

# HANDS-ON WORKSHOP: MCUXpresso SOFTWARE AND TOOLS

CLARK JARVIS

MCUXpresso SOFTWARE AND TOOLS

PRODUCT MARKETER

AMF-DES-T2632 | MAY 2017



SECURE CONNECTIONS  
FOR A SMARTER WORLD

NXP and the NXP logo are trademarks of NXP B.V. All other product or service names are the property of their respective owners. © 2017 NXP B.V.

PUBLIC



# AGENDA

- Overview of MCUXpresso Software and Tools
  - MCUXpresso IDE
  - MCUXpresso SDK
  - MCUXpresso Config Tools
- Walk-through of Tools
- Hands-on Lab
  - MCUXpresso SDK – Configuration and Build
  - MCUXpresso IDE – Getting Started
  - MCUXpresso Config Tools – Board Configuration



# Overview of MCUXpresso Software and Tools

MCUXpresso IDE, SDK, Config Tools



## MCUXpresso Software and Tools

COMMON TOOLKIT  
FOR THOUSANDS  
OF KINETIS® & LPC  
MICROCONTROLLERS



[www.nxp.com/mcuxpresso](http://www.nxp.com/mcuxpresso)



# MCUXpresso Software and Tools

for Kinetis and LPC microcontrollers



## MCUXpresso IDE

Edit, compile, debug and optimize in an intuitive and powerful IDE



## MCUXpresso SDK

Runtime software including peripheral drivers, middleware, RTOS, demos and more



## MCUXpresso Config Tools

Online and desktop tool suite for system configuration and optimization



# MCUXpresso Software and Tools

- Common toolkit across Kinetis and LPC microcontrollers
- Easy to use
- High quality
- Shared software experience and broader portfolio support
- Offers easy migration and scalability
- Supports large ARM® Cortex®-M ecosystem
- Built on the 'best of' Kinetis SDK, LPCXpresso and Kinetis Design Studio IDEs



## MCUXpresso Software and Tools

- IDE
- SDK
- Config Tools

For NXP's ARM® Cortex®-M controllers

- Kinetis MCUs
- LPC Microcontrollers
- i.MX Application Processors



# Origins of MCUXpresso Software & Tools

## Kinetis and LPC SW

Independent software and tools

## MCUXpresso Software and Tools

Supporting Kinetis & LPC Cortex-M MCUs

LPCXpresso IDE  
& Kinetis Design Studio



MCUXpresso IDE

Kinetis SDKv2



MCUXpresso SDK

Kinetis Expert



MCUXpresso Config Tools



# MCUXpresso IDE

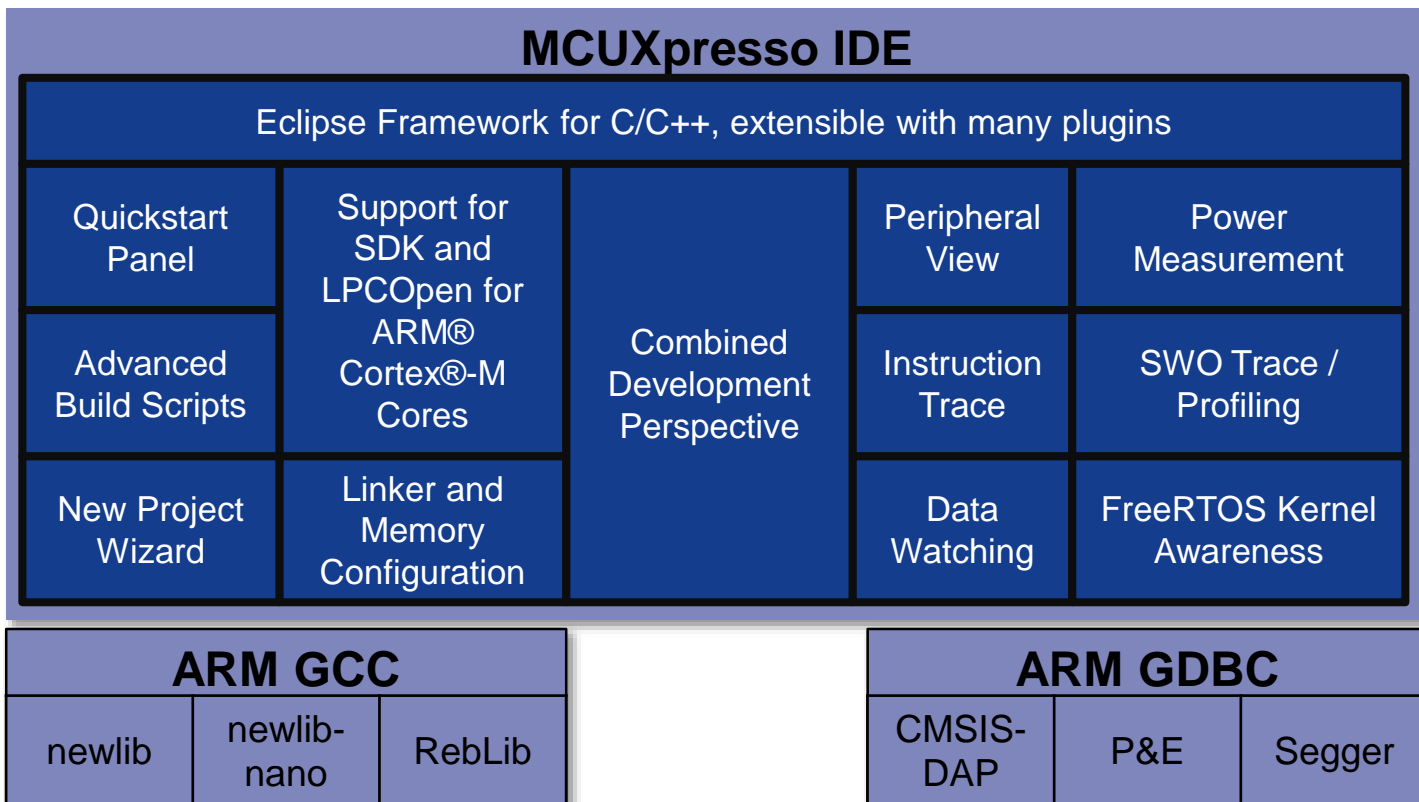
MCUXpresso IDE



# MCUXpresso IDE



Free Eclipse and GCC-based IDE for C/C++ development on Kinetis and LPC MCUs



## Product Features

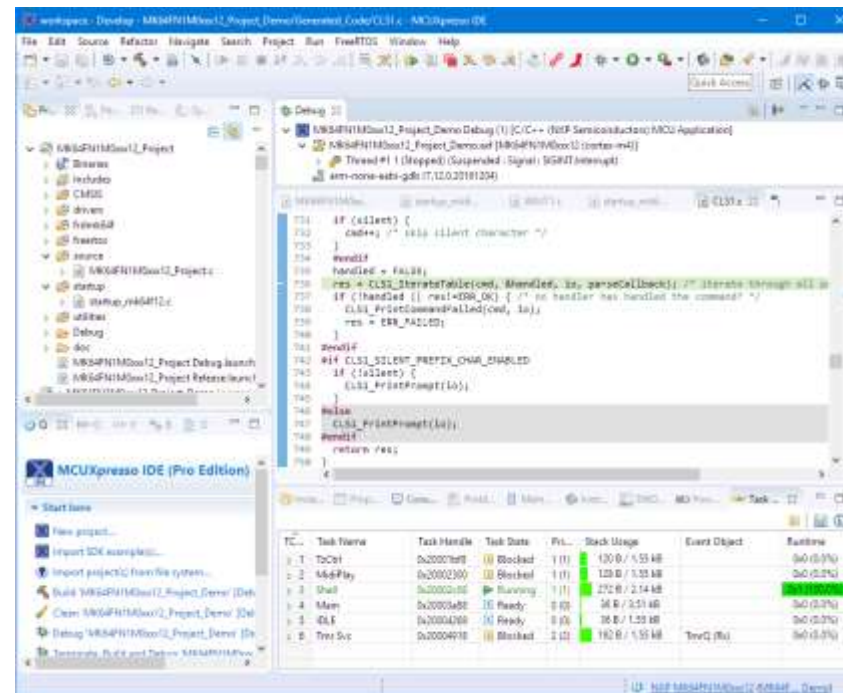
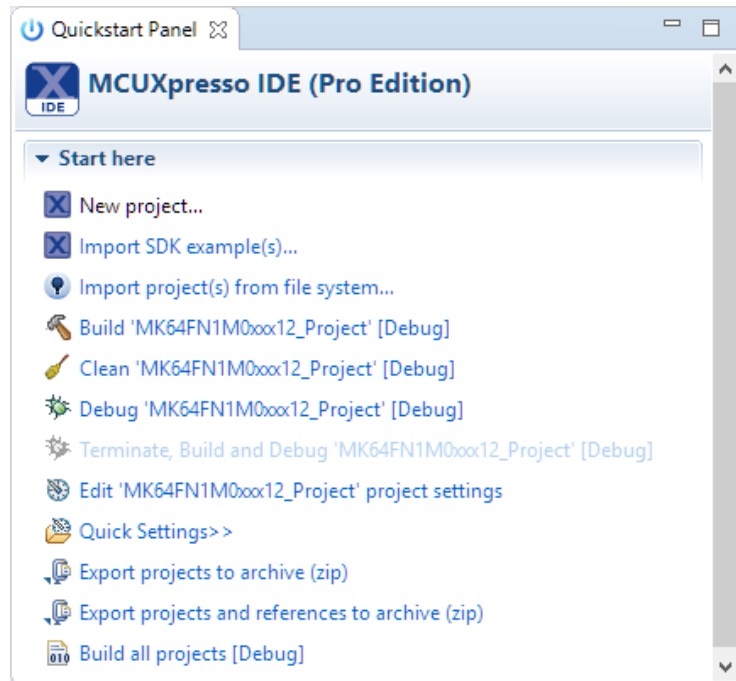
- **Feature-rich, unlimited code size**, optimized for **ease-of-use**, based on **industry standard Eclipse** framework for **NXP's Kinetis and LPC MCUs**
- **Application development** with Eclipse and GCC-based IDE for advanced **editing, compiling and debugging**
- Supports **custom** development boards, **Freedom, Tower** and **LPCXpresso** boards with debug probes from **NXP, P&E** and **Segger**
- **Free Edition:** Full Featured, **unlimited Code Size**, no special activation needed, community based support
- **Pro Edition:** Email IDE support, **Advanced Trace Features**





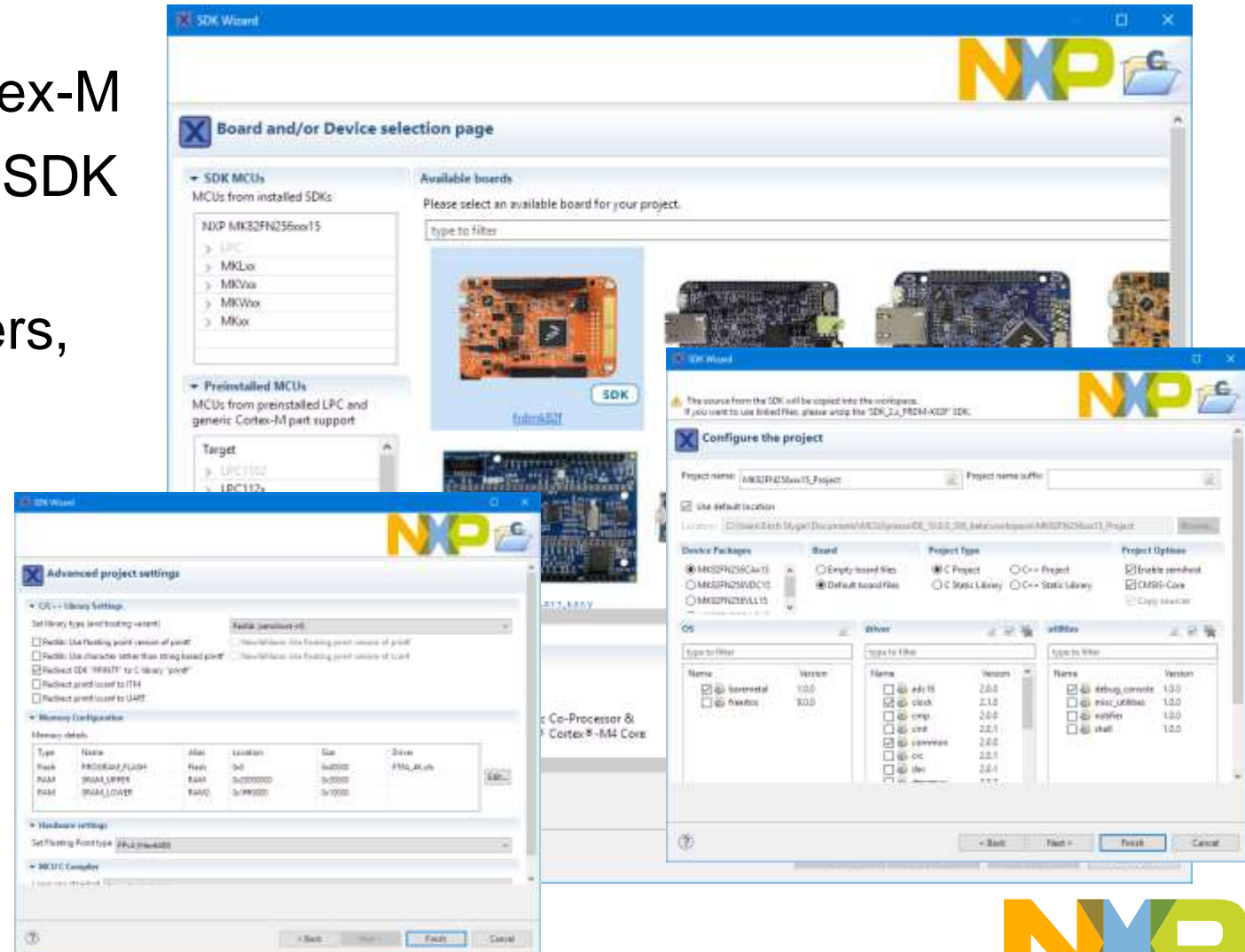
# Built for Ease-of-Use

- **Quickstart Panel** guides users to most commonly used options
  - One-Click access to most used functions
- **Develop Perspective** for both project editing and debugging
  - Simplifies Eclipse usage



