GSM-1000-BMX Quick Start

1. Prologue

This quick start mainly aims for helping new users to understand how to get started with GSM-1000-BMX and how to create your own project on it as quickly as possible by using our Vector pre-defined templates.

The quick start has three parts. Part 1 introduces how to setup your GSM-1000-BMX along with our TCX2 universal communicating controller. Part 2 describes the way GSM-1000-BMX can work with its pc-side user client "OPENView" as well as an overview of the programming tool kit "FXL". Part 3 contains the general guide on how to use the OPEN Project Builder to create your own project conveniently based on our Vector templates with TCX2s.

For detailed client software "OPENView" use please see the OPENView user manual. For further info about the FXL programming tool kit please consult our technicians.

2. Part 1: Setup your GSM-1000-BMX

The GSM-1000-BMX is a central SCADA platform which is capable of connecting all the sub-communicating devices together. It can gather, monitor and control all the points pulled to it from those devices. So we usually count it as a mini-server which helps managing your site.

A particular structure of GSM-1000-BMX site with TCX2s normally looks like this:
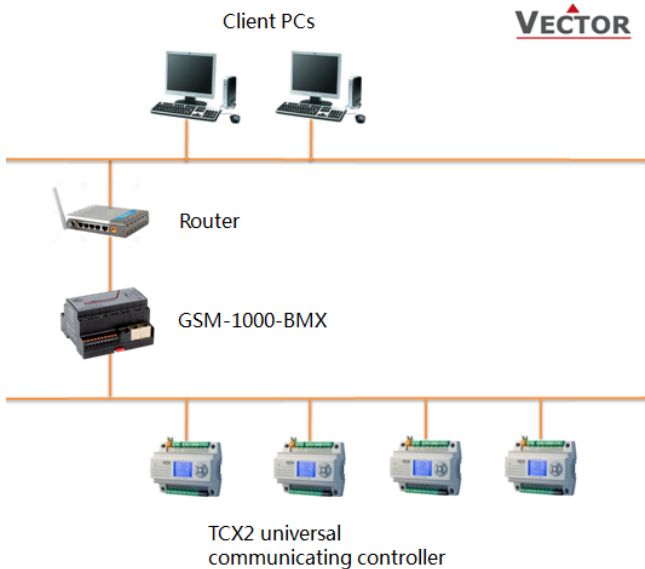
As you can see, it's quite easy to deploy. With RS485 based BAC or MOD protocols a series of TCX2s will be integrated. Have the GSM-1000-BMX placed in your local network then all PC terminals within the range can visit it.

Till here you may still be unclear about how to wiring and configure its parameters to pair with TCX2s. In the following steps we will go through these in detail.

2.1. Wiring preparations for GSM-1000-BMX with TCX2

Please make sure to use **DC** 24 V as power supply for GSM-1000-BMX, and then use an internet cable to connect it with your computer.

Twisted wire needs to be used to connect TCX2s to GSM-1000-BMX via their 485 terminals.

2.2. Setup TCX2s' CO (communication) parameters

Please keep all your TCX2s connected under the same CO parameter settings except for their address and device ID.

For instance we have 2 TCX2s using BACnet MS/TP connected so we keep their baud rate all at 19200 but set their address respectively to 1 and 2 as well as their device ID respectively to 100 and 101. The device ID setting is the very important. The ID must be unique too so the GSM-1000-BMX can identify every device correctly.

2.3. Visit GSM-1000-BMX using computer browser

Switch on the GSM-1000-BMX power until the "run" LED start blinking.

GSM-1000-BMX has a default IP of 192.168.170.100 so what you need to do is setting your own Ethernet adapter's IP which you will use to 192.168.170.xxx.

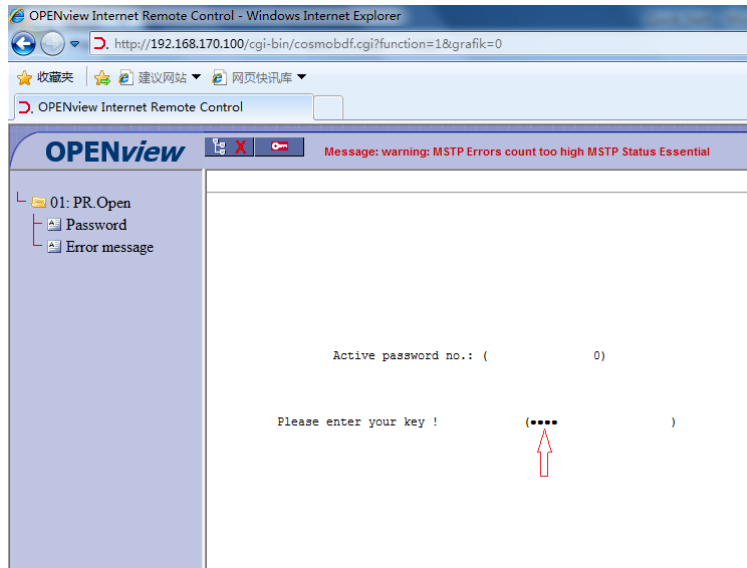For instance we set our IP to 192.168.170.20 with mask as 255.255.255.0 and leave gateway and DNS blank.

Now type in the IP 192.168.170.100 in your browser and press "enter" so it will lead you to the GSM-1000-BMX configuration page.

2.4. Configure GSM-1000-BMX

1. Click "Text Based" to see the configuration tree.
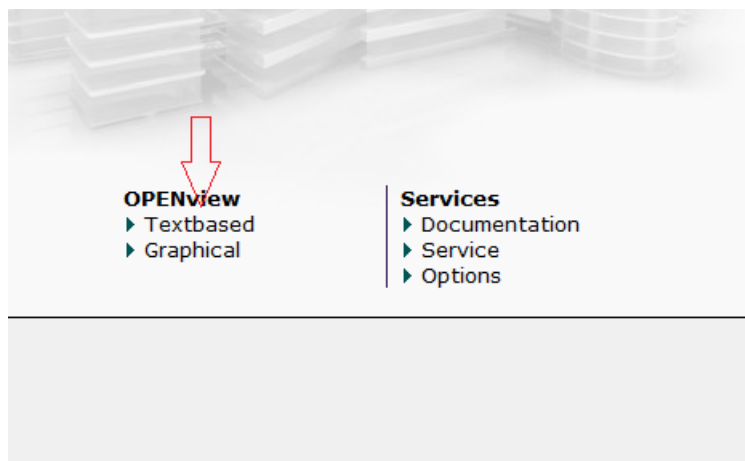
2. Click "password" and input "0007" to login.

3. Then all the items will be listed for you to adjust. The only thing we need to set is the communication parameters, changing them according to the TCX2s CO settings.
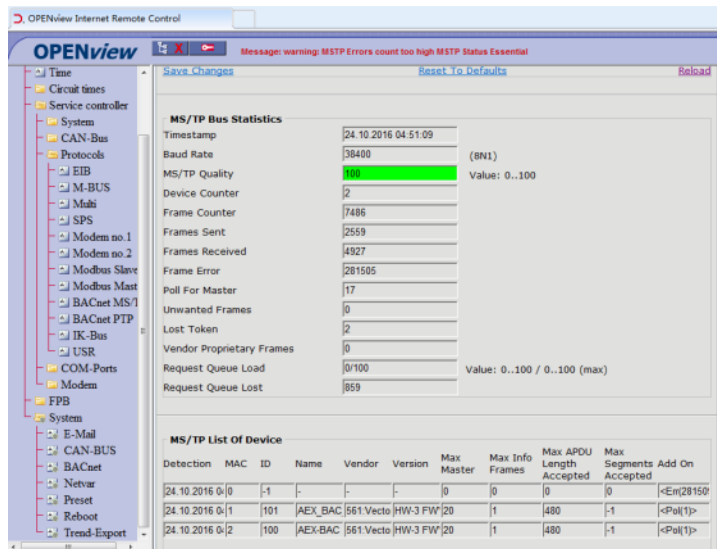
   Click into the CO parameter setting page through "service controller\protocols\..." and choose the corresponding protocol your device used. Since we are using BACnet MS/TP in our case we choose that one. Here we set "Set point" to 3 and "Apply" to 1 to enable GSM's COM3 port (RS485).
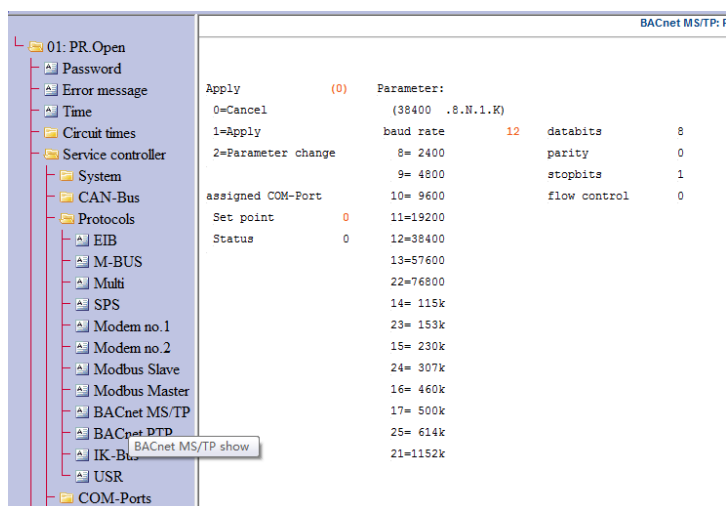
   By using BACnet we also need to go to "System/BACnet", then choose the MS/TP tag and define the "Max master", "Max frame" and the GSM's Mac address for the BACnet network. After all settings done, save and wait for restarting of the BACnet Process.
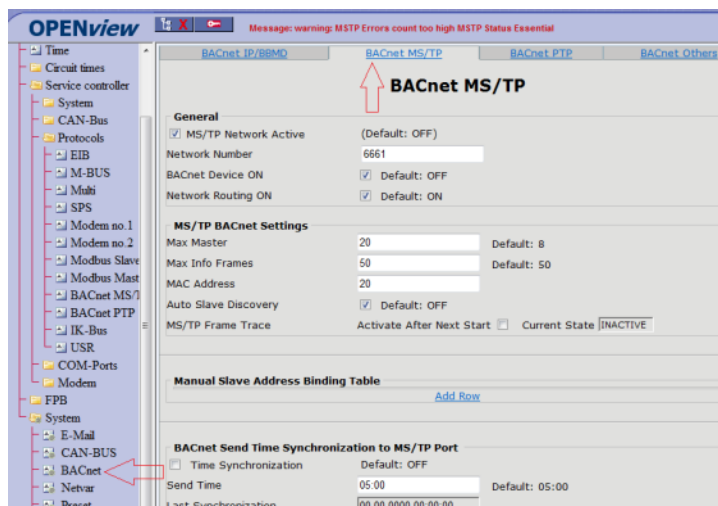
4. Till now the setups are done and you should be able to see all devices in your network including the communicating quality.

3. Part 2: PC Software for GSM-1000-BMX

In part 1 we are already able to discover devices in GSM-1000-BMX. But the points are not pulled to it yet and the GUIs are not done for interaction. In this part we will go deeper and let you know the process of binding points and the basic art of drawing GUIs.

We will need to install 3 software for later uses one by one:

OPENView Control Panel – Free PC Client for visiting GSM-1000-BMX's GUI and default control panel.
FXL – Charged programming kit which creates logic, GUI, pulls points, compile and deploy projects to GSM.
**OPEN Project Builder** – Free PC tool to let you deploy projects based on templates without paying for FXL.

## 3.1. OPENView Control Panel quick instruction

The OPENView is the client for convenient visiting of GSM. It is recommended to install it on a PC with X64 windows 7+. Otherwise its built-in BACnet browser will not work. You can use the executable file "OPENview_ControlPanel_2.0.9" acquired from us to do the installation after you finished setting up your GSM-1000-BMX based on part 1.

