

MCAL应用_Spi模块配置及实例（一）

版本

Config Tool Version: 1.10.0

ME0 MCAL version: 1.3.1

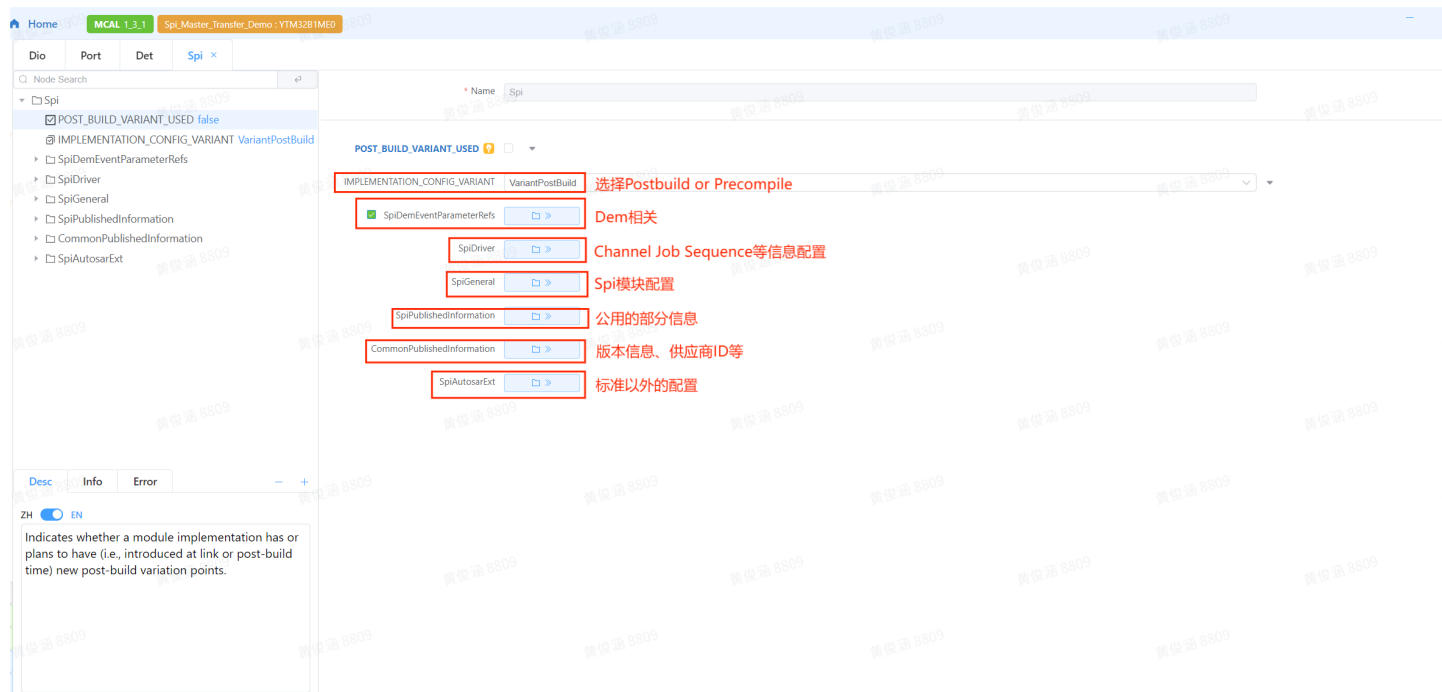
前言

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. 如何配置SPI模块

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。

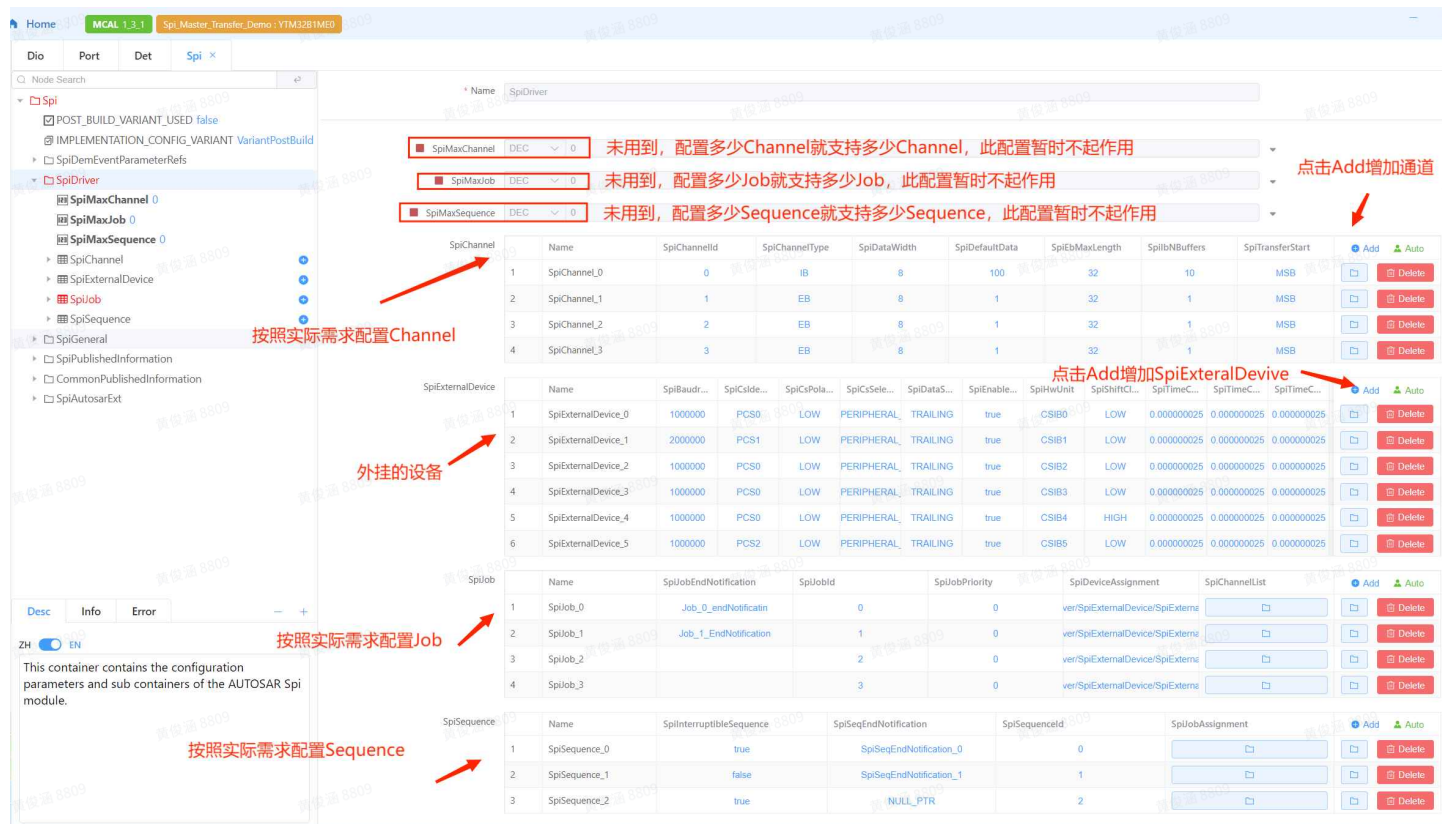


1.2 SpiDemEventParameterRefs



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

此处配置SPI的Channel信息，此部分是AUTOSAR标准定义的配置项，详细说明可参考AUTOSAR_SWS_SPIHandlerDriver文档。



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。

1.3.2 SpiExternalDevice

此处配置ExternalDevice，根据需要通信设备的通信协议配置即可。



Tips:

