

MPLAB® Harmony 3 之基础篇（03）

— 如何使用 Harmony 3 PLIB 外设驱动开发应用程序

Microchip Technology Inc.
MCU32 产品部

一、 简介

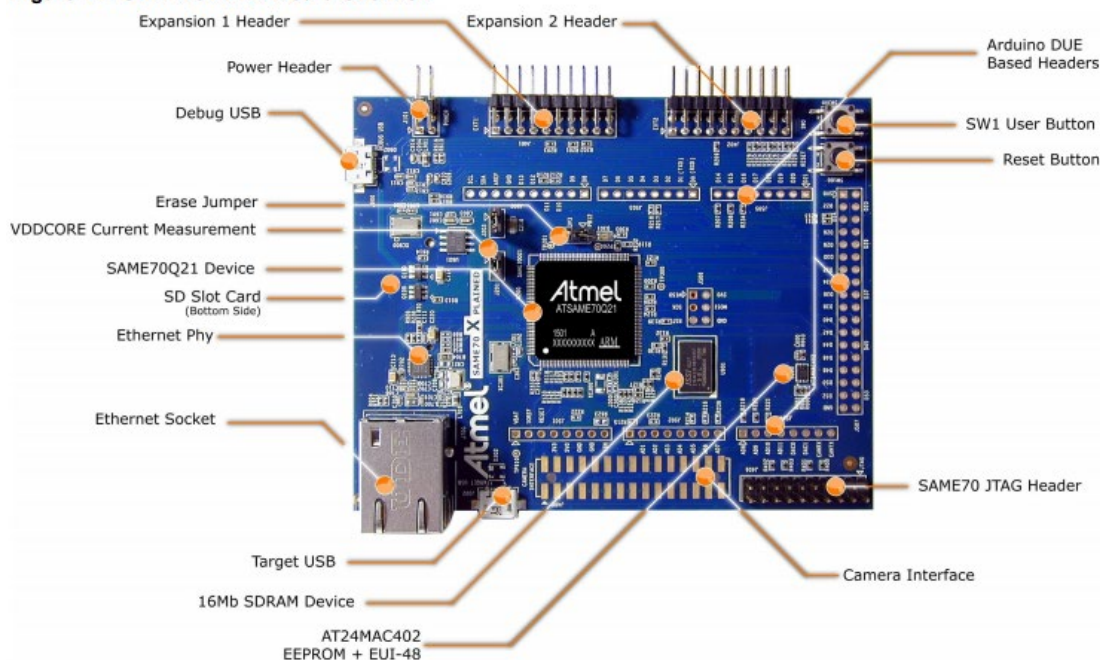
本文主要介绍如何利用 MPLAB X IDE 创建一个工程，利用 MPLAB Harmony 3 Configurator(MHC)添加 USART 外设驱动到工程文件，并利用 MHC 的配置工具 (CLOCK, PIN 等)完成 USART 外设的配置。添加 APP 到工程文件，调用 USART PLIB API 完成串口输入和输出的功能。

二、 硬件工具和软件平台

硬件： SAM E70 Xplained Board

http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-44050-Cortex-M7-Microcontroller-SAM-E70-XPLD-Xplained_User-guide.pdf

Figure 1-2 SAME70-XPLD Board Overview



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® Harmony 3 之基础篇（02） -- 了解 MHC”

