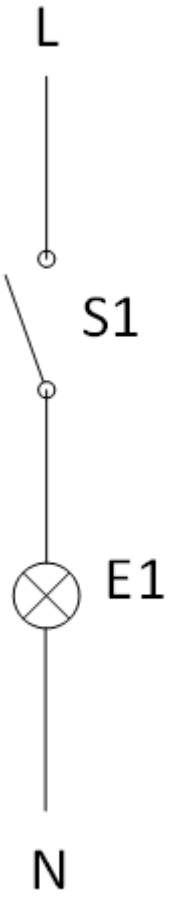
Conventional System: simple switch

- **1 light: E1**
- **1 switch: S1**
  - Closed: current flow
  - Open: current interruption

S1	E1
Open	0
Closed	1



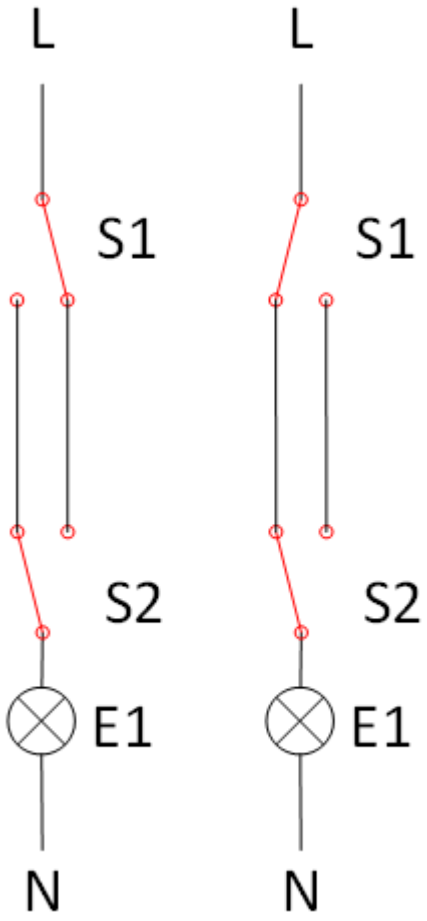
2 toggle switches

- **1 light: E1**
- **2 toggle switches: S1, S2**

S1	S2	E1
Left	Right	0
Left	Left	**1**
Right	Right	**1**
Right	Left	0

S1 & S2:

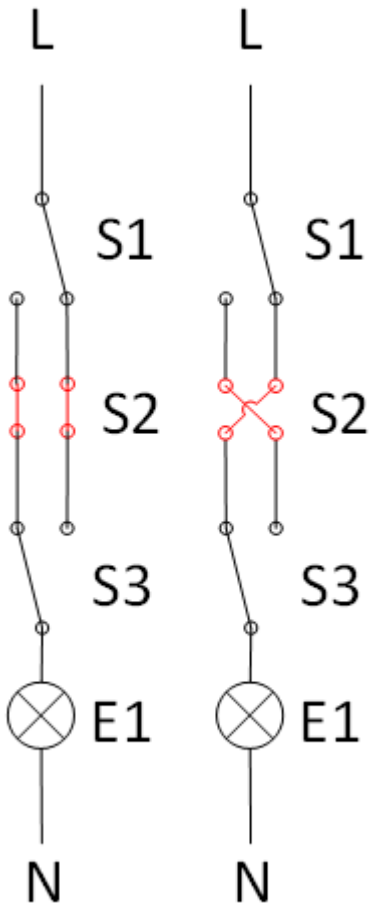
- ➢ on the same side: current flow
- ➢ on opposite sides: current interruption



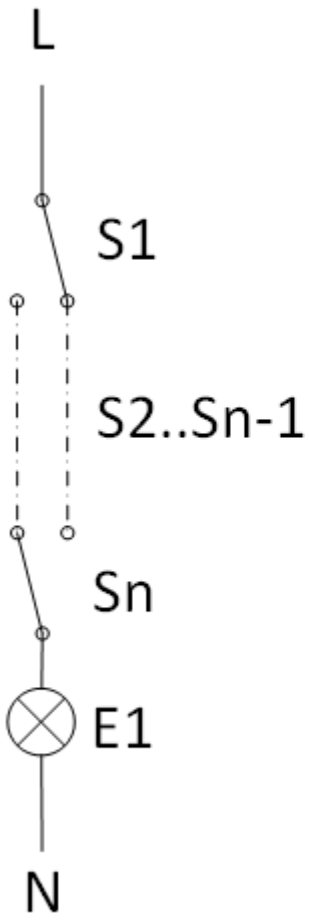
Add a cross switch

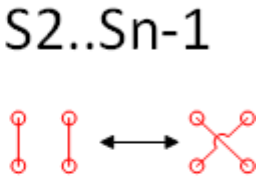
S1	S2	S3	E1
Left	||	Left	**1**
Left	||	Right	0
Left	X	Left	0
Left	X	Right	**1**
Right	||	Left	0
Right	||	Right	**1**
Right	X	Left	**1**
Right	X	Right	0

- **S1, S3 on the same side**
  S2 ‘parallel’: current flow
- **S1, S3 on opposite sides**
  S2 ‘crossed’: current flow