# New Project Wizard: Data Driven Device Support

- SDK MCUs (LPC and Kinetis)
- Preinstalled LPC and generic Cortex-M
- Installable device support through SDK packages (data driven)
- Selection of package, RTOS, drivers, utilities
- Standalone and linked projects
- Advanced project settings



# MCUXpresso IDE Supported Debug Probes



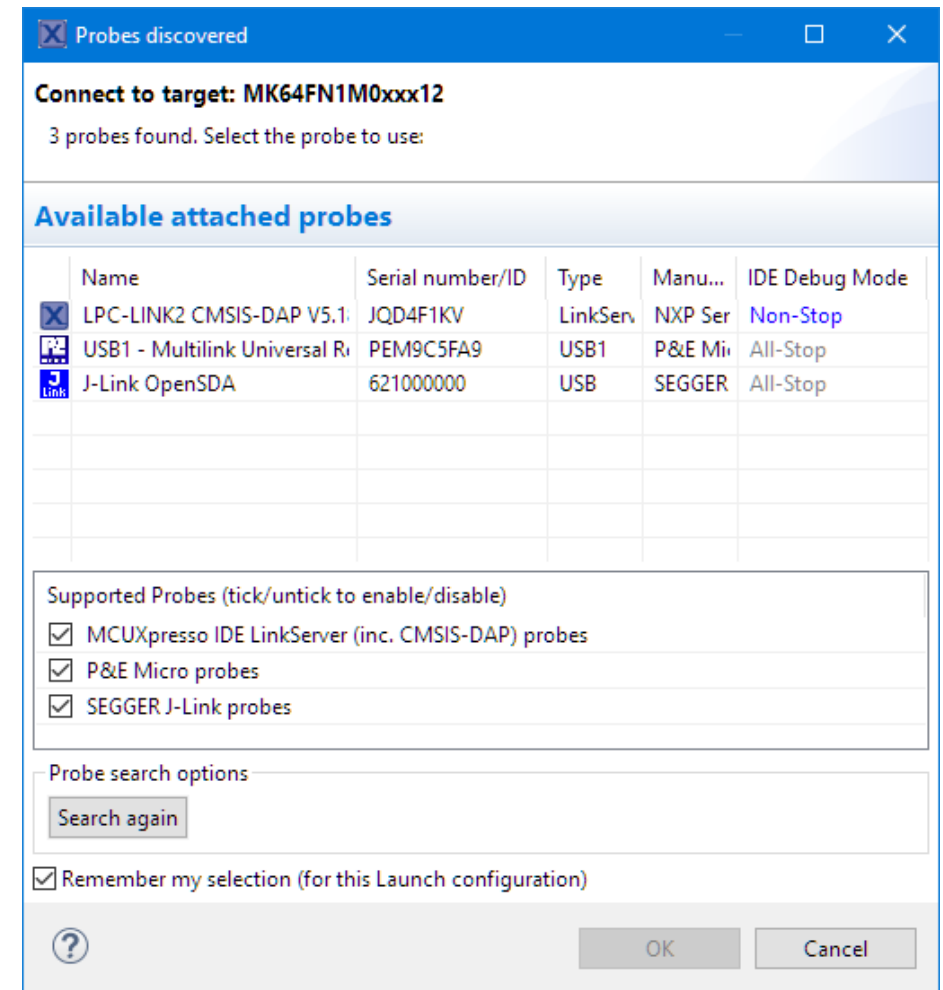
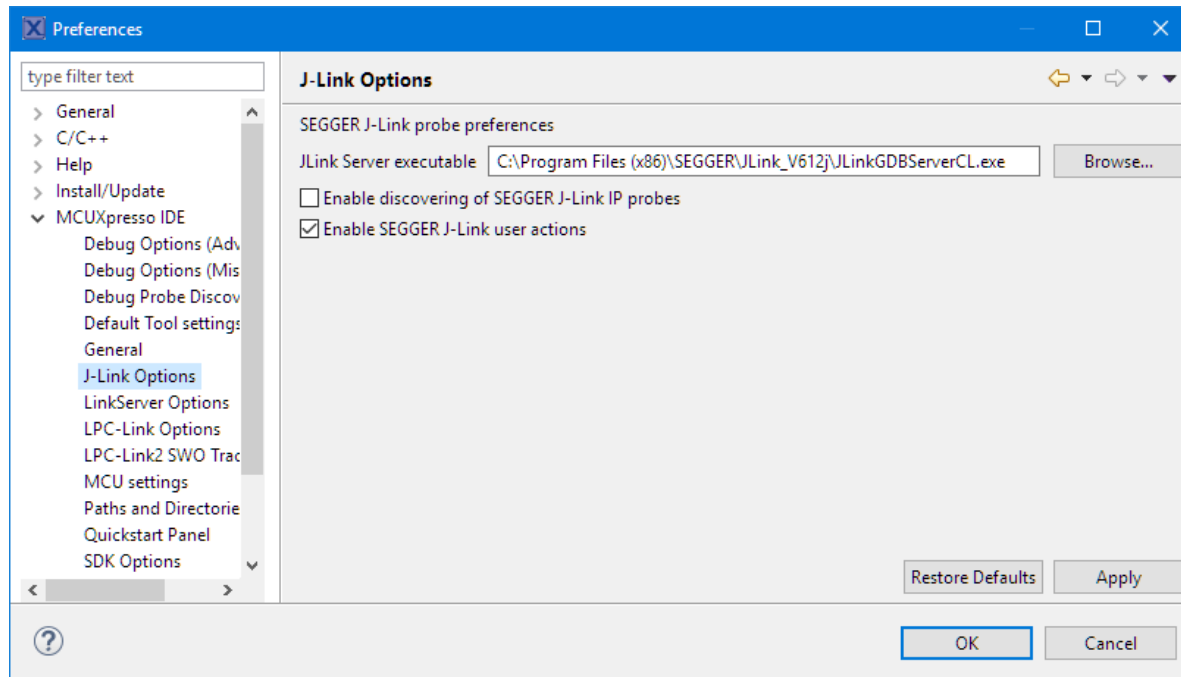
- **LinkServer MCUXpresso Probes**
  - LPC-Link, LPC-Link2, including debug probes on LPCXpresso V2/V3 boards
  - CMSIS-DAP (including CMSIS-DAP on Kinetis FRDM and TOWER boards)
- **Segger (J-Link)**
  - Native and OEM J-Link for Kinetis and LPC
  - OpenSDA Segger Firmware on FRDM/TOWER boards
  - LPC-Link2 Segger Firmware on LPC-Link/LPC-Link2
- **P&E (Multilink)**
  - Native Multilink for LPC and Kinetis
  - OpenSDA P&E Firmware on FRDM/TOWER boards



# Automatic Probe Discovery and Configuration



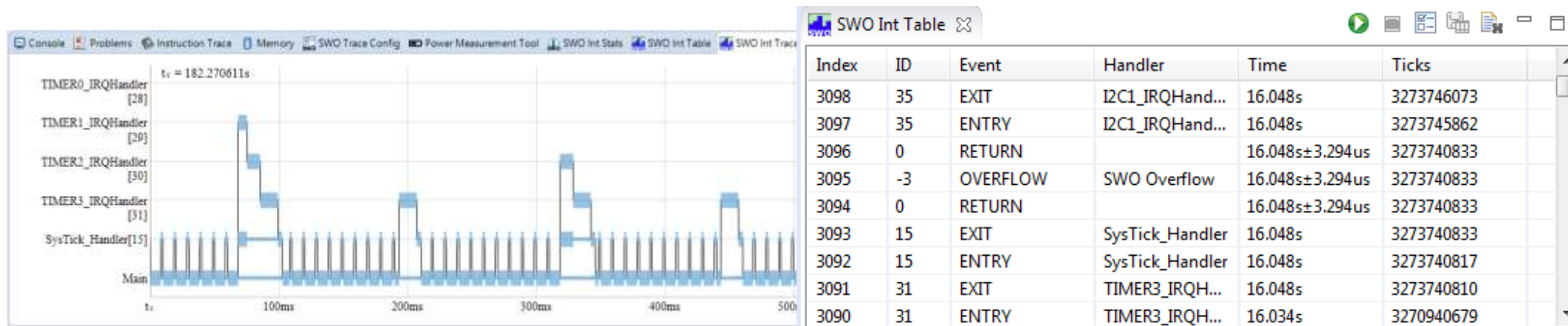
- Discovery of attached supported probes
- Automatic creation of Debug Launch Configurations
- Setting to locate SEGGER GDB



# Professional Edition



- PRO Edition License available for \$495
  - Extended PRO features perpetual
  - SWO Interrupt Trace with interrupt trace table, including timeline
  - Data watch: one with free edition, up to four with PRO edition
  - 1 Year of email support for MCUXpresso IDE, ticket based
- Existing valid LPCXpresso PRO licenses automatically enabled for MCUXpresso





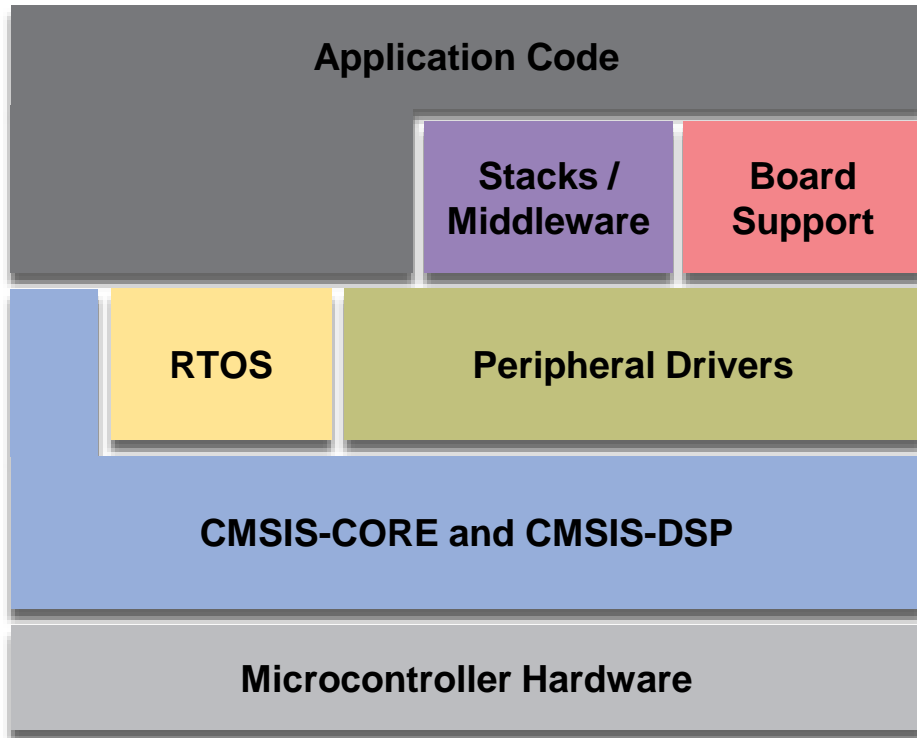
# MCUXpresso SDK



# MCUXpresso SDK



The software framework and reference for Kinetis & LPC MCU application development



