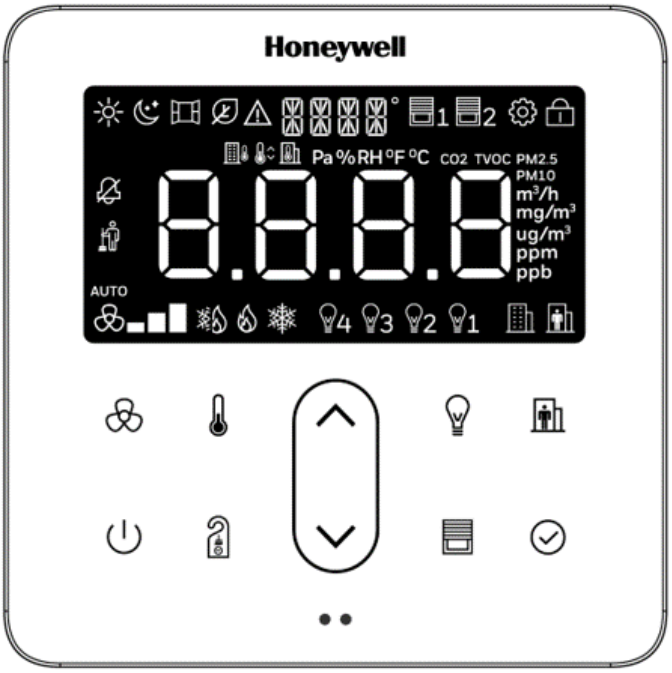
Saia-Burgess Controls AG

Bahnhofstrasse 18, 3280 Murten, Switzerland
T +41 26 580 30 00, F +41 26 580 34 99
www.saia-pcd.com

Short description of PCD7.LRxx-P5 programmable room controller and TR80 wall module example program



Versioning:

Version	Date	Modifications	Owner
V001	2021.08.17	First version	Attila Kovács
V003	2021.09.22	Program has major modifications, so the document is completely rewritten.	Attila Kovács

Table of Contents

General

This document in short describes the “IRM_TR80_Modbus_Example_V3” application program.
It will explain the program structure, the realized main functions and how to enable or disable program parts depending on the necessity. The application example is created to support the System Integrators using the new TR80 wall module in real applications. The aim is to provide a good starting point, which can be customized according to the requirements of the project.

The TR80 is a functions rich and configurable wall module with one slave and one master Modbus RTU communication port, which is providing an interface between the HVAC, light and blind control of the room and the user of the room (guest of the room).

The PCD7.LRxx-P5 is a room controller which is freely programmable by PG5. It can be used to create cross-plant room automation functions like HVAC applications, controlling fan coil devices, radiators, cooled ceilings, and CO2 / air quality.
It has 2x RS-485 interfaces and 1x Sylkbus interface. The first RS-485 communication port can be used to connect the controller to superior controller (Plant controller) and/or to SCADA system. The second RS-485 port and/or the Sylkbus port can be used to connect wall unit(s), remote I/O extension(s) etc.

There is one Blind and one DALI 64 light installation connected to the system to demonstrate the control capabilities of the TR80.

Abbreviations

- TR80: wall module
- IRM: PCD7.LRxx-P5 series of room controller
- RIO: SBC Remote Input / Output module
- PG5: Programming environment to program PCD7.LRxx-P5 or other SBC controllers
- SPM: Saia Project Manager, part of PG5
- Fbox: Function box to be placed in the FUPLA page to create program.
- FUPLA: FUPLA is SBCs own function block diagram editor.
- PB: Program Block
- HVAC: Heating Ventilation and Air Conditioning
- AHU: Air Handling Unit
- Sylkbus: Honeywell proprietary communication bus
- RS-485: physical layer of communication
- Modbus RTU: protocol layer on serial communication line
- SBus: SBC proprietary protocol based on RS-485 or Ethernet physical layer
- SCADA: Supervisory control and data acquisition
- OCC: occupancy
- SI: System Integrator
- EC fan: Electronically commuted fan, 100% speed controllable

Additional documents

- 27-653_ENG_Manual_IRM-PG5.pdf
- Manual_Room_Template_V2_01.pdf
- sbc_DALI64SYLK_application_template_ab_14072020.ppt
- 31-00482M-01 - TR80 Modbus Wall Module - IOG_D1.4.pdf
- 31-00480M-01 - TR80 Modbus Wall Module – Datasheet_English.pdf
- 52-004_ENG_DS_HON-SBC_DALI64-SYLK.pdf
- Ex-Or_DALI64_A4Infosheet_v1.pdf
- **ModbusRegistersV1.1.25.0.xlsx**

Necessary software, hardware, firmware and configurations

Hardware

1. Laptop or personal computer with USB support
2. USB cable
3. Micro USB cable
4. 24 V DC, minimum 1 Ampere power supply
5. Some wires to connect the power supply, controller, wall unit
6. Some communication cables to build up the RS-485 network and the Sylbus network
7. Blind from Griesser with control relays and pushbuttons properly wired to the controller
8. Sylkbus interfaced DALI 64 demo light cube form Ex-Or (DALI64SYLKPSUF)
9. Modbus interfaced DALI64 demo light cube from Ex-Or (DALI64MODPSUF)
10. Room controller PCD7.LRxx-P5
11. Wall unit TR80

Software

1. PG5 2.3.184 to program the PCD7.LRxx-P5 + ESuite V2 library.
2. Backup of the application program

   “Project IRM_TR80_Modbus_Example_V3_23184_20210922_0853.zip”
3. Light Touch applet for Android device to configure DALI 64 devices

Firmware

1. For PCD7.LRxx-P5: **FW 1.10.07 or newer**
2. For TR80: **FW V.1.1.25**

	**The application is created by using the above-mentioned Firmware. SI must take care, to use the proper FW version in the TR80, which has a corresponding Modbus registers mapping excel file! HW_Modbus_V.1.1.25.hex + ModbusRegistersV1.1.25.xlsx**

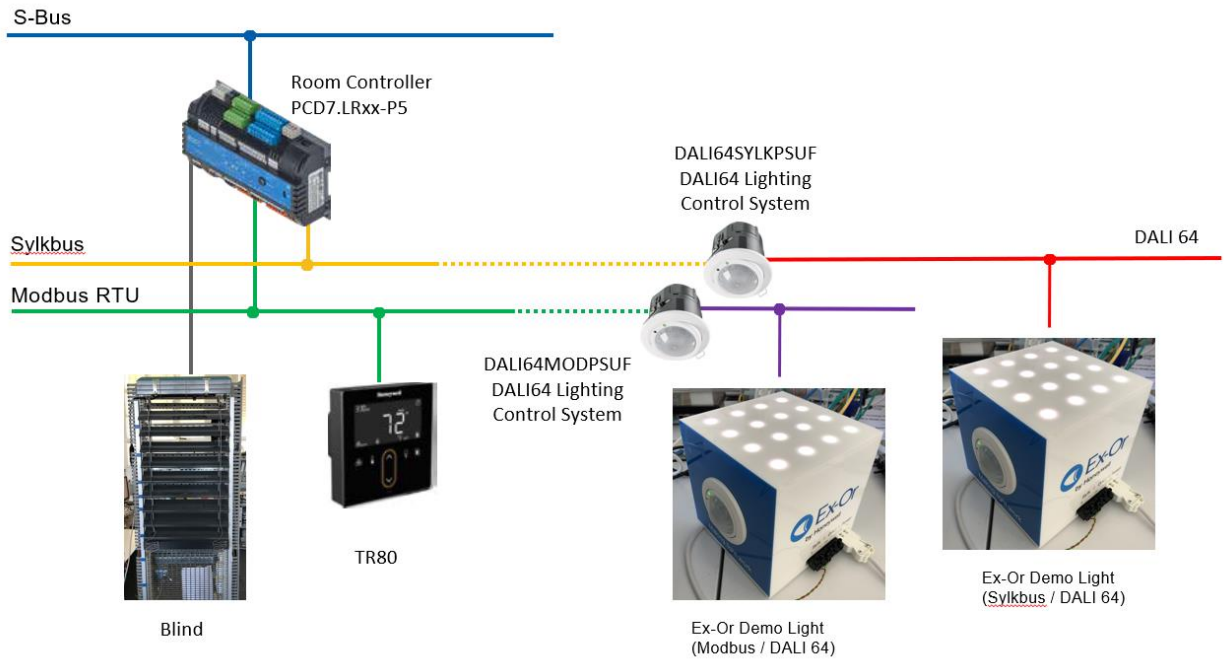
Configuration

The Ex-Or Dali 64 device must be configured so, to provide at least 4 light groups and 4 scenes per groups. To configure it, the Light Touch software needs to be installed to an android device.
The Ex-Or Dali 64 device can be substituted by any other DALI 64 lights installation. Important that the installation uses the DALI64SYLKPSUF as Sylkbus / DALI 64 interface or DALI64MODPSUF as Modbus / DALI 64 interface.

	We are not providing in this document wiring diagram for the electrical connections of the hardware elements. The SI must do proper mains, control, and communication connection on his own. For that, the SI may need to refer to the user manuals of the devices.

System architecture

The example program has Modbus / DALI64 and Sylkbus / DALI64 program part. It is possible to select which communication bus is used.



Communication connections:

- S-Bus: possible connection on serial RS-485 to upper level devices (Plant controller or SCADA)
- Sylkbus: connection to the DALI 64 PIR sensor (selectable)
- Modbus RTU: connection to the DALI 64 PIR sensor (selectable)
- DALI 64: connection to the lights in the cube
- Modbus RTU: connection to the TR80 wall unit

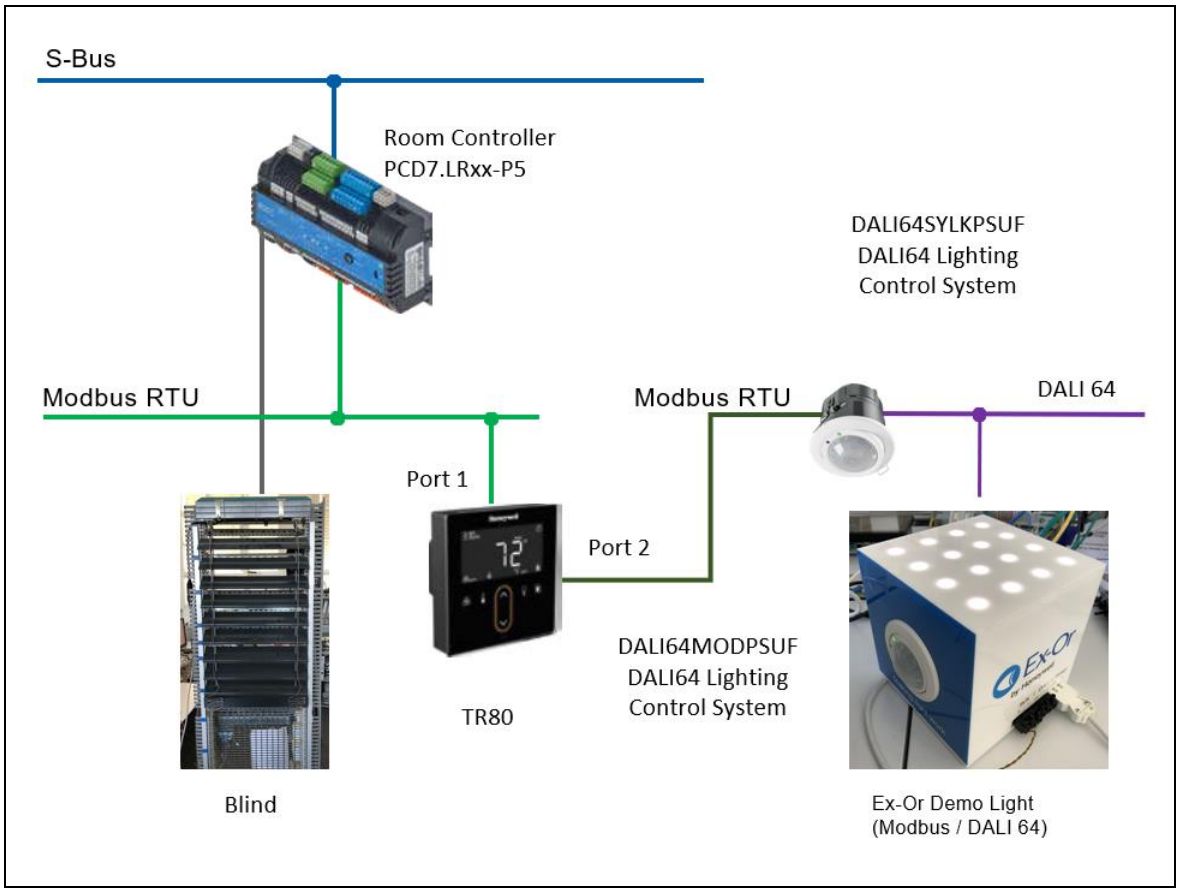
The black line is representing normal relay control of the Blind.

Other possible light control concept

In the previous system architecture, regarding Modbus/DALI64 sensor, the lighting control is done in the room controller. The following control loop is used: the light level request is coming from TR80 over Modbus, the room controller process it, and sends the command to the Modbus/DALI64 sensor over the same Modbus line. Minimum 3 communication cycles are need from the room controller to properly send the command. Depending on the communication speed, and the program execution speed, the response time to a light request can be high.

Instead of the above-mentioned control concept, it is possible to create a control loop, where the TR80 can control the Modbus/DALI64 sensor directly over its second Modbus Master port.
In this way the light request from the TR80 is directed to the Modbus/DALI64 sensor with fast communication. The room controller can get the light level or scene number feedback, because TR80 has a built-in gateway between the Modbus Port 1 and the Modbus Port 2. The Room controller still can send command to the DALI64 device because its command goes through the gateway.

This control concept is not implemented in the example program.



PCD7.LRXX-P5 Input / Output possibilities

The used room controller is **PCD7.LRL2-P5** in this project, but other PCD7.LRXX room controller can also be used. For pure Modbus installation any PCD controller can be used from SBC portfolio.

Item number	Housing	Power supply	Analogue outputs (AO)	Universal inputs (UI)	Relay	Triacs (24/230 VAC)	I/O total	Micro-USB	2× RS-485	All connections with plug	72h real time clock power reserve	24 VAC output for field devices and triac outputs
PCD7.LRL2-P5	Large	230 VAC	2	6	4	4	16	×	×	×	×	max. 300 mA

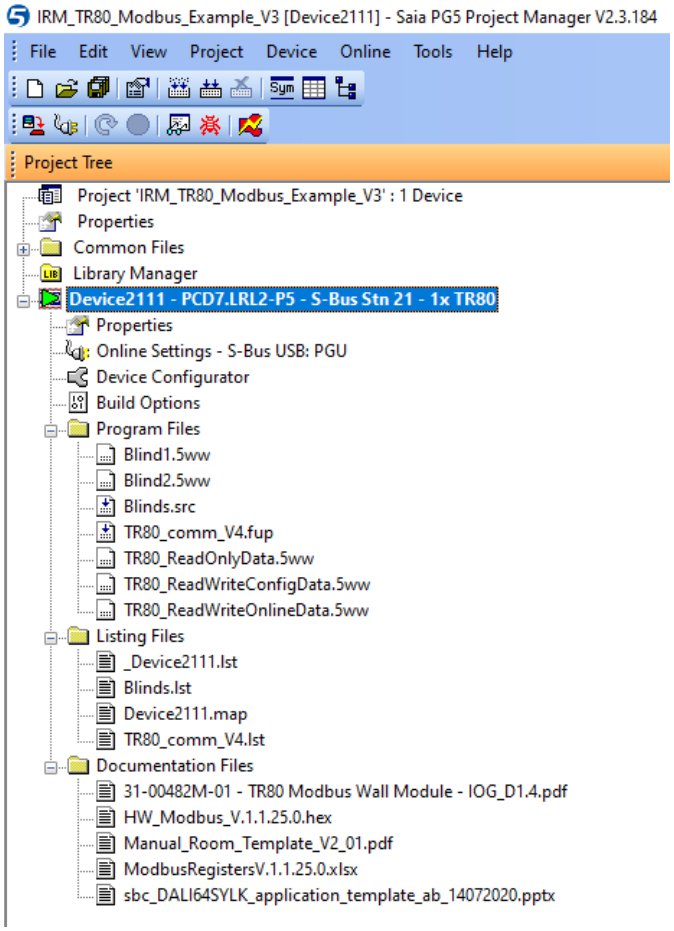
	The application is created for PCD7.LRL2-P5 without considering the Input / Output limitation of the real hardware. It means that HVAC, Lights and Blinds control might use an IO distribution which cannot be solved with built in IOs, so SI must consider using RIO (e.g. E-Line RIO).

Application program

The application program is developed in PG5 V2.3.184 programming environment. The FUPLA and Instruction List programming languages are used.

Restore program

The provided application backup “Project IRM_TR80_Modbus_Example_V3_23184_20210922_0853.zip” must be restored in PG5.



	It is possible to use different type of PCD7.LRxx-P5 device as the memory resources are the same in all type of programmable room controller

Structure of the program in the SPM

The project has standard lookout in the SPM.

Program files

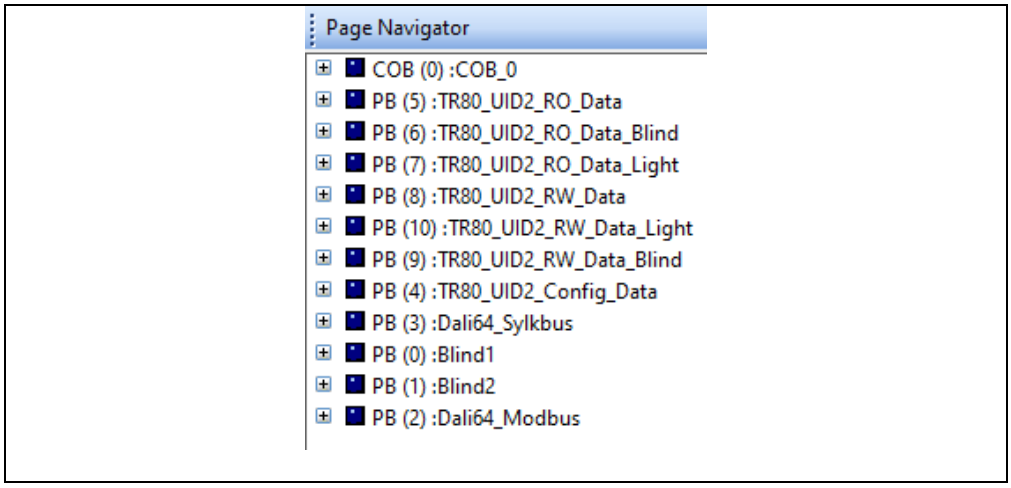
Blind1.5ww	Watch window file for Blinds group 1 control.	
Blind2.5ww	Watch window file for Blinds group 2 control.	
Blinds.src	Program Blocks for controlling Blind 1 and Blind 2	executed code
TR80_comm_V4.fup	Main program to control the communication between the IRM and TR80, Room HVAC application, control lights over Sylkbus / DALI64 communication and control the Blind.	executed code
TR80_ReadOnlyData.5ww	Watch window file for Read Only Data from TR 80.	
TR80_ReadWriteConfigData.5ww	Watch window file for Configuration Data (R/W) of TR 80.	
TR80_ReadWriteOnlineData.5ww	Watch window file for Online Read/Write Data of TR80.	

Documentation files

Some manuals, firmware and Modbus register mapping file are provided here.

Details of TR80_comm_V4.fup

This program is the core of the application, which has the following parts:

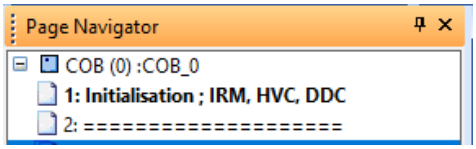


COB 0: Main program	Main program which calls the program blocks
PB 5: TR80_UID2_RO_Data	Reading Read Only data of TR80
PB 7: TR80_UID2_RO_Data_Light	Reading Light related Read Only data of TR80
PB 6: TR80_UID2_RO_Data_Blind	Reading Blind related Read Only data of TR80
PB 8: TR80_UID2_RW_Data	Reading and writing Read/Write data of TR80
PB 10: TR80_UID2_RW_Data_Light	Reading and writing Light related Read/Write data of TR80
PB 9:TR80_UID2_RW_Data_Blind	Reading and writing Blind related Read/Write data of TR80
PB 4: TR80_UID2_Config_Data	Reading and writing Configuration data of TR80
PB 3: Dali64_Sylkbus	Communicating to DALI64 lights over Sylkbus
PB 0: Blind1	Signal Handover for Blind 1 and calling the PB79 from “Blinds.src”
PB 1: Blind2	Signal Handover for Blind 2 and calling the PB78 from “Blinds.src”
PB 2: Dali64_Modbus	Communicating to DALI64 lights over Modbus

	When the device is programmed with FUPLA, it automatically creates a COB in the background. Thus, only 1 COB is available to the programmer.

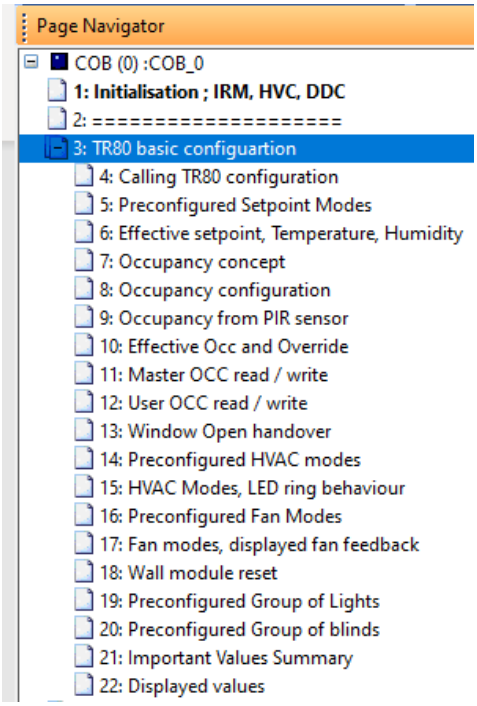
Initialisation

The code of the Initialisation page was taken over from the Room template for PG5 programmable room controller PCD7.LRxx-P5. Please check the "**Manual_Room_Template_V2_01.pdf**" for details. There is 2 seconds start delay which enables the communication to TR80. There are additional conditions later to decide, which part of the communication should start.



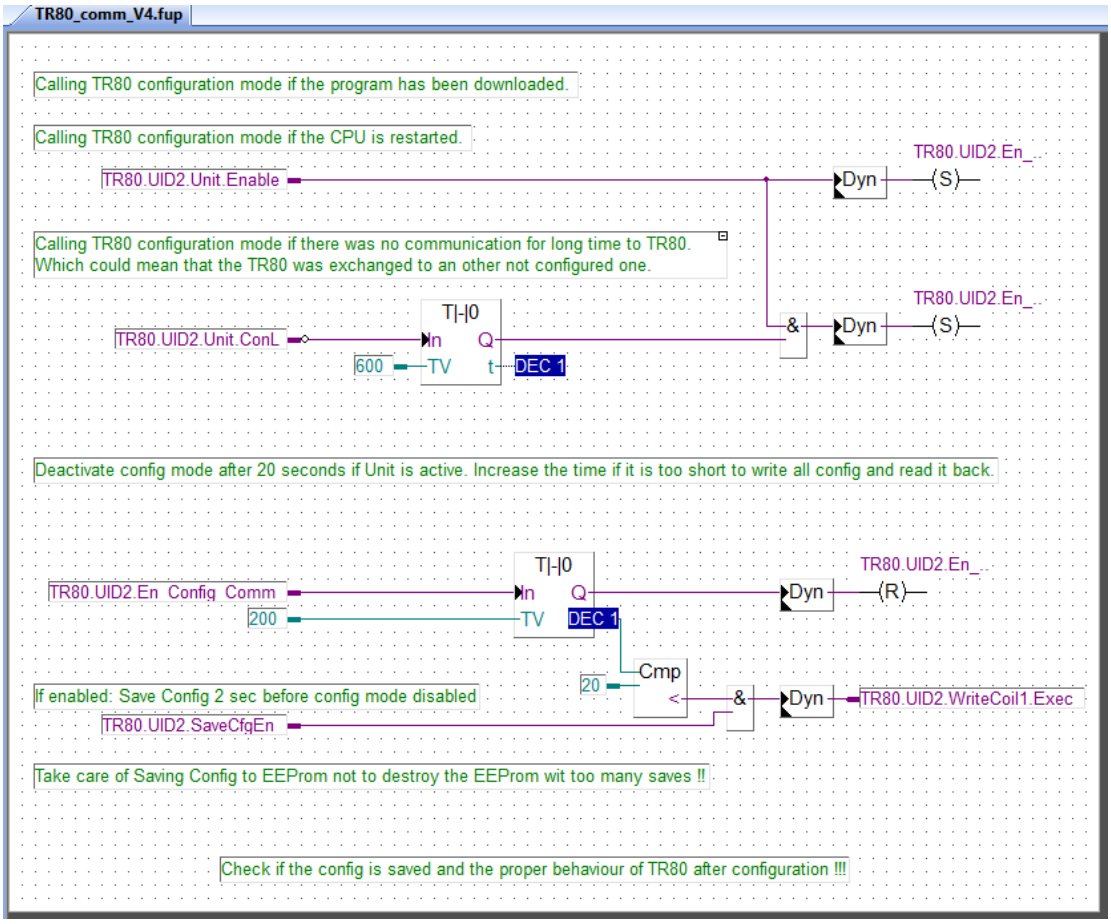
TR80 basic configuration

This part of the program is responsible to configure TR80 to match its behaviour of the application program requirements. And it is a collection of handover pages where the commands and feedbacks of TR80 are transferred.



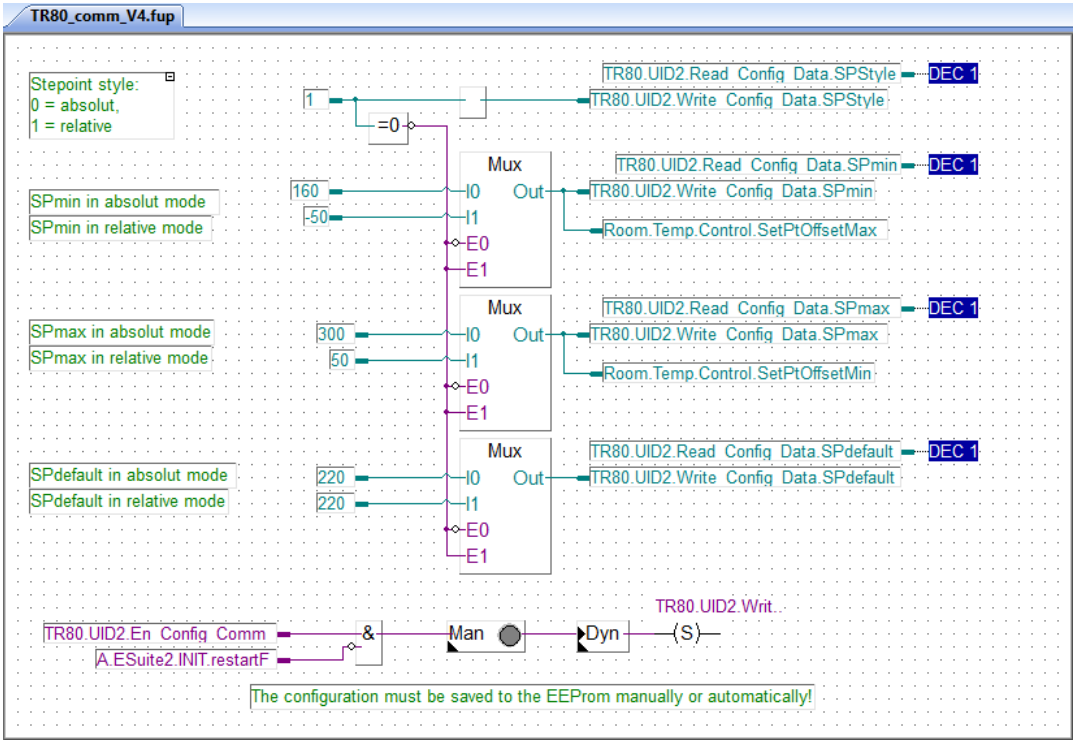
Calling TR80 configuration

- Two seconds after start-up of the controller the communication is enabled to TR80. In the first 20 seconds only the configuration part is enabled.
- Configuration mode is enabled also, if there was no communication to TR80 at least for a minute, but the communication works again, it is supposed that the TR80 has been exchanged, and the new device needs configuration.
- After 20 seconds the configuration mode is disabled.
- Before the configuration mode ends up, the program sends a Save Config command to the TR80 if the function is enabled. Take care of saving too many times to the EEPROM.



