Kuka Matlab Connector Tutorial

Matthias Seehauser

07-06-2014

Version: 1.0

Contents

1 Introduction

Matthias Seehauser
Version 1.0

1.1 Version history

- V1.0: first documentation published (seehma)

1.2 Introduction

KMC was build to give the user an easy to use toolbox to control a kuka six axis degree of freedom robot out of Matlab® over a realtime network connection. This tutorial starts with the basic configuration steps and at the end the author will show some easy examples with the Matlab toolbox.

This Tutorial has the following steps:

- Configure the robot controller
- Configure the external PC-System
- Get all necessary Files
- Establish connection (step by step)
- Some Information about KMC

1.3 Version information

KMC was build and tested on a KRC2ed05 with a Kuka KR60HA robot. The RSI was available with version 2.3. Unfortunately the author was not able to test other versions of RSI. The chance that KMC works with other RSI versions on a KRC2 is high. There may be some bugs, but feel free to fix them.

For RSI 3 and further on newer KRC controllers some additions have to be made. The author was not able to test the whole toolbox, maybe someone is able to do this and can give some feedback.

2 Configure Robot

This tutorial describes how to create an ethernet connection from the robot controller to an external PC-System. The xml-packets are send out of VxWorks over the Kuka Router to the windows ethernet port on the kuka controller PC-system then to the ethernet port of the external PC-System to the KMC-object and last but not least to Matlab. In figure 1 the path of the xml-packets is shown.

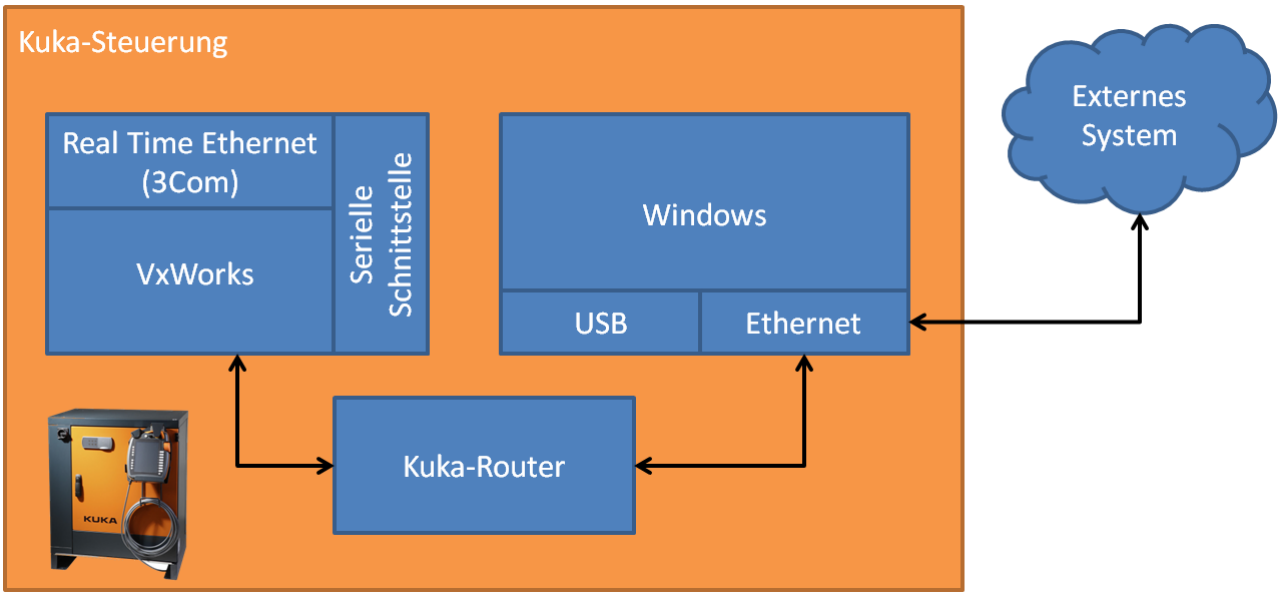

Figure 1: scheme of how to connect the external pc to the vxworks realtime system

This part of the tutorial includes the configuration of the kuka controller, kuka router programm and the usage of the right files for the rsi-realtime-communication.

2.1 Ethernet LAN-Adapter configuration

First we have to configure the Windows LAN-Adapter on the Kuka-controller. Therefore minimize the Kuka-HMI with clicking on the lower-left-Statusbar as shown in figure 2 and then click on the button minimize.

