Lesson 01

Lesson 01 will take you through the basics of opening a new program in FREE Studio, assigning some parameters, writing a small application that switches an output depending on a temperature Input.

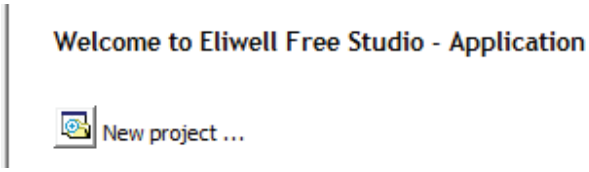
The Digital Output will Switch On, when temperature exceeds Setpoint + Differential and will switch off again, when Temperature falls below Setpoint.

Contents

Creating a new program

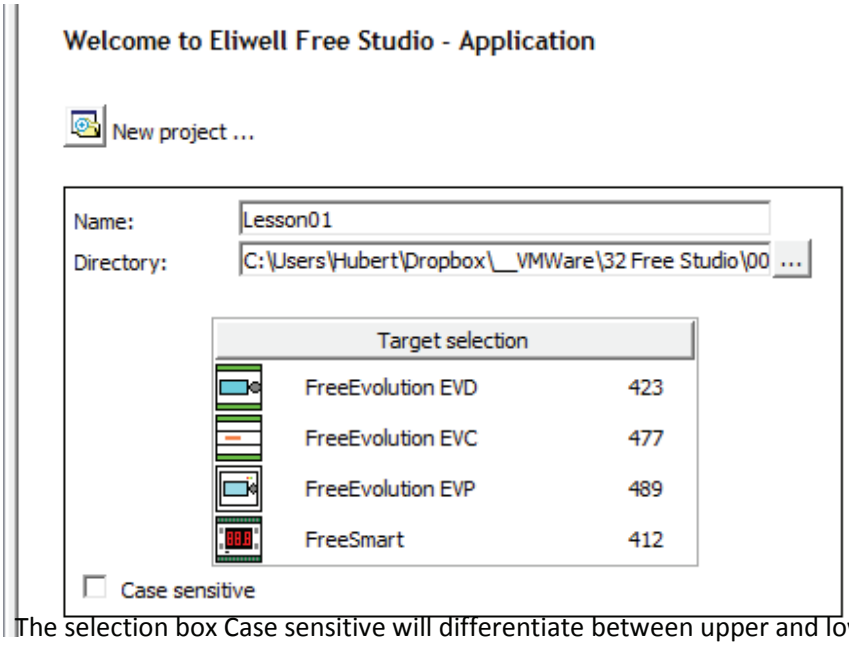
Start Free Studio Application program by clicking on

Click on New project ….



Type a Name for the project and select Directory where to save it.