Add more cross switches

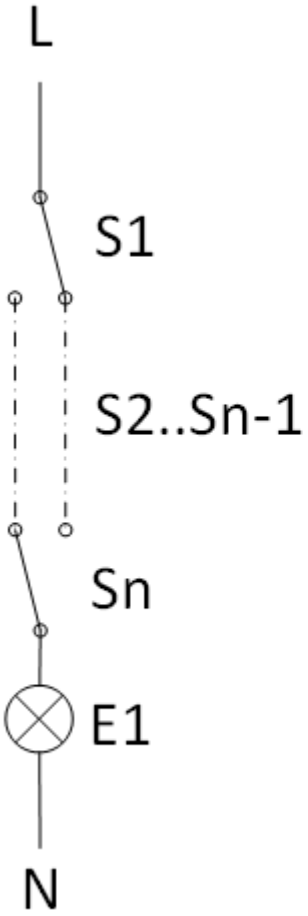


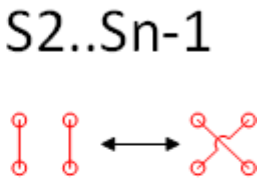


Current flow if:

- **S1, Sn on the same side**
  S2..Sn-1: #‘crossed’ connections = odd
- **S1, Sn on opposite sides**
  S2..Sn-1: #‘crossed’ connections = even

Conventional System





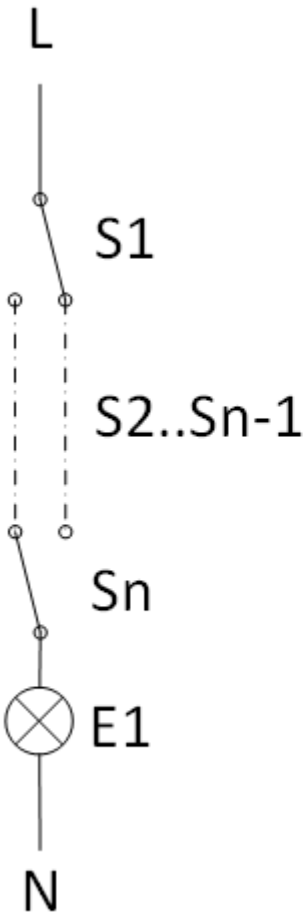
Functionality = hardwired

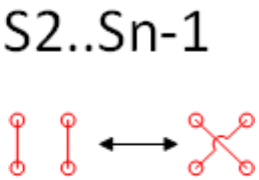
- S1 & Sn: 3 wires
- S2..Sn-1: 4 wires

Disadvantages:

- Cabling = complex = time consuming
- Unfavorable functionality/cable ratio
- Not flexible: adding switches = cumbersome
- No separation between power & operation

Hardwiring vs. ICT



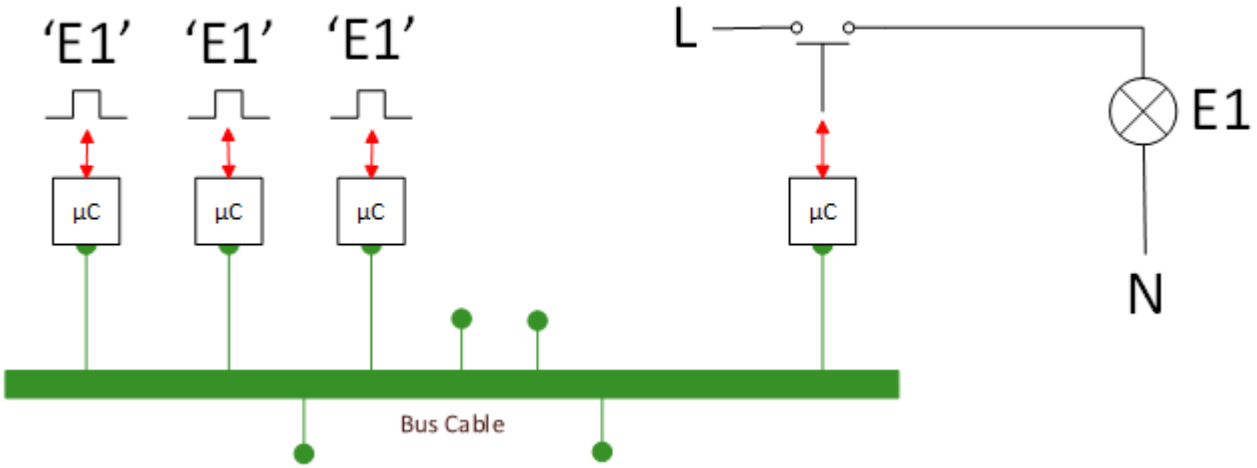


Alternative = ICT

Alternative = bus system

- Replace all switches by µCs
- Link the µCs via cable

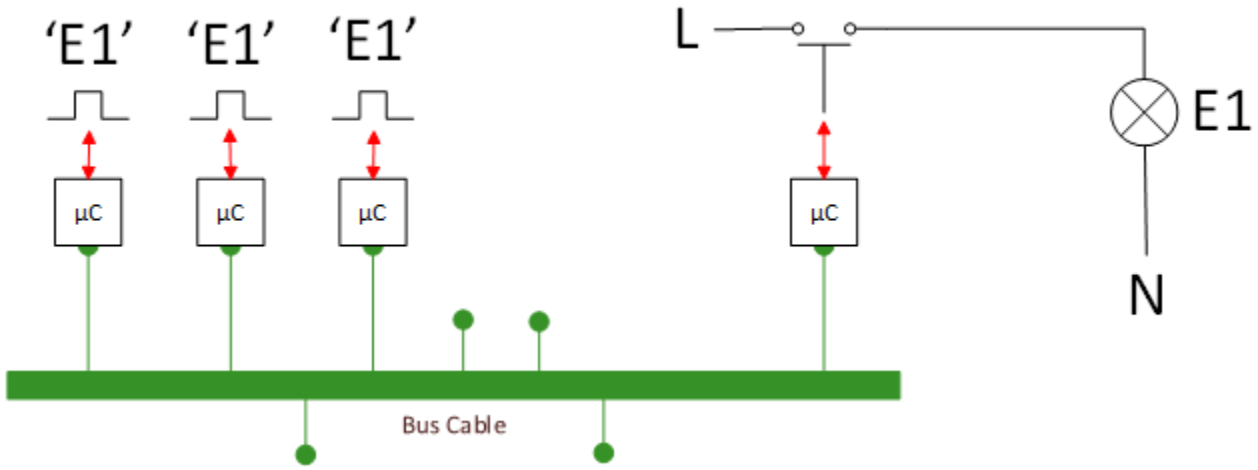
Bus System = divided functionality



- Sensors: µC interfaced with e.g. a push button
- Actuators: µC interfaced with e.g. a relay

