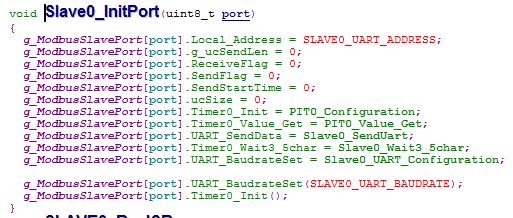
OpenModbus移植文档

1. 概述

本文主要介绍如何使用OpenModbus，并移植OpenModbus到不同的处理器平台。

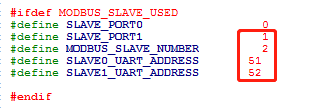
1. 如何添加Modbus Slave

OpenModbus使用了Modbus\_Port结构体封装了所有需要的函数，所以只要初始化它所需的几个指针函数，以及从站配置信息即可运行Modbus从站，全局变量g\_ModbusSlavePort在成功初始化后记录相关的配置及指针函数，每个port代表一路实际的Modbus。需要说明的是所有Modbus主/从站share了同一个Timer(PIT0)，所以用户仅需要完成Slave0\_SendUart, Slave0\_Uart\_Configuration, Slave0\_Wait3\_5char，这三个函数即可。

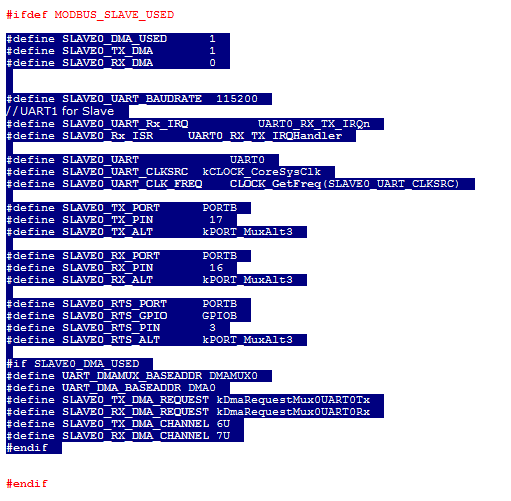


具体步骤如下：

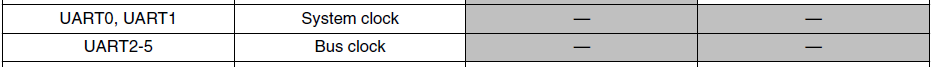
1. 修改ModbusUserConfig.h中的MODBUS\_SLAVE\_NUMBER为2，并添加SLAVE\_PORT1的宏，并设置从站地址为52

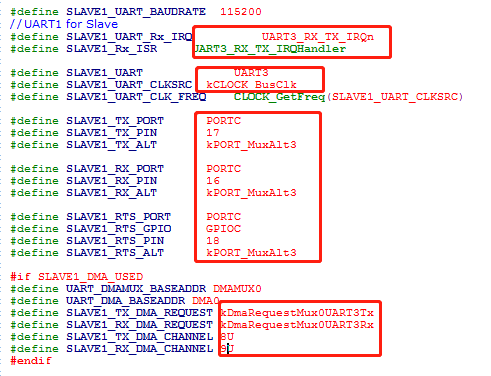


1. 在Modbus\_Porting.h中，复制如下配置并修改SLAVE0->SLAVE1

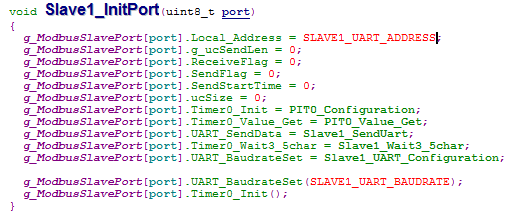


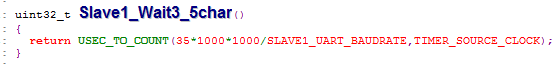
并根据硬件设计的UART口来配置SLAVE1的Pin以及该UART所选的时钟源（需要注意有些UART的时钟源并不相同），如果需要使用DMA进行发送，需要注意修改DMA的request源也要根据UART口来进行修改，下面以K64的UART2为例：





1. 在Modbus\_Porting.c中Copy一份 #ifdef MODBUS\_SLAVE\_USED中的Slave0\_xxxx函数，并将所有Slave0修改为Slave1，如果需要使用DMA，则需要复制DMA相关变量，并替换Slave0->Slave1.



