Training

KUKA Robotics USA

KORE: Basic Course

KUKA Official Robot Education

Target Group: School and College Students

Issued: 19.09.2014

Version: KORE: Basic Course – V1.1

Contents

3 Overview of the components of a robot system

3.1 Overview

The following contents are explained in this training module:

- Components of a robotic cell
- Selection criteria for a robot
- Control of robot and external axes
- Tool selection
- Selection of the energy supply system
- Periphery connection
- Use of sensors
- Safety equipment

3.2 Components of a robotic cell

A robot system / robotic cell consists of the following components:

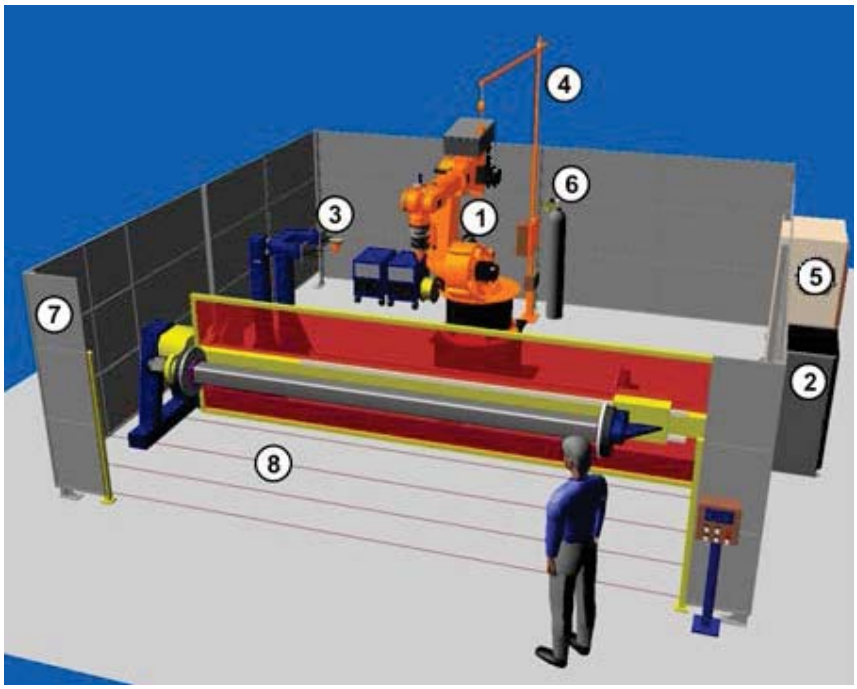


Fig. 3-1: Arc welding cell

Item	Description
1	Robot
2	Controller
3	Tool / tool changer
4	Energy supply system
5	Periphery connection
6	Sensor system
7	Safety fence
8	Loading area with photoelectric curtain

4 Industrial robots

4.1 Overview

The following contents are explained in this training module:

- What is a robot?
- Structure of a robot
- Arrangement of the main axes
- Absolute accuracy and repeatability

4.2 Introduction to robotics

What is a robot?

The term *robot* comes from the Slavic word *robota*, meaning *hard work*.

According to the official definition of an industrial robot: “A robot is a freely programmable, program-controlled handling device”.

The robot thus also includes the controller and the operator control device, together with the connecting cables and software.

Fig. 4-1: Industrial robot

1 Controller ((V)KR C4 control cabinet)
2 Manipulator (robot arm)
3 Teach pendant (KUKA smartPAD)

Everything outside the system limits of the industrial robot is referred to as the *periphery*:

- Tooling (end effector/tool)
- Safety equipment
- Conveyor belts
- Sensors
- Machines
- Etc.

