Primary interface (XML RPC)

Extensible Markup Language Remote Procedure Call, XML-RPC for short. This technology enables methods to be executed remotely, with data being transferred via http. The data to be transferred has XML formatting.

Table of Contents

> This article is valid for all horstOS versions as well as horstFX.

The data being transferred is formatted in XML. XML-RPC is language-independent, allowing for XML-RPC clients in various programming languages, making integration into existing projects straightforward. The commands are described independently of the programming language used. Additionally, example clients are provided to facilitate a smoother onboarding process.

This article explains how to perform the actions described below through the External Interface. To begin, a connection with the robot must be established, as detailed in the article "Python 3: Establishing Connection with the Robot."

> Robot control software version: 2020.10 and newer

1. Switching Views

1.1 Command

HorstFX.Activity.switchActivity(id)

1.2 Parameters

The documentation on horstOS/FX Views provides information on which view is assigned to which ID.

Name	Description	Example value
id	View ID	0

1.3 Return value

true Upon success.

2. Accessing current joint angles

2.1 Command

HorstFX.Robotcontrol.getCurrentRobotJoints()

2.2 Parameters

This function does not have any function arguments.

2.3 Return value

Returns a map upon success.

Joint angles are specified in degrees.

Name	Description
j1	axis 1
j2	axis 2
j3	axis 3
j4	axis 4
j5	axis 5
j6	axis 6

3. Accessing the Current Pose

3.1 Command

HorstFX.Robotcontrol.getCurrentRobotPose()

3.2 Parameters

This function does not have any function arguments.

3.3 Return value

Returns a map in case of success.

Name	Description
x	X-coordinate in the base coordinate system
y	Y-coordinate in the base coordinate system
z	Z-coordinate in the base coordinate system
q0	Quaternion 0
q1	Quaternion 1
q2	Quaternion 2
q3	Quaternion 3
rx	Euler angles X
ry	Euler angles Y
rz	Euler angles Z

4. Turning Outputs On/Off

4.1 Command

HorstFX.Robotcontrol.setOutput(outputName, value)

4.2 Parameters

Name	Description	Example value
outputName	Output Name	"OUTPUT_1"
value	Value to be set	1

4.3 Return value

true upon successful completion

5. Confirming Operation Mode Change

5.1 Command

HorstFX.Safety.confirmChangeOperationMode()

5.2 Parameters

This function does not have any function arguments.

5.3 Return value

true when the operation mode change could be acknowledged.

6. Reading Boolean Registers

6.1 Command

HorstFX.Variable.getBoolRegister(registerIndex)

6.2 Parameters

Name	Description	Example value
registerIndex	Register index position ranges from 0 to 127.	0

6.3 Return value

Current value of the requested register (Bool value)

7. Writing to Boolean Registers

7.1 Command

HorstFX.Variable.setBoolRegister(registerIndex, value)

7.2 Parameters

Name	Description	Example value
value	Value to be set	true
registerIndex	Position index of the Bool Register ranges from 0 to 127.	0

7.3 Return value

true upon successful completion

8. Reading Inputs

8.1 Command

HorstFX.Robotcontrol.getInput(inputName)

8.2 Parameters

Name	Description	Example value
inputName	Input Name	"INPUT_1"

8.3 Return value

Current value of the requested input (Integer value)

9. Extended Move Command

9.1 Command

HorstFX.Robotcontrol.moveAdvanced(configMap)

9.2 Parameters

configMap	Map <String, Object>

The map contains the same mandatory and optional parameters as the extended Move Command in textual programming. For a detailed description, refer to the article "Extended Move Command."

9.3 Return value

true upon successful completion

10. Confirming Acknowledgment of External Emergency Stop

> Can only be acknowledged if the external emergency stop has been reset.

10.1 Command

HorstFX.Safety.confirmExternalEmergencyStop()

10.2 Parameters

This function does not have any function arguments.

10.3 Return value

true when the external emergency stop could be acknowledged.

11. Reading Float Registers

11.1 Command

HorstFX.Variable.getFloatRegister(registerIndex)

11.2 Parameters

Name	Description	Example value
registerIndex	Position index ranges from 0 to 63.	0

11.3 Return value

Current value of the requested register (Float value)

12. Writing to Float Registers

12.1 Command

HorstFX.Variable.setFloatRegister(registerIndex, value)

12.2 Parameters

Name	Description	Example value
registerIndex	Position index of the Float Register ranges from 0 to 63.	0
value	Value to be set	13.37

12.3 Return value

true upon successful completion

13. Reading Global Speed

13.1 Command

HorstFX.Program.getGlobalSpeed()

13.2 Parameters

This function does not have any function arguments.

13.3 Return value

Current global speed (Float value between 0 - 1, representing 0 - 100%)

14. Setting Global Speed

14.1 Command

HorstFX.Program.setGlobalSpeed(globalSpeed)

14.2 Parameters

Name	Description	Example value
globalSpeed	New value for the global speed	0.5

14.3 Return value

true upon successful completion

15. Acknowledging Internal Errors

15.1 Command

HorstFX.Safety.confirmInternalError()

15.2 Parameters

This function does not have any function arguments.

15.3 Return value

true when the internal errors could be acknowledged.

