

Crossover Code Challenge: Getting Started with the i.MX RT1010 EVK and MCUXpresso

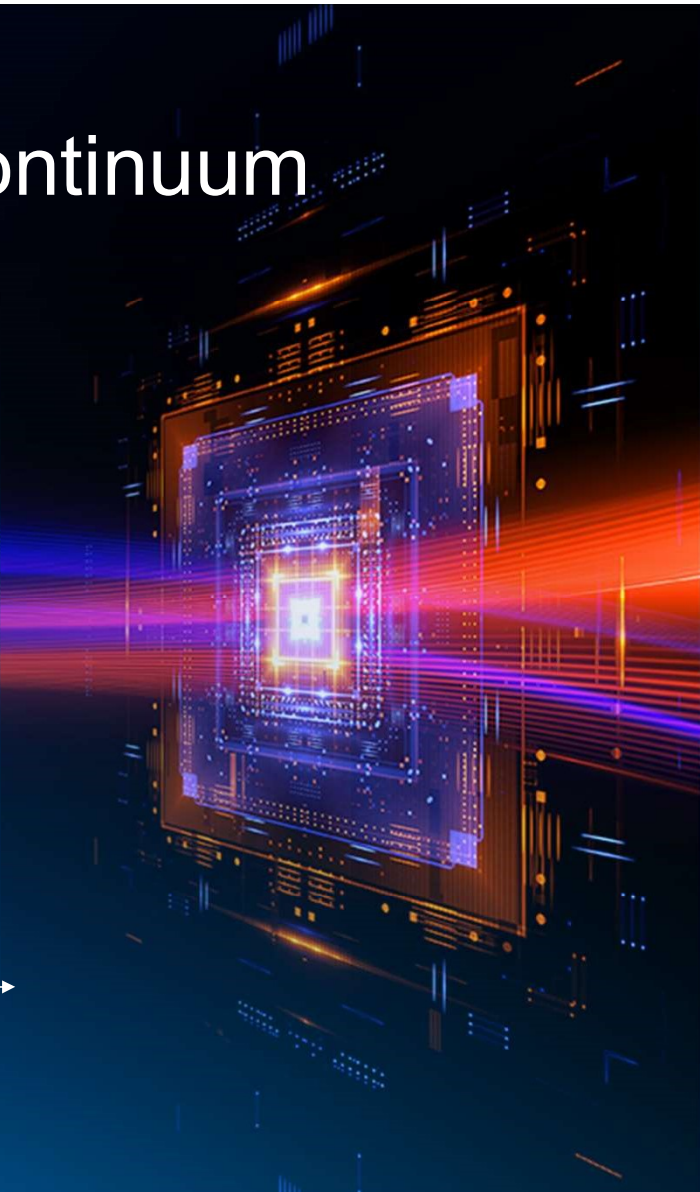
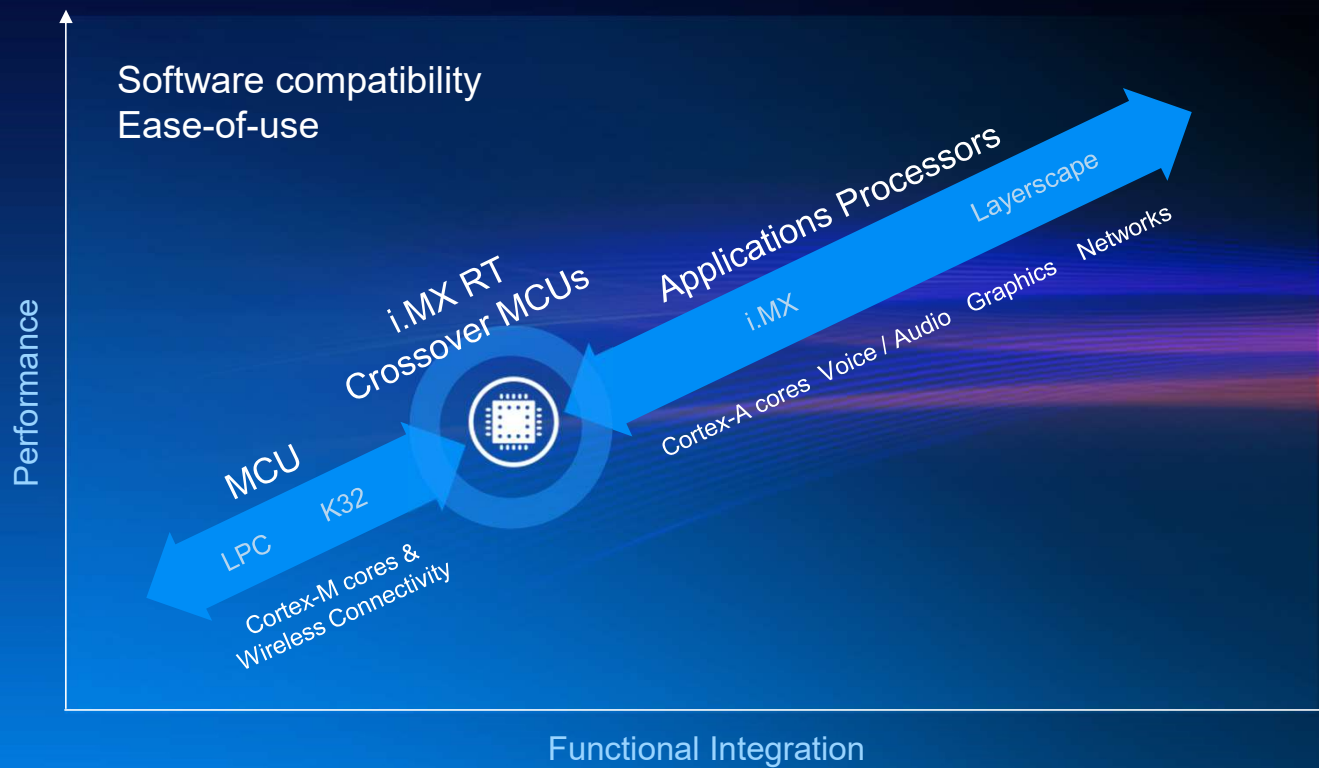
Juan Mendoza, Systems Engineering Manager, NXP
Clark Jarvis, MCUXpresso Product Manager, NXP