5 Robot controller

5.1 Overview

The following contents are explained in this training module:

- Description of the robot system
- Overview of KR C4 compact
- Technical data
- Interfaces

5.2 Description of a robot system

The industrial robot consists of the following components:

- Manipulator
- Robot controller
- smartPAD teach pendant
- Connecting cables
- Software
- Options, accessories

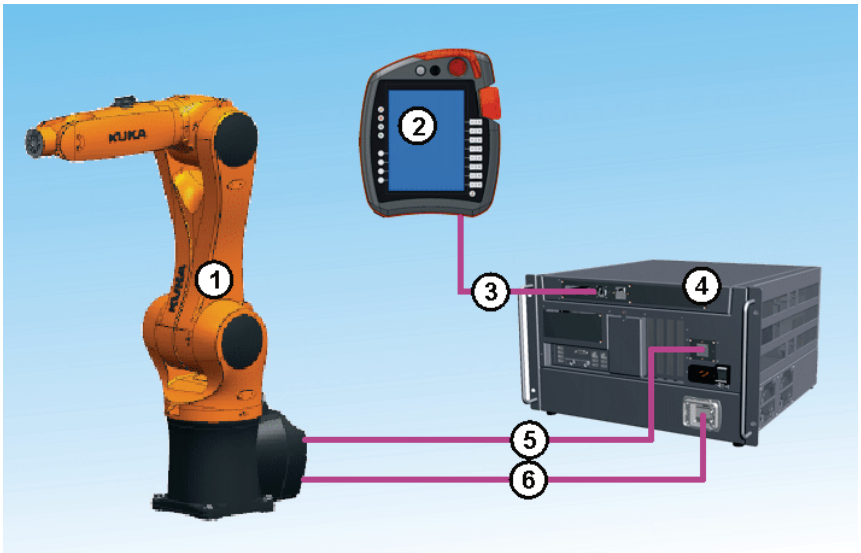


Fig. 5-1: Example of an industrial robot

1 Manipulator
2 Teach pendant
3 Connecting cable, smartPAD
4 Robot controller
5 Connecting cable, data cable
6 Connecting cable, motor cable

⚠ CAUTION For safe operator control of the robot system illustrated here, additional safety measures are necessary, e.g.:

- a safety fence
- external Emergency Stop
- possibly an external safety controller