Then you need to follow the steps below to add the working GSM controller to your OPENView operation panel.
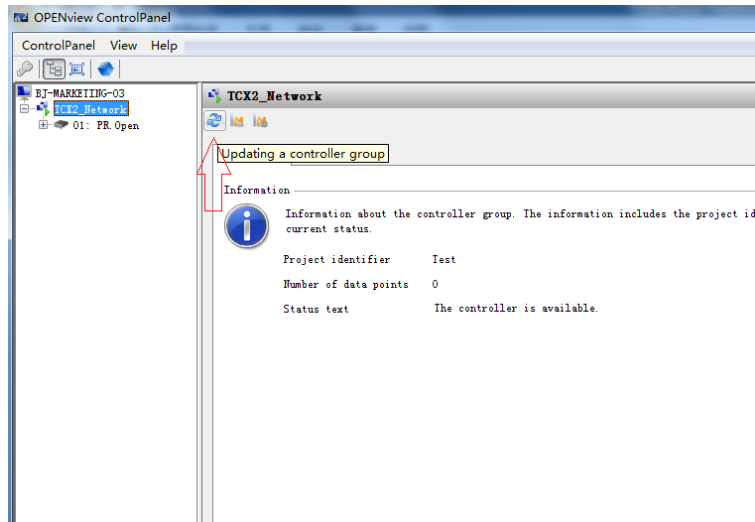
1. Login to the OPENView Control Panel by clicking the "key" icon and then entering "Adminstrator"/"admin".
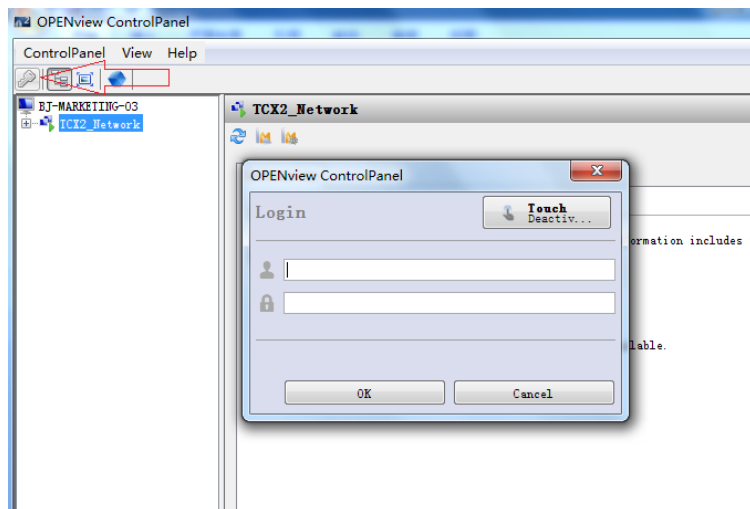
2. Click the "file" icon to create your own GSM controller. Just remember to enter the IP of your GSM controller correctly.
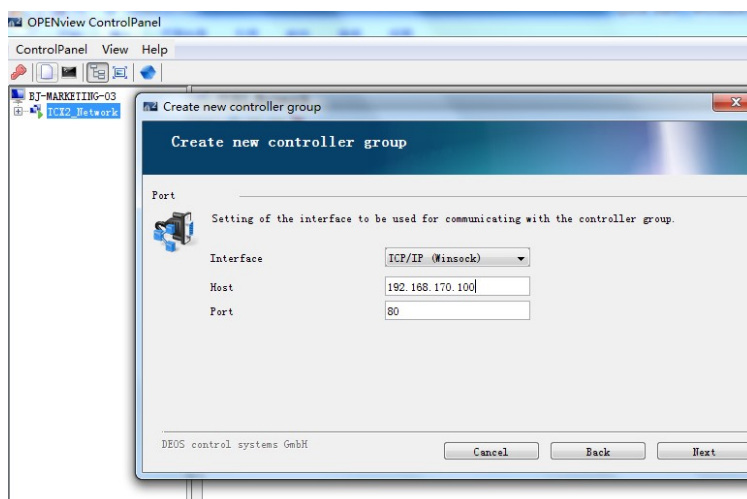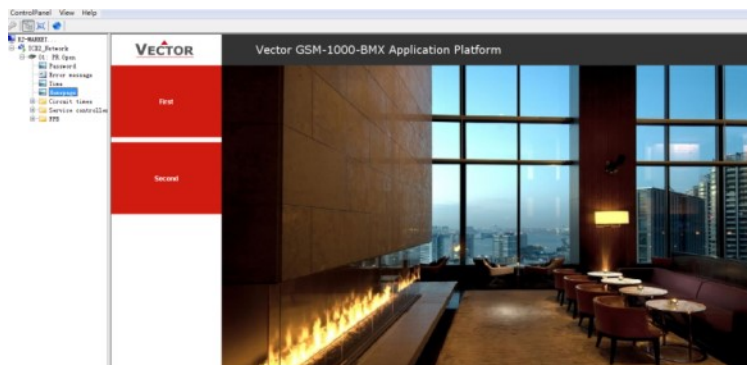
3.   Click "update" icon to load up your GSM settings to OPENView.

4.   After the updating was done, expand the controller icon in the tree left and login using"0007" to your project.
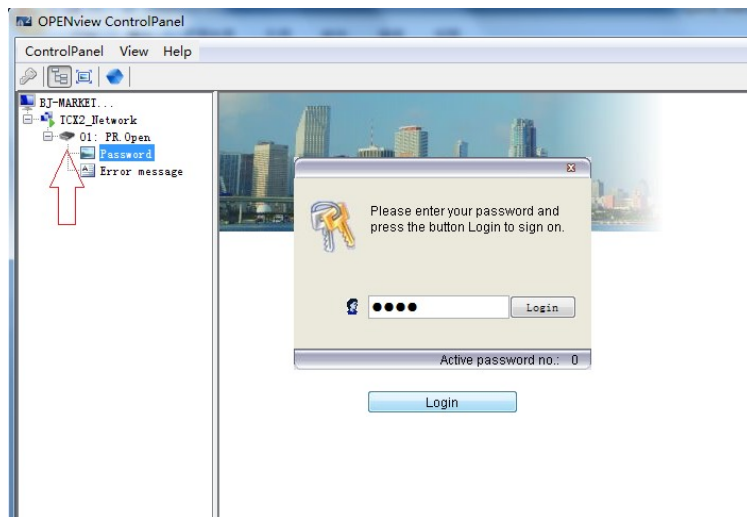
5.   Here the GSM is ready for you to use (surely you need to have all graphics and logics done and upload them to your GSM in advance).

Note: in the tree left you can also do the communication settings like via the browser setting page, but the system setting part is removed here. So you can only configure the system settings completely via the browser visiting.

3.2. FXL programming Tool Kit

FXL programming Tool Kit is the specific programming software for the GSM-1000-BMX. You will need to install this because it's required to upload your programs to the GSM controller.

The dongle key for FXL's full functions is not free of charge, but for users who choose to generate projects based on our templates the key is not a must.

We recommend X64 Windows 7+ as OS and at least 20 G free disk space for the FXL installation. And please use the executable files "FXL_4.53g-1.050c_10_en", "Update_FXLe4.53jV94-1.050c_12", "Update_FXLe4.53kV94-1.050c_14", "Update_FXLe4.53lV94-1.050d_15" acquired from us one by one in the described order for successful installation.

3.2.1. FXL Panel Overview
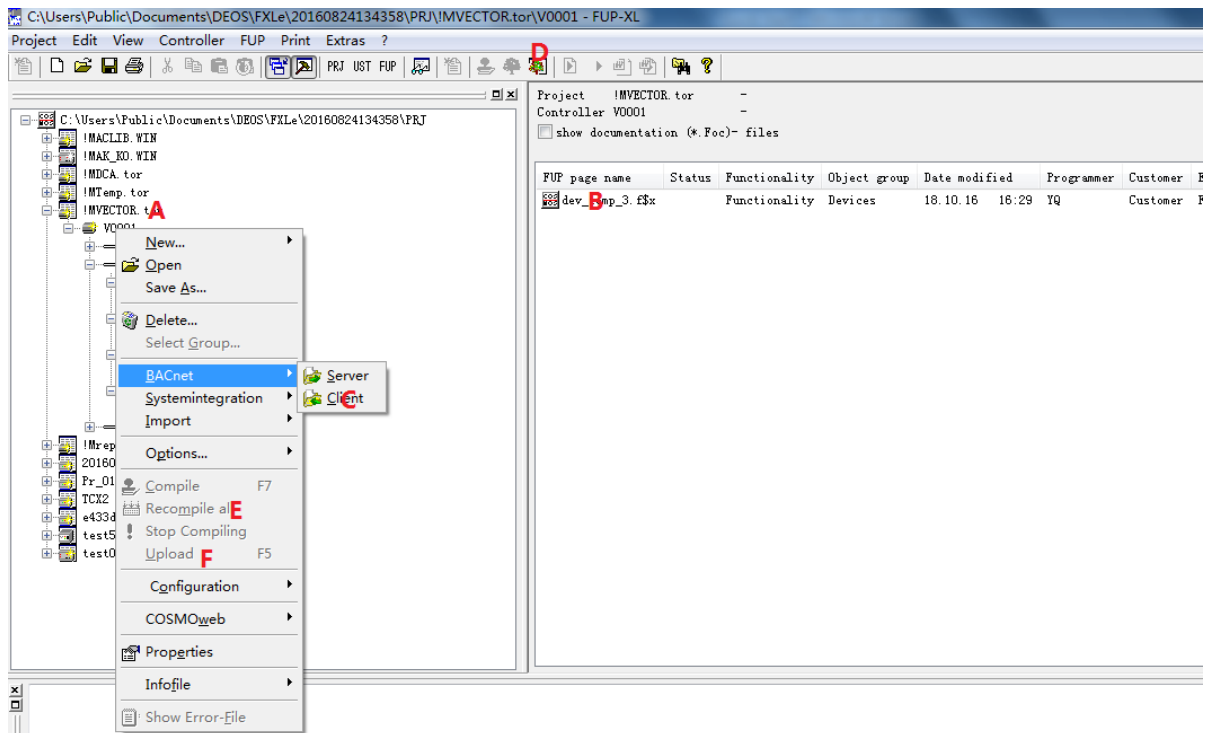

A: Existing projects and libraries saved in FXL.

B: Logic pages in your project.

C: Points adding and binding page.

D: Navigation builder for your OPENView Control panel, after creating new graphics don't forget to add them in the left tree of OPENView from here.

E: Compile function to make your project ready for uploading.

F: Upload your project to target GSM controller. (Dongle needed)

3.2.2. Creating Project

If you choose to make your own project instead of using our template, you will need to create a blank project in FXL first.
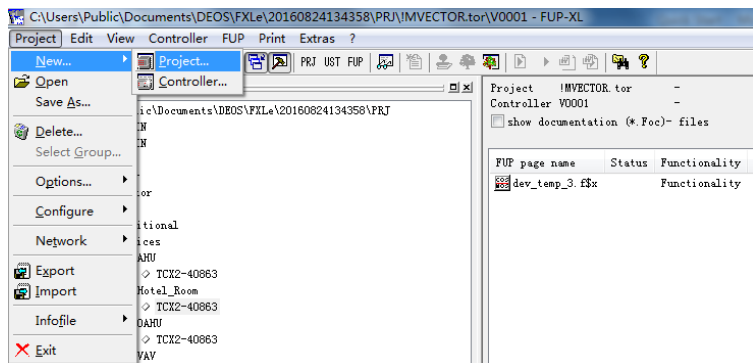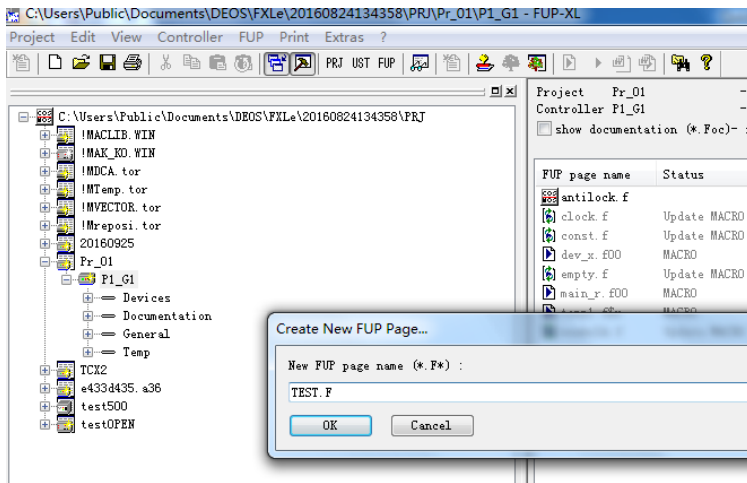

Then add the controller with type "COSMOS 810 OPEN to your project.
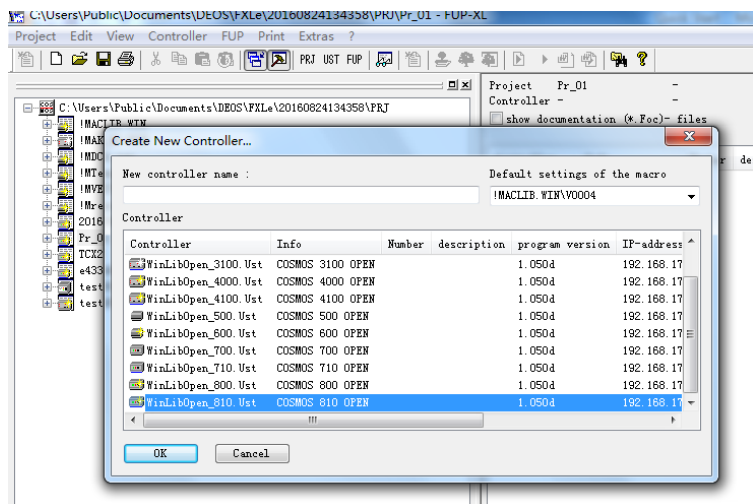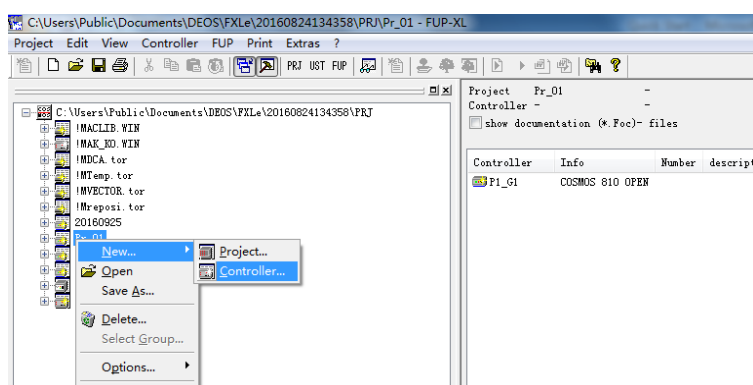
3.2.3. Adding logics and graphics to your project

Now you have your own project, but it's empty. You need to add content holders – FUP pages to it. Those pages will hold all the logics, value indicators and graphics for you to bind.

