MaticStudio 1.3

User guide

This manual has been carefully written, however we cannot exclude that any error could be slipped. Some information placed here could slightly differs from the actual appareance or behaviour of the program.

This manual is refered to the version 1.3.30, or higher, of the software.
Notice that when updates are made via the Internet, the manual could be not correspondently updated.

Getting start with MaticStudio

MaticStudio allows to develop and to program the devices based on the networked cells technology Networked Shared Controllers (short NSC), either for applications in industrial automation, and for building automation.

The programs so made can apply to applications in systems to control machineries, and for control of automation in buildings.

Terminology

To distinguish among different elements that are part of a program, its functions and accessories, specific terms are in use:

Development envirnoment	Is the MaticStudio software
User program	Is your program
Project	Collection of elements that are required to build the user program
Object	User program's element, typically it corresponds to a device or in ceratin cases an “abstract device”, in example the supervision software
Device	Physical equipment, typically a NSC controllor
Source text	User program as text (see advanced mode programming)
Virtual component	Component which perform functions and can be embedded in an object (pratically a NSC controller) (see Virtual components)
Compilation	Conversion of the user program from source text to machine binary code
Programming of the devices	Send to the devices of the compiled user program
Links among inputs and outputs	Are the bonds among input events and output actions in the object itself or among different objects, that made up the logic relations of the user program.

Working requirements

To work properly, MaticStudio must be installed in a computer with operating system Windows (98, ME, 98SE, NT4SP4, 2000, XP, 2003). Microsoft Vista at this time does not properly supports yet USB-serial drivers available in the market, and it shown some incompatibility that Microsoft will probably solve in next future.
Other requirements: almost a serial RS232 port onto which should be connected a serial SFBP over 485 - RS232 adapter, model PCAdapStd or PCAdap503, or via a USB to serial adapter; almost 50MB of free hard disk space, authorization to write in the disk folder onto which the software is installed; 32 bit color graphic card with resolution of almost 800x600 pixels; memory almost 256MB RAM; mouse and keyboard; PentiumII 800MHz or Duron Athlon 900MHz class processors or above; serial port RS232 or USB2 with USB to serial converter. Further space on disk is required to save data and your user programs.
Internet access to download updates, components, and to show on-line documentation.

Beware that some USB converters have synchronization problems, the PCMCI serial card with multiple port are not recommended.

Programming

Programming can be done in two ways. Visual mode and advanced, textual, mode.
For textual mode, please refer to the language programming manuals and API documentation for the NSC operating system which control the physical devices.
Such manual is useful even to write virtual components by the means of ComponentBuilder.

Virtual components

They are components that can handle some specific tasks, and they can be embedded in an object (pratically a NSC controller).

The Virtual components can be seen as true devices, in example a device to control temperature, or a burglar alarm device, or a positioning controller; with the specificity that they are not physical until they are embedded within an object, and in turn a physical device that embodied them.

It can be embedded several virtual components, in different combinations, to shape this way a personalized device. The virtual components have a simple configuring interface which allows to defines how they will works, in example setting timers and selecting what inputs and outputs should be used when perfoming the tasks for which they are designed.

Finally the virtual components have the advantage that they can be added or modified simply downloading them from our server via Internet, mailed or copied from discs, given the fact that they become physical only at the time they are embodied in the final device.

Making components

ComponentBuilder is the tool designed to make virtual components, it is sold as a separate suit, and it is not included in the standard installation of MaticStudio.

Installation

Insert CD into the drive reader, so open the Setup folder, and start the Setup.exe program located within.

Follow the directions shown on video.

When finished, start the program. A dialog box will be shown in which you'll be prompted to enter your company name, and your name, as the author of your programs.

In addition, the RS232 serial port number is proposed, and the SFBP driver adapter settings (they are not related with any external USB to serial converter).

Normally the only one setting to do is the serial port number.

Click on the OK button.

You have successfully installed your copy of MaticStudio, we feel confident that you will appreciate the features and characteristics of our program, and we thank you to choose our product.

Check our for updates

Just after the installation check out if a new version is already available, this will also align your installation with the latest version.

If the new project wizard is open, close it clicking on the X button, on top-right.

Be sure to be connected on the Internet, then from the Tools menu choose the item Find MaticStudio updates... if a new version is available click OK to let update it. Be patient to let donwload be completed, then when prompted hit the OK button to finish the update.

Program guide

How MaticStudio shows itself

The development enviroment offers two main views: Project and the text editor of the program, useful when in advanced programming mode, and to make special programs for machineries.
The Project view is in turn split into three other views: Objects, Inputs & Outputs (also called connections), and the Virtual Components & Links (or bonds); this last is useful for the development of domotic and industrial applications in visual mode programming. Each one of these views are enhanced by a different color to help you focus better in what area you are working on.

When in the Virtual components & Links view (actually the visual programming mode core), are available other two views: the Components view and the Links view, respectively enhanced by a violet command and a yellow command.

Creating a new project, MaticStudio starts a “wizard”, a guided sequence that drives you to fill all the information required to make a new project.

While in the wizard stage, on top are shown the tools to enter the objects (the devices), the connection manager to enter all the I/O connections, and the virtual components toolbox, this last reachable also clicking on the icon .

Menu bar and quick command bar

On top the menu of the progam, and below if it the quick command bar, as shown below:

Menu, and corresponding quick commands icon, placed aside to the menu items.

File	File and print management	
	New	Create a new project
	Open...	Opens a project saved onto disk
	Open compile C	Opens the generated text C source
	Save, Save as...	Save or Save as named file on disk, the project
	Print program...	Prints the text source of the program
	Print project	Prints all the project, making a full report with project properties, inputs and outputs, objects and related bond device type, the source text program, and the embedded components.
	Quit	Exits MaticStudio

Edit	Edit text, find & replace, swap view Project / Text code, and arranges connection I/O list:	
	Undo Redo Cut Copy Paste	Edit text
	Find & replace	Find in text (or in list) and replace portion of text (if required, not allowed on lists)
	Edit list / Reorder	Reorder the connection list
	Name automatically	Automatically set names to all the connections provided by each device bond to each object in the project
	Project	Shows the project (is shown the last selected view)
	Text	Shows the source text code
Project	Compile and send the program to the devices, project properties:	
	Compile	Compiles the project to machine code, ready to send the program to the devices
	Program devices	Send the program to the devices (and compile if needed). Note: if project has only one device, this command has the same effect to click on the menu item Single progam.
	Program single	Send (and compile if needed) the program to a single device of the project. Note: same action of clicking on Progam devices and select only one device.
	Properties	Shows the project's properties dialog box, in which is possible to make modification (title, Customer, etc.)

