

KNX Basics

- ❖ **Why KNX ?**
- ❖ **KNX Architecture**
- ❖ **KNX Topology**
- ❖ **ETS Configuration Steps**
- ❖ **ETS Demo**

KNX = saving energy

Proven with facts:

- up to 40 % with KNX shading control
- up to 50 % with KNX individual room control
- up to 60 % with KNX lighting control
- up to 60 % with KNX ventilation control

Extend KNX installations with:

- **Window contacts:** avoid open windows when the heating is in comfort mode
- **Presence detection:** unoccupied rooms should not be illuminated

KNX: the future proof bus system

KNX Logo = triple token, it means that the device:

1. is compatible with any other KNX device, regardless of the manufacturer
2. can be configured with one and the same tool, i.e. ETS
3. has passed quality control tests (membership = ISO 9001 compliance)

KNX = backwards compatibility

- Extend 20 year old installations with new KNX devices
- Extend today's installations with any future KNX device

KNX = combining all possible building application types

- Lighting, Shutters/Blinds
- HVAC
- Security
- Metering, Energy management
- Audio/Video, White goods
- etc.

KNX architecture

- **Physical network**
 - topological structure based on individual addresses
 - **Role: management**
- **Logical network**
 - functional blocks (lighting, blinds, etc.)
 - **Role: runtime**

- **Based on 3 core elements**
 - **Devices**
 - (devices have) **Group Objects**
 - (and the group objects are linked via) **Group Addresses**

KNX architecture details

- **Communication Media: TP, PL, IP, RF**
- **Passive components**
 - **no ICT**
 - **i.e. power supply (PS), din rails, cable, connectors,...**
- **Active components: sensor vs. actuators**
 - **ICT**
 - **Sending vs. listening**
- **Interfaces: RS232, USB, IP, RF**
 - **Connect tools (ETS) to the bus**
- **Couplers: backbone, line, repeaters**
 - **Optimize bus traffic**

KNX topology: from bottom to top

■ Devices

- Identified by 3 numbers: **x.y.z** (= individual address)

■ Lines

- Identified by 2 numbers: **x.y** (= line address)
- Up to 255 devices can be linked to 1 line

■ Areas

- Identified by 1 number: **x** (= area address)
- Contains: 1 main line (address = x.0) + up to 15 lines
- Lines are connected to a main line via line couplers

■ Backbone Line

- Line address = 0.0
- Main lines are connect to the backbone line via backbone couplers

KNX topology details

■ Devices

- x.y.z: line = x.y, area = x, device = z:1..255

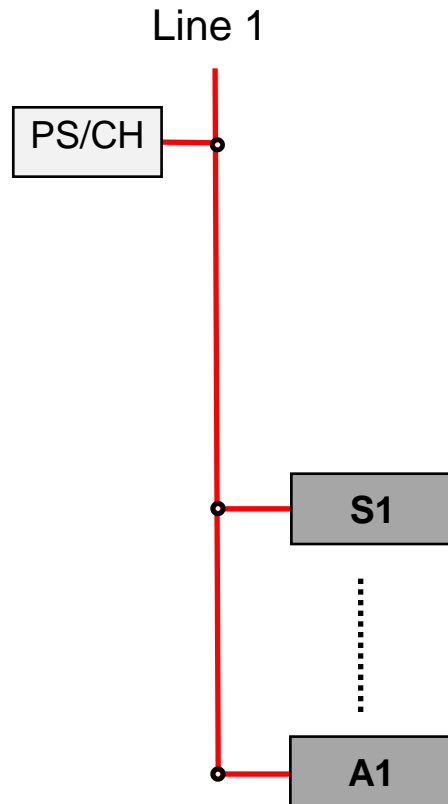
■ Lines

- x.y: where y:0..15, if y = 0: main line of area x
- x: area, x:0..15, if x = 0 and y = 0: backbone line

