# MCS2

## Getting Started Guide


www.smaract.com

SmarAct



The information given in this document was carefully checked by our team and is constantly updated. Nevertheless, it is not possible to fully exclude the presence of errors. In order to always get the latest information, please contact our technical sales team.

SmarAct GmbH, Schuette-Lanz-Strasse 9, D-26135 Oldenburg
Phone: +49 (0) 441 - 800879-0, Telefax: +49 (0) 441 - 800879-21
Internet: www.smaract.com, E-Mail: info@smaract.com

*Document Version: 1.0.6*

# *GETTING STARTED*

This document is a getting started guide for the SmarAct Modular Control System 2 (MCS2). It describes the installation of software and guides the user through the first steps of working with the SmarAct MCS2.

## 1.1 What Is New?

The SmarAct MCS2 is the follow-up of the MCS controller system. It has several new features and improvements. The following table gives an overview of the new features and the differences to the previous MCS system.

Table 1.1 – Feature Summary

| Feature / Component | Description |
|---|---|
| Trajectory Streaming | Participating positioners can be moved synchronously along a defined trajectory |
| Command Groups | Grouping several commands to be triggered by software or external trigger (via I/O modules) |
| "Quiet Mode" | Actuator mode for noise-sensitive applications |
| PicoScale Support | SmarAct PicoScale laser interferometer can be used as high precision sensor module |
| Hand Control Module | User-friendly 3.5“ touch-display on hand control module (HCM) |
| Hot-Plugging | Hot-plugging of the hand control module |
| Auxiliary I/O | Analog / digital inputs and outputs (via I/O modules) |
| Auxiliary I/O | Analog inputs may be used as feedback for the control-loop |
| Auxiliary I/O | Digital input trigger (emergency stop, trajectory streaming synchronization, command group trigger) |
| Auxiliary I/O | Digital outputs with position compare unit |
| Control loop | Fast control loop with 50kHz |
| Control loop | User accessible PID control loop tuning (custom positioner types) |
| Move Control | Following error detection feature to inform the application if a commanded trajectory cannot be followed by a positioner |

*Continued on next page*

Table 1.1 – *Continued from previous page*

| Feature / Component | Description |
|---|---|
| Sensor Power Mode | The sensor power mode may be configured individually for each channel |
| API | Property based application programming interface (API) |
| API | Synchronous and asynchronous property access can be mixed in one session depending on the individual requirements |
| API | Event notification system |
| Base Unit | Base unit for position values is pico meters (pm) for linear positioners and nano degree (n°) for rotatory positioners |
| Rotary Positioners | The position of rotary positioners is given by a signed absolute value instead of the combination of angle and revolution |
| Referencing | On-the-fly referencing mode without the need to stop or return to the physical reference mark allows to easily determine the reference position of mechanically coupled positioners |
| Movement | Movement parameters like holdtime, open-loop frequency and amplitude, etc. are set independently via separate properties instead of as explicit parameters of the movement command. |
| Update | In the field firmware update and feature upgrade system |

## 1.2 Software Installation

The MCS2 software installers contain all software and documentation needed to use the MCS2 and to develop your own software.

Software installers are available for Windows™ (Windows™ 7 or higher) and Linux operating systems. Note that the service and firmware uploader tools are currently only available for Windows™.

The software installers can be found on the software CD or USB flash drive you received with your MCS2 device.

### 1.2.1 Software Installer

The following descriptions refer to the Windows™ software installer.