Tools	Utility tools and options:	
	Autolearning...	Opens the dialog box for autolearning
	Bus...	Opens the dialog box to configure, diagnostic and test the SFBP network onto which the NSC devices are connected (see Tools chapter)
	Device detection	Opens the dialog box to grab the devices
	Ladder editor	Opens the ladder editor to insert ladder expressions in the source text code. (Ladder editor automatically pops up even to draw contact sequences in the visual programming mode linking with an expression, and can be shown even to configure some components).
	Virtual componenti (pops up the following submenu):	
	Virtual components	Opens the tool to select and embed virtual components
	Install new components...	install new components from disk
	Download & install components...	Connects to our server via the Internet and check out if new components or updated are available. The dialog box that appears will shows the components that are new (preselecting the items), and let select what components to actually download and install.
	Icon deskdraw	Opens the tool to draw icons for LCD 128x64 display units.
	Scheduler	Opens the tool for reading and setting scheduling to the devices that have this feature.
	Change DST parameters	Opens the dialog box to set the Daylight Saving Time parameters, for the automatic swicth between DST and standard time for the devices that provides this feature.
	Component builder	Starts ComponentBuilder (only if installed).
	Options...	Opens the options dialog box.
	Find MaticStudio updates...	Connects to our server via the Internet to check out if an updated version of MaticStudio is available. If yes a promt will ask you to proceed with the installation.
	Download new supervision styles...	Connects to our server via the Internet, to check out if new styles for supervision pages are available.
	Copy display icons onto clipboard	Copies onto clipboard icons created with Icon deskdraw, to let paste them in ComponentBuilder.
Supervision	Opens the supervision editor window	
Information	Shows the information of the program MaticStudio	

Others quick commands

	Active only on text, if insertion cursor of text is within the parentheses of an intrinsic function, it shows a tip with the prototype of the function.
	Serial communication port is working. Releasing this button the communication is stopped, and other programs in the computer may access to it.
	Serial communication low level error, the error disappear if it is temporary. Releasing the button and then pressing it again, it is possible to manually reset the communication channel.
	Serial communication port is closed. Pressing the button the serial port is open again. Notice that this condition is true also if an other program in the computer is engaging the same serial port before MaticStudio opens the port for itself, if the serial channel does not works or when a second instance of MaticStudio is already in execution.

Create a new project

A new MaticStudio project hold in a full and ordered way all the information related to a program finalized to a given application: it holds the components, the options, the devides (objects), the I/O connections of each unit, and of course the program.

To create a new project click on the icon , the first dialog box, the project properties, of the “wizard” guided path is shown. The following steps will be performed:

Dialog box of project properties

- enter the name of the Customer and the title of the project, the version, and optional references and notes.
- choose Make to create a project in which you can enter the objects bond to the type of devices of the installation, or Make & Grab to grab the units already placed and connected in the network, do develop a program on the fly

Next, in the project it will be possible to enter the objects, their connections and add the function of the program. A new project can be prepared also without having the physical devices, that will can be grabbed in a second time.

Enter new objects

To enter new objects must be selected the view **Project – Objects**

Each object is the representation of a real device. Since many devices of the same type can belong the same project, each one of them is represented by an object and the object identified by a univoc name. For example, a domotic project could have objects called “kitchen”, “stairs”, “bedroom”, and so on, based on their own location or main provided tasks; similarly in a instrustrial automation project they could be called with the name of their scope, such as “head”, “transfer”, and so on.
Names are free, provided that it begins by a letter (a to z, A to Z) and contains only letters, numbers or underscores. No spaces are allowed. Names must not be longer then 40 characters, we suggest to choose short names because they are best to read and write.

It is possibile also enter the serial code number of the device (each device have one, printed on their label). however if the device is not available at the moment you can omit to enter the serial number leaving the default (`S000000`) and entering it in a second time, or launching the grabbing procedure.

While in the wizard, on the top is shown the devices toolbox, to enter a new object (see fig.1), select the type of the device clicking on the picture, and enter the name of the object, then hit the Add to project button. When finished to enter all the object intended in the project, click on the Next button to go to the next step, the I/O connections.

fig.1

On the right of the devices toolbox, there is a list of the family. Clicking on the first item are shown all the devices, while clicking on the name of a family are shown only the devices related to that family.

If the wizard has been terminated, reviewing the objects list and clicking the last row under the column of the “type of device”, appears a button that let add a new object popping up the devices toolbox.

While in the guided process (wizard mode), clicking on the view **Inputs/Outputs** after entered a new object, MaticStudio will prompt you to confirm if you want that MaticStudio automatically fills the I/O connections of the new added object. Automatically filled names for I/O connections are composed by the name of the object followed by the number and type of I/O.

Defining inputs and outputs connections

The view **Inputs/Outputs** let specify a name for each input and output connection of each object, representing a device, loaded in the project.
The table shown allows to specify the object, the name of the connection, the type of connection, and an optional description or comment (see fig.2).
Once selected the objecte and the type of connection, the physical clamp terminal number is automatically shown as well.

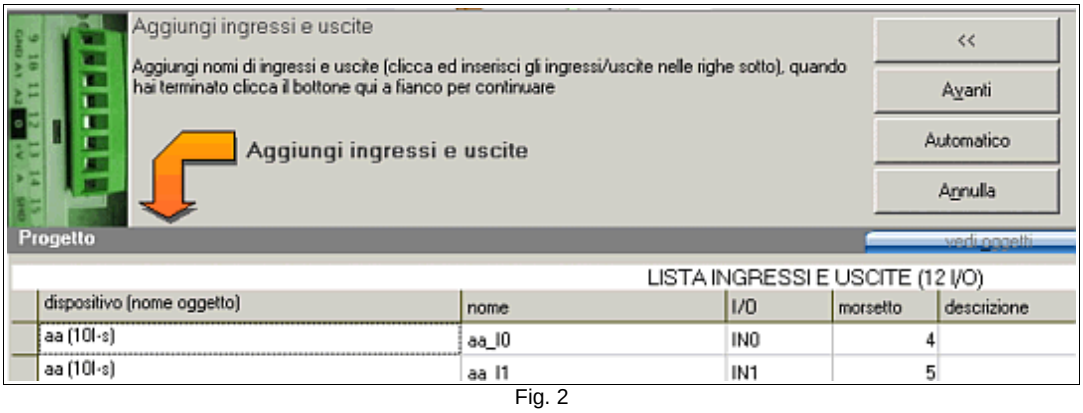


Fig. 2

To enter a new row hit the key ENTER, the rows left empty will automatically removed changing view.
To remove a row, click on the gray box on the left of the row to remove, and hit the key DEL.

NOTES
The names must follow the same rules of the names used by the objects: they must begins by a letter, contains letters, numbers and underscores, and be limited to 40 characters. We suggest to keep names short as they are better readable and writtable and easy to recall.
Clicking on the green button I/O connections the table is updated removing the empty rows, if any.
Changing a device type after input and output connections has been entered, MaticStudio will attempt to reassign each I/O to the new device, however if the input or output does not exist in the new device a warning message pops up and the invalid connections are enhanced.

The button Automatic automatically place the names of all the connections of input and output of each object in the project. Only the I/O connection undefined are named, not the connection already defined and named. The names so given can changed thereafter.

Keyboard commands: Use the key ENTER to add a new row, or to accept a selection if a list is shown; use the arrow keys to move among columns and rows, or to change the item of a list if it is shown.

While in the wizard mode, once completed the definition of the I/O connections, click on the button Next to step ahead and go to the visual programming, or click the button << to go back at the previous stage.

Visual programming

Making bonds between inputs and outputs and embedding virtual components

The view **Components & Links** allows to make bonds between inputs and outputs of a given object and among different objects. Remember tha each object represent a device. Further, can be managed and embedded virtual components in each object.
To switch between the view **Links** (bonds) ad **Components** click on the violet command or alternatively the yellow command.

