

YTMicro MCAL应用 - Spi模块配置及实例

版本

Config Tool Version: 3.0.0

ME0 MCAL version: 2.3.0

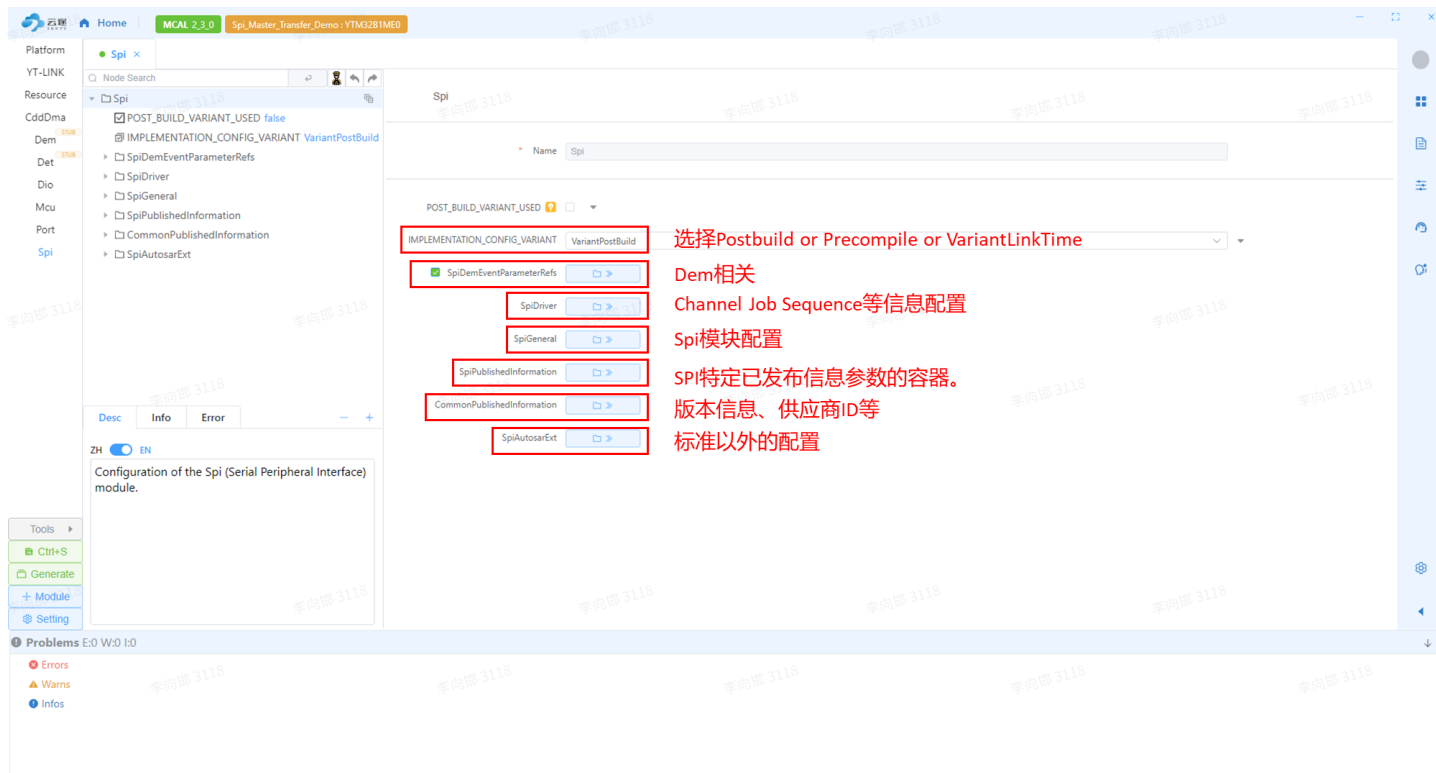
前言

1. Spi一般依赖Mcu、Platform和Port模块，Mcu模块里面需要配置外设时钟，Platform里面可以配置外设中断，Port配置使用到的端口资源，如果选择DMA方式则需要另外添加CddDMA模块。
2. Port、Mcu和Platform的配置参考《MCAL应用 - 工程建立及工具通用设置》和《MCAL应用 - Mcu模块配置及实例》文档，Port的配置参考《MCAL应用 - Port模块配置及实例》，CddDMA的配置参考《MCAL应用 - CddDMA模块配置及实例》。
3. 配置中包含AutoSar规范标准选项和云途自定义选项，左边树选框中粗体字为AutoSar标准选项，细体字为云途自定义选项。
4. MCAL应用_Spi模块配置及实例（一）主要说明SPI基本配置，以及作为主机节点的应用实例。

1. 基础模块配置介绍

1.1 框架说明

此处主要介绍了整个SPI的框架，需要我们着重关注的是SpiDriver和SPIGeneral两部分，下文会详细介绍。



1.2 SpiDemEventParameterRefs



1.3 SpiDriver

此处为SPI的核心配置，涉及Sequence、Job、Channel以及需要通信设备的相关配置信息。



Tips:

AUTOSAR 提供了三种SPI的抽象。Channel Job Sequence。

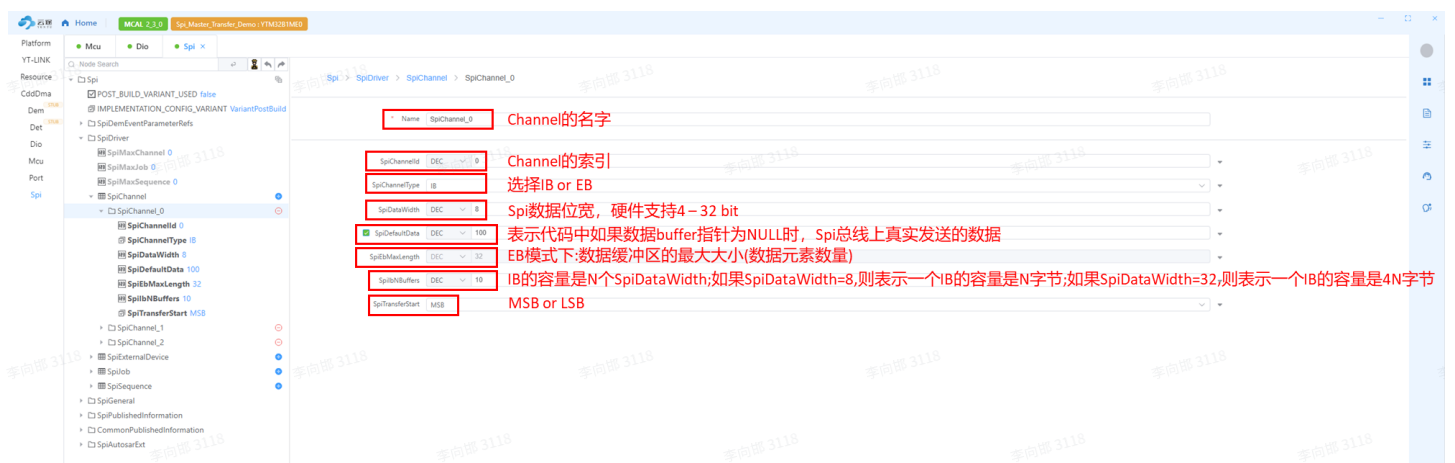
一个Job 代表了一次完整的SPI通信，一个Job可以由一个或者多个Channel组成，Channel用来装载SPI的发送与接收数据。

一个Sequence可以由一个或者多个Job,可以将一个操作序列抽象成一个Sequence。

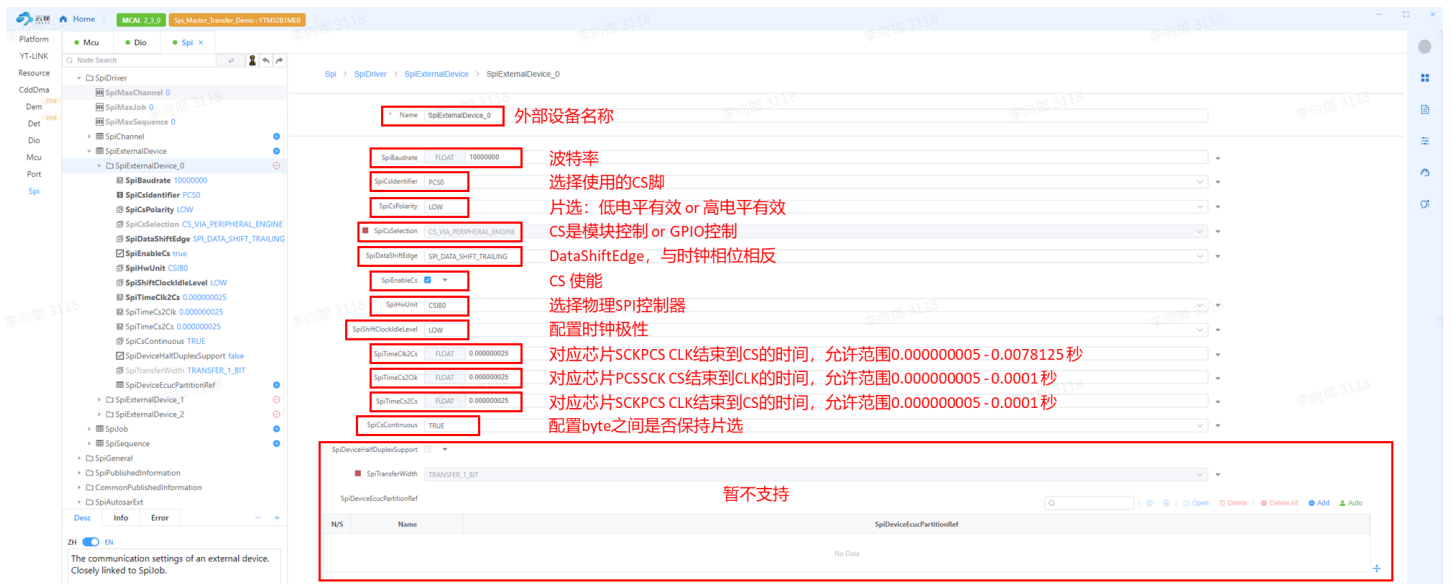
Tips:

实际应用中，常常一个Sequence下挂一个Job，一个Job下挂一个Channel。

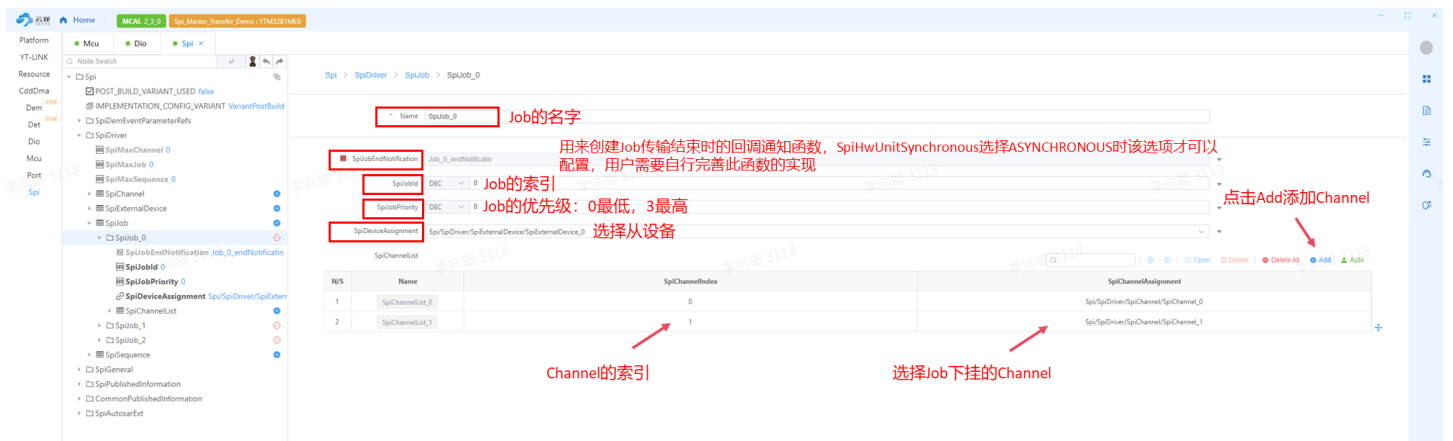
1.3.1 SpiChannel



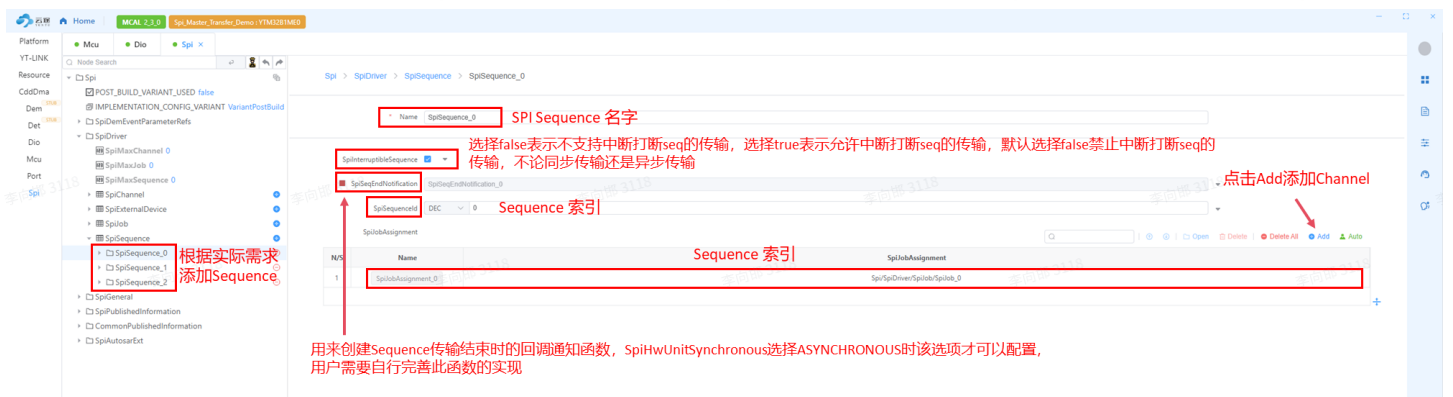
1.3.2 SpiExternalDevice



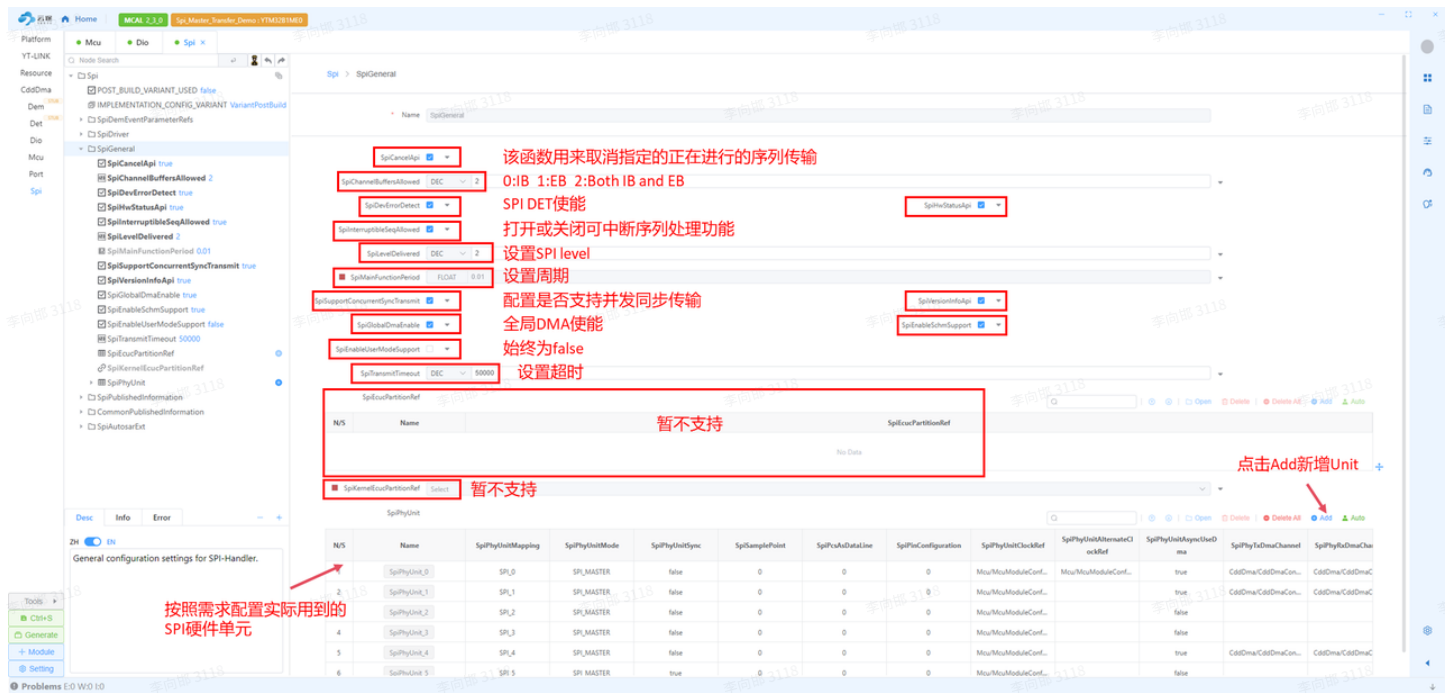
1.3.3 SpiJob



1.3.4 SpiSequence



1.4 SpiGeneral



Tips:

IB: 静态分配 EB: 可以静态分配也可以动态分配，取决于应用场景

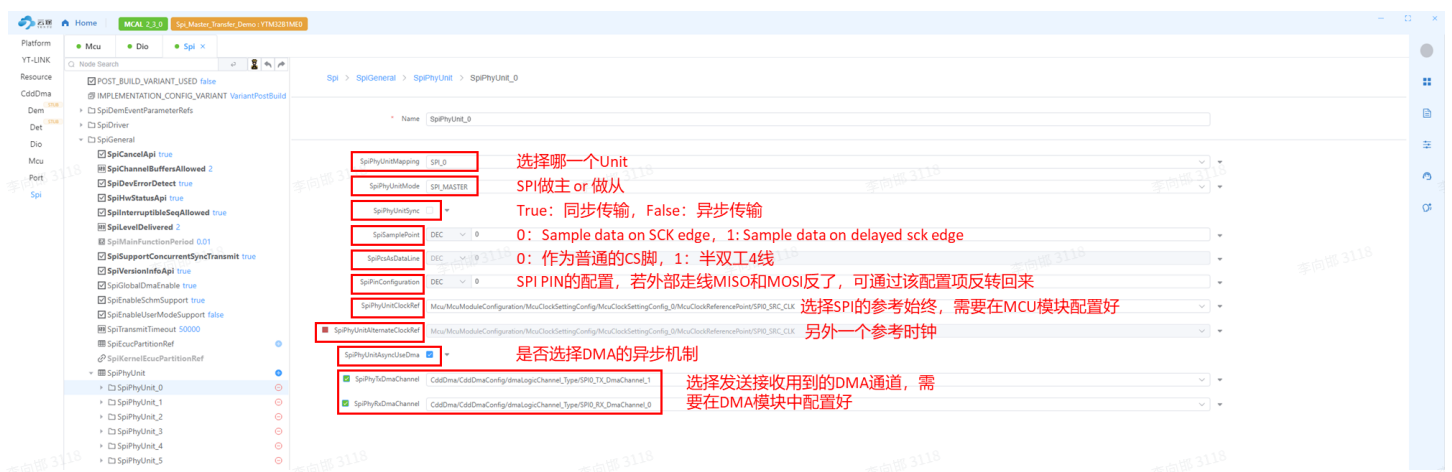
IB类型buffer大小是固定的，EB类型buffer可以通过API进行设置。

Tips:

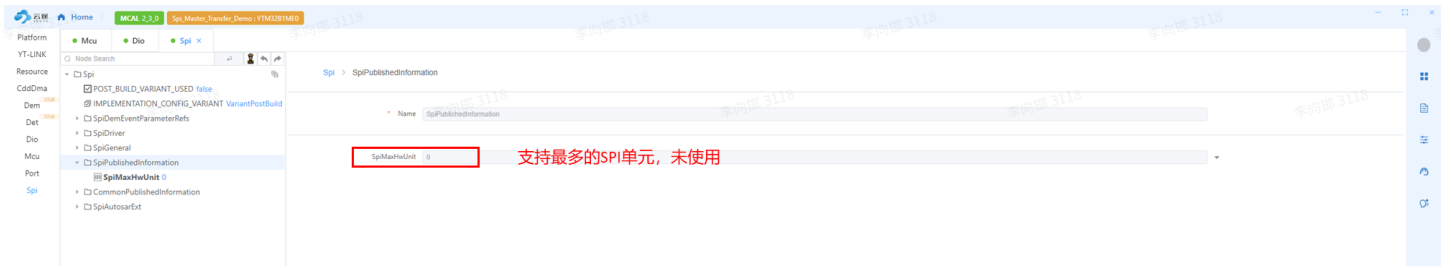
SpiDataWidth: 支持8bit - 32bit

如果DataWidth设置成了24Bit，这种情况下，发送只能发送24bit，忽略高4位，接收的时候也只接收24bit，线上的波形是按照24bit进行通信，软件层面上需要对buffer进行处理。