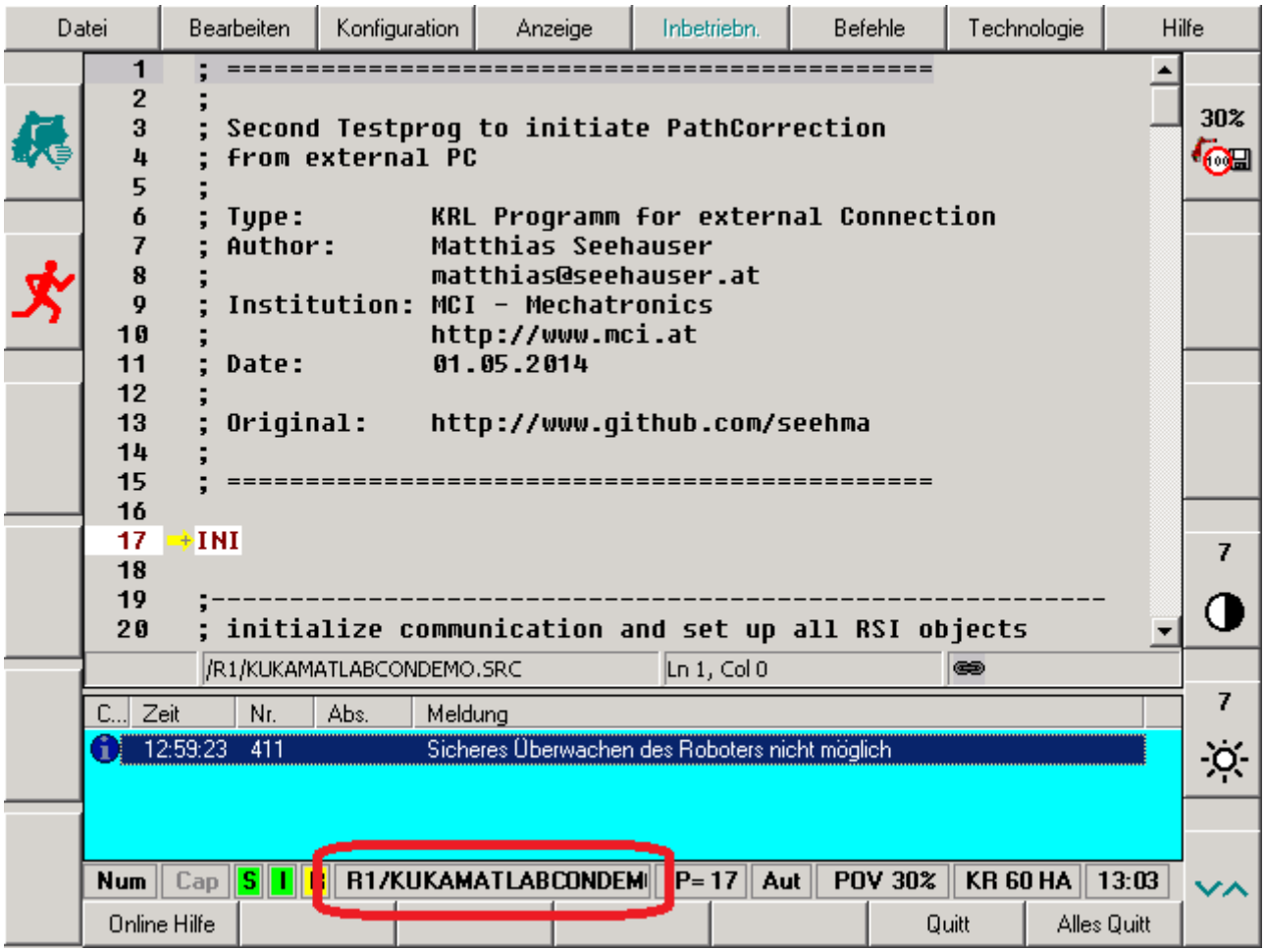

Figure 2: hmi statusbar to minimize the kuka hmi

Now click on the Windows Start Button and then open the control panel. Now do a right click on the symbol network connections and click on properties. Choose the right Windows LAN-Adapter and do a right click on it. Now click on Properties.

In this tutorial we configure the windows LAN-Adapter with the IP-Address 192.168.1.1 and the LAN-Adapter of the external PC-System with 192.168.1.11. In figure 3 one can see how the adapter should be configured.

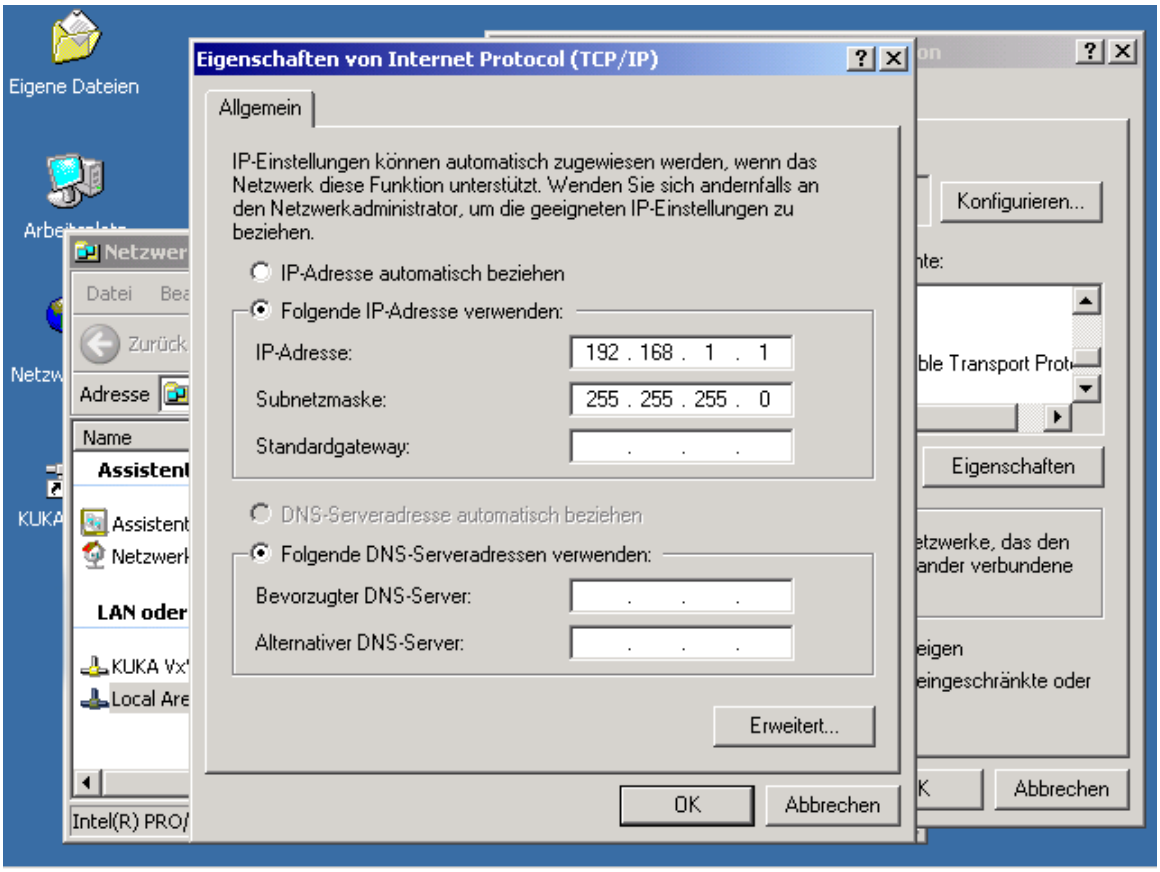

Figure 3: screenshot of the lan-adapter properties page

2.2 Connect KRC and PC

To establish a connection between those two systems one has to connect the KRC with a LAN switch and on the same switch the PC has to be connected. Or one can connect the two systems together over a cross over cable. Newer systems can do this cross over on the Ethernet port and one can connect the two systems with a normal Ethernet cable. In figure 4 the right ethernet port on the KRC is marked in red.

