FESTO

CIROS®
Robotics

User's Guide

572753 EN
01/2010

Order No.: 572753
Edition: 01/2010
Authors: Ulrich Karras
Graphics: Ulrich Karras
Layout: 01/2010 Susanne Durz

Internet: www.festo-didactic.com
e-mail: did@de.festo.com

Contents

1. Introduction

Welcome to the new release 1.1 of CIROS® Robotics. This new version includes a completely new collision detection module, see chapter 5.1, and a new modelling of the work pieces for the MPS robot stations. As well all changes of the Studio version 1.1 were integrated. In particular, the software is now also running under Windows7®.

CIROS® Robotics provides you with a virtual learning environment in the field of robotics. Step by step, you'll be able to advance independently from very simple robotics applications right through to highly complex work cells in a highly realistic, simulated 3D work environment. The virtual learning environment consists of:

- Programming and simulation environment for predefined robotic work cells that represent typical industrial applications
- The Robotics Assistant online tutorial offering comprehensive robotics knowledge via multimedia presentations

The Robotics Assistant is not a CBT (computer based training), but rather a multimedia information system that provides teachers with support in designing courses of study, and that can be used by trainees for autodidactic learning.

You decide yourself how you'll proceed with your course of study. With its integrated library of work cells, CIROS® Robotics provides you with an introduction to robotics covering various degrees of complexity. The library of work cells encompasses innumerable examples of typical industrial robotic work cells, including appropriate function descriptions and technical documentation. A sample application is included for each work cell, and instructions are provided for implementing each respective application. You can decide whether or not you'd like to install the sample solutions while installing the software. Of course you'll also have the opportunity of developing and solving a host of other tasks for any or all of the predefined robotic work cells.

The library of work cells got an additional structure as follows:

- Intro models
- MPS robot stations
- iCIM robot stations
- Special robot applications
- Production systems
- microFMS

The CIROS® Robotics leaning environment provides you with user help in a number of ways. The CIROS® online help function is based on the standard HTML Windows help format. The Microsoft Internet explorer (version 5.0 or higher) is required in order to use the help function. The help system was created by Help&Manual which enables the user to extend the help.

This new release incorporates many of the comments and suggestions we have received from CIROS® Robotics users. In order to continue improving CIROS® Robotics, we invite all future users to send us their comments, suggestions and criticism as well. We would also be happy to answer any questions that might arise regarding CIROS® Robotics. Just send us an e-mail at: dka@de.festo.com

1.1 The CIROS® 3D-simulation system

CIROS® is an industrial 3D simulation system for PC based operating systems including Windows 2000® and XP®/VISTA®/Windows7®. CIROS® facilitates the planning of robotic work cells, testing the reach ability of all required positions, the development of robotics and control programs, and layout optimisation. All motion sequences and handling operations can be simulated in order to rule out the possibility of collision, and to optimise cycle times.

Work-cells can be created using library components such as machines, robots, tools, assembly lines, loaders and more with the help of CIROS® model expansion modules. You can also create your own work cell components, and import part models and work pieces from other CAD systems.

1.2 Overview

- The **user interface** will be described in detail in chapter 3.
- The structure of the **library of robot work cells** is subject of chapter 2.4.
- There are two ways to open the work cell in order to work with it
  – Reference model
  – User model
  The reference model is write-protected. This allows you to present the work cell always in the original status. If you want to do changes or to create new programs you should select the user model. The user model can be opened in user specified folders, see chapter 3.2 for further details.
- Programs and position lists can be opened and edited in any text editor. As well CIROS® Robotics provides a print function to print out complete projects, see chapter 4.3 for further details.
- The documentation of the sample work cells includes a new section **Coordinates**. This section provides the dimensioning of all the layout of this work cell including all integrated components. This helps you to create interesting new tasks related to a modification of the layout of the work cells.
- Note that any change of the layout cannot be saved in CIROS® Robotics.
- The concept of **Automation Suite** overcomes this limitation of the Robotics version. Applying CIROS® Studio (which replaces the old version COSIMIR Professional) you may modify the predefined work cells or create new ones and release them for using in CIROS® Robotics.
- **Licensing** is exclusively done via one USB license key. This license key can be online modified at any time.

1.3 System requirements

Minimum configuration

- PC operating system: Windows 2000® and XP®/VISTA®/Windows7®
- Microsoft Internet Explorer at least version 5.0
- Processor: Pentium IV 1 GHz
- RAM: 512 MB
- Hard Disc Space: 5 GB
- Graphic Adapter: Any card supporting OpenGL; 3D-acceleration increase performance; 128 MB RAM
- Monitor: 17" with 1024 x 768 resolution
- DVD-ROM drive
- USB-interface

Recommended configuration

- PC operating system: Windows 2000® and XP®/VISTA®/Windows7®
- Microsoft Internet Explorer at least version 5.0
- Processor: Intel Core Duo 2,2 GHz
- RAM: 1 GB
- Hard Disk Space: 10 GB
- Graphic Adapter: NVIDIA 7800GTX, 512 MB RAM
- Monitor: 19" with 1280 x 1024 resolution
- DVD-ROM drive
- USB-interface
- Ethernet interface
- Internet access
- Email-client with email-account on the PC to arrange online upgrade of the license key

1.4 Installation Instructions

The product package CIROS® Robotics consists of a DVD, a manual with comprehensive installation instructions, this user guide as pdf-file on the DVD and a USB license key. You may separately order this user guide as a print out version. The installation does not need a license key. The license key is only required for running the software. You may find all further details in the installation instruction manual.

2. Concept of CIROS® Robotics

2.1 Didactic Concept

CIROS® Robotics software is based upon the concept of an open learning environment.
Open learning environment means:

- An open approach to learning characterised by constructivism, i.e. various tools including basic knowledge, a lexicon and simulations are made available which can be combined and utilised as desired in accordance with your own learning objectives.

This open concept has also been implemented in organising the basic knowledge. The central topic is robotics, which is why we call it the **Robotics Assistant**. It's not laid out as a CBT or a WBT, but rather as an interactive, multimedia knowledge and information system. The contents of the program are presented as individual information modules including:

- Texts (concepts, explanations, regulations, examples etc.)
- Graphics
- Videos and animations

The information modules are interconnected by means of hyperlinks.

The Robotics Assistant provides you with various options for accessing information in a targeted fashion:

- Searches for keywords or topics
- Tree structure navigator
- List of selected topics

Selected information can also be printed out at any time.

Why have we selected this open concept for imparting knowledge?

- We do not perceive the acquisition of knowledge and information as an end unto itself, but rather as a necessity for solving problems.
- The project task or the problem to be solved are at the heart of our concept, resulting in the need to acquire new knowledge in order to solve the problem at hand.
- Acquiring knowledge and information with modern methods based on software technology is one of the central learning tasks in today's technological society.

A further didactic concept is the provision of virtual work environments in the form of simulated robotic work cells. These are represented in 3 dimensions in order to create as realistic an image as possible.

- Options for experimenting with the work cells effectively place the trainee in a close relationship to the object under study. Knowledge is tested and reinforced.
- Realistic experience provided by the work cell gives rise to a new quality of knowledge: theoretical knowledge is transformed into practical application and skills.
- The work cells promote learning by discovery at different levels of difficulty (it works, it doesn't work, it works more efficiently etc.).

Before you start to work with a robot cell you have the option to study and analyse the movement of a robot system without integration into a complex environment, see chapter 4.

2.2 Approach and Learning Goals

Robotics is a fascinating, but at the same time highly complex and intricate technology. We restrict ourselves here to the field of industrial robotic systems, and the area of mobile robotics will not be addressed at all.

Target groups and prerequisites

Our approach is aligned to vocational training in the following areas:

- Mechatronics
- Various technical qualifications for metalworking and electrical engineering
- Information technology

Our approach is aligned to technical colleges and universities. We also assume that you, the trainee, are familiar with the Windows PC environment.

Trainees must be equipped with certain basic knowledge in order to get started in the field of robotics. The Robotics Assistant provides comprehensive basic knowledge on the subject of industrial robots (see chapter 2.1 above). The Robotics Assistant makes it possible for the trainee to:

- Acquire basic knowledge independently, and in a targeted fashion
- Prepare for problem solving tasks
- Retrieve, and if necessary print out additional information during the problem solving stage

We also provide teachers with the opportunity of using the Robotics Assistant as a multimedia supplement to their own course. Thus CIROS® Robotics assists you in organising the basics for your projected learning approach in a highly flexible way right from the beginning of the introductory phase. We recommend covering at least the following subjects with the Robotics Assistant for introductory courses:

- Definition of robots including characteristic values
- Robot design with subchapters covering hardware, different types of robots and work safety
- Robot programming languages

That which has been read or heard can then be subjected to practical testing, analysed and implemented by the trainee in his work with the numerous virtual robotic work cells. Of course we are aware of the fact that a virtual work cell is not capable of imparting all of the many aspects of this technology. Problems associated with drive technology, accuracy and dynamics are not taken into consideration in the simulations. For this reason, we also offer the respective hardware environments for several of the work cells:

- BP70
- MPS® RobotStation
- MPS® RobotAssemblyStation
- MPS® PunchingStation

- microFMS model MTLR 10: Work cell with a CNC milling machine, a CNC lathe machine and the
- Mitsubishi RV-2AJ mounted on a linear axis
- iCIM station Assembly RV-1A: Assembly work cell with robot, image processing and various magazines
- iCIM station Assembly RV-3SB: Assembly work cell with robot, image processing and various magazines
- iCIM station Assembly RH-5AH55: Assembly work cell with SCARA robot, image processing and various magazines
- iCIM station Mill55: CNC feeding station with robot and milling machine
- iCIM station Turn55: CNC feeding station with robot and lathe machine
- iCIM station Mill and Turn RV-3SB: FMS work cell with robot, CNC milling and CNC lathe machine.

An ideal learning environment can be created with these work cells by fulfilling the following basic requirements:

- At least one real robotic work cell
- One workstation (learning station) with CIROS® Automations Suite
- Each trainee has their own CIROS® Robotics license

In this way, each trainee has the opportunity of downloading their program to the robot controller at the real work cell, and can start up and run their own solution to the specified problem at the actual system.

2.3 Learning via Virtual Work cells

The virtual work cells create an experimental environment for trainees, allowing them to experience and grasp the required basic knowledge. At the same time, they are a point of departure for the examination of new questions and problems, i.e. for building upon existing knowledge.

An explorer (see figure 2.1) provides you with direct access to all virtual work cells. Selection of the title of a work cell in the explorer tree will open the descriptions corresponding work cell

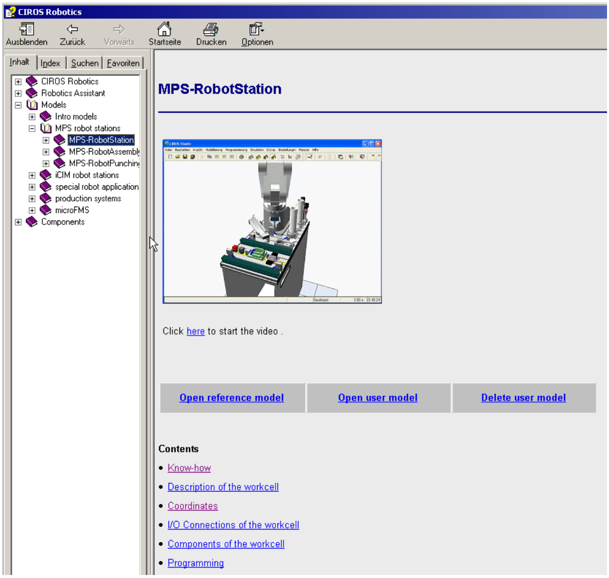


Figure 2.1: Documentation

An animation is started by clicking the model image, and the respective work cell is demonstrated by means of a simulation sequence. The trainee is thus provided with visual support in addition to the work cell's function description. The following additional information can be displayed:

- **Learning objectives (know how)**
  Here we've listed the typical learning objectives that can be realised with the respective work cell as examples. Of course it is also possible to establish additional objectives with the selected robotic work cell, depending upon the specified tasks.
- **Description of the work cell**
  This section provides a function description of the work cell, creating the basis for the generation of one's own tasks.
- **Components of the work cell**
  This section contains a brief technical documentation of the most significant components included in the respective work cell.
- **Coordinates**
  This new section provides a graphic of the layout of the work cell with a list of coordinates of all components, see figure 2.2. It allows you by changing the coordinates to create easy new tasks.

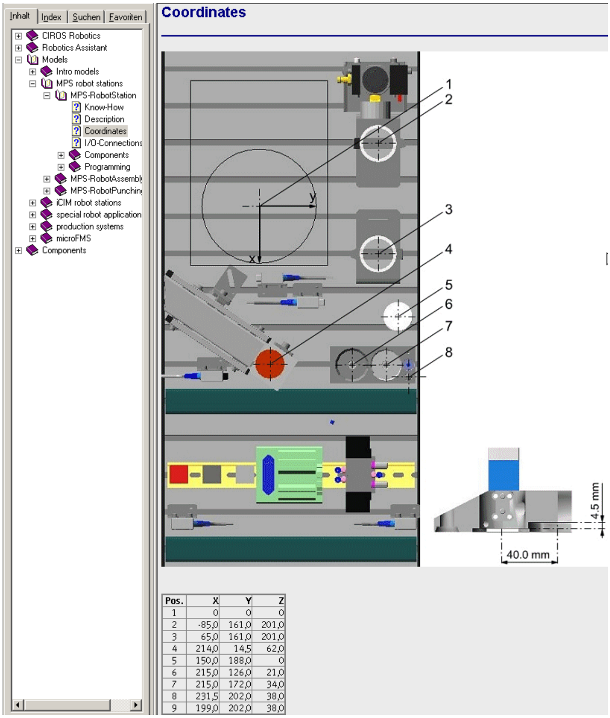


Figure 2.2: Coordinates of components

- **I/O connections**
  Here you'll find a commentated list of I/O assignments for the robot controller, as well as for the PLC if included.
- **Programming**
  The structure of the sample program is explained, and helpful hyperlinks and tips are provided for program writing.

