

SoMachine HVAC Software

Quick Start

09/2015



The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein.

If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

© 2015 Schneider Electric. All rights reserved.

TABLE OF CONTENTS



CHAPTER 1. INTRODUCTION	10
1.1. GENERAL DESCRIPTION	10
1.2. COMPONENTS	10
1.3. SYSTEM RESOURCES	11
CHAPTER 2. MINIMUM SYSTEM REQUIREMENTS	12
2.1. OPERATING SYSTEMS.....	12
2.2. MINIMUM HARDWARE.....	12
2.2.1. M171 OPTIMIZED HARDWARE AND ACCESSORIES	12
2.2.2. M171 PERFORMANCE HARDWARE AND ACCESSORIES	12
2.2.3. M172 PERFORMANCE HARDWARE AND ACCESSORIES	12
CHAPTER 3. INSTALLATION & SOFTWARE REGISTRATION	13
3.1. INSTALLING SoMachine HVAC	13
3.2. REGISTRATION SoMachine HVAC	14
3.2.1. APPLICATION ICONS	15
CHAPTER 4. TARGET	16
4.1. M171 OPTIMIZED	17
4.2. M171 PERFORMANCE	17
4.3. M172 PERFORMANCE	17
4.4. INSTALLING PROGRAMMING CABLE	18
4.4.1. PROGRAMMING CABLE - PC CONNECTION	18
4.4.2. PROGRAMMING CABLE SETUP WIN 7	18
CHAPTER 5. DEFINING HARDWARE REQUIREMENTS	21
5.1. SMALL APPLICATIONS	21
5.2. M171 OPTIMIZED SELECTION CRITERIA.....	21
5.3. SELECTING AN INTERFACE	22
5.4. SELECTING OPTIONAL M171O MODULES.....	23
5.5. COMPLEX APPLICATIONS	24
5.6. M171 PERFORMANCE SELECTION CRITERIA.....	24
5.7. M172 PERFORMANCE SELECTION CRITERIA.....	25
5.8. ACCESSORIES AND COMMUNICATION MODULES FOR THE M171P AND M172P	26
5.9. M171 PERFORMANCE FLUSH SELECTION CRITERIA.....	27
5.10. CONNECTIVITY	28
CHAPTER 6. APPLICATION PANEL DEFINITIONS	30
6.1. START-UP AND TEST.....	30
6.2. FIRST START-UP	30
6.2.1. ICONS.....	32
6.2.2. PROJECT PANEL.....	32
CHAPTER 7. GETTING STARTED	33
7.1. FIRST START-UP.....	33
7.1.1. CREATING A NEW PROJECT	34

7.2. CREATING A NEW PROGRAM	35
7.3. SELECTING THE IEC LANGUAGE	35
7.4. OPENING AN EXISTING PROGRAM.....	35
7.5. WRITING A PROGRAM.....	36
7.5.1. PROJECT PANEL / PROJECT TAB	36
7.5.2. ASSOCIATING A PROGRAM TO A TASK.....	38
7.5.3. PROJECT PANEL / RESOURCES TAB	38
7.5.4. MODBUS OBJECTS	38
7.5.5. TARGET MENU M171O	40
7.5.6. TARGET MENU M171P/M172P	42
CHAPTER 8. MANAGING PROJECTS	44
8.1. SAVING PROJECTS	44
8.2. TRANSFERRING PROJECTS AND FILES.....	45
8.3. CONNECTION.....	46
8.3.1. PROGRAMMING OF THE M171O	46
8.3.2. PROGRAMMING OF THE M171P	48
8.3.3. PROGRAMMING OF THE M172P	49
8.4. DOWNLOAD CONTROLLER APPLICATION ONTO TARGET	50
8.4.1. PRELIMINARY OPERATIONS	50
8.4.2. M171O TARGET	51
8.4.3. M171P/M172P TARGETS	52
CHAPTER 9. DEVICE PANEL DEFINITIONS	57
CHAPTER 10. FIELD OPERATION.....	59
10.1. FIRST START-UP	59
10.2. OPENING THE CONTROLLER APPLICATION	60
10.3. CONNECTING TO THE TARGET DEVICE	61
10.3.1. READING PARAMETERS	61
CHAPTER 11. USER INTERFACE PANEL DEFINITIONS	64
CHAPTER 12. TO START USER INTERFACE	66
12.1. FIRST START-UP	66
12.1.1. CREATING A NEW PROJECT	67
12.2. CREATING PAGES	68
12.2.1. PAGES.....	68
12.2.2. PROPERTIES PANEL	70
12.2.3. PROPERTIES TAB	70
12.2.4. ACTIONS PANEL	71
12.2.5. APPLICATION PARAMETERS	72
12.2.6. PROPERTIES TAB/VARIABLE	72
12.2.7. PROJECT PANEL / RESOURCES TAB	74
CHAPTER 13. CONNECTION PANEL DEFINITIONS	78
CHAPTER 14. TO START CONNECTION	80
14.1. FIRST START-UP	80
14.1.1. CREATING A NEW PROJECT (WORKSPACE).....	80
14.1.2. PROJECT PANEL	83
CHAPTER 15. SIMULATION PANEL DEFINITIONS.....	86

CHAPTER 16. SIMULATION	88
16.1. FIRST START-UP	88
16.2. OPENING SIMULATION FROM APPLICATION	89
CHAPTER 17. CUSTOMER CARE	91
17.1. CUSTOMER CARE CENTER	91
17.2. ONLINE HELP	91
17.2.1. ONLINE HELP - LED.....	92
17.3. LICENSE AGREEMENT (EULA).....	92
CHAPTER 18. TROUBLESHOOTING.....	93
18.1. APPLICATION	93
18.2. DEVICE	93
18.3. SIMULATION	93
CHAPTER 19. SHORTCUTS.....	94
19.1. APPLICATION SHORTCUTS	94
19.2. DEVICE SHORTCUTS	96
19.3. USER INTERFACE SHORTCUTS	97
19.4. CONNECTION SHORTCUTS	97
CHAPTER 20. APPENDIX PROJECT DEVELOPMENT EXAMPLES.....	98
20.1. PROJECT 1 - COUNTER.....	98
20.1.1. DEVICE.....	102
20.2. PROJECT 2 - COUNTER2 FUNCTION BLOCK	103
CHAPTER 21. APPENDIX EXPORTING CSV FILES	107
21.1. EXPORTING CSV FILES	107
21.1.1. DEVICE.....	108

SAFETY INFORMATION



Important Information

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to inform of potential hazards or to call attention to information that clarifies or simplifies a procedure.



The addition of this symbol to a Danger safety label indicates that an electrical hazard exists, which will result in personal injury if the instructions are not followed.



This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

DANGER

DANGER indicates a hazardous situation which, if not avoided, **will result** in death or serious injury.

WARNING

WARNING indicates a hazardous situation which, if not avoided, **could result** in death or serious injury.

CAUTION

CAUTION indicates a hazardous situation which, if not avoided, **could result** in minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material. A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

BEFORE YOU BEGIN

Do not use this product on machinery lacking effective point-of-operation guarding. Lack of effective point-of-operation guarding on a machine can result in serious injury to the operator of that machine.

WARNING

UNGUARDED EQUIPMENT

- Do not use this software and related automation equipment on equipment which does not have point-of-operation protection.
- Do not reach into machinery during operation.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

This automation equipment and related software is used to control a variety of industrial processes. The type or model of automation equipment suitable for each application will vary depending on factors such as the control function required, degree of protection required, production methods, unusual conditions, government regulations, etc. In some applications, more than one processor may be required, as when backup redundancy is needed.

Only you, the user, machine builder or system integrator can be aware of all the conditions and factors present during setup, operation, and maintenance of the machine and, therefore, can determine the automation equipment and the related safeties and interlocks which can be properly used. When selecting automation and control equipment and related software for a particular application, you should refer to the applicable local and national standards and regulations. The National Safety Council's Accident Prevention Manual (nationally recognized in the United States of America) also provides much useful information.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the operator's hands and other parts of the body are free to enter the pinch points or other hazardous areas and serious injury can occur. Software products alone cannot protect an operator from injury. For this reason the software cannot be substituted for or take the place of point-of-operation protection.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of the Function Block Library, System User Guide, or other implementation referenced in this documentation.

START-UP AND TEST

Before using electrical control and automation equipment for regular operation after installation, the system should be given a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such a check be made and that enough time is allowed to perform complete and satisfactory testing.

⚠ CAUTION
EQUIPMENT OPERATION HAZARD <ul style="list-style-type: none">• Verify that all installation and set up procedures have been completed.• Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices.• Remove tools, meters, and debris from equipment. Failure to follow these instructions can result in injury or equipment damage.

Follow all start-up tests recommended in the equipment documentation. Store all equipment documentation for future references.

Software testing must be done in both simulated and real environments.

Verify that the completed system is free from all short circuits and temporary grounds that are not installed according to local regulations (according to the National Electrical Code in the U.S.A, for instance). If high-potential voltage testing is necessary, follow recommendations in equipment documentation to prevent accidental equipment damage.

Before energizing equipment:

- Remove tools, meters, and debris from equipment.
- Close the equipment enclosure door.
- Remove all temporary grounds from incoming power lines.
- Perform all start-up tests recommended by the manufacturer.

OPERATION AND ADJUSTMENTS

The following precautions are from the NEMA Standards Publication ICS 7.1-1995 (English version prevails):

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly operated.
- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or unsafe operation. Always use the manufacturer's instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer's instructions and the machinery used with the electrical equipment.
- Only those operational adjustments actually required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.

Product Related Information

WARNING

LOSS OF CONTROL

- The designer of any control scheme must consider the potential failure modes of control paths and, for certain critical control functions, provide a means to achieve a safe state during and after a path failure. Examples of critical control functions are emergency stop and overtravel stop, power outage and restart.
- Separate or redundant control paths must be provided for critical control functions.
- System control paths may include communication links. Consideration must be given to the implications of unanticipated transmission delays or failures of the link.
- Observe all accident prevention regulations and local safety guidelines.⁽¹⁾
- Each implementation of this equipment must be individually and thoroughly tested for proper operation before being placed into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

(1) For additional information, refer to NEMA ICS 1.1 (latest edition), "Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control" and to NEMA ICS 7.1 (latest edition), "Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems" or their equivalent governing your particular location.

WARNING

UNINTENDED EQUIPMENT OPERATION

- Only use software approved by Schneider Electric for use with this equipment.
- Update your application program every time you change the physical hardware configuration.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

ABOUT THE BOOK



Document Scope

This document is aimed at designers and developers and requires a knowledge of one or more IEC61131-3 standard programming languages and is designed to provide a first-level overview of the installation, functions and use of **SoMachine HVAC**.

Validity Note

This document is valid for **SoMachine HVAC (v.2.0 or greater)**.

The technical characteristics of the devices described in this manual also appear online.

The characteristics that are presented in this manual should be the same as those characteristics that appear online. In line with our policy of constant improvement, we may revise content over time to improve clarity and accuracy. If you see a difference between the manual and online information, use the online information as your reference.

Related Documents

Title of Documentation	Reference Document Code
Modicon M172 Performance Logic Controllers, Hardware User Manual	EIO0000002015 (ENG) EIO0000002017 (GER)
Modicon M171 Optimized Logic Controllers, Hardware User Manual	EIO0000002032 (ENG) EIO0000002033 (GER)
Modicon M171 Performance Logic Controllers, Hardware User Manual	EIO0000002030 (ENG) EIO0000002031 (GER)
Modicon M171 Electronic Expansion Valve Drivers, User Manual	EIO0000002034 (ENG) EIO0000002037 (GER)
SoMachine HVAC software, Programming Guide	EIO0000002036 (ENG)
SoMachine HVAC HVAC&R Function Library, User Guide	EIO0000002057 (ENG)

You can download these technical publications and other technical information from our website at:

www.schneider-electric.com

1. INTRODUCTION

1.1. GENERAL DESCRIPTION

SoMachine HVAC is a quick, effective development tool for creating and customizing new programs for different types of controller applications, which is particularly suited to HVAC/R requirements.

The use of several different programming languages in accordance with **IEC61131-3** regulations (programming standard for industrial control), makes it possible to develop new algorithms or entire programs independently, which can then be downloaded to:

- **M171 Optimized** via PC or Programming stick TM171AMFK
- **M171/M172 Performance** via PC or USB memory key.

**IEC 61131-3
development
environment**

1.2. COMPONENTS

Components, interfaces, converters and accessories are described below. **The interfaces and converters are dedicated to M171O, M171P Flush or M171P/M172P.**

SoMachine HVAC software

The **SoMachine HVAC** software suite and its functions will be illustrated in this document.

The **SoMachine HVAC** software suite consists of 5 development environments for programming the **M171 Optimized, M171 Performance Flush & M171/M172 Performance** programmable controllers:

- **SoMachine HVAC Application**, dedicated to developers, for creating and managing libraries, Controller applications and diagnostics.
- **SoMachine HVAC Device**, dedicated to users, for managing previously developed controller applications, downloading them to the **target** (programmable controller) device and modifying device parameters from a serial port.
- **SoMachine HVAC Simulation**, dedicated to software developers, for executing controller applications and **HMI** pages (**M171P/M172P** only) in a simulation environment without the need to have a **target**. A controller application can be downloaded directly onto the Simulator without the need to have a physical M171O or M171P/M172P

The following 2 development environments are only available for **M171/M172 Performance**:

- **SoMachine HVAC Connection**, dedicated to software developers, for creating networks. **Connection is the entry point of the software suite - see Target chapter**
- **SoMachine HVAC User Interface**, dedicated to software developers, for personalizing the graphical interface on built-in displays and other displays.

M171 Optimized Programming cable

Component required for **M171 Optimized** only:

Component required for connection to PC

The USB/TTL-I2C hardware interface to be used in association with the software suite enables you to:

- use the software itself.
- connect to the **target** device in order to control it.
- connect to the Programming Stick component.

Programming Stick

Component required for **M171 Optimized** only:

This is a memory support, which allows you to:

1. update the firmware of the **target** device.
2. update the controller application of the **target** device.
3. update the parameter values of the **target** device.
4. upload the parameter values from the **target** device.

Connection cables

Components required for **M171 Optimized** only:

- “Yellow” cable with JST – molex terminals, see Connection Modes chapter for use instructions
- “Blue” cable with JST – JST terminals, see Connection Modes chapter for use instructions
- USB-A/A extension lead, 2 m

Converters

NOTE: M171/M172 Performance only: USB/485 adapter TSXCUSB485, cable VW3A83O6D3O

NOTE: alternatively if there is a **RS232** serial port, **M171P/M172P** can be connected to the PC using an RS485/RS232 adapter

Communication Modules (M171P/M172P)

A wide range of Communication Modules, 2DIN rail-mounted modules, allows integration with industrial systems, BMS and Ethernet networks.

NOTE: not available for **M171 Performance Flush**, which has 3 on-board RS485, CAN and ETHERNET serials.

NOTE: M172 Performance has 4 on-board 2xRS485, CAN and ETHERNET serials.

Abbreviations and definitions

- **Application, Device, Connection, Simulation, UserInterface:** abbreviations of **SoMachine HVAC Application, SoMachine HVAC Device, SoMachine HVAC Connection, SoMachine HVAC Simulation and SoMachine HVAC UserInterface** respectively. Software suite
- **BIOS menu, BIOS:** factory-set BIOS parameters menu. Not modifiable.
The BIOS cannot be modified but can be updated
- **Controller application:** an application made to **IEC61131-3** (industrial control programming standards) by means of the **Application** development environment to be downloaded to the **target** using **Application** or **Device**
- **M171O:** abbreviation of **M171 Optimized**; **M171P:** abbreviation of **M171 Performance**; **M171P Flush:** abbreviation of **M171 Performance Flush**; **M172P:** abbreviation of **M172 Performance**
- **HMI:** acronym of **Human Machine Interface**. Graphical interface developed with **UserInterface**
- **Instance:** object of a predefined object class (function block, **template**, etc)
- **Programming Stick:** abbreviation of Multi Function Key
- **Plugin : Communication Modules** i.e. 2DIN rail-mounted modules allows integration with industrial systems, BMS and Ethernet networks.
- **Quick Start:** this document
- **Panel** or **section** and **Tab**. The workspace is divided into sections or **Panels**. Each Panel may in turn be subdivided into tabs (for example Resources tab)
- **Target device, Target:** name given to the **M171 Optimized, M171/M172 Performance** or **M171 Performance M171P Flush** programmable controller
- **Template:** in this document it denotes a page template created with **UserInterface**

NOTE: Many definitions and abbreviations are standard information technology and are not listed here.

For example a Function is a standard term. Other terms, such as **Function Block**, will be described in the relevant paragraphs.

1.3. SYSTEM RESOURCES

The developer has the following resources available on the **target** devices for controller application:

	M171O	M171P/M171P Flush	M172P
CPU	14.7 MHz	72 MHz, 32MB RAM	120 MHz, 32MB RAM
Available memory for Application	188 kB	1 MB	1 MB
Available memory for User Interface	-	1.5 MB	1.5 MB
FLASH memory data	-	126 MB	5 MB
RAM Memory	2300 B	512 kB	512 kB
automatic mapping	(1150 word)	(256000 word)	(256000 word)
RAM Memory	1024 B	10 kB	10 kB
Modbus mapping	(512 word)	(5000 word)	(5000 word)
EEPROM variables	1024 B (512 word)	8 kB (4000 word) general purpose + 20 kB (10000 word) BACNet Objects	8 kB (4000 word) general purpose + 20 kB (10000 word) BACNet Objects

2. MINIMUM SYSTEM REQUIREMENTS

2.1. OPERATING SYSTEMS

- Windows 7 Home Premium English
- Windows 7 Professional English
- Windows 7 Ultimate English
- Windows 8 / 8.1.

For all OS: other languages not explicitly tested.

OTHER SOFTWARE

(M171 Optimized only)

Driver AT90USBxxx CDC USB to UART MGM for use of the M171 Optimized Programming cable - available in the installation package

See M171 Optimized Programming cable Installation

Adobe Acrobat Reader is required to view PDF files. This is a free program available from the Adobe web site. Follow the download directions on the Adobe web site to get your copy of Adobe Acrobat Reader:



2.2. MINIMUM HARDWARE

- Processor: Pentium 1.6 GHz or greater
- RAM: 1 GB; 2 GB preferred
- Hard disk: 500 MB
- Drive: DVD drive
- Display: SVGA video card; 800×600, 128 MB; 1024×768, 256 MB preferred
- Peripheral device: A mouse or compatible pointing device
- USB interface:
 - For M171P it is also possible to use a RS232 port by means of a RS232/RS485 converter.
 - For M171O it is possible to use the RS232 port by means of RS232/RS485 converter with maximum speed 19200 baud. TTL cannot be used in this case.

NOTE: A typical installation (available in English, default libraries) requires about 500 Mbyte of disk space. Allow a further 1GByte of disk space for installing libraries and dedicated applications.

To download the application, you need the following components:

2.2.1. M171 OPTIMIZED HARDWARE AND ACCESSORIES

- 1 **M171 Optimized**
- 1 **M171 Optimized Programming cable 100-3 Manufacturer** + yellow TTL cable
- 1 optional **Programming Stick** + TTL blue cable
- **M171 Optimized** power cables. M171 Optimized does not need to be powered separately if it is connected to a PC by means of M171 Optimized Programmable cable. Use yellow TTL cable

2.2.2. M171 PERFORMANCE HARDWARE AND ACCESSORIES

- 1 **M171 Performance**
- 1 **USB/RS485** or **plug-in Ethernet** converter

2.2.3. M172 PERFORMANCE HARDWARE AND ACCESSORIES

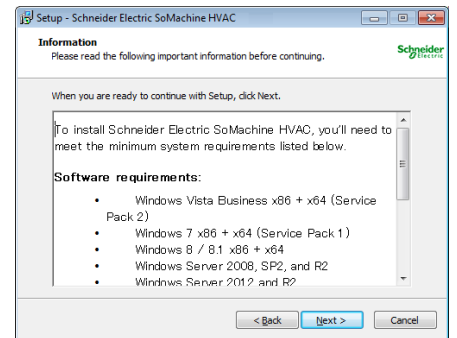
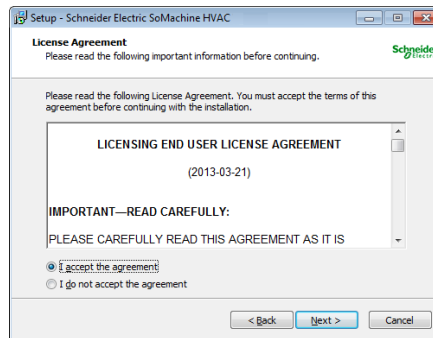
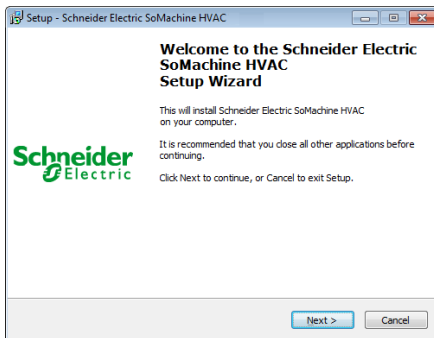
- 1 **M172 Performance**
- 1 **USB/RS485** converter or **Ethernet** port

3. INSTALLATION & SOFTWARE REGISTRATION

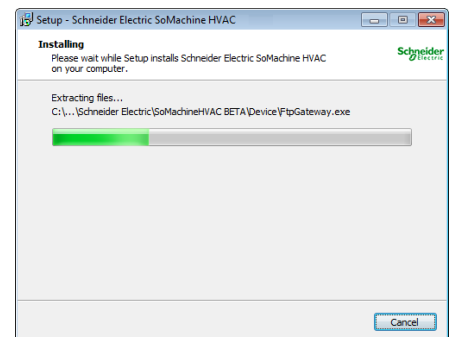
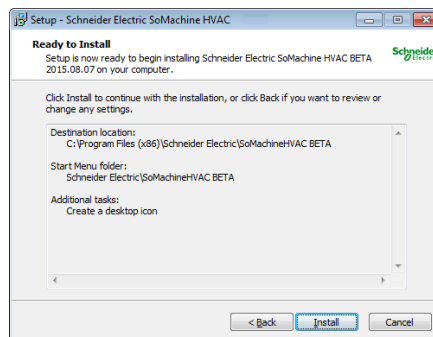
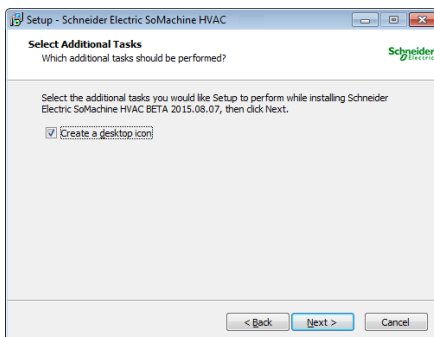
3.1. INSTALLING SoMachine HVAC

Run **Schneider_SoMachine HVAC_xxx.exe** supplied by Schneider Electric and then follow the installation wizard as indicated in the following screenshots. Check the software version in advance with Customer Care Center.

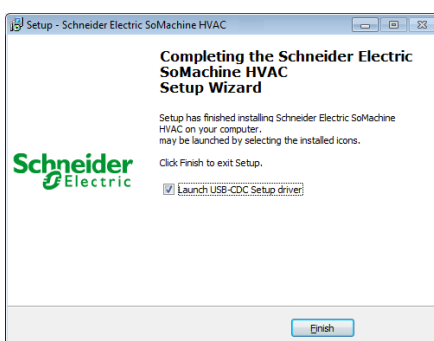
1. Click **"Next"**
2. Click **"I accept the agreement"**
To continue with installation, you must accept the License Agreement. Carefully read the End User License Agreement (EULA) before continuing. The EULA is also available on the OnLineHelp (Menu Help > About > License Agreement)
3. Check Minimum System Requirements.



4. Click **"Next"**. When installation is complete, the **SoMachine HVAC** icon will appear on the desktop.
5. Click **"Install"**.
6. Installation in progress.



7. Click **"Finish"**.
With the option 'launch USB-CDC Setup driver' the USB driver will be installed.



3.2. REGISTRATION SoMachine HVAC

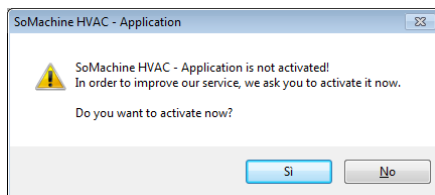
In order to activate your SoMachine software you need to create and validate your Software License Portal (SLP) account.

SLP is a cloud based service of Schneider Electric that provides access to your Schneider Electric Software licenses on the web. The SLP allows to overview and manage licenses and registrations.

Follow next steps to create and validate your account.

- Go to www.schneider-electric.com/licensing.
- Choose “Register license into my Account”. Enter your email that will be your user ID for your SLP account
- Follow web instructions, fill in mandatory fields and create the account.
- Check your mailbox: an email with subject 'Schneider Electric Web User Account - Account Creation' from src-eeurope@schneider-electric.com will arrive. Click on the link to validate the account.
- Once you have activated your account you can go on it. Your email is your account ID.

Once you have activated your account open SoMachine Application



1



2









3

1. You will be requested to activate the software. Click 'Yes'. (1)
2. Enter the e-mail address of your Web Account. Ensure you are connected to a network. Press 'Activate'.
3. The Activation Status will confirm your license activation

(1) You may click 'No' and register afterwards. At any time, **access the Registration menu (Menu Help > Registration) to activate the license.**

3.2.1. APPLICATION ICONS

Click the **SoMachine HVAC** icon on the desktop to access a folder containing the following links.

M1710 • M171P FLUSH • M171P • M172P		M171P FLUSH • M171P • M172P	
icon	description	icon	description
	Application development tool for M1710 & M171P/M172P		Connection development tool for M171P Flush & M171P/M172P
	Device development tool for M1710 & M171P/M172P		User Interface development tool for M171P Flush & M171P/M172P
	Simulation development tool for M1710 & M171P/M172P		SoMachine HVAC 2.0 or greater

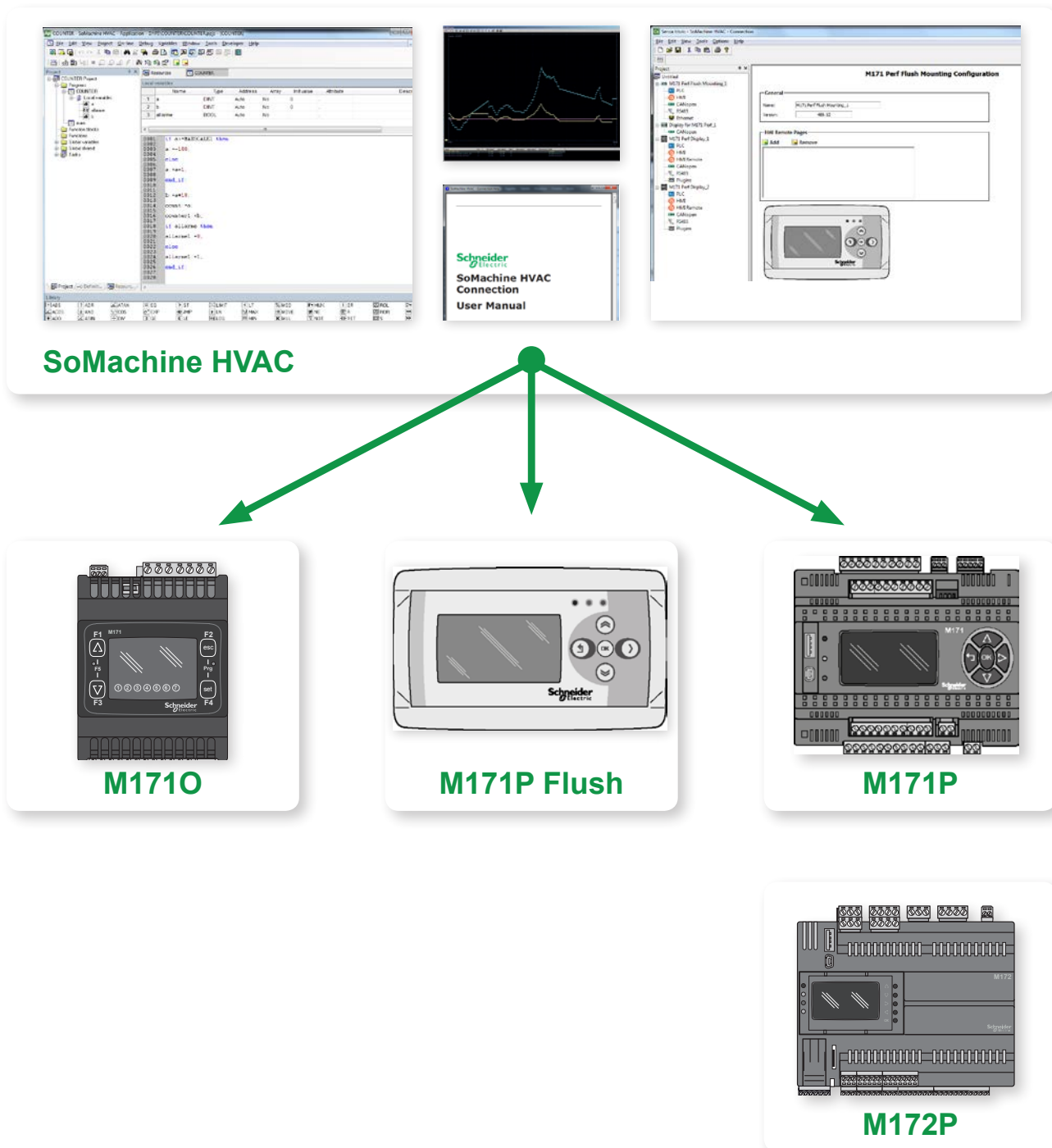
4. TARGET

The **SoMachine HVAC** software suite is designed for use with **M171O**, **M171P Flush** and **M171P/M172P** target devices.

The selection of the appropriate target for the needs of the developer is indicated in the chapter **Defining Hardware Requirements**.

The **SoMachine HVAC** software suite is unique for **M171O** and **M171P/M172P**.

The entire system is the **SoMachine HVAC** software + **M171 Optimized**, **M171P/M172P** and **M171P Flush** hardware.



4.1. M171 OPTIMIZED

The **M171O** developer will only be able to use the **Application** and **Device** workspaces

The **M171O** developer will be able to launch **Application** to start programming.

The **M171O** user can use **Device** to download the Controller application previously compiled by the developer. From **Application** it will always be possible to open **Device** directly without having to launch the program using the **SoMachine HVAC** icon.

4.2. M171 PERFORMANCE

The **M171P & M171P Flush** developer will be able to use the different workspaces.

The **M171P Flush** developer will be able to launch **Connection** to get started. **Connection** is the entry point for the development activities, from controller programming to creation of the HMI menu and downloading onto the **target** using **Device**.

The target for M171P is M171P Display or M171P Blind.

The target for M171P Flush is M171P Flush Mounting. The M171P menus must be downloaded onto M171P and then “uploaded” from M171P Flush as remote menus.

From **Connection** it will always be possible to directly open **User Interface**, **Application** and **Device** without having to launch the relative program using the **SoMachine HVAC** icon.

4.3. M172 PERFORMANCE

The **M172P** developer will be able to use the different workspaces.

The **M172P** developer will be able to launch **Connection** to get started. **Connection** is the entry point for the development activities, from controller programming to creation of the HMI menu and downloading onto the **target** using **Device**.

From **Connection** it will always be possible to directly open **User Interface**, **Application** and **Device** without having to launch the relative program using the **SoMachine HVAC** icon.

4.4. INSTALLING PROGRAMMING CABLE

This operation must be carried out after installing SoMachine HVAC. Installation required exclusively for M1710.

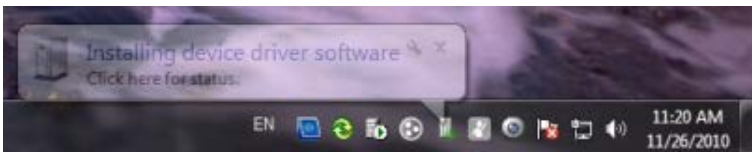
4.4.1. PROGRAMMING CABLE - PC CONNECTION

Connect the M171 Optimized Programming cable to one of the USB ports on the PC.
Use the supplied USB extension lead for the connection.

4.4.2. PROGRAMMING CABLE SETUP WIN 7

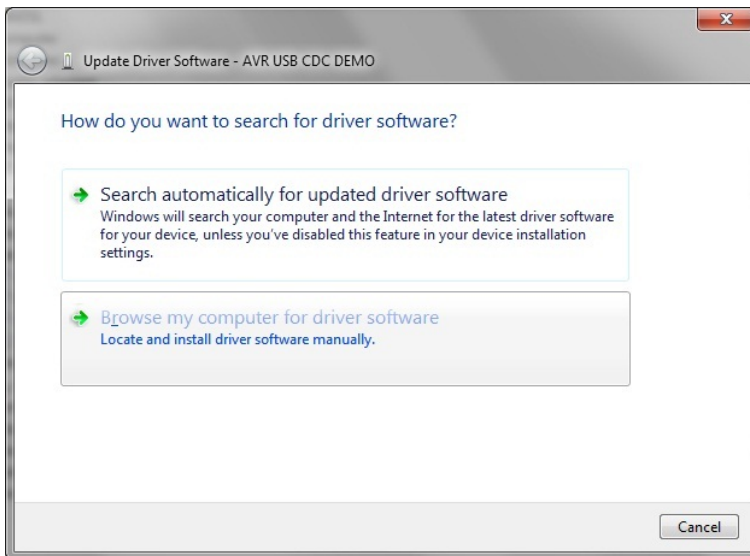
As soon as the M171 Optimized Programming cable is connected, the Windows 7 operating system recognizes the newly connected hardware. The steps to be followed are described below.

1. Once the hardware is connected, the message shown in the figure will appear:



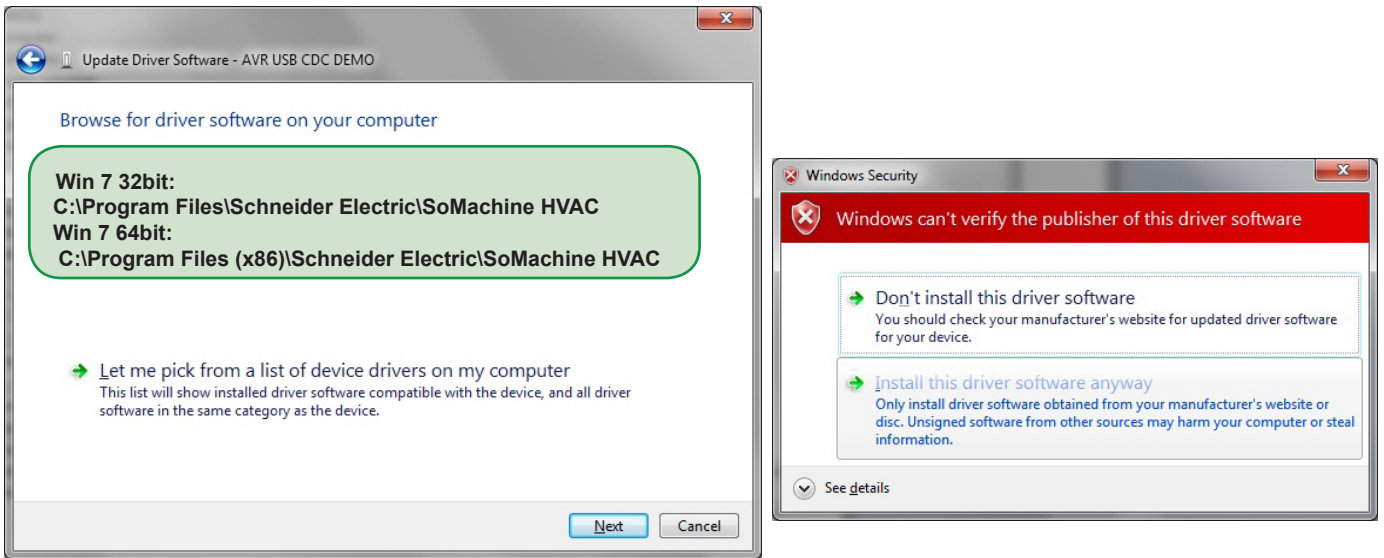
Click on the message to start the guided installation procedure

2. The screen shown below will appear: select the second option to identify the driver

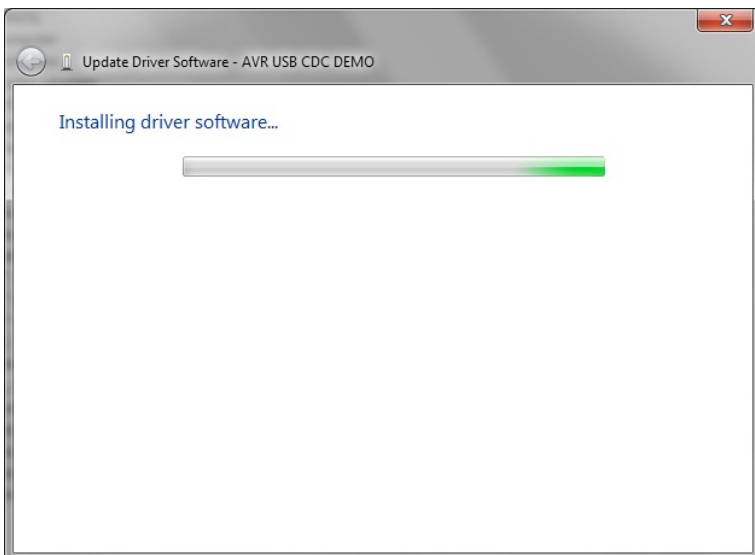


3. In the next screen, select the installation path for the SoMachine HVAC software. Unless changed during installation, the path will be as shown in the figure.

