ITIA Target Programming (Robot) Lab Report

Stefan Adelmann (01633044) and Hannes Brantner (01614466)

May 15, 2020

1 SysML Review

For the robot station in phase 2, no two SysML models had to be merged. The presented model from phase 1 of this lecture contains the whole station. As a self review we remodeled certain aspects of the robot station.

1.1 Parametric diagram

The parametric diagram that was created in phase 1 has no direct link to the real robot station. Distances do not need to be handled by the user in any way and the robot itself has sophisticated movement schemes pre-installed. Measurements between two points in 3D space therefore cannot be seen as a straight line. The diagram was dropped from the revised SysML model.

1.2 Internal Block Diagram

The diagram does not show the internal wiring of individual sensors and actuators. This was an deliberate choice to not overcrowd the diagram. Information on the internal wiring of the used connectors can be found in separate tables.

2 I/O Mapping

The following I/O mapping was created for the robot station.

2.1 Robot

Movement of the 6 axis robot arm, shown in 1 as well as the function of the multi-functional gripper is handled internally by the robot controller. The user does not need to define these I/O mapping themselves, instead the programming language Melfa uses commands shown in table 2.1 to actuate the arm. Additionally OPC UA methods are shown that execute the corresponding action.

Melfa-Command	**OPC UA**	**Beschreibung**
Mov(X,Y,Z,A,B,C)	move(X,Y,Z,A,B,C)	Modul Roboter (Hand) - Position, Rotation anfahren
HOpen 1	gripperOpen()	Modul Roboter (Hand) - Gripper öffnen
HClose 1	gripperClose()	Modul Roboter (Hand) - Gripper schließen

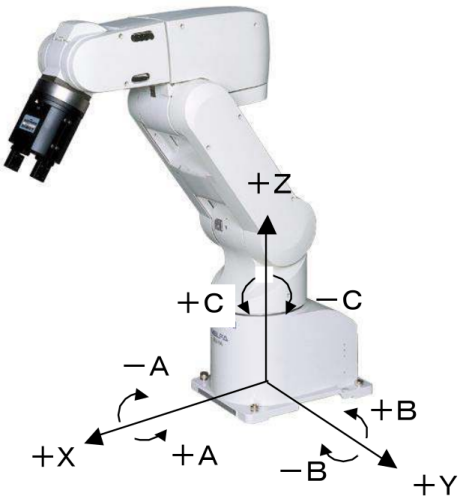

Figure 1: Robot coordinate system in XYZ mode

2.2 Robot Control

The robot station is controlled by a controller that is programmed via the Melfa programming language.

Controller	
Device:	Robot Controller
ID:	CR750-D
MAC:	
IP:	192.168.162.82/25

The user can execute commands and movements of the robot station via an OPC Ua server running on a Raspberry Pi.

OPC UA Gateway	
Device:	RaspberryPi 3
ID:	BCM2835 (a02082)
MAC:	b8:27:eb:09:db:ca
IP:	192.168.162.84/25

2.3 Sensors/Actuators

Inputs and outputs are addressed via their unique index in Melfa and subsequently in the OPC Ua interface.

Index	Beschreibung
1	Modul Roboterhandling - Werkstück ausgerichtet
2	Modul Roboterhandling - Werkstück in Abholposition
3	Bedienfeld - Start (Schließer)
4	Bedienfeld - Stopp (Öffner)
5	Bedienfeld - Reset (Schließer)
7	Bedienfeld - COM Brücke (I7)
8	Modul Robotermontage (Federmagazin) - Schieber eingefahren
9	Modul Robotermontage (Federmagazin) - Schieber ausgefahren
10	Modul Robotermontage (Federmagazin) - Feder vorhanden
12	Modul Robotermontage (Deckelmagazin) - Schieber eingefahren
13	Modul Robotermontage (Deckelmagazin) - Schieber ausgefahren
15	Modul Robotermontage (Deckelmagazin) - Deckel auf Ablage
900	Modul Roboter (Hand) - Teil nicht schwarz

Index	Beschreibung
0	Bedienfeld - Start (LED)
1	Bedienfeld - Reset (LED)
2	Bedienfeld - Q1 (LED)
3	Bedienfeld - Q2 (LED)
4	Bedienfeld - COM Brücke (Q4)
8	Modul Robotermontage (Federmagazin) - Schieber ausfahren
12	Modul Robotermontage (Deckelmagazin) - Schieber ausfahren

3 Handover Protocol

Since the robot station by itself is capsuled no handover protocol between the internal modules is necessary. The robot can however be used in combination with other stations via the OPC Ua interfaced that was created in this phase. A Raspberry Pi is used as a OPC Ua server connected to the robot controller via a telnet interface. This setup provides the following methods:

Method	**Function**
OpenGripper()	Opens the multi function gripper
CloseGripper()	Closes the multi function gripper
Move(X,Y,Z,A,B,C)	Moves the robot arm to the given coordinates
GetErrorLog(NUMLOGS)	Returns the last NUMLOGS errors that occurred
ReadInput(INDEX)	Returns the state of the input at the given index
WriteOutput(INDEX, STATE)	Sets the output at the given INDEX to the given STATE
ResetError()	Resets the robot controller from the error state