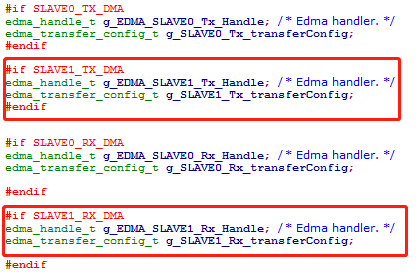


void SLAVE1\_Rx\_ISR()

void Slave1\_UART\_Configuration(uint32\_t buadrate)

void Slave1\_SendUart(uint8\_t \*pbyData, uint16\_t uCount, uint8\_t port)

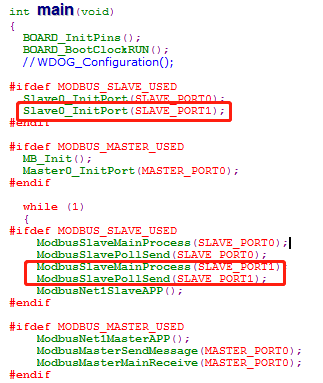
完成以上5个函数的修改。



1. s\_SlaveuartBase添加SLAVE1\_UART变量



1. mian函数中添加如下代码：



1. 如何添加Modbus Master

类似于Modbus Slave，首先完成相关结构体内的函数，之后需要根据总线内接入的从站模块信息编写ModbusNet1MasterAPP函数。

1. 移植详解

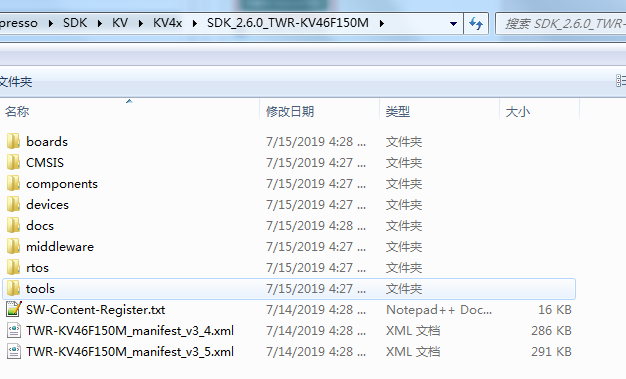
由于OpenModbus默认使用K64作为测试平台，所以非常容易将其移植到NXP其他系列单片机，这里以移植至Kinetis KV46为例，仅需要如下几步即可：

1. 下载对应平台SDK(<https://mcuxpresso.nxp.com>)

