

# AN11784

## PN7462AU How to integrate RTOS

Rev. 1.1 — 13 September 2016  
357811

Application note  
COMPANY PUBLIC

### Document information

Info	Content
<b>Keywords</b>	PN7462AU How to integrate RTOS
<b>Abstract</b>	The following document describes steps required for integration of RTOS with PN7462AU Firmware.



## Revision history

Rev	Date	Description
1.1	20160913	Figures updated
1.0	20160309	First official release

## Contact information

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

## 1. Introduction

---

This document enlists steps and guidelines required for integration of FreeRTOS with PN7462AU Firmware.

Unless and otherwise mentioned, the files and folders referred here are relative to the ROOT Folder of the PN7462AU Firmware Package.

## 2. Preconditions

---

The following pre-conditions are assumed before integration Free RTOS with PN7462AU Firmware.

1. PN7462AU Firmware source and evaluation package is available  
(If not available, contact PN7462AU CAS)
2. LPCXpresso IDE version 8.0.0\_526 or above is installed  
(If not available, download from [www.nxp.com/lpcware](http://www.nxp.com/lpcware))
3. PN7462AU Plug-in is installed into LPCXpresso IDE  
(If not available, contact PN7462AU support)
4. FreeRTOSV8.2.0 is available.  
(If not available, download from [www.nxp.com/redirect/freertos.org/](http://www.nxp.com/redirect/freertos.org/))
5. Awareness of FreeRTOS Porting and Integration.  
(This application note is a guideline document, it is not a tutorial on integration of FreeRTOS.)
6. Board bring-up of the DEMO/Evaluation board already completed.  
e.g. Jumper settings/Power Supply/Reference EEPROM contents inside the PN7462AU IC already completed.

### 3. Integrating FreeRTOS package

#### 3.1 Location to copy FreeRTOS code

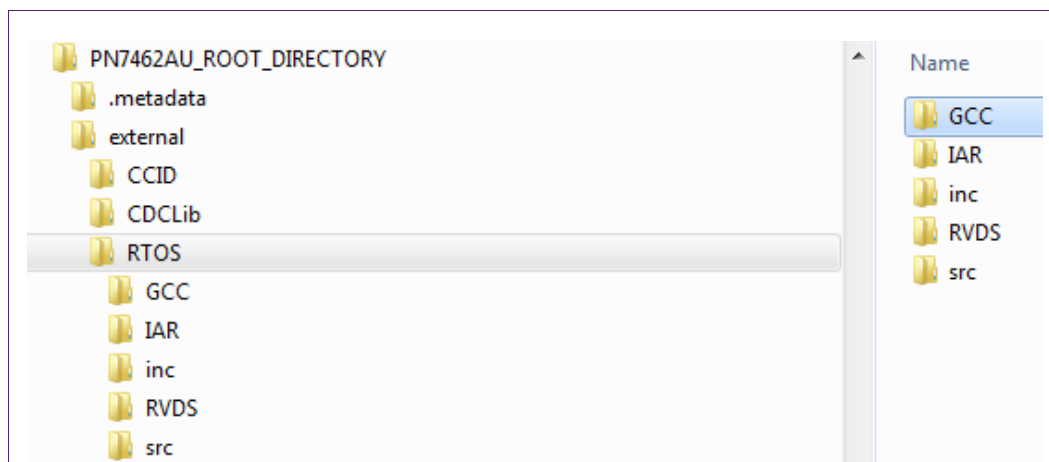


Fig 1. Location to copy FreeRTOS source code

FreeRTOS source code has to be copied to specific folders within  
<RootFirmwareFolder>/external/RTOS.

### 3.2 Files used for porting/integration

The following files have to be copied from FreeRTOS Package to the corresponding sub directory inside <RootFirmwareFolder>/external/RTOS.