## Product Features

### Architecture:

- CMSIS-CORE compatible
- Single driver for each peripheral
- Transactional APIs w/ optional DMA support for communication peripherals

### Integrated RTOS:

- FreeRTOS v9
- RTOS-native driver wrappers

### Integrated Stacks and Middleware

- USB Host, Device and OTG
- lwIP, FatFS
- Crypto acceleration plus wolfSSL & mbedTLS
- SD and eMMC card support

### Reference Software:

- Peripheral driver usage examples
- Application demos
- FreeRTOS usage demos

### License:

- BSD 3-clause for startup, drivers, USB stack

### Toolchains:

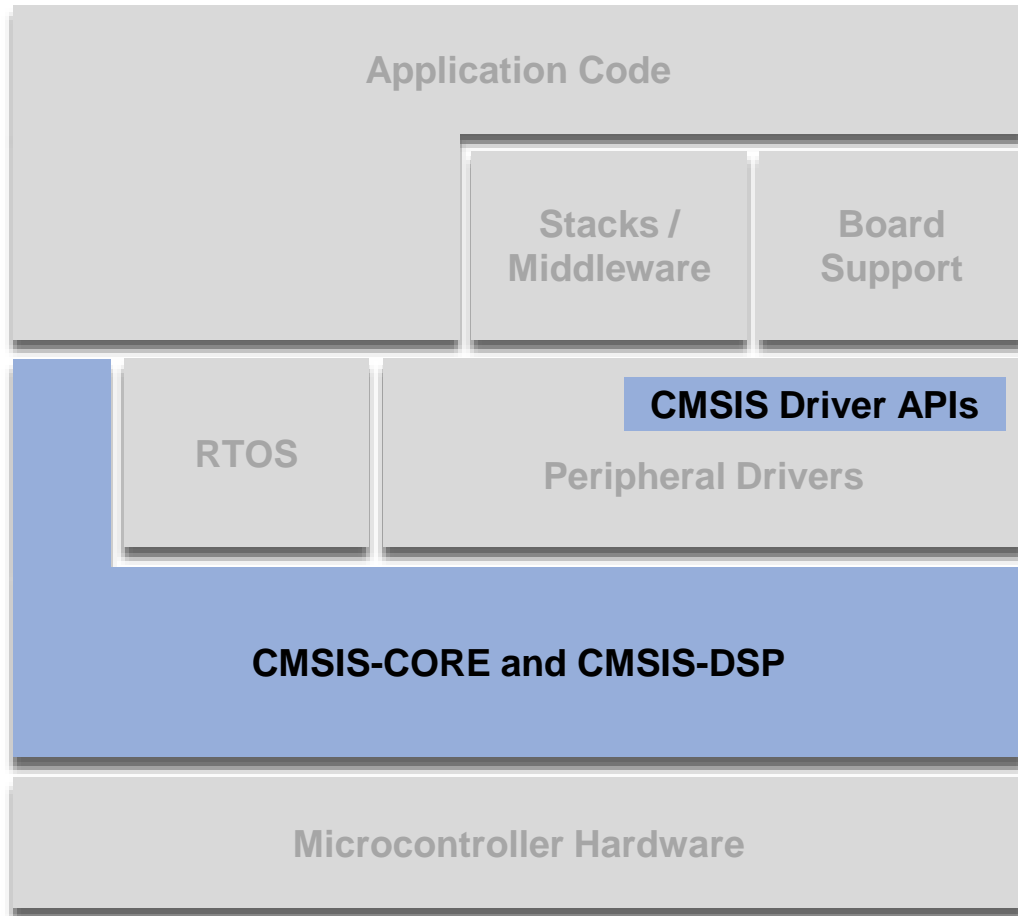
- MCUXpresso IDE
- IAR®, ARM® Keil®, GCC w/ Cmake

### Quality

- Production-grade software
- MISRA 2004 compliance
- Checked with Coverity® static analysis tools



# MCUXpresso SDK – CMSIS Device Support



**CMSIS-CORE** provides a standard for a basic run-time system and user access to the core and the peripherals:

- Hardware Abstraction Layer (HAL) - definitions for the SysTick, NVIC, FPU registers, and core access functions
- Standardized MCU header file format – common register/bit access methods, system exception and interrupt naming
- Standard methods for system initialization – for example, the [SystemInit\(\)](#) function for essential system configuring
- Intrinsic functions used to generate CPU instructions that are not supported by standard C functions.

**CMSIS-DSP** is a suite of common signal processing functions including math, filters, matrix, transforms, motor control, statistical, and interpolation functions.

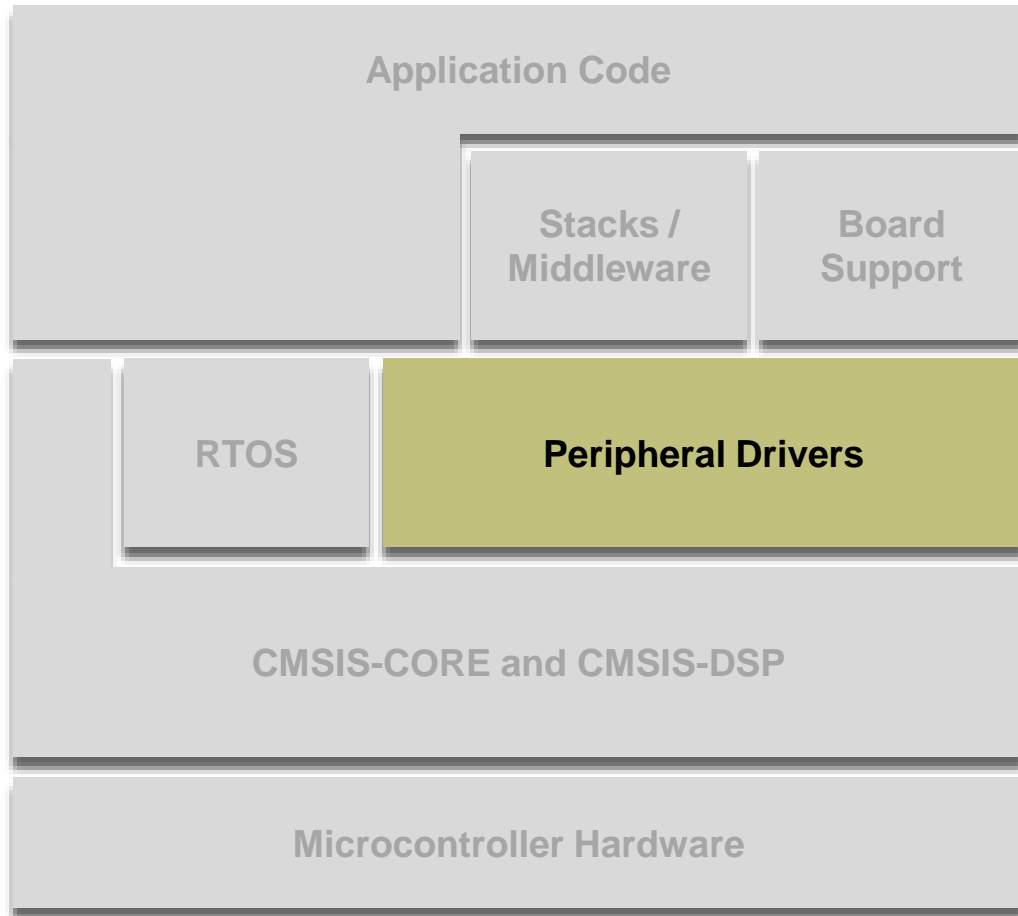
**CMSIS-Driver** is an API standard for common peripheral and on-chip interfaces.

Learn more at: [www.keil.com/cmsis](http://www.keil.com/cmsis)





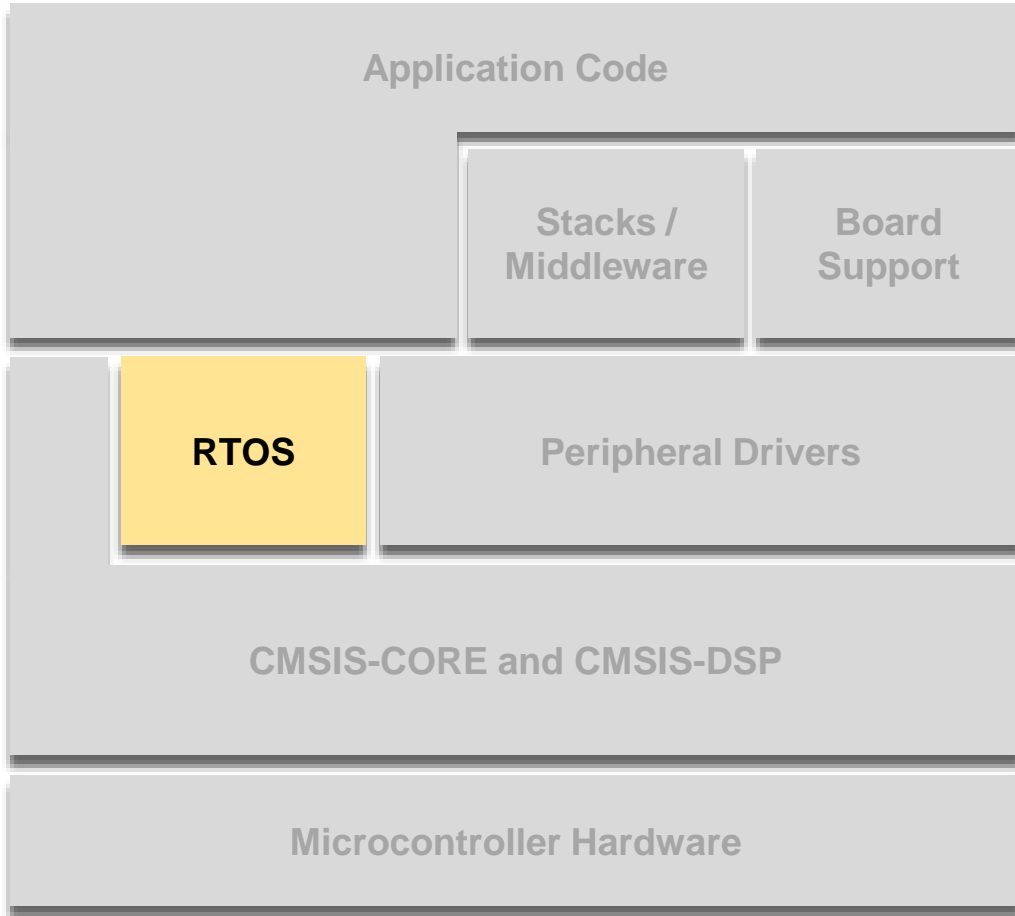
# MCUXpresso SDK – Peripheral Drivers



- Single driver for each peripheral
- Full peripheral coverage for each MCU
- All drivers include low-level functional APIs
- Communication peripheral drivers feature transactional APIs
  - Non-blocking, interrupt based
- Communication peripheral drivers also have optimized RTOS wrapper drivers
  - Uses native RTOS APIs – no operating system abstraction



