



Smart home and building solutions.  
Global. Secure. Connected.

## KNX RF development with Weinzierl solutions

Dr. Thomas Weinzierl  
Weinzierl Engineering GmbH  
[www.weinzierl.de](http://www.weinzierl.de)

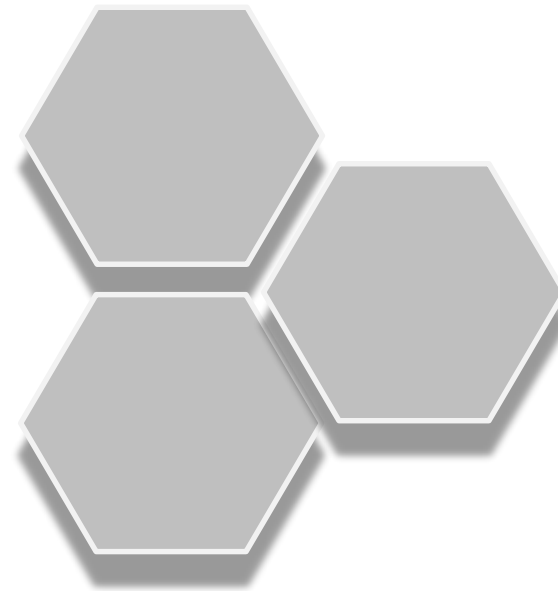
# WEINZIERL



## Agenda

---

- About Weinzierl
- KNX System Overview
- KNX RF with ETS
- KNX BAOS Modules
- KNX Stack NGS
- Demo with Net'n Node



## Weinzierl Engineering GmbH

---

- Founded 2001
- Burgkirchen / Alz
  - South East of Germany
- System solutions for KNX
  - KNX Stacks & Modules
  - KNX Development Tools
- Testing
  - KNX accredited Test Lab
- Production
  - KNX Devices



## KNX RF History

---

- Since 2002
  - Medium RF specified
  - Mainly for Easy Mode, no ETS
  - Only a few manufacturers
- Since 2014
  - For System Mode
  - Now in-line with TP & IP
  - Fully integrated in ETS
- Since 2017
  - KNX Security
  - Growing number of products



## Application Area

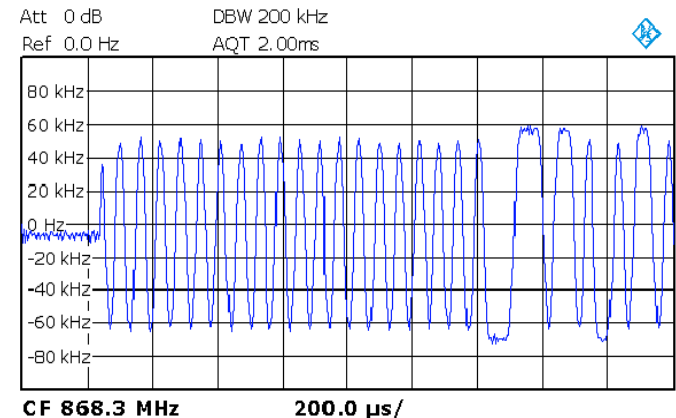
---

- KNX wireless devices
  - Sensors
  - Actuators
  - Remote control
  - ...
- RF-only Installations
  - Renovation market
- Hierarchical Installations
  - RF combined with TP and IP
  - In one ETS project

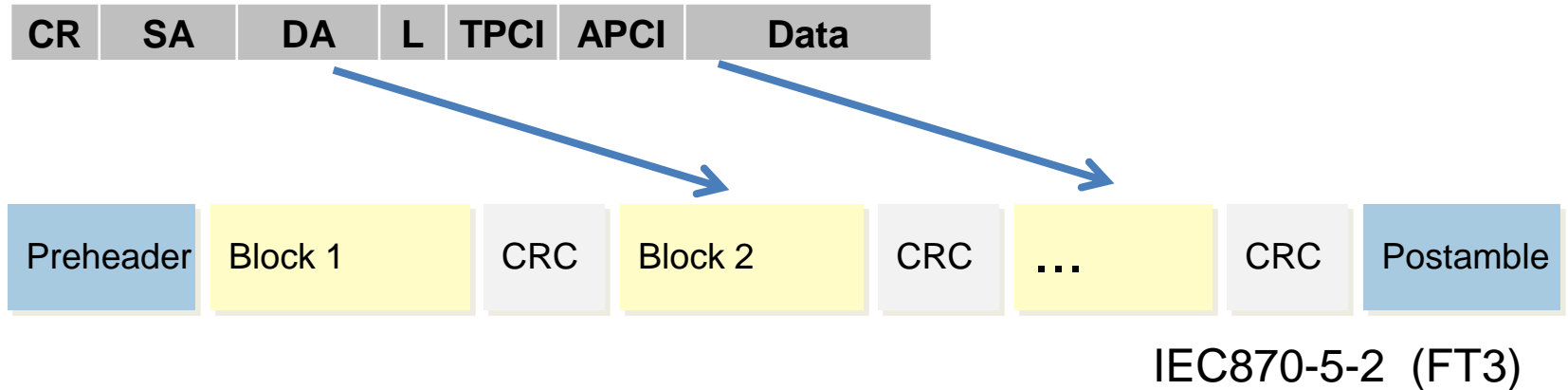


## KNX Radio Frequency - RF

- Short range device
  - ISM Band 868 MHz
  - FSK Modulation
- Good range within buildings
  - Very reliable
  - No Mesh required
  - Re-transmitters possible
- No dedicated KNX transceiver required
- Low cost production



## Frame format KNX RF ready



- KNX Data in FT3 block format
  - Preheader 176 chips
  - First block 10 bytes (fixed length), RF specific (SN / DoA)
  - Further blocks 16 bytes (variable length), KNX telegram
  - Last block 1 to 16 bytes
  - CRC 2 bytes CRC after each block
  - Postamble 2 to 8 chips, end of frame

## Addressing Modes

- Destination Address on KNX TP

Destination Address		Addressing Mode
IA	DAF = 0	Individual Address
GA	DAF = 1	Group Address
GA = 0x0000	DAF = 1	Broadcast

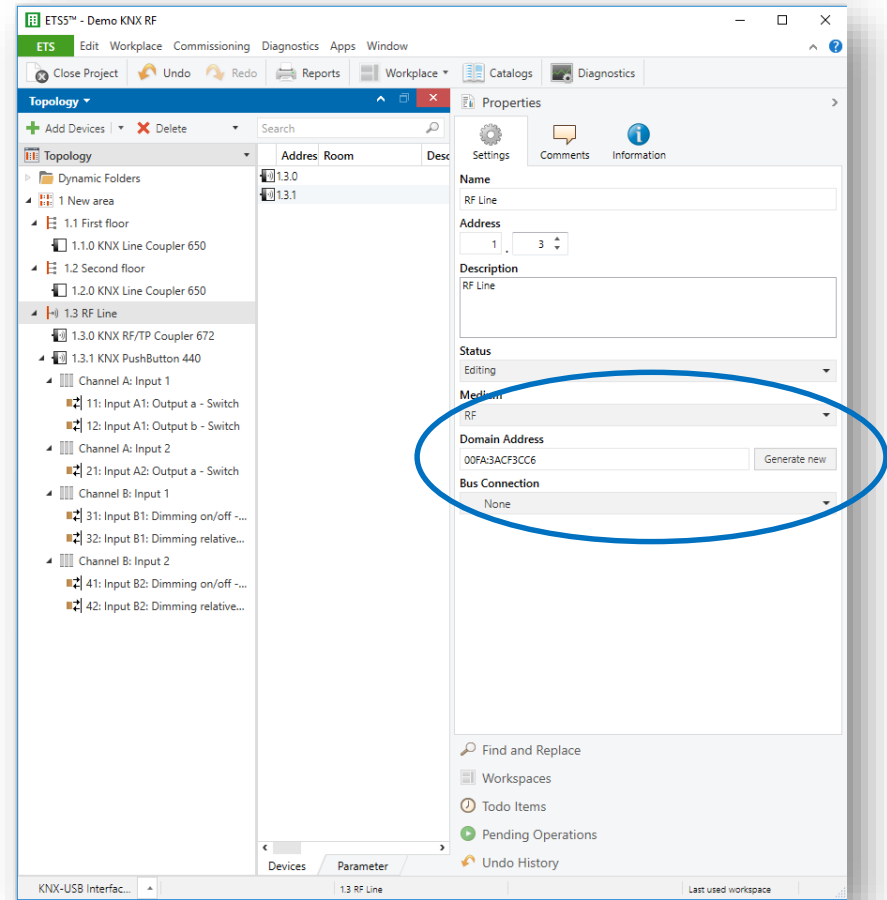
- Destination Address on KNX RF

Block 1	Block 2, Dest. Address		Addressing Mode
SN	GA	DAF = 1	Ext. Group Address (PB Mode)
DoA	GA	DAF = 1	Group Address
DoA	IA	DAF = 0	Individual Address
DoA	GA = 0x0000	DAF = 1	Broadcast
SN	GA = 0x0000	DAF = 1	System Broadcast



