

CPR Command Reference

V04-02 for CPRog V902-10-050, TinyCtrl V980-04-071

October 2019

1. Summary

	Command	Functionality	CPRog	TinyCtrl
	Motion			
0	All: Abort Condition	Motions commands can be interrupted	✓	✓
0	All: Smoothing	Smooth transition between motions	✓	✓
0	All: Acceleration	Defined accelerations for motions	✓	✓
6.1	Joint	Move in joint coordinates	✓	✓
6.2	JointByVariable	As above, target defined by variable	✓	✓
6.3	Linear	Move linear in cartesian xyzabc position	✓	✓
6.4	LinearByVariable	As above, target defined by variable	✓	✓
6.5	RelativeBase	Relative motion in base coordinates	✓	✓
6.6	RelativeTool	Relative motion in tool coordinates	✓	✓
6.7	RelativeJoint	Relative motion in joint coordinates	✓	✓
	Input / Output			
7.1	DigitalOut	Set a digital output to high or low	✓	✓
	Gripper	Open or close the gripper	✓	✓
	Structure			
	Simple conditions	e.g. DIn21	✓	✓
	Complex conditions	e.g. DIn21 AND DIn24 or targetpos.x > 200	✓	✓
	Internal Signals	e.g. GSig1 to remember internal states	✓	✓
8.1	LoopCounting	Repeat a defined number of iteration	✓	✓
8.2	LoopConditional	Repeat until a condition is true	✓	✓
8.3	If-Then-Else	Conditional branch	✓	✓
8.4	Wait	Wait a defined time span	✓	✓
8.5	WaitConditional	Wait until a condition is true	✓	✓
8.6	Sub	Call a sub program defined in another .xml file	✓	✓
	Advanced			
9.1	DefPosVariable	Defines a Position Variable	✓	✓
9.2	DefNumberVariable	Defines a Number Variable	✓	✓
	Variable Operations	Add, Subtract, Set values to variables and variable sections	✓	✓
9.6	PluginTargetPos	Get a target position from e.g. a camera	✓	partly

2. Names of Programs and Sub-Programs

2.1 Program Names

Program names can consist of the standard characters. Only some special characters are allowed.

Standard characters: a-z A-Z 0-9

Special characters on Windows: . - _ \

Special characters on Linux: . - _ /

2.2 Program Location

Robot programs are .xml files in the /Data/Programs/ folder. They can be stored directly in the Programs/ folder, or in a subfolder, e.g. Programs/robotStation01/

2.3 Sub-Programs

The main program can call sub-programs, please see section 7.6. The location of the called sub-program is relative to Data/Programs/.

3. Variables and Variable Access

CPRog can work with number variables (for both integer and float values) and position variables (Cartesian positions and joint positions). It is also possible to access the elements of the variables.

3.1 User Variables

With the commands "DefPosVariable" and "DefNumberVariable" user variables can be defined.

User variable names are case insensitive, they are transformed to lower case.

3.2 System Variables

The following variables are always available:

- #position

The current robot position with Cartesian coordinates in mm for XYZ and degrees for ABC, as well as joint positions in degrees or mm depending on the type of joint.

3.3 Element Access

The following elements of the variables can be accessed with the "." Operator:

- Position: x, y, z
- Rotation: A, B, C
- Joints: A1, A2, A3, A4, A5, A6

3.4 Example

The following example moves the robot to the left or to the right, depending on the starting position.

Nr	Cmd	X	Y	Z	A	B	C	E1	E2	E3	vel	acc	smoo	cond	Description
0	Wait	1 s													
1	If	#position.y > 0.0													
2	RelativeBase	X 0	Y 50	Z 0	A 0	B 0	C 0	E1 0	E2 0	E3 0	10...	40 %	50 %	false	
3	Else														
3	RelativeBase	X 0	Y -50	Z 0	A 0	B 0	C 0	E1 0	E2 0	E3 0	10...	40 %	50 %	false	
4	EndIf														

4. Conditions

Conditions are used in if-then-else commands, conditional loops and as abort condition in motion commands. The conditional statements can be combinations of digital inputs, Boolean operators and comparisons. E. g.:

- DIn23 True if digital input 23 is high
- DIn23 AND !DIn27 True if digital input 23 is high and digital input 27 is not high
- modelclass = 31 True if variable modelclass is 31
- mempos.x > 350.0 True if the x value of the position variable mempos is higher then 350

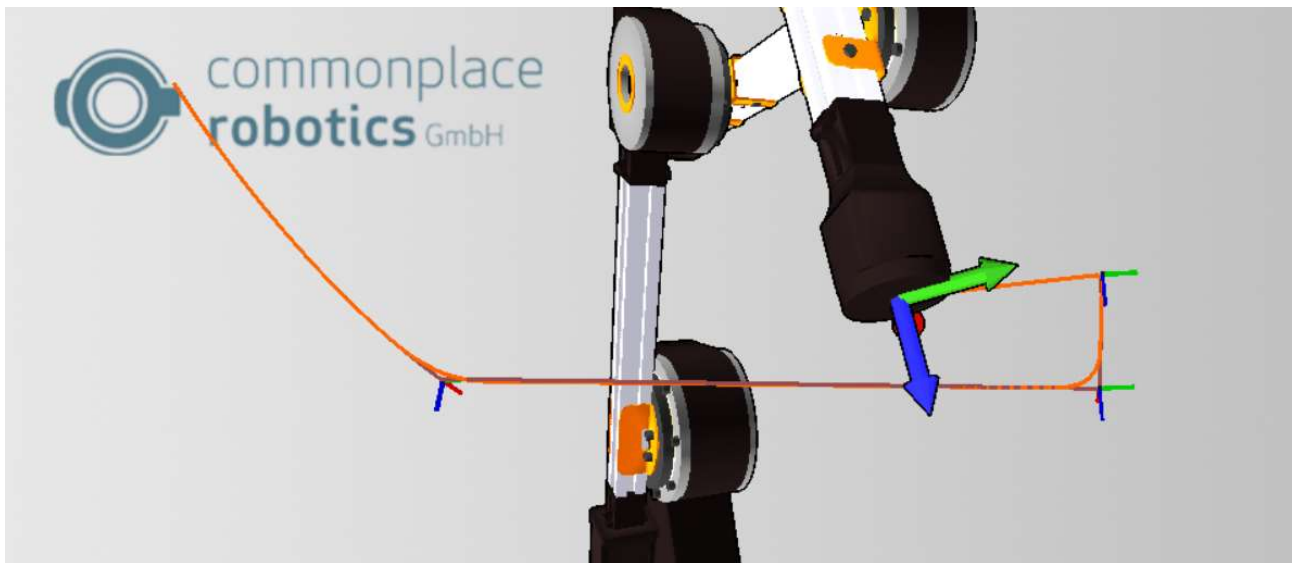
EBNF definition:

Expression	:= ["!"] <Boolean> { <BooleanOperator> <Boolean> } ...
Boolean	:= <BooleanConstant> <Expression> "(" <Expression> ")" CompExpression "(" <CompExpression> ")"
BooleanOperator	:= "And" "Or"
BooleanConstant	:= "True" "False"
DigitalInputs	:= "DIn1" "GSig1"
CompExpression	:= <CompValue> <Comparator> <CompValue>
CompValue	:= <Variable> <Number>
Variable	:= some variable name
Number	:= Integer or Float number
Comparator	:= "=" ">" "<" ">=" "<="

5. Acceleration and Smoothing

Motion commands are adapted to follow the joint accelerations. This can be done for a single motion command, but also for a combination of up to 10 commands.

It is also possible to smooth out corners. This allows the robot to go through the corner with a remaining velocity. If corner smoothing is not active the robot stops at every corner for a short moment. This leads to more abrupt motion.



Parameters:

- Acc: Percentage of the possible acceleration, 0 .. 100
- Smooth: Scope of smoothing in percent, 0..100

Non-motion commands like "Wait" or "DOut" interrupt the smoothing of a motion set.

6. Motion - Detailed Command Reference

6.1 Joint Motion

Moves to a target position in joint coordinates. The joint velocities during the motion are constant. The resulting motion of the TCP is normally not a straight line, but curved.