Links (bonds between inputs and outputs)

Be sure to be in the view Links (the violet button Components is visible).
Three panes are visible. On the left, the input pane, are listed all the inputs of all the objects loaded in the project (and of which has been defined the input names, undefined inputs are not listed).

On center, the output pane, are listed all the outputs of all the objects loaded in the project (and of which has been defined the output names, undefined outputs are not listed).

On the right, the summary pane, is shown the summary of the links, ladder expressions, and links between functions and event functions exposed by the embedded components.

(see fig.3)

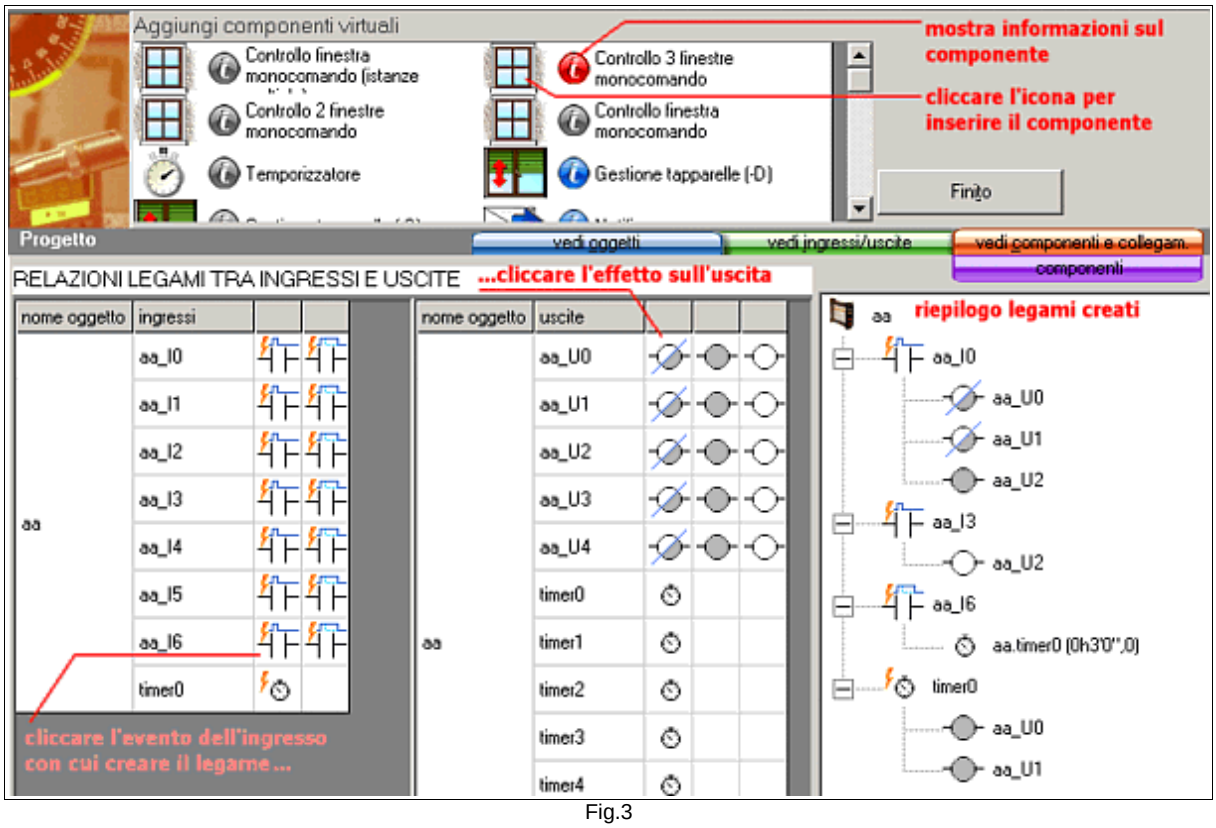


Fig.3

The input pane

For each input are listed the possible events (more events and functions are available working in the advanced text mode).
A given input can infact detect several events. The main events are click (as a button press and release), longclick (as a button keep pressed for almost 3 seconds), normally open status change, normally closed status change.
In addition are listed the internal auxiliary contacts of the outputs as normally open status change and normally closed status change.

In the list of inputs can be also listed other events, such as a timer timeout, or events exposed by embedded components. Finally is listed a special constructor event for ladder expression, where after created they are listed as new ladder expression events. Timers timeout are listed only if they are first set as output (see below).

Events description:

event	description
 click	Happens when a medium quick variation off/on/off is seen by the input.
 longclick	Happens when a variation off to on is seen by the input, and is stable for almost 3 seconds, then released to off again.
 normally open status change	Happens when changes the status of an input as it close: off to on. Remark: it is the stable status, not the transient.
 normally closed status change	Happens when changes the status of an input as it open: on to off. Remark: it is the stable status, not the transient.
 normally open output status change	Happens when changes the status of the auxiliary internal contact of an output, as it close: off to on. Remark: it is the stable status, not the transient.
 normally closed output status change	Happens when changes the status of the auxiliary internal contact of an output, as it open: on to off. Remark: it is the stable status, not the transient
 (*or:* ) timer timeout	Happens when a timer timesout. This event is not listed until the timer has been linked as output.

 This icon let to construct a new ladder expression that thereafter is added as a new custom event. The event will be raised only if a variation in ones of the inputs involved in the expression will happens. Only inputs related to the object in which the expression is linked can raise the event, inputs coming from other objects cannot.

The output pane

In the output pane, for each output are listed all the possible actions on the outputs, the timers, the merkers, and the functions exposed by components embedded, if any.

Output actions description:

action	description
 toggle	The output status toggles: if it were energized is de-energized, if it were de-energized is energized. Warning: linking different outputs with this action, on different devices, could lead to possible desynchronizations, with unpredictable effects. In such a case, we suggest to link the first output with this acyion, and to synchronize the others linking them with the normally open auxiliary contact of the output that will toggle.
 energize	Energize the output (contact will close)
 de-energize	De-energize the output (contact will open)
 set merker	Set a merker. Clicking on the icon, a box pops up, where it is possibile to click on the number of merker to set or reset.
 set timer timeout	Set (or cancel) a timer timeout. Clicking on the icon a box pops up to let specify the timeout or if desired to stop (cancel) the timeout. After the timer has been linked this way, it is shown among the input events. The event will be fired when the timer will timesout. A timer can be linked to itself, to reiterate a cyclic timeout to infinite. This way it is possibile to set a first delay time, and then a cyclic delay time.

How to make a link

Click on the event icon, of the input of the desired object. The icon is bordered in red, to enhance that it is selected; then click on the output action icon of the desired object, or the timer icon, or a merker, or a function.
As soon as you click on the output icon the new link appears to the right summay pane.
If the output action requires to be set, before the link is created the set dialog box pops up. You must enter the required setting and hit ok to complete the link.
If has been clicked on the ladder constructor icon (in the inputs pane) as soon you click on the output action the ladder designer will pops up to promt you design the desired ladder expression. After you hit the OK button the link is created.

The links summary behave as a ladder, electrical diagram.

It is possibile to link an input event with more than one output action. In this case, press the CTRL key so that the input selection is kept alive until you press again the CTRL key.
It is also possible to link different events of a same input with the same action on the same output, and all the other combination, with a great flexibility and power.

