

SDK应用_eTMR 模块配置及应用

版本:

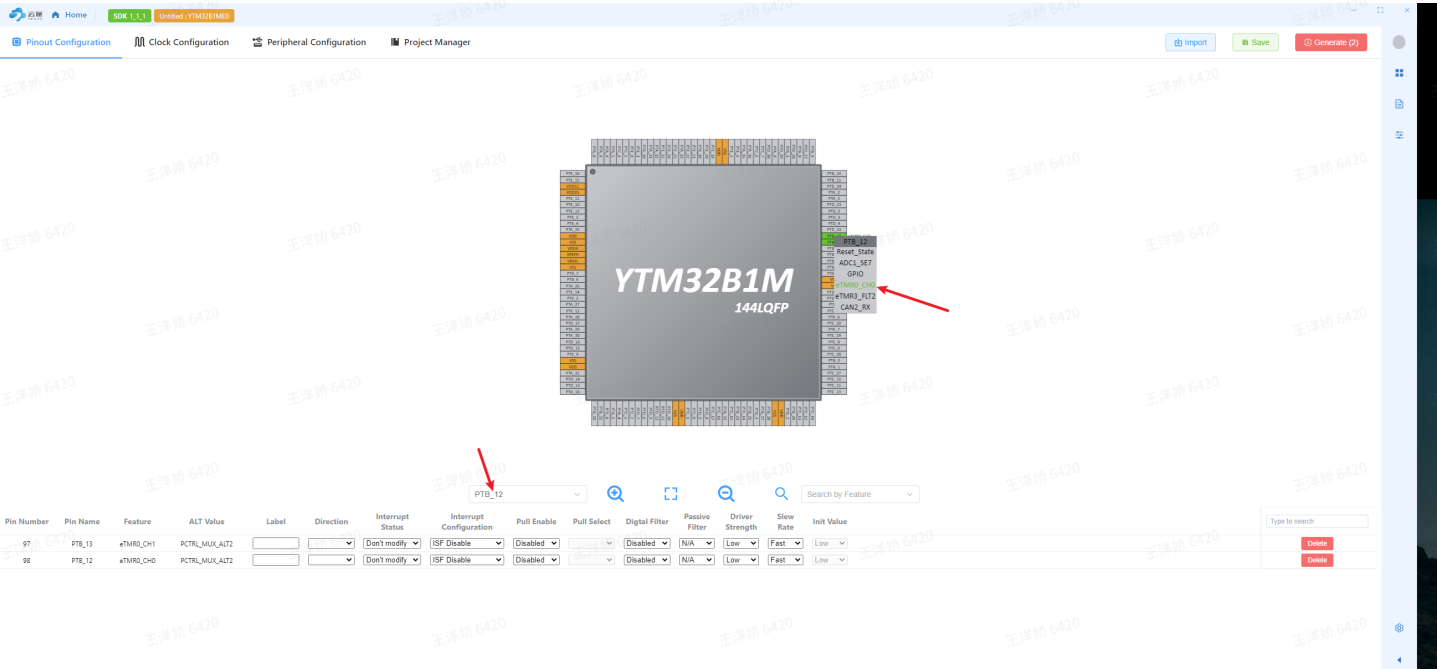
Config Tool Version: 2.0.2

ME0 SDK version: 1_1_1

简介

eTMR是增强型定时器，它可以当作普通定时器，可以产生任意占空比的pwm，可以输出固定电平，也可以捕获脉冲宽度，捕获信号周期等。

引脚配置(Pinout Configuration)

