iPLC - Cybrotech expansion modules

User Manual

rev.1

cybroTech

Cybrotech Ltd
14 Brinell Way, Harfreys Industrial Estate
Great Yarmouth, Norfolk, Nr31 OLU - UK
tel: +44 (0)1493 650 222
www.cybrotech.co.uk
info@cybrotech.co.uk

Safety information

⚠ Warning

These products can only function correctly if transported, stored, set up and installed correctly, and operated and maintained as recommended. Failure to comply with applicable codes and standards can result in damage to equipment or serious injury to personnel.

To minimize the risk of potential safety problems, you should follow all applicable local and national codes that regulate the installation and operation of your equipment. These codes vary geographically and usually change with time. It is your responsibility to determine which codes should be followed, and to verify that the equipment, installation, and operation comply with the latest revision of these codes.

We do not guarantee the products described in this publication are suitable for your particular application, nor do we assume any responsibility for your product design, installation or operation.

⚠ Warning

Control devices may fail in an unsafe condition, resulting in unexpected operation of controlled equipment.

Such unexpected action could result in death or serious personal injury, and/or equipment damage.

Install emergency stop switch, electromechanical overrides, or other redundant safeguards that

⚠ Warning

Connect a power supply that meets the voltage rating, shown on the front panel.

Devices may fail in an unsafe manner or present an electrical shock hazard to personnel if high voltage is applied to terminals intended for 24V= circuit connections.

Such a failure or shock could result in death or serious personal injury, and/or equipment damage.

Always supply 24V= circuits from a source that provides safe electrical separation from 120/230V~ power.

Provide a circuit breaker rated 10A/Type B that removes power from all connected devices. The circuit breaker or separate disconnect switch should be near the controller.

⚠ Warning

Never touch the terminals while the power is on. There is a risk of an electric shock, which could result in death or serious personal injury.

Index

1 Introduction

This document explains how to setup and use Cybrotech expansion modules with a PLC controller installed inside UniOP touch panel.

This document can be used as a reference for a user already familiar with UniOP panels, iPLC controllers or Cybrotech expansion modules looking to combine them in an automation system.

Also this manual can be used as a guide for new users unfamiliar with any of the mentioned devices but are interested in home automation and are looking for information on available solutions.

User's Manual consists of several chapters.

In chapter 2 a brief overview of hardware components will be presented.

Chapter 3 gives some more information on iPLC controller, how to install it and how to establish communication.

Follows chapter 4 with explanations on how to install/update controller's Target files for CoDeSys development tool.

Chapter 5 gives some brief description on CYBROTECH_IEX2 library, what are requirements and how to use the library in an application.

In the last chapter (6) a step-by-step guide on creating a demo application is presented. It is assumed that reader has a basic knowledge of using personal computers. No previous knowledge of any programming language is required to be able to create the project presented in this chapter.

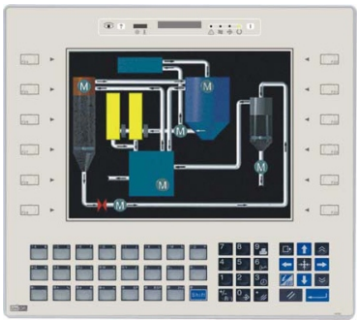
2 Hardware equipment

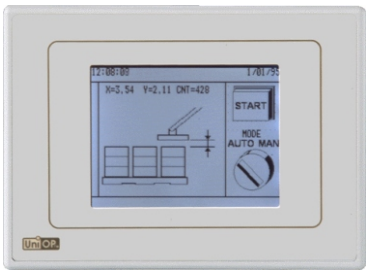
To create a functioning automation system several devices must be included and connected properly. Connection instruction of these devices are out of the scope of this document. Find documents on this subject on Cybrotech homepage together with detailed specification and description of the products presented in this chapter.

2.1 UniOP touch panel

UniOP panels are a touch sensitive panels used for creating HMI (human-machine) interfaces. Through such interfaces customers can overview and control automation system (process) in which such panels are used.

Panels are available in several sizes with monochrome or colored displays and as such offer a wide selection for many different customer's needs.





2.2 SCM12-C controller

SCM12-C (from now on iPLC) is an high performance board, targeted for CAN OPEN and 10Mbit Ethernet and has installed CoDeSys TM softlogic system.

Board is installed directly in any UniOP panel.

The controller on this board can be programmed with CoDeSys development software. CAN interface enables the use of the board in conjunction with Cybrotech expansion modules.

2.3 Cybrotech expansion modules

Cybrotech expansion devices are Cybrotech LTD. products used as a building blocks of home and industrial automation.

Each device offers various data points that can be used to control an automation system. Depending on the system requirement customer can select from several groups of products:

Operator panels

Used for displaying and setting system parameters. Some types also include temperature and humidity sensing and IR receivers.

Cabinet mounted expansion units

Include products with digital sensing inputs; digital relay and transistor outputs; analog voltage, current and resistor inputs and outputs. Devices are designed to be mounted in a cabinet on a DIN rail (35mm).

Field mounted expansion units

Field mounted devices are designed to be placed in different sections of the building close to devices they are controlling. In this group you can find dimmable light controllers, digital relay outputs for controlling lights and blinds, room temperature sensors, switch input modules etc.

Accessories

Accessories are devices that are not connected directly to a PLC controller but are instead connected to other expansion modules. Various other devices (remotes..) that can be used in Cybrotech automation networks.

Power Supplies

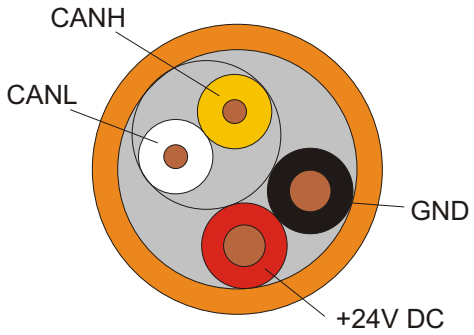
For powering of networked devices several power supplies can be selected with different maximum output currents.

For more information on available products and their details please visit Cybrotech web page at www.cybrotech.co.uk.

2.3.1 IEX-2

Cybrotech expansion devices are connected into an IEX-2 network with a PLC controller. Communication between devices and a PLC is serviced by a proprietary CAN 2.0B based protocol named IEX-2. Physical levels are that of a CAN network.

In addition, the physical wiring connecting all devices is called IEX-2 Bus. The bus consist of 4 wires, two of which are reserved for powering the connected expansion modules (24V DC) and two communication wires as specified for CAN networks.



Detailed explanation of IEX-2 protocol, bus and network are not in the scope of this document. Further information can be found in various manuals and technical notes available at Cybrotech website.

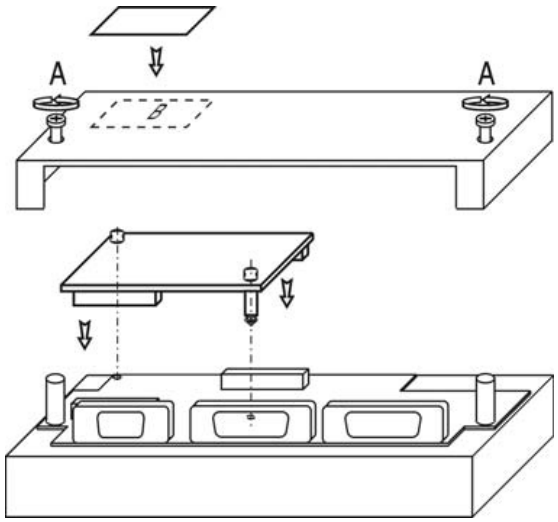
3 iPLC Controller

3.1 Controller installation

The mounting procedure for the module is as follows:

1. Turn off the operator panel.
2. Release (not completely) with a screwdriver the two screws “A” fixing the rear cover.
3. Remove the rear cover.
4. Plug the module in the red connectors and make sure they are properly latched.
5. Replace the rear cover.
6. Fix the screws “A”.

