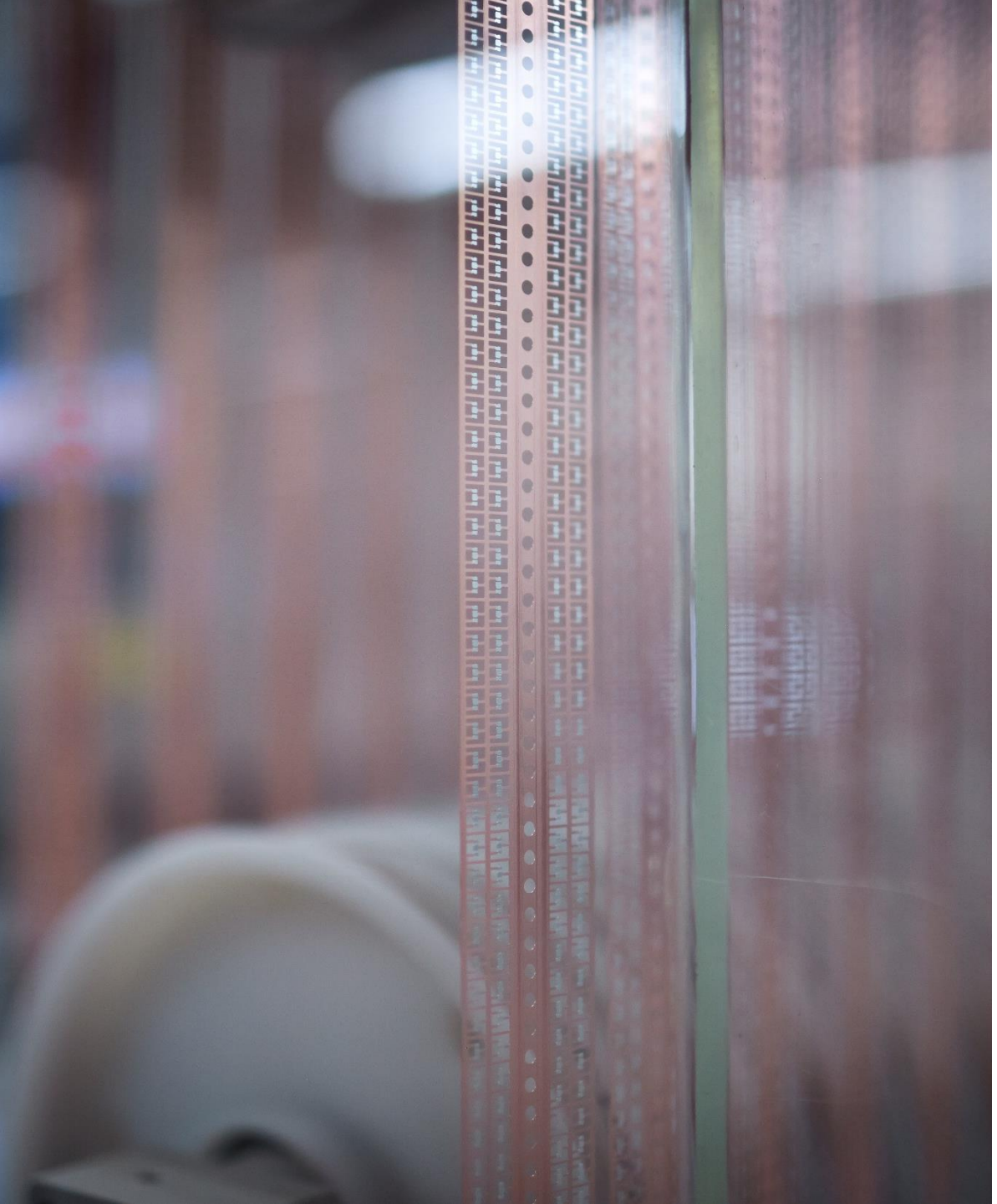
PARAMETRIZATION VIA NFC

WEBINAR SERIES:
HOW TO BUILD NFC APPLICATIONS

JORDI JOFRE
NFC READERS
NFC EVERYWHERE
21/02/2017

SECURE CONNECTIONS
FOR A SMARTER WORLD



Parametrization via NFC

Agenda

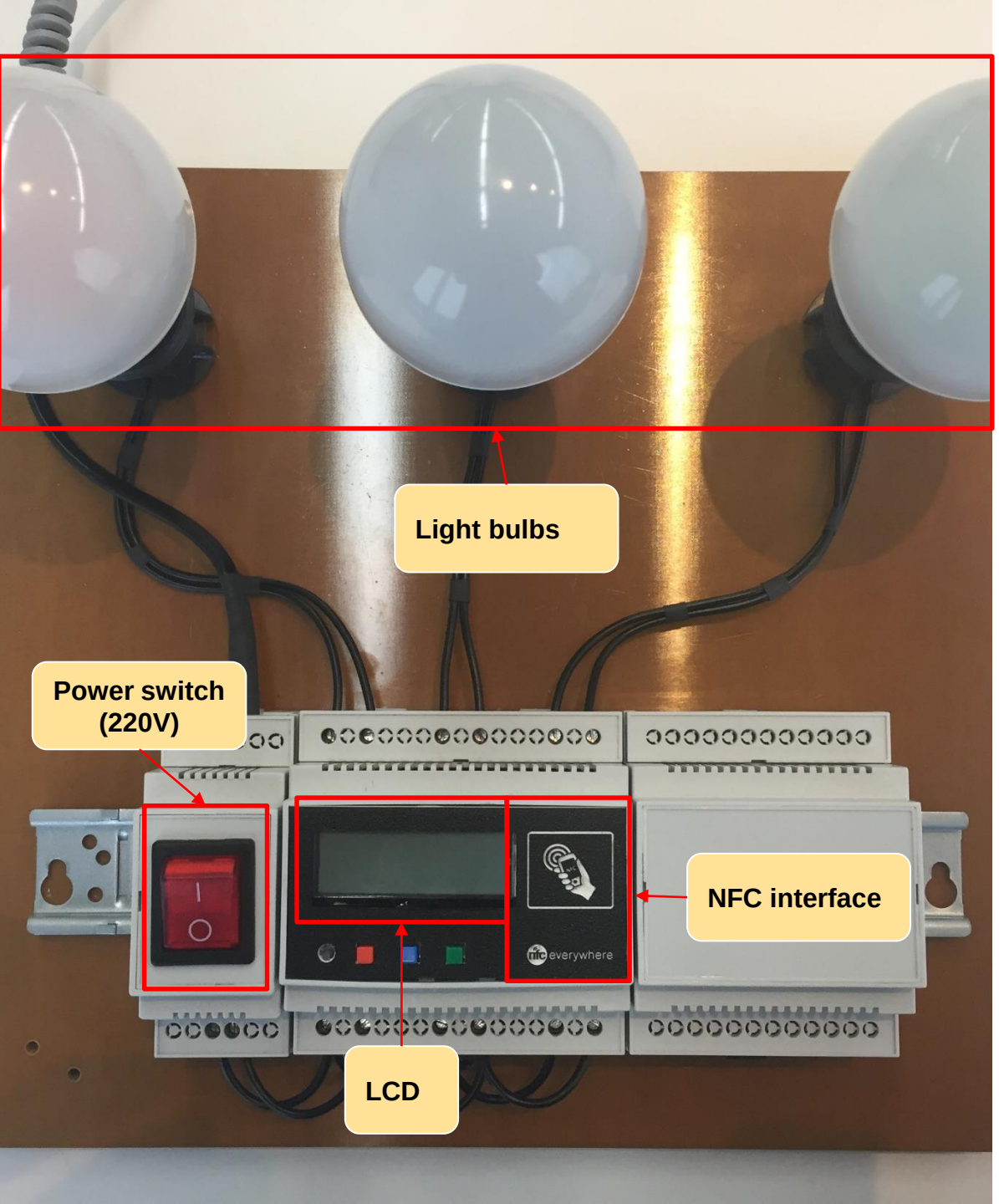
- NFC DIN rail module demo
  - Demo functionality
  - Hardware details
  - How the NTAG I²C *plus* is used
  - MCU / embedded software integration
  - Phone / NFC device software integration
  - Available resources
- Wrap up and Q&A

