

# Table of Contents

1.MCU 集成开发环境	1.1
2.基于 NXP 的 Keil 开发环境搭建	1.2
2.1.下载并安装 Keil	1.2.1
2.2.下载 Keil SDK	1.2.2
2.3.下载 Package	1.2.3
2.4.Keil 工程搭建	1.2.4
2.4.1.添加目录结构树	1.2.4.1
2.4.2.修改 Options	1.2.4.2
2.4.3.使用 MCUXpresso Config Tools 生成代码	1.2.4.3
2.4.4.HelloWorld 示例	1.2.4.4
3.基于 NXP 的 IAR 开发环境搭建	1.3
3.1.下载并安装 IAR	1.3.1
3.2.下载 IAR SDK	1.3.2
3.3.IAR 工程搭建	1.3.3
3.3.1.添加目录结构树	1.3.3.1
3.3.2.修改 Options	1.3.3.2
3.3.3.使用 MCUXpresso Config Tools 生成代码	1.3.3.3
3.3.4.HelloWorld 示例	1.3.3.4
4.基于 NXP 的 MCUXpresso 开发环境搭建	1.4
4.1.下载并安装 MCUXpresso IDE	1.4.1
4.2.下载并安装 MCUXpresso Config Tools	1.4.2
4.3.下载 MCUXpresso SDK	1.4.3
4.4.使用 MCUXpresso Config Tools 生成代码	1.4.4
4.5.使用 MCUXpresso IDE 调试代码	1.4.5
4.5.1.开启视图窗口	1.4.5.1
4.5.2.加载 Xpresso SDK	1.4.5.2
4.5.3.HelloWorld 示例	1.4.5.3



产业首选·通路标竿

## 1.MCU 集成开发环境

MCU 集成开发环境 IDE，在这里主要介绍三类

IDE	公司	支持操作系统
Keil uVision5 for MDK-Arm	Keil Software	Windows
IAR Embedded Workbench	IAR	Windows
MCUXpresso IDE	NXP	Windows、Linux、MAC



产业首选·通路标竿

## 2. 基于 NXP 的 Keil 开发环境搭建

### 概述

Keil C51 是美国 Keil Software 公司出品的 51 系列兼容单片机 C 语言软件开发系统，与汇编相比，C 语言在功能上、结构性、可读性、可维护性上有明显的优势，因而易学易用。Keil 提供了包括 C 编译器、宏汇编、链接器、库管理和一个功能强大的仿真调试器等在内的完整开发方案，通过一个集成开发环境（μVision）将这些部分组合在一起。运行 Keil 软件需要 WIN98、NT、WIN2000、WINXP 等操作系统。如果你使用 C 语言编程，那么 Keil 几乎就是你的不二之选，即使不使用 C 语言而仅用汇编语言编程，其方便易用的集成环境、强大的软件仿真调试工具也会令你事半功倍。

### 特性

1. 引入了灵活的窗口管理系统，能够拖放到视图内的任何地方，包括支持多显示器窗口。
2. 增加了更多大众化的功能。

多显示器和灵活的窗口管理系统  
系统浏览器窗口的显示设备外设寄存器信息  
调试还原视图创建并保存多个调试窗口布局  
多项目工作区简化与众多的项目



产业首选·通路标竿

## 2.1.下载并安装 Keil

没提供下载地址

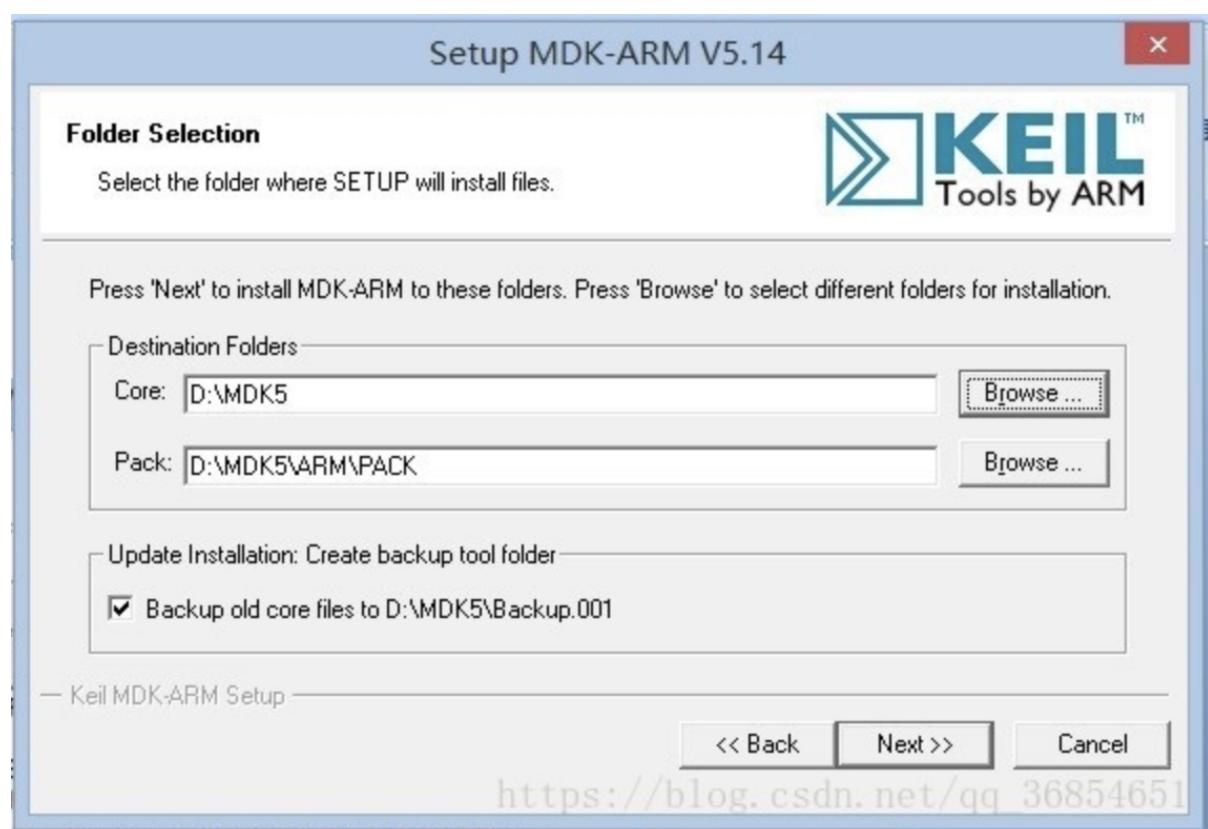
Step 1.双击安装包，点击 Next



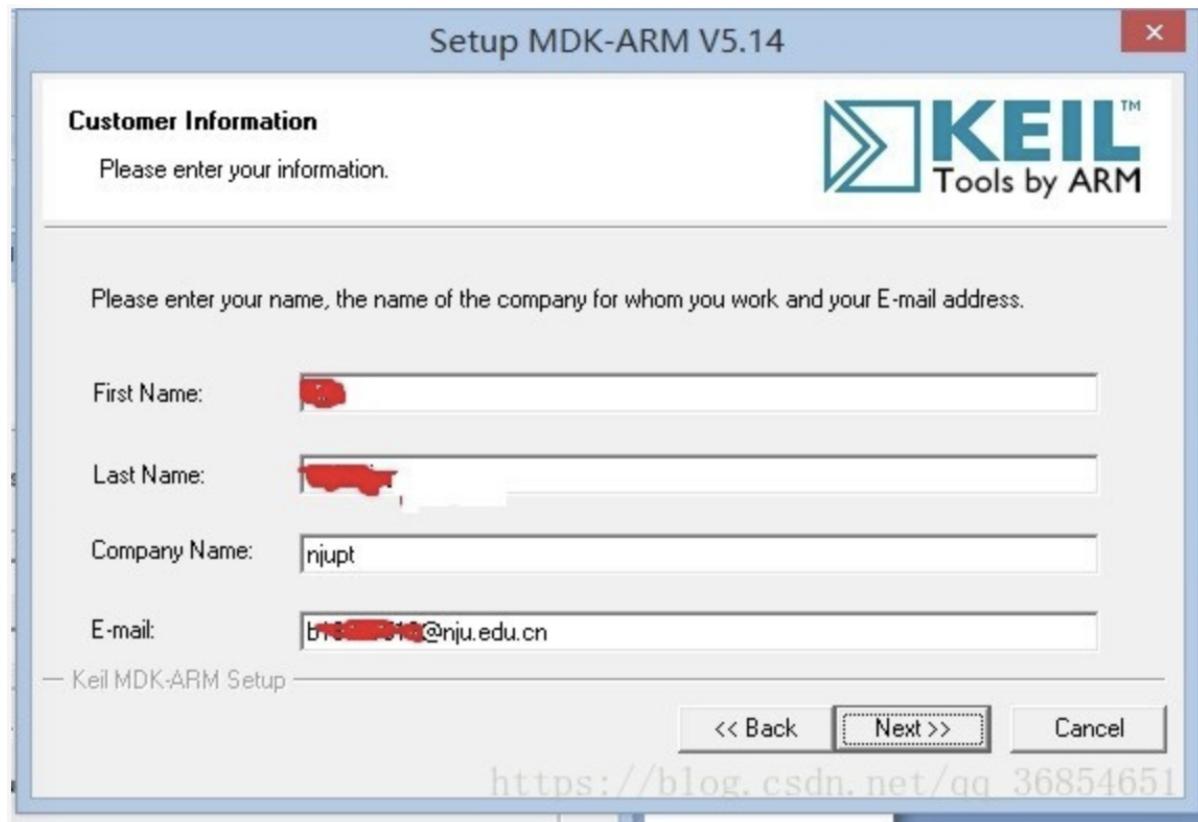
Step 2.选择 I agree, 点击 Next



Step 3. 选择安装目录，点击 Next



Step 4. 填写个人信息，点击 Next



Step 5. 等待安装完成







产业首选·通路标竿

## 2.2. 下载 Keil SDK

[SDK 下载地址](#)

需注册 NXP 官网账号

Step 1. 进入下载地址, 搜索你要下载芯片的 SDK, 这里以 MKV30F64xxx10 为例, 选择 MKV30F64xxx10, 右方会显示该芯片的信息, 然后点击 Build MCUXpresso SDK

**Search by Name**

X

**Select a Device, Board, or Kit**

- ▼ Boards
- ▼ Kits
- ▼ Processors

MKV30F64xxx10

**Name your SDK**

Don't use: <,>,;,"/,|,?,\*,\ in the name of your SDK

**Hardware Details**

Included Part Numbers	MKV30F64VLF10, MKV30F64VFM10, MKV30F64VLH10
Board(s)	TWR-KV31F120M, FRDM-KV31F
Device	MKV30F12810
Core Type / Max Freq	Cortex-M4F / 100MHz
Device Memory Size	64 KB Flash 16 KB RAM

**Actions**

- Build MCUXpresso SDK
- Explore selection with Clocks tool
- Explore selection with Pins tool

Step 2. 选择 Windows 主机系统, 下载的工具链为 Keil MDK, 组件默认即可, 若有特别的组件需求可自己添加进去, 点击 Download SDK

## Developer Environment Settings

Selections here will impact files and examples projects included in the SDK and Generate

Host OS

Windows ▾

Toolchain / IDE

Keil MDK ▾

## Select Optional Middleware

Add middleware, operating systems, and software libraries to your SDK.

+ Add software component

This MCUXpresso SDK configuration is available for direct download

Download SDK

Archive Name

SDK\_2.5.0\_MKV30F64xxx10

Don't use: <, >, :, ", /, |, ?, \*, \ in the name  
of your SDK

Step 3. 点击 I Agree

## SDK Downloads

X

### Software Terms and Conditions

Please read the following agreement and click "I AGREE" at the bottom before downloading your software.

[EULA](#)[Software Content Register](#)

LA\_OPT\_NXP\_Software\_License v4 November 2018

**IMPORTANT.** Read the following NXP Software License Agreement ("Agreement") completely. By selecting the "I Accept" button at the end of this page, or by downloading, installing, or using the Licensed Software, you indicate that you accept the terms of the Agreement and you acknowledge that you have the authority, for yourself or on behalf of your company, to bind your company to these terms. You may then download or install the file. In the event of a conflict between the terms of this Agreement and any license terms and conditions for NXP's proprietary software embedded anywhere in the Licensed Software file, the terms of this Agreement shall control. If a separate license agreement for the Licensed Software has been signed by you and NXP, then that agreement shall govern your use of the Licensed Software and shall supersede this Agreement.

#### NXP SOFTWARE LICENSE AGREEMENT

This is a legal agreement between your employer, of which you are an authorized representative, or, if you have no employer, you as an individual ("you" or "Licensee"), and NXP B.V. ("NXP"). It concerns your rights to use the software provided to you in binary or source code form and any accompanying written materials (the "Licensed Software"). The Licensed Software may include any updates or error corrections or documentation relating to the Licensed Software provided to you by NXP under this Agreement. In consideration for NXP allowing you to access the Licensed Software, you are agreeing to be bound by the terms of this Agreement. If you do not agree to all of the terms of this Agreement, do not download or install the Licensed Software. If you change your mind later, stop using the Licensed Software and delete all copies of the Licensed Software in your possession or control. Any

[I Agree](#)[← Back](#)[Close](#)



产业首选·通路标竿

## 2.3. 下载 Package

[Package 下载地址](#)

Step 1. 搜索你要下载的 arm 芯片，在这里以 NXP MKV30F64xxx10 为例

The screenshot shows the Arm KEIL website's navigation bar with links for Home, Products, Download, Events, and Support. Below the navigation bar is a search bar containing the text "MDK5 - NXP MKV30F64xxx10". To the right of the search bar is a green "Go" button.

Step 2. 选择 MDK5-NXP MKV30F64xxx10

The screenshot shows the search results page with a search bar containing "MDK5 - NXP MKV30F64xxx10". Below the search bar, it says "Results 1-2 of 2". The first result is a link labeled "MDK5 - NXP MKV30F64xxx10".

[MDK5 - NXP MKV30F64xxx10](#)

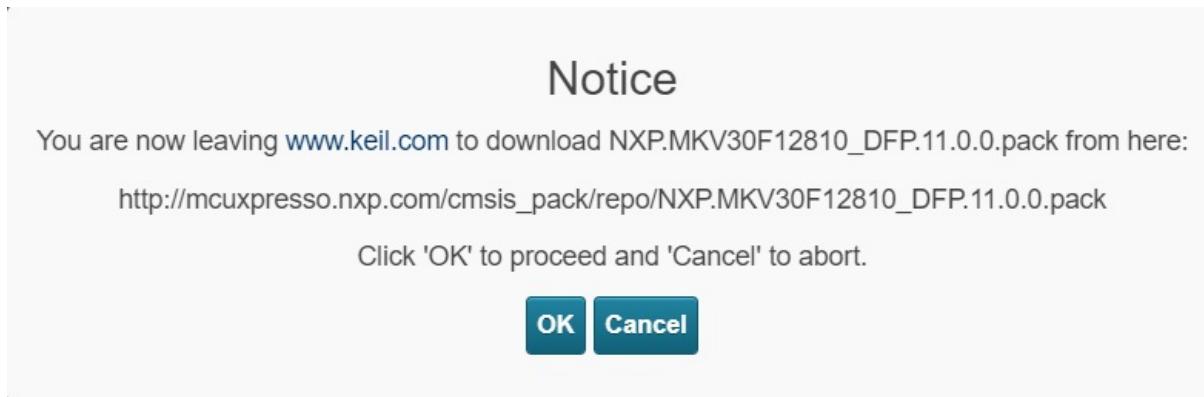
Distributors Home / MDK5 Device List / MKV30F64xxx10 NXP MKV30  
our cookies.

<http://www.keil.com/dd2/nxp/mkv30f64xxx10/>

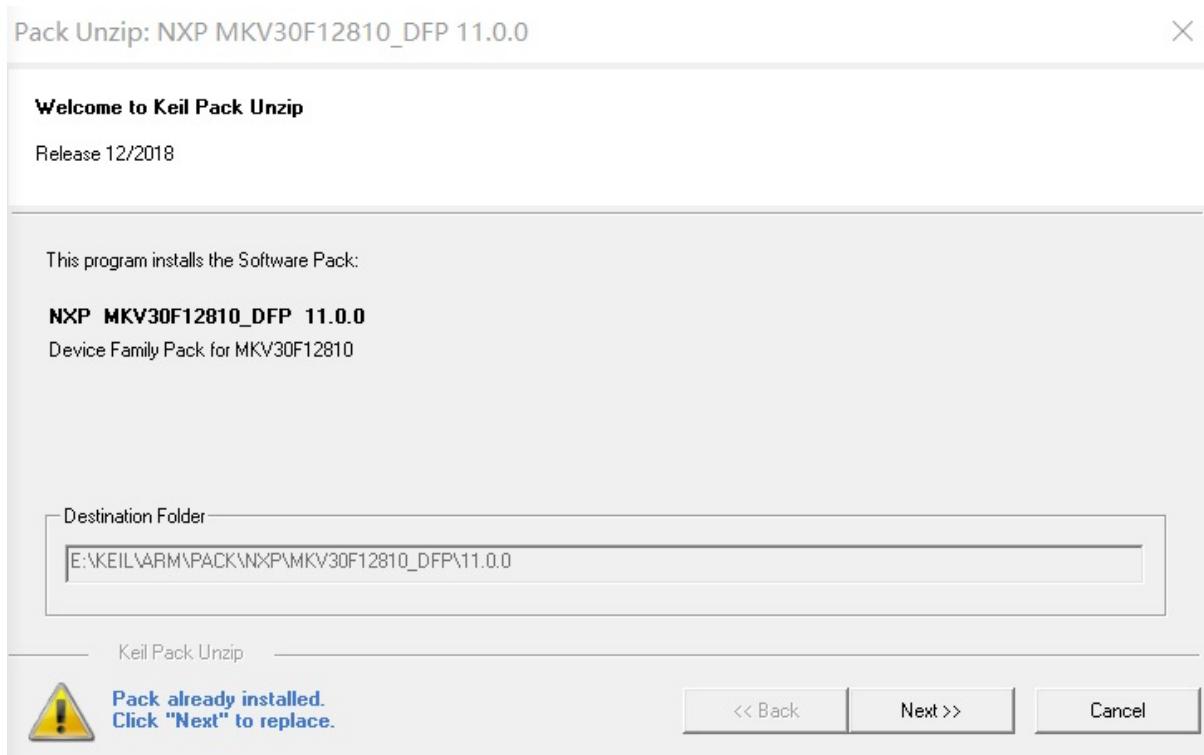
Step 3. 页面右下角下载 Pack，点击 Download

The screenshot shows the "Device Family Pack" page for the MKV30F12810 device. The page features a large title "Device Family Pack" and a green button labeled "DFP". Below the title, it says "Support for this device is contained in: Device Family Pack for MKV30F12810". At the bottom right is a large blue "Download" button with a white downward arrow icon.

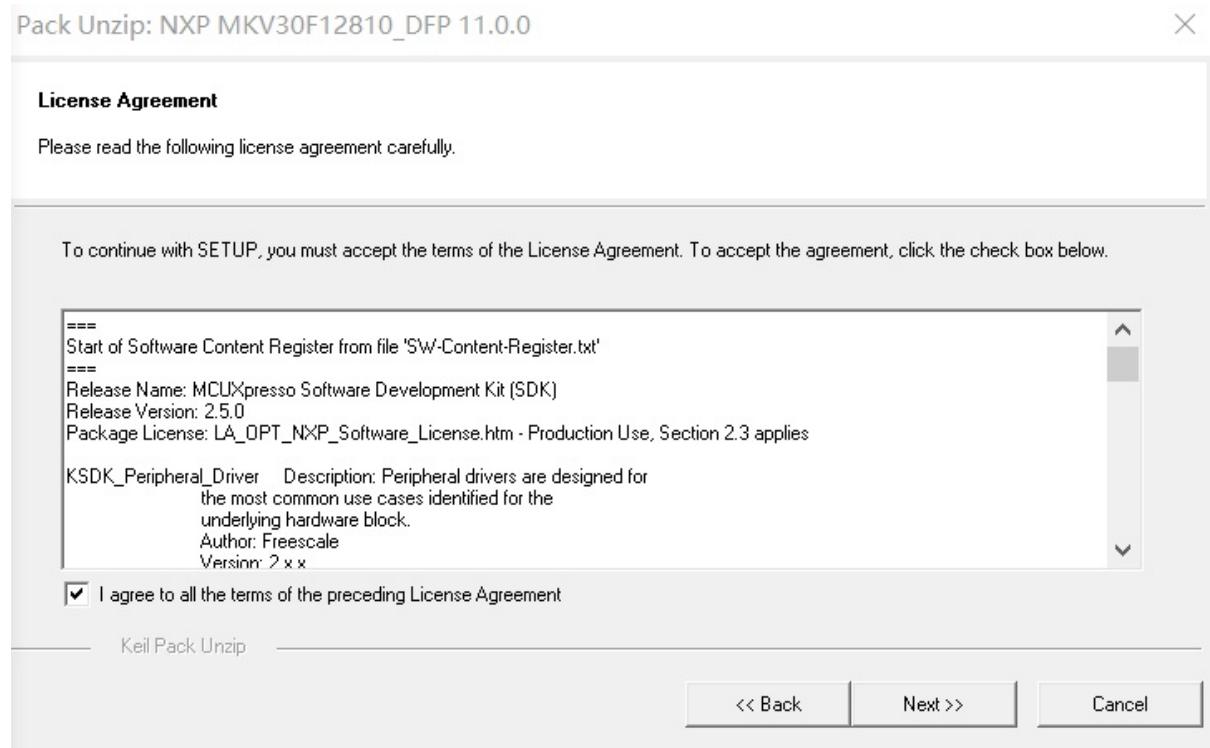
Step 4. 点击 OK



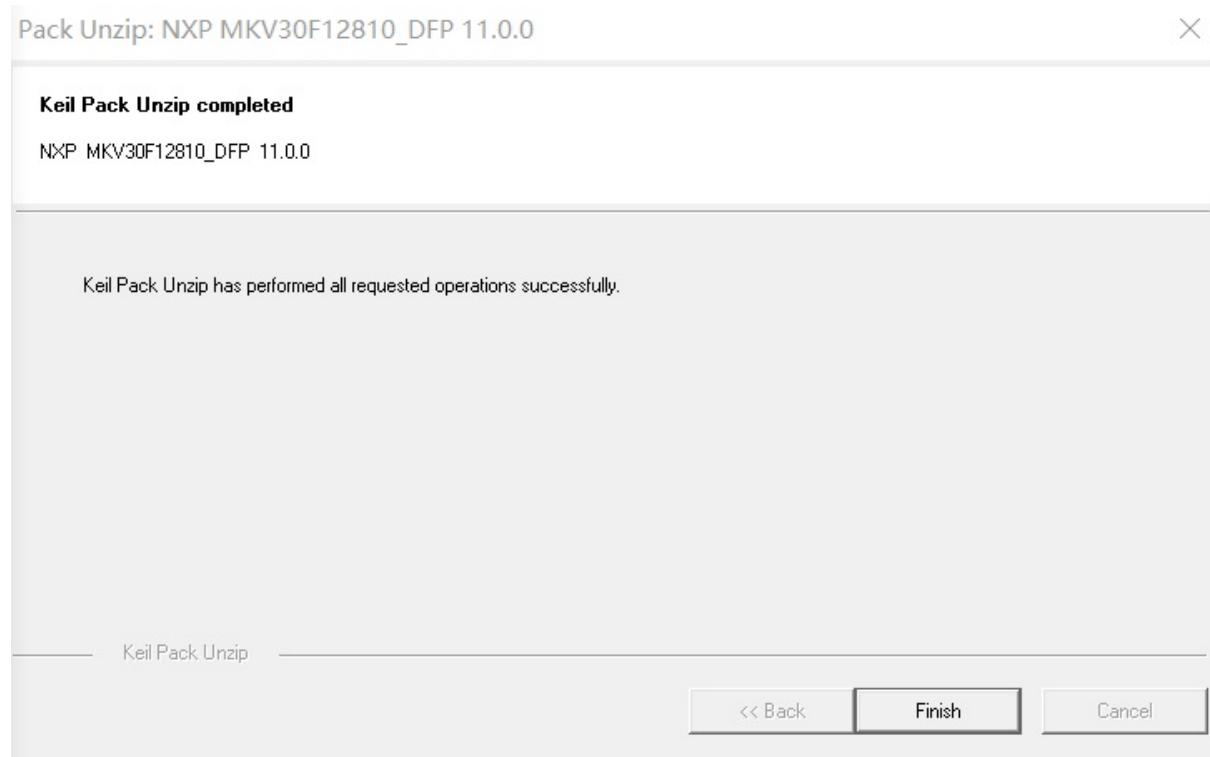
Step 5. 打开下载好的 Pack，它会自动搜寻你的 Keil 路径，点击 Next



Step 6. 选择 I agree, 点击 Next



Step 7.点击 Finish

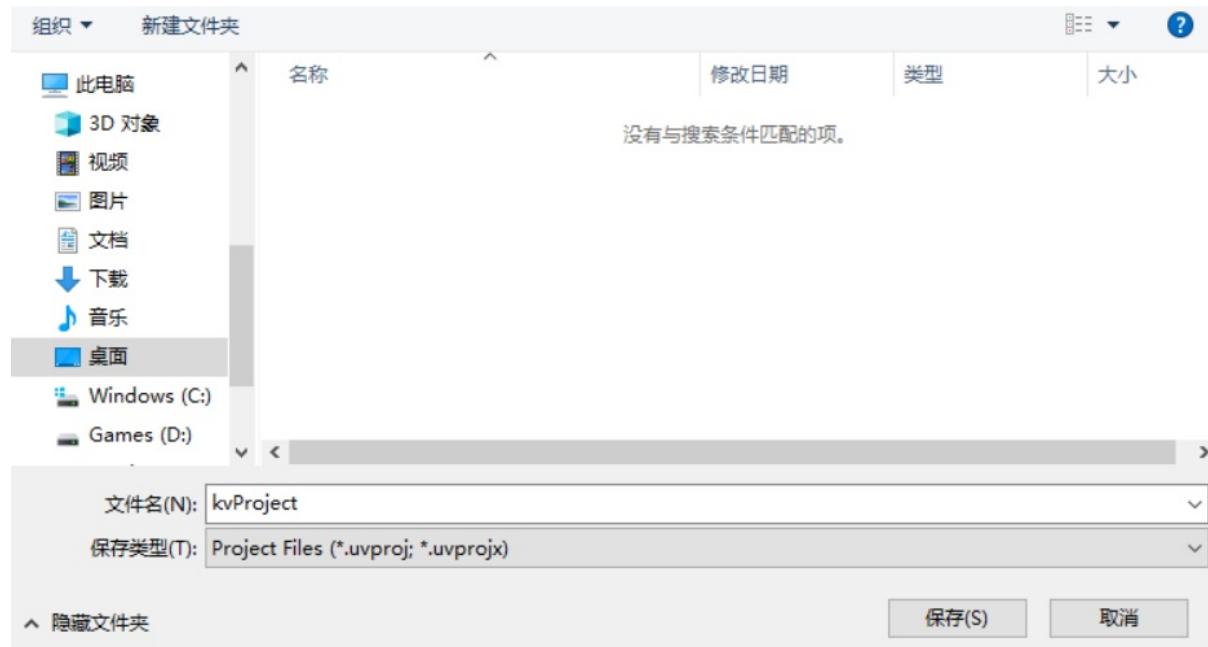




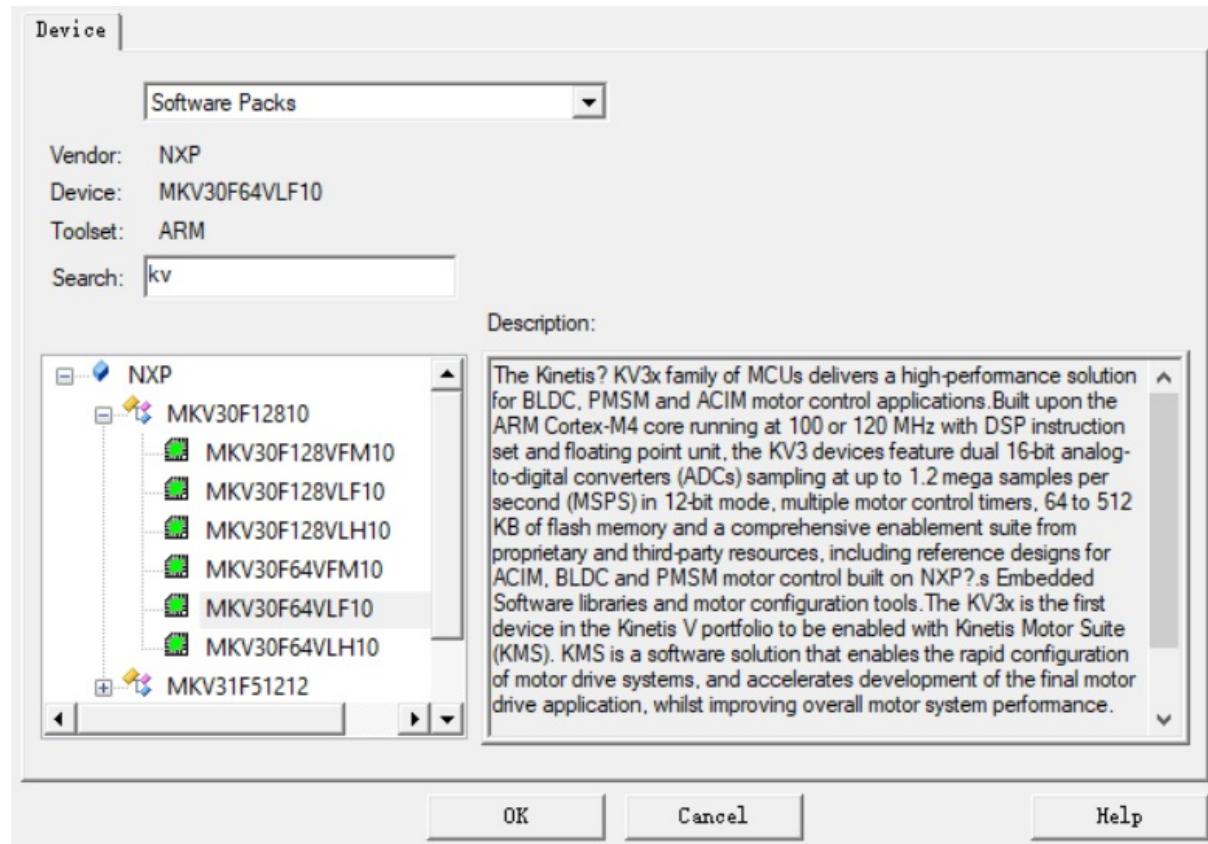
## 2.4.Keil 工程搭建

Step 1. 在指定位置新建工程文件夹，指定名字为 kv30Project

Step 2. 双击下载好的 Keil，点击工具栏 Project -> New uVision Project...，保存你的工程文件到上一步新建的工程文件夹中，取名为 kv30Project



Step 3. 在这里使用的是 MKV30F64VLF10



#### Step 4.点击 OK

Software Component	Sel.	Variant	Version	Description
Board Support				Generic Interfaces for Evaluation and Development Boards <a href="#">Cortex Microcontroller Software Interface Components</a>
CMSIS				NXP MCUXpresso SDK Peripheral CMSIS Drivers
CMSIS Driver				<a href="#">Compiler Extensions for ARM Compiler 5 and ARM Compiler 6</a>
Compiler		ARM Compiler	1.2.0	<a href="#">Startup, System Setup</a>
Device				<a href="#">File Access on various storage devices</a>
File System		MDK-Pro	6.9.8	<a href="#">User Interface on graphical LCD displays</a>
Graphics		MDK-Pro	5.36.6	<a href="#">IPv4/IPv6 Networking using Ethernet or Serial protocols</a>
Network		MDK-Pro	7.5.0	<a href="#">USB Communication with various device classes</a>
USB		MDK-Pro	6.11.0	

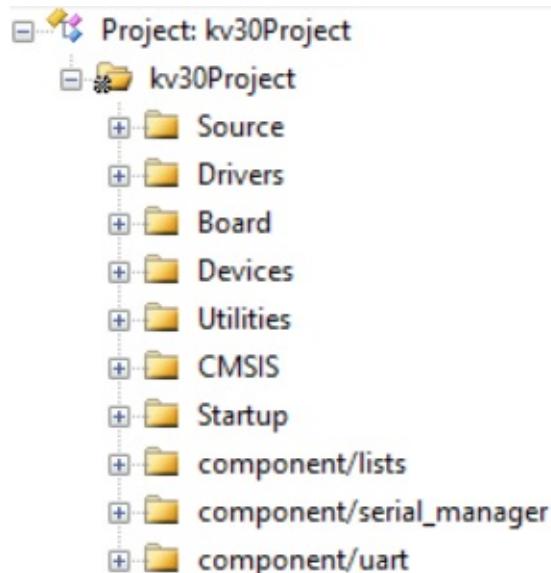


## 2.4.1.添加目录结构树

Step 1.左边的工程结构树如下，修改结构树，鼠标点击结构树文件夹名称，再单击一次文件夹名称则可以重命名，在结构树文件夹右键 Add Group 则可以添加文件夹



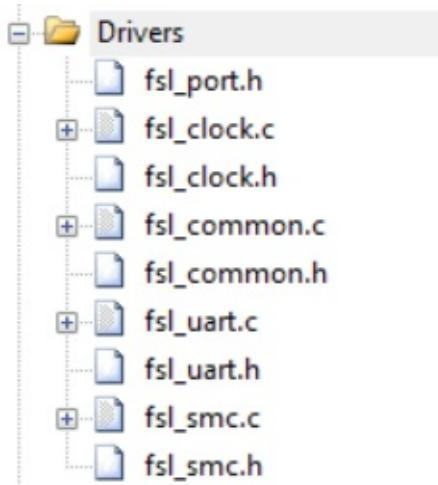
Step 2.工程结构树修改完成，如图



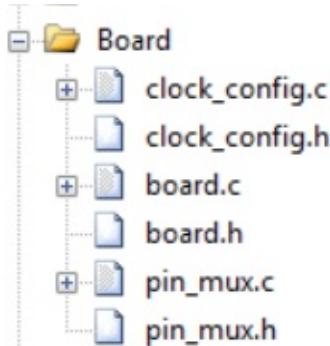
Step 3.将下载的 SDK 解压，并将 SDK 根目录下的 components、CMSIS、devices/MKV30F12810 复制到新建的工程文件夹下，如图