December 10, 2019



NXP Scalable Edge Processing Continuum



i.MX RT Crossover MCUs – Target Applications

Voice & Audio

High-end, consumer audio devices, including specialty equipment such as:

- Home Audio
- Professional microphone
- Guitar pedals
- Audio Tuners



Consumer & Healthcare

- Wearables
- Smart appliances
- TVs
- Game Controllers
- Mobile patient care, Blood pressure monitor
- Activity & wellness monitor
- Exercise equipment with display



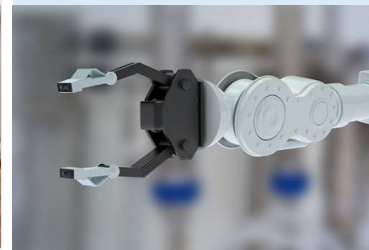
Home & Building Automation

- HVAC climate control
- Security
- Lighting control panels
- IoT gateways
- Smoke Detectors
- Thermostats



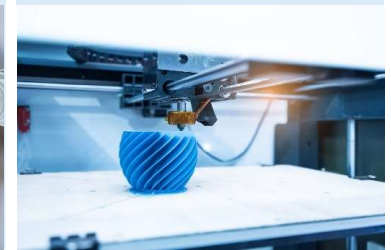
Industrial Computing

- EBS
- PLCs
- Factory automation
- Test and measurement
- HMI control assembly line robotics
- QR Readers
- Barcode Scanners



Motor Control & Power Conversion

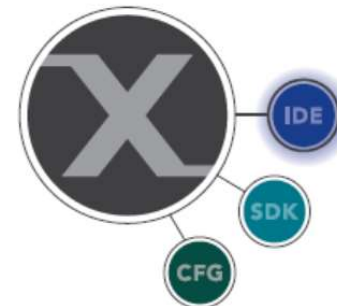
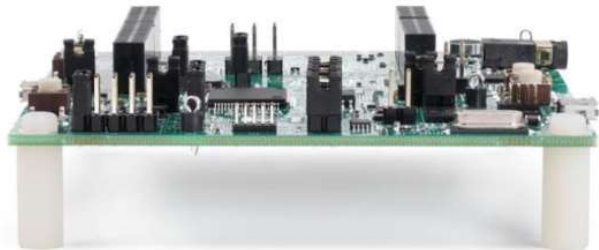
- 3D printers
- Thermal printers
- Unmanned autonomous vehicles
- Robotic vacuum cleaners
- Drones



Crossover Code Challenge



Objective: Develop example software code for multiple features of the i.MX RT1010 crossover MCU.



Features of the i.MX RT1010 Crossover MCU and i.MX RT1010 EVK

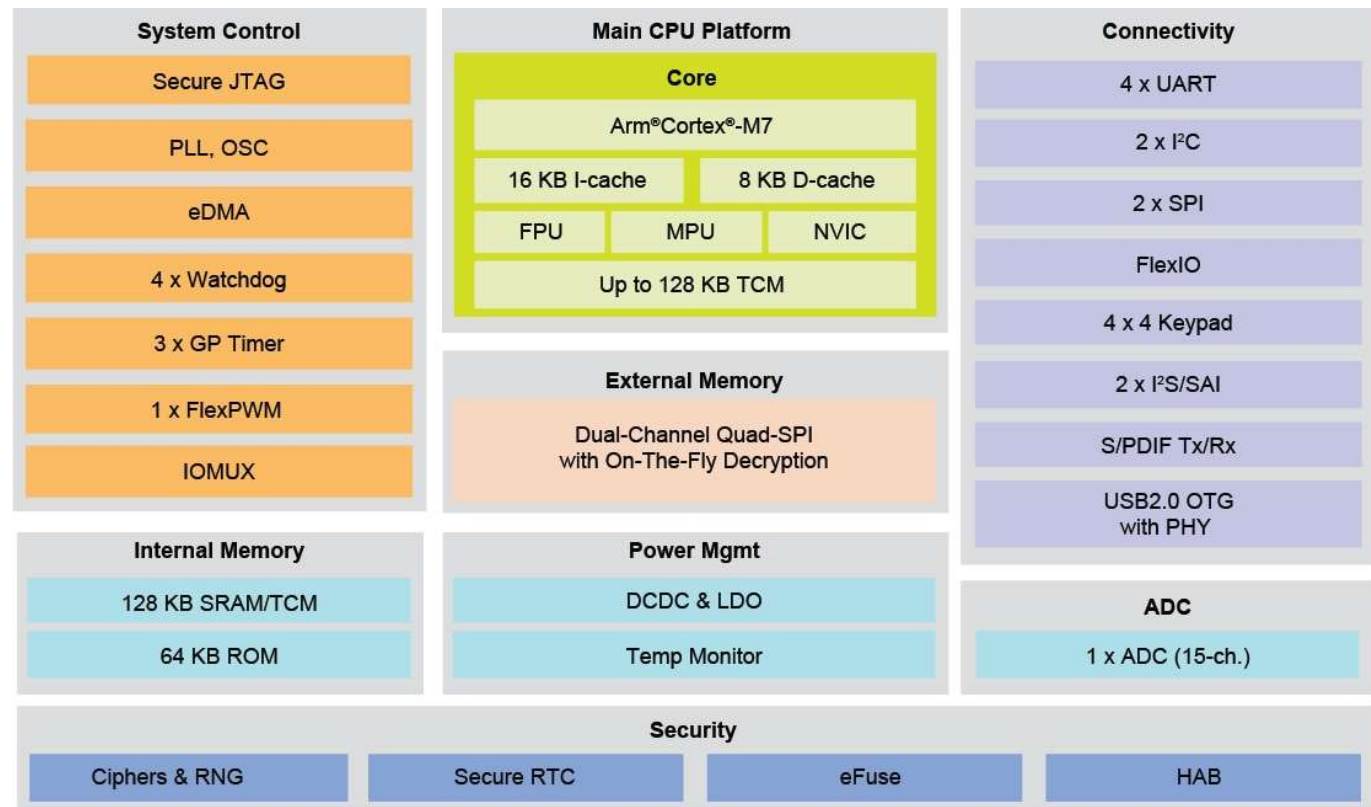
Juan Mendoza, Systems Engineering Manager, NXP



