MPLAB® Harmony 3 之基础篇（14）

--轻松创建 USB CDC Device 应用

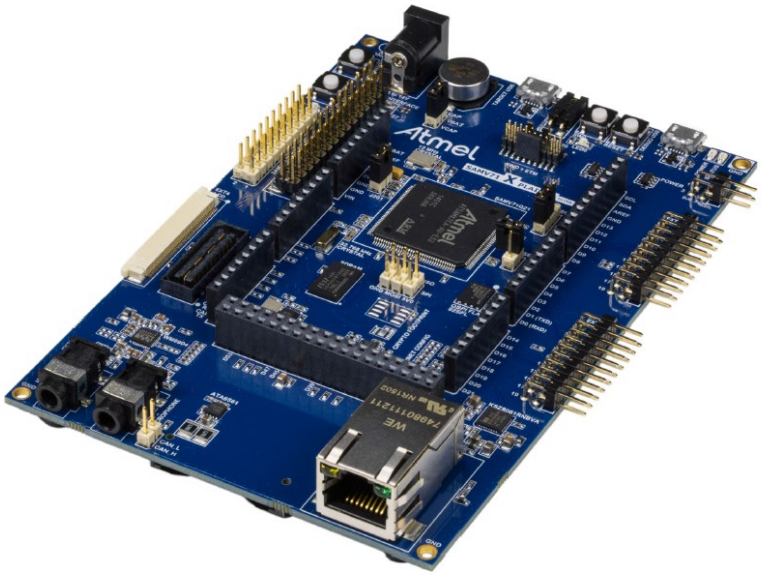
Microchip Technology Inc.
MCU32 产品部

一、 简介

本文主要介绍如何利用 MPLAB X IDE 创建一个工程，利用 MPLAB Harmony 3 Configurator(MHC)添加所需要的硬件驱动，USB 协议栈，其他服务等，在 SAM E70 Xplained Ultra Board 开发板上轻松创建一个完整的 USB CDC Device 的例程。

二、 硬件工具和软件平台

硬件：SAM E70 Xplained Ultra Board



*软件(开发工具和环境的安装和使用，见“**MPLAB® Harmony 3 之基础篇（01） -- Harmony 3 开发环境搭建**”)*：
MPLAB® X: v5.20 或者更新

XC32：　　v2.15 或者更新
Harmony 3：v3.20 或者更新

三、 详细步骤

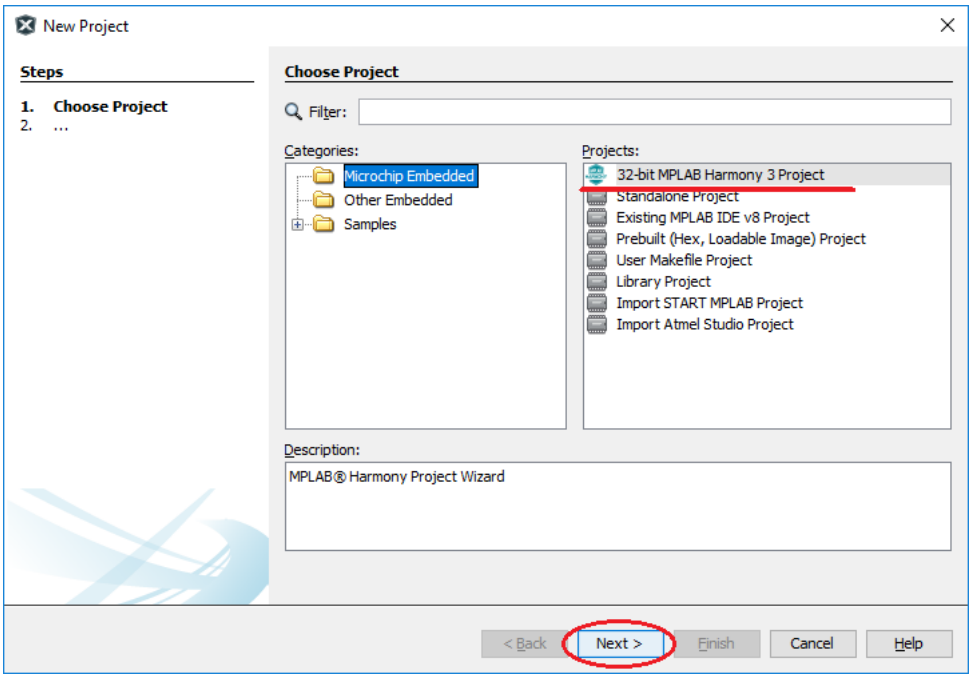
首先我们来了解一下我们需要用到的这一块 SAM E70 Xplained Ultra Board。这块开发板，使用的主控 MCU 是 SAME70Q21B。这是一颗 ARM Cortex-M7 内核的 MCU，主频高达 300MHz，带有 2MB 的 Flash，384KB 的 SRAM。该开发板同时自带了 EDBG，可以直接通过 Micro USB 连接电脑，用 MPLAB 进行下载，或者在线调试。