In addition to, and independent of the included hyperlinks, you can also access any explanations of terminology and basic theory which you need necessary for your problem solving task at any time with the help of the assistant explorer or the index and search functions.

Introductory work cell

Which work cell should you start with? If you have no previous robotics knowledge, we recommend beginning with the "First Steps" and "Next Steps" models. In the First Steps work cell, simple rectangular workpieces can be picked up from a table, moved to a pallet, and finally positioned on a second pallet. A glass plate is located between the two pallets in the Next Steps work cell, and an alternate position must thus be added to the pick & place sequence in order to avoid possible collision.
Work-cells with either the Mitsubishi RV-2AJ robot or the RV-M1 predecessor model can be selected. The RV-2AJ can be programmed with the modern, high-level Melfa Basic IV robot language, whereas the simple command language, Movemaster Command (MRL), must be used with the older RV-M1. We only recommend the model with the RVM1 robot if your hardware environment also includes RV-M1 robots.

Before writing a robot program, you must first learn to move the robot and actuate its gripper. The robot can be moved within various coordinate systems:

- Joint coordinates
- World coordinates
- Tool coordinates

The various coordinate systems can be visualised in the work cell window, see chapter 3.4. Robots can be set into motion with a so-called teach panel. A universal control module is also replicated in the simulation by means of the Teach-in window and can also be used to move the robot, see chapter 3.4. For example, attempt to move the robot by simply changing the axis coordinates such that it is able to securely grasp a work piece with its gripper.

Three-dimensional navigation

Three-dimensional navigation within the work cell presents you with an additional problem. The representation of the work cell changes depending upon the point of view:

- From the top left or top right
- From the front or the back
- From up close or far away

At least two different views are required for trouble-free, three dimensional orientation. With CIROS® Robotics, the number of views is only limited by the performance characteristics of your PC.

Robot motion

You'll discover that it's quite advantageous to make use of motion within the other coordinate systems in order to grasp a work piece. On the other hand, each movement executed by the robot is the result of coordinated motion of the individual joints. These can be viewed in the status window, see chapter 3.4, for example in order to observe the means by which axes must be moved in order to advance the gripper along the X-axis in the world coordinate system. In order to execute the gripping operation, the gripper must be appropriately oriented. Consider whether or not restrictions would result in this area through the use of a 5-axis articulated robot?

The position list

Now that you've brought the robot into a position from which it can grasp the work piece with its gripper, you can save this point to the position list. The position list contains all of the points to which the robot must move directly for a given program, as well as important ancillary points for moving along a path (mid-point, diverging point etc.).

Why is a position list so important?
One could argue that as long as the cell is known, any desired point can be calculated. Why, then, should the robot first move to certain teaching points? The answer is quite simple: As a rule, industrial robots demonstrate very good repetition accuracy, but their absolute positioning accuracy is entirely inadequate for most applications. Further details are included in the Robotics Assistant.

One of the main tasks during commissioning of a robotics application is testing the position list, i.e. positions established in the simulation are tested via the real system, and are modified if necessary. It is thus extremely important for trainees to become well acquainted with the teach-in procedure in the simulation.

Each work cell has its own position list which you can take advantage of in order to reduce the time required for teaching in all of the positions.

The first robot program

As is also the case with the teach-in procedure, two different types of motion commands are also used for programming robot motion:

- Movement from a starting point to an end point, which is known as point-to-point movement (abbreviated PTP). The actual path to the robot's end point is not defined, because all axes travel to their endpositions independent of one another.
- Movement of the robot to the end point via a predefined path (for example along a straight line).

Sample task

The blue workpiece in the First Steps model must first be set onto the middle section of the first pallet. After a waiting period of 2 seconds, it must then be sorted into the bottom section of the second pallet.

First, a sequence plan is created for the program:

Sequence plan	
1	The robot's gripper is open.
2	The robot moves the gripper to the gripping position (blue work piece) with a PTP movement
3	The gripper is closed.
4	The robot moves the gripper to the middle section of the first pallet with a PTP movement.
5	The gripper is opened.
6	The robot moves linearly back to a point above the first pallet.
7	2 second waiting period
8	The robot moves the gripper back to the middle section of the first pallet (linear movement).
9	The gripper is closed.
10	The robot moves the gripper to a point above the final position with a PTP movement (for safety reasons).
11	The robot moves the gripper to the final position (linear movement).
12	The gripper is opened.
13	The robot returns to its initial position with a PTP movement.
14	End

Of course the robot's controller is unable to understand this text, which must be translated step by step into, for example, the Melfa Basic IV programming language:
10 HOPEN 1
20 MOV P1, -30 "P1 = gripping position"
etc.

The sequence plan should be laid out such that each step can be implemented by means of a command or a subprogram. At the same time, the sequence plan provides you with ideal documentation of your program. Details regarding Mitsubishi programming languages can be found in the "Programming" chapter included in the CIROS® help function.

Downloading to the robot controller

The program has now been created, and must be downloaded to the robot controller. This procedure can be replicated in CIROS® Robotics, because the simulation includes a fully fledged robot controller. The downloading procedure is completed in two steps:

- Compile the program, i.e. the syntax of the programming language is checked and is translated into universal IRDATA machine code.
- The machine code is downloaded to the robot controller, i.e. the code is linked to the controller.

Any errors that might occur are displayed. The details for this procedure are presented in chapter 4.2.

Simulation

The program has now been downloaded without error to the robot controller. Start the program and observe the 3D motion sequence. You can select either the

- automatic mode

or the

- single step mode,
  and you're able to determine whether or not the sequence is executed in a logically and functionally correct fashion.

Collision detection

If the sequence is error-free, you should then check to see if any undesired collisions occur. The Next Steps model is used to illustrate this procedure, which includes an additional glass plate between the two pallets. Start collision detection (see chapter 5.2), and then start the above described program. If a collision occurs, the robot's path must be suitably changed. Check to see whether or not any other collisions might occur during the sequence. Why, for example, does the robot move to a point above the final position for safety reasons in the above represented sequence plan? Check the other movements and gripping positions with this in mind.

Flexibility in designing tasks

Various problems can be posed for each of the work cells. The layout of any given work cell can be readily changed with the help of the model explorer (see chapter 6.3), for example the pallets or the work pieces in the First Steps model can be repositioned. Can the tasks still be completed after repositioning?

Note that changes of the layout cannot be saved. You can easily solve this problem with the Automation Suite. Open the work cell in CIROS® Studio. Do all changes you are looking for. Save and release it for using this new work cell in CIROS® Robotics.

The IRL programming language

Robot systems from various manufacturers use different programming languages, although there is a standardised, universal robot programming language known as IRL (industrial robot language). We have selected the following didactic solution for CIROS® Robotics:

- We offer the Melfa Basic IV high-level programming language or the simple MRL command language for all robotic work cells with Mitsubishi robots.
- If the work cell does not include any Mitsubishi robots, we offer the standardised IRL language. Please note that the robots in these cells can also be programmed with Melfa Basic IV, but not all of the language's attributes will be supported in this case.

Keep in mind that IRL is a significantly more complex language than Melfa Basic IV. Details regarding IRL are included in the CIROS® help function under "Programming".

Sensor technology

A robot can only be used flexibly if it is capable of communicating with its work environment. The analysis of sensor signals is utilised to this end. We have provided numerous work cells for this purpose. We recommend the BP70 model, and in particular the MPS® RobotStation, for getting started with this subject matter. In this work cell, the positions of objects are detected by the robot in an elementary way, and are evaluated for further processing. You can make use of a simulation box with 8 inputs and outputs that are connected to the robot's controller in the BP70 work cell.

2.4 The Work cells

The sequence in which the work cells are laid out is organised such that, as a rule, knowledge gained in working with previous work cells is very helpful in solving the problems posed by subsequent work cells. However, if the trainee has prepared himself adequately, the work cells can be processed in any other desired order. In any case, before you begin work with any given work cell, you should carefully examine the respective video animation, as well as instructions regarding programming and I/O connections, and included component descriptions.

The library of work cells is structured in sub libraries such that you get a more comfortable overview.

- Introduction models
- MPS robot stations
- iCIM robot stations
- Special robot applications
- Production systems
- microFMS

Introduction models

FirstSteps / NextSteps

These robotic work cells have already been described in detail in chapter 2.3. They are available with RV-2AJ and RV-M1 robots. The sample programs for the RV-2AJ have been created with Melfa Basic IV, and for the RV-M1 with MRL.

PickandPlaceABB

The PickandPlaceABB.mod work cell includes a very simple handling task with a type 2400-16 ABB robot, which serves as a basis for all further tasks. Simple examination of the working space can be executed with this work cell by repositioning the robot and the pick & place library component. The sample programs are written in IRL.
Please note that it is very easy to replace the robot included in the work cell with any other robot from the robot library (see also chapter 6.4) using CIROS® Studio of the Automation Suite.

PickandPlaceFesto

A similar handling task is implemented with this work cell using a Festo 2-axis pneumatic linear system. This model is also available with sensors that detect the objects to be handled. The sample program is written in IRL.

PalletAssembly

The PalletAssembly.mod work cell includes a Mitsubishi RV-2AJ robot, that has the task of filling a pallet with work pieces. This is also a handling task, but robot movements take place to calculated positions as well as to predefined positions in this case. For this task, it is also useful to introduce the programming of loops. Feeding work pieces from a magazine necessitates additional I/O interrogations. The sample program is written in Melfa Basic IV.

BP 70

This work cell is available with the Mitsubishi RV-M1 robot, as well as the RV-2AJ. It additionally includes two work piece holders, one tool holder with tool, a pallet with work pieces and a simulation box with 8 inputs and outputs. A large number of different tasks can thus be executed with the work cell:

- Handling task
- Machining task
- Palletising task

The tasks section in our "Basic Robotics" workbook includes concrete task suggestions. This was the first robotic work cell offered by Festo as part of the MPS® product range.

MPS robot stations

MPS RobotStation

This work cell is a simulation of the new MPS® Robot Station, and is equipped with the RV-2AJ robot. Geometric data are based upon a CAD import of the associated design engineering data. The station performs the following task sequence:

- Remove work pieces from the seat in a chute after a signal has been generated.
- Determine the material characteristics of a workpiece held by the robot's gripper with the help of a sensor.
- Detect the position of workpieces and set them down correctly orientated at an assembly point. There are holes in the bottom of the work pieces such that the collision detection enables you to check if the adjusting pin for assembly exactly fits in one of the holes.
- Sort work pieces into magazines according to material characteristics.

This is the standard MPS® robotic work cell. The sample program is written in Melfa Basic IV.

MPS RobotAssemblyStation

This work cell is a simulation of a combination including the new MPS® "Robot" and "Assembly" stations. The combination replaces the above described assembly station. The "Assembly" station is controlled by a simulated S7 PLC, or by the robot controller. A comprehensively documented sample program is available for both variants. The programs are identical to the respective programs for the real robotic work cells. The task consists of assembling model cylinders from the following components:

- Cylinder housing
- Piston
- Spring (piston return spring)
- Cylinder cap

Through the use of various cylinder housings (red, black and silver), it is possible to assemble various cylinders with different piston diameters (identified by the colours black and silver).

MPS RobotPunchingStation

This work cell is a virtual representation of a combination including the three MPS® stations "Robot", "Assembly" and "Hydraulic Punch". As before, the "Assembly" station is controlled by a simulated S7 PLC or the robot controller. The hydraulic punch is controlled by a simulated S7 PLC. The hydraulic punch produces the cylinder caps in this combination station. Blank caps are fed to the punch from a cap magazine. The hole for the piston rod is then punched into the cylinder cap and the cap is set into a tray.

iCIM robot stations

Station Assembly RV-1A

The Robot assembly station is responsible for the assembly of various types of deskset. First a baseplate is positioned at the assembly position, then the first instrument is inserted into the baseplate. The camera system checks the orientation of the instrument - the robot turns the instrument to the correct orientation. Once the first instrument has been inserted into the baseplate and the orientation has been corrected, the second instrument is inserted and oriented in the same way. The instruments come from the magazines. A penholder - aluminum or brass depending on the order - is inserted into the baseplate. The penholder comes from a pallet. Finally, the pen is inserted into the penholder. The pen comes from the station magazine. The assembled deskset is brought back to its retrieval position and is placed in the automatic storage/retrieval system for further use.

Station Assembly RV-3SB

In this work cell a similar task will be realized using the Mitsubishi robot of type RV-3SB.

Station Assembly RH-5AH55

In this work cell a similar task will be realized using the Mitsubishi SCARA robot of type RH-5AH55.

Station Mill55

The FCM 56 station comprises two functional units (CNC feed station with Mitsubishi RV-1A robot and EMCO MILL 55 lathe). The first functional unit is the feed station for the machine tool, the second the CNC lathe. Work pieces are handled by the Mitsubishi RV-21A robot.

- The unmachined parts are delivered on pallets with a baseplate or penholder.
- After machining, the parts are placed back on their pallets.
- There are three pallet buffer stations.

Baseplates and penholders can be produced with this CNC machine.