Click on the selected PLC type (FreeSmart) to save new project



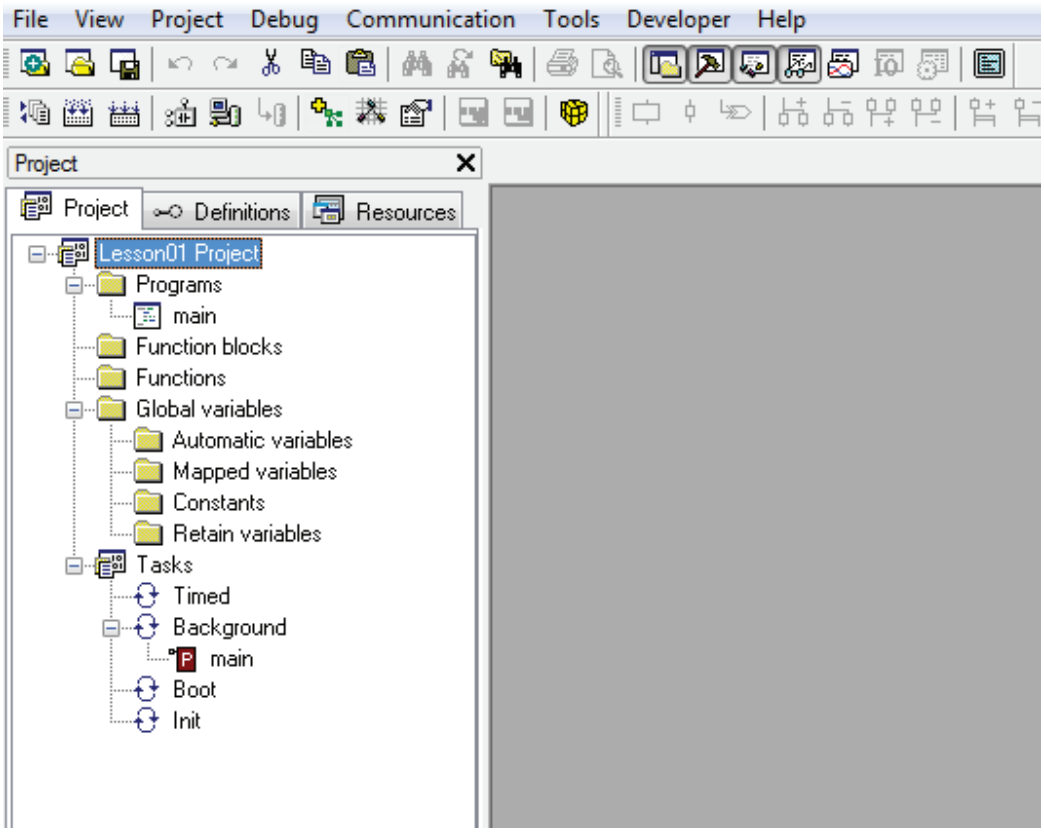
The selection box Case sensitive will differentiate between upper and lower case letters.
E.g. when ticked ‘THEN’ is not the same as ‘then’ and won’t be recognized by the system.

There will always be a default program ‘main’ created. It can but does not have to be used in the project. The program is assigned to ***Background***, which means that the program will be executed in the sequence of assigned programs whenever there is time to do so.

All programs assigned to ***Timed*** will be executed in a 100ms Intervall.

Init – Programs will be executed only once at initial Start-up.

Boot – Programs will be executed once after every re-Boot.



Assigning Resources

Definitions

In the Resources Tab following settings are to be made.

EEPROM Parameters – Create Parameters that need to keep value even when power is switched off. In general , these are parameters that are used in Menus – operational parameters

Status variables – Status variables are created with a modbus address and therefore can be used for communication with other systems, for example SCADA systems, Modbus Masters, etc.

Enums – are used to set a selection of values that can be assigned to a declared variable. For example 0= Off, 1= StandBy, 2= On ➔ Only those three values can be selected if the variable is used in a menu

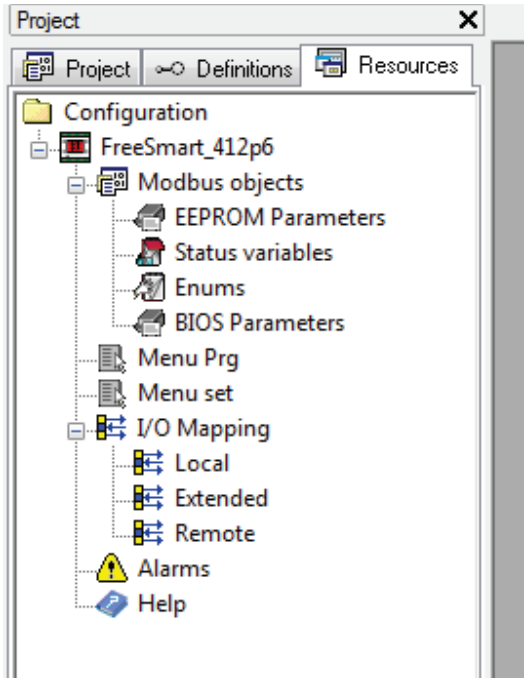
BIOS Parameters – BIOS Parameters are all settings that are available in every PLC. E.g Modbus Address, Analog Input configuration,….

