ROBOTICS

Application manual

SafeMove

Trace back information:
Workspace 20A version a11
Checked in 2020-03-11
Skribenta version 5.3.033

Application manual

SafeMove

RobotWare 7.0.2

Document ID: 3HAC066559-001

Revision: A

© Copyright 2020 ABB. All rights reserved.
Specifications subject to change without notice.

The information in this manual is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this manual.

Except as may be expressly stated anywhere in this manual, nothing herein shall be construed as any kind of guarantee or warranty by ABB for losses, damages to persons or property, fitness for a specific purpose or the like.

In no event shall ABB be liable for incidental or consequential damages arising from use of this manual and products described herein.

This manual and parts thereof must not be reproduced or copied without ABB's written permission.

Keep for future reference.

Additional copies of this manual may be obtained from ABB.

Original instructions.

© Copyright 2020 ABB. All rights reserved.
Specifications subject to change without notice.

Table of contents

Overview of this manual 9
Product documentation 11
Safety 13

1 Introduction 15

1.1 Overview of functional safety 15
1.2 Safety 18
1.2.1 Safety aspects for the safety module and SafeMove 18
1.2.2 Standards conformance 19
1.2.3 Specific safety requirements 20
1.2.4 Safe design of SafeMove 21
1.2.5 Certifications 23
1.2.6 Conclusion 24
1.3 Limitations 25
1.4 Terminology 28
1.5 Abbreviations and acronyms 29

2 SafeMove functions 31

2.1 Overview of SafeMove functions 31
2.2 General functions 33
2.2.1 Manual Operation Supervision 33
2.2.2 Contact Application Tolerance (CAP) 34
2.3 Synchronization functions 36
2.3.1 Software synchronization 36
2.3.2 Hardware synchronization 37
2.4 Supporting functions 39
2.4.1 Safe Brake Ramp 39
2.5 Supervision functions 40
2.5.1 Stand Still Supervision (SST) 40
2.5.2 Axis Speed Supervision (ASP) 42
2.5.3 Tool Speed Supervision (TSP) 43
2.5.4 Axis Position Supervision (APO) 45
2.5.5 Tool Position Supervision (TPO) 47
2.5.6 Tool Orientation Supervision (TOR) 48
2.5.7 Control Error Supervision 49

3 Installation 51

3.1 Hardware 51
3.1.1 Safety module 51
3.2 Software 52

4 The Visual SafeMove user interface in RobotStudio 53

4.1 About Visual SafeMove 53
4.2 The user interface 54
4.3 The **Visual SafeMove** tab 57
4.3.1 About the **Visual SafeMove** tab 57
4.3.2 Adding SafeMove functions 60
4.4 The **Modify** tab 65
4.5 The **Visual SafeMove** browser 68
4.6 The **Safe IO Configurator** 72
4.6.1 Introduction 72
4.6.2 Signals view 73
4.6.3 Function mappings 75
4.6.4 Pre-logic and post-logic 77
4.6.4.1 Description of the complex operators 80

© Copyright 2020 ABB. All rights reserved.

4.6.5 Safe I/O system rules and limitations 85
4.7 Simulating SafeMove 86
4.8 Protected basic configuration 87

5 The FlexPendant user interface 89

5.1 The **Safety Controller** control panel 89

6 Configuring SafeMove 93

6.1 Recommended working procedure 93
6.2 Preparations 95
6.3 Configure system parameters 96
6.4 Set up safety user grants 97
6.5 Starting Visual SafeMove 99
6.6 Configure Manual Operation Supervision 100
6.7 Configure the robot 101
6.8 Configure additional axes 103
6.9 Configure the synchronization position 104
6.10 Configure the tools 105
6.11 Configure safe I/O 108
6.12 Configure the zones 110
6.13 Configure the ranges 113
6.14 Global supervision functions 116
6.15 Configure the supervision functions 117
6.16 Configure other functions 127
6.17 Save the configuration 129
6.18 Load the configuration to the safety controller 130
6.18.1 Loading a safety configuration using RobotStudio 130
6.19 Validate the configuration 131
6.20 Restore configuration 143
6.21 Reset safety controller to factory settings 144
6.22 Upgrading and installing new systems 145

7 Running in production 147

7.1 Reaction time 147
7.2 Restarting the controller 148
7.3 Recovery after safety violation 149
7.4 Changes to robot or robot cell 150

8 Maintenance 151

8.1 Required maintenance activities 151

9 RAPID components 153

10 Reference information 155

10.1 Synchronization guidelines 155
10.1.1 Software synchronization guidelines 155
10.1.2 Hardware synchronization guidelines 157
10.2 Cyclic Brake Check guidelines 158
10.2.1 Cyclic Brake Check signal description 161
10.3 Servo Delay Factor and Servo Lag 165
10.4 Connection of external emergency stop 167
10.5 SafeMove geometry configuration file 169
10.5.1 Introduction 169
10.5.2 Use cases 171
10.5.3 Explanation of the configuration file 173
10.5.3.1 Introduction 173
10.5.3.2 Drive module configuration 175
10.5.3.3 Robot data 176

© Copyright 2020 ABB. All rights reserved.

10.5.3.4 External axis 178
10.5.3.5 Tool 179
10.5.3.6 Safe zones 182
10.5.4 Example 184

Index **187**

© Copyright 2020 ABB. All rights reserved.

This page is intentionally left blank

Overview of this manual

About this manual

This manual describes the safety module, the functional safety options, and the second generation of SafeMove. It contains descriptions of the functionality, and how to configure that functionality. It also describes user interfaces and recommendations on how to use the safety module.

Usage

This manual should be used during installation and configuration of the safety module, SafeMove, and the functional safety options.

Who should read this manual?

This manual is mainly intended for:

- personnel that are responsible for installations and configurations of hardware/software
- personnel that make configurations of the I/O system
- system integrators

Prerequisites

The reader should have the required knowledge of:

- mechanical installation work
- electrical installation work
- working with industrial robots
- using RobotStudio
- personal safety, see *Safety manual for robot - Manipulator and IRC5 or OmniCore controller*.

References

Reference	Document ID
Application manual - PROFINET Controller/Device	*3HAC066558-001*
Application manual - EtherNet/IP Scanner/Adapter	*3HAC066565-001*
Operating manual - RobotStudio	*3HAC032104-001*
Safety manual for robot - Manipulator and IRC5 or OmniCore controller	*3HAC031045-001*
Product manual - OmniCore C30	*3HAC060860-001*
Technical reference manual - RAPID Instructions, Functions and Data types	*3HAC065038-001*
Product specification - Robot stopping distances according to ISO 10218-1	*3HAC048645-001*
Technical reference manual - System parameters	*3HAC065041-001*

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Continued

Revisions

Revision	Description
A	Released with RobotWare 7.0.2.

© Copyright 2020 ABB. All rights reserved.

Product documentation

Categories for user documentation from ABB Robotics

The user documentation from ABB Robotics is divided into a number of categories. This listing is based on the type of information in the documents, regardless of whether the products are standard or optional.

All documents can be found via myABB Business Portal, *www.myportal.abb.com*.

Product manuals

Manipulators, controllers, DressPack/SpotPack, and most other hardware is delivered with a **Product manual** that generally contains:

- Safety information.
- Installation and commissioning (descriptions of mechanical installation or electrical connections).
- Maintenance (descriptions of all required preventive maintenance procedures including intervals and expected life time of parts).
- Repair (descriptions of all recommended repair procedures including spare parts).
- Calibration.
- Decommissioning.
- Reference information (safety standards, unit conversions, screw joints, lists of tools).
- Spare parts list with corresponding figures (or references to separate spare parts lists).
- References to circuit diagrams.

Technical reference manuals

The technical reference manuals describe reference information for robotics products, for example lubrication, the RAPID language, and system parameters.

Application manuals

Specific applications (for example software or hardware options) are described in **Application manuals**. An application manual can describe one or several applications.

An application manual generally contains information about:

- The purpose of the application (what it does and when it is useful).
- What is included (for example cables, I/O boards, RAPID instructions, system parameters, software).
- How to install included or required hardware.
- How to use the application.
- Examples of how to use the application.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Continued

Operating manuals

The operating manuals describe hands-on handling of the products. The manuals are aimed at those having first-hand operational contact with the product, that is production cell operators, programmers, and troubleshooters.

© Copyright 2020 ABB. All rights reserved.

Safety

Safety regulations

Before beginning mechanical and/or electrical installations, ensure you are familiar with the safety information in the product manuals for the robot.

The integrator of the robot system is responsible for the safety of the robot system.

© Copyright 2020 ABB. All rights reserved.

This page is intentionally left blank

1 Introduction

1.1 Overview of functional safety

Purpose

The purpose of the safety module and the functional safety options is to provide a robust and easy-to-use safety controller in the robot system. Functional safety includes a complete software and hardware solution that is fully integrated with the robot controller and the RobotStudio programming environment.

SafeMove is the main functional safety option, but the safety module can also be used in various applications without the SafeMove option. For example to communicate with a safety PLC through safe fieldbus communication, or when using the keyless mode selector.

SafeMove functions

When using SafeMove, the safety controller ensures a high safety level in the robot system by using supervision functions that can stop the robot. Note that the safety module and the functional safety options is one component in the safety system of a complete robot cell, normally complemented by other equipment (for example light barriers) for detecting the whereabouts of the operator.

Function	Description
Tool supervision functions	Protects the operator and enhances machine and equipment safety by supervising the position (*Tool Position Supervision*), speed (*Tool Speed Supervision*) and orientation (*Tool Orientation Supervision*) of the tool.
Axis supervision functions	Protects the surroundings by supervising the axis position (*Axis Position Supervision*) and the axis speed (*Axis Speed Supervision*).
Standstill supervision	Supervises the stand-still of robot axes without having to switch the robot to Motors Off. It enables operators to perform tasks in the immediate vicinity of the robot.
Contact application tolerance	Allows the robot to be in contact with the work-piece in limited areas. This can for example be used in applications where the robot is used for grinding or during tool change.
Cyclic brake check	Supervises that the brakes are checked with a cyclic interval.
Stop functions	Triggers stop of the robot using safe fieldbus inputs from the safety PLC.

Some examples of applications:

- Manual loading stations
- Manual workpiece inspection during operation
- Optimization of cell size
- Protection of sensitive equipment
- Ensuring safe orientation of emitting processes

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Functional safety options

The safety controller is included as standard in the robot controller. To use any of the safety functionality, it is necessary to have the corresponding option:

- [3043-1] *SafeMove Basic*
- [3043-2] *SafeMove Pro*
- [3044-2] *2 modes Keyless*

Visual SafeMove

The safety module option gives you access to the Visual SafeMove configurator in RobotStudio. With Visual SafeMove you can:

- configure and visualize supervision functions in a 3D environment.
- configure stop functions, such as automatic stop.
- configure Cyclic Brake Check.
- configure safe signals (safe Ethernet communication and I/Os)
- configure signal logics.
- configure system status outputs.

SafeMove Basic and SafeMove Pro

The below table lists the differences between SafeMove Basic and SafeMove Pro.

Function	SafeMove Pro	SafeMove Basic
Supported number of axes	9	9
Safe ranges	8	8
Safe zones	16	1
Tool changer support	Yes (16 tools)	-
Axis Position Supervision	Yes	Yes
Axis Speed Supervision	Yes	-
Tool Orientation Supervision	Yes	-
Tool Position Supervision	Yes	1
Tool Speed Supervision	Yes	-
Stand Still Supervision	Yes	-
Contact application support	Yes	Yes

Basic approach

This is the general approach for setting up the safety module and SafeMove.

1. Connect the safety controller to other safety hardware and configure the safe I/O connections.
2. Configure the settings for the SafeMove functions via Visual SafeMove.
3. Download the configuration to the the safety controller. Restart the controller.
4. Synchronize the safety controller.
5. Make sure the activation input signals are activating the desired supervision functions.
6. Validate the configuration.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

7 Lock the configuration.

For more detailed instructions, see sections *Installation on page 51* and *Configuring SafeMove on page 93*.

Requirements

Robust supervision functionality in SafeMove requires correct settings of payload and additional axes, since this will affect the calculated accepted servo lag. Please also note that external forces applied on the manipulator can cause a negative influence on the supervision functions, since the servo lag might differ from the calculated values, due to such external forces.

DANGER

A SafeMove configuration must always be validated to verify that the desired safety is achieved. If no validation is performed, or the validation is inadequate, the configuration cannot be relied on for personal safety.

The validation must also consider that the braking starts after a zone is violated, so additional stopping distances may be required, which depend on many factors, for example mass and speed.

© Copyright 2020 ABB. All rights reserved.

1.2 Safety

1.2.1 Safety aspects for the safety module and SafeMove

Overview

The safety module is an integrated safety controller in the robot controller, with the purpose of providing safety functionality for the robot. Safe output and input signals are typically connected to cell safety circuitry by safe communication with a safety PLC. The safety PLC can take care of interlocking in the robot cell, for example, in order to prevent robot and operator to enter a the same area at the same time.

In this chapter we describe how the safety module and SafeMove comply with relevant safety standards and regulations.

Note

The safety module and SafeMove is only a part of the robot system, it is the responsibility of the user to do a risk assessment of the robot system. It is also the responsibility of the user of SafeMove to ensure that the robot system is designed and installed in accordance with the safety requirements set forth in the standards and regulations of the country where the robot system is installed.

© Copyright 2020 ABB. All rights reserved.

1.2.2 Standards conformance

Standards

The safety module and SafeMove has been designed to fulfill applicable parts of the following standards.

- EN ISO 12100:2010 Safety of machinery - General principles for design - Risk assessment and risk reduction
- EN 60204-1:2006/A1:2009 Safety of machinery - Electrical equipment of machines - Part 1: General requirements
- EN ISO 10218-1:2011, Robots for industrial environments - Safety requirements - Part 1: Robot
- EN 61000-6-2:2005 EMC, Generic immunity
- EN 61000-6-4:2007/A1:2011 EMC, Generic emission
- EN ISO 13849-1:2015 Safety of machinery - Electrical equipment of machines - Part 1: General requirements

  EN ISO 13849-2:2012 Safety of machinery - Safety-related parts of control systems - Part 2: Validation

© Copyright 2020 ABB. All rights reserved.

1.2.3 Specific safety requirements

Specific safety requirements for SafeMove

SafeMove complies with EN ISO 10218-1 in general and specifically complies with chapter 5.4.2, that is, the following requirements.

When safety related control systems are required, the safety related parts shall be designed so that:

- A single fault in any of these parts shall not lead to the loss of the safety function.
- Whenever reasonably practicable, the single fault shall be detected at or before the next demand upon the safety function.
- When the single fault occurs, the safety function is always performed and a safe state shall be maintained until the detected fault is corrected.
- All reasonably foreseeable faults shall be detected.

This requirement is considered to be equivalent to structure category 3 as described in ISO 13849-1. Category 3 is normally fulfilled by redundant circuits, such as dual channels, which is the case for SafeMove. SafeMove together with the safety module and the robot controller also complies with performance level (PL) "d" according to ISO 13849-1. This safety level is equivalent to SIL 2 as defined in IEC 61508.

© Copyright 2020 ABB. All rights reserved.

1.2.4 Safe design of SafeMove

Overview

SafeMove has two important types of supervision functionality.

The first one being to ensure that the axis computer and the drive system are working correctly, making the robot follow the ordered value from the main computer as expected.

The second being to supervise the robot position and speed and stopping the robot or setting output signals to indicate a hazard.

Supervision of axis computer and drive system

The main computer calculates the absolute motor position values sent as reference to the axis computer, and simultaneously sends them to the safety controller. The axis computer reports the actual rotational motor position values via the main computer to the safety controller, as a separate process from the reference value. Since these values are within one revolution, the absolute position is calculated by adding values from internal revolution counters in both the axis computer and in SafeMove.

By comparing the ordered motor position and the actual motor position, SafeMove can detect any difference (outside a permitted lag deviation) between the two positions, thereby ensuring that the drive system is working properly according to the first supervision function as described above.

It is important to ensure that the safety controller and the robot controller are synchronized. The safe sync position is defined during configuration and stored in the safety controller. In this position, SafeMove will calculate the robot joint positions and check against a stored value to confirm that the synchronization is correct, covering the following points:

- SafeMove is working correctly with the right revolution counter value.
- The right manipulator is used.
- The calibration value is correct.
- The SMB is working correctly.

Category 3 supervision

The supervision complies with category 3, that is, two separate channels shall always give the same result. One channel consists of the axis computer with the drive system, motors, resolvers, and measurement system. The second channel consists of the ordered value from the main computer. These channels are compared using the SafeMove evaluation circuits, which in itself is dual channel.

Additional safety design

Additional safety, over and above what is formally required, is brought to the concept by the inherent dual channel character of the resolver, thanks to its dual sine and cosine output, where the square sum is supervised to be close to 1.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

1.2.4 Safe design of SafeMove
Continued

Supervision of robot position and speed

The second type of supervision functionality (to supervise the robot position and speed) is fulfilled by letting SafeMove compare the robot position and speed with limit values configured by an authorized user (so called Safety User). If any value is outside its defined safe area, the supervision functions will stop the robot (or set an output signal).

To ensure that also this supervision complies with the category 3 requirement, SafeMove is internally working with a two channel microprocessor based system. Both processors make parallel calculations comparing the actual position and the reference position.

© Copyright 2020 ABB. All rights reserved.

1.2.5 Certifications

Overview

The safety module and SafeMove has been certified by external organizations as described below.

Certifications by RISE Research Institutes of Sweden

RISE Research Institutes of Sweden has made an assessment of safety controller DSQC1015, and included software, according to EN ISO 13849-1:2015 and EN ISO 13849-2:2012 and issued an EC type-examination certificate with regard to 2006/42/EC, Annex V, item 4, as a logic unit.

RISE has also assessed that the software blocks, implementing safety functions as defined in ISO 10218-1:2011, are correctly implemented.

Certifications by UL

The safety controller DSQC1015, and included software, is approved by UL according to the following standards:

- UL 1740, Standard for Robots and Robotic Equipment
- ANSI/RIA R15.06, Industrial Robots and Robotic Systems
- CAN/CSA Z434-14, Industrial Robots and Robot Systems - General Safety Requirements
- CAN/CSA C22.2 No. 73-1953, Test of Electrically Operated Machine Tools

© Copyright 2020 ABB. All rights reserved.

1.2.6 Conclusion

Conclusion

As has been shown above and confirmed by third party certifications, the safety module and SafeMove fulfills all relevant current safety standards globally.

© Copyright 2020 ABB. All rights reserved.

1.3 Limitations

Robots supported by SafeMove

For RobotWare 7.0.2, the following robot families are supported by SafeMove:

- IRB 14050

Note

YuMi robots with SafeMove requires using the enabling device to enable the motors in manual mode. For more information, see:

Product manual - IRB 14050

Note

Other robot models are not supported, for example:

- SafeMove does not support parallel arm robots, such as the IRB 360.
- SafeMove does not support SCARA robots, such as the IRB 910.

WARNING

Even if a model is supported by SafeMove, each installed robot must be verified individually to ensure that no mechanical or other deviations exists which would make SafeMove position measurements incorrect.

This is normally done during safety function verification, see *Validate the configuration on page 131*.

Supported mounting angles

SafeMove supports any mounting angle. For example floor mounted, tilted, inverted, etcetera.

Supported tracks

SafeMove supports all ABB track motion units.

Supported positioners

SafeMove supports positioners that are single axis mechanical units. Positioners with several axes are treated as multiple single axes, for example two axes positioners will be treated by SafeMove as two single axes and can be monitored as such using axis supervision.

Positioners that are used with activation/deactivation feature, is not supported. The axes of the positioner must be active at all times.

Servo welding gun

SafeMove does not support supervision of servo welding guns. That is, the robot axes can be supervised, but not the axis of the servo welding gun.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Non ABB additional axes

Non ABB track motion units, non ABB positioners, and other additional axis may be supported by the SafeMove option but this needs to be verified case by case.

To verify if a non ABB additional axis can be used with SafeMove, tune the additional axis before configuring the SafeMove parameters. If a properly tuned and configured non ABB additional axis still generates error messages regarding servo lag, then it cannot be used with SafeMove. For more information about tuning an additional axis see .

Work area for additional axes

There are always the following upper and lower work area limitations for additional axes:

- Track unit length (arm side) max ± 448 m
- Rotating axis (arm side) max ± 25700 degrees or ± 448 radians

On the motor side there is also a limitation of ± 32000 revolutions.

Combined external axes and robot with gantry

SafeMove does not support supervision of combined external axes, such as a gantry.

SafeMove does not support supervision of robots moved by the gantry.

Stand alone controller

SafeMove supports *Stand alone controller* and drive modules without TCP-robot with up to six additional axes. The axes are handled as external single axis.

Only SafeMove functions working on axes are available. The functions are:

- Axis Position Supervision (APO)
- Axis Speed Supervision (ASP)
- Stand Still Supervision (SST)
- Safe Brake Ramp

All other functions are deactivated.

Tool changer

SafeMove Pro supports up to 16 different tools. All included tools must have their appropriate settings in the configuration file. The selection of tool must be supervised using a safe fieldbus.

Robot mounted on rotational axis

SafeMove does not support supervision of a robot mounted on a rotational axis. Axis monitoring will be possible but the TCP functions, like Tool Position Supervision, are not supported.

No deactivation

Additional axes that are used with activation/deactivation feature are not supported. If additional axes are to be used, they must also be active at all times.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Independent joint

Independent joint cannot be monitored by SafeMove.

Electronically linked motors

SafeMove supports supervision of additional axes using Electronically Linked Motors. Using this feature may cause servo lag problems, particularly if the follower axis is included in the safety configuration. It may be necessary to modify the tuning of the axes, or increase the tolerances in the SafeMove configuration. The performance should be verified case by case.

Shared drive modules

Drive units of supervised axes cannot be shared, for instance between positioner axes.

RAPID non motion execution

This test feature cannot fully be used together with the SafeMove option.

Responsive jogging

Category 1 stop is deactivated in manual mode when responsive jogging is active. This is because a robot that is stopped with a category 1 stop follows its programmed path while decelerating. When using responsive jogging there is no defined path available.

Note

Responsive jogging can be deactivated by changing the parameter *Jog Mode* from *Responsive* to *Standard*.

For more information about parameter *Jog Mode*, see type *Jog Parameters*, topic *Motion* in *Technical reference manual - System parameters*.

SoftMove

When SafeMove is used together with SoftMove there is a risk for servo lag problems. The recommended action is to add a Contact Application Tolerance (CAP) in the area where SoftMove is active.

For more information about SoftMove, see *Application manual - SoftMove*.

© Copyright 2020 ABB. All rights reserved.

1.4 Terminology

About these terms

Some words have a specific meaning when used in this manual. It is important to understand what is meant by these words. This manual's definitions of these words are listed below.

Term list

Term	Definition
Category 0 stop	Stop by immediate removal of power to the actuators. Mechanical brakes are applied. A robot that is stopped with a category 0 stop does not follow its programmed path while decelerating.
Category 1 stop	Controlled stop with power available to the actuators to achieve the stop. Power is removed from the actuators when the stop is achieved. A robot that is stopped with a category 1 stop follows its programmed path while decelerating.
Occupationally safe	Safe for a person to be in an area.
Operationally safe	Safe for the machinery but not safe for persons to enter the area.
Safe input	Dual monitored digital input.
Safe output	Dual monitored digital output.
Safety controller	A safety board used with the robot controller, handling SafeMove functionality.

© Copyright 2020 ABB. All rights reserved.

1.5 Abbreviations and acronyms

Overview

This section specifies typical abbreviations and acronyms used in this manual.

Abbreviatons/acronyms list

Abbreviation/acronym	Description
APO	Axis Position Supervision
ASP	Axis Speed Supervision
CAP	Contact Application Tolerance
SST	Stand Still Supervision
TOR	Tool Orientation Supervision
TPO	Tool Position Supervision
TSP	Tool Speed Supervision

© Copyright 2020 ABB. All rights reserved.

This page is intentionally left blank

2 SafeMove functions

2.1 Overview of SafeMove functions

Overview

The SafeMove functions can be divided into the following categories:

- General functions, see:
  - *Manual Operation Supervision on page 33*
  - *Contact Application Tolerance (CAP) on page 34*
- Synchronization functions, see:
  - *Software synchronization on page 36*
  - *Hardware synchronization on page 37*
- Supporting functions, for example verification of brakes, see:
  - *Safe Brake Ramp on page 39*
- Supervision functions, can stop the robot or set a safe output signal, see:
  - *Stand Still Supervision (SST) on page 40*
  - *Axis Speed Supervision (ASP) on page 42*
  - *Tool Speed Supervision (TSP) on page 43*
  - *Axis Position Supervision (APO) on page 45*
  - *Tool Position Supervision (TPO) on page 47*
  - *Tool Orientation Supervision (TOR) on page 48*
  - *Control Error Supervision on page 49*

About the supervision functions

Supervision functions can be activated and deactivated with safe input signals or be configured to be permanently active.

The supervision functions can stop the robot and additional axes, or set a safe output signal, if a violation occurs.

Signal	Status
Activation	The signal is set to 0 for activation.
Function active status	The signal is set to 1 when active.
Violation action • **Signal**	The signal is set to 0 at violation. **Note** When a signal is set to 0 at violation, it will remain 0 for at least 250 ms even after the violation has ended.

Local and global functions

There are two types of functions in SafeMove, local functions and global function.

Local functions are active when the robot is in defined parts of its working area, it could be inside a safe zone or within specified ranges. Those functions are used

Continues on next page

© Copyright 2020 ABB. All rights reserved.

for setting a speed limitation in specific areas (Tool Speed Supervision) or for protecting equipment being hit by the robot (Tool Position Supervision).

Global functions are general functions that are active regardless of the position of the robot. It could for example be a general speed limitation activated when the cell door is opened.

For more information, see *The Global Functions button on page 62*.

Combining functions

The supervision functions can be used separately or in a variety of combinations.

Exclude from configuration

In the implementation of the safety controller, some safety supervision functions are included in the system even if the SafeMove option is not selected. This includes Safe Brake Ramp, Manual Operation Supervision, and Control Error Supervision. The included safety supervision functions require synchronization of the safety controller and that the configuration is validated and locked.

In some cases it is necessary to exclude the safety settings from the configuration. For example when configuring a safe fieldbus on robots that are not supported by SafeMove. This is done with the setting *Exclude from configuration*. This setting can also be used to exclude supported robots, for example individual robots in a MultiMove application, and supported robots running only a safe fieldbus.

When excluding the safety supervision from the configuration, the robot behaves as if no safety module is installed. That means:

- Instead of Safe Brake Ramp a one second delay is used between the request of a category 1 stop and the completing category 0 stop.
- No safety supervision of manual reduced speed.
- No safety supervision on the position data received from the serial measurement board, SMB, and the axis computer, AXC, (resolver input and revolution counter).
- No need to synchronize and lock the configuration.

For information on how to exclude a robot from the safety configuration, see *Robot parameters node on page 68*.

DANGER

A SafeMove configuration must always be validated to verify that the desired safety is achieved. If no validation is performed, or the validation is inadequate, the configuration cannot be relied on for personal safety.

© Copyright 2020 ABB. All rights reserved.

2.2 General functions

2.2.1 Manual Operation Supervision

Manual Operation Supervision

Manual Operation Supervision is a function that is active in manual operating mode and supervises that motion is below the configured manual mode supervision speed.

Functionality

While Manual Operation Supervision is active, a supervision makes sure that the tool center point (TCP), wrist center point (WCP), and elbow speed does not exceed 250 mm/s (unless a lower value is configured).

Manual Operation Supervision overrides safety functions by muting stops from the safety controller due to supervision functions.

> **Note**
>
> If Manual Operation Supervision is active and the robot is jogged to a non-violation position and then into a supervision violation position again, the robot will stop again. The new violation must be confirmed by releasing the enabling device on the FlexPendant before the jogging can be resumed.

Settings

The following parameters can to be configured for Manual Operation Supervision:

- Max speed in manual mode.

See *Configure Manual Operation Supervision on page 100*.

Function activation

Manual Operation Supervision is activated by switching to manual mode.

Dependencies to other supervision functions

Manual Operation Supervision can be used in combination with all other SafeMove functions.

© Copyright 2020 ABB. All rights reserved.

2.2.2 Contact Application Tolerance (CAP)

Contact Application Tolerance

Contact Application Tolerance relaxes the supervision of the servo lag if either:

- all configured axes are within the corresponding safe axis range,
- the TCP is within the corresponding safe zone,
- the activation signal for the Contact Application Tolerance function is 0 (if used).

Functionality

Contact Application Tolerance relaxes the Control Error Supervision (servo lag) to a higher value if all configured axes are within the defined axis range, or the TCP is within the defined zone, and the activation signal is 0 (if used).

Contact Application Tolerance can be used, for instance, in machine tending, when the servo loop gain is reduced (soft servo), or during Force Control. It is also useful when external forces are applied to the robot, for example during tool change.

If the robot is within the defined range/zone, then the safety level is considered to be operationally safe rather than occupationally safe. That means it is not safe for personnel to be in the range/zone defined for Contact Application Tolerance.

For axis ranges, both reference value and measured value for all axes must be within the defined range to be able to activate the relaxed control error. For zones, both reference value and measured value for the TCP must be within the defined zone to be able to activate the relaxed control error.

Up to 9 axes can be supervised simultaneously.

WARNING

When the Contact Application Tolerance is active then the dual channel safety tolerance is reduced with the configured value. This must be considered in the design of the robot application.

Settings

The following settings can be configured for Contact Application Tolerance:

- An axis range or a zone to apply Contact Application Tolerance for.
- Permissible control error for each axis, in degrees or mm on arm side.
- Set an output signal if a violation occurs.
- Set a status signal when the function is active.

How to define these settings is described in *Configuring Contact Application Tolerance on page 124*.

Dependencies to other supervision functions

If Contact Application Tolerance is active, it overrides the Control Error Supervision function. That means that all other active safety controller functions work with relaxed Control Error Supervision.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Contact Application Tolerance can be used in combination with all other SafeMove functions.

Limitations

Contact Application Tolerance is not considered to be active if the run chain is open.

Related information

Control Error Supervision on page 49.

© Copyright 2020 ABB. All rights reserved.

2.3 Synchronization functions

2.3.1 Software synchronization

Software synchronization

Software synchronization is a function that makes sure that the safety controller has the correct information regarding robot position.

Unsynchronized state can, for example, occur:

- If one or more axes were moving during shutdown or power off.
- After a failed synchronization.

> **Note**
>
> On YuMi robots with SafeMove, the synchronization position must be defined because default synchronization position cannot be used.

Functionality

Software synchronization is initiated from the FlexPendant. How to execute a software synchronization is described in section *Performing a synchronization on page 155*.

If the synchronization attempt is unsuccessful, the synchronization procedure must be executed again until successful.

> **Note**
>
> The supervision functions can only be active while SafeMove is synchronized. When unsynchronized, only manual mode operation with reduced speed is possible until synchronization is executed successfully.

Settings

The following settings need to be configured for software synchronization:

- Status signal.
- Angles and positions of robot (and additional axes) at the synchronization position.

Dependencies to other supervision functions

Software synchronization is always available even if hardware synchronization is configured.

Related information

Configure the synchronization position on page 104

Software synchronization guidelines on page 155.

Recovery after safety violation on page 149.

© Copyright 2020 ABB. All rights reserved.

2.3.2 Hardware synchronization

Hardware synchronization

Hardware synchronization is a function that makes sure that the robot calibration is correct by using a physical synchronization switch.

Unsynchronized state can, for example, occur:

- If one or more axes were moving during shutdown or power off.
- After a failed synchronization.

Note

On YuMi robots with SafeMove, the synchronization position must be defined because default synchronization position cannot be used.

Functionality

The robot must move to a safe synchronization position to ensure that the safety controller and the robot controller are synchronized. The safe synchronization position is defined during configuration and stored in the safety controller.

The robot must move to the safe synchronization position and activate a switch. When the switch is activated, the safety controller assumes that the robot revolution counters are correct. It also calculates the arm position from the motor positions, the gear ratio, and its internal revolution counter. If the position matches the stored synchronization position within half a motor revolution, then the synchronization is assumed to be correct.

If the synchronization is correct, the safety controller then sends a message to the robot controller, confirming that the safety controller is synchronized to its mechanical units, and continues with its regular operation.

Note

The supervision functions can only be active while SafeMove is synchronized. When unsynchronized, only manual mode operation with reduced speed is possible until synchronization is executed successfully.

Settings

The following settings need to be configured for hardware synchronization:

- Synchronization signal.
- Angles and positions of robot (and additional axes) at the synchronization position.

Dependencies to other supervision functions

Software synchronization is always available even if hardware synchronization is configured.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

2.3.2 Hardware synchronization
Continued

Limitations

- The safe sync position must be within reach for the robot. It must not be a singularity, that is all six axis must have unique positions.