i.MX RT1010 SoC

• Features

- 500Mhz Cortex M7
- 128K SRAM
- FlexIO (Displays)
- Security
- Rich Audio Features
- Motor Control



i.MX RT1010 Evaluation Kit (MIMXRT1010-EVK)

FEATURES

Memory

- 128Mb QSPI Flash

Audio

- Wolfson Stereo Codec
- 3.5mm headphone jack
- Microphone
- L/R Speaker out connections
- S/PDIF (coax, not populated)

Connectivity

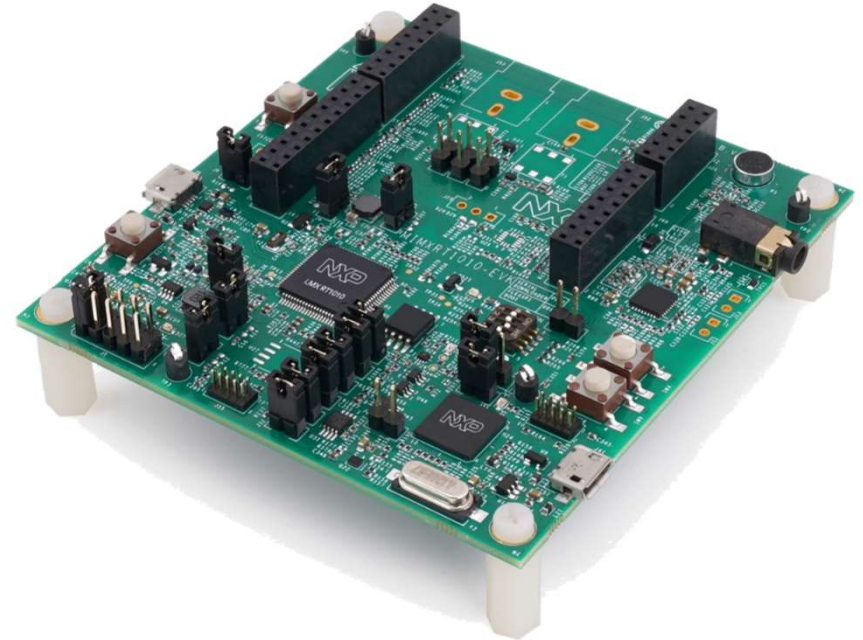
- HS USB
- Arduino® interface
- Motor Control Expansion port

Software

- MCUXpresso SDK with optional Amazon FreeRTOS support
- MCUXpresso IDE
- MCUXpresso Configuration Tools
- Arm Mbed™ OS
- Zephyr® OS

RESOURCES

- **SoC reference material**
 - [i.MXRT1010RM](#)
 - [Documentation Application-Notes](#)
- **Users Guides**
 - [Quick Start Guide – i.MX RT1010 Evaluation Kit](#)
 - [MIMXRT1010 Evaluation Kit Hardware User's Guide](#)
 - [Getting Started with i.MX RT1010 Evaluation Kit](#)
- **Design Resources** – [i.MX RT1010 EVK Design Files](#), Schematics



