

TUTORIAL

PLC PROJECT

How to work with plc's in PC|SCHEMATIC Automation.
Developed by PC|SCHEMATIC A/S

PREFACE

This booklet contains a description of how to document a simple plc installation with PC|SCHEMATIC Automation.

You will see a project in which you place an appropriate plc from the database and subsequently load an IO-list with various data.

You will see a project in which the plc is not in the database, instead you place symbols with your own data. Subsequently the plc can be updated with an external IO-list.

The first two projects look like PLCDEMO.PRO which is one of the demo projects that comes with the program.

The possibility to re-address plc-symbols in large projects where many plc-symbols come from subdrawings has become significantly different from version 14. These new options are also covered. Try the functions yourself with the subdrawings that you find in the PLC-folder.

PC|SCHEMATIC has a series of booklets covering different subjects:

- Quick Start
 - Shows the different parts of the user screen, the tool bars, where to find the drawing objects
 - Use it as a reference guide when you start drawing on your own.
- Tutorial – House Installation
 - Shows you how to make a house installation project like PCSINSTDEMO.
 - Make your own pickmenu, edit your lists and learn how to change database
- Tutorial – Motor Control
 - Shows you how to make a small control circuit, where all components are found in the component database. The finished project contains electrical diagrams, panel mechanical layout and various lists.
 - The project looks like PCSMOTORDEMO1. In this way you can always check that you have been through all steps.
- Tutorial – PanelBuilder
 - Document a household panel by using the PanelBuilder module
 - Create front page drawing, power dissipation calculation and single line diagram
 - Declaration of Conformity, CE-label and other markings can be printed directly from the program
- Quick Start – Plc project
 - Shows you how to document a small plc.
 - The project looks like PCSPLCDEMO1.
- Quick Start – Reference Designations
 - Learn to create and apply reference designations in a project.
 - Learn to work with wire numbering
 - See how to make graphical plans
 - You work with projects like PCSDEMO1, 2 and 3.

When you read the booklet you can follow the example. When you have worked your way through this you have tried the most common functions in the program, and you will be able to make your own project.

All examples are based on demo components from the PCSDEMO database.

The examples are made in Automation 14. With a few exceptions all examples can be made in earlier versions.

There are text highlights in the book to indicate further explanations about some of the program functions. You don't need to read these explanations to make the project.

Why a special PLC function

Like all documentation plc-projects must “only” contain component and terminal names. Additionally, plc-addresses and descriptions for the individual addresses must be included in the documentation. This can, of course, be done manually, that is writing the data as you meet them in the project. Alternatively, you can use a special symbol type – plc-symbol – which allows you to load an external IO-list which contains addresses and descriptions.

BEFORE YOU START

The descriptions in this booklet are based on the presumption that you are fairly familiar with the program. If you haven't worked with PC|SCHEMATIC Automation we recommend that you make the exercises as described in the booklet **Tutorial – Motor Control**.

CONTENTS

PREFACE	2
WHY A SPECIAL PLC FUNCTION	3
BEFORE YOU START	3
START THE PROJECT	6
PROJECT WITH PLC FROM THE DATABASE	7
PLACE THE PLC REFERENCE SYMBOLS	8
PLACE THE PLC INPUTS	9
PLACE THE PLC OUTPUTS	10
I/O ADDRESSES AND DESCRIPTIONS	10
OBJECT LISTER	10
READ THE PLC I/O LIST	12
PROJECT WITH A PLC NOT IN THE DATABASE	14
PLACE REFERENCE SYMBOL	14
I/O ADDRESSES AND DESCRIPTIONS	15
PLACE OUTPUTS FROM THE FIRST PLC	15
PLACE THE OUTPUTS FROM PLC 2	17
FINISH THE PROJECT	17
PLC PROJECTS WITH SUBDRAWINGS	18
START THE PROJECT	18
EXAMPLE 1	18
EXAMPLE 2	20
I/O STATUSTYPE	20
ADDRESS PREFIX	21
LOCKED ADDRESSES	22
REFERENCE DESIGNATIONS AS FILTER	22
RE-ADDRESSING REFERENCE SYMBOLS	23
A LITTLE MORE ABOUT PLC SYMBOLS AND COMPONENT	24
PLC-SYMBOLS	24
PLCs IN THE DATABASE	25
SIMPLE PLC IN THE DATABASE	25
CREATE PLC IN THE DATABASE	26
FORMAT FILE FOR I/O LIST	27
INDEX	28

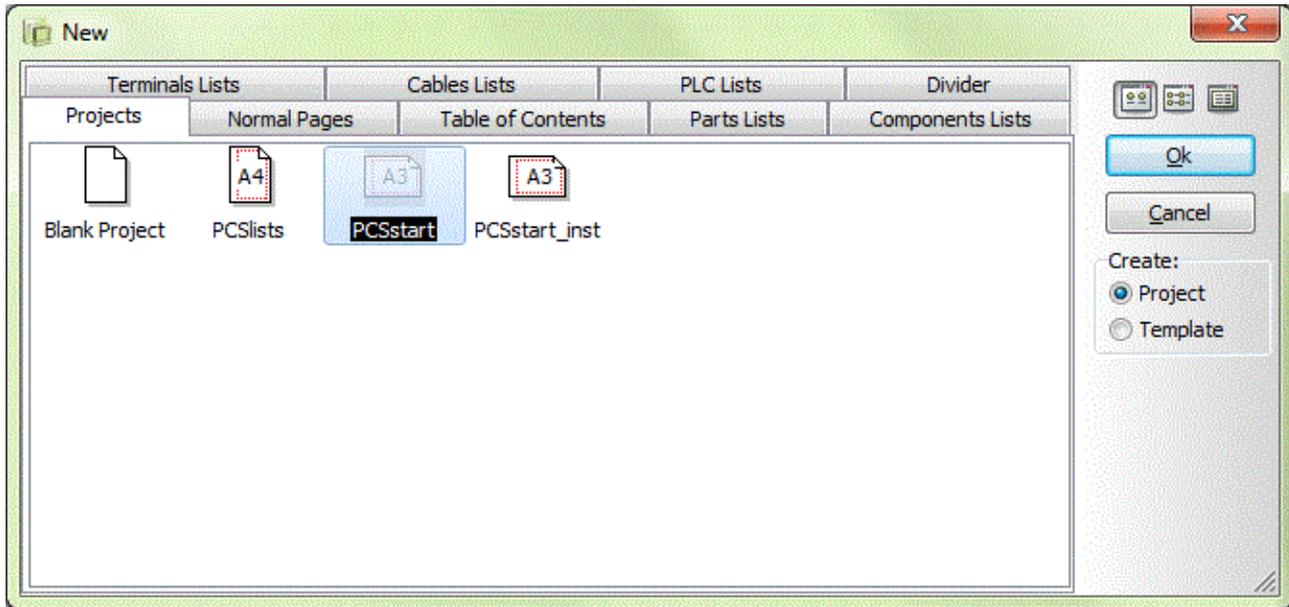


MY FIRST PROJECT

In this part you can see how to make a simple plc project. Three different methods are described. The first project looks like PLCDEMO1.PRO, which you can find in the PROJECT folder.

START THE PROJECT

Go to Files|New and select the template PCSstart.



You may give the project and the customer a name, which will be seen in the title block.

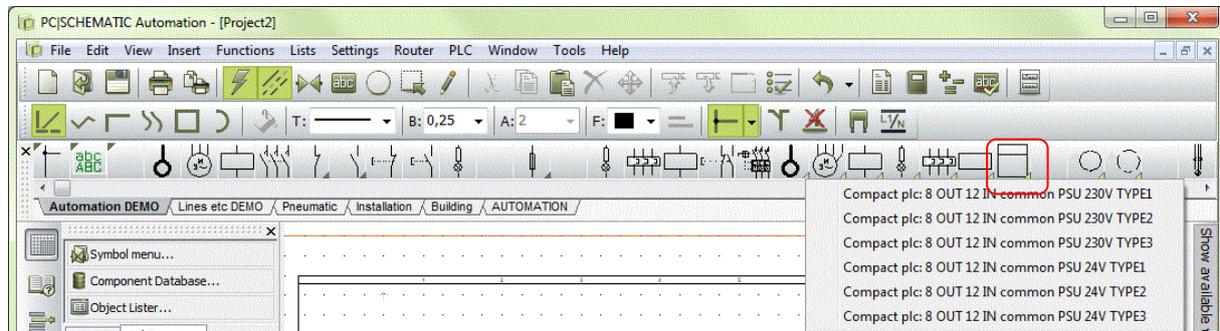
In this exercise you only make the control circuits, so you miss a couple of pages in the project: start by selecting the tab Subdrawings in the Pathfinder window. Open the folder STANDARD\DEMO\PLC. Drag a PCSA3BASIC (empty page) and a PCSL1 into the project after page 6.

Use the Page menu to renumber the project, you can find a description in the booklet Tutorial – Motor Control in the paragraph The last finish.

PROJECT WITH PLC FROM THE DATABASE

Go to the empty page (page 7).

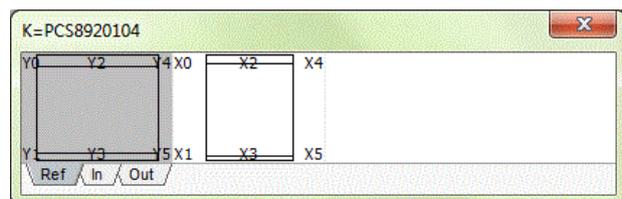
Select a plc in the pickmenu Automation DEMO: Select the top one.



