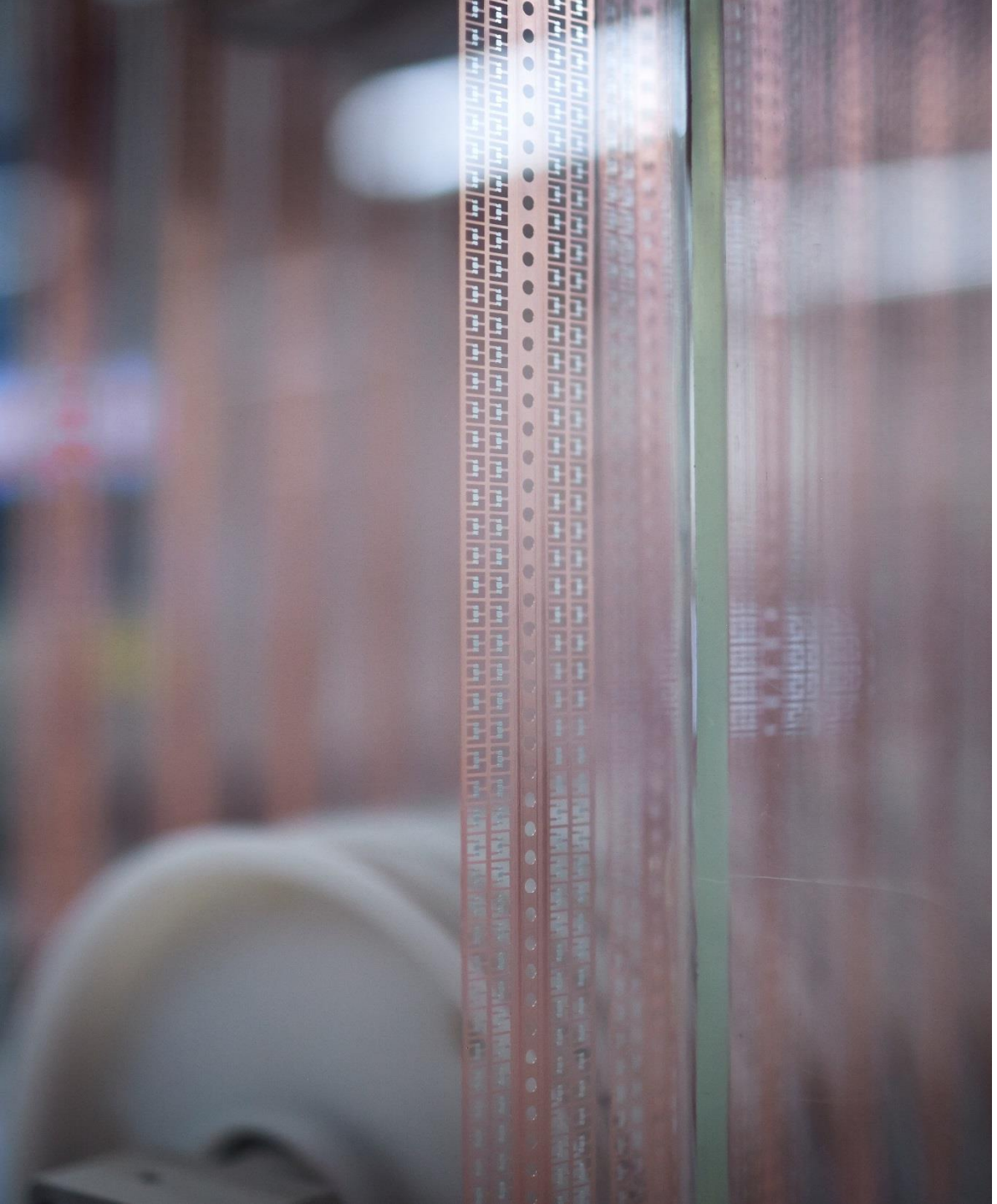
NFC FOR CONSUMABLES AND ACCESSORIES

WEBINAR SERIES:
HOW TO DEVELOP NFC APPLICATIONS

JORDI JOFRE
NFC READERS
NFC EVERYWHERE
22/02/2018

SECURE CONNECTIONS
FOR A SMARTER WORLD



Agenda

- NFC for product authentication & identification
- NFC portfolio for product authentication & identification
- NFC Nutshell Kit
- Consumable authentication sample application logic

NFC for product authentication & identification

NFC for product authentication & identification

Combat counterfeits by authenticating accessories

Create more interactive and personal experiences

Order branded replacements/ consumables with a single tap

Automatically choose the right tool every time

NFC Benefits

- Adjust settings of the main unit based on the accessory attached
- Ensure authenticity of the consumable / fight counterfeits
- Improve accuracy by storing calibration data on the tag
- Identify users and immediately provide personalized settings
- Send notifications when accessories are nearing replacement

Training Mobile Knowledge

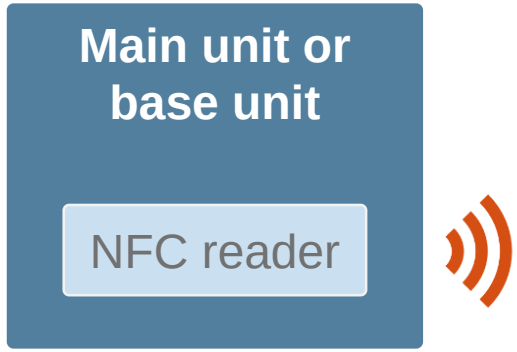
How NFC works in product authentication & identification

Use case

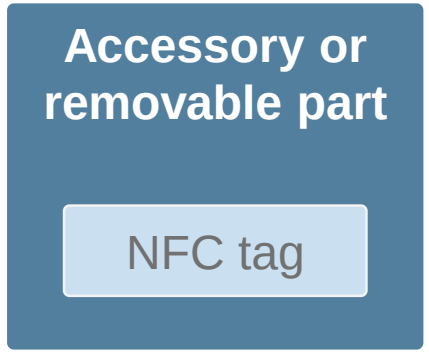
- Validate the originality of the consumable / accessory
- Optionally, configure the device with related settings

Goals

- Ensure originality including recurring revenues on consumables
- Enhance consumer experience and convenience
- Ensure product safety



Data read by NFC Reader inside the base unit, e.g. fridge, blender, then sent to MCU



NFC Tag in the removable part, e.g. brush head, water or air filter, …

Training Mobile Knowledge

NFC success stories

High-end blender

NFC reader: in the base unit

NFC tag: in the jug/container

Application: Check lid is closed before starting configuration: settings

Face brush

NFC reader: in the handle

NFC tag: in the brush heads

Application: Automatically configure the brush speed & spinning parameters

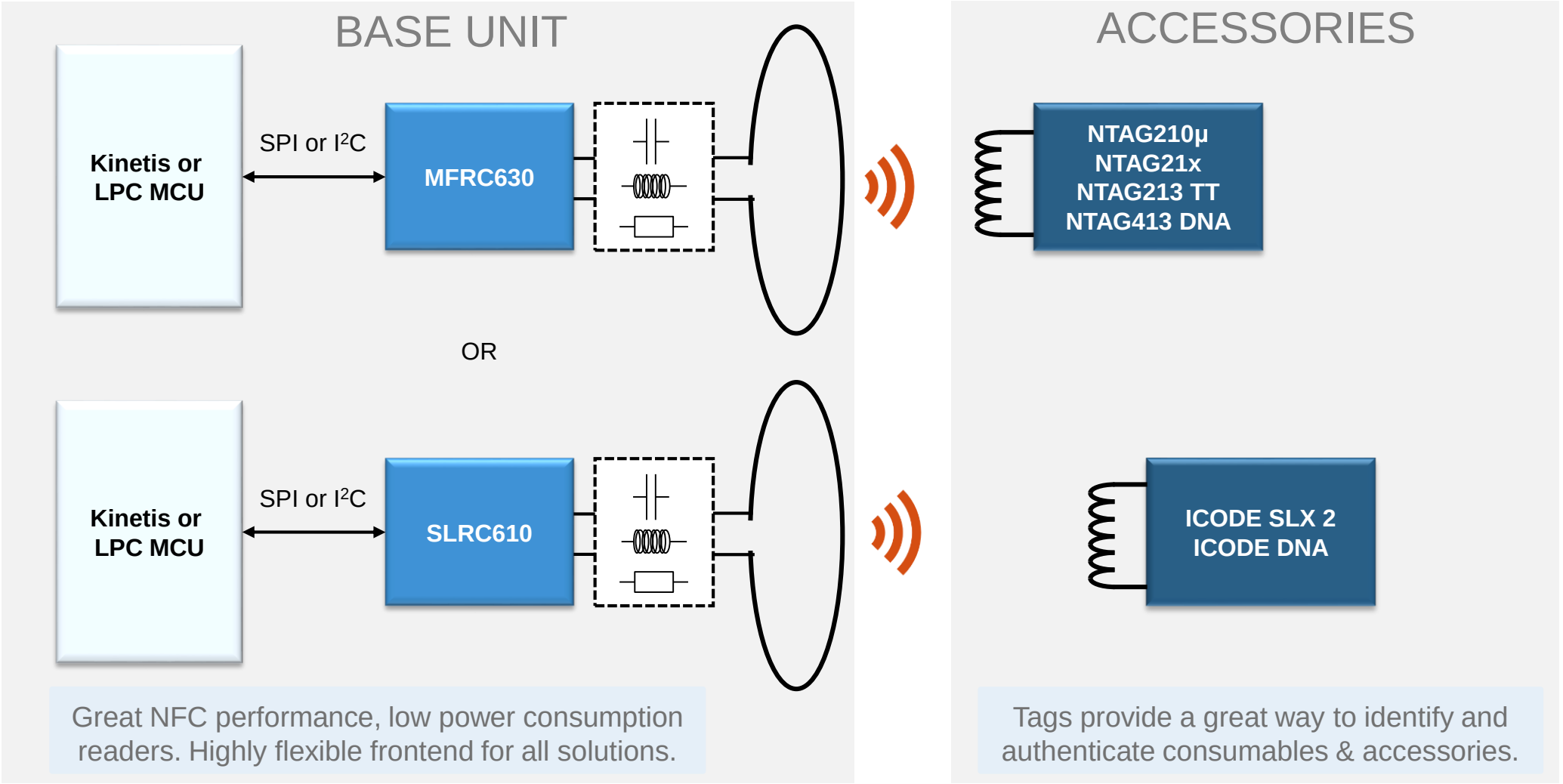
Water filter for fridges

NFC reader: in the fridge base

NFC tag: in the water filter

Application: Check that the right & genuine water filter is in place

How to implement the use case



Great NFC performance, low power consumption readers. Highly flexible frontend for all solutions.