**NOTICE**
Note that administrative privileges on the PC are required to run the software installer.

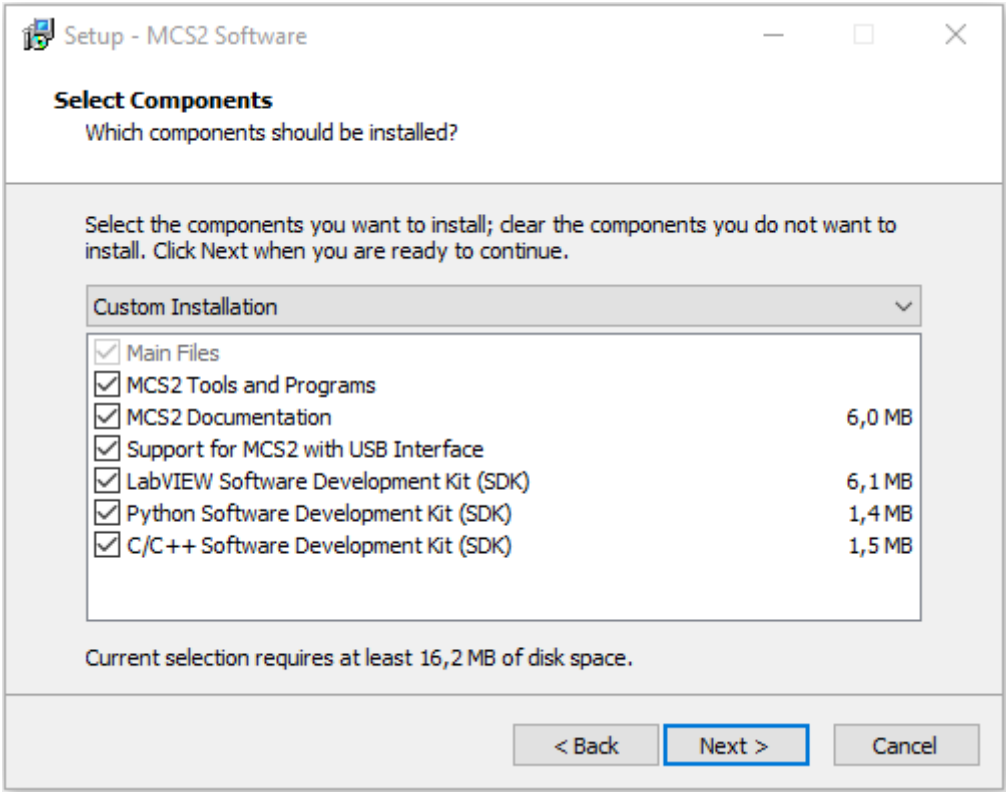


Figure 1.1: MCS2 Software Installer

A configuration wizard guides you through the installation process and allows to select different software components for installation. After starting the installer a dialog informs about changes and updates of the programs and documentation files. The following dialog allows to select which components should be installed:

- Select **"MCS2 Tools and Programs"** to install utilities like the firmware uploader and service tool.
- Select **"MCS2 Documentation"** to install the documentation files like user manuals and programmers guide.
- Select **"Support for MCS2 with USB Interface"** to enable the installation of drivers needed to connect to MCS2 devices with USB interface. If the option is selected the installer will ask for permission to run the Usb-To-Serial driver installer at the end of the installation process. The driver can also be installed manually by running the program "CDMxxxx_Setup.exe" in the "Drivers" folder on the software CD or USB flash drive. Note that USB support is not needed for devices with ethernet interface.
- Select **"LabVIEW™ Software Development Kit (SDK)"** for developing applications with LabVIEW™. If more than one LabVIEW™ version is installed you can choose the version to install the LabVIEW API in one of the next steps. To install for additional LabVIEW™ versions just run the installer again and select another LabVIEW™ version (the other software doesn't have to be installed again). Note that the LabVIEW™ IDE should be closed to let the installer create the VI palettes. The installed VIs can be used with LabVIEW™ 32-bit and 64-bit development systems.

- Select **"Python Software Development Kit (SDK)"** to install the "SmarActCTL" Python package and SDK for developing applications with Python. You can choose in one of the following steps if the package should be installed automatically using the Python package manager (pip). Note that this may require an internet connection in case that support packages must be installed or updated. Alternatively, the Software Development Kit (SDK) may be installed only. In this case the package will not be installed but will be available in the SDK folder as a pip-compatible ZIP file for manual installation.
- Select **"C Software Development Kit (SDK)"** to install libraries, header files and programming examples for writing your own programs in C or C++ (or other languages that are compatible with the programming libraries).