NFC DIN RAIL MODULE DEMO

DEMO FUNCTIONALITY

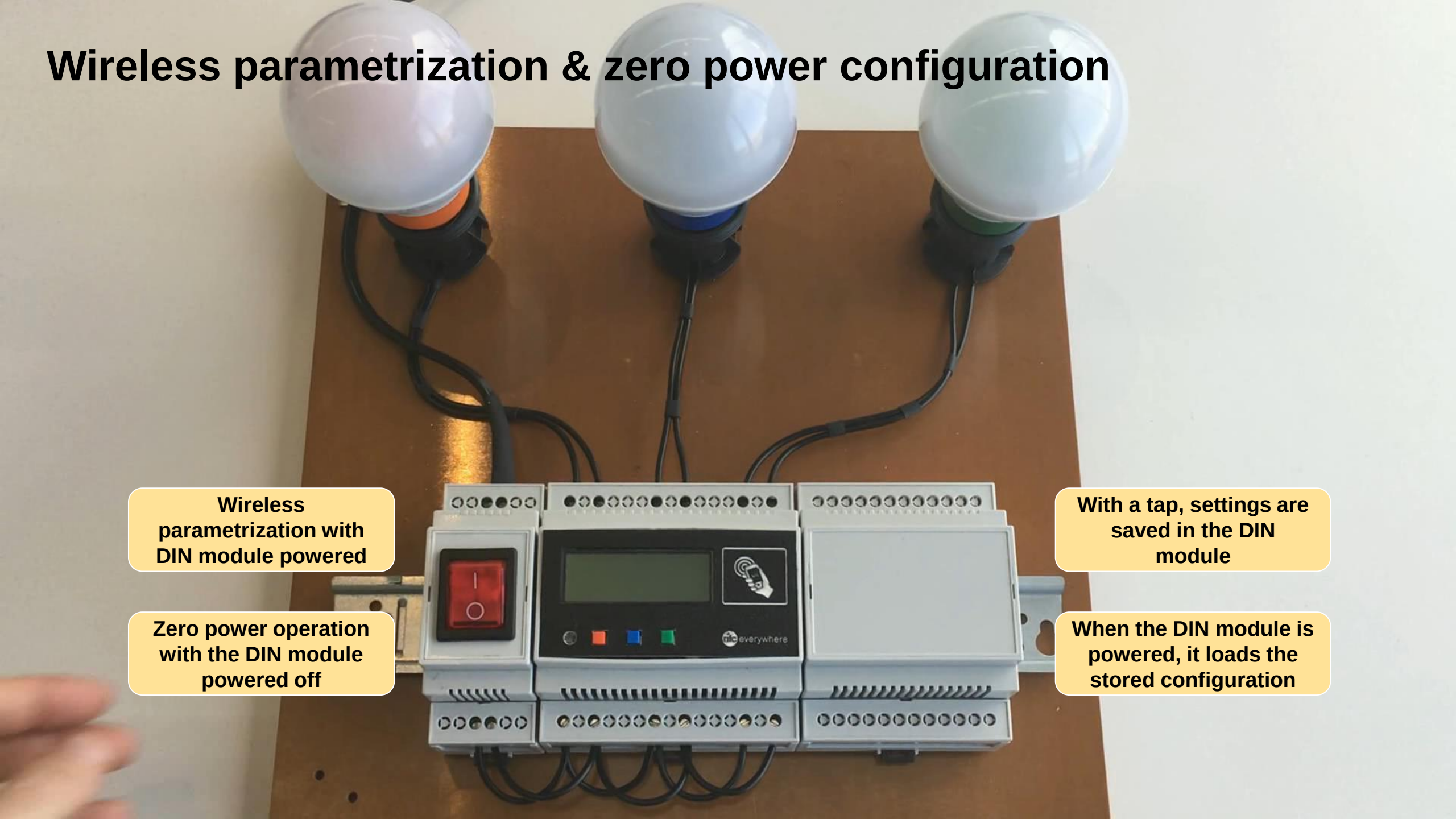
NFC DIN rail module demo

Elements

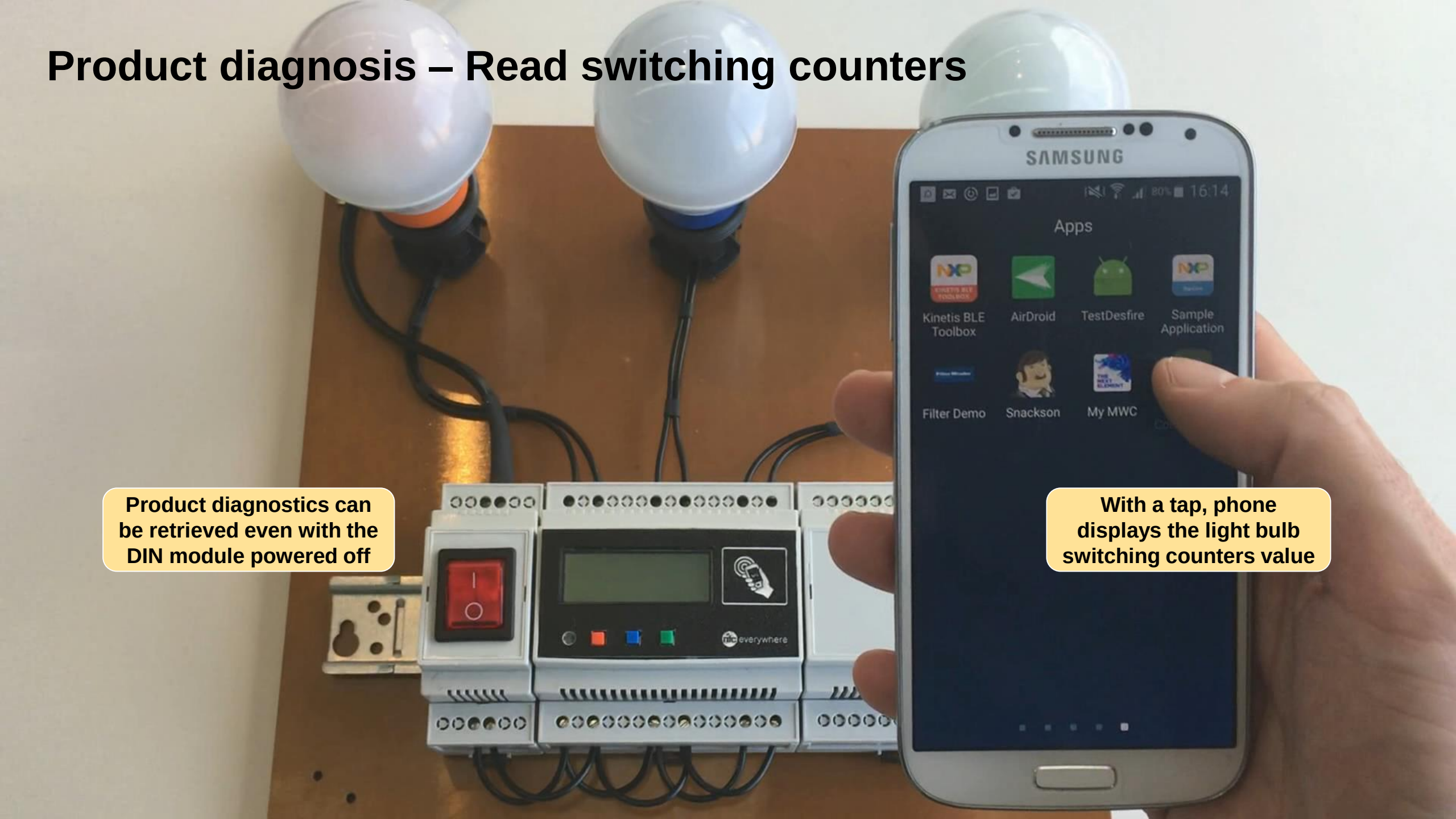


- It demonstrates the use of NFC for:
  1. Wireless parametrization & Zero power configuration
  2. Product diagnosis
  3. Wireless firmware update

Wireless parametrization & zero power configuration



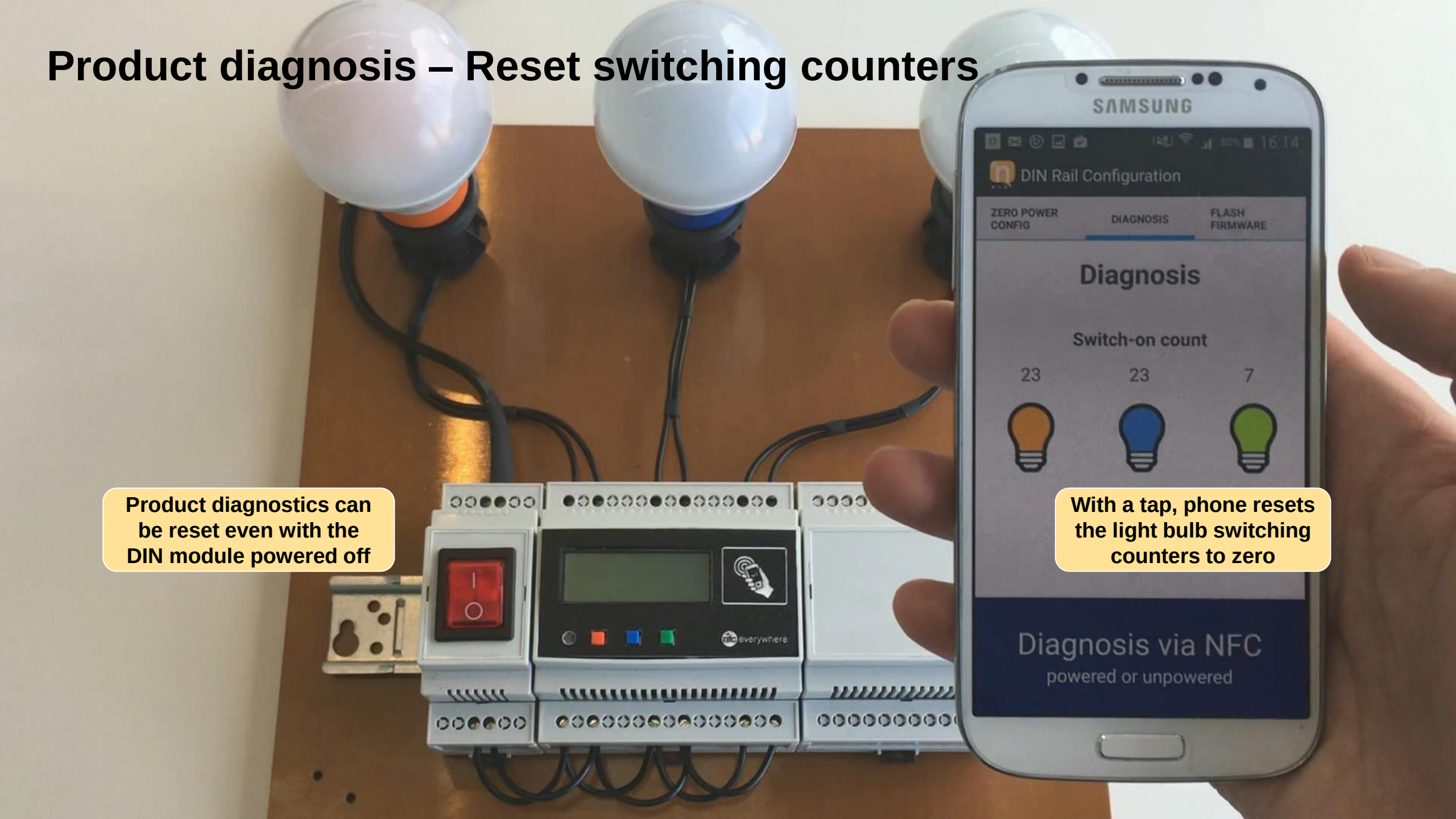
Product diagnosis – Read switching counters



Product diagnostics can be retrieved even with the DIN module powered off

With a tap, phone displays the light bulb switching counters value

Product diagnosis – Reset switching counters



Product diagnostics can be reset even with the DIN module powered off

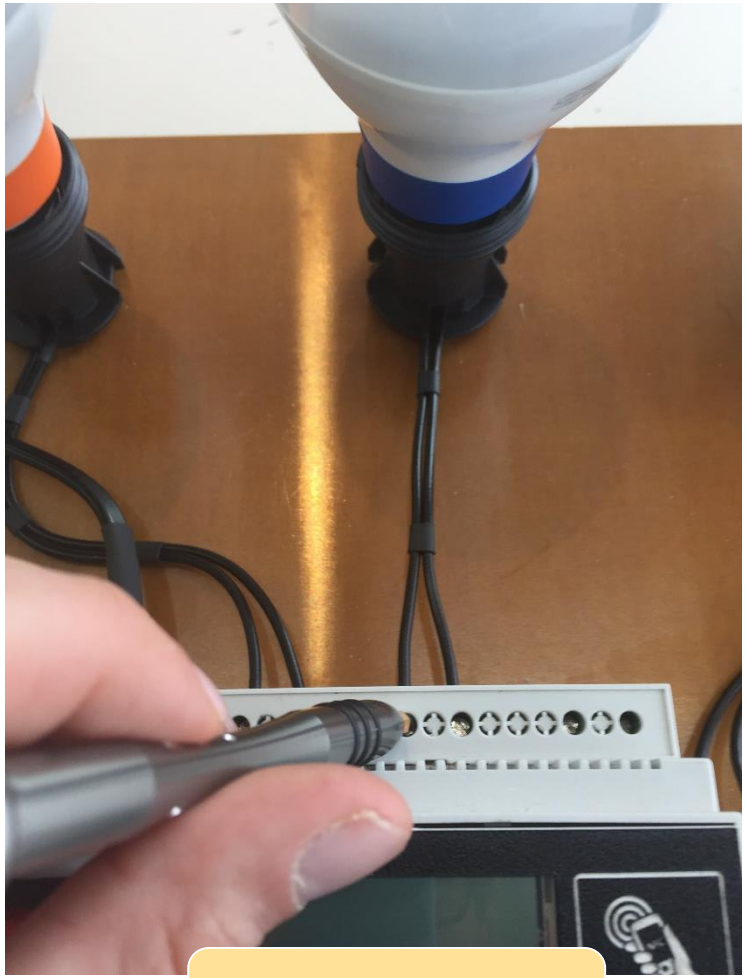
With a tap, phone resets the light bulb switching counters to zero