名称	修改日期	类型
MKV30F12810	2019/4/3 18:21	文件夹
CMSIS	2019/4/3 18:31	文件夹
components	2019/4/3 18:34	文件夹

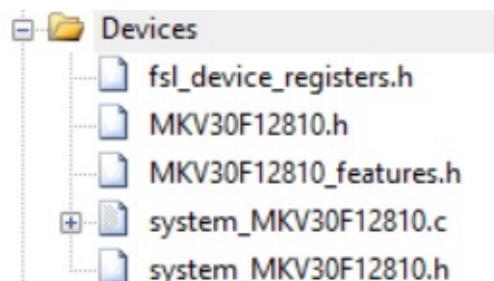
Step 4.双击工程结构树的 Drivers 文件夹，添加以下文件，文件位于 ./MKV30F12810/drivers



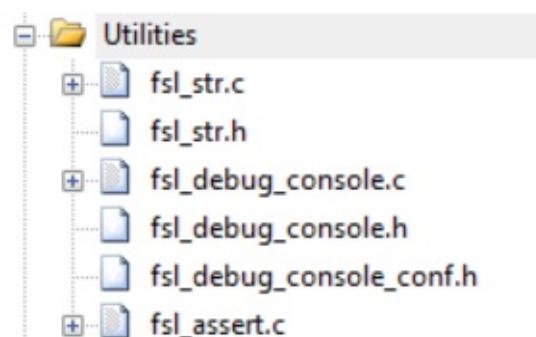
Step 5. 双击工程结构树的 Board 文件夹，添加以下文件，文件位于 ./MKV30F12810/project\_template



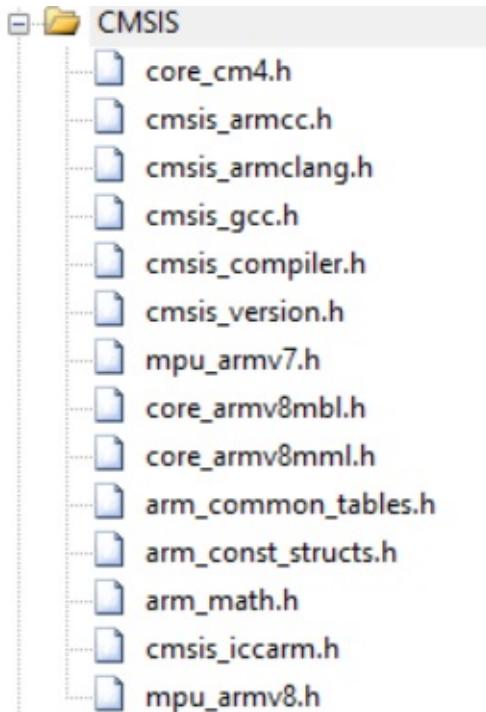
Step 6. 双击工程结构树的 Devices 文件夹，添加以下文件，文件位于 ./MKV30F12810



Step 7. 双击工程结构树的 Utilities 文件夹，添加以下文件，文件位于  
./MKV30F12810/utilities、./MKV30F12810/utilities/debug\_console、./MKV30F12810/utilities/str



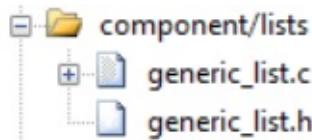
Step 8. 双击工程结构树的 CMSIS 文件夹，添加以下文件，文件位于 ./CMSIS/Include



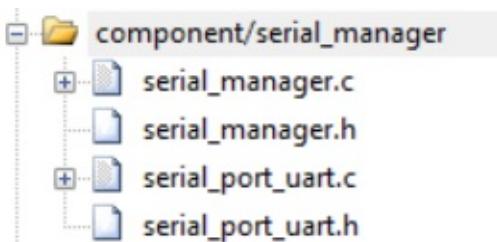
Step 9. 双击工程结构树的 Startup 文件夹，添加以下文件，文件位于 ./MKV30F12810/arm



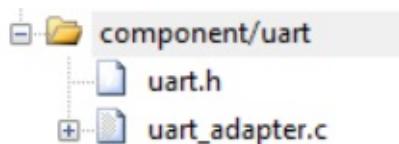
Step 10. 双击工程结构树的 component/lists 文件夹，添加以下文件，文件位于 ./components/lists



Step 11. 双击工程结构树的 component/serial\_manager 文件夹，添加以下文件，文件位于 ./components/serial\_manager



Step 12. 双击工程结构树的 component/uart 文件夹，添加以下文件，文件位于 ./components/uart





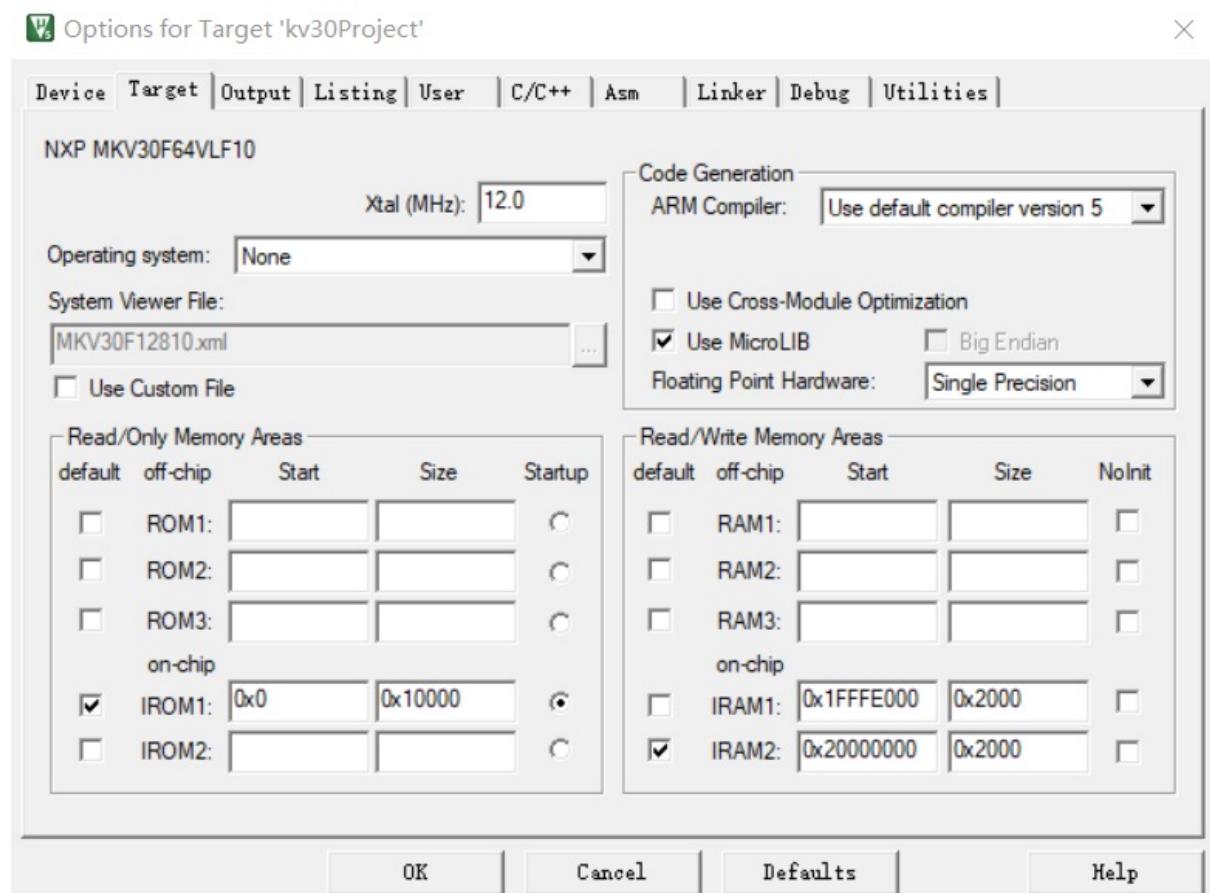


## 2.4.2.修改 Options

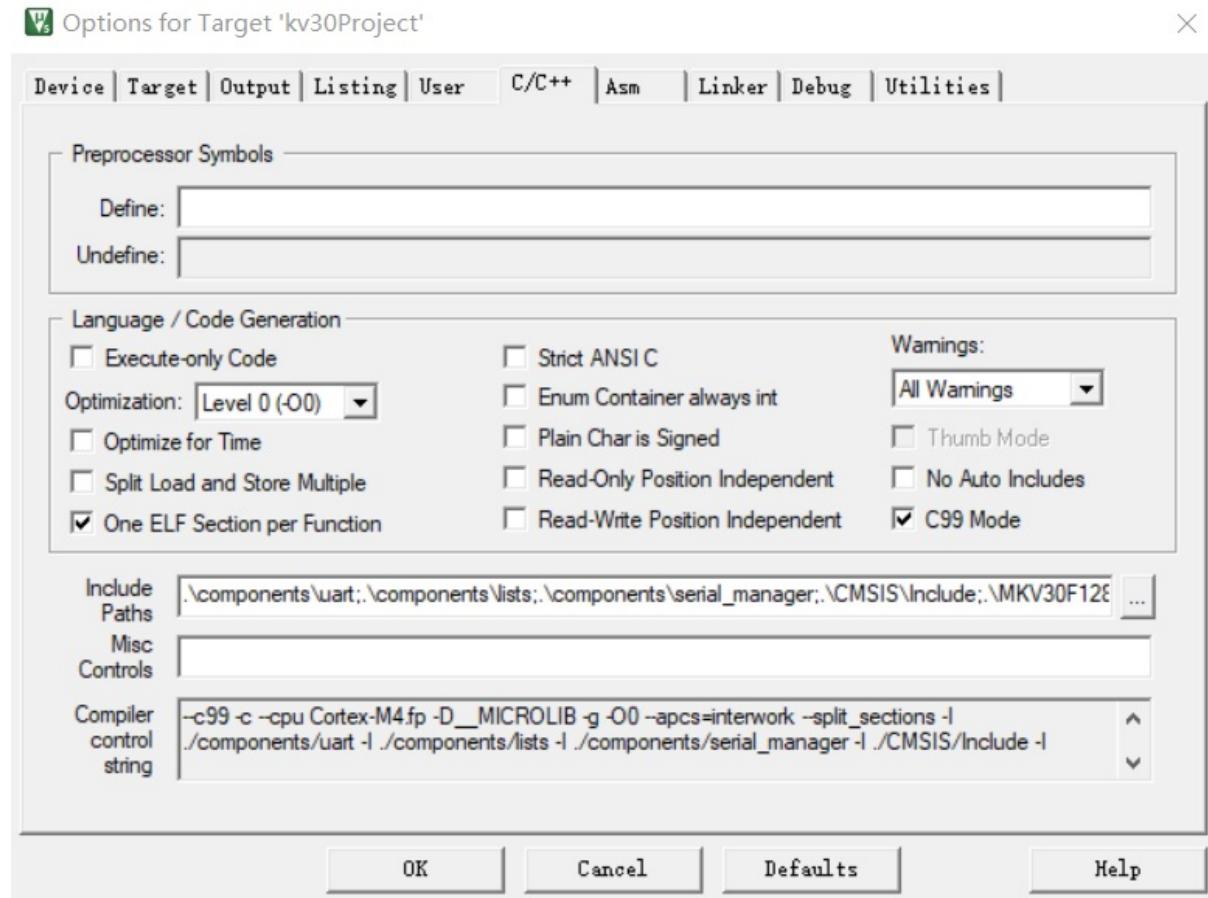
Step 1.点击上方工具栏的 Option 按钮



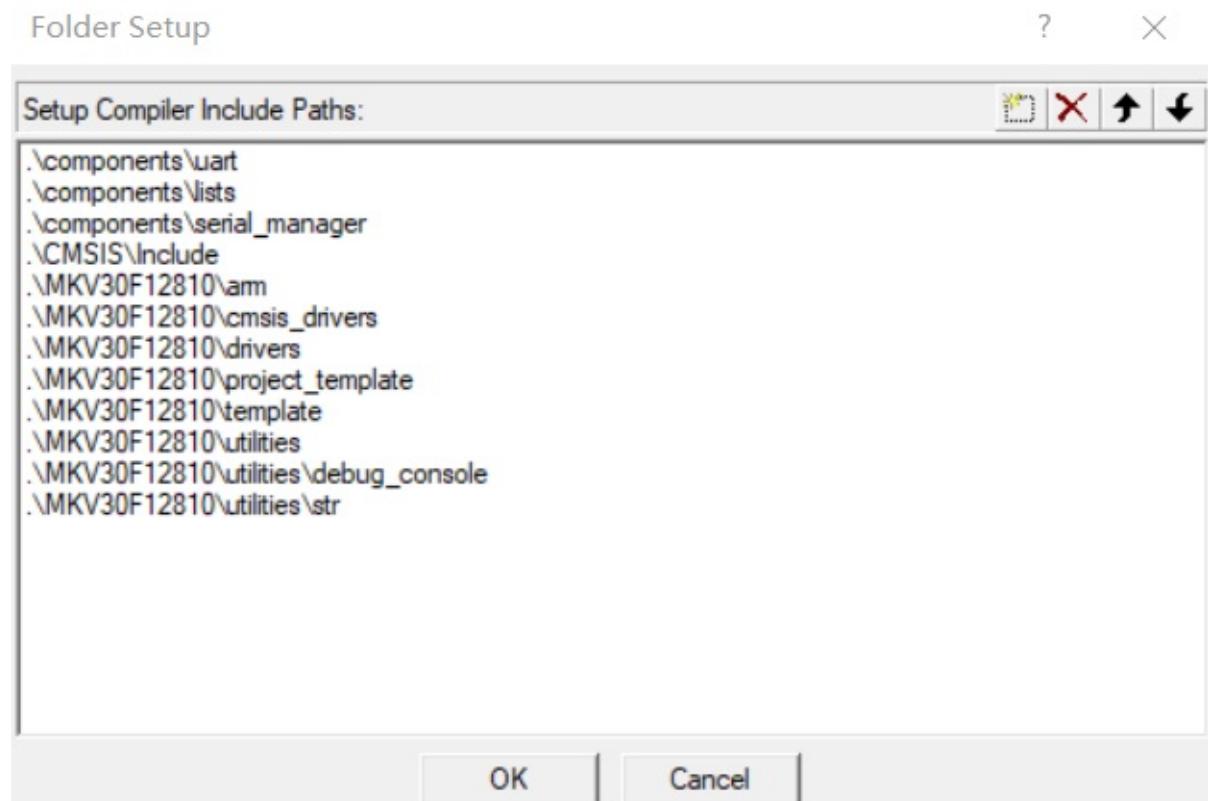
Step 2.选择 Target 子窗口，勾选上 Use MicroLIB



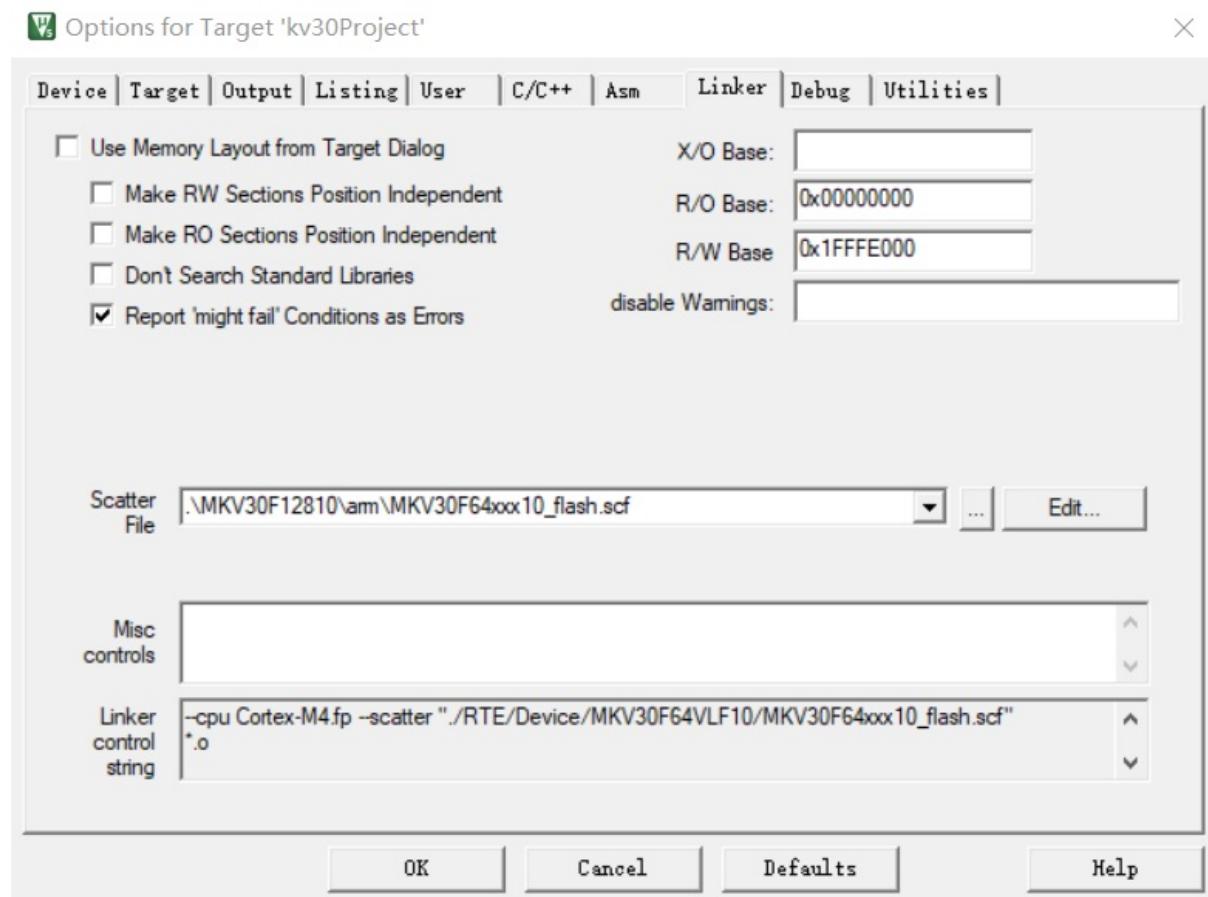
Step 3.选择 C/C++ 子窗口，Misc Controls 添加以下参数



Step 4.选择 C/C++ 子窗口，Include Paths 添加以下路径



Step 5.选择 Linker 子窗口，取消勾选 Use Memory Layout from Target Dialog，修改 Scatter File，添加 .scf 文件，文件路径位于 ./MKV30F12810/arm/MKV30F64xxx10\_flash.scf





## 2.4.3. 使用 MCUXpresso Config Tools 生成代码

### 生成配置外部晶振的时钟文件

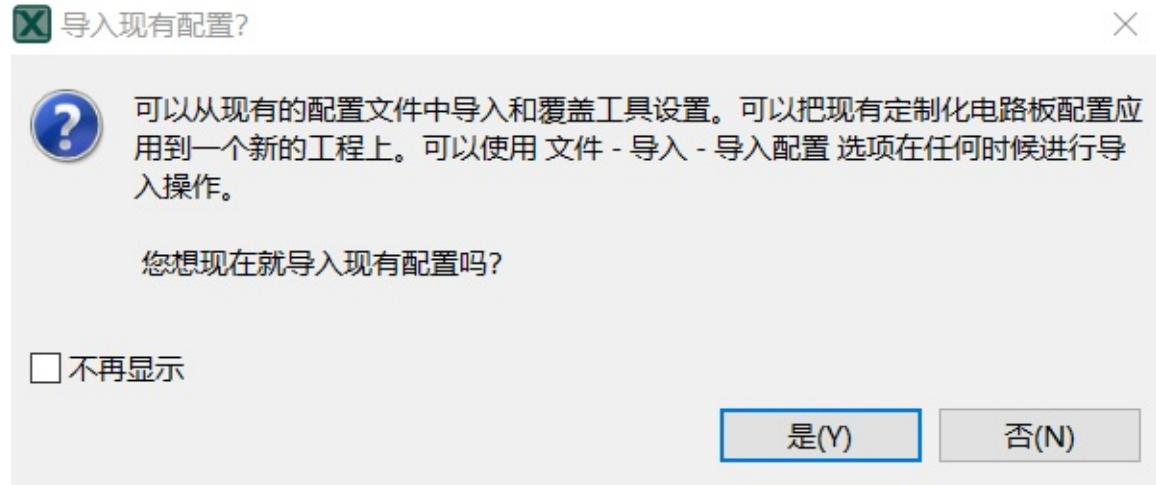
Step 1. 打开 MCUXpresso Config Tools，选择为处理器，电路板，和套件创建新配置，点击下一步



Step 2. 查询你配置的芯片，在这里为 MKV30F64xxx，封装为 MKV30F64VLF10 - LQFP 48 封装



Step 3. 点击否



Step 4. 点击工具栏：工具->时钟，该芯片默认配置为内部时钟，点击外部 OSC 的 Inactive 框

时钟源	
名称	可 值
内部	
FAST_IRCLK	4 MHz
SLOW_IRCLK	32.768 kHz
IRC48M	Inactive
LPO	1 kHz
外部	
<input checked="" type="checkbox"/> OSC (System Oscillator)	<input type="checkbox"/> Inactive

Step 5. 输入芯片的外部晶振参数，在这里使用的是无源的 8MHz 晶振，点击 OK



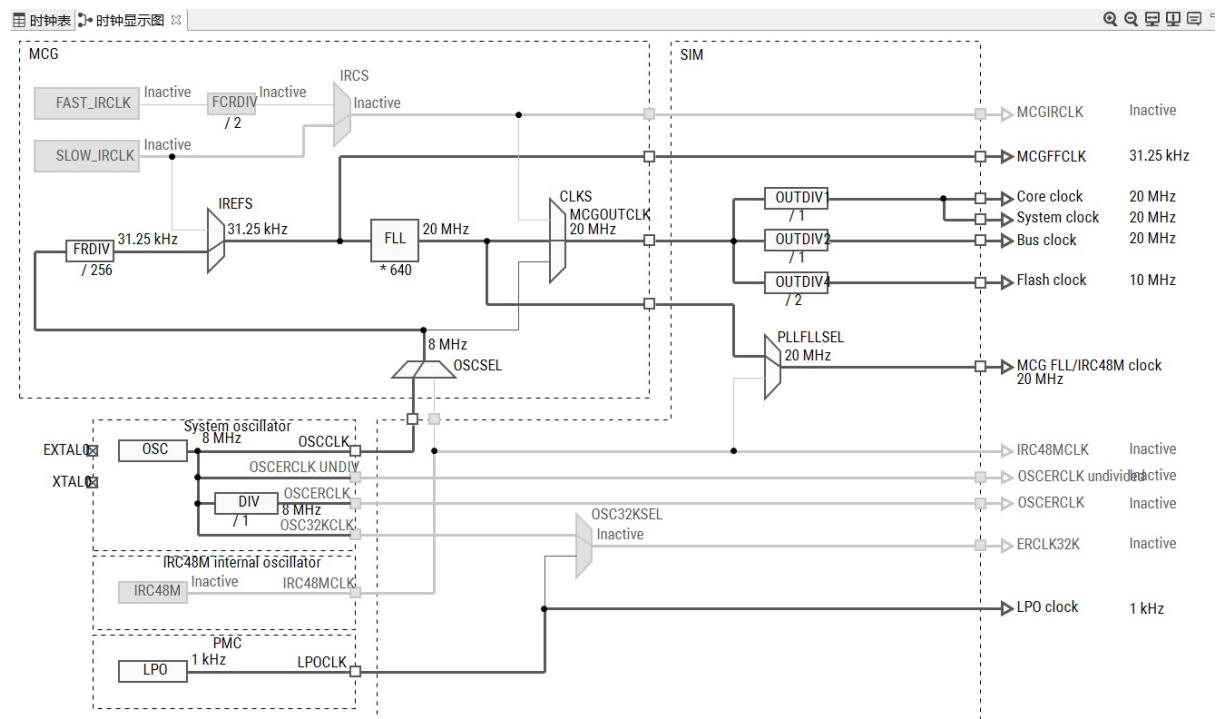
Step 6. 将外部 OSC 下拉框展开，OSC mode 选择 Using oscillator with external crystal(low power)，Frequency Range 选择 High frequency range 3-8 MHz，System Osc.Capacity Load 选择 0pF (芯片内部晶振的负载电容)

外部	
<input checked="" type="checkbox"/> OSC (System Oscillator)	<input checked="" type="checkbox"/> 8 MHz
OSC mode	Using oscillator...tal (low power)
Frequency Range	High frequency range 3-8 MHz
System O...ty Load	0 pF

Step 7. 工具栏的 MCG Mode 从 FEI 模式切换成 FEE 模式

## MCG Mode FEE (FLL Engaged External)

Step 8. 观察时钟显示图，时钟已配置成外部晶振输入



Step 9. 右方工具栏点击源代码，可查看自动生成的时钟代码文件，右方有个导出按钮，将时钟的 .c 和 .h 文件导出到指定目录

三 详情 源代码 寄存器 模块时钟

i 生成的代码预览

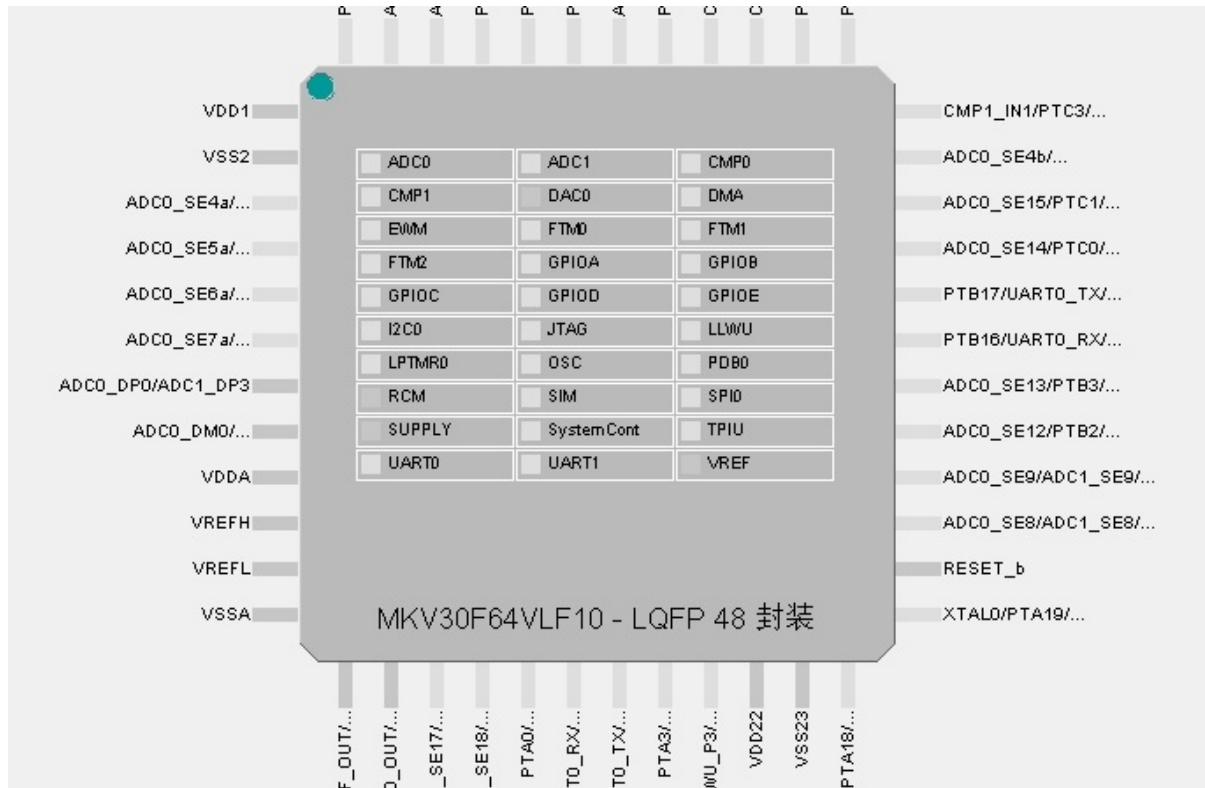
clock\_config.c clock\_config.h

```
/****************************************************************************
 * This file was generated by the MCUXpresso Config Tools
 * will be overwritten if the respective MCUXpresso Config
 ****************************************************************************
 */
/* How to setup clock using clock driver functions:
 *
 * 1. CLOCK_SetSimSafeDivs, to make sure core clock, bus
 *    and flash clock are in allowed range during clock n
 *
 * 2. Call CLOCK_Osc0Init to setup OSC clock, if it is us
 *
 * 3. Set MCG configuration, MCG includes three parts: F
 *    internal reference clock(MCGIRCLK). Follow the step
 *
 *    1). Call CLOCK_BootToXxxMode to set MCG to target m
 *
 *    2). If target mode is FBI/BLPI/PBI mode, the MCGIRC
 *       correctly. For other modes, need to call CLOCK_
 *       explicitly to setup MCGIRCLK.
 *
 *    3). Don't need to configure FLL explicitly, because
 *       mode, then FLL has been configured by the funct
 *       if the target mode is not FLL mode, the FLL is
 *
 *    4). If target mode is PEE/PBE/PEI/PBI mode, then th
 *       setup by CLOCK_BootToXxxMode. In FBE/FBI/FEE/FE
 *       be enabled independently, call CLOCK_EnablePlle
 *
 * 4. Call CLOCK_SetSimConfig to set the clock configurat
 */

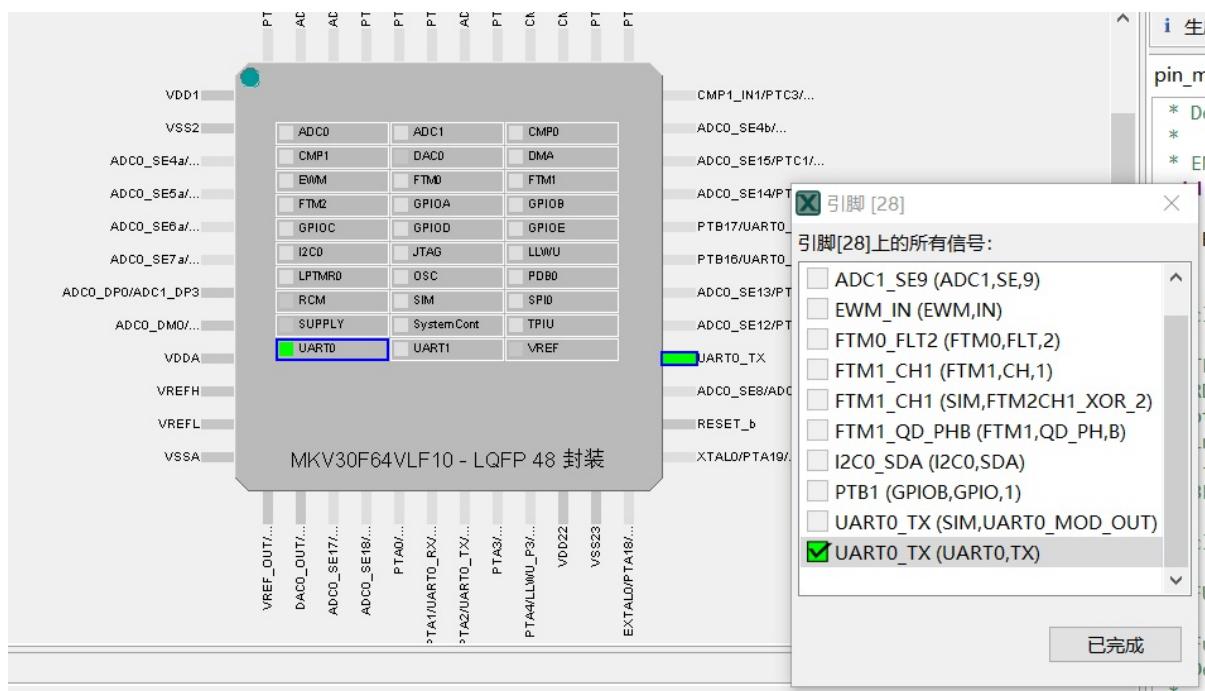
```

## 生成串口输出引脚文件

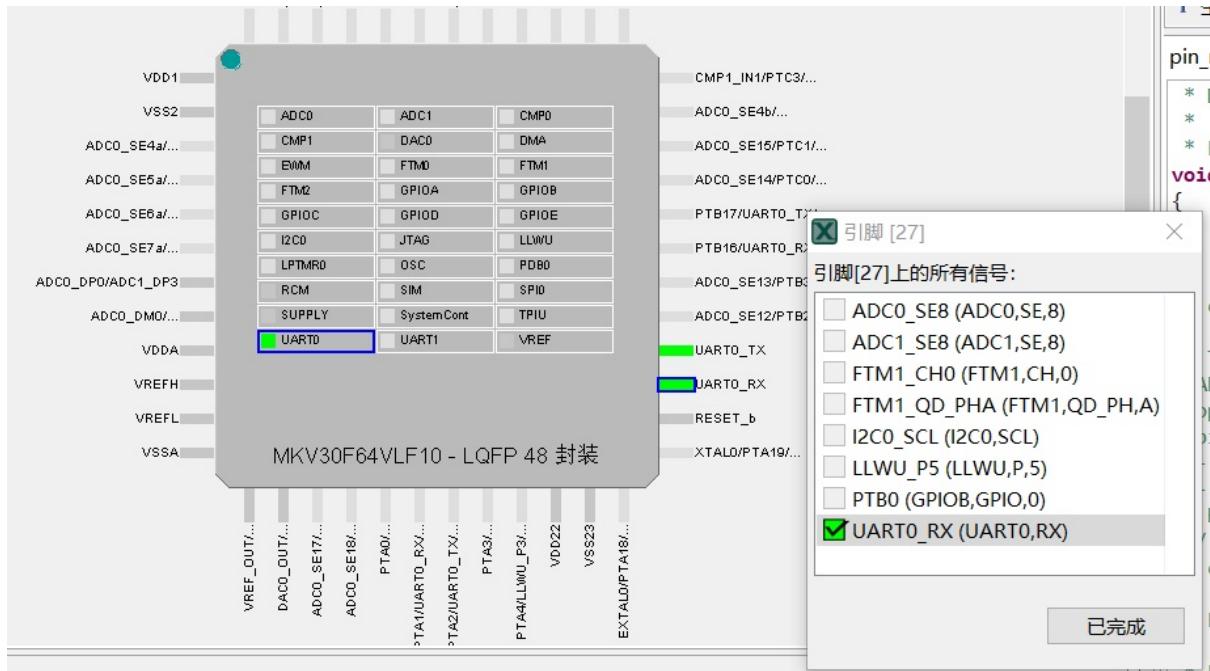
Step 1. 点击工具栏：工具-> 引脚， 默认所有引脚都不配置



Step 2. 这里要配置 TX 串口引脚，该芯片的扩展串口 TX 为 PTB1，点击该引脚然后勾选 UART0\_TX，点击已完成



Step 3. 这里要配置 RX 串口引脚，该芯片的扩展串口 RX 为 PTB0，点击该引脚然后勾选 UART0\_RX，点击已完成



Step 4.点击左方工具栏的源代码，可查看生成的引脚配置文件，点击右方的导出按钮，导出到指定文件夹

**寄存器** **源代码**

**生成的代码预览**

```

pin_mux.c pin_mux.h

/*
 * Description : Calls initialization functions.
 *
 * END ****
void BOARD_InitBootPins(void)
{
    BOARD_InitPins();
}

/* clang-format off */
/*
 * TEXT BELOW IS USED AS SETTING FOR TOOLS ****
BOARD_InitPins:
- options: {callFromInitBoot: 'true', coreID: core0, enableClock: 'true'}
- pin_list:
    - {pin_num: '28', peripheral: UART0, signal: TX, pin_signal: ADC0_SE9/ADC1_SE9/PTB1/I2C0_SCL}
    - {pin_num: '27', peripheral: UART0, signal: RX, pin_signal: ADC0_SE8/ADC1_SE8/PTB0/LLWU_P5}
    * BE CAREFUL MODIFYING THIS COMMENT - IT IS YAML SETTINGS FOR TOOLS ****
*/
/* clang-format on */

/*
 * FUNCTION ****
*
* Function Name : BOARD_InitPins
* Description   : Configures pin routing and optionally pin electrical features.
*
* END ****
void BOARD_InitPins(void)
{
    /* Port B Clock Gate Control: Clock enabled */
    CLOCK_EnableClock(kCLOCK_PortB);
}