Related information

Configure the synchronization position on page 104

Hardware synchronization guidelines on page 157.

Recovery after safety violation on page 149.

© Copyright 2020 ABB. All rights reserved.

2.4 Supporting functions

2.4.1 Safe Brake Ramp

Safe Brake Ramp

Safe Brake Ramp is an active supervision function that supervises category 1 stops initiated by the safety controller.

Functionality

When a category 1 stop is triggered by the safety controller, the motors are used for a controlled deceleration along the planned motion path. Safe Brake Ramp supervises this deceleration. If the deceleration is too slow, a category 0 stop is triggered. After 1 second, a category 0 stop is always triggered regardless.

A category 1 stop usually stops faster than the margins for Safe Brake Ramp, so under normal circumstances Safe Brake Ramp does not trigger.

> **Note**
>
> Depending on the application, Safe Brake Ramp may trigger more often, for example for tilted robot or heavy load. This results in a category 0 stop.

Settings

For track motions and other additional axes, the parameters **Brake Ramp Limit** and **Ramp Delay** have to be set. The parameter **Start Speed Offset** is used for both manipulator and all additional axes.

Function activation

Safe Brake Ramp cannot be dynamically activated/deactivated. If it is configured to be active, it is always active.

Dependencies to other supervision functions

Safe Brake Ramp will be used in combination with all other SafeMove functions using category 1 stops.

Limitations

- Safe Brake Ramp only supervises category 1 stops initiated by the safety controller. Stops initiated elsewhere, e.g. by the robot controller, are not supervised.

Related information

Category 1 stop (see *Terminology on page 28*)

Category 0 stop (see *Terminology on page 28*)

Explanation of Safe Brake Ramp on page 71

© Copyright 2020 ABB. All rights reserved.

2.5 Supervision functions

2.5.1 Stand Still Supervision (SST)

Stand Still Supervision

Stand Still Supervision is an active supervision function ensuring that all supervised axes are standing still.

Functionality

Stand Still Supervision can supervise that a robot is standing still even if the servo and drive system are in regulation. If any supervised axis starts to move, Stand Still Supervision will cause a category 0 stop.

When Stand Still Supervision is active for all axes (including all additional axes), the operator is safe from harm related by robot motion when entering the working space of the robot.

8 different sets of up to 9 axes can be defined. When Stand Still Supervision is activated for a set, all axes in that set are supervised.

DANGER

Working under an axis affected by gravity which has no balancing may require a safety performance level (PL) "e", which is not provided by SafeMove. If this kind of work is intended, the risk must be added to the risk analysis of the installation and eliminated by other means (for example additional mechanical stops).

DANGER

It is not allowed to enter the working space of the robot if a Contact Application Tolerance function is active, even if a Stand Still Supervision function is active at the same time.

DANGER

For additional axes, a standstill reference tolerance must be configured.

WARNING

Safe Stand Still is not active in manual operating mode and shall therefore not be used to guarantee operator safety in that mode.

Note

If the robot tries to move due to an error during active Stand Still Supervision, SafeMove will detect this and initiate a stop. Since there is a certain reaction time involved a slight jerk may occur.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Settings

The following parameters can be configured for Stand Still Supervision:

- Assignment of safe inputs for activation of Stand Still Supervision.
- Which axes to supervise, with specified stand still measurement tolerance, for each stand still set.
- Set an output signal if a violation occurs.
- Set a status signal when the function is active.

See *Configuring Stand Still Supervision on page 125*.

Function activation

Stand Still Supervision is activated by a safe input signal, or is permanently active if only output signal and no stop is used.

Note

If SafeMove becomes unsynchronized, the robot will stop and the Stand Still Supervision function will be deactivated. A movement with reduced speed is possible.

Limitations

- Stand Still Supervision is only available for SafeMove Pro, see *Functional safety options on page 16*.

© Copyright 2020 ABB. All rights reserved.

2.5.2 Axis Speed Supervision (ASP)

Axis Speed Supervision

Axis Speed Supervision is an active supervision function that supervises the speed of robot axes and additional axes.

Functionality

Supervision of the speed for up to 9 axes (robot axes and additional axes). Up to 8 sets can be configured.

If any of the supervised axes is outside its allowed speed, the safety controller triggers. This violation will cause a category 0 stop, a category 1 stop, and set an output signal, depending on the configuration.

WARNING

Axis Speed Supervision is not active in manual operating mode and shall therefore not be used to guarantee operator safety in that mode.

Settings

The following parameters can be configured for Axis Speed Supervision:

- An optional axis range or a zone for which the Axis Speed Supervision is applied.
- Which axes to supervise.
- Maximum speed and minimum speed, defined per axis.
- Category 0 stop, category 1 stop, or no stop if a violation occurs.
- Set an output signal if a violation occurs.
- Set a status signal when the function is active.
- Assignment of safe input for activation of Axis Speed Supervision.

How to define these settings is described in *Configuring Axis Speed Supervision on page 119*.

Function activation

Axis Speed Supervision is activated by a safe input signal, or is permanently active.

Limitations

- Axis Speed Supervision is only available for SafeMove Pro, see *Functional safety options on page 16*.
- The highest speed limit that can be configured is 600 degrees/s for rotational axes and 10000 mm/s for linear axes.

© Copyright 2020 ABB. All rights reserved.

2.5.3 Tool Speed Supervision (TSP)

Tool Speed Supervision

Tool Speed Supervision is a supervision function that supervises the speed of the active safety tool, arm check point, and configured speed supervised points.

Functionality

Tool Speed Supervision supervises the linear speed (in mm/s) for:

- TCP for the active safety tool.
- Arm check point, "elbow".

  (The position is depending on robot type and can be user defined, but is located around axis 3).
- Wrist center point (WCP), in manual mode only.
- A number of configurable speed supervised points on the current tool.

If any of these points exceed the maximum speed, the safety controller triggers. If the TCP moves slower than the minimum speed, the safety controller will also trigger. The speed violation will cause a category 0 stop, a category 1 stop, or set an output signal, depending on the configuration.

There can be up to 8 global sets of Tool Speed Supervision plus one for each zone and axis range (up to 16 zones and 8 axis ranges).

WARNING

Tool Speed Supervision is not active in manual operating mode and shall therefore not be used to guarantee operator safety in that mode.

CAUTION

Since the TCP speed is determined by the programmed speed it is very important that the TCP of the active tool in SafeMove corresponds to the active tool of the robot program.

Note

The resultant robot TCP speed can in some situations be higher than the programmed TCP speed. This could happen for some robot types if the move instructions are of type `MoveJ` or `MoveAbsJ`. If this occurs, either increase the *Max Speed* for Tool Speed Supervision, or try to add intermediate robot targets in the RAPID program.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

2.5.3 Tool Speed Supervision (TSP)
Continued

Note

When the robot is running in manual mode, neither the elbow point nor the TCP point will exceed 250 mm/s. When the robot is running in auto mode, the robot controller will not consider the elbow speed when generating the path, only the defined TCP speed and reorientation speed. (If additional axis exists in the system, the speed data for this will also be considered.)

The result from this is that the elbow speed is sometimes higher than the programmed TCP speed. Since Tool Speed Supervision supervises the TCP, the elbow, and the speed supervision points on the tool, the speed of these points must be taken into account when creating the RAPID program.

Settings

The following parameters can be configured for each set of Tool Speed Supervision:

- An optional axis range or a zone for which the Tool Speed Supervision is applied.
- Maximum allowed speed (in mm/s) for TCP, elbow, and speed supervised points.
- An optional minimum speed for the TCP.
- Category 0 stop, category 1 stop, or no stop if a violation occurs.
- Set an output signal if a violation occurs.
- Set a status signal when the function is active.
- Assignment of a safe input for activation, or set as permanently active.

How to define these settings is described in *Configuring Tool Speed Supervision on page 118*.

Function activation

Tool Speed Supervision is activated by a safe input signal, or is permanently active.

Limitations

- Tool Speed Supervision is only available for SafeMove Pro, see *Functional safety options on page 16*.
- The highest speed limit that can be configured is 600 degrees/s for rotational axes and 10000 mm/s for linear axes.

© Copyright 2020 ABB. All rights reserved.

2.5.4 Axis Position Supervision (APO)

Axis Position Supervision

Axis Position Supervision is an active supervision function that triggers a violation if any axis is outside of the defined ranges.

Functionality

Supervision of up to 9 axes (robot axes and additional axes) in each set. Up to 8 sets can be configured, one for each safe axis range.

If an axis in an active set exceeds its allowed range, the safety controller triggers. This violation will cause a category 0 stop, a category 1 stop, and/or set an output signal, depending on the configuration.

Settings

The following parameters can be configured for Axis Position Supervision:

- A safe range to which it should be applied.
- Category 0 stop, category 1 stop, or no stop if an axis is outside its range.
- Set an output signal if an axis is outside its range.
- Set a status signal when the function is active.
- Assignment of safe inputs for activation of each set of axis ranges, or set as permanently activated.

How to define these settings is described in *Configuring Axis Position Supervision on page 123*.

Function activation

Axis Position Supervision is activated by a safe input signal, or is permanently active.

Continues on next page

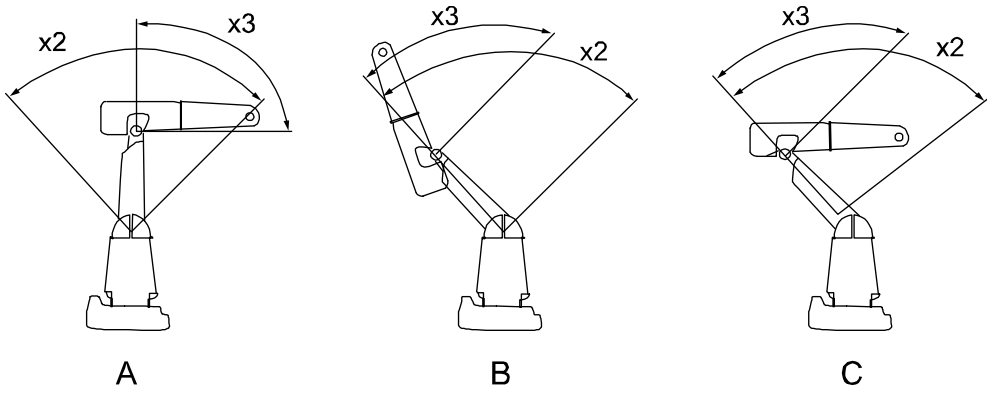
© Copyright 2020 ABB. All rights reserved.

2.5.4 Axis Position Supervision (APO)
Continued

Examples

This example shows a robot with defined axis ranges for axes 2 and 3 in three different positions. The function Axis Position Supervision supervises that axis 2 is within range x2 and that axis 3 is within range x3.

In positions A and B, all supervised axes are within the allowed ranges. In position C, axis 3 is not within the defined range.



xx0600003331

x2	Allowed axis position range for axis 2.
x3	Allowed axis position range for axis 3.
A	Robot position A. Both axis 2 and axis 3 are within the allowed ranges.
B	Robot position B. Both axis 2 and axis 3 are within the allowed ranges.
C	Robot position C. Axis 2 is within the allowed range but axis 3 is not within its allowed range. This will trigger a violation.

Note

The ranges define axis angles, not the position of the TCP. In robot position C, the TCP is still within what seems to be a safe range, but axis 3 is outside its defined range.

Limitations

WARNING

Be aware of that the braking starts when the axis exceeds the configured limit value.

The braking distance depends on robot type, load, position and speed, and therefore an additional stopping distance may sometimes be required to achieve the desired safety.

© Copyright 2020 ABB. All rights reserved.

2.5.5 Tool Position Supervision (TPO)

Tool Position Supervision

Tool Position Supervision is a supervision function that supervises that the robot is within the allowed safe zone.

Functionality

Tool Position Supervision supervises that the robot and the active safety tool (and any configured encapsulation around them) are within the defined zone. Up to 32 sets can be configured, max two per safe zone.

If the robot is outside its allowed zone, the safety controller triggers. This violation will cause a category 0 stop, a category 1 stop, and/or set an output signal, depending on the configuration.

Settings

The following parameters can be configured for Tool Position Supervision:

- A safe zone to which it should be applied.
- Assignment of a safe input for activation, or set as permanently active.
- Category 0 stop, category 1 stop, or no stop if the robot violates its zone limits.
- Set an output signal if the robot violates its zone limits.
- Set a status signal when the function is active.
- If the upper arm should be included in the supervision, or only the tool.
- If the robot should be allowed only inside or only outside of the zone.

How to define these settings is described in *Configuring Tool Position Supervision on page 117*.

Function activation

Tool Position Supervision is activated by a safe input signal, or is permanently active.

Limitations

- Tool Position Supervision is only available for SafeMove Pro, see *Functional safety options on page 16*.

WARNING

Be aware of that the braking starts when the tool or robot exceeds the configured limit value.

The braking distance depends on robot type, load, position and speed, and therefore an additional stopping distance may sometimes be required to achieve the desired safety.

© Copyright 2020 ABB. All rights reserved.

2.5.6 Tool Orientation Supervision (TOR)

Tool Orientation Supervision

Tool Orientation Supervision is an active supervision function that supervises that the tool orientation of the active safety tool is within the allowed tolerance.

Functionality

Tool Orientation Supervision supervises the tool orientation. If the tool orientation is outside its allowed tolerance, the safety controller triggers. This violation will cause a category 0 stop, a category 1 stop, or set an output signal, depending on the configuration.

Up to 8 sets can be configured.

CAUTION

Since the tool orientation is determined by the programmed tool orientation, it is very important that the active tool in SafeMove corresponds to the active tool of the robot program.

Settings

The following parameters can be configured for Tool Orientation Supervision:

- An optional axis range or a zone for which the Tool Orientation Supervision is applied.
- Allowed orientation of the tool in x and z directions including a tolerance.
- Assignment of a safe input for activation, or set as permanently active.
- Category 0 stop, category 1 stop, or no stop if a violation occurs.
- Set an output signal if the tool orientation violates its limits.
- Set a status signal when the function is active.

How to define these settings is described in *Configuring Tool Orientation Supervision on page 121*.

Function activation

Tool Orientation Supervision is activated by a safe input signal, or is permanently active.

Limitations

- Tool Orientation Supervision is only available for SafeMove Pro, see *Functional safety options on page 16*.

© Copyright 2020 ABB. All rights reserved.

2.5.7 Control Error Supervision

Control Error Supervision

Control Error Supervision is a function that supervises the difference between the reference value and the measured value of the motor position of each axis. Control Error Supervision is required to ensure the accuracy in the supervision functions and the structural category 3 of the safety system, i.e. dual channel supervision.

Functionality

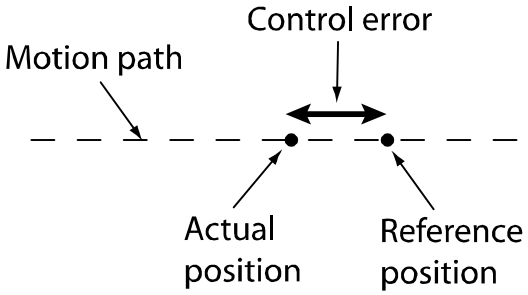
The control error (servo lag) is the absolute value of the difference between the reference value and the measured value of the motor position of each axis.

Control Error Supervision is activated automatically after the safety controller has been synchronized with the robot position.

When Control Error Supervision triggers, the following happens:

- The robot is stopped with a category 1 stop.
- An event log message (90511) is sent to the robot controller.

Illustration of control error



en0700000723

Function activation

Control Error Supervision is always active. It can only be relaxed by Contact Application Tolerance.

Dependencies to other functions

If Contact Application Tolerance is active, then Control Error Supervision is relaxed according to user definitions.

© Copyright 2020 ABB. All rights reserved.

This page is intentionally left blank

3 Installation

3.1 Hardware

3.1.1 Safety module

Safety module

The safety module is installed at delivery inside the main computer of the robot controller. For more information, see the product manual for the controller.

© Copyright 2020 ABB. All rights reserved.

3.2 Software

Install RobotStudio

The configuration environment, *Visual SafeMove* is included in the standard installation of RobotStudio.

	Action
1	Install RobotStudio. Visual SafeMove is included in both the *Minimal* and *Full* version of RobotStudio.
2	Start RobotStudio.
3	Start Visual SafeMove by going to the **Controller** tab on the ribbon, click **Safety** and select **Visual SafeMove**.

For more information see *Operating manual - RobotStudio*.

Note

RobotStudio must be of the same version or later than the RobotWare used.

Create a robot system

A RobotWare license with the option *SafeMove Basic* or *SafeMove Pro* is required to run SafeMove on the robot controller.

Use RobotStudio to configure, build, and download a RobotWare system to the robot controller.

For more information see *Operating manual - RobotStudio*.

© Copyright 2020 ABB. All rights reserved.

4 The Visual SafeMove user interface in RobotStudio

4.1 About Visual SafeMove

What is Visual SafeMove

Visual SafeMove is the configuration tool for SafeMove and the functional safety options. The tool is completely integrated into the RobotStudio user interface and takes full advantage of the user interface elements such as tabs, browsers, and 3D graphics.

Visual SafeMove is enabled for robots with the safety module option. It offers an intuitive way to visualize and configure safety zones. Zones can be adjusted by direct manipulation in the 3D window. Users with previous experience from SafeMove will recognize the same terminology used as before.

Visual SafeMove works both with the real controller and the virtual controller. For a virtual controller, a RobotStudio station should be used, which allows zones to be generated automatically. When not running a RobotStudio station, **Online Monitor** is used to visualize the robot.

Starting Visual SafeMove

	Action
1	Start RobotStudio with a virtual controller (with or without a station) or connect a real controller.
2	In the **Controller** tab, click **Online Monitor**. (Not needed when running a RobotStudio station.)
3	In the **Controller** tab, click **Safety**, then select **Visual SafeMove**.

Prerequisites

- Some functionality is only available for SafeMove Pro, see *Functional safety options on page 16*.
- Only a user with the grant **Safety Services** is allowed to download a configuration. See *Set up safety user grants on page 97*.

Limitations

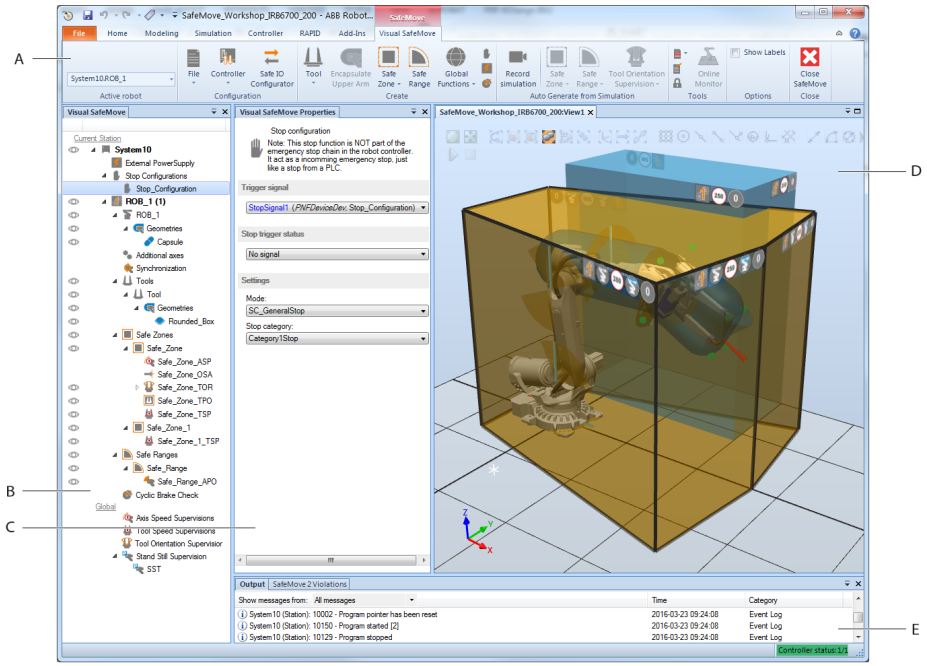
The simulation functions are only available when running a RobotStudio station.

© Copyright 2020 ABB. All rights reserved.

4.2 The user interface

Overview of the user interface

This section presents an overview of the Visual SafeMove graphical user interface.



xx1500000804

	Parts	Description
A	The **Visual SafeMove** ribbon	Displays groups of icons organized in a logical sequence of function.
	The **Modify** tab	Is used by some functions in the **Visual SafeMove** ribbon to display additional functionality.
B	**Visual SafeMove** browser	Displays all available SafeMove functions.
C	**Visual SafeMove Properties** browser	Displays all available properties and settings of the selected SafeMove function.
D	Graphics window	Is used to to visualize and configure safety zones in the RobotStudio station. When not running a RobotStudio station, **Online Monitor** is used to visualize the robot.
E	**Output** window	The output window displays information about events that occur in RobotStudio, both general events and Visual SafeMove events.
	SafeMove Violations window	Displays all violations that has occurred since the last restart. Click a violation in the list to show a detailed view of all geometries that are related to the violation. The geometries that are not related are automatically hidden. The violations window can also be used to view a violation when RobotStudio is connected to a real robot.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

The tabs

The **Visual SafeMove** tab and the **Modify** tab contains groups of commands organized in a logical sequence of functions that simplifies the configuration of SafeMove, see *The Visual SafeMove tab on page 57* and *The Modify tab on page 65*.

The browsers

The configured safety functions are available from the **Visual SafeMove** browser. When a function is selected by clicking the node in the browser, the properties and settings are displayed in the **Visual SafeMove Properties** browser, see *The Visual SafeMove browser on page 68*.

Use the **Visual SafeMove** browser, or standard keyboard shortcuts, to cut, copy, and paste zones both between zone types and between robots.

The graphics window

In general the Visual SafeMove graphics window is navigated using the same commands as in RobotStudio. A few additional navigation tools are available in Visual SafeMove for editing zones, those are listed below.

For more information on navigating RobotStudio, see *Operating manual - RobotStudio*.

Editing zones in the graphics window

Zones are displayed as semi-transparent (opaque) planes.

The following navigation options are available in Visual SafeMove for editing zones:

- Drag and drop on spheres in the graphics window (at corners and surfaces) to modify a zone.

  Tip

  To lock an axis while dragging, press **x** or **y** on the keyboard.

- Double click vertices or surfaces in the graphics window to create new corners/vertices at that location.
- Delete vertices that are no longer desired by selecting and deleting them from the graphics window.
- Move, rotate, and change the size of a zone by dragging the zone frame arrows in the graphics window, see *The Freehand group on page 67*.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

4.2 The user interface
Continued

Symbols in the graphics window

The following symbols are used in the graphics window to illustrate the properties of the zones.

 xx1600000241	 xx1600000242	 xx1600000243	 xx1600000244	 xx1600000245	 xx1600000246
Robot number	Upper arm included in Tool Position Supervision	Zone minimum speed limit	Zone maximum speed limit	Robot allowed inside zone	Robot not allowed inside zone

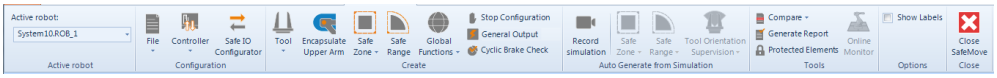
© Copyright 2020 ABB. All rights reserved.

4.3 The Visual SafeMove tab

4.3.1 About the Visual SafeMove tab

Layout of the ribbon

The **Visual SafeMove** tab contains groups of commands organized in a logical sequence of functions that simplifies the configuration of SafeMove.



xx1600000060

The ribbon consists of the following groups:

Group	Functions used for
Active robot	Selects the active robot in a MultiMove system.
Configuration	Creating, saving and loading SafeMove configurations. *Configuration group on page 57*
Create	Adding and configuring SafeMove functions. *Adding SafeMove functions on page 60*
Auto Generate From Simulation	Creating a SafeMove function based on a simulated path in RobotStudio. *Simulating SafeMove on page 86* **Note** The simulation functions are only available when running a RobotStudio station.
Tools	Different tools for analyzing SafeMove configurations. *The Tools group on page 59*
Options	Display options for the Visual SafeMove user interface. *Options group on page 59*

Tip

Most commands in the ribbon are also available from the **Visual SafeMove** browser by right-clicking the nodes in the tree.

Configuration group

The **Configuration** group is used when creating, saving and loading SafeMove configurations.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

4.3.1 About the **Visual SafeMove** tab
Continued

The safety configuration in RobotStudio that has not yet been written to the controller is referred to as "local configuration". The safety configuration on the controller is referred to as "controller configuration".

The **File** button

Button	Description
New / **Controller Configuration**	Create a new controller configuration.
New / **Drive Module Configuration**	Create a new drive module configuration. The new configuration will be in the latest format version.
Open configuration	Open a controller configuration. The format version of the opened configuration will be kept.
Save configuration as	Save the safety or geometry configuration to file. For more information about the geometry configuration file, see *SafeMove geometry configuration file on page 169*.
Import geometries	Imports safety geometries such as zones, tools and robot upper arm geometry from a safety configuration file or a simplified configuration file generated by an external tool. Geometries with the same name will be overwritten.
Import protected elements	Imports the protected elements from the selected file. All currently protected elements will be unprotected and the protected elements in the selected file will be imported to the local configuration. The protected elements checksum of the local configuration will be the same as the selected files protected checksum after this operation.

The **Controller** button

Button	Description
Read from controller	Reads the configuration from the safety controller.
Write to controller	Writes the configuration to the safety controller.
Upgrade configuration to latest version	Upgrades the safety configuration to the latest format version. The checksums will be updated.
Reset to factory settings	Resets the configuration in the safety controller to default settings. See also *Reset safety controller to factory settings on page 144*.
Restore configuration	Restores the configuration from file. The file is restored as it is. It is not opened in RobotStudio.

The **Safe IO Configurator** button

The **Safe IO Configurator** button is used to start the SafeMove IO configurator.

The safe IO configurator is described in a separate chapter, see *The Safe IO Configurator on page 72*.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

The Tools group

The **Tools** group contains different tools for analyzing and protecting SafeMove configurations.

Button	Description
Compare	The **Compare** button starts a tool that is used to compare the differences between two configuration files.
Show report	Shows the report for the safety configuration on the controller. To show the local safety configuration report, click **Controller** and select **Write to controller**. The local report is shown and if you do not want to write it to the controller, select **Cancel**.
Protected Elements	The **Protected Elements** button opens the **Protected Elements** browser. From the **Protected Elements** browser it is possible to write-protect settings in the SafeMove configuration so that the user cannot change them. The write-protected element are visualized with a padlock icon in the **Visual SafeMove** browser. It is also possible to create a basic configuration that can be used as a template when configuring a new system, see *Protected basic configuration on page 87*.

Options group

The **Options** group contains display options for the Visual SafeMove user interface.

Checkbox	Description
Show Labels	Displays labels in the graphics window.

© Copyright 2020 ABB. All rights reserved.

4.3.2 Adding SafeMove functions

About the Create group

The **Create** group is used when creating and configuring SafeMove functions.

> **Tip**
>
> Most commands in the ribbon are also available from the **Visual SafeMove** browser, by right-clicking the nodes in the tree.

The Tool button

Button	Description
New	Create a new tool.
Encapsulate	Creates a geometry that encapsulates the selected tool. Up to four different geometries can be used to create a more accurate encapsulation of the tool.

Settings

The following settings are available in the **Visual SafeMove Properties** browser after the tool has been created.

Setting	Description
Make default Tool	If more than one tool is configured, one of them must be selected as default tool. The default tool will be used when no tool is selected by input signals. The tool that already is the default tool has a greyed out button with the text **Is default Tool**.
Activation	Specifies the safety signal that activates the selected tool. The setting **Permanently active** is used for a tool that is always active. This setting needs to be deactivated to be able to create more than one tool.
Function active status	Specifies a safety signal that indicates that the selected tool is active. The setting **No signal** is used for a tool that is always active.
Tool data	Modify the tool data, or load a tool data from the RobotStudio station. **CAUTION** Since the tool data is determined by the programmed tool, it is very important that the active tool in SafeMove corresponds to the active tool of the robot program.
Speed supervised points	The tool can use up to eight points to supervise the speed of the tool. By default all eight points are created. It is recommended to reduce the number of points if all points are not required depending on the shape of the tool.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

The Encapsulate Upper Arm button

The **Encapsulate Upper Arm** button is used to create a geometry that encapsulates the upper arm. The geometry can then be adapted to fit the needs of the current application.

Up to two different geometries can be used to create a more accurate encapsulation of the upper arm.

The Safe Zone button

The **Safe Zone** button is used to create safe zones. The safe zone is displayed in the graphics window as a rectangular box with vertices.

Button	Description
Safe Zone	Creates a default safety zone in the shape of a rectangular box.
Wrapped Safe Zone	Crates a safe zone by wrapping a part in the RobotStudio station. The part must first be selected.

After a zone has been created, the next step is to add safety functions to the zone. This is described in section *The Modify tab on page 65*.

Tip

It is recommended to change the default name of the safe zone in the **Visual SafeMove** browser to a name that better corresponds to the current installation.

Settings

The following settings are available in the **Visual SafeMove Properties** browser after the zone has been created.

Setting	Description
Tool Speed Supervision Priority	Used to set the priority for overlapping zones. The zone with the highest priority will set the speed limit for the overlapping space. • **BASE** - lowest priority. • **NORMAL** - normal priority. • **OVERRIDE** - highest priority. For more information, see *Configure the zones on page 110*. **Tip** Create a **BASE** zone with a low speed limit that covers the entire robot cell. This zone is then used to protect the fence. Add **NORMAL** zones for the areas where the robot performs most of the work. Additionally, add **OVERRIDE** zones for any areas where the robot is allowed to move fast.
Reference	The reference coordinate system in the RobotStudio station. **Note** The base frame cannot be used for reference if it is rotated around the controller world x or y-axes, for example when using a tilted robot.
Height	The height of the zone.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

4.3.2 Adding SafeMove functions
Continued

Setting	Description
Vertices	Displays the vertices of the safe zone. It is possible to add or delete vertices from the **Visual SafeMove Properties**.  **Tip** Add a vertex by double-clicking the edge of a zone in the graphics window. Delete a vertex by first selecting the vertex in the graphics window and then pressing delete.

The Safe Range button

Click the **Safe Range** button to create a safe range. The safe ranges are displayed in the graphics window as a discs around the rotation centers of the selected axes.

After a range has been created, the next step is to add safety functions to the range. This is described in section *The Modify tab on page 65*.

Settings

The following settings are available in the **Visual SafeMove Properties** browser after the range has been created.

Setting	Description
Joint	The joints of the robot.
Enabled	Safe range is enabled for the joint.
Lower bound	The lower bound limit in degrees.
Upper bound	The upper bound limit in degrees.
Invert	If selected, the supervised angles for that axis will be below the **Lower bound** and above the **Upper bound**.

The Global Functions button

The **Global Functions** button is used to create global safety functions. Global safety functions are always active and are not connected to a specific safe zone or safe range.

The below functions, except Stand Still Supervision, can also be assigned to individual safe zones and safe ranges.

The following global functions are available:

Button	Description
Global Tool Orientation Supervision	*Tool Orientation Supervision (TOR) on page 48*
Global Tool Speed Supervision	*Tool Speed Supervision (TSP) on page 43*
Global Axis Speed Supervision	*Axis Speed Supervision (ASP) on page 42*
Stand Still Supervision	*Stand Still Supervision (SST) on page 40*

Settings

The settings for the global safety functions are the same as for the safe zones and safe ranges. For information about the settings see *The Add Safety Function group on page 66*.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

The Stop Configuration button

The **Stop Configuration** button is used configure a safety stop that is sent on the safe fieldbus from the safety PLC to the robot controller.

The **Mode** setting defines which stop mode that shall be activated on the robot controller (AS or ES). The corresponding warning will be written to the event log and presented on the FlexPendant.

CAUTION

A safety stop from the safety PLC will not affect external units that may be connected to the automatic stop input (AS) or the emergency stop input (ES) of the robot controller. For example: external emergency stop buttons, light curtains, door breakers, etc.

Setting	Description
Trigger signal	Specifies the safety signal that activates the safety stop. The signal is set to 0 for activation.
Stop trigger status	Specifies a safety signal that indicates that the the safety stop is active. The signal is set to 0 when triggered. The setting **No signal** is used if no status signal should be used.
Settings • **Mode**	The **Mode** setting defines which stop mode that shall be activated on the robot controller. The corresponding warning will be written to the event log and presented on the FlexPendant. • **SC_AutoStop** - puts the robot controller in auto stop mode (AS). • **SC_EmergencyStop** - puts the robot controller in emergency stop mode (ES).
Settings • **Stop category**	• **Category0Stop** - Stop by immediate removal of power to the actuators. Mechanical brakes are applied. A robot that is stopped with a category 0 stop does not follow its programmed path while decelerating. • **Category1Stop** - Controlled stop with power available to the actuators to achieve the stop. Power is removed from the actuators when the stop is achieved. A robot that is stopped with a category 1 stop follows its programmed path while decelerating.