Nr	Cmd	X	Y	Z	A	B	C	E1	E2	E3	vel	acc	smoc	cond	Description
0	Joint	A1 0	A2 20	A3 10	A4 15	A5 0	A6 0	E1 0	E2 0	E3 0	50 %	40 %	50 %	false	

- A1 to A6: Joint values of the target position in degree
- E1 to E3: Joint values of the external joints, e.g. linear joints
- Vel: Velocity in percent of the max joint velocities
- Acc: Acceleration in % of the max joint acceleration
- Smooth: Interpolation at the edge points
- Condition: Abort condition, if this is true the motion will be aborted
- Description: Free field to comment the line

6.2 Linear Motion

A standard linear motion interpolating both position and orientation.

Nr	Cmd	X	Y	Z	A	B	C	E1	E2	E3	vel	acc	smoo	cond	Description
0	Linear	X 568.8	Y 15	Z 230.1	A 177.9	B 26.21	C 178.4	E1 0	E2 0	E3 0	10...	40 %	50 %	false	

- X, Y, Z: Cartesian position of the target position in mm
- A, B, C: Orientation of the target position in degree, encoding in Euler angles
- E1 to E3: Joint values of the external joints, e.g. linear joints
- vel: Velocity in mm/s
- acc: Acceleration in % of the max joint acceleration
- smooth: Interpolation at the edge points
- condition: Abort condition, if this is true the motion will be aborted
- Description: Free field to comment the line

6.3 LinearByVariable

A linear motion with the target position defined by a variable

Nr	Cmd	X	Y	Z	A	B	C	E1	E2	E3	vel	acc	smoo	cond	Description
0	Joint	A1 25.68	A2 22.9	A3 -1.42	A4 17.92	A5 3.24	A6 0	E1 0	E2 0	E3 0	50 %	40 %	50 %	false	
1	DefPosVariable	myposvar	current												
2	Linear	X 568.8	Y 15	Z 230.1	A 177.9	B 26.21	C 178.4	E1 0	E2 0	E3 0	10...	40 %	50 %	false	
3	LinearByVaria...	myposvar									10...	40 %	50 %	false	

- Field X: Name of the target position variable
- vel: Velocity in mm/s
- acc: Acceleration in % of the max joint acceleration
- smooth: Interpolation at the edge points
- condition: Abort condition, if this is true the motion will be aborted
- Description: Free field to comment the line

6.4 JointByVariable

A joint motion with the target position defined by a variable. Details are as above.

6.5 Relative Base Motion

A relative linear motion in base coordinates, interpolating the position (without orientation).

Nr	Cmd	X	Y	Z	A	B	C	E1	E2	E3	vel	acc	smooth	condition	Description
0	RelativeBase	X 0.0	Y 0.0	Z 50.0							100 mm/s	40.0 %	50 %	false	

- X, Y, Z: relative Cartesian motion in mm base coordinate system
- vel: Velocity in mm/s
- acc: Acceleration in % of the max joint acceleration
- smooth: Interpolation at the edge points
- condition: Abort condition, if this is true the motion will be aborted
- Description: Free field to comment the line

6.6 Relative Tool Motion

A relative linear motion in tool coordinates, interpolating the position (without orientation).

Nr	Cmd	X	Y	Z	A	B	C	E1	E2	E3	vel	acc	smooth	condition	Description
0	RelativeBase	X 0.0	Y 0.0	Z 50.0							100 mm/s	40.0 %	50 %	false	

- X, Y, Z: relative Cartesian motion in mm tool coordinate system
- vel: Velocity in mm/s
- acc: Acceleration in % of the max joint acceleration
- smooth: Interpolation at the edge points
- condition: Abort condition, if this is true the motion will be aborted
- Description: Free field to comment the line

6.7 Relative Joint Motion

A relative joint motion, interpolating the joint position.

