BOSCH
Invented for life

Script runner

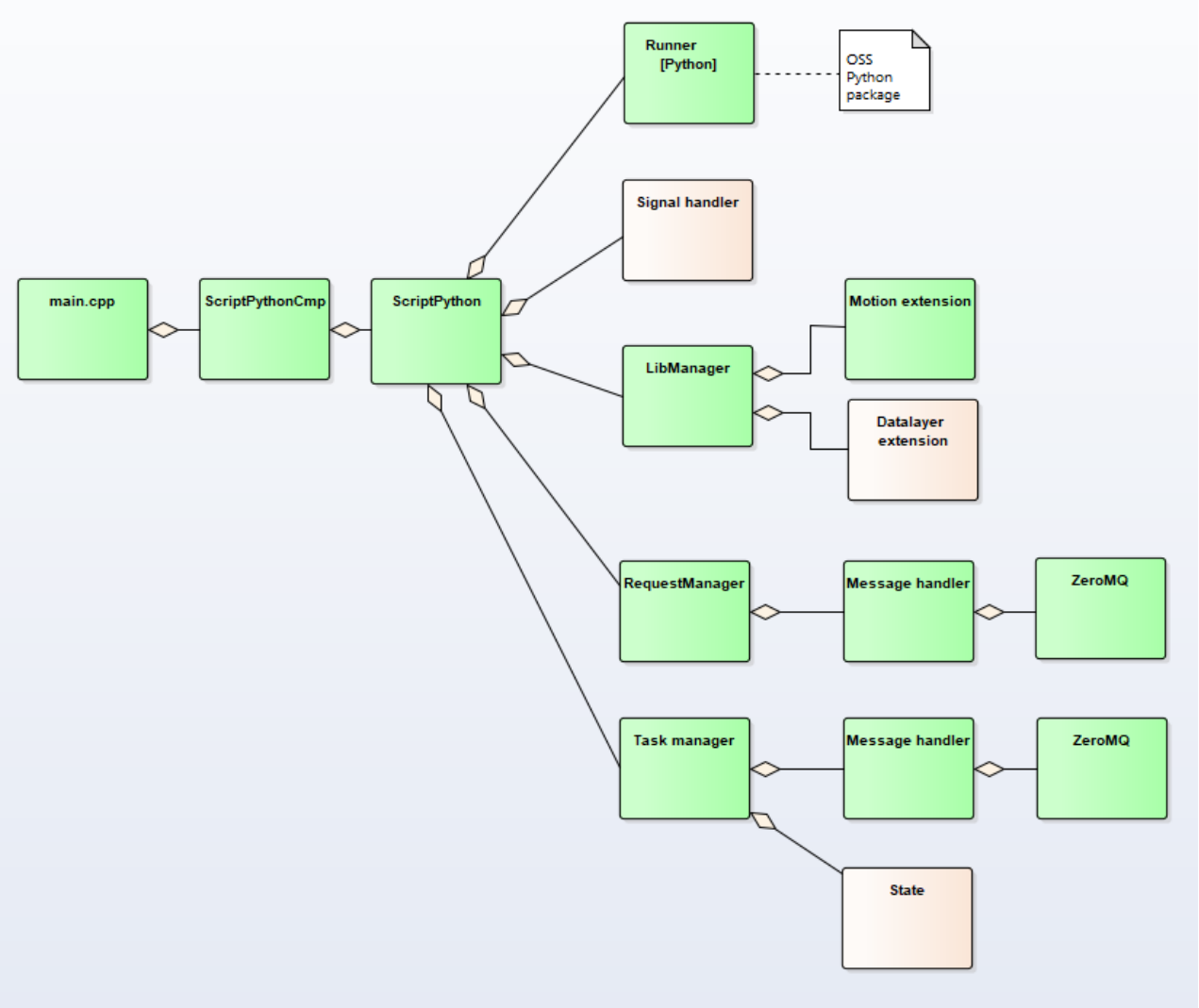
inside.Docupedia Export

Author: Schroeder Thomas (DC-AE/ESW2)
Date: 05-May-2020 10:50

Table of Contents