MPLAB X IDE: v5.10 或者更新

XC32: v2.10 或者更新

Harmony 3: v3.10 或者更新

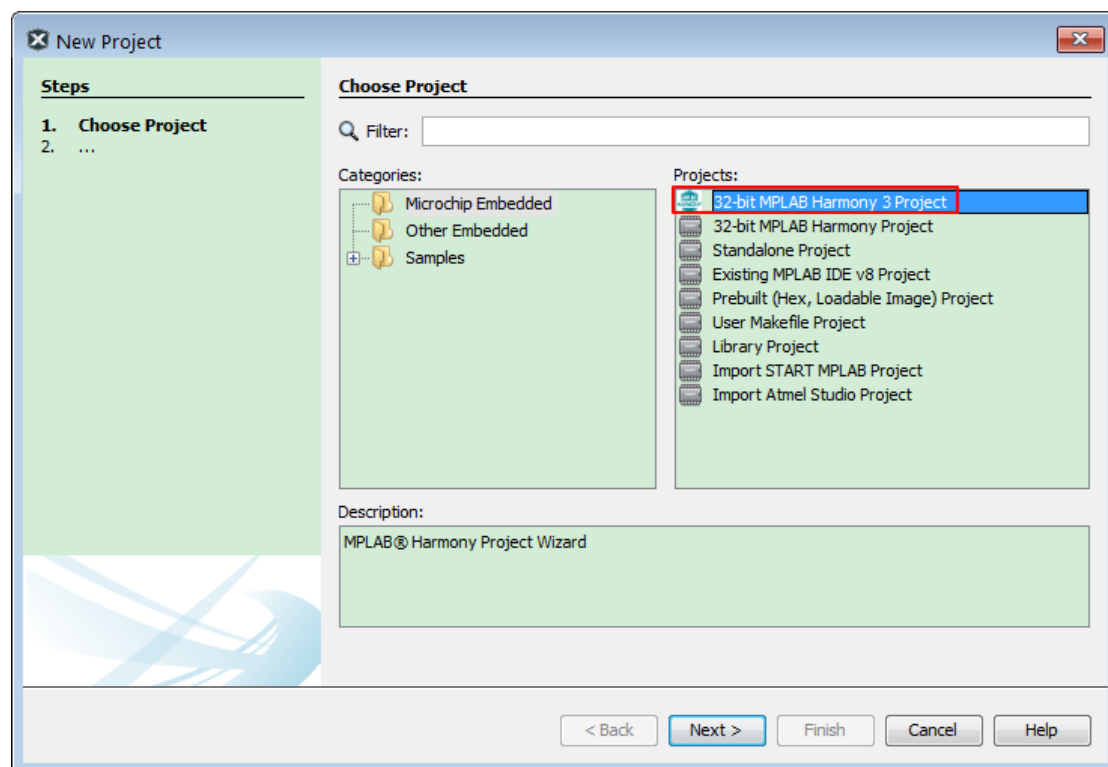
三、 详细步骤

接下来我们就可以用 MPLAB X IDE 和 MHC 一步步地创建和配置 USART 外设驱动的程序。

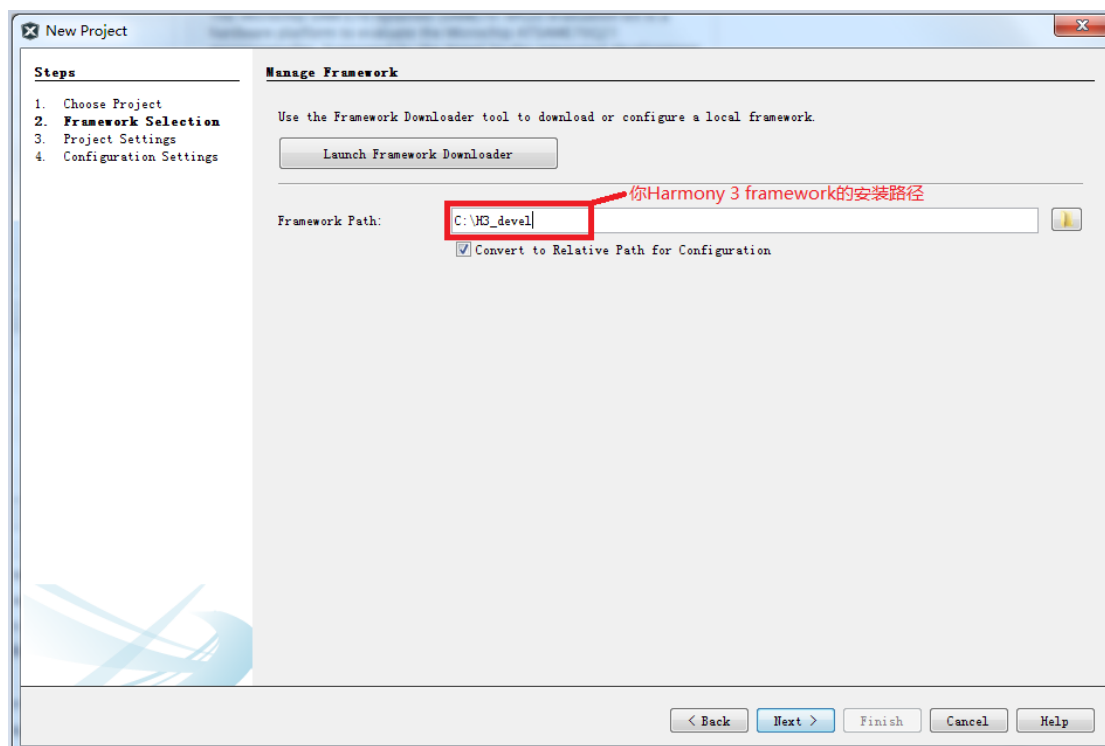
注：以下 MHC 配置里没有特别标注出来的地方，说明使用的是默认选项。

(一)在 MPLAB X 里新建一个 Harmony 3 项目

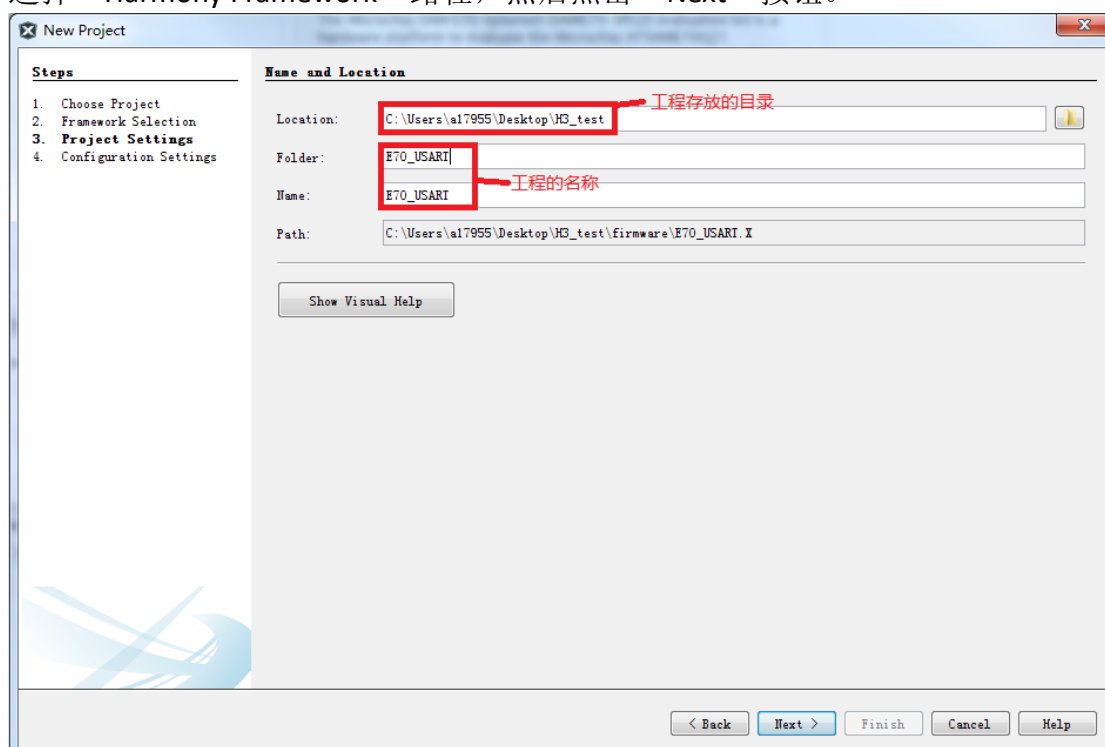
在 MPLAB X IDE 里点击 File > New Project:



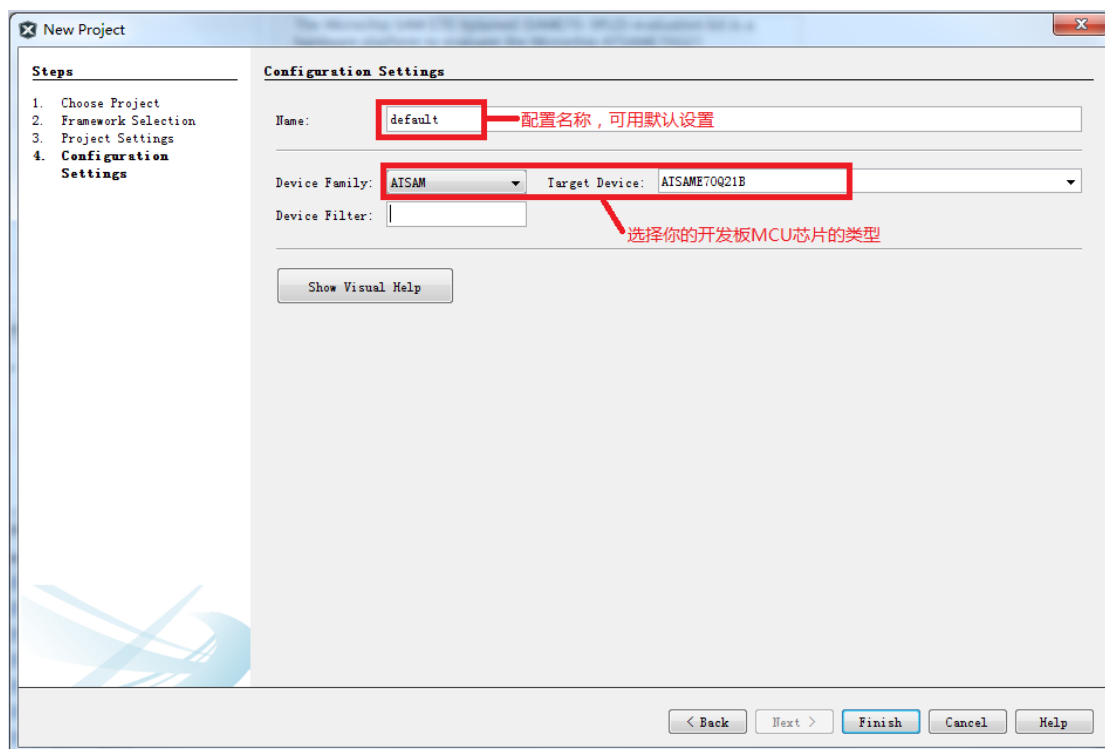
选择“32-bit MPLAB Harmony Project”，然后点击“Next”按钮。



选择“Harmony Framework”路径，然后点击“Next”按钮。



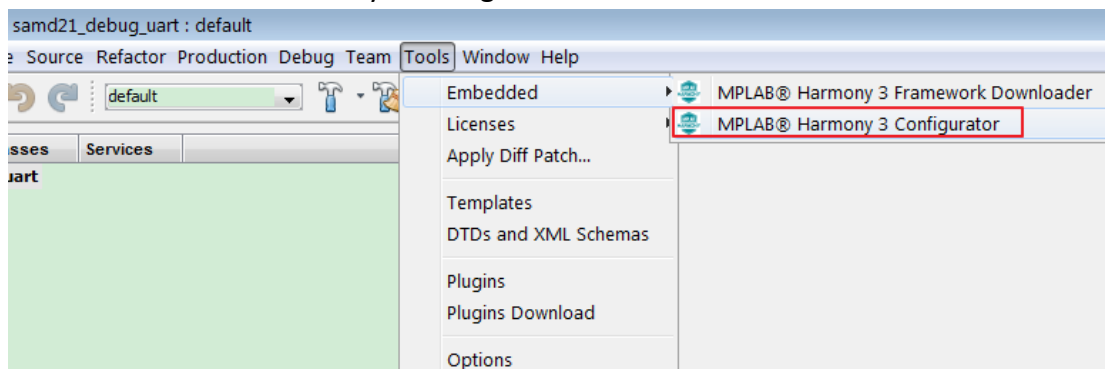
填写项目名称，本示例使用“E70_USART”，然后点击“Next”按钮。



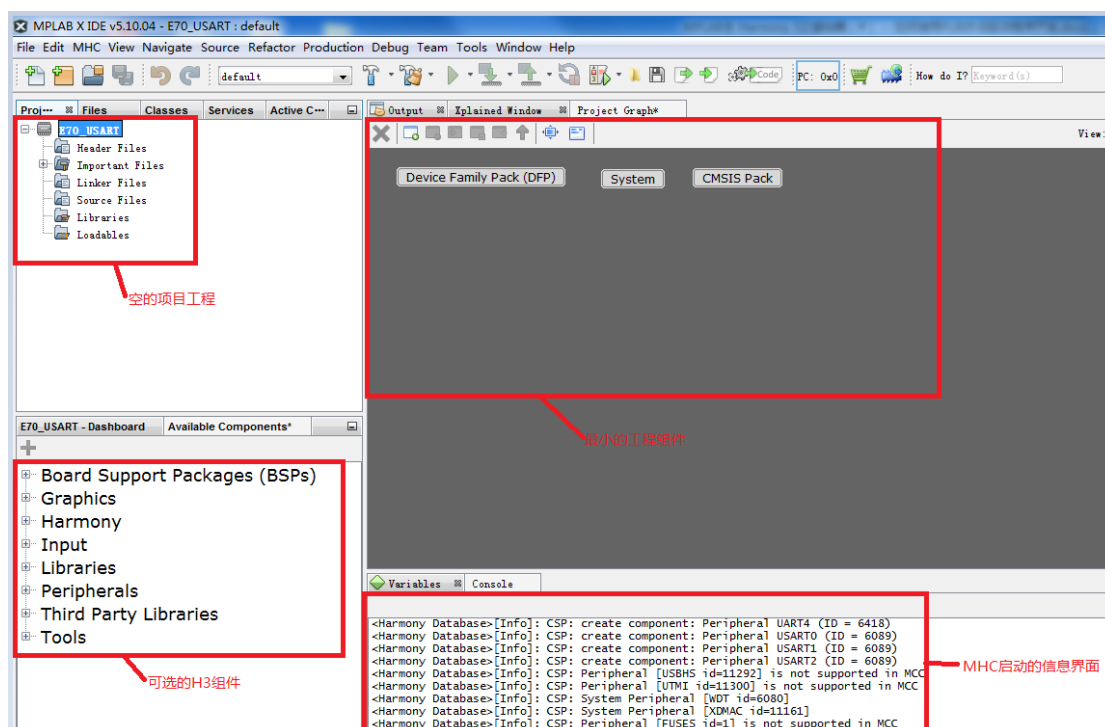
选择芯片类型“ATSAME70Q21B”，最后点击“Finish”按钮启动 MHC 配置界面。

