

MCAL应用_Port模块配置及实例（一）

版本

Config Tool Version: 1.10.0

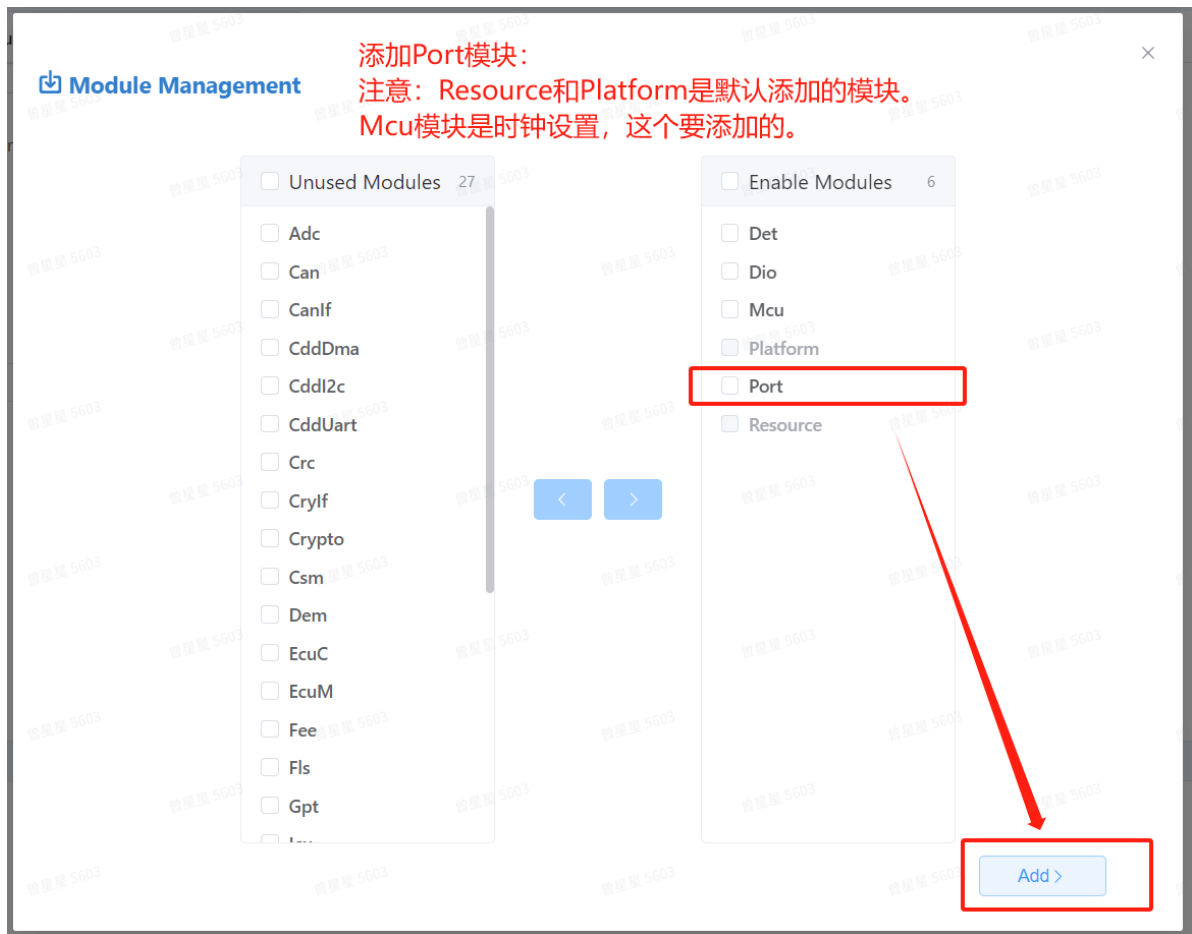
ME0 MCAL version: 1.3.1

前言

1. 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。

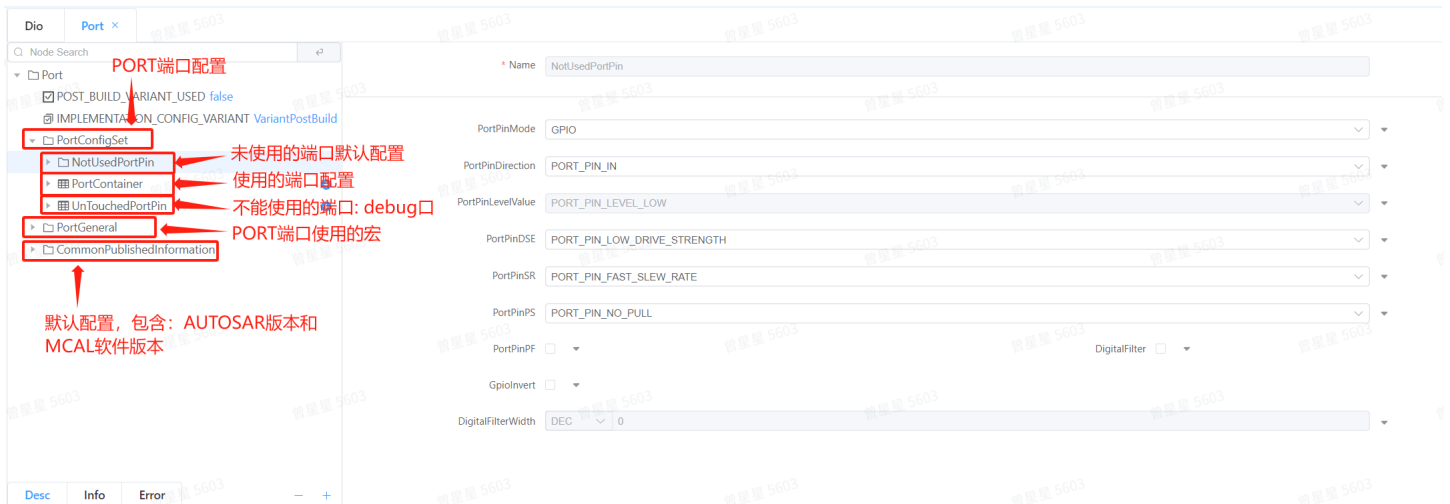
1. 添加Port模块

- 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。



2. Port模块配置介绍

2.1 模块简介



2.2 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。

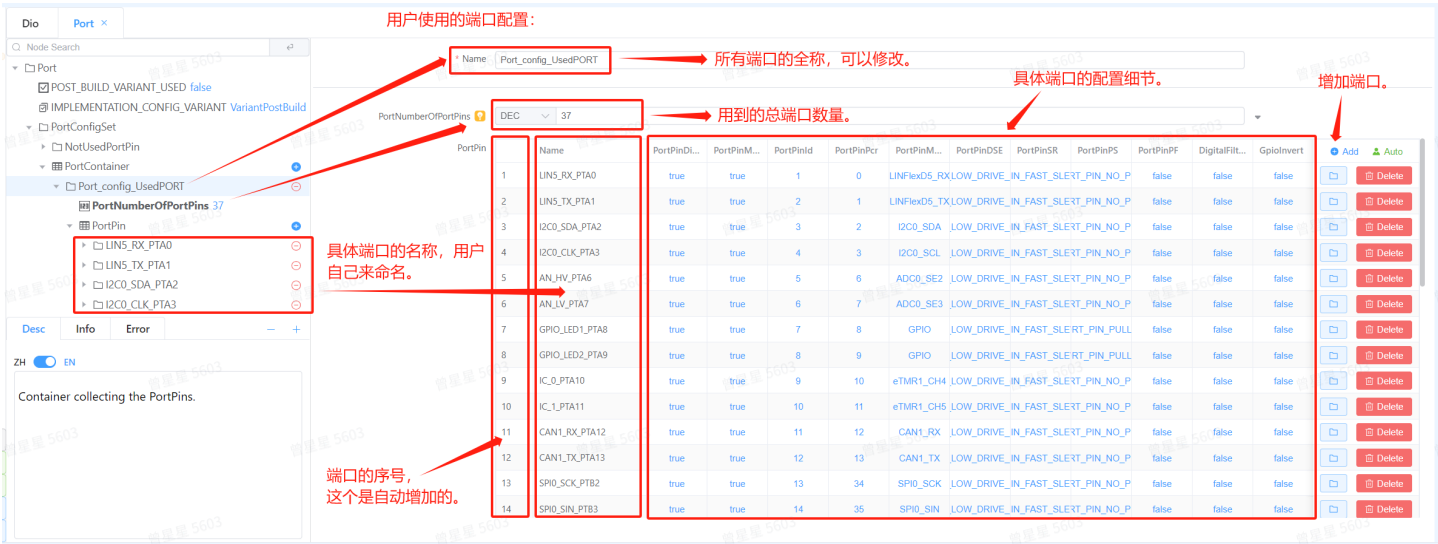
2.3 禁用端口的设置



Tips: 禁用的引脚是SWD/JTAG口，按照图片上的默认配置。

2.4 用户配置端口：

2.4.1 用户配置的所有端口预览：

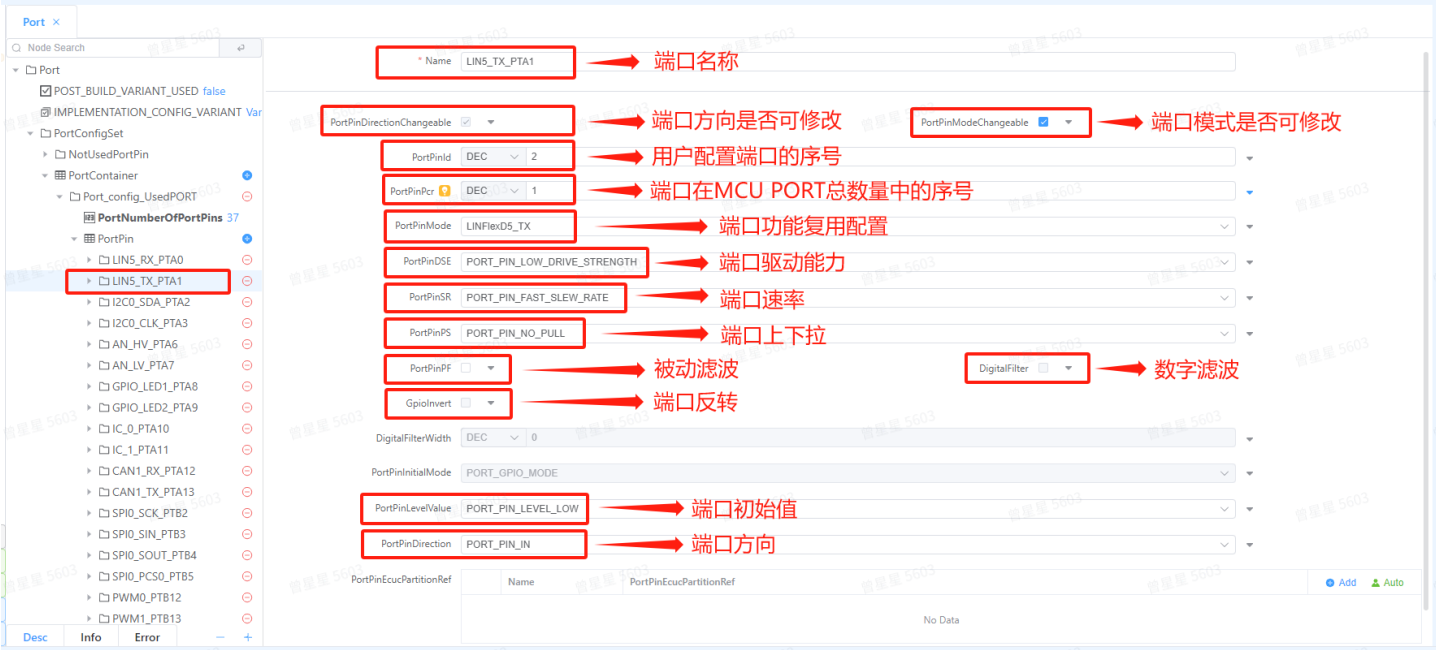


Tips: PortPinMode就是引脚复用功能的选择。

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

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

2.4.2.1 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：

=> passive filter, 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。