Tags provide a great way to identify and authenticate consumables & accessories.

Training Mobile Knowledge

Solution selection guidelines

What do you need to achieve?

- Brand protection (only original supplies work), automation, safety / security issues, etc.

Which are your security needs?

- Identification, authentication, signature, integrity check, encrypted communication, etc.

What reading distance do you need?

- A few cm, a few tenths of cm, etc.

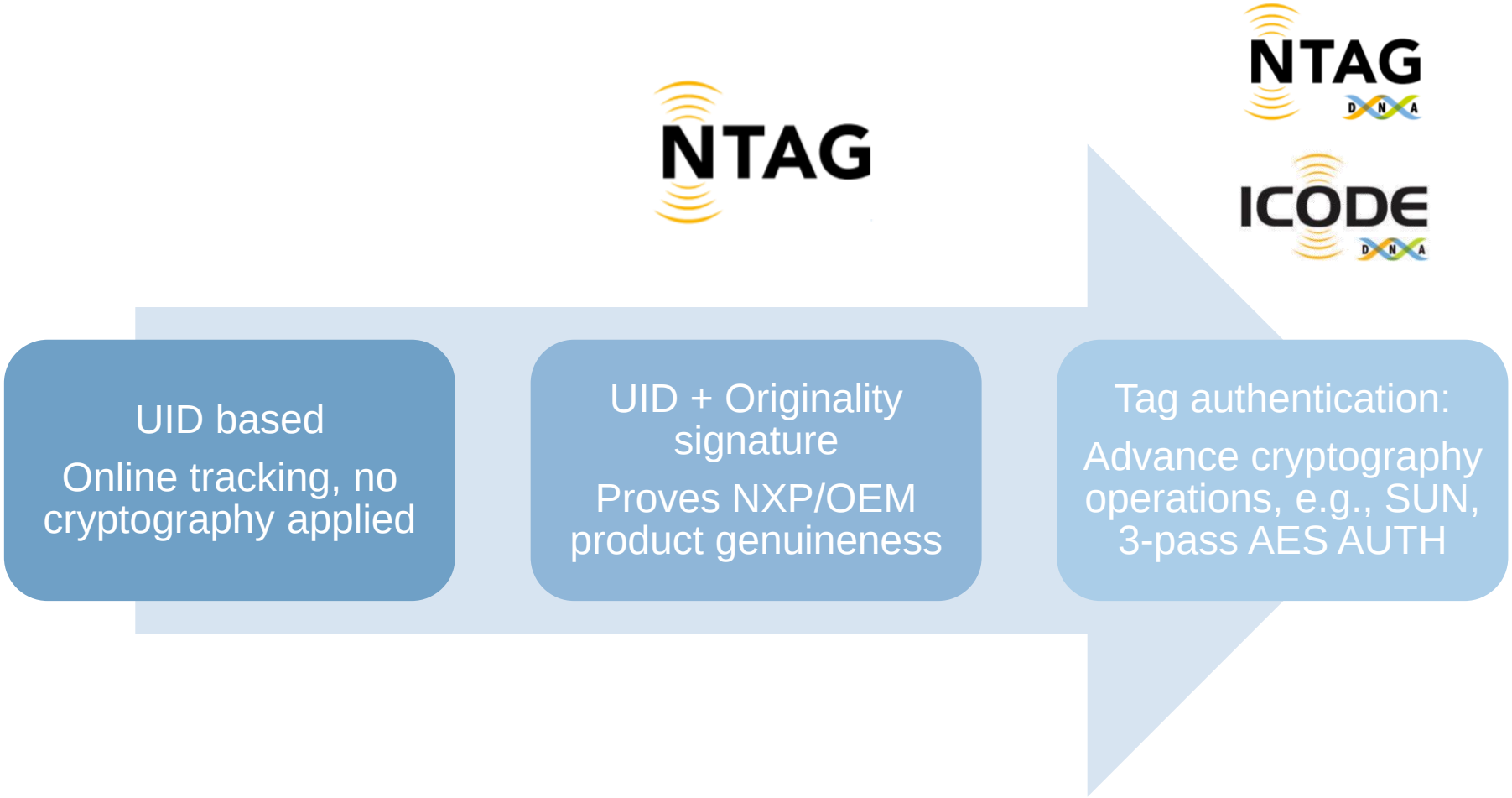
Do you have space constraints in the product?

- Directly impacts the antenna size.

Training Mobile Knowledge

NXP portfolio for product authentication & identification

NFC security features in NXP portfolio



Training Mobile Knowledge

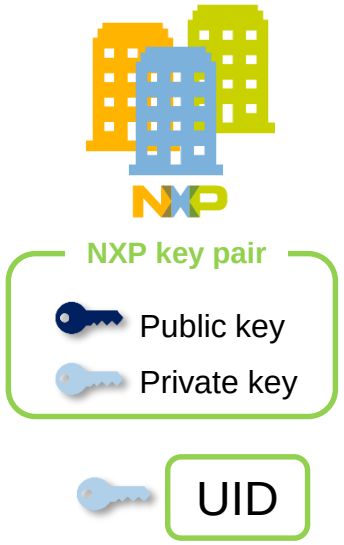
Originality Signature generation during IC production

NTAG

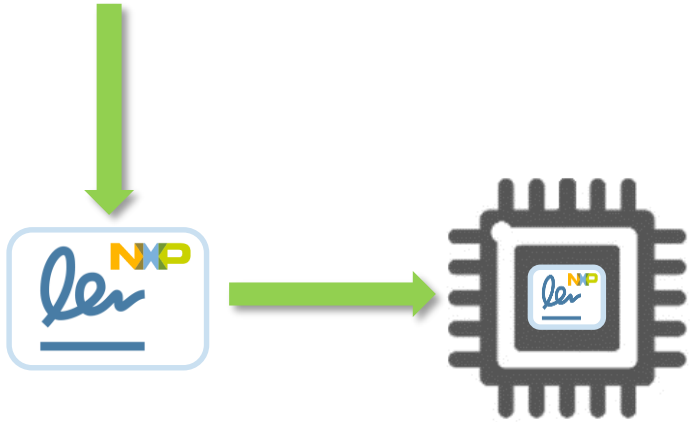
NTAG

ICODE

1. Unique ID per IC is signed by NXP



2. Signature is stored inside the IC



Training Mobile Knowledge

Originality Signature verification

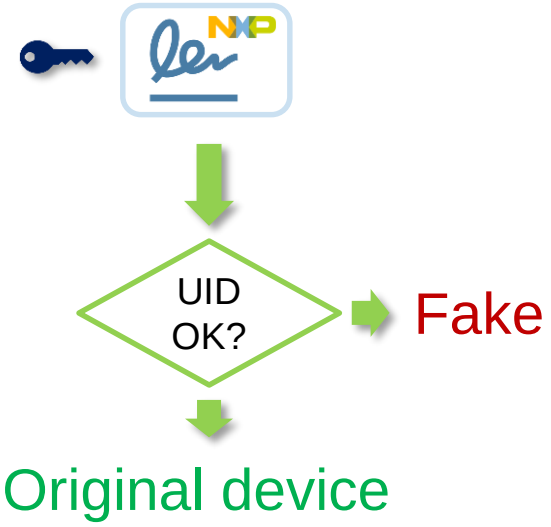
1. UID and signature are retrieved

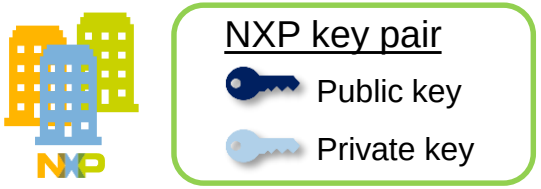
NTAG

NTAG

ICODE

2. Signature is verified with the IC UID





Training Mobile Knowledge

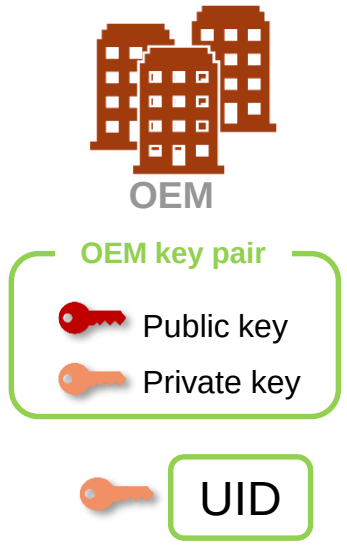
OEM Customizable originality signature

NTAG (*)

