

使用 MCUXpresso Config Tool 生成独立 SDK 工程并适配不同芯片

Ji Cheng/May Li 2020.10

North China CAS Team

NXP 通过 MCUXpresso 网站为客户提供完整的 SDK 开发包来方便其评估使用，不过 SDK 提供的例程都是基于 NXP 官方开发板运行和测试的，那客户如果实际项目中选择的芯片非开发板板载芯片型号时（虽然与其是同一个产品家族，同源不同型，可以理解为一个 Die 出来的，只是 Mark 掉不同的 feature），则往往需要客户手动去做移植，这个对刚开始接触 NXP 平台的用户来讲有点难度，本文档先通过 MCUXpresso Config Tool 先 clone 一个独立且无 SDK 路径依赖的工程，然后以该工程为基础移植到用户自己的芯片上来。

测试环境：

SDK 版本：SDK_v2.8.0

IDE 工具：Keil_v5.31

MCUXpresso Config Tool 版本：v8.0

目标芯片：MKL17Z128VFM4

* 之所以选择 MKL17Z128 是因为其与 MKL43 是同一个家族且官方开发板为 FRDM-KL43，这个差别更大些，以此为例方便用户更清楚了解移植步骤，学会了这个那其他芯片的移植就更简单了。注意 KL17Z32/Z64 跟 KL27Z32/Z64 同源，他们的官方开发板为 FRDM-KL27，而 128KB 以上的 KL17 则与 KL43 同源，其开发板为 FRDM-KL43，这个一定别弄错了。

移植步骤：

1. 首先需要登录 mcuxpresso.nxp.com 下载 MKL17Z128VFM4 对应的 SDK 开发包，在选择类型上要注意按照 Processor 型号来选择如下图 1，然后进入"Build MCUXpresso SDK"里面，建议选择 All toolchains 和 Select All 如下图 2；

Select Development Board
Search for your board or kit to get started.

Search by Name
MKL17Z128

Select a Board, Kit, or Processor

- Boards
- Kits
- Processors

MKL17Z128xxx4

Name your SDK
SDK_2.8.0_MKL17Z128xxx4

Hardware Details

Included Part Numbers	MKL17Z128VFT4, MKL17Z128VLH4, MKL17Z128VFM4, MKL17Z128VMP4
Board(s)	FRDM-KL43Z
Device	MKL17Z4
Core Type / Max Freq	Cortex-M0P / 48MHz
Device Memory Size	128 KB Flash 32 KB RAM

Actions

Build MCUXpresso SDK

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

Developer Environment Settings
Selections here will impact files and examples projects included in the SDK and Generated Projects

SDK Version
2.8.0 2020-07-21 REL_2.8.0_REL12_R

Toolchain / IDE
All toolchains

Host OS
Windows

Embedded real-time operating system
FreeRTOS

Note: Optional bare-metal components are always available for inclusion in the SDK build configuration, including along with the selected RTOS.

Filter by Name, Category, or Description...