Now you get a window with three tabs, ie:

- Ref (reference symbols)
- In (inputs)
- Out (outputs)

When you fetch a component from the database you will see all (electrical) symbols belonging to the component. Plc-symbols are grouped as there is a large amount of them for a given component.



A little info about the symbols

When you work with plc's in PC|SCHEMATIC Automation you use two different kinds of symbol types:

- Reference symbols
- Plc-symbols

Reference symbols are used to show the plc's functions and where the individual in- and outputs are located in the project.

Plc-symbols (IO-symbols) are symbols that are placed for each adresse or in some cases for several adresses.

Both symbol types show the plc's connection points, that is the terminals used for connections to wires and other symbols in the project and in the control panel.

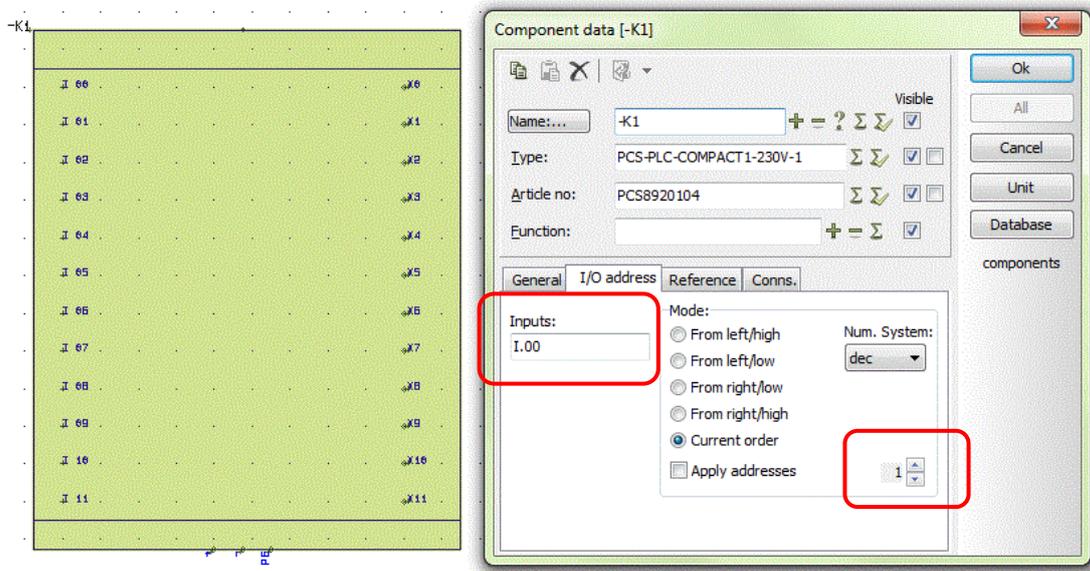
In the program it works in the way that there is a reference between the two, which means that you can "jump" between the reference symbol and the IO-symbol and the circuit that carries out the function.

Finally, you have the option to import data directly to the connections points, a subject you can read more about on page 12 in this booklet.

The plc'en can also contain "normal" symbols, ie symbols for power supply and bus connections.

Place the plc reference symbols

Start by placing the reference symbol for inputs on your plc (the right one):.

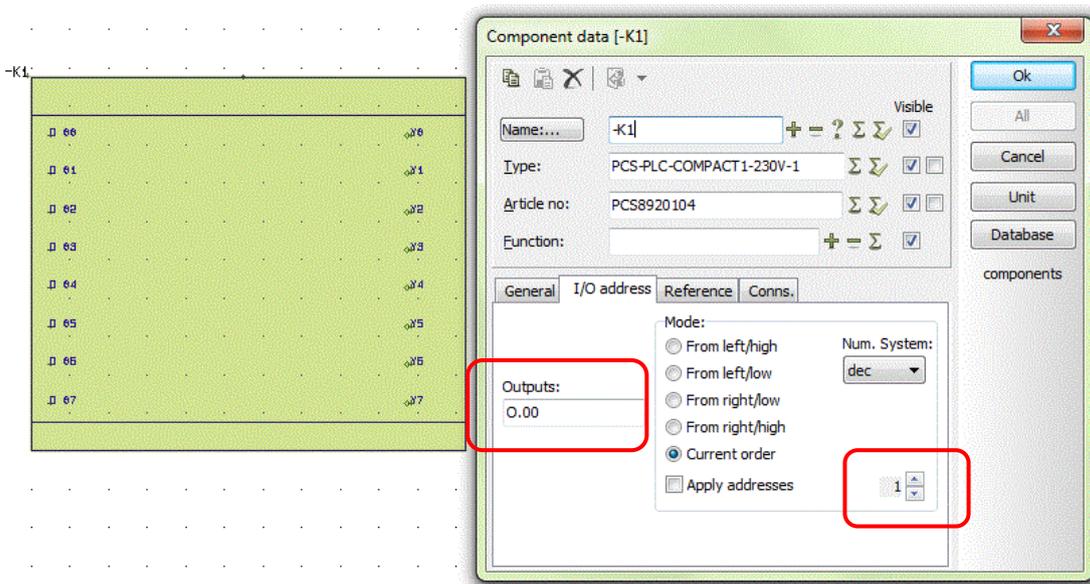


Name the symbol –K1.

On the reference symbol you find a tab – I/O addresses – in which you type the first address of the symbol. Here you type I.00.

As the symbol has 12 inputs on it and you count by "1" it means that you have told the program that addresses on this symbol goes from I.00 to I.11.

You must also place the reference symbol for the plc's outputs. You find the symbol in the Show available window. When you have placed the symbol, you must double-click to open the dialog. Then type the start address for the symbol, which means that the symbol covers the addresses O.00 to O.07.

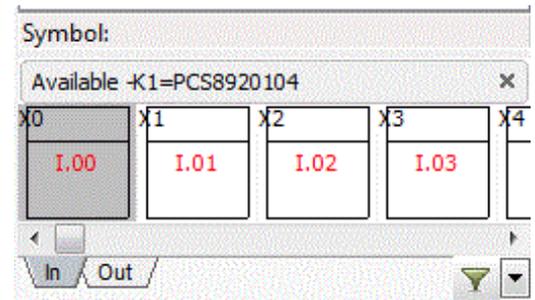


Place the plc inputs

Now place your input with the new adresse in the documentation.

Show available

In the Show available window you can see the addresses for the individual inputs. You select your input based on this address or based on the physical connection name.

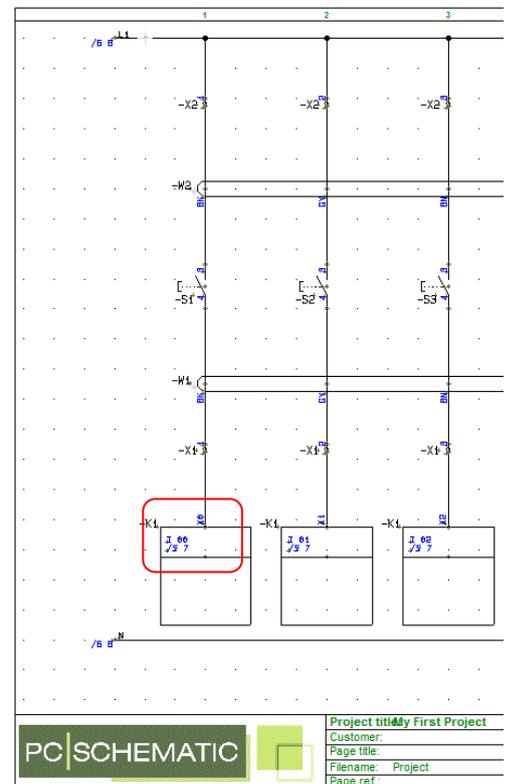


Go to the next page – page 8 – in the project.

Place the first eight input on the page in current paths 1-8 and start by connecting them to the power supply. Then connect the inputs to push-buttons, terminals and cables as shown here. The selected components are found in the pickmenu Automation DEMO.

Zoom into the plc symbol and see that automatic reference.

This reference is between the address (connection point) on the plc symbol and the same adresse on the refence symbol.



Place the plc outputs

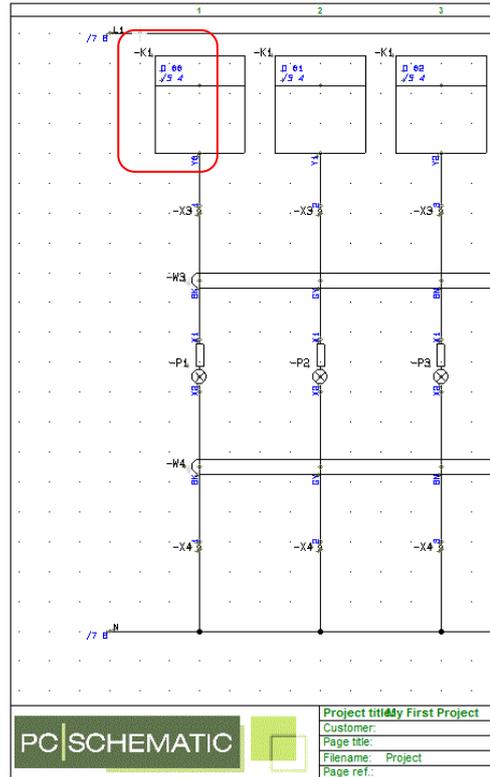
Now use the Show available function again, this time to select the individual outputs.