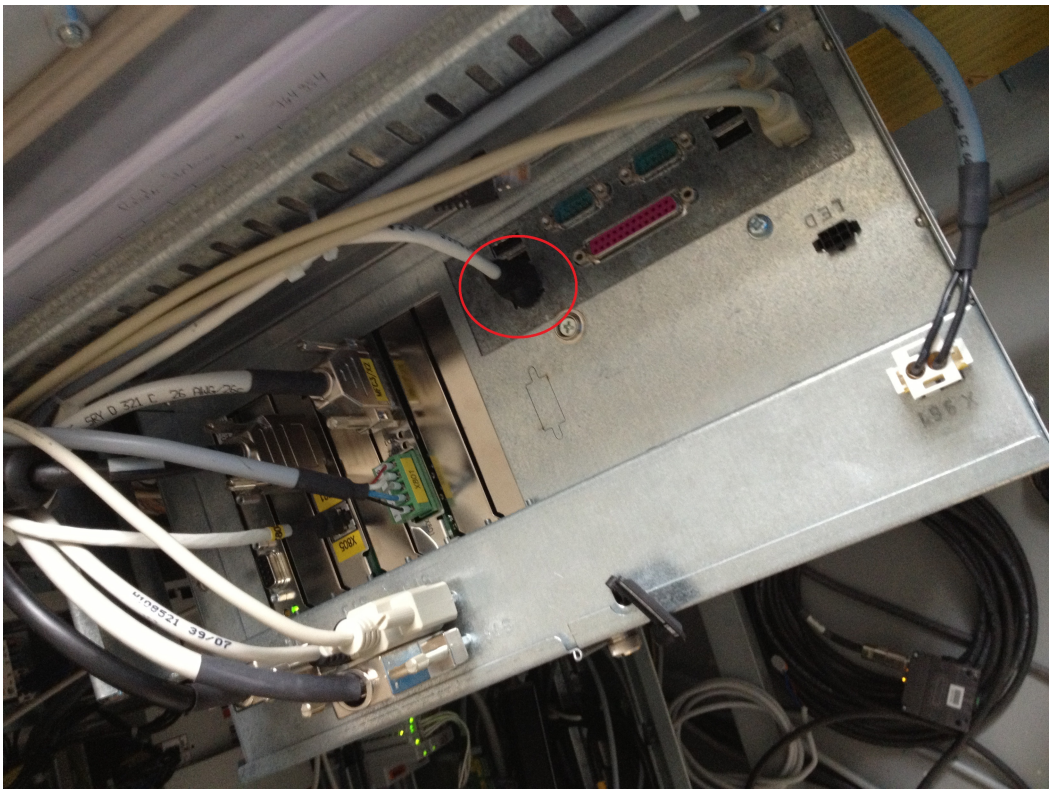

Figure 4: picture of the krc2ed05 windows lan-connector

2.3 Copy Files

The last step of preparing the robot controller is copying all necessary files to the controller. You need two types of files:

- Configfile for RSI Ethernet connection
- KRL-RSI-programm source and data file

The configuration-file is placed in `C:\KRC\ROBOTER\INIT` and is used to configure which data is send from the robot controller to the external pc-system and which is needed on the KRC as answer from the external pc-system. You also have to configure the target ip-address and port.

1. On the KMC-repository under `https://github.com/seehma/KMC/tree/master/filesOnController` one have to download the KukaRobotInfo_minẋml, KukaMatlabConnection_minṡrc and KukaMatlabConnection_mindat files.
2. Copy the KukaRobotInfo_minẋml file into the directory `C:\KRC\Roboter\INIT` (do it with a USB-Flash-Drive or burn those files on a CD-ROM)
3. Copy the KukaMatlabConnection_minṡrc and KukaMatlabConnection_mindat files into a subdirectory in `C:\KRC\Roboter\KRC\R1\Programm` (see figure 5) or somewhere else where you can start this programm

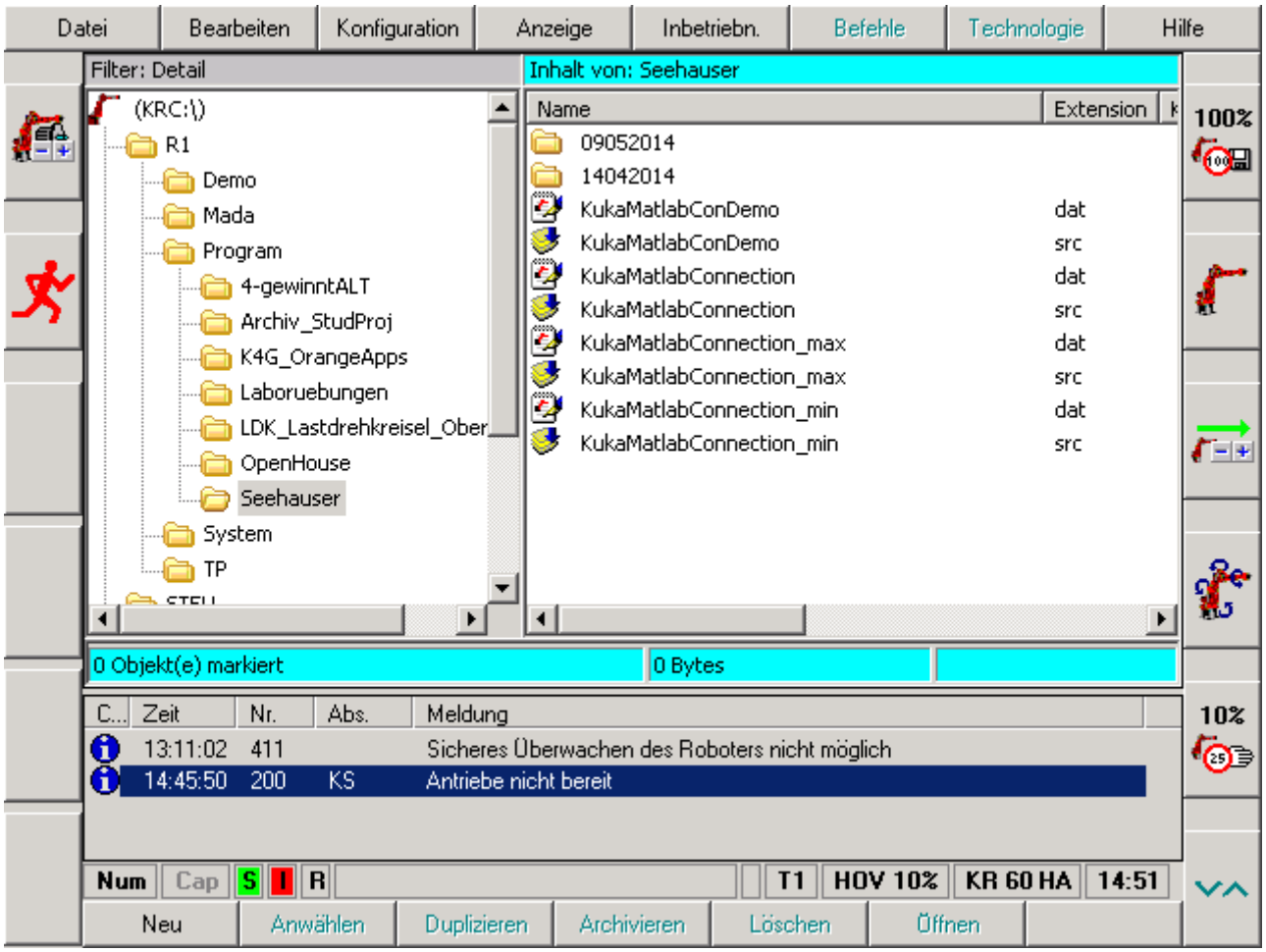

Figure 5: copy all necessary files to the KRC

There are several other examples in this workspace, but first try this simple one.

NOTICE: After copying the files you will have to restart the KRC, otherwise the files will not be shown in the menue of the HMI.

2.4 Configure Kuka-Router

To get the xml-files out of VxWorks over the Windows Lan-Adapter they have to be routed with a program from kuka. This can be installed from the RSI-Setup-Package on the CD-Rom or directly located on the kuka controller. For the installation Readme please look in the Kuka-manual.

