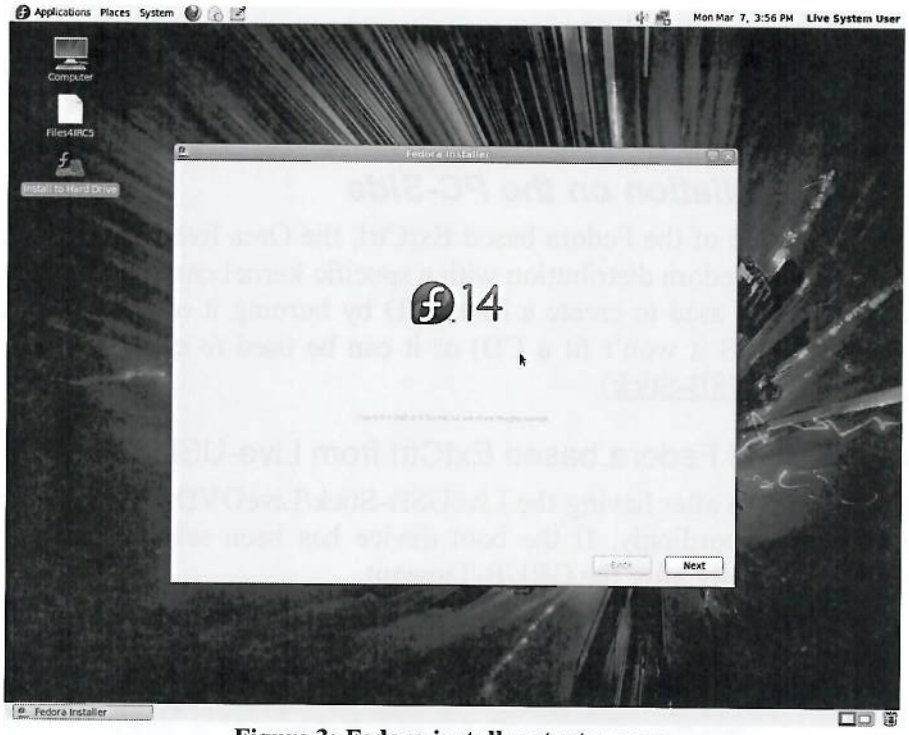


Figure 3: Fedora installer start screen.

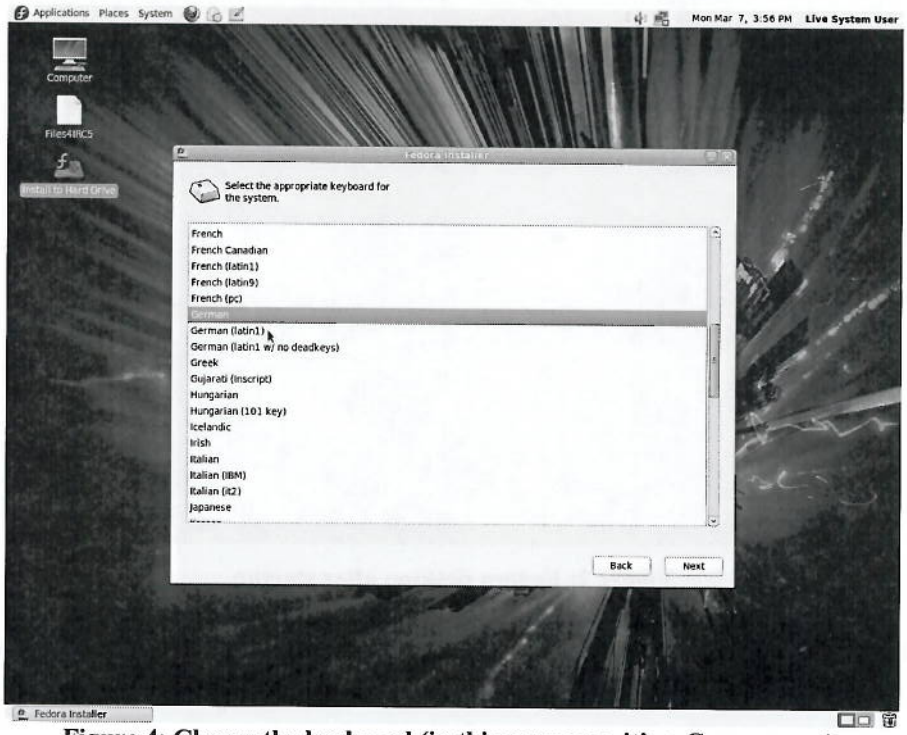


Figure 4: Choose the keyboard (in this very case, it's a German one).

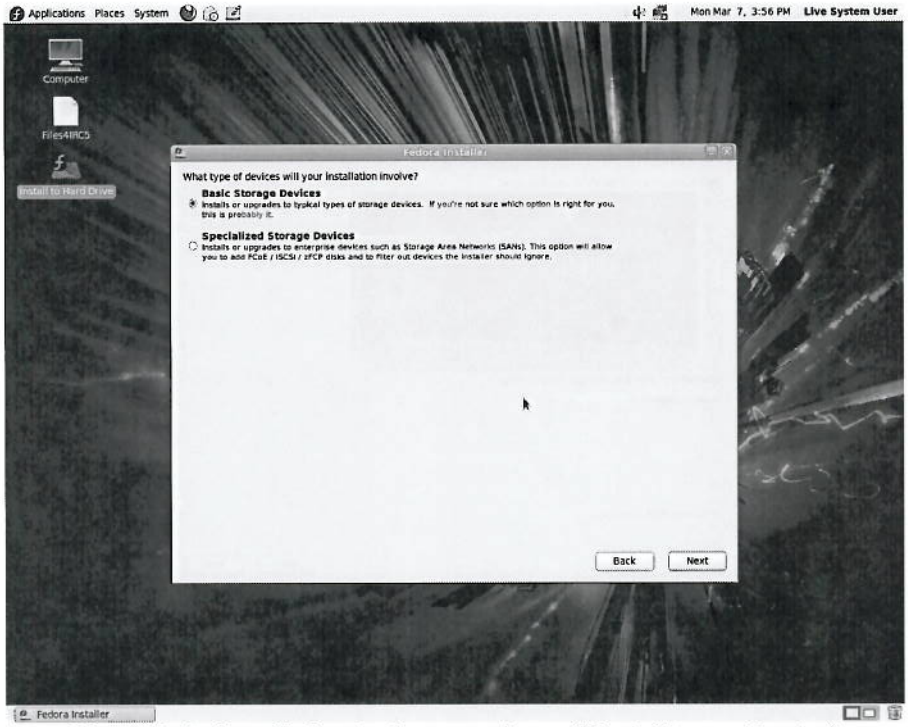


Figure 5: As installation device type, choose "Basic Storage Device".

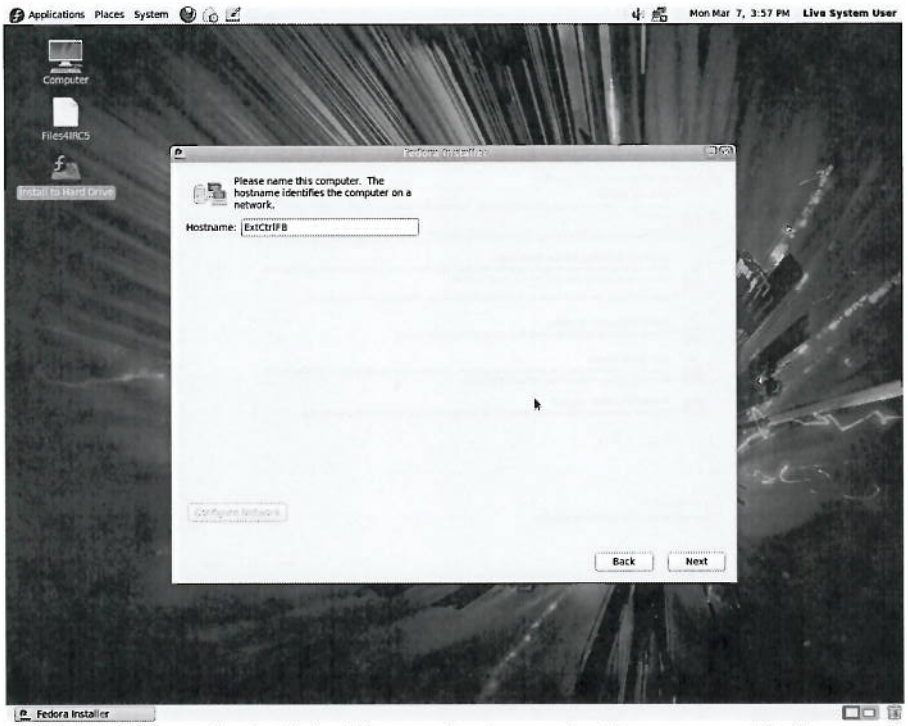


Figure 6: The installer will check for Storage devices and will come up with the question for a hostname (in this very case, it's "ExtCtrlFB").