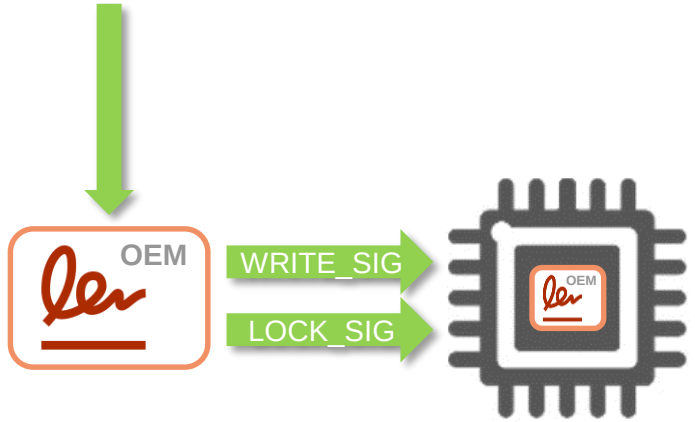
ICODE DNA

(*) Only some NTAG family members support this feature

1. Unique ID per IC is signed by OEM



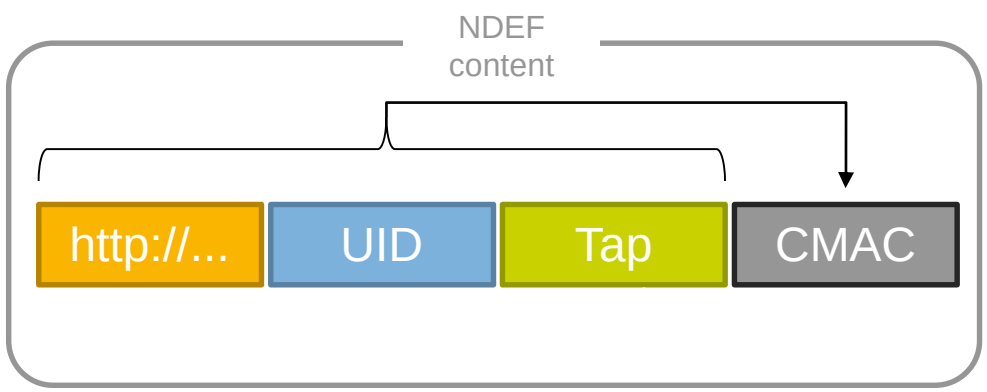
2. Signature is stored and locked inside the IC

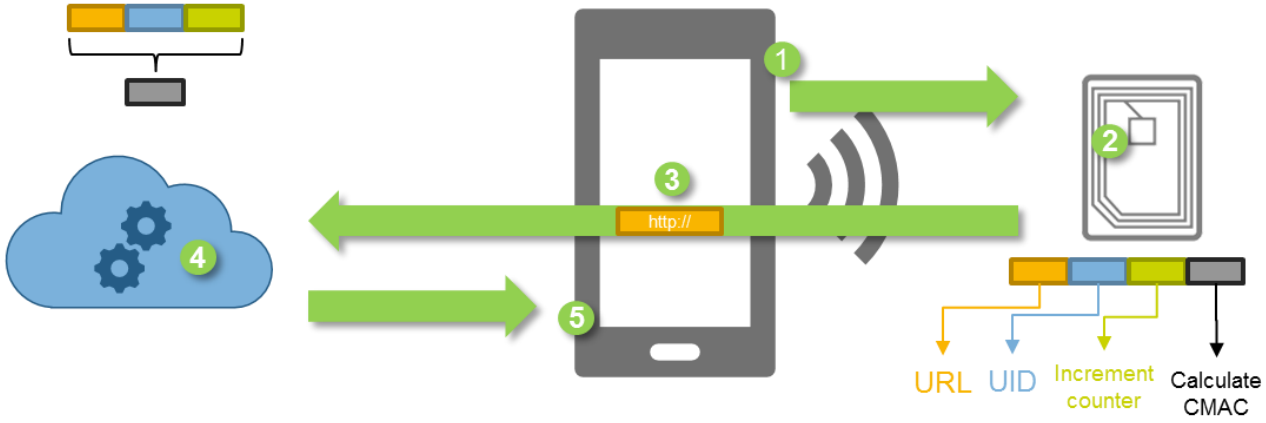


Training Mobile Knowledge

Secure Unique NFC Message (SUN)

- Unique NDEF message generated each tap
- Incremental NFC counter each tap available
- Direct connection to web-service with no app required
- AES based dynamic CMAC as part of the NDEF data





AES 3-pass mutual authentication

- Tag and reader authentication
- 3 AES 128-bit application keys available
- Used key is known to both receiver and sender

Advanced Encryption Standard is a specification for symmetric encryption

ISO/IEC 29167 defines procedures for tag authentication using AES-128

Training Mobile Knowledge

AES 3-pass mutual authentication

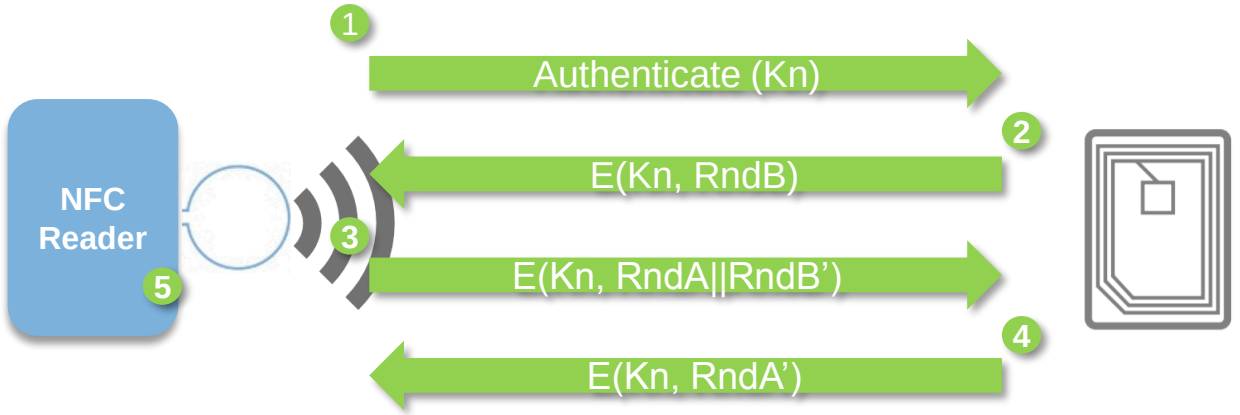
❶ Reader sends authentication command with key number to use

❷ Tag generates random challenge, encrypts it and sends the response

❸ Reader decrypts the challenge, combines it with a new challenge, encrypts the result and sends the response

❹ Tag decrypts the message, and sends the reader's challenge encrypted

❺ If all challenges have been successful, both ends are now authenticated and have a shared secret



Training Mobile Knowledge

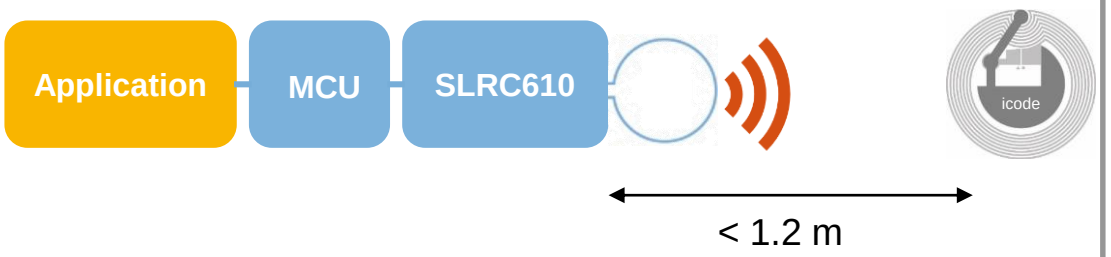
Tag Comparison

	NTAG21x	NTAG210µ	NTAG413 DNA	ICODE DNA
Type	Type 2	Type 2	Type 4	Type 5
Operating distance up to	10 cm	10 cm	10 cm	1.2 m
Originality signature	32 Bytes (NXP signature)	32 Bytes customizable	56 Bytes (NXP signature)	32 Bytes customizable
3-pass AES Auth	✗	✗	✓	✓
SUN	✗	✗	✓	✗
Memory	144-888 Bytes	64 Bytes	128 Bytes	256 Bytes

NFC frontends

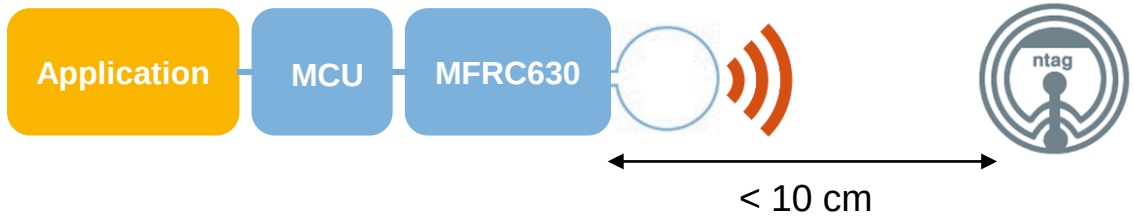
SLRC610 plus

- Multiprotocol: ISO/IEC 15693, ISO/IEC 18000-3
- Supporting vicinity products **ICODE**
- Host interfaces: SPI I²C, UART
- Separate I²C bus for a SAM



MFRC630 plus

- RF standard compliance: ISO 14443A
- Recommended solution for **NTAG** and **MIFARE® products**
- Host interfaces: SPI I²C, UART
- Separate I²C bus for a SAM



Training Mobile Knowledge

NFC Nutshell Kit

NFC Nutshell Kit introduction

Need to add NFC into your products ?

