YTMicro MCAL应用- Port模块配置及实例

1. 版本

Config Tool Version: 3.2.0

ME0 MCAL version: 2.3.0

2. 前言

1. Port一般依赖Mcu和Platform，Mcu模块里面需要配置外设时钟，Platform里面配置外设中断。
2. 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。
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。

3. 功能描述

按下按键2，点亮三个LED灯；按下按键3，熄灭三个LED灯。

按键 (SW) 对应的引脚

- 按键 2 (KEYSW2)：连接在芯片的 PTB2 引脚上。代码通过 `Dio_ReadChannel(DioConf_DioChannel_PTB2_KEYSW2)` 来读取它的状态。
- 按键 3 (KEYSW3)：连接在芯片的 PTB3 引脚上。

LED 灯对应的引脚

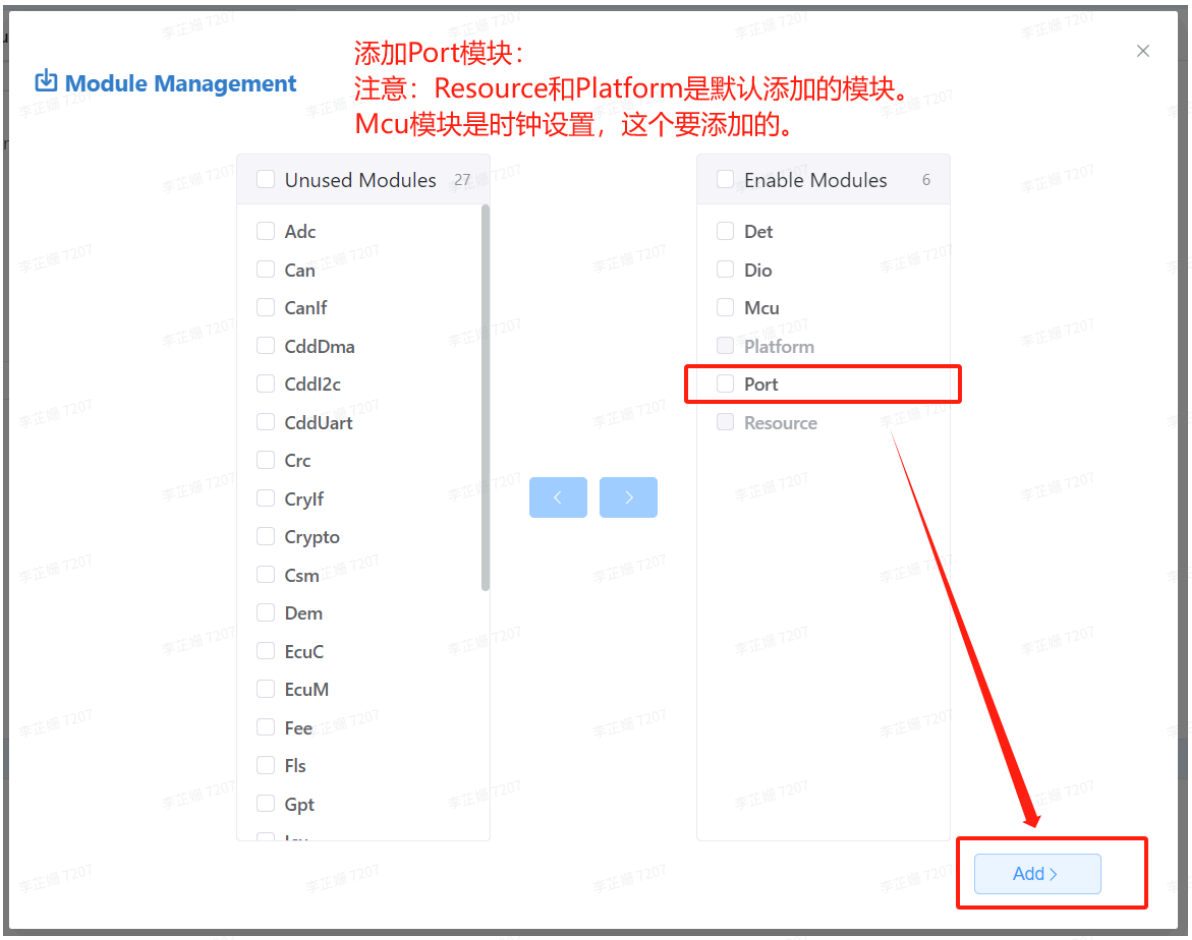
- LED5：连接在 PTB4 引脚。
- LED6：连接在 PTB5 引脚。
- LED7：连接在 PTE8 引脚。