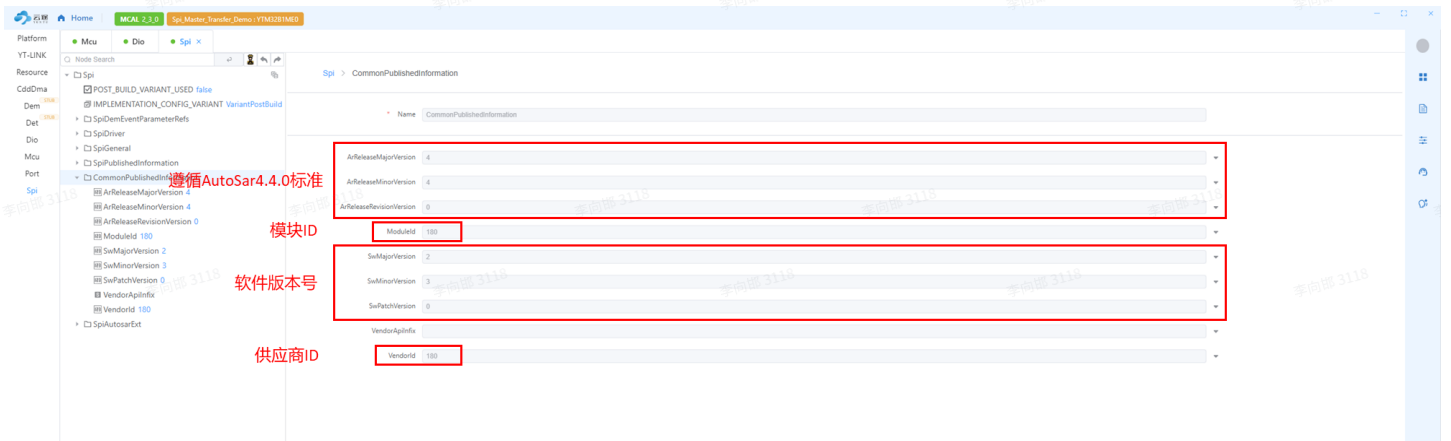
1.5 SpiPhyUnit



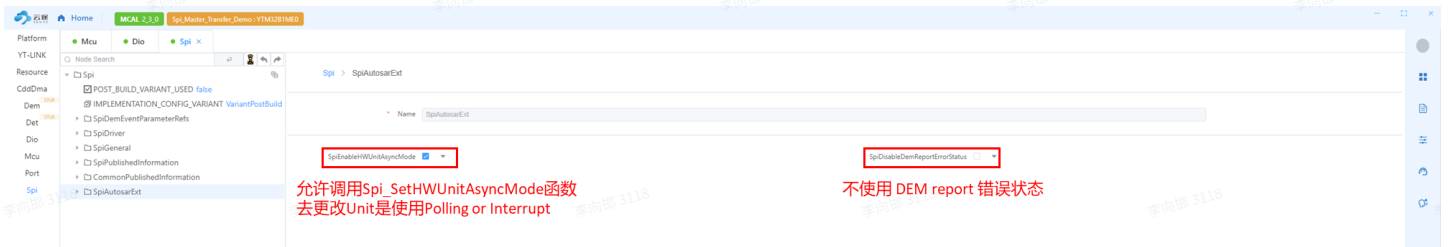
1.6 SpiPublishedInformation



1.7 CommonPublishedInformation



1.8 SpiAutosarExt



1.9 一图总结SpiDriver、SpiGeneral配置中的级联关系

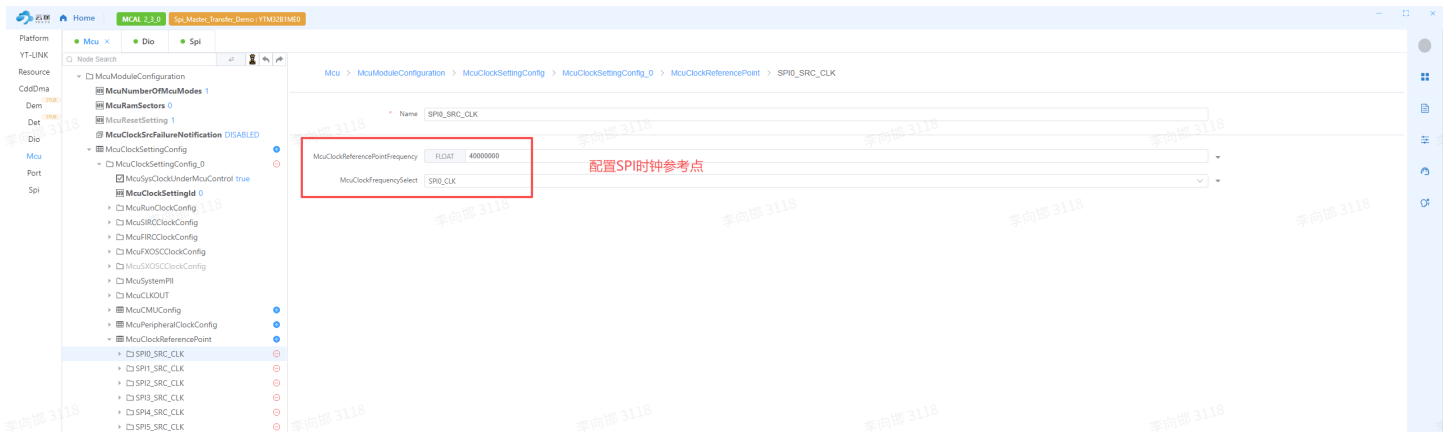
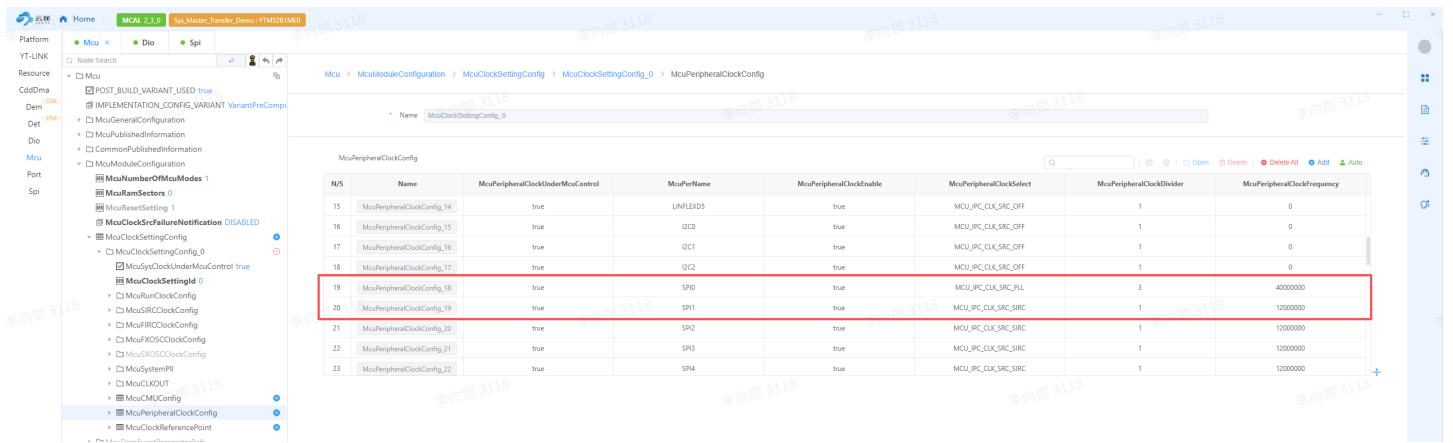


