Start-up Guide for FRDM-KW41Z Evaluation Board Bluetooth Paring example with NTAG I²C *plus*

Rev. 2.0 — 5 June 2018 422220

User manual COMPANY PUBLIC

Document information

Info	Content
Keywords	NTAG I ² C <i>plus</i> , FRDM-KW41Z
Abstract	This document gives a start-up guide for Bluetooth BLE pairing demonstration between FRDM-KW41 and NFC mobile device with use of NTAG I ² C <i>plus</i> and also explains how the pairing part is included in the connectivity stack.



Revision history

Rev	Date	Description
2.0	20180605	The BT pairing was updated with a different demo application and migrated from Kinetis Design Studio to MCUXpresso IDE and SDK.
1.2	20180206	Change of the information on how to use the demo in <u>chapter 2</u>
1.1	20170307	Text: LPCXpresso IDE changed to Kinetis Design Studio IDE
1.0	20170307	Initial version

Contact information

For more information, please visit: http://www.nxp.com

UM10950

All information provided in this document is subject to legal disclaimers.

User manual COMPANY PUBLIC Downloaded from Arrow.com.

1. Introduction and HW setup

This document can be divided into three basic parts. This the first one introduces and describes the HW setting. The second part is a quick startup guide and describes how to quickly and easily work with the prepared sample of the Bluetooth demo application. The third part of the document describes how to import NTAG I2C middleware with a NDEF library into the selected Bluetooth application.

The whole integration process of the NTAG I²C middleware and the NDEF library can be applied to any customer application.

1.1 HW setup

For the NTAG I²C *plus* development kit, there are 2 kits that can be attached to the FRDM-KW41Z board. The first OM5569 [2] is the original and it represents the typical HW design, the second OM23221ARD [3] is adapted to Arduino pinout. Both kits are fully fledged and its on the user which is chosen.

1.1.1 NTAG I²C *plus* board

The HW connection and wiring between the NTAG I²C *plus* board and FRDM-KW41Z board is shown at the following picture. RGB LED and micro switches SW3, SW4 which are used in HID_device demo application (more information is in the chapter 2.2) and which signal usage of the NTAG I²C chip are mounted directly on the FRDM-KW41Z. Setting functions and control functions for this LED is a part of the SDK. From this reason there is not required extra HW setup for these elements (see reference[2]).



UM10950

User manual COMPANY PUBLIC

Example	Description	GPIO	GPIO direction	specification
I2C – SCL	Interface clock	PTC 2	I ² C specific	open drain SCL
I2C – SDA	Interface data	PTC 3	I ² C specific	open drain SCL
GND	Ground	GND	output	-
VCC_SW	NTAG I ² C Antenna board power supply	P3V3	output	-
LED – RED	FRDM-KW41Z - RGB LED driver	PTC 1	output	Default SDK configuration
LED – GREEN	FRDM-KW41Z RGB LED driver	PTA 19	output	Default SDK configuration
LED – BLUE	FRDM-KW41Z RGB LED driver	PTA 18	output	Default SDK configuration
SW3		PTC 4	input	Default SDK configuration
SW4		PTC 5	input	Default SDK configuration

1.1.2 Arduino NTAG I²C *plus* board

OM23221ARD development kit with NTAG I²C *plus* is an easy add-on to many popular MCU boards. It gives connectivity to any device with Arduino pinout. The HW setting is the identical with previous description in the chapter 1.1.1 (see reference [3])



Downloaded from Arrow.com.

User manual

2. Quick start-up the demo application

This chapter provides a procedure how to easily install the development environment with the necessary components. Next, how to compile and debug a demo application and finally how to verify it's functionality.

2.1 Installation of the MCUXpresso IDE and SDK

The whole project of the demo application is written for the MCUXpresso IDE, so it is necessary to install this development environment. The next step is to generate and import the SDK (Software Development Kit) needed for the application to be compiled and debugged. The procedure to achieve this is described in the following chapters.

2.1.1 How to download the installation file of the MCUXpresso IDE

The following steps show how to easily download the installation file for the MCUXpresso IDE.

- 1.) Open the welcome page of the MCUXpresso site [4]: https://mcuxpresso.nxp.com/en/welcome
- 2.) Select the "SOFTWARE AND TOOLS" and press the "Learn More >"



UM10950

Downloaded from Arrow.com.

User manual COMPANY PUBLIC

- 3.) In the next step you will be redirected to the NXP web page
- 4.) Choose the "DOWNLOADS" tab and press the "Download"

	NP					Ŧ	• O ENGLISH ~ ¥ CA	ART
	PRODUCTS		SUPPORT		ALL	- v Search	t Environment //DE\	Q
	MCUX	presso In	tegrate	d Develor	oment Envi	ronment (IE	DE) 🛛	<
	OVERVIEW	W DOCUME	NTATION	DOWNLOADS	DEVELOPMENT TOOLS	BUY/PARAMETRIC	S TRAINING & SUPPORT	
	Product D	Downloads	♥	MCUXpresso In Environment (IE Latest Version Previous Versio Updates & Pate Existing and Ne	tegrated Developn)E) on ches ew licenses	nent	Download 🗹	
	ABOUT NXP Investors Press, News, Careers	Blogs	RESOURCE Mobile App Contact Us	ES IS	FOLLOW US	New NXP1 1.0 Rea	rs 4 May 2018 < MX 8M Supports Android Thing d More	 ac
4.	IDE – sta	arting the	downlo	ad				

- 5.) Within the next step is necessary to sign in.
- 6.) Select the version of the installation file for MCUXpresso. It is recommended to select the latest version (especially at the start of development) because it contains the latest updates.