SpiDataShiftEdge在2.0.0版本以后与时钟相位相同。

Tips: 芯片支持在一定范围内配置时钟信号与PCS的间隙时间。

SpiTimeCs2Cs: 此部分是依赖Spi模块的CLK Register的FMDLY实现，

当SPI配置成Continuous Frame: 此部分表示相邻Frame的CLK之间的时间；

当SPI配置成非Continuous Frame: 此部分表示相邻Frame中CS空闲的时间；

SpiTimeClk2Cs: 此部分是依赖Spi模块的CLK Register的PCSSCK实现，

表示Clk结束后到PCS空闲之间的时间；

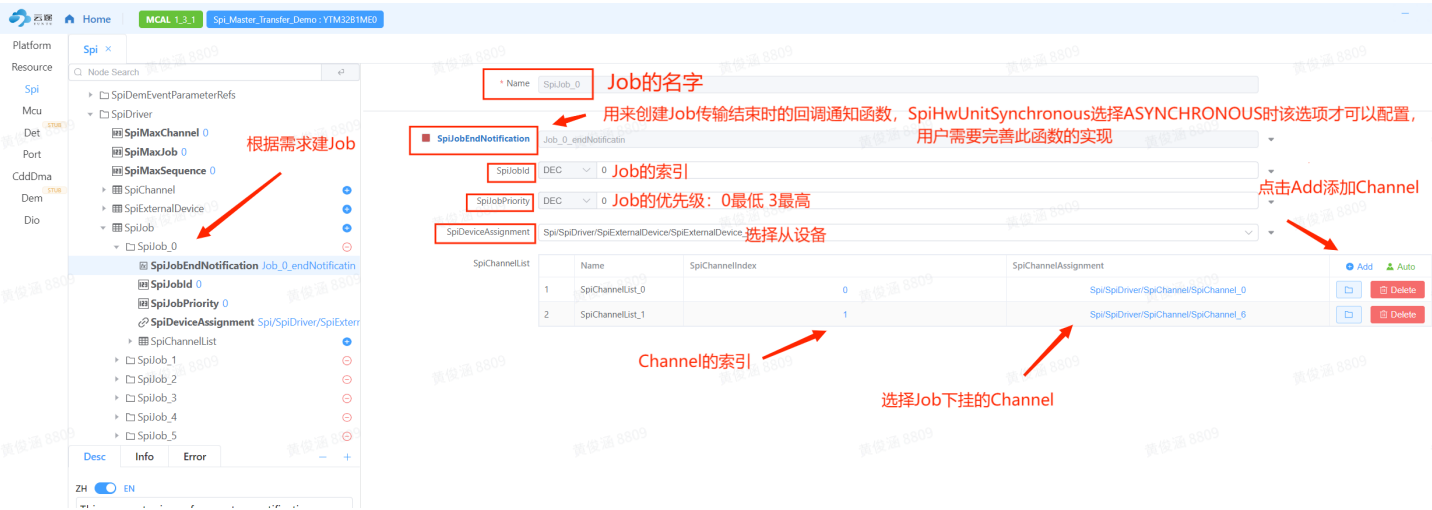
SpiTimeCs2Clk: 此部分是依赖Spi模块的CLK Register的SCKPCS实现，

表示PCS激活到产生Clk信号之间的时间；

Note：如果对于PCS到时钟有更长时间的要求（某些外设需要产生PC信号以后xxms后再建立通信），则建议使用GPIO作为片选信号。

1.3.3 SpiJob

此处配置SPI的Job信息，此部分是AutoSAR标准定义的配置项，参照下图说明配置即可。详细说明可参考AUTOSAR_SWS_SPIHandlerDriver文档。



1.3.4 SpiSequence

此处配置SPI的Sequence信息，此部分是AutoSAR标准定义的配置项，参照下图说明配置即可，更多说明可直接阅读AUTOSAR_SWS_SPIHandlerDriver文档。



1.4 SpiGeneral

此处是配置SPI的通用配置以及芯片的SPI模块的配置

Home MCAL 1.3.1 Spi Master Transfer Demo - YIM321ME0

Dio Port Det Spi

Node Search

Spi

- POST_BUILD_VARIANT_USED false
- IMPLEMENTATION_CONFIG_VARIANT VariantPostBuild
- SpiDemEventParameterRefs
- SpiDriver
 - SpiGeneral
 - SpiCancelApi true 该函数用来取消指定的正在进行的序列传输
 - SpiChannelBuffersAllowed DEC 2 0:IB 1:EB 2:Both IB and EB
 - SpiDevErrorDetect true SPI DET使能
 - SpiInterruptibleSeqAllowed true 打开或关闭可中断序列处理功能
 - SpiLevelDelivered DEC 2 设置SPI level
 - SpiMainFunctionPeriod FLOAT 0.01 设置周期
 - SpiSupportConcurrentSyncTransmit true 配置是否支持并发同步传输
 - SpiGlobalDmaEnable true 全局DMA使能
 - SpiEnableUserModeSupport true 时钟为false
 - SpiTransmitTimeout DEC 50000 设置超时
 - SpiEcucPartitionRef Name SpiEcucPartitionRef 暂不支持
 - SpiKernelEcucPartitionRef Select 暂不支持
 - SpiVersionInfoApi true SPI 版本信息函数 API使能
 - SpiEnableSchmSupport true SPI 临界区保护使能
- SpiPhyUnit
 - SpiPhyUnit_0
 - SpiPhyUnit_1
 - SpiPhyUnit_2
 - SpiPhyUnit_3
 - SpiPhyUnit_4
 - SpiPhyUnit_5

General configuration settings for SPI-Handler.

