ROBOTWARE ADD-IN

User Manual
StateMachine Add-In 2.0.0

Contents

Requirements

- One or more ABB robots.
- RobotWare 7.x.x based system.

Overview

This RobotWare Add-In provides a RAPID state machine implementation, which listens for changes in certain IO-signals. Most of the IO-signals are coupled with requesting transitions into different states, which are only made if allowed (depends on the currently active state).

If the Add-In has been selected during system installation, then several RAPID modules and system configurations are loaded into the system. Each motion task in the system receives its own instances of the modules (that are loaded for such tasks). An optional RAPID watchdog implementation can also be selected during system installation. The watchdog is loaded into a separate task for multitask systems, and into the main task for non-multitask systems.

Several core modules and configurations are always loaded, while a set of additional modules and configurations might also be loaded depending on the system specifications. For example:

Robot Types:
- [**General case**]: 1 regular robot.
- [**Special case**]: 1 IRB14050 robot (a.k.a. Single-Arm YuMi (SAY) robot).

RobotWare System Options:
- Multitasking (3114-1).
- Externally Guided Motion (3124-1).
- SmartGripper for IRB14050.

RobotWare Add-In Options:
- StateMachine Add-In Watchdog.

The combination of the modules and configurations compose several services, which are intended to provide ease-of-use of a subset of the robot controller's functionalities.

The state machine implementation is intended to be used in combination with **external component(s)**, which are using Robot Web Services 2.0 (RWS2) and (optionally) Externally Guided Motion (EGM) for communication with the robot controller. An **external component** can for example be a C++ RWS client program, or a C++ EGM server program, executing on a user's own external computer.

In short: **RWS2 is an interface for general communication** (for example, reading/writing IO-signals and RAPID program data, starting/stopping the RAPID program, reading robot controller status etc.). On the other hand, **EGM is an interface used for specialized communication** (for example, streaming motion references for direct motion control).

There are several other options for communication between external components and ABB robot controllers, but as mentioned earlier, this Add-In is intended to be used with RWS2 and (optionally) EGM.

See the **Additional Material** section for some information concerning the usage of RWS2 and EGM interfaces.

RAPID Modules and System Configurations

The following tables specify the RAPID modules and system configurations loaded by the Add-In. The tables also indicate the dependencies for each item. Noteworthy information is **highlighted in red**. The "X" and "_X" suffixes depend on either the number of robots in the system, or the selected options.

Table 1: RAPID modules.

Dependency	Module	Comment
Core	TRobMain.mod	The StateMachine's main RAPID routine.
Core	TRobUtility.mod	Provides RAPID utility routines and functions.
Core	TRobRAPID.mod	Facilitates execution of predefined (**general**) RAPID routines.
Core	TRobSystem.sys	Facilitates execution of predefined (**system-specific**) RAPID routines.
Core	TRobSystemExample.sys	Examples of custom (**system-specific**) RAPID routines and data.
Watchdog	TRobWatchdog.mod	Watchdog timer, supervising external status signals (**used in non-multitask systems**).
Watchdog	TWatchdogMain.mod	The watchdog's main RAPID routine, supervising external status signals (**used in multitask systems**).
EGM	TRobEGM.mod	Facilitates execution of EGM-specific RAPID routines.
SmartGripper	TRobSG.mod	Facilitates execution of SmartGripper-specific RAPID routines.

Table 2: Configurations (**Communication domain**).

Dependency	Type	Instance(s)	Comment
EGM	UDP Unicast Device	ROB_X	EGM communication settings (**important to update with correct remote IP-address and port number after system installation**).

Table 3: Configurations (**Controller domain**).

Dependency	Type	Instance(s)	Comment
Core	Automatic Loading of Modules	TRobX.mod TRobX.sys	Indicates the RAPID modules that are loaded into each motion task.
Watchdog	Event Routine	Quick Stop	RAPID routine for resetting the watchdog's status, executed when a quick stop event occurs.
Watchdog	Task	T_WATCHDOG	Specifies a separate watchdog RAPID task (**used in multitask systems**).
Watchdog	Automatic Loading of Modules	TWatchdogMain.mod	Loads the watchdog task's main RAPID module (**used in multitask systems**).
EGM	Automatic Loading of Modules	TRobX.mod TRobX.sys	Indicates the RAPID modules that are loaded into each motion task.
SmartGripper	"	"	"

Table 4: Configurations (**Motion domain**).

Dependency	Type	Instance(s)	Comment
EGM	External Motion Interface Data	ROB_X	EGM settings for applying filter to motion references.
EGM	External Motion Interface Data	ROB_X_RAW	EGM settings for using raw (**non-filtered**) motion references.

Table 5: Configurations (I/O System domain).

Dependency	Type	Instance(s)	Comment
Core	Signal	RUN_RAPID_ROUTINE	Intended for external RWS-based component(s) to request run of (predefined) RAPID routines.
Core	Signal	OUTPUT_STATIONARY_ROB_X	Indicates if a mechanical unit is moving or not.
Core	Signal Output	OUTPUT_STATIONARY_ROB_X	Internal signal output connected to the stationary indicator signal.
Watchdog	Signal	WD_EXTERNAL_STATUS	Intended for external RWS-based component(s) to periodically set. Supervised by the watchdog (if activated and set to do so).
Watchdog	Signal	WD_STOP_REQUEST	Intended for external RWS-based component(s) to request stop of RAPID execution via the watchdog (if activated).
Watchdog	Signal	WD_STOP_TRIGGER	Internal signal to indicate stop of RAPID execution.
Watchdog	Signal	WD_STOP	Internal signal to stop RAPID execution.
Watchdog	Signal Input	WD_STOP	Internal signal input connected to the stop RAPID execution signal.
Watchdog	Cross Connection	WD_STOP_CC	Internal cross connection between stop indicator and stop signals for stopping RAPID execution.
EGM	Signal	EGM_START_JOINT EGM_START_POSE EGM_START_STREAM	Intended for external RWS-based component(s) to request start of EGM joint/pose motions or position streaming.
EGM	Signal	EGM_STOP EGM_STOP_STREAM	Intended for external RWS-based component(s) to request stop of EGM motions or position streaming
SmartGripper	Signal	RUN_SG_ROUTINE	Intended for external RWS-based component(s) to request run of (predefined) SmartGripper RAPID routines.