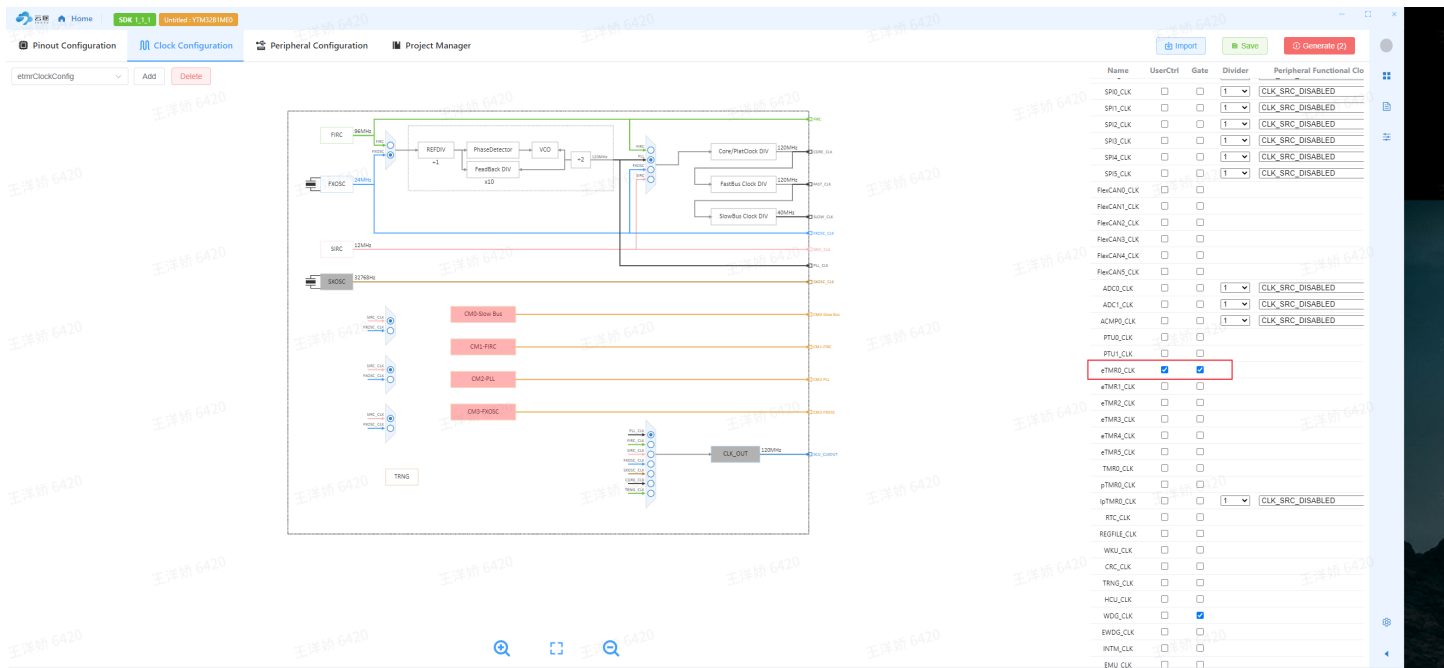


如图，在图片的左下角输入对应的pin，然后左键点击该pin，选择eTMR0_CH0。

时钟配置(Clock Configuration)

ME0的etmr的时钟源分为 fast bus clock 与 external clock。fast bus clock是由系统时钟经过fast bus clock divider分频之后得到的时钟源，而external clock是通过配置为TCLK_INx功能的pin来往里输入时钟信号。