**Table 1. Free RTOS Files**

Source	Destination sub directory inside <RootFirmwareFolder>/external/RTOS
FreeRTOS\Source\include\croutine.h	inc\croutine.h
FreeRTOS\Source\include\deprecated_definitions.h	inc\deprecated_definitions.h
FreeRTOS\Source\include\event_groups.h	inc\event_groups.h
FreeRTOS\Source\include\FreeRTOS.h	inc\FreeRTOS.h
FreeRTOS\Source\include\list.h	inc\list.h
FreeRTOS\Source\include\mpu_wrappers.h	inc\mpu_wrappers.h
FreeRTOS\Source\include\portable.h	inc\portable.h
FreeRTOS\Source\include\projdefs.h	inc\projdefs.h
FreeRTOS\Source\include\queue.h	inc\queue.h
FreeRTOS\Source\include\semphr.h	inc\semphr.h
FreeRTOS\Source\include\StackMacros.h	inc\StackMacros.h
FreeRTOS\Source\include\task.h	inc\task.h
FreeRTOS\Source\include\timers.h	inc\timers.h
FreeRTOS\Source\event_groups.c	src\event_groups.c
FreeRTOS\Source\portable\MemMang\heap_3.c	src\heap_3.c
FreeRTOS\Source\portable\MemMang\heap_2.c	src\heap_2.c
FreeRTOS\Source\list.c	src\list.c
FreeRTOS\Source\queue.c	src\queue.c
FreeRTOS\Source\tasks.c	src\tasks.c
FreeRTOS\Source\timers.c	src\timers.c
FreeRTOS\Source\portable\GCC\ARM_CM0\portmacro.h	GCC\portmacro.h
FreeRTOS\Source\portable\GCC\ARM_CM0\port.c	GCC\port.c
FreeRTOS\Source\portable\IAR\ARM_CM0\port.c	IAR\port.c
FreeRTOS\Source\portable\IAR\ARM_CM0\portasm.s	IAR\portasm.s
FreeRTOS\Source\portable\IAR\ARM_CM0\portmacro.h	IAR\portmacro.h
FreeRTOS\Source\portable\RVDS\ARM_CM0\port.c	RVDS\port.c
FreeRTOS\Source\portable\RVDS\ARM_CM0\portmacro.h	RVDS\portmacro.h
FreeRTOS\Demo\CORTEX_M0_LPC1114_LPCXpresso\RTOSDemo\Source\FreeRTOSConfig.h (After Modifications. See Section 3.3 FreeRTOSConfig.h below)	inc\FreeRTOSConfig.h

### 3.3 FreeRTOSConfig.h

FreeRTOS\Demo\CORTEX\_M0\_LPC1114\_LPCXpresso\RTOSDemo\Source\FreeRTOSConfig.h has been used as reference code for the RTOS Porting.

On top of the reference FreeRTOSConfig.h the following modifications are needed.

**Table 2. Configuration changes to FreeRTOSConfig.h**

Configurable #define	Value	Notes
SystemCoreClock	20000000	20MHz Clock. Replace extern uint32_t SystemCoreClock; with #define SystemCoreClock 20000000
configMAX_PRIORITIES	8	
configMINIMAL_STACK_SIZE	32	
configTOTAL_HEAP_SIZE	5500	Required if heap_2.c being used. With heap_3.c, this is optional
configTIMER_TASK_PRIORITY	1	
configTIMER_QUEUE_LENGTH	10	
INCLUDE_vTaskDelete	0	
INCLUDE_vTaskCleanUpResources	0	
INCLUDE_vTaskDelayUntil	0	
INCLUDE_xEventGroupSetBitFromISR	1	
INCLUDE_xTimerPendFunctionCall	1	
INCLUDE_xSemaphoreGetMutexHolder	0	
INCLUDE_xEventGroupSetBitFromISR	1	
INCLUDE_xTimerPendFunctionCall	1	
INCLUDE_xSemaphoreGetMutexHolder	0	