Table 6: Configurations (**Man-Machine Communication domain**).

Dependency	Type	Instance(s)	Comment
Core	Most Common I/O Signal	RUN_RAPID_ROUTINE OUTPUT_STATIONARY_ROB_X	For ease-of-access, for example via a FlexPendant.
Watchdog	"	WD_EXTERNAL_STATUS WD_STOP_REQUEST	"
EGM	"	EGM_START_JOINT EGM_START_POSE EGM_START_STREAM EGM_STOP EGM_STOP_STREAM	"
SmartGripper	"	RUN_SG_ROUTINE	"

State Machine Runtime Illustration

The following is an illustration of different states, and how the state transitions are made during runtime.

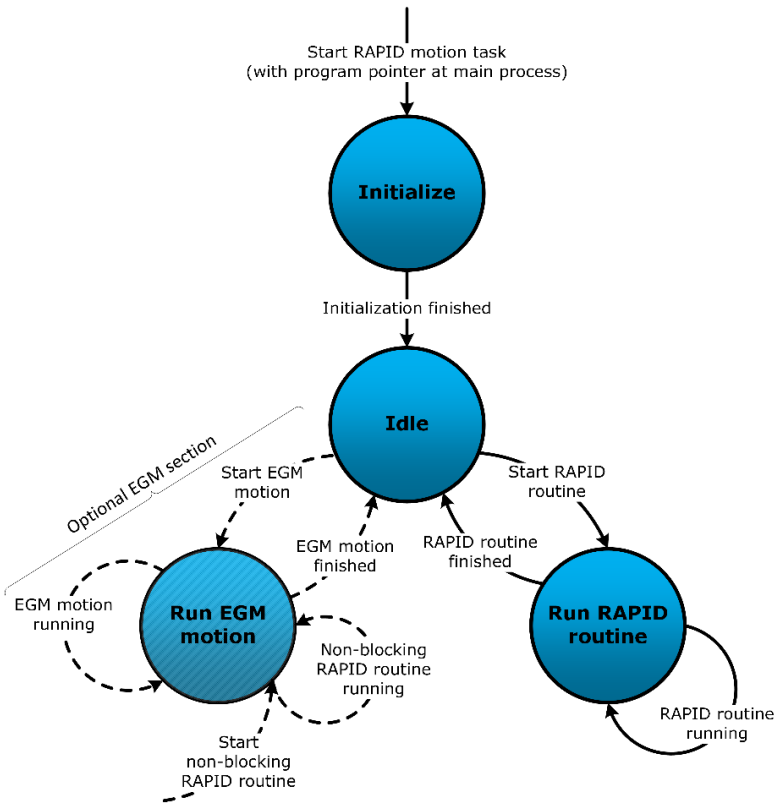

Figure 1: Sketch of the state machine and state transitions. The EGM section is only included if the EGM option exists in the system.

Example 1:

When the IO-signal **RUN_RAPID_ROUTINE** goes from 0 to 1, and the state machine is in the idle state, then the program will attempt to execute the desired RAPID routine (**specified by setting RAPID variable(s) beforehand**). If the RAPID routine has been added to the list of known routines, then the state machine enters the corresponding state and executes the routine. If the routine is non-blocking, then the state machine doesn't change state.

[**Note**]: There are a few predefined RAPID routines in the **TRobRAPID.mod** module, and additional custom user-specific routines are recommended to be added/registered in the **TRobSystem.sys** module.

Example 2:

If using EGM, then when either of the IO-signals **EGM_START_JOINT**, and **EGM_START_POSE** goes from 0 to 1, and the state machine is in the idle state, then the program will start the desired EGM motion session. The robot controller starts by sending out a feedback message to the external EGM server.

If no reply is received from the EGM server, then a timeout occurs, and the robot controller ends the EGM motion session, and the state machine goes back to the idle state.

If the EGM server replies with motion references, then the robot till attempt to follow the references. The EGM motion session continues until either the predefined EGM conditions are met, a timeout occurs or the IO-signal **EGM_STOP** goes from 0 to 1. When the EGM motion session ends, then the state machine goes back the to idle state.

[**Note**]: The **TRobEGM.mod** module contains several parameters that should be updated to meet the needs of the user.

System Installation

Installation of a virtual controller via RobotStudio is used as an example. The installation process for a real robot controller is practically identical, with a few differences that are **highlighted in red**.

The StateMachine Add-In is assumed to have been added to RobotStudio's pool of installed packages, so that it is available during system installation. This can be verified by inspecting: **RobotStudio → Add-Ins tab → Installed Packages list → StateMachine Add-In <version> exists**.

Installation Manager 7 is used to install the system: **RobotStudio → Controller tab → Installation Manager → Installation Manager 7**.

Step 1: Controllers

Under the Virtual controllers tab, press New, give the system a reasonable name, and press Next.

For a real robot controller, use the Real controllers tab to find your controller (if it doesn't show up, make sure that there is a network connection to the physical controller). Press New, give the system a reasonable name, and press Next (remove an old system if required to free up memory).