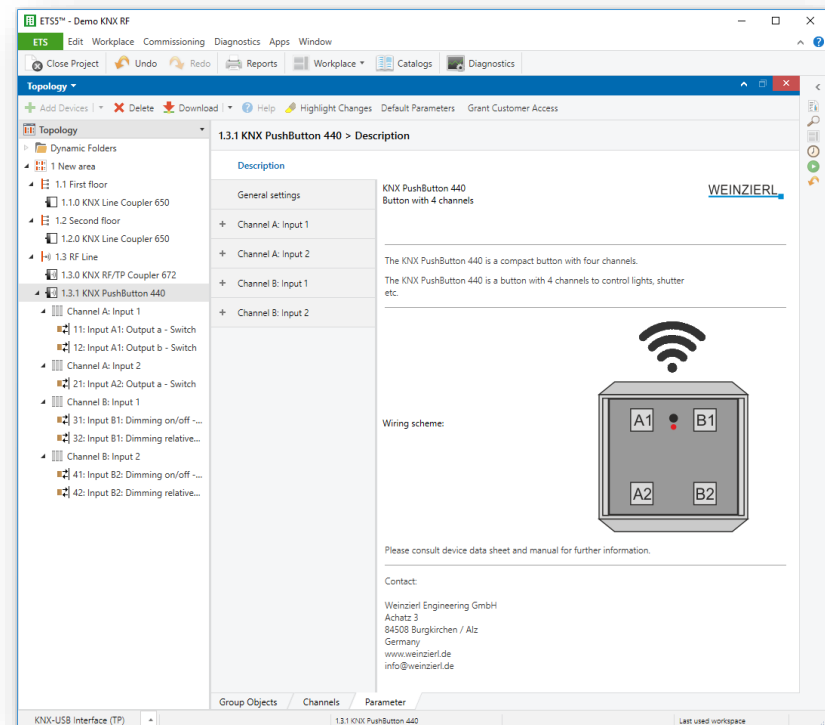
## ETS & RF

- What's new
  - Lines for RF
  - Domain addresses
  - Connection via USB / RF Interface
  - RF / TP Coupler
  
- What's unchanged
  - Group Objects
  - Group Addresses
  - Parameters
  - Download



## ETS – Group Objects & Parameters

- Group Objects
  - Linked via group addresses
  - Configured by flags
- Parameters
  - Device configuration
  - Set in parameter dialog
- Download
  - Via KNX network
  - Via USB / RF Interface
  - Via TP and RF / TP Coupler
  - Local USB connection



## Topology: RF-only Installation

- Minimal RF Setup
  - KNX RF Devices
    - 2..255
  - KNX Interface to PC
    - USB, optional
    - For commissioning only



