KNX Cookbook

KNX development on basis of existing system components

Components by TAPKO

Summary

This document is a development help for KNX newcomers.

This document describes the development of a KNX device based on existing KNX components provided by the company TAPKO.

This document is part of the KNX Specifications v2.1.

Document updates

Version	Date	Modifications
1.0.0	2011.05.13	Preparation of the final version.
1.01.01	2011.08.11	Update according final Manufacturer Tool release.
01.01.02	2013.10.14	Editorial updates for the publication of KNX Specifications 2.1.

References

[01]	Chapter 3/1/2	“Glossary”
[02]	Volume 6	“Profiles”

Filename:	02_05_02 Components by TAPKO v01.01.02.docx
Version:	01.01.02
Status:	Final version
Savedate:	2013.10.14
Number of pages:	19

Contents

1 Developing applications on the basis of Tapko Component KNX-SIM 4
1.1 Needed software 4
1.2 About the TAPKO SIM-KNX Evaluation Device 4

2 TAPKO SIM-KNX solution 5
2.1 What you should know about the TAPKO SIM-KNX GOs............ 5
2.2 Concept of the TAPKO SIM-KNX solution 6
2.2.1 Step 1 6
2.2.2 Step 2 7
2.2.3 Step 3 8
2.3 Phase 1: Evaluation............ 8
2.4 Phase 2: AM integration 9

3 In practice 10
3.1 Group Object vs. Communication Object 10
3.2 Connect & Play............ 10
3.3 Create an ETS4 product with MT4............ 11
3.4 Import into ETS4 16
3.5 Use the terminal SW to check whether ETS did configure the device correctly 18
3.6 Use the terminal SW in order to simulate the AM 19

1 Developing applications on the basis of Tapko Component KNX-SIM

1.1 Needed software

- Installed MT4 software and valid license.
- Installed ETS4 software and valid license.
- FTDI FT232BL driver.
- Any terminal software.

1.2 About the TAPKO SIM-KNX Evaluation Device

- This device is based on the TAPKO **SIM-KNX** device. This is defined as follows.
  - Profile Class = System 7, see [02].
  - Interface = serial asynchronous.
  - ASCII protocol.
- TAPKO SIM-KNX is available in two editions: 128 or 254 GOs.
- In this sample the 128 edition will be used.
- Two variants available: RS232 or USB.
- This example will feature the USB variant:
  - AM = another PCB based on the FT232BL chip from FTDI
  - The FT232BL chip converts USB into RS232 legacy peripherals.
- TAPKO SIM-KNX devices come delivered with 128 pre-defined GOs.
- TAPKO SIM-KNX devices have additional resources available, which will not be discussed here because they are not needed for our sample, in summary:.
  - memory range for user parameters, address: 503Ch to 523Bh.
  - advanced transmit conditions
  - string specific resources

2 TAPKO SIM-KNX solution

2.1 What you should know about the TAPKO SIM-KNX GOs

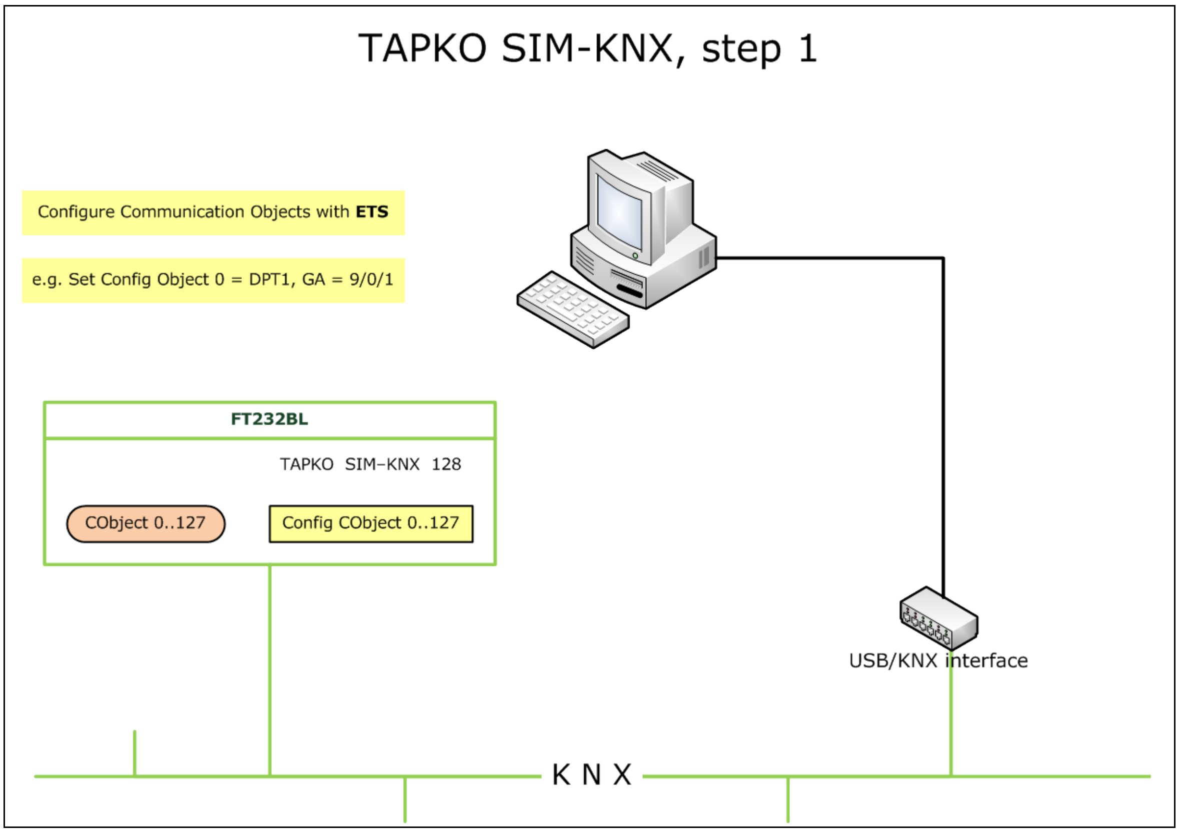
- All 128 GOs are pre-defined.
- GO Table Address = 43FCh
- Association Table address = 41FFh
- GA Table address = 4000h
- Also important is to remember the following restrictions:
  - number of associations = 254
  - number of GAs = 254
- every single GO is configured via 4 parameters
  - object Type = DPT, 8 bit
  - repTime = cyclic sending or receive timeout, 8 bit
  - sendConfig = send configuration, 16 bit
  - rcvConfig = receive (indication) configuration, 16 bit
- GO parameter addresses:
  - DPT -> 4846h + 6*[0..127] + 0
  - repTime -> 4846h + 6*[0..127] + 1
  - sendConfig -> 4846h + 6*[0..127] + 2
  - rcvConfig -> 4846h + 6*[0..127] + 4