You can right click on the controller to create new FUP pages.

Then double click on the FUP page you created to open it and you will see:
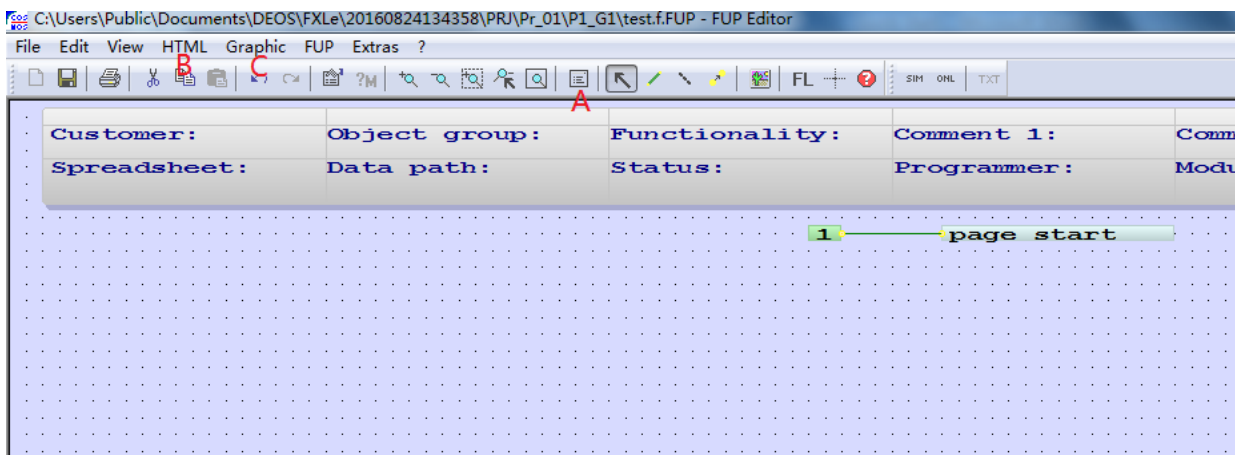
A: Basic elements and modules panel includes all constants, variables and function modules for your logic creation.

B: HTML page, for every bondable element on the field there will also be an HTML indicator in that page for your point bindings later.

C: Graphic editor to create your GUIs in this FUP file.

Since normally all the logics are already set in the TCX2 controllers you only need to add point displays here to prepare the bindings. Thus we first click icon "A" to open "basic elements and modules" panel and drag a "add" block to the field from math category.
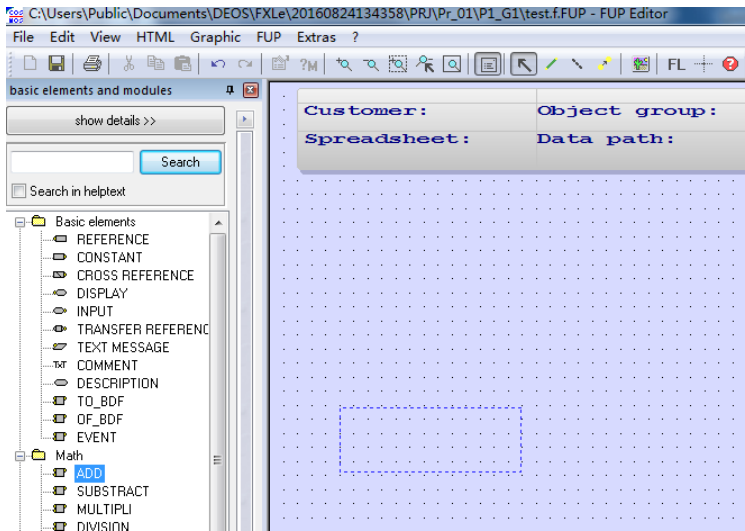

Then drag a constant 0 and a display element from Basic elements category to the field and bind them as below. The "DO1" is the "display" for point binding later.
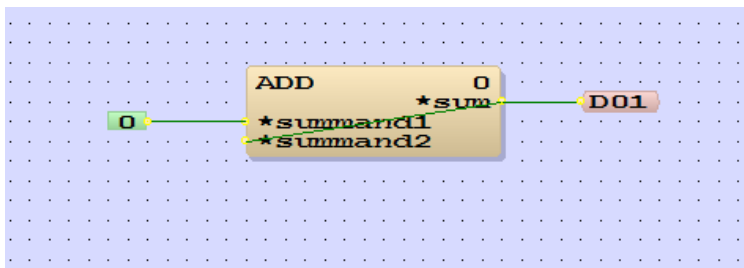

For BACnet protocols right click on "DO1", change the "activate BACnet" to 1 and set "Object type" as well as the tickable items below accordingly based on which point you want to bind to using this "display".
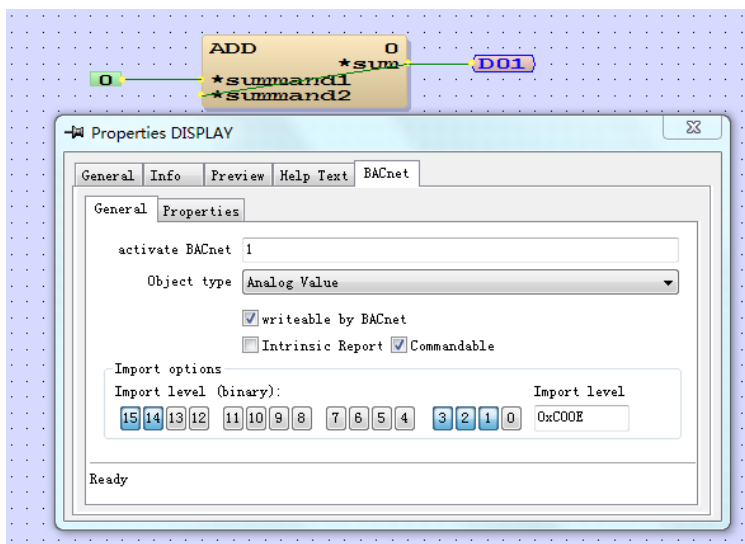
Now open the HTML page from "B". You will see the "-ff" HTML indicator for you to bind to the real point later.
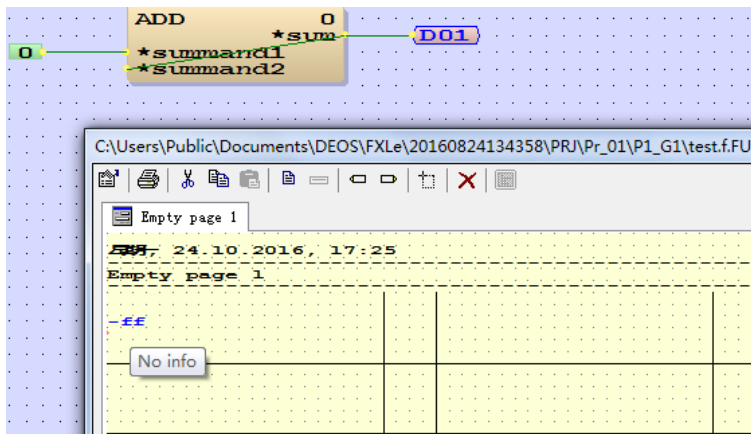

Open the graphic editor by clicking "C" to create a graphic for demonstration.
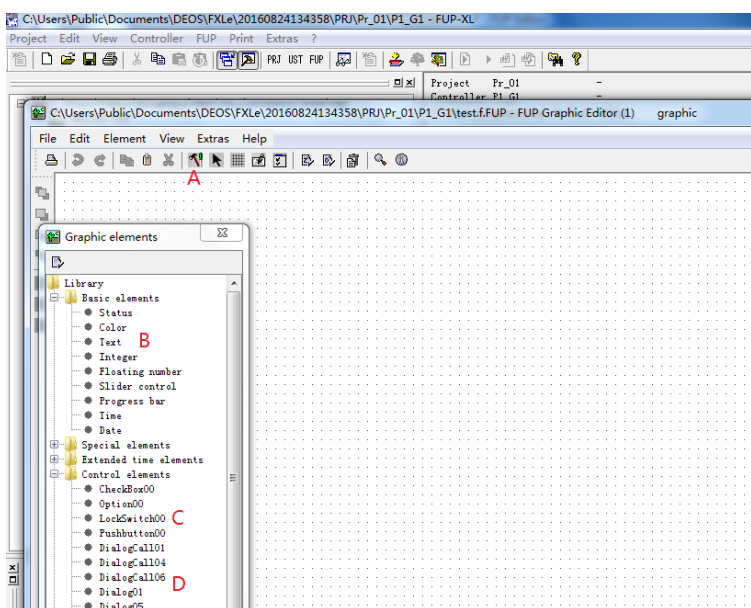

Most used elements for beginner:

A: Show/hide Graphic elements panel for your drawing

B: Basic graphic elements for value showing, value interacting, line and polygon drawing, picture inserting and etc.

C: Buttons

D: Page jump and popup

For example, to show a point on the canvas, you just need to create a value indicator on the field. If you want to have decimals then use the "floating number" indicator.
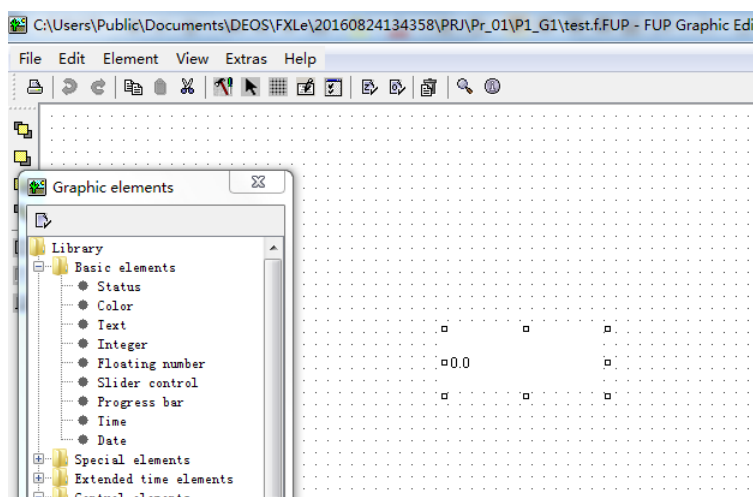
Right click on it and select property and click "…" after the value field of "Name" parameter:
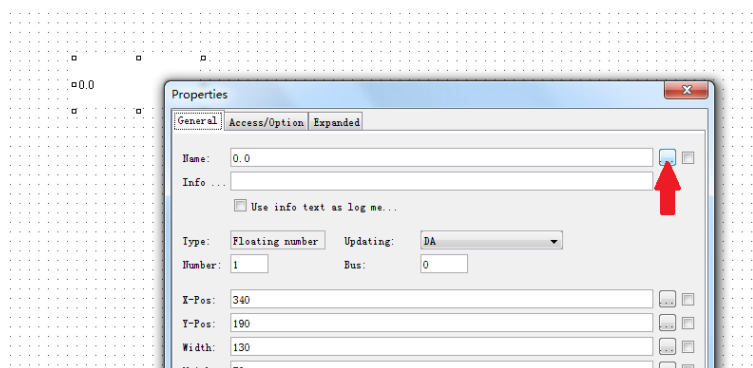

And it will lead you to the logic field and let you to choose which logic display's value you want this indicator to show. We want to display DO1 so we click on it. At last you could see the relation appears in the value field of "Name".
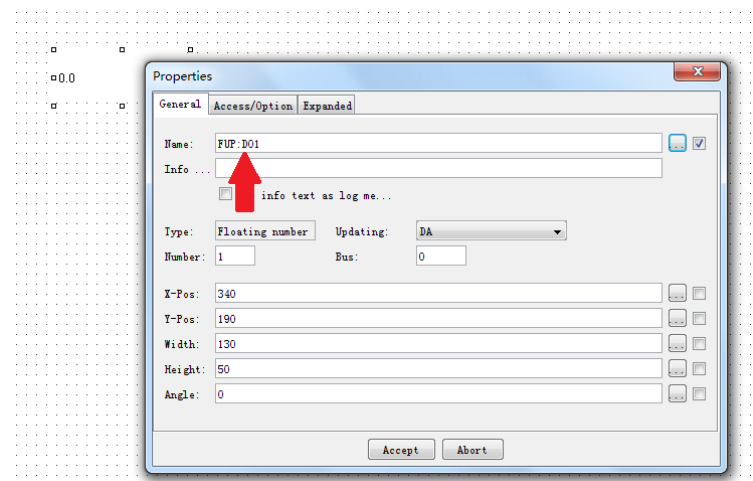
Now all the point binding preparations for DO1 are finished. **Note that for every point you would like to pull up and bind later you will need to create a display respectively repeating the steps above**

Don't forget to click "save" at the end:
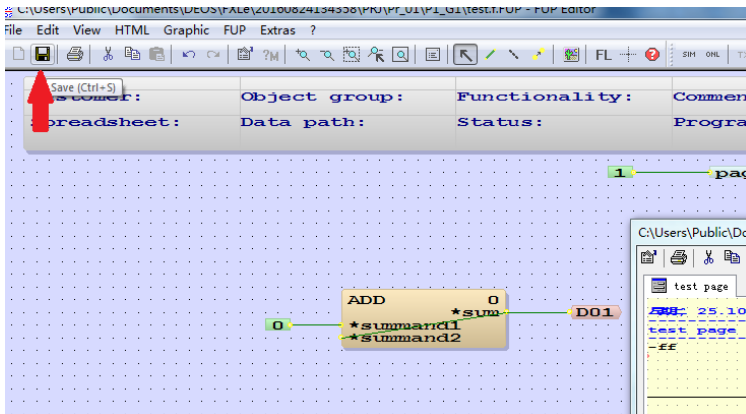
3.2.4. Navigation builder for tree customization and graphic page/HTML page insertion

Navigation builder helps you adjust your tree leaves to be shown in the client. After your new FUB page and logic creation both the new generated HTML page and the new graphics are not auto included in your tree. So it's important not to forget to add them, as HTML is for point binding and graphics for demonstration.

