

SmartFusion2 Kit UART demo User Guide

文档性质：例程指导

LEADERSHIP IN FPGA SECURITY

- State-of-the-art security enables root-of-trust applications
- Radically transforms the usefulness of FPGAs in security applications

LEADERSHIP IN FPGA RELIABILITY

- Only SoC FPGA with SEU immune FPGA configuration cells and processor
- Reliability designed for safety critical and mission critical systems

LEADERSHIP IN LOW POWER FPGAS

- 100X lower static power in Flash*Freeze mode
- 50% to 30% lower static power

LEADERSHIP IN REAL-TIME FPGA PERFORMANCE

- ARM® Cortex™-M3 real-time microcontroller
- Flash*Freeze real-time power management
- Instant-on real-time availability



文档说明

关键字

SmartFusion2、Liberio11.x、Uart

摘要

修订版本	修订时间	修订人	修订内容
v1.0			创建文档

目 录

1. N a m e	
2. Content.....	1
3. P u r p o s e	1
4. P l a t f o r m	1
5. Key words.....	1
6. S t e p s	2
6.1 Hardware demo.....	2
6.1.1 n e w p r o	2
6.1.2 C o n f i g	3
6.1.3 C o m p i l e	7
6.1.4 D o w n l o a d	7
6.2 Soft Demo (Keil)	8
6.2.1 G e n r a t e	8
6.2.2 Keil Project.....	9
6.2.3 L i b g e n r a t e	9
6.2.4 F P G A f i l e	10
6.2.5 main.c and debug ..	11
6.2.6 R e s u l t	11
A.1 main.c	13

1. 例程名称

SmartFusion2 简单的 Uart 使用例程。

2. 例程内容

本例程主要为测试 SmartFusion2 的 32 位 Uart 的使用，通过 MCU 输出在串口输出一个字符串。

3. 例程目的

- 演示 MSS UART 的使用；
- 熟悉 SmartFusion2 的开发流程。

4. 平台介绍

- 开发环境：Libero11.5SP2、Keil MDK4.7/5.1
- 硬件环境：Low Cost SmartFusion2 Kit、FlashPro4、J-Link

5. 例程关键词

- M3 Timer
- Keil、J-Link
- SoftConsole
- Libero11.5SP2

6. 实验步骤

6.1 硬件设计步骤

6.1.1 新建工程

- (1) 打开 Libero 开发环境（Start > Programs > Microsemi Libero SoC v11.x > Libero SoC v11.x）或者双击桌面图标；
- (2) 新建工程：点击菜单栏：Project > New Project，按照图 6.1 所示，输入工程名（这里命名为 UartDemo）、存储位置、器件（M2S010T-FGG484）、勾选“Use Design Tool”、选中 MSS。