Timered links

If an input event is linked with a timer, a timeout set is required. Then in the input pane it will be shown the new timer event. Is then possibile to link this new event with other outputs, which action will happens when timer will timesout.

Instable links

Some combination can result instable, because by a logical point of view they are not sustainable.
Example, if a link is made between a normally open status change of an input with a timer, an instable condition is priduced, because it is not well defined when to set the timer, infact since the input contact will be closed the link could produce anytime a true condition to set the timer (contact is a status, not a transient) causing an instable behaviour. MaticStudio warns whit a message when it detects instable conditions, helping you to make stable and reliable programs.

Deleting a link

To delete a link, click on the summary pane on the item to delete, and select the item Delete from the menu that pops up. If a input item is selected, all the following branch is deleted as well.

Modifying a link

To add new events or actions simply make the new desired link, as described above.
To change settings of timers or ladder expressions, click on the item to change in the summary pane, and select the item Modify in the menu that pops up.

Changing the evaluation order

The process of evaluation is executed in order as shown in the summary pane: from top to bottom, and by the case of happening of the event involved (the links not involved by events are ignored). To change the order of evaluation of the expressions click on the item to move below, and in the menu that pops up click on the item Move below.
This is useful for example if a ladder expression check for a merker, but such merker is set by a link that appears below. Moving the ladder expression low, changes the order, allowing to correctly detect the status of the merker.

Properties

Clicking on an item in the summary pane and then clicking on the item Properties in the menu that pops up, will show a dialog box with the properties of the link, showing all items that links that item. Double clicking on an item listed in the properties dialog box, the summary pane will move the cursor to show that item.
To keep the dialog box on top of the screen, while working on the summary pane, click on the Fix icon.

Bookmark

Working on large projects, with several inputs and outputs, can be useful to set some bookmarks to help moving forward and backward along the summary pane.
To set a bookmark click on an item in the summary pane, then in the menu that pops up click the item Bookmark. Clicking again will remove the bookmark. Click instead to the item Remove all bookmark to cancel all bookmarks.
To move between bookmarks hit the key F6.

Hiding menu

To hide the popup menu that appears clicking on the summary pane click on the item Cancel, or click elsewhere in the summary pane (not over an item).

Magnufy the pane

For a better sight of the links you can click on the icon  placed on the top-right of the pane itself. Clicking it again will shrink the pane to its normal size.

Two summary panes

To have two simultaneous sight of the summary pane, click on the icon  . To hide the secondary summary pane window click again the icon.

Virtual components

Embedding and modifying virtual components

Click the violet button Components, at its place appears the yellow button Links, and in the lower area is shown the list of the objects with components embedded. On top the toolbox of the available components appears as well. The toolbox shows per each component the icon, a 'i' button to show a guide of the component, and the title of the component.

Aside of the list of the components (in the toolbox) there is a list of the family of components. Clicking on the first item (all) will be shown all the components, while clicking on the name of a family will be shown only the components belonging of that family.

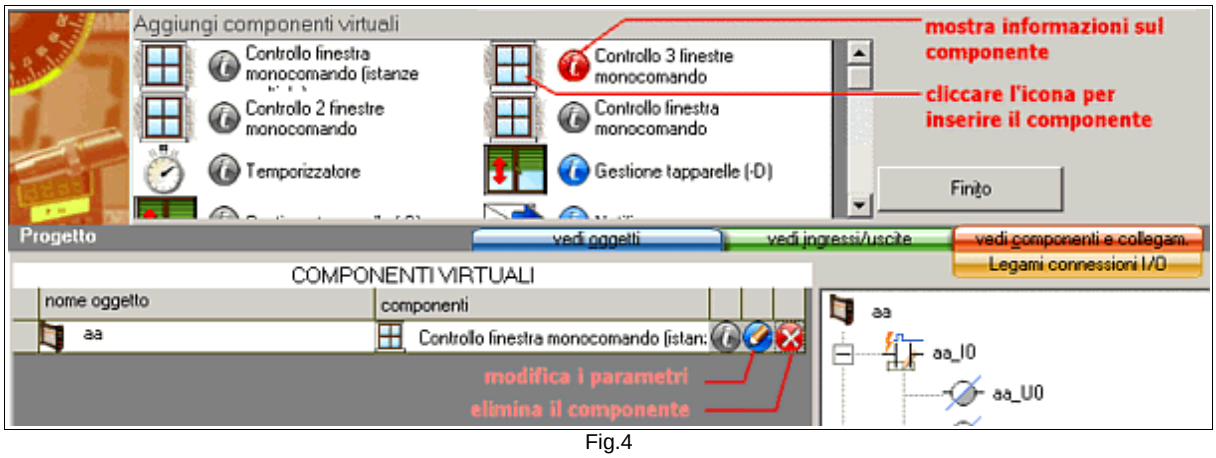


Fig.4

To embed a component click on the icon  so that is shoen the component toolbox, on top (see fig.4)

For each component is shown an icon, a button  clicking it will open the guide of the component (if the button is grayed the guide is not available), and the title of the component.

Click the icon of the component to embed.

A dialog box is shown, in which appears selected the required component, click the button Next.
Now it is prompt to select the object in which embed the component. At the next step are proposed one or more settings specific of the component to configure. (see fig.5).

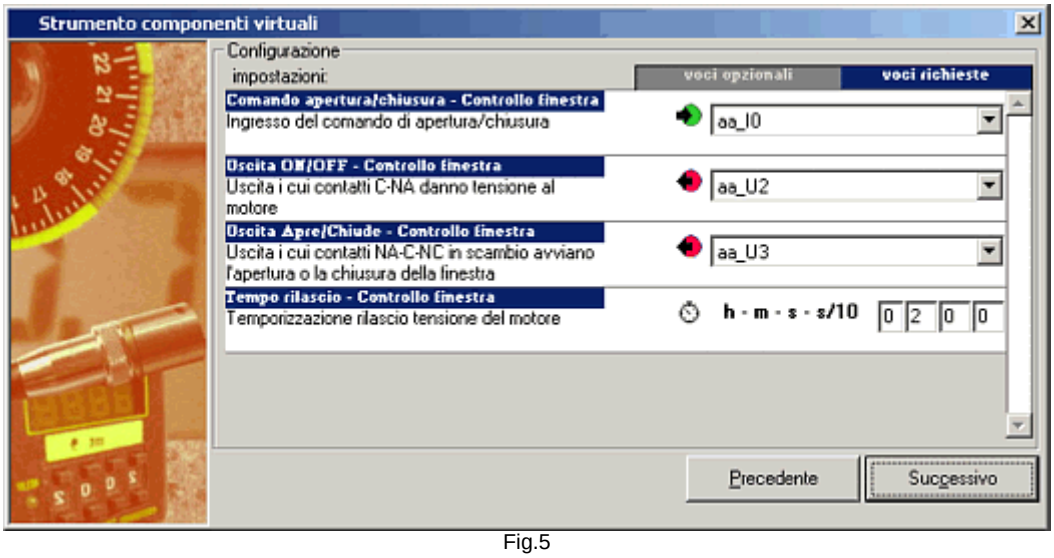


Fig.5

In this example si proposed a configuration of an input (green symbol) with which is activated a function of the component, two outputs (red symbols) onto which actuate the task of the component, and a timeout.
Each component has its own configuration which depends by the functions and tasks offered by the component istelf.

