



BAT32G137（库函数版本）

Rev 1.0

修订历史

版本	日期	修订人	修订内容
Rev1.1	22.6.27	缪勤文 张刚	

目 录

1.前言	3
2.通用定时器单元功能	3
3.中微 BAT32G137 TIMER4 应用库简介	3
3.1.应用例程使用	4
3.1.1. 方波输出	4
3.1.2.外部事件计数	5
3.1.3. 输入脉冲间隔测量	6
3.1.4. 间隔定时器	8
3.1.5. 多通道主从联动输出可调 PWM	9
3.1.6. 多通道主从联动输出单脉冲	10
4.示例演示	12

1. 前言

BAT32G137 具有 1 个通用定时器单元 Timer4，以下定义为：Timer40;对于 Timer40 时单元有 4 个定时通道，每个通道含有一个 16 位定时器。

2. 通用定时器单元功能

以 BAT32G137 为例，Timer4 具有以下功能：

- (1) 间隔定时器：能以固定间隔产生中断的基准定时器(递减计数)
- (2) 方波输出：(递减计数)
- (3) 外部事件计数器(递减计数)
- (4) 分频(递减计数)
- (5) 输入脉冲的间隔测量(递增计数)
- (6) 输入信号高低电平宽度测量(递增计数)
- (7) 延时计数器：在定时器输入引脚输入信号的有效边沿开始计数，并经过任意延迟后产生中断
- (8) 多通道主从联动，进行 pwm 输出(递减计数)
- (9) 主从通道联动，输出单脉冲

3. 中微 BAT32G137 TIMER4 应用库简介

中微 BAT32G137 Timer 应用库是一个便于移植的标准库代码风格，用户只需要配置软件接口以及相关定时器参数进行配置、以及封装接口函数调用即可实现所需功能，节约时间，提高开发效率。应用库提供了以上功能的所有 demo code。

使用方式：

需要将应用层 `tim_demo.c` `tim_demo.h` 驱动层 `tim.c` `tim.h`、`gpio.c` `gpio.h`、`isr.c` `isr.h` 加入到工程中去；

3.1.应用例程使用

BAT32G137 芯片引脚图



1、根据芯片引脚图：Timer40 的 4 个通道 作为输出时候：T000~ T003 分别对应引脚 P01 P16 P17 P31；作为输入时候：TI00~TI03 分别对应引脚 P00, P16 P17 P31；

由上面分析：在使用 Timer40 时候需要分配固定的引脚，Timer40 定时通道不能任意映射。

3.1.1. 方波输出

```
1. void Tim40_SquareOutput()
2. {
3.     TIM_InitTypeDef TIM_InitStructure;
4.     GPIO_InitTypeDef GPIO_InitStructure;
5.
6.     GPIO_PinAFConfig(GPIO_PORT0,GPIO_Pin_1,GPIO_P01,GROUP_AF_ODEFAULT);
7.     //P01 used as T000 output(P01 default function)
8.
9.     GPIO_InitStructure.GPIO_Pin = GPIO_Pin_1;
10.    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
11.    GPIO_InitStructure.GPIO_Level = GPIO_Level_LOW;
12.    GPIO_InitStructure.GPIO_Ctrl = GPIO_Control_DIG;
13.    GPIO_Init(GPIO_PORT0,&GPIO_InitStructure);
```

```

13.
14.
15. TIM_InitStructure.TIM           = TIM40;
16. TIM_InitStructure.TIM_Channel   = TIM_Channel_0;
17. TIM_InitStructure.TIM_ClkDivision = TIM_CLK0_Div1;    //specify the
    operation clk of tim
18. TIM_InitStructure.TIM_Period[0]  = 2000;              //specify
    the number of count clock
19. TIM_InitStructure.TIM_Trigger    = TIM_Trigger_Software; //specify
    the software trigger
20. TIM_InitStructure.TIM_Mode       = TIM_Mode_Square;    // square
    count mode
21. TIM_InitStructure.TIM_StartInt   = TIM_StartInt_Disable;
22. TIM_Init(&TIM_InitStructure);
23. }