```





## 2.4.4.HelloWord 示例

Step 1. 将工程结构树 Board 中的 pin\_mux.c、pin\_mux.h、clock\_config.c、clock\_config.h 文件替换成 MCUXpresso Config Tools 生成的对应文件

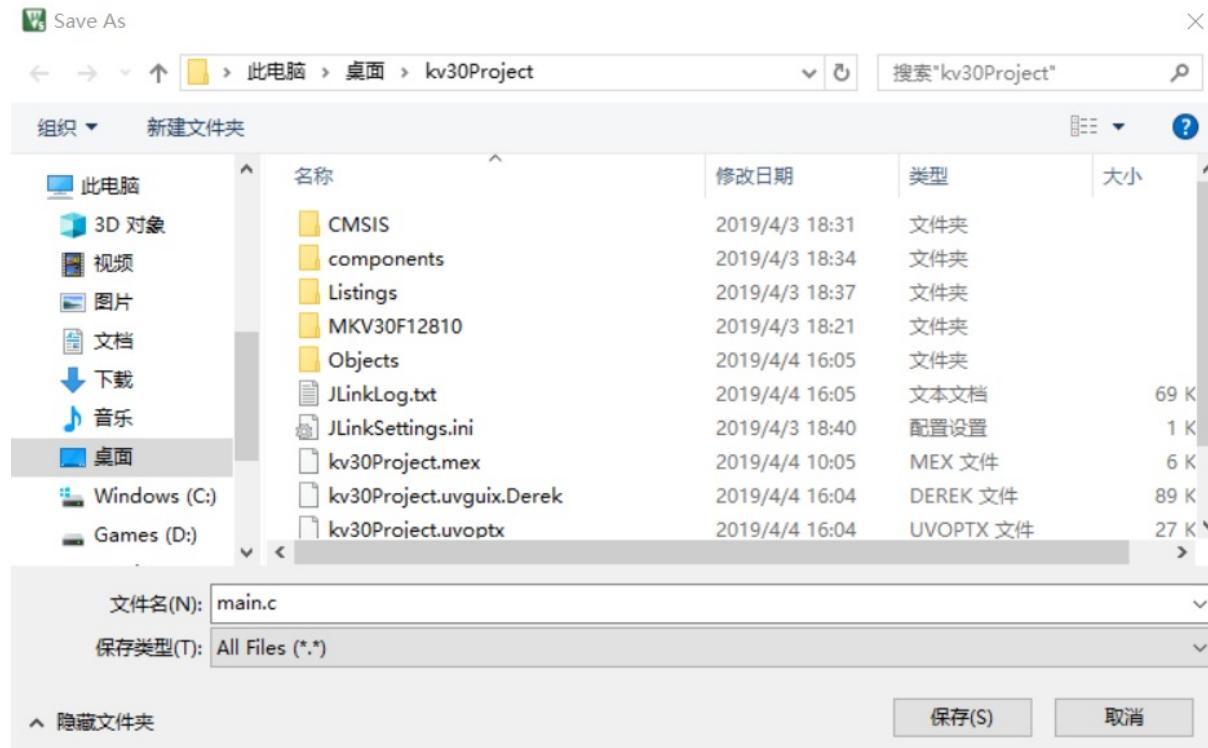
Step 2. 修改 board.c 文件，文件代码如下

```
#include <stdint.h>
#include "board.h"
#include "fsl_common.h"
#include "fsl_debug_console.h"

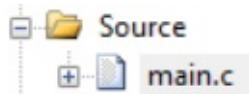
#define BOARD_DEBUG_UART_CLK_FREQ CLOCK_GetCoreSysClkFreq()
#define BOARD_DEBUG_UART_INSTANCE 0U
#ifndef BOARD_DEBUG_UART_BAUDRATE
#define BOARD_DEBUG_UART_BAUDRATE 115200
#endif
#define BOARD_DEBUG_UART_TYPE kSerialPort_Uart

void BOARD_InitDebugConsole(void) {
    uint32_t uartClkSrcFreq = BOARD_DEBUG_UART_CLK_FREQ;
    DbgConsole_Init(BOARD_DEBUG_UART_INSTANCE, BOARD_DEBUG_UART_BAUDRATE, BOARD_DEBUG_UART_TYPE, uartClkSrcFreq);
}
```

Step 3. 点击工具栏的新建文件，生成一个空白 Text 文件，然后点击保存，文件名字修改成 main.c，保存到工程的根目录



Step 4. 双击工程结构树的 Souce 文件夹，将 main.c 文件引入



Step 5.修改 main.c 文件如下

```
#include "board.h"
#include "fsl_debug_console.h"
#include "clock_config.h"
#include "pin_mux.h"

int main(void)
{
    BOARD_InitPins();
    BOARD_BootClockRUN();
    BOARD_InitDebugConsole();
    PRINTF("hello world!\r\n");
    while(1)
    {
    }
}
```

Step 6.编译，烧录代码进板子，串口助手打印以下信息，完成工程文件的搭建





产业首选 · 通路标竿

## 3. 基于 NXP 的 IAR 开发环境搭建

### 概述

IAR Embedded Workbench 是瑞典 IAR Systems 公司为微处理器开发的一个集成开发环境(下面简称 IAR EW), 支持 ARM, AVR, MSP430 等芯片内核平台。

### 特性

1. 高度优化的 IAR ARM C/C++ Compiler
2. IAR ARM Assembler
3. 一个通用的 IAR XLINK Linker
4. IAR XAR 和 XLIB 建库程序和 IAR DLIB C/C++ 运行库
5. 功能强大的编辑器
6. 项目管理器
7. 命令行实用程序
8. IAR C-SPY 调试器(先进的高级语言调试器)



产业首选 · 通路标竿

### 3.1. 下载并安装 IAR

[IAR 下载地址](#)

Step 1. 进入下载页面，下载 IAR 安装包

The screenshot shows the Arm IAR software download page. At the top, there's a large blue header with the Arm logo and the text "Download a free trial". Below it, a sub-header reads "Complete, powerful compiler and debugger tools in a user-friendly IDE—try for yourself!". On the left, a yellow button says "Download Software". Below that, text indicates "(Version 8.32, 1328.42 MB)". To the right, there's descriptive text about the evaluation license and a list of evaluation options. Further down, there's a section titled "Restrictions to the 30-day time-limited evaluation" with a list of restrictions.

**Download a free trial**

Complete, powerful compiler and debugger tools in a user-friendly IDE—try for yourself!

**Download Software**

(Version 8.32, 1328.42 MB)

The evaluation license is completely free of charge and allows you to try the integrated development environment and evaluate its efficiency and ease of use. When you start the product for the first time, you will be asked to register to get your evaluation license.

After installation, you have the following evaluation options to choose from:

- a 30-day time-limited but fully functional license
- a size-limited Kickstart license without any time limit

**Restrictions to the 30-day time-limited evaluation**

- A 30-day time limitation.
- Source code for runtime libraries is not included.
- No support for MISRA C.
- C-RUN is size-limited to 12 Kbytes of code, excluding constant data.
- Limited technical support.
- Must not be used for product development or any other kind of commercial use.

Step 2. 点击安装包，选择 Install IAR Embedded Workbench for Arm



**IAR Embedded  
Workbench**

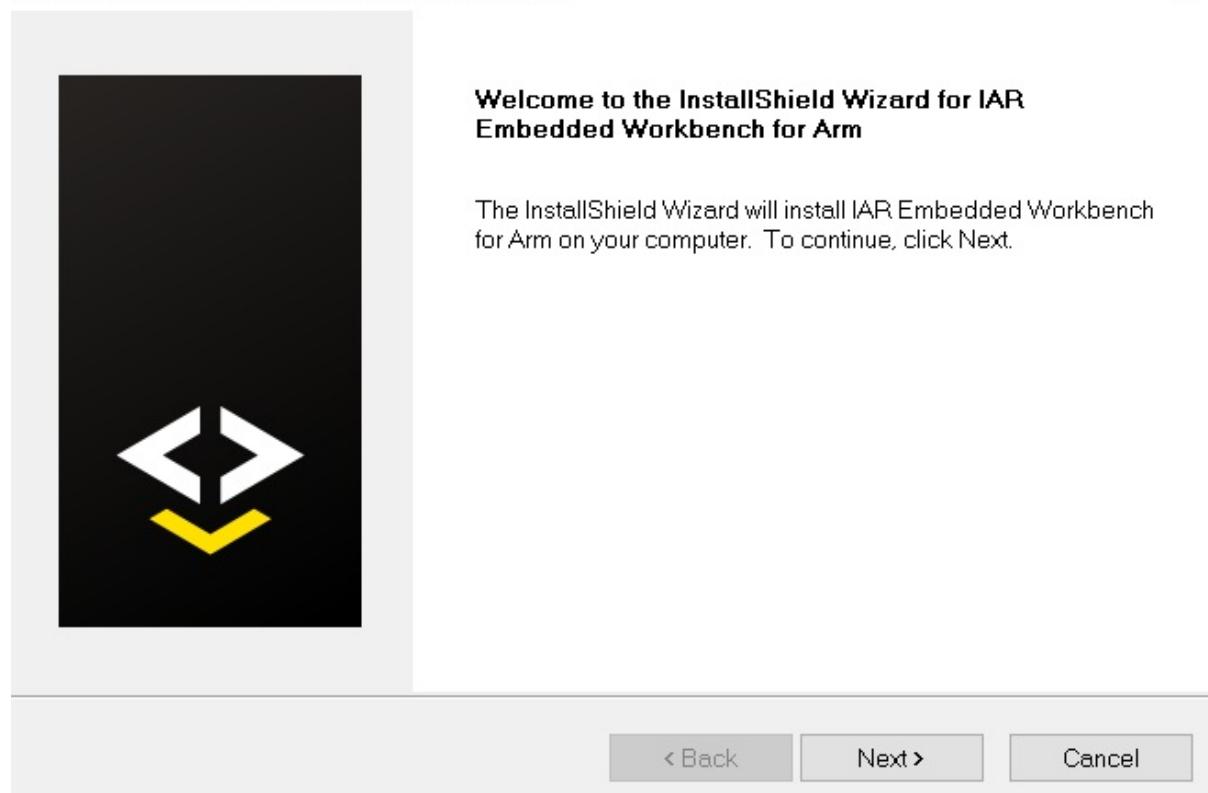
**IAR**  
SYSTEMS

- Install IAR Embedded Workbench® for Arm
- Installation and licensing information
- Release notes
- Install drivers
- Explore the installation media
- Exit

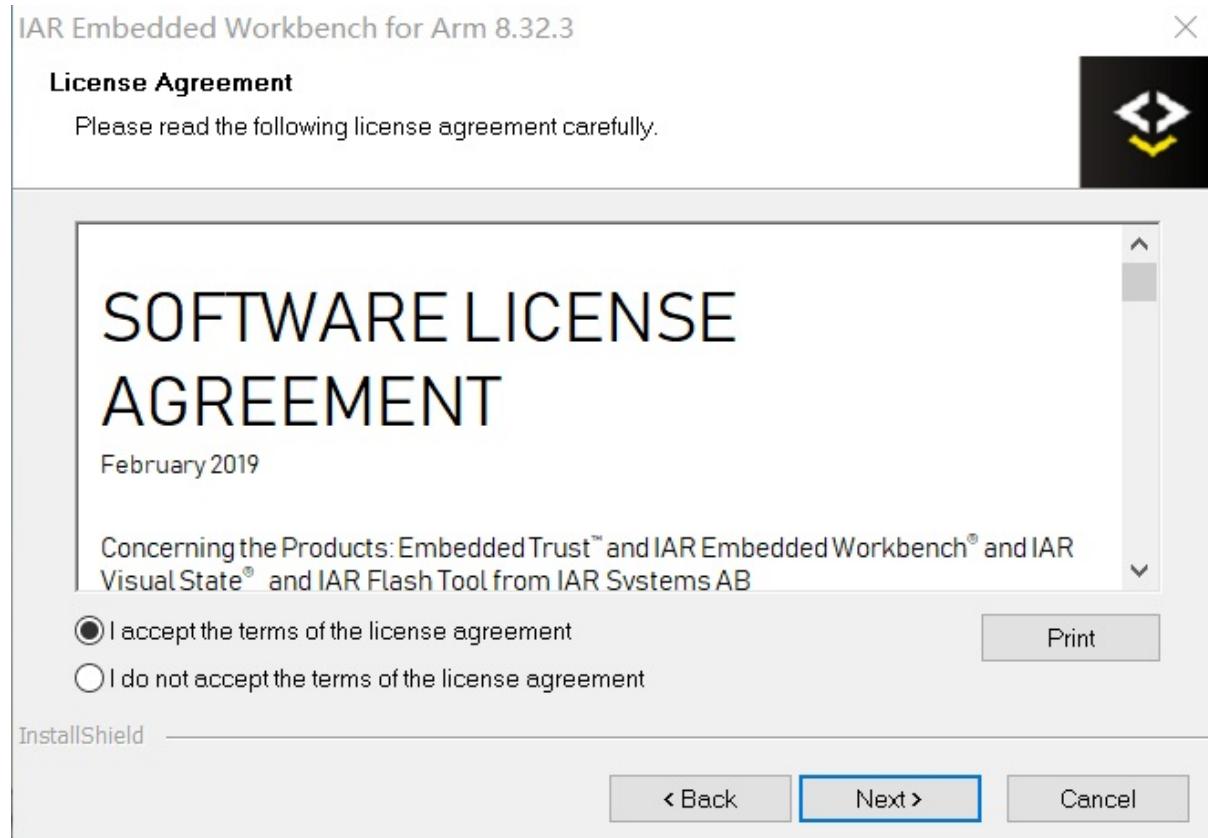
Japanese

Step 3.点击 Next

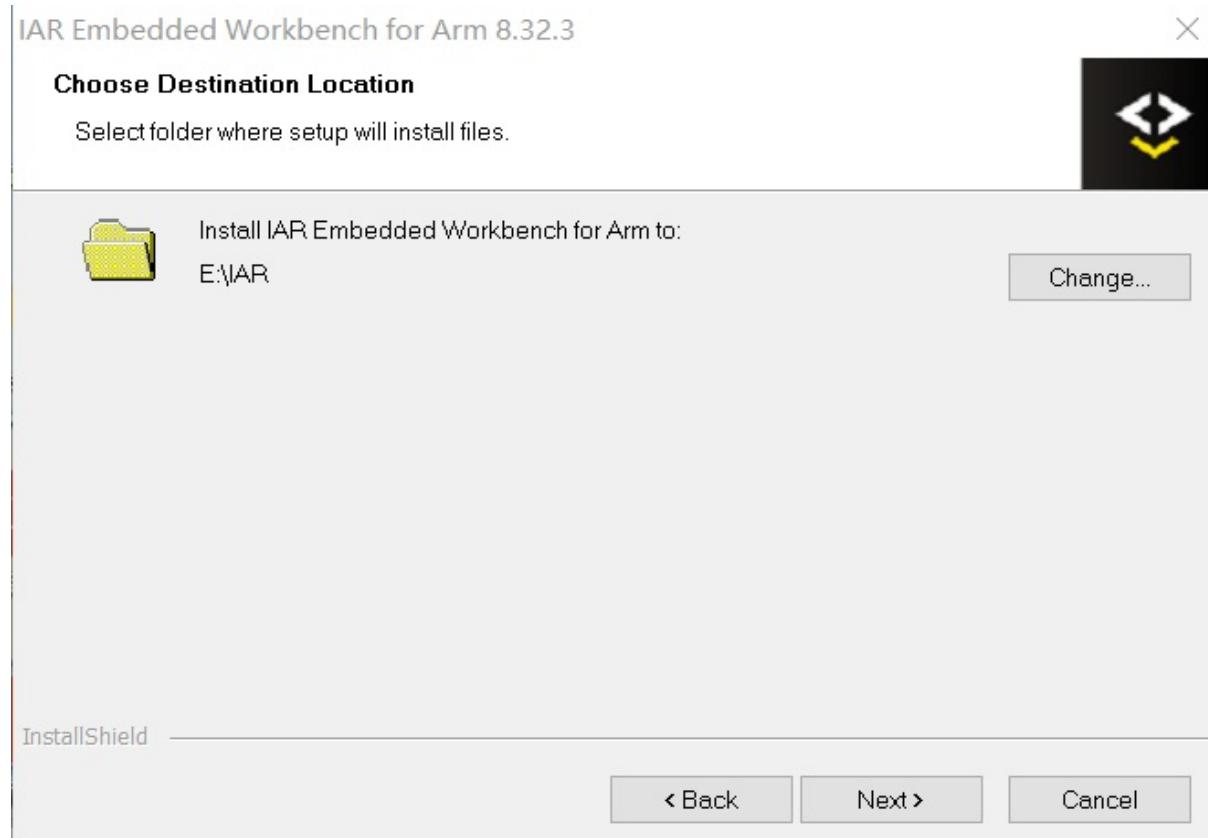
IAR Embedded Workbench for Arm 8.32.3



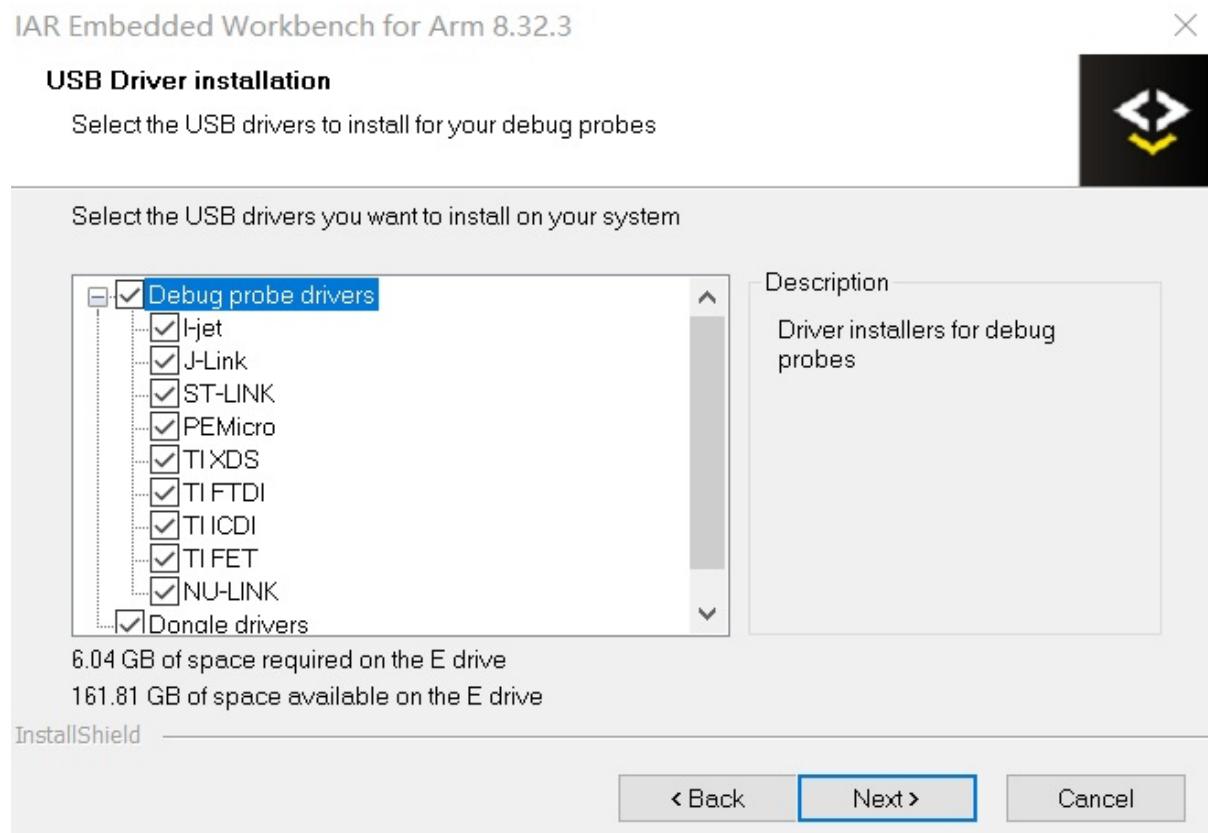
Step 4.选择 I accept, 点击 Next



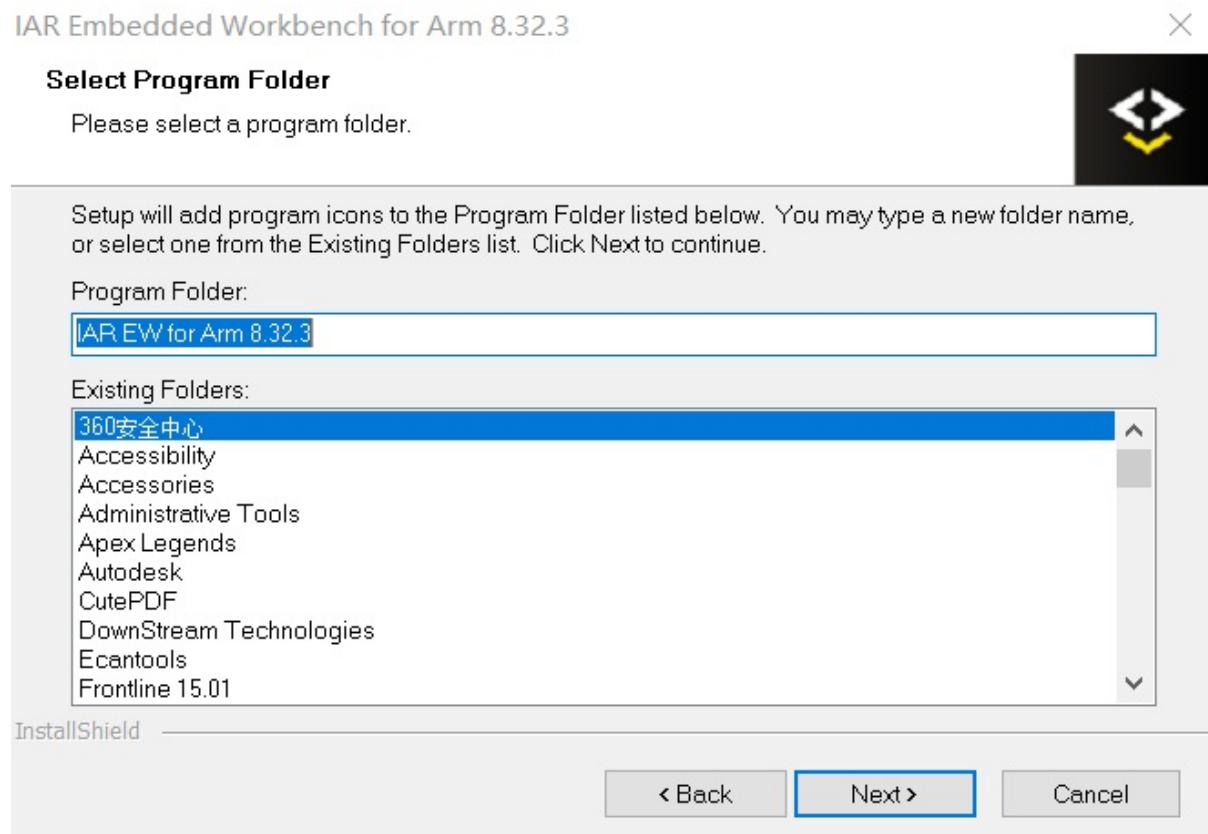
Step 5.选择安装路径, 点击 Next



Step 6.选择调试工具，点击 Next



Step 7.快捷方式文件夹命名，点击 Next



Step 8.移除所有的 USB Dongle, 点击确定

IAR Embedded Workbench for Arm - InstallShield Wizard X



IAR Systems Setup will now install dongle drivers on your system.

The driver installation for USB dongles might fail if a USB dongle is attached to your computer during the installation.

You must remove all USB dongles before continuing.

确定

Step 9.点击 Install

IAR Embedded Workbench for Arm 8.32.3 X

**Ready to Install the Program**

The wizard is ready to begin installation.



Click Install to begin the installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