DVC 1

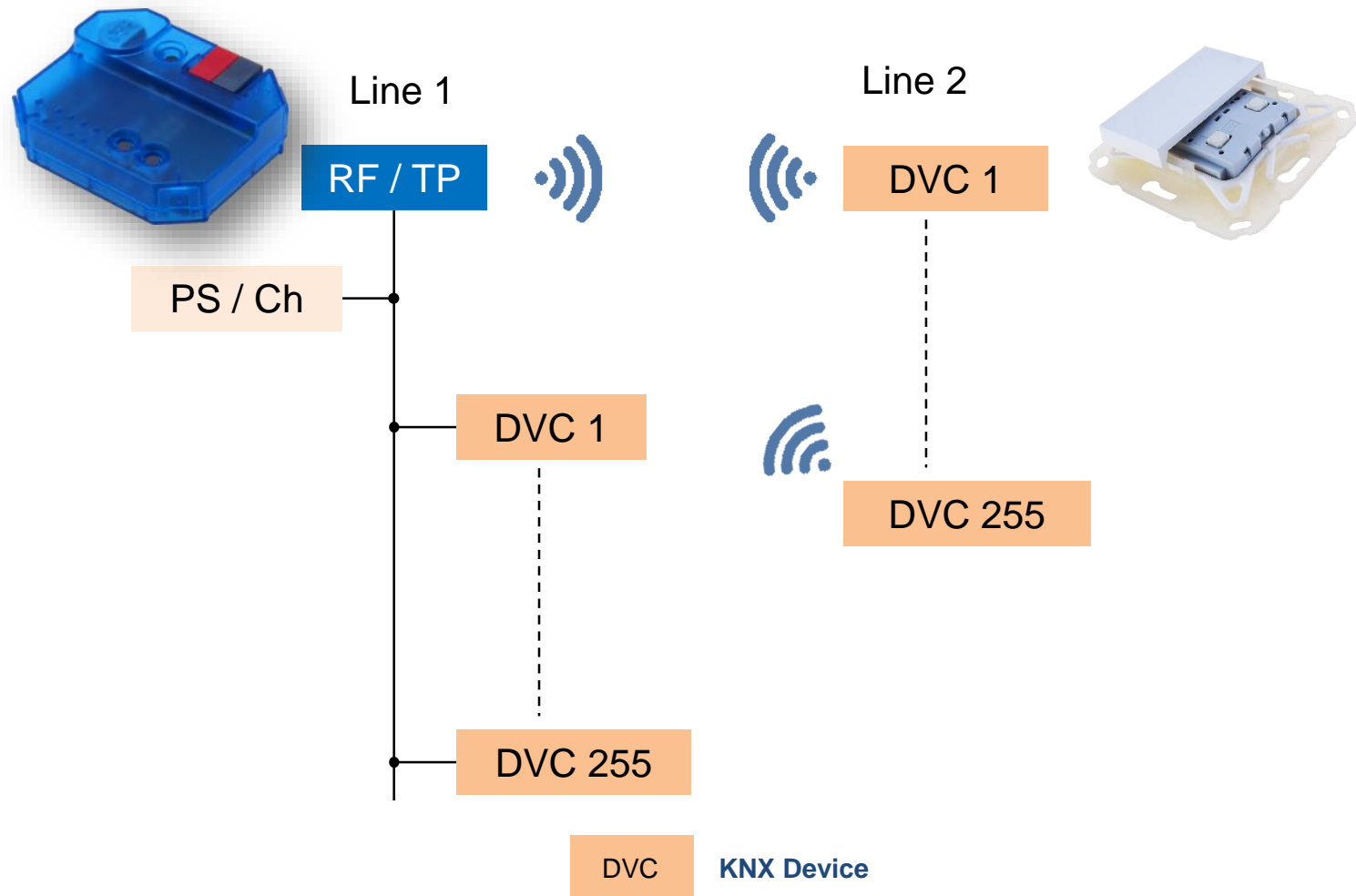


DVC 255

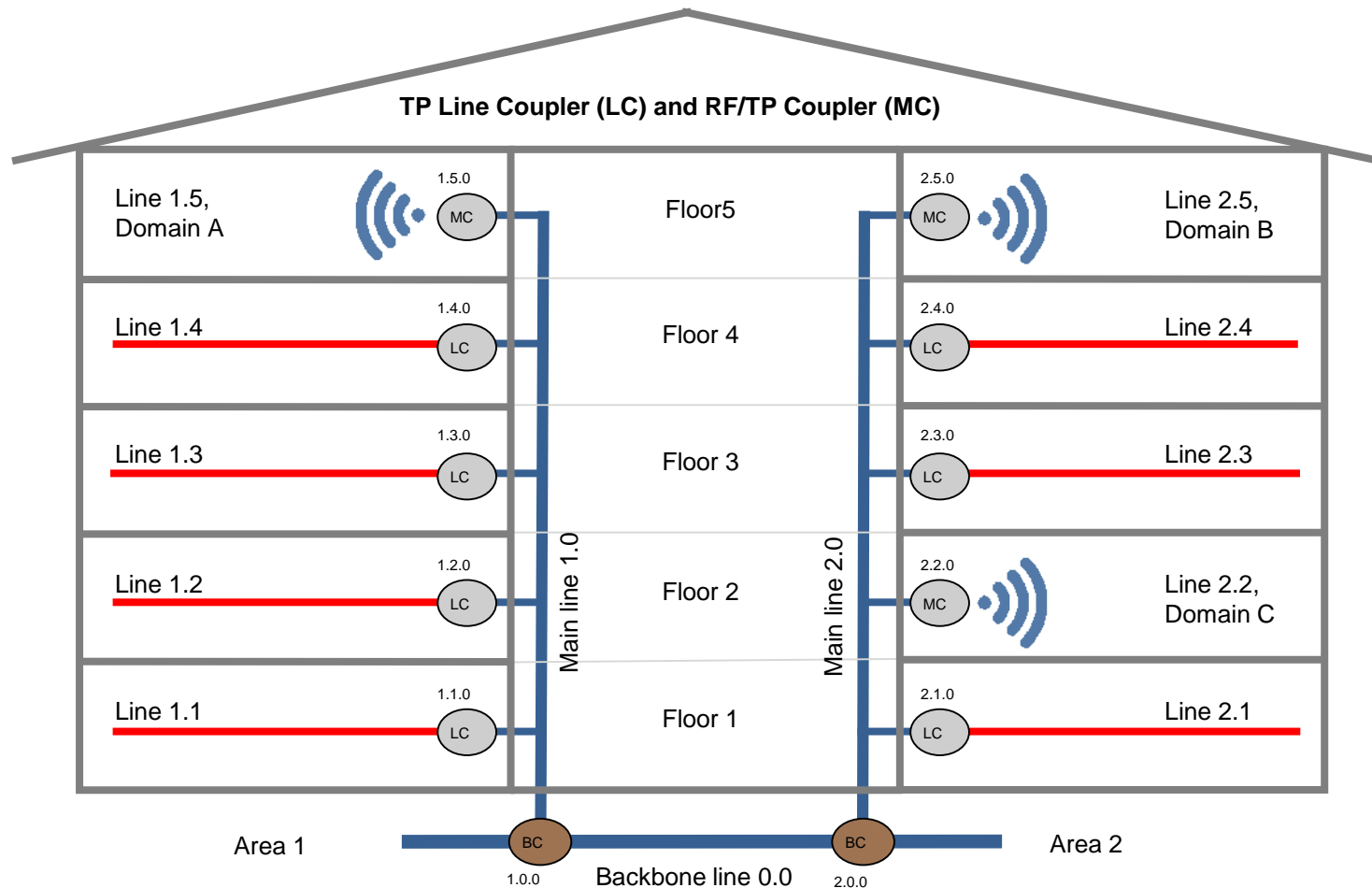
DVC

KNX RF Device

## Topology: TP line + RF line



## Topology: Structure in building



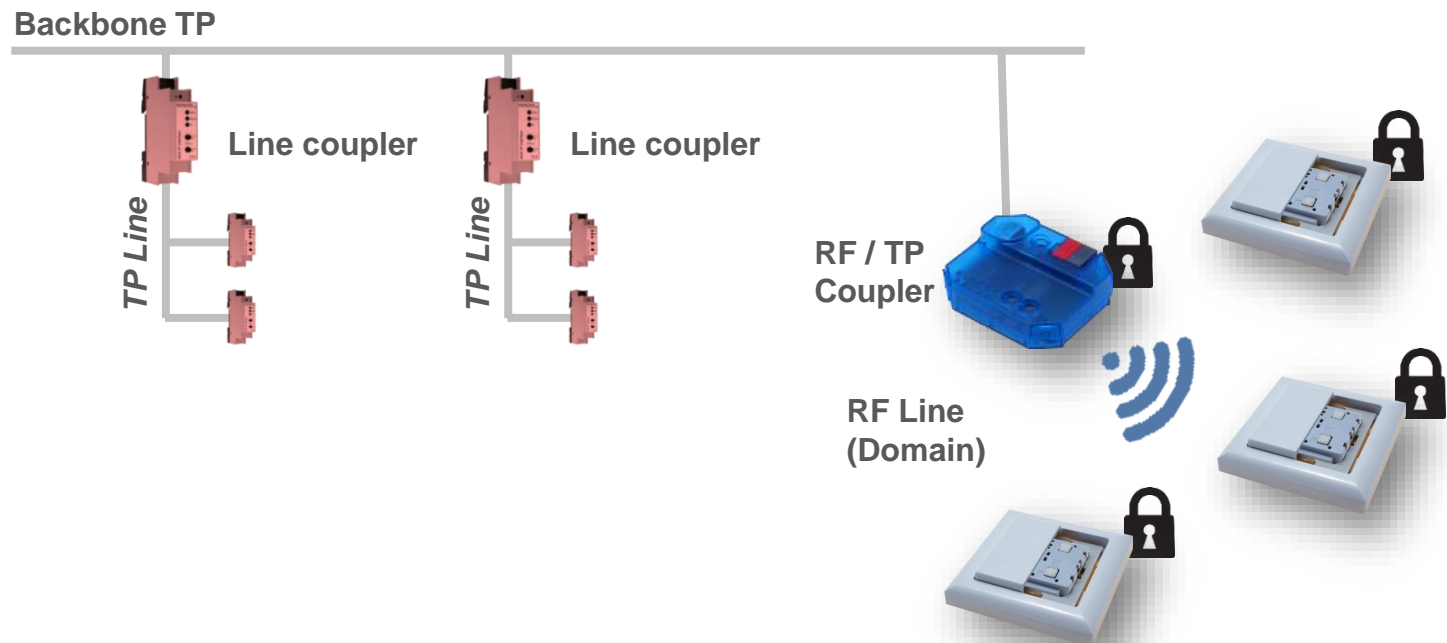
## Further aspects about KNX RF

- Semi-bidirectional devices
  - For battery driven sensors
  - Bidirectional for commissioning
  - Unidirectional during runtime
- Segment coupler
  - Extend TP line with RF/TP Coupler
- Secure Proxy
  - Connect secure and unsecure segments



## Mixing secure and non-secure Communication

- Security per link
- Security in sub-systems
  - KNX Secure Proxy
  - Supported by ETS



## KNX Security

---



- Based on standard security technologies
- Based on AES
  - Advanced Encryption Standard
  - Block length: 128 bits
  - Key length: 128 bits
- Challenge: KNX System aspects
  - E.g. group addressing



## KNX Data Security: Frame Format



KNX Frame with Authentication and Encryption

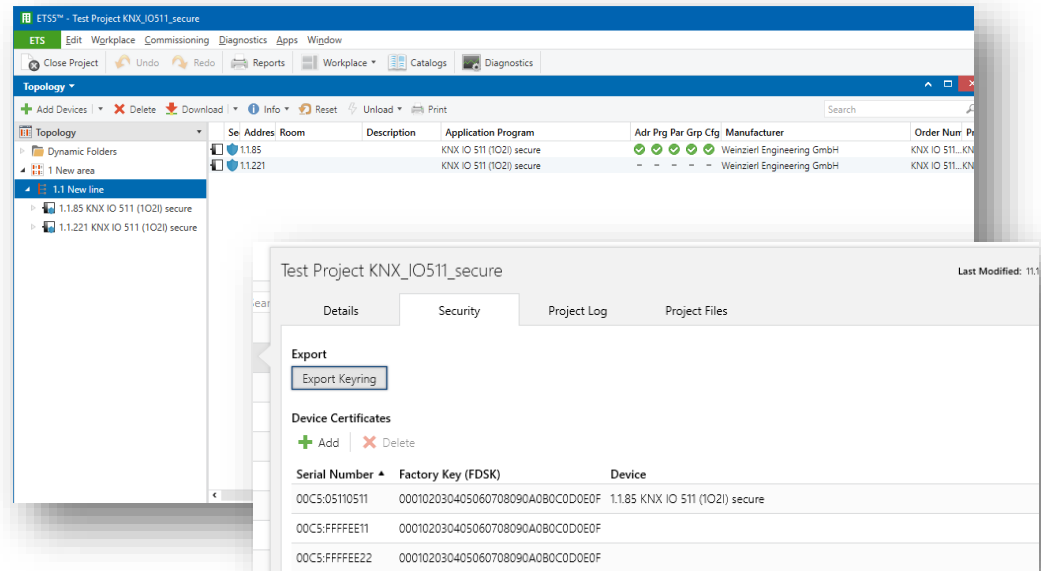
## KNX Data Security: Keys & Counters



- FDSK (Factory default setup key)
  - Set by manufacturer, shown as QR-Code
  - Reactivated after master reset
- Tool key
  - Replace FDSK
- Group keys
  - Runtime communication
- Sending counter
  - One per device
  - Persistency required

## KNX Security in ETS

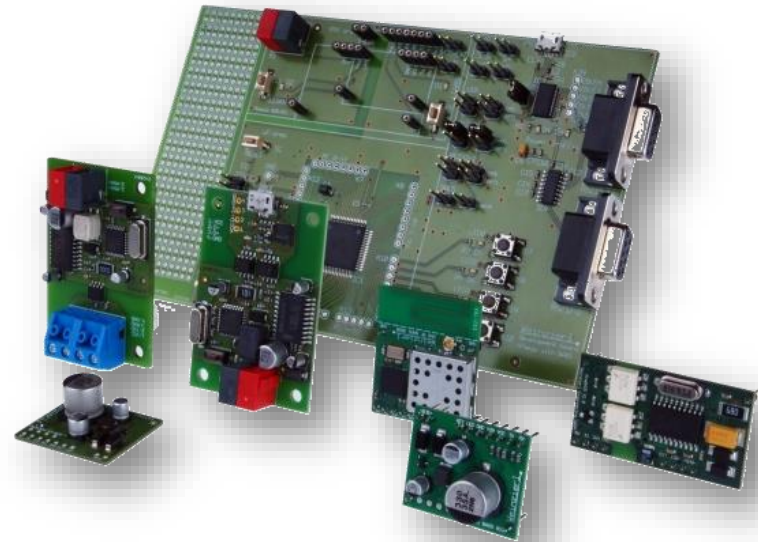
- Fully Integrated since ETS5
- Using camera for QR-Code
- Secure configuration
- Secure Download



## KNX Development

---

- Development solutions for KNX manufacturers
- KNX BAOS Modules
  - KNX Stack on HW
- KNX Stack NGS
  - Full integration
  - Scalable
- Tools
- Services
  - Development
  - Test Lab