NOTE 1 4846h = 43FCh + 044Ah

2.2 Concept of the TAPKO SIM-KNX solution

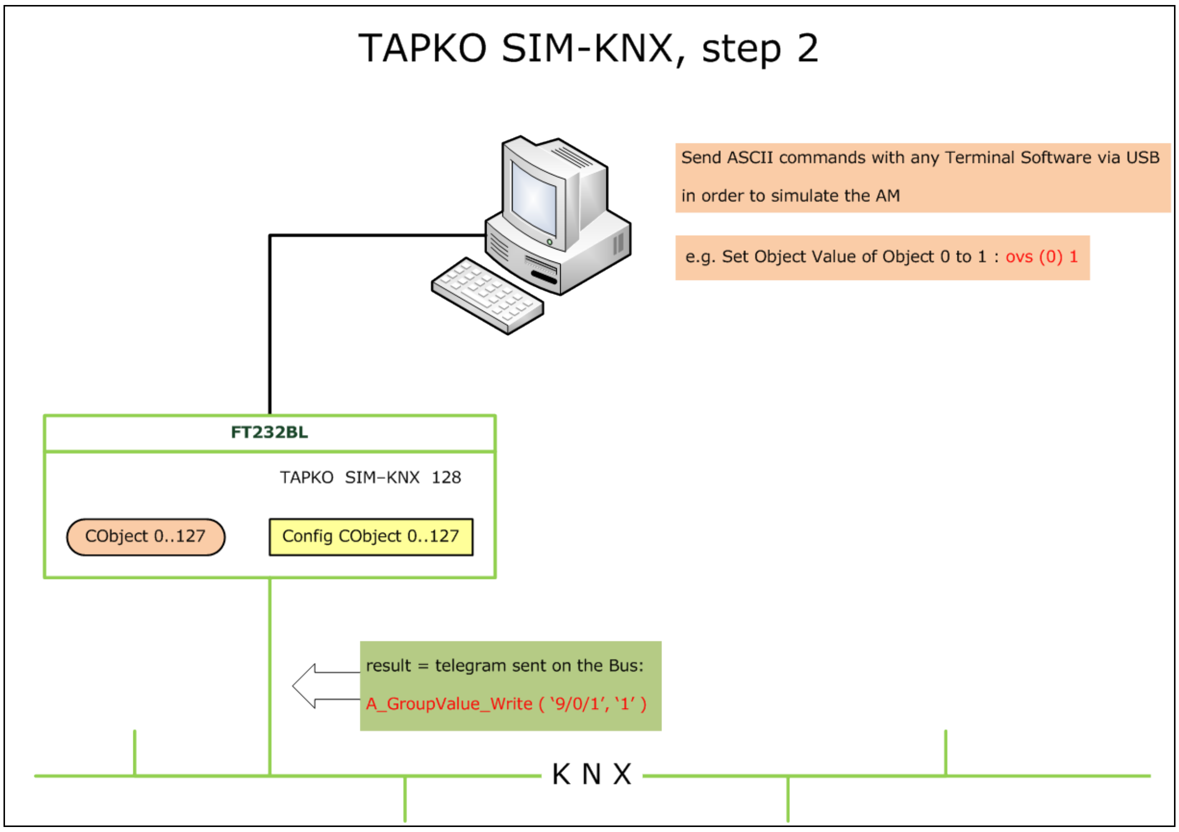
2.2.1 Step 1

- Use the SIM-KNX Evaluation device SIM-KNX + FT232BL.
- Configure the GOs with ETS.



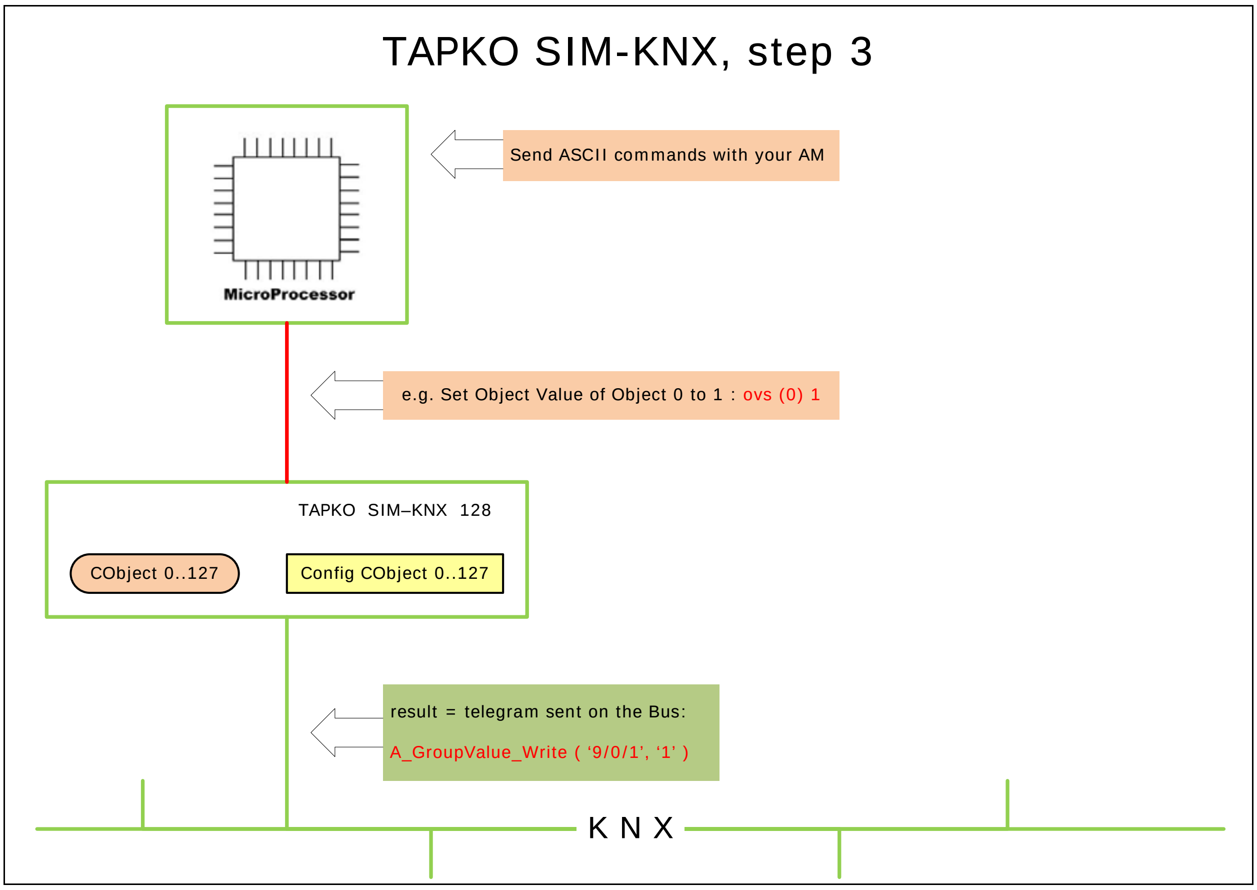
2.2.2 Step 2

- Simulate the AM.
- Send ASCII commands from the PC to the FT232BL chip via USB.
- The ASCII commands between FT232BL and KNX-SIM are sent as RS232.
- Check for all GOs.