■ Backbone Coupler, Line Coupler & Repeater

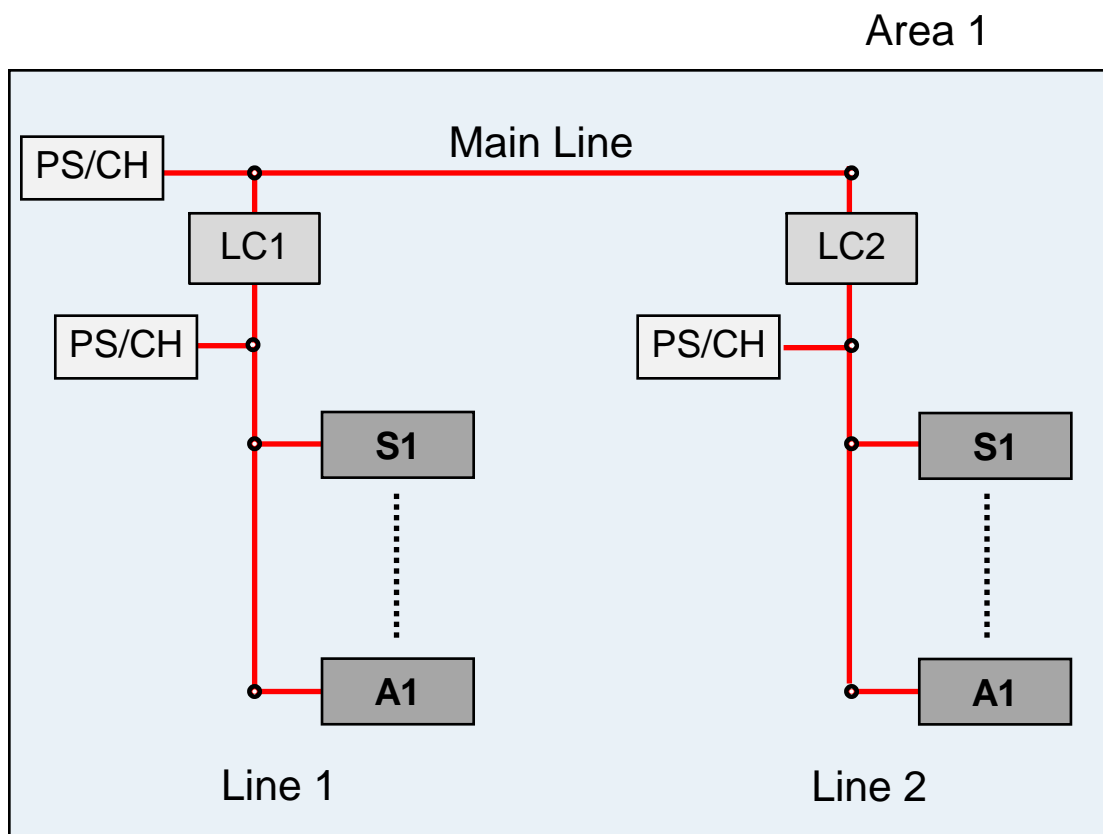
- Same device type
- But with a different configuration
 - BC: x.0.0 where x:1..15
 - LC: x.y.0 where x:0..15 and y:1..15 – **note**: x = 0: backbone line
 - Repeater: x.y.z same as LC but z <> 0