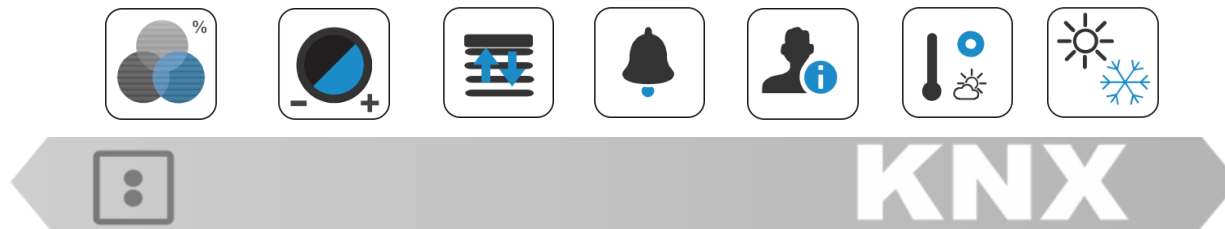
## Development with BAOS solutions



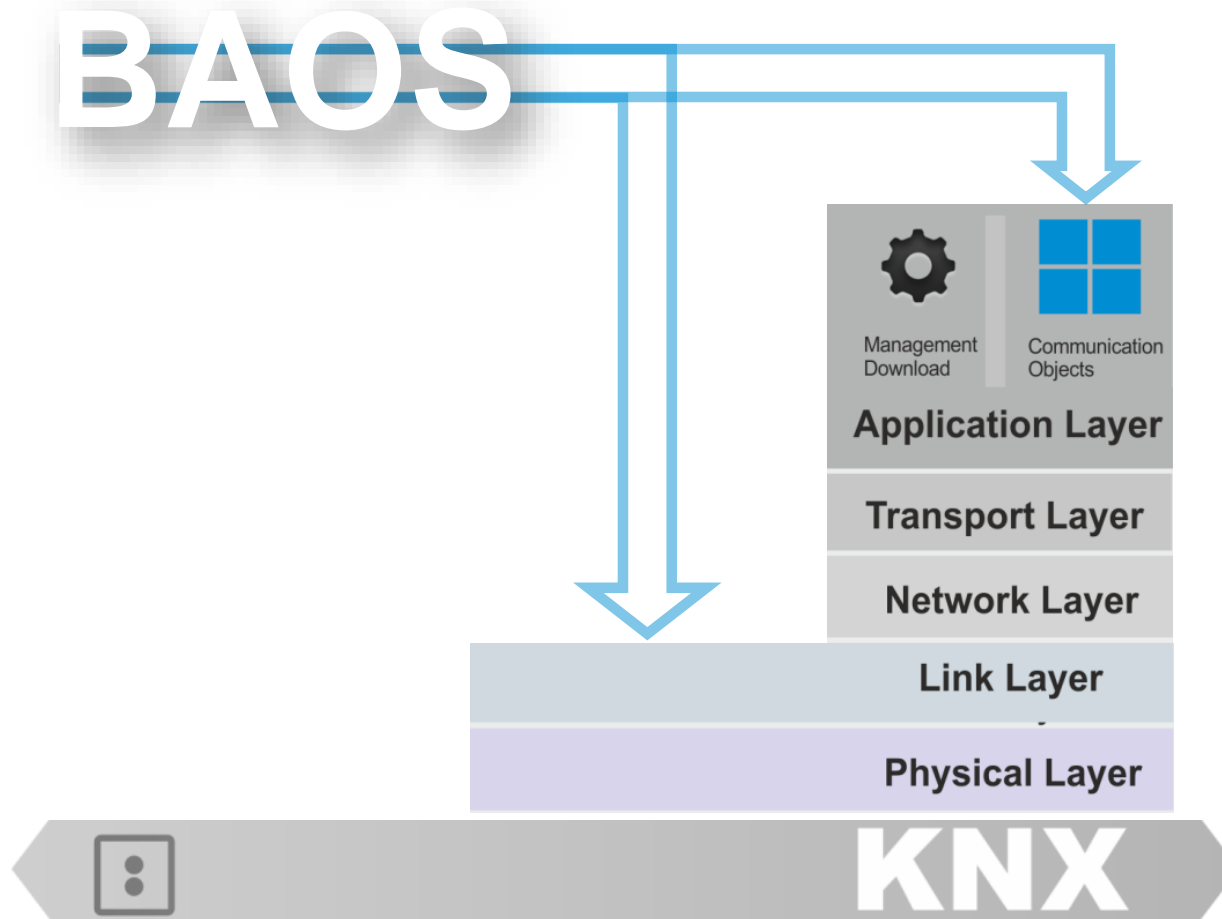
- Bus Access and Object Server
- Bus Access
  - Telegram level
  - Programming interface for the ETS
- Object Server
  - Object Level
  - Allows a fast integration of non-KNX devices into a KNX network
- Host interface
  - Serial, USB, IP

## KNX BAOS Modules: Application Area

- When using a module?
  - Low and medium quantities
  - Extension for existing devices
- Benefits
  - Fast time to market
  - Low impact to application
  - Low investments
  - Certified system software
- ETS product database
  - Available as generic
  - Individualization possible
  - Up to 1000 data points

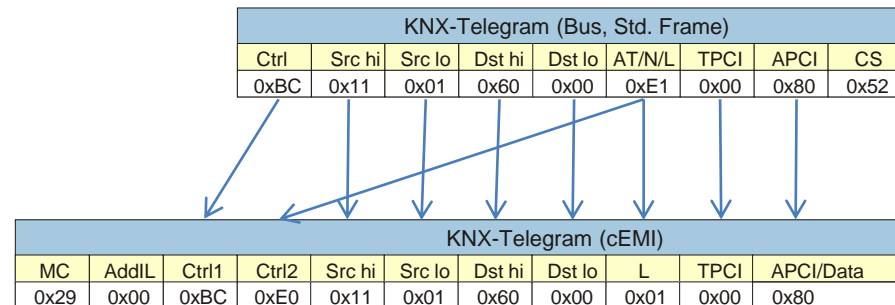


## Access on different layers



## Telegram Access: common EMI

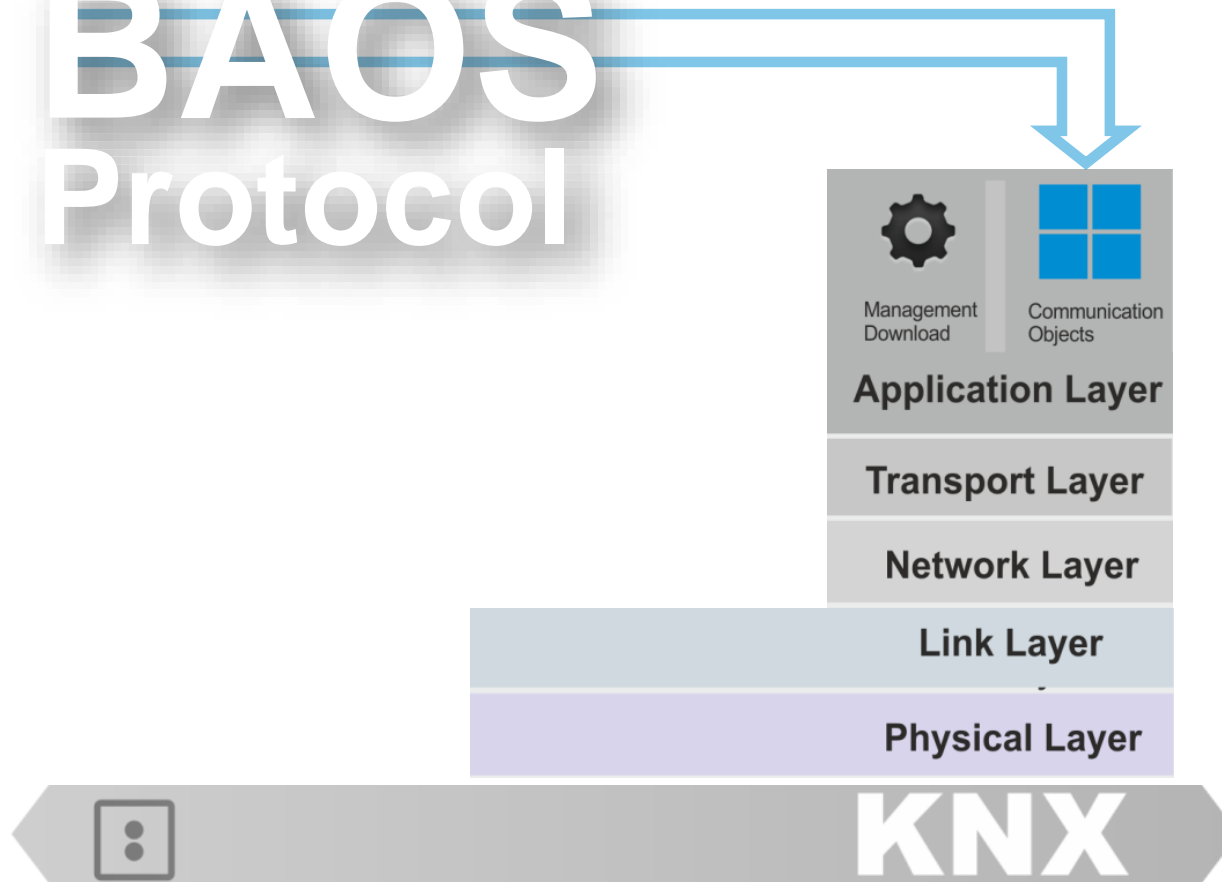
- Send and receive KNX telegrams
- Access to Data Link Layer
- Message Protocol cEMI defined in KNX
- Embedded into an FT1.2 frame