2.4.2.2 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

* Name LIN5_TX_PTA1

PortPin DEC 2

PortPinPcr DEC PTA1

PortPinMode LINFlexDS_TX

PortPinDSE PORT_PIN_LOW_DRIVE_STRENGTH

PortPinSR PORT_PIN_FAST_SLEW_RATE

PortPin DEC 2

PortPinPcr DEC 1

PortPinMode LINFlexDS_TX

PortPinDSE PORT_PIN_LOW_DRIVE_STRENGTH

PortPinSR PORT_PIN_FAST_SLEW_RATE

注意：我们MCU中PORT共有5个，分别为PORTA/B/C/D/E，对应的编号分别为0/1/2/3/4，每个PORT中分配32个pin脚，所以，编号0对应的是PTA0，编号31对应的是PTA31；编号32对应的是PTB0，编号63对应的是PTB31；编号64对应的是PTC0，编号95对应的是PTC31；编号96对应的是PTD0，编号127对应的是PTD31；编号128对应的是PTE0，编号159对应的是PTE31；

* Name SPI0_SIN_PTB3

PortPin DEC 14

PortPinPcr DEC PTB3

PortPinMode SPI0_SIN

PortPinDSE PORT_PIN_LOW_DRIVE_STRENGTH

PortPinSR PORT_PIN_FAST_SLEW_RATE

PortPin DEC 14

PortPinPcr DEC 35

PortPinMode SPI0_SIN