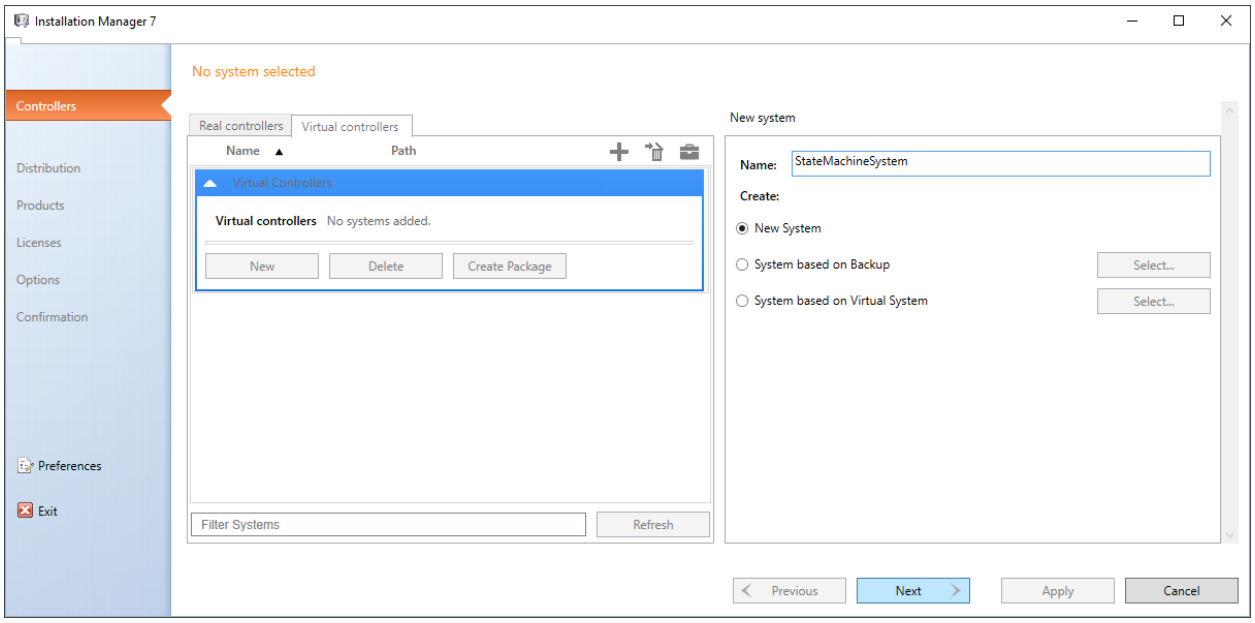

Figure 2: Create a new system.

Step 2: Distribution

Add a RobotWare distribution by pressing Add… and selecting the desired distribution. If the desired distribtion doesn’t show up in the list, abort the installation and add it to RobotStudio’s installed packages

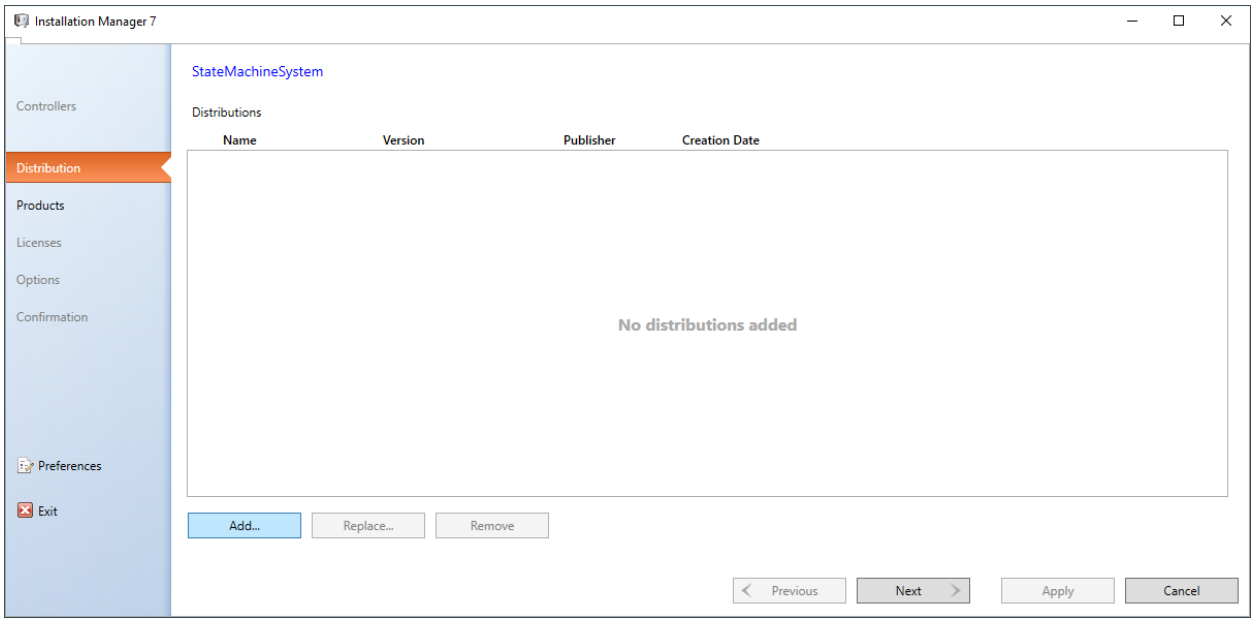

Figure 3: Add a RobotWare distribution.