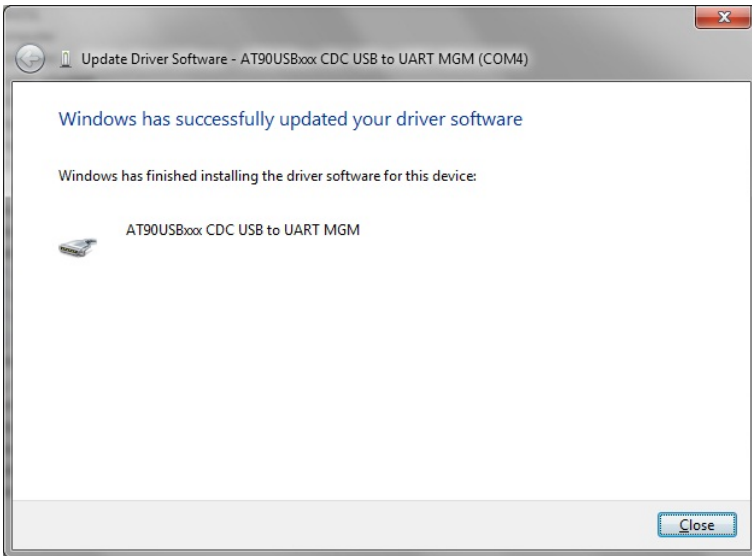
4. Once you have selected the correct path, the screen shown below will appear: select **Install this driver software anyway**



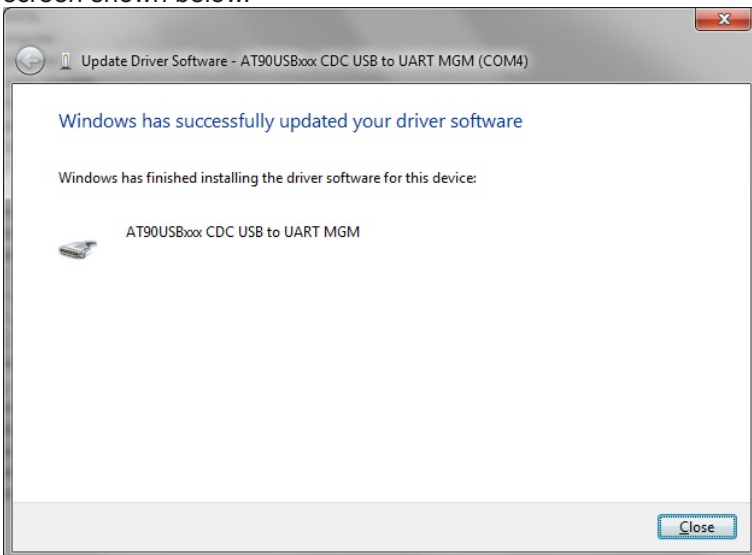
5. The screen shown below will appear, indicating that the action has been performed.



6. On completion of the process, the screen shown below will appear.



To check correct installation of the driver and the port to which the hardware has been allocated, check the Windows screen shown below.



5. DEFINING HARDWARE REQUIREMENTS

5.1. SMALL APPLICATIONS

M171 Optimized was designed for small plants and in particular for:

Manufacturers of

- A.H.U.s (Air Handling Units)
- Chillers & Heat Pumps
- Rooftops
- Compressor Rooms

Installers/integrators of

- Air systems
- Hydronic systems
- Combi systems (air/water)

5.2. M171 OPTIMIZED SELECTION CRITERIA

The **M171 Optimized** references are available in both a DIN rail-mounted version, which significantly reduces wiring times, or in the 32x74 format for Flush mounting.



M171 Optimized 22 I/Os Flush mounting has 2 references, giving you 6 digital inputs, up to 5 relay outputs, 2 PWM analog outputs, 3 configurable 0...10 V / 0...20 mA / 4...20 mA analog outputs and an Open Collector digital output for an external relay.
The 32x74mm format helps ensure versatility and ease of installation.



M171 Optimized 22 I/Os comes in a variety of references, giving you 6 digital inputs, up to 5 relay outputs, up to 2 TRIAC outputs, up to 2 PWM analog outputs, up to 3 configurable 0...10V/0...20mA/4...20mA analog outputs and up to 2 Open Collector digital outputs for an external relay.



The 4DIN format provides extra flexibility and easy installation.

It runs on 12-24 Vac or 12-24 Vac / 24 Vdc power supplies.



M171 Optimized 14 I/Os have removable connectors both for the low voltage inputs and the relay outputs.

M171 Optimized 14 I/Os has different references, giving you 4 relay outputs, up to 2 PWM Open Collector analog outputs / 2 digital inputs as an alternative, up to 2 x 0...10V analog outputs, one configurable 20mA / 4...20mA output or, as an alternative 1 x 0...10 V output on dedicated references.



The 4DIN format provides extra flexibility and easy installation.



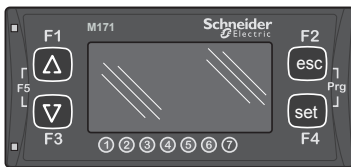
100-240 Vac power supply.

For further details, refer to Modicon M171 Optimized Logic Controllers, Hardware User Manual.

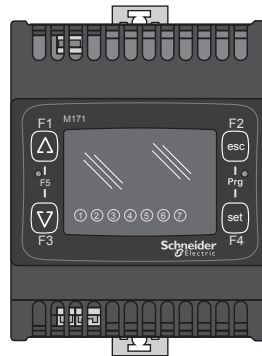
5.3. SELECTING AN INTERFACE

M171 Optimized 22 I/Os, 12-24 Vac/dc / 24 Vdc is available in 3 versions:

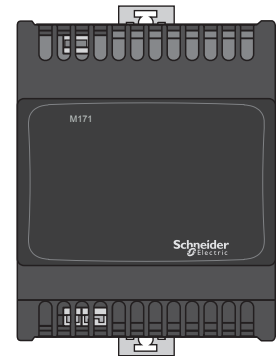
1. M171 Optimized Flush mounting 22 I/Os for Flush mounting
2. M171 Optimized Display 22 I/Os, for DIN rail mounting.
3. M171 Optimized Blind 22 I/Os, for DIN rail mounting.



**32x74 mm panel mounted
with display**



4DIN with display



**4DIN expansion
without display**

The controller application created by **Application** can be loaded on every version. Menu navigation is identical for the different references.

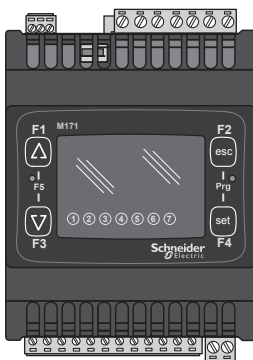
The **32x74** and **4DIN with display** references have a 7-segment, 4-digit LED display.

The **4DIN without display** reference has no display so you need a remote display for configuration in the field.

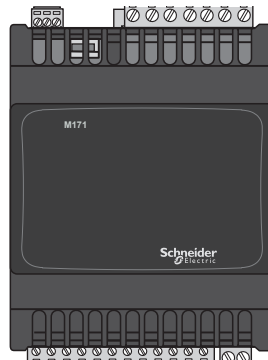
For a description of the keys, user interface and connections, refer to Modicon M171 Optimized Logic Controllers, Hardware User Manual.

M171 Optimized 14 I/O 100-240 Vac is available in 3 versions:

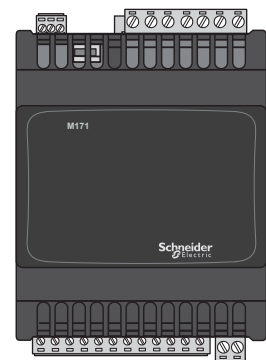
1. M171 Optimized Display 14 I/O 100-240 Vac with DIN rail mounting
2. M171 Optimized Blind 14 I/O 100-240 Vac with DIN rail mounting
3. M171 Optimized expansion 22 I/Os expansion module with DIN rail mounting:



4DIN with display



4DIN without display

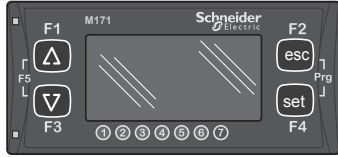


**4DIN expansion
without display**

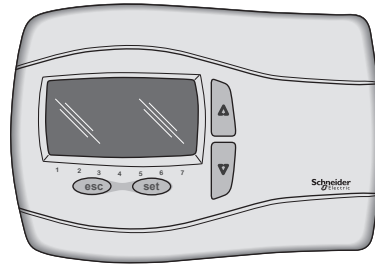
- The controller application created by **Application** can be loaded on every version.
- Menu navigation is identical for the different references.
- The M171 Optimized Display references have a 7-segment, 4-digit LED display.
- The M171 Optimized Blind reference has no display so you need a remote display for configuration in the field. For a description of the keys, user interface and connections, refer to Modicon M171 Optimized Logic Controllers, Hardware User Manual.

5.4. SELECTING OPTIONAL M171O MODULES

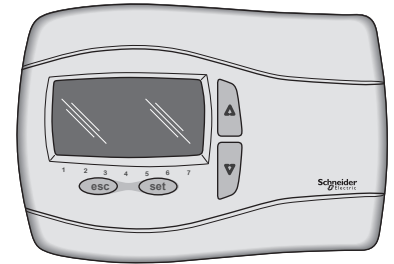
Some of the accessories used by **M171O** are listed below:



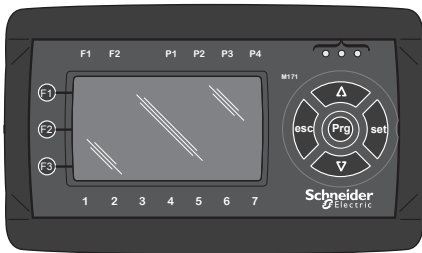
M171 Optimized Display LED



M171 Optimized Wall thermostat with backlight



M171 Optimized Wall thermostat without backlight



M171 Optimized Display LCD without backlight



**expansion modules
(2 references available)
4DIN without display**

- The **M171 Optimized Display LED** remote display for M171P Flush mounting has the same interface as the main **M171O** module.
- The **M171 Optimized Wall thermostat** remote display for wall-mounting has a dedicated interface with LCD display with or without backlighting.
- The **M171 Optimized Display LCD** remote display for M171P Flush mounting has a dedicated interface with LCD display with or without backlighting.
- The expansion modules - available in 2 references - have I/O similar to those of **M171O 22 I/Os**.
For details of connections and further modules and accessories, refer to Modicon M171 Optimized Logic Controllers, Hardware User Manual.

5.5. COMPLEX APPLICATIONS

M171/M172 Performance logic controllers are designed for medium and large plants and in particular for

Manufacturers of:

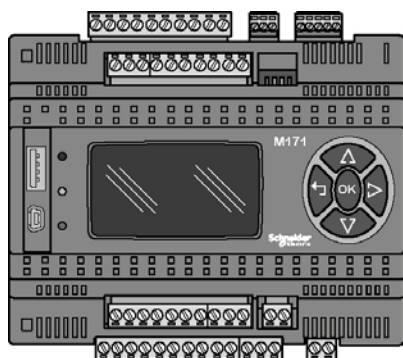
- A.H.U.s (Air Handling Units)
- Chillers & Heat Pumps
- Rooftops
- Compressor Rooms.

Installers/integrators of:

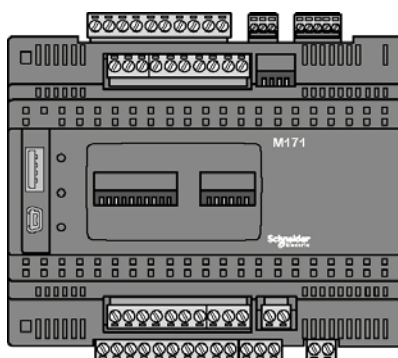
- Air systems
- Hydronic systems
- Combi systems (air/water).

M172 Performance logic controllers increase the scalability and the connectivity achieved by M171 performance controllers. They control the same type of applications as the Modicon M171 performance do, and embed more universality and connectivity.

5.6. M171 PERFORMANCE SELECTION CRITERIA



M171 Perf. Display 27 I/Os



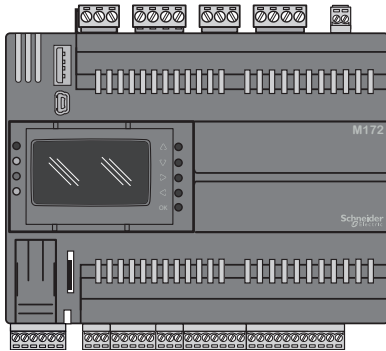
M171 Perf. Blind 27 I/Os

M171 Performance references are available in the 8 DIN rail-mounted version, with display (**M171 Performance Display 27 I/Os** reference) or without display (**M171 Performance Blind 27 I/Os** reference), both supplied with removable screw terminals:

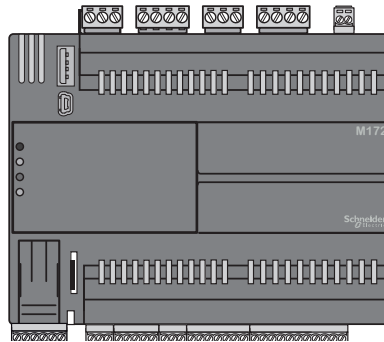
- M171 Performance comes in a variety of references based on which you can mount 9 digital inputs, up to 7 relay outputs, 5 configurable 0...10V/4...20mA analog outputs.
- The different references are also equipped with one USB port and one mini-B USB port.
- The 8DIN format provides extra flexibility and easy installation.
- 24 Vac/dc or 48 Vdc power supply.

For further details, refer to Modicon M171 Performance Logic Controllers, Hardware User Manual.

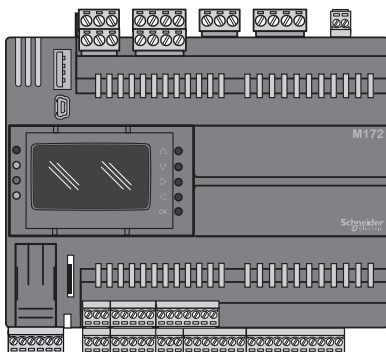
5.7. M172 PERFORMANCE SELECTION CRITERIA



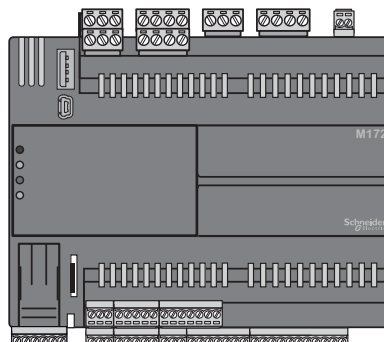
M172 Perf. Display 28 I/Os



M172 Perf. Blind 28 I/Os



M172 Perf. Display 42 I/Os



M172 Perf. Blind 42 I/Os

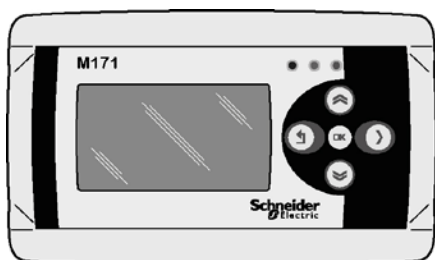
M172 Performance references are available in the 8 DIN rail-mounted version, with display (**M171 Performance Display 28/42 I/Os** references) or without display (**M172 Performance Blind 28/42 I/Os** references), both supplied with removable screw terminals:

- **M172 Performance** comes in a variety of references based on which you can mount up to 12 digital inputs, up to 12 relay outputs, up to 6 configurable 0...10V/4...20mA analog outputs.
- The different references are also equipped with CAN, 2x RS485 Modbus RTU and Ethernet TCP serial port.
- The different references are also equipped with one USB port and one mini-B USB port.
- The 8DIN format provides extra flexibility and easy installation.
- 24 Vac/dc power supply.

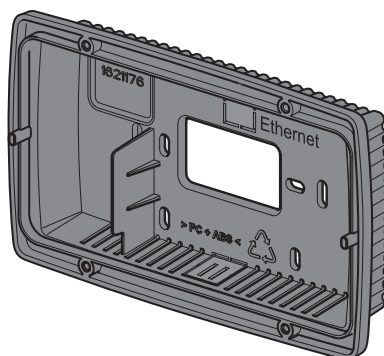
For further details, refer to Modicon M172 Performance Logic Controllers, Hardware User Manual.

5.8. ACCESSORIES AND COMMUNICATION MODULES FOR THE M171P AND M172P

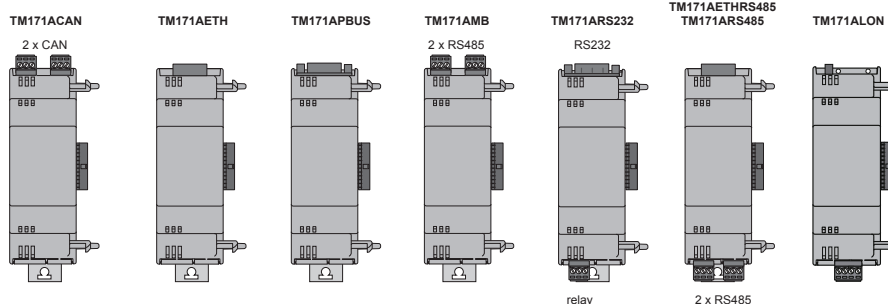
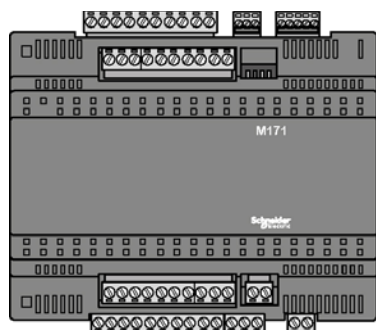
Some of the accessories used by **M171P** are listed below:



M171 Perf. Display Graphic



wall mounting M171 Perf. Display Graphic



M171 Performance expansion

Communication Module

- M171 Performance Display comes with a built-in graphics whilst M171 Performance Blind has no display and must be connected to a remote **M171 Performance Display Graphic** for the configuration of BIOS parameters
- **M171 Performance Display Graphic** is available in the M171P Flush mounting version: a special backplate, available as an accessory, enables it to be wall mounted
- Integration into industrial systems, BMS and Ethernet networks is permitted thanks to the communication modules, available in 2DIN format, that connect quickly and intuitively to the main modules

The **M171 Performance** expansion module, without display, has the same format and the same I/O as the main module.

Communication Modules Compatibility With M171 Performance

The following **TM171A●●●●** Communication Modules can be connected to M171 Performance controllers:

M171 Performance Communication Module	Description	Protocols
TM171ACAN	M171 Performance Communication Module CAN	1 x CAN - Daisy chain
TM171AMB	M171 Performance Communication Module Modbus SL	Modbus Serial Line (SL)
TM171ARS485	M171 Performance Communication Module BACnet MSTP or Modbus	Modbus Serial Line or BACnet MS/TP
TM171ARS232	M171 Performance Communication Module RS232 with relay	RS232 ASCII - 1 Relay 5 A SPDT
TM171ALON	M171 Performance Communication Module LonWorks	LonWorks
TM171AETH TM171AETHRS485	M171 Performance Communication Module Ethernet	ETHERNET+ Modbus Serial Line (SL)
TM171APBUS	M171 Performance Communication Module Profibus	Profibus DP Slave-V0

Communication Modules Compatibility With M172 Performance

The following **TM171A●●●●** Communication Modules can be connected to **M172 Performance** controllers:

M171 Performance Communication Module	Description	Protocols
TM171ACAN	M171 Performance Communication Module CAN	1 x CAN - Daisy chain
TM171AMB	M171 Performance Communication Module Modbus SL	Modbus Serial Line (SL)
TM171ARS485	M171 Performance Communication Module BACnet MSTP or Modbus	Modbus Serial Line or BACnet MS/TP
TM171ARS232	M171 Performance Communication Module RS232 with relay	RS232 ASCII - 1 Relay 5 A SPDT
TM171ALON	M171 Performance Communication Module LonWorks	LonWorks

⚠ WARNING

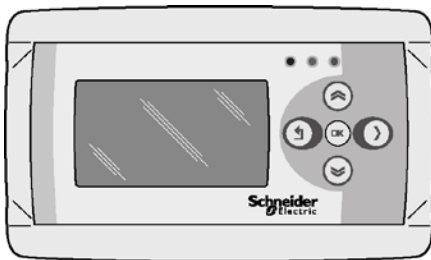
UNINTENDED EQUIPMENT OPERATION

Verify all wiring connections before applying power.

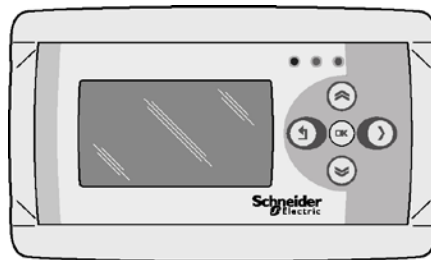
Failure to follow these instructions can result in death, serious injury, or equipment damage.

Use only the listed compatible communication modules in association with the **M172 Performance** logic controller.

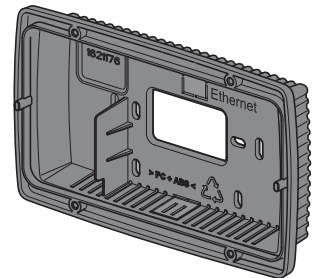
5.9. M171 PERFORMANCE FLUSH SELECTION CRITERIA



M171 Perf. Flush 3 I/Os



M171 Perf. Flush 3 I/Os RH



wall mounting backplate

M171 Performance Flush references are available in the Flush mounting version: a special backplate, available as an accessory, enables them to be wall mounted

There are two **M171 Performance Flush**

- M171 Performance Flush 3 I/Os, MB & BACnet** is equipped with CAN, RS485 Modbus RTU and Ethernet TCP serial, 3 inputs (1 on-board NTC + 1 remote configurable NTC/DI + 1 remote current/voltage configurable input).
- M171 Performance Flush 3 I/Os RH, MB & BACnet** is equipped with CAN, RS485 Modbus RTU and Ethernet TCP serial, 3 inputs (1 on-board humidity module + 1 on-board NTC + 1 remote configurable NTC/DI).

- They come with a built-in 128x64px graphical user display.
- 24 Vac/dc or 48 Vdc power supply.
- For further details, refer to Modicon M171 Performance Logic Controllers, Hardware User Manual.

5.10. CONNECTIVITY

M171 Performance Flush

M171 Performance Flush can be used as a system controller, with gateway functions; used in combination with the M171 Performance and M171 Optimized or third-party controllers it helps provide high performance in terms of memory, connectivity and user interface as well as programming, maintenance and servicing.

M171 Performance Flush references are available with CAN RS-485 and Ethernet.

M171 Performance Flush can be used to:

- manage a local menu created with SoMachine HVAC User Interface.
- manage up to 10 remote menus, created with SoMachine HVAC User Interface and uploaded from the respective M171 Performance devices connected to the network.
- share network variables (binding).

M171P/M172P Performance

- **M171P/M172P** references are available with CAN.
- **M172P** references available with Ethernet.
- **M171P** allows integration into industrial systems, BMS and Ethernet networks thanks to the communication modules, available in 2DIN format, that connect to the main modules.
- Integrated RS485 serials are also available.
- Each controller can be connected to 12 expansion modules via CAN or RS485 serial.
- Each **M171P** can be connected to up to two displays to view the integrated machine display. Using the RS485 network, a **M171O** system acting as a slave can be connected in a network in which **M171P** acts as master.
- The **M171 Display Graphic** is available with CAN. The **M171 Display Graphic** can read menus from **M171P**. Integrated RS485 is also available.

Web functionalities

The **Modicon M172 Performance** also features web functionalities, offering makers of machinery and systems integrators remote access. Having a web-based connection in machines reduces support and maintenance by minimizing call-out charges. End users also benefit, as they can monitor their own systems both locally and from distance, using the graphics interface of any browser.

Main web functionalities:

- Web-based access.
- Remote reading and support.
- Local and remote system control, including alarms management.
- Preventive and predictive maintenance.
- Email alarm alerts.

Care must be taken and provisions made for use of this product as a control device to avoid inadvertent consequences of commanded machine operation, controller state changes, or alteration of data memory or machine operating parameters.

WARNING

UNINTENDED EQUIPMENT OPERATION

- Configure and install the mechanism that enables the remote HMI local to the machine, so that local control over the machine can be maintained regardless of the remote commands sent to the application.
- You must have a complete understanding of the application and the machine before attempting to control the application remotely.
- Take the precautions necessary to assure that you are operating remotely on the intended machine by having clear, identifying documentation within the application and its remote connection.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

6. APPLICATION PANEL DEFINITIONS

6.1. START-UP AND TEST

Before using electrical control and automation equipment for regular operation after installation, the system should be given a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such a verification be made and that enough time is allowed to perform complete and satisfactory testing.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Verify that all installation and set up procedures have been completed.
- Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices.
- Remove tools, meters, and debris from equipment.

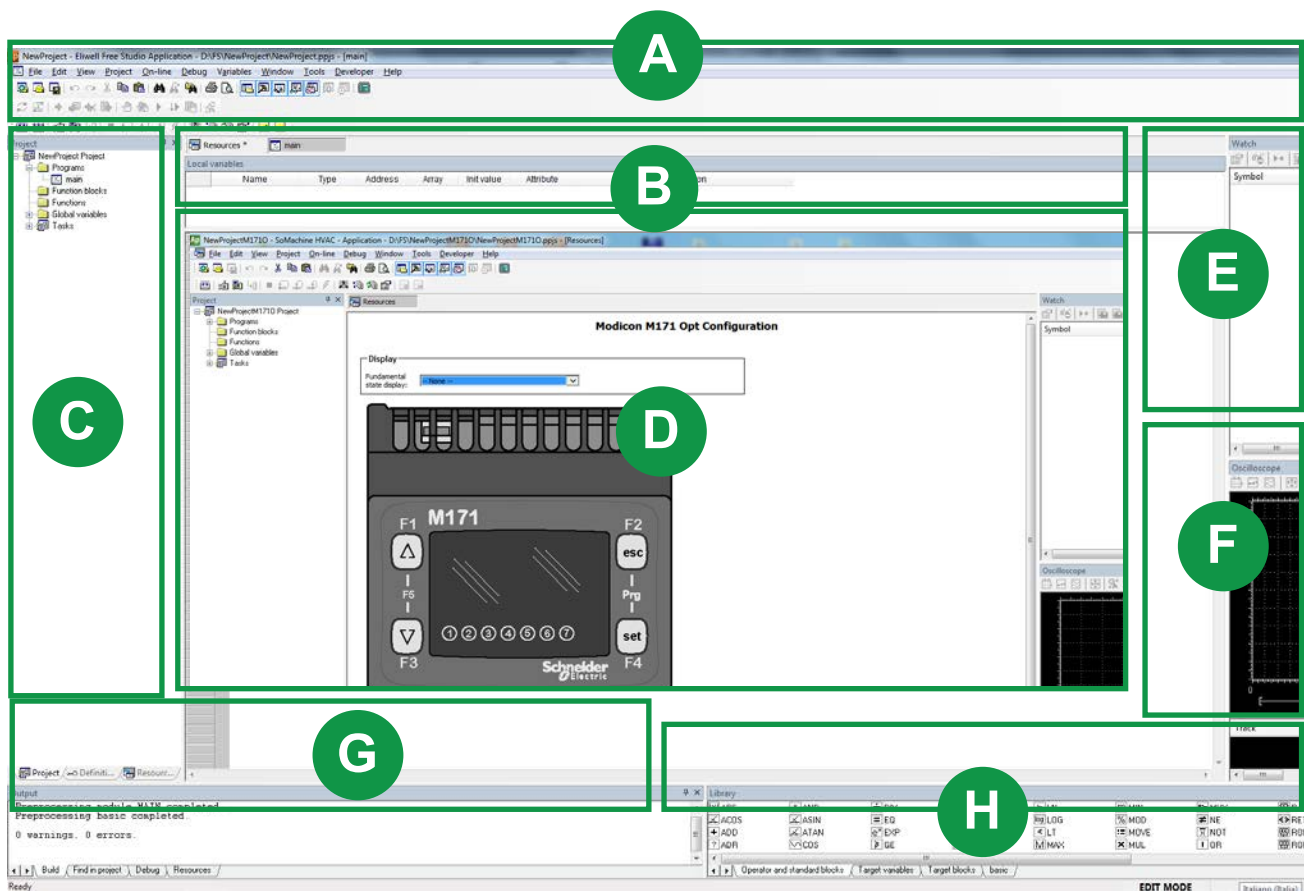
Failure to follow these instructions can result in death, serious injury, or equipment damage.

6.2. FIRST START-UP

The **Application** work environment has various sections for developing the controller application , for example programming in IEC 61131-3 compatible language, testing, debugging and controller application downloading to the target device.

NOTE: Application only downloads the controller application. Parameters (BIOS and PLC) and their default values are downloaded by Device and not by Application.

The sections (Panels) are listed below:



Section	Panel Name / Section Application	Panel	Description
A	/	Tool bar	This section shows the work tools in the form of icons. To display or hide the toolbars, right-click and select or deselect the desired tools. Some tools can be selected by means of key combinations (shortcuts). See relevant chapter for details
B	/	Variables	The global and local variables of the code displayed in section D "Edit" (programs, function blocks and functions) appear in this section
C	Project	Project	The Project Panel is divided into 3 tabs : <ul style="list-style-type: none"> • Project - for managing the application code • Definitions - for managing and defining complex variables defined by the developer • Resources - for managing the target device menu
D	/	Edit	This section enables you to write the program
E	Watch	Display variables with application running	This Panel enables you to manage variables debugging by displaying their status in numerical format when the application is running and connected to the target device
F	Oscilloscope	Display variables graph	Like the Watch Panel, this Panel enables you to manage variables debugging by displaying their status in graph format when the application is running and connected to the target device
G	Library	Functions library	This tool enables you to manage default functions libraries or functions libraries created by the developer. The Panel is divided into various tabs , one for each library. The tabs relating to the following are always available: <ul style="list-style-type: none"> • Operator and standard blocks: operators (AND, OR, etc) • Target variables: specific variables of the target device • Target blocks: specific functions of the target device Additional tabs are managed by means of the drop-down menu Project > Library manager
H	Output	Messages	This Panel shows the messages relating to the development of the project (file opening, compilation errors, status of connection to target device, etc.) NOTE : the connection to the target device is also visible in the bottom right of the screen by means of colored messages (see Application Download)

The Panels appear or remain hidden (see **Shortcuts**) and can be organized as the user wants in terms of size and layout.

SoMachine HVAC saves the last settings made

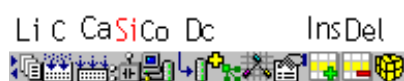
6.2.1. ICONS

The work tools are available via the drop-down menus or **shortcuts**
 Panel A shows a series of icons that help the developer to create the application.
 The toolbars are displayed via the menu **View > Toolbars**
 Below are the most frequently used icons in the **Main** and **Project** toolbars

Main Toolbar



Project Toolbar



The first icons are common icons (Open project, Save project, Copy, Paste etc.), the following ones, with letters above them, have the following functions and shortcuts:

	Toolbar	Function	Function description	Shortcut
F	Main	Full Screen	full screen mode	CTRL+U
W	Main	WORKSPACE	expand workspace window	CTRL+W
O	Main	OUTPUT	show/hide messages Panel	CTRL+R
L	Main	LIBRARY	show/hide library Panel	CTRL+L
Wa	Main	WATCH WINDOW	show/hide watch window	CTRL+T
A	Main	ASYNCH GRAPHIC WINDOW	show/hide asynch graphic window	CTRL+K
C	PROJECT	COMPILE	compile selected project only	F7
Ca	PROJECT	COMPILE ALL	compile the different projects	-
Yes	PROJECT	SIMULATION MODE	Open "simulator" See Simulation chapter	-
Co	PROJECT	CONNECTS TO THE TARGET	connect to the target	F5
Dc	PROJECT	DOWNLOAD CODE	download to the target	-
Ins	PROJECT	INSERT RECORD	Insert record (variable, constant, alarm etc.)	-
Del	PROJECT	DELETE RECORD	Delete record (variable, constant, alarm etc.)	-

6.2.2. Project Panel

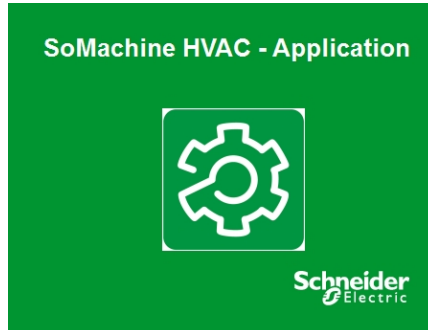
Detailed display of **Panel C / Project**

7. GETTING STARTED

7.1. FIRST START-UP

1. Click on the **Launch SoMachine HVAC** icon on your Desktop
2. Click on the **SoMachine HVAC Application** icon.

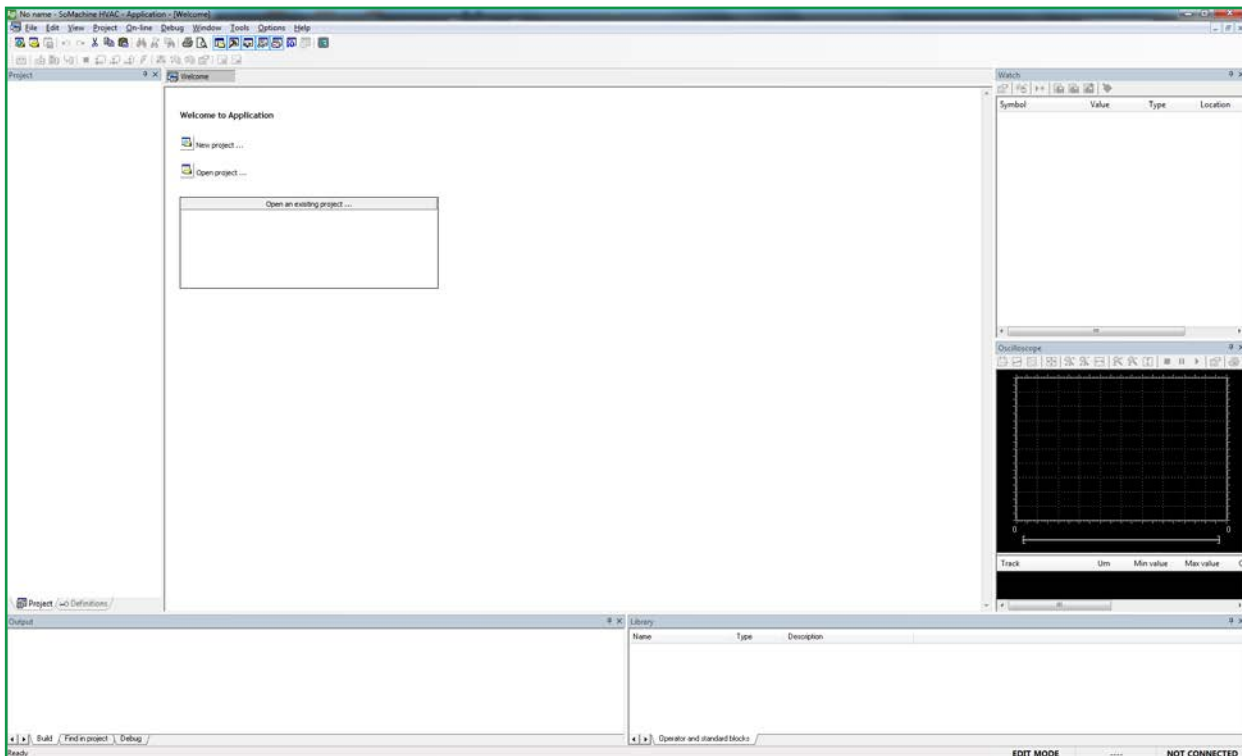
Nome	Ultima modifica	Tipo	Dimensione
SoMachine HVAC - Application	17/07/2014 08:46	Collegamento	2 KB
SoMachine HVAC - Connection	17/07/2014 08:46	Collegamento	2 KB
SoMachine HVAC - Device	17/07/2014 08:46	Collegamento	2 KB
SoMachine HVAC - Simulation	17/07/2014 08:46	Collegamento	2 KB
SoMachine HVAC - UserInterface	17/07/2014 08:46	Collegamento	2 KB



1

2

The program appears as shown in the picture:



You can now:

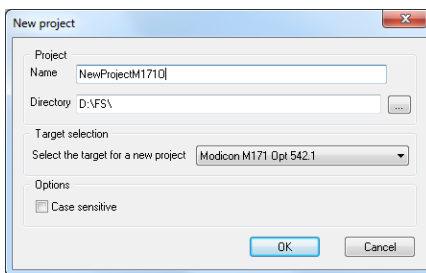
- Create a new project “**New Project ...**”
NOTE: the program suggests selection of the M171O, M171P/M172P or M171P Flush target; the numbers at the side indicate the firmware version (screen)
 - Open an existing project “**Open project ...**”
 - Open recent projects “**Open an existing project ...**” by selecting from a list
- Existing projects have the extension **.ppjs**

7.1.1. CREATING A NEW PROJECT

- Click **New Project ...** or
- select **File > New Project** from the drop-down menu

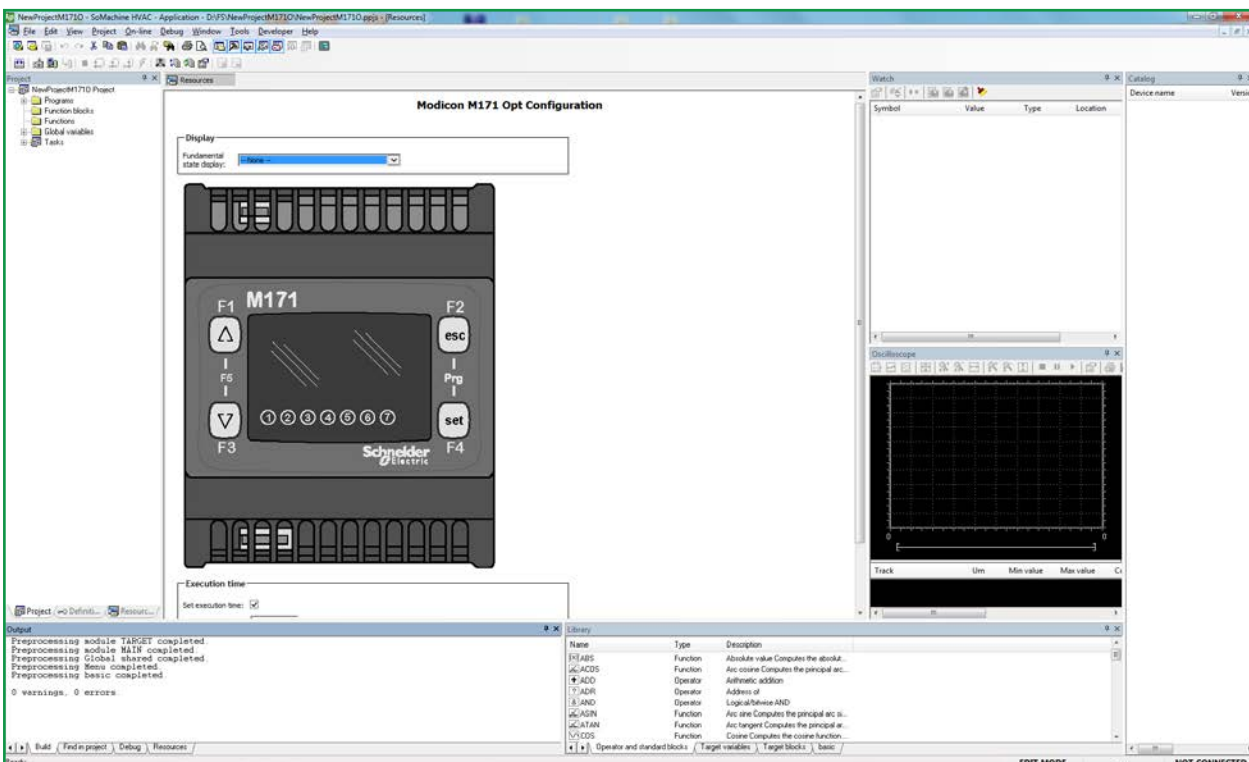
The following procedure illustrates a project for M171 Optimized logic controller. A similar procedure applies for M171P/M172P logic controllers, exceptions are pointed out.

