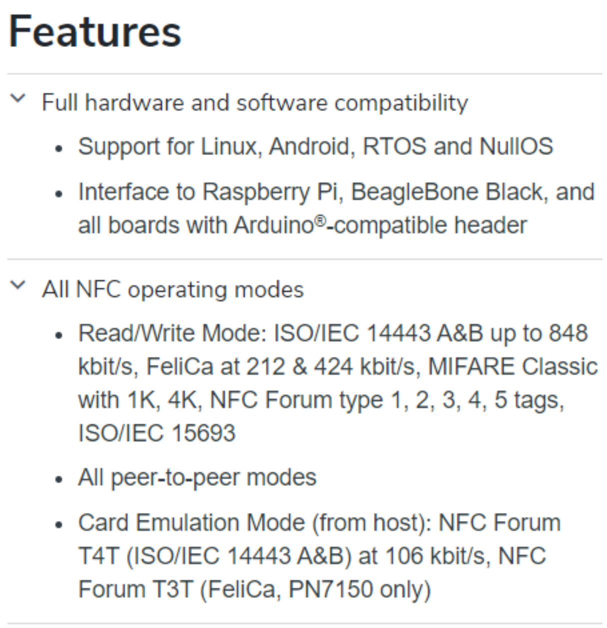
NFC ON KW

In the process of practical application, customers often need the combination of ble + NFC. At present, our IOT-DK006 is the only development board with NFC module. But the NFC example is not perfect. So we porting the library of NFC reader- PN7150, to support KW series microcomputer so that KW series can handle the demand of ble + NFC function.

Now I will introduce you how to port the NFC lib to KW.

1 PN7150 Introduction

PN7150 is the high-performance version of PN7120, the plug'n play NFC solution for easy integration into any OS environment, reducing Bill of Material (BOM) size and cost. PN71xx controllers are ideal for home-automation applications such as gateways and work seamlessly with NFC connected tags.



Features

Full hardware and software compatibility

- Support for Linux, Android, RTOS and NullOS
- Interface to Raspberry Pi, BeagleBone Black, and all boards with Arduino®-compatible header

All NFC operating modes

- Read/Write Mode: ISO/IEC 14443 A&B up to 848 kbit/s, FeliCa at 212 & 424 kbit/s, MIFARE Classic with 1K, 4K, NFC Forum type 1, 2, 3, 4, 5 tags, ISO/IEC 15693
- All peer-to-peer modes
- Card Emulation Mode (from host): NFC Forum T4T (ISO/IEC 14443 A&B) at 106 kbit/s, NFC Forum T3T (FeliCa, PN7150 only)

2 Tools

hardware： FRDM-KW36， PN7150 , some wire

software： mcuxpresso11.3

package： NXP-NCI MCUXpresso example Project

This package contains the nfc library and example that we need. We will refer the 'NXPNCI-K64F_example' firstly.