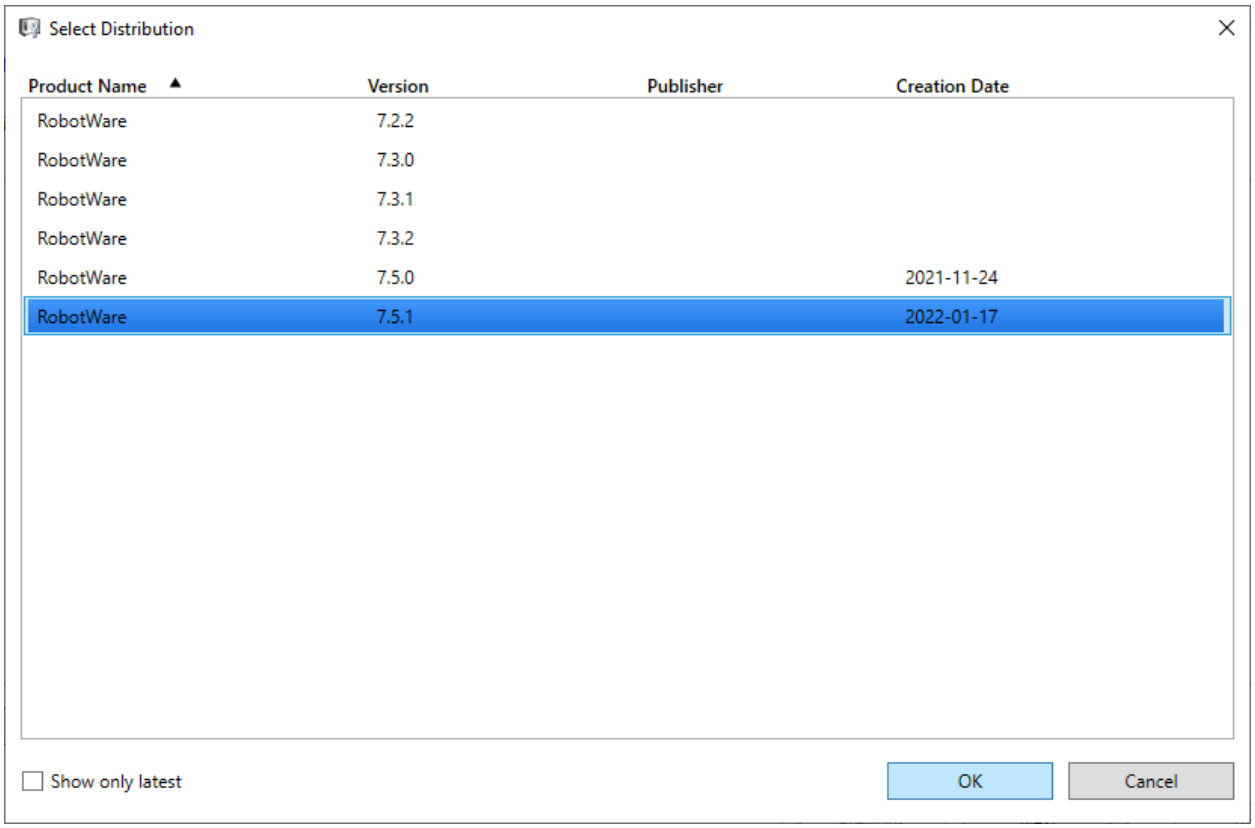

Figure 4: Select the desired RobotWare distribution in the list.

Step 3: Products

A set of core products should automatically have been added based on the selected RobotWare distribution. Add the StateMachine Add-In product, as well as any additional products, by pressing Add… and selecting the desired product(s). If the StateMachine Add-In product doesn't show up in the list, abort the installation and add it to RobotStudio's installed packages.

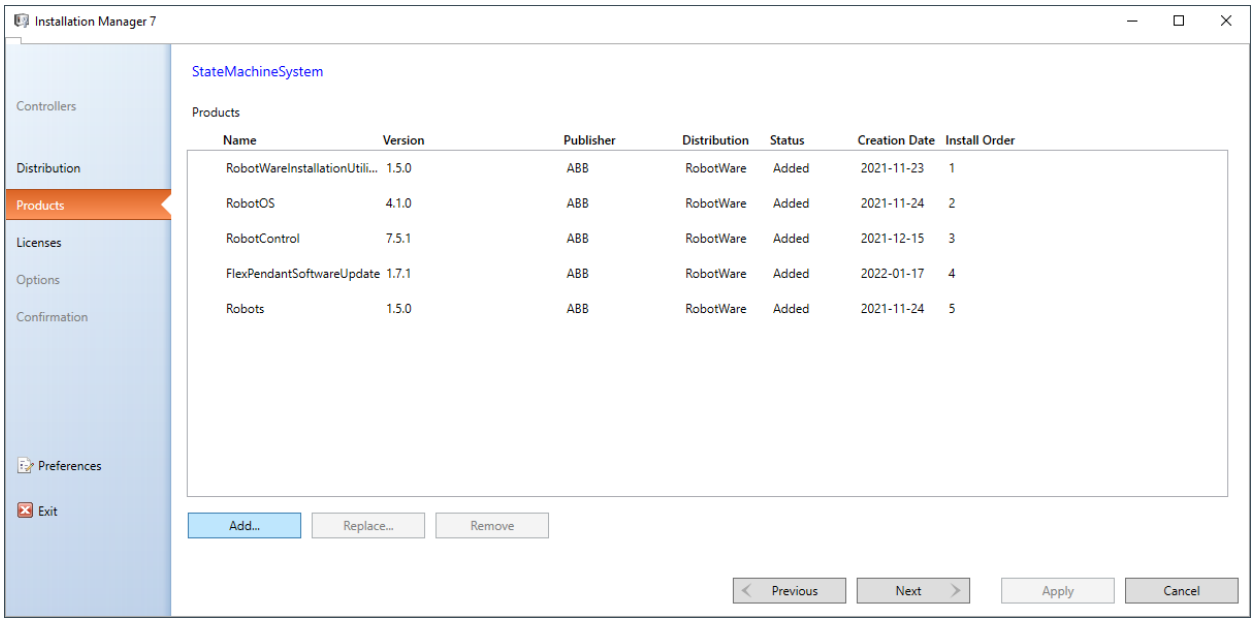

Figure 5: Add additional products.