(二) 启动 MHC

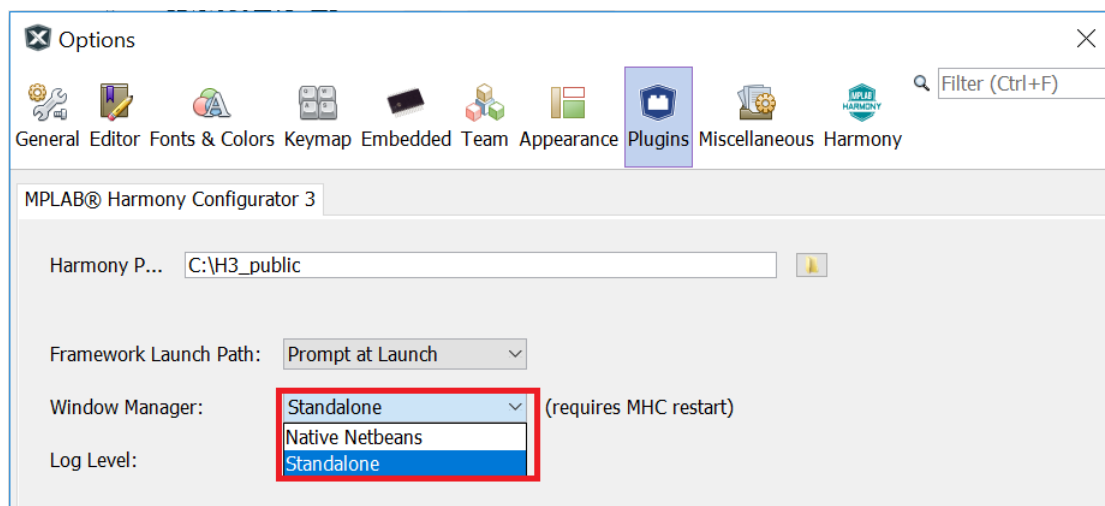
第一次创建项目时，MHC 配置界面会自动启动。或者手动在 MPLAB X 里点击 Tools > Embedded > MPLAB Harmony 3 Configurator 启动 MHC：



启动完成后的，主界面如下图：

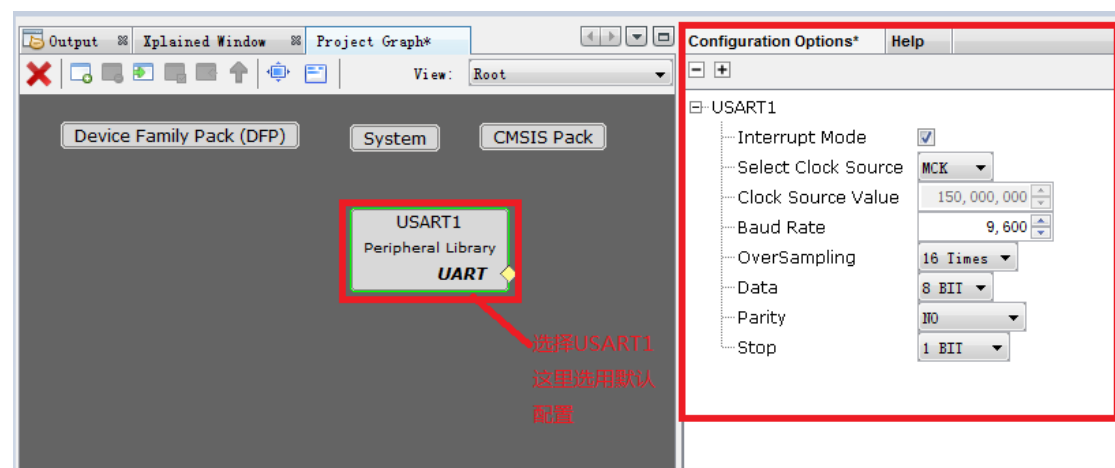
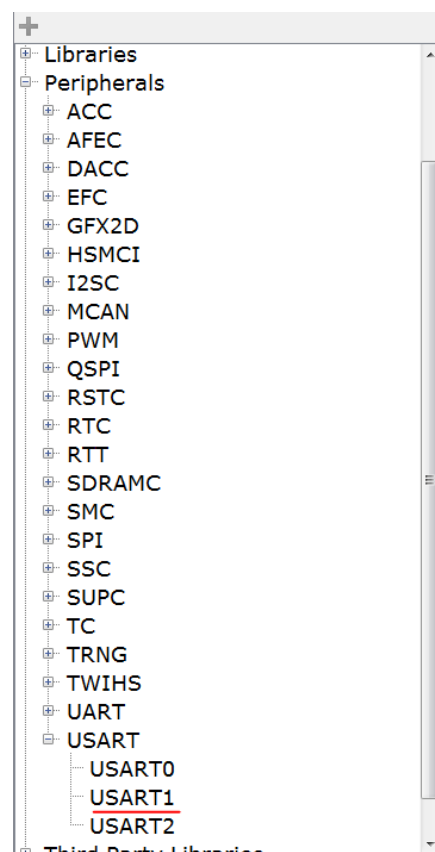


注：如果你看到的 MHC 菜单不是集成在主菜单的，可以通过 *Tools>>Options* 的 *Windows Manager* 来切换。这个不影响功能使用。

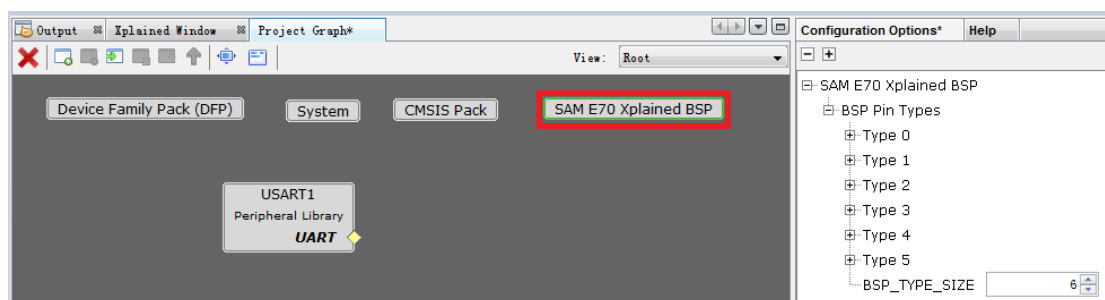
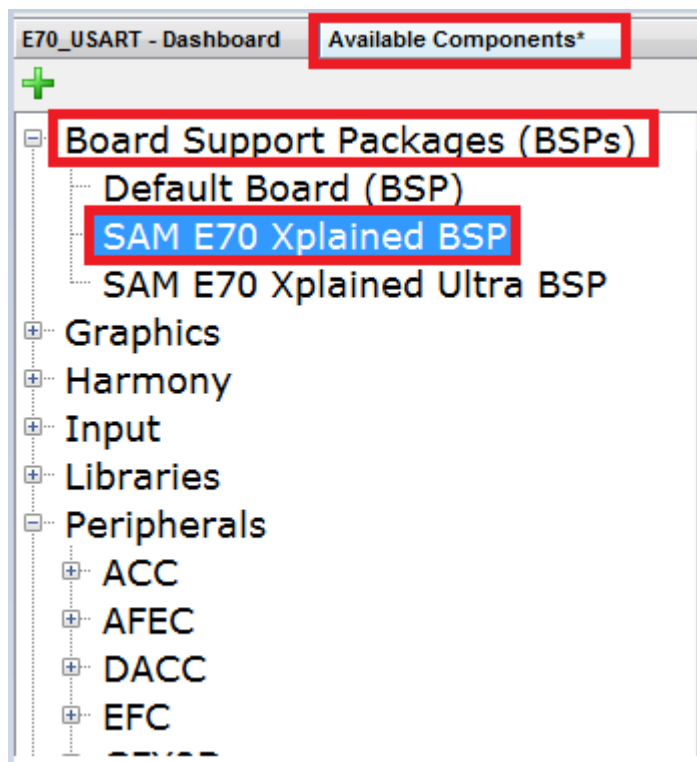