Sdk version: 2.2.8,

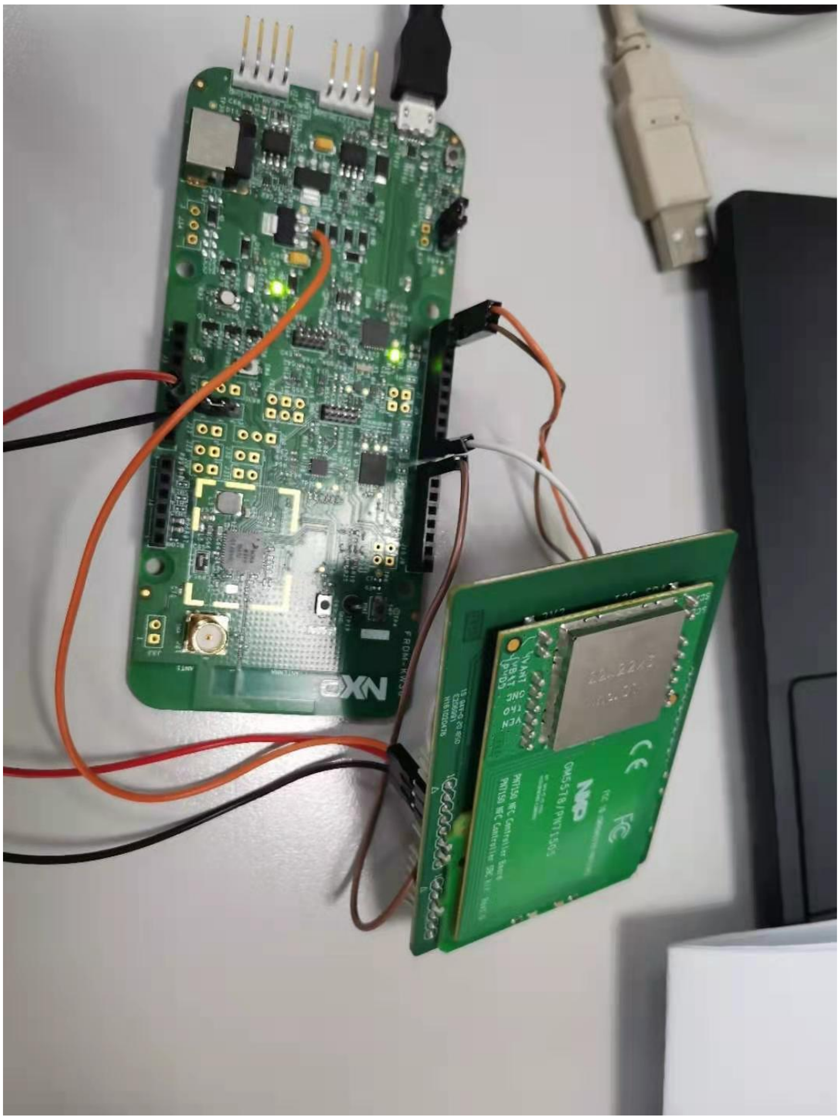
Example: frdmkw36_rtos_examples_freertos_i2c

3 Steps

Hardware part： We need connect the PN7150 to KW36 like the picture. Although we can connect the PN7150 to board through the ardunio connector, the pin's voltage is not enough to drive the PN7150. So we need a wire connected to U1 to get 3.3V.

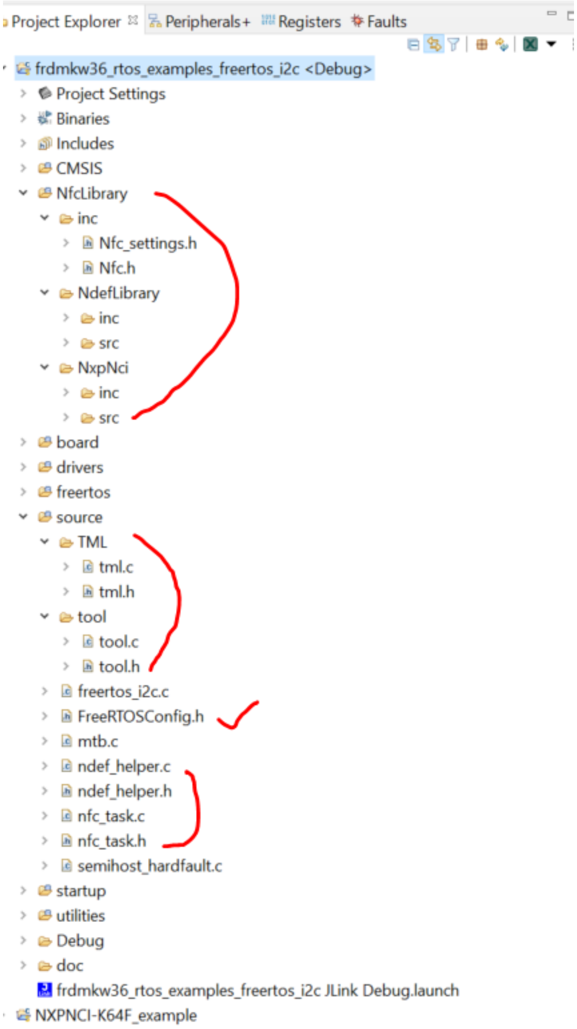
PN7150	FRDM-KW36
VBAT/PVDD	3.3V
VANT	5V
GND	GND
IRQ	PTA16
VEN	PTC15
SCL	PTB0，I2C0
SDA	PTB1，I2C0

