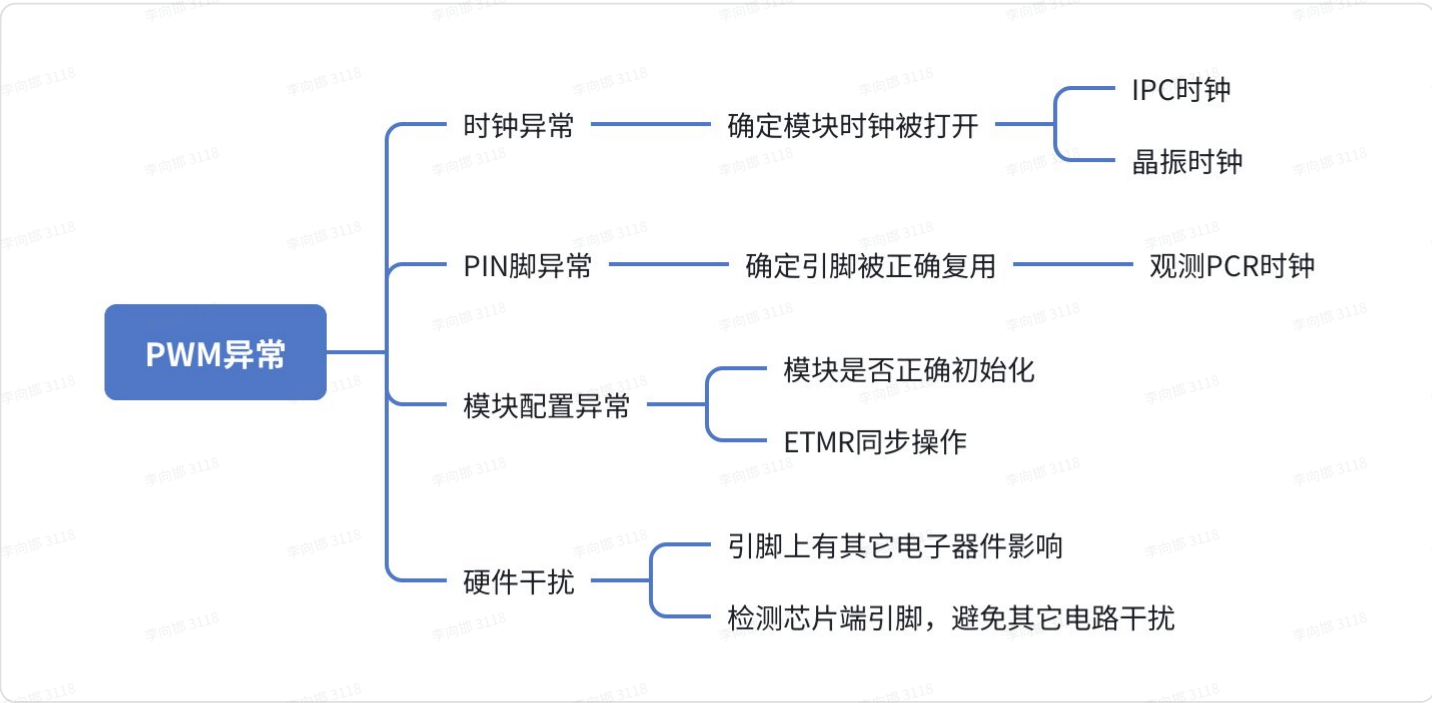


PWM输出异常排查指南



1. 确定相应ETMR时钟是否打开、

a. 在YCT界面确认

FlexCAN5_CLK	☐	☐		
ADC0_CLK	☐	☐	1 ▼	CLK_SRC_DISABLED
ADC1_CLK	☐	☐	1 ▼	CLK_SRC_DISABLED
ACMP0_CLK	☐	☐	1 ▼	CLK_SRC_DISABLED
PTU0_CLK	☐	☐		
PTU1_CLK	☐	☐		
eTMR0_CLK	☒	☒		
eTMR1_CLK	☐	☐		
eTMR2_CLK	☐	☐		
eTMR3_CLK	☐	☐		
eTMR4_CLK	☐	☐		
eTMR5_CLK	☐	☐		

b. 当选择内部时钟时，可以排查相应的寄存器是否已经打开，如果选择的是晶振，那就排查晶振是否已经起振

73	eTMR0	0x124	FAST_BUS_CLK	OFF	-	YES	TCLK0/1/2
74	eTMR1	0x128	FAST_BUS_CLK	OFF	-	YES	TCLK0/1/2
75	eTMR2	0x12C	FAST_BUS_CLK	OFF	-	YES	TCLK0/1/2
76	eTMR3	0x130	FAST_BUS_CLK	OFF	-	YES	TCLK0/1/2
77	eTMR4	0x134	FAST_BUS_CLK	OFF	-	YES	TCLK0/1/2
78	eTMR5	0x138	FAST_BUS_CLK	OFF	-	YES	TCLK0/1/2