4. 添加Port模块

a. Resource和Platform是系统默认必须添加的， 其它是自选模块。

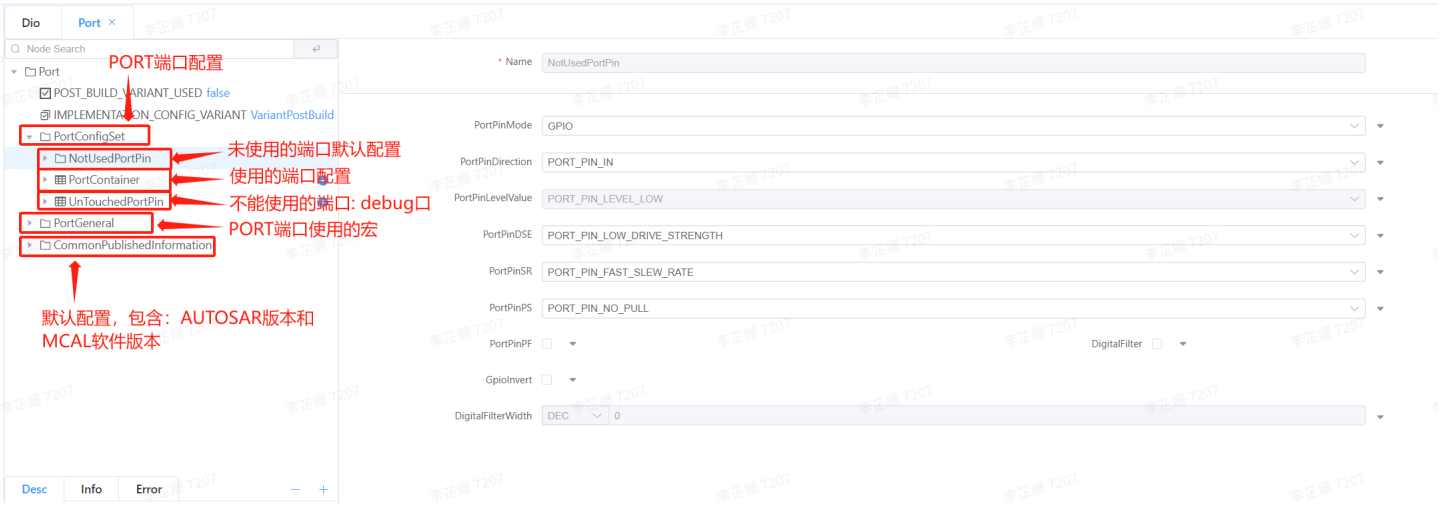
b. Det模块也需要添加，但不需要配置，不然工程在编译时会提示“Det.h 没有包含”。