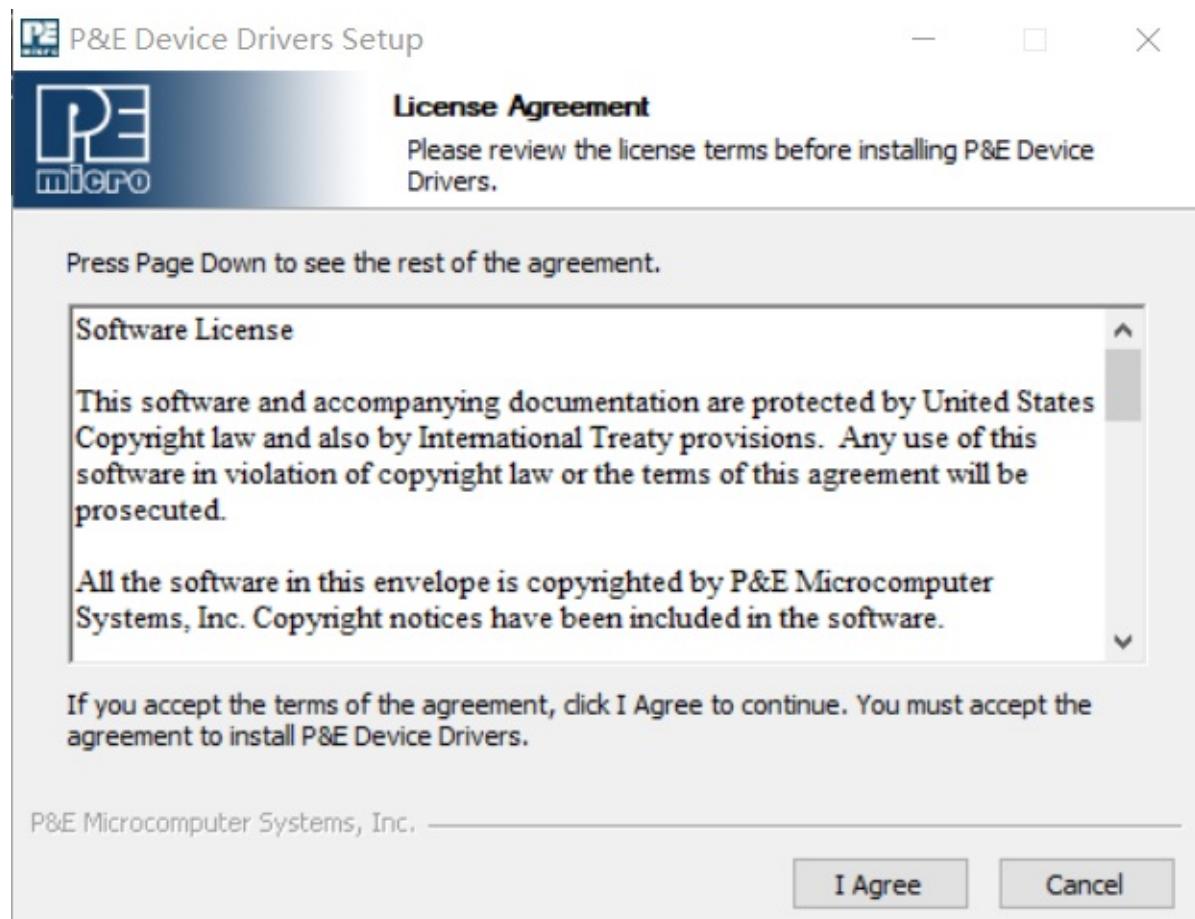
InstallShield

< Back

Install

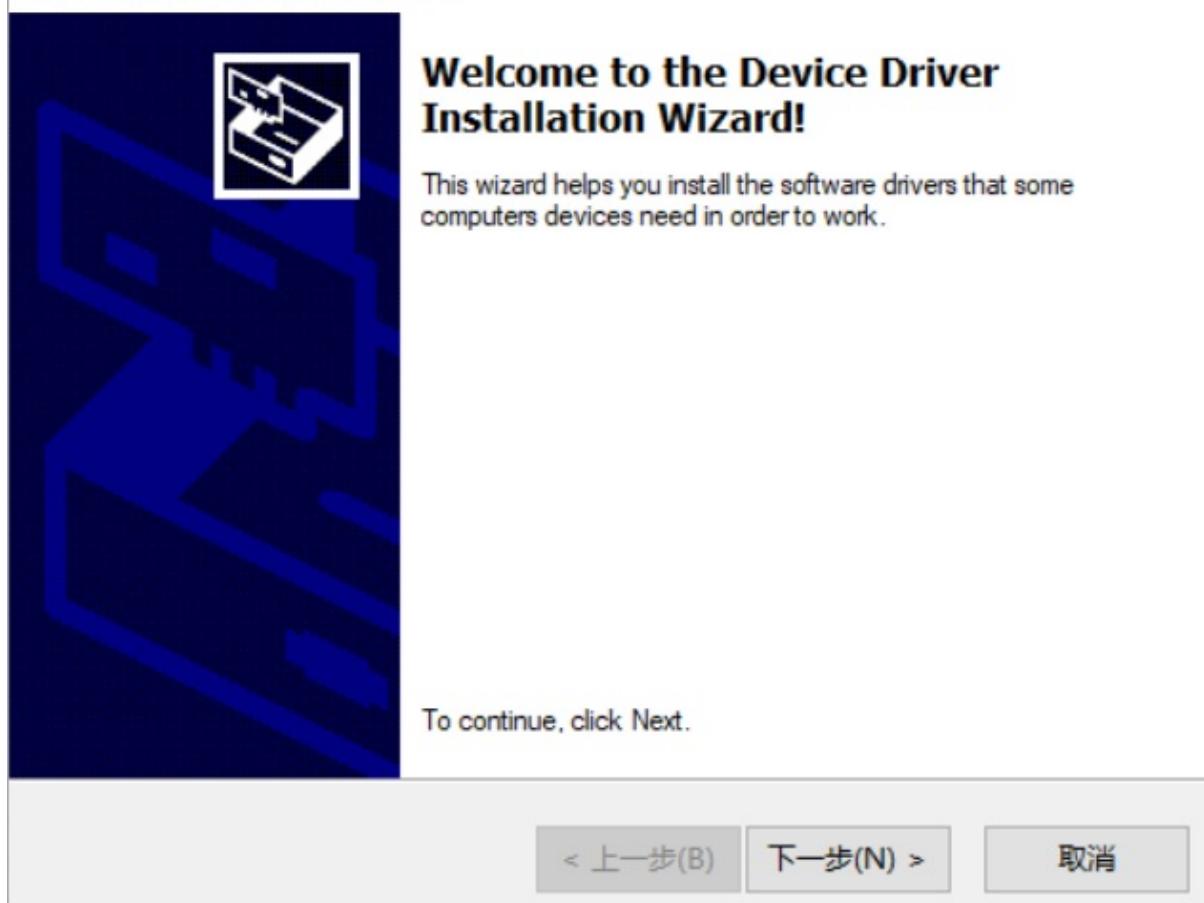
Cancel

Step 10.点击 I Agree



Step 11. 安装设备驱动，点击下一步

Device Driver Installation Wizard

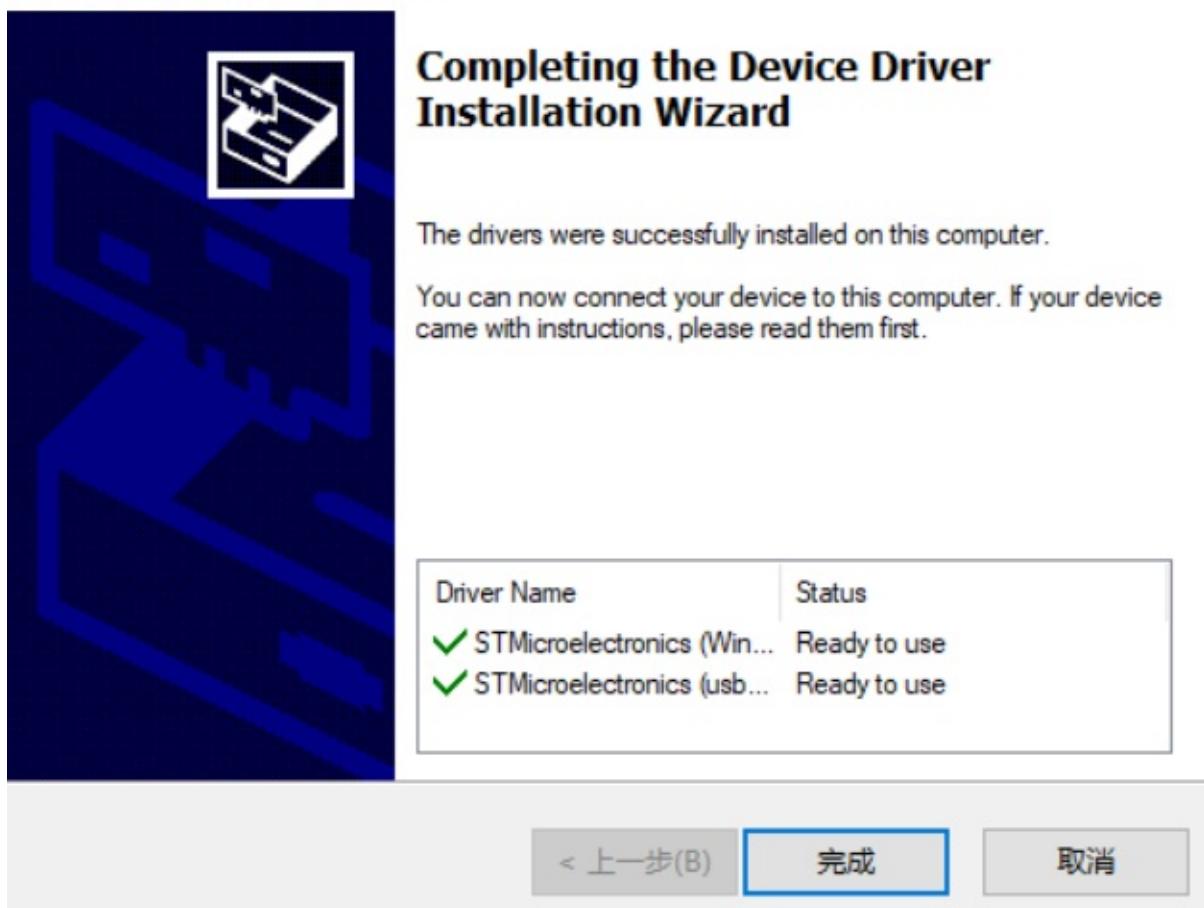


Step 12. 点击安装

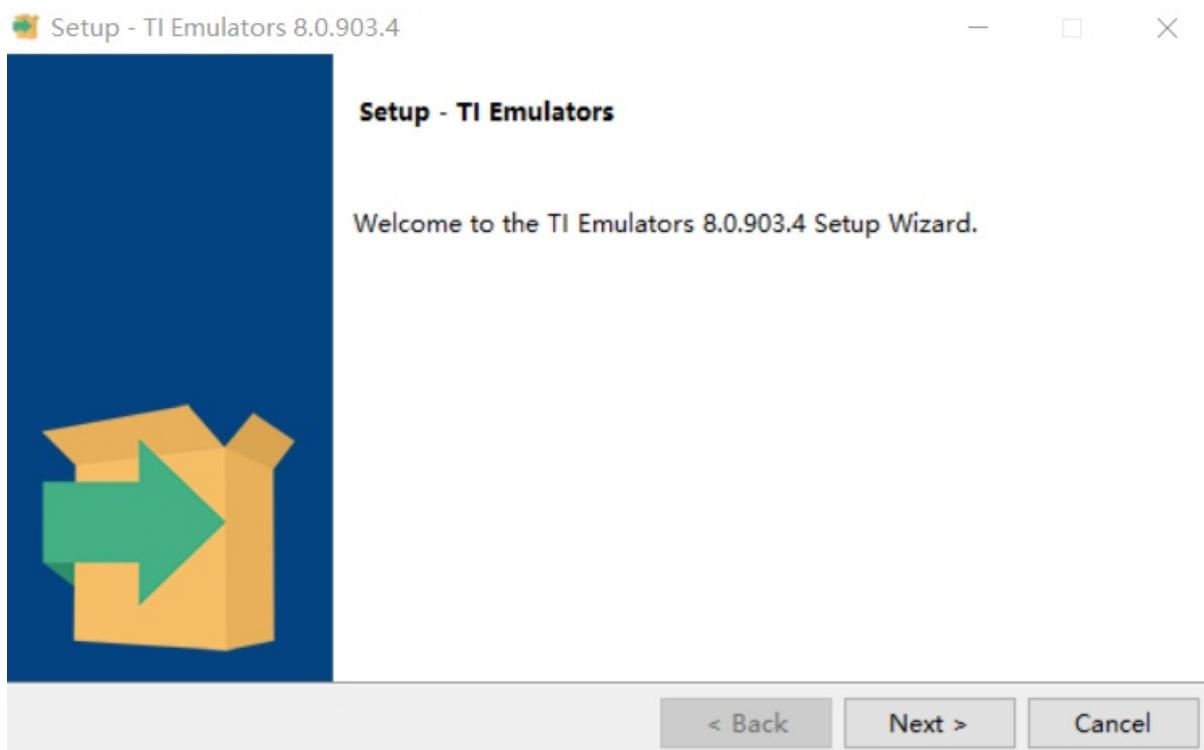


Step 13. 安装设备驱动，点击下一步

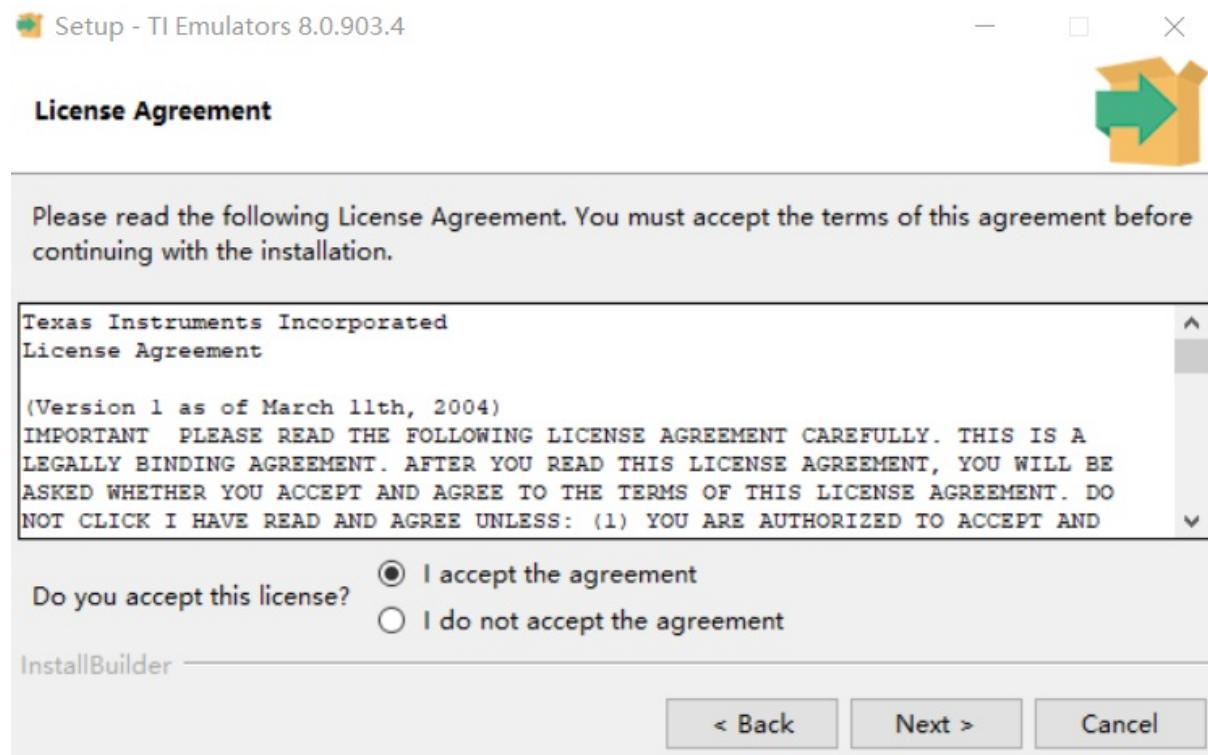
## Device Driver Installation Wizard



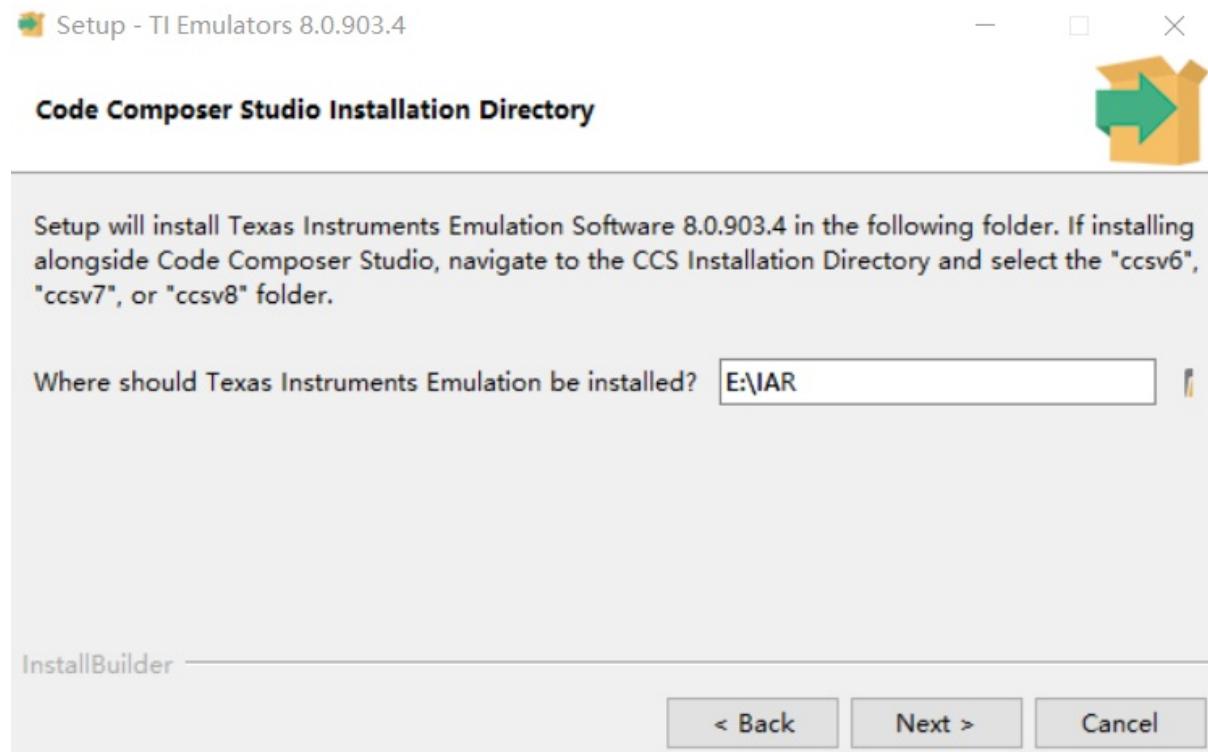
Step 14.点击安装



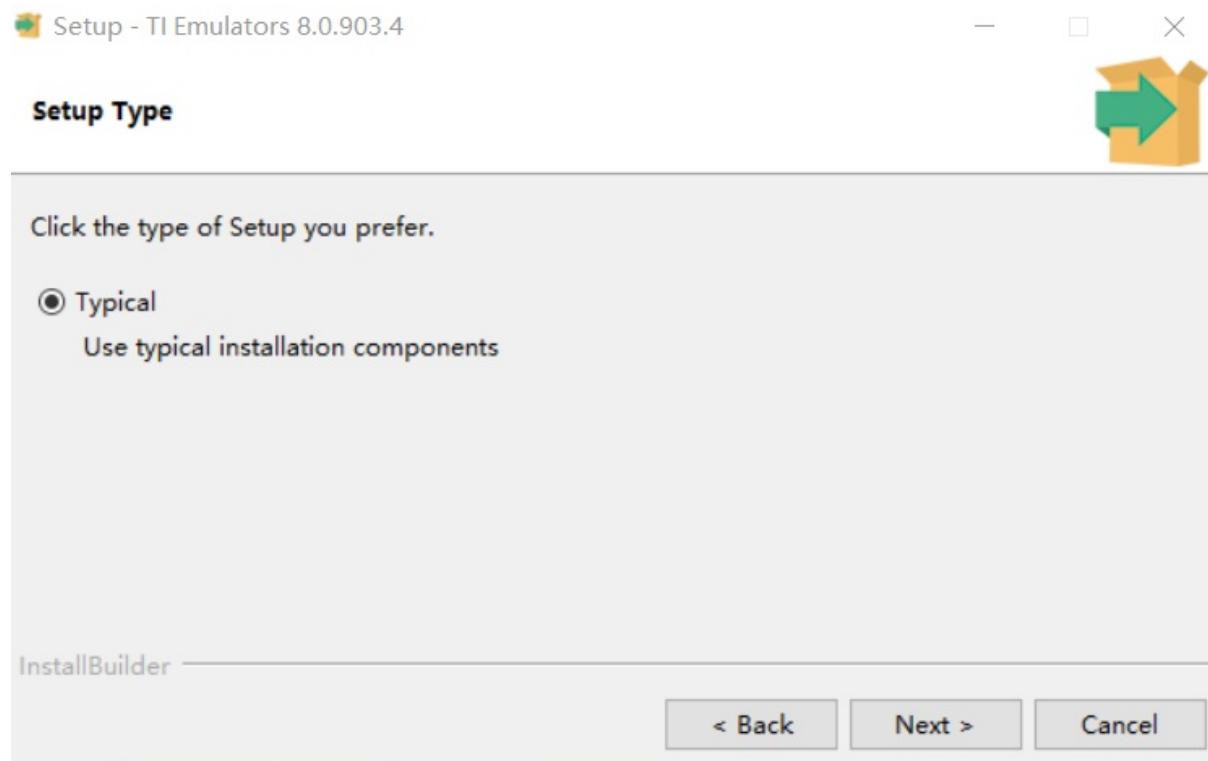
Step 15.选择 I accept, 点击 Next



Step 16.选择安装路径，点击 Next



Step 17.点击 Next



Step 18.点击安装





产业首选 · 通路标竿

## 3.2. 下载 IAR SDK

[SDK 下载地址](#)

需注册 NXP 官网账号

Step 1. 进入下载地址, 搜索你要下载芯片的 SDK, 这里以 MKV30F64xxx10 为例, 选择 MKV30F64xxx10, 右方会显示该芯片的信息, 然后点击 Build MCUXpresso SDK

Hardware Details	
Included Part Numbers	MKV30F64VLF10, MKV30F64VFM10, MKV30F64VLH10
Board(s)	TWR-KV31F120M, FRDM-KV31F
Device	MKV30F12810
Core Type / Max Freq	Cortex-M4F / 100MHz
Device Memory Size	64 KB Flash 16 KB RAM

Step 2. 选择 Windows 主机系统, 下载的工具链为 IAR Embedded Workbench for ARM, 组件默认即可, 若有特别的组件需求可自己添加进去, 点击 Download 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*

Host OS

Windows

Toolchain / IDE

IAR Embedded Workbench for ARM

## Select Optional Middleware

Add middleware, operating systems, and software libraries to your SDK.

Add software component

This MCUXpresso SDK configuration is available for direct download

Download SDK

Archive Name

SDK\_2.5.0\_MKV30F64xxx10

Don't use: <,>, :, ", /, |, ?, \*, \ in the name of your  
SDK

Step 3.点击 I Agree

## SDK Downloads

x

### Software Terms and Conditions

Please read the following agreement and click "I AGREE" at the bottom before downloading your software.

[EULA](#)

[Software Content Register](#)

LA\_OPT\_NXP\_Software\_License v4 November 2018

**IMPORTANT.** Read the following NXP Software License Agreement ("Agreement") completely. By selecting the "I Accept" button at the end of this page, or by downloading, installing, or using the Licensed Software, you indicate that you accept the terms of the Agreement and you acknowledge that you have the authority, for yourself or on behalf of your company, to bind your company to these terms. You may then download or install the file. In the event of a conflict between the terms of this Agreement and any license terms and conditions for NXP's proprietary software embedded anywhere in the Licensed Software file, the terms of this Agreement shall control. If a separate license agreement for the Licensed Software has been signed by you and NXP, then that agreement shall govern your use of the Licensed Software and shall supersede this Agreement.

#### NXP SOFTWARE LICENSE AGREEMENT

This is a legal agreement between your employer, of which you are an authorized representative, or, if you have no employer, you as an individual ("you" or "Licensee"), and NXP B.V. ("NXP"). It concerns your rights to use the software provided to you in binary or source code form and any accompanying written materials (the "Licensed Software"). The Licensed Software may include any updates or error corrections or documentation relating to the Licensed Software provided to you by NXP under this Agreement. In consideration for NXP allowing you to access the Licensed Software, you are agreeing to be bound by the terms of this Agreement. If you do not agree to all of the terms of this Agreement, do not download or install the Licensed Software. If you change

[I Agree](#)

[← Back](#)

[Close](#)

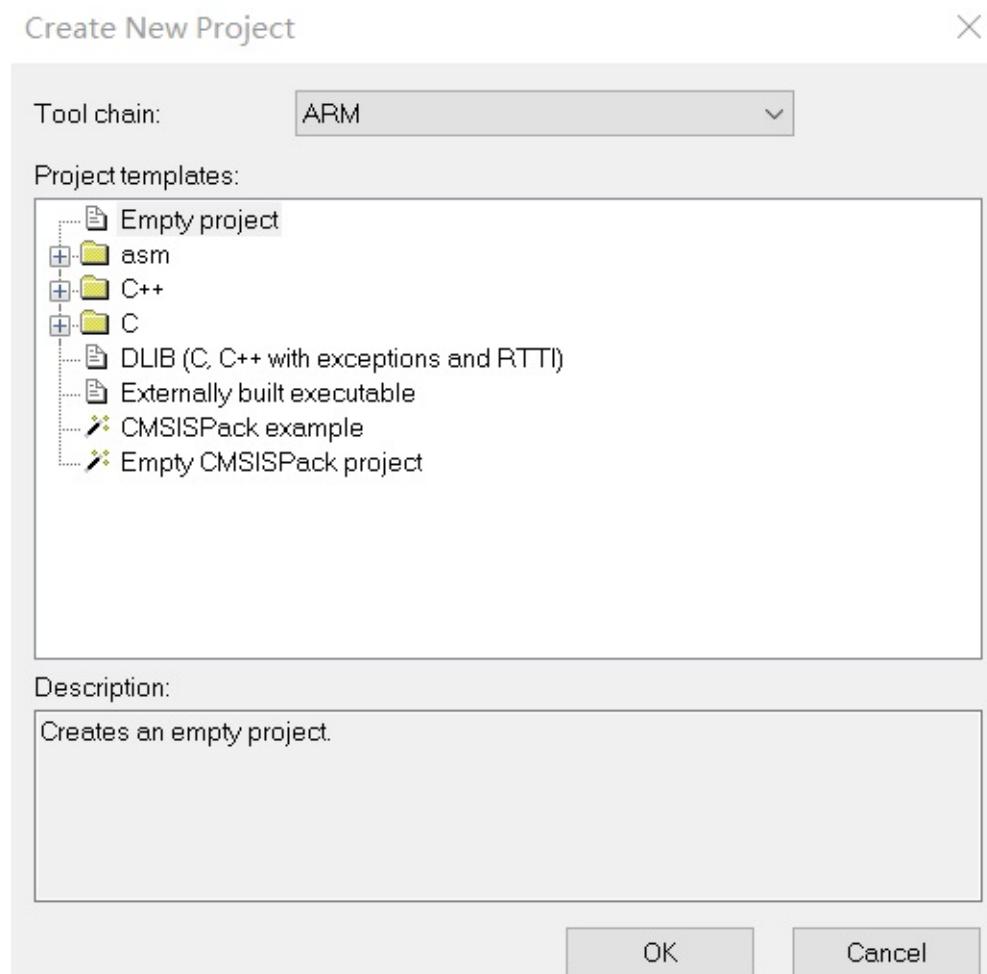


产业首选·通路标竿

### 3.3.IAR 工程搭建

Step 1. 在指定位置新建工程文件夹，指定名字为 iarProject

Step 2. 双击下载好的 IAR，点击工具栏 Project -> Create New Project...，选择工具链为 ARM，选择 Empty project，点击 OK，保存工程文件到上一步新建的工程文件夹中，取名为 iarProject



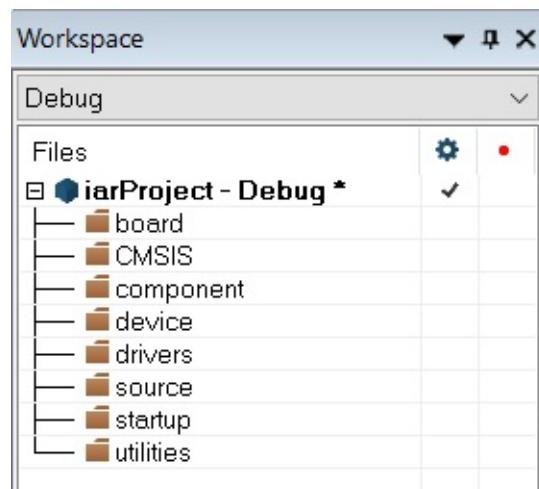


### 3.3.1.添加目录结构树

Step 1.左边的工程结构树如下，修改结构树，在结构树文件夹右键 Add Group 则可以添加文件夹



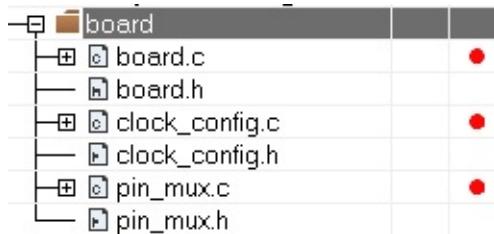
Step 2.工程结构树修改完成，如图



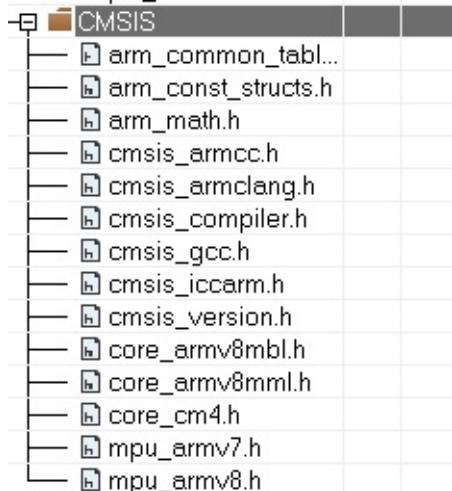
Step 3.将下载的 SDK 解压，并将 SDK 根目录下的 components、CMSIS、devices/MKV30F12810 复制到新建的工程文件夹下，如图

名称	修改日期	类型
CMSIS	2019/4/15 13:57	文件夹
components	2019/4/15 13:57	文件夹
MKV30F12810	2019/4/15 13:58	文件夹
settings	2019/4/15 13:50	文件夹

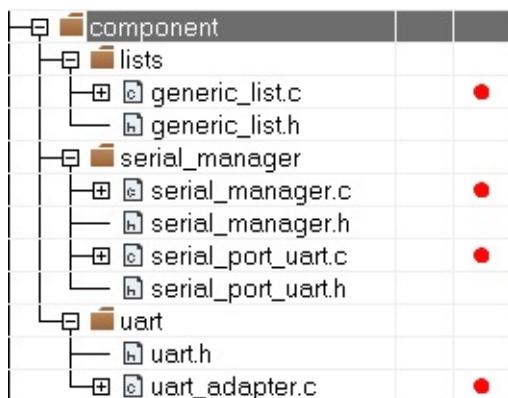
Step 4.右键工程结构树的 Board，添加以下文件，文件位于 ./MKV30F12810/project\_template



Step 5.右键工程结构树的 CMSIS，添加以下文件，文件位于 ./CMSIS/Include



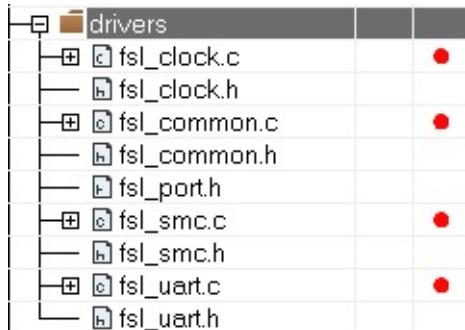
Step 6.右键工程结构树的 component，添加以下文件，文件位于  
./component/lists、./component/serial\_manager、./component/uart



Step 7.右键工程结构树的 device，添加以下文件，文件位于 ./MKV30F12810/



Step 8.右键工程结构树的 drivers，添加以下文件，文件位于 ./MKV30F12810/drivers

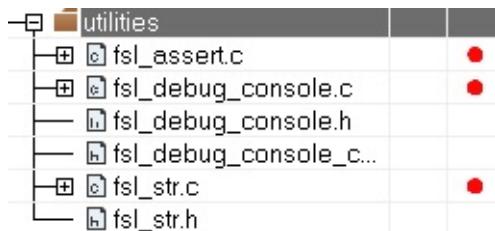


Step 9.右键工程结构树的 startup, 添加以下文件, 文件位于 ./MKV30F12810/iar



Step 10.右键工程结构树的 utilities, 添加以下文件, 文件位于

./MKV30F12810/utilities、./MKV30F12810/utilities/debug\_console、./MKV30F12810/utilities/str



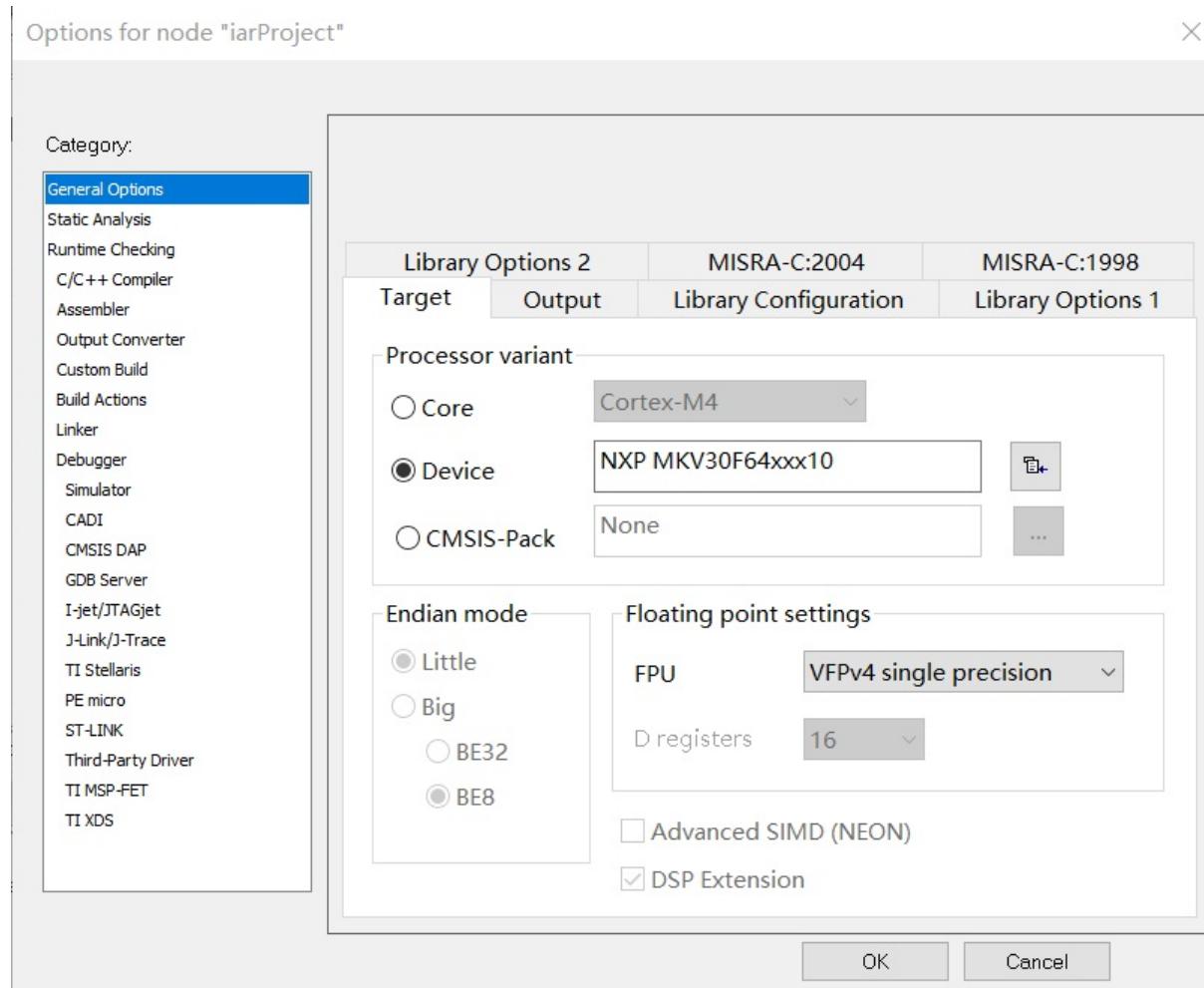


产业首选·通路标竿

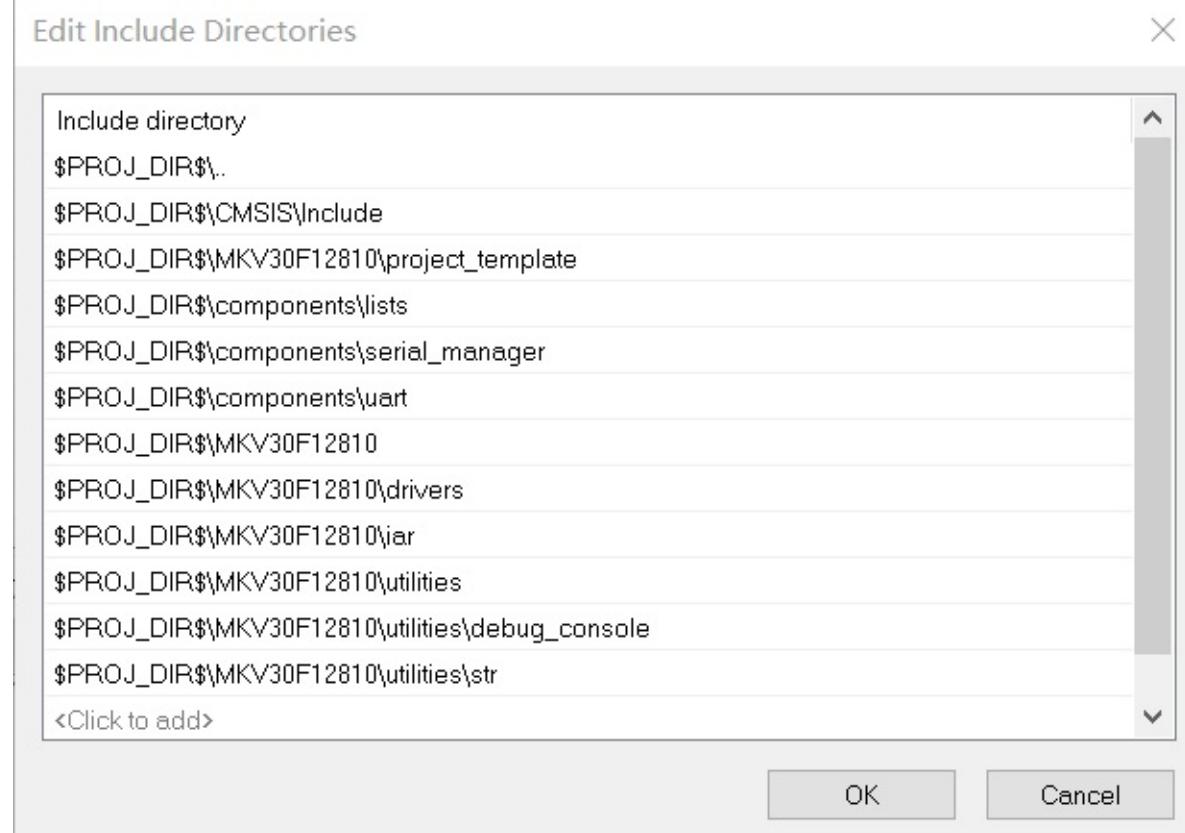
### 3.3.2.修改 Options

Step 1.右键工程结构树最上层目录(iarProject), 点击 Options

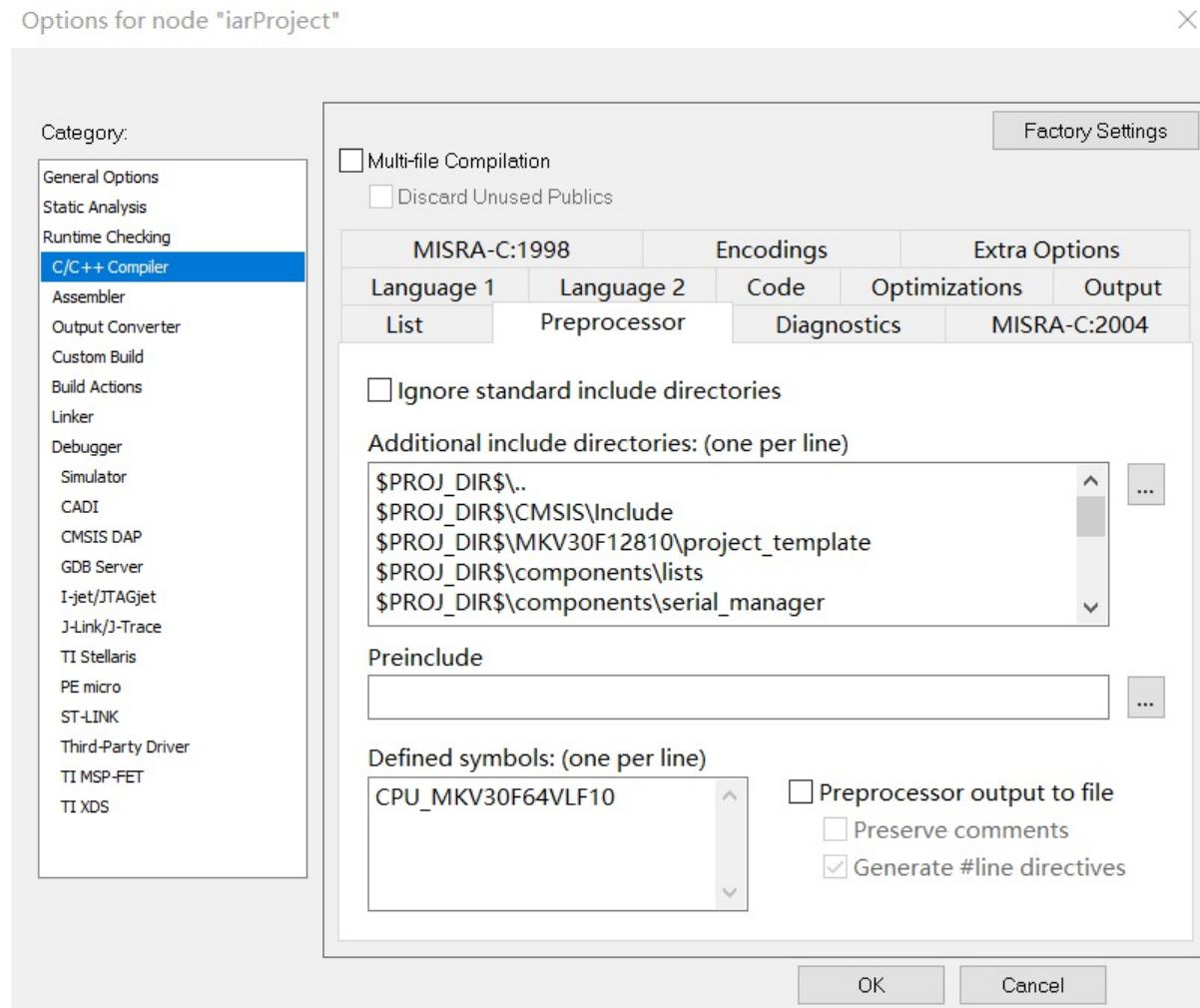
Step 2.General Options->Target,Processor variant 选择 Device,NXP MKV30F64xxx10