				ALL -			Q
PRODUCTS	APPLICATIONS	SUPPORT	ABOUT				
NXP > Software & Support > I	Product Information : MC	UXpresso IDE					
Software & Support	Product	Informati	ion				
Product List	TTOULOU	morman					
Product Search Order History	MCUXpresso	IDE					
Recent Product Releases							
Recent Updates	To register a New	Product please click o	n the button below				
icensing	Register						
License Lists	\square						
Offline Activation	Current Prev	ious					
AQ	Version Desc	ription					
Download Help	10.1.1 MCU	Xpresso IDE				Download Log	
Table of Contents							
FAQs							

7.) Select the correct type of the operating system that is used by user and download the installation file.

<complex-block> Accord Pedded Products APPLICATION Canced Particle Canced Canced Products APPLICATION Canced Products APPLICATION Canced Products APPLICATION Canced Canced Products APPLICATION Canced <p< th=""><th></th><th></th><th></th><th></th></p<></complex-block>				
Note: Note: Note: Product Search Product/genesotic: Product/genesotic: Product Search Product Search Recent Updates Product Search Product Search Product Search <td< th=""><th>NP</th><th></th><th></th><th>ount ⊕ English ▼</th></td<>	NP			ount ⊕ English ▼
Rectified of Contents Robinson Rectified of Contents Robinson Rectified of Contents Robinson Rectified of Contents Robinson Robinson <th></th> <th></th> <th>ALL</th> <th>~</th>			ALL	~
<page-header> Starte & Support Public Lead Pack Ling Public Lead Public Lead Public Lead <!--</th--><th>PRODUCTS</th><th>APPLICATIONS SUPPORT ABOU</th><th>JT</th><th></th></page-header>	PRODUCTS	APPLICATIONS SUPPORT ABOU	JT	
Software & Support Product User Product Startin Product Startin Rest Usdars Product Startin Product Startin Product Startin	NXP > Software & Support > M	ICUXpresso IDE > MCUXpresso IDE : Files		
Product List Product Search Order Helov Recent Product Releases Recent Product Releases Recent Product Releases Recent Updates License Lists Offine Activation + File Description + MCUXpressolDE_1011_606 Link + MCUXpressolDE_1011_606 Link + MCUXpressolDE_1011_606 Link + MCUXpressolDE_1011_606 Link + MCUXpressolDE_1011_606 Main - MOUXpressolDE_1011_606 Main <tr< td=""><td>Software & Support</td><td>Product Download</td><td></td><td></td></tr<>	Software & Support	Product Download		
MCUXpresso IDE Order History MCUXpresso IDE Recent Product Releases Files © Download Help Leense Lists The Decorption Image: Control of the Name Image: Control of the Name Image: Control of the Name FAQ MCUXpressoIDE_10.11_808 Linux Tools Mail Image: Control of the Name Image: Control of the Name Image: Control of the Name FAQ MCUXpressoIDE_10.11_808 Linux Tools Mail Image: Control of the Name	Product List	Freddot Bonnodd		
Racent Pool Racent Pool </td <td>Product Search Order History</td> <td>MCUXpresso IDE</td> <td></td> <td></td>	Product Search Order History	MCUXpresso IDE		
Recent Updates Image: Construction of the location of the	Recent Product Releases	Files License Keys Notes		O Download Help
Licensing 4 Files Licensing 	Recent Updates			25
License Lists Offline Activation FAQ Download Help Table of Contents FAO Select MCUXpressolDE_10.11_605 Linux Table of Contents FAO Select MCUXpressolDE_10.11_605 Mac Get 8 MB & MCUXpressolDE_10.11_605 Mac FAOS Select MCUXpressolDE_10.11_605 Mac FAOS Select FOLLOWUS NEWS	Licensing	Show All Files		4 Files
Offline Activation + LPC684patch_MCUXpressolDE_v10p1p1 378.84 B_LPC684patch_MCUXpressolDE_v10p1p1 zp FAQ + MCUXpressolDE_v10.11_606.kdg + MCUXpressolDE_v10.11_606.kdg Download Help + MCUXpressolDE_v10.11_606.kdg + MCUXpressolDE_v10.11_606.kdg Table of Contents + MCUXpressolDE_v10.11_606.kdg + MCUXpressolDE_v10.11_606.kdg FAQ + MCUXpressolDE_v10.11_606.kdg + MCUXpressolDE_v10.11_606.kdg Table of Contents + MCUXpressolDE_v10.11_606.kdg + MCUXpressolDE_v10.11_606.kdg FAOs Select Download + MCUXpressolDE_v10.11_606.kdg Proversion Selected Files Mobile Apps Mobile Apps Partners Press, News, Blogs MCU More NEWS NAP News Careers Contact Us Mobile Apps News, Blogs NAP News Read More Privacy Terms of Use Terms of Sale Feedback 62006-2017 NXP Semiconductors. All rights reserved	License Lists	File Description	File Size File Name	\$
FAQ + MCUXpressolDE_10.11_606.kbg + MCUXpressolDE_10.11_606.kbg + MCUXpressolDE_10.11_606.kbg + MCUXpressolDE_10.11_606.kbg + MCUXpressolDE_10.11_606.ccc Table of Contents + MCUXpressolDE_10.11_606.ccc Select Download Help - MCUXpressolDE_10.11_606.ccc ABOUT NXP RESOURCES FOLLOW US NEWS > NKP News Partners Careers Mobile Apps Press, News, Blogs Careers NEWS > Careers Press, News, Blogs Careers Press, News, Blogs Careers NEWS > Careers Press, News, Blogs Careers Press, News, Blogs Careers Press, News, Blogs Careers Press, News, Blogs Careers Careers Press, News, Blogs Careers Press, News, Blogs Careers Press, Partners Press	Offline Activation	+ LPC804patch_MCUXpressoIDE_v10p1p1	379.8 KB 🞍 LPC804patch_MCUXpressolDE	_v10p1p1.zip
Download Help + MCUXpressolDE_10.11_606 Mac e94.8 MB MCUXpressolDE_10.11_606 pkg + MCUXpressolDE_10.11_606 cws Table of Contents + MCUXpressolDE_10.11_606 Windows 638.1 MB MCUXpressolDE_10.11_606 cws FAGs Select Townload Stelected Files Windows Molie Apps Policownload Pathers Press, News, Blogs Image: Molie Apps Press, News, Blogs Contact Us Image: Molie Apps Press, Verses, Blogs Contact Us Molie Apps Press, Verses, Blogs Contact Us Image: Molie Apps Press, Verses, Blogs Contact Us Image: Molie Apps Press, News Press, News Read More	FAQ	HCUXpressoIDE_10.1.1_606 Linux	700.8 MB 🛓 mcuxpressoide-10.1.1_606.x86_	64.deb.bin
Table of Contents + MCUXpressolDE_10.11_606 exe Select Download eds.1 MB MCUXpressolDE_10.11_606 exe FAGs Select Download Mobile Apps Policy Mobile Apps Partners Careers Privacy Terms of Use Terms of Sale Feedback	Download Help	MCUXpressoIDE_10.1.1_606 Mac	694.8 MB & MCUXpressoIDE_10.1.1_606.pt	9
FAGS Select Download Select Download ABOUT NXP RESOURCES Mobile Apps NEWS > Partners Press, News, Blogs Contact Us Im Im Im NXP News Read More Privacy Terms of Use Terms of Sale Feedback Eedback @2006-2017 NXP Semiconductors. All rights reserved.	Table of Contents	MCUXpressoIDE_10.1.1_606_Windows	638.1 MB <u><u></u>MCUXpressoIDE_10.1.1_606.ex</u>	te
ABOUT NXP RESOURCES FOLLOW US NEWS < > Investors Mobile Apps Mobile Apps Im Efficiency NXP News Partners Press, News, Blogs Im Efficiency Read More Careers Contact Us Contact Us Read More	select	download Selected Files		
Partiners Careers Contact Us Privacy Terms of Use Terms of Sale Feedback Careers Contact Us Privacy Terms of Use Terms of Sale Feedback Careers Contact Us Careers Contact Us Careers Contact Us Careers Contact Us Careers Contact Us Careers Careers Careers Contact Us Careers	ABOUT NXP	RESOURCES	FOLLOW US	NEWS < >
Careers Contact Us Read More Privacy Terms of Use Terms of Sale Feedback ©2006-2017 NXP Semiconductors. All rights reserved.	Partners	Press, News, Blogs	Y in f 🛛	NXP News
Privacy Terms of Use Terms of Sale Feedback ©2006-2017 NXP Semiconductors. All rights reserved.	Careers	Contact Us		Read More
	Privacy Terms of Use	Terms of Sale Feedback	©2006-2017 NXP Semiconductors. All rights	reserved.
IDE – select and download	IDE - selec	t and download		

