

MPLAB® Harmony 3 之基础篇 (04)

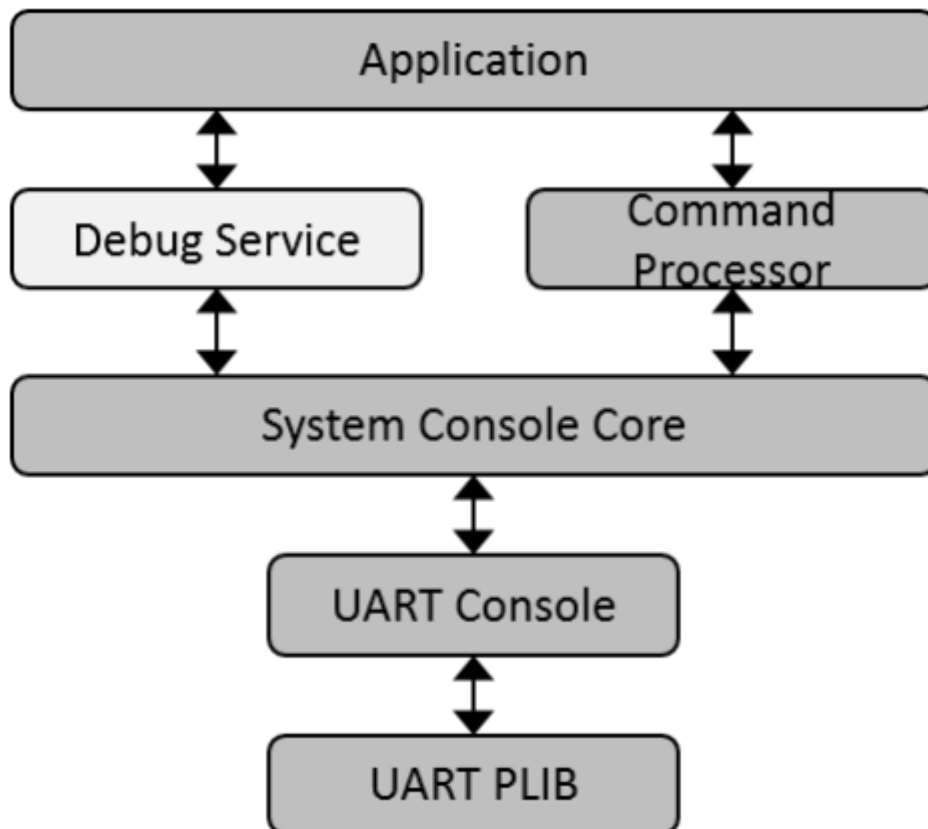
— 如何使用调试信息系统服务

Microchip Technology Inc.
MCU32 产品部

一、 简介

嵌入式工程师经常会通过 UART 打印一些调试信息，Harmony 3 提供了标准的调试系统服务功能，支持调试信息的打印。这样，用户不需要再创建自己的打印接口等，节省开发时间。

调试系统服务 (Debug System Service) 的消息是发给控制台系统服务 (Console System Service) 的， Console 系统服务可以中转给 UART。



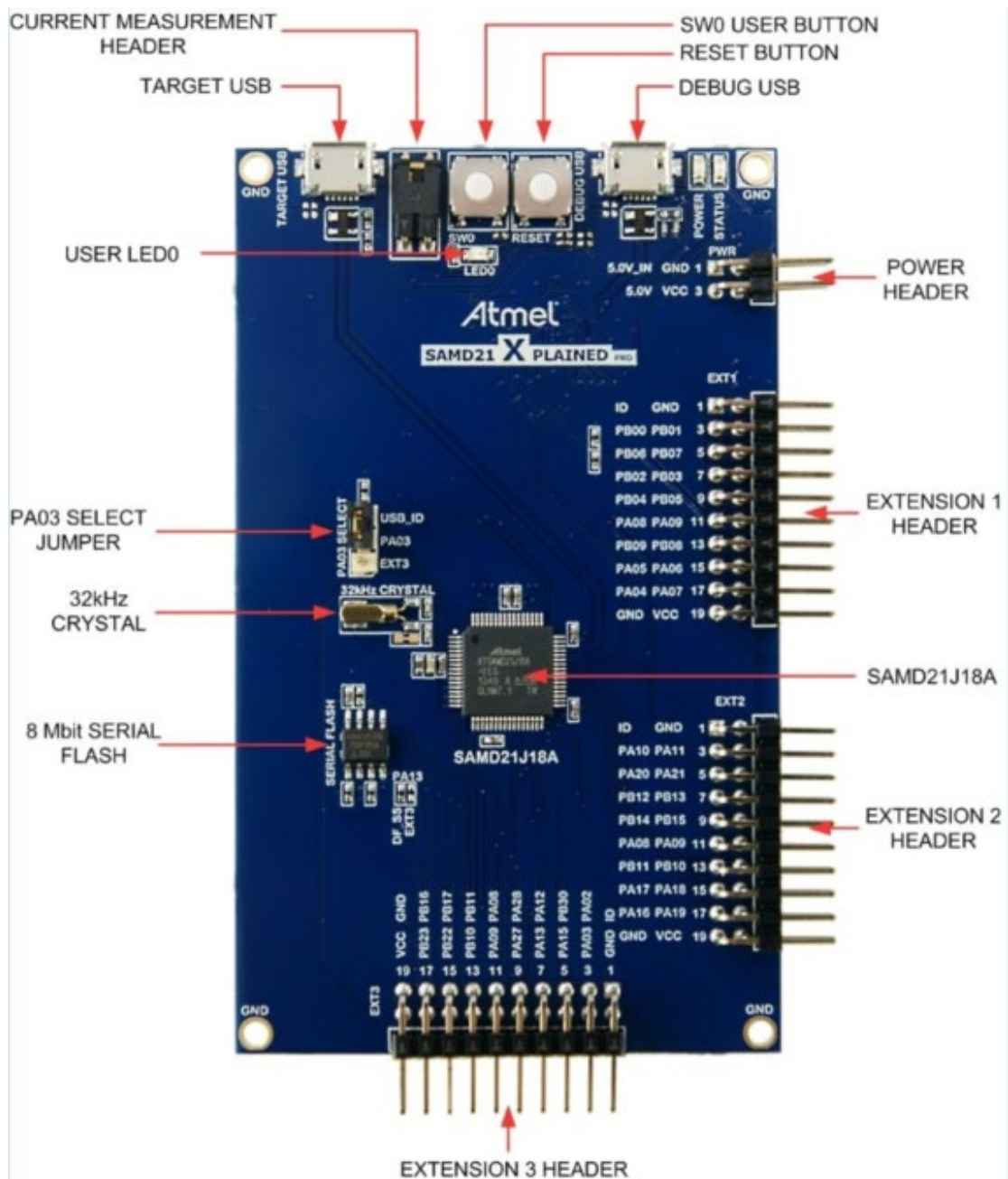
Debug Service Software Abstraction Block Diagram

