



Copyright (c) 2016 Genesis Technologies AG Switzerland, all rights reserved

KNX Pool driver (GTSPC4ZKNXPL) for KNX enabled pools

1. Disclaimer

- Genesis Technologies does provide this driver as it is. There is no commitment to support, nor does Genesis Technologies assure a decent reaction time in case of a problem. Genesis Technologies can refuse all support, discontinue support, end the driver lifetime at any time. Genesis Technologies will not refund in any case licenses
- Notwithstanding the foregoing, Genesis Technologies is not liable to licensee for any damages, including compensatory, special, incidental, exemplary, punitive, or consequential damages, connected with or resulting from this license agreement or licensee's use of this software. Licensee's jurisdiction may not allow such a limitation of damages, so this limitation may not apply.

By using this driver you accept that you have read and accepted the terms above!

2. What is the pool driver? What is it used for?

This driver does integrate pools that have a KNX module.

3. What pool proxy features are supported?

- All of them for air temperature, pool, spa and extras

4. Additional features

- Free configurable extras page
- Only pool, only spa can be activated. Or none.

5. Theory of operation / general info

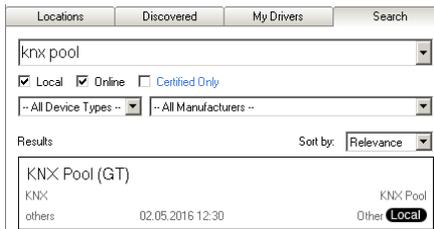
Every KNX enabled pool has at least a setpoint and status address for temperature and as well addresses for special functions like the cover or so. The pool driver takes the water/air temperature (DPT_9) and shows it on the Navigators. New targets for the water temp can be set also via a 2 Byte address (DPT_9). Additional addresses (DPT_1) can be added for heating sources, for pump speeds/operation (DPT_1) and (DPT_5). The extras page is able to take switch and list entries (DPT_1) and (DPT_5).

6. Software installation Control4

6.1. Prerequisites

- The driver uses the Control4 KNX network driver or similar drivers to connect to KNX. Please make sure that this driver is installed in your project and is connected to KNX, running
- Download the driver and placed them in %USERPROFILE%\Control4\Drivers.

6.2. Start Composer, go to *System Design* and select the tab *Search* in the right top window. Search for *knx pool GT*

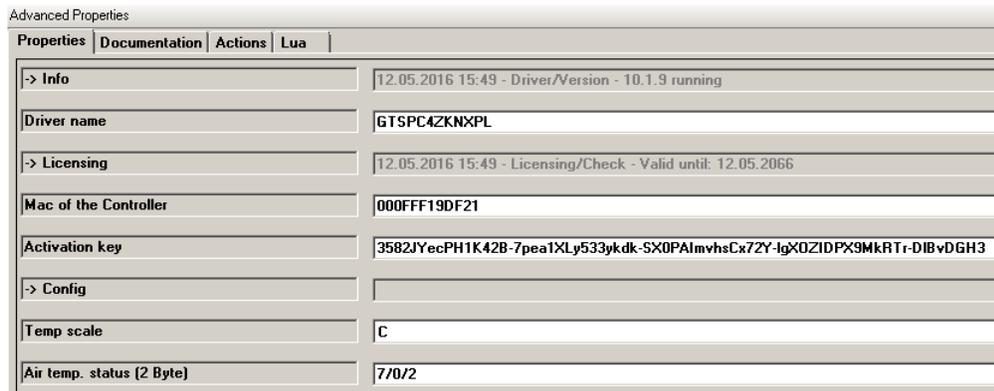


6.3. Drag and drop the *knx pool GT* driver into the wished room. Click on it and find the Properties page. (Hovering with the mouse over the Properties will show up hints)



6.4. Proxy properties

- Set the feature set with the checkboxes *Has Pool Controls*, *Has Spa Controls*, *Has Air Temperature*. The properties below will change and only addresses for the selected modes can be added. **Do not touch the Auxiliary Controls.** They are auto generated from the driver Properties and will be overwritten.



6.5. Driver properties

- -> Info: Does show the driver health info. If the driver experience issues this line will display it. If *Version x.x.x running* is shown the driver is fully started and works. If another message appears the driver will not work correct, a decent action is required.
- Driver name: The product identification of the driver
- Mac of the Controller: Not changeable.
- Activation key: Insert here the license key which is displayed in the driver dashboard after you bought a license and assigned it to the system. Once assigned in the Dashboard the license may jump in automatically once the driver connects to the online services
- Temp scale: Select your local scale, C of F
- Air temp status (2 Byte): Insert the address the air temperature (DPT_9).