TAPKO SIM-KNX, step 2
Send ASCII commands with any Terminal Software via USB in order to simulate the AM
e.g. Set Object Value of Object 0 to 1 : ovs (0) 1
FT232BL
TAPKO SIM-KNX 128
CObject 0..127
Config CObject 0..127
result = telegram sent on the Bus:
A_GroupValue_Write ('9/0/1', '1')
K N X

2.2.3 Step 3

- Replace the evaluation device by a regular SIM-KNX device.
- This is; replace the FT232BL chip by your AM.
- Make sure that the AM sends the same ASCII commands as in step 2.
- Make sure to send these commands as RS232.
- Check for all GOs.


TAPKO SIM-KNX, step 3
MicroProcessor
Send ASCII commands with your AM
e.g. Set Object Value of Object 0 to 1 : ovs (0) 1
TAPKO SIM–KNX 128
CObject 0..127
Config CObject 0..127
result = telegram sent on the Bus:
A_GroupValue_Write ('9/0/1', '1')
K N X

2.3 Phase 1: Evaluation

- Use the TAPKO SIM-KNX Evaluation Device
- create an ETS4 product with MT4 in order to pre-configure the required GOs for the AP
  - define the individual DPTs
  - set up the required parameter structure for the further GO settings fine-tuning
- add the product to an ETS4 database/project
  - link its GOs with GAs
  - set its parameters
  - set its GOs flags
  - program the device (with ETS via KNX)
- use the terminal SW to check whether ETS4 did configure the device correctly
  - connect it to both KNX (TP) and your computer (USB)
  - play by sending ASCII commands via the terminal SW

- use the terminal SW in order to simulate the AM -> sending ASCII commands in order to set the values of the individual GOs

2.4 Phase 2: AM integration

Reaching this stage really means you should get in contact with TAPKO.

The work to be done in this phase comes basically down to:

- Replace the TAPKO ‘SIM-KNX Evaluation’ Device with the (regular) TAPKO ‘SIM-KNX’ Device.
- Integrate the AM with the TAPKO SIM-KNX PCB.
- Test: make sure the same commands are sent as during the evaluation phase.
- Certify the entire product, this counts for both
  - the HW, being the TAPKO SIM-KNX device with integrated AM
  - the AP.

3 In practice

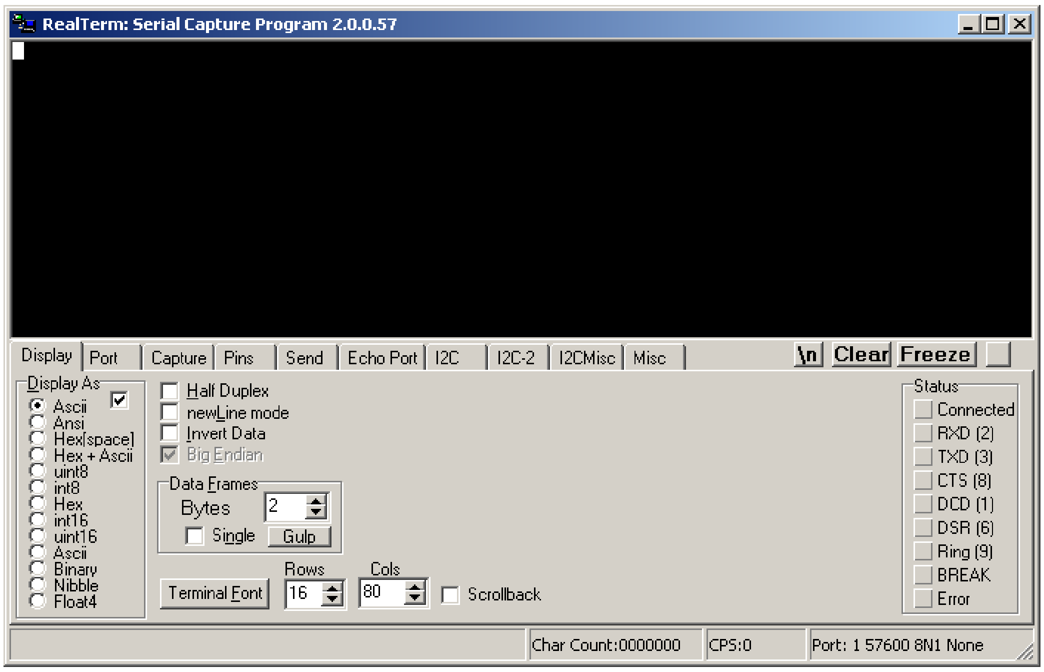
3.1 Group Object vs. Communication Object

- First of all, the terms ‘Group Object’ and ‘Communication Object’ are synonyms.
- ‘Communication Object’ is used in MT, ETS and other tools.
- ‘Group Object’ is the only term used in the rest of the KNX specifications and is therefore considered as the only correct one.
- Both terms will however be used in this Cookbook because it is here were practice and theory meet. ‘Communication Object’ will only be used when absolutely necessary, e.g. when the context is MT.

3.2 Connect & Play

Install the FT232BL driver

- Link: http://www.ftdichip.com/Drivers/VCP.htm
- This will add a virtual com port to you computer
- Use any terminal software, e.g. Serial Terminal from RealTerm
  - Link: http://realterm.sourceforge.net/
  - A screenshot is show in the below Figure 1.



```
Char Count:0000000   CPS:0   Port: 1 57600 8N1 None
```

Figure 1 – Realterm terminal screenshot

Connect

- Wire the device.
- Open the terminal software (see Figure 1).
- click tab page Port.
- select set the virtual com port that has been added.
- put its settings to 9600, 8, 1, none.