c. Mcu模块是配置时钟的， 这个也要添加并配置。详细配置见Mcu章节配置。



5. Port模块配置介绍

5.1 模块简介



5.2 未使用的端口设置



Tips: 未使用的引脚都是按照图片上的默认配置。

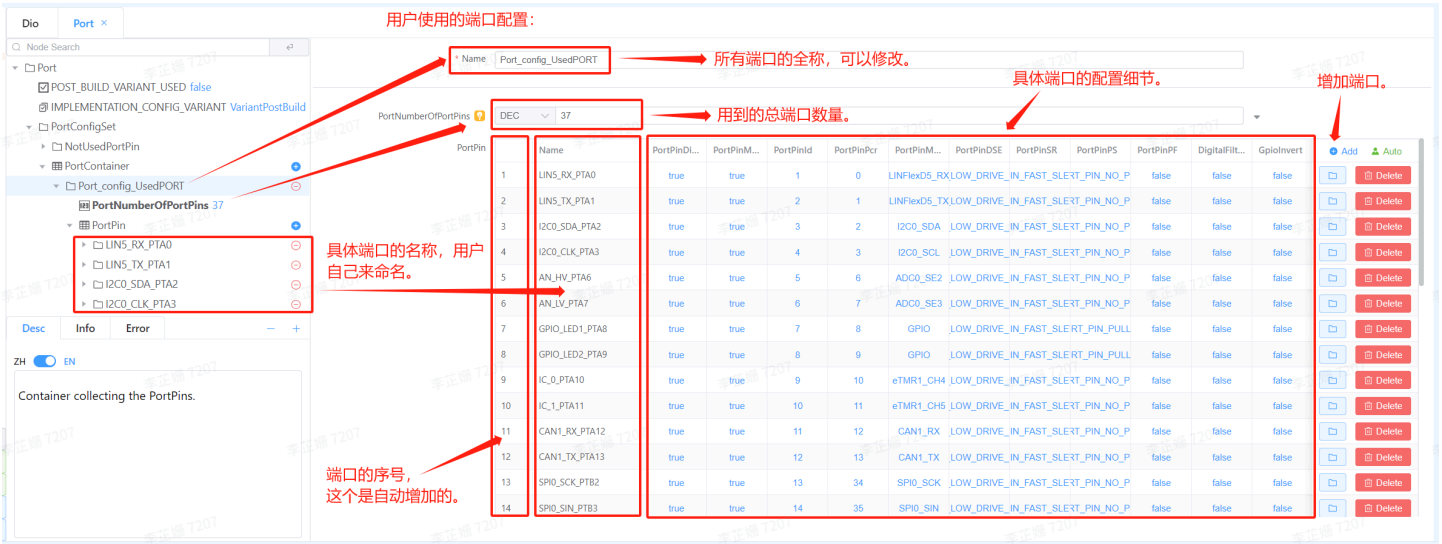
5.3 禁用端口的设置



Tips: 禁用的引脚是SWD/JTAG口，按照图片上的默认配置。

5.4 用户配置端口：

5.4.1 用户配置的所有端口预览：

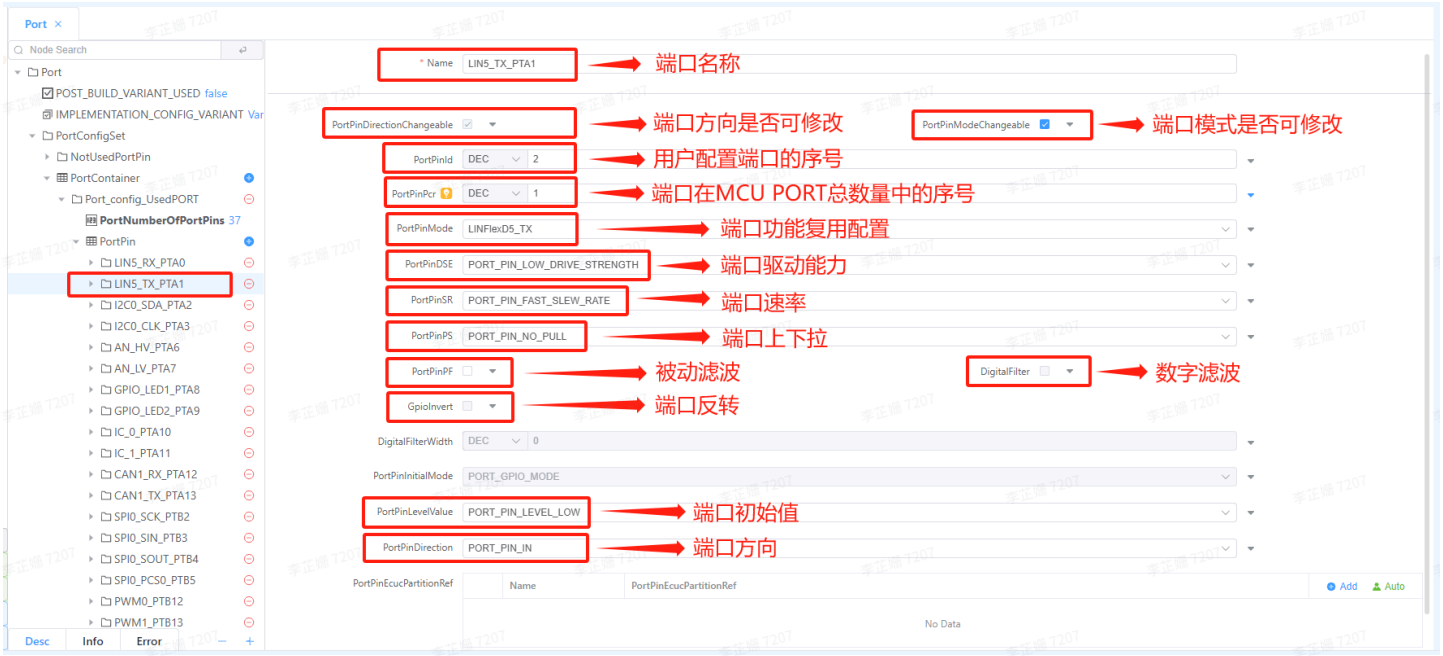


Tips: PortPinMode就是引脚复用功能的选择, 对应芯片RM中的PinMUX。

144LQFP	NAME	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7
63	PTB_3	ADC0_SE7	PTB_3	eTMR1_CH1	SPI0_SIN	eTMR1_QD_PHA		TMU_IN2	
113	PTA_1	ADC0_SE1/ACMP0_IN1	PTA_1	eTMR1_CH1	LINFlexD5_TX		eTMR1_QD_PHA		TMU_OUT0