Wireless firmware upgrade

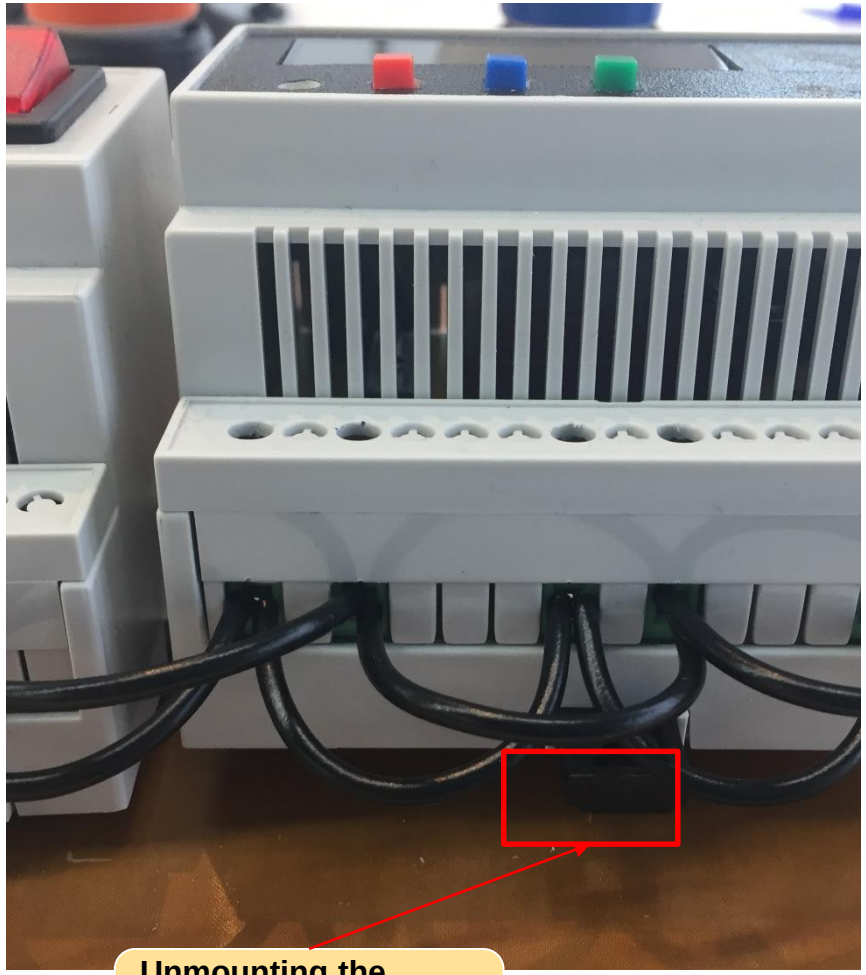
HARDWARE DETAILS

Disassembling NFC DIN rail module demo (I)

Unscrew and release the power wires



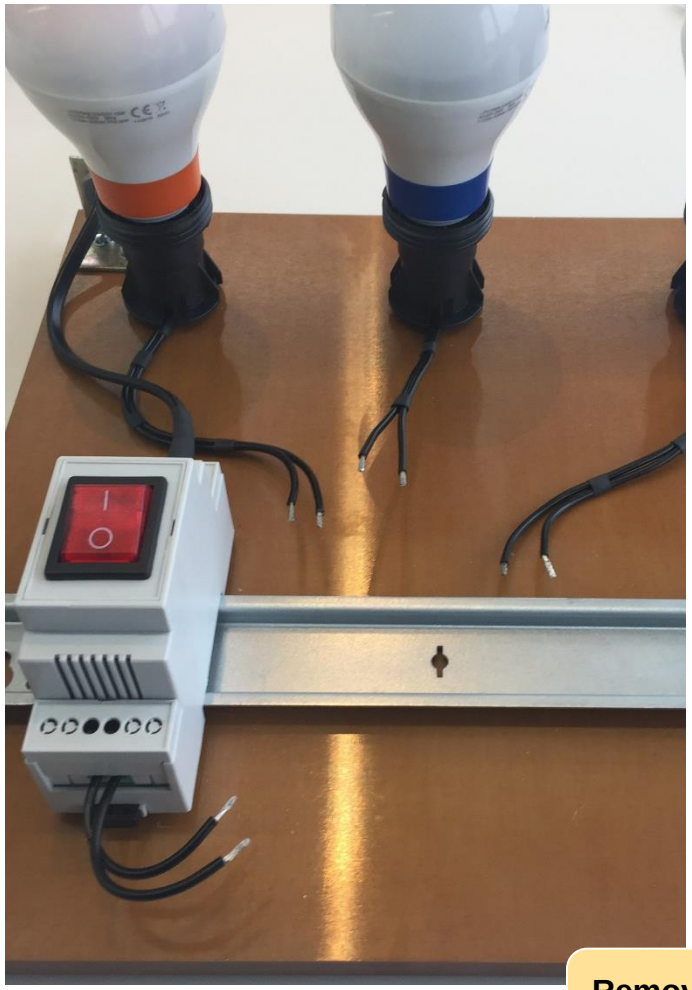
Unscrew and release light bulb power wires



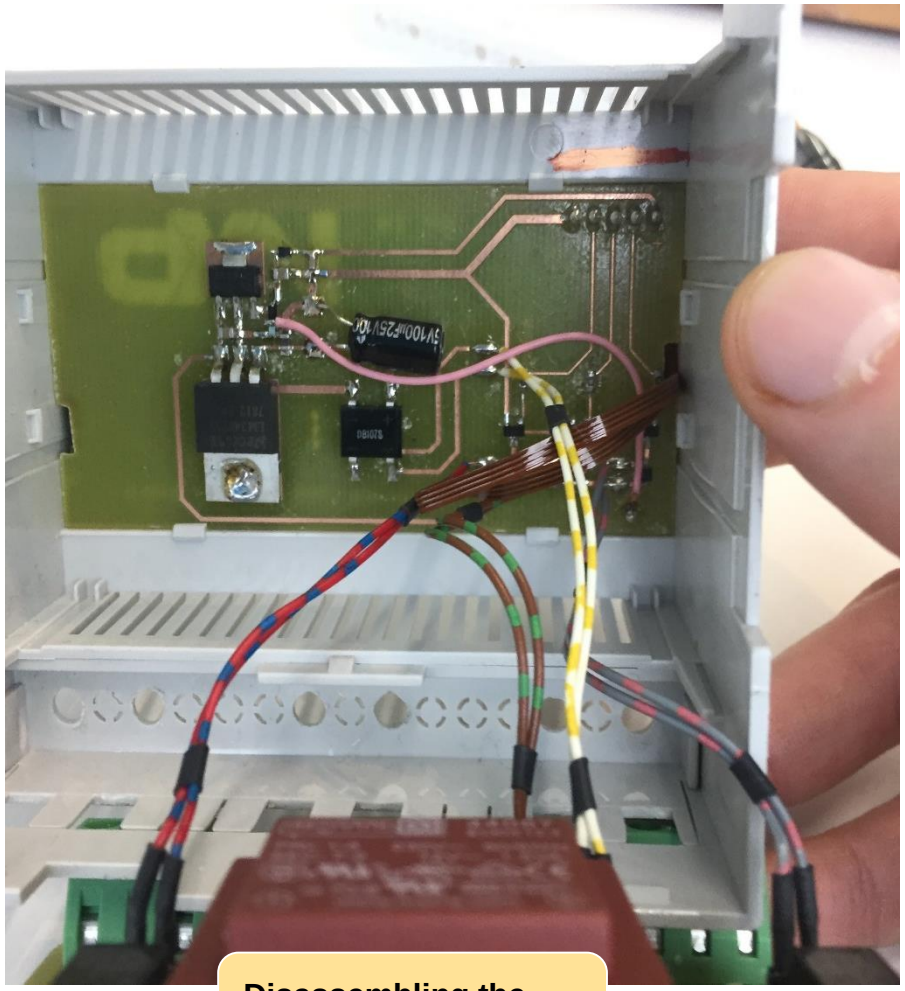
Unmounting the module from the DIN rail

Training Mobile Knowledge

Disassembling NFC DIN rail module demo (II)



Remove the module from the DIN rail



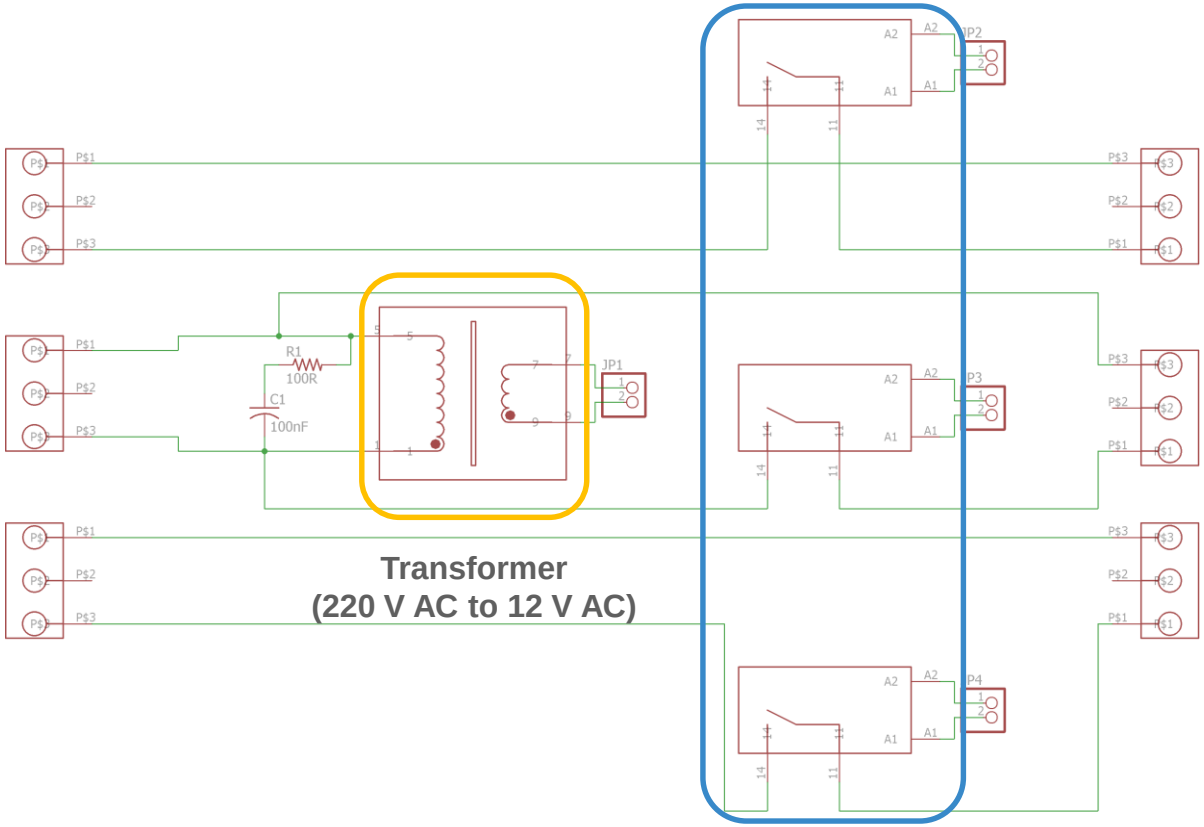
Disassembling the module

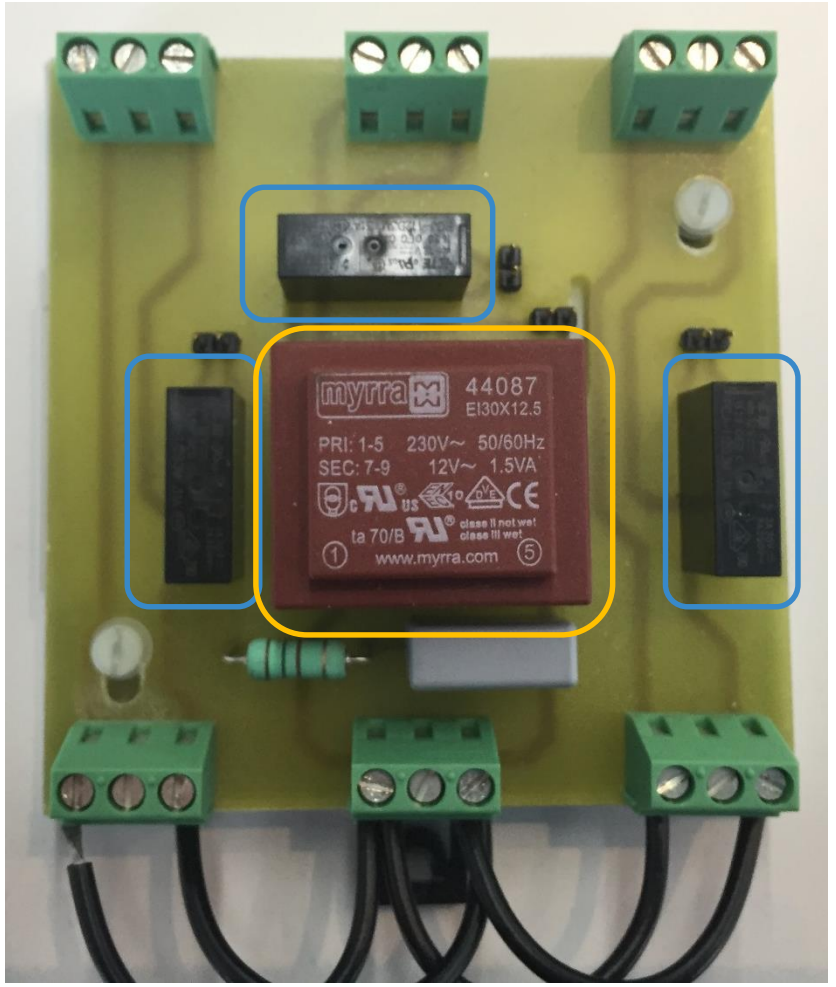
Training Mobile Knowledge

Hardware details - Transformer PCB

The transformer PCB includes:

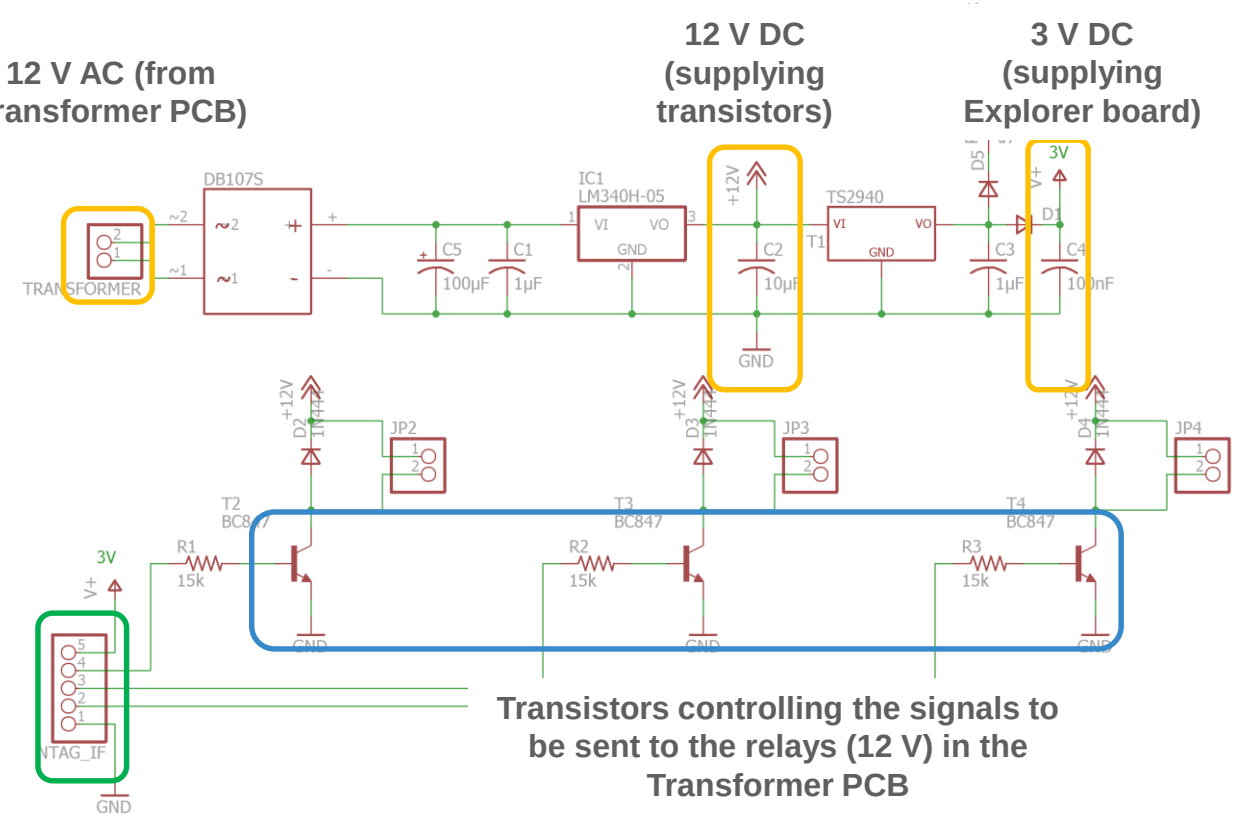
- Three electromechanical relays controlling the light bulbs.
- Transformer converting 220 V to 12 V to supply the switching PCB.





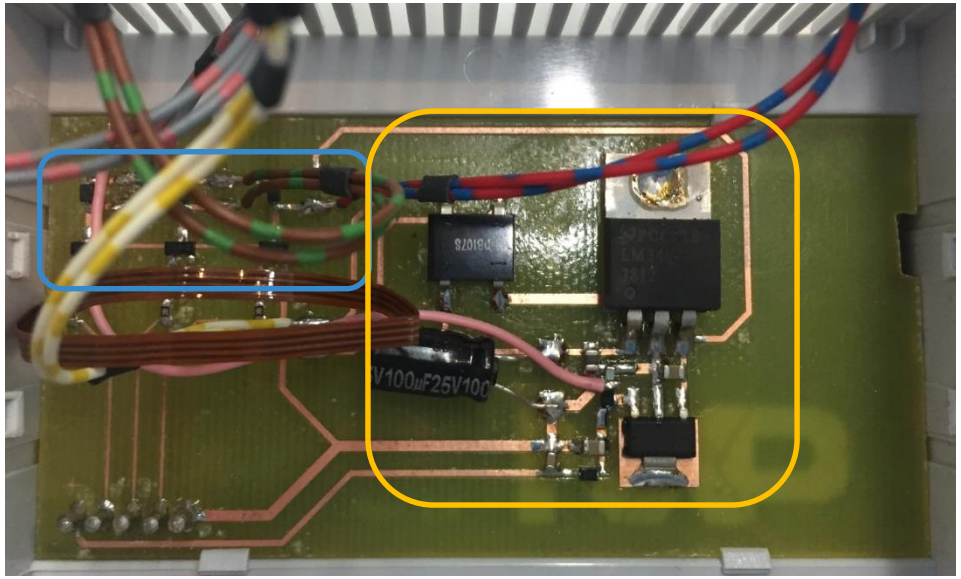
Training Mobile Knowledge

Hardware details - Switching PCB



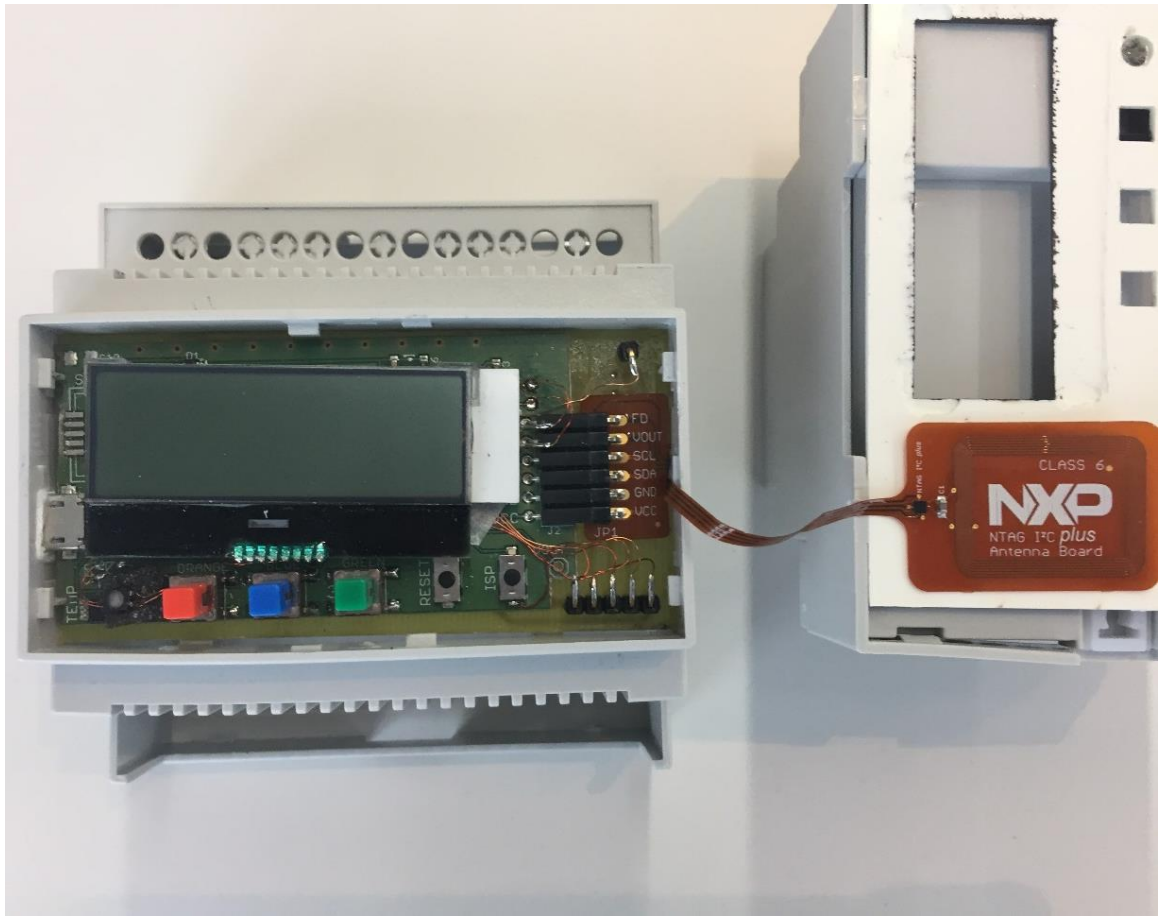
The switching PCB contains:

- Power supply unit (AC/DC converter from 12V AC to 12V DC and 3V DC)
- Transistors controlling the input signal to be sent to the relays in the Transfomer PCB



Training Mobile Knowledge

Hardware details - Explorer board + NTAG I²C *plus* flex antenna



- About the Explorer board HW:
  - 5 buttons (3 Color buttons, RESET and ISP)
  - NXP LM75B digital temperature sensor
  - NXP **LPC11U24** 32-bit ARM Cortex-M0 microcontroller
  - thermal watchdog
  - RGB light
  - LCD display
  - microUSB for PC connection
  - JTAG for FW upload
  - I²C Bus Connector
- Class 6 flex antenna embedding NTAG I²C *plus*