## 4. Updating phRtos wrapper

The phRtos module/component has been created to facilitate porting across RTOS and NonRTOS based designs.

phCommon\inc\wrapper\phRtos\_RTOS.h and phRtos\src\phRtos\_RTOS.c have to be updated to integrate phRtos with FreeRTOS. The APIs called by the supplied phRtos\_RTOS.h and phRtos\_RTOS.c correspond to APIs as provided by a dummy EXAMPLE\_RTOS. The end user (who has Awareness of FreeRTOS Porting and Integration – as mentioned in Section 2 - Preconditions on page 3 ) is expected to update phRtos\_RTOS.h and phRtos\_RTOS.c as listed in API Mapping between EXAMPLE\_RTOS and FreeRTOS ([Table 3](#) belowbelow).

**Table 3. API Mapping between EXAMPLE\_RTOS and FreeRTOS**

EXAMPLE_RTOS API/Value	FreeRTOS API/Value
>Defines / PreProcessor Macros<	
EXAMPLE_RTOS_MAXIMUM_DELAY	portMAX_DELAY
EXAMPLE_RTOS_STACK_TYPE	portSTACK_TYPE
EXAMPLE_RTOS_FALSE	pdFALSE
EXAMPLE_RTOS_TRUE	pdTRUE
>Header Files<	
EXAMPLE_RTOS_PortingMacros.h	portmacro.h
EXAMPLE_RTOS_MAIN_API.h	FreeRTOS.h
EXAMPLE_RTOS_TASK_API.h	task.h
EXAMPLE_RTOS_QUEUE_API.h	queue.h
EXAMPLE_RTOS_SEMAPHORE_API.h	semphr.h
EXAMPLE_RTOS_EVENT_API.h	event_groups.h
EXAMPLE_RTOS_TIMER_API.h	timers.h
>Data Types<	
EXAMPLE_RTOS_DefaultType_t	BaseType_t
EXAMPLE_RTOS_TaskHandleType_t	TaskHandle_t
>Task/Task Scheduler APIs<	
EXAMPLE_RTOS_TaskSchedulerStart	vTaskStartScheduler
EXAMPLE_RTOS_TaskSchedulerStop	vTaskEndScheduler
EXAMPLE_RTOS_TaskSchedulerGetState	xTaskGetSchedulerState
EXAMPLE_RTOS_TaskCreate	xTaskCreate
EXAMPLE_RTOS_TaskDelete	vTaskDelete
EXAMPLE_RTOS_TaskGetName	pcTaskGetTaskName
EXAMPLE_RTOS_TaskDelay	vTaskDelay
EXAMPLE_RTOS_TaskSuspend	vTaskSuspend
EXAMPLE_RTOS_TaskSuspendAll	vTaskSuspendAll
EXAMPLE_RTOS_TaskResumeAll	xTaskResumeAll
EXAMPLE_RTOS_TaskResume_ISRSAFE	xTaskResumeFromISR
EXAMPLE_RTOS_TaskResume	vTaskResume
EXAMPLE_RTOS_TaskYield	taskYIELD
EXAMPLE_RTOS_TaskGetState	eTaskGetState
EXAMPLE_RTOS_TaskSetPriority	vTaskPrioritySet
EXAMPLE_RTOS_TaskGetPriority	uxTaskPriorityGet
EXAMPLE_RTOS_TaskGetStackWaterMark	uxTaskGetStackHighWaterMark
EXAMPLE_RTOS_YIELD_FROM_ISR	portYIELD_FROM_ISR