Select All Unselect All

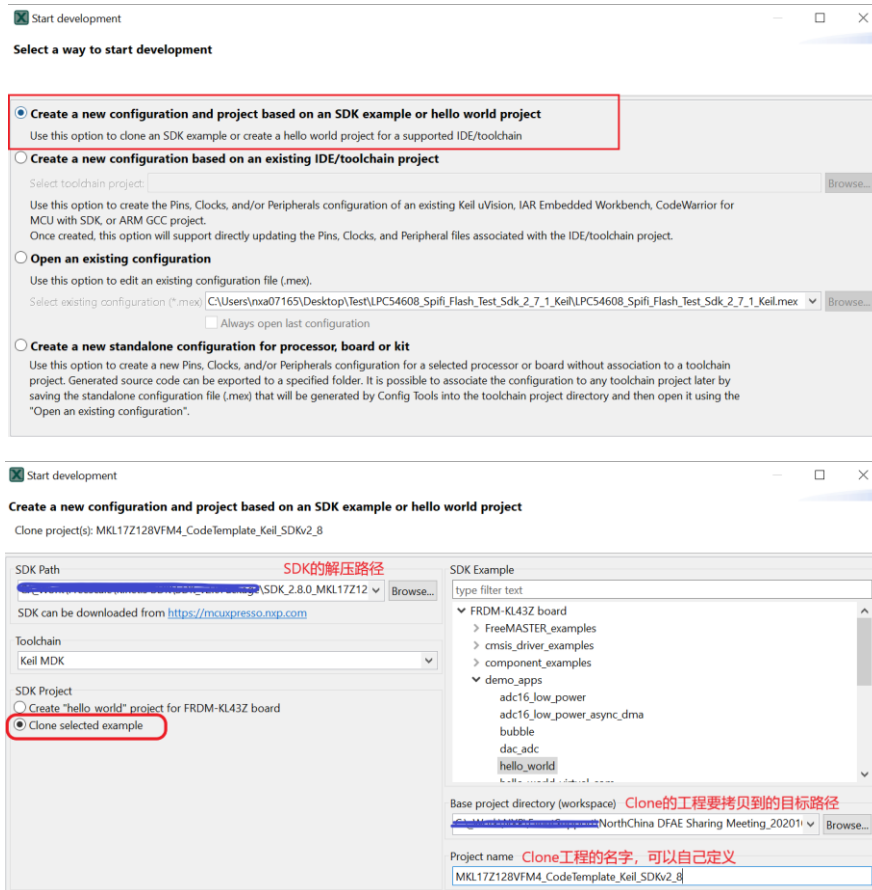
Name	Category	Description	Dependencies
------	----------	-------------	--------------

Hardware Details

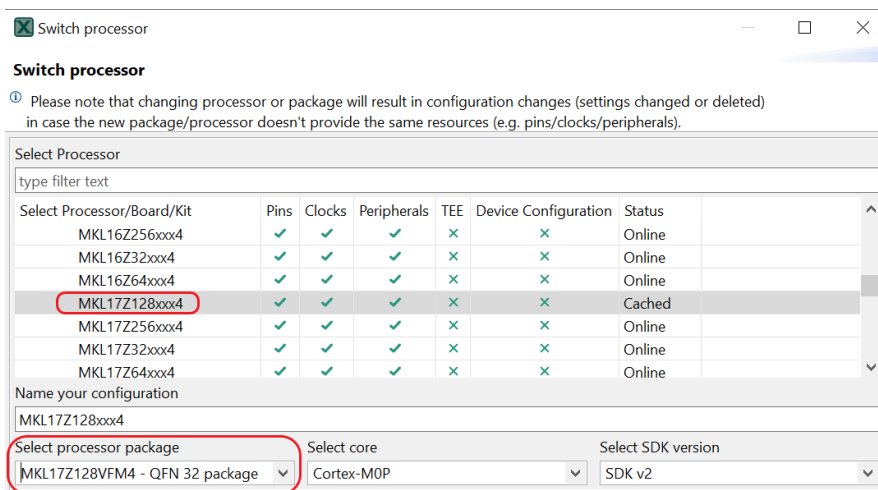
Included Part Numbers	MKL17Z128VFT4, MKL17Z128VLH4, MKL17Z128VFM4, MKL17Z128VMP4
Board(s)	FRDM-KL43Z
Device	MKL17Z4
Core Type / Max Freq	Cortex-M0P / 48MHz
Device Memory Size	128 KB Flash 32 KB RAM

