YTMicro MCAL应用 - Dio模块配置及实例

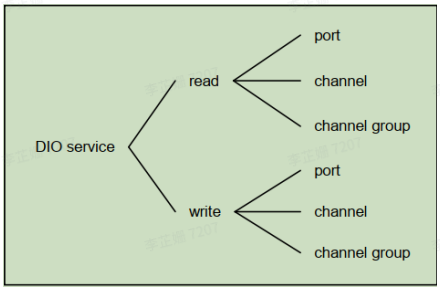
版本

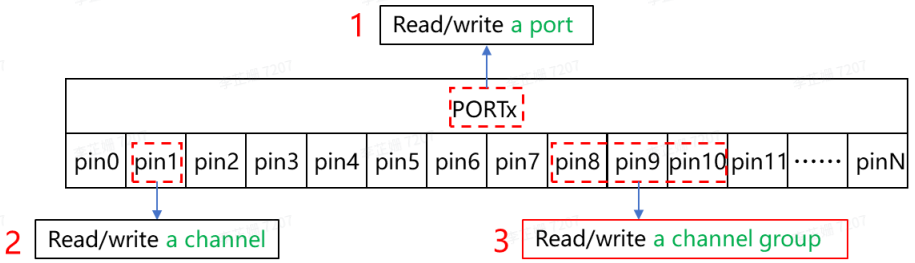
Config Tool Version：3.0.0

HA0 MCAL version：2.3.0

前言

1. DIO，即Digital Input/Output，该模块用于读取数字引脚输入及控制数字输出引脚电平状态。AUTOSAR标准中对于DIO模块的要求是需要定义这样一些函数，以满足读写某Port口、读写某个通道、读写某个通道组的操作，分别对应下图1、2、3。





其中，要实现读写通道组，首先需要用户自行定义通道组。定义规则：同一Port口用到2个或2个以上连续通道时（如图中的pin8、9、10），即可配置为一个通道组，调用读写通道组函数即可实现对这几个通道的同步操作。

2. 本文档将通过逐步配置一个MCAL工程，介绍Dio模块的配置方式及应用。Dio模块的使用需要依赖于一些其它的模块，工程包含的模块及功能如下图。本文档详细介绍其中标红部分Dio模块的配置方式，其它模块请参考对应文档。



3. Dio一般依赖Mcu、Platform和Port模块，Mcu模块里面需要配置外设时钟，Port配置使用到的端口资源。

4. 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》。
5. 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。