Station Turn55

The FCT 56 station comprises two functional units (CNC feed station and PC Turn 55). The first functional unit is the feed station for the machine tool, the second the CNC lathe. Work pieces are handled by the Mitsubishi RV-1A robot.
The unmachined parts for penholders are delivered on pallets. After machining, the parts are returned to their original positions. There are three pallet buffer stations. This CNC machine is used to manufacture penholders

Station Mill and Turn RV-3SB

The CNC station comprises three functional units
(CNC feed station, CONCEPT MILL 155, CONCEPT TURN 155).
The first functional unit is the feed for the machine tools, the second is the CNC milling machine and the third is the CNC lathe. Workpieces are handled by a Mitsubishi RV-2A robot. The robot is mounted on a linear axis, as the distance between the CNC lathe and the CNC milling machine is greater than the operating range of the robot. All required positions can be easily reached using this linear axis.

The unmachined baseplate and penholder parts are delivered to the Mill & Turn station on pallets from store or from station magazines for machining. Following machining, the parts are returned to their original locations. The station has four buffer locations for the pallets.
The milling machine can machine the baseplate and the penholder. The lathe machines penholders made either of brass or aluminum - and in various designs.

Special robot applications

Model LabAutomation

This model shows the simulation of a robot executing different tasks in a chemical laboratory.
A gripper exchange system is attached to the endeffector of the robot (Object name RV-E2). With this system the robot can grasp different tools like a pipett, a gripper, and a measurement tool. At first using the pipett the robot mixes two chemical solutions. Afterward it puts the pipett into a cleaner. Second task is to move the test tube with the mixed solutions to a magentic mixer. The third task includes the measurement of the pH-value of the mixed solution.
To control the different tools the robot uses its connected I/Os.

Model Packaging

The work of a palletizing robot is simulated in this workcell. A vacuum gripper (Object name VacuumGripper) is attached to the endeffector of the robot (Object name Palletizer). By this gripper packages are grasped and moved from a conveyor belt to a pallet. The packages are created by a replicator mechanism (Object name Replicator).
The connected I/Os of the robot are used to request new packages and detect a package at the end of the conveyor belt.

Model Disassembly

The bolts must be removed from an automobile wheel using a Reis RV-16 robot in the Disassembly.mod work cell. An inductive sensor is used to determine whether or not the robot is using the right socket wrench to remove the bolts. Programming must be written in IRL. Knowledge of procedural and modular programming must be acquired. Sensor interrogations must also be incorporated into the communications sequence in this work cell.

Model Disk Test

Hard disks are tested at four different stations with the Disktest.mod work cell. The test stations perform a surface inspection of the metal coatings on the disks. This task can be expanded with a requirement for calculating the Cartesian coordinates of the disks, and corresponding organisation of generated work cell data into data structures as part of the programming.

Model RobWeld

This work cell simulates the actual Festo Didactic FMS welding station. Welding is performed by a Kawasaki FS03N robot. The gripper system consists of a pneumatic 3-finger gripper and a welding torch, which is connected to the robot flange via a collision-shutdown device for safety reasons. The task is to weld three raw metal components together into a cylinder housing. This can be accomplished by means of spot welding or path welding. The sample program is written in IRL, and executes a spot welding sequence. A glass shield for the prevention of electroophthalmia must be brought into position during welding for safety reasons. The welding torch must be cleaned after welding.

Production Systems

Model PressAutomation

The workcell demonstrates how to interconnect a Siemens SIMATIC S7 PLC that controls a press and two KUKA KR 125 robots.
One of the robots has a blue base plate and a blue gripper. Therefore it is called blue robot (object name KR125Blue). The blue robot takes a door from a magazine and inserts it into a press. Before the robot can insert the door into the press, he sends a signal to the PLC that controls the press, to open the press. He waits until the press is open, inserts the door and finally starts the pressing.
The other robot with a yellow base plate and a yellow gripper is called yellow robot (object name KR125Yellow). The yellow robot waits until the door is pressed and the press is open again. Afterwards he grips the door, transports it to a table and deposits it on the table.
The PLC controlled press communicates with the two robots via digital inputs and outputs.
With a mouse click on to the cube (Object: SwitchNewDoor) in the middle of the workcell, the user can insert an new door into the magazine of the blue robot and simultaneously remove a pressed door from the table of the yellow robot.

Model PCBMounting

The PCBMounting.mod work cell is highly demanding, and is well suited for project work. It simulates a PCB production line which consists of 6 work stations:

- Station for inserting ICs
- Station for soldering ICs
- Station with three robots which position the PCB holder
- Station for assembling the PCB to the holder
- Station for screwing the PCB to the holder

The individual robot programs must be created. Finally, master controls must be developed which coordinate the individual actions.

Model PlantSimulation

The PlantSimulation.mod work cell simulates an entire production facility that consists of several manufacturing cells:

- The AGV work cell includes an automated guided vehicle system (AGVS) that interconnects the individual manufacturing cells within the entire production facility. The AGVS receives picking orders which it fulfils autonomously. The work cell consists of the AGVS, a robot and various work piece carrier trays with sensors.
- The Workshop work cell consists of two Mitsubishi robots, one of which is mounted to an additional linear axis. The robots must execute simple handling tasks in a work-order related fashion.
- The Storage work cell controls automated warehousing. It is linked to the AGVS by means of a conveyor belt.
- The Production work cell consists of a robot, an injection moulding machine, a press, a laser labelling unit and a conveyor belt which links it to the AGVS. A ventilator fan base is produced in this work cell.
- The ventilator fan base must be painted in the Paintshop work cell, which consists of a robot, a rotary table, a gripper changeover module for grippers with various paint spray guns for different colours, and a conveyor belt which links it to the AGVS.
- The individual parts of the ventilator are then assembled in the Assembly work cell. This work cell consists of two robots and a conveyor system.
- The ventilator is inspected and packaged in the CheckPack work cell. It consists of a robot, packaging materials and a conveyor belt which links it to the AGVS.

The individual work cells are available as separate cell models, so that each work cell can initially be processed alone. Integration can then be accomplished in the form of a large project.

microFMS

Model MTLR 10

The CNC station is comprised of three functional units (CNC loading robot, CONCEPT MILL 105, CONCEPT TURN 105). The first functional unit is the processing machine loading robot, the second is the CNC milling machine, and the third functional unit is the CNC lathe. The work piece handling is carried out by a Mitsubishi RV-2AJ robot. The robot is mounted on a linear axis, since the work path between the CNC lathe and the CNC milling machine extends beyond the robot's operating range. This linear axis makes it possible for all positions to be reached without difficulty. The unmachined parts that are placed on belts 1 and 2 are processed. After processing, the parts are put on belt 3.
The lathe can turn 4 slots into the work piece. The milling machine mills 4 recesses into the front side of the work piece.
A robot handles the workpiece.

3. Working with CIROS®

After starting CIROS® Robotics, you can directly access the CIROS® Assistant, see figure 2.1.

3.1 CIROS® Help

The CIROS® help function is subdivided into two parts:

- Online help for working with CIROS®
- The CIROS® Robotics Assistant, see chapter 2.3

The menu bar

The menu bar provides access to functions like those of a standard Internet browser. You can scroll forwards and backwards. You can display or hide the navigation bar. You can select a home page, as well as other options for Internet connections. You can print out any selected topics that serve your needs.

Fig. 3.1: The menu

Additional index cards

You also have the option of conveniently navigating within the CIROS® help function using additional index cards including contents, index, search and favourites.

- The Contents index card displays the entire contents of the CIROS® help function in an explorer layout, which can be navigated just like the Microsoft explorer.
- The Index displays all of the keywords used by the entire help function, by means of which information can also be accessed.
- The Search function facilitates full-text retrieval using all of the terms that occur within the entire CIROS® help function.
- You can create your own explorer structure for the CIROS® help function with the Favourites index card.

3.2 The CIROS® Assistant

The CIROS® Assistant provides you with an online learning environment for robotics applications in the field of automation technology. The assistant is subdivided into two parts:

- The CIROS® Robotics Assistant
- Models

The model library is structured as follows:

Introduction models	• Model FirstSteps-NextSteps • Model -Pick-And-Place FESTO • Model -Pick-And-Place ABB • Model Pallet Assembly • Model BP70
MPS robot stations	• MPS-RobotStation • MPS-RobotAssemblyStation • MPS-RobotPunchingStation
iCIM robot stations	• Station Assembly RV-1A • Station Assembly RV-3SB • Station Assembly RH-5AH55 • Station Mill and Turn RV-3SB • Station Mill 55 •Station Turn 55
Special robot applications	• Model LabAutomation • Model Packaging • Model Disassembly • Model DiskTest • Model DiskTest • Model RobWeld
Production systems	• Model PressAutomation-S7 • Model PCBMounting • Model PlantSimulation
microFMS	• Model MTLR 10

The Robotics part provides you with a description of all of the robotic work cells. All applications can be accessed directly via a link .

Figure 3.2: Open model

Open model

There are two options to open a work cell model:

- Open reference model
- Open user model

The reference model is write-protected. Teacher should select this option in order to demonstrate this sample always at any time in its original version.

Select the second option if you want to work with the model. Here you can copy or rebuild all models and robot programs from CIROS Reference to your personal folder. The default setting installs the files under **My Documents\CIROS Robotics Models**. If you wish to save your files to another folder, change the file **ModelHandler.ini** in the CIROS program folder in the subfolder **\bin\Tools**.

[CusttomUserFolder]	
UseCustomUsersDirectory=1	Set this option to 1 in order to uses a custom user folder
CustomUserDirectory=e:\	Path and name of the custom folder
AddUsernameAsSubfolder=1	Extends the path of the custom folder by a subfolder which has the name of the current user e.g. e:\AnnyMiller
AddCustomSubfolder=1	Extends the path for the user models by a variable subfolder in order to allow multiple users under the same operating system account / login to keep their models in individual folders. Creating or changing the user folders can be easily done using the self explaining dialog. Due to security reasons, only letters and digits are allowed for the user folder names [a-Z][0-9]. The name of the currently selected user subfolder is stored in the file "CurrentUserFolder.ini" which is located one level above the user folders. The property used in this ini-file is named "CurrentUserFolder_At_‹Hostname›".When using the same shared network folder for several host, this allows to save a current users folder name for each host.

Default setting: [CustomUserFolder]
UseCustomUsersDirectory=0
CustomUserDirectory=e:\
AddUsernameAsSubfolder=0

If you have finished your work, you may delete your model folder by clicking on the button **Delete user model**.

The opened work cell includes a display of all of the windows that are required for solving the assigned problem. If you chose not to install the solutions during installation, the position list and the programming window are empty, but they are set up such that you can begin work.

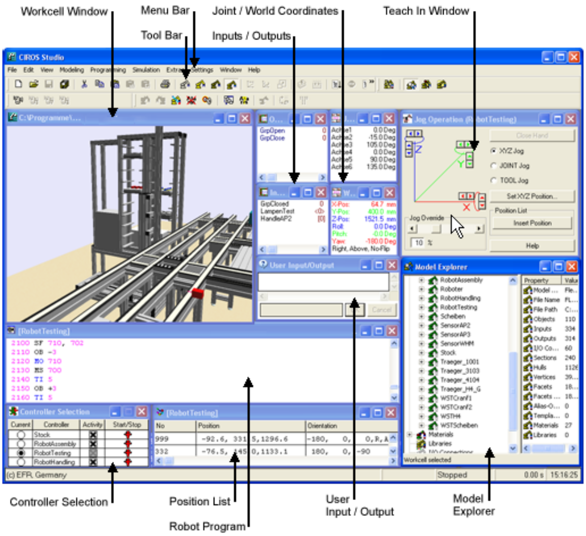
Classroom

The following approach is recommended:

- As a teacher you provide a prepared virtual work cell as a user model, e.g. with modified position list and sample program, in an only read folder.
- The students copy the user model folder in their own work folder. This guarantees that all students can work with the same prepared work cell.

The basic procedures for working with CIROS® are described in the following pages.

3.3
The CIROS® User Interface



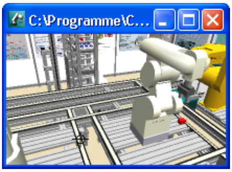
The user interface was new designed:

- The menus **File** and **Edit** include all Windows standard functions you expect in these menus.
- The menu **View** includes all functions supporting you to use the graphic representation of the 3D-simulation.
- The menu **Modeling** includes all functions you need in order to create or modify models.
- The menu **Programming** includes all functions in order to program robots.
- The menu **Simulation** includes all functions to start and stop the simulation, to configure the setting of the simulation and to activate the collision detection.
- The menu **Extras** provides the two functions to use the camera cruise and the master frame concept.

- The menu **Settings** allows you to configure following components:
  - Display of work cell
  - Setting of robot gripper
  - Use of IRDATA interpreter
  - Setting of camera cruise
  - Setting of collision display
  - Type of orientation representation
  - Configuration of programming editor
  - Setting of simulation analyse
  - Setting of the TCP (Tool Centre Point)
- The menu **Window** provides the standard Windows functions and the new submenu **Workspaces**. This menu supports you with the configuration of your work windows. You can save your complete window configuration and can restore it just by one mouse click. Further there are numerous predefined workspaces for you:
  - Work window + program + Position list display
  - Work window + Teachpanel + Position list display
  - Work window + Program + Position list + Teachpanel display
  - Work window + Program + Position list + I/O display
  - Work window + Program + I/O display
  - Work window + Joint- and Worldcoordinates + Position list display
  - Work window + second work window +Teachpanel display
  - Work window + Program + Joint- and Worlcoordinates + I/O display
- The menu **Help** includes the online help to use the software CIROS and the online assistant CIROS Robotics.