ME0 etmr内部时钟源固定为fast bus clock，因此IPC内部的时钟源选项是无用的。

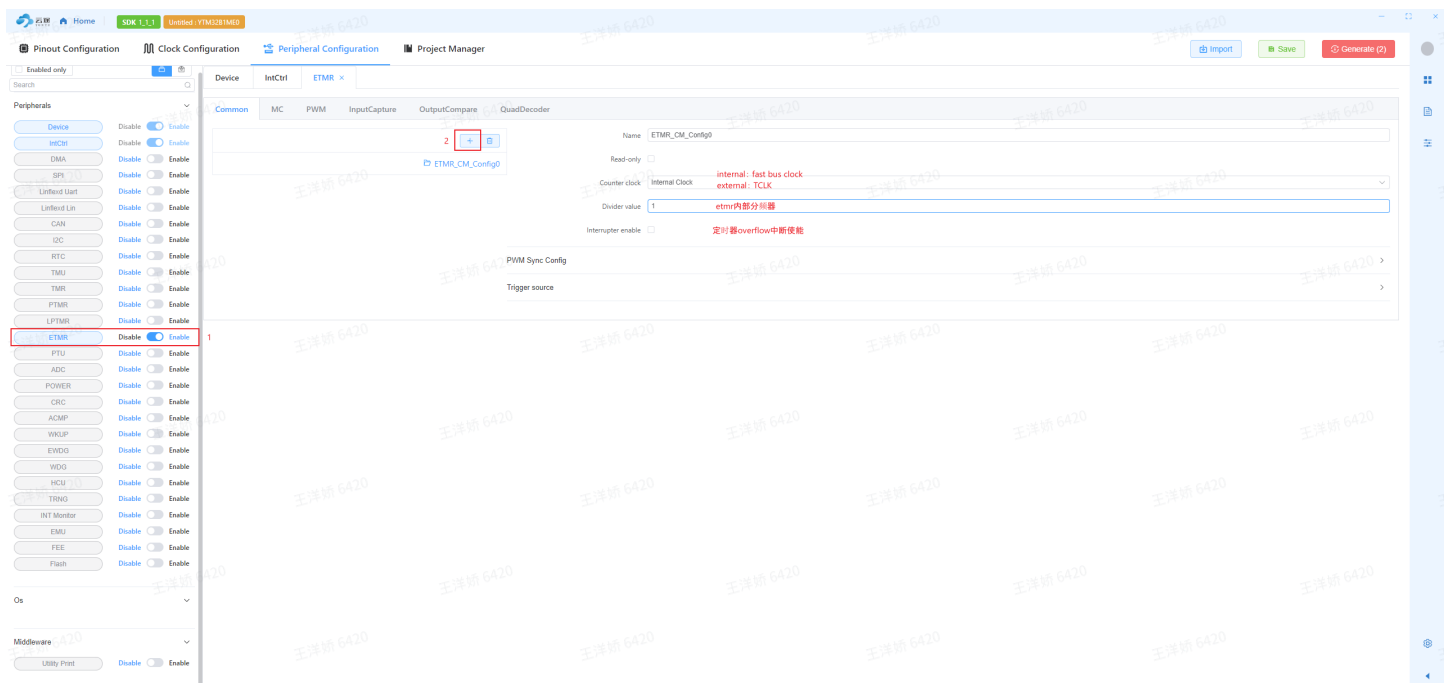


如图，将eTMRx_CLK的Gate和Divider勾选上。

eTMR模块配置

普通定时器配置

1. 首先配置Common



如图：选择internal clock，分频系数为1(即不分频)，使能overflow中断

2. 配置MC

Common

MC

PWM

InputCapture

OutputCompare

QuadDecoder

Disable

Enable

+

ETMR_MC_Config0

Name

ETMR_MC_Config0

Read-only

Counter init value

0

Counter final value

1000

如图配置MOD值为1000

3. API

SDK API Init Sequence

1:CLOCK_SYS_Init(g_clockManConfigsArr,CLOCK_MANAGER_CONFIG_CNT,g_d
ockManCallbacksArr,CLOCK_MANAGER_CALLBACK_CNT);

2:CLOCK_SYS_UpdateConfiguration(CLOCK_MANAGER_ACTIVE_INDEX,CLOCK_
MANAGER_POLICY_AGREEMENT);

3:PINS_DRV_Init(NUM_OF_CONFIGURED_PINS0,g_pin_mux_InitConfigArr0);

4:eTMR_DRV_Init(0,&ETMR_CM_Config0,&ETMR_CM_Config0_State);

5:eTMR_DRV_InitCounter(0,&ETMR_MC_Config0);