本文利用 MPLAB Harmony 3 Configurator(MHC)一步步的轻松配置并生成一个带有调试信息打印的工程，例程中采用大家熟悉的 UART 作为调试信息打印的物理

接口。

二、 硬件工具和软件平台

硬件: [SAM D21 Xplained Pro](#)



软件:

MPLAB® X IDE: v5.10 或者更新

XC32: v2.15 或者更新

Harmony 3: v3.10 或者更新

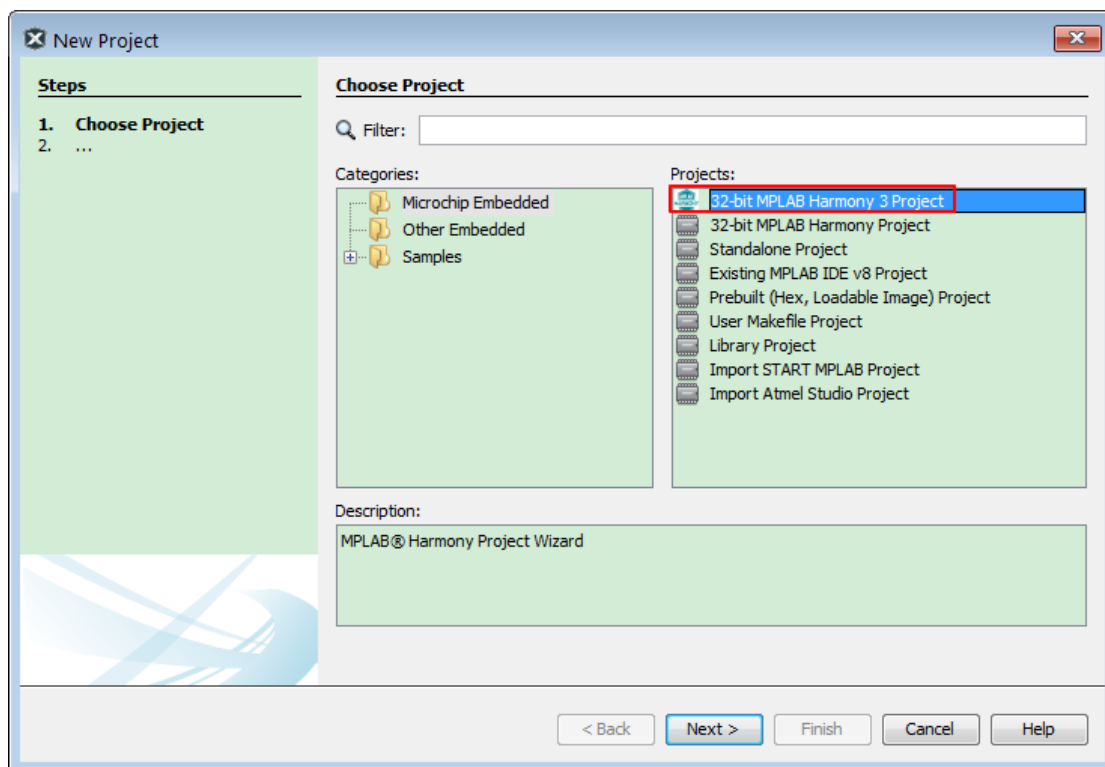
三、 详细步骤

接下来我们就可以用 MHC 一步步的进行配置和创建调试信息打印项目。