EXAMPLE_RTOS API/Value	FreeRTOS API/Value
<b>&gt;Queue APIs&lt;</b>	
EXAMPLE_RTOS_QueueCreate	xQueueCreate
EXAMPLE_RTOS_QueueDelete	vQueueDelete
EXAMPLE_RTOS_QueueReset	xQueueReset
EXAMPLE_RTOS_QueuePost_ISRSAFE	xQueueSendFromISR
EXAMPLE_RTOS_QueuePost	xQueueSend
EXAMPLE_RTOS_QueueGet_ISRSAFE	xQueueReceiveFromISR
EXAMPLE_RTOS_QueueGet	xQueueReceive
<b>&gt;Semaphore/Mutex APIs&lt;</b>	
EXAMPLE_RTOS_SemaphoreRelease_ISRSAFE	xSemaphoreGiveFromISR
EXAMPLE_RTOS_SemaphoreRelease	xSemaphoreGive
EXAMPLE_RTOS_SemaphoreGetMutexHolder	xSemaphoreGetMutexHolder
EXAMPLE_RTOS_SemaphoreDelete	vSemaphoreDelete
EXAMPLE_RTOS_SemaphoreCreate_Mutex	xSemaphoreCreateMutex
EXAMPLE_RTOS_SemaphoreCreate_Count	xSemaphoreCreateCounting
EXAMPLE_RTOS_SemaphoreAcquire_ISRSAFE	xSemaphoreTakeFromISR
EXAMPLE_RTOS_SemaphoreAcquire	xSemaphoreTake
<b>&gt;Events APIs&lt;</b>	
EXAMPLE_RTOS_EventBitsCreate	xEventGroupCreate
EXAMPLE_RTOS_EventBitsSet_ISRSAFE	xEventGroupSetBitsFromISR
EXAMPLE_RTOS_EventBitsSet	xEventGroupSetBits
EXAMPLE_RTOS_EventBitsWait	xEventGroupWaitBits
EXAMPLE_RTOS_EventBitsClear_ISRSAFE	xEventGroupClearBitsFromISR
EXAMPLE_RTOS_EventBitsClear	xEventGroupClearBits
EXAMPLE_RTOS_EventDelete	vEventGroupDelete
<b>&gt;Timer APIs&lt;</b>	
EXAMPLE_RTOS_TimerCreate	xTimerCreate
EXAMPLE_RTOS_TimerGetID	pvTimerGetTimerID
EXAMPLE_RTOS_TimerStart	xTimerStart
EXAMPLE_RTOS_TimerStop	xTimerStop
EXAMPLE_RTOS_TimerReset	xTimerReset
EXAMPLE_RTOS_TimerDelete	xTimerDelete
EXAMPLE_RTOS_TimerChangeDuration	xTimerChangePeriod
<b>&gt;MISC APIs&lt;</b>	
EXAMPLE_RTOS_GetTickCount_ISRSAFE	xTaskGetTickCountFromISR
EXAMPLE_RTOS_GetTickCount	xTaskGetTickCount
EXAMPLE_RTOS_CallBackOnEveryTick	vApplicationTickHook
EXAMPLE_RTOS_NoFreeMemory	vApplicationMallocFailedHook
EXAMPLE_RTOS_StackOverflowDetected	vApplicationStackOverflowHook



## 5. Legal information

### 5.1 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.

### 5.2 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 non-infringement, 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 of 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.

### 5.3 Licenses

#### Purchase of NXP ICs with ISO/IEC 14443 type B functionality



This NXP Semiconductors IC is ISO/IEC 14443 Type B software enabled and is licensed under Innovatron's Contactless Card patents license for ISO/IEC 14443 B.

The license includes the right to use the IC in systems and/or end-user equipment.

#### RATP/Innovatron Technology

#### 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.

### 5.4 Trademarks

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

**MIFARE** — is a trademark of NXP B.V.

6. Contents

1. Introduction .....3

2. Preconditions .....3

3. Integrating FreeRTOS package .....4

3.1 Location to copy FreeRTOS code .....4

3.2 Files used for porting/integration .....5

3.3 FreeRTOSConfig.h .....6

4. Updating phRtos wrapper.....7

5. Legal information .....9

5.1 Definitions .....9

5.2 Disclaimers.....9

5.3 Licenses .....9

5.4 Trademarks.....9

6. Contents.....10

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