按照需求配置实际用到的SPI硬件单元

点击Add新增Unit

Name	SpiPhyUnit...	SpiPhyUn...	SpiPhyUn...	SpiSamp...	SpiPcsAs...	SpiPinCo...	SpiPhyUn...	SpiPhyUn...	SpiPhyUn...	SpiPhyTr...	SpiPhyRx...	Add	Delete
SpiPhyUnit_0	SPI_0	SPI_MASTER	false	0	0	0	McuClockSetti	McuClockSetti	true	.logicChannel_LogicChannel_1			
SpiPhyUnit_1	SPI_1	SPI_MASTER	false	0	0	0	McuClockSetti	McuClockSetti	true	.logicChannel_LogicChannel_1			
SpiPhyUnit_2	SPI_2	SPI_MASTER	false	0	0	0	McuClockSetti	McuClockSetti	false				
SpiPhyUnit_3	SPI_3	SPI_MASTER	false	0	0	0	McuClockSetti	McuClockSetti	false				
SpiPhyUnit_4	SPI_4	SPI_MASTER	false	0	0	0	McuClockSetti	McuClockSetti	true	LogicChannel_LogicChannel_1			
SpiPhyUnit_5	SPI_5	SPI_MASTER	true	0	0	0	McuClockSetti	McuClockSetti	false				

Tips:

Spi Delivered Level	Sync Mode	Async Mode		Spi_SetAsyncMod Api	Sync Mode Concurrent Fetaute
		Polling Mode	Interrupt Mode		
LEVEL 0	√	×	×	×	√
LEVEL 1	×	√	×	×	×
LEVEL 2	√	√(default)	√	√	√

Tips:

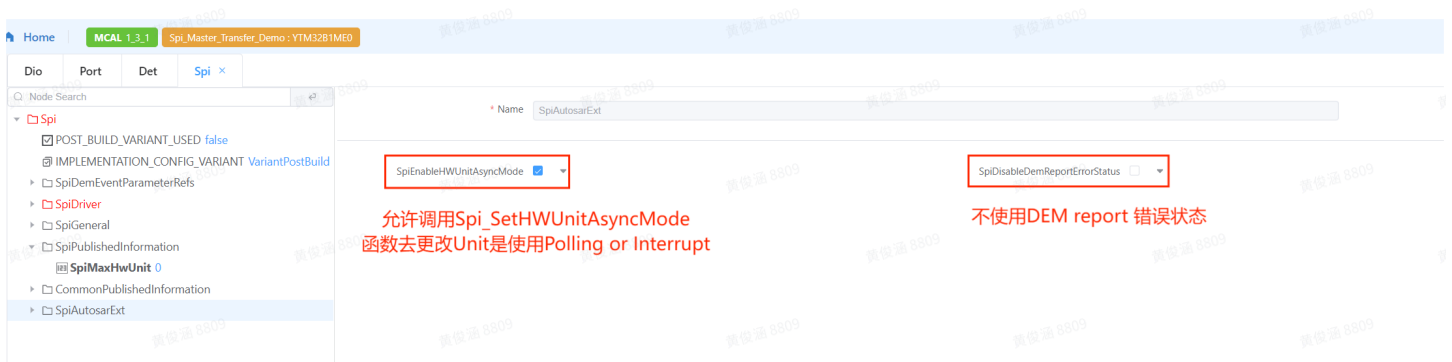
注意设置为Level2时，默认下使用的是轮询模式，需要用户自己调用Spi_SetAsyncMode切换到中断模式。