All files will be installed to the default location: C:/SmarAct/MCS2.

### 1.2.2 Uninstalling the Software

An uninstaller program can be found in the SmarAct software folder in C:/SmarAct/MCS2/Uninstall. It removes all MCS2 software from your PC except for the Python package which is not uninstalled automatically from your local Python installation. Other SmarAct software is not removed.

### 1.2.3 Documentation

All documents are installed in C:/SmarAct/MCS2/Documentation.
The following documents are available:

Table 1.2 – Documentation

| Document | Description |
|---|---|
| MCS2HCMUserManual.pdf | The user manual for the MCS2 hand control module (HCM). |
| MCS2PositionerTypes.pdf | This document lists all currently supported positioner types for the MCS2 controller. A cross-reference allows to find the correct positioner type according to the SmarAct positioner series or sensor type. |
| MCS2ProgrammersGuide.pdf | The programmers guide mainly describes the application programming interface (API) used to develop control applications for the MCS2. But it also supplies background information and explains general concepts of the MCS2. Therefore it may be of interest for non-programmers too. |
| MCS2ServiceToolManual.pdf | The user manual for the MCS2 service tool. |
| MCS2UserManual.pdf | The user manual contains safety notes for the operation of the MCS2. It gives a product overview of the different main controllers and available modules. Furthermore the technical and electrical specifications and connector pin assignments can be found here. |

*Continued on next page*

Table 1.2 – *Continued from previous page*

| Document | Description |
|---|---|
| SmarActNetConfigManual.pdf | A user manual for the SmarAct network configuration tool. |
| Using a SmarAct Python SDK.pdf | This document describes the Smaract Python APIs. It gives information about the installation and requirements of the Python package and shows the differences between the C and the Python API. |

Additionally, the sub folder UserGuides contains several documents that describe specific use cases of the MCS2.

### 1.2.4 Software Tools

Software tools for the MCS2 are installed in C:/SmarAct/MCS2/Programs.
The following tools are available:

Table 1.3 – Software Tools

| Program | Description |
|---|---|
| MCS2ServiceTool | The MCS2 service tool may be used to configure MCS2 devices. The tool also shows device information like firmware versions, etc. and may be used to perform automatic diagnostic routines. |
| SmarActFirmwareUploader | The firmware uploader tool allows customers to update the firmware of a device to apply bug fixes and feature upgrades. |
| SmarActNetConfig | The network configuration tool allows to configure SmarAct devices with ethernet interface. The device discovery may be used to find the IP addresses of devices in the network. |

## 1.3 Setup and Configuration

### 1.3.1 Setting up the Device

Before initial operation be sure to read the safety notes in the first chapter of the "MCS2 User Manual". The user manual also gives a step-by-step guide to set up the device.

While devices with USB interface do not need any interface configuration, the ethernet interface must be configured to match the users network settings. The configuration of the network parameters is described in the "MCS2 User Manual".

Note that in contrast to previous SmarAct controller systems the MCS2 supports hot-plugging of the Hand Control Module.

The following figure shows the structural setup of a typical MCS2 system.

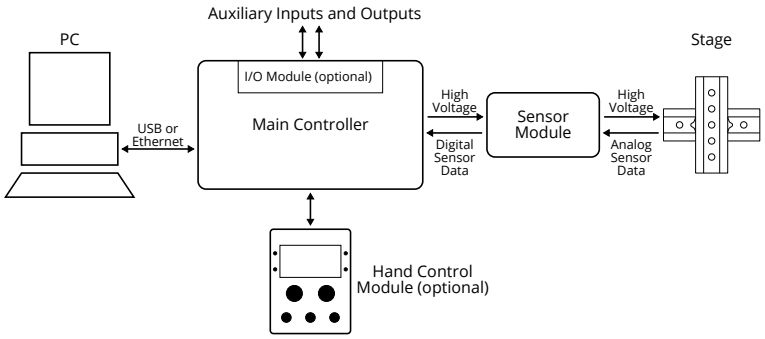


Figure 1.2: Structural Setup of an MCS2 Device

### 1.3.2 Positioner Type Configuration

During the configuration of the device each channel must be configured with the type of positioner that is connected to the channel. The **positioner type** implicitly gives the controller information about how to calculate positions, handle the referencing, configure the control-loop, etc.