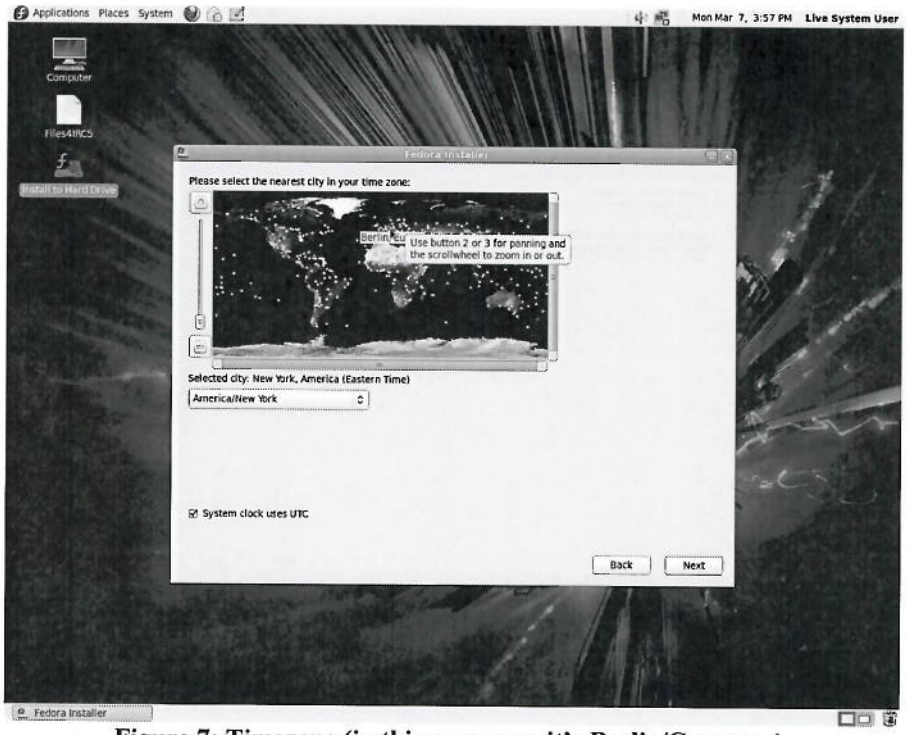


Figure 7: Timezone (in this very case, it's Berlin/Germany).

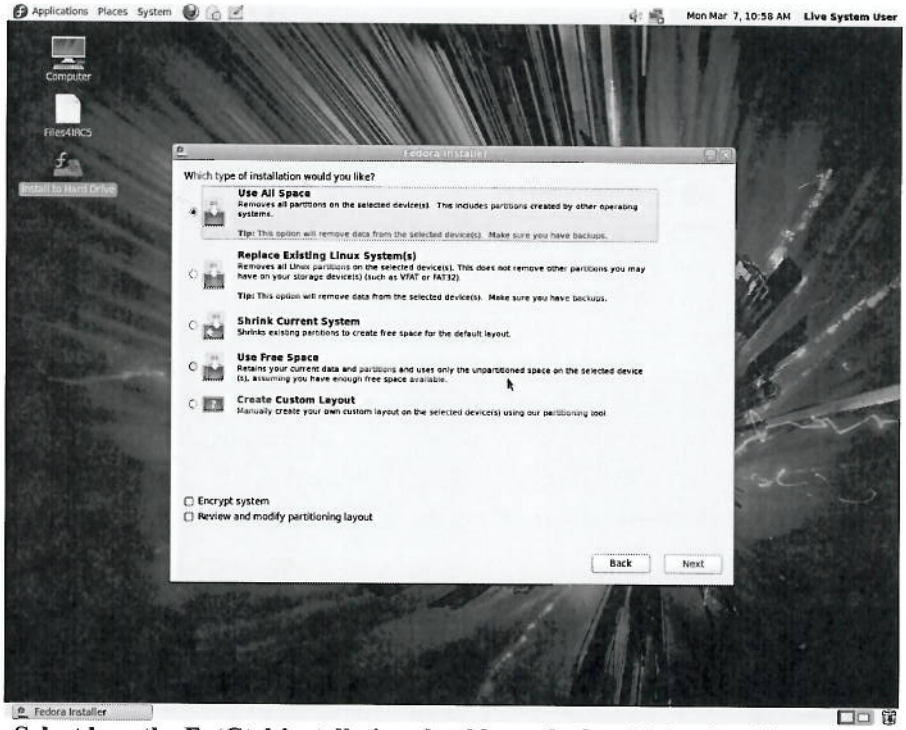


Figure 8: Select how the ExtCtrl-installation should use the harddrive (in this very case, "Use All Space" has been selected, because the Fedora-ExtCtrl-installation is the only OS on the computer).

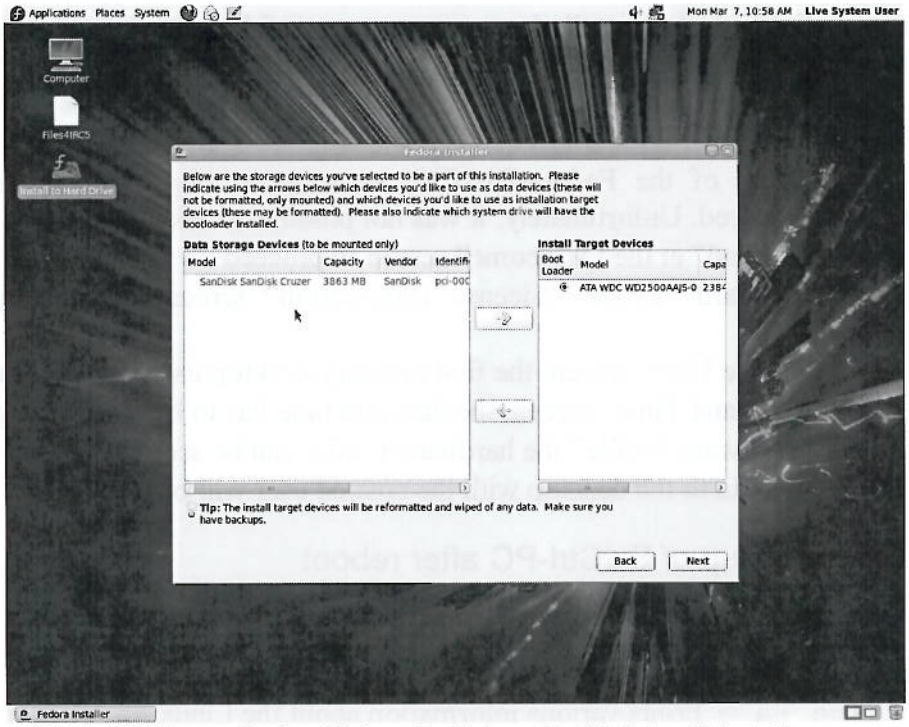


Figure 9: Choose the harddrive, the Fedora-ExtCtrl should install on.

Then, the root password needs to be chosen. This password is needed when system files are changed later. The root password should preferably not be the same as that for the ordinary user.

When the installation process is finished, the following screen appears.

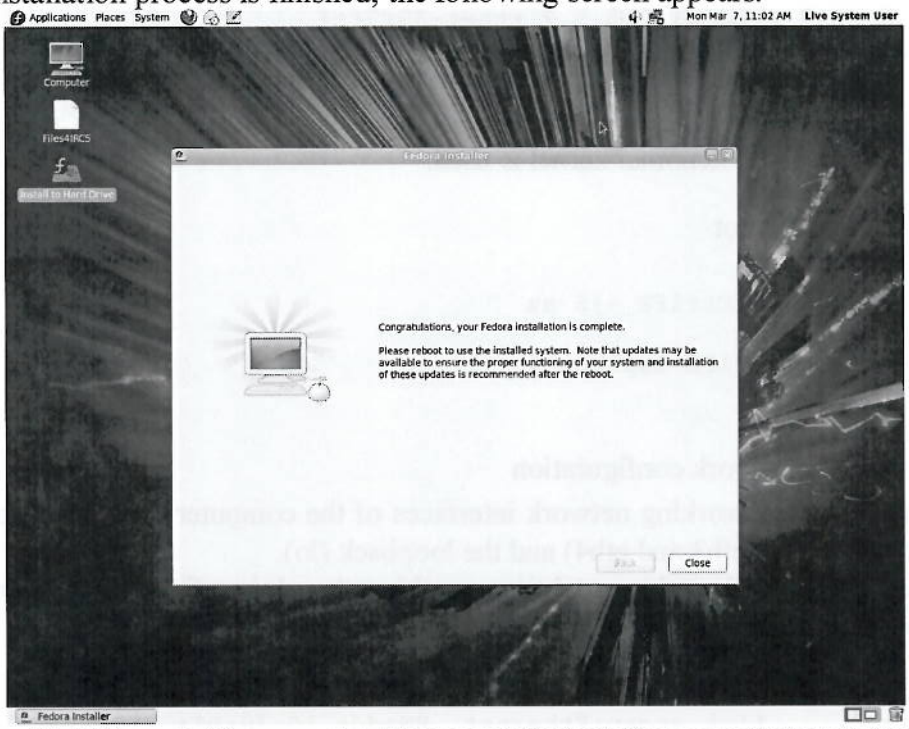


Figure 10: At the end, this screen should appear to indicate the successful installation.

After having finished the installation process successfully, the installation has to be completed by restarting the PC and booting the fresh installed Fedora-ExtCtrl.

6.1.2 Finish Fedora based ExtCtrl installation during reboot

During the booting of the Fedora-ExtrCtrl, some more steps for the setup of the installation are required. Unfortunately, it was not possible to take screenshots of that:

- Click "Forward" at the "Welcome" screen to proceed
- Click "Forward" at the "License Information" screen after having read the information
- At the "Create User" screen, the first ordinary desktop user has to be created
- At the "Date and Time" screen, the date and time has to be acknowledged
- At the "Hardware Profile" the hardware profile can be sent to Fedora
- Now, the login to the desktop with the created user will be possible

6.1.3 Further setup of ExtCtrl-PC after reboot

6.1.3.1 Check for right kernel

To check for the right kernel, the command line program uname is used.

- `uname -a` = Prints various information about the Linux installation
- `uname -r` = Prints the kernel name

```
[laplace@ExtCtrlFB ~]$ uname -a
Linux ExtCtrlFB 2.6.35.7_xenomai-2.5.5.2_rtnet-39f7fcf #1 SMP
Thu Feb 24 11:25:23 CET 2011 i686 i686 i386 GNU/Linux
[laplace@ExtCtrlFB ~]$ uname -r
2.6.35.7_xenomai-2.5.5.2_rtnet-39f7fcf
[laplace@ExtCtrlFB ~]$
```

`uname -r` returns `2.6.35.7_xenomai-2.5.5.2_rtnet-39f7fcf`, what proofs that the real time capable xenomai kernel is used.

