

SICK AppSpace SDK 1.6.0 release note



Revision history

Status	RELEASED
Version	1.0
Last update	16.02.2026
Revision comments	

Summary

SICK AppSpace - SDK is the Software Development Kit for the creation of application software for programmable sensors, released as an extension in the Visual Studio Code marketplace. This document pertains to the version 1.6.0, released on 16.02.2026 . It contains an overview of the important new features, improvements, fixes, remaining known limitations and issues, since version 1.5.0 from July 2025.

Table of contents

- [Highlights](#)
 - [Free usage](#)
- [Features](#)
 - [Develop & Execute](#)
 - [Device file system access](#)
 - [Device Console](#)
- [Changelog](#)
 - [1.6.0 - 16.02.2026](#)
 - [Features](#)
 - [Changes](#)
 - [Fixes](#)
 - [Removals](#)
 - [1.5.0 - 09.07.2025](#)
 - [Features](#)
 - [Changes](#)
 - [Fixes](#)
 - [1.4.0 - 10.03.2025](#)
 - [Features](#)
 - [Changes](#)
 - [Fixes](#)
 - [1.3.0 - 18.11.2024](#)
 - [Features](#)
 - [Changes](#)
 - [Fixes](#)
 - [1.2.0 - 27.03.2024](#)
 - [Features](#)
 - [Changes](#)
 - [Fixes](#)
 - [1.1.1 - 19.09.2023](#)
 - [Fixes](#)
 - [1.1.0 - 29.08.2023](#)
 - [Features](#)
 - [Fixes](#)
 - [1.0.3 - 01.02.2023](#)
 - [Fixes](#)
 - [1.0.2 - 10.01.2023](#)
 - [Fixes](#)
 - [1.0.1 - 14.12.2022](#)
 - [Fixes](#)
- [Roadmap](#)
- [Limitations](#)
- [Known issues](#)
 - [Workspace & building](#)
 - [Language server](#)
 - [UI-Builder](#)
 - [Device connectivity](#)
 - [Device / App Model](#)
 - [Device console](#)

- [Debugging](#)
 - [Device file system](#)
- [Supported hardware & software](#)
- [Installation](#)
 - [Components](#)
 - [System requirements](#)
 - [Installation instructions & first steps](#)

Highlights

Free usage

SICK AppSpace - SDK (free demo) is published in the VSCode Marketplace.

It is publicly available and usable for free. The user receives and agrees to a free demo license according to the [general terms and condition](#). The user further agrees to send telemetry data according to <https://www.sick.com/de/en/privacy-policy-data-processing-information-sick-appspace-tools/w/dataprotection-dataprocessing-information/>.

To productively use the SDK and enable advanced features, see the comparison below, the user may login with their SICK ID. Register a new SICK ID in just a few seconds here: <https://id.sick.com/>

Included in this release is a **getting started guide** which provides installation instructions and a **how-to build your first sensor app** (<https://github.com/SICKAG/SICK-AppSpace-SDK-Docs>).

For a complete onboarding training, checkout the Web-Based-Training, here: [SIA](#)

The user may download the SICK AppEngine(x64) for [Windows](#) or for [Linux](#) and use it for non-productive use in a demo mode.

Features

Develop & Execute

LUA language support is added to the editor. This includes support for SensorApp specific language features and device and algorithm APIs. With the UI-Builder, user interfaces for SensorApps are created in a structured view, just by drag and drop of UI Elements.

Execute the SensorApp directly to a connected programmable device.

Building the source code as a SensorApp package, completes the development journey.

Device file system access

The device file system is integrated with VSCode to provide easy access to files on the device.

Device Console

The device console shows the output of deployed SensorApps in VSCode and thus provides immediate feedback.