To complete the procedure of embedding the component, after the configuration, click further the button Next. Here you can enter a name to identify this component in the project, then hit Finish to embed the component.

The component so will be reported by the list of objects and the components embedded, placed under the toolbox. On the right of this list there are three buttons:  to open the guide of the component,  to modufy the configuration of the component, and  to remove the component from the object.

Embedding multiple instances

It is possibile to embed more than one component in the same object, not only different components but even the same type of component, in multiple instances. In example you might desire to have several counters of pulses, that reached different levels they activate different outputs.

When a second instance of a component is embedded in the same object, MaticStudio prompt you for a confirm to continue. Each instance has its own description name and its own configuration.

Removing a component

To remove a component click the button  .
Warning: It is not possible to remove a component which its functions still are in use in links. Must first remove the links that refers to it.
Warning: Removing a component that is embedded in more than one single instance, will remove all the instances of the same component from the object. It is not possibile to remove a single instance.

Automatic detection

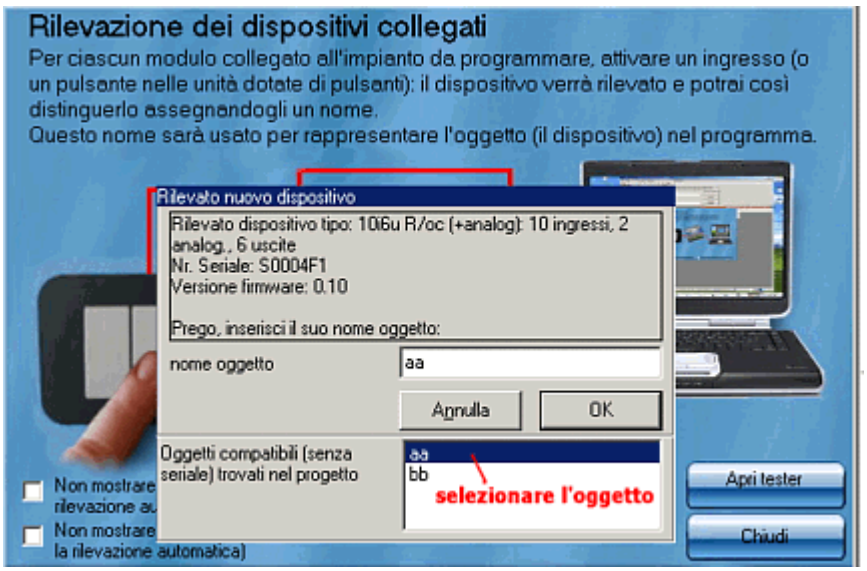
Grabbing devices

The grabbing function is mainly designed for domotic projects and in building automation, nevertheless it can be used even in industrial automation projects, keeping in mind that is required to toggle the inputs to let the device be grabbed.

If a project has been made without specifying the serial code numbers of the devices, it is possibile to make the grabbing procedure, following these steps::

1) open the project saved before
2) choose from the menu Tools the item Device detection

The following dialog box will appear. Be sure to have the computer connected to the network via the serial adapter, and the devices properly powered.



Now for each device to grab toggle one of their inputs, usually hitting a command button connected to it. For each detected device is shown a box in which are shown the type of the device, below, a list of all the objects in the project compatible with that device. So click on the object you intend to bind to that device and hit OK.

Remark. Can be detected only devices that has never programmed or are in protection status (yellow led is flashing each 1.5 seconds). To force to detect all the devices (even programmed and not in protection status) click the button Tester and from the tester dialog box select Devices & Diagnostic, so click the STOP button making sure to have selected the broadcast checkbox.

Autolearning

Autolearning

The autolearning function is designed for building automation or domotic projects, not for industrial automation projects.
It allows to have a vocal guided path that allows to detect and learn the devices and the main toggle function in a building (tipically for lighting).

Click the menu Tools, and then Autolearning.
In the dialog box that will appear, enter the names of the devices that are wished in the project, hitting the ENTER key to add a name that will flow to the list below. To remove a name click it from the list and hit the DEL key.

Once entered all the device names (the object names) to grab and learn, click the button Start autolearning process.
Be sure to have the computer connected to the serial adapter and this to the network, that the devices are properly powered, and no electric short were found particularly between the A and B channels and +24 and GND.
Be sure to have the volume of the loudspeakers high to maximun, place a walkie-talkie near to one of the loudspeakers, so that the voiced guide can be heard through the walkie-talkie allowing you to move freely along the building and follow the vocal instructions.

Remark: if your walkie-talkie has the auto-talk function (aka VOX), open the options (menu Tools, item Options...) and be sure to check the checkbox Auto talk signal, then hit OK.

Tools

Tester

Hitting the button Open Tester from the dialog box of the device detection, or by opening it from the menu Tools, item Bus, the Tester tool is opened.
The tester allows to check the status of the inputs and outputs of each device connected in the network (and programmed). The tested can me used also to temporarly set some outputs, while perfetioning the program. The device requires to be programmed, but can be sent even an emty program (just the list of the objects).

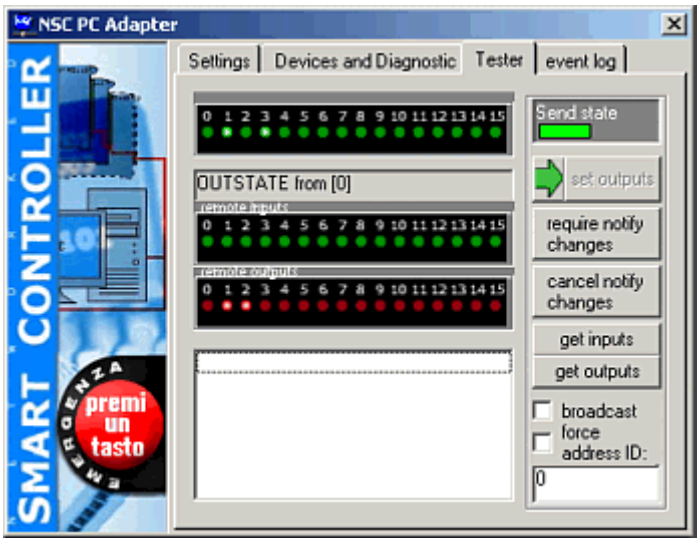
The adapter driver dialog box has the section Tester, so click on one of the buttons “I am in a safe environment” or “ok, proceed with safeties”

WARNING DANGER !

Use of tester can be dangerous – BYBUS refuse any liablility due by an improper use of the tester !

Warning! You can inadvertently energize different outputs than the intended one.
Electric shock danger
Remote energization of outputs lead the connected lines to be under voltage
Danger moving machines
Remote enegization of outputs could start connected appliances

Choose the button “I am in a safe environment” only if you are working in a laboratory, with the devices not connected under power line voltage.



