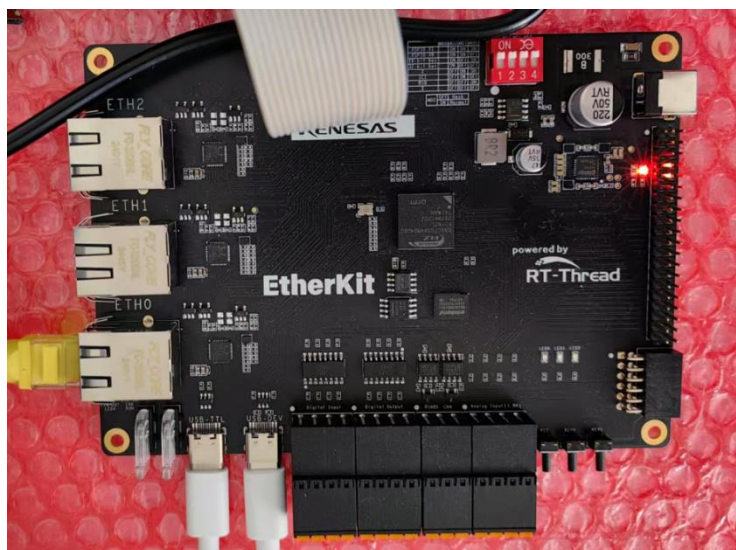


RZN2L RS485 操作手册-----基于 Etherkit 开发板

简介

本应用笔记介绍了基于 RZ/N2 Etherkit 开发板的定时器 RS485 的操作。分别介绍 IDE IAR 和 E2studio 软件下的操作。



开发工具

- IDE: IAR EW for Arm 9.50.2
E2studio 2024-01.1
- FSP: RZ/N2 FSP V2.0
- 仿真器: Jlink V12