Play

- Click tab page Send.
- Make sure to set EOL (end of line) settings to '+CR'.
- Type e.g. 'dag' in the top entry field and click 'Send ASCII'. This command will ask the device to send back its Individual Address. This is a perfect method to test the principal working of this test setup.
- Other commands that could be used in this context are: das, dpg, dps, dr, dsg, dvg, gci . Check the TAPKO documentation for more details

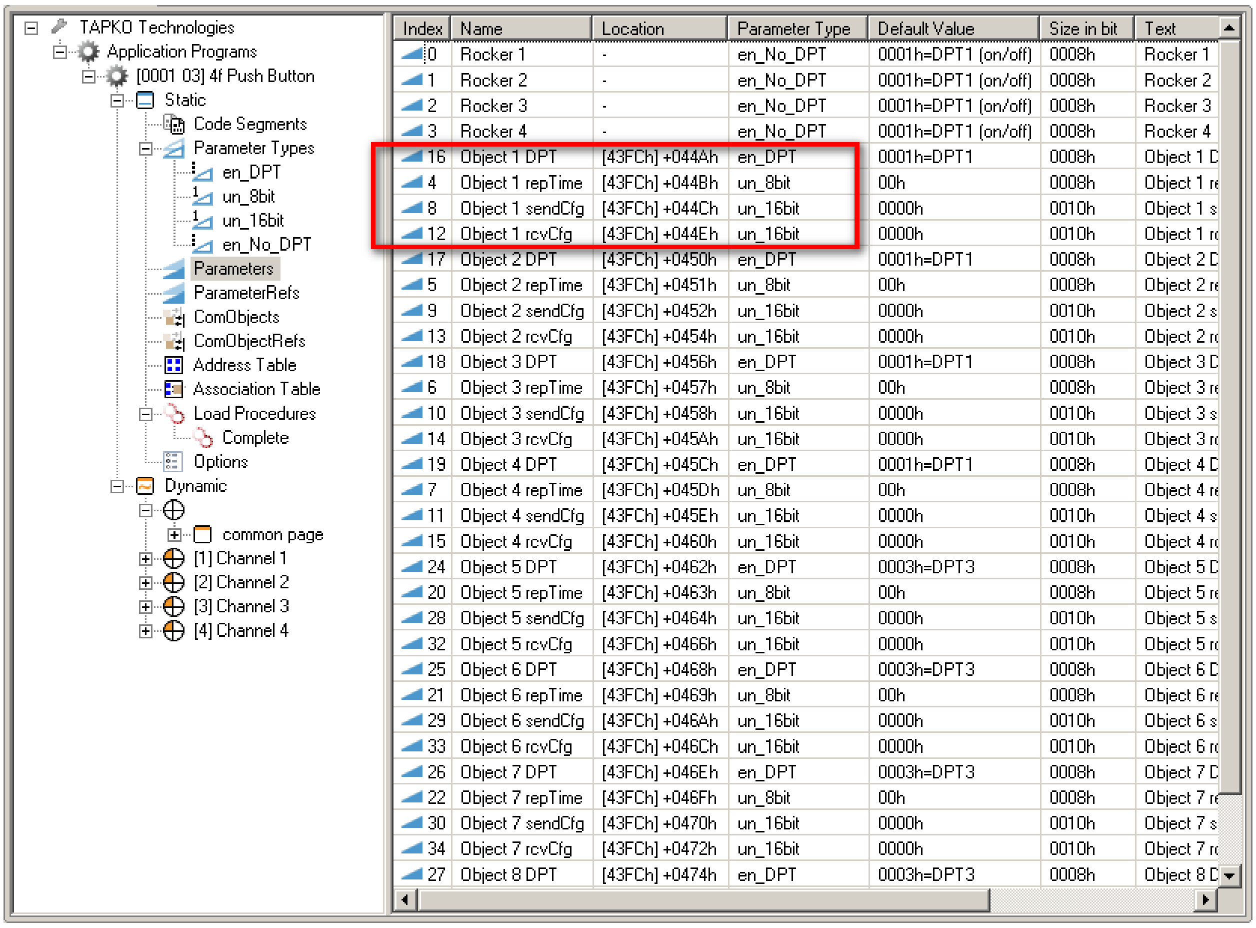
3.3 Create an ETS4 product with MT4

- Add 'Project'
  - KNX Manufacturer Tool Project
  - Name = KNXproduct100203-01
  - Target ETS Version = ETS4
- Add 'Hardware' to the project
- Serial Number = SN17.06.2010-01
- Name = PCB SN17.06.2010-01
- Version = 1
- Define Product (= commercial realization of HW) within Hardware
- Order Number = 4f-PB SN17.06.2010-01
- Product Name = 4 fold Push Button V1.0
- Language
- Add 'Application Program' to the project
  - Number = 1 & Version = 1
  - Name = 4f Push Button
  - Mask Version = 7.1
- Add 'Catalog' to the project
  - Add 'CatalogSection'
  - Name = MyTapkoSim01
  - Number = 1
  - Add 'CatalogItem' within the above CatalogSection
- Open Hardware
  - select Application Programs
  - add new Hardware2Program
  - Select 4f Push Button
- Open the AP, in static:
- Import binary data -> select tapko-sim.s19, this will:
- Add Code Segments
  - Segment 0: Address = 4000h, Size = 01FFh
  - Segment 1: Address = 41FFh, Size = 01FDh

- Segment 2: Address = 0700h, Size = 0290h
- Segment 3: Address = 0990h, Size = 0001h
- Segment 4: Address = 43FCh, Size = 1AA0h

- Add Address Table
  - Code Segment -> [4000h], Max Entries = 00FEh, Offset=0
- Add Association Table
  - Code Segment -> [41FFh], Max Entries = 00FEh, Offset=0
- Add Load Controls
  - LC00: Connect
  - LC01: CompareProp, Count=1, Inline Data={00 01 02 03 04 07 00 00 00 00}, Object Index=0, Property ID=4Eh, Start Element=1
  - LC02: Unload, LSM Index=1
  - LC03: Unload, LSM Index=2
  - LC04: Unload, LSM Index=3
  - LC05: Load, LSM Index=1
  - LC06: AbsSegment, Access=FFh , Address=4000h, LSM Index=1, MemType=3, SegFlags=80h, SegType=0, Size=01FFh
  - LC07: TaskSegment, Address=4000h, ApplicationNumber=0, ApplicationVersion=0, LSM Index=1, Manufacturer ID=0, PEI Type=0
  - LC08: LoadCompleted, LSM Index=1
  - LC09: Load, LSM Index=2
  - LC10: AbsSegment, Access=FFh , Address=41FFh, LSM Index=2, MemType=3, SegFlags=80h, SegType=0, Size=01FDh
  - LC11: TaskSegment, Address=41FFh, ApplicationNumber=0, ApplicationVersion=0, LSM Index=2, Manufacturer ID=0, PEI Type=0
  - LC12: LoadCompleted, LSM Index=2
  - LC13: Load, LSM Index=3
  - LC14: AbsSegment, Access=0 , Address=0700h, LSM Index=3, MemType=2, SegFlags=0, SegType=0, Size=0290h
  - LC15: AbsSegment, Access=0 , Address=0990h, LSM Index=3, MemType=2, SegFlags=0, SegType=1, Size=1
  - LC16: AbsSegment, Access=FFh , Address=43FCh, LSM Index=3, MemType=3, SegFlags=80h, SegType=0, Size=1AA0h
  - LC17: TaskSegment, Address=43FCh, ApplicationNumber=0, ApplicationVersion=0, LSM Index=3, Manufacturer ID=0, PEI Type=0
  - LC18: TaskSegment, Address=43FCh, ApplicationNumber=0, ApplicationVersion=0, LSM Index=5, Manufacturer ID=0, PEI Type=0
  - LC19: LoadCompleted LSM Index=3
  - LC20: Restart
  - LC21: Disconnect