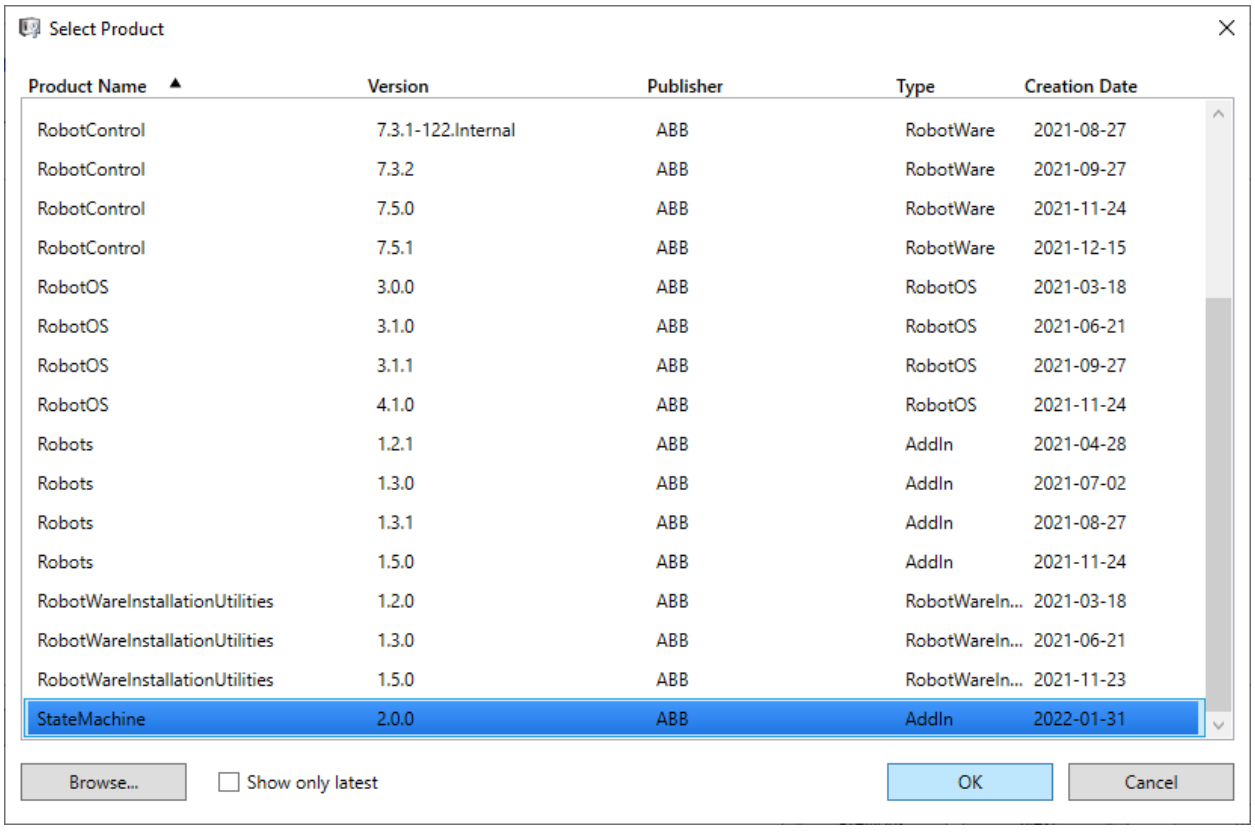

Figure 6: Select the desired product in the list.

Step 4: Licenses

For a virtual controller, virtual licenses should have been added by default. Press Next to continue.

For a real robot controller, the license list will be empty. Press Add… to add the license(s) for your specific robot controller. Press Next to continue.

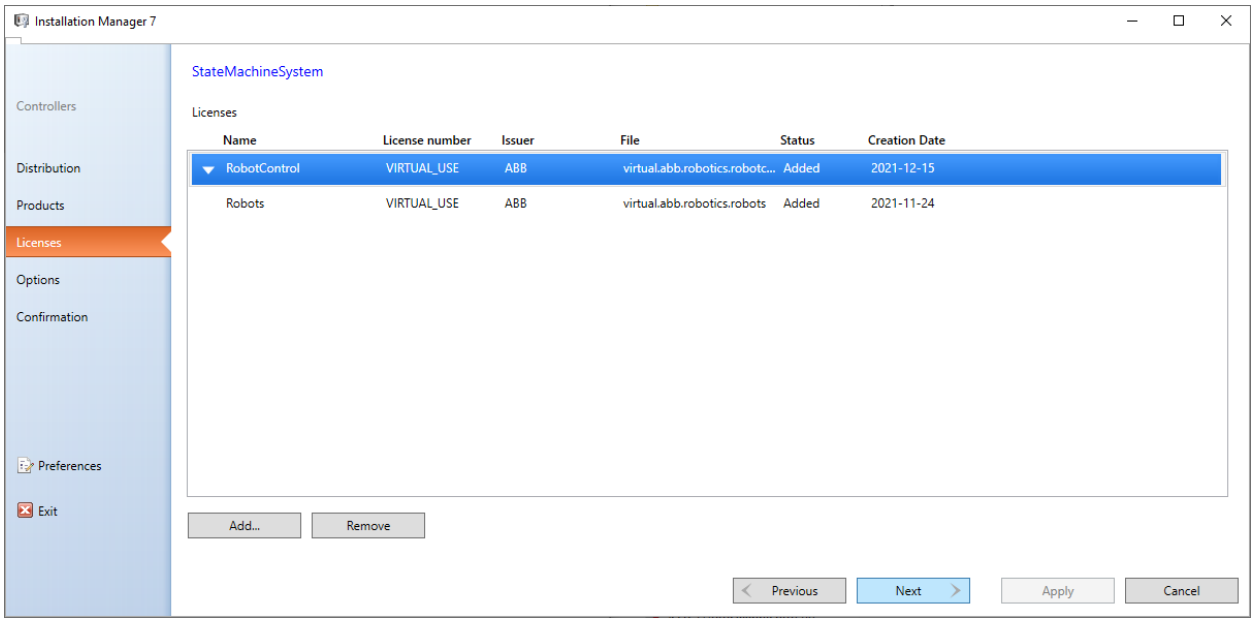

Figure 7: Add licenses.