While the positioner type is usually configured as part of the factory end-test of the system it may be necessary to reconfigure it in some cases. E.g. a firmware update resets the type to a default. Changing the physical setup may also make it necessary to update the positioner types.

**NOTICE**
When the positioner type of a channel is changed, the channel must be calibrated to ensure proper operation of the positioner.

The "MCS2 Service Tool" may be used to configure the positioner types. Please refer to the "MCS2 Positioner Types" document for a list of available positioner types.

**NOTICE**
When using positioners with **M- or L-sensor** on *at least one* channel of a specific driver module *all* positioner types of *this* module must be set to a **M- or L-sensor** type too to configure the correct sensor supply voltage. This rule applies even if the other channels of the driver module are unused, respectively no positioners are connected.

### 1.3.3 Positioner Calibration

Even though every positioner is categorized by its type (which is configured to the channel via the positioner type) each individual positioner may have slightly different characteristics that require the tuning of some internal parameters for correct operation and optimal results. The **calibration sequence** is used to adapt to these characteristics and automatically detects parameters for an individual positioner. It must be executed **once for each channel** if the mechanical setup changes

(different positioners connected to different channels). The calibration data will be saved to non-volatile memory. If the mechanical setup is unchanged, it is not necessary to run the calibration on each power up, but newly connected positioners have to be calibrated in order to ensure proper operation.

The "MCS2 Service Tool" or the MCS2 hand control module may be used to perform this calibration sequence. Refer to the "MCS2 Service Tool Manual" and "MCS2 HCM User Manual" for more information on the usage and features of the service tool and hand control module.

#### Positioners with Multiple Reference Marks

Some positioners are equipped with multiple reference marks. The advantage of these distance coded reference marks is that the referencing sequence requires less positioner movement to detect the absolute position. The referencing sequence measures the distance between any two neighboring reference marks in order to determine the physical position.

**NOTICE**
In some cases the detection of the physical position might yield false results. In these cases a special calibration routine must be performed to correct the absolute position calculation when referencing positioners with multiple reference marks. Make sure to execute the **"Detect Dist Code Inversion"** routine of the "MCS2 Service Tool" for these positioners.

#### Positioners with Limited Travel Range (Micro Grippers)

The signal correction calibration sequence requires positioner movement to correct the sensor signals of the position sensor. Some positioners (e.g. **micro grippers**) have a very limited travel range. For these positioners the movement distance may be too small to successfully finish the calibration. The calibration sequence automatically detects the end of the travel range ("end stop") and reverts the movement direction to continue the calibration in the opposite direction. If more than one end stop is detected the calibration sequence is aborted with an error. While the default of one end stop is suitable for the majority of positioners, the **"Limited Travel Range"** calibration option may be used to increase the number of allowed end stops for micro grippers. The calibration sequence then moves back and forth between the two end stops to perform the signal corrections.

The "MCS2 Service Tool" may be used to start the calibration with the "Limited Travel Range" option.

### 1.3.4 Firmware Update

The MCS2 controller is equipped with a bootloader system that allows customers to update the firmware of the device to apply bug fixes and feature upgrades. Please refer to the user guide "MCS2-UG00001_PerformingAFirmwareUpdate" for detailed information about the firmware update process. In order to get the latest firmware please contact the SmarAct technical sales team.