(三) 添加 USART1 外设驱动和 E70 Xplained Board (BSP) 支持

从 *Peripherals* 选择 *USART1* 外设驱动，双击，添加到 *Project Graph*，*USART* 采用默认的设置。

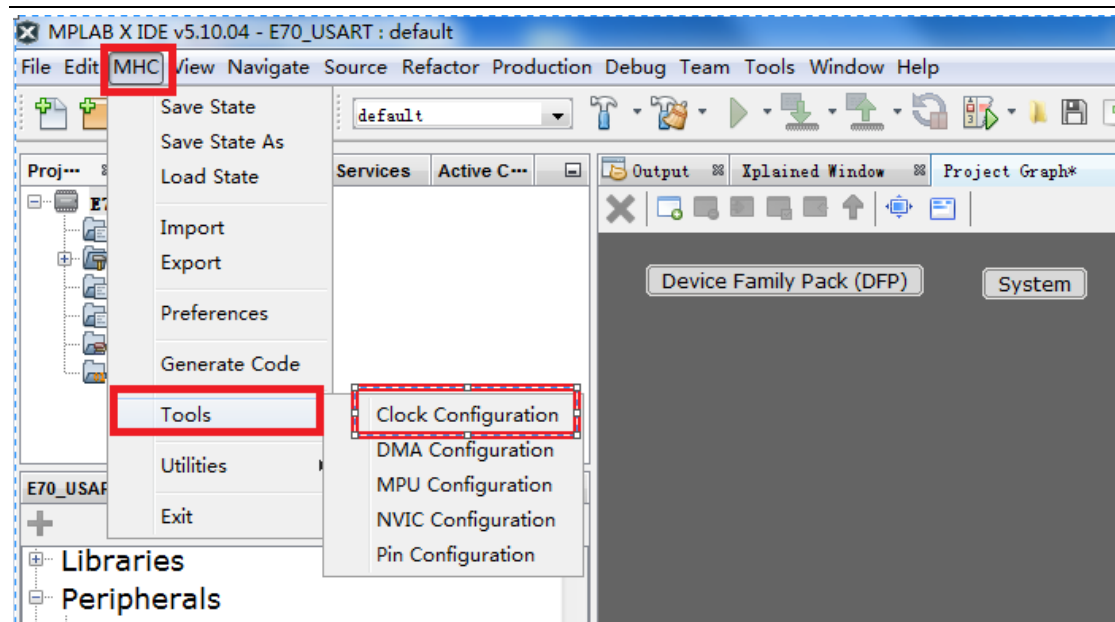


从 Board Support Packages 选择 E70 Xplained BSP 的支持，双击添加到 Project Graph.

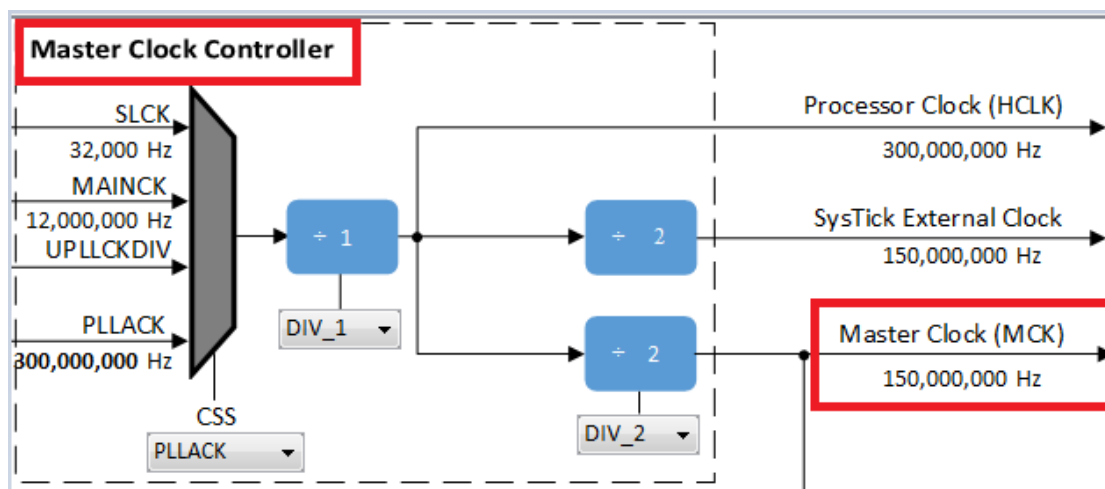
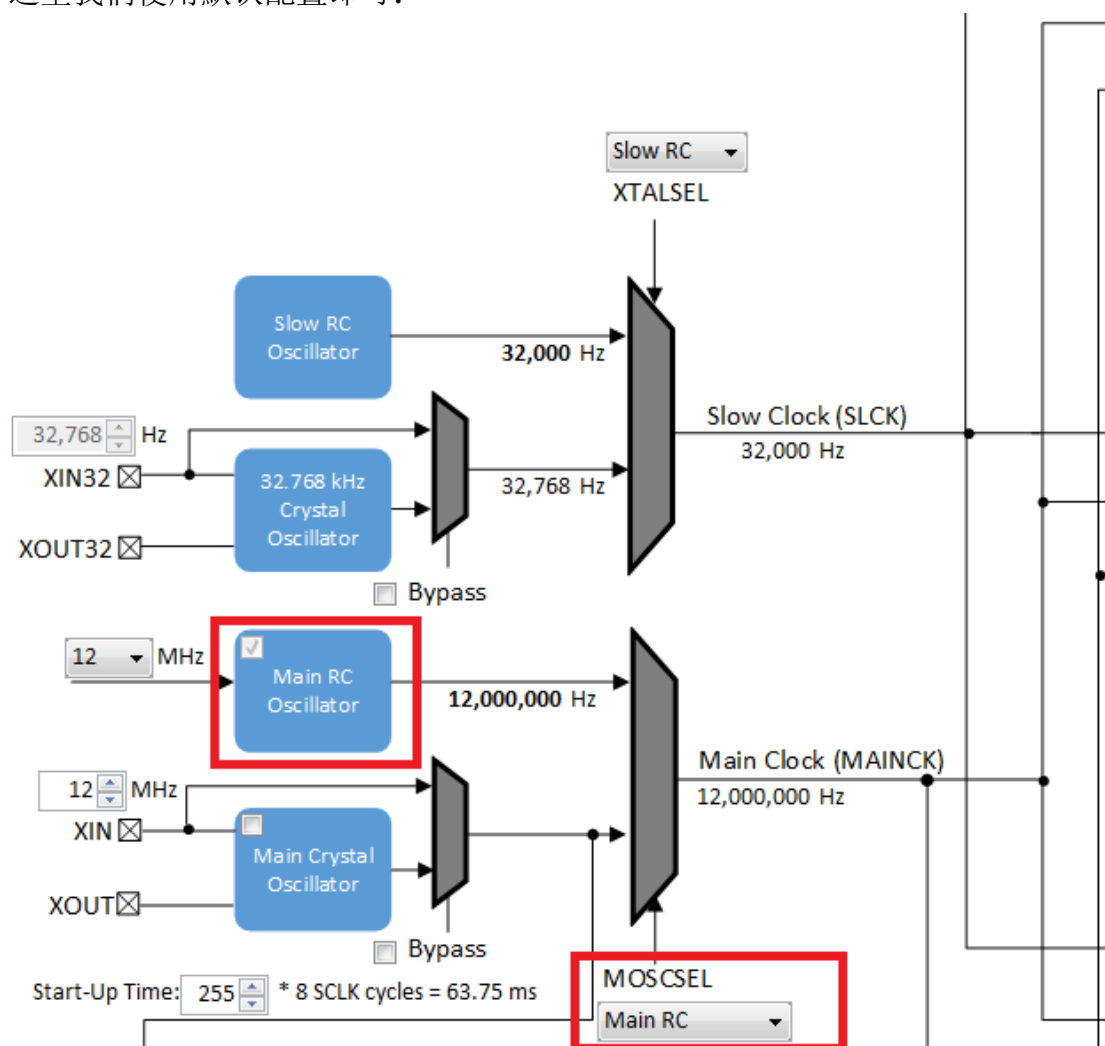