Menu Prg – Here are menus assigned that are accessible via Program Menu (Push right two buttons on SMART) – in general here are parameters assigned that don't need to be accessed frequently.

Menu set – Here are menus assigned that are accessible via Set Menu (Push bottom right button on SMART) – in general here are Setpoints assigned that need to be changed regularly

I/O Mapping:
Local – assign names for all local (On the PLC main unit) Inputs and outputs so they can be addressed in the program
Extended – assign names for the Inputs and Outputs on an extension unit (only used when extension unit is present)
Remote – assign names for Inputs on Remote keypad

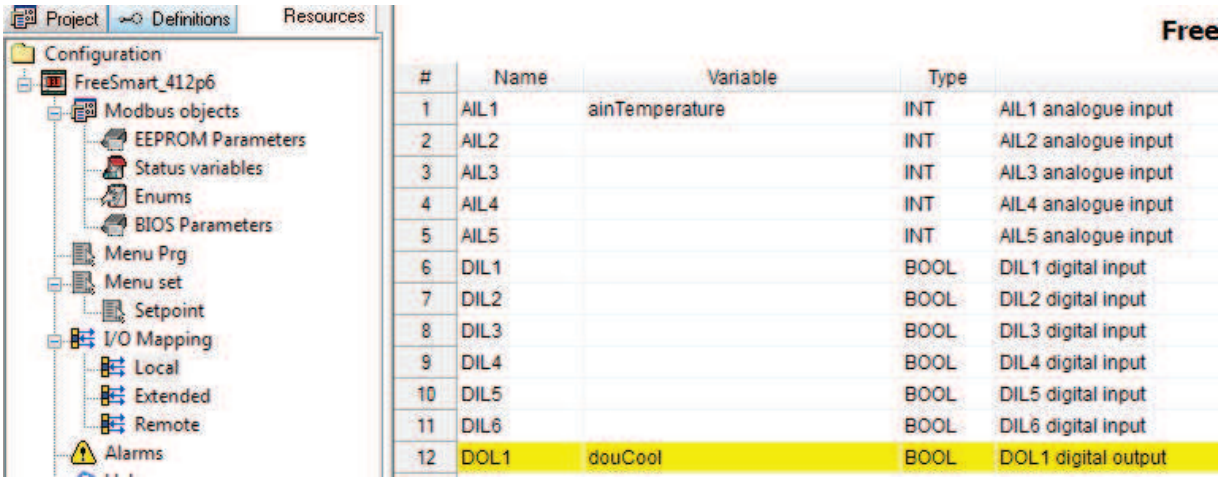
Alarms – assign names for Alarms. Variables defined as Alarms will automatically turn on Alarm indicator whenever they are not 0!!



Resources for Example

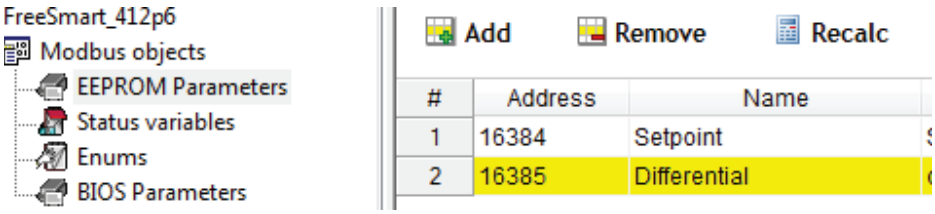
I/O Mapping

For our first example we only use one Analog Input AIL1 and one Digital Output DOL1. The Name can be freely assigned (must start with a letter and no spaces allowed).