The Cyclic Brake Check button

The **Cyclic Brake Check** button is used to setup the cyclic brake check function.

For more information, see *Cyclic Brake Check guidelines on page 158*.

Setting	Description
Warning only, no stop	If this checkbox is selected, the robot will not stop if a cyclic brake check has not been performed on time. Only a warning will be written to the event log and presented on the FlexPendant.
Max CBC test interval	The maximum allowed time between cyclic brake checks.
Pre warning time	The warning time before a cyclic brake check must be performed.
Standstill tolerance	The maximum allowed movement during a cyclic brake check.
Supervision threshold	The minimum servo lag used for detecting that the cyclic brake check is performed.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

4.3.2 Adding SafeMove functions
Continued

Setting	Description
ROB1	If a checkbox is deactivated, then cyclic brake check is deactivated for that joint.

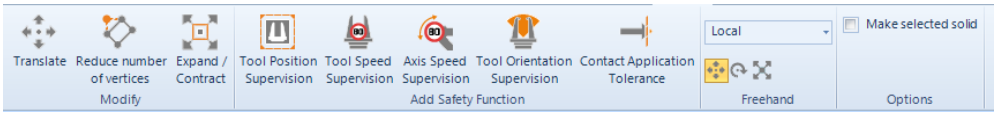
© Copyright 2020 ABB. All rights reserved.

4.4 The Modify tab

Layout of the ribbon

The **Modify** tab is a sub-tab to the **Visual SafeMove** tab and contains additional functions and settings for the selected object.

The functions are different depending on which object that is selected. The below picture show the available functions for a safe zone.



xx1600000063

Tip

Most commands in the ribbon are also available from the **Visual SafeMove** browser, by right-clicking the nodes in the tree.

The Create group

The **Create** group is only available when modifying a tool.

Up to four different geometries can be used to create a more accurate encapsulation of the tool.

Button	Description
Sphere	Creates the corresponding geometry that can be used to create a more accurate encapsulation of the tool.
Capsule	
Rounded box	

Settings

It is possible to modify the settings for the geometries from the **Visual SafeMove Properties** browser.

The Modify group

The **Modify** group is only available when modifying a safe zone.

Button	Description
Translate	Translates the zone.
Reduce number of vertices	Reduces the number of vertices for the zone without reducing the area covered by the safe zone. Note that the shape of the safe zone will change, and possibly grow, but it will still cover the original area.
Expand/Contract	Expands the zone in all directions except the height.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

The Add Safety Function group

The **Add Safety Function** group is used when creating and configuring safe zone and safe range safety functions.

The following functions are available for both safe zones and safe ranges:

Button	Description
Axis Speed Supervision	*Axis Speed Supervision (ASP) on page 42*
Contact Application Tolerance	*Contact Application Tolerance (CAP) on page 34*
Tool Orientation Supervision	*Tool Orientation Supervision (TOR) on page 48*
Tool Speed Supervision	*Tool Speed Supervision (TSP) on page 43*

The following function is only available for safe zones:

Button	Description
Tool Position Supervision	*Tool Position Supervision (TPO) on page 47*

The following function is only available for safe ranges:

Button	Description
Axis Position Supervision	*Axis Position Supervision (APO) on page 45*

The following function is only available as a global function:

Button	Description
Stand Still Supervision	*Stand Still Supervision (SST) on page 40*

Note

For information about global functions, see *The Global Functions button on page 62*.

Settings

The following settings are available in the **Visual SafeMove Properties** browser, and are common for all safety functions.

Setting	Description
Activation	Specifies the safety signal that activates the supervision. The signal is set to 0 for activation. The setting **Permanently active** is used for a supervision that is always active.
Function active status	Specifies a safety signal that indicates that the selected safety function is active. The signal is set to 1 when active. The setting **No signal** is used if no signal should be used.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Setting	Description
Violation action • **Stop type**	• **Category0Stop** - Stop by immediate removal of power to the actuators. Mechanical brakes are applied. A robot that is stopped with a category 0 stop does not follow its programmed path while decelerating. • **Category1Stop** - Controlled stop with power available to the actuators to achieve the stop. Power is removed from the actuators when the stop is achieved. A robot that is stopped with a category 1 stop follows its programmed path while decelerating. • **NoStop** - Passive monitoring with signaling function only.
Violation action • **Signal**	Specifies a safety signal that indicates that the the axis speed supervision is violated. The signal is set to 0 at violation. The setting **No signal** is used if no violation signal should be used. **Note** When a signal is set to 0 at violation, it will remain 0 for at least 250 ms even after the violation has ended.

The following settings are available in the **Visual SafeMove Properties** browser, and are individual for each safety function.

Function	Setting	Description
ASP	**Speed limits**	The minimum and maximum speed for each robot joint.
TOR	**X Vector** **Z Vector**	The tolerance around the x and z axis of the tool.
	Get vectors for active tool (button)	Gets the vectors from the active tool in RAPID. The tool tip shows which tool is active. **CAUTION** Since the tool data is determined by the programmed tool, it is very important that the active tool in SafeMove corresponds to the active tool of the robot program.
TSP	**Speed limits**	The minimum and maximum speed for the tool.
SST	**Tolerances**	Specifies if stand still supervision is enabled and the tolerance for each robot joint.

The Freehand group

The **Freehand** group is only available when modifying a tool or a safe zone.

Button	Description
Move	In the graphics window, click one of the axes and drag the zone into position.
Rotate	In the graphics window, click the rotational ring and rotate the zone into position.
Size	In the graphics window, click the arrow and drag to change the size of the zone.

© Copyright 2020 ABB. All rights reserved.

4.5 The Visual SafeMove browser

About the Visual SafeMove browser

When adding a safety function from the **Visual SafeMove** tab or the **Modify** tab, it is automatically displayed in the **Visual SafeMove** browser. These functions and settings are described in the corresponding section where they are added.

In addition to that, the **Visual SafeMove** browser also has some system related settings that are described in this section.

Robot properties node

Setting	Description
Max speed in manual mode	The max speed cannot be higher than the default value of 250 mm/s, but a lower value can be set.

Note

If reducing the max speed that SafeMove allows in manual mode, the jogging speed of the robot has to be reduced to the same value. Change the parameter *Teach Mode Max Speed*, topic *Motion* and type *Motion Planner*. See *Technical reference manual - System parameters*.

Robot parameters node

Setting	Description
Elbow offset • **Position X, Y, Z**	If any extra equipment is attached to the upper arm, a point on this equipment can be defined as a check point. The robot will then monitor the speed of this point so that it does not exceed 250 mm/s in manual reduced speed mode. See *Explanation of Elbow offset on page 69*.
Safe Brake Ramp Data • **Start Speed Offset**	A speed offset for the Safe Brake Ramp function. For track motions and other additional axes, the parameters **Brake Ramp Limit** and **Ramp Delay** have to be set. The parameter **Start Speed Offset** is used for both manipulator and all additional axes, see *Explanation of Safe Brake Ramp on page 71*.

Continues on next page

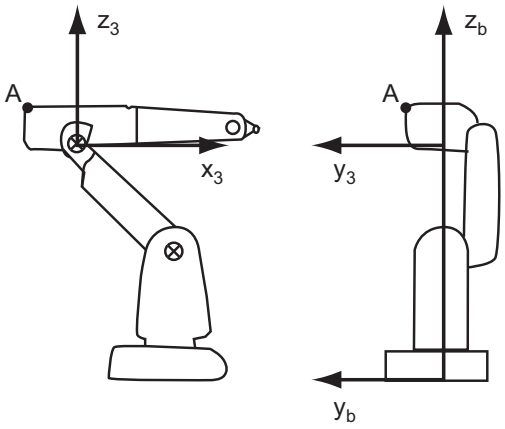
© Copyright 2020 ABB. All rights reserved.

The following settings are available when right-clicking the robot parameters node:

Setting	Description
Exclude from configuration	Excludes the robot from the safety configuration. This setting must be used when configuring a safe fieldbus on robots that are not supported by SafeMove. This setting can also be used to exclude supported robots, for example individual robots in a MultiMove application and robots running only a safe fieldbus. See *Overview of SafeMove functions on page 31*. **DANGER** A SafeMove configuration must always be validated to verify that the desired safety is achieved. If no validation is performed, or the validation is inadequate, the configuration cannot be relied on for personal safety.

Explanation of **Elbow offset**

An elbow point is considered for Tool Speed Supervision. The elbow point is given a default value based on robot model. This value can be changed in the configuration. Specify the elbow point's x, y and z offsets relative to the center of robot axis 3. Note that the X value should always be negative.



xx1300002628

A	Elbow point
x_3, y_3, z_3	Axis 3
y_b, z_b	Robot base

Note

The values that are entered into **Elbow Offset** should also be entered into the parameter *Arm Check Point*, topic *Motion*. This is to avoid different speed calculations between the robot controller and SafeMove in manual reduced speed mode.

For more information, see *Technical reference manual - System parameters*.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Explanation of **Base Frame**

All values for the base frame are automatically loaded from the robot controller and cannot be changed from the **Visual SafeMove** browser.

Setting	Description
Base frame • **Reference**	Zone can be defined in either task frame, world coordinate system, user coordinate system, or robot base frame. These coordinate systems are often identical, but for MultiMove systems it may be desired to do the configuration in the robot base frame.
Base frame • **Position X, Y, Z**	X, Y and Z values for the selected reference frame's origin, expressed in the world coordinate system.
Base frame • **Orientation**	Defines the orientation of the selected reference frame, compared to the world coordinate system.

Additional axes parameters node

If the axis should be part of the SafeMove supervision, select the check box **Is supervised**.

Joint parameters

Setting	Description
Servo lag	**Servo lag** is the estimated lag (in radians on motor side) for the additional axis. For more information, see *Servo Delay Factor and Servo Lag on page 165*.
Servo delay factor	Estimated delay factor between reference position and measured position (number of 4 ms units) when moving the additional axis. (See TuneMaster, signal number 17 and 18.) For more information, see *Servo Delay Factor and Servo Lag on page 165*.
Max speed manual mode	The maximum speed in manual mode.

Brake ramp supervision parameters

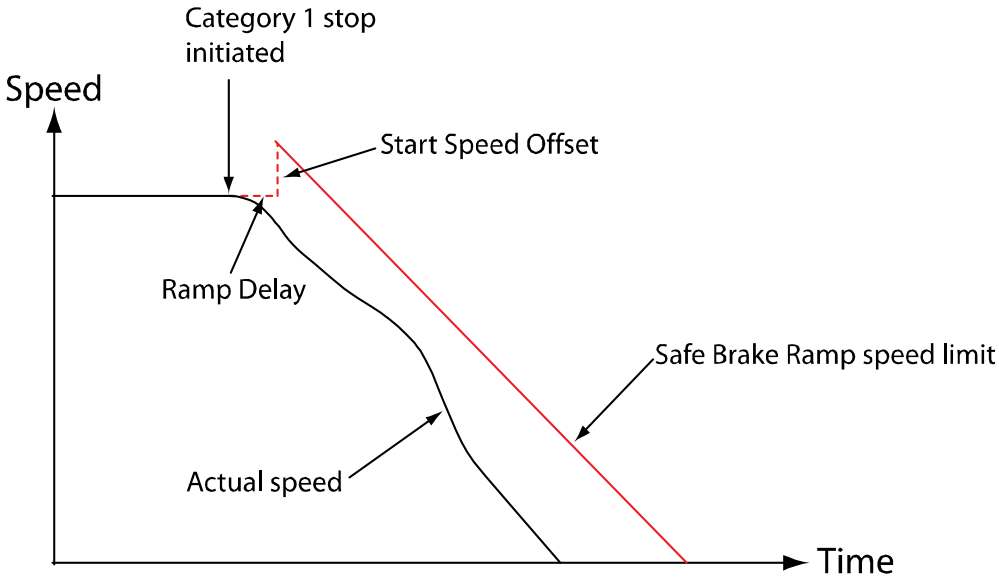
Setting	Description
Safe brake ramp enabled (check box)	Selects if safe brake ramp should be used for the additional axis.
Ramp delay	Delays the Safe Brake Ramp function. See figure below. Default value: 200 ms.
Brake ramp limit	If the actual deceleration is lower than the specified **Brake Ramp Limit**, then Safe Brake Ramp will cause a category 0 stop. The value is specified for the arm side.
Start speed offset	A speed offset for the Safe Brake Ramp function.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Explanation of Safe Brake Ramp

Safe Brake Ramp supervises that a category 1 stop decelerates fast enough. To avoid that Safe Brake Ramp triggers every time, some margins are necessary. For additional axes, it is possible to set the margins both in time and start speed, and the slope of the speed limit ramp. For the robot, it is only possible to set the margin for the start speed.



en0700000724

Synchronization node

Setting	Description
Activation	To use software synchronization, select **Software synchronization**. To use hardware synchronization, select the input signal from the sync switch.
Synchronization status	Specifies a safety signal that indicates that the safety controller is synchronized The signal is set to 1 when synchronized. The setting **No signal** is used if no signal should be used.
ROB1 • **Joint** • **Position**	The axis position values of the robot.
Additional axis • **Joint** • **Position**	The axis position value of the additional axis.

© Copyright 2020 ABB. All rights reserved.

4.6 The Safe IO Configurator

4.6.1 Introduction

About the Safe IO Configurator

The **Safe IO Configurator** in the Visual SafeMove user interface is used to configure safe I/O signals.

The safe I/O configurator has the following main purposes:

- Configure safety signals.
- Connect signals to general purpose safety functions.
- Configure combinatory logic.

The **Safe IO Configuration** window consists of the following views:

View	Description
Signals	Configuration of parameters for the safe fieldbus modules in the robot controller and configuration of safety signals for the safe fieldbuses and for global (virtual) safety signals.
Function Mappings	Configuration for mapping of signals to specific safety status information and functions.
Pre Logic	Configuration of combinatory logic for the safety signals that is to be executed before the safety application (AND, OR, XOR, etc.).
Post Logic	Configuration of combinatory logic for the safety signals that is to be executed after the safety application (AND, OR, XOR, etc.).

For instructions on how to perform the configuration, see *Configure safe I/O on page 108*. For information on general rules and limitations see *Safe I/O system rules and limitations on page 85*.

© Copyright 2020 ABB. All rights reserved.

4.6.2 Signals view

Introduction

The **Signals** view contains configuration of global safety signals (virtual signals) to be used internally in RAPID and in the safety controller, see *Global signals on page 73*. If a safe fieldbus is installed, the **Signals** view also contains configuration of parameters for the safe fieldbuses and configuration of safe input and safe output signals for the safe fieldbus in the robot controller.

The safety signals can be read from RAPID, but they can only be set from the safety controller, see *Accessing safe signals and feedback signals on page 74*.

For more information on how to configure safe I/O, see *Configure safe I/O on page 108*.

Configuring signals

The following safe signal settings are available.

Setting	Description
Signal Name	The name of the safety signal.
Default value	For input signals, the default value has no significance. For global and output signals, the default value can be used to define constant TRUE or FALSE.
Offset	The mapping of the signal.
Direction	Input or output.
Signal uses	Lists all readers and writers of the signal. A signal can only have one writer, but it can have more than one reader.

Tip

It is possible to use a spreadsheet application, or text editor, to edit the names of the signals and then copy-paste them into RobotStudio.

Global signals

The global signals are virtual signals that are to be used internally in RAPID and in the safety controller. The user can create up to 2048 global signals. The signals can be read from RAPID, but they can only be set from the safety controller, see *Accessing safe signals and feedback signals on page 74*.

There is a set of predefined global signals that corresponds to safety functions with the same names. It is allowed to change the names of these signals as long as the signal is still mapped to the corresponding safety function. It is also allowed to delete them as long as the function is replaced by another signal.

Predefined global signals

The following global signals are predefined and corresponds to safety functions with the same names:

- AutomaticMode

Continues on next page

© Copyright 2020 ABB. All rights reserved.

4.6.2 Signals view
Continued

- DriveEnable
- LocalEmergencyStopStatus
- ManualFullSpeedMode
- ManualMode
- DriveEnableFeedback
- SafetyEnable

For a description of the safety functions, see *Description of safety functions on page 75*.

Accessing safe signals and feedback signals

The input, output, and global signals are automatically made available in RAPID. The signals can be found together with all other I/O-signals in the I/O-configuration views both in RobotStudio and on the FlexPendant.

The signals can be read from RAPID, but they can only be set from the safety controller.

The global signals are made available in RAPID as feedback signals. The feedback signals are inputs located on the simulated I/O device *SC_Feedback_Dev* on the simulated bus *SC_Feedback_Net*.

© Copyright 2020 ABB. All rights reserved.

4.6.3 Function mappings

Configuring function mappings

Some safety functions, for example the system modes, need defined safety signals to interact with the rest of the system.

To allow the user to choose their own name for signals, these safety functions use predefined signal aliases instead, thus ensuring that the correct signal is read.

The role of the function mappings is to connect safety signals to aliases.

Setting	Description
Function	The name of the safety function or state.
Direction	Defines if the safety function writes the signal or reads the signal.  Safety function writes to signal.  Safety function reads signal.
Signal	The signal that is currently mapped to the function (input, output, or global).
Source	The source gives information on which part of the system writes value to the signal. This can be either **Safe local I/O**, **Safety system**, or user defined. The source also gives information on when a signal/alias gets its value updated, i.e. **Safe local I/O** is first and **Safety system** is after the pre-logic.
Mandatory	True or false. Some mappings are mandatory since they are used by the safety controller.
Description	See *Description of safety functions on page 75*.
Errors	Displays error, for example when the mapping for a mandatory function is missing.

Note

ExtComShutdownAck and *SafetyEnable* has no source if they are not mapped to a signal which is written. These signals should be written to before the safety application (e.g. device input or from pre-logic) or used with their default values.

Description of safety functions

The following safety functions are available:

Function	Description
AutomaticMode	True if **Automatic mode** is selected.
ConfigurationLocked[i]	True if configuration is locked. Can, for example, be used as extra precaution by connecting to a PLC that disables robot operation in automatic mode when the configuration is unlocked.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

4.6.3 Function mappings

Continued

Function	Description
DriveEnable	True if power is enabled to the actuators. DriveEnable is the safety controllers way of ensuring that there is only power enabled to the actuators when there are no safety violations. This is done through the superior stop input on the panel board.
LocalEmergencyStopStatus	False if the emergency stop on the FlexPendant is pressed. This can be used to send local emergency stop status to other devices.
ExternalEmergencyStop-Status	False if the external emergency stop is pressed. This can be used to send external emergency stop status to other devices.
ExtComShutdownReq [ii]	Set to True when the safe external communication is about to be terminated, beginning in no less than 1 second. This will happen in case of controlled shutdown, for example during robot controller shutdown.
ExtComShutdownAck [ii]	Only used only with ExtComShutdownReq. If True, the safety controller is allowed to shut down before 1 second has passed after the ExtComShutdownReq was set. For example, this can be set to True by the PLC when preparations have been made after ExtComShutdownReq is set to True.
ManualMode	True if manual mode is selected.
ManualFullSpeedMode	True if manual full speed mode is selected.
DriveEnableFeedback	True if the power to the motors is enabled.
SafetyEnable	The SafetyEnable function should, when possible, be connected to a safe input signal from, for example, a PLC. This signal shall be set to True as soon as communication is up and running. When False, all signal activated supervision functions, such as stop configurations, will be deactivated, and DriveEnable will be set to False, disabling robot motion.
ServiceModeActive [i]	True if **Service mode** is activated.
Stop0Status	False if a category 0 stop is active.
Stop1Status	False if a category 1 stop is active. Note that this will be a very brief indication since a category 1 stop is converted to a category 0 stop when all robot motion has ceased.
SafetyControllerOperational	True if the safety controller is running without errors. It is False during start-up before SafetyEnable is set to True, during shutdown of the controller or if there is a critical failure of the safety controller.

i Only used for option *SafeMove Basic* or *SafeMove Pro*.
ii Only with safe fieldbus option (3023-1, 3023-2 or 3026-2).

© Copyright 2020 ABB. All rights reserved.

4.6.4 Pre-logic and post-logic

Configuring combinatory logic

The combinatory logic has two layers:

- Pre-logic - executed before the safety application
- Post-logic - executed after the safety application

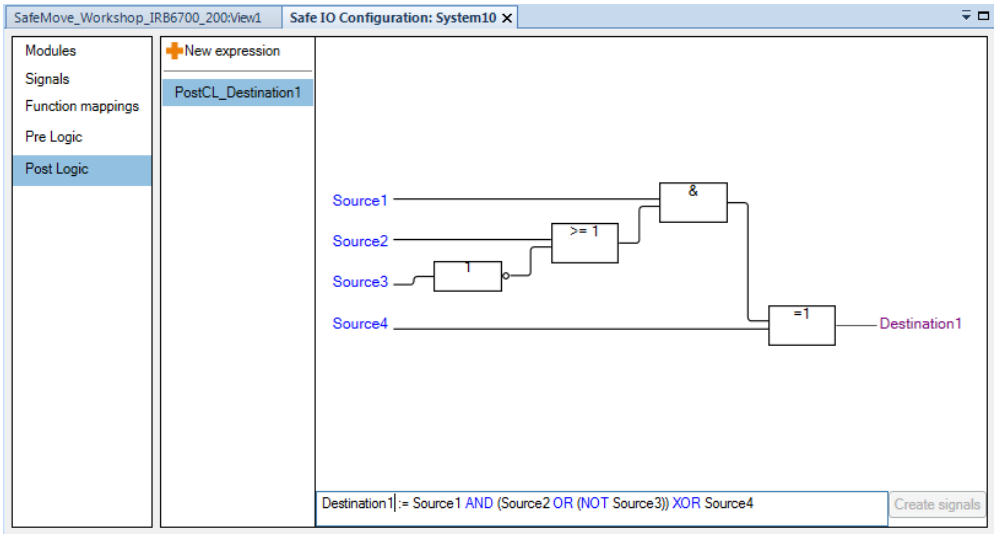
The user defines logic diagrams by writing statements. The combinatory logic operators can be grouped into logical operators, arithmetic operators, and complex operators.

For more detailed information including rules and limitations, see *Safe I/O system rules and limitations on page 85*.

Creating statements

Operators can be combined into complex statements.

Global signals can automatically be created from the statements by clicking **Create signals**.



xx1600000054

Note

Signals between blocks are created internally and will not be available to the user. For example the signal between the *OR* and the *AND* block in the above picture.

Tip

First use the button **Create signals** to automatically create global signals from the statement, then use cut and paste to move signals to safe inputs or outputs.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

4.6.4 Pre-logic and post-logic
Continued

Tip

When viewing large statements, press the left mouse button to pan and use the scroll wheel to zoom.

Overview of the combinatory logic operators

The following operators can be used:

Logical (Boolean) operators

The logical operators work only on actuators and resultants of type `bool`.

Operator	Description	Syntax example
AND	Logical AND operator. 1 if all source signals are 1.	Destination := Source1 AND Source2 ... AND Source8
OR	Logical OR operator. 1 if at least one of the source signals is 1.	Destination := Source1 OR Source2 ... OR Source8
XOR	Logical exclusive OR operator. 1 if odd number of source signals are 1.	Destination := Source1 XOR Source2 ... XOR Source8
NOT	Inverted value	Destination := NOT Source

Arithmetic (integer) operators

The arithmetic operators work only on actuators and resultants of type `INT32`.

Operator	Description	Syntax example
+	Add integer value.	Destination := Source1 + Source2
-	Subtract integer value.	Destination := Source1 - Source2
*	Multiply integer value.	Destination := Source1 * Source2
/	Divide integer value. See also the operator DIV. **Note** Division by zero is undefined. If the denominator is zero, then the quotient keeps its current value.	Destination := Source1 / Source2
<, <=, ==, >, >=, !=	Compares two integer values and produces a Boolean result. (smaller than, smaller/equal, equal, larger than, larger/equal, not equal)	Destination := Source1 < Source2 Destination := Source1 != Source2 ...
ABS	Absolute value of integer.	Destination := ABS Source
COPY	Copy integer value. Copy integer value from source to up to eight destination signals.	COPY(Source, Destination1, Destination2,...,Destination8)

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Complex operators

These operators use a mix of `bool` and `INT32` actuators and resultants. For a more detailed description, see *Description of the complex operators on page 80*.

Operator	Description
COUNT	Counts number of pulses.
DELAY	Timer
DECODE4LOW	Integer to Boolean converter. Decodes the binary value when for example selecting a tool.
DECODE4HIGH	Integer to Boolean converter. Decodes the binary value when for example selecting a tool.
DIV	Divides two integer values with boolean error flag when dividing with zero. **Note** Division by zero is undefined. If the denominator is zero, then the quotient keeps its current value.
EDGE	Filters signal on edge. 0 when trigger signal is 0. Stays 0 until trigger signal is set to 1 followed by the reset signal set to 1.
MUX	Multiplexer between two integer values.
REM	Remainder, modulo, after division with boolean error flag when dividing with zero.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

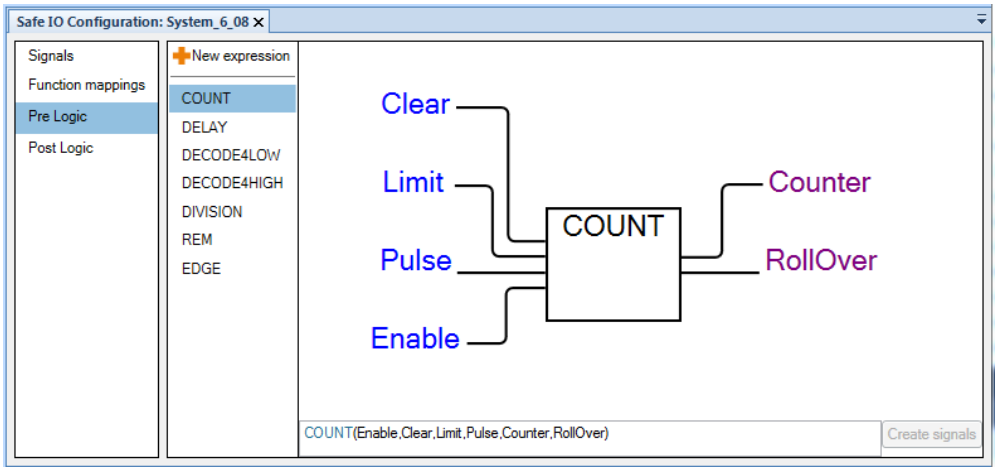
4.6.4.1 Description of the complex operators

Operator COUNT

This operator counts pulses on an actuator. If the Boolean actuator `enable` is inactive then this operator does nothing. The integer resultant `count` retains its current value.

If `enable` is active then `count` counts the number of inactive-to-active transitions of the Boolean actuator `pulse` since the last active-to-inactive transition of `clear`.

The integer resultant `count` is limited by the value of the integer actuator `limit`. When `count` reaches `limit`, then `count` will restart from 0 (zero) on the next activation of the Boolean actuator `pulse`. When the resultant `count` is restarted from zero the `rollover` resultant is activated and then deactivated on the following execution.



xx1800002598

Operator DELAY

The delay operator is similar to the count operator, but instead of counting pulses on an actuator it counts execution loops.

If the Boolean actuator `enable` is inactive then this operator does nothing. The integer resultant `count` retains its current value.

If the Boolean actuator `reset` is activated (and `enable` is active) then the integer resultant `count` is set to 0 (zero) and the Boolean resultant `rollover` is inactivated.

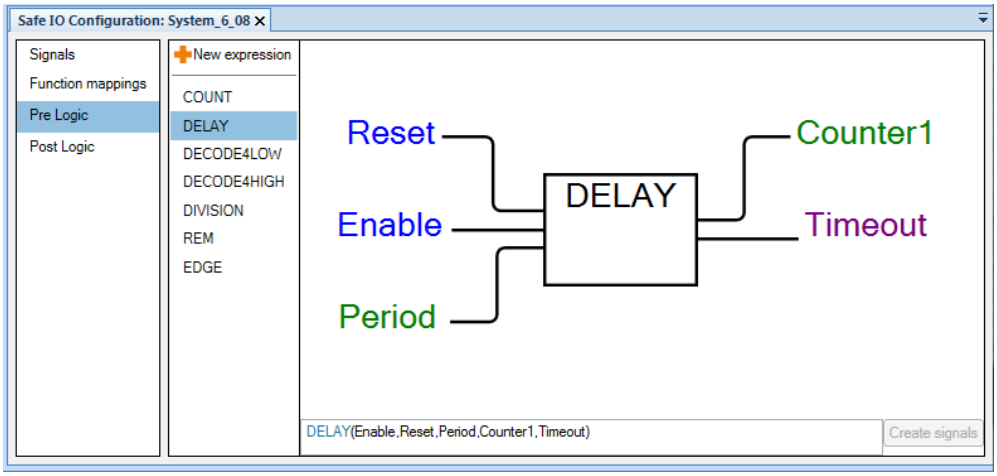
While `enable` is active, and `reset` is inactive, then the integer resultant count counts the number of executions loops since the last active-to-inactive transition of `reset`.

The integer resultant `count` is limited by the value of the integer actuator `period`. When `count` reaches `period`, then `count` will restart from 0 (zero) on the next

Continues on next page

© Copyright 2020 ABB. All rights reserved.

execution. When the resultant `count` is restarted from zero and `rollover` is activated until the next execution.



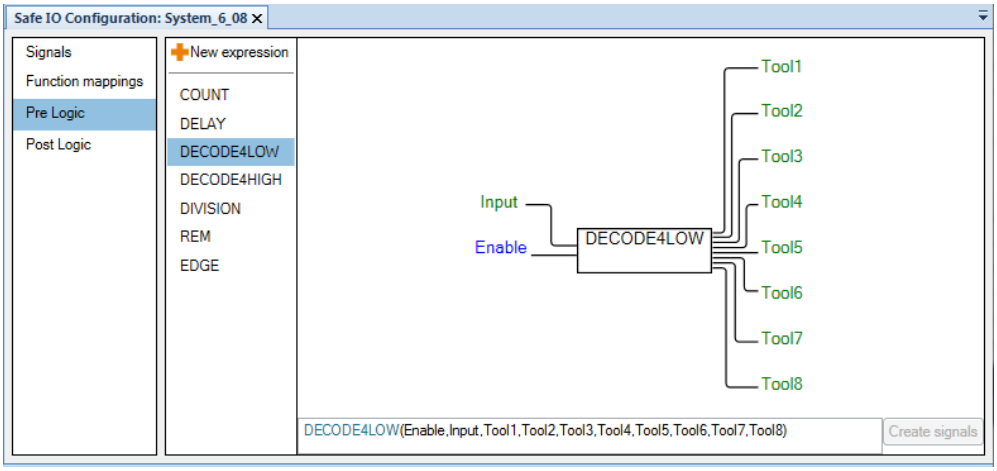
xx1800002599

Operator DECODE4LOW

Decodes the thee least significant bits of an integer actuator into eight individual Boolean resultants if the fourth least significant bit of the integer actuator `input` is low. I.e. for integer values between 0 and 7, (0xxx).

Maximum one output will ever be activated at the same time.

DECODE4LOW works in conjunction with the DECODE4HIGH operator to create a 4 bits-to-16 decoder.



xx1800002600

Operator DECODE4HIGH

Decodes the thee least significant bits of an integer actuator into eight individual Boolean resultants, if the fourth least significant bit of the integer actuator `input` is high. I.e. for integer values between 8 and 15, (1xxx).