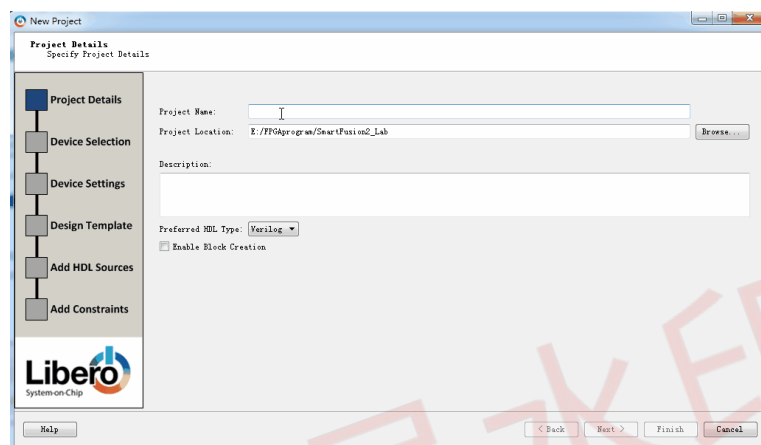


图 6.1 工程设置

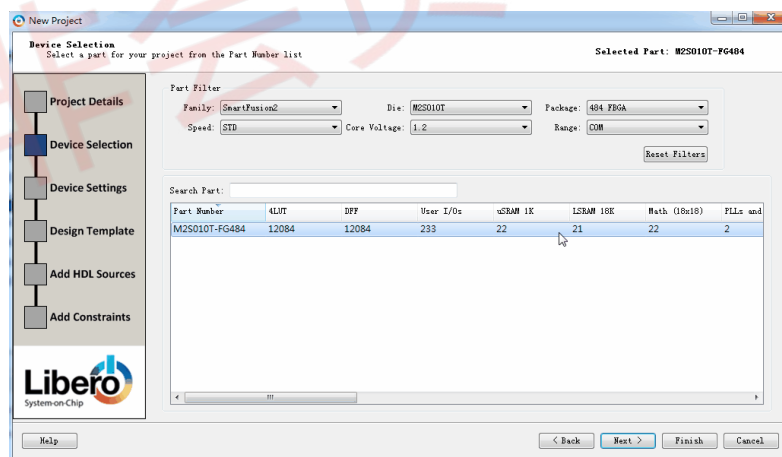


图 6.2 选择器件

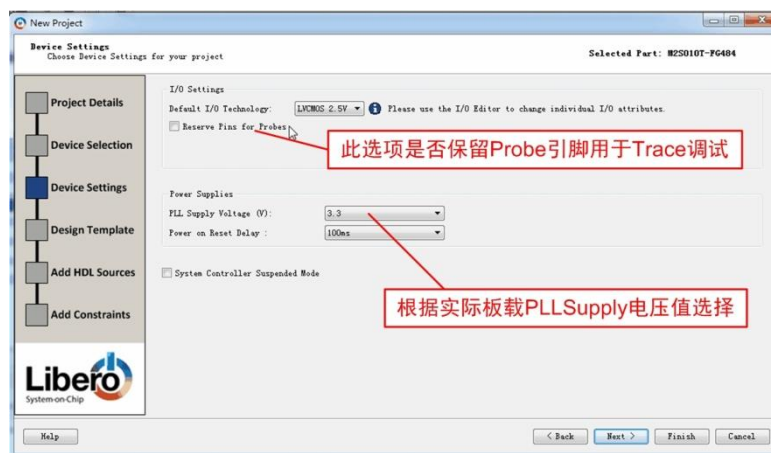


图 6.3 选择器件

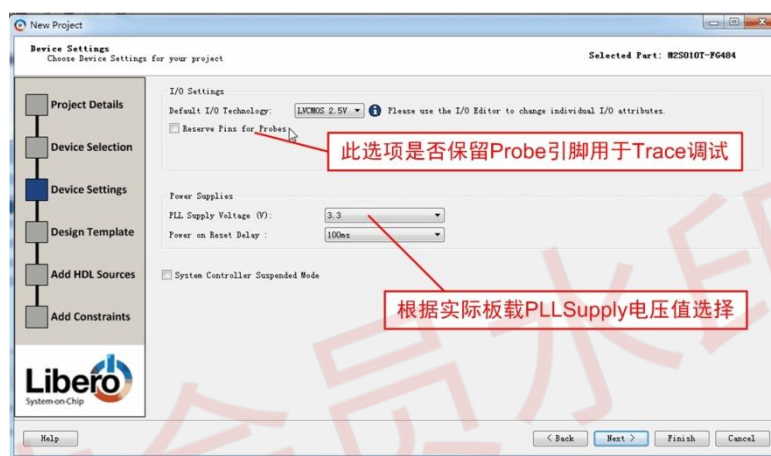


图 6.4 选择器件

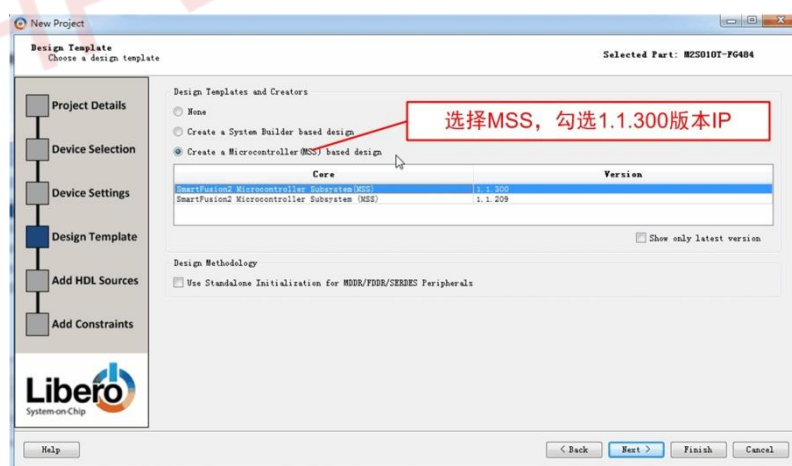


图 6.5 选择器件

其他“Add HDL Source”和“Add Constraints”这里默认，选择 Next，最后点击 Finish 工程建立。

6.1.2 配置工程

(1) 在打开的环境界面中，双击画布中如图 6.6 中的“UartDemo_MSS_0”，弹出配置界

面，双击进入配置界面；

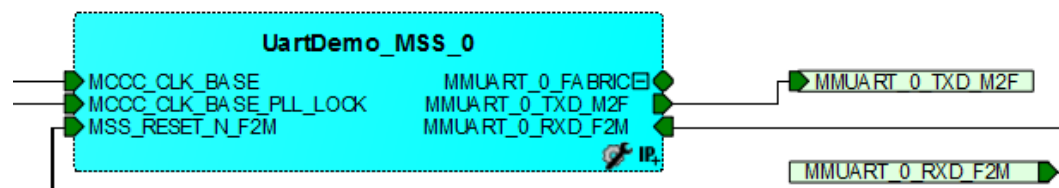


图 6.6 编辑工程 (1)