3.4 Window Types

The most important window types used in the CIROS® user interface are described below.



Work cell Window

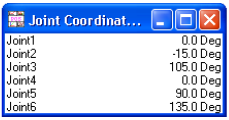
A graphic representation of the currently selected work cell is displayed in the work cell window. Additional views can be opened in the work cell window with the menu function **View → New Window**, allowing you to observe different perspectives simultaneously. The three dimensional representation of the work cell is dependent upon the selected point of view.

- Zoom:
  Mouse wheel or left mouse button and function keys **Ctrl+shift**. The mouse pointer appears in the form of this button, and can then be used to enlarge or reduce the display by moving the mouse.
- Translate:
  Left mouse button and function key **shift**. The mouse pointer appears in the form of this button, and can then be used to move the display by moving the pointer along the coordinate axis.
- Rotation:
  Left or right mouse button and function key **CTRL**. The display can be rotated around the individual coordinate axes.

You can also select various predefined standard views. Use the menu function **View → Standar**d to this end. A dialogue box appears which includes various options:

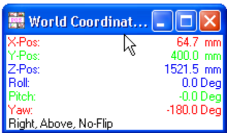
- Default Setting (O)
- Front view (V)
- Rear view (U)
- Top view (A)
- Left-hand side view (L)
- Right-hand side view (R)

The desired view appears after clicking one of the above options, as long as the work cell window is open. This can also be accomplished by simply activating the corresponding keyboard keys.



Joint Coordinates

Press the F7 key or select the menu function View → **Robot position → Show Joint coordinates**. The Joint coordinates window displays the individual positions of each of the robot's joints. Position is specified in degrees for rotary axes, and in millimetres for linear axes. The Set joint coordinates dialogue box can be accessed by double clicking this window.



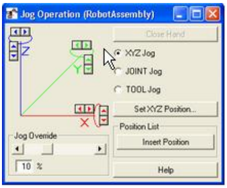
World Coordinates

Activate the shift+F7 key combination or select the menu function **View → Robot position → Show world coordinates.**

In this case the world coordinate system is always equal to the base coordinate system of the robot.

The World coordinates window displays the position and orientation of the TCP (tool centre point) in world coordinates. In addition to position and orientation, the robot's configuration appears in the bottom most line in the window. You may select following different orientation representations by the menu **Settings → Orientation Representation**:

- Roll-Pitch-Yaw angles representation
- Quaternions representation
- Mitsubishi 5-axis coordinates representation



Teach-In

Activate the F8 key or select the menu function **Programming → Teach-in**. In addition to the designations of the robot's joints, the window that now appears includes two small buttons which can be used to advance the robot's individual joints. The performance of a real robot is simulated when these buttons are activated. The robot is accelerated to the preset speed (override) if one of these buttons is pressed and held. The preset speed is then held constant, and braking to a speed of 0 ensues when the button is released, controlled by means of a acceleration ramp. By clicking the corresponding option, teach-in can be performed using world coordinates or tool coordinates.
Further details are included in chapter 4.1.

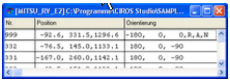
Display Coordinate Systems

Various types of coordinates systems can be displayed for support.
Select the menu **View → Coordinate systems** to this end **(Ctrl+K)**:

- Work cell:
  Show world coordinate system and orientation representation
- Objects:
  Show object coordinate systems
- Gripper:
  Show grip and gripper points
- Robot:
  Show TCP coordinate system, base coordinate system and joint coordinate system

The axis of the coordinate systems are uniformly coloured
(red = x-axis; green = y-axis; blue = z-axis)

You can record the movement of the TCP by the menu function
View → TCP Tracking.



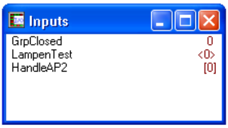
Position list

The screenshot shown on the left contains a position list for a robot. The name of the associated object is specified in the header.

Click the menu function File → Open and select the desired file type, i.e.

- *.pos (for Mitsubishi robot)
- *.psl (for programming in IRL).

Alternatively, create a new position list with the menu function **File → New** and select the desired data type as above.

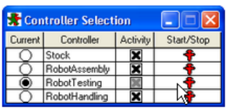


Inputs/Outputs

Press the F9 key, activate the Ctrl+F9 key combination or select the menu function

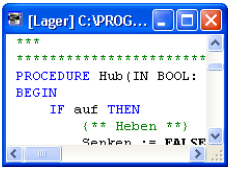
View → Inputs/outputs → Show inputs or Show outputs.

The Inputs window shows which signals are being applied to the inputs of the simulated controller. 0 signals are displayed in red, and 1 signals in green. If the input signal is forced, this is indicated by the fact that the input value appears in angle brackets, e.g. ‹1›. If the input is linked to an output, the input value appears in brackets, e.g. [1]. the same applies to output displays.



Controller Selection

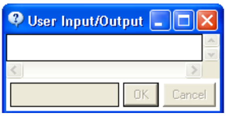
Select the menu function **Programming → Controller selection**. CIROS® Robotics includes work cells with several controllers, for example one PLC and two robot controllers, which work together simultaneously in the simulation mode. However, if a procedure is to be taught into a robot, the teach panel must be first allocated to the desired robot. This task is executed by the controller selection window. It is used to display and select a master, and to activate and/or deactivate individual controllers. The display of robot positions, the display of inputs and outputs, and teach-in are only possible for the robot that has been selected as a master.



Robot Program
Click the menu function **File → Open** and select the desired file type:

- ***.mb4** (for programming in Melfa Basic IV),
- ***.mrl** (for programming in Movemaster Command)
- ***.IRL** (for programming in IRL = Industrial Robot Language)

Or create a new program with the menu function **File → New** and select the desired data type.



User Input/Output
The User Input/Output window appears automatically if the robot program contains commands with which data can be read in or read out, for example via the serial interface at the robot controller. Due to the fact that the robot controller is only replicated in the simulation, data are not transmitted via the serial interface, but rather via the **User Input/Output** window.

3.5 Camera Cruise

The Camera Cruise can record different views of an active work cell window. During simulation these views are recovered in rotation. A new view between two views is determined by linear interpolation. Thus the viewpoint moves uniformly. At the configuration of the Camera Cruise you can schedule times for holding a certain view and for zooming to another viewpoint. As the Camera Cruise is synchronized to the simulation time the viewpoint movement is always synchronized to the simulation of the work cell.
You can also save a Camera Cruise in a video file. At this several compression methods are supported. In the video file (File extension.AVI) all view during the cruise are saved. The video file has the same name and is stored in the same directory as the model file (Extension .MOD) of the actual simulation model.

Switching Camera Cruise on

To switch a camera cruise on, use the menu function **Extras → Camera Cruise → Camera Cruise**. If the camera cruise is switched on, the view follows the configured cruise of the camera during simulation.

Recording Camera Cruise

To record the view of a camera cruise, first switch on the camera cruise. Then use the menu function
Extras → Camera Cruise → Camera Cruise Record. The view will then be recorded to a video-file which will be saved in the model folder under the name ‹model name›.avi.

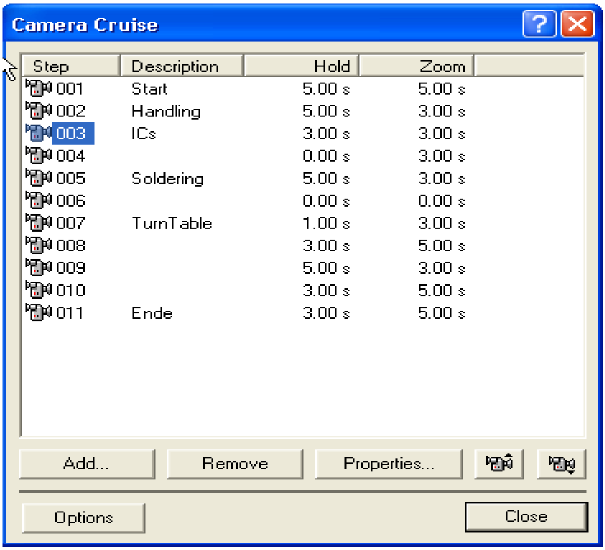
Playback a Camera Cruise Video

To play back a recorded Camera Cruise in CIROS use the menu function **Extras → Camera Cruise → Camera Cruise Play**. This will open the video file in your operating systems default media player.

Stop recording

The menu function **Extras → Camera Cruise → Camera Cruise Stop** stops the recording of a camera cruise.

Configure Camera Cruise
To setup a Camera Cruise for a simulation model use the menu function **Settings → Camera Cruise**. All setting of the camera cruise are saved to the current work cell's .ini file. To apply changes, write access to this file must be granted.
For backup purposes or further use in other work cells, the list of steps can be exported to and imported from a file. To import or export the list use the menu functions **File → Import and File → Export** and select file type CIROS Camera Cruise (file extension .ccc).



Options

View list
This list contains all views of the Camera Cruise. To select a certain view click the number in column step. You can open a context-sensitive menu by clicking the right mouse button.
Double-clicking a step changes the view of the active work cell window to the view of the camera cruise step.

Add
To add the current view to the list click **Add.**

Remove
To remove the selected view from the list click **Remove.**

Properties
To edit the properties of the selected view with dialog box **Camera Cruise - Step X** click **Properties**.

To move up the selected view click this button.

To move down the selected view click this button.

Options
To change the options for recording the Camera Cruise with dialog box options - video.

You can name the selected view, assign a dwell time and a zoom time,

4. Programming

The following programming languages can be used in CIROS® Robotics for programming robots:

- Mitsubishi MELFA Basic IV robot programming language
- Mitsubishi MRL robot programming language
- Standardised industrial robot language (IRL DIN 66312)

We have proceeded as follows in creating sample programs for the models:
All Mitsubishi robots have been programmed using MELFA Basic IV, as long as this language is supported by the respective controller. MRL has been used for all other Mitsubishi robots. Other types of robots have been programmed using IRL. The "Robot Programming" section of the Robotics Assistant includes comprehensive information regarding the programming of robots.

Details regarding the programming languages are contained in the chapter entitled "Programming Languages" in the CIROS® help function.

4.1 Teach-In

In order to create a robot program, certain positions must be defined to which the robot travels under certain conditions. Generally speaking, a robot can be advanced with the help of a manually operated control panel in order to teach such positions.

To start the teaching of a robot it is recommended to use at first the robot without integration in a work cell.

Project Wizard

Open the menu **File → New → Project Wizard**. Following box will be shown:

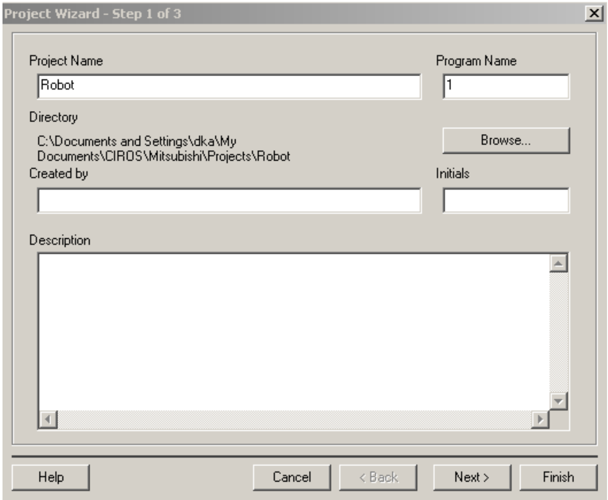


Figure 4.1: Project Wizard – Step1

The project name is used identify the project. It will be the filename after saving, and you must use this name to open the project later. The default suggestion for the project name is "UNTITLED". During installation a directory "Project Name" below the CIROS/CIROS Programming directory is created automatically. According to the selected project name a subdirectory with just that name is created and all files belonging to the project are stored there.

– You may alter the location by using the **browse** button. The default suggestion for the directory is the actual directory.
– Enter a name in the field of **Created by** to identify the author of the project, robot program, etc.
– Click on the button **Next** to do the second step.

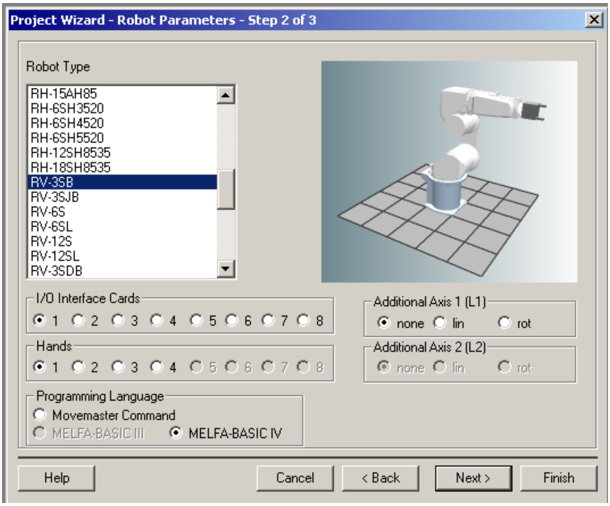


Figure 4.2: Project Wizard – Step

Select one of the robot types, e.g. the 6-axis robot Rv-3SB. Confirm your selection by clicking on the button **Finish**. It will be generated

– a work cell window only with the selected robot type,
– a position list window and a programming window.

Close the added message window and arrange the three remaining windows.