Changelog

1.6.0 - 16.02.2026

Features

- App details information on hover
- Automatic device reachability checks via configurable watchdog
- Folder of apps downloaded from a device can now be selected
- Aliases for devices in the device list
- Export apps as zip
- Automatic setting of app author on app creation
- Project quick deploy button on editor top bar
- New command to create flow files
- Visual indication for locked devices
- Single app workspaces
- Remembering network selection on successive device scans
- Reupload of all apps on a device
- Jump to function/event definition/reference capability
- Schema for flow files
- Visual error if device does not match device driver for connection
- Support unpacking of legacy MSDDs with source code
- Extended code navigation
- Display current active Lua version in the bottom status bar
- Added current Lua version to SAPK package
- Option to restart Lua Language Server via command palette
- Device hover information:
 - Displaying IP Address of device if connection address is a hostname
 - Device firmware version added
 - Lua version of device added
 - Displaying multi app debug capability of device

Changes

- Upgrade SCX runtime to version 3.2.0
- Faster adding of new devices
- Device list refreshing improved
- Removed device id's in favor of name in all user directed UIs
- "Device Console" renamed to "SICK AppSpace Device Console" for consistent naming
- Uses generation is now optional when building SPAK
- Added build processor config into project hash to better detect configuration changes
- Streamlined titles of commands, views, notifications,...
- Restart of languages server now available as command
- Recreating Lua API only on relevant manifest changes
- Lua version now only adjustable on User and Workspace level, not on Folder level anymore

Fixes

- Unreachable devices are no longer listed in the device filesystem
- App hashes are now correct for watched apps
- CROWNs with single-quoted entries can now be renamed correctly
- Successive deployment of protected apps are now possible without errors
- Only valid blocks are now displayed in the blocks editor (flows)
- Fixed USES information in SAPK manifest
- Create API documentation now also available for watched projects
- More robust app management
- Memory leak in SCX due to removed devices
- Reduced load on device to be less likely to crash COLA connections
- Device console more robust
- Type warnings when using Script.notify fixed
- Type lookup for global API elements fixed
- *Script.Queue* API code fixed
- More robust USES calculation for large workspaces
- USES calculation: USES now correctly added, even if no previous uses have been detected

Removals

- Lua doc settings via Language Server in favor of DocTool API documentation feature
- AppSpace subscriptions no longer displayed

1.5.0 - 09.07.2025

Features

- Blocks editor to modify control and data flow files
- Sick AppEngine emulator extension (*windows only*)
- Device console
 - Highlighting of console output
 - Linked files (e.g. *on Log.severe*)
- Automatic saving of project files on deployment
- SICK ID login offering on app deployment and SAPK build (*if not logged in*)
- Manifest documentation toolchain
 - Documentation of single apps via context menu
 - Option for local and bundled documentation for SAPKs during build process
- Full support for refactoring of CROWN paths, functions and events
- Create new apps from CSK templates
- "Go to code" CodeLens in manifest files
- Extracting of used CROWNs during app build for compatibility checks
- Deployment of protected apps is now possible
- Walkthrough for AppSpace SDK essentials
- Language Server:
 - Support for Lua 5.4
 - Support for "Go to code" Manifest CodeLens
 - New setting to customize connection timeout
 - Support for Sumneko language configuration

Changes

- Possibility to trigger CROWN path, function and event renaming in manifest files
- Allow an arbitrary number of projects to be watched
- Apps to be downloaded from device can now be selected
- Subscriptions are no longer offered and needed
- SICK ID login state gets not cached way longer
- Improved UX for importing samples, e.g. no errors anymore if sample is already downloaded
- Language Server:
 - Update to Sumneko 3.14.0
 - Possibility to trigger CROWN path, function and event renaming in script files
 - Removed support for documentation export in favor of manifest documentation support
 - Removed need to restart language server on every manifest change

Fixes

- SIM300 is now supported
- Breakpoints in scripts inside subfolders are now reachable on Windows
- Title of workspace tab now correctly changes
- Downloaded apps from device on Windows now have correct file paths in the zip file
- SICK ID Login is no possible even if login was abandoned before
- Exporting of logs now possible again on Windows
- Language Server:
 - Manifest switching now works even if math constants are not defined in the manifest
 - Consolidated output channels if language server is restarted
 - Updated language server state in status bar
 - Fixed annotations of math lib to reduce warnings
 - Diagnosis for *require* statements is corrected
 - Fixed minor memory leaks

1.4.0 - 10.03.2025



Starting with this release, VSCode is required in version 1.92 or above.