The dialog box shown below appears, in which you can create, name and save a new project:



New Project

The new project appears as shown in the figure:



- For a description of the icons, see **Shortcuts**
- A description of the programming languages is available in the **Online Help**
- The Panels in the development environment are described in the next chapter

The pages that follow explain how to create, develop and test a new application, with examples of libraries, function blocks and elementary programs.

7.5. WRITING A PROGRAM

In the **Edit Panel**, write the program that you intend to develop.

The elements of the development tools are presented generically below.

The appendix Project Development Examples gives some examples of simple programs which you can use to create a project.

7.5.1. PROJECT PANEL / PROJECT TAB

Creating local variables

Using the drop-down menu, select **Variables > Insert** (new record) or click on the respective icon.

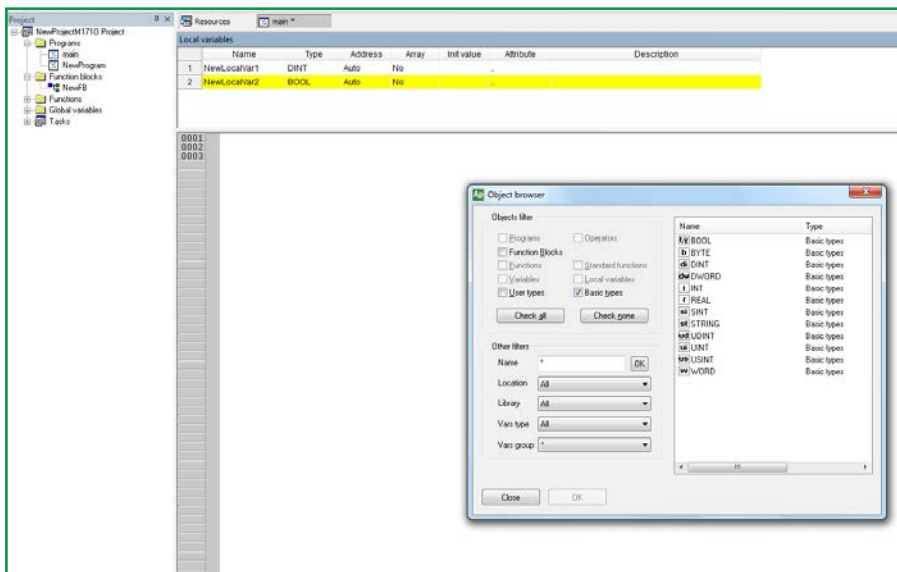
The variable appears in yellow in the central section, and you can define its characteristics by clicking on the respective boxes.

Where multiple choices exist, you can choose from a set of options by clicking on the ellipsis (...).
a dialog window will open

- Name
- Type, to choose from one of the preset options or variables defined by the user (see picture)
- Address: the default setting is automatic
- Array: defines whether the variable is array type (if so, define its dimension) or not
- Init value: initial value
- Description

In the navigation tree below the program folder, the local variables will appear, identified by an icon.

NOTE: The local variables are **NOT** static.



Creating a function block/function

In the same way as we create a program, it is possible to create

- a **function**
- a **function block**

in the desired programming language.

NOTE: Creating functions is available in 4 programming languages. **SFC** language is not supported for functions.

A function or function block is a (sub)program with inputs and outputs:

- a **function** requires **n** inputs and a single output (**RESULT**) with the same name as the function. The function's local memory is initialized each time the function is called.
- a **function block** requires **n** inputs and **m** outputs. The local memory of each instance of the function block is kept between one call and the next (static memory).

Each function or function block can be used within a program by dragging and dropping the icon into the **Edit** section of the program.

The function is used within the program by passing the input variables.

The function block is used within the program as an **instance** in the same way as the declaration of a variable.

Example in ST language of call to function and function block

The screenshot shows a software development environment. On the left is a project tree for 'aaaa Project' containing 'Programs', 'Local variables', 'Function blocks', 'Functions', and 'Global variables'. The 'Local variables' folder is expanded, showing 'functionblockinstancename', 'varinput1', 'varinput2', 'varoutput0', 'varoutput1', and 'varoutput2'. The 'Function blocks' folder is also expanded, showing 'functionblock', 'Input variables', and 'Output variables'. The 'Functions' folder is expanded, showing 'functionname', 'Input variables', and 'Global variables'. The 'Global variables' folder is expanded, showing 'Tasks'.

On the right, there is a table with the following data:

	Name	Type	Address	Array	Init value	Attribute
1	functionblockinstancename	functionblock	Auto	No	0	..
2	varinput1	BOOL	Auto	No	FALSE	..
3	varinput2	BOOL	Auto	No	FALSE	..
4	varoutput0	BOOL	Auto	No	FALSE	..
5	varoutput1	BOOL	Auto	No	FALSE	..
6	varoutput2	BOOL	Auto	No	FALSE	..

Below the table is a code editor showing the following ST code:

```

0001 varoutput0:=functionname(varinput1, varinput2);
0002
0003 functionblockinstancename(input1:=varinput1, input2:=varinput2);
0004 varoutput1:= functionblockinstancename.output1;
0005 varoutput2:= functionblockinstancename.output2;
0006

```

Creating global variables

The navigation tree contains the folder **Global Variables**.

From here you can create new:

- Global variables
- Mapped variables
- Constants

Shared global resources

The shared global resources will appear in this section but will be defined in the **Resources Panel**.

7.5.2. ASSOCIATING A PROGRAM TO A TASK

For a program to run, it must be associated to a **task**.

There are various types of task:

- **BOOT** Task executed once only on starting up the system
- **Init**. Task executed at each download of the application and on starting up the system (after Boot).
NOTE. The associated program initializes slaves and messages according to the configuration, with fixed values that are independent of the run time
- **Timed**. Task runs at regular intervals which can be set by the developer. The default setting is every 100ms
NOTE. Modbus messages do not interfere with this task.
- **Background**. Task runs with low priority after the **Timed** tasks (between the end of one **Timed** task and the start of the next).
- **Modbus**. Task executed to implement Modbus Master, calling relative function blocks, and to send messages.
NOTE Each new project has the **main** program associated to the background task (the **main** program can still be eliminated and/or associated to other tasks).

To activate a task, go to the task you want, right-click and select **Add program**

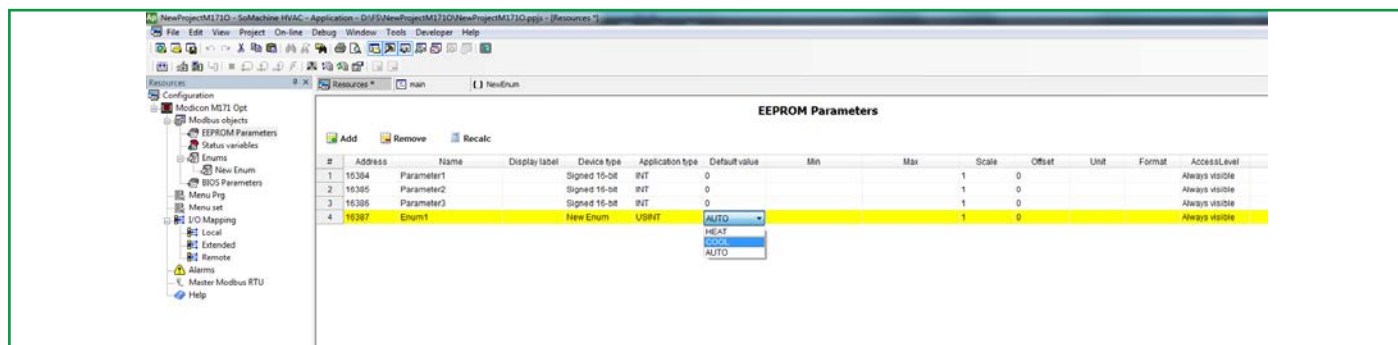
7.5.3. PROJECT PANEL / RESOURCES TAB

The **Resources** tab enables you to define the EEPROM parameters and status variables which may then be used in the application code (they will appear in the section **Project>Global Shared**) and allows you to construct the menu **Prg** (Programming), the menu Set Status (**M171 Optimized only**) and Alarms of the **M171 Optimized/M171P/M172P target** as well as the interface of the current project, which will be displayed in **SoMachine HVAC Device**.

7.5.4. MODBUS OBJECTS

This section enables you to define EEPROM parameters and variables, which can be displayed on the **target, Device** and read using the Modbus RTU protocol.

In the various sections, it is possible to add or remove parameters/variables in the same way as for variables in the **Project** tab:



EEPROM Parameters (M171O)

This section is for creating the variables which the developer intends to save even if the power to the **target** device is switched off.

NOTE: the user of parameters in the code is read-only; to write, use the appropriate functions of **M171 Optimized**.

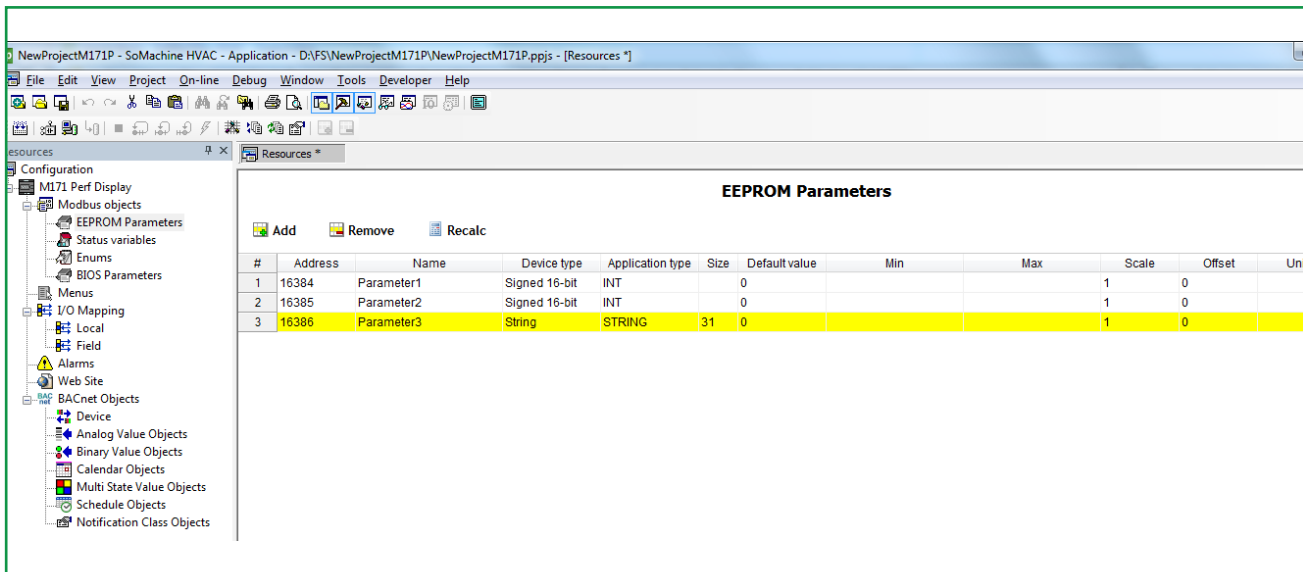
Status Variables

In this section we define the status variables which can be displayed in the menu of the **target** device.

Each variable will have a transcoding on the controller due to the 4-digit / 7-segment display. In the Display label box, you can select the transcoding and see a preview on the display by clicking on the ellipsis (...)

NOTE: some letters will not be displayed (for example **x** and **z**) so there will be a blank space on the display.

If the display label is **SET**, 5E€ will appear on the display.



EEPROM Parameters (M171P/M172P)

Same as **M171O**

Status Variables

In this section we define the status variables which can be displayed in the menu of the **target** device.

In the **target** column, only columns that are visible from the **M171O**, **M171P Flush** or **M171P/M172P target** are indicated.

Refers to both unless indicated

target	column	Description
	IPA	pre-assigned index
	address	resource Modbus address
	Name	resource name which may be used by the developer in the controller application
M171O	display label	name displayed in the application menu of the target (4-digit)
	Device Type	type of data displayed on target and Device
	Default Value /Min/Max	minimum/maximum default values of the resource displayed on target and Device
	Format	display format for Default Value /Min/Max for example XXX.Y display of whole number with decimal point
	Application Type	type of data used in controller application
	Scale/Offset	conversion coefficients between Device Type and Application Type Application Type = scale x Device Type + offset
M171P Flush M171P/ M172P	Size	Significant only in the case of STRING type. Dimension (Length) of the string. Default and max= 31 characters
	Unit	unit of measurement of Device Type displayed on Device and if available with icon on target
M171O	Access Level	See Visibility table. This column does not apply to M171P/M172P
	Read only	enables/disables editing of Status variables.
	Description/Note	free text displayed on Device

Enums

In this section we define the types of additional variable which can be used in the **Device Type** column.

NOTE. see EEPROM parameters and Status Variables table.

BIOS parameters

In this section we define variations in the default BIOS parameters map which is factory-set by Schnedier Electric.

7.5.5. TARGET MENU M1710

The target menu consists of a BIOS menu and an Application menu.

The BIOS menu is factory-set.

The table below defines the main functions of the keys / LEDs of the **target** device.

key	press	Description
F5	short	switch from BIOS menu to Application menu and vice versa
F1 or F3	short	navigate folders and edit values
F2	short	cancel operation (ESC function)
F4	short	access to set menu
F2+F4	short	access to Prg menu
F1/F2/F3/F4	long	managed by developer (by means of target variable sysKeyFunctions[])

The LEDs are managed by the developer by means of **target variable sysLocalLeds[]**

The elements entered in the table in this section will be displayed on Device.

Prg menu

In this section we define the Prg Menu and the folders/variables of which it is composed.

The Prg Menu can consist of one or more folders, defined by the developer, into which are inserted:

- EEPROM parameters
- Status variables
- BIOS parameters
- inputs and outputs

Set menu

In this section we define the Set Menu and the folders/variables of which it is composed.

The set menu is created in the same way as the Prg menu.

The set menu contains the AL folder - see below.

Visibility of menu resources

The visibility of the resources created by the developer is indicated in the following table:

Access Level column	visibility on Device	visibility on target	note
Always visible	YES	YES	object assigned to a Prg or set menu
Level 1	YES	Yes Level 1	
Level 2	YES	Yes Level 2	
Never visible	YES	NO	object NOT assigned to any Prg or Set menu
Never visible	YES Visible in the folder ALL PARAMETERS	NO	

Alarms

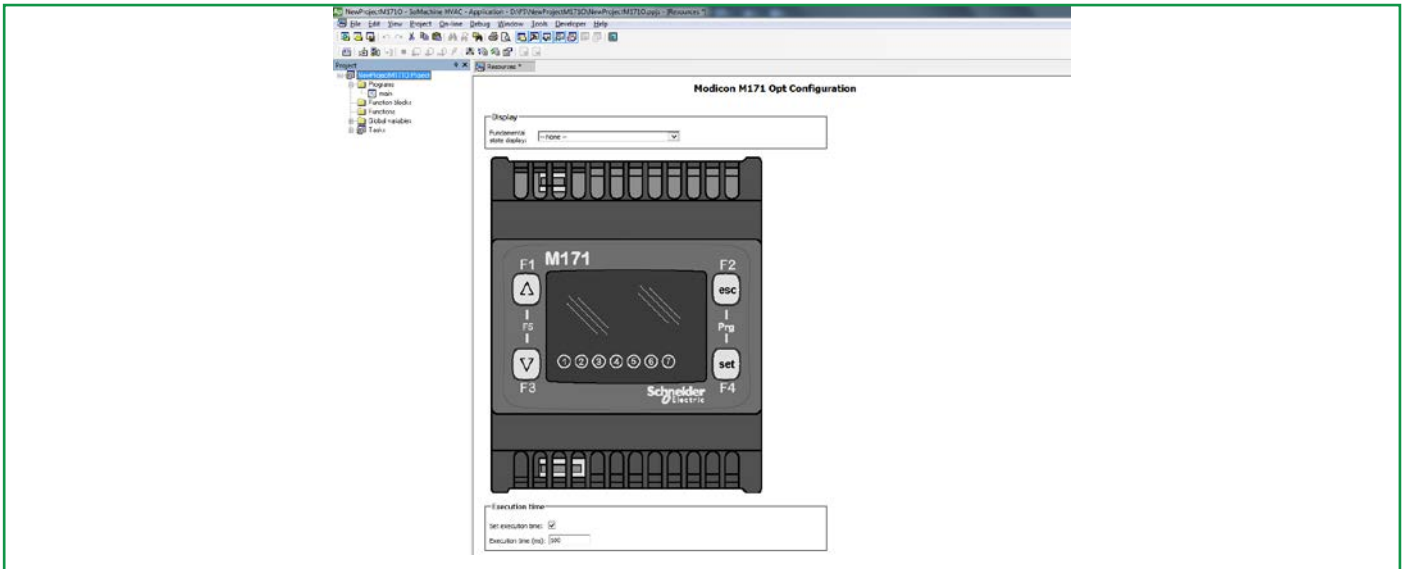
It is possible to define alarm variables which status must be managed by the developer.

If the variable assumes a value **other than zero**, the label will be displayed in the Alarms folder (AL) of the **set** menu in **M171 Optimized**.

In the **M171P/M172P** target it is only a Global type **USINT** declaration.

M171O configuration

From the **Resources** tab, click on **M171 Optimized** to access the following Panel **M171 Optimized Configuration**.



Using the drop-down menu you can set the value of the variable / parameter you intend to show on **M171 Optimized** as the main display (**Fundamental state display**). If you set no value (**none**) the message **PLC** will be displayed.

The values available are the **I/O** and the values defined by the developer in the **Set menu**.

The **I/O** are the local and remote inputs and outputs, including those of the displays.

Execution time

From the same Panel it is possible to set the execution time of the project in milliseconds (**ms**).

The default setting is 100 ms. The available range is from **20 ms** to **100 ms**.

I/O mapping

In this section we define the links between variables and physical **I/O** of **M171 Optimized**.

- **Local**: local variables of the controller module
- **Extended**: variables of the expansion module
- **Remote**: variables on the displays.

7.5.6. TARGET MENU M171P/M172P

The target menu must be created by means of the **UserInterface**.

The BIOS menu is factory-set and is visible from **Device**.

The main functions of the keys/LEDs of the **target** device can be programmed by means of the **UserInterface**
LEDs also programmable from ApplicationMenus.

In this section we define a Menu that does not appear on the display and the folders/variables of which it is composed. The Menu can consist of one or more folders, defined by the developer, into which are entered:

- EEPROM parameters
- Status variables.

Alarms

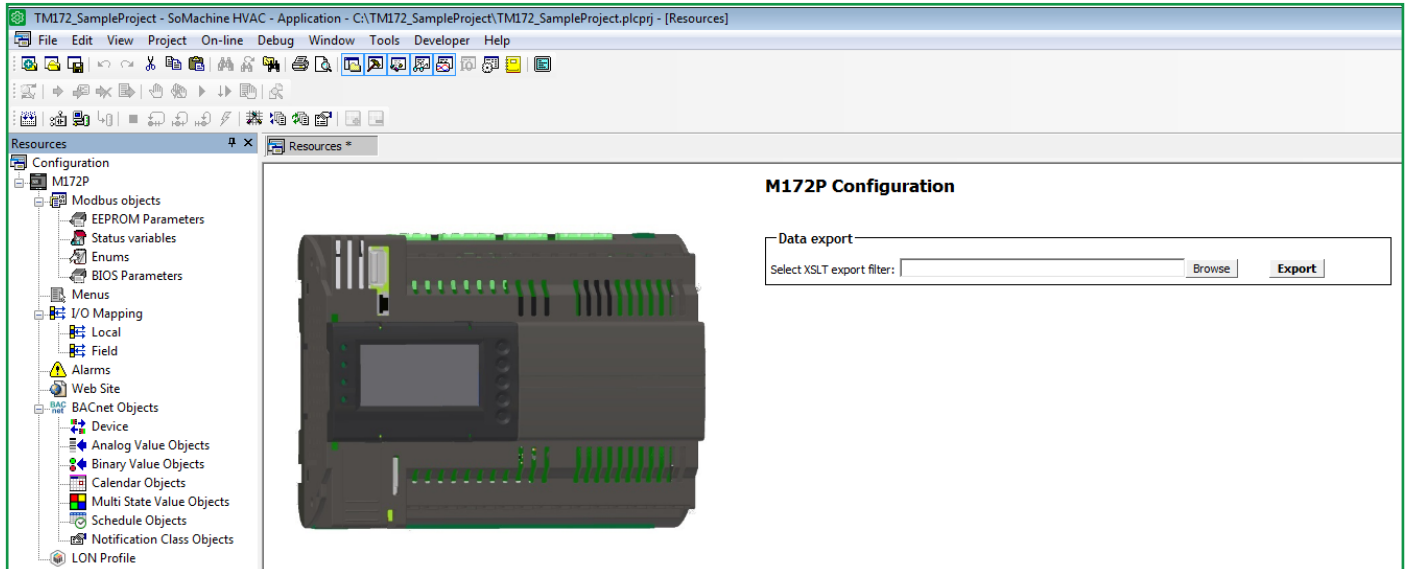
The alarms for **M171P/M172P** are only defined to enable the portability of an **M171O** project.

From the **Resources** tab, click on **M171/M172 Performance** to access the following Panel **M171/M172 Performance Configuration**.

M171P configuration

The screenshot shows a software application window titled "NewProjectM171P - SoMachine HVAC - Application - C:\INVSDATA\D_Z400VFS\NewProjectM171P\NewProjectM171P.pjjs - [Resources]". The interface includes a menu bar (File, Edit, View, Project, On-line, Debug, Window, Tools, Developer, Help) and a toolbar. On the left, a "Resources" tree view is expanded to "M171P Display", showing sub-items like Modbus objects, EEPROM Parameters, Status variables, Enums, BIOS Parameters, Menu, I/O Mapping, Local, Field, Alarms, Web Site, BACnet Objects, Device, Analog Value Objects, Binary Value Objects, Calendar Objects, Multi State Value Objects, Schedule Objects, Notification Class Objects, and LON Profile. The main area displays a "M171P Display Configuration" panel. This panel features a central image of the M171P device. To the right of the device image, there are two configuration sections: "Execution time" with a checked "Set execution time" checkbox and a text box containing "20"; and "Data export" with a "Select XSLT export filter:" label, a "Browse" button, and an "Export" button.

M172P configuration



Execution time

From the same Panel it is possible to set the execution time of the project in milliseconds (ms). The default setting is 20 ms. The available range is from **20 ms** to **100 ms**.

NOTE: See also the section on Associating a program to a task.

I/O mapping

In this section we define the links between variables and physical I/O of M171/M172 **Performance**.

- **Local:** local variables of the **M171P/M172P base module**
- **Field:** variables and/or binding of third-party controllers.

NOTE: If correctly defined, the variables defined in **Resources** will automatically be located in the **Global Shared** section. The project must be saved without errors for the variables to be visible in the Project tab in Global Shared.

Analog Inputs

NOTE that the presence of the inputs depends on the reference.

8. MANAGING PROJECTS

8.1. SAVING PROJECTS

Once you have completed the project you can save it in various forms and send it to third-party **Developers** or **Users**.

File types

Target	File type	Application	Description
M171O	.PPJS	Application	Project created with Application to be opened with Application .
M171P Flush M171P/ M172P	.PPJS	Application Connection	Project created with Application that can be added to an M171P/ M172P/M171P Flush device with Connection .
	.PLL	Application	Library created by the developer to be used in Application or Device .
	.RSM	Application	Compressed .PPJS project
M171P Flush M171P/ M172P	.PAJX	User Interface	Project created with UserInterface to be opened with UserInterface .
M171P Flush M171P/ M172P	.PPJX	User Interface	Same as .PPJS. Shared projects.
M171O	.CFN	Device	Project created with Application to be opened with Device NOTE . The source code is not available but the project can be downloaded to the target device.
M171P Flush M171P/ M172P	.CFN	Device	Project created with Application or Connection to be opened with Device . If the device is created with Connection it will contain both the controller application and the HMI menu. NOTE . The source code is not available but the project can be downloaded to the target device.

8.2. TRANSFERRING PROJECTS AND FILES

The table below shows several operations involving the transfer of projects from one developer to another or from a developer to a user. The data transfer involves the sending of one or more files defined in the previous section.

Project	From -->To	Application	File type	NOTES
<p>The developer wants to transfer a project to another developer. The developer sends a .ppjs file</p> <p>The developer who receives it can open and edit it with Device Application; The developer connects to the target, and compiles and downloads the application.</p>	<p>Developer</p> <p>↓</p> <p>Developer</p>	Application	.PPJS	
<p>The developer sends a .ppjs file to another developer</p> <p>The developer who receives it can open and edit it with Application; The developer connects to the target device and compiles the application. After compilation, the developer selects the menu heading Developer > Open with SoMachine HVAC Device and configures the target device BIOS (parameterization of the target device)</p>	<p>Developer</p> <p>↓</p> <p>Developer</p>	Application Device	.PPJS	
<p>The developer wants to transfer a project to a user The developer sends a .CFN file The user who receives it can open it with Device; The user connects to the target device, downloads the application and configures the target device BIOS (parametrization)</p>	<p>Developer</p> <p>↓</p> <p>User</p>	Device	.CFN	From Device the user CANNOT compile the application
<p>The developer wants to debug an Controller application already present on the target device but does not have the source code. The developer receives the project folder (which includes the binary files and debug symbols) used for downloading the controller application onto the target device. NOTE: the project has not been modified (no modification to the source and no recompilation and/or download)</p>	<p>Developer</p> <p>↓</p> <p>Developer</p>	Application	project folder	

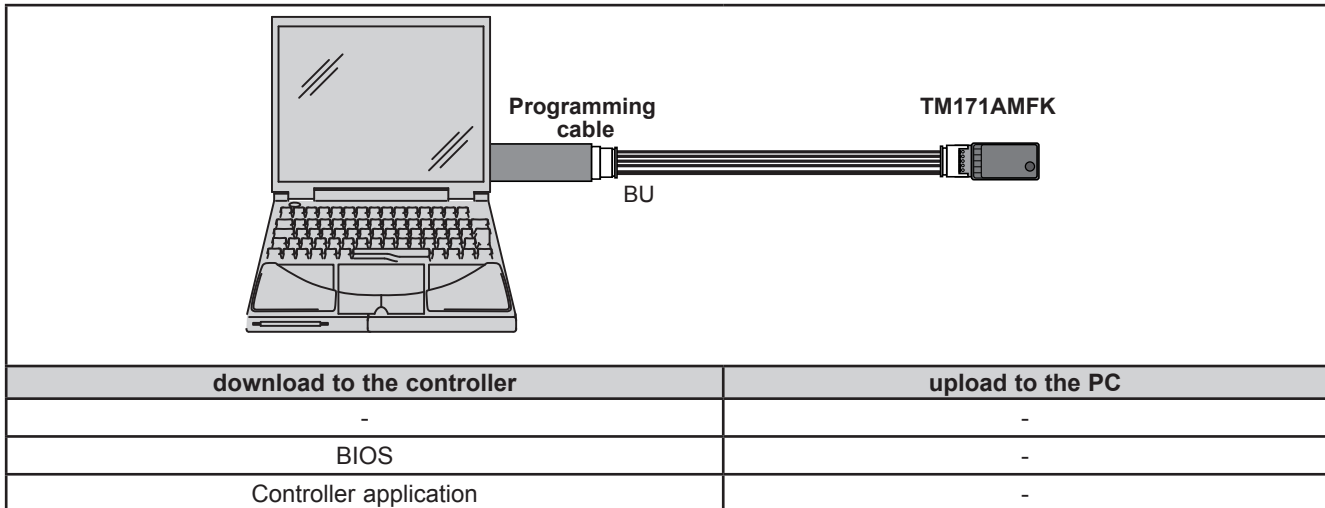
8.3. CONNECTION

To download the controller applications of **SoMachine HVAC** from the PC to the M171O **target device**, various communication modules and accessories are available.

8.3.1. PROGRAMMING OF THE M171O

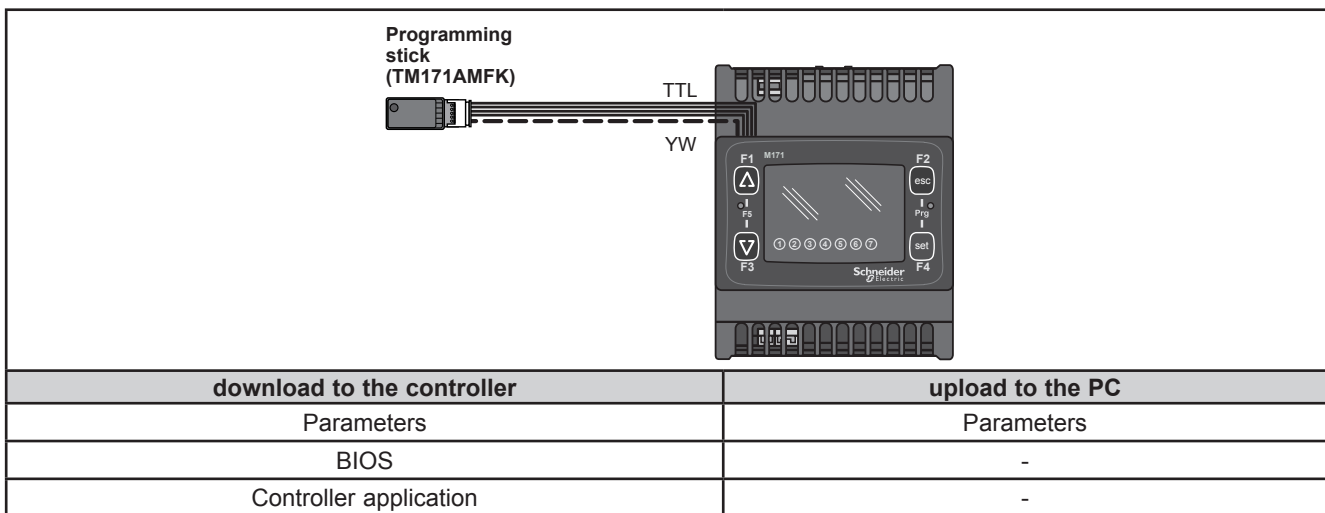
Connection of Programming Stick to PC

NOTE: To connect the PC to TM171AMFK the TM171ADMI and the Blue cable are used.



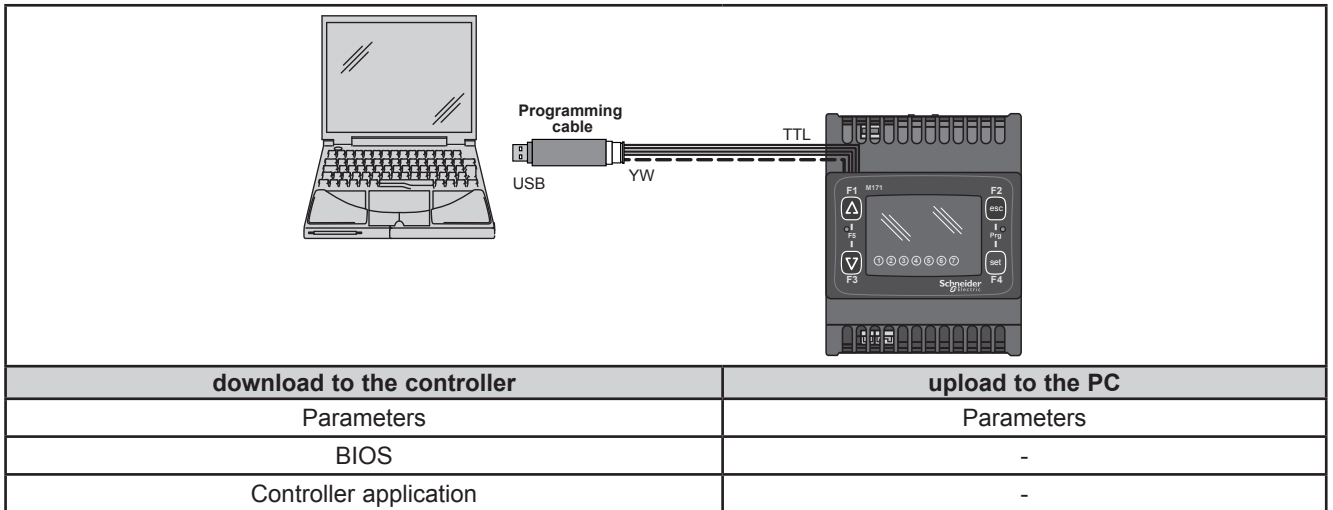
Connection of Programming Stick to M171O

NOTE: To connect TM171AMFK to M171O the Yellow cable is used.