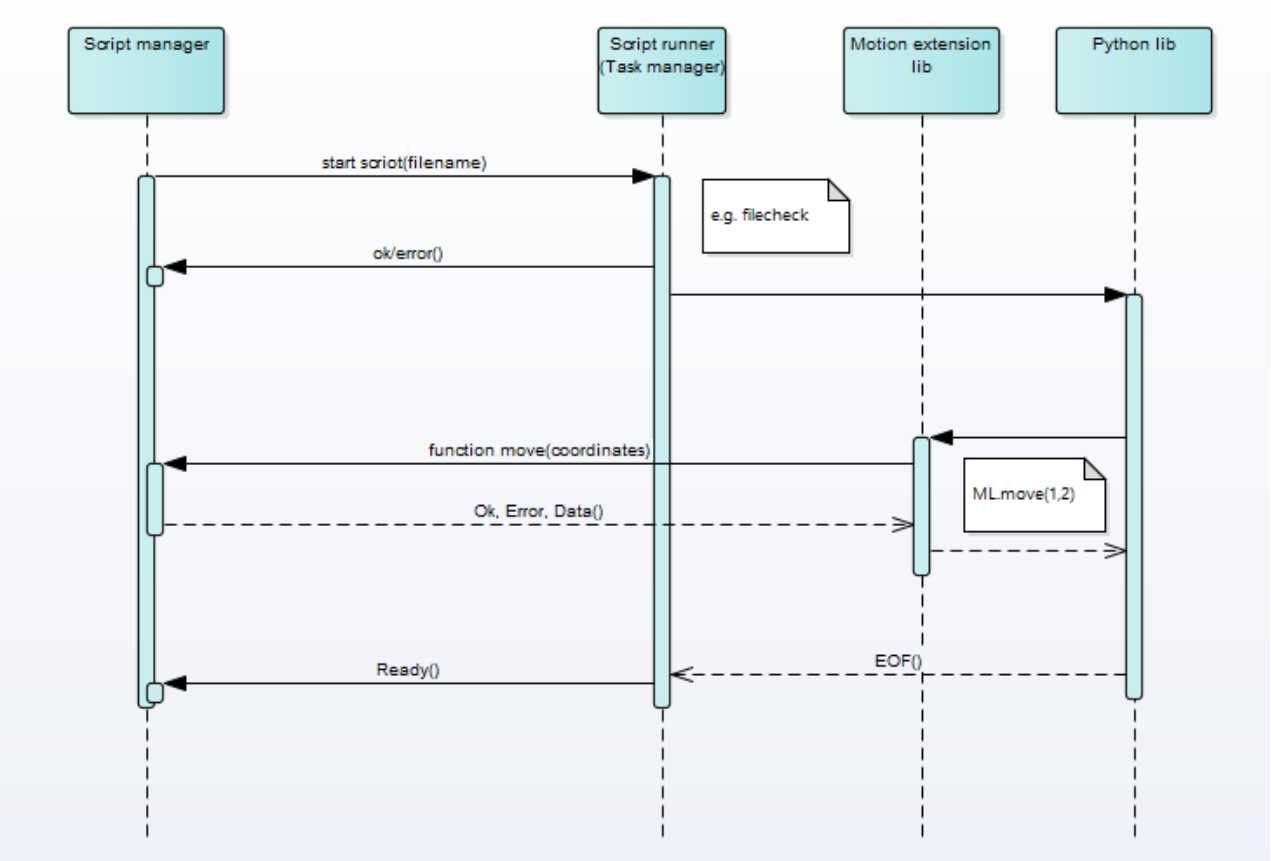
Copyright Robert Bosch GmbH. All rights reserved, also regarding any disposal, exploration, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

Components (green for prototype)