Features

- CROWN function & event renaming as beta feature (activate in settings)
- Drag n' drop of zipped apps into workspace pane imports them (not supported inside WSL)
- Welcome views for empty device and workspace list
- Support to deploy prebuilt msdd files (pages/customName.msdd)
- Jump into main script of app (context menu of application in workspace pane and command palette)
- New command to change device selection
- Possibility to deploy apps without selected device (selection will pop up)
- Extended command palette
- Language Server:
 - Support for ARM on Linux/macOS
 - Highlighting of CROWN traits
 - Display errors/warnings related to `require()` statements

Changes

- Updated device connectivity to SCX v1.3.10
- Merge icons to scan for devices and add device manually
- "Run without Debug" (CTRL + F5) no starts the apps on the device, if identical
- Increased performance of device list
- Icon for device page changed from globe to preview
- Clarified Device Monitor UI
- Updated Manifest XSD
- Language Server update to Sumneko 3.13.5

Fixes

- Debugging of threads is now possible
- App model configuration updates on app changes of device
- Improved error handling in case of device connectivity issues
- Full UTF8 support in device console
- Installation and removal of protected apps is now possible again
- Filesystem access for devices with low max buffer size is fixed
- Language Server:
 - Using strings for function parameters in combination with aliases do not result in type mismatches anymore
 - Manifest file is now correctly opened, if affected via code actions



Performance on first start after update

Caution on systems with anti virus: The internal update process, running after the first start after the extension upgrade, may take a long time (5 minutes and more).

The progress is logged in the output "SICK AppSpace Log (Device connectivity process)"

Update still running

```
[WARN] [AggregatedProfileImplementationRepository] The profile implementation with id will still be installed
```

Update finished

```
[INFO] [AggregatedProfileImplementationRepository] Profile Implementation with id created
```

If you additionally run into a Timeout *Extension activation failed* , see output "SICK AppSpace Log", please restart Visual Studio Code after the Update has finished, see Code block above.

Features

- Device Monitor
- Support for dockable api in AppEngine
- SICK ID integration
- Import of CSK Modules
- Load all apps from a device
- Import manifest of selected device for offline usage
- Remove imported offline manifests
- Debug features (only if supported by device):
 - Multi app debugging (Only with active subscription, subscription only available for SICK internals)
 - Step Out
 - Return values of debug code execution (Debug console -> *return xyz*)
 - Support to set variables via variables tab in debug view
- Sail UI builds (experimental)
- Preview of sail build UIs

Changes

- Debugging now needs the user to be logged in via SICK ID (no subscription needed)
- Harmonized device login
- Device login now only shows relevant subset of userlevels
- Ctrl + F5 no longer starts the debugger
- Checksum calculation of SAR files now matches the calculation on the device
- Deprecation of *license.xml* in favor of SICK ID
- UIBuilder - ControlLibraries:
 - *Update to @sick-davinci/basic-elements@7.2.3*: minor styling changes, that affect e.g. height and width of the elements
 - *number-input* type of value changed from *string* to *number*
 - *FileUploadButton* payload of *finished* event changed from *boolean* to *{success: boolean, path: string}*

Fixes

- Restored support for images in visual debugger
- Manifest changes are now correctly applied to build app
- Hash calculation of app fixed



Starting with this release, VSCode is required in version 1.85 or above.