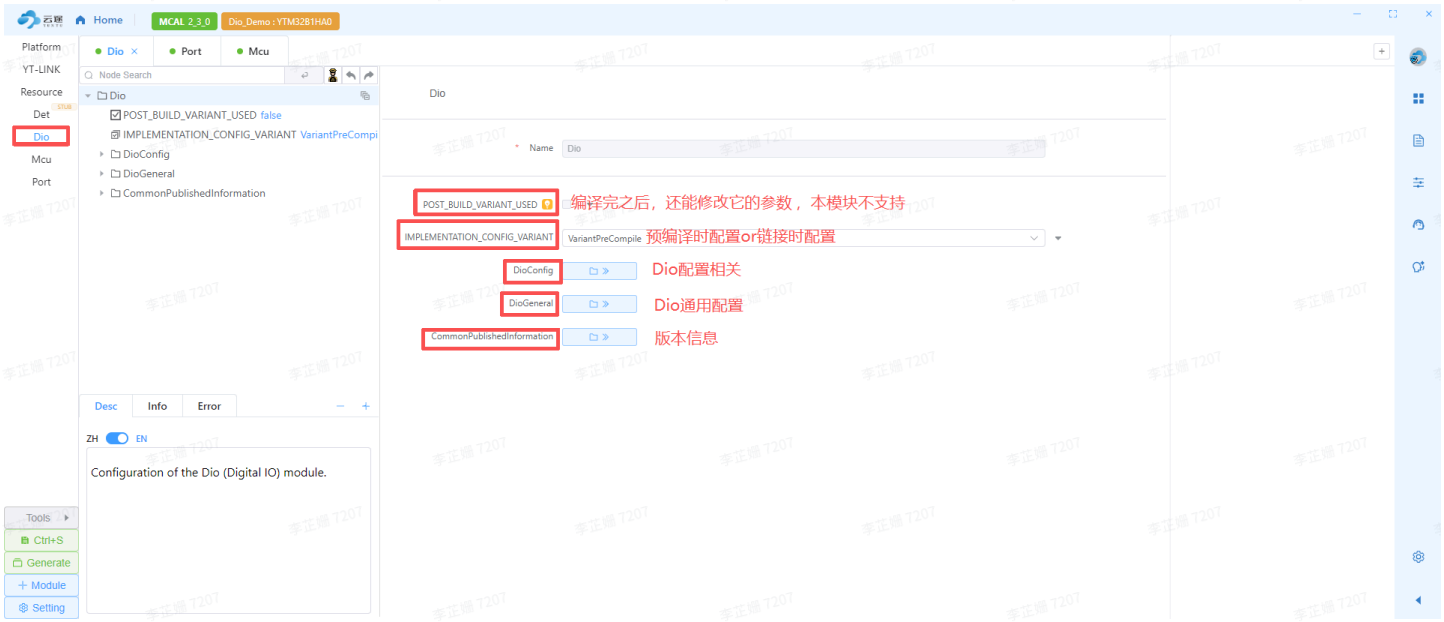
MCAL Dio配置介绍

1. 如何配置Dio模块

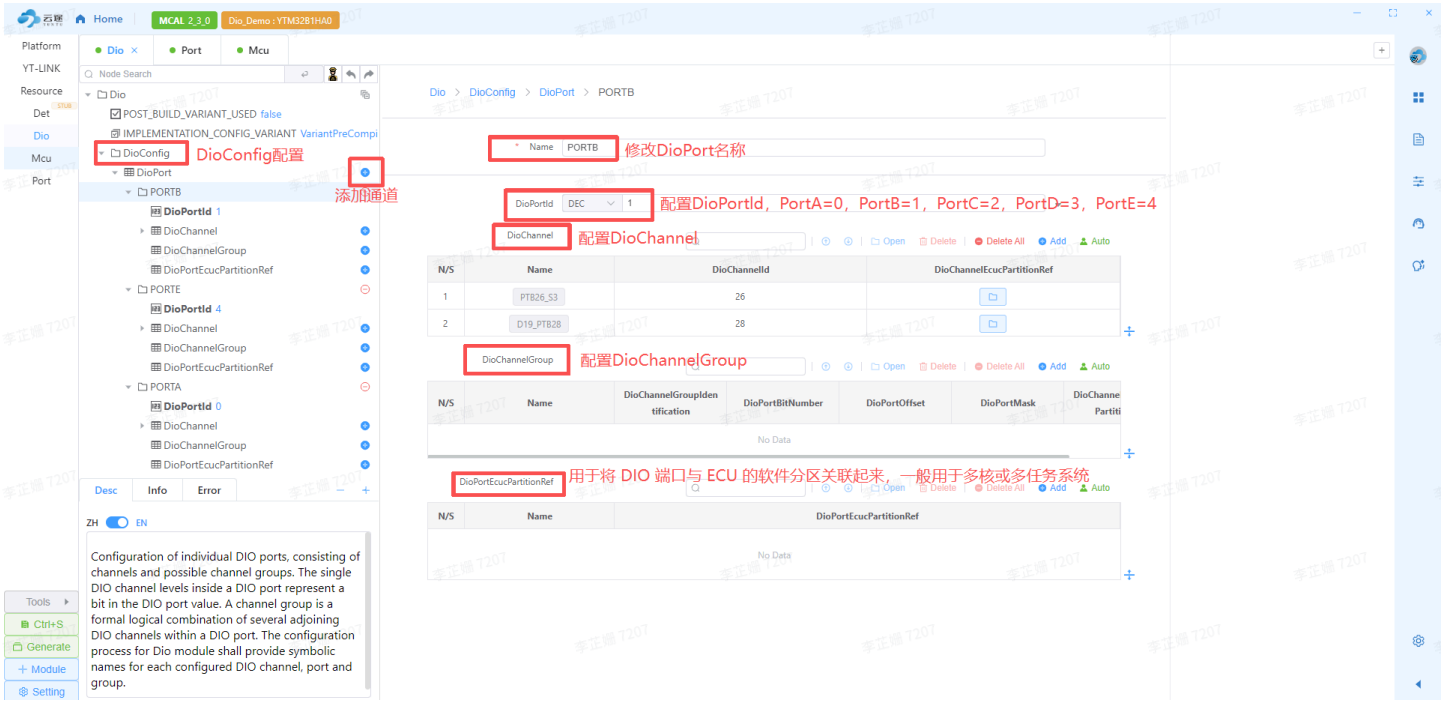
1.1 框架说明

这里宏观描述了Dio模块配置框架，其中需要我们着重关注的是DioConfig、DioGeneral。