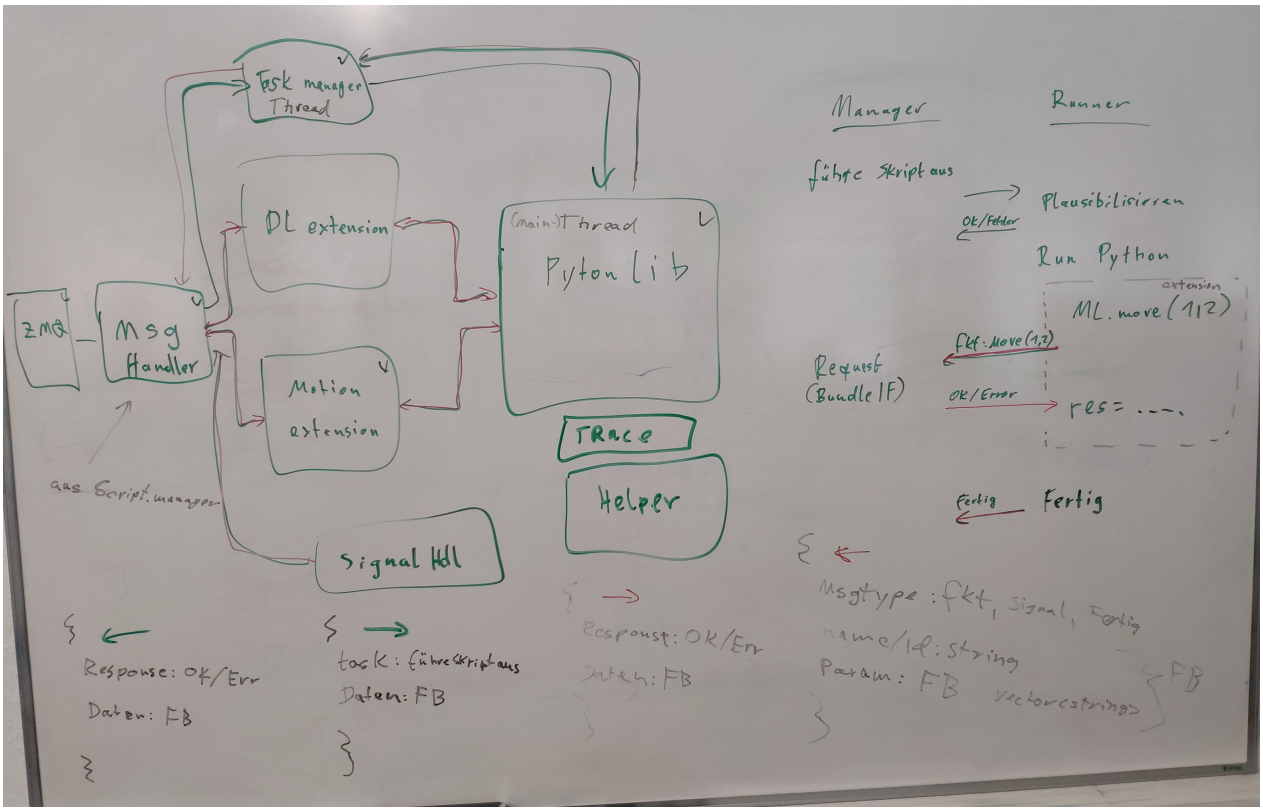
Request Sequence

Copyright Robert Bosch GmbH. All rights reserved, also regarding any disposal, exploration, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



Data flow and Data types

Copyright Robert Bosch GmbH. All rights reserved, also regarding any disposal, exploration, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.



How to embed a Python runtime (an add extension functions): https://docs.python.org/3/extending/

Copyright Robert Bosch GmbH. All rights reserved, also regarding any disposal, exploration, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

1 Python functions

- Error handling is done via exceptions.
- Names arguments may be used out of order. All names arguments can be used as unnamed arguments (then the order is relevant).

Copyright Robert Bosch GmbH. All rights reserved, also regarding any disposal, exploration, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

2 Python datalayer functions

(nodes) = Datalayer.browse(<path>)

- <path> - string, datalayer path to browse
- (nodes) - Tuple of node names

<value> = Datalayer.read(<path>)

- <path> - string, datalayer path to read
- <value> - value of the node (only simple values supported)

Datalayer.write(<path>, <value>)

- <path> - string, datalayer path to write
- <value> - value to write (only simple values supported)

Datalayer.create(<path>, <value>)

- <path> - string, datalayer path to create
- <value> - value of the node (only simple values supported)

Datalayer.remove(<path>)

- <path> - string, datalayer path to remove

<value> = Datalayer.readJSON(<path>)

- <path> - string, datalayer path to read
- <value> - value of the node as JSON string

Datalayer.writeJSON(<path>, <value>)