- Interfaced with = electronically connected to µC I/O ports
- Push buttons convert rocker manipulations into electronic pulses
- Relays switch electrical consumers indirectly via electronic pulses

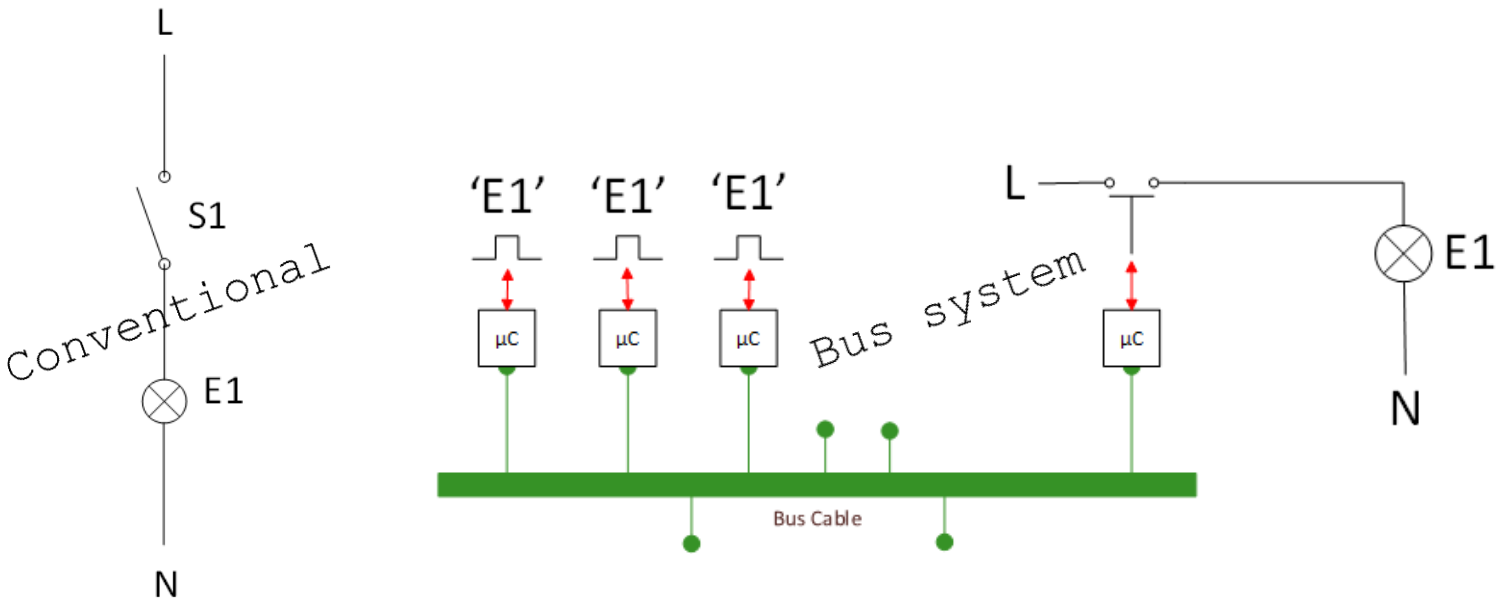
Bus System = exchanging messages



Example

- At one of the push buttons: user manipulation for the 'E1'-rocker
- This sensor broadcasts a message for each such user manipulation
- Each such message has two parameters:
  - Address = 'E1'
  - Value = either 0 or 1
- All sensors & actuators interpret any received message
- All actuators addressed by 'E1' switch E1 accordingly (here only one)

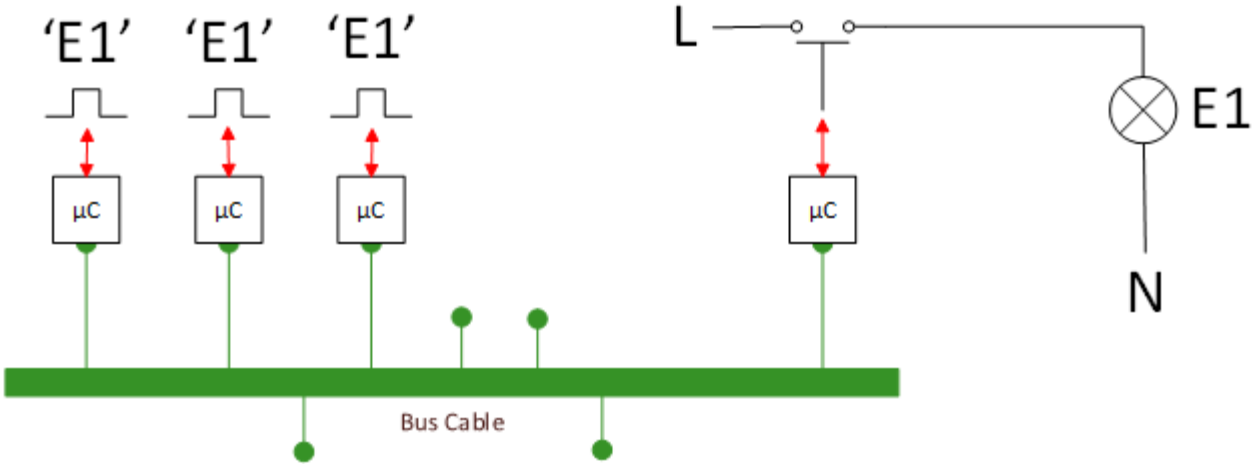
Connections: physical vs. virtual



Connection between operating elements & electrical consumers

- **Conventional system: directly & physically linked via wires**
- **Bus system: indirectly, virtually linked via address parameters in messages exchanged between µCs**