# MCUXpresso SDK – RTOS



FreeRTOS kernel pre-integrated

- Demonstration applications
- RTOS Usage examples

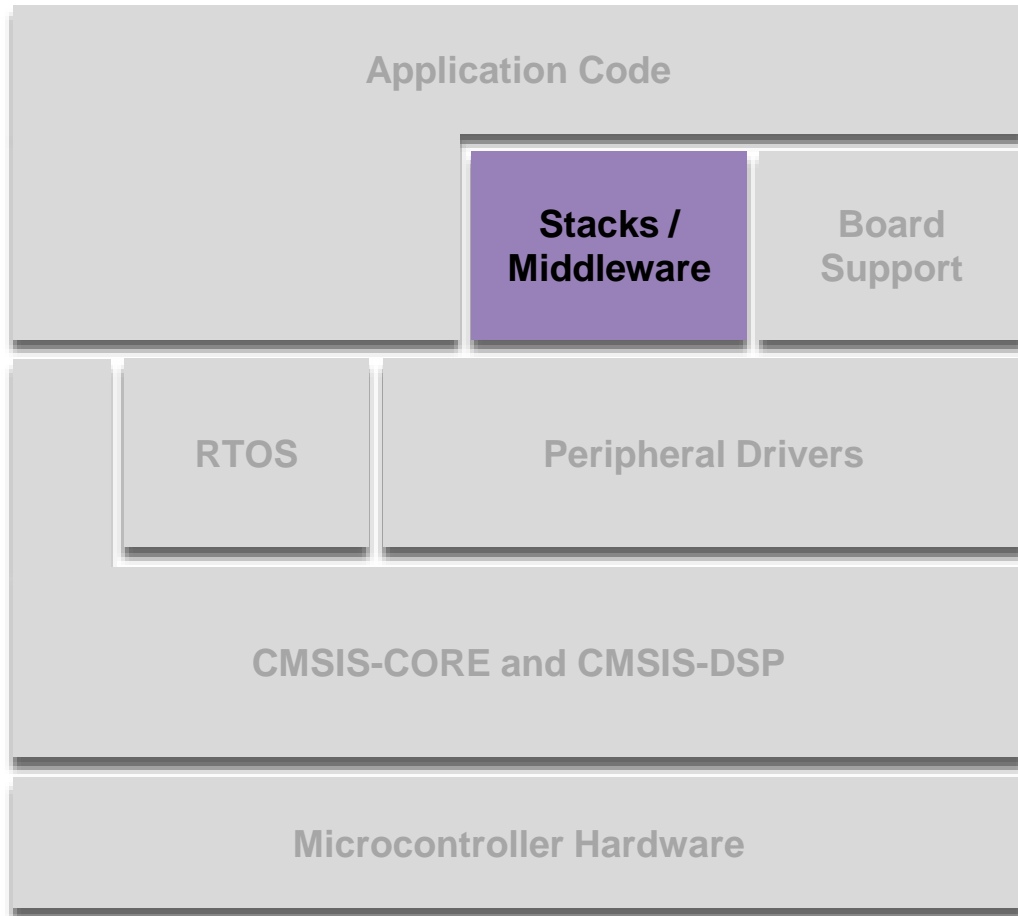


RTOS examples include:

- freertos\_dspi
- freertos\_event
- freertos\_generic
- freertos\_hello
- freertos\_i2c
- freertos\_mutex
- freertos\_queue
- freertos\_sem
- freertos\_swtimer
- freertos\_tickless
- freertos\_uart
- ucosiii\_hello
- ucosii\_hello



# MCUXpresso SDK – USB Stack

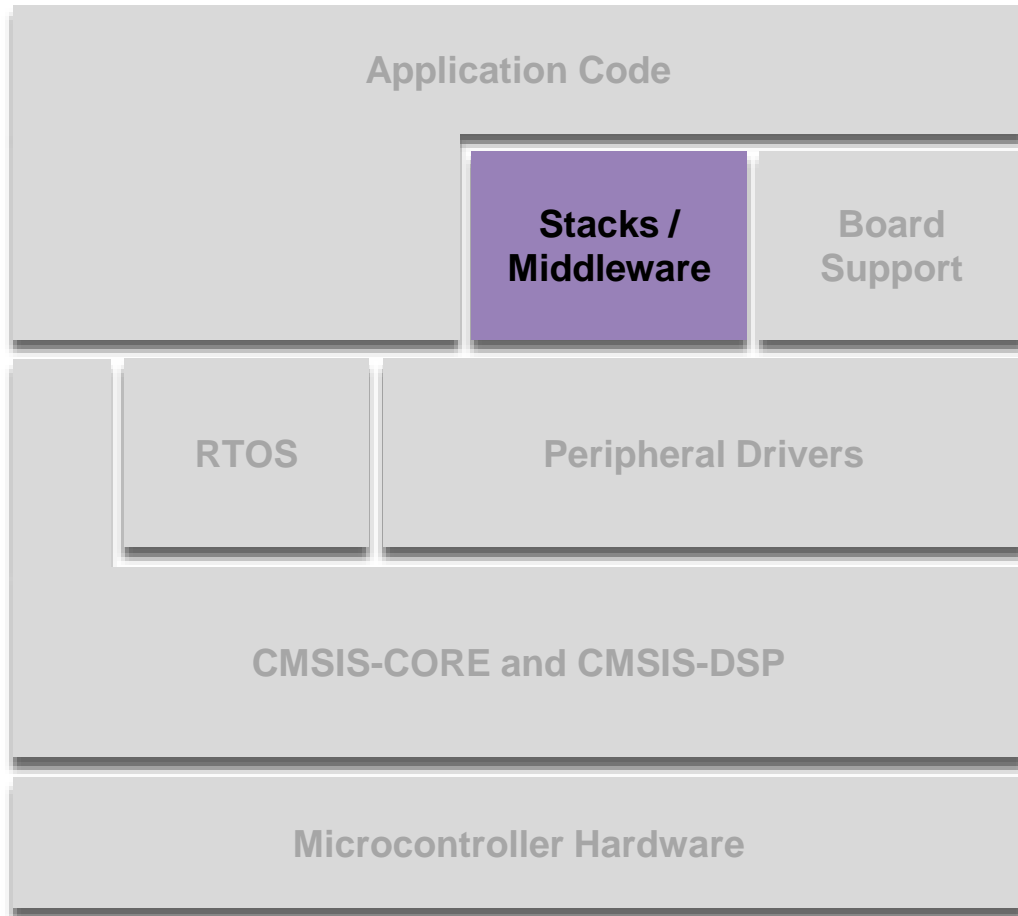


**MCUXpresso SDK USB Stack** is a comprehensive, open-source device and host stack. It supports baremetal and RTOS application, multiple class implementations, several demo applications.

- **70+ demo applications** that support extensive features including:
  - 7 device classes with 3 **composite** examples
  - 5 host classes with **USB hub** support
  - Full-speed and high-speed under USB 2.0 specs
- **High quality stack, ready for production use**
  - **USB-IF certification** on both FS and HS
  - Optimized for code size - down to **6K flash** and 2K RAM – and performance among competition
  - Device demos have “**Lite**” versions that are even smaller in code size



# MCUXpresso SDK – FatFS



FatFS is a generic FAT file system module for small embedded systems. Features include:

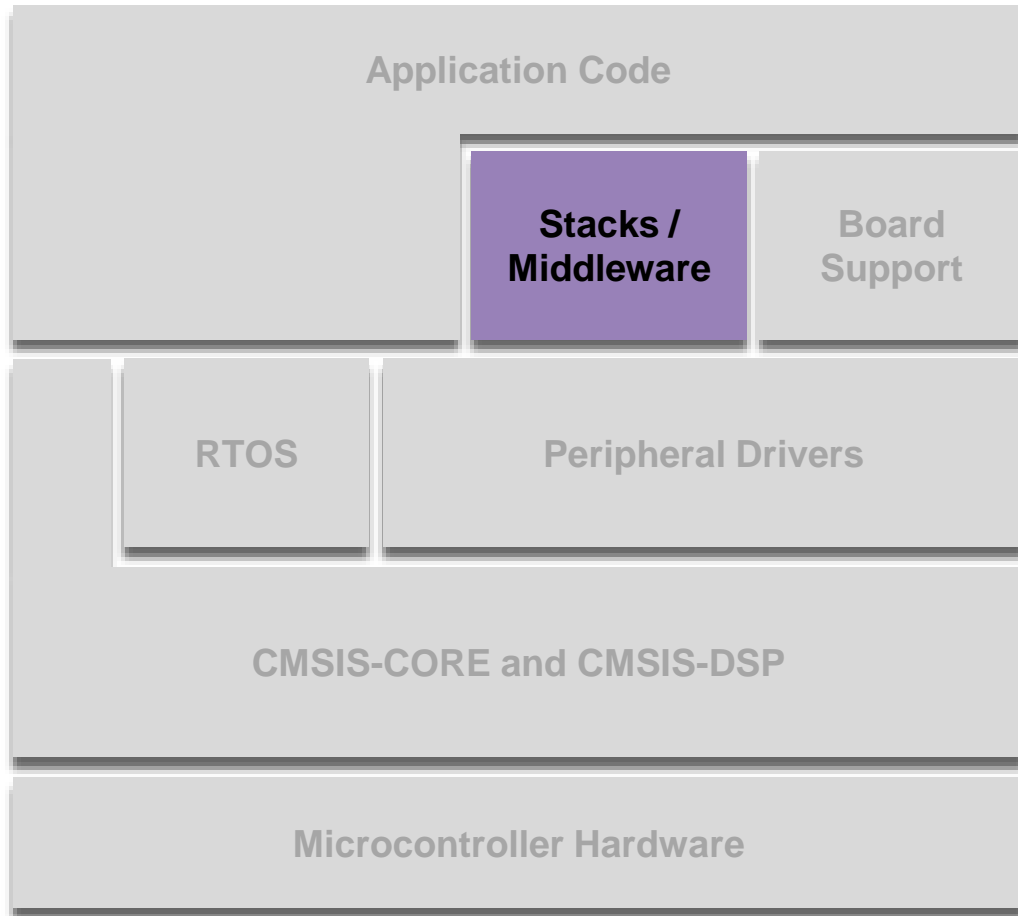
- ANSI C compliant and completely separated from disk I/O layer
- Windows compatible FAT file system
- Very small footprint
- Various configuration options
  - Multiple volumes (physical drives and partitions)
  - Long file name support in ANSI/OEM or Unicode
  - RTOS support
  - FAT sub-types: FAT12, FAT16 and FAT32.

Available demos:

- `sdcard_fatfs`
- `usb_host_msd_fatfs` (baremetal & FreeRTOS)



# MCUXpresso SDK – lwIP



- lwIP – the lightweight Internet Protocol
- A full scale TCP/IP stack for embedded systems
- lwIP supports the following protocols:
  - ARP
  - DHCP
  - IPv4 and v6
  - ICMP
  - TCP
  - IGMP
  - UDP
  - PPP
  - DNS
  - PPPoE
  - SNMP
- Example applications include:
  - lwip\_httpsrv
  - lwip\_ping
  - lwip\_tcpecho
  - lwip\_udpecho





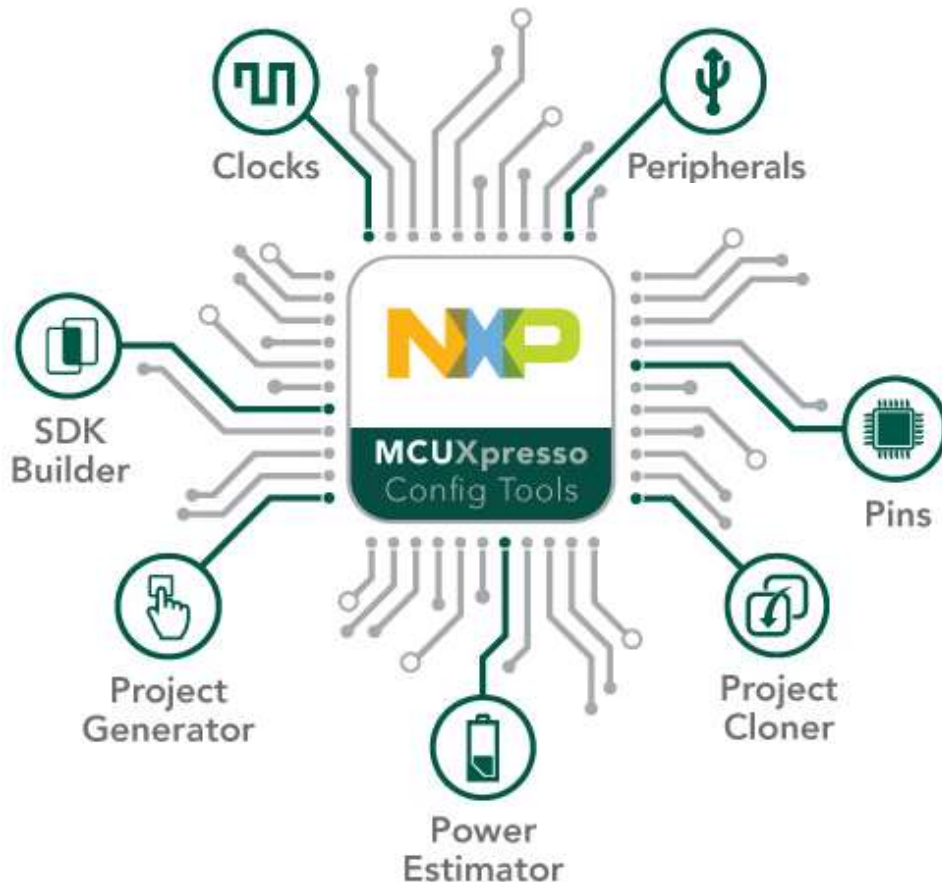
# MCUXpresso Config Tools



# MCUXpresso Config Tools



Integrated configuration and development tools for LPC and Kinetis MCUs



**MCUXpresso Config Tools** is a suite of evaluation and configuration tools that helps guide users from first evaluation to production software development.