- <path> - string, datalayer path to write
- <value> - value to write JSON string

<result> = Datalayer.createJSON(<path>, <value>)

- <path> - string, datalayer path to create
- <value> - value of the node JSON string
- <result> - result of the create operation (when successful) as JSON string

Copyright Robert Bosch GmbH. All rights reserved, also regarding any disposal, exploration, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

3 Python motion functions

3.1 commands

<cmdID> = MotionLib.axsCmdPosAbs(axs=<axsName>, pos=<targetPos> [,vel=<velocity>] [,acc=<acceleration>] [,dec=<deceleration>] [,jrkAcc=<jerkAcceleration>] [,jrkDec=<jerkDecelerationn>] [,buffered=<buffered>])

Create a posAbs command for a single axis. (object have to be attached)

- <axsName> - string, name of the axis object
- <pos> - double, target position of the command
- <velocity> - double, velocity limit for the command (use the last programmed velocity limit of this axes, when not programmed)
- <acceleration> - double, acceleration limit for the command (use the last programmed acceleration limit of this axes, when not programmed)
- <deceleration> - double, acceleration limit for the command (use the last programmed acceleration limit of this axes, when not programmed)
- <jerkAcceleration> - double, jerk limit in acceleration for the command (use the last programmed jerk limit in acceleration of this axes, when not programmed)
- <jerkDeceleration> - double, jerk limit in deceleration for the command (use the last programmed jerk limit in deceleration of this axes, when not programmed)
- <buffered> - bool, set if the command should be buffered (default is true)

<cmdID> = MotionLib.axsCmdPosAdd(axs=<axsName>, pos=<targetPos> [,vel=<velocity>] [,acc=<acceleration>] [,dec=<deceleration>] [,jrkAcc=<jerkAcceleration>] [,jrkDec=<jerkDecelerationn>] [,buffered=<buffered>])

Create a posAdd command for a single axis. (object have to be attached)

- <axsName> - string, name of the axis object
- <pos> - double, target position of the command
- <velocity> - double, velocity limit for the command (use the last programmed velocity limit of this axes, when not programmed)
- <acceleration> - double, acceleration limit for the command (use the last programmed acceleration limit of this axes, when not programmed)
- <deceleration> - double, deceleration limit for the command (use the last programmed deceleration limit of this axes, when not programmed)
- <jerkAcceleration> - double, jerk limit in acceleration for the command (use the last programmed jerk limit in acceleration of this axes, when not programmed)
- <jerkDeceleration> - double, jerk limit in deceleration for the command (use the last programmed jerk limit in deceleration of this axes, when not programmed)
- <buffered> - bool, set if the command should be buffered (default is true)

<cmdID> = MotionLib.axsCmdPosRel(axs=<axsName>, pos=<targetPos> [,vel=<velocity>] [,acc=<acceleration>] [,dec=<deceleration>] [,jrkAcc=<jerkAcceleration>] [,jrkDec=<jerkDecelerationn>] [,buffered=<buffered>])

Create a posRel command for a single axis. (object have to be attached)

- <axsName> - string, name of the axis object
- <pos> - double, target position of the command
- <velocity> - double, velocity limit for the command (use the last programmed velocity limit of this axes, when not programmed)
- <acceleration> - double, acceleration limit for the command (use the last programmed acceleration limit of this axes, when not programmed)

Copyright Robert Bosch GmbH. All rights reserved, also regarding any disposal, exploration, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

- <deceleration> - double, deceleration limit for the command (use the last programmed deceleration limit of this axes, when not programmed)
- <jerkAcceleration> - double, jerk limit in acceleration for the command (use the last programmed jerk limit in acceleration of this axes, when not programmed)
- <jerkDeceleration> - double, jerk limit in deceleration for the command (use the last programmed jerk limit in deceleration of this axes, when not programmed)
- <buffered> - bool, set if the command should be buffered (default is true)

<cmdID> = MotionLib.axsCmdJogIncr(axs=<axsName>, dir=<direction>, incr=<increment> [,vel=<velocity>] [,acc=<acceleration>] [,dec=<deceleration>] [,jrkAcc=<jerkAcceleration>] [,jrkDec=<jerkDeceleration>])

Create a jog increment command for a single axis. (object have to be attached)