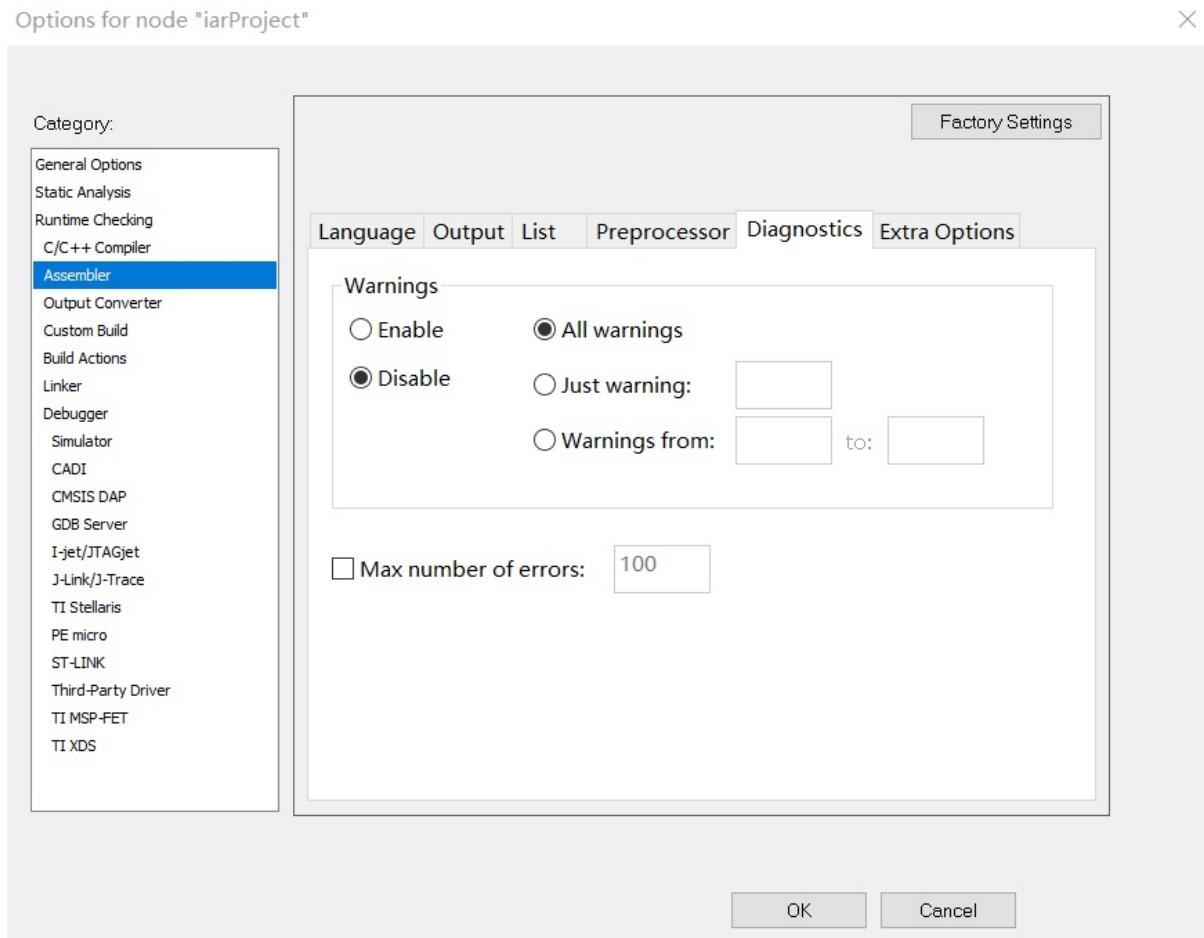
Step 3.C/C++ Complier->Preprocessor,添加文件路径



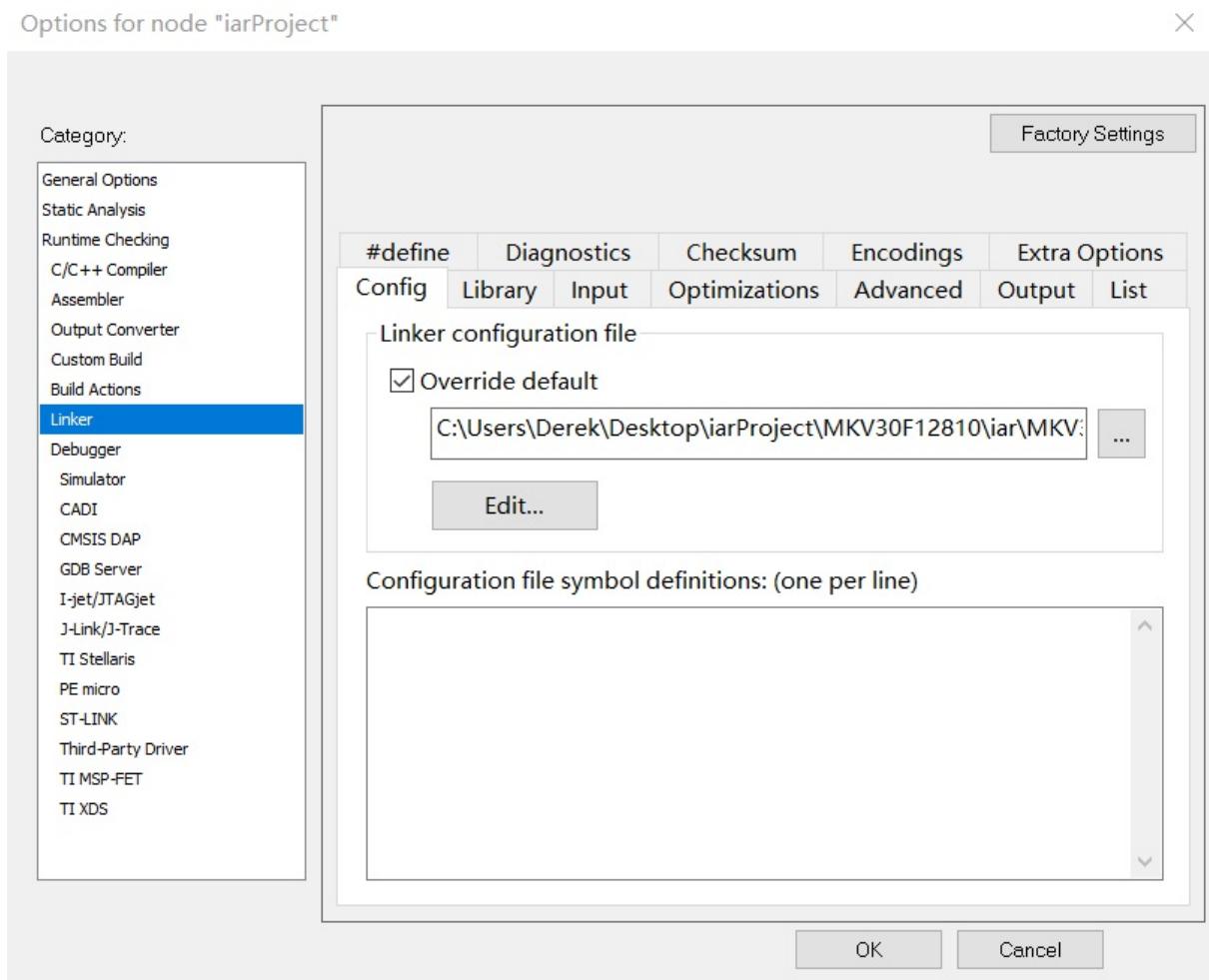
Step 4.C/C++ Complier->Preprocessor,添加芯片定义



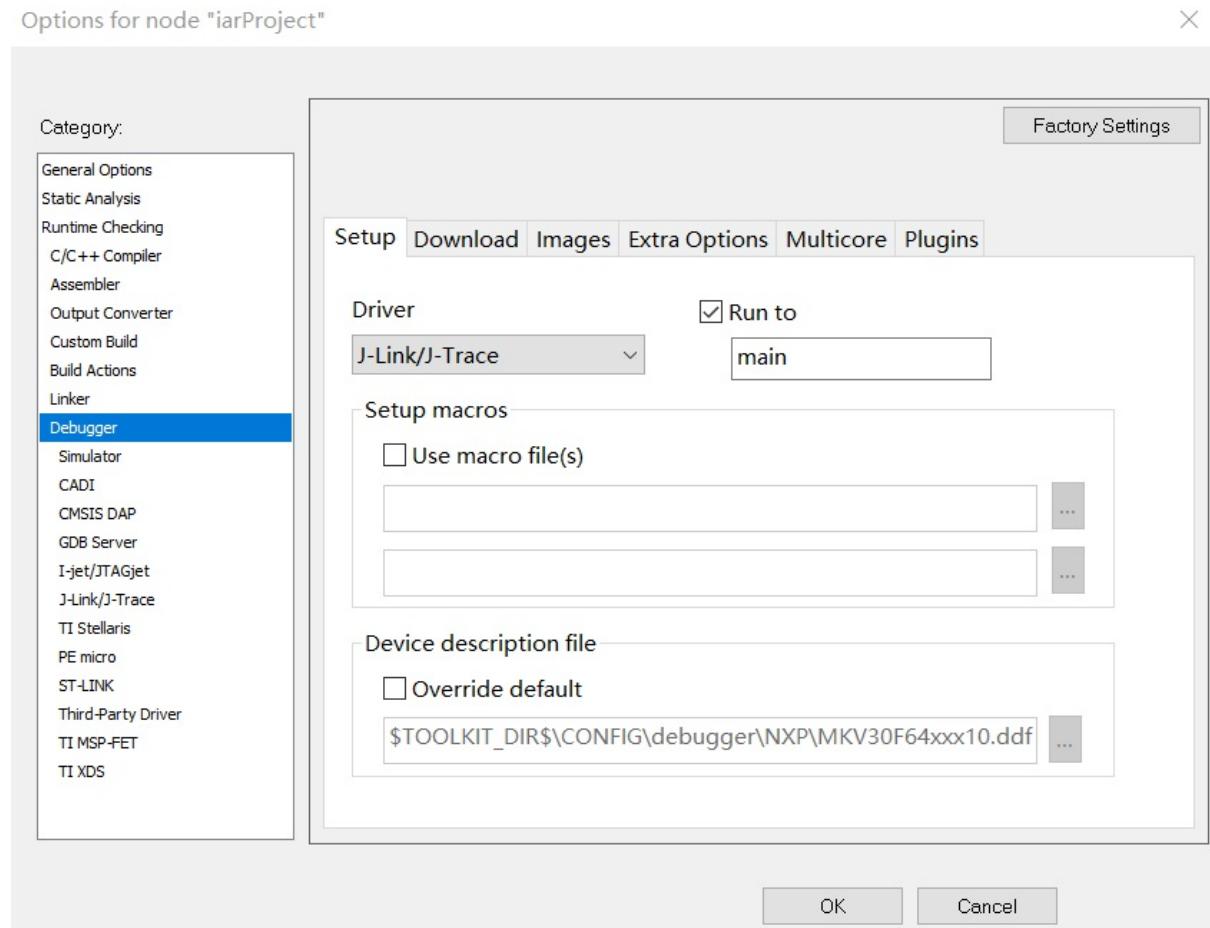
Step 5. Assembler-&gt;Diagnostics, Warnings Disable



Step 6. Linker->Config, 勾选 Override default, 选择 MKV30F64xxx10\_flash.icf 文件, 文件位于 ./MKV30F12810/iar



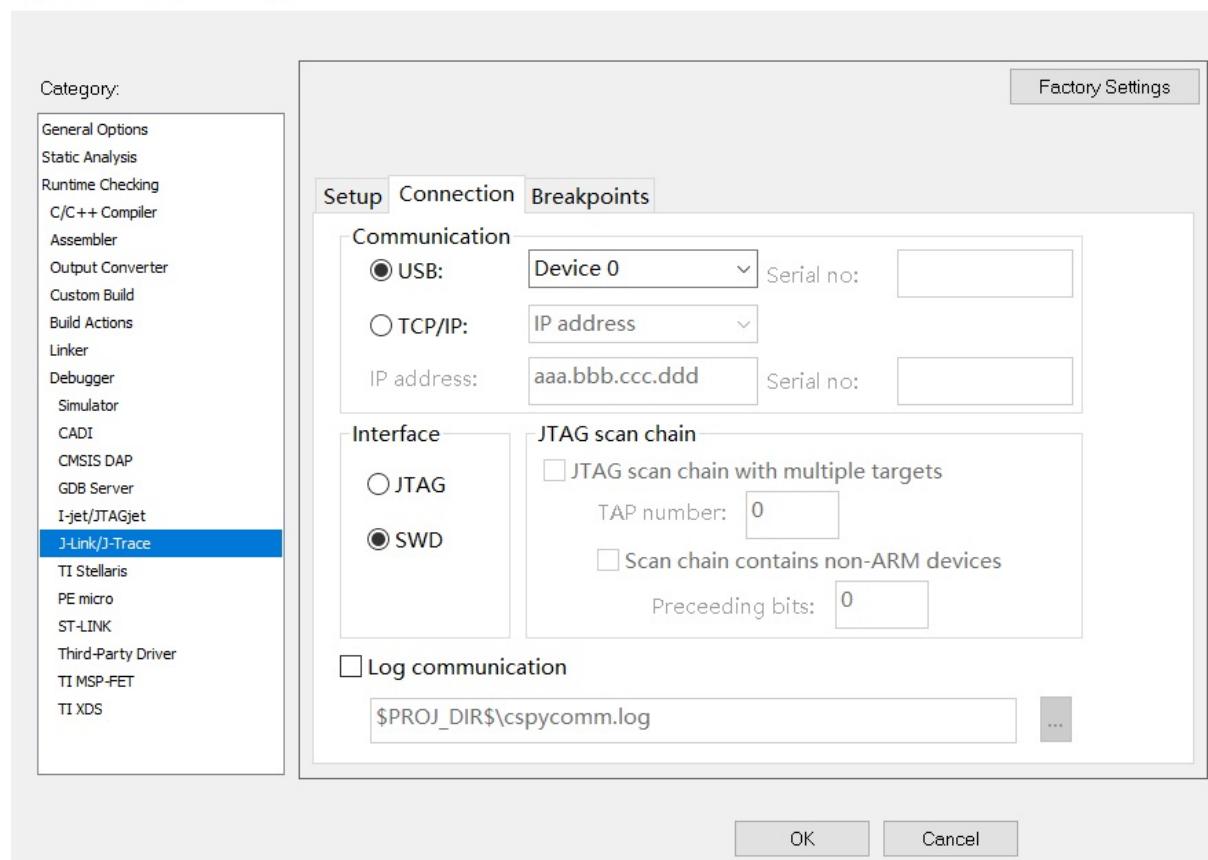
Step 7. Debugger->Setup, Driver 选择 J-Link/J-Trace



Step 8. Debugger.J-Link/J-Trace->Connection, Driver Interface 选择 SWD

## Options for node "iarProject"

X

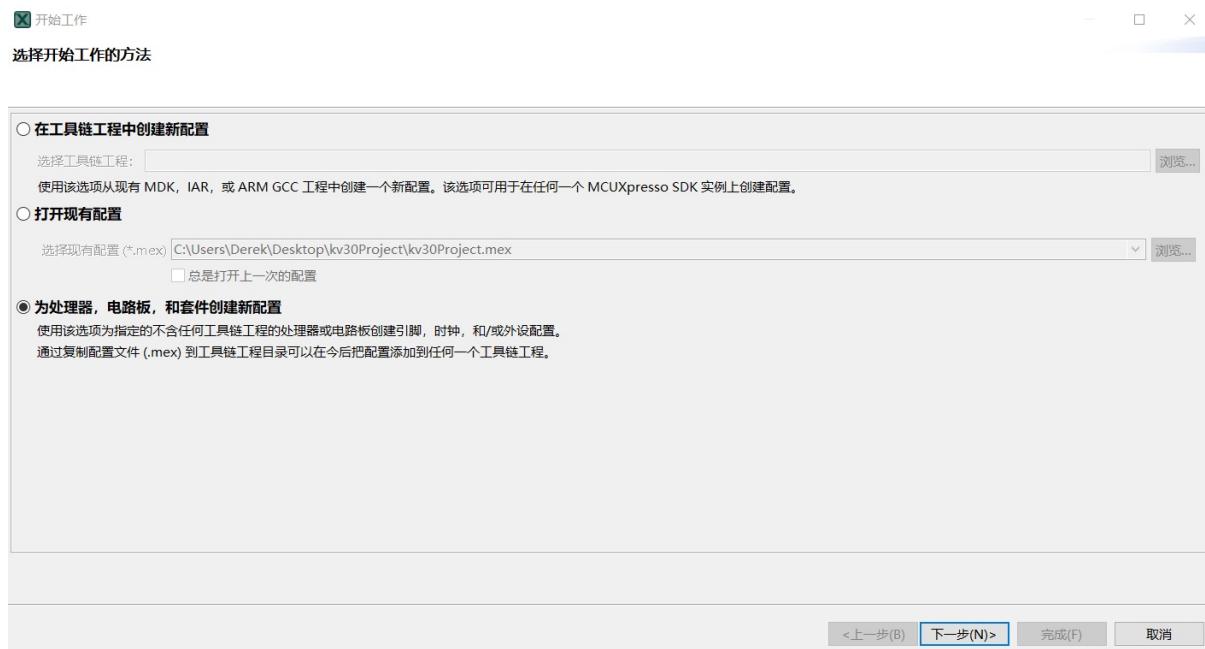




### 3.3.3. 使用 MCUXpresso Config Tools 生成代码

#### 生成配置外部晶振的时钟文件

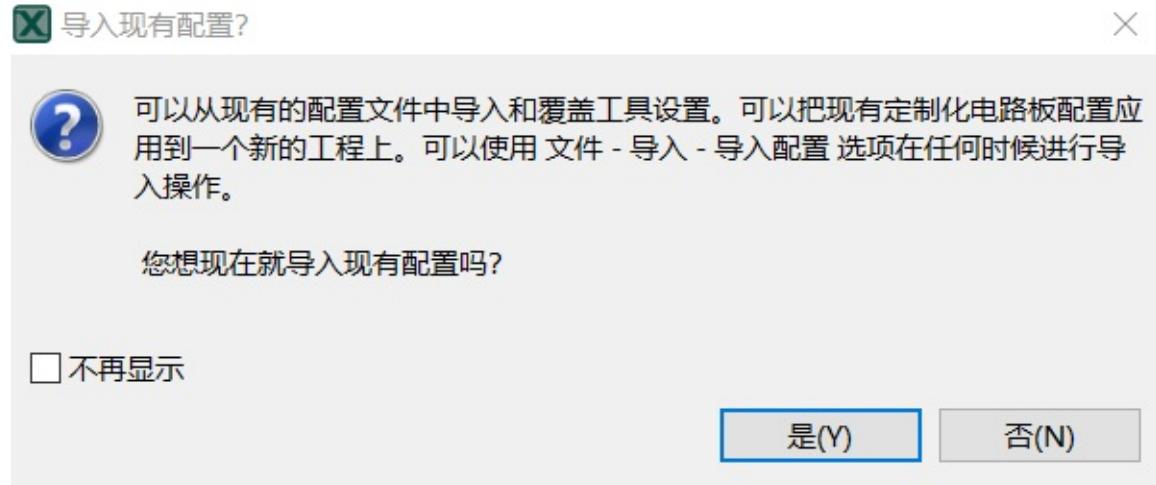
Step 1. 打开 MCUXpresso Config Tools，选择为处理器，电路板，和套件创建新配置，点击下一步



Step 2. 查询你配置的芯片，在这里为 MKV30F64xxx，封装为 MKV30F64VLF10 - LQFP 48 封装



Step 3. 点击否



Step 4. 点击工具栏：工具->时钟，该芯片默认配置为内部时钟，点击外部 OSC 的 Inactive 框

时钟源	
名称	可 值
内部	
FAST_IRCLK	4 MHz
SLOW_IRCLK	32.768 kHz
IRC48M	Inactive
LPO	1 kHz
外部	
<input checked="" type="checkbox"/> OSC (System Oscillator)	<input type="checkbox"/> Inactive

Step 5. 输入芯片的外部晶振参数，在这里使用的是无源的 8MHz 晶振，点击 OK



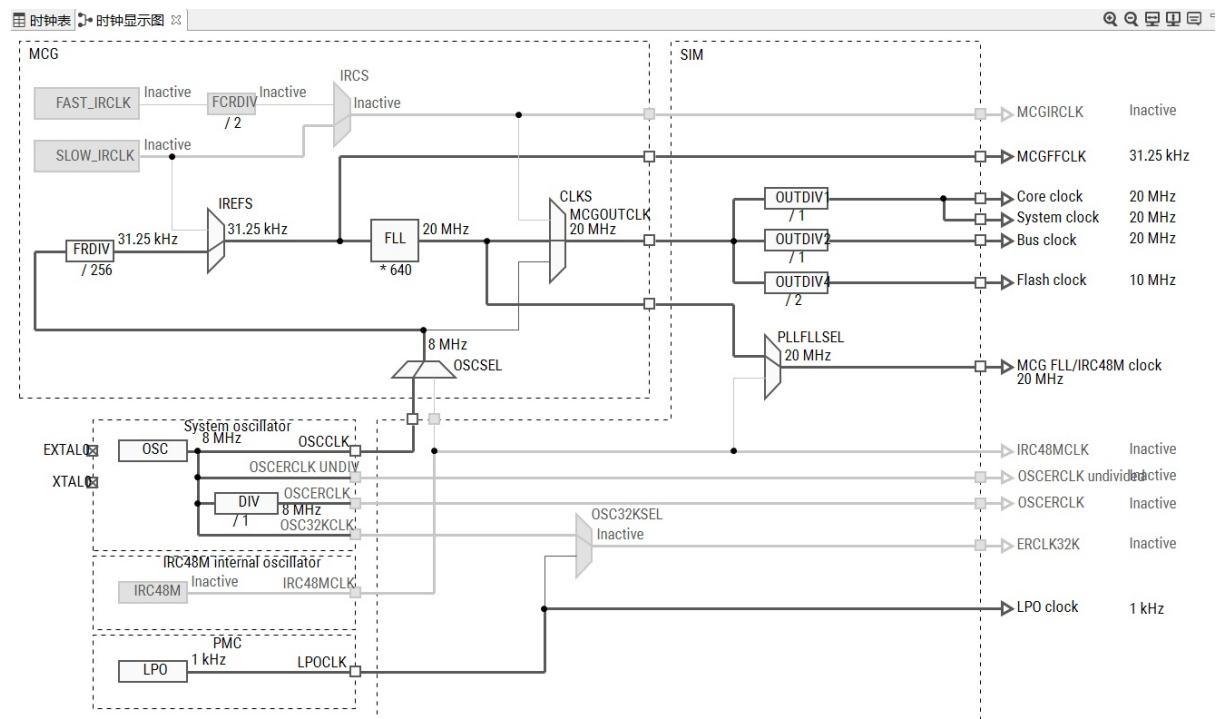
Step 6. 将外部 OSC 下拉框展开，OSC mode 选择 Using oscillator with external crystal(low power)，Frequency Range 选择 High frequency range 3-8 MHz，System Osc.Capacity Load 选择 0pF (芯片内部晶振的负载电容)

外部	
<input checked="" type="checkbox"/> OSC (System Oscillator)	<input checked="" type="checkbox"/> 8 MHz
OSC mode	Using oscillator...tal (low power)
Frequency Range	High frequency range 3-8 MHz
System O...ty Load	0 pF

Step 7. 工具栏的 MCG Mode 从 FEI 模式切换成 FEE 模式

## MCG Mode FEE (FLL Engaged External)

Step 8. 观察时钟显示图，时钟已配置成外部晶振输入



Step 9. 右方工具栏点击源代码，可查看自动生成的时钟代码文件，右方有个导出按钮，将时钟的 .c 和 .h 文件导出到指定目录

三 详情 源代码 寄存器 模块时钟

i 生成的代码预览

clock\_config.c clock\_config.h

```
/****************************************************************************
 * This file was generated by the MCUXpresso Config Tools
 * will be overwritten if the respective MCUXpresso Config
 ****************************************************************************
 */
/* How to setup clock using clock driver functions:
 *
 * 1. CLOCK_SetSimSafeDivs, to make sure core clock, bus
 *    and flash clock are in allowed range during clock n
 *
 * 2. Call CLOCK_Osc0Init to setup OSC clock, if it is us
 *
 * 3. Set MCG configuration, MCG includes three parts: F
 *    internal reference clock(MCGIRCLK). Follow the step
 *
 *    1). Call CLOCK_BootToXxxMode to set MCG to target m
 *
 *    2). If target mode is FBI/BLPI/PBI mode, the MCGIRC
 *       correctly. For other modes, need to call CLOCK_
 *       explicitly to setup MCGIRCLK.
 *
 *    3). Don't need to configure FLL explicitly, because
 *       mode, then FLL has been configured by the funct
 *       if the target mode is not FLL mode, the FLL is
 *
 *    4). If target mode is PEE/PBE/PEI/PBI mode, then th
 *       setup by CLOCK_BootToXxxMode. In FBE/FBI/FEE/FE
 *       be enabled independently, call CLOCK_EnablePlle
 *
 * 4. Call CLOCK_SetSimConfig to set the clock configurat
 */

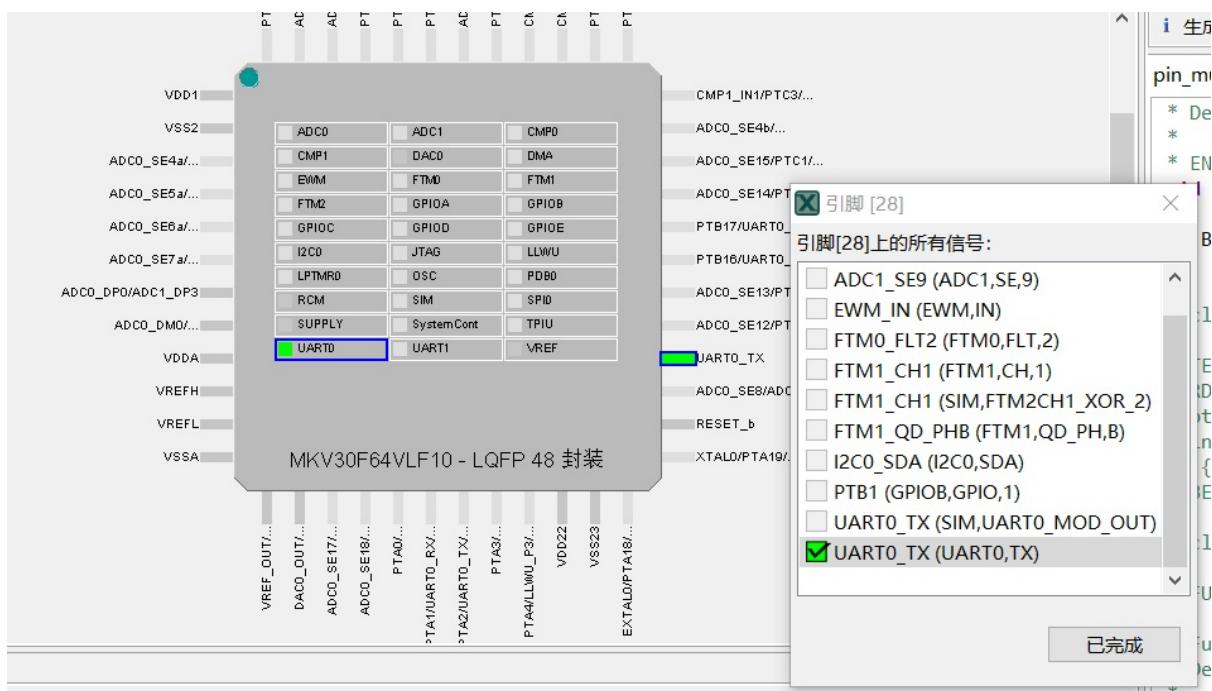
```

## 生成串口输出引脚文件

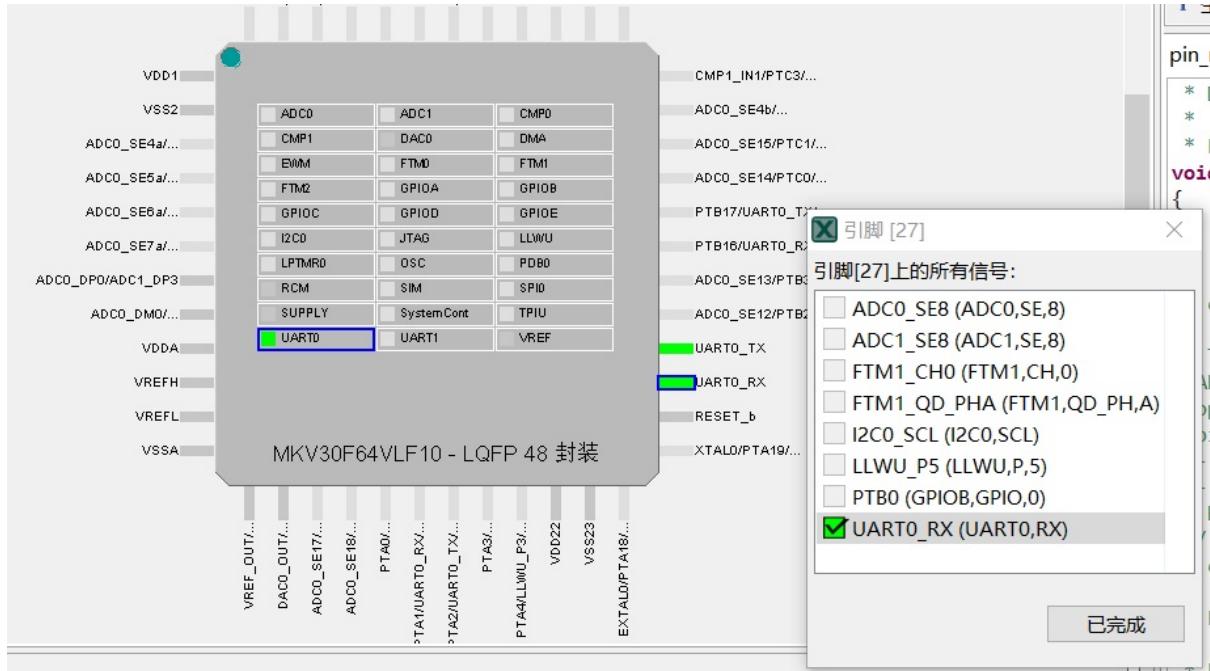
Step 1. 点击工具栏：工具-> 引脚， 默认所有引脚都不配置



Step 2. 这里要配置 TX 串口引脚，该芯片的扩展串口 TX 为 PTB1，点击该引脚然后勾选 UART0\_TX，点击已完成



Step 3. 这里要配置 RX 串口引脚，该芯片的扩展串口 RX 为 PTB0，点击该引脚然后勾选 UART0\_RX，点击已完成



Step 4.点击左方工具栏的源代码，可查看生成的引脚配置文件，点击右方的导出按钮，导出到指定文件夹

寄存器 源代码

生成的代码预览

```
pin_mux.c pin_mux.h
```

```
* Description : Calls initialization functions.
*
* END ****
void BOARD_InitBootPins(void)
{
    BOARD_InitPins();
}

/* clang-format off */
/*
 * TEXT BELOW IS USED AS SETTING FOR TOOLS ****
BOARD_InitPins:
- options: {callFromInitBoot: 'true', coreID: core0, enableClock: 'true'}
- pin_list:
    - {pin_num: '28', peripheral: UART0, signal: TX, pin_signal: ADC0_SE9/ADC1_SE9/PTB1/I2C0_SCL}
    - {pin_num: '27', peripheral: UART0, signal: RX, pin_signal: ADC0_SE8/ADC1_SE8/PTB0/LLWU_P5}
    * BE CAREFUL MODIFYING THIS COMMENT - IT IS YAML SETTINGS FOR TOOLS ****
*/
/* clang-format on */

/* FUNCTION ****
*
* Function Name : BOARD_InitPins
* Description   : Configures pin routing and optionally pin electrical features.
*
* END ****
void BOARD_InitPins(void)
{
    /* Port B Clock Gate Control: Clock enabled */
    CLOCK_EnableClock(kCLOCK_PortB);
}
```





产业首选·通路标竿

### 3.3.4.HelloWorld 示例

Step 1. 将工程结构树 Board 中的 pin\_mux.c、pin\_mux.h、clock\_config.c、clock\_config.h 文件替换成 MCUXpresso Config Tools 生成的对应文件