Bus System: practical advantages

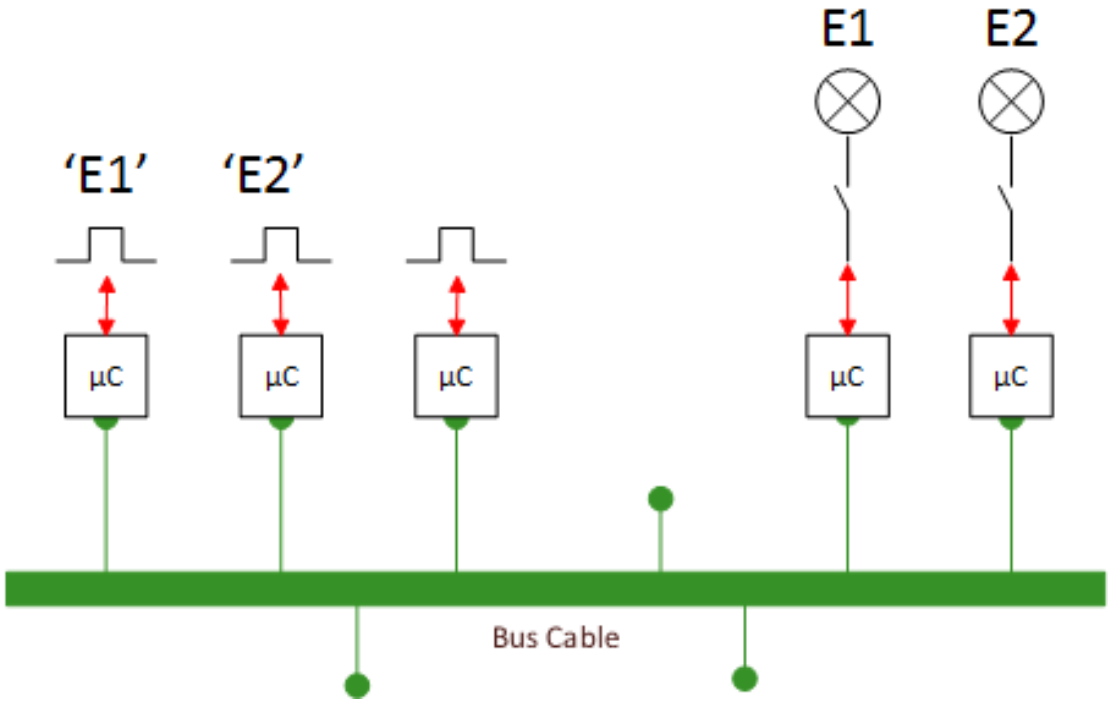


- Cabling = easy = time saving = cheaper
- Optimal functionality/cable ratio
- Flexible: add more sensors and actuators
- Separation between power (230V) & operation

Add more lamps…

The same bus can of course be used for additional lamps

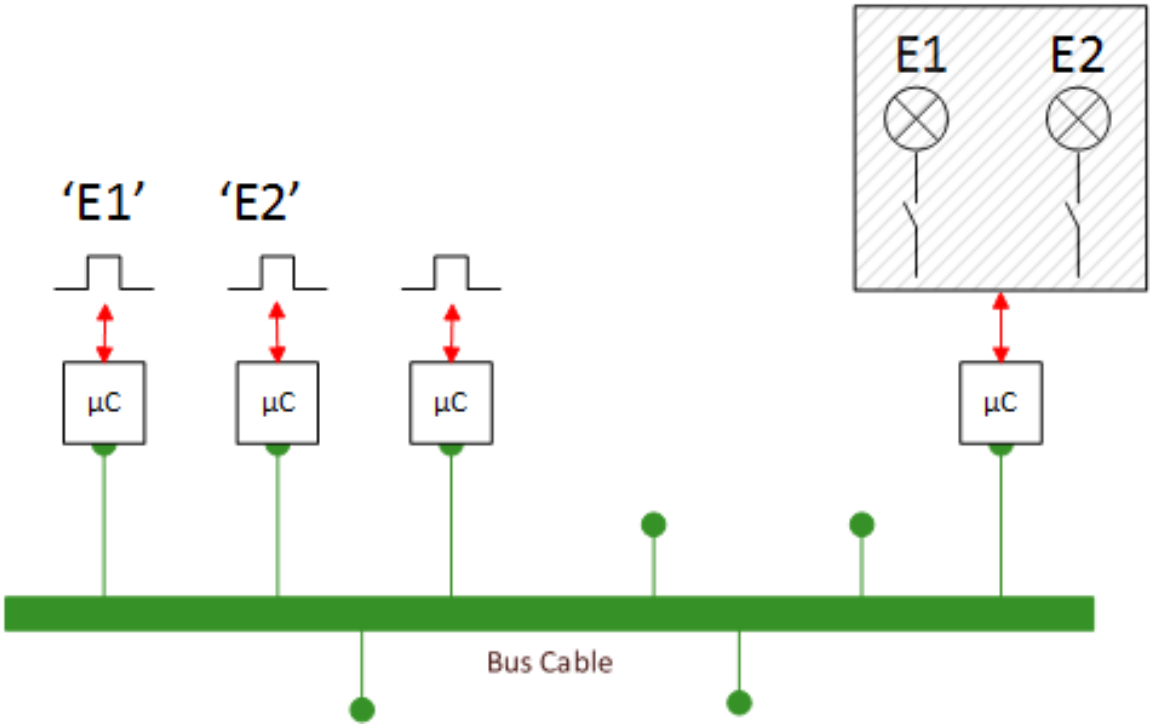
- sensor µC ↔ E1 pulses ➔ actuator µC ↔ E1 relay
- sensor µC ↔ E2 pulses ➔ actuator µC ↔ E2 relay