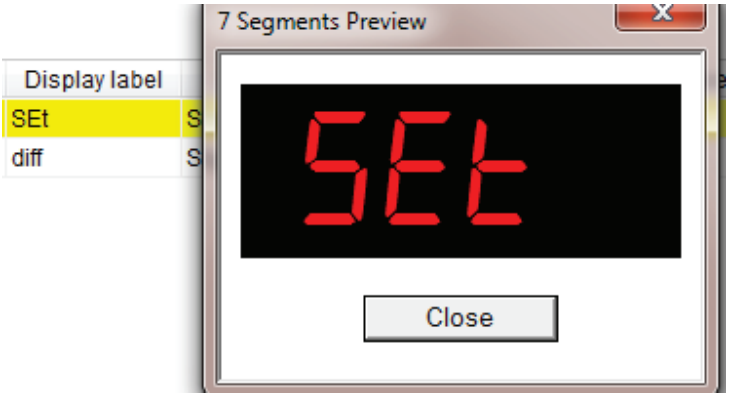


EEPROM Parameters

For our example we define a Setpoint and a Differential to be changed via menus.



Display labels indicate how the parameter is displayed on the smart 4-seg display



Application type – indicates the type of variable that is used.
E.g. INT (-32768 to 32768), Bool (TRUE or FALSE)
Default Value – can be downloaded with the program
Min – When parameter is changed via menu the value won't be lower than minimum value
Max – When parameter is changed via menu the value won't be higher than maximum value

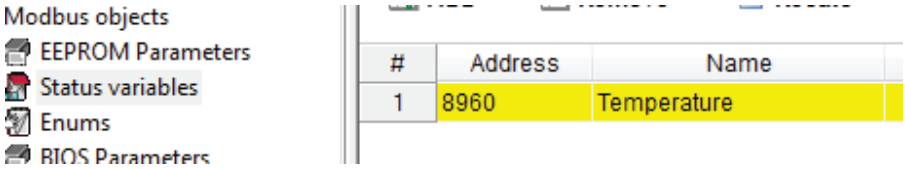
Application type	Default value	Min	Max
INT	0	-200	500
INT	0	0	50

Unit – If one of the available units selected (C, bar) that unit will displayed with Value on the screen (only limited to available units on 4-seg display).
Format – Select if decimal point should be displayed. In our case a Setpoint of 500 represents 50.0 degrees, so all temperatures will be represented in the program with 10* their value.
Access Level – Password protection for certain parameters can be set here

Unit	Format	AccessLevel
°C	XXX.Y	Always visible
°C	XXX.Y	Always visible

Status Variables

For our example we will only use one Status Variable. The Analog input value we will transfer to this status variable so we can assign a Unit and Format for display purposes.

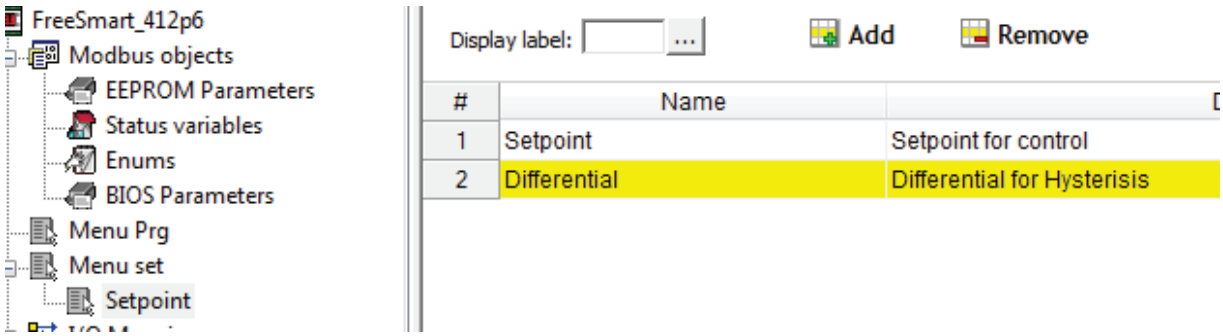


Assign Unit and Format (no other settings necessary)