The NFC Nutshell Kit modules are specifically designed for:

- NFC technology integration / retrofitting into existing or new product designs
- Building up of NFC enabled demonstrators
- NFC technology evaluation
- Application testing and development

Developed by GMMC, the kit contains several modules covering most of NXP portfolio for NFC solutions. GMMC is an approved engineering consultant of NXP for NFC (https://nxp.surl.ms/NFC_AEC)

Training Mobile Knowledge

GMMC we deliver for your success

NXP

Benefits & features

- Nano sized modules for space constrained environments
- Flexible configuration to adapt to different conditions
- Support of most popular NXP NFC reader/writer ICs
  - CLRC663plus family, PN5180, PN7150, PN7462 family
- Compatibility with existing NXP NFC and MCU development tools
  - NFC Cockpit, RFIDDiscover, MCU Expresso, LPC Link2

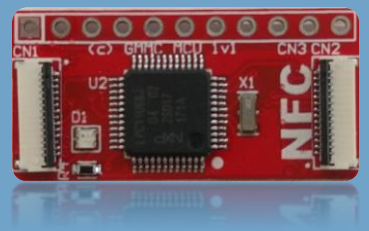
Modules

Host interfaces:

- USB Plug
- Programmable converter USB to UART, I2C, SPI
- Signal Debug Extender

Microcontrollers:

- LPC11u68 (JBD48)
- LPC1769



RF-Frontends:

- CLRC663 plus family, including SLRC610, MFRC630 MFRC631
- PN5180

RF-Frontend with integrated MCU:

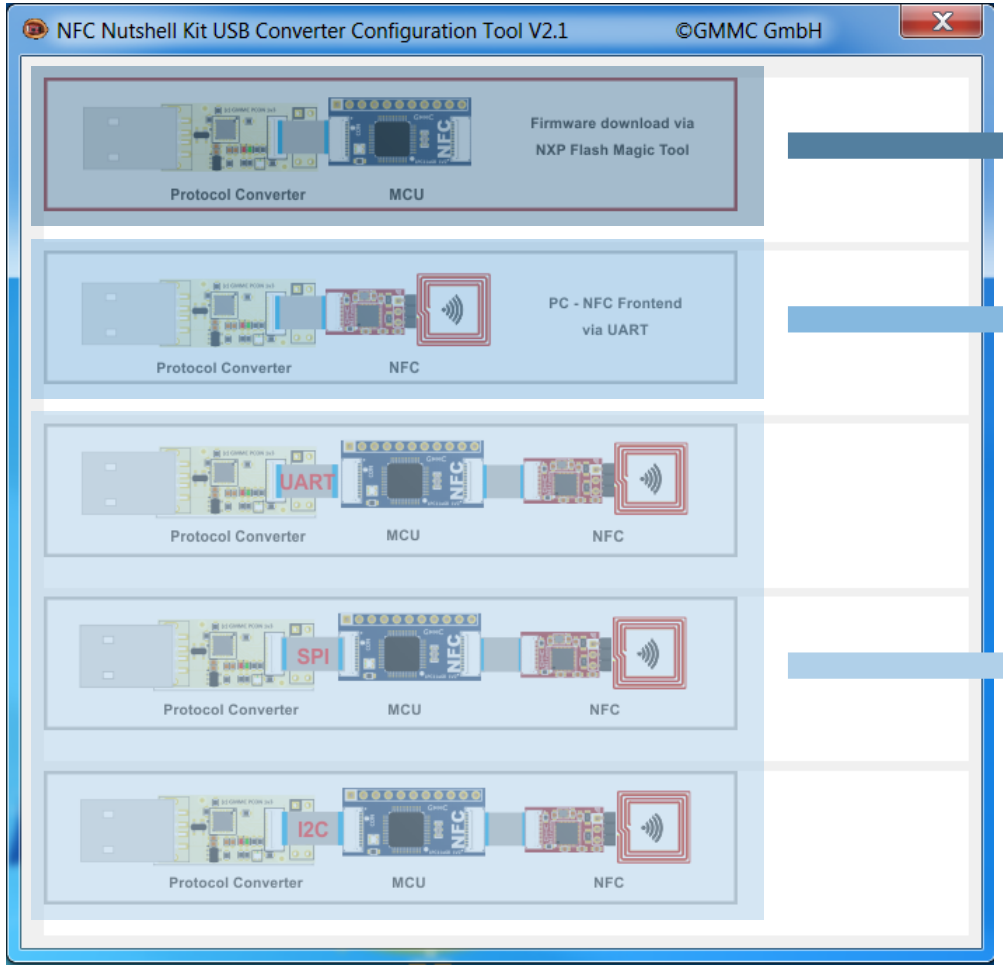
- PN7462 (Q2 2018)
- PN7150 (Q2 2018)

Antennas:

- 20x10 mm
- 20x20 mm
- 40x40 mm
- 72x48 mm
- Twisted wire connection between antenna and RF modules

Training Mobile Knowledge

Modes of operation for USB protocol converter



MCU stays in programming mode, only for MCU flashing

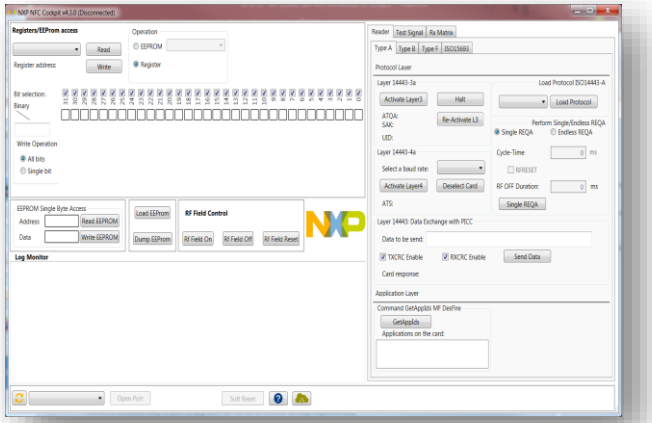
No MCU is used. The computer talks with the frontend via UART

The computer can talk with the MCU over the specified protocol

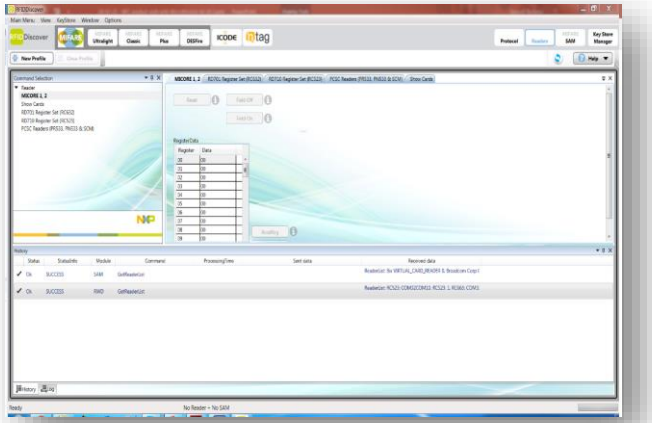
Training Mobile Knowledge

Supported NXP development tools

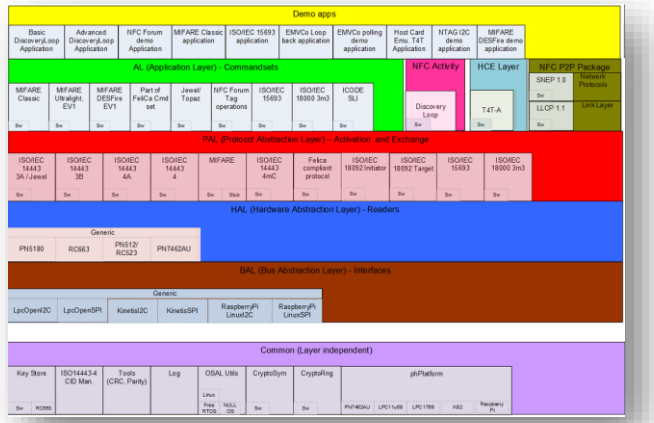
NFC Cockpit



RFIDDiscover



NFC Reader library



[More information on NFC-Cockpit]

[More information on RFIDDiscover]

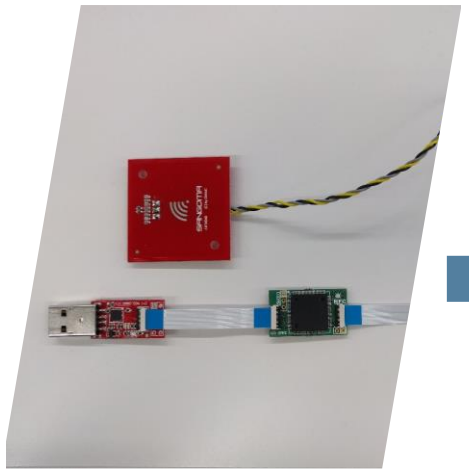
[More information on NFC Reader library]

Training Mobile Knowledge

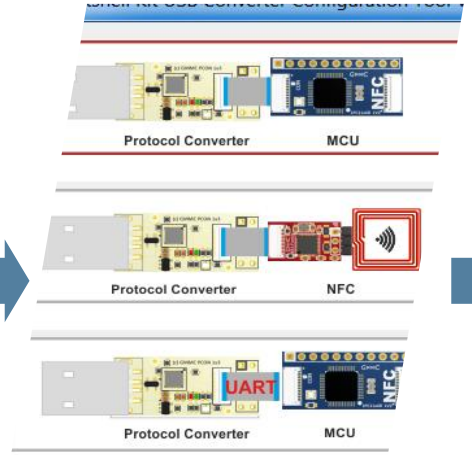
Consumable authentication

Sample application logic

Running the NFC reader library in the Nutshell Kit



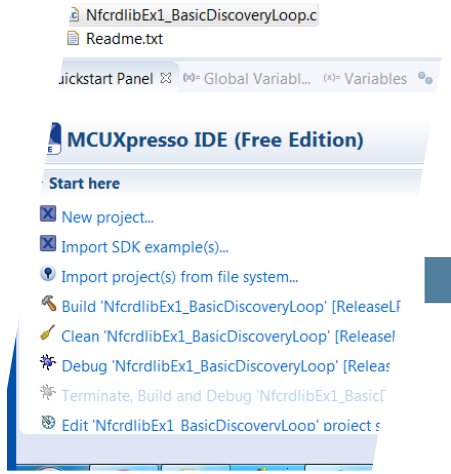
1. Prepare the hardware



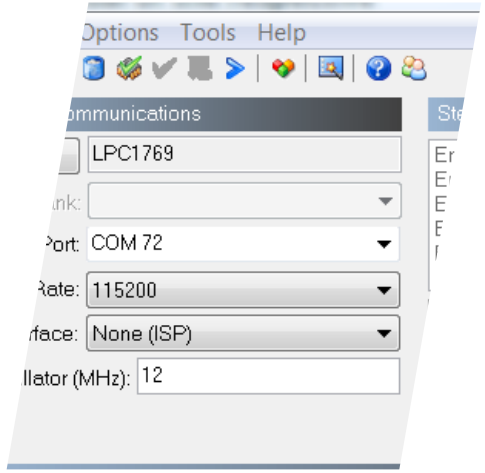
2. Configure USB converter

```
ders */
crdlibEx1_BasicDiscoveryLoop.h"

. Defines

op_Sw_DataParams_t       * pDiscLoop;

ow variables needs to be initialized acco

sens_res[2]     = {0x04, 0x00};
nfc_id1[3]      = {0xA1, 0xA2, 0xA3};
sel_res         = 0x40;
nfc_id3         = 0xFA;
poll_res[18]    = {0x01, 0xFE, 0xB2, 0x
                   0xB6, 0xB7, (
                   0xC4, 0xC5,

f PHOSAL_FREERTOS_STATIC_MEM_ALLOCATION
2_t aBasicDiscTaskBuffer[BASIC_DISC_DEMO_
/* uint32_t aBasicDiscTaskBuffer[BASIC_[
ine aBasicDiscTaskBuffer    NULL
```