How To Get Started with MCUXpresso Software and Tools on i.MX RT1010 EVK

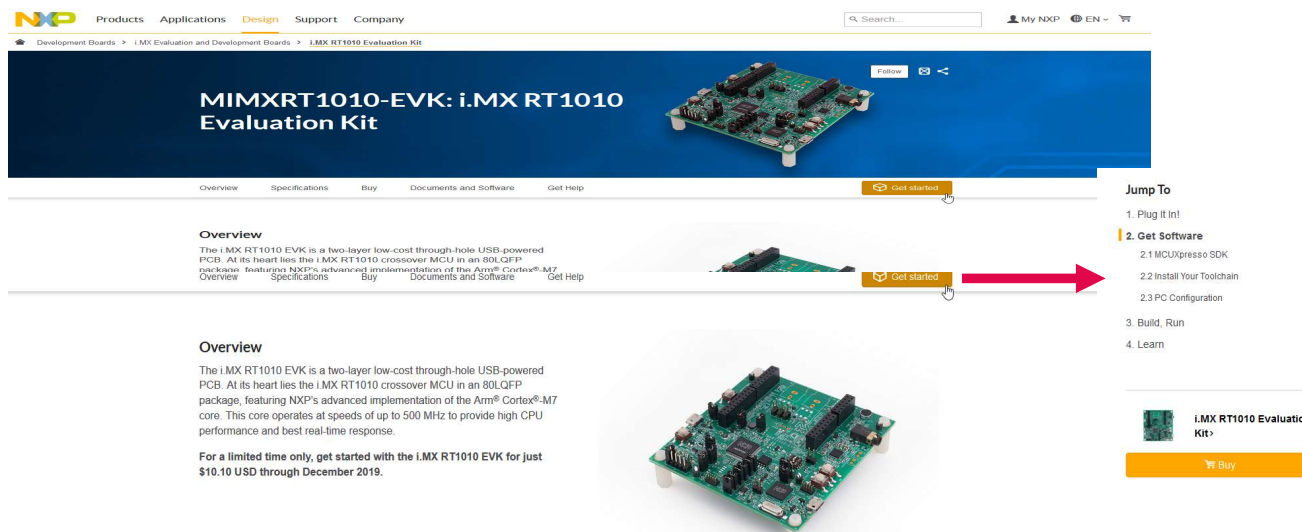
Clark Jarvis, MCUXpresso Product Manager, NXP



MCUXpresso SW and Tools

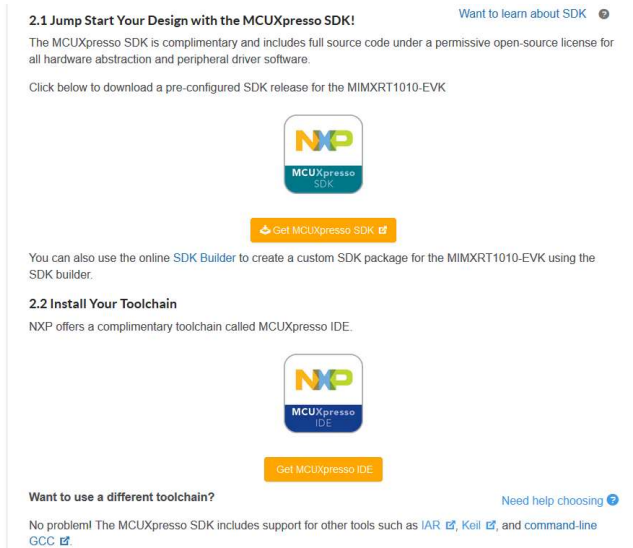
Getting Started Guide for i.MX RT1010

<https://www.nxp.com/MIMXRT1010-EVK>



Includes:

- Getting started videos
- Links to MCUXpresso IDE and MCUXpresso SDK
- Basic examples walk-through



MCUXpresso SW and Tools

Download and Install SDK for i.MX RT1010

<https://mcuxpresso.nxp.com/en/select?device=EVK-MIMXRT1010>

SDK Builder
Generate a downloadable SDK archive for use with desktop MCUxpresso Tools.

Developer Environment Settings
Selections here will impact files and examples projects in

Host OS: Windows
SDK Version: 2.6.0 2019-10-08

Select Optional Middleware
Add middleware, operating systems, and software libraries

MCUXpresso IDE
Add software component

Online Documentation
View SDK API Reference Manual
Download Config Tools data

Downloads
Download SDK Archive (18 MB)
Download SDK Documentation
Download Standalone Example Project

MCUXpresso SDK
For MCUxpresso IDE, example projects can be imported as standalone projects directly within the IDE by downloading the SDK Archive

Hardware Details
EVK-MIMXRT1010
MIMXRT1011
Cortex-M7F / 500MHz
0 KB Flash
128 KB RAM
2.6.0 (released 2019-10-08)

MCUXpresso IDE - Quickstart Panel
To install an SDK, simply drag and drop an SDK (zip file) into the "Installed SDKs" view. [Context menu "Import SDK" is available.]

Installed SDKs
Name Version
[List of SDKs and versions]

- As desired add additional software component
- Download SDK and Documentation
- Drag-and-Drop downloaded archive into IDE

MCUXpresso SW and Tools

Import example application

Quickstart Panel -> Import SDK example(s)...

The image illustrates the process of importing an SDK example application into the MCUXpresso IDE. It consists of four main panels:

- MCUXpresso IDE - Quickstart Panel:** Shows the 'Create or import a project' section with the option 'Import SDK example(s)...' highlighted by a red arrow.
- Board and/or Device selection page:** Displays a list of SDK MCUs. The 'MIMXRT1010' series is selected, and a red arrow points to the 'evkmimxrt1010' board image.
- Import projects dialog:** Shows the 'Project name prefix' as 'evkmimxrt1010' and the 'Project name suffix' as an empty field. The 'Use default location' checkbox is checked. The 'Project Type' is set to 'C Project'. The 'Project Options' section has 'SDK Debug Console' checked, 'Semihost' unchecked, 'UART' selected, and 'Example default' unchecked. The 'Copy sources' and 'Import other files' checkboxes are also checked.
- Examples:** A table listing available examples. The 'hello_world' example is selected, indicated by a red arrow.