Connection of PC to M1710

NOTE: To connect the PC to M1710, the TM171ADMI and the yellow cable are used.



NOTE: When supplying power to the controller from the PC, **M1710** must not be connected to earth. If a ground (earth) connection exists for the PC as well as the **M1710** a ground loop condition could develop and render either PC or the **M1710** inoperable.

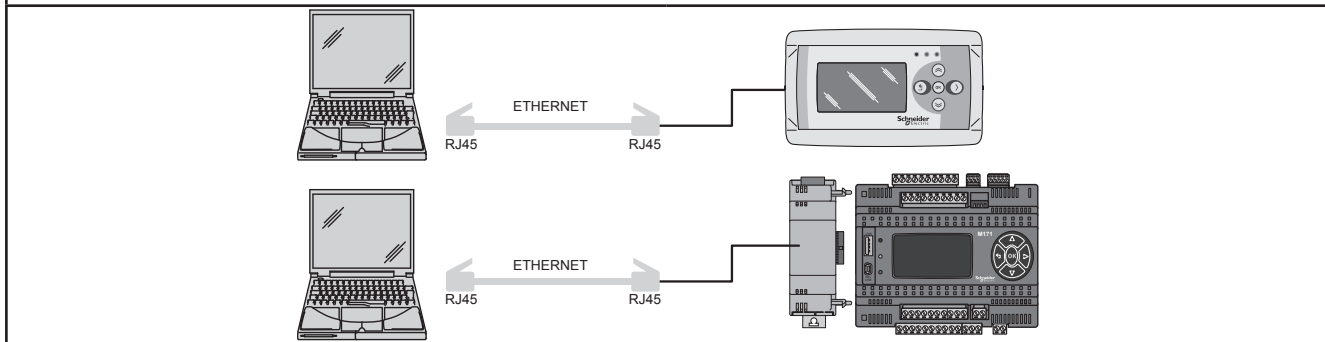
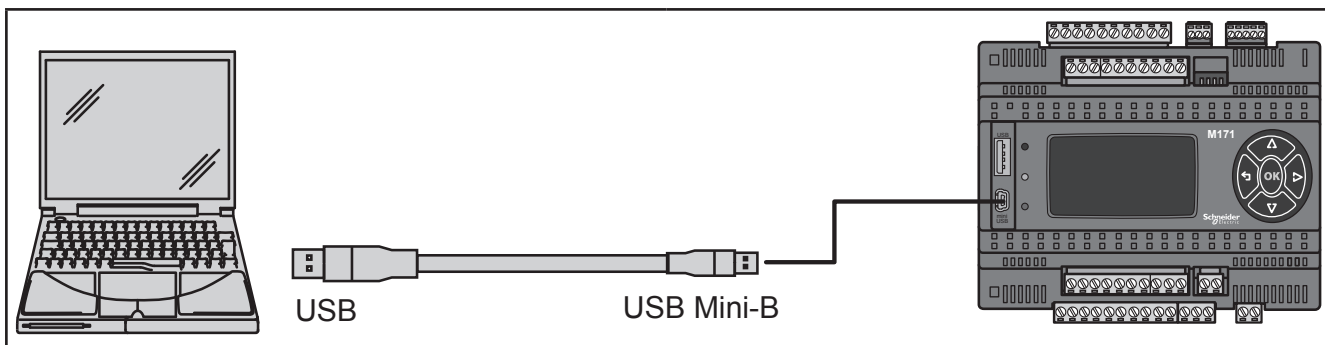
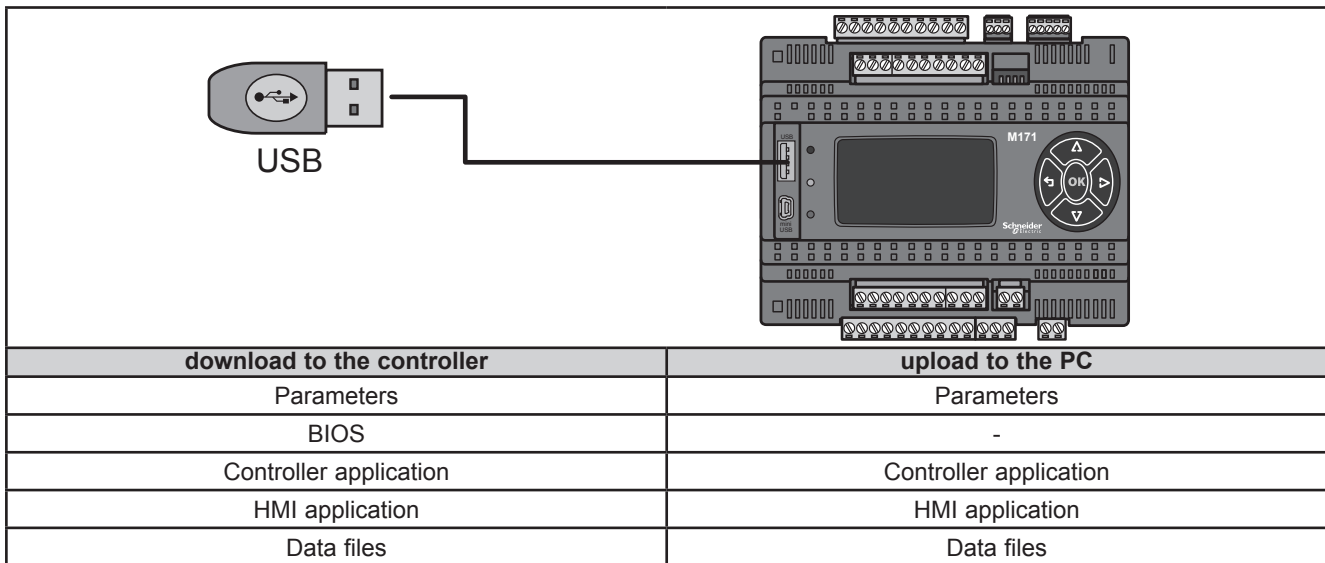
NOTICE

INOPERABLE EQUIPMENT

If you are unsure whether the M1710 is connected to an SELV power supply, disconnect the power supply connection to the M1710 before connecting a PC.

Failure to follow these instructions can result in equipment damage.

8.3.2. PROGRAMMING OF THE M171P

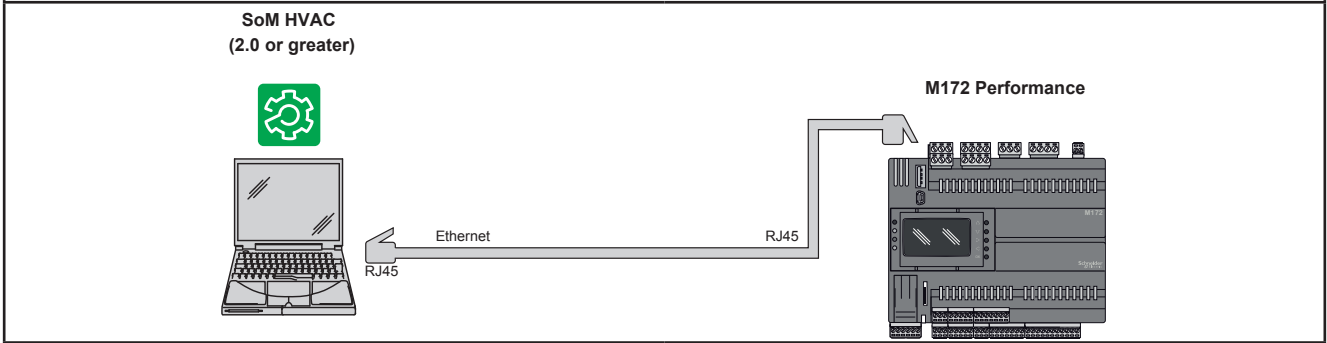
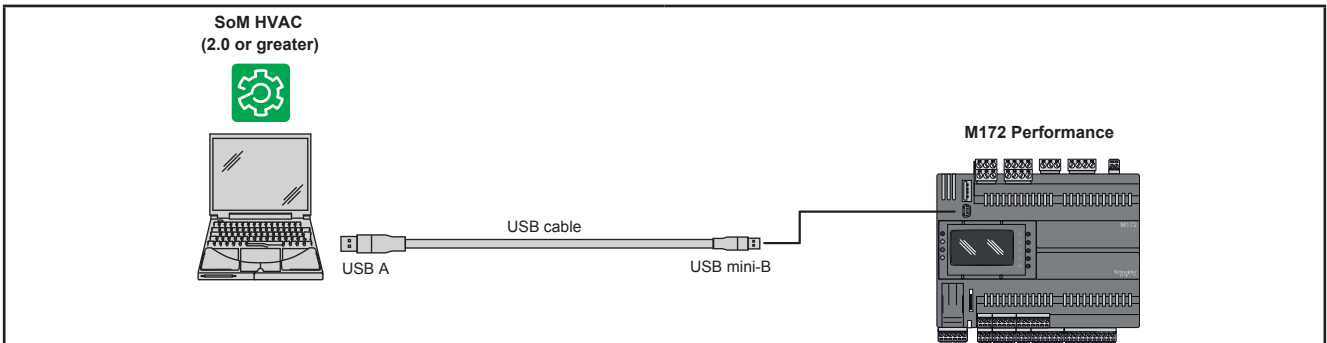
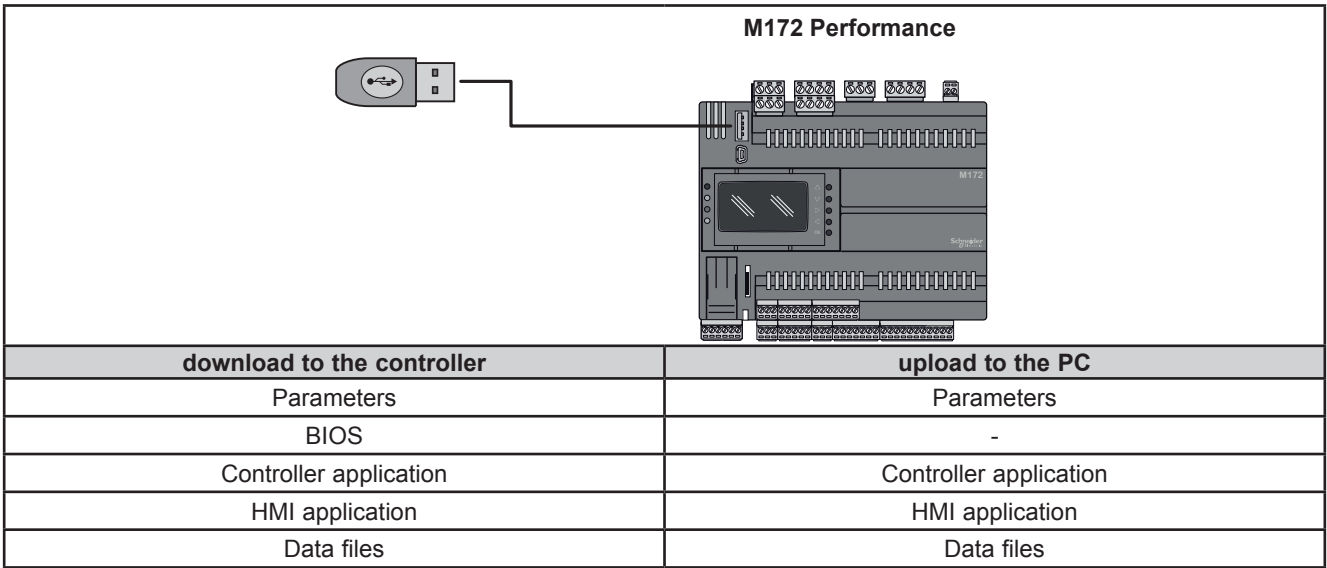


download to the controller	upload to the PC
Parameters	Parameters
BIOS	-
Controller application	Controller application
HMI application	HMI application
Data files	Data files
Commissioning	-

NOTE. M171P: connection via Ethernet requires a communication module.

NOTE. M171P Flush: connection via Ethernet does NOT require a communication module.

8.3.3. PROGRAMMING OF THE M172P



download to the controller	upload to the PC
Parameters	Parameters
BIOS	-
Controller application	Controller application
HMI application	HMI application
Data files	Data files
Commissioning	-

NOTE. USB Mini-B. Used to connect M172 Performance to a PC via mini-B USB cable for debugging, commissioning, downloading, uploading with TM171SW (SoMachineHVAC v2.0 or greater): M172 Performance seen as a virtual COM. Serial communication is performed with a CDC profile (USB).

8.4. DOWNLOAD CONTROLLER APPLICATION ONTO TARGET

Connection and application downloading operations for **M171O** and **M171P/M172P** are different. The connection procedures are illustrated below:

- Network for **M171O**
- USB-RS485 for **M171P/M172P**

The actual download adopts the same procedure for both **targets**.

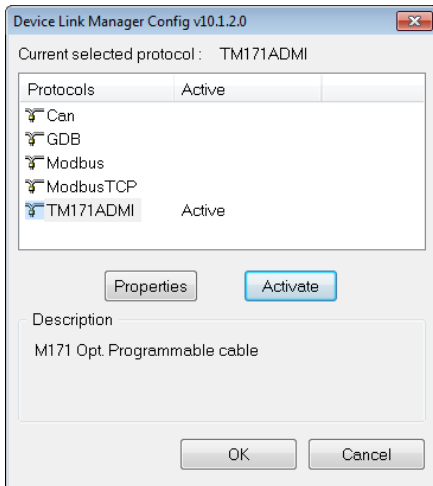
8.4.1. PRELIMINARY OPERATIONS

In order to download the application correctly, the user must:

1. Connect the DMI hardware interface to the PC.
2. Make sure that the driver has been installed.

COM settings

COM settings must be set for the different workspaces: Application, Device and UserInterface. From the drop-down menu, select **On-line > Set up Communication:**

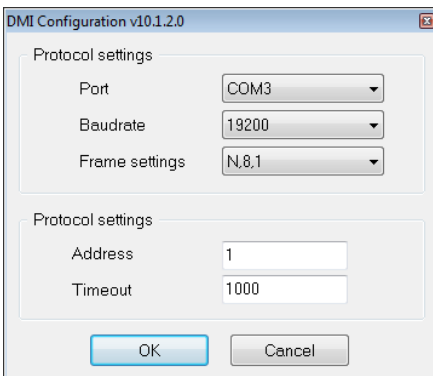


The COM port must be read/set in "Peripherals Management" for it to be recognized. If errors are detected, refer to the paragraph "**M171 Optimized Programming cable connection error**".

For **M171O** select **TM171ADMI** or **Modbus**. If the protocol is not activated press the **Activate** button.

Only with maximum speed 19200 baud. TTL not for use.

The value selected for the COM port will be saved and will reappear each time the program is accessed, until it is changed. The properties are visible and can be edited from the Panel **On-line > Set up Communication > Properties**. Note that the protocol must be activated beforehand.



M171O cable connection error

If the “Error opening serial port” message appears, proceed as follows:

1. Check that the COM port setting in the program is the same as one read in the COM port reading by the **M171 Optimized Programmabing** cable.
2. If they are the same, physically disconnect and reconnect the **M171 Optimized Programming** cable from the USB port. This should make the operating system recognize the interface.
3. Repeat the **M171 Optimized Programming cable** Detection function.

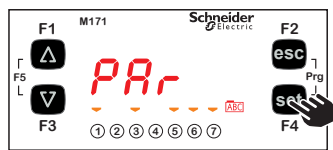
8.4.2. M171O TARGET

M171O has certain parameters in the CF folder for managing the connection between the **target** and **SoMachine HVAC**

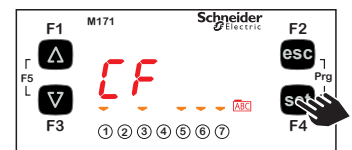
If the **target** is “empty”, for example there is no controller application on the device, **M171O** will display the message Otherwise (controller application exists on M171O) the message PLC will appear on the display. Simultaneously press the UP and DOWN keys to view the message



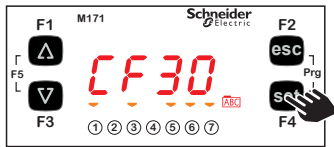
To view the parameter menu, press the Esc and Set keys at the same time. This will open the **PRr** menu.



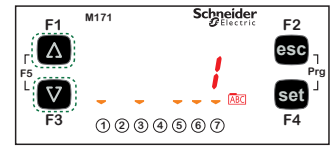
The parameters menu **PRr** contains the controller folders. Press the set key to view folders.



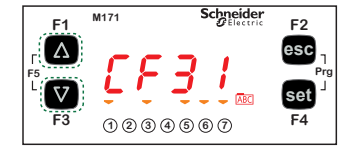
The first folder shown is the **CF** configuration folder. Press the set key to view the folder parameters.



The first parameter shown is **CF30**. To view the value of the parameter press the set key.



Use the UP and DOWN keys to change the value if necessary. To confirm the value press the set key. To exit press esc.



Use the UP and DOWN keys to scroll the other parameters and repeat the procedure to view the values and - if necessary - edit them.

Parameters needed for correct connection between the **M171O target** and **SoMachine HVAC**.

For other parameters and to manage visibility levels refer to Modicon M171 Optimized Logic Controllers, Hardware User Manual.

parameter	description	values	default	visibility	note
CF01 (1)	select COM1 (TTL) protocol	0 = reserved 1 = Modbus	1	2	Must be set to 1
CF30	Modbus protocol controller address	1...255	1	3	Check that the set values correspond to those defined by the Panel On-line > Set up Communication > Properties
CF31 (2)	Modbus protocol baud rate	0, 1, 2 = not used 3 = 9600 baud 4 = 19200 baud 5 = 38400 baud 6 = 57600 baud 7 = 115200 baud	3	3	
CF32	Modbus protocol controller parity	1 = EVEN 2 = NONE 3 = ODD	1	3	

(1) COM1 = TTL / RS485: cannot be used simultaneously

(2) CF31
5=38400 baud (RS485: not supported)
6=57600 baud (RS485: non supported)
7=115200 baud (RS485: non supported)

COM settings

The COM settings are valid on both Application and Device.

8.4.3. M171P/M172P TARGETS

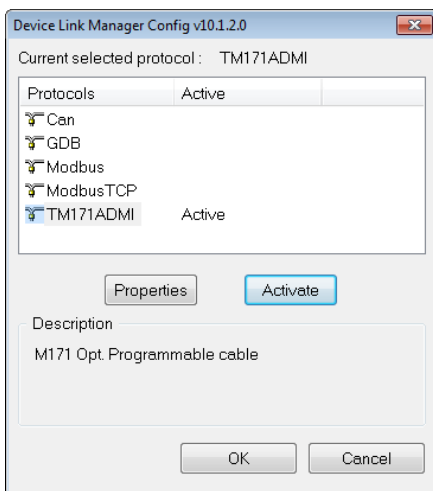
The settings are identical to those given for **M171O**:

Panel **On-line** > **Set up Communication**

NOTE: For M171P/M172P **TM171ADMI** must NOT be selected.

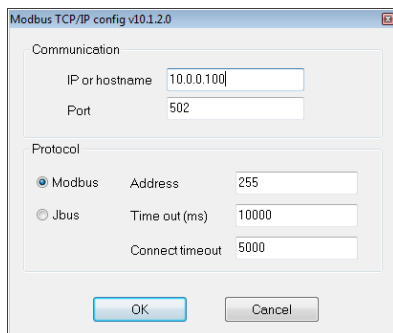
- Select **Modbus TCP** in the case of Ethernet connection using the relevant communication module
- Select **Modbus** in the case of USB/RS485 connection
- Select **CAN** in the case of USB/CAN connection.

If the protocol is not activated press the **Activate** button:

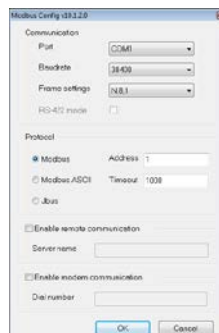


Also in this case, the properties are visible and can be edited from the Panel **On-line** > **Set up Communication** > **Properties**.

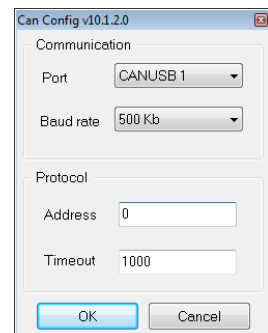
Note that the protocol must be activated beforehand. The Panel displayed depends on which protocol is activated:



ModbusTCP/IP



ModBus serial



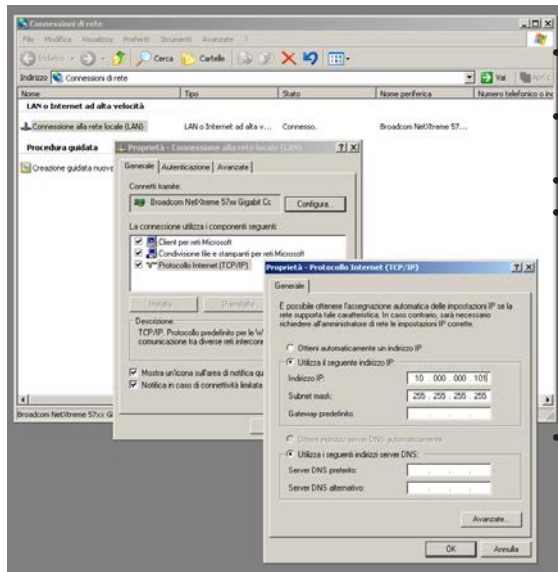
CAN

Modbus TCP

The **IP or Hostname** field is for entering either an IP address (the default setting for M171P/M172P is 10.0.0.100) or a hostname on a local network if the router (static IP) is configured accordingly.

The TCP/IP communication **Port** field is set by default to 502.

Modbus TCP/IP



- Connect the PC Ethernet cable to M171/M172 Performance.
- Double click on the PC icon at the bottom to the right of the application bar.
- Select TCP protocol and click on Properties
- Configure the Ethernet port of your PC with the address shown in the picture (10.0.0.101).
- **NOTE:** The default M171P/M172P configuration is 10.0.0.100: the PC Ethernet port will thus be configured with an address different to the default address (for example 10.0.0.101, the first 3 fields must be the same, the fourth different).
- Click on ok: the PC is configured to dialog with M171/M172 Performance via the Ethernet port.

M171P/M172P has a number of BIOS parameters for managing the connection between the **target** and **SoMachine HVAC** but, unlike **M1710**, it does not have a default menu shown on the on-board or remote display. Therefore in order to view the parameters you need to open **Device**. Parameters needed for correct connection between the **M171P/M172P target** and **SoMachine HVAC**.

On-board RS485

- **M171P** : 1 x RS485 On Board
- **M172P** : 2 x RS485 On Board; RS485-1 On Board + RS485-2 On Board

parameter	description	values	default	note
Baud_RS485_OB	Modbus protocol baud rate On-board	0=9600 baud 1=19200 baud 2=38400 baud 3=57600 baud 4=76800 baud 5=115200 baud	2	
Addr_RS485_OB	On-board RS485 serial address	1...255	1	The actual address is determined by the sum of this value + the value of the DIP switch
Proto_RS485_OB	On-board RS485 protocol selection	2 = uNET 3 = Modbus/RTU 4 = BACnet MS/TP	3	
Databit_RS485_OB	RS485 data bit number On-board		8	Fixed at 8
Stopbit_RS485_OB	On-board RS485 stop bit number	1= 1 stop bit 2= 2 stop bit	1	
Parity_RS485_OB	On-board RS485 protocol parity	0= NULL 1= ODD 2= EVEN	2	

On-board CAN

parameter	description	values	default	note
Baud_CAN_OB	CAN protocol baud rate On-board	2=500 Kbaud 3=250 Kbaud 4=125 Kbaud 5=125 Kbaud 6=50 Kbaud	2	
Addr_CAN_OB	On-board CAN serial address	1...127	1	The actual address is determined by the sum of this value + the value of the DIP switch

PASSIVE ETHERNET PLUGIN

The Ethernet passive plug-in configuration parameters involve the configuration of the TCP/IP communication port (for example 502), the **IP** address, the gateway and the **subnet mask**.

The "Default Gateway" and "Net mask" parameters are of no significance in the local point-to-point network.

For connections via a router the "Default Gateway" parameters must match the IP address, as in the following example:

parameter	description	value	parameter	description	value
Ip_1_ETH_PI	Ethernet passive Plug-in IP address (1st part)	192	DefGtwy_1_ETH_PI	Default Gateway (1st part)	192
Ip_2_ETH_PI	Ethernet passive Plug-in IP address (2nd part)	168	DefGtwy_2_ETH_PI	Default Gateway (2nd part)	168
Ip_3_ETH_PI	Ethernet passive Plug-in IP address (3rd part)	0	DefGtwy_3_ETH_PI	Default Gateway (3rd part)	0
Ip_4_ETH_PI	Ethernet passive Plug-in IP address (4th part)	100	DefGtwy_4_ETH_PI	Default Gateway (4th part)	1

ETHERNET CONNECTION On Board M171 Flush / M172P

The settings are identical to those given for **M171P**: Select **Modbus TCP** in the case of Ethernet connection.

No additional module necessary, ETHERNET is onboard - see table above.

The available on-board RS485 and CAN protocols are the same as **M171P**.

The table below shows the Ethernet protocol parameters.

HMI management

In addition to the BIOS parameters, **M171P Flush** manages the HMI menu:

parameter	description	values	default	note
Hmi_language	Display language (local or remote)	0 ... 65535	0	
HMIList_current	Current HMI	0= Remote HMI 1 / 1= Remote HMI 2 2= Remote HMI 3 / 3= Remote HMI 4 4= Remote HMI 5 / 5= Remote HMI 6 6= Remote HMI 7 / 7= Remote HMI 8 8= Remote HMI 9 / 9= Remote HMI 10 10 = not used 11= Local HMI	11	Local HMI is identified on the display as Network In Connection as HMI Remote HMI is identified In Connection as Remote HMI

Remote HMI

Ten remote menus are available. The first menu parameters are listed below. The others are the same.

parameter	description	values	default	note
HmiList_ID_1	Remote HMI 1 navigation ID list	0 _ 254	0	
HmiList_Res_1	Remote HMI 1 navigation resource type	1=RTU (RS485 Modbus RTU) 2=TCP (Modbus TCP) 3=CAN (CAN)	3=CAN	
HmiList_Addr_1	Remote HMI 1 navigation resource address for CAN, RTU and TCP (IP part 1)	0 _ 255	0	for example CAN: 2. 500000 for example 485s: 1. 38400. P81 ETH: 010.000.000.100
HmiList_Addr_2	Remote HMI 1 navigation resource address for TCP (IP part 2)	0 _ 255	0	
HmiList_Addr_3	Remote HMI 1 navigation resource address for TCP (IP part 3)	0 _ 255	0	
HmiList_Addr_4	Remote HMI 1 navigation resource address for TCP (IP part 4)	0 _ 255	0	
HmiList_File_1	Remote HMI navigation file 1 (DOS 8.3 uppercase format)	alphanumeric string, 8 characters	*****	typically the default name will be HMIREM.KBD

CONNECTING TO THE TARGET

Before connecting, the bottom right of the screen displays the message **NOT CONNECTED**.

From the drop-down menu select **On-line > Connect**.

If connection is successful, the following screen appears in the menu bar:



This screen will remain present as long as the **target** device is connected. To interrupt the connection, repeat the **On-line > Connect** procedure. At the bottom right of the Panel is a connection bar divided into 3 sections.

Before connecting to the target the Panel shows that there is no connection:



When connected to the **target** there are various situations shown below:



- **CONNECTED** (in green): if the connection is established and the **target** application coincides with the **SoMachine HVAC** application.
- **DIFF.CODE** (in yellow): if the connection is established and the **target** application does NOT coincide with the **SoMachine HVAC** application.



- **CONNECTED** (in green): if the connection is established.
- **NO CODE** (in blue): if the connection is established and there is no application on the **target**.

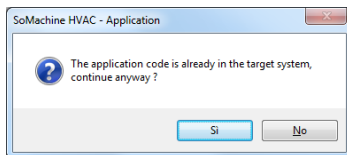


- **CONNECTED** (in green): if the connection is established and the target application coincides with the **SoMachine HVAC** application.
- **SOURCE OK** (in green): if the connection is established and the target application coincides with the **SoMachine HVAC** application.
- **ERROR** (in red): if a communication error is detected.

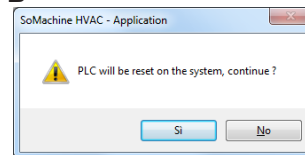
DOWNLOAD

To download the application, press **F5** or select **Communication > Download code** from the drop-down menu. The following dialog boxes will appear:

A



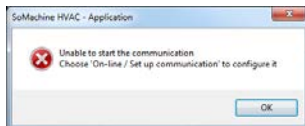
B



- A.** If an application is already present, you will be asked for confirmation to overwrite the previous version.
- B.** Message that the controller will be reset.

If you confirm with “Yes” the application will be downloaded to the target. The process may take a few seconds: the “**Download in progress**” window appears to show the status of the download.

If a communication error is detected, the following dialog box appears:



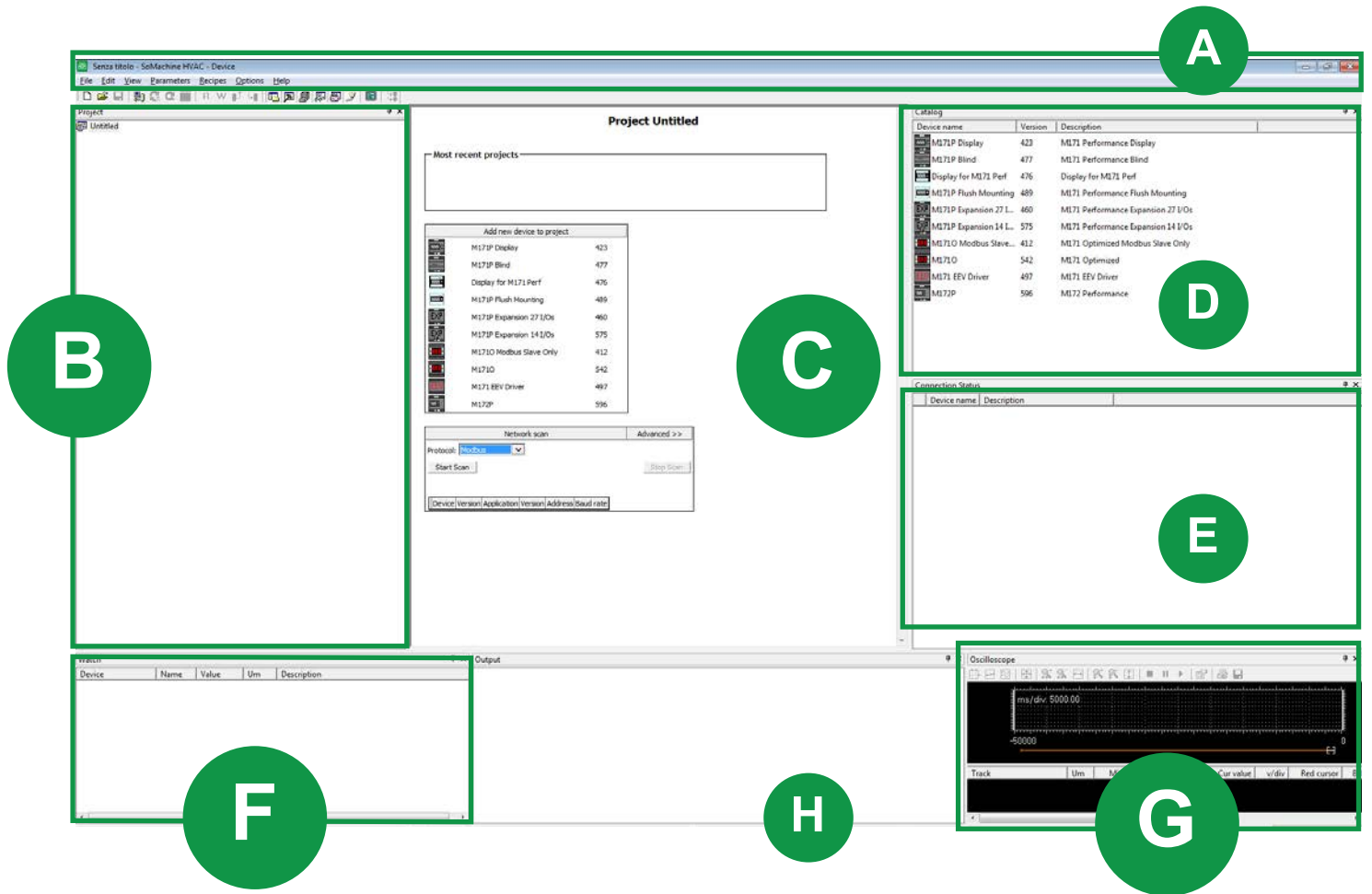
9. DEVICE PANEL DEFINITIONS

The **Device** workspace is shown below with screenshots and descriptions for **M171O**. Where necessary, the differences compared to **M171P/M172P** are highlighted.

The **Device** workspace has various sections for managing an controller application generated by a developer, for example **M171 Optimized** parameter configuration, testing, BIOS updating and application downloading to the target device.

NOTE: the source code of the application is NOT visible from **Device**.

The sections (Panels) are listed below:



Section	Panel Name Section	Panel	Description
A	/	Menu bar	This section shows the work tools in the form of icons. To display or hide the toolbars, right-click and select or deselect the desired tools. Some tools can be selected by means of key combinations (shortcuts).
B	Project	Project	The Project Panel enables you to manage one or more controllers.
C	/	Edit Panel	According to the selection made in the Project Panel, the following appear in this section: <ul style="list-style-type: none"> • BIOS parameters (parameters + I/O values). • parameters and variables defined by the developer in Application associated to the menu.
D	Catalog	Applications catalog	This Panel shows the applications available from the catalog.
E	Connection Status	-	This Panel shows the connection status of the targets in the Project Panel.
F	Watch	Display variables with application running	This Panel enables you to manage variables debugging by displaying their status when the application is running and connected to the target device.
G	Oscilloscope	Display variables graph	This Panel enables you to manage variables debugging by displaying their status in graph format when the application is running and connected to the target device.
H (1)	Output	Messages	This Panel shows the messages relating to the development of the project (file opening, reading/writing errors, status of connection to device, etc). NOTE: the connection to the target device is also visible in the bottom right of the screen by means of colored messages (see Application Download).

(1) Panel not visible at first start-up.

To display it, check the heading in the drop down menu **View > Output**.

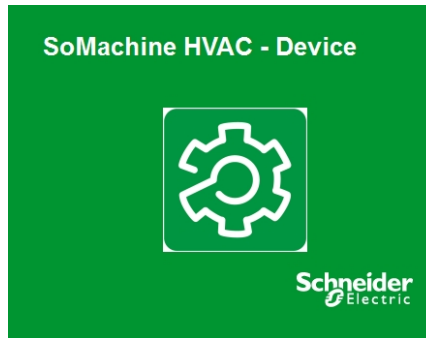
The Panels appear or remain hidden (see **Shortcuts**) and can be organized as the user wants in terms of size and layout. **SoMachine HVAC** saves the last settings made.