## Application: data points

# BAOS Protocol



## BAOS Protocol: Object Server

---

- BAOS
  - **B**us **A**ccess and **O**bject **S**erver
- Abstraction
  - Separates KNX handling from client
  - Client communicates to BAOS module
  - No KNX telegrams to be handled by client
- BAOS data
  - Data points (group objects)
  - Parameters (ETS)
  - Server items (Module related)



## Advantages of the object server

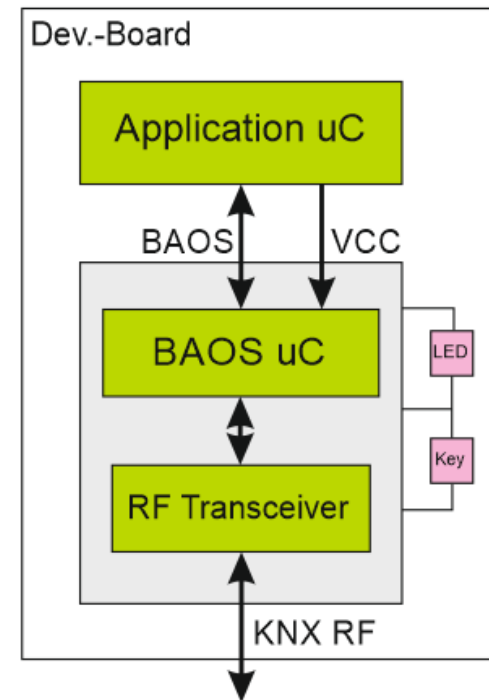
---

- No handling of KNX telegrams required
  - No knowledge of group addresses
- The object server always keeps the current values
  - Even if the application is not connected
  - No group-value-read required
  - Short latencies
- ETS download handled by BAOS
  - Decoupling from user application



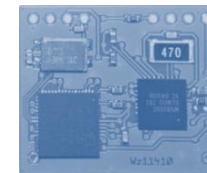
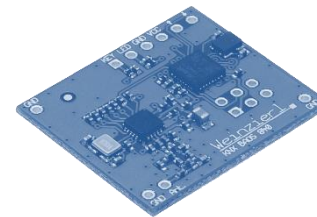
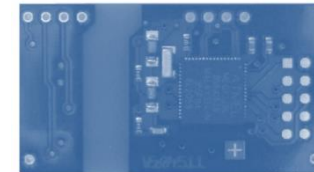
## HW structure using BAOS RF Modules

- KNX BAOS 840
  - Power supply
    - By application
  - Module
    - Microcontroller
    - KNX RF Transceiver
    - Antenna output
- Development board
  - Microcontroller for application
  - USB, virtual Comport
  - Some IOs



## KNX BAOS Firmware

- Certified KNX Stack
- BAOS Protocol
  - Bus access (telegrams)
  - Object server (data points)
- Serial protocol 'FT1.2'
  - Universal frame format
  - IEC 0870-5-1 /-2
  - Well-tried in KNX

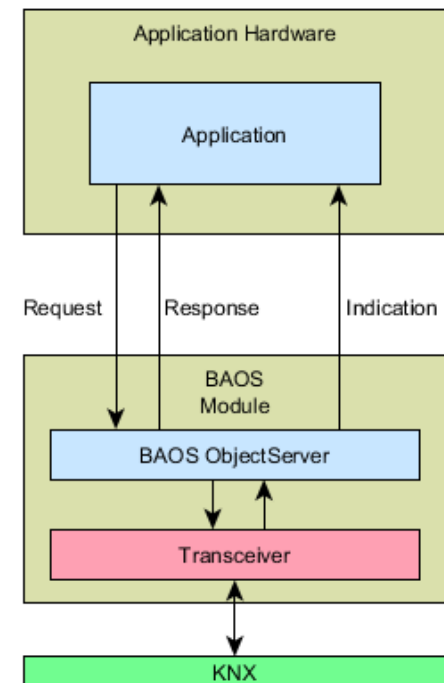
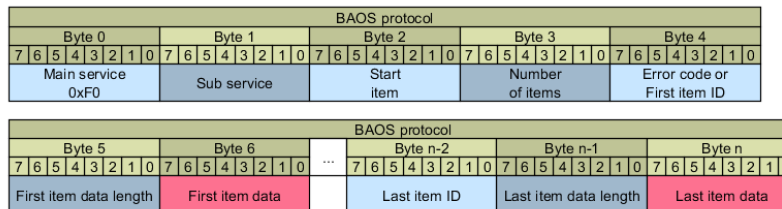


FT1.2 Header Frame				
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Start character 0x68	Length (bytes 4-n)	Length (repeated)	Start character 0x68 (repeated)	FT 1.2 control field

FT1.2 Information Frame			FT1.2 End Frame	
Byte 5	...	Byte n	Byte n+1	Byte n+2
7 6 5 4 3 2 1 0		7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Information data (0 - n bytes)			Checksum (bytes 4-n)	Stop character 0x16

## BAOS Protocol: Communication

- Request-response architecture
- Indications for object updates
- Access to Group Objects
  - `GETDATAPOINTVALUE.REQ / RES`
  - `SETDATAPOINTVALUE.REQ / RES`
  - `DATAPOINTVALUE.IND`
- Access to ETS Parameter
  - `GETPARAMETERBYTE.REQ / RES`