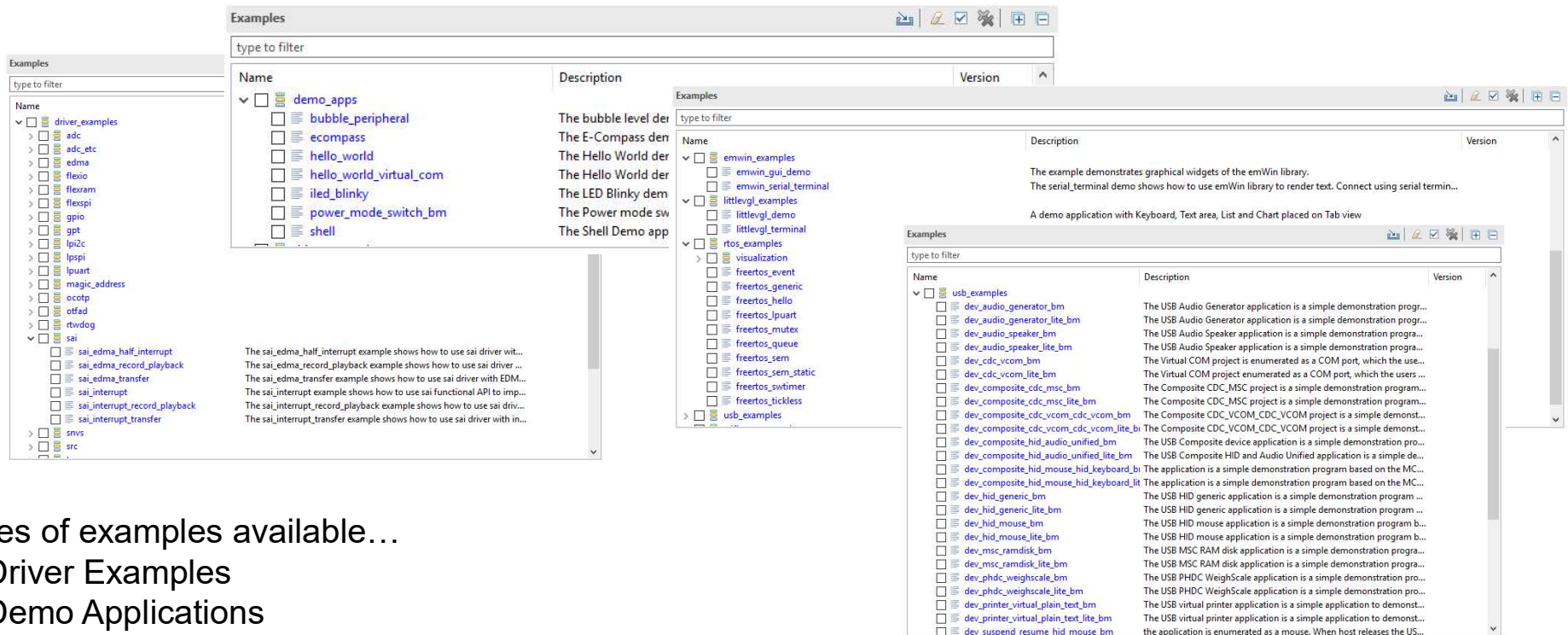
| Name | Description |
|-------------------------|---|
| > cmsis_driver_examples | |
| ✓ demo_apps | |
| bubble_peripheral | The bubble level demo demonstrates basic usage of the on-board accelerometer to in |
| ecompass | The E-Compass demo application demonstrates the use of the FXOS8700 sensor. The |
| hello_world | The Hello World demo application provides a sanity check for the new SDK build envi |
| hello_world_virtual_com | The Hello World demo application provides a sanity check for the new SDK build envi |
| iled_blinky | The LED Blinky demo application provides a sanity check for the new SDK build envi |
| power_mode_switch_bm | The Power mode switch demo application demonstrates the use of power modes in t |
| shell | The Shell Demo application demonstrates to control Leds by commands. |
| ✓ driver_examples | |
| adc | |
| adc_etc | |
| adma | |

Steps

- Import (copy) an existing project based on MIMXRT1010-EVK
- Build and Debug basic “Hello World” application
- Step-through application and observed console output

MCUXpresso SW and Tools

Library of example application



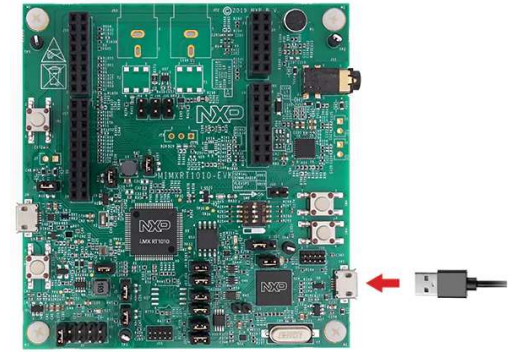
Types of examples available...

- Driver Examples
- Demo Applications
- FreeRTOS Examples
- Software Specific (Graphics, Wi-Fi, USB)

MCUXpresso SW and Tools

Build / Debug Example Application in IDE

Flash and Debug Application on Board

The image shows the MCUXpresso IDE interface with several components:

- Quickstart Panel:** On the left, it lists actions like 'New project...', 'Import SDK example(s)...', and 'Build your project'. Red circles with numbers 1 and 2 highlight the 'Build' and 'Debug' buttons respectively.
- CDT Build Console:** The central pane displays the output of the build process, including compiler flags, memory region usage, and the final build completion message: '12:27:23 Build Finished. 0 errors, 0 warnings. (took 18s.396ms)'.
- Debug Console:** The right pane shows the debug session details, including the target 'MyLED LinkServer Debug' and the current execution point in the 'main()' function.
- Source Code:** The bottom pane shows the 'MyLED.c' source file with the 'main()' function. A red arrow points from the 'PRINTF("Hello World\\n");' line in the code to the serial terminal.
- Serial Terminal:** A window titled 'COM3 - PuTTY' displays the output 'Hello World!'.