接下来我们就用 MPLAB X IDE 和 MHC 一步步地创建和配置基于 USB 的应用。

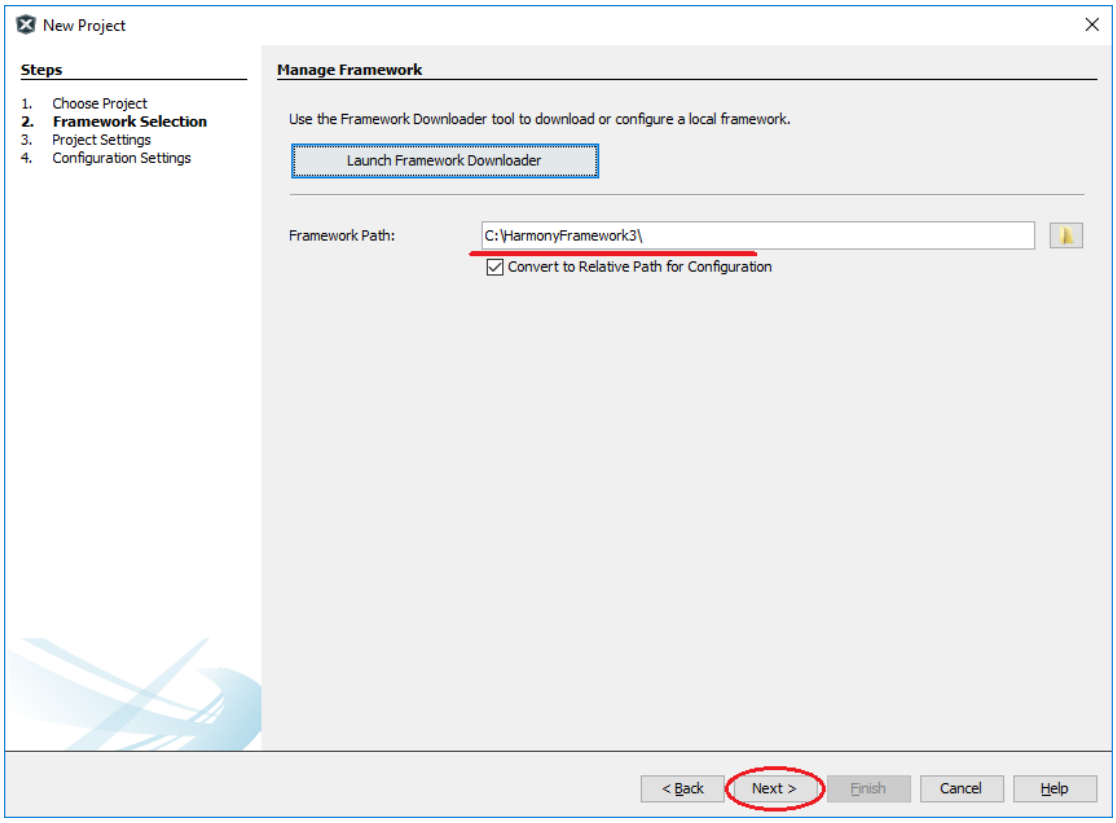
注：以下 MHC 配置里没有特别标注出来的地方，说明使用的是默认选项。

(一) 新建一个 MPLAB Harmony 3 的工程

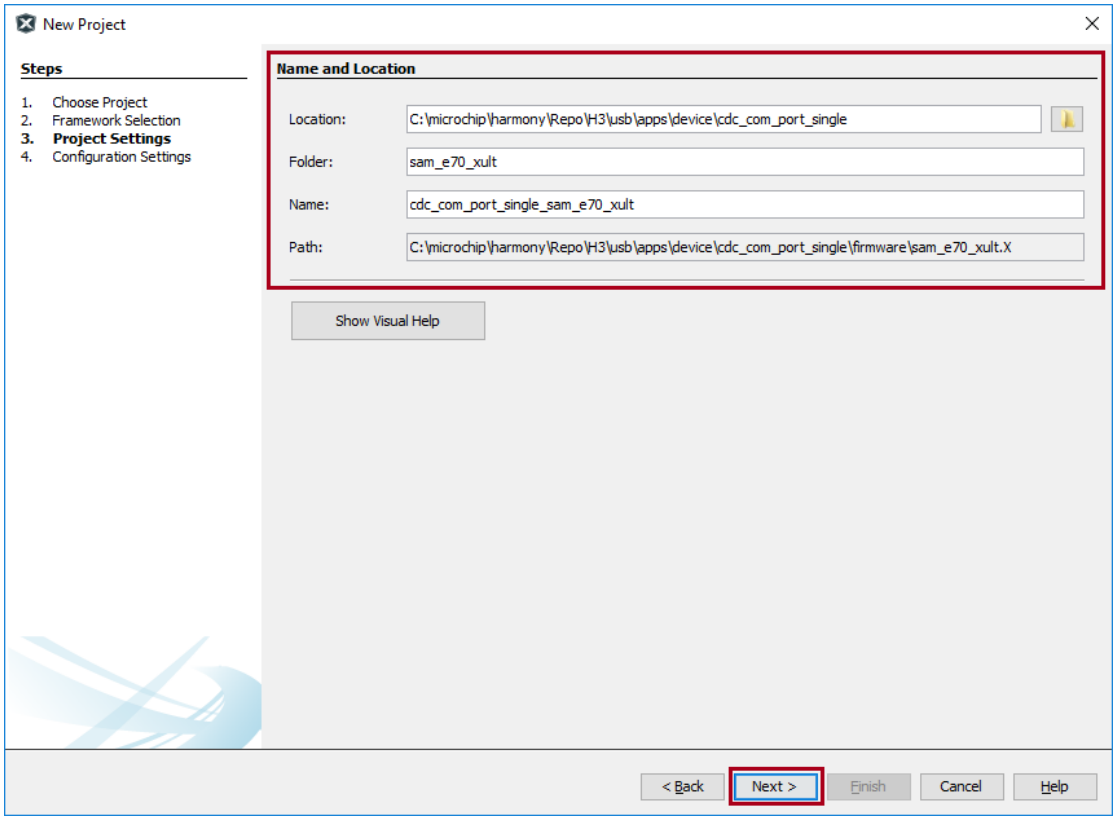
在 MPLAB X IDE 里点击 File > New Project：



选择“32-bit MPLAB Harmony Porject”，然后点击“Next”按钮。

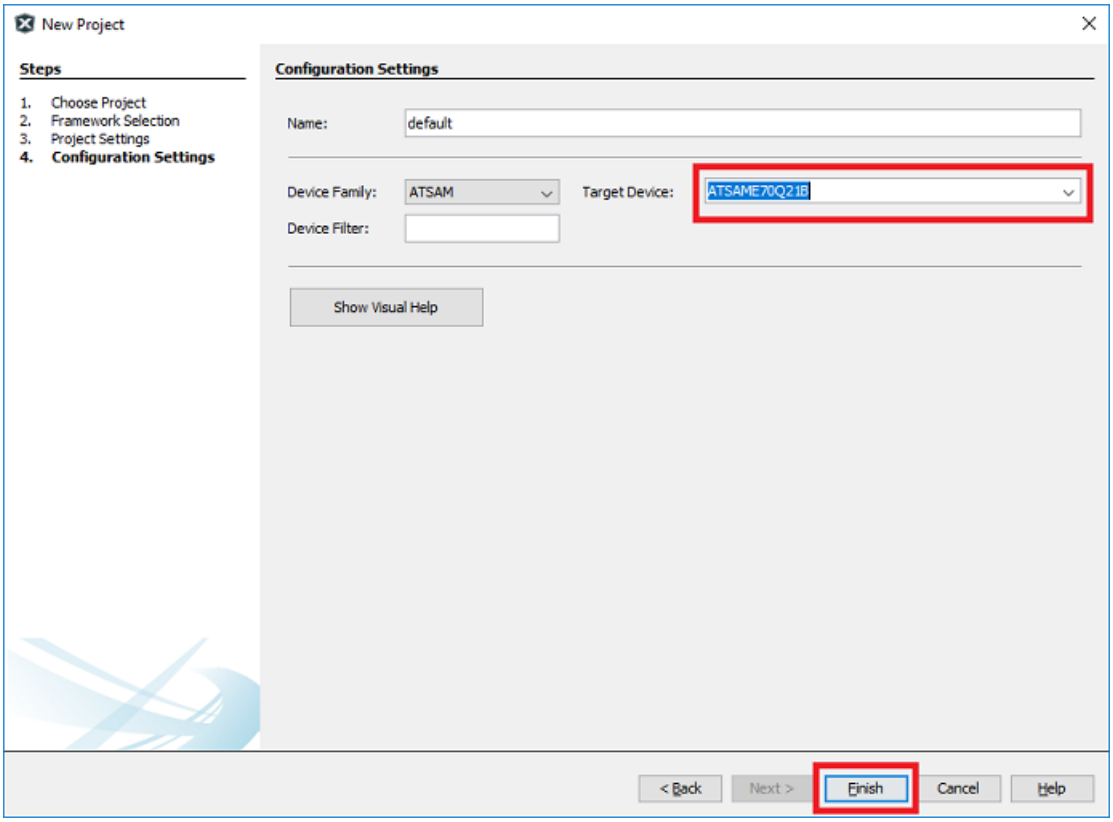


选择“Harmony Framework”路径，然后点击“Next”按钮。



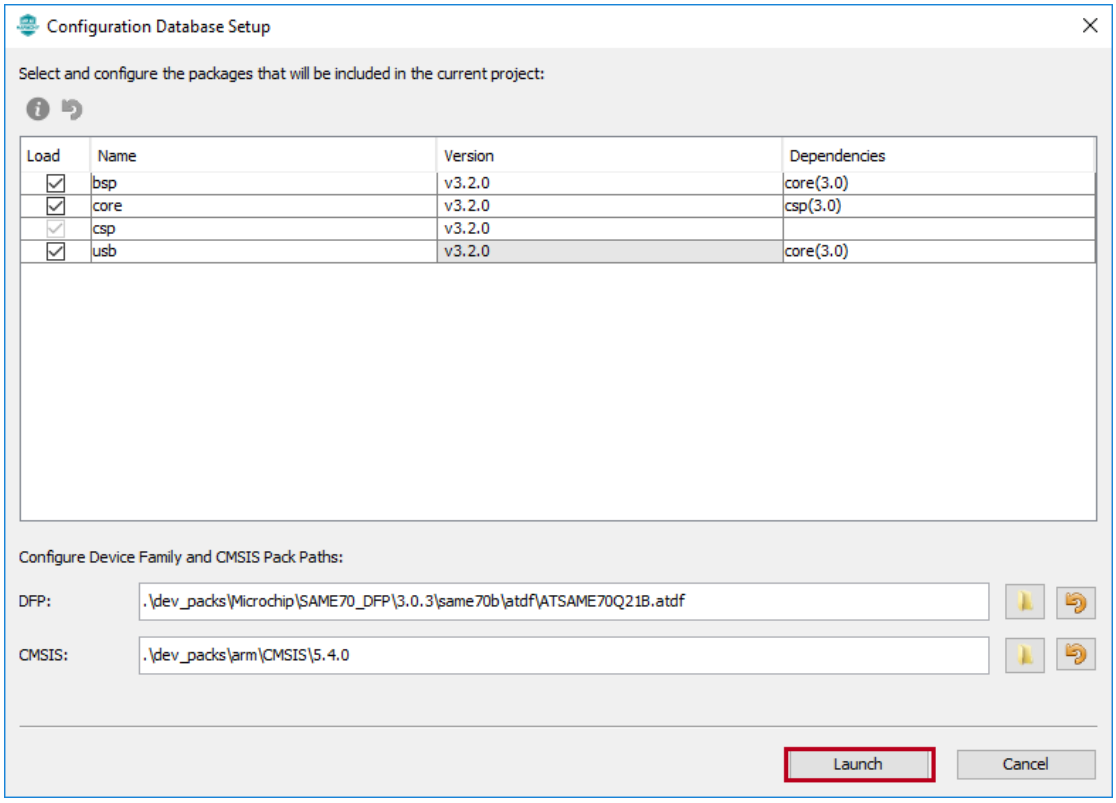
- Location: 创建项目所在的位置，比如 "cdc_com_port_single" 。
- Folder: 项目的文件夹名称，比如 “sam_e70_xult” 。
- Name: 项目的名称，比如 “cdc_com_port_single_sam_e70_xult” 。
- Path: 系统根据你的设定自动产生的路径。

在这里输入需要的项目名称，项目代码的存放位置，然后点击“Next”按钮。



这个页面最主要是选择我们需要使用的芯片，我们用的这个开发板的 MCU 是 ATSAM 的 ATSAME70Q21B。另外需要设置当前配置的名称，用于保存配置文件。然后点击“Finish”按钮。