Place them on page 9 (if you kept all pages in the template). Connect to lamps, terminals and cables as shown here.

The selected components are green lamps, through terminals and 10-conductor cable from the pickmenu Automation DEMO.

If you zoom into the plc-symbol you can see that a reference has come up.

This reference is between the address (connection point) on the plc-symbol and the same address on the reference symbol.



I/O ADDRESSES AND DESCRIPTIONS

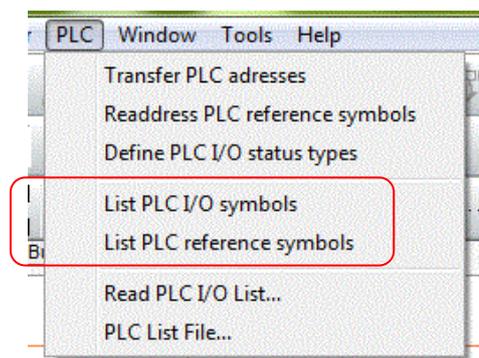
Now you have placed inputs and outputs in the project. However, you still miss descriptions for the individual addresses. There are different ways to get this information into the symbols:

- Open each IO-symbol (a lot of work and not efficient)
- Open each reference symbol (a little faster)
- Use the Object lister (ok with small projects)
- Import IO-list from an external file (efficient and easy)

Object lister

The Object lister is opened via F7 or through the PLC menu.

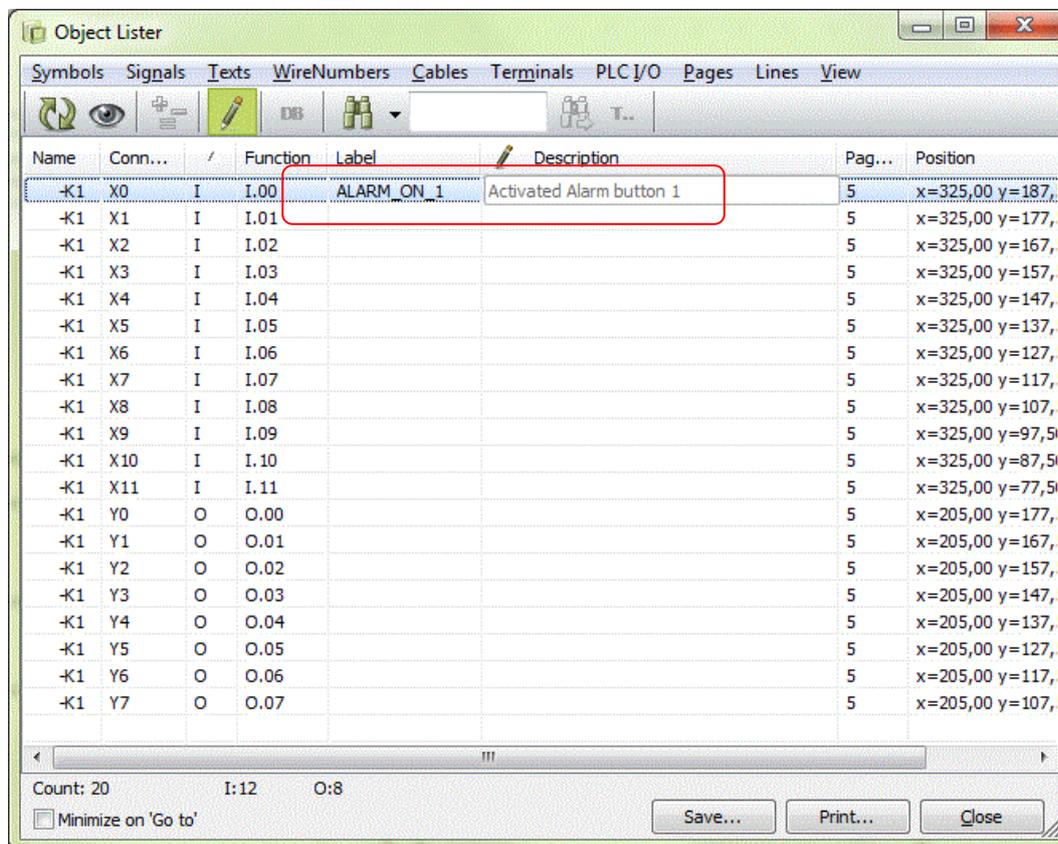
You can open either all defined addresses in the project, that is all addresses from reference symbols, or you can open all used addresses.



Here you have selected List PLC reference symbol.

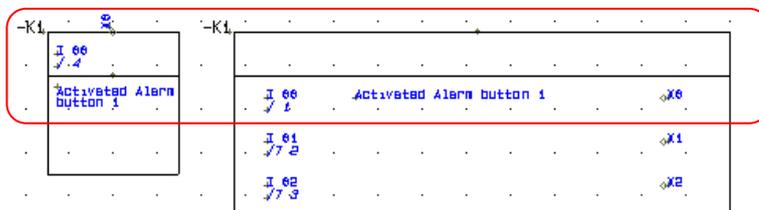
You can edit data in the Function (IO adress), Label (possible part of programming) and Description (which is part of the documentation) columns.

Start editing by activating the Pencil. Save the text by pressing Enter (Arrow down does not save the text!).



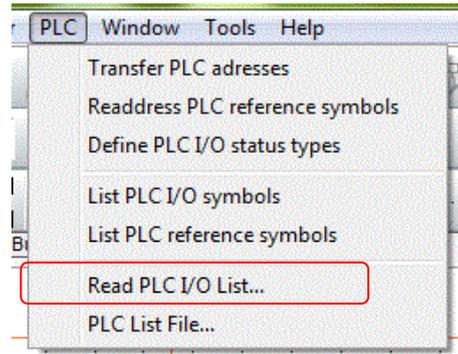
When you edit an object using the Objectlister you edit directly in the project as well. That means, that you should *not* press Save in the dialog when finished, simply close the window.

The text above looks like this on the reference symbol and the plc symbol, respectively:



Read the PLC I/O list

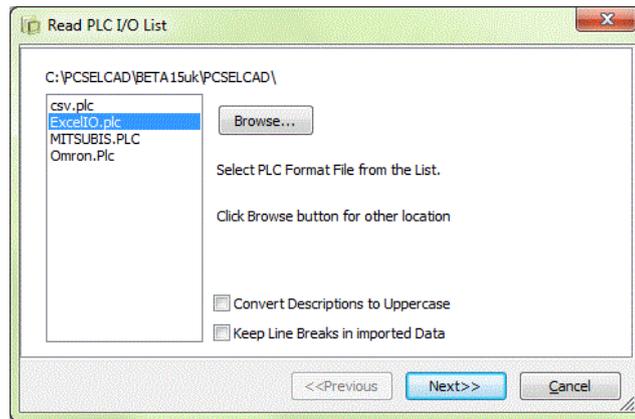
Another possibility to enter the right addresses and descriptions into the project is by importing the IO-list, which is part of the programming tools or in another way can be found in Excel or csv-format.



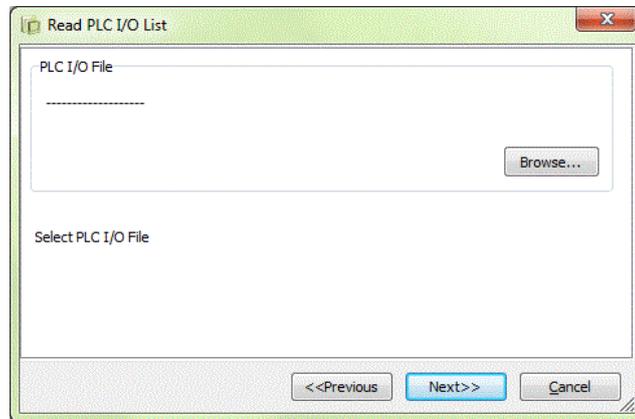
When you select the menu item above, the following dialog pops up:

First you select a format file. This file describes the import format, that is a file that “tells” Automation where and how the contents of the plc-list should be used by the program.

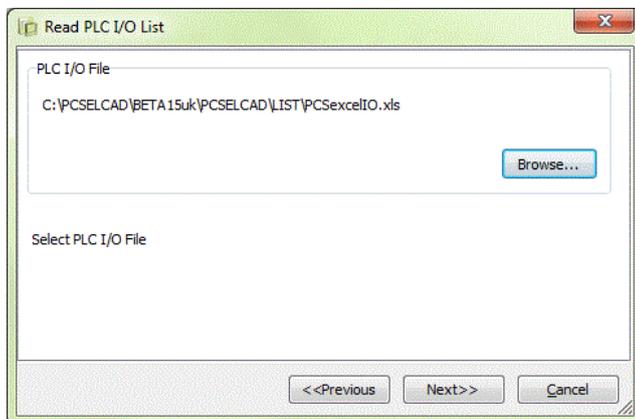
In this example you select ExcelIO.plc. Then press **Next >>**.



Here you must press the button **Browse**. You then enter the folder LISTS where you select the file PCsexcelIO.xls. The file contains IO-data for this project.