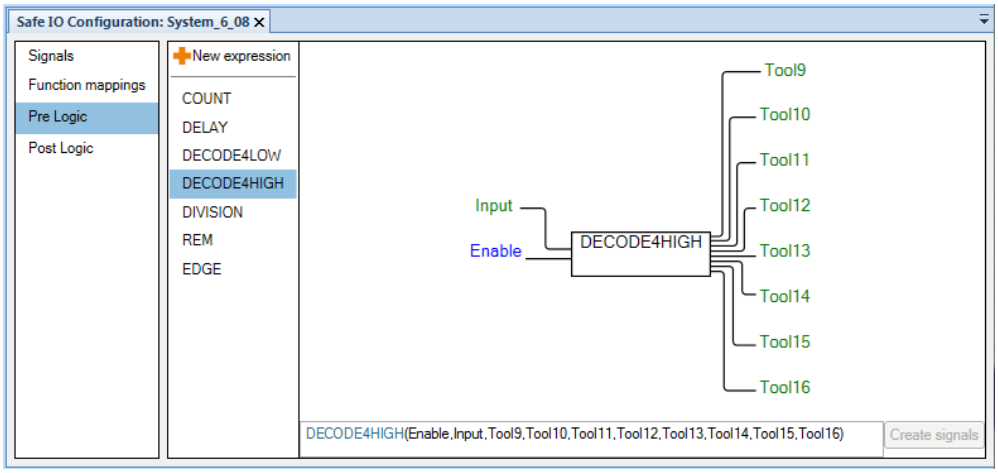
Maximum one output will ever be activated at the same time.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

4.6.4.1 Description of the complex operators
Continued

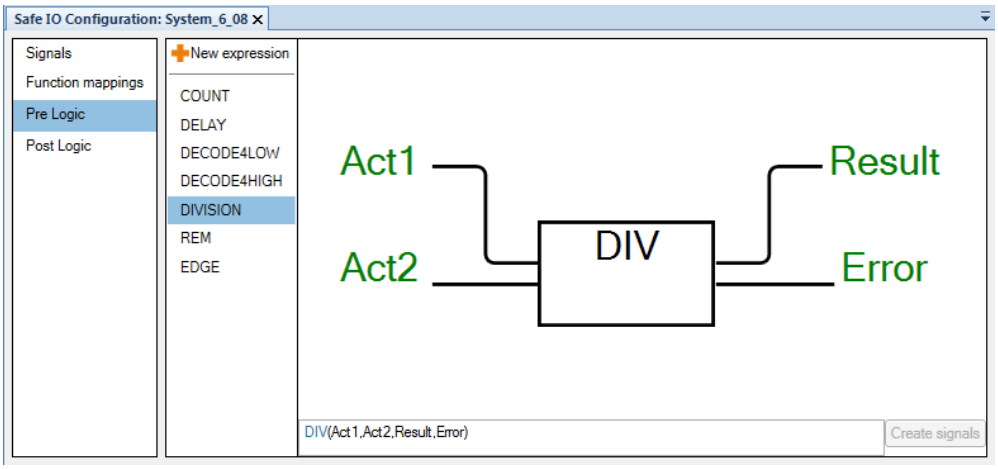
DECODE4HIGH works in conjunction with the DECODE4LOW operator to create a 4 bits-to-16 decoder.



xx1800002601

Operator DIV

This operator divides two integer actuators and produces the quotient as an integer resultant. If the denominator `act2` is zero, then `error` is set and `res1` keeps its current value.



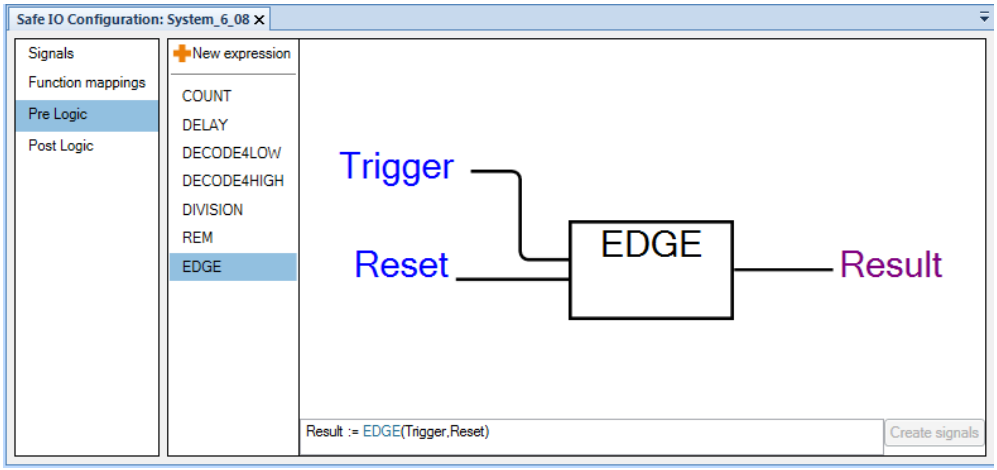
xx1800002602

Continues on next page

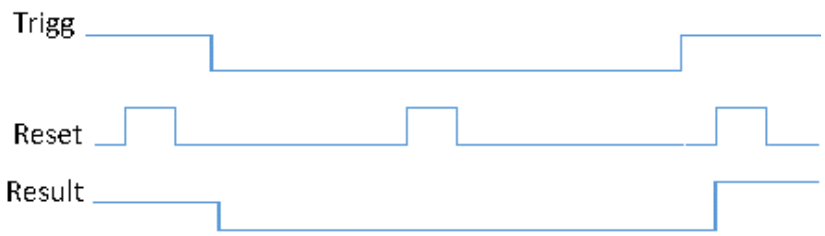
© Copyright 2020 ABB. All rights reserved.

Operator EDGE

The EDGE operator has two Boolean actuators, `trigg` and `reset`, and one Boolean resultant `result`. When `trigg` is inactive then `result` is also inactive. When `trigg` is active then the EDGE operator sets `result` to active upon a transition, from inactive to active, on `reset`.



xx1800002603



xx1700000469

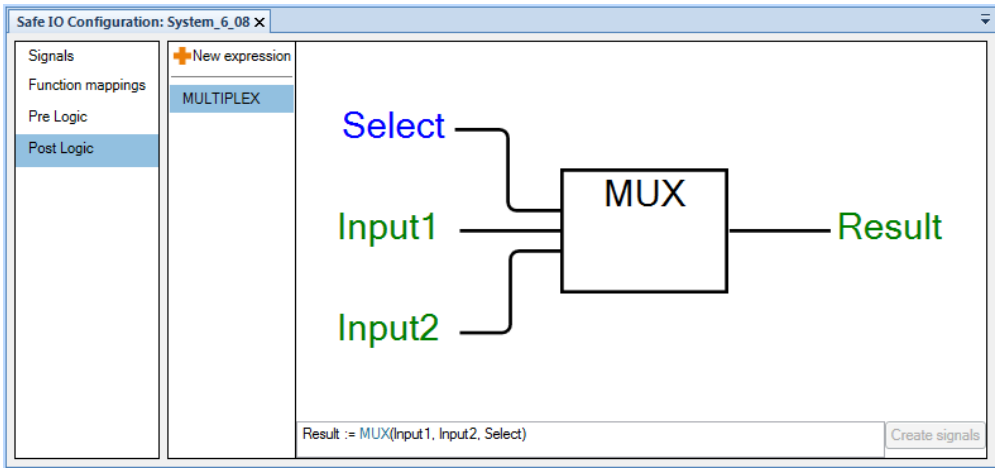
Continues on next page

© Copyright 2020 ABB. All rights reserved.

4.6.4.1 Description of the complex operators
Continued

Operator MUX

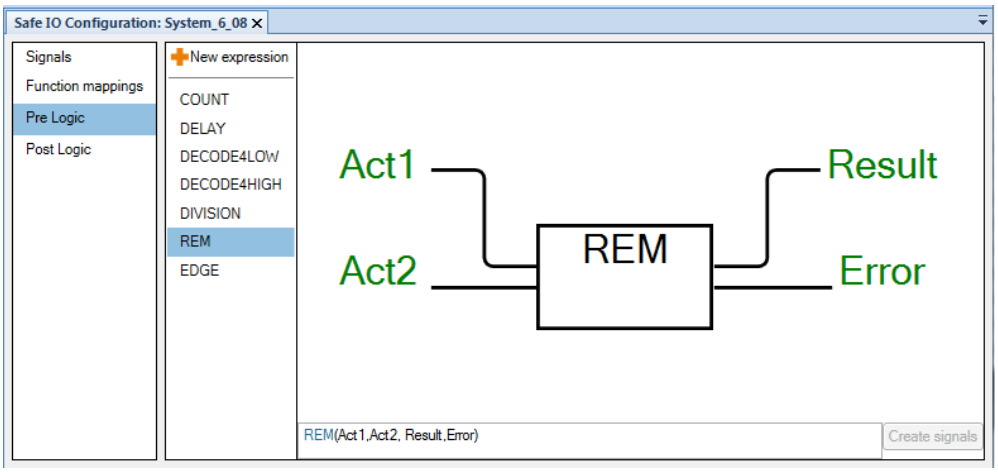
The MUX operator has two integer actuators, `input1` and `input2`, one Boolean actuator `select`, and one integer resultant `result`. When `select` is inactive then `result` obtains the value of `input1`. When `select` is active then `result` obtains the value of `input2`.



xx1800002604

Operator REM

This operator divides two integer actuators and produces the remainder, modulo, as an integer resultant. If the denominator `act2` is zero, then `error` is set and `res1` keeps its current value.



xx1800002605

© Copyright 2020 ABB. All rights reserved.

4.6.5 Safe I/O system rules and limitations

General rules and limitations

- Signal names must be unique.
- Signals must have a name.
- Offsets do not have to have a corresponding signal, for example it is OK to only define offset 1, 7, and 63 as signals.
- Each signal is assigned a default value which is given at startup.
- Each signal, and equivalent alias, can have at most one writer.
- Each signal can have any number of readers.
- Each devices input and output has a limit of 2048 signals.

Rules and limitations of combinatory logic

- A maximum of 100 combinatory logic operators for each level are allowed, not counting *NOT*.
- In some cases the configurator creates operators which are not shown to the user, for example in case of more than 8 operands to an AND function. These additional operators are also counted from the limit of 100.

© Copyright 2020 ABB. All rights reserved.

4.7 Simulating SafeMove

About the Auto Generate From Simulation group

Automatic generation of safety parameters from simulation is a quick and easy way of creating zones and axes ranges based on a simulated path in RobotStudio. The function creates a zone or axes range that encapsulate the recorded simulated path.

A prerequisite is that there has to be at least one SafeMove tool in the configuration, and a simulation has to be defined.

> **Note**
>
> The simulation functions are only available when running a RobotStudio station.

The Record Simulation button

A simulation is setup, started, and stopped using the standard RobotStudio commands. For more information, see *Operating manual - RobotStudio*.

The simulation has to be recorded to be able to automatically generate safe functions from the simulation. Before the simulation is started, the **Record simulation** button in the Visual SafeMove tab must be pressed.

When there is a recorded simulation available, the **Safe Zone**, **Safe Range**, and **Tool Orientation Supervision** buttons become active.

The Safe Zone button

From the simulation it is possible to create a **Tool Position Supervision** and/or a **Tool Speed Supervision**.

The function creates a safe zone that encapsulates the path of the tool, or the tool and elbow, from the recorded simulation.

Function	Description
Tool only	Create a new safe zone that encapsulates the path of the tool from the recorded simulation.
Tool and elbow	Create a new safe zone that encapsulates the path of the tool and elbow from the recorded simulation.

The Safe Range button

From the simulation it is possible to create an **Axis Position Supervision** and/or an **Axis Speed Supervision**.

The function creates a safe range with the max and min values of each joint set to the max and min value from the recorded simulation.

The Tool Orientation Supervision button

From the simulation it is possible to create a **Tool Orientation Supervision**.

The function creates a **Tool Orientation Supervision** where the tolerances around the x and z axis of the tool are calculated from the simulation.

© Copyright 2020 ABB. All rights reserved.

4.8 Protected basic configuration

Introduction

When configuring a large number of robot systems in a factory or a large production line, it is often desired to make the signal configurations similar or identical.

If every system has different signal names and different setups, then it will be very difficult for the operator to understand the systems.

Therefore it is possible to create a basic configuration that can be used as a template when configuring a new system. It is also possible to protect elements in the configuration.

The protected elements checksum

When RobotStudio saves the configuration, a separate checksum is calculated for the protected elements. The protected elements checksum is placed in the checksum protected part of the configuration file.

The customer can verify that the protected basic configuration is unchanged by reading the protected elements checksum.

Only system parts of the configuration can be selected as protected. This includes signals, combinatory logic, and stop functions. The drive module specific parts are not possible to protect, for example zones, ranges, and speed supervision.

When the configuration is loaded into the safety controller, the protected elements checksum is extracted and made available for the safety PLC using a RAPID variable in the same way as the checksum of the complete safety configuration, see *RAPID components on page 153*.

By reading the checksum from RAPID, the safety PLC can compare the checksum with the expected checksum given from RobotStudio when the configuration was created.

Note

There is not any automatic check of the protected elements checksum in the robot system, this must be implemented by the customer.

Creating a protected basic configuration

Use this procedure to create a protected basic configuration:

	Action	Note/illustration
1	Create a basic configuration using Visual SafeMove in RobotStudio.	
2	When the basic configuration is completed, open the **Protected Elements** browser from the **Tools** group.	Only system parts of the configuration can be selected as protected. This includes signals, combinatory logic, and stop functions. The drive module specific parts are not possible to protect, for example zones, ranges, and speed supervision.
3	Protect the elements by ticking the checkboxes in the tree structure.	The write-protected element are visualized with a padlock icon in the **Visual SafeMove** browser.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

	Action	Note/illustration
4	Save the controller configuration.	

Using a protected basic configuration

After the customer has created a basic configuration that should be used as a template in all robots, the configuration should be distributed to the persons creating the robot specific configurations.

Use this procedure to import and use a protected basic configuration:

	Action	Note/illustration
1	Open, or import, the protected basic configuration into Visual SafeMove in RobotStudio.	
2	Add the robot specific parts of the configuration.	The write-protected elements are visualized with a padlock icon in the **Visual SafeMove** browser.
3	Continue with the configuration by adding the safety supervision parts.	
4	Save, download, and verify the safety configuration.	
5	Make sure that the safety PLC is programmed with the protected elements checksum. The same checksum is used for all robots using the protected basic configuration.	
6	By reading the checksum from RAPID, the safety PLC can compare the checksum with the expected checksum given from RobotStudio when the basic configuration was created.	

Note

A user can change protected elements if he deselects them in the **Protected Elements** browser. However, this will affect the protected elements checksum. Therefore it is necessary to verify the protected elements checksum for each system.

© Copyright 2020 ABB. All rights reserved.

5 The FlexPendant user interface

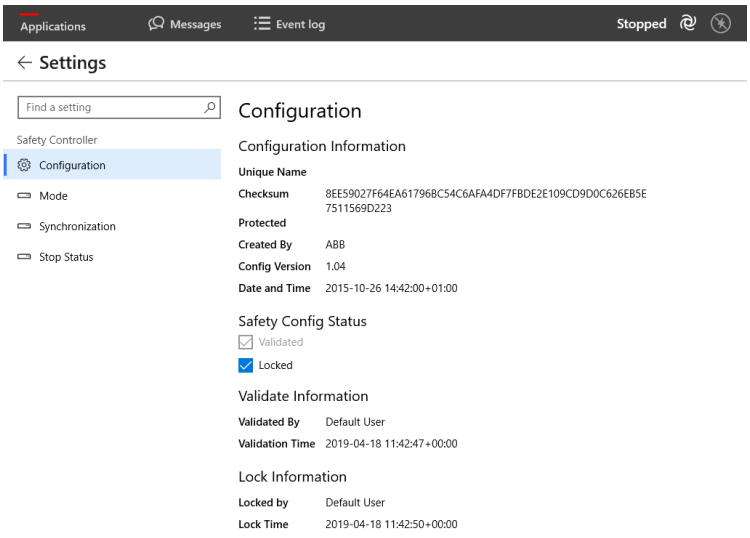
5.1 The Safety Controller control panel

Introduction

This section gives an overview of the **Safety Controller** control panel on the FlexPendant. Views, buttons, and other parts of the user interface are described in respect to their content and how they are accessed.

For more information on how to use the FlexPendant in general, see *Operating manual - OmniCore*.

The safety configuration can be viewed from the **Settings** app.



xx1900000976

User authorization (UAS)

The following UAS settings related to the safety controller are available for setting the user access on the FlexPendant.

For more information about creating user grants, see *Set up safety user grants on page 97*.

User grant	Description
Lock Safety Controller Configuration	Required for locking the safety configuration.
Safety Services	Required for the following actions: • Loading a safety configuration. • Setting the safety configuration status to **Validated**. • Changing the operating mode of the safety controller from the **Mode** tab.
Software synchronization	Required for synchronizing the controller from the **Synchronization** tab.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

The Configuration tab

The **Configuration** tab shows the status of the safety configuration file. From the configuration tab it is possible to view the safety configuration, validate, and lock the configuration file.

For more information, see *Validate the configuration on page 131*.

Safety Config Status	Description
Validated	When the safety technician has validated the configuration, the status of the configuration is changed to **Validated** on the FlexPendant.
Locked	When the safety responsible has approved the configuration, the status of the configuration is changed to **Locked** on the FlexPendant.

Tip

Tap the checksum to view the complete string in a larger window.

View safety controller data

Click the **View** button to view the safety configuration in a separate window.

The view function shows the safety configuration as it is defined in the xml-format, which means that structure and function names etc. does not have a one-to-one mapping with the configurator or the safety report.

For a more user friendly view, use RobotStudio to view configuration, or create a new report.

The Mode tab

The **Mode** tab is used to change the operating mode of the safety controller.

Mode	Description
Safety Supervision Mode	The safety supervision mode is the default mode where the safety supervision is active.
Service Mode	Service mode is intended to be used during service and commissioning. All stops from the supervision functions are deactivated, so it is possible to jog and run the robot without limitations. The violation output signals from the functions are still enabled. The communication to the safety PLC is active so the configured safety stops are active. Service mode is only allowed in manual mode. If the operating modes manual full speed or automatic mode is selected, then service mode is deactivated and safety supervision mode is activated by the safety controller. The status bar on the FlexPendant displays that the safety controller is in service mode.

The Synchronization tab

The safety controller has to be synchronized with the robot controller before the safety supervision can be used. For more information, see *Configure the synchronization position on page 104* and *Synchronization guidelines on page 155*.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

The **Status** LED has the following indication:

- Green - Synchronized
- Red - Unsynchronized
- Grey - Undefined, no information is present.

Button	Description
Synchronize	Synchronizes the safety controller with the robot controller.

The Stop Status tab

The **Stop Status** tab is only present when a *Stop configuration* is configured in the safety configuration. Up to 8 stop configurations are shown with a green or red LED indicating the status.

For more information, see *The Stop Configuration button on page 63*.

© Copyright 2020 ABB. All rights reserved.

This page is intentionally left blank

6 Configuring SafeMove

6.1 Recommended working procedure

General

This section describes the recommended working procedure when configuring SafeMove for the first time. The working procedure helps to understand the dependencies between the different steps. A good approach when creating a new configuration is to start with the basic functionality. When that works as expected, expand the application.

A prerequisite is that all steps in the hardware and software installation procedures must have been performed, see chapter *Installation on page 51*.

For more information on what can be done with the configuration tool, see chapter *The Visual SafeMove user interface in RobotStudio on page 53*.

Note

Some functionality is only available for SafeMove Pro, see *Functional safety options on page 16*.

Basic steps

Use this procedure when configuring SafeMove.

	Action	See
1	Make some initial preparations.	*Preparations on page 95*
2	Configure system parameters.	*Configure system parameters on page 96*
3	Set the input and output size and name of the PROFINET internal device.	*Application manual - PROFINET Controller/Device*
4	Set up safety user grants.	*Set up safety user grants on page 97*
5	Configure robot properties.	*Configure the robot on page 101*
6	Configure the synchronization position.	*Configure the synchronization position on page 104*
7	Configure the SafeMove tool definitions.	*Configure the tools on page 105*
8	Configure safe I/O signals.	*Configure safe I/O on page 108*
9	Configure zones and/or ranges.	*Configure the zones on page 110* *Configure the ranges on page 113*
10	Configure the supervision functions.	*Configure the supervision functions on page 117*
11	Configure other functions.	*Configure other functions on page 127*
12	Load the configuration to the safety controller.	*Load the configuration to the safety controller on page 130*
13	Restart the robot controller.	
14	Validate the configuration.	*Validate the configuration on page 131*

Continues on next page

© Copyright 2020 ABB. All rights reserved.

6.1 Recommended working procedure

Continued

	Action	See
15	Set the safety configuration to validated and lock it.	*Setting the configuration to validated on page 142* *Setting the configuration to locked on page 142*

© Copyright 2020 ABB. All rights reserved.

6.2 Preparations

Preparations

Experience shows that when starting with a clean system it is good to first configure the robot system and make some initial preparations before configuring SafeMove.

- Create a robot system with the option **SafeMove Basic** or **SafeMove Pro**.
- Define the coordinate systems that should be used in the robot system.
- Create tool data for all needed tools, and define the TCPs, tool loads, payloads, and arm loads.
- Create work object data for all needed fixtures and define them.

© Copyright 2020 ABB. All rights reserved.

6.3 Configure system parameters

About the system parameters

The configuration of system parameters required for a robot system should be made before starting with the safety configuration.

In addition to the system parameters that need to be configured for a robot system without SafeMove, there are a few parameters in topic *Motion* that are specific for SafeMove. These are described in this section.

> **Note**
>
> It is important to restart the robot controller after changing system parameters, before configuring SafeMove.

Type Mechanical Unit

All mechanical units for additional axes that shall be supervised must have the parameters *Activate at Start Up* and *Deactivation Forbidden* set to On. (Supervised mechanical units must always be active.)

Type Arm

If an axis should be excluded from Cyclic Brake Check, set the parameter *Deactivate Cyclic Brake Check for axis* to On.

The maximum working area for additional axes has to be limited according to limitations specified in section *Work area for additional axes on page 26*. This must be taken into consideration when entering the parameters *Upper Joint Bound* and *Lower Joint Bound*. (The parameter values in radians or meters on arm side.)

Type Brake

If Cyclic Brake Check is executed on an additional axis, a lowest safe brake torque must be defined. A 5% margin is added during the test for setting the fail limit. The parameter used is *Max Static Arm Torque* defined in Nm on motor side. A warning limit is set with a higher torque value (depending on the brake).

Type Motion Planner

If reducing the max speed that SafeMove allows in manual mode, the jogging speed of the robot has to be reduced to the same value by changing the parameter *Teach Mode Max Speed*.

The parameter *Use checkpoint limitation in world* enables the robot to limit checkpoint speed in world coordinate system in teach mode. In this way the additional speed from, for example, a track motion is added to the checkpoint speed and the robot speed is reduced. This parameter is useful when combining SafeMove with a robot on track. SafeMove supervises the speed of check points in world coordinate system and if this parameter is not active there is a risk that SafeMove will trigger overspeed error when robot and track is moved simultaneously.

© Copyright 2020 ABB. All rights reserved.

6.4 Set up safety user grants

Why do you need safety users

Configuring SafeMove is normally done initially and then never changed (until the robot is used for a different purpose). It is vital that the safety configuration is not changed by unauthorized personnel. It is therefore recommended to have specific safety users with grants permitting the respective assignments.

Grants specific for SafeMove

User grant	Description
Lock Safety Controller Configuration	Required for locking the safety configuration.
Safety Services	Required for the following actions: • Loading a safety configuration. • Setting the safety configuration status to **Validated**. • Changing the operating mode of the safety controller from the **Mode** tab.
Software synchronization	Required for performing software synchronization from the FlexPendant.

Note

Anyone is allowed to create or view a SafeMove configuration, but only a user with the grant **Safety Services** is allowed to download it to the controller.

Tip

Create different user groups as described in *Operating manual - RobotStudio*. Make sure that one administrator has the grant *Manage UAS settings* and that the regular users (operators, Default user, etc.) do not have any the following grants:

- **Lock Safety Controller configuration**
- **Safety services**
- **Key-less mode selector**
- **Write access to controller**
- **Manage UAS settings**

WARNING

Users must be educated appropriately before given any grants related to safety configuration or operation.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Additional grants that may be required when working with SafeMove

User grant	Description
Remote restart	Required when configuring from RobotStudio (remote restart of the controller is required after downloading a new configuration).
Program debug	Required when configuring from RobotStudio (allows request write access).
Read access to controller disks	Required when loading a configuration, already stored on controller disk, from the FlexPendant.

Example of recommended users

User	Service technician (line builder)	Safety responsible	Safety operator	Supervisor
Safety Services	X	X		
Lock Safety Controller Configuration		X		
Software synchronization			X	
Key-less mode selector				X
Remote restart	X	X		
Program debug	X	X		
Read access to controller disks	X			

© Copyright 2020 ABB. All rights reserved.

6.5 Starting Visual SafeMove

Starting Visual SafeMove

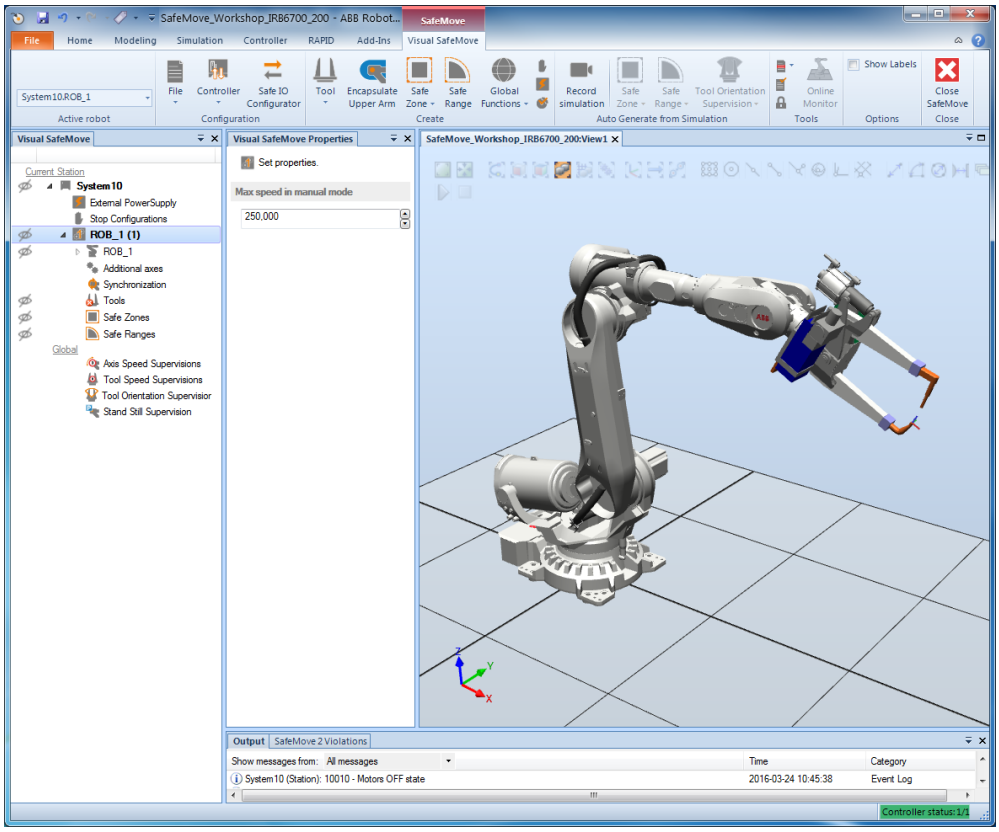
	Action
1	Start RobotStudio with a virtual controller (with or without a station) or connect a real controller.
2	In the **Controller** tab, click **Online Monitor**. (Not needed when running a RobotStudio station.)
3	In the **Controller** tab, click **Safety**, then select **Visual SafeMove**.

© Copyright 2020 ABB. All rights reserved.

6.6 Configure Manual Operation Supervision

Setting the max speed in manual mode

	Action	Note/illustration
1	Click on **ROB_1 (1)** in the SafeMove.	
2	In the SafeMove Properties browser, set **Max speed in manual mode**.	The max speed cannot be higher than the default value of 250 mm/s, but a lower value can be set. See also system parameters *Teach Mode Max Speed* and *Use checkpoint limitation in world* in section *Configure system parameters on page 96*.



xx1600000247

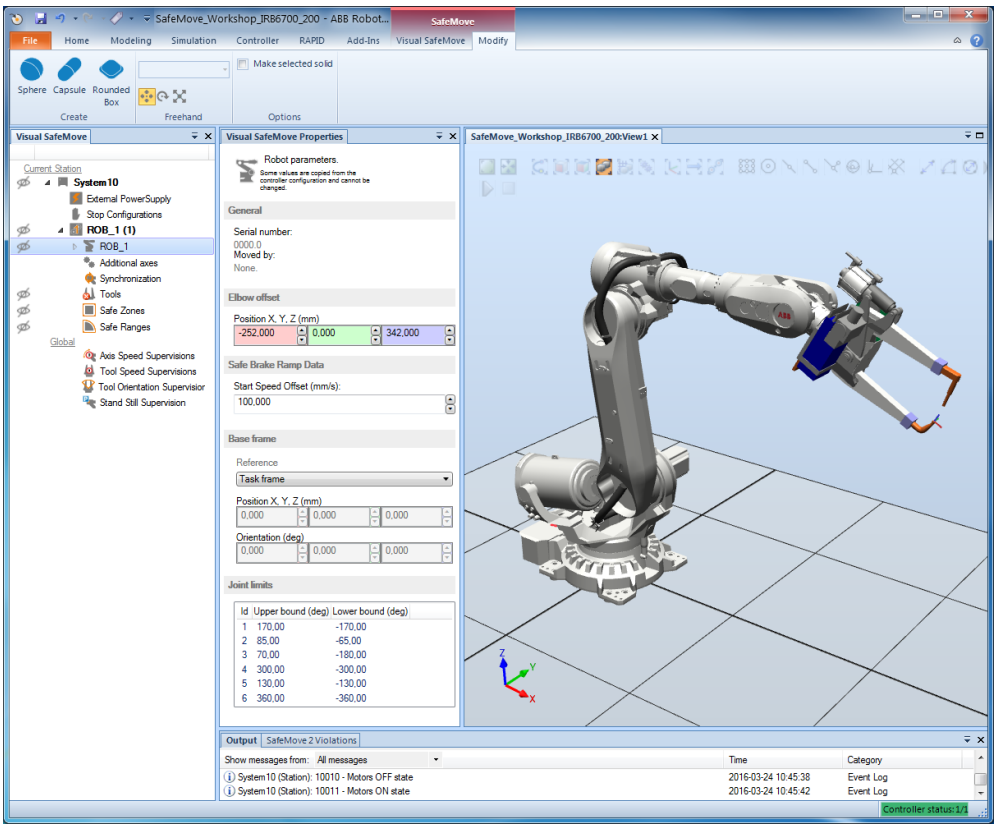
© Copyright 2020 ABB. All rights reserved.

6.7 Configure the robot

Setting the robot properties

	Action	Note/illustration
1	Click on **ROB_1** in the SafeMove browser to specify the robot properties in the SafeMove Properties browser.	
2	Under **Elbow offset**, set the **X**, **Y** and **Z** values for the elbow point.	See *Explanation of Elbow offset on page 69*.
3	Under **Safe Brake Ramp Data**, set the **Start Speed Offset**.	See *Explanation of Safe Brake Ramp on page 71*.
4	Under **Base frame**, select reference co-ordinate system.	See *Explanation of Base Frame on page 70*.

The joint limits are shown here but cannot be changed.



xx1500000538

Encapsulate upper arm

Automatically create encapsulation

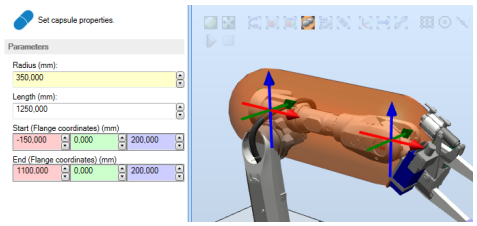
In the SafeMove ribbon, click on **Encapsulate Upper Arm**. This will create a geometry surrounding the upper arm.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

6.7 Configure the robot
Continued

Manually create encapsulation

	Action	Note/illustration
1	In the **Visual SafeMove** browser, select the robot (for example **ROB_1**).	
2	In the **Visual SafeMove** ribbon, click on the **Capsule**.	
3	Click on the shape you have just created to select it.	
4	Fill in the parameters for the shape to specify the position and size of the shape.	 xx1600000129

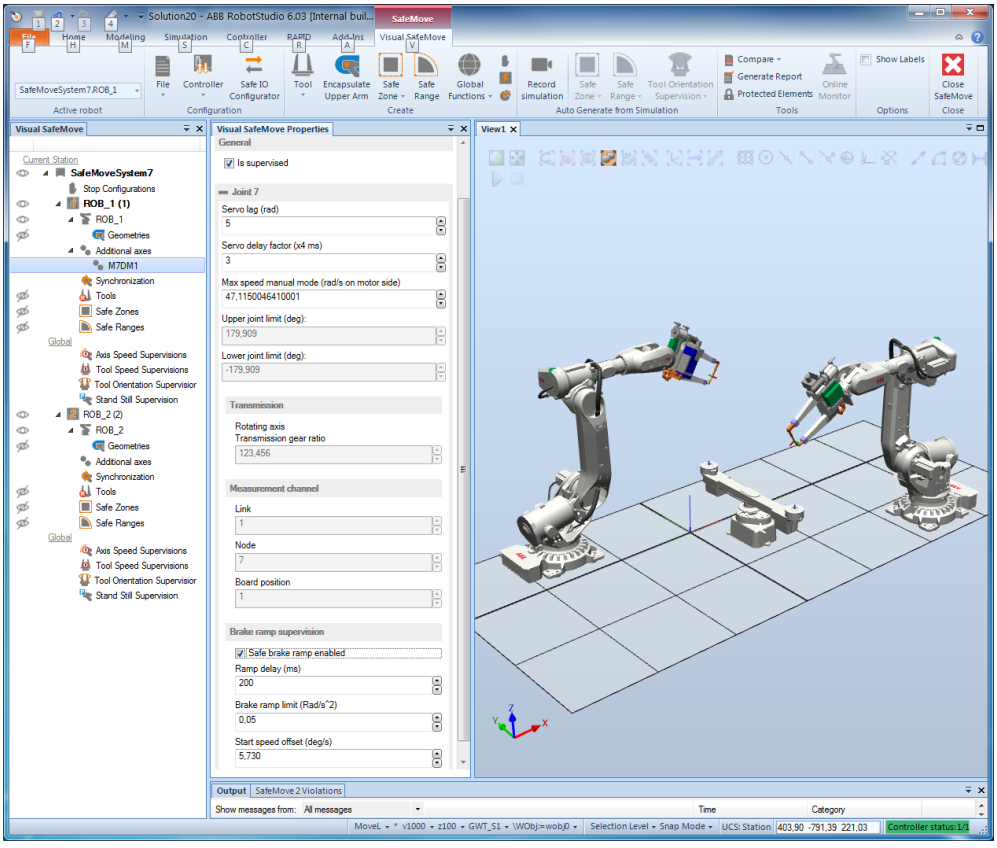
© Copyright 2020 ABB. All rights reserved.