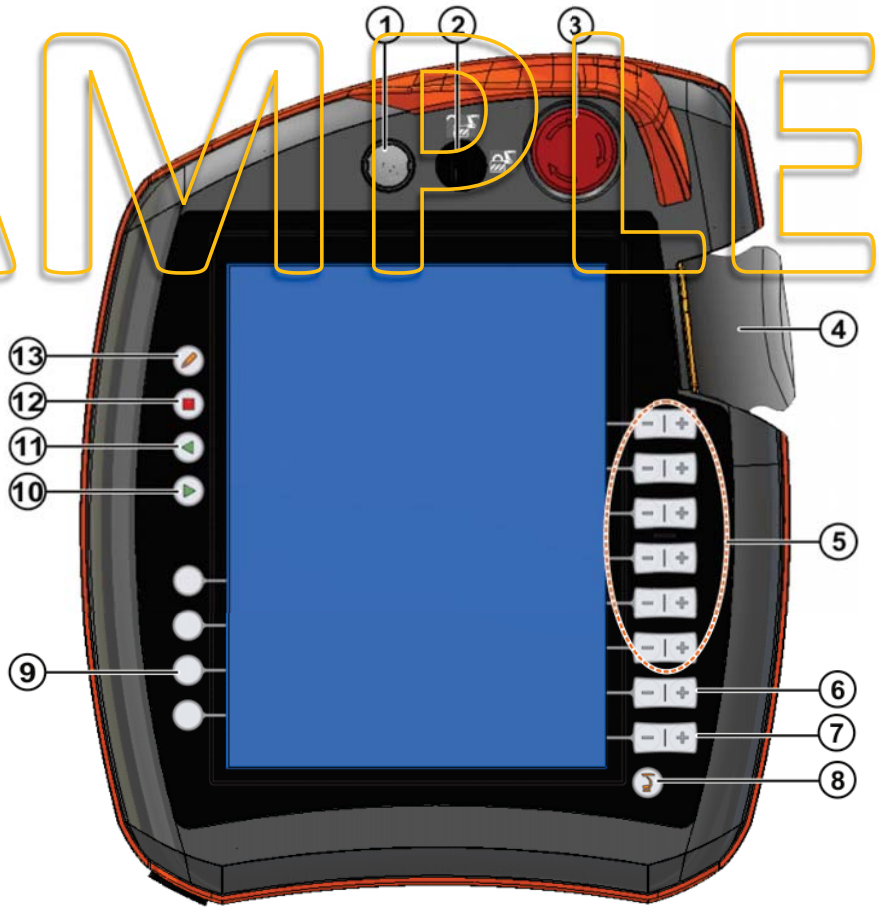


Fig. 6-1: KUKA smartPAD, front view

Item	Description
1	Button for disconnecting the smartPAD
2	Keyswitch for calling the connection manager. The switch can only be turned if the key is inserted. The operating mode can be changed by using the connection manager.
3	EMERGENCY STOP button. Stops the robot in hazardous situations. The EMERGENCY STOP button locks itself in place when it is pressed.
4	Space Mouse: For moving the robot manually.
5	Jog keys: For moving the robot manually.
6	Key for setting the program override
7	Key for setting the jog override
8	Main menu key: Shows the menu items on the smartHMI

Item	Description
9	Status keys. The status keys are used primarily for setting parameters in technology packages. Their exact function depends on the technology packages installed.
10	Start key: The Start key is used to start a program.
11	Start backwards key: The Start backwards key is used to start a program backwards. The program is executed step by step.
12	STOP key: The STOP key is used to stop a program that is running.
13	Keyboard key Displays the keyboard. It is generally not necessary to press this key to display the keyboard, as the smartHMI detects when keyboard input is required and displays the keyboard automatically.

6.2.2 Rear view

Overview

Fig. 6-2: KUKA smartPAD, rear view

1	Enabling switch	4	USB connection
2	Start key (green)	5	Enabling switch
3	Enabling switch	6	Identification plate

6.8 Moving the robot in the tool coordinate system

Jogging in the tool coordinate system

Fig. 6-12: Robot tool coordinate system

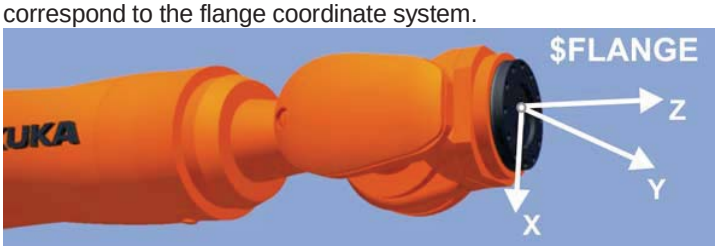
- In the case of jogging in the tool coordinate system, the robot can be moved relative to the coordinate axes of a previously calibrated tool.

  The coordinate system is thus not fixed (cf. world/base coordinate system), but guided by the robot.

  In this case, **all** required robot axes move. Which axes these are is determined by the system and depends on the motion.

  The origin of the tool coordinate system is called the **TCP** and corresponds to the working point of the tool.
- The jog keys or Space Mouse of the KUKA smartPAD are used for this.
- There are 16 tool coordinate systems to choose from.
- The velocity can be modified (jog override: HOV).
- Jogging is only possible in T1 mode.
- The enabling switch must be pressed.

In the case of jogging, uncalibrated tool coordinate systems always correspond to the flange coordinate system.



Principle of jogging – tool

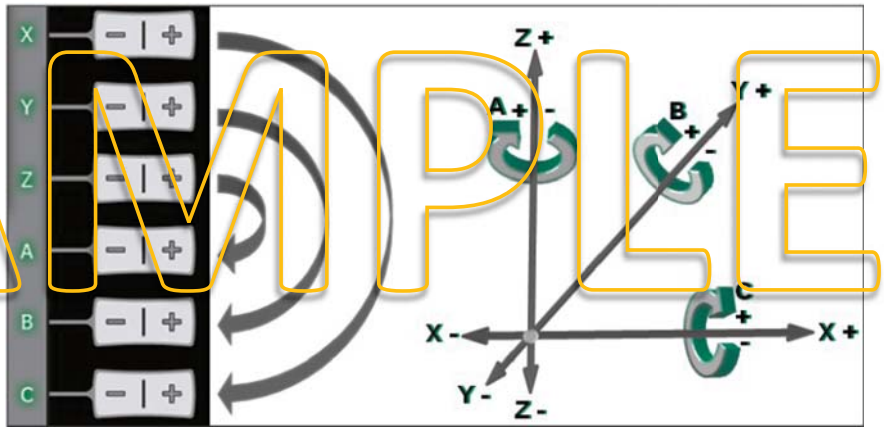


Fig. 6-13: Cartesian coordinate system

A robot can be moved in a coordinate system in two different ways:

