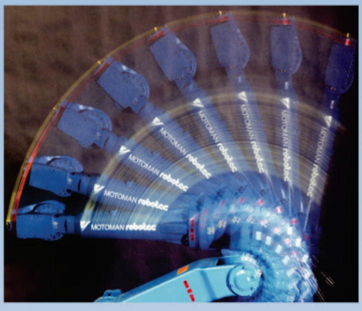
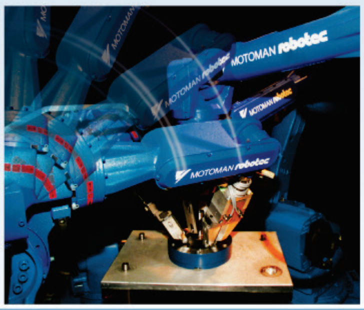




Motocom32

Users Manual

V1.10

MOTOMAN®

Table of contents

FIGURES LIST

TABLES LIST

History

Date	Version	User	Modification
15.4.2011	1.0	Waibel	Initial version, merge from Quickstart guide
18.9.2012	1.1	Waibel	Added FS100, added HSEServer

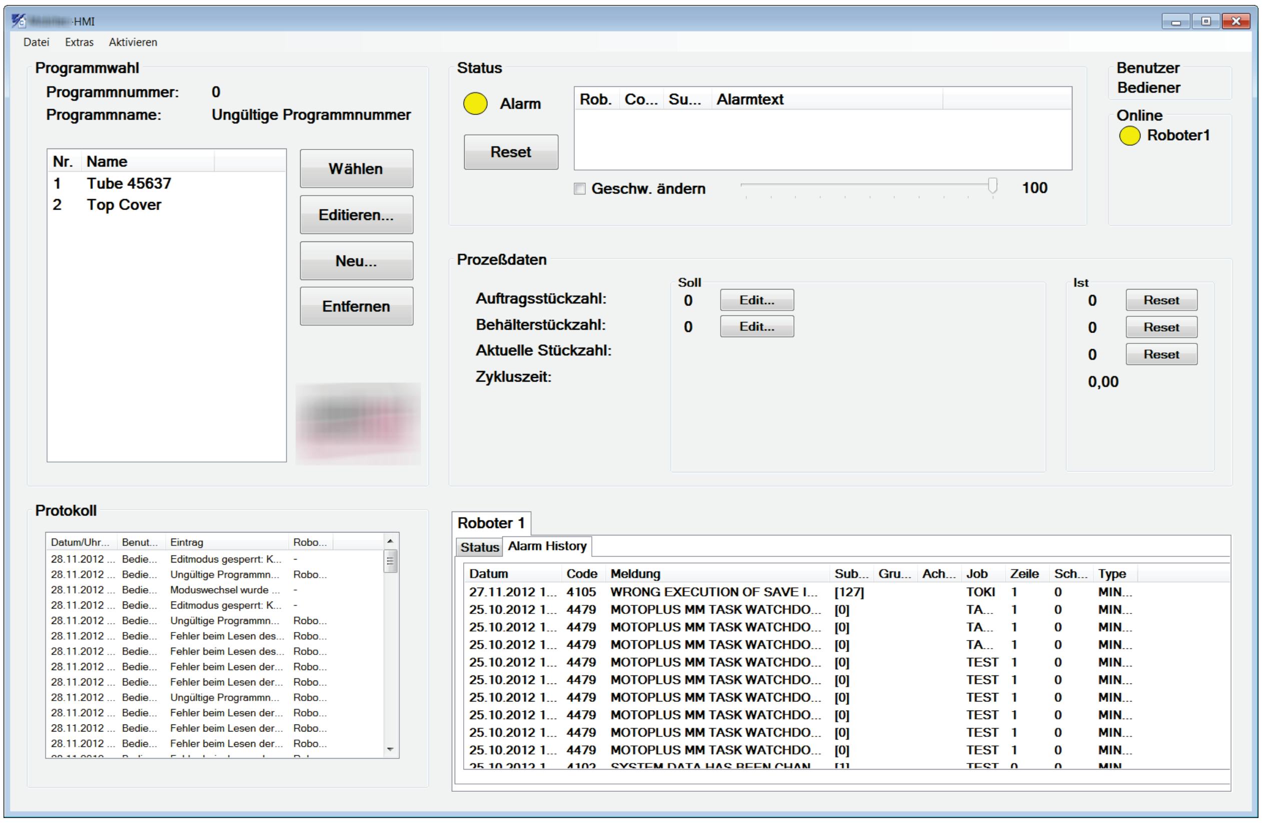
1 Introduction

Motocom32 provides the functionality for accessing Motoman Yasnac robot controllers by a Windows-PC.
Therefore this manual is basically valid for all software, which uses Motocom32 DLL. Motocom32 DLL supports three different modes. Two of them are shown in the following table. Also the software packages using these modes are listed.

Mode	**Host**			**DCI**
Protocol	**BSC**	**EServer**	**HSEServer**	**BSC**
JobExchanger32	X			
MotoAdmin	X	X		
MotoOPCServer		X		
MotoDCI/MotoJob32				X
Proface HMI			X	

Table 1 Motoman software and corresponding communication modes.

Motocom32 makes it easier to access data on Yasnac controllers by Windows PC. It is used to include data access abilities to customer built applications.



Programmwahl

Programmnummer: 0
Programmname: Ungültige Programmnummer

Nr.	Name
1	Tube 45637
2	Top Cover

Status

Alarm

Rob.	Co...	Su...	Alarmtext

Geschw. ändern 100

Benutzer
Bediener

Online
Roboter1

Prozeßdaten

	Soll	Ist
Auftragsstückzahl:	0	0
Behälterstückzahl:	0	0
Aktuelle Stückzahl:		0
Zykluszeit:		0,00

Protokoll

Datum/Uhr...	Benut...	Eintrag	Robo...
28.11.2012 ...	Bedie...	Editmodus gesperrt: K...	-
28.11.2012 ...	Bedie...	Ungültige Programmn...	Robo...
28.11.2012 ...	Bedie...	Moduswechsel wurde ...	-
28.11.2012 ...	Bedie...	Editmodus gesperrt: K...	-
28.11.2012 ...	Bedie...	Ungültige Programmn...	Robo...
28.11.2012 ...	Bedie...	Fehler beim Lesen des...	Robo...
28.11.2012 ...	Bedie...	Fehler beim Lesen des...	Robo...
28.11.2012 ...	Bedie...	Fehler beim Lesen der...	Robo...
28.11.2012 ...	Bedie...	Fehler beim Lesen der...	Robo...
28.11.2012 ...	Bedie...	Ungültige Programmn...	Robo...
28.11.2012 ...	Bedie...	Fehler beim Lesen der...	Robo...
28.11.2012 ...	Bedie...	Fehler beim Lesen der...	Robo...
28.11.2012 ...	Bedie...	Fehler beim Lesen der...	Robo...