2.1.2 Installation of the MCUXpresso IDE

The installation file was obtained using the procedure in the previous chapter 2.1.1. Once it is started, it is recommended that user follows the standard recommended installation setup. Custom installation requires a deeper knowledge of the MCUXpresso IDE.

No additional licenses are required after installing MCUXpresso. The development environment can be used immediately in its entirety.

2.1.3 How to generate MCUXpresso SDK with Bluetooth and NTAG I2C software

From the reason the *ntag_i2c_plus* middleware (which covers the NTAG I²C chip functionality) is missing in the general SDK here is the procedure how to generate the SDK based on the necessary parts for the Bluetooth demo applications and *ntag_i2c_plus* middleware software. Follow the steps bellow.

UM10950

- 1.) Open the welcome page of the MCUXpresso site [4]: https://mcuxpresso.nxp.com/en/welcome
- 2.) Select the configurator based the board selection.



3.) Sign in to the web page – it is necessary step

ODUCTS SOLUTIONS S	UPPORT ABOUT		ALL ~	Search
Sign In or Register				
Sign In Email Address <user e-mail=""> Password Sign in Forgot your password Don't have an account? Reg</user>	17 gister Now	Having trouble? If you are having trouble please open a support re Employees: If you have a core ID (L- and new employees in using you password. If you do NOT have a co employees and new emp NXP facility): Please follo your nxp.com login, using required for account active	with regis quest. egacy Frr d at a leg r NXP co ore ID (Le loyees hi winstruc g your WI ration).	stration or login, eescale employees acy Freescale re ID and oneIT egacy NXP red at a legacy tions to activate BI ID. (VPN access
ABOUT NXP Investors Press, News, Blogs Careers	RESOURCES Mobile Apps Contact Us	Follow US	f 🛛	News 2 May 2018 < > NXP Semiconductors Reports First Quarter 2018 Results Read More

User manual COMPANY PUBLIC Downloaded from Arrow.com. 4.) Enter the keyword "FRDM-KW41Z" and configure the SDK



5.) Select the SDK components

SDK Dashboard	SDF		
	Genera	Select optional components from the list be	elow to be added to your SDK.
GENERAL	MCUX	3 items selected -	
Select Board			
	Develop		
ADMINISTRATION	Selection Generate	Select All Deselect All	
A Notifications		Middleware	
the Destaurance	HOS	CMSIS DSP Library	
Preferences		NTAG I2C	
		USB stack	
	Select (mbedtls	
MCUXpresso IDE		wolfssl	
K MCUXpresso	🖸 Ad	Operating systems	
Config Tools		FreeRTOS ✓	
	This MC		
	downlo	Wireless stack	
		8U2.15.4 MAC	Save changes
	Download	GenESK	Documentation
		SMAC	Base SDK
		Thread	Save obit.
			Middleware:

Downloaded from Arrow.com.