MCUXpresso SW and Tools

Create new IDE project

Quickstart Panel -> New Project...

Quickstart P... (*)= Variables Breakpoints

MCUXpresso IDE - Quickstart Panel
Project: MyBlinkingLED [Debug]

Create or import a project

- New project...
- Import SDK example(s)...
- Import project(s) from file system...

Build your project

- Build
- Clean

Debug your project

- Debug
- Terminate, Build and Debug

Board and/or Device selection page

SDK MCUs

MCUs from installed SDKs.
Please visit mcuxpresso.nxp.com to obtain additional SDKs.

- NXP MIMXRT1011xxxx
- > LPC55xx
- > LPC804
- > LPC84x
- > MIMXRT1010
- > MIMXRT1011xxxx
- > MIMXRT1050
- > MIMXRT1060
- > MIMXRT1064

Preinstalled MCUs

MCUs from preinstalled LPC and

Available boards

Please select an available board for your project

Supported boards for device: MIMXRT1011x



evkmimxrt1010

SDK

Configure the project

Project name: MyLED Project name suffix:

☒ Use default location
Location: C:\Users\mxa08675\Documents\MCUXpressoIDE_11.0.1_2563\workspace\hackster\MyLED

Device Packages: ☐ MIMXRT1011CA4A ☒ MIMXRT1011DAE5A

Board: ☒ Default board files ☐ Empty board files

Project Type: ☒ C Project ☐ C++ Project ☐ C Static Library ☐ C++ Static Library

Project Options: ☐ SDK Debug Console ☒ Semihost ☐ UART ☒ CMSIS-Core ☒ Copy sources ☒ Import other files

Components: Add or remove SDK soft

Operating Systems: ☒ pit

Advanced project settings

C/C++ Library Settings: Set library type (and hosting variant) Redlib (semihost-rt) ☐ Redlib: Use floating point version of printf ☐ Redlib: Use character rather than string based printf ☒ Redirect SDK "PRINTF" to C library "printf" ☒ Include semihost HardFault handler ☐ NewlibNano: Use floating point version of printf ☐ NewlibNano: Use floating point version of scanf ☐ Redirect printf/scanf to ITM ☐ Redirect printf/scanf to UART

Hardware settings: Set Floating Point type FPU5-SP-D15 (Hard ABI)

MCU C Compiler: Language standard Compiler default

MCU Linker: ☐ Link application to RAM

Memory Configuration: Memory details

| Type | Name | Alias | Location | Size | Driver |
|-------|---------------|-------|------------|-----------|--------------------------|
| Flash | BOARD_FLASH | Flash | 0x60000000 | 0x1000000 | MIMXRT1010_SFDP_QSPI.cfx |
| RAM | SRAM_ITC | RAM1 | 0x20000000 | 0x8000 | |
| RAM | SRAM_ITC | RAM2 | 0x0 | 0x8000 | |
| RAM | SRAM_OC | RAM3 | 0x20200000 | 0x8000 | |
| RAM | NCACHE_REGION | RAM4 | 0x20208000 | 0x8000 | |

Buttons: Add Flash, Add RAM, Split, Join, Delete, Import..., Merge..., Export..., Generate...

Navigation: < Back Next > Finish Cancel

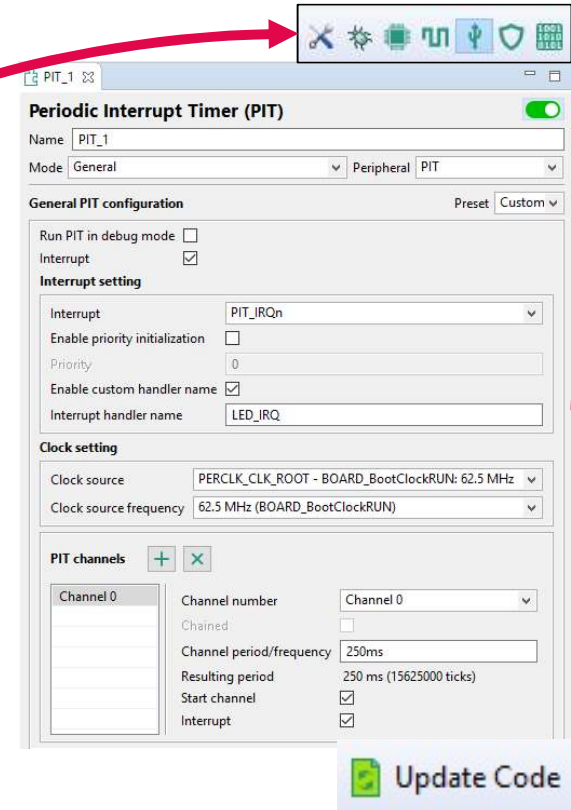
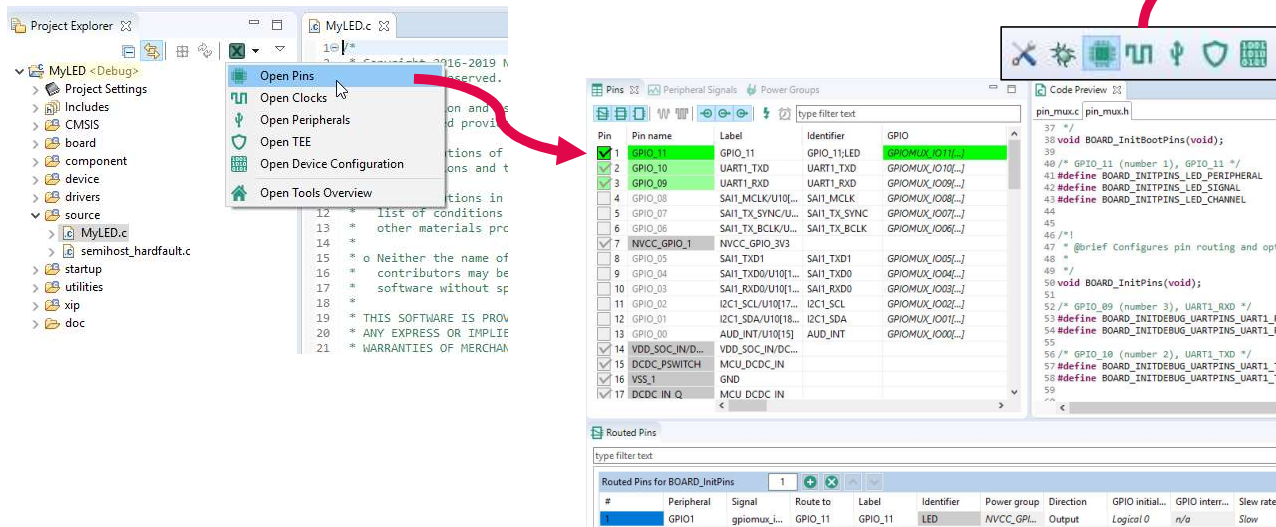
Steps