124h	Control Register (CTRL73)	32	RW	00000000h
128h	Control Register (CTRL74)	32	RW	00000000h

Offset	Register	Width	Access	Reset Value
12ch	Control Register (CTRL75)	32	RW	00000000h
130h	Control Register (CTRL76)	32	RW	00000000h
134h	Control Register (CTRL77)	32	RW	00000000h
138h	Control Register (CTRL78)	32	RW	00000000h

2. 排查引脚是否配置成ETMR 通道，

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

3118

李向娜 3118

李向娜 3118

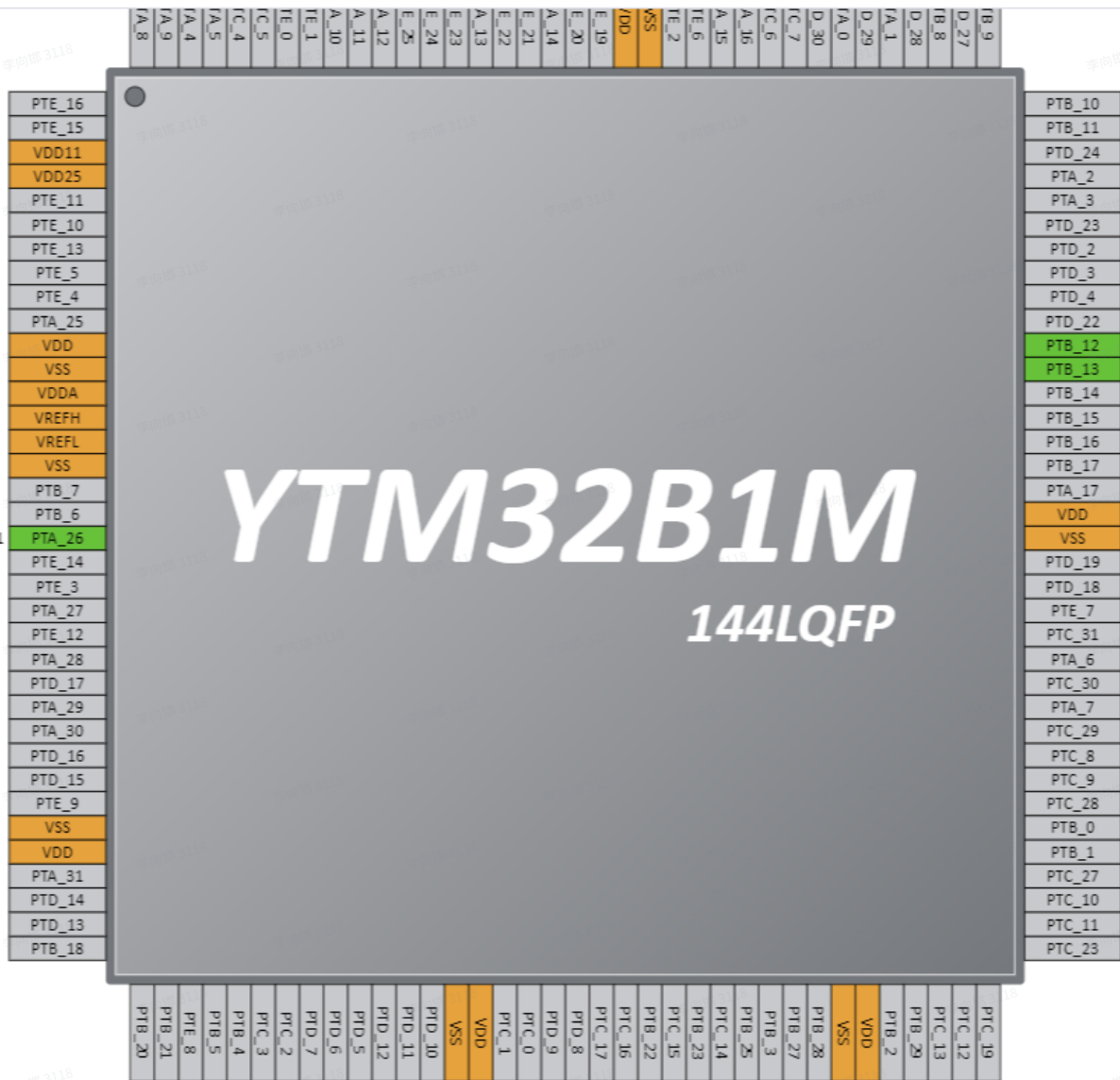
李向娜 3118

李向娜 3118

李向娜 3118

李向娜 3118

3118



可通过PCR寄存器确认引脚复用功能正确

Offset	Register	Width	Access	Reset Value
0h	Port Control Register (PCR0)	32	RW	see description
4h	Port Control Register (PCR1)	32	RW	see description
8h	Port Control Register (PCR2)	32	RW	see description
ch	Port Control Register (PCR3)	32	RW	see description
10h	Port Control Register (PCR4)	32	RW	see description
14h	Port Control Register (PCR5)	32	RW	see description
18h	Port Control Register (PCR6)	32	RW	see description
1ch	Port Control Register (PCR7)	32	RW	see description
20h	Port Control Register (PCR8)	32	RW	see description
24h	Port Control Register (PCR9)	32	RW	see description
28h	Port Control Register (PCR10)	32	RW	see description
2ch	Port Control Register (PCR11)	32	RW	see description
30h	Port Control Register (PCR12)	32	RW	see description
34h	Port Control Register (PCR13)	32	RW	see description
38h	Port Control Register (PCR14)	32	RW	see description