**SDK Builder** packages custom SDKs based on user selections of MCU, evaluation board, and optional software components.



**Pins**, **Clocks**, and **Peripheral** tools generate initialization C code for custom board support. Features validation of inputs and cross-tool conflict resolution.



**Project Generator** creates new SDK projects with generated Pins and Clocks source files.



**Project Cloning** creates a standalone SDK project based on an example application available within SDK release.

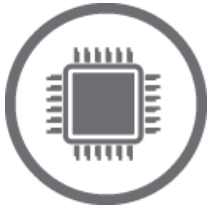


**Power Estimation** tool provides energy and battery-life estimates based on a user's application model.

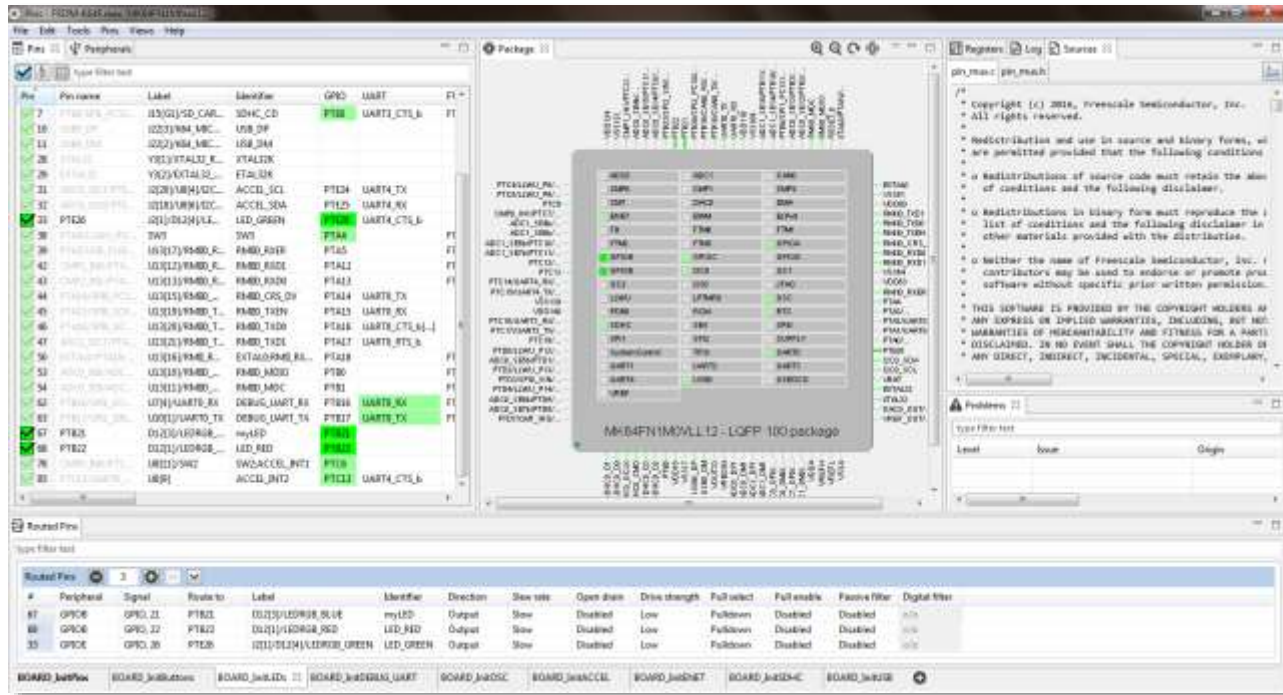
*Available as a standalone tool for select devices.*



# MCUXpresso Config - Pins Tool



Easy-to-use muxing and pin assignments for Kinetis & LPC microcontrollers



## Product Features:

- Part of the MCUXpresso suite of system configuration tools
- Muxing and pin configuration with consistency checking
- ANSI-C configuration code
- Graphical processor package view
- Multiple configuration blocks/functions
- Wizard for optimized assignments of functionality to available pins
  - Selection of Pins and Peripherals
  - Package with IP blocks
  - Routed pins with electrical characteristics
  - Registers with configured and reset values
  - Source code for C/C++ applications
- Documented and easy to understand source code
- Report generation
- Integrates with any compiler and IDE

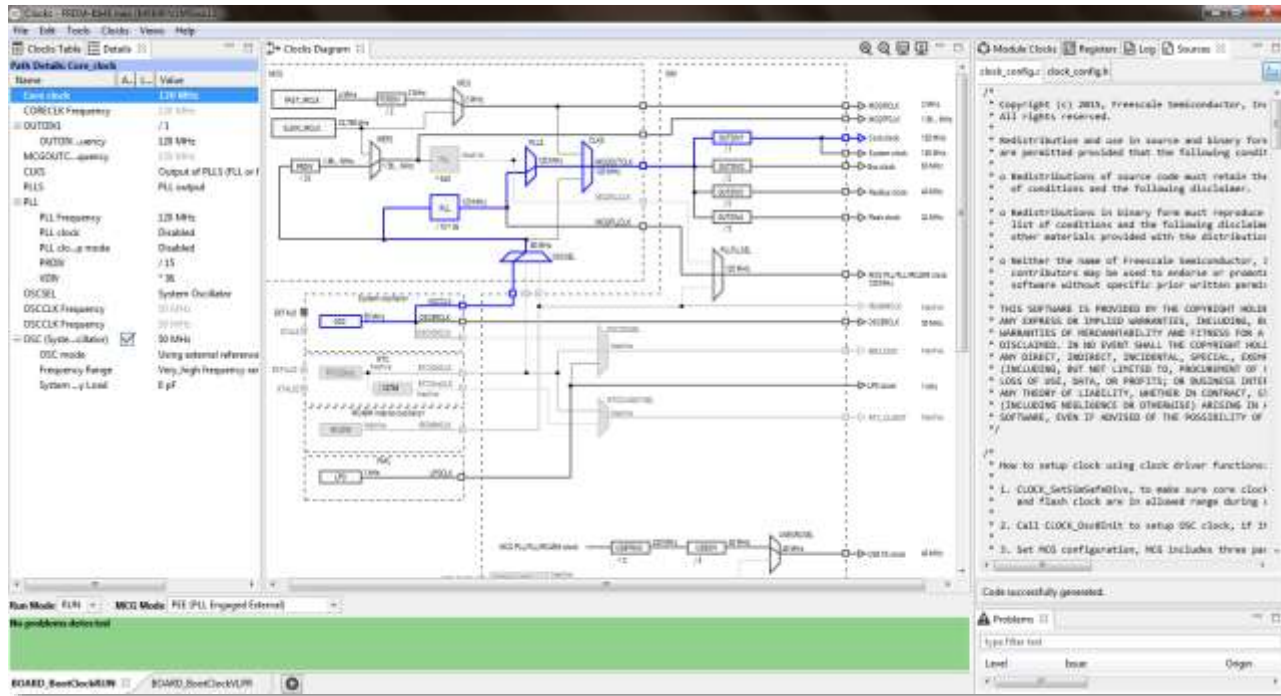




# MCUXpresso Config - Clocks Tool



Easy-to-use clock configuration for Kinetis & LPC



## Product Features:

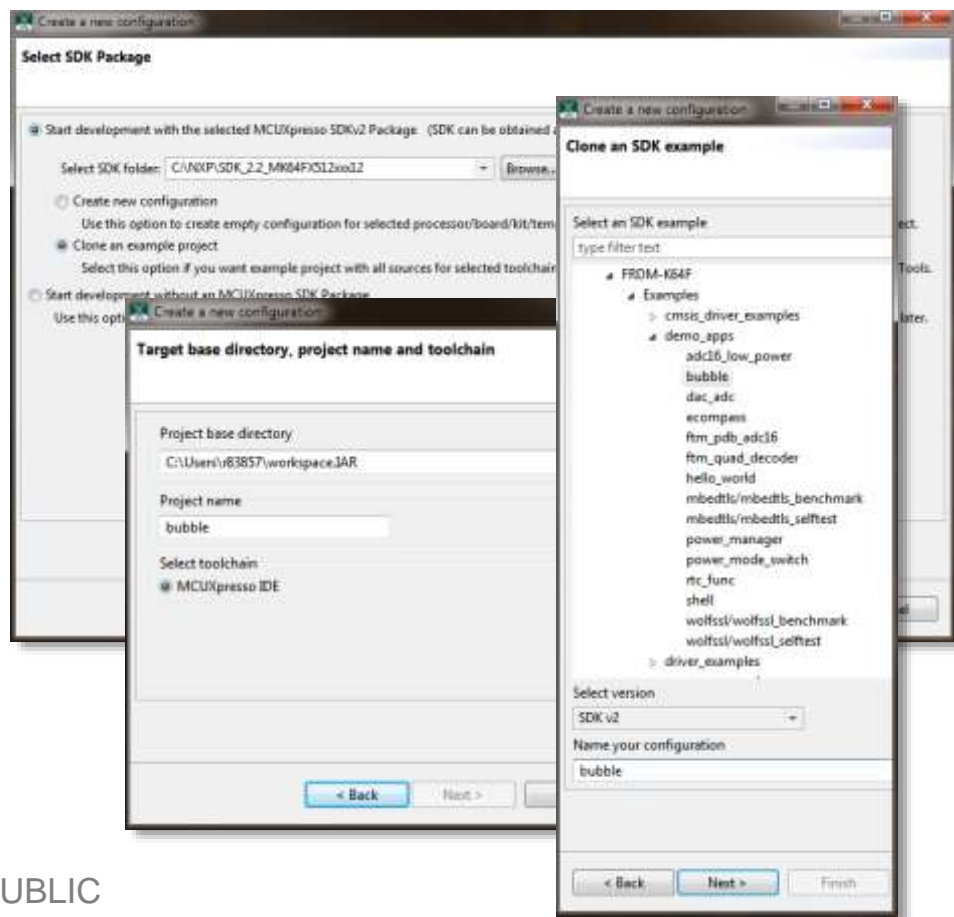
- Part of the MCUXpresso system configuration tools
- System clock configuration with consistency checking
- ANSI-C initialization code
- Graphical clock diagrams
- Multiple configuration blocks/functions
- Easy-to-use guided graphical user interface
  - Selection of Clock Sources
  - Configuration of prescalers and clock outputs
  - Details and Full Diagram views with clock path
  - Registers with configured and reset values
  - Source code for C/C++ applications
- Documented and easy to understand source code
- Report generation



# MCUXpresso Config – Project Cloner



Create a standalone copy of an MCUXpresso SDK example project



## Product Features:

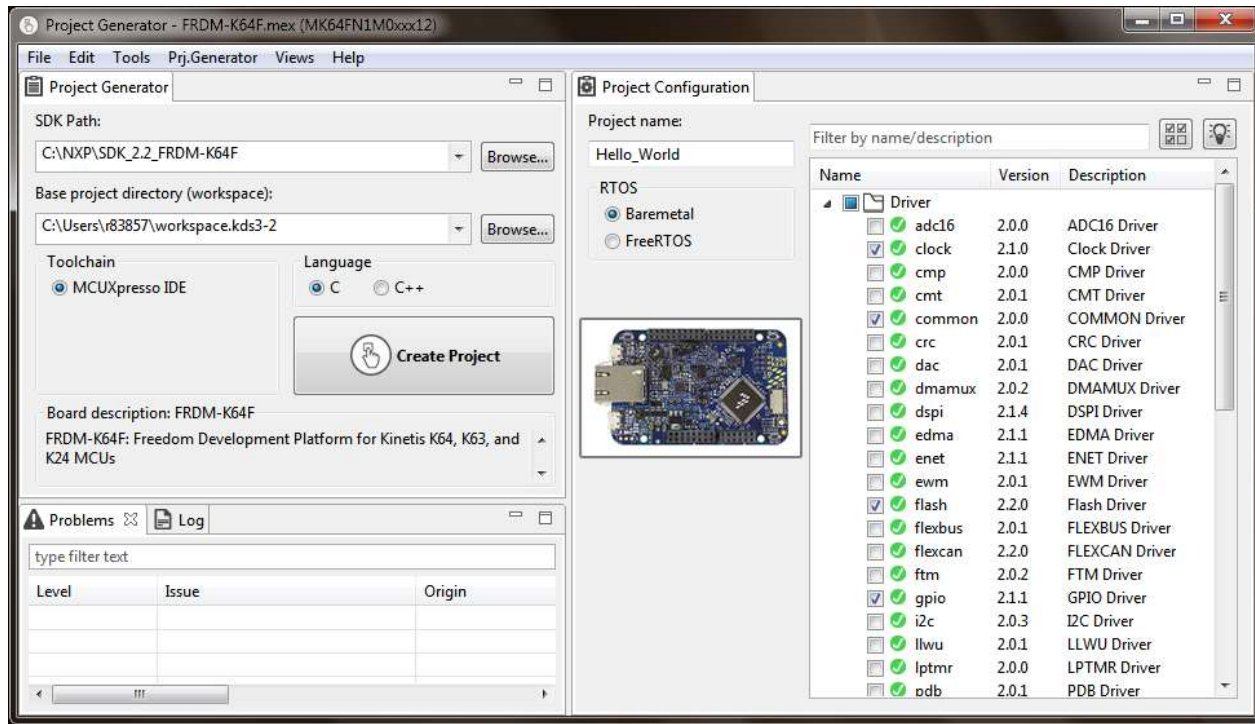
- Ability to generate a fully standalone MCUXpresso project cloned from one of the many included examples.
- Provide a native IDE project for any toolchain supported in your SDK configurations
- Available in the desktop version on the MCUXpresso Config Tool as part of the “New configuration dialog”
- Available in the online version of the MCUXpresso SDK Builder and webpage.
- Clones example projects can be downloaded directly from the MCUXpresso webpage. Online cloned projects provide all project and SDK files required to quickly have an application running on a support NXP development board in a single download.