Start the Kuka-Router program (minimize HMI as described in 2.1) and create a new route. Every created Route takes the packages from 192.0.1.2 and routes it to the given destination address and port and vice versa. In figure 6 the right settings are shown. To get this Route working the program has to be started. You can minimize it and then it will work in background (see at your task bar from windows on the lower right corner is the router icon).

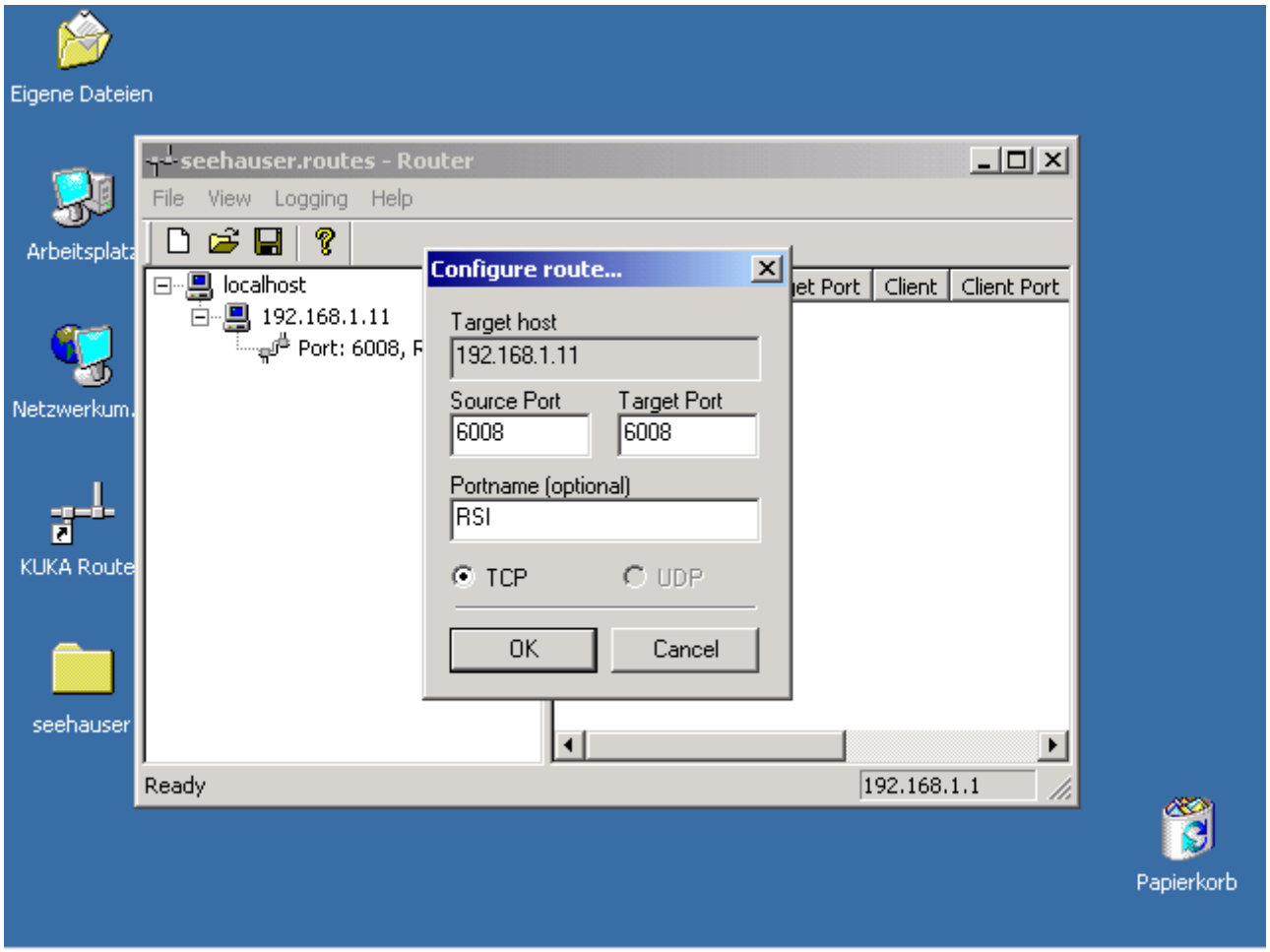

Figure 6: configure the kuka-router program on the KRC

NOTICE: After every restart of the KRC the Kuka-router program has to be started manually. Another option would be taking the program into the autorun folder of Windows, so it is started automatically.

3 Configure PC

To configure your PC the first is to configure the right IP-Address for your LAN-adapter. Go on your windows machine to Start - control panel - network and sharing center and then click on change adapter settings. There right click on the LAN-adapter which is connected to the KRC and then choose properties. Configure the IP address to 192.168.1.11 and the subnet mask to 255.255.255.0

3.1 Ping the KRC

To check if the connection to the KRC is up, just make an ICMP-Request. For that open in Windows a command window (or just press Ctrl+R and type cmd - Return) and then enter ping 192.168.1.1. The window should look like the one in figure 7.

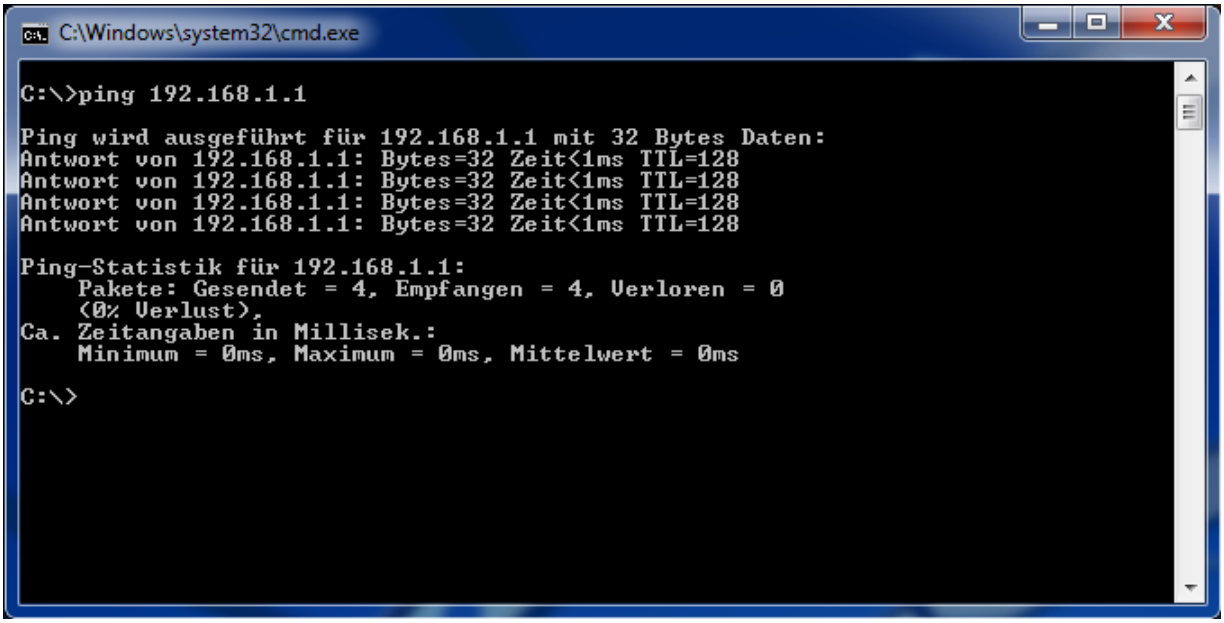

Figure 7: ping the KRC out of your Windows external PC-system

NOTICE: To have no extra load on the network connection to the KRC, it is recommended to switch off all network protocols on this LAN-adapter except of TCP/IP.