3. Development



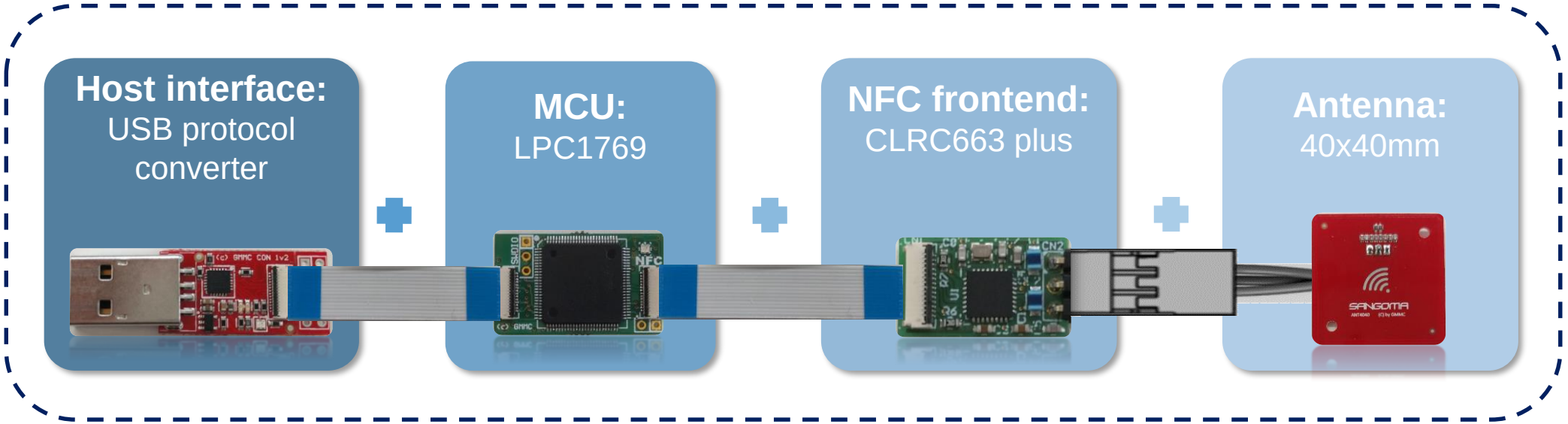
4. Build project in MCUXpresso



5. Flash the MCU image

Training Mobile Knowledge

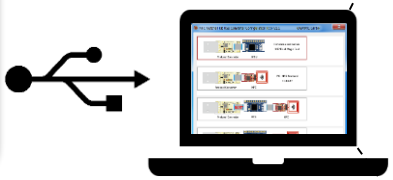
1. Prepare the hardware

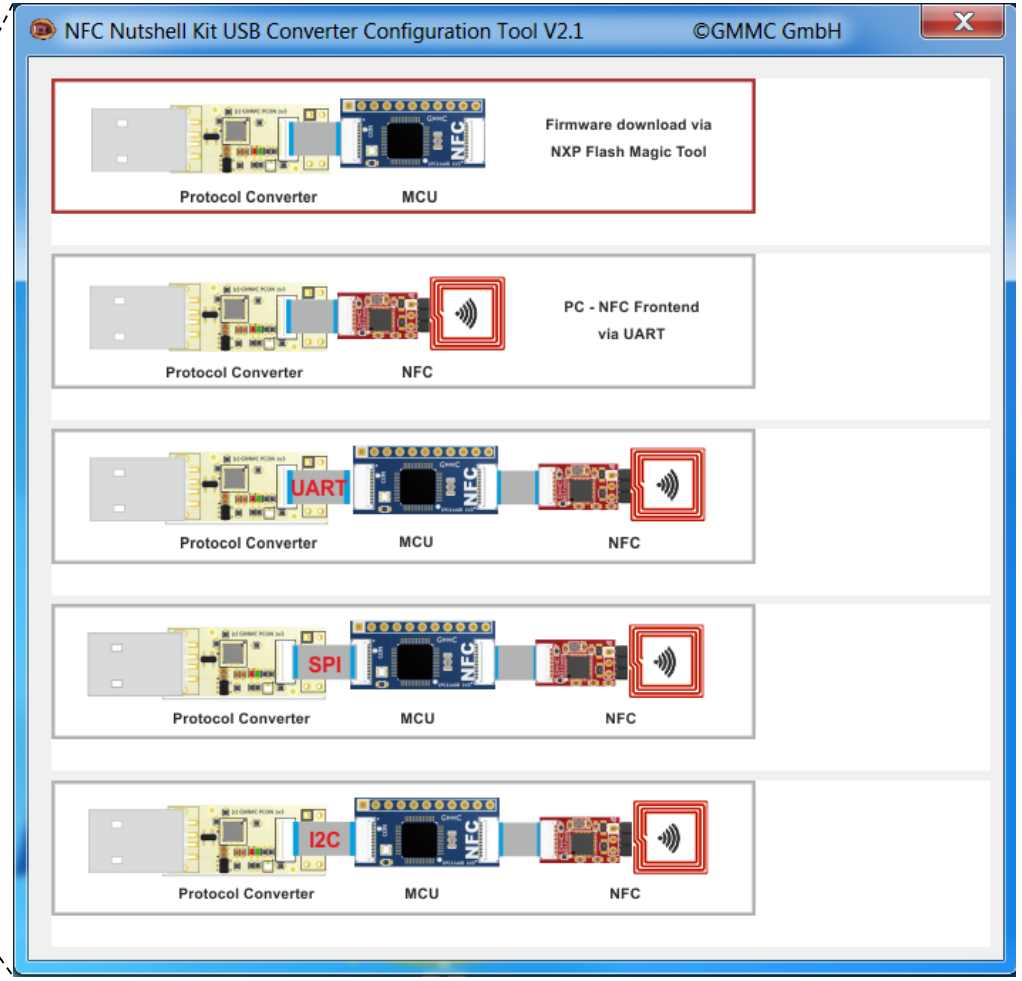


Training Mobile Knowledge

2. Configure USB converter

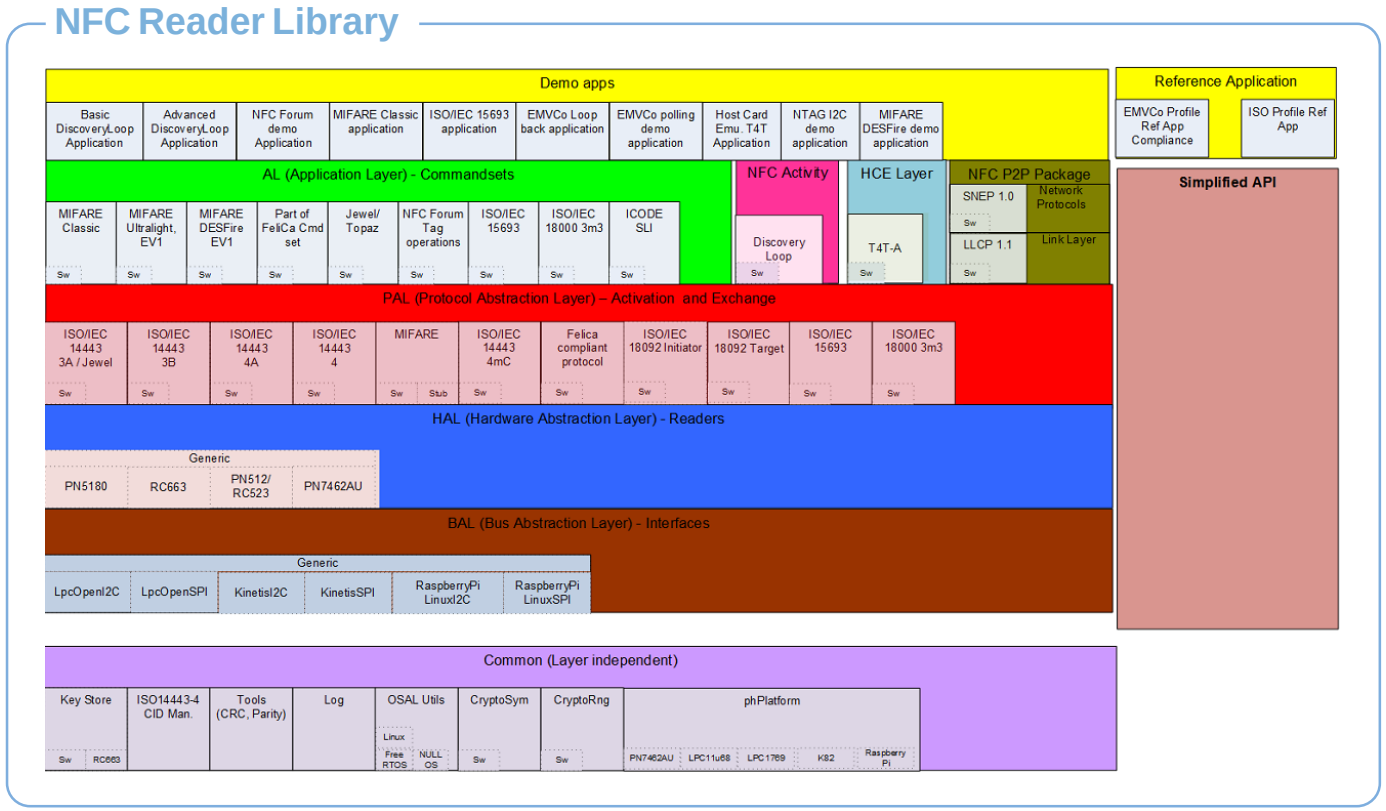
Host interface: USB protocol converter





Training Mobile Knowledge

3. Development: NFC Reader Library



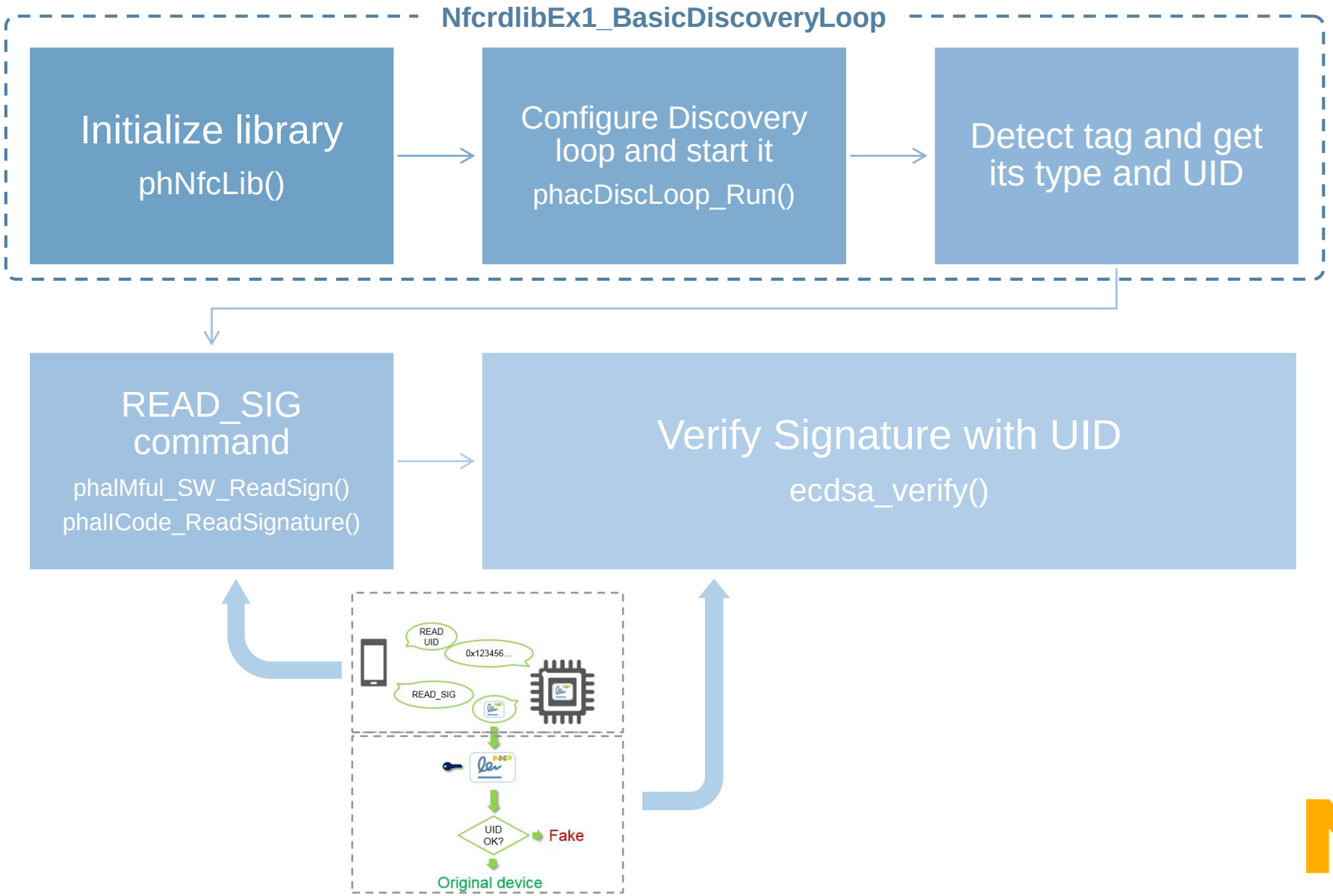
Software examples

Example 1: BasicDiscoveryLoop
Example 2: AdvancedDiscoveryLoop
Example 3: NFCForum
Example 4: MIFARE Classic®
Example 5: ISO15693
Example 6: EMVCo Loopback
Example 7: EMVCo Polling
Example 8: HCE T4T
Example 9: NTAG I2C
Example 10: SimplifiedAPI_EMVCo
Example 11: SimplifiedAPI_ISO

The NFC Reader Library is everything you need to create your own software stack and application for a contactless reader

Training Mobile Knowledge

3. Development: Originality signature verification



Training Mobile Knowledge

3. Development: Signature verification