Unit	Format
°C	XXX.Y

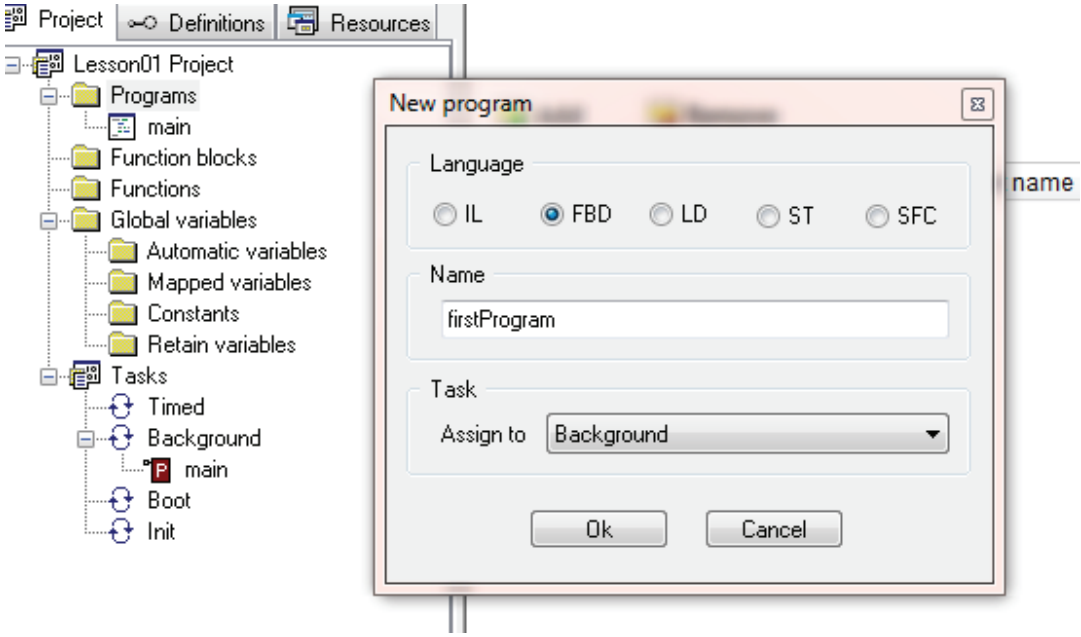
Menu

We will only use one Set-Menu for our example. Right click On Menu Set, add a menu and Name it. Click on ADD and select Setpoint and Differential to be members of this Menu.



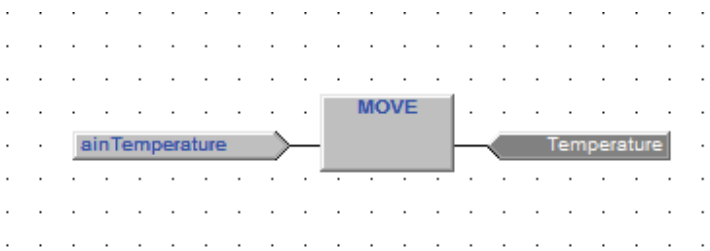
First Program

Right Click on Programs, select FBD (Function Block Program), enter a Name and assign to Background Task.