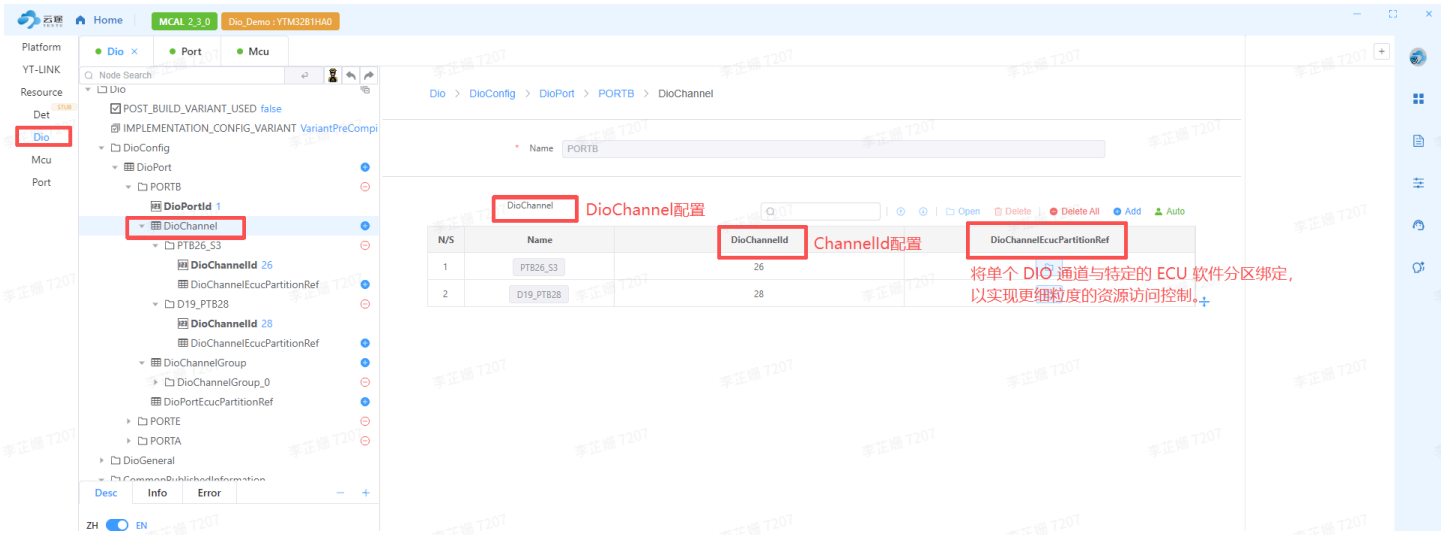
1.1.1 Dio宏观配置介绍



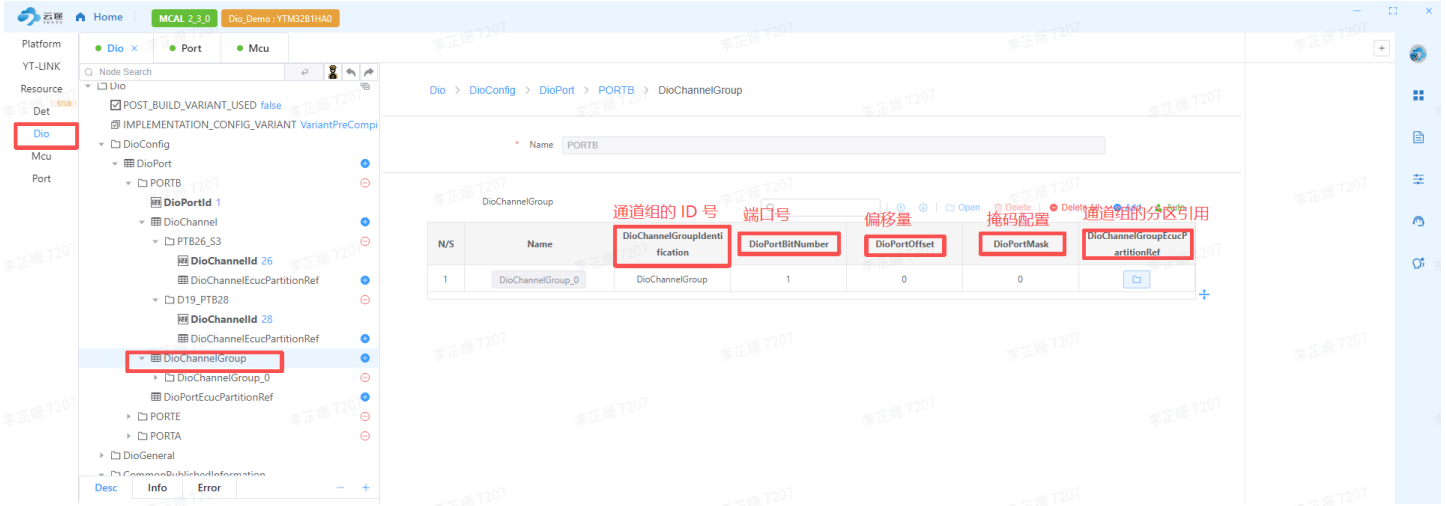
1.1.2 DioConfig配置介绍



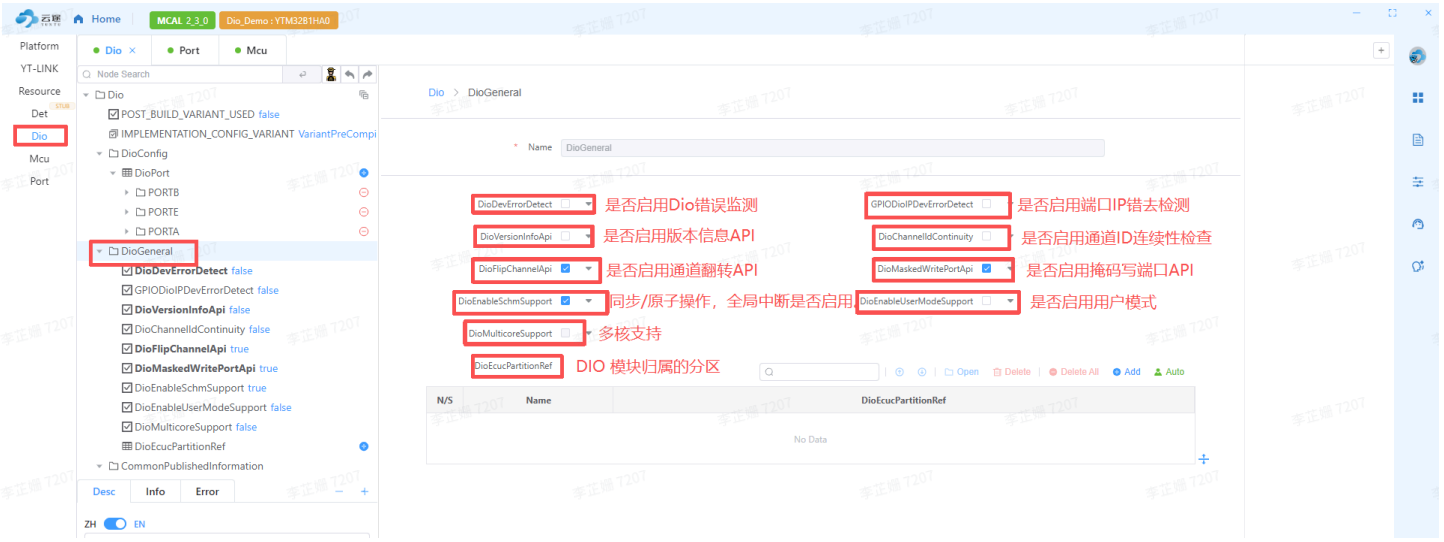