(2) 在配置界面中去掉所有模块的勾选项，只保留 UART 模块，如图 6.7 所示。其中一个外设对应一个模块，点击模块右下角选框可以使能/失能某一模块，使能模块呈现浅蓝色，失能模块呈现灰色；

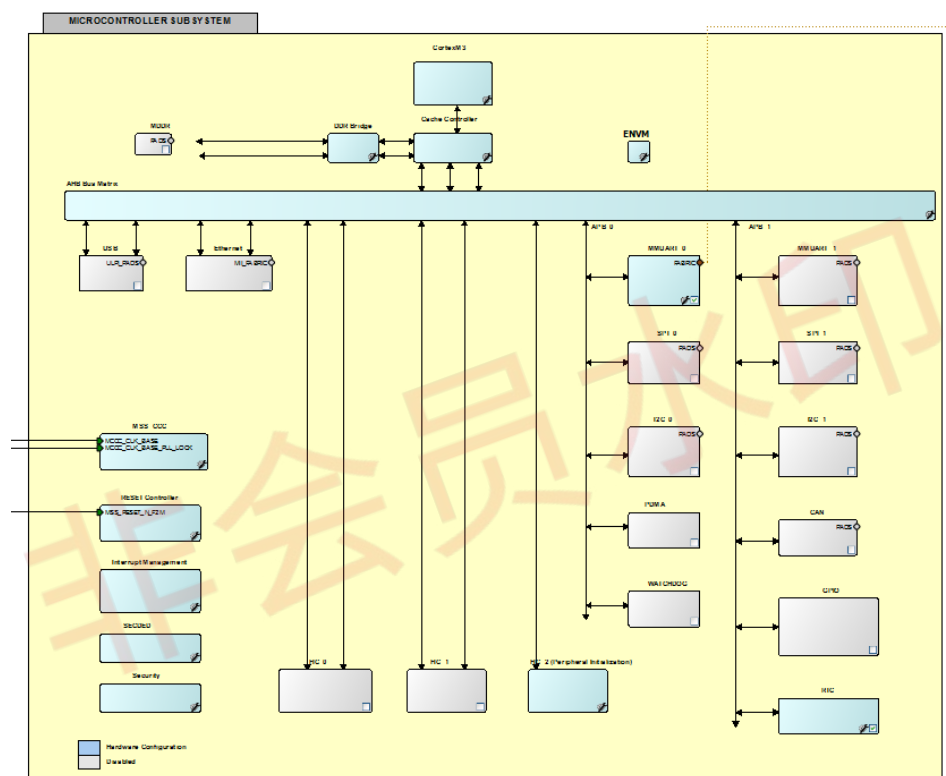


图 6.7 配置工程 (2)

(3) 双击“MSS_CCC 模块”，在弹出的配置窗口中，设定“CLK_BASE”为“100MHz”，勾选“Monitor FPGA Fabric PLL Lock”，M3_CLK 设定为 100MHz，其他不变，点击“OK”关闭配置窗口，如图 6.8 所示；

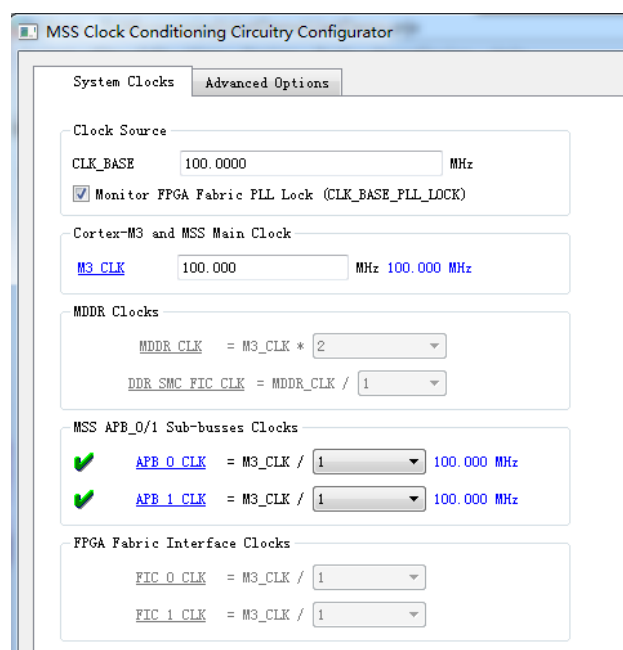


图 6.8 配置 MSS_CCC

- (4) 双击“RESET_Controller”，在弹出的配置窗口中，勾选“Enable FPGA Fabric to MSS Reset(MSS_RESET_N_F2M)”，如图 6.9 所示，点击“OK”关闭窗口；