# MCUXpresso Config – Project Generator



Create a new SDK project incorporating the generated source code available in the MCUXpresso Config Tools



## Product Features:

- Part of the MCUXpresso suite of system configuration tools
- Provide a quick process for integrating output from the Pins, Clocks, and Peripheral tools into a user's custom application.
- Creates a standalone MCUXpresso SDK application, with all required SDK driver files.
- Includes validation checks to ensure that driver files and utilities required as specified by the MCUXpresso Config Tools are included in the generated project.
- One-click solution to reduce the number of included SDK driver to only those required for the project.
- Allows for iterative development by providing the ability to create a new application or update an existing application previously created with Project Generator.
- Available as part of the desktop MCUXpresso Config Tool application.



# MCUXpresso Config - Power Estimation Tool

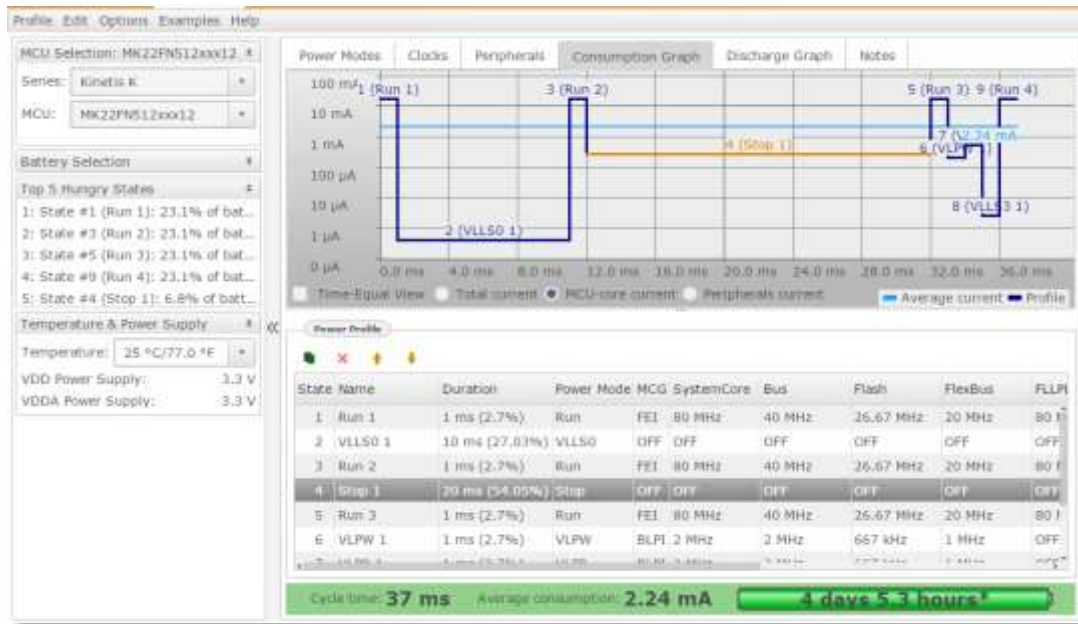
Learn more at: [www.nxp.com/mcuxpresso/config](http://www.nxp.com/mcuxpresso/config)  
Or [www.nxp.com/kinetis/powertool](http://www.nxp.com/kinetis/powertool)



Estimate and optimize your system's power consumption



Helps you design for efficient use of energy



## Product Features

- Available as a standalone desktop application
- Models application states and estimates the power profile
- Provides immediate energy consumption & battery life estimations
- Generates consumption and battery discharge graphs
- Provides ability to save & load profiles and generate reports
- Local and online versions to be available
- English & limited Chinese language support
- Backed by real power measurement data
- Quickly evaluate which Kinetis MCU fits your use-case and power budget
- Accelerates learning curve for advanced power management features
- Ideal tool for developing wearable and other battery-operated applications.



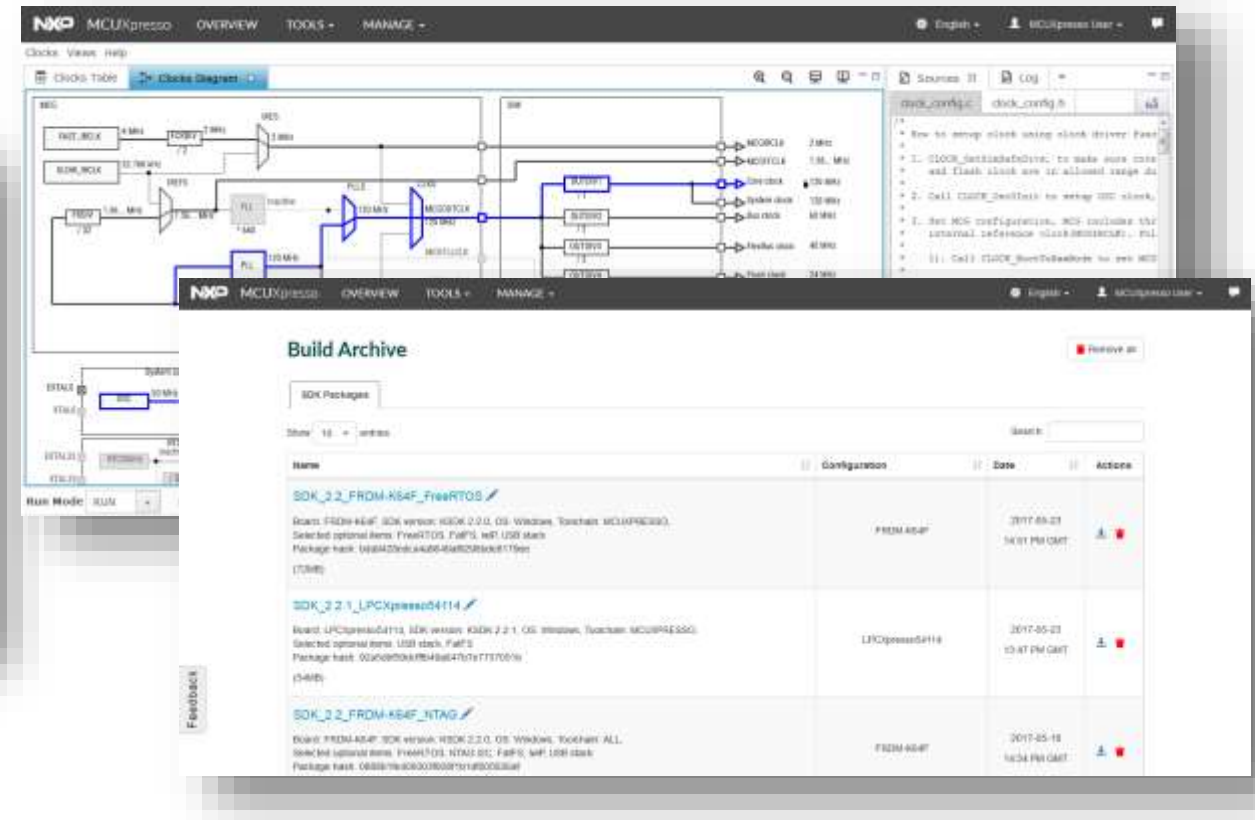


# Walk-through of tools

MCUXpresso SDK Builder, IDE, Config Tools (desktop)

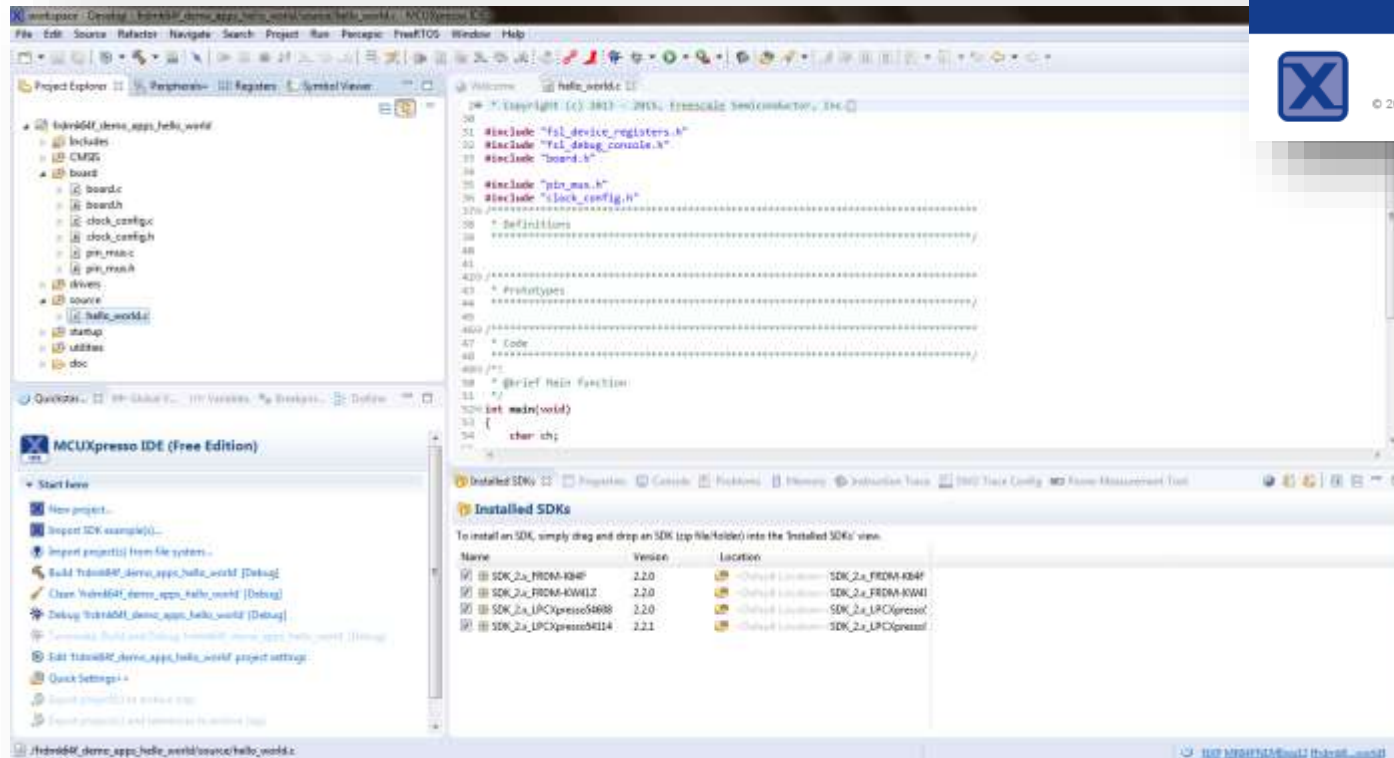
# MCUXpresso SDK Builder (<http://mcuxpresso.nxp.com>)

- MCUXpresso SDK Builder
- MCUXpresso SDK Archive
- Online MCUXpresso Config Tools