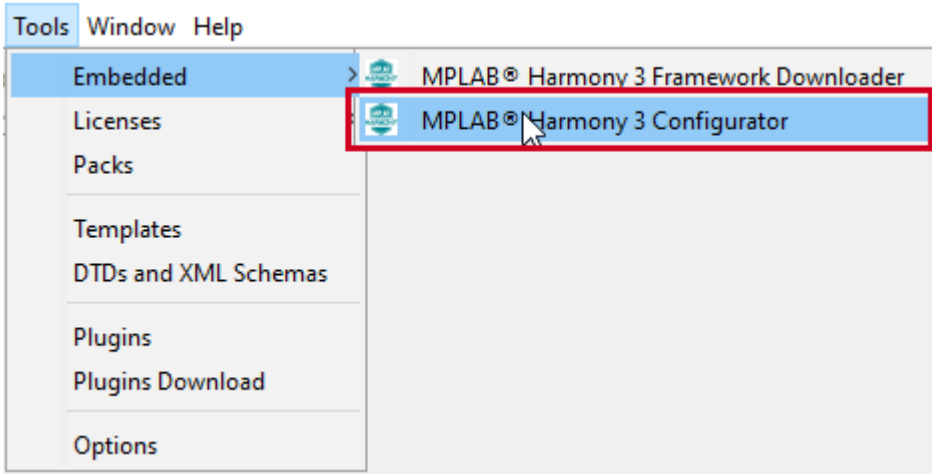
(二) 添加所需要的基础软件包

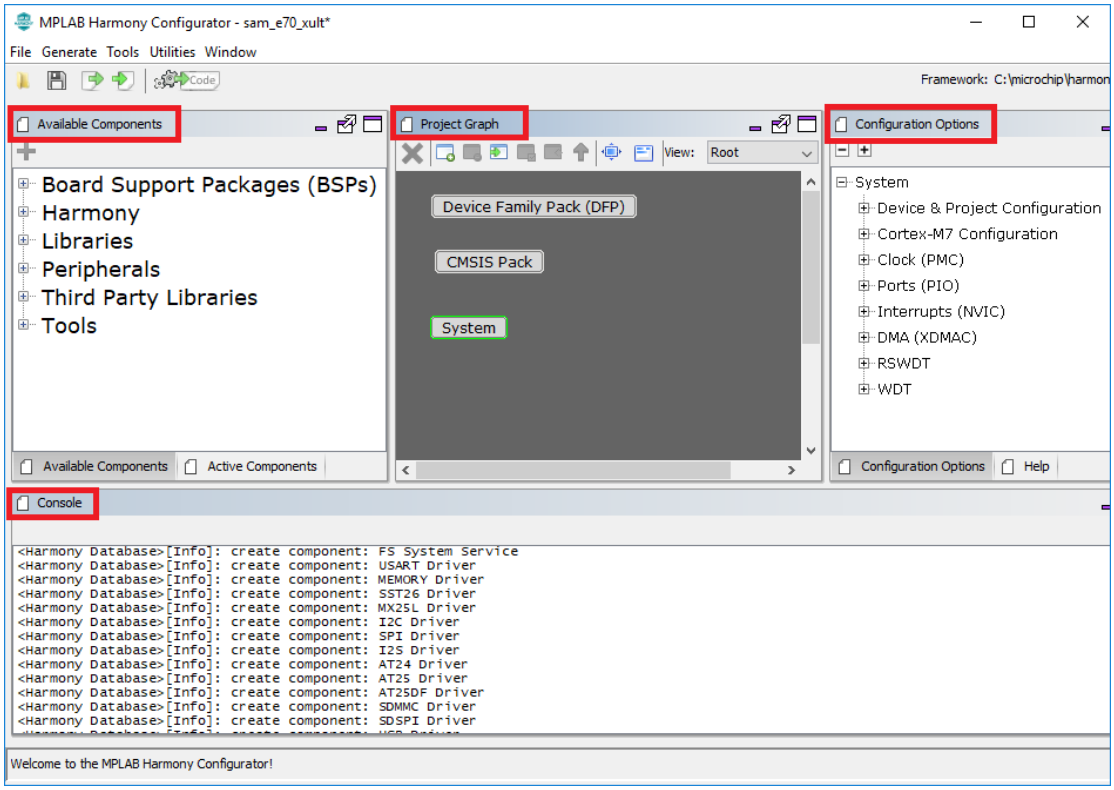


这个页面是选择我们需要加入使用的基础的 Harmony 软件 package。在这里我们选择 bsp，core，usb 三个主要的 package。点击 “Launch” 完成。

(三) 往当前新建工程里面添加 BSP 功能

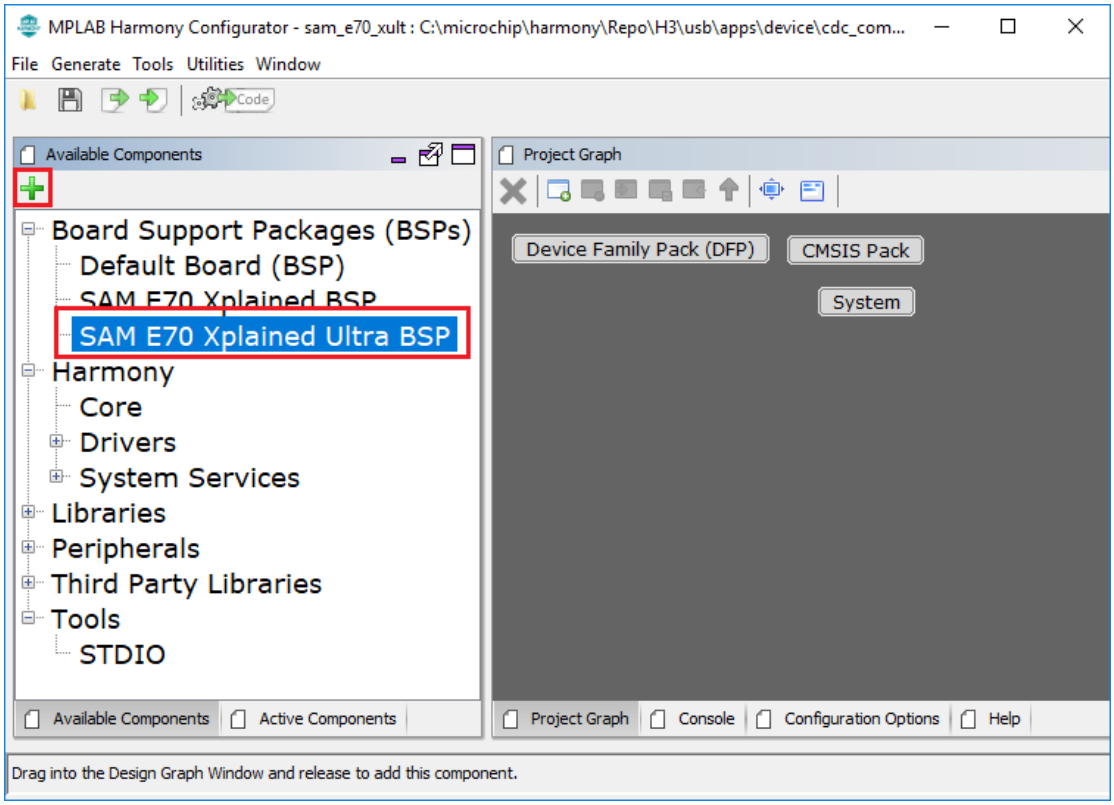
从 MPLAB X IDE 的 Tools 菜单，选择 Embedded，MPLAB Harmony3 Configurator。





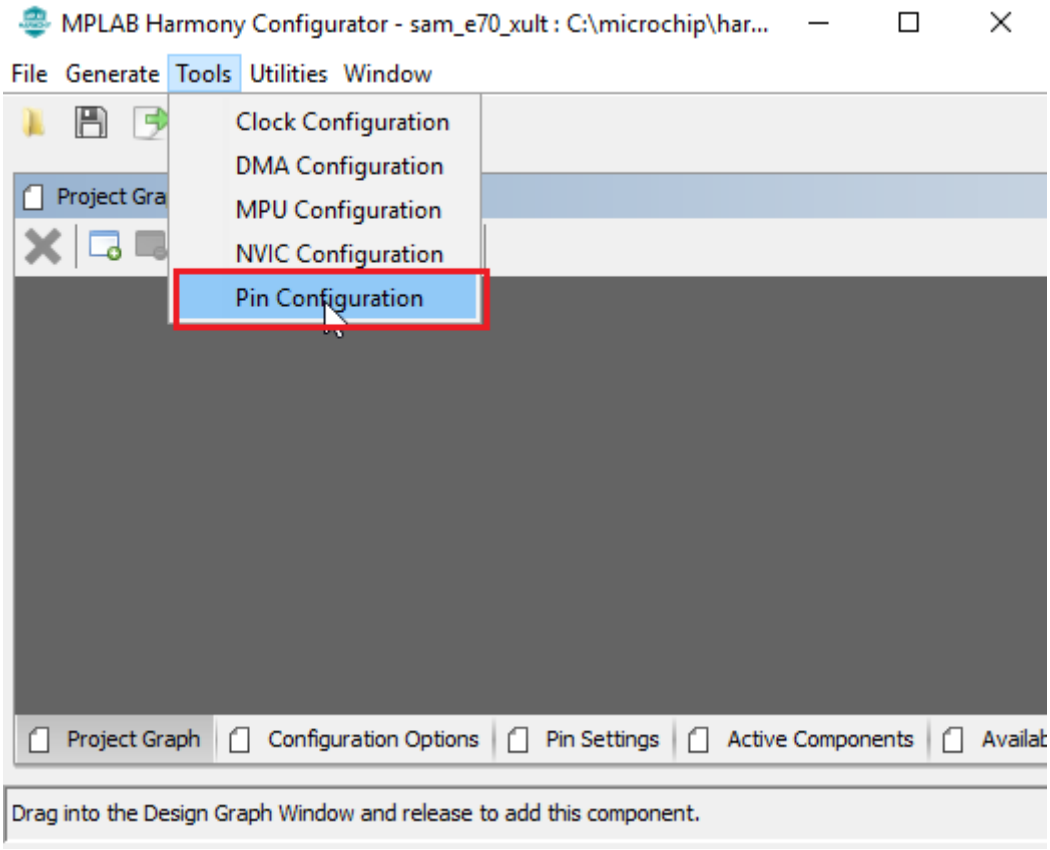
在 MHC 的界面，我们可以看到 MHC 主要有以下几个主要的页面：

- Available Components
- Active Components
- Project Graph
- Configuration Options
- Console Window.



从 Available Components 页面中，选择添加当前使用开发板的 BSP。

(四) 配置相关的 I/O 引脚



打开 MHC 的 Tools->Pin Configuration 来进行 I/O 的配置。

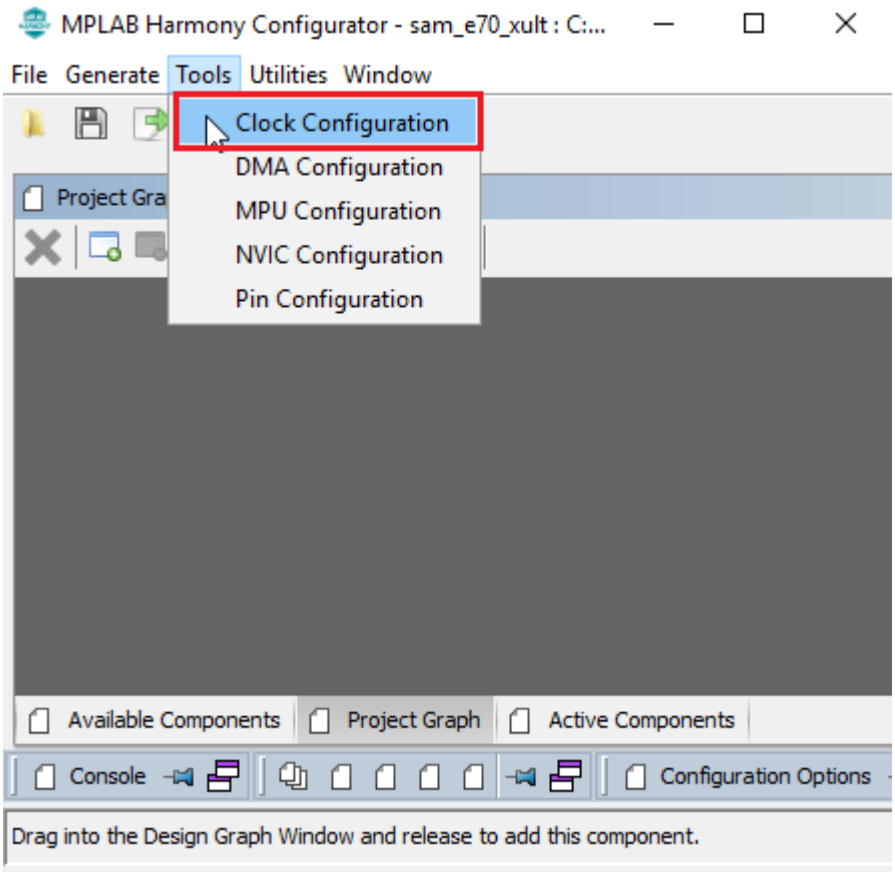
SAM E70 Xplained Ultra Board 的 I/O 配置如图：