Actuators: combine lamps

One µC can control several electrical consumers

- sensor µC ↔ E1 pulses ➔ actuator µC ↔ E1 & E2 relays
- sensor µC ↔ E2 pulses ➔ actuator µC ↔ E1 & E2 relays

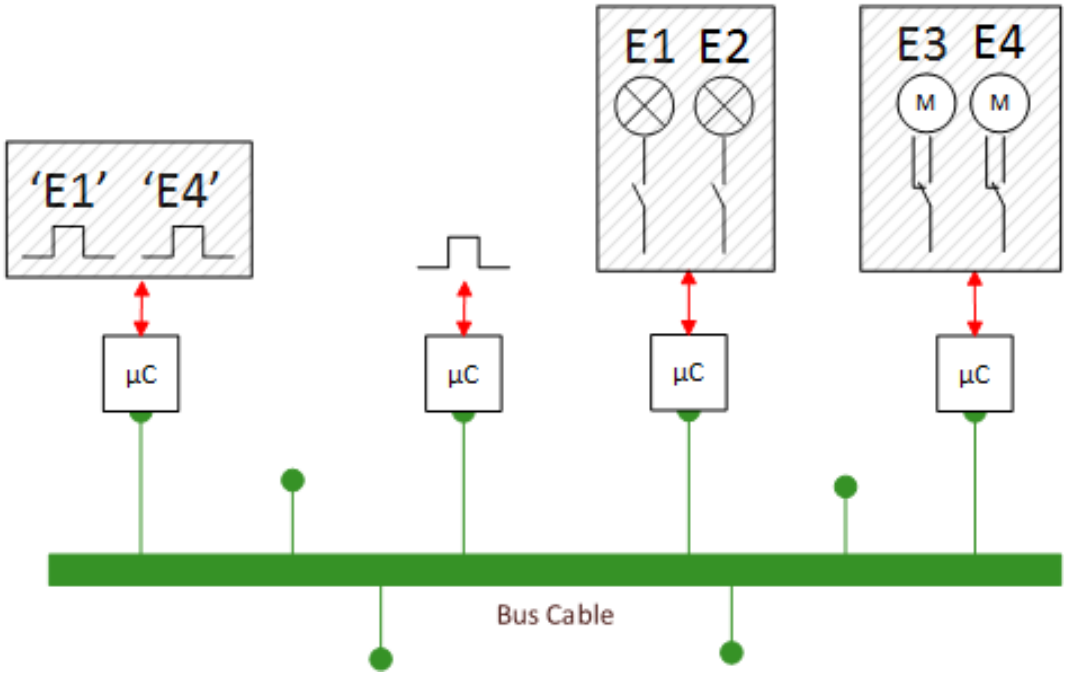


Sensors: combine function types

Sensor µC ↔ E1 & E4 pulses

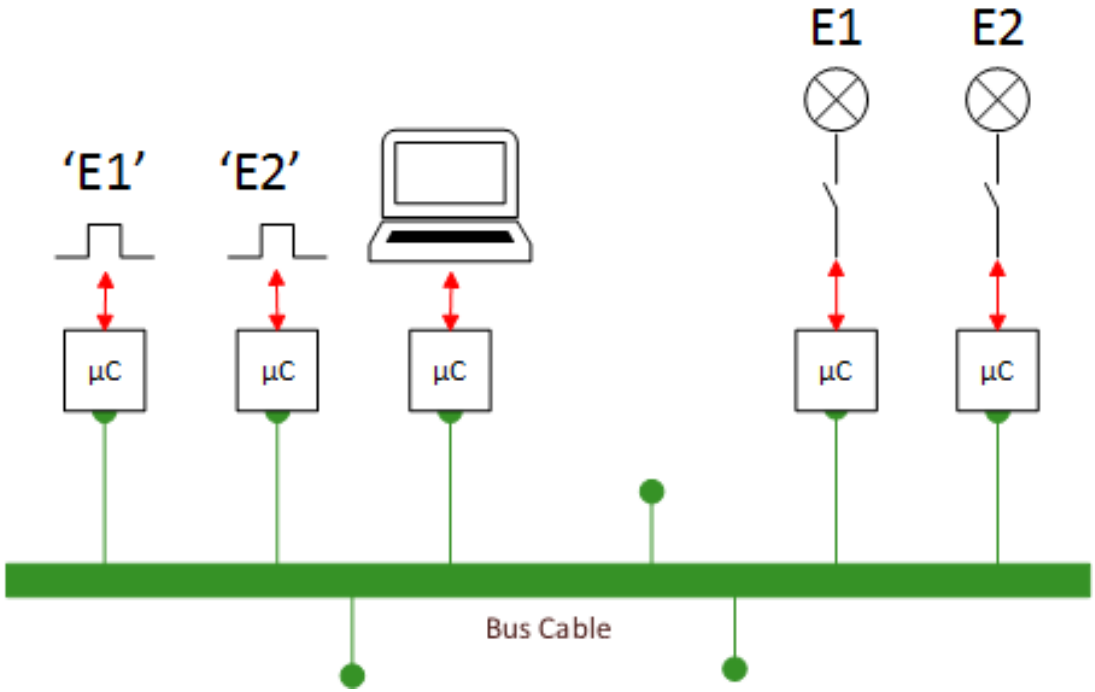
➔ actuator µC ↔ E1 & E2 relays: control **light** E1