2. 示例

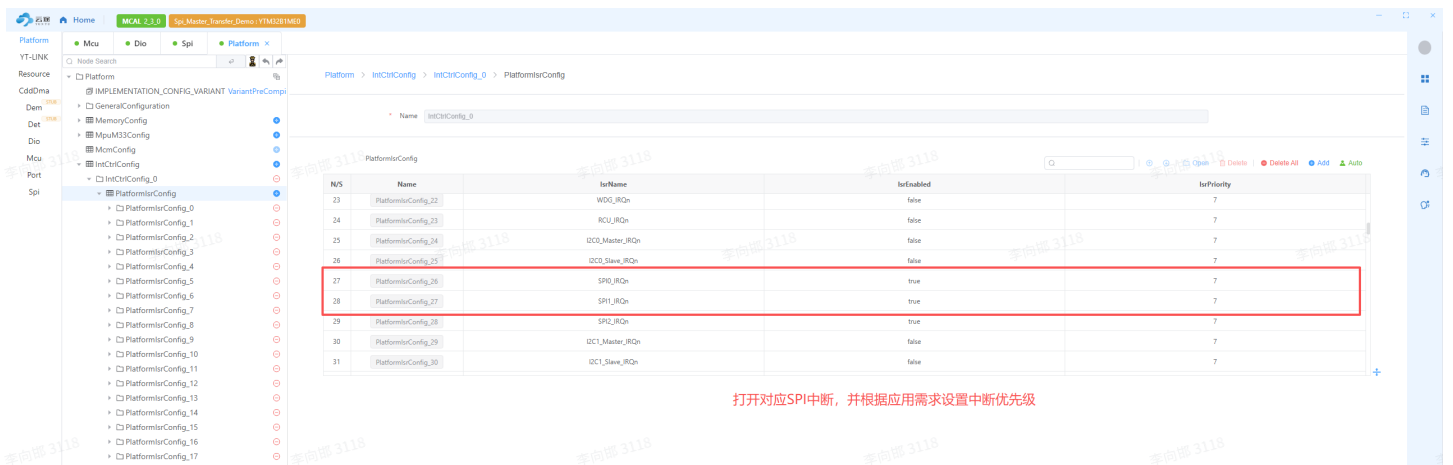
2.1 配置步骤

基于ME0 Demo板，配置SPI0和SPI1两路SPI

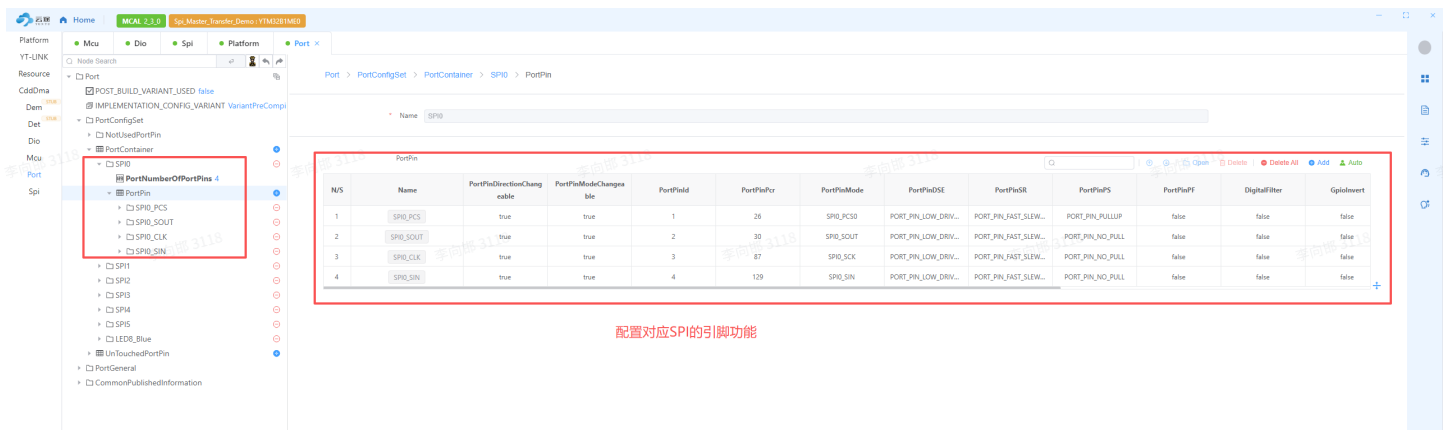
Step1: 时钟配置



Step2: 中断配置

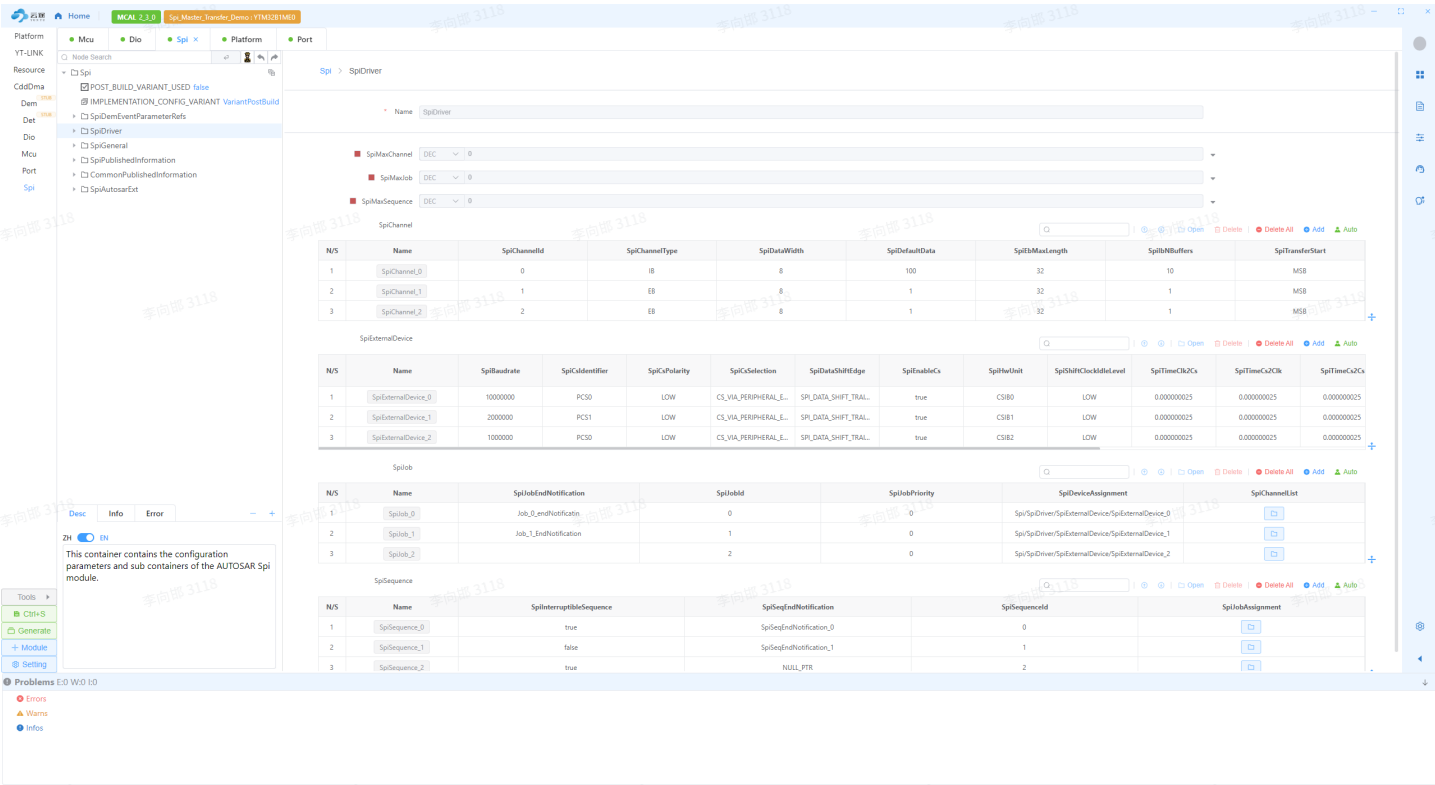


Step3: Port配置



Step4: 配置SPI模块

根据需求参考上述“1.基础模块配置介绍”进行配置



2.2 API说明

2.2.1 Spi_Init

void Spi_Init(const Spi_ConfigType* ConfigPtr)

- 初始化所有ConfigPtr相关寄存器
- 定义ConfigPtr相关的默认值
- 设置SPI状态为“IDLE”，job、sequence的状态设为“ok”
- 对于Level2，异步模式设为polling，禁用spi相关中断

2.2.2 Spi_DelInit

Std_ReturnType Spi_DelInit(void)

- 如果驱动状态不是BUSY，将spi外设置为复位上电后的状态
- 如果驱动状态是BUSY，请求被拒绝
- 去初始化后，模块的状态是UINIT

2.2.3 Spi_WriteIB

Std_ReturnType Spi_WriteIB(Spi_ChannelType Channel, const Spi_DataBufferType* DataBufferPtr)

- 返回值：E_OK：该命令被接受，E_NOT_OK：该命令被拒绝。