1.1.3 DioChannel配置介绍



1.1.4 DioChannelGroup配置介绍



1.1.5 DioGeneral配置介绍



2. 实例配置说明

2.1 功能介绍

通过按键控制 LED 灯的闪烁速度

默认状态：不按任何按键，D17、D18、D19 这三个 LED 灯会以1000ms速度闪烁。

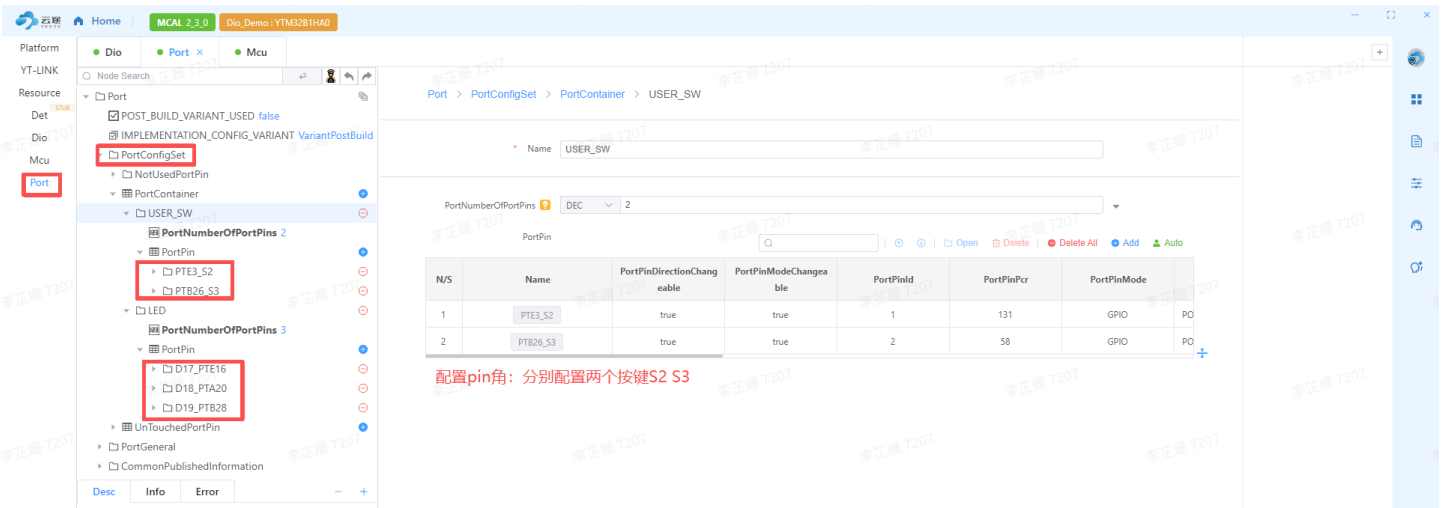
加速模式：

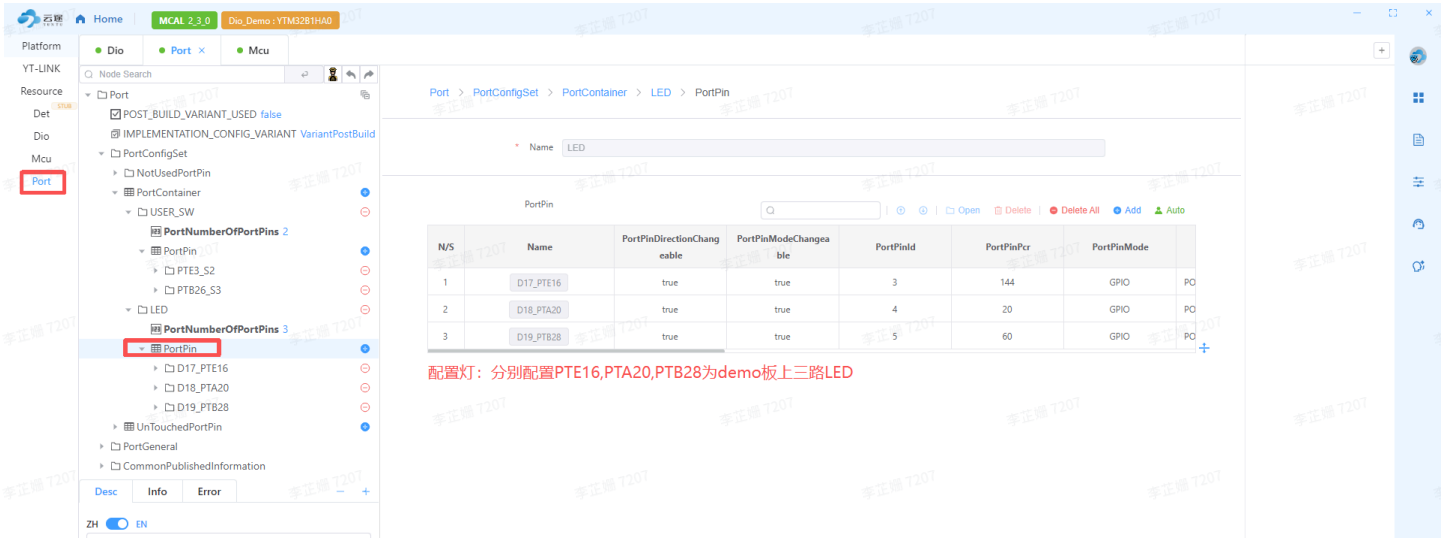
- 按 S3 键：闪烁速度变快 5 倍。
- 按 S2 键：闪烁速度变快 2 倍。
- 同时按 S2 和 S3：速度最快，变快 10 倍。