User manual

6.) Verify and start SDK building



7.) Download the SDK

My Recent SDRS	SHOW ALL		~
	SDK Archive Details	Download SDK archive and documentation	_
A Contraction	FRDM-KW41Z 🖋 NEW	* 8 4 *	×
	Build Date: 2018-03-19, Board: FRDM-KW41 OS: Windows, Toolchain: MCUXpresso IDE	z 🔨	
	Components: FreeRTOS, NTAG 12C, BLE SDK Version: KSDK 2.2.0 (2018-01-19)		

2.1.4 Importing the SDK to the MCUXpresso IDE

The final step that needs to be implemented to achieve a fully functional development environment is importing the SDK. This step is very simple and it is done in a drag and drop way. The following procedure shows how to import SDK.

- 1.) Launch the MCUXpresso IDE and select the workspace via the occurred popup window.
- 2.) Switch window to the "Installed SDKs" and via drag & drop technique install the downloaded SDK (see figure Fig 13).

Downloaded from Arrow.com.

User manual



2.2 How to show the demo

2.2.1 HID Demo from website

A ready-made version of the HID_device demo application is available on NXP website. The application consists of integrated *ntag_i2c_plus* middleware and already supports the BT pairing.

This sample project can be directly imported into MCUXpresso as an archived project. In the following chapter is the procedure how to debug this prepared project.

2.2.1.1 MCUXpresso and FRDM-KW41Z SDK

It is necessary to install the MCUXpresso, generate the FRDM-KW41Z SDK and import the SDK to MCUXpresso in accordance the chapter 2.1.

2.2.1.2 Importing of Archived Project

The archived project is packaged in the ZIP format. This project is then imported through the "Quickstart menu" and then "Import project(s) from file system..." (see Fig 14).

	Import project(s) from file system Select the examples archive file to import.
U Quickstart Panel 💁 Breakpoints 🔅 D	eb Projects are contained within archives (.zip) or are unpacked within a directory. Select your project archive or root directory and press <next>. On the next page, select those projects you wish Armoort, and press <finish>.</finish></next>
▼ Start here	Project archives for LPCOpen and 'legacy' examples are provided.
New project Import SDK example(s)	Project archive (zip) Archive Browse
Import project(s) from file system Suild "[]	Project directory (unpacked)
 Clean " [] Debug " [] 	Root directory Browse
	III PAADED

UM10950

Find the archived project and continue importing. Project is automatically copied to the workspace.

Select a directory to corr	n file system	-		
Select a unectory to sea	terror existing cerpse project			
Projects:				
Jid_device_frdm	nkw41z_freertos (hid_device_fre	dmkw41z_freertos/)		Select All
				Deselect All
				R <u>e</u> fresh
I				
Options				
✓ Oppy projects into w	vorkspace			
Working sets			_	
Add project to work	king sets			Ne <u>w</u>
Working sets:				S <u>e</u> lect
	<i>[</i>			
?	< <u>B</u> ack	Next >	Einish	Cancel

2.2.2 Demo running/debugging

Debug the application by clicking on the "Quickstart Panel" from the MCUXpresso on "Debug 'hid_device_frdmkw41z_freertos". Accept the popup window of the J-link term of use.

 	<pre>1056@ static void* osObjectAlloc(const 1057 { 1058</pre>
U Quickstart Panel See Breakpoints 🗱 Debug 🗖 🗖	<pre>1001 l 1062 if(((osObjStruct_t*)pObj 1063 { 1064 ((osObjStruct_t*)pOb 1065 return (void*)pObj; 1066 }</pre>
 Start here New project Import SDK example(s) Import project(s) from file system Build 'hid_device_frdmkw41z_freertos' [Debug] Clean 'hid_device_frdmkw41z_freertos' [Debug] Debug 'hid_device_frdmkw41z_freertos' [Debug] Terminate, Build and Debug 'hid_device_frdmkw41z_freertos' [Debug] 	<pre>1065 } 1067 } 1068 return NULL; 1069 } 1070 #endif 1071 1072© /*! **********************************</pre>
Fig 16. Demo compiling-flashing-debugging	

UM10950

The debugging starts via pressing the F8 on the keyboard (main menu tabs "Run->Resume"), see the Fig 17.

On the FRDM-KW41Z board, the white LED shall be blinking, notifying that the application is running and is in the standby mode.



2.2.3 Behavior of the demo

By pressing the SW4 button for the first time after startup (now the red LED on KW41Z board is blinking). After all this the application is ready for usage or testing.

The following states describe the behavior of the HID_device demo application:

SW3 button pushed and phone is presented to the board:

The FRDM-KW41Z is paired over NFC automatically to the mobile phone. Depending on the version of Android, the application of FRDM-KW41Z is automatically connected and this is demonstrated by a moving cursor on your phone. For some versions of Android, it is needed to go into settings menu, BTconnections and tab on the paired BT-device. Then the connection to the application is finally performed.

<u>NOTE:</u> This step was typically implemented from the mobile manufacturer as they wanted the user to define what the application is allowed to do.