6.1.3.2 Login as root

```
[laplace@ExtCtrlFB ~]$ su
Password:
[root@ExtCtrlFB laplace]#
```

6.1.3.3 Check network configuration

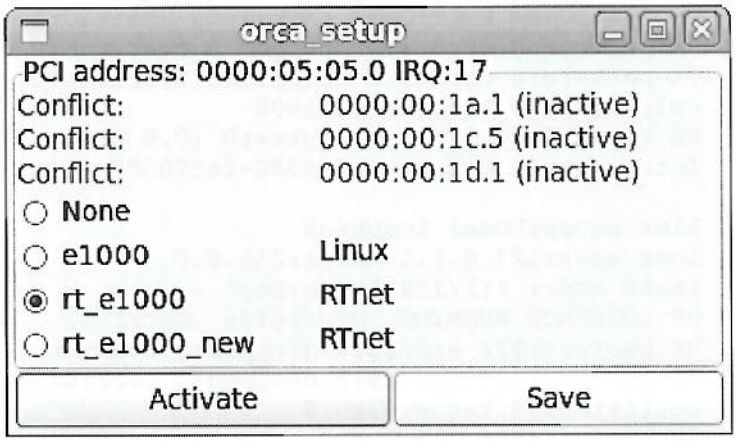
`ifconfig` lists all working network interfaces of the computer. This very computer has two network cards (eth3 and eth4) and the loopback (lo).
`rtifconfig` lists all working real time capable network interfaces of the computer. This very computer does not have any yet.

```
[root@ExtCtrlFB laplace]# ifconfig
eth3      Link encap:Ethernet  HWaddr 10:78:D2:C6:C7:5A
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
```

```
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:20 Memory:fe500000-fe520000

eth4      Link encap:Ethernet  HWaddr 00:1B:21:96:B1:CF
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:216 errors:0 dropped:0 overruns:0 frame:0
          TX packets:216 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:16288 (15.9 KiB)  TX bytes:16288 (15.9 KiB)

[root@ExtCtrlFB laplace]# rtifconfig
[root@ExtCtrlFB laplace]#
```

6.1.3.4 Run `orca_setup`

Open a new terminal, require root privileges by the command "`su`" and then issue the command "`orca_setup --setup`". A GUI similar to the one below will appear.



The GUI for selecting the appropriate network interface for realtime communication. Note that the latest version of the GUI has a button "Save" next to the button "Activate" if started with > orca_setup –setup.
The red conflicts are "only relevant" if you choose the realtime-properties for this network-card so choose another if you have the possibility (commands to help you determine which cards/drivers)
 # tail -f /var/log/messages
 # ifconfig
 # rtifconfig
$ ip addr
$ lspci

Depending on the number of network interfaces the computer has, the appearance of the GUI is different. Select one of the RTnet-capable interfaces for the realtime communication. Depending on the network card, different versions, such as `rt_e1000` or `rt_e1000_new`, have to be chosen for reasons of compatibility. Then click on activate. Note that some of the other PCI-interfaces might get inactivated when activating the RTnet-interface. This is indicated by the Conflict-entries printed in red in the GUI (but only relevant if this card is chosen for rt-interface!).

6.1.3.5 Saving the current configuration

By clicking on save in the GUI, a dialog window will appear. In this dialog window just click on "Save" again. In this way, the current configuration is saved and automatically initiated after a reboot of the computer. By this procedure, there is no need to activate the realtime-interface after each reboot manually.

6.1.3.6 Check for real time ethernet card

The ethernet card, which should be used for real time communication, should after activation of the RTnet-interface not be listed by `ifconfig` any more. Now, it should be listed by `rtifconfig`. In this very case, `rteth0` is the Ethernet device that will be used for real time communication with the robot later.

```
[root@ExtCtrlFB laplace]# ifconfig
eth3      Link encap:Ethernet  HWaddr 10:78:D2:C6:C7:5A
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:20 Memory:fe500000-fe520000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:224 errors:0 dropped:0 overruns:0 frame:0
          TX packets:224 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:16896 (16.5 KiB)  TX bytes:16896 (16.5 KiB)

[root@ExtCtrlFB laplace]# rtifconfig
     OR /usr/local/sbin/rtifconfig if NOT IN PATH
rteth0    Medium: Ethernet  Hardware address: 00:1B:21:96:B1:CF
          UP BROADCAST RUNNING  MTU: 1500
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

[root@ExtCtrlFB laplace]#
```

6.2 Wiring

6.2.1 RS232 serial port

The serial port of the ExtCtrl-PC needs to be connected via a nullmodem cable to the VxWorks console of the IRC5's main computer. Depending on whether the ExtCtrl-PC has a serial port or not, the serial port has to be addressed differently:

- Serial port: /dev/ttyS*
- USB to serial port adapter: /dev/ttyUSB*

The * is a wild card for a number. Depending on the number of ports, the number can be greater than 0.

6.2.1.1 Rights of /dev/tty*

It might be possible, that the user does not have the rights to write on the serial port. To be able to run ExtCtrl as ordinary user, the rights need to be changed:

```
[laplace@ExtCtrlFB ~]$ su
Password:
[root@ExtCtrlFB laplace]# chmod 0666 /dev/ttyS*
```

Respectively:

```
[root@ExtCtrlFB laplace]# chmod 0666 /dev/ttyUSB*
```

Since the /dev/ttyS* and /dev/ttyUSB* are owned by the `root` user and the `dialout` group, ordinary users will not have the rights to write on the serial port after every restart. The easiest way to fix this, is to add the user to the `dialout` group:

```
[root@ExtCtrlFB laplace]# usermod -a -G dialout laplace
```

where "laplace" has to be exchanged with the user name on the specific computer.

6.2.2 Real time Ethernet

The ExtCtrl-PC's real time capable Ethernet card needs to be connected to the same LAN as the IRC5 via the main computer's LAN-Port. NOTE: The connection has to be made directly to the LAN-port of the main computer. It is not possible to connect to the service port on the control cabinet. The easiest way to do this is by using an Ethernet crossover cable and connect them directly. In case the ExtCtrl-PC's real time Ethernet card supports Auto Crossover, on ordinary twisted pair Ethernet cable is sufficient.

6.3 Installing correct RobotWare-system

Install if needed RobotStudio. You may also need to install a separate RobotWare-version which supports ExtCtrl

Current version of RobotWare is 5.10.1037

(and the RobotStudio-version needs to be this or newer). We will later replace the mc/mc.sym in this robot system and it is important that the versions match.

6.3.1 Making copy/backup of an already running system

The easiest way to install a new system is to start by copying and subsequently modify an already installed system. Connect your computer to the service port of the IRC5-cabinet (should be a crossed network cable, even if most modern network cards also support a straight TCP-cable; use DHCP).

- Make a backup (stored on the cabinet) from the FlexPendant-menu
- Start RobotStudio and connect to IRC5 with "OneClickConnect"
- Making a backup of a current system in RobotStudio (stored locally on computer):
  - RobotStudio/Online-tab: Request write-access (right-click on Controller system-name)
  - Choose "Create backup" (right-click on Controller system-name) or the Backup-button in the main menu
- System Builder → "Create from Backup"
  - Browse for the folder with the backup you just made above
  - Give new name and step forward among the options until you can change the RobotWare-version to the one which supports ExtCtrl (see version above).
- Issue with keys if going from newer RW-version to 'older' extctrl-RW? – check!! (new (i.e., older) keys for extctrl needed from ABB...)

6.3.2 Installing the robot system

If possible, run the robot to HOME-position [0,0,0,0,0,0] (exact!) by GoHome-program before you change the system. Put the IRC5-controller in AUTO-mode and in RobotStudio->SystemBuilder mark the new system you just created and choose "Download to Controller".

If there is not enough free space left in the flash memory in the control cabinet, an error will be raised by RobotStudio. In this case, the old robot system has to be removed before the new one can be installed. The procedure is then as follows:

- Firstly, a boot media with the newly built robot system is created in RobotStudio on an USB-stick. Note that the IRC5 control cabinet only accepts certain types of USB-sticks.
- Connect the USB-stick in the USB-port on the control cabinet and restart the robot system using an X-start. This will erase the old system and install the newly created robot system from the USB-stick.

6.4 Installation Robot-Side

6.4.1 Set write access on the IRC5

To install ExtCtrl on the IRC5, the "mc" and the "mc.sym" need to be replaced on the controller. To be able to do this, the write access on the controller is required. Therefore a connection to the console of the controllers VxWorks is needed.

6.4.1.1 Terminal with MS Windows

Run terminal program via RS232 to the main-computer console of VxWorks using Putty with COM* on Windows: (9600 baud, 8 databits, 1 stop bit, parity none, Xon/Xoff). For finding active com-port, see Control panel → System properties → Hardware → ...). Also activate logging and save a Putty-profile.

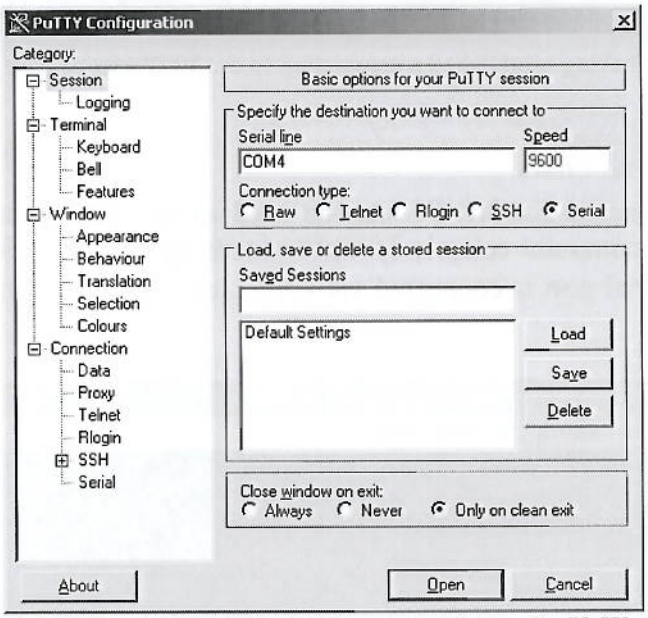


Figure 11: Settings window of Putty for setting up the connection to the VxWorks console running on the IRC5.

