



## MCR20A 802.15.4 2.4GHz Standalone Transceiver Integration with Kinetis K or L Series MCUs FTF-SNT-F1227

Tudor Stănescu | Software Development Manager J U N E . 2 0 1 5





External Use

Presexule, the Pressum tops, AIVAec, C-5, CodeTEST, CodeMercor, ColdPine, ColdPine, O-New, the Emergy Efficient Solutions tops, Kinetis, Magnin, motival T, PED, PowerGLACG, Processer Espen, CuricQ, Control Commenge, Convex, Ready Page, SaleAsawan, the SaleAsawan tops, Ster Cine, Styrpmonzy, Vertillä, Vythrit and Xinnals are trademarks of Presscale Semicontaction: (in, Reju, U.S. Pat, S. Th. CH. Antata, BeeKS, Baseltan, Cameliel, Freise, Layerscape, MXC, Partorn in a Fackage, CURCC Engine, SAMATMOS, Tawar, Turbolink and UMEMS are trademarks of Presscale Beniconductor, Inc. All other product or service norms in the property of their respective owners. B 2015 Presscale Beniconductor, Inc.

## **Session Introduction**

- This hands-on session aims to teach how to use the Freescale MCR20A IEEE® 802.15.4 wireless transceiver with Freescale software and development boards.
- This presentation is relevant to all who want to develop IEEE® 802.15.4based connectivity and Internet of Things (IoT)-enabling applications or add connectivity capabilities to existing designs:
  - This presentation is important because it shows how to use a readily-available, easy to use development environment for IoT.
  - The attendees will be become familiar with Freescale 802.15.4 technology and will have a great starting point in their application and product development.
  - One major problem addressed in this presentation is porting/migrating 802.15.4 connectivity capabilities from one MCU to another.
- Presenter: Tudor Stănescu manager of the IEEE® 802.15.4 and Bluetooth® software team.
- Time allocated to this presentation is 2 hours, out of which 15 minutes will be allotted for questions. Questions will also be taken during the session itself.



## **Session Objectives**

- After completing this session you will be able to:
  - Understand how the MCR20A Freescale offering can benefit your product design by enhancing it with connectivity capabilities and provide you a competitive edge in the IoT space.
  - Configure the various MCR20A-based evaluation board setups.
  - Install and configure the MCR20A IEEE® 802.15.4 software package on top of Kinetis SDK.
  - Understand the IEEE 802.15.4 software architecture and functionality.
  - Appreciate the ease with which an IEEE® 802.15.4-based IoT application or stack can be migrated from one MCU to another, using the MCR20A wireless transceiver.





## Agenda

- Hardware Introduction 20 minutes
  - MCR20A Silicon Overview
  - MCR20A Development Board Overview
  - KL46Z, KL26Z and K64F FRDM Boards Overview
- Software Introduction 40 minutes
  - Kinetis SDK as Connectivity Foundation
  - IEEE® 802.15.4 MAC/PHY Software and Applications
- Porting the MCR20A IEEE® 802.15.4 Software to FRDM-KL26Z (Hands-on Session with Examples) – 45 minutes
  - Clock Configurations
  - Transceiver Driver and SPI Interfacing
  - Serial Connectivity
  - GPIOs
  - Linker configurations
- Conclusions and Q&A 15 minutes



## Hands-On Session Infrastructure

- The following software and hardware is provided for this session:
  - 18 x training laptops
  - Development Boards:
    - 36 x FRDM-CR20A shields
    - 18 x FRDM-K64F
    - 5 x FRDM-KL46Z
    - 18 x FRDM-KL26Z
  - Kinetis SDK v1.1.0 installation
  - Kinetis SDK v1.1.0 SA for KL26Z and KW01Z
  - MCR20A IEEE® 802.15.4 Software Package
  - IAR Embedded Workbench v7.40.1 installation
  - Scooter Software Beyond Compare
  - PuTTY terminal by Simon Tatham



## Hardware Introduction



External Use 5 #

#FTF2015



ŏ

# Hardware Introduction - Summary

This section outlines the most important features and competitive advantages of the MCR20A wireless transceiver and the applicable evaluation boards.



**MCR20AVHM Silicon** Main Features and Differentiators



FRDM-CR20A Arduino<sup>™</sup> Shield Features and Configuration Options



**MCU FRDM Boards** Comparative Presentation of FRDM-KL46Z, FRDM-K64F and FRDM-KL26Z

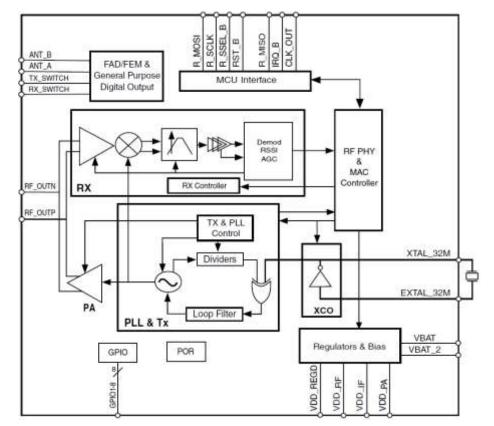








## MCR20A – Silicon Overview



#### **Features:**

- High-performance 2.4 GHz IEEE 802.15.4 RF transceiver
- -102 dBm sensitivity +8 dBm maximum output power reducing the need for external power amplifiers
- · Low-power preamble search receive mode (LPPS)
- TX 17 mA @ 0 dBm and RX 19 mA typical
- + TX 18 mA @ 0 dBm and RX 19.5 mA max
- Dual PAN support
- Support for MBAN frequencies (2.36-2.48GHz)
- Supports single ended and fast diversity antenna options: single 50 ohm antenna uses single balun to reduce component count and cost
- Packet processor for hardware acceleration PHY/MAC support
- Supports clock output to host microcontroller of either 32 kHz or 4 MHz
- SPI Interface
- 8 GPIOs (+4 outputs via FAD manual control)
- Package: 5x5 32-pin LGA
- Operating parameters: -40 °C to 105°C, 1.8 to 3.6 V



#FTF2015

## **MCR20A Differentiating Features**

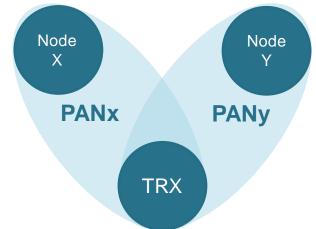
### Dual PAN operation:

- Duplicate registers for PAN ID, short address, channel, long address and coordinator capability to allow the network node using MCR20A to participate in two 802.15.4 personal area networks
- Active promiscuous mode:
  - Promiscuous mode with ACK for packets that are addressed to the node; allows implementation of smart sniffers, that are part of the network and analyze the protocol at the same time
- Fast Antenna Diversity (FAD):
  - Control of an external antenna switch that will select between two antennas based on RSSI
- Low Power Preamble Search (LPPS):
  - The demodulator and other sub-sections of the receiver can be independently duty cycled at 50%. 1.5 dB sensitivity traded off for 4 mA current saving



## **Dual-PAN Operation**

- Allows a single 802.15.4 radio to participate in 2 different PANs simultaneously, with 2 different stacks as well
- Maintains 2 sets of network parameters for each PAN :
  - Channel0/1 PAN Channel
  - MacPanID0/1 PAN ID
  - MacShortAddrs0/1 Device short address
  - MacLongAddrs0/1 Device extended address
  - PANCORDNTR0/1 PAN Coordinator capability



- The transition from one PAN to the other one can be manual (under software control) or automatic
- Automatic transition is done using a programmable timer with a PAN Dwell Time from 0.5ms to 3.2s
- If both PAN are defined on the same channel, TRX is able to process both PAN simultaneously (no PAN Dwell Time to define or channel switch time to consider)
- If PANs are on different channels, channel switch is 56 µs Time to switch, poll, receive packet and switch back is <10mS</li>

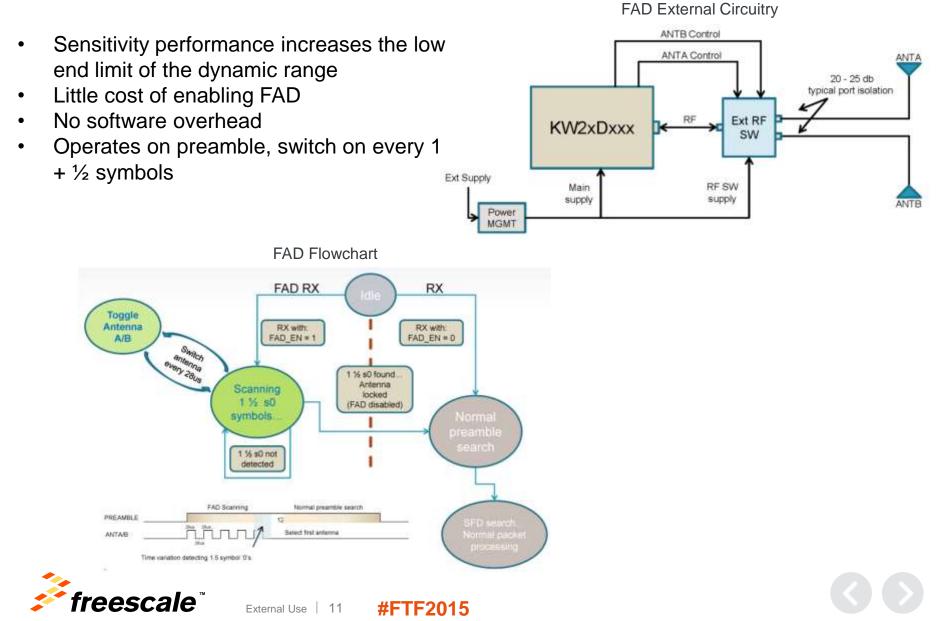


### **Active Promiscuous Mode**

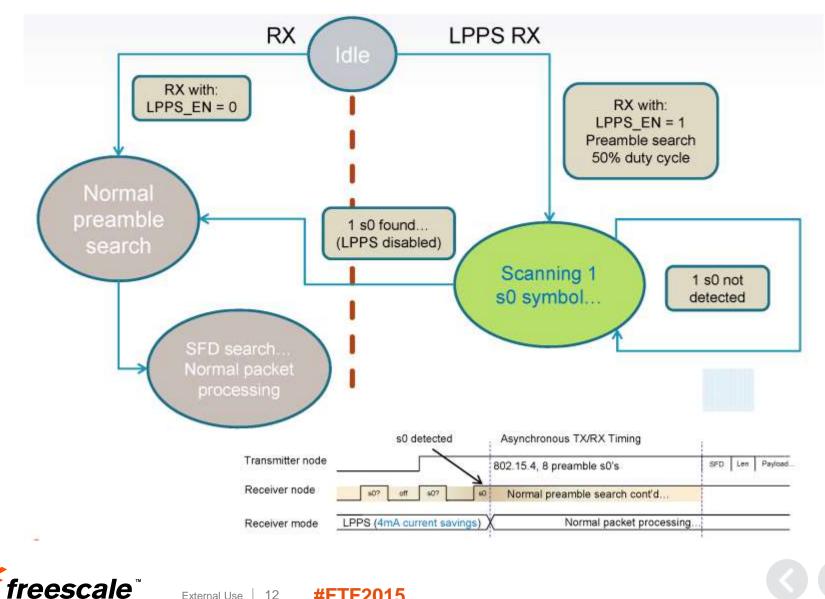
- Works like a typical Promiscuous Mode, except:
- Device will automatically transmit an ACK packet for frames that are addressed to it.
- Allows for building smart sniffers that can be used as commissioners in various networks
- The active promiscuous device can monitor the network and configure it at the same time
- The active promiscuous device can operate as a node in the network and as a sniffer at the same time