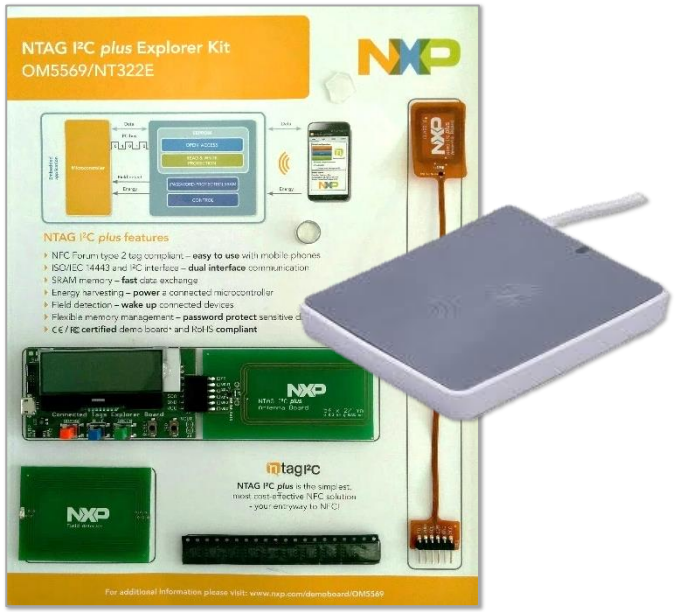
* Design files for the **Explorer board**, **Class 4 antenna** and **Class 6 flex antenna** can be found in:
http://www.nxp.com/products/wireless-connectivity/nfc-and-reader-ics/connected-tag-solutions/ntag-ic-plus-explorer-kit-development-kit:OM5569-NT322E?tab=Design_Tools_Tab

NTAG I²C *plus* Explorer demokit and variants

Demokits for **NTAG I²C *plus***

Ordering details

- Orderable part number: **OM5569-NT322E**
- 12NC: 935307849699
- URL: http://www.nxp.com/demoboard/OM5569-NT322E.html



Ordering details

- Orderable part number: **OM5569-NT322ER**
- 12NC: 935307848699
- URL: http://www.nxp.com/demoboard/OM5569-NT322ER.html

Ordering details

- Orderable part number: OM5569-NT322F
- 12NC: 935307851699
- URL: http://www.nxp.com/demoboard/OM5569-NT322F.html

* NTAG I²C *plus* Explorer demokit info and ordering details:
http://www.nxp.com/products/identification-and-security/nfc-and-reader-ics/connected-tag-solutions/ntag-ic-iplus-i-explorer-kit:OM5569-NT322E

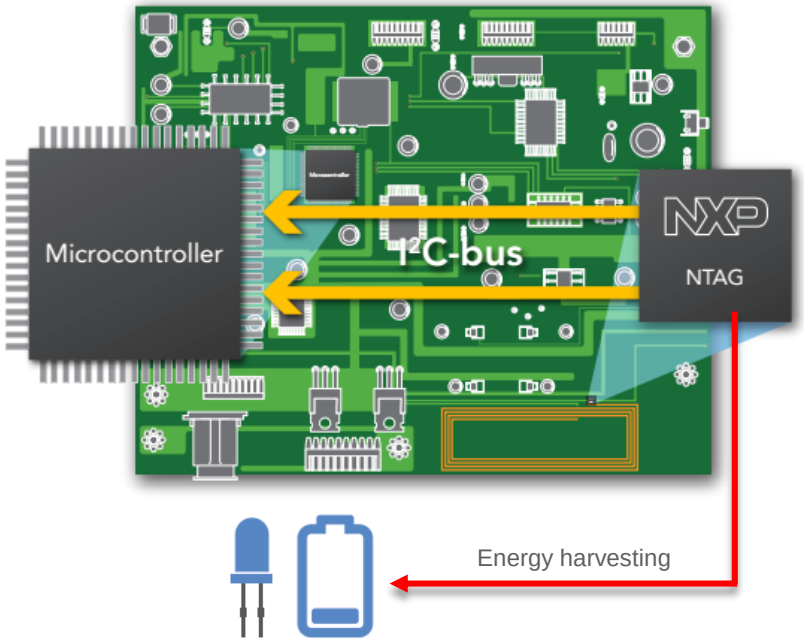
Training Mobile Knowledge

HOW THE NTAG I²C *PLUS* IS USED

PRODUCT FEATURES

How NTAG I²C *plus* works

The NTAG I²C *plus* connects to the microcontroller via the I²C serial bus interface



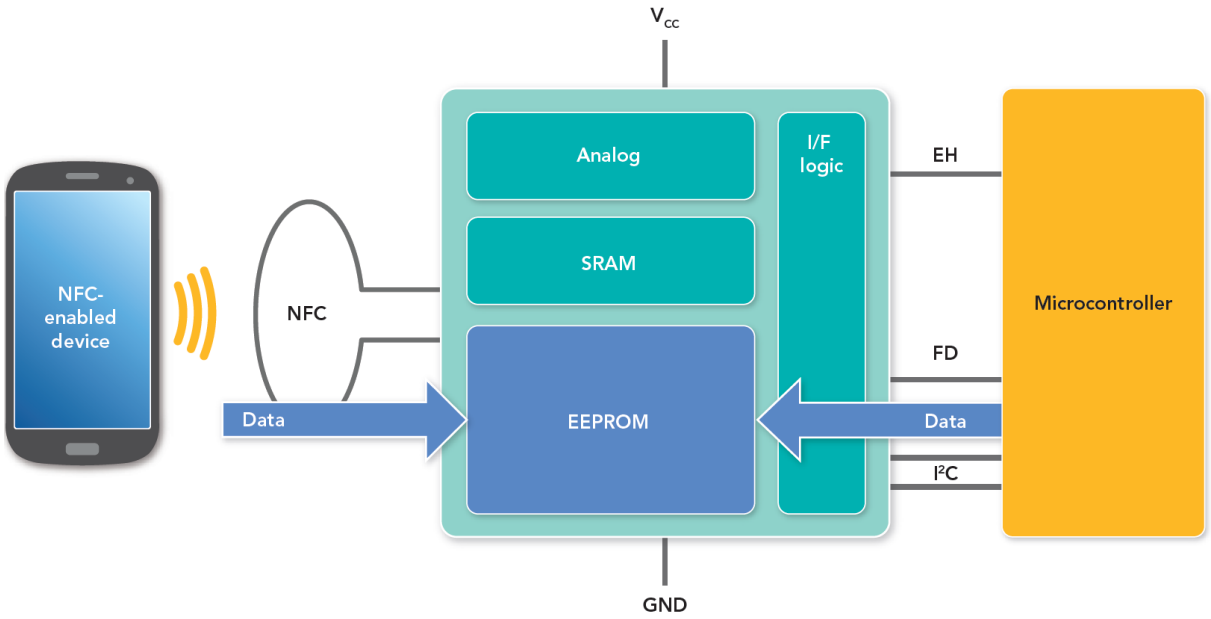
NFC-enabled mobile device connects to the MCU via the NFC interface, using the I²C as the communication conduit

- Field Detection
- Energy harvesting
- SRAM memory
- SRAM mirroring
- Pass-through mode
- Memory access management
- Originality signature

Training Mobile Knowledge

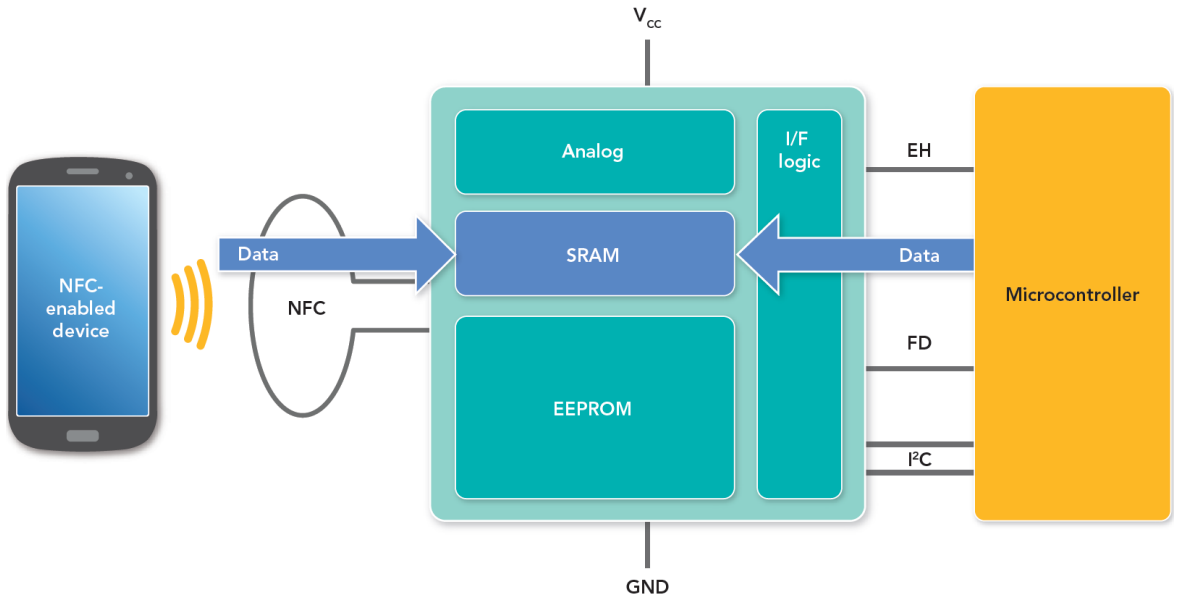
Data exchange between I²C and NFC interfaces

Using the EEPROM in Passive or Static mode



- With the power to the device off, the NFC interface can still operate and write into the EEPROM.
- Later, when the device has power, the microprocessor can access the previously written data via the I²C interface.
- Similarly, the microprocessor can write data to the EEPROM while powered for later access via the NFC interface whether or not the device has power.

Using the SRAM in pass-through mode



- The NTAG I²C *plus* tag chip operates like a modem when in this mode.
- Data flows from the NFC interface through an SRAM buffer to the I²C serial bus interface or vice versa
- The on-chip, 64-byte SRAM preserves EEPROM access limits

Training Mobile Knowledge

HOW THE NTAG I²C *PLUS* IS USED

ZERO POWER CONFIGURATION DEMO USE CASE

Application logic for zero power configuration use case (I)

1. Phone writes config in EEPROM

Page address (HEX)	Byte number within a page			
	0	1	2	3
0x00				
…	…… ……			
0x34				B1
0x35	B2	B3	L	
0x36				
0x37				
…	… … … …			
0xF0				
…	….			
0xFB				
0xFC				
0xFD				
0xFE				
0xFF				

Fig. Simplified NTAG I²C *plus* memory map (NFC interface perspective)

EEPROM memory map (NFC perspective)	
Page address 0x34	Stores the Light bulb 1 setting (**B1**) ‘0’ – light bulb ON ‘1’ – light bulb OFF ‘2’ – light blink
Page address 0x35	Stores the Light bulb 2 setting (**B2**) ‘0’ – light bulb ON ‘1’ – light bulb OFF ‘2’ – light blink
Page address 0x35	Stores the Light bulb 3 setting (**B3**) ‘0’ – light bulb ON ‘1’ – light bulb OFF ‘2’ – light blink
Page address 0x35	Stores the language setting (**L**) ‘0’ – Deutsch ‘1’ – Bavarian ‘2’ – Swiss ‘3’ – English ‘4’ – French

SRAM mirror memory area (volatile memory)

EEPROM memory area (non-volatile memory)

Training Mobile Knowledge

Application logic for zero power configuration use case (II)

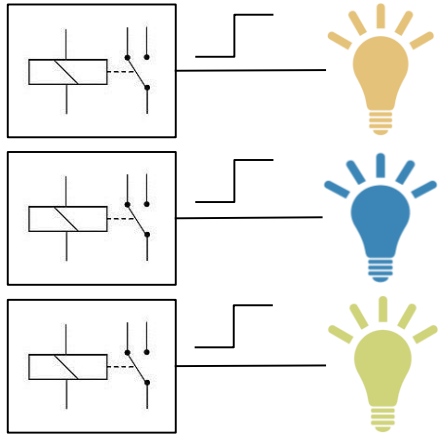
Page address (HEX)	Byte number within a page			
	0	1	2	3
0x00				
…	…… ……			
0x34				B1
0x35	B2	B3	L	
0x36				
0x37				
…	… … … …			
0xF0				
…	….			
0xFB				
0xFC				
0xFD				
0xFE				
0xFF				