➔ actuator µC ↔ E3 & E4 relays: control **blind** E4



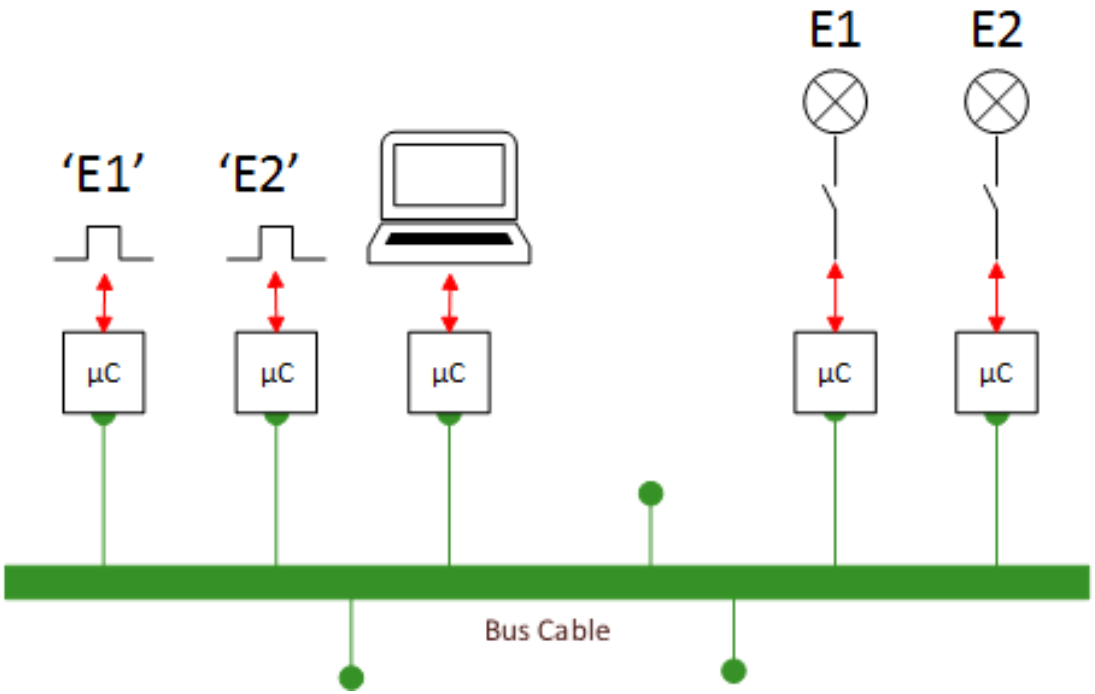
Bus System: easy modifications…

- Sensor & actuators exchange runtime messages
- Interfaces exchange management & diagnostic messages
- Interface = µC connecting the bus to a PC



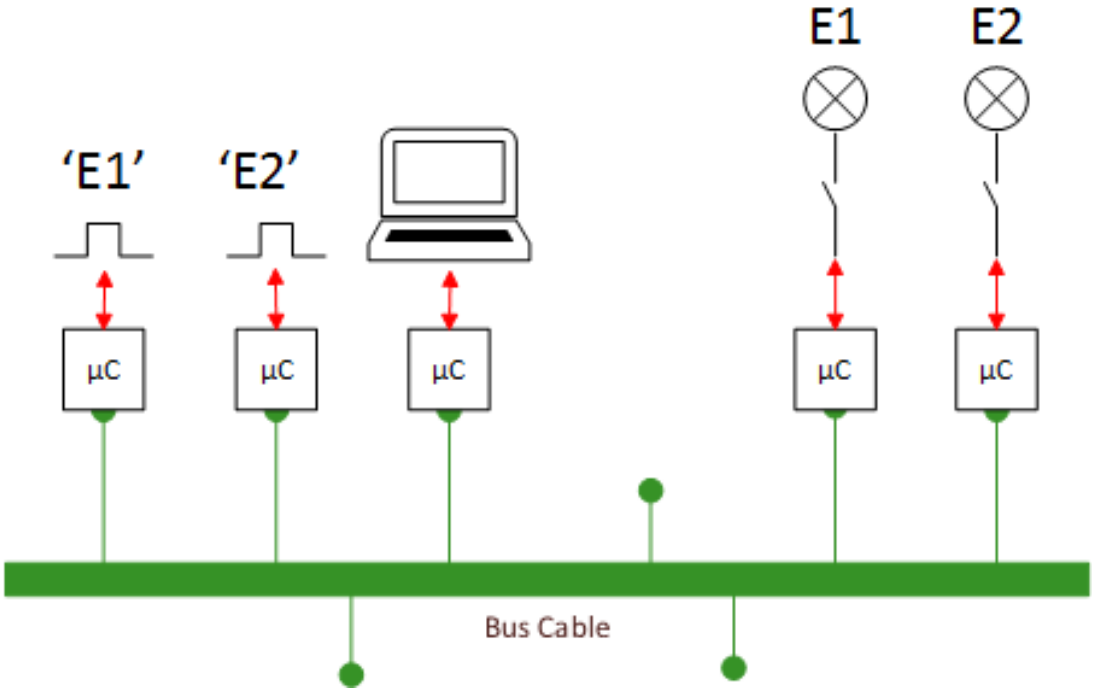
(re-)programming vs. rewiring

- µC functionality = µC memory image
- Changing the µC functionality = changing the µC memory image
- Software tool + PC + interface ➔ change µC memory images