Stick in the area “B” the label indicating the type of module, which has been plugged.



Successful mounting can be confirmed by powering UniOP panel and recalling System Menu (press and hold anywhere on the display where there are no keys).

With keys displayed at the bottom of the touch screen enter System Menu (System menu for UniOP model eTOP03 shown below).

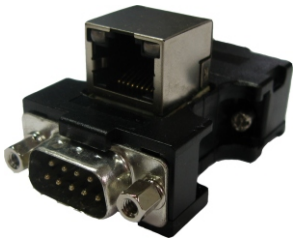
```
EXIT
LO <- CONTRAST -> HI
OFF <- LAMP TEST -> ON
PLC . . . . . .  SCMxx #0 APPLICATION
IPLC CoDeSys 3.00
PRINTER . . . . . ON-LINE
BATTERY . . . . . OK
SCM12 0 H280 X240 OK
empty 1 H000 X000
empty 2 H000 X000
empty 3 H000 X000
```

```
->   SCM12 0  4.01Q
```

1. In line 4 set PLC to APPLICATION.
2. Line 5 must be as shown above.
3. Line 8 will be similar as above if installed controller is working and is detected. Pressing key LEFT when will display firmware version of SCM12-C. Make sure it is version 4.01Q or later.

3.2 Communication adapter

SCM12 is connected via the additional adapter board ETAD 03, equipped with a standard RJ45 and DB9 male connectors:

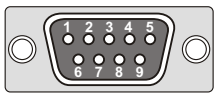


Ethernet connector

Pin Signal:
1 TD out +
2 TD out -
3 RX in +
6 RX in -

CAN Port

Pin Signal:
2 CAN_L
3 CAN_GND
7 CAN_H



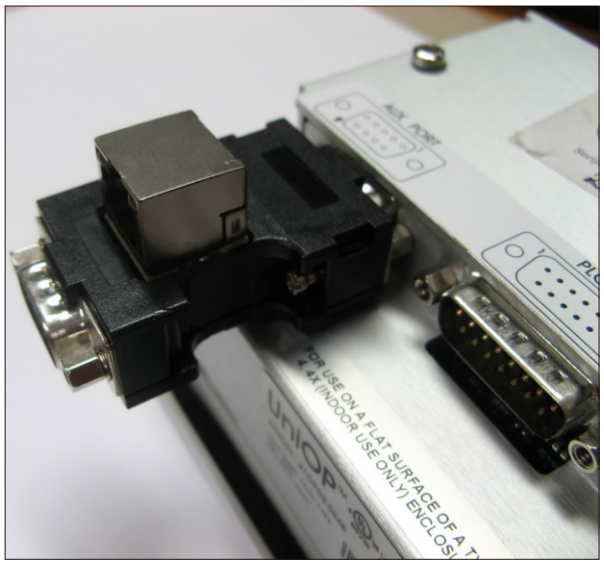
DB-9
male

There are two diagnostic indicators :

LINK (red) : LED is on if no network signals have been detected or in case of general failure of the SCM12.

Use of the adapter is mandatory for creation of working IEX-2 network!

Adapter is connected to AUX port of UniOP panel.



3.3 Communicating with iPLC

When UniOP does not contain a valid project, it stays in Configuration Mode; in this situation the PLC port is assigned by default to the Internal Controller.
To program the Internal Controller when no project is loaded in the panel, use the normal UniOP programming cable connected to the PLC port of the panel.
If the operator panel has no PC/Printer Port, then the PC/PLC Port will be used also for communication with the Designer software. The Internal Controller will be programmable only after a valid project has been transferred and the panel is in Operation Mode.
If the SCM11-C/SCM12-C module is used, then the Ethernet port is always available for communication with the PLC run-time system.
When a valid project is present in UniOP and the panel is in Operation Mode, the System Menu will allow the user selecting the mode of communication for the Internal Controller using the following procedure:

- recall the System Menu in the panel,

- use the Up/Down arrow keys to scroll the menu lines until PC/Printer or PLC are highlighted,

- use the Left/Right arrow keys to change the assignment of the selected port.

Two options related to the Internal Controller are available for each UniOP port:

Application Mode	Ports should always be assigned to the Internal Controller in Application Mode if it has to be used for normal operations as application downloading and debugging.
Service Mode	Service Mode is reserved for special Internal Controller maintenance and should not be used.

The Port is assigned to the Internal Controller in Application Mode when the corresponding row of the System Menu displays the text "**Application**". This message is reduced to "**A**" for displays with 20 characters per row.
When the PC/Printer port is not assigned to the Internal Controller, it reports the printer status as usual.
When the PLC port is not assigned to Internal Controller and the Designer project does not use an external controller, the System Menu contains the string "**NOT IN USE**" in the PLC row. In case an external controller is used, the PLC row reports the communication error code as usual.

Note: Any modification to the port assignment done in System Menu becomes effective after you exit the menu.

Communication with the Internal Controller is possible both when the panel is in Configuration Mode and when it is in Operation Mode.
Standard UniOP programming cables CA2 or CA114 can be used to connect the CoDeSys software UniOP. A gender changer may be required to connect to the UniOP PLC port.

3.4 Setting-up communication

The HMI control system is composed by two subsystems, the UniOP Operator Panel and the SCM Internal Controller Module. Programming both subsystems can in case of SCM12-C be done via serial or ethernet communication. Either can be used for both subsystems but not at the same time!

The set-up of the CoDeSys communication is described in the following chapters.

3.4.1 IP address

Ethernet programming for SCM12-C modules is possible only if the UniOP panel is equipped with firmware version V5.20 or greater. If using ethernet port for programming of UniOP or iPLC it is recommended setting a static IP address on which both devices will be accessible.

To set the this enter Configuration Menu (long press anywhere on the panel then select with direction keys and Enter).

Example of Configuration menu from eTOP3 can be seen below.

```
UniOP 5.51C
CONFIGURATION MODE
 IP: 192.168.1.135
ETH: 0030D8011585
 SM: 255.255.255.0
GTW: 192.168.1.1
07:40:14 gmt+00:00
DST: off
```

By pressing anywhere on lines 3 through 6 an option of changing ethernet address settings will be activated.

With the use of directional and Enter keys you can input appropriate IP address of device. It is recommended to use a static IP address by defining a fix IP address (line 3) and subnet mask (line 5).

Be careful to set both parameters to such values as not to cause conflict with any other devices already connected to targeted local network!

If only GTW parameter is set, IP address and subnet mask will be set by connected network gateway.

This IP address, set automatic or manually, is used in CoDeSys and Designer when establishing connection.

3.5 Limitations

There are some limitations in the configurations available for programming the Internal Controller. This chapter provides an overview.

1. If UniOP contains a valid project that uses the PLC port to communicate with an external controller and it is in Operation Mode, then communication with the Internal Controller is not allowed through the PLC Port, because it is already assigned to the PLC communication.

2. If UniOP contains a valid project configured to work with Remote Passthrough, the communication with the Internal Controller through the PC/Printer Port is not allowed. The PC/Printer port is already assigned to wait for incoming commands for the Remote Passthrough operation.

3. If UniOP contains a valid project configured to use the UniNet network and the PC/Printer port is assigned to network communication, the same UniOP port cannot be used to communicate with the UniOP Internal Controller. A similar consideration applies in case the PLC port is used as network port: communication with the PLC is not allowed through the same port.

4. If UniOP contains a valid project where the external controller is configured with a protocol that requires a TCM module, then the Internal Controller may not work properly. Operations with external controllers that require Ethernet interface via SCM12-C are instead always allowed.