 <u>SW4 button pushed and phone is presented to the board:</u> The NTAG I²C *plus* chip has now the URL of the NTAG I²C demo app written in an NDEF container. It automatically opens up either playstore or the app on the phone.

Downloaded from Arrow.com.

User manual

3. How to add NTAG I²C to the HID_device demo project

For using the NTAG I²C chip the *ntag_i2c_plus* middleware package should be added to the HID Bluetooth demo application. In the next chapters there is procedure how to add the *ntag_i2c_plus* middleware to the FRDM-KW41Z demo application.

3.1.1 HID_device – The demo application

The demo application, which we took as a basis for adding NTAG I²C is **hid_device**. This demo is based on operation system FreeRTOS or like "bare metal" (without operation system). We will take the version with the FreeRTOS. The demo shows a moving arrow on the smart phone display.

3.1.2 Import the HID_device demo application

The HID_device demo application is imported from the SDK. This step assumes that the MCUXpresso and SDK are installed in accordance with chapter 2.1. Then import the project will be done in the following steps.

 In the "Quickstart menu" click the "Import SDK example(s)...". The "SDK Import Wizard" window is open. Select the required SKD (frdmkw41z) and press "Next".



2.) In the "Examples" window search the "wireless_examples" and then "bluetooth" directory (see Fig 19).

Import projects	
Project name prefix: frdmkw41z_	Project name suffix:
Use default location	
Location: workspace\frdmkw41z_	Browse
Project Type	Project Options
	vrary SUK Debug Console
Examples	
type to filter	Varia
Image: cmsis_driver_examples Image: mail matrix	E

- 3.) Then search "hid_device" demo application and select the freeRTOS version only.
- 4.) Press the "Finish" button and the project should be imported.

Name		Version	^
image: maintenance in the second s			
a 🔲 🗧 bluetooth			
> > alert_notification_server			
b eacon			
ble_fsci_black_box			
b e_relay_proxy			=
ble_shell			
blood_pressure_sensor			
Image: power_sensor			
Image: boost in the second			
Image: plucose_sensor			
Image: box is the image of t			
Image:			
heart_rate_sensor			
Image: A market and A market			
m = bm			
V = freertos			
init_host			-
		3	
?)	< Back N	ext > <u>F</u> inish	Cancel
-			

UM10950

User manual COMPANY PUBLIC Downloaded from Arrow.com.

3.1.3 "ntag_i2c_plus" middleware

The procedure how to add the NTAG middleware package is universal for all demo applications which supports FRDM-KW41Z development board. The name of the middleware package for NTAG I²C chip is **ntag_i2c_plus** and the actual version has the extension **ntag_i2c_plus_1.0.0**. It contains whole support software for NTAG I²C chip. The directory with middleware should be located at following directory path, directly in the SDK zip file:

SDK_2.2_FRDM-KW41Z.zip\middleware\ntag_i2c_plus_1.0.0

The internal structure of middleware should has following structure:

Name	Ext	Size
Ê		<dir></dir>
HAL_12C		<dir></dir>
HAL_ISR		<dir></dir>
HAL_NTAG		<dir></dir>
HAL_TMR		<dir></dir>
inc		<dir></dir>
ChangeLogKSDK	txt	17

Fig 21. File structure of the ntag_i2c_plus_1.0.0

3.1.4 How to add the ntag_i2c_plus middleware to the Bluetooth demo

To have a HW support of the NTAG middleware SW there is necessary to add following to the Bluetooth demo application:

- setting for GPIO pins for communication interface I2C
- add the NTAG software handler declaration in to the application source C file
- implement the NTAG timer
- add #includes to the C sources of the BT demo application

<u>NOTE:</u> Parts of C code which have been added for support the NTAG I2C chip are separated by following conditional define:

#ifdef NTAG_I2C

#endif //NTAG_I2C

or following comment is added behind the C code, at the end of the line

#include "fsl_common.h" // added for NTAG middleware

3.1.4.1 GPIO pins setting

The NTAG I2C chip has, in addition to I2C pins, field detection (FD pin) and voltage out (Vout pin - energy harvesting). Only I2C bus is used to connect the NTAG I2C chip to the FRDM-KW41Z development kit. Other pins will remain unoccupied.

I²C pins

UM10950

User manual COMPANY PUBLIC Downloaded from Arrow.com.

GPIO pins of the I²C interface are generally defined in the $pin_mux.c$ file in the function $BOARD_InitI2C(void)$ and should not be redefined in another location. GPIO pins for both I²C interfaces (I2C0 or I2C1) are set to I²C mode within this function.

3.1.4.2 NTAG SW and HW initialization

SW and HW initialization is called in the *main_task()* function in the *ApplMain.c* file.

First the HW initialization is performed by the function *hardware_init()* and there should be added following C-code, bellow the initialization DCDC module:

```
/* Init DCDC module */
BOARD_DCDCInit();
#ifdef NTAG_I2C
/* Init I2C pins for NTAG communication */
BOARD_InitI2C();
#endif // NTAG_I2C
```

Second is the SW initialization performed by function <code>HAL_I2C_InitDevice()</code> and <code>NTAG I²C handler (ntag_handle)</code> is filled by the <code>NFC_InitDevice()</code> function.

At the following C-code lines the SW initialization have to put before application thread calling $(App_Thread())$ in main_task() function.

```
#ifdef NTAG_I2C
    /* Initialize I2C for NTAG communication */
    HAL_I2C_InitDevice(HAL_I2C_INIT_DEFAULT, I2C_MASTER_CLK_SRC,
        NTAG_I2C_MASTER_BASEADDR);
    SystemCoreClockUpdate();
    /* Initialize the NTAG I2C components */
    ntag_handle = NFC_InitDevice((NTAG_ID_T)0, NTAG_I2C_MASTER_BASEADDR);
#endif // NTAG_I2C
    }
    /* Call application task */
    App_Thread( param );
}
```

The last step will be to insert the *ntag_handle* declaration at the beginning of the *ApplMain.c* file.