实验材料

- Etherkit 开发板
- Jlink 仿真器, 需支持瑞萨 R52 内核
- RS485 通信需要两个板子

实验部分

1.硬件设置及软件安装	2
2 .IAR 环境工程介绍	3
3 .E2studio 环境工程介绍	9

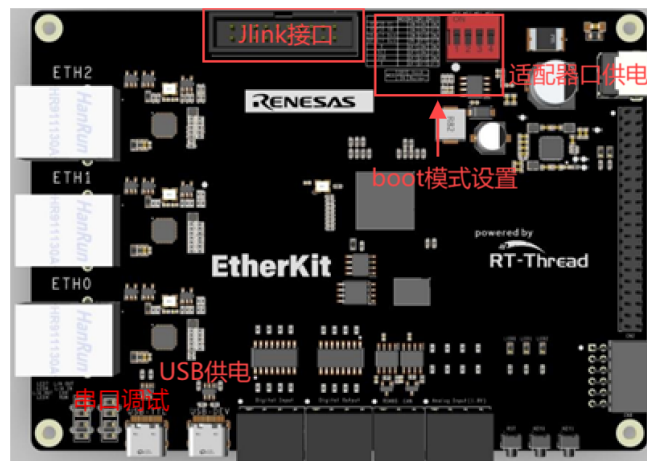
1 .硬件设置及软件安装

本节 EtherKit 开发板硬件设置。

1.1

开发板设置：

- 供电：可选 USB 供电或适配器供电
- Boot 模式设置：推荐 xSPI0 x1 boot mode
- Jlink v12

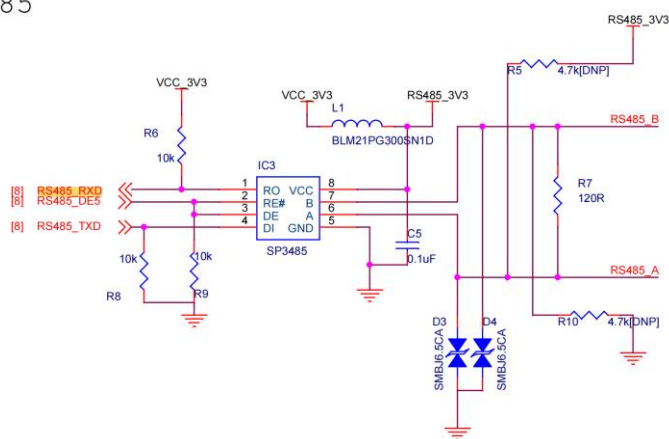


1.2

硬件原理图：

板载资源 RS485 接口，连接到 N2L 芯片的 SCI5；

RS485



本节完

2 .IAR 环境工程介绍

本节介绍 IAR 环境下 RS485 工程介绍。

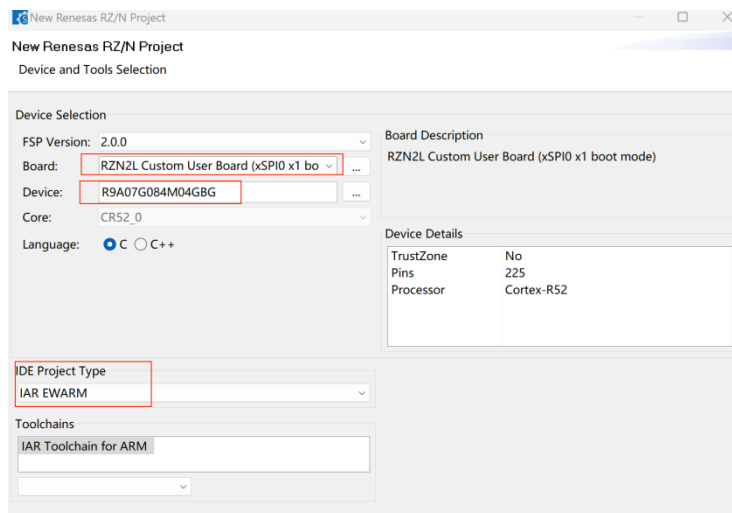
2.1

● 打开 FSP 新建工程：RS485

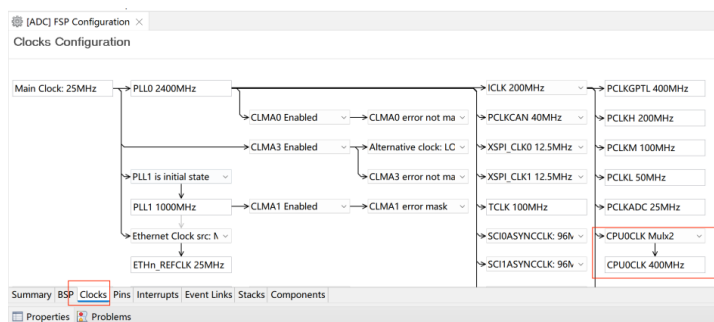
1. Boot 模式选择 RZN2L Custom User Board (XSPI0 x1 boot mode)

芯片型号：R9A07G084M04GBG

IDE Project Type 选择 IAR EWARM

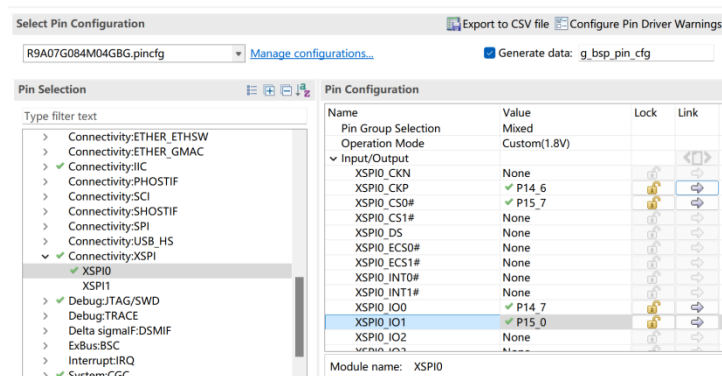


1. 主时钟设置 400MHZ



2. 配置 XSPI0

Pin Configuration



3. P143 引脚配置

P143 设置输出模式，初始电平 0,做为指示灯。

Select Pin Configuration Export to CSV file Configure Pin Driver Warnings

R9A07G084M04GBG.pincfg Manage configurations... Generate data: g_bsp_pin_cfg

Pin Selection
Type filter text

- > P12
- > P13
- > P14
 - P14_0
 - P14_1
 - P14_2
 - P14_3**
 - P14_4
 - P14_5
 - P14_6
 - P14_7
- > P15
- > P16

Pin Configuration

Name	Value
Comment	
Mode	Output mode (Low & Not Into Input)
Pull up/down	None
Output Type	CMOS
Drive Capacity	Low
Region	Safety
Schmitt Trigger	None
Slew Rate	Slow
Input/Output	P14_3
	GPIO

Module name: P14_3
Port Capabilities: ETHER, ETH0, COL, ETH0, ETH2, COL

Pin Function Pin Number

Summary BSP Clocks **Pins** Interrupts Event Links Stacks Components

4. 配置 SCIS

Pin Configuration Export to CSV file Configure Pin Driver Warnings Generate Project Content

R9A07G084M04GBG.pincfg Manage configurations... Generate data: g_bsp_pin_cfg

Pin Selection
Type filter text

- > Connectivity:PHOSTIF
- > Connectivity:SCI
 - SCI0
 - SCI1
 - SCI2
 - SCI3
 - SCI4**
- > Connectivity:SHOSTIF
- > Connectivity:SPI
- > Connectivity:USB_HS
- > Connectivity:XSPI
- > Debug:JTAG/GWD
- > Debug:TRACE
- > Delta:signalIF-DSMIF

Pin Configuration

Name	Value	Lock	Link
Pin Group Selection	Mixed		
Operation Mode	Custom		
Input/Output	None		
CTS#	None		
CTS_RTS_SSS#	None		
DES	P22_0		
RXD_MISO	P21_2		
SCIS	None		
SCL5	None		
SDA5	None		
TXD_MOSI	P21_3		

Module name: SCIS
Usage: In initial register value, when CCR0.TE bit is 0, and the terminal function is TXDn, the terminal outputs high impedance.