- Translational (in a straight line) along the orientation directions of the coordinate system: X, Y, Z
- Rotational (turning/pivoting) about the orientation directions of the coordinate system: angles A, B and C

Advantages of using the tool coordinate system:

- The motion of the robot is always predictable as soon as the tool coordinate system is known.
- It is possible to move in the tool direction or to orient about the TCP.

  The *tool direction* is the working or process direction of the tool: the direction in which adhesive is dispensed from an adhesive nozzle, the direction of gripping when gripping a workpiece, etc.

Procedure

1. Select **Tool** as the coordinate system to be used.

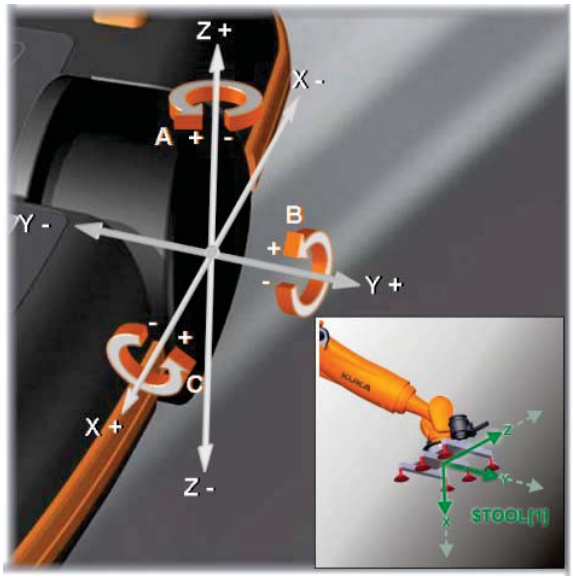
2. Select the tool number.

3. Set jog override.

4. Press the enabling switch into the center position and hold it down.

5. Move the robot using the jog keys.

6. Alternatively: Move in the corresponding direction using the Space Mouse.



6.8.1 Exercise: Operator control and jogging in the tool coordinate system

Aim of the exercise

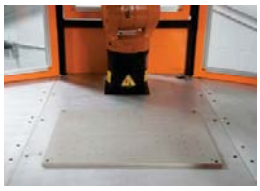
On successful completion of this exercise, you will be able to carry out the following activities:

- Jog the robot, in the tool coordinate system, by means of the jog keys and Space Mouse
- Jog the robot in the working direction of the tool

Preconditions

The following are preconditions for successful completion of this exercise:

- Completion of safety instruction
- Theoretical knowledge of jogging in the tool coordinate system
- Marker holder mounted on grid plate in holes A1 / A2
- Pointer tool mounted on the grid plate in a location that will be easy to reach from multiple different robot orientations.



Task description

Carry out the following tasks:

1. Switch the control cabinet on and wait for the system to boot
2. Release and acknowledge the Emergency Stop.
3. Ensure that T1 mode is set.
4. Activate the tool coordinate system.
5. Select **"Demo_Gripper_1"** as your tool.
6. Jog the robot in the tool coordinate system with various different jog override (HOV) settings using the jog keys and space mouse. Test motion in the working direction of the tool and re-orientation about the TCP.
7. Fetch the pen from the holder using the tool **"Demo_Gripper_1"**.
8. Return the pen to the holder using the tool **"Demo_Gripper_1"**.

What you should now know:

1. How many tools exist in the robot?

. . . . . .

. . . . . .

2. What steps are required for jogging relative to the desired tool coordinate system?

. . . . . .

. . . . . .

3. Where is the location of an un-calibrated tool?

. . . . . .

. . . . . .