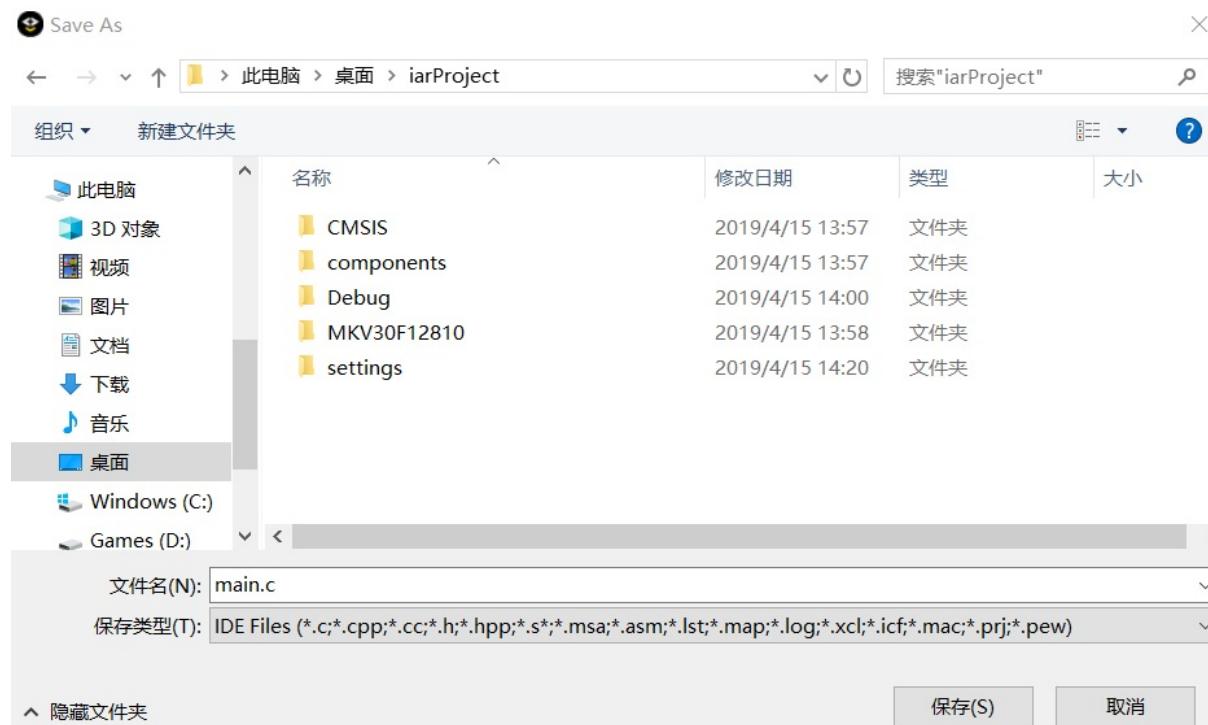
Step 2. 修改 board.c 文件，文件代码如下

```
#include <stdint.h>
#include "board.h"
#include "fsl_common.h"
#include "fsl_debug_console.h"

#define BOARD_DEBUG_UART_CLK_FREQ CLOCK_GetCoreSysClkFreq()
#define BOARD_DEBUG_UART_INSTANCE 0U
#ifndef BOARD_DEBUG_UART_BAUDRATE
#define BOARD_DEBUG_UART_BAUDRATE 115200
#endif
#define BOARD_DEBUG_UART_TYPE kSerialPort_Uart

void BOARD_InitDebugConsole(void) {
    uint32_t uartClkSrcFreq = BOARD_DEBUG_UART_CLK_FREQ;
    DbgConsole_Init(BOARD_DEBUG_UART_INSTANCE, BOARD_DEBUG_UART_BAUDRATE, BOARD_DEBUG_UART_TYPE, uartClkSrcFreq);
}
```

Step 3. 点击工具栏新建文件，生成一个空白 Text 文件，然后点击保存，文件名字修改成 main.c，保存到工程的根目录



Step 4. 右键点击工程结构树的 Source 文件夹，添加 main.c 文件



### Step 5.修改 main.c 文件如下

```
#include "board.h"
#include "fsl_debug_console.h"
#include "clock_config.h"
#include "pin_mux.h"

int main(void)
{
    BOARD_InitPins();
    BOARD_BootClockRUN();
    BOARD_InitDebugConsole();
    PRINTF("hello world!\r\n");
    while(1)
    {
    }
}
```

### Step 6.编译，烧录代码





产业首选·通路标竿

## 4. 基于 NXP 的 MCUXpresso 开发环境搭建

### 概述

MCUXpresso IDE 为开发人员带来了一个易于使用的基于 eclipse 的开发环境，适用于基于 Arm® Cortex®-M 内核的恩智浦 MCU，包括 LPC 和 Kinetis® 微控制器和 i.MX RT 跨界处理器。MCUXpresso IDE 提供高级编辑、编译和调试功能，增加了 MCU 特定的调试视图、代码跟踪和分析、多核调试和集成配置工具。MCUXpresso IDE 调试连接采用来自恩智浦、P&E Micro® 和 SEGGER® 的业界领先的开源和商用硬件调试器，支持 Freedom、塔式系统、i.MX RT、LPCXpresso 和定制开发板。

### 特性

1. 为 Kinetis、LPC MCU 和 i.MX RT 跨界处理器提供代码大小无限制、易于使用的 IDE 免费版。
2. 使用 MCU 专用的调试视图、代码跟踪和分析进行高级编辑、编译和编辑
3. 集成配置工具，包括引脚、时钟和外设工具
4. 业界标准的 GNU 工具链来选择库，如优化的 C 库或标准的 GNU Newlib/Nano 库，可支持半托管
5. 成功创建文件和链接器文件
6. 按照 LPCXpresso IDE v8.22，内置支持 LPC Cortex-M MCU
7. 通过与 MCUXpresso SDK 数据包整合来支持其他 LPC 和 Kinetis MCU
8. 支持 CMSIS-DAP 硬件调试器
  - 包括对 MTB 和 ETB 指令跟踪的支持
  - 包括支持 SWO 跟踪和分析(仅通过 LPC-Link2)
9. 支持 P&E 和 Segger 硬件调试器
  - 包括对 MTB 和 ETB 指令跟踪的支持
  - 包括支持 SWO 跟踪和分析
10. GUI 闪存编程工具、外设寄存器视图、全局变量视图、文本/图形实时变量视图、项目向导等等
11. 支持 FreeRTOS 感知调试
12. 可以通过许多 Eclipse 插件实现扩展
13. 主机操作系统：
  - Microsoft® Windows® 7/8/10
  - Ubuntu Linux® 16.04 LTS 及更高版本(仅 64 位主机操作系统)
  - Mac OS X 10.11 及更高版本



产业首选 · 通路标竿

## 4.1. 下载并安装 MCUXpresso IDE

[MCUXpressoIDE 下载页面](#)

需注册 NXP 官网账号

Step 1. 进入下载页面，点击 Download

### MCUXpresso Integrated Development Environment (IDE)



#### Jump To

- [Overview & Features](#)
- [Supported Devices](#)
- [Target Applications](#)
- [System Requirements](#)

#### Overview

The MCUXpresso IDE brings developers an easy-to-use Eclipse-based development environment for NXP® MCUs based on Arm® Cortex®-M cores, including LPC and Kinetis® microcontrollers and i.MX RT crossover processors. The MCUXpresso IDE offers advanced editing, compiling and debugging features with the addition of MCU-specific debugging views, code trace and profiling, multicore debugging, and integrated configuration tools. The MCUXpresso IDE debug connections support Freedom, Tower® system, LPCXpresso, i.MX RT, and your custom development boards with industry-leading open-source and commercial debug probes from NXP, P&E Micro®, and

#### Features

- A free-of-c for Kinetis processors
- Advanced specific de
- Integrated and periph
- Industry-st libraries: o| Newlib/Nai
- Managed r

[More ▾](#)

User Guide

Download

Step 2. 点击 MCUXpresso IDE

## Product Information

### MCUXpresso IDE

To register a New Product please click on the button below

[Register](#)

Current	Previous
Version	Description
10.3.1	MCUXpresso IDE
<a href="#">Download Log</a>	

Step 3. 选择 MCUXpresso v10.3.1 - Windows, 等待下载完成

## Product Download

### MCUXpresso IDE

[Files](#) [License Keys](#) [Notes](#) [Download Help](#)

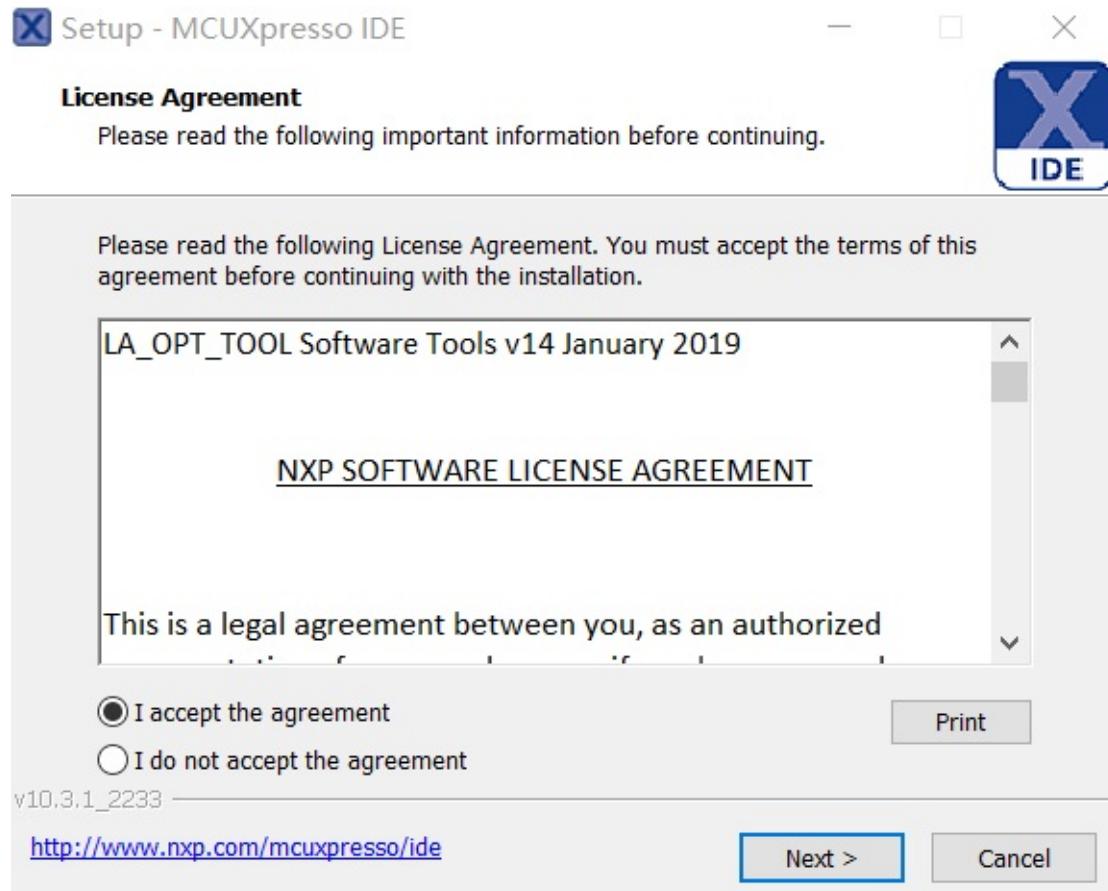
Show All Files

4 Files

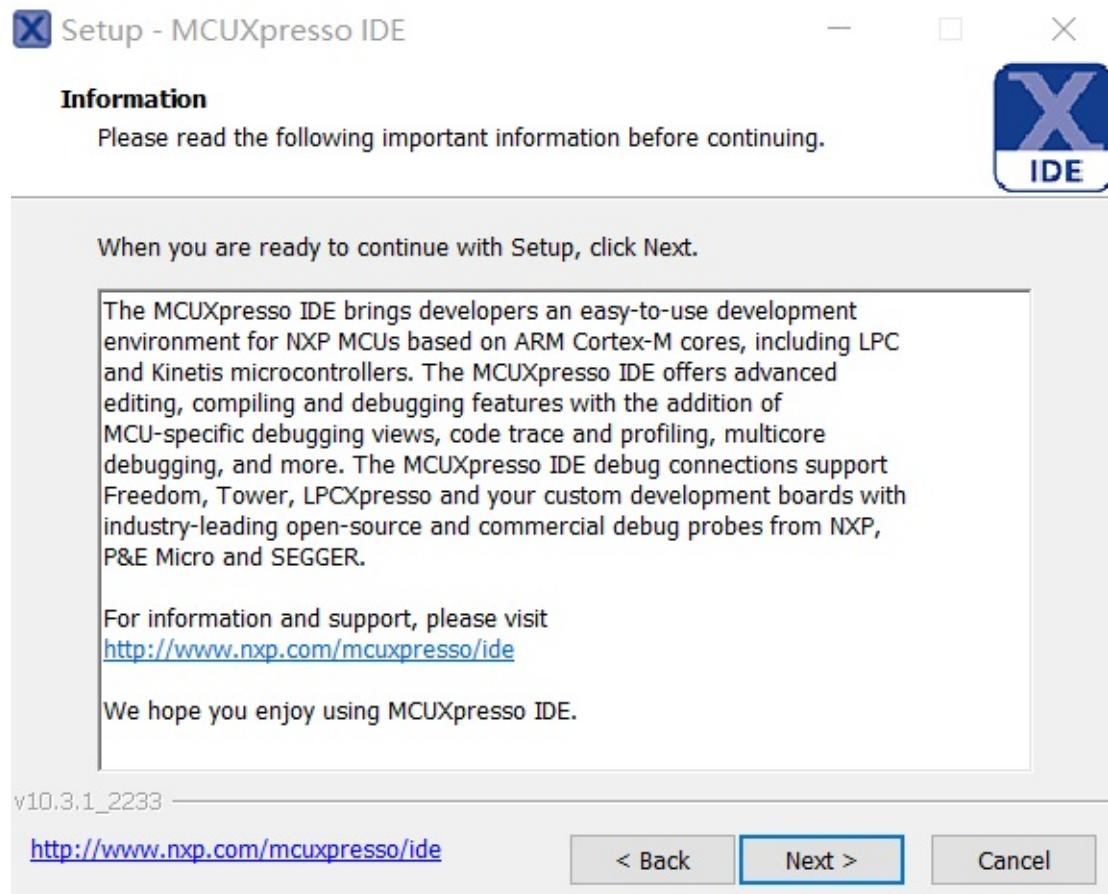
<input type="checkbox"/>	<input type="checkbox"/>	File Description	File Size	File Name
<input type="checkbox"/>	<input type="checkbox"/>	MCUXpresso v10.3.1 - Linux	753.4 MB	<a href="#">mcuxpressoide-10.3.1_2233.x86_64.deb.bin</a>
<input type="checkbox"/>	<input type="checkbox"/>	MCUXpresso v10.3.1 - Mac	720.8 MB	<a href="#">MCUXpressoIDE 10.3.1_2233.pkg</a>
<input type="checkbox"/>	<input type="checkbox"/>	MCUXpresso v10.3.1 - Windows	675.8 MB	<a href="#">MCUXpressoIDE 10.3.1_2233.exe</a>
<input type="checkbox"/>	<input type="checkbox"/>	Readme MCUXpresso 10.3.1	53.7 KB	<a href="#">Readme MCUXpresso 10.3.1.pdf</a>

[Download Selected Files](#)

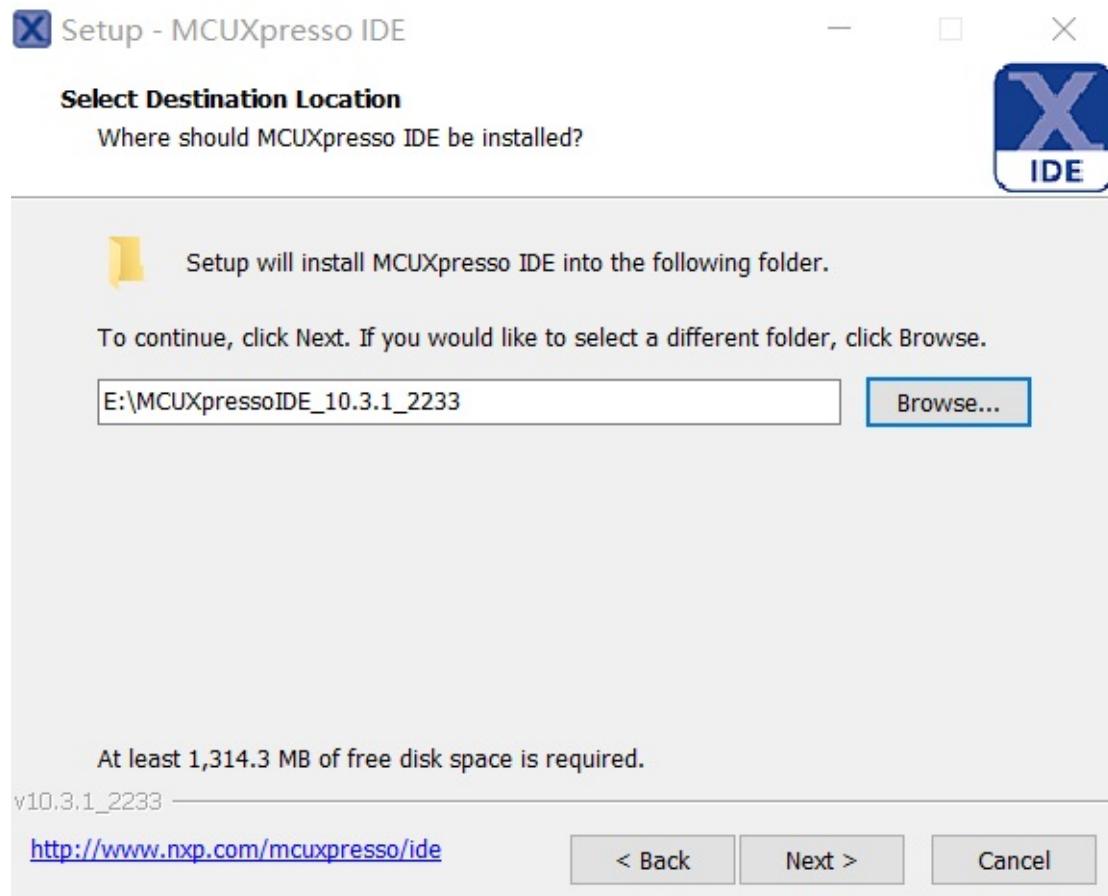
Step 4. 双击安装包，选择 I accept the agreement，点击 Next



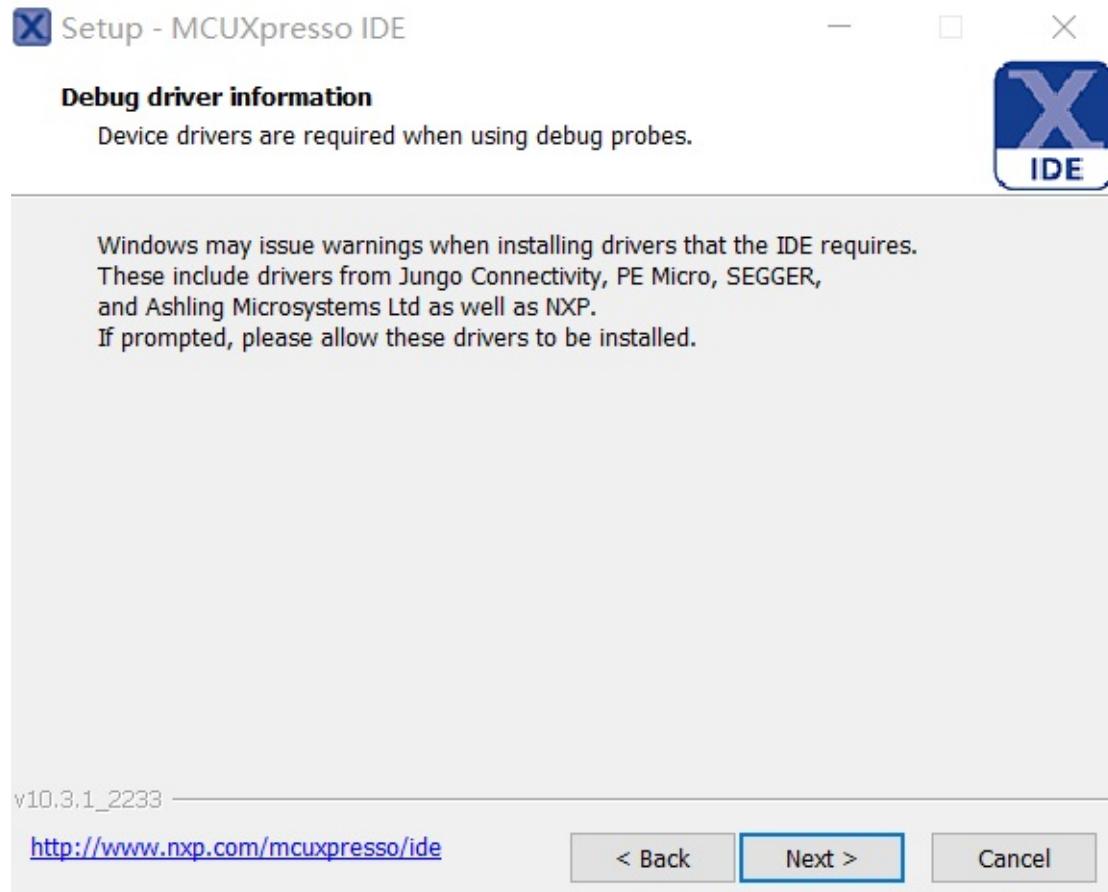
Step 5. 点击 Next



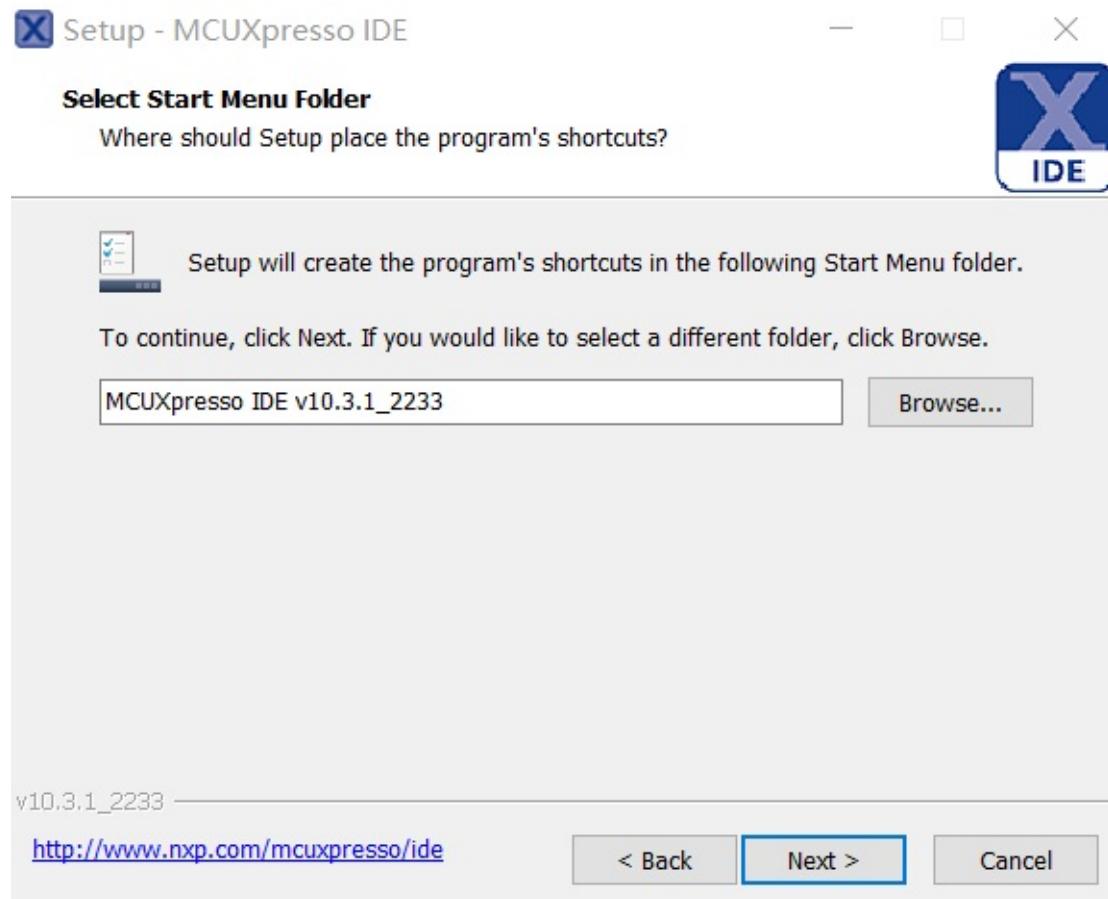
Step 6. 选择安装路径，点击 Next



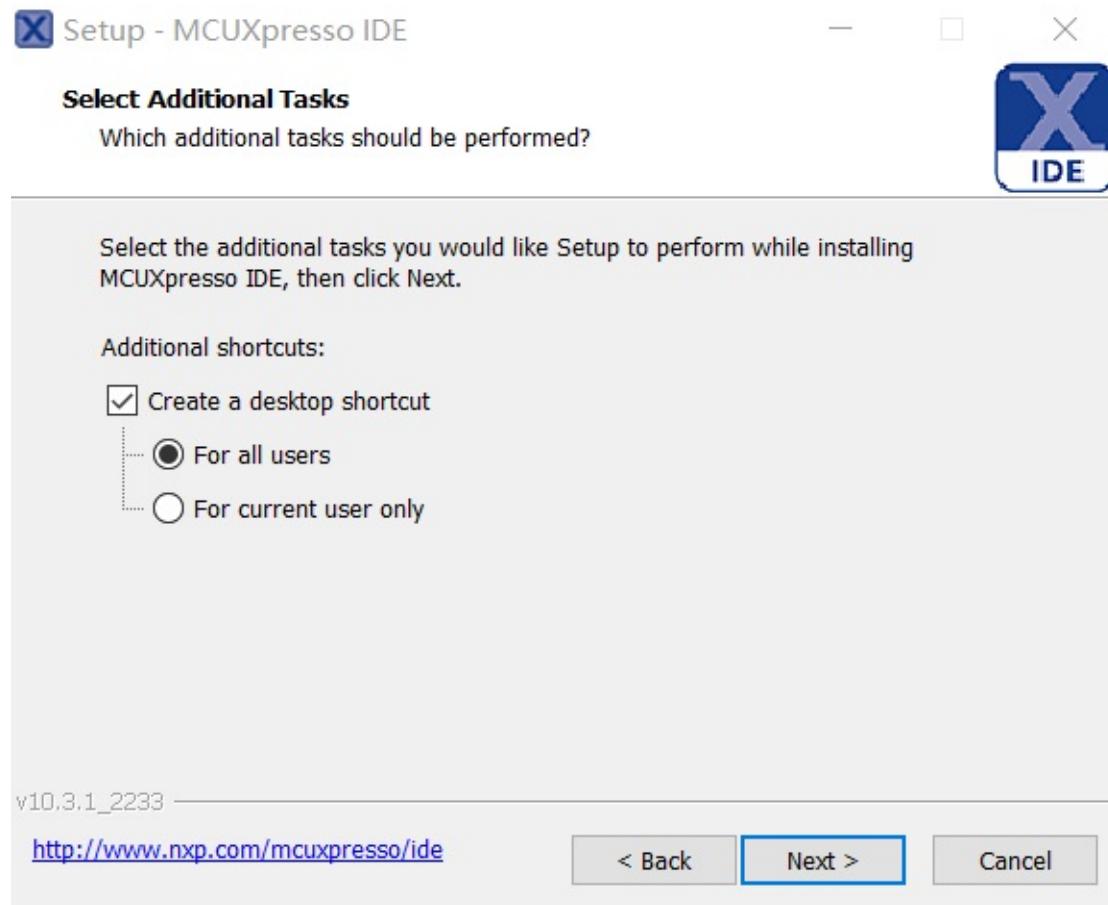
Step 7. 点击 Next



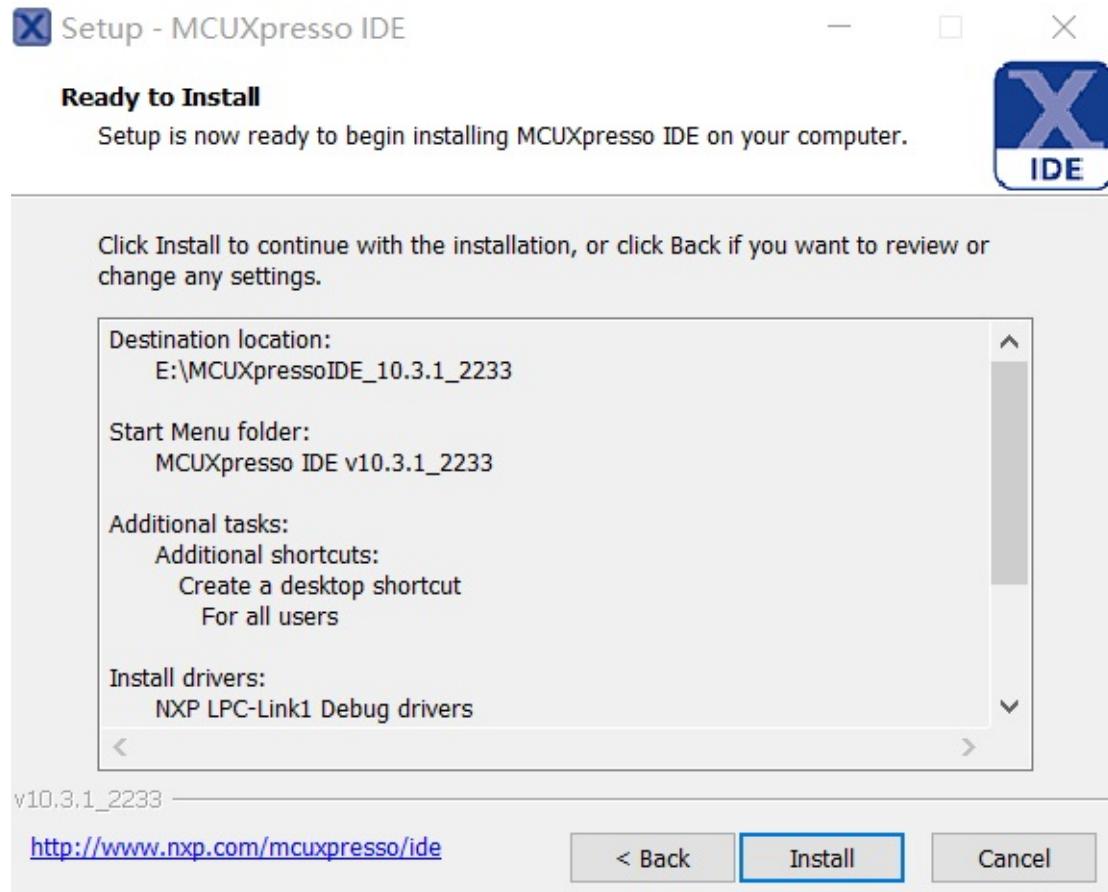
Step 8. 选择创建快捷键名称，点击 Next



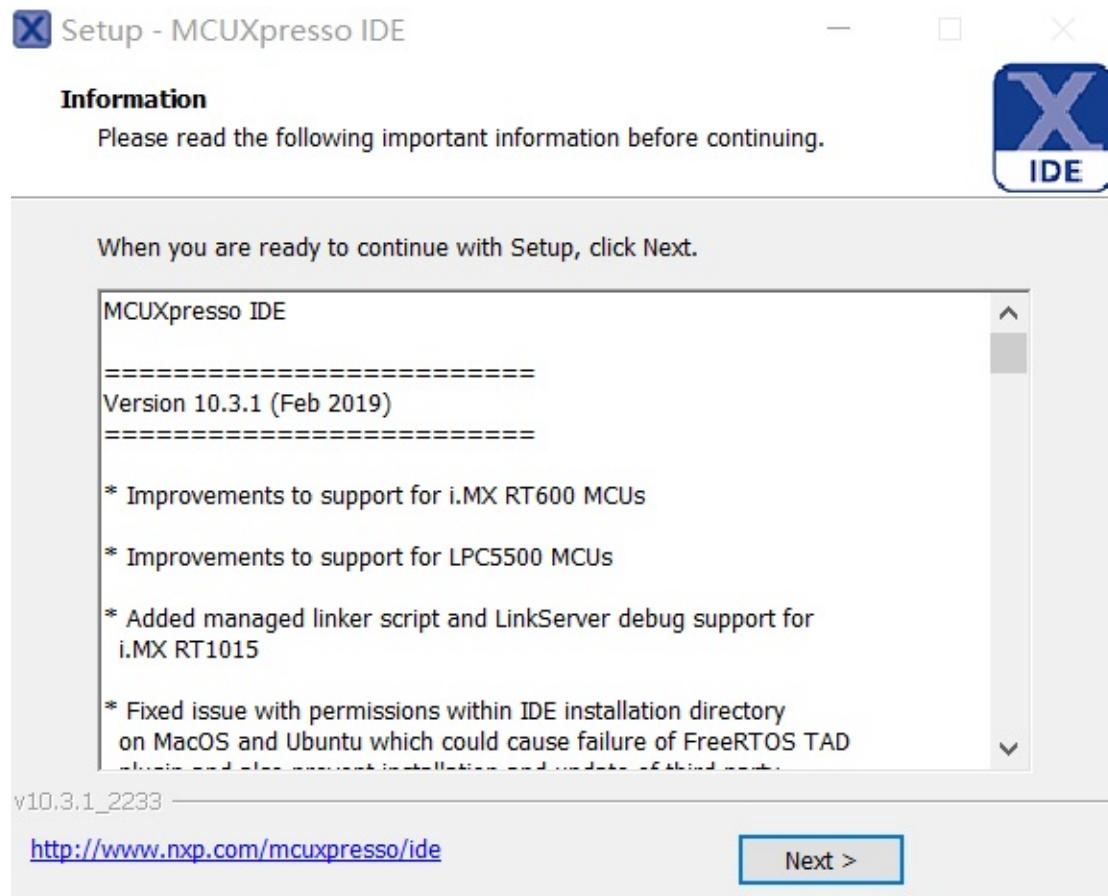
Step 9. 允许任何人打开，点击 Next



Step 10.点击 Install



Step 11.点击 Next



Step 12.点击 Finish





产业首选 · 通路标竿

## 4.2. 下载并安装 MCUXpresso Config Tools

[MCUXpressoConfigTools 下载页面](#)

需注册 NXP 官网账号

Step 1. 进入下载页面，点击下载

### Recommended Software & Tools (4)



#### MCUXpresso Config Tools, Windows 64bit package (REV 4)

下载

MCUXpresso Config Tools v4 for Windows 64-bit. Supports MCUXpresso SDK. Includes Pins and Clocks configuration, project cloning, and project generation.