Roboter 1

Status Alarm History

Datum	Code	Meldung	Sub...	Gru...	Ach...	Job	Zeile	Sch...	Type
27.11.2012 1...	4105	WRONG EXECUTION OF SAVE I...	[127]			TOKI	1	0	MIN...
25.10.2012 1...	4479	MOTOPLUS MM TASK WATCHDO...	[0]			TA...	1	0	MIN...
25.10.2012 1...	4479	MOTOPLUS MM TASK WATCHDO...	[0]			TA...	1	0	MIN...
25.10.2012 1...	4479	MOTOPLUS MM TASK WATCHDO...	[0]			TA...	1	0	MIN...
25.10.2012 1...	4479	MOTOPLUS MM TASK WATCHDO...	[0]			TEST	1	0	MIN...
25.10.2012 1...	4479	MOTOPLUS MM TASK WATCHDO...	[0]			TEST	1	0	MIN...
25.10.2012 1...	4479	MOTOPLUS MM TASK WATCHDO...	[0]			TEST	1	0	MIN...
25.10.2012 1...	4479	MOTOPLUS MM TASK WATCHDO...	[0]			TEST	1	0	MIN...
25.10.2012 1...	4479	MOTOPLUS MM TASK WATCHDO...	[0]			TEST	1	0	MIN...
25.10.2012 1...	4479	MOTOPLUS MM TASK WATCHDO...	[0]			TEST	1	0	MIN...
25.10.2012 1...	4102	SYSTEM DATA HAS BEEN CHAN...	[1]			TEST	0	0	MIN...

Figure 1 Example HMI created based on Motocom

Motocom32 API is provided as a standard 32bit Windows DLL, so it can be used with the common development environments like Microsoft Visual C++, Visual Basic etc.
Die DLL functions can also be called from Microsoft Visual Studio.NET development tools like VB.NET or VC# (see examples). But you have to take into account that the Motocom32-DLL itself is provided as 32Bit (x86) „unmanaged" code.

2 System Requirements

OS	Microsoft 2000 / XP(32-bit) / 7(32-bit)
Memory	128 Mbyte or more
Hardware Disk Capacity for Installation	50 Mbyte or more
Display	Supported by MS-Windows
Robot Controller	FS100, DX100, NX100, YASNAC XRC, MRC, MRC2, ERC, and ERC2
Transmission Cable	Ethernet cable or RS232C cable

3 Installation

- Insert CDROM in drive, CD starter application should start automatically.
- If not locate CDROM drive and run *cdstarter.exe.*
- Make sure that you have administrative rights

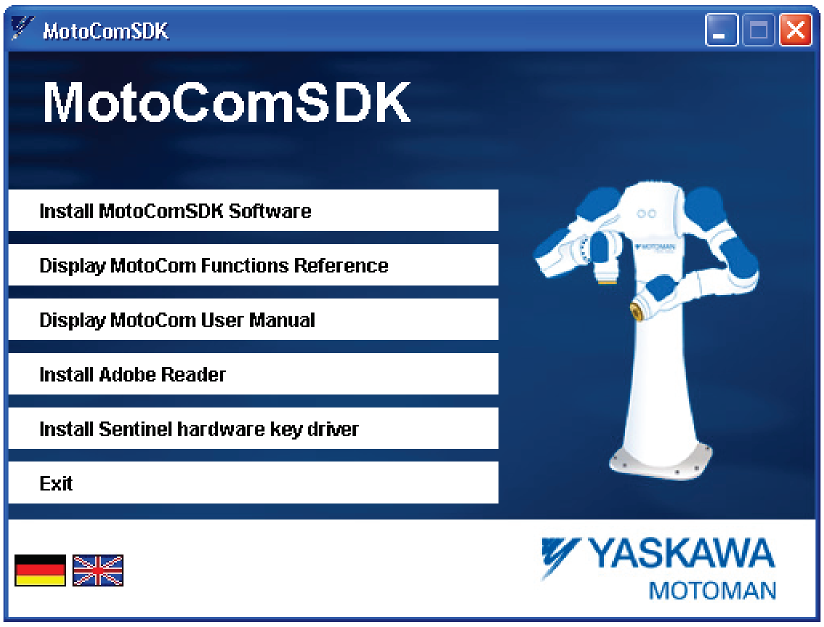
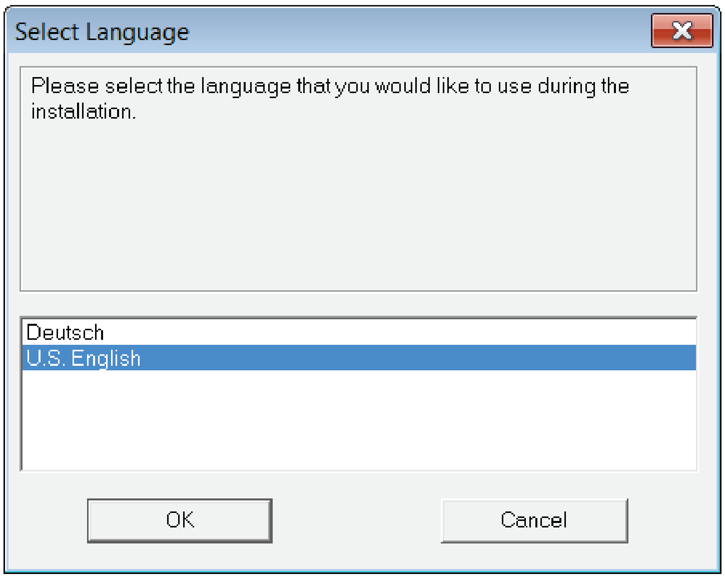

MotoComSDK
MotoComSDK
Install MotoComSDK Software
Display MotoCom Functions Reference
Display MotoCom User Manual
Install Adobe Reader
Install Sentinel hardware key driver
Exit
YASKAWA
MOTOMAN