### 1.3.5 Troubleshooting

In case the MCS2 controller or a SmarAct positioner does not react as expected always check the following steps:

- check all cable connections
- make sure that the correct positioner type is configured for a channel
- perform the calibration sequence

If the calibration generates an error or the unexpected behavior lasts, the "MCS2 Service Tool" offers diagnostic tests to perform device and positioner tests. These tests perform some positioner movements and generate a log file which may be sent to the SmarAct technical support team for further analysis. The service tool also shows hardware and firmware versions and serial numbers of a connected device, which may be requested by the SmarAct support team. Please refer to the "MCS2 Service Tool Manual" for more information.

## 1.4 Developing Software for the MCS2

When starting to program your own software make yourself familiar with the "MCS2 Programmers Guide". The programmers guide supplies some background information for a better understanding of the overall system and covers general concepts of the MCS2. The function and property references describe all available API functions and properties. Every function and property description also includes a short programming example which may be used as a code-template for your own program.

The software installer offers options to install the Software Development Kit (SDK) for different programming languages. Currently, C/C++, Python and LabVIEW™ SDKs are available. By selecting one or more SDK options, folders for the respective SDK are created in the C:/SmarAct/MCS2/SDK folder which contain programming examples and additional components needed to develop software for the MCS2.

**Programming Examples**

The fully functional programming examples for C / C++, Python and LabVIEW™ allow the developer a quick start into the application programming with the MCS2.

All examples are designed to build upon each other. This way programming concepts and details are explained step-by-step. Therefore it is recommended to work through the examples starting with the first one.

### 1.4.1 C/C++ Software Development Kit (SDK)

The Software Development Kit (SDK) for C/C++ software development consists of the following files:

- "SmarActCTL.lib" library
- "SmarActControl.h" and "SmarActControlConstants.h" header files
- "SmarActCTL.dll" dynamic library
- "SmarActIO.dll" dynamic library
- Programming examples for C/C++

The SmarActCTL dynamic library is available for Windows™ and Linux 32bit and 64bit operating systems. The **calling convention** is *cdecl*.

All files of the C SDK are installed to the C:/SmarAct/MCS2/SDK/C folder. The path of the C SDK is defined in the environment variable "MCS2_SDK" which can be used in development tools to find the SDK folder.

### 1.4.2 Python Software Development Kit (SDK)

The Software Development Kit (SDK) for Python consists of the following files:

- a pip compatible ZIP archive of the "smaract.ctl" package
- Programming examples for Python

Python version 3.4 or higher is required for SmarAct Python packages. Please refer to the "Using a SmarAct Python SDK" document for more information on the installation and usage of SmarAct Python SDKs.

Python programming examples and the SmarAct Python package for manual installation are installed to the C:/SmarAct/MCS2/SDK/Python folder.

### 1.4.3 LabVIEW™ Software Development Kit (SDK)

If the Software Development Kit (SDK) for LabVIEW™ was selected in the installer the MCS2 commands are installed in the program folder of the selected LabVIEW™ version and can be dragged directly from a LabVIEW™ functions palette into your program. The installed VIs can be used with LabVIEW™ 32-bit and 64-bit development systems.

The "SmarActCTL" command palette can be found in the "User Libraries" menu. The base API functions are located in the top level of the menu while the properties are categorized in the corresponding sub folders of the "Properties" folder.

Note that the VIs of the "Get" and "Set" properties are colored slightly different to easily distinguish them from each other.

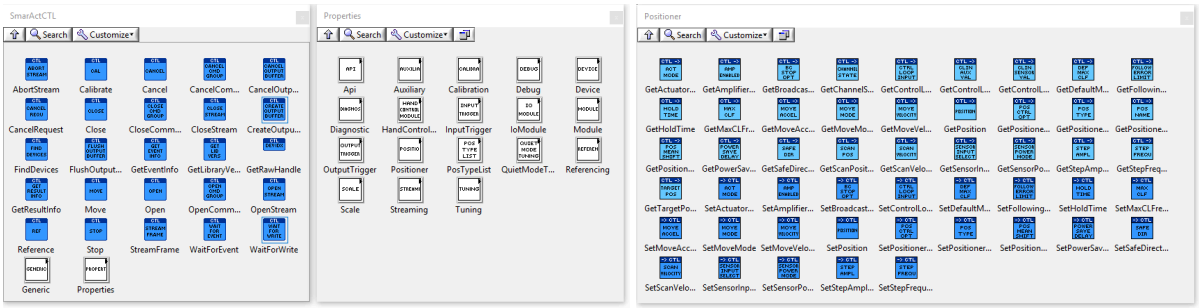


Figure 1.3: LabVIEW™ Functions Palettes

LabVIEW™ programming examples are installed to the C:/SmarAct/MCS2/SDK/LabVIEW folder.