连接如图:



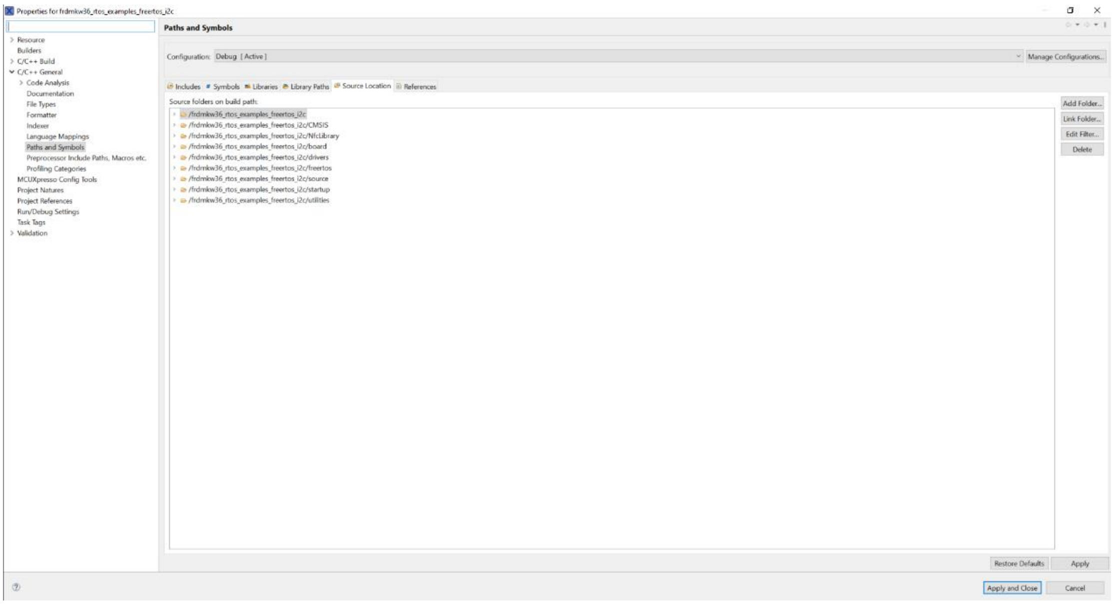
Software part：We should add the nfc library and directory into our project. You can check the following picture to know what file is necessary. If you want to know how to add directory into our project, you can refer this link.

The red line shows what file we need.



```
frdmkw36_rtos_examples_freertos_i2c <Debug>
  Project Settings
  Binaries
  Includes
  CMSIS
  NfcLibrary
    inc
      Nfc_settings.h
      Nfc.h
    NdefLibrary
      inc
      src
    NxpNci
      inc
      src
  board
  drivers
  freertos
  source
    TML
      tml.c
      tml.h
    tool
      tool.c
      tool.h
    freertos_i2c.c
    FreeRTOSConfig.h
    mtb.c
    ndef_helper.c
    ndef_helper.h
    nfc_task.c
    nfc_task.h
    semihost_hardfault.c
  startup
  utilities
  Debug
  doc
  frdmkw36_rtos_examples_freertos_i2c JLink Debug.launch
NXPNCI-K64F_example
```

Please notice that when we add file path into the mcuxpresso configuration, we also need add the path into ‘Path and Symbols’ .