Archived

EXE 139062 KB MCUXPRESSO-CT-WIN64-PACKAGE

2017-11-14 15:27:00

Step 2. 点击 I Accept

Agreement : MCUXpresso Config Tools, Windows 64bit package

LA\_OPT\_TOOL Software Tools v14 January 2019

**IMPORTANT.** Read the following NXP Software License Agreement ("Agreement") completely. By selecting the "I Accept" button at the end of this page, you indicate that you accept the terms of the Agreement and you acknowledge that you have the authority, for yourself or on behalf of your company, to bind your company to these terms. You may then download or install the file.

#### NXP SOFTWARE LICENSE AGREEMENT

This is a legal agreement between you, as an authorized representative of your employer, or if you have no employer, as an individual (together "you"), and NXP B.V. ("NXP"). It concerns your rights to use the software identified in the Software Content Register and provided to you in binary or source code form and any accompanying written materials (the "Licensed Software"). The Licensed Software may include any updates or error corrections or documentation relating to the Licensed Software provided to you by NXP under this License. In consideration for NXP allowing you to access the Licensed Software, you are agreeing to be bound by the terms of this Agreement. If you do not agree to all of the terms of this Agreement, do not download or install the Licensed Software. If you change your mind later, stop using the Licensed Software and delete all copies of the Licensed Software in your possession or control. Any copies of the Licensed Software that you have already distributed, where permitted, and do not destroy will continue to be governed by this Agreement. Your prior use will also continue to be governed by this Agreement.

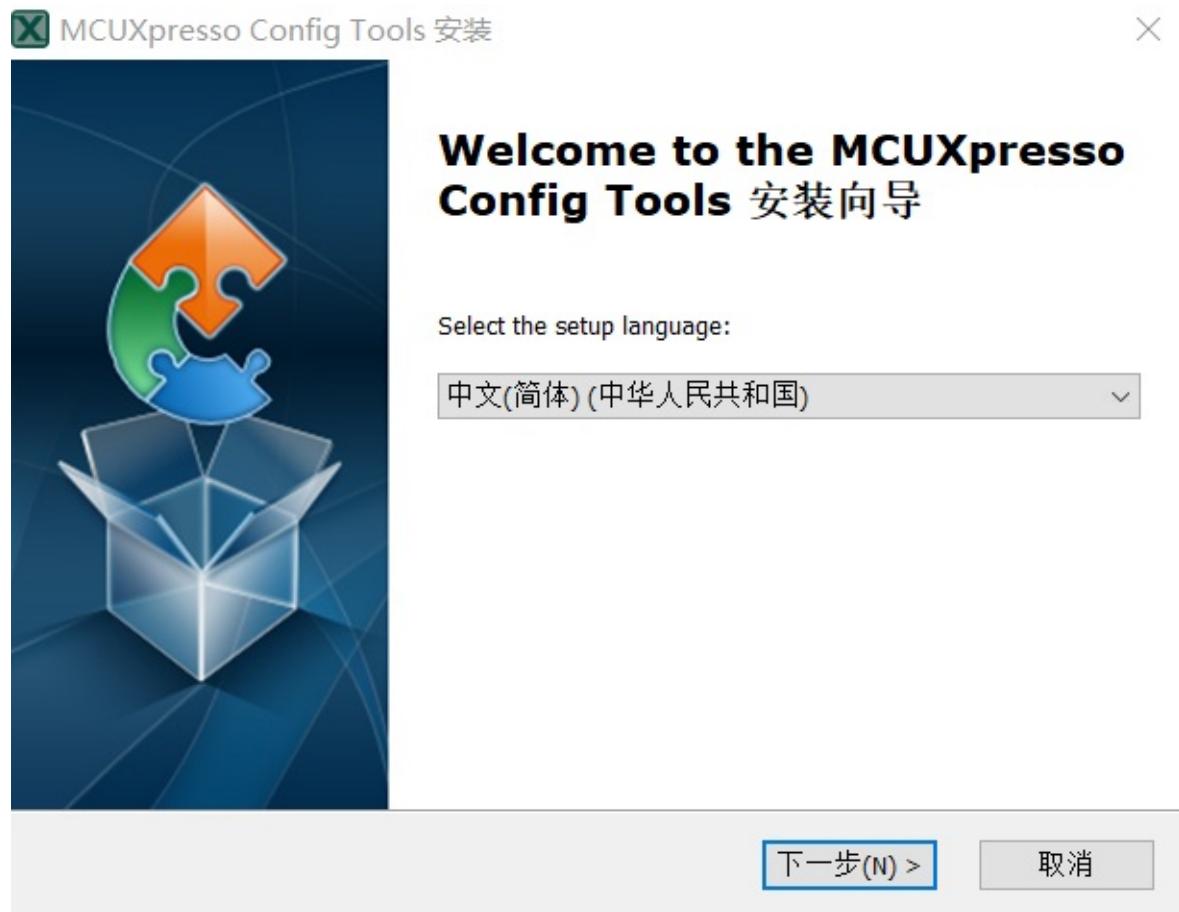
##### 1. DEFINITIONS

1.1 For NXP, the term "Affiliate" means (i) any Person Controlled by NXP Semiconductors N.V. or (ii) any Person Controlled by any transferee of all or substantially all of the assets of NXP Semiconductors N.V.

I Accept

Decline

Step 3. 选择语言，点击下一步



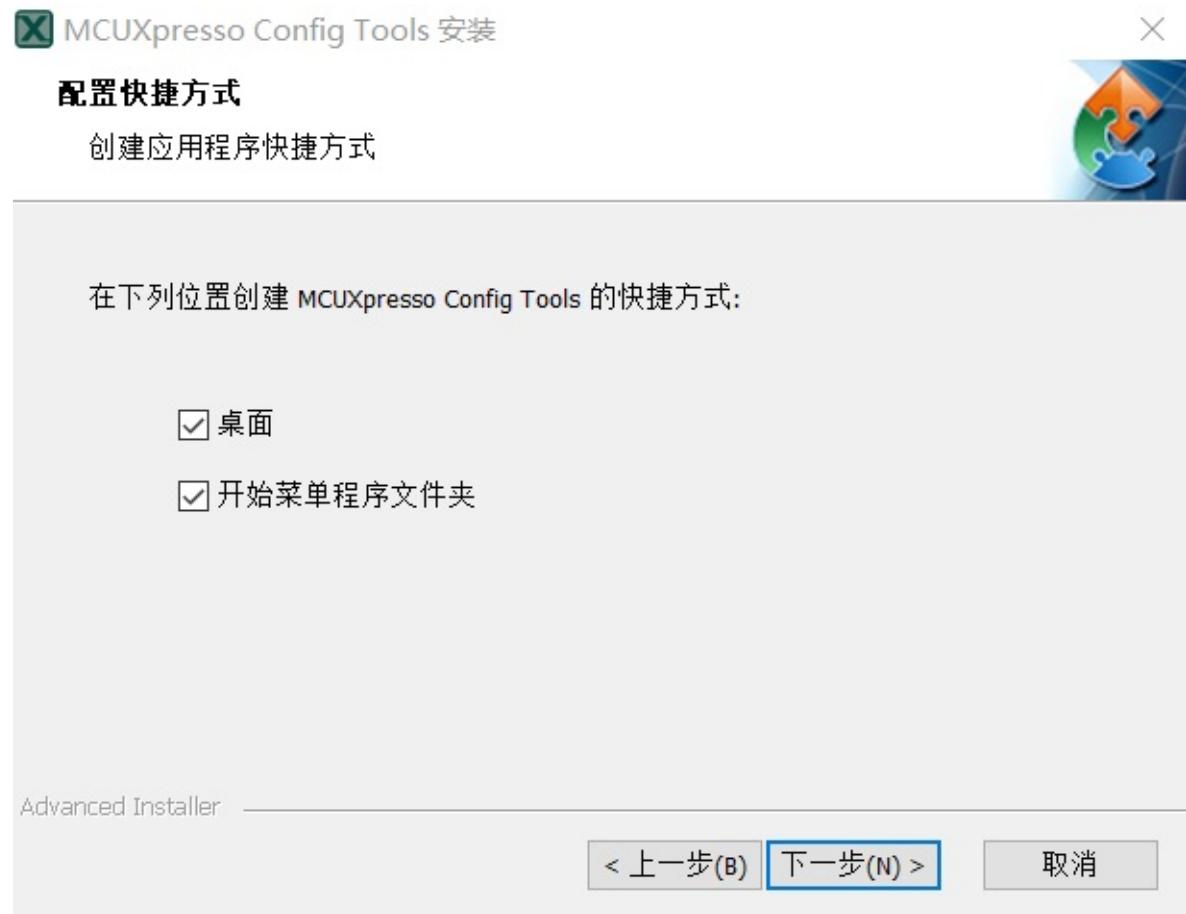
Step 4.点击下一步



Step 5. 选择安装路径，点击下一步



Step 6. 创建快捷方式，点击下一步



Step 7.点击安装



Step 8.点击完成





产业首选 · 通路标竿

## 4.3. 下载 MCUXpresso SDK

[SDK 下载地址](#)

需注册 NXP 官网账号

Step 1. 进入下载地址, 搜索你要下载芯片的 SDK, 这里以 MKV30F64xxx10 为例, 选择 MKV30F64xxx10, 右方会显示该芯片的信息, 然后点击 Build MCUXpresso SDK

**Search by Name**

(x)

**Select a Device, Board, or Kit**

- ▼ Boards
- ▼ Kits
- ▼ Processors
- MKV30F64xxx10

**Name your SDK**


Don't use: <, >, :, ", /, |, ?, \*, \ in the name of your SDK

**Hardware Details**

Included Part Numbers	MKV30F64VLF10, MKV30F64VFM10, MKV30F64VLH10
Board(s)	TWR-KV31F120M, FRDM-KV31F
Device	MKV30F12810
Core Type / Max Freq	Cortex-M4F / 100MHz
Device Memory Size	64 KB Flash 16 KB RAM

**Actions**

- [Build MCUXpresso SDK](#)
- 🔍 [Explore selection with Clocks tool](#)
- 💡 [Explore selection with Pins tool](#)

Step 2. 选择 Windows 主机系统, 下载的工具链为 MCUXpresso IDE, 组件默认即可, 若有特别的组件需求可自己添加进去, 点击 Download SDK

# SDK Builder

Generate a downloadable SDK archive for use with desktop MCUXpresso

## Developer Environment Settings

*Selections here will impact files and examples projects included in the SDK and Generated Project*

Host OS

Windows

Toolchain / IDE

MCUXpresso IDE

## Select Optional Middleware

Add middleware, operating systems, and software libraries to your SDK.

+ Add software component

This MCUXpresso SDK configuration is available for direct download

Download SDK

Archive Name

SDK\_2.5.0\_MKV30F64xxx10

Don't use: <,>, ;, ", /, \, ?, \*, \ in the name of your  
SDK

Step 3. 点击 I Agree

X

## SDK Downloads

### Software Terms and Conditions

Please read the following agreement and click "I AGREE" at the bottom before downloading your software.

[EULA](#)[Software Content Register](#)

LA\_OPT\_NXP\_Software\_License v4 November 2018

**IMPORTANT.** Read the following NXP Software License Agreement ("Agreement") completely. By selecting the "I Accept" button at the end of this page, or by downloading, installing, or using the Licensed Software, you indicate that you accept the terms of the Agreement and you acknowledge that you have the authority, for yourself or on behalf of your company, to bind your company to these terms. You may then download or install the file. In the event of a conflict between the terms of this Agreement and any license terms and conditions for NXP's proprietary software embedded anywhere in the Licensed Software file, the terms of this Agreement shall control. If a separate license agreement for the Licensed Software has been signed by you and NXP, then that agreement shall govern your use of the Licensed Software and shall supersede this Agreement.

#### NXP SOFTWARE LICENSE AGREEMENT

This is a legal agreement between your employer, of which you are an authorized representative, or, if you have no employer, you as an individual ("you" or "Licensee"), and NXP B.V. ("NXP"). It concerns your rights to use the software provided to you in binary or source code form and any accompanying written materials (the "Licensed Software"). The Licensed Software may include any updates or error corrections or documentation relating to the Licensed Software provided to you by NXP under this Agreement. In consideration for NXP allowing you to access the Licensed Software, you are agreeing to be bound by the terms of this Agreement. If you do not agree to all of the terms of this Agreement, do not download or install the Licensed Software. If you change your mind later, stop using the Licensed Software and delete all copies of the Licensed Software in your possession or control. Any

[I Agree](#)[Back](#)[Close](#)



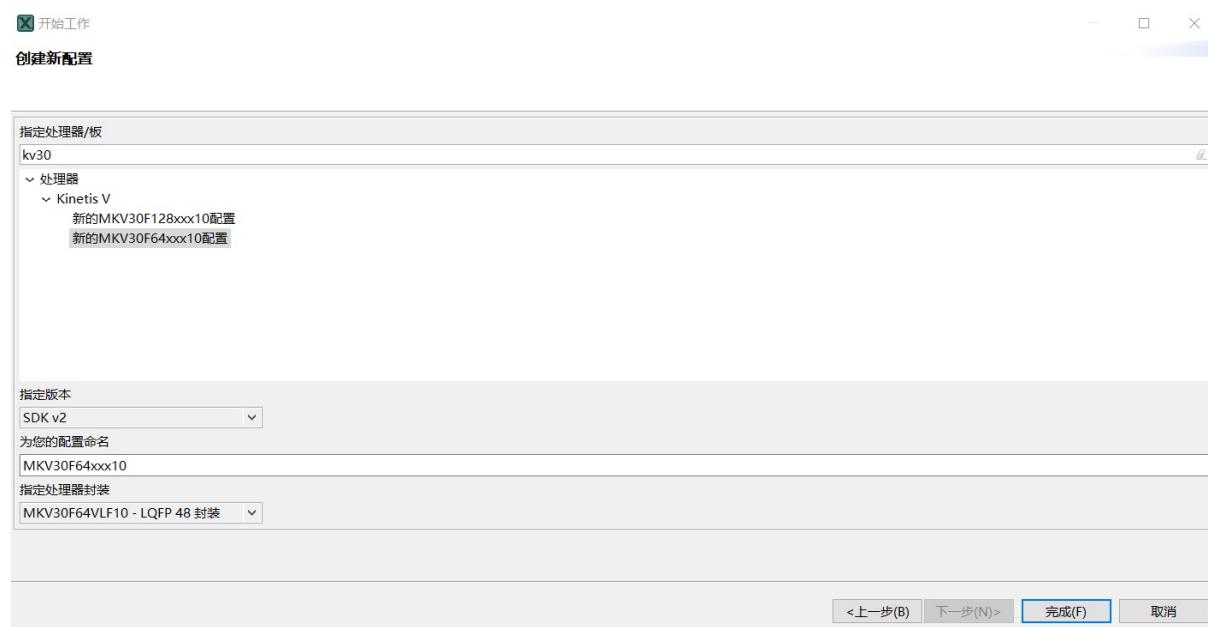
## 4.4. 使用 MCUXpresso Config Tools 生成代码

### 生成配置外部晶振的时钟文件

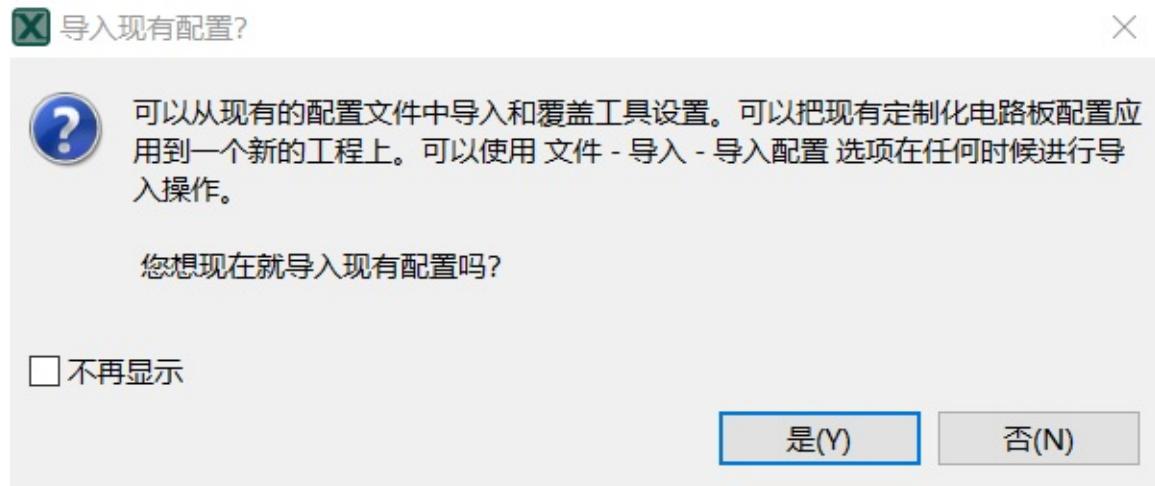
Step 1. 打开 MCUXpresso Config Tools，选择为处理器，电路板，和套件创建新配置，点击下一步



Step 2. 查询你配置的芯片，在这里为 MKV30F64xxx，封装为 MKV30F64VLF10 - LQFP 48 封装



Step 3. 点击否



Step 4.点击工具栏：工具->时钟，该芯片默认配置为内部时钟，点击外部OSC的Inactive框

时钟源	
名称	可 值
内部	
FAST_IRCLK	4 MHz
SLOW_IRCLK	32.768 kHz
IRC48M	Inactive
LPO	1 kHz
外部	
OSC (System Oscillator)	<input type="checkbox"/> Inactive

Step 5.输入芯片的外部晶振参数，在这里使用的是无源的8MHz晶振，点击OK



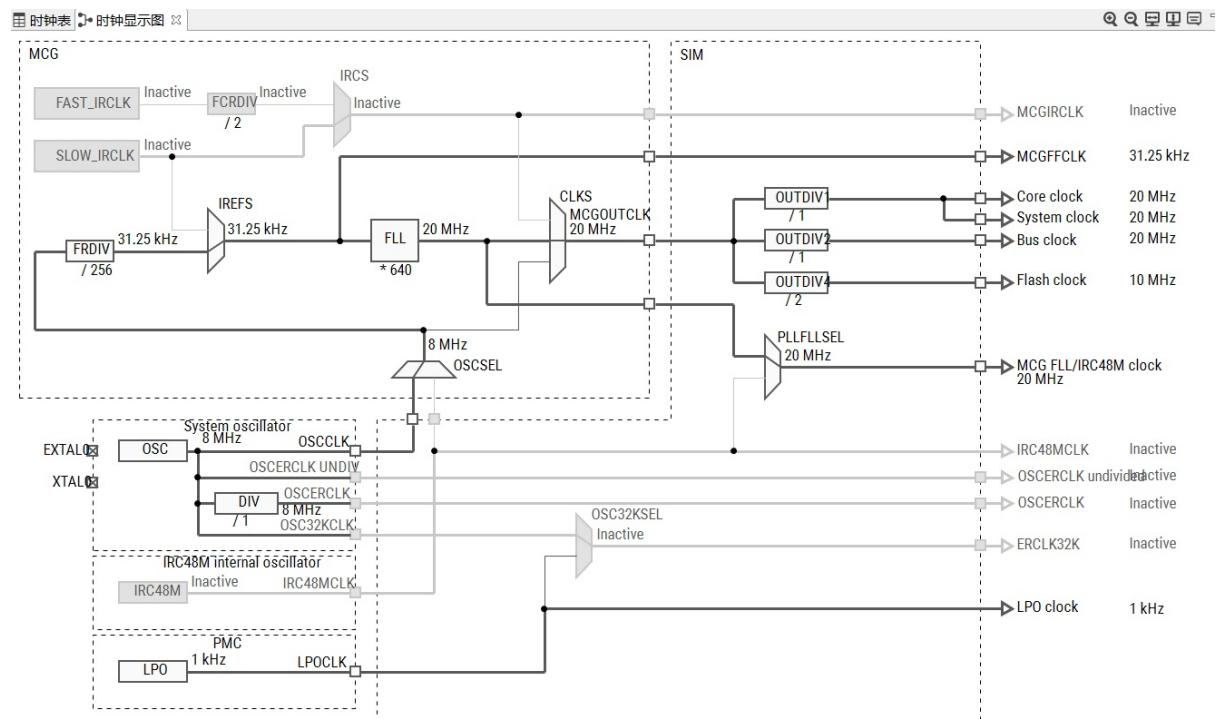
Step 6.将外部OSC下拉框展开，OSC mode选择Using oscillator with external crystal(low power)，Frequency Range选择High frequency range 3-8 MHz，System Osc.Capacity Load选择0pF(芯片内部晶振的负载电容)

外部	
OSC (System Oscillator)	<input checked="" type="checkbox"/> 8 MHz
OSC mode	Using oscillator...tal (low power)
Frequency Range	High frequency range 3-8 MHz
System O...ty Load	0 pF

Step 7.工具栏的MCG Mode从FEI模式切换成FEE模式

## MCG Mode FEE (FLL Engaged External)

Step 8. 观察时钟显示图，时钟已配置成外部晶振输入



Step 9. 右方工具栏点击源代码，可查看自动生成的时钟代码文件，右方有个导出按钮，将时钟的 .c 和 .h 文件导出到指定目录

The screenshot shows the MCUXpresso Config Tools interface with the 'Generated Code Preview' tab selected. The code editor displays the contents of `clock_config.c` and `clock_config.h`. The code is a generated configuration file for the clock system, providing instructions on how to set up the clock using driver functions. It includes comments explaining the steps for setting up the core, bus, and flash clocks, as well as the MCG configuration (including FLL and MCGIRCLK setup) and Pll configuration.

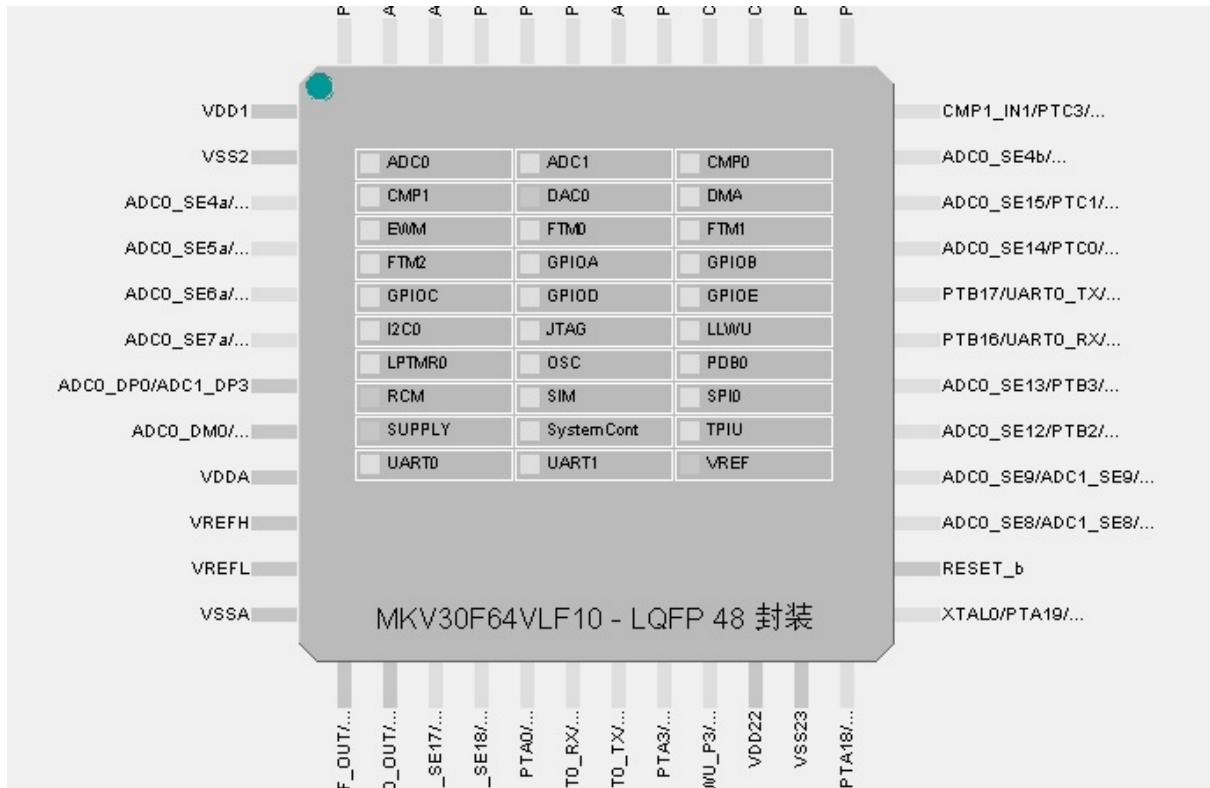
```

/*
 * This file was generated by the MCUXpresso Config Tools
 * will be overwritten if the respective MCUXpresso Config
 */
/*
 * How to setup clock using clock driver functions:
 *
 * 1. CLOCK_SetSimSafeDivs, to make sure core clock, bus
 *    and flash clock are in allowed range during clock n
 *
 * 2. Call CLOCK_Osc0Init to setup OSC clock, if it is us
 *
 * 3. Set MCG configuration, MCG includes three parts: FL
 *    internal reference clock(MCGIRCLK). Follow the step
 *
 *   1). Call CLOCK_BootToXxxMode to set MCG to target m
 *
 *   2). If target mode is FBI/BLPI/PBI mode, the MCGIRC
 *      correctly. For other modes, need to call CLOCK_
 *      explicitly to setup MCGIRCLK.
 *
 *   3). Don't need to configure FLL explicitly, because
 *      mode, then FLL has been configured by the funct
 *      if the target mode is not FLL mode, the FLL is
 *
 *   4). If target mode is PEE/PBE/PEI/PBI mode, then th
 *      setup by CLOCK_BootToXxxMode. In FBE/FBI/FEE/FE
 *      be enabled independently, call CLOCK_EnablePll()
 *
 * 4. Call CLOCK_SetSimConfig to set the clock configurat
*/

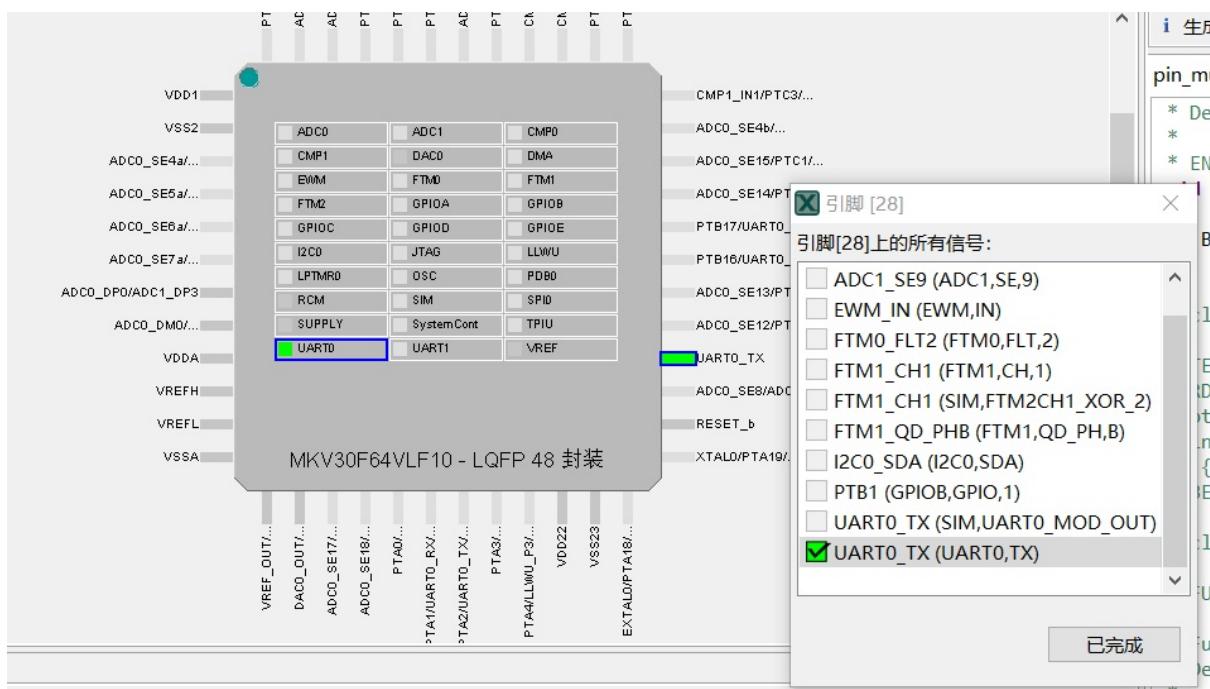
```

## 生成串口输出引脚文件

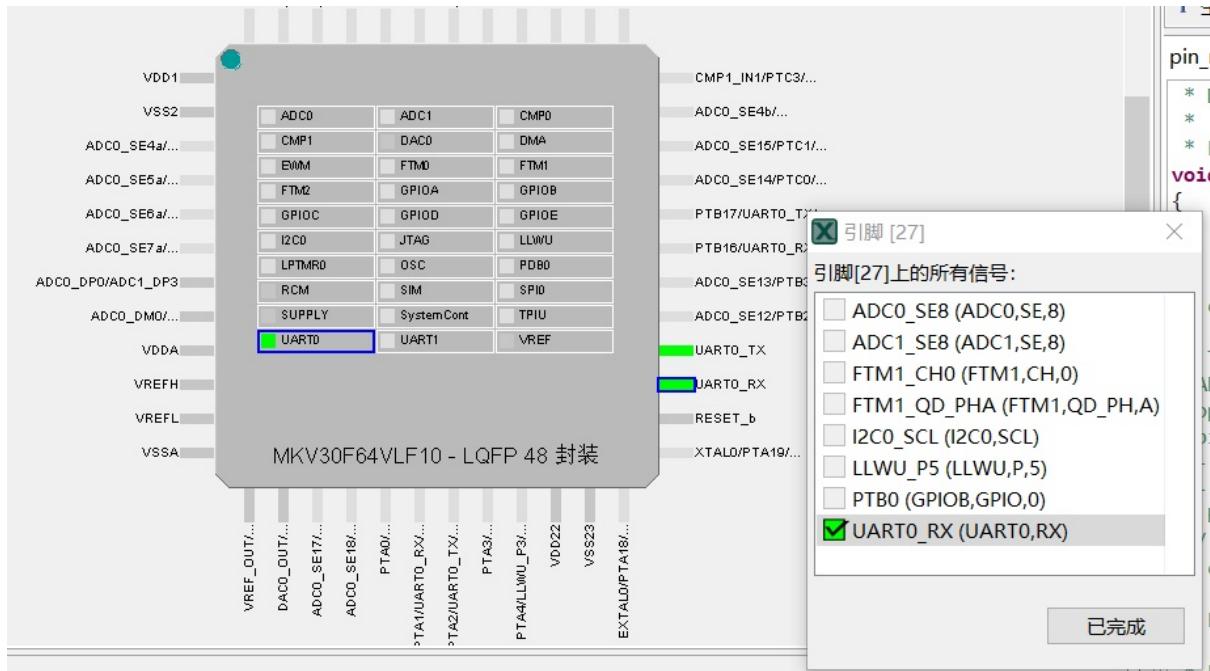
Step 1.点击工具栏：工具->引脚， 默认所有引脚都不配置



Step 2. 这里要配置 TX 串口引脚，该芯片的扩展串口 TX 为 PTB1，点击该引脚然后勾选 UART0\_TX，点击已完成



Step 3. 这里要配置 RX 串口引脚，该芯片的扩展串口 RX 为 PTB0，点击该引脚然后勾选 UART0\_RX，点击已完成



Step 4.点击左方工具栏的源代码，可查看生成的引脚配置文件，点击右方的导出按钮，导出到指定文件夹

**寄存器** **源代码**

**生成的代码预览**

```

pin_mux.c pin_mux.h

/*
 * Description : Calls initialization functions.
 *
 * END ****
void BOARD_InitBootPins(void)
{
    BOARD_InitPins();
}

/* clang-format off */
/*
 * TEXT BELOW IS USED AS SETTING FOR TOOLS ****
BOARD_InitPins:
- options: {callFromInitBoot: 'true', coreID: core0, enableClock: 'true'}
- pin_list:
    - {pin_num: '28', peripheral: UART0, signal: TX, pin_signal: ADC0_SE9/ADC1_SE9/PTB1/I2C0_SCL}
    - {pin_num: '27', peripheral: UART0, signal: RX, pin_signal: ADC0_SE8/ADC1_SE8/PTB0/LLWU_P5}
    * BE CAREFUL MODIFYING THIS COMMENT - IT IS YAML SETTINGS FOR TOOLS ****
*/
/* clang-format on */

/*
 * FUNCTION ****
*
* Function Name : BOARD_InitPins
* Description   : Configures pin routing and optionally pin electrical features.
*
* END ****
void BOARD_InitPins(void)
{
    /* Port B Clock Gate Control: Clock enabled */
    CLOCK_EnableClock(kCLOCK_PortB);
}