Figure 2 CD starter menu

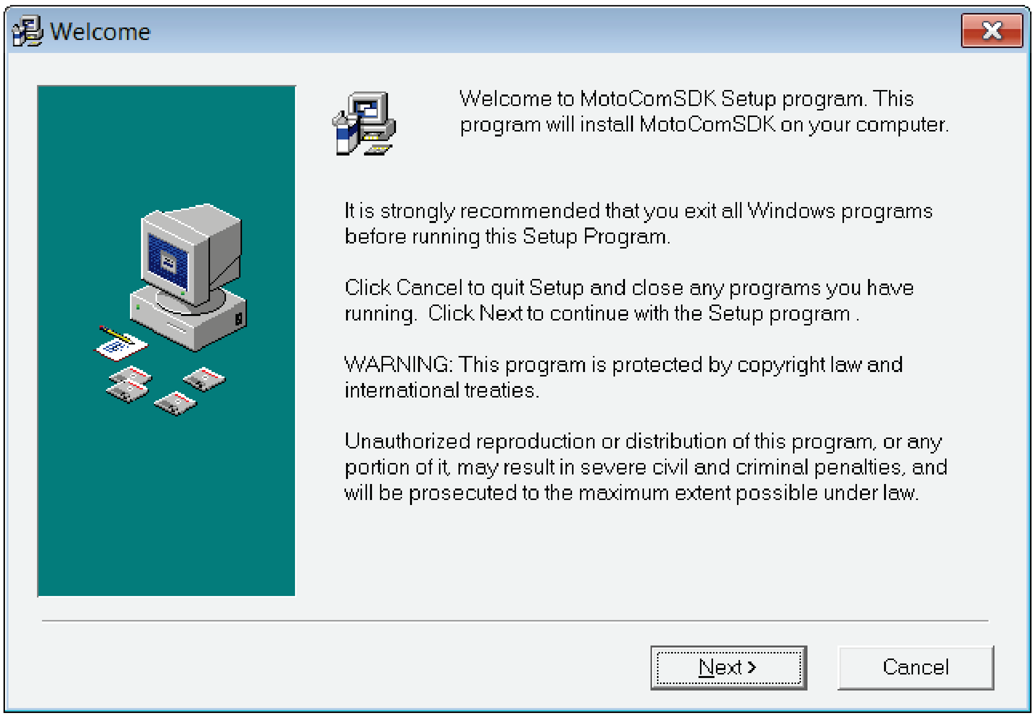
- Select Installation.


Select Language

Please select the language that you would like to use during the installation.

Deutsch
U.S. English

- Select language for installation.



Welcome to MotoComSDK Setup program. This program will install MotoComSDK on your computer.

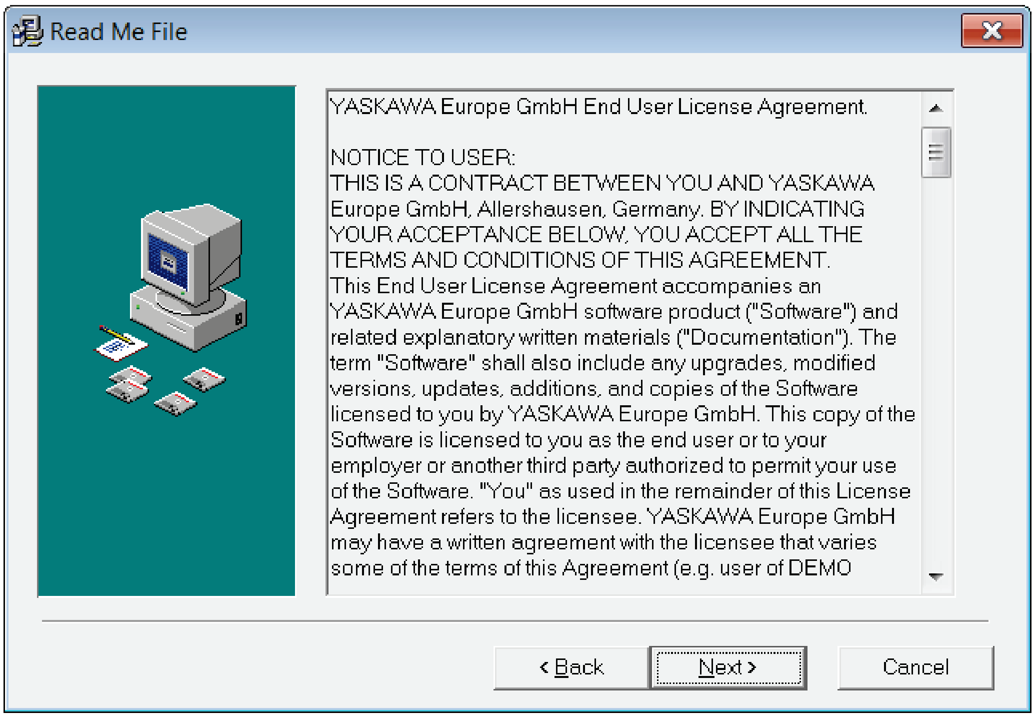
It is strongly recommended that you exit all Windows programs before running this Setup Program.

Click Cancel to quit Setup and close any programs you have running. Click Next to continue with the Setup program .

WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

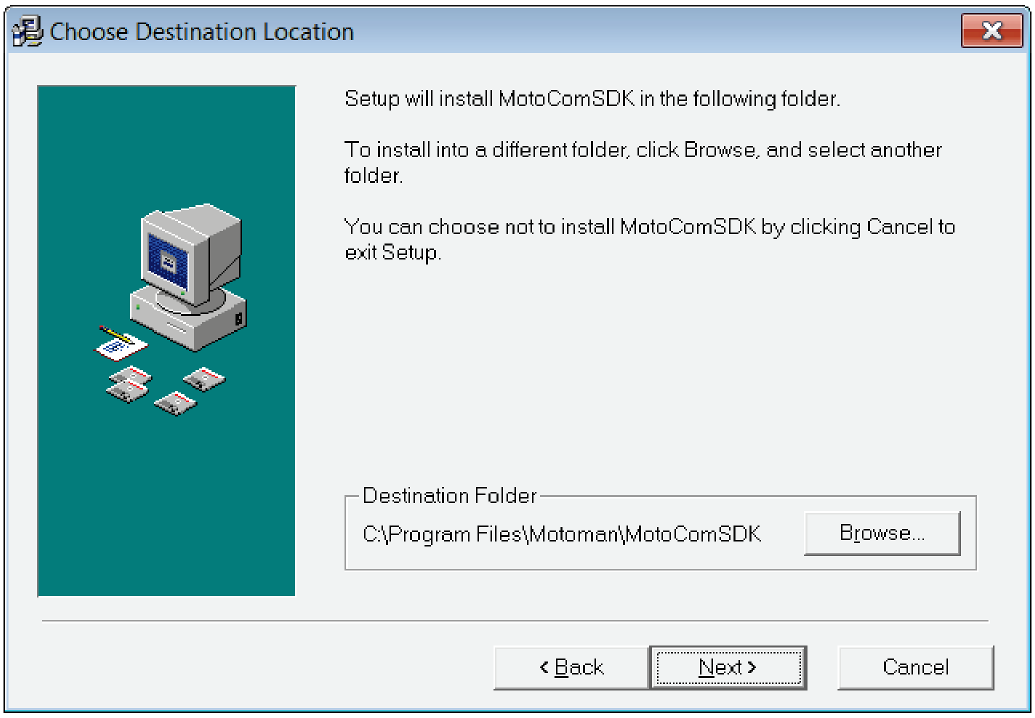
► Select Next.



YASKAWA Europe GmbH End User License Agreement.

NOTICE TO USER:
THIS IS A CONTRACT BETWEEN YOU AND YASKAWA Europe GmbH, Allershausen, Germany. BY INDICATING YOUR ACCEPTANCE BELOW, YOU ACCEPT ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT.
This End User License Agreement accompanies an YASKAWA Europe GmbH software product ("Software") and related explanatory written materials ("Documentation"). The term "Software" shall also include any upgrades, modified versions, updates, additions, and copies of the Software licensed to you by YASKAWA Europe GmbH. This copy of the Software is licensed to you as the end user or to your employer or another third party authorized to permit your use of the Software. "You" as used in the remainder of this License Agreement refers to the licensee. YASKAWA Europe GmbH may have a written agreement with the licensee that varies some of the terms of this Agreement (e.g. user of DEMO

► Confirm license agreement.