- Reader library does not include asymmetric crypto
- Easy-ecc: a simple and secure ECDH and ECDSA library written in C
- Easy integration and use

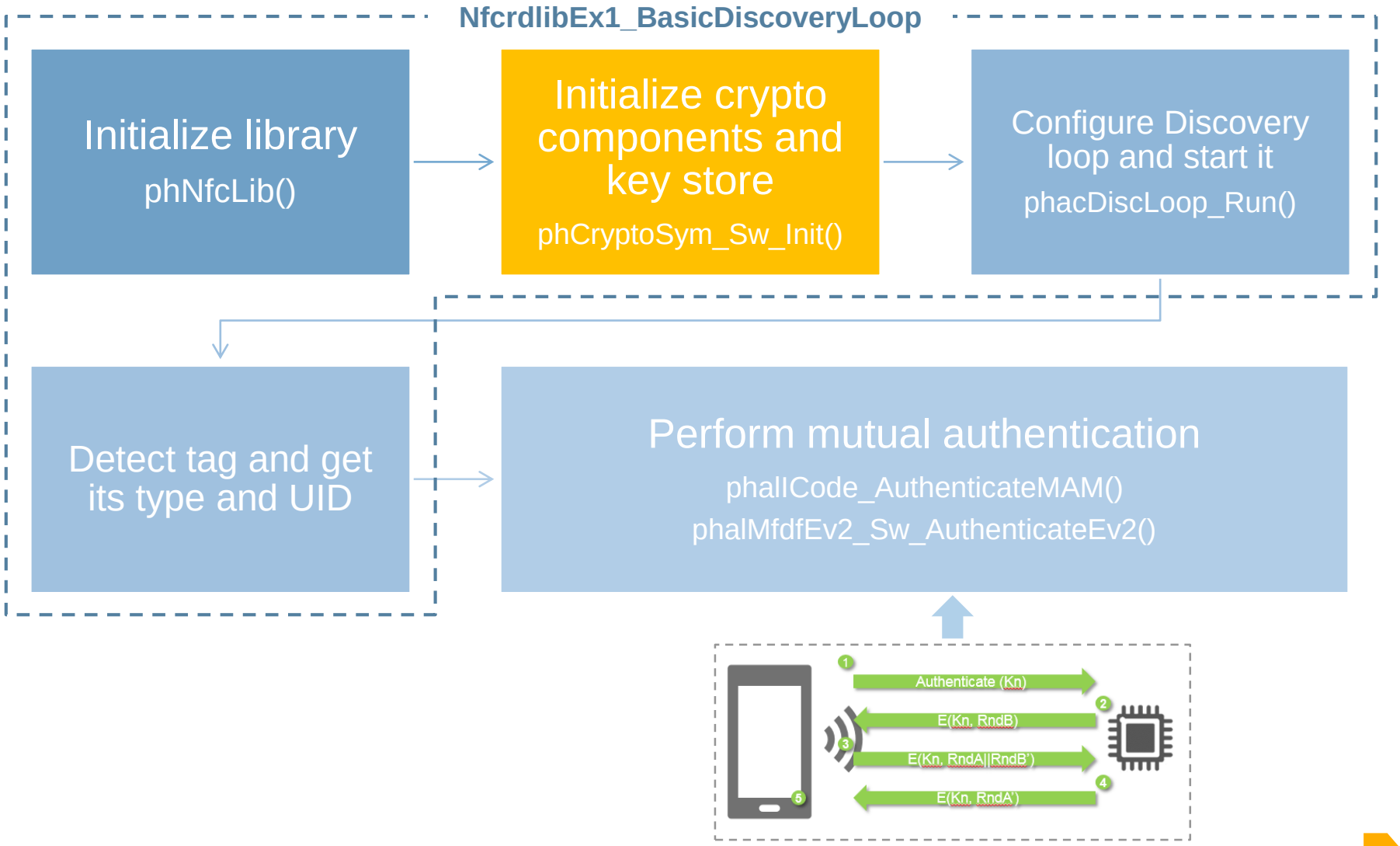
https://github.com/esxgx/easy-ecc

```
if (ecdsa_verify(PubKey, UID, Sig)) {
DEBUGOUT("Signature is a Valid NXP Signature.");
} else {
DEBUGOUT("Signature is Not valid.");
}
```

```
/* ecdsa_verify() function.
Verify an ECDSA signature.
Usage: Compute the hash of the signed data using the same
hash as the signer and
pass it to this function along with the signer's public key
and the signature values (r and s).
Inputs:
    p_publicKey - The signer's public key
    p_hash      - The hash of the signed data.
    p_signature - The signature value.
Returns 1 if the signature is valid, 0 if it is invalid.
*/
int ecdsa_verify(const uint8_t p_publicKey[ECC_BYTES+1],
    const uint8_t p_hash[ECC_BYTES], const uint8_t
    p_signature[ECC_BYTES*2]);
```

More information can be found in AN11350 NTAG Originality Signature Validation (1.2) document