Step 5: Options

Add the desired system options (for example, 3124-1 Externally Guided Motion (EGM)), robot type, and application options (for example, the StateMachine Watchdog).

For a real robot controller, then the system options and robot type included in the license(s) should have automatically been selected. Add any additional desired application options.

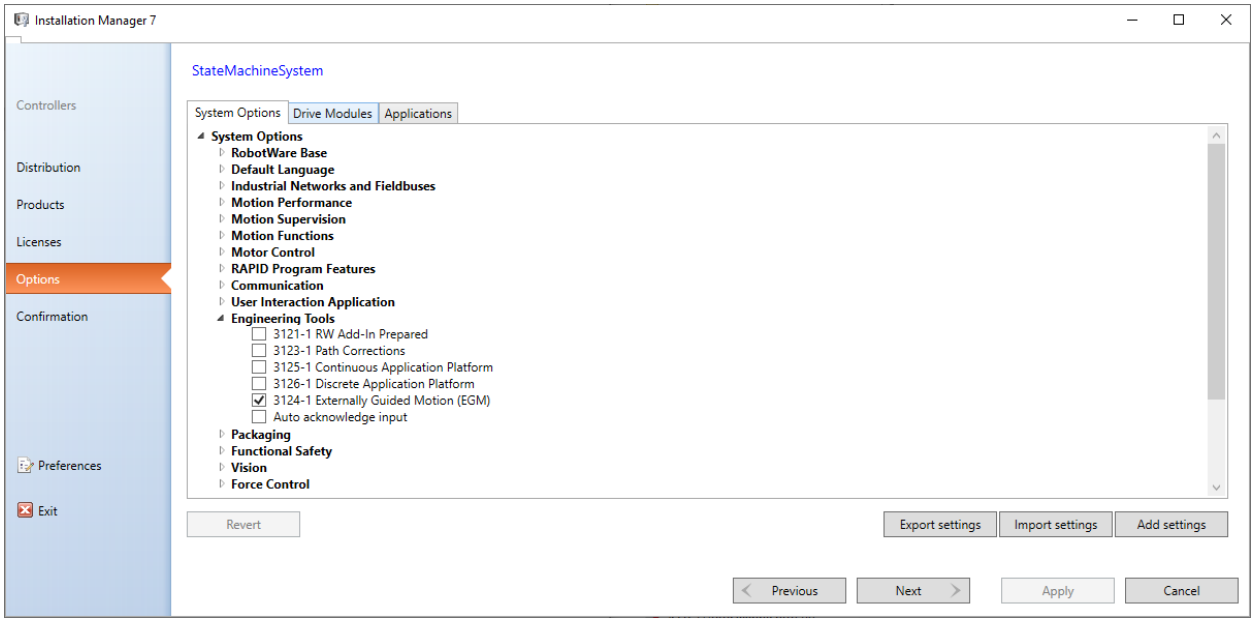

Figure 8: Select system options.

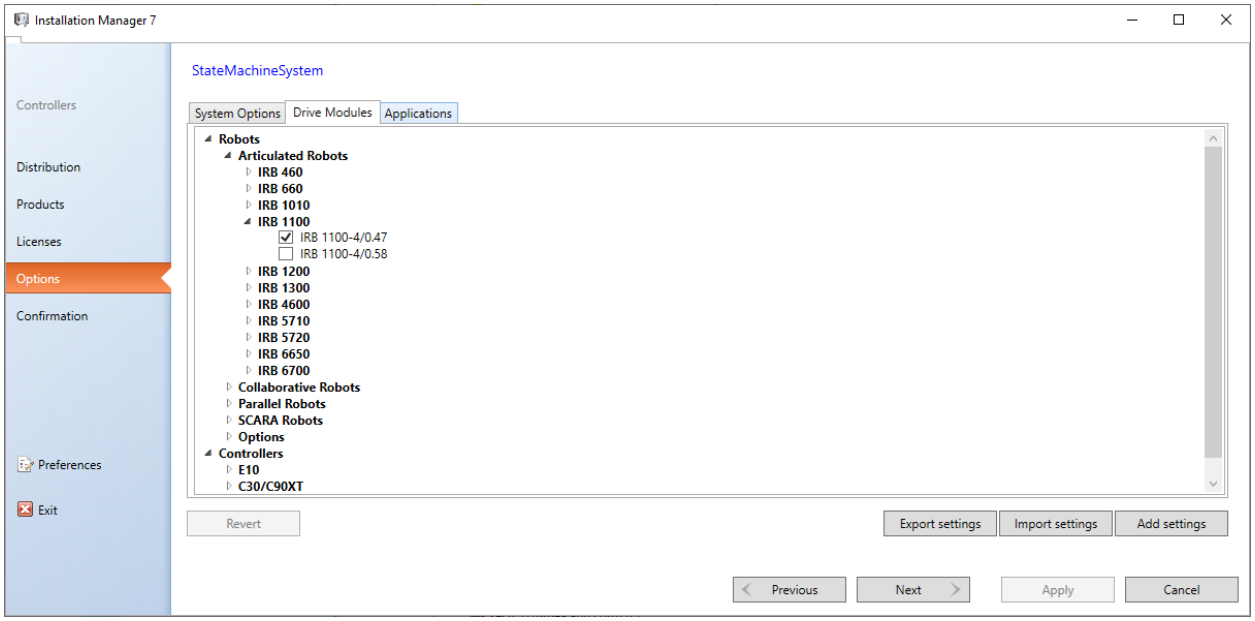

Figure 9: Select robot type.