Nr	Cmd	X	Y	Z	A	B	C	E1	E2	E3	vel	acc	smoo	cond
0	RelativeJoint	A1 0.0	A2 0.0	A3 10.0	A4 0.0	A5 0.0	A6 0.0	E1 0.0	E2 0.0	E3 ...	50 %	40....	50 %	false

- A1 to A6: relative joint motions in degrees
- E1 to E3: relative external joint motions in degree or mm
- vel: Velocity in %
- acc: Acceleration in % of the max joint acceleration
- smooth: Interpolation at the edge points
- condition: Abort condition, if this is true the motion will be aborted
- Description: Free field to comment the line

7. Input / Output – Detailed Command Reference

7.1 DigitalOut

This command sets available outputs:

- Digital outputs, that means real hardware outputs on the robot
- Global signals, that means CPRog-internal markers to allow synchronization e.g. with conveyors

Nr	Cmd	X	Y	Z	A	B	C	E1	E2	E3	vel	acc	smoo	cond	Description
0	Wait	1.0 s													
1	DigitalOut	DOut1	true												
2	DigitalOut	GSig3	true												
3	Wait	1.0 s													
4	DigitalOut	DOut1	false												
5	Wait	0.1 s													

- Field X: The digital output to set:
 - Hardware outputs: DOut1 to DOut4, or DOut21 to DOut27, DOut 31 to DOut 37 and DOut41 to DOut47 for the digital output modules.
 - Internal marker: GSig1 to GSig100
- Field Y: The state to set the output to

8. Structure – Detailed Command Reference

8.1 LoopCounting

The commands between the start tag „LoopCounting“ and the end tag „LoopEnd“ are repeated.

Nr	Cmd	X	Y	Z	A	B	C	E1	E2	E3	vel	acc	smoo	cond	Description
0	LoopCounting	Times: 5													
1	RelativeB...	X -30	Y 0	Z -30	A 0	B 0	C 0	E1 0	E2 0	E3 0	10...	40...	50...	false	
2	Wait	1 s													
3	LoopEnd														

- Times: Insert the number or repetition times here
- Description: Free field to comment the line

8.2 LoopConditional

The LoopConditional command repeats the enclosed commands until the condition in field x is evaluated as true. The condition can be a single digital input, but also a more complex condition as described in section 4.

Nr	Cmd	X	Y	Z	A	B	C	E1	E2	E3	vel	acc	smo	conc	Description
0	LoopConditional	DIn4													
1	RelativeBase	X -30	Y 0	Z -30	A 0	B 0	C 0	E1 0	E2 0	E3 0	10...	40...	50...	false	
2	Wait	2 s													
3	LoopEnd														

- Field X: condition, the loop stops if this is evaluated true
 - Simple example: DIn4 - waiting for digital input 4 being high
 - More complex example: DIn4 OR varCnt >= 5
- Description: Free field to comment the line

The more complex example is shown below. The RelativeBase motion is repeated until DIn4 is high, or until the variable varcnt is equal (or higher) than 5. So max 5 iterations are done.

Nr	Cmd	X	Y	Z	A	B	C	E1	E2	E3	vel	acc	smoo	cond	Description
0	Def Number ...	varcnt	0												
1	LoopConditio...	DIn4 or varCnt >= 5													
2	RelativeBase	X -30	Y 0	Z -30	A 0	B 0	C 0	E1 0	E2 0	E3 0	10...	40 %	50 %	false	
3	Variable Add	varcnt	1												
4	Wait	2 s													
5	LoopEnd														

8.3 If-then-else

This command evaluates a condition. Depending on the result different command blocks are executed.

Nr	Cmd	X	Y	Z	A	B	C	E1	E2	E3	vel	acc	smoo	cond	Description
0	Wait	1 s													
1	If	#position.y > 0.0													
2	RelativeBase	X 0	Y 50	Z 0	A 0	B 0	C 0	E1 0	E2 0	E3 0	10...	40 %	50 %	false	
3	Else														
3	RelativeBase	X 0	Y -50	Z 0	A 0	B 0	C 0	E1 0	E2 0	E3 0	10...	40 %	50 %	false	
4	EndIf														

Condition: The condition in the X field is described in section 4.

8.4 Wait

The Wait command tells the robot to wait for a specified time span without motion.

Nr	Cmd	X	Y	Z	A	B	C	E1	E2	E3	vel	acc	smo	con	Description
0	Wait	1.5 s													

- Field X: Time span in seconds
- Description: for your comments

8.5 WaitConditional

This command waits until a condition is met, e.g. a digital input is high.

Nr	Cmd	X	Y	Z	A	B	C	E1	E2	E3	vel	acc	smo	con	Description
0	WaitConditional	DIn2													

- Field X: Condition as shown in section 0
- Description: for your comments

8.6 Sub

Call of a sub program.

The filename is relative to the location of the calling program.