Features

- "Home view": Central view for starters with useful links & quick access to some functions
- Create zip with log files for bug reports
- Import apps from zip
- Import app samples from GitHub
- "Parameter Editor": XML-based editing of Parameter files (*.cid.xml) with an intelligent outline view and wizard-like editing capabilities.
- Display the LanguageServer state with the possibility to restart it
- Show the user-defined name "LocationName" in the device list, if it has been set
- Partial app transfer - If an app is already installed on a device only the changes will be transferred

Changes

- Minor usability improvements
 - Disable/hide buttons when they cannot be used
 - Align "Workspace explorer" behavior with VSCode (if there is only one workspace folder open, the projects will be rendered as a flat list)
 - All file chooser now have a title on Linux systems

Fixes

- LanguageServer
 - Stability improvements
 - Detection of concurring extensions (Sumneko LanguageServer)
- Image not displayed in the debugger ImageView
- Adding / Removing workspace folders not being detected

1.1.1 - 19.09.2023

Fixes

- Lua support (LanguageServer) not working with VSCode >= 1.82

1.1.0 - 29.08.2023



Due to the update of the "Device connectivity services" a reinstallation of those services is required. The device configuration (device list) will be lost.

See "known issue [device list empty after update](#)"



We replaced the formatter that is used for Lua files. Therefore you might have to update your configuration.

See "known issue [lua format options not used](#)"

Features

- Multiple instances of VSCode are now supported
- Debugging
 - See line execution time
 - View & traverse the callstack (AppEngine >= 3.0.0)
 - View local variables
 - View global variables
 - Use "watch" (Expressions are not supported)
 - 2D/3D image viewer (experimental)
- App development & Packaging
 - Jump to CROWN from the CROWN viewer
 - SAPKs now contain build information about the tooling and build-environment
- AppManagement
 - Start, stop and delete apps on a device
- UI-Builder
 - Update to @sick-appspace/uibuilder@3.5.0, @sick-appspace/uibuilderservice@6.0.0, @sick-appspace/controllibrary@4.0.1
 - The control library will now be extracted to AppData, so it won't be replaced on updates

Fixes

- Device connectivity
 - Closing VSCode while installing the device connectivity services no longer results in broken installation
- Device file system
 - Renaming directories
- App development & deployment
 - Manifest changes of newly created apps are now handled
 - App names are now determined by the project.mf.xml instead of the directory name
 - App deployment issues with slow devices
 - Packaging SAPKs that include apps with "big" or very many files
- Lua language support
 - Code completion now works on empty lines after comments
 - Code completion issues when typing too fast
 - Diagnostics & completion for standard lua functions

1.0.3 - 01.02.2023



Currently updates of the "Device connectivity" extension results in a loss of the device list/configuration. The device list is not automatically migrated, but you can restore it.

Fixes

- Fix occasional crash of the device communication services
- Fixed bug that caused the AppSpace SDK to load indefinitely
- Fixed bug that resulted in huge log files
- Fix some issues after a device list refresh, when the selected device is no longer reachable

1.0.2 - 10.01.2023

Fixes

- Fixed bug, preventing the SICK AppSpace Programmable Core plugin to activate. For users having such issues in 1.0.1: Please update first to 1.0.2, then delete the following folder:

```
windows: %appdata%\SICK/AppSpace/MaxwellConnect
linux: ~/.config/SICK/AppSpace/MaxwellConnect
```

Be aware that this removes all your devices from the device list. We apologize for this inconvenience.

1.0.1 - 14.12.2022

Fixes

- Correct and faster manifest download
- Loosen file name restrictions
- Errors in "Redhat XML" no longer prevents start of the SICK AppSpace SDK
- Trigger manifest change for new apps
- Generate "desc" attribute for served function parameters (was shown as an error before)

Roadmap

Limitation	Description
CLI	A Command Line Interface is planned to be released in a future version.
Usability	Settings and detailed configurations are based on text file. Depending on user feedback, we may offer user interface for certain configurations.

Limitations

Limitation	Description
Protect SensorApp	Protection settings for your SensorApp to restrict reading the source code by others or duplicating the app from one sensor to another are currently not available. The source code of all sensor apps build with this version are readable by others.
Import app from device	Only unprotected apps can be imported from a device.