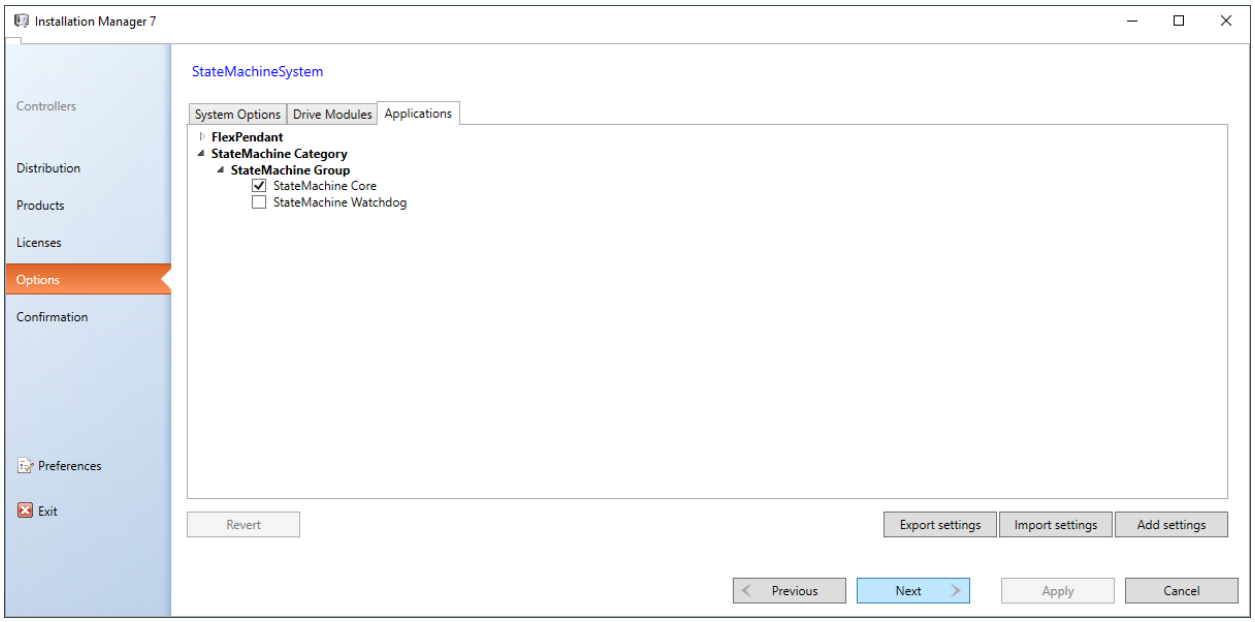

Figure 10: Select application options.

Step 6: Confirmation

Verify that all the desired settings have been made, then press Apply. Wait until the system installation has been finished. Finally, close the Installation Manager.

For a real robot controller, verify that all the desired settings have been made, then press Apply. Follow the instructions and wait until the system installation has been finished (can take a while). Finally, close the Installation Manager.

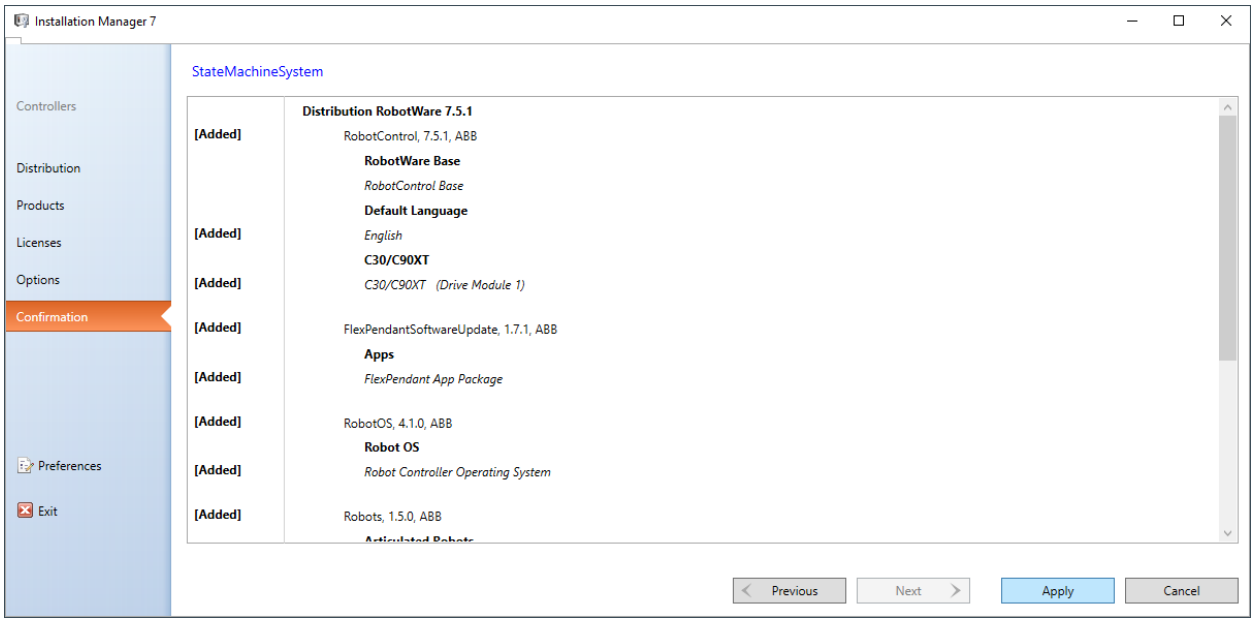

Figure 11: Confirm and apply the settings.

RobotStudio Simulation

To simulate a virtual controller, for example installed according to the **System Installation** section, it needs to be added to a RobotStudio station.

Step 1: Add Virtual Controller

Follow these steps to add the virtual controller to a RobotStudio station.

1. RobotStudio → File tab → New → Create an Empty Station (or a Solution with an Empty Station).
2. RobotStudio → Home tab → Virtual Controller → Exisiting Controller...
    a. Find the virtual controller → Press Ok.
    b. Wait until the virtual controller has been loaded into the station.