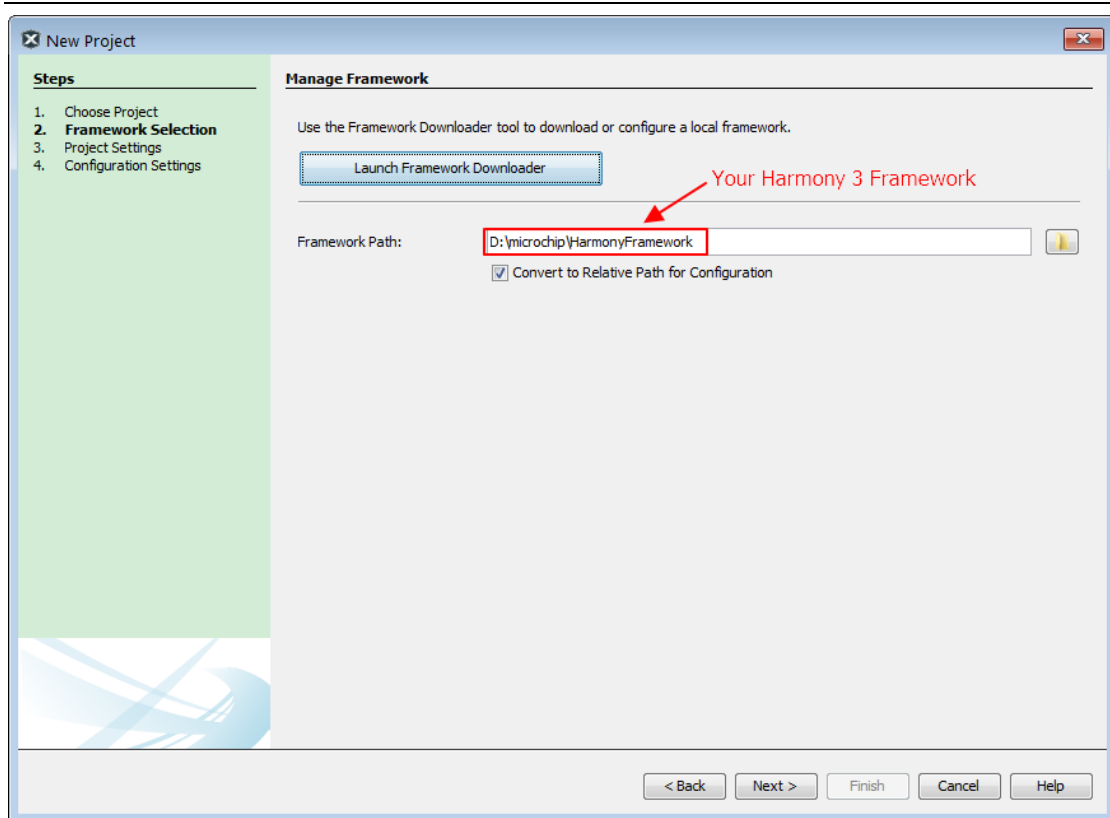
注：以下MHC 配置里没有特别标注出来的地方，说明使用的是默认选项。

(一) 在 MPLAB X IDE 里新建一个 Harmony 3 项目

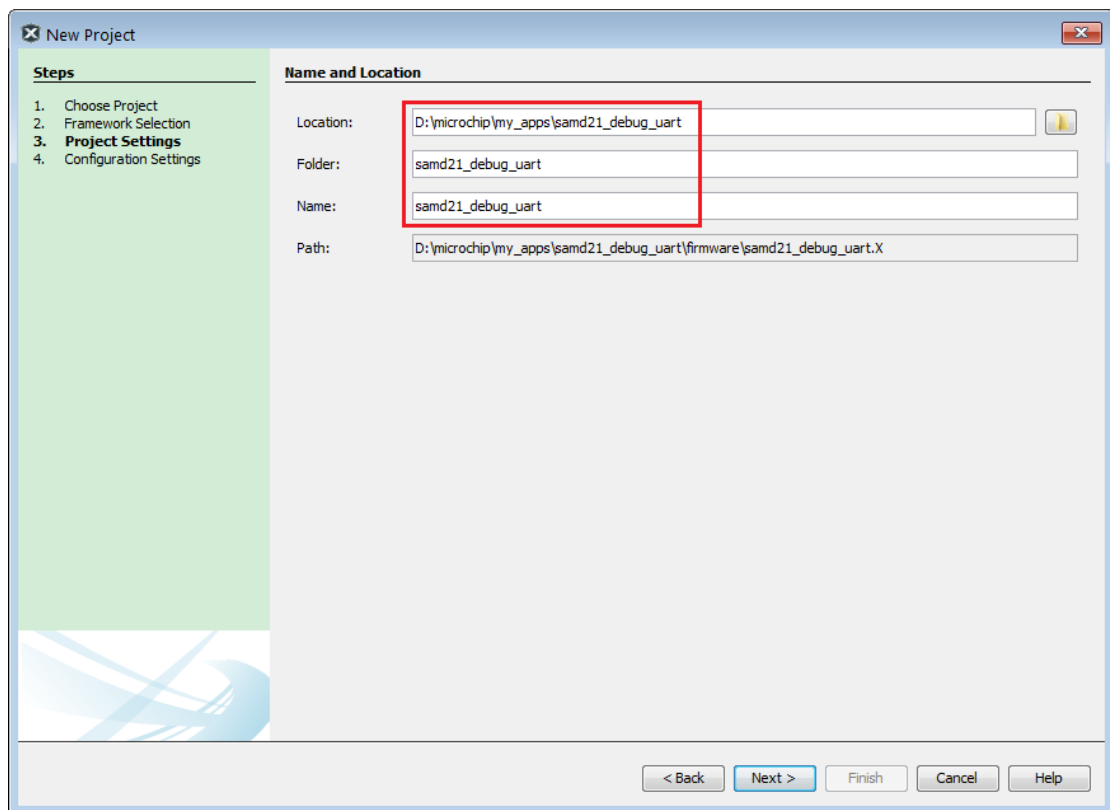
在 MPLAB X IDE 里点击 File > New Project:



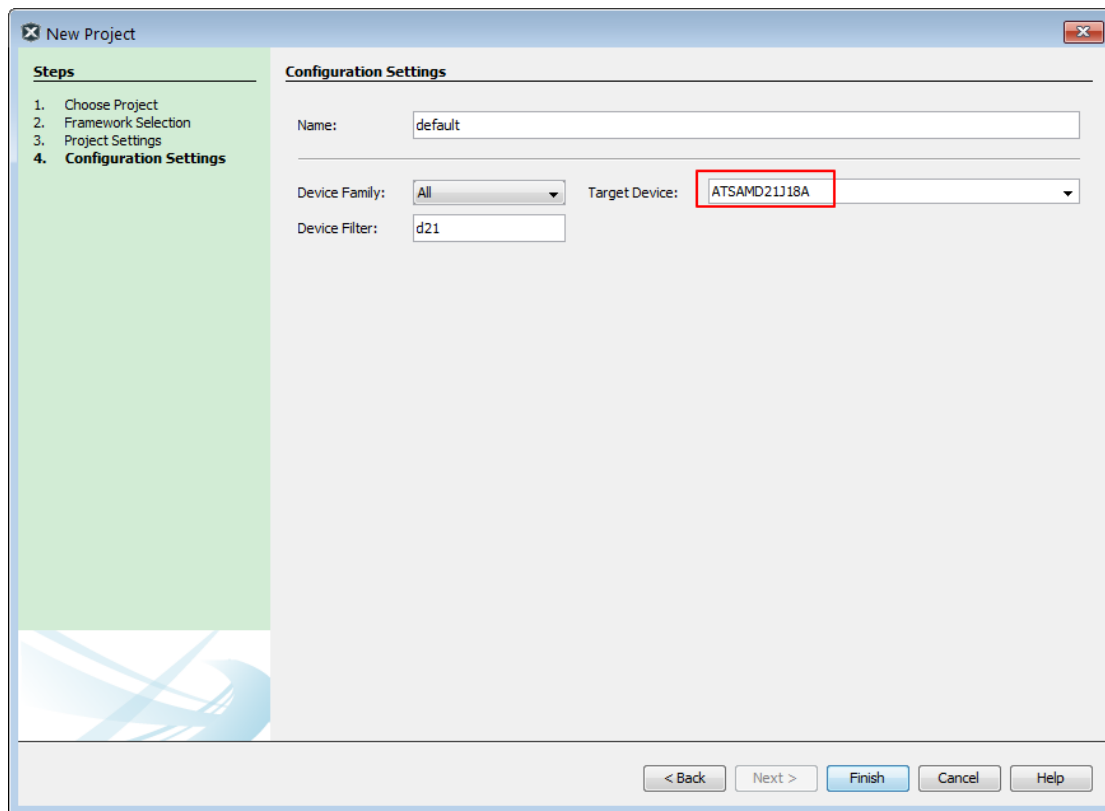
选择“32-bit MPLAB Harmony Project”，然后点击“Next”按钮。