如图，在board_init里

eTMR_DRV_Init()	eTMR common初始化
eTMR_DRV_InitCounter()	eTMR 计数器初始化

在main函数中使用eTMR_DRV_CounterStart()来开启计数器。如下图。

```
51 /* Private user code -----*/
52 /* USER CODE BEGIN 0 */
53 void etmr_ovf_handler(void)
54 {
55 }
56 /* USER CODE END 0 */
57
58
59
60 /**
61  * @brief The application entry point.
62  * @retval int
63  */
64 int main(void)
65 {
66     /* USER CODE BEGIN 1 */
67     INT_SYS_InstallHandler(eTMR0_Ovf_IRQn, etmr_ovf_handler, NULL);
68     /* USER CODE END 1 */
69     Board_Init();
70     /* USER CODE BEGIN 2 */
71     eTMR_DRV_CounterStart(0);
72     /* USER CODE END 2 */
73
74     /* Infinite loop */
75     /* USER CODE BEGIN WHILE */
76     while (1)
77     {
78         /* USER CODE END WHILE */
79         /* USER CODE BEGIN 3 */
80     }
81     /* USER CODE END 3 */
82 }
83
84 static void Board_Init(void)
85 {
86     CLOCK_SYS_Init(g_clockManConfigsArr,CLOCK_MANAGER_CONFIG_CNT,g_clockManCallbacksArr,CLOCK_MANAGER_CALLBACK_CNT);
87     CLOCK_SYS_UpdateConfiguration(CLOCK_MANAGER_ACTIVE_INDEX,CLOCK_MANAGER_POLICY_AGREEMENT);
88     PINS_DRV_Init(NUM_OF_CONFIGURED_PINS0,g_pin_mux_InitConfigArr0);
89     eTMR_DRV_Init(0,&ETMR_CM_Config0,&ETMR_CM_Config0_State);
90     eTMR_DRV_InitCounter(0,&ETMR_MC_Config0);
91 }
```

PWM配置

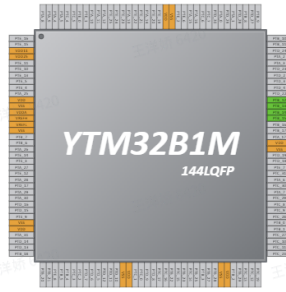
1. Pinout 配置eTMR0_CH0, eTMR0_CH1, eTMR0_CH2, eTMR0_CH3, 如图。

Pinout Configuration

Clock Configuration

Peripheral Configuration

Project Manager



PTB_12

eTMR0_CH3

Pin Number	Pin Name	Feature	ALT Value	Label	Direction	Interrupt Status	Interrupt Configuration	Pull Enable	Pull Select	Digital Filter	Passive Filter	Driver Strength	Slew Rate	Init Value
98	PTB_12	eTMR0_CH0	PCTRL_MUX_ALT2			Don't modify	ISF Disable	Disabled		Disabled	N/A	Low	Fast	Low
97	PTB_13	eTMR0_CH1	PCTRL_MUX_ALT2			Don't modify	ISF Disable	Disabled		Disabled	N/A	Low	Fast	Low
96	PTB_14	eTMR0_CH2	PCTRL_MUX_ALT2			Don't modify	ISF Disable	Disabled		Disabled	N/A	Low	Fast	Low
95	PTB_15	eTMR0_CH3	PCTRL_MUX_ALT2			Don't modify	ISF Disable	Disabled		Disabled	N/A	Low	Fast	Low