Choose Destination Location

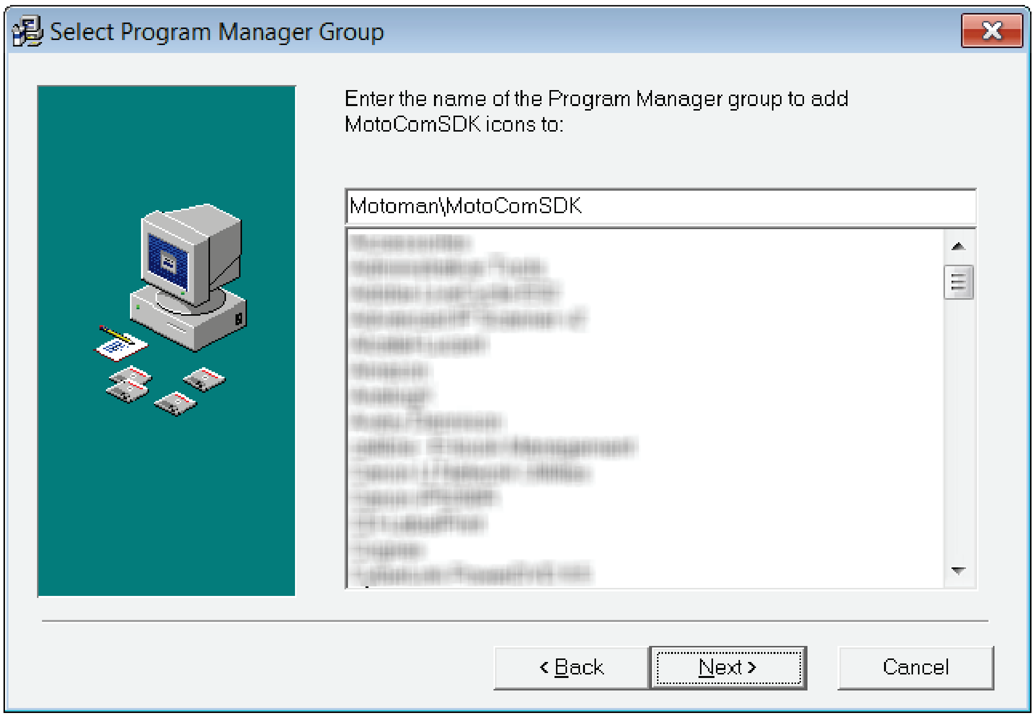
Setup will install MotoComSDK in the following folder.

To install into a different folder, click Browse, and select another folder.

You can choose not to install MotoComSDK by clicking Cancel to exit Setup.

Destination Folder
C:\Program Files\Motoman\MotoComSDK

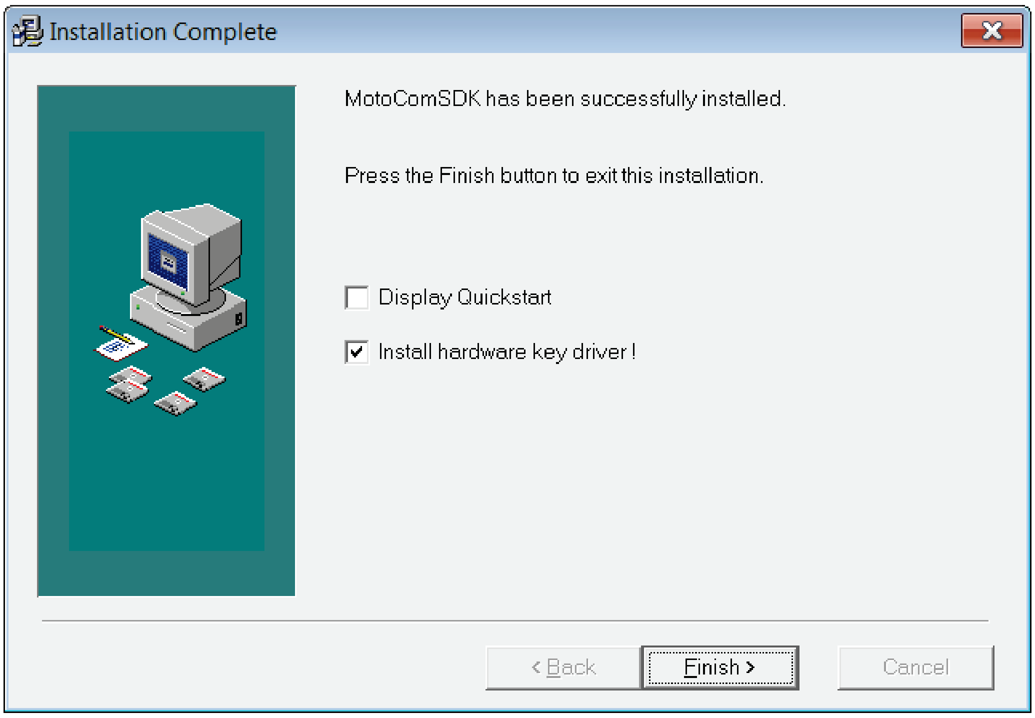
▶ Select application folder.


Select Program Manager Group

Enter the name of the Program Manager group to add MotoComSDK icons to:

Motoman\MotoComSDK

▶ Select Program Manager Group.


Installation Complete

MotoComSDK has been successfully installed.

Press the Finish button to exit this installation.

☐ Display Quickstart

☑ Install hardware key driver !

▶ Finish Installation process:

- To Install hardware key driver select the corresponding option.
- For displaying manual select “Display Quickstart”.

4 Quickstart

4.1 Step 1: How to connect to the robot controller ?

First hardware interface has to be selected:

► TCP/IP-Ethernet
By default Ethernet interface is only available at NX100, DX100 and FS100 controllers. ERC does not support Ethernet but for MRC and XRC controllers Ethernet interface board is optionally available. That means it might be necessary to purchase an additional board before using Ethernet functionality. The Ethernet speed is specified with 10Mbit (MRC, XRC) respectively 100 Mbit in case of NX100, DX100 and FS100. The maximum distance between communication hosts is unlimited if additional Ethernet hardware is used. One PC can be connected to an unlimited number of robot controllers.

► RS232C serial interface.
By default every Yasnac robot controller (ERC, MRC, XRC, NX100, DX100 and FS100) is equipped with a RS232C interface. The maximum data rate is 19200 baud in case of FS100, DX100, NX100 and XRC (9600 recommended), 9600 Baud in case of MRC and 4800 Baud in case of ERC. The maximum cable length is limited to short distances because of the noise sensitivity of RS232C standard. PC's normally provide a maximum of 4 serial interface (standard is 1 or 2). That means a maximum of 4 robots can be connected to one PC simultaneously. Special interface boards are available to increase the number of Com ports.