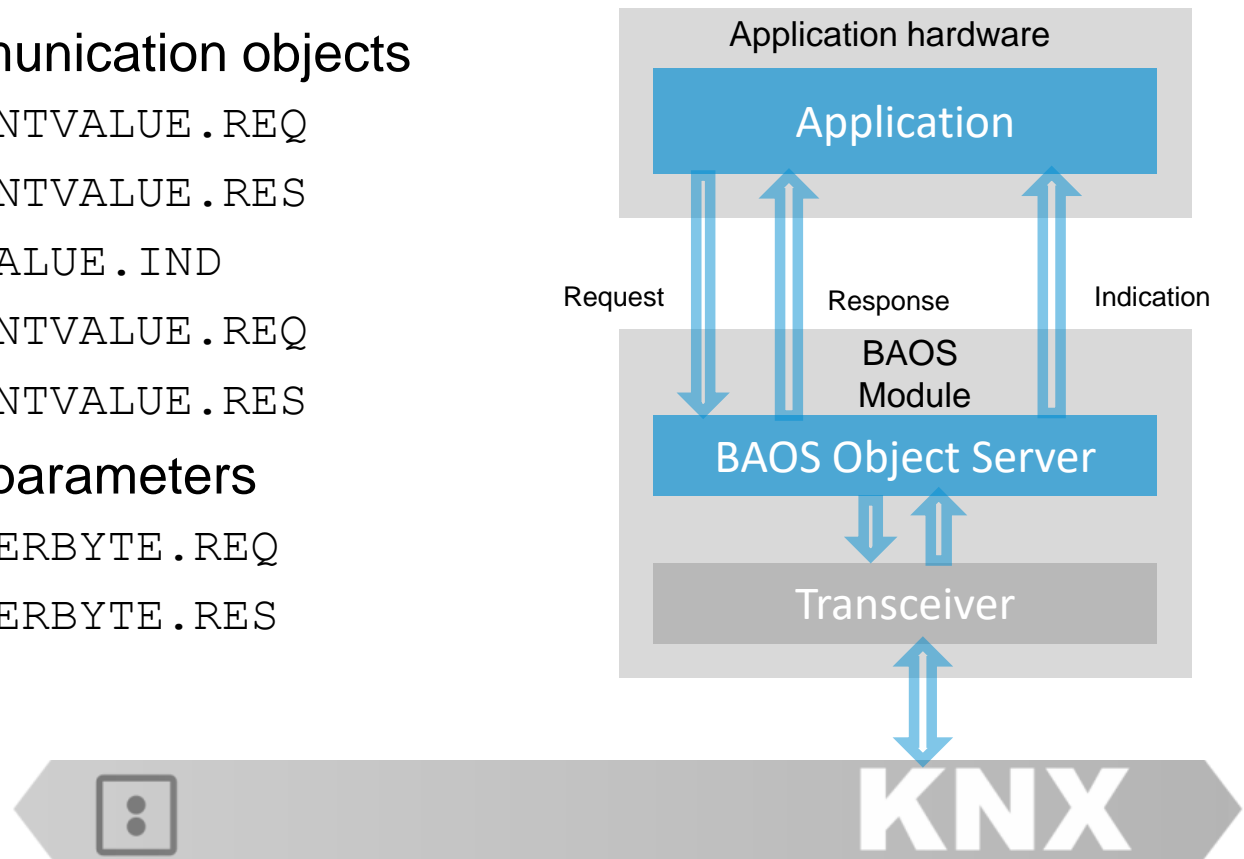
## BAOS Protocol: Communication

### ■ Access to communication objects

- GETDATAPOINTVALUE.REQ
- GETDATAPOINTVALUE.RES
- DATAPOINTVALUE.IND
- SETDATAPOINTVALUE.REQ
- SETDATAPOINTVALUE.RES

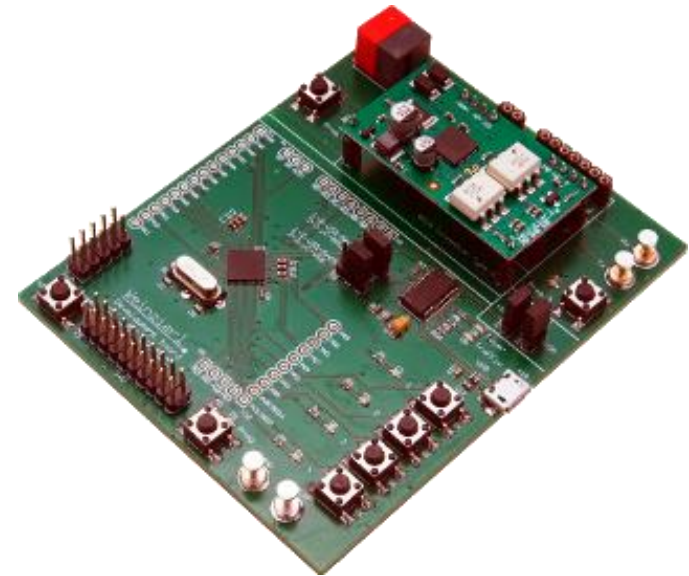
### ■ Access to ETS parameters

- GETPARAMETERBYTE.REQ
- GETPARAMETERBYTE.RES



## SW Framework & API

- Free demo code
  - Framework for KNX devices
  - Handling of BAOS protocol
  - Simple demo application
- For BAOS dev. board
- Prepared for Cortex M0x
  - Project for Atmel Studio
  - GCC compiler
- Can be ported to any  $\mu$ C
  - UART / Timer





## Framework & API

---

- Change and send a data point value:

- `uint8_t nValue = LED_OFF;`
- `KnxBaos_SendValue(1, DP_CMD_SET_SEND_VAL, 1, &nValue);`

- Get response of data point change:

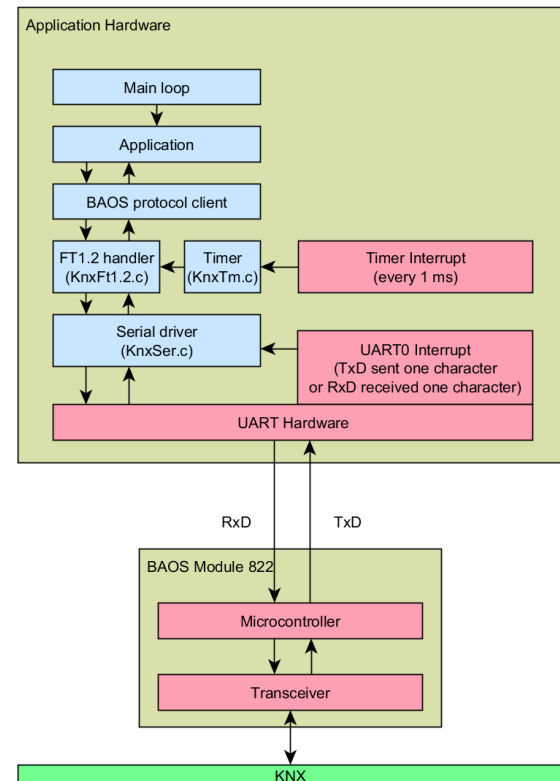
- `void App_HandleSetDatapointValueRes (`
- `uint8_t nStartDatapoint)`
- `{`
- `// Implement this`
- `}`

## Framework & API

- Get notification of data point value change:

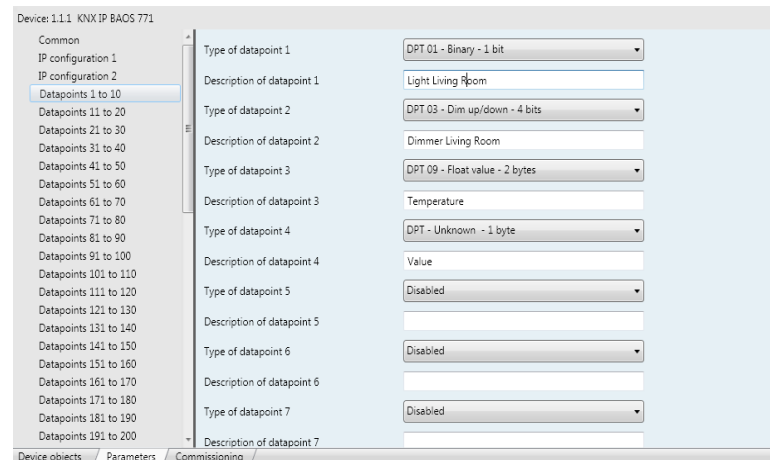
```

■ void App_HandleDatapointValueInd(
■     uint8_t nDpId,
■     uint8_t nDpState,
■     uint8_t nDpLength,
■     uint8_t* pData)
■ {
■     // Implement this
■ }
    
```



## ETS database entry

- Generic ETS database entry for a quick start
- DCA for import/export



Device: 1.1.1 KNX IP BAOS 771

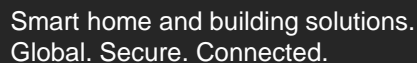
Group	Type of datapoint	Description of datapoint
Datapoints 1 to 10	DPT 01 - Binary - 1 bit	Light Living Room
Datapoints 11 to 20	DPT 03 - Dim up/down - 4 bits	Dimmer Living Room
Datapoints 21 to 30	DPT 09 - Float value - 2 bytes	Temperature
Datapoints 31 to 40	DPT - Unknown - 1 byte	Value
Datapoints 41 to 50	Disabled	
Datapoints 51 to 60	Disabled	
Datapoints 61 to 70	Disabled	
Datapoints 71 to 80	Disabled	
Datapoints 81 to 90	Disabled	
Datapoints 91 to 100	Disabled	
Datapoints 101 to 110	Disabled	
Datapoints 111 to 120	Disabled	
Datapoints 121 to 130	Disabled	
Datapoints 131 to 140	Disabled	
Datapoints 141 to 150	Disabled	
Datapoints 151 to 160	Disabled	
Datapoints 161 to 170	Disabled	
Datapoints 171 to 180	Disabled	
Datapoints 181 to 190	Disabled	
Datapoints 191 to 200	Disabled	

Device objects / Parameters / Commissioning

- Individual ETS database entries
  - By the manufacturer using KNX MT
  - As service from Weinzierl Engineering



- # net'n node



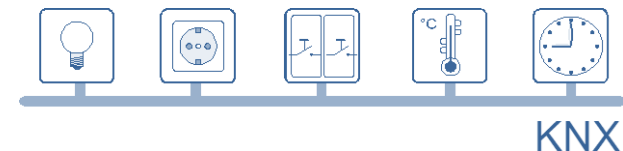
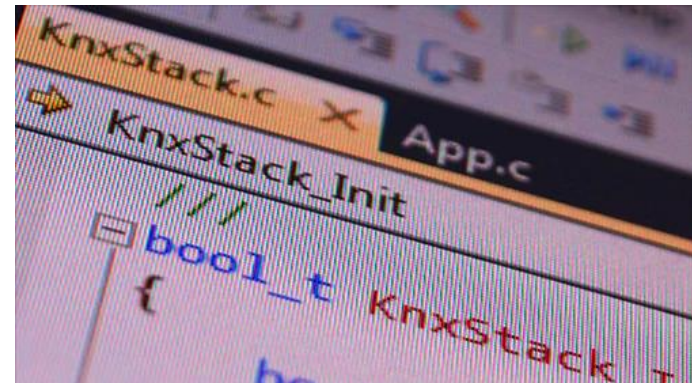
## KNX BAOS solutions for RF

- KNX BAOS Module 840 RF
  - Host interface with serial FT1.2
  - To extend embedded devices with KNX RF
- KNX USB Stick 340 RF
  - USB Stick for ETS
  - Host interface with USB HID
  - Support of KNX BAOS Protocol
  - As external KNX RF stack for devices with OS like Linux