2. eTMR0 时钟配置，如图

PTU0_CLK	☐	☐
PTU1_CLK	☐	☐
eTMR0_CLK	☒	☒
eTMR1_CLK	☐	☐
-TMR2_CLK	☐	☐

3. Etmr Common配置

Pinout Configuration

Clock Configuration

Peripheral Configuration

Project Manager

Enabled only

Search

Peripherals

Device

Disable

Enable

IntCtrl

Disable

Enable

DMA

Disable

Enable

SPI

Disable

Enable

Linflexd Uartf

Disable

Enable

Linflexd Lin

Disable

Enable

CAN

Disable

Enable

I2C

Disable

Enable

RTC

Disable

Enable

TMU

Disable

Enable

TMR

Disable

Enable

PTMR

Disable

Enable

LPTMR

Disable

Enable

ETMR

Disable

Enable

PTU

Disable

Enable

ADC

Disable

Enable

POWER

Disable

Enable

CRC

Disable

Enable

ACMP

Disable

Enable

WKUP

Disable

Enable

EWDOG

Disable

Enable

WDG

Disable

Enable

HCU

Disable

Enable

TRNG

Disable

Enable

INT Monitor

Disable

Enable

EMUJ

Disable

Enable

FEE

Disable

Enable

Flash

Disable

Enable

Os

Middleware

Utility Print

Disable

Enable

Device

IntCtrl

ETMR

Common

MC

PWM

InputCapture

OutputCompare

QuadDecoder

ETMR_CM_Config0

Name

ETMR_CM_Config0

Read-only

Counter clock

Internal Clock

Divider value

1

Interrupter enable

PWM Sync Config

Register loading frequency

1

1表示每加载1次。影子寄存器变化一次；2表示每加载2次。影子寄存器变化一次

Register loading opportunity

Synchronized with counter matching MOD

表示counter计到MOD后进行加载。加载flag置1

Counter initial loading opportunity

Synchronized with register loading

表示跟随MOD、INIT、VAL0、VAL1寄存器的变化而变化

Mask output loading opportunity

Synchronized with register loading

表示跟随MOD、INIT、VAL0、VAL1寄存器的变化而变化

Register loading trigger source

Disable trigger

禁用trigger

Counter initial loading trigger source

Disable trigger

Mask output loading trigger source

Disable trigger

Hardware trigger from TMUJ

这里硬件trigger才使用的到

Hardware trigger from CIM

Hardware trigger from pad

Trigger source

Trigger source

TRIGGER_FROM_MATCHING_EVENT

trigger来自匹配事件

Output trigger frequency

0

0表示每出现匹配一次。输出一个trigger；1表示没出现2次匹配。输出一个trigger

PWM output channel

0

这里根据Trigger source来选择。如果是来自匹配事件，则不需要选择。如果是来自pwm，那么这里需要选择是哪个channel的pwm

Output trigger width

0

0表示1个bus clock的脉宽

Enable mod match trigger

表示在MOD点的时候产生一次匹配事件

Enable mid match trigger

表示在MID点的时候产生一次匹配事件

Enable init match trigger

表示在INIT点的时候产生一次匹配事件

Channel ID

Val #0 Trigger Enable

Val #1 Trigger Enable

如上图：
pwm更新在MOD点，同时在MOD点输出一个trigger

4. eTMR PWM配置

a. 如下图，通道数根据需求选择个数，设置PWM周期，设置计数器开始计数值

Name ETMR_PWM_Config0

Read-only ☐

Number of independent PWM channels 配置通道数

PWM mode PWM mode

PWM period in Hz 10000 10KHz

Counter initial value from INIT register(INIT=0) ☐

Counter initial value from CNT register 1 CNT寄存器值为1, 先从CNT寄存器值开始计数

b. 如下图

Independent Channel Config 0

Hardware channel ID 0 通道ID

Polarity of the channel signal Normal 整个通道的信号极性是否翻转

Polarity of the channel PWM ☐ 通道的pwm信号是否翻转, 不包括初始状态

PWM alignment mode Right pwm对齐方式

Channel output initial enable 0 通道初始值

PWM type of update eTMR_PWM_UPDATE_IN_DUTY_CYCLE 占空比更新方式 duty cycle 或者 ticks

PWM duty cycle HEX 0x2000 $0x2000 / 0x8000 = 25\%$

Offset for asymmetrical mode 0 非asymmetrical对齐方式, 这里不用管

Enable complementary mode on next channel ☐ 互补模式

Polarity of the complementary odd channel signal Normal 互补模式下奇数通道是否翻转

c. 如下图

Fault Channel Config

Enable PWM fault interrupt ☐ fault事件中断使能, 产生fault事件后是否进入中断

Fault filter sample counter 0 fault信号过滤采样数

Fault filter sample period 0 fault信号过滤采样频率

Fault input stretch 0 fault信号放大

PWM Recovery opportunity Disable pwm恢复时机

PWM Recovery automatic mode Recovery needs clearing fault flag first pwm自动恢复模式

Fault mode Clock is still existing 故障模式

Row No.	Fault channel enable	Fault input polsignal	
1	☐	High	故障通道设置
2	☐	High	
3	☐	High	
4	☐	High	

Row No.	Fault state	
1	eTMR_LOW_STATE	发生故障后通道信号值, 其优先级最高, 直到故障解除
2	eTMR_LOW_STATE	
3	eTMR_LOW_STATE	
4	eTMR_LOW_STATE	