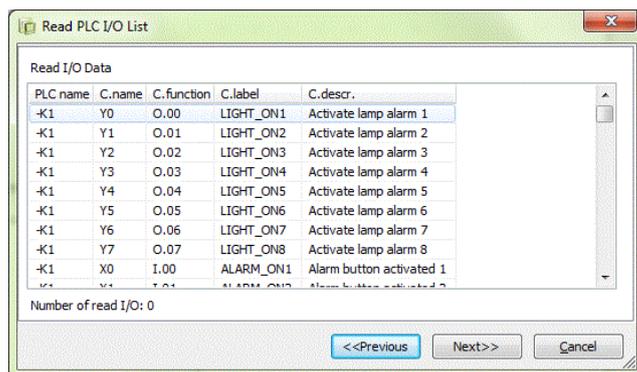


Here the file has been selected. Press **Next >>**.



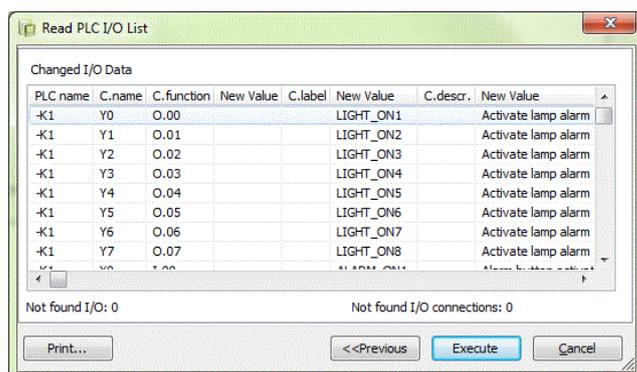
Here you can check the contents of the selected IO-list.

Press **Next >>**.



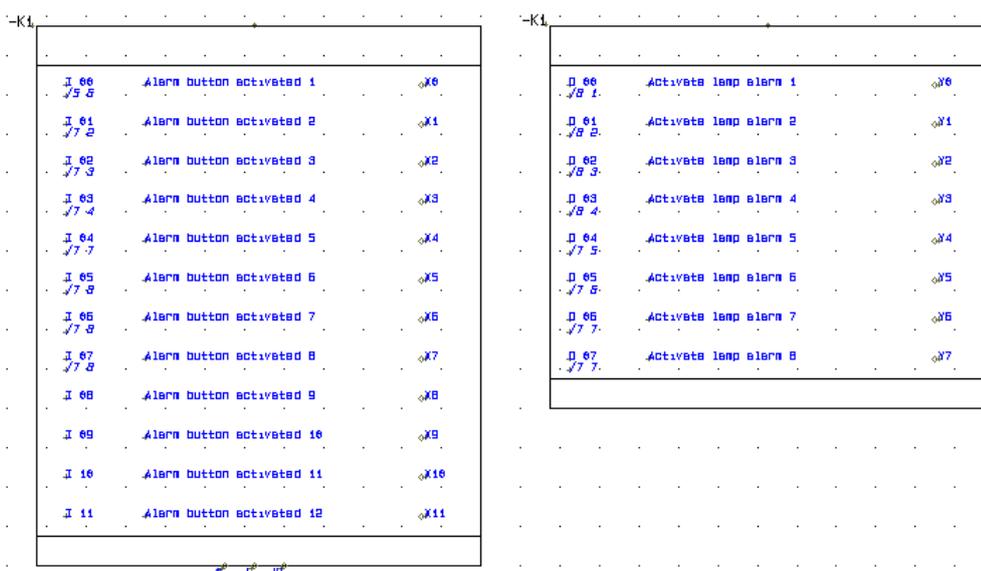
Here you can compare the new values of the IO-list with the existing values in your project.

Press **Execute >>**.



All addresses and descriptions are now updated in the entire project and you can see that on the reference symbol as well as the plc-symbols.

You can also see descriptions for addresses not yet placed in the project. They were part of the import, but as they are not placed in the project you won't see any references to their locations.



The finished project looks like PCSPLCDEMO.PRO which you can find in the program.

PROJECT WITH A PLC NOT IN THE DATABASE

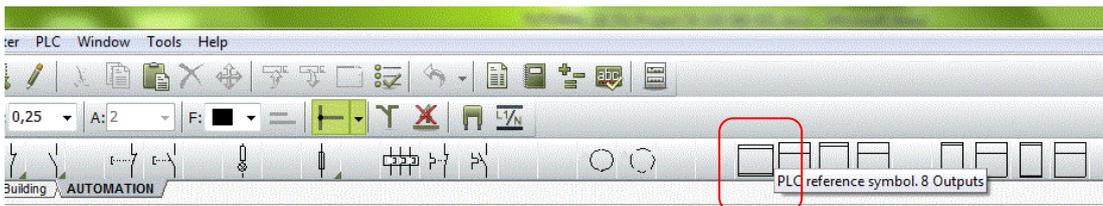
Start a new project as described on page 6. Go to page 7 (the empty page).

In this section you are going to make a project with non-database-plc's, but a plc where you use empty symbols from the pickmenu. The project contains digital as well as analog function which allows you to see how various set of symbols work.

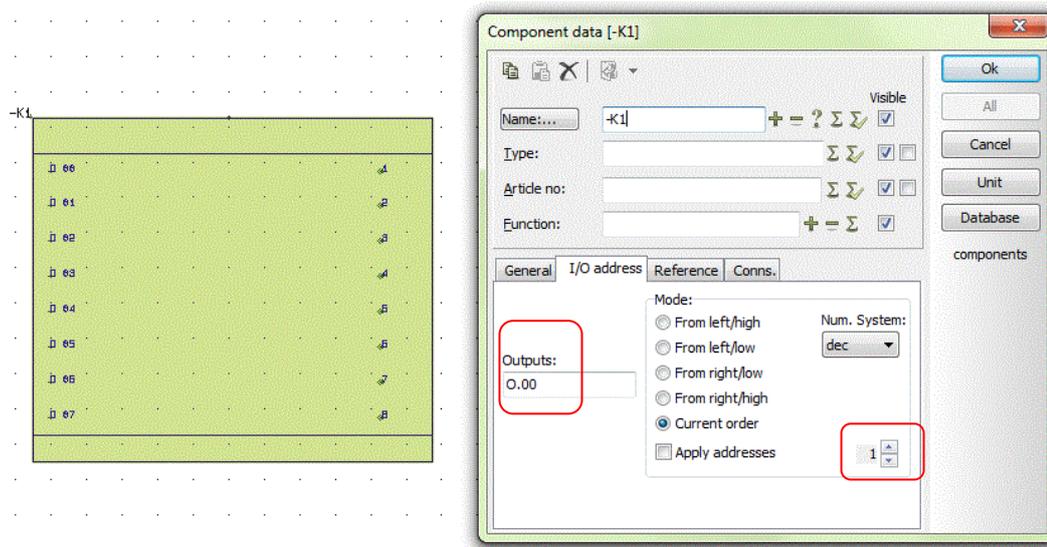
Place reference symbol

In the pick menu AUTOMATION you can find more sets of plc symbols.

Select the first reference symbol: "PLC reference symbol. 8 outputs":



Place it on page 7 in the project and name it -K1 and assign it the start address O.00:



Also place a reference symbol for outputs with individual power supply. Select the reference symbol "16 outputs with individual power". Name it -K2 and assign it the start address O.100.



Plc symbols in this pickmenu are paired: two sets of inputs and two sets of outputs. You can see the descriptions for each symbol if you place the mouse on top of it. The difference is with or without individual power.

The symbols must match to have connection points automatically synchronised.

When you place plc symbols in your project you actually double-place all connections related with inputs and outputs: they are on the reference symbol and they are on the plc-symbolsto which you connect all other components.

When the connections for all addresses are both places and a created properly, the program will automatically update all data on all connection points.

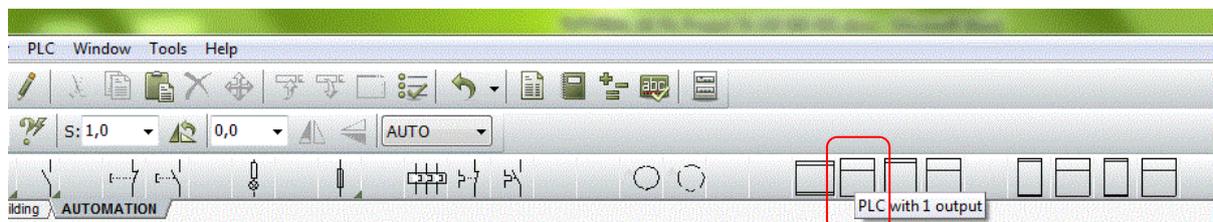
Read more about this in the section PLC-symbols on page 24.

I/O ADDRESSES AND DESCRIPTIONS

In the following paragraph it may easier to obtain an overview if you make descriptions for each output. You can see how to do that from page 10.

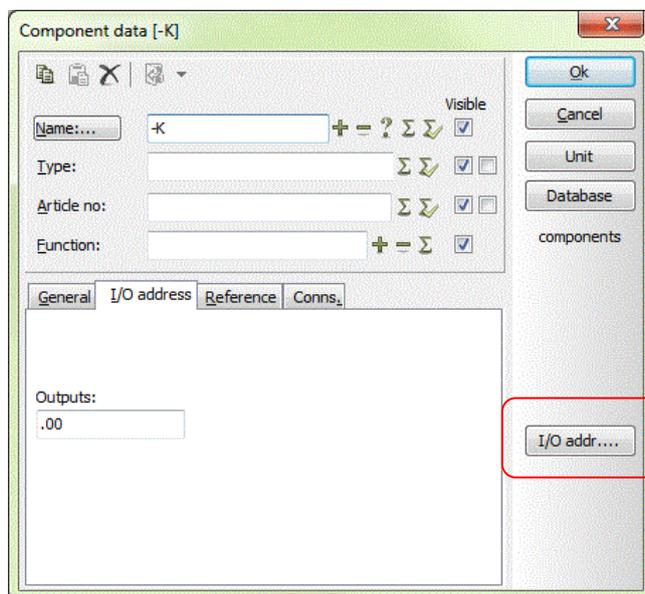
Place outputs from the first plc

Select the next symbol in the pickmenu "PLC with 1 output" and place it at the desired position in the documentation.