- 将入参数据写入对应channel的内部缓存
- 如果入参DataBufferPtr为NULL，将传输默认值

2.2.4 Spi_AsyncTransmit

Std_ReturnType Spi_AsyncTransmit(Spi_SequenceType Sequence)

- 返回值：E_OK：该命令被接受，E_NOT_OK：该命令被拒绝。
- 在SPI总线上传输数据
- 驱动状态设为“BUSY”，Sequence状态设为“PENDING”后函数返回
- Job状态设为SPI_JOB_QUEUED

2.2.5 Spi_ReadIB

Std_ReturnType Spi_ReadIB(Spi_ChannelType Channel, Spi_DataBufferType* DataBufferPointer)

- 返回值：E_OK：该命令被接受，E_NOT_OK：该命令被拒绝。
- 同步读取一条或多条数据
- 应该在Transmit后调用，以获取相关数据

2.2.6 Spi_SyncTransmit

Std_ReturnType Spi_SyncTransmit(Spi_SequenceType Sequence)

- 返回值：E_OK：该命令被接受，E_NOT_OK：该命令被拒绝。
- 在SPI总线上传输数据
- 驱动状态设为“BUSY”
- Sequence、Job状态设为“PENDING”
- 此时如果有其他Sequence正在传输，应该返回 E_NOT_OK，并上报错误。

2.2.7 Spi_GetStatus

Spi_StatusType Spi_GetStatus(void)