► RS232C serial with Com server device
Instead of using pure serial connection a Com server may be an interesting alternative. Robot controller is accessed by serial interface. Therefore the speed is same as with pure serial connection. Com server device is connected to PC by Ethernet hardware, which leads into much higher distances between PC and controller. Com server is usually mounted inside controller cabinet. Accessing the controller is done by virtual com ports, which are provided by special drivers delivered with Com server device. The maximum number of (virtual) Com ports is limited to 255.

4.2 Step 2: What kind of tasks should be handled ?

As already mentioned there are two different modes which provide different features:

► Host mode
- Robot controller can be controlled from remote pc side.
- Communication is executed asynchronously.
- Examples for available functions are:

  - Getting job list.
  - Running jobs.
  - Moving manipulator.
  - Reading variables, I/O’s.
  - …
- Writing abilities may be limited depending on controller type, operation mode (Teach, Play and Operating) and robot firmware.
- FS100, DX100 controllers and NX100 controllers with newer firmware versions installed support EServer protocol. These protocol is interesting because of the following features:
  - Faster,
  - more stable concerning remote switching,
  - Multi-client ability.
  - Source code compatible to BSC protocol. Only functions call for opening the communication must be adapted.
  - **The EServer function must be activated by special manufacturer parameters.**

  EServer protocol is the recommended protocol if Host mode is used.
- FS100 controllers and DX100 controllers with newer firmware versions installed support High Speed Ethernet Server protocol. These protocols offer higher performance in comparison to previous BSC and EServer protocols.

► DCI mode
- DCI stands for Data Communication By Instruction. The communication is initiated by robot controller by executing special instructions in a running job.
- Communication is synchronous, Yasnac controller is client, PC has to provide server functionality.
- Mode allows transmission of job and variables only.

4.3 Step 3: Configuration of the communication

The configuration depends on the selected hardware connection.

4.3.1 Hardware independent settings

	ERC		**PC side**
Firmware	->4.00	4.20->	
Protokoll (internal inter-face/Ethernet)	-	-	-
Protocol (external interface)	-	-	-
Writing (Jobs/Variables) dur-ing Playback Opera-tion	-	-	-

	MRC	**XRC**	**NX100, DX100**	**FS100**	**PC side**
Firmware	all	all	all	all	
Protokoll (internal inter-face/Ethernet)	RS001=2* (not MRCII)	RS000=2	RS000=2	RS000=2	BSC Protocol
Protocol (external interface	RS000=2*	only internal !	-	-	BSC Protocol
Writing (Jobs/Variables) during Playback Operation	-	-	RS029=1	RS029=1	-

Table 2 Hardware independent settings

(*) Attention: In case of MRC RS000 and RS001 may not have the same value. So for example set parameter of selected port to 2 and the remaining parameter to 3.
Example for using external 25pin interface MRC
RS000=2
RS001=3
In case of XRC, only internal interface can be used with Motocom.

4.3.2 Serial communication

- Cable connection.
  Wiring of cable to be used must be according to a standard null modem cable. Such a cable is included in Motocom32 software package.
- To configure the serial interface of the robot controller the following RS parameters must be adjusted.

	ERC		**PC side**
Firmware	->4.00	4.20->	
Data bits	RS00=8	RS020=8	8 Data bits
Stop bits	RS01=0	RS021=0	1 Stop bit
Parity	RS02=2	RS022=2	even parity
Baud rate	RS03=32	RS023=32	4800 Baud
Timer A	-	-	-

Timer B	-	-	-
ENQ Retry	-	-	-
Data Retry	-	-	-
Block Check	-	-	-

	MRC	XRC	NX100, DX100	FS100	PC side
Firmware	all	all	all	all	
Data bits	RS030=8	RS030=8	RS030=8	RS030=8	8 Data bits
Stop bits	RS031=0	RS031=0	RS031=0	RS031=0	1 Stop bit
Parity	RS032=2	RS032=2	RS032=2	RS032=2	even Parity
Baud rate	RS033=7	RS033=7 (8)	RS033=7 (8)	RS033=7 (8)	9600 (19200) Baud
Timer A	RS034=30	RS034=30	RS034=30	RS034=30	-
Timer B	RS035=200	RS035=200	RS035=200	RS035=200	-
ENQ Retry	RS036=10	RS036=10	RS036=10	RS036=10	-
Data Retry	RS037=3	RS037=3	RS037=3	RS037=3	-
Block Check	RS038=0	RS038=0	RS038=0	RS038=0	-

Table 3 Serial interface parameter configuration.

- Remote mode settings
  - ⇒ Refer to Table 4.

4.3.3 Ethernet communication

Before successfully establishing an Ethernet connection some arrangements must be done:

- Installing Ethernet interface card.
  This is not necessary in case of NX100, DX100 or FS100, because these controllers are already delivered with inbuilt network interface.
- Connecting network port to the local network.
  Standard Ethernet hardware like switches, routers etc. can be used to attach controller to the company network:
  - Network speed
    Yasnac Ethernet interfaces currently support 10 MB/s (MRC, XRC) or 100 MB/s (NX100, DX100, FS100). If the controller should be connected to a network with higher speed, partner port must also be configured to use this speed. High-quality switches and pc network cards normally are configured to use “auto-negotiation” which means they adjust their own speed to the speed of the partner port.
  - AUI interface
    MRC-/XRC- network boards are equipped with an AUI interface. Modern networks are based on different network schemes. To convert the signals from AUI to today’s Fast Ethernet or Gigabit networks based on RJ45 connectors a transceiver is used. This transceiver is bundled with the Yasnac Ethernet network board.
    In case of NX100, DX100 and FS100 controller transceiver is not necessary.
- Specifying network address data.
  To address data by TCP/IP protocol
  - IP address,

  - Subnet mask,
  - Gateway,
  - Server address,

  must be configured. To get this data please contact local network administrator. Yasnac Ethernet interfaces do not support automatic address distribution (DHCP).

- Testing Ethernet connection.
  - Basic command of the TCP/IP protocol suite to test Ethernet connection is the ping command. Use “ping <ip address>“ to test proper wiring and configuration of Ethernet network. First make sure this command is successful before starting any other communication task.
  - A valid configuration example can be described as follows
    - ⇒ Pc with installed network card and TCP/IP protocol. IP address: 192.168.10.10, subnet mask: 255.255.255.0.
    - ⇒ Yasnac Ethernet configuration: IP address: 192.168.10.11, Subnet mask: 255.255.255.0, Gateway: 0.0.0.0, Server: 0.0.0.0.
    - ⇒ Direct cable connection between PC and Yasnac controller with crosslink cable.
- Remote mode settings:
    - ⇒ Refer to Table 4.