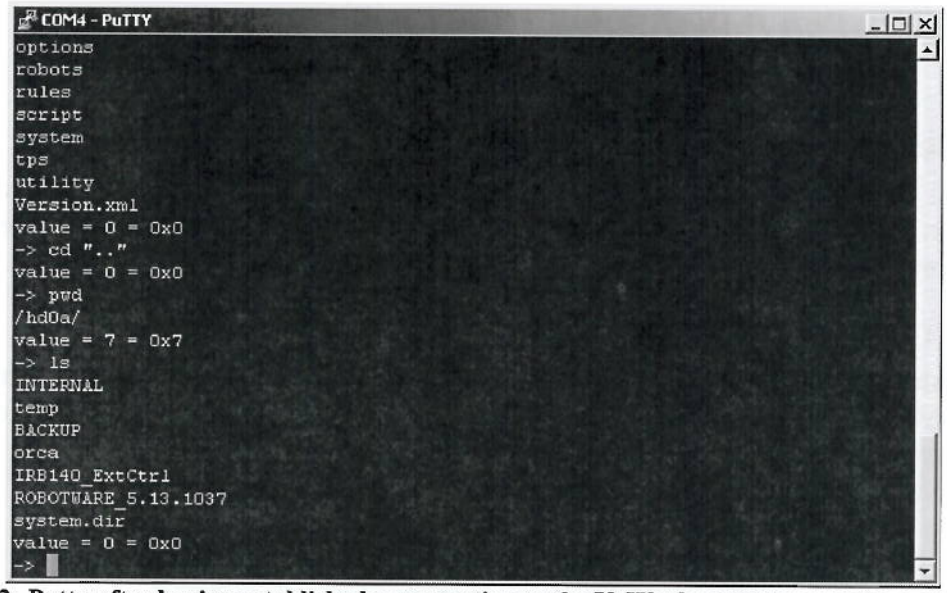


Figure 12: Putty after having established a connection to the VxWorks console running on the IRC5.

6.4.1.2 Terminal with Linux

Alternatively, the Fedora ExtCtrl PC can be used to establish a serial connection via RS232 to the main computer console of VxWorks using simcom with `/dev/ttyS*` (in the case, that the serial port is connected via USB, `/dev/ttyUSB*` must be used instead of `/dev/ttyS*`).

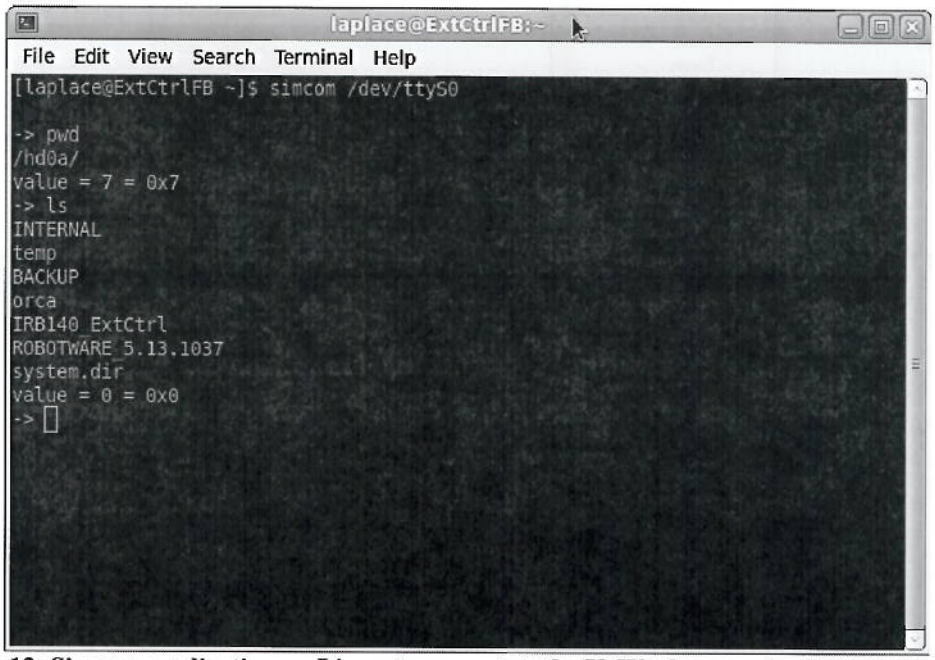


Figure 13: Simcom application on Linux to connect to the VxWorks console of the IRC5 via the physical serial port.

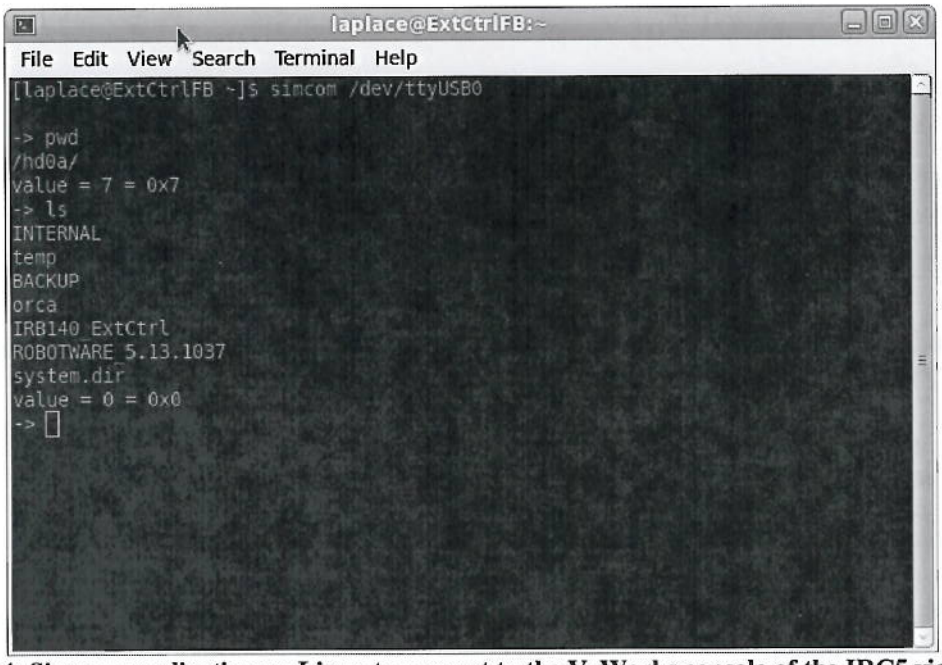


Figure 14: Simcom application on Linux to connect to the VxWorks console of the IRC5 via an USB to serial converter.

Pressing the Ctrl button and D at the same time terminates simcom. (DON'T write exit as this will exit the VxWorks...)

The "Backspace-button" may not work, but you can use CTRL+h instead.

6.4.1.3 VxWorks commands

The following steps are independent from the OS used on the computer that is used as terminal. After connection to the console

The VxWorks command prompt appears:

```
->
```

Command: cd and parameter: "/hd0a/" to change the directory to /hd0a/:

```
-> cd "/hd0a/"
value = 0 = 0x0
```

Command: pwd without parameters to check if the current working directory is really /hd0a/:

```
-> pwd
/hd0a/
value = 7 = 0x7
```

Command: ls without parameters (DON'T use "ls -l") to list content of directory to check for the ROBOTWARE directory:

```
-> ls
INTERNAL
temp
BACKUP
IRB140_ExtCtrl
ROBOTWARE_5.13.1037
system.dir
value = 0 = 0x0
```

Command: BootsupROreset and Robotware directory as parameter: "ROBOTWARE_5.13.1037" to change the write access of this directory:

```
-> BootsupROreset "ROBOTWARE_5.13.1037"
changing attributes on ROBOTWARE_5.13.1037/*
[...]
value = 0 = 0x0
```

After the reset of the read only of the Robotware directory, the VxWorks command prompt appears again:

```
->
```

6.4.2 Exchange mc and mc.sym

To upload the mc and mc.sym, which is specially build for ExtCtrl, to the IRC5, a connection to the IRC5's FTP-Server is required. If the computer is connected directly to the service port on the control cabinet, the ip-number to connect to is usually 192.168.125.1 if no other static ip-address has been selected. A copy of the original mc and mc.sym can be useful later:

```
Copy  /hd0a/ROBOTWARE_5.13.1037/bin/mc
To    /hd0a/ROBOTWARE_5.13.1037/bin/mc_ORIGINAL

Copy  /hd0a/ROBOTWARE_5.13.1037/bin/mc.sym
To    /hd0a/ROBOTWARE_5.13.1037/bin/mc.sym_ORIGINAL
```

Copy the mc and mc.sym which is specially built for ExtCtrl to the IRC5:

```
Copy  mc_extctrl_RW5.13.1037_20110311
To    /hd0a/ROBOTWARE_5.13.1037/bin/mc_extctrl_RW5.13.1
037_20110311
And   /hd0a/ROBOTWARE_5.13.1037/bin/mc

Copy  mc.sym_extctrl_RW1037_20110311
To    /hd0a/ROBOTWARE_5.13.1037/bin/mc.sym_extctrl_RW5.
13.1037_20110311
And   /hd0a/ROBOTWARE_5.13.1037/bin/mc.sym
```

Since the file names of the mc and mc.sym do not contain any information about if it is the original or the ExtCtrl one, it is good to have a copy of both available. This makes it easier to inspect which mc is running by comparing the file sizes.

6.4.3 Change install files

The important folders to do this are:

/hd0a/ROBOTWARE_5.13.1037/robots/irb1_140

and /hd0a/ROBOTWARE_5.13.1037/robots/irb1_140/irbcfg/

6.4.3.1 Identify robot type

To be able to activate the ExtCtrl after having changed the mc, the robot type has to be identified precisely.

If there are uncertainties about which robot type it is, an echo-printout can be added after each label in "/hd0a/ROBOTWARE_5.13.1037/robots/irb1_140/install.cmd", which will be printed after performing an I-Start.