Pin Function Pin Number

Summary BSP Clocks **Pins** Interrupts Event Links Stacks Components

5. 添加 new Stack: UART

New Stack---Connectivity--UART(r_sci_uart)

波特率 115200

Threads New Thread Remove HAL/Common Stacks New Stack >

g_ioport I/O Port (r_ioport)
Memory config check
g_uart5 UART (r_sci_uart)

Objects New Object Remove

Summary BSP Clocks Pins Interrupts Event Links **Stacks** Components

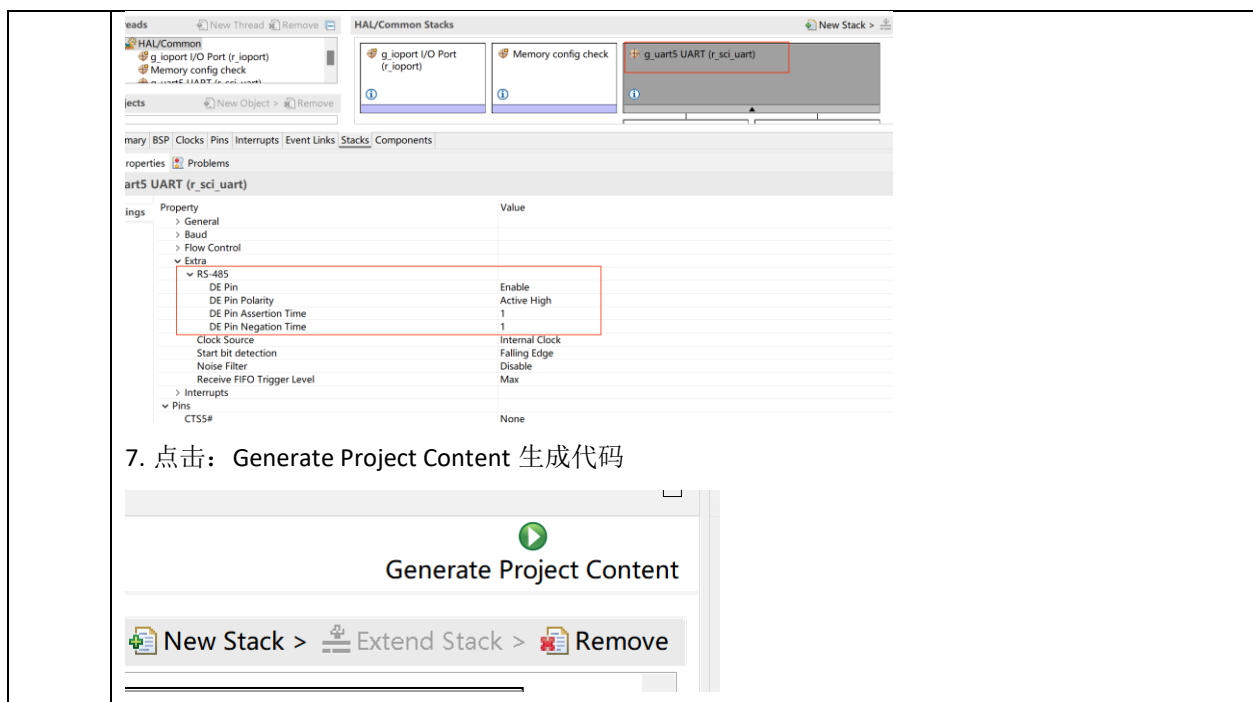
Properties Problems

g_uart5 UART (r_sci_uart)

Settings

Property	Value
Module g_uart5 UART (r_sci_uart)	
General	
Name	g_uart5
Channel	5
Data Bits	8bits
Parity	None
Stop Bits	1bit
Baud	
Baud Rate	115200
Baud Rate Modulation	Disabled
Max Error (%)	5
Synchronizer Bypass	Not Bypassed (The operation clock is SCInASYNCCLK)
Flow Control	
Extra	
Interrupts	
Pins	
CTS#	None