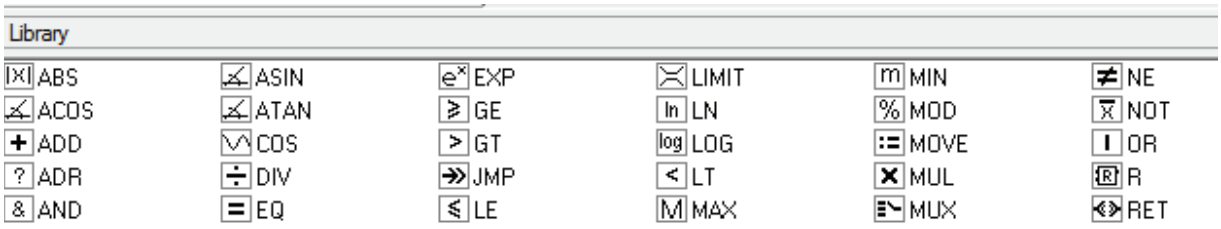


Assign Analog Input to Status Variable

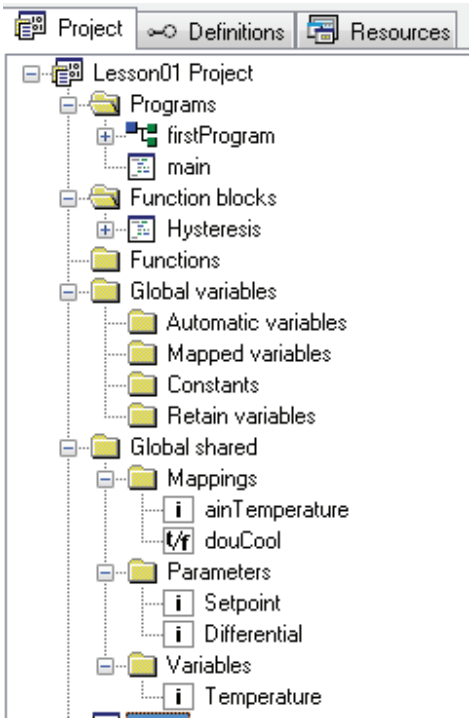
In a first Network we just assign the Analog Input ainTemperature to our Status variable Temperature.



First click and drag the Move Block from the library into the Network.



If the Library is not visible then click on View > Tool Windows > Library.



Under the Project Tab open Global shared and Mappings. Click and drag ainTemperature into the Network and select Input when asked. Connect the block then to the left side of the Move Block.

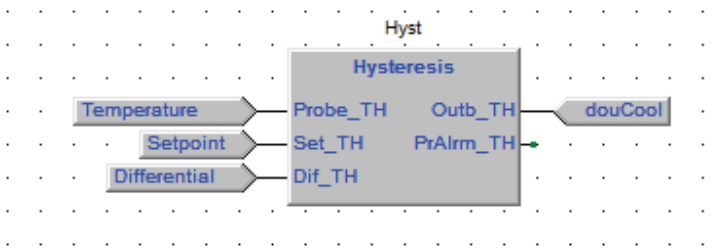
Open Variables and click and drag Temperature into the Network. Select Output when asked. Connect the block to the right side of the Move Block.

Now every time this program is executed the Analog Input value of AIN will be transferred into Status Variable Temperature.

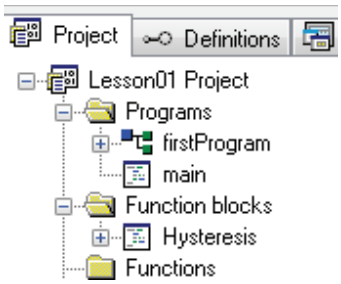
Use Hysteresis Block

We will now use a custom made Function block which contains the logic for switching the output on and off according to Temperature, Setpoint and Differential. The handling and Import/Export of custom Function blocks, creating of custom Libraries will be explained in another Lesson. Function block Hysteresis will be used in this example without detailed analysis of the coding in the FB.

Insert a new Network by clicking on 



First click and drag the Hysteresis function from the function blocks tree into the Network and name it.



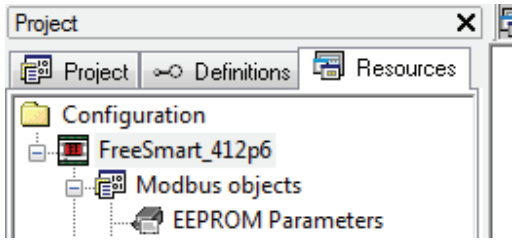
To the left of the Hysteresis block connect Variable Temperature and Parameters Setpoint and Differential as previously explained as Inputs.