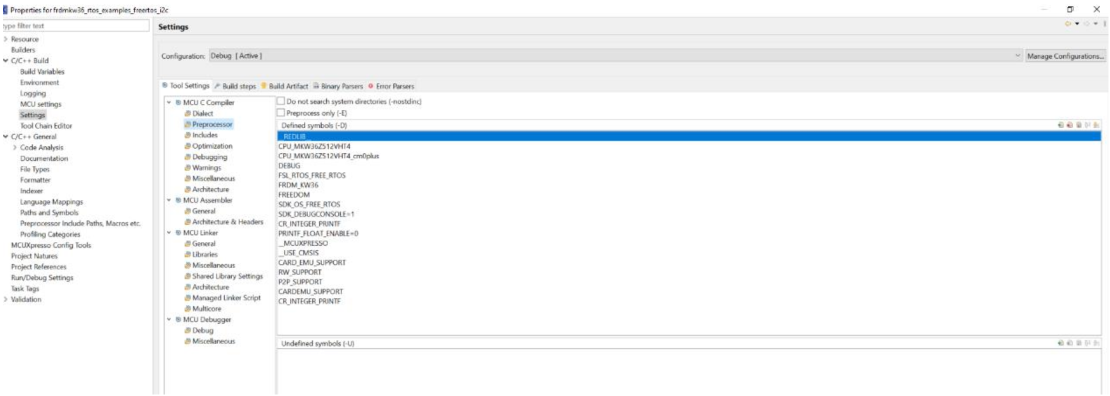


Configuration: Debug [Active]

Source folders on build path:

```
/frdmkw36_rtos_examples_freertos_i2c
/frdmkw36_rtos_examples_freertos_i2c/CMSIS
/frdmkw36_rtos_examples_freertos_i2c/NfcLibrary
/frdmkw36_rtos_examples_freertos_i2c/board
/frdmkw36_rtos_examples_freertos_i2c/drivers
/frdmkw36_rtos_examples_freertos_i2c/freertos
/frdmkw36_rtos_examples_freertos_i2c/source
/frdmkw36_rtos_examples_freertos_i2c/startup
/frdmkw36_rtos_examples_freertos_i2c/utilities
```

We need add some macro into ‘Preprocessor’.



Properties for frdmkw36_rtos_examples_freertos_i2c

Settings

Configuration: Debug [Active]

□ Do not search system directories (-nostdinc)
□ Preprocess only (-E)

Defined symbols (-D)

```
__REDLIB__
CPU_MKW36Z512VHT4
CPU_MKW36Z512VHT4_cm0plus
DEBUG
FSL_RTOS_FREE_RTOS
FRDM_KW36
FREEDOM
SDK_OS_FREE_RTOS
SDK_DEBUGCONSOLE=1
CR_INTEGER_PRINTF
PRINTF_FLOAT_ENABLE=0
__MCUXPRESSO
__USE_CMSIS
CARD_EMU_SUPPORT
RW_SUPPORT
P2P_SUPPORT
CARDEMU_SUPPORT
CR_INTEGER_PRINTF
```

Undefined symbols (-U)

We copy the NXPNCI-K64F_example's main file content into our 'freertos_i2c.c'. Next, we need modify the file pin_mux.c, tml.c and board.h

In file board.h, add the following macro.

```
/* NXPNCI NFC related declaration */
#define BOARD_NXPNCI_I2C_INSTANCE  I2C0
#define BOARD_NXPNCI_I2C_BAUDRATE  (100000)
#define BOARD_NXPNCI_I2C_ADDR       (0x28)
#define BOARD_NXPNCI_IRQ_PORTIRQn  PORTA_IRQn
#define BOARD_NXPNCI_IRQ_GPIO      (GPIOA)
#define BOARD_NXPNCI_IRQ_PORT      (PORTA)
#define BOARD_NXPNCI_IRQ_PIN       (16U)
#define BOARD_NXPNCI_VEN_GPIO      (GPIOC)
#define BOARD_NXPNCI_VEN_PORT      (PORTC)
#define NXPNCI_VEN_PIN             (5U)
```

In file pin_mux.c, add head file 'board.h'. Add the following code in function ' BOARD_InitPins'. The step is to configure the VEN, IRQ and I2C0. This example contains the I2C1's code, you can comment them.