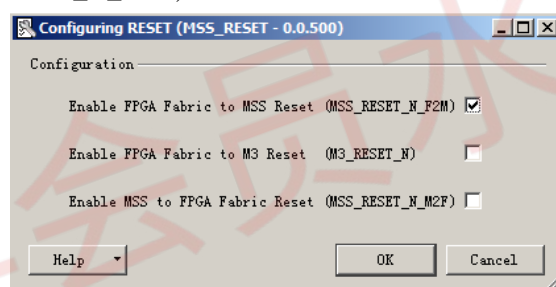



图 6.9 配置 MSS_REST

- (5) 配置完成后，点击  保存文件；
- (6) 回到主界面，会发现“UartDemo_MSS_0”模块上有一个感叹号，表示 MSS 模块已被更新，右击模块，在弹出的菜单上选择“Update Instance(s) with Latest Component...”，如图 6.10 所示，配置完成后，效果如图 6.11 所示；

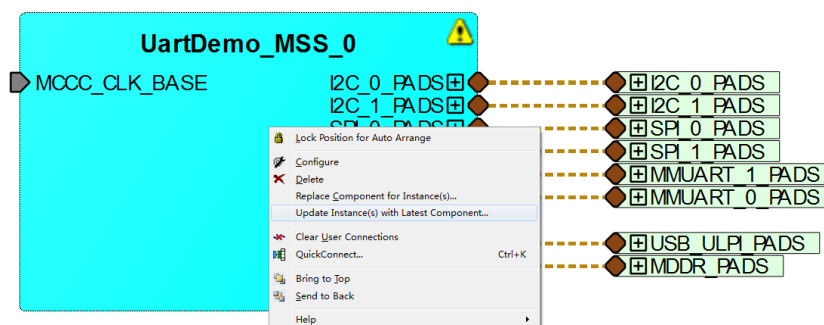


图 6.10 更新 MSS (1)

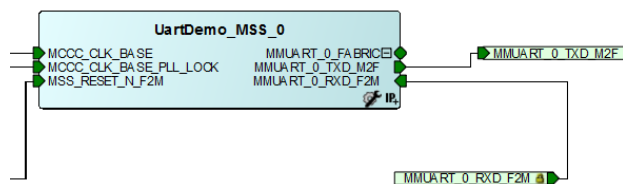


图 6.11 更新 MSS (2)

- (7) 双击“FCCC_0”进入配置界面，输入时钟设置为内部 50MHz 晶振，输出为 100MHz，（也可选择其他晶振，这里仅作例程演示），如图 6.12 所示；

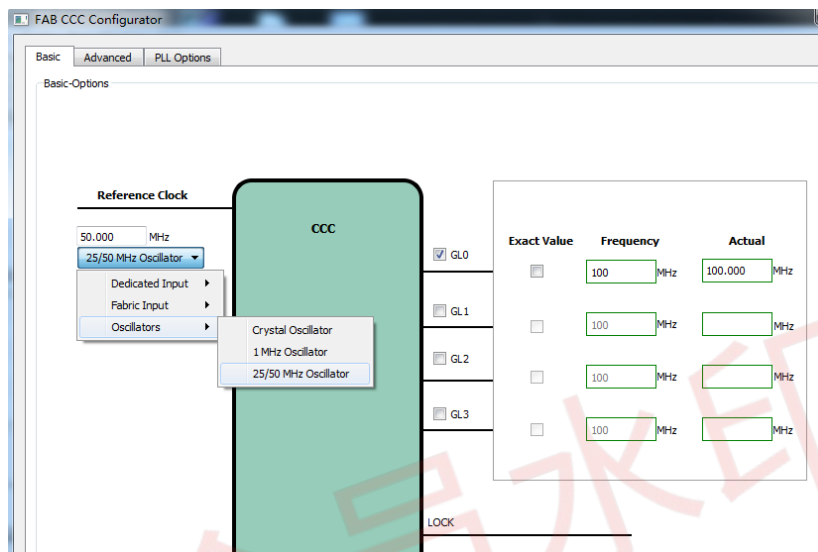


图 6.12 配置 FABRIC CCC

- (8) 双击“OSC_0”模块，进入时钟源配置界面，选择时钟源为“On-chip 50MHz RC...”，勾选“Drives Fabric CCC(s)”，点击确定，如图 6.13 所示；