5.4.2 以PTA1设置为LIN5_TX, 和PTB3设置为SPI0_SIN为例，做具体配置介绍：

5.4.2.1 PTA1配置为LIN5_TX所需的参数设置如下，同理PTB3设置为SPI0_SIN同样操作：



Tips: 关于被动滤波PortPinPF和数字滤波DigitalFilter和说明：

=> PF是passive filter的缩写, 相当于RC滤波器，主要作用是滤除高频脉冲信号(MHz级别)， 不是所有的pin都有这个滤波器，目前只有reset pin和NMI pin才有，通用的GPIO里面没有这个滤波器。

=> digital filter，可以认为是低通滤波器， 可以设置它的滤波参数：滤波器宽度，滤波器时钟默认是GPIO口的功能时钟。若功能时钟是96MHz, 宽度选为8(96M/8=12M)， 那么滤波器的截止频率就是12MHz.

5.4.2.2 重点介绍：

关于PortPinPcr输入的数值，有以下2个方式：

1. 可以直接输入端口名称PTA1或者PTB3，在下拉菜单中选择Convert，便会自动转换为它们在PORT中的序号，分别为1和35。
2. 直接输入数值1和35便可。用户可以使用自己习惯的方式。

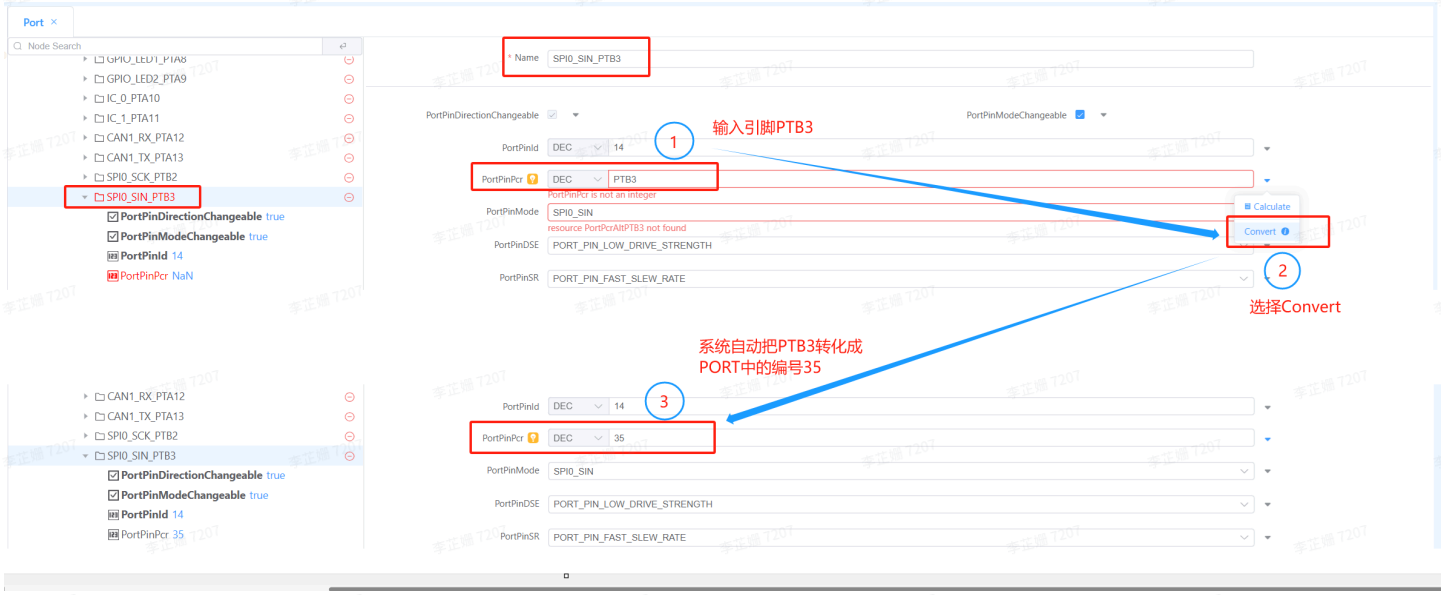
Tips:

ME0共有5个PORT(A/B/C/D/E)，序号为0/1/2/3/4， 每个PORT分配32个pin，所以 PortPinPcr的值从0开始依此排列下去，见下图：

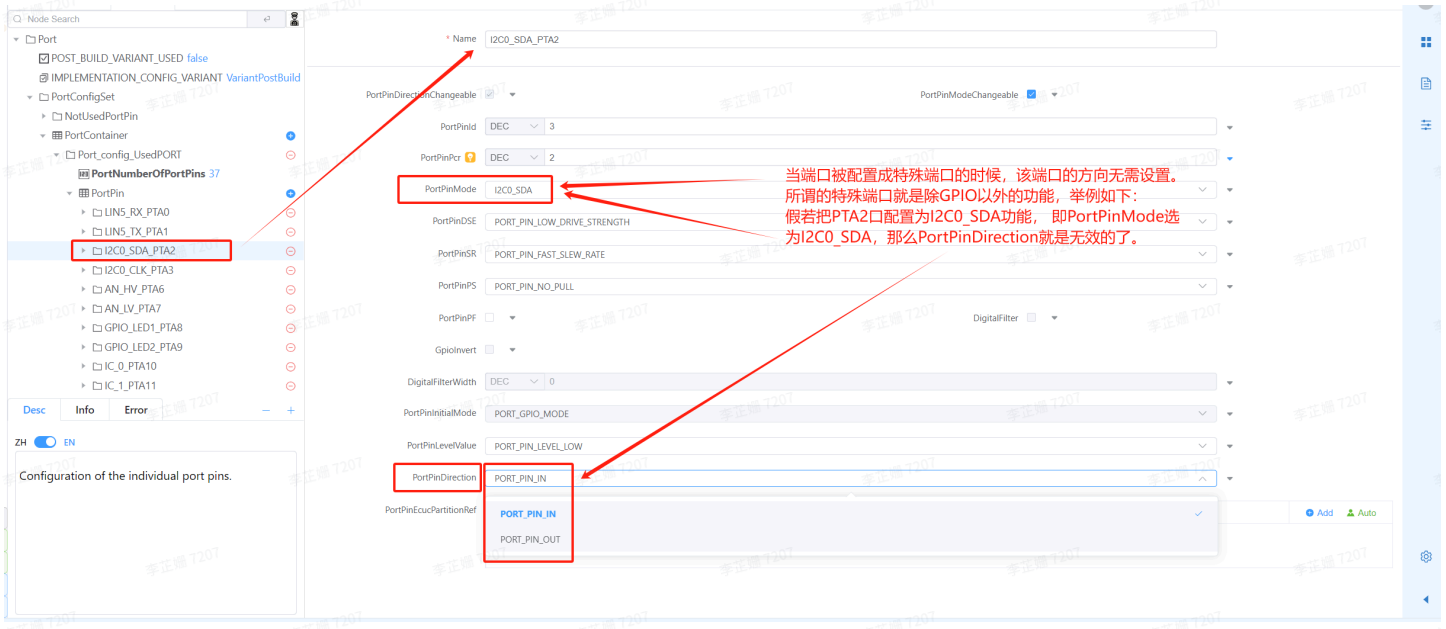
	PORTA			PORTB			PORTC			PORTD			PORTE		
	PTA0	...	PTA31	PTB0	...	PTB31	PTC0	...	PTC31	PTD0	...	PTD31	PTE0	...	PTE31
PortPinPcr	0	...	31	32	...	63	64	...	95	96	...	127	128	...	159