Click on the "Edit local HTML tree" icon on the main view to open the navigation editor:
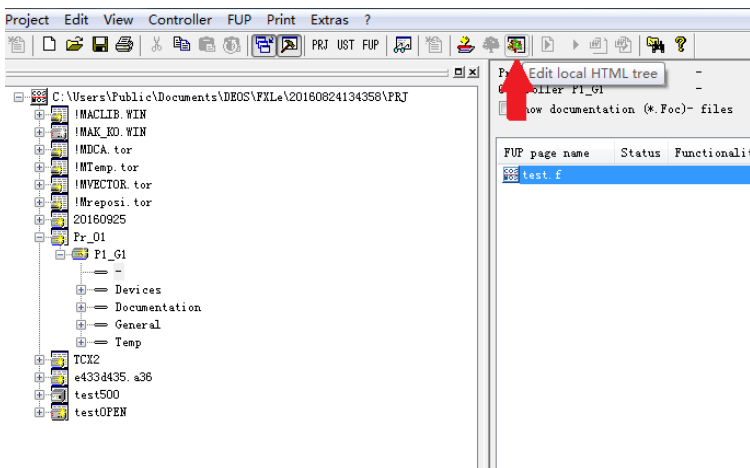
And drag the HTML page and graphic page created from the right side into the left tree:
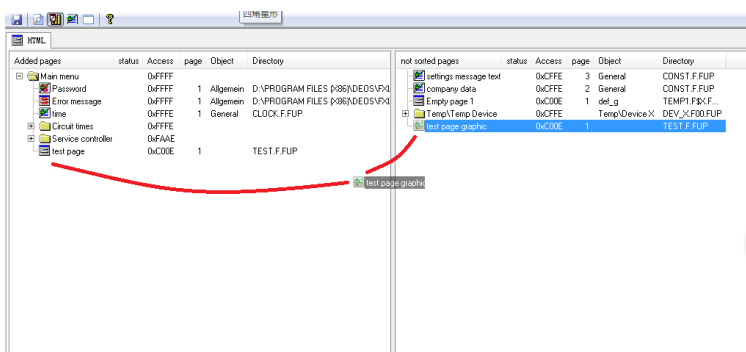
At this point the HMTL page is ready for point bindings and the graphic pages will be seen after uploading the program to your controller.

3.2.5. Points adding and binding page

After you finished all the preparations above, you can start point binding process.

Right click on the controller under your project and choose BACnet/client for the slave controller point pulling (for Modbus choose systemintegration/Modbus instead):
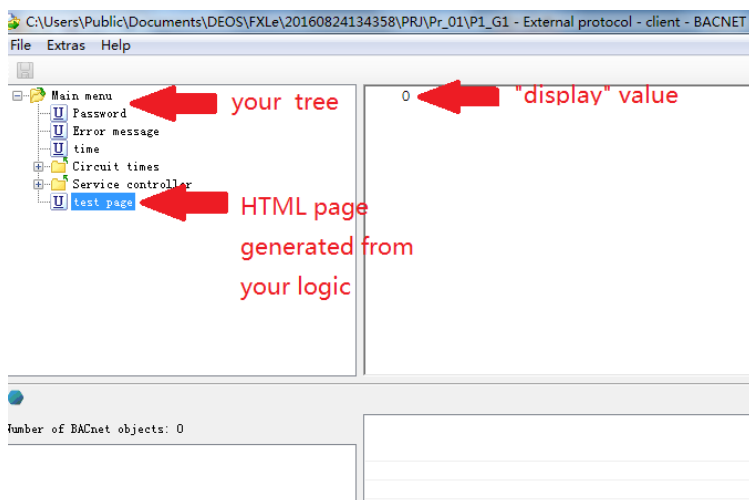

Now create your slave device for point binding by click file/new on the menu to create a slave controller.
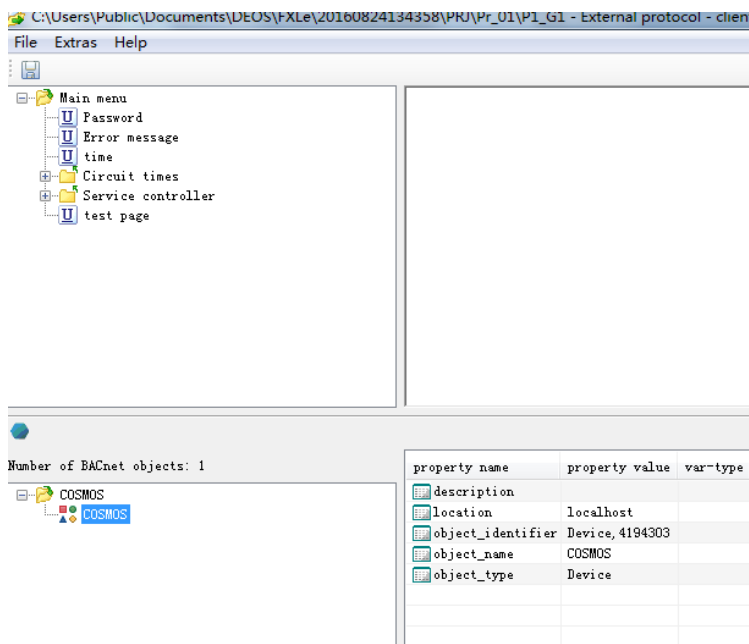
Then choose your HTML page and drag the display value (A) into the controller you created, change your controller device ID (B, important for succesful binding) and your controller name (C).
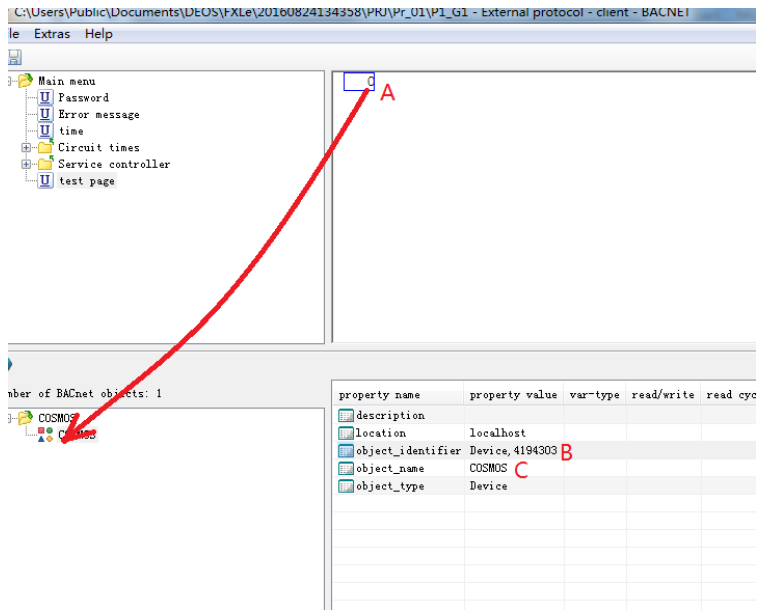
Go to the display value you inserted, you can change its name (D), its type and address (E, important for successful binding) and its refresh rate/read/write access (F).
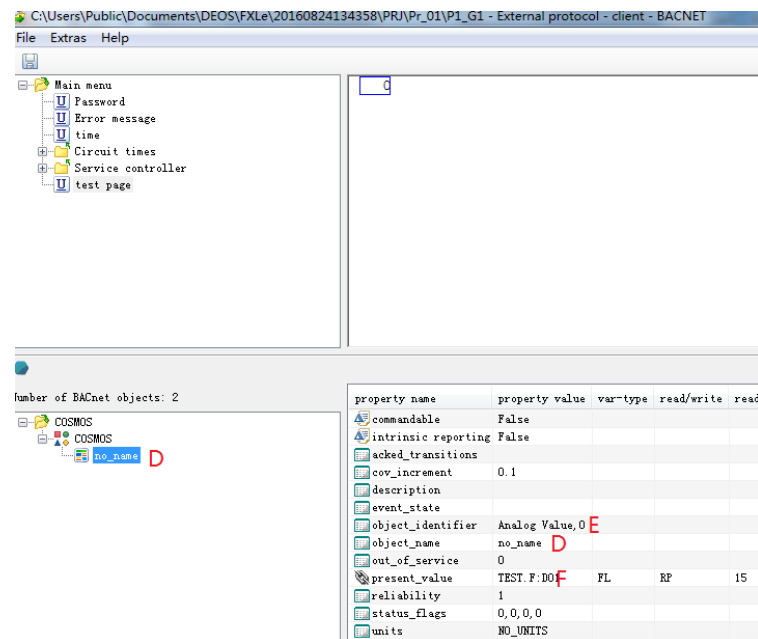

Each display value in the HTML page can be assign to a controller and only be bound to one point, so repeat the above steps for all your displays respectively.

3.2.6. Compile and upload

Once you completed the logic/graphic creations and point bindings you are finally able to compile your project for deployment.

Right click on the controller and choose "recompile all" to start the compile process.
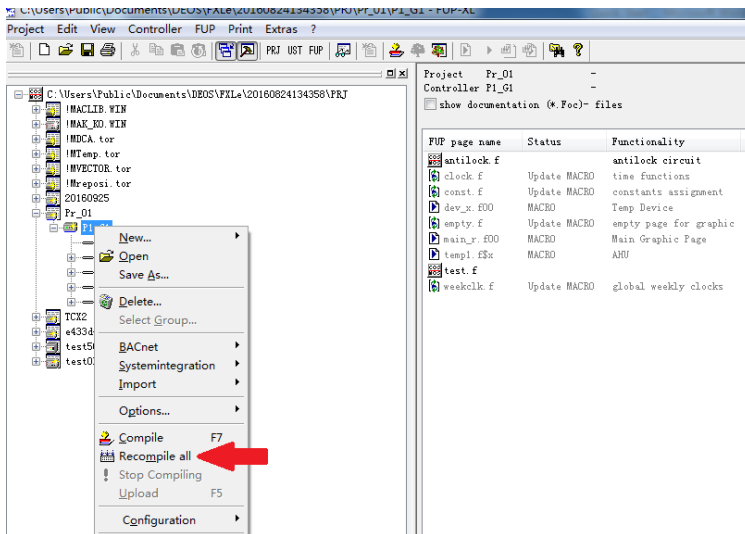