3.1.4.3 Added #includes to the BLE demo application

Using the "**ntag_i2c_plus**" middleware requires include of headers into BLE demo application source code. Here is the list of files which require to include new headers:

ApplMain.c

#ifdef NTAG_I2C /* NTAG middleware module */ #include "HAL_I2C_driver.h" #include "app_ntag.h" #endif //NTAG_I2C

hid_device.c

#ifdef NTAG_12C /* NTAG handler */ #include "app_ntag.h" #endif //NTAG_12C

3.1.4.4 Added new application NTAG files

There were created 2 application files $(app_ntag.c \text{ and } app_ntag.h)$ that creates an interface between NTAG middleware SW and common demo application.

The *app_ntag.c* source file contains sample functions for working with NDEF messages. Function <u>NFC_MsgWrite()</u> creates and writes the NDEF message in the Type-2 Tag format to the NTAG I²C chip through the ntag_i2c_plus middleware. The write algorithm is NFC-Forum compliance. Function <u>NDEF_Pairing_Write()</u> contains a procedure to create a BTSSP record via using the NDEF library. The same is performing function <u>NDEF_Demo_Write()</u> function. Here is shown how to create NDEF multi-record that contains several types of NDEF records.

The $app_ntag.h$ header file contains predefined blocks of constants (constant fields of data) that are written to the NTAG I²C chip by default during the communication which requires set the default content to the chip's registers or erase the NTAG I²C chip user memory and registers of lock bytes.

3.1.5 BLE Demo Application Extension

The body and state machine of the *HID_device* demo application is in the $hid_device.c$ file. Its extension for BLE pairing and writing NDEF messages to NTAG I²C chip is added to the function <u>BleApp_HandleKeys()</u>. This is because the writing of NDEF messages is done by pressing microswitch SW3 and SW4.

3.1.5.1 NDEF Timer

Within the extension of the HID_device demo application there was necessary to create NDEF timer. This one performs the time counter from the moment when SW3 or SW4 button is pressed. The timer counter is set to 2 seconds. During this time, it is indicated to write the NDEF administration to the NTAG I²C chip. The timer has no other function.

To add the timer, it is necessary to add the following C-code to the *hid_device.c* file:

Add the declaration of the timer handler

This declaration is placed at the beginning to the *hid_device.c* file in to the part "Private memory declarations".

UM10950

#ifdef NTAG_I2C
static tmrTimerID_t mNDEFTimerId;
static bool boNDEFState = FALSE;
#endif

Add the declaration of the timer callback function

NDEF timer callback function declaration is placed to the part "Private functions prototypes".

Allocate / Initialize the timer

There are 3 timers used within the HID_device demo application. The NDEF timer is also necessary to allocate in the function $BleApp_Config()$ in the hid_device.c file, at the same place as the common timers are allocated. Function $TMR_AllocateTimer()$ returns timer ID value which is stored in the variable mNDEFTimerId. The timer ID allocation must be added behind the other timer as it is done at following C-code printout

```
/* Allocate application timers */
    mAdvTimerId = TMR_AllocateTimer();
    mHidDemoTimerId = TMR_AllocateTimer();
    mBatteryMeasurementTimerId = TMR_AllocateTimer();
#ifdef NTAG_12C
    mNDEFTimerId = TMR_AllocateTimer();
```

#endif

Add the timer callback function

There is necessary add the *NDEFTimerCallback()* function at the end of the *hid_device.c* file. If NDEF timer counter expires timer is stop. Then RGB LED is switched off. There is the printout of the call back function at the following lines.

3.1.6 Security change

The sample project for adding NTAG I²C middleware is **hid_device** and is described in chapter 3.1.1. This project requires to enter the password "999999" during the Bluetooth pairing. From this reason is necessary to degrease the security level to remove the password sequence.

Security level is a part of the configuration and is set in the *app_config.c* file. In this file following parameter must be changed

gSecurityMode_1_Level_3_c

to the new parameter

gSecurityMode_1_Level_1_c

Parameter gSecurityMode_1_Level_3_c is used on several places within the app_config.c file. Use the *FIND* function (short key is "CTRL+F") of the KDS IDE to find it and update.

There are last two parameters of the *gPairingParameters* structure which are necessary to change.

parameter:

.securityModeAndLevel = gSecurityMode_1_Level_3_c,

has to be changed to:

.securityModeAndLevel = gSecurityMode_1_Level_1_c,

parameter:

.localloCapabilities = gloDisplayOnly_c,

has to be changed to:

.localloCapabilities = gloNone_c,

parameter

.leSecureConnectionSupported = TRUE,

has to be changed to

.leSecureConnectionSupported = FALSE,

Downloaded from Arrow.com.

User manual

3.2 HID_device project properties

3.2.1 Symbols

Within the project setting is necessary to add following "symbols":

- FRDM_KW41Z
- NTAG_I2C
- I2C_FSL
- HAVE_STDBOOL_H

These symbols are conditional defines for compiler and allows using of the *ntag_i2c_plus* middleware and allows to add required GPIO pins configuration for HW connection with NTAG I²C *plus* PCB board. Following picture shows the place where the symbols are located within the project properties.

There are changes for NDEF library by the green color. This is described in the next chapter 0.