2.2 实例配置介绍

2.2.1 pin配置

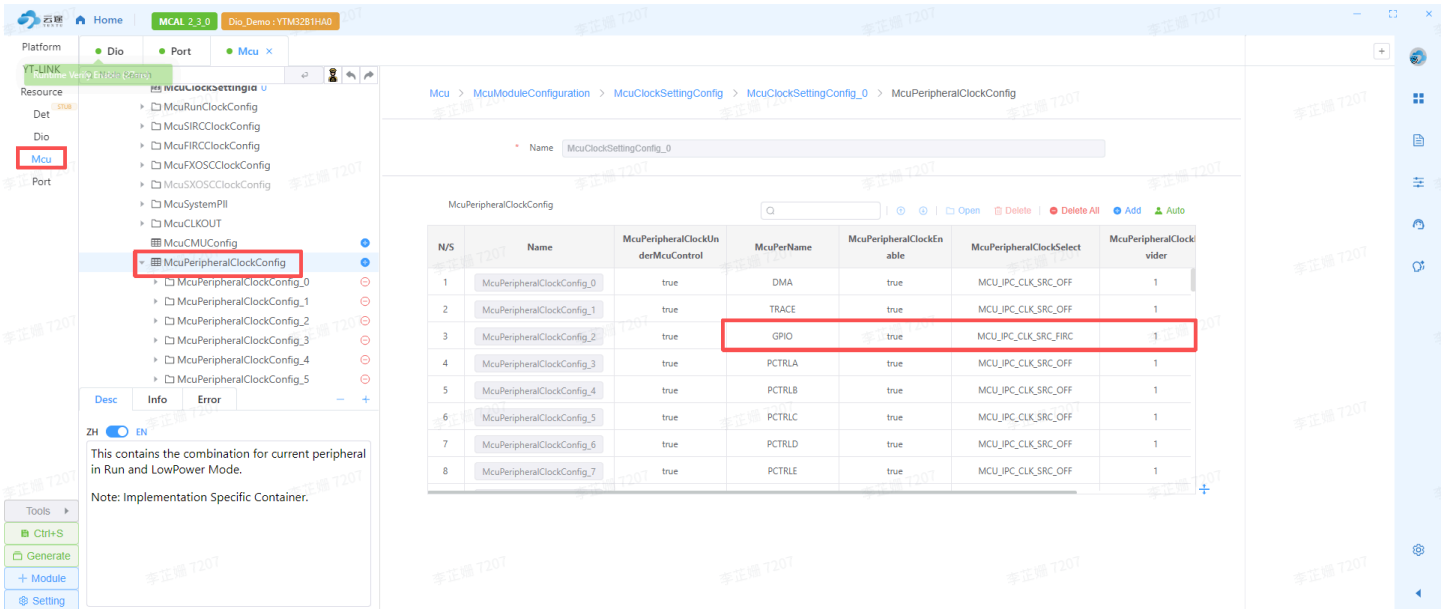
配置好pin角通道以及输入输出配置





2.2.2 时钟配置

需要开启gpio时钟，其时钟源可选功能时钟，时钟源可从FIRC、SIRC、FXOSC、SXOSC、PLL中选择。



如图为RM IPC时钟表

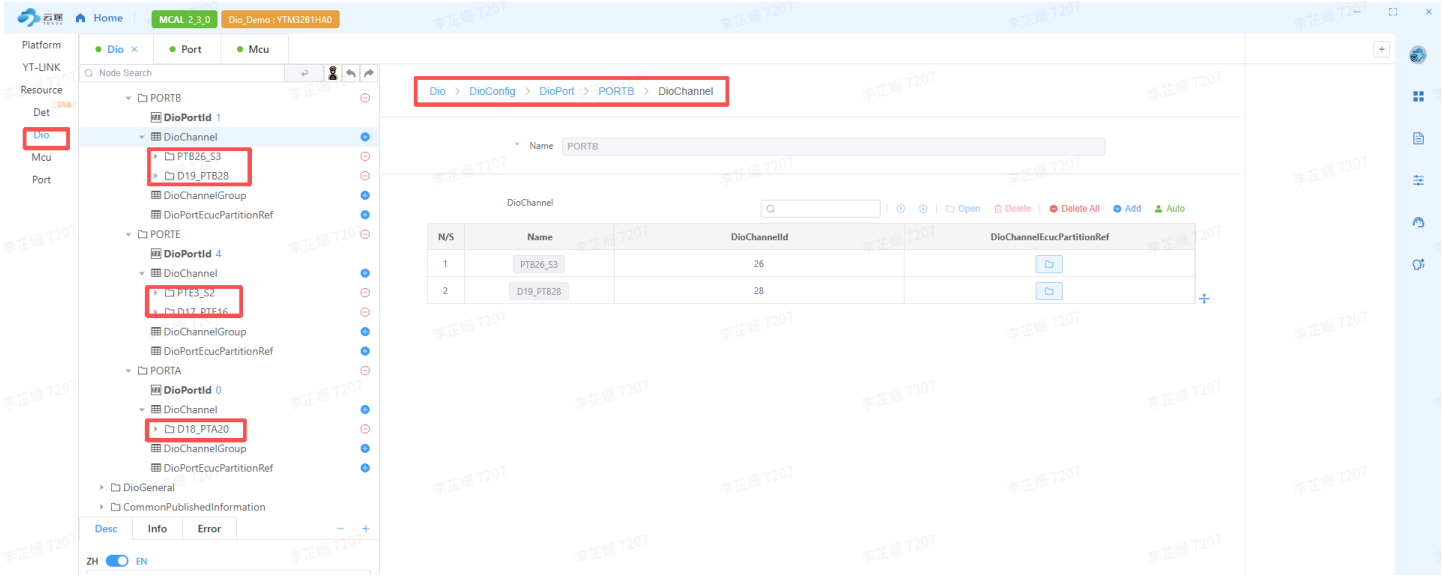
16	-	0x040	-	-	-	-	-	-
17	GPIO	0x044	CORE_CLK	OFF	YES	YES	-	≤ 100MHz
18	PCTRLA	0x048	SLOW_BUS_CLK	-	-	YES	-	-
19	PCTRLB	0x04C	SLOW_BUS_CLK	-	-	YES	-	-
20	PCTRLC	0x050	SLOW_BUS_CLK	-	-	YES	-	-