16. Reading Integer Registers

16.1 Command

HorstFX.Variable.getIntRegister(registerIndex)

16.2 Parameters

Name	Description	Example value
registerIndex	Position index ranges from 0 to 63.	0

16.3 Return value

Current value of the requested register (Integer value)

17. Writing to Integer Registers

17.1 Command

HorstFX.Variable.setIntRegister(registerIndex, value)

17.2 Parameters

Name	Description	Example value
registerIndex	Position index of the Integer Register ranges from 0 to 63.	0
value	Value to be set	42

17.3 Return value

true upon successful completion

18. Joint Movement

18.1 Command

HorstFX.Robotcontrol.moveJoint(x, y, z, q0, q1, q2, q3, speed)

18.2 Parameters

Name	Description	Example value
x	X-coordinate in the base coordinate system	0.50975
y	Y-coordinate in the base coordinate system	0.0
z	Z-coordinate in the base coordinate system	0.82743
q0	Quaternion 0	0.7071
q1	Quaternion 1	0
q2	Quaternion 2	0.7071
q3	Quaternion 3	0
speed	Speed in percentage (0-1)	0.5

18.3 Return value

true upon successful completion

19. Lineare Bewegung

> Depending on the robot's position, a linear path may not be feasible. This can occur, for instance, if the robot would collide with itself or if the movement cannot be executed linearly due to kinematic constraints.

19.1 Command

HorstFX.Robotcontrol.moveLinear(x, y, z, q0, q1, q2, q3, speed)

19.2 Parameters

Name	Description	Example value
x	X-coordinate in the base coordinate system	0.50975
y	Y-coordinate in the base coordinate system	0.0
z	Z-coordinate in the base coordinate system	0.82743
q0	Quaternion 0	0.7071
q1	Quaternion 1	0
q2	Quaternion 2	0.7071
q3	Quaternion 3	0
speed	Speed in percentage (0-1)	0.5

19.3 Return value

true upon successful completion

20. Reading Next Joints

20.1 Command

HorstFX.Variable.getNextJoints(

20.2 Parameters

This function does not have any function arguments.

20.3 Return value

Returns a map upon successful completion.

Name	Description
j1	Joint angles axis 1
j2	Joint angles axis 2
j3	Joint angles axis3
j4	Joint angles axis 4
j5	Joint angles axis 5
j6	Joint angles axis 6

21. Setting Next Joints

21.1 Command

HorstFX.Variable.nextJoints(j1, j2, j3, j4, j5, j6)

21.2 Parameters

Name	Description	Example value
j1	Joint angles axis 1	10
j2	Joint angles axis 2	-5
j3	Joint angles axis 3	15
j4	Joint angles axis 4	20
j5	Joint angles axis 5	0
j6	Joint angles axis 6	100

21.3 Return value

true upon successful completion

22. Setting Next Pose

22.1 Command

HorstFX.Variable.nextPose(x, y, z, q0, q1, q2, q3,)

22.2 Parameters

Name	Description	Example value
x	X-coordinate in the base coordinate system	0.50975
y	Y-coordinate in the base coordinate system	0.0
z	Z-coordinate in the base coordinate system	0.82743
q0	Quaternion 0	0.7071
q1	Quaternion 1	0
q2	Quaternion 2	0.7071
q3	Quaternion 3	0

22.3 Return value

true upon successful completion

23. Nothalt quittieren

> Can only be acknowledged if the emergency stop on the PANEL has been released.

23.1 Command

HorstFX.Safety.confirmEmergencyStop()

23.2 Parameters

This function does not have any function arguments.

23.3 Return value

true when the emergency stop acknowledgment was successful.

24. Canceling Program Execution

24.1 Command

HorstFX.Program.abort()

24.2 Parameters

This function does not have any function arguments.

24.3 Return value

true if the program can be aborted.

25. Pause Program Execution

25.1 Command

HorstFX.Program.pause()

25.2 Parameters

This function does not have any function arguments.

25.3 Return value

true if the program can be stopped.

26. Sending and starting a program

26.1 Command

HorstFX.Program.execute()

26.2 Parameters

Name	Description
program	programHorstSCRIPT as a string26.3 Return value

true when the program has been sent and started.

27. Starting a Program

27.1 Command

HorstFX.Program.play()

27.2 Parameters

This function does not have any function arguments.

27.3 Return value

true if the program can be started.

28. Proceeding with Program Execution

28.1 Command

HorstFX.Program.proceed()

28.2 Parameters

This function does not have any function arguments.

28.3 Return value

true when the program can be restarted.

29. Checking if the program is running

29.1 Command

HorstFX.Program.isRunning()

29.2 Parameters

This function does not have any function arguments.

29.3 Return value

true when a program is currently being executed.

30. Querying Robot Status

30.1 Command

HorstFX.Safety.status()

30.2 Parameters

This function does not have any function arguments.

30.3 Return value

Returns a text describing the error. If no error is present, "OK" will be returned.

31. Changing Tool

31.1 Command

HorstFX.Robotcontrol.setTool(toolName)

31.2 Parameters

Name	Description	Example value
toolName	Tool Name	"No Tool"

31.3 Return value

true when the corresponding tool has been found and set.

32. Reading Tool Offset

4.1 Command

HorstFX.Robotcontrol.getToolOffset(toolName)

4.2 Parameters

This function does not have any function arguments.

5.3 Return value

Returns a map upon successful completion.

Name	Description
x	X-coordinate in the base coordinate system
y	Y-coordinate in the base coordinate system
z	Z-coordinate in the base coordinate system
rx	Euler angles X
ry	Euler angles Y
rz	Euler angles Z

Related articles

Activating the Profinet interface on the robot

EUROMAP67 Interface

Cable interface (Machine Tending)

Example programs XML/RPC

Profinet communication