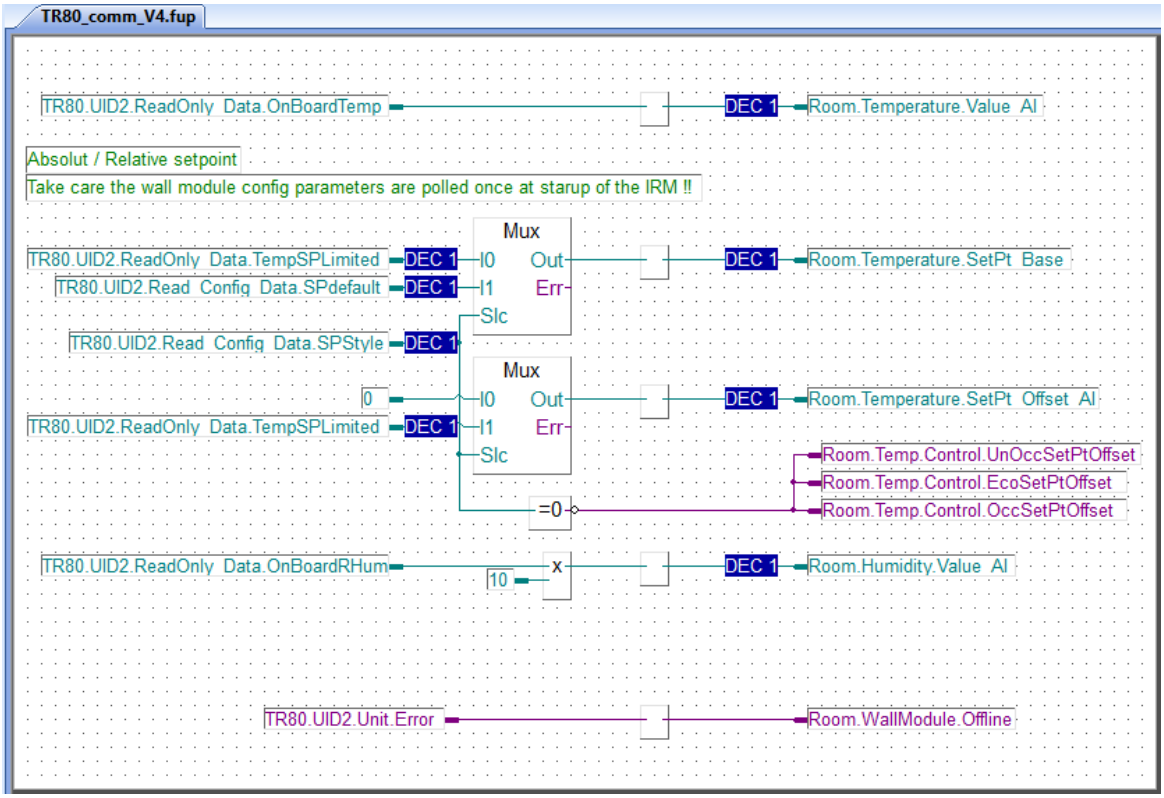
Preconfigured Setpoint modes

It is possible to change on TR80 between relative and absolute setpoint. The setpoint limits are set accordingly. The setpoint limits and default setpoint are also transferred to SetPoint Fbox of the Room Application.



Effective Setpoint, Temperature, Humidity

This page is used to hand over the temperature setpoint (absolute, relative) the room temperature and the humidity to the Room Application.



Occupancy concept

This is a textual page only, which tries to explain the occupancy concept.

The Occupancy source could be from Modbus & Override from button.

Occupancy can come from:

1. External Scheduler (this can be SCADA or Plant controller). This occupancy will be the master occupancy in the Room application and in the TR80.

   Possible modes:

External Scheduler	TR80
OFF	off
(Protect)*	holiday
ECO	Standby
UNOCCUPIED	unoccupied / away
OCCUPIED	occupied / comfort
Occupied	bypass**
	unknown / auto
	not used

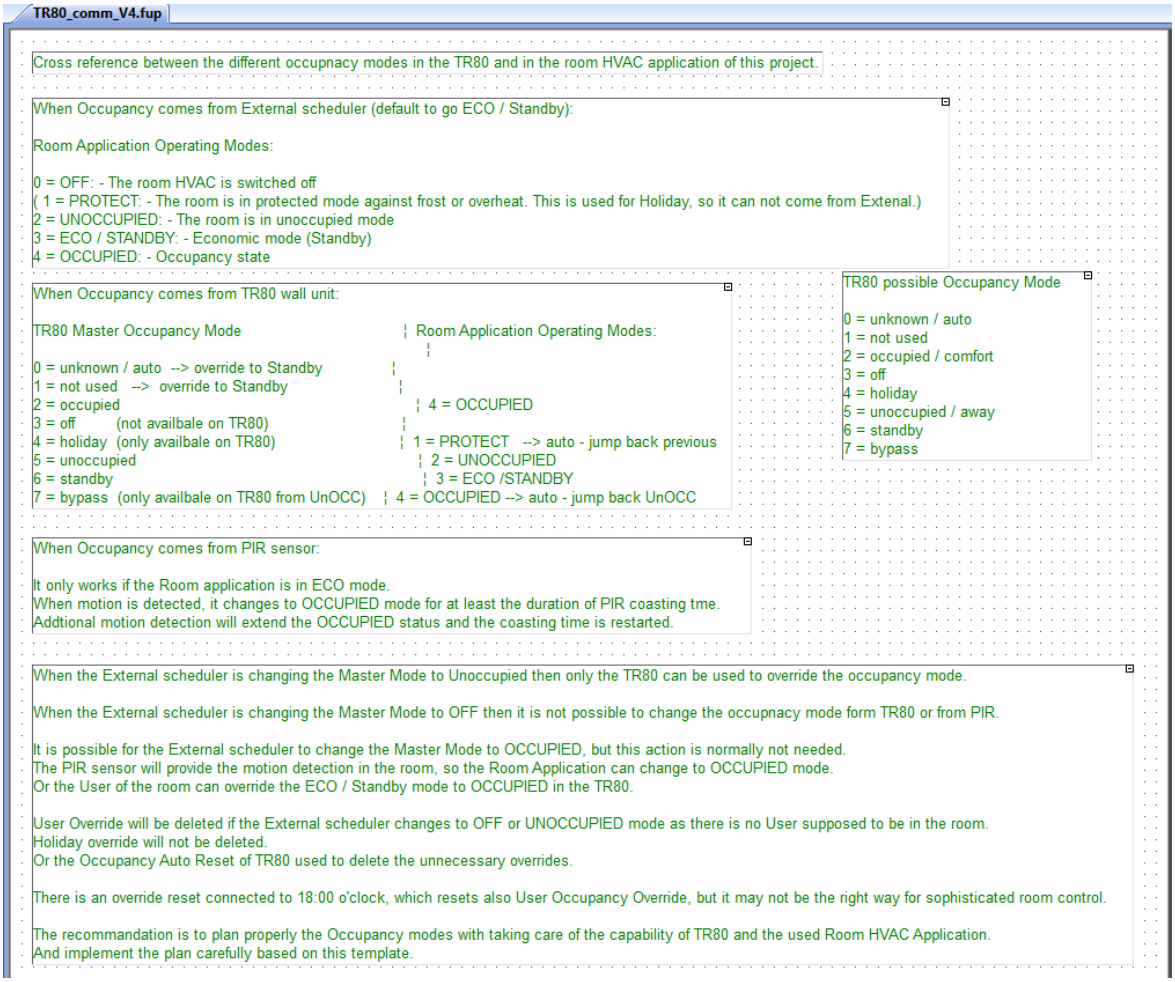
*Protect mode is removed from master modes to substitute with Holiday mode in TR80. The Holiday mode is a user override, so it cannot be initiated from the Master.
** Bypass mode is initiated on the TR80 by the user, and it overrides the UNOCCUPIED mode (Master) to Occupied mode for the user selected time duration.

2. External PIR sensor is giving motion detection signal which is then used to create an Occupancy. **This PIR sensor signal is working if the Master occupancy mode is ECO**. Then the signal from PIR sensor is changing the Occupancy mode to OCCUPIED till the motion (+ off delay called coasting time) is detected in the room. If the Master mode changes to other modes than ECO, the PIR sensor signal is blocked.

3. TR80: The Wall Unit can display the actual Occupancy mode with Occ icon and text description. The user can override the current occupancy mode by pressing the Occ button on the unit and select the required mode. There are modes which are blocked completely or modes which can only come from a specific other mode. These are called “Allowed overrides”.

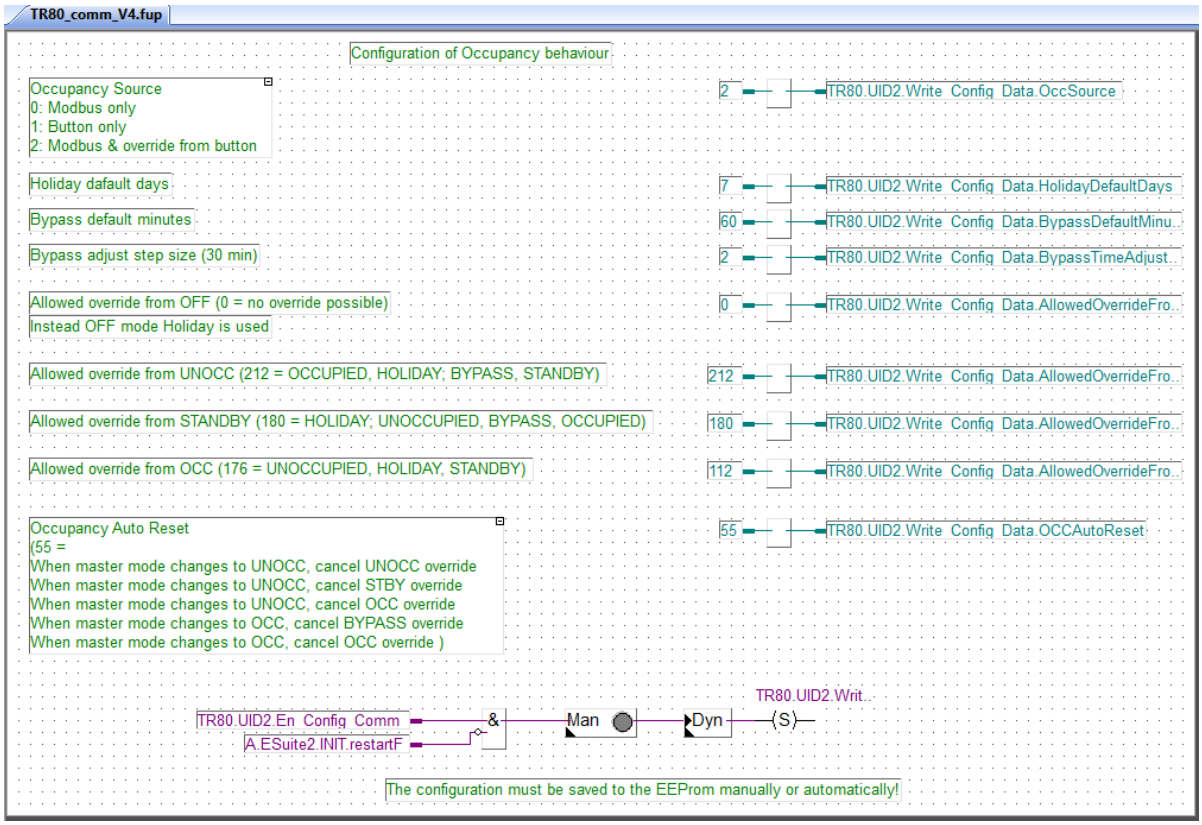
TR80 mode	User Possibility
Off mode	This can only be selected by the Master
From unoccupied	Occupied, holiday, bypass, standby
From standby	Occupied, holiday, bypass, unoccupied
From occupied	standby, holiday, unoccupied

   The overrides will be automatically deleted, if the Master modes is changing to the same mode as the override is, e.g. User override is unoccupied, when master changes to unoccupied, then the user override is cancelled.
   Or the room application program will delete the user override except Holiday, e.g. when Master mode changes to unoccupied (at 18:00 o’clock for office room), it deletes every user override except holiday. So, if user works late, he must go to the TR80 and initiate a bypass mode to get the Occupancy mode of the Room Application.


TR80_comm_V4.fup
Cross reference between the different occupnacy modes in the TR80 and in the room HVAC application of this project.
When Occupancy comes from External scheduler (default to go ECO / Standby):
Room Application Operating Modes:
0 = OFF: - The room HVAC is switched off
(1 = PROTECT: - The room is in protected mode against frost or overheat. This is used for Holiday, so it can not come from Extenal.)
2 = UNOCCUPIED: - The room is in unoccupied mode
3 = ECO / STANDBY: - Economic mode (Standby)
4 = OCCUPIED: - Occupancy state
When Occupancy comes from TR80 wall unit:
TR80 Master Occupancy Mode
Room Application Operating Modes:
0 = unknown / auto --> override to Standby
1 = not used --> override to Standby
2 = occupied
3 = off (not availbale on TR80)
4 = holiday (only availbale on TR80)
5 = unoccupied
6 = standby
7 = bypass (only availbale on TR80 from UnOCC)
4 = OCCUPIED
1 = PROTECT --> auto - jump back previous
2 = UNOCCUPIED
3 = ECO /STANDBY
4 = OCCUPIED --> auto - jump back UnOCC
TR80 possible Occupancy Mode
0 = unknown / auto
1 = not used
2 = occupied / comfort
3 = off
4 = holiday
5 = unoccupied / away
6 = standby
7 = bypass
When Occupancy comes from PIR sensor:
It only works if the Room application is in ECO mode.
When motion is detected, it changes to OCCUPIED mode for at least the duration of PIR coasting tme.
Addtional motion detection will extend the OCCUPIED status and the coasting time is restarted.
When the External scheduler is changing the Master Mode to Unoccupied then only the TR80 can be used to override the occupancy mode.
When the External scheduler is changing the Master Mode to OFF then it is not possible to change the occupnacy mode form TR80 or from PIR.
It is possible for the External scheduler to change the Master Mode to OCCUPIED, but this action is normally not needed.
The PIR sensor will provide the motion detection in the room, so the Room Application can change to OCCUPIED mode.
Or the User of the room can override the ECO / Standby mode to OCCUPIED in the TR80.
User Override will be deleted if the External scheduler changes to OFF or UNOCCUPIED mode as there is no User supposed to be in the room.
Holiday override will not be deleted.
Or the Occupancy Auto Reset of TR80 used to delete the unnecessary overrides.
There is an override reset connected to 18:00 o'clock, which resets also User Occupancy Override, but it may not be the right way for sophisticated room control.
The recommandation is to plan properly the Occupancy modes with taking care of the capability of TR80 and the used Room HVAC Application.
And implement the plan carefully based on this template.

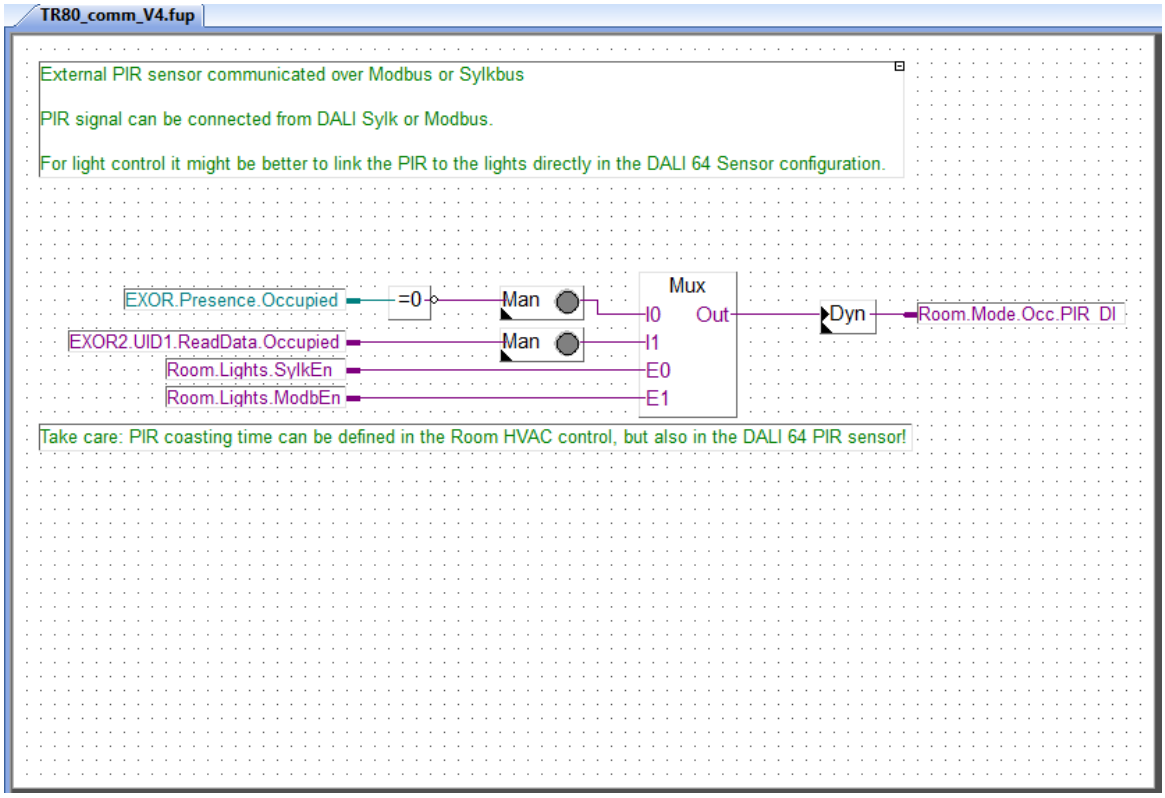
Occupancy configuration

This page configures the TR80 occupancy related settings to match to the occupancy concept.



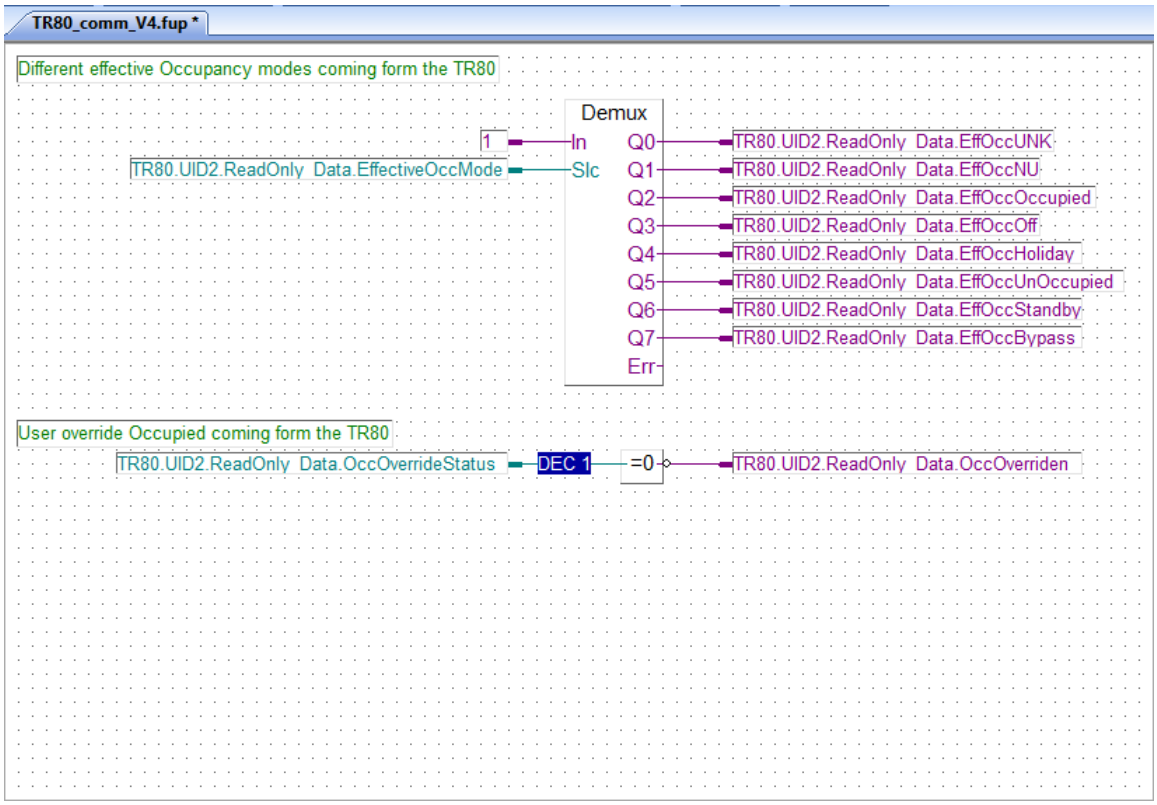
Occupancy from PIR sensor

This page hands over the PIR sensor (motion detector) signal to the Room Application. It is used there to switch the Standby (Economic) mode to Occupied (Comfort) mode when motion is detected.



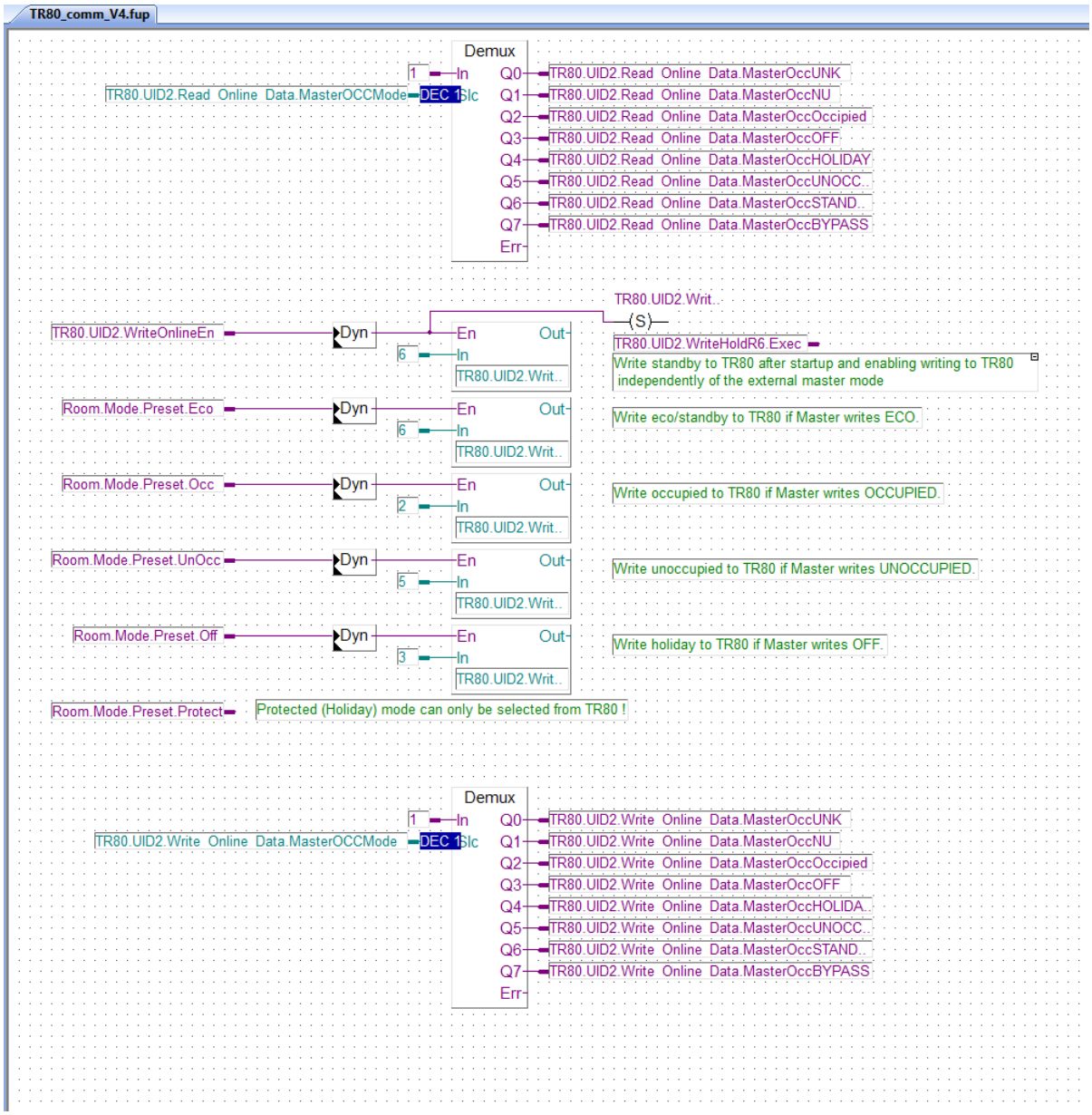
Effective Occ and Override

This shows the Effective Occupancy mode. This mode is the summary of the Master and User Occupancy Mode in TR80. The Master Occupancy mode is coming from External Scheduler (SCADA, Plant Controller) and it is sent to TR80. When User do change of User Occupancy Mode of TR80, the change overrides the Master Occupancy Mode for short or longer period (depending on which mode has been selected) and the result is the Effective Occupancy Mode. The Effective Occupancy mode is transferred to the Room Application. This Occupancy mode has effect only on the Room Application. If Light or Blind must be controlled according to this Occupancy, then it must be programmed so.



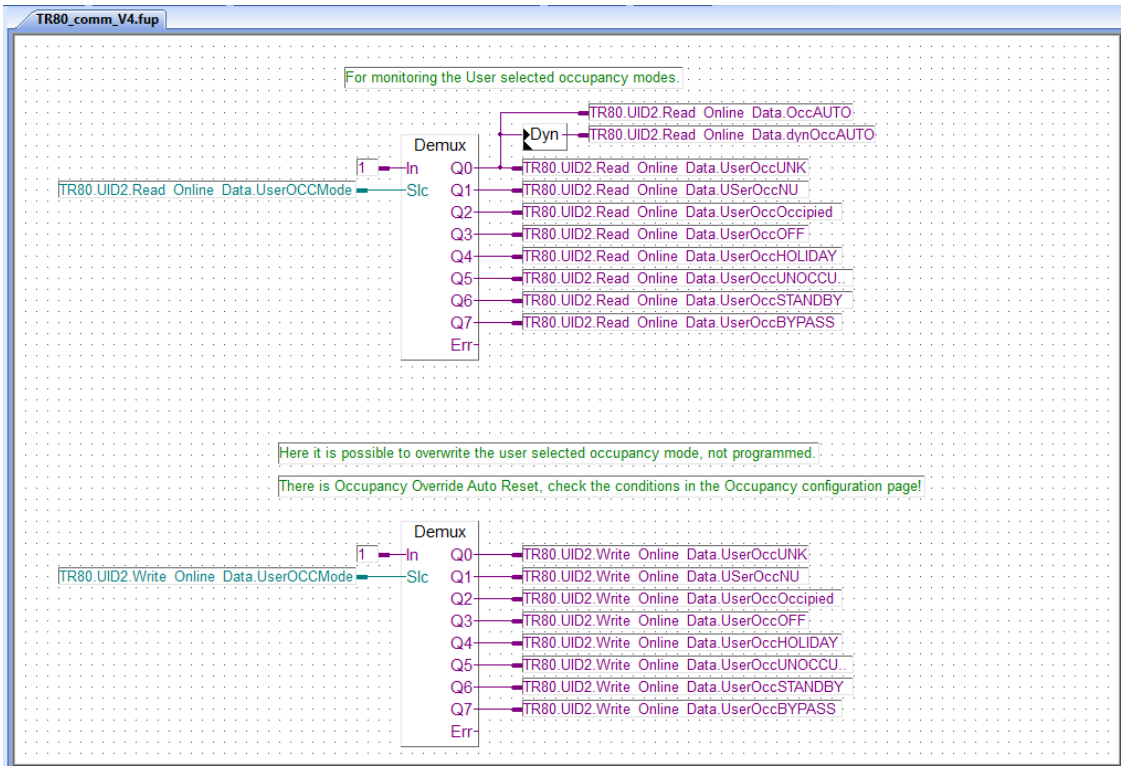
Master OCC Read / Write

This page hands over the External Scheduler of Occupancy (SCADA, or Plant controller) to the TR80 Master Occupancy mode. The changes of the mode is detected by other part of the program, and it is transmitted to TR80 to show the actual occupancy mode. There is a scheduled read of the same datapoint of TR80, where it can be seen, if the change is properly transmitted.