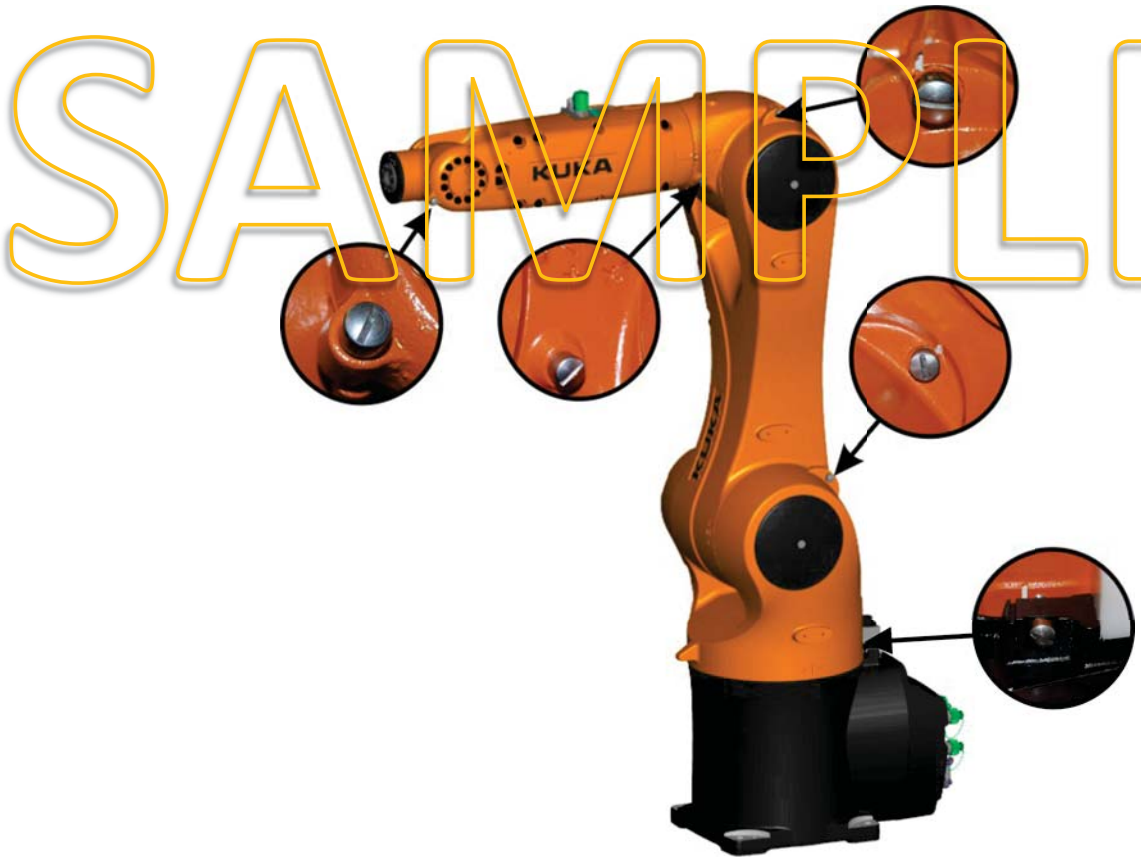


Fig. 7-1: Mastering position for KR AGILUS

Angle values of the mechanical zero position (= reference values)

Axis	KR AGILUS
A1	0°
A2	-90°
A3	+90°
A4	0°
A5	0°
A6	0°

When is mastering carried out?

A robot must always be mastered. Mastering must be carried out in the following cases:

- During commissioning
- Following maintenance work to components that are involved in the acquisition of position values (e.g. motor with resolver or RDC (Resolver digital converter))
- If robot axes are moved without the controller (e.g. by means of a release device)
- Following mechanical repairs/problems, the robot must first be unmastered before mastering can be carried out:
  - After exchanging a gear unit
  - After an impact with an end stop at more than 250 mm/s
  - After a collision

> Before carrying out maintenance work, it is generally a good idea to check the current mastering.

Safety instructions for mastering

The functionality of the robot is severely restricted if robot axes are not mastered:

- Program mode is not possible: programmed points cannot be executed.
- No Cartesian jogging: motions in the coordinate systems are not possible.
- Software limit switches are deactivated.

NOTICE The software limit switches of an unmastered robot are deactivated. The robot can hit the end stop buffers, thus damaging the robot and making it necessary to exchange the buffers. An unmastered robot should not be jogged, if at all avoidable. If it must be jogged, the jog override must be reduced as far as possible.