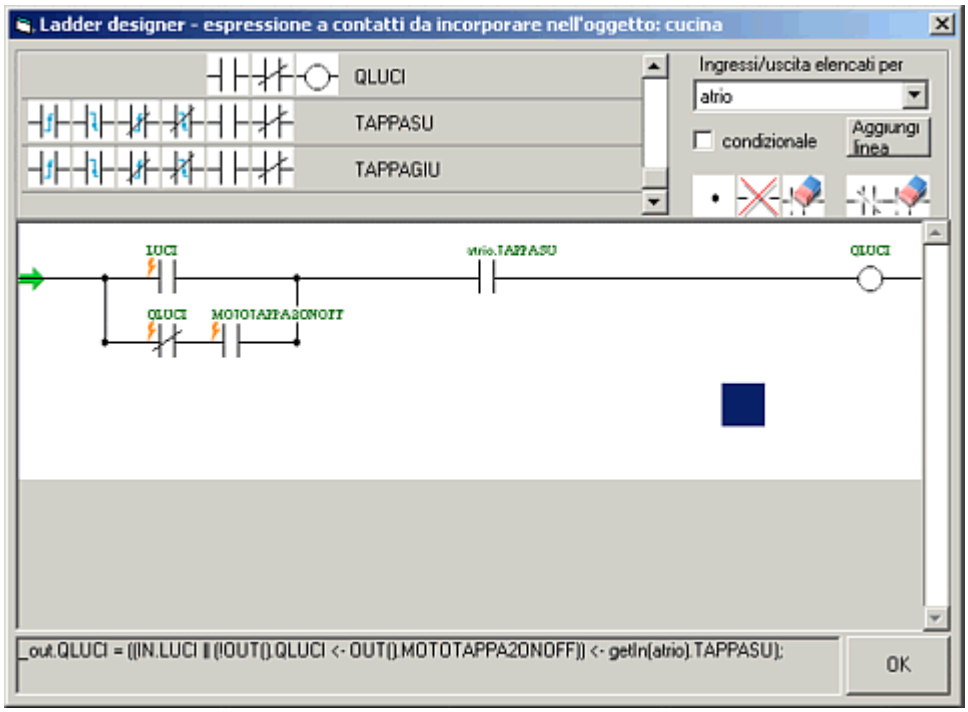
The tester has three bar of signal lights. The bar on top are the settings to send to a remote device: click on each light to toggle.
The bar below reports the status of inputs of the remote device.
The lower bar reports the status of outputs of the remote device.

Aside the picture of a emergency red push button indicates that you can hit ANY key of your keyboard to strike the emergency, which send a general signal to release all the outputs of all reachable devices. Beware to add a REAL emergency button to strike down power as well, since logic emergency is never sure 100%. Once the emergency button has been striked, it can be resetted only by rearming of the outputs clicking on the button with the green arrow.

To set the outputs, or read the remote status select a device from the list below. If the list is empty, click on the section Devices & Diagnostic, then click Map.

Usage of the ladder editor

To draw a contacts expression use the ladder editor. With it it can be designed complex expressions to check contacts status, transients, auxiliary output status, merker status, and timers timeout status.



To draw a contact simply click on the blank area, then choose a contact input or output and its related event (example a raise transient of an input), or choose a “coil” as output to energize/de-energize.

Outputs can be placed only in the last column at right. It can be placed more outputs one over the others that will be automatically connected in parallel.
Notice that is not allowed to enter outputs if the expression is conditional, and this is always true when entering a ladder expression in links, since the output is already defined by the link itself.

The contacts with the lightening symbol indicates that are active events, while if missing of such symbol indicates that the contact will be only checked if something else raise the event, but will not raise any event by itself.
Active events are all the ones of inputs (and outputs, merkers) located in the same object in which the expression is placed.

The dropdown list on the right allows to select an object of the project from which the contact should ben listed on the left pane.

The rubber clear a contact (the right rubber will clear the whole line). When placing contacts stacked one over the other they are automatically paralleled, however this is done in a way that may differ from the one desired. The join icon let define how the circuit should actually behave.
The X icon is to breaks the line, and can be placed only over the green arrow at the left. A broken line turn off outputs always, regardless of the rest of the expression.
To make more room in the design, click on the last low row: the space will be automatically expanded.

To add more lines click the button Add line. Each line is independent form the others.
Please, notice that this is only available to enter ladder expressions as text (in the advanced text mode). To add more lines in the visual mode simply add a new link expression.
In visual mode to add more outputs to the same expression simply re-use the expression (which appears as new event) and link it with the desired output.

When finished hit OK.

Note. Some expressions, such as to execute a function based by a combination of input contacts, requires only input contacts or are already preselected to act only as a determined event. In such a case if a contact not allowed is selected, or is selected an output, will be emitted a warning and refused the offending contact.

Diagnostic

Open the dialog box of the settings of the SFBP adapter and driver, by clicking on the Tools menu and choosing the item Bus..

In the dialog box, clicck the section “Devices & Diagnostic”.
This section allows to do several diagnostic operations:

Map	Executes the map of the devices connected to the filedbus network.
	Note. Only the programmed devices can be mapped, the virgin devices (never programmed) cannot be detected by the map function.
Abort	Click this button to cancel a running map
restart all	send a signal to reset and reboot all the devices connected
erase all	send a signal to erase all the connected devices: warning!
verify all	Execute a verify process of the status of all the devices connected and mapped, reporting error status (protection) if any.
	If verify version checkbox is checked, instead is made a report of the version and model of all the devices mapped.
quality check	Execute a quality check process, related to the data transmitted to the mapped devices. It gives a rate of network errors and loss of data.
stop	Stops the selected device, or all the devices if is selected the checkbox broadcast.
reboot	reboot the selected device, or all if the broadcast checkbox is selected.
clean error	Send a reset error (resetting the protection) to the selected device
read error	read the error register from the selected device
firmware info	read the firmware information of the selected device
reset addr id	send a reset of the logical address id to the selected device
force addr id	select this checkbox and enter a logical address id to send signals to this specific address

MaticStudio

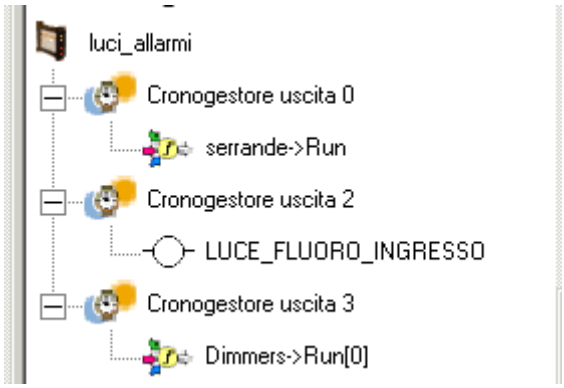
Scheduler Guide

Introduction

In the NSC devices equipped with scheduler you can set of operations that run on the basis of a daily programme on weekly cycle.
These operations can be any of those possible through links feasible with MaticStudio. There are more operational functions, through advanced programming but will not be examined in this document.

How it works the scheduler

The scheduler checks every 1 minute if there are scheduled events at a certain time of day and for a certain day of the week.
If it is an event, this will produce the effect determined by the link as it was made in the links panel in MaticStudio (see figure).



The scheduler provides the ability to run an event between the hours of zero (the first minute of the first hour of the morning) to 23:59 (last minute of the last hour of the day).

The scheduling restart the following day, relatively to the settings made for that day.