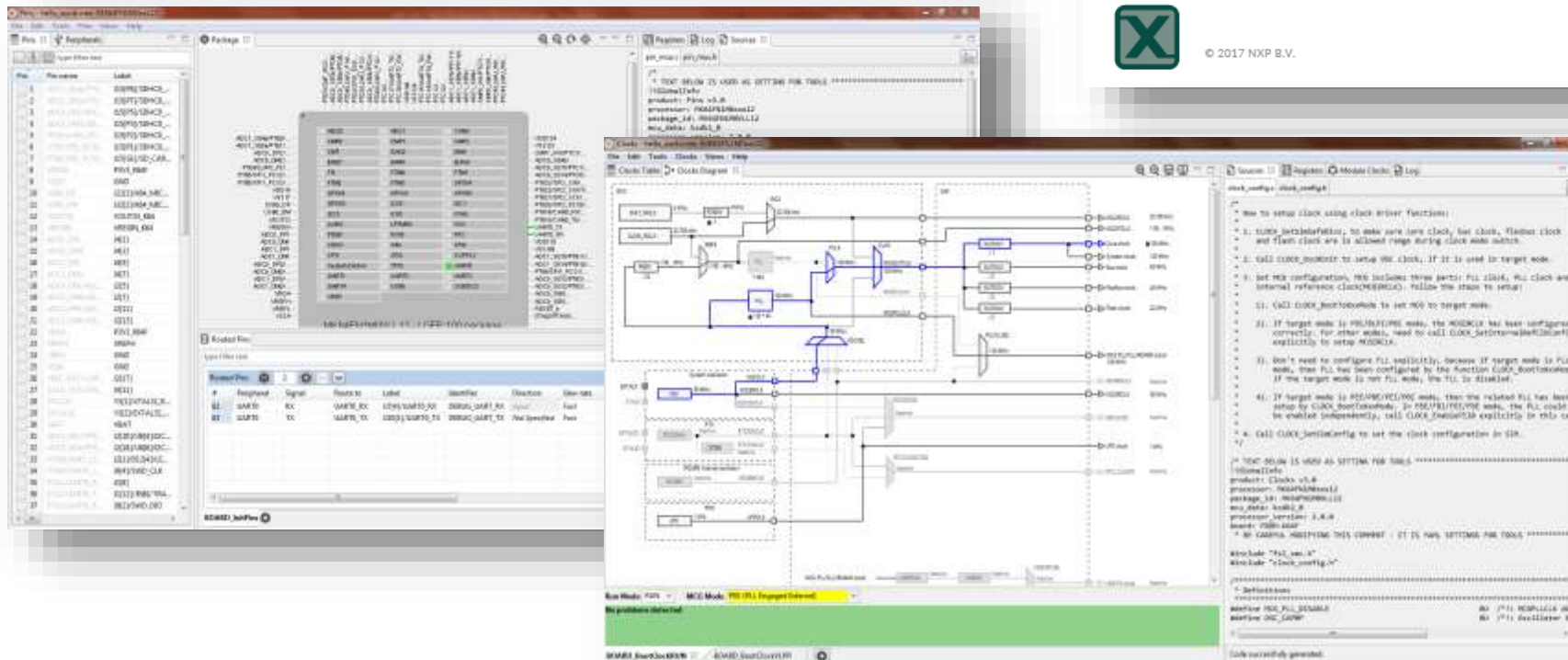
# MCUXpresso IDE

- MCUXpresso SDK Installation
- Quickstart Panel
- Development Perspective



# MCUXpresso Config Tools (Desktop Version)

- Pins and Clock Configuration
- Code Generation
- Conflict Resolution





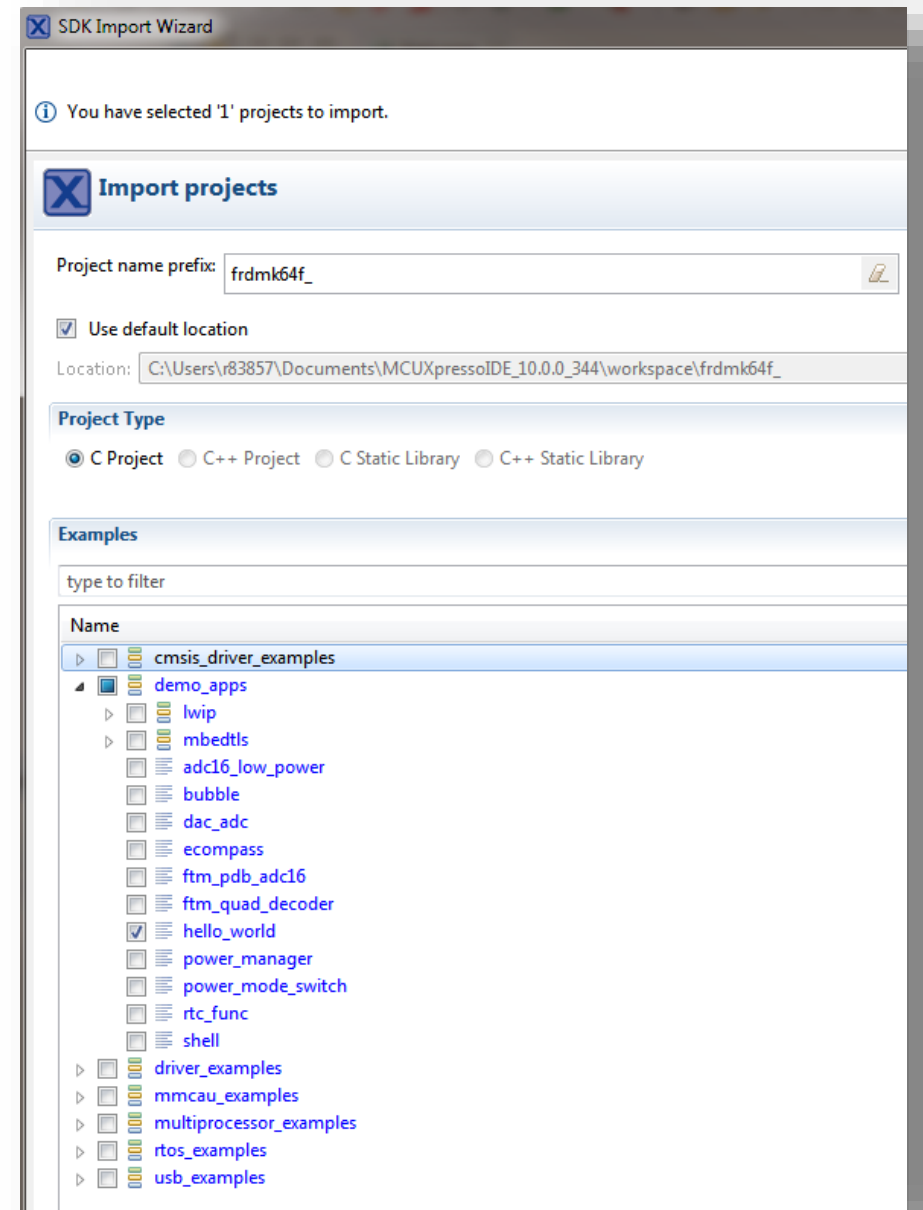


# Hands-on Lab

SDK Configuration and Build, Getting Started with IDE, Board Configuration using Config Tools

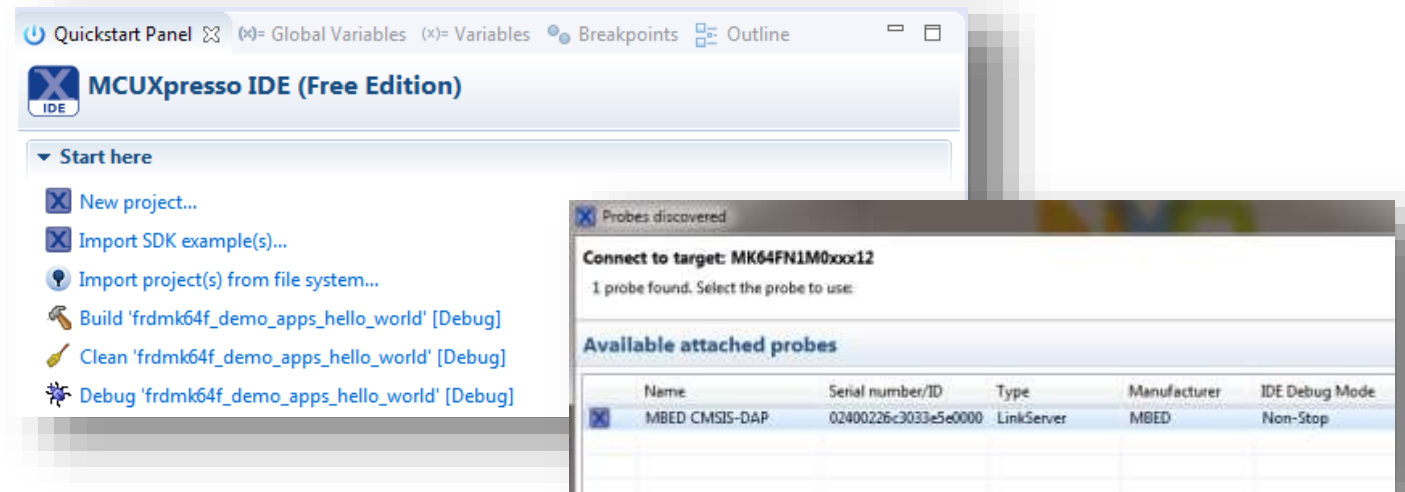
# Import Example Project

- Prerequisite:
  - Workspace: `C:\MCUXpressoSW_Lab`
  - `SDK_2.x_FRDM-K64F` installed in IDE
- Use Quickstart Panel to “Import SDK example(s)...”
- Select `frdmk64f` board image
- Locate “hello world” project (listed under `demo_apps`)
- Redirect `printf` to UART



# Hello World – Build and Debug

- Connect **FRDM-K64F** board (mini USB to **OpenSDA**)
  - Board may need to enumerate USB connection
- Connect terminal application (**Virtual COM port**)
- Use IDE **Quickstart Panel** to:
  - Clean
  - Build
  - Debug



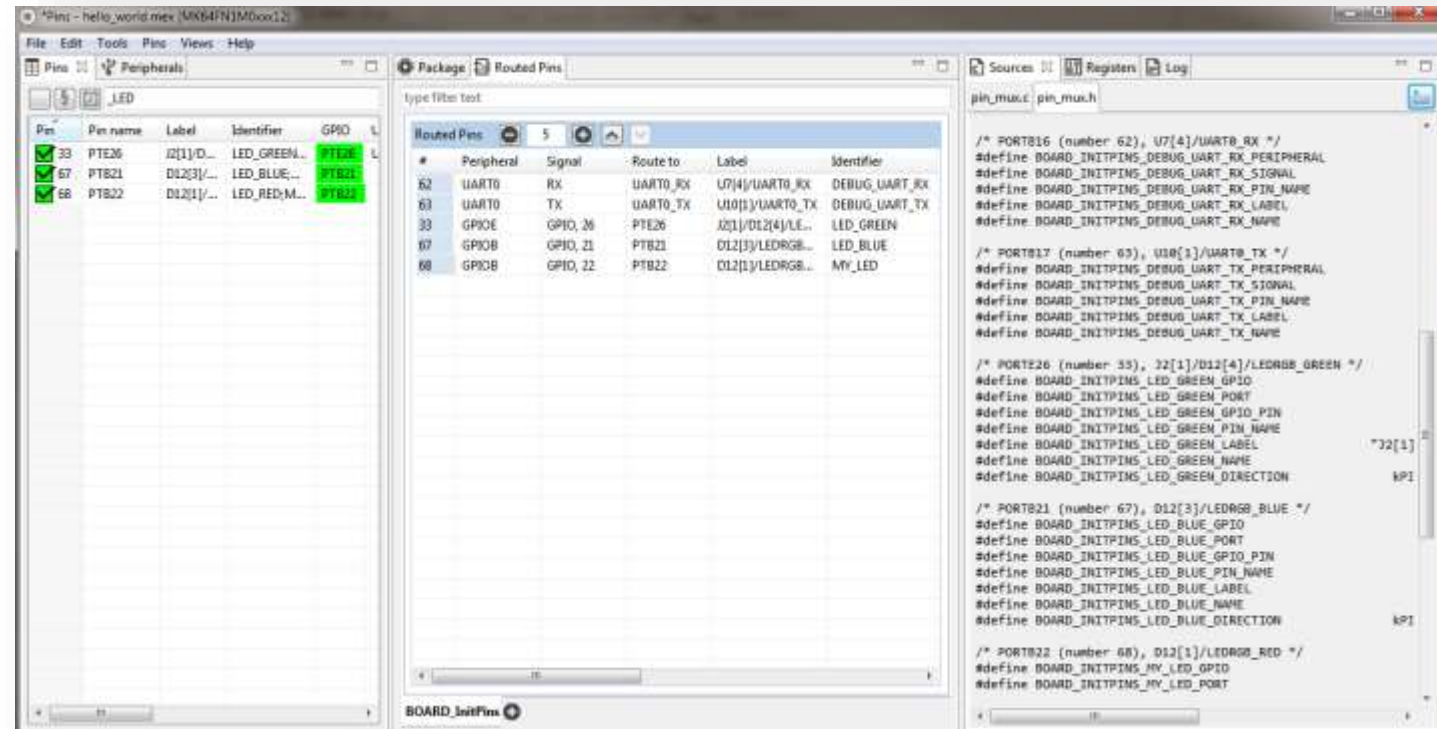
# Add SysTick Delay Timer

- Code additions to “hello\_world.c”
  - Global Variables:
    - `DelayTimerTick = 0`
  - Functions:
    - `InitSysTick(void) /* Initialize SysTick Registers */`
    - `SysTick_Handler(void) /* Override Interrupt Handler Function for SysTick */`
    - `Delay_SysTick(uint32_t SysTicks) /* Delay based on SysTick Counter */`
  - Main() Additions
    - Call `InitSysTick()`

*\*All code changes are detailed in “MCUX SW and Tools Lab.docx”*

# MCUXpresso Config Tools – Pins Configuration (Output)

- Pin muxing configuration for:
  - **PTB21**: “Blue LED”
  - **PTB22**: “Red LED”
  - **PTE26**: “Green LED”
- Routed Pins Table
  - Selection of “**MY\_LED**” identifier
- Export Generated Code
  - **pin\_mux.c** / **pin\_mux.h**



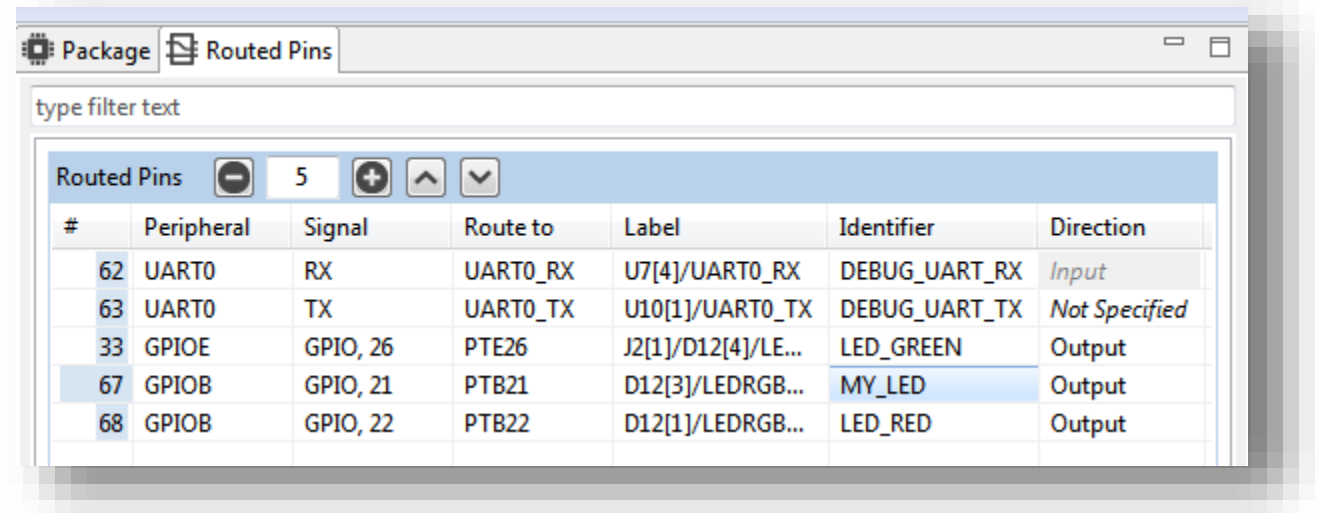
# Blinky LED

- Code additions to “hello\_world.c”
  - Peripheral Initialization
    - GPIO Driver: “fsl\_gpio.h”
    - `BOARD_InitPeripheral(void) /* GPIO Peripheral Initialization Function */`
  - User Integration (updated main while loop)
    - `Delay_SysTick /* User created delay function based on SysTick */`
    - `GPIO_TogglePinsOutput /* SDK GPIO Driver API */`
  - Main() Additions
    - Call `BOARD_InitPeripheral ()`
    - New while(1) loop

*\*All code changes are detailed in “MCUX SW and Tools Lab.docx”*

# Switch LED Pin using Config Tool

- Change “**MY\_LED**” identifier to different LED
- Export updated code
- **Refresh** (F5) in IDE
- Build and Debug



The screenshot shows the 'Routed Pins' window in an IDE. The window title is 'Package Routed Pins'. Below the title bar is a search filter labeled 'type filter text'. The main content is a table with 7 columns: '#', 'Peripheral', 'Signal', 'Route to', 'Label', 'Identifier', and 'Direction'. The table contains 5 rows of data. The row with 'MY\_LED' in the Identifier column is highlighted in blue.

#	Peripheral	Signal	Route to	Label	Identifier	Direction
62	UART0	RX	UART0_RX	U7[4]/UART0_RX	DEBUG_UART_RX	Input
63	UART0	TX	UART0_TX	U10[1]/UART0_TX	DEBUG_UART_TX	Not Specified
33	GPIOE	GPIO, 26	PTE26	J2[1]/D12[4]/LE...	LED_GREEN	Output
67	GPIOB	GPIO, 21	PTB21	D12[3]/LEDRGB...	MY_LED	Output
68	GPIOB	GPIO, 22	PTB22	D12[1]/LEDRGB...	LED_RED	Output

# MCUXpresso Config Tools – Pins Configuration (Input)

- Pin muxing configuration for:

- PTA4: “SW3”

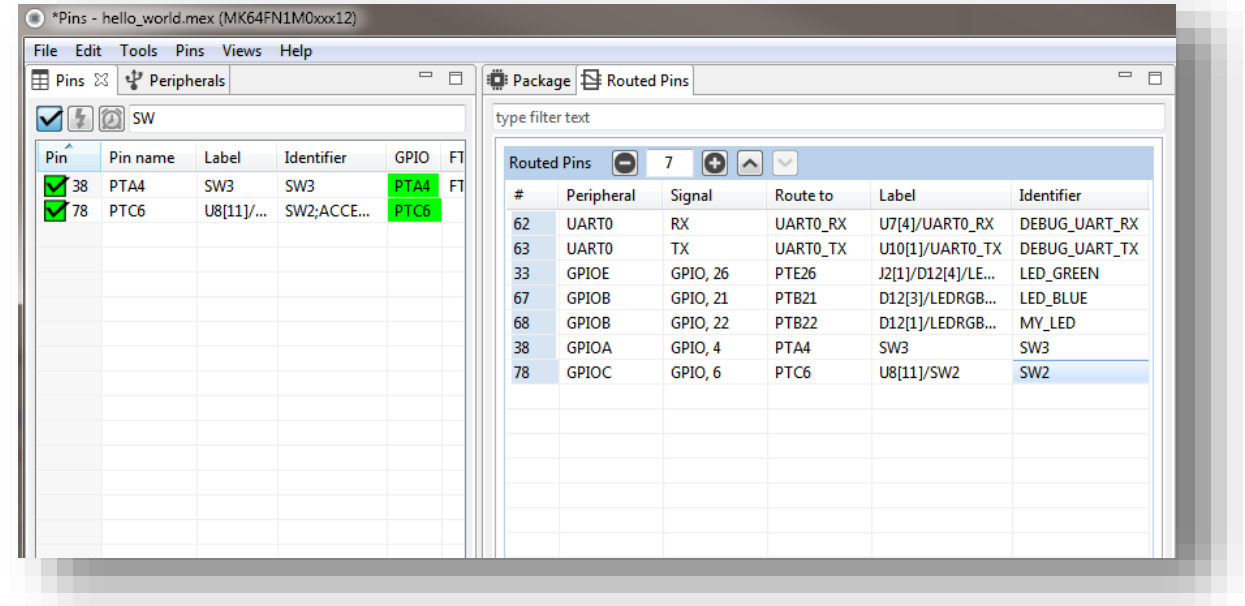
- PTC6: “SW2”

- Routed Pins Table

- Selection of “SW2” identifier

- Export Generated Code

- `pin_mux.c` / `pin_mux.h`





# Interrupt Push Button

- Code additions to “hello\_world.c”
  - Peripheral Initialization
    - Update `BOARD_InitPeripheral(void) /* GPIO Peripheral Initialization Function */`
    - GPIO Driver: “`fsl_port.h`”
  - Global Variable:
    - `g_count = 0 /* Global Counter for pushbutton */`
  - Functions:
    - `PORTC_IRQHandler(void) /* User created delay function based on SysTick */`

*\*All code changes are detailed in “MCUX SW and Tools Lab.docx”*

# MCUXpresso Config Tools – Clock Configurations

- Changing Core Clock
  - 60 MHz?
  - 64 MHz?
- Export Generated Code
  - `clock_config.c` / `clock_config.h`
- Additional clock mode functions (VLPR – 4MHz)

The screenshot displays the MCUXpresso Config Tools interface. The main window shows a clock configuration diagram with various clock sources and dividers. The 'Processor Details' panel on the right provides a comprehensive list of clock-related parameters and their current settings.

Name	Value	Ac.y
Core clock	120 MHz	±0.1%
System clock	120 MHz	
Bus clock	60 MHz	
FlexBus clock	40 MHz	
Flash clock	24 MHz	
MCGIRCLK	32,768 kHz	
MCGFFCLK	1.56... MHz	
OSCCERCLK	50 MHz	
ERCLK32K	Inactive	
RTC_CLKOUT	Inactive	
Initialize RTC_CLKOUT	yes	
MCG-PLL/FLL/IRC48M clock	120 MHz	
LPO clock	1 kHz	
IRC48MCLK	Inactive	
USB FS clock	Inactive	
Initialize USB clock	no	
Trace clock input	Inactive	
Initialize Trace clock	no	
ENET IEEE 1588 timestamp clock	Inactive	
Initialize ENET 1588 clock	no	
ENET RME clock	Inactive	
Initialize RME clock	no	
SDHC clock	Inactive	
Initialize SDHC clock	no	
CLKOUT(FB_CLK)	Inactive	
Initialize CLKOUT	no	
OSC (System Oscillator)	<input checked="" type="checkbox"/>	
OSC mode	Using external reference clock	
Frequency Range	Very_high frequency range 8-32 MHz	
System Osc. Capacity Load	0 pF	
OSCCERCLK Frequency	50 MHz	
OSCCERCLK output	Enabled	
OSCCERCLK out...n Stop mode	Disabled	
OSCCCLK Frequency	50 MHz	
OSC32KCLK Frequency	Inactive	
RTC32KHz	<input type="checkbox"/>	
RTC 32KHz oscillator	Enabled	
RTC32KCLK output	Enabled	



**MCUXpresso**  
Software and Tools

COMMON TOOLKIT  
FOR THOUSANDS  
OF KINETIS® & LPC  
MICROCONTROLLERS



[www.nxp.com/mcuxpresso](http://www.nxp.com/mcuxpresso)

# MCUXpresso Software and Tools Additional Resources

## Web pages

- MCUXpresso Software and Tools – [www.nxp.com/mcuxpresso](http://www.nxp.com/mcuxpresso)
- MCUXpresso SDK – [www.nxp.com/mcuxpresso/sdk](http://www.nxp.com/mcuxpresso/sdk)
- MCUXpresso IDE – [www.nxp.com/mcuxpresso/ide](http://www.nxp.com/mcuxpresso/ide)
- MCUXpresso Config Tools – [www.nxp.com/mcuxpresso/config](http://www.nxp.com/mcuxpresso/config)

**Supported Devices:** [Supported Devices Table \(Community Doc\)](#)

## Communities

- MCUXpresso Software and Tools - <https://community.nxp.com/community/mcuxpresso>
- MCUXpresso SDK- <https://community.nxp.com/community/mcuxpresso/mcuxpresso-sdk>
- MCUXpresso IDE- <https://community.nxp.com/community/mcuxpresso/mcuxpresso-ide>
- MCUXpresso Config Tools - <https://community.nxp.com/community/mcuxpresso/mcuxpresso-config>



SECURE CONNECTIONS  
FOR A SMARTER WORLD