In Lund:

```
#IRB140_Y
echo -text "Installing extctrl-version 2011-03-11"
config -filename $BOOTPATH/irbcfg/SEC_140_0.81_5_TYPEB_1.cfg.enc -domain MOC -internal
config -filename $BOOTPATH/irbcfg/INT_140_0.81_5_TYPEB_1.cfg -domain MOC -internal
config -filename $BOOTPATH/irbcfg/EXT_140_0.81_5_TYPEB_1.cfg -domain MOC
getkey -id "PSCOption" -var 10 -strvar $DUMMY -errlabel END_LABEL
copy -from $BOOTPATH/irbcfg/PSC_140_0.81_5_TYPEB_1.sxml -to $OPTIONS/PSC/psc_irobot_1.sxml
goto -label END_LABEL
```

In Freidberg:

```
#IRB140
config -filename  $BOOTPATH/irbcfg/SEC_140_0.81_5_TYPEA_1.cfg.enc -domain MOC -internal
config -filename  $BOOTPATH/irbcfg/INT_140_0.81_5_TYPEA_1.cfg -domain MOC -internal
config -filename  $BOOTPATH/irbcfg/EXT_140_0.81_5_TYPEA_1.cfg -domain MOC
getkey -id "PSCOption" -var 10 -strvar $DUMMY -errlabel END_LABEL
copy -from $BOOTPATH/irbcfg/PSC_140_0.81_5_TYPEA_1.sxml -to $OPTIONS/PSC/psc_irobot_1.sxml
goto -label END_LABEL
```

In Darmstadt:

```
#IRB140T-YH6
config -filename $BOOTPATH/irbcfg/SEC_140T_0.81_6_1.cfg.enc -domain MOC -internal
config -filename $BOOTPATH/irbcfg/INT_140T_0.81_6_1.cfg -domain MOC -internal
config -filename $BOOTPATH/irbcfg/EXT_140T_0.81_6_1.cfg -domain MOC
getkey -id "PSCOption" -var 10 -strvar $DUMMY -errlabel END_LABEL
copy -from $BOOTPATH/irbcfg/PSC_140T_0.81_6_1.sxml -to $OPTIONS/PSC/psc_irobot_1.sxml
goto -label END_LABEL
```

6.4.3.2 Activate ExtCtrl

Depending on the robot type, the ExtCtrl needs to be activated in the according EXT_*.cfg-file by adding " -use_ilc "ext_ctrl" " in the ROBOT-entry.

Lund:

/hd0a/RobotWare_5.13.1037/robots/irb1_140/irbcfg/EXT_140_0.81_5_TYPEB_1.cfg

Friedberg:
/hd0a/RobotWare_5.13.1037/robots/irb1_140/irbcfg/EXT_140_0.81_5_TYPEA_1.cfg

Fiedberg:
/hd0a/RobotWare_5.13.1037/robots/irb1_140/irbcfg/EXT_140T_0.81_6_1.cfg.cfg

Edit:

```
MOC:CFG_1.0::
[...]
ROBOT:
-name "ROB_1" -use_robot_serial_number "rob_1" \
-use_robot_type "ROB1_MIA_0.81_5" -use_robot_calib "r1_uncalib" \
-use_joint_0 "rob1_1" -use_joint_1 "rob1_2" -use_joint_2 "rob1_3" \
-use_joint_3 "rob1_4" -use_joint_4 "rob1_5" -use_joint_5 "rob1_6" \
-gravity_beta 0 -base_mass 14 -base_mass_centre_x -0.0668 \
-base_mass_centre_y 0.0016 -base_mass_centre_z 0.0582 -use_lin_jog "ROB1" \
-use_reorient_jog "ROB1" -use_joint_jog "ROB1" -use_kinematic_corvec \
-use_ilc "ext_ctrl"

ROBOT_SERIAL_NUMBER:
[...]
```

6.4.3.3

!!! You need to have activated with **`-use_ilc "ext_ctrl" before you continue.`**

Choose feedforward mode for better tracking with extctrl

Normally a standard ABB-robot runds with trq-feedforward (ffw_mode 2). If you are in ffw_mode =2 the system will automatically reset it to ffw_mode=0 if -use_ilc="ext_ctrl" is activated, which severely will degrade the tracking performance for ordinary RAPID-commands!! (Delibarate choice from ABB to protect against "reverse engineering" of ffw-component) A remedy to get better (but still reduced) tracking performance is to put the system in ffw_mode=1 (i.e., keeping a speed feedforward component).

To get the parameters for the cfg-file:
Change manually of via RobotStudio to Speed feedforward
ABB->Control Panel->Configuration-> Topics=Motion -> Lag Control Master 0
rob1_1 -> FFW Mode : Spd (corresponds to -ffw_mode 1)

Change to "Spd" and restart system, then make a backup of system and read from BACKUP-folder to get the parameters for LCM0:

less SYSPAR/MOC.cfg
...
#

LCM0:

-name "rob1_1" -Kp 10 -Kv 0.1 -Ti 0.1 -ffw_mode 1

-name "rob1_2" -Kp 10 -Kv 0.1 -Ti 0.1

-name "rob1_3" -Kp 10 -Kv 0.03 -Ti 0.1

-name "rob1_4" -Kp 10 -Kv 0.01 -Ti 0.1

-name "rob1_5" -Kp 10 -Kv 0.01 -Ti 0.1

-name "rob1_6" -Kp 10 -Kv 0.01 -Ti 0.1

#

Include this in your configuration file (with all robX_1-entries
changed to have -ffw_mode 1
).

TODO: What about going back with deactivation-script??? Should it have entry for ffw_mode 2???

6.4.3.4 Brake timer

By default, the robot system will lock the brakes if the ABB-robot is not jogged for 120 seconds. This is of course not desirable when controlling the IRB with extctrl. The limit value can however be changed to a maximum 3600000 seconds to avoid locking the brakes due to inactivity on the Teach Pendant. This can be done directly in the Teach Pendant. Tap the ABB logo in the top left corner to enter the menu. Tap the Control Panel and thereafter Configuration. Tap Topics and choose Motion. Scroll down until the category Motion Planner is found and enter it by selecting it and then tapping Show All. There are six different subcategories called motion_planner_1 through motion_planner_6 which represent the different robot joints. Go through each joint and change the parameter called Brake On Time.

- An alternative is to install the software Robot Studio and to connect to

the robot system via the network port. In Robot Studio, write access must be claimed (which must be granted on the Teach Pendant). Then by opening the tab corresponding to the menu above, the time limits can be set to maximum for each joint.

OR

- change in the corresponding EXT_ROBOTNAME.cfg-file

(example
/hd0a/ROBOTWARE_5.13.1037/robots/irb1_120/irbcfg/EXT_120_0.58_3_1.cfg)

- MOTION_PLANNER:
  - -name "motion_planner_1" -brake_on_timeout 3.6E+06\ -std_servo_queue_time ... -name "motion_planner_2" -brake_on_timeout 3.6E+06
  - ...

(Probably only need to do this for motion_planner_1)

6.4.3.5 Configuration-file for extctrl, ffwmode and brake_timer

It may be convenient to have a separate file with only the entries described above (for ROBOT:, LCM0: and MOTION_PLANNER:) One should also have one file without use_ilc, to turn off the extctrl-functionality and recover full tracking performance with the torque feedforward).

Create e.g. (change 140 below to your robot model)
/hd0a/ROBOTWARE_5.13.1037/robots/irb1_140/irbcfg/Activate_extctrl_ffwmode_1.cfg
?????
/hd0a/ROBOTWARE_5.13.1037/robots/irb1_140/irbcfg/Dectivate_extctrl.cfg
from the corresponding
/hd0a/ROBOTWARE_5.13.1037/robots/irb1_140/irbcfg/EXT_*.cfg

To load Activate_extctrl_ffwmode_1.cfg
FlexPendant: ABB -> Control Panel -> Configuration -> File
"Load Parameters ..." -> "Load parameters and replace duplicates"

/hd0a/ROBOTWARE_5.13.1037/robots/irb1_140/irbcfg/

You then get a question about "Restart after having loaded parameters" and you need to do that before extctrl-functionality is activated/deactivated.

6.4.4 Perform I-Start

The I-Start can either be performed using the FlexPendant or the VxWorks console.
FlexPendant: ABB -> Restart -> Advanced… -> I-Start -> OK
VxWorks console: restart_ctrl 3

6.4.5 Copy ORCA files to the robot

In addition to the new mc, there are some ORCA programs needed on the IRC5 to be able to run an external control. These programs can be found on the Fedora ExtCtrl PC in the folder /opt/robot/abb/irc5_main/ and must be copied to /hd0a/orca/ on the IRC5:

Copy /opt/robot/abb/irc5_main/extctrl_show_config.main
To /hd0a/orca/extctrl_show_config.main

Copy /opt/robot/abb/irc5_main/lth_net_extctrl_20110428.main
To /hd0a/orca/lth_net_extctrl.main

Copy /opt/robot/abb/irc5_main/lth_net.main
To /hd0a/orca/lth_net.main

In addition you need to copy the file robotdata

To /hd0a/orca/robotdata

The robotdata-file contains a description of which axes are sent

Use 101..104 for numbering upp to 4 robots (36 axes) in a moltimove-config...