Pin Settings

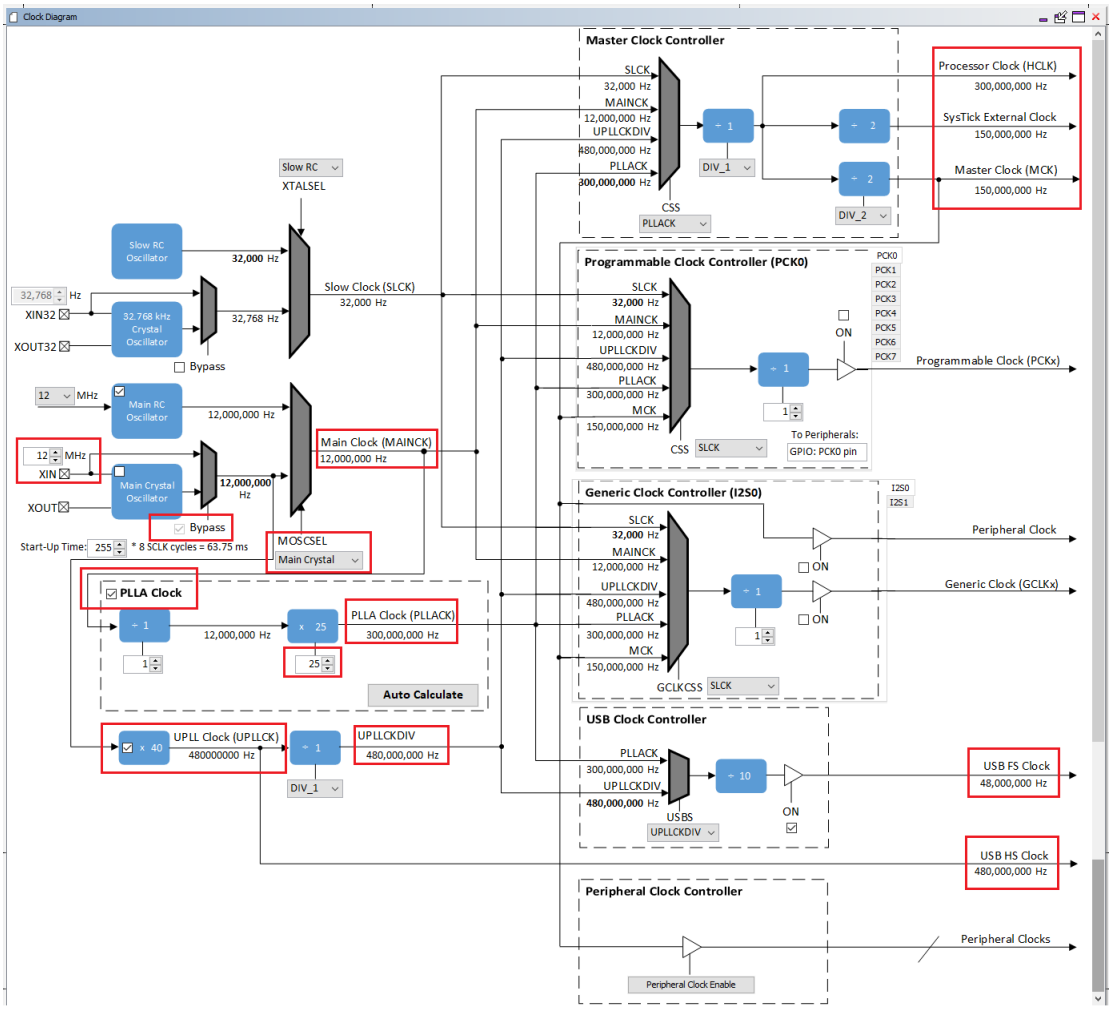
Order: Pins | Table View

Pin Number	Pin ID	Custom Name	Function	Direction	Latch	Open Drain	PIO Interrupt	Pull Up	Pull Down	Glitch/Debounce Filter
64	PA11	SWITCH	SWITCH_AL	In	Low	☐	Disabled	☑	☐	Disabled
73	PA5	LED1	LED_AL	Out	High	☐	Disabled	☐	☐	Disabled
141	PB8	USB_VBUS_SENSE	GPIO	In	n/a	☐	Disabled	☐	☑	Disabled

(五) 配置系统的 Clock



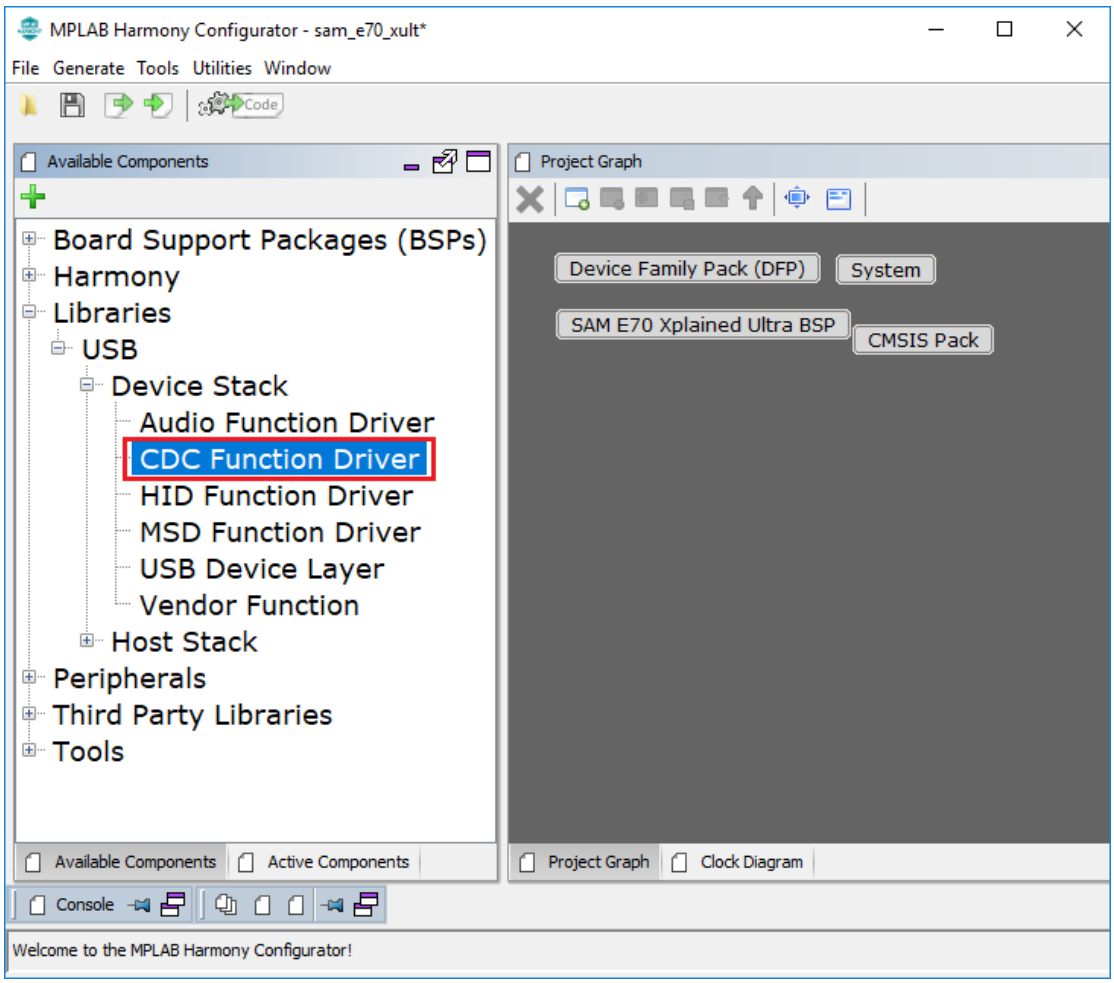
打开 MHC 的 Tools->Clock Configuration 来进行系统 Clock 的配置。

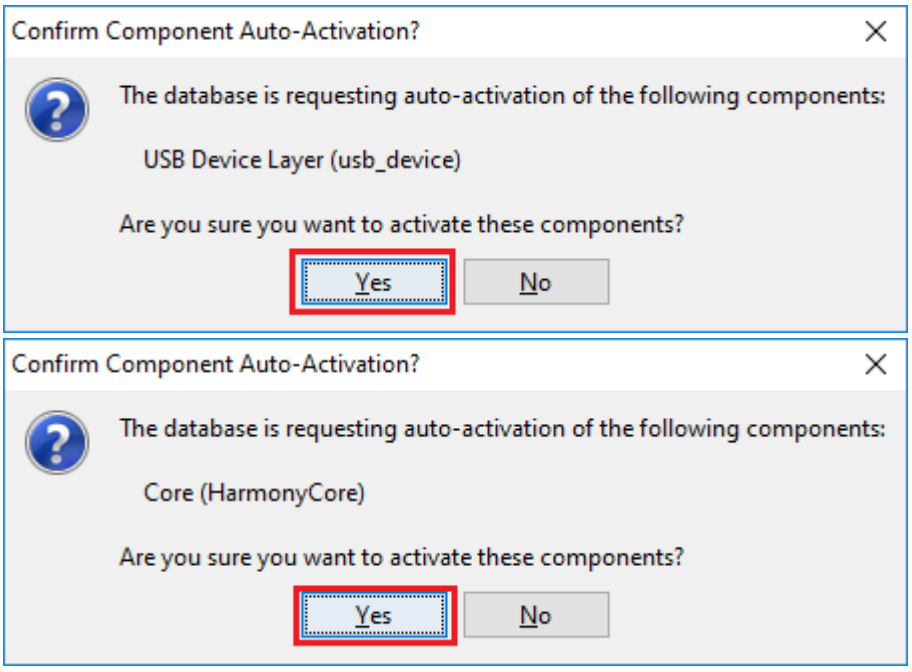


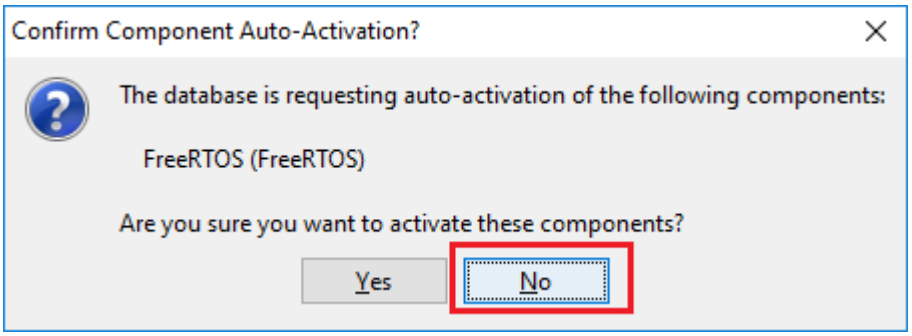
SAM E70 Xplained Ultra Board 上有一个 12MHz 的外部晶振。这里我们需要把它使能。这个主要是输出给 PLLA Clock。PLLA Clock 输出 300MHz 的 Clock。最后 Master Clock Controller 输出 150MHz 的 Master Clock (MCK)，150MHz 的 SysTick External Clock (SysTick)，和 300MHz 的 Processor Clock (HCLK)。