Topology: line



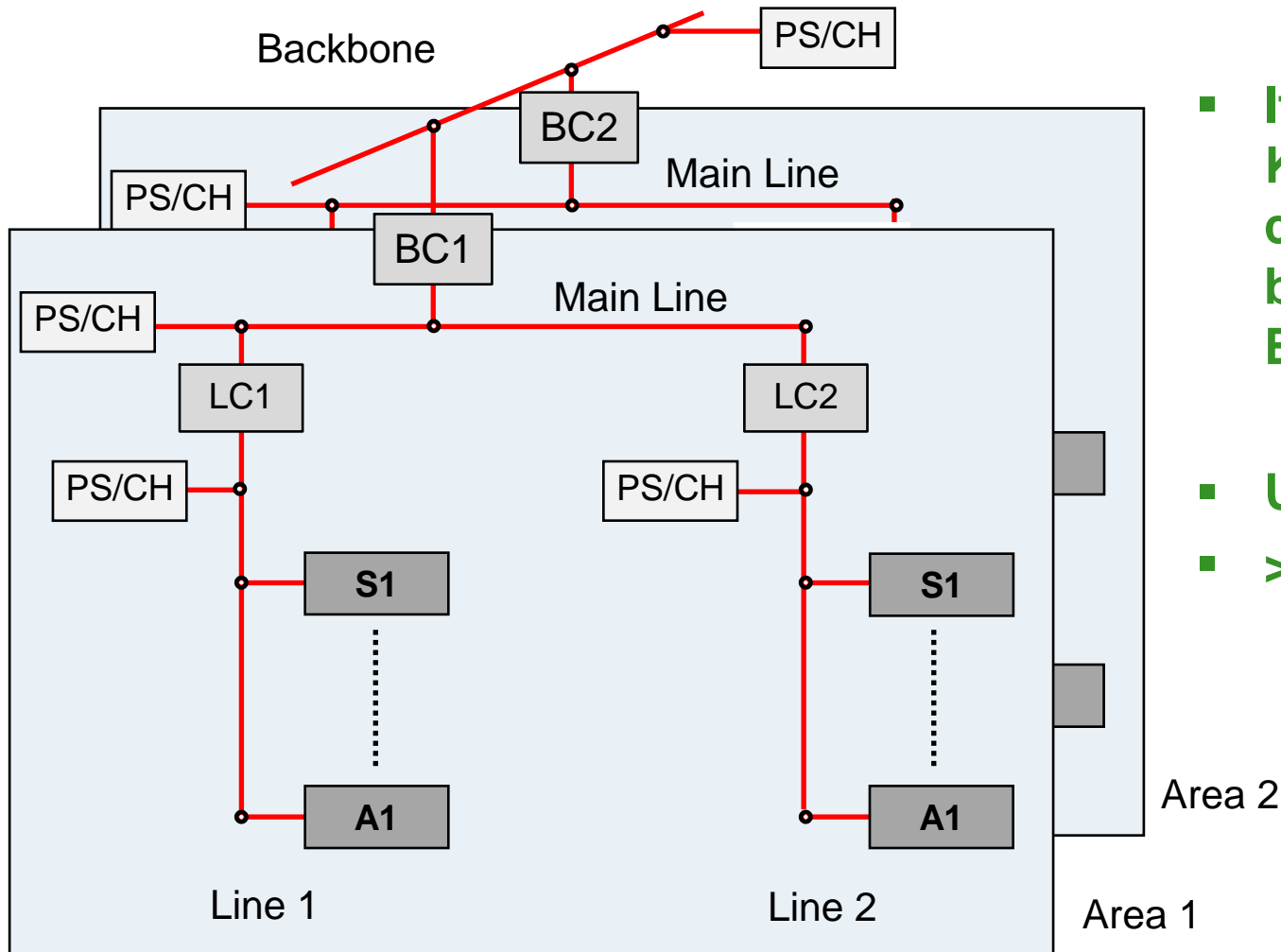
- **Maximum 255 devices**
- **PS limitation = 64 devices**
- **Solution: up to 4 line segments**
- **Per extra line segment:**
 - **PS+CH**
 - **Repeater**
- **Remark #1: per line in total**
 - **Max 4 PS+CH**
 - **Max 3 Repeaters**
- **Remark #2: CH = choke**
 - **AC/DC separation**
 - **AC: '1' vs. '0'**

Topology: area



- Area = group of lines connected by Line Couplers (LC) to a main line.
- Up to 15 lines