## **Fast Antenna Diversity**



### Low Power Preamble Search



External Use | 12

**#FTF2015** 

## **Development Hardware**

FRDM/ArduinoTM Form Factors for MCR20A and MCUs

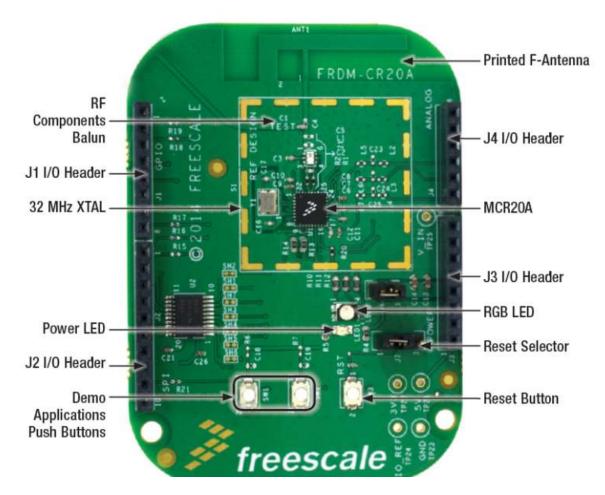








### **FRDM-CR20A - Overview**



External Use | 14

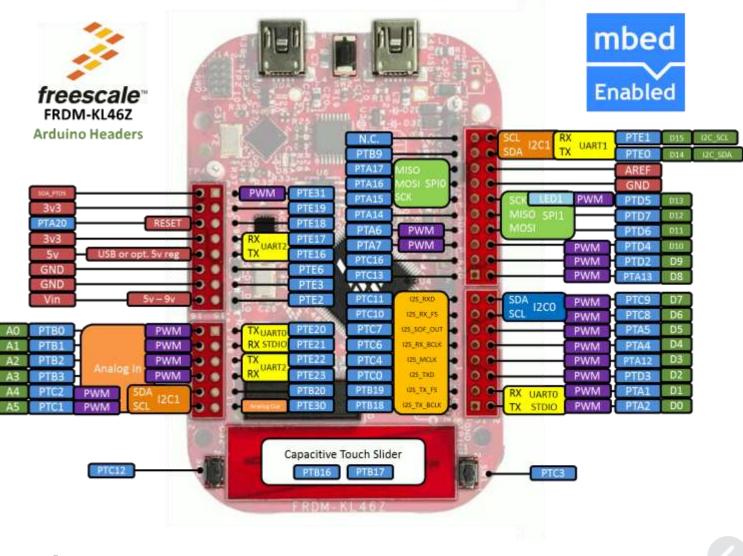
#FTF2015

#### **Features:**

- MCR20AVHM 802.15.4 Transceiver 2.36-2.48GHz frequencies
- SPI interface
- Two (2) user push-button switches for interrupts (SW2/SW3) driven by external MCU
- One (1) RGB LED indicator driven by external MCU
- PCB inverted F-type antenna and SMA RF port available
- Minimum number of matching components and external balun
- Form factor compatible with Arduino<sup>™</sup> R3 pin layout
- Standard FRDM daughter card mounting interface (Shield)
- Can be directly connected to the FRDM-K64F, FRDM-KL46 and FRDM-KL26Z

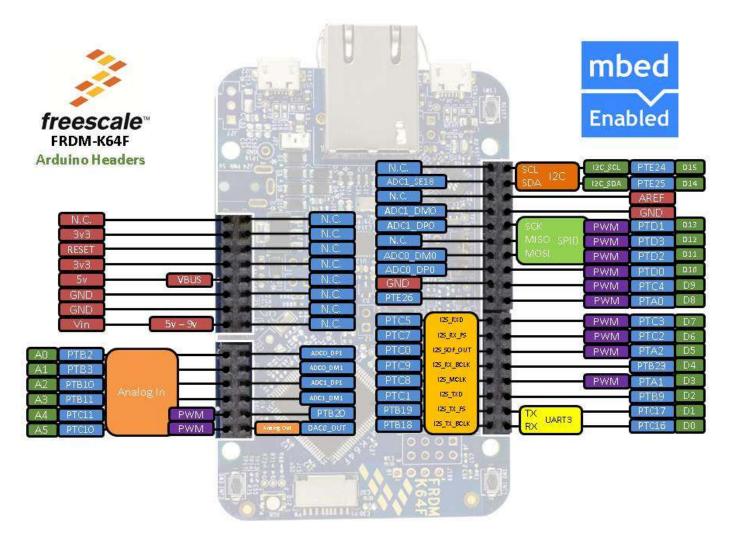


### **FRDM-KL46Z Arduino Headers**





### **FRDM-K64F Arduino Headers**







 $\langle \rangle$ 

### **Supported Network Nodes in the Software**



FRDM-CR20A + FRDM-K46

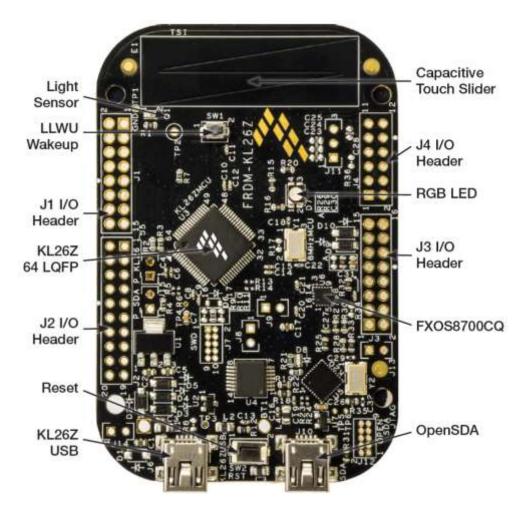






External Use | 17 **#FTF2015** 

### **FRDM-KL26Z - Overview**



External Use | 18

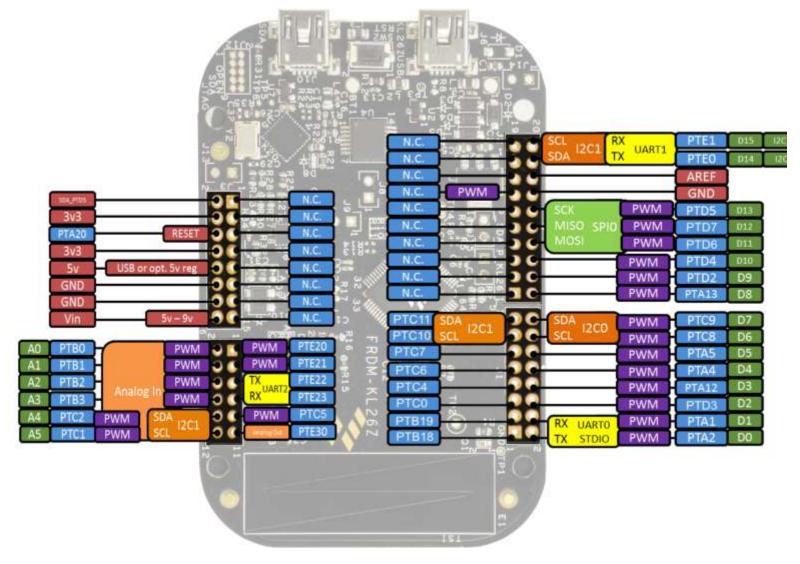
**#FTF2015** 

#### **Features:**

- MKL26Z128VLH4 MCU 48MHz, 128KB Flash, 16KB SRAM, USB OTG (FS), 64LQFP
- Capacitive touch slider, FXOS8700CQ accelerometer and magnetometer, Tri-color LED, ambient light sensor
- Flexible power supply options USB, coin cell battery, external source
- Easy access to MCU I/O
- Battery-ready, power-measurement access points
- Form factor compatible with Arduino<sup>™</sup> R3 pin layout
- OpenSDA debug interface
  - Mass storage device flash programming interface (default) no tool installation required to evaluate demo apps
  - P&E Debug interface provides run-control debugging and compatibility with IDE tools
  - CMSIS-DAP interface: ARM standard for embedded debug interface
  - More information at <u>www.freescale.com/opensda</u>



### **FRDM-KL26Z Arduino Headers**





External Use | 19

**#FTF2015** 

### **Newly Supported Network Node After Porting**



### FRDM-CR20A + FRDM-KL26Z





External Use | 20 **#FTF2015** 

# Software Introduction

14

**#FTF2015** 

External Use 21





# Software Introduction - Summary

This section outlines the building blocks of the software enablement for the MCR20A wireless transceiver, running on Kinetis microcontrollers and serving as the essential foundation for IoT applications.



Kinetis SDK

Foundation for Connectivity Solutions

External Use 22



Freescale IEEE® 802.15.4 MAC and PHY Standard, Software Architecture, Features and Applications





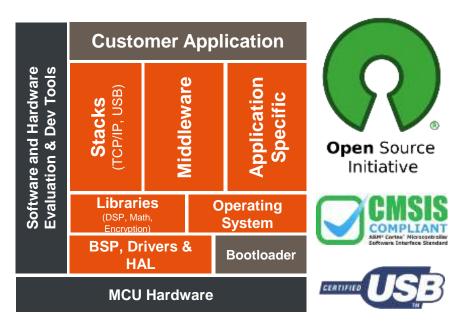
## **Kinetis Software Development Kit (SDK)**



The software framework and reference for Kinetis MCU application development



Hardware abstraction, peripheral drivers, stacks, RTOS's, utilities, and usage examples; delivered in C source



### **Product Features**

- Open source hardware abstraction layer (HAL) provides APIs for all Kinetis hardware resources
- BSD-licensed set of peripheral drivers with easy-to-use C-language APIs
- Comprehensive HAL and driver usage examples and sample applications for RTOS and bare-metal
- CMSIS-CORE compatible startup plus CMSIS-DSP library and examples
- RTOS Abstraction Layer (OSA) with support for FreeRTOS, Freescale MQX, Micrium uC/OS, and bare-metal
- Integrates new Freescale unified USB stack, open source FAT file system, encryption math/DSP libraries, and open source TCP/IP stack (lwIP).
- Support for multiple toolchains: IAR Embedded Workbench, Kinetis Design Studio, Keil uVision and Atollic TrueStudio.