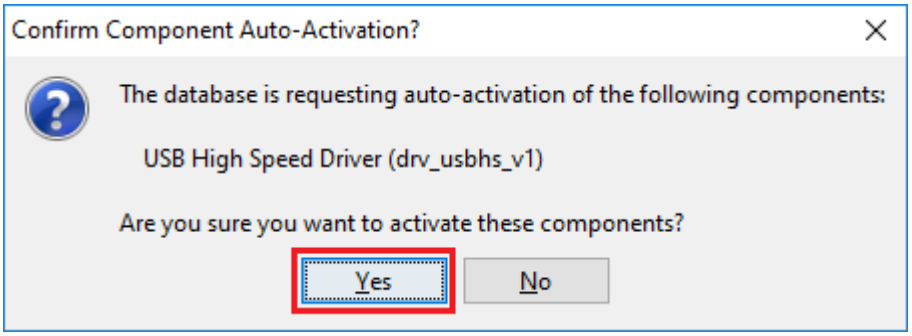
(六) 添加 USB Device 相关功能

打开 Available Components -> Library -> USB -> Device Stack。双击“CDC Function Driver”将其加入项目中。在添加的过程中，系统会提示加入其他项目组件，点击“Yes”一并加入。FreeRTOS 在这里暂不需要，点击“No”选择忽略。

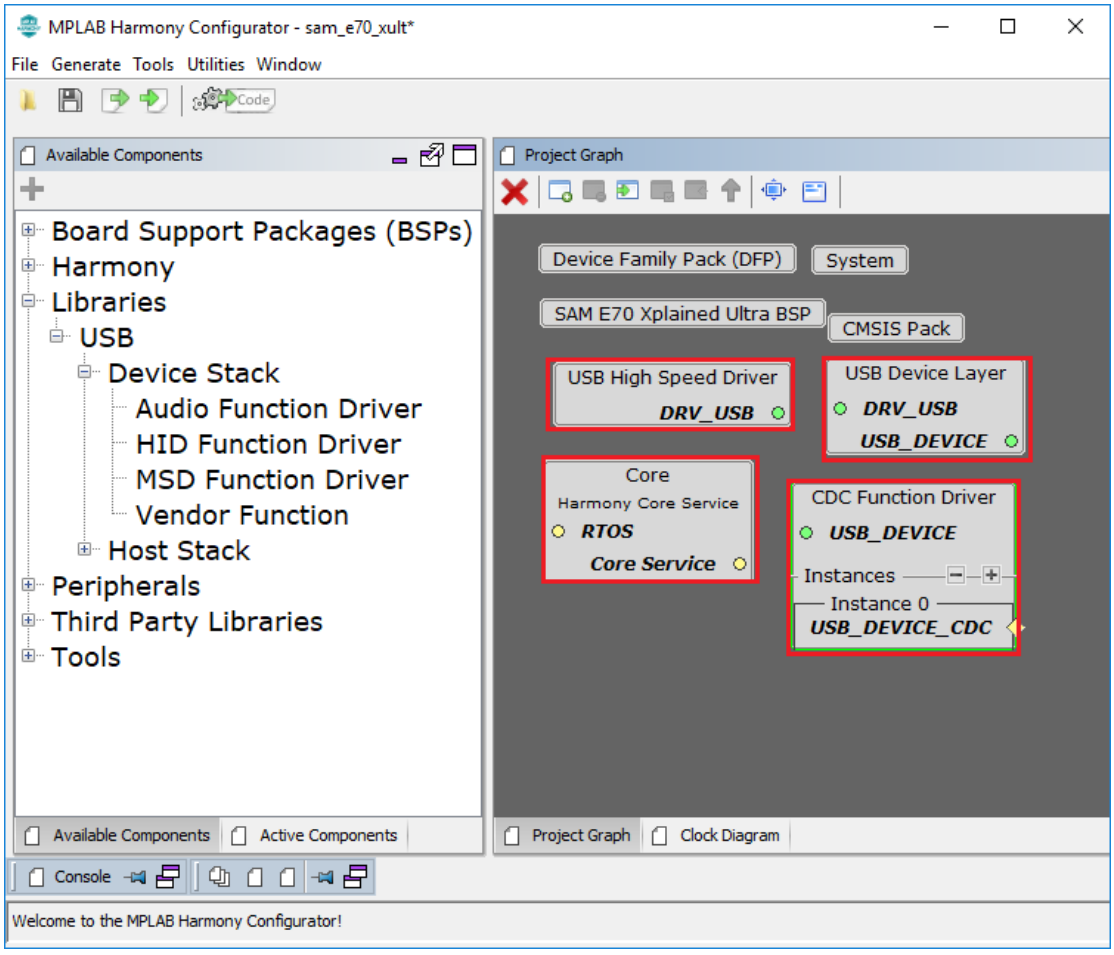

MPLAB Harmony Configurator - sam_e70_xult*
File Generate Tools Utilities Window
Code
Available Components
Board Support Packages (BSPs)
Harmony
Libraries
USB
Device Stack
Audio Function Driver
CDC Function Driver
HID Function Driver
MSD Function Driver
USB Device Layer
Vendor Function
Host Stack
Peripherals
Third Party Libraries
Tools
Available Components
Active Components
Project Graph
Device Family Pack (DFP)
System
SAM E70 Xplained Ultra BSP
CMSIS Pack
Project Graph
Clock Diagram
Console
Welcome to the MPLAB Harmony Configurator!


Confirm Component Auto-Activation?
The database is requesting auto-activation of the following components:
USB Device Layer (usb_device)
Are you sure you want to activate these components?
Yes
No
Confirm Component Auto-Activation?
The database is requesting auto-activation of the following components:
Core (HarmonyCore)
Are you sure you want to activate these components?
Yes
No