- Add Parameter Types
  - Restriction : name = en_DPT, range: DPT1 = 1, DPT3 = 3
  - Restriction : name = en_No_DPT, range: DPT 1 (on/off) = 1, DPT 3 (dimming) = 3, No Function = 99
  - Number : name = un_8bit , 8 bit
  - Number : name = un_16bit, 16 bit
- Add Parameters
  - Virtual : Rocker [1..4], en_No_DPT , default = DPT1
  - Memory : Object [1..8] repTime, un_8bit, default = 0, memory location =
  - Memory : Object [1..8] sendCfg, un_16bit, default = 0, memory location =
  - Memory : Object [1..8] rcvCfg, un_16bit, default = 0, memory location =
  - Memory : Object [1..4] DPT, en_DPT, default = DPT1, memory location =
  - Memory : Object [5..8] DPT, en_DPT, default = DPT3, memory location =
- Add Group Objects
  - Object #0..3 : object [1..4] DPT1, Text = on/off , 1 bit
  - Object #4..7 : object [5..8] DPT3, Text = dimming , 1 bit
- Result:


- TAPKO Technologies
  - Application Programs
    - [0001 03] 4f Push Button
      - Static
        - Code Segments
        - Parameter Types
          - en_DPT
          - un_8bit
          - un_16bit
          - en_No_DPT
        - Parameters
        - ParameterRefs
        - ComObjects
        - ComObjectRefs
        - Address Table
        - Association Table
        - Load Procedures
          - Complete
        - Options
      - Dynamic
        - common page
        - [1] Channel 1
        - [2] Channel 2
        - [3] Channel 3
        - [4] Channel 4

Index	Name	Location	Parameter Type	Default Value	Size in bit	Text
0	Rocker 1	-	en_No_DPT	0001h=DPT1 (on/off)	0008h	Rocker 1
1	Rocker 2	-	en_No_DPT	0001h=DPT1 (on/off)	0008h	Rocker 2
2	Rocker 3	-	en_No_DPT	0001h=DPT1 (on/off)	0008h	Rocker 3
3	Rocker 4	-	en_No_DPT	0001h=DPT1 (on/off)	0008h	Rocker 4
16	Object 1 DPT	[43FCh] +044Ah	en_DPT	0001h=DPT1	0008h	Object 1 D
4	Object 1 repTime	[43FCh] +044Bh	un_8bit	00h	0008h	Object 1 re
8	Object 1 sendCfg	[43FCh] +044Ch	un_16bit	0000h	0010h	Object 1 s
12	Object 1 rcvCfg	[43FCh] +044Eh	un_16bit	0000h	0010h	Object 1 rc
17	Object 2 DPT	[43FCh] +0450h	en_DPT	0001h=DPT1	0008h	Object 2 D
5	Object 2 repTime	[43FCh] +0451h	un_8bit	00h	0008h	Object 2 re
9	Object 2 sendCfg	[43FCh] +0452h	un_16bit	0000h	0010h	Object 2 s
13	Object 2 rcvCfg	[43FCh] +0454h	un_16bit	0000h	0010h	Object 2 rc
18	Object 3 DPT	[43FCh] +0456h	en_DPT	0001h=DPT1	0008h	Object 3 D
6	Object 3 repTime	[43FCh] +0457h	un_8bit	00h	0008h	Object 3 re
10	Object 3 sendCfg	[43FCh] +0458h	un_16bit	0000h	0010h	Object 3 s
14	Object 3 rcvCfg	[43FCh] +045Ah	un_16bit	0000h	0010h	Object 3 rc
19	Object 4 DPT	[43FCh] +045Ch	en_DPT	0001h=DPT1	0008h	Object 4 D
7	Object 4 repTime	[43FCh] +045Dh	un_8bit	00h	0008h	Object 4 re
11	Object 4 sendCfg	[43FCh] +045Eh	un_16bit	0000h	0010h	Object 4 s
15	Object 4 rcvCfg	[43FCh] +0460h	un_16bit	0000h	0010h	Object 4 rc
24	Object 5 DPT	[43FCh] +0462h	en_DPT	0003h=DPT3	0008h	Object 5 D
20	Object 5 repTime	[43FCh] +0463h	un_8bit	00h	0008h	Object 5 re
28	Object 5 sendCfg	[43FCh] +0464h	un_16bit	0000h	0010h	Object 5 s
32	Object 5 rcvCfg	[43FCh] +0466h	un_16bit	0000h	0010h	Object 5 rc
25	Object 6 DPT	[43FCh] +0468h	en_DPT	0003h=DPT3	0008h	Object 6 D
21	Object 6 repTime	[43FCh] +0469h	un_8bit	00h	0008h	Object 6 re
29	Object 6 sendCfg	[43FCh] +046Ah	un_16bit	0000h	0010h	Object 6 s
33	Object 6 rcvCfg	[43FCh] +046Ch	un_16bit	0000h	0010h	Object 6 rc
26	Object 7 DPT	[43FCh] +046Eh	en_DPT	0003h=DPT3	0008h	Object 7 D
22	Object 7 repTime	[43FCh] +046Fh	un_8bit	00h	0008h	Object 7 re
30	Object 7 sendCfg	[43FCh] +0470h	un_16bit	0000h	0010h	Object 7 s
34	Object 7 rcvCfg	[43FCh] +0472h	un_16bit	0000h	0010h	Object 7 rc
27	Object 8 DPT	[43FCh] +0474h	en_DPT	0003h=DPT3	0008h	Object 8 D

NOTE 2 The red rectangle indicates the 4 configuration parameters for one CO.