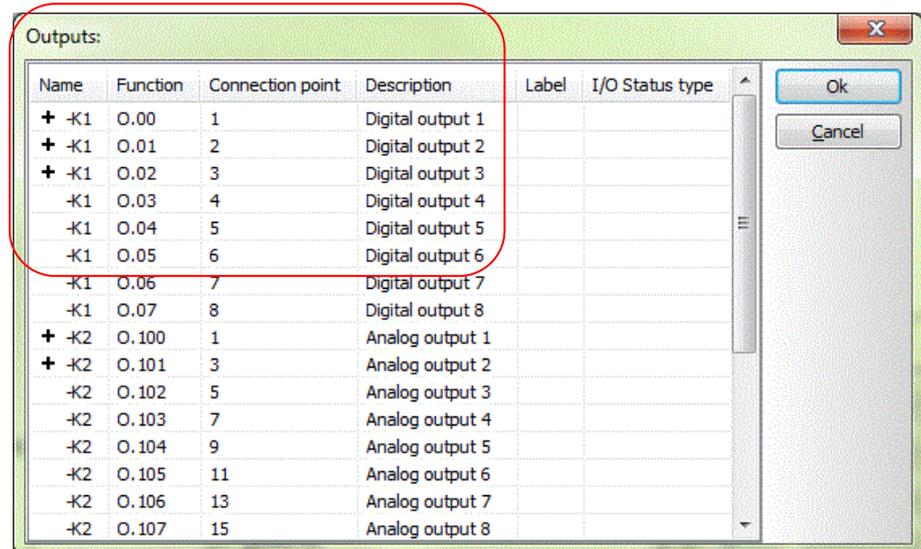


The PLC symbol must now be connected with the reference symbol:

The selected plc symbol fits with the first reference symbol –K1. You have defined a number of output addresses in the project which can be fetched via the button **I/O addr...**

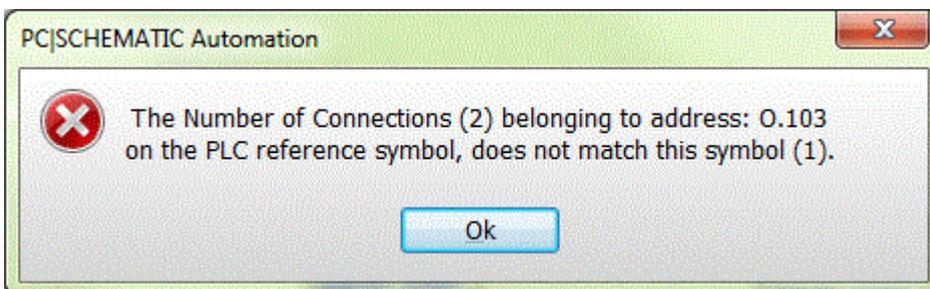


This window pops up when you press the button. In the window you can select the desired address (and description) for the symbol. Select the address by either double-clicking or by marking the address followed by an OK. Addresses already used in the project are marked with a +.



The plc symbol gets its component name, connection name, address and description through this selection. When you make a valid selection, you can see references on the reference and plc symbols.

¹If you select an address in which the symbols are not corresponding you will get the following fault message which tells you about mismatch between the selected symbols:



¹ Depending on program version, the warning might be disabled in the program.

Place the outputs from PLC 2

Fetch the PLC-symbol "PLC with 1 output and power" in the pickmenu.

Place it in the project and click the button **I/O addr...** which takes you to the same list as before.

This symbol matches the -K2 reference symbol, as each address has a plc-connection and another connection.

When the two symbols match, you will see that the plc-symbol gets both connection names, the address and the description from the reference symbol.

That means, that you can find all information about the address at both locations in the documentation and use the link to "jumpe" between the locations.

Address	Description	1	2
0.100 /.5	Analog output 1	1	2
0.101 /.5	Analog output 2	3	4
0.102	Analog output 3	5	6
0.103 /.5	Analog output 4	7	8
0.104	Analog output 5	9	10
0.105	Analog output 6	11	12
0.106	Analog output 7	13	14
0.107	Analog output 8	15	16
0.108	Analog output 9	17	18
0.109	Analog output 10	19	20
0.110	Analog output 11	21	22

Finish the project

You can now finish the project in the same way as you finished the first project. Then you can either type data for each address via the Object Lister or import an IO-list; all in all finishing the project is similar to finishing the first project apart from the Show available functions.

The two project types covered here can be made in Automation from version 10. The following and last example shows how to use plc-functions that have been released with version 14.

PLC PROJECTS WITH SUBDRAWINGS

The two methods described earlier are not always enough when you work with plcs. In “the real world” you often work with modules eg plc-inputs connected to various sensors, cables and terminals and corresponding outputs activating various equipment.

The disadvantage of working this way is that you don't know the necessary size of your plc, that is how many inputs and outputs, before you have finished designing the system. That means that you cannot use the Show available functions.

Here you can see how to work with subdrawings and the function **Transfer PLC addresses**. The examples are based on subdrawings in the folder STANDARD\DEMO\PLC.

START THE PROJECT

Start a new, empty project and drag two 2 PCSA3BASIC and 4 PCSL1N into the project. Save it as StartPLC.pro. You are going to use the file again later.

EXAMPLE 1

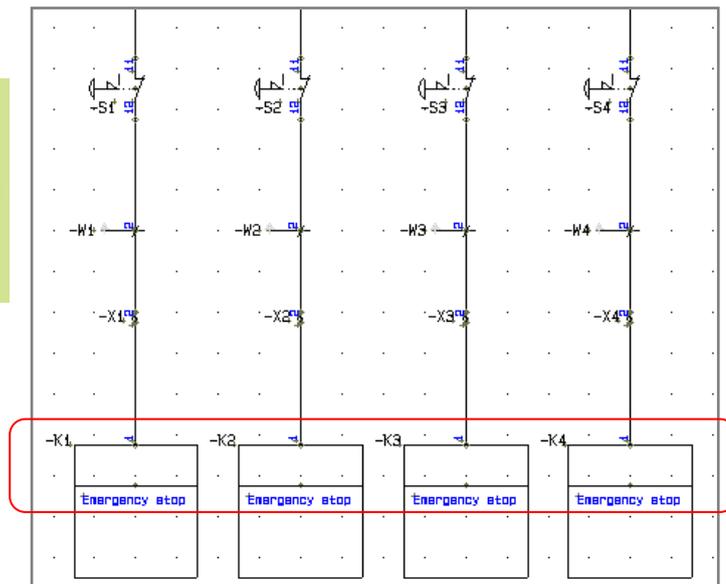
The first example shows how to transfer addresses and connection names one way and function descriptions the other way. Start by placing the inputs:

On page 3 you drag the subdrawing Input1 into the project four times.

Drag the subdrawing into the project by holding the left mousebutton down on the selected subdrawing and let go on the desired page. Now it will place itself correctly..

In order to use this to make a real project you have a couple of issues to overcome:

- The various inputs has come into the project with different component names
- Connection names are identical
- Correct addresses are missing.

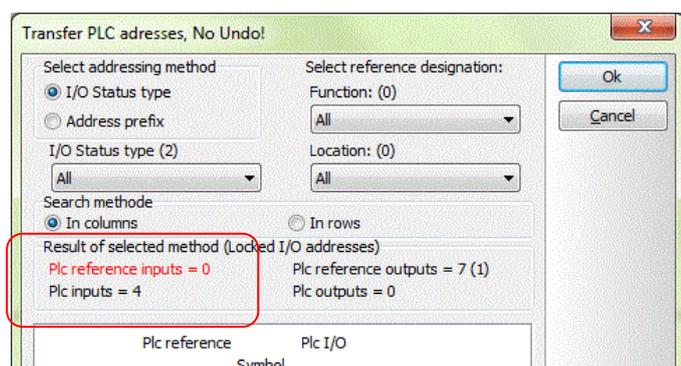


Open the dialog **PLC => Transfer PLC addresses**.

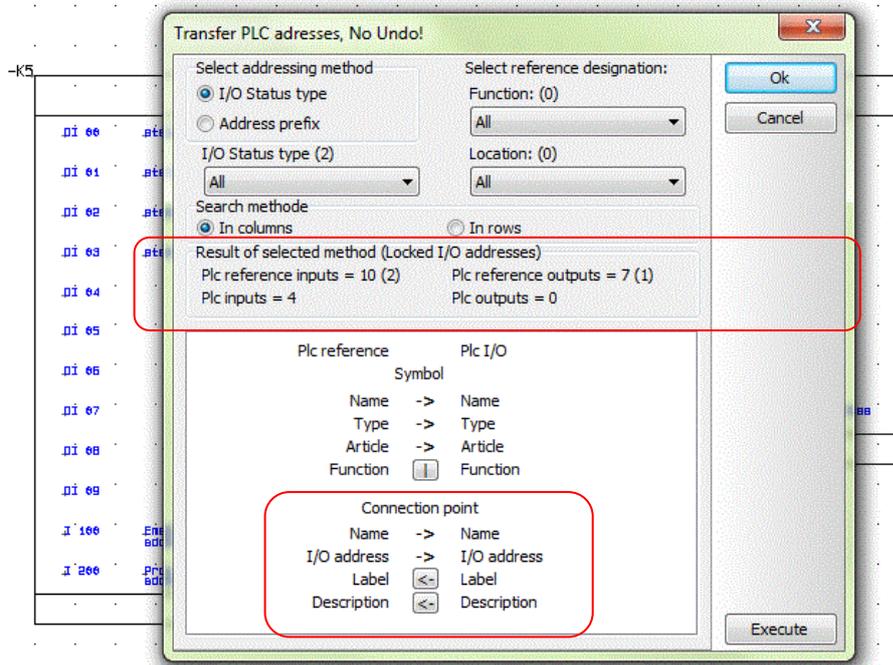
The program sees the 4 plc-inputs, that is the program tells you the needed size plc for the control circuit.