Fig. Simplified NTAG I²C *plus* memory map (NFC interface perspective)

I²C

2. MCU reads config from EEPROM

3. MCU changes GPIO **17**, **18** and **19** output config according to **B1**, **B2** and **B3** settings



4. Relays turn ON / OFF light bulbs

4. The relay status changes accordingly to GPIO 17,18,19 output value

3. MCU changes LCD language message according to **L** setting

SRAM mirror memory area (volatile memory)

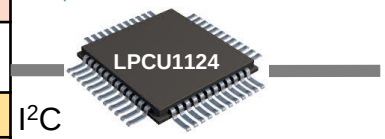
EEPROM memory area (non-volatile memory)

Training Mobile Knowledge

Application logic for zero power configuration use case (III)

Page address (HEX)	Byte number within a page			
	0	1	2	3
0x00				
…	…… ……			
0x34				B1
0x35	B2	B3	L	C1
0x36	C1	C2	C2	C3
0x37	C3			
…	… … … …			
0xF0				
…	….			
0xFB				
0xFC				
0xFD				
0xFE				
0xFF				

5. MCU updates light bulb switching counters (+1)



I²C

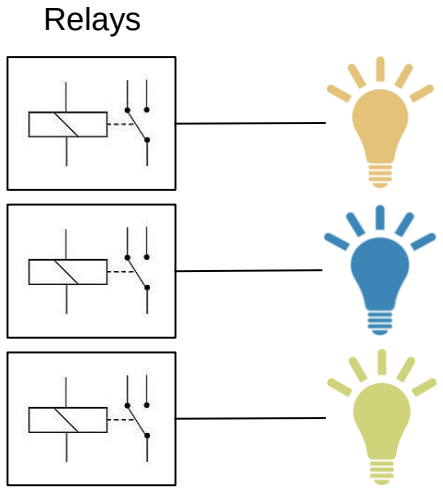


Fig. Simplified NTAG I²C *plus* memory map (NFC interface perspective)

EEPROM memory map (NFC perspective)
Page address 0x35 - 0x36
Page address 0x36
Page address 0x36 – 0x37

(**C1**) Stores light bulb 1 switching counter (range 0x0000 – 0xFFFF)

(**C2**) Stores light bulb 2 switching counter (range 0x0000 – 0xFFFF)

(**C3**) Stores light bulb 3 switching counter (range 0x0000 – 0xFFFF)

SRAM mirror memory area (volatile memory)

EEPROM memory area (non-volatile memory)

Training Mobile Knowledge

HOW THE NTAG I²C *PLUS* IS USED

PRODUCT DIAGNOSIS DEMO USE CASE

Application logic for product diagnosis use case

Read diagnosis

1. Phone reads switch counters values from EEPROM

Page address (HEX)	Byte number within a page			
	0	1	2	3
0x00				
…	…… ……			
0x34				B1
0x35	B2	B3	L	C1
0x36	C1	C2	C2	C3
0x37	C3			
…	… … … …			
0xF0				
…	….			
0xFB				
0xFC				
0xFD				
0xFE				
0xFF				

Fig. Simplified NTAG I²C *plus* memory map (NFC interface perspective)

EEPROM memory map (NFC perspective)
Page address 0x35 - 0x36
Page address 0x36
Page address 0x36 – 0x37

(**C1**) Stores light bulb 1 switching counter (range 0x0000 – 0xFFFF)

(**C2**) Stores light bulb 2 switching counter (range 0x0000 – 0xFFFF)

(**C3**) Stores light bulb 3 switching counter (range 0x0000 – 0xFFFF)

SRAM mirror memory area (volatile memory)

EEPROM memory area (non-volatile memory)

Training Mobile Knowledge

Application logic for product diagnosis use case

Reset diagnosis

1. Phone resets switch counters to zero by writing to EEPROM

Page address (HEX)	Byte number within a page			
	0	1	2	3
0x00				
…	…… ……			
0x34				B1
0x35	B2	B3	L	0x00
0x36	0x00	0x00	0x00	0x00
0x37	0x00			
…	… … … …			
0xF0				
…	….			
0xFB				
0xFC				
0xFD				
0xFE				
0xFF				

Fig. Simplified NTAG I²C *plus* memory map (NFC interface perspective)

EEPROM memory map (NFC perspective)
Page address 0x35 - 0x36
Page address 0x36
Page address 0x36 – 0x37

(**C1**) Stores light bulb 1 switching counter (range **0x0000** – 0xFFFF)

(**C2**) Stores light bulb 2 switching counter (range **0x0000** – 0xFFFF)

(**C3**) Stores light bulb 3 switching counter (range **0x0000** – 0xFFFF)

SRAM mirror memory area (volatile memory)

EEPROM memory area (non-volatile memory)

Training Mobile Knowledge

HOW THE NTAG I²C *PLUS* IS USED

FIRMWARE UPGRADE DEMO USE CASE

Application logic for firmware upgrade use case

Secondary bootloader application

- NXP's LPC11U24 microcontroller provides two methods to update the flash contents:
  - **In-system-programming (ISP)**: This method is used when we flash new contents using the USB port and a computer (drag-and-drop a binary file).
  - **In application programming (IAP):** Programming is performed by erase and write operation on the on-chip flash memory, as directed by the end-user application code

IAP methods are used for firmware upgrade via NFC

- A **secondary bootloader** application is a piece of code which allows a user application code to be downloaded via alternative channels to the USB (i.e.: firmware upgrade via NFC).

The LPC11U24 flash memory is divided into 8 sectors of 4 Kb each one (total 32kB)

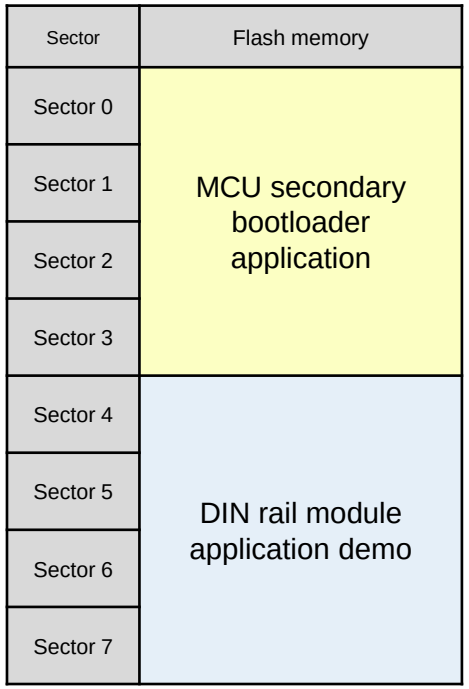


Fig. Simplified LPC11U24 flash memory map

Training Mobile Knowledge

Application logic for firmware upgrade use case

1. Phone uses SRAM to transmit the binary file in pieces of 64 bytes

Page address (HEX)	Byte number within a page			
	0	1	2	3
0x00				
…	…… ……			
0x34				
0x35				
0x36				
0x37				
…	… … … …			
0xF0	Transmission of 64 byte data via SRAM in pass-through mode			
0xFC				
0xFD				
0xFE				
0xFF				

Fig. Simplified NTAG I²C *plus* memory map (NFC interface perspective)

I²C

Sector	Flash memory
Sector 0	MCU secondary bootloader application
Sector 1	
Sector 2	
Sector 3	
Sector 4	IAP functions
Sector 5	
Sector 6	Flash memory area under upgrade
Sector 7	

Fig. Simplified LPC11U24 flash memory map

Data flow direction (SRAM in pass through mode)

SRAM mirror memory area (volatile memory)

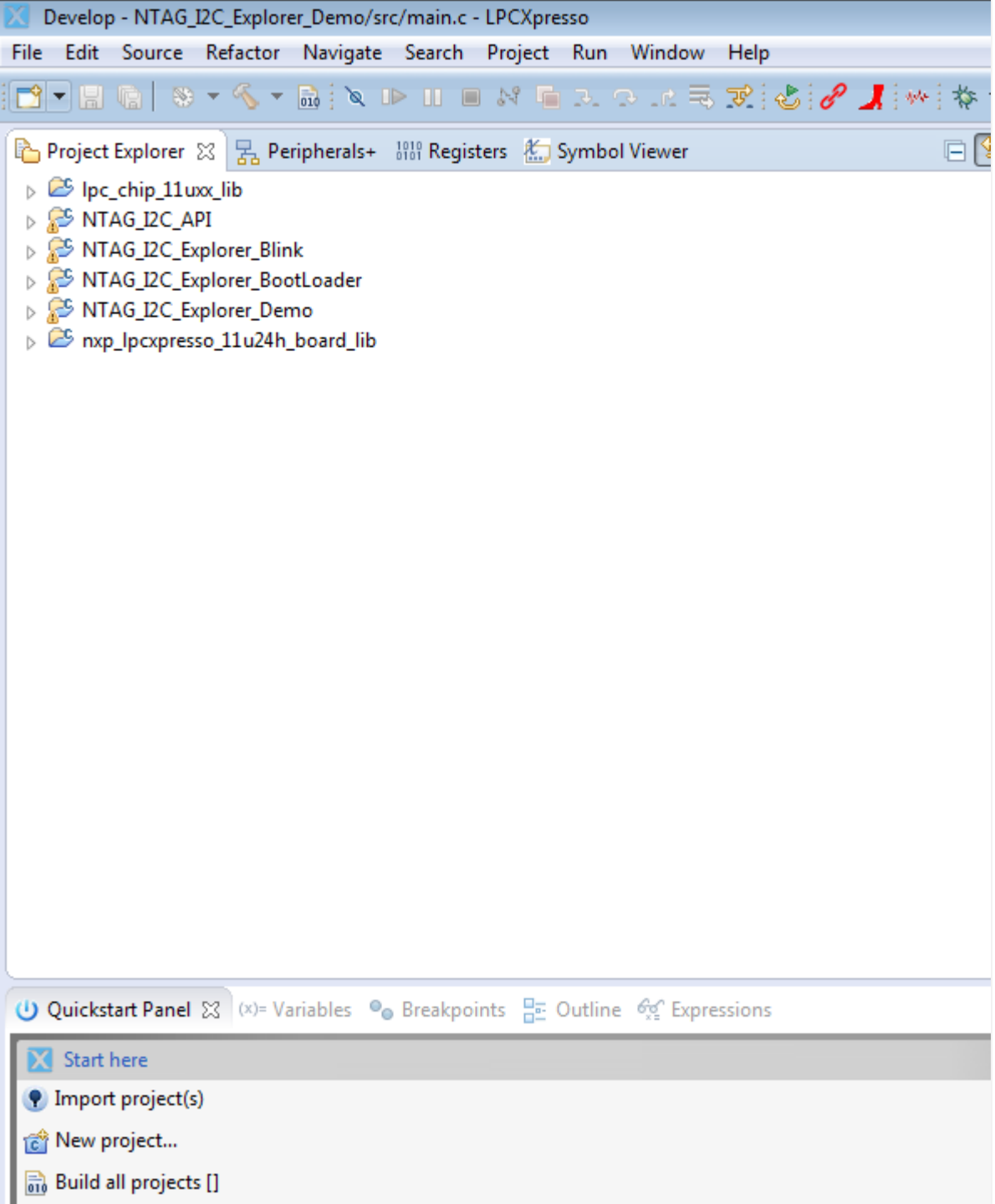
EEPROM memory area (non-volatile memory)

Firmware upgrade can be performed with the DIN module off using NTAG I2C plus **energy harvesting** feature

Training Mobile Knowledge

SOFTWARE INTEGRATION

MCU / EMBEDDED



MCU code overview

Lpc_chip_11uxx_lib & nxp_lpcxpresso_11u24h_board_lib

LPCOpen software libraries (drivers and middleware) supporting development on top of LPC MCU solutions. These two libraries bring support to LPC11U24 LPCXpresso board

NTAG_I2C_ API

Implements the NTAG I²C *plus* command set and offers an API to developers to communicate with NTAG I²C *plus* from the I²C interface.

