

定时器入门之一

定时器输出占空比为 50%的方波

GPTM 库函数

定时器的时基及计数方式初始化函数：

```
void GPTM_TimeBaseInit ( GPTM_TypeDef * GPTMx,  
    GPTM_TimeBaseInitTypeDef * TimeBaseInit  
)
```

初始化 GPTM 的计数重载寄存器 CCR，预分频，计数模式。

参数：

GPTMx: 选择 GPTM 中的哪一个, x 可以是 0 或 1. , 为 0 代表 GPTM0, 为 1 代表 GPTM1。

TimeBaseInit: 指向包含配置信息的 GPTM_TimeBaseInitTypeDef 类型结构指针。

返回值：无

GPTM_TimeBaseInitTypeDef的定义

typedef struct

```
{  
    GPTM_CNT_MODE_Enum CounterMode;      计数模式  
    u16 CounterReload;                  计数重载寄存器  
    u16 Prescaler;                      预分频寄存器  
    GPTM_PSC_RLD_Enum PSCReloadTime;     预分频器重载模式  
} GPTM_TimeBaseInitTypeDef;
```

CounterMode: 是以下几个取值之一

GPTM_CNT_MODE_UP 边沿对齐向上计数模式

GPTM_CNT_MODE_CA1 中心对齐模式1 模式

GPTM_CNT_MODE_CA2 中心对齐模式2 模式

GPTM_CNT_MODE_CA3 中心对齐模式3 模式

GPTM_CNT_MODE_DOWN 边沿对齐向下计数模式

CounterReload: CCR寄存器的值，取值为（0~65535），这个值决定定时器的最大计数范围。

Prescaler: 预分频寄存器的值，取值为（0~65535），当使能预分频时，定时器的输入脉冲被（Prescaler+1）分频。

PSCReloadTime: 预分频计数器重载模式，下面2个取值之一

GPTM_PSC_RLD_UPDATE 下一个更新事件重载预分频寄存器

GPTM_PSC_RLD_IMMEDIATE 立即重载预分频寄存器

定时器的输出控制初始化函数

```
void GPTM_OutputInit ( GPTM_TypeDef * GPTMx,  
    GPTM_OutputInitTypeDef * OutInit  
)
```

初始化GPTMx的通道x输出

参数：

GPTMx: 选择GPTM中的哪一个, x 可以是 0 或 1. , 为 0 代表 GPTM0, 为 1 代表 GPTM1。

OutInit: 指向包含配置信息的GPTM_OutputInitTypeDef类型结构指针。

返回值：无

GPTM_OutputInitTypeDef的定义

typedef struct

```
{
    GPTM_CH_Enum      Channel;    通道选择
    GPTM_CHCTL_Enum   Control;    输出控制
    GPTM_CHP_Enum     Polarity;   极性控制
    GPTM_OM_Enum      OutputMode; 输出模式
    u16               Compare;     /*!< Value for CHxCCR register */
} GPTM_OutputInitTypeDef;
```

Channel: 通道选择，是下面取值之一

GPTM_CH_0 选择 GPTM 通道 0

GPTM_CH_1 选择 GPTM 通道 1

GPTM_CH_2 选择 GPTM 通道 2

GPTM_CH_3 选择 GPTM 通道 3

Control: 下面 2 个取值之一

GPTM_CHCTL_DISABLE GPTM 通道禁止

GPTM_CHCTL_ENABLE GPTM 通道使能

Polarity: 极性，下面的 2 个取值之一

GPTM_CHP_NONINVERTED GPTM 通道极性是高电平或上升沿

GPTM_CHP_INVERTED GPTM 通道极性是低电平或下降沿

OutputMode: 输出模式，下面的几个取值之一

GPTM_OM_MATCH_NOCHANGE 发生匹配时 GPTM 通道输出不变化

GPTM_OM_MATCH_INACTIVE 发生匹配时 GPTM 通道输出低电平

GPTM_OM_MATCH_ACTIVE 发生匹配时 GPTM 通道输出高电平

GPTM_OM_MATCH_TOGGLE 发生匹配时 GPTM 通道输出翻转

GPTM_OM_FORCED_INACTIVE GPTM 通道输出强制为低电平

GPTM_OM_FORCED_ACTIVE GPTM 通道输出强制为高电平

GPTM_OM_PWM1 GPTM 通道 pwm1 输出模式

GPTM_OM_PWM2 GPTM 通道 pwm2 输出模式

Compare: CHxCCR 寄存器的值，取值范围为 0~65535.