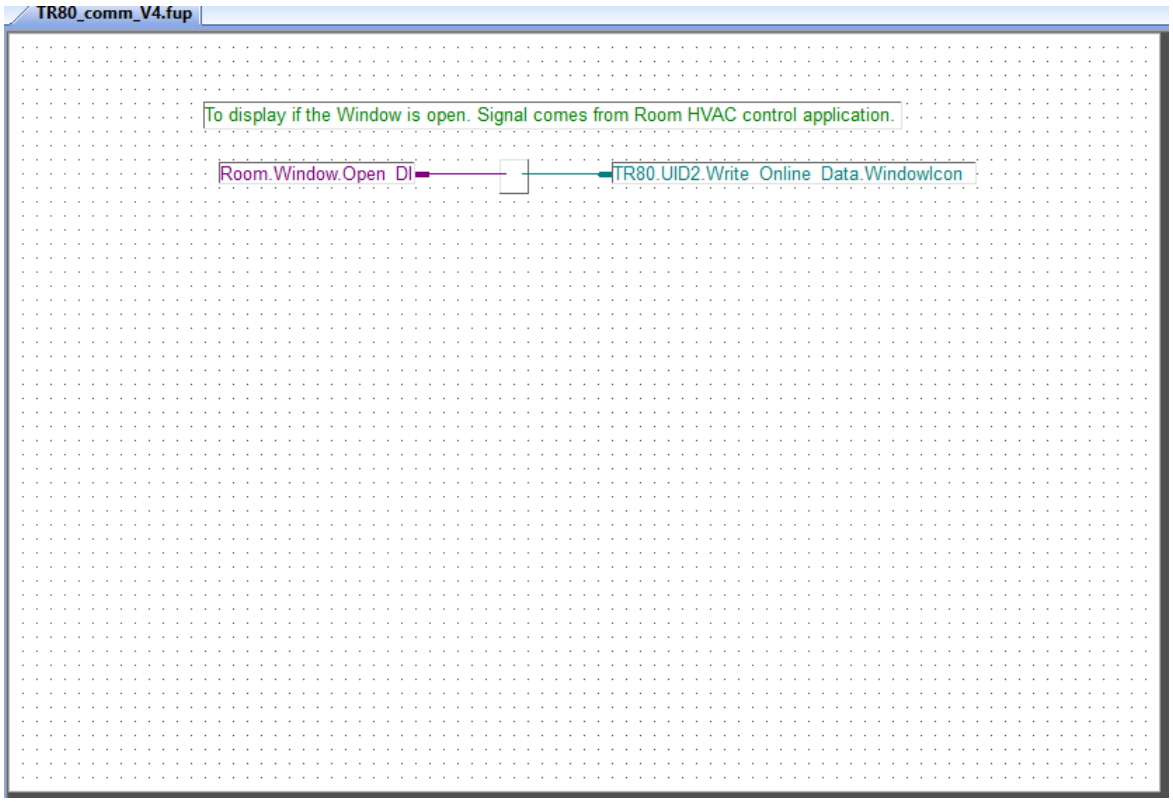
User OCC Read / Write

On this page it can be seen the User overrides of Occupancy modes. By writing value back the user override can be manipulated or cancelled. It is not programmed here, because program uses another mechanism to cancel the user overrides, or it will be cancelled automatically.



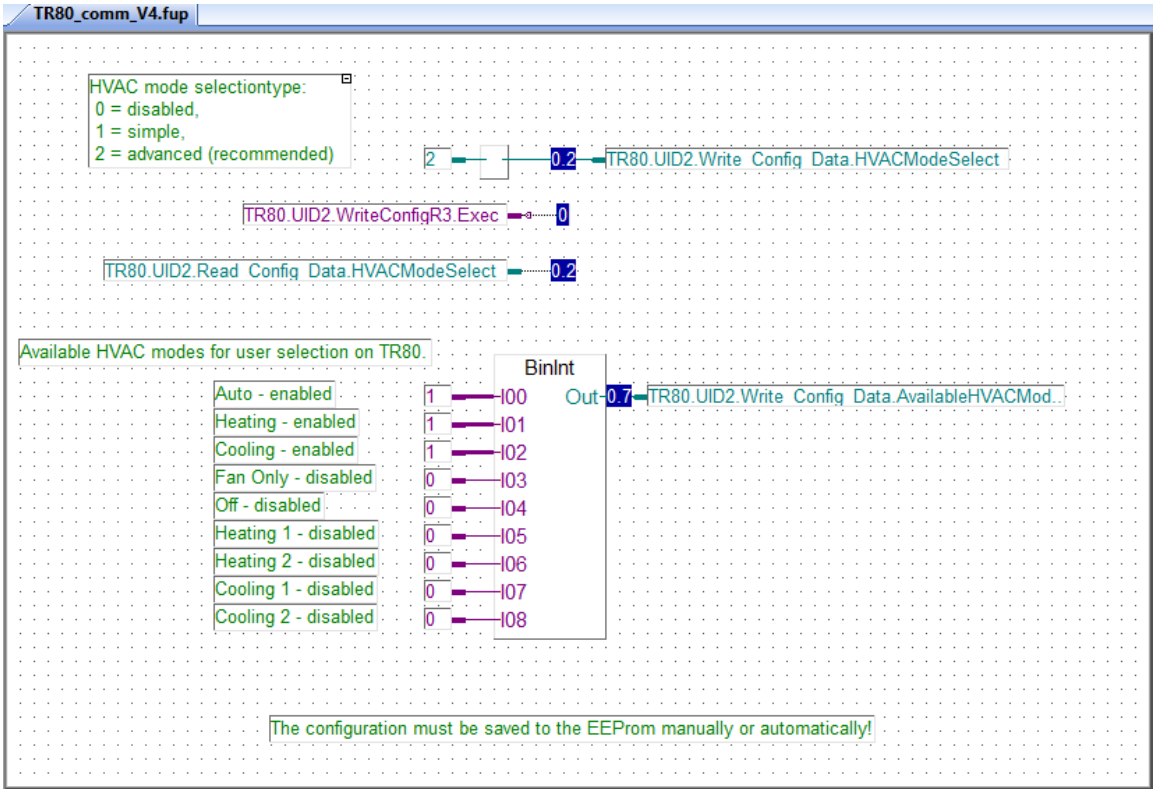
Window Open handover

The logic on the page forwards the Window open signal to the TR80 to be displayed.



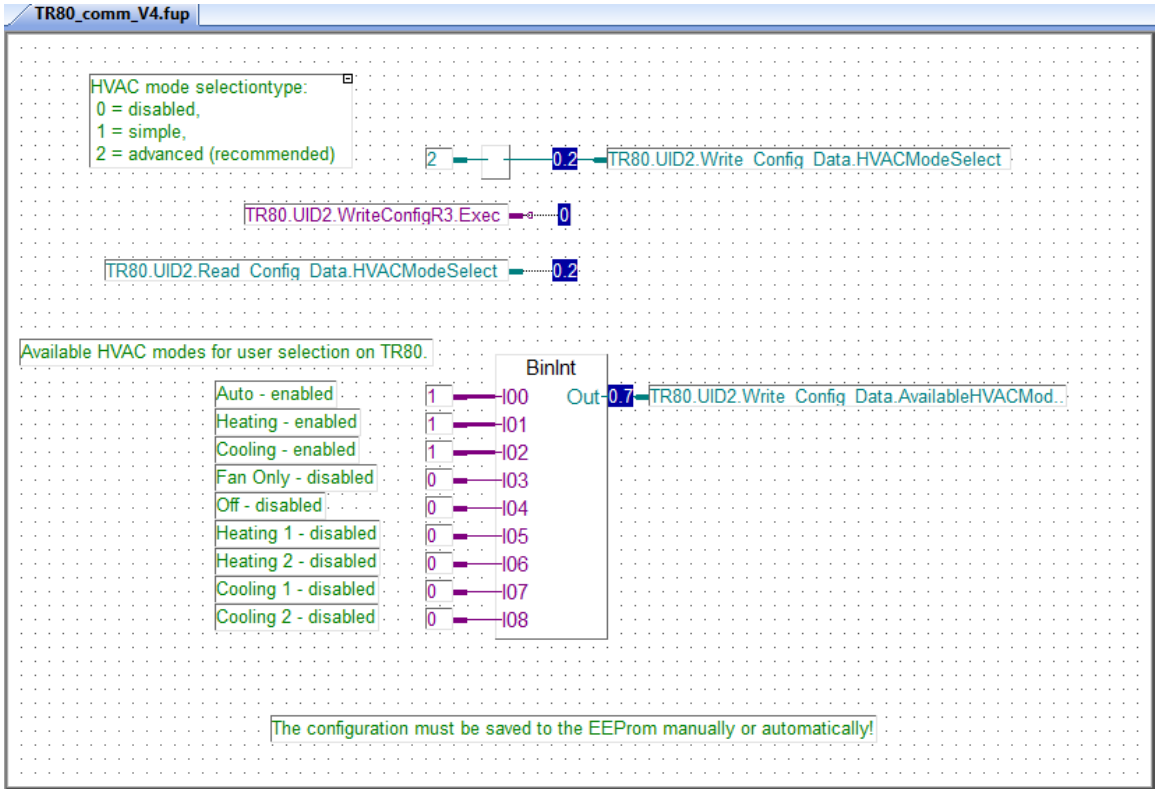
Preconfigured HVAC modes

Here the HVAC mode is configured as advanced mode. This mode enables the room application to send signal to TR80 to change the colour of the LED ring according to the real HVAC mode applied in the room. Or it is possible to send the user overrides to the Room Application.
In the configuration the Auto, Heating only, Cooling only are enabled, means user can select only these modes on TR80.



Preconfigured HVAC modes

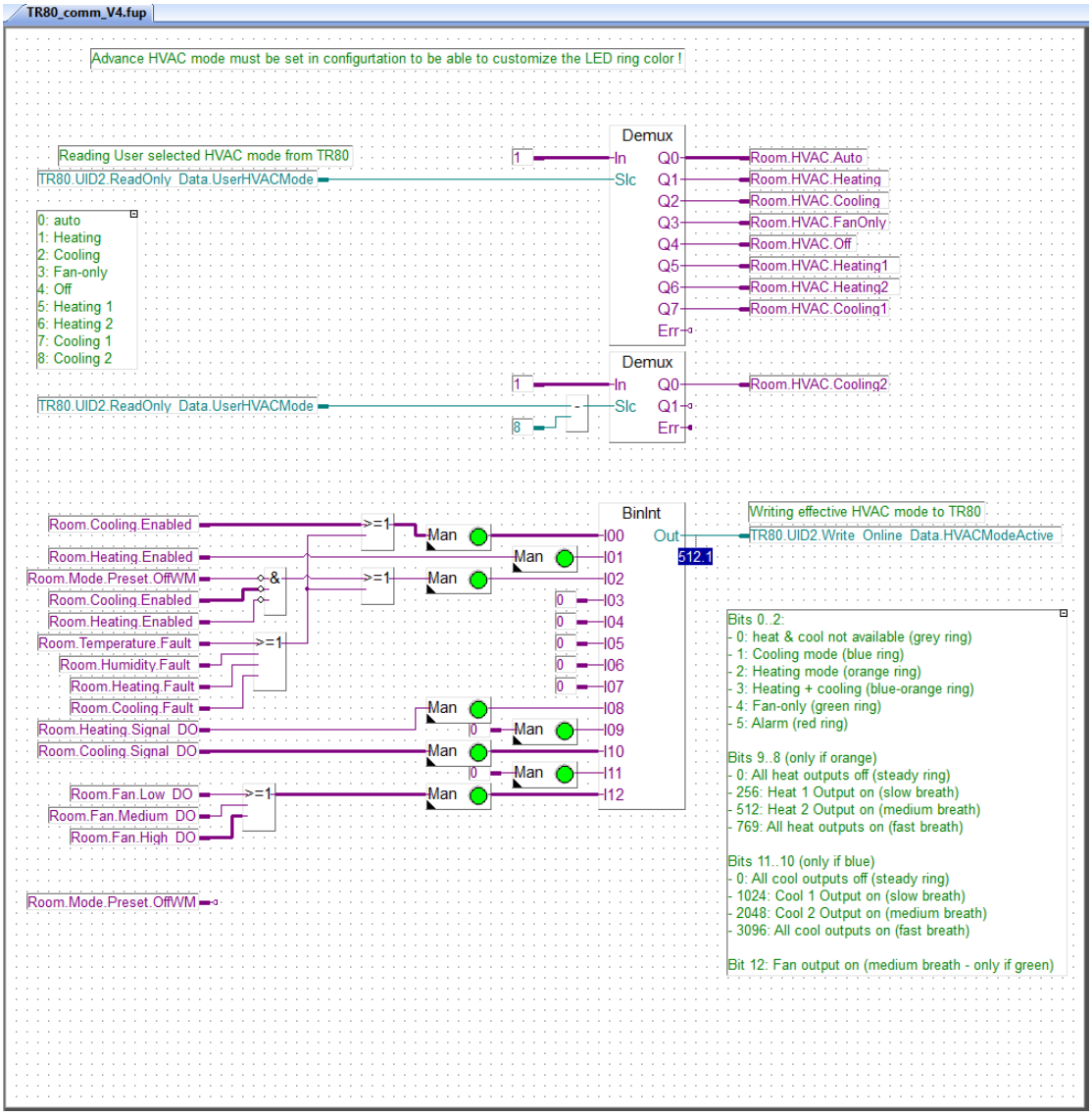
Here the HVAC mode is configured as advanced mode. This mode enables the room application to send signal to TR80 to change the colour of the LED ring according to the real HVAC mode applied in the room. Or it is possible to send the user overrides to the Room Application.
In the configuration the Auto, Heating-only, Cooling only are enabled, means user can select only these modes.



HVAC modes, LED ring behaviour

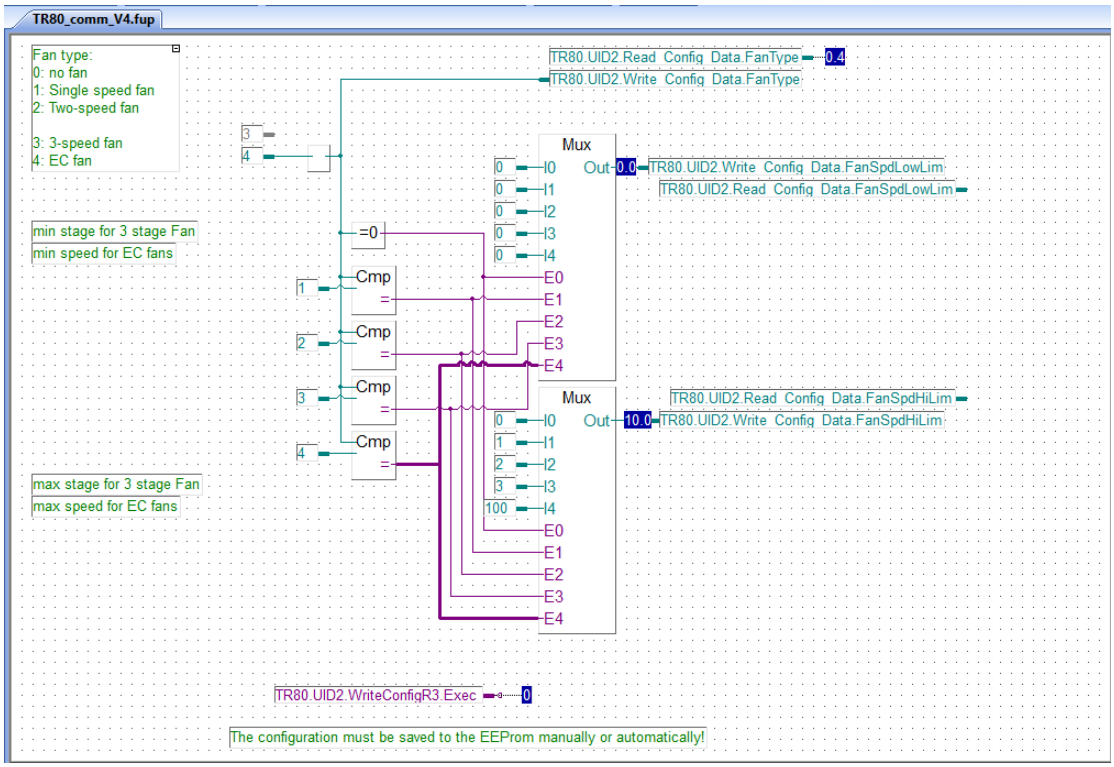
First on this page the, User selected (override) HVAC mode is handed over to the Room Application. The user selected modes are displayed as Icon on the TR80.

Second on this page, the current behaviour and general alarm coming from the Room Application is transmitted to the TR80. The current HAVC mode is then displayed on the LED Ring, around the setpoint arrows, with different colour and “breathing” of the intensity.
The explanation of the different enumeration of TR80 colour behaviour is in the FUPLA page.



Preconfigured Fan modes

Here the Fan mode is configured as EC fan (0-100% controllable) for the HVAC Room application. The minimum and maximum speed limit must be also set to 0% and 100% according to the used fan type.

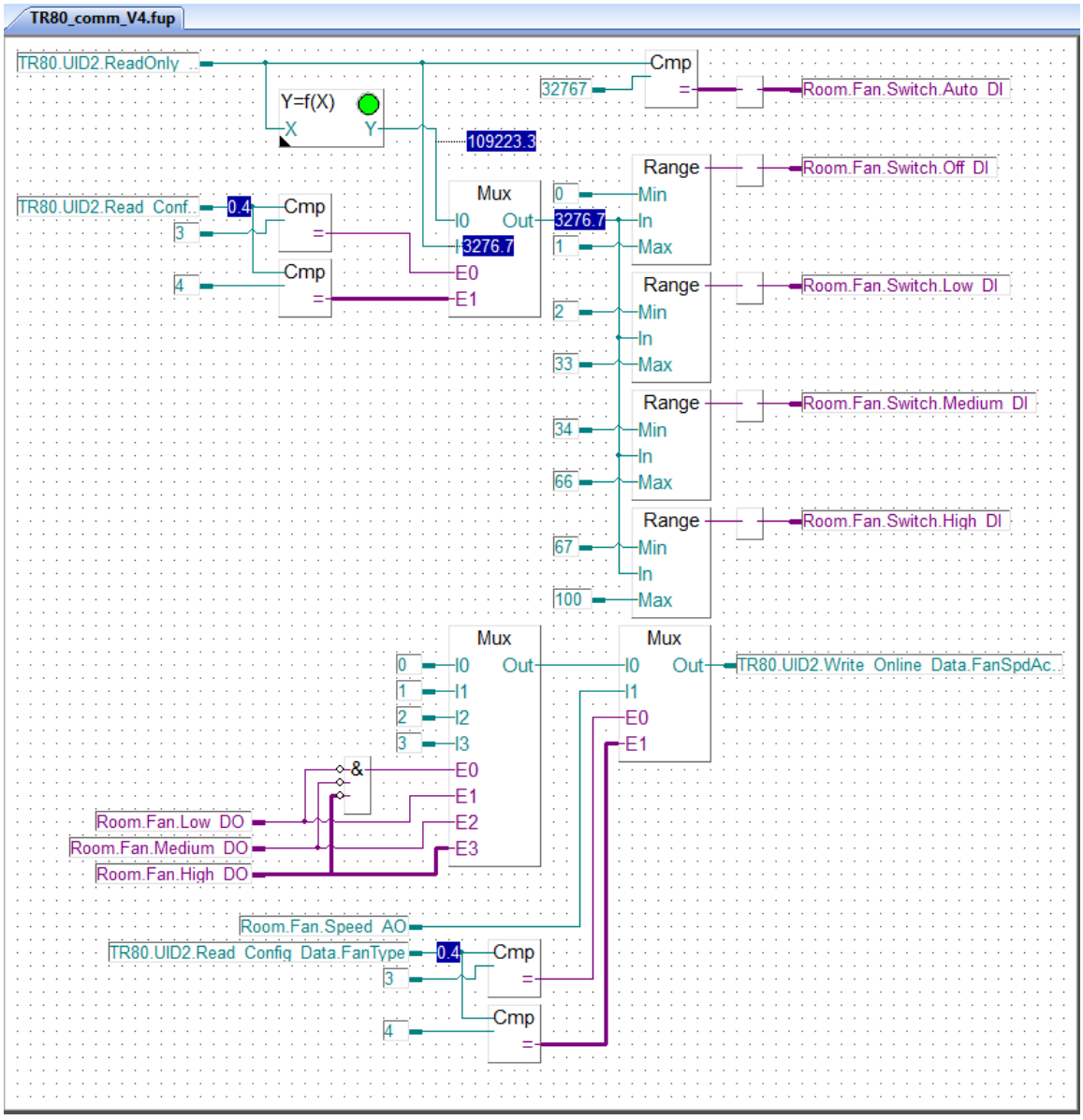


Fan modes, displayed fan feedbacks

In this page the User Fan speed override is processed and handed over to the Room Application. The Room Application is waiting for Fan feedback as stages, so the continuous Fan speed is staged here.

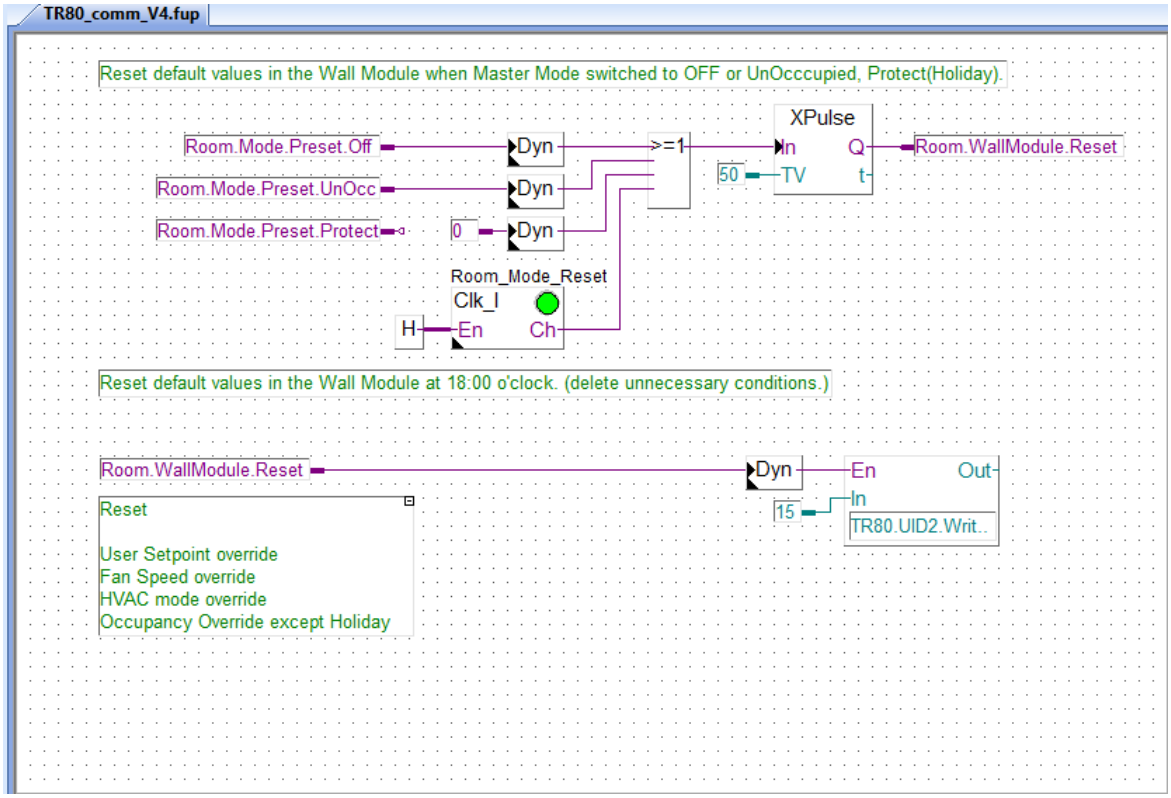
The real fan speed is transmitted back to the TR80.

The User fan speed and the fan speed feedback calculation is depending on the Fan Type (continuous or 3 stage fan).



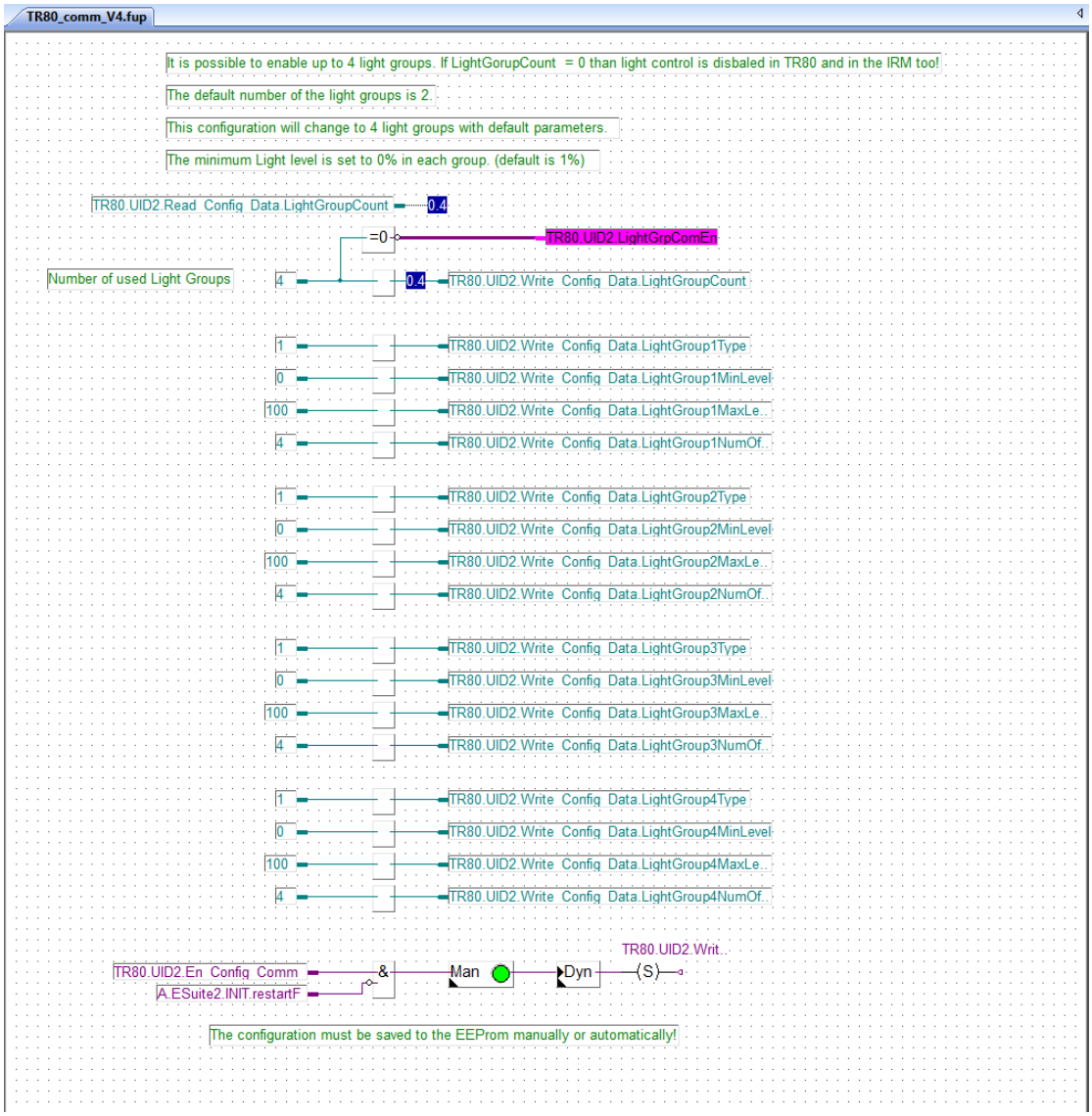
Wall module reset

Here the Wall module operational values and user overrides are getting reset (set back to default) not the wall module itself!



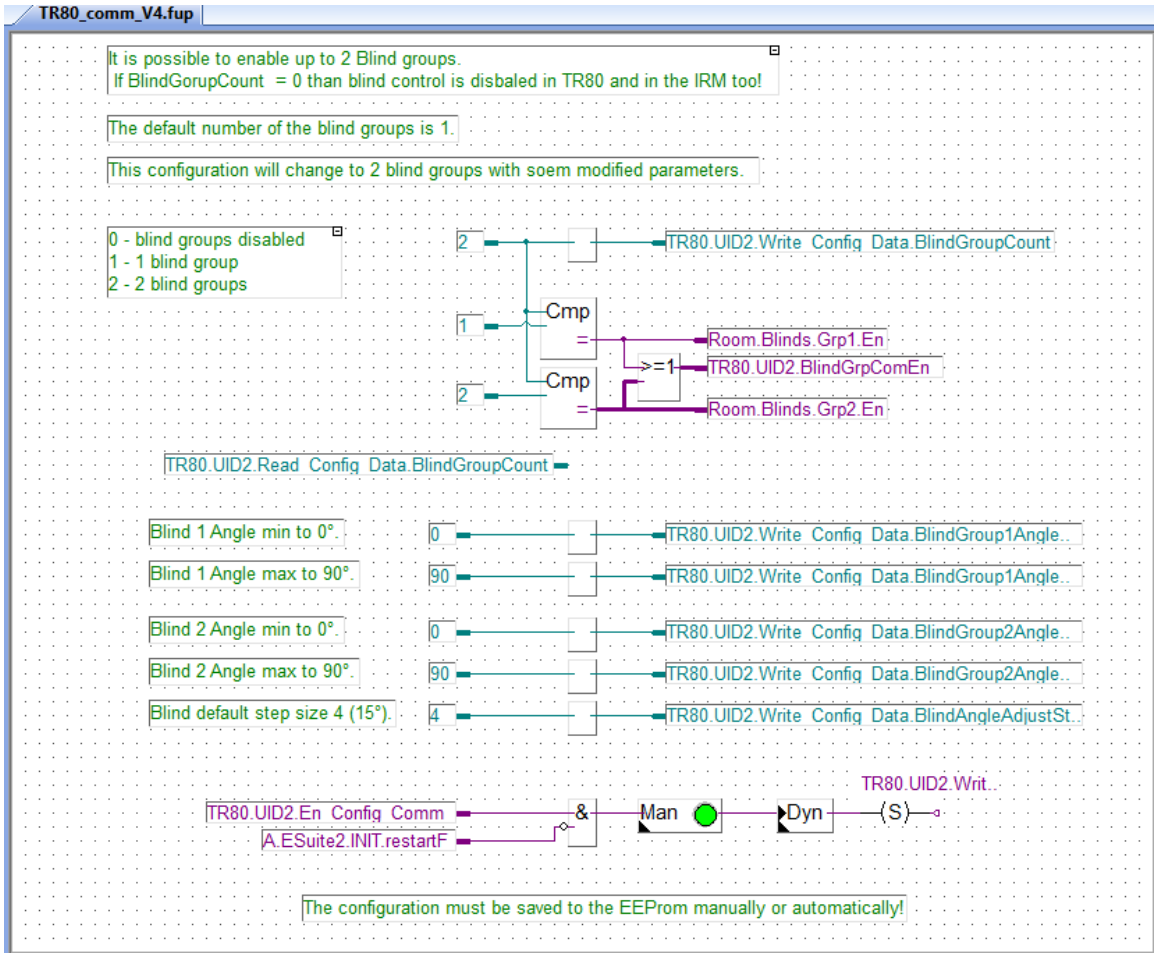
Preconfigured group of lights

The TR80 can handle up to 4 light groups with dimming the intensity (0-100%) or setting up scene number (1-4) per group. This page sets up the TR80 to handle 4 groups of lights. All lights are set to the same type dimming + scene, but it is possible to setup different behaviour, see the ModbusRegistersV1.1.25.0.xlsx “group type” configuration.