```





产业首选·通路标竿

## 4.5. 使用 MCUXpresso IDE 调试代码

---

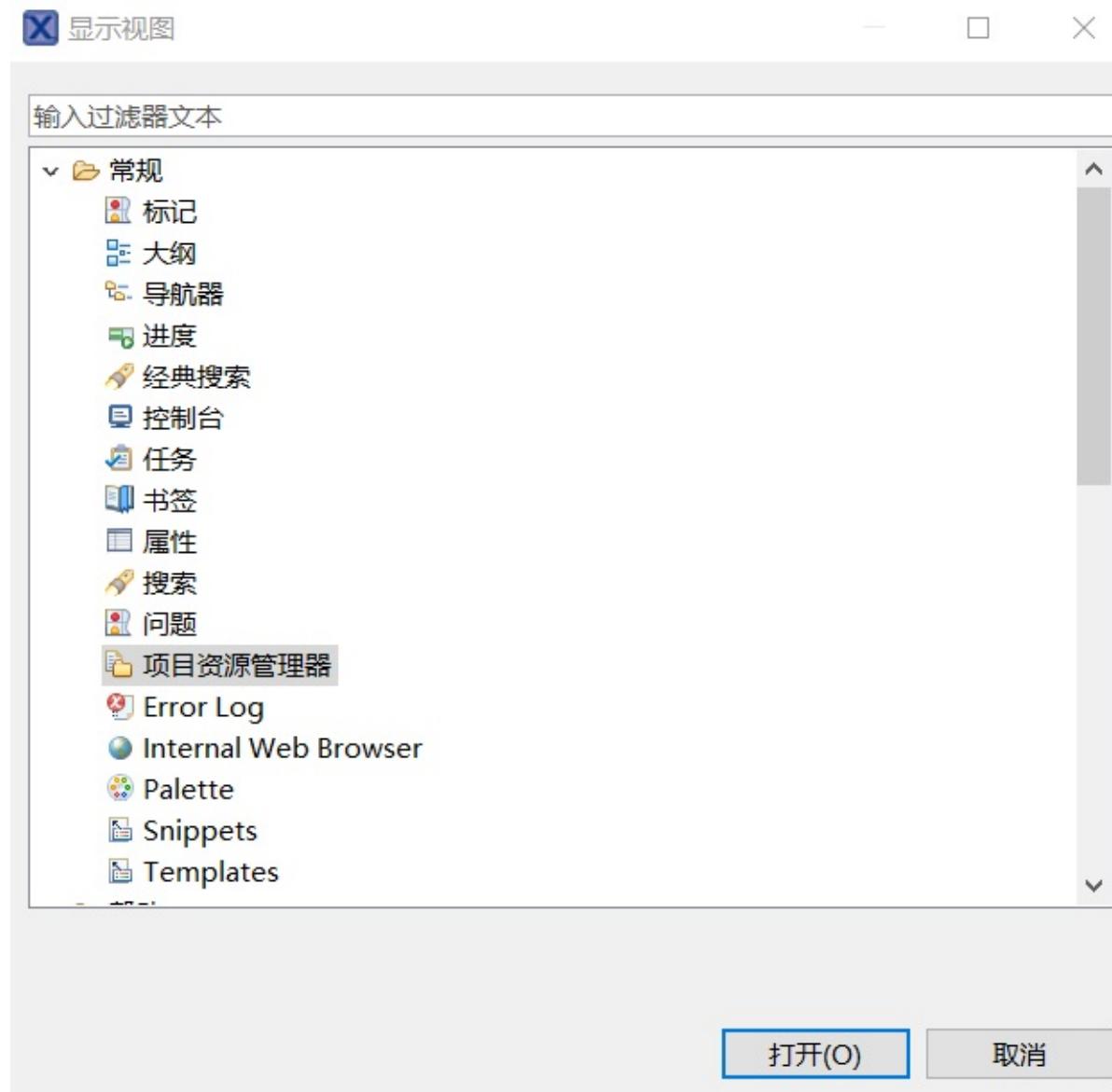


产业首选·通路标竿

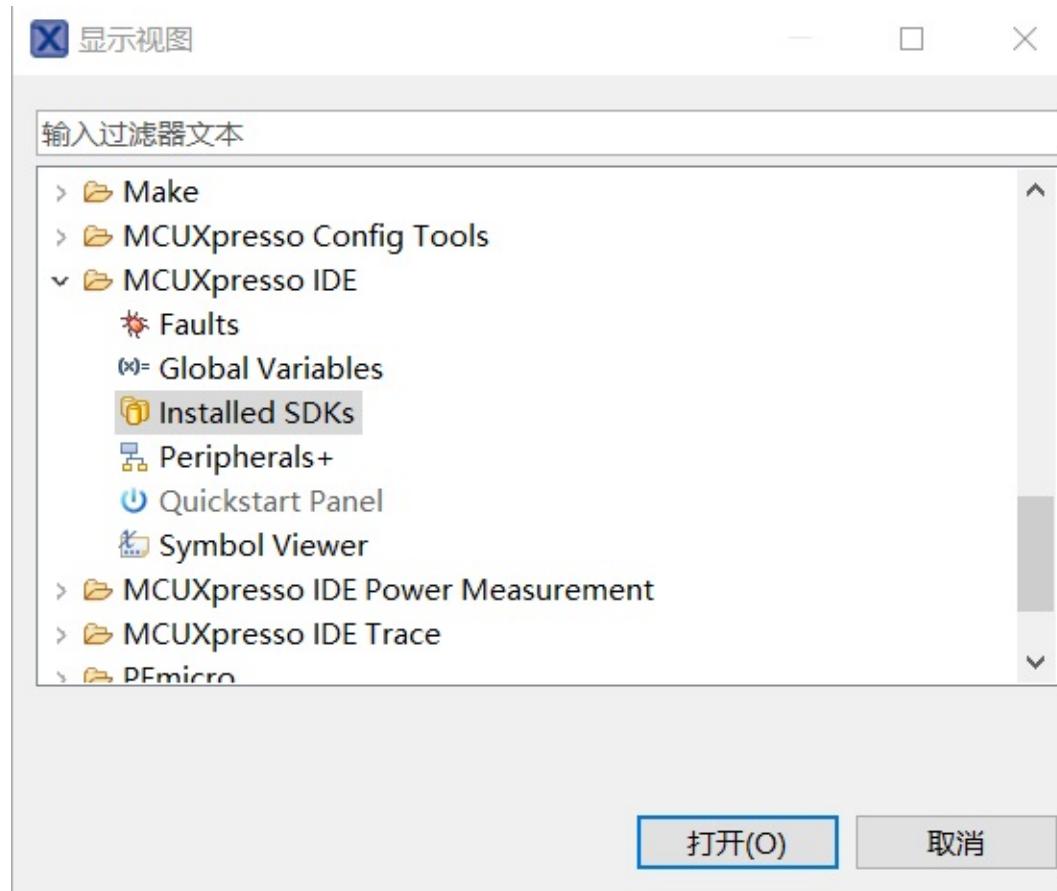
## 4.5.1.开启视图窗口

打开三个必要的视图窗口，选择工具栏：窗口->显示视图->其它

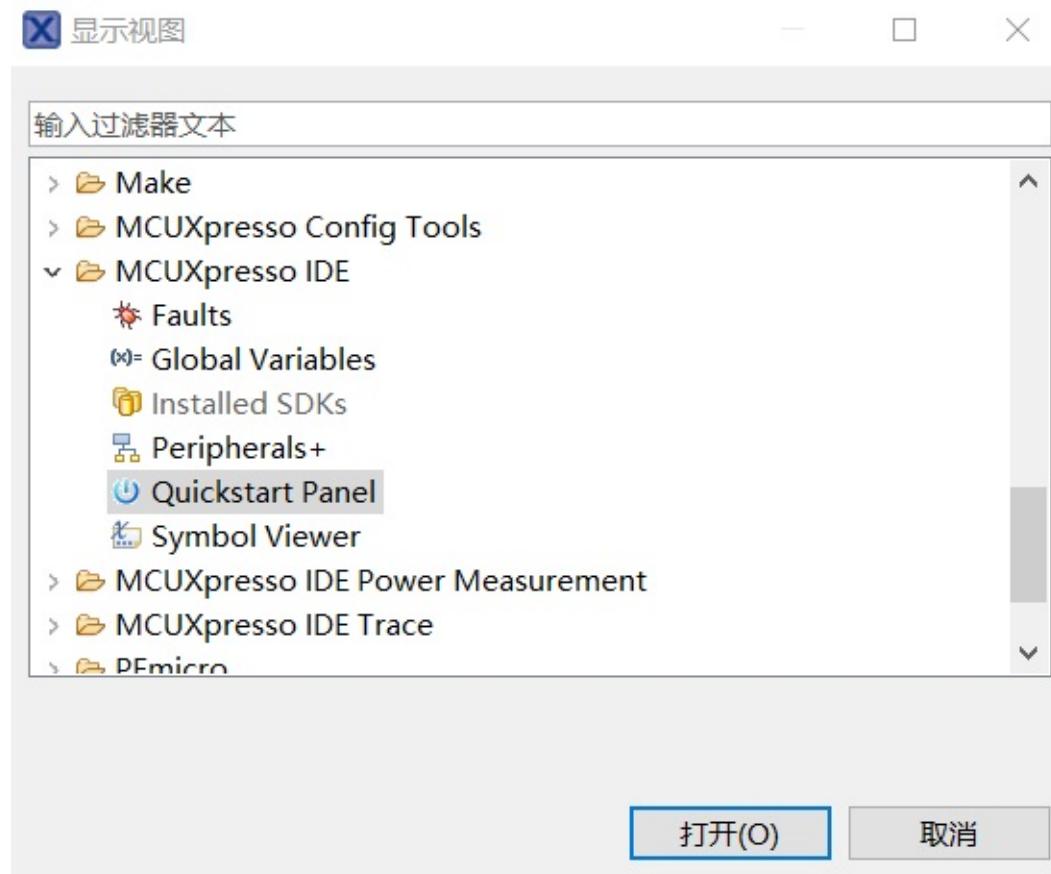
Step 1.选择常规 -> 项目资源管理器，点击打开



Step 2.选择 MCUXpresso IDE -> Installed SDKs, 点击打开



Step 3.选择 MCUXpresso IDE -> Quickstart Panel, 点击打开







产业首选 · 通路标竿

## 4.5.2.加载 Xpresso SDK

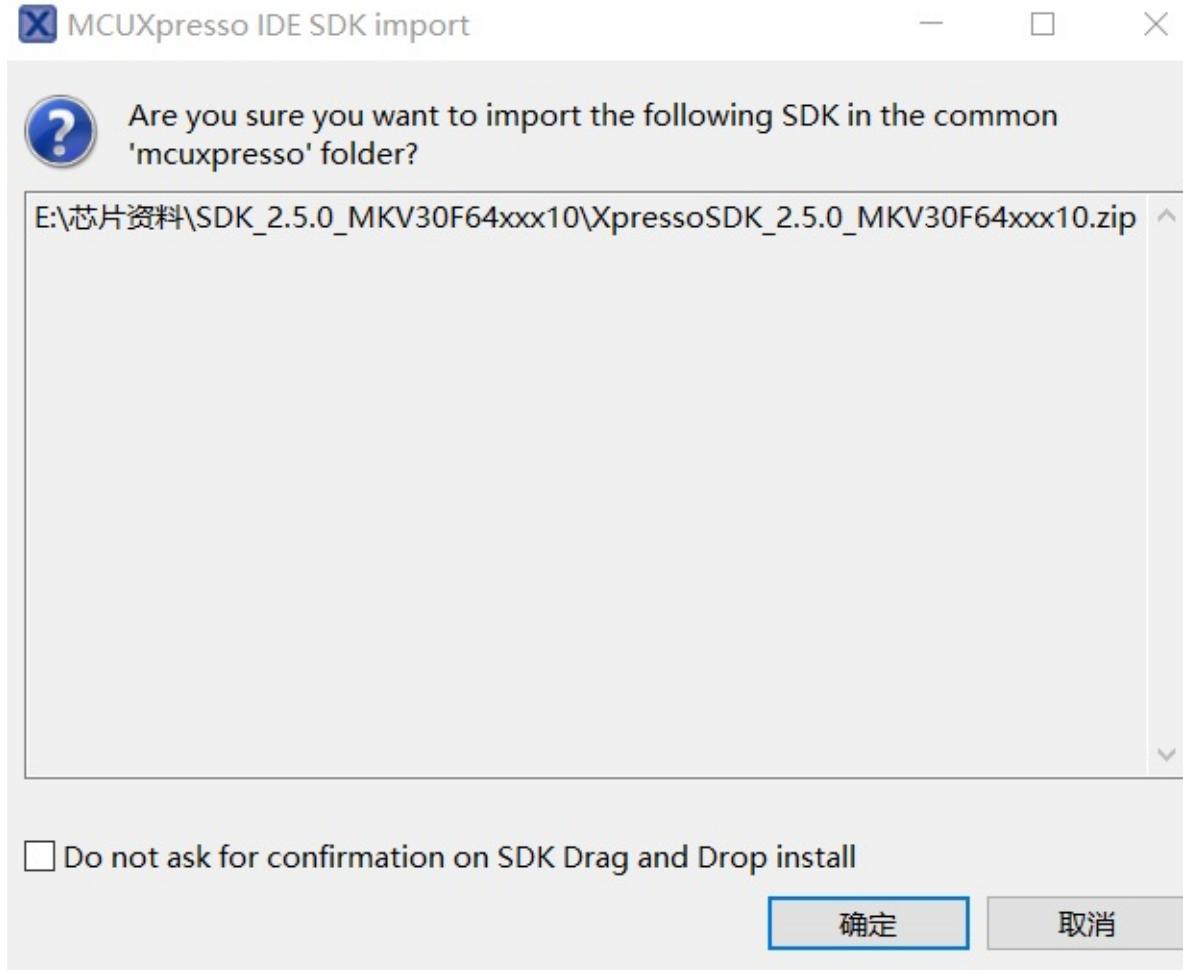
Step 1.将下载好的 SDK 压缩包拖动到视图的 Installed SDKs 中



### Installed SDKs

To install an SDK, simply drag and drop an SDK (zip file/folder) into the 'Installed SDKs' view. [Common Name]

Step 2.点击确定



Step 3.当视图的 InstalledSDKs 出现你放进来的 SDK 包则证明你已经完成了 SDK 的载入

Installed SDKs			
To install an SDK, simply drag and drop an SDK (zip file/folder) into the 'Installed SDKs' view. [Common 'mcuxpresso' folder]			
Name	SDK Version	Manifest Version	Location
<input checked="" type="checkbox"/> SDK_2.x_MKV30F64xxx10	2.5.0	3.4.0	<Common>\XpressoSDK_2.5.0_MKV30F64

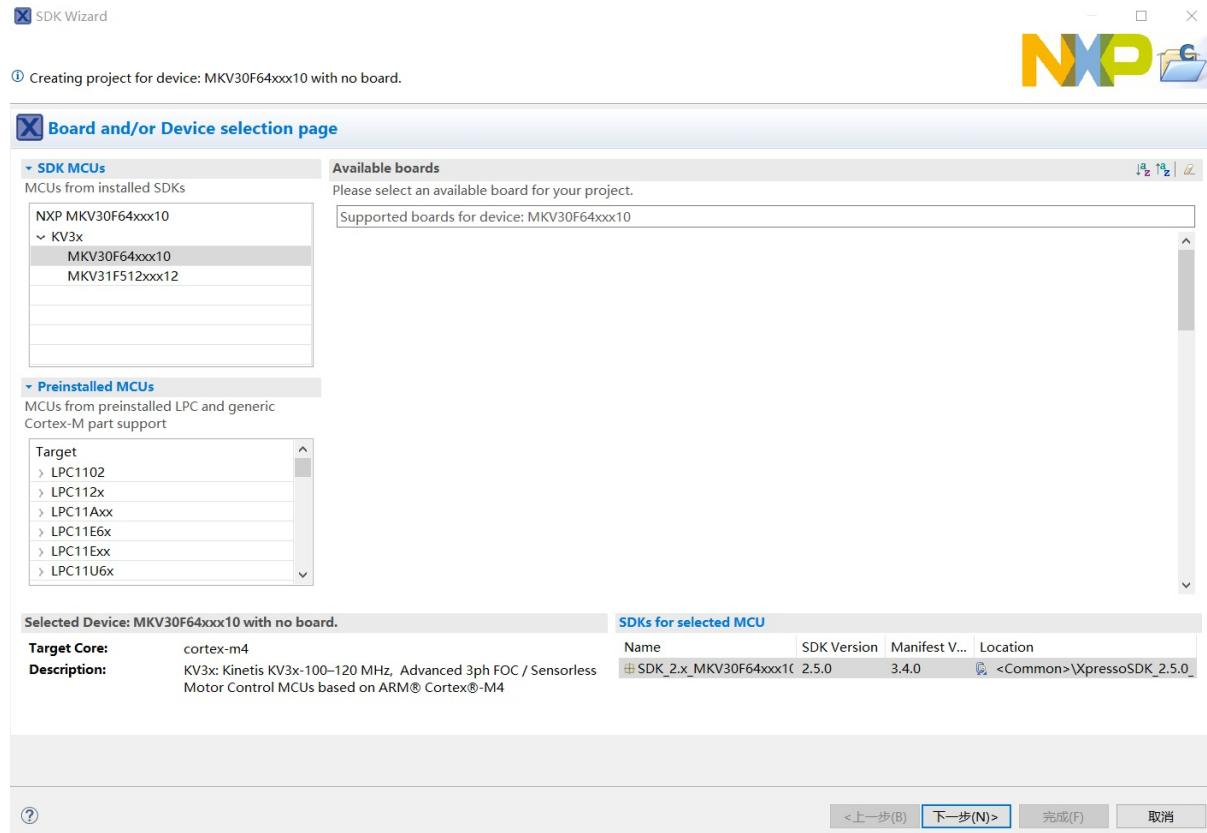


### 4.5.3.HelloWorld 示例

Step 1.点击视图中的 Quickstart Panel->New project...



Step 2.选择已被你载入的 SDK 包 KV3x, 然后选择对应的芯片, 在这里为 MKV30F64xxx10, 点击下一步



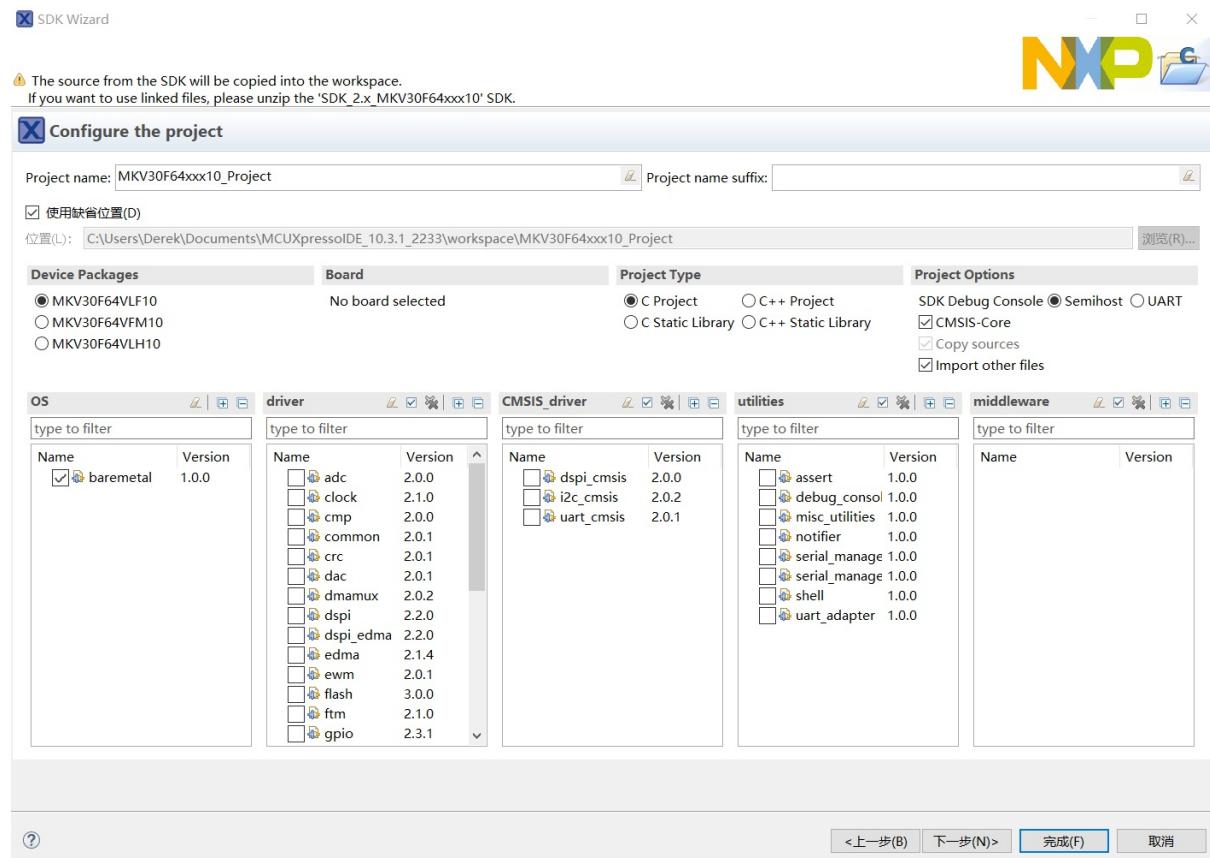
Step 3.SDK 包引入文件选择, 在这里不引入任何 Driver, 选择 , 默认配置, 点击完成

Project name prefix:生成的文件夹名的前缀

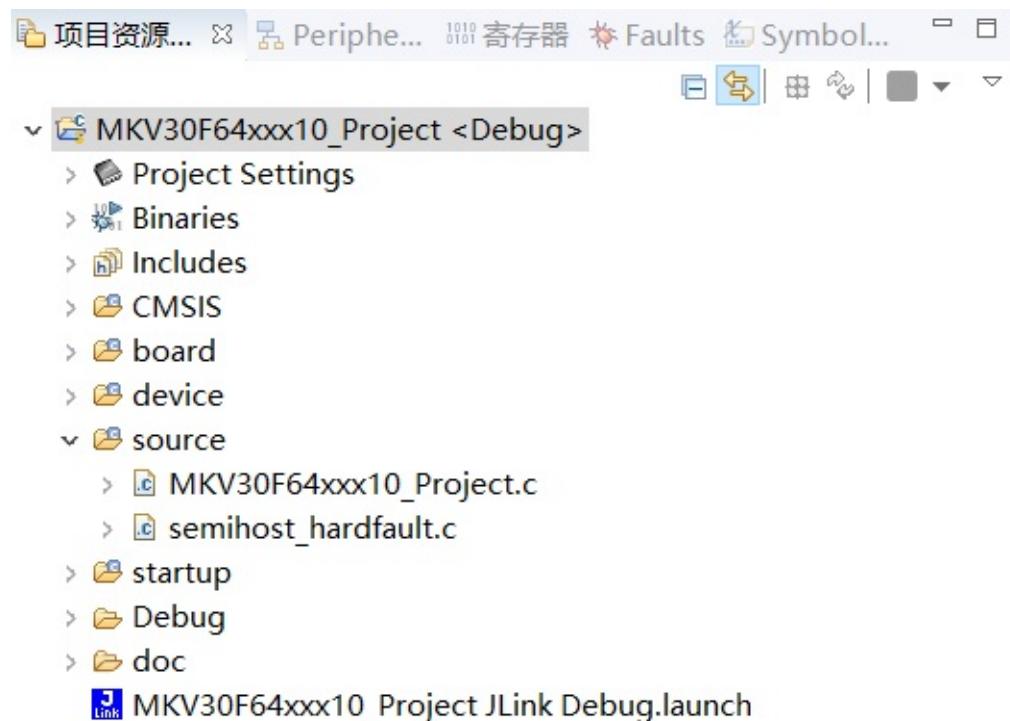
Project name suffix:生成的文件夹名的后缀

Device Packages:MKV30F64VLF10

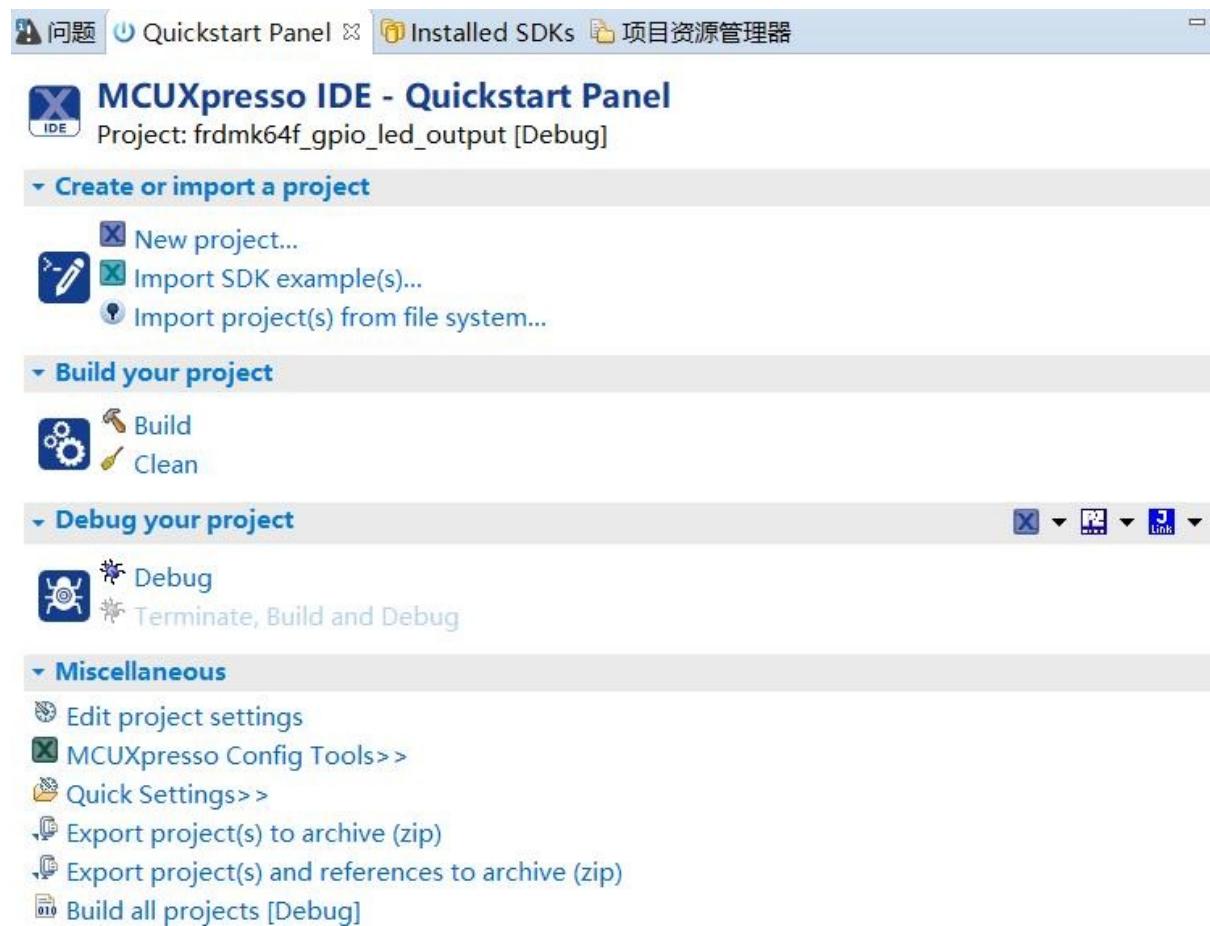
SDK Debug Console:Semihost(允许代码在 Debug Viewer 中查看)



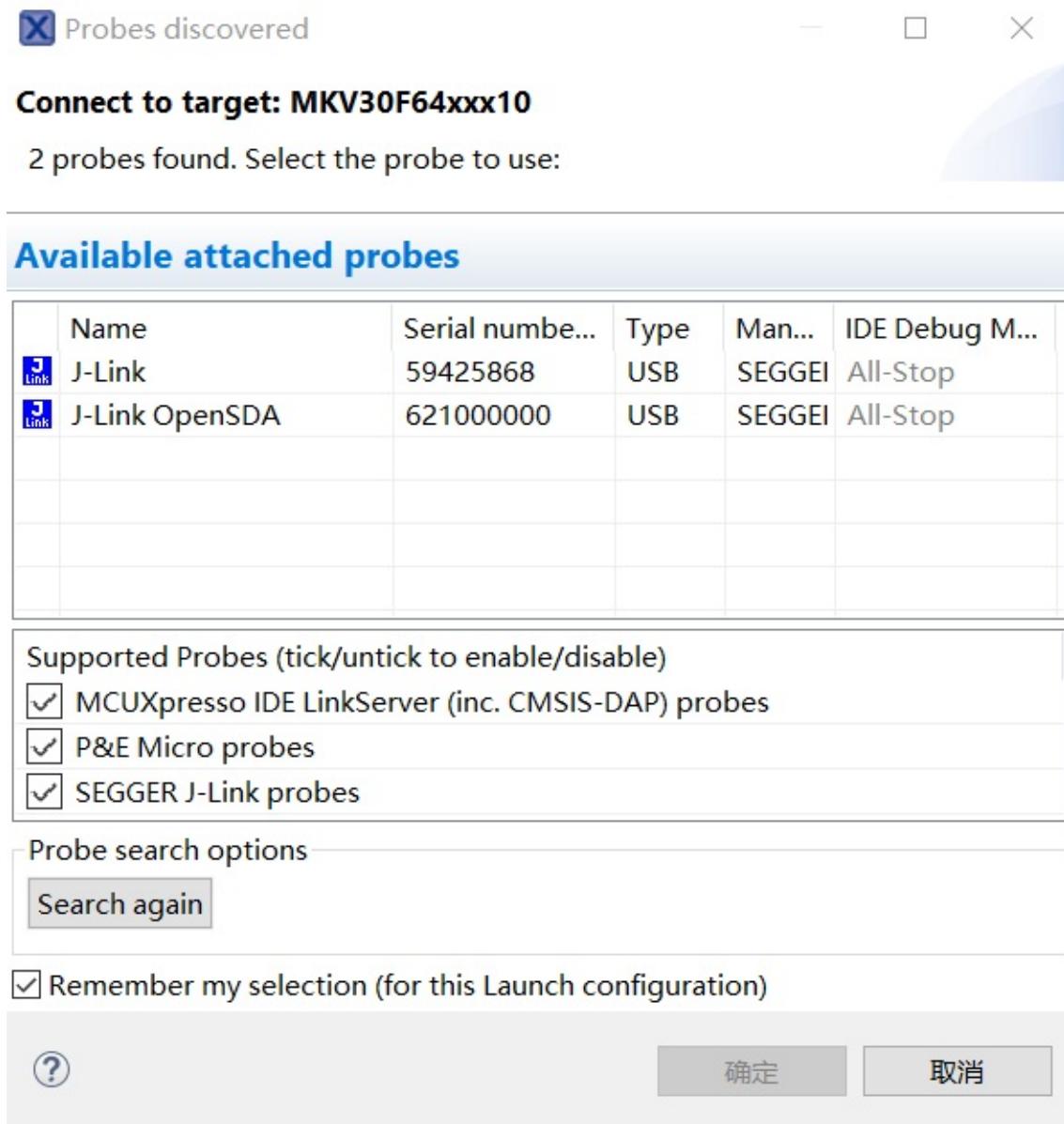
Step 4.查看视图的项目资源管理器，出现 SDK 项目，则新建工程完成



Step 5.把开发版接入电脑，点击视图的 Quickstart Panel，选择 Debug，右上的小图标也有对应的调试器，在这里采用 JLink Debug



Step 6.选择 J-Link 配置，点击确定



Step 7.由于开发版固件是 Jlink，所以会弹出以下窗口，点击 accept

## J-Link V6.42b - Terms of use

The connected emulator is an OpenSDA running a J-Link compatible firmware.  
In order to make use of this firmware, the following Terms Of Use must be accepted.

## TERMS OF USE

- 1) The firmware is only to be used with NXP target devices. Using it with other devices is prohibited and illegal.
- 2) The firmware is for use with evaluation boards only. It is not for use with custom hardware.
- 3) The firmware may only be used for development and/or evaluation purposes. It may not be used for production purposes.
- 4) The firmware is made available without any warranty and without support.
- 5) The firmware may be used with the OpenSDA platform only.

For more information, please refer to <http://www.segger.com/opensda.html>

If there is any doubt if a certain use may be considered within the foregoing scope,  
it is strongly recommended to consult SEGGER prior to use.

In order to contact SEGGER, please visit <http://www.segger.com/contact-us.html>

For development on target hardware, we recommend our industry leading

J-Link PRO (<http://www.segger.com/jlink-pro.html>)

J-Link Ultra+ (<http://www.segger.com/jlink-ultra-plus.html>)

J-Link PLUS (<http://www.segger.com/jlink-plus.html>)

J-Link (<http://www.segger.com/jlink.html>)

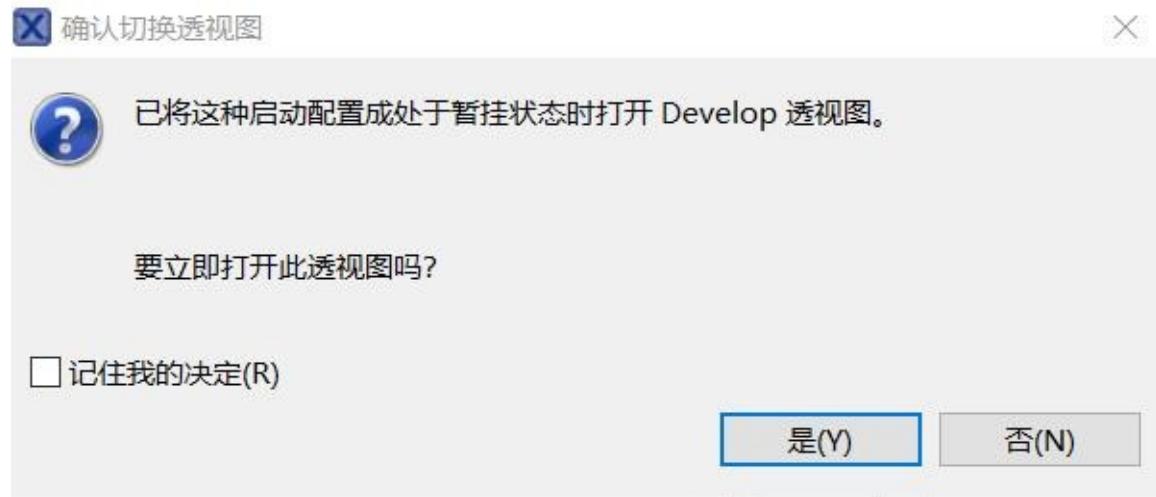
For professional production flash programming we recommend:

Do not show this message again for today

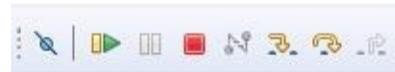
Decline

Accept

Step 8. 打开调试必要视图，点击是



Step 9. 你会发现界面已经切换成 Debug 模式，上方工具栏，有以下调试图标，点击 Start



Step 10. 在下方视图的控制台中看到打印出来的信息

The screenshot shows a software interface with a toolbar at the top containing icons for 'Installed SDKs', '属性' (Properties), '控制台' (Console) which is selected, '问题' (Problems), '内存' (Memory), 'Debugger Console', and a gear icon. Below the toolbar, the title bar displays 'MKV30F64xxx10\_Project JLink Debug [GDB SEGGER Interface Debugging] MKV30 [MCUXpresso Semihosting Telnet console for 'MKV30F64xxx10\_Project'']'. The main area is a terminal window showing the output of the SEGGER J-Link GDB Server V6.42b, specifically the 'Terminal output channel'. The text 'Hello World' is visible in the output.

```
MKV30F64xxx10_Project JLink Debug [GDB SEGGER Interface Debugging] MKV30
[MCUXpresso Semihosting Telnet console for 'MKV30F64xxx10_Project']

SEGGER J-Link GDB Server V6.42b - Terminal output channel
Hello World
```