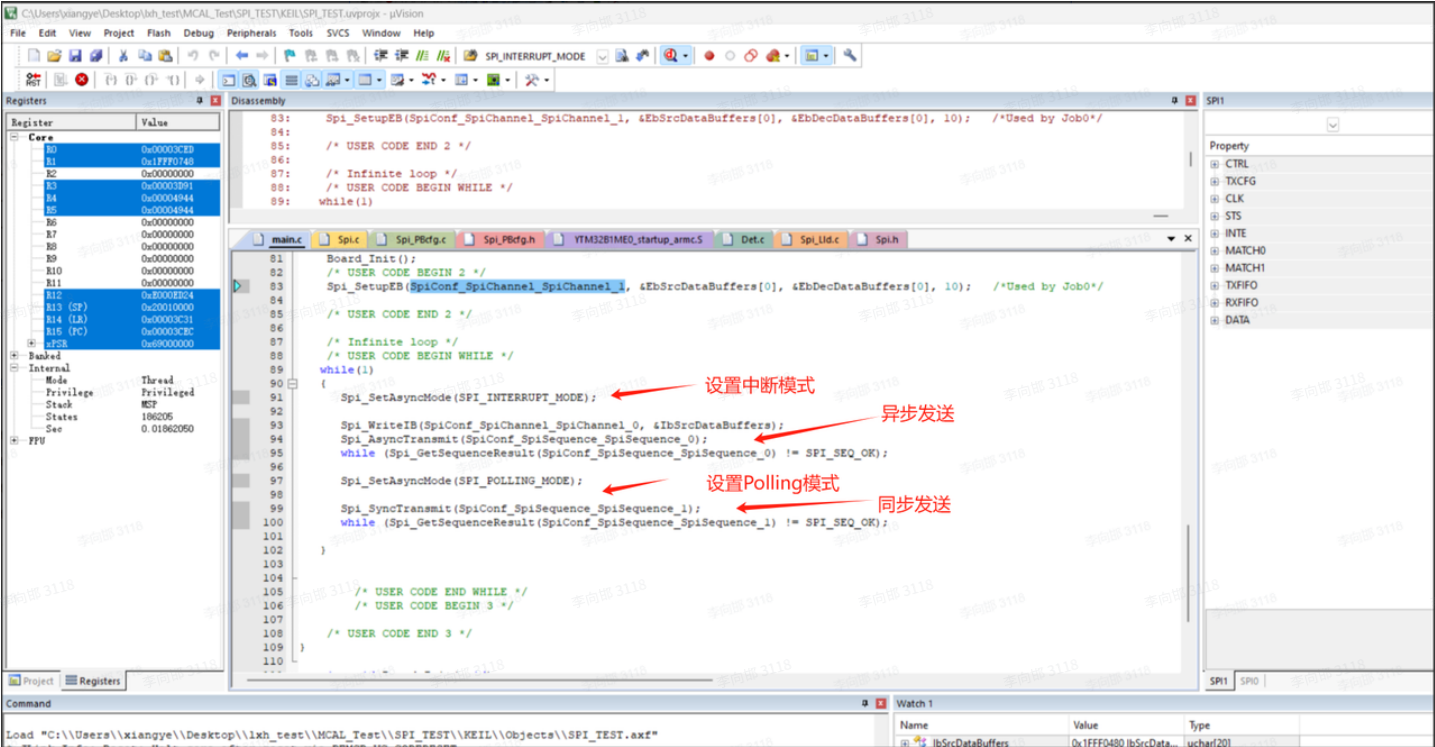
- 获取驱动状态

3. 输出现象

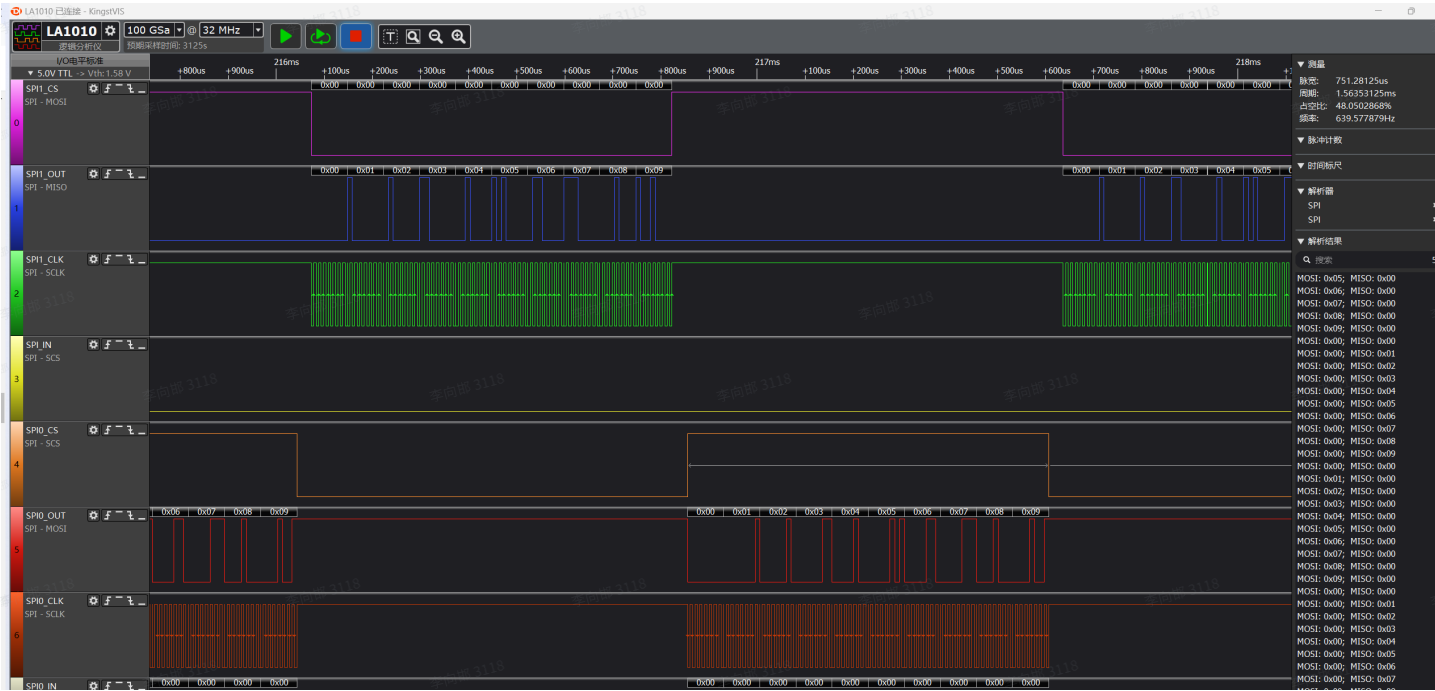
SPI0和SPI1都作为主机发送

SPI0异步发送 (0x00,0x01.0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09)

SPI1同步发送 (0x00,0x01.0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09)



线上抓到的数据和代码中发送的数据一致。



文档历史

版本号	日期	修订记录
V1.0	2024.01.04	初始版本

V2.0	2026.05.11	1. 更新配置工具YCT版本为3.0.0及MCAL版本为2.3.0; 2. 增加SpiDriver层级关联介绍
------	------------	---