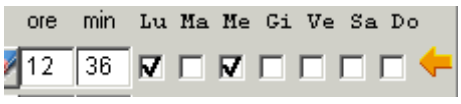
Types of schedule

The scheduler can handle up to 4 types of scheduling at the same time, further they are also distinguished between analog type scheduling, and logical type scheduling:

- **Analog type scheduling**
  They are on the determination of an analog value, for example to make the programming of the daily-weekly temperature
- **Logical type scheduling**
  They allow you to enable or disable logical outputs or to call functions, and offers up to 8 channels of activation for each time slot.

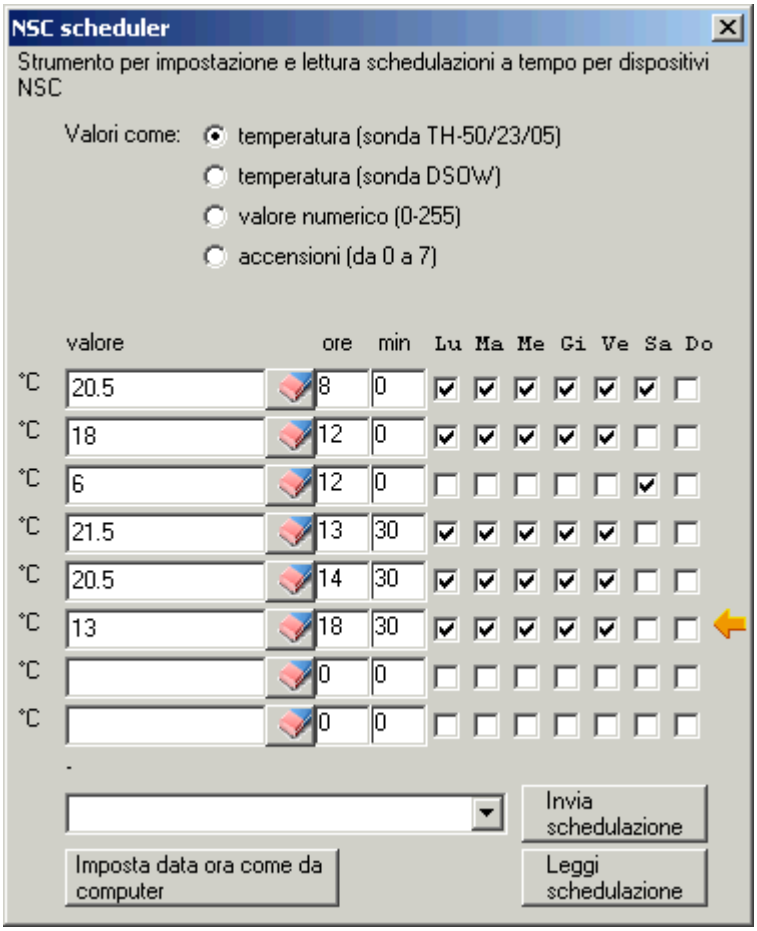
Time settings

All types of scheduling can be made on different time slots for each day of the week. For each time slot is possible to establish that it applies to one or more days of the week



Method of activation for analog scheduling

The mode is simple: when the time provided in daily programming expires, it is set the new analog value set for that time slot.



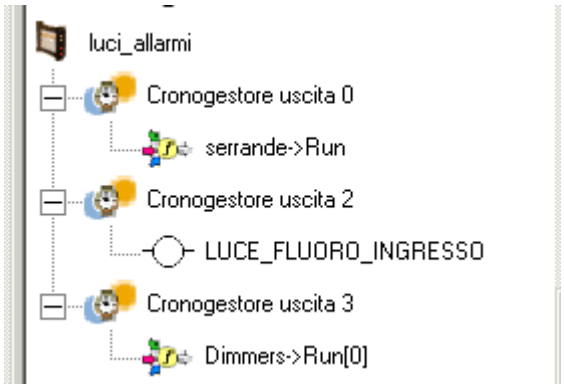
The picture on the right shows an example of setting performed by MaticStudio, and in the picture below, the same approach as seen through the DomoServer.

This example set the temperature of 20.5 C at 8:00 every day except Sunday, a new setting, 18 C is established at 12:00 hours every day except Saturday and Sunday, setting change back at 13:30 to 21.5 C and then until 14:30 and finally at 18:30 for all days except Saturday and Sunday.
Notice that on Saturday it is set to 6 C at 12:00.
Look at the example and notice that the setting 6 C does not act on the days from Monday to Friday, even though it is in the same time slot of another, different setting (18 C, 12:00) that instead acts from Monday through Friday. The Saturday setting therefore will remain valid until the day on Monday, at 8:00, when a new set is scheduled. Similarly, you can tell that by 18:30 hours from Monday through Friday, the temperature is set at 13C and so remains until the next new event, or the following day at 8:00 hours (except Sunday).

Method of activation for logical scheduling

The logic scheduling may trigger outputs or functions, and on the basis of these two types of activity operates in a different way, offering a wide range of operational possibilities.

In logical scheduling are available 8 channels (or sections). Each channel exposes an event, respectively numbered from zero to seven, (see figure ***Cronogestore uscita***...) that can be linked with outputs or functions (see figure).

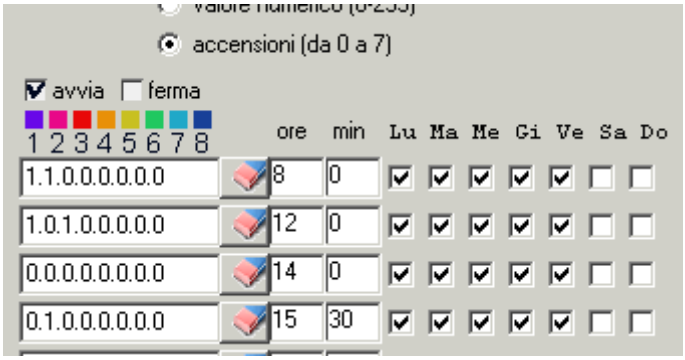


example of links of events triggered by logical scheduling

Note that events are numbered starting from zero, which correspond to channels that are numbered from 1 to 8, to comply with this as channels or sections are seen by the final user in the presentation shown by the DomoServer.

Bonds of scheduling events with action to energize outputs
(as an example see in the picture above **Cronogestore uscita2 > Luce_fluoro_ingresso** *)*

With this operational mode each channel produces the corresponding event, and is always effective when there is a setting to 1 (active), also has effect (causing shutdown of output tied) when occurs a transition from 1 (active) to zero (off), but has no effect when a setting 0 (off) follows a previous 0.

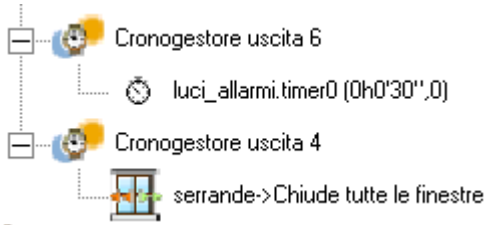


In this example, suppose that all channels are linked with the action energize outputs.
At 8:00, the channel 1 along with channel 2 would be activated producing to energize the outputs related to its events, while channel 3 will remain inactive and will not produce any effect.
At 12:00 the channel 1 is confirmed, while channel 2 is turned off, and the output will be turned off; channel 3 is activated instead so its output is energized. At 14:00 nothing happens to channel 2 (if for example the output had been turned on somewhere else, here would not be turned off), while channels 1 and 3 result to turn off. Finally at 15:30 will only activate channel 2 while there will be no effect on channels 1 and 3.