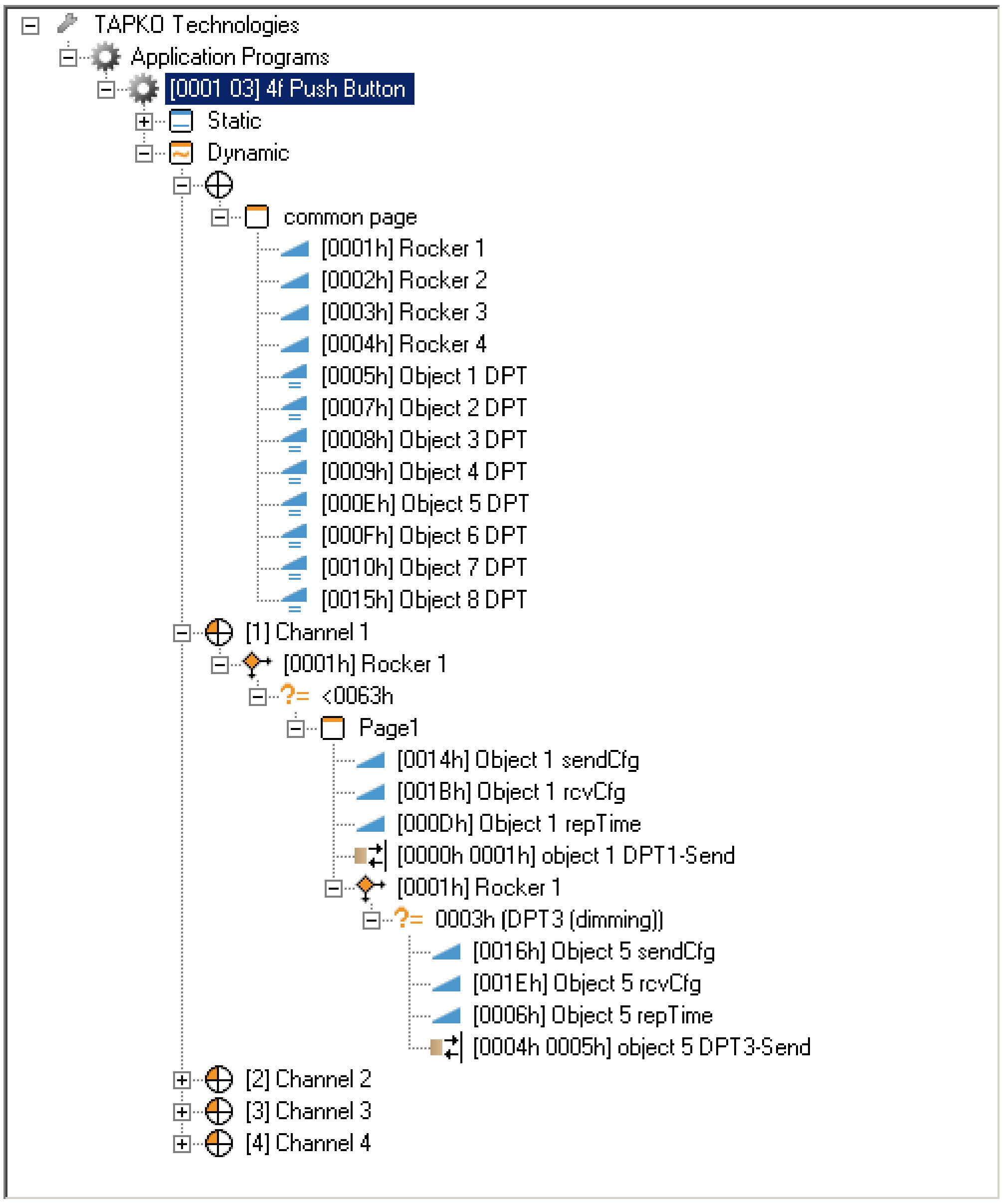
- In the dynamic part of the AP:
- Add ChannelIndependentBlock
  - Add ParameterBlock: Name = common page, Text = common
    - Add ParameterRefRef: Rocker 1..4
    - Add Assign: Object [1..4] DPT, Fixed Value = DPT1
    - Add Assign: Object [5..8] DPT, Fixed Value = DPT3
- Add Channel: Name = Channel 1, Text = Rocker 1
  - Add Choose: Parameter = Rocker 1
  - Add When: default = False, test: <99

    NOTE 3 99' is the value for 'No Function'
  - Add ParameterBlock: Name = Page1, Text = parameters
    - Add ParameterRefRef: Object 1 sendCfg
    - Add ParameterRefRef: Object 1 rcvCfg
    - Add ParameterRefRef: Object 1 repTime
    - Add ComObjectRefRef: object 1 DPT1
    - Add Choose: Parameter = Rocker 1
      - Add When: default = False, test: 3

        NOTE 4 Rocker 1 has dim function => add dim object
      - Add ParameterRefRef: Object 5 sendCfg
      - Add ParameterRefRef: Object 5 rcvCfg
      - Add ParameterRefRef: Object 5 repTime
      - Add ComObjectRefRef: object 5 DPT3
- Add Channel: Name = Channel 1, Text = Rocker 2
  - Add Choose: Parameter = Rocker 2
  - Add When: default = False, test: <99
  - Add ParameterBlock: Name = Page2, Text = parameters
    - Add ParameterRefRef: Object 2 sendCfg
    - Add ParameterRefRef: Object 2 rcvCfg
    - Add ParameterRefRef: Object 2 repTime
    - Add ComObjectRefRef: object 2 DPT1
    - Add Choose: Parameter = Rocker 2
      - Add When: default = False, test: 3
      - Add ParameterRefRef: Object 6 sendCfg
      - Add ParameterRefRef: Object 6 rcvCfg
      - Add ParameterRefRef: Object 6 repTime
      - Add ComObjectRefRef: object 6 DPT3
- Add Channel: Name = Channel 1, Text = Rocker 3
  - Add Choose: Parameter = Rocker 3

    - Add When: default = False, test: <99
    - Add ParameterBlock: Name = Page3, Text = parameters
        - Add ParameterRefRef: Object 3 sendCfg
        - Add ParameterRefRef: Object 3 rcvCfg
        - Add ParameterRefRef: Object 2 repTime
        - Add ComObjectRefRef: object 3 DPT1
        - Add Choose: Parameter = Rocker 3
            - Add When: default = False, test: 3
            - Add ParameterRefRef: Object 7 sendCfg
            - Add ParameterRefRef: Object 7 rcvCfg
            - Add ParameterRefRef: Object 7 repTime
            - Add ComObjectRefRef: object 7 DPT3
- Add Channel: Name = Channel 1, Text = Rocker 4
    - Add Choose: Parameter = Rocker 4
    - Add When: default = False, test: <99
    - Add ParameterBlock: Name = Page4, Text = parameters
        - Add ParameterRefRef: Object 4 sendCfg
        - Add ParameterRefRef: Object 4 rcvCfg
        - Add ParameterRefRef: Object 4 repTime
        - Add ComObjectRefRef: object 4 DPT1
        - Add Choose: Parameter = Rocker 4
            - Add When: default = False, test: 3
            - Add ParameterRefRef: Object 8 sendCfg
            - Add ParameterRefRef: Object 8 rcvCfg
            - Add ParameterRefRef: Object 8 repTime
            - Add ComObjectRefRef: object 8 DPT3