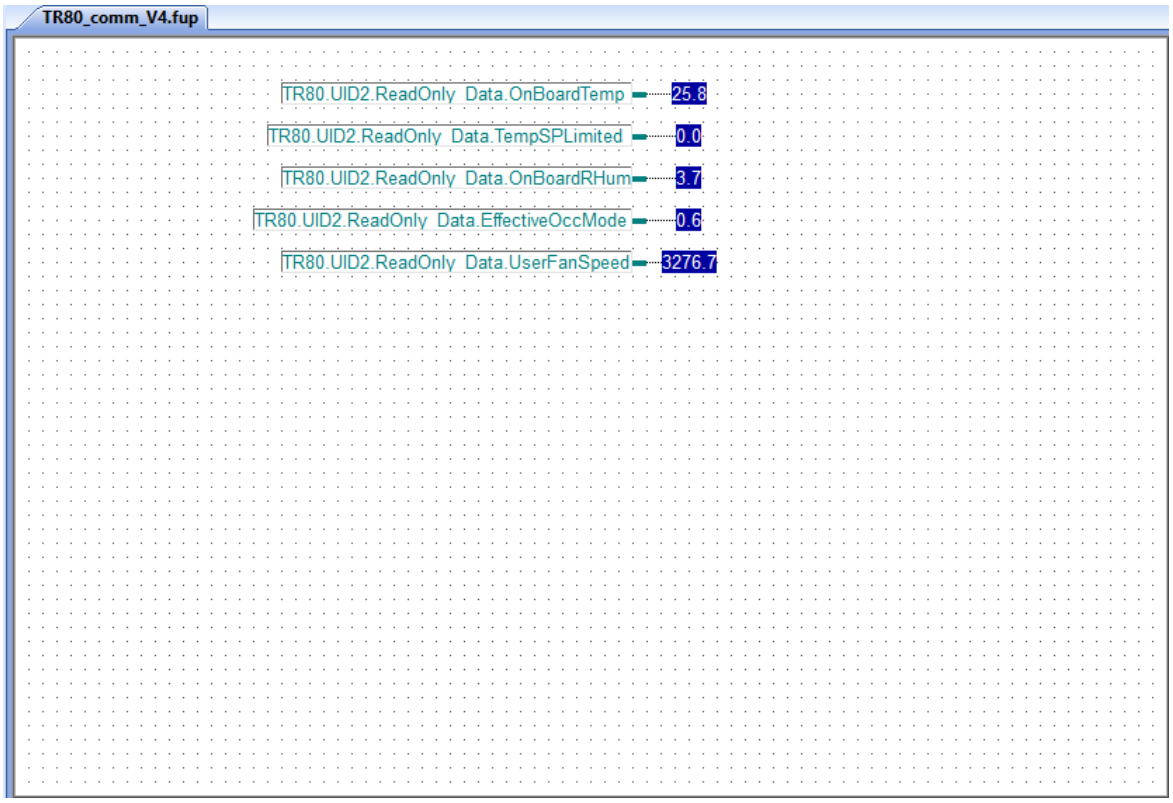
Preconfigured group of lights

The TR80 can handle up to 2 blind groups with position and angle control. This page sets up the TR80 to handle 2 groups of blinds. All blinds are set for position and angle control but possible to setup only position control if need, see the ModbusRegistersV1.1.25.0.xlsx “group type” configuration.



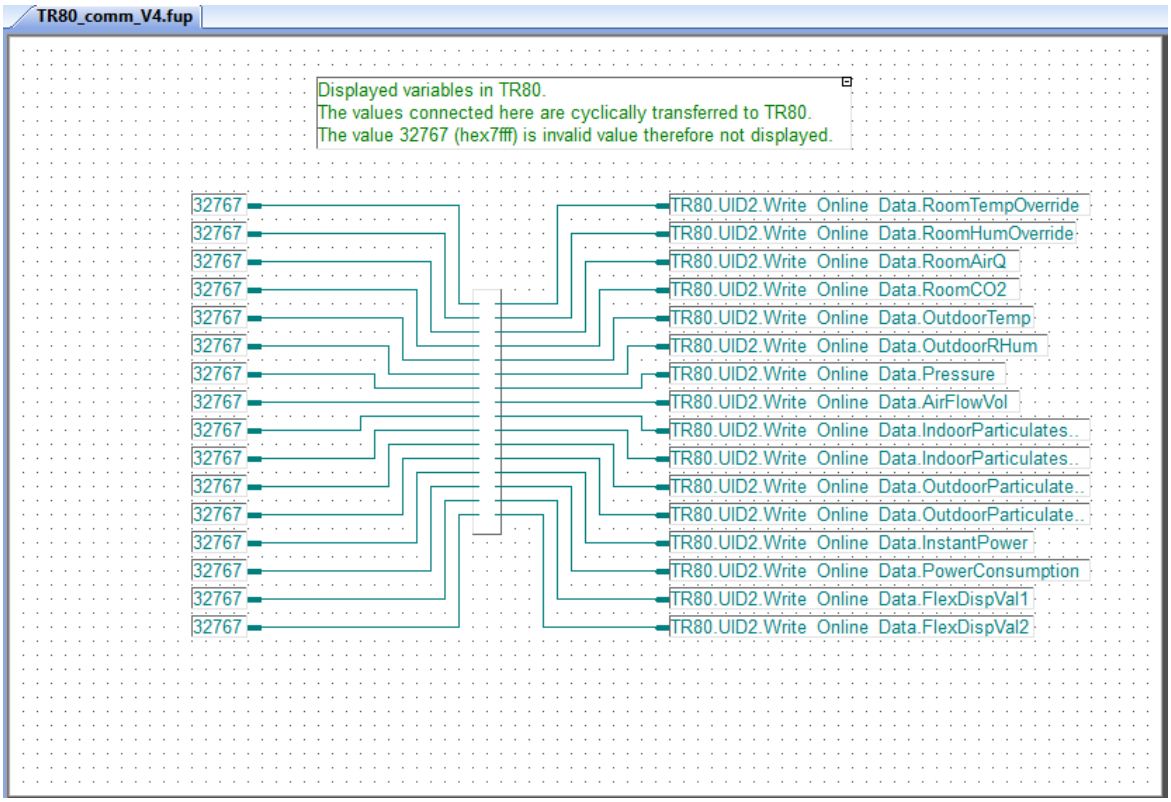
Important Values Summary

Important values from TR80 are collected here to monitor and to see if the configuration has been properly applied, if needed more symbols (values) can be added to the page.



Displayed values

TR80 is capable to display more measurements coming over Modbus communication or from the physical input. The values can be connected to this page and will be transmitted to TR80. If the value is “32767” TR80 does not display it. It is also possible to setup engineering unit for the values and possible to send some ASCII characters to TR80 through FlexDispVal1 and FlexDispVal2.

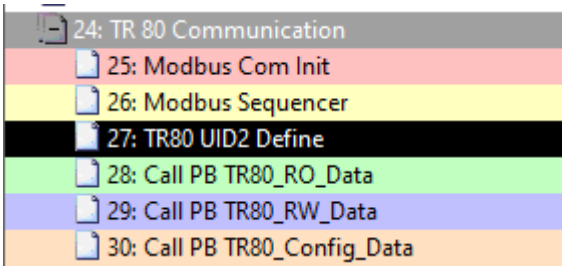


TR80 communication

The communication to TR80 is based on Modbus RTU. The Port 0 of the controller is used to connect to the TR80 RS-485 port 1. The default setting is applied for communication parameters (19200 baud, even parity, 1 stop bit, address:2).

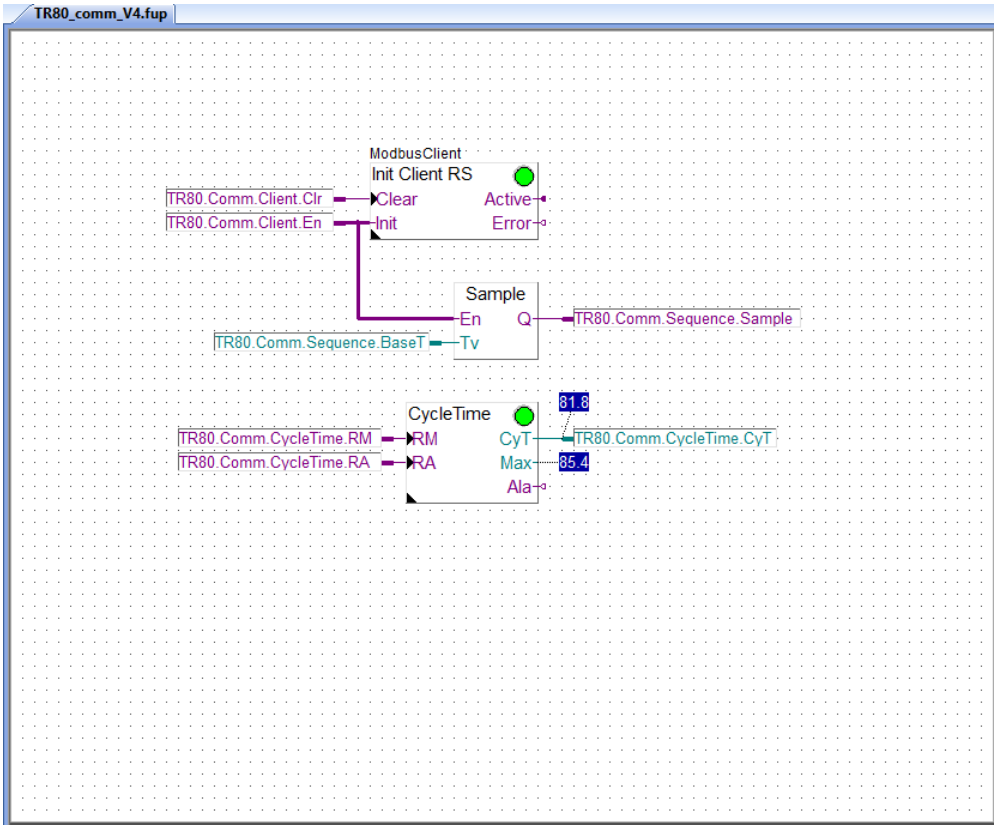
The TR80 offers lots of parameters to communicate, these are described in the ModbusRegistersV1.1.25.xlsx excel file.

Basically, Read Only online data (light green), Read/Write-able online data (lilac and blight blue) and Configuration data (orange and yellow) are communicated. The corresponding pages are colorized in the Page Navigator.



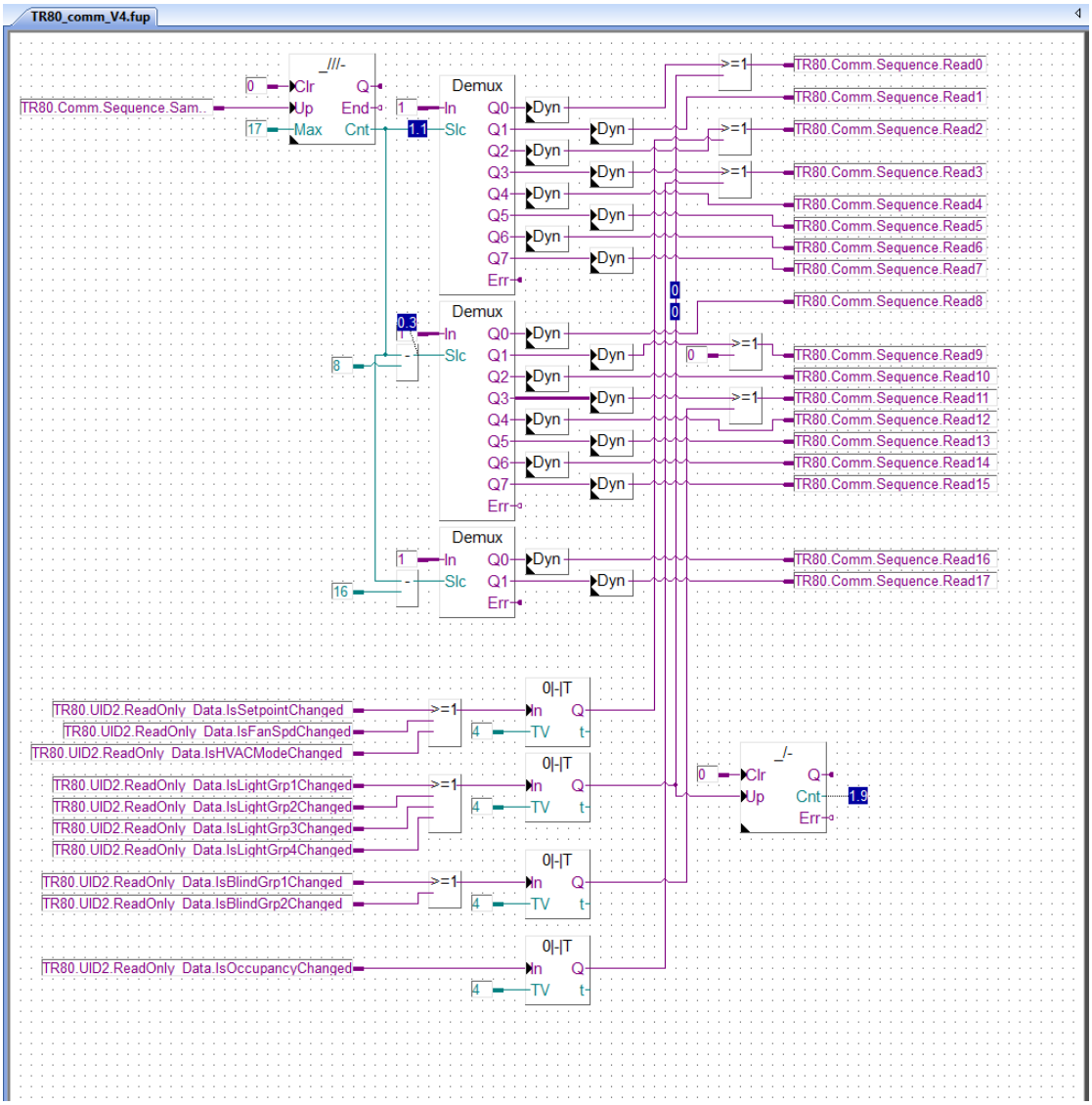
Modbus Com Init

Defining the Port 0 of the controller for Modbus communication with the right communication parameters.
Enables the Read sequencer
Checking the cycle time of the program.



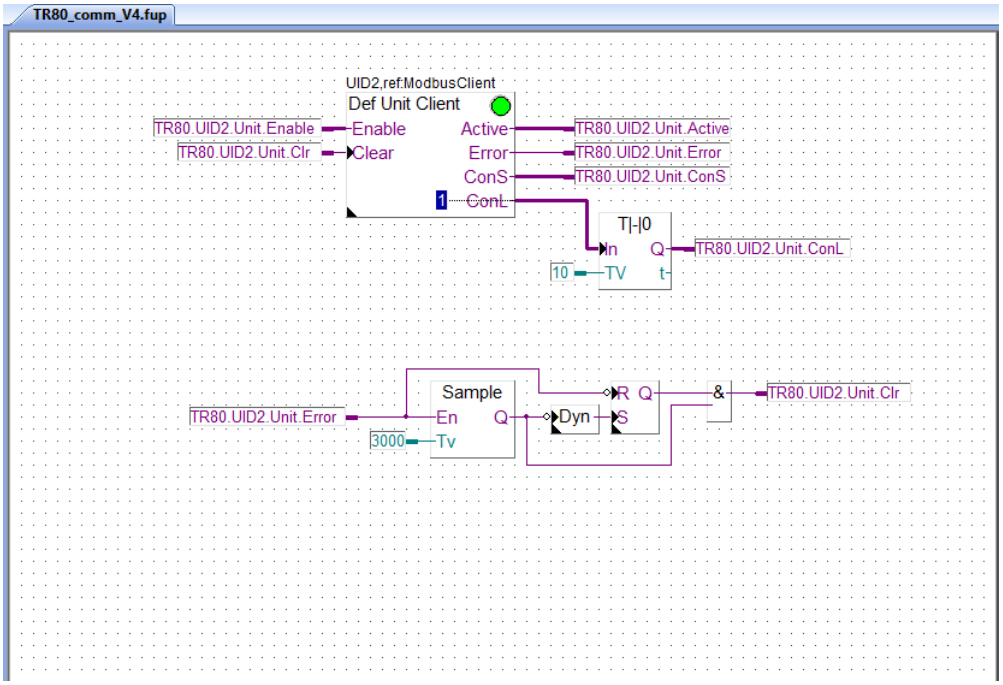
Modbus Sequencer

It is not possible to read all the data from TR80 cyclically wit reasonable communication cycle time. Therefore, a sequencer is used to enable only one read per second. There are 18 reads which are sequentially organised. There is one important read which is cyclically sent. This important read has information about the user actions, e.g. user has changed the setpoint or dimmed the light. This information will force the Read sequence to enable the Modbus read with 400ms off-delay the specific topic like HAVC, Light, Blind, Occupancy, to read the user actions properly. For example, reading the light dimming values as fast as it is possible when user is changing the light level. The programmable IRM controller is considered as slow device. If the unnecessary parts of the program are removed and/or faster controller is used, it can happen, that this sequence mechanism is not needed.



TR80 UID2 define

Defining the Modbus communication partner TR80 has Modbus address 2.
Automatically resets the error communication error after 5 minutes.
The ConL is used to monitor if the TR80 is not communicated for longer period. Take care, if this happen the program supposes that TR80 was exchanged, and the program starts the configuration mode.

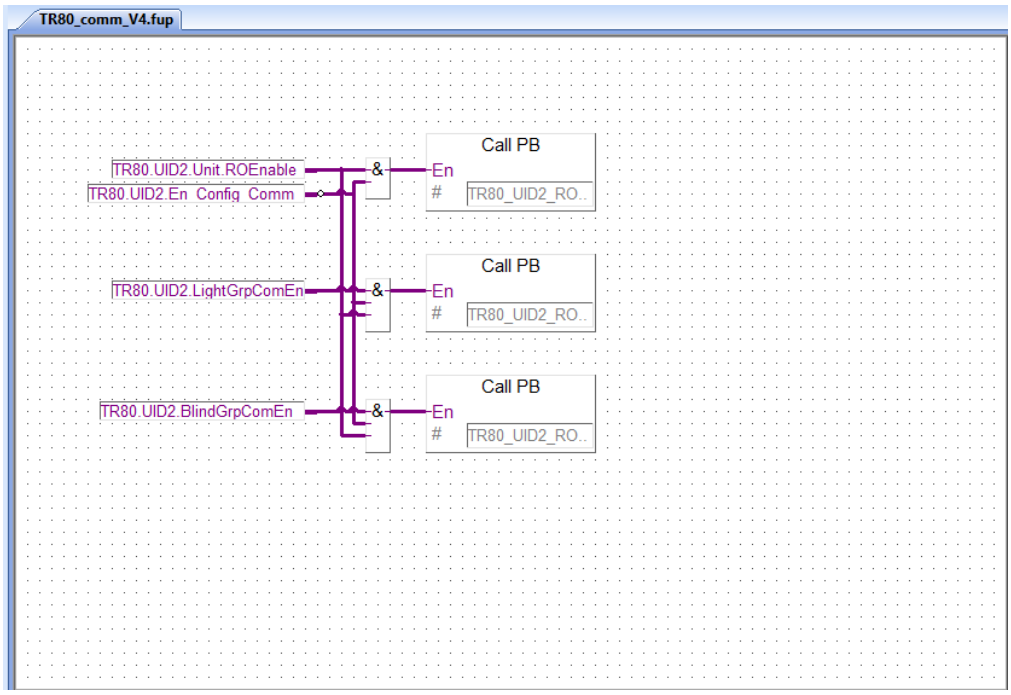


Call PB TR80_RO_Data

Three different Program Block is created to read the so called Read Only online data from the TR80.

1. Room HVAC application data
2. Light groups related data
3. Blind groups related data

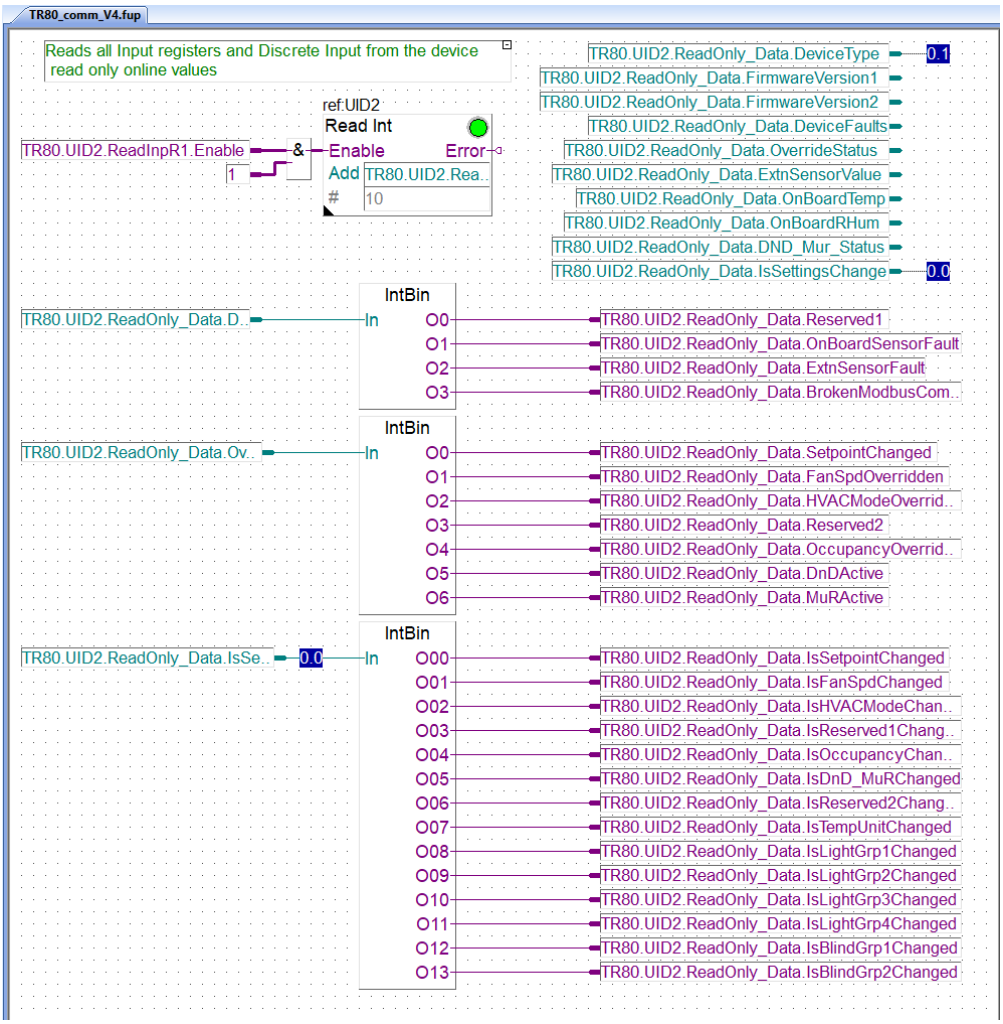
With this segmentation it is easier to switch off (or completely delete) those program parts which are not needed.
The communication is enabled, if the configuration has been done.



Online Read Only data - Read values Device sensors

Online read only data means that, such data are provided by TR80 as it is, and they are not overwritable. The Modbus register map of the TR80 is divided into topics such as Device Sensor, Operating status, HVAC, Light groups, Blind groups, etc.
The program is capable to read the online read only data per topic. This gives the possibility to switch off those reads (topics) which are not needed.

Example of reading read only data:
This Modbus read request is always enabled



The Read Integer function box is reading the Device sensors related data cyclically.
Some data is binary and compacted into registers. The binary data is then extracted to Flags by the program.

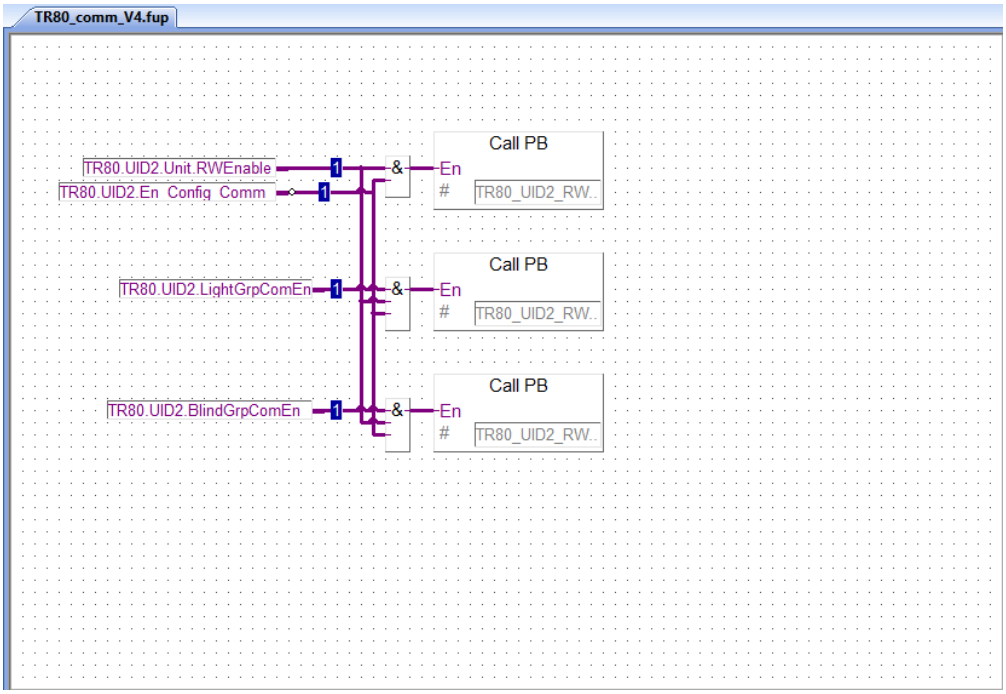
All the readings of the other topics of online read only data are connected to the Modbus sequencer described above.

Call PB TR80_RW_Data

Three different Program Blocks are created to read and write the so called Read/Write online data from the TR80.

1. Room HVAC application data
2. Light groups related data
3. Blind groups related data

With this segmentation, it is easier to switch off (or completely delete) those program parts which are not needed.
The communication is enabled if the configuration has been done.



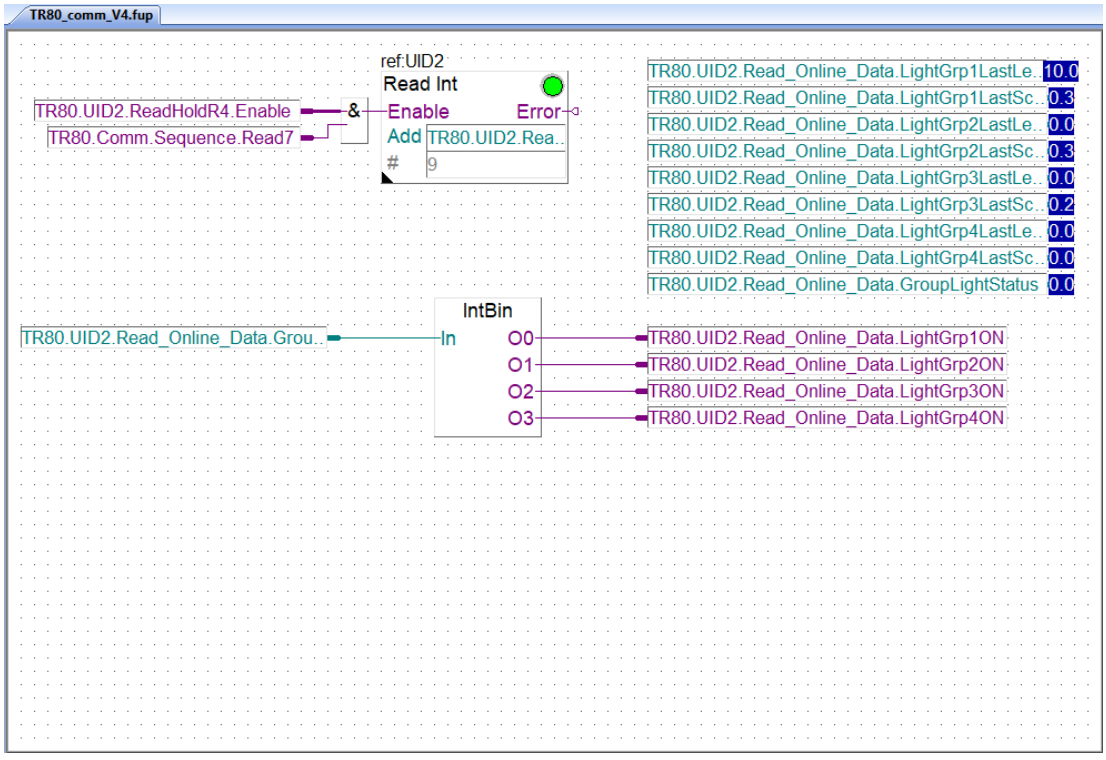
Online Read-Writable data - Read values Device sensors

TR80 has lots of values which are basically online data, but overwritable. A good example for such a data the room temperature setpoint. The user/guest of the room can change the room temperature setpoint any time, but also the controller program (BMS system) can overwrite the room temperature setpoint (no guest → go to default setpoint). Obviously the new setpoint, coming from any source must be displayed in the TR80 properly.