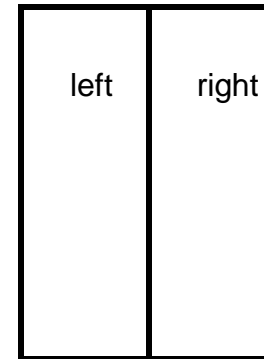
Topology: several areas



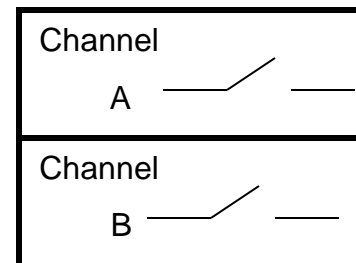
- If required, the KNX installation can be expanded by means of a Backbone line.
- Up to 15 areas
- > 65,000 devices

2 (active) device categories

- **Sensors**
e.g. 2 fold push button

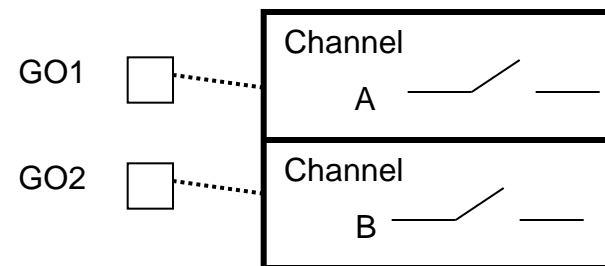
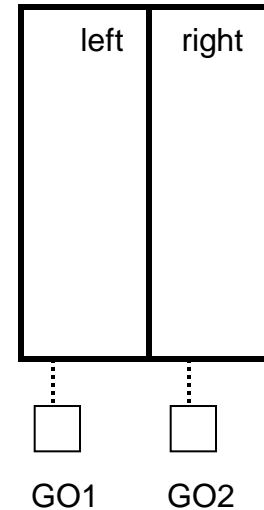