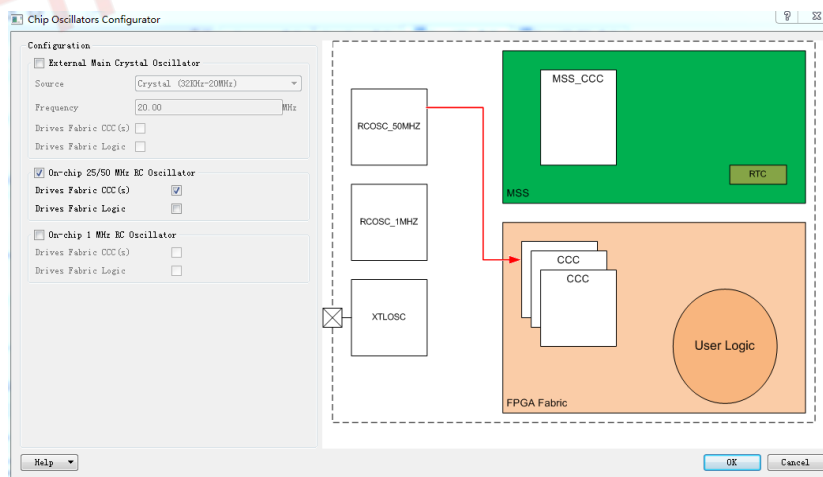




图 6.13 配置时钟源

- (9) 配置好模块之后，使用工具栏的连线工具，进行连线（或使用前面介绍的其他几种布线方式均可），连线如图 6.14 所示，保存，然后点击；

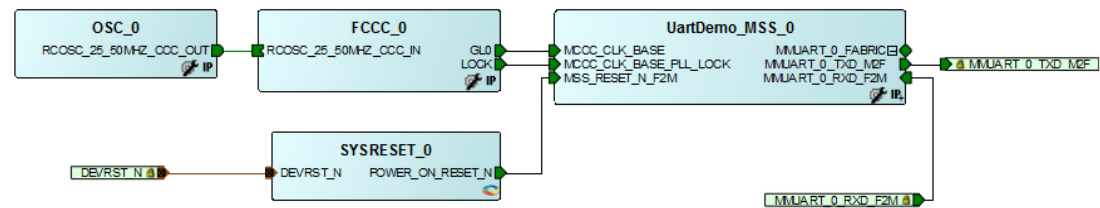


图 6.14 画布布线

6.1.3 编译工程

- (1) 在左侧“Design Flow”窗口中，依次双击“Synthesize”、“Compile”两项，编译工程，如图 6.15 所示；

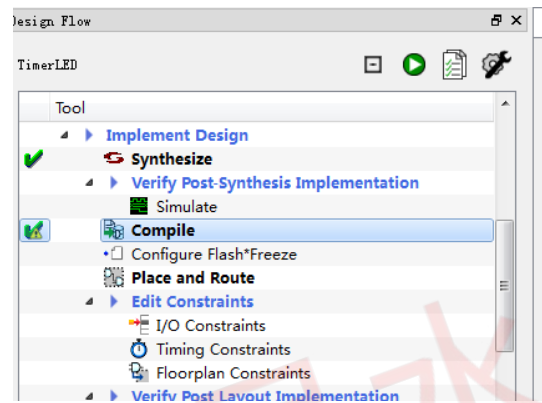



图 6.15 编译

- (2) 双击“Edit Attributes-I/O Constraints”，打开 I/O 设置界面，分配串口的输入输出脚分别到 T18、T19；

Ports		Package Pins		Package Viewer				
	Port Name	Direction	I/O Standard	Pin Number	Locked	Macro Cell	Bank Name	
1	DEV_RST_N	Input	--	R15	<input checked="" type="checkbox"/>	ADLIB:SYSRESET	--	
2	MMUART_0_RXD_F2M	Input	LVC MOS25	T18	<input checked="" type="checkbox"/>	ADLIB:INBUF	Bank2	
3	MMUART_0_TXD_M2F	Output	LVC MOS25	T19	<input checked="" type="checkbox"/>	ADLIB:OUTBUF	Bank2	

图 6.16 引脚设置

- (3) 再依次双击主窗口左侧“Place and Route”、“Verify Timing”、“Generate Bitstream”（或直接点击 Generate Bitstream；或直接点击 ），在弹出的选择提示选择 OK，即可完成下载前所有工作。

6.1.4 下载工程

- (1) 最后连接开发板和 FlashPro4（注意 P3 拨码开发位置拨至左侧），双击“Run Program Action”，下载工程，如图 6.17 所示，等待下载完成。

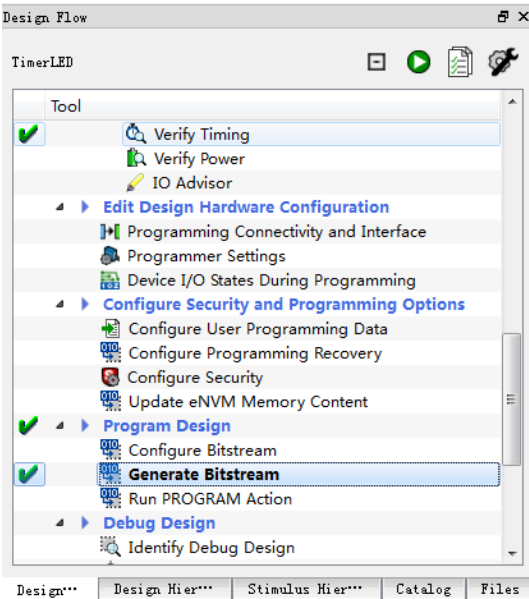


图 6.17 下载工程

- (2) 如果后面软件开发使用 Keil 或 IAR，请下载完成后拨出 FlashPro4 调试器后，再连接 Jlink 或 Ulink 等其他第三方软件开发调试器，并将 P3 拨码开关拨至右侧。若如两种下载器同时占用 JTAG 接口，可能导致调试器烧毁；如果软件开发使用 SoftConsole 则不必做任何更改。