1.4.1 SpiPhyUnit1

此处是配置芯片的SPI模块。



1.5 SpiAutosarExt



2. SPI模块应用实例

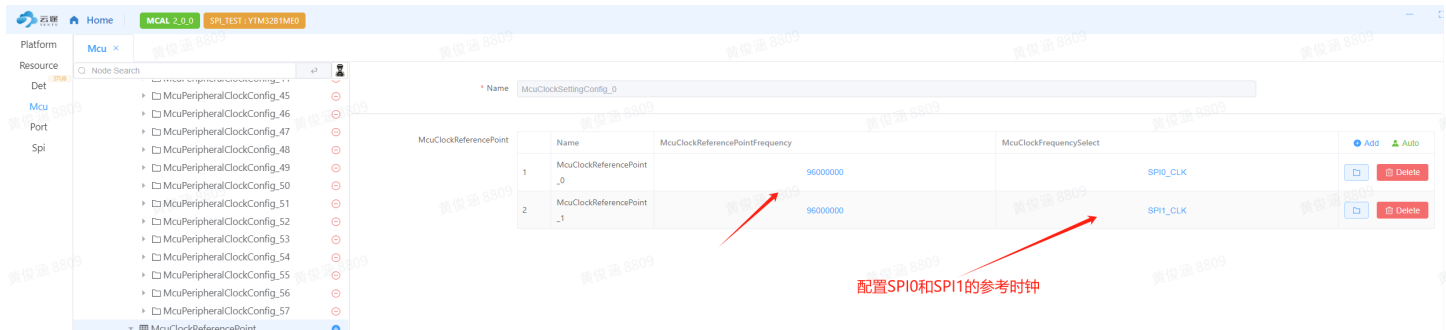
2.1 应用场景

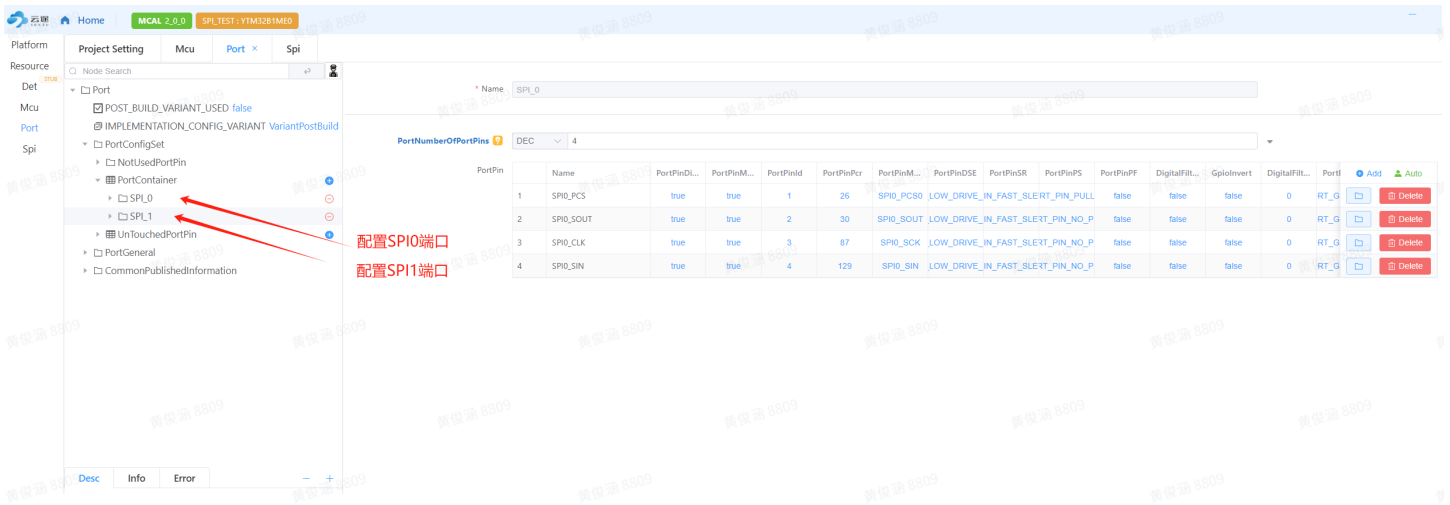
- 基于ME0 Demo板，配置SPI0和SPI1两路SPI作为主机发送。
- 配置PORT口

PTA15 -- SPI0_PCS0	PTB18 -- SPI1_PCS1
PTA30 -- SPI0_SOUT	PTA27 -- SPI1_SOUT
PTC23 -- SPI0_CLK	PTA28 -- SPI1_CLK
PTE01 -- SPI0_SIN	PTA29 -- SPI0_SIN
- 配置100K波特率。
- 分别使用异步中断和异步Polling。
- 逻辑分析仪抓取发送数据 (0x00,0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09)