6-DOF standard ABB/IRC robot
cat robotdata

0000003f 0000000 00000000 00000000 103

MultiMove (two 6DOF ABB/IRB-robots with one external axis each)
cat robotdata

0000003f 00010000 00000000 00000000 101

00000fc0 00020000 00000000 00000000 102

These patterns should be shown if ExtCtrl is activated when executing
extctrl_show_config.main

- - > cd "/hd0a/orca"
- - > ld < extctrl_show_config.main
- - > lkup "extctrl"
- - > extctrl_show_config

Samples: 16
0 submit_active [0]: 3f 10000 0 0
1 submit [2]: 3f 10000 0 0

```
2 submit_active [23]: fc0 20000 0 0
3 submit       [24]: fc0 20000 0 0
4 obtain_active [43]: 3f 10000 0 0
5 obtain_active [43]: fc0 20000 0 0
6 submit_active [4029]: 3f 10000 0 0
7 submit       [4030]: 3f 10000 0 0
8 submit_active [4057]: fc0 20000 0 0
9 submit       [4058]: fc0 20000 0 0
10 obtain_active [4076]: 3f 10000 0 0
11 obtain_active [4076]: fc0 20000 0 0
12 submit_active [8073]: 3f 10000 0 0
13 submit       [8074]: 3f 10000 0 0
14 submit_active [8098]: fc0 20000 0 0
15 submit       [8098]: fc0 20000 0 0
value = 0 = 0x0
```

Troubleshooting: If the pattern from the robotdata-file is not shown, probably the
-use_ilc "ext_ctrl"
has not been activated and no data from the IRC5-system will be exported.

OBS!! error message

Unhandled State 2: unsync
-- most probably problem with network card in earlier versions and make sure that the simulink model is built with latest version (presently rtw2orca_irc5_main_20120112_xeno)

7 Build a controller for ExtCtrl

7.1 MATLAB/Simulink

7.1.1 Installation and setup

The MATLAB installation needs to contain **Simulink** and the **Simulink Real-Time Workshop** to be able to generate controllers for ExtCtrl from a Simulink model. If those Toolboxes are available, the installation is pretty much straight forward.

In case the MATLAB installation files are present as a CD, it can just be placed in the CD drive and will be mounted automatically to the file system:

```
[laplace@ExtCtrlFB ~]$ cd /media/R2009b_UNIX
```

In case the MATLAB installation files are present as an *.iso file, it can easily be mounted to the file system:

```
[laplace@ExtCtrlFB ~]$ sudo mkdir /mnt/cdrom
[laplace@ExtCtrlFB ~]$ sudo mount -o loop,ro
/home/laplace/R2009b_UNIX.iso /mnt/cdrom
[laplace@ExtCtrlFB ~]$ cd /mnt/cdrom
```

After having changed the directory to the mounting point of the MATLAB installation files, the installation can be started as root by executing the installer:

```
[laplace@ExtCtrlFB /media/R2009b_UNIX]$ sudo ./install
```

Respectively:

```
[laplace@ExtCtrlFB /mnt/cdrom]$ sudo ./install
```

MATLAB can be installed to its default suggestion. In this very installation, it has been installed to the folder "/opt/matlab/", which might not be the best solution in case that several MATLAB version should be installed.

Now, MATLAB can be started from the command line by:

```
[laplace@ExtCtrlFB ~]$ /opt/matlab/bin/matlab
```

After the installation, the executable MATLAB files will not be part of the path right away. The easiest way to fix that manually is by generating a symbolic link to the executables in the "/usr/bin" folder. Then, the absolute path does not need to be specified

```
[laplace@ExtCtrlFB ~]$ sudo ln -s /opt/matlab/bin/matlab
/usr/bin/matlab
[laplace@ExtCtrlFB ~]$ sudo ln -s /opt/matlab/bin/mex
/usr/bin/mex
[laplace@ExtCtrlFB ~]$ matlab
```

To be able to build controllers for ExtCtrl from a Simulink model using the Real-Time Workshop, some robot specific files are needed. In case a robot is used which is not part of the ExtCtrl installation yet, it needs to be declared in the file "/opt/robot/matlab/irb/include/robotdata.h". Otherwise it is sufficient to run "make" in the folder "/opt/robot/matlab/irb/mex":

```
[laplace@ExtCtrlFB /opt/robot/matlab/irb/mex]$ sudo make
```

While running "make", a folder for each robot type is made with all the required MATLAB files. This folder needs to be specified in the Simulink model later on, to refer to the right robot while building a controller.

@TODO: clarify remaining steps
Copy /home/robot/
Needed: /opt/robot/labcomm/lth_net_defs.h
Scp log und readlog.m

7.1.2 Build a controller from MATLAB/Simulink

@TODO

7.2 Scilab/XCos

7.2.1 Installation and setup

@TODO

7.2.2 Build a controller from Scilab/XCos

@TODO

7.3 Plain C-Code

7.3.1 Installation and setup

@TODO

7.3.2 Build a controller from plain C-Code

@TODO

8 Run ExtCtrl (Test ExtCtrl installation)

8.1 Booting the system

IMPORTANT: The real time ethernet drivers need to be activated every time the PC is restarted if the configuration has not previously been saved with the `orca_setup` GUI. (use `orca_setup -- setup to get svae-option in GUI.`)

8.2 Run Opcom

To run the test application, the following command needs to be executed on the commandline of the ExtCtrl-PC:

```
[laplace@ExtCtrlFB ~]$ /opt/robot/example/opcom/lth_net_xeno -v
/dev/ttyS0 localhost 2000
```

By the parameter `-v /dev/ttyS0`, the serial port of the ExtCtrl-PC is specified. This parameter needs to be adjusted in case a USB to serial adapter is used.
@TODO: Write text describing the actions while running opcom

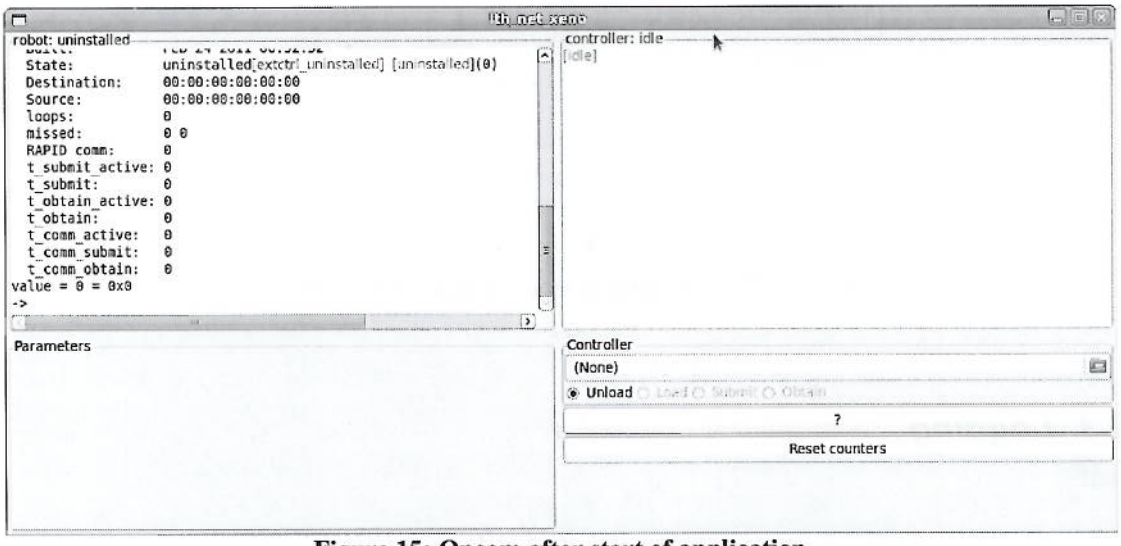


Figure 15: Opcom after start of application.

After

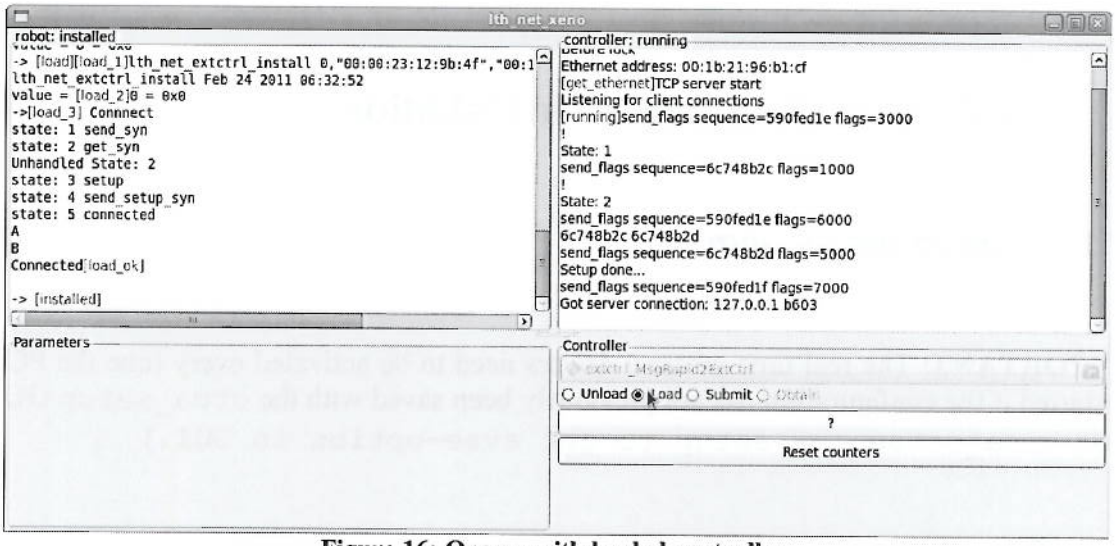


Figure 16: Opcom with loaded controller.