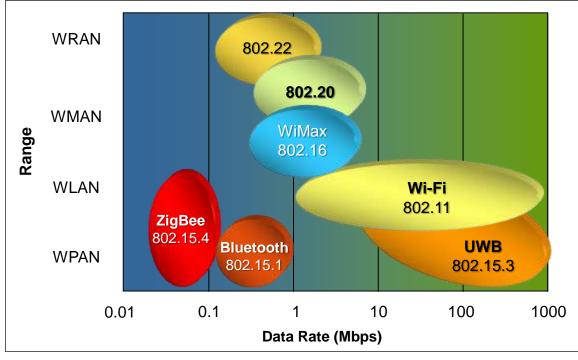
Learn more at: <u>www.freescale.com/KSDK</u>





### The IEEE® 802.15.4 Standard - Overview

- Stands out in the 802.x wireless space through: low power, low duty cycle, short frame format, low latency and generally low cost/complexity of implementation
- Allows self-forming, self-healing mesh networks
- Multiple RF bands: sub-GHz, 2.4 GHz
- Multiple modulation techniques: FSK, O-QPSK, OFDM

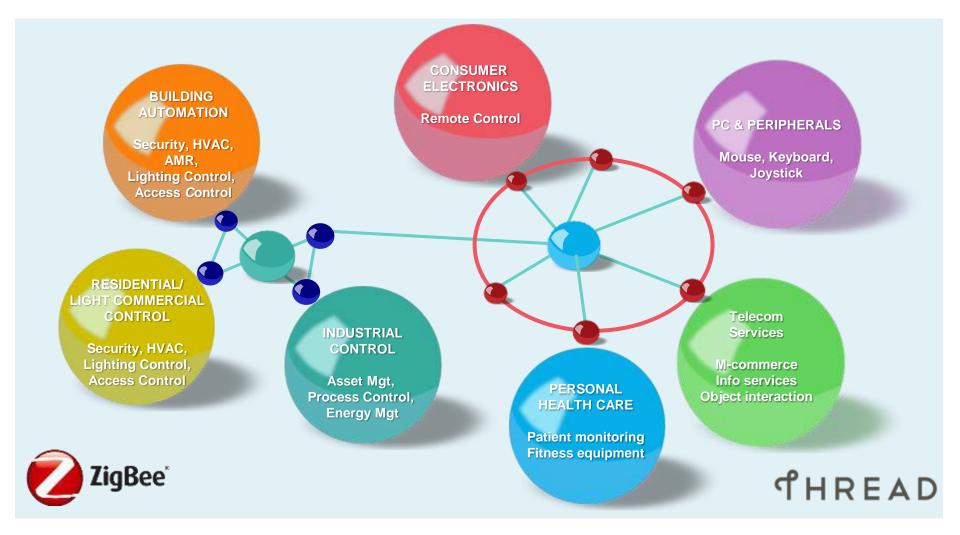


The IEEE® 802.x Wireless Space



2015

### **IEEE® 802.15.4-enabled Markets and Applications**





**#FTF2015** 

### The IEEE® 802.15.4 Standard – 2.4 GHz PHY Layer

- Low rate, low power, low cost O-QPSK DSSS modulation
- 16 Channels in the 2.4 GHz ISM band spaced at 5 MHz
- 250 kb/s (4 bits/symbol)
- 127 bytes PHY protocol data unit payload
- Tx power of at least 1 mW (0 dBm)
- Tx center frequency tolerance +/- 40 ppm
- Receiver sensitivity -85 dBm
- Two service access points: data (PD-DATA) and management (PLME)



### The IEEE® 802.15.4 PHY Protocol Data Unit

Sync Header		PHY Header		PHY Payload		
Preamble	Start of Packet Delimiter	cket Length (1 bit)		PHY Service Data Unit (PSDU)		
4 Octets	1 Octets	1 Octets		← 0-127 Bytes ─•		

- The 32-bit (8 O-QPSK symbols) all zeros preamble is used for synchronization
- "11100101" SFD indicates start of packet
- 7 out of the 8 PHY header bits are used to indicate the length of the PHY Service Data Unit (PSDU)
- The 2.4 GHz O-QPSK PSDU has a variable length between 0 and 127 bytes



## The IEEE® 802.15.4 Standard – MAC Layer

- Ensures reliable and secure data transfers
- Essential foundation for technologies like ZigBee® or Thread
- Collision avoidance algorithm through clear channel assessment
- Acknowledgement-based transmissions and re-transmissions
- Integrity checks with CRC-16
- AES-128 encryption and CCM\* block ciphers authentication of data
- Allows star or peer-to-peer topologies
- IEEE standard 64-bit or short, dynamic 16-bit addressing
- Dynamic device addressing allowing routed meshes in upper layers
- Optional slotted mode with superframe-based duty cycles
- Device segregation based on capabilities and roles in a network: coordinator and end device
- Exposes two service access points: management (MLME) and data (MCPS)



### **IEEE® 802.15.4 MAC Layer-enabled Topologies**

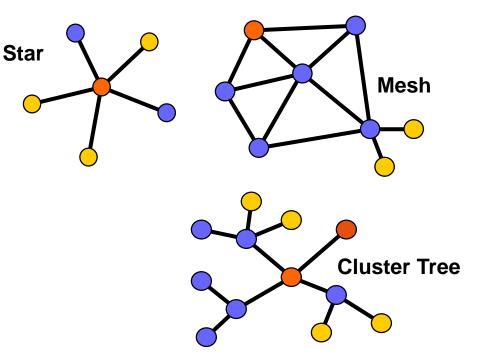
### **PAN Coordinator**

### **Full Function Device (FFD)**

- •Any topology •Network coordinator capable
- •Talks to any other device

### **Reduced Function Device (RFD)**

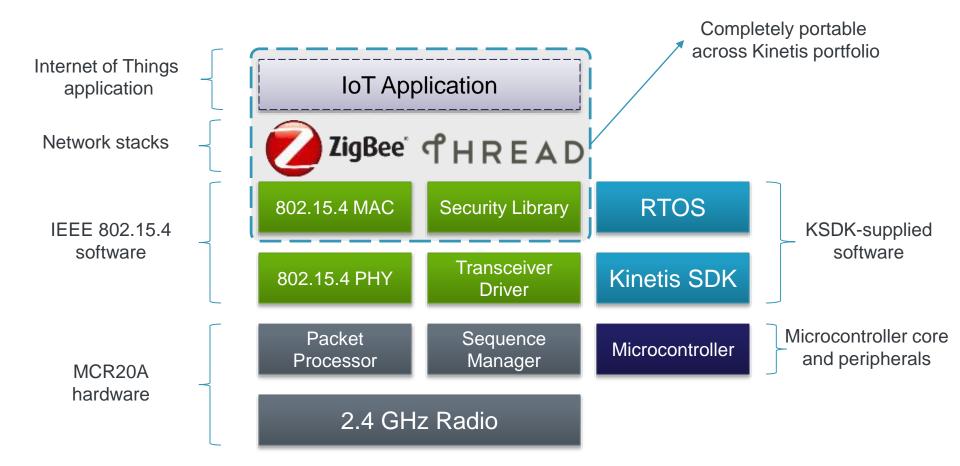
- •Limited to being leaf devices
- Cannot become a network coordinator
- •Talks only to a network coordinator
- •Very simple implementation





#FTF2015

### Freescale IEEE® 802.15.4 MCR20A System Architecture



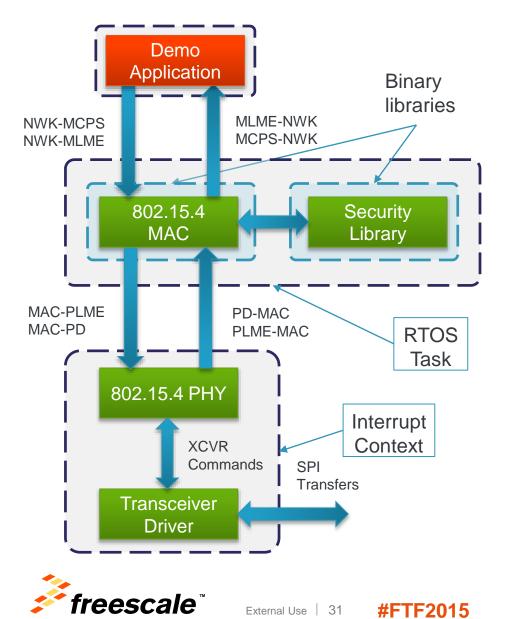


External Use 30 #FT

#FTF2015

 $\langle \rangle$ 

### Freescale IEEE® 802.15.4 Software for MCR20A



### **Features:**

- Add-on for the Kinetis SDK
- Uses RTOS services (either FreeRTOS or MQX)
- Needs two contexts: one RTOS task per MAC instance and the interrupt context.
- Contains 4 main modules: MAC library, Security library, PHY module and transceiver driver
- Transceiver driver communicates with any SPI driver of virtually any Kinetis microcontroller connected to the MCR20A.
- The PHY exposes standard service access points to the MAC (PLME and PD-DATA) and can operate in dual PAN mode.
- The MAC and security library are Freescale software IP and are distributed only in binary format.
- The MAC is instantiable and exposes for each instance the standard service access points (MLME and MCPS) for the network (NWK) stack(s) above, which can also be a simple application for demonstration purposes.
- This software is currently used by Freescale for as the basis for the ZigBee and the Thread stacks, which can even coexist on the same microcontroller thanks to the dual PAN feature.



# Porting the MCR20A 802.15.4 Software to FRDM-KL26Z

**#FTF2015** 



External Use 32



# Software Porting - Summary

This section is a hands-on presentation of how to port the MCR20A software to FRDM-KL26Z as an example of the ease of use of the KSDK-Connectivity ecosystem.