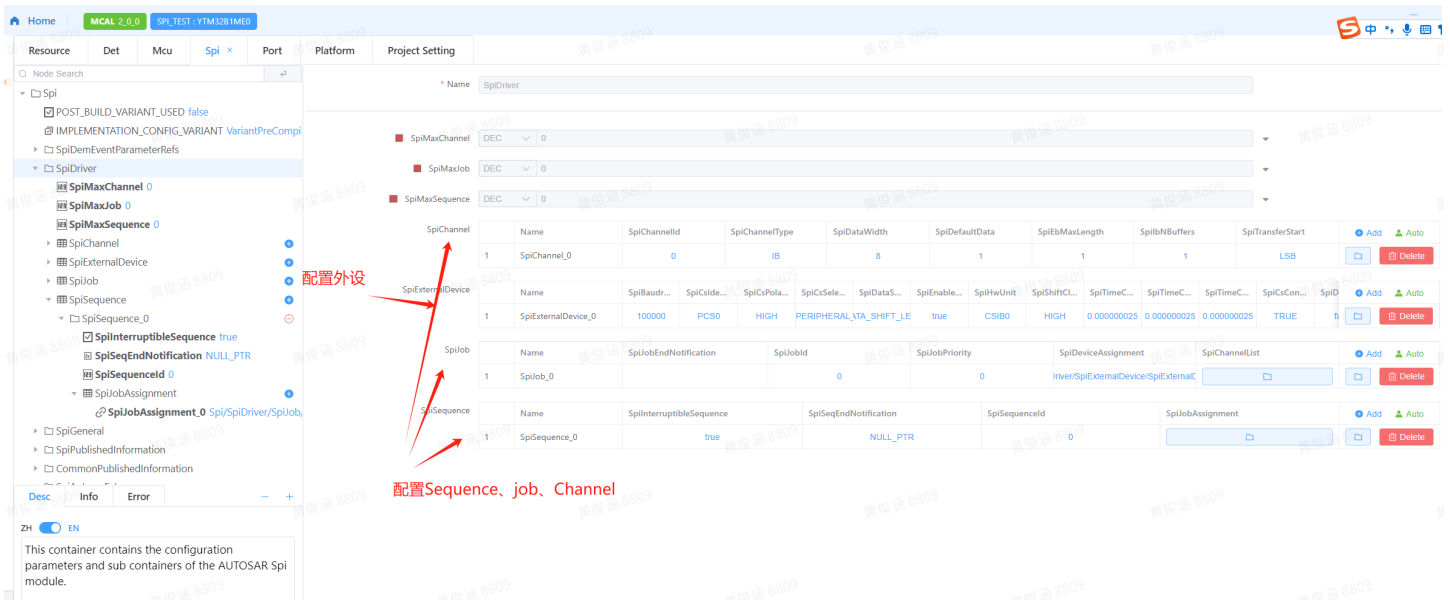
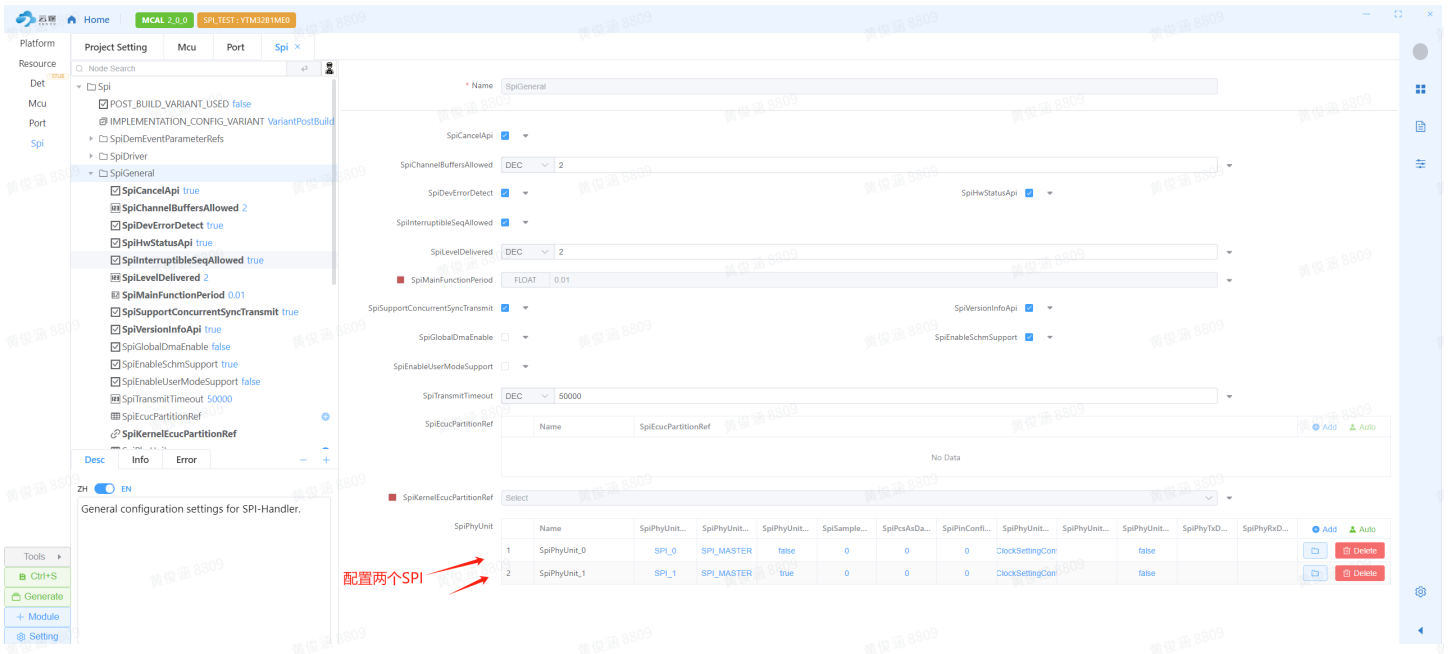
2.2 配置步骤