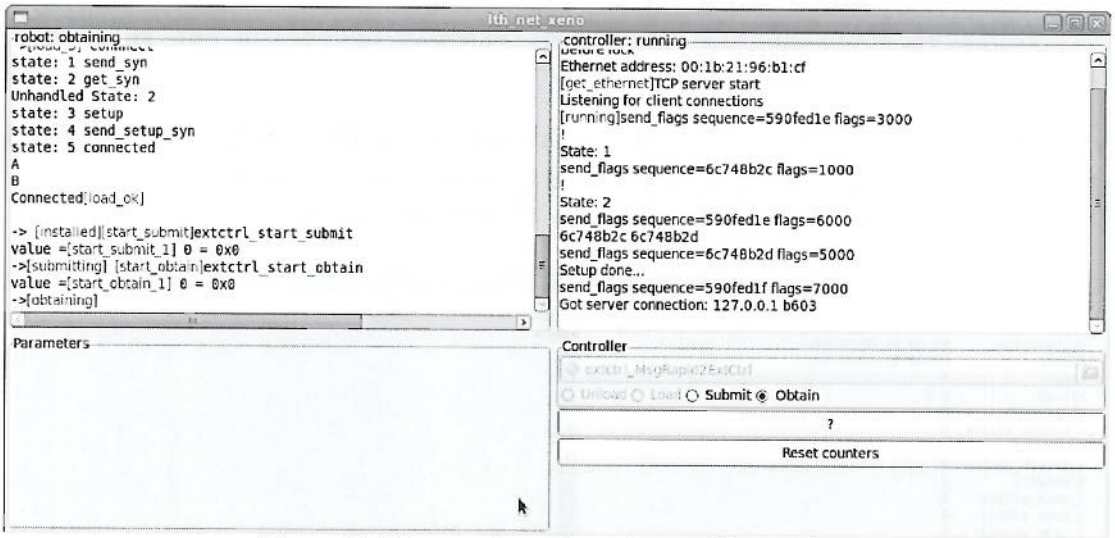


Figure 17: Opcom when external control is running.

8.3 Logging

asdf

8.4 Switch between ExtCtrl and normal operation

I-Start is required

Parameter store and restore using RobotStudio and warmstart is not sufficient.

Alternative:
Load Activate_extctrl.cfg/Deactivate_extctrl.cfg from FlexPendant:

FlexPendant: ABB -> Control Panel -> Configuration -> File
"Load Parameters ..." -> "Load parameters and replace duplicates"

/hd0a/ROBOTWARE_5.13.1037/robots/irb1_140/irbcfg/

You then get a question about "Restart after having loaded parameters" and you need to do that before extctrl-functionality is activated/deactivated.

9 Appendix

9.1 Some useful Linux commands

CMD	Explanation
ls	Lists the content of a directory
cd	Changes the working directory
pwd	Print the full filename of the current working directory.
uname	Print certain system information.
ifconfig	
rtifconfig	
su	

9.2 Some useful VxWorks commands

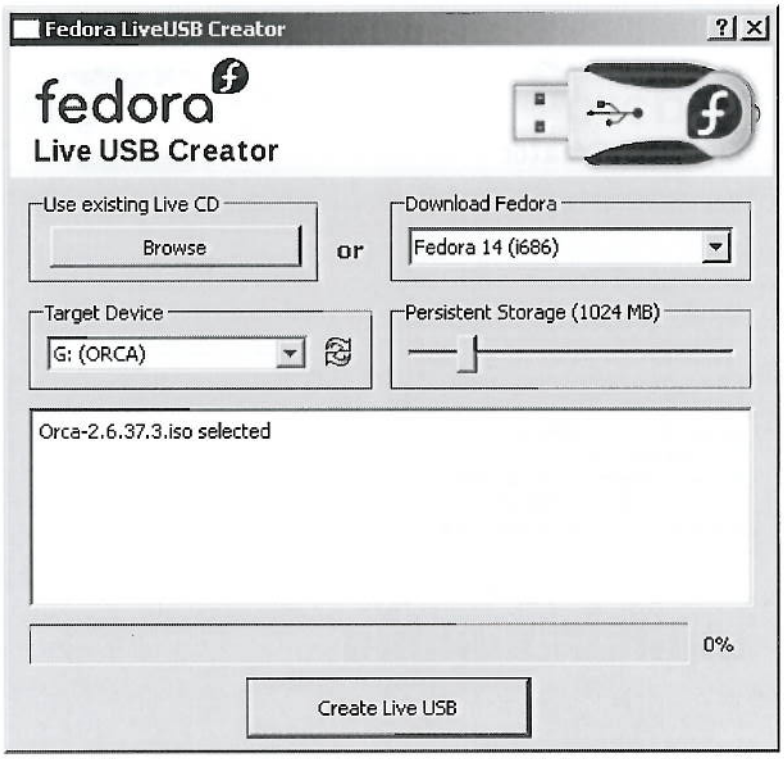
CMD	Explanation
ls	Lists the content of a directory
cd	Changes the working directory
pwd	Print the full filename of the current working directory.
Rename "old","new"	

9.3 How to create a LiveUSB-Stick

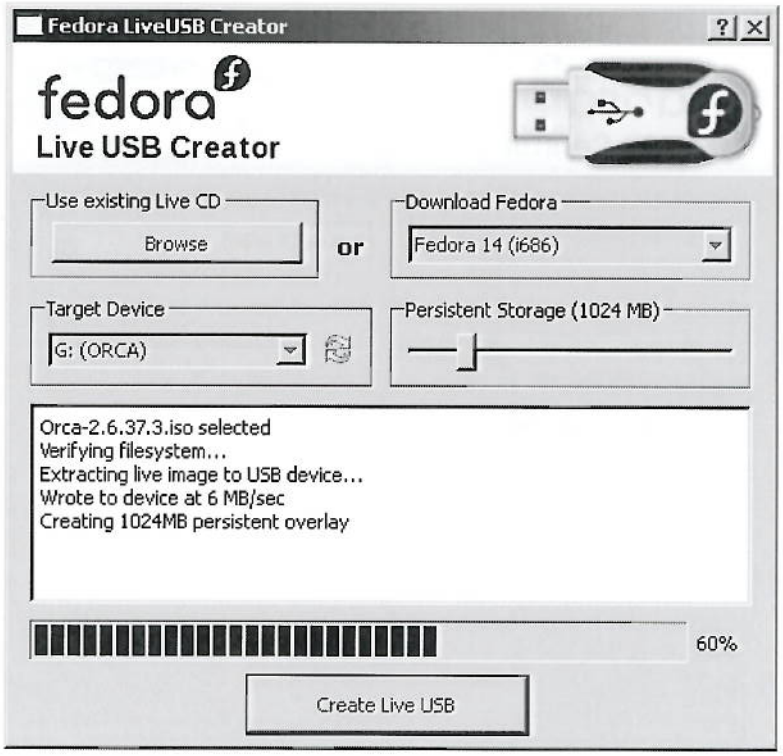
@TODO: Sort and rewrite. Subtitle for pictures
For creating a LiveUSB-Stick, the Fedora-Wiki provides very good help and hints:
https://fedoraproject.org/wiki/How_to_create_and_use_Live_USB

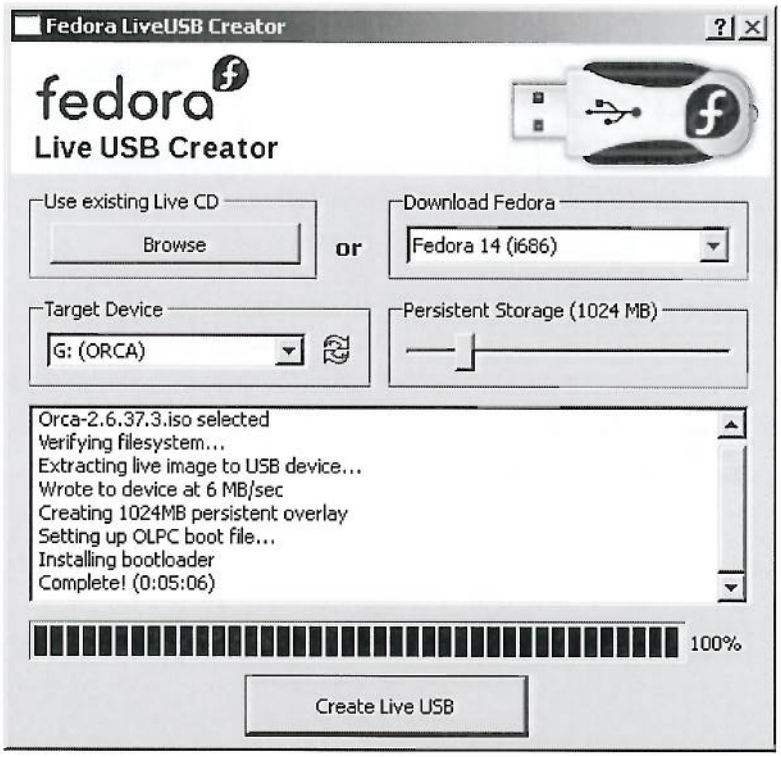
9.3.1 On Windows

https://fedorahosted.org/liveusb-creator/


Fedora LiveUSB Creator
fedora
Live USB Creator
Use existing Live CD
Browse
or
Download Fedora
Fedora 14 (i686)
Target Device
G: (ORCA)
Persistent Storage (1024 MB)
Orca-2.6.37.3.iso selected
0%
Create Live USB


Fedora LiveUSB Creator
fedora
Live USB Creator
Use existing Live CD
Browse
or
Download Fedora
Fedora 14 (i686)
Target Device
G: (ORCA)
Persistent Storage (1024 MB)
Orca-2.6.37.3.iso selected
Verifying filesystem...
Extracting live image to USB device...
0%
Create Live USB


Fedora LiveUSB Creator
fedora
Live USB Creator
Use existing Live CD
Browse
or
Download Fedora
Fedora 14 (i686)
Target Device
G: (ORCA)
Persistent Storage (1024 MB)
Orca-2.6.37.3.iso selected
Verifying filesystem...
Extracting live image to USB device...
Wrote to device at 6 MB/sec
Creating 1024MB persistent overlay
60%
Create Live USB


Fedora LiveUSB Creator
fedora
Live USB Creator
Use existing Live CD
Browse
or
Download Fedora
Fedora 14 (i686)
Target Device
G: (ORCA)
Persistent Storage (1024 MB)
Orca-2.6.37.3.iso selected
Verifying filesystem...
Extracting live image to USB device...
Wrote to device at 6 MB/sec
Creating 1024MB persistent overlay
Setting up OLPC boot file...
Installing bootloader
Complete! (0:05:06)
100%
Create Live USB