3. Development: 3-pass mutual authentication

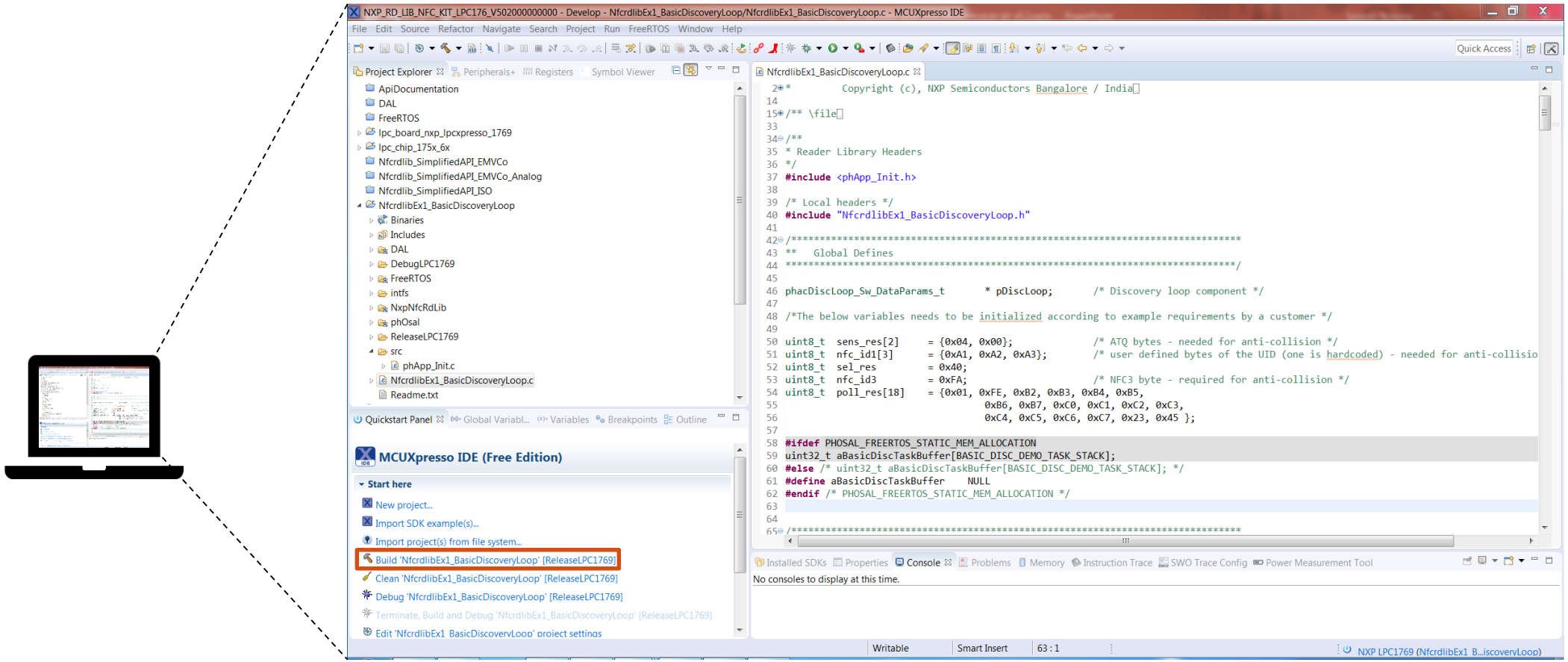


Training Mobile Knowledge

3. Development: 3-pass mutual authentication

Training Mobile Knowledge

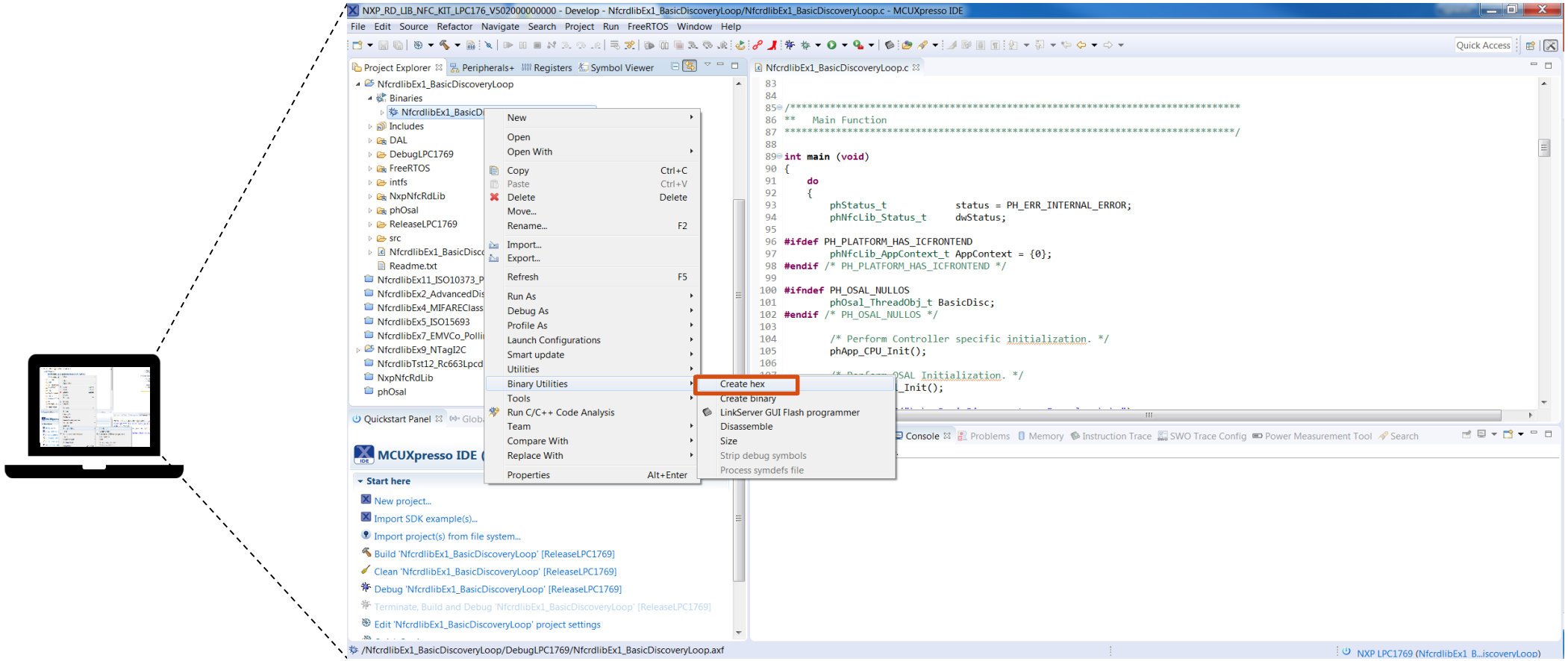
4. Build project in MCUXpresso




```
 2 *         Copyright (c), NXP Semiconductors Bangalore / India
14
15 /** \file
33
34 /**
35  * Reader Library Headers
36  */
37 #include <phApp_Init.h>
38
39 /* Local headers */
40 #include "NfcrdlibEx1_BasicDiscoveryLoop.h"
41
42 /*******************************************************************************
43 **   Global Defines
44 *******************************************************************************/
45
46 phacDiscLoop_Sw_DataParams_t       * pDiscLoop;       /* Discovery loop component */
47
48 /*The below variables needs to be initialized according to example requirements by a customer */
49
50 uint8_t  sens_res[2]     = {0x04, 0x00};             /* ATQ bytes - needed for anti-collision */
51 uint8_t  nfc_id1[3]      = {0xA1, 0xA2, 0xA3};       /* user defined bytes of the UID (one is hardcoded) - needed for anti-collisio
52 uint8_t  sel_res         = 0x40;
53 uint8_t  nfc_id3         = 0xFA;                     /* NFC3 byte - required for anti-collision */
54 uint8_t  poll_res[18]    = {0x01, 0xFE, 0xB2, 0xB3, 0xB4, 0xB5,
55                                0xB6, 0xB7, 0xC0, 0xC1, 0xC2, 0xC3,
56                                0xC4, 0xC5, 0xC6, 0xC7, 0x23, 0x45 };
57
58 #ifdef PHOSAL_FREERTOS_STATIC_MEM_ALLOCATION
59 uint32_t aBasicDiscTaskBuffer[BASIC_DISC_DEMO_TASK_STACK];
60 #else /* uint32_t aBasicDiscTaskBuffer[BASIC_DISC_DEMO_TASK_STACK]; */
61 #define aBasicDiscTaskBuffer    NULL
62 #endif /* PHOSAL_FREERTOS_STATIC_MEM_ALLOCATION */
63
64
65 /*******************************************************************************
```