6.2 软件设计步骤（使用 Keil）

6.2.1 生成固件驱动和工程

在 Libero 软件左侧流程栏里，Handoff Design for Firmware Development 窗口下双击 **Configure Firmware Cores** 打开固件库选择界面，如下图所示，本实验使用 M3 内核库文件、MMUART 驱动库即可，用户可选择或不勾选其他模块。如有修改点击保存。

	Generate	Instance Name	Core Type	Version	Compatible Hardware Inst
1	<input checked="" type="checkbox"/>	SmartFusion2_CMSIS_0	SmartFusion2_CMSIS	2.2.10	VartDemo_MSS
2	<input type="checkbox"/>		SmartFusion2_MSS_HPDMA_Driver	2.1.10	VartDemo_MSS
3	<input checked="" type="checkbox"/>	SmartFusion2_MSS_MMUART_Driver_0	SmartFusion2_MSS_MMUART_Driver	2.0.101	VartDemo_MSS:MMUART_0
4	<input checked="" type="checkbox"/>	SmartFusion2_MSS_NVM_Driver_0	SmartFusion2_MSS_NVM_Driver	2.2.10	VartDemo_MSS
5	<input checked="" type="checkbox"/>	SmartFusion2_MSS_RTC_Driver_0	SmartFusion2_MSS_RTC_Driver	2.1.10	VartDemo_MSS:RTC
6	<input type="checkbox"/>		SmartFusion2_MSS_System_Servie	2.6.10	VartDemo_MSS
7	<input type="checkbox"/>		SmartFusion2_MSS_Timer_Driver	2.1.10	VartDemo_MSS

图 6.18 固件库选择

双击“Export Firmware”选择 SoftConsole，并勾选“Create project...”，点击 OK，生成 Keil 所需工程，如下图所示。

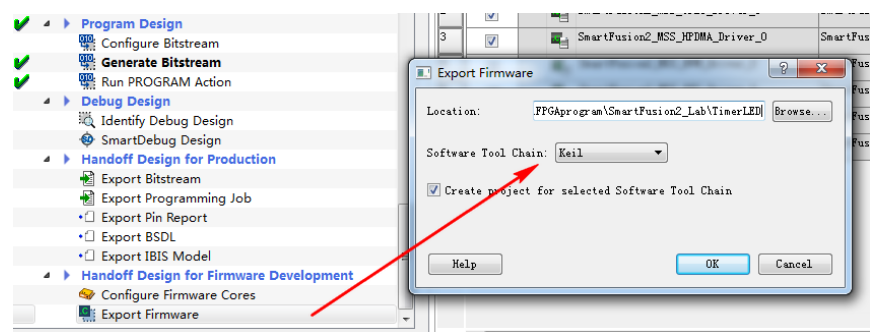


图 6.19 生成软件工程文件

6.2.2 打开 Keil 工程

生成的工程目录为“用户目录\UartDemo\Keil”，自动生成的工程使用 Multi-Project 管理方式，如错误！未找到引用源。所示，双击.uvmpw 工程文件打开 Keil 工程。（Keil 4.7 或以上版本才支持 Microsemi SmartFusion2 器件，请注意安装 Keil 版本信息）

名称	修改日期	类型	大小
CMSIS	2015/5/31 16:18	文件夹	
drivers	2015/5/31 16:18	文件夹	
drivers_config	2015/5/31 16:18	文件夹	
hal	2015/5/31 16:18	文件夹	
main.c	2015/5/31 16:18	C Source	1 KB
README.txt	2015/5/31 16:18	TXT 文件	1 KB
UartDemo_hw_platform.h	2015/5/31 16:18	C/C++ Header	1 KB
UartDemo_MSS_CM3.uvmpw	2015/5/31 16:18	Microvision4 Multi-Pr...	1 KB
UartDemo_MSS_CM3_app.ini	2015/5/31 16:18	EditPlus INI (.ini)	1 KB
UartDemo_MSS_CM3_app.uvproj	2015/5/31 16:18	Microvision4 Project	31 KB
UartDemo_MSS_CM3_hw_platform.uv...	2015/5/31 16:18	Microvision4 Project	38 KB

图 6.20 打开 Keil 工程

6.2.3 编译硬件驱动库工程，生成系统硬件 lib 库映射

打开 Keil，软件自动生成了两个工程：第一个工程为硬件对应库工程，包括 M3 内核、外设驱动库文件、硬件地址分配等；第二个为用户工程，用户直接在此工程下编写代码完成设计。