4 CoDeSys

CoDeSys is a complete development environment for your PLC (CoDeSys stands for Controlled Development System).

CoDeSys puts a simple approach to the powerful IEC language at the disposal of the PLC programmer. Use of the editors and debugging functions is based upon the proven development program environments of advanced programming languages (such as Visual C++).

Tool is available for download with registration from manufacturers homepage 3S - Smart Software Solutions.

For software installation see user manual also available from manufacturer.

4.1 SCM12-C Target filles

Upon successful installation of CoDeSys development software an installation of SCM12 target files is required to enable CoDeSys to program this type of controllers.

The UniOP Target-Package-Support required to support the SCM12-C hardware is available for download from the www.uniop.com web site in the Document area under the CoDeSys folder.

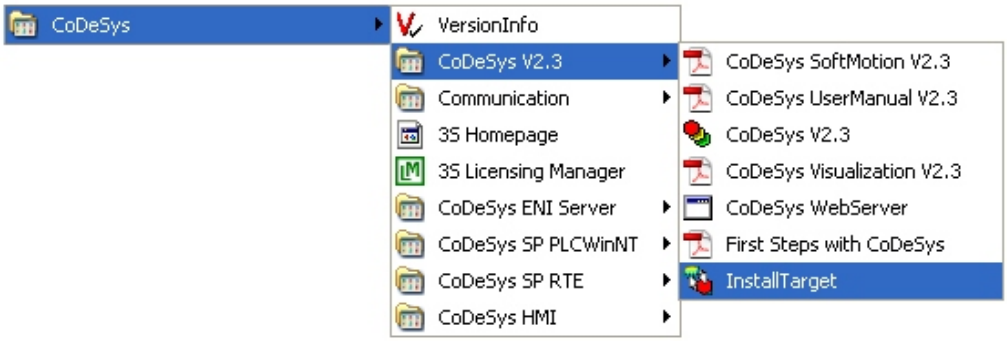
The Target support is provided as a zip file that can be un-zipped in any directory respecting the internal folder structure.

NOTE: Use version 4.01M or later (*SITEK_targen 4.01M install.zip*).

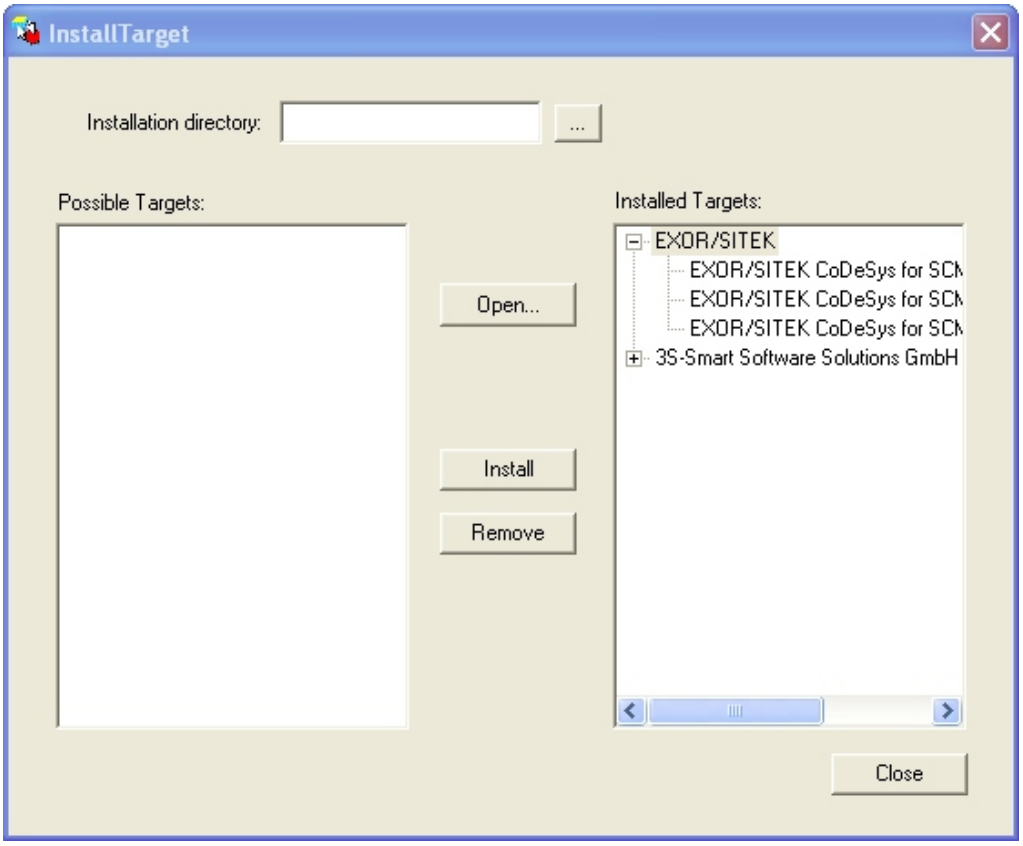
The CoDeSys installation package includes a tool for Target installation.

The tool is called "Install Target" and can be used both for checking the installed target support packages and for installing new ones. To install the SCM Target Support Package just click "Open" and locate in the hard disk the file with extension ".tnf" that was supplied with the zip package. The target tool does the rest of the installation automatically.

The Install Target tool can be found in the CoDeSys program group that you will find in the Start Menu after the installation of the software.



Correctly installed Target files will be displayed as seen on the picture above.



4.1.1 Updating old Target Support Packages

Any new version of the CoDeSys "Target Support Package" can be installed over the existing one; this would keep the support for the older modules, still requiring old versions of the support package.

Start the "Install target" tool from the 3S CoDeSys program group, click "Open", select the .TNF file from the "Target Support Package" folder, click on "EXOR/SITEK" and click "Install".

Depending on the presence of old support packages, the software will ask to overwrite existing settings; please confirm to overwrite them.

After the new Support Package has been installed the conversion of existing projects may require some steps depending on the level of compatibility of the new Support Package in compare with the old one. The conversion is done automatically by the CoDeSys programming software when opening the old application. Any update of the "Target Support Package" will come together with proper instruction for existing projects conversion, if needed.

5 CYBROTECH_IEX2 library

The CYBROTECH_IEX2 library was created with a purpose to enable the connection between UniOP touch panels with installed PLC controllers (SCM12-C) and Cybrotech expansion modules.

The goal was to achieve simple set-up and easy use of the library, minimizing the time learning new system.

Library was created in CoDeSys development tool and is so directly usable in any projects also developed with this tool.

Including this library in a project enables the user (programmer) to directly use I/O points of any Cybrotech modules connected to the automation system in the application. No communication protocol is needed to be handled by user's code, all is serviced by the library.

5.1 Installing library

Library is available for download from Cybrotech home page. Library comes archived with another library also required to be used in a project.

After download, extract the file. Three files should be created:

- EXOR_CAN2.LIB
- CYBROTECH_IEX2.LIB
- CYBROTECH_IEX2 Version history.txt

See version history to check the changes made in new releases.

Store all extracted files on hard drive. It is recommended to store them in library directory of the CoDeSys development tool (e.g.: C:\Program Files\3S Software\CoDeSys V2.3\Library).

How to include libraries to a CoDeSys project please find explained in chapter 6.2.

5.2 Using library

Note that both EXOR_CAN2 and CYBROTECH_IEX2 libraries must be included to ensure proper operation.