SDK Details

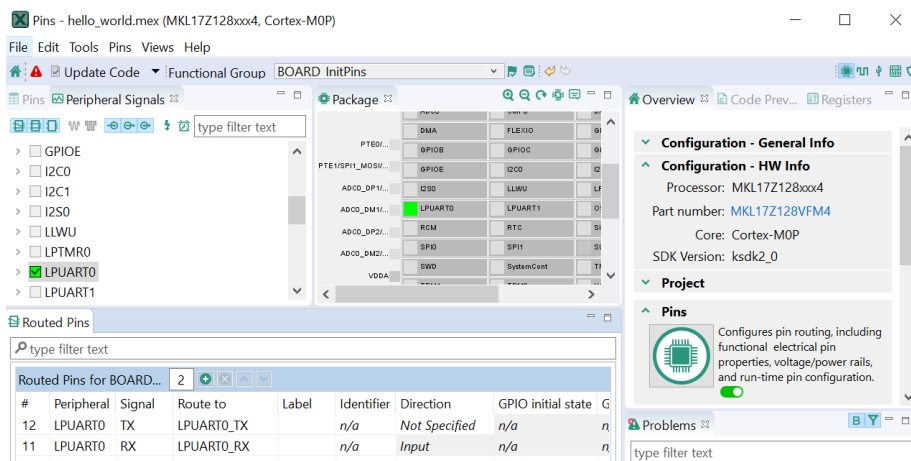
2. SDK 开发包下载完之后解压，然后打开 MCUXpresso Config Tool，选择第一个配置，即从 SDK 例程里 Clone 一个完整工程出来，点击 Ok 进入之后，选择如下图（我们可以看到 MKL17Z128 的开发板为 FRDM-K43），这里以 SDK 包里面 demo_app/hello_world 为例，将其拷贝 clone 一份出来放到用户指定的目标文件夹下，实际使用中用户可以选择任意 SDK 例程拷贝，建议选择最接近其所需功能的例程，这样移植改动东西最少；



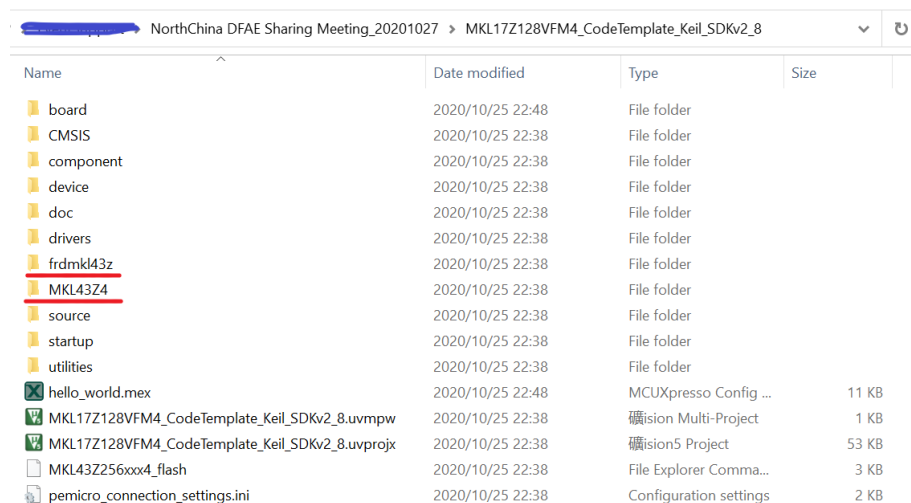
3. 上一步选择完成之后点击 finish 即可完成工程的拷贝，拷贝完之后 Config Tool 会进入 PinConfig 和 ClockConfig 界面，默认为 MKL43Z256VLH4 的配置，这里我们需要把它切换到 MKL17Z128VFM4 芯片上，以方便用户后续使用 PinConfig 和 ClockConfig 对其自己的板子做适配。点击 ConfigTool 菜单栏 File->Switch Processor 然后在 Processor 列表里找到 MKL17Z128xxx4 并在下面选择好 VFM4 即 QFN32 的封装，然后点击最下面的 Finish；



4. 此时在 Config Tool 里面我们可能会发现报错，因为之前默认的 hello world 有在 PinConfig 里面配置了 LPUART 的管脚，而该管脚可能会在新选择的芯片和封装里没有，如果报错的话，可以先把原来的 LPUART 管脚配置删掉，然后按照用户自己的板子设计选择好新的 LPUART 管脚配置如下。而对于 ClockConfig 来说，如果用户需要对时钟配置修改则可以直接在 ConfigTool 里直接更改即可，我这里保持原来 KL43 默认的配置（因为 KL43 和 KL17 同源，所以时钟配置部分兼容）：