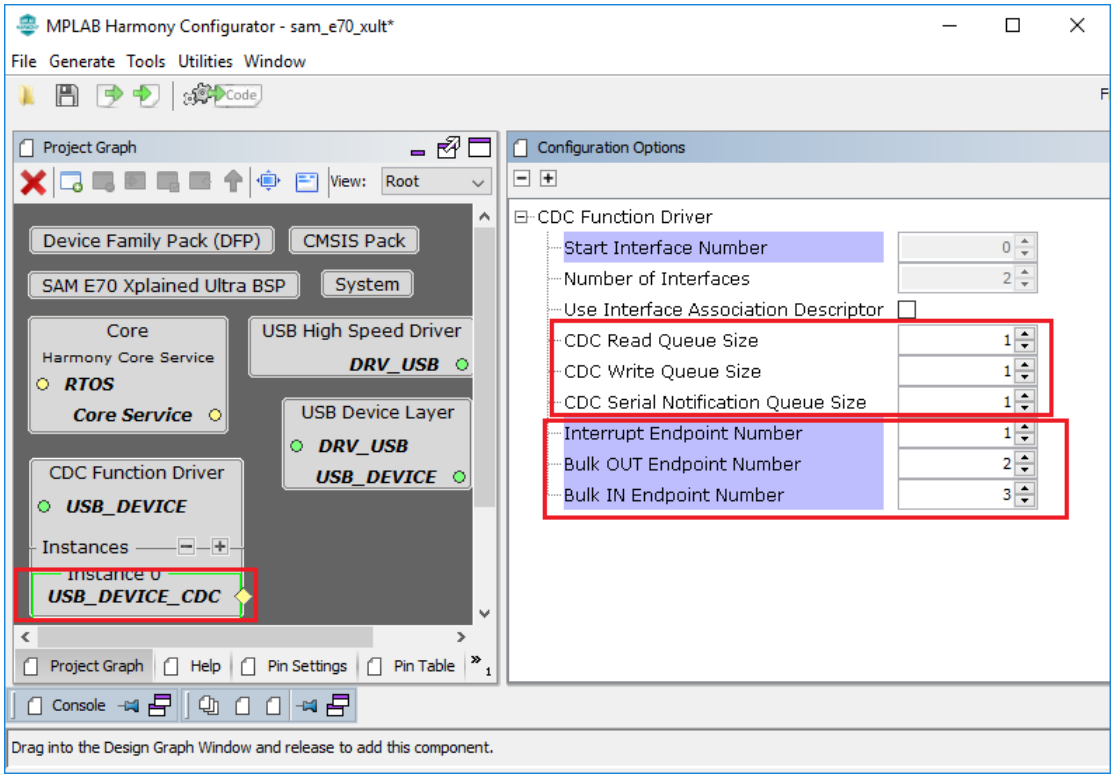
在添加了所有的相关组件之后，Project Graph 如下图所示：



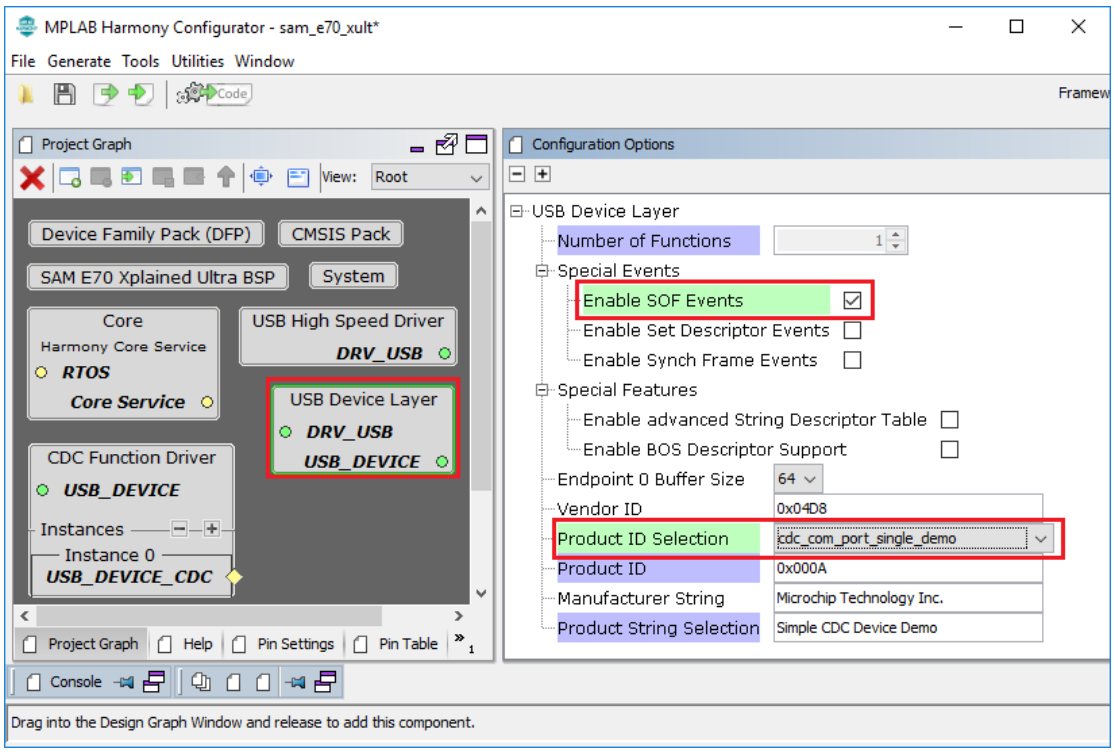
(七) 配置 USB Device 相关的组件

前往 Project Graph 然后选择所需配置的组件。所有跟该组件相关的配置，都会在 Configuration Options 中显示，并且所有的选项都已经配置有默认值。

配置 CDC Function Driver：
在 CDC Function Driver 中选择 Instance 0(USB_DEVICE_CDC)。用户可以根据项目的需求在 Configuration Options 中修改 Endpoint Numbers(Interrupt/Bulk OUT/Bulk IN)的配置。

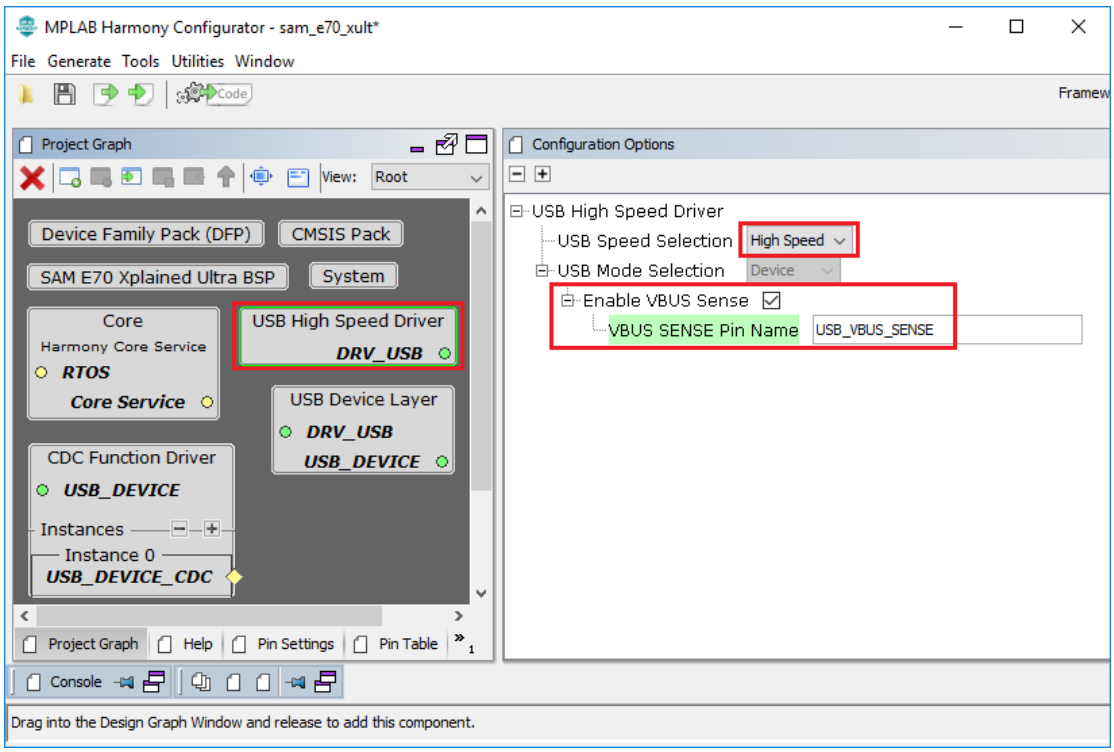


配置 USB Device Layer:
在这个配置页面中，需要把“Enable SOF Events”勾上。这样的话，在应用运行过程中，就可以得到所有的 SOF 的通知。在 Product ID Selection 中，选择“cdc_com_port_single_demo”。其他的使用默认值就可以了。



配置 USB Controller Driver：

在 Project Graph 中选择 USB High Speed Driver。在 USB Speed Selection 项，当前项目选择 High Speed。同时需要把 Enable VBUS Sense 勾上。该功能会默认使用一个名为 USB_VBUS_SENSE 的 GPIO。如果是其他的板子，用户可以任意选择一个 GPIO 作为 USB_VBUS_SENSE。但是需要注意该 GPIO 在配置的时候，必须将其配置为 GPIO IN，并且 Pull-down 使能。

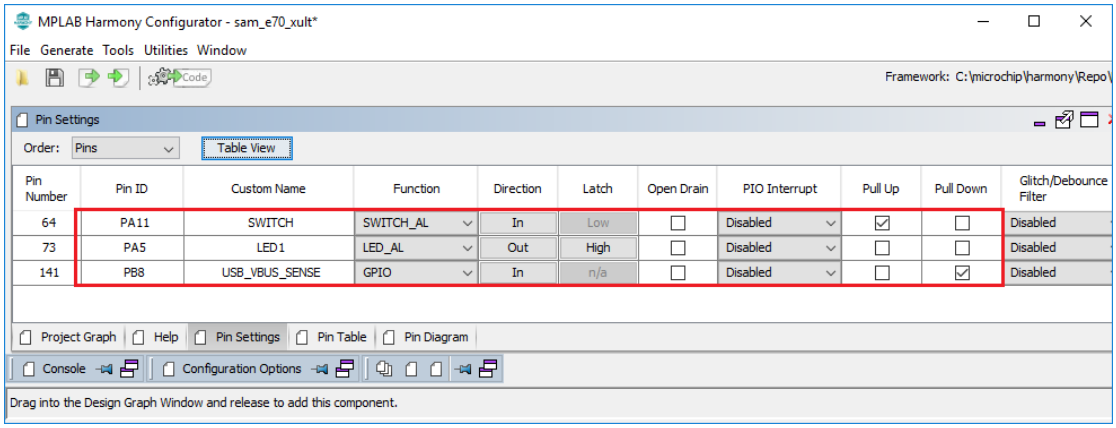


接下来我们需要打开菜单 MHC -> Tools -> Pin Configuration，在 Pin Settings 页面根据 SAM E70 Xplained Ultra Board 的硬件配置，把相应的 GPIO 配置好。下面是一个列表：

PB08 -> USB_VBUS_SENSE (GPIO IN with Pull-down)

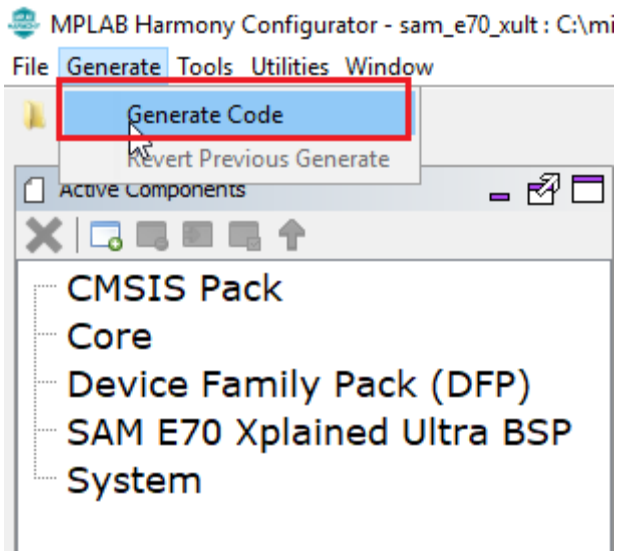
PA11 -> SWITCH (GPIO IN with Pull-up)

PA05 -> LED1 (GPIO OUT Latch High)