Nr	Cmd	X	Y	Z	A	B	C	E1	E2	E3
0	Sub	filename.xml								

9. Advanced – Detailed Command Reference

9.1 DefPosVariable

Defines a position variable. Regarding the naming, only the characters are relevant. Upper or lower case is not differentiated.

Nr	Cmd	X	Y	Z	A	B	C	E1	E2	E3	vel	acc	smoo	cond	Description
0	DefPosVariable	myposvar	current												

Incomplete implementation: Currently there are two possibilities for initialization:

- Single word in field Y: The variable is initialized with the current position
- Six float values in field Y: The float values are used as xyzabc values
- It is currently not possible to define a variable based on another variable

An example for the usage of variables and operations is shown below:

Nr	Cmd	X	Y	Z	A	B	C	E1	E2	E3	ve
0	Joint	A1 0	A2 10	A3 0	A4 15.65	A5 0	A6 0	E1 0	E2 0	E3 0	50
1	DefPosVariable	myposvar	current								
2	DefNumberVariable	mynrvar	20								
3	Joint	A1 10	A2 10	A3 0	A4 15.65	A5 0	A6 0	E1 0	E2 0	E3 0	50
4	VariableAdd	mynrvar	-3.5								
5	VariableSubtract	myposvar.x	10.0								
6	VariableSet	myposvar.a	42.42								
7	Joint	A1 0	A2 10	A3 0	A4 15.65	A5 0	A6 0	E1 0	E2 0	E3 0	50

The current values of the variables are shown in the “Variable” tab at the bottom of CPRog. Please pay attention to the A value of 42.42 degrees, and the value of mynrvar: $20 - 3.5 = 16.5$

Nr. of Programs: 1 --> P0: test_VariableOperations.xml with 2 Variables: myposvar: x=634.7 y=0.0 z=525.7 - A=42.4 B=64.3 C=180.0 mynrvar: 16.5															
Log Messages	InfoCenter	Jog	Input / Output	DIN-Rail Input/Output	Variables										

9.2 DefNumberVariable

A number variable contains a floating point value. It can be used in conditions, e.g. to stop a loop. It is initialized with the value in the field Y. See the example above for usage.

9.3 VariableAdd

This command adds two values:

- First value in field X:
 - A position variable with the selected element, e.g. `targetpos.x`
 - A number variable
- Second value in field Y:
 - An integer or floating point value, e.g. `-32.6`
 - A position variable with the selected element, e.g. `targetpos.x`
 - A number variable

The result will be stored into the variable or variable element that is specified in the field X. See the above example for the usage.

9.4 VariableSubtract

This command subtracts two values:

- First value in field X:
 - A position variable with selected element, e.g. `targetpos.x`
 - A number variable
- Second value in field Y:
 - An integer or floating point value, e.g. `-32.6`
 - A position variable with selected element, e.g. `targetpos.x`
 - A number variable

The result will be stored into the variable or variable element that is specified in the field X. See the above example for the usage.

9.5 VariableSet

This command sets a variable to a value:

- First value in field X:
 - A position variable with selected element, e.g. `targetpos.x`
 - A number variable
- Second value in field Y:
 - An integer or floating point value, e.g. `-32.6`
 - A position variable with selected element, e.g. `targetpos.x`
 - A number variable

See the above example for the usage.

9.6 PluginTargetPos

This command receives a target position from a plugin module. It is intended e.g. to forward gripping positions found by a camera system to the CPRog program.

Nr	Cmd	X	Y	Z	A	B	C	E1	E2
0	Joint	A1 0	A2 14.04	A3 2.8	A4 73.63	A5 0	A6 0	E1 0	E2
1	DefPosVariable	targetvar	current						
2	DefNumberV...	modelvar	0						
3	Plugin	Plugin3DCameraPickit	targetvar	modelvar					
4	LinearByVaria...	targetvar							

Before calling the plugin command the necessary variables must be defined:

- A position variable. The plugin writes the target position into it. Depending on the plugin the old values may be completely replaced, the new value is added.
- A number variable: Used to e.g. provide a result like the type of object found. Plugin specific semantics.

Afterwards a linear motion to the target position can be performed.

For TinyCtrl a camera can be included with the same syntax. A library interface is currently not implemented.