6.8 Configure additional axes

Setting the additional axis properties

	Action	Note/illustration
1	If the axis should be part of the SafeMove supervision, select the check box **Is supervised**.	
2	Specify **Servo lag**.	**Servo lag** is the estimated lag (in radians on motor side) for the additional axis. For more information, see *Servo Delay Factor and Servo Lag on page 165*.
3	Specify **Servo delay factor**.	Estimated delay factor between reference position and measured position (number of 4 ms units) when moving the additional axis. (See TuneMaster, signal number 17 and 18.) For more information, see *Servo Delay Factor and Servo Lag on page 165*.
4	If safe brake ramp should be used for the axis, select the check box **Safe brake ramp enabled**. Set the values for **Ramp delay**, **Brake ramp limit** and **Start speed offset**.	See *Explanation of Safe Brake Ramp on page 71*.

Some information about the additional axis is shown, but cannot be changed. This includes joint limits, transmission gear ratio and measurement channel information.



xx1600000133

© Copyright 2020 ABB. All rights reserved.

6.9 Configure the synchronization position

Setting the synchronization position

	Action	Note/illustration
1	Click on **Synchronization** in the **Visual SafeMove** browser.	
2	Select **Activation**. To use software synchronization only, select **Software synchronization**. To also use hardware synchronization, select the input signal from the sync switch.	Software synchronization is always available even if hardware synchronization is configured.
3	If a status signal should be set when the safety controller is synchronized, select the signal to use in **Synchronization status**. If no output signal should be used, select **No signal**.	Signal definitions: 0 = not synchronized 1 = synchronized
4	Jog the robot to the synchronization position. In the **Visual SafeMove Properties** browser, click on **Read current values**.	It is also possible to specify the axis position values manually.

© Copyright 2020 ABB. All rights reserved.

6.10 Configure the tools

About tool configuration

A SafeMove tool can have one or both of the following:

- Up to 4 shapes surrounding the tool. This is used by the function Tool Position Supervision, which supervises that no part of the shape is outside its allowed zone.
- Up to 8 speed supervision points. The function Tool Speed Supervision always supervises that the TCP and robot elbow do not move faster than allowed speed. By adding speed supervision points, Tool Speed Supervision will supervise these points as well. If speed supervision points surrounds the tool, no part of the tool can move faster than allowed max speed.

Up to 16 SafeMove tools per drive module can be defined, in case a tool changer is used. Input signals are used to specify which tool is used. In order to configure more than one tool in SafeMove, an activation signal for each tool must be specified (if the first tool is set to be permanently active, no new tool can be created).

Orientation of tool shapes

The user can set the orientation of the shapes surrounding the tool using Euler ZYX angles. This rotation can also be found in the safety report.

The same rotation can be described as Euler ZYX angles in multiple ways. For example Euler ZYX angles (90, 90, 0) is the same rotation as (0, 90, -90). The orientation control updates the Euler angles immediately when the user is done editing the orientation. This may cause the Euler angles to change to another representation of the same rotation.

The underlying mathematical representation of the rotation in the controller is in the form of a quaternion which has a unique representation.

Creating a SafeMove tool

Automatically create a SafeMove tool

In the SafeMove ribbon, click on **Tool**, select **Encapsulate** and then select the tool you want to represent with a SafeMove tool. This will create 8 speed supervision points around the tool and also create a geometry surrounding the tool.

> **Note**
>
> The tool must be correctly defined in RobotStudio for the automatic encapsulation to work.

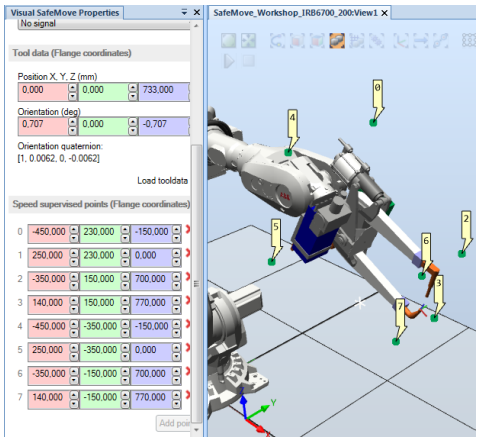
Manually create a SafeMove tool

	Action	Note/illustration
1	In the SafeMove ribbon, click on the **Tool** menu and select **New**.	

Continues on next page

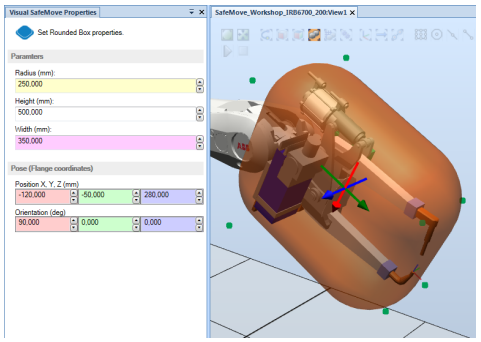
© Copyright 2020 ABB. All rights reserved.

6.10 Configure the tools
Continued

	Action	Note/illustration
2	Select **Activation**. If no tool changer is used, select **Permanently active**. If a tool changer is used, select which input signal should be used to activate this SafeMove tool.	Tool signal definitions: 0 = deactivate tool 1 = activate tool **Note** One, and only one, tool must be active at all times.
3	If a status signal should be set when the tool is active, select the output signal to use in **Function active status**. If no output signal should be used, select **No signal**.	
4	Under **Tool data**, fill in the position and orientation of the tool's TCP in relation to tool0 (the mounting flange).	**CAUTION** Since the tool data is determined by the programmed tool, it is very important that the active tool in SafeMove corresponds to the active tool of the robot program.
5	Specify **Speed supervision points**. Click on **Add point** until you have the number of speed supervision points you want (max. 8). Fill in the X, Y, and Z values for each speed supervised point. A speed supervised point is specified by its X, Y and Z coordinates in the tool0 coordinate systems (mm from the mounting flange).	 xx1500000237
6	In the ribbon, click on a geometry (**Sphere**, **Capsule**, or **Rounded box**).	
7	Click on the shape you have just created to select it.	

Continues on next page

© Copyright 2020 ABB. All rights reserved.

	Action	Note/illustration
8	Fill in the parameters for the shape to specify the position and size of the geometry.	 xx1500000238 **Note** The minimum size of a geometry is 50 mm.

Rename a SafeMove tool

To change the name of a tool, right-click on the tool in the SafeMove browser, select **Rename** and type the desired name.

Change default tool

If more than one tool is configured, one of them must be selected as default tool. The default tool will be used when no tool is selected by input signals.

Select the tool that should be default. Click on the button **Make default Tool**.

Copying a tool

	Action	Note/illustration
1	Right-click on a tool and select **Copy**.	
2	Right-click on the tool again and select **Paste**.	A new tool is created with exactly the same data as the first tool. The parameter values can then be adjusted.

Deleting a tool

	Action
1	Right-click on a tool and select **Delete object**.

© Copyright 2020 ABB. All rights reserved.

6.11 Configure safe I/O

Configure safe signals

Any signals used by the SafeMove functions must be configured before they can be used in the function configurations.

It is important that the signals are configured in the same way for both the host and the device.

Note

The exact approach depends on the type of fieldbus.

	Action
1	In the SafeMove ribbon, click on the **Safe IO Configurator**.
2	Select the **Signals** view.
3	Expand **Input signals**,**Output signals**, or **Global signals**, depending on which type of signal you want to create.
4	Click on an empty line and type the signal name.
5	Set **Default value**.

Note

The usage of a signal cannot be changed from the **Safe IO Configuration**, but in the column **Signal uses** it is shown what functions use each signal.

Tip

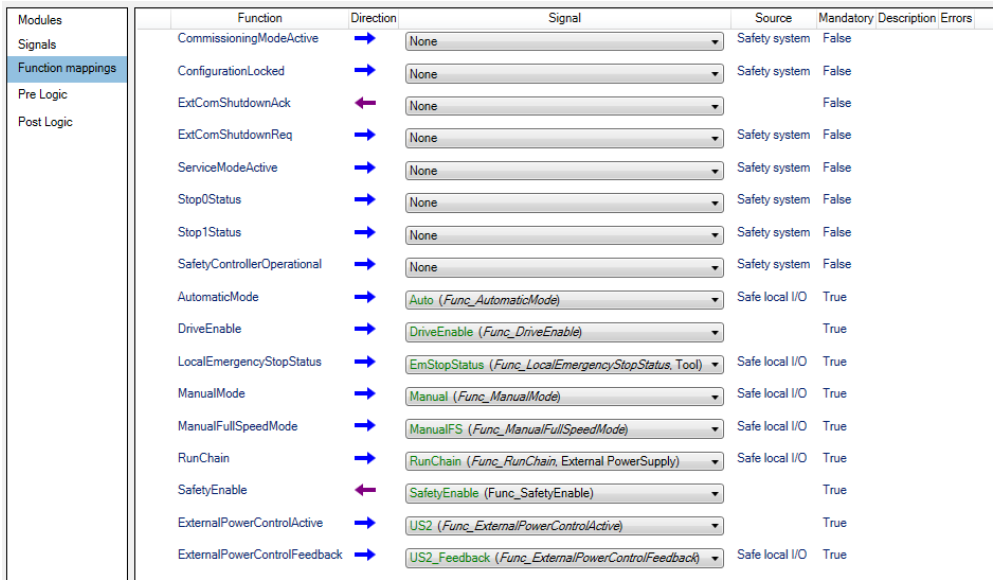
It is possible to use a spreadsheet application, or text editor, to edit the names of the signals and then copy-paste them into RobotStudio.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Configure function mappings

Use the **Function mappings** view of the **Safe IO Configuration** to select which signal to use for each function.



xx1600000139

Configure pre logic and post logic

For more detailed information about configuring combinatory logic including rules and limitations, see *Configuring combinatory logic on page 77* and *Safe I/O system rules and limitations on page 85*.

	Action	Note/illustration
1	Select the **Pre Logic** or **Post Logic** view in **Safe IO Configuration**.	
2	Click on **New expression**.	
3	At the bottom of the **Safe IO Configuration**, type the logical expression.	If the result signal is not already configured, click **Create signal** to automatically define it as a global signal.

Tip

It is recommended to configure the pre logic before configuring the supervision functions.

It is recommended to configure the post logic after configuring the supervision functions. An error message is shown if a signal value is used in a logical expression without having a defined writer.

© Copyright 2020 ABB. All rights reserved.

6.12 Configure the zones

Creating a zone

DANGER

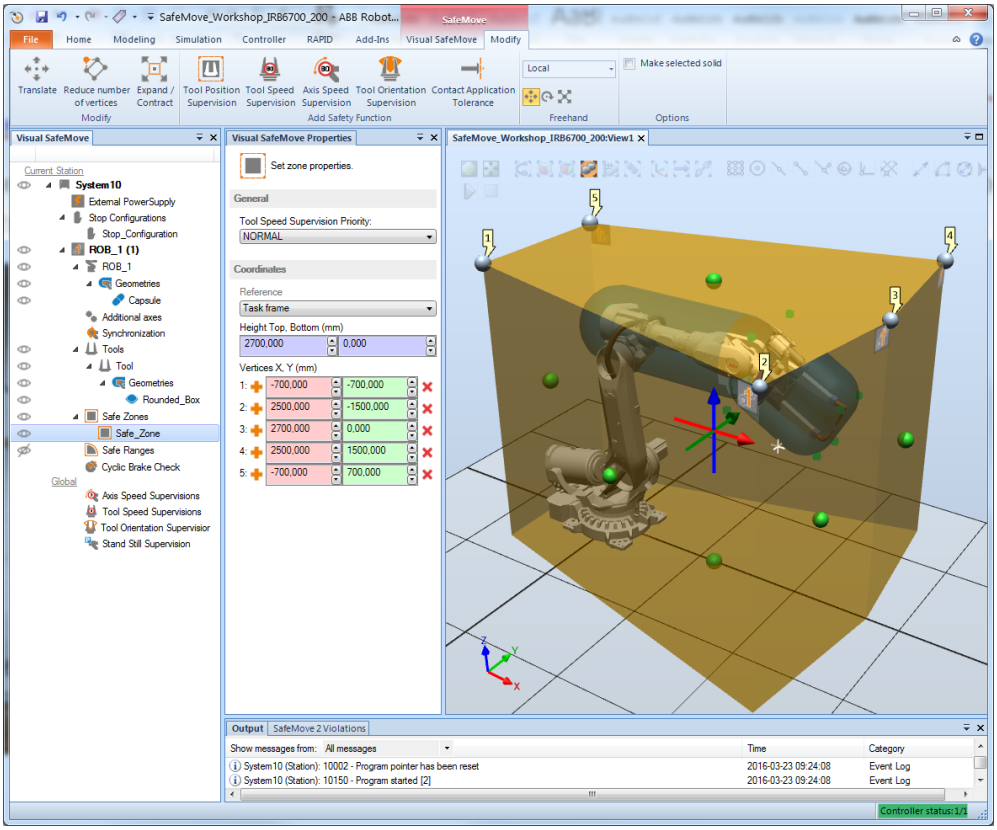
When setting zone limits, brake distances must be taken into consideration, so that the SafeMove functions are configured with enough margin. If the robot hits the zone limit, it starts to brake and needs the brake distance to stop. This occurs outside the allowed zone.

Note that if the robot starts accelerating strongly just before reaching a configured zone limit, there will occur a speed overshoot before decelerating. This may result in a somewhat increased speed and extended braking distance compared to a smoother speed situation.

	Action	Note/illustration
1	In the **Visual SafeMove** ribbon, click on **Safe Zone**.	A zone of default size is shown in the graphics window.
2	Specify **Tool Speed Supervision Priority** for the zone. If two zones are overlapping, the zone with the highest priority will set the speed limits for the overlapping space. If two overlapping zones have the same priority, the most restrictive values will be used in the overlapping space (lowest max speed and highest min speed).	**Tip** If there is a small zone with higher allowed speed (e.g. 1000 mm/s) inside a larger zone with lower allowed speed (e.g. 250 mm/s), use higher priority for the small zone.
3	Fill in the height of the box and the X and Y values for each corner.	If you want to state the coordinates in another coordinate system, select it in the field **Reference**. If the zone base should have another shape than square, add a corner point by clicking on a **+** button. For example, to add a new point between point 2 and 3, click on the **+** button at point 2. An alternative to writing coordinates in the **Visual SafeMove Properties** browser, is to click and drag on the corners or sides. By pressing the key X while dragging, the dragging is done along the X-axis while the Y value remains unchanged. By pressing the key Y while dragging, the dragging is done along the Y-axis while the X value remains unchanged.

Continues on next page

© Copyright 2020 ABB. All rights reserved.



xx1500000535

Tip

To change the name of a zone, right-click on the zone in the **Visual SafeMove** browser, select **Rename** and type the desired name.

Zone supervision functions

About zone supervision functions

Each zone can be used by several zone supervision functions. There can be two Tool Position Supervision functions per zone, but only one function of each type per zone for all other zone supervision functions.

The default names of the functions are the zone name combined with:

- TPO for Tool Position Supervision
- TSP for Tool Speed Supervision
- ASP for Axis Speed Supervision
- TOR for Tool Orientation Supervision
- CAP for Contact Application Tolerance

To change the name of a function, right-click on the function in the **Visual SafeMove** browser, select **Rename** and type the desired name.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

6.12 Configure the zones
Continued

Creating zone supervision functions

	Action	Note
1	In the **Visual SafeMove** browser, click on the zone to select it.	
2	In the **Visual SafeMove** ribbon, click on the function to create.	
3	Configure the function according to *Configure the supervision functions on page 117*.	

© Copyright 2020 ABB. All rights reserved.

6.13 Configure the ranges

Creating a range

DANGER

When setting range limits, brake distances must be taken into consideration, so that the SafeMove functions are configured with enough margin. If the robot hits the range limit, it starts to brake and needs the brake distance to stop. This occurs outside the allowed range.

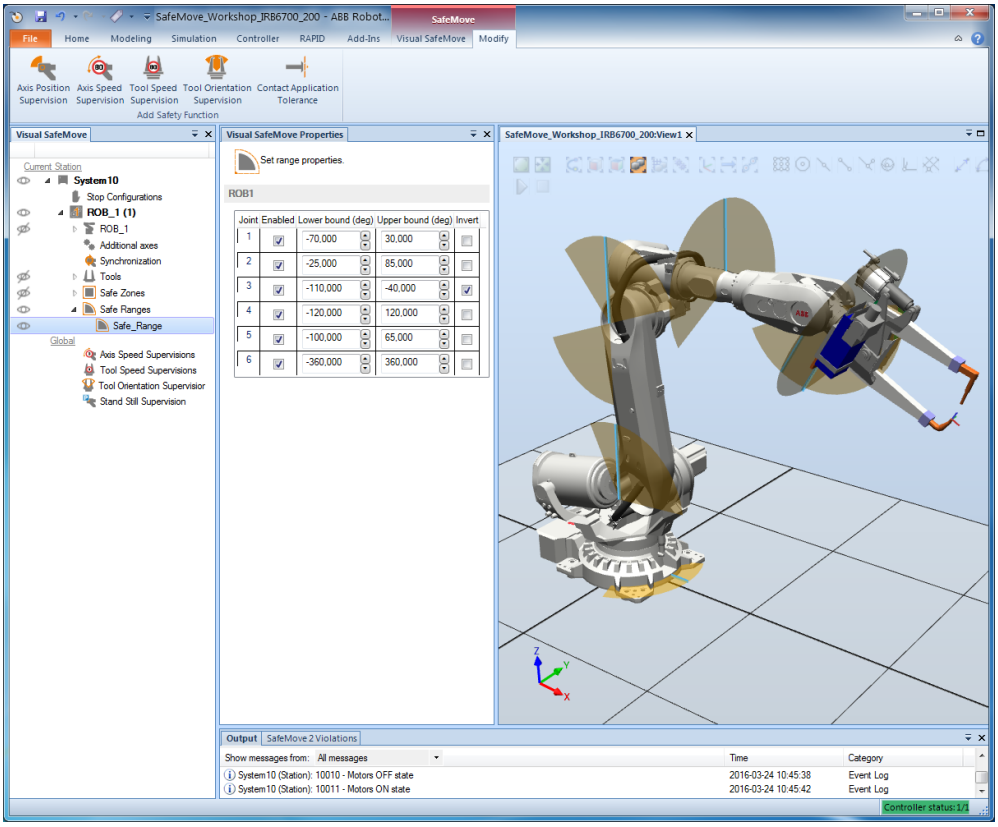
Note that if the robot starts accelerating strongly just before reaching a configured range limit, there will occur a speed overshoot before decelerating. This may result in a somewhat increased speed and extended braking distance compared to a smoother speed situation.

	Action	Note/illustration
1	In the **Visual SafeMove** ribbon, click on **Safe Range**.	A range of default size is shown for each axis in the graphics window.
2	Specify **Upper bound** and **Lower bound** for each axis.	
3	If an axis should be excluded from the safe range, clear the check box **Enabled** for that axis.	
4	To supervise the inverted range for an axis (below **Lower bound** and above **Upper bound**) select the check box **Inverted** for that axis.	

Continues on next page

© Copyright 2020 ABB. All rights reserved.

6.13 Configure the ranges
Continued



xx1600000128

Tip

To change the name of a range, right-click on the range in the **Visual SafeMove** browser, select **Rename** and type the desired name.

Range supervision functions

About range supervision functions

Each range can be used by several range supervision functions, but only one function of each type (for example, there cannot be two Axis Position Supervision functions for the same range).

The default names of the functions are the range name combined with:

- APO for Axis Position Supervision
- ASP for Axis Speed Supervision
- TSP for Tool Speed Supervision
- TOR for Tool Orientation Supervision
- CAP for Contact Application Tolerance

To change the name of a function, right-click on the function in the **Visual SafeMove** browser, select **Rename** and type the desired name.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Creating range supervision functions

	Action	Note
1	In the **Visual SafeMove** browser, click on the range to select it.	
2	In the **Visual SafeMove** ribbon, click on the function to create.	
3	Configure the function according to the descriptions in section *Configure the supervision functions on page 117*.	

© Copyright 2020 ABB. All rights reserved.

6.14 Global supervision functions

About global supervision functions

The default names of the global functions are:

- Global_ASP for Axis Speed Supervision
- Global_TSP for Tool Speed Supervision
- Global_TOR for Tool Orientation Supervision
- SST for Stand Still Supervision

There can be several global functions of the same type. A number is then added to the name, for example Global_ASP_1.

To change the name of a function, right-click on the function in the **Visual SafeMove** browser, select **Rename** and type the desired name.

Creating global supervision functions

	Action	Note
1	In the **Visual SafeMove** browser, make sure that no zone or range is selected.	
2	In the **Visual SafeMove** ribbon, click **Global Functions** and select function.	
3	Configure the function according to *Configure the supervision functions on page 117*.	

© Copyright 2020 ABB. All rights reserved.

6.15 Configure the supervision functions

Configuring Tool Position Supervision

Tool Position Supervision is only used as a zone supervision function.

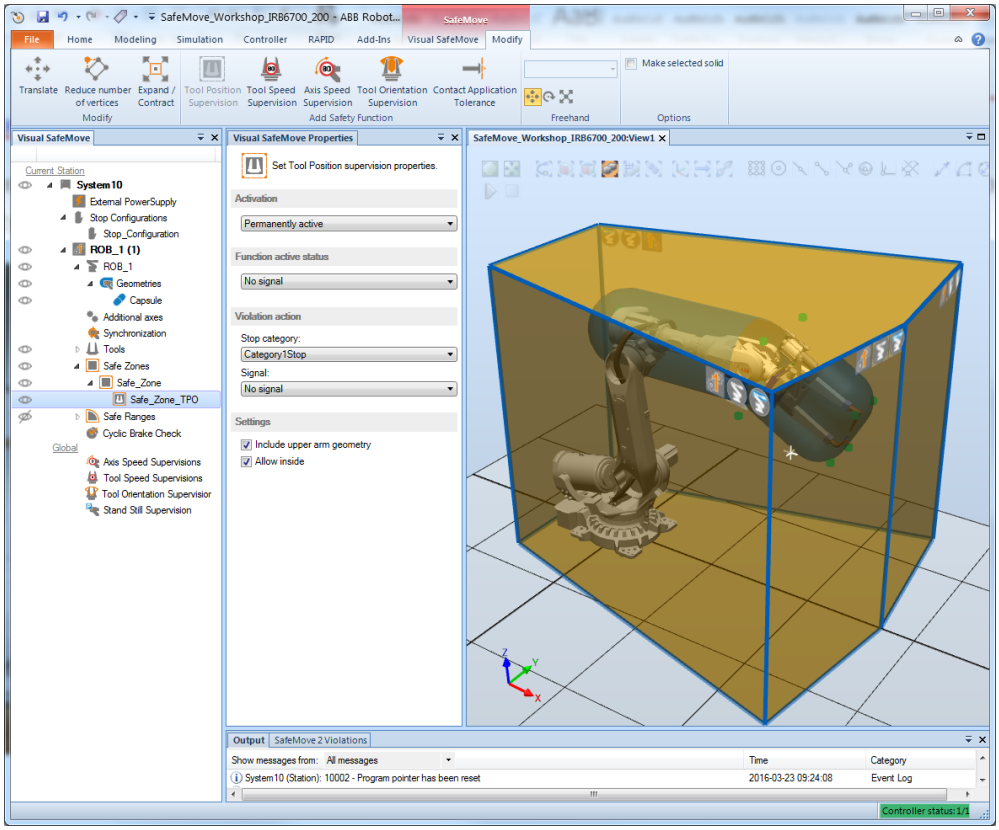
Configure the following settings in the **Visual SafeMove Properties** browser.

	Action	Note
1	Select **Activation**. If no activation signal should be used, select **Permanently active**. If an activation signal should be used, select which input signal should be used to activate this Tool Position Supervision function.	Input signal definitions: 0 = activate function 1 = deactivate function
2	If a status signal should be set when all violation functionality is active, select the signal to use in **Function active status**. If no output signal should be used, select **No signal**.	**Function active status** is a status signal telling if all configured violation actions (e.g. stop and output signal) will be activated at violation. For example, in service mode, some functions have disabled stops and therefore the status signal will not show that all functionality is active. Signal definitions: 0 = at least one violation action disabled 1 = all violation actions enabled
3	In **Stop category**, select if violation of the function should stop the robot with category 0 stop, category 1 stop or not stop the robot.	
4	Select **Signal** to be set at violation. If no signal should be set at violation, select **No signal**.	Note that if **Stop category** is **NoStop**, then **Signal** cannot be **No signal**. Signal definitions: 0 = violation 1 = no violation
5	If only the tool (not the upper arm) should be supervised, clear the check box **Include upper arm geometry**.	
6	If the robot must be inside the zone, select **Allow inside**. If the robot must be outside the zone, clear the check box **Allow inside**.	

Continues on next page

© Copyright 2020 ABB. All rights reserved.

6.15 Configure the supervision functions
Continued



xx1500000536

Configuring Tool Speed Supervision

Tool Speed Supervision can be used as:

- zone supervision function
- range supervision function
- global supervision function

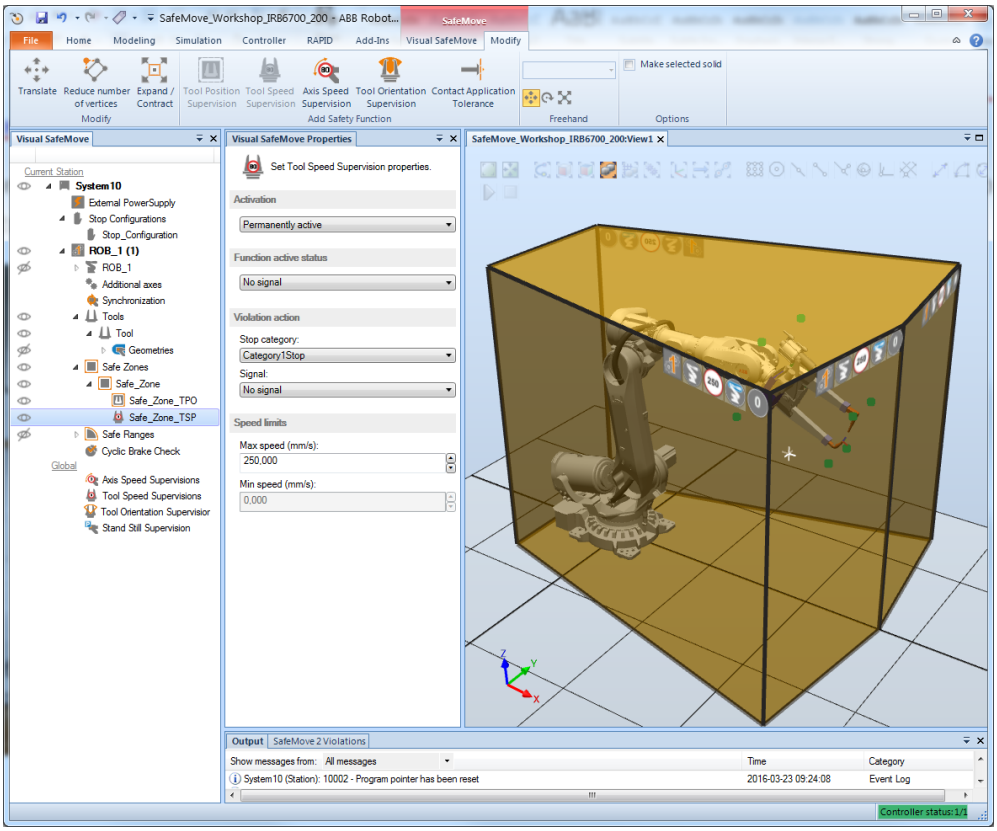
Configure the following settings in the **Visual SafeMove Properties** browser.

	Action	Note/illustration
1	Select **Activation**. If no activation signal should be used, select **Permanently active**. If an activation signal should be used, select which input signal should be used to activate this Tool Speed Supervision function.	Input signal definitions: 0 = activate function 1 = deactivate function
2	If a status signal should be set when all violation functionality is active, select the signal to use in **Function active status**. If no output signal should be used, select **No signal**.	**Function active status** is a status signal telling if all configured violation actions (e.g. stop and output signal) will be activated at violation. For example, in service mode, some functions have disabled stops and therefore the status signal will not show that all functionality is active. Signal definitions: 0 = at least one violation action disabled 1 = all violation actions enabled

Continues on next page

© Copyright 2020 ABB. All rights reserved.

	Action	Note/illustration
3	In **Stop category**, select if violation of the function should stop the robot with category 0 stop, category 1 stop or not stop the robot.	
4	Select **Signal** to be set at violation. If no signal should be set at violation, select **No signal**.	Note that if **Stop category** is **NoStop**, then **Signal** cannot be **No signal**. Signal definitions: 0 = violation 1 = no violation
5	Specify the maximum allowed speed in **Max speed**.	The maximum speed limit must always be larger than the minimum speed limit.
6	If a minimum TCP speed is going to be used, specify the minimum allowed speed in **Min speed**.	If a minimum TCP speed shall be used, the function cannot be both permanently active and stopping the robot. The minimum speed limit has a lower supervision limit of 2 mm/s. The minimum speed limit supervision is disabled by entering a limit of 0 mm/s.



xx1500000537

Configuring Axis Speed Supervision

Axis Speed Supervision can be used as:

- zone supervision function
- range supervision function
- global supervision function

Continues on next page

© Copyright 2020 ABB. All rights reserved.

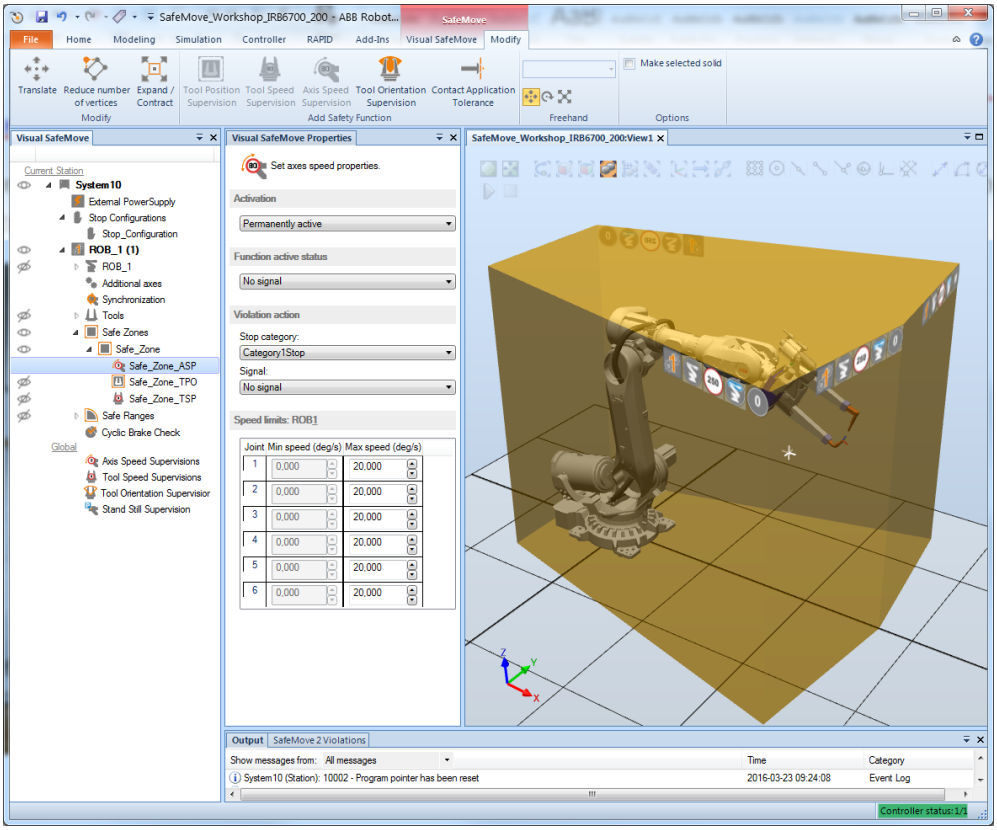
6.15 Configure the supervision functions
Continued

Configure the following settings in the **Visual SafeMove Properties** browser.

