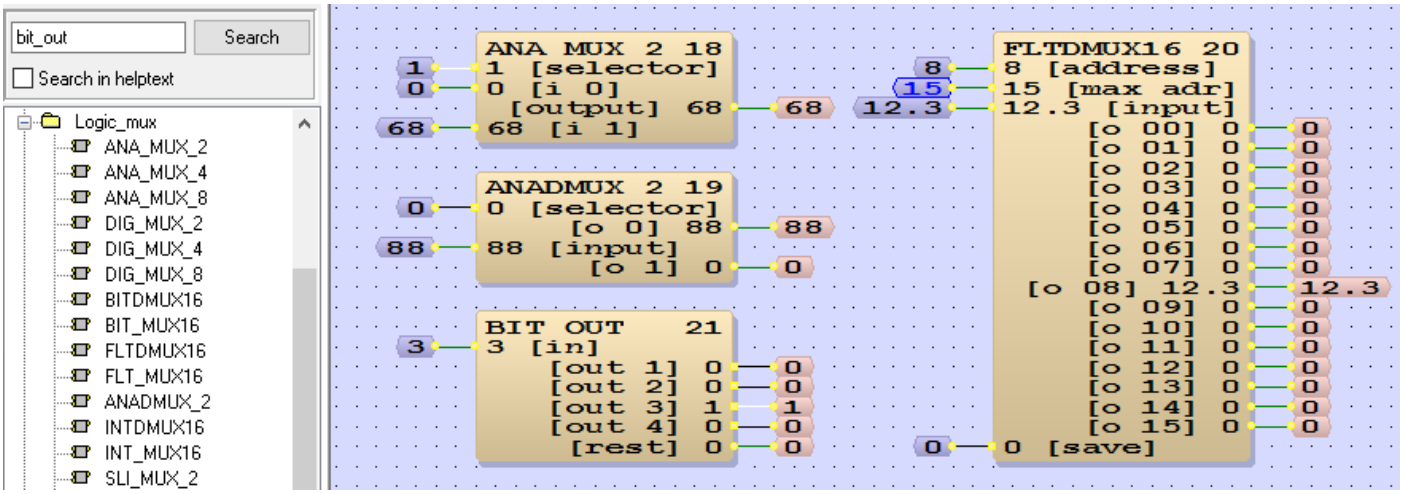
TT200206 – FUP - Multiplexer and Demultiplexer Functions

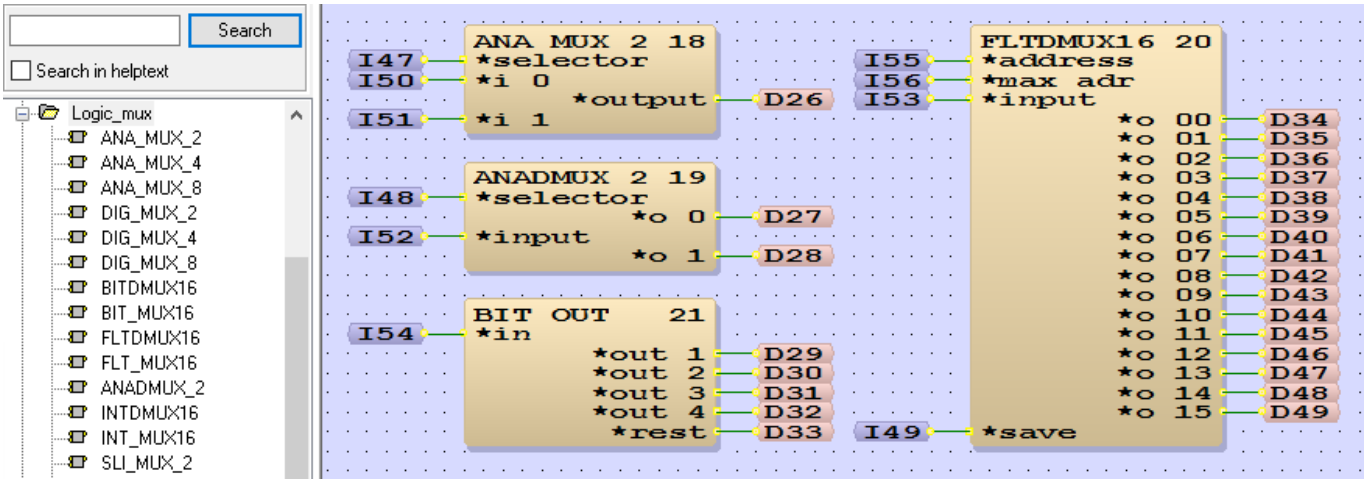
Note	This Support Knowledge Base article KB is the result of a support request. It is not part of the official documentation of DEOS AG and does not claim to be complete. The article is intended to support the solution of a similar problem. If you have any questions, comments or additions, please contact DEOS AG Support.
Title	Multiplexer and Demultiplexer Functions (TT200206)
Object	FUP
Reference version	2
Date	02.2020
Author	EK
Goal	To explain the usage of the Multiplexer and Demultiplexer Function Blocks

Content:

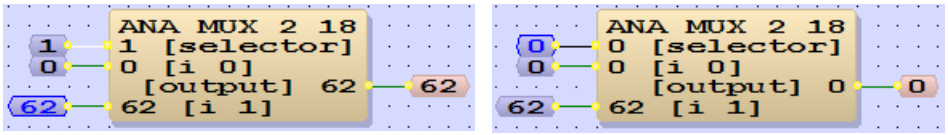


TT200206 – FUP - Multiplexer and Demultiplexer Functions

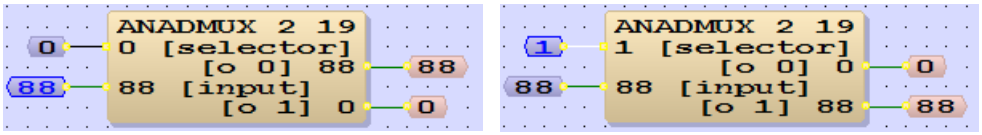
1. Multiplexer and demultiplexer modules are used frequently in our program. We've many modules for different point types and they're under the "Logic_mux" group