### Premises

Leveraging the Software and Hardware Ecosystem for Porting



### Setting-up the Software Environment

Kinetis SDK IAR Emebedded Workbench Connectivity Package

External Use 33



### Porting

Getting Familiar with a Pre-configured MCR20A Application Porting a MCR20A Application





### Premises – Why it is so Easy to Port

### • Kinetis SDK:

- Ensures availability of Kinetis microcontroller basic enablement: peripheral drivers, clock configurations and linker files.
- Provides readily configured RTOS libraries and abstraction layers for RTOS services which allow switching from one RTOS to another.
- Arduino standard for FRDM boards:
  - Allows the FRDM-CR20A shield to be paired with virtually any microcontroller board supporting the standard
- Tasks left for the user
  - Pair the FRDM-CR20A shield with the desired MCU motherboard
  - Use the software building blocks to design a connectivity application



## KSDK v1.1 and KSDK v1.1 SA for KL26Z Installation

- Navigate to <u>http://www.freescale.com/ksdk</u> and click "Download"
- Download Kinetis SDK v1.1.0
- Download KSDK v1.1.0 standalone release for the FRDM-KL26Z and MRB-KW01
- Default Install Paths: C:\Freescale\KSDK\_1.1.0 and C:\Freescale\KSDK1.1.0\_KL26Z\_KW01Z\_1.0.0

Image: Control of the second of t	the second s			H Gall	I sign in	Co Erigits		
Carlos Alberto's Treescale · · · Login Annetae instory My Recommendations My Favorites   Record · Annetae · Instory · My Recommendations · My Favorites Record · Annetae · Instory · Login Annetae · Instory · My Recommendations · My Favorites Record · Annetae · Instory · Instored	freescale	All ~	All - Search:					
And the second of the second for these and these has been block of the second of t	Products Applications Software & Tools Training &	communities Sam	ple & Buy	About				
INETIS_SDK: Software Development Kit for Kinetis MCUS ☆         Overview       Documentation       Downloads       Hardware & Tools       Training & Support         The Kinetis software development NI (SDK) is an extensive suite of robut peripheral drivers, stacks, middleware and example applications designed by software and board support configuration development on any Kinetis MCU. The additional designed by software soft example applications designed by and accelerate application development on any Kinetis MCU. The additional designed by and additional development on any Kinetis MCU. The additional development on any Kinetis SDK is complimentation and peripheral drivers, and baard support configuration for additional development on any Kinetis SDK is complimentation and peripheral drivers of ware.       Vaer Applications         Development Kit (SDK) Community Forum       Board Configuration development (kt (SDK)) Community Forum       Board Configuration development (kt (SDK)) Community Forum         Download       Board Configuration development (kt (SDK)) Community Forum       Board Configuration development (kt (SDK)) Community Forum	arlos Alberto's Freescale * 🕜 Login Annutate History My Reco	nmendations My Favor	ites	8	000	() Shar		
Overview         Documentation         Downloads         Hardware & Tools         Training & Support           The Kinetis software development NI (SDK) is an extensive pulse of robust eripheral drivers, stacks, middleware and example applications designed to simplify and accelerate application development on any Kinetis MCU. The addition provides unmatched ease of use and Brainliny. The Kinetis SDK is complimentary and includes full source code under a permissive open-source license for all hardware abstraction and peripheral driver software.         Vaer Applications         Board Configuration           The Kinetis SDK is offered to free, and support for it is provided through the Kinetis Stetware Development Kit (SDK) Community Forum         Board Configuration         Board Configuration         Board Configuration         Mathematication Layer         Board Configuration         Board Configuration         States and Middleware         Board Configuration         States and Middl				nabæmant				
The kinetis software development Nt (SDK) is an extensive suite of robust peripheral drivers, stacks, middleware and example applications designed to simplify and accelerate application development on any Kinetis SDK. Its complimentary and hourds support configuration provides unmatched ease of use and flexibility. The Kinetis SDK is complimentary and hourds that source code under a permissive open-source license for at hardware abstraction and peripheral driver software. The kinetis SDK is offered for free, and support for it is provided through the Kinetis Software Development Kt (SDK) Community Forum. Download	INETIS_SDK: Software Development Kit f	or Kinetis MCI	Js 🌣					
Peripheral drivers, stacks, middleware and example applications designed to simplify and accelerate application development on any Kinets MCU. The addition of Processor Expert technology for software and board support configuration provides unmatched ease of use and flexibility. The Kinetis SDK is complimentary and includes full source code under a permissive open-source license for all hardware abstraction and peripheral driver software. The Kinetis SDK is offered for file, and support for it is provided through the Kinetis Software Development K4 (SDK) Community Forum Download	Overview Documentation Downloads Hardware & Tools Tr	aining & Support						
And solution and peripheral driver solution and peripheral driver solution intersector as hardware abstraction and peripheral driver solution intersector and hardware abstraction and peripheral driver solution in a provided through the Kinstis Software Development Kit (SDK) Community Forum      Download      OMSIS Cove     Is SOC Header, IP Extension Reader Files     DSP	provides unmatched ease of use and flexibility. The Kinetis SDK is complimentary							
provides unmatched ease of use and flexibility. The Kinetis SDK is complimentary and includes full source code under a permissive open-source license for all hardware abstraction and peripheral driver software. The Kinetic SDK is offered for free, and support for it is provided through the Kinetis Software Development Kit (SDK) Community Forum Download OMSIS Core Header Files IP Extension Reader Files DSP	simplify and accelerate application development on any Kinetia MCU. The additio		User Applications					
The Kinetic SDK is offered for free, and support for it is provided through the Kinetic Software Development kit (SDK) Community Forum Download  CMSIS Core SOC Header, CMSIS DP Extension Reader Files DSP	and includes full source code under a permissive open-source license for all	4	Stacks and Middleware					
Download         Abstraction Layer         System Services           CMSIS Core         SOC Header, Header Files         CMSIS IP Extension Reader Files         CMSIS DSP	The Kinetis SDK is offered for free, and support for it is provided through the Kinet	5 (G 06A	-	Peripheral	Conguration	0		
OMSIS Core SOC Header, CMSIS Header Files IP Extension Header Files DSP	Download			Byū	tem Services			
Hardware	Download							
(increased)				Hardware				
		(manufacture)		Handware				



and a second state of the second state of the

#### **KSDK Installation Package**

• During installation, select all the RTOSes. This session focuses on FreeRTOS

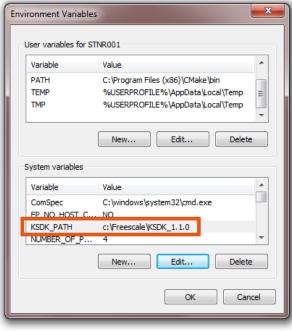




# **KSDK Environment Variable**

- The KSDK\_PATH system variable is used to point to the desired KSDK installation directory
- · Useful in the case when there are multiple versions of KSDK are installed
- · Batch files provided on your machine to easily modify this variable
- · This variable is used by connectivity projects to refer KSDK files
- Control Panel->System Properties->Advanced Tab
  - ->Environment Variables:

rstem Properties	Environment Va
Computer Name         Hardware         Advanced         System Protection         Remote           You must be logged on as an Administrator to make most of these changes.         Image: Computer System Protection         Image: Computer System Protection	User variable Variable
Performance Visual effects, processor scheduling, memory usage, and virtual memory Settings	PATH TEMP TMP
User Profiles Desktop settings related to your logon Settings	System varial Variable
Startup and Recovery System startup, system failure, and debugging information Settings	ComSpec FP_NO_HO KSDK_PATH NUMBER_O
Environment Variables OK Cancel Apply	





External Use | 37 #FT

# **Environment Set-up – IDE/Toolchain**

#### IAR Embedded Workbench for ARM

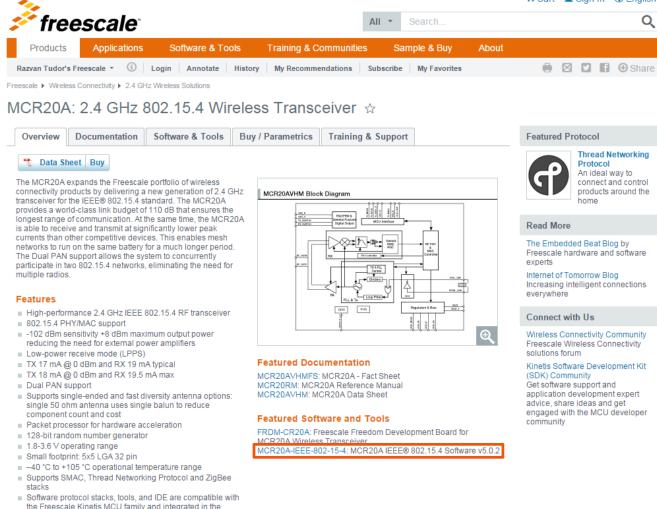
- The connectivity applications and applications are currently limited to the IAR Embedded Workbench for ARM.
- Your machine comes with version 7.40.1 of this IDE/Toolchain preinstalled.
- Check your Windows Start Menu for launching it.
- For more information about this development environment, please visit <u>www.iar.com/ewarm</u>





#### MCR20A Connectivity Software Package Installation

 MCR20A webpage and software download link: <u>www.freescale.com/MCR20A</u> Cart 👤 Sign In 🛞 Englis



Kinetis software development kit (SDK)

External Use



39



## **MCR20A Connectivity Software Package Installation**

Gnetis SDK Configuration Validate the Kinetis SDK 1.1.0 Main	line folder.
Choose the folder where KSDK 1.	. 1.0 was previously installed
	software package for MCR20A setup will make several
additions to the Kinetis Softwar	software package for MCR20A setup will make several re Development Kit (KSDK) v1.1.0 Mainline installation modified, nor will any KSDK applications be affected. Click

hoose Path	
Choose IEEE 802.15.4 Software Folder	\$
Choose the folder where the IEEE 802, 15.4 software for M	ICR20A will be installed
C:\Freescale\MCR20A_IEEE_802_15_4	Browse
< Back	Next > Cancel

External Use

- MCR20A webpage and download link: <u>www.freescale.com/MCR20A</u>
- The package requires KSDK to be previously installed
- Make sure to select the correct KSDK 1.1.0 installation folder when prompted
- Select the default installation folder for the MCR20A software package as: C:\Freescale\MCR20A\_IEEE\_802\_15\_4



# Getting Familiar with a Preconfigured MCR20A Application

FRDM-K64F "MyWirelessApp" Coordinator









#### **Opening a K64F/MCR20A Application (MyWirelessApp)**

 Open the K64F MyWirelessApp Coordinator IAR workspace from C:\Freescale\MCR20A\_IEEE\_802\_15\_4\app\ieee\_802\_15\_4\MyWirelessApp\Coor dinator\frdm-k64f\_frdm-mcr20a\ by double-clicking on the .eww file.

Winnerstapp Coo	rain	ator 🖡 frdm-k641_frdm-mcr20a 🖡	• 49 Se	anth Belon klift fritenn	nikka
Organize 🔹 🎉 Open 🔹 Burn		New folder		10 A	• 🖽 6
🍌 Demo	*	Name	Date modified	Тури	Size
di Drivers		J. Sources	4/6/2015 3:49 PM	File folder	
Freescale	1	MyWirelessAppCoordinator.ewd	2/12/2015 1:07 PM	EWD File	145 KB
KSOK_1.1.0		MyWirelessAppCoordinator.evp	2/24/2015 2:03 PM	EWP File	141.KB
KSDK1.1.0_KL26Z_KW01Z_1.0.0 MCR20A_JEEE_802_15_4		MyWirelessAppCoordinator.eww	4/3/2015 2:32 PM	<b>IAR IDE Workspace</b>	1 KB
🌲 Intel					

#FTF2015



External Use | 42

# Configuring K64F/MCR20A Application (MyWirelessApp)

- Select the frdm-k64f\_frdm-MCR20 FreeRTOS build configuration for the MyWirelessAppCoordinator project from the top drop-down box.
- Navigate to App.h, line 52 to configure the radio channel bit mask (bits 11 26).