The Modbus register mapping of the TR80 is divided into topics such as Device Sensor, Operating status, HVAC, Light groups, Blind groups, etc.
The program is capable to read and write the online read-writable data per topic. This gives the possibility to switch off those Reads&Writes (topics) which are not needed.

The Light groups topic is shown as example how the program works.

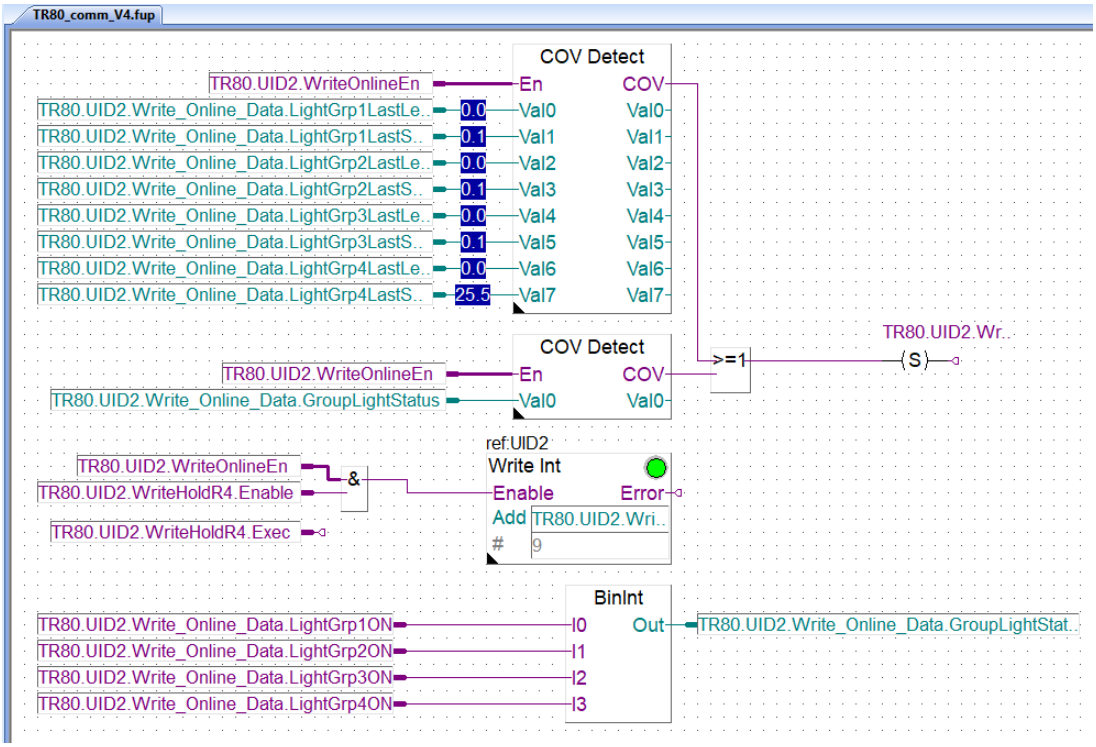
Read Values Light Groups



The Read Integer function box is reading the Light groups related data, the “Last Level” and the “Last Scene” according to the Modbus Sequencer.
Some data is binary and compacted into registers. The binary data is then extracted to Flags by the program.
The Last Level and Last Scene are normally update by the TR80 itself when the user is changing the light level or scene. In this case the controller needs only to read them out to be aware of the current light level or scene in the room.
It is also possible, that light level or scene is controlled by the program, so the controller is sending to all lights to 100% (e.g. at housekeeping) over Sylkbus or Modbus. In this case the controller must update the TR80 device with the new light level for that light group to show the real light level on the display. So, controller must write back the last light level to TR80.

All the readings of the other topics of online read-writable data are working in the same way.

Write Values Light Groups



Writing the online read-writable data is a bit complicated.
First it is necessary to understand how the light group control is working.

Getting group 1 light level command from TR80

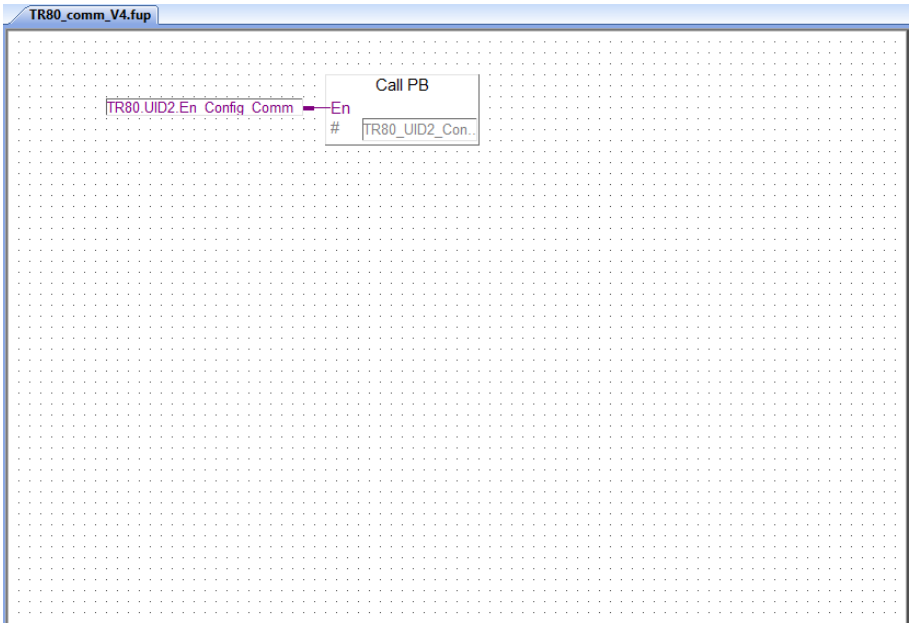
The Group1 new light level from TR80 comes as **online read only data** to "TR80.UID2.ReadOnly_Data.LightGrp1NewLevel". When the program is reading the new light level of the Light Group 1 it usually gets "32767" in the "TR80.UID2.ReadOnly_Data.LightGrp1NewLevel" register. This value is a kind of dummy value which must not be executed by the light driver (DALI64).
When the "TR80.UID2.ReadOnly_Data.LightGrp1NewLevel" register has a value from 0-100% then this must be executed by the light driver. So, the program forwards this value to the Ex-Or cube over Sylkbus or Modbus.
The TR80 inside updates his Group 1 last level writable parameter, but this is getting to be overwritten by the level feedback coming from the DALI64 sensor. The program reads back the Light intensity level from the DALI64 sensor and update the TR80 "Last Level" with the real feedback. This means the program writes the level feedback to the "TR80.UID2.Write_Online_Data.LightGrp1LastLevel" register, which is sent to the TR80 when change of value detected. After the write, read command is sent to read the value back to the "TR80.UID2.Read_Online_Data.LightGrp1LastLevel". Then the process is finished.

Getting group 1 light level command from controller

The controller writes the new light level to the Ex-Or cube over Sylkbus or Modbus. Then the program reads back the Light intensity level from the DALI64 sensor and update the TR80 "Last Level" with the real feedback. This means the program writes the level feedback to the "TR80.UID2.Write_Online_Data.LightGrp1LastLevel" register, which is sent to the TR80 when change of value detected. After the write, read command is sent to read the value back to the "TR80.UID2.Read_Online_Data.LightGrp1LastLevel". Then the process is finished.

Call PB TR80_Config_Data

When the communication is enabled in the first 20 seconds the Configuration data are transmitted and read back.

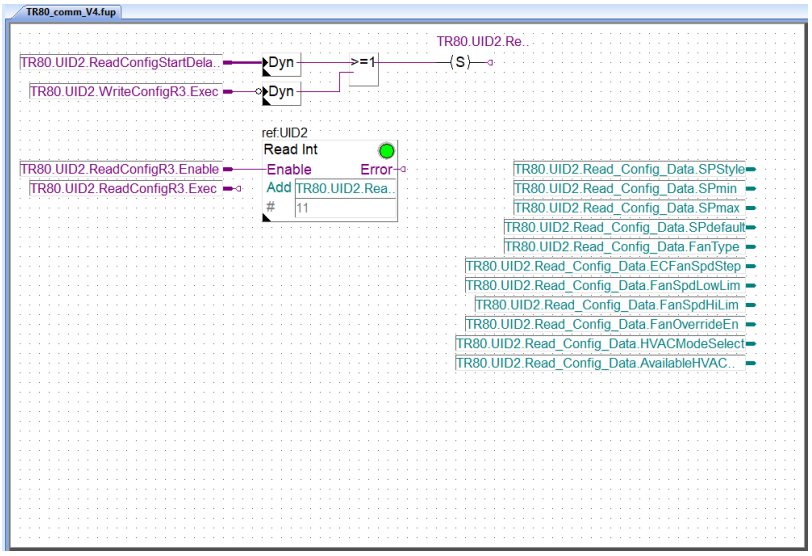


Configuration Read-Writable data

TR80 has lots of parameter to change the behaviour of different components (communication ports, pushbuttons, displaying elements, external / internal sensor behaviour etc.).
It is possible to change the configuration of TR80 from the FUPLA program, but also possible to change the configuration directly in the configuration menu of TR80. Later (with new FW variants) it will be possible to select between profiles in TR80 config menu. The profile has predefined configuration valid for an application e.g. hotel room. Then the SI does not need to go through all the configuration parameters and set them, but he needs to select the corresponding profile.
Using these profiles, it is likely, that the Configuration Read-Writable data part of the program becomes unnecessarily and can be deleted.

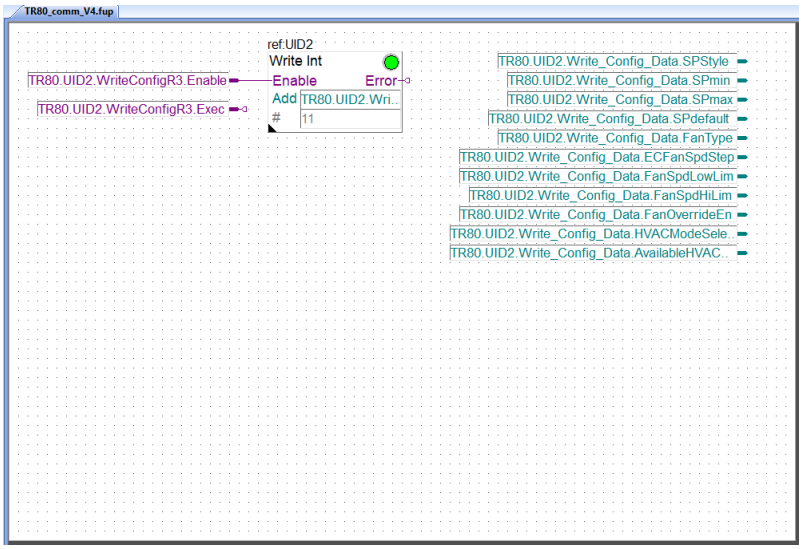
In the program, the Configuration Modbus register mapping of the TR80 is divided into topics such as COM ports, HVAC, Light groups, Blind groups, etc. The HVAC topic is shown as example how the program works.

Read config HVAC



Reading the configuration parameters cyclically are not needed, so read happens only after start-up of the controller once.
And if there was a configuration write command, after it the program reads back the configuration from the TR80.

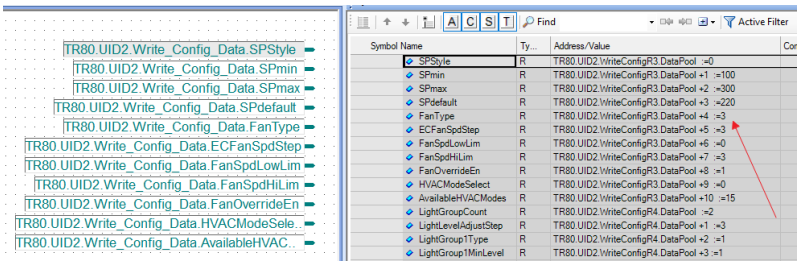
Writing config HVAC



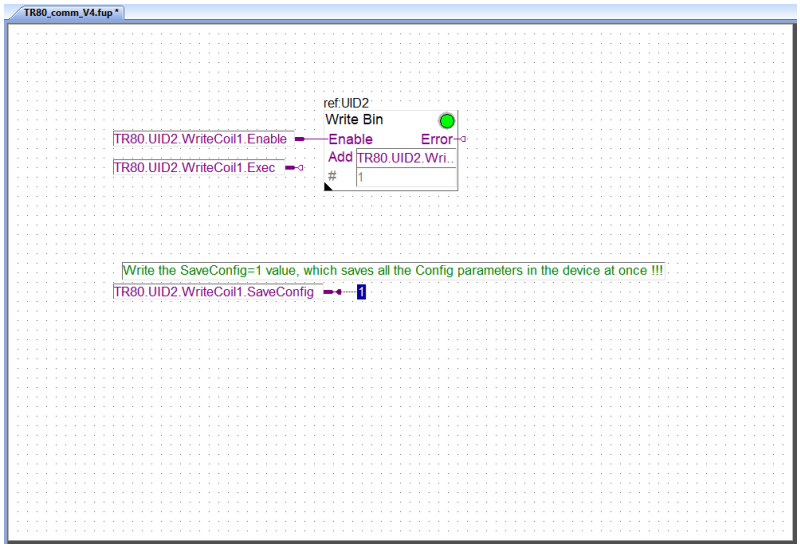
Writing the configuration parameters cyclically are not needed, so one time-write happens in the first 20 second after the controller starts-up and Modbus communication is enabled. The program tries to the detect if the TR80 is exchanged (communication is lost for longer period) and enables the configuration.

In this example, the Fbox is writing 11 configuration parameters together into TR80. This means even though the SI wants to change one parameter, the other are also getting to be written.
So, all the other parameters must be pre-set too with the right value!

Each config write register has the TR80 default parameter set as **Init value** in the symbol editor.



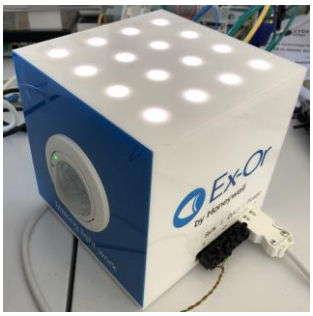
Saving Configuration to the EEPROM



The configuration parameters are stored in EEPROM of TR80. The parameters are automatically saved to the EEPROM when configuration happens, and if the “TR80.UID2.SaveCfgEn” flag is high. See the “Calling TR80 configuration” page.

DALI64 over Sylk - / Mod -bus

DALI64 is a fully featured DALI lighting control system embedded within a best-in-class high performance PIR sensors using DALI64 communication for dimming lighting loads.
There are two variants of sensor used in this project, Sylkbus/DALI64 and Modbus RTU/DALI64. Using the sensor, a demo device with 16 dimmable LED lights is created called Ex-Or cube



The sensor has intelligence to behave as DALI64 master unit with a Sylk- / Modbus gateway, an external master (in our case the room controller) can send command to the lights through the gateway.

In this application, at one time only Modbus or Sylkbus device can be used!

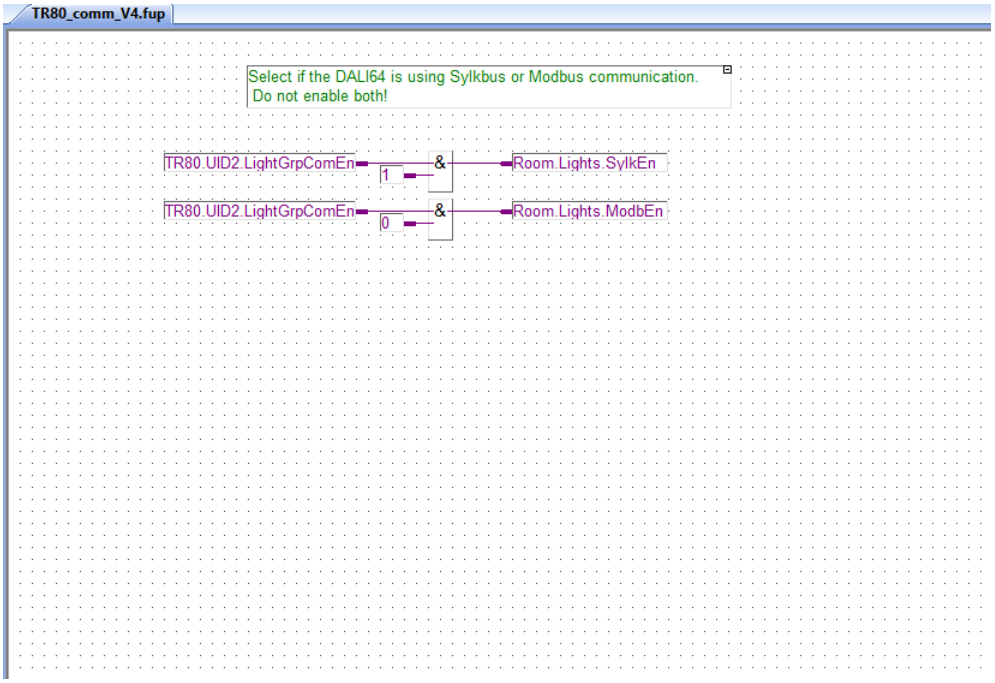
The room controller has Modbus RTU (RS-485) port which is used already to communicate with the wall unit (TR80), the Modbus/DALI64 device is connected to this port as Slave.
Communication settings: Port 0, 19200 baud, even parity, 1 stop bit, address:1

The room controller has a dedicated Sylkbus port the Sylkbus/DALI64 device is connected to this port.
Communication settings: Sylkbus port, address:1.

Each DALI64 sensor has a basic DALI configuration with 4 light dimmable groups and each group has 4 scenes defined. This configuration just fits to the features of the TR80. The DALI64 configuration can be done with an Android device using Light Touch application. The DALI64 configuration can be done individually, it is not delivered with this project.

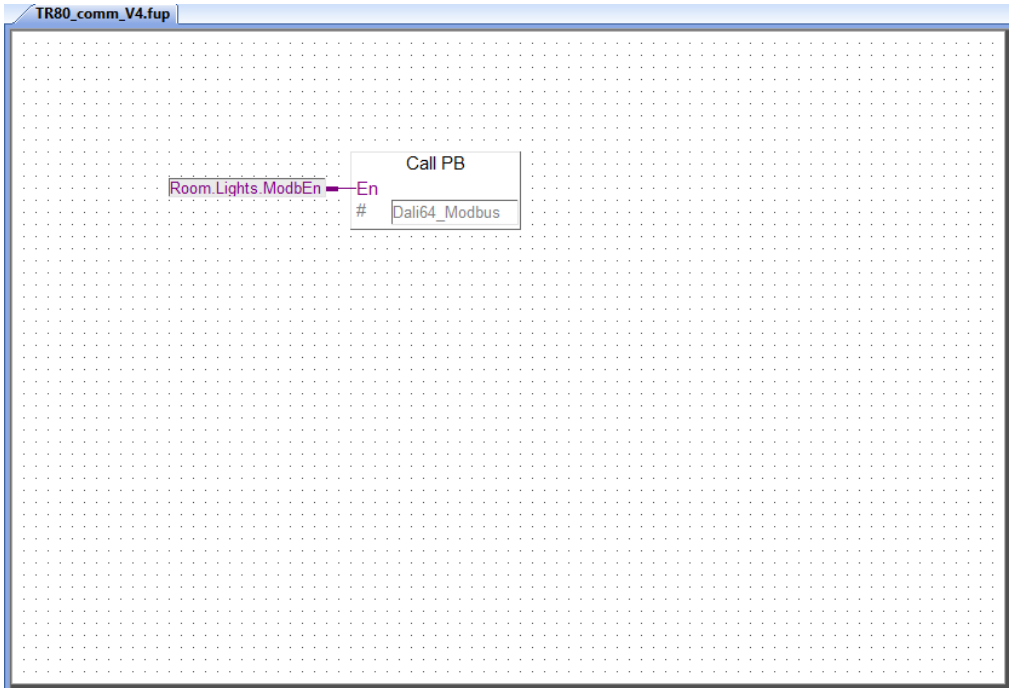
DALI64 Sylkbus or Modbus configuration

The page enables the Modbus or the Sylkbus communication to the DALI64 sensor. Constant is used for enabling, if other communication shall be used the program must be change and be downloaded. Do not use both communication at the same time. Some variables are mutually used and getting overwritten if both (Sylk and Modbus) are enabled.



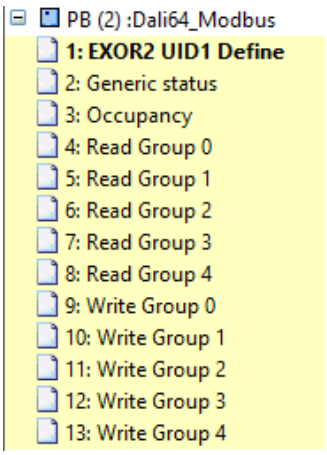
Call PB Dali64_Modbus

The page calls the Dali64_Modbus program block.



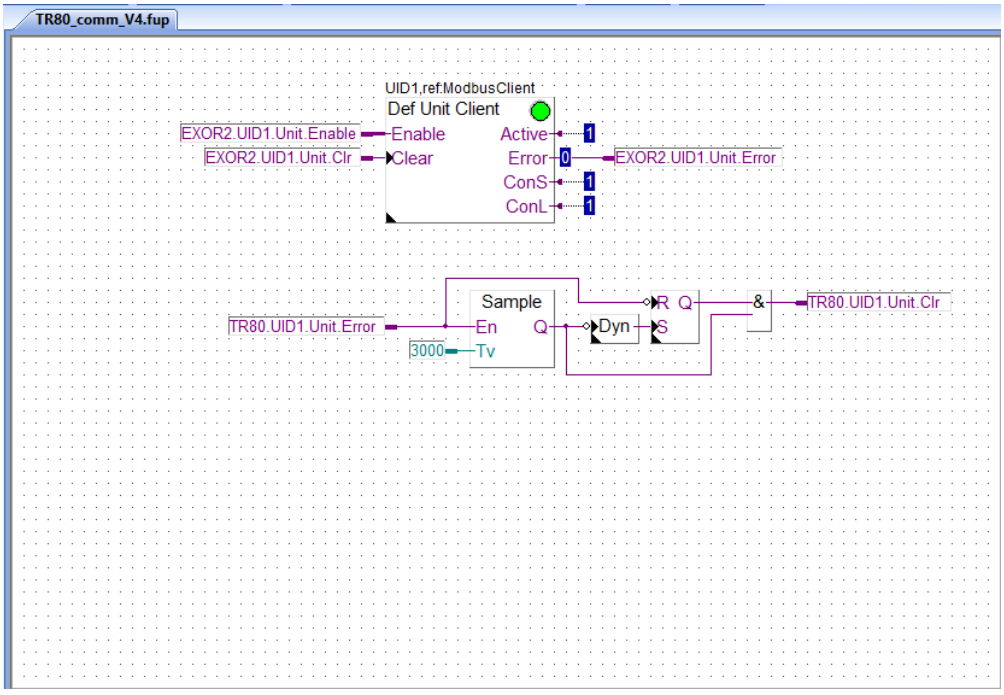
Dali64_Modbus program block

The code of the DALI 64 over Modbus RTU is a very basic program where occupancy, light intensity and calling scenes are controlled. Please check the "**Ex-Or_DALI64_A4Infosheet_v1.pdf**" for Modbus/DALI64 sensor details.



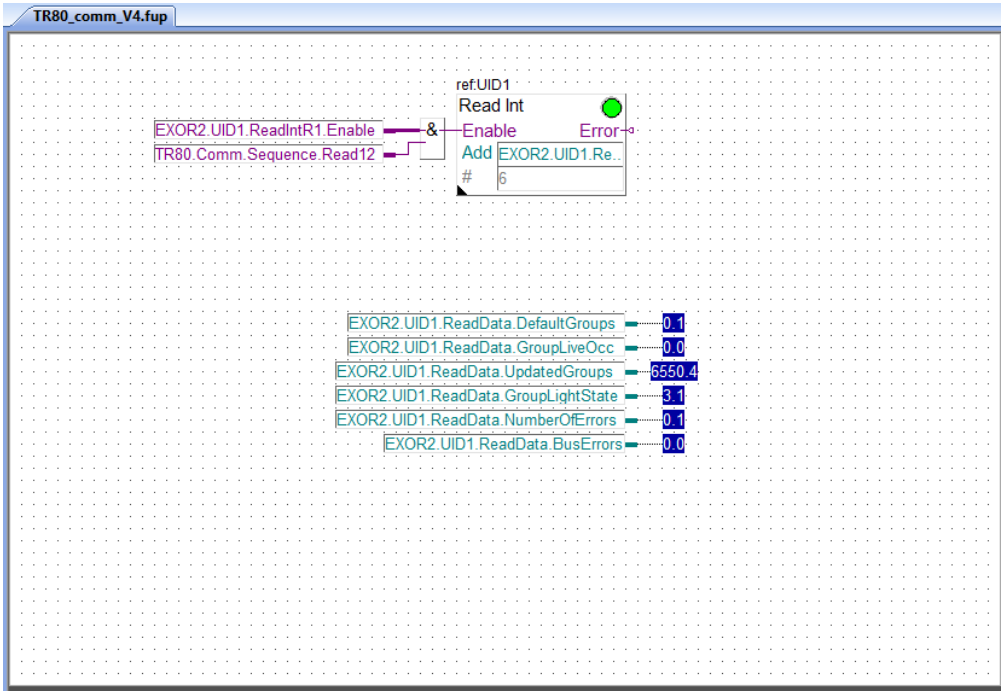
EXOR2 UID1 Define

The page defines the Modbus client for Modbus/DALI64 sensor.
If communication error happens 5 minutes later, it will be automatically deleted.



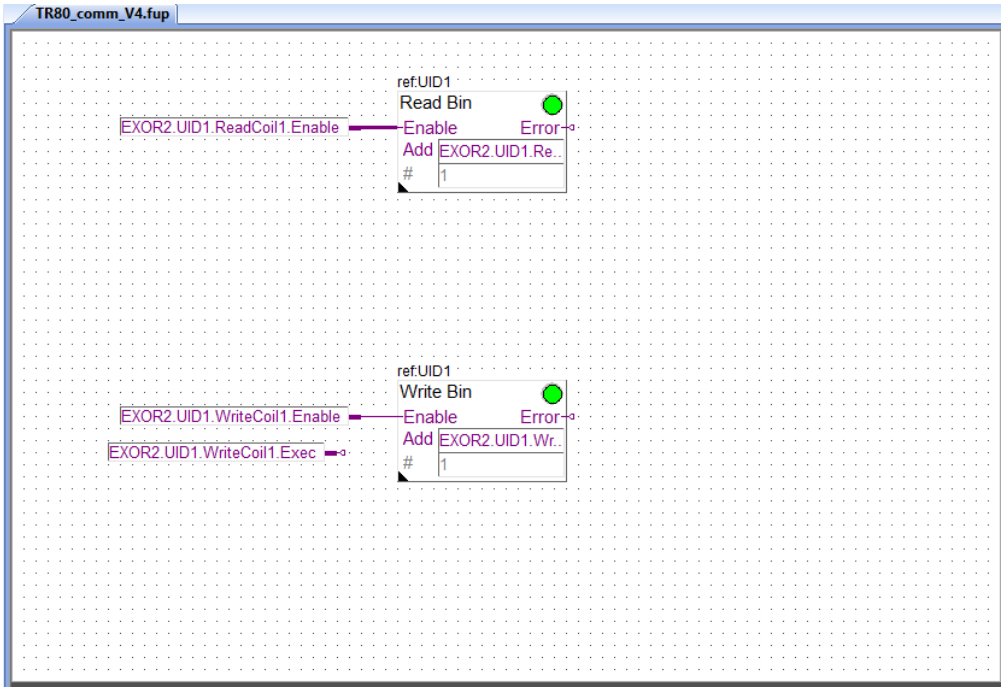
Generic status

The page reads some status information out of the Modbus/DALI64 sensor.



Occupancy

The page reads out the occupancy information out of the Modbus/DALI64 sensor. This occupancy is used on the page 9, called “Occupancy from PIR sensor”.
It is possible to write back occupancy information to the sensor, but it is not so programmed.