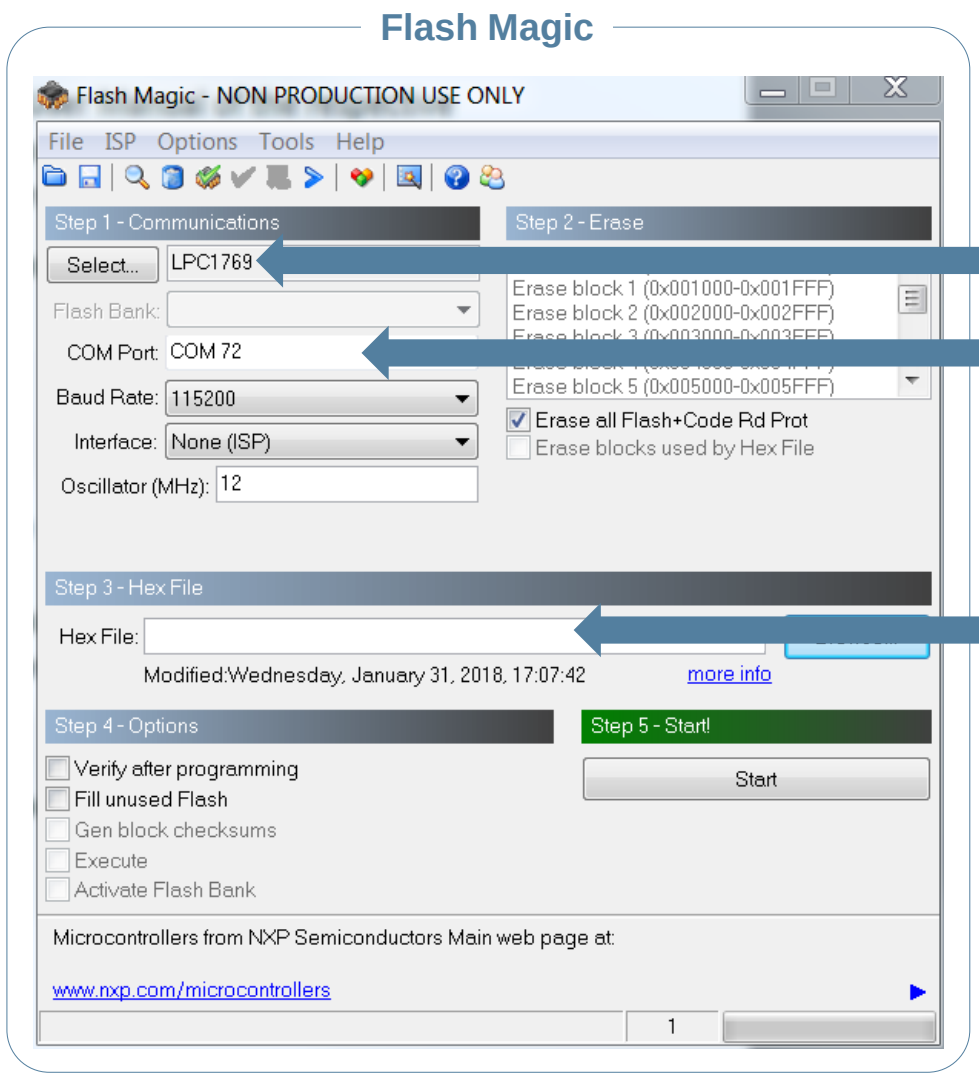

Training Mobile Knowledge

4. Build project in MCUXpresso: Create hex file



Training Mobile Knowledge

5. Flash the MCU image

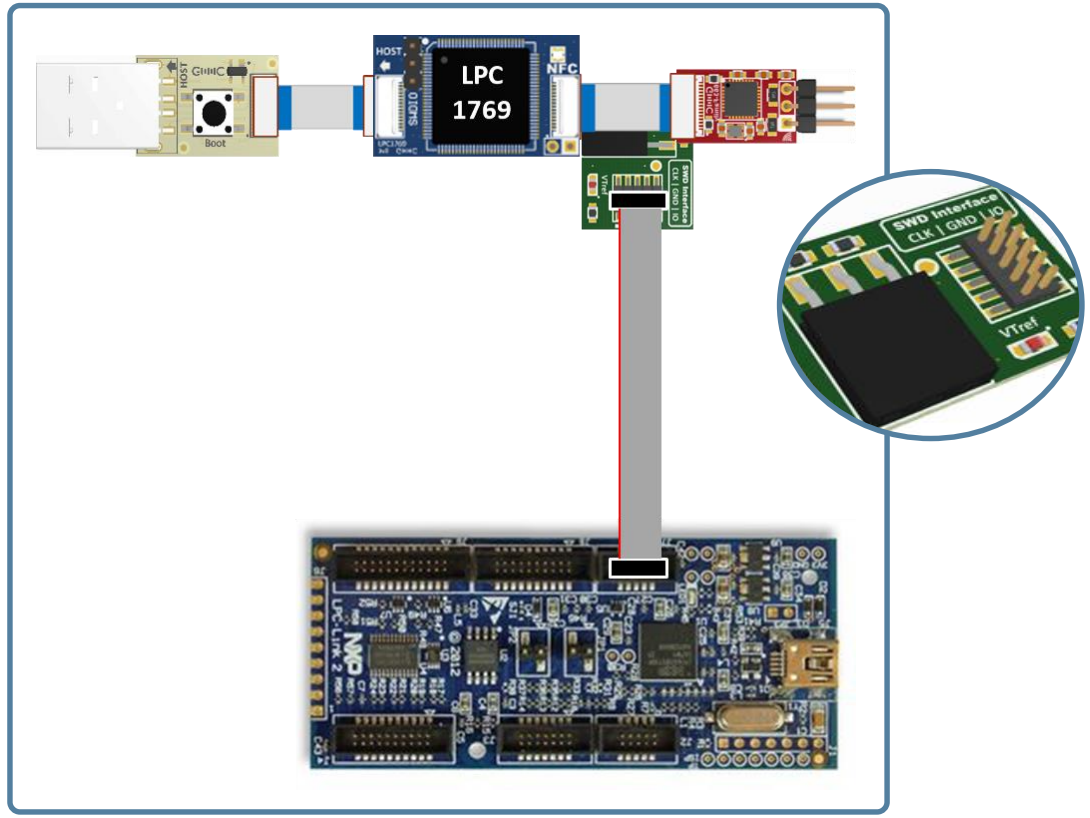


MCU

Hex image

Training Mobile Knowledge

Additional tool: Debugging module



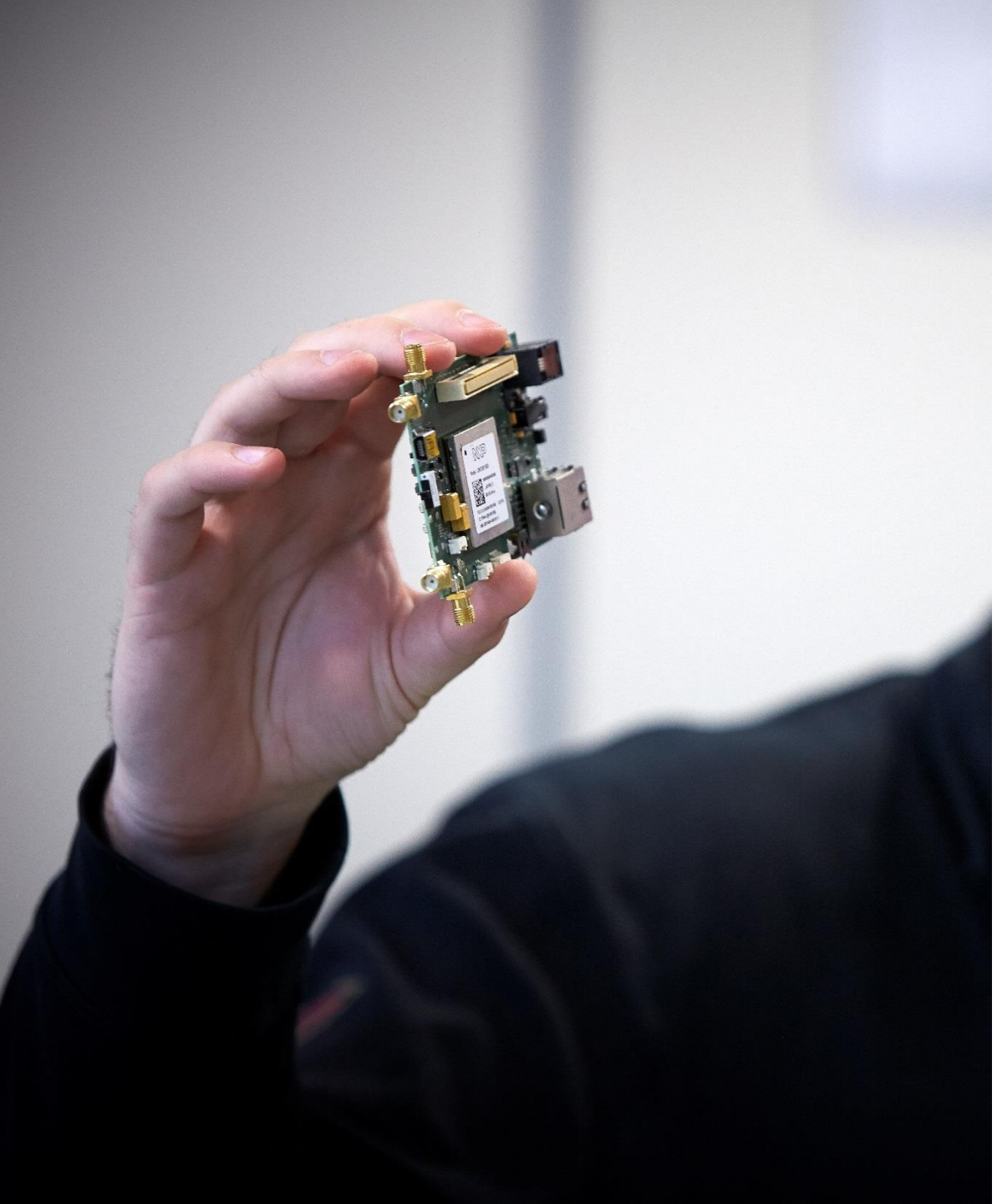
- Compatible with LPC1769 and LPC11u68, both included in the Nutshell Kit
- SWD-LPC LINK2 adapter, standard protocol for debugging
- Requires pre-programmed USB Bootloader and specific user application

Training Mobile Knowledge

EVERYTHING YOU NEED TO BUILD YOUR

NFC CONSUMABLE AUTHENTICATION

SOLUTION IS HERE!



Summary of available resources

Tags:

- NTAG 413 DNA
- NTAG 210µ
- NTAG 213 TT
- ICODE DNA

Readers:

- MFRC630 plus
- SLRC610 plus

AN11350 NTAG Originality Signature Validation
(Requires registration)

GMMC

NFC for consumables and accessories

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

Mobile Knowledge

Secure hardware design

Embedded software development

NFC antenna design and **evaluation**

EMV L1 pre-certification support

Mobile and **cloud application development**

Secure e2e system design

Advanced **technical training**

www.themobileknowledge.com

mk@themobileknowledge.com

MIFARE Accredited Training Partner

Approved Engineering Consultant NXP

MOBILE WORLD CAPITAL BARCELONA

We help companies leverage the **secure IoT revolution**

NXP