```

- 使用 Timer40 通道 0 输出方波，对应芯片引脚 P01
- 配置 GPIO 为输出、数字功能、初始电平为低电平
- 选择 TIM40 定时器（BAT135 有 2 个 Timer4:Timer40, Timer41）
- 设置通道运行时钟：系统时钟/分频系数
- 选择方波模式
- 配置计数装载值 2000
- 开始计数时是否产生定时器中断

注意：在配置计数装载值时候，必须一一对应：即通道 0 需要和 Period[0] 对应、通道 1 和 Period[1] 对应……

3.1.2. 外部事件计数

例程介绍：P16 或者 P17 用杜邦线连接到 P00，使用 Timer40 通道 0，对外部事件进行计数；每捕捉到 10 次上升沿，便会产生中断（计数值增加 1）；计数值通过串口输出

```

1. void Tim40_EventCount()
2. {
3.     TIM_InitTypeDef TIM_InitStructure;
4.     GPIO_InitTypeDef GPIO_InitStructure= {0};
5.
6.     GPIO_PinAFConfig(GPIO_PORT0,GPIO_Pin_0,GPIO_P00,GROUP_AF_ODEFAULT);
    //P00 used as TI00 input
7.

```

```

8.  GPIO_InitStruct.GPIO_Pin = GPIO_Pin_0;
9.  GPIO_InitStruct.GPIO_Mode = GPIO_Mode_IN;
10. GPIO_InitStruct.GPIO_Ctrl = GPIO_Control_DIG;
11. GPIO_Init(GPIO_PORT0,&GPIO_InitStruct);
12.
13. TIM_InitStructure.TIM = TIM40;
14. TIM_InitStructure.TIM_Channel = TIM_Channel_0;
15. TIM_InitStructure.TIM_ClkDivision = TIM_CLK0_Div1;    //specify the
    operation clk of tim
16. TIM_InitStructure.TIM_Period[0]    = 10;                //specify t
    he count num of event
17. TIM_InitStructure.TIM_Trigger      = TIM_Trigger_Software; //specify
    the software trigger
18. TIM_InitStructure.TIM_Pulse_Edge  = TIM_Pulse_Rising;    // TIM effec
    tive edge choose
19. TIM_InitStructure.TIM_Mode        = TIM_Mode_EventCount;    // ev
    ent count
20. TIM_InitStructure.TIM_StartInt    = TIM_StartInt_Disable;
21. TIM_InitStructure.TIM4_Input      = TIM4_CH0_Input_TI00;    // specif
    y the input of TI
22. TIM_Init(&TIM_InitStructure);
23.
24. ISR_Register(TM00_IRQn,tim40_channel0_interrupt);    //TIM40 通道
    0 中断服务路径注册
}

```

- 选择 TIM40 定时器（BAT135 有 2 个 Timer4:Timer40, Timer41）
- 选择事件计数模式
- 装载值 10
- 设置上升沿计数
- 选择外部输入引脚
- 开始计数时是否产生定时器中断

3.1.3. 输入脉冲间隔测量

```

1.  uint32_t Tim40_GetPulsePeriod()
2.  {
3.  uint32_t width;
4.
5.  TIM_InitTypeDef TIM_InitStructure={0};