That means that you need to place reference symbols with sufficient addresses in the project.

Drag the subdrawing RefSymbol1 into page 1.



Place the reference symbols on the page and open the dialog **PLC => Transfer PLC addresses** again.

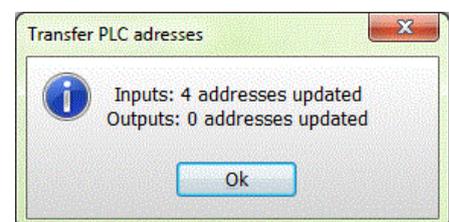


Now you are going to transfer the addresses of the reference symbol to the input symbols placed in the project.

At the top of the dialog you find various options that will be described in a later example in this booklet.

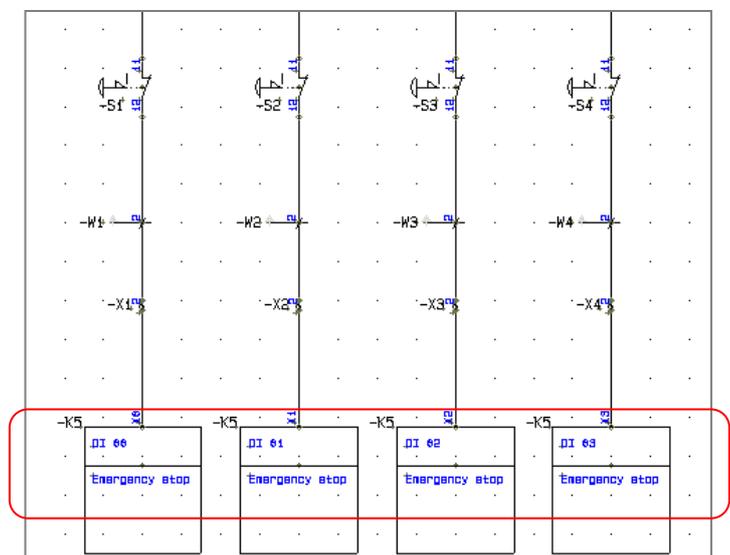
At the lower half of the dialog you see that you transfer data concerning name, type, article and address FROM the reference symbol TO the input symbols; you are also sending data concerning the input's function (label and description) FROM the input symbol TO the reference symbol.

Click **Execute** and **Ok**.



The inputs on page 3 have been updated:

- All inputs have the same component name
- All inputs have an address from the reference symbol
- All inputs have their connection name
- All inputs have references back to the reference symbol.



EXAMPLE 2

In example 2 you can see how to update several plcs in the same project by using the filtering function in the **Transfer PLC addresses** function.

If you work with subdrawings in connection with plcs, you will have subdrawings for each type of connection to inputs and outputs respectively: A certain type of sensor that connects to a certain type of input, an input to the emergency stop with the fixed/locked address that is used throughout all projects; the output that activated a particular component etc..

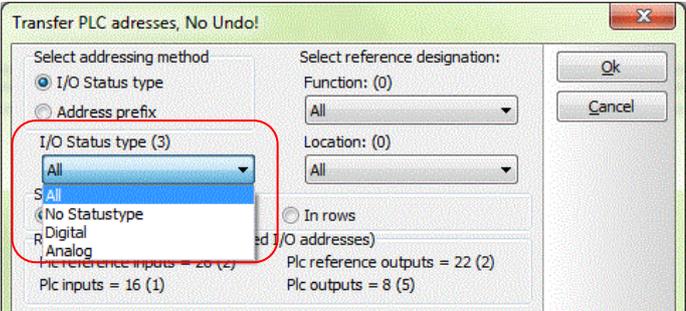
In the PLC-folder you will find a set of different subdrawings that illustrates some of the options available in the dialog.

Start by opening the file StartPLC.pro.

Place the subdrawings RefSymbol 1 and RefSymbol2 on pages 1 and 2, and on the following pages place at least one of each of the other subdrawings. In this way you have different kinds of subdrawings in the project and then you can try (some of) the filtering functions.

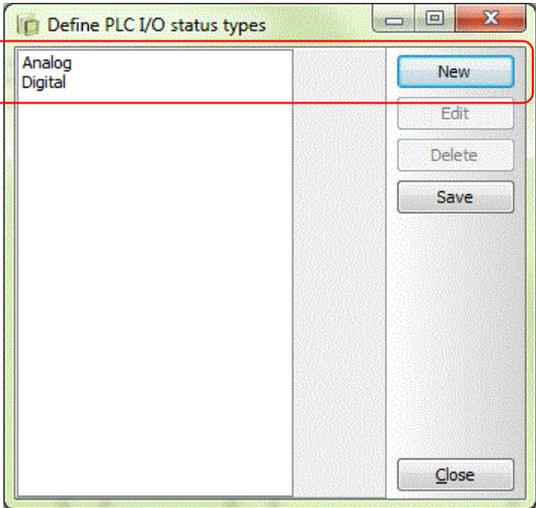
I/O statustype

You can transfer addresses to PLC IOs with a certain IO statustype:



Selections in the subdrawing

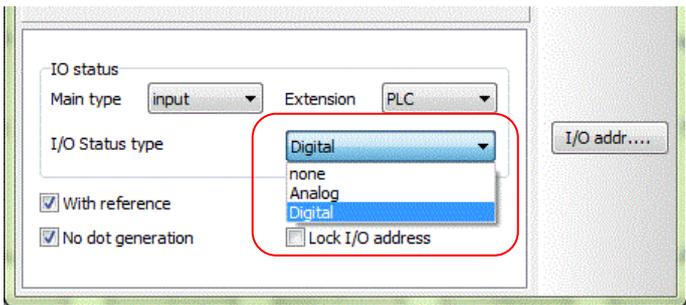
Status type can be defined freely in the dialog **PLC => Define PLC I/O status types**.



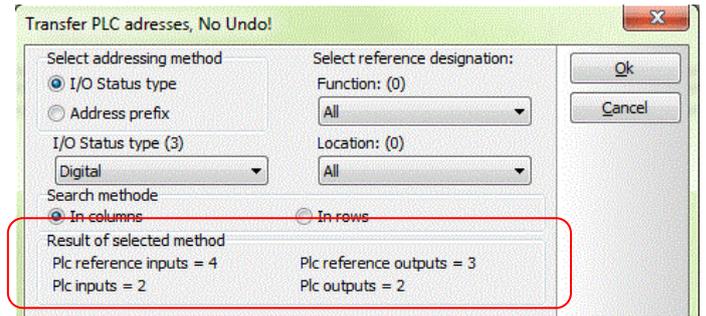
Status types are relevant when you have no other marks on a plc than the overall type – input or output.

Status types are selected on the individual IO symbol and on the reference symbol. In order to select, you must open the connection point with the Extension PLC, which are the the connection points updated by the plc functions.

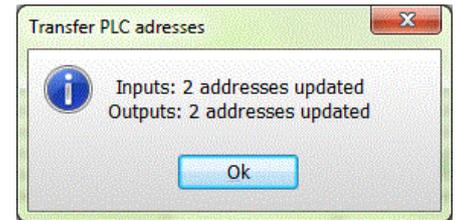
You can choose between the defined status types on your pc..



When you select to use one of the filters you can see how many IOs (PLC inputs and outputs) and PLC reference symbols are found by the selected filter.

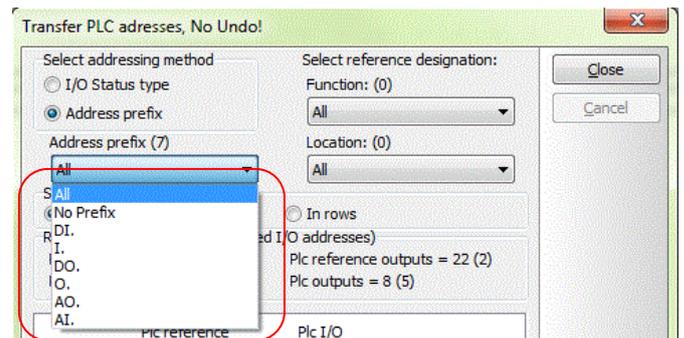


If you choose to transfer adresse with the selected filter you get this status box back: That means that addresses have been transferred FROM the reference symbols TO the PLC-symbols where both symbols had the selected setting.