PortPinDSE PORT_PIN_LOW_DRIVE_STRENGTH

PortPinSR PORT_PIN_FAST_SLEW_RATE

2.4.2.3 端口配置完成汇总

PortPin	Name	PortPinDir	PortPinM...	PortPinID	PortPinPcr	PortPinM...	PortPinDSE	PortPinSR	PortPinPS	PortPinPF	DigitalFil...	Add	Auto	Delete
1	LIN5_RX_PTA0	true	true	1	0	LINFlexDS_RX	LOW_DRIVE_IN_FAST_SLERT_PIN_NO_P	false	false	false	false			
2	LIN5_TX_PTA1	true	true	2	1	LINFlexDS_TX	LOW_DRIVE_IN_FAST_SLERT_PIN_NO_P	false	false	false	false			
3	I2C0_SDA_PTA2	true	true	3	2	I2C0_SDA	LOW_DRIVE_IN_FAST_SLERT_PIN_NO_P	false	false	false	false			
4	I2C0_CLK_PTA3	true	true	4	3	I2C0_SCL	LOW_DRIVE_IN_FAST_SLERT_PIN_NO_P	false	false	false	false			
5	AN_HV_PTA6	true	true	5	6	ADC0_SE2	LOW_DRIVE_IN_FAST_SLERT_PIN_NO_P	false	false	false	false			
6	AN_LV_PTA7	true	true	6	7	ADC0_SE3	LOW_DRIVE_IN_FAST_SLERT_PIN_NO_P	false	false	false	false			