Each project must contain a template segment of programming code that initializes PLC CAN port and handles a part of IEX-2 communication:

- Function *Cybrotech_iex2_init()* must be called once after each reset of the controller with a valid value of parameter *BaudRate*.
- Function *Cybrotech_iex2()* called once each program scan-cycle.
- All connected Cybrotech modules must be represented by allocating an instance of respective function block from the library.
- Each allocated function block instance must be called once each scan-cycle with a valid *Address* (serial number of module) as a parameter.

This can be done in different programming languages supported by CoDeSys.

To access (use) an I/O point of a connected IEX2 module use notation in structure text:

<instance_name>.<i/o_datapoint>

<instance_name> - name of an instance that represents an actual hardware module
<i/o_datapoint> - input or output of respective module

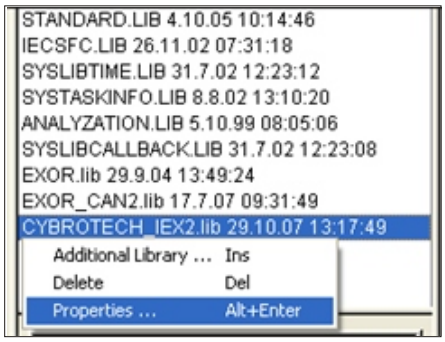
Examples:

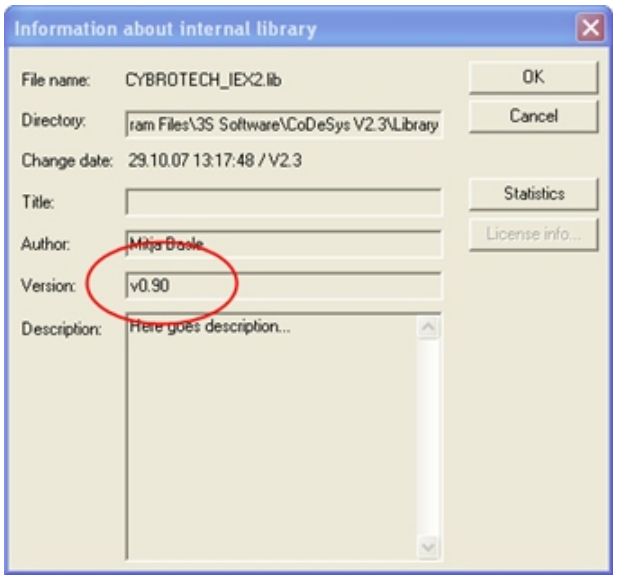
c800.QX0 - digital relay output of an 8C module
fc01.TS0 - measured temperature of a FC module
aov00.QW0 - output voltage level of AoV12 module

5.3 Library version updating

As the range of products available from Cybrotech is constantly expanding, changes and/or additions will be constantly made to the library. Please make sure to visit our web site regularly to check for releases of new version.

To check the version of the library right+click on the file in **Library Manager** and select ***Properties...***





Download any new version and overwrite old library files. Any old project must be ***Rebuild all*** and downloaded to controller.

6 Creating simple appliaction

In this chapter a detailed step-by-step guide is provided in creating an application using UniOP with installed iPLC and Cybrotech expansion modules.

6.1 Application description

To show the required steps in creating a custom application using UniOP panels, SCM12-C controller and Cybrotech expansion modules we will create a demo application for controlling lights.

Interface specification:

- UniOP panel must hold 3 keys for controlling lights and 3 indicators each representing the current state of each light
- 3 in-wall mounted keys

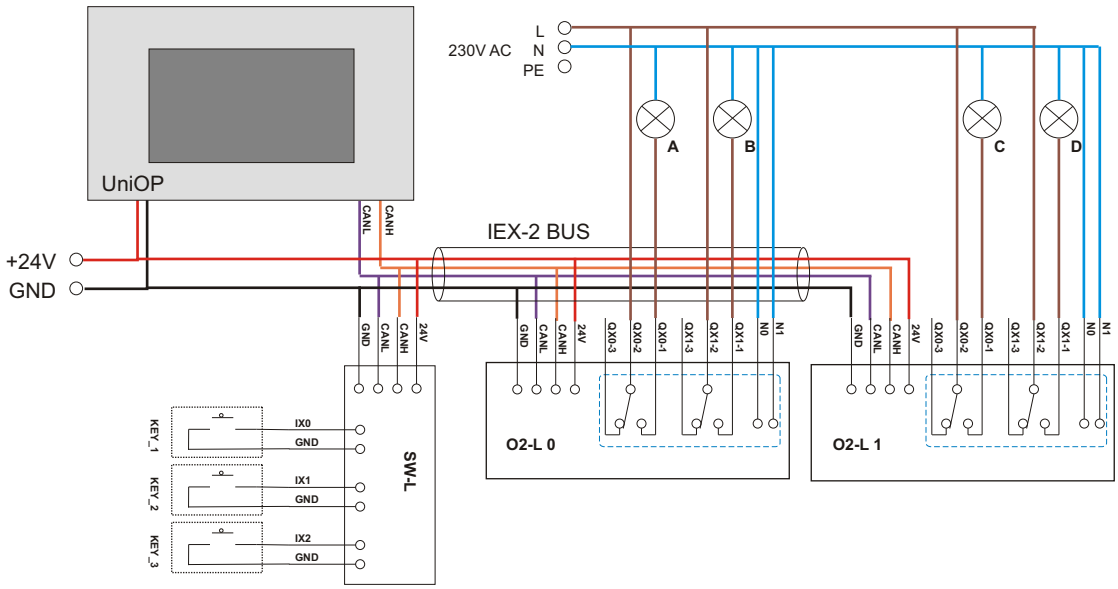
Operating specification:

- press of **key_1** must lights **A** and **B** must be toggled ON/OFF
- press of **key_2** turns ON light **C,** which automatically turns OFF after 30 seconds
- while **key_3** is pressed light **D** must be ON

Based on these specification the following hardware is required:

1 x UniOP panel (eTOP03 will be used in this application)
1 x SCM12-C installed inside of UniOP panel
1 x SW-L (in-wall keys interface)
2 x O2-L (relay module for switching lights)

Schematic of the system is shown below.



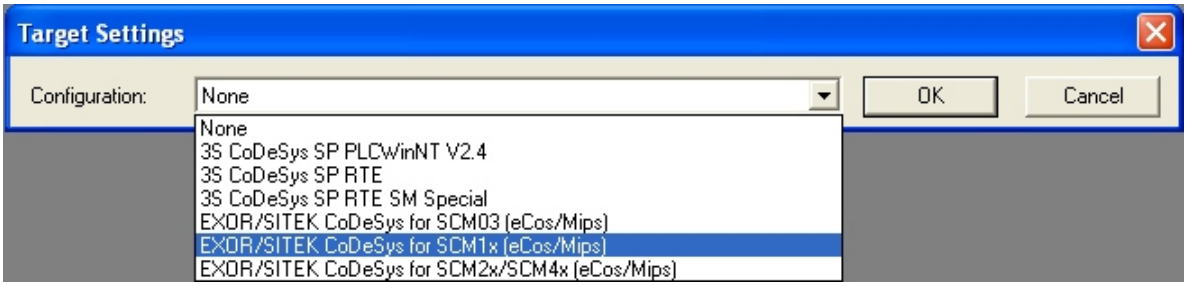
6.2 Creating project in CoDeSys

Before creating a new project in CoDeSys development tool make sure that Target files for controller are properly installed as explained in chapter 4.1.