- Result:

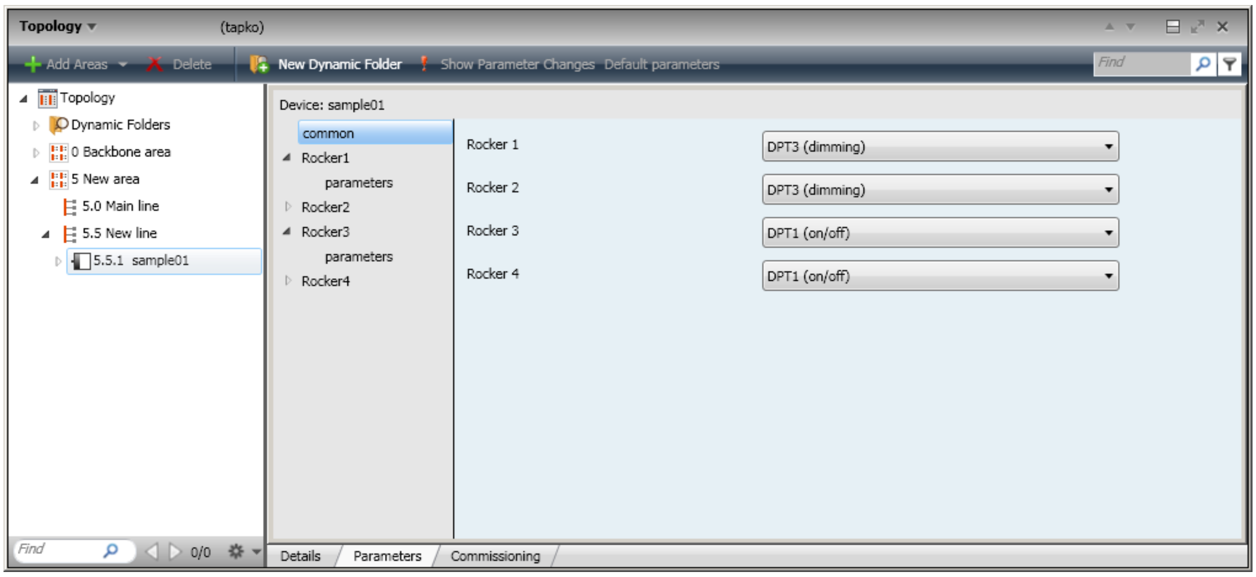


```
TAPKO Technologies
  Application Programs
    [0001 03] 4f Push Button
      Static
      Dynamic
        
          common page
            [0001h] Rocker 1
            [0002h] Rocker 2
            [0003h] Rocker 3
            [0004h] Rocker 4
            [0005h] Object 1 DPT
            [0007h] Object 2 DPT
            [0008h] Object 3 DPT
            [0009h] Object 4 DPT
            [000Eh] Object 5 DPT
            [000Fh] Object 6 DPT
            [0010h] Object 7 DPT
            [0015h] Object 8 DPT
        [1] Channel 1
          [0001h] Rocker 1
            <0063h
              Page1
                [0014h] Object 1 sendCfg
                [001Bh] Object 1 rcvCfg
                [000Dh] Object 1 repTime
                [0000h 0001h] object 1 DPT1-Send
                [0001h] Rocker 1
                  0003h (DPT3 (dimming))
                    [0016h] Object 5 sendCfg
                    [001Eh] Object 5 rcvCfg
                    [0006h] Object 5 repTime
                    [0004h 0005h] object 5 DPT3-Send
        [2] Channel 2
        [3] Channel 3
        [4] Channel 4
```

3.4 Import into ETS4

- build the project in MT4
- create a test project via the 'Edit' menu in MT4
- import the test project in ETS4
- the device is then visible in the 'Device' panel under 'All Devices'
- link its GOs with GAs, in our example:
  - 9/0/1, 9/0/2, 9/0/3, 9/0/4 for object 1..4 (on/off)
  - 9/1/1, 9/1/2, 9/1/3, 9/1/4 for object 5..8 (dimming)
- set the individual parameters
- set the individual GOs flags

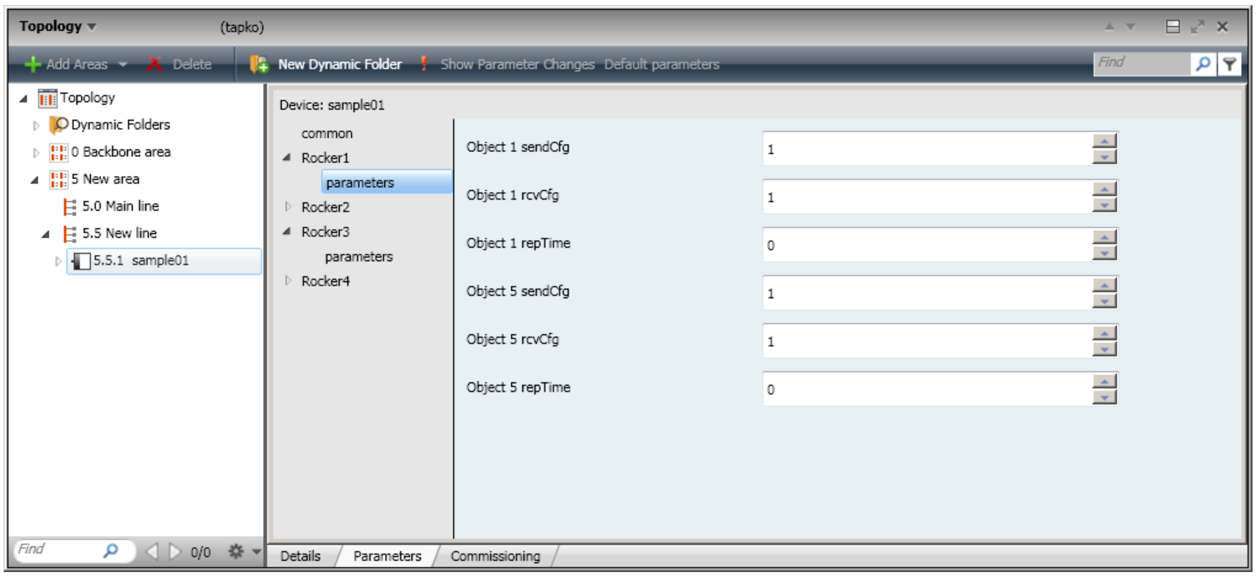
- program the device (with ETS via KNX)
- ETS4 Screenshots
    - Common parameters



Device: sample01

Rocker 1	DPT3 (dimming)
Rocker 2	DPT3 (dimming)
Rocker 3	DPT1 (on/off)
Rocker 4	DPT1 (on/off)