4.3.4 Remote mode settings.

According to the desired communication mode the following settings are necessary:

Remote mode enabled					**Remote mode disabled**		
Command Remote	RS005	RS007	Available Functions:		RS005	RS007	Available Functions:
X	any	any	Host All-Commands				
O	1	2	Host Read-Only Plus Plus (IO r/w, File r/w, Var r)		1	2	Host Read-Only Plus Plus (IO r/w, File r/w, Var r)
O	1	1	Host Read-Only Plus (IO r, File r, Var r)		1	1	Host Read-Only Plus (IO r, File r, Var r)
O	1	0	Host Read-Only (IO r, Var r)		1	0	Host Read-Only (IO r, Var r)
O	0	any	DCI		0	any	DCI

Table 4 Remote mode settings Command Remote Function (X: Enabled, O: Disabled).

(*) Menu for „Command Remote“ setting can be reached by selecting IN/OUT -> Pseudo Input Signals in case of FS100, DX100, NX100 and XRC. In case of MRC controllers Maintenance Mode must be entered.

4.4 Step 4: Testing the connection on application level

- ▶ Installation of Motocom32 software package.
- ▶ Attaching hardware key to the parallel or USB interface of the PC.
- ▶ Start MotocomDemoVB testing application.
- ▶ Select connection type and destination address (Com port or ip address).
- ▶ 4 different tests are available:
  - Host mode.
    - Read B000
      Receives the value of B variable B000.
    - Write B000
      Writes edited value to variable B000.
  - DCI mode.
    DCI mode is only functional if a corresponding job is executed on controller side. First start robot job, then immediately hit corresponding button in test application. If the robot job is not executed a timeout will occur after some amount of time.
    - GETPOS B-VAR
      Receives the value of the B variable specified in job. On robot side the following job must be executed.
      NOP
      SAVEV B000
      End
    - SETPOS B-VAR
      Transmits value of the edit box to the variable specified in job. On robot side the following job must be executed.
      NOP
      LOADV B000
      End

4.5 Step 5: Building own applications

Building applications based on Motocom32 library with standard development environments can be divided into the following steps:

- ▶ Open new project.
- ▶ Copy Motocom32 files into application folder. These are
  - Motocom32.dll,
  - Motolk.dll,
  - Motolkr.dll,
  - Vrp32.dll,
  - Hslsrv32.dll (not necessary, if EServer mode is used),

- Motocom32.bas or Motocom.h, Motocom32.lib or MotoCom32.vb or CMotocom.cs depending on selected development environment.

▶ Import include/declaration files into project.

- Visual Basic: Motocom32.bas (contains function declares).
- Visual C++: Motocom.h, Motocom32.lib.
  Add include directive to each module (file) Motocom32 functions are called:
  #include "Motocom.h"
- Visual Basic .NET: MotoCom32.vb
- Visual C#: CMotocom.cs

▶ Using WinAPI LoadLibrary Function with Motocom32 und C/C++.
If Motocom32.lib file should not be used, calling DLL functions could also be done by using „LoadLibrary“ API function. Then only the binary files Motocom32.dll, Motolk.dll, Motolkr.dll, Vrp32.dll, Hslsrv32.dll are necessary. As an example the BscGetVarData is called as follows:

- Header file xxx.h

```
...
typedef short APIENTRY (*TBSCGETVARDATA) (short, short, short, double *);
TBSCGETVARDATA BscGetVarData;
...
```

- Source file xxx.cpp

```
...
#include "xxx.h"
...
hInstDll = LoadLibrary ("MOTOCOM32.DLL");
if (hInstDll != NULL)
BscGetVarData = (TBSCGETVARDATA) GetProcAddress (hInstDll, "_BscGetVarData@16");
...
```

- Function names
  When LoadLibrary function is used the basic function names must be used. To get the right function name Motocom.bas file can be used.

  For example the BscHostGetVarDataM function has the following declaration:

  BscHostGetVarDataM Lib "MotoCom32" Alias "_BscHostGetVarDataM@20"

  In this case "_BscHostGetVarDataM@20" is required function name. The signature (arguments and corresponding types) of the function can be seen in the Motocom.h c-header file.

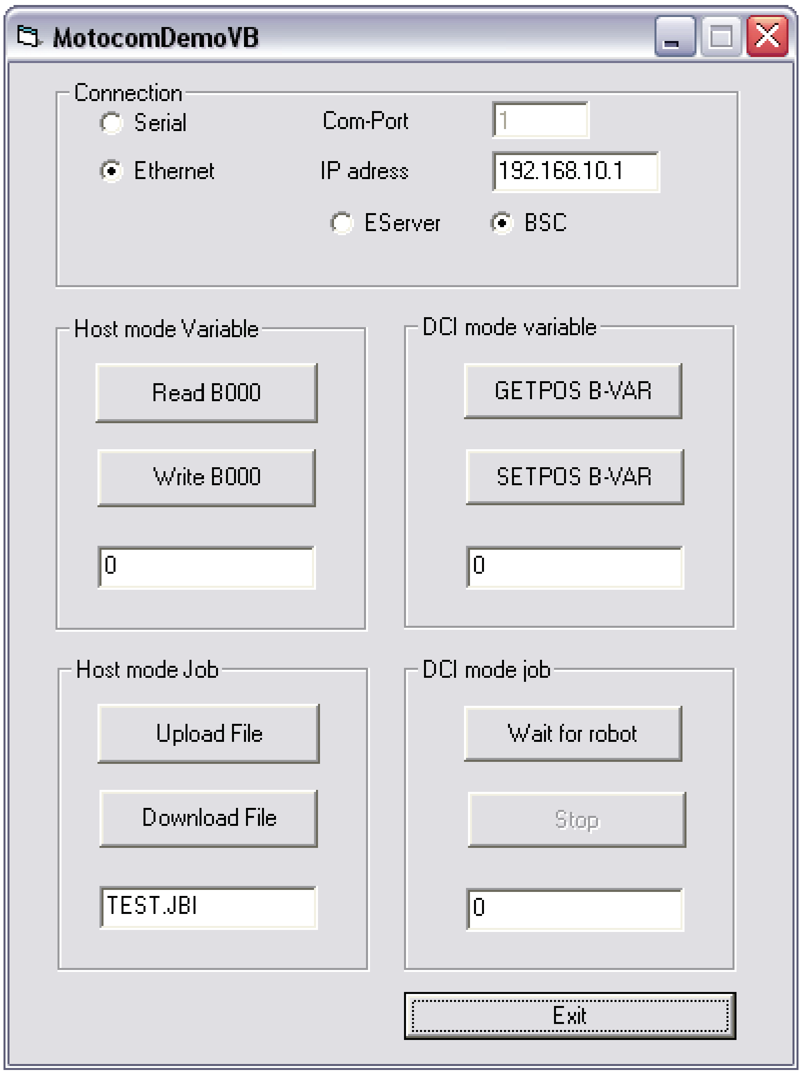
▶ Creating application frame.
A typical application frame of an Motocom32 application can be defined as follows:

- BscOpen…
- BscSetCom, BscSetEther or BscSetEServer...
- BscConnect...
- Working functions like BscGetVarData, BscUpLoad etc.
- BscDisconnect...
- BscClose...