	Action	Note/illustration
1	Select **Activation**. If no activation signal should be used, select **Permanently active**. If an activation signal should be used, select which input signal should be used to activate this Axis Speed Supervision function.	Input signal definitions: 0 = activate function 1 = deactivate function
2	If a status signal should be set when all violation functionality is active, select the signal to use in **Function active status**. If no output signal should be used, select **No signal**.	**Function active status** is a status signal telling if all configured violation actions (e.g. stop and output signal) will be activated at violation. For example, in service mode, some functions have disabled stops and therefore the status signal will not show that all functionality is active. Signal definitions: 0 = at least one violation action disabled 1 = all violation actions enabled
3	In **Stop category**, select if violation of the function should stop the robot with category 0 stop, category 1 stop or not stop the robot.	
4	Select **Signal** to be set at violation. If no signal should be set at violation, select **No signal**.	Note that if **Stop category** is **NoStop**, then **Signal** cannot be **No signal**. Signal definitions: 0 = violation 1 = no violation
5	For each axis, specify the minimum speed, **Min Speed**, and maximum speed, **Max Speed**. The values are given for arm side in deg/s for rotating axes and mm/s for linear axes.	If minimum axis speeds shall be used, the function cannot be both permanently active and stopping the robot.

Continues on next page

© Copyright 2020 ABB. All rights reserved.



xx1600000130

Configuring Tool Orientation Supervision

Tool Orientation Supervision can be used as:

- zone supervision function
- range supervision function
- global supervision function

Configure the following settings in the **Visual SafeMove Properties** browser.

	Action	Note/illustration
1	Select **Activation**. If no activation signal should be used, select **Permanently active**. If an activation signal should be used, select which input signal should be used to activate this Tool Orientation Supervision function.	Input signal definitions: 0 = activate function 1 = deactivate function
2	If a status signal should be set when all violation functionality is active, select the signal to use in **Function active status**. If no output signal should be used, select **No signal**.	**Function active status** is a status signal telling if all configured violation actions (e.g. stop and output signal) will be activated at violation. For example, in service mode, some functions have disabled stops and therefore the status signal will not show that all functionality is active. Signal definitions: 0 = at least one violation action disabled 1 = all violation actions enabled

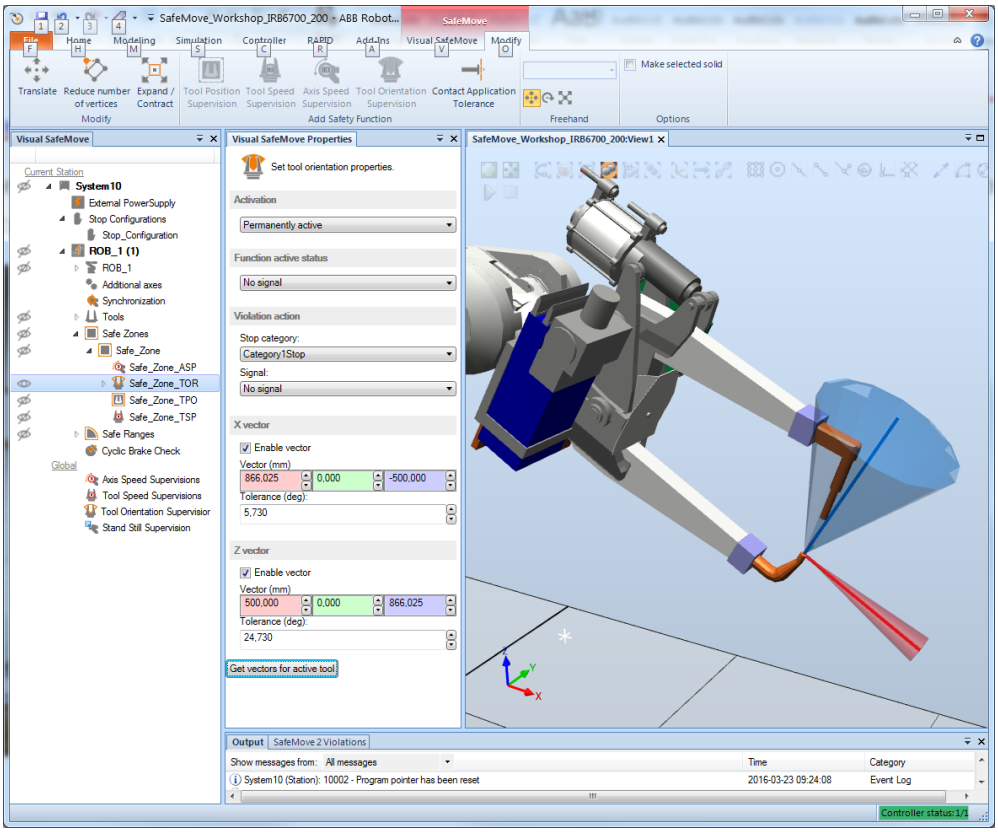
Continues on next page

© Copyright 2020 ABB. All rights reserved.

6.15 Configure the supervision functions

Continued

	Action	Note/illustration
3	In **Stop category**, select if violation of the function should stop the robot with category 0 stop, category 1 stop or not stop the robot.	
4	Select **Signal** to be set at violation. If no signal should be set at violation, select **No signal**.	Note that if **Stop category** is **NoStop**, then **Signal** cannot be **No signal**. Signal definitions: 0 = violation 1 = no violation
5	Jog the robot so that the tool is in desired orientation and click on **Get vectors for active tool**. This will set both X and Z vector for the tool orientation. or Manually specify the tool's X and Z vectors in the world coordinate system.	To exclude supervision for either X vector or Z vector, clear the check box **Enable vector**. **CAUTION** Since the tool data is determined by the programmed tool, it is very important that the active tool in SafeMove corresponds to the active tool of the robot program.
6	Specify the **Tolerance** for both the X vector and the Z vector.	The allowed orientation of X and Z are shown as cones in the graphics window.



xx1600000127

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Configuring Axis Position Supervision

Axis Position Supervision is only used as a range supervision function.

Configure the following settings in the **Visual SafeMove Properties** browser.

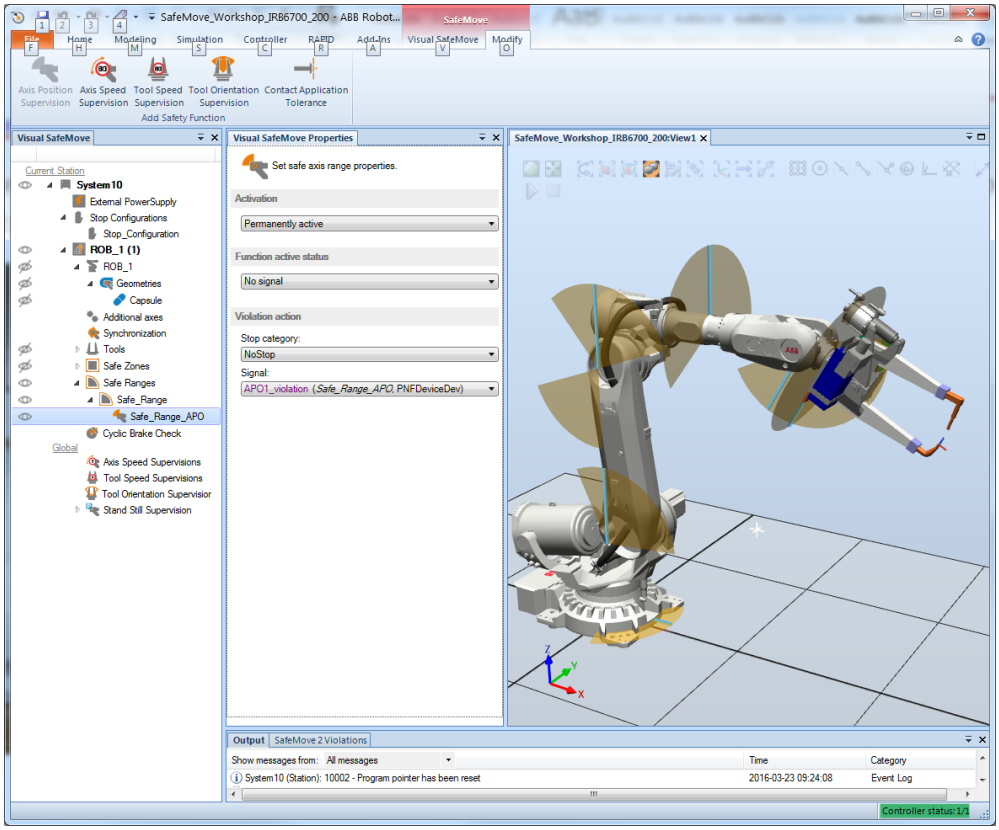
	Action	Note
1	Select **Activation**. If no activation signal should be used, select **Permanently active**. If an activation signal should be used, select which input signal should be used to activate this Axis Position Supervision function.	Input signal definitions: 0 = activate function 1 = deactivate function
2	If a status signal should be set when all violation functionality is active, select the signal to use in **Function active status**. If no output signal should be used, select **No signal**.	**Function active status** is a status signal telling if all configured violation actions (e.g. stop and output signal) will be activated at violation. For example, in service mode, some functions have disabled stops and therefore the status signal will not show that all functionality is active. Signal definitions: 0 = at least one violation action disabled 1 = all violation actions enabled
3	In **Stop category**, select if violation of the function should stop the robot with category 0 stop, category 1 stop or not stop the robot.	
4	Select **Signal** to be set at violation. If no signal should be set at violation, select **No signal**.	Note that if **Stop category** is **NoStop**, then **Signal** cannot be **No signal**. Signal definitions: 0 = violation 1 = no violation

Continues on next page

© Copyright 2020 ABB. All rights reserved.

6.15 Configure the supervision functions

Continued



xx1600000131

Configuring Contact Application Tolerance

Contact Application Tolerance can be used as:

- zone supervision function
- range supervision function

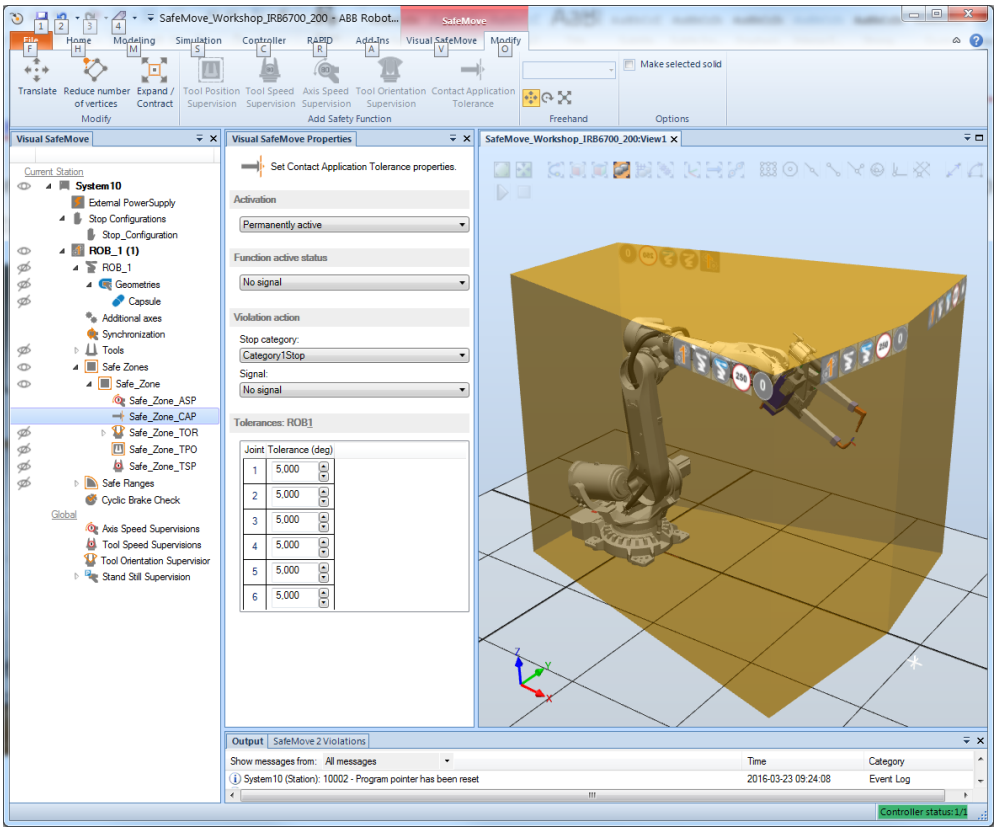
Configure the following settings in the **Visual SafeMove Properties** browser.

	Action	Note/illustration
1	Select **Activation**. If no activation signal should be used, select **Permanently active**. If an activation signal should be used, select which input signal should be used to activate this Contact Application Tolerance function.	Input signal definitions: 0 = activate function 1 = deactivate function
2	If a status signal should be set when all violation functionality is active, select the signal to use in **Function active status**. If no output signal should be used, select **No signal**.	**Function active status** is a status signal telling if all configured violation actions (e.g. stop and output signal) will be activated at violation. For example, in service mode, some functions have disabled stops and therefore the status signal will not show that all functionality is active. Signal definitions: 0 = at least one violation action disabled 1 = all violation actions enabled

Continues on next page

© Copyright 2020 ABB. All rights reserved.

	Action	Note/illustration
3	In **Stop category**, select if violation of the function should stop the robot with category 0 stop or category 1 stop.	
4	Select **Signal** to be set at violation. If no signal should be set at violation, select **No signal**.	Signal definitions: 0 = violation 1 = no violation
5	Under **Tolerances**, specify how much deviation from ordered position that is tolerated for each axis.	Set as low tolerances as the applications permit, to keep the deviations as small as possible.



xx1600000135

Configuring Stand Still Supervision

Stand Still Supervision can be used as:

- global supervision function

Configure the following settings in the **Visual SafeMove Properties** browser.

	Action	Note/illustration
1	Select **Activation**. If an activation signal should be used, select which input signal should be used to activate this Safe Stand Still function. If no activation signal should be used, select **Permanently active**, but then **Stop category** must be set to **NoStop**.	Input signal definitions: 0 = activate function 1 = deactivate function

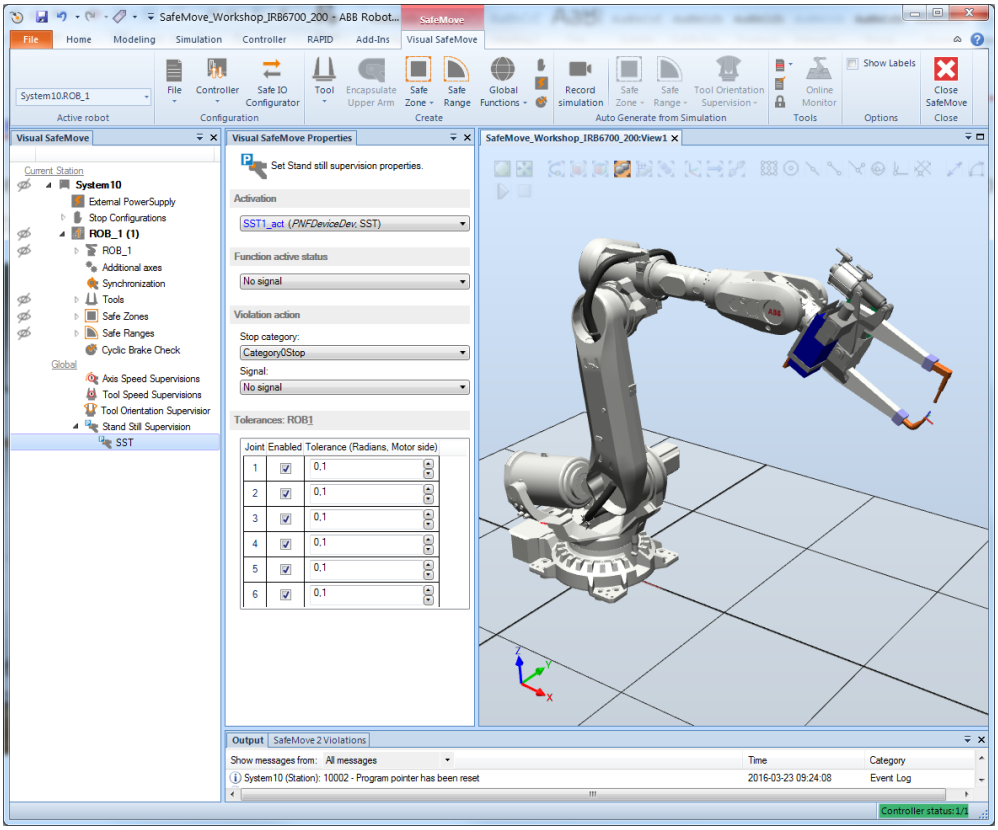
Continues on next page

© Copyright 2020 ABB. All rights reserved.

6.15 Configure the supervision functions

Continued

	Action	Note/illustration
2	If a status signal should be set when all violation functionality is active, select the signal to use in **Function active status**. If no output signal should be used, select **No signal**.	**Function active status** is a status signal telling if all configured violation actions (e.g. stop and output signal) will be activated at violation. For example, in service mode, some functions have disabled stops and therefore the status signal will not show that all functionality is active. Signal definitions: 0 = at least one violation action disabled 1 = all violation actions enabled
3	In **Stop category**, select if violation of the function should stop the robot with category 0 stop or not stop the robot.	Note that if **Activation** is **Permanently active**, then **Stop category** must be **NoStop**.
4	Select **Signal** to be set at violation. If no signal should be set at violation, select **No signal**.	Note that if **Stop category** is **NoStop**, then **Signal** cannot be **No signal**. Signal definitions: 0 = violation 1 = no violation
5	If an axis should be omitted from the supervision, clear the check box under **Enabled** for that axis.	
6	For each axis, specify the maximum allowed tolerance for that axis (in radians on axis side).	



xx1600000132

© Copyright 2020 ABB. All rights reserved.

6.16 Configure other functions

Configuring signal triggered stop

	Action	Note/illustration
1	In the **Visual SafeMove** ribbon, click **Stop Configuration**.	
2	In **Trigger signal**, select the signal that should trigger the stop.	Trigger signal definitions: 0 = activate stop 1 = deactivate stop
3	If a status signal should be set when the functionality is active, select the signal to use in **Stop trigger status**. If no output signal should be used, select **No signal**.	**Stop trigger status** is a status signal telling if the configured stop has triggered. Signal definitions: 0 = stop triggered 1 = stop not triggered
4	In **Mode**, select if it should be: • **SC_AutoStop** - puts the robot controller in auto stop mode (AS). • **SC_EmergencyStop** - puts the robot controller in emergency stop mode (ES).	
5	In **Stop category**, select if the function should stop the robot with category 0 stop or category 1 stop.	

Configuring Cyclic Brake Check

	Action	Note/illustration
1	In the **Visual SafeMove** ribbon, click **Cyclic Brake Check**.	
2	If the robot should not be stopped when the test interval has elapsed, select the check box **Warning only, no stop**.	
3	In **Max CBC test interval**, set the maximum allowed time (in hours) between brake checks.	A value between 2 and 720 hours.
4	In **Pre warning time**, set how long before the end of the interval a warning should be shown on the FlexPendant.	A value between 1 and 11 hours.
5	Do not change the default value for **Standstill tolerance** unless absolutely necessary.	**Standstill tolerance** is used for Stand Still Supervision during brake test. The motor is in regulation during brake test, and a small movement may be allowed. The size of the allowed movement is specified in **Standstill tolerance** (in radians on motor side). Typical value is 2 radians.
6	Do not change the default value for **Supervision threshold** unless absolutely necessary.	**Supervision threshold** defines the threshold to verify that a brake check has been made.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

6.16 Configure other functions
Continued

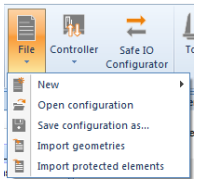
	Action	Note/illustration
7	If one axis should be excluded from the Cyclic Brake Check, clear the check box **Enabled** for that axis.	This must correspond with the axes that has the system parameter *Deactivate Cyclic Brake Check for axis* set to *On*. For axes not included in SafeMove, deactivation of the axes must be done by setting the parameter *Deactivate Cyclic Brake Check for axis* to *On* via RobotStudio for all axes not included.

For more information, see *Cyclic Brake Check guidelines on page 158*.

© Copyright 2020 ABB. All rights reserved.

6.17 Save the configuration

Saving the configuration

	Action	Note/illustration
1	In the **Visual SafeMove** ribbon, click on **File** and then select **Save configuration as**.	 xx1500000802
2	Select a file name and location for the file. Click on **Save**.	

Loading a saved configuration

	Action
1	In the **Visual SafeMove** ribbon, click on **File** and then select **Open configuration**.
2	Browse and select a file. Click on **Open**.

Note

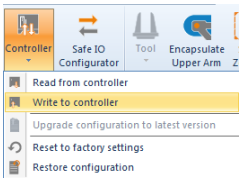
The configuration file can only be edited using RobotStudio. Changing the configuration file in any other way will make the file invalid and it is not possible to run the robot if this file is loaded.

© Copyright 2020 ABB. All rights reserved.

6.18 Load the configuration to the safety controller

6.18.1 Loading a safety configuration using RobotStudio

Writing configuration to the safety controller

	Action	Note/illustration
1	Log in as a user with the grant **Safety Services**.	See *Set up safety user grants on page 97*.
2	In the **Visual SafeMove** ribbon, click on **Controller** and then select **Write to controller**.	 xx1500000801
3	A report of the safety configuration is shown. The report can be printed by clicking on **Print** (it is recommended to print the report since it should be used when validating the configuration as described in *Validate the configuration on page 131*). Click OK to close the report.	
4	Answer **Yes** when asked if you want to restart the controller.	After the restart, the downloaded configuration is active. Before running in auto mode, the configuration should be validated and locked, see *Validate the configuration on page 131*.

Reading the configuration from safety controller

It is possible to upload the configuration from the safety controller to Visual SafeMove. This makes it easy to view the configuration or to make changes to it and download it again.

	Action
1	In the **Visual SafeMove** ribbon, click on **Controller** and then select **Read from controller**.

© Copyright 2020 ABB. All rights reserved.

6.19 Validate the configuration

DANGER

A SafeMove configuration must always be validated to verify that the desired safety is achieved. If no validation is performed, or the validation is inadequate, the configuration cannot be relied on for personal safety.

Tip

Do the following checks before you start the validation procedure:

1. Carry out the synchronization procedure. See *Synchronization guidelines on page 155*.
2. If configured, run the service routine for the function Cyclic Break Check. See *Cyclic Brake Check guidelines on page 158*.
3. Start the validation procedure.

About the validation

Each new or modified safety configuration must be validated before running in production. The validation should verify that the following is configured correctly:

- All I/O settings and signals used for safety interlocking including connected functionality
- All Pre and Post logic used for safety interlocking
- All function mappings used for safety interlocking
- All Stop configuration functions
- All safety ranges with connected supervision functions including connected signals used for safety interlocking
- All safety zones with connected supervision functions and signals used for safety interlocking
- All global supervision functions
- Upper arm encapsulation in combination with Tool Position Supervision (if used)
- All tools with corresponding supervision functions (Tool Position Supervision, Tool Speed Supervision and Tool Orientation Supervision)

Note

Depending on the combination of functions and how they are activated, the validation procedures described below may have to be modified for the specific configuration.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

> **Tip**
>
> Use RAPID programs in order to perform testing faster and to be able to repeat them.

ABB Safety Configuration Report

The validation of each function should be documented in the safety report by signature of the validator.

The safety configuration report lists all parameters that are set for the installation. The report also includes a visual representation of the installation, a floor plan. This shows the robot and safety zones as seen from above.

Recovery after safety violation

The validation procedures test when the safety functions trigger. If the functions Axis Position Supervision, Tool Position Supervision or Tool Orientation Supervision triggers with a stop, recovery is achieved by performing the following:

	Action
1	Select manual operating mode.
2	Press the enable device and jog the robot into a position in which the supervision functions are not in violation.

Validate the safe fieldbus and signal configuration

Validate the safe fieldbus parameters, including I/O settings and signals used for safety interlocking, by comparing the safety report with the configured values.

The user must visually verify that the data in the safety report is correct and that it is the same as entered in the Visual SafeMove configuration GUI.

After controller restart the configuration is applied. The user must verify that no safe fieldbus related event logs were generated, that the status of the connection in the originator indicates "running", and that the connection to the intended adapter device has been established.

Validate range limits

> **Tip**
>
> To validate that the ranges have enough margins, let the robot move with maximum allowed speed when reaching the range limits. Verify that the limits are set with enough margin with respect to the breaking distance.

Validate the range limits for all configured ranges. It is enough to validate the limits of each range for one function, since the range limits are treated equally for all

Continues on next page

© Copyright 2020 ABB. All rights reserved.

functions in the range. Perform the range limits validation using one of the following functions (preferably Axis Position Supervision).

Validate range using Axis Position Supervision

If a range has an Axis Position Supervision function, perform the validation for that function. See *Axis Position Supervision validation on page 139*.

Validate range using Axis Speed Supervision

	Action	Expected result
1	If not permanently active, activate the activation input signal for Axis Speed Supervision for the range. Deactivate all other supervision functions that are signal activated.	
2	Run a RAPID program that moves one configured axis at a speed that is allowed outside the range but not allowed inside the range. At the same time the program shall test one axis at a time by moving it from outside the range to inside the range. Note that the violating speed axis does not have to be the same axis as the one you are testing the range limits for.	Axis Position Supervision will trigger when the tested axis reach its range limit.
3	Repeat this for both lower and upper limit of the axis range.	
4	Repeat this for all axes configured in the range, including additional axes.	

Validate range using Tool Orientation Supervision

	Action	Expected result
1	If not permanently active, activate the activation input signal for Tool Orientation Supervision for the range. Deactivate all other supervision functions that are signal activated.	
2	Jog the robot (reorient jogging) to an orientation that is allowed outside the range but not inside the range.	
3	Move the robot, one axis at a time, from outside the range to inside the range.	Tool Orientation Supervision will trigger when the tested axis reach its range limit.
4	Repeat this for both lower and upper limit of the axis range.	
5	Repeat this for all axes configured in the range, including additional axes.	

Validate range using Tool Speed Supervision

	Action	Expected result
1	If not permanently active, activate the activation input signal for Tool Speed Supervision for the range. Deactivate all other supervision functions that are signal activated.	

Continues on next page

© Copyright 2020 ABB. All rights reserved.

	Action	Expected result
2	Run a RAPID program that moves the tool at a speed that is allowed outside the range but not allowed inside the range. At the same time the program shall test one axis at a time by moving it from outside the range to inside the range.	Tool Speed Supervision will trigger when the tested axis reach its range limit.
3	Repeat this for both lower and upper limit of the axis range.	
4	Repeat this for all axes configured in the range, including additional axes.	

Validate range using Contact Application Tolerance

	Action	Expected result
1	Make sure that Soft Servo is active and set the stiffness to a value that is allowed outside the range but not allowed inside the range.	
2	If not permanently active, activate the activation input signal for Contact Application Tolerance for the range. Deactivate all other supervision functions that are signal activated.	
3	Run a RAPID program moving the TCP with maximum speed allowed (e.g. `vmax`). At the same time the program shall test one axis at a time by moving it from outside the range to inside the range.	Control Error Supervision will trigger when the tested axis reach its range limit.
4	Repeat this for both lower and upper limit of the axis range.	
5	Repeat this for all axes configured in the range, including additional axes.	

Validate zone limits

Tip

To validate that the zones have enough margins, let the robot move with maximum allowed speed when reaching the zone limits. Verify that the limits are set with enough margin with respect to the breaking distance.

Validate the zone limits for all configured zones. It is enough to validate the limits of each zone for one function and one tool, since the zone limits are treated equally for all functions and tools in the zone. Perform the zone limits validation using one of the following functions (preferably Tool Position Supervision).

Validate zone using Tool Position Supervision

If a zone has a Tool Position Supervision function, perform the validation for that function. See *Tool Position Supervision validation on page 137*.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Validate zone using Tool Orientation Supervision

	Action	Expected result
1	If not permanently active, activate the activation input signal for Tool Orientation Supervision for the zone. Deactivate all other supervision functions that are signal activated.	
2	Jog the robot (reorient jogging) to an orientation that is allowed outside the zone but not inside the zone.	
3	Move the TCP from outside the zone to inside the zone.	Tool Orientation Supervision will trigger when the TCP reach the zone limit.
4	Repeat this for at least two points on each side of the zone, including top and bottom.	
5	If the system is equipped with a track motion, check that the tool zone border is in correct position for different positions of the track motion.	

Validate zone using Tool Speed Supervision

	Action	Expected result
1	If not permanently active, activate the activation input signal for the Tool Speed Supervision for the zone. Deactivate all other supervision functions that are signal activated.	
2	Run a RAPID program that moves the tool at a speed that is allowed outside the zone but not allowed inside the zone. The tool shall be moved at this speed from outside the zone to inside the zone.	Tool Speed Supervision will trigger when one of the configured speed supervision points (or the TCP) reach the zone limit.
3	Repeat this for at least two points on each side of the zone, including top and bottom.	
4	If the system is equipped with a track motion, check that the tool zone border is in correct position for different positions of the track motion.	

Validate zone using Axis Speed Supervision

	Action	Expected result
1	If not permanently active, activate the activation input signal for Axis Speed Supervision for the zone. Deactivate all other supervision functions that are signal activated.	
2	Run a RAPID program that moves one configured axis at a speed that is allowed outside the zone but not allowed inside the zone. At the same time the program shall move the TCP from outside the zone to inside the zone.	Axis Position Supervision will trigger when the TCP reach the zone limit.
3	Repeat this for at least two points on each side of the zone, including top and bottom.	
4	If the system is equipped with a track motion, check that the tool zone border is in correct position for different positions of the track motion.	

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Validate zone using Contact Application Tolerance

	Action	Expected result
1	Make sure that Soft Servo is active and set the stiffness to a value that is allowed outside the zone but not allowed inside the zone.	
2	If not permanently active, activate the activation input signal for Contact Application Tolerance for the zone. Deactivate all other supervision functions that are signal activated.	
3	Run a RAPID program moving the TCP with maximum speed allowed (e.g. `vmax`). The tool shall be moved at this speed from outside the zone to inside the zone.	Control Error Supervision will trigger when the reference position of the TCP reach the zone limit.
4	Repeat this for at least two points on each side of the zone, including top and bottom.	
5	If the system is equipped with a track motion, check that the tool zone border is in correct position for different positions of the track motion.	

Validate the tools

If no tool changer is used, the tool geometry, speed supervision points and TCP are validated in the respective functions. If more than one tool is configured, each tool must be validated according to the following.

Note

It is not necessary to validate each tool with each range/zone. It is enough if all ranges/zones are validated with one tool and all tools are validated with one range/zone.

Validate tool geometry

	Action	Expected result
1	If not permanently active, activate the activation input signal for the Tool Position Supervision set to use. Deactivate all other supervision functions that are signal activated.	
2	Move the robot so that the configured geometry surrounding the tool crosses the border of the configured zone. Verify that Tool Position Supervision triggers when the border is crossed.	Tool Position Supervision will trigger.
3	Rotate the tool and repeat the test with different parts of the tool crossing the zone border. Make sure Tool Position Supervision always triggers on the surrounding geometry before the tool itself reach the zone border.	

Validate tool speed supervision point

For each tool, perform the validation for Tool Speed Supervision with that tool. See *Tool Speed Supervision validation on page 138*.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Validate the TCP position

For each tool, validate the TCP position if possible. It is possible to validate the TCP position with the following zone supervision functions:

- If a tool does not have speed supervision points surrounding the TCP, it is possible to validate the TCP position using Tool Speed Supervision. See *Tool Speed Supervision validation on page 138*.
- Tool Orientation Supervision can be used to validate the TCP position. See *Validate zone using Tool Orientation Supervision on page 135*.
- Contact Application Tolerance can be used to validate the TCP position. See *Validate zone using Contact Application Tolerance on page 136*.

If none of these criteria is met for a tool, the TCP position cannot be validated for that tool, but then it is not necessary to validate it.

Note

The TCP for the SafeMove tool must correspond with the TCP of the active tool for jogging and RAPID instructions.