10. FIELD OPERATION

10.1. FIRST START-UP

1. Click on the **Launch SoMachine HVAC** icon on your Desktop
2. Click on the **SoMachine HVAC Application** icon.

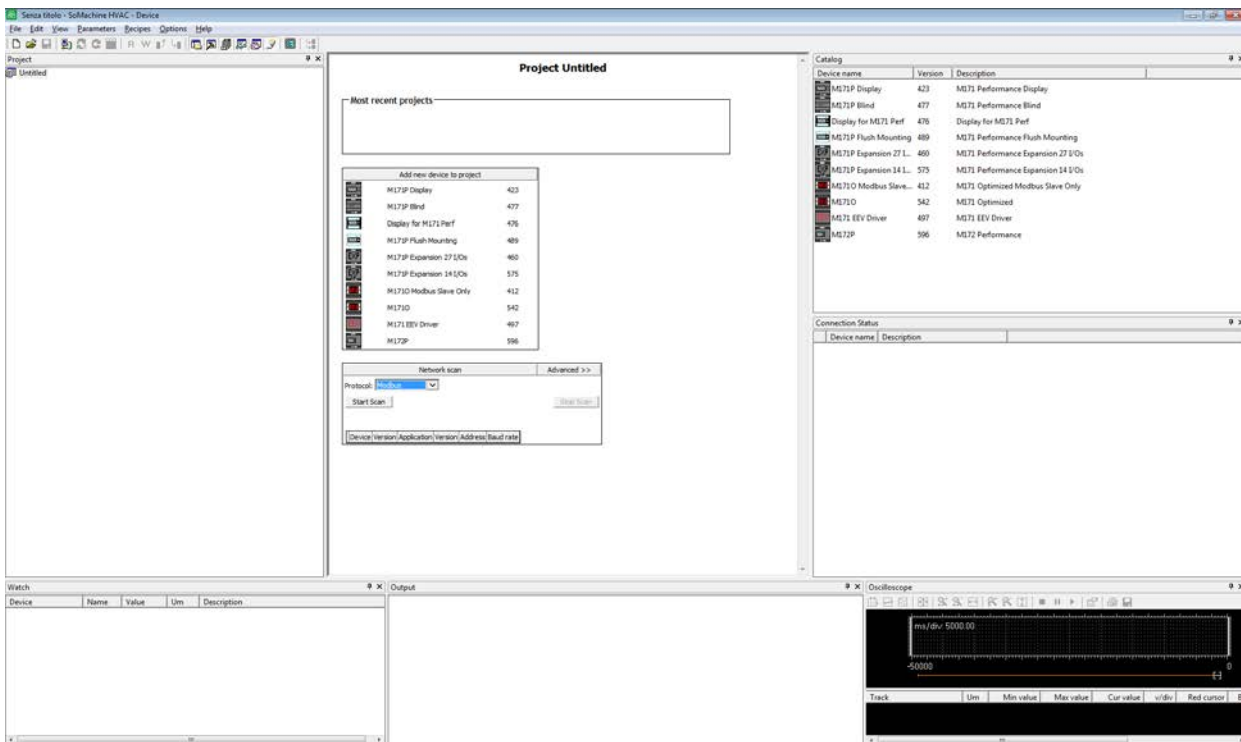
Nome	Ultima modifica	Tipo	Dimensione
SoMachine HVAC - Application	17/07/2014 08:46	Collegamento	2 KB
SoMachine HVAC - Connection	17/07/2014 08:46	Collegamento	2 KB
SoMachine HVAC - Device	17/07/2014 08:46	Collegamento	2 KB
SoMachine HVAC - Simulation	17/07/2014 08:46	Collegamento	2 KB
SoMachine HVAC - User Interface	17/07/2014 08:46	Collegamento	2 KB



1

2

The program appears as shown in the picture:



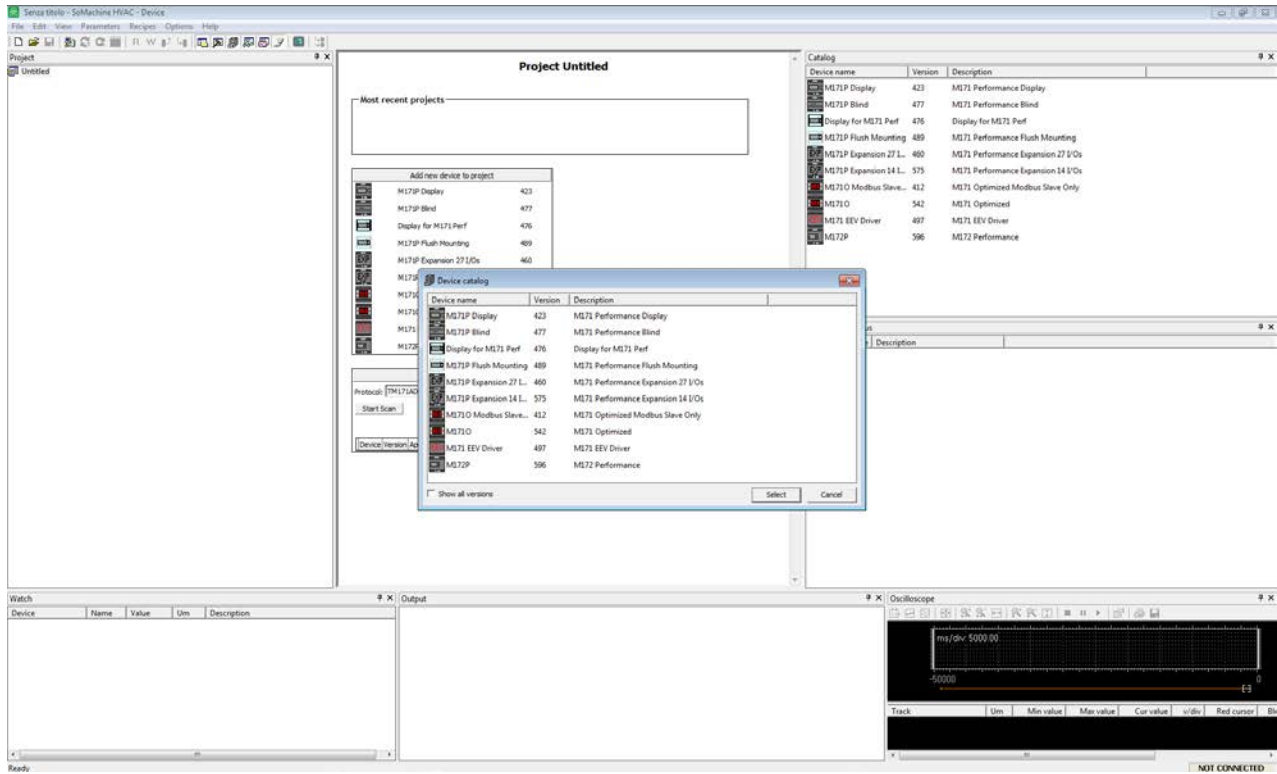
You can now:

- Create a new project **.CFN** “**New Project ...**”
- Open an existing project “**Open project ...**”
- Open recent projects “**Most recent projects ...**” by selecting from a list

Device projects have the extension **.CFN**

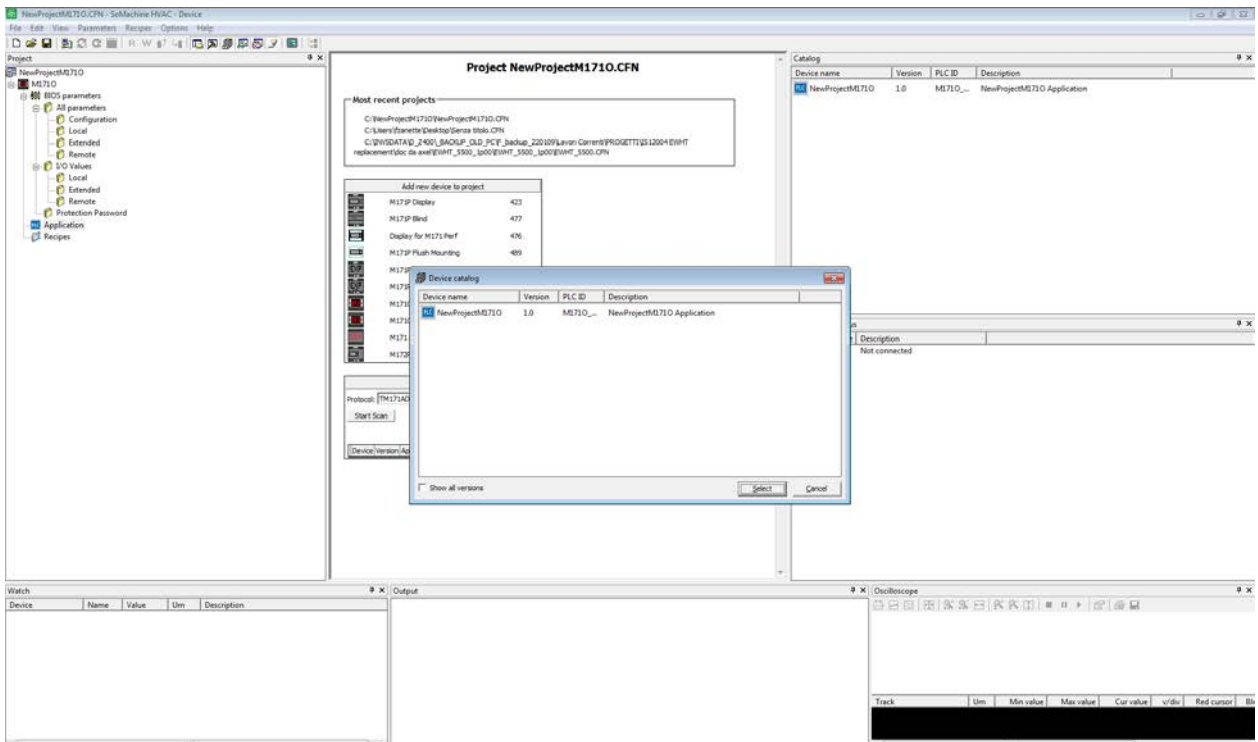
In the same project, you can add more than one drive and associate to each drive an controller application created with **Application**.

A new **Untitled** project is empty. Right-click to add a new drive to the existing project in the catalog by selecting **Add**.
 Alternatively click on the chosen target in the box 'Add new device to project' or drag'n'drop from the Catalog.



10.2. OPENING THE CONTROLLER APPLICATION

In the same way you can add the respective controller application present in the catalog. The catalog can contain various applications but only one can be associated to a project. The controller application will be added to the drive with name_1.



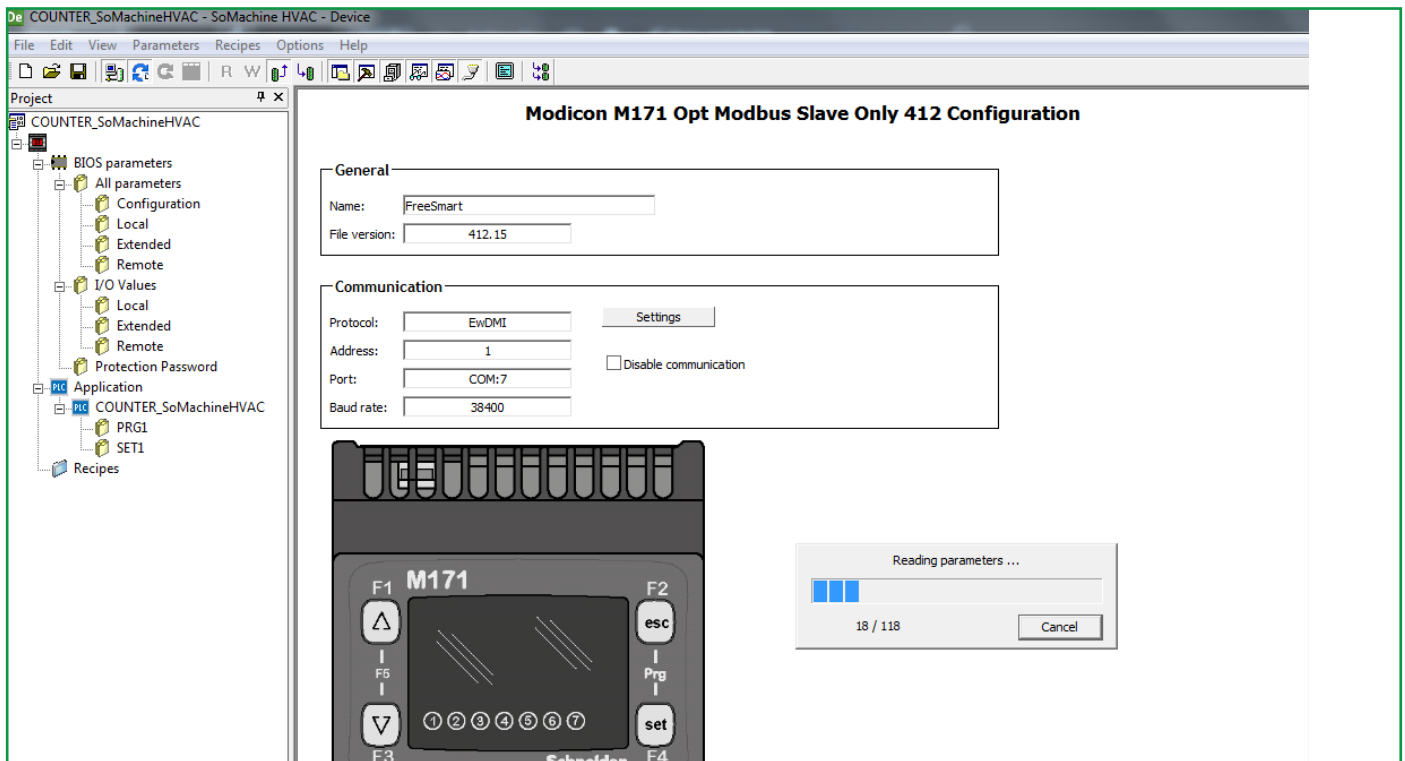
Existing projects can be opened as described previously or directly from **Application**: the menu **Developer > Open with Device** will start the **Device** containing the project created by the developer with the development environment **Application**.

10.3. CONNECTING TO THE TARGET DEVICE

Device requires you to connect to the target device again (click on the respective icon). The central Panel shows the protocol references, address, etc. The default name of the target device is M171O_1. By using the copy/paste function, you can create a twin project M171O_2, change the BIOS parameters and download them to the target device or a second target device. When the target device is connected, you can read or write parameters.

10.3.1. READING PARAMETERS

The term “parameter” refers to any resource present on the target device: genuine parameters, I/O and variables of the application present on the target. By default, the project contains a list of the BIOS parameters and I/O values with their respective default settings. From the target device, the user can read all or some of the parameters. The default value of the analog inputs and outputs is 0.0; the digital inputs and outputs are set to FALSE. To read a parameter: select it with the mouse and click on the icon R. To read the parameters: use the drop-down menu **Parameters>Read all device parameters**. The operation takes a few seconds.



If there are configuration parameters which have default values other than those present on the target device, they will be highlighted in an appropriate color (blue)

WRITING PARAMETERS

The user can download all or some parameters to the target device, either with their default values or with modified values inserted by the user.

To download all the CL parameters (local BIOS) and no others:

Select all the parameters - drop-down menu **Parameters>Select all** (CTRL+A) or using the respective icon. The parameters are all highlighted in yellow. Write drop-down menu **Parameters>Write selected** (CTRL+SHIFT+W) (or using the icon **W**). All the CL parameters will be downloaded to the target device.

As explained in the **NOTES**, you can also download the values of the I/O variables set by the user or their default values.

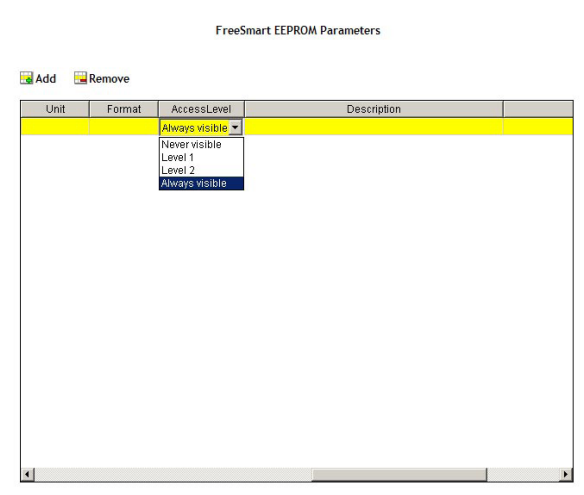
Downloading **all** the parameters (drop-down menu **Parameters>Write all device parameters**) replaces all the values present on the target device with the values listed in the Value column.

If you select Parameters>Write all default values, all the default values including I/O will be downloaded to the target device.

NOTE: the inputs are read-only and are shown in grey. See the following Parameter values table:

colour	column	description	cases
black	value	value aligned with default	target already read
blue	default	default column value different from value in value column	target already read
red	value	value NOT aligned with default	Device just opened value changed by user
grey	name	read-only parameters	I/O values analog inputs AI digital inputs DI
green	name	parameters not visible on target display	See visibility table

When declaring the status variables and EEPROM parameters in **Application** you can also define their level of visibility in the **M171O target** menu as shown in the figure (has no effect on M171P/M172P):



BIOS Download

The user can also upgrade the **M1710 BIOS** from Device. Select the name of the **target** and click the right mouse button. Select **BIOS download** and open the **.fwf** file that you want to download.

The **Direct** mode is executed via an **M171 Optimized Programming cable** directly on PC whereas the **MultiFunctionKey** mode requires passing through **Programming Stick** and then downloading onto **M1710**.
BIOS Download not available via RS485.

Do NOT power up M1710 externally (only power up via an **M171 Optimized** programming cable)

The default BIOS menu of **M171P/M172P** is visible from the folder **All Parameters**.

In the **All Parameters** folder you can also view user parameters defined with **Application**
Downloading the controller application and the **HMI** menu is only performed from **Device**.

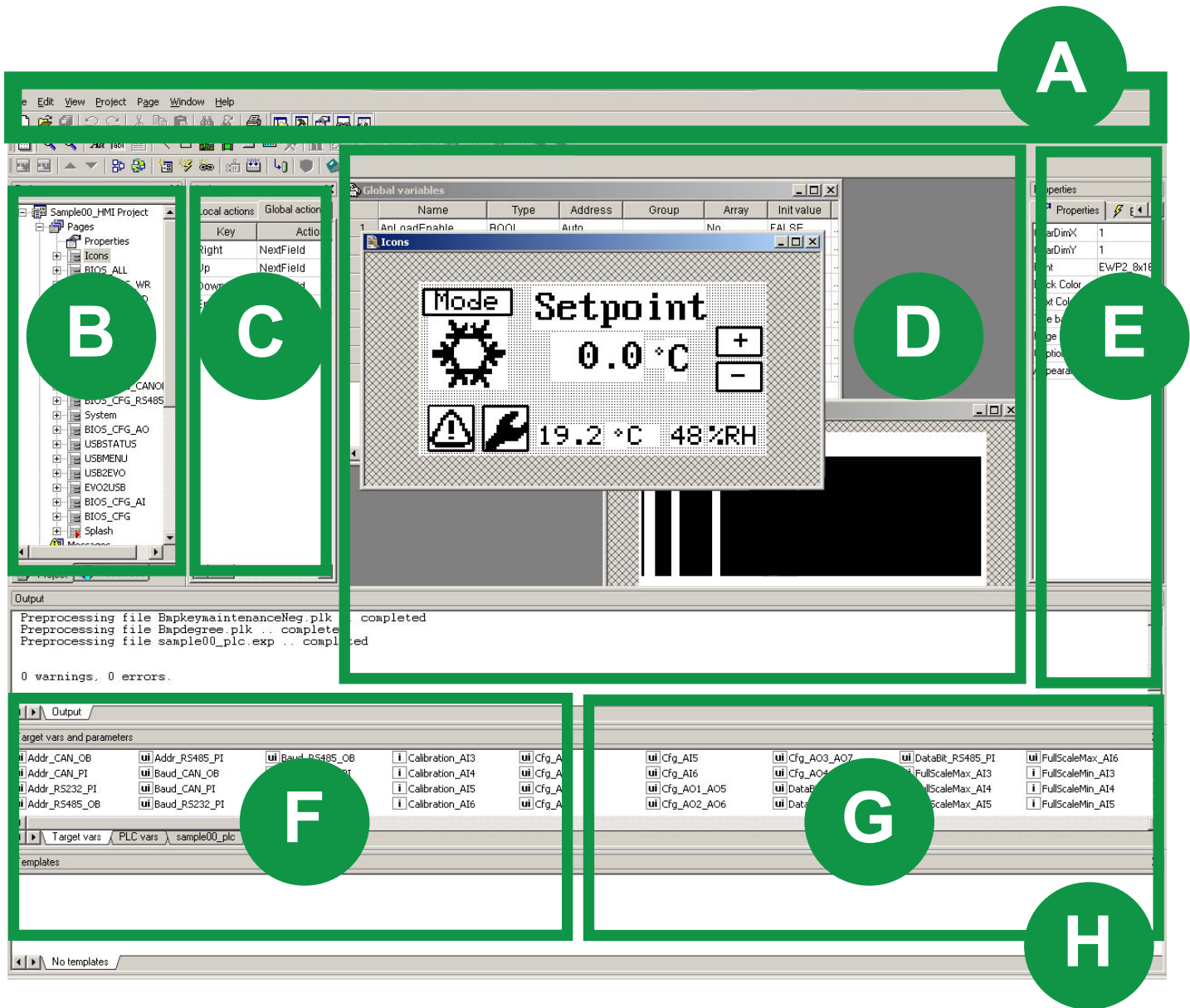
The default BIOS menu of **M171 Performance Flush** is visible from the folder **All Parameters**. In the **HMI management** folder you can view the parameters for managing local and remote HMI pages.

In the **All Parameters** folder you can also view user parameters defined with **Application**.

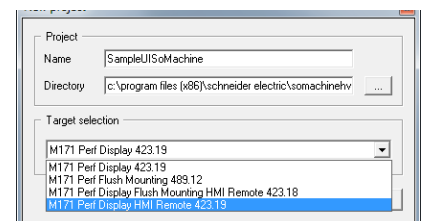
11. USER INTERFACE PANEL DEFINITIONS

The **User Interface** workspace dedicated to developers features various sections for creating and managing customized pages, for example defining the graphical interface of **M171 Performance**, using and creating templates and downloading **HMI** onto the **target** device.

The sections (Panels) are listed below:



The **User Interface** workspace for **M171 Performance Flush** requires that page creation and management and local **HMI** downloading occurs on **M171 Performance Flush**. The characteristic of **M171 Performance Flush** is that it manages **M171P** remote menus created with **UserInterface**. The menus that **M171 Performance Flush** will use as **remote HMIs** must be created specifically as such. See **Creating a new project**.



Section	M171P/M172P Name / Section User Interface	M171P/M172P	Description
A	/	Menu	This section shows the work tools in the form of icons. Some tools can be selected by means of key combinations (shortcuts).
B	Project	Project	The Project M171P/M172P is divided into 2 tabs: <ul style="list-style-type: none"> • Project: for managing the pages that will be shown on the M171P/M172P graphical displays • Resources: for managing images, lists and strings that may be used in the pages
C	Actions		According to the selection made in the tab, the following appear in this section: <ul style="list-style-type: none"> • Local Actions • Global Actions actions linked to the User Interface keys.
D	Document bar	Edit M171P/M172P	This M171P/M172P is used to configure the graphics of the pages.
E	Properties	View properties	This M171P/M172P is used to define the properties of the pages or of their constituent elements
F	Output	Messages	This M171P shows messages relating to the development of the project (file opening, compilation errors, etc).
G	Target vars & parameters	View target variables and parameters	The M171P is divided into 2 or more tabs: <ul style="list-style-type: none"> • Target vars contains the variables/BIOS parameters of M171P. • PLC vars contains the variables/parameters defined in the controller application of Application installed in the M171P base module where the HMI will be downloaded. • Other tabs display variables/parameters with Modbus address to be shown on the HMI interface. NOTE: these applications have been downloaded using SoMachine HVAC onto other M171P/M172P modules connected in a network to the M171P/M172P onto which the HMI that you intend to develop will be downloaded.
H	Templates	View available templates	This M171P/M172P shows the pages created as “templates”, for example as reusable references available in the catalog.

The Panels may be visible or hidden at first start-up.

To view them, check the relative heading in the drop-down menu. For example: **View > Templates**

The Panels can be organized as the user wants in terms of size and layout.

SoMachine HVAC saves the last settings made

12. TO START USER INTERFACE

UserInterface vs Connection

As already mentioned in the **TARGET** chapter and as described in the **Connection** chapter, **UserInterface** is typically opened from **Connection** by clicking on the HMI node.

Read the **Connection** chapter before developing a menu in **UserInterface**.

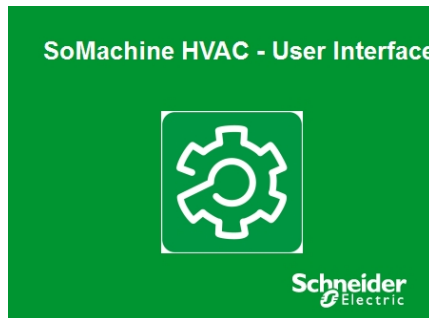
To get started and create a simple **stand-alone** menu follow the outline illustrated below.

A menu created in this way is subsequently linked to a Connection project. See corresponding chapter.

12.1. FIRST START-UP

1. Click on the **Launch SoMachine HVAC** icon on your Desktop
2. Click on the **SoMachine HVAC User Interface** icon.

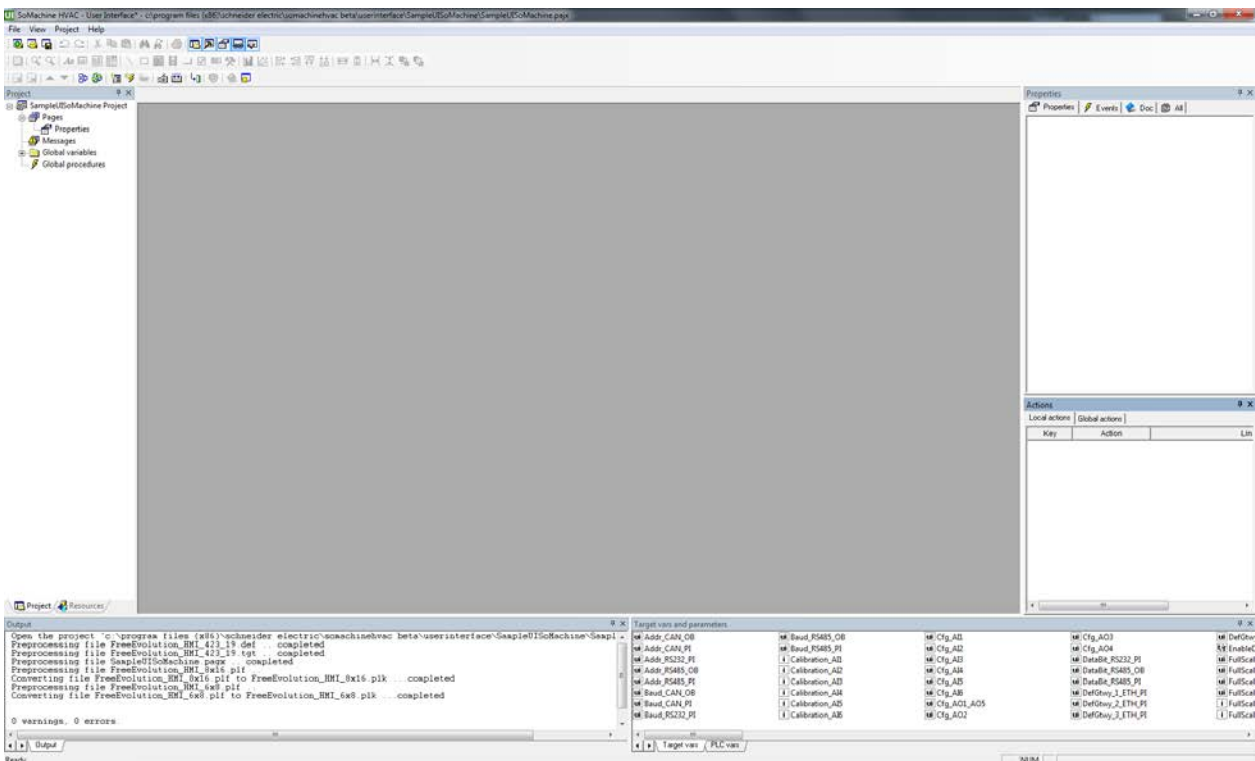
Nome	Ultima modifica	Tipo	Dimensione
SoMachine HVAC - Application	17/07/2014 08:46	Collegamento	2 KB
SoMachine HVAC - Connection	17/07/2014 08:46	Collegamento	2 KB
SoMachine HVAC - Device	17/07/2014 08:46	Collegamento	2 KB
SoMachine HVAC - Simulation	17/07/2014 08:46	Collegamento	2 KB
SoMachine HVAC - User Interface	17/07/2014 08:46	Collegamento	2 KB



1

2

The program appears as shown in the picture:



You can now:

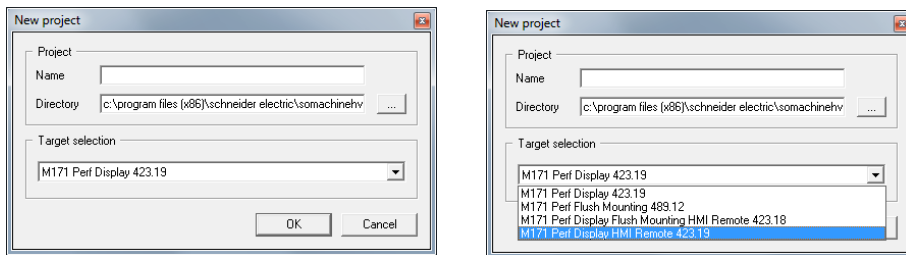
- Create a new project “**New Project ...**” - **CTRL+N**
- Open an existing project “**Open project ...**” - **CTRL+O**

Existing projects have the extension **.pajx**

12.1.1. CREATING A NEW PROJECT

- Click **New Project ...** or
- Select **File > New Project (CTRL+N)** from the drop-down menu

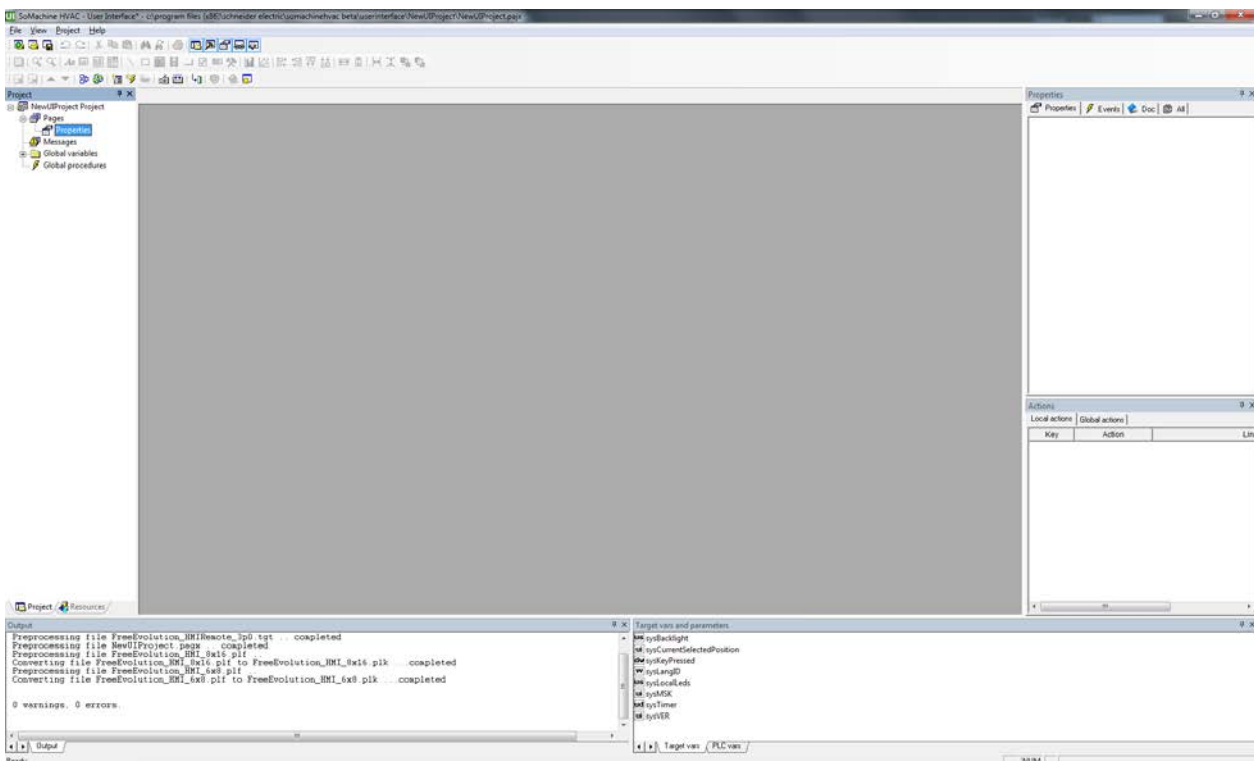
The dialog box shown below appears, in which you can create, name and save a new project:



New Project

New remote HMI

The new project appears as shown in the figure:



- A description of the icons is available in the **Online Help** (in English).
- The development environment is developed in this chapter.

The pages that follow explain how to create, develop and test a new menu, with simple examples of pages, templates and elementary programs.

12.2. CREATING PAGES

The created pages that will comprise the menu shown on the controller display will appear in the **Document bar** Panel. Below is a general description of the elements used for creating pages. The Appendix gives examples of menus that you can use to define a working menu.

Project Panel/Project TAB

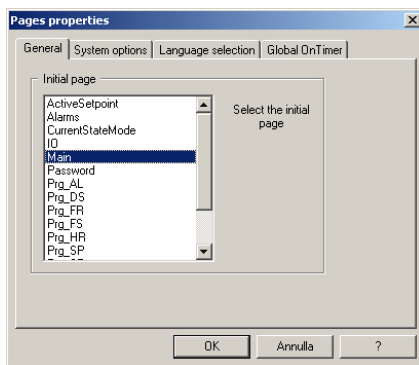
In this tab the following elements (nodes) can be used:

- **Pages** for managing pages (create, delete, import/export, etc.)
- **Messages** for managing message pages
- **Global variables** for managing “variables” common to the various pages
- **Global procedures** for managing “procedures” common to the various pages. **NOTE:** procedures are written in ST language only.

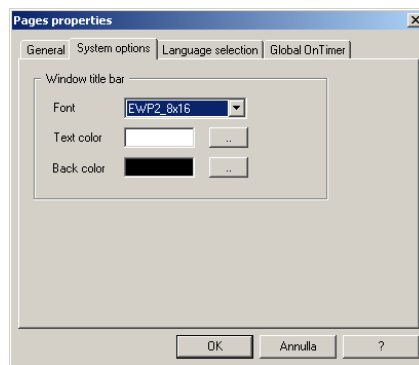
12.2.1. PAGES

Properties

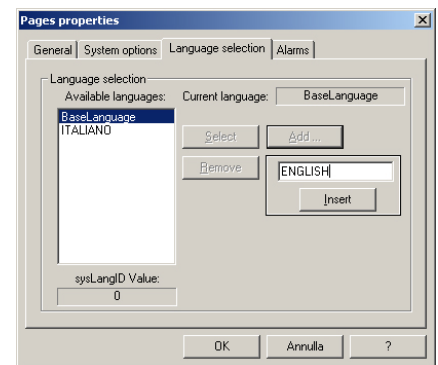
From the **Project** tab, **Pages** node, by right-clicking on **Properties** you can edit the common properties of pages, as indicated:



General



System options



Language selection

1. From the **General** tab you can select which page to set as the initial page
2. From the **System options** tab you can select the type of font, the text color and the background color (choice of black or white)
3. From the **Language selection** tab you can manage the language settings:
 - click **Select** to choose the current language (indicated in the **Current language** box),
 - click **Add** to add a new language
 - click **Delete** to delete a language
 - click **Export** to export the **.txt** text file to be translated into a text editor
 - click **Import** to import the translated **.txt** text file
4. from the tab **Global On timer** the developer can add specially created global “on timer” procedures.

Creating pages

Using the drop-down menu, select **Project > Objects** (insert page) or click on the respective icon **New Page**. A window will open where you can insert the name of the page.

The page name must not contain spaces (for example New Page returns an error; NewPage (with no spaces) or New_Page (with underscore) are acceptable).

The created page appears in the **Document bar** section (where it can be edited and changed).

NOTE: the page can be set as a **pop-up** page. Only one page is shown on the display at a time. A pop-up page is a page that is appears on top of another page.

Initial page

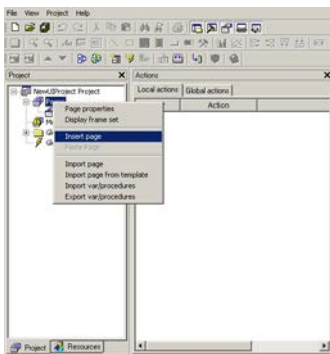
The property of “initial” page **must** be assigned.
It can be modified from the Pages section (node).

The initial page is identified by a “play” icon (red triangle). The initial page can be the page used for the “fundamental state display” for example the default page that appears when the controller is not in use. The fundamental state display can therefore be a splash image (for example logo or company name) or a page that displays values (for example temperature, pressure, setpoint etc).