Table 15.1 continued from previous page

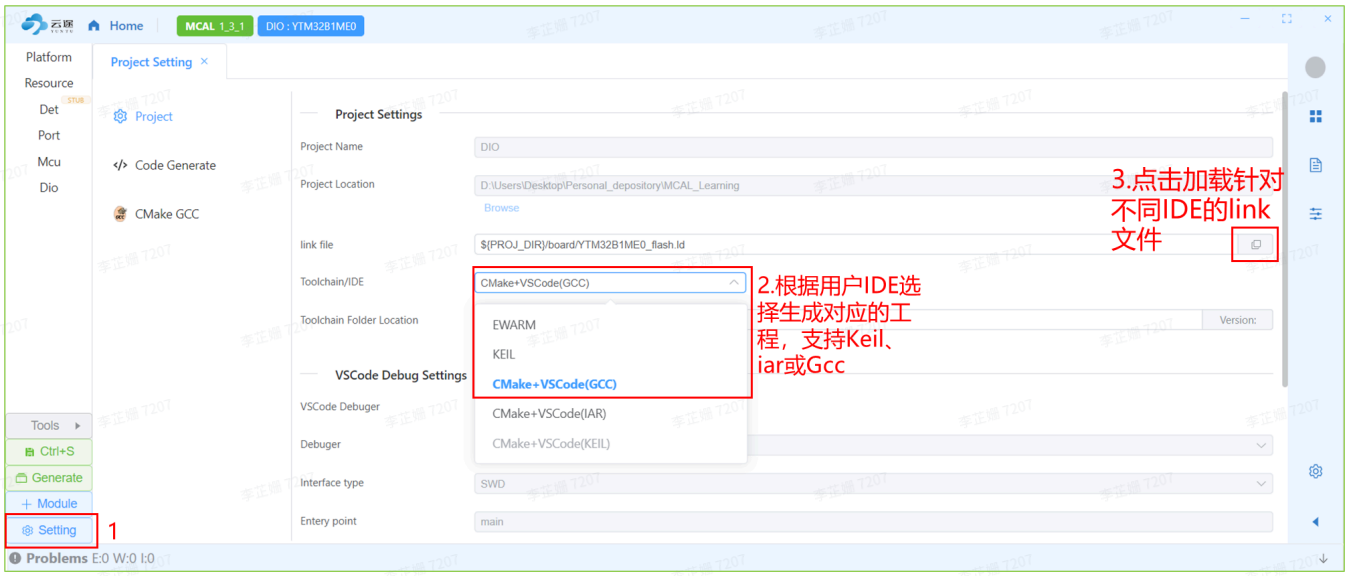
No.	Module Name	Offset	Bus Clock	Bus Clock Enable	Peripheral Functional Clock	Software Reset	Additional Clock	MAX Functional Clock Frequency
21	PCTRLD	0x054	SLOW_BUS_CLK	-	-	YES	-	-
22	PCTRLE	0x058	SLOW_BUS_CLK	-	-	YES	-	-

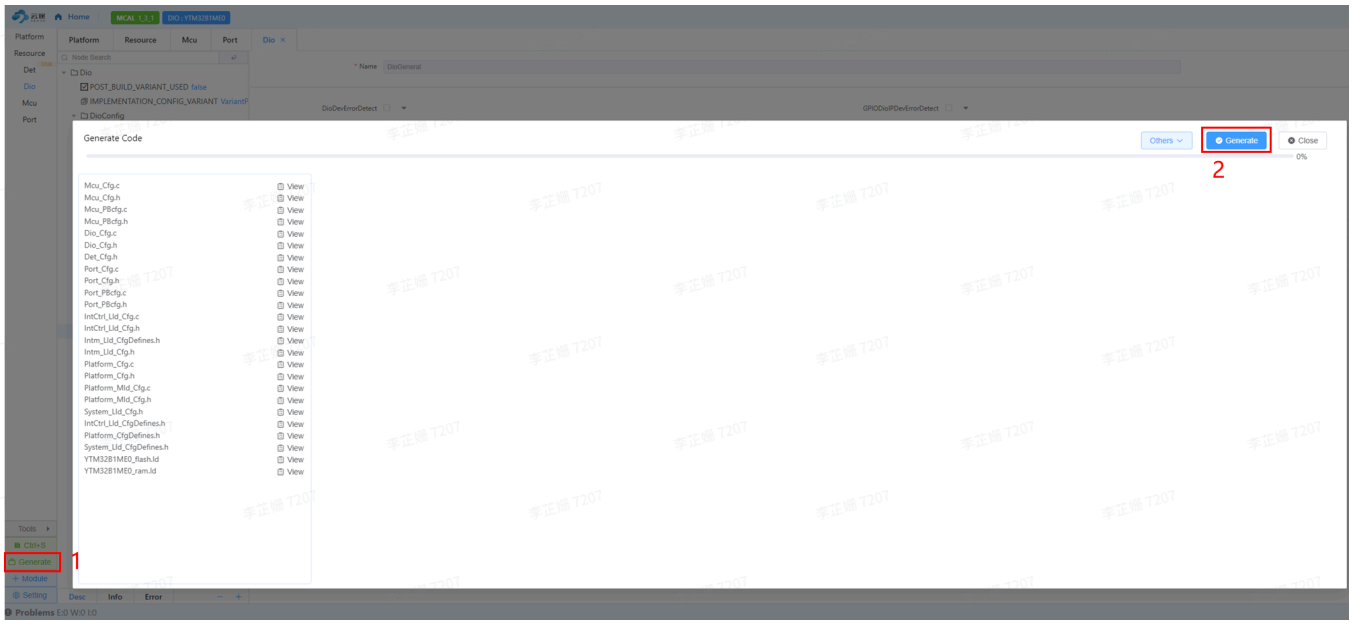
2.2.3 Dio模块配置

此demo分别配置了PTB26,PTB28,PTE3,PTE16,PTA20



2.2.4 生成工程





3. 代码解释

1. 设置延时时间。按下 S3，间隔缩小 5 倍；按下 S2，间隔缩小 2 倍；

```
145         uint16 DelayMs = 1000;
146         if (Dio_ReadChannel(DioConf_DioChannel_PTB26_S3) == STD_HIGH)
147         {
148             DelayMs /= 5;
149         }
150         if (Dio_ReadChannel(DioConf_DioChannel_PTE3_S2) == STD_HIGH)
151         {
152             DelayMs /= 2;
153         }
154         Delay_Ms(DelayMs);
```