4 Prepare Files on PC

In this Section you will have to copy the necessary files of the toolbox in some specific directories.

4.1 Matlab Files

In the KMC-Repository under `https://github.com/seehma/KMCMatlab` you will be able to download all necessary files for Matlab. This includes also some testing examples to calculate a sinoid profile and to make some tests with a six axis degree of freedom force torque sensor.

Copy those files into a workspace which you can specify in Matlab for example `C:\Temp\KMCMatlab`. Now Start Matlab and change the working directory into that place by clicking on the choosin button as shown in figure 8

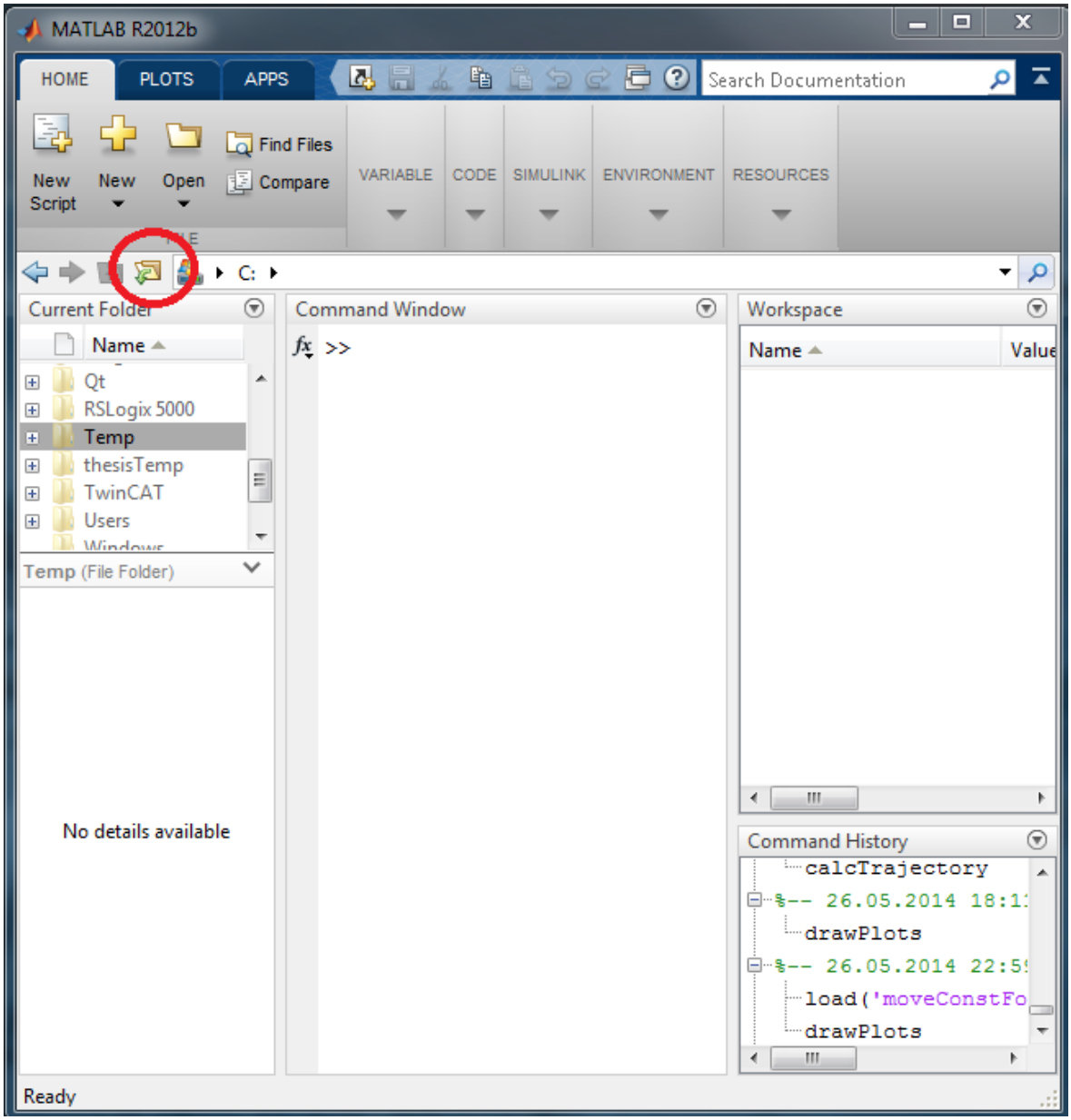

Figure 8: Start Matlab and choose the right working directory

Now select the directory like in figure 9 and you should now see all files in Matlab.