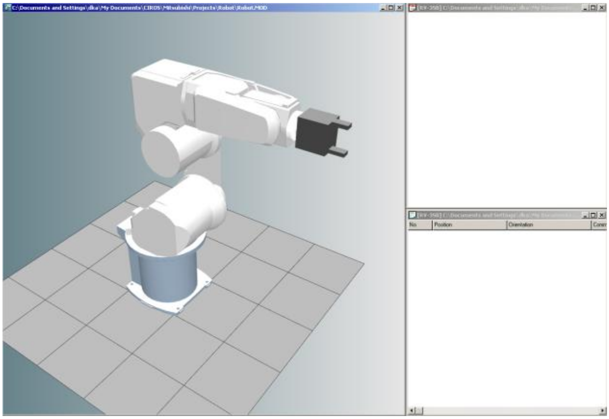


Figure 4.3: Project Wizard – Step3

CIROS® provides users with two different methods for advancing the robot manually:

- With the mouse
- In the teach-In window

Click in close proximity to the gripper end point with the left mouse key. A voxel (pixel in 3D space) is marked at the clicked point. If you double click the voxel, the robot moves to the selected point, if it lies within its workspace. You can represent the workspace by using the menu **View → Show Workspace**.

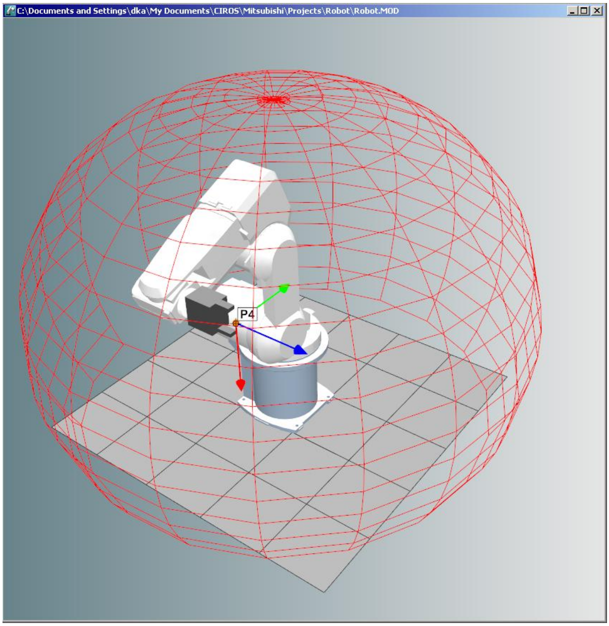


Figure 4.4: Workspace

Use the short key Shift+F2 to add a point the robot moved to in the position list. Selected positions will be shown by name and a cartesian coordinate frame.

Note

This coordinate frame coincides to the tool coordinate system (=TCP coordinate system) of the robot if the robot is placed in this position.

The robot can be advanced in a much more targeted way with the universal teach panel. The teach panel can be accessed via the menu function **Programming → Teach-in** (F8).

Select the "Joint coordinates" mode from the teach-in window

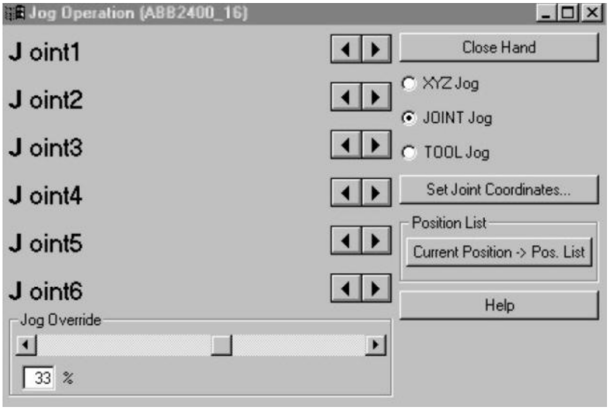


Figure 4.5: Teach panel with joint coordinates

Select one of the robots six joints and click one of the corresponding arrow buttons: The robot moves around the selected joint in the corresponding direction. Speed can be selected with the override slider. After clicking the Set Joint coordinates button, a dialogue box appears to which joint coordinate values can be explicitly entered. The robot's current position can be transferred to the respective position list by clicking the **Current Position → Pos. List** button.

Exercise

Generate for example four positions P1,..,P4 and save them in the position list. By double click on the positions the robot will be placed at these positions. More precisely, the TCP of the robot will be placed at these positions. Youi can easily create a program moving the robot to all defined positions.

– Activate the programming window and open the menu **Programming → Programming-Wizard**. The following dialog box will be opened:

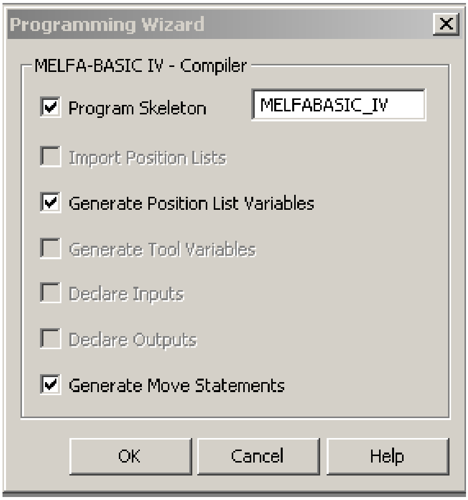


Figure 4.6: Programming Wizard

Confirm the selected options and it will be automatically a program generated.

– Download this program in the virtual robot controller by using the menu command **Programming → Compile+ Link (Ctrl+F9)**.
– Close the message menu showing the information about the download.
– Keep the programming window active and start the program by use of the menu command **Simulation-→ Start** (**F5**).
– Using the function key F10 you can go step by step through the program.

In case of many applications it is much more helpful if you can move the robot in a Cartesian coordinate system. Select the **XYZ Jog** mode in the teach-in window.

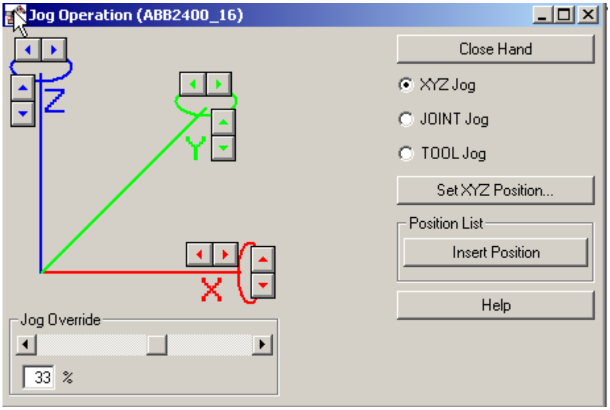


Figure 4.7: Teach panel with world coordinates

The mode XYZ Jog corresponds to the Cartesian base coordinate system of the robot. The robot can be moved along the world coordinate axes, and the gripper can be rotated around these axes by clicking the corresponding arrow buttons. Select the "Tool coordinates" mode in the teach-in window in order to move the robot within the tool coordinate system. The tool coordinate system is the robot's basic coordinate system, but the zero point has been shifted to the robot's TCP.

Corresponding coordinate systems can be displayed at the robot, see chapter 3.3

The world coordinate system in the Teach-in window corresponds to the base coordinate system of assigned robot.

4.2
Example:
Programming a Work cell

This example necessitates the creation of a program for the Mitsubishi RV-2AJ robot that solves the sample task posed in section 2.3 for the First Steps work cell. Open the First Steps RV-2AJ work cell as user model. Select the workspace window configuration **Program + Position list** to get an ideal representation on your screen:

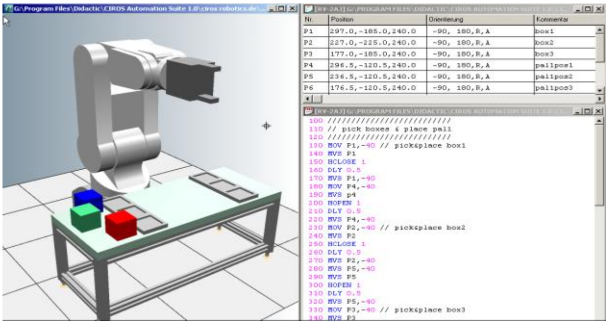


Figure 4.8: FirstSteps RV-2AJ.mod

Example
Programming a Work cell

Sample task
As a reminder, the blue work piece must first be set onto the middle section of the first pallet. After a waiting period of 2 seconds, it must then be sorted into the bottom section of the second pallet.

Creating a position list

We've already created the sequence plan in chapter 2.3. Now we'll need to create a position list. First, delete the contents of the predefined MRL position list and save it under the following new name: "FirstStepsTest.pos".

- Add the robot's initial position as the first entry to the position list. Click the **Current Position → Pos**. List button in the teach-in window to this end.
- The second position (P2) is the gripping position for the blue work piece. A line in the position list is highlighted after clicking underneath the first position. Click the **Current Position → Pos**. List

button in the teach-in window once again. As an exercise, position P2 will be edited manually. Select position entry P2 to this end.

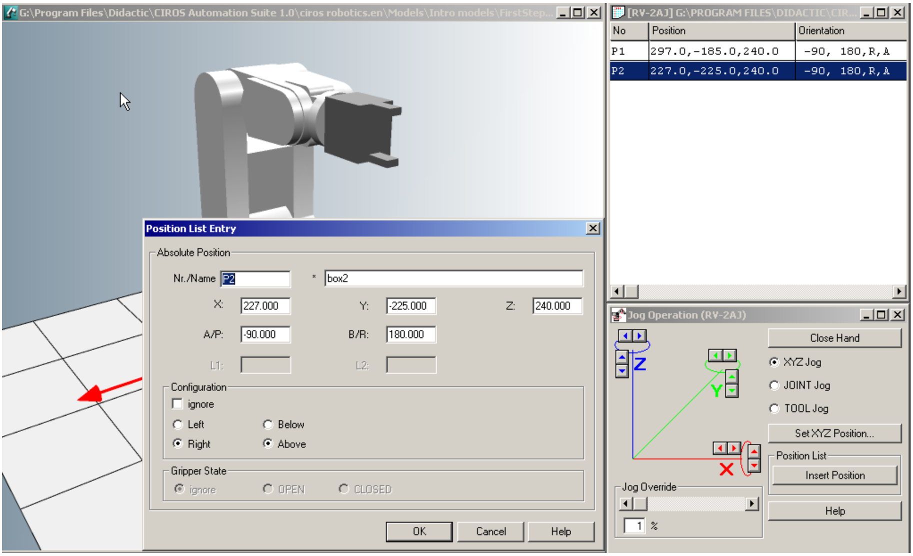


Figure 4.9: Position list entry

- The Position list entry dialogue box can be accessed with the context menu function **right mouse button → Properties** (Alt+Enter). Edit the displayed position data as follows:
  – Positions (X, Y, Z) = (167.00, -185.00, 240.00)
  – Orientation (roll = A/P, pitch= B/R) = (-90.0, 180.0)
- Move the robot to the new P2 position by double clicking the position list entry.
- Close the gripper by clicking the Close gripper button in the teach-in window.
- Use the world coordinate system in order to position the robot such that the blue work piece is set into the middle section of the first pallet.

Tip

The coordinate axes can be displayed for improved orientation: **View → Coordinate systems → Show Tool Centre Point**

Gripper settings

The gripper settings dialogue box can be opened with the menu function **Settings → Grip**.

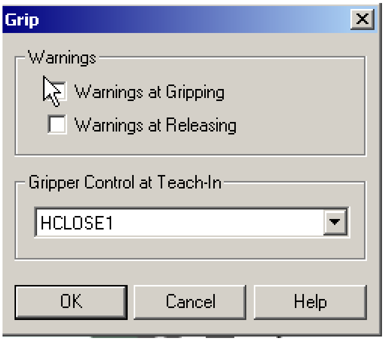


Figure 4.10: Gripper setting

All outputs are included in the Teach-in gripper control drop-down list, which are assigned to objects capable of executing gripping tasks. This output is activated whenever you click the Close gripper button in the teach-in window. You can also choose to have possible warnings displayed for gripper operations.

3-dimensional navigation

It is helpful to open a second work cell window to facilitate 3-dimensional navigation (see also 3.4).

- After moving to the desired position has been successfully completed, add this position to the position list as point P3.
- P4 is the final position in the second pallet.

MasterFrame

By use of the **MasterFrame** you can easily calculate the positions without teaching the robot:

- At first move the robot such that the gripper is parallel aligned to the table, i.e. the orientation (Roll = A/P, Nick= B/R) is equal to (-90.0, 180.0). Temporarily, save this position as P2.
- The MasterFrame is a cartesian coordinate system that you can arbitrarily place in your work cell by use of the menu command **Extras → MasterFrame**. At first switch on the MasterFrame mode.
- By use of the option **Frame → Selection** you can place the MasterFrame such that it corresponds to the coordinate system of a selected object of your work cell.
- For calculation of robot positions it is convenient to place the MasterFrame in the base coordinate system of the robot.
- Open the Model Explorer (Ctrl+T), see 6.2 for further details, and select the object **RV-2AJ**.
- Select the option **Frame → Selection** and the MasterFrame coordinate system will be shown in the origin of the robot.
- Next you have to select the blue work piece for calculation of P2, i.e. you have to select the object Box3 in the Model Explorer.
- Open the command **Properties** of the context menu of this component.
  A new dialog box will be opened. Select the **Pose** page.