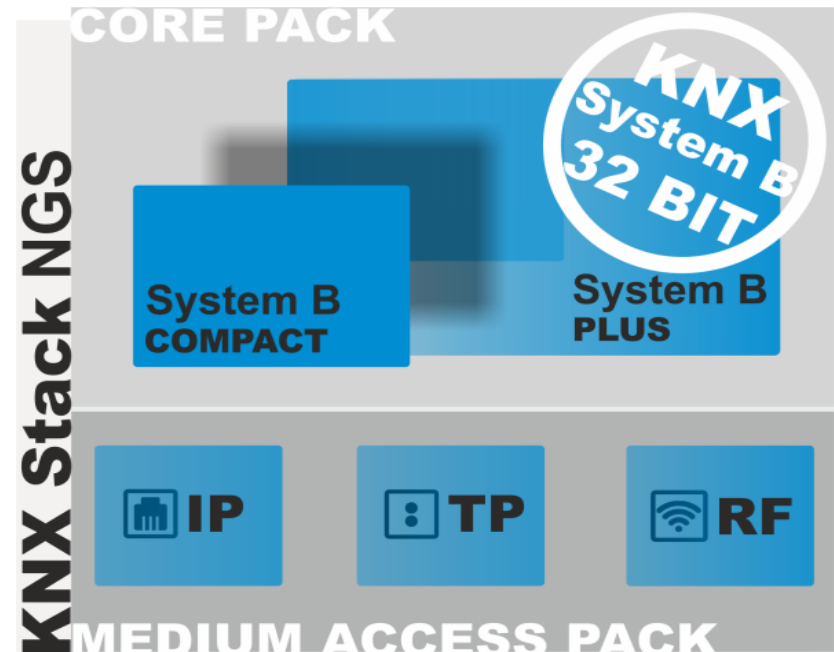
## KNX Stack NGS: Application Area

- For KNX end devices
- When using a stack?
  - High volumes
  - Many device types
- Benefits
  - High flexibility
  - Low cost production
- Take into account
  - High investment, also in the team
  - For skilled developers



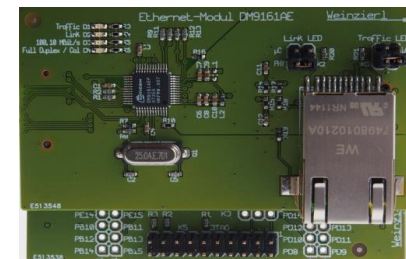
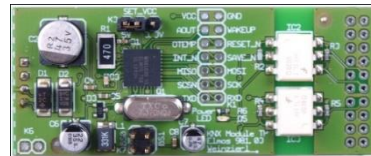
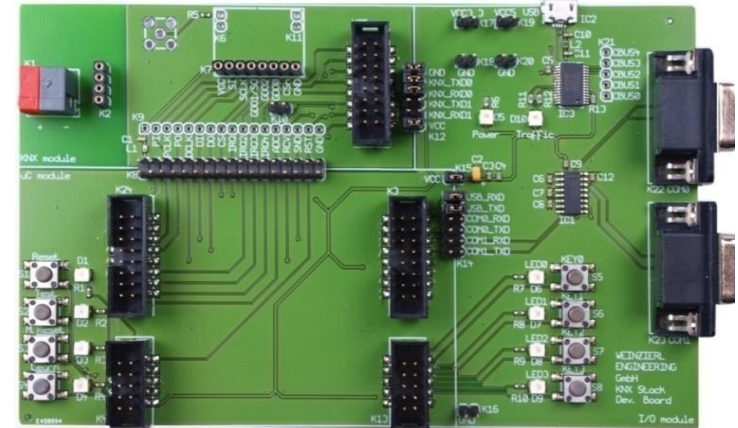
## KNX Stack Implementation NGS

- Professional solution for high volume products
- **KNX Security integrated**
- Modular
  - Twisted Pair TP
  - **Radio Frequency RF**
  - Ethernet / IP
- Scalable
  - Compact
  - Plus
- Development HW
- Tools



## Modular Development Boards

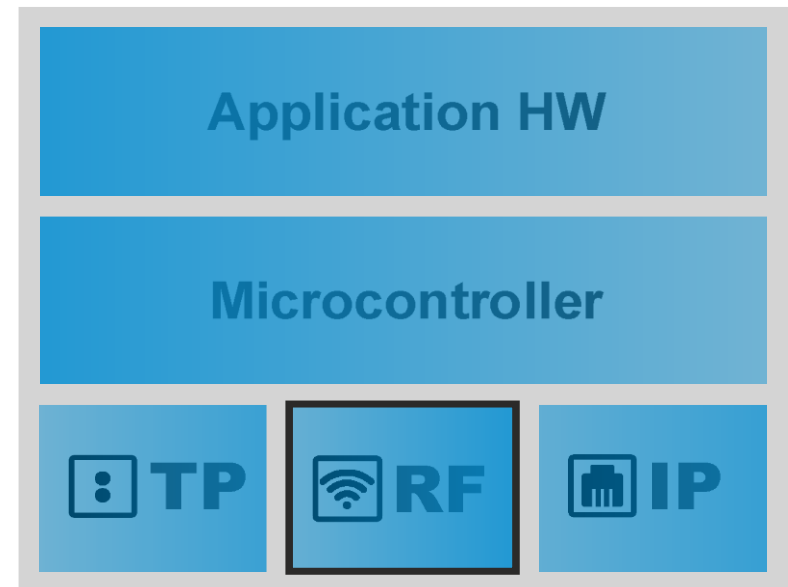
- Base Board
- Micros
  - Broad range 8..32 Bit
- Bus Access
  - KNX TP
  - KNX RF
  - KNX IP





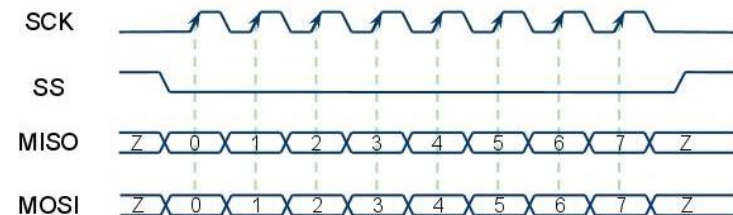
## Medium Access Pack RF

- Bus-Access
  - KNX RF
  - Data link layer for ISM Transceiver
  - SPI Driver for Data & Configuration



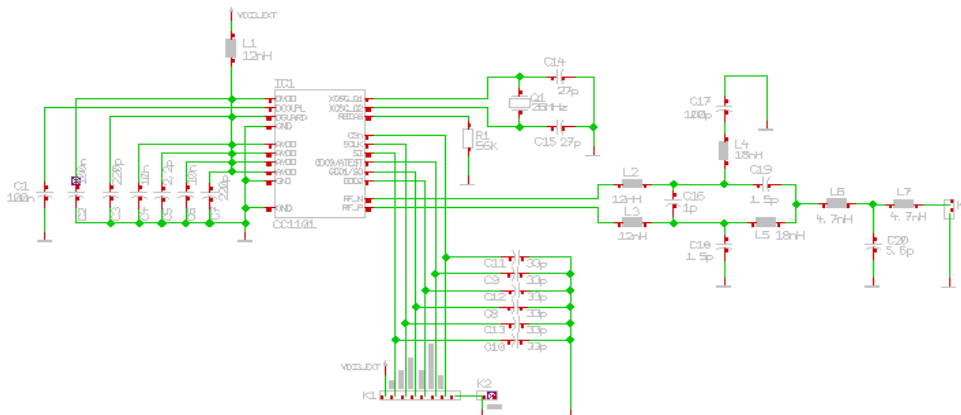
## Hardware Connection

- Use schematic of RF transceiver
- Add microcontroller with clock, etc.
- Connect RF configuration interface
  - 4-wire interface
  - uC is master
- Connect RF data interface
  - SPI interface
  - uC is slave
  - Bit clock generated by RF chip
- Add application specific parts



## ISM Transceivers

- Microchip MRF89XA
- Semtech SX1211
- Silicon Labs Si4431 / Si4432
- TI / Chipcon CC1101
- ST S2LP



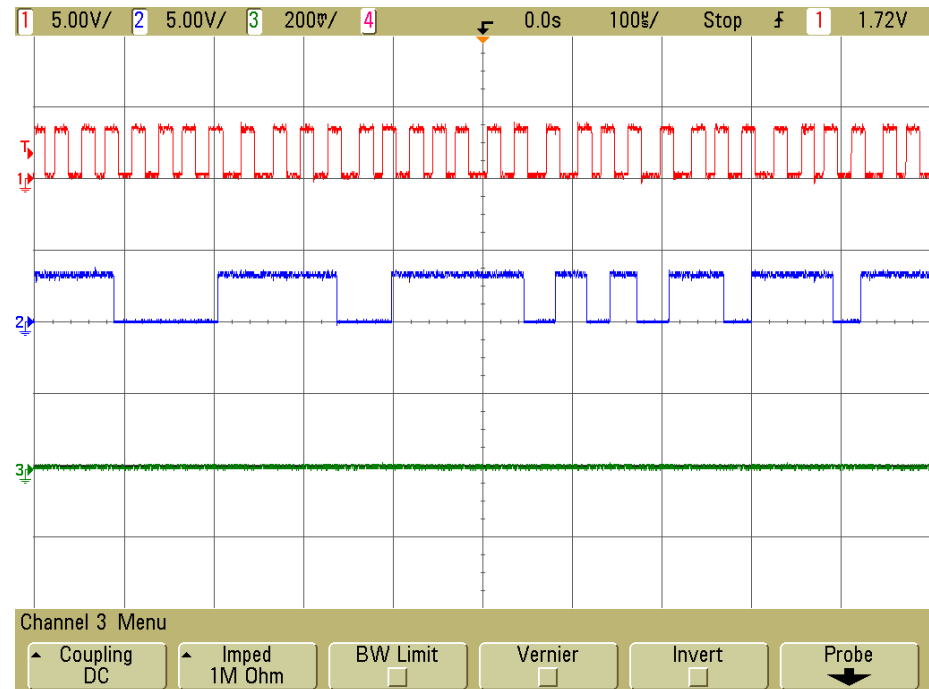
## Modules for KNX RF

- Standard ISM Solutions
  - Microchip MRF89XA / Semtech SX1211
  - Silicon Labs Si4431 / Si4432
  - TI / Chipcon CC1101
  - ST Spirit 1 / S2LP
- Transceiver Modules
  - For Weinzierl Development Boards
  - For Products
- Antenna
  - Wire, PCB, Chip
  - Matching required