使能 GPTM

```
void GPTM_Cmd ( GPTM_TypeDef * GPTMx,
    ControlStatus NewState
)
```

使能或禁止 GPTM 计数.

参数:

GPTMx,: 选择 GPTM 中的哪一个, x 可以是 0 或 1. , 为 0 代表 GPTM0, 为 1 代表 GPTM1.

NewState,: 这个参数是 ENABLE 或 DISABLE.

返回值: 无

定时器输出方波

使用了以下库函数

1. 设置时基

GPTM_TimeBaseInit(GPTM0, &TimeBaseInit);

2. 设置定时器的输出方式

```
GPTM_OutputInit(GPTM0, &OutInit);
```

定义初始化定时器定时结构变量

```
GPTM_TimeBaseInitTypeDef TimeBaseInit;
```

定义初始化定时器输出结构变量

```
GPTM_OutputInitTypeDef OutInit;
```

使能复用 IO 时钟 AFIO

```
CKCU_APBPerip0ClockConfig(CKCU_APBEN0_AFIO, ENABLE);
```

使能 GPTM0 时钟

```
CKCU_APBPerip1ClockConfig(CKCU_APBEN1_GPTM0, ENABLE);
```

E 口的 5,6,7,8 做定时器 0 的输出引脚

```
AFIO_GPEConfig(AFIO_PIN_5 | AFIO_PIN_6 | AFIO_PIN_7 | AFIO_PIN_8, AFIO_MODE_2);
```

初始化定时器 0 时基

```
TimeBaseInit.CounterMode = GPTM_CNT_MODE_UP;
```

```
TimeBaseInit.CounterReload = 65535;
```

```
TimeBaseInit.Prescaler = 0;
```

```
TimeBaseInit.PSCReloadTime = GPTM_PSC_RLD_IMMEDIATE;
```

```
GPTM_TimeBaseInit(GPTM0, &TimeBaseInit);
```

计算输出频率参数

```
CH0_Frequency = 1000;
```

```
CH1_Frequency = 2000;
```

```
CH2_Frequency = 3000;
```

```
CH3_Frequency = 4000;
```

```
CH0_DeltaCCR = SystemCoreClock / (CH0_Frequency * 2 * (TimeBaseInit.Prescaler+1));
```

```
CH1_DeltaCCR = SystemCoreClock / (CH1_Frequency * 2 * (TimeBaseInit.Prescaler+1));
```

```
CH2_DeltaCCR = SystemCoreClock / (CH2_Frequency * 2 * (TimeBaseInit.Prescaler+1));
```

```
CH3_DeltaCCR = SystemCoreClock / (CH3_Frequency * 2 * (TimeBaseInit.Prescaler+1));
```

配置定时器的输出

```
OutInit.Channel = GPTM_CH_0;
```

```
OutInit.OutputMode = GPTM_OM_MATCH_TOGGLE;
```

```
OutInit.Control = GPTM_CHCTL_ENABLE;
```

```
OutInit.Polarity = GPTM_CHP_NONINVERTED;
```

```
OutInit.Compare = CH0_DeltaCCR;
```

```
GPTM_OutputInit(GPTM0, &OutInit);
```

```
OutInit.Channel = GPTM_CH_1;
```

```
OutInit.Compare = CH1_DeltaCCR;
```

```
GPTM_OutputInit(GPTM0, &OutInit);
```

```
OutInit.Channel = GPTM_CH_2;
```

```
OutInit.Compare = CH2_DeltaCCR;
```

```
GPTM_OutputInit(GPTM0, &OutInit);
```

```
OutInit.Channel = GPTM_CH_3;  
OutInit.Compare = CH3_DeltaCCR;  
GPTM_OutputInit(GPTM0, &OutInit);
```

使能定时器中断

```
GPTM_IntConfig(GPTM0, GPTM_INT_CH0CC | GPTM_INT_CH1CC | GPTM_INT_CH2CC |  
GPTM_INT_CH3CC, ENABLE);  
NVIC_EnableIRQ(GPTM0_IRQn);
```

启动定时器

```
GPTM_Cmd(GPTM0, ENABLE);
```

3.配置定时器的中断