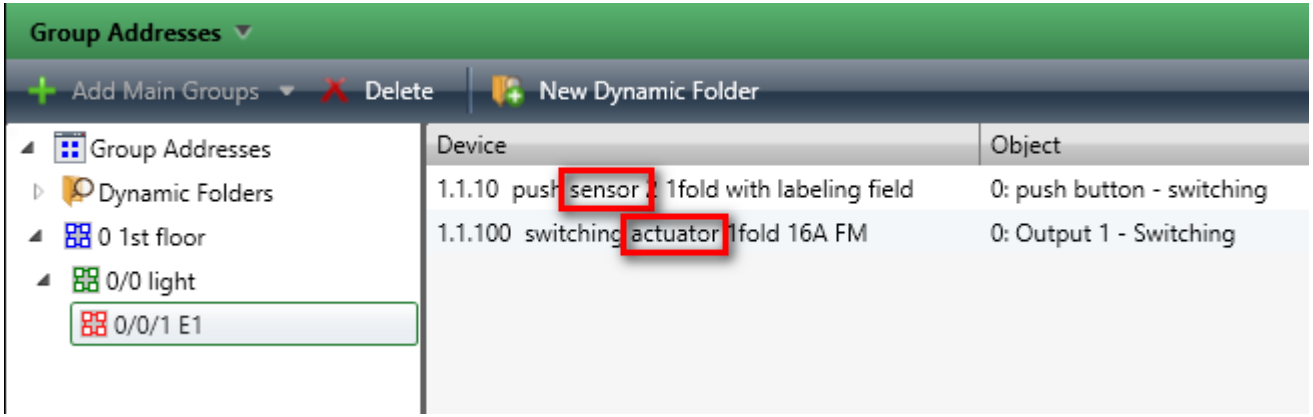
With a software tool it's easy to…

- **Fine tune:** e.g. stairway function for E2: automatic (delayed) switch off
- **Modify:** e.g. sensor shall serve E2 instead of E1
- **Add:** e.g. sensor already serving E1 shall also serve E2
- **Extend:** e.g. add a sensor to the installation



KNX + software tool = ETS (*)

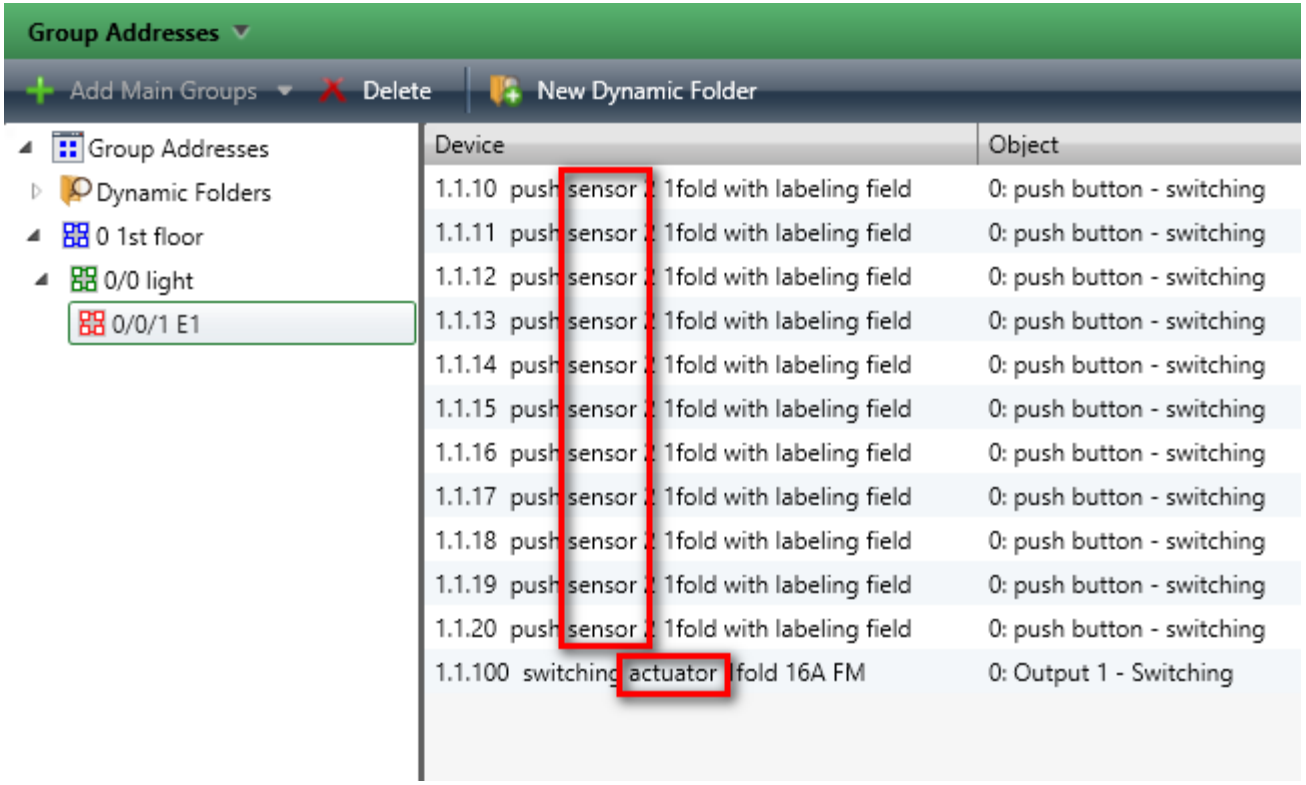
- Remember: bus system = sensors & actuators virtually linked
- KNX + virtual link = group address
- E1: group address = 0/0/1
- Sensor: device 1.1.10
- Actuator: device 1.1.100



(*) More details in ‘KNX Basics’ webinar: don’t hesitate to subscribe!