- **Actuators**
e.g. 2 fold binary output



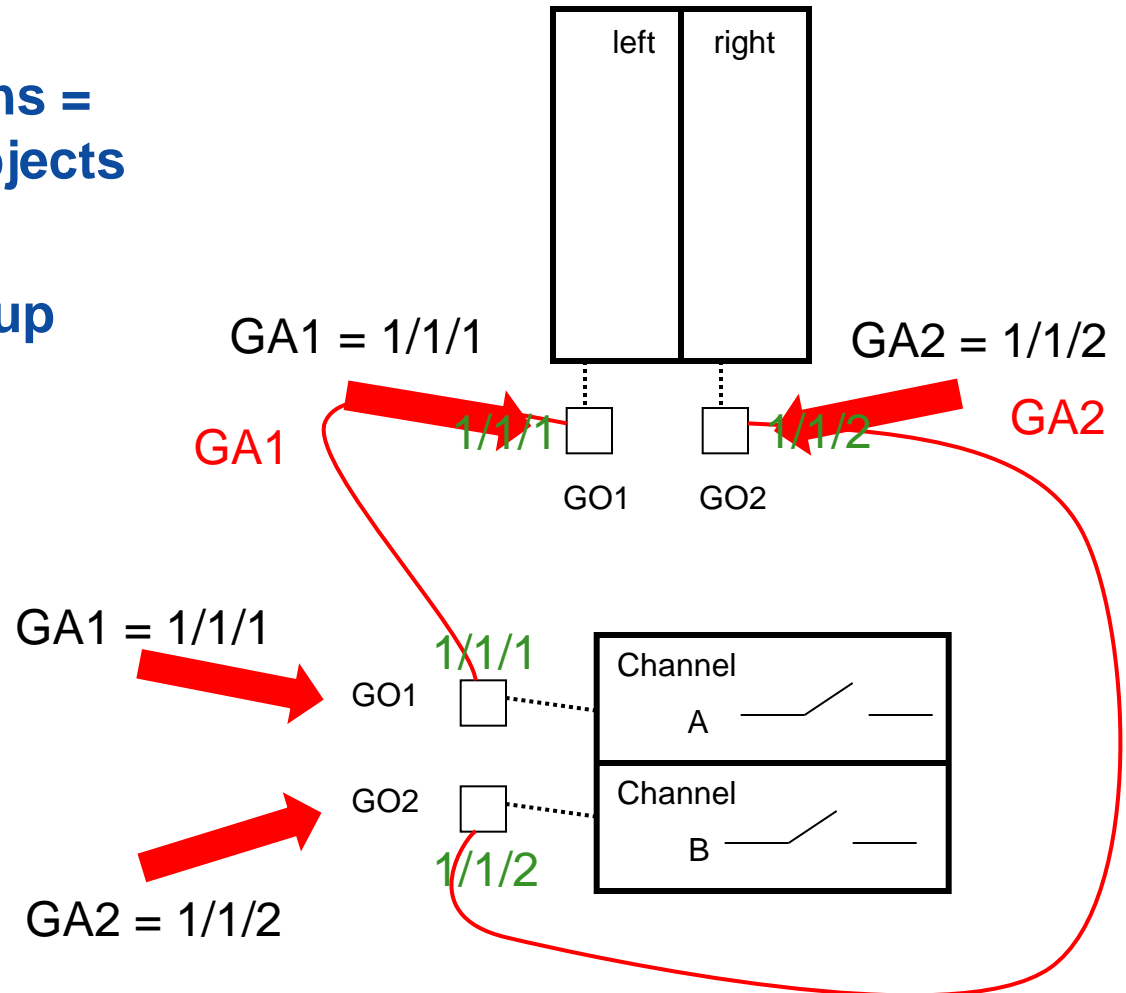
Device functionality = Group Objects

- **Group Objects (GO)**
- **GO = representation of a button, switch, etc. in ETS**

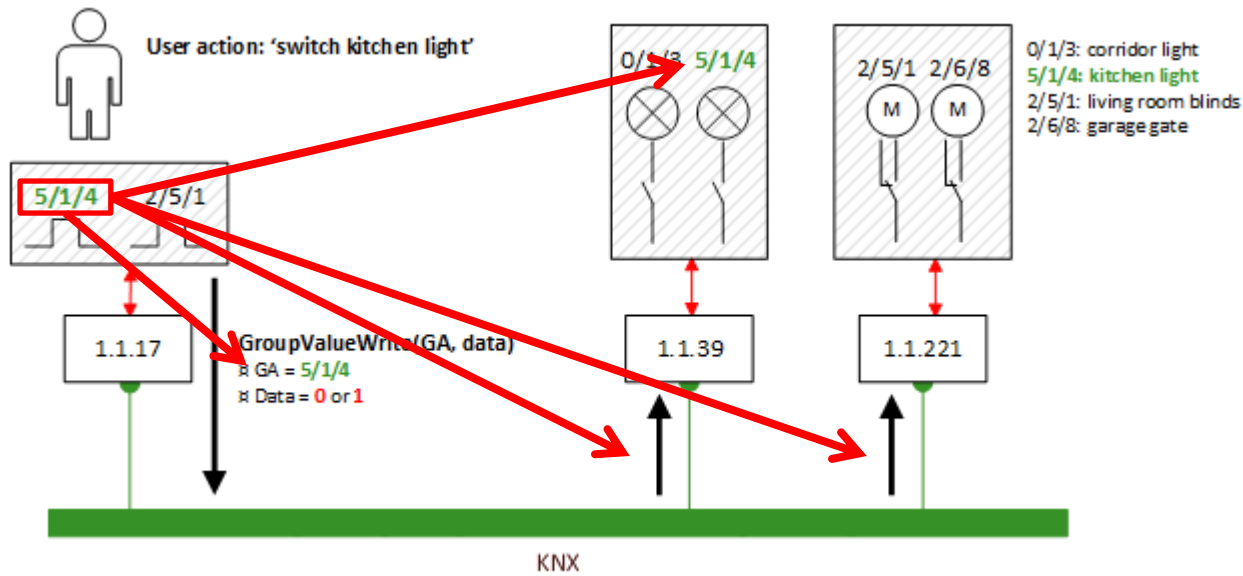


Functions = Group Addresses

- defining Functions = linking Group Objects
- by means of Group Addresses (GA)



KNX telegrams: switch on that light!