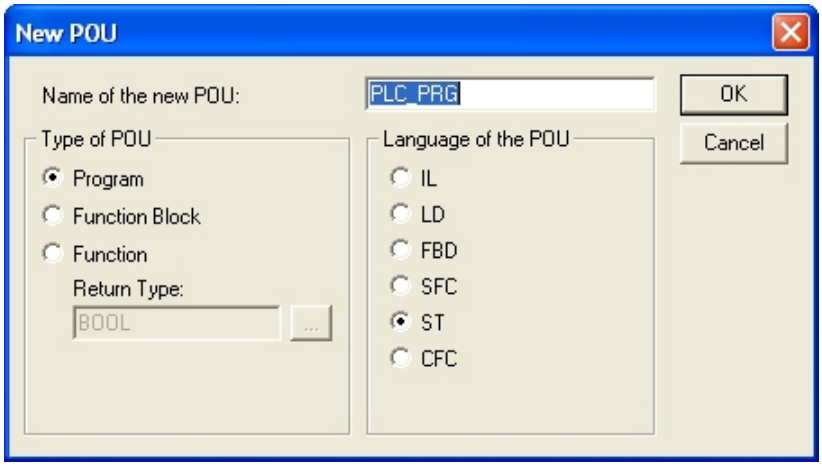
1. Run **CoDeSys** development tool.

2. Start a ***New Project*.**

3. In the Target Settings dialog box select **SCM1x** configuration and proceed with ***OK***.

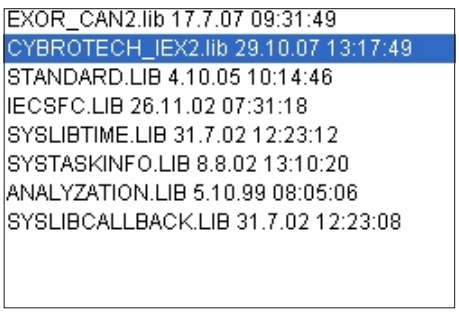


4. First POU (Program Organization Unit) must be created using **PLC_PRG** as a name. For this demo application select Language of the POU as structured text (**ST**) and type as **Program**.



5. Open ***Window* -> *Library Manger*.**

6. Insert **CYBROTECH_IEX2.LIB** and **EXOR_CAN2.LIB** libraries stored locally on hard drive into the project. CYBROTECH_IEX2.LIB is editing password protected. When including in project no password is required to be entered to be able to use it. When asked for password proceed with ***CANCEL.***



7. Under **Global_Variables** in **Resources tab** define instances of all Cybrotech expansion modules used for this application and virtual keys that will be used by UniOP.

```
VAR_GLOBAL
        o200:           IEX2_O2;
        o201:           IEX2_O2;
        swl00:          IEX2_SWL;

        key_1:          BOOL;
        key_2:          BOOL;
        key_3:          BOOL;
END_VAR
```

Use short and descriptive names for instances of modules like shown in the code segment above.

8. ***Save*** project.

Each project using CYBROTECH_IEX2 library must contain a template segment of programming code that initializes PLC CAN port and handles a part of IEX-2 communication.

9. In PLC_PRG window under allocate variable *FirstScan* of type BOOL.

```
VAR
        FirstScan:          BOOL;
END_VAR
```

10. Add following code in the body of PLC_PRG:

```
IF FirstScan=0 THEN
        CYBROTECH_IEX2_INIT(BaudRate:=100);      (*Initialize IEX-2*)
        FirstScan:=1;
END_IF;
CYBROTECH_IEX2();                    (* IEX-2 service routine*)

(*Hardware calls of all connected devices*)
o200(Address:=6731);
swl00(Address:=3781);
```

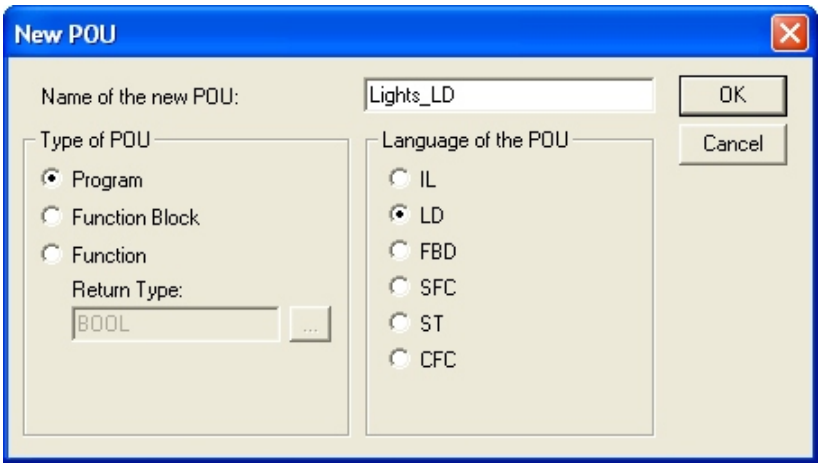
Parameter *BaudRate* can be selected from range of 20,50,100,250,500 kbit/s. For small number of connected devices 100 kbit/s is sufficient speed.

In each project all function block instances (FB) of connected expansion modules must be called with input parameter *Address* which must mach device type and the actual serial number of the device.

As show in this project these calls can also be made from different POU but each FB should be called only once.

Application can be written in different programming languages supported by CoDeSys. To demonstrate this a part of light controlling functionality will be written in ladder diagram (LD), function block diagram (FBD) and structure text (ST).

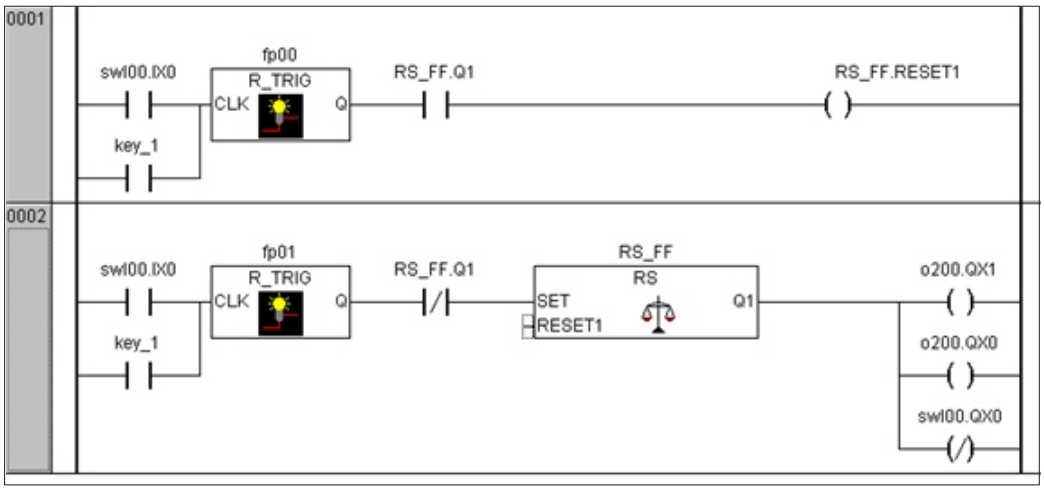
11. Create a new POU named **Lights_LD**. Select ladder diagram (**LD**) as language.



12. In declaration part declare two triggers (positive flank) and RS flip-flop.

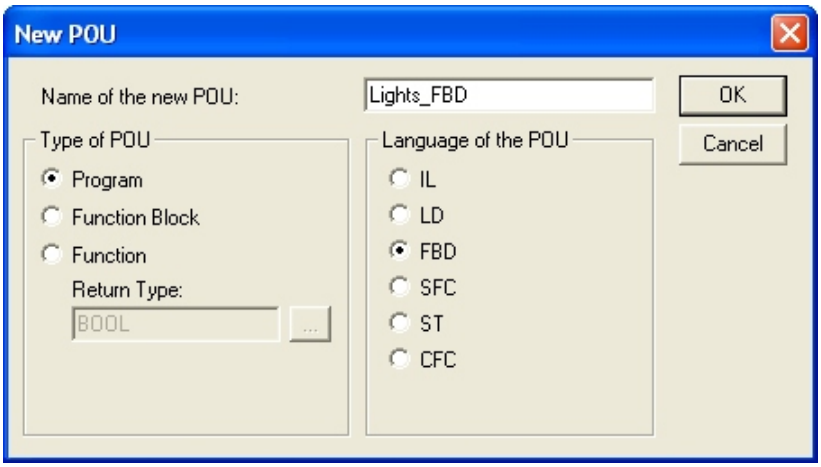
```
PROGRAM Lights_LD
VAR
        fp00:       R_TRIG;
        fp01:       R_TRIG;
        RS_FF:      RS;
END_VAR
```

13. Create a ladder diagram whose functionality is as specified for **key_1** in chapter 5.1.



The next part of application will be written in function block diagram. This segment will service the functionality as specified for **key_2**.