Address prefix

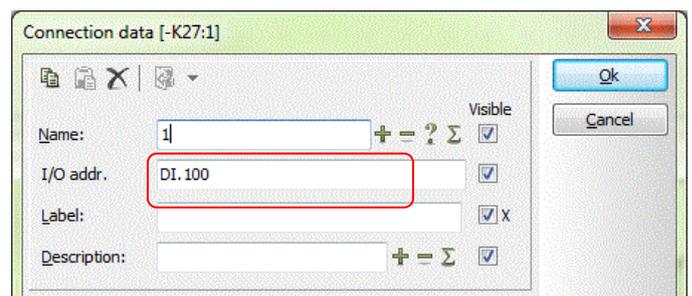
Another filter option is the address prefix. Prefix'es are a relatively fixed types as address prefixes mostly are related to certain manufactures and types.



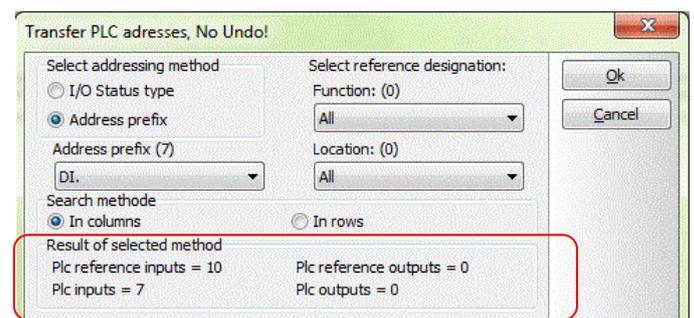
Selections in the subdrawing

The address prefix is typed on each IO and on the reference symbol. In order to type here, open the connection point with the extension PLC.

The prefix is typed in front of the address, eg DI.100.



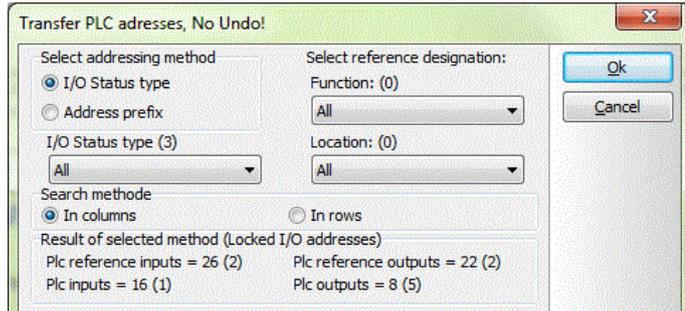
Again the filter choice means that you only transfer the selected address to the selected IOs.



Locked addresses

You can have functions that always need the same address in all projects.

When you have the list here, you can see how many locked addresses you have in (). These addresses must be handled manually, as you cannot transfer new addresses to them.

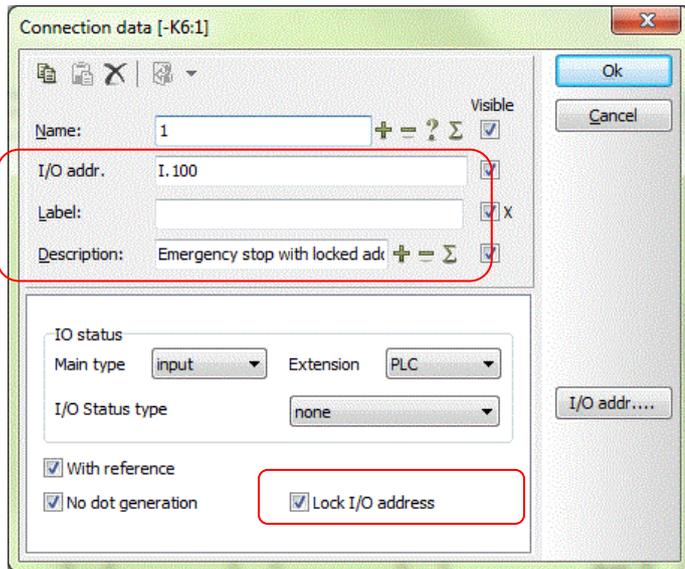


Selections in the subdrawing

You can lock an address to the individual IO and to the reference symbol. In order to lock the address you must open the connection point with the extension PLC.

Here you have to type the address and to click the Lock IO address box.

A piece of advise is to write a description for the actual address on both symbols.

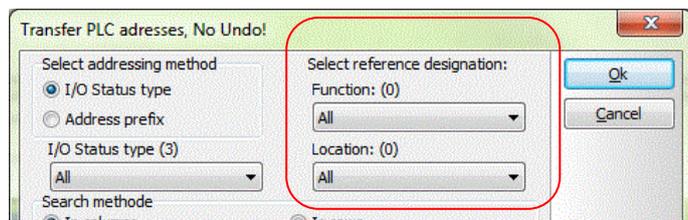


Select a locked address

When you have transferred the various addresses in you project, you are left with the locked addresses. Locked addresses must be handled manually. That means that in order to fetch the connection name and thus make reference between the two symbols you click the button **I/O addr ...** as seen on page 14.

Reference designations as filter

Apart from the filters covered here, you can filter by reference designations used in the project.



RE-ADDRESSING REFERENCE SYMBOLS

If you make very large project you may need to transfer addresses to many inputs and outputs. This also means that you need many reference symbols in the project, which need to be addressed.

Of course, you can open each reference symbol and type its start address, but if you have a lot of ref.symbols of a kind in the project, it is nice to be able to re-address all those ref.symbols with one command.

The dialog **PLC => Re-address PLC reference symbols** has been made to this purpose.

You can re-address the reference symbols through the same filters as described in the previous pages, ie IO statustype or address prefix.

You can also filter on input/output and component names.

You can transfer the following information to one or more components:

- Start address
- Number system
- Count value for the selected statustypes or address prefix.

Readdress PLC reference symbols, No Undo!

Select addressing method

I/O Status type
 Address prefix

I/O Status type (3)
All

Select reference designation:
Function: (0)
All
Location: (0)
All
Component: (0)
All

Search methode
 In columns
 In rows

Result of selected method (Locked I/O addresses)
Plc reference inputs = 26 (2)
Plc reference outputs = 0

Address start: 0.00
Num. System: Oct
Select component: All

Start address for
 Inputs:
 Outputs:

I/O Status type	Count Value
All	1

Ok
Cancel
Execute

A LITTLE MORE ABOUT PLC SYMBOLS AND COMPONENT

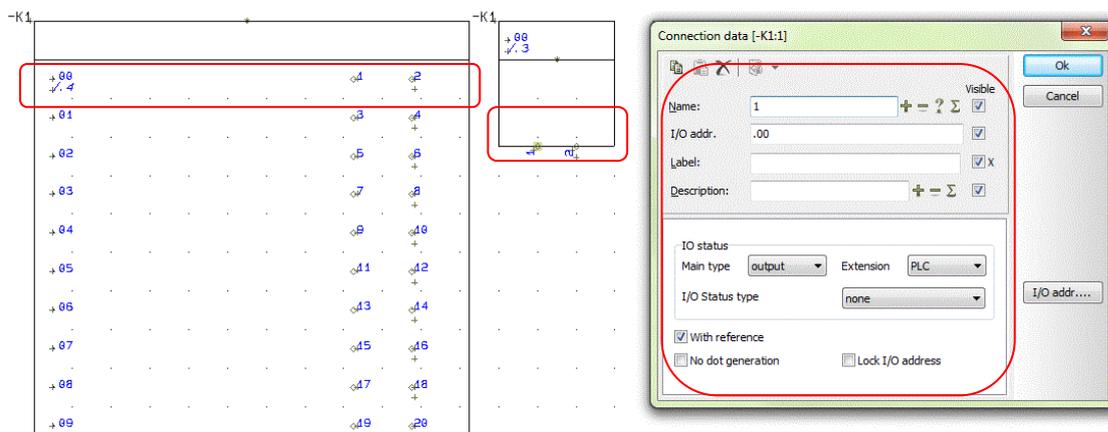
If you wish to start creating your own plc's in the database after having read this booklet, the next section contains a little piece of advice and some ideas for further reading.

PLC-SYMBOLS

Symbol design is a chapter in the manual which is recommended reading.

When you design your own symbols you need to pay attention to make the *symbols match in pairs* – in the same way as the symbols in the pickmenu match. Below you can see a list of the points in which the symbols must match:

- The top row of the reference symbols must be identical with the contents of the plc-symbol. That means that the name of the *first* connection point of the reference symbol (*here '1'*) is also the name of the plc-symbol connection point.
- The connection point of both symbols must have *Main type = Input or Output* and *Extension = plc or plc/term*.
- The start address for reference symbol and the plc symbol must be identical and is written in the field *I/O addr (here '.00')*
- The plc-connection points of both symbols must be ticked in the *With reference* field.
- If the symbols has more connection points for the same address *all* connection points must have this address written in the *I/O-address (function)*field, if you want the symbols to be synchronized.
- Extra connection points (e.g. supply pin) must have *Main type* and *Extension* as *None*.
- If you want more connection points pr address to be synchronized for each address, all of them must be part of the reference symbol.
- When you design the rest of the plc reference symbol, simply make a copy of the top row and make sure that you renumber connection points and addresses. The plc-symbols will be related automatically when you follow those rules.
- Be sure that you save the symbols as PLCreference or PLC types, respectively.
- You only make the first plc-symbol, that is one symbol, which is given the same address as in the top row of the reference symbol.



It is a good idea to open some of the *symbols* in the *pickmenu* to see how settings are on these *symbols*: In the pickmenu Automation DEMO you find plc's from the database, in the pickmenu AUTOMATION you find four sets of plc-symbols,

PLCs IN THE DATABASE

It is a great advantage to create the plc's in the database. It gives the opportunity to use the same symbols for several component as you can control connection point names, and the option to pre-address the symbols, alter the IO-status for each connection and more.