Tool Position Supervision validation

	Action	Expected result
1	If not permanently active, activate the activation input signal for the Tool Position Supervision set you want to validate. Deactivate all other supervision functions that are signal activated.	
2	If used, verify that the corresponding function active status signal is set.	Status signal is 1.
3	Move the robot so that the configured geometry surrounding the tool crosses the border of the configured zone. Verify that Tool Position Supervision triggers when the border is crossed.	Tool Position Supervision will trigger.
4	Repeat this for at least two points on each side of the zone, including top and bottom.	
5	If the system is equipped with a track motion, check that the tool zone border is in correct position for different positions of the track motion.	

Tool Orientation Supervision validation

	Action	Expected result
1	If not permanently active, activate the activation input signal for the Tool Orientation Supervision set you want to validate. Deactivate all other supervision functions that are signal activated.	
2	If applicable, make sure the TCP is inside the Tool Orientation Supervision zone or all axes inside the range.	
3	If used, verify that the corresponding function active status signal is set.	Status signal is 1.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

6.19 Validate the configuration
Continued

	Action	Expected result
4	Jog the robot (reorient jogging) to the tolerance limits of the tool orientation. Verify that Tool Orientation Supervision triggers for violation of both the tool's x direction and the tool's z direction.	Tool Orientation Supervision will trigger.

Tool Speed Supervision validation

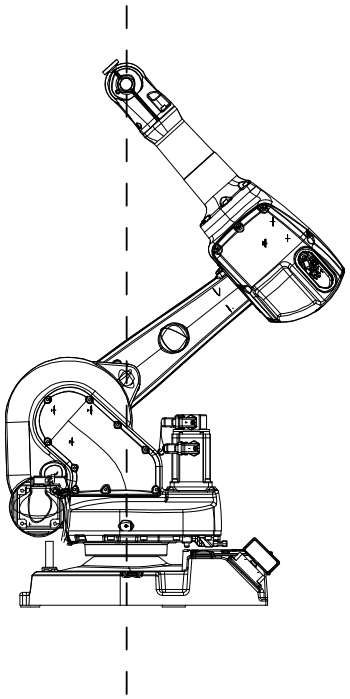
Validate all points supervised by Tool Speed Supervision:

- tool center point (TCP)
- robot elbow (near axis 3 according to configuration)
- any configured speed supervision points for the tool

	Action	Expected result
1	If not permanently active, activate the activation input signal for Tool Speed Supervision. Deactivate all other supervision functions that are signal activated.	
2	If applicable, make sure the TCP is inside the Tool Speed Supervision zone or all axes inside the range.	
3	If used, verify that the corresponding function active status signal is set.	Status signal is 1.
4	Create and run a RAPID program with a `MoveL` instruction. The `Speed` argument should be slightly higher than the configured max speed. The `Tool` argument should be set to the tool that is to be supervised by Tool Speed Supervision.	Tool Speed Supervision will trigger.
5	Test the speed supervision of all configured speed supervision points of the tool. How to do this test depends on the position of the points. Usually a rotation of the tool can make a configured speed supervision point move faster than the TCP. Use `MoveAbsJ`, moving axis 5 or 6, to rotate the tool.	Tool Speed Supervision will trigger.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

	Action	Expected result
6	Jog the robot to a position where the elbow is pointing out as much as possible, while the tool is close to the rotation axis of axis 1.  xx0700000696 Create and run a RAPID program with a `MoveAbsJ` instruction moving axis 1 fast enough for the elbow to exceed the configured max speed.	Tool Speed Supervision will trigger.

Axis Position Supervision validation

	Action	Expected result
1	If not permanently active, activate the activation input signal for the Axis Position Supervision set you want to validate. Deactivate all other supervision functions that are signal activated.	
2	If used, verify that the corresponding function active status signal is set.	Status signal is 1.
3	Move the robot, one axis at a time, from inside the allowed range to outside the range. Verify that Axis Position Supervision triggers when the axis reaches the range limit.	Axis Position Supervision will trigger.
4	Repeat this for both lower and upper limit of the axis range.	
5	Repeat this for all axes configured for Axis Position Supervision, including additional axes.	

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Axis Speed Supervision validation

Tip

To determine which `speeddata` settings to use for validation, use an `MoveAbsJ` instruction to move relevant axis and check the axis speed in signal analyzer in RobotStudio. Change the speed of the `MoveAbsJ` instruction until the desired axis speed is achieved.

	Action	Expected result
1	If not permanently active, activate the activation input signal for Axis Speed Supervision. Deactivate all other supervision functions that are signal activated.	
2	If applicable, make sure the TCP is inside the Axis Speed Supervision zone or all axes inside the range.	
3	If used, verify that the corresponding function active status signal is set.	Status signal is 1.
4	Create a RAPID program with a `MoveAbsJ` instruction moving the first configured axis with a speed slower than the configured Max Speed for that axis. Run the program in auto or manual full speed mode.	No triggered function.
5	Change the program so that the axis is moved with a speed higher than the configured Max Speed. Run the program in auto or manual full speed mode.	Axis Speed Supervision will trigger.
6	Repeat this for all axes configured for Axis Speed Supervision.	

Stand Still Supervision validation

	Action	Expected result
1	If not permanently active, activate the activation input signal for the Stand Still Supervision set you want to validate. Deactivate all other supervision functions that are signal activated.	
2	If used, verify that the corresponding function active status signal is set.	Status signal is 1.
3	Create a RAPID program with `MoveJ` instructions that move one axis at a time at maximum allowed speed. Let the program wait for user input between each move instruction to let you validate each move instruction. If there are additional axes in the system, include instructions that move these axes, one at a time, as well.	
4	Run the program in auto or manual full speed mode and verify that Stand Still Supervision triggers for every move instruction.	Stand Still Supervision will trigger.
5	If the Stand Still Supervision is configured to stop at violation, determine that the movement until the robot is stopped again is within a tolerable limit.	

Continues on next page

© Copyright 2020 ABB. All rights reserved.

	Action	Expected result
6	If the Stand Still Supervision is configured to set an output signal at violation, verify that the signal is 0 when the robot is moving and 1 when the robot is standing still.	

Contact Application Tolerance validation

Contact Application Tolerance only needs to be configured when using Soft Servo, Force Control or when external forces are applied to the robot. It cannot be verified unless it is used in one of these ways.

	Action	Expected result
1	Make sure that Soft Servo is active and set the stiffness to a reduced value (but not lower than needed by the application).	
2	If not permanently active, activate the activation input signal for Contact Application Tolerance. Deactivate all other supervision functions that are signal activated.	
3	Make sure the TCP is inside the Contact Application Tolerance zone or all axes inside the range.	
4	If used, verify that the corresponding function active status signal is set.	Status signal is 1.
5	Run the application and verify that Contact Application Tolerance does not trigger within the range/zone to verify that the tolerance is high enough.	
6	To verify that the tolerance values are as low as possible, change the Soft Servo stiffness to a lower value (e.g. 10% lower). Run a RAPID program moving the TCP with maximum speed allowed in the range/zone (e.g. `vmax`) inside the range/zone for Contact Application Tolerance.	Control Error Supervision will trigger.

Configured stop validation

	Action	Expected result
1	Deactivate all supervision functions that are signal activated.	
2	Move the robot, for example with a move instruction.	
3	Set the signal configured to stop the robot in relevant operating modes. Relevant operating modes are: • **SC_GeneralStop**: All modes • **SC_AutoStop**: Automatic mode • **SC_EmergencyStop**: All modes.	The robot will stop.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

6.19 Validate the configuration
Continued

Setting the configuration to validated

When the safety technician has validated the configuration and signed the safety report, the status of the configuration shall be changed to **Validated** on the FlexPendant.

1. Log in as a user with the grant **Safety Services**.
2. In the **Settings** app, select the **Safety Controller**, and then **Configuration**.
3. Select the check box **Validated**.

Setting the configuration to locked

When the responsible safety user has approved the validation of the configuration, the status of the configuration should be changed to **Locked** on the FlexPendant.

Running the robot in auto mode with the configuration unlocked will result in a warning message.

1. Log in as a user with the grant **Lock Safety Controller Configuration**.
2. In the **Settings** app, select the **Safety Controller**, and then **Configuration**.
3. Select the check box **Locked**.

For more information, see *Locked safety configurations on page 145*.

© Copyright 2020 ABB. All rights reserved.

6.20 Restore configuration

Restore configuration from backup

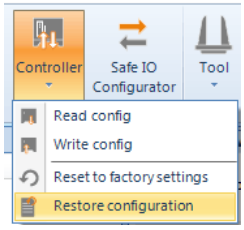
When performing a system backup, a SafeMove safety configuration file is included. This file gives the possibility to restore the SafeMove safety configuration without changing it. The advantage is that the configuration is identical, so the SafeMove safety configuration does not have to be validated and no new safety report has to be generated.

The safety settings includes the safety configuration, the EIO configuration, and the version information.

When restoring a backup, select the **Include Safety Settings** check box to also restore the safety configuration.

Restore configuration from RobotStudio

The file can also be restored separately from the system backup by using the **Restore Configuration** function in **Visual SafeMove**.

	Action	Note/illustration
1	Open **Visual SafeMove**.	
2	Click on **Controller** in the **Visual SafeMove** ribbon and select **Restore Configuration**.	 xx1500000533
3	Browse to the *BACKINFO* folder in the backup.	
4	Select the file *sc_cfg.xml* and click **Open**.	
5	The **Restart Controller** dialog is shown when the safety configuration has been downloaded. Click **Yes** to restart the controller.	

© Copyright 2020 ABB. All rights reserved.

6.21 Reset safety controller to factory settings

Introduction

The function **Reset safety controller to factory settings** clears all user settings and loads a default configuration. The firmware of the safety controller is not affected.

In rare cases the safety controller can get locked in safety state, for example when loading an incompatible or poorly configured safety configuration or replacing the robot controller. Then it is not possible to load another safety configuration without first resetting the safety controller to factory settings.

Resetting the safety controller

Use this procedure to reset the safety controller from the FlexPendant:

1. On the start screen, tap **Settings**, and then select **Backup & Recovery** from the menu.
2. On the sidebar tap **Reset user data**.
3. Select the check box **Reset safety settings**.
4. Tap **Reset**.
5. Synchronize the safety controller with the robot controller, see *Synchronization guidelines on page 155*.
6. A new safety configuration can now be loaded and validated, see *Load the configuration to the safety controller on page 130*.

Note

It is also possible to reset the safety controller to factory settings from RobotStudio, see *Configuration group on page 57*.

© Copyright 2020 ABB. All rights reserved.

6.22 Upgrading and installing new systems

Upgrading RobotWare

If an upgrade of RobotWare is done by using the Installation Manager, the safety configuration is kept as it is. A restore of the safety configuration is not needed.

Since the configuration is kept as it is, the checksum is the same and the configuration is automatically locked after upgrade.

Downgrading RobotWare

SafeMove is not forward compatible. A safety configuration from a newer system is not possible to use in an older system.

Installing a new system

After installing a new system, or after a factory reset, the safety configuration is empty and no safety functionality is active.

Use this procedure when installing a new system:

	Action
1	Backup the old system.
2	Install the new system. If the configuration was locked in the old system, there will be an error message at startup.
3	Restore the backup and include the safety settings, see *Restore configuration on page 143*.
4	Restart the system. If the configuration was locked in the old system, then it will also be locked in the new system.

Locked safety configurations

The safety configuration file in itself cannot be in locked state, it is the safety controller hardware that is locked to a specific configuration. This has the following implications on the system:

- If a locked configuration is taken from one system and loaded into another system, the configuration is not automatically locked in the new system.
- If the configuration is locked to one system which is upgraded, the configuration is automatically locked if it is loaded into the new system.
- If two systems with different safety configurations are installed on one controller, where one of the configurations is locked, the other system will not be possible to use since the hardware is locked to the first system.

For information on locking the safety configuration, see *Setting the configuration to locked on page 142*.

© Copyright 2020 ABB. All rights reserved.

This page is intentionally left blank

7 Running in production

7.1 Reaction time

Stopping response time

When a stopping supervision function is triggered, the reaction time until a stop is ordered is maximum 4 ms. This reaction time must be added to the stopping time of the manipulator, specified in the product specification for the manipulator, to get the complete stopping time for the manipulator.

Output signal response time

The time from when the robot makes a safety violation, for example entering a forbidden zone, until a safe Ethernet signal is generated shall not exceed 24 ms.

Note

The time depends on the cycle time on the communication with the PLC. 24 ms is the case when the cycle time is set to 8 ms.

Note

When a signal is set to 0 at violation, it will remain 0 for at least 250 ms even after the violation has ended.

© Copyright 2020 ABB. All rights reserved.

7.2 Restarting the controller

Restart modes

None of the restart modes **Restart**, **Reset RAPID**, or **Reset system** will affect the safety configuration.

For more information about restart procedures, see *Operating manual - Integrator's guide OmniCore*.

Removing the current system

If the current system is deleted, all safety configurations for that system will also be deactivated and deleted. After installing a new system, the safety configuration must be downloaded to the safety controller again by an authorized user, and the configuration must be validated.

If a backup is available, then the SafeMove safety configuration can be restored without the need of a validation, see *Restore configuration on page 143*.

DANGER

Installing a new system, without downloading or restoring the safety configuration to the safety controller, leaves the robot system without any of SafeMove's safety functions. It can easily be perceived as if the robot system still has SafeMove active, which causes a dangerous situation.

Tip

Set up the User Authorization System so that only the safety user is allowed to administrate installed systems.

Backup and restore

Performing a backup and restore of the system does not affect the SafeMove safety configuration.

The SafeMove safety configuration file is included in the backup. For information on how to restore the SafeMove safety configuration, see *Restore configuration on page 143*.

© Copyright 2020 ABB. All rights reserved.

7.3 Recovery after safety violation

Recovery after a supervision function has triggered in Automatic mode

When a supervision function triggers and the robot stops, perform the following to be able to move the robot again.

Speed and stand still violations

Press the motors on button on the robot controller to confirm the violation.

Position and orientation violations

	Action
1	Switch to Manual mode on the robot controller.
2	Jog the robot back to a position that does not trigger any supervision function.

Recovery after a supervision function has triggered in Manual mode

When a supervision function triggers and the robot stops, perform the following to be able to move the robot again.

Speed and stand still violations

Release and activate the three-position enabling device on the FlexPendant.

Position and orientation violations

	Action
1	Release and activate the three-position enabling device on the FlexPendant.
2	Jog the robot back to a position that does not trigger any supervision function.

Recovery from unsynchronized state

	Action	Note
1	Press the motors on button on the robot controller.	This allows the robot to be moved at reduced speed.
2	Perform a synchronization.	

© Copyright 2020 ABB. All rights reserved.

7.4 Changes to robot or robot cell

Always update safety configuration

If the following is done the safety configuration must be updated and validated again:

- A new version of the safety controller software is installed.

Evaluate if the safety configuration needs to be updated

If any of the following is done, the safety responsible person must evaluate if the safety configuration needs to be updated and validated again:

- The tool is replaced.
- Any robot part is replaced.
- The robot cell is rebuilt in any way.
- The relation between the world coordinate system and the robot base coordinate system is changed.
- The tool coordinate system is changed.
- Changes to system parameters.

Perform synchronization

If any of the following is done, a new synchronization is required:

- Revolution counter update
- Fine calibration
- Axis calibration

© Copyright 2020 ABB. All rights reserved.

8 Maintenance

8.1 Required maintenance activities

Test the motor contactors for a protective stop

Verify that a protective stop opens the motor contactors.

	Action	Note
1	In automatic mode, make SafeMove trigger a protective stop. For example, a stop issued from a safe fieldbus or a zone violation.	
2	Verify that the event log *90523 Safety Controller Protective Stop triggered* is shown on the FlexPendant.	
3	Verify that it is not possible to close the motor contactors, that is, go to state motors on.	

Internal functions are self tested

All internal functionality in the SafeMove safety controller is subject to self tests and requires no maintenance activities.

© Copyright 2020 ABB. All rights reserved.

This page is intentionally left blank

9 RAPID components

About the RAPID components

This is an overview of all instructions, functions, and data types for functional safety and SafeMove.

For more information, see *Technical reference manual - RAPID Instructions, Functions and Data types*.

Instructions

Instructions	Description
SafetyControllerSyncRequest	`SafetyControllerSyncRequest` is used to initiate the hardware synchronization procedure.

Functions

Functions	Description
SafetyControllerGetChecksum	`SafetyControllerGetChecksum` is used to get the safety controller checksum for the user configuration file.
SafetyControllerGetSWVersion	`SafetyControllerGetSWVersion` is used to get the safety controller firmware version.
SafetyControllerGetUserChecksum	`SafetyControllerGetUserChecksum` is used to get the safety controller checksum for the area with protected parameters in the user configuration file.

Data types

There are no RAPID data types for functional safety and SafeMove.

© Copyright 2020 ABB. All rights reserved.

This page is intentionally left blank

10 Reference information

10.1 Synchronization guidelines

10.1.1 Software synchronization guidelines

Uniquely defined position

The robot position for the synchronization must be chosen so that the position of the robot axes are unambiguously defined. One way to make sure the synchronization position is well-defined for all axes is to use the instruction `MoveAbsJ` to move to the sync position. See *Technical reference manual - RAPID Instructions, Functions and Data types*.

Note that the sync position should be allowed by all active functions. For example, the robot must be inside the allowed zones for all active Tool Position Supervision functions.

Note

On YuMi robots with SafeMove, the synchronization position must be defined because default synchronization position cannot be used.

Note

Software synchronization is always available even if hardware synchronization is configured.

Use easily verified sync position

Select a sync position where it is easy to verify the position of the robot axes. It is helpful to use a position where the TCP touches a spike or something where it is easy to see if the robot is in the correct position or not.

Performing a synchronization

WARNING

If the robot position is not visually verified, to make sure all robot axes are in correct position, the synchronization can jeopardize the safety.

Note

Synchronization can only be performed by a user with the grant **Software synchronization**. See *Set up safety user grants on page 97*.

	Action	
1	Move the robot to its sync position (for example with `MoveAbsJ`).	

Continues on next page

© Copyright 2020 ABB. All rights reserved.

10.1.1 Software synchronization guidelines
Continued

	Action	
2	Visually verify that the robot is in its sync position (all axes must be in correct position).	If an axis is in wrong position, the revolution counters are most likely incorrect.
3	Go to the **Safety Controller** view.	
4	On the tab **Synchronization**, tap **Synchronize**.	

© Copyright 2020 ABB. All rights reserved.

10.1.2 Hardware synchronization guidelines

Uniquely defined position

The robot position for the synchronization must be chosen so that the position of the robot axes are unambiguously defined. One way to make sure the synchronization position is well-defined for all axes is to use the instruction `MoveAbsJ` to move to the sync position. See *Technical reference manual - RAPID Instructions, Functions and Data types*.

Note that the sync position should be allowed by all active functions. For example, the robot must be inside the allowed zones for all active Tool Position Supervision functions.

Note

On YuMi robots with SafeMove, the synchronization position must be defined because default synchronization position cannot be used.

Note

Software synchronization is always available even if hardware synchronization is configured.

Small sync switch surface

For hardware synchronization, the sync switch surface that the robot touches must be small. The surface of the tool touching the sync switch must also be small. If any robot axis moves one motor revolution, the robot must be out of reach for the sync switch.

Always activate sync switch in the same way

For hardware synchronization, always use the same tool. The robot should always touch the sync switch with the same point on the tool.

Performing a synchronization

	Action	Note
1	Run a RAPID program with the instruction `SafetyControllerSyncRequest`.	This will make the controller responsive to the synchronization input signal for 30 seconds.
2	Move the robot to a position close to the sync switch.	If an axis is in wrong position, the revolution counters are most likely incorrect.
3	Slowly press the sync switch from the desired direction.	If the approach is too fast, the accuracy of the robot position may be too low.

Note

If the RAPID instruction `SafetyControllerSyncRequest` is executed and no sync signal is received within 30 seconds, the robot is stopped with an error message and the safety controller becomes unsynchronized.

© Copyright 2020 ABB. All rights reserved.

10.2 Cyclic Brake Check guidelines

About Cyclic Brake Check

If using Cyclic Brake Check on a SafeMove system, then Cyclic Brake Check should be configured in the safety configuration, see *Configuring Cyclic Brake Check on page 127*. The brake check must then be performed regularly (within the configured time interval).

Prerequisites for brake test

- The robot and all additional axes must be moved to a safe and relaxed position (away from people, equipment and not too much stretched) before performing Cyclic Brake Check. Normally the robot moves only a few centimeters during the brake tests.
- Move the robot to a stop point before performing Cyclic Brake Check.
- Cyclic Brake Check can only be performed at normal execution level (not from a trap routine, error handler, event routine or store path level).
- Brakes are tested in consecutive order and each test takes 10-15 seconds.
- Do not change the speed from the FlexPendant and do not use `VelSet`, `AccSet`, `SpeedRefresh`, or any other instruction that affects motion performance in `TRAPS` or event routines while `CyclicBrakeCheck` is active.

Note

The RAPID function `IsBrakeCheckActive` can be used to check if `CyclicBrakeCheck` is active.

Note

If Cyclic Brake Check is called together with the below combinations, then Cyclic Brake Check will be executed for all axes and the safety controller will not take any actions when the Cyclic Brake Check interval has passed or if the Cyclic Brake Check fails:

- SafeMove system with an empty safety configuration.
- SafeMove system with a safety configuration, but without a configured Cyclic Brake Check.

Performing Cyclic Brake Check

Start the brake check from RAPID

Call the procedure `CyclicBrakeCheck`.

Start the brake check from the FlexPendant

	Action	Note/illustration
1	Move the robot to a safe brake check position.	
2	On the FlexPendant, open the **Safety Controller** view.	

Continues on next page

© Copyright 2020 ABB. All rights reserved.

	Action	Note/illustration
3	On the tab **CBC**, tap **Execute**.	

Cyclic Brake Check output

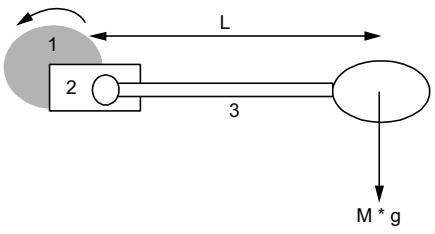
An error or warning message is logged for each axis with low brake torque. A status message is also logged for each complete brake cycle.

Cyclic Brake Check on additional axes

To be able to run CBC on additional axes, the parameter *Max Static Arm Torque* (in topic *Motion* and type *Brake*) needs to be calculated for the additional axis and entered into the configuration. CBC uses this value when testing the brake at error-level.

The parameter should be the maximum static torque that the brake needs to withstand when the additional axis is positioned in maximum gravity. The following formula should be used:

*Max Static Arm Torque = (M*L*g)/n*



xx1600001289

Length to mass (L)
Mass (M) * Gravity constant (g)
Transmission (n)
Motor (1)
Gearbox (2)
Arm (3)

To calculate the parameter for an axis that has no gravity, for example a track, the below formula may be used:

Max Static Arm Torque = Tbrake min/1.35

Tbrake min for ABB motor units can be found in the product specification for the specific motor unit, see *Product specification - Motor Units and Gear Units*.

Note

Note that the calculated value should be entered in [Nm] and calculated to the motor side.

Brake maintenance

Brake maintenance is a feature in the `CyclicBrakeCheck` functionality.

`CyclicBrakeCheck` automatically detects if maintenance of the mechanical brakes is needed and activates the *Brake maintenance* functionality during execution. *Brake maintenance* applies the brake and turns the motor shaft 1 radian five times, which gives a movement of the robot arm of less than 1 degree.

There are event logs that tell if *Brake maintenance* is needed, and if it has been run.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

For more information see parameter *Brake Maintenance*, type *General Rapid*, topic *Controller*, in *Technical reference manual - System parameters*.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

10.2.1 Cyclic Brake Check signal description

Introduction

Description of different signal states for Cyclic Brake Check (CBC).

Timing sequence for CBC signals

Description of which signals are set at different times during the Cyclic Brake Check.

Beginning of CBC

The following signals are set in the beginning of the CBC.

Signal	Set to
SC1CBCOK	0
SC1CBCACT	1
SC1CBCERR	0
SC1CBCWAR	0

End of CBC

The following signals are set in the end of the CBC.

Signal	CBC test OK Set to	CBC test WARNING Set to	CBC test ERROR Set to
SC1CBCOK	1	0	0
SC1CBCREQ	0	0	1
SC1CBCERR	0	0	1
SC1CBCWAR	0	1	0
SC1CBCACT	0	0	0
SC1CBCPREWARN	0	0	No change

Program pointer moved to Main after interrupted CBC

When the program pointer is moved to Main after an interrupted CBC the following signals are set.

Signal	Set to
SC1CBCOK	0
SC1CBCREQ	1
SC1CBCACT	0

New configuration downloaded, synchronization done, no CBC done

Signal	Signal state
SC1CBCOK	0
SC1CBCREQ	1
SC1CBCERR	0
SC1CBCWAR	0

Continues on next page

© Copyright 2020 ABB. All rights reserved.

10.2.1 Cyclic Brake Check signal description
Continued

Signal	Signal state
SC1CBCACT	0
SC1CBCPREWARN	0

Max. allowed speed: 250 mm/s

During the first CBC test

Signal	Signal state
SC1CBCOK	0
SC1CBCREQ	No change
SC1CBCERR	0
SC1CBCWAR	0
SC1CBCACT	1
SC1CBCPREWARN	No change

CBC done with the result OK

Signal	Signal state
SC1CBCOK	1
SC1CBCREQ	0
SC1CBCERR	0
SC1CBCWAR	0
SC1CBCACT	0
SC1CBCPREWARN	0

Max. allowed speed: Max speed

CBC done with the result WARNING

Signal	Signal state
SC1CBCOK	0
SC1CBCREQ	0
SC1CBCERR	0
SC1CBCWAR	1
SC1CBCACT	0
SC1CBCPREWARN	0

Max. allowed speed: Max speed

CBC done with the result ERROR

Signal	Signal state
SC1CBCOK	0
SC1CBCREQ	1
SC1CBCERR	1

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Signal	Signal state
SC1CBCWAR	0
SC1CBCACT	0
SC1CBCPREWARN	No change

Max. allowed speed: 250 mm/s

Prewarning time has expired

Signal	Signal state
SC1CBCOK	No change
SC1CBCREQ	No change
SC1CBCERR	No change
SC1CBCWAR	No change
SC1CBCACT	No change
SC1CBCPREWARN	1

Max. allowed speed: Max speed

Max CBC test interval has elapsed

Signal	Signal state
SC1CBCOK	No change
SC1CBCREQ	1
SC1CBCERR	No change
SC1CBCWAR	No change
SC1CBCACT	No change
SC1CBCPREWARN	1

Max. allowed speed: 250 mm/s

Interrupted CBC test, program pointer still in CBC routine

Signal	Signal state
SC1CBCOK	0
SC1CBCREQ	No change
SC1CBCERR	0
SC1CBCWAR	0
SC1CBCACT	1
SC1CBCPREWARN	No change

Interrupted CBC test, program pointer moved from CBC routine

Signal	Signal state
SC1CBCOK	0
SC1CBCREQ	1

Continues on next page

© Copyright 2020 ABB. All rights reserved.

10.2.1 Cyclic Brake Check signal description
Continued

Signal	Signal state
SC1CBCERR	0
SC1CBCWAR	0
SC1CBCACT	0
SC1CBCPREWARN	No change

© Copyright 2020 ABB. All rights reserved.

10.3 Servo Delay Factor and Servo Lag

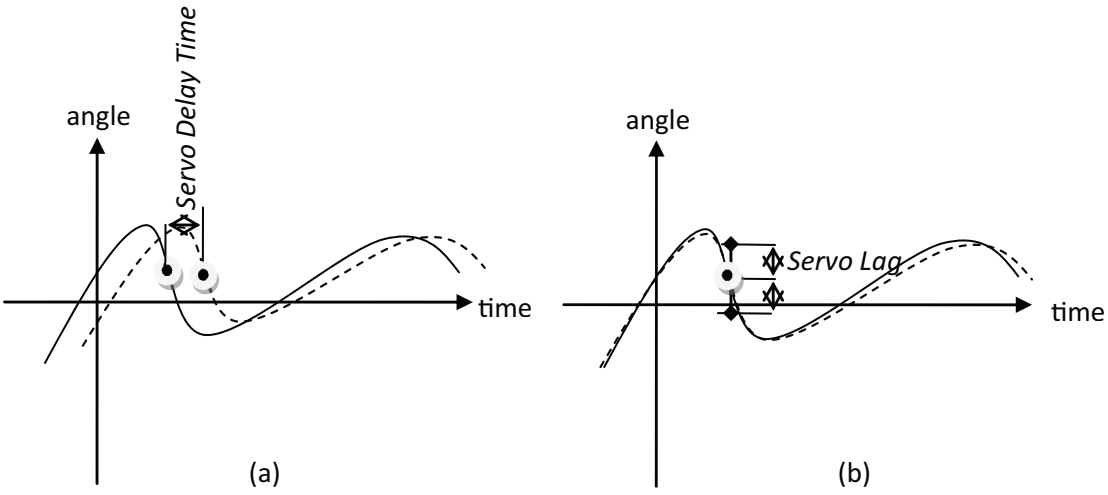
System parameters Servo Delay Factor and Servo Lag

To explain what is affected by the parameters *Servo Delay Factor* and *Servo Lag* in the SafeMove configuration (see *Configure additional axes on page 103*) it is best to consider an example.

In graph (a) in the figure below, the *Servo Delay Time* is illustrated using a constructed example where a reference and a corresponding measured angular motor position is shown. The corresponding *Servo Delay Factor* can be computed using:

Servo Delay Factor = Servo Delay Time / 4

In graph (b) in the figure below, the *Servo Lag* parameter is illustrated when the measured signal is shifted with the *Servo Delay Time*. The measured position should now be within a distance of +/- *Servo Lag* from the reference at every time instance. If the measured position is outside the specified region the SafeMove position supervision is triggered.



en1100000626

Figure 10.1: Illustration of Servo Delay Time (a) and Servo Lag (b). Solid line is motor angular position reference (test signal 17) and dashed is the corresponding measured motor angular position (test signal 18). In (b) the measured motor angular position signal is shifted Servo Delay Time in order to illustrate that it is the time shifted signal (samples) that should lie within +/- Servo Lag radians from the reference.

Note

The signals in the figure are only for illustrative purpose. The *Servo Delay Time* in a real system is small, typically in the range 12-16 ms, which means a *Servo Delay Factor* in the range 3-4. If the *Servo Delay Factor* is incorrect the *Servo Lag* has to compensate this since the difference between reference and measured motor angular position will be large when the axis accelerates, runs at high speed, or decelerates.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

Tuning the parameters

If the SafeMove position supervision is triggered for the axis, take the following actions to verify and possibly tune the parameters:

1. Check that the load of the robot and additional load on the axis is included in the configuration.
2. Make sure no mechanical problem is present, for example giving abnormal friction.
3. If possible, use TuneMaster and log the test signals 17 (motor angular position reference) and 18 (measured motor angular position) for the axis. If TuneMaster is not available, the *Servo Lag* can be increased (according to step *3b*).
   a. Move the axis and measure the *Servo Delay Time*. Compute the corresponding *Servo Delay Factor*:

      Servo Delay Factor = Servo Delay Time / 4

      where it is assumed that *Servo Delay Time* is measured in milliseconds. Use the value in the SafeMove configuration.
   b. If the supervision is still triggered, increase the *Servo Lag* until supervision is not triggered.

Tuning a non ABB additional axis

If attempting to use a non ABB additional axis, start with doing a tuning of the axis. To be able to reduce servo lag below the configured maximum allowed value during the tuning, make sure to set the parameter *FFW Mode* (feed forward mode) to *Spd* (speed) or *Trq* (torque). For more information about tuning an additional axis see .

WARNING

The system is unstable and therefore dangerous during the tuning process, since bad parameters or parameter combinations may be used! The safety procedures of the robot system must be carefully followed throughout the tuning process.

© Copyright 2020 ABB. All rights reserved.