If there are errors in your logic it will hints you with red highlight telling where they are for your easy check, otherwise just go through the compiling process step by step.
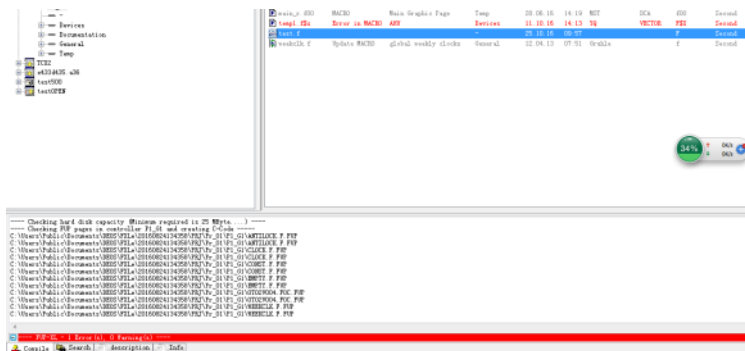
Upon finish you can right click on the controller under your project again and choose upload. Before starting it will ask you for the IP of target GSM controller, please fill in correct IP for successful upload.
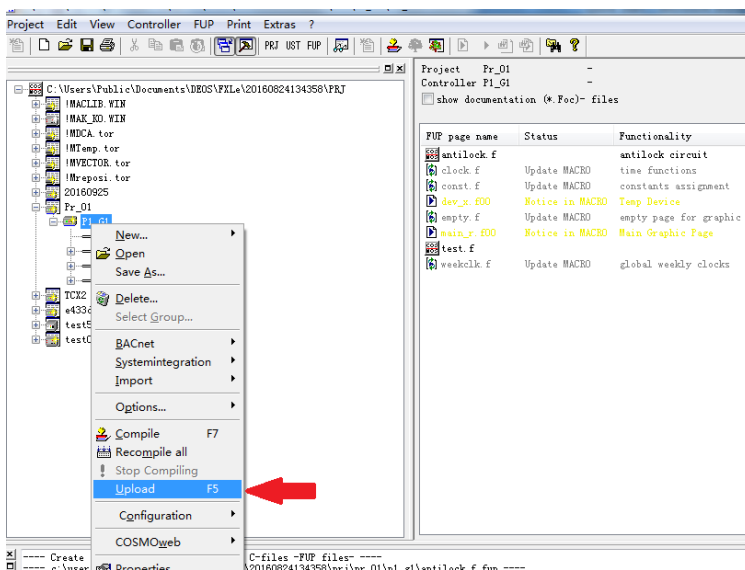
It will take a while for the controller to restart once a program is uploaded (dongle required). When it's done, start OPENVIEW, click "update" and then you will be able to login and see your own project contents.
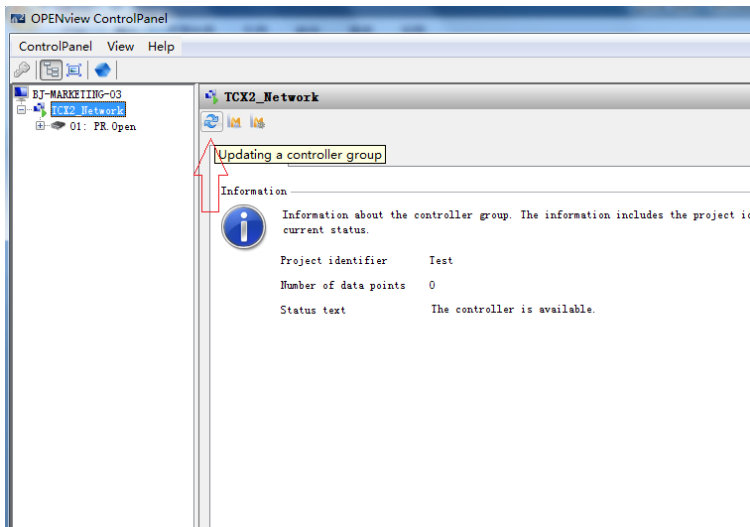
# 4. Part 3: Fast project creation and deployment

Because the upload function of FXL requires a purchased dongle from VECTOR, for your convenience we also offer you a fast and cost effective deployment solution based on our TCX2 project templates which doesn't need the dongle by using the free PC client software OPEN Project Builder.

4.1. Overview of the Solution

The solution involves 3 things: the FXL installation, the OPEN Project Builder installation and the VECTOR project template file import to FXL.

How it works?

The OPEN Project helps you create your Project hierarchies.
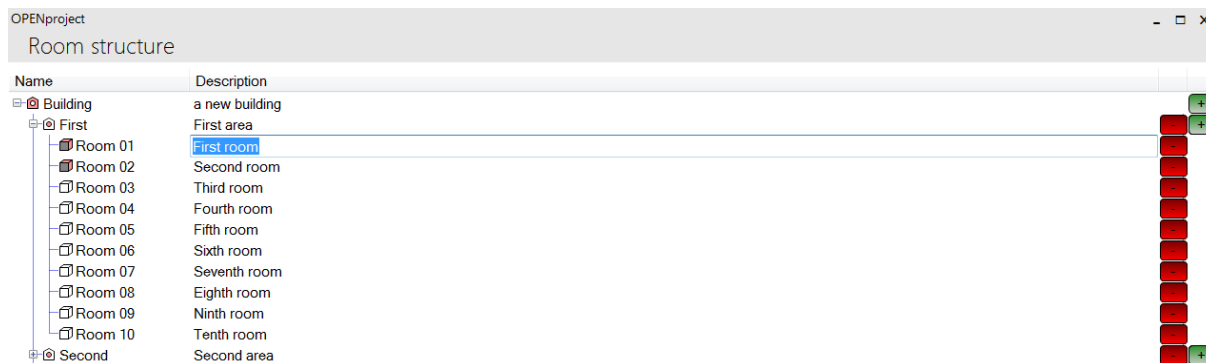
And let you assign different types of configured TCX2 controllers we prepared for you by dragging and dropping.
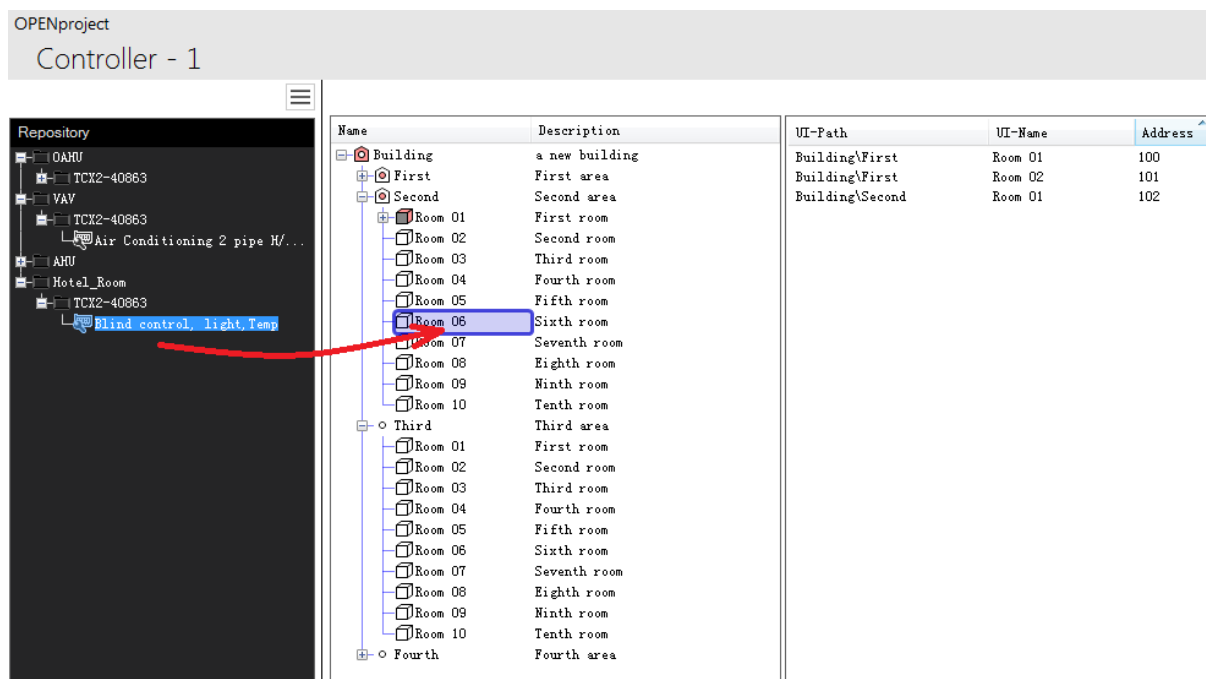
You only need to set your TCX2's device ID correctly.

By compiling it calls the FXL and uses the VECTOR templates for project generation. All the GUIs and bindings are completed in advance. You don't need to create GUIs, point logics and bindings like in part 2 at all.

After compiling it will generate an executable project file for you to upload the program to target GSM controller.
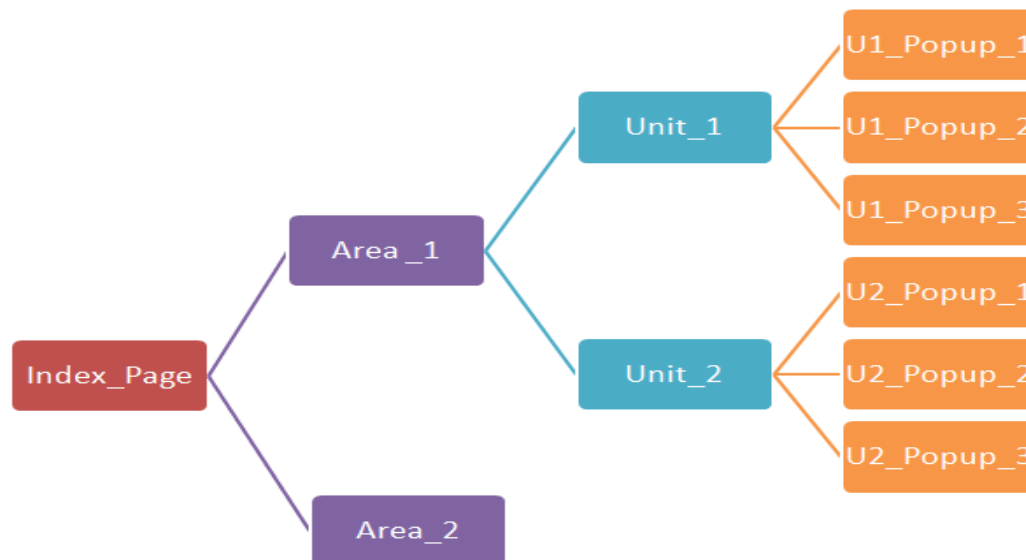
4.2. VECTOR Project Template

Vector Project Template is a finished FXL template project for OPEN Project Builder to call from. It contains GUIs, TCX2 controller applications and corresponding bindings. You can acquire it from us.

To import it, start your FXL and click "import" in menu "project". Then choose our template file "!MVEVTOR.tor" for it to be uploaded.
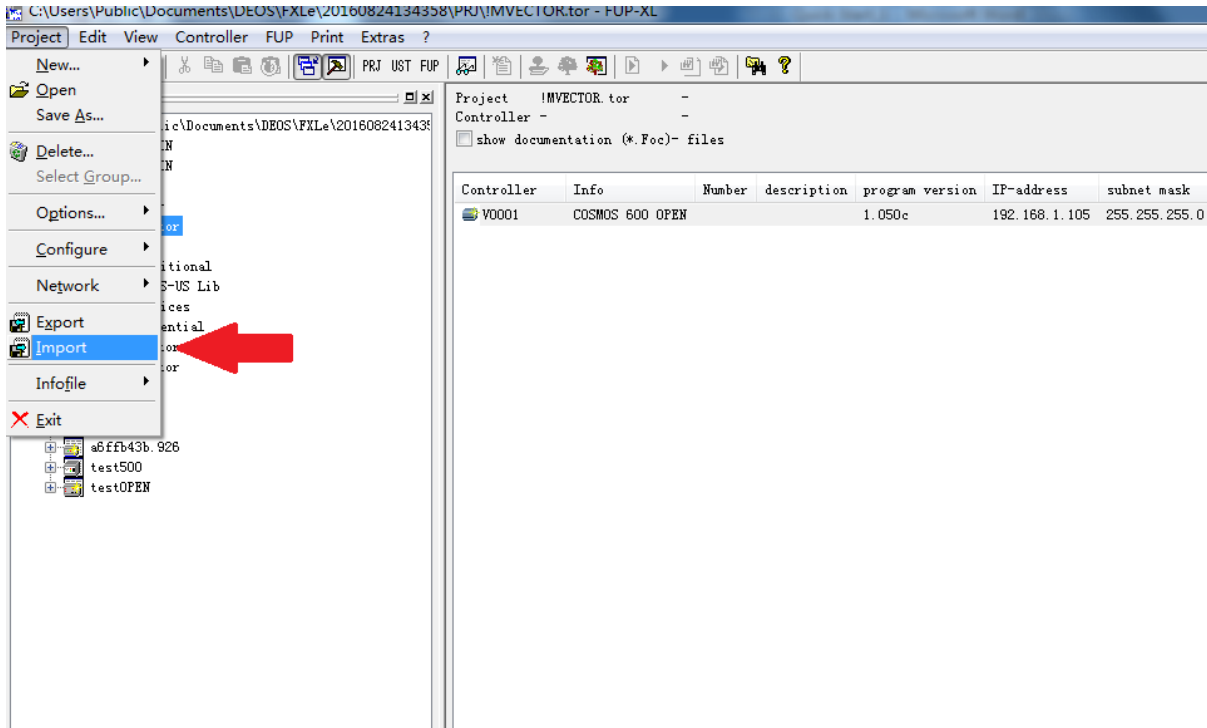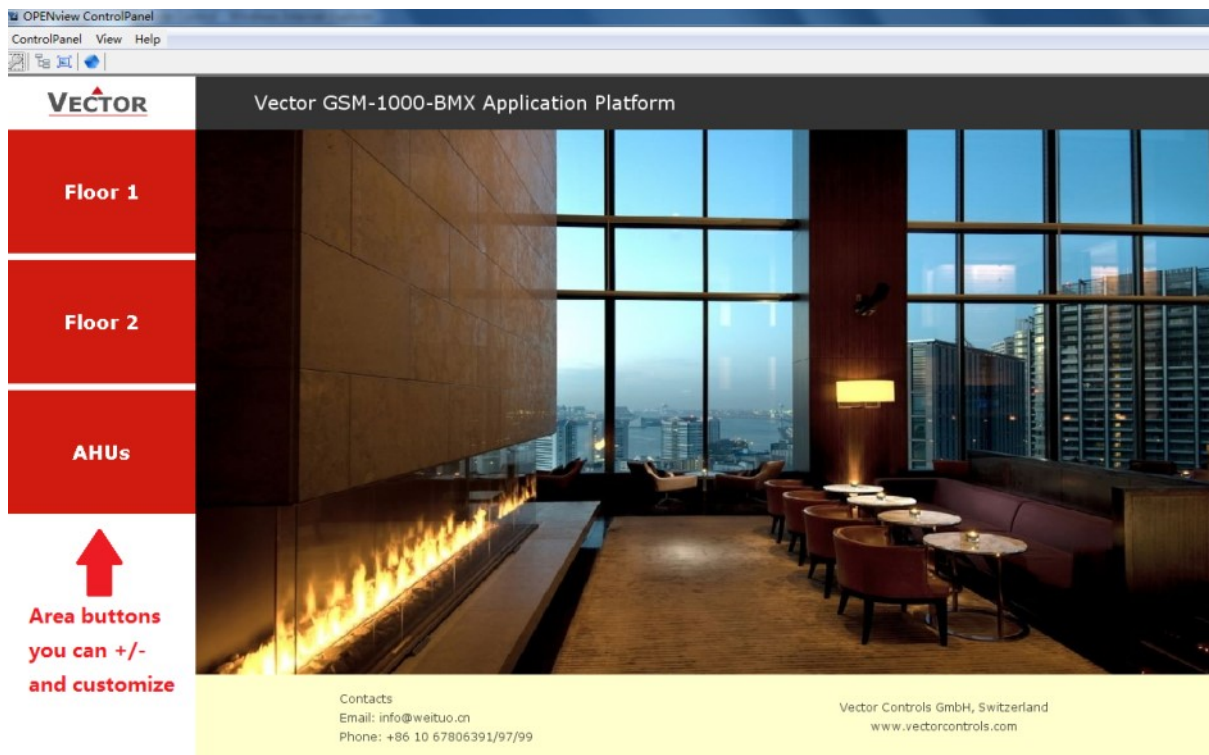