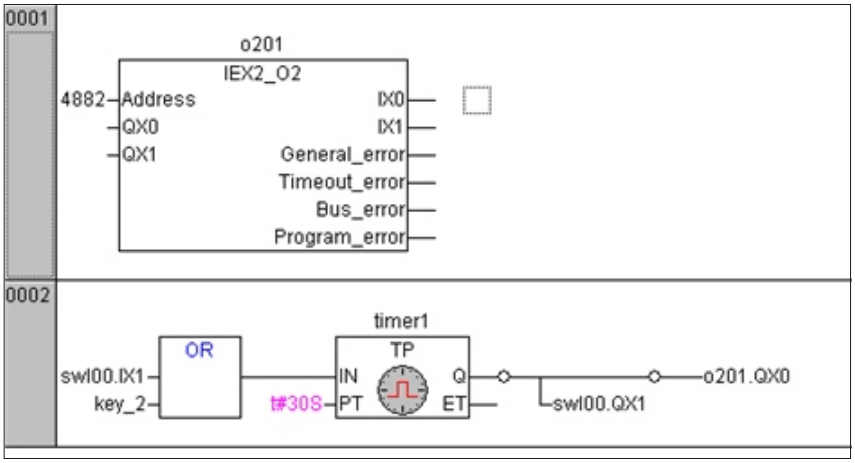
14. Create a new POU named **Lights_FBD**. Select **FBD** as language.



15. In declaration part add a timer.

```
PROGRAM Lights_FBD
VAR
        timer1: TP;
END_VAR
```

16. Create FBD whose functionality is as specified for **key_2** in chapter 5.1.



17. Add a program call in the PLC_PRG body after hardware calls.

18. Add **key_3** functionality in structured text.

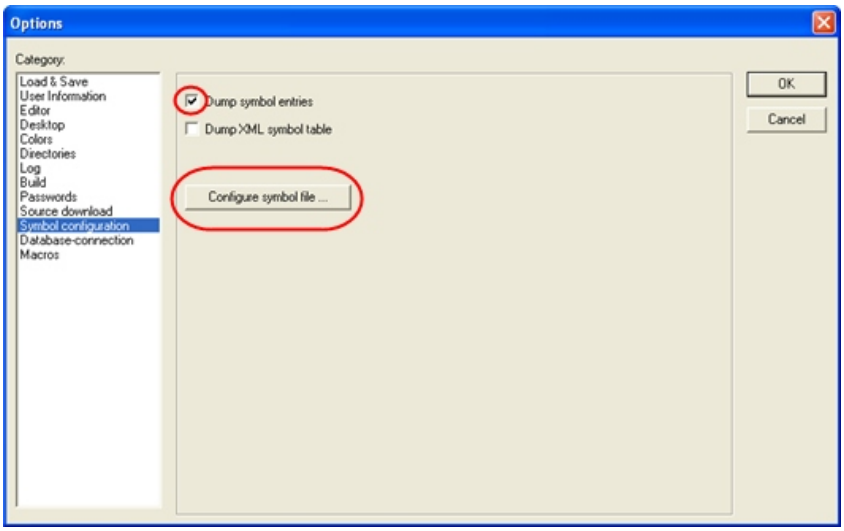
PLC_PRG body should now look like this:

```
IF FirstScan=0 THEN
        CYBROTECH_IEX2_INIT(BaudRate:=100);      (*Initialize IEX-2*)
        FirstScan:=1;
END_IF;
CYBROTECH_IEX2();                    (* IEX-2 service routine*)

(*Hardware calls of all connected devices*)
o200(Address:=6731);
swl00(Address:=3781);



(*User Application*)
Lights_LD();                                    (*Call of PROGRAM blocks*)
Lights_FBD();


IF swl00.IX2=1 OR key_3 THEN                    (*if key_3 is pressed light D is ON*)
        swl00.QX2:=0;                           (*LED indicator inside the key*)
        o201.QX1:=1;
ELSE
        swl00.QX2:=1;
        o201.QX1:=0;
END_IF;
```

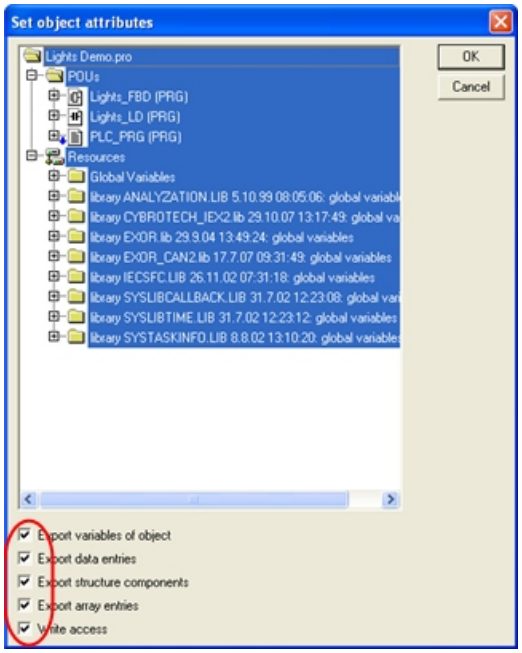
An allocation list must be created so it will be later used when creating a HMI interface for UniOP panel in Designer.

18. Select ***Project*** -> ***Options...***

19. Under ***Symbol configuration*** enable **Dump symbol entries** then click on **Configure symbol file**.



20. Enable all options and end configuration with ***OK***.



21. ***Save*** project.

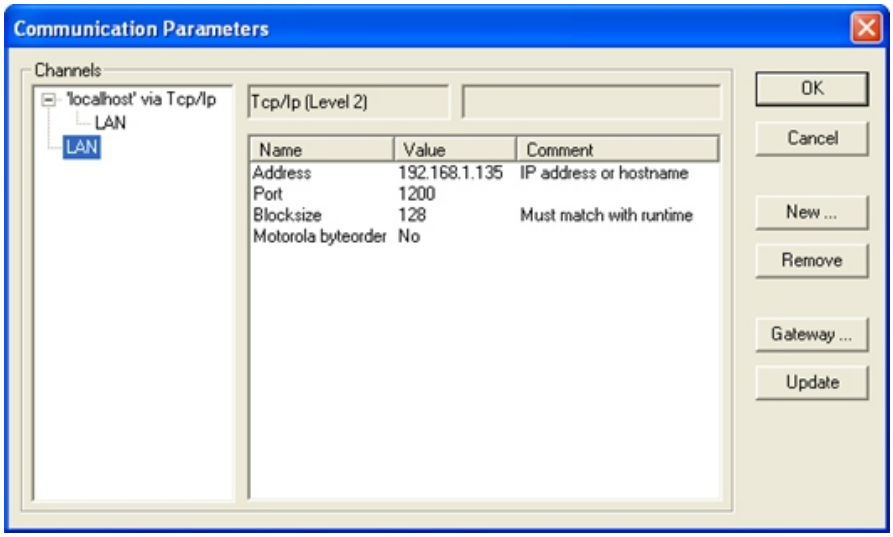
22. Rebuild project by selecting ***Project*** -> ***Rebuild All***.