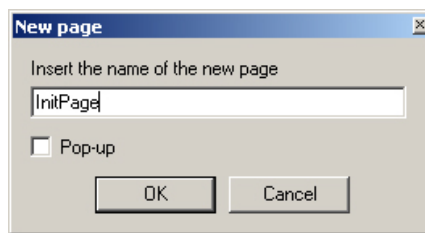
NOTE: if the initial page is not set, a compilation error is detected.

Managing pages

Click the right mouse button to rename, copy or delete a page, etc.



Insert page



New page

Messages

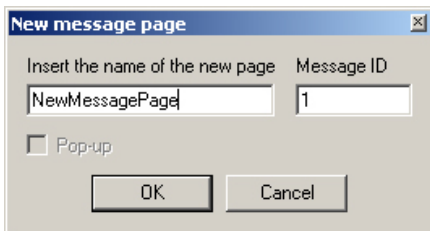
From the **Project** Panel, right-click the **Messages** node and select **Insert new message page** to create a new message page identified in the tree as a Warning triangle on a yellow background

Message pages are similar to other pages but each one bears a unique **ID**

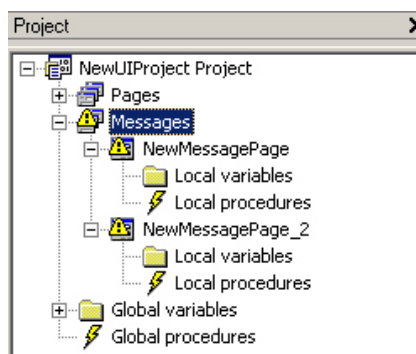
Below is an example of a “message” page call with syntax

Call from a **User Interface script**: dummyboolean:=Video_SendEvent(kWM_MSG, MessageID)

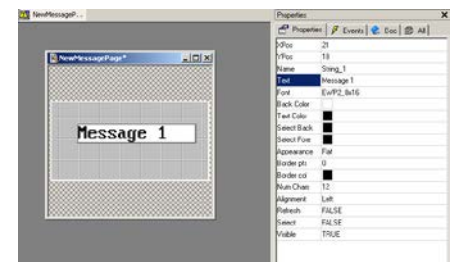
Call from an **Application** program: dummyboolean:=sysHmi_Message(MessageID).



Message page



example of message pages tree



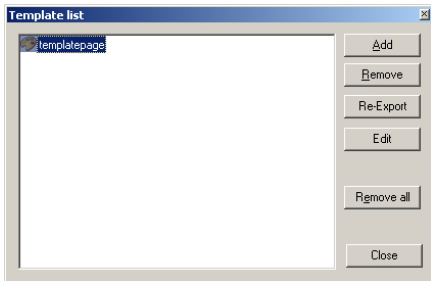
example of message 1 page

Template

From the **Project** Panel, by right-clicking on a page you can export the page (ending in **.pex**), or select **Export page as template** to save the page as a template page. (ending in **.petx**)

A template is a model or form that must be compiled based on a predefined structure

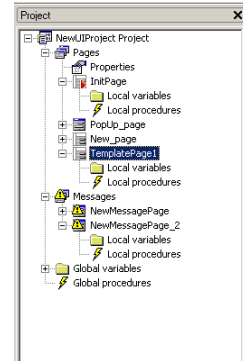
From the drop-down menu **Project > Template management** you can manage templates by adding **.petx** pages to the **Template List**. The added pages will also appear in the **Templates** Panel as a library and can be used by dragging and dropping them in the project, creating the desired template.



Template list



Template library



Instance of a template page

12.2.2. PROPERTIES PANEL

Every object that will be drawn on the page has properties that can be set from the **Properties** tab.

Depending on the object selected from the **Page bar** in the **Properties** tab, the elements required to manage the object on the page will be shown or hidden.

The complete list can be consulted in the **Online Help**. The most frequently used ones are listed here.

12.2.3. Properties TAB

Each individual page has properties listed as follows:

	Default	Description		Default	Description
CharDimX	6	Defines the page grid guides min(1,1) for positioning the objects	Title bar	NO	Inserts a title with a preset width
CharDimY	8		Page border	NO	Inserts a page border
Font	EWP2_8x16	character size	Caption	*	defines the content of the title bar
Back color	"white"	Background color	Appearance	Flat	defines the page border
Text color	"black"	text color			

Each time that these fields are modified, an asterisk (*) appears next to the name of the page. To save the change, use the RETURN key.

12.2.4. ACTIONS PANEL

This Panel enables you to define the “actions” assigned to the **M171P/M172P** keys for navigating the menu on the display. Actions are divided into:

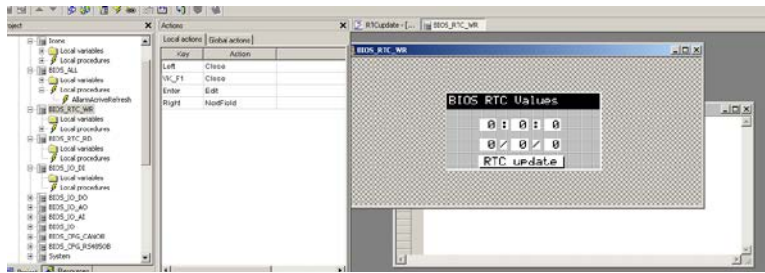
- **Local Actions:** actions that are specific to the individual page concerned
- **Global Actions:** actions that are common to all pages in the menu/project.

M171P/M172P has 5 keys: Right, Left, Up, Down and Enter. You can also associate an action to a long press of each of the 5 keys (identified by the prefix **Long**, for example **LongUp**).

In addition to the 4 keys there are 10 virtual keys identified with **VK_F1**, ..., **VKF10**

The virtual keys are associated to a **procedure** written in **ST** language.

For example, to set the time of the RTC clock, create the relative page:



Associate the page to a virtual key **VK_F1 (Close)** in Local Actions.

The procedure in **ST** language is as follows:

```
sysClock_update := TRUE;
uint_ret := Video_SendEvent(kWM_KEY, kKEY_VK_F1);
```

The functions predefined as VideoSentEvent, etc. (to be typed) are described in the SoMachine HVAC software, Programming Guide.

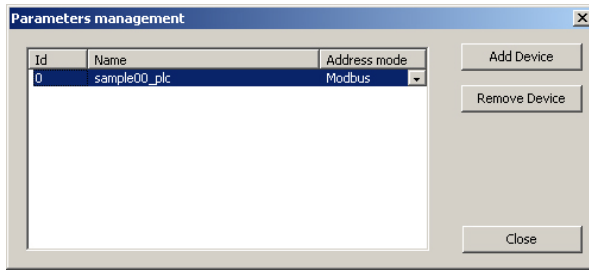
To insert an action, click the right mouse button and select **Insert action**. The selected action is highlighted yellow. Select **Remove action** to delete the action highlighted in yellow.

The following actions can be associated to each **key**:

Action column	Description	NOTES
Call	call of a procedure	to call a procedure write in the Link column the name of the procedure you want to call
OpenPage	opens a page	write in the Link column the name of the page you want to open
Close	closes the page	pair with the relevant key. NOTE: automatically returns to the start page
NextField	moves to the next value in a field	
PrevField	moves to the previous value in a field	
Edit	enables you to edit, for example write a value in a field	

12.2.5. APPLICATION PARAMETERS

From the menu **Project > Parameter management** you can link the parameters of an application developed in **Application**:



NOTE:

- Id=0 is reserved for **M171 Display Graphic** or links between a **UserInterface** project and an **Application** that is on the same hardware
- Id≠0 to be set in the event of a connection to an external device (i.e. displaying data from another device or from hardware other than the target on which the **UserInterface** is executed)

Click **Add Device** to import the desired parameters file (.parx). Modbus or TM171DMI (M171O) or Modbus and CAN (M171P/M172P) addressing can be associated to each set of parameters. The set of parameters will be shown in the appropriate tab of the **Target vars and parameters Panel**.

12.2.6. PROPERTIES TAB/VARIABLE

To use the variable, drag and drop it from the imported tab into the page:

- The properties of the variable will be identified in the **Properties/Variable** tab with the “@ syntax”
- The variable will be identified with **@file name.variable name**
- To modify it select the **Variable** field, move onto the box and click on the ellipsis (...)
- The **Variable parameter selection** window appears, which shows the available resources that can be associated to the box.

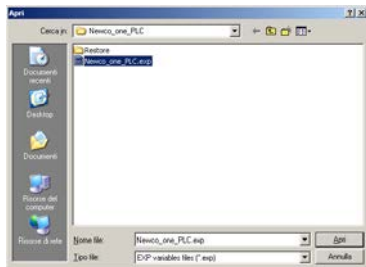
NOTE: in the same way you can create a “progress” box and associate a variable. In this case, the status / graphic progress bar of the variable is shown instead of its value.

PLC VARIABLES

NOTE: Valid only for projects with M171P/M172P target.

The remote HMI target only supports the @ syntax for the use of parameters/status variables of the related Application project.

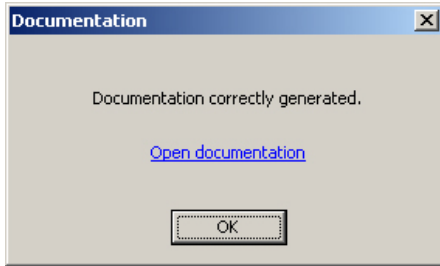
From the menu **Project > Link PLC variables file** you can link the variables of a **PLC application** developed in **Application**:



Import the desired “EXP variable” file (.exp). The set of parameters will be shown in the relative **PLC vars** tab in the **Target vars and parameters Panel**.

GENERATING DOCUMENTATION

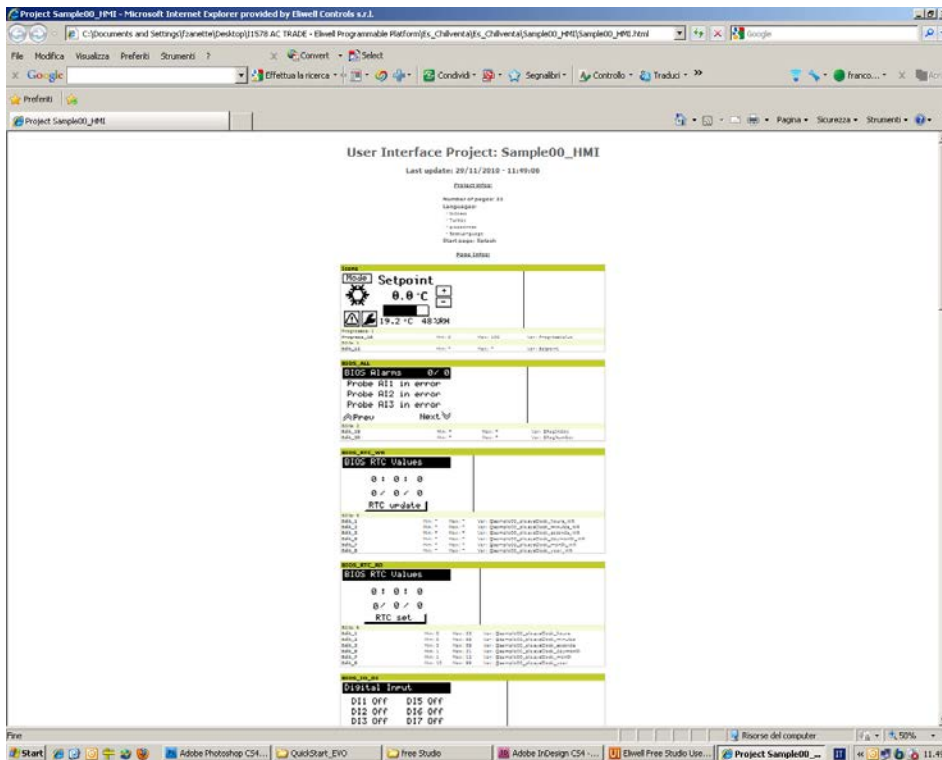
Using the icon **Generate Doc.** you can create documentation linked to the **Doc** fields (**Properties Panel, Doc tab**):



Through a **browser** you can view the generated documentation by clicking on the link Open documentation. the documentation gives information relating to the project:

- number of pages
- program languages
- initial page (name of initial page)
- visual list of pages (as they are presented on the display).

Example of documentation generated using the icon **Generate Doc**:



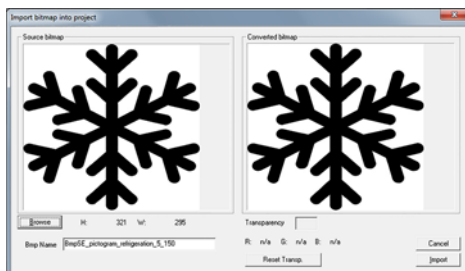
12.2.7. PROJECT PANEL / RESOURCES TAB

In this tab the following elements (nodes) can be used:

- **Fonts** for managing pages (create, delete, import/export etc.)
- **Bitmaps** for managing images to be inserted in the pages
- **String table** for managing strings to be used in the various pages
- **Enumeratives** for managing enumerable sets to be used in the various pages
- **Image lists**
- **Sets** for managing sets of strings / parameters to be used in the various pages.

Bitmaps

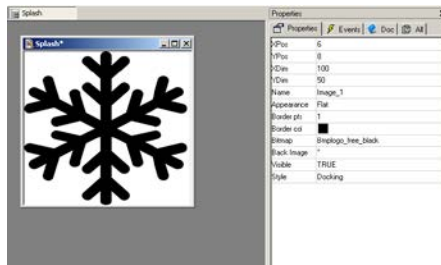
UserInterface can be used to import images (specifically **bitmaps**, such as icons, logos etc.) to be used in the menu pages. Click on the node **Bitmaps > Import bitmap**. The corresponding window appears for importing the image. Click the **browse** button to select the image and the **Import** button to import it. Example of import of **sample** logo:



Import black/white images of suitable size. The display can support an image of approximately 100x50px. The imported bitmap will be in the **Bitmaps** node.

To insert the bitmap in a page:

- insert an “image” box (menu Page > New Image or use the corresponding icon)
- define the dimensions of the image from the **Properties** Panel using **XDim** and **YDim** (for example 100x50px)
- from the **Properties** Panel select the bitmap from the field of the same name.



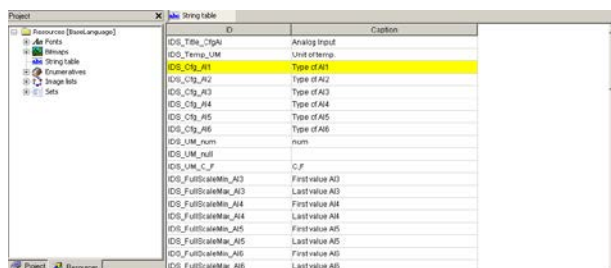
String table

UserInterface allows you to define strings that can be used when creating pages.

Using the menu **String>Insert** or its respective icon **Insert Record** you can create an association between a “string” and its corresponding description.

In the **ID** field the string is defined as **IDS_stringname**. Use this prefix to prevent the descriptions that appear on the display being confused with the transcoding in **UserInterface**

In the **Caption** field enter a description of the string.



By selecting the page from the menu **Project/Project** you can associate the desired string to a text field using the menu **Properties>Caption**. The text that will appear on the display will be the text inserted in the description.

Enumeratives

UserInterface also allows you to define strings by means of enumerable sets that can be used when creating pages. An enumerable set is an array of values that are not necessarily consecutive. Each value is associated to a description that will be shown on the **M171P/M172P** display.

Move onto the **Enumeratives** node and use the menu **Record>Insert** or the respective **Insert Record** icon to create the sets that you want to use:

The default name is Enum1, Enum2, etc. Select the box to change the name. Each set will be shown as a subnode of Enumeratives.

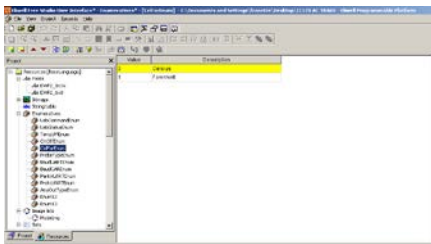
Move onto the corresponding node and use the menu **Record>Insert** or the respective **Insert Record** icon to enter the array values.

After creating (or defining) the enumerative from the **Resources** tab, open the **Project** tab and, selecting the page / element of the desired page, associate the corresponding **Enum** to the Format field.

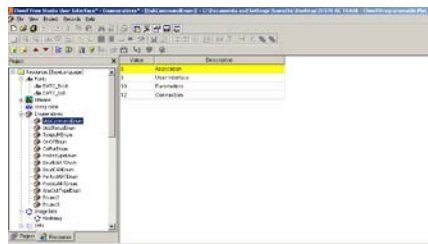
The three examples show

- **definition** of an enumerative **CelFarEnum** consisting of 2 values for defining the degrees of temperature (0=Celsius, 1=Fahrenheit).
- **definition** of an enumerative consisting of 4 non-consecutive values for defining USB commands (values 8,9,10,12)
- use of **CelFarEnum** on a page.

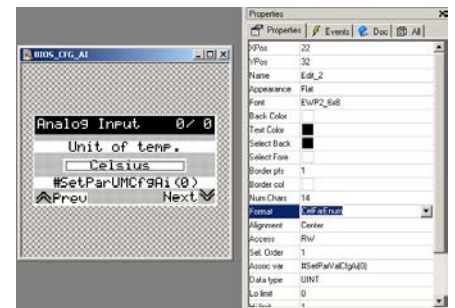
The order of the elements can be changed with the icons **Move Up ▲/ Move Down ▼**



definition of CelFarEnum



Enum with non-consecutive values



use of CelFarEnum

Image Lists

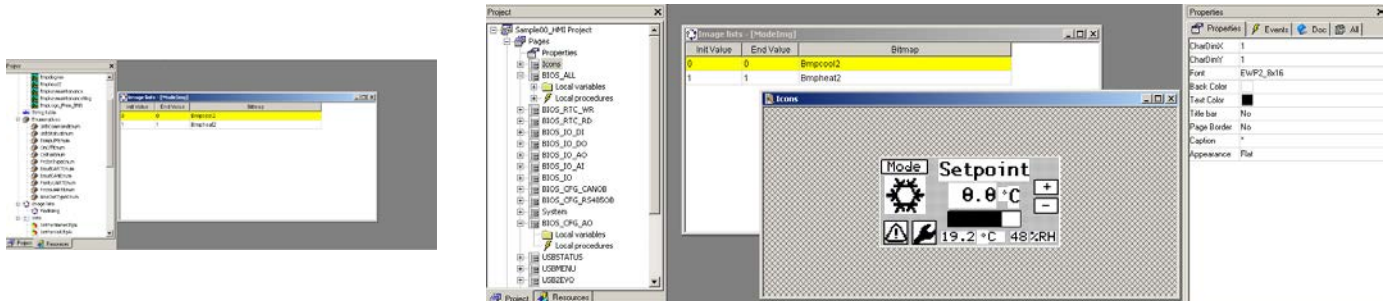
Similarly to **Enum** it is possible to create lists of images associated to ranges of values. The image shown on the display will change depending on the state of the variable associated to the list.

A list is created and a **bitmap** (see corresponding node) is associated to each image, indicating its reference range. After creating (or defining) the list from the **Resources** tab, open the **Project** tab and, selecting the page/element (created with **"New animation"**) of the desired page, associate the bitmap list to the Imagelist field.

The three examples show:

- **definition** of a **Modeimg** list of 2 bitmaps for displaying Cool mode and Heat mode. The range of values is reduced to one value in this case (0 for Cool, 1 for Heat)
- use of **Modeimg** on an “Icons” page, associating the **Modeimg** list with the animation. A state variable must be associated, depending on which the image will change.

However a static bitmap must be inserted as an image via **Page > New Image** or using the respective icon.



definition of Modeimg

use of Modeimg

Sets

Often there is the need to create a group of very similar pages (for example a scroll menu for showing a set of parameters in the same folder)

UserInterface allows you to define **Sets** that can be used when creating this type of page, without having to replicate the page **n** times.

A set is an array of strings or variables/parameters/alarms

Move onto the **Sets** node and use the menu **Record>Insert** or the respective icon **Insert Record** to create the sets that you want to use. The type of set can be:

- **String** for example a string defined in a **string table**
- **Variante** for example a variable/parameter

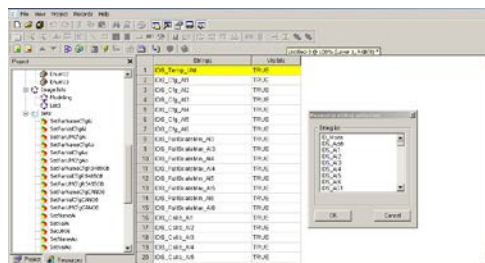
Each set will be shown as a subnode of **Sets**.

Move onto the corresponding node and use the menu **Record>Insert** or the respective icon **Insert Record** to enter new items in the array.

Strings

In the case of a **String** type **Set** the string is selected by moving onto the box and clicking on the ellipsis (...)

The **Resource string selection** window appears, which shows the available strings, previously defined in the string table, that can be associated to the array. The visibility (**Visible** column) for strings is TRUE - not modifiable.

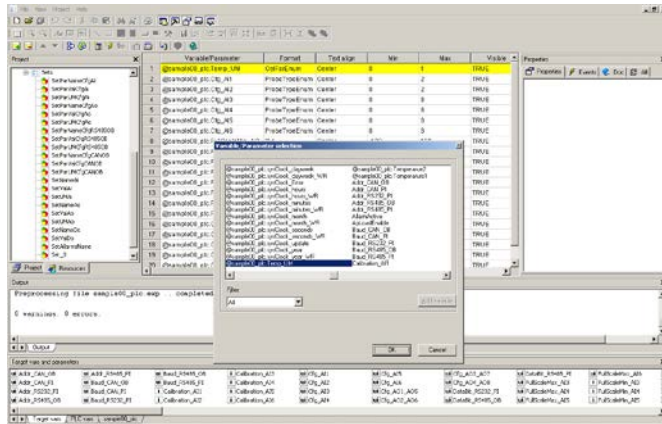


Variante

In the case of a **Variante** type **Set**, the variable is selected in the same way by moving onto the column **Variable/Parameter**.

In this case the term “variable” refers to genuine variables, controller parameters, BIOS, alarms, etc.

The **Variable/parameter selection** window appears which shows the available variables that can be associated to the array.



Using the Filter option you can filter the display of the variables/parameters:

- The **Format** column shows the default U.M. of the parameter. The user can manually associate an **Enum**
- Similarly the Min Max columns show the default range of the parameter, which can be modified by the user
- The **Text align** column aligns the string to the left, right or center.

for example for a parameter that defines a COOL, HEAT and AUTO machine mode it is possible to hide one or more values at UI level even if at controller (or application level) there are 3 usage modes.

After creating the Set from the **Resources** tab, open the **Project** tab:

- Previously a page was created that you want to replicate using **Sets** (for example in order to manage the analog outputs **BIOS_CFG_AO**)
- Selecting the appropriate field, you need to associate the corresponding Set to the field **Assoc_var** using the prefix **#**, and with the suffix **(0)** to indicate the first element of the array. **NOTE**: The field must be of the **Edit** type (for example in order to manage the analog outputs **#SetParValCfgA0(0)**).

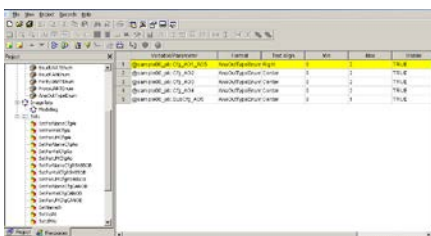
The three examples show:

- **definition** of a Variant Set **SetParValCfgA0** of size 5 to define array of output configuration parameters
- **definition** of an enumerative **AnaOutTypeEnum** of 3 values (voltage, current, ON/OFF)
- **use** of **#SetParValCfgA0(0)** on a page (which results in the production of 5 pages that differ in terms of the configuration of parameter **SetParValCfgA0(1),..., SetParValCfgA0(5)**)
- **NOTE** that the Edit field "IO Modulated" is associated by means of the Format field to the Enumerative **AnaOutTypeEnum**

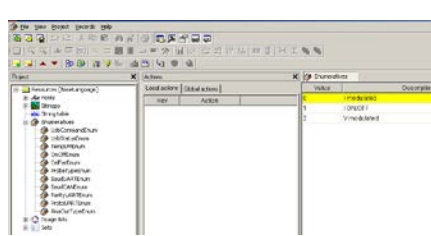
NOTE that the array starts from the value **(0)** whereas when defining Sets the first value is **1**, therefore **Set(1)** corresponds to **#Set(0)**.

NOTE that in the example there is another Edit field associated to a **Set**.

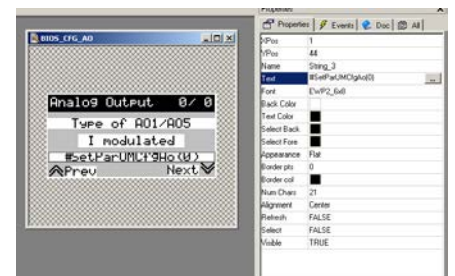
It is possible to use multiple **Sets** on the same page. Define **Sets** of the same size if used on the same page.



SetParValCfgA0



AnaOutTypeEnum



use of #SetParValCfgA0(0)

13. CONNECTION PANEL DEFINITIONS

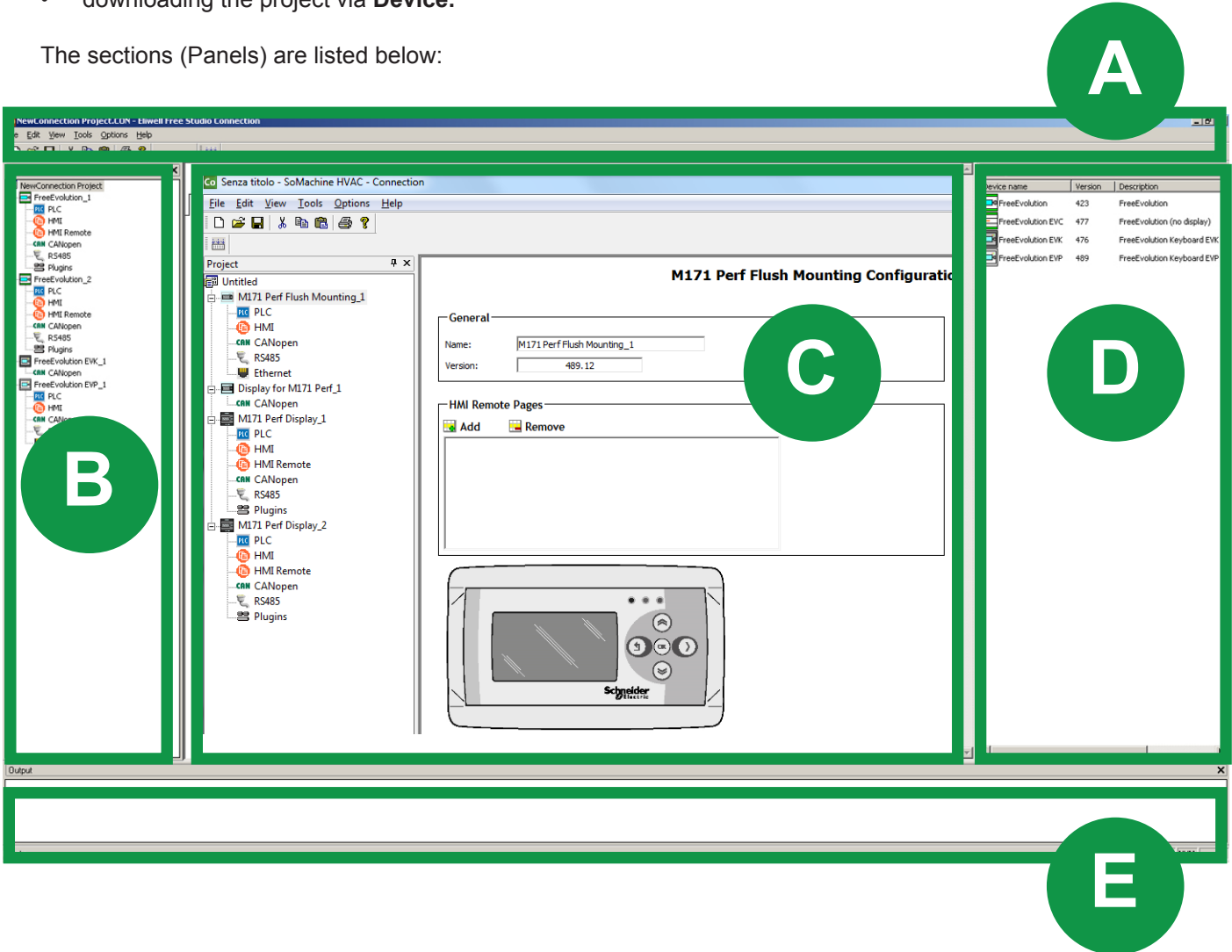
Connection is the entry point for starting to develop projects.

The **Connection** workspace defines the hardware structure of an **M171P/M172P** project and manages the network and data exchange protocols between **M171P/M172P** (binding).

The **Connection** workspace dedicated to developers has various sections for creating and managing **M171P/M172P** devices, as well as for compiling the project, for example:


- creating the **PLC** application using **Application**
- defining the graphical interface of **M171P/M172P** using **UserInterface**
- compiling the **Build project**
- downloading the project via **Device**.

The sections (Panels) are listed below:



Section	Panel Name / Section Application	Panel	Description
A	/	Tool bar (Main bar)	This section shows the work tools in the form of icons. Some tools can be selected by means of key combinations (shortcuts).
B	Project	Project (Workspace)	The Project Workspace can be used to manage the project and one or more M171P/M172P (1) devices: <ul style="list-style-type: none"> • from the project you can manage the device network (Network list) • The various protocols are configured in each device.

Section	Panel Name / Section Application	Panel	Description
C	/	Configuration Panel (Edit)	<p>This section contains:</p> <ul style="list-style-type: none"> • General project management tab • Network list tab <p>Depends on the selected M171P/M172P device:</p> <ul style="list-style-type: none"> • General M171P/M172P configuration tab <p>Inside the device (for example M171P/M172P Display):</p> <ul style="list-style-type: none"> • PLC for configuration of Controller application developed with Application • HMI for configuration of User Interface menu • CAN for configuration of corresponding protocol • RS485 for configuration of corresponding protocol • Plugins for configuration of protocols using communication modules
D	Catalog	Device catalog	<p>This Panel shows the devices available from the catalog.</p> <p>NOTE: Dynamic visibility of devices based on selections (for example communication modules).</p>
E	Output	Messages	<p>This Panel shows the messages relating to the development of the project (file opening, reading/writing errors, status of connection to device, etc).</p> <p>NOTE: the connection to the target device is also visible in the bottom right of the screen by means of colored messages (see Application Download).</p>

(1) **Connection** manages one or more **M171P (Display, Blind, Display Graphic), M172P (Display, Blind) and M171 Flush** devices.
M171O can be inserted in the network as a **slave** device via **RS485**.
The **M171 Performance Expansion** module can be inserted in the network as a **slave** device via **RS485 or CAN**.
The display of the Panels can be set from the drop-down menu **View > Panel name** and the Panel can be organized as the user wants in terms of size and layout.
SoMachine HVAC saves the last settings made.
The Panel **View > Compiler toolbar**  is an icon that is visible or hidden in the **Main** menu.

Configuration Panel (Edit)

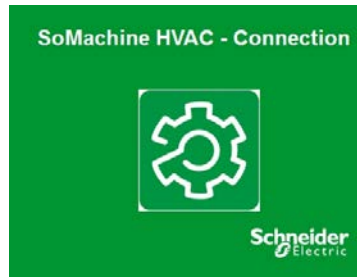
The screens below show the various default **devices/nodes** available for the various **targets**. The devices for **M171P/M172P Display** are indicated in the table above in section **C**.
M171P/M172P Display - Blind / M171 Flush and M171 Display Graphic are **devices** available as “**root**” nodes of the project.

14. TO START CONNECTION

14.1. FIRST START-UP

1. Click on the **Launch SoMachine HVAC** icon on your Desktop
2. Click on the **SoMachine HVAC Connection** icon.

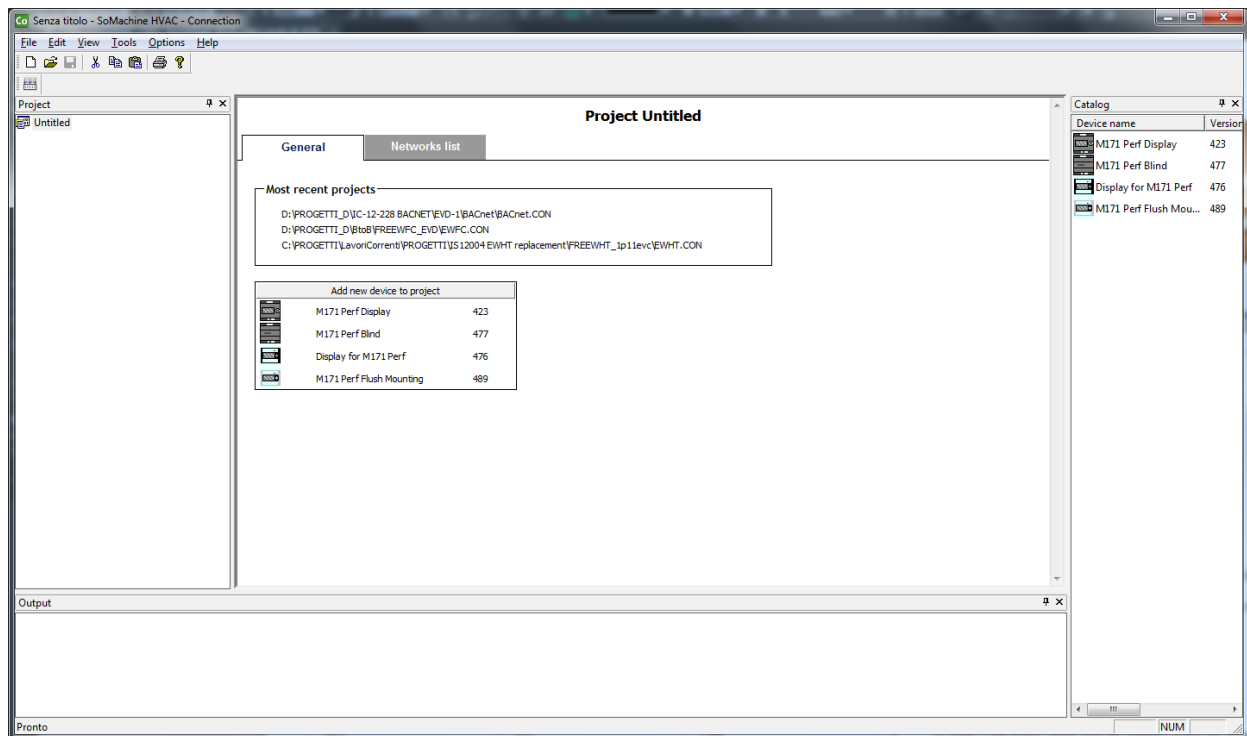
Nome	Ultima modifica	Tipo	Dimensione
SoMachine HVAC - Application	17/07/2014 08:46	Collegamento	2 KB
SoMachine HVAC - Connection	17/07/2014 08:46	Collegamento	2 KB
SoMachine HVAC - Device	17/07/2014 08:46	Collegamento	2 KB
SoMachine HVAC - Simulation	17/07/2014 08:46	Collegamento	2 KB
SoMachine HVAC - User Interface	17/07/2014 08:46	Collegamento	2 KB



1

2

The program appears as shown in the picture:



You can now:

- Add a new **device** to the project
- Add/remove a network from the **Network list** .

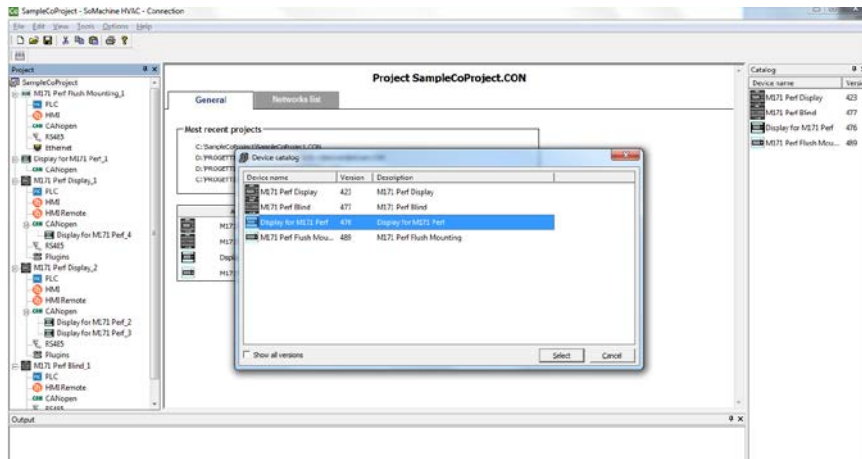
Existing projects have the extension **.con**.