Step 2: Update System Configurations (Optional)

If using RWS, then it might be needed to specify the SSL certificate used for the HTTPS communication:

- RobotStudio → Controller tab → Properties → Manage Certificates → Replace the default certificate with the desired certificate (for example, a self-signed certificate).

If using EGM, then it is important to update the configurations for where the remote EGM server is located:

- RobotStudio → Controller tab → Configuration → Communication → UDP Unicast Device → Update all the ROB_X UDPUC instances (X depends on the number of robots in the system).
    - Remote Address and Remote Port Number are important to update.

If using RWS and/or EGM, then it might be needed to update the Firewall Manager configurations:

- RobotStudio → Controller tab → Configuration → Communication → Firewall Manager → Update the RobotWebServices and UDPUC network services.

After the configuration(s) has been updated, then the controller most likely need to be restarted before the new settings are applied.

Step 3: Run Simulation

Start a simulation by either:

- RobotStudio → Simulation tab → Play.
- RobotStudio → RAPID tab → Program Pointer → Set Program Pointer to Main in all tasks → Start.

When the simulation has started, then the state machine will begin with initialization, and then enter an idle state. Trigger a state request by, for example, setting any of the following IO-signals from 0 to 1:

- RUN_RAPID_ROUTINE.
- EGM_START_JOINT/ EGM_START_POSE/EGM_STOP (**only available if using EGM**).
- RUN_SG_ROUTINE (**only available if using SmartGripper**).

Tips:

- Trigger IO-signals manually with the I/O Simulator.
    - RobotStudio → Simulation tab → I/O Simulator.
    - It is useful to create a User List (Filter → User List → Edit Lists... → Add desired IO-signals).
- Activate the Operator Window (to see printouts from the state machine RAPID program).
    - RobotStudio → Controller tab → Operator Window.

Additional Material

External Documentation

See the online documentation[1] for a detailed description of the Robot Web Services 2.0 (RWS2) interface.

See the Application manual - Externally Guided Motion (document ID: 3HAC073318-001) for a detailed description of the Externally Guided Motion (EGM) interface.

Example Sketch

The following example sketch illustrates the relationship between:

- An ABB robot controller, running a RobotWare system using the StateMachine Add-In.
- An external computer with external RWS client and EGM server implementations.

[**Note**]: This sketch was made for the predecessor StateMachine Add-In 1.x versions, which are for RobotWare 6 systems. In RobotWare 7 systems, then the RWS communication is based on HTTPS instead of HTTP.

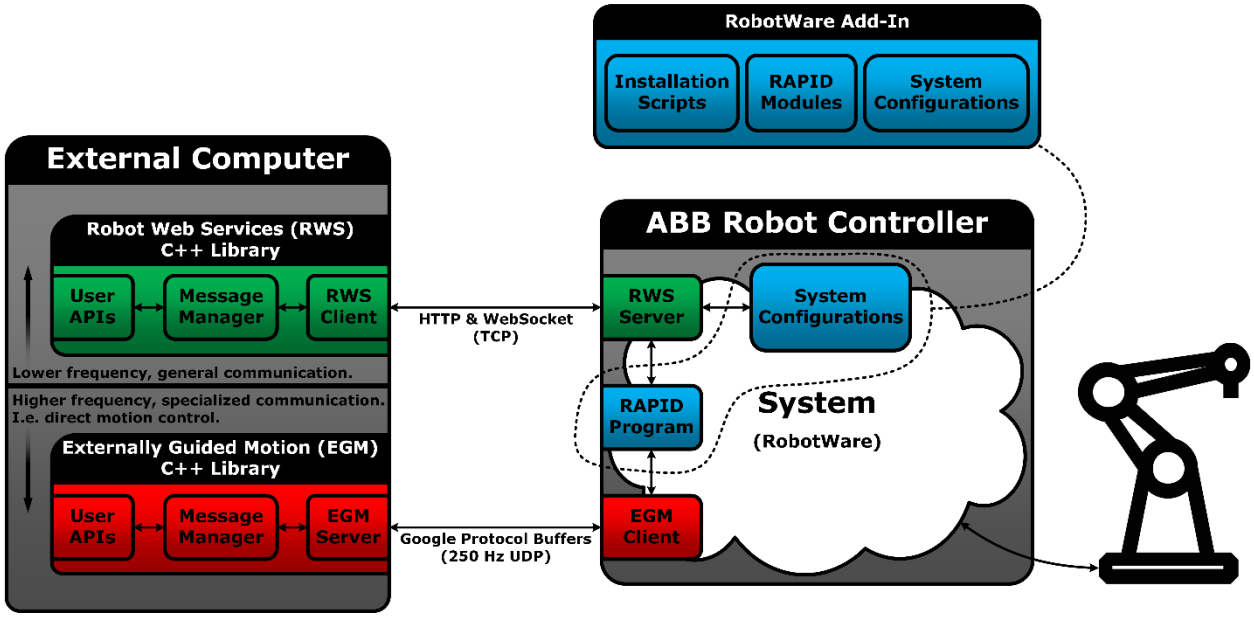

Figure 12: Example sketch for robot controller, StateMachine, RWS and EGM connections.

[1] https://developercenter.robotstudio.com/api/RWS

Acknowledgements

This Add-In is based on the predecessor StateMachine Add-In 1.x versions, which were mainly developed in the context of two European projects.

The core development was supported by the European Union's Horizon 2020 project SYMBIO-TIC[2].

The SYMBIO-TIC project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 637107.

Figure 13: Logo of SYMBIO-TIC.

The distribution process was supported by the European Union's Horizon 2020 project ROSIN[3].

The ROSIN project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 732287.

Figure 14: Logo of ROSIN.

[2] http://www.symbio-tic.eu

[3] http://rosin-project.eu