- <axsName> - string, name of the axis object
- <dir> - string, '+' or '-' direction of jogging
- <incr> - double, increment to move
- <velocity> - double, velocity limit for the command (use the last programmed velocity limit of this axes, when not programmed)
- <acceleration> - double, acceleration limit for the command (use the last programmed acceleration limit of this axes, when not programmed)
- <deceleration> - double, deceleration limit for the command (use the last programmed deceleration limit of this axes, when not programmed)
- <jerkAcceleration> - double, jerk limit in acceleration for the command (use the last programmed jerk limit in acceleration of this axes, when not programmed)
- <jerkDeceleration> - double, jerk limit in deceleration for the command (use the last programmed jerk limit in deceleration of this axes, when not programmed)

<cmdID> = MotionLib.axsCmdJogCont(axs=<axsName>, dir=<direction>, [,vel=<velocity>] [,acc=<acceleration>] [,dec=<deceleration>] [,jrkAcc=<jerkAcceleration>] [,jrkDec=<jerkDeceleration>])

Create a jog continuous command for a single axis. (object have to be attached)

- <axsName> - string, name of the axis object
- <dir> - string, '+' or '-' direction of jogging
- <velocity> - double, velocity limit for the command (use the last programmed velocity limit of this axes, when not programmed)
- <acceleration> - double, acceleration limit for the command (use the last programmed acceleration limit of this axes, when not programmed)
- <deceleration> - double, deceleration limit for the command (use the last programmed deceleration limit of this axes, when not programmed)
- <jerkAcceleration> - double, jerk limit in acceleration for the command (use the last programmed jerk limit in acceleration of this axes, when not programmed)
- <jerkDeceleration> - double, jerk limit in deceleration for the command (use the last programmed jerk limit in deceleration of this axes, when not programmed)

<cmdID> = MotionLib.axsCmdPower(axs=<axsName> [,switchOn=<switchOn>])

Create a power command for a single axis. (object have to be attached)

- <axsName> - string, name of the axis object
- <switchOn> - bool, set to true to switch power on (default is true)

<cmdID> = MotionLib.axsCmdAbort(axs=<axsName> [,dec=<deceleration>] [,jrkDec=<jerkDeceleration>])

Copyright Robert Bosch GmbH. All rights reserved, also regarding any disposal, exploration, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

Create a abort command for a single axis. (object have to be attached)

- <axsName> - string, name of the axis object
- <deceleration> - double, acceleration limit for the command (use the last programmed acceleration limit of this axes, when not programmed)
- <jerkDeceleration> - double, jerk limit in deceleration for the command (use the last programmed jerk limit in deceleration of this axes, when not programmed)

```
<cmdID> = MotionLib.axsCmdReset( axs=<axsName> )
```

Create a reset command for a single axis. (object have to be attached)

- <axsName> - string, name of the axis object

```
<cmdID> = MotionLib.axsCmdAddToKin( axs=<axsName>, kin=<kinName> [,buffered=<buffered>]  )
```

Create an addToKin command for a single axis. (object have to be attached)

- <axsName> - string, name of the axis object
- <kinName> - string, name of the kinematics object
- <buffered> - bool, set if the command should be buffered (default is true)

```
<cmdID> = MotionLib.axsCmdRemoveFromKin( axs=<axsName> )
```

Create a removeFromKin command for a single axis. (object have to be attached)

- <axsName> - string, name of the axis object

```
<cmdID> = MotionLib.kinCmdMoveLinAbs( kin=<kinName>, pos=<targetPos> [,vel=<velocity>] [,a
cc=<acceleration>] [,dec=<deceleration>] [,jrkAcc=<jerkAcceleration>] [,jrkDec=<jerkDecele
ration>] )
```

Create a moveLinAbs command for a kinematics. (object have to be attached)