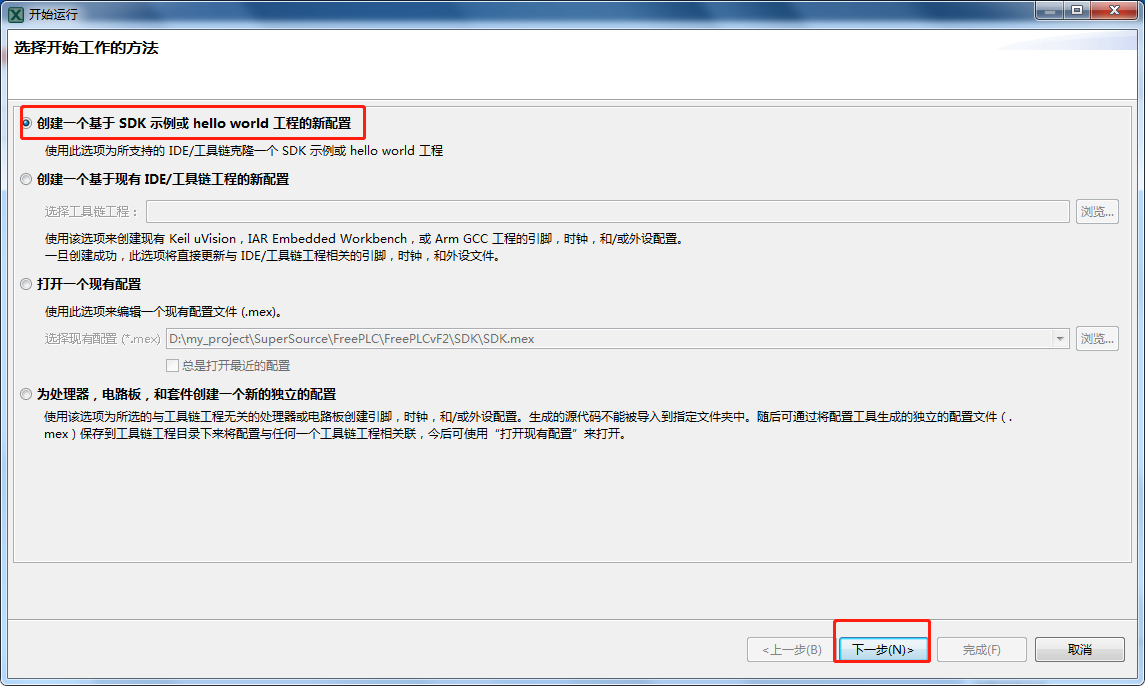


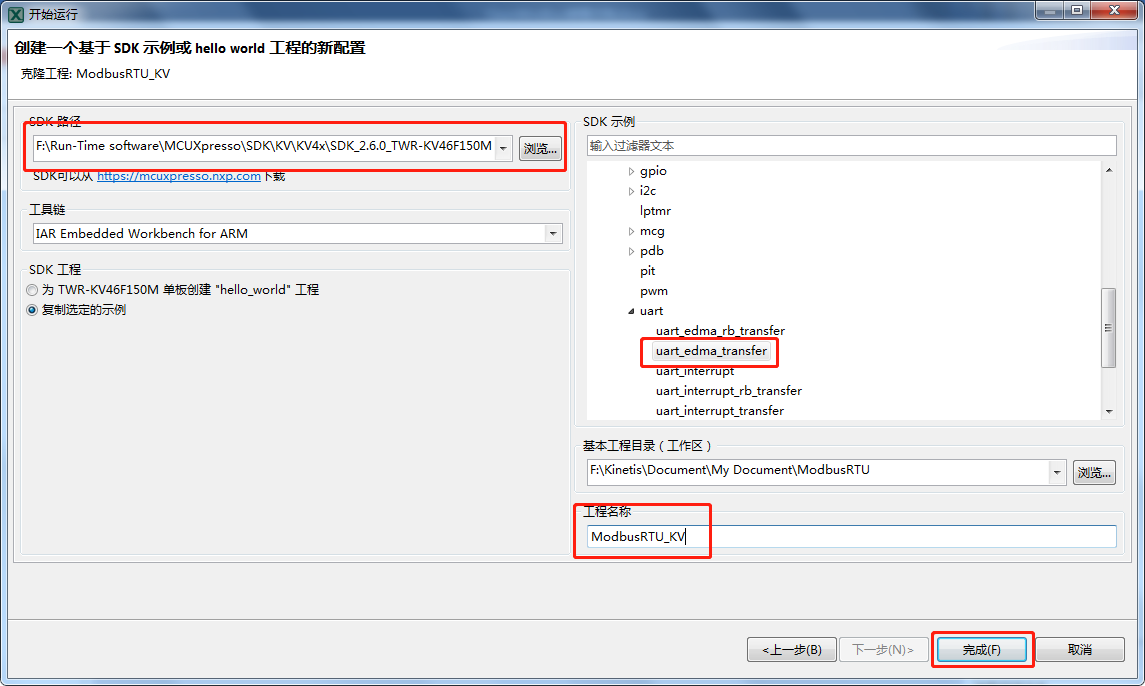


1. 解压下载到的压缩包SDK\_2.6.0\_TWR-KV46F150M.zip

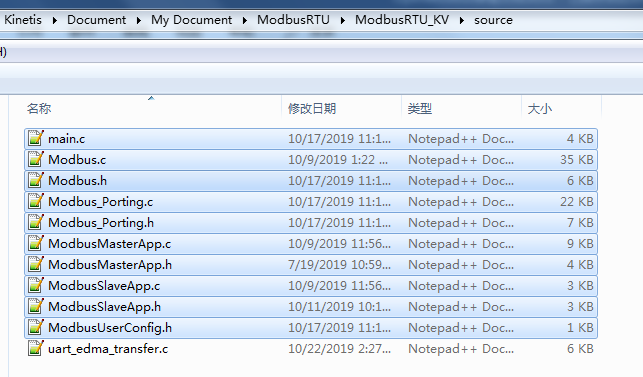


1. 通过MCUXpresso Config Tools v6工具Clone SDK\_2.6.0\_TWR-KV46F150M\boards\twrkv46f150m\driver\_examples\uart\edma\_transfer这个工程并rename为ModbusRTU