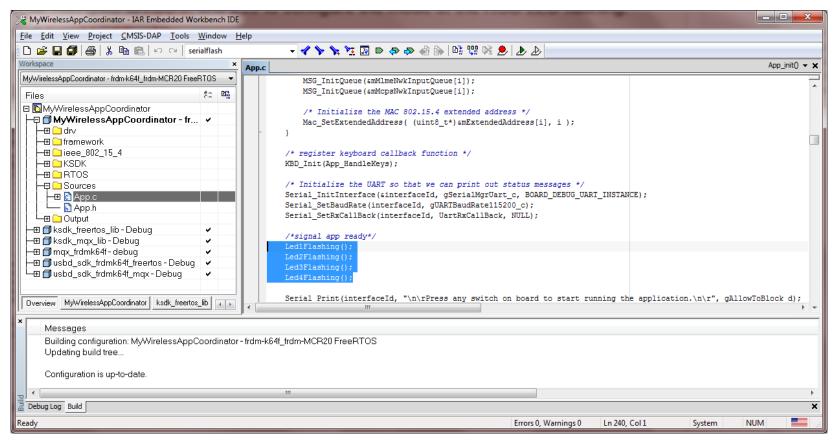
😹 MyWirelessAppCoordinator - IAR Embedded Workbench IDE
<u>File Edit View Project CMSIS-DAP Tools Window H</u> elp
🗅 🖙 🖬 🕼 🐇 ங 🋍 🗠 🜼 Serial_Print 💿 🗸 🍾 🦎 📆 🖻 🧇 🛷 🖑 🕼 🛤 🐯 😣 🌛 🕭
Workspace × App.h f0 • ×
MyWirelessAppCoordinator · frdm+k64f_frdm+MCR20 Free
Files       *         B       MyWirelessAppC         H       MyWirelessA         H       MyKotk         H       My
Dverview MyWirelessAppCoordinator ksdk_free
Log Mon May 18, 2015 17:02:37: Loading the Hjet/JTAGjet driver           Image: Constraint of the state of the stat
Debug Log     Build   Find in Files
Ready Ln 52, Col 39 Syster



FTF2015

# **Configuring K64F/MCR20A Application (MyWirelessApp)**

- Optional: Navigate to App.c, line 239 to configure the mode of the RGB LED flashing:
  - Default: \*very\* brightly white (some people might find it bothersome), or
  - Replace the 4 shown lines with LED\_StartSerialFlash; for RGB flashing.





# Configuring K64F/MCR20A Application (MyWirelessApp)

- Navigate to App.c, line 239 to configure the transmit power:
  - Default: 0 dBm (1milliwatt) relatively high power, might interfere with neighbors on the same channel
  - Add the following line above the LED function calls: Asp\_SetPowerLevel(0x0D); for -20 dBm Tx power

rispace	×	il osa ext. freertos.c. App.c. App.h   PhyPadietProcessor.c.   PhyPimeData.c.   ASP.c. App.jnttD +
In +64_Indm MCR29 FreeRT0S	10	/* Prepare input queues.*/
iles  MyWirelessAppCoordinator - frdi  div  fiee_802_15_4  div  ieee_802_15_4  div  fieee_802_15_4  div  power of the second of	822 *	<pre>MSG_InitQueus(addimeNvkInputQueus[1]); MSG_InitQueue(addimeNvkInputQueus[1]); /* Initialize the MAC 802.15.4 extended address */ Mac_SetExtendedAddress( uint5_t*)amExtendedAddress[1], 1 ); /* register keyboard callback function */ MSD_Init(App_HandleHeys); /* Initialize the UART so that we can print out status messages */ Serial_InitInterface(sinterfaceId, gSerialMyrUart_c, BOARD_DEBUG_UART_INSTANCE); Serial_SetExadRate(interfaceId, gSerialMyrUart_c, BOARD_DEBUG_UART_INSTANCE); Serial_SetExadRate(interfaceId, GEARDADAStellS200_c1); Serial_SetExadRate(interfaceId, UartExaCallBack, MULL); /*signal app ready*/ LedIFlashing(); LedJFlashing(); LedJFlashing(); Serial_Frint(InterfaceId, "\n\rFress any switch on board to start running the app } </pre>
Messages		



#### **Building a Pre-configured K64F Application (MyWirelessApp)**

- Make sure to first build (right click->"Make") the two KSDK RTOS libraries required by the application: ksdk\_freertos\_lib and usb\_sdk\_frdmk64f\_freertos.
- Build the application (right click->"Make").

K MyWirelessAppCoordinator - IAR Embedded Workbench IDE	
<u>Eile Edit View Project CMSIS-DAP Tools Window H</u> elp	
□ 🗳 🖬 🗿 🍜 🗼 階 💼 🗠 🖙 🗸 🖤 🐨 🗸 🗸	> > > > > > > > > > >
MyWirelessAppCoordinator - frdm-k64f_frdm-MCR20 FreeRT0S	× · · · ·
□ ⓑ MyWirelessAppCoordinator □ ☐ MyWirelessAppCoordinator - frdm-k64f_frdm-MCR20 FreeR.	
	v
H → → ksdk_mqx_lib - Debug	
- 🖽 🗇 max_frdmk64f-debug	
usbd_sdk_trdmk64t_mqx - Debug	· · · · · · · · · · · · · · · · · · ·
Overview MyWirelessAppCoordinator ksdk_freertos_lib ksdk_mqx_lib mqx_frdmk64f	usbd_
x	
Mon May 04, 2015 10:00:31: Loading the Hjet/JTAGjet driver	
8	
	- F
Ready	



#### Flashing a Pre-configured K64F Application (MyWirelessApp)

- Make sure you plug in the FRDM-K64F + FRDM-CR20A (referred to as coordinator) setup to your computer via the micro USB port on the K64F motherboard, labelled "SDAUSB".
- Make sure that the mBed CMSIS-DAP firmware derivers (pre-installed on your computer) have been successfully enabled by Windows by checking Device Manager
- Make sure to select CMSIS-DAP as the debugger driver and SWD as the interface in the project options (right-click->"Options").
- After the debugger settings are done and the application is built, click the green arrow labelled "**Download and Debug**" and wait about 10 seconds for the flash download.

- Devce Manager	Options for nade "MyN	VinimiAppCoordinator	Options for node "MyWi	relessAppCoordinat	*)
	Calegory General Options Rundims Checking CrC++ Complet Assanthier Output Connector Custon Build Build Actions Linker Coloropat Simulator Angel CrSLSI DWP GDB Server JAN, RCM-vecetor T-jett(JTArlijet J-Link)-Trace T-Stalliers		Category: Seneral Options Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Smiduter Angel Checurote GOS Server LAR SCM-monitor Linker/DiffActions	Setup JTAG/SW Pobe config @ Auto © From tile © Explicit Interface ©TAG @	Factory Settings
Intel(R) 8 Sameu/CL20 Series US8 EHC1 +2 - SC2D     Intel(R) US8 3.0 eXtensible Most Controller     Intel(R) US8 3.0 Root Hub     Hond RU US8 3.0 Root Hub     Hond Ruber     mbed Composite Device	Nex steps PE micro RDI ST-LIM Therei-Party Driver XDS 180/200,0000	BTOOLKIT_DIPS/CONPS/ debugger/Feesode MIKEBFIITIND	T3 Stellern Macraiger PE more BDI ST-CINK Third Party Driver X05 100/200,0CD1	JTAG-SWD speed	Children and Angel





#### **Debugging a Pre-configured K64F Application (MyWirelessApp)**

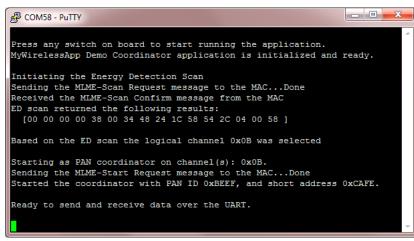
- If everything went well, you should see the debugger window and the application stopped in the main() function.
- Click "Go" then "Stop debugging". The Board LED should be flashing either bright white or RGB, depending on how it was configured in the previous step.

MyWorkstAppContinues - IAR Embedded Workbergh IDE		California
Die Enkt View Broject Debug Disassembly CMIIS-DAP I		
	・ イトアガロマキキ 白田 目間の ● 命会	
5- • 8228835 × < 🗔		
	A Intrast set Sectors	· * Description ·
My/wieksuAppCoordinatur Indink64 to day CR20 FreeRT05		- Gate - Ma
Files	F m E/m	Disessembly *
Ties		0x9306 0x1000
MyWirelessAppCoordinator - trdm-kE4t_trdm	Private functions	A 444 - A 1 444
Har a head to be		
-fil 🗊 kadk_max_lib - Debug		0x930c 0x2003
- 💷 🗊 mgx_hūmkii-ti - debug	✓ int min (vold)	lbde0 s069a0
- I gusbd_sdk_trdmk64t_treentos - Debug	· .	LiepeToHaxt - Liepe
	<ul> <li>Lasi_hanfler_1 handler:</li> </ul>	0x9310 0x0221
	ona Inin O r	0x9312 0x9806
		0x9314 0x9000
	CSA_TabWCzestm)(task_t)startup_task,	0x9316 0x0=0*
	"aain",	0x9318 0x003;
	gMainThreadStackSize_c, HULL,	IISdaD GIFERD
	gMainThreadPriority_c,	6x933c 6x2203
	5. · · · · · · · · · · · · · · · · · · ·	0x931m 0x4633 0x9320 0x1420
	TALIE,	0x9322 0x17fc
	shandler;;	0x9326 0xbdf2
	OSA Start()	Ant Assa [Wood]
		1.6
	retara Gr	RAIN:
	- b 1	uw932p_ 0wb50(
		0x932a 0xb08!
Eveniew M/WielesulppCoordinates Lock Neeton (b) Lock right	mprata a management	
[ ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ] ]		
Messages	Last Laster Brok	
Building configuration: MyWirelessAppCoordinator-http: Updating build hee	n+s+4_nom-MCH28 FreeH105	
Chonesi francia de		
Configuration is up-to-date.		
answedents weren en en fal		
		4
Orthug Log Duty		×
estr	Errors 0, Warrings 0	NUM
104.		I HAVE BOOM



### Running the K64F MyWirelessApp Coordinator Demo

ategory:	Basic options for your PuTTY session	
Logging ⊡- Terminal Keyboard Bell	Specify the destination you want to connect to Serial line Speed COM58 11520	0
- Features - Window - Appearance - Behaviour - Translation - Selection - Colours - Connection - Data - Proxy - Telnet - Rlogin RD-SSH	Connection type: Raw Delnet Rogin SSH Load, save or delete a stored session Saved Sessions	ierial
	Default Settings	<u>/</u> e
Serial	Close window on exit: Aways   Never  Only on clean exit	



External Use 49

- Open a PuTTY terminal program (shortcut should be available on the desktop of your machine)
- Configure the same serial port shown in Device Manager as "mBed Serial Port" with 115200 baud.
- Connect or reset the coordinator. The message to press any key should appear.
- Press either SW1 or SW2 on the FRDM-CR20A.
- The RGB LED should stop flashing.
- The coordinator should start on the least occupied channel in the mask and the serial terminal should show the progress.



# Porting a MCR20A Application

"MyWirelessApp" End Device from KL46Z to KL26Z









# Porting a MCR20A Connectivity Application

- Steps:
  - Set as baseline the FRDM-KL46Z + FRDM-CR20A MyWirelessApp End Device application
  - Identify the FRDM-KL46Z board and KL46Z processor specific components.
  - Create new application workspace and replace the board/processor specific components with FRDM-KL26 and KL26Z ones.
  - Build and run the FRDM-KL26Z + FRDM-CR20A application.
  - Associate with the FRDM-K64F + FRDM-CR20A previously configured 802.15.4 coordinator.