A file containing defined variables will be created in the directory on the hard drive where project is stored (**Lights Demo.sym**).

Application is now ready to be loaded on the SCM12-C.

23. Select ***Online*** -> ***Communication parameters...***

24. Create new connection. Use TCP/IP **(Level 2)** and input the IP address of the UniOP panel that was set in chapter 3.4.1.



25. Connect UniOP panel to ethernet via RJ45 connector on the adapter and power it up.

26. Download the application into the controller ***Online*** -> ***Login***.

27. Save project in boot sector ***Online*** -> ***Create boot project***.

28. Run controller.

Completed demo project created in this chapter can be downloaded, archived in "*Lights Demo.zip*" file, from ***Downloads*** -> ***Software*** directory found on Cybrotech homepage.

This concludes programming part of the CoDeSys. Next chapter will show how to create an interface on the UniOP panel.

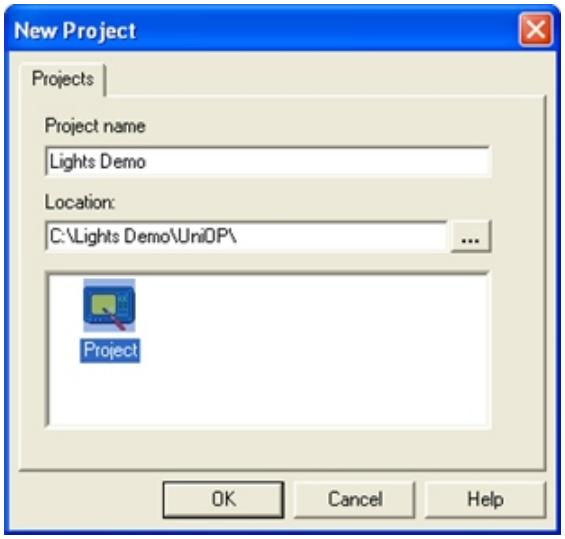
6.3 Creating HMI in Designer

Designer is a software development tool used to easily create a powerful interface between user and automation system. It is a tool made specifically for the programming of UniOP panels.

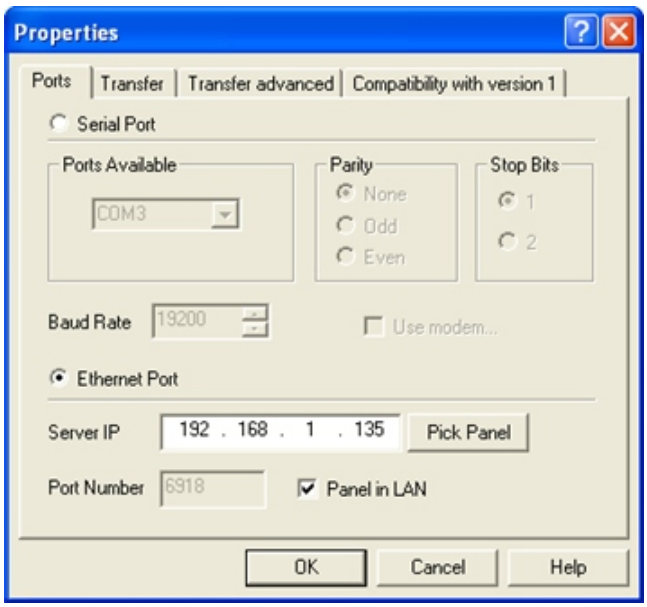
For this part of the guide it is assumed that the user has a Designer version 6.03 installed on his personal computer.

Also UniOP panel model eTOP03 will be used for this demonstration.

1. Start a ***New*** project in Designer.



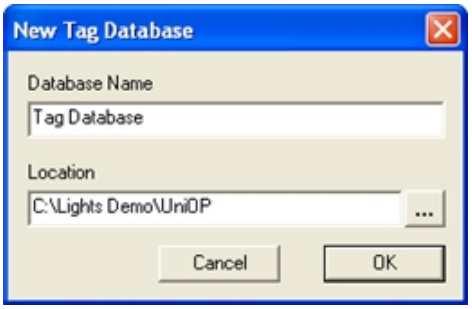
2. Under ***Transfer*** -> ***Options...*** set communication to ethernet port and input the IP address as defined in chapter 3.4.1.



3. Power up the panel.

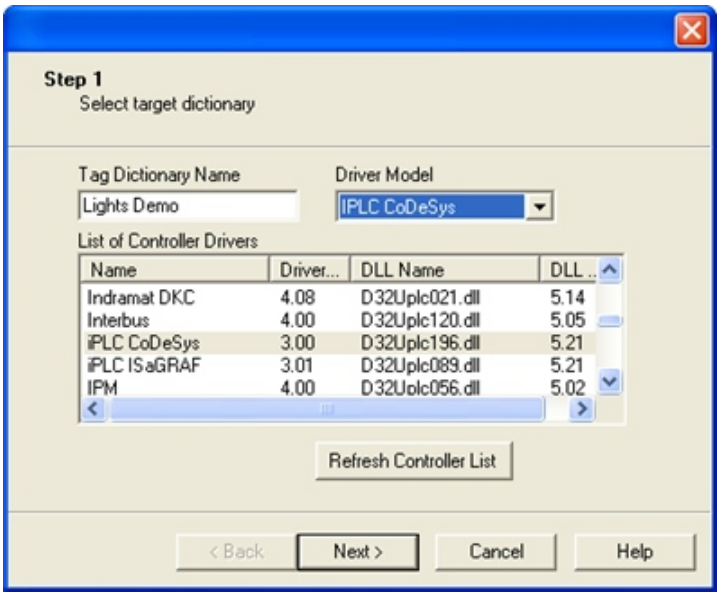
4. Copy panel resources to the project by going ***Transfer*** -> ***Get Panel Resources***.

5. Run ***Tools*** -> ***Tag Editor*** and create new tag database.

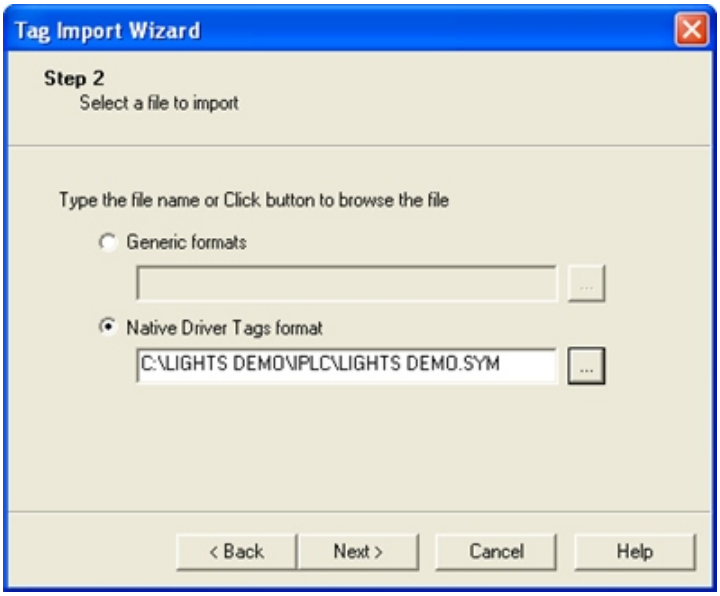


6. Import tags to this newly created database by right-clicking on it and selecting ***Import Tags***.

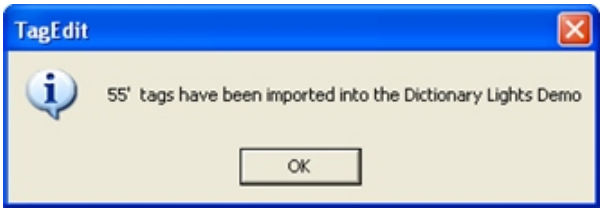
9. Name the new dictionary and select Driver Model as **IPLC CoDeSys**.