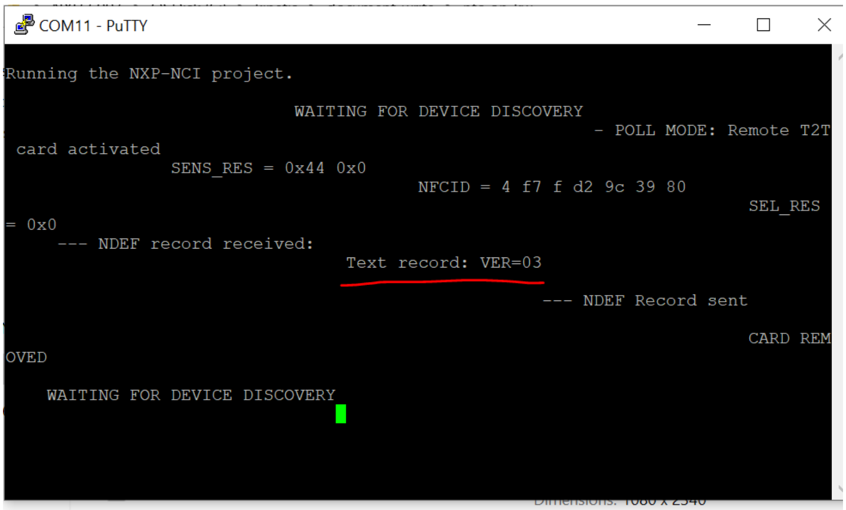
```
  /* Initialize NXPNCI GPIO pins below */
  /* IRQ and VEN PIN_MUX Configuration */
  PORT_SetPinMux(BOARD_NXPNCI_IRQ_PORT, BOARD_NXPNCI_IRQ_PIN,
kPORT_MuxAsGpio);
  PORT_SetPinMux(BOARD_NXPNCI_VEN_PORT, NXPNCI_VEN_PIN, kPORT_MuxAsGpio);
  /* IRQ interrupt Configuration */
  NVIC_SetPriority(BOARD_NXPNCI_IRQ_PORTIRQn, 6);
  EnableIRQ(BOARD_NXPNCI_IRQ_PORTIRQn);
      PORT_SetPinInterruptConfig(BOARD_NXPNCI_IRQ_PORT, BOARD_NXPNCI_IRQ_PIN,
    kPORT_InterruptRisingEdge);
```

Finally, in file tml.c, modify `PORTC_IRQHandler` as `PORTA_IRQHandler`
We finished all steps.

4 Results

We use ntag to test the reading and writing operation.

When the tag is closed to the PN7150, we will get the following message.

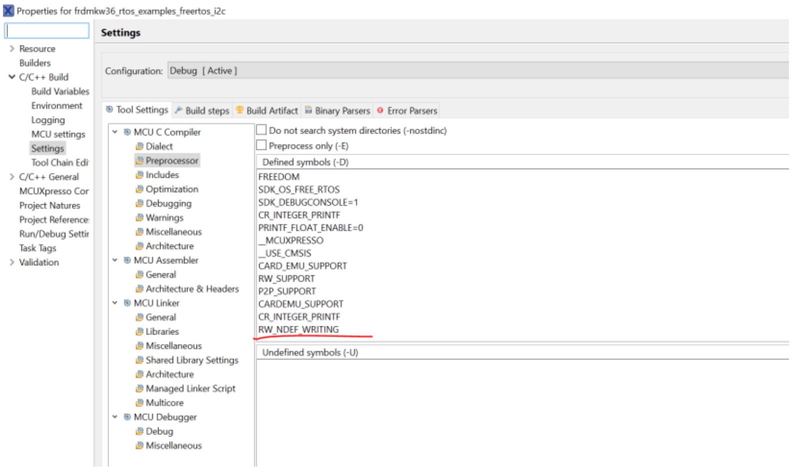


```
Running the NXP-NCI project.
                            WAITING FOR DEVICE DISCOVERY
                                                             - POLL MODE: Remote T2T
 card activated
               SENS_RES = 0x44 0x0
                                       NFCID = 4 f7 f d2 9c 39 80
                                                                SEL_RES
= 0x0
     --- NDEF record received:
                              Text record: VER=03
                                                 --- NDEF Record sent

                                                                CARD REM
OVED

    WAITING FOR DEVICE DISCOVERY
```

The text recording is ‘VER=03’.

Next, we will modify the text recording

We need add the new macro to preprocessor.