UM10950

© NXP B.V. 2018. All rights reserved.

User manual COMPANY PUBLIC Downloaded from Arrow.com.

3.2.2 Include paths

Within the project setting is necessary to add following "includes":

- "\${workspace_loc:/\${ProjName}/ntag_i2c_plus_1.0.0/inc}"
- "\${workspace_loc:/\${ProjName}/ntag_i2c_plus_1.0.0/HAL_I2C/inc}"
- "\${workspace_loc:/\${ProjName}/ntag_i2c_plus_1.0.0/HAL_NTAG}"
- "\${workspace_loc:/\${ProjName}/ntag_i2c_plus_1.0.0/HAL_NTAG/inc}"
- "\${workspace_loc:/\${ProjName}/ntag_i2c_plus_1.0.0/HAL_ISR}"
- "\${workspace_loc:/\${ProjName}/ntag_i2c_plus_1.0.0/HAL_ISR/inc}"
- "\${workspace_loc:/\${ProjName}/ntag_i2c_plus_1.0.0/HAL_TMR/inc}"

These includes represent the paths which point to source files of the *ntag_i2c_plus* middleware. Following picture shows the place where the includes are located within the project properties.

There are changes for NDEF library by the green color. This is described in the next chapter 0.

type filter text > Resource	Settings		⇔ ♥ ↔ ♥
Builders ⊿ C/C++ Build Build Variables Environment Logging <u>MCU settings</u>	Configuration: Debug [Active] Image: Tool Settings Image: Provide the set of th	Build Artifact 📷 Binary Parsers 🥹 Error Parsers	Manage Configurations
Settings Tool Chain Editor Code Analysis Documentation File Types Formatter Indexer Language Mappings Paths and Symbols Preprocessor Include Pi MCUXpress Config Tools Project References Refactoring History Run/Debug Settings Task Tags > Validation	By MCU C Compiler Delact Preprocessor Debugging Warnings Warnings Miscellaneous McU Assembler General McU Assembler General Dibraries Miscellaneous Multicore Multicore	Include paths (-1) Storotopace_loc:/\$(ProjName//source)* (\$Vorotopace_loc:/\$(ProjName//source)* (\$Vorotopace_loc:?\$(ProjName//source)* (\$Vorotopace_loc:?\$(ProjName)*) (\$Vorotopace]oc:?\$(ProjName)*	의 한 한 와 Includes for ntag_i2c_plus middleware Includes for NDEF library UART_Adapter)*
			Restore Defaults Apply OK Cancel

User manual

3.3 NDEF library usage

The NDEF library supports the use of NDEF messages and greatly facilitates it. This library is used to create NDEF records and to encode and decode NDEF messages. The NDEF library is not currently available in the generated SDK from the NXP website. Its basic version is part of only this BLE pairing HID_device demo application package.

The library is added to the project as uncompiled. This implies the need to add a not a library binary file, but the entire directory with its source code to the HID_device demo application. From this reason is necessary to set the Symbols and include the paths to the MCUXpresso project of the HID_device demo application.

3.3.1 How to add the NDEF library to the Bluetooth demo

For this demo, the NDEF library functions are used only within the *app_ntag.c* file. Therefore, it is necessary to add a following includes to the file.

```
/* NDEF library headers */
#ifdef NDEF_LIBRARY
#include "NDEF_Rec_Text.h"
#include "NDEF_Rec_Uri.h"
#include "NDEF_Rec_BluetoothSsp.h"
#include "NDEF_Rec_Aar.h"
#include "NDEF_Rec_GenericPayload.h"
#include "NDEF_Record.h"
#endif // NDEF_LIBRARY
```

3.3.2 Symbols

Within the project setting is necessary to add only one "symbols".

• NDEF_LIBRARY

The setting where to place it is in the picture in chapter 3.2.1

3.3.3 Include paths

Within the project setting is necessary to add following "includes":

- "\${workspace_loc:/\${ProjName}/Ndef_Library/inc}"
- "\${workspace_loc:/\${ProjName}/Ndef_Library/inc/Internal}"
- "\${workspace_loc:/\${ProjName}/Ndef_Library/inc/Records}"

The setting where to place it is in the picture in chapter 3.2.2

3.3.4 How not to use the NDEF library

An option for not using the NDEF library is added to the HID_device demo application. NDEF messages (the BTSSP and smart poster about the NTAG I²C plus demo board) are hardcoded in this case and the C-code is already written in the *app_ntag.c* file (see chapter 0). This can be done if **NDEF_LIBRARY** define is not defined.

Then the hardcoded NDEF message is processed in to the standard Tag-2 Type format and written to the NTAG I²C chip. There is no difference to the case when the NDEF library is used.

<u>NOTE 1:</u> The BTSSP NDEF message is also described by the comments. The description consists only of the mandatory parameter and necessary optional data, which are required for BT pairing. This is mainly from reason to very <u>quickly understand the BTSSP NDEF message format</u>.

UM10950

Downloaded from Arrow.com.

User manual COMPANY PUBLIC

4. Abbreviations

Table 2.	Abbreviation	S
Acronym		Description
BLE		Bluetooth Low Energy – standard 4.2.
BTSSP		Bluetooth Secure Simple Pairing (NDEF record or message)
CMSIS		Cortex Microcontroller Software Interface Standard
FLASH		an electronic non-volatile computer storage medium
FRDM-KW	41	Freedom board development kit based on MKW41Z microcontroller
GPIO		General Purpose Input Output
HW		Hardware
IDE		Integrated Development Environment
IRQ		Interrupt Request
MAC addre	ess	Media Access Control address
MCU		Microcontroller Unit
NDEF		NFC Data Exchange Format
NFC		Near Field Communication
OS		Operation System
PCB		Printed Circuit Board
RGB LED		Full color LED (Red-Green-Blue)
SDK		Software Development Kit
SW		Software

UM10950

Downloaded from Arrow.com.