Bonds of scheduling events with action to de-energize outputs
This option is possible but must be managed carefully. Indeed, the output is turned off either when you have an event in activation (1) both when then there will be the next deactivation (0)!

Bonds of scheduling events with action to toggle outputs
This option is not recommended, as it will switch to every setting to 1 (active) and any setting to 0 (disable) if the first was to 1.

Bonds of scheduling events with functions
(i.e. see in the picture: ***Cronogestore uscita4 > Chiude*** *)*

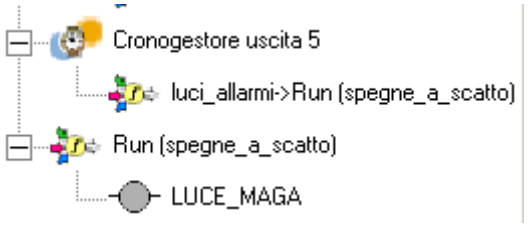


With this operational mode each channel produces the corresponding event only when it is set to 1 (activation), while the setting to 0 (disable) is ignored.
Notice that the link to the timer (see picture) is managed as a link function for the event scheduling point of view, in other words it will set the timer delay when the event happens on the 1 (active). From that moment the timer starts its countdown and the expiration of the time set in turn unleashes its event on timers.

Taking the example of the first, but this time referring to cases where events are linked to scheduling functions, which will at 8:00 will be activated functions related to events corresponding to channels 1 and 2 (event cronogestore_uscita0 and cronogestore_uscita1 respectively).
At 12:00 hours the event will be repeated for the channel 1 and there will be the event on Channel 3.
At 14:00 hours nothing happens.
At 15:30 hours will be performed the function linked to the event cronogestore_uscita1, corresponding to Channel 2.

Putting together different methods

By combining together the different methods, as seen previous, you can get edge operations (tie functions) or state (bond with output up).



The virtual component Proxy allows you to perform actions on status controlling them as a function. For example (see figure at the side) is possible in this way tie the execution of proxy-out to shutdown an output, and the proxy-in at the scheduler event. In this way the output can be switched, turned and managed by other functions and events, but at the scheduled hour will be properly turned off by the scheduler.

Appendix

Solutions to problems

Compilation Error: no more timers

Occurs if they were used all the timers of an object and in the same object were embedded components that in turn uses timers. Solution. Try to review the logic.

Error object not found

Occurs if you do not have locally one or more virtual components, such as uploading a project carried out on a computer other than where you run the compilation / programming.
Solution: Install the missing virtual components.

It can also occur if you have altered the text source for the inclusion of an object code of virtual components (Directive # include <*codeobject: nomeoggetto>), in this case rewrite the correct reference.

The functions of virtual components not occur

If after programming devices, the functions provided by the embodied virtual components does not occur, verify that the directive inclusion of components:
include <*codeobject: nomeoggetto>
was not inadvertently deleted from the text source code.

Exausted memory

Occurs if you have too many functions included in the text source, or too many components virtual, or too many links in a certain object. The limits of memory depends on the device, for an assessment indicative see Limits.
Solution. Try to split the functions on two items or consider the replacement of the type of device with another with more capacity.

Area text exhausted

Occurs if the text source is excessive, the presence of many connections (inputs and outputs) contributes to causing the problem. If there are many connections try to check the "enter define inputs / outputs as a external file included" in options (Tools menu> Options). If the problem still has not solved try to replace part of the source code with # include directives.

Send program Error

Rare cases of data collisions may be the source of this problem, in that case is resolved simply by repeating the operation.
If the problem occurs systematically or with great frequency instead needs to be done the following:

- If occurs when sending a program, ensure that there are no inputs that switch quickly on any device;
- Check the network, the quality of cable, and which were not reversed the wires of channels A and B;
- With long lines (> 400 m) make sure that they are not made too many configurations star, also in lines close to 1000 meters you should put terminators at the ends of the line (or stars).

Typically, the first case is the most probable, while the last relates to very long lines and with many connected devices.

Limits

MaticStudio (standard edition)

Number of objects: 128
Number di connections: 4096
Number of virtual components: limited by the memoria of the computer
Number of the embeddable virtual components in an object: depends by the components itself, by the size of memory available in the physical device, and by the resources used by each one of the embedded components: approximantely from 1 to 30.
Maximum size of text: 2MB
Maximum size of source text code: 10MB

Compiler
Compression ratio of machine code: from 14:1 to 26:1 (strings not included)
Number of classes: limited only by the available computer memory
Number of defines: 4200
Size of the generated code machine: limited only by the available computer memory
Size of the deployer code machine manager: limited only by the available computer memory

Devices

Note: The information depends on the device, refer to the technical file. The values below are typical.

Number of timers: 8
Number of local variables: 20 byte*
Number of global variables: 32 byte
Number of function arguments: 12 byte*

* sum must not be greater then 32 bytes

Memory:

Typo of memory	***Device version***	***Size***
User program	Versions S	500 bytes
User static data		-
User program	Version 0.x	980 bytes
User static data		20 bytes
User program	Version 1.x	from 2048 bytes to 3092 bytes
User static data		from 300 bytes to 980 bytes

Advanced programming – Usage of text editor

MaticStudio allows you to achieve sophisticated distributed programs, acting on a single program.
The text editor allows you to precisely define the functions you want to run on devices and offers a wide range of functions.
Creating automated project, MaticStudio prepare in the text a framework of the names of objects, the defines of the name of the connections, and the inclusions to embed the virtual components, including the links between I / O.
If you remove these references the program is unhooked from the project structure, but it is possible to create, compile and submit a program to remote devices.

For more information about the programming language C simplified, for the NSC, refer to specific documentation or literature.
Here are given only the particularities linked to MaticStudio, and is a summary of the major key structures and limits.

Collision of events
If you use virtual components and complete the program by acting on the text, take care to the risk of collide with events already used by components.
MaticStudio does not currently support the sharing of an event between virtual components and text.

Key structures

object	specify an object
class	specify a class

Example:

```
class class1 {
  pubevent _test(byte value, byte sender) {
     setOut(value,0xffffu,this);   // set outputs
  }
}

object obj1:class1 {  // implements class1
}

object obj2 {
  event click(int Clicked) {
    obj1._test(1);  // invoke _test() implemented by class1 in obj1
  }
}
```

Events

click, longclick, input, outchange*, net, timer0...timer7, analog0**, analog1**, zeroEncoder***
See API (Application Programming Interface) documentation for a complete list)

* available only in firmware version 1.x and where outputs are available
** available only in devices with analog inputs
*** available only in devices with encoder inputs

.....

First published, Febbraio 2004
Reprint: Gennaio 2006

Information and data are available for printing and publishing houses specialized in information technology.

BYBUS srl
IT 36075, via S. Bernardino 44
Montecchio M., Vicenza, EU
info@bybus-italia.com
www.bybus-italia.eu – www.domosmart.eu – www.automazion-domotics.com
member of : www.smartcontroller.org