7	GPIO_LED1_PTA8	true	true	7	8	GPIO	LOW_DRIVE_IN_FAST_SLERT_PIN_PULL	false	false	false	false			
8	GPIO_LED2_PTA9	true	true	8	9	GPIO	LOW_DRIVE_IN_FAST_SLERT_PIN_PULL	false	false	false	false			
9	IC_0_PTA10	true	true	9	10	eTMR1_CH4	LOW_DRIVE_IN_FAST_SLERT_PIN_NO_P	false	false	false	false			
10	IC_1_PTA11	true	true	10	11	eTMR1_CH5	LOW_DRIVE_IN_FAST_SLERT_PIN_NO_P	false	false	false	false			
11	CAN1_RX_PTA12	true	true	11	12	CAN1_RX	LOW_DRIVE_IN_FAST_SLERT_PIN_NO_P	false	false	false	false			
12	CAN1_TX_PTA13	true	true	12	13	CAN1_TX	LOW_DRIVE_IN_FAST_SLERT_PIN_NO_P	false	false	false	false			
13	SPI0_SCK_PTB2	true	true	13	34	SPI0_SCK	LOW_DRIVE_IN_FAST_SLERT_PIN_NO_P	false	false	false	false			
14	SPI0_SIN_PTB3	true	true	14	35	SPI0_SIN	LOW_DRIVE_IN_FAST_SLERT_PIN_NO_P	false	false	false	false			

