ITIA
Information Model Robot
Lab Report

Stefan Adelmann (01633044)
Hannes Brantner (01614466)

June 28, 2020

The information model of the Robot Station consists of a OPC UA Gateway providing methods that interact with the Robot Controller. The OPC UA Gateway is provided by a Raspberry Pi, that can be accessed via the information given in Table 1.

OPC UA Gateway	
Device:	RaspberryPi 3
ID:	BCM2835 (a02082)
MAC:	b8:27:eb:09:db:ca
IP:	192.168.162.84/25
Port:	4840

Table 1: OPC UA Gateway information

The robot controller itself is accessed by the OPC UA gateway via the connection information given in Table 2.

Robot Controller	
Device:	Robot Controller
ID:	CR750-D
MAC:	38:e0:8e:9e:89:8d
IP:	192.168.162.82/25
PORT:	10003

Table 2: Robot Controller information

Together these two components form a system in which all aspects needed for the implementation of the Robot Station can be controlled via the OPC UA interface. In order to control aspects like the spring storage or retrieve information of the built in sensors, the I/O mapping presented in the following section has to be used.

Sensors	
Index	**Description**
1	Robot handling module - workpiece alignment
2	Robot handling module - workpiece in pick-up position
3	Control panel - Start (make contact)
4	Control panel - Stop (normally closed)
5	Control panel - Reset (no contact)
7	Control panel - COM bridge (I7)
8	Robot assembly module (spring magazine) - slider retracted
9	Robot assembly module (spring magazine) - slide extended
10	Module robot assembly (spring magazine) - spring available
12	Robot assembly module (lid magazine) - slide retracted
13	Robot assembly module (lid magazine) - slide extended
15	Module robot assembly (lid magazine) - lid on tray
900	Robot module (hand) - part not black
Actuators	
Index	**Description**
0	Control panel - Start (LED)
1	Control panel - Reset (LED)
2	Control panel - Q1 (LED)
3	Control panel - Q2 (LED)
4	Control panel - COM bridge (Q4)
8	Robot assembly module (spring magazine) - slide out
12	Robot assembly module (lid magazine) - slide out

Table 3: I/O mapping

1 I/O Mapping

The I/O mapping is divided into the two groups of sensors and actuators. It is to note that the movement of the robot arm as well as the movement of the multi-function gripper is omitted from these tables as they are accessed via proprietary OPC UA methods. This distinction is not only present in the OPC UA model but also in the native programming language of the Robot Controller.

2 OPC UA Methods

Table 4 presents all custom methods that are provided by the OPC UA gateway using the generated information model. They are a collection of direct control elements needed to implement certain robot behavior as well as management elements needed to keep the robot running. The later primarily involves error handling as error states need to be reset in order to operate.

OPC UA Methods			
Method	**Description**	**ObjectId**	**MethodId**
ReadInput(Index)	Returns the `Value` of the input at the given `Index`	ns=4;i=1066	ns=4;i=1067
WriteOutput(Index, Value)	Sets the output at the given `Index` to the given `Value`	ns=4;i=1066	ns=4;i=1111
GetMostRecentError()	Returns the most recent error that occurred	ns=4;i=1103	ns=4;i=1105
GetRecentErrors(NumberOfErrors)	Returns the last `NumberOfErrors` errors that occurred	ns=4;i=1066	ns=4;i=1140
ResetError()	Resets the error state of the robot controller	ns=4;i=1102	ns=4;i=1108
OpenGripper()	Opens the multi function gripper	ns=4;i=1130	ns=4;i=1131
CloseGripper()	Closes the multi function gripper	ns=4;i=1130	ns=4;i=1134
Move(X,Y,Z,A,B,C)	Translates the robot arm to the position (`X,Y,Z`) with the rotation (`A,B,C`)	ns=4;i=1130	ns=4;i=1137
MoveToSafePosition()	Moves the robot arm into the predefined safety position	ns=4;i=1113	ns=4;i=1115
RestartServer()	Reboots the OPC UA Gateway device	ns=0;i=2253	ns=4;i=1164

Table 4: Implemented OPC UA methods

The I/O mapping is needed for the first two methods, ReadInput and WriteOutput where the needed index parameter corresponds to the indices given in Table 3.
The ObjectID and the MethodId can directly be used in the NodeRED interface implementation in this laboratory.

3 OPC UA Objects

The OPC UA methods from Table 4 are encapsulated in the OPC UA objects given in Table 5.

OPC UA Objects		
Object	**Description**	**ObjectId**
MethodSet (Controller)	The MethodSet object, which is a component of the Controller object	ns=4;i=1066
GetMostRecentError	The GetMostRecentError object, which is a component of the ErrorLog Functions folder	ns=4;i=1103
ResetError	The ResetError object, which is a component of the ResetError Functions folder	ns=4;i=1102
MethodSet (MotionDevice)	The MethodSet object, which is a component of the MotionDevice object	ns=4;i=1130
MoveToSafePosition	The MoveToSafePosition object, which is a component of the MoveToSafeState Functions folder	ns=4;i=1113
Server	The Server object, which is organized by the Objects folder	ns=0;i=2253

Table 5: OPC UA objects

4 OPC UA Namespaces

As it can be seen in Table 6, most of the objects and therefore methods are provided by the namespace 4, Mitsubishi Electric Robot Nodeset, with the exception of the restartServer() method in the Server object (ns=0). This method is capsuled from the Robot namespace as it will actually restart the OPC UA gateway Raspberry Pi and therefore acts on the Server directly.

OPC UA Namespaces		
NamespaceIndex	**NamespaceUri**	**Description**
0	http://opcfoundation.org/UA/	Standard Nodeset
2	http://opcfoundation.org/UA/DI/	Devices Nodeset
3	http://opcfoundation.org/UA/Robotics/	Robotics Nodeset
4	http://auto.tuwien.ac.at/UA/MitsubishiElectricRobot/	Mitsubishi Electric Robot Nodeset

Table 6: OPC UA namespaces