Carrying out mastering

Fig. 7-2: MEMD screwed in

Mastering is carried out by determining the mechanical zero point of the axis. Every axis is thus equipped with a mastering cartridge and a mastering mark.

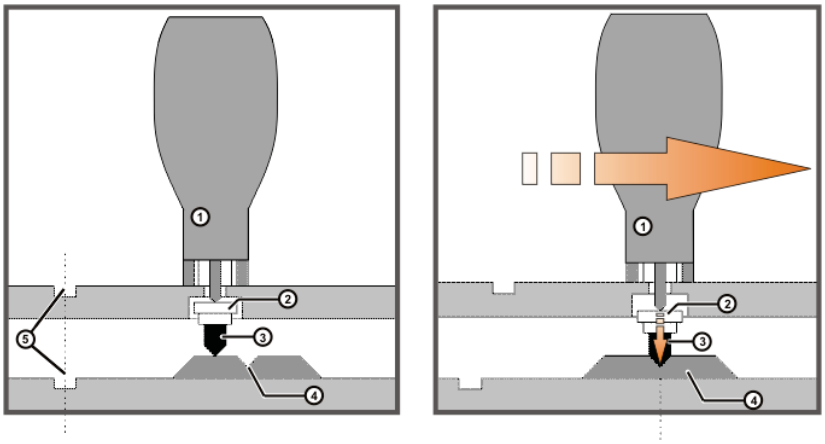


Fig. 7-3: EMD mastering sequence

1 MEMD (Micro Electronic Mastering Device)
2 Gauge cartridge
3 Gauge pin
4 Reference notch
5 Premastering mark

7.7.1 Exercise: Tool calibration

Aim of the exercise

On successful completion of this exercise, you will be able to carry out the following activities:

- Calibration of a tool origin using the XYZ 4-point and XYZ reference methods
- Calibration of a tool orientation using the ABC World and ABC 2-point methods
- Calibration of a tool using the numeric input method
- Activation of a calibrated tool
- Moving the robot in the tool coordinate system
- Moving the robot in the tool direction
- Reorientation of the tool about the Tool Center Point (TCP)

Preconditions

The following are preconditions for successful completion of this exercise:

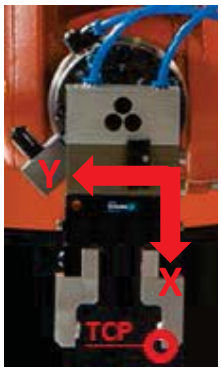
- Theoretical knowledge of the various TCP calibration methods
- Theoretical knowledge of the various tool orientation calibration methods
- Theoretical knowledge of robot load data
- Marker holder mounted on grid plate in holes A1 / A2
- Ring tool holder mounted on the grid plate in hole A8
- Pointer tool mounted on the grid plate in a location that will be easy to reach from multiple different robot orientations.




Task description

Carry out the following tasks:

1. Use the name **"My_Gripper"** and tool #3 for tool calibration of the gripper.
2. Calibrate the TCP of the gripper using the XYZ 4-point method as illustrated.
3. The tolerance should not exceed 0.95 mm. In practice, this value is not sufficient. It is better to achieve tolerances of 0.5 mm or even 0.3 mm.
4. Calibrate the orientation of the gripper coordinate system using the ABC 2-point method.
5. Save the TOOL data and test jogging with the gripper in the tool coordinate system.



Training gripper – TCP

10 Creating and modifying programmed motions

10.1 Overview

The following contents are explained in this training module:

- Creating cycle-time optimized motions
- Creating CP motions
- Modifying motion commands