2.4.2.4 生成代码预览

用户配置端口生成的代码：

```
113 #define PortConfigSet_Port_config_UsedPORT_LIN5_RX_PTA0 ((Port_PinType)((0)<<8)|0)
114 #define PortConfigSet_Port_config_UsedPORT_LIN5_TX_PTA1 ((Port_PinType)((0)<<8)|1)
115 #define PortConfigSet_Port_config_UsedPORT_I2C0_SDA_PTA2 ((Port_PinType)((0)<<8)|2)
116 #define PortConfigSet_Port_config_UsedPORT_I2C0_CLK_PTA3 ((Port_PinType)((0)<<8)|3)
117 #define PortConfigSet_Port_config_UsedPORT_AN_HV_PTA6 ((Port_PinType)((0)<<8)|6)
118 #define PortConfigSet_Port_config_UsedPORT_AN_LV_PTA7 ((Port_PinType)((0)<<8)|7)
119 #define PortConfigSet_Port_config_UsedPORT_GPIO_LED1_PTA8 ((Port_PinType)((0)<<8)|8)
120 #define PortConfigSet_Port_config_UsedPORT_GPIO_LED2_PTA9 ((Port_PinType)((0)<<8)|9)
121 #define PortConfigSet_Port_config_UsedPORT_IC_0_PTA10 ((Port_PinType)((0)<<8)|10)
122 #define PortConfigSet_Port_config_UsedPORT_IC_1_PTA11 ((Port_PinType)((0)<<8)|11)
123 #define PortConfigSet_Port_config_UsedPORT_CAN1_RX_PTA12 ((Port_PinType)((0)<<8)|12)
124 #define PortConfigSet_Port_config_UsedPORT_CAN1_TX_PTA13 ((Port_PinType)((0)<<8)|13)
125 #define PortConfigSet_Port_config_UsedPORT_SPI0_SCK_PTB2 ((Port_PinType)((1)<<8)|2)
126 #define PortConfigSet_Port_config_UsedPORT_SPI0_SIN_PTB3 ((Port_PinType)((1)<<8)|3)
127 #define PortConfigSet_Port_config_UsedPORT_SPI0_SOUT_PTB4 ((Port_PinType)((1)<<8)|4)
128 #define PortConfigSet_Port_config_UsedPORT_PCS0_PTB5 ((Port_PinType)((1)<<8)|5)
129 #define PortConfigSet_Port_config_UsedPORT_PWM0_PTB12 ((Port_PinType)((1)<<8)|12)
130 #define PortConfigSet_Port_config_UsedPORT_PWM1_PTB13 ((Port_PinType)((1)<<8)|13)
131 #define PortConfigSet_Port_config_UsedPORT_PWM2_PTB14 ((Port_PinType)((1)<<8)|14)
132 #define PortConfigSet_Port_config_UsedPORT_PWM3_PTB15 ((Port_PinType)((1)<<8)|15)
133 #define PortConfigSet_Port_config_UsedPORT_PWM4_PTB16 ((Port_PinType)((1)<<8)|16)
134 #define PortConfigSet_Port_config_UsedPORT_PWM5_PTB17 ((Port_PinType)((1)<<8)|17)
135 #define PortConfigSet_Port_config_UsedPORT_AN_Curr1_PTC0 ((Port_PinType)((2)<<8)|0)
136 #define PortConfigSet_Port_config_UsedPORT_AN_Curr2_PTC1 ((Port_PinType)((2)<<8)|1)
137 #define PortConfigSet_Port_config_UsedPORT_FAN1_PTC30 ((Port_PinType)((2)<<8)|30)
138 #define PortConfigSet_Port_config_UsedPORT_FAN2_PTC31 ((Port_PinType)((2)<<8)|31)
139 #define PortConfigSet_Port_config_UsedPORT_XTAL32_PTD15 ((Port_PinType)((3)<<8)|15)
140 #define PortConfigSet_Port_config_UsedPORT_XTAL32_PTD16 ((Port_PinType)((3)<<8)|16)
141 #define PortConfigSet_Port_config_UsedPORT_ACMPO_PTD6 ((Port_PinType)((3)<<8)|6)
142 #define PortConfigSet_Port_config_UsedPORT_ACMPO_PTD7 ((Port_PinType)((3)<<8)|7)
143 #define PortConfigSet_Port_config_UsedPORT_GPIO_Relay1_PTE15 ((Port_PinType)((4)<<8)|15)
144 #define PortConfigSet_Port_config_UsedPORT_GPIO_Relay2_PTE16 ((Port_PinType)((4)<<8)|16)
145 #define PortConfigSet_Port_config_UsedPORT_XTAL_PTB6 ((Port_PinType)((1)<<8)|6)
146 #define PortConfigSet_Port_config_UsedPORT_EXTAL_PTB7 ((Port_PinType)((1)<<8)|7)
147 #define PortConfigSet_Port_config_UsedPORT_SWIO_PTA4 ((Port_PinType)((0)<<8)|4)
148 #define PortConfigSet_Port_config_UsedPORT_RESET_PTA5 ((Port_PinType)((0)<<8)|5)
149 #define PortConfigSet_Port_config_UsedPORT_SWCLK_PTC4 ((Port_PinType)((2)<<8)|4)
```