Known issues

Workspace & building

Issue	Description	Workaround
Build watcher does not register changes	The watcher does NOT work for symlinked files it only works for directories.	If possible, include the parent directory as symlink.
Multiple build triggered	After removing a symlinked directory, the VSCode watcher still triggers the change event as if the symlink was never removed. This shouldn't have any bad side effects, except for performance (unnecessary rebuilds)	Restart VSCode
Name conflicts in project manifest are not validated	Name conflicts in the project.mf.xml are not displayed as error to the users and are not validated when building SAR or SAPK files. This can lead to unpredictable behavior.	None
Building SAPK fails	Deploying apps to a device or building SAPK fails. In the SICK AppSpace Log output channel a message shows up: "Failed to build msdd" This error message is caused by a corrupted UIBuilder control-library	<ol style="list-style-type: none">1. Close vscode.2. In the file system, navigate to %APPDATA%\SICK\AppSpace\ui-builder\control-library and delete the folder with the version number, eg. '5.0.2'.3. Restart vscode. The UIBuilder will reinstall the control-library, this takes a moment.

Language server

Issue	Description	Workaround
Custom path structure not loaded	Directories defined by <i>Lua.workspace.library</i> are not loaded.	A further <i>scripts</i> folder must be placed in which the Lua code must be placed.
No effect of certain <i>Lua.format.*</i> options	The old formatter was replaced by a new one including the settings options.	Use <i>Lua.format.defaultConfig</i> instead.

UI-Builder

Issue	Description	Workaround
Missing parameter bindings	"Parameter Bindings" work only with parameters in the workspace. Parameters from the device are not loaded.	
Unexpected behavior on shortcut	Changed keyboard shortcuts are not respected by the UI-Builder.	If changed one should add <i>activeCustomEditorId != 'sickag.ui-builder-plugin.HtmlEditor'</i> to the <i>When</i> condition in Keyboard Shortcuts.
Unexpected behavior with edit menu	It is not recommended to use VSCode menu Edit > operations with UIBuilder as <ul style="list-style-type: none">◦ Cut-copy-paste operates on VSCode's history and not the UIBuilder's.◦ Other operations should have no effect.	Use the built-in functionality of the UI-Builder.
No log output for broken MSDD builds	There is no error log in OUTPUT > SICK AppSpace Build, if your .msdd build breaks.	Open Help > Toggle Developer Tool to display UI logs.
App not updated though build process changed	App change not detected when switching msdd build from legacy sopasjs to new sail build	Delete app from device and retry

<p>Issues with sick-viewer2d and sick-viewer3d</p>	<ul style="list-style-type: none">• click "Options" then click "Fullscreen" options menu is mispaced• click "Fullscreen" type errors "Element is not connected" and "Permissions check failed" may appear	
--	--	--

Device connectivity

Issue	Description	Workaround
"Failed to activate extension"	Possible reasons: <ol style="list-style-type: none">VSCode has been closed during the installation of the device connectivity services. The installation might be corrupt.	If the device connectivity services fail to start, a webview should open and perform several diagnostics. Try to resolve the issue with the offered solutions (buttons). Possible solution (manual): Restart VSCode if this does not help, reset SickConnectX: <ol style="list-style-type: none">Check the log of the device connectivity (see output channel) The blocked ports should be listed in the log.Delete the following folder:<ul style="list-style-type: none">Windows: '%APPDATA%/SICK/AppSpace/MaxwellConnect/instances/scx-runtime-3.0.1'Linux: '~/config/SICK/AppSpace/MaxwellConnect/instances/scx-runtime-3.0.1'Restart VSCode
Device list empty after update	Currently updates of the "Device connectivity" results in a loss of the device list/configuration. Migration from SDK 1.5.0 to ≥ 1.6.0 is currently not supported	Populate the device list again
WSL: Devices are not reachable	USB Support might be blocked by the IT device connectivity refuses to work	Disable USB capability of the device-connectivity by setting the environment variable <code>NASRT_ENABLE_USB</code> to false, e.g. via <code>.bashrc</code> : <code>echo "export NASRT_ENABLE_USB=false" >> \$HOME/.bashrc</code> Or via windows variables (this is mandatory if VSCode is not started via WSL terminal!): <code>NASRT_ENABLE_USB=false</code> <code>WSLENV=NASRT_ENABLE_USB</code>