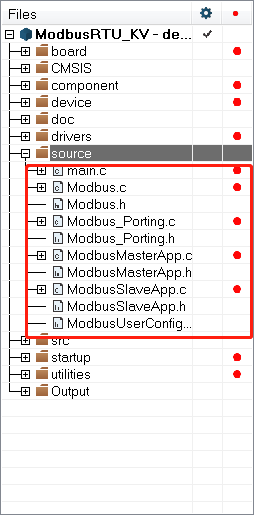




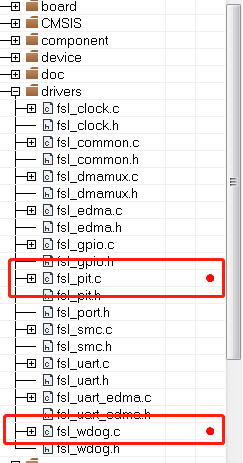
1. Copy OpenModbus\source\所有文件至新建的ModbusRTU\_KV\source中并覆盖



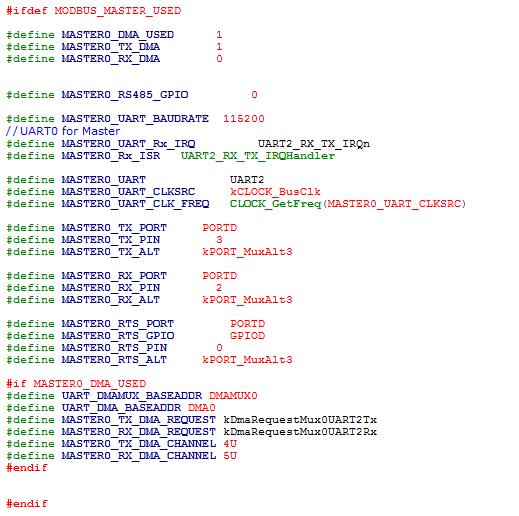
1. Copy SDK\_2.6.0\_TWR-KV46F150M\devices\MKV46F16\drivers目录下所有文件到ModbusRTU\_KV\drivers并覆盖。
2. 打开ModbusRTU\_KV.eww工程，删除uart\_edma\_transfer.c，并添加之前copy的所有文件



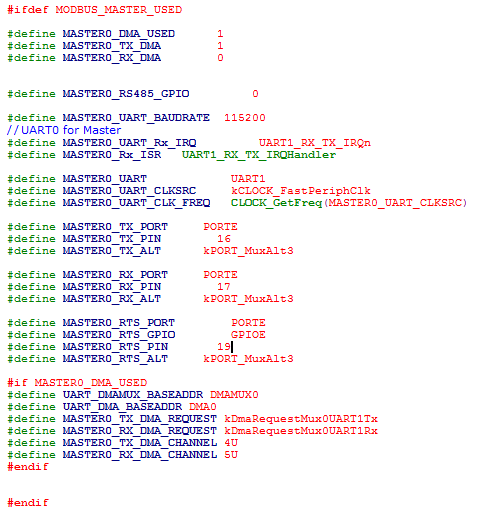
1. Device目录下添加fsl\_pit.c/h以及fsl\_wdog.c/h



1. 修改Modbus\_Porting.h文件



这里需要根据KV相关UART的配置，这里以Modbus Master为例，把UART2改为UART1，选择kCLOCK\_FastPeriphClk为UART1的时钟源，修改TX、RX、RTS(RS485 only)的pin定义，修改DMA的Request为kDmaRequestMux0UART1T/Rx



1. 重新编译即可，默认使用UART TX DMA发送，中断RX接收