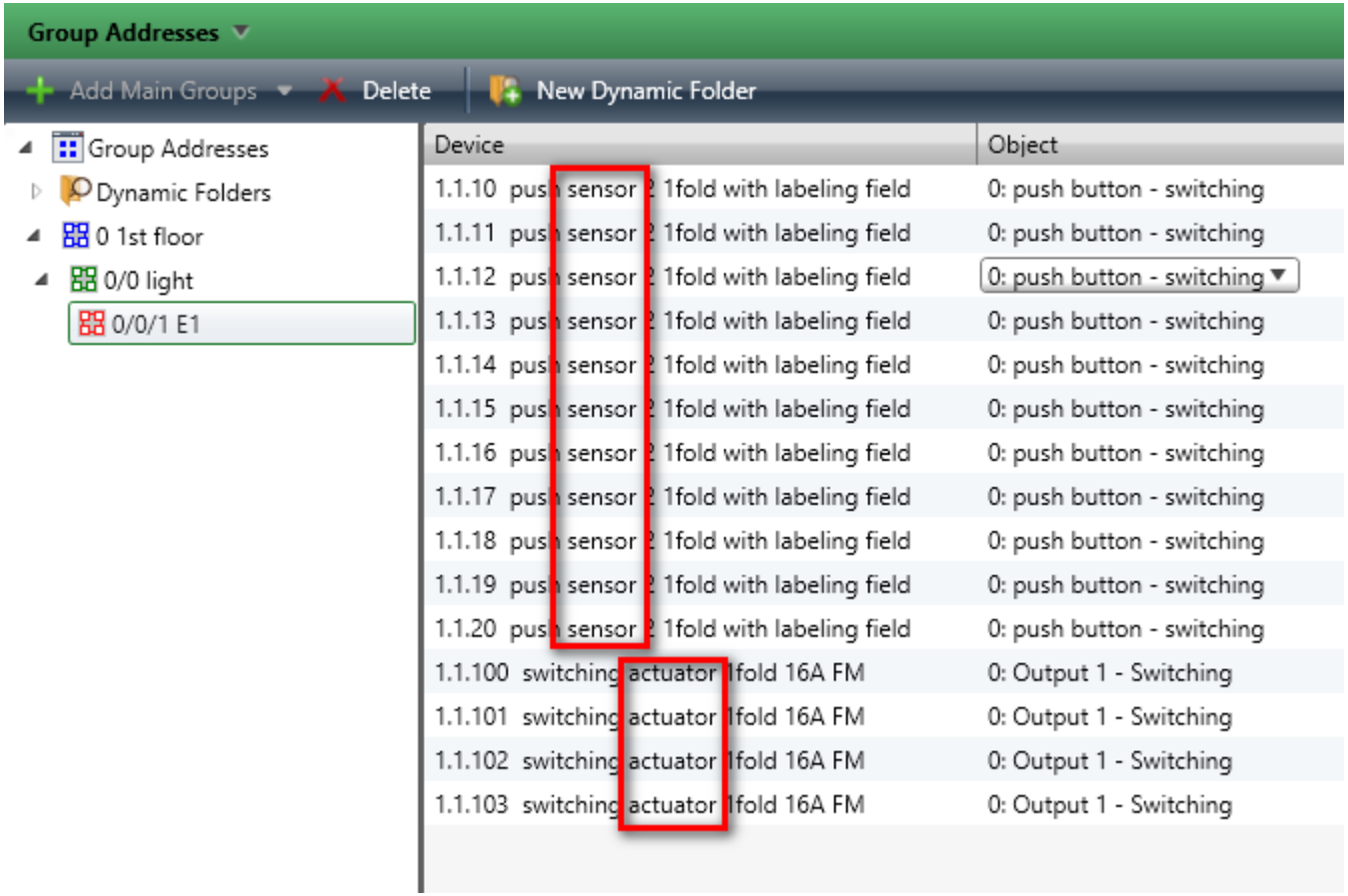
ETS: add more sensors to E1

- E1: group address = 0/0/1
- Sensors: devices [1.1.10]..[1.1.20]



ETS: add more actuators to E1

- E1: group address = 0/0/1
- Actuators: devices [1.1.100]..[1.1.103]



KNX: the future proof bus system

KNX = quality

- ISO 9001 = prerequisite for product certification

KNX = compatibility

- Compatibility = manufacturer independent
- Assured via product certification

KNX = backwards compatibility

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

KNX = saving energy

- Today: window contacts, presence detection, etc.
- Tomorrow: link KNX installations to smart grids (KNX city)

KNX = the reference bus system

KNX = combining all possible building application types

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

KNX = combining communication media

- **TP: the base**
- **PL: renovation**
- **IP: high speed and secure backbone**
- **RF: flexibility and self-sustaining sensors** (no batteries)

KNX = the reference bus system

KNX = linking to 3rd party systems

- Visualization
- Gateways

KNX = one software tool only

- Software tool: ETS
- ETS compatibility is assured via product certification