由于使用 MDK4.7，版本较低不支持 Microsemi 全系列器件，需要按下图重新选择器件，具体操作如下图所示。

设置完成后必须全编译此工程，完成硬件 lib 映射，其他不再赘述。

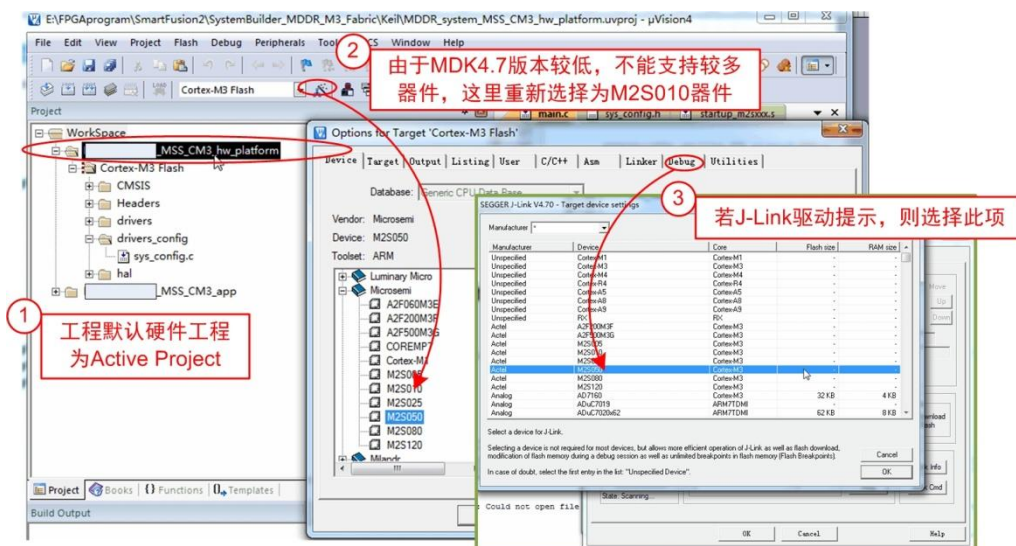


图 6-21: 修改工程信息, 编译

6.2.4 编译应用库工程，生成下载文件

编译完成硬件库工程后，重新设置用户库工程为 Active Project，如下图所示。

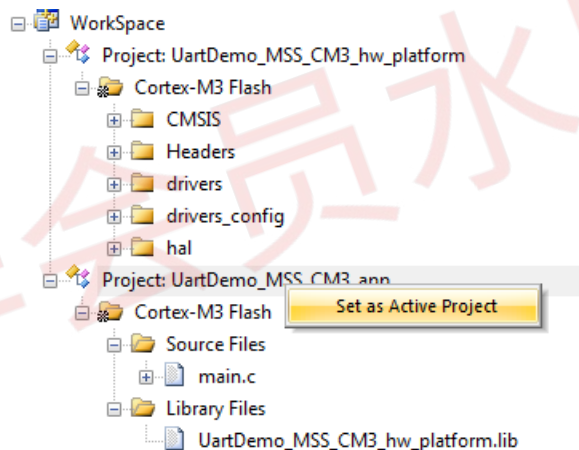


图 6-22: 设置 Active Project

设置系统 Target Options, 同图 6-21 类似, 修改器件、设置 Debug 目标等, 具体如下。其中值得注意的是: 在重选择器件后 Target 目录下需要重新勾选 ROM, 否则编译不通过。