下面以PTA1和PTB3这2个引脚做具体操作演示，见下面图片说明：

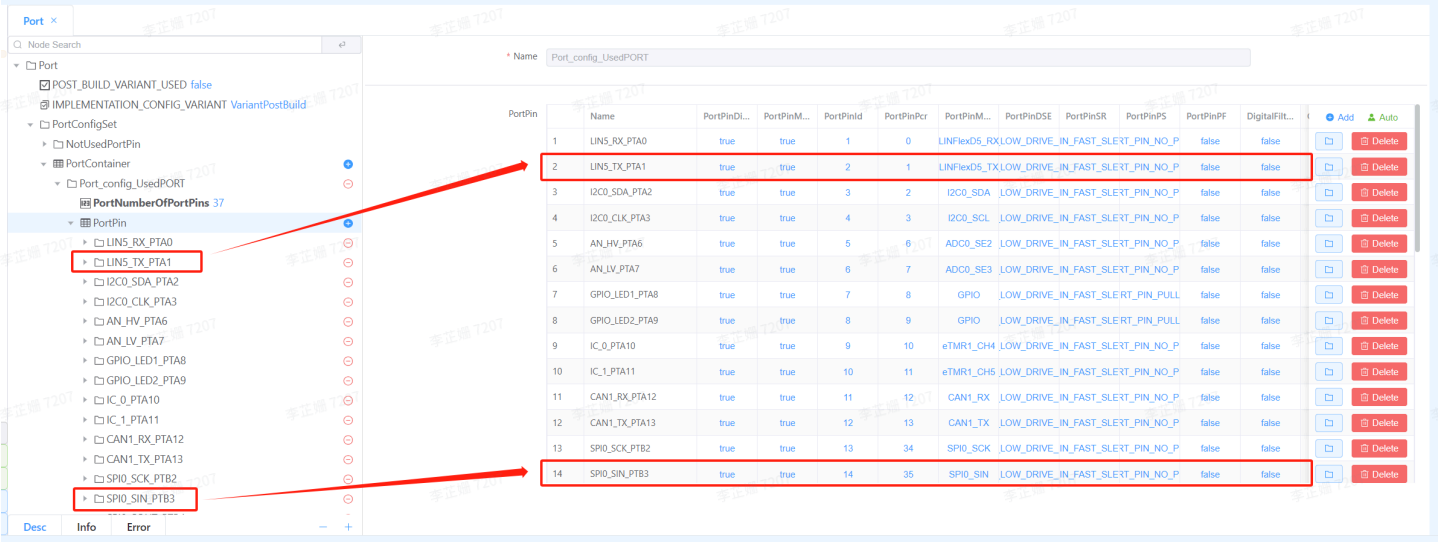




Tips: 当PortPinMode被配置成特殊端口【即:当不做GPIO口功能时】，这个口的端口方向就无需配置，见下面图片解释：



5.4.2.3 端口配置完成汇总



5.4.2.4 生成代码预览

用户配置端口生成的代码：

```
#define PortConfigSet_Port_config_UsedPORT_LIN5_RX_PTA0   ((Port_PinType)(((0)<<8)|0))
#define PortConfigSet_Port_config_UsedPORT_LIN5_TX_PTA1   ((Port_PinType)(((0)<<8)|1))
#define PortConfigSet_Port_config_UsedPORT_I2C0_SDA_PTA2   ((Port_PinType)(((0)<<8)|2))
#define PortConfigSet_Port_config_UsedPORT_I2C0_CLK_PTA3   ((Port_PinType)(((0)<<8)|3))
#define PortConfigSet_Port_config_UsedPORT_AN_HV_PTA6   ((Port_PinType)(((0)<<8)|6))
#define PortConfigSet_Port_config_UsedPORT_AN_LV_PTA7   ((Port_PinType)(((0)<<8)|7))
#define PortConfigSet_Port_config_UsedPORT_GPIO_LED1_PTA8   ((Port_PinType)(((0)<<8)|8))
#define PortConfigSet_Port_config_UsedPORT_GPIO_LED2_PTA9   ((Port_PinType)(((0)<<8)|9))
#define PortConfigSet_Port_config_UsedPORT_IC_0_PTA10   ((Port_PinType)(((0)<<8)|10))
#define PortConfigSet_Port_config_UsedPORT_IC_1_PTA11   ((Port_PinType)(((0)<<8)|11))
#define PortConfigSet_Port_config_UsedPORT_CAN1_RX_PTA12   ((Port_PinType)(((0)<<8)|12))
#define PortConfigSet_Port_config_UsedPORT_CAN1_TX_PTA13   ((Port_PinType)(((0)<<8)|13))
#define PortConfigSet_Port_config_UsedPORT_SPI0_SCK_PTB2   ((Port_PinType)(((1)<<8)|2))
#define PortConfigSet_Port_config_UsedPORT_SPI0_SIN_PTB3   ((Port_PinType)(((1)<<8)|3))
#define PortConfigSet_Port_config_UsedPORT_SPI0_SOUT_PTB4   ((Port_PinType)(((1)<<8)|4))
#define PortConfigSet_Port_config_UsedPORT_SPI0_PCS0_PTB5   ((Port_PinType)(((1)<<8)|5))
#define PortConfigSet_Port_config_UsedPORT_PWM0_PTB12   ((Port_PinType)(((1)<<8)|12))
#define PortConfigSet_Port_config_UsedPORT_PWM1_PTB13   ((Port_PinType)(((1)<<8)|13))
#define PortConfigSet_Port_config_UsedPORT_PWM2_PTB14   ((Port_PinType)(((1)<<8)|14))
#define PortConfigSet_Port_config_UsedPORT_PWM3_PTB15   ((Port_PinType)(((1)<<8)|15))