(四)选择 MHC>Tools>Clock Configuration 菜单，启动时钟配置

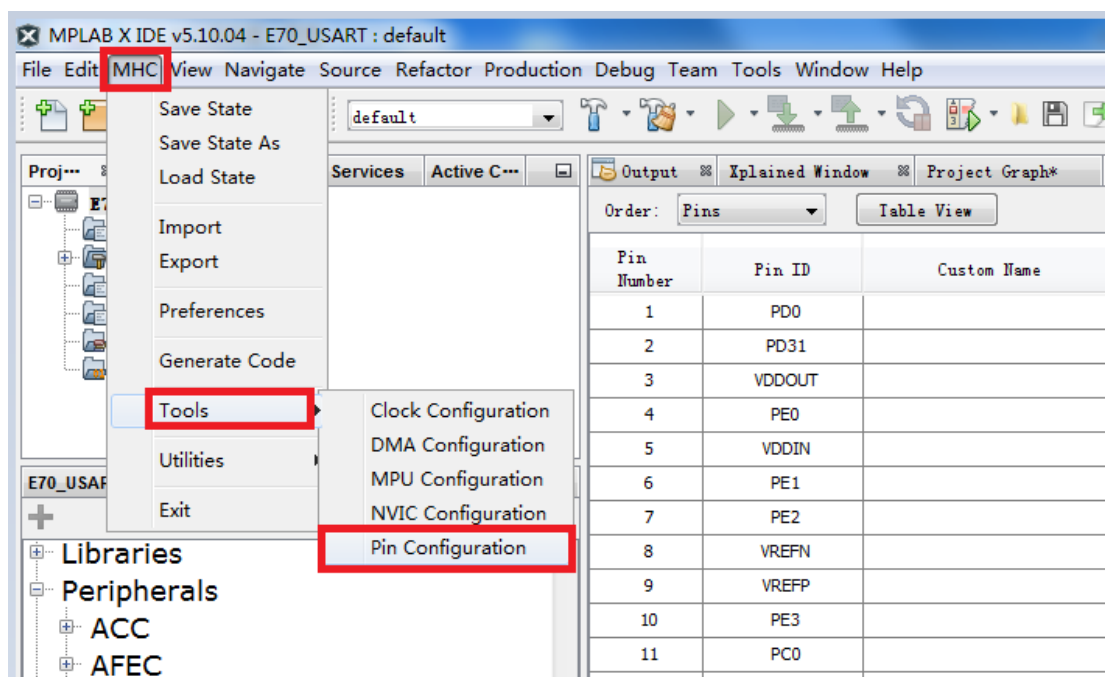
E70 的时钟配置界面，确认时钟源和 USART 的时钟是我们需要的配置



这里我们使用默认配置即可：



(五) 使用 MHC>Tools>Pin Configuration, 配置 USART 引脚分配

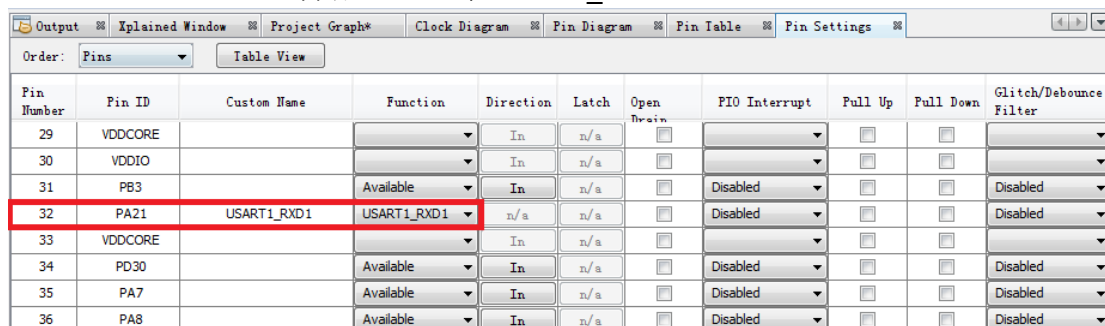


USART 的管脚分配可以参考以下 E70 Xplained Board User guide 文档
(Atmel-44050-Cortex-M7-Microcontroller-SAM-E70-XPLD-Xplained_User-guide.pdf)

Table 4-25 Virtual COM Port Connections

SAM E70 Pin	Function	Shared Functionality
PB4	TXD1 (SAM E70 UART TX Line)	EXT2 Header , J507 Header , Embedded Debugger
PA21	RXD1 (SAM E70 UART RX Line)	EXT2 Header , J507 Header , Embedded Debugger