2.2.3 Port配置

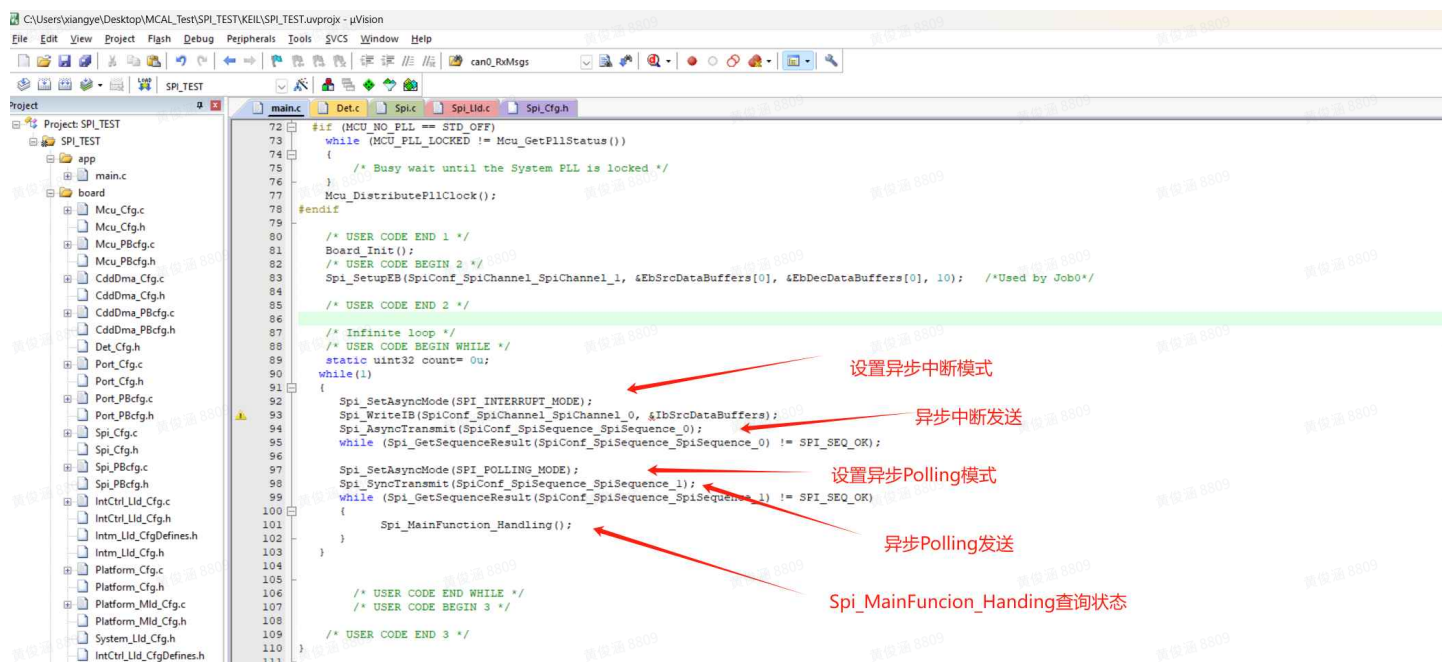




2.2.4 SPI模块配置



2.3 API说明



Tips:

Spi_SetAsyncMode用于改变通信方式，即异步中断和异步Polling