Example

- 1.1.17: user manipulation for the 'kitchen light'-rocker
- 1.1.17 broadcasts a telegram(GA, data)
- All sensors & actuators interpret any received telegram
- All actuators addressed by GA react accordingly (here only one)

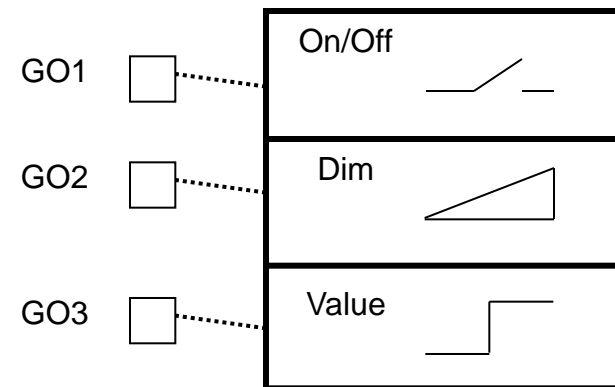
Other KNX functions (FBs)

- **Dimming**
- **Blinds/Shutters**
- ...

Dimming 1/2

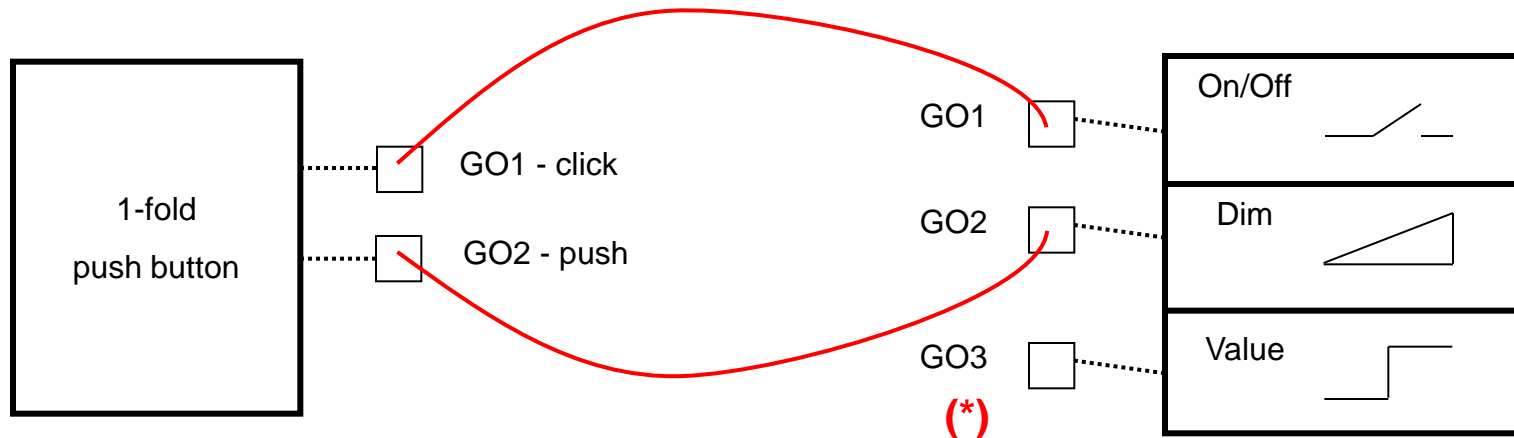
3 Group Objects:

1. On/off: 1-bit (GO1)
2. Dimming: 4-bit (GO2)
 - Up/Down – 1 bit
 - # Step/Stop – 3 bit
 - 000b: stop
 - 001b: 50%
 - 010b: 25%
 - 011b: 12.5%
3. Value: 8-bit (GO3)