Properties for frdmkw36_rtos_examples_freertos_i2c

Settings

Configuration: Debug [Active]

Tool Settings | Build steps | Build Artifact | Binary Parsers | Error Parsers

MCU C Compiler > Preprocessor

□ Do not search system directories (-nostdinc)

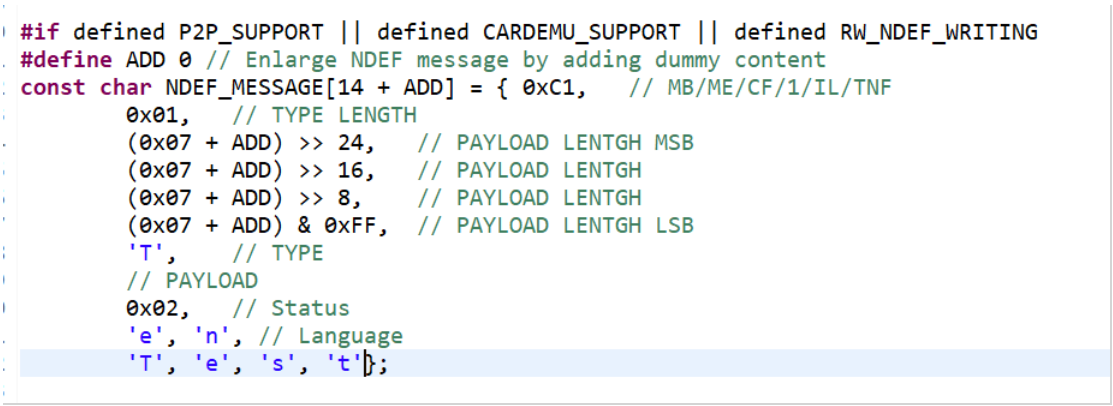
□ Preprocess only (-E)

Defined symbols (-D)

```
FREEDOM
SDK_OS_FREE_RTOS
SDK_DEBUGCONSOLE=1
CR_INTEGER_PRINTF
PRINTF_FLOAT_ENABLE=0
__MCUXPRESSO
__USE_CMSIS
CARD_EMU_SUPPORT
RW_SUPPORT
P2P_SUPPORT
CARDEMU_SUPPORT
CR_INTEGER_PRINTF
RW_NDEF_WRITING
```

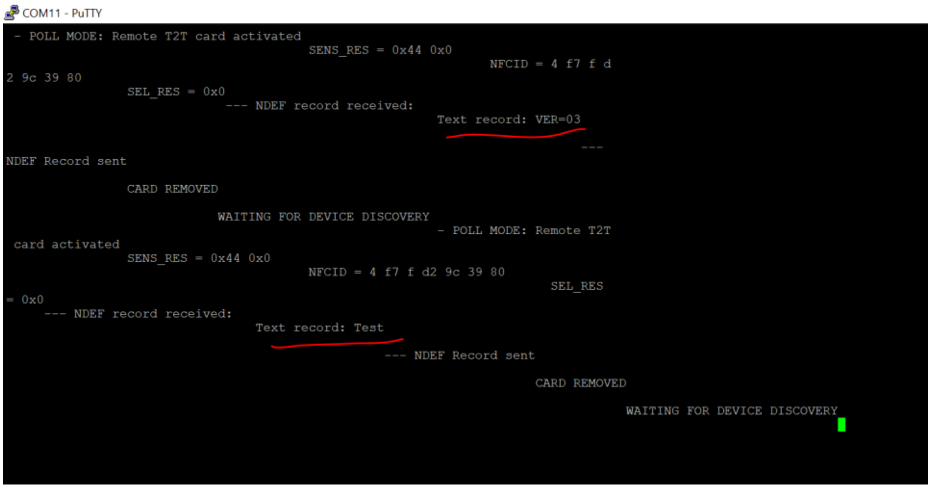
Undefined symbols (-U)