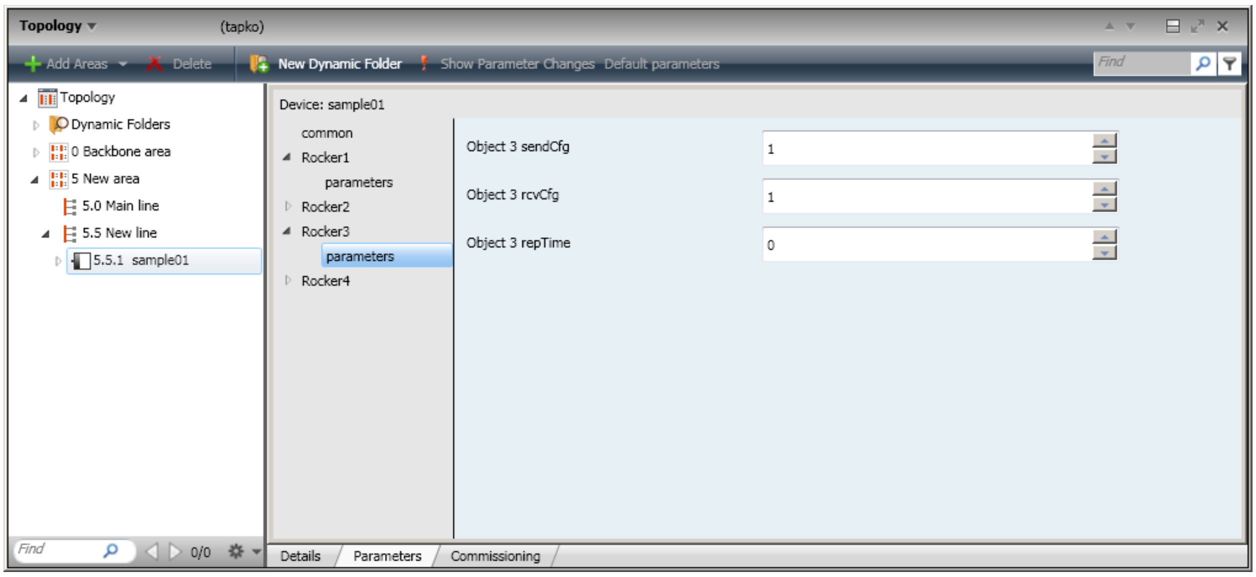
- Parameters for Rocker1



Device: sample01

Object 1 sendCfg	1
Object 1 rcvCfg	1
Object 1 repTime	0
Object 5 sendCfg	1
Object 5 rcvCfg	1
Object 5 repTime	0

- parameters for Rocker3

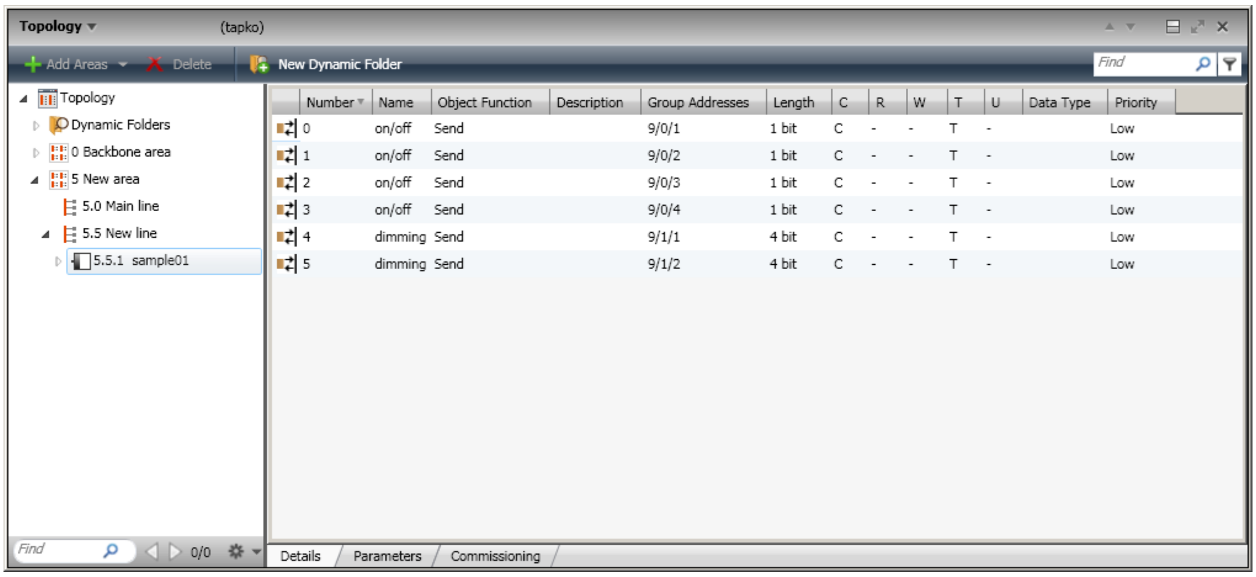


Device: sample01

common
Rocker1
parameters
Rocker2
Rocker3
parameters
Rocker4

Object 3 sendCfg	1
Object 3 rcvCfg	1
Object 3 repTime	0

- Group Objects



Number	Name	Object Function	Description	Group Addresses	Length	C	R	W	T	U	Data Type	Priority
0	on/off	Send		9/0/1	1 bit	C	-	-	T	-		Low
1	on/off	Send		9/0/2	1 bit	C	-	-	T	-		Low
2	on/off	Send		9/0/3	1 bit	C	-	-	T	-		Low
3	on/off	Send		9/0/4	1 bit	C	-	-	T	-		Low
4	dimming	Send		9/1/1	4 bit	C	-	-	T	-		Low
5	dimming	Send		9/1/2	4 bit	C	-	-	T	-		Low

3.5 Use the terminal SW to check whether ETS did configure the device correctly

- Connect via USB.
- Send the following ASCII commands.
  - ocg (0) to check the configuration of GO nr. 0 (= ‘object 1’)
  - expected response: <ocg (0)> 1 0 $df $0001 $0001 0
  - ogg (0) to check the assigned GA(s) for the same GO
  - expected response: <ogg (0)>$4801
  - repeat for all other GOs
  - check the TAPKO documentation for more details

3.6 Use the terminal SW in order to simulate the AM

- connect via USB
- send the following ASCII commands:
    - ovs (0) 1 to value of GO nr. 0 (= object 1) to 1
    - expected result: a A_GroupValue_Write telegram for GA = 9/0/1 on the bus
    - this can be checked in three ways:
        - with an actual actuator -> lamp switches on
        - with the ETS4 Group- or Busmonitor
        - with the command ovg (0)
    - repeat for all other GOs
    - please check the TAPKO documentation for more details