E70 Xplained board 上分配 PA21 到 USART1_RXD1



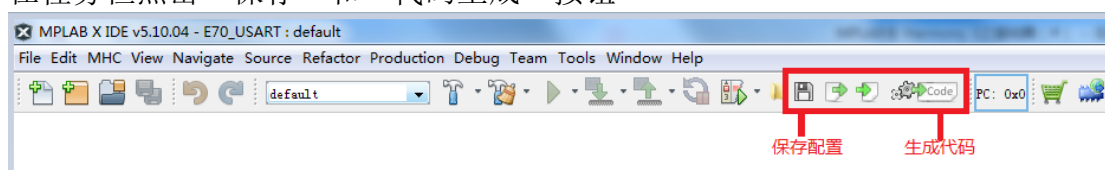
Pin Number	Pin ID	Custom Name	Function	Direction	Latch	Open Drain	PIO Interrupt	Pull Up	Pull Down	Glitch/Debounce Filter
29	VDDCORE			In	n/a	☐		☐	☐	
30	VDDIO			In	n/a	☐		☐	☐	
31	PB3		Available	In	n/a	☐	Disabled	☐	☐	Disabled
32	PA21	USART1_RXD1	USART1_RXD1	n/a	n/a	☐	Disabled	☐	☐	Disabled
33	VDDCORE			In	n/a	☐		☐	☐	
34	PD30		Available	In	n/a	☐	Disabled	☐	☐	Disabled
35	PA7		Available	In	n/a	☐	Disabled	☐	☐	Disabled
36	PA8		Available	In	n/a	☐	Disabled	☐	☐	Disabled

E70 Xplained board 上分配 PB4 到 USART1_TXD1

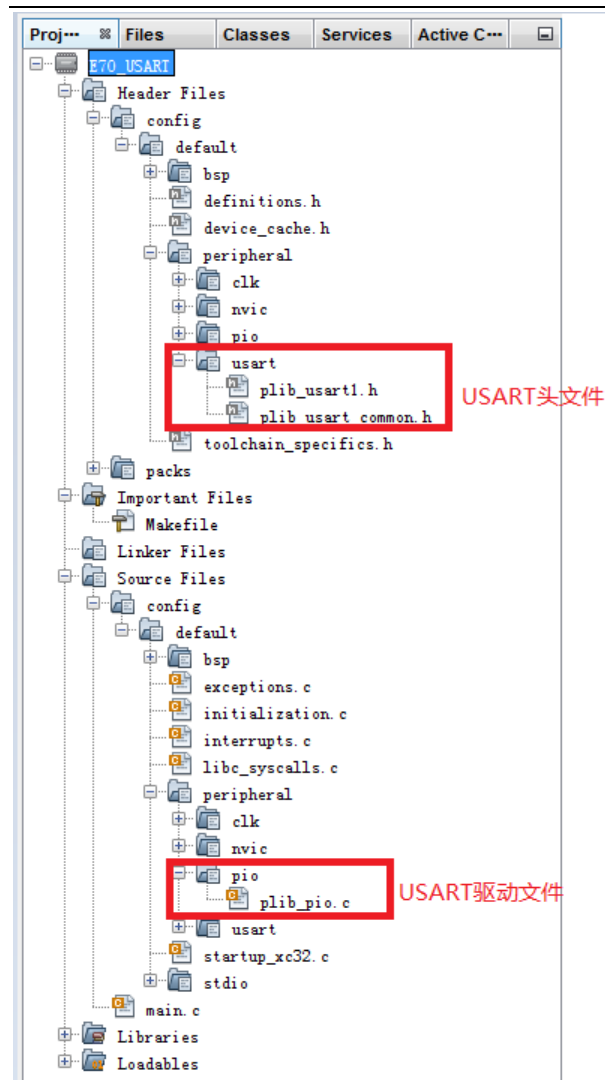
Output Xplained Window Project Graph* Clock Diagram Pin Diagram Pin Table Pin Settings										
Order: Pins Table View										
Pin Number	Pin ID	Custom Name	Function	Direction	Latch	Open Drain	PIO Interrupt	Pull Up	Pull Down	Glitch/Debounce Filter
101	PD10		Available	In	n/a	☐	Disabled	☐	☐	Disabled
102	PA0		Available	In	n/a	☐	Disabled	☐	☐	Disabled
103	PC17		Available	In	n/a	☐	Disabled	☐	☐	Disabled
104	JTAGSEL			In	n/a	☐		☐	☐	
105	PB4	USART1_TXD1	USART1_TXD1	n/a	n/a	☐	Disabled	☐	☐	Disabled
106	PD15		Available	In	n/a	☐	Disabled	☐	☐	Disabled

(六) “保存” 项目配置和 “代码生成”