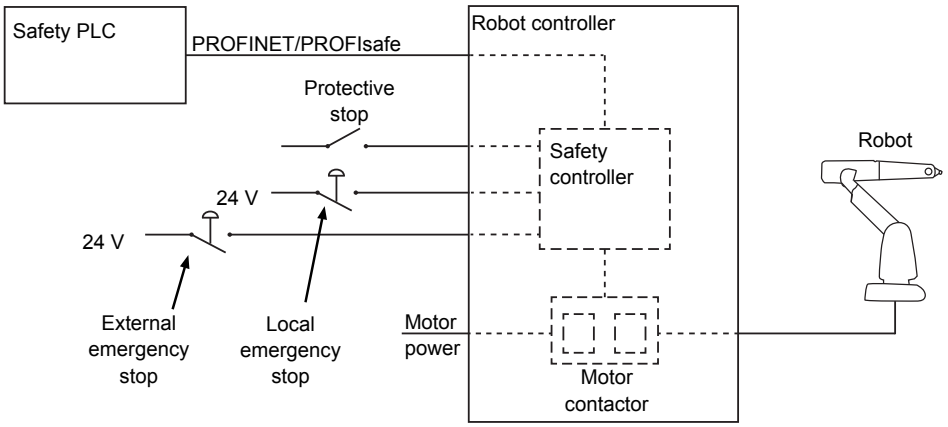
10.4 Connection of external emergency stop

About emergency stop connections

An external emergency stop can be connected either directly to the safety module of the robot controller or to the safety PLC that communicates the emergency stop to the safety controller.

Connecting emergency stop to the safety module

If the external emergency stop is connected to the safety module, it is connected in serial with the local emergency stop on the same circuit.



xx1800000411

Connecting emergency stop to a safety PLC

If the external emergency stop is connected to the safety PLC, it is connected as a safe digital input.

DANGER

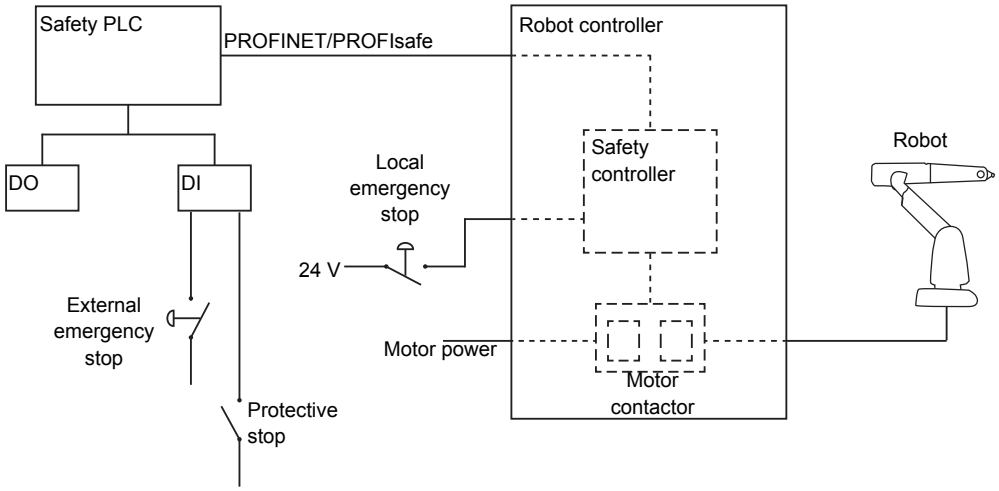
When using this solution, the communication between the safety PLC and the safety controller must be active at all times.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

10.4 Connection of external emergency stop

Continued



xx1800000412

© Copyright 2020 ABB. All rights reserved.

10.5 SafeMove geometry configuration file

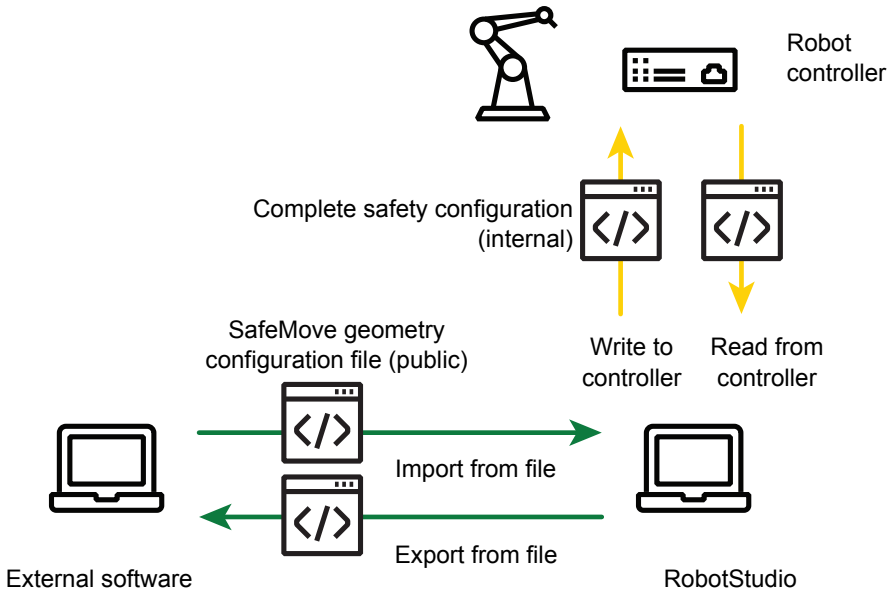
10.5.1 Introduction

Overview

This section describes the SafeMove geometry configuration file that can be used to import and export safety zones and related geometry to and from Visual SafeMove.

The intended use of the file is to provide means to define and analyze the geometry related part of a safety configuration in a tool external to RobotStudio and Visual SafeMove.

The illustration below shows how the safety related geometry information can be read from a file and written to a robot controller via Visual SafeMove. Vice versa, the safety-related geometry information can be read from the controller by Visual SafeMove and exported to a geometry configuration file.



xx1700000702

Unsupported functions

The following functions must be configured using Visual SafeMove in RobotStudio and cannot be configured through the geometry configuration file.

- Contact Application Tolerance
- Stand Still Supervision
- Axis Speed Supervision
- Axis Position Supervision
- Tool Orientation Supervision
- External Power Supply

Continues on next page

© Copyright 2020 ABB. All rights reserved.

10.5.1 Introduction
Continued

- Cyclic Brake Check
- Stop Configuration

© Copyright 2020 ABB. All rights reserved.

10.5.2 Use cases

Use case 1: Import of nominal safety related geometry information

If the planning and geometry layout is created in a tool other than RobotStudio and Visual SafeMove, the file can be used to transfer the safety-related geometry information to RobotStudio and Visual SafeMove.

The safety configuration must be completed in Visual SafeMove by adding non-geometry related safety configuration, for example brake check and activation signals, before writing it to a real controller on the shop floor.

Procedure:

1. Create the cell layout in the external planning tool.
2. Create the safety geometry in the external planning tool, for example safety zones, tool encapsulation, and speed supervision points.
3. Export the safety related geometries from the external planning tool to a SafeMove geometry configuration file.
4. Import the SafeMove geometry configuration file into Visual SafeMove in RobotStudio.
5. Complete the safety configuration by adding remaining non-geometry related safety configurations, for example brake check and activation signals.
6. Load the safety configuration to the safety controller.

Use case 2: Import of calibrated safety related geometry information

This use case is very similar to use case 1, the difference being that the safety-related geometries are adjusted with respect to measurement data of the robot cell.

Steps 3A to 3C are added to the procedure in use case 1:

A. Import the SafeMove geometry configuration file into a second tool that can adjust zones based on measurement data from the real robot cell.
B. Adjust the zones relative to the offsets of the corresponding real objects in the robot cell, for example robot base frame, fixtures, tip dressers, material supply units, etc.
C. Export the adjusted zones to a new SafeMove geometry configuration file.

Use case 3: Export of real safety related geometry information

In this use case, the actual zones and related geometry information of the real controller are exported to a SafeMove geometry configuration file using Visual SafeMove in RobotStudio. The purpose may be to update the nominal robot cell in the external planning tool.

Procedure:

1. Connect RobotStudio to the robot controller.
2. Use Visual SafeMove to read the safety configuration from the robot controller.
3. Export the configuration to a SafeMove geometry configuration file.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

4 Import the SafeMove geometry configuration file into the external planning tool so that the nominal safety zones are updated with the values from the real robot.

© Copyright 2020 ABB. All rights reserved.

10.5.3 Explanation of the configuration file

10.5.3.1 Introduction

Overview

The purpose of the SafeMove geometry configuration file is to support data related to the supervision functions:

- Tool Position Supervision
- Tool Speed Supervision
- Tool Orientation Supervision

The file can contain the tool encapsulation, speed supervision points, and the zone definitions. This data is necessary, but not sufficient, to specify the functions. To completely specify the functions then also activation signals, speed limits, and whether a zone is an inside or an outside zone must be specified and added in Visual SafeMove.

Summary

XML-tag	Description
`DriveModuleConfiguration`	1-2 per controller

`Robot`
`Baseframe`
`ElbowOffset`
`UpperArmGeometry`
	0-1 per `DriveModuleConfiguration`
	0-1 per `Robot`
	0-1 per `Robot`
	0-2 per `Robot`

`ExternalAxis` `BaseFrame`	0-3 per `DriveModuleConfiguration` 1 per `ExternalAxis`

`Tool`
`TCP`
`ToolOrientation`
`ToolGeometry`
`SpeedSupervisionPoint`
`Name`
	0-16 per `DriveModuleConfiguration`
	1 per `Tool`
	1 per `Tool`
	0-4 per `Tool`
	0-8 points per `Tool`
	1 per `Tool`

`SafeZone`
`Point`
`Name`
	0-16 per `DriveModuleConfiguration`
	4-24 per `SafeZone`
	1 per `SafeZone`

XML schema

There is an XML schema available that defines the exact syntax and content of the interface file. This is stored in an xsd-file and can be used to validate the content of the xml-file.

The template file *SimplifiedSafetyControllerSchema.xsd* can be obtained from the PC or the robot controller.

- **In the RobotWare installation folder in RobotStudio**: *...\RobotPackages\RobotWare_RPK_<version>\utility\SafeMove2*

Continues on next page

© Copyright 2020 ABB. All rights reserved.

10.5.3.1 Introduction
Continued

- **On the robot controller:**
  \products\<RobotControl_xx.xx.xxxx>\utility\SafeMove2

Note

Navigate to the RobotWare installation folder from the RobotStudio **Add-Ins** tab, by right-clicking on the installed RobotWare version in the **Add-Ins** browser and selecting **Open Package Folder**.

© Copyright 2020 ABB. All rights reserved.

10.5.3.2 Drive module configuration

Overview

The `DriveModuleConfiguration` tag contains information about the motion task.

There is a drive module configuration for each motion task of a controller system monitored by SafeMove.

ID

The `driveModuleId` tag contains the ID of the motion task.

Each motion task must have a unique ID between 1 and 4. The ID is not visible to the user.

© Copyright 2020 ABB. All rights reserved.

10.5.3.3 Robot data

Overview

The `Robot` tag contains information about the robot.

Name

The `name` tag is `ROB_1` for a controller with only one manipulator. For a MultiMove system the subsequent robot is called `ROB_2`.

Robot moved by

For track mounted robots, the `movedBy` tag specifies the name of the external axis that moves the robot.

Base frame

The `Baseframe` tag specifies the robot mounting position relative to the controller world coordinate system.

When importing a SafeMove geometry configuration file into Visual SafeMove, the base frame values contained in the file will be replaced by the base frame of the connected controller when creating the complete safety configuration. When exporting the file, the `Baseframe` tag will contain the values of the actual base frame of the connected robot.

Elbow offset

The `ElbowOffset` tag defines the point on the robot elbow that is speed monitored. It should be the top-most point of the upper arm including any cable packages or additional equipment.

Upper arm geometry

The `UpperArmGeometry` tag defines the encapsulation of the upper arm.

The upper arm may optionally be encapsulated by up to two shapes, so-called sphere-swept volumes, and used in combination with a tool encapsulation for the function Tool Position Supervision. The shapes should include the upper arm of the robot including any cable packages or additional arm-mounted equipment. The

Continues on next page

© Copyright 2020 ABB. All rights reserved.

reference point of the upper arm geometry is the same as the elbow offset of the motion configuration file of the controller, *moc.cfg*.

xx1700000703

© Copyright 2020 ABB. All rights reserved.

10.5.3.4 External axis

Overview

The `ExternalAxis` tag contains information about external axes.

Name

The `name` tag contains the name of the external axis. For example `TRACK_1`.

Axis type

The `axisType` tag must be `Track` for a track motion.

Base frame

The `Baseframe` tag specifies the position and orientation of the track motion.

© Copyright 2020 ABB. All rights reserved.

10.5.3.5 Tool

Overview

The `Tool` tag contains information about the tools.

A tool consists of a set of shapes and a set of points. The shapes are used by the function Tool Position Supervision and the points by the function Tool Speed Supervision.

Up to 16 tools can be defined.

ID

The `id` tag contains the ID of the tool.

Each tool must have a unique ID between 1 and 16. The ID is not visible to the user.

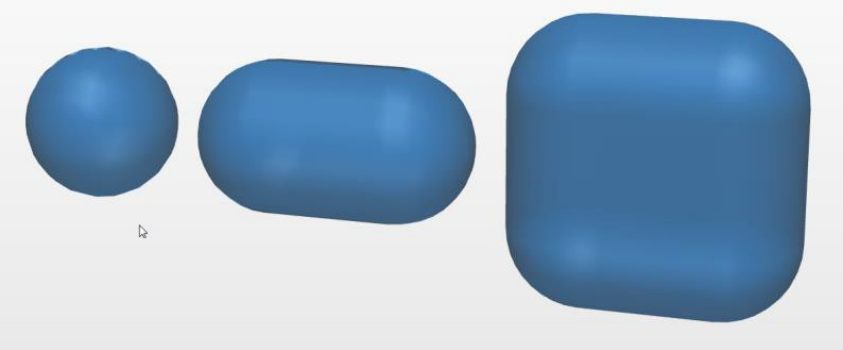
Name

The `name` tag contains the name of the tool.

Tool geometry

The `ToolGeometry` tag defines the number of shapes.

A tool consists of up to four shapes, so called sphere-swept volumes. A shape can be a sphere, a capsule, or a rounded box (lozenge).



xx1700000705

The shapes are generated by a sphere with a certain radius in combination with a point, a line, or a plane, respectively.

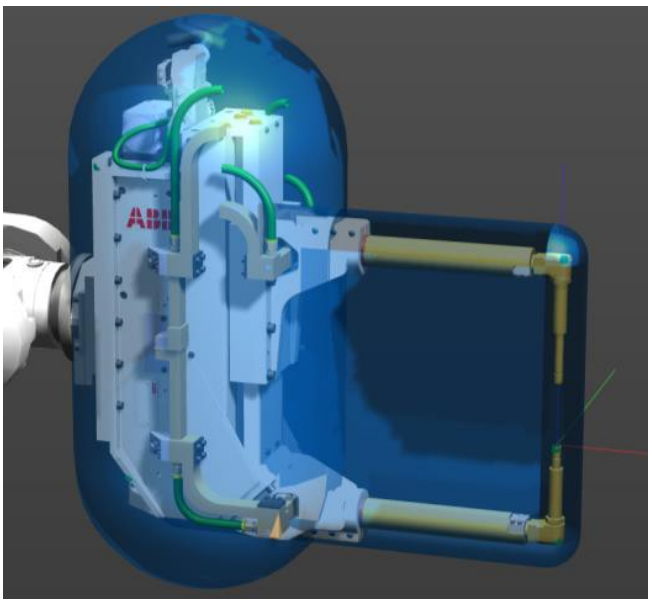
The sphere is trivially generated by placing the generating sphere on a point. A capsule is generated by letting the center of the generating sphere travel along a line. A rounded box is generated by letting the center point of the generating sphere travel along a plane.

Continues on next page

© Copyright 2020 ABB. All rights reserved.

10.5.3.5 Tool
Continued

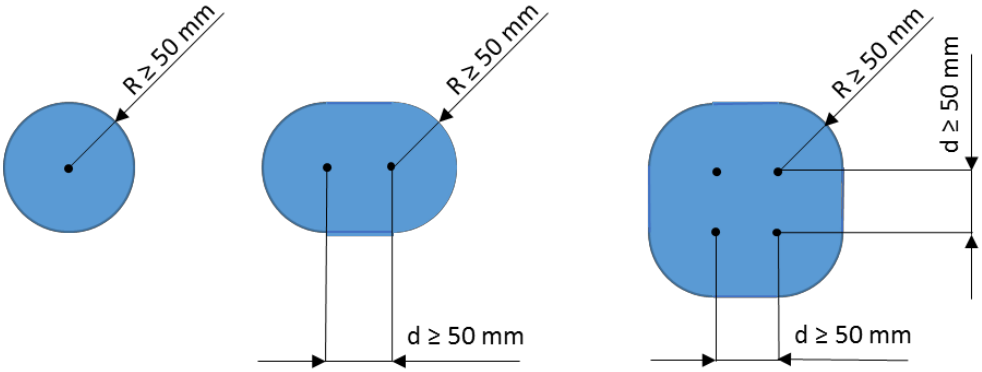
The below figure shows an example of a tool encapsulation with an ABB FlexGun covered by a capsule and a rounded box.



xx1700000706

Limitations of the tool shapes

The radius of any tool shape must be at least 50 mm. The distance between the two end points of the generating line of a capsule must be at least 50 mm. The length and width of the generating plane of a rounded box must be at least 50 mm.



xx1700000707

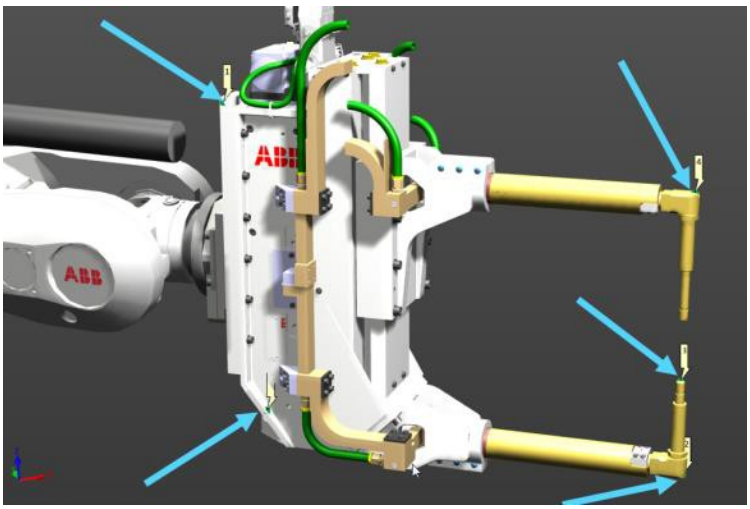
Continues on next page

© Copyright 2020 ABB. All rights reserved.

Speed supervision points

The `SpeedSupervisionPoint` tag defines the number of speed supervision points.

Up to eight points can be defined on the robot tool. The points are defined in relation to the wrist point (WCP) which is in the center point of the flange of the robot.



xx1700000708

TCP and tool orientation

The `TCP` and `ToolOrientation` tags defines the TCP and orientation of the current `tooldata` and must be specified.

© Copyright 2020 ABB. All rights reserved.

10.5.3.6 Safe zones

Overview

The `SafeZone` tag contains information about the safe zones.

The functions Tool Position Supervision, Tool Orientation Supervision, and Tool Speed Supervision are based on zones. Up to 16 safe zones can be defined.

ID

The `id` tag contains the ID of the zone.

Each zone must have a unique ID between 1 and 16. The ID is not visible to the user.

Name

The `name` tag contains the name of the zone.

The name will be visible to the user and presented in the controller event log in case of a safety violation.

Top and bottom

The `top` and `bottom` tags defines the top and bottom of the zone and must be specified.

Points

The `Point` tag defines the points of the zone.

A zone is defined by minimum 4 up to 24 points in the X-Y plane of the controller world coordinate system in addition to the bottom and top in the Z direction.

Speed limit priority

The `speedLimitPriority` tag defines the speed limit priority of the zone.

Zones can have three different speed limit priorities: `BASE`, `NORMAL`, and `OVERRIDE`.

Zones can be nested, which means that one zone can overlap or be contained in another zone. Nested zones with different speed priorities can be used to define tighter robot cells without sacrificing cycle time. In particular, the stopping distances can be kept low by defining a large `BASE` zone with low speed limits and interior zones with increasing priorities and speed limits.

The interior, high speed zones, ensure high robot productivity and low cycle times.

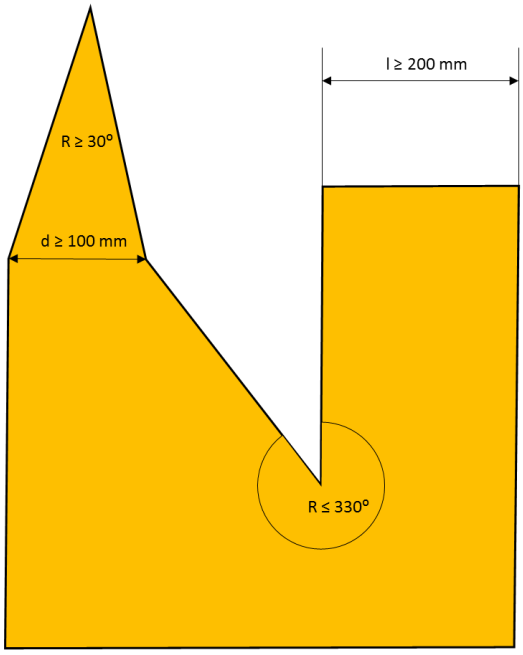
Continues on next page

© Copyright 2020 ABB. All rights reserved.

Limitations

Any of the coordinates must not exceed ± 1000 meters. The edges of a zone must be at least 0.2 m and they must not intersect. The zone must be wider than 0.1 m. A corner of a zone must have an angle between 30 and 330 degrees.

The below illustration shows the constraints that must be met by any zone definition.



xx1700000704

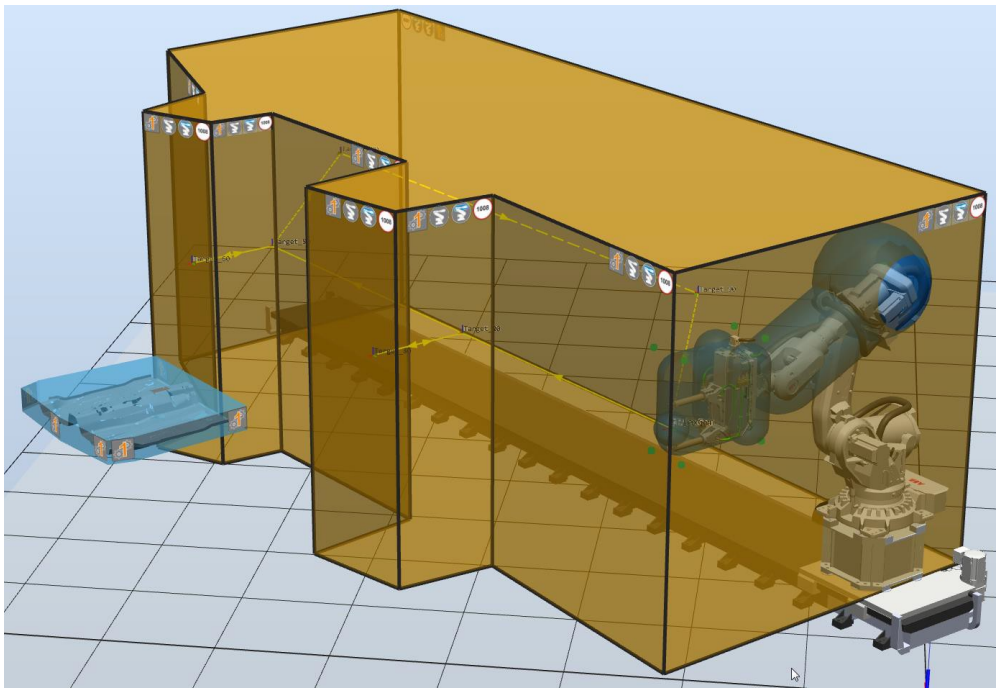
© Copyright 2020 ABB. All rights reserved.

10.5.4 Example

Overview

This section contains an example of an XML-file for a MultiMove systems with two robots. The first robot, ROB_1, is mounted on a track motion, TRACK_1, whereas the second, ROB_2 is mounted on the floor.

The example file defines the zones in the below illustration.



xx1700000709

Example file

```
<SimplifiedSafetyConfiguration
     xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
     xmlns="urn:abb-robotics-simplified-safety-controller-configuration">
<DriveModuleConfiguration DriveModuleId="1"
     maxSpeedManualMode="0.25">
  <Robot name="ROB_1" startSpeedOffset="0.1" movedBy="TRACK_1">
    <Baseframe>
      <Translation x="0" y="0" z="0" />
      <Quaternion q1="0.707107" q2="0" q3="0" q4="0.707107" />
    </Baseframe>
    <ElbowOffset x="-0.252" y="0" z="0.342" />
    <UpperArmGeometry xs:type="Capsule" name="UpperArm"
         radius="0.357727">
      <Start x="-0.268245" y="-0.017373" z="0.190867" />
      <End x="1.03026" y="-0.096637" z="0.167713" />
    </UpperArmGeometry>
    <UpperArmGeometry xs:type="Sphere" name="HosePackage"
         radius="0.5">
      <Center x="-0.084216" y="-0" z="0.305324" />
```

Continues on next page

© Copyright 2020 ABB. All rights reserved.

```
  </UpperArmGeometry>
</Robot>
<ExternalAxis name="TRACK_1" axisType="Track">
  <Baseframe>
    <Translation x="0" y="0" z="0" />
    <Quaternion q1="1" q2="0" q3="0" q4="0" />
  </Baseframe>
</ExternalAxis>
<Tool id="1" name="tFlexGun">
  <TCP x="0" y="0.1485" z="0.8015" />
  <ToolOrientation q1="0.5" q2="0.5" q3="-0.5" q4="0.5" />
  <ToolGeometry xs:type="Lozenge" name="GunBody" radius="0.183332"
       width="0.542759" height="0.5">
    <Pose>
      <Translation x="0.063143" y="0" z="0.377878" />
      <Quaternion q1="0.707107" q2="0" q3="-0.707107" q4="0" />
    </Pose>
  </ToolGeometry>
  <ToolGeometry xs:type="Capsule" name="BodyRight"
       radius="0.181797">
    <Start x="0.2" y="0.2" z="0.15" />
    <End x="0.2" y="-0.3" z="0.15" />
  </ToolGeometry>
  <ToolGeometry xs:type="Sphere" name="Electrode"
       radius="0.144212">
    <Center x="0" y="0.162418" z="0.78145" />
  </ToolGeometry>
  <ToolGeometry xs:type="Capsule" name="BodyLeft"
       radius="0.181797">
    <Start x="-0.2" y="0.2" z="0.15" />
    <End x="-0.2" y="-0.3" z="0.15" />
  </ToolGeometry>
  <SpeedSupervisionPoint x="-0.212" y="-0.499717" z="-0.005" />
  <SpeedSupervisionPoint x="-0.212" y="0.3855" z="-0.005" />
  <SpeedSupervisionPoint x="-0.212" y="0.3855" z="0.8515" />
  <SpeedSupervisionPoint x="-0.212" y="-0.499717" z="0.8515" />
  <SpeedSupervisionPoint x="0.15799" y="-0.499717" z="-0.005" />
  <SpeedSupervisionPoint x="0.15799" y="0.3855" z="-0.005" />
  <SpeedSupervisionPoint x="0.15799" y="0.3855" z="0.8515" />
  <SpeedSupervisionPoint x="0.15799" y="-0.499717" z="0.8515" />
</Tool>
<SafeZone id="1" name="RobotZone1" top="2.889806" bottom="-0.5"
     speedLimitPriority="NORMAL">
  <Point x="3.181233" y="3.743311" />
  <Point x="2.13778" y="3.552878" />
  <Point x="2.042189" y="2.624328" />
  <Point x="-0.430474" y="2.750736" />
  <Point x="-0.359415" y="-0.735027" />
  <Point x="8.319961" y="-0.790136" />
  <Point x="8.181677" y="2.193857" />
  <Point x="6.422987" y="2.805629" />
```

Continues on next page

© Copyright 2020 ABB. All rights reserved.

10.5.4 Example
Continued

```
      <Point x="6.193726" y="3.606499" />
      <Point x="5.458064" y="3.311171" />
      <Point x="5.399194" y="2.682248" />
      <Point x="3.298018" y="2.624001" />
    </SafeZone>
    <SafeZone id="2" name="Safe_Zone" top="0.931383" bottom="0.710112"
         speedLimitPriority="NORMAL">
      <Point x="3.41184" y="5.366856" />
      <Point x="3.462365" y="4.22674" />
      <Point x="5.035686" y="4.224637" />
      <Point x="5.101747" y="5.381664" />
      <Point x="5.023219" y="5.600441" />
      <Point x="4.195253" y="5.658945" />
      <Point x="3.479269" y="5.592942" />
    </SafeZone>
  </DriveModuleConfiguration>
  </SimplifiedSafetyConfiguration>
```


© Copyright 2020 ABB. All rights reserved.

Index

A
additional axes, 25–26
  work area limitations, 26
APO, 29, 45
ASP, 29, 42
axis calibration, 150
Axis Position Supervision, 29, 45
  description, 45
axis range, 46
Axis Speed Supervision, 29, 42
  description, 42

B
base coordinate system changed, 150
brake check
  guidelines, 158

C
CAP, 29, 34
category 0 stop, 28
category 1 stop, 28
Certifications, 23
compatibility, 25
Configuring SafeMove, 93
Contact Application Tolerance, 29, 34
  description, 34
Control Error Supervision
  description, 49
Cyclic Brake Check
  guidelines, 158
  signal description, 161

D
deactivation, 26
download configuration, 130

E
electronically linked motors, 27
emergency stop, 167
exclude from configuration, 32, 69
external axes, combined, 26
external emergency stop, 167

F
fine calibration, 150
floor plan, 132

G
gantry, 26
grants, 97

H
hardware synchronization
  description, 37

I
include safety settings, 143
independent joint, 27
installing
  RobotStudio, 52
  RobotWare, 52
  software, 52
integrator responsibility, 13
IsBrakeCheckActive, 158
ISO 10218, 23
ISO 13849-1, 23

L
limitations, 25
load configuration, 129
locked configuration file, 142, 145

M
maintenance, 151
Manual Operation
  using, 149
Manual Operation Supervision
  description, 33
MoveAbsJ, 155, 157

N
non ABB additional axes, 26

O
occupationally safe, 28
open configuration, 129
operationally safe, 28

R
range, 46
RAPID components, 153
RAPID non motion execution, 27
reaction time, 147
rebuilt robot cell, 150
recovery, 149
replaced robot part, 150
replaced tool, 150
response time, 147
restarting controller, 148
revolution counter updated, 150
ribbon, 55, 57, 65
robots supported by SafeMove, 25

S
Safe Brake Ramp, 71
  description, 39
safe input, 28
safe output, 28
safety, 13
safety configuration report, 132
safety controller, 15, 28
SafetyControllerGetChecksum, 153
SafetyControllerGetSWVersion, 153
SafetyControllerGetUserChecksum, 153
SafetyControllerSyncRequest, 153
safety controller version, 150
safety users, 97
save configuration, 129
Servo Delay Factor, 165
servo lag, 34, 49
Servo Lag, 165
servo tool changer, 26
servo welding gun, 25
shared drive modules, 27
signals for Cyclic Brake Check, 161
soft servo, 34
software synchronization
  description, 36
SST, 29, 40
stand alone controller, 26
Stand Still Supervision, 29, 40
  description, 40

© Copyright 2020 ABB. All rights reserved.

stopping time, 147
supervision functions, 31, 40
supported positioners, 25
supported robots, 25
supported tracks, 25
synchronization
  description, 36
  guidelines, 155
sync position, 37, 155, 157
system integrator requirements, 13
system parameters changed, 150

T
tool, 157
tool changer, 26
tool coordinate system changed, 150
Tool Orientation Supervision, 29, 48
  description, 48
Tool Position Supervision, 29, 47
  description, 47
Tool Speed Supervision, 29, 43
  description, 43
TOR, 29, 48
TPO, 29, 47
TSP, 29, 43

U
UAS grants, 97
unsynchronized state, 149
user interface
  FlexPendant, 89

V
validated configuration file, 142
violation signal, 31, 67, 147
virtual signals for brake check, 161
Visual SafeMove, 53

W
work area for additional axes, 26
world coordinate system changed, 150
write configuration, 130

© Copyright 2020 ABB. All rights reserved.

ABB AB, Robotics
Robotics and Motion
S-721 68 VÄSTERÅS, Sweden
Telephone +46 (0) 21 344 400

ABB AS, Robotics
Robotics and Motion
Nordlysvegen 7, N-4340 BRYNE, Norway
Box 265, N-4349 BRYNE, Norway
Telephone: +47 22 87 2000

ABB Engineering (Shanghai) Ltd.
Robotics and Motion
No. 4528 Kangxin Highway
PuDong District
SHANGHAI 201319, China
Telephone: +86 21 6105 6666

ABB Inc.
Robotics and Motion
1250 Brown Road
Auburn Hills, MI 48326
USA
Telephone: +1 248 391 9000

abb.com/robotics

3HAC066559-001, Rev A, en

© Copyright 2020 ABB. All rights reserved.
Specifications subject to change without notice.