5. API

eTMR_DRV_InitPwm()	初始化pwm
eTMR_DRV_Enable()	使能eTMR
eTMR_DRV_SetLdok()	设置load ok, 表示等待更新时机的到来
eTMR_DRV_UpdatePwmChannel()	更新pwm占空比
eTMR_DRV_UpdatePwmPeriod()	更新pwm频率
eTMR_DRV_UpdatePwmPeriodAndDuty()	更新pwm频率和占空比

```

63 int main(void)
64 {
65     /* USER CODE BEGIN 1 */
66     /* USER CODE END 1 */
67     Board_Init();
68     /* USER CODE BEGIN 2 */
69     eTMR_DRV_Enable(0);
70     OSIF_TimeDelay(10); // delay 10ms
71     eTMR_DRV_UpdatePwmChannel(0, ETMR_PWM_Config0.pwmChannelConfig[1].hwChannelId, 0x4000U, 0);
72     eTMR_DRV_SetLdok(0);
73     /* USER CODE END 2 */
74
75     /* Infinite loop */
76     /* USER CODE BEGIN WHILE */
77     while (1)
78     {
79         /* USER CODE END WHILE */
80         /* USER CODE BEGIN 3 */
81     }
82     /* USER CODE END 3 */
83 }
84
85 static void Board_Init(void)
86 {
87     CLOCK_SYS_Init(g_clockManConfigsArr,CLOCK_MANAGER_CONFIG_CNT,g_clockManCallbacksArr,CLOCK_MANAGER_CALLBACK_CNT);
88     CLOCK_SYS_UpdateConfiguration(CLOCK_MANAGER_ACTIVE_INDEX,CLOCK_MANAGER_POLICY_AGREEMENT);
89     PINS_DRV_Init(NUM_OF_CONFIGURED_PINS0,g_pin_mux_InitConfigArr0);
90     eTMR_DRV_Init(0,&ETMR_CM_Config0,&ETMR_CM_Config0_State);
91     eTMR_DRV_InitPwm(0,&ETMR_PWM_Config0);
92 }

```

输出比较配置(Output Compare)

1. Pinout 配置如上
2. etmr时钟配置如上
3. Etmr common配置如上
4. Output compare配置如下

Name

Read-only ☒ **加const**

Number of output compare channels **通道数**

Counter initial value from INIT register(INIT=0) ☐

Counter initial value from CNT register **计数器初始值**

Maximum counter value **计数器MOD**

Output compare channels configuration 0

Hardware channel ID **通道ID**

Init value **通道初始值**

Matching output mode (VAL0) **当匹配时置1**

Compared value (VAL0) **匹配点为VAL0=500**

Matching output mode (VAL1) **当val1匹配时通道值为0**

Compared value (VAL1) **匹配点为VAL1=750**

External trigger (VAL0) ☒ **使能val0匹配时输出脉冲**

External trigger (VAL1) ☒ **使能val1匹配时输出脉冲**

Enable interrupt ☐ **使能匹配事件中断**

5. API

eTMR_DRV_Init()	common 初始化
eTMR_DRV_InitOutputComapre()	oc初始化
eTMR_DRV_DeinitOutputCompare()	oc反初始化

输入捕获配置(Input Capture)

1. pinout配置
2. etmr时钟配置
3. Etmr common配置

4. Input capture配置

NameETMR_IC_Config0

Read-only☐

Number of input capture channels

-

2

+

通道数

Maximum counter value65535计数器最大值

Input capture channels configuration 0

Hardware channel ID0通道ID

Edge alignment ModePositive上升沿

Capture hold enable☐当抓取到边沿时将计数器保持住

DMA enable☐DMA使能

Filter Sample Counter0过滤器采样个数

Filter Sample Period0过滤器采样频率

Enable interrupt☐使能抓取事件中断

Enable notification☐使能通知，配置回调函数

CallbackNULL

Callback ParamsNULL

5. API

eTMR_DRV_InitInputCapture()	IC 初始化
eTMR_DRV_DeinitInputCapture()	IC 反初始化