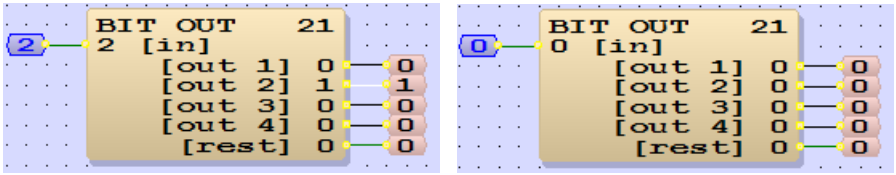
2. First, we look at the "ANA_MUX_2" module. This module can be used to select 2 floating points value based on a digital input and send it to the output. For example, we can use it to send the PID output (i_1) to the cooling valve (output) when the AHU (selector) is ON, and close the valve (i_0 = 0) when AHU is OFF



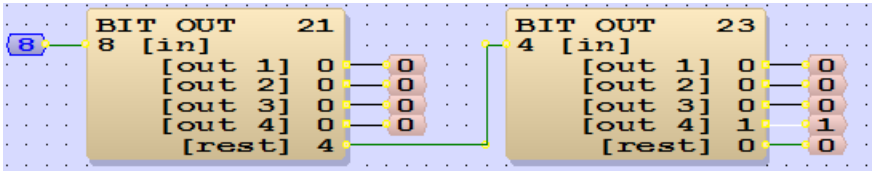
3. The second module is "ANADMUX_2" which is the opposite to "ANA_MUX_2". This module sends the input to 2 output respectively, based on the digital input. For example, we can send the PID output (input) to the cooling valve (o_0) or heating valve (o_1), based on the operating mode of the AHU (selector)



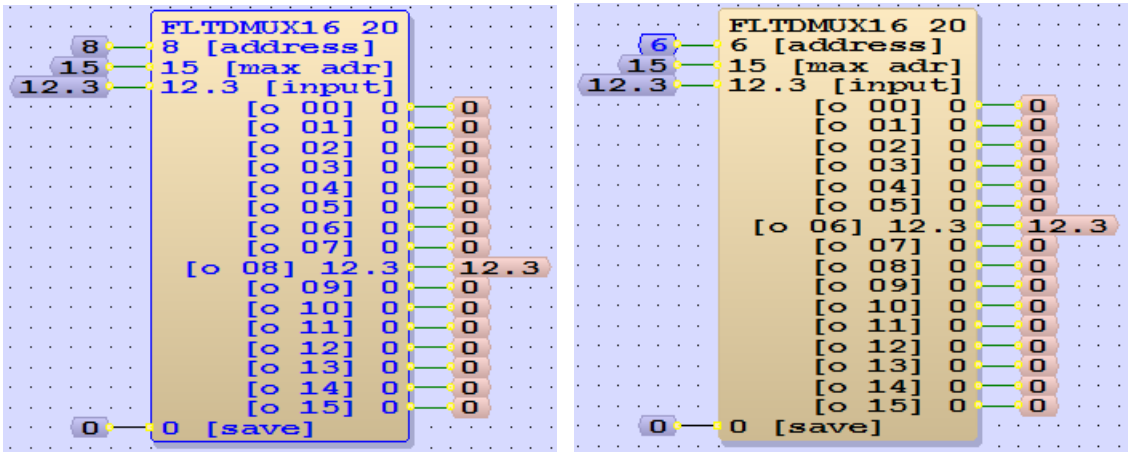
4. For the "BIT_OUT" module (under "Logic_dig"), one of the outputs (out_1 to out_4) will turn on based on the input. For example, when in=2, out_2 will become 1 and the other outputs are 0. If in=0 or in>4, all outputs are 0. This module can be used for mode operation (e.g. auto/on/off), to perform different actions based on the mode input



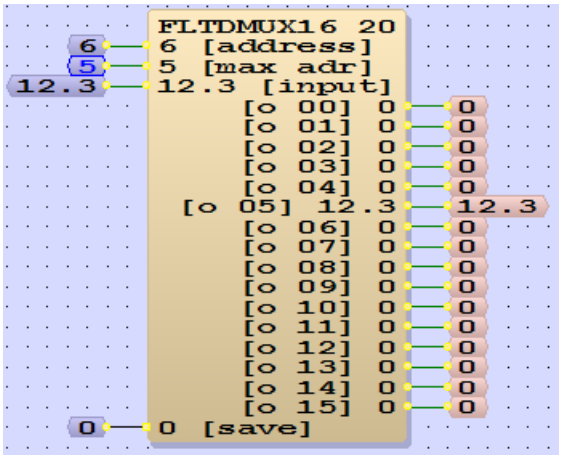
5. This module can be used in cascaded mode, to handle more than 4 outputs, like this



6. The last module to show you is “FLTDMUX16”. The module sends the floating point input to one of the outputs (o_00 to o_15) based on the address input



7. If you don’t have 16 outputs, you can limit the number of outputs using the “max_adr” input. So, when the “address” is greater than the “max_adr”, the “max_adr” output will be set, like below



8. If the input “save” is set to 1, then you can send different values to different outputs, by changing the “address” and the “input” at the same time, like below. To reset the “old” outputs, set “save” to 0