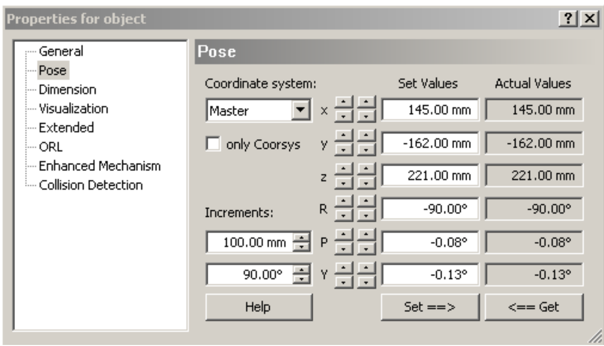


Figure 4.11: Properties for objects

- Select **Master** as coordinate system.
- Since the grip point of the object is not at the edge but in the center the coordinate values must be changed by 25 mm.
- Use the new values for the position P2.
- By double click on P2 the robot will be now exactly place at the position to grip the blue work piece.
- Apply the same method to calculate the positions P3 and P4. Note that the center points of the pallet places respectively have an offset of 60 mm.

Create program

Click into the programming window in order to activate it. Delete its contents and save it as a Melfa Basic IV program under the name of the position list: "FirstStepsTest.mb4".

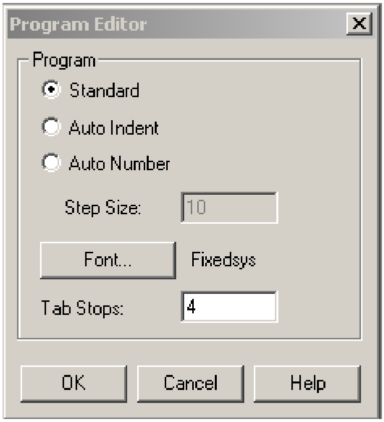
The names of the program and the associated position list must be identical!!!

The MELFA Basic IV programming language is a dialect of Basic, and each program line must thus be numbered. However, numbering has been automated. First create the program lines without any numbering. Now click the button shown in the toolbar screenshot on the left **(Programming → Renumber)**.

It is helpful that the editor markes the compopnents of a program line by different colours

- Numbering of program line: pink
- Command: blue
- Variable name: black
- Parameter: pink
- Comment: black

The menu function Settings → Program Editor enables you to configure the editor



Implement the sequence plan from chapter 2.3 step by step in order to create the program.

Sequence plan	
Open gripper of robot	10 HOPEN
The robot moves the gripper to the gripping position (blue work piece) with a PTP movement	20 MOV P2
Close the gripper	30 HCLOSE 1
The robot moves the gripper to the middle section of the first pallet with a PTP movement	40 MOV P3
Open gripper	50 HOPEN 1
The robot moves linearly back to a point above the first pallet	60 MVS P3,-40
1 Second waiting period	70 DLY 1
The robot moves the gripper back to the middle section of the first pallet (linear movement)	80 MVS P3
Close the gripper	90 HCLOSE 1 100 MOV P4,-40 110 MVS P4 120 HOPEN 1 130 MOV P1 140 END

Add an empty line at the end of the program!

For assistance during programming, execute a right click inside the programming window. You are then provided with a list of the most important function calls, and the corresponding function commands are edited in the programming window via mouse click.
Comprehensive, structured documentation of all Melfa Basic IV programming commands can be accessed in the Robotics Assistant under Programming robots → Robot programs → Basic course. Save your program after it has been completed.

Activate the programming window and download the program into the virtual robot controller (**Strg+F9**). You will be asked if you want to add your program file to an active project or if you want to create a new project:

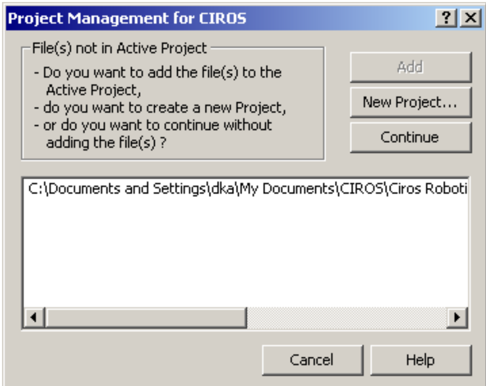


Figure 4.12: Project Management

Select the option **New Project**:

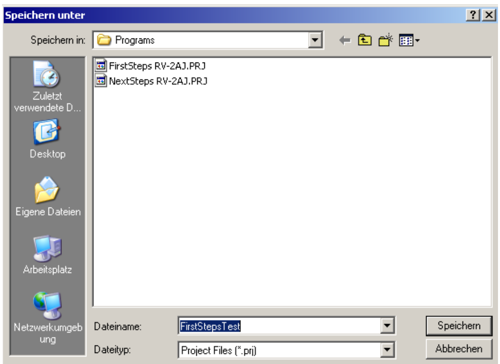


Figure 4.13: Project entry

Enter the project name 'FirstStepsTest' and click on the button **Save**. Compiling will be started and the message window now shows at least four warnings since the position variables are not yet defined. You have to add the position list to the project.

Melfa Basic IV Project

Select the menu function **Programming → Project Management** or click on the button shown in the toolbar screenshot on the left.
The following project management configuration window appears:

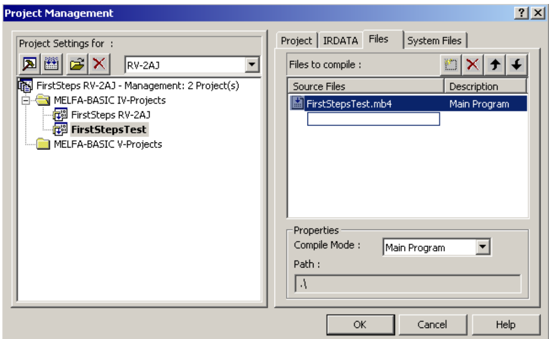
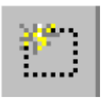
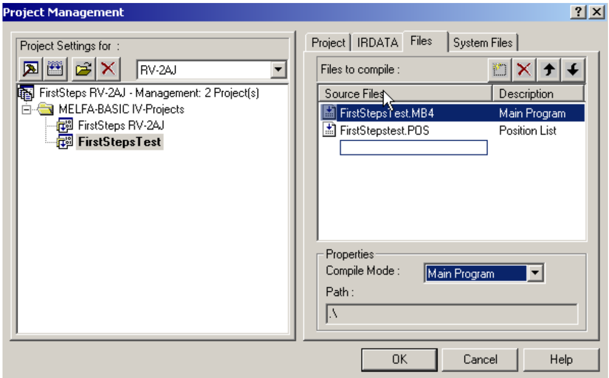
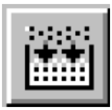


Figure 4.14: Project Management



Select the **Files** register card and click on the empty entry below the program name. After clicking on the left button you will be asked for opening a data file. Add corresponding position list. Note that you have to select the data type 'MELFABASIC IV-Position list (*.POS)'.
In order to create a new project, select the MELFA-BASIC IV projects entry and click Add project in the context menu.





Click on the left button to download your project into the virtual robot controller. Confirm your project entry by clicking on the button OK. Utilised program and system modules, as well as the number of errors and warnings, appear in the **Messages** window.
If error messages appear, the corresponding program line can be highlighted in the program window by double clicking the respective error message.

As a result of cause and effect, it is entirely possible that a different line will be highlighted which appears underneath the actually faulty program line.

4.3 Print Project

Use the menu function **File → Print (Ctrl+P**) to print your project data. You get following dialogue box to configure your print out.

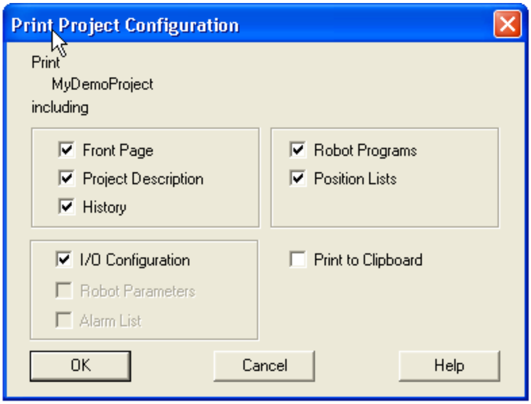


Figure 4.15: Print Project

You can find further details in the chapter **Extensions/Project Wizard/Command File/Print**.

4.4 Download in Mitsubishi Robot Controller

CIROS® Robotics does not provide a communication interface to the Mitsubishi robot controller, but all programs (Movemaster Command or Melfa Basic IV) can be downloaded in the controller via CIROS® Studio.

- Open the RCI-Explorer of CIROS® Studio
- Establish a communication to the controller
- Download programs and corresponding position lists.

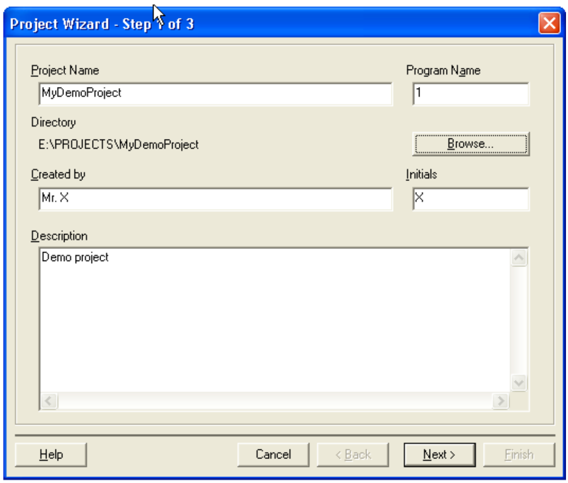
Do not start immediately your program after successful download. At first you should check the following:

- Are all position points correctly adjusted?
- Are all In- and Outputs correctly wired?
- Is the TCP correctly adjusted?

4.5 Project Wizard

The Project Wizard, see also 4.1, assists you during the creation of a new Project. Use the Project Wizard to create a robot in an empty work cell including a programming environment in only three simple steps. To create a new project, just use command **File → Project Wizard**. You will then be guided through the creation process step by step.

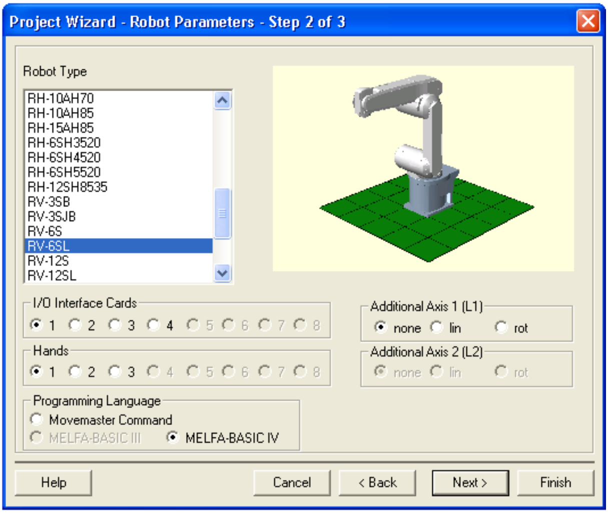
Step 1of 3



Project Name	The project name is used identify the project. It will be the filename after saving, and you must use this name to open the project later. The default suggestion for the project name is "UNTITLED". During installation a directory "Project Name" below the CIROS/CIROS Programming directory is created automatically. According to the selected project name a subdirectory with just that name is created and all files belonging to the project are stored there. Note: The character ‹'› (apostrophe) within the project name is automatically replaced by the character ‹_› (underscore).
Program Name	Enter the desired program name into this edit field. The program name is used as a suggestion when downloading a program into the drive unit. After downloading the program you may use this name to start the program or for a subprogram call.
Directory / Browse...	The Directory field shows the currently selected location to save the actual project. It shows the drive and path, not including the filename. You may alter the location by using the browse button. The default suggestion for the directory is the actual directory.
Created by	Enter a name to identify the author of the project, robot program, etc..
Initials	Enter the author,s initials, e.g. for referencing in the project description or the project history
Description	This field may be used for a description of the project.
General	All data entered within this dialog will be saved, if you change to another step of the project wizard or leave the wizard using Finish.

All data entered within this dialog and during the actual use of the project wizard will be lost if leaving the wizard using Cancel.