9.3.2 On Linux

9.3.2.1 Fedora

In case, a Fedora installation (not necessarily Fedora based ExtCtrl, ordinary Fedora installation is sufficient) is available and a LiveUSB-Stick with a persistent partition is needed:

```
[laplace@ExtCtrlDA ~]$ sudo yum install livecd-tools
[sudo] password for laplace:
[...]
[laplace@ExtCtrlDA ~]$ sudo livecd-iso-to-disk --format --reset-
mbr --overlay-size-mb 512 Orca-2.6.37.3.iso /dev/sdb1
Verifying image...
/home/laplace/Orca-2.6.37.3.iso:
78b6e24080462fcd10ebb6a18c4fb554
Fragment sums:
6b48c3b1a543f84ba9f38d334ed7cda629853fa8ccd15e15c3ec78a588ce
Fragment count: 20
Press [Esc] to abort check.
Checking: 100.0%

The media check is complete, the result is: PASS.

It is OK to use this media.
WARNING: THIS WILL DESTROY ANY DATA ON /dev/sdb!!!
Press Enter to continue or ctrl-c to abort

Waiting for devices to settle...
mke2fs 1.41.12 (17-May-2010)
Filesystem label=LIVE
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
490560 inodes, 1959424 blocks
97971 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2009071616
60 block groups
32768 blocks per group, 32768 fragments per group
8176 inodes per group
Superblock backups stored on blocks:
 32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 37 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to
override.
Copying live image to USB stick
progress 65536/1292832768 (64.0 KB/s)
[...]
progress 1292832768/1292832768 (13.4 MB/s)
```

```
progress 16384/16384 (16.0 KB/s)
progress 16384/16384 (16.0 KB/s)
Updating boot config file
Initializing persistent overlay file
0+0 records in
0+0 records out
0 bytes (0 B) copied, 2.1355e-05 s, 0.0 kB/s
Installing boot loader
/media/usbdev.B1JrC8/syslinux is device /dev/sdb1
USB stick set up as live image!
[laplace @ ExtCtrlDA ~]$
```

9.3.2.2 Any other Linux or Unix distribution

The easiest way to bring an iso-file on a USB-Stick, is by using the command line tool dd, which is present in almost every Linux or Unix installation:

```
[laplace @ ExtCtrlDA ~]$ sudo dd if=Orca-2.7.37.3.iso
of=/dev/sdb bs=8M
[sudo] password for laplace:
155+1 records in
155+1 records out
1305477120 bytes (1.3 GB) copied, 113,038 s, 11.5MB/s
[laplace @ ExtCtrlDA ~]$
```

9.4 Some Experiments

9.4.1 Gear ratios IRB140 Type C

@TODO: Rewrite

```
After I got ExtCtrl to work at my University in Darmstadt, I installed
MATLAB and
Simulink (with
RTW) to be able to build controllers from Simulink models myself. It
didn't take
much time until
everything worked after I scp some stuff from Lund. But then I realized,
that
something is not
working properly.
When I apply a step function (with a second order transfer function, to
make the
behavior
smoother) to all six axes, axis 1 to 4 behave normally. But the axes 5
and 6 behave
strangely.
I did some experiments to identify the problem, with and without the
"Arm2Motor/Motor2Arm"
Simulink blocks.
The following tables with two times 8 measurements show the measured
joint angles
and the
according input values. The "X" means, that I applied a step to this
axis, e.g.: [ X
```

```
|   |   ] ->
[ 0.19818 | 0.00000 | 0.00000 ] means, that I applied a step or 0.2rad
only to axis
4 and the
measured joint angle of axis 4 is 0.19818rad. The measured joint angle
of axis 5 and
6 are zero.

Using Arm2Motor and Motor2Arm [Input: desired value of 0.2rad]:
[ 4 | 5 | 6 ] -> [ A4/rad  | A5/rad  | A6/rad  ]
[   |   |   ] -> [ 0.00000 | 0.00000 | 0.00000 ]
[   |   | X ] -> [ 0.00000 | 0.00000 | 0.19072 ]
[   | X |   ] -> [ 0.00000 |-0.18987 |-0.08845 ]
[   | X | X ] -> [ 0.00000 |-0.18987 | 0.10227 ]
[ X |   |   ] -> [ 0.19818 | 0.00000 | 0.00000 ]
[ X |   | X ] -> [ 0.19818 | 0.00000 | 0.19060 ]
[ X | X |   ] -> [ 0.19818 |-0.18979 |-0.08857 ]
[ X | X | X ] -> [ 0.19818 |-0.18979 | 0.10214 ]

Direct (without using Arm2Motor and Motor2Arm)[Input: desired value of
10.0rad
(motor angle)]:
[ 4 | 5 | 6 ] -> [ A4/rad  | A5/rad  | A6/rad  ]
[   |   |   ] -> [ 0.00000 | 0.00000 | 0.00000 ]
[   |   | X ] -> [ 0.00000 | 0.00000 |-0.31250 ]
[   | X |   ] -> [ 0.00000 | 0.29166 |-0.14583 ]
[   | X | X ] -> [ 0.00000 | 0.29166 |-0.45833 ]
[ X |   |   ] -> [-0.32787 | 0.01543 |-0.02262 ]
[ X |   | X ] -> [-0.32787 | 0.01543 |-0.33512 ]
[ X | X |   ] -> [-0.32787 | 0.30709 |-0.16845 ]
[ X | X | X ] -> [-0.32787 | 0.30709 |-0.48095 ]

I think that those errors are due to the reason, that we have a
different type of
the IRB140 here
in Darmstadt. We have an IRB140 Type C with 6kg payload and this one
might have
different gear
ratios than you IRB140 Type B in Lund and that's why the
"Arm2Motor/Motor2Arm"
Simulink blocks
don't work properly.
```

9.4.2 Gear ratios IRB140 Type A

Using Arm2Motor and Motor2Arm [Input: 0.2rad]:
[4 | 5 | 6] -> [A4/rad | A5/rad | A6/rad]
[| |] -> [0.00000 | 0.00000 | 0.00000]
[| | X] -> [0.00000 | 0.00000 | 0.20000]
[| X |] -> [0.00000 | 0.20000 | 0.00000]
[| X | X] -> [0.00000 | 0.20000 | 0.20000]
[X | |] -> [0.20000 | 0.00000 | 0.00000]

[X | | X] -> [0.20000 | 0.00000 | 0.20000]
[X | X |] -> [0.20000 | 0.20000 | 0.20000]
[X | X | X] -> [0.20000 | 0.20000 | 0.20000]

Direct (without using Arm2Motor and Motor2Arm)[Input: 10.0rad]:
[4 | 5 | 6] -> [A4/rad | A5/rad | A6/rad]
[| |] -> [0.00000 | 0.00000 | 0.00000]
[| | X] -> [0.00000 | 0.00000 |-0.32770]
[| X |] -> [0.00000 |-0.30721 |-0.29540]
[| X | X] -> [0.00000 |-0.30721 |-0.62310]
[X | |] -> [-0.33088 |-0.01640 |-0.03155]
[X | | X] -> [-0.33088 |-0.01640 |-0.35925]
[X | X |] -> [-0.33088 |-0.32362 |-0.32694]
[X | X | X] -> [-0.33088 |-0.32362 |-0.65465]

9.4.3 Fast movements with and without having ExtCtrl enabled

@TODO: Rewrite
Describe experiment with fast motions

Hej

I did a small experiment to test and visualize the difference between the ExtCtrl-Mode switched on or off (-use_ilc "ext_ctrl" or not). Therefore I mounted a pen as tool and wrote a small RAPID program executing two very fast linear movements to draw a line on the table the robot is mounted on.

I executed the program in automatic mode, to get the fastest possible movement of the TCP to make the influence of the dynamic model visible.

Conclusion:
Robot in ExtCtrl-Mode:
The robot is not able to draw a straight line with full speed, due to the missing dynamic model, the inertias are not compensated. The deviation reached a maximum of 15mm on the 800mm long path. The line was shaped like an circular arc with the concave side facing towards the robot.

Robot in normal mode:
Even with full speed, the robot still draws a straight line.

In both cases, the start and the end point were reached exactly (by visual judgment). With this experiment, I didn't get any information about the accuracy of velocity and acceleration of the robot's movement. By visual judgment, the motion in normal mode (without -use_ilc "ext_ctrl") looked faster, but I am not really sure about that.

RAPID Code:

```
MODULE MainModule

 CONST jointtarget jtHome:=[[0,0,0,0,0,0],
[9E9,9E9,9E9,9E9,9E9,9E9]];

 CONST robtarget rtStartA :=[[-450, 500, 20],[0,0,-1,0],
[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
 CONST robtarget rtStart  :=[[-450, 500, 0],[0,0,-1,0],
[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
 CONST robtarget rtMiddl  :=[[ 350, 500, 0],[0,0,-1,0],
[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
 CONST robtarget rtEnd    :=[[ 350,-300, 0],[0,0,-1,0],
[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
 CONST robtarget rtEndA   :=[[ 350,-300, 20],[0,0,-1,0],
[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

 VAR num tpAnswTask := 0;

 PERS tooldata tdStift:=[TRUE,[[0,0,220],[1,0,0,0]],[1,
[0,0,100],[1,0,0,0],0,0,0]];

 PROC main()
       ! Turn off the configuration control
       ConfL \Off;

       TPWrite "Execution started";
       TPReadFK tpAnswTask, "Choose the task:", "Go Home", "",
"", "Long Line", "Break";
       TEST tpAnswTask
             CASE 1:     MoveAbsJ jtHome, v1000, fine, tool0; !
GoHome
             CASE 2:
             CASE 3:
             CASE 4: DrawLongLine;
             CASE 5:     break;
             DEFAULT: break;
       ENDTEST
 ENDPROC !! main

 PROC DrawLongLine()
       MoveJ rtStartA, v1000, z0, tdStift;
       MoveL rtStart, v1000, fine, tdStift;
       WaitTime 1;
       MoveL rtMiddl, vmax, fine, tdStift;
       MoveL rtEnd, vmax, fine, tdStift;
       WaitTime 1;
       MoveL rtEndA, v1000, fine, tdStift;
 ENDPROC

ENDMODULE
```