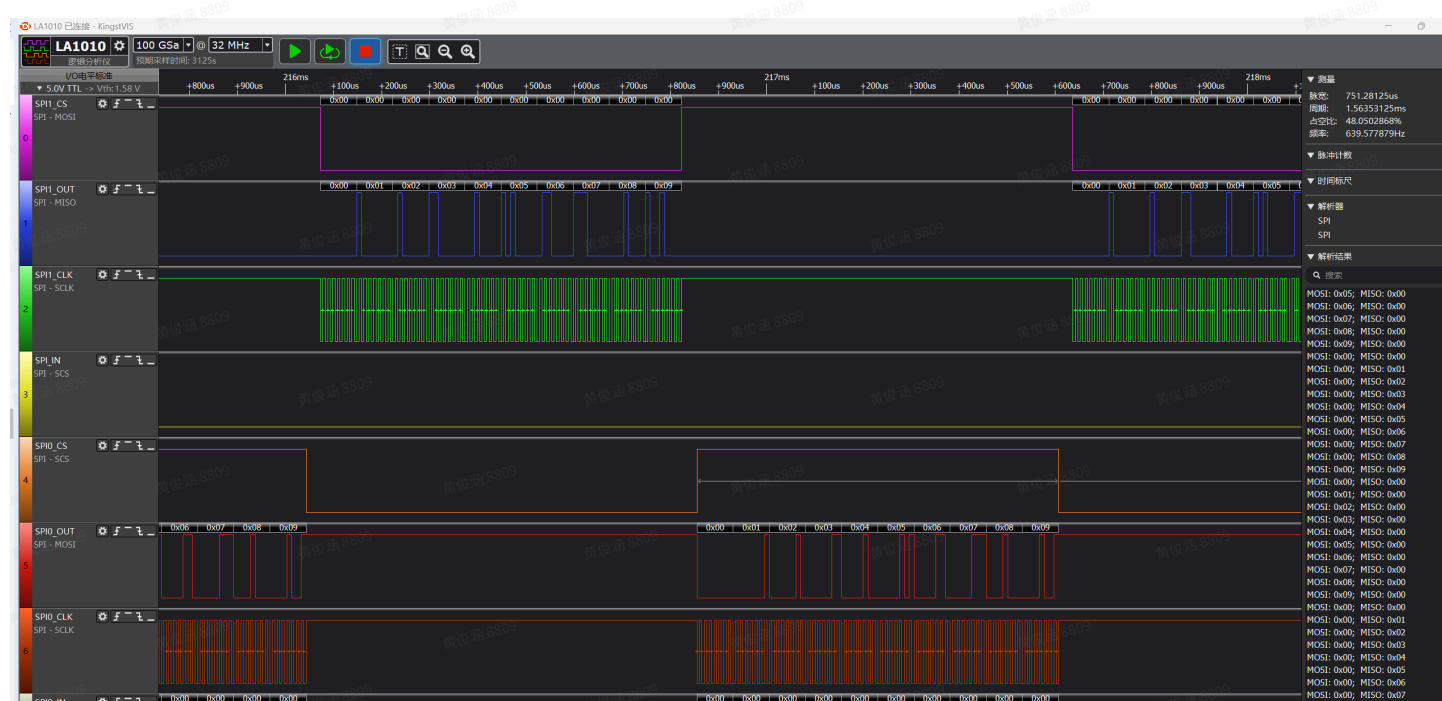
2.3.1 输出现象

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