► Demo applications written in Microsoft Visual C++, Visual Basic, VisualBasic.NET and C# are included in the Motocom32 package.

In the application folder of Motocom32 software there is a subfolder called “Samples”, where some application examples can be found.

- MotocomDemoVB



Connection: Serial [✓ Ethernet]
Com-Port: 1
IP adress: 192.168.10.1
EServer [✓ BSC]
Host mode Variable: 0
DCI mode variable: 0
Host mode Job: TEST.JBI
DCI mode job: 0

Figure 3 MotocomDemoVB

- MotocomDemoVCplusplus
  Small example in C/C++.
- MotocomDemoVCplusplus-LoadLibrary
  Demonstrates calling Motocom32 functions with LoadLibrary.
- MotocomDemoVB.NET
  Demonstrates use of VB.NET environment.
- MotocomDemoCSharp
  Demonstrates calling Motocom32 functions with Visual C# (C-Sharp).

5 Special Topics

5.1 Writing IOs

As specified in the function reference only network inputs can be written. To get access to other IO signals or to use this signals on job level ladder program must be changed.

5.1.1 Linking Networks Inputs to Universal Inputs

The following example links the first byte of the network inputs to group 20 of the universal inputs. This offers the following possibilities:

- Signals are accessible by Inform instructions in robot job (WAIT IN#, DIN)
- Special functions use input groups as parameters. For example: “Speed over ride function by external signal input”

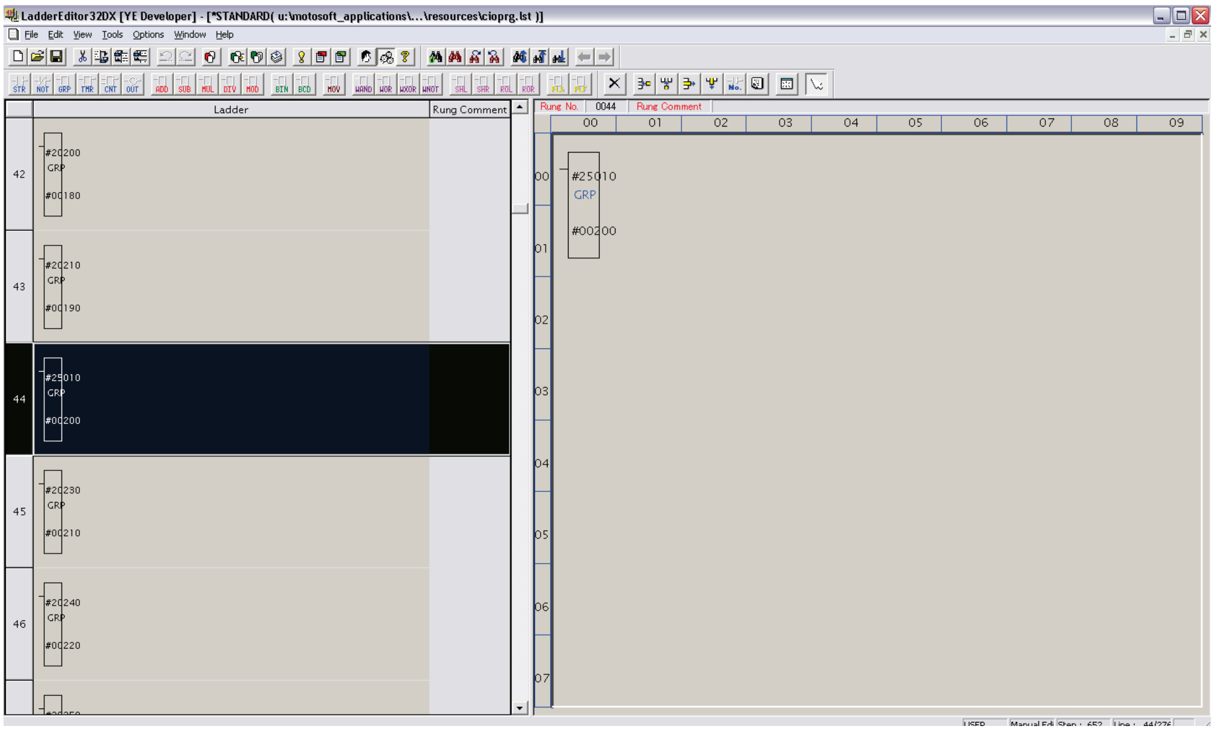


Figure 4 Ladder Editing.

<u>Example:</u>
Default setting:
GSTR #20210
GOUT #00190

New setting
GSTR #25010
GOUT #00200

After reloading modified ladder program, 1st byte of network inputs can be accessed from job level using Universal Input group number 20 (IN#0153-IN#0160).

5.1.2 Linking Network Inputs to External Outputs

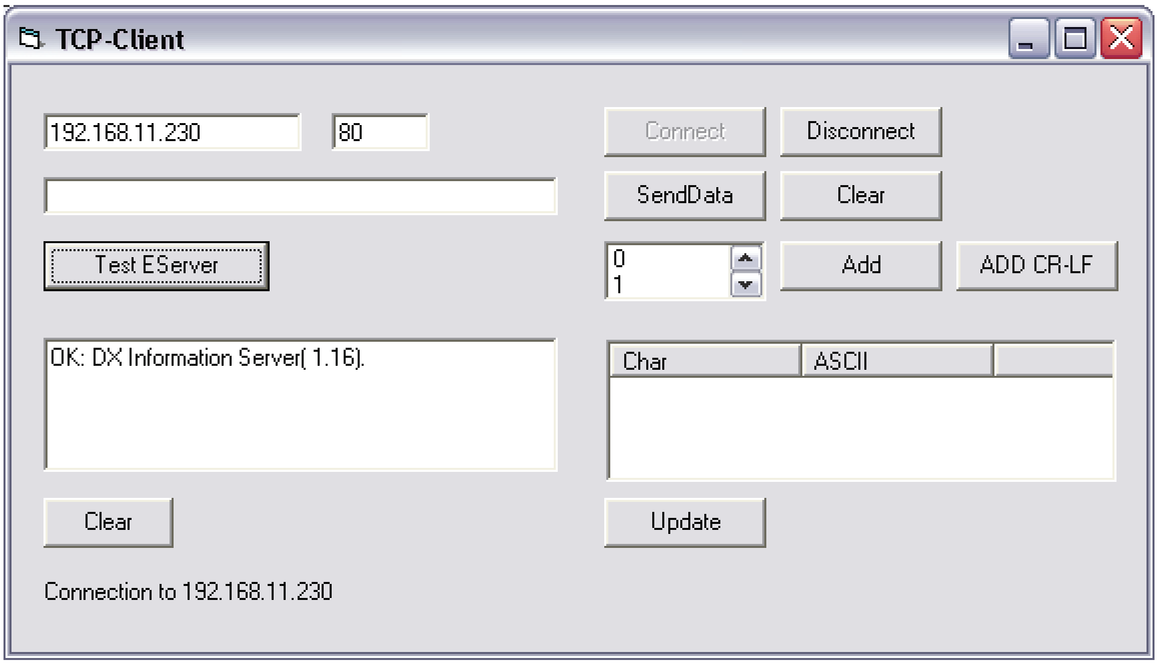
To control physical outputs same procedure has to be done. But instead of linking the network inputs to Universal Outputs, this time External Outputs are used.