-> Pool	
Pool temp. status (2 Byte)	7/2/2
Pool setpoint (2 Byte)	7/0/0
Pool setpoint status (2 Byte)	7/0/1
Pool pump mode (1 bit)	W=2/0/0, S=2/0/1, 0 = Off, 1 = On;
Pool pump modes (1 Byte)	W=0/0/2, S=0/0/4, 0 = Off, 1=Squirrel, 2=Mad, 3=Insane
Pool heating sources (1 bit)	W=2/0/3, S=2/0/4, D=Solar, W=2/0/6, S=2/0/7, D=Wind;

- Pool temp. status (2 Byte): Insert the address for the pool temperature
- Pool setpoint (2 Byte): Insert the address for the pool setpoint
- Pool setpoint status (2 Byte): Insert the address for the pool setpoint status
- Pool pump mode (1 bit): For the pump modes a 1 bit or a 1 Byte address can be used. If both property lines are inserted the 1 Byte mode has priority. If old Flash navigators are in use the 1 bit addressing should be preferred.
Notation: Comma delimited string, parameters:
write(W)=address, status(S)= address, value=mode

Example: 1bit datapoint, 2/0/0 is the write address, 2/0/1 is the status (RW), 0 should be send to switch off the pump, 1 should be send to switch on the pump
W=2/0/0, S=2/0/1, 0=Off, 1=On

- Pool pump modes (1 Byte): For the pump modes a 1 bit or a 1 Byte address can be used. If both property lines are inserted the 1 Byte mode has priority. If old Flash navigators are in use the 1 bit addressing should be preferred.
Notation: Comma delimited string, parameters:
write(W)=address, status(S)= address, value=mode

Example: 1Byte datapoint, 0/0/2 is the write address, 0/0/4 is the status (RW), 0 should be send to switch off the pump, 1 should be send for mode Squirrel, 2 for mode Mad, 3 for mode Insane
W=0/0/2, S=0/0/4, 0=Off, 1=Squirrel, 2=Mad, 3=Insane

- Pool heating sources (1 bit): Multiple heating sources can be present. All of them can be configured in one property line, delimited by;
Notation: Comma delimited string, use semicolon for multiple heating sources. Parameters for each heating source:
write(W)=address, status(S)= address, description(d)=name

Example: 2 heating sources: Wind energy with write address 2/0/6 and status 2/0/7, solar energy with write address 2/0/3 and status 2/0/4
W=2/0/3, S=2/0/4, D=Solar; W=2/0/6, S=2/0/7, D=Wind;

If only one source is available only insert one

Pool cover open/close address (1 bit)	1/0/2
Pool cover invert open/close operation	false
Pool cover stop address (1 bit)	1/0/3
Pool cover moving time open (s)	3
Pool cover moving time close (s)	3
Pool cover localisation	Cover
Pool cover open,close,stop localisation	Open,Close,Stop

- Pool cover open/close address (1 bit): Insert the address for the pool cover to open and close if the cover is connected to a blind actuator, works as a standard blind
- Pool cover invert open/close operation: If the actuator that opens the pool cover is not cabled

- right the operation can be inverted
- Pool cover stop address (1 bit): Address to stop the pool cover from moving, if the cover is connected to a blind actuator
- Pool cover moving time open (s): How long it takes to open the pool cover
- Pool cover moving time close (s): How long it takes to close the pool cover
- Pool cover localization: In extras page a list named with *Cover* will show up, if the system is not set to English the *Cover* may be another localized phrase. Just replace it and the extras menu will then also be localized
- Pool cover open,close,stop localization: See above, the covers list in extras will contain open, close, stop entries for the pool cover. Replace the phrases by the local ones to get a complete localized system.

-> Spa	
Spa temp. status (2 Byte)	770/2
Spa setpoint (2 Byte)	772/0
Spa setpoint status (2 Byte)	772/1
Spa pump mode (1 bit)	W=2/0/0, S=2/0/1, 0 = Off, 1 = On;
Spa pump modes (1 Byte)	W=0/0/2, S=0/0/4, 0 = Off,1=Squirrel,2=Mad,3=Insane
Spa heating sources (1 bit)	W=2/0/3, S=2/0/4, D=Sun;W=2/0/6,S=2/0/7, D=Tesla Battery;

- Spa temp. status (2 Byte): Insert the address for the spa temperature
- Spa setpoint (2 Byte): Insert the address for the spa setpoint
- Spa setpoint status (2 Byte): Insert the address for the spa setpoint status
- Spa pump mode (1 bit): For the pump modes a 1 bit or a 1 Byte address can be used. If both property lines are inserted the 1 Byte mode has priority. If old Flash navigators are in use the 1 bit addressing should be preferred.