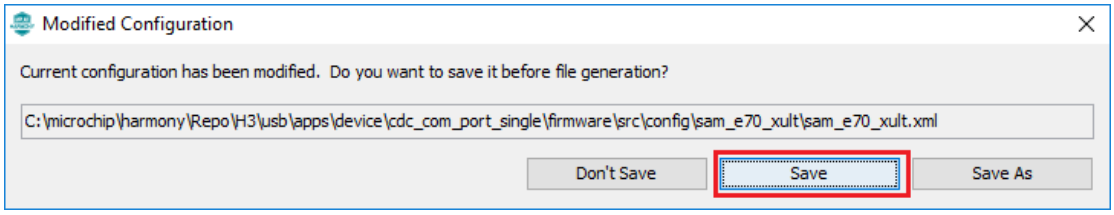


(八) 产生代码

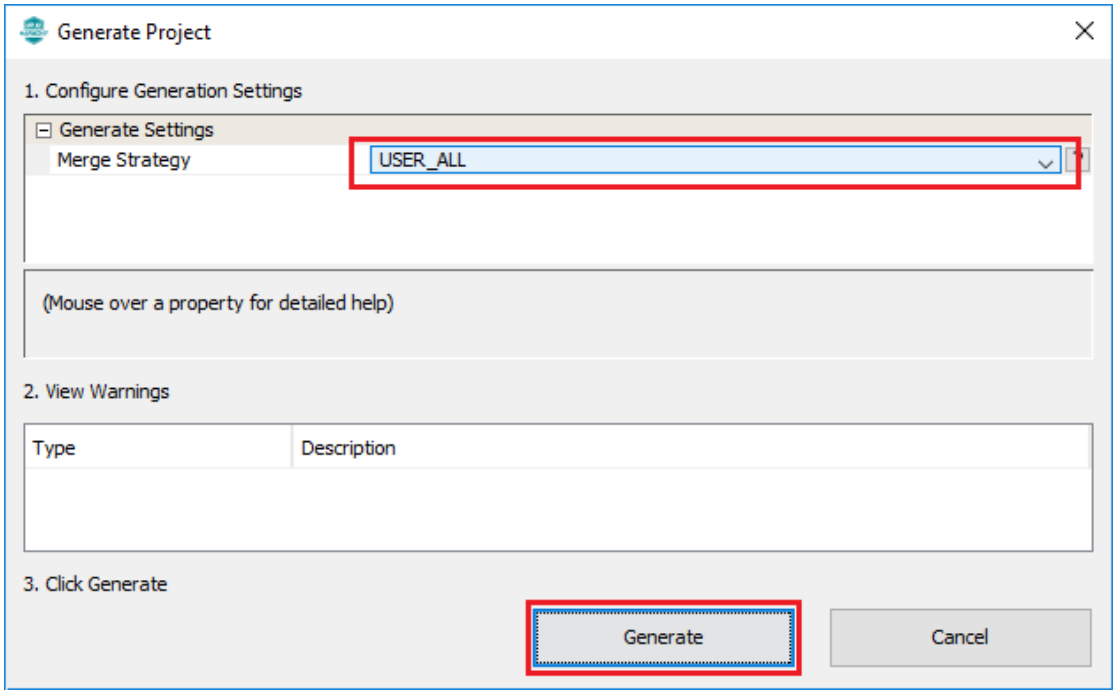
打开 Generate -> Generate Code 来产生项目代码



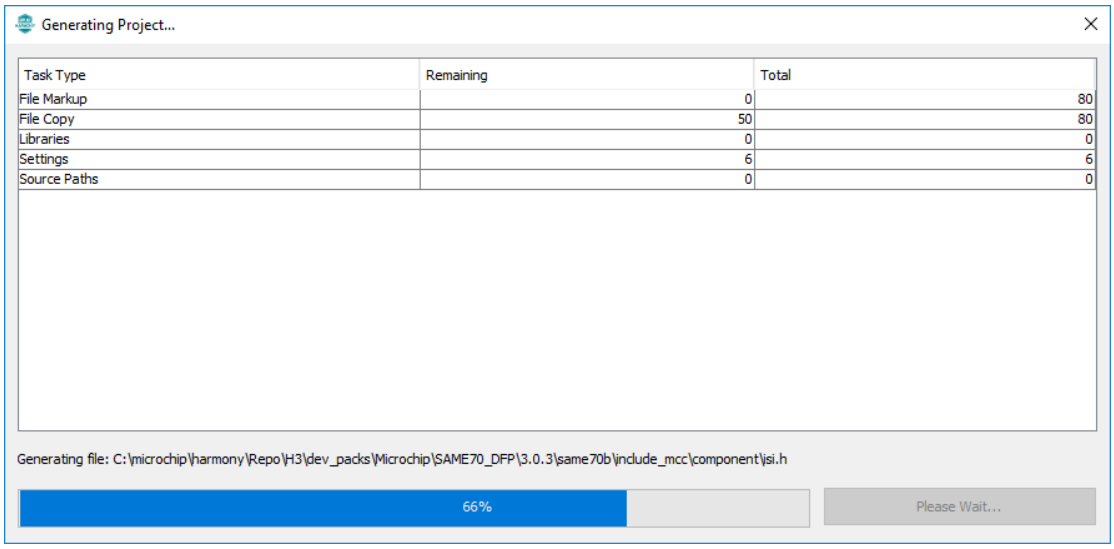
在弹出来的窗口，选择保存当前的项目配置



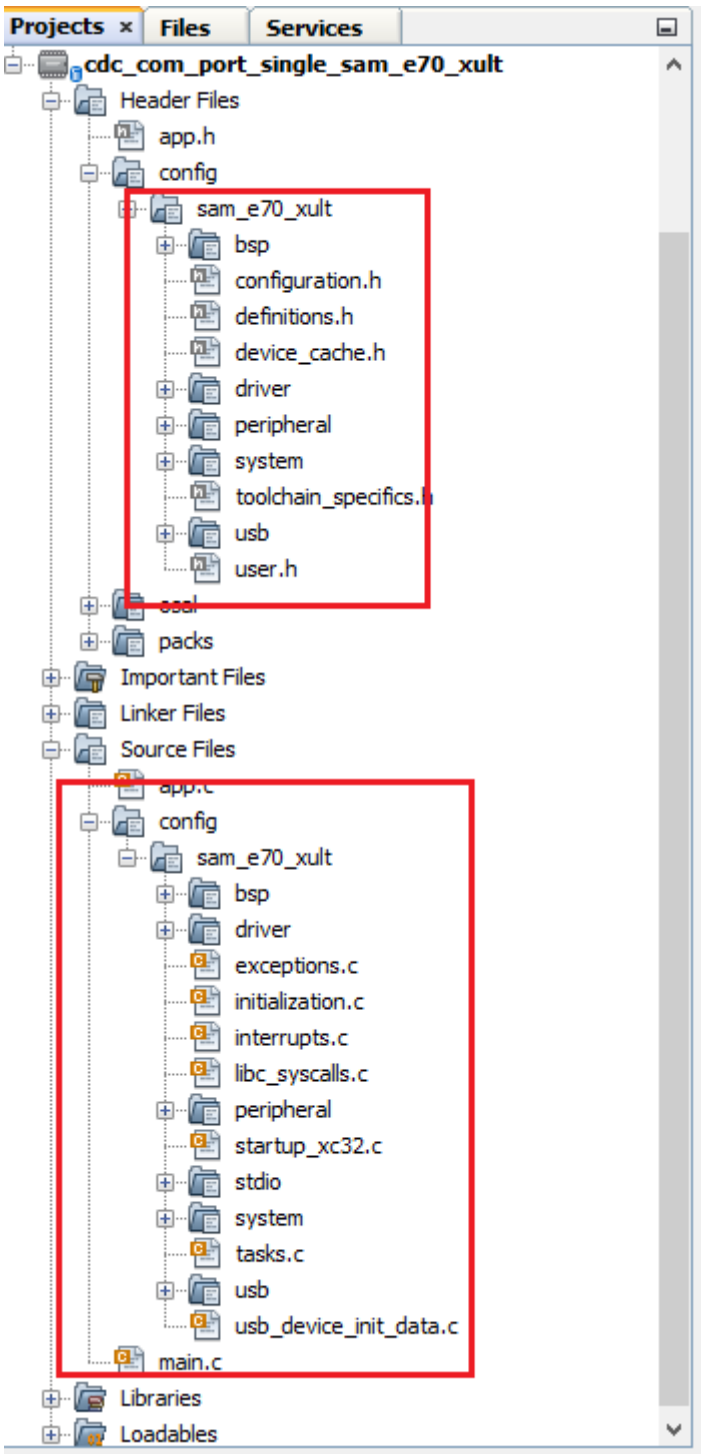
选择所需的 Merge Strategy



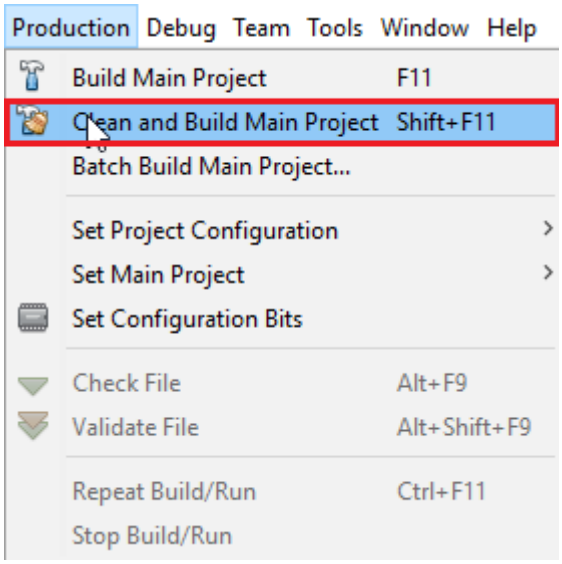
然后就会看到一个 MHC 生成代码的进度窗口



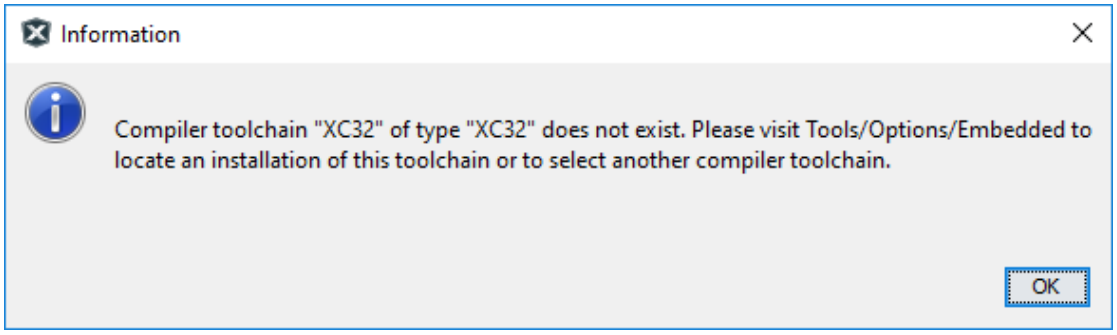
完成之后，就可以看到下面的代码列表



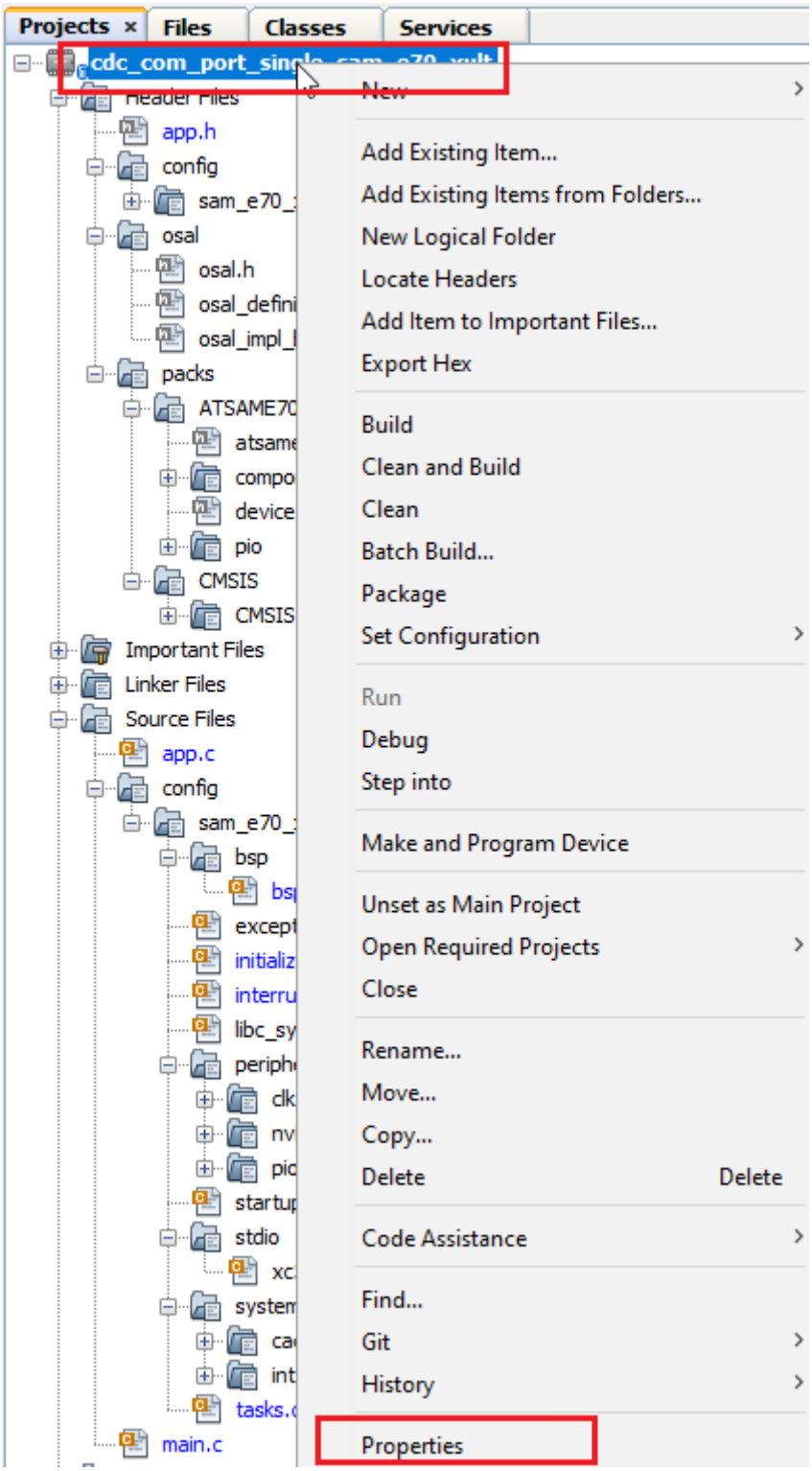
然后可以打开 Production -> Clean and Build Main Project 来进行编译。



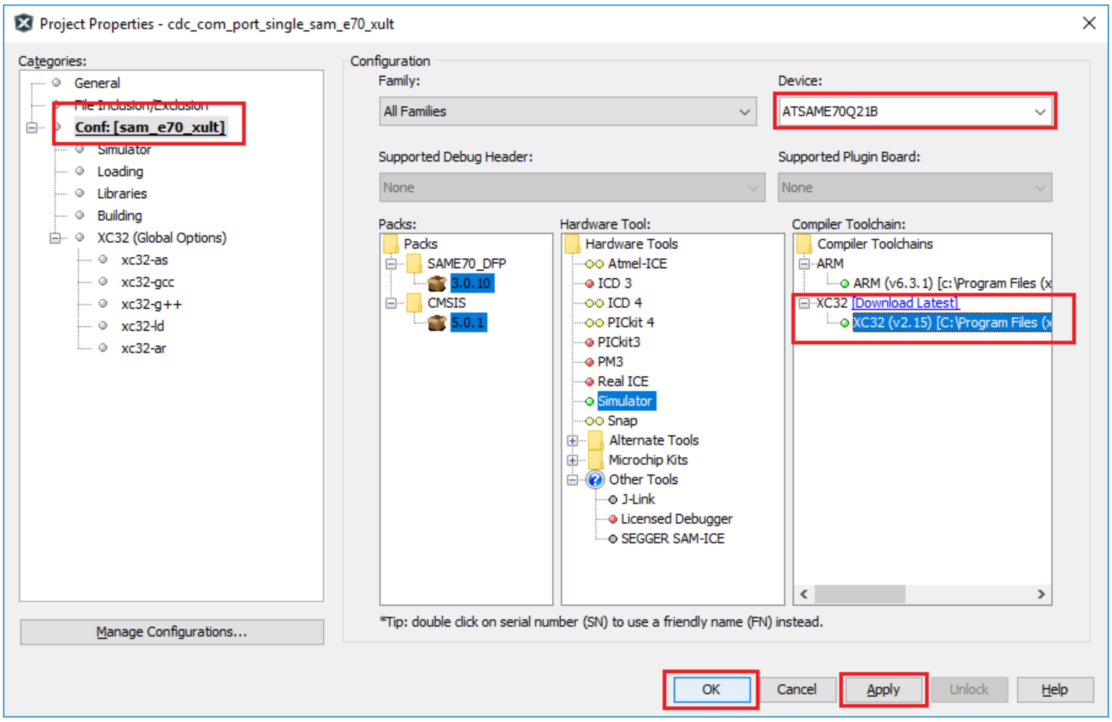
如果在编译的时候出现了下面的提示，需要配置一下项目属性里的编译工具



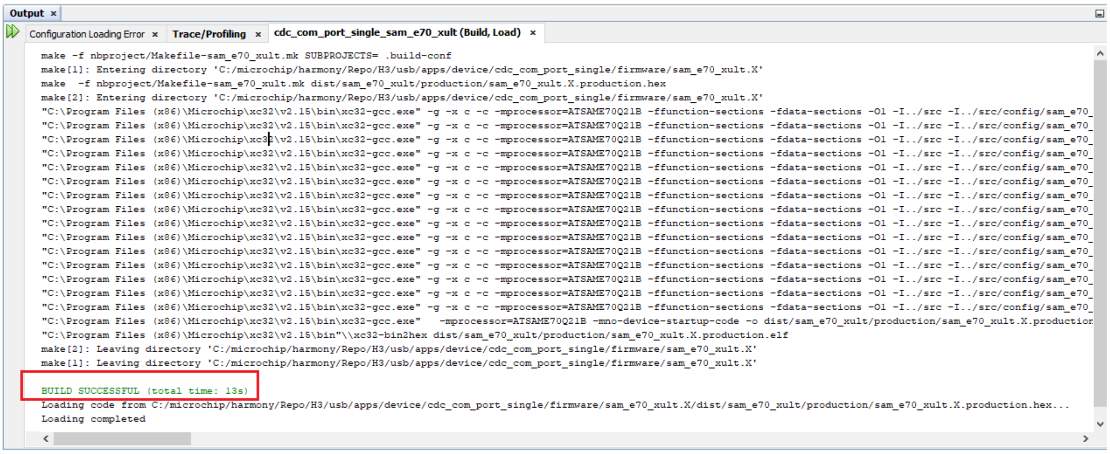
打开 File -> Project Properties



然后在 Configuration -> Compiler Toolchains -> XC32 中选择最新版本的 XC32。点击 Apply，然后 OK。



编译完成后，在 Output 窗口可以看到编译结果的提示信息



(九) USB CDC 的相关功能代码

在 application 中，主要需要处理的是两个层次的 event。

1. 一个是 USB Device 相关的，比如 attach，detach 等 event。这个在应用启动的时候，打开 USB Device 设备。打开成功之后，调用驱动 API 注册 USB Device 消息处理的回调函数。

```
case APP_STATE_INIT:

    /* Open the device layer */
    appData.deviceHandle = USB_DEVICE_Open( USB_DEVICE_INDEX_0, DRV_IO_INTENT_READWRITE );

    if(appData.deviceHandle != USB_DEVICE_HANDLE_INVALID)
    {
        /* Register a callback with device layer to get event notification (for end point 0) */