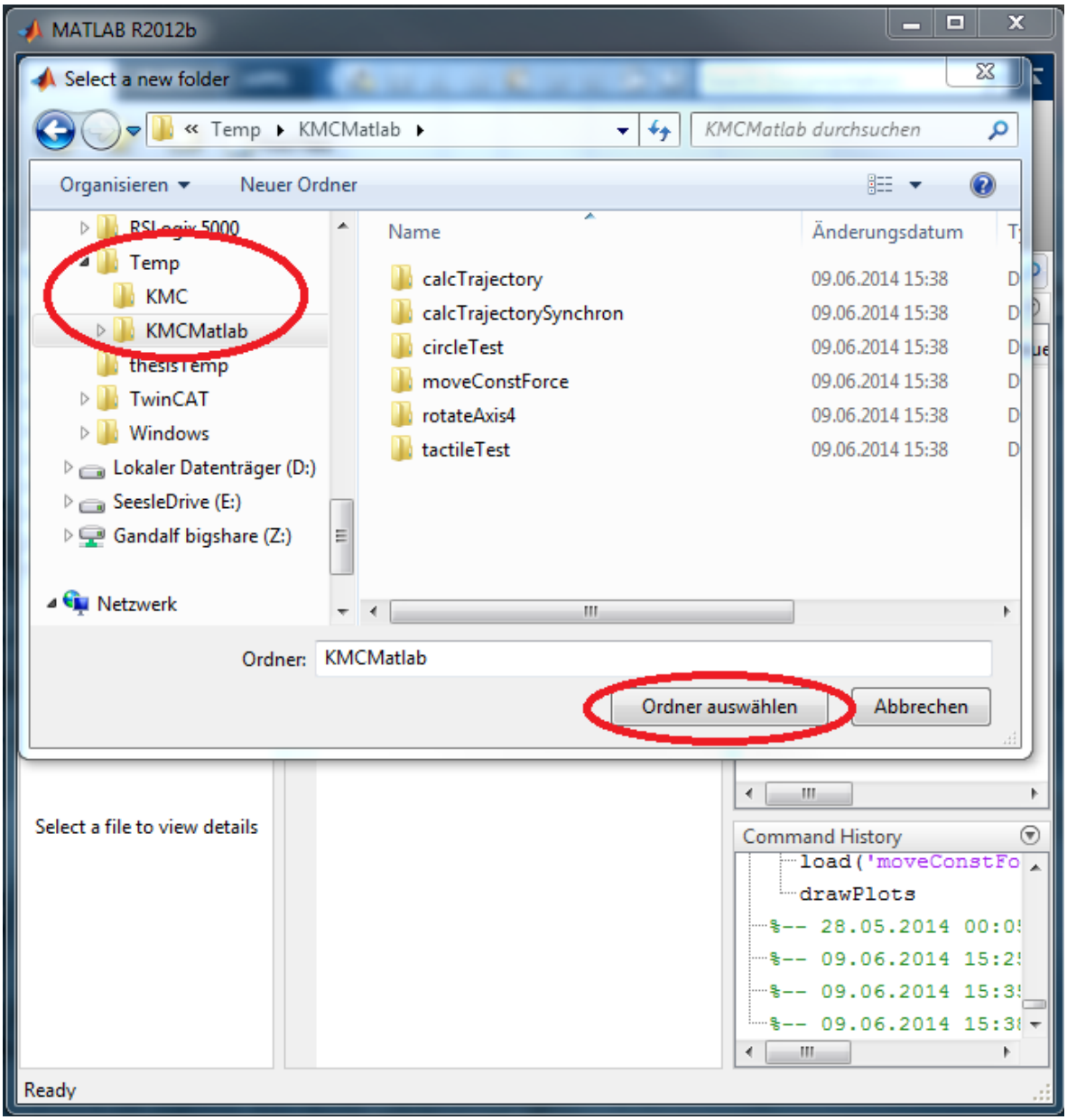

Figure 9: Choose the working directory

Matlab has to send every 12 Milliseconds one package to the kuka controller. As described above we have chosen the minimized example to get the connection to work on every controller (also without Force Torque sensor) so one has to choose the right command-xml-file for Matlab. This file is opened when the connection starts and modified everytime when the user changes the correction for the robot.

Go to the KMC repository under `https://github.com/seehma/KMC/tree/master/filesOnKMC` and download the file commanddoc_min.xml. Copy the file into the Matlab workspace where all other files are located.

As standard the Matlab KMC object opens the command file named commanddoc.xml. So rename the current commanddoc.xml file to commanddoc_old.xml and rename the commanddoc_min.xml file to commanddoc.xml.

NOTICE: Every configuration on the Kuka controller has its own command file in Matlab. Different configurations won't work because the Kuka controller does not get the attributes needed.

4.2 KMC-Object Dll's

To get the connection from the robot controller to Matlab done, Matlab needs the KMC as .NET assembly. One can download the KukaMatlabConnector.dll files from `https://github.com/seehma/KMC/releases/tag/0.1.0`. One will also need the TextLogger dll Files from `https://github.com/seehma/TextLogger/releases/tag/1.0.0`. Put these files together in the directory `C:\Temp\KMC` as shown in figure 10.

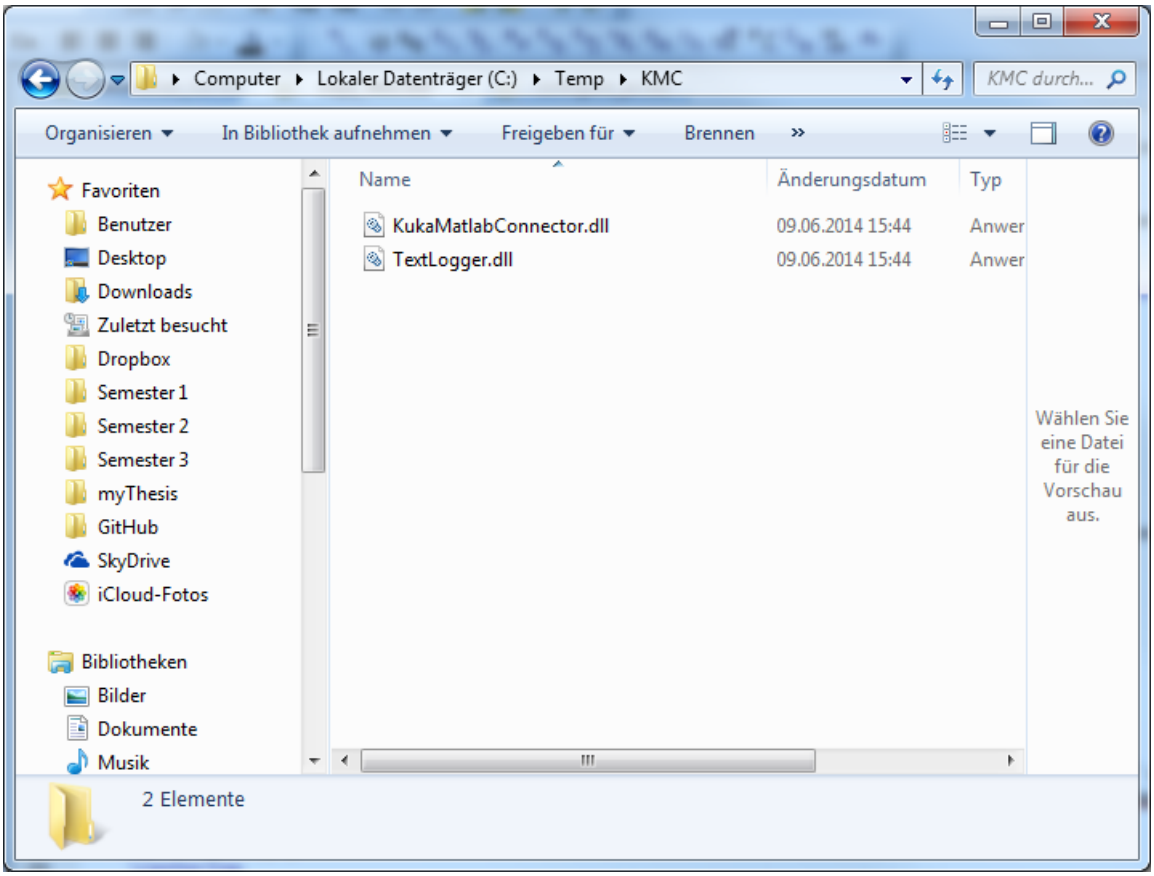

Figure 10: Select directory and confirm

5 Establish Connection

To establish the connection there are three main steps:

- Start toolbox in Matlab and listen on the right LAN-Adapter
- Open program on KRC-HMI
- Go on KRC to ST_SKIPSENS command

First you have to start the Matlab toolbox. Go into the desired Workspace and open a new M-File. Type the following commands into the M-File and Run it to start the connection:

```
1 disp('starting wrapper...');
2 t=robotConnector('192.168.1.11',6008,0.012,'C:\KukaMatlabConnector.dll');
3 disp('starting connection to robot...');
4 t.connect();
5 disp('starting GUI...');
6 t.initGUI();
```

Listing 1: Aufbau der Verbindung zum Roboter mit Matlab

On some windows systems a windows comes up to get the users confirmation that the Toolbox wants to communicate through the firewall, just allow this connection to go on. Also the Matlab Toolbox GUI is starting and you can see in the lower right status field that the status should be on "'listening"'.