Connect the Mapping douCool to the right side of the Hysteresis block as Output.

Set Default display

Once the program will be downloaded the PLC will display 'PLC' while it is running. In order to display a default value, in our case Temperature, follow these steps.

Select Resources and click on the first line in the tree view below Configuration. In our case it is 'FreeSmart_412p6'.



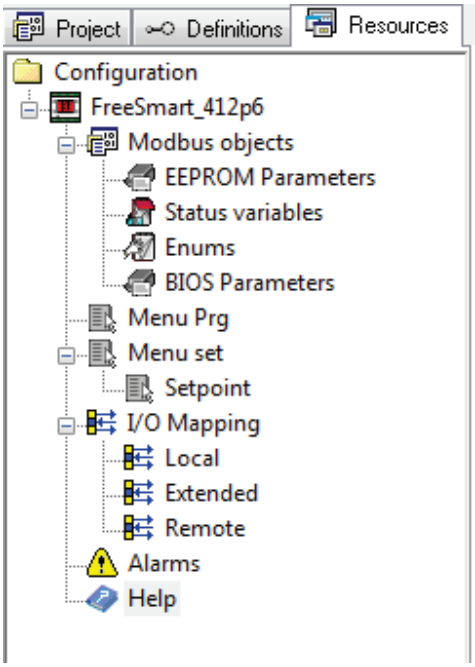
In the main screen ('*FreeSmart Configuration'*) click on the arrow next to *Fundamental state display*: and select Temperature from the drop down list.

Cooling Indicator

There are several indicators available on the SMART display that can be utilized – switched on, off, flashing.

In order to change their display you have to use the Target variable ***sysLocalLeds*** and set it to 0 (OFF), 1 (ON) or 2 (Flashing).

To find out the instance of the LEDs click Help in tree view on Resources tab.



The LED number we will be using is No. 6 for Cooling. The LED should turn on, when the Cooling Output in the program is turned on.

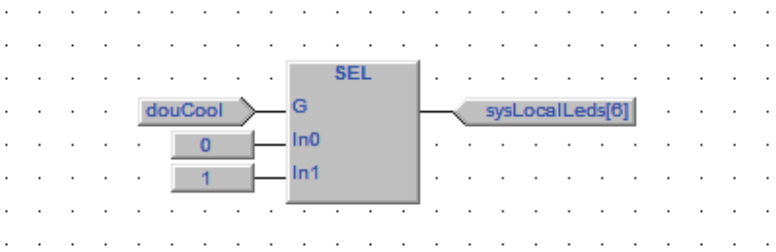
LED reference for the developer

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

LED number	Symbol or icon	Description	Off	On (steady)	On (blinking)
0	:	Colon	SYSLED[0]=0	SYSLED[0]=1	SYSLED[0]=2
1	%R.H.	%RH	SYSLED[1]=0	SYSLED[1]=1	SYSLED[1]=2
2		Defrost	SYSLED[2]=0	SYSLED[2]=1	SYSLED[2]=2
3	Bar	Bar	SYSLED[3]=0	SYSLED[3]=1	SYSLED[3]=2
4		Stand-by	SYSLED[4]=0	SYSLED[4]=1	SYSLED[4]=2
5	°C	°C	SYSLED[5]=0	SYSLED[5]=1	SYSLED[5]=2
6		Cooling	SYSLED[6]=0	SYSLED[6]=1	SYSLED[6]=2

This means that sysLocalLeds[6] must be set to 1 When the Output is on and 0 when the Output is Off.

Insert a new Network by clicking on 



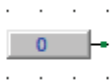
From the library Operator and Standard Blocks click and drag the Selector 

into the Network.