NTAG_I2C_Explorer_Blink

Sample project that sets into blinking mode the NTAG I2C Explorer board LED as soon as an RF field is detected.

NTAG_I2C_Explorer_Bootloader

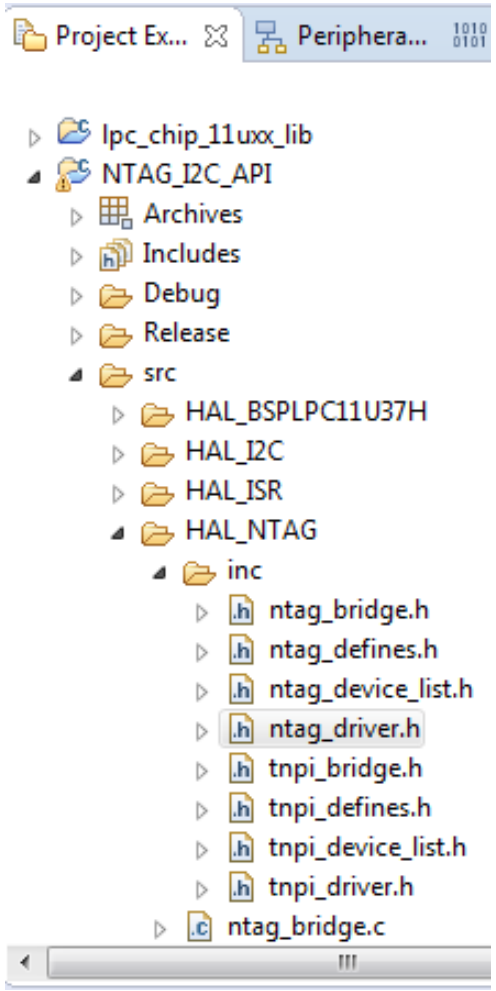
Implements the secondary boot loader application. It is flashed at on-chip flash memory address starting at 0x0000 0000 and it is the first application to be executed after MCU boots.

NTAG_I2C_Explorer_Demo

Implements the logic supporting the DIN rail module demo application for Zero power operation, diagnostics and firmware upgrade via NFC use cases.

* Reference MCU code can be downloaded from:
http://www.nxp.com/documents/software/SW3647.zip

Useful functions dealing with NTAG I²C *plus* from I²C interface



Memory operations (I²C side)

- **NTAG_ReadBytes** (NTAG_HANDLE_T **ntag**, uint16_t **address**,uint8_t *bytes, uint16_t **len**);
- **NTAG_WriteBytes**(NTAG_HANDLE_T **ntag**, uint16_t **address**, const uint8_t *bytes, uint16_t **len**);

e.g.: Read SRAM memory

- **NTAG_ReadBytes**(ntag_handle, NFC_MEM_ADDR_START_SRAM, sram_buf, NFC_MEM_SRAM_SIZE);

Register operations

- **NTAG_ReadRegister** (NTAG_HANDLE_T **ntag**, uint8_t **reg**, uint8_t ***val**);
- **NTAG_WriteRegister**(NTAG_HANDLE_T **ntag**, uint8_t **reg**, uint8_t mask, uint8_t **val**);

e.g. Detect RF field presence

- **NTAG_ReadRegister**(ntag_handle, NFC_MEM_OFFSET_NS_REG, &reg);

Setting SRAM for pass-throug mode operation

- **NTAG_SetPthruOnOff**(NTAG_HANDLE_T **ntag**, BOOL **on**)
- **NTAG_SetTransferDir**(NTAG_HANDLE_T **ntag**, NTAG_TRANSFER_DIR_T **dir**)

E.g. Set transfer direction RF → I2C

- **NTAG_SetTransferDir**(ntag_handle, **RF_TO_I2C**);

… and more

NTAG_I2C_Explorer_bootloader application flow

Power on reset

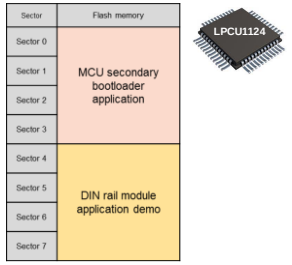
HW setup and initialization

Check buttons

No button pressed

Enter DIN rail user application

Boot jump to Sector 4



Right button pressed

Enter firmware upgrade mode

Is binary file transmission finished?

No

Read SRAM data (64 bytes)

Received sector size?