We can modify the variable `NDEF_MESSAGE` in function **`task_nfc_reader`** to modify the text recording.



```
#if defined P2P_SUPPORT || defined CARDEMU_SUPPORT || defined RW_NDEF_WRITING
#define ADD 0 // Enlarge NDEF message by adding dummy content
const char NDEF_MESSAGE[14 + ADD] = { 0xC1,    // MB/ME/CF/1/IL/TNF
        0x01,   // TYPE LENGTH
        (0x07 + ADD) >> 24,   // PAYLOAD LENTGH MSB
        (0x07 + ADD) >> 16,   // PAYLOAD LENTGH
        (0x07 + ADD) >> 8,    // PAYLOAD LENTGH
        (0x07 + ADD) & 0xFF,  // PAYLOAD LENTGH LSB
        'T',    // TYPE
        // PAYLOAD
        0x02,   // Status
        'e', 'n', // Language
        'T', 'e', 's', 't'};
```

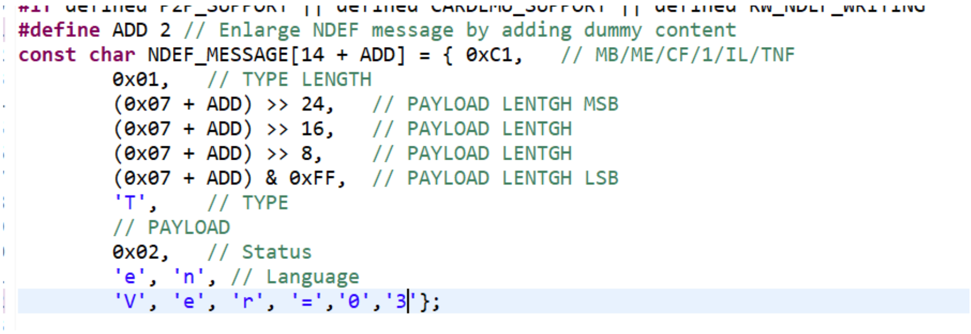
Then we download the program again. We will see the original text ‘VER=03’ and the text has been modified. Then we read the tag again. We will see the new text.



COM11 - PuTTY

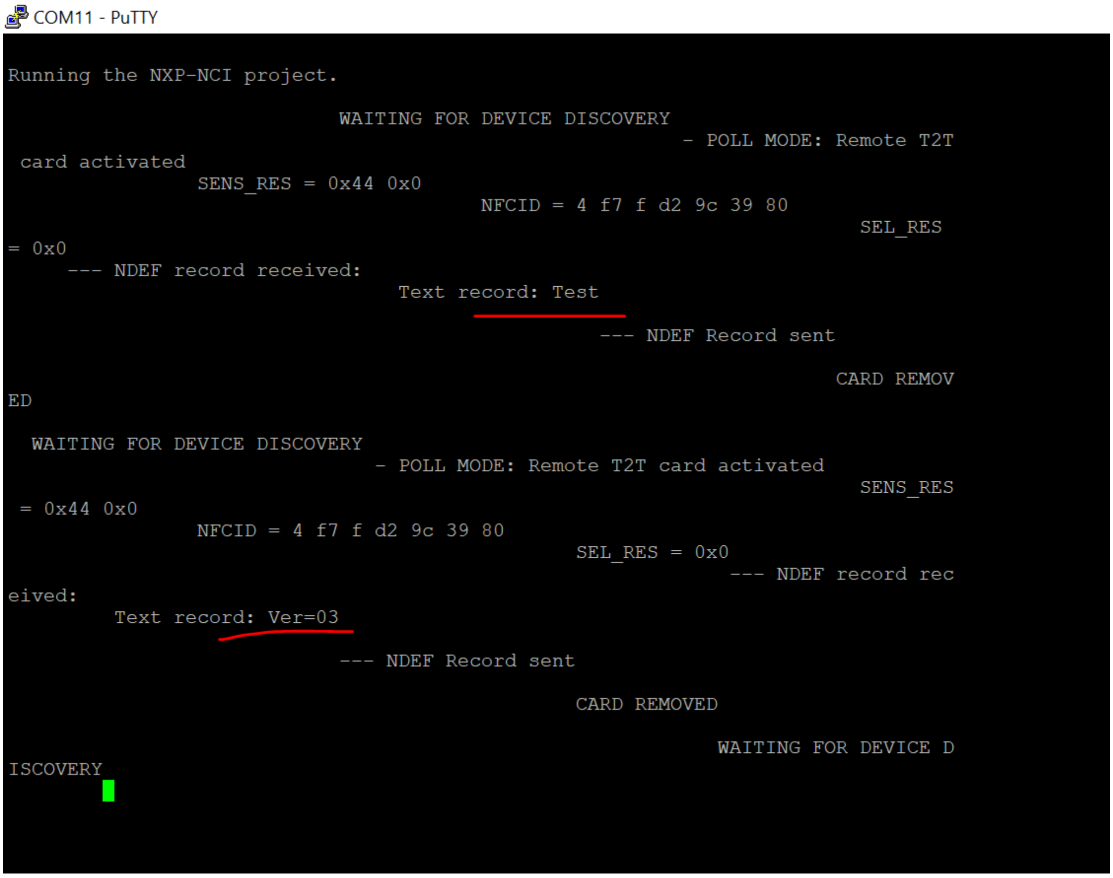
```
 - POLL MODE: Remote T2T card activated
                                        SENS_RES = 0x44 0x0
                                                               NFCID = 4 f7 f d
2 9c 39 80
                SEL_RES = 0x0
                             --- NDEF record received:
                                                        Text record: VER=03

                                                                           ---
NDEF Record sent

                CARD REMOVED

                            WAITING FOR DEVICE DISCOVERY
                                                        - POLL MODE: Remote T2T
 card activated
                SENS_RES = 0x44 0x0
                                        NFCID = 4 f7 f d2 9c 39 80
                                                                       SEL_RES
= 0x0
     --- NDEF record received:
                                 Text record: Test

                                                  --- NDEF Record sent

                                                                     CARD REMOVED

                                                                                 WAITING FOR DEVICE DISCOVERY
```