6. RS485 DE pin 设置



7. 点击：Generate Project Content 生成代码

Generate Project Content

New Stack > Extend Stack > Remove

2.2

8. 打开生成的代码

- 仿真器由 Ijet 切换为 Jlink
- 编写用户代码：一个工程两套代码，由 send_board 定义编译哪套代码。

```

61 {
62     /* TODO: add your own code here */
63     /* Open the transfer instance with initial configuration. */
64     fsp_err_t err = R_SCI_UART_Open(&g_uart5_ctrl, &g_uart5_cfg);
65
66     __asm volatile ("cpsie i");
67     __asm volatile ("isb");
68     for (uint32_t i = 0; i < 20; i++)
69     {
70         g_src[i] = (uint8_t) ('A' + (i % 26));
71     }
72     while(1)
73     {
74         #if send_board
75             err = R_SCI_UART_Write(&g_uart5_ctrl, g_src, 10);
76             g_transfer_complete = 0;
77             while (!g_transfer_complete)
78             {
79             }
80             R_BSP_SoftwareDelay (1000, BSP_DELAY_UNITS_MILLISECONDS);
81         #else
82             err = R_SCI_UART_Read(&g_uart5_ctrl, g_dest, 10);
83             g_receive_complete = 0;
84             while (!g_receive_complete)
85             {
86             }
87             R_BSP_SoftwareDelay (800, BSP_DELAY_UNITS_MILLISECONDS);
88         #endif
89     }
90 }

```

- #define send_board 1 编译工程，下载代码到发送数据用的板子。板子自由运行，连续发送数据。
- #define send_board 0 编译工程，用于接收数据板子仿真运行。
- Rebuild All---编译工程 无报错

```

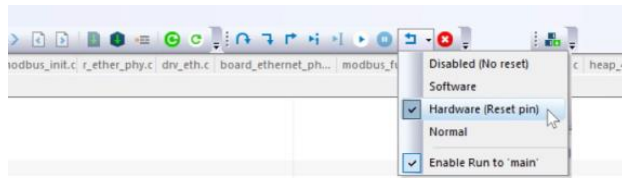
Build
Messages

Total number of errors: 0
Total number of warnings: 0
Resolving dependencies...
Build succeeded

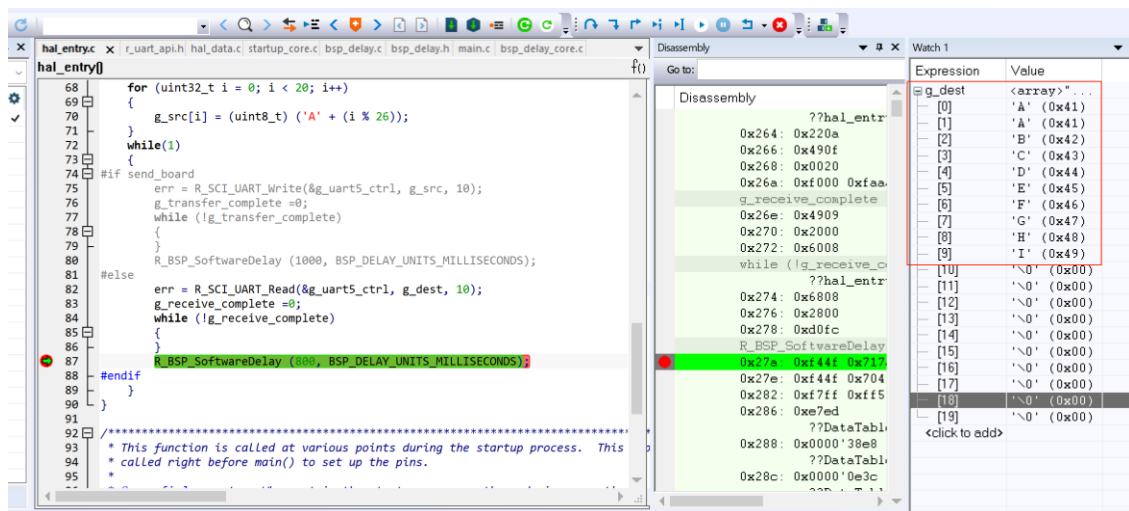
```