3. 确定时钟、引脚、模块均被正常调用；

同时需要考虑PWM同步；

```

62  /*
63  int main(void)
64  {
65      /* USER CODE BEGIN 1 */
66      /* USER CODE END 1 */
67      Board_Init();
68      /* USER CODE BEGIN 2 */
69
70      /* PWM is generated when eTMR enable, the duty cycle is configured in the initial function */
71      eTMR_DRV_Init(0, &ETMR_CM_Config0, &ETMR_CM_Config0_State);
72      eTMR_DRV_InitPwm(0, &ETMR_PWM_Config0);
73
74      eTMR_DRV_Enable(ETMR_INST);
75      OSIF_TimeDelay(10);
76
77      /* The following code shows how to switch the different duty cycle */
78      /* channel 0 duty cycle is 50% */
79      eTMR_DRV_UpdatePwmChannel(ETMR_INST, ETMR_CHANNEL_INDEX0, 0x4000U, 0);
80      /* channel 1 duty cycle is 75% */
81      eTMR_DRV_UpdatePwmChannel(ETMR_INST, ETMR_CHANNEL_INDEX1, 0x6000U, 0);
82      eTMR_DRV_SetLdok(ETMR_INST);
83      OSIF_TimeDelay(10);
84
85      /* channel 0 & 1 duty cycle is 100%, here is the workaround for 100% duty cycle */
86      g_etmrBase[ETMR_INST]->CH[ETMR_CHANNEL_INDEX0].VAL0 = 0;
87      g_etmrBase[ETMR_INST]->CH[ETMR_CHANNEL_INDEX0].VAL1 = g_etmrBase[ETMR_INST]->MOD + 1;
88      g_etmrBase[ETMR_INST]->CH[ETMR_CHANNEL_INDEX1].VAL0 = 0;
89      g_etmrBase[ETMR_INST]->CH[ETMR_CHANNEL_INDEX1].VAL1 = g_etmrBase[ETMR_INST]->MOD + 1;
90      eTMR_DRV_SetLdok(ETMR_INST);
91      OSIF_TimeDelay(10);
92
93      /* channel 0 duty cycle is 0 */
94      eTMR_DRV_UpdatePwmChannel(ETMR_INST, ETMR_CHANNEL_INDEX0, 0, 0);
95      /* channel 1 duty cycle is 0 */
96      eTMR_DRV_UpdatePwmChannel(ETMR_INST, ETMR_CHANNEL_INDEX1, 0, 0);
97      eTMR_DRV_SetLdok(ETMR_INST);
98      OSIF_TimeDelay(10);
99
100     /* channel 0 duty cycle is 0x6000 */
101     eTMR_DRV_UpdatePwmChannel(ETMR_INST, ETMR_CHANNEL_INDEX0, 0x6000U, 0);
102     /* channel 1 duty cycle is 0x4000 */
103     eTMR_DRV_UpdatePwmChannel(ETMR_INST, ETMR_CHANNEL_INDEX1, 0x4000U, 0);
104     eTMR_DRV_SetLdok(ETMR_INST);
105
106     /* USER CODE END 2 */
107
108     /* Infinite loop */
109     /* USER CODE BEGIN WHILE */
110     while (1)
111     {
112         /* USER CODE END WHILE */
113         /* USER CODE BEGIN 3 */
114     }

```

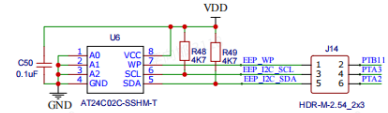
4. 排查外部引脚干扰，以Demo板为例，所选用的引脚没有被其它电路干扰

Encoder & 3-phase Motor Control Interface + SPI for pre-driver

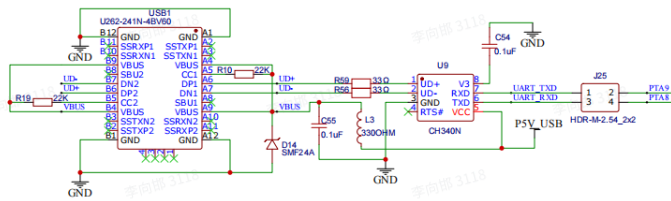


Notes: only support 5V QD Encoder, and use the same placement & layout to be compatible with YTM32B1L EVB

I2C EEPROM Memory



USB(Type-C) UART Debug Interface with 1.5A@5V



SPI FRAM