- <kinName> - string, name of the kinematics object
- <pos> - tuple of double or list of double (max. 16 values), target position of the command
- <velocity> - double, velocity limit for the command (use the last programmed velocity limit of this kinematics, when not programmed)
- <acceleration> - double, acceleration limit for the command (use the last programmed acceleration limit of this kinematics, when not programmed)
- <deceleration> - double, deceleration limit for the command (use the last programmed deceleration limit of this kinematics, when not programmed)
- <jerkAcceleration> - double, jerk limit in acceleration for the command (use the last programmed jerk limit in acceleration of this kinematics, when not programmed)
- <jerkDeceleration> - double, jerk limit in deceleration for the command (use the last programmed jerk limit in deceleration of this kinematics, when not programmed)

```
<cmdID> = MotionLib.kinCmdMoveLinRel( kin=<kinName>, pos=<targetPos> [,vel=<velocity>] [,a
cc=<acceleration>] [,dec=<deceleration>] [,jrkAcc=<jerkAcceleration>] [,jrkDec=<jerkDecele
ration>] )
```

Create a moveLinRel command for a kinematics. (object have to be attached)

- <kinName> - string, name of the kinematics object
- <pos> - tuple of double or list of double (max. 16 values), target position of the command
- <velocity> - double, velocity limit for the command (use the last programmed velocity limit of this kinematics, when not programmed)

Copyright Robert Bosch GmbH. All rights reserved, also regarding any disposal, exploration, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

- <acceleration> - double, acceleration limit for the command (use the last programmed acceleration limit of this kinematics, when not programmed)
- <deceleration> - double, deceleration limit for the command (use the last programmed deceleration limit of this kinematics, when not programmed)
- <jerkAcceleration> - double, jerk limit in acceleration for the command (use the last programmed jerk limit in acceleration of this kinematics, when not programmed)
- <jerkDeceleration> - double, jerk limit in deceleration for the command (use the last programmed jerk limit in deceleration of this kinematics, when not programmed)

<cmdID> = MotionLib.kinCmdJogIncr(kin=<kinName>, dir=<direction>, incr=<increment> [,vel=<velocity>] [,acc=<acceleration>] [,dec=<deceleration>] [,jrkAcc=<jerkAcceleration>] [,jrkDec=<jerkDeceleration>])

Create a jog increment command for a kinematics. (object have to be attached)

- <kinName> - string, name of the kinematics object
- <dir> - tuple of double or list of double (max. 16 values), direction of jogging
- <incr> - double, increment to move
- <velocity> - double, velocity limit for the command (use the last programmed velocity limit of this kinematics, when not programmed)
- <acceleration> - double, acceleration limit for the command (use the last programmed acceleration limit of this kinematics, when not programmed)
- <deceleration> - double, deceleration limit for the command (use the last programmed deceleration limit of this kinematics, when not programmed)
- <jerkAcceleration> - double, jerk limit in acceleration for the command (use the last programmed jerk limit in acceleration of this kinematics, when not programmed)
- <jerkDeceleration> - double, jerk limit in deceleration for the command (use the last programmed jerk limit in deceleration of this kinematics, when not programmed)

<cmdID> = MotionLib.kinCmdJogCont(kin=<kinName>, dir=<direction>, [,vel=<velocity>] [,acc=<acceleration>] [,dec=<deceleration>] [,jrkAcc=<jerkAcceleration>] [,jrkDec=<jerkDeceleration>])

Create a jog continuous command for a kinematics. (object have to be attached)

- <kinName> - string, name of the kinematics object
- <dir> - tuple of double or list of double (max. 16 values), direction of jogging
- <velocity> - double, velocity limit for the command (use the last programmed velocity limit of this kinematics, when not programmed)
- <acceleration> - double, acceleration limit for the command (use the last programmed acceleration limit of this kinematics, when not programmed)
- <deceleration> - double, deceleration limit for the command (use the last programmed deceleration limit of this kinematics, when not programmed)
- <jerkAcceleration> - double, jerk limit in acceleration for the command (use the last programmed jerk limit in acceleration of this kinematics, when not programmed)
- <jerkDeceleration> - double, jerk limit in deceleration for the command (use the last programmed jerk limit in deceleration of this kinematics, when not programmed)

<cmdID> = MotionLib.kinCmdEnable(kin=<kinName>)

Create a kinematics enable command. (object have to be attached)

- <kinName> - string, name of the kinematics

<cmdID> = MotionLib.kinCmdDisable(kin=<kinName>)

Copyright Robert Bosch GmbH. All rights reserved, also regarding any disposal, exploration, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