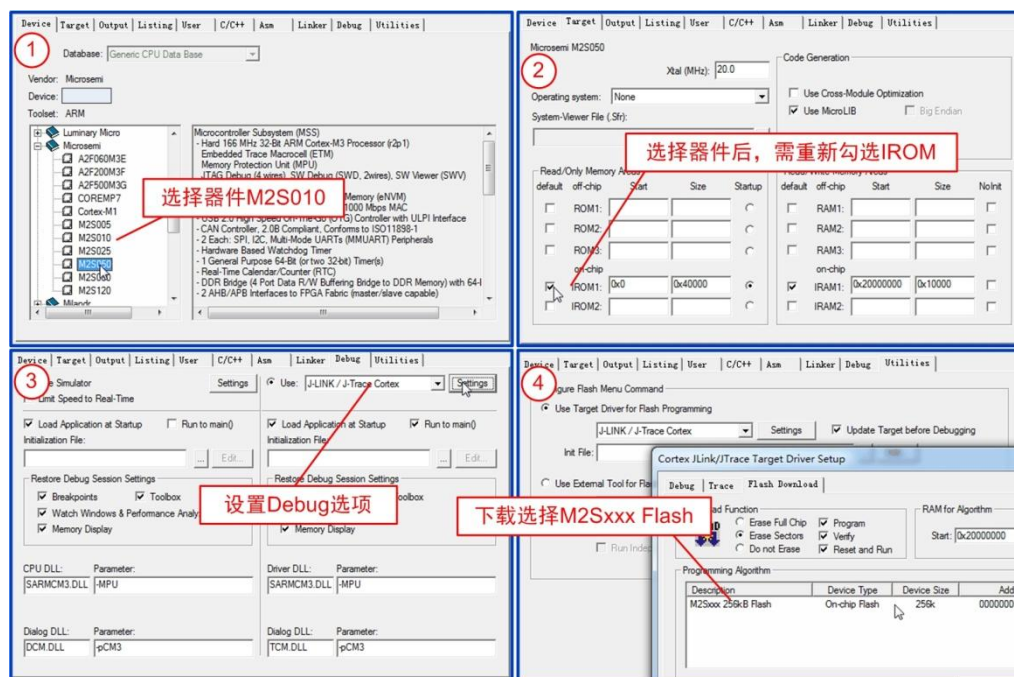


图 6-23: Target Options...设置

6.2.5 编写 main.c 代码，编译下载

打开 main.c 文件，键入附录 A.1 的代码，点击保存，全编译工程，若操作正确则信息框提示如下。正确连接 Keil 调试器，点击下载即可。

```
Build Output
Build Project 'UartDemo_MSS_CM3_app' - Target 'Cortex-M3 Flash'
compiling main.c...
linking...
Program Size: Code=4088 RO-data=424 RW-data=28 ZI-data=4228
".\obj\UartDemo_MSS_CM3_app.axf" - 0 Error(s), 0 Warning(s).
```

图 6-24: Keil 编译通过

6.2.6 实验结果与分析

串口连接开发板右侧排针的第 3、5、7 号引脚。使用下图中参数连接串口，复位开发板后，在串口程序中可以看到发送过来的串口信息，如下图：

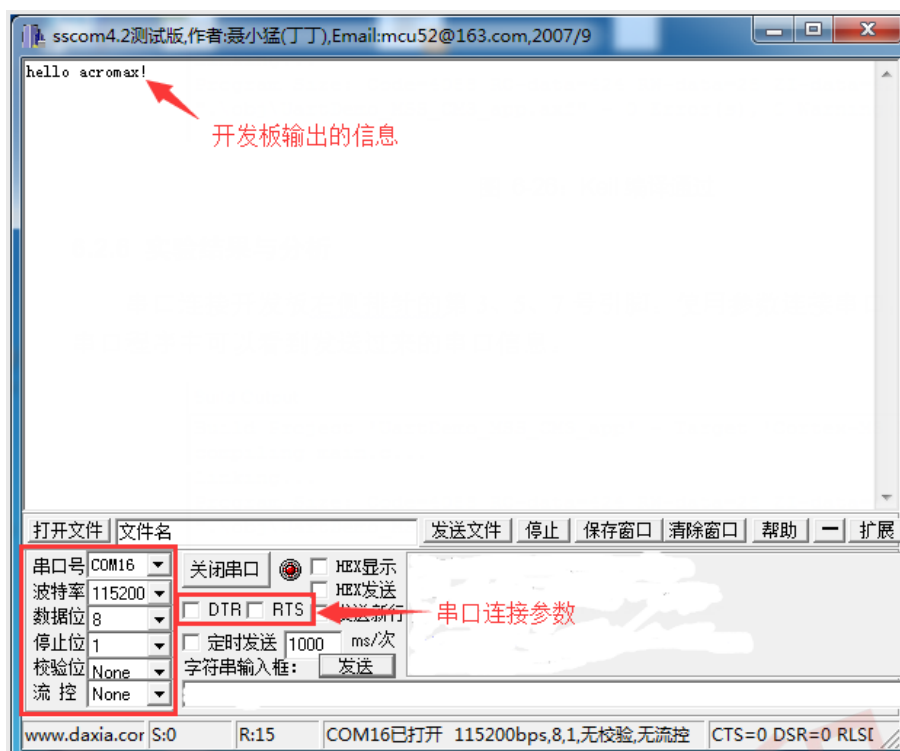


图 6-25: Keil 编译通过

附录A 附录

A.1 main.c 主程序源代码

```
/******版权申明*****  
** 文件名称:      main.c  
** 最后修改日期:   2015-05-06  
** 最新版本:      V1.0  
** 功能描述:      SmartFusion2 Uart 通信程序  
*****/  
  
#include "mss_uart.h"  
  
/******/  
// 函数名称: 主函数  
// 函数功能: 从串口输出一个字符串  
/******/  
int main()  
{  
    uint8_t message[] = "hello acromax!"; //要输出的字符串信息  
    // 初始化串口  
    MSS_UART_init(&g_mss_uart0,  
                  MSS_UART_115200_BAUD,  
                  MSS_UART_DATA_8_BITS | MSS_UART_NO_PARITY | MSS_UART_ONE_STOP_BIT);  
    // 串口发送字符串  
    MSS_UART_polled_tx(&g_mss_uart0, message, sizeof(message));  
  
    for(;;)  
    {  
    }  
}/******文件结束*****/  

```