Dimming 2/2

- Define the function of the button(s) via the ETS Parameter Dialog
- Link the Group Objects with the Actuator



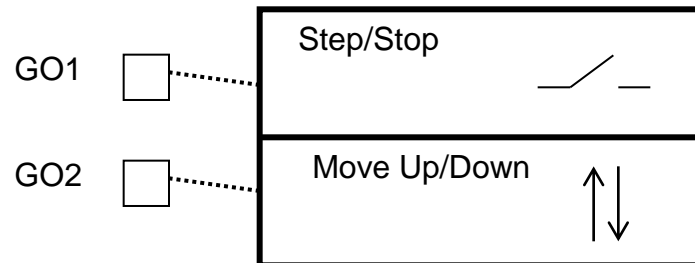
(*) Use cases:

- Scenarios = telegram 'multiplier'
- visualization: read (status) & initialization

Blinds/Shutters 1/2

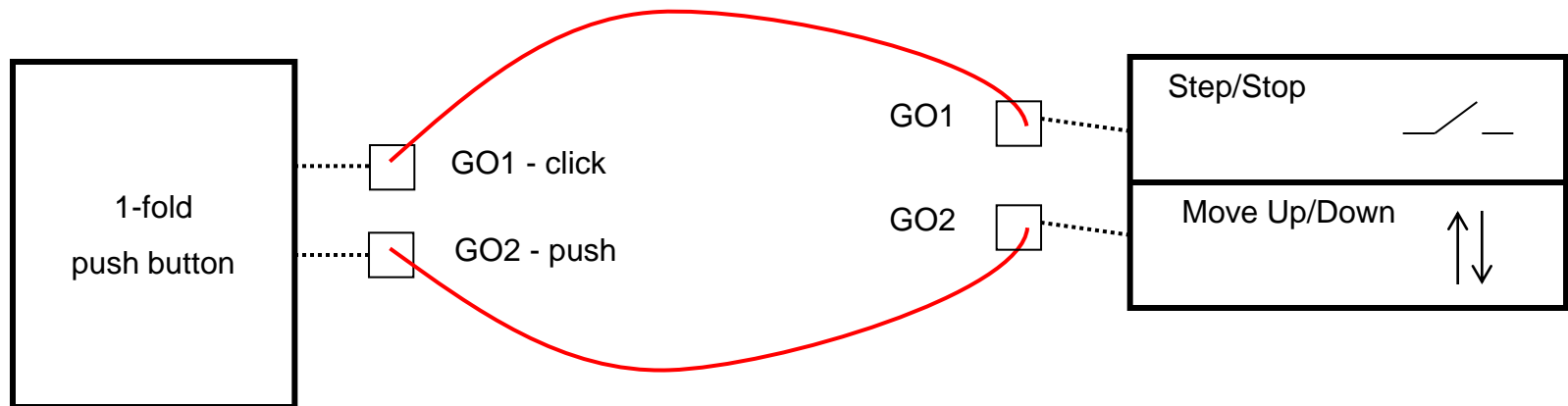
2 Group Objects

1. Step/Stop: 1-bit (G01)
2. Move Up/Down: 1-bit (G02)



Blinds/Shutters 2/2

- Define the function of the button(s) via the ETS Parameter Dialog
- Link the Group Objects with the Actuator



ETS configuration steps

- **Set up individual addresses**
 - by creating the **topology** structure
 - **result:** all devices have a unique network identification

- **Set up individual device functionality**
 - by configuring the **parameters** of all devices
 - **result:** all required group objects are available

- **Set up functions**
 - by assigning **group addresses**
 - **result:** all group objects of all devices are linked according to the requirements

Next recommended steps

- This was only an introduction!!
- eCampus = test ETS by yourself
- Become a KNX partner via a **KNX training center** of your choice.