Create a kinematics disable command. (object have to be attached)

- <kinName> - string, name of the kinematics

<cmdID> = MotionLib.kinCmdAbort(kin=<kinName>)

Create a abort command for a kinematics. (object have to be attached)

- <kinName> - string, name of the kinematics object

<cmdID> = MotionLib.kinCmdReset(kin=<kinName>)

Create a reset command for a kinematics. (object have to be attached)

- <kinName> - string, name of the kinematics object

MotionLib.kinCmdBlend(kin=<kinName> , d1=<D1>, d2=<D2>)

Create a blending command option for a kinematics. (object have to be attached)

- <kinName> - string, name of the kinematics object
- <D1> - Blending length on the first move command
- <D2> - Blending length on the second move command

MotionLib.kinCmdBlendP(kin=<kinName> [,d1=<D1>] [,d2=<D2>])

Create a permanent blending command option (or switch it off) for a kinematics. (object have to be attached)

When you don't program d1 and d2, then the permanent command option is switched off.

- <kinName> - string, name of the kinematics object
- <D1> - Blending length on the first move command
- <D2> - Blending length on the second move command

<cmdID> = MotionLib.setOverride(obj=<objName>, val=<Value>)

Set the override value for the motion object. (object does not have to be attached)

The command works at once, getCmdState() with the returned command ID will always return DONE.

The value have to be a double value in [0; 1]. A value of 0 means ‘zero override’, that causes the motion object to stop. A value of 1 means ‘100% override’, so the motion object moves with the commanded velocity.

- <objName> - string, name of the motion object
- <Value> - new override value

<cmdID> = MotionLib.setErrLevel(<objName>, <errLvl>)

Set the error level of a motion object. (object does not have to be attached)

The command works at once, getCmdState() with the returned command ID will always return DONE.

The set the error level back to NONE, you have to create a reset command.

- <objName> - string, name of the motion object
- <errLvl> - string, error level to set ('LIGHT', 'SEVERE', 'CRITICAL')

Copyright Robert Bosch GmbH. All rights reserved, also regarding any disposal, exploration, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

3.2 states

<ipoValues> = MotionLib.getAxsIpoValues(<axsName>)

Get the interpolated values of a single axis.

- <axsName> - string, name of the axis object
- <ipoValues> - dictionary with 'pos', 'vel', 'acc', 'jrk' with double values

<actValues> = MotionLib.getAxsActValues(<axsName>)

Get the actual values of a single axis.

- <axsName> - string, name of the axis object
- <actValues> - dictionary with 'pos', 'distLeft', 'vel', 'acc', 'torque' with double values

<state> = MotionLib.getAxsPLCOpenState(<axsName>)

Get the PLC Open state of a single axis.

- <axsName> - string, name of the axis object
- <state> - string, current PLC Open state

<ipoValues> = MotionLib.getKinIpoValues(<kinName>)

Get the interpolated values of a kinematics.

- <kinName> - string, name of the kinematics object
- <ipoValues> - dictionary with 'pos', 'vel', 'acc', 'jrk' with double values

<state> = MotionLib.getKinPLCOpenState(<kinName>)

Get the PLC Open state of a kinematics.

- <kinName> - string, name of the kinematics object
- <state> - string, current PLC Open state

<state> = MotionLib.getCmdState(obj=<objName>, id=<cmdID>)

Get the state of a command with it's command ID.

- <objName> - string, name of the motion object
- <cmdID> - uint64, command id
- <state> - string, current state ('CREATED', 'PREPARED', 'ACTIVE', 'DONE')

<Value> = MotionLib.getOverride(obj=<objName>)

Get the current override value of the motion object.

- <objName> - string, name of the motion object
- <Value> - the current override value of the motion object

3.3 misc

MotionLib.attachObj(<objNames>)

Attach one or more motion objects to this runner instance.

Copyright Robert Bosch GmbH. All rights reserved, also regarding any disposal, exploration, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

- <objNames> - strings, names of the motion objects, comma separated

`MotionLib.detachObj( <objNames> )`

Detach one or more motion objects to this runner instance.

- <objNames> - strings, names of the motion objects, comma separated

Copyright Robert Bosch GmbH. All rights reserved, also regarding any disposal, exploration, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.