4.2.1. Project structure

The Vector project template has a four level hierarchy. The "Index_Page" contains area buttons to enter each area. In each area the chosen applications will be generated dynamically based on your assignment in OPEN Project Builder.

4.2.2. Index Page

Below is the structure we will build quickly using OPEN Project Builder. We use this sample structure for your easy understanding.

The generated index page can be viewed in OPENView after uploading.
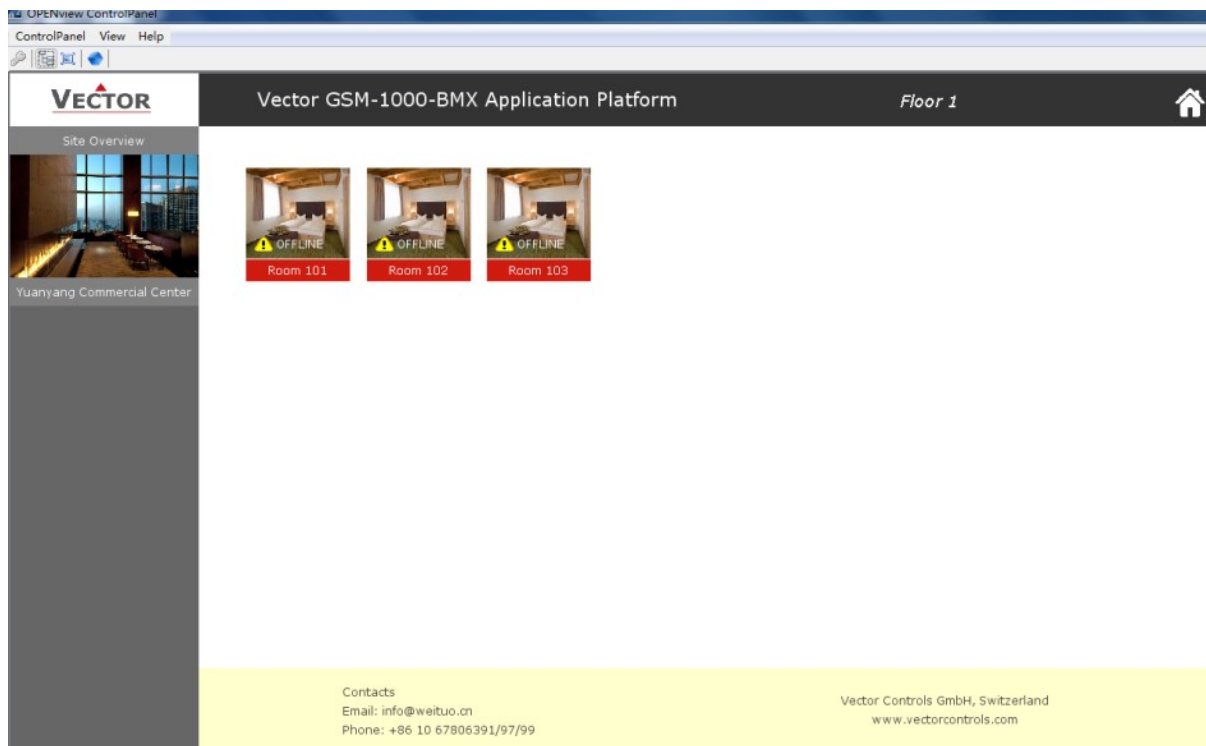

Source Graphic in FXL: 1_Hompage of Area.f$x.

You can customize your area buttons there.

4.2.3. Area Page

You can enter each area to see the units in it by clicking the area button.
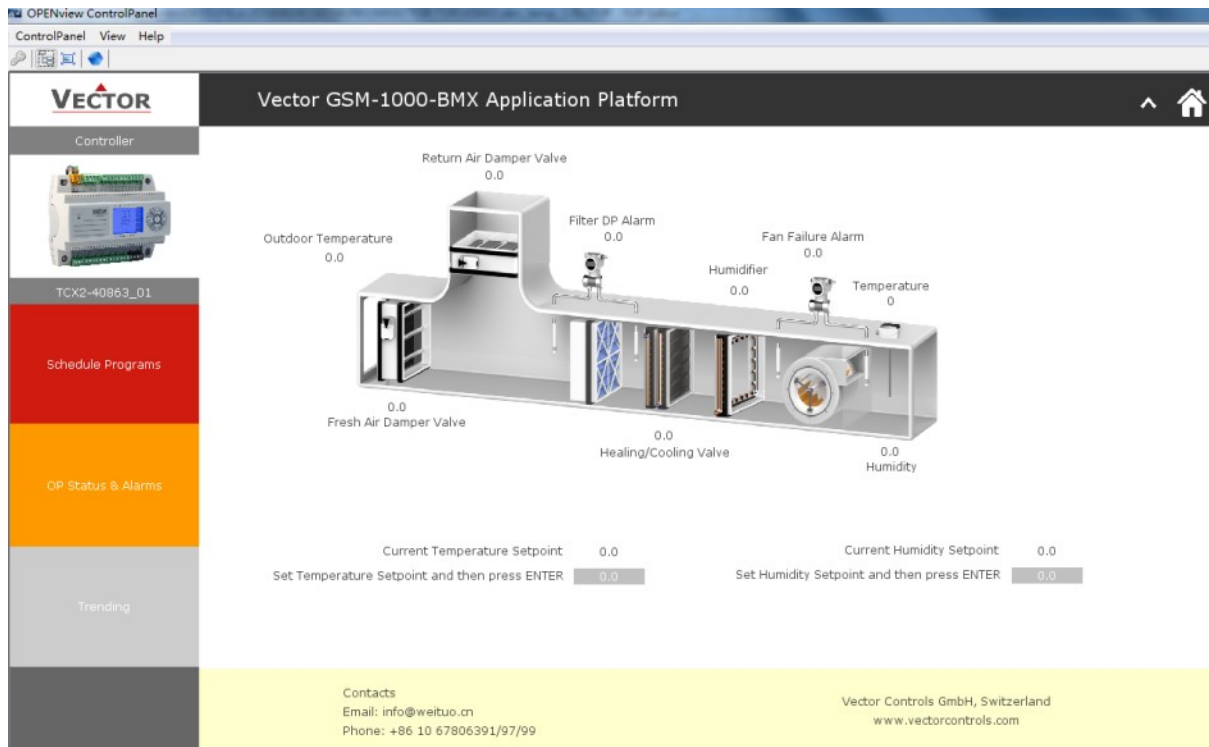

Source graphic in FXL: 1_Homepage of Home.f$x.

You can re-order the dockings of those application icons in there in advance.

41_DialogCall of dev_temp_x.f$x is the icon graphic for you to edit. "dev_temp_x.f$x" is the logic and graphic FUP file for a concrete application. For every application there is one. Go to that file and you can edit its application icon, bindings and application graphics.
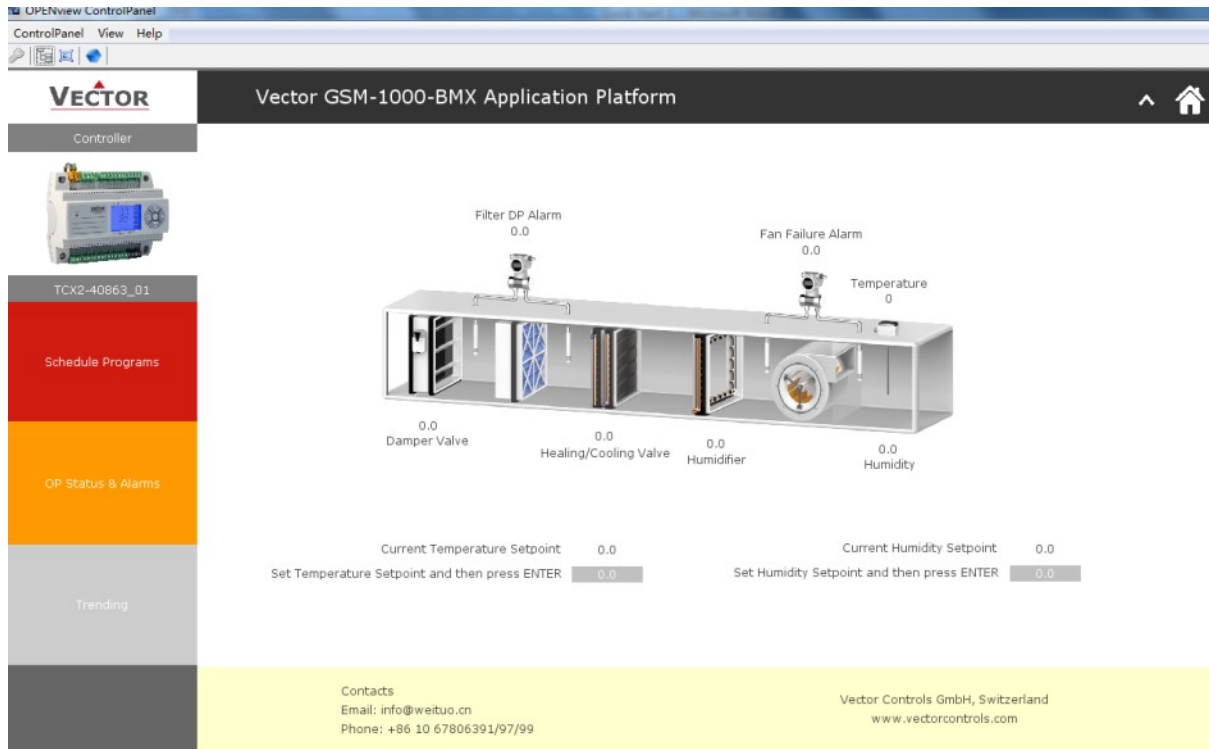
4.2.4. Unit Page

By clicking the application icon you will enter that unit's monitoring page. Currently we have prepared 4 application templates for you: AHU, OAHU, VAV terminal and hotel room as below。 You will have an overview of that unit's running status there and can send commands to adjust changeable point values.