Now go to the KRC and bring the controller into automatic mode. NOTICE: To get the connection to work, the controller has to be error free so the breaking test has to be done and the safety controller has to be OK. To compare the status of the KRC to a valid one see figure 11.

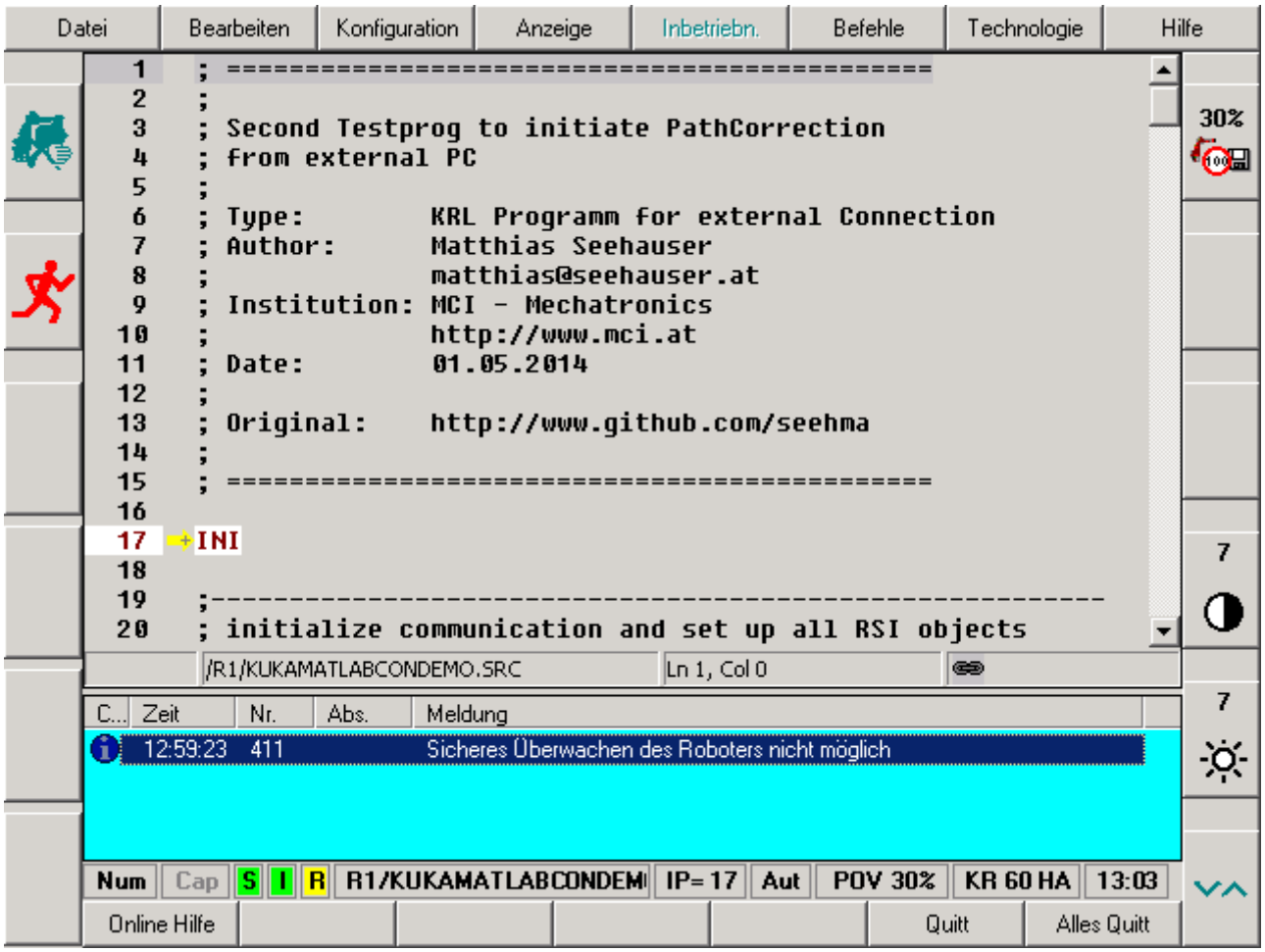

Figure 11: valid KRC status to get connection to work

As last step go on the KRC to the command ST_SKIPSENS() (figure 12).