Device / App Model

Issue	Description	Workaround
Device manifest not loaded	Device must be in the same network and accessible from the current device.	<ul style="list-style-type: none">Ensure device is in the same networkAdd device IP to <code>NO_PROXY</code> environment variableRemove and add the device again
Device manifest not loaded (SAE 1.6.x, - 1.8.0)	The device manifest can currently not be loaded from those SAE versions	None

Device console

Issue	Description	Workaround
No output received anymore for selected device	The device console stopped printing console messages	Internal connection to the device lost (SCX does not support an event for this at the moment) 1. Unselect the device 2. Select the device again
No output printed after (re-selecting) device	The device console does not print any more messages after following these steps: 1. View device console 2. Unselect device from list 3. Select device again	Restart VSCode Note As the device connectivity is shared between all instances, all VSCode instances with active SDK have to be restarted.
No device console output of any selected device	No matter which device is connected or how often VSCode was restarted, no output is printed	Reset SickConnectX: 1. Close all VSCode instances 2. Delete your MaxwellConnect instance Linux: '~/config/SICK/AppSpace/MaxwellConnect/instances/scx-runtime-3.0.1' Windows: '%APPDATA%/SICK/AppSpace/MaxwellConnect/instances/scx-runtime-3.0.1' 3. Restart VSCode Device Connectivity will be reinstalled

Debugging

Issue	Description	Workaround
2D/3D viewer shapes are drawn incorrectly		None
Debugging does not start	After starting the debugger VSCode stays in a loop indicating the debugger is starting but nothing happens	Reset SickConnectX: 1. Close all VSCode instances 2. Delete your MaxwellConnect instance Linux: '~/config/SICK/AppSpace/MaxwellConnect/instances/scx-runtime-3.0.1' Windows: '%APPDATA%/SICK/AppSpace/MaxwellConnect/instances/scx-runtime-3.0.1' 3. Restart VSCode Device Connectivity will be reinstalled

Device file system

Issue	Description	Workaround
Create new file on device file system shows error	When creating a new file on the "Device filesystem" an error message is shown. The file is created regardless, but not shown in the UI until refresh.	Refresh the "File explorer" view after the error message appeared.
Files with special symbols in name are no longer accessible	If you create or rename a file with special non-ASCII symbols in their name, they will no longer be accessible.	None. Be cautious when choosing file names.

Supported hardware & software

This release supports the following SICK AppSpace-enabled devices:

Product family	Part number	Device type
SICK AppEngine	1613796	SICK AppEngine (x64)

InspectorP61x	1116350	V2D611P-MMSCE4
	1114809	V2D611P-MMSBE4
InspectorP62x	1110847	V2D621P-2MSFBB5
	1110848	V2D621P-2MSFFB5
	1110849	V2D621P-2MDFGB5
InspectorP63x	1082298	V2D631P-2MXCXB0
	1082299	V2D631P-2MXSXB0
	1082300	V2D632P-2MXCXB0
	1082301	V2D632P-2MXSXB0
InspectorP64x	1082302	V2D642P-2MCXXA6
InspectorP65x	1082303	V2D652P-2MCXXA6
	1082304	V2D654P-2MCXXA6
	1082305	V2D652P-2MEWHA6
	1082306	V2D654P-2MEWHA6
MRS1000P	1104278	MRS1104P-111011
RFU61x	1091102	RFU610-10600
	1099890	RFU610-10601
	1101394	RFU610-10605
	1104441	RFU610-10614
	1104443	RFU610-10603
	1104444	RFU610-10604
	1104445	RFU610-10613
	1104446	RFU610-10610
	1104447	RFU610-10607
	1104448	RFU610-10618
	1104449	RFU610-10609
RFU62x	1062599	RFU620-10100
	1062600	RFU620-10400
	1062601	RFU620-10500
	1062602	RFU620-10101
	1062603	RFU620-10401
	1062604	RFU620-10501
	1068727	RFU620-10107
	1068728	RFU620-10105
	1069453	RFU620-10503
	1069677	RFU620-10104
	1070407	RFU620-10504
	1077860	RFU620-10505
	1077863	RFU620-10514
	1083557	RFU620-10510