AHU

OAHU

VAV terminal

Hotel Room

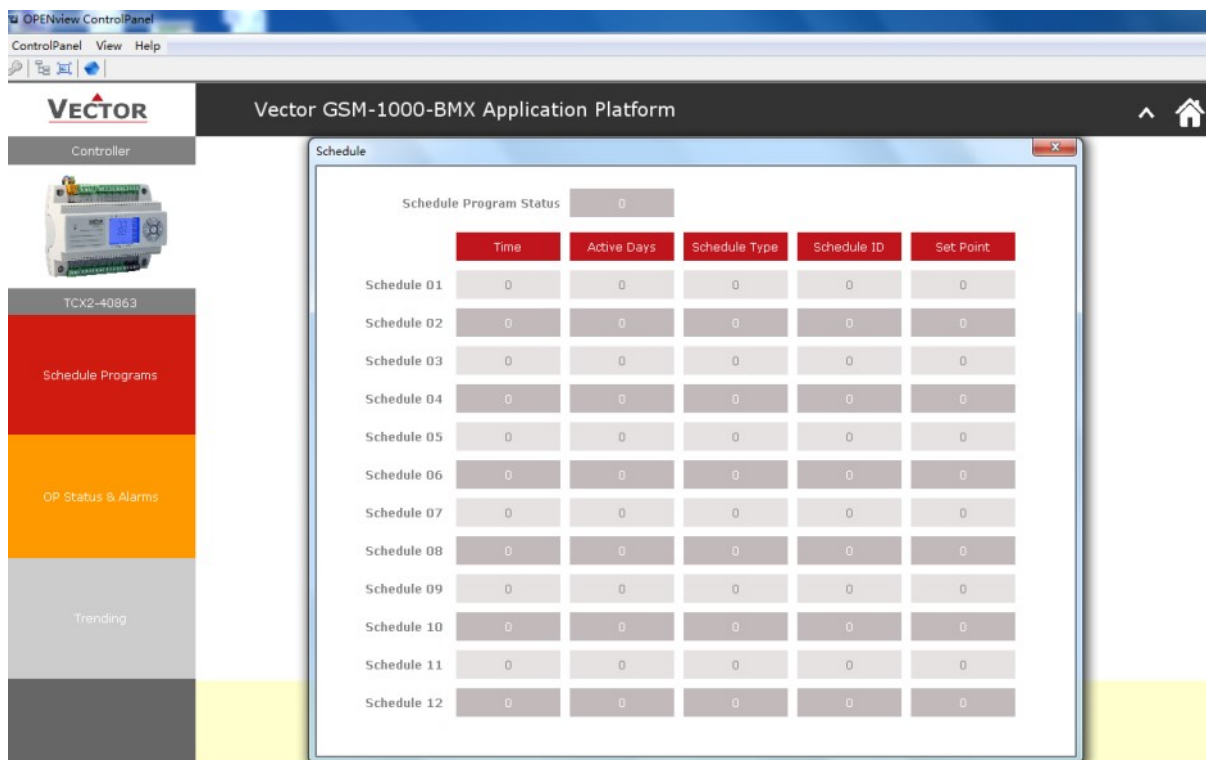Source graphic in FXL: 1_Home of dev_temp_x.f$x, 48_Display_Main of dev_temp_x.f$x

1_Home contains the main background and button bar for each application. 48_Display_Main is the application graphic area for you to edit.

4.2.5. Popups

In each application page we have prepared 3 function blocks to call out popup windows for you based on TCX2's utilities: the "Schedule Programs", the "OP Status & Alarms" and the "Trending".

Schedule programs

In this popup you can remotely manage the 12 schedule programs offered by a TCX2 controller. The screenshot below is the example interface for –MOD TCX2s. For –BAC TCX2s the interface is simplified because of its address pointing but the managing capability is the same.

Source graphic in FXL: 2_Schedule of dev_temp_x.f$x (for –BAC TCX2s the source graphic is 6_Schedule)

OP Status & Alarms

In this popup the controller operation status and all the alarm status are listed.
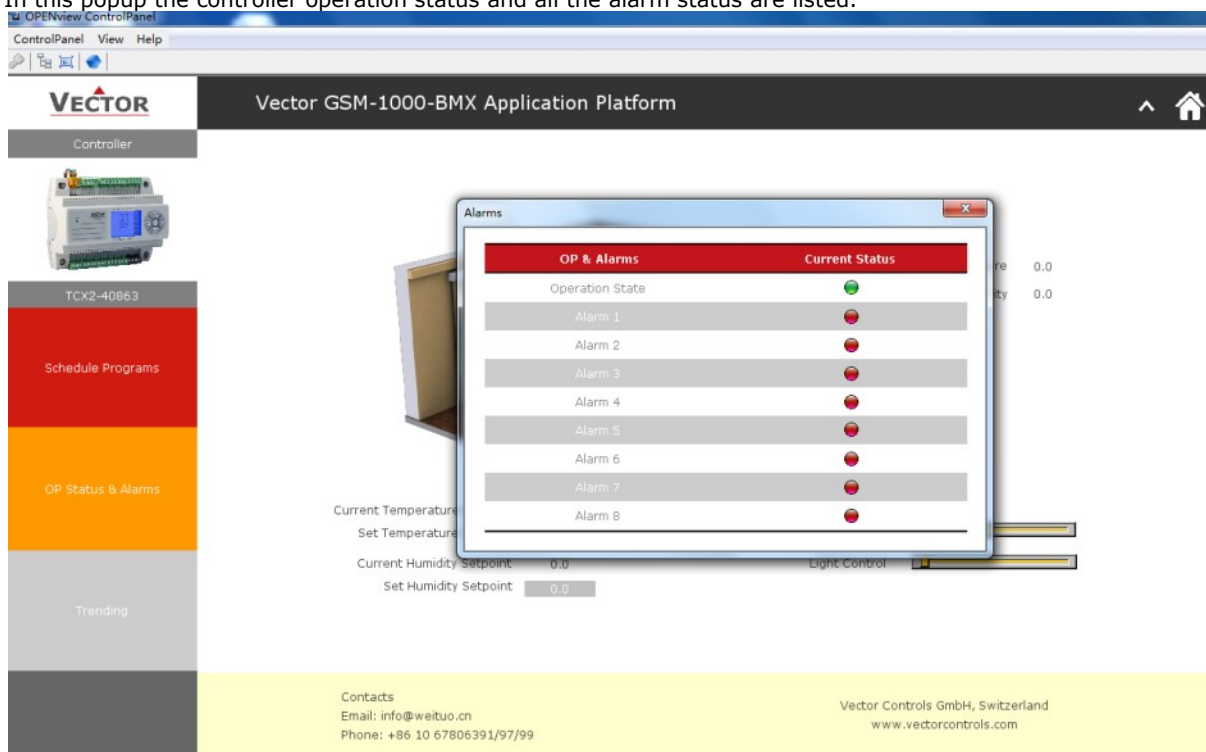
Source graphic in FXL: 3_Alarms of dev_temp_x.f$x

Trending

In this Popup the trends of 2 chosen points will be displayed. By default the temperature and humidity are recorded. The trend can hold up to 10240 sample records (when exceeding, the newest will overwrite the oldest one).
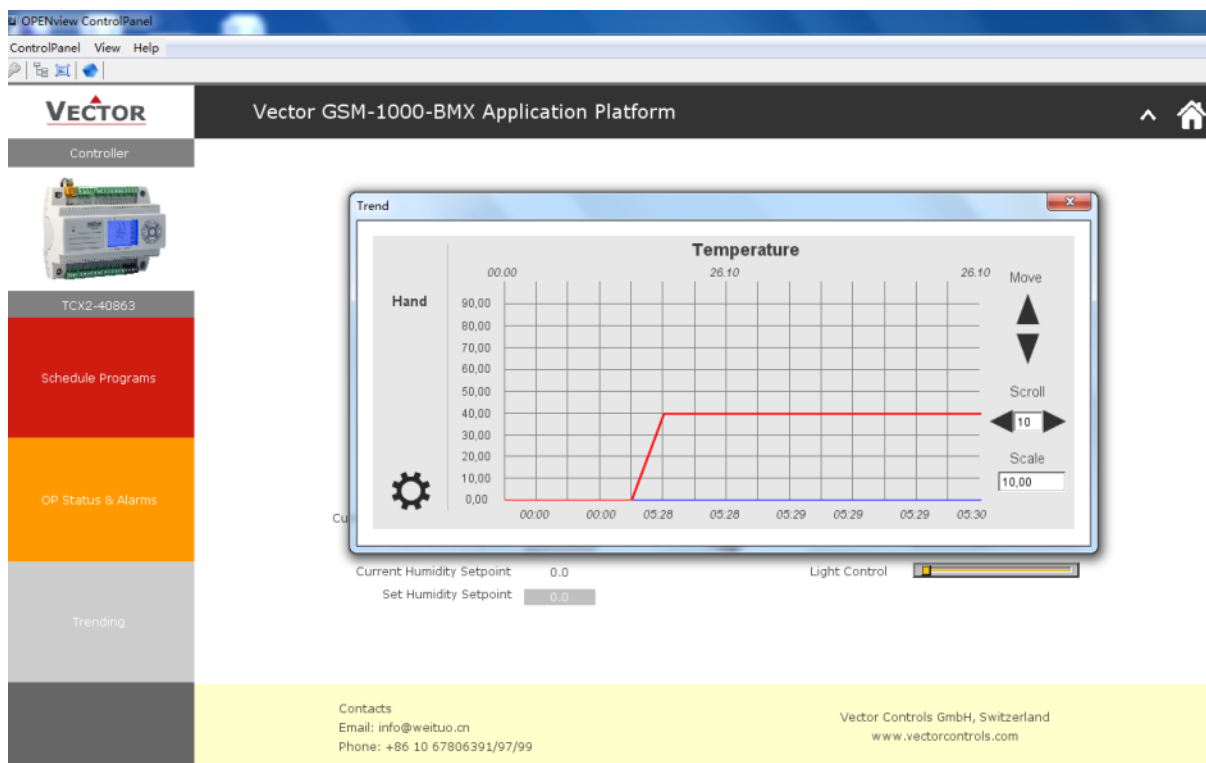
Source graphic in FXL: 4_Trend of dev_temp_x.f$x.

For sampling rate changing please go to file trend128.f$x.

4.3. OPEN Project Builder

OPEN Project Builder is a free pc-side software which helps you create your project conveniently without FXL dongle key.

4.3.1. Installation

You can acquire the installation package "OPENprojectSetup" from us. Then extract all the files out of it and run "setup".

OPEN Project Builder requires Microsoft .net framework 4.51 or later to run. So If you have already installed it the installation duration will be much shorter.
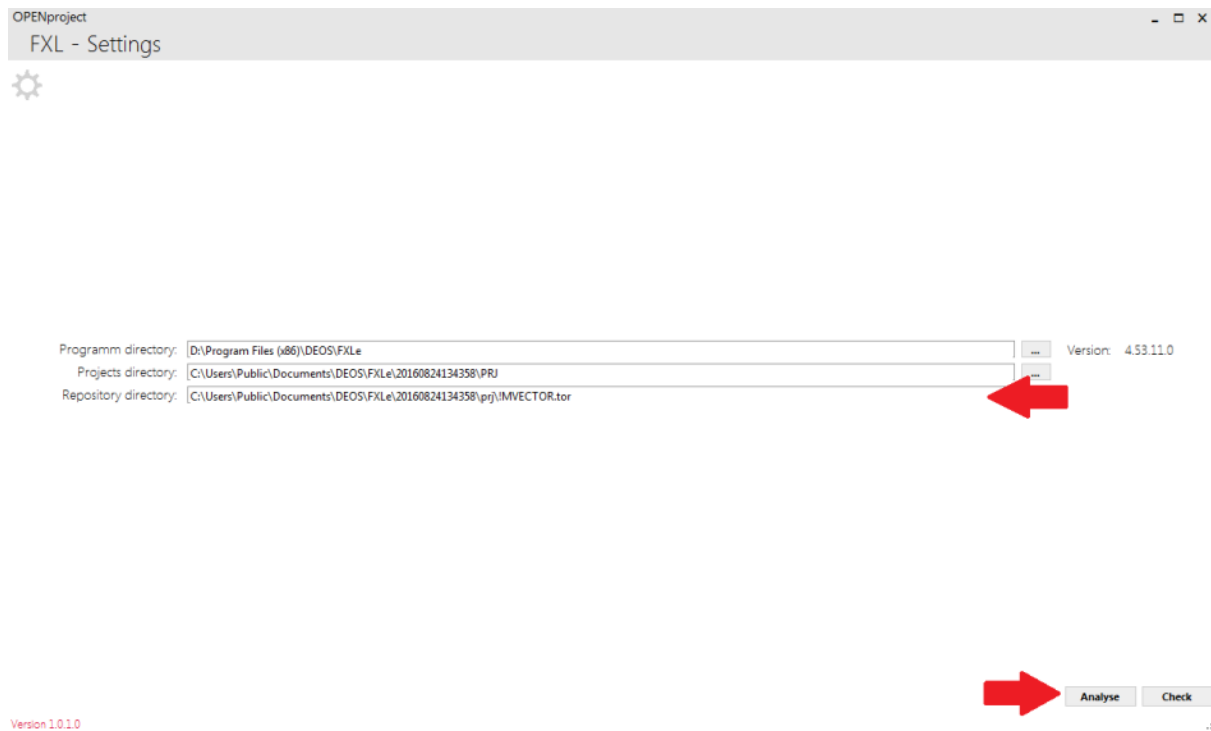
4.3.2. Configuration

After installation, by running it, you will need to set the correct path of FXL resources for it to call.