# **KL46Z/MCR20A MyWirelessApp End Device Application**

Navigate to

C:\Freescale\MCR20A\_IEEE\_802\_15\_4\app\ieee\_802\_15\_4\MyWirelessApp\EndDevice\frd m-kl46z\_frdm-mcr20a and double click on the .eww file.

• Select the frdm-kl46z\_frdm-mcr20a KSDK FreeRTOS build configuration.

K MyWirelessAppEndDevice - IAR Embedded Workbench IDE		_ 🗆 🗙	
<u>File Edit View Project CMSIS-DAP Tools Window Help</u>			
- C 😂 🖬 🕼 👃 🌡 🏷 🗠 🖂 🗠 🖂 🖌 😽 🔁 🖾	-	D 🐢 🐢 🍪 (	
Workspace		× v	×
MyWirelessAppEndDevice - frdm-kl46z_frdm-mcr20a KSDK FreeRT0S		-	
Files	23	₿ <b>₽</b>	
🖻 🖸 MyWirelessAppEndDevice			
H → I MyWirelessAppEndDevice - frdm-kl46z_frdm-mcr20a KSDK FreeR			
│	Ĵ		
Hard Start	Ĵ,		
usbd_sdk_frdmkl46z_mqx - Debug	~		
└─⊞ 🗇 usbd_sdk_frdmkl46z_freertos - Debug	~		
Overview MyWirelessAppEndDevice mqx_frdmkI46z ksdk_mqx_lib ksdk_freertos_lib usbd_sdk_	frdmk	4 >	
Log Mon May 04, 2015 17:25:23: Loading the I-jet/JTAGjet driver			•
		4	Ŧ
Ready			đ



# **Board Configurations (Existing)**

Organize	»	<ul> <li>✓ 49 Search frdmkl4</li> <li>III ▼ □</li> </ul>
<ul> <li>Freescale</li> <li>KSDK_1.1.0</li> <li>Freescale Kinetis SDK v1.1.1</li> <li>boards</li> <li>common</li> <li>frdmk22ft</li> <li>frdmk22fk02</li> <li>frdmk22fk0264</li> <li>frdmk64f</li> <li>frdmk64f</li> <li>frdmk64f</li> <li>frdmk103z</li> <li>frdmk16z</li> </ul>	D_ii E	Name board.h gpio_pins.c gpio_pins.h hardware_init.c pin_mux.c pin_mux.h
🔒 frdmkl46z_frdmmcr20a	-	* <u> </u>

- FRDM-KL46Z and FRDM-K64F board configurations are duplicated.
- This avoids overwriting existing KSDK default configurations.
- Ensures proper configuration for operation with FRDM-CR20A.
- The files in the frdmkl46z\_frdmmcr20a and frdmk64f\_frdm-mcr20a folders contain the specific board configurations done for FRDM-CR20A: SPI, push-buttons and RGB LED



# Platform Specific – Board Configurations (FRDM-KL26Z)

🖌 🖉 📕 « boards 🕨 frdmkl2 🔹	49	Search frdmkl26z_fr
Organize Vinclude in library VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV	* E	Name board.h gpio_pins.c gpio_pins.h hardware_init.c pin_mux.c pin_mux.h
<ul> <li>activity</li> <li>doc</li> <li>filesystem</li> <li>files</li> <li>lib</li> <li>platform</li> <li>rtos</li> </ul>	Ŧ	< <u>III</u>

- Need to create FRDM-KL26Z board configuration in the KSDK v1.1 SA for KL26Z folder structure that will allow FRDM-CR20A compatibility and not overwrite the default KSDK settings
- A copy-paste of the existing frdmkl26z configuration is a good starting point
- For simplicity, this has already been done on your machine, in the form of the frdmkl26z\_frdm-mcr20a folder.
- The files in the above folder are the most important board configuration elements which need to be modified for FRDM-CR20A proper operation.

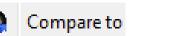


# **Configuring the Board – Compare the Sources**

- FRDM-KL46Z Generic Sources:
  - C:\Freescale\KSDK 1.1.0\boards\frdmkl46z\
- FRDM-KL46Z Specific Sources for FRDM-CR20A operation:
  - C:\Freescale\KSDK\_1.1.0\boards\frdmkl46z\_frdmmcr20a\
- FRDM-K26Z Specific Sources:
  - C:\Freescale\KSDK1.1.0\_KL26Z\_KW01Z\_1.0.0\boards\frdmkl26z\
- FRDM-KL26Z Specific Sources for FRDM-CR20A operation:
  - C:\Freescale\KSDK1.1.0\_KL26Z\_KW01Z\_1.0.0\boards\frdmkl26z\_frdm-mcr20a\
- Tool used: Beyond Compare from Scooter Software
  - Right click on first file and choose Select Left File for Compare
  - Right click on first file and choose 🧔 Compare to

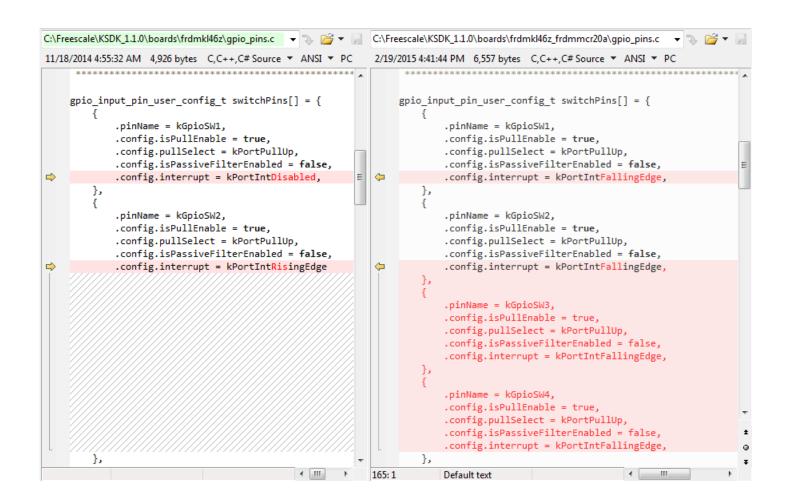
External Use 55





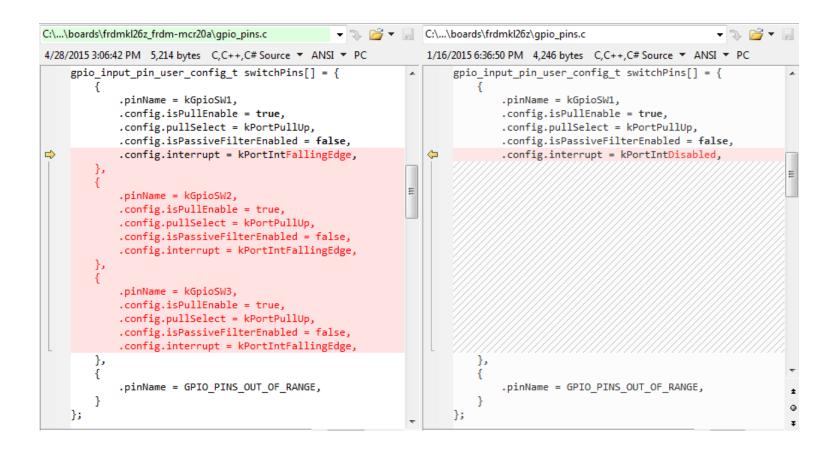


#### **GPIO Pins – FRDM-KL46Z Switches Configuration**





#### **GPIO Pins – FRDM-KL26Z Switches Configurations**





 $\langle \rangle$ 

# GPIO Pins – FRDM-KL46Z & FRDM-CR20A LED & Transceiver Pins Declaration

C:\Freeso	ale/KSDK_1.1.0\boards/JrdmkH6z_frdmmcs20a/gpio_pins.h 🔹 🐄 💆 •	51	C:\F	\Freescale\XSDK_1.1.0\boardx\/rdmkH6z\gpio_pim.h 🔹 🖓 🥥 🔹 🚽
2/19/201	543826 PM 4,179 bytes C,C++,C# Source * ANSI * PC		11/1	1/18/2014 4:55:32 AM 3:538 bytes C,C++,C# Source * ANSI * PC
	I This should be defined according to board setting."/ um_gpio_pins	1		/*1 This should be defined according to board setting.*/
•	k6picLED1 = 6PIO_NAKE_PIN(HW_GPIOE, 20), /* FROM-KL4624 Red LED */ k6picLED2 = 6PIO_NAKE_PIN(HW_GPIOE, 3), /* WCR26A_Blue ROB_Led*/ k6picLED3 = 6PIO_NAKE_PIN(HW_GPIOC, 2), /* WCR26A_Green_RGB_Led*/		0	<pre>b k0pioLED1 = GPIO_MAKE_PIN(HW_GPIOD, 5), /* FRDM-KL46I4 Green LED */ k0pioLED2 = GPIO_MAKE_PIN(HW_GPIOE, 29), /* FRDM-KL46I4 Red LED */</pre>
4	<pre>kdpioL6D4 - GPI0_MAKE_PIN(He_SPI0C, 1), /* MCR20A_RED_R0H_Led*/ kdpioSpi5ckPin = GPI0_MAKE_PIN(He_SPI0C, 3), /* MCR20A_RED_R0H_Led*/ kdpioAccelINT1 = GPI0_MAKE_PIN(He_GPI0C, 3), /* FRDM-KL4624_FVMADA5L0/FXD587000CE_INT1 */ kdpioIAccelINT2 = GPI0_MAKE_PIN(He_GPI0C, 1), /* FRDM-KL4624_MMADA5L0/FXD587000CE_INT2 */ kdpioIAccedINT = GPI0_MAKE_PIN(He_GPI0C, 24), /* FRDM-KL4624_IN2 address_pin */ kdpioIaCceddr2 = GPI0_MAKE_PIN(He_GPI0C, 25), /* FRDM-KL4624_I2C address_pin */ kdpioIaccedIXT = GPI0_MAKE_PIN(He_GPI0C, 25), /* FRDM-KL4624_I2C address_pin */ kdpioIacrDemoXX = GPI0_MAKE_PIN(He_GPI0A, 1), /* FRDM-KL4624_I2C address_pin */ kdpioIacrDemoXX = GPI0_MAKE_PIN(He_GPI0A, 1), /* FRDM-KL4624_I2C address_pin */ kdpioIacrDemoXX = GPI0_MAKE_PIN(He_GPI0A, 1), /* FRDM-KL4624_I2C address_pin */ kdpioSM1 = GPI0_MAKE_PIN(He_GPI0A, 12), /* MCR20e_switchPin1 */ kdpioSM2 = GPI0_MAKE_PIN(He_GPI0A, 12), /* MCR20e_switchPin1 */ kdpioSM2 = GPI0_MAKE_PIN(He_GPI0A, 400, /* MCR20e_switchPin1 */ kdpioSM2 = GPI0_MAKE_PIN(He_GPI0</pre>		e.	<pre>k6pioAccelINT1 = GPIO_PAKE_PIN(Hei_GPIOC, 5), k6pioAccelINT2 = GPIO_PAKE_PIN(Hei_GPIOC, 5), k6pioI2Ceddr1 = GPIO_PAKE_PIN(Hei_GPIOC, 24), k6pioI2Ceddr2 = GPIO_PAKE_PIN(Hei_GPIOE, 25), k6pioUartDemoRX = GPIO_PAKE_PIN(Hei_GPIOE, 25), k6pioUartDemoRX = GPIO_PAKE_PIN(Hei_GPIOE, 1), k6pioUartDemoRX = GPIO_PAKE_PIN(Hei_GPIOE, 2), k6pioUartDemoRX = GPIO_</pre>
	<pre>k6pioSW3 = GPI0_NAKE_PIN(Hw_GPIOC, 120), /* FRDM-KL46Z4 switchPin1 */ k6pioSW4 = GPI0_NAKE_PIN(Hw_GPIOC, 30), /* FRDM-KL46Z4 switchPin2 */ /* ConnSw */ k6pioXcv=SpiCuPin = GPID_NAKE_PIN(Hw_GPIOD, 4u), kipioXcv=SpiCuPin = OPID_NAKE_PIN(Hw_GPIOD, 4u), kEpioXcv=IrgPin = GPID_NAKE_PIN(Hw_GPIOD, 3u),</pre>			<pre>kSpidSW1 = GPIO_MAKE_PIN(HW_GPIOC, 3), /* FRDM-KL4624 switchPin1 */ kSpidSW2 = GPIO_MAKE_PIN(HW_GPIOC, 120), /* FRDM-KL4624 switchPin2 */</pre>
33				h
52:41	Identifier *	-		