#define PortConfigSet_Port_config_UsedPORT_PWM4_PTB16   ((Port_PinType)(((1)<<8)|16))
#define PortConfigSet_Port_config_UsedPORT_PWM5_PTB17   ((Port_PinType)(((1)<<8)|17))
#define PortConfigSet_Port_config_UsedPORT_AN_Curr1_PTC0   ((Port_PinType)(((2)<<8)|0))
#define PortConfigSet_Port_config_UsedPORT_AN_Curr2_PTC1   ((Port_PinType)(((2)<<8)|1))
#define PortConfigSet_Port_config_UsedPORT_FAN1_PTC30   ((Port_PinType)(((2)<<8)|30))
#define PortConfigSet_Port_config_UsedPORT_FAN2_PTC31   ((Port_PinType)(((2)<<8)|31))
#define PortConfigSet_Port_config_UsedPORT_XTAL32_PTD15   ((Port_PinType)(((3)<<8)|15))
#define PortConfigSet_Port_config_UsedPORT_EXTAL32_PTD16   ((Port_PinType)(((3)<<8)|16))
#define PortConfigSet_Port_config_UsedPORT_ACMP0_PTD6   ((Port_PinType)(((3)<<8)|6))
#define PortConfigSet_Port_config_UsedPORT_ACMP0_PTD7   ((Port_PinType)(((3)<<8)|7))
#define PortConfigSet_Port_config_UsedPORT_GPIO_Relay1_PTE15   ((Port_PinType)(((4)<<8)|15))
#define PortConfigSet_Port_config_UsedPORT_GPIO_Relay2_PTE16   ((Port_PinType)(((4)<<8)|16))
#define PortConfigSet_Port_config_UsedPORT_XTAL_PTB6   ((Port_PinType)(((1)<<8)|6))
#define PortConfigSet_Port_config_UsedPORT_EXTAL_PTB7   ((Port_PinType)(((1)<<8)|7))
#define PortConfigSet_Port_config_UsedPORT_SWDIO_PTA4   ((Port_PinType)(((0)<<8)|4))
#define PortConfigSet_Port_config_UsedPORT_RESET_PTA5   ((Port_PinType)(((0)<<8)|5))
#define PortConfigSet_Port_config_UsedPORT_SWDCLK_PTC4   ((Port_PinType)(((2)<<8)|4))
```

PTA1配置对应的代码：

```
/* PCR-1, Feature-LINFlexD5_TX */

{
    .HwPinId = 1,
    .Direction = PORT_PIN_IN,
    .IsDirectionChangeable= PORT_PIN_DIRECTION_CHANGEABLE,
    .InitLevel = PORT_PIN_LEVEL_LOW,
    .InitMode=  PORT_PIN_MODE_ALT3,
    .IsModeChangeable = PORT_PIN_MODE_CHANGEABLE,
    .PullConfig = PORT_PIN_NO_PULL,
    .DriveStrength = PORT_PIN_LOW_DRIVE_STRENGTH,
    .SlewRate = PORT_PIN_FAST_SLEW_RATE,
    .PassiveFilter = (boolean)0,
    .DigitalFilter = (boolean)0,
    .DigitalFilterWidth = 0,
    .InvertEnable = (boolean)0,
},
```