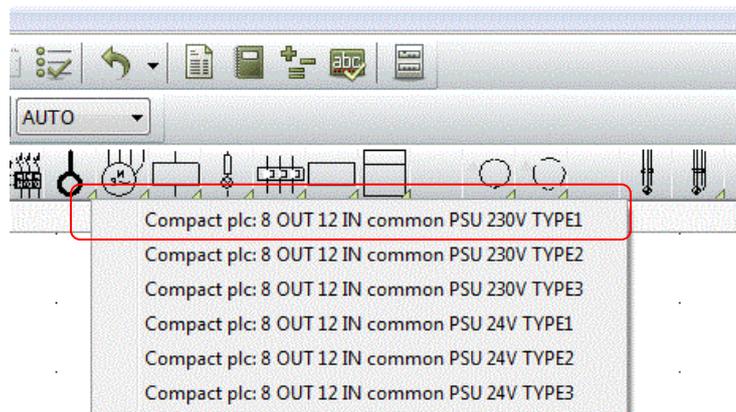
When you create components in the database you should have this data:

- EANNUMMER (barcode number) which is used as article number by Automation. It is also used by Automation to fetch all other data on the component. It must be unique.
- TYPE for the component.
- Preferably a DESCRIPTION which is included in most parts and component lists.
- PCSTYPE contains filenames for all used symbols.
- PINDATA contains pin-names (and addresses) for all electrical symbols.
 - PCSTYPE and PINDATA can contain a link to a file if the componnet contains many symbols and connection pins.
- MECTYPE contains the filename for the components mechanical symbol. Alternatively, it may contain a code similar to "#X100MMY120MM" which means a box of 100x120 mm. The box is without connection pins.

Below you can read more about the data found on one of the plc's in the database.

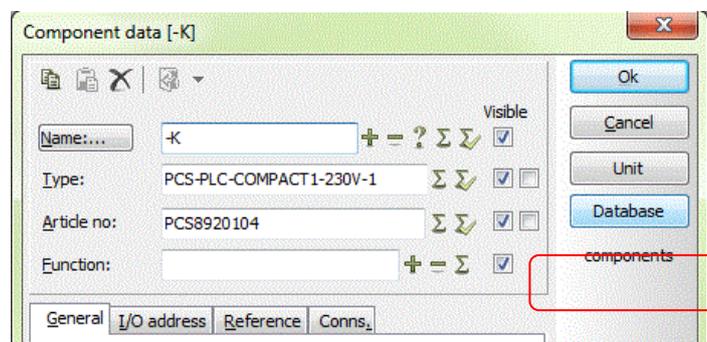
Simple plc in the database

Pick the top plc from the Automation DEMO pickmenu and place its first reference symbol.



In the dialog Component data, click the button Database.

In the next window click the pen (at the bottom of the next dialog) to open the record.



It is a good idea to open some of the *components* in the *database* to see how data is written for those *components*.

The screenshot shows the 'Record Editor' window with the following data entered:

- ID: 78162
- EANUMBER: PCS8920104
- STOCKNO: (empty)
- ORDERNO: (empty)
- TYPE: PCS-PLC-COMPACT1-230V-1
- DESCRIPT: Compact plc: 8 OUT 12 IN common PSU 230V TYPE1
- UKDESCRIPT: Compact plc: 8 OUT 12 IN common PSU 230V TYPE1
- DEDESCRIPT: (empty)
- REMARKS: (empty)
- CATALOGUE: (empty)
- SOURCE: (empty)
- MANUFACTUR: PCS
- UNITPRPACK: 1
- PRICE: (empty)
- DISCOUNT: (empty)
- NETPRICE: (empty)
- TAX: (empty)
- WEIGHT: (empty)
- PCSTYPE: PLCREF8-;PLCREF12I;PLC1OUT-UF#8;PLC1IN-UF#12
- PINDATA: FILE=PCSp1c1.pin
- SINGLELINE SYMBOL: (empty)
- MECTYPE: PCSPLC230

PCSTYPE contains the filename for all plc symbols. Here the 'sentence' means 1 ref-symbol for outputs, 1 ref-symbol for inputs. 8 plc-symbols for outputs, 12 plc-symbols for inputs. The symbol names are separated by *semicolons*.

The symbols contains too many pins for the datafield PINDATA. Instead it contains a link to a file, in which each symbols connection pins are written and separated by *commas*.

Create plc in the database

If you are going to create a component like this one you might just copy and paste this one: *Ctrl+click* the copy-icon and *ctrl+click* the paste-icon and edit type and eanumber.

You can also create a new plc with two connection pins per address:

Type EANUMMER and TYPE (remember that EANUMMER *must* be unique.

In PCSTYPE type "PLCREF16;PLCREF16I-1;PLC1OUT-P#16;PLC1IN-PT#16"

In PINDATA write "FILE=PCSp1c2.pin"

Save the record.

You have now created a new plc, which consists of 16 inputs and outputs, each with two connection pins per address – a typical analog plc – which is ready for use.

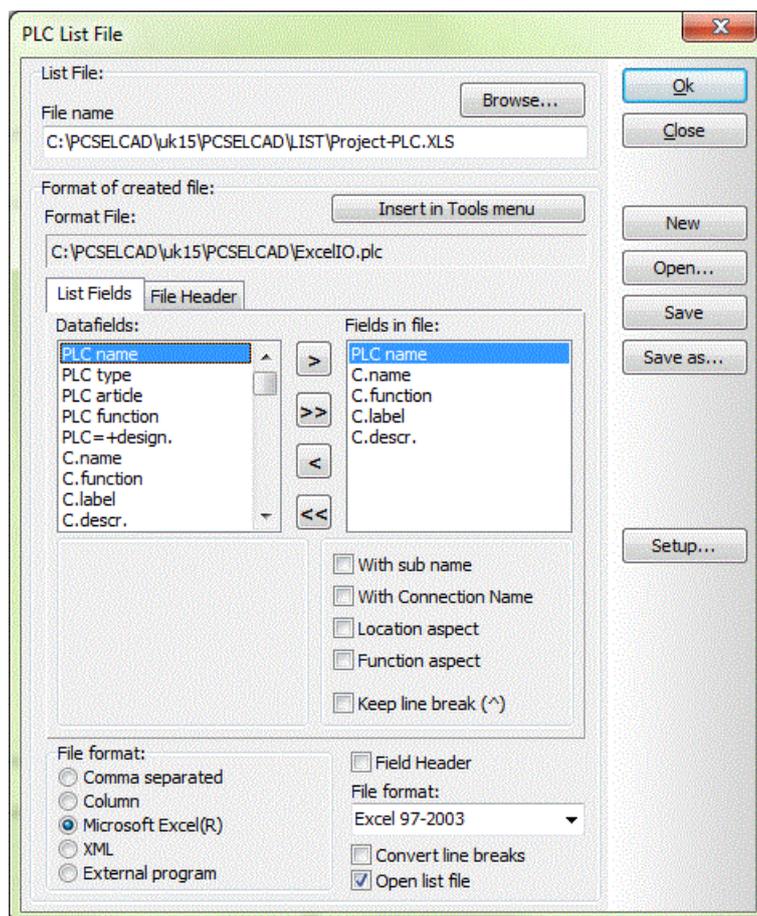
FORMAT FILE FOR I/O LIST

Import and export of IO-listes from e.g. Excel-files is an efficient way to get data to and from the plc-symbols. However, if this is going to work, you need to have the correct data in the correct column in the file. Which data to have where depends on the tool that delivers the data. If you open one of the included formatfiles you can see which data is included. After seeing (and understanding) you can design your own format file if necessary.

Open **PLC => PLC list file** to see what the format file contains. Here you see the ExcelIO file.

The file contains these columns:

- PLC name, that is -K1, -K2, +A1-K1 etc
- C.name, that is the connection pin name, the physical connection terminal name on the plc
- C.function, which contains the plc-address for each connection point
- C.label, in which you can write a program label or the like
- C.descr., which is the description of the address' function.



In this file you see a separate column for the connection pin. For some I/O-lists, however, this information is included in the same column as the plc-name. If this is so for your plc, you simply select the option *With connection name* on the PLC-name.

Some (larger) plcs are designet with separate cards where each card gets a subname to indicate its relation to the plc (controller): The cpu is called -K1.1, the power supply is -K1.2, an input card is -K1.3 etc. If this is the case, you have to select the option *With subname*.

In the same way, you can add/delete desired fields and you can change the datafield order for your list.

Remember, that the Automation project and the Excel file must contain the same plc-names and pin-names in order to exchange data.

INDEX

Address prefix.....	21	Reference symbol and plc symbol	11, 15
Component in database.....	7	Reference symbols	
Database	25	Re-addressing.....	23
Format file.....	27	Show available	9
IO list	12	Start project.....	6
IO-address	10, 15	Sub-drawings	18
IO-statustype	20	Symbol	
Define	20	Input symbol	9, 24
Locked address	22	Output symbol	10, 15, 24
Object lister.....	10	Plc input	9, 24
Page menu	6	Plc output	10, 15, 17, 24
Plc symbol and reference symbol	11, 15	Reference symbol.....	24
Plc-address.....	10	Symbols	7
Plc-symbol	24	Transfer plc addresses.....	18
Reference designations	22		
Reference symbol.....	8, 14		