10.2 Creating new motion commands

Programming robot motions

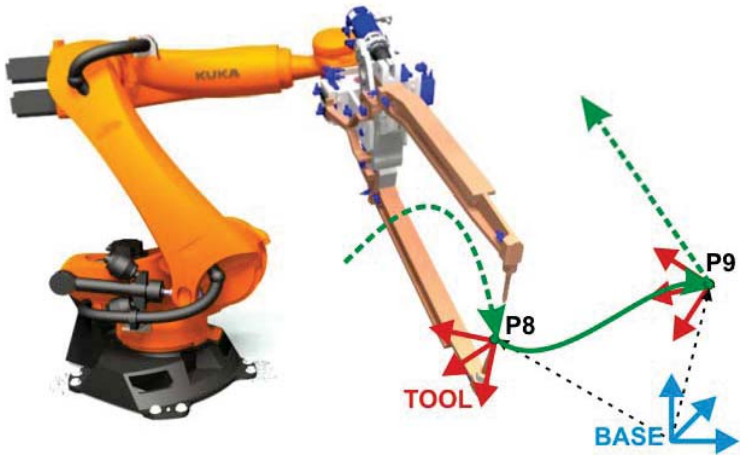


Fig. 10-1: Robot motion

When robot motions have to be programmed, many questions are raised:

Question	Solution	Keyword
How does the robot remember its positions?	The positions of the tool in space are saved (robot position in accordance with the tool and base that are set).	POS E6POS
How does the robot know how to move?	From the specification of the motion type: point-to-point, linear or circular.	PTP LIN CIRC
How fast does the robot move?	The velocity between two points and the acceleration are specified during programming.	Vel. Acc.
Does the robot have to stop at every point?	To save cycle time, points can also be approximated; no exact positioning is carried out in this case.	CONT
What orientation does the tool adopt when a point is reached?	The orientation control can be set individually for each motion.	ORI_TYPE
Does the robot recognize obstacles?	No, the robot “stubbornly” follows its programmed path. The programmer is responsible for ensuring that there is no risk of collisions. There is also a collision monitoring function, however, for protecting the machine.	Collision detection

This information must be transferred when programming robot motions using the teaching method. Inline forms, into which the information can easily be entered, are used for this.

11 Using logic functions in the robot program

11.1 Overview

The following contents are explained in this training module:

- Programming wait functions
- Programming switching functions
- Viewing the current state of inputs and outputs in the I/O monitor

11.2 Introduction to logic programming

Use of inputs and outputs in logic programming

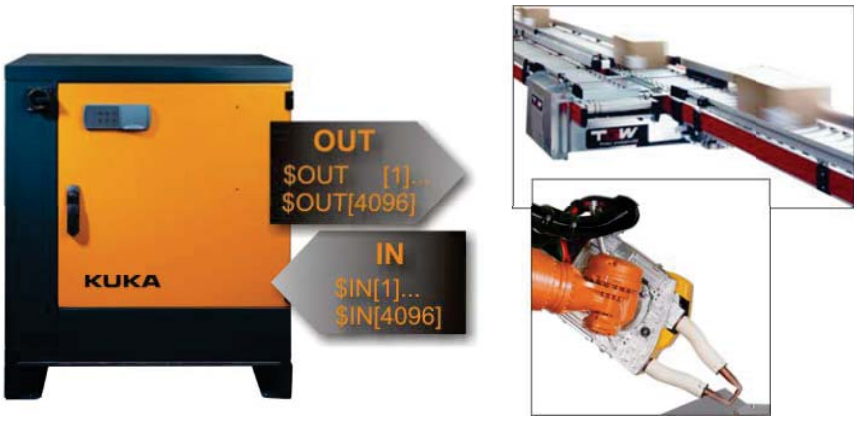


Fig. 11-1: Digital inputs and outputs

In order to implement **communication** *with the* **periphery** *of the robot controller,* **digital** *and* **analog inputs/outputs** *can be used.*

Explanation of terms

Term	Explanation	Example
Communication	Signal exchange via a serial interface	Polling a state (gripper open/closed)
Periphery	“Surroundings”	Tool (e.g. gripper, weld gun, etc.), sensors, material conveyor systems, etc.
Digital	Digital technology: value- and time-discrete signals	Sensor signal: part present: value 1 (TRUE), part not present: value 0 (FALSE)
Analog	Mapping of a physical variable	Temperature measurement
Inputs	The signals *arriving* in the controller via the field bus interface	Sensor signal: gripper is open / gripper is closed
Outputs	The signals *sent* by the controller to the periphery via the field bus interface	Command for switching a valve to close a finger gripper.