2. Toggle D17, D18, D19

```
141         Dio_FlipChannel(DioConf_DioChannel_D17_PTE16); /* toggle D17 */
142         Dio_FlipChannel(DioConf_DioChannel_D18_PTA20); /* toggle D18 */
143         Dio_FlipChannel(DioConf_DioChannel_D19_PTB28); /* toggle D19 */
```

4. DIO 常用接口

功能	AUTOSAR 标准接口	说明
读取单个引脚	Dio_ReadChannel()	读取高低电平
设置单个引脚	Dio_WriteChannel()	输出高低电平
读取整个端口	Dio_ReadPort()	读取 Port 所有引脚状态
写入整个端口	Dio_WritePort()	设置整个 Port 的输出状态
读取引脚组	Dio_ReadChannelGroup()	读取指定引脚组
写入引脚组	Dio_WriteChannelGroup()	修改指定引脚组
翻转引脚电平	Dio_FlipChannel()	高低电平切换
读取输出电平	Dio_ReadChannel()	具体读取引脚还是输出锁存值，取决于硬件和驱动实现

5. DIO 和 PORT 模块有什么区别?

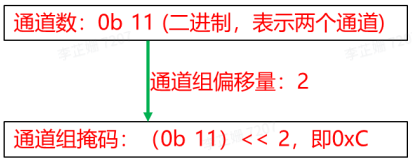
这是 AUTOSAR MCAL 中容易混淆的两个模块。

对比	PORT	DIO
主要作用	配置引脚属性	读写数字电平
输入/输出方向	支持配置	不负责
上拉/下拉	支持配置（取决于硬件）	不支持
引脚复用	支持配置	不支持
输出高低电平	可配置初始电平	运行时控制
读取输入电平	不负责	支持

简单理解：PORT 负责配置 GPIO，DIO 负责使用 GPIO。

注：DioChannelGroup配置。配置通道组的目的是实现对若干连续通道的整体控制，非必配项，用户根据需求选配。

当选择PORT_B的pin2、Pin3作为一个通道组时:



3. 读写channelgroup、写掩码通道及翻转IO口电平几个API函数进行Dio模块的基础应用说明。

（1）读通道组

读通道组的API为：

```
DIO_FUNC Dio_PortLevelType Dio_ReadChannelGroup(const Dio_ChannelGroupType *ChannelGroupIdPtr);
```

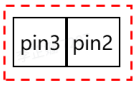
其中，返回值为32位无符号数，形参为通道组的结构体指针

例如，通道组0均为输入引脚，读取通道组0引脚输入状态

即：channel_group_read = Dio_ReadChannelGroup(DioConf_DioChannelGroup_USER_SW);

由于硬件上通道组0的两个引脚均连接电源，故均为高电平， channel_group_read = 0x3 = 0b 0011;

读操作对象-channelgroup0



channel_group_read = 0b 1 1 （表示PTB3、 PTB2电平为高）

（2）写通道组

写通道组的API为：

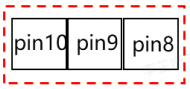
```
Dio_WriteChannelGroup(const Dio_ChannelGroupType *ChannelGroupIdPtr, Dio_PortLevelType Level);
```

其中，形参1为通道组的结构体指针，形参2为通道组的目标输出电平

例如，通道组1均为输出引脚，想要将通道组1的PTB10、 PTB9置高，PTB8置低时(0x6)，形参2被配置为0x06

即：Dio_WriteChannelGroup(DioConf_DioChannelGroup_RGB_LED, 0x6); 其中，0x6 = 0b 110

写操作对象-channelgroup1



形参2 (0x6) 0b 1 1 0 （表示将PTB10、 PTB9置高，PTB8置低）

（3）写掩码通道

写掩码通道的API为：

```
Dio_MaskedWritePort(Dio_PortType PortId, Dio_PortLevelType Level, Dio_PortLevelType Mask);
```

通过掩码的方式对若干通道进行写操作，使用更加灵活:

例如，需要将PTB8，PTB10两个通道置高时：

```
Dio_MaskedWritePort(Dio_PortType PortId, 0x500, 0x500);
```

写操作对象Mask-pin10 pin8



形参2,目标电平设置 (0x500) 0b 1 0 1 0 0 0 0 0 0 0 0

形参3,目标通道设置 (0x500) 0b 1 0 1 0 0 0 0 0 0 0 0

(形参3配置为0x500,表示形参2操作的引脚为PTB10、PTB8，此时将形参2配置为0x500，表示将PTB10、PTB8电平置高)

文档历史

版本号	日期	修订记录
V2.0	2026.05.11	初始版本