Notation: Comma delimited string, parameters:
write(W)=address, status(S)= address, value=mode

Example: 1bit datapoint, 2/0/0 is the write address, 2/0/1 is the status (RW), 0 should be send to switch off the pump, 1 should be send to switch on the pump
W=2/0/0, S=2/0/1, 0=Off, 1=On

- Spa pump modes (1 Byte): For the pump modes a 1 bit or a 1 Byte address can be used. If both property lines are inserted the 1 Byte mode has priority. If old Flash navigators are in use the 1 bit addressing should be preferred.

Notation: Comma delimited string, parameters:
write(W)=address, status(S)= address, value=mode

Example: 1Byte datapoint, 0/0/2 is the write address, 0/0/4 is the status (RW), 0 should be send to switch off the pump, 1 should be send for mode Squirrel, 2 for mode Mad, 3 for mode Insane
W=0/0/2, S=0/0/4, 0=Off, 1=Squirrel, 2=Mad, 3=Insane

- Spa heating sources (1 bit): Multiple heating sources can be present. All of them can be configured in one property line, delimited by;

Notation: Comma delimited string, use semicolon for multiple heating sources. Parameters for each heating source:
write(W)=address, status(S)= address, description(d)=name

Example: 2 heating sources: Solar energy with write address 2/0/3 and status 2/0/4, battery from your Tesla with write address 2/0/6 and status 2/0/7
W=2/0/3, S=2/0/4, D=Sun;W=2/0/6,S=2/0/7, D=Tesla Battery;

If only one source is available only insert one

-> Extras	
Extras (1 bit)	W=2/0/0, S=2/0/1, D=Cover;W=4/0/2,S=4/0/3, D=chute;
Extras (1 Byte)	W=0/0/7, S=0/0/9, D=RGB, 10=red,20=green,30=blue;W=0/0/12, S=0/0/14, D=Colors, 10=yello,20=purple,30=black;
-> Log	
Lua window	Debug
C4 log	None
Log period (h)	1
Log auth	

- Extras (1 bit): Property to manage the Extras page, adds toggle buttons to the page
Notation: Comma delimited string, use semicolon for multiple buttons.
write(W)=address, status(S)= address, description(d)=name

Example: The cover can be opened and closed in this example with write address 2/0/0 and status 2/0/1 (usually covers works like blinds, so see the section above), the chute functions uses write address 4/0/2 and status 4/0/3

W=2/0/0, S=2/0/1, D=Cover;W=4/0/2,S=4/0/3, D=chute;

- Extras (1 Byte): Property to manage the Extras page, adds list buttons to the page
Notation: Comma delimited string, use semicolon for multiple buttons. Parameters for each button:
write(W)=address, status(S)= address, description(d)=name, value=list entry

Example: The pool has 2 RGB lights which are using 1 Byte (DPT 5) values to change the color:
W=0/0/7, S=0/0/9, D=RGB, 10=red,20=green,30=blue;W=0/0/12, S=0/0/14, D=Colors, 10=yello,20=purple,30=black;

- -> Log / Lua window / C4 log: Selects the log level printed out in Lua window from Composer. Use the level *Debug* to see what the driver does
- Log period (h): The time frame the log should output
- Log auth: Developer can give you a code to see more logs. Usually not really interesting, only for in deep debug

6.6. Connections

KNX link

7. Known issues

- For maximum performance we recommend not to use the native Control4 knx network driver, especially in larger installations it may fail or be overloaded. Instead use a KNX gateway from B+B Steuerungstechnik and our driver "*EIB Medientechnik Gateway IP*". More info on <https://technet.genesis-technologies.ch>
- Any changes on properties that affect the *Extras* page may be followed by a *Refresh Navigators* to complete reflect all changes
- The pump operating status may be only accurate (blue round icon) if the language is set to English

8. Updates / Support

- Updates are available on: <https://technet.genesis-technologies.ch>
- Bugs should be reported to: info@genesis-technologies.ch
- Updates from older .c4i drivers can be done via *Drivers -> Manage drivers...* in Composer. The old driver will be marked as obsolete and can be updated to the new .c4z driver. Properties are maintained.