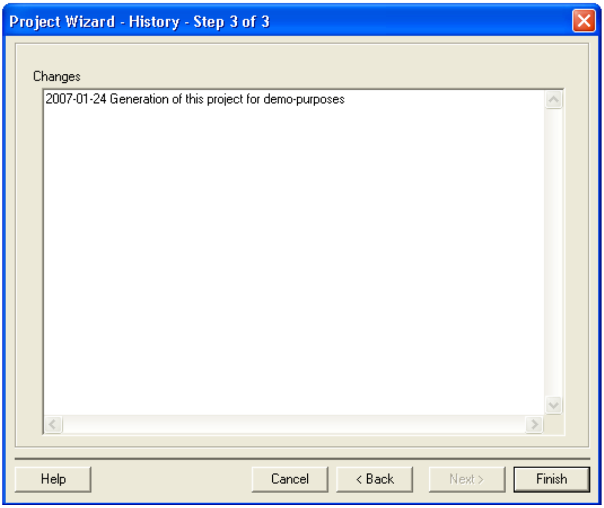
Step 2of 3



Robot Type	Use this list box to select your robot type. The selected robot is shown in the upper right area of the dialog.
I/O-interface cards	Selects the number of interface cards of your drive unit. The maximum number of cards to select depends on the actual robot type: • Movemaster RE-xxx: 3 cards • Movemaster RV-M1/2: 2 cards
Hands	Select the number of hands of the robot here.
Programming Language	The programming language selected here is used for the creation of a program file and selects the der syntax checker. This item is only available if the selected robot type supports more than one programming language. If there is only one possible language for the robot or the controller this language is selected automatically.
Additional Axis 1 (L1)	Use this item to determine the first additional axis. You may specify the type of the axis. The selection lin describes a travel axis, rot a rotating table.
Additional Axis 2 (L2)	Use this item to determine the second additional axis. You may specify the type of the axis. The selection lin describes a travel axis, rot a rotating table. This item is only available if the first additional axis is selected as lin or rot and if the robot type supports 2 additional axis

Step 3of 3

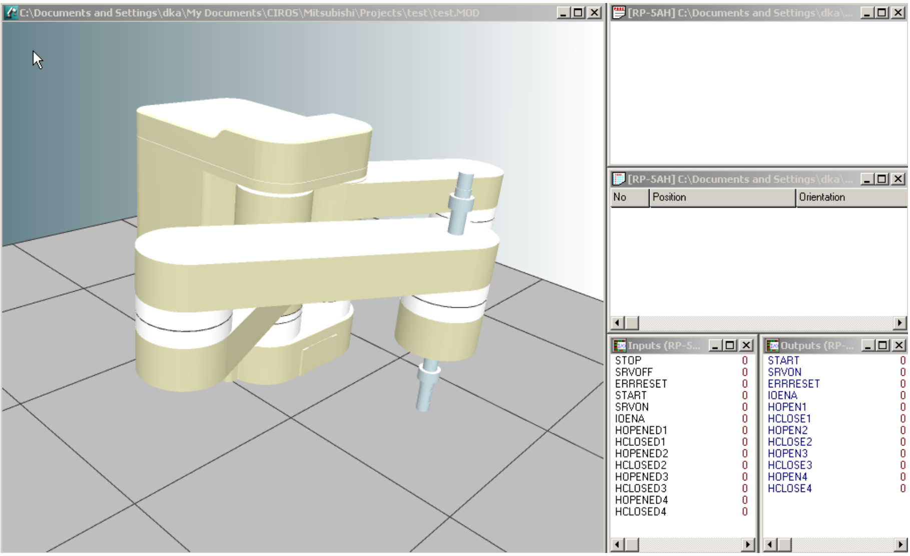
Use this item to determine the second additional axis. You may specify the type of the axis. The selection lin describes a travel axis, rot a rotating table. This item is only available if the first additional axis is selected as lin or rot and if the robot type supports 2 additional axis.



Changes

This field may be used for a description of the changes of the project. You may enter any sentences, words, characters and symbols.

Then click on the button Finish and the work cell with the selected robot, the programming window and position list will be shown. Using the menu function **Window → Workspace → Robot programming → Program, Position List and I/O's** you get an ideal display of your application windows.



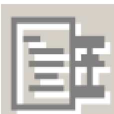
5. Simulation

The simulation of programs that have been written offline using CIROS® is described in the following pages.

5.1
Example: Work cell Simulation

Open the First Steps RV-2AJ work cell with the First Steps Test project from the proceeding chapter. Start the simulation with the menu function **Simulation → Start (F5)**, or click the button shown in the toolbar screenshot on the left.

The program is simulated step by step. Simulation time is displayed in the Status line. The program line that is currently being simulated is highlighted in the program window. At first, you can execute each program step individually with the help of the button shown in the toolbar screenshot on the left.



If you want to start a new simulation cycle, it is advisable to return the robotic work cell to its initial position. Use the menu function **Simulation → Reset Work** cell to this end.

Simulation serves to check your program for two important criteria:

- Is the functional sequence correct?
- Can run time be further optimised?
-

We'll concentrate here on the first question, i.e.

- Is the logical sequence correct?
- Are there any collisions?
-

You should be able to answer the first question on your own.

Collision detection

The collision detection module is completely new developed. As regards collision detection, you should first decide which components are to be examined for possible collisions.

Example

Consider our sample program to this end. The first critical point is certainly the transfer of the blue work piece to the first pallet. The task in this case is to specify that these two objects will be examined for possible collision.

Use the menu function **Settings → Collision** detection to this end. Click the Selection index card

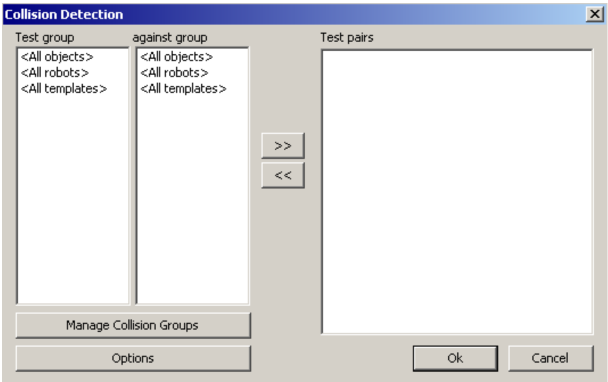


Figure 5.1: Settings of Collision Detection

The first step is to define your own group of components you want to test for collision. Click on the button **Manage Collision Groups**. Following configuration dialog will be displayed:

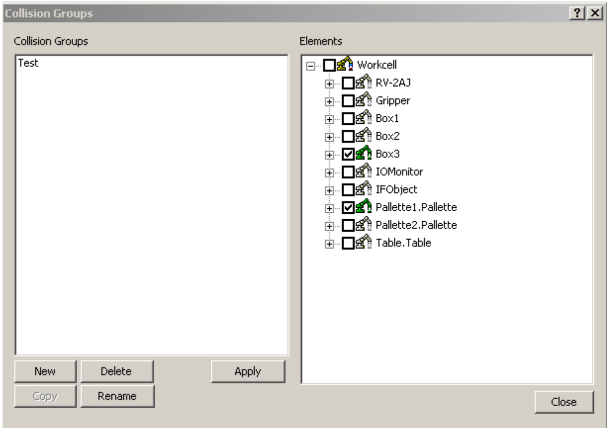
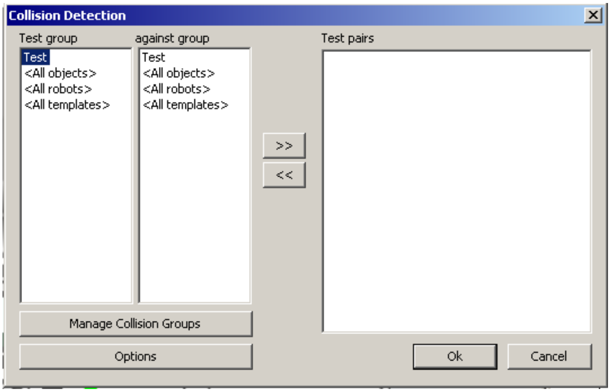


Figure 5.2: Collision Group

Select the function button **New**.

- You are asked for a name of the new collision group, e.g. **Test**.
- The right part of the configuration dialog box displays all elements of the work cell. Select the two elements **Box3** and **Pallette1.Pallette.**
- Confirm your selection by clicking on the button **Apply**.
- **Close** the configuration dialog box.



The new collision group **Test** will be displayed. Select the test pairs for collision. Here, you select **Test** in both cases. By clicking on the double arrow sign the test pairs will be displayed in the right part of above figure.

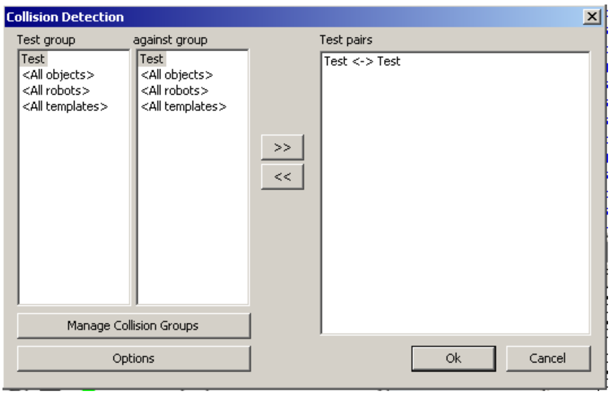


Figure 5.3: Selection of test pairs

Confirm your selection by **Ok**. The button **Options** allows you to change the setting of the visualization of a collision detection.

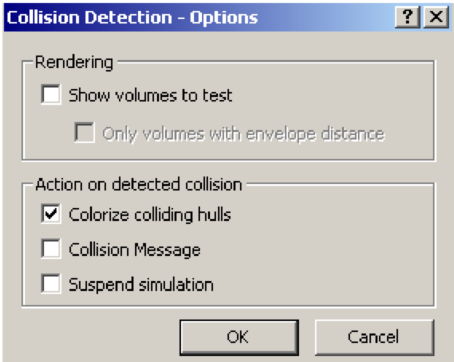
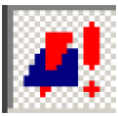


Figure 5.4: Options to represent collision detection



Click the button shown in the toolbar screenshot on the left in order to activate collision detection, or select the Collision detection function in the Execute menu. Start the simulation once again. Notice that the blue work piece turns red during transfer before it is set down onto the first pallet. This indicates that a collision has occurred. This collision persists, because the work piece is set down onto the pallet. How can we eliminate this collision before the work piece is set down?

Recommended solution

Replace line 40 with the following:
40 MVS P2,-30
41 MOV P3,-30
42 MVS P3

For a more detailed visualisation of collision detection use your sample program in the slightly modified NextSteps RV-2AJ.mod work cell, and test for collisions with the glass plate.

5.2
Sensor Simulation

The sensor simulation functions expand the capabilities of CIROS® such that complete robotic work cells can be simulated. Many of the sensors utilised in manufacturing automation can be realistically configured and simulated. Visualisation of sensor measuring ranges, which is not possible in real applications, provides additional help in avoiding design errors during the planning stages. Sensors are utilised in numerous work cells, for example in the MPS® Robot Station, for detecting objects and materials. The characteristics of these sensors can be analysed with the model explorer (see also chapter 6.3).

5.3
PLC simulation

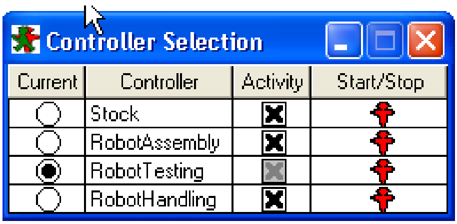
The CIROS® S7 simulator interprets executable S7 programs. Each work cell may include several stored program controllers. Each PLC is controlled by an S7 program. It is not possible to change the S7 program furnished with CIROS® Robotics. An overview of the S7 controllers and the installed S7 programs can be accessed with the S7 Program Manager function in the Execute menu. Presented in a clear-cut tree structure, the S7 program administration window displays the name and elucidates the structure of the PLC programs that have been installed to each of the controllers within the selected work cell. Programs may consist of the following elements:

- Organisational modules
- Function modules
- Data modules
- Functions
- System functions

The contents of each type of element can be displayed by double clicking the respective element.

5.4 Controller Selection

By choosing the menu command **Programming → Controller Selection**, you open the Window for observation and selection of different controller states. Select the controllers to be activated. In this list all objects having a controller of their own are displayed.



Every robot in CIROS is using a controller of its own. Programs can be loaded into these controllers, and they can be started, stopped or executed step by step. To enable these features the controller must be activated.
Display of robot positions, inputs, outputs and teach-in is always done for the emphasized robot (master).
Controllers can be activated, deactivated or emphasized by simple clicking on the referring radio box or check box.
Position lists and programs are assigned to a specific robot, too. By activation of a position list or a program window the assigned robot will be made the emphasized robot (master) automatically.

Master

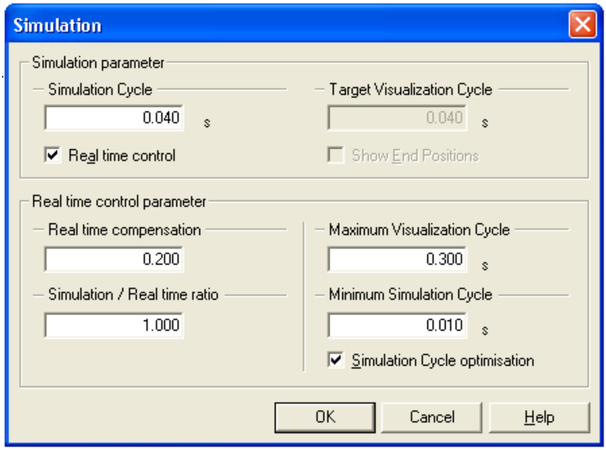
By clicking on the appropriate radio button you assign the master state to the referring controller. In case the chosen controller has been inactive before, it is activated at the same time. The following window contents or parameters are adjusted to the respective master controller:

- Position window (joint or world coordinates)
- Position list window
- I/O window
- Teach-In window
- Loading of programs (IRL, S5 SPS, etc.)

Controller	Shows the name of the controller.
Activity	By clicking into the appropriate check box you activate or inactivate the referring controller. The master controller is always active, it cannot be disabled. Activation or deactivation of controllers has got the following consequences: • Start/Stop of programs is only performed, if the controller is activated. • State notifications are processed only, if the controller is activated.
Start/Stop	This field shows the if the controller is running or not. All controllers are started or stopped consecutively after choosing 'Start, or 'Stop, from the execute menu.

5.5 Simulation Settings

Use the menu function Settings → Simulation for configuration.



Simulation cycle

The simulation cycle specifies the intervals, in which the simulation controller interpolates the states of robots. Additionally it specifies the cycle time for available PLCs and the recalculation time for all extension modules (e.g. sensor simulation, transport simulation,...).
A high value results in a fast simulation, but with only very few interpolation steps. Too high values may result in important steps not being calculated. A low value calculates more interpolation steps, but therefore decreases the simulation speed due to the need of more machine time.