Go to the path defining page.
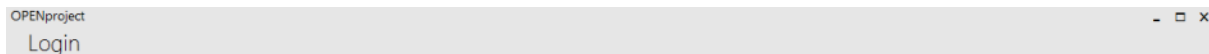
Then change the path. You can click the "analyse" to let it detect the path automatically. Afterwards, please choose the Vector template file in the third blanket (you need to import it to FXL first to see it) and click check to confirm.
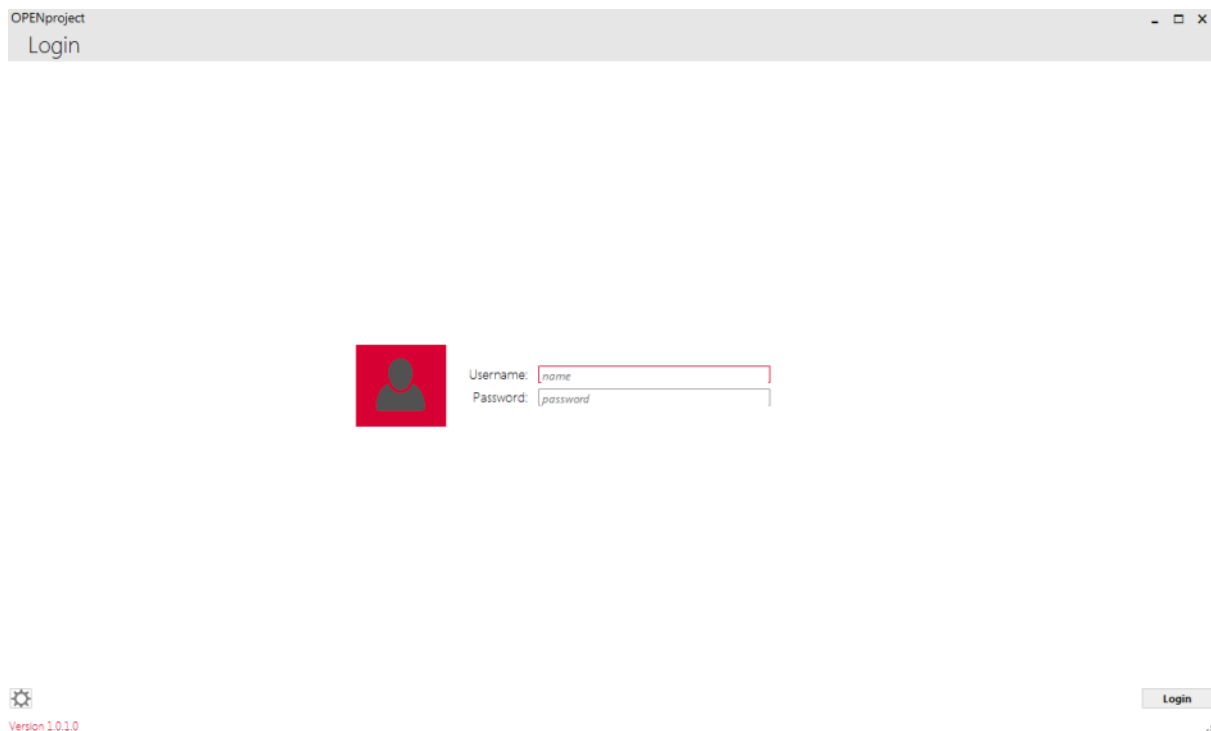
4.3.3. Create a project using OPEN Project Builder

Login without user name and password
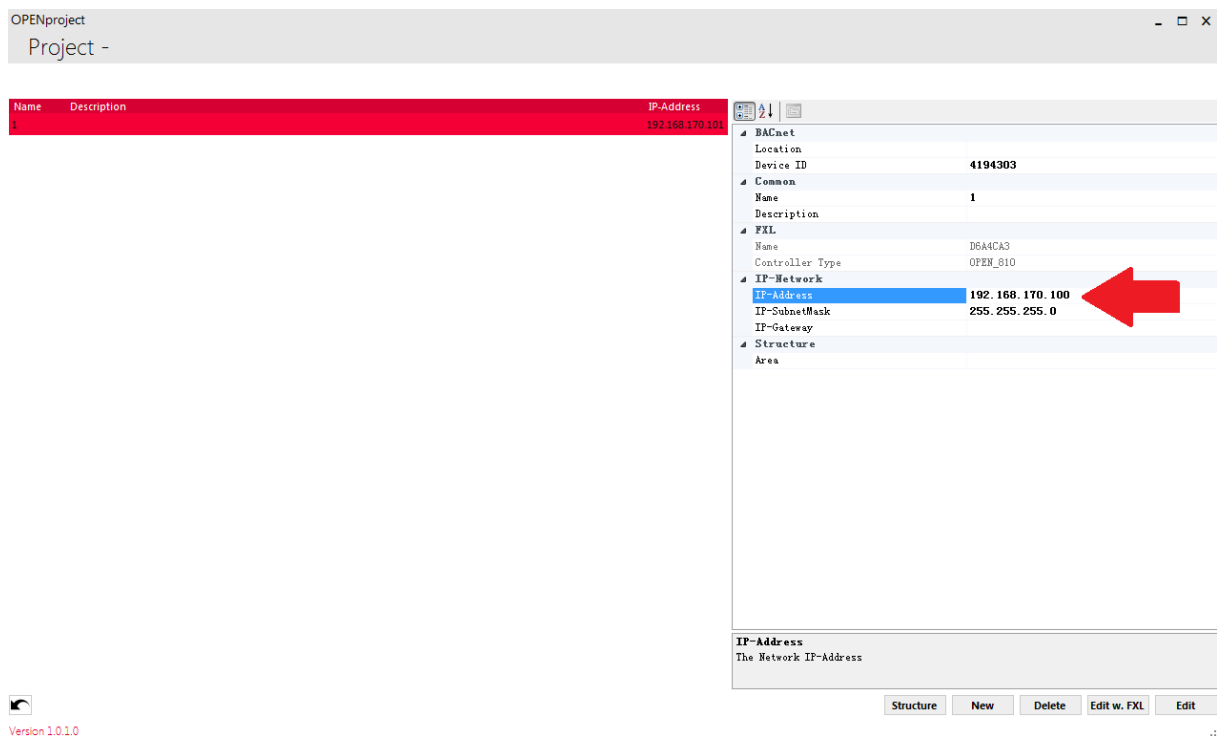
Create a new project

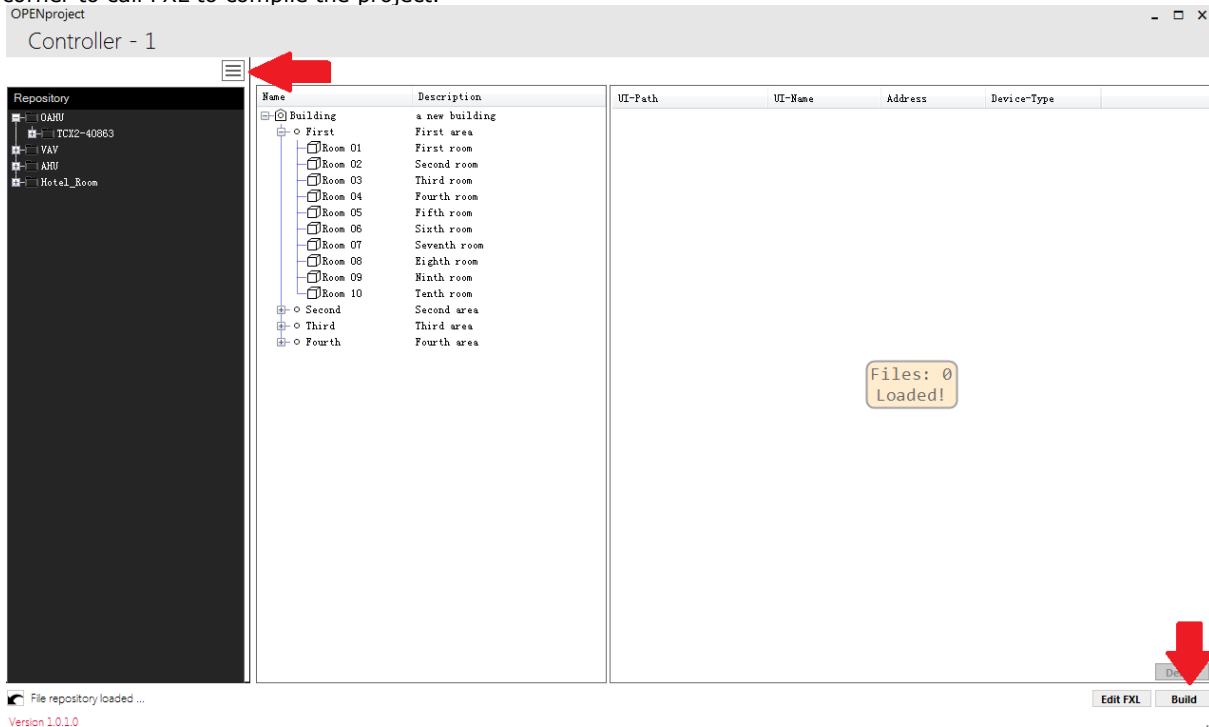Use "edit" to add target GSM controller and edit its hierarchy.
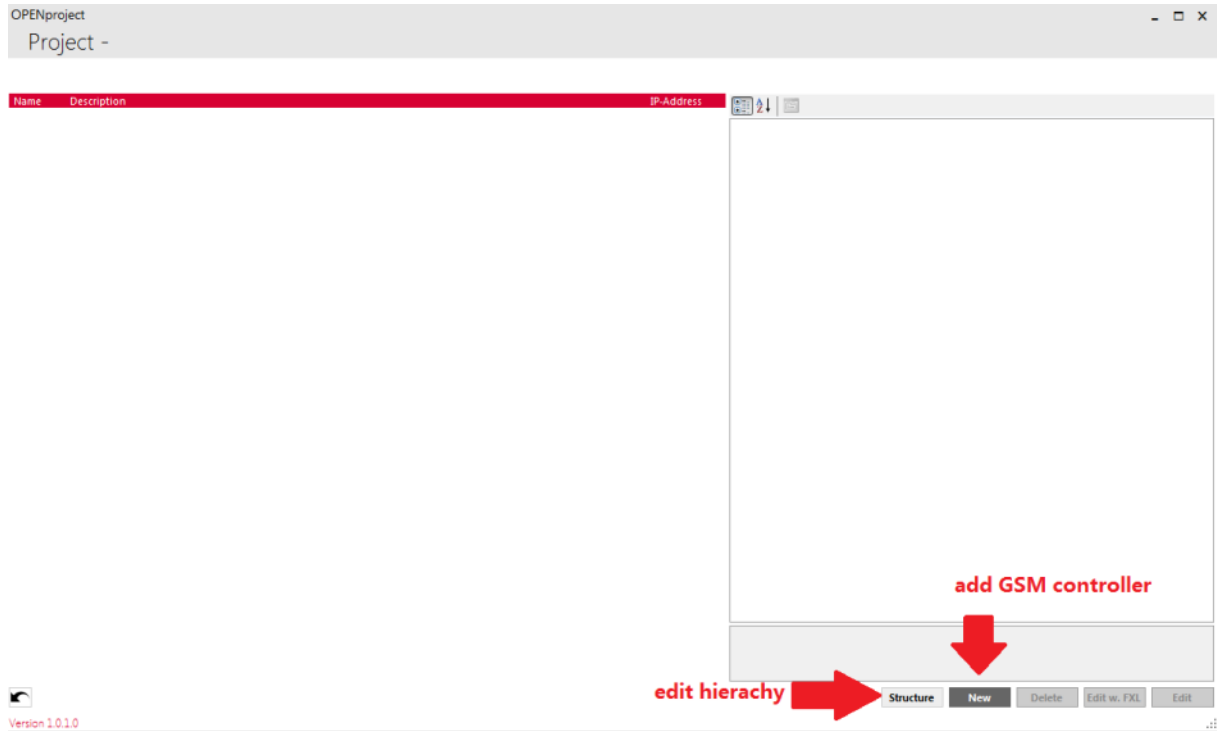
Your GSM controller's IP needs to be filled in correctly. The click "edit" at the bottom right corner to enter the application assignment page.

Expand your application templates and drag & drop them to the project tree. At the end click "build" at the bottom right corner to call FXL to compile the project.

You will get an executable file if the compiling ends without errors. Now you can run it to deploy the compiled project to target GSM controller. Remember to fill in the target GSM IP correctly.