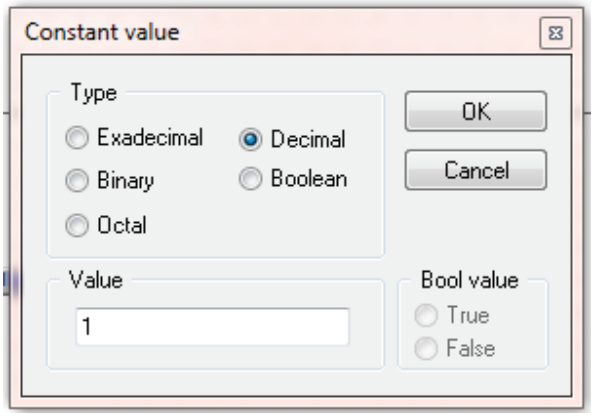
Connect the Output ***douCool*** to the G-Pin of the Selector. The Selector Block will now switch the Value connected to In0 through to the output whenever ***douCool*** is FALSE (=0) and the value connected to In1 whenever ***douCool*** is TRUE (=1).

Now connect a constant with Value 0 to IN0 and a constant with Value 1 to In1 and the LED function sysLocalLeds[6] to the Output of the Selector Block.

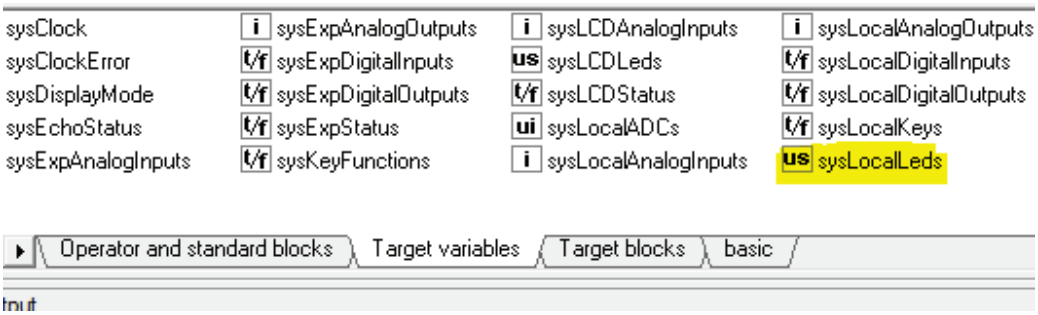
Click on the K-icon and then somewhere in the Network and a constant with value 0 is displayed.



Repeat this step and double click on the 0. A window will be displayed and the constant value needs to be changed to 1. Connect the two constants as described above.



Now select Target variables in the library menu and click and drag sysLocalLeds into the Network



The variable will be displayed without the number for the relevant LED

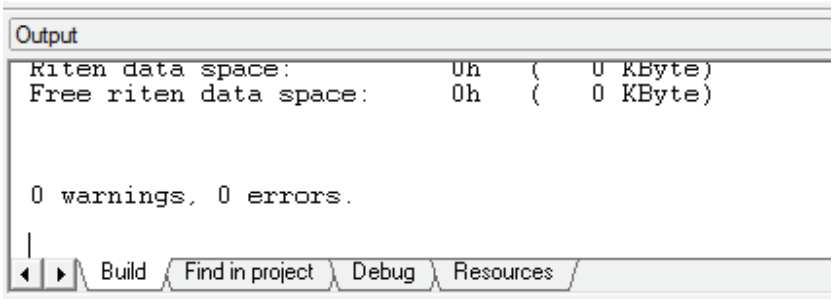
Double click on ***sysLocalLeds*** and change the name to sysLocalLeds[6].

Connect it as an Output to the Selector Block.

Check for errors

In order to check for errors click on *Program Compile* 

If there are no problems (warnings or errors) the output Window will display *0 warnings, 0 errors*



If the Output window is not visible click on *View > Tool Windows > Output.*

Conclusion

This is the first simple program written in ***FreeStudio.***

In the next Lesson we will show how to run a Simulation on the screen and download and monitor the program without connecting to a physical PLC.