14.1.1. CREATING A NEW PROJECT (WORKSPACE)

To create a new project, save the Untitled project and rename it with a name of your choice (**CTRL+S**).

To add a device to the project, either click on **Add new device to the project** or **drag and drop** the **M171P/M172P** icon from the **Catalog** Panel on to the project name.

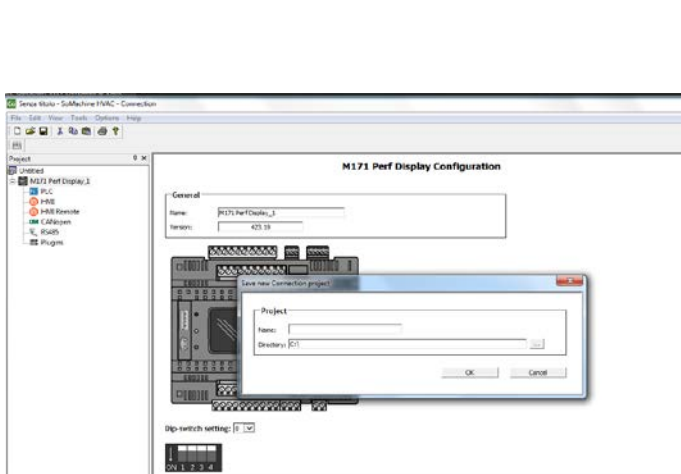
example of Add (Device)



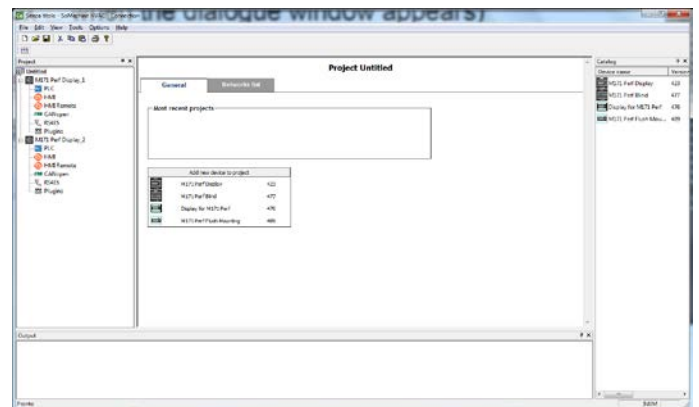
You can rename each **M171P/M172P** device with an appropriate name (the default name is **M171P Display_1**, **M171 Display_2**, etc. / **M172P_1**, **M172P_2** etc).

In the workspace you can see the name of the device and the **version**).

There may be more than one M171P/M172P in Catalog with different versions (to view the different versions, go to menu Options > Show all versions in catalog or tick the tick box when the dialog window appears).

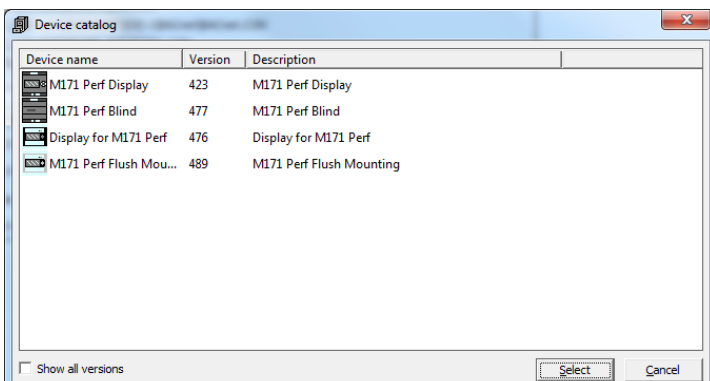


creating a new project CTRL+S

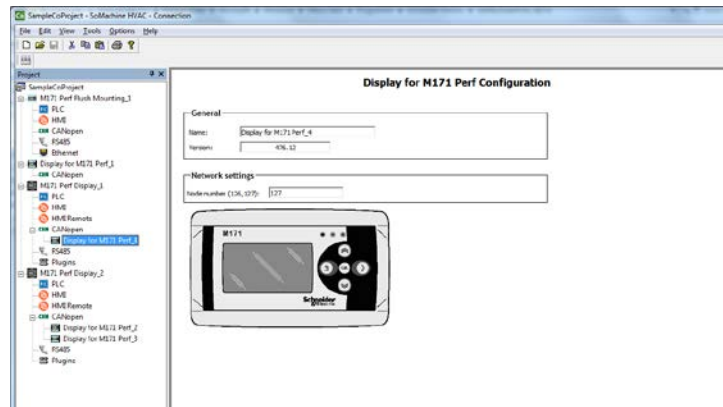


adding M171P to the project
NOTE example of adding 2xM171P devices

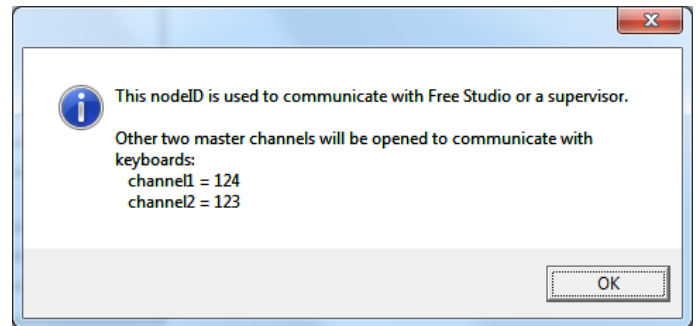
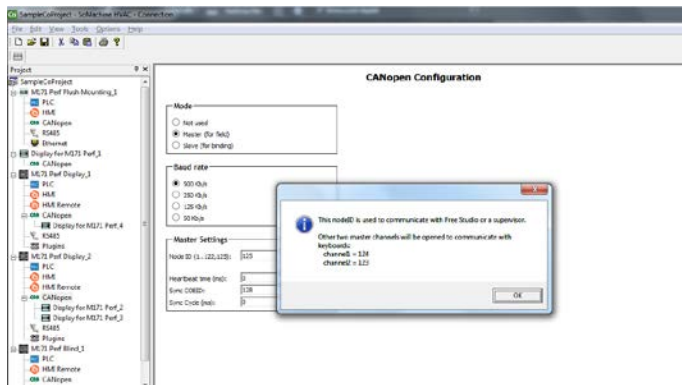
Device catalog



example: M171P + 2 x M171 Display Graphic



The two M171P Display Graphic displays are both connected to the CAN node:



**CAN node virtual channels
channel1:124; channel2: 123**

**virtual channels
channel1:124; channel2: 123**

The CAN node has two default virtual channels that can be connected to a maximum of 2 x M171 Display Graphic. The default virtual channels are 124 for the first display and 123 for the second M171 Display Graphic. Click on ? from the CAN node to view the values.

The default address of the **Display for M171 Perf_2** display is 127, the default virtual channel 124 and the default CAN baudrate 500Kb/s.

Thus, when physically connecting an **M171 Display Graphic** to an M171P with the default settings, upload HMI from the **M171 Display Graphic** BIOS menu.

In other cases, such as for **Display for M171 Perf_1** (which has the address 126), set the address 126 and the virtual canal 123 from the **M171 Display Graphic** BIOS menu.

14.1.2. PROJECT PANEL

Each **M171P** device has the following resources, which are shown as nodes of the **target**:

target	workspace	description	what to do	Notes
M171P/ M172P	PLC	PLC project	move onto the node and click the right mouse button to open Application	see Getting Started (Application)
M171P Flush				
M171P/ M172P	HMI	menu for creating pages	move onto the node and click the right mouse button to open User Interface	see Getting Started (User Interface)
M171P Flush M171				
Display Graphic				
M171P	Remote HMI	menu for creating pages	move onto the node and click the right mouse button to open User Interface	see Getting Started (User Interface)

target	device	description	what to do	Notes
M171P/ M172P	CAN (1)	on-board M171P serial	select the Mode and add the device from the catalog	always available
M171P Flush				
M171 Display Graphic				
M171P/ M172P	RS485 (1)	on-board M171P serial references and expansion modules	select the Mode and add the device from the catalog	check availability
M171P Flush				
M172P	Ethernet			check availability
M171P	Communication modules (2)	optional modules available separately		check availability
(1) also available as serials on communication module (2) for example Ethernet communication module				

You can create both **PLC** and **HMI** projects from scratch or import existing projects and projects from the catalog.

PLC project

From the Panel **Project** > **PLC node** click the right mouse button and select Open with **SoMachine HVAC Application**

By default you will be prompted to save the project **SoMachineM171P_n_PLC**

See **Application** for information on development. In short:

- develop the PLC code with **Application**
- create parameters and state variables of the PLC application
- compile and download application onto **M171P** and **debug** if necessary
- close **Application** and return to **Connection**.

The saved project **.ppjs** is located in the **Connection** folder / workspace.

HMI project

From the Panel **Project** > **HMI node** click the right mouse button and select Open with **SoMachine HVAC UserInterface**

By default you will be prompted to save the project **SoMachineM171P_n_HMI**

See **User Interface** for information on development. In short:

- develop the **HMI** menu or pages with **UserInterface**
- the parameters and state variables of the previously created **PLC** application are linked to the menu and will be visible in section **H** of **UserInterface Target vars & parameters**
- check the connection mode, appropriately configuring the connection parameters
- compile the **HMI** pages and download them onto **M171P**
- close **UserInterface** and return to **Connection**.

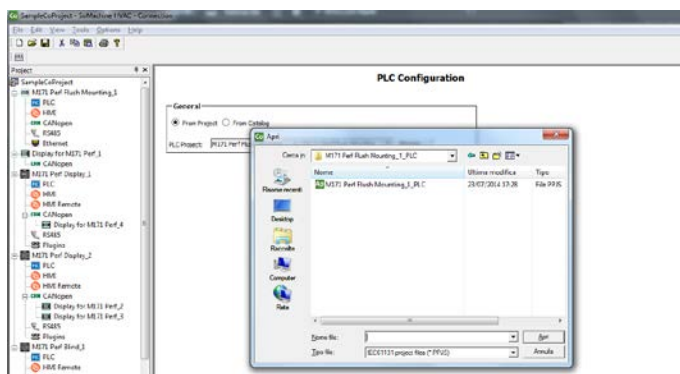
The saved project **.pajx** is located in the **Connection** folder / workspace.

The device will be added to a list (highlighted in yellow)

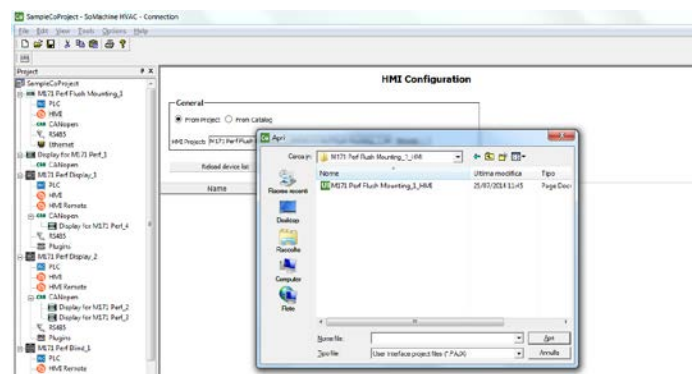
Press the **Reload Device List** button to populate the table with the list of devices connected to the **HMI** project and to use the pages remotely if so required.

Remote HMI project

Same as HMI Project.



PLC Configuration
adding an existing project



HMI configuration
adding an existing project and reloading devices list

Download project

Use the menu **Tools>Build** or **F7** to create the **.CFN** configuration file.

From the Panel **Tools > Open with SoMachine HVAC Device**, open **Device**.

After configuring the device, for example the **BIOS** communication parameters, you will be able to download:

- **PLC** application
 - **HMI** pages
 - configuration file **Cfg files**
- using the **Download All** button.

NOTE: each of the 3 files can be downloaded independently from the other two by moving the mouse pointer onto the respective icon, clicking the right mouse button and running the corresponding **download**.

PROTOCOL MANAGEMENT

Each protocol can be used in different ways. If **Mode** is **Not used** the corresponding device is not used.

target	CAN		RS485	Ethernet	Profibus
M171P Flush	Master (for field)	Master (for HMI remoting and binding)	Modbus Master (for HMI remoting and field)	Enable Modbus Master (for HMI remoting and binding)	-
M171P M171P Flush	Master (for field)	Slave (for binding)	Modbus Master (for field)	-	DP Slave-V0
M171P Display Graphic	Master (for HMI remoting)	-	-	-	-

The catalog will show the devices that can be added (added using **drag and drop**) to the corresponding protocol. In the case of **Slave** and **Ethernet** the “**Binding**” device is added. To “Add a source device” the **network** must be set. The maximum number of devices that can be added to the individual node is checked from **SoMachine HVAC** and indicated by means of the relative message. (For example max **2 M171 Display Graphic** on **CAN**). On the RS485 protocol you can also connect generic Modbus devices.

15. SIMULATION PANEL DEFINITIONS

The **Simulation** workspace dedicated to the developer has various sections for simultaneously executing applications and pages (**M171P**) in a simulation environment without needing to have a **target** connected to the PC.

Simulation can simulate the execution of:


- **PLC** application created using **Application**
- Graphic **HMI** interface of **M171P** created using **UserInterface**.

Execution can therefore also occur on the same PC used for development.

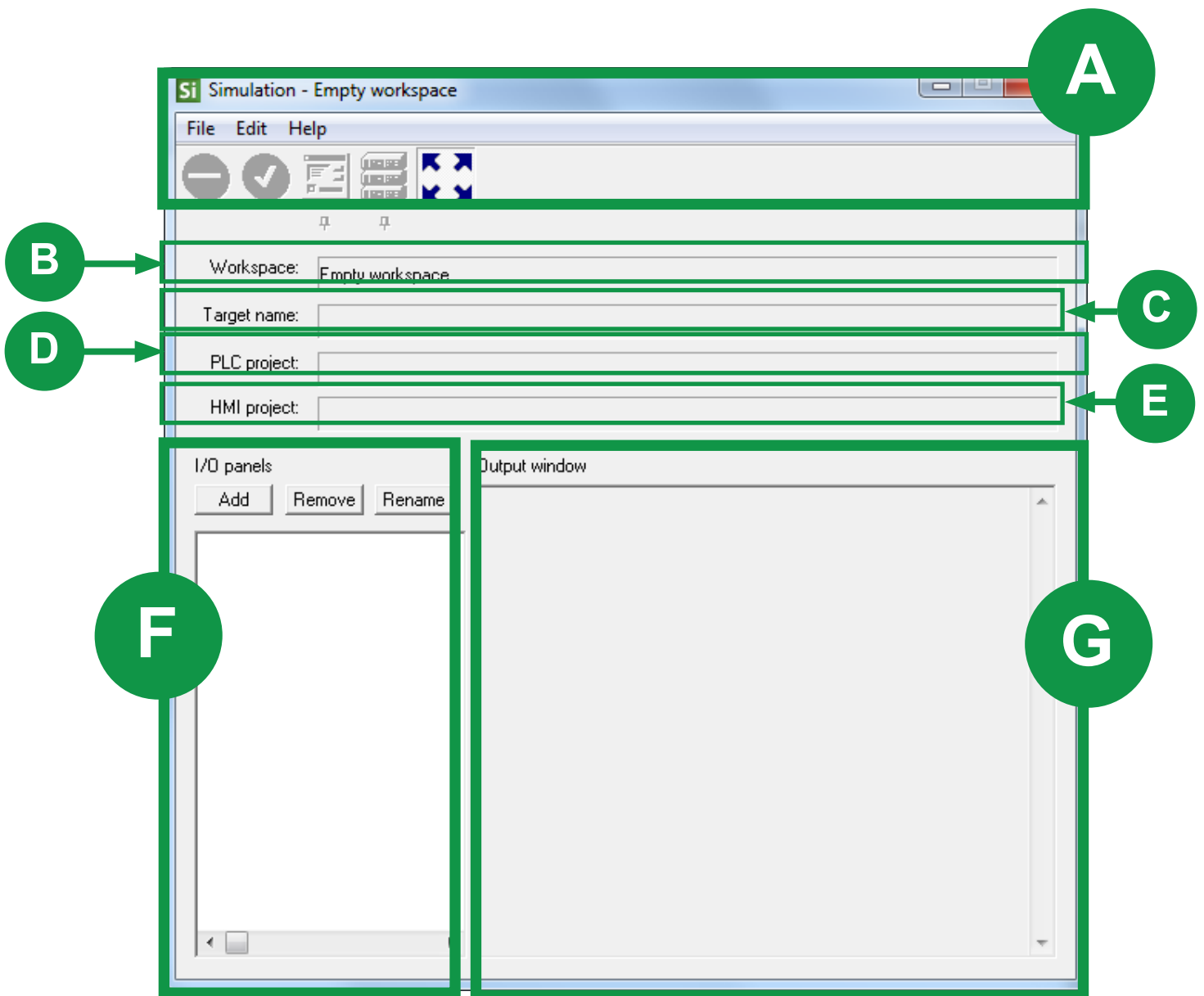
The controller application can therefore be “downloaded” onto the Simulator rather than onto the physical target.




NOTE: You can start **Simulation** from **Application**.

Below is the list of sections (Panels) as it appears on the first screen (empty workspace) compared with other workspaces (Application, UserInterface etc.) it is not essential to interact with the Simulation window.

When not in use it can be reduced with the  key.

Here you need to manually connect the controller application and/or **HMI** pages (M171P only) to the Simulator.

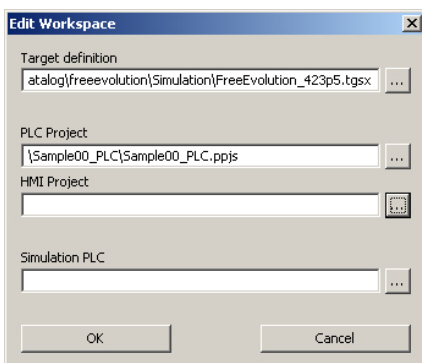


Section	Panel Name / Section Simulation	Panel	Description
A	/	Tool bar (Main bar)	<p>This section shows the work tools in the form of icons.</p> <p>✓ to start controller Application ⊖ to stop controller Application</p> <p> Show HMI window to show/hide HMI (virtual controller)</p> <p> Show I/O Panels to show/hide the I/O Panels</p> <p> Show full control window to reduce/expand the window</p>
B	Workspace	Workspace	<p>file with the preselected name that functions as a workspace</p> <p>You can create multiple workspaces with different applications (M171O+M171P) and/or HMI pages (M171P)</p>
C	Target name	Target name	<p>Not modifiable by the user. Examples:</p> <ul style="list-style-type: none"> • M171O 412.5 for M171O target • M171P423.5 for M171P target <p>NOTE: check from Application >Project > Select target that the version number in the target name is equal to or higher than 5 (for example M171P423.5).</p>
D	PLC Project	PLC application name	if field is blank, from Edit > Edit workspace , click on the ellipsis (...) to select the corresponding file
E	HMI Project	name of HMI pages	
F	I/O Panels	I/O Panels	I/O window management
G	Output window	messages	-

Edit Workspace

Manual editing. There is no need to “edit” the workspace when opening Simulation from Application from **Edit > Edit workspace**, click on the ellipsis (...) to select:

- **Target name:** file containing the information needed to reconstruct the runtime of the physical target **.TGSX file**.
.TGSX file is created automatically by Application in the folder SoMachine HVAC > Catalog > M171P > Simulation
- **PLC Project:** controller application **.PPJS / .PPJX file**
- **HMI Project:** HMI pages **.PAJX file**.



controller application
already present



select HMI pages
.PAJX

16. SIMULATION

The simulation is not intended as a substitute for real, empirical testing during commissioning. It is a means for the programmer to submit his application, or parts of application, to unit testing and verification. Only empirical testing with live equipment in the complete application can be considered a valid mechanism for validation.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Always empirically test your application during commissioning before placing your application and associated equipment into service.

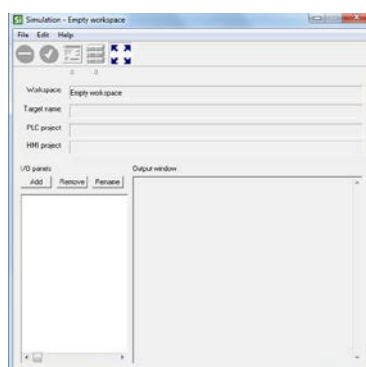
Failure to follow these instructions can result in death, serious injury, or equipment damage.

16.1. FIRST START-UP

1. Click on the **Launch SoMachine HVAC** icon on your Desktop
2. Click on the **SoMachine HVAC Simulation** icon.

Nome	Ultima modifica	Tipo	Dimensione
SoMachine HVAC - Application	17/07/2014 09:46	Collegamento	2 KB
SoMachine HVAC - Connection	17/07/2014 09:46	Collegamento	2 KB
SoMachine HVAC - Device	17/07/2014 09:46	Collegamento	2 KB
SoMachine HVAC - Simulation	17/07/2014 09:46	Collegamento	2 KB
SoMachine HVAC - User Interface	17/07/2014 09:46	Collegamento	2 KB

1



2

The (empty) workspace appears as shown in figure 2

NOTE: You can open Simulation directly from Application or UserInterface. Click on the corresponding **Simulation Mode** icon. It is a toggle function. Clicking on the Simulation mode icon opens / closes the Simulation workspace.

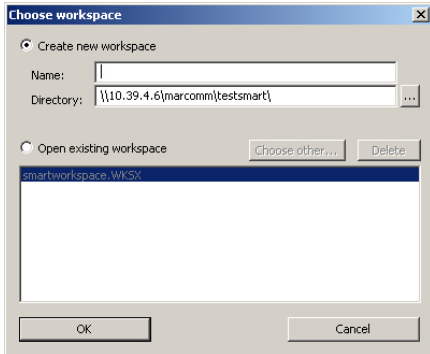
You can now:

- Create a new project (**workspace**) **.WKSX** “New ...”
- Open an existing project “**Open ...**” **CTRL+O**
- Open recent projects “**Recent file ...**” by selecting from a list.

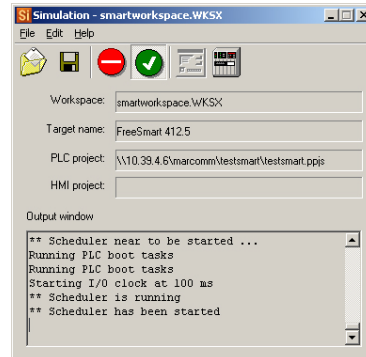
Simulation projects (**workspace**) have the extension **.WKSX**.

16.2. OPENING SIMULATION FROM APPLICATION

You can open Simulation directly from Application. Available for M171O and M171P/M172P. Alternatively, from the **Application** workspace click on the corresponding **Simulation Mode** icon toggle function. Clicking on the Simulation icon opens / closes the Simulation workspace:



1



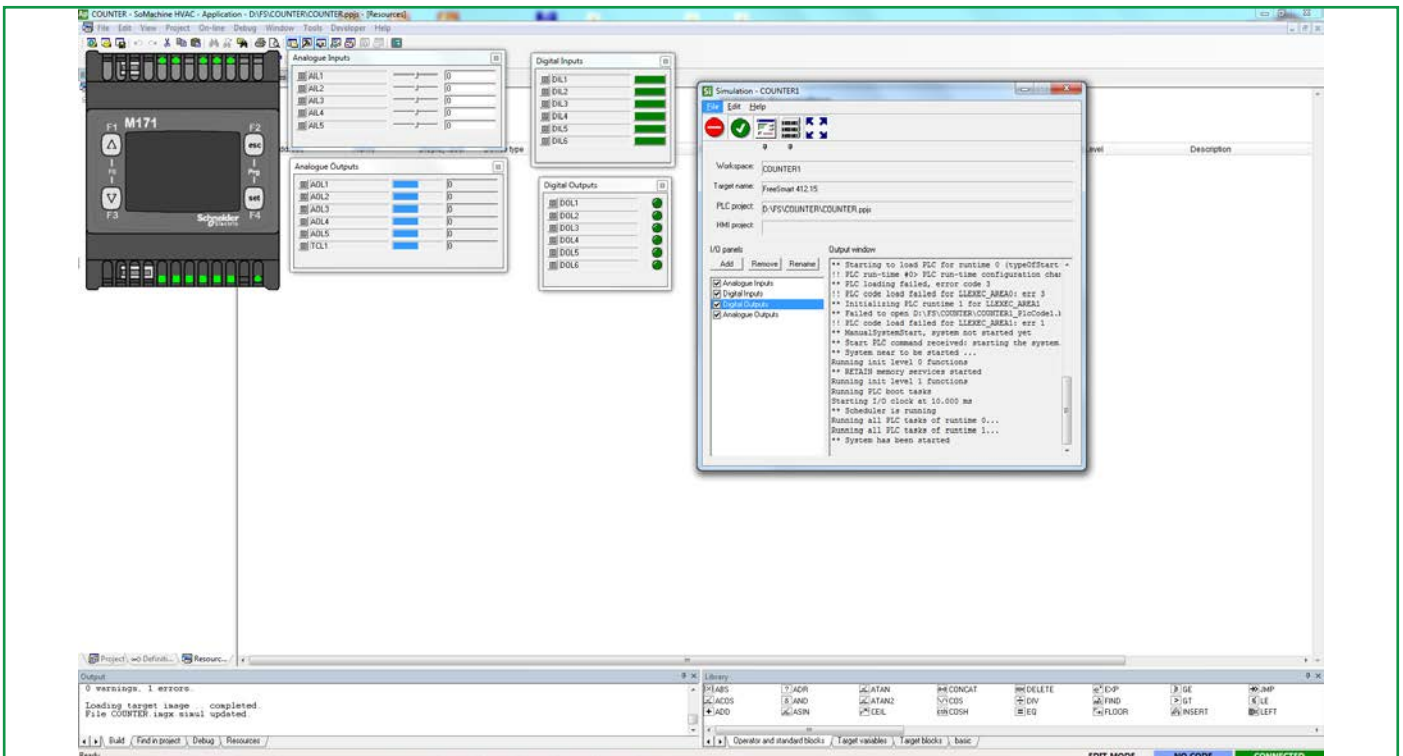
2

1. A dialog window opens from where you can:
 - Create a new **workspace**
 - Open an existing **workspace (.WKSX)**
2. Click OK. The following will appear:
 - the **Simulation Panel** section (if Simulation does not start you need to change the target. See Troubleshooting)
 - TCP/IP connection (reduced to an icon on the main bar): local connection between the **Application** and **Simulation** workspaces.


From **Application** click on **Download code** to “download” the Controller application to the Simulator. In **Panel A** there are two icons for viewing the corresponding Panels

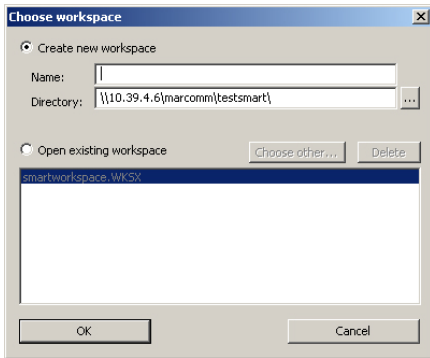
- **Show IO Windows** Panels that indicate the status of the I/O
- **Show HMI Window** graphic and interactive representation of the controller.

example for an M171O target

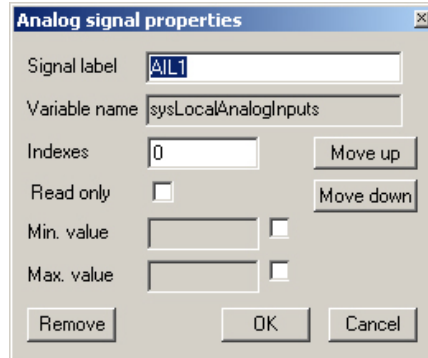


IO Panels

Clicking on the corresponding icon  brings up various windows/Panels corresponding to the inputs/outputs.



I/O Panels



Analog signal properties

The **I/O Panels** window is an “index” window that lists the windows (visible if the tick box is selected).

From this window you can add a window or delete/rename the listed windows.

To add a variable, drag and drop the element you want to monitor from **Application** to one or more windows.


Analog inputs and outputs can be simulated (for digital I/O see HMI Window):

- For inputs, a slider is available for modifying the read value.
- For outputs, a status / progress bar is available. For digital outputs, green if the output is not active, red if active.

The other elements that can be simulated are (in general each object that has a Modbus address) are:

- **Global shared**
- **Target variables.**

NOTE: Also includes BIOS parameters and EEPROM parameters. M171P/M172P only.

Each element on the various Panels has properties that can be edited using icon 

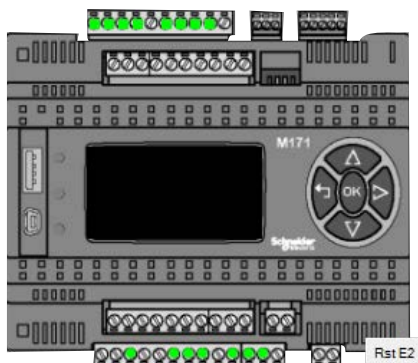
Signal Properties

From this window you can view and, for some, modify:

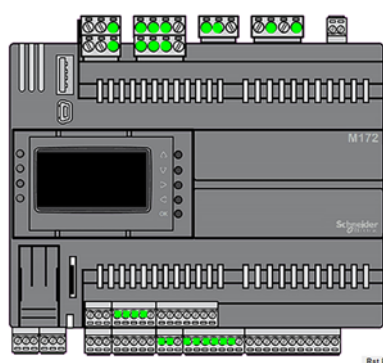
- **Signal label:** label of the variable/resource
- **Variable name:** non-modifiable. Indicates the name present in the controller application
- **Read only** if the tick box is selected, the variable is read-only
- **Min value/Max value:** the ranges of the resources are set by default. You can manually define / reduce the range that you want to view by selecting the tick box

HMI Window

Clicking on the corresponding icon brings up the visual representation of the **M171P/M172P target** (virtual controller):



HMI Window M171P



HMI Window M172P

From this window you can simulate the status of digital inputs and outputs. If the output is active it is colored red.

EEPROM parameters can be reset using the **Rst E2** key. Finally you can navigate in the menu using the keys of the virtual tool, as if operating on the physical target.

17. CUSTOMER CARE

17.1. CUSTOMER CARE CENTER

If you have questions or require support, contact our Customer Care Center.

Before contacting our Customer Care Center please check the software version of **SoMachine HVAC**.

From the drop-down menu select **Help > About**.

The following screen appears:



Note software **Version**, Build and Computer ID.

17.2. ONLINE HELP

During each program development stage, pressing the **F1** key opens a **PDF** document containing the Online Help in **printable format**:



SoMachine HVAC offers developers On-line Help to assist programmers with each stage of program development.

17.2.1. ONLINE HELP - LED

Only available for M171O.

From the Panel **Project>Resources>Help** access the **LED reference for the developer** section which contains a description of the characteristics of the icons used in **M171 Optimized** and how to manage them using **target variable sysLocalLeds[]**:

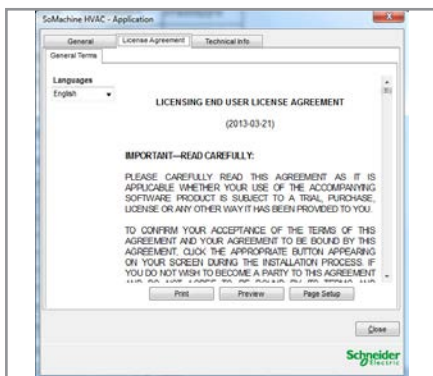
The IEC developer can turn on (either steady or blinking) and off the whole range of local display LEDs, by properly setting the array SYSLED.

LED number	Symbol or icon	Description	Off	On (steady)	On (blinking)
0	:	Colon	SYSLED[0]=0	SYSLED[0]=1	SYSLED[0]=2
1	%R.H.	%RH	SYSLED[1]=0	SYSLED[1]=1	SYSLED[1]=2
2	☸	Defrost	SYSLED[2]=0	SYSLED[2]=1	SYSLED[2]=2
3	Bar	Bar	SYSLED[3]=0	SYSLED[3]=1	SYSLED[3]=2
4	⏻	Stand-by	SYSLED[4]=0	SYSLED[4]=1	SYSLED[4]=2
5	°C	°C	SYSLED[5]=0	SYSLED[5]=1	SYSLED[5]=2
6	❄	Cooling	SYSLED[6]=0	SYSLED[6]=1	SYSLED[6]=2
7	🕒	Clock (RTC)	SYSLED[7]=0	SYSLED[7]=1	SYSLED[7]=2
8	☀	Heating	SYSLED[8]=0	SYSLED[8]=1	SYSLED[8]=2
9	🔻	User-defined 1	SYSLED[9]=0	SYSLED[9]=1	SYSLED[9]=2
10	🔻	User-defined 2	SYSLED[10]=0	SYSLED[10]=1	SYSLED[10]=2
11	🔻	User-defined 3	SYSLED[11]=0	SYSLED[11]=1	SYSLED[11]=2
12	🔻	User-defined 4	SYSLED[12]=0	SYSLED[12]=1	SYSLED[12]=2
13	🔻	User-defined 5	SYSLED[13]=0	SYSLED[13]=1	SYSLED[13]=2
14	🔻	User-defined 6	SYSLED[14]=0	SYSLED[14]=1	SYSLED[14]=2
15	🔻	User-defined 7	SYSLED[15]=0	SYSLED[15]=1	SYSLED[15]=2
16	⚠	Alarm	SYSLED[16]=0	SYSLED[16]=1	SYSLED[16]=2
17	ABC	Menu	SYSLED[17]=0	SYSLED[17]=1	SYSLED[17]=2
18	🏠	Economy	SYSLED[18]=0	SYSLED[18]=1	SYSLED[18]=2

17.3. LICENSE AGREEMENT (EULA)

From the Panel **Help>About>** access the **License Agreement**.

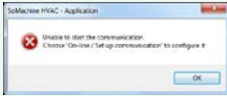
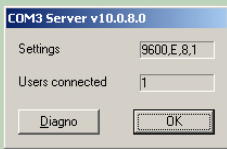
Carefully read the End User License Agreement (EULA) before installing the software.



18. TROUBLESHOOTING

The tables below show some examples of error messages and the corrective action to be taken. For messages not shown here, contact Customer Care Center or consult www.schneider-electric.com.

18.1. APPLICATION

Error message	Panel	Effect	Resolution
	Communication	target not connected	Check the COM port settings using the drop-down menu On-line > Set up Communication > Settings TM171DMI Properties.
 <p>ComServer icon not visible in the applications bar</p>	Communication	target not connected	disconnect communication hardware (for example M171 Optimized Programming cable, USB/485 converter, etc). NOTE: disconnect the target end first and then the PC end
Invalid memory image file. Please upload memory image from the target	Output	compilation error	connect M171 Optimized target device to PC by means of M171 Optimized Programming cable.

18.2. DEVICE

Error message	Panel	Effect	Resolution
invalid connection string	Connection Status	target not connected	Check the communication settings by using the menu On-line > Set up Communication > Settings TM171DMI Properties or by editing the string from the menu Communication > Advanced .

18.3. SIMULATION

Error message	Panel	Effect	Resolution
no target definition for simulator (TGSX) found Unable to run simulator	-	Simulation doesn't start	from Panel Project > Select target select the correct target from those available and confirm with Change .

19. SHORTCUTS

The table shows some of the key combinations (**shortcuts**) available for enabling/disabling the main functions of **Application**. For **Device** see next table and/or **NOTES** column.

Connection and **User Interface** have shortcuts for opening, saving and compiling projects

The “**toggle**” function indicates that the function is enabled if disabled and vice versa by using the same sequence of keys.