2.3

- Download and Debug --- 下载程序
- 下载工程到开发板，进入仿真界面
- Debug 复位设置为 Hardware



- 设置断点，全速运行代码，接收到一次数据，停到断点处：



可以看到 g_dest 中接收到另一个板子发送来的数据。

本节完

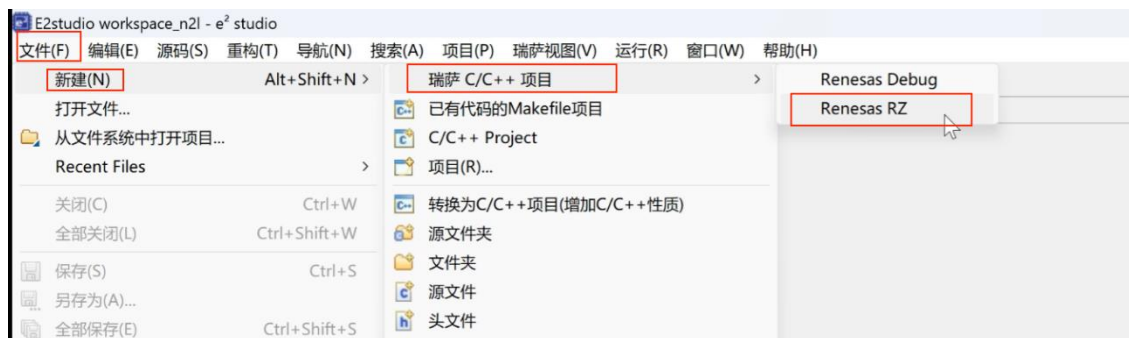
3 .E2studio 环境工程介绍

本节介绍使用 E2studio 环境创建 IIC 工程。

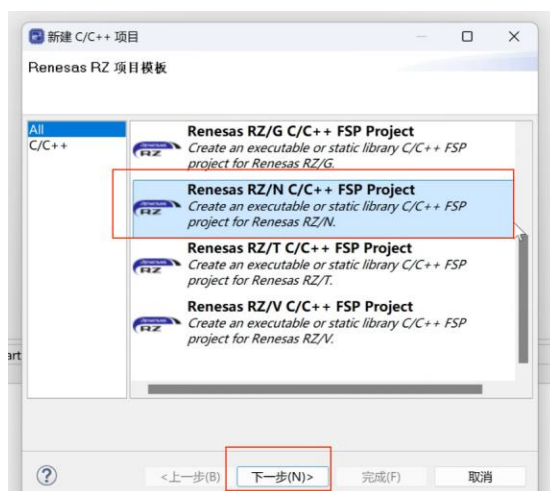
3.1

● 打开 E2studio, 新建工程

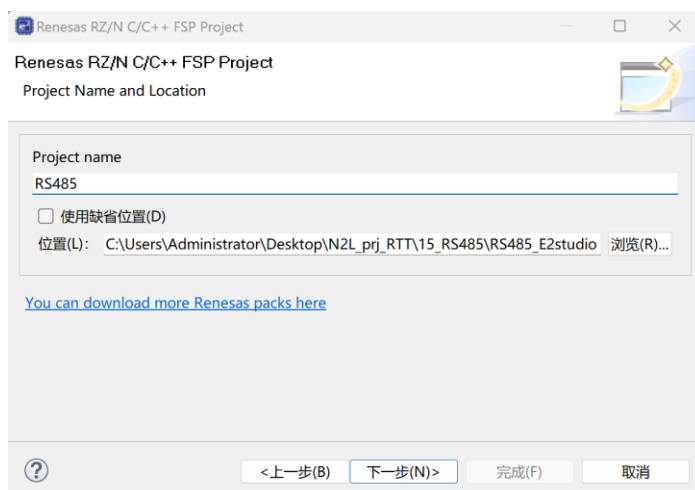
1. 选择 文件--新建--瑞萨 C/C++项目--Renesas RZ:



2. 选择 Renesas RZ/N C/C++ FSP Project

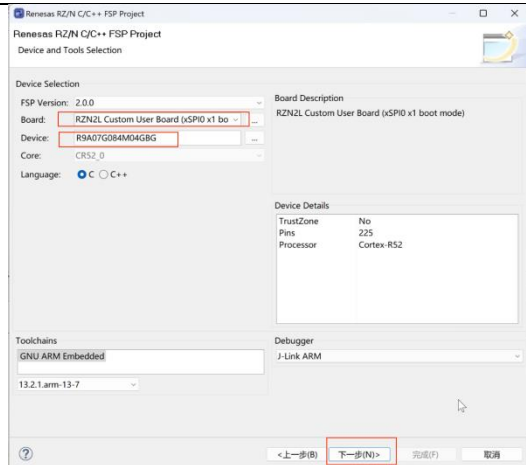


3. 设置项目名称: RS485, 选择工程保存路径

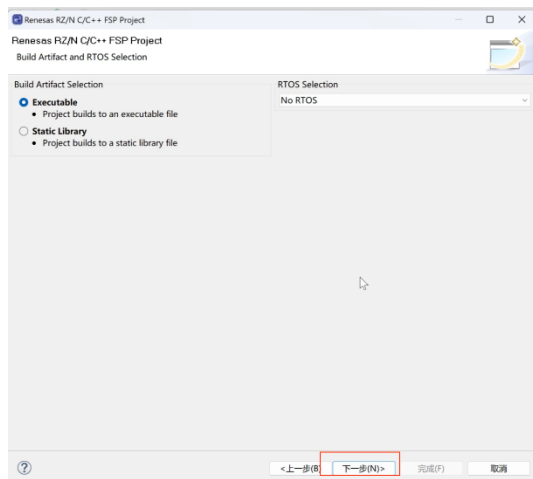


4. Boot 模式选择 RZN2L Custom User Board (XSPI0 x1 boot mode)

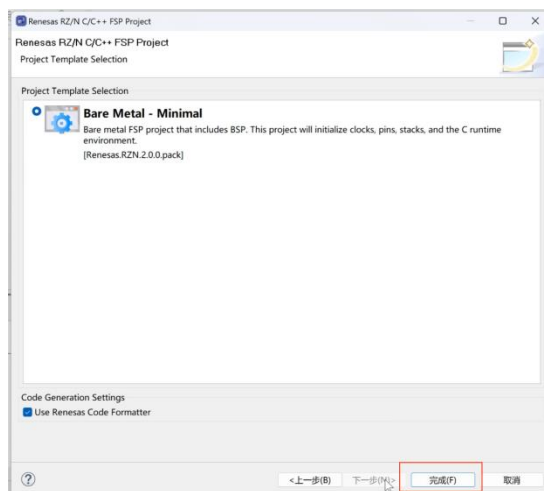
芯片型号: R9A07G084M04GBG



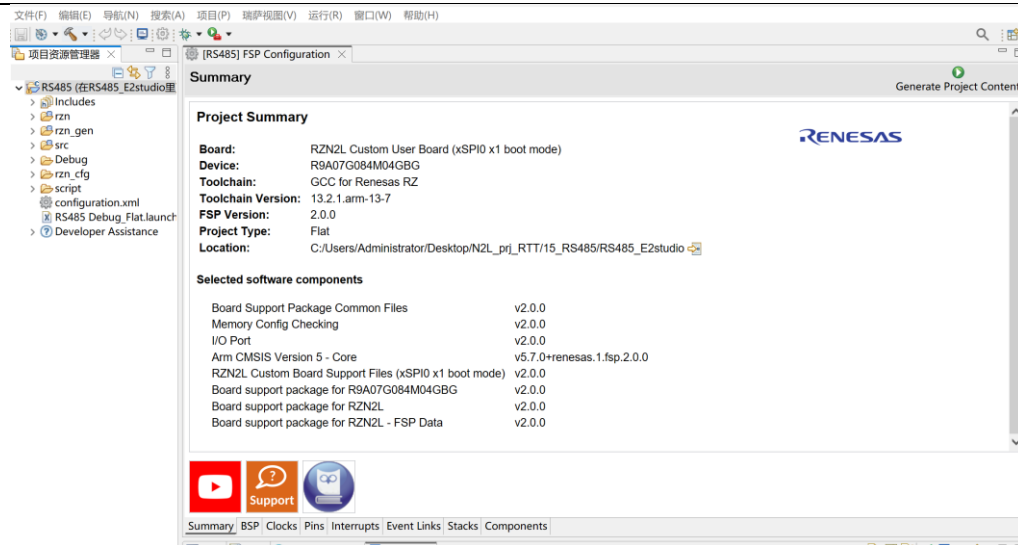
5. 直接下一步



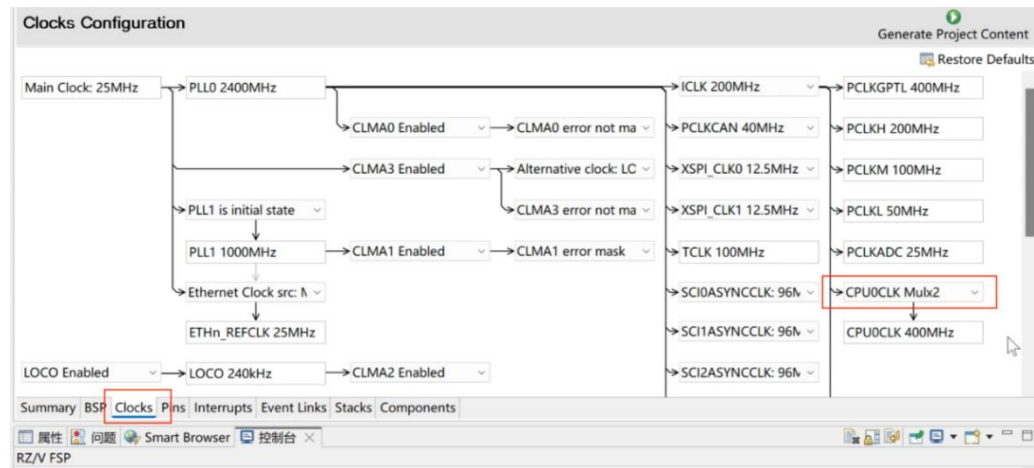
6. 点击 完成



7. 进入到工程界面



8. 设置时钟 CPU0CLK 400MHZ



3.2

9. 配置 XSPI0

The screenshot shows the 'Select Pin Configuration' window for R9A07G084M04GBG.pincfg. The 'Pin Selection' list on the left includes 'Connectivity:XSPI' and 'XSPI0'. The 'Pin Configuration' table on the right shows the following settings:

Name	Value	Lock	Link
Pin Group Selection	Mixed		
Operation Mode	Custom(1.8V)		
Input/Output			
XSPI0_CKN	None		
XSPI0_CKP	✓ P14_6		
XSPI0_CS0#	✓ P15_7		
XSPI0_CS1#	None		
XSPI0_DS	None		
XSPI0_ECS0#	None		
XSPI0_ECS1#	None		
XSPI0_INT0#	None		
XSPI0_INT1#	None		
XSPI0_IO0	✓ P14_7		
XSPI0_IO1	✓ P15_0		

Module name: XSPI0

10. P143 引脚配置

P143 设置输出模式，初始电平 0, 做为指示灯

The screenshot shows the 'Select Pin Configuration' window for R9A07G084M04GBG.pincfg. The 'Pin Selection' list on the left includes 'P14_3'. The 'Pin Configuration' table on the right shows the following settings:

Name	Value
Comment	
Mode	Output mode (Low & Not Into Input)
Pull up/down	None
Output Type	CMOS
Drive Capacity	Low
Region	Safety
Schmitt Trigger	None
Slew Rate	Slow
Input/Output	✓ GPIO

Module name: P14_3
Port Capabilities: ETHER, ETH0: ETH0_COL, ETH1: ETH1_COL

11. 配置 SCI5

The screenshot shows the 'Select Pin Configuration' window for R9A07G084M04GBG.pincfg. The 'Pin Selection' list on the left includes 'Connectivity:SCI' and 'SCI5'. The 'Pin Configuration' table on the right shows the following settings:

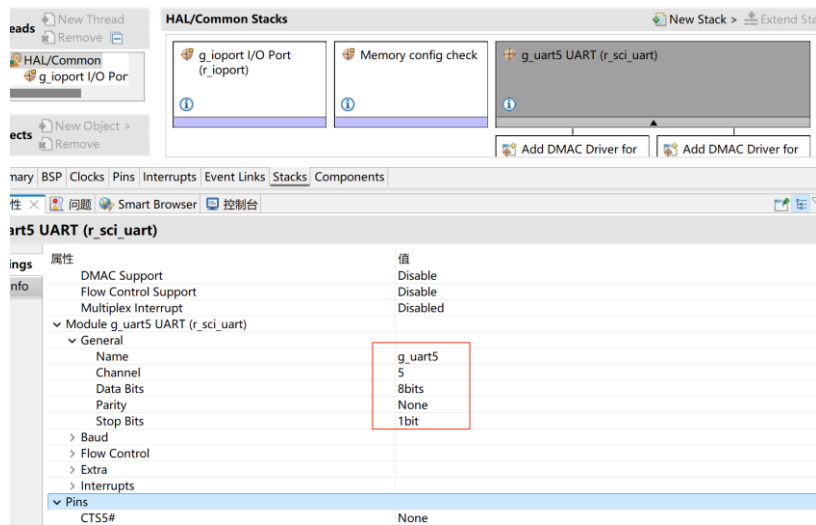
Name	Value	Lock	Link
Pin Group Selection	Mixed		
Operation Mode	Custom		
Input/Output			
CTS#	None		
RXS#	None		
RXS_MISO	✓ P21_2		
SCK	None		
SCL	None		
SDA	None		
TXD_MISO	✓ P21_3		

Module name: SCI5
Usage: In initial register value, when CCR0.TE bit is 0, and the terminal function is TXDn, the terminal outputs high

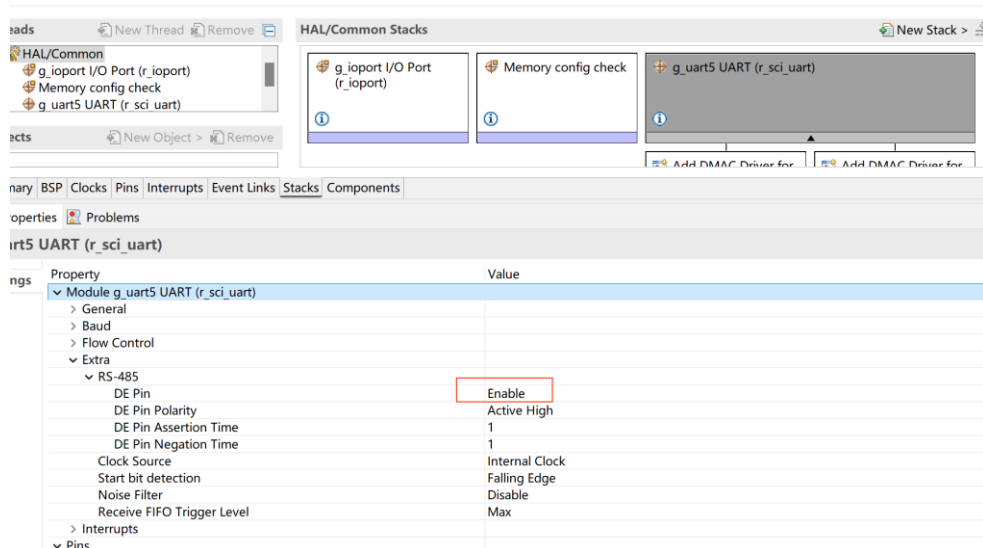
12. 添加 new Stack: UART

New Stack---Connectivity--UART(r_sci_uart)