5. References

- [1] FRDM-KW41Z: NXP[®] Freedom Development Kit for Kinetis[®] KW41Z/31Z/21Z MCUs: http://www.nxp.com/products/software-and-tools/hardware-developmenttools/freedom-development-boards/nxp-freedom-development-kit-for-kinetis-kw41z-31z-21z-mcus:FRDM-KW41Z?fsrch=1&sr=1&pageNum=1
- [2] NTAG I²C *plus* Explorer Kit (OM5569) general NXP web site: <u>http://www.nxp.com/products/identification-and-security/nfc-and-reader-ics/connected-tag-solutions/ntag-ic-plus-explorer-kit-demo-kit:OM5569-NT322E</u>
- [3] OM23221ARD: NTAG I²C *plus* kit for Arduino® pinout <u>https://www.nxp.com/products/identification-and-security/nfc/nfc-tags-for-</u> <u>electronics/ntag-ic-iplus-i-kit-for-arduino-</u> <u>pinout:OM23221ARD?fsrch=1&sr=1&pageNum=1</u>
- [4] MCUXpresso SDK Builder https://mcuxpresso.nxp.com/en/welcome

UM10950

6. Legal information

6.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

6.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the

customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of noninfringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US\$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

6.3 Licenses

Purchase of NXP ICs with NFC technology

Purchase of an NXP Semiconductors IC that complies with one of the Near Field Communication (NFC) standards ISO/IEC 18092 and ISO/IEC 21481 does not convey an implied license under any patent right infringed by implementation of any of those standards. Purchase of NXP Semiconductors IC does not include a license to any NXP patent (or other IP right) covering combinations of those products with other products, whether hardware or software.

6.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

NTAG - is a trademark of NXP B.V.

I²C-bus — is a trademark of NXP B.V.

User manual

7. List of figures

Fig 1.	HW connection with the NTAG I2C plus board (OM5569)	1 3
Fig 2.	HW connection with the Arduino NTAG I2C pl board (OM23221ARD)	lus 4
Fig 3.	MCUXpresso SDK builder – software tools	5
Fig 4.	IDE – starting the download	6
Fig 5.	IDE – select a version	6
Fig 6.	IDE – select and download	7
Fig 7.	MCUXpresso SDK builder – board selection	8
Fig 8.	MCUXpresso login page	8
Fig 9.	KW41Z board selection	9
Fig 10.	SDK components selection	9
Fig 11.	SDK creating	.10
Fig 12.	SDK downloading	.10
Fig 13.	SDK installing to the MCUXpresso	.11
Fig 14.	Setting the archived project	.11
Fig 15.	Importing of archived project	.12
Fig 16.	Demo compiling-flashing-debugging	.12
Fig 17.	Start debugging	.13
Fig 18.	SDK selection	.14
Fig 19.	Finding the required project	.15
Fig 20.	Importing the required project	.15
Fig 21.	File structure of the ntag_i2c_plus_1.0.0	.16
Fig 22.	Demo application symbols for NTAG I2C plus	21
Fig 23.	Demo application includes for NTAG I2C plus	22

Downloaded from Arrow.com.

User manual

8. Contents

1.	Introduction and HW setup3
1.1	HW setup3
1.1.1	NTAG I ² C plus board
1.1.2	Arduino NTAG I ² C plus board4
2.	Quick start-up the demo application5
2.1	Installation of the MCUXpresso IDE and SDK5
2.1.1	How to download the installation file of the MCUXpresso IDE5
2.1.2	Installation of the MCUXpresso IDE7
2.1.3	How to generate MCUXpresso SDK with
	Bluetooth and NTAG I2C software7
2.1.4	Importing the SDK to the MCUXpresso IDE10
2.2	How to show the demo11
2.2.1	HID Demo from website11
2.2.1.1	MCUXpresso and FRDM-KW41Z SDK11
2.2.1.2	Importing of Archived Project11
2.2.2	Demo running/debugging12
2.2.3	Behavior of the demo13
3.	How to add NTAG I ² C to the HID_device demo
	project14
3.1.1	HID_device – The demo application14
3.1.2	Import the HID_device demo application14
3.1.3	"ntag_i2c_plus" middleware16
3.1.4	How to add the ntag_i2c_plus middleware to the
	Bluetooth demo16
3.1.4.1	GPIO pins setting16
3.1.4.2	NIAG SW and HW initialization
3.1.4.3	Added #includes to the BLE demo application.18
3.1.4.4	Added new application NTAG files
3.1.5	BLE Demo Application Extension
3.1.3.1	NDEF TIMEL
3.1.0	Security change
3.Z	Symbols 21
222	Symbols
3.2.2	NDEE library usage
331	How to add the NDEE library to the Bluetooth
0.0.1	demo 23
332	Symbols 23
3.3.3	Include paths
3.3.4	How not to use the NDEF library
4.	Abbreviations25
5.	References
6.	Legal information
	U

6.1	Definitions	27
6.2	Disclaimers	27
6.3	Licenses	27
6.4	Trademarks	27
7.	List of figures	28
8.	Contents	29

Please be aware that important notices concerning this document and the product(s) described herein, have been included in the section 'Legal information'.

© NXP B.V. 2018.

For more information, please visit: http://www.nxp.com

Date of release: 5 June 2018

422220 Document identifier: UM10950