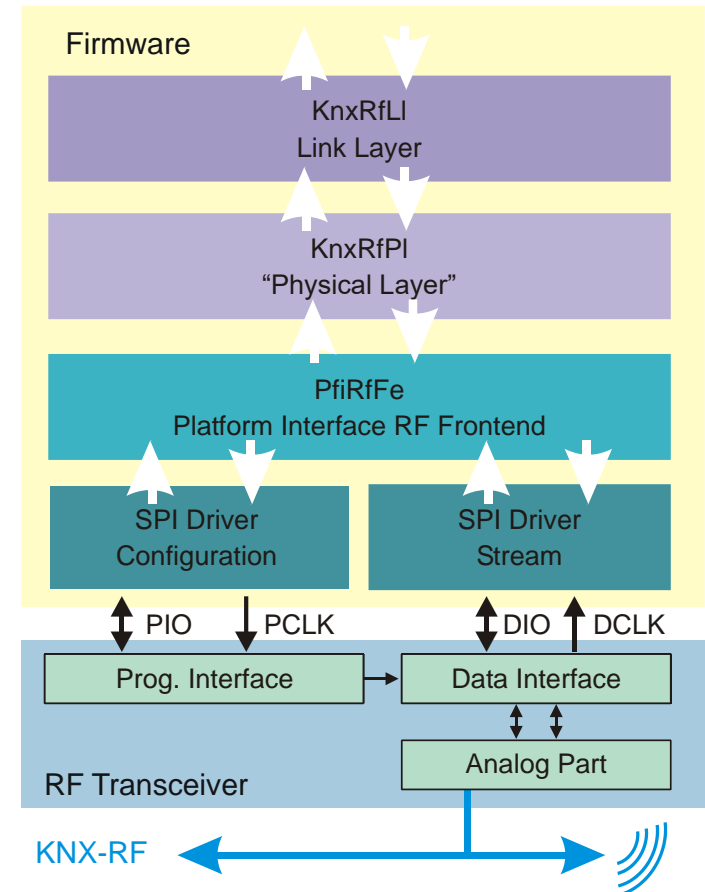
## Sending & Receiving

- Sending
  - Pre-header
  - Manchester Coding
- Receiving
  - Transceiver in NRZ mode
  - Not synchronized
  - Permanent sampling
  - Chip rate 32 768 cps
  - 4 096 patterns per second
  - Manchester decoding

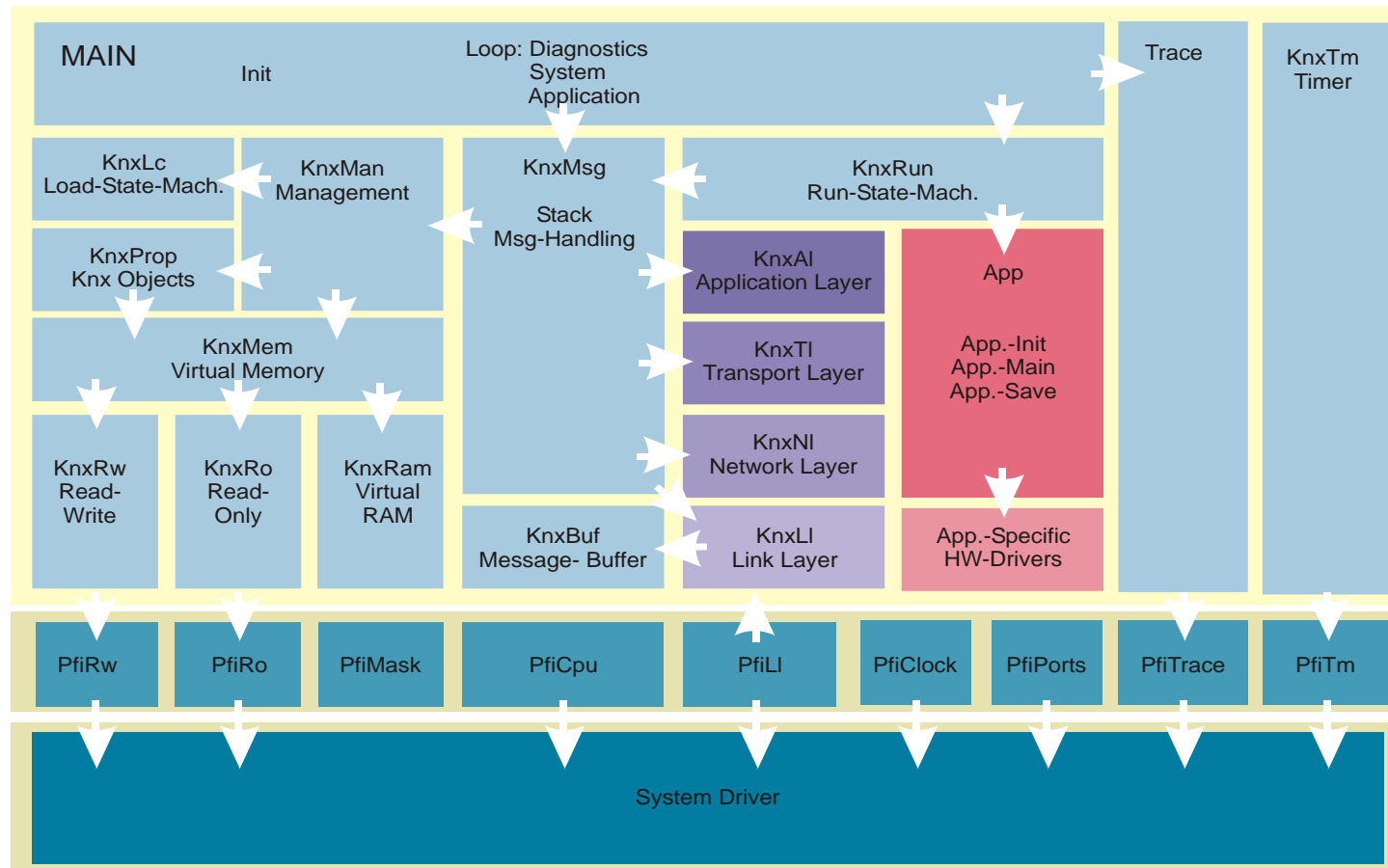


## Physical & Link Layer

- Link Layer
  - Block building
  - Checksum
- Bit handling
  - Pre-header sending/detection
  - Manchester coding/decoding
- HW driver
  - Interface to transceiver
  - Interrupts



## Firmware Overview



## Additional Services for RF

---

- Services for open media
  - APCI\_DOMAINADDRESS\_WRITE
  - APCI\_DOMAINADDRESS\_READ
  - APCI\_DOMAINADDRESS\_SER\_NUM\_WRITE
  - APCI\_DOMAINADDRESS\_SER\_NUM\_READ
  - APCI\_DOMAINADDRESS\_SER\_NUM\_RESP



## KNX Development Tools

---

- Net'n Node
  - Bus Monitor and Analyzer
- TraceMon
  - Optimized debug support
- kScript
  - Model driven design
  - Script based system
  - Automated generation of ETS product entries
- kDrive SDK
  - For tool development
  - Bus access and services
  - Free and commercial versions

**net'n'node**  
**kScript**  
**kDrive**

## Conclusion

---

- KNX RF
  - Fully integrated in the KNX System
  - One tool fits all: ETS6
  - Potential for complex topologies
  - Security enables new applications
  
- Technology available
  - Specification, ETS, EITT
  - KNX Stack and Modules
  - Development tools



## Thanks.

### KNX RF development with Weinzierl solutions

For general questions:  
[info@knx.org](mailto:info@knx.org) – [www.knx.org](http://www.knx.org)

Weinzierl Engineering GmbH  
DE-84508 Burgkirchen / Alz  
+49 (0)8677 916 36 – 0  
[info@weinzierl.de](mailto:info@weinzierl.de)  
[www.weinzierl.de](http://www.weinzierl.de)



Smart home and building solutions.  
Global. Secure. Connected.