        USB_DEVICE_EventHandlerSet(appData.deviceHandle, APP_USBDeviceEventHandler, 0);

        appData.state = APP_STATE_WAIT_FOR_CONFIGURATION;
    }
```

2. 另外一个就是 USB CDC Device 相关的，比如 set line coding，read，write 等 event。这个一般是在 USB Device 设备成功的配置之后，调用驱动 API 注册 USB CDC Device 消息处理的回调函数。

```
case USB_DEVICE_EVENT_CONFIGURED:

    /* Check the configuration. We only support configuration 1 */
    configuredEventData = (USB_DEVICE_EVENT_DATA_CONFIGURED*)eventData;

    if ( configuredEventData->configurationValue == 1)
    {
        /* Update LED to show configured state */
        LED_On();

        /* Register the CDC Device application event handler here.
         * Note how the appData object pointer is passed as the
         * user data */

        USB_DEVICE_CDC_EventHandlerSet(USB_DEVICE_CDC_INDEX_0, APP_USBDeviceCDCEventHandler, (uintptr_t)&appData);

        /* Mark that the device is now configured */
        appData.isConfigured = true;
    }
```

具体的应用代码课参考：
<Harmony 3 Framework path>\usb\apps\device\cdc_com_port_single

四、 总结

本文通过 MPLAB X IDE 的 MHC 一步步的配置，最后生成代码，完成了一个 USB CDC Device 的简单应用。基本的应用框架以及所有的底层驱动代码都是系统自动生成了。用户只在 USB Device 和 USB CDC Device 的事件回调处理中，处理相应的 event 就可以了。