选择“Harmony Framework”路径，然后点击“Next”按钮。



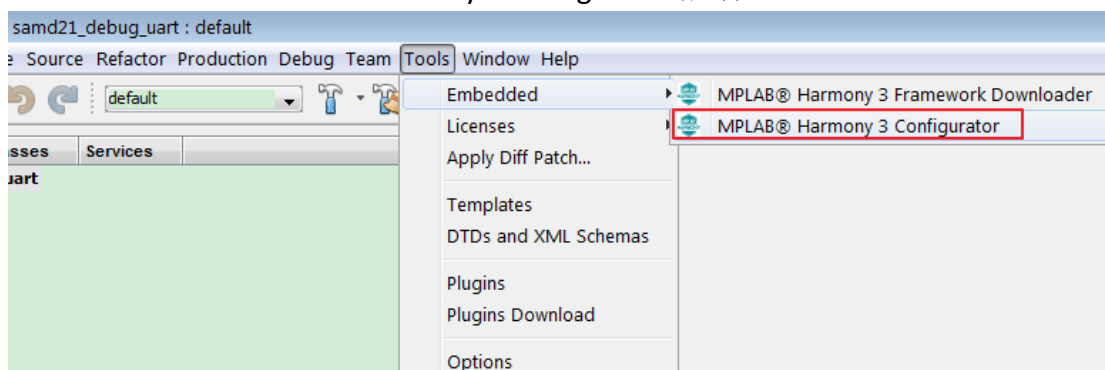
填写项目名称，本示例使用“samd21_debug_uart”，然后点击“Next”按钮。

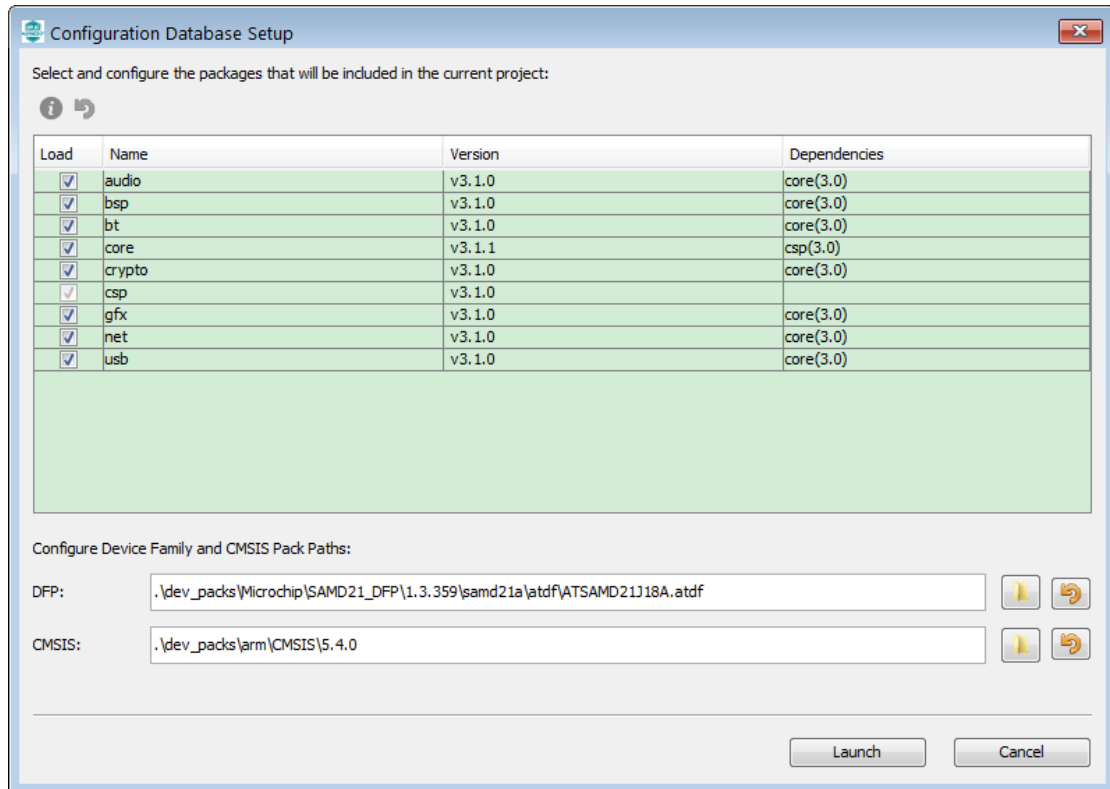


选择芯片类型“ATSAMD21J18A”，最后点击“Finish”按钮启动 MHC 配置界面。

(二) 启动 MHC

第一次创建项目时，MHC 配置界面会自动启动。或者手动在 MPLAB X IDE 里点击 Tools > Embedded > MPLAB Harmony 3 Configurator 启动：