If we want to send the larger text, what should we do? We need modify the macro ‘ADD’. When only 4 characters are sent, ‘ADD’ is 0. And every additional character is added, the ‘ADD’ will add. We modify the tag as ‘Ver=03’, and we have two more characters. So ‘ADD’ needs to be defined as 2



```
#if defined P2P_SUPPORT || defined CARDEMU_SUPPORT || defined RW_NDEF_WRITING
#define ADD 2 // Enlarge NDEF message by adding dummy content
const char NDEF_MESSAGE[14 + ADD] = { 0xC1,   // MB/ME/CF/1/IL/TNF
        0x01,   // TYPE LENGTH
        (0x07 + ADD) >> 24,   // PAYLOAD LENTGH MSB
        (0x07 + ADD) >> 16,   // PAYLOAD LENTGH
        (0x07 + ADD) >> 8,    // PAYLOAD LENTGH
        (0x07 + ADD) & 0xFF,  // PAYLOAD LENTGH LSB
        'T',    // TYPE
        // PAYLOAD
        0x02,   // Status
        'e', 'n', // Language
        'V', 'e', 'r', '=','0','3'};
```

It firstly shows the text ‘Test’. Then it will show the new text ‘Ver=03’.



```
Running the NXP-NCI project.
                            WAITING FOR DEVICE DISCOVERY
                                                         - POLL MODE: Remote T2T
 card activated
                SENS_RES = 0x44 0x0
                                        NFCID = 4 f7 f d2 9c 39 80
                                                                        SEL_RES
= 0x0
     --- NDEF record received:
                                 Text record: Test

                                                  --- NDEF Record sent

                                                                      CARD REMOV
ED

  WAITING FOR DEVICE DISCOVERY
                               - POLL MODE: Remote T2T card activated
                                                                        SENS_RES
 = 0x44 0x0
                NFCID = 4 f7 f d2 9c 39 80
                                                SEL_RES = 0x0
                                                             --- NDEF record rec
eived:
         Text record: Ver=03

                            --- NDEF Record sent

                                                CARD REMOVED

                                                            WAITING FOR DEVICE D
ISCOVERY
```

Other tags’ reading and writing operation can be enabled by defining some macro.