No

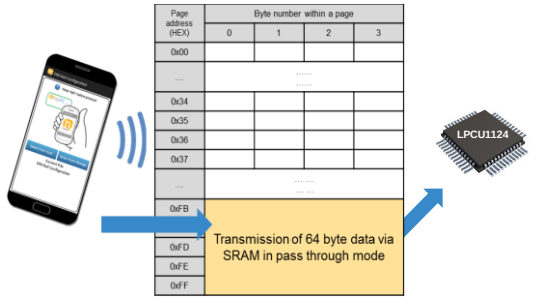
Yes

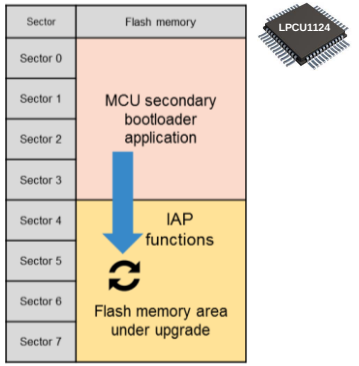
Flash MCU sector

IAP functions execution

Yes

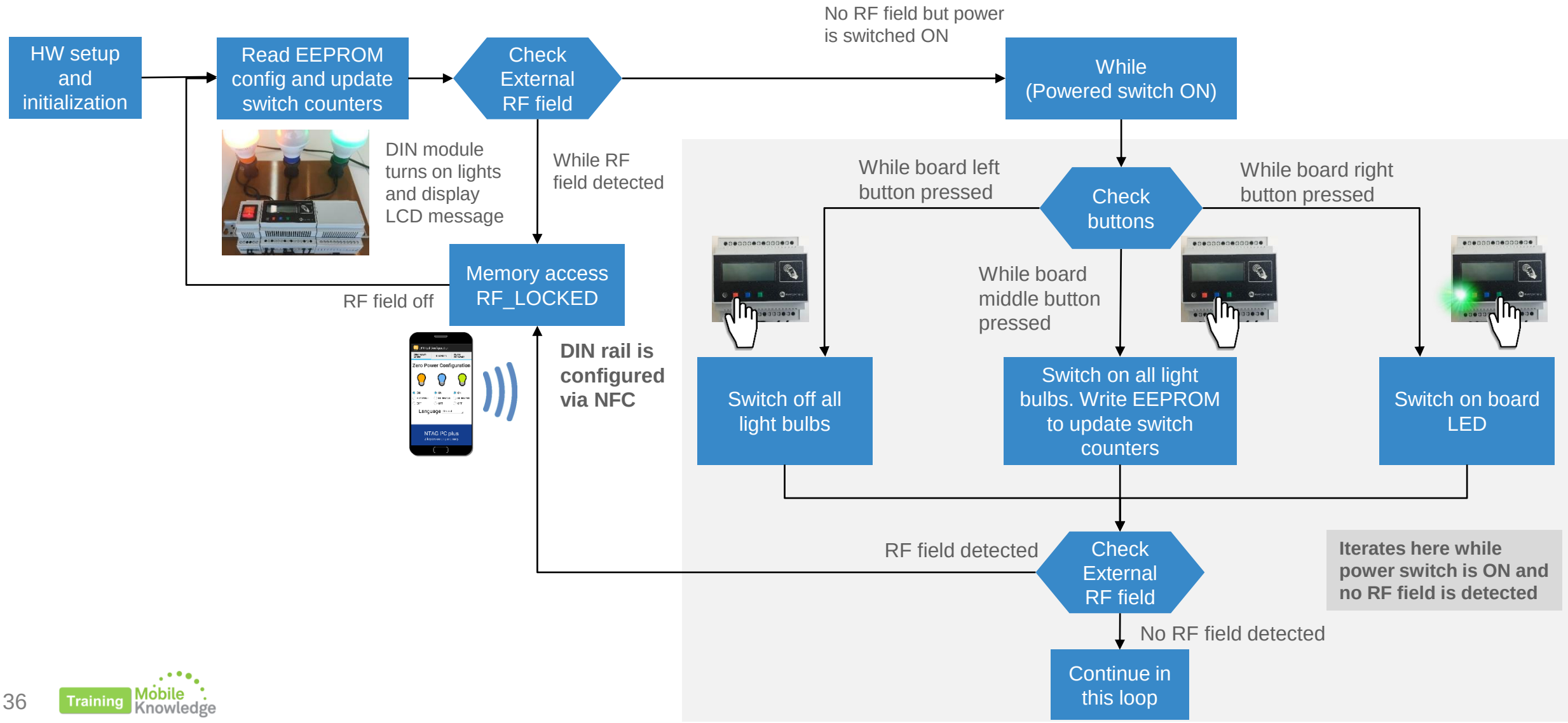
Show flash operation status





Training Mobile Knowledge

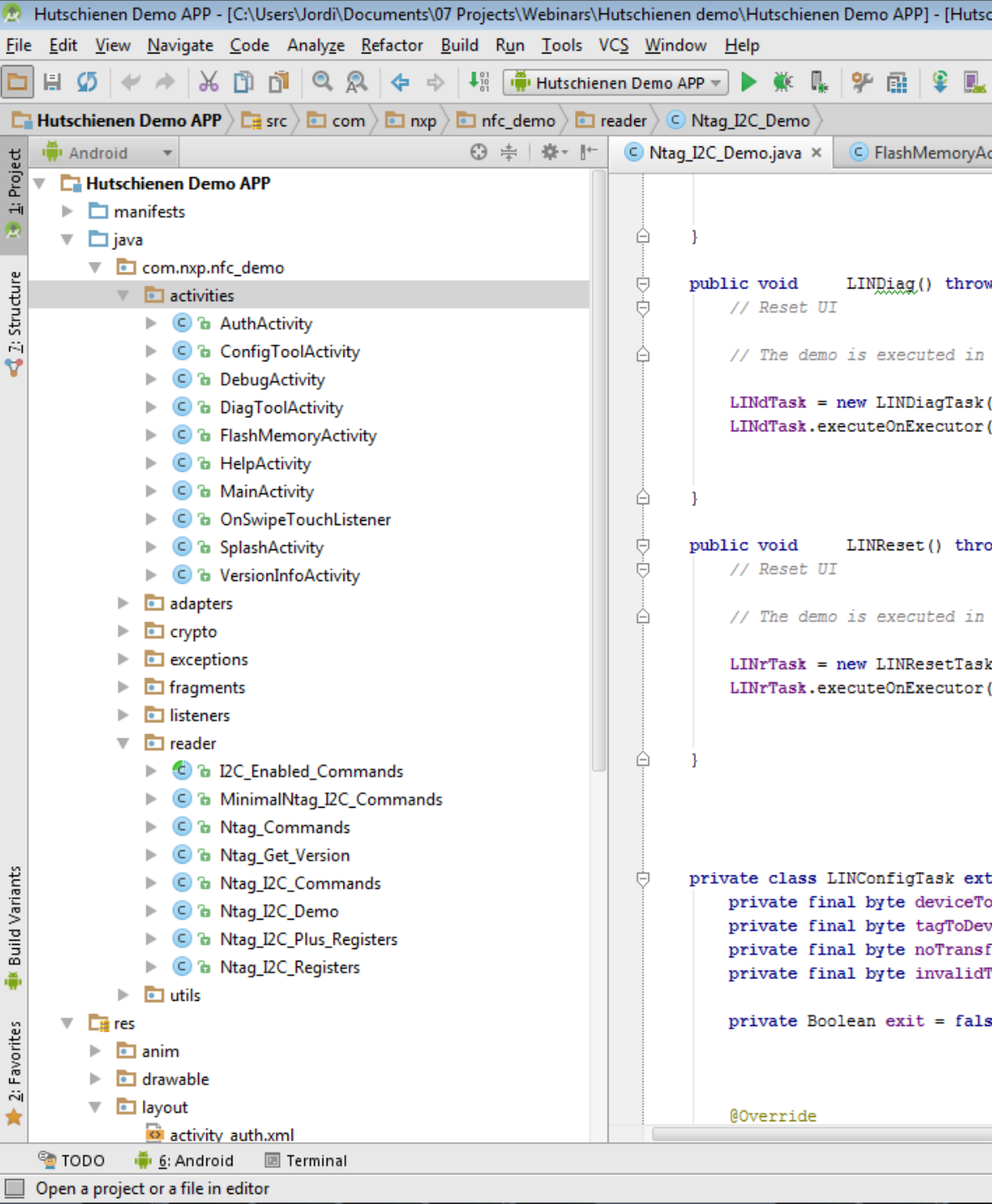
NTAG_I2C_Explorer_demo application flow



Training Mobile Knowledge

SOFTWARE INTEGRATION

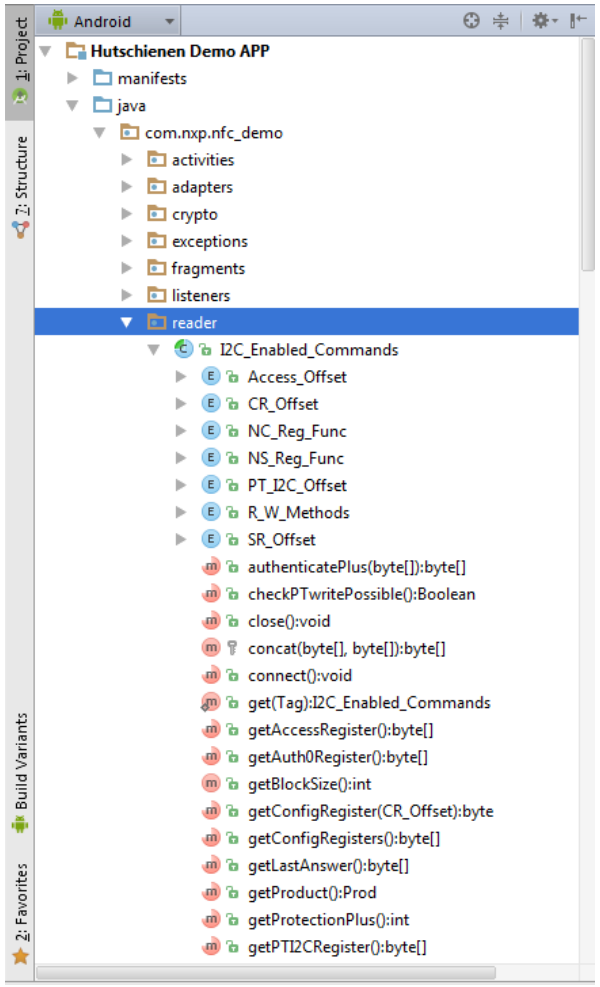
PHONE / NFC DEVICE



NFC DIN Rail Android demo app code overview

- The Android application is intended to operate on devices running Android version 4.0 and beyond.
- It can be imported in Android Studio IDE.
- Classes and methods are implemented according NTAG I²C plus datasheet.
- NTAG I²C *plus* command set is implemented and an API for developers is offered, so they may communicate with NTAG I²C *plus* from the NFC interface.

Useful functions dealing with NTAG I²C *plus* from RF interface



Memory operations (RF side)

- `public void write(byte[] data, byte blockNr)`
- `public byte[] read(byte blockNr)`
- `public byte[] fast_read(byte startAddr, byte endAddr)`
- `public void fast_write(byte[] data, byte startAddr, byte endAddr)`
- `public void SectorSelect(byte sector)`
- `public byte[] pwdAuth(byte[] pwd)`

Register operations

- `public abstract byte getSessionRegisters()`
- `public abstract byte getConfigRegisters()`
- `public abstract void writeConfigRegisters(byte NC_R, byte LD_R, byte SM_R,byte WD_LS_R, byte WD_MS_R, byte I2C_CLOCK_STR)`

Additional functions to deal with SRAM (previous ones can be used as well)

- `public abstract byte[] readSRAMBlock()`
- `public byte[] readSRAM(int blocks, R_W_Methods method)`
- `public void writeSRAM(byte[] data, R_W_Methods method, WriteSRAMListener listener)`
- `public abstract void writeSRAMBlock(byte[] data, WriteSRAMListener listener)`

… and more

Training Mobile Knowledge

NFC DIN Rail Android demo application flow

Splash activity

Main activity

Zero power configuration activity

Select configuration

With a tap

Write EEPROM memory

Diagnostics activity

With a tap

Read / Write EEPROM memory

Flashing activity

Select binary file to be flashed

Divide binary file in chunks of 64 bytes

With a tap

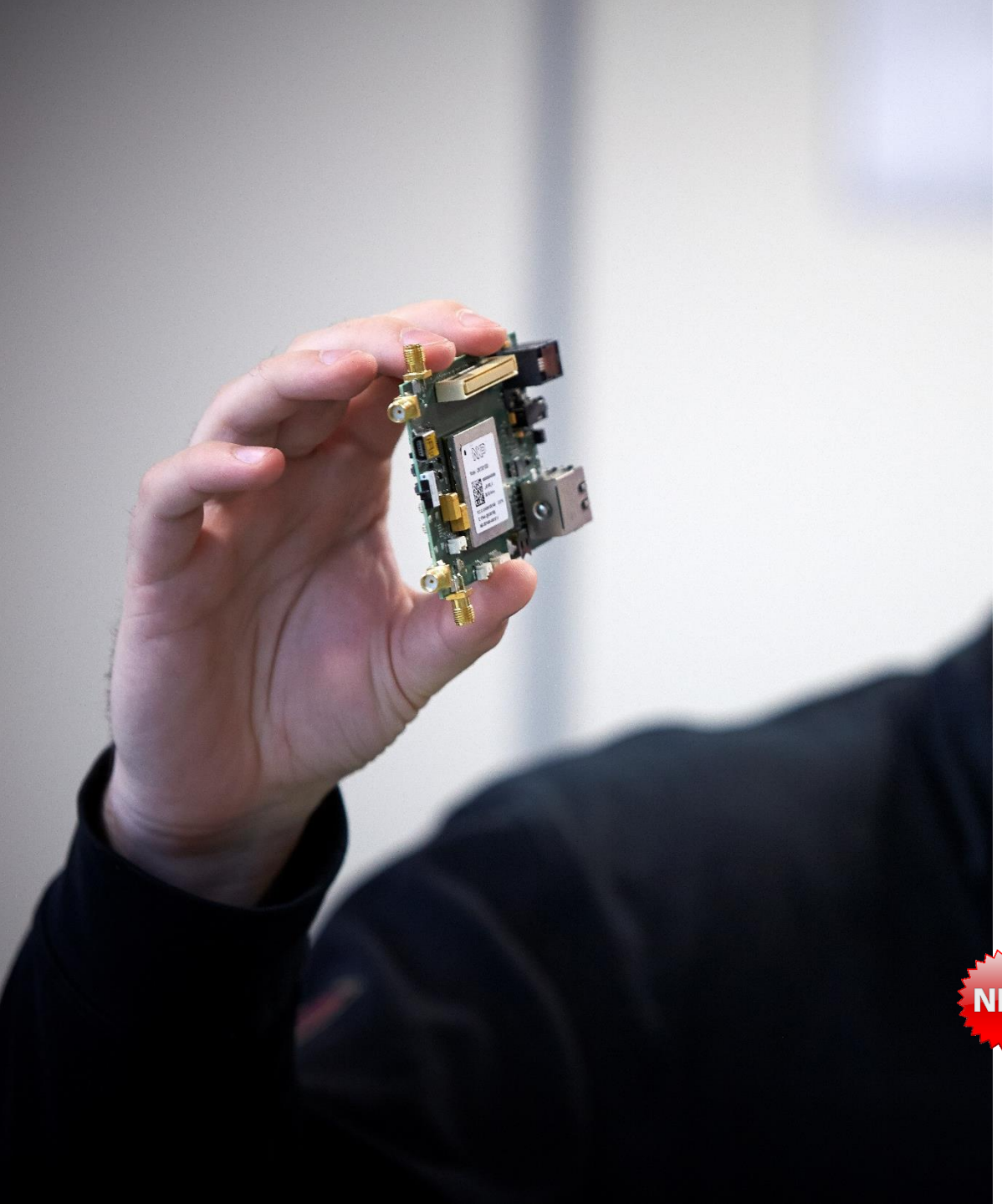
Write SRAM blocks until entire file is transferred

Training Mobile Knowledge

EVERYTHING YOU NEED TO BUILD YOUR

PARAMETRIZATION VIA NFC

SOLUTION IS HERE!



Summary of available resources

- Android reference source code
  - http://www.nxp.com/documents/software/SW3648.zip
- MCU reference source code
  - http://www.nxp.com/documents/software/SW3647.zip
- NTAG I²C *plus* Explorer kit
  - http://www.nxp.com/products/wireless-connectivity/nfc-and-reader-ics/connected-tag-solutions/ntag-ic-plus-explorer-kit-with-nfc-reader-development-kit:OM5569-NT322ER
- NTAG I²C *plus* Flex kit with additional antennas
  - http://www.nxp.com/products/wireless-connectivity/nfc-and-reader-ics/connected-tag-solutions/ntag-ic-plus-flex-kit-containing-additional-flex-antennas:OM5569-NT322F
- HW design files
  - http://www.nxp.com/documents/software/SW3641.zip
  - http://www.nxp.com/documents/software/SW3639.zip
  - http://www.nxp.com/documents/software/SW3638.zip
- DIN rail module source code
  - https://community.nxp.com/docs/DOC-333834

NEW

MobileKnowledge

Software development in Android and iOS
Embedded software for MCUs
JCOP, Java Card operating Systems
Hardware design and development
Digital, analog, sensor acquisition, power management
Wireless communications WiFi, ZigBee, Bluetooth, BLE
Contactless antenna RF design, evaluation and testing

MIFARE® product-based applications
End-to-end systems, readers and card-related designs
EMVco applications
Readers, cards, design for test compliancy (including PCI)
Secure Element management
GlobalPlatform compliant backend solutions
Secure services provisioning OTA, TSM services

We help companies leverage the mobile and **contactless revolution**

MobileKnowledge
Roc Boronat 117, P3M3
08018 Barcelona
(Spain)

Get in touch with us
www.themobileknowledge.com
mk@themobileknowledge.com

Parametrization via NFC

Jordi Jofre (Speaker)
Angela Gemio (Host)

Thank you for your kind attention!

Please remember to fill out our **evaluation survey** (pop-up)

Check your email for **material download** and on-demand **video** addresses

Please check NXP and MobileKnowledge websites for **upcoming webinars** and **training sessions**

http://www.nxp.com/support/classroom-training-events:CLASSROOM-TRAINING-EVENTS

www.themobileknowledge.com/content/knowledge-catalog-0