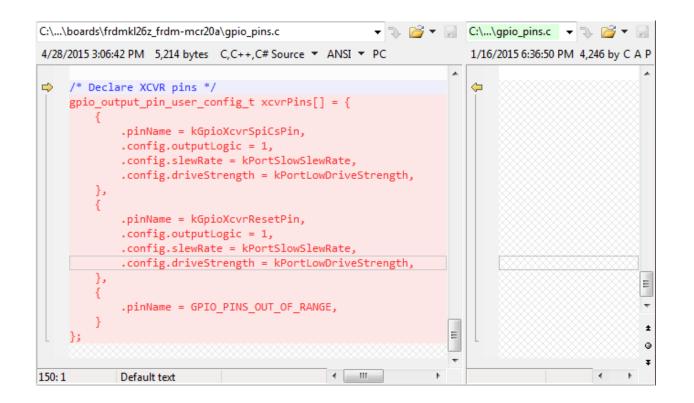
#### **GPIO Pins – FRDM-KL26Z & FRDM-CR20A LED & Transceiver Pins Declaration**

C:\Freesci	ele/KSDK3.1.0_K126Z_KW01Z_1.0.0\boardri\frdmk26z_frdm-mcr20e\gpix_pins.h 🔹 🦷 💆 🔹 .	C	:/Freescale/KSDK1.1.0_KL26Z_KW01Z_1.0.0/boards//rdmkl26z/gpio_pins.h 🔹 🐄 🥥 🔹 🚽
4/28/2019	54:15:32 PM 3,694 bytes C,C++,C# Source = ANSI = PC	1/	/16/2015 6:36:50 PM 3,316 bytes C,C++,C# Source + ANS + PC
177	Definitions / gbrief gpic pin nomes.*/ */ ! This should be defined according to board setting.*/ m_gpic_pins	*	* Definitions
0	kGpioLED1       - GPIO_MAKE_PIN(HW_GPIOC, lu), /* FRDM-MCR20A Red RGB Led */         kGpioLED2       - GPIO_MAKE_PIN(HW_GPIOC, 2u), /* FRDM-MCR20A Green RGB Led */         kGpioLED3       - GPIO_MAKE_PIN(HW_GPIOC, 2u), /* FRDM-MCR20A Green RGB Led */         kGpioLED3       - GPIO_MAKE_PIN(HW_GPIOC, 2u), /* FRDM-MCR20A Green RGB Led */         kGpioAccelINT1       - GPIO_MAKE_PIN(HW_GPIOC, 0u), /* FRDM-MCR20A Glue RGB Led */         kGpioAccelINT1       - GPIO_MAKE_PIN(HW_GPIOC, 0u), /* FRDM-KL2624 HWARASIQ/FXDSR2000CB INT2 */         kGpioUartDewoTX       = GPIO_MAKE_PIN(HW_GPIOA, 2u), /* FRDM-KL2624 UART 0 TX pin (OpenSDA port) *         kGpioUartDewoRX       - GPIO_MAKE_PIN(HW_GPIOA, 2u), /* FRDM-KL2624 UART 0 TX pin (OpenSDA port) *         kGpioSMI       - OPIO_MAKE_PIN(HW_GPIOA, 1u), /* FRDM-KL2624 UART 0 TX pin (OpenSDA port) *         kGpioSMI       - OPIO_MAKE_PIN(HW_GPIOA, 1u), /* FRDM-HL2624 UART 0 TX pin (OpenSDA port) *	0	<pre>k6pioLED1 - GPIO_MAKE_PIN(Mi_GPIOE, 31u), /* FROM-x12624 Green LED */ k0pioLED2 - OPIO_MAKE_PIN(Mi_GPIOE, 29u), /* FROM-x12624 Red LED */ k0pioLED3 - GPIO_MAKE_PIN(Mi_GPIOD, 50), /* FROM-x12624 PNA64510/PXOSE7000CE INT1 */ k0pioAccelINT2 - GPIO_MAKE_PIN(Mi_GPIOD, 00), /* FROM-x12624 PNA64510/PXOSE7000CE INT1 */ k0pioUsertDemoTX = GPIO_MAKE_PIN(Mi_GPIOA, 20), /* FROM-x12624 UABT 0 TX pin (OpenSDA port) = k0pioUsertDemoTX = GPIO_MAKE_PIN(Mi_GPIOA, 10), /* FROM-x12624 UABT 0 TX pin (OpenSDA port) = k0pioUsertDemoTX = GPIO_MAKE_PIN(Mi_GPIOA, 10), /* FROM-x12624 UABT 0 TX pin (OpenSDA port) = k0pioUsertDemoTX = GPIO_MAKE_PIN(Mi_GPIOA, 10), /* FROM-x12624 UABT 0 TX pin (OpenSDA port)</pre>
Ъ	KGpioSk3       + GPI0_MAKE_PIN(HW_GPIOC, 3u), /* FRDM-KL2624 switchPin1 */         /* ConnSw */          kGpioKcvrSpitsPin *       GPI0_MAKE_PIN(HW_GPIO0, 4u),         kGpioKcvrResetPin *       GPI0_MAKE_PIN(HW_GPIOA, 5u),         kGpioKcvrResetPin *       GPI0_MAKE_PIN(HW_GPIOA, 5u),         kGpioKcvrIngPin *       GPI0_MAKE_PIN(HW_GPIOA, 5u),		<pre>k6piaSki - GPI0_MAKE_PIN(Hk_6PIOC, 3u), /* FRDM-KL26Z4 switchPin1 */ };</pre>
50: 80	Comment *	50	1:80 Comment • It It It



 $\langle \rangle$ 

#### **Transceiver Pins Configuration (Both Setups)**





 $\langle \rangle$ 

# **Platform Specific – Linker File**

External Use | 61

- Connectivity applications use dedicated linker files located in the platform\linker folder in the KSDK installation.
- For porting a MCR20A application from KL46Z to KL26Z, only the **memory map** needs to be re-defined in the new connectivity KL26Z linker file, starting from the KL46Z one.

:\Freescale\KSDK_1.1.0\platform\linker\MKL26Z4\iar\MKL26Z128xxx4_connectivity.icf 🔹 🔹 📄	C:\Freescale\KSDK_1.1.0\platform\linker\MKL46Z4\iar\MKL46Z256xxx4_connectivity.icf 🔹 🔹 😂 🗲
/29/2015 10:39:52 AM 9,219 bytes <default> ▼ ANSI ▼ PC</default>	2/4/2015 4:12:56 PM 9,010 bytes <default> ▼ ANSI ▼ PC</default>
<pre>/* By default, the Bootloader is not used. */ if (!isdefinedsymbol(gUseBootloaderLink_d)) {    define symbol gUseBootloaderLink_d = 0; }</pre>	<pre>/* By default, the Bootloader is not used. */ if (!isdefinedsymbol(gUseBootloaderLink_d)) {    define symbol gUseBootloaderLink_d = 0; }</pre>
<pre>/* By default, the NVM is not used. */ if (lisdefinedsymbol(gUseNVMLink_d)) {     define symbol gUseNVMLink_d = 0; }</pre>	<pre>/* By default, the NVM is not used. */ if (!isdefinedsymbol(gUseNVMLink_d)) {     define symbol gUseNVMLink_d = 0; }</pre>
<pre>/* By default, the internal storage is not used. */ if (!isdefinedsymbol(gUseInternalStorageLink_d)) {     define symbol gUseInternalStorageLink_d = 0; }</pre>	<pre>/* By default, the internal storage is not used. */ if (!isdefinedsymbol(gUseInternalStorageLink_d)) {    define symbol gUseInternalStorageLink_d = 0; }</pre>
/*-Memory Regions-*/ define symbol region ROM start = 0x00000000;	/*-Memory Regions-*/ define symbol region ROM start = 0x00000000;
<pre>&gt; define symbolregion_ROM_end = 0x0001FFFF; define symbolregion_RAM_start = 0x1FFFF000; define symbolregion_RAM_end = 0x20002FFF;</pre>	<pre>define symbolregion_ROM_end = 0x0003FFFF; define symbolregion_RAM_start = 0x1FFFE000; define symbolregion_RAM_end = 0x20005FFF;</pre>
<pre>define symbolvector_table_size = 192; define symbolram_vector_table_size = isdefinedsymbol(ram_vector_table) ?v</pre>	<pre>define symbolvector_table_size = 192; define symbolram_vector_table_size = isdefinedsymbol(ram_vector_table) ?</pre>
if (gUseBootloaderLink_d)	if (gUseBootloaderLink_d)
<pre>i define symbol m_bootloader_start =region_ROM_start; define symbol m_bootloader_end = ((region_ROM_end + 1) / 32) - 1;</pre>	<pre>i define symbol m_bootloader_start =region_ROM_start; define symbol m_bootloader_end = ((region_ROM_end + 1) / 32) - 1;</pre>
<pre>define symbol m_interrupts_start = m_bootloader_end + 1; 1</pre>	<pre>define symbol m_interrupts_start = m_bootloader_end + 1; 4.1 Default text &lt; """</pre>
):1 Default text	i4:1 Default text



#### Creating the IAR EWARM Project for the "MyWirelessApp End Device" Application for FRDM-KL26Z

🗆 🖸 MyWirelessAppEndDevice		
	<b>~</b>	
		*
		*
		*
Here Here Manager		*
Hessaging		*
		*
		*
⊢⊕ Danic		*
		*
- E CaralManager		*
Hend Contraction C		*
		*
		*
		*
gpio_pins.c		*
E gpio_pins.h		
Handware_init.c		*
⊨⊞ D pin_mux.c		-
⊢□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □		
		-
		*
App.h		-
—⊞ 🗖 mqx_frdmkl26z - debug *	~	
$= \exists \prod mqx_n dmki2s2 - debug$	ž	
	Ĵ	
	•	bm

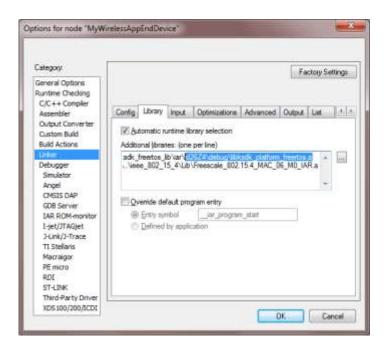
- Make sure to change the KSDK\_PATH variable to the KL26Z SA SDK folder before opening any KL26Z workspace.
- A copy-paste of the existing frdmkl26z project and workspace is a good starting point
- For simplicity, this has already been done on your machine, in the form of the frdmkl26z\_frdm-mcr20a folder.
- The files in the above folder are the most important board configuration elements which need to be modified for FRDM-CR20A proper operation.