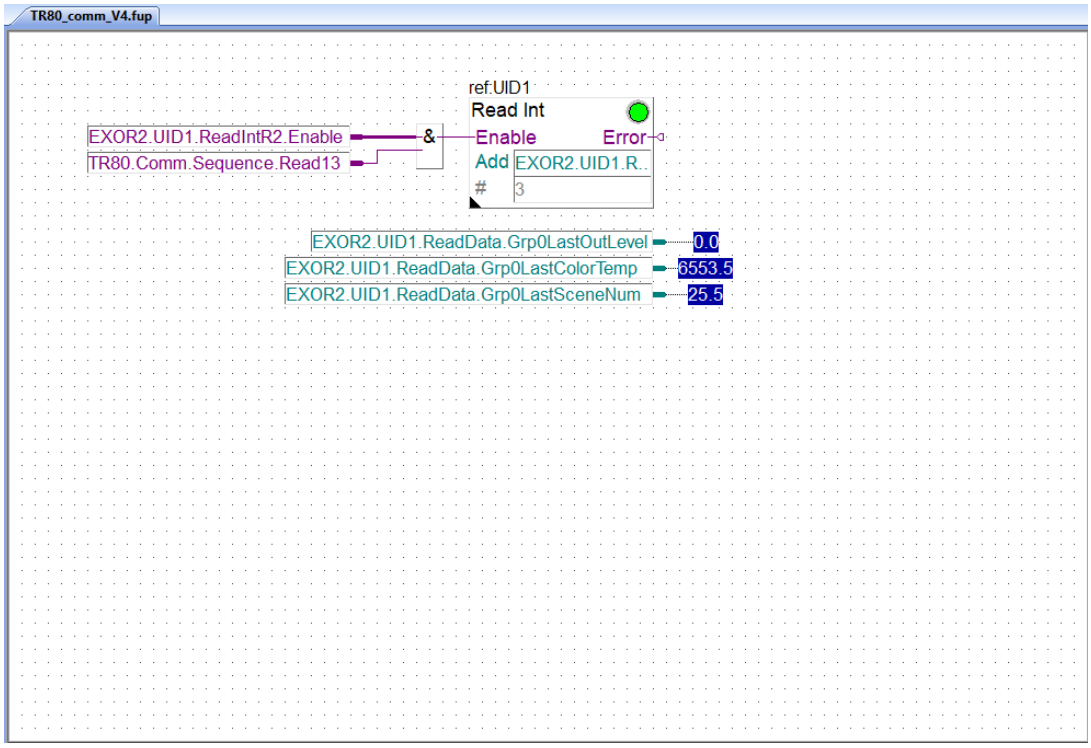


Light Group control

In the TR80 it is possible to control 4 light groups (1..4), and for each group it is possible to send a scene number.
In the DALI64SYLKPSUF PIR sensor there are 16 groups (0..15) and for each group many scenes can be defined.
The Group 0 is normally the default group to where all lights belong.

Read Group 0

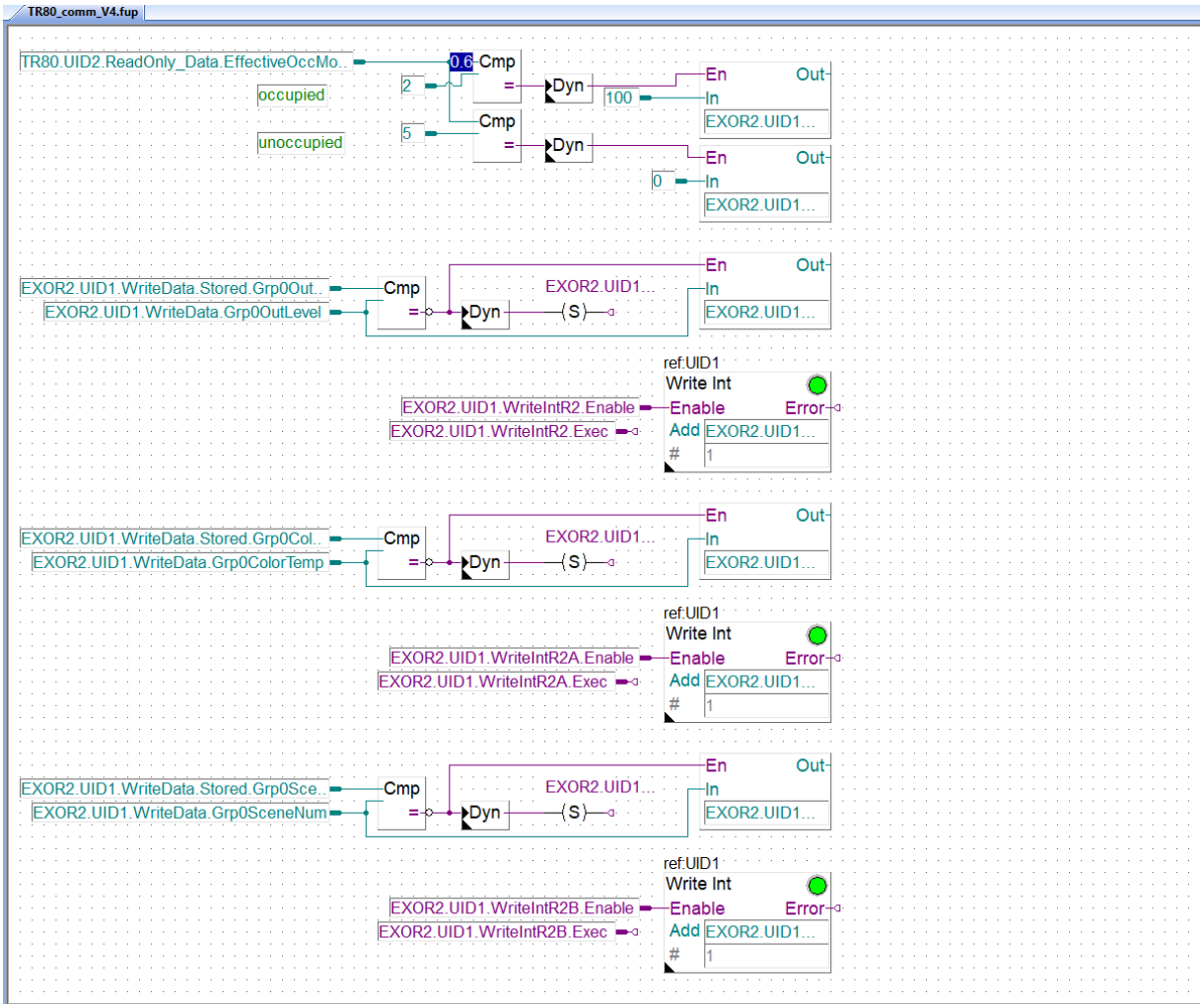
The page reads out the Group 0 last commanded property like light level, colour temperature and scene number. The property which was not commanded will get value 255 (not touched property).



Write Group 0

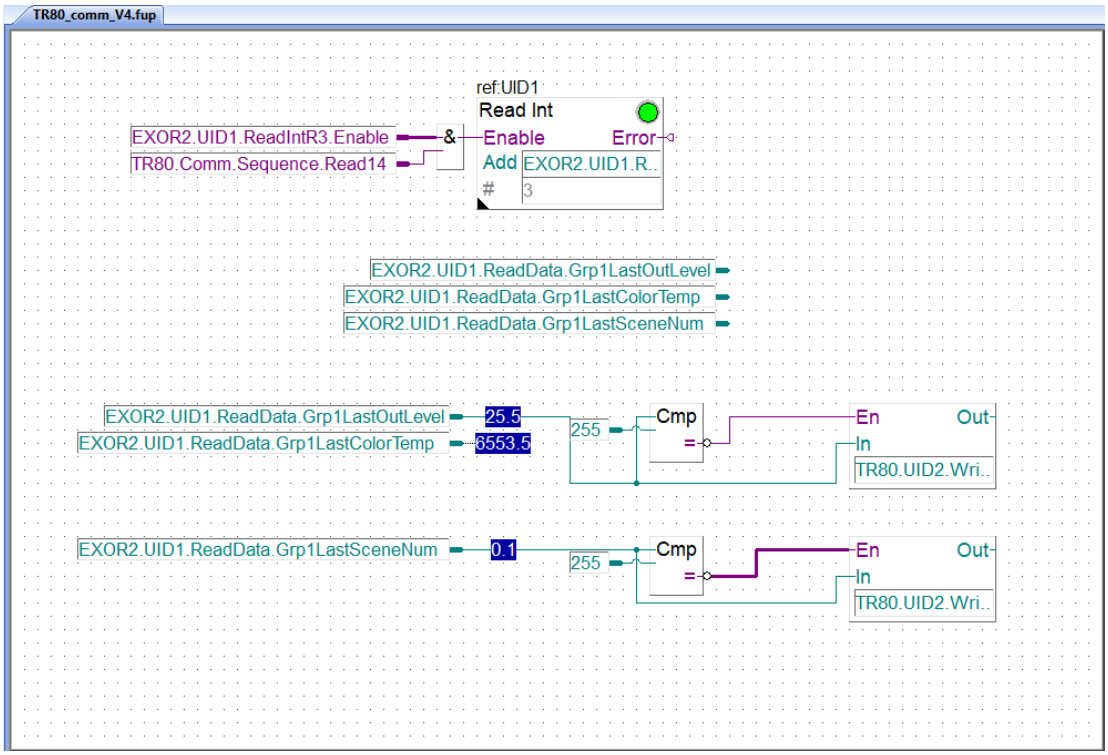
Group 0 is a global group, in the sensor all connected lights will follow the commands given to this group. This example page gives the possibility to send command to the sensor to change light level, colour temperature and scene number.
There is all light On, and Off command connected to the effective occupancy information of the room.

The Modbus/DALI64 sensor has the PIR sensor to detect motion in the room, the sensor itself can control the lighting standalone. But it is possible to send the PIR info (occupancy) to the room application, and the occupancy can be combined, with the information of external occupancy scheduler, and the User Occupancy Override from the TR80 wall unit. For good lighting control, fist proper lighting concept must be designed, then the PIR sensor settings must be adjusted properly, then the program must be adapted to create the necessary logic.



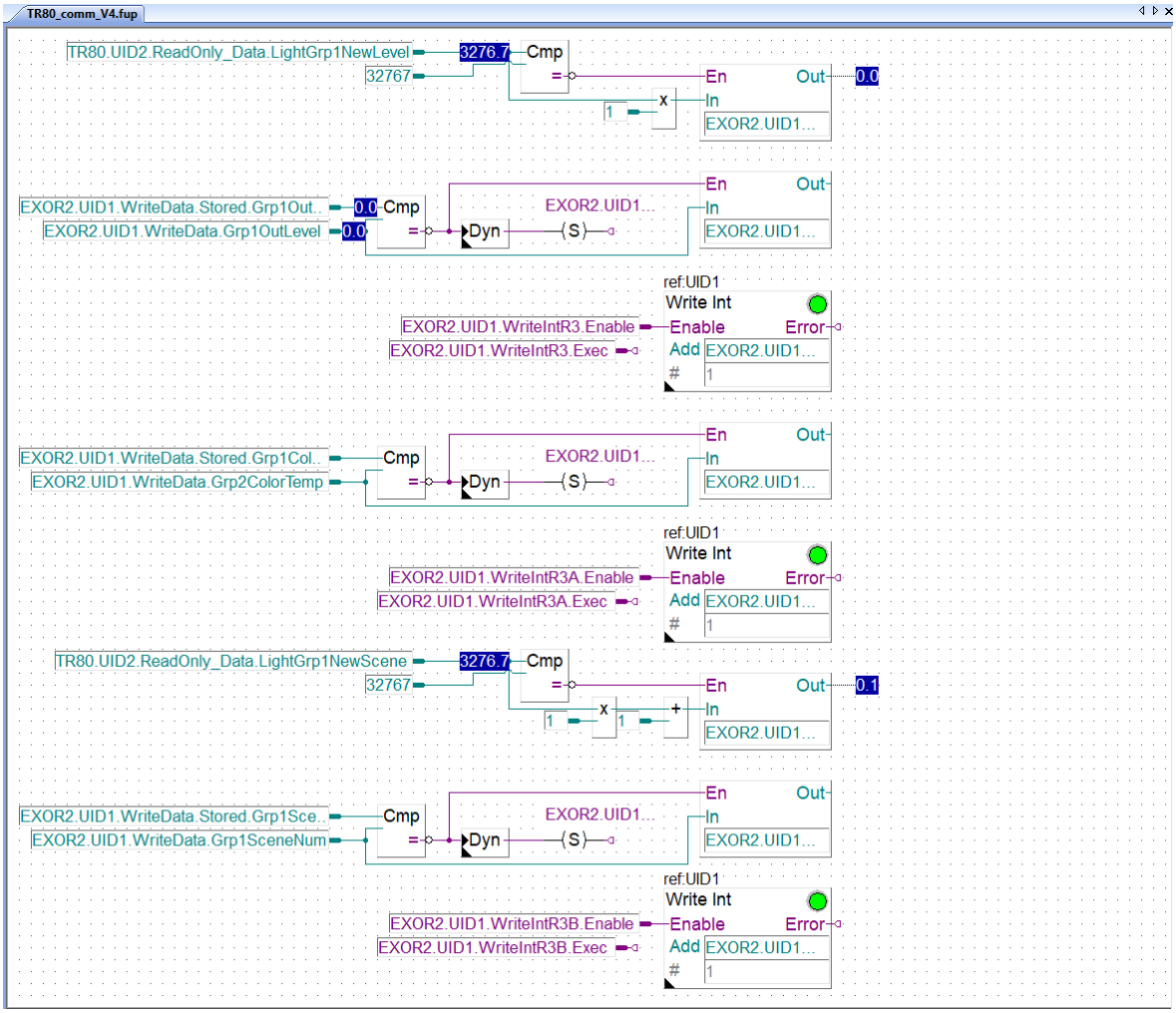
Read Group 1

The page reads out the Group 1 last commanded property light level or scene number. The property which was not commanded will get value 255 (not touched property).
Then the feedback (if not 255) is sent to the TR80 wall unit.
The other light group readings are working in the same way.



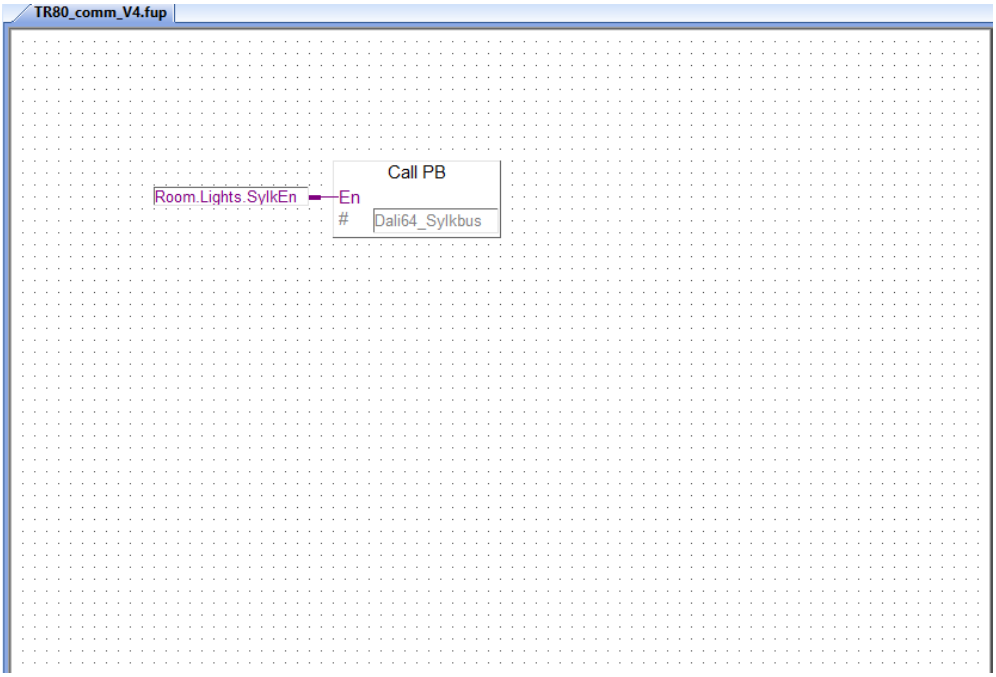
Write Group 1

This page gives the possibility to send command to the sensor to change light level in Group 1, colour temperature and scene number from the program.
And the light level, and scene number request from TR80 are forwarded to the Modbus/DALI64 sensor too.
The other light group writings are working in the same way.



Call PB Dali64_Sylkbus

The page calls the Dali64_Sylkbus program block.



DALI64_Sylkbus program block

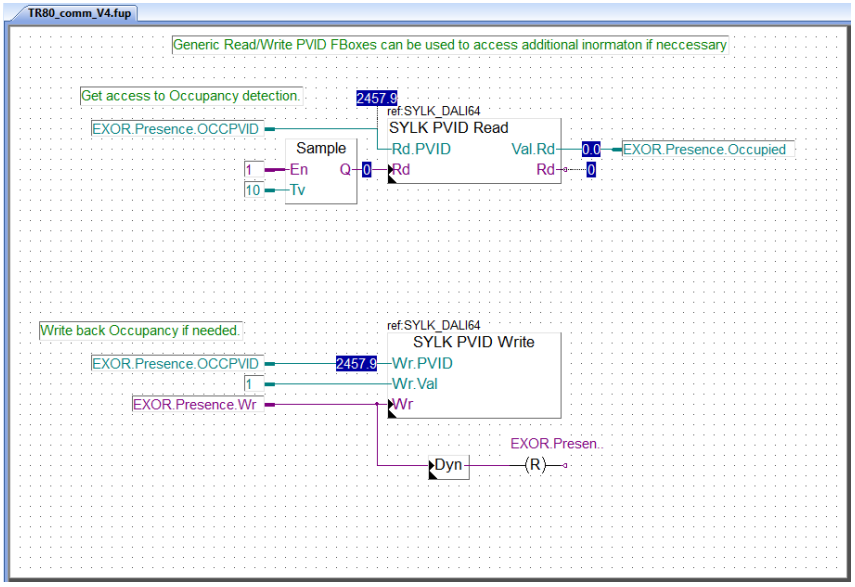
The code of the DALI 64 over Sylkbus is based on the DALI64 template project for programmable room controller PCD7.LRxx-P5. Please check the "**sbc_DALI64SYLK_application_template_ab_14072020.pptx**" and "**Ex-Or_DALI64_A4Infosheet_v1.pdf**" for details.

Occupancy detection

The occupancy is detected by the PIR sensor (depending on the settings of PIR) and it is read out by the controller over Sylkbus. However, this occupancy is not transferred to the TR80 because the occupancy source is the TR80 itself (default settings).

The occupancy can be defined by the User, by pressing the occupancy button on TR80 wall unit, this information is read out by the controller.
The occupancy can be switched off by the User on TR80, but this absence information is not transferred to the PIR sensor properly. The PIR sensor has its own occupied timer (occupancy coasting time) to switch to absence.

The SI must decide the source of the occupancy, the way occupancy works, and then he needs to modify the application program and the settings of the devices accordingly.



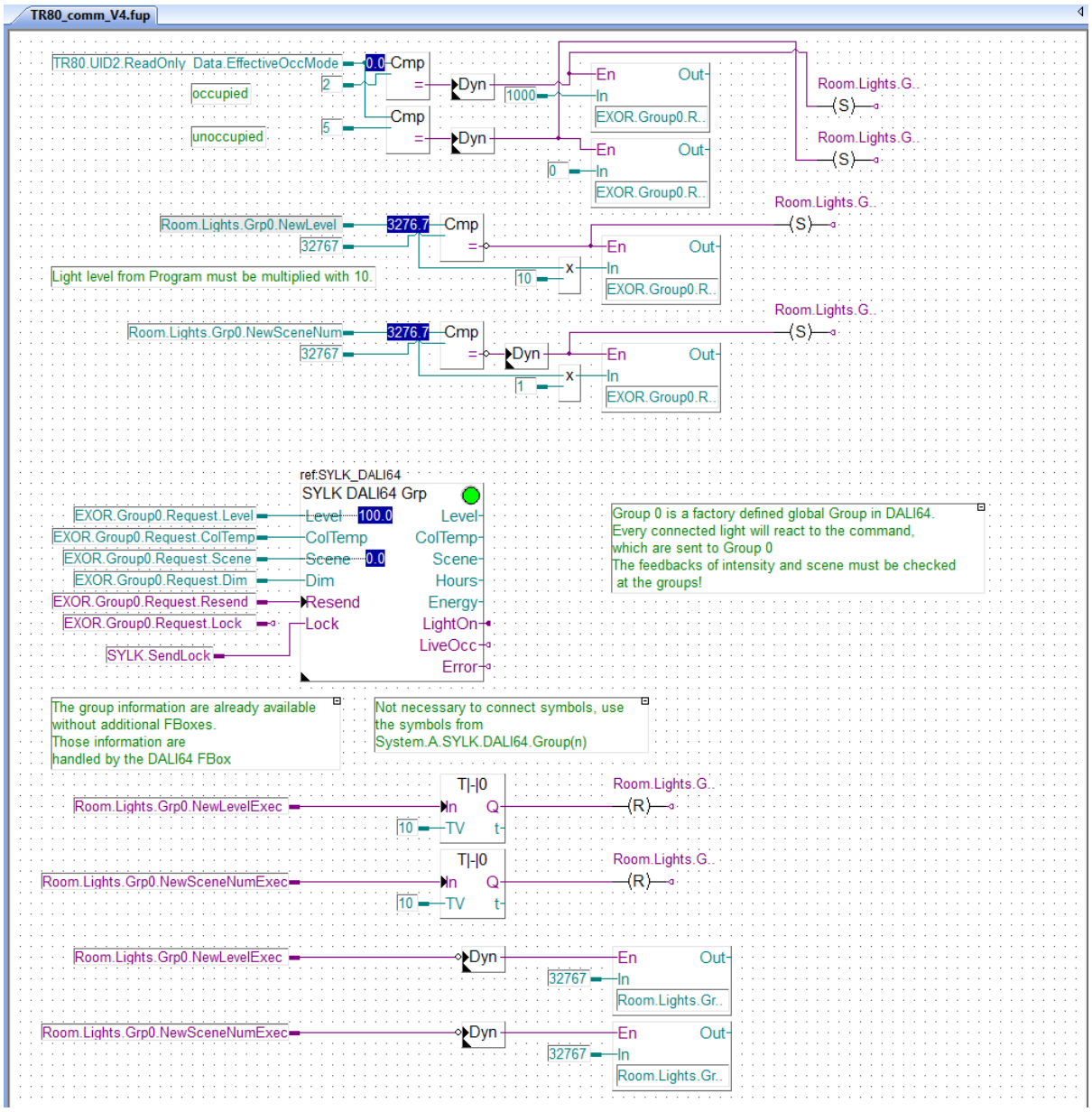
Light Group control

In the TR80 it is possible to control 4 light groups (1..4), and for each group it is possible to send a scene number.
In the DALI64SYLKPSUF PIR sensor there are 16 groups (0..15) and for each group many scenes can be defined.

The Group 0 is normally the default group to where all lights belong.

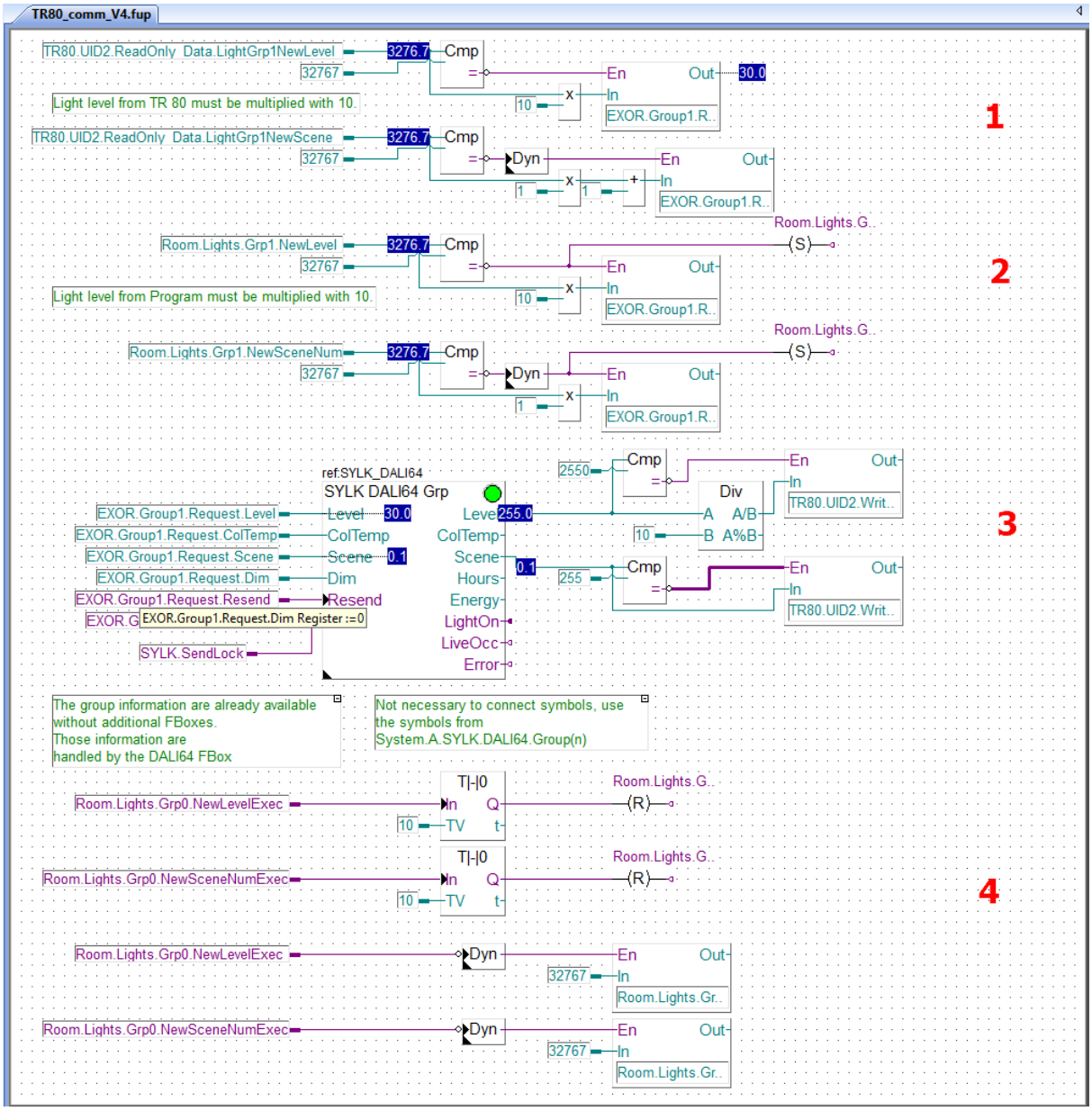
DALI64 Light Group 0

Light Group 0 in the DALI64 sensor is the global group, commanding Group 0 influences all lights connected to the sensor. The TR80 does not have dedicated global command, may be the power button or occupancy detection can be used to send global light command to DALI64.
As an example, here the Occupancy detection sends all lights on command and all lights off command. It is also possible to send global light intensity level and scene command from the program.



DALI64 Light Group 1

The following FUPLA page shows the control of the Group 1 in the PIR sensor.
All other light groups are controlled in the same way.



1. Triggered copy of the TR80 light group 1 “new light level” and “new scene number” to registers to be sent to the PIR sensor. The light level the TR 80 must be multiplied with ten before sending to PIR sensor.
2. Triggered copy of the controller light group 0 “new light level” and “new scene number” to register to be sent to PIR sensor. This gives the possibility to send light level and scene number from the program not only form the TR80 wall unit.
3. The function box is sending immediately the “new light level” and “new scene number” (based on COV) to the PIR sensor. The last value will win, either value from TR80 or from controller. And it is cyclically reading back the “last light level” and “last scene number” values from the PIR sensor to registers which will be transferred to TR80 as feedback values of the lights in this group.
4. When the controller is successfully reading meaningful value(s) (0 – 100%) from TR80 after the TR80 is giving back a dummy value (32767 decimal). This function is implemented in the TR80 to

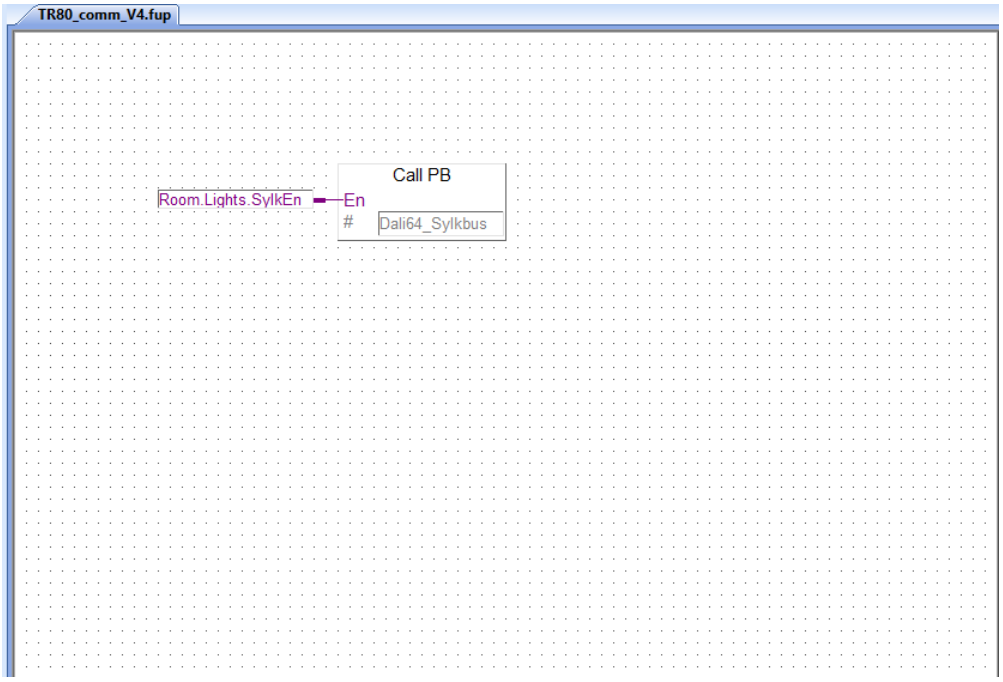
safely recognise a change of meaningful value. This function is also implemented here for the values coming from the controller itself.