Example

Granted that a robot needs exactly one second for a certain motion command. Depending on the simulation cycle, the number of interpolations would be as following:

Simulation cycle	0.040	0.100	0.200	0.500	1.000
Number of interpolations	25	10	5	2	1

Target Visualization Cycle

The visualization cycle specifies the intervals, in which the model in the work cell window shall be refreshed. The value can be interpreted as "refresh work cell window each visualization cycle seconds". A very low value means that the window will be refreshed very often, which may due to a higher need of machine power, result in slowing down the simulation.
Since the simulation is recalculated each simulation cycle, the value of the visualization cycle must always be equal to of greater than the simulation cycle.

Show End Positions

This option ensures that the state at the end position of a robot motion visualized, even if it lies between two visualization.

Real Time Control

Select this option to enable the real time visualization. The visualization cycle will then be adjusted dynamically to provide real time views of a running simulation.

Real Time Compensation

This parameter determines a constant (amplification P) to control the Visualization Cycle. Values range from 0.1 to 0.6. A small value means a slower compensation, higher values may force fluctuations or even oscillations.

Maximum Visualization Cycle	Selecting Realtime makes the system set the Visualization Cycle automatically to reach a synchronization between the simulation time and the real clock. In case that a model is very complex, it can happen that real time control is not possible due to too high machine power requirements. This would result in permanently increasing the visualization cycle. To avoid this effect, the maximum visualization cycle can be limited to a certain value. The range of the visualization cycle is always between the simulation cycle and the simulation cycle.
Simulation / Real Time Ratio	The parameter entered determines the relation between simulation time and real time. The default value 1.0 controls the simulation time according to realtime, a value greater 1.0 makes the simulation time run faster then realtime. Selecting the value 5.0 makes the simulation time run five times faster than realtime. A simulation period of 50 seconds will take 10 seconds in realtime.
Simulation Cycle Optimization	Select this option to use spare computing power of your machine in order to improve the simulation cycle. The simulation cycle will then be optimized dynamically, depending on the unused computing power. The lower limit for the simulation cycle can be defined in the field Minimum simulation cycle
Minimal simulation cycle	This field defines a lower limit for the option Simulation Cycle Optimization.

6. Modelling

Although new work cells cannot be saved to CIROS® Robotics, you are provided with numerous modelling functions within the work cells that allow you to change layouts, and to analyse alternatively configured problems.

No changes of the work cell can be saved!

Various tools are made available by CIROS® for modelling robot controlled work cells, for example model libraries and the model explorer. We'll help you get acquainted with the modelling function for changing the layout of a work cell using the example provided in the chapter on programming. CIROS® Automation Suite provides full functionality of modelling which enables you to create new work cells for the Robotics version.

6.1 Model hierarchy

The following types of elements are included in the CIROS® model hierarchy:

	**Objects**	Objects are at the top of the model hierarchy. Example: A robot is an object.
	**Groups**	Groups are assigned to objects. Each group may enjoy a given degree of freedom, and can thus be moved relative to the previous group Example: A robot jont is a group
	**Components**	Components are assigned to groups and determine the graphic representation. Example: Surfaces, cuboids and polyhedrons are components.

	**Gripper Points**	A gripper point is assigned to a group included in the gripping object, so that one object can grip another. Example: A gripper point is located on the flange of a robot's sixth axis.
	**Grip Points**	A gripping point is assigned to a group included in the object to be gripped, so that one object can be gripped by another. Example: A work piece that is gripped has a grip point.

6.2 The Model Explorer

All of the elements included in a work cell can be accessed via the model explorer. In addition to objects and their subordinate elements, this also applies to materials, libraries, illumination settings and all I/O connections.

The model explorer is opened by clicking the menu function **Modelling → Model Explorer (Ctrl+T)**.

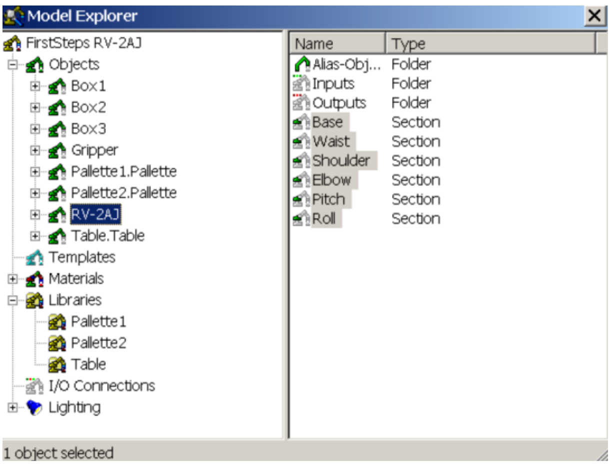


Figure 6.1: Model Explorer

The model explorer window is subdivided into two sections:
A tree structure used for navigation appears in the left-hand section including folders for the individual work cell elements. The element list included in the right-hand side of the window displays the elements included in the folder that has been selected in the tree structure. Elements can be accessed by clicking the desired element in the tree structure (if it appears there), or in the element list.

An element selection context menu containing the most important commands can be displayed by double clicking an element or an element folder.

The Objects folder contains all of the work cells components. We make reference to this folder name in the work cells function descriptions.

Example

You want to determine the exact position of the green work piece in the **First Steps** work cell expressed in world coordinates.

1. Activate the editing mode using the Edit mode function in the Modeling menu (Ctrl+E), and open the model explorer. Click the green work piece. The object is then selected and the associated object coordinate system is displayed. The "Objects" file is selected in the tree structure, as well as the appropriate component, i.e. "Box2", in the display window. The desired allocation has now been established.
2. Click the "Box2" object in the tree structure and select Properties from the context menu. The Object properties dialog box appears, from which the Position index card must now be selected.

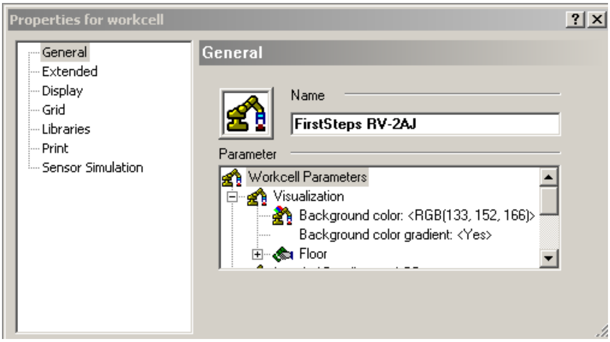


Figure 6.2: Object Properties

The Cartesian coordinates of the zero point from the object coordinate system are displayed here, as well as the orientation of the object relative to the world coordinate system (roll: rotation around the Z-axis, pitch: rotation around the Y-axis, yaw: rotation around the X-axis).

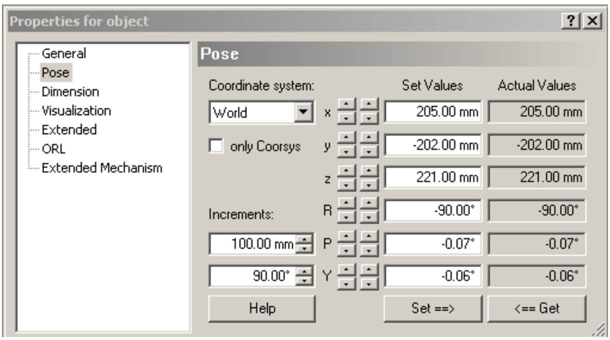


Figure 6.3: Pose Property

Changing object properties

We want to expand our sample task by requiring that the cell is changed such that the green work piece is approximately at the centre of the table, turned 45° relative to the world coordinate system:

Solution

1. The display of Cartesian coordinates and orientation values in the Object properties dialogue box can be directly overwritten, or you can change the displayed values using the arrow buttons in steps according to the selected increment. The work piece is immediately moved to its new *p*osition if the new set value is confirmed by clicking the **Set** → button.
2. Change the Y coordinate and the roll angle accordingly.

The Object properties dialogue box includes additional parameters that are contained in the **General, Dimension, Visualisation, Extended, ORL, Extended Mechanism** index cards etc. These additional object properties can be only modified in CIROS® Studio of the Automation Suite.

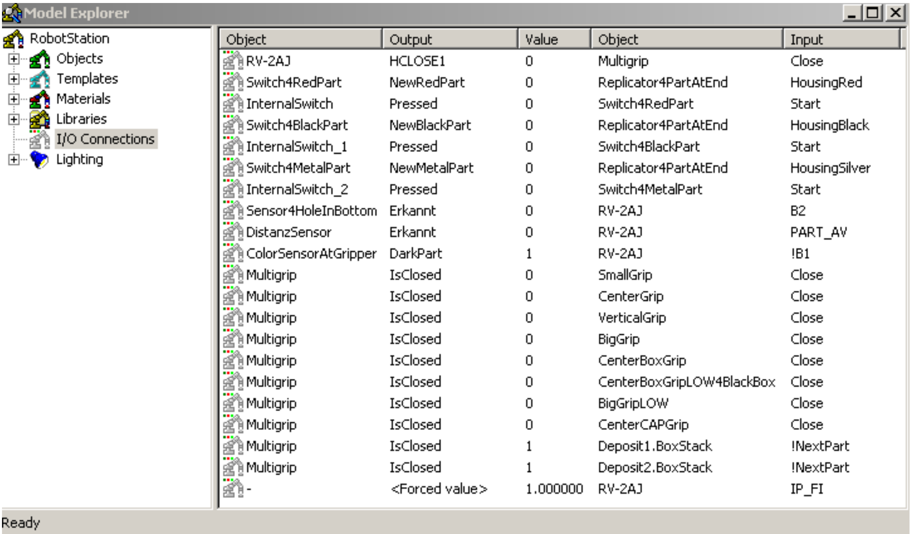
Library elements

Objects in a work cell can also be grouped together as library elements. The goal is to assure that the included objects are always arranged in a fixed geometric constellation in relation to one another. Library elements are recognised by means of their designation. Library elements always have two-part names: **Library_name.Object_name**
Example: (First Steps work cell): Pallet1.Pallet

Only the properties of the corresponding library element can be changed. For example, if you want to change the position of the first pallet, you must click the "Pallet1" object in the Library folder included in the tree structure, and open the Properties dialogue box from the context menu.

I/O connections

Select the folder I/O connections in the model explorer. Then you get a list of all I/O connections where to each in- and output the corresponding object is pointed out.

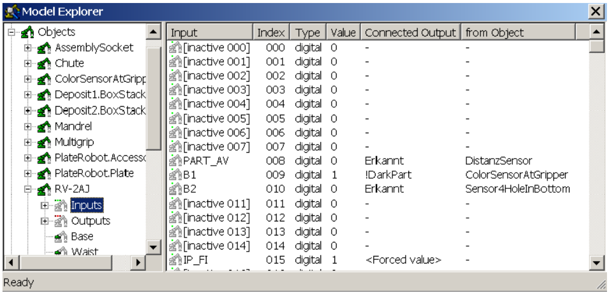


Object	Output	Value	Object	Input
RV-2AJ	HCLOSE1	0	Multigrip	Close
Switch4RedPart	NewRedPart	0	Replicator4PartAtEnd	HousingRed
InternalSwitch	Pressed	0	Switch4RedPart	Start
Switch4BlackPart	NewBlackPart	0	Replicator4PartAtEnd	HousingBlack
InternalSwitch_1	Pressed	0	Switch4BlackPart	Start
Switch4MetalPart	NewMetalPart	0	Replicator4PartAtEnd	HousingSilver
InternalSwitch_2	Pressed	0	Switch4MetalPart	Start
Sensor4HoleInBottom	Erkannt	0	RV-2AJ	B2
DistanzSensor	Erkannt	0	RV-2AJ	PART_AV
ColorSensorAtGripper	DarkPart	1	RV-2AJ	!B1
Multigrip	IsClosed	0	SmallGrip	Close
Multigrip	IsClosed	0	CenterGrip	Close
Multigrip	IsClosed	0	VerticalGrip	Close
Multigrip	IsClosed	0	BigGrip	Close
Multigrip	IsClosed	0	CenterBoxGrip	Close
Multigrip	IsClosed	0	CenterBoxGripLOW4BlackBox	Close
Multigrip	IsClosed	0	BigGripLOW	Close
Multigrip	IsClosed	0	CenterCAPGrip	Close
Multigrip	IsClosed	1	Deposit1.BoxStack	!NextPart
Multigrip	IsClosed	1	Deposit2.BoxStack	!NextPart
-	<Forced value>	1.000000	RV-2AJ	IP_FI

Figure 6.4: List of I/O connections

Sample task: You would also like to know which input bit is allocated to the symbolic "Part_AV" input at the robot controller in the "MPS® RobotStation.mod" work cell.

Solution: Open the folder for the RV-2AJ object and select the inputs subfolder. All input bits are then displayed in the right-hand window.



You can now see that the input in question is allocated to input bit no. 8.

6.3 Example: Modelling in a Work Cell

In chapter 6.2 you learned how to change object properties, enabling you to easily modify the work cell layout.
There are numerous other possibilities of remodelling your work cell in a sensible fashion in CIROS® Robotics.

Illumination

You wish to change the illumination in the graphic representation. Illumination consists of ambient light and up to 7 additional light sources. Open the Illumination folder and select the "Ambient light" object. Open the properties dialogue box from the context menu. The intensity and colour of the light can be changed.

You wish to find out which light sources are turned on in the sample work cell, and what effect they have on the work cell. Select, for example, light source 1 with a left click. The orientation of the light source is graphically represented in the work cell window by means of a light beam, and the associated object properties window is opened. Light sources can be turned on and off, and their orientation, intensity and colour can be changed.

Show sensor signals

You want to visualize the sensor measuring range in the work cell MPSRobotStation.
Open the Model Explorer and open the Properties for Work cell window.
Select the menu Sensor Simulation and click on the option Show measuring range