在任务栏点击“保存”和“代码生成”按钮

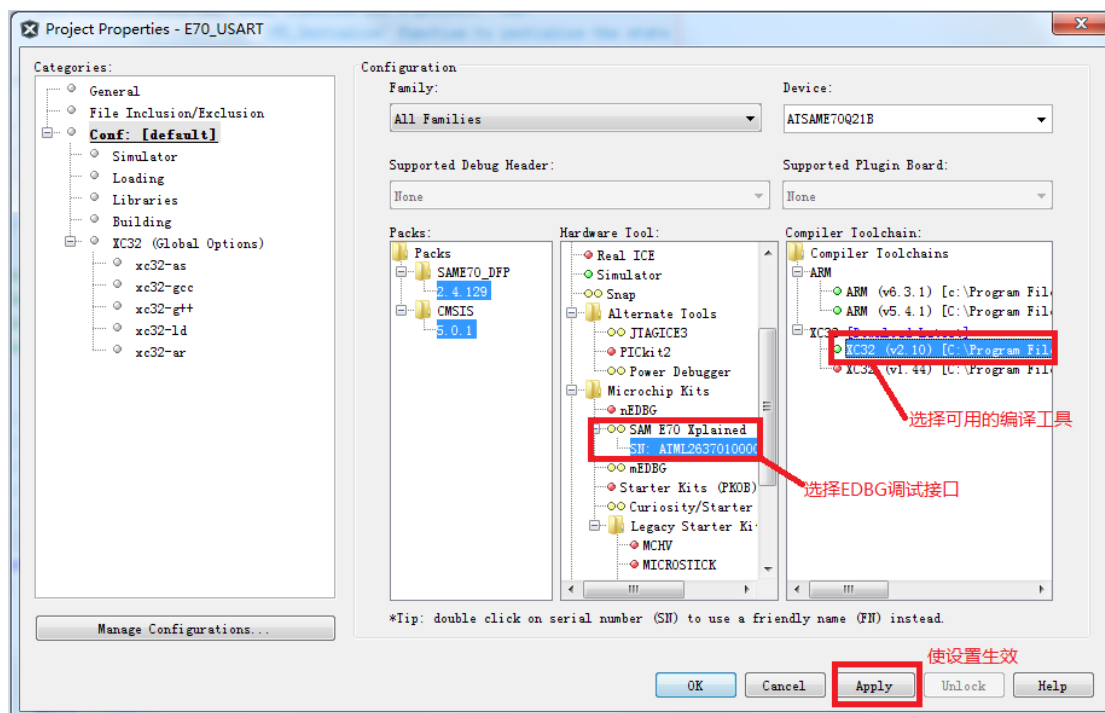
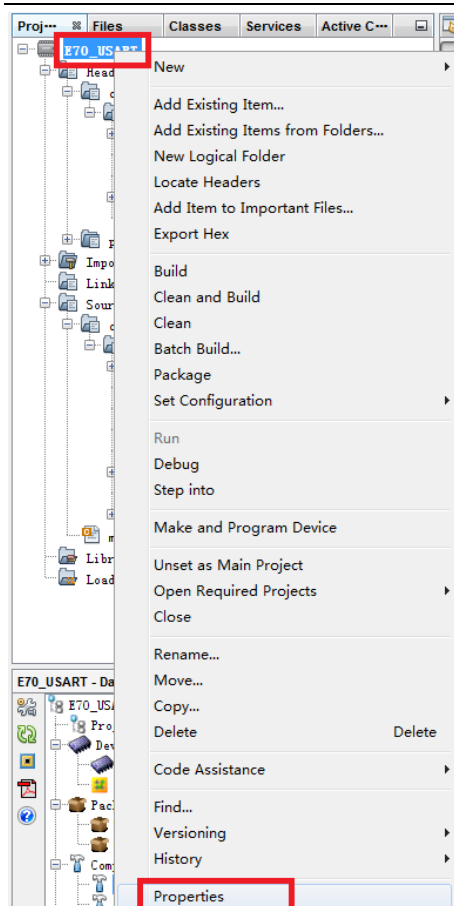


展开左侧的工程项目管理目录树，可以看到相关的 USART 头文件和源代码已经生成了。



(七) 设置项目的调试接口和编译器

选择项目 E70_USART，右键选择 Properties



(八)main.c 里增加如下测试代码

```

#include <stddef.h>                // Defines NULL
#include <stdbool.h>               // Defines true
#include <stdlib.h>                // Defines EXIT_FAILURE
#include <string.h>
#include "definitions.h"          // SYS function prototypes

// *****
// *****
// Section: Main Entry Point
// *****
// *****

#define RX_BUFFER_SIZE            10

char messageStart[] = "****  USART  echo  demo: Non-blocking Transfer with the interrupt
****\r\n\
**** Type 10 characters. The received characters are echoed back, and the LED is toggled ****\r\n";
char receiveBuffer[RX_BUFFER_SIZE] = {};
char echoBuffer[RX_BUFFER_SIZE + 4] = {};
char messageError[] = "**** USART error occurred ****\r\n";

bool errorStatus = false;
bool writeStatus = false;
bool readStatus = false;

void APP_WriteCallback(uintptr_t context)
{
    writeStatus = true;
}

void APP_ReadCallback(uintptr_t context)
{
    if(USART1_ErrorGet() != USART_ERROR_NONE)
    {
        /* ErrorGet clears errors, set error flag to notify console */
        errorStatus = true;
    }
    else
    {
        readStatus = true;
    }
}

```

```
// *****
// *****
// Section: Main Entry Point
// *****
// *****

int main ( void )
{
    /* Initialize all modules */
    SYS_Initialize ( NULL );
    LED_Off();

    /* Register callback functions and send start message */
    USART1_WriteCallbackRegister(APP_WriteCallback, 0);
    USART1_ReadCallbackRegister(APP_ReadCallback, 0);
    USART1_Write(&messageStart, sizeof(messageStart));

    while ( true )
    {
        if(errorStatus == true)
        {
            /* Send error message to console */
            errorStatus = false;
            USART1_Write(&messageError, sizeof(messageError));
        }
        else if(readStatus == true)
        {
            /* Echo back received buffer and Toggle LED */
            readStatus = false;

            echoBuffer[0] = '\n';
            echoBuffer[1] = '\r';
            memcpy(&echoBuffer[2], receiveBuffer,sizeof(receiveBuffer));
            echoBuffer[RX_BUFFER_SIZE+2] = '\n';
            echoBuffer[RX_BUFFER_SIZE+3] = '\r';

            USART1_Write(&echoBuffer, sizeof(echoBuffer));
            LED_Toggle();
        }
        else if(writeStatus == true)
        {
            /* Submit buffer to read user data */

```

```
        writeStatus = false;
        USART1_Read(&receiveBuffer, sizeof(receiveBuffer));
    }
    else
    {
        /* Repeat the loop */
        ;
    }
}

/* Execution should not come here during normal operation */

return ( EXIT_FAILURE );
}
```

(九) 编译下载测试

用 Micro USB 线通过 EDBG 调试口将 SAM E70 Xplained Board 开发板连接到电脑

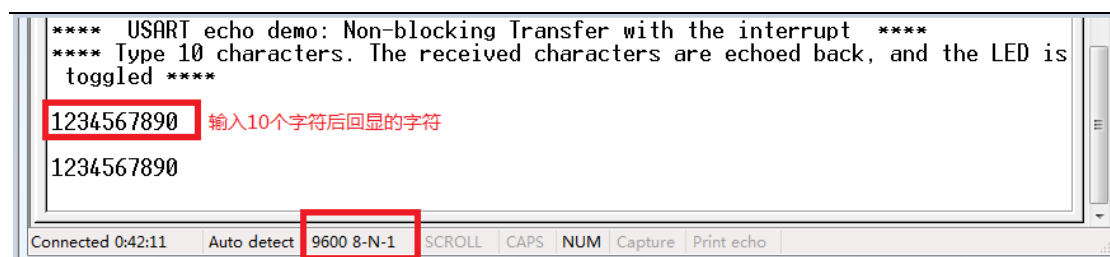
编译并下载程序：



在 Windows 的设备管理器里面，确认 EDBG 的串口已经出现。



在串口终端程序打开 EDBG 的串口，你将看到这样的打印信息：



四、 总结

本文展示了如何通过 MPLAB X IDE 和 MHC 一步步完成了一个通过 USART 口输入和输出的程序，开发人员可以从这个过程了解到 Harmony 配置开发的全过程。