	1083976	RFU620-10507
	1084997	RFU620-10111
	1086439	RFU620-10110
	1088871	RFU620-10508
	1091355	RFU620-10103
	1092037	RFU620-10112
	1094605	RFU620-10108
	1096414	RFU620-10114
	1101686	RFU620-10118
	1101700	RFU620-10102
RFU63x	1054396	RFU630-13100
	1054397	RFU630-13101
	1057943	RFU630-13105
	1058117	RFU630-04100
	1058775	RFU630-13102
	1059999	RFU630-04101
	1061498	RFU630-13107
	1067133	RFU630-13106
	1067473	RFU630-13103
	1068569	RFU630-04106
	1068726	RFU630-13104
	1070903	RFU630-13108
	1070904	RFU630-04108
	1073196	RFU630-04105
	1073376	RFU630-04102
	1073377	RFU630-04109
	1073442	RFU630-13110
	1074302	RFU630-13112
	1077861	RFU630-13113
	1077862	RFU630-13111
	1083558	RFU630-13115
	1087776	RFU630-04117
	1093152	RFU630-04104
	1095224	RFU630-13114
	1104670	RFU630-04103
RFU65x	1073556	RFU650-10100
	1076522	RFU650-10101
	1083559	RFU650-10105
	1083560	RFU650-10106
	1087587	RFU650-10102

	1092036	RFU650-10104
	1096413	RFU650-10103
SID	1098321	SID120
	1101360	SID70
SIM10xx	1097816	SIM1000-0P0B100
	1098146	SIM1012-0P0G200
	1111314	SIM1012-0P0G200S01
	1098148	SIM1004-0P0G311
SIM2x00	1080579	SIM2000-0A10A00
	1081902	SIM2000-2P04G10
	1092673	SIM2500-2P03G10
SIM10xx Flexi Soft	1097817	SIM1000-0P0B110
SIM4x00	1078787	SIM4000-0P03G10
TIM8xxP	1090292	TIM881P-2100101
TriSpectorP1000	1091318	V3T11P-MR12A8
	1091319	V3T12P-MR32A8
	1091320	V3T13P-MR62A8
	1091321	V3T12P-MR32A7
	1091322	V3T11P-MR12A7
	1091323	V3T13P-MR62A7
Visionary-T AP	1102953	V3S140-2AAAAAA
	1102954	V3S140-2AABAAB
Visionary-S AP	1114319	V3S142-1AAAAAA
	1114320	V3S142-1AABAAB

Installation

Components

SICK AppSpace SDK	1.6.0
Code samples	https://gitlab.com/sick-appspace/samples
UI Builder	3.5.0, WYSIWYM and web programming tool for creating app UIs, based on DaVinci basic elements 5.6.4

System requirements

OS	Windows 10 x86_64 Windows 11 x86_64 Ubuntu 22.04 x86_64
HDD	min 1 GB
RAM	min 4 GB, 8 GB recommended
Processor	1 GHz CPU
Browser	Chrome for support of WebGL and WebSockets
IDE	Microsoft Visual Studio Code v1.92

Installation instructions & first steps

Please find installation instructions and first steps tutorials here: <https://github.com/SICKAG/SICK-AppSpace-SDK-Docs>