PTA1配置对应的代码：

```
58 /* PCR-1, Feature-LINFlexDS_TX */
59
60 {
61     .HwPinId = 1,
62     .Direction = PORT_PIN_IN,
63     .IsDirectionChangeable= PORT_PIN_DIRECTION_CHANGEABLE,
64     .InitLevel = PORT_PIN_LEVEL_LOW,
65     .InitMode= PORT_PIN_MODE_ALT3,
66     .IsModeChangeable = PORT_PIN_MODE_CHANGEABLE,
67     .PullConfig = PORT_PIN_NO_PULL,
68     .DriveStrength = PORT_PIN_LOW_DRIVE_STRENGTH,
69     .SlewRate = PORT_PIN_FAST_SLEW_RATE,
70     .PassiveFilter = (boolean)0,
71     .DigitalFilter = (boolean)0,
72     .DigitalFilterWidth = 0,
73     .InvertEnable = (boolean)0,
74 },
```

PTB3配置对应的代码：

```
487 /* PCR-35, Feature-SPI0_SIN */
488
489 {
490     .HwPinId = 3,
491     .Direction = PORT_PIN_IN,
492     .IsDirectionChangeable= PORT_PIN_DIRECTION_CHANGEABLE,
493     .InitLevel = PORT_PIN_LEVEL_LOW,
494     .InitMode= PORT_PIN_MODE_ALT3,
495     .IsModeChangeable = PORT_PIN_MODE_CHANGEABLE,
496     .PullConfig = PORT_PIN_NO_PULL,
497     .DriveStrength = PORT_PIN_LOW_DRIVE_STRENGTH,
498     .SlewRate = PORT_PIN_FAST_SLEW_RATE,
499     .PassiveFilter = (boolean)0,
500     .DigitalFilter = (boolean)0,
501     .DigitalFilterWidth = 0,
502     .InvertEnable = (boolean)0,
503 },
```

3. Port模块应用示例

3.1 PIN脚定义：

```
183 /** @{ */
184 #define DioConf_DioChannel_PTA5_RESET ((0U << 8U) + 5U)
185 #define DioConf_DioChannel_PTA4_SWIO ((0U << 8U) + 4U)
186 #define DioConf_DioChannel_PTB2_KEYSW2 ((1U << 8U) + 2U)
187 #define DioConf_DioChannel_PTB3_KEYSW3 ((1U << 8U) + 3U)
188 #define DioConf_DioChannel_PTB4_LED5 ((1U << 8U) + 4U)
189 #define DioConf_DioChannel_PTB5_LED6 ((1U << 8U) + 5U)
190 #define DioConf_DioChannel_PTB6_XTAL ((1U << 8U) + 6U)
191 #define DioConf_DioChannel_PTB7_EXTAL ((1U << 8U) + 7U)
192 #define DioConf_DioChannel_PTC4_SWCLK ((2U << 8U) + 4U)
193 #define DioConf_DioChannel_PTE8_LED7 ((4U << 8U) + 8U)