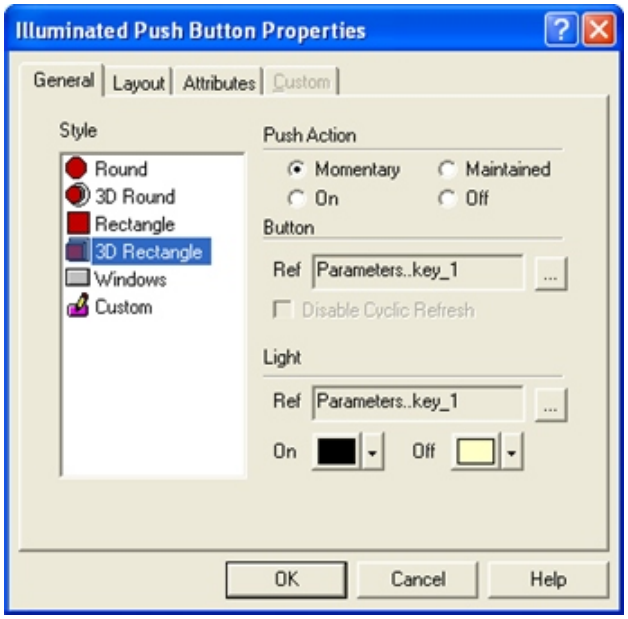
10. In Step 2 locate the tag file created by CoDeSys in chapter 4.2 step 22.



11. After finishing the import wizard 55 tags should have been imported. ***Exit*** the Tag Editor.

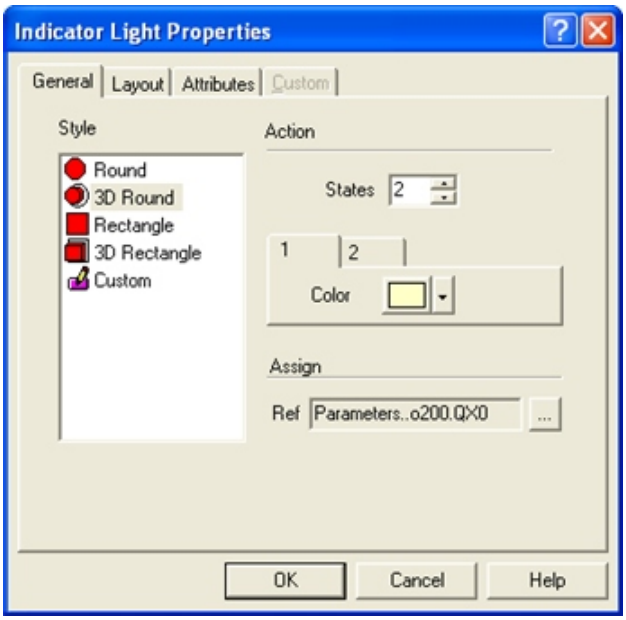


12. On the blank **Page 1** create a new **Push button object**, name it ***Key 1*** and link the button with iPLC tag *Parameters..key_1*. Set push action to ***Momentary***.



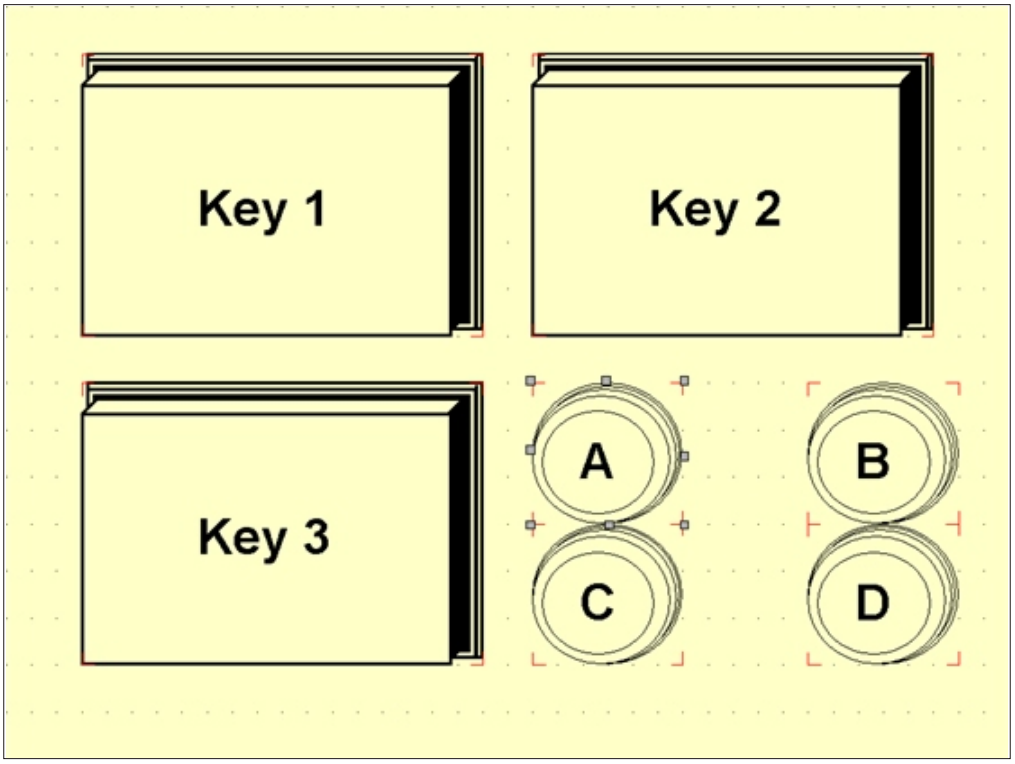
13. Create two more buttons like in step 12 and link them to tag *Parameters..key_2* and *Parameters..key_3* respectively.

14. Place a new **Indicator Light object** on the page. Name it as light **A** and link the object to tag *Parameters..o200.QX0*.

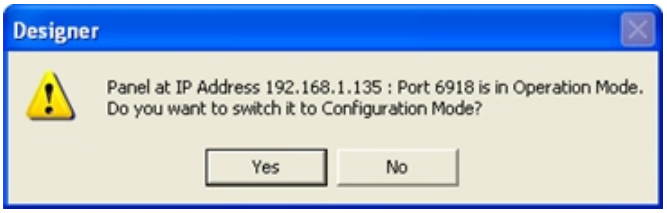


15. Create three more indicators for the remaining lights.

16. ***Save*** project.



17. Download the project to the panel by selecting ***Transfer*** -> ***Download***. If a pop-up window asking you to switch panel to Configuration mode shows, select ***Yes***.



UniOP panel is now programmed and ready to use.

18. ***Exit*** Designer.

Completed HMI project created in this chapter can be downloaded, archived in "*Lights Demo.zip*" file, from ***Downloads*** -> ***Software*** directory found on Cybrotech homepage.

6.4 Testing demo automation system

If all hardware is connected as shown in schematic in chapter 5.1 and both iPLC and UniOP are loaded with application that were created in previous chapters, then we can start using the system.

Pressing any of the keys on the UniOP is equivalent to pressing actual hardware keys mounted in-wall. Also, the current state of each light (ON/OFF) will be shown by indicators A through D.

Pressing in-wall keys will control the lights as specified in chapter 5.1.

Step-by-step tutorial in this chapter is meant to be a learning guide for any first-time user of UniOP panel or CoDeSys programming, and also for users who would like to expand functionality of their UniOP iPLC systems with Cybrotech products.