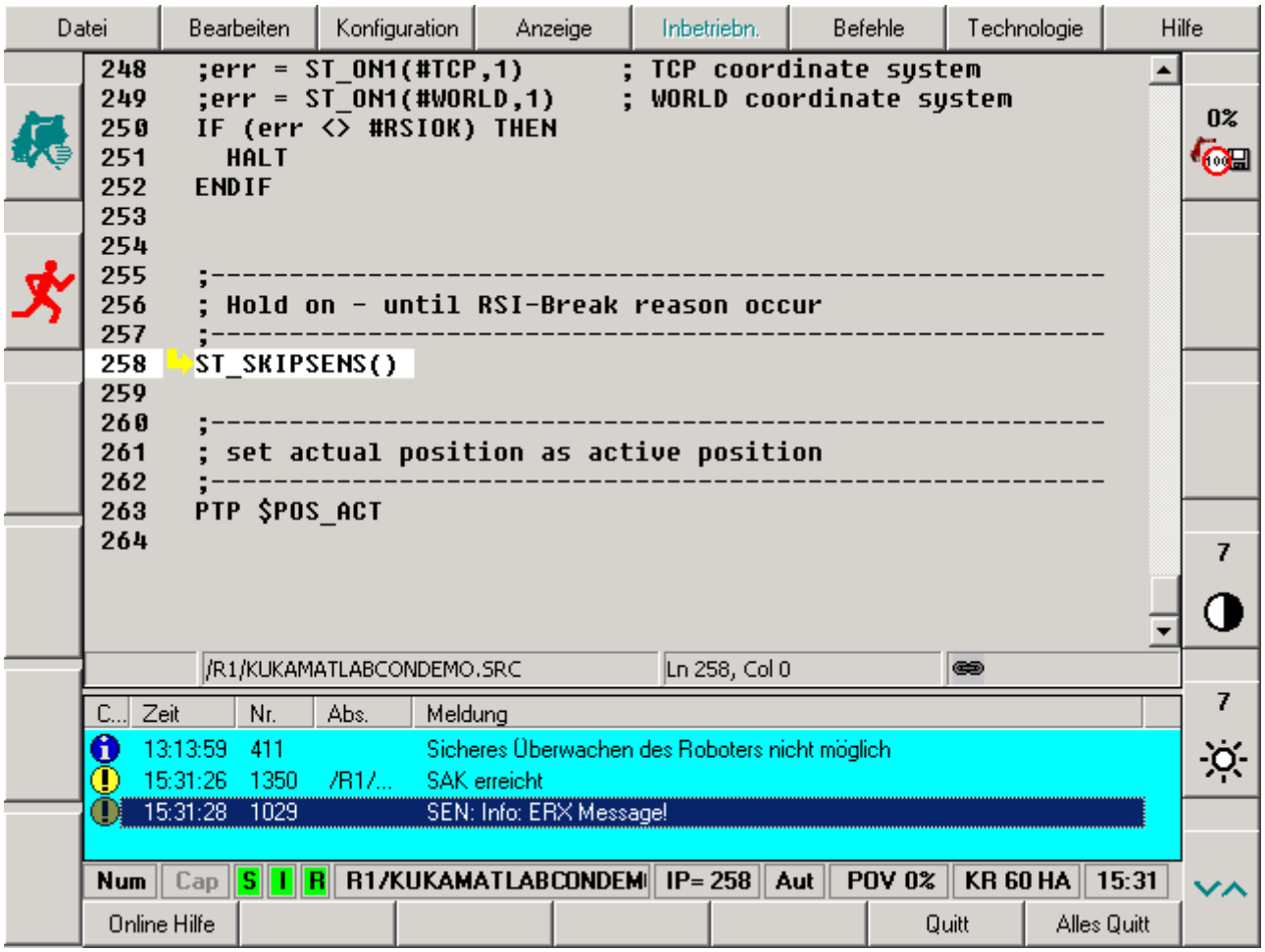

Figure 12: Status of the KRC when connected to the external system

Now the connection should work and in the Matlab GUI the xml-packets are shown as you can see in figure 13.

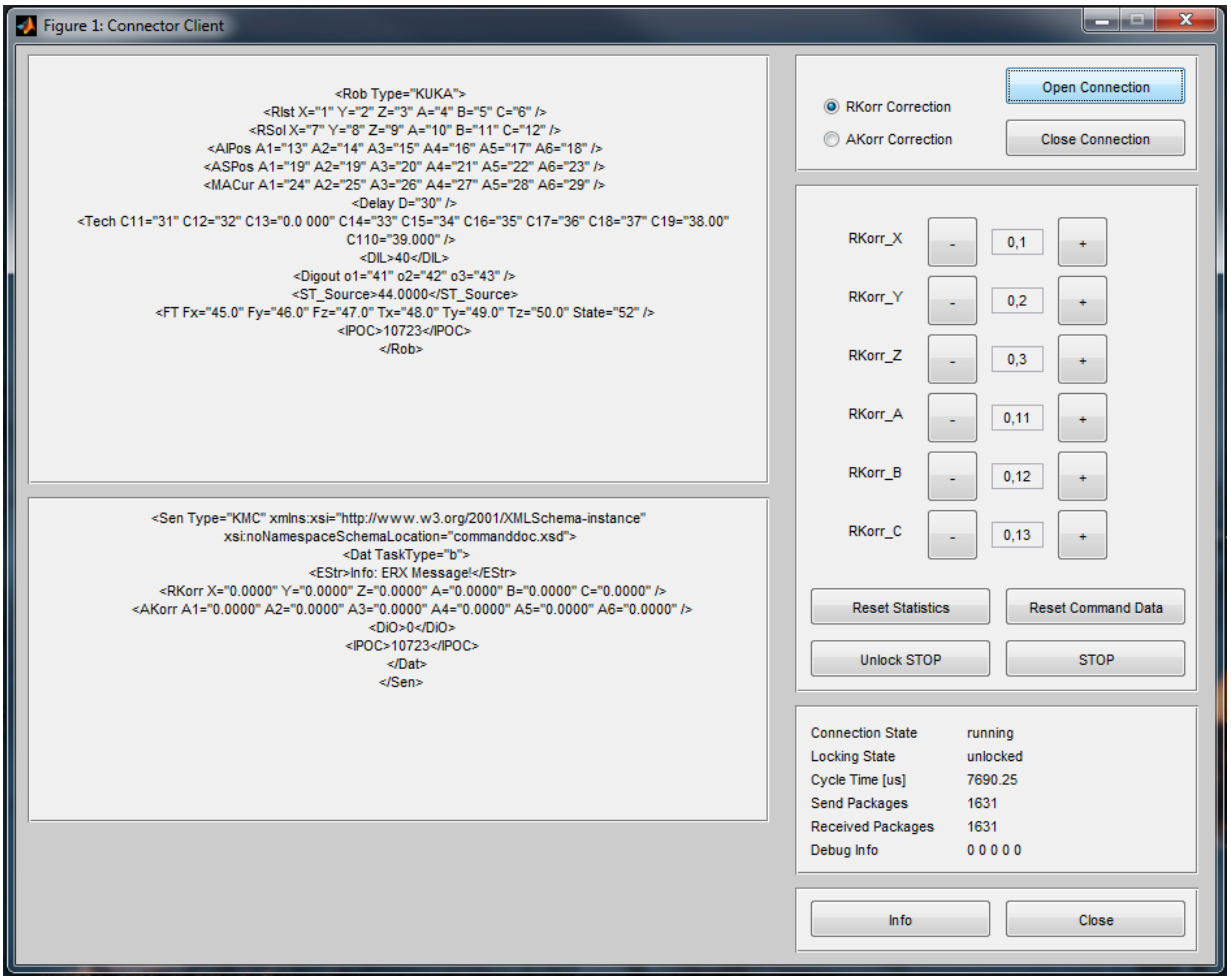

Figure 13: KMC connected (note: this was an example with a force torque sensor)

6 Optional Configurations

In this chapter some optional configurations are described. If you have some ideas of more examples please write an E-Mail to matthias@seehauer.at and i will add them to this documenation.

6.1 Maximum Allow Path Correction

In the described example the maximum allowed correction is in axis Angles set to

- Axis1: -10 ... 10 degrees
- Axis2: -10 ... 10 degrees
- Axis3: -10 ... 10 degrees
- Axis4: -10 ... 10 degrees
- Axis5: -10 ... 10 degrees
- Axis6: -10 ... 10 degrees

and in cartesian coordinates

- X: -5 ... 380 mm
- Y: -540 ... 540 mm
- Z: -100 ... 230 mm
- A: -90 ... 90 mm
- B: -45 ... 45 mm
- C: -90 ... 90 mm

To get a custom maximum range just edit them in the file KukaMatlabConnection_min.src in the maximum path and maximum axis correction section.

7 Info

List of Figures