# **Configuring the Linker Options**

- In the Linker->Config panel, select the connectivity linker file
- The linker file is already created on your machine having as a starting point the connectivity KL46Z one and NOT the original KSDK KL26Z file.
- In the Linker-> Library Panel, make sure that the KSDK KL26Z platform library for FreeRTOS is selected

Category.	Factory	Settings
Seneral Options Runtime Checking C/C++ Compiler Assembler Dutput Converter Custom Build	Config Library Input Optimizations Advanced Output List	].a.].a
Build Actions	Verride default	
Debugger Simulator Angel CMSIS DAP GDB Server	Laforn Vinker/Md2624van/ <u>MKLa(st1700008_covvectody</u> )	
IAR ROM-monitor L-jet/JTAGjet 3-Link/J-Trace TT.Stellaris Macraigor PE micro PDT	heap_size=7400 raen_vector_table=1	47
RDE		





# **Configuring the Preprocessor Definintions**

Options for node "MyW	irelessAppEndDevice"
Options for node "MyW Category: General Options Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator Angel CMSIS DAP GDB Server IAR ROM-monitor I-jet/JTAGjet	Factory Settings         Multi-file Compilation         Discard Unused Publics         Language 1 Language 2 Code Optimizations Output List         gnore standard include directories         Additional include directories: (one per line)         SPROJ_DIR\$\.\.\.\.\.\.\.\.\.\.\.\.\.\.\.\.\.\.\.
I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro	Preinclude file:
RDI ST-LINK Third-Party Driver XDS 100/200/ICDI	FSL_RTOS_FREE_RTOS FREEDOM gMacFeatureSet_d=gMacFeature

External Use 64

**#FTF2015** 

 IAR Preprocessor Definitions: CPU MKL46Z256VLL4=1 FSL RTOS FREE RTOS gMacFeatureSet\_d=gMacFeatureSet\_06M0\_d gFsciIncluded\_c=0 gRNG\_HWSupport\_d=gRNG\_NoHWSupport\_d gRNG\_UsePhyRngForInitialSeed\_d=1 gSecLib\_HWSupport\_d=gSecLib\_NoHWSupport\_d gKeyBoardSupported d=1 gKBD\_KeysCount\_c=4 gLEDsOnTargetBoardCnt\_c=4 gMCR20\_ClkOutFreq\_d=gCLK\_OUT\_FREQ\_DISABLE



# **Preprocessor Definitions - Continued**

Definition on FRDM-KL46Z	Description	Definition on FRDM-KL26Z
CPU_MKL46Z256VLL4=1	Processor definition	CPU_ MKL26Z128VLH4=1
FSL_RTOS_FREE_RTOS	RTOS definition used in the abstraction layer	FSL_RTOS_FREE_RTOS
gMacFeatureSet_d= gMacFeatureSet_06M0_d	IEEE 802.15.4 MAC feature set (2006, Cortex M0+)	gMacFeatureSet_d= gMacFeatureSet_06M0_d
gFsciIncluded_c=0	Freescale Serial Connectivity Interface included	gFsciIncluded_c=0
gRNG_HWSupport_d=	Hardware Random Number generator	gRNG_HWSupport_d=
gRNG_NoHWSupport_d	support	gRNG_NoHWSupport_d
gRNG_UsePhyRngForInitialSeed_d= 1	Get initial RNG seed from over-the-air energy measurements	gRNG_UsePhyRngForInitialSeed_d= 1
gSecLib_HWSupport_d=	Hardware acceleration for security	gSecLib_HWSupport_d=
gSecLib_NoHWSupport_d	(AES-128)	gSecLib_NoHWSupport_d
gKeyBoardSupported_d=1	Keyboard on the FRDM-CR20A supported	gKeyBoardSupported_d=1
gKBD_KeysCount_c=4	Number of keys avalable	gKBD_KeysCount_c=4
gLEDsOnTargetBoardCnt_c=4	Number of LEDs available	gLEDsOnTargetBoardCnt_c=4
gMCR20_ClkOutFreq_d=	Have the MCR20A output a clock signal	gMCR20_ClkOutFreq_d=
gCLK_OUT_FREQ_DISABLE	for the MCU	gCLK_OUT_FREQ_DISABLE



#### **Preprocessor Definitions in Sources - SPI**

```
#include "EmbeddedTypes.h"
#include "SPI.h"
#include "fsl_clock_manager.h"
#include "pin_mux.h"
#if BOARD_USE_DSPI
```

```
#include "fsl_dspi_hal.h"
    #include "fsl_dspi_master_driver.h"
#else
    #include "fsl_spi_hal.h"
    #include "fsl_spi_master_driver.h"
#endif
```

- Transceiver Driver SPI transfers support
  - C:\Freescale\MCR20A\_IEEE\_802\_15\_4\ieee\_ 802\_15\_4\Source\Phy\Source\XcvrSpi\SPI.c
  - Contains wrapper code over KSDK drivers that enable different types of SPI modules: SPI (KL46Z, KL26Z) or DSPI (K64F).
  - Ensures unified SPI transfer API for the MCR20A transceiver driver.
  - The BOARD\_USE\_DSPI macro selects DSPI or SPI



# Putting it all Together

Connect the FRDM-K64F Coordinator with the FRDM-KL26Z End Device







# Build and Download the FRDM-KL26Z Application

- Configure the RF channel mask to be the same as on the K64F Coordinator.
- Build the KSDK KL26Z FreeRTOS platform library from the workspace. The USB library is not necessary.
- Build the MyWirelessApp End Device KL26Z configuration for FreeRTOS
- Plug the FRDM-KL26Z/FRDM-CR20A node to the USB port of the FRDM-KL26Z labelled "SDA" and wait for the CMSIS-DAP drivers to be installed.



#### **Create a Simple Two Node Network**

- Open two PuTTY terminals corresponding to the virtual COM Ports of exposed by the two nodes: FRDM-KL26Z/FRDM-CR20A and FRDM-K64F/FRDM-CR20A.
- Configure the serial ports with 115200 baud.
- Reset the boards. Each RGB LED should start flashing and each terminal should display the following text:

Please press any key on board to start running the application

- Press SW1 or SW2 on the FRDM-CR20A of the coordinator (K64F).
- The RGB LED should **stop flashing** and the corresponding terminal should display the same text as shown before in this presentation.
- Now press SW1 or SW2 on the FRDM-CR20A of the end device (KL26Z). The RGB led on this board should also stop flashing.



#### **Create a Simple Two Node Network - Continued**

# The terminals for the two nodes should show the following messages:

	B COM68 - PuTTY			Putty
	Received an MLME-Beacon Notify Indication	A		
	Received an MLME-Beacon Notify Indication		n board to start running the application.	Press any switch on board t
	Received an MLME-Beacon Notify Indication		Coordinator application is initialized and ready.	MyWirelessApp Demo Coordina
	Received an MLME-Beacon Notify Indication			
	Received an MLME-Beacon Notify Indication		rgy Detection Scan	Initiating the Energy Detec
	Received an MLME-Beacon Notify Indication		can Request message to the MACDone	Sending the MLME-Scan Reque
	Received an MLME-Beacon Notify Indication		Scan Confirm message from the MAC	Received the MLME-Scan Conf
	Received an MLME-Beacon Notify Indication		he following results:	ED scan returned the follow
	Found a coordinator with the following properties:		40 68 60 30 00 34 2C 30 00 00 00 ]	[00 0C 24 3C 00 40 68 60
	Address0xCAFE		an the logical channel $0x0B$ was selected	Based on the ED scan the lo
	PAN ID0xBEEF			
	Logical Channel0x0B	=	ordinator on channel(s): 0x0B.	-
	Beacon Spec0xCFFF		tart Request message to the MACDone	
	Link Quality0x9F		nator with PAN ID 0xBEEF, and short address 0xCAFE.	Started the coordinator wit
	Associating to PAN coordinator on channel 0x0B		receive data over the UART.	Ready to send and receive d
MACDone	Sending the MLME-Associate Request message to the MACDone			
	Successfully associated with the coordinator.		ssociate Indication from the MAC	
	We were assigned the short address 0x0001		ssociate Response message to the MACDone	Sending the MLME-Associate
			omm-Status Indication from the MAC	Received an MLME-Comm-Statu
	Ready to send and receive data over the UART.			
		-		
	C	)		
	Ready to send and receive data over the UART.	•	omm-Status Indication from the MAC	Received an MLME-Comm-Statu







External Use 70

### **Create a Simple Two Node Network - Conclusion**

- What actually happens: **Coordinator**:
  - Starts Energy Detection Scan on the channel mask selected (only one in our case)
  - If the channel is free, the coordinator starts a network with PAN ID 0xBEEF and assigns itself a short 16-bit address of 0xCAFE
  - It then waits with the receiver open to receive the a message from an end device
  - Upon receiving the Active Scan from the device it sends a beacon
  - Upon receiving the association request from the end device it starts the association procedure
- What actually happens: End Device:

External Use | 71

- Starts an Active Scan procedure on the selected channel mask
- Upon receiving the beacon from the coordinator it sends an association request
- After the association, characters typed in a terminal console will be sent over the air and printed in the other. Try it!



# Conclusions, Q&A and Session Closing



External Use 72



# Conclusions

• By now, you should be able to:

External Use 73

- Appreciate the flexibility and differentiating features that MCR20A offers in adding connectivity to microcontroller systems.
- Effectively describe, at a high level, the Freescale MCR20A development setup for IEEE® 802.15.4 applications.
- Understand the way the IEEE® 802.15.4 MAC and PHY integrate with the Kinetis SDK.
- Describe the **key configuration elements** for switching a microcontroller with another for MCR20A usage.
- Apply the knowledge gained in this presentation to begin or refine your design efforts for **IoT applications** using MCR20A.



# Thank you for being a great audience! Please come forward with questions.



111:1

External Use 74





www.Freescale.com

© 2015 Freescale Semiconductor, Inc. | External Use