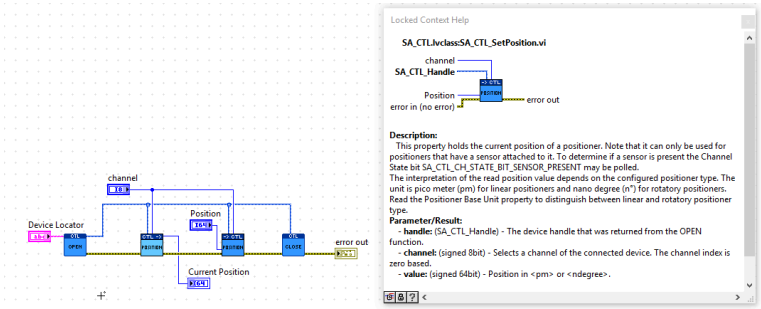


Figure 1.4: LabVIEW™ VIs and Context Help

The "Context Help" (the yellow question mark at the top right corner of the IDE) shows a property description for each VI and may be used as a quick reference of the function details and valid ranges of property parameters etc.

**NOTICE**
While many details are given in the programming examples and the context help of the VIs the "MCS2 Programmers Guide" should be the first reference for the programming of the MCS2.

# Sales partner / Contacts

## Germany

**SmarAct GmbH**

Schuette-Lanz-Strasse 9
26135 Oldenburg
Germany

T: +49 441 - 800 879 0
Email: info-de@smaract.com
www.smaract.com

## France

**SmarAct GmbH**

Schuette-Lanz-Strasse 9
26135 Oldenburg
Germany

T: +49 441 - 800 879 956
Email: info-fr@smaract.com
www.smaract.com

## USA

**SmarAct Inc.**

2140 Shattuck Ave. Suite 302
Berkeley, CA 94704
United States of America

T: +1 415 - 766 9006
Email: info-us@smaract.com
www.smaract.com

## China

**Dynasense Photonics**

6 Taiping Street
Xi Cheng District,
Beijing, China

T: +86 10 - 835 038 53
Email: info@dyna-sense.com
www.dyna-sense.com

**Natsu Precision Tech**

Room 515, Floor 5, Building 7,
No.18 East Qinghe Anning
Zhuang Road,
Haidian District
Beijing, China

T: +86 18 - 616 715 058
Email: chenye@nano-stage.com
www.nano-stage.com

**Shanghai Kingway Optech Co.Ltd**

Room 1212, T1 Building
Zhonggeng Global Creative Center
Lane 166, Yuhong Road
Minhang District
Shanghai, China

Tel: +86 21 - 548 469 66
Email: sales@kingway-optech.com
www.kingway-optech.com

## Japan

**Physix Technology Inc.**

Ichikawa-Business-Plaza
4-2-5 Minami-yawata,
Ichikawa-shi
272-0023 Chiba
Japan

T/F: +81 47 - 370 86 00
Email: info-jp@smaract.com
www.physix-tech.com

## South Korea

**SEUM Tronics**

# 801, 1, Gasan digital 1-ro
Geumcheon-gu
Seoul, 08594,
Korea

T: +82 2 - 868 10 02
Email: info-kr@smaract.com
www.seumtronics.com

## Israel

**Trico Israel Ltd.**

P.O.Box 6172
46150 Herzeliya
Israel

T: +972 9 - 950 60 74
Email: info-il@smaract.com
www.trico.co.il