```

```

6.  GPIO_InitTypeDef GPIO_InitStructure;
7.
8.  GPIO_PinAFConfig(GPIO_PORT0,GPIO_Pin_0,GPIO_P00,GROUP_AF_ODEFAULT);
    //P00 used as TI00 input
9.
10. GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0;
11. GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN;
12. GPIO_InitStructure.GPIO_Ctrl = GPIO_Control_DIG;
13. GPIO_Init(GPIO_PORT0,&GPIO_InitStructure);
14.
15. TIM_InitStructure.TIM = TIM40;
16. TIM_InitStructure.TIM_Channel = TIM_Channel_0;
17. TIM_InitStructure.TIM_ClkDivision = TIM_CLK0_Div1; //specify the
    operation clk of tim
18. TIM_InitStructure.TIM_Trigger = TIM_Trigger_IputEdge; //IputEdge
    e is used to start trigger and capture trigger
19. TIM_InitStructure.TIM_Pulse_Edge = TIM_Pulse_Rising; // Measure t
    he high or low level pulse width of P00/TI00 /TIM_Pulse_Both/TIM_Pu
    lse_Rising
20.
21. TIM_InitStructure.TIM_Mode = TIM_Mode_PluseInterval; // pul
    se interval measure: use capture function
22. TIM_InitStructure.TIM_StartInt = TIM_StartInt_Disable;
23. TIM_InitStructure.TIM4_Input = TIM4_CH0_Input_TI00; // specif
    y the input of TI
24. TIM_Init(&TIM_InitStructure);
25.
26. ISR_Register(TM00_IRQn,tim40_channel0_interrupt);
27. INTC_EnableIRQ(TM00_IRQn);
28.
29. g_Int_Taken = 0;
30. while(g_Int_Taken < 2);
31. width = pData.len;
32.
33. TIM_Channel_Stop(&TIM_InitStructure);
34. return (width);
35. }

```

- 选择脉冲间隔模式也就是捕捉模式
- 选择设置输入信号触发：使用输入有效边沿来进行开始触发和捕捉触发
- 设置输入信号有效边沿：上升沿有效

注意：当选择有效边沿为双边沿，输入信号触发为：输入信号双边沿来进行开始触发和捕捉触发，这种为测量脉冲宽度；

3.1.4. 间隔定时器

间隔定时，作为定时器最基本功能，可以精确的定时中断。每一个定时通道都可以用作一个定时器；即 Timer40/ Timer40 的 ch0~ch3 的每一个通道都可以用作定时器。

```

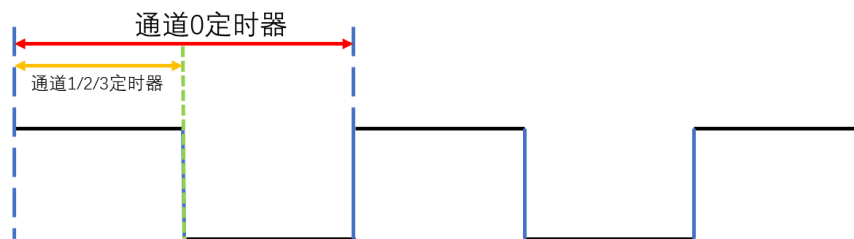
1.  /*****
2.  * Function Name: Tim41_IntervalTimer
3.  * @brief This function interval timer and generate interval interrup
   t.
4.  * @param
5.  * @return none
6.  *****/
7.  void Tim40_IntervalTimer()
8.  {
9.      TIM_InitTypeDef TIM_InitStructure;
10.
11.     TIM_InitStructure.TIM           = TIM40;           //specify the 8 channel
   s TIM
12.     TIM_InitStructure.TIM_Channel = TIM_Channel_0;
13.     TIM_InitStructure.TIM_ClkDivision = TIM_CLK0_Div32768; // specify th
   e operation clk=1khz of tim for 32Mhz fclk
14.     TIM_InitStructure.TIM_Period[0]  = 20;           //specify the time =20
   *(1/1000) =20ms
15.     TIM_InitStructure.TIM_Trigger    = TIM_Trigger_Software; //specify
   the software trigger
16.     TIM_InitStructure.TIM_Mode       = TIM_Mode_Interval; // external
   event
17.     TIM_InitStructure.TIM_StartInt   = TIM_StartInt_Disable;
18.     TIM_Init(&TIM_InitStructure);
19.
20.     ISR_Register(TM00_IRQn,tim40_channel0_interrupt); //TIM41 通道0 中断
   服务注册
}

```

- 选择定时器 tim40
- 选择通道为 TIM_Channel_0
- 设置运行时钟分频假如使用 32MHz 的系统主频分频系数为 32768，定时器运行频率为 1MHz
- 设置 TIM_Channel_0 计数周期 20 个 CLK 即定时 $20 \times 1/1000 = 20\text{ms}$