5. 我们回到 Clone 拷贝的目标路径下，可以看到拷贝的工程很简洁清晰，脱离了 SDK 的文件结构以及路径依赖，也就是说该工程可以独立移到任意路径甚至另外的电脑上而不影响编译，然后下面红色标识的两个文件夹实际上跟工程编译没有关系，所以我们直接把这两个文件夹删掉即可；



6. 接下来开始着手 MKL17Z128 芯片相关的文件替换工作：

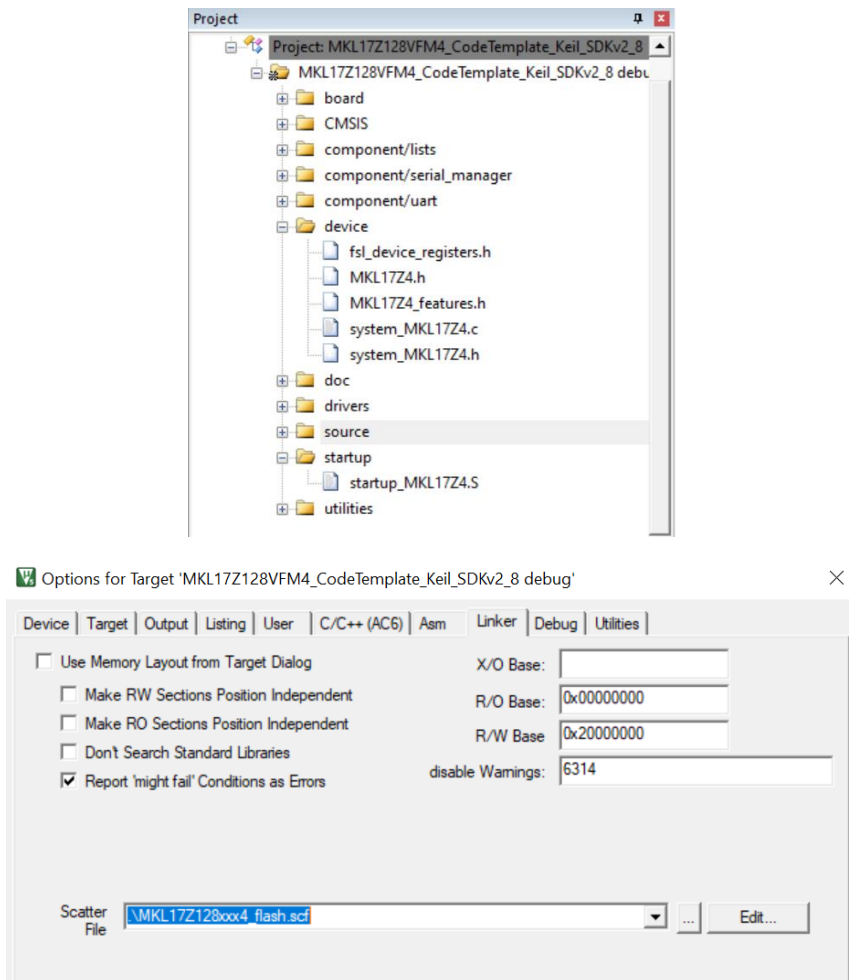
(1) 首先将目标工程路径下的 device 文件下的几个文件都删掉，然后找到 MKL17Z128 的 SDK 开发包解压路径\SDK_2.8.0_MKL17Z128xxx4\devices\MKL17Z4，将里面的 fsl_device_registers.h, MKL17Z4.h, MKL17Z4_features.h, system_MKL17Z4.c 和 system_MKL17Z4.h 这 5 个文件拷贝到目标工程路径下 device 文件夹内；