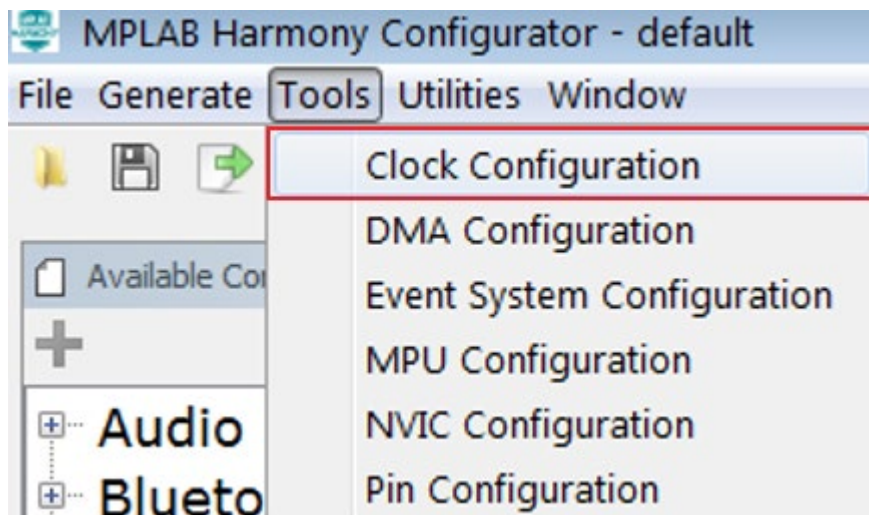




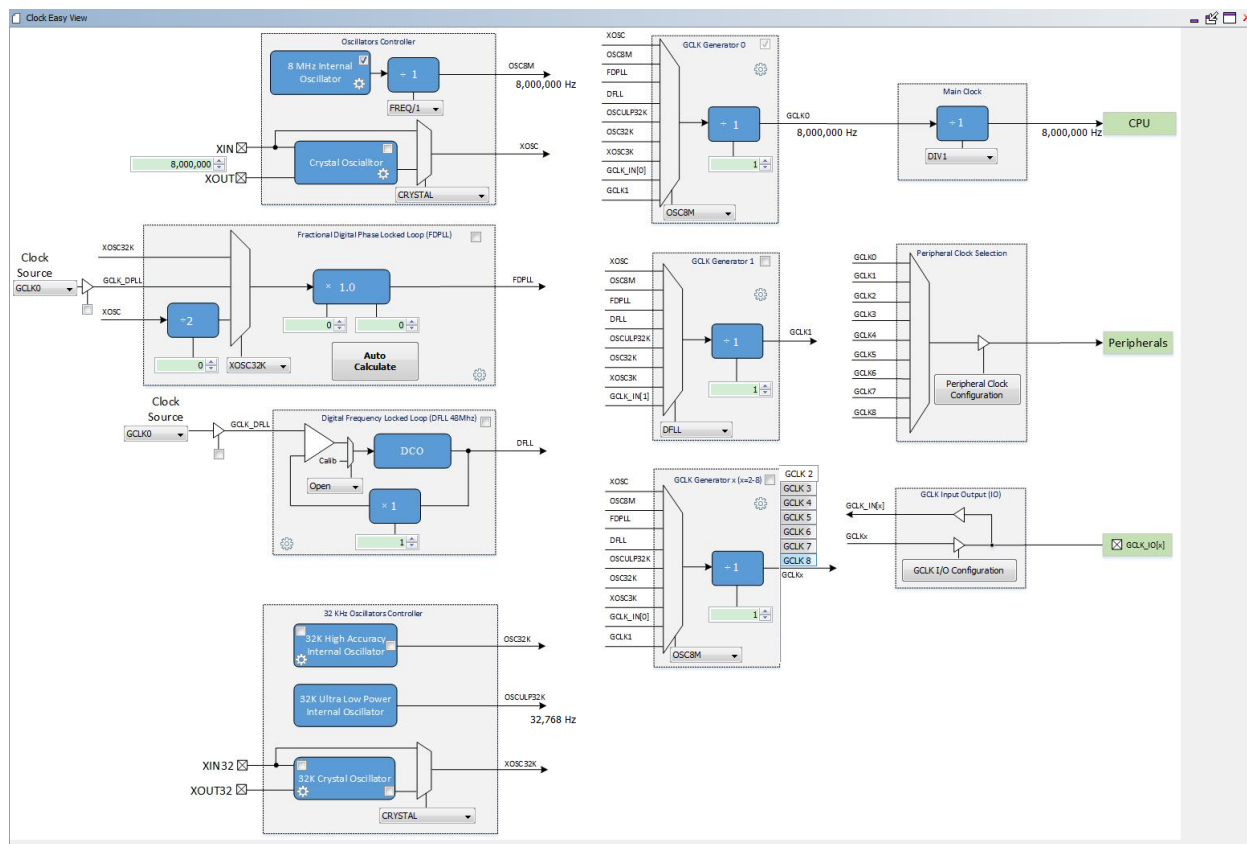
点击“Launch”按钮启动 MHC。

(三) 使用 MHC, 配置时钟

启动时钟配置界面：



使用默认配置即可：

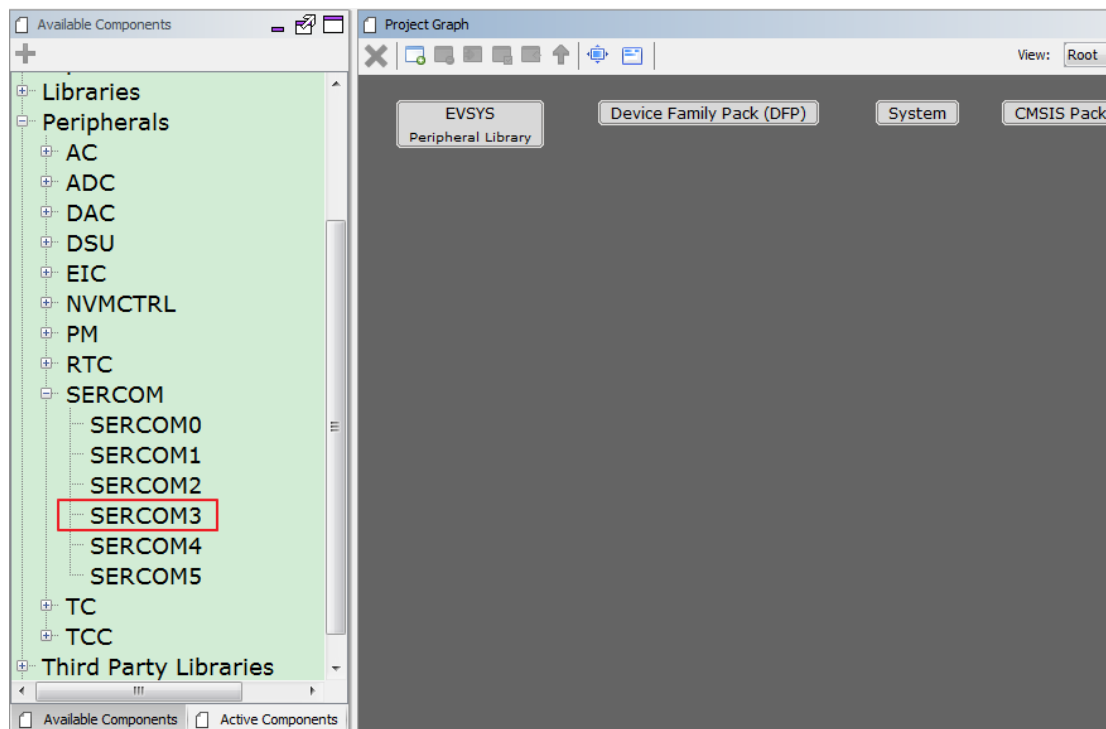


(四) 使用 MHC, 配置 UART 外设和引脚

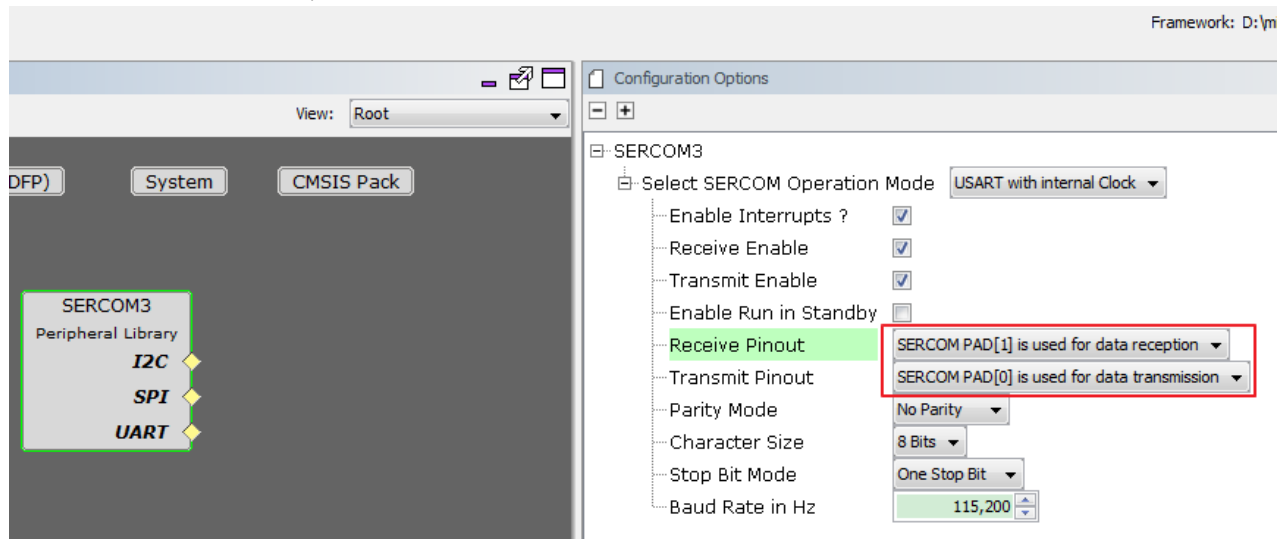
在 SAM D21 Xplained Pro 开发板上，有一个 EDBG 接口，它可以用于调试程序，并且包含一个 USB 虚拟串口。这个虚拟串口接到了 SERCOM3 的 UART 上：

Pin on SAM D21	Function
PA22	SERCOM3 PAD[0] UART TXD (SAM D21 TX line)
PA23	SERCOM3 PAD[1] UART RXD (SAM D21 RX line)

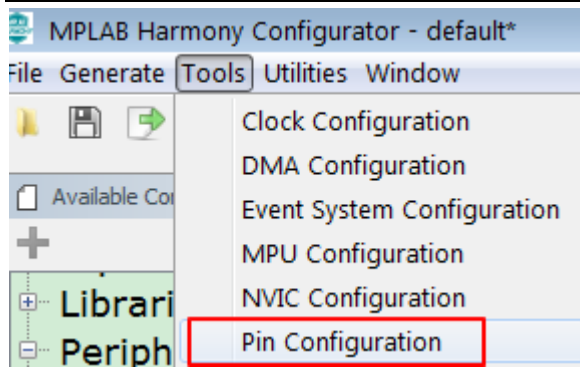
1. 在 Available Component 列表里选择 SERCOM3 外设，双击加入工程



2. 配置 SERCOM3 参数



3. 配置 SERCOM3 的引脚
启动引脚配置界面：



配置 PA22, PA23 作为 SERCOM3 的 PADx 引脚:

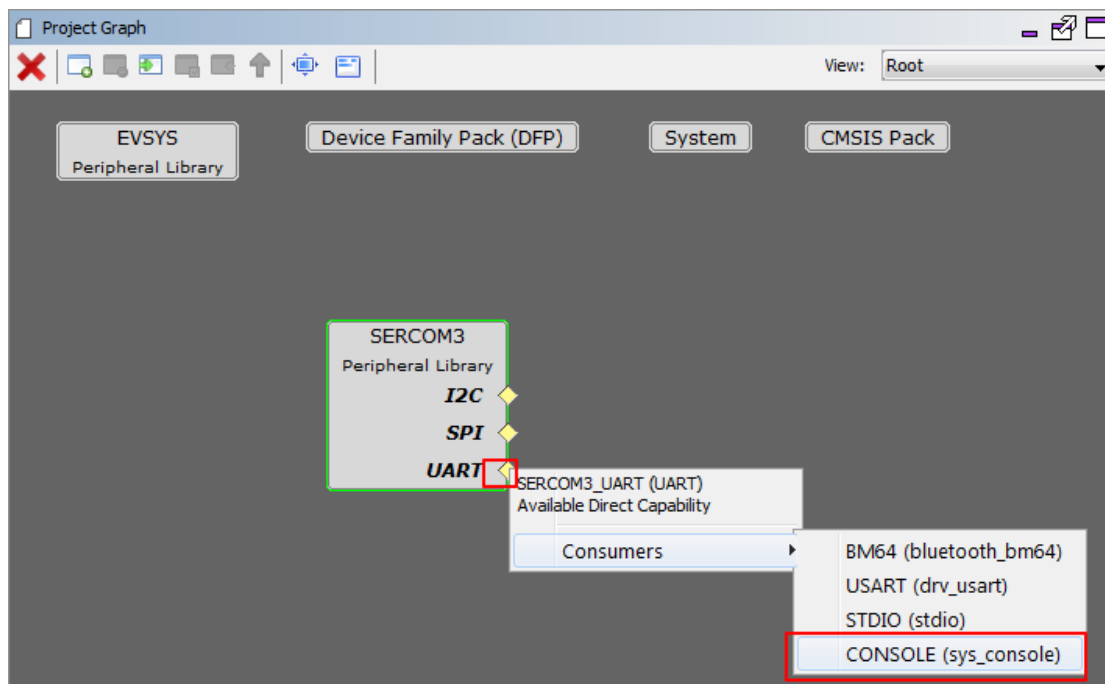
Pin Settings

Order: Ports Table View Easy View

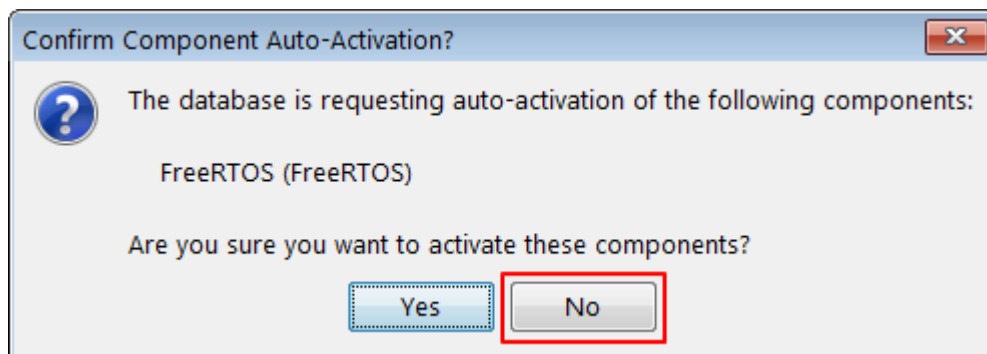
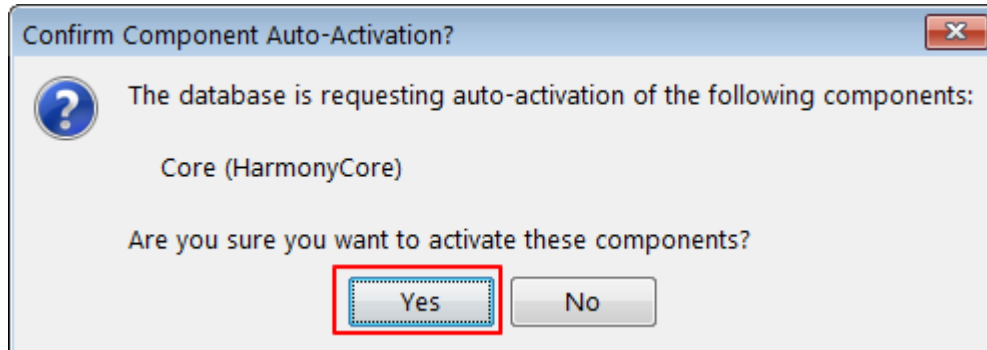
Pin Number	Pin ID	Custom Name	Function	Mode
1	PA00		Available	Digital
2	PA01		Available	Digital
3	PA02		Available	Digital
4	PA03		Available	Digital
13	PA04		Available	Digital
14	PA05		Available	Digital
15	PA06		Available	Digital
16	PA07		Available	Digital
17	PA08		Available	Digital
18	PA09		Available	Digital
19	PA10		Available	Digital
20	PA11		Available	Digital
29	PA12		Available	Digital
30	PA13		Available	Digital
31	PA14		Available	Digital
32	PA15		Available	Digital
35	PA16		Available	Digital
36	PA17		Available	Digital
37	PA18		Available	Digital
38	PA19		Available	Digital
41	PA20		Available	Digital
42	PA21		Available	Digital
43	PA22	SERCOM3_PAD0	SERCOM3_PAD0	Digital
44	PA23	SERCOM3_PAD1	SERCOM3_PAD1	Digital

(五) 使用 MHC, 配置调试系统服务

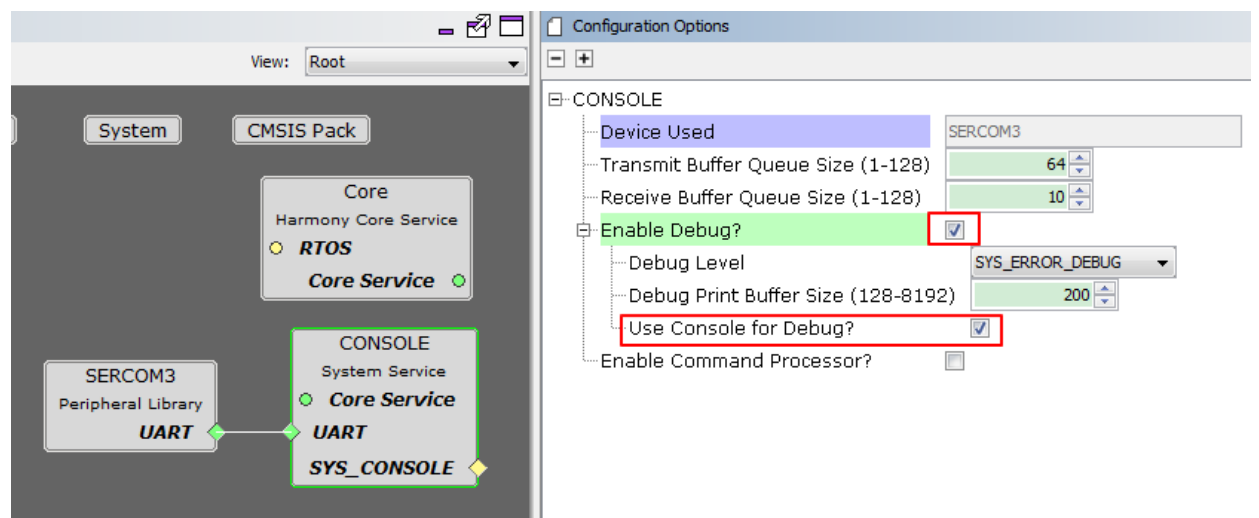
1. 鼠标右键点击 SERCOM3 右下方黄色的菱形, 然后选择 CONSOLE 组件



注：选择 *CONSOLE* 后，如提示添加关联组件“*Harmony Core Service*”，添加即可。如提示添加组件“*FreeRTOS*”，选择不添加，因为本例程中不使用 RTOS。



2. 配置 CONSOLE 组件，使能 Debug 功能



The image shows a screenshot of the Microchip IDE configuration interface. On the left, a system tree view displays components under 'System' and 'CMSIS Pack'. The 'CONSOLE' component is highlighted with a green border. It is categorized as a 'System Service' and includes 'Core Service' (with a green dot), 'UART', and 'SYS_CONSOLE' (with a yellow diamond). A 'SERCOM3 Peripheral Library' is also shown with a 'UART' component. On the right, the 'Configuration Options' panel is open for the 'CONSOLE' component. The 'Device Used' is set to 'SERCOM3'. Other options include 'Transmit Buffer Queue Size (1-128)' set to 64, 'Receive Buffer Queue Size (1-128)' set to 10, 'Debug Level' set to 'SYS_ERROR_DEBUG', and 'Debug Print Buffer Size (128-8192)' set to 200. Two options are highlighted with red boxes: 'Enable Debug?' (checked) and 'Use Console for Debug?' (checked). 'Enable Command Processor?' is unchecked.