Example
Default setting:
GSTR #10210
GOUT #30230

New setting
GSTR #25020
GOUT #30230

5.2 Checking EServer mode

To check if the EServer mode is activated or not a small tool (TCPClient.exe) can be used. This tool can be found in the application folder in the subfolder "Tools".



OK: DX Information Server(1.16).

Connection to 192.168.11.230

Figure 5 EServer testing tool.

To check EServer mode following steps are necessary:
- Insert IP address of your controller.
- Insert Port number 80.
- Push "Connect" button.

- ► Push “Test EServer” button
- ► Immediately controller should respond with “OK…” message.

5.3 Checking “Command remote” mode

“Command Remote” mode is required to have full access to the robot controller (refer to Table 4). To check if “Command Remote” mode is activated the following signals must be reviewed:

XRC	SOUT0038
NX100, DX100, FS100	SOUT0039

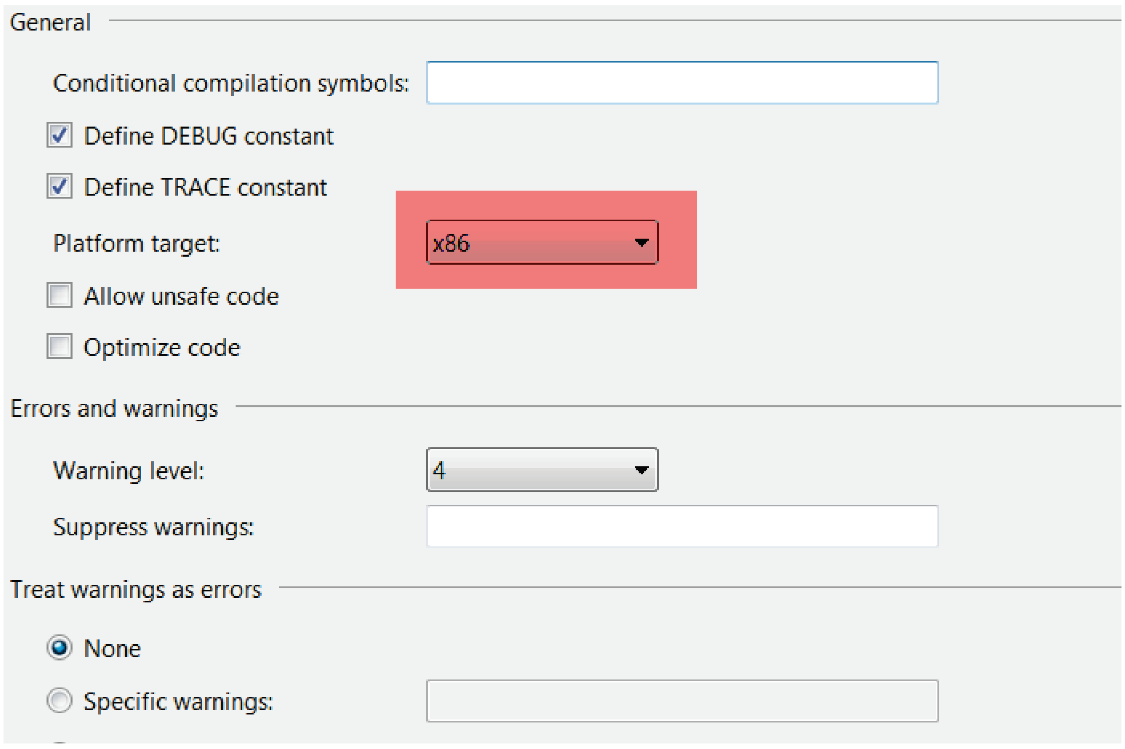
The presence of “Command Remote” mode can also be checked from outside by using Motocom32 function BscGetStatus().

6 Known issues

6.1 General

6.1.1 Running 32Bit DLL on 64Bit operating system in .NET environment (C# sample)

By default .NET applications are compiled for “Any CPU”. This setting does not work if a 32Bit Dll is called. Therefore Platform target has to be changed to “x86”.



General

Conditional compilation symbols:

[✓] Define DEBUG constant

[✓] Define TRACE constant

Platform target: x86

[ ] Allow unsafe code

[ ] Optimize code

Errors and warnings

Warning level: 4

Suppress warnings:

Treat warnings as errors

● None

○ Specific warnings:

6.2 EServer and BSC mode

6.2.1 Limited Access during alarm

If there is an active alarm, accessing of variables is not possible. File and IO access is still possible.

Folgende Änderungen sind für eine Veröffentlichung des Handbuchs für Motoman Europe nötig (eine Bearbeitung der pdf-Datei nicht möglich):

6.2.2 BscDownloadEx returns wrong return value

It is recommended to use BscDownload instead.

6.2.3 BscDownload, BscDownloadEx

It is not possible to overwrite a job on the controller with this function. To do so, please execute BscSelectJob and BscDeleteJob first.

6.2.4 BscUpload, BscUploadEx

Files on the PC will be overwritten without warning."

It is recommended to verify the existance of the file on controller side before executing this function. To do so, please use BscFindFirst and BscFindNext."

6.2.5 BscPutVarData, BscPutVarData2

It is recommended to use BscHostPutVarData or BscHostPutVarDataM instead.

Please check if the variable value is within the valid range before executing this function.

6.2.6 BscGetVarData, BscGetVarData2

It is recommended to use BscHostGetVarData or BscHostGetVarDataM instead.

6.2.7 BscHostPutVarData, BscHostPutVarDataM

Please check if the variable value is within the valid range before executing this function."

6.2.8 BscHostGetVarDataM

Maximum number of variables: 15.

6.2.9 BscGetError2, BscIsErrorCode, BscGetFirstAlarm, BscGetNextAlarm

It is recommended to use BscGetStatus for retrieving the current status (active alarm or error) and then use BscReadAlarmS, BscGetFirstAlarmS, BscGetNextAlarmS instead.

6.2.10 BscIsLoc

It is recommended to use BscIsRobotPos instead unless pulse data is required.

6.2.11 BscIsRobotPos

Bei "Arguments *framename Coordinate name" ergänzen: "(case-sensitive)"

6.2.12 BscSelectJob

The job name must contain the file extension (".jbi").

This function is not applicable to test for existence of a certain job. No communication takes place between Motocom and controller.

6.2.13 BscOpen

BscOpen changes the current directory to <*path>.

6.2.14 BscConnect

BscConnect should not be used for connection checking. Even if the return value equals 1 the target IP address might be wrong.

6.2.15 BscStatus

It is recommended to use BscGetStatus instead.

Imprint

Yaskawa Europe GmbH
Kammerfeldstraße 1
85391 Allershausen
Germany
Phone 00498166900
Fax 0049816690103