(2) 将\SDK_2.8.0_MKL17Z128xxx4\devices\MKL17Z4\drivers 路径下的 KL17Z128 底层驱动文件拷贝并覆盖目标工程路径下 drivers 文件夹内的文件；

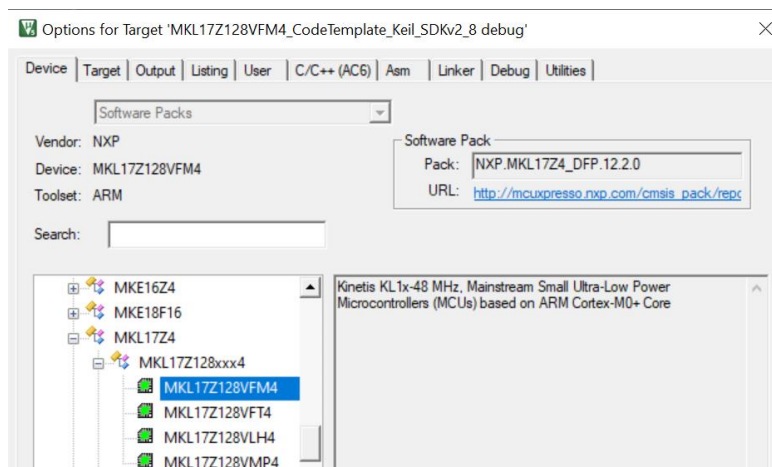
(3) 删掉目标工程目录下 startup 文件夹的原.s 启动汇编文件，将\SDK_2.8.0_MKL17Z128xxx4\devices\MKL17Z4\arm 下的 startup_MKL17Z4.S 文件拷贝到 startup 文件夹内；

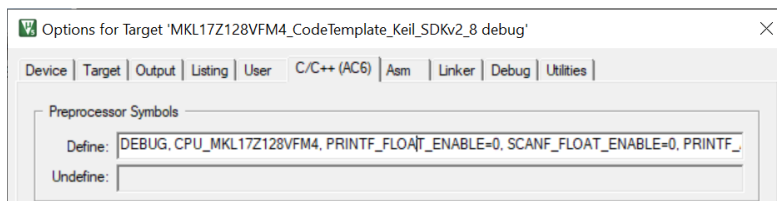
(4) 删掉目标工程目录下 MKL43Z256xxx4_flash 分散加载文件，将\SDK_2.8.0_MKL17Z128xxx4\devices\MKL17Z4\arm 下的 MKL17Z128xxx4_flash 文件拷贝到目标工程根目录下；

7. 文件替换完毕，双击打开目标工程目录下的 Keil 工程，将上一步替换的文件在 Keil 工程里逐一替换生效，如下图



8. 将 Device 型号修改为 MKL17Z128VFM4，并在 C/C++ 的 Define 宏列表里将原 CPU_MKL43Z256VLH4 修改为 CPU_MKL17Z128VFM4 以使 KL17 对应的寄存器头文件生效（打开 fsl_device_registers.h 就明白了），然后把该 Define 宏列表里的 FRDM_KL43Z 和 FREEDOM 这两个宏删掉（整个工程里实际没有用这两个宏，直接删掉即可）。另外注意，如果 Keil 工程里有 Debug 和 Release 两个工程配置，如果当前改的是 Debug，则需要再切换到 Release 配置然后参照第 7 步和第 8 步同步修改一下；





9. Keil 工程下的 board.h 实际上定义了很多跟 FRDM-KL43 板级相关的宏，当然因为它仅仅是宏，所以用户如果不调用的话对工程编译没有影响，如果从代码规整角度来考虑则建议把不需要的宏去掉即可（UART 和 I2C 相关的宏由于在 board.c 里有函数调用，所以建议保留）；

至此整个移植过程就结束了，虽然步骤很多但是实际上熟悉了之后修改起来还是很简单。另外，虽然本文档仅以 KL17Z128 移植为例，但是用户完全可以参考该文档步骤对 NXP 所有其他芯片做移植适配。