Blind 1 Control

The TR80 can command two blind groups, providing angle and position command to them. Two program blocks are created to control the blind. The first one is made in FUPLA to give an overview how the blind is working and hand over the necessary parameters to the second program block. The second program block is the real control of the blind, written in instruction list, and it is called from the first one. It was necessary to do so, as there is no such Blind control Fbox which can handle position and angle control in our libraries. New version Blind control Fbox with angle and position control is under development to replace the IL code.

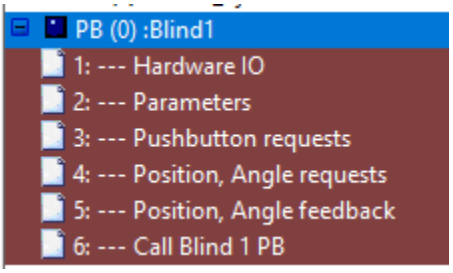
Call PB Blind1

The page calls the Blind1 program block.



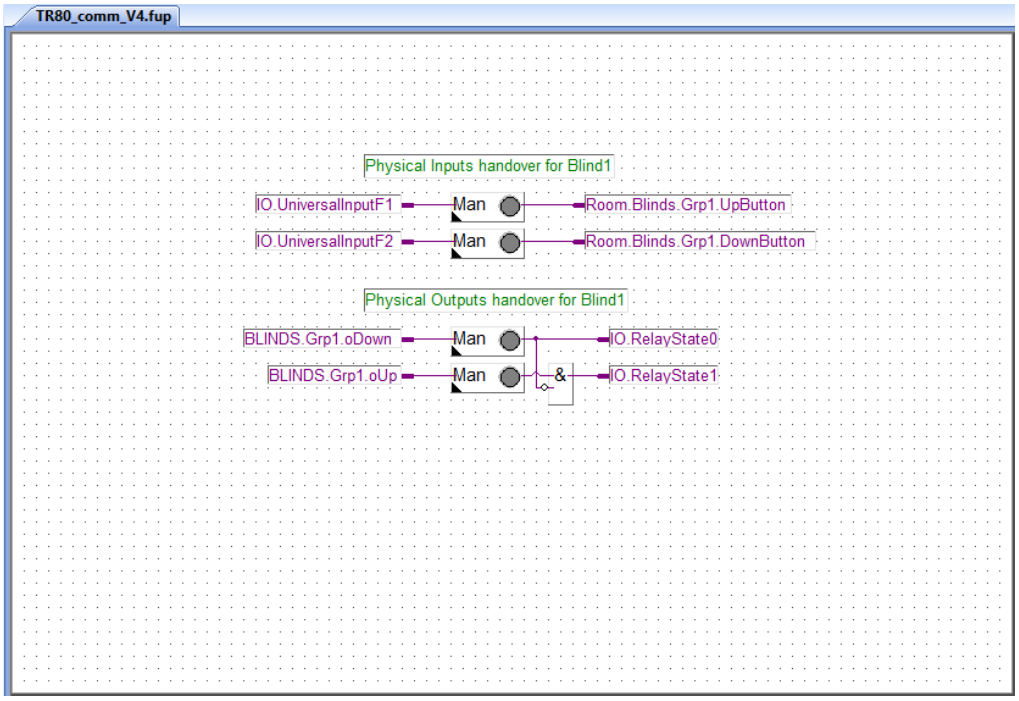
Blind1 program block

From TR80, it is possible to send a **position** and **angle** requests to the Blind 1 program block.
From the program it is possible to send **short and long push** request, **stop** request, **synchronisation** request to the Blind 1 program block.
It is possible to **lock the inputs** and command the blind upmost position and block the inputs.
The Blind 1 program block receives and hands over the request and forwards back the feedbacks of the blind 1 to the main program and to the wall unit.
This program block is then cyclically called from FUPLA.



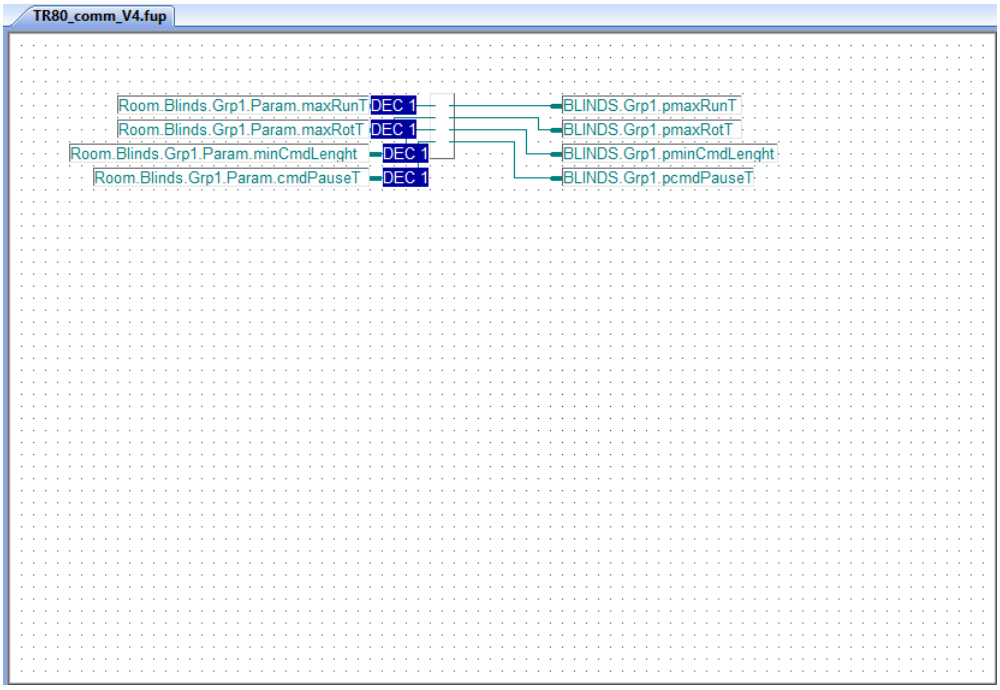
Hardware IO

This page hands over the up and down pushbutton requests and the up and down relay command to the physical outputs.



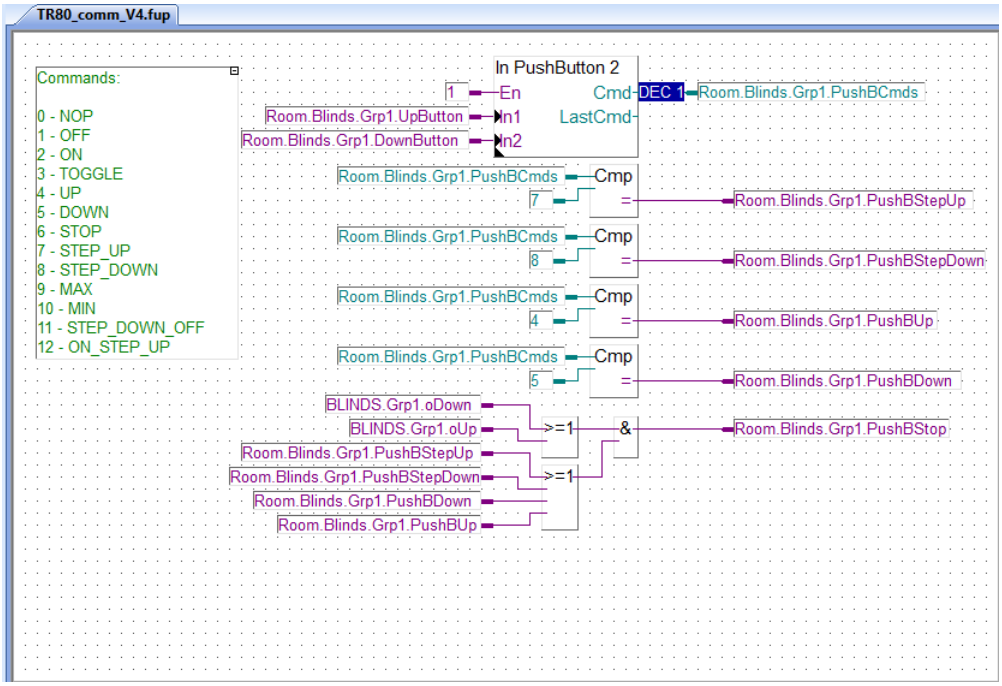
Parameters

This page hands over the parameters (like maximum running time, maximum rotation time, minimum command length, pause time between two commands) to the PB 79- (Blind 1 control).



Pushbutton requests

This page us generating the real requests to the PB 79- (Blind 1 control) out of the up and down pushbutton actions and the time duration while the button is pushed.



Position, Angle requests

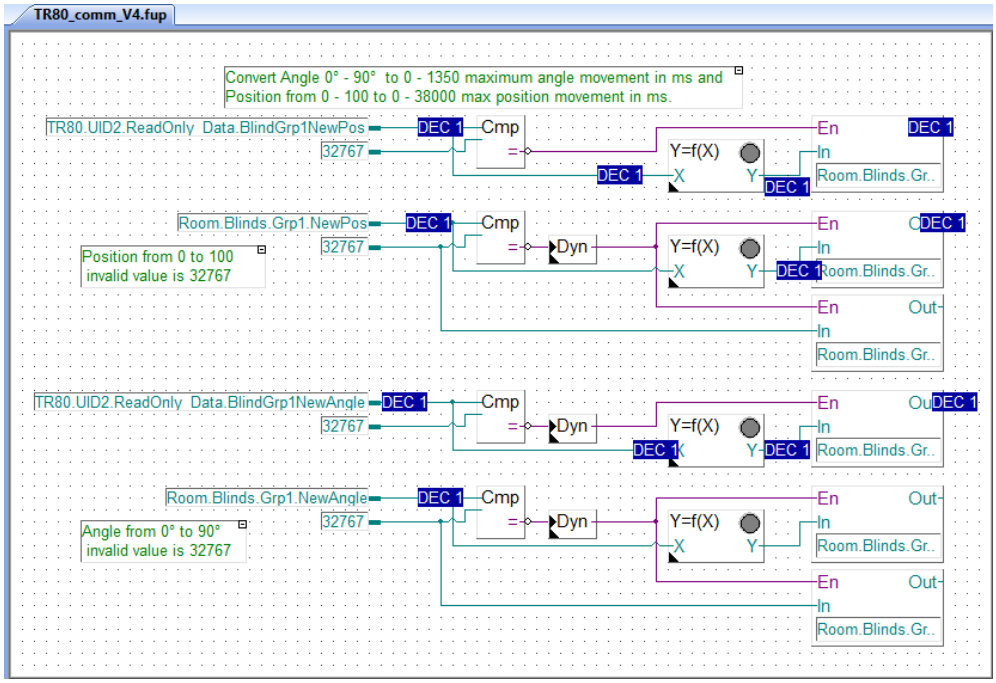
This page is receiving the position and angle request from the TR80 , converting them by using the timing parameters and sending them to the PB 79- (Blind 1 control).

It is also possible to forward position and angle request from the program itself to the PB 79 (Blind 1 control).

The physical Blind, which is used in the application is changing its angle within 1.35 second and it moves completely down in additional 38 seconds. Other Blind surely have different timings.

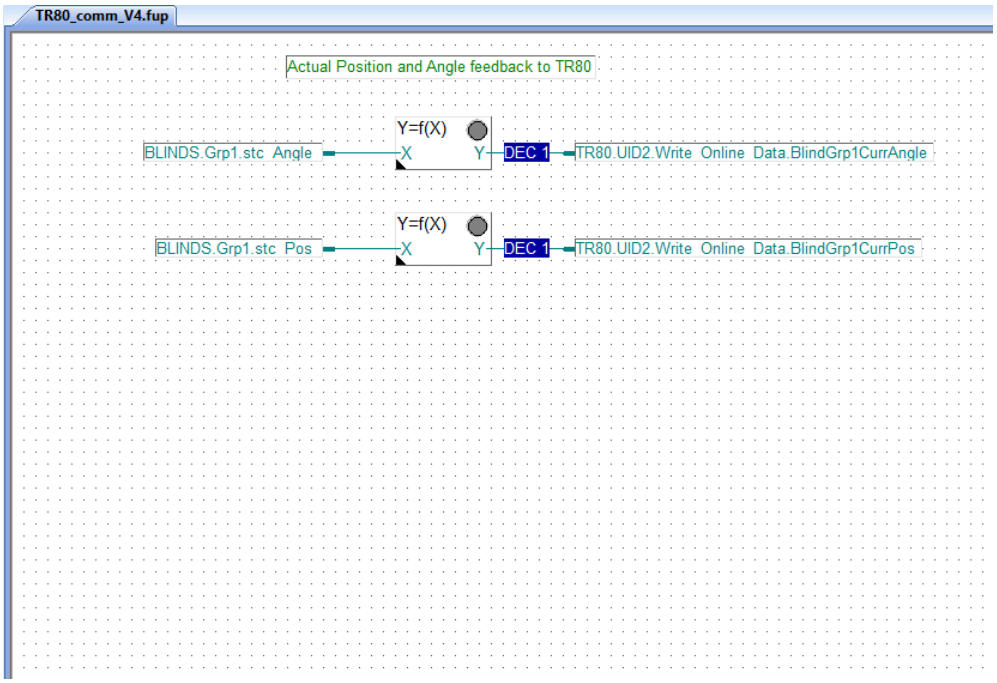
The TR80 is providing Angle position from 0° to 90°, and position from 0 to 100%. This must be converted to time in millisecond based, because the blind control program is creating a virtual position and angle feedback, based on counting the time while the Up and Down output is high.

The accuracy of angle and position calculation is depending on the cycle time of the program, lower cycle time will result better accuracy.



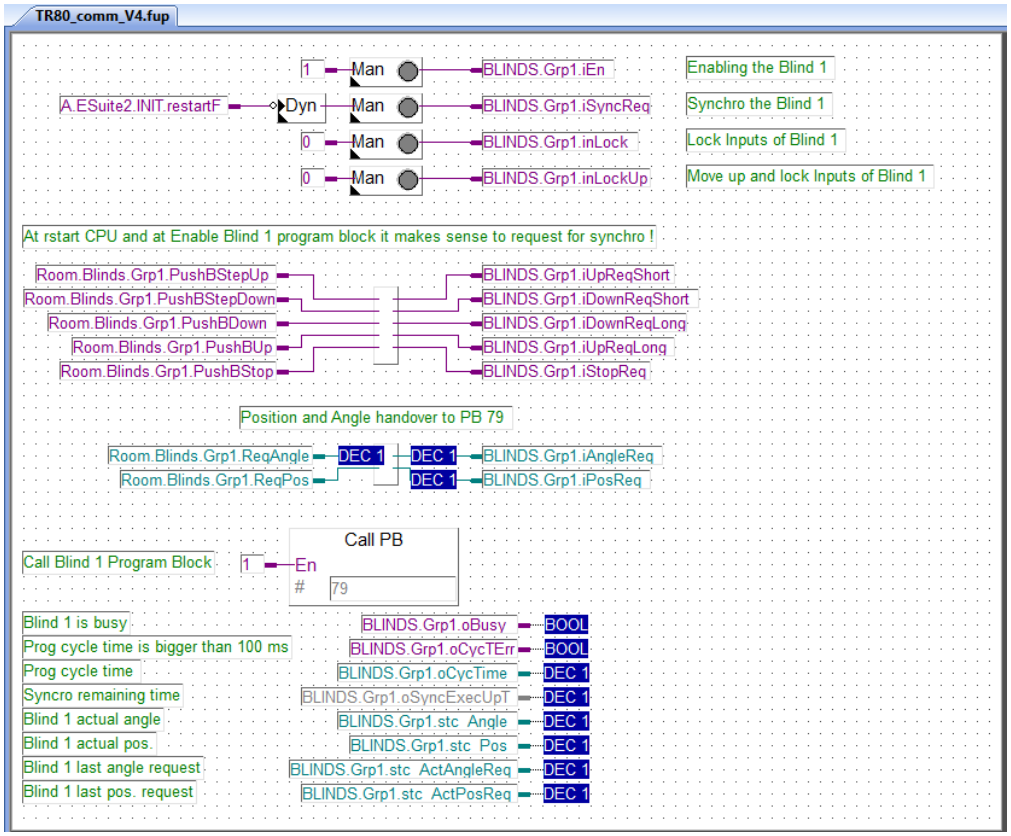
Position, Angle feedback

This page is receiving the position and angle feedback from the PB 79 (Blind 1 control) converting them and forwarding them to the TR80.



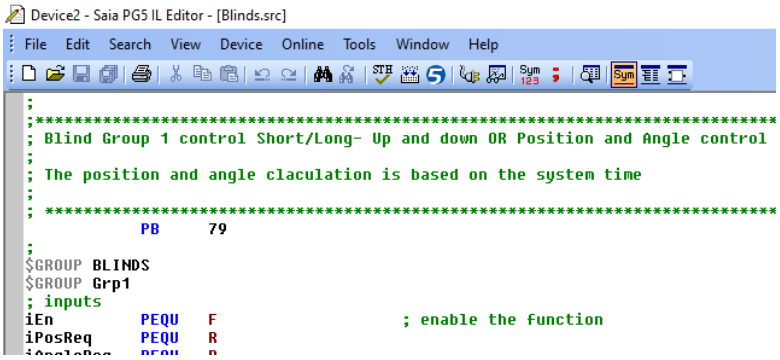
Call Blind 1 PB

This page is a summary page where all the signals are visible which must be transferred to and from the PB 79 (Blind 1 control).
This page is calling the PB 79 (Blind 1 control).
It is possible to enable the program block, synchronise the blind, lock it or lock it in upmost position.



Blinds.src: PB 79- (Blind 1 control)

This program block is the core of controlling the Blind 1 by position and angle. It is temporary solution of until having a dedicated FBOX. It has been tested, but not guaranteed, that it works always correct.
It is possible, that in the future SBC provides a proper Fbox to realize the function.
We do not provide here description of the program, but only describe the interface.



Inputs:

iEn	F	Enabling the program block
iPosReq	R	Position request input, has priority over angle
iAngleReq	R	Angle request input
iUpReqShort	F	Short push up request -> short move up (Tpulse= pminCmdLenght)
iDownReqShort	F	Short push down request -> short move up (Tpulse= pminCmdLenght)
iUpReqLong	F	Long push up request -> set position to “0” → moves up
iDownReqLong	F	Long push down request -> set position to “pmaxRunT” → moves down
iStopReq	F	Stop request → stops ongoing synchronisation, long push, position, and angle requests
iSyncReq	F	Synchronisation request → moves the blind to the upmost position and zeroing the position and angle counters
inLock	F	Lock the inputs,
inLockUp	F	Move the blinds up and Lock the inputs

Outputs:

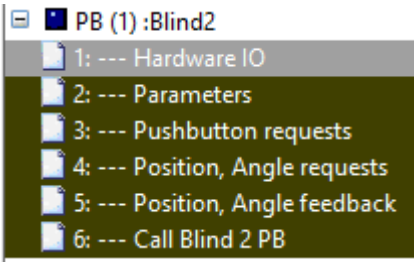
oUp	F	Up command
oDown	R	Down command
oBusy	R	If any command is being executed (except stop command)
oCycTErr	F	Cycle time error, high if the measured cycle time of the program exceeds 100ms
oCycTime	R	Cycle time of the program
oSyncExecUpT	T	Remaining time of synchronisation (during synchronisation up command is executed long enough to move the blind to upmost position).

Parameters:

pmaxRunT	R :=38000	maximum running time without Angle time (in x1 ms)
pminRunT	R :=0	minimum running time
pmaxRotT	R :=1350	maximum time for changing the Angle (in x1 ms)
pminRotT	R :=0	minimum rotation time
pminCmdLenght	R :=2	minimum length of the up/down command output at short push commands (in x100 ms)
pcmdPauseT	R :=5	pause Time between up-down commands (in x100 ms)

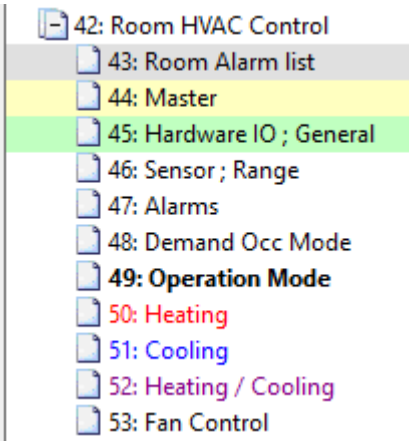
Blind 2 control

The Blind 2 control is working in the same way as Blind 1 but calling PB 78 and using different symbols.



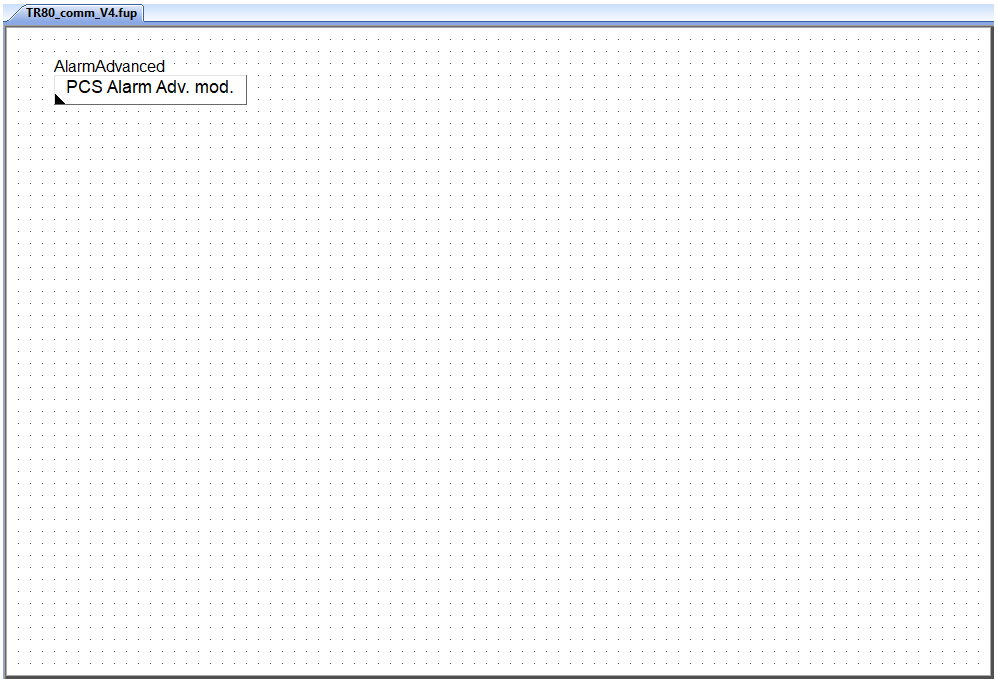
Room HVAC control

The code of the Room HVAC page was taken over from the Room template for PG5 programmable room controller PCD7.LRxx-P5 with some small modifications. Please check the "**Manual_Room_Template_V2_01.pdf**" for details.



Room Alarm list

Contains the FBoxes PCS Alarm Adv. mod. – to be used to modify a specific level in the plant coding system for alarms.

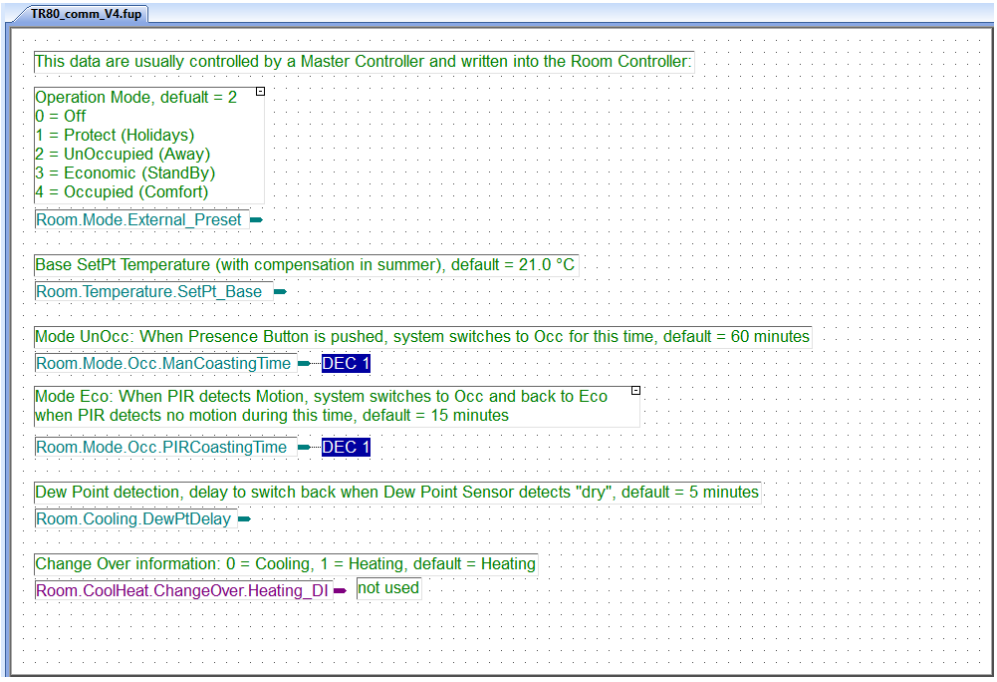


Master

This page simply shows the most important data which may be written from a Master PLC

- Operation mode e.g. UnOccupied, Eco (StandBy), Occupied
- Base Temperature Set Point – This is configured and coming from the TR80 in this example.
- Time Delay for Presence or Dew Point detection
- ChangeOver information for 2-pipe application

In many cases, a Master PLC changes the operation mode in the morning from UnOccupied to Eco (StandBy) and at the end of the day back to UnOccupied. Also, the Base Temperature, e.g. shifted in summer and the information for 2-pipe application, to inform the room application if heating or cooling is active.
Therefore, those symbols are predefined with a fix address, starting for flags and register with 300.



Hardware IO

This page is used to map physical IO to application data.

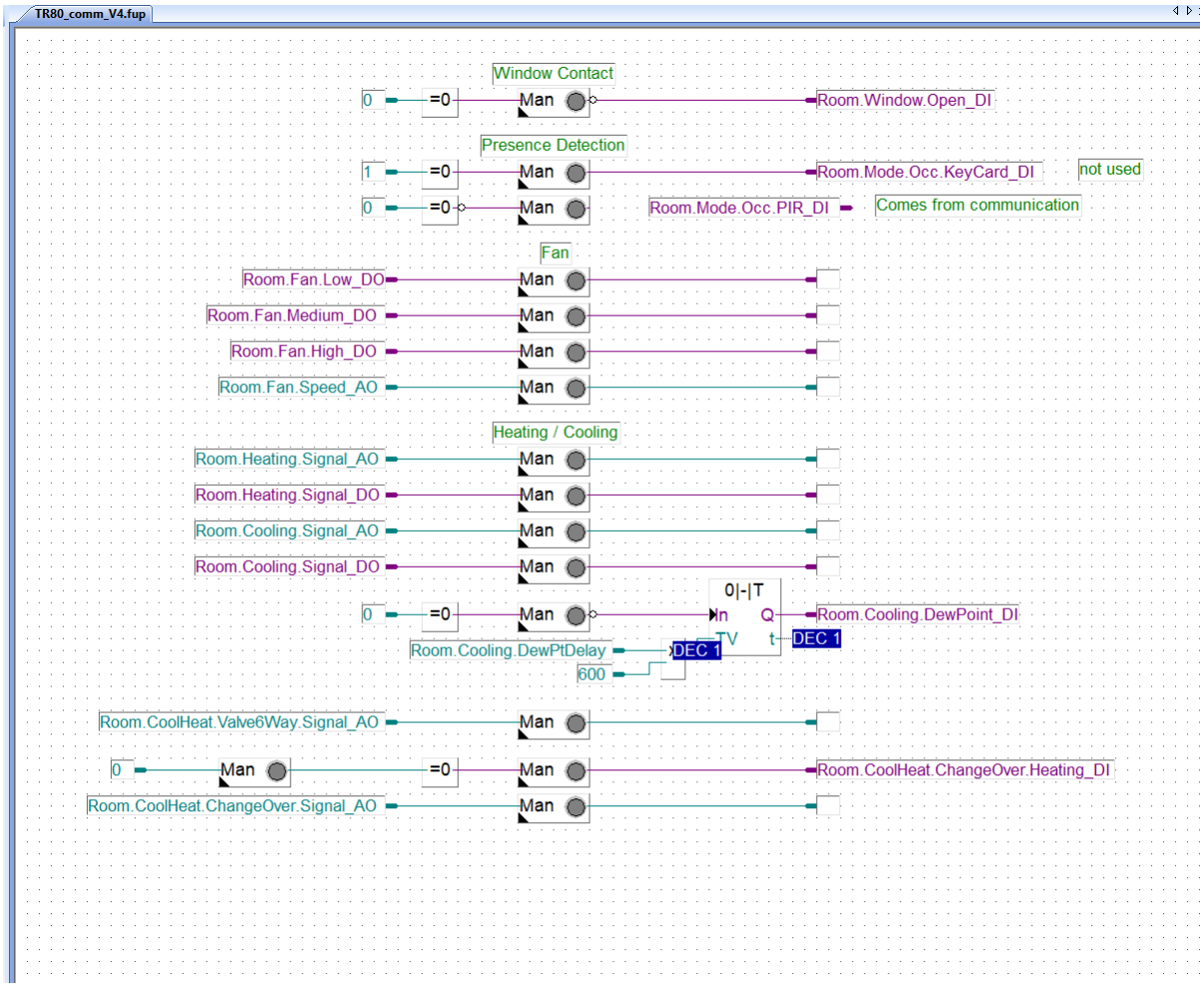
- 1xDI: Window contact
- 2xDI: Presence detection via Key Card holder or/and PIR Motion Detection
- 3xDO: Fan with up to 3 speed or/and 1xAO for frequency speed

Heating/Cooling – depending on application type and/or drive type

- 2xAO: continuous drive
- 2xDO: TRIAC/PWM drive
- 1xAO: continuous 6-way valve drive
- 1xAO: continuous drive for ChangeOver

The outputs can be used also in a mix, e.g. 1xAO for continuous drive heating and 1xDO for PWM drive cooling.