View: Root

System CMSIS Pack

Core
Harmony Core Service
RTOS
Core Service

SERCOM3
Peripheral Library
UART

CONSOLE
System Service
Core Service
UART
SYS_CONSOLE

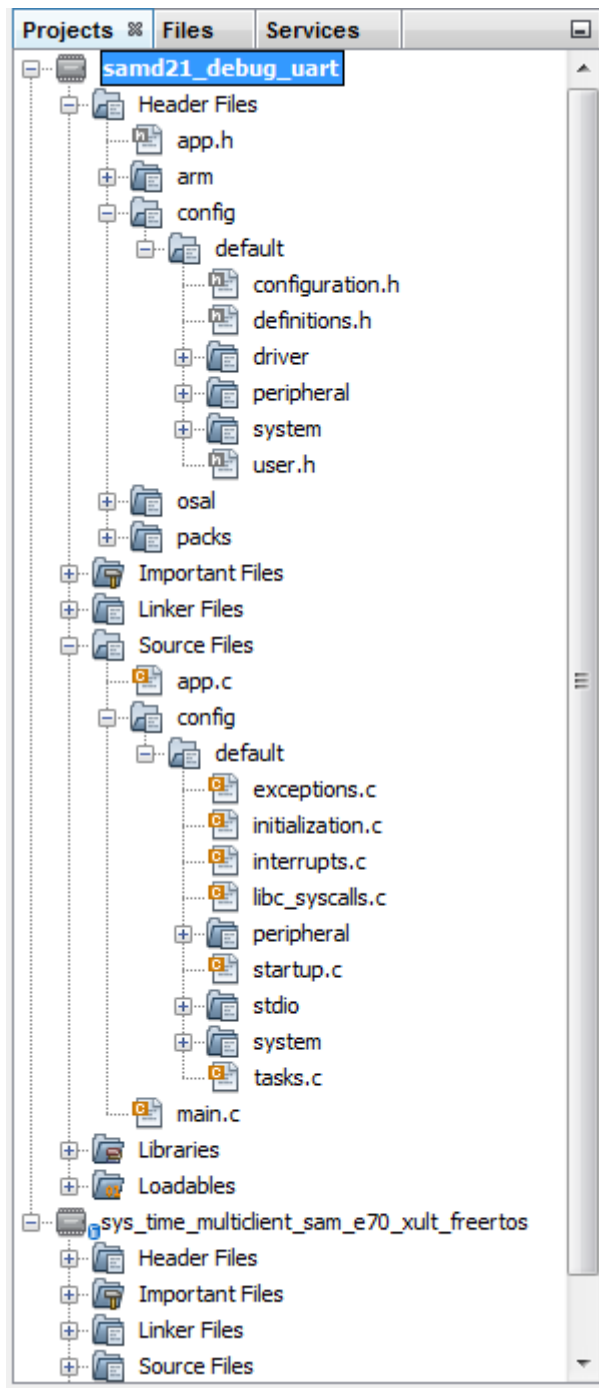
Configuration Options

CONSOLE

- Device Used: SERCOM3
- Transmit Buffer Queue Size (1-128): 64
- Receive Buffer Queue Size (1-128): 10
- Enable Debug?
- Debug Level: SYS_ERROR_DEBUG
- Debug Print Buffer Size (128-8192): 200
- Use Console for Debug?
- Enable Command Processor?

(六) 使用 MHC, 生成代码

保存并生成代码：



(七) 在 app.c 里增加如下测试代码

1. 包含相关头文件

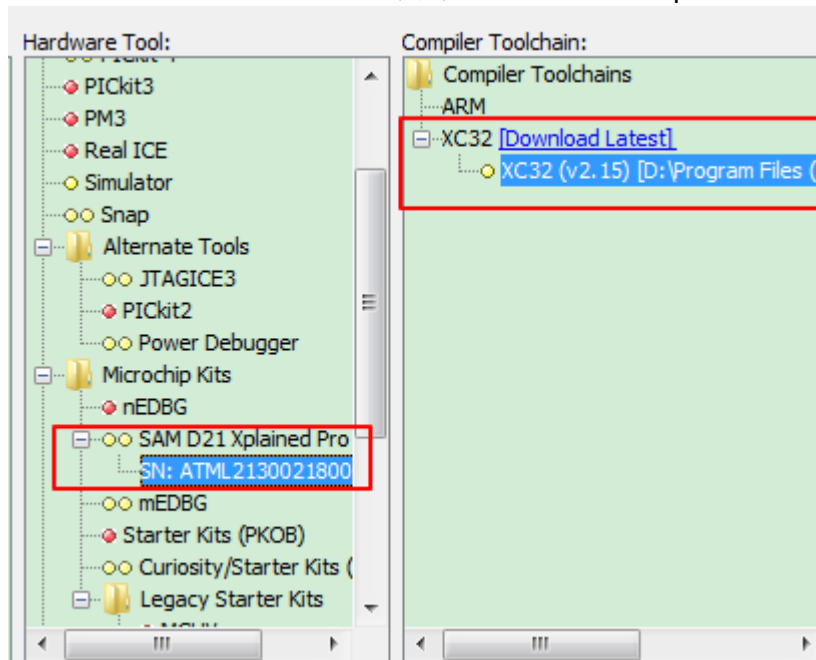
```
#include "app.h"  
#include "definitions.h" // SYS function prototypes
```

2. 添加打印代码

```
void APP_Initialize ( void )  
{  
    /* Place the App state machine in its initial state. */  
    appData.state = APP_STATE_INIT;  
  
    SYS_MESSAGE("My first Test Message\r\n");  
    SYS_PRINT("Test print: appData.state = %x\r\n", appData.state);  
}
```

(八) 编译下载测试

1. 用 Micro USB 线将 SAM D21 Xplained Pro 开发板连接到电脑
2. 在项目属性里选择 XC32 编译器和 SAM D21 Xplained Pro EDBG 调试接口

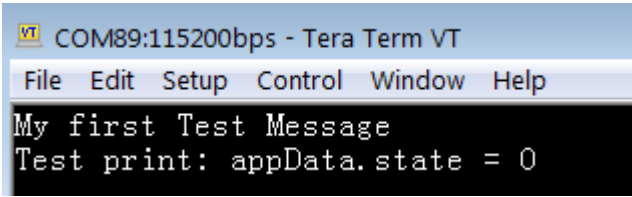


3. 编译下载

编译并下载程序：



在串口终端程序里，你将看到这样的打印信息：



```
COM89:115200bps - Tera Term VT
File Edit Setup Control Window Help
My first Test Message
Test print: appData.state = 0
```

四、 总结

本文通过 MHC 一步步配置和生成代码完成了一个调试信息打印的简单工程应用。如需详细了解调试系统服务的更多特点，请参考 [Harmony 3 core 的帮助文档](#)（章节：Debug System Service Library）。