194 /** @} */
```

3.2 程序代码：


```

75  * @brief The application entry point.
76  * @retval int
77  */
78  int main(void)
79  {
80      /* USER CODE BEGIN 1 */
81
82      Mcu_Init(&Mcu_Config);
83      Mcu_InitClock(0);
84      #if (MCU_NO_PLL == STD_OFF)
85          while ( MCU_PLL_LOCKED != Mcu_GetPllStatus() )
86          {
87              /* Busy wait until the System PLL is locked */
88          }
89      Mcu_DistributePllClock();
90      #endif
91
92      /* USER CODE END 1 */
93      Board_Init();
94      /* USER CODE BEGIN 2 */
95
96      /* USER CODE END 2 */
97
98      /* Infinite loop */
99      /* USER CODE BEGIN WHILE */
100     while (1)
101     {
102         /* USER CODE END WHILE */
103         /* USER CODE BEGIN 3 */
104
105         if(Dio_ReadChannel(DioConf_DioChannel_PT2_KEYSW2) == STD_LOW)
106         {
107             /* turn on LED5/LED6/LED7 */
108             Dio_WriteChannel(DioConf_DioChannel_PT4_LED5, STD_LOW);
109             Dio_WriteChannel(DioConf_DioChannel_PT5_LED6, STD_LOW);
110             Dio_WriteChannel(DioConf_DioChannel_PTE8_LED7, STD_LOW);
111         }
112
113         /* delay 100ms */
114         Delay_Ms(100);
115         if(Dio_ReadChannel(DioConf_DioChannel_PT3_KEYSW3) == STD_LOW)
116         {
117             /* turn off LED5/LED6/LED7 */
118             Dio_WriteChannel(DioConf_DioChannel_PT4_LED5, STD_HIGH);
119             Dio_WriteChannel(DioConf_DioChannel_PT5_LED6, STD_HIGH);
120             Dio_WriteChannel(DioConf_DioChannel_PTE8_LED7, STD_HIGH);
121         }
122     }
123 }
124 /* USER CODE END 3 */
125
126
127 static void Board_Init(void)
128 {
129     Platform_Init(NULL_PTR);
130     Port_Init(&Port_Config);
131 }
132
133 /* USER CODE BEGIN 4 */
134 /* USER CODE END 4 */

```

3.3 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