The Media Mapping IO Symbols from the PG5 Device Configurator must be connected to the required function. It is not necessary to delete unused connections. This might be helpful when a drive must be replaced in future and the drive type is changing.

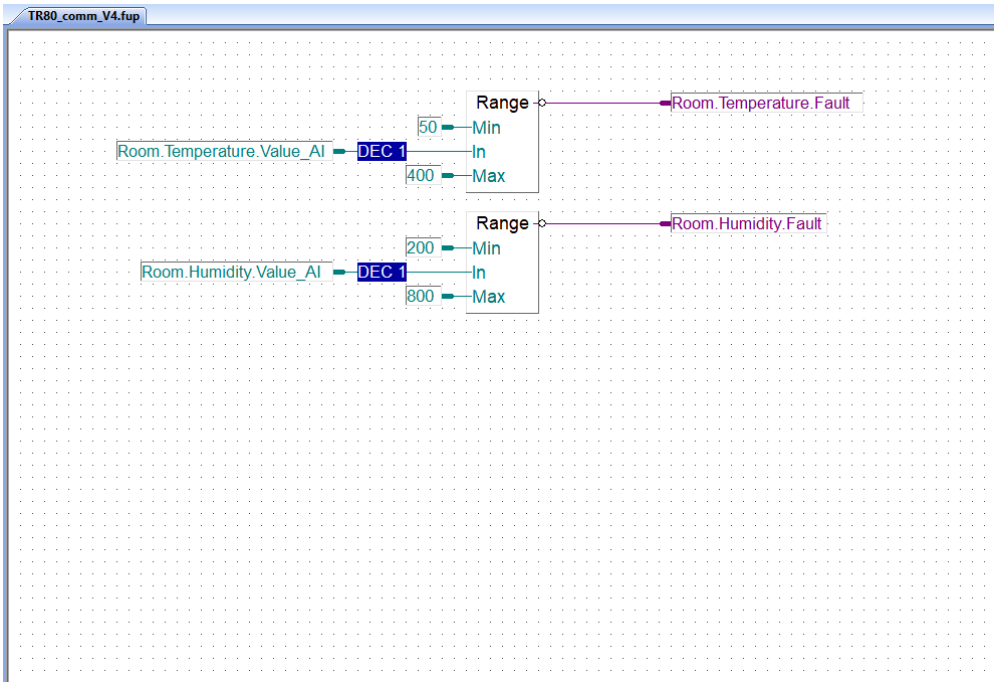


Sensor range

This page is used to detect a sensor fault

- Temperature, normal range between 5.0 and 40.0 degrees
- Rel. Humidity, normal range between 20.0 and 80.0 %

Each sensor signalizes a fault if the measured value exceeds the defined range.
The fault message has no effect in the Room Application, it is just collected in the Alarm List

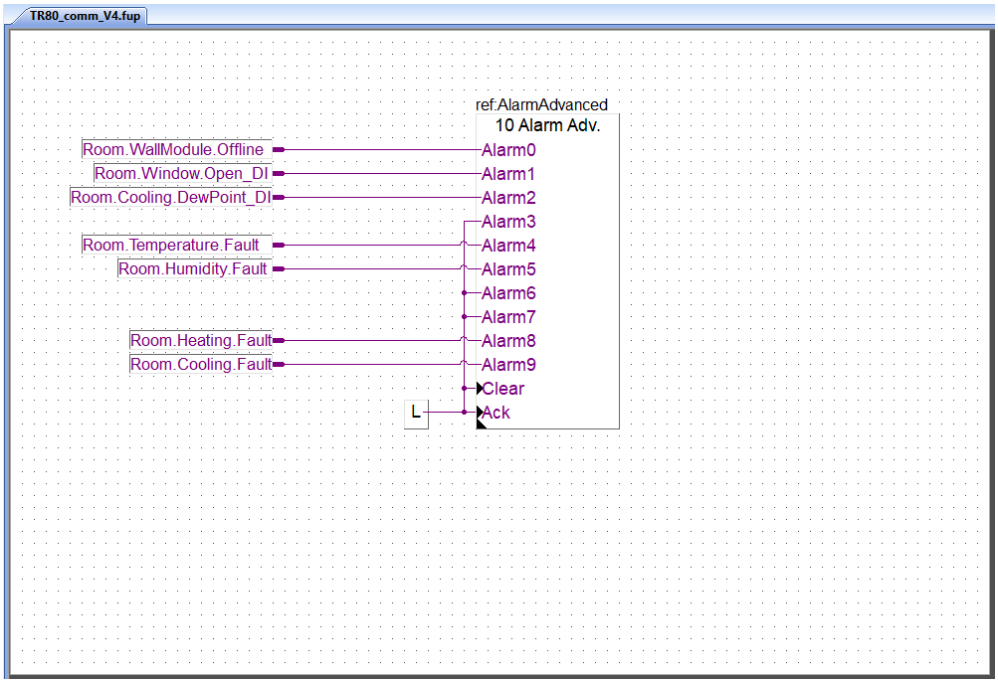


Alarms

This page is used to collect alarms and map them into the Alarm List.

The Alarm List on a E-Line C-15 or LRxx-P5 device is a distributed Alarm List, means that the alarms can be configured with Group, Priority, Plant Code System and Alarm Text.
And merge the Alarms in a Master PLC (PCD or pWeb Panel) to get a unique Alarm List in a Web Application.

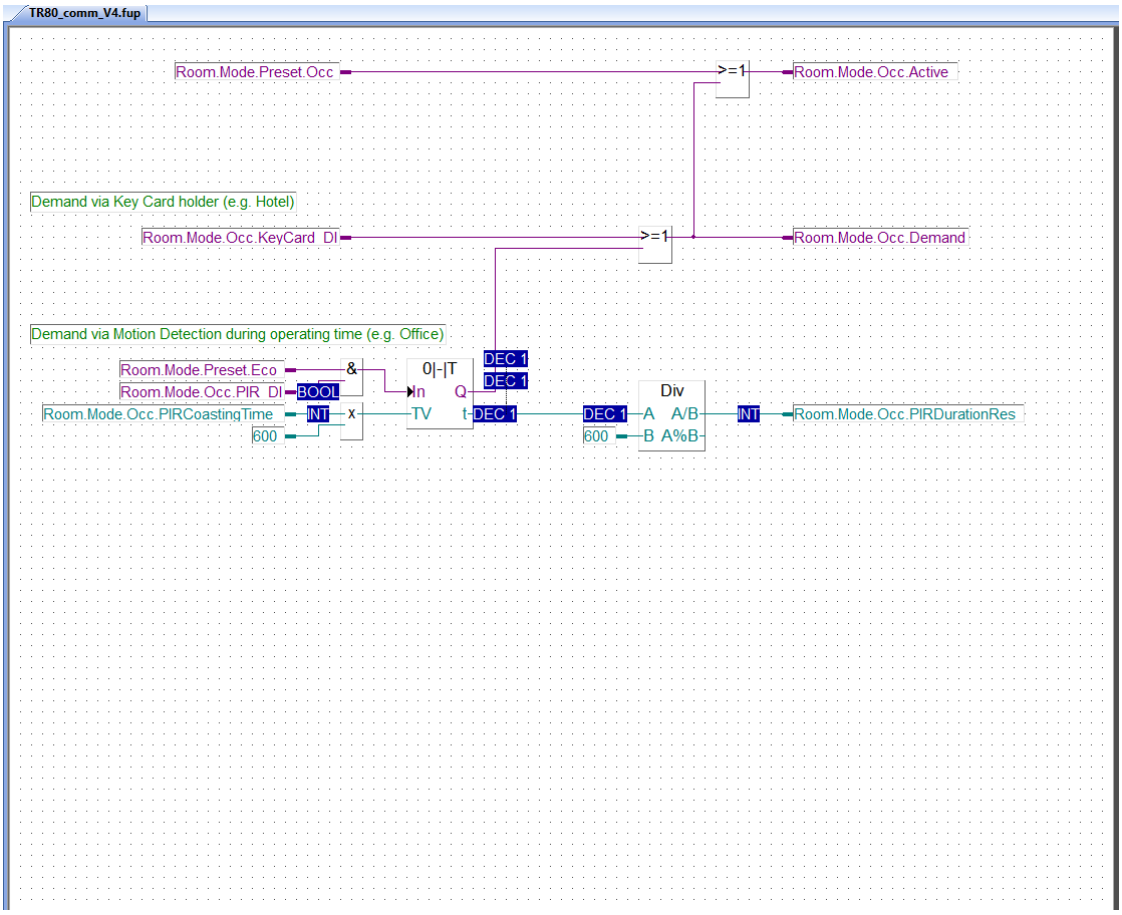
See also page “Initialisation”, FBox “Distrib.Alarm List”



Demand OCC mode

The Occupancy Mode can be required from several functionalities

- By a Supervision system or a Master PLC, e.g. in the morning to Occupied, at the evening to Unoccupied. No User Action required, controlled by a Scheduler.
- By a Key Card holder, e.g. Hotel Application
- By a Motion Detection (PIR) and Room is in Eco Mode. Occ mode is enabled as long motion is detected. If no motion is detected, mode is set back to Eco after a coasting time, default = 10 minutes

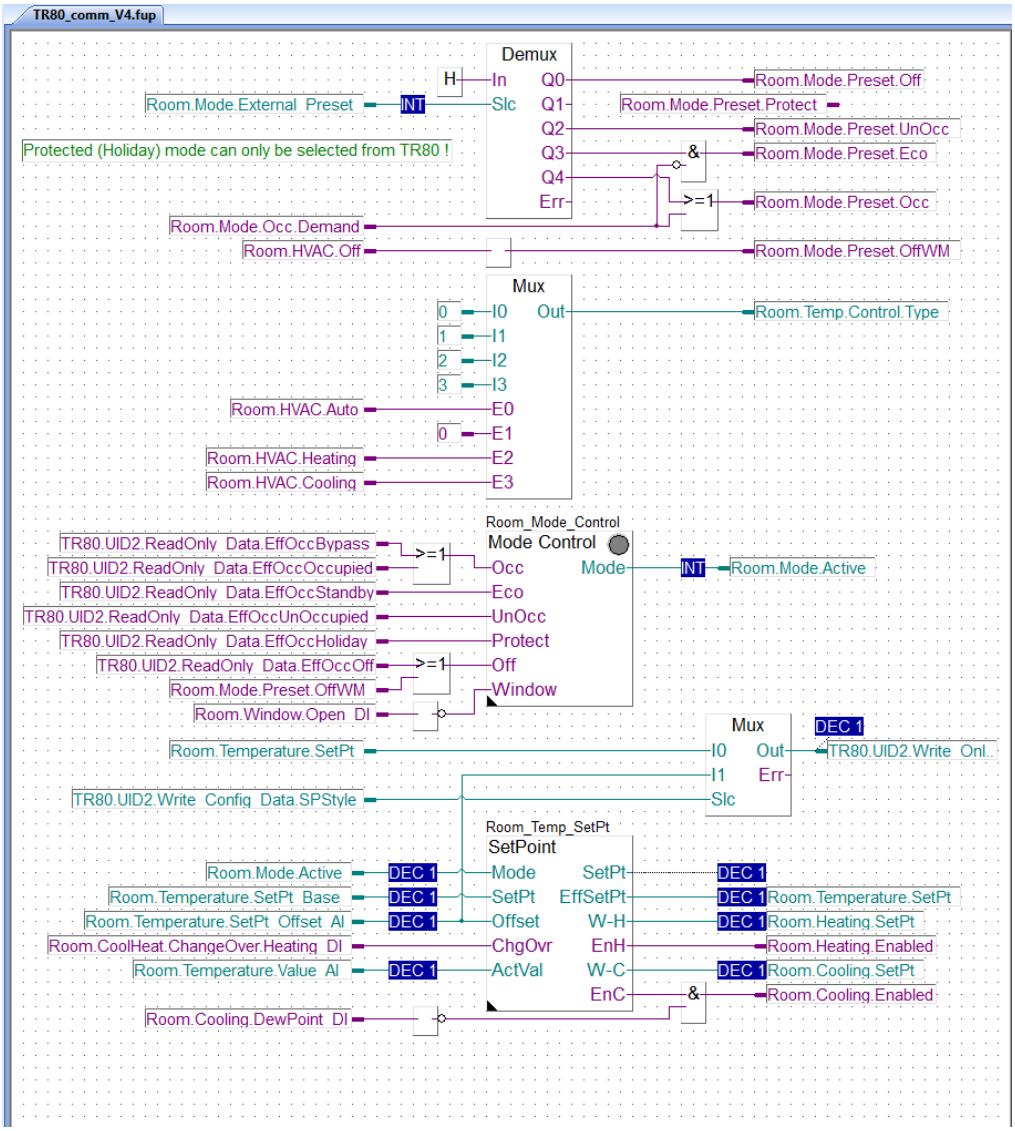


Operation mode

FBox “Mode Control” is used to define the mode if Window open detected and when Eco and Occ are activated at the same time.

FBox “Set Point” calculates the Set Point for Heating and Cooling depending on active mode. Occ, Eco and UnOcc are using different dead-band for energy saving reasons. Also the Application type can be selected.

The FBox outputs are connected to the control functionality for Heating/Cooling,



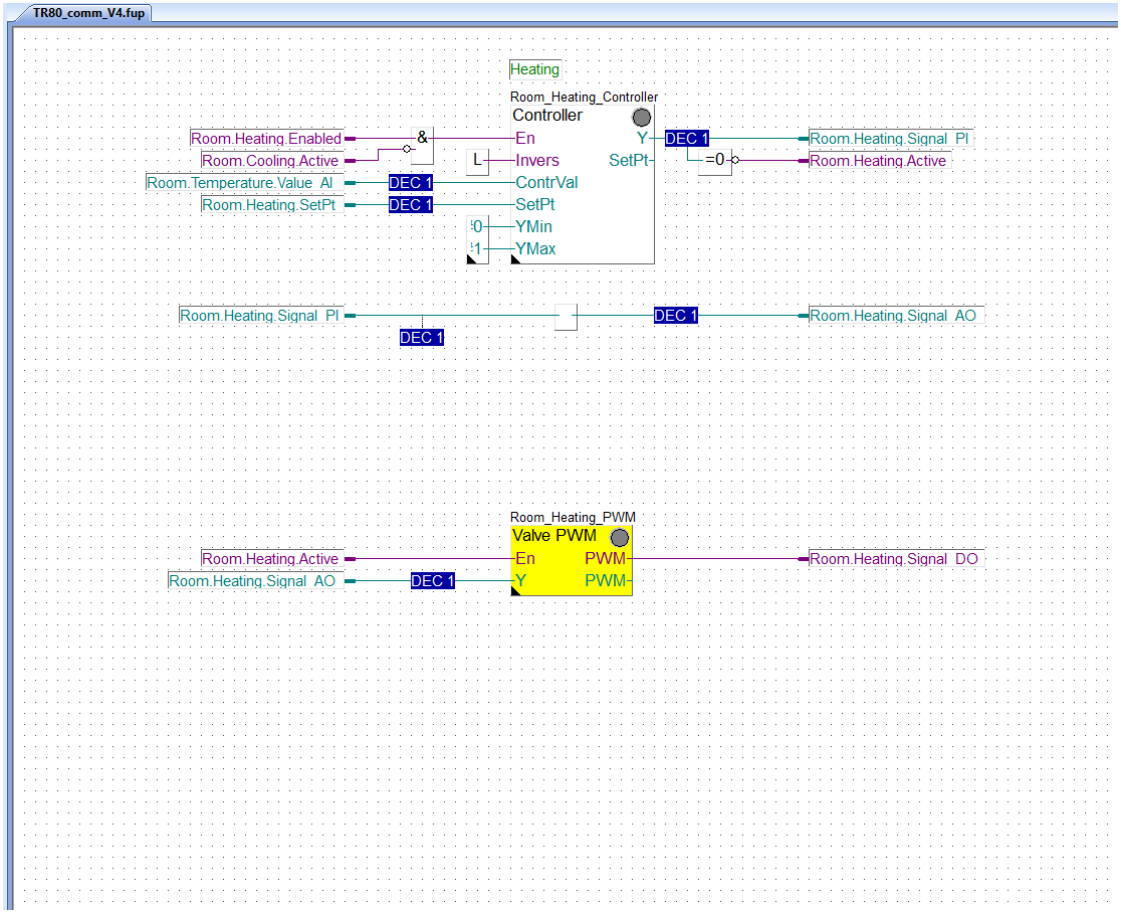
Heating

This page contains the control for Heating.

The control loop can be defined as pure P or as PI-controller.

This page prepares a 0.0 to 100.0 % and a digital signal.

Heating signal can be overruled by setting the Room Application to Slave Mode – see also page “Master”.



Cooling

This page contains the control for Cooling.

The control loop can be defined as pure P or as PI-controller.

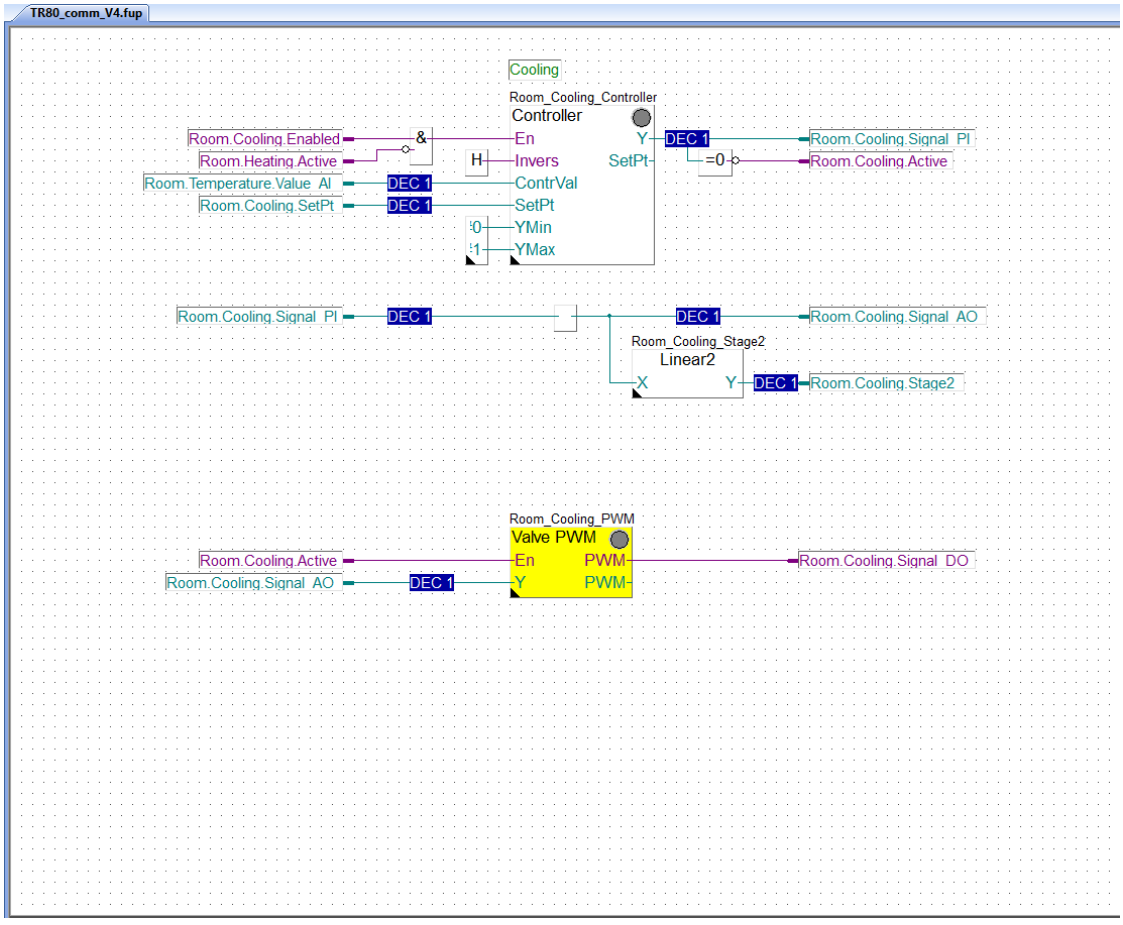
The Cooling Signal can be used to activate a 2nd stage for cooing. If the cooling signal exceeds a threshold value, a 2nd cool stage signal is calculated linear to the cooling signal.

The 2nd cool stage signal usual is used to control a damper to increase air volume.

This page prepares a 0.0 to 100.0 % and a digital signal for cooling signal.

Cooling 2nd stage signal is connected to page “Air Quality Controller” where the greater signal is selected for damper signal.

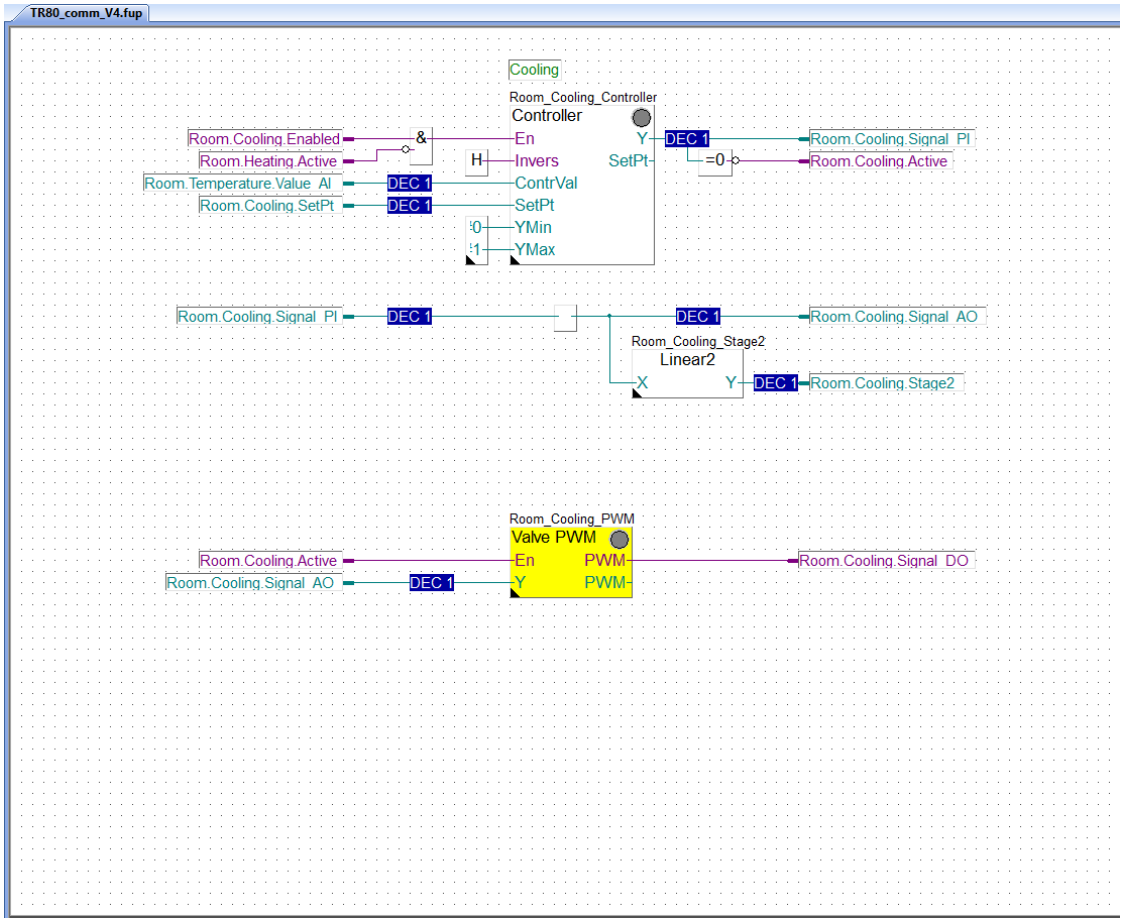
Cooling signal can be overruled by setting the Room Application to Slave Mode – see also page “Master”.



Heating / Cooling

When Heating and Cooling signal is acting on the same valve, the signals must be merged.
This page prepares 2 signals, to be used for

- 2-pipe application with Change Over
- 4-pipe application with 6-way valve



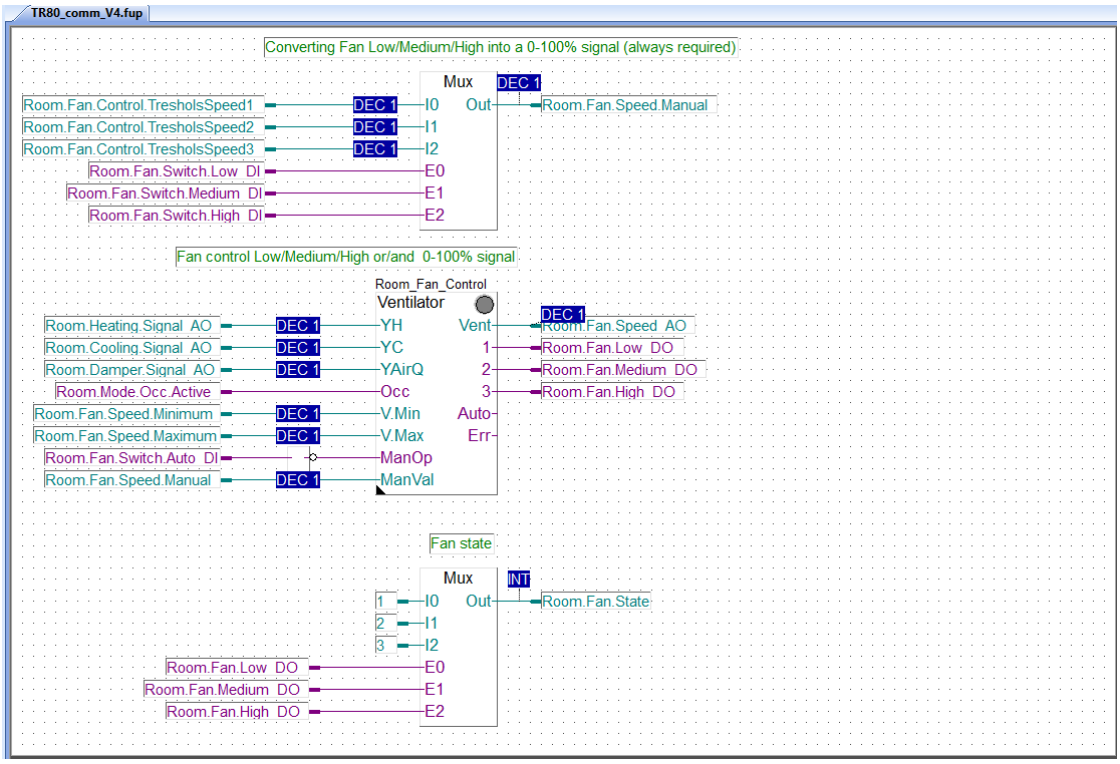
Fan Control

Room Application often uses a Fan Coil Unit. The control of the fan is implemented on this page.

The fan can be controlled by

- Manual User intervention via Wall Module
- Heating valve signal
- Cooling valve signal

Fans may have up to 3 speed or can be controlled with a continuous 0.0. to 100.0 % signal. Both functions, digital or analogue control is supported.



A Appendix

A.1 Symbols

This symbol refers the reader to additional information, either in this manual, in another manual, or in technical documents on this topic. There are no direct references to such documents.

This symbol indicates instructions which must be followed strictly.

A.2 Address of Saia Burgess Controls

Saia-Burgess Controls AG
Bahnhofstrasse 18
CH-3280 Murten
Switzerland
Tel: +41 26 580 30 00
SBC on the web:
www.saia-pcd.com
Support-site:
www.sbc-support.com