For example: CTRL+U enables Full Screen; pressing CTRL+U disables it again

in Device, the display of the Panels is defined by the View menu, by checking individual headings or leaving them unchecked

The Online Help is activated using the **F1** key

Menu	Function	Function	Shortcut	NOTES
ALL	ONLINE HELP	HELP	F1	printable HELP in PDF format ALWAYS AVAILABLE

19.1. APPLICATION SHORTCUTS

Menu	Function	Function	Shortcut	NOTES
Edit	Print	Print	CTRL+P	***
Edit	Cut	Cut	CTRL+X	***
Edit	Copy	Copy	CTRL+C	***
Edit	Paste	Paste	CTRL+V	***
Edit	Save	Save	CTRL+S	***
View > Toolbars	Full screen	Full Screen	CTRL+U	Toggle***
View > Toolbars	FBD bar	FBD BAR	CTRL+D	Toggle
View > Toolbars	LD bar	LD BAR	CTRL+A	Toggle
View > Toolbars	SFC bar	SFC BAR	CTRL+Q	Toggle
View > Toolbars	project bar	PROJECT BAR	CTRL+J	Toggle
View > Toolbars	network	NETWORK	CTRL+M	Toggle
View > Toolbars	document bar	DOCUMENT BAR	CTRL+N	Toggle
View > Tool Windows	workspace	WORKSPACE	CTRL+W	Toggle
View > Tool Windows	output	OUTPUT	CTRL+R	Toggle***
View > Tool Windows	library	LIBRARY	CTRL+L	Toggle
View > Tool Windows	watch window	WATCH WINDOW	CTRL+T	Toggle***
View > Tool Windows	asynch graphic window	ASYNCH GRAPHIC WINDOW	CTRL+K	Toggle***
PROJECT	COMPILE	COMPILE	F7	if more than one project, compiles only selected project
COMMUNICATION	Download code	DOWNLOAD CODE	F5	

Menu	Function	Function	Shortcut	NOTES
DEBUG	Inserts new items into watch	ADD SYMBOL TO WATCH	F8	when the variable is selected (drag & drop not needed)
DEBUG	Inserts new items into watch	INSERTS NEW ITEMS INTO WATCH	SHIFT + F8	chosen from a list
DEBUG		ADD SYMBOL TO A DEBUG WINDOW	F10	
DEBUG		INSERTS NEW ITEMS INTO A DEBUG WINDOW	SHIFT + F10	
DEBUG		ADD/REMOVE BREAK POINT	F12	
DEBUG		ADD/REMOVE TEXT TRIGGER	F9	
DEBUG		ADD/REMOVE GRAPHIC TRIGGER	SHIFT + F9	
DEBUG		REMOVE ALL TRIGGERS	CTRL+SHIFT + F9	
DEBUG		Trigger list	CTRL+I	
*** also applies to Device				

Shown in the table are some of the key combinations available (**Shortcuts**) for opening the respective drop-down menus

Menu	Function	Function	Shortcut	NOTES
Drop Down Menu	Open drop-down menu	File	ALT+F	***
Drop Down Menu	Open drop-down menu	Edit	ALT+E	***
Drop Down Menu	Open drop-down menu	View	ALT+V	***
Drop Down Menu	Open drop-down menu	Project	ALT+P	
Drop Down Menu	Open drop-down menu	On-line	ALT+O	
Drop Down Menu	Open drop-down menu	Debug	ALT+D	
Drop Down Menu	Open drop-down menu	Window	ALT+W	
Drop Down Menu	Open drop-down menu	Tools	ALT+T	
Drop Down Menu	Open drop-down menu	Developer	ALT+D	
Drop Down Menu	Open drop-down menu	Help	ALT+H	***
*** also applies to Device				

19.2. DEVICE SHORTCUTS

Menu	Function	Function	Shortcut	NOTES
Edit	Print	Print	CTRL+P	***
Edit	Cut	Cut	CTRL+X	***
Edit	Copy	Copy	CTRL+C	***
Edit	Paste	Paste	CTRL+V	***
View > Toolbars	Full screen	Full Screen	CTRL+U	Toggle***
View > Tool Windows	output	OUTPUT	CTRL+R	Toggle***
View > Tool Windows	watch window	WATCH WINDOW	CTRL+T	Toggle***
View > Tool Windows	asynch graphic window	ASYNCH GRAPHIC WINDOW	CTRL+K	Toggle***
PARAMETERS	read selected	READ SELECTED	CTRL+SHIFT+R	the selected parameters are highlighted in yellow
PARAMETERS	write selected	WRITE SELECTED	CTRL+SHIFT+R	
PARAMETERS	select all	SELECT ALL	CTRL+A	Operation to be performed with care The next operation could be writing all parameters and/or resetting the default values
*** also applies to Application				

Shown in the table are some of the key combinations available (**Shortcuts**) for opening the respective drop-down menus

Menu	Function	Function	Shortcut	NOTES
Drop Down Menu	Open drop-down menu	File	ALT+F	***
Drop Down Menu	Open drop-down menu	Edit	ALT+E	***
Drop Down Menu	Open drop-down menu	View	ALT+V	***
Drop Down Menu	Open drop-down menu	Parameters	ALT+P	
Drop Down Menu	Open drop-down menu	Recipies	ALT+R	
Drop Down Menu	Open drop-down menu	Options	ALT+O	
Drop Down Menu	Open drop-down menu	Help	ALT+H	***
*** also applies to Application				

19.3. USER INTERFACE SHORTCUTS

Menu	Function	Function	Shortcut	NOTES
MENU BAR	new project	NEW PROJECT	CTRL+N	***
MENU BAR	open project	OPEN PROJECT	CTRL+O	***
MENU BAR	generate code (compile)	GENERATE CODE (COMPILE)	F7	

*** also applies to **Connection**

19.4. CONNECTION SHORTCUTS

Menu	Function	Function	Shortcut	NOTES
MAIN BAR	new project	NEW PROJECT	CTRL+N	***
MAIN BAR	open project	OPEN PROJECT	CTRL+O	***
MAIN BAR	Save	Save	CTRL+S	

*** also applies to **User Interface**

20. APPENDIX PROJECT DEVELOPMENT EXAMPLES

20.1. PROJECT 1 - COUNTER

In the pages that follow, we will guide you through the procedure for writing, testing and downloading to M171 Optimized a simple COUNTER program written in ST language, which performs a count and displays a “saw-tooth” on the oscilloscope.

The COUNTER program counts the local variable **a** from 0 to MAXSCALE1. When **a** reaches MAXSCALE1 it will be re-initialized to -100.

As previously described in **First Start-up**,

- start **Application**,
- create a new **COUNTER** project
- create a **COUNTER** program in **ST** language.

In the central **Edit** section, write the **COUNTER** program code in **ST** language.

COUNTER program

```
if a>=MAXSCALE1 then
a:=-100;
else
a:=a+1;
end_if;

b:=a*10;
count:=a;
counter1:=b;
```

Local variables

In the variables section we create the DINT type local variables **a** and **b** (use the drop-down menu **Variables > Insert**, or the icon **Insert record**) and associate the **Init Value** 0 (zero).

The variables **a** and **b** are of no use outside the COUNTER program, and are therefore defined as local.

Variables and global constants

In the program, we have also included a constant which we will define as “global”:

the constant MAXSCALE1 must be created in the Panel **Project >Global Variables Constants**, which we will create here in the same way as we create a local variable. MAXSCALE1 has a preset value of -100 so it is not necessary to define it as a variable but as a constant.

Global shared variables

To display the result of the counter on the **target**, in the program we have also included 1 variable which we will define as **Global shared (NOTE that the folder “Global shared” appears if, and only if, we define the variable in the tab Resources>Modbus Objects > Status variables)**:

The variable **count**, which we will create, is obtained from the Panel **Resources>Modbus Objects > Status variables** and not from the section Edit/Variables, when we enter a new record.

In this case it will be necessary to also define how the name of the variable will appear on the **M171 Optimized** 7-segment display (for example label of up to 4 digits, for example **CNT** for **count**) and mapping between **Application Type** and **Device Type**.

In the COUNTER program we have included another variable, `counter1`, which we will define in the same way. **NOTE** that if we had created a variable `counter` and not `counter1` at the compilation stage, we would have had an error (name conflict between COUNTER program and variable)
 In this case, we do not define how it appears on the display.
 The software will propose **COUn** by default (instead of COUnTer), in other words it will use the first 4 letters on the 4-segment display.

Fundamental state display

By clicking on **Resources>M171 Optimized** we choose what to show as the fundamental state display
 As you can see, the choice includes the default I/O + the variables created by the developer
 We choose the variable `count` (which is the value of the local variable "a")
 We update the program by entering an always active or always inactive alarm

```
if alarm then
alarm1:=0;
else
alarm1:=1;
end_if;
```

The alarm variable must be defined as a local variable - of BOOL type (Boolean). **FALSE** is proposed by default as the initial value - to change in **TRUE** if needed.
 The alarm will be always active. We can enter a description such as "Alarm always active"
 Alarm1 must be defined in the Panel Resources>Alarms. As a label we enter ALM1.

Menu creation

- At this point we can create
- the customized menu PRG1 for entering any parameters defined by the developer (or use the existing BiOS parameters)
 - the customized menu SET1 for entering the 2 variables `count` and `counter1`.

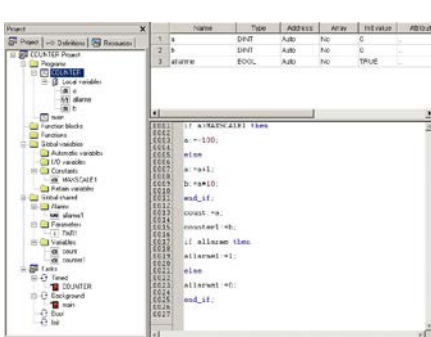
Task

If we go back to the Project tab and select the task `Timed` we can add the program `COUNTER`.

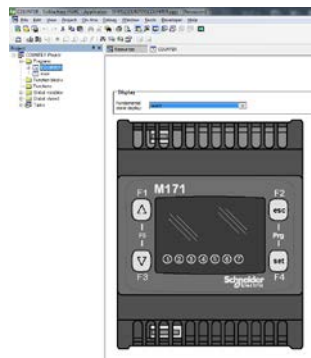
Compilation

We compile the project - drop-down menu **Project >Compile** (or F7)
NOTE: the project is compiled without errors and without the need to use an **M171 Optimized** connected to the PC

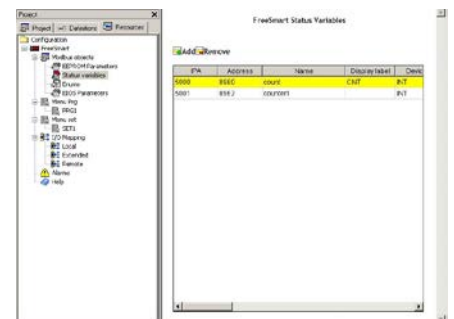
In short...



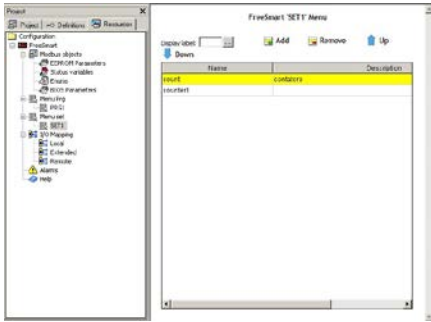
COUNTER program - Project tab



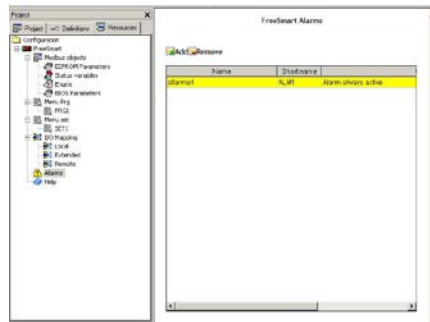
COUNTER program - Resources tab fundamental state display



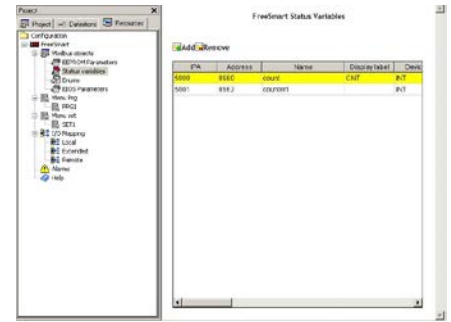
COUNTER program - Resources tab defining Variables



COUNTER program - Resources tab menu Set (SET1)



COUNTER program - Resources tab Alarms menu



COUNTER program - Resources tab defining Variables

NOTE: in the Project tab, the global shared variables appear only after they have been declared in the Resources tab. In the example:

- Alarms → alarm1
- Parameters → PAR1
- Variables → count and counter1.

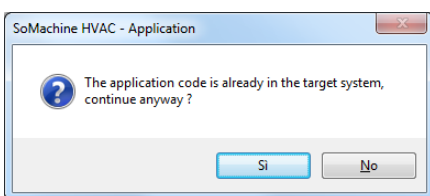
Connecting to M171O and downloading the application

After connecting M171 Optimized to the M171 Optimized Programming cable and PC on the USB port and making the appropriate checks on the COM and baud rate (see relevant chapter) we can connect to the target:

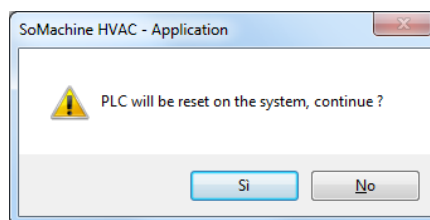
- from the drop-down menu **Communication > Connect** or icon **Connects to the target** connect to **M171 Optimized**
- if connection is successful, the message **CONNECTED** appears in green at the bottom right of the screen*
- from the drop-down menu **Communication > Download code (F5)** or icon **Code download** now download the application to **M171 Optimized**
- one or more dialog boxes** will appear, indicating that the application will be downloaded. Downloading could take a few seconds

*Should this not be the case, see Download on target or Troubleshooting.

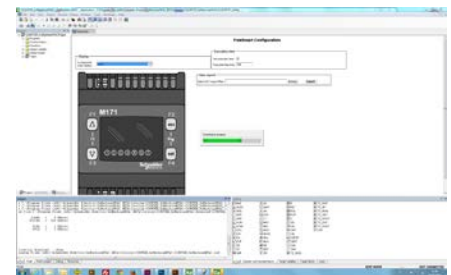
**depending on whether the target has already been programmed or not.



Controller application existing on target continue confirmation



reset PLC reset confirmation



application download in progress

The application is now operating on M171 Optimized.

Watch and Oscilloscope

On the display of the target device, the count value appears which ranges from -100 to +100.

count is set as a fundamental display. If you do not set any variable, the message PLC appears

By pressing F5, you can access the variables and parameters of BIOS

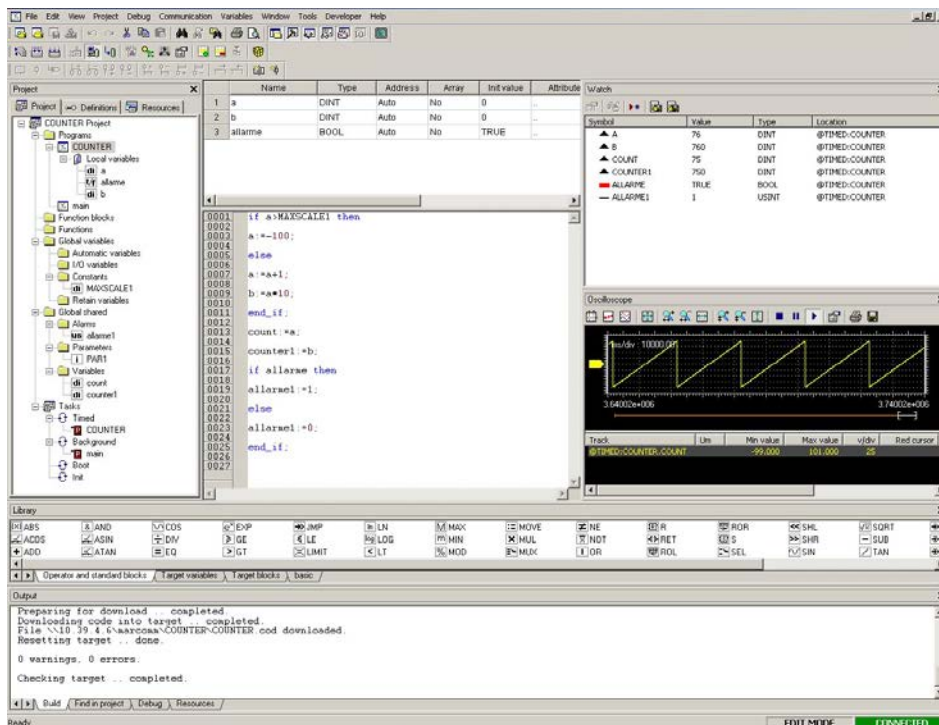
NOTE that the alarm LED appears because the alarm is active

In the Status menu, we can check for the presence

- of the folder SET1 with the variables CNT and COUNT
- of the folder AL with the alarm ALM1 to 1

On **Application** we can test the value of the variables and display them as text or graphics by dragging the variable from the code in the Panel **Watch** and **Oscilloscope**

See screenshot below



- **NOTE** how the local variables **a** and **b** cycle from -100 to 100 (x 10 for variable b), as do the corresponding variables count and counter1
- "alarm1" is an always true Boolean variable (TRUE shown in red) and alarm1 is fixed at 1
- **NOTE.** if you change "alarm1" from TRUE to FALSE, re-compile and download the new application, the alarm LED will switch off and the variable will be indicated in green in the Watch Panel
- The variable count is displayed in the oscilloscope and we can see the "saw-tooth"

Saving the project

We can save the project as counter.ppjs or a name of our choice, generate an .rsm file, password-protect it and export it as a library object (.pll).

20.1.1. DEVICE

The project can be opened with **Device**:

we open the tool of the same name, from which we can read or write one or more BIOS parameters, change them and/or download the default again.

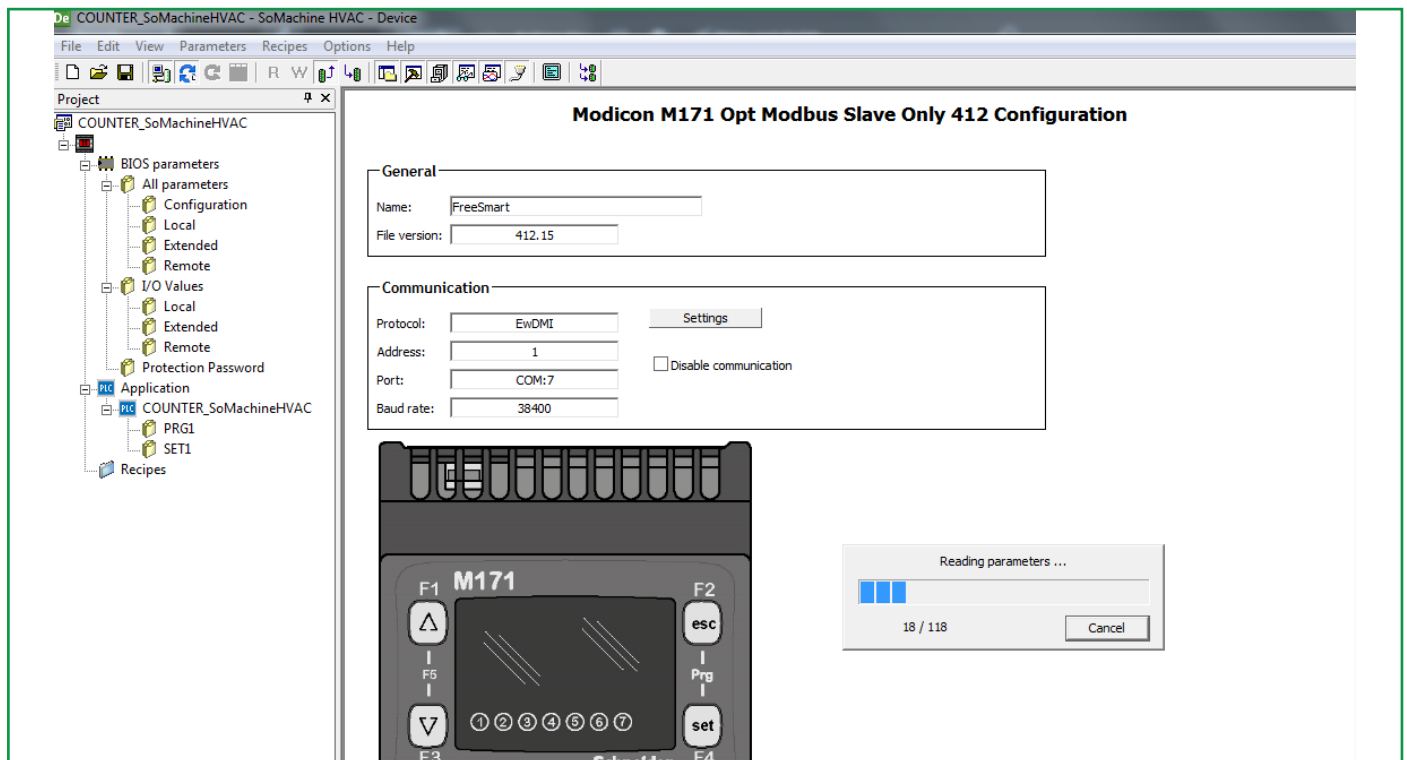
It is possible to read the current value of the variables defined by the developer but not change them

Device does not require the user to know any programming language; the source code is not visible.

Device requires you to connect to the target device again (click on the respective icon). The central Panel shows the protocol references, address, etc.

When the target device is connected, you can read the parameters or write the default or changed parameters.

Reading parameters





Displaying variables

As with **Application**, with **Device** it is also possible to display the variables by dragging them to the Watch and Oscilloscope Panels.

Writing parameters

To download all the CL parameters (local BIOS) and no others:

Select all the parameters - drop-down menu **Parameters>Select all** (CTRL+A) or using the respective icon . The parameters are all highlighted in yellow. Write drop-down menu **Parameters>Write selected** (CTRL+SHIFT+W) (or using the icon )

All the CL parameters will be downloaded to the target device.

We can save the project as Counter.CFN or a name of our choice.

20.2. PROJECT 2 - COUNTER2 FUNCTION BLOCK

Starting from the COUNTER program described in the previous example, we create a function block and we modify the controller application by adding code for the switching of a LED / relay.

Saving the project

We save the COUNTER project by renaming it COUNTER2:
from the drop-down menu **File>Save project as** proposes the COUNTER work folder.
Go up one level, create a new COUNTER2 folder and save the project as COUNTER2.

Creating a function block

Create a new function block **COUNTERST** in **ST** language.
Copy/paste the COUNTER program into the Edit Panel of the function block with several variants

COUNTERST program (function block)

```
if a>MAX then

a:=-100;

elsif enable then

a:=+1;

end_if;

out:=a;
```

Local input and output variables

In section B (see Application Panels Definitions) **variables** we create the following variables:

- **a** as local variable
- **MAX** and **enable** as input variables to the function block
- **out** as output variable from the function block

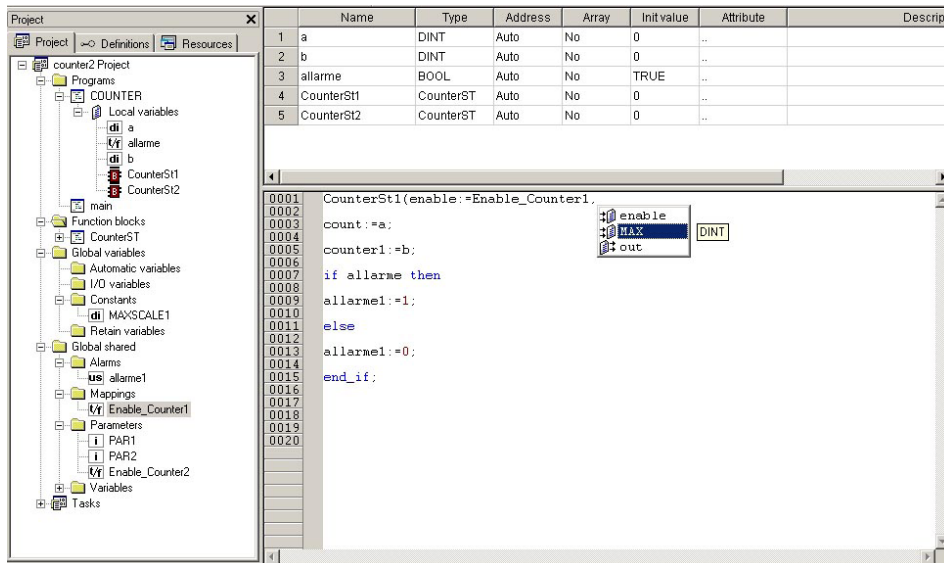
The function block will therefore have **2 input variables, one output variable and one local variable** which will save the status of the counter in each instance of the function block

The screenshot displays the SIMATIC Manager interface. On the left, the Project Explorer shows the hierarchy: counter2 Project > Programs > COUNTER > main > Function blocks > CounterST. Under CounterST, there are sections for Input variables (enable, MAX), Output variables (out), and Local variables (a). On the right, the Edit Panel shows the ST program code for CounterST, which is a copy of the code provided in the previous section. Above the code editor, a table lists the variables defined in the function block:

	Class	Pin	Name	Type	Array	Init value	Attribute
1	VAR		a	DINT	No	0	..
2	VAR_INPUT	0	MAX	DINT	No	100	..
3	VAR_INPUT	1	enable	BOOL	No	FALSE	..
4	VAR_OUTPUT	0	out	DINT	No	0	..

We now write the COUNTER2 program in which the functional block will be called.

The function block will have 2 instances `CounterSt1`, `CounterSt2` defined as local variables of `CounterSt` type. As we write the code, **NOTE** how **Application** guides the developer writing the code by proposing the variables to pass to the function block (in the example 2 input variables (identified by an icon with the arrows on the left, and the output with the arrows on the right)).



The Boolean variable in this case will serve to switch:

- a suitably configured relay
- an LED (we use the variable `target sysLocalLeds`. See relevant section) of the target on the basis of a defined condition.

We now write the new code COUNTER2:

COUNTER2 program

```
CounterSt1(enable:=not Enable_Counter1,MAX:=MAXSCALE1);
counter1:=CounterSt1.out;
```

```
CounterSt2(enable:=Enable_Counter2,MAX:=MAXSCALE1/2);
counter2:=CounterSt2.out;
```

```
alarm:=(counter1>counter2);
```

```
if alarm then
```

```
alarm1:=1;
Alarm_relay := TRUE;
```

```
else
```

```
alarm1:=0;
Alarm_relay := FALSE;
```

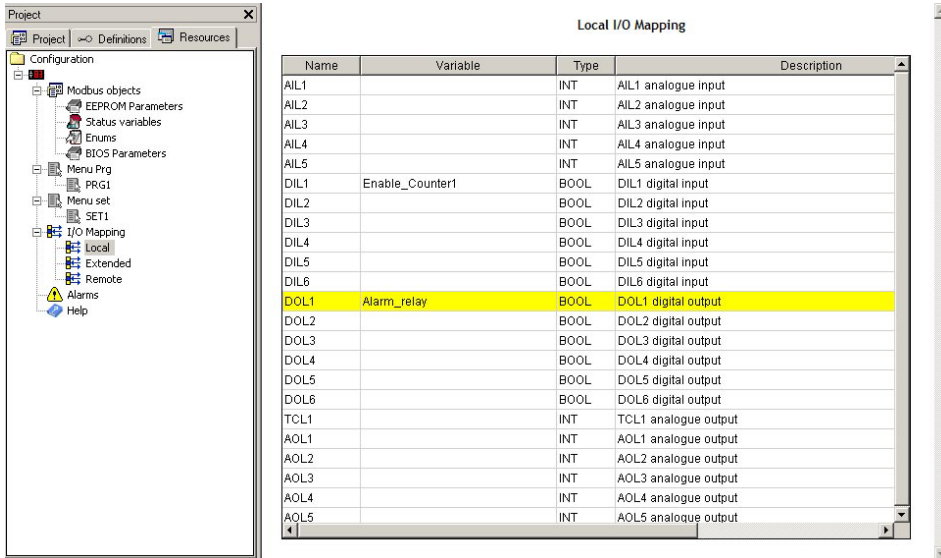
```
end_if;
```

```
if (counter1<=counter2) then
    sysLocalLeds[6]:=1;
```

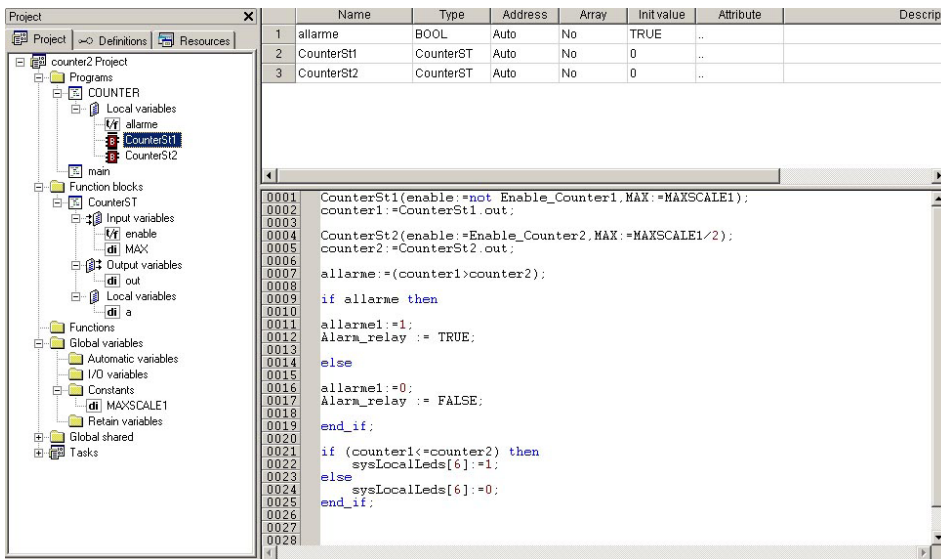
```
else
    sysLocalLeds[6]:=0;
end_if;
```


In the **Resources** tab, it is necessary to create and/or rename, from the previous project, the EEPROM parameters, status variables, etc.
 It is also necessary, if desired, to add the resources to the **Prg menu** and **set menu**.

In the **I/O mapping** menu, lastly, it is possible to associate resources to the inputs/outputs
 We associate **Enable_Counter1** to a digital input and **Alarm_relay** to an output.



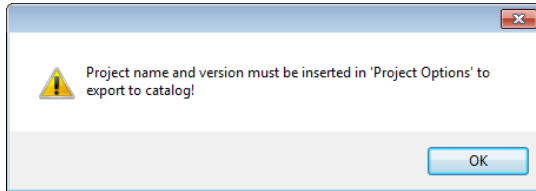
The program is now completed:



Exporting to target device

From the drop-down menu **Developer>Export application to catalog** remember to export the project to catalog so as to enable **Device** to re-use it for downloading to the **target**.

Before exporting to catalog, use the drop-down menu **Project>Options** to assign to the project a release (for example the same name **counter2**), a version (for example **1.0**) and compile the project again.



This will enable **Device** to recognize the controller application on the **target** when it is downloaded.

Compile the project and download it to **target** by means of **Application** or **Device**.

The counter will display a count from -100 to 100 in the same way as the COUNTER controller application previously described.

In this case, the Alarm LED will switch to the compressor LED (SysLocalLeds[6], The alarm LED is SysLocalLeds[16]) once the defined threshold has been reached.

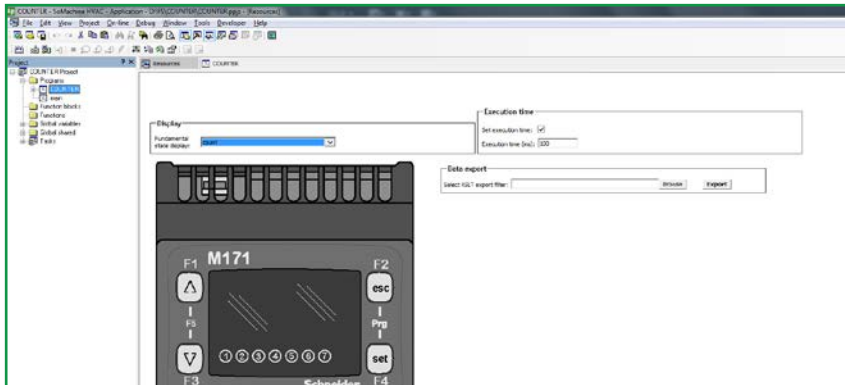
21. APPENDIX EXPORTING CSV FILES

21.1. EXPORTING CSV FILES

SoMachine HVAC allows you to export parameters and variables defined in **.CSV** format which can be used for sharing information and developing documentation to be supplied with the product.

Starting from an existing project (for example PROJECT 1 - COUNTER of the previous chapter):

Application

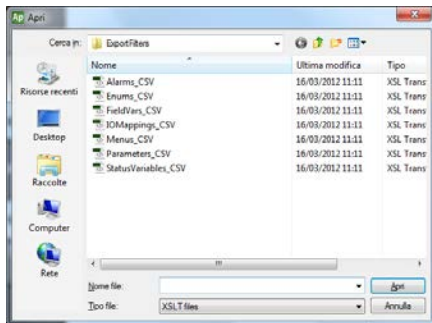


- start **SoMachine HVAC Application**
- click on **Browse**

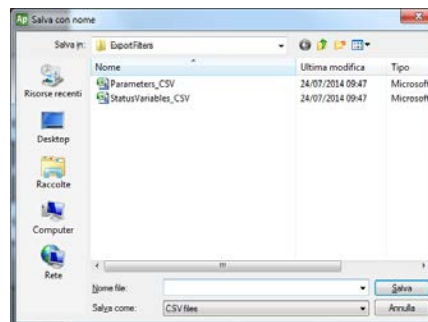
The dialog window appears in which you can choose the type of resource to be exported:

Export Filters

From the **Export Filters** folder choose the filter to be applied for exporting the desired resources. Once you have chosen, click on **Export**, give the file name and **Save**. A message will confirm that it has been saved.



Export Filters



Export CSV



Export successfully completed

The table shows the resources that can be exported:

XLST	Panel	Notes
Alarms_CSV	Resources > Alarms	-
Enum_CSV	Resources > Modbus Objects > Enum	-
FieldVars_CVS	Resources > I/O Mappings>Field	-
IOMappings_CSV	Resources > I/O Mappings	for M171P only
Menus_CSV	Resources > Menu set	M1710 : Local, Extended, Remote M171P/M171P Flush : Local
Parameters_CSV	Resources > Menu prg	-
StatusVariable_CSV	Resources > Modbus Objects > Status Variables	-

Export example **StatusVariable_CSV**, **COUNTER** project:

Global shared variables

The excel file contains the 'shared' variables (**Global Shared**).

The **count** variable, created by the Panel **Resources>Modbus Objects > Status variables**.

The **M171 Optimized** 7-segment display indicates the label with max 4 figures - **CNT** for count.

In the **COUNTER** program we have also defined another variable `counter1` which we shall define in the same way.

NOTE that if we had created a variable `counter` and not `counter1` at the compilation stage, an error would be detected (name conflict between **COUNTER** program and variable).

In this case we do not define how it appears on the display.

Address	Name	Display label	Device type	App type	Size	Default	Min	Max	Scale	Offset	Unit	Format	Level	ReadOnly	Description	Note
8960	count	CNT	Signed 16-bit	DINT					1	0			Always visible	R	contatore	
8962	counter1		Signed 16-bit	DINT					1	0			Always visible	R		

This description applies to **Application**.

21.1.1. Device

In the **Device** environment it is possible to export the parameters and the variables in **txt** format from the menu **Parameters > Export to text file**:

Value	Unit	Lim	Default	Min	Max	Description
0	num	0	0	0	14	Serial COM1 protocol
0	num	0	0	0	14	Protocol controller address
0	num	0	0	0	14	Protocol controller family
1	num	1	1	1	255	Modbus protocol controller address
3-9500	num	3-9500	0	0	7	Modbus baud rate protocol
1=Even	num	1=Even	1	3		Modbus parity protocol
0	num	0	0	0	999	Customer code 1
0	num	0	0	0	999	Customer code 2
2=NTC	num	2=NTC	0	8		AI1.1 analogue input type
2=NTC	num	2=NTC	0	8		AI2 analogue input type
2=NTC	num	2=NTC	0	7		AI3 analogue input type
2=NTC	num	2=NTC	0	7		AI4 analogue input type
2=NTC	num	2=NTC	0	8		AI5 analogue input type
500	°CBar	500	-9999	9999		AI3 analogue input full scale value
0	°CBar	0	-9999	9999		AI3 analogue input start of scale value
500	°CBar	500	-9999	9999		AI4 analogue input full scale value
0	°CBar	0	-9999	9999		AI4 analogue input start of scale value
0	°C	0	-120	120		AI1.1 analogue input differential
0	°C	0	-120	120		AI2 analogue input differential
0	°CBar	0	-120	120		AI3 analogue input differential
0	°CBar	0	-120	120		AI4 analogue input differential
0	°C	0	-120	120		AI5 analogue input differential
0	0-0.20mA	0	0	2		AO1.1 analogue output type

For the other working environments (**User Interface**) it has already been described how to export the text files for translating the pages, or for generating documentation with the key **Generate Doc**.