PTB3配置对应的代码：

```
/* PCR-35, Feature-SPI0_SIN */

{
    .HwPinId = 3,
    .Direction = PORT_PIN_IN,
    .IsDirectionChangeable= PORT_PIN_DIRECTION_CHANGEABLE,
    .InitLevel = PORT_PIN_LEVEL_LOW,
    .InitMode=  PORT_PIN_MODE_ALT3,
    .IsModeChangeable = PORT_PIN_MODE_CHANGEABLE,
    .PullConfig = PORT_PIN_NO_PULL,
    .DriveStrength = PORT_PIN_LOW_DRIVE_STRENGTH,
    .SlewRate = PORT_PIN_FAST_SLEW_RATE,
    .PassiveFilter = (boolean)0,
    .DigitalFilter = (boolean)0,
    .DigitalFilterWidth = 0,
    .InvertEnable = (boolean)0,
},
```

6. Port模块应用示例

6.1 PIN脚定义：

```
/** @{ */
#define DioConf_DioChannel_PTA5_RESET        (((0U << 8U) + 5U))
#define DioConf_DioChannel_PTA4_SWDIO        (((0U << 8U) + 4U))
#define DioConf_DioChannel_PTB2_KEYSW2       (((1U << 8U) + 2U))
#define DioConf_DioChannel_PTB3_KEYSW3       (((1U << 8U) + 3U))
#define DioConf_DioChannel_PTB4_LED5         (((1U << 8U) + 4U))
#define DioConf_DioChannel_PTB5_LED6         (((1U << 8U) + 5U))
#define DioConf_DioChannel_PTB6_XTAL         (((1U << 8U) + 6U))
#define DioConf_DioChannel_PTB7_EXTAL        (((1U << 8U) + 7U))
#define DioConf_DioChannel_PTC4_SWDCLK       (((2U << 8U) + 4U))
#define DioConf_DioChannel_PTE8_LED7         (((4U << 8U) + 8U))
/** @} */
```

6.2 程序代码：

```
* @brief  The application entry point.
* @retval int
*/
int main(void)
{
    /* USER CODE BEGIN 1 */

    Mcu_Init(&Mcu_Config);
    Mcu_InitClock(0);
#if (MCU_NO_PLL == STD_OFF)
    while ( MCU_PLL_LOCKED != Mcu_GetPllStatus() )
    {
        /* Busy wait until the System PLL is locked */
    }
    Mcu_DistributePllClock();
#endif

    /* USER CODE END 1 */
    Board_Init();
    /* USER CODE BEGIN 2 */

    /* USER CODE END 2 */

    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    while (1)
    {
        /* USER CODE END WHILE */
        /* USER CODE BEGIN 3 */

        if(Dio_ReadChannel(DioConf_DioChannel_PTB2_KEYSW2) == STD_LOW)
        {
            /* turn on LED5/LED6/LED7 */
            Dio_WriteChannel(DioConf_DioChannel_PTB4_LED5, STD_LOW);
            Dio_WriteChannel(DioConf_DioChannel_PTB5_LED6, STD_LOW);
            Dio_WriteChannel(DioConf_DioChannel_PTE8_LED7, STD_LOW);

        }
        /* delay 100ms */
        Delay_Ms(100);
        if(Dio_ReadChannel(DioConf_DioChannel_PTB3_KEYSW3) == STD_LOW)
        {
            /* turn off LED5/LED6/LED7 */
            Dio_WriteChannel(DioConf_DioChannel_PTB4_LED5, STD_HIGH);
            Dio_WriteChannel(DioConf_DioChannel_PTB5_LED6, STD_HIGH);
            Dio_WriteChannel(DioConf_DioChannel_PTE8_LED7, STD_HIGH);

        }
    }
    /* USER CODE END 3 */
}

static void Board_Init(void)
{
    Platform_Init(NULL_PTR);
    Port_Init(&Port_Config);
}

/* USER CODE BEGIN 4 */
/* USER CODE END 4 */
```

6.3 测试结果：





文档历史

版本号	日期	修订记录
V1.0	2023.11.20	初始版本
V2.0	2026.11.09	