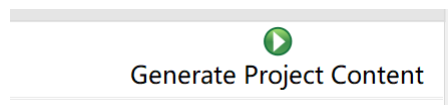
改为通道 5，波特率默认 115200



使能 RS485 DE pin



13. 点击: Generate Project Content 生成代码



14. 编写用户代码

➤ 编写用户代码: 一个工程两套代码, 由 send_board 定义编译哪套代码。

```

68     for (uint32_t i = 0; i < 20; i++)
69     {
70         g_src[i] = (uint8_t) ('A' + (i % 26));
71     }
72     while(1)
73     {
74         #if send_board
75             err = R_SCI_UART_Write(&g_uart5_ctrl, g_src, 10);
76             g_transfer_complete = 0;
77             while (!g_transfer_complete)
78             {
79             }
80             R_BSP_SoftwareDelay(1000, BSP_DELAY_UNITS_MILLISECONDS);
81         #else
82             err = R_SCI_UART_Read(&g_uart5_ctrl, g_dest, 10);
83             g_receive_complete = 0;
84             while (!g_receive_complete)
85             {
86             }
87             R_BSP_SoftwareDelay(800, BSP_DELAY_UNITS_MILLISECONDS);
88         #endif
89     }
90 }
91

```

发送数据板子代码

接收数据板子代码

- #define send_board 1 编译工程，下载代码到发送数据用的板子。板子自由运行，连续发送数据。
- #define send_board 0 编译工程，用于接收数据板子仿真运行。

3.3

- 下载代码到接收数据开发板

E2studio 在线仿真下载，全速运行

3.4

- 全速运行代码，停在断点，watch 窗口查看发送板发来的数据。

```

69     {
70         g_src[i] = (uint8_t) ('A' + (i % 26));
71     }
72     while(1)
73     {
74         #if send_board
75             err = R_SCI_UART_Write(&g_uart5_ctrl, g_src, 10);
76             g_transfer_complete = 0;
77             while (!g_transfer_complete)
78             {
79             }
80             R_BSP_SoftwareDelay(1000, BSP_DELAY_UNITS_MILLISECONDS);
81         #else
82             err = R_SCI_UART_Read(&g_uart5_ctrl, g_dest, 10);
83             g_receive_complete = 0;
84             while (!g_receive_complete)
85             {
86             }
87             R_BSP_SoftwareDelay(800, BSP_DELAY_UNITS_MILLISECONDS);
88         #endif
89     }
90 }
91

```

断点

表达式	类型	值	地址
g_dest	uint8_t	0x1760 <g_dest>	0x1...
g_src	uint8_t	0x41 (十六进制)	0x1...
g_dest	uint8_t	0x42 (十六进制)	0x1...
g_src	uint8_t	0x43 (十六进制)	0x1...
g_dest	uint8_t	0x44 (十六进制)	0x1...
g_src	uint8_t	0x45 (十六进制)	0x1...
g_dest	uint8_t	0x46 (十六进制)	0x1...
g_src	uint8_t	0x47 (十六进制)	0x1...
g_dest	uint8_t	0x48 (十六进制)	0x1...
g_src	uint8_t	0x49 (十六进制)	0x1...
g_dest	uint8_t	0x4a (十六进制)	0x1...
g_src	uint8_t	0x0 (十六进制)	0x1...
g_dest	uint8_t	0x0 (十六进制)	0x1...
g_src	uint8_t	0x0 (十六进制)	0x1...
g_dest	uint8_t	0x0 (十六进制)	0x1...
g_src	uint8_t	0x0 (十六进制)	0x1...
g_dest	uint8_t	0x0 (十六进制)	0x1...
g_src	uint8_t	0x0 (十六进制)	0x1...
g_dest	uint8_t	0x0 (十六进制)	0x1...
g_src	uint8_t	0x0 (十六进制)	0x1...
g_dest	uint8_t	0x0 (十六进制)	0x1...
g_src	uint8_t	0x0 (十六进制)	0x1...
g_dest	uint8_t	0x0 (十六进制)	0x1...
g_src	uint8_t	0x0 (十六进制)	0x1...

本节完