- Create a new project based on MIMXRT1010-EVK
- Add SDK driver for Periodic Interrupt Timer (PIT)
- Review new project settings

MCUXpresso SW and Tools

Generate code with Config Tool Perspective

Project -> Open Pins (then Peripheral Tool)



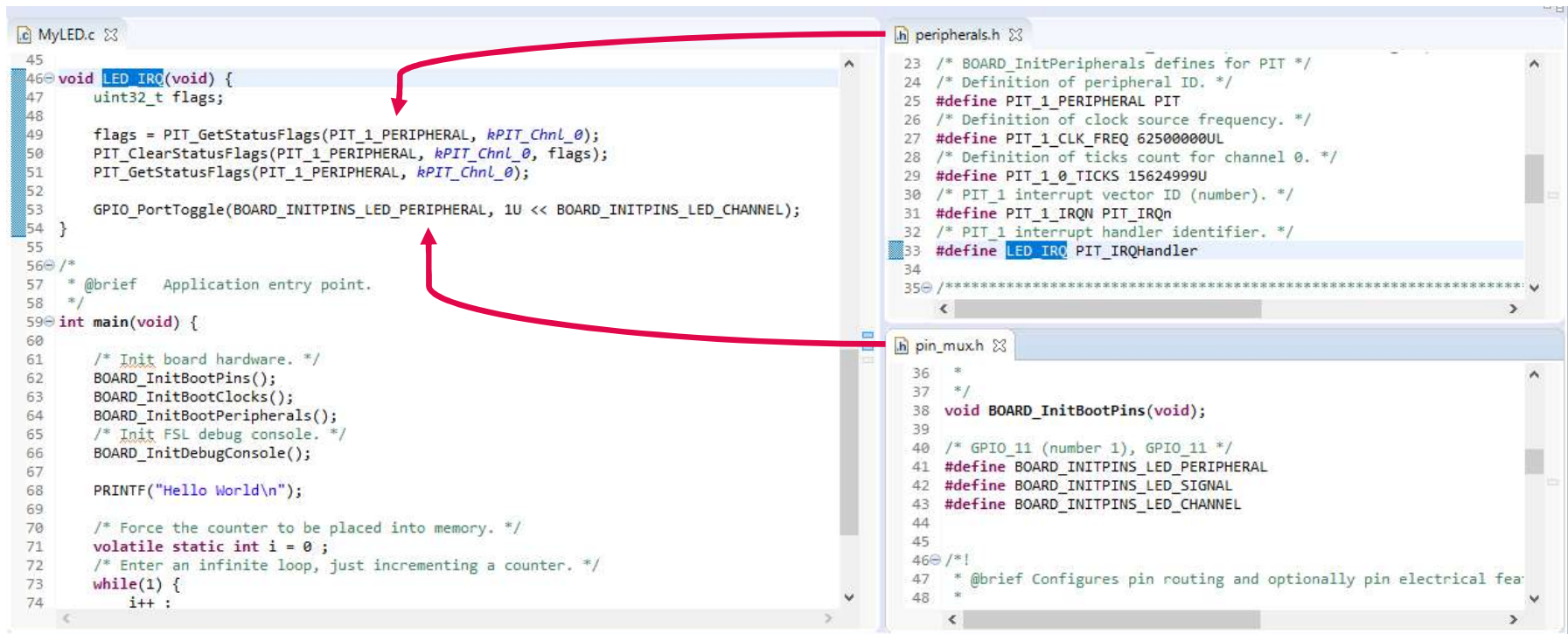
Steps

- Configure LED (Pin 1) as output with custom identifier
- Add PIT peripheral with 250ms frequency and Interrupt
- "Update Code" to generate configuration files and return to Development Perspective

MCUXpresso SW and Tools

Create an interrupt function for the LED

Create IRQ function to Blink LED



```
MyLED.c
45
46 void LED_IRQ(void) {
47     uint32_t flags;
48
49     flags = PIT_GetStatusFlags(PIT_1_PERIPHERAL, kPIT_Chnl_0);
50     PIT_ClearStatusFlags(PIT_1_PERIPHERAL, kPIT_Chnl_0, flags);
51     PIT_GetStatusFlags(PIT_1_PERIPHERAL, kPIT_Chnl_0);
52
53     GPIO_PortToggle(BOARD_INITPINS_LED_PERIPHERAL, 1U << BOARD_INITPINS_LED_CHANNEL);
54 }
55
56 /*
57  * @brief Application entry point.
58  */
59 int main(void) {
60
61     /* Init board hardware. */
62     BOARD_InitBootPins();
63     BOARD_InitBootClocks();
64     BOARD_InitBootPeripherals();
65     /* Init FSL debug console. */
66     BOARD_InitDebugConsole();
67
68     PRINTF("Hello World\n");
69
70     /* Force the counter to be placed into memory. */
71     volatile static int i = 0;
72     /* Enter an infinite loop, just incrementing a counter. */
73     while(1) {
74         i++;
```

```
peripherals.h
23 /* BOARD_InitPeripherals defines for PIT */
24 /* Definition of peripheral ID. */
25 #define PIT_1_PERIPHERAL PIT
26 /* Definition of clock source frequency. */
27 #define PIT_1_CLK_FREQ 62500000UL
28 /* Definition of ticks count for channel 0. */
29 #define PIT_1_0_TICKS 15624999U
30 /* PIT_1 interrupt vector ID (number). */
31 #define PIT_1_IRQN PIT_IRQN
32 /* PIT_1 interrupt handler identifier. */
33 #define LED_IRQ PIT_IRQHandler
34
35 /******
```

```
pin_mux.h
36 *
37 */
38 void BOARD_InitBootPins(void);
39
40 /* GPIO_11 (number 1), GPIO_11 */
41 #define BOARD_INITPINS_LED_PERIPHERAL
42 #define BOARD_INITPINS_LED_SIGNAL
43 #define BOARD_INITPINS_LED_CHANNEL
44
45
46 /*!
47  * @brief Configures pin routing and optionally pin electrical fea
48  *
```

MCUXpresso SW and Tools

Build / Debug MyLED Application in IDE

Flash and Debug Application on Board



MCUXpresso IDE - Quickstart Panel

Project: MyLED [Debug]

Create or import a project

- New project...
- Import SDK example(s)...
- Import project(s) from file system...

Build your project

- Build
- Clean

Debug your project

- Debug
- Terminate, Build and Debug

CDT Build Console [MyLED]

```
arm-none-eabi-gcc -DCPU_MIMXRT1011DAE5A -DCPU_MIMXRT1011DAE5A_c
Finished building: ../board/clock_config.c

Finished building: ../board/peripherals.c

Finished building: ../board/pin_mux.c

Building target: MyLED.axf
Invoking: MCU Linker
arm-none-eabi-gcc -nostdlib -Xlinker -Map="MyLED.map" -Xlinker
Memory region      Used Size  Region Size  %age Used
BOARD_FLASH:       29080 B      16 MB         0.17%
SRAM_DTC:           4360 B       32 KB        13.31%
SRAM_ITC:            0 GB       32 KB         0.00%
SRAM_OC:            0 GB       32 KB         0.00%
NCACHE_REGION:     0 GB       32 KB         0.00%
Finished building target: MyLED.axf

make --no-print-directory post-build
Performing post-build steps
arm-none-eabi-size "MyLED.axf"; # arm-none-eabi-objcopy -v -O b:
text    data    bss    dec     hex filename
29076     4    4356   33436   829c MyLED.axf

12:27:23 Build Finished. 0 errors, 0 warnings. (took 18s.396ms)
```

Debug

MyLED LinkServer Debug [C/C++ (NXP Semiconductors) MCU Application]

- MyLED.axf [MIMXRT1011xxxxx (cortex-m7)]
 - Thread #1 1 (Suspended : Breakpoint)
 - main() at MyLED.c:62 0x6000239c
- arm-none-eabi-gdb (8.2.50.20181213)

MyLED.c

```
59 int main(void) {
60
61     /* Init board hardware. */
62     BOARD_InitBootPins();
63     BOARD_InitBootClocks();
64     BOARD_InitBootPeripherals();
65     /* Init FSL debug console. */
66     BOARD_InitDebugConsole();
67
68     PRINTF("Hello World\n");
69
70     /* Force the counter to be placed into memory. */
71     volatile static int i = 0;
72     /* Enter an infinite loop, just incrementing a counter. */
73     while(1) {
```




MCUXpresso Software and Tools

Additional Resources

Web pages

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

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>

Supported Devices

- [Supported Devices Table \(Community Doc\)](#)

Get Started Now

- **Pro Tips:** use the examples project provided with the SDK:
 - **Device examples** are excellent tutorial projects for how to use the respective peripheral for a basic use case.
 - **Demo Applications** provide examples of multiple drivers, middleware, and other available code packages working together.
 - Other **software specific examples** provide basic use cases of a particular software package that is included directly within the SDK as an enabling software technology.
- Register for the contest - <https://www.hackster.io/contests/nxpcrossover>
- How to post to GitHub (general guidelines) - <https://help.github.com/en/github/getting-started-with-github/create-a-repo>
- Get technical help –
 - <https://www.hackster.io/contests/nxpcrossover>
 - <https://community.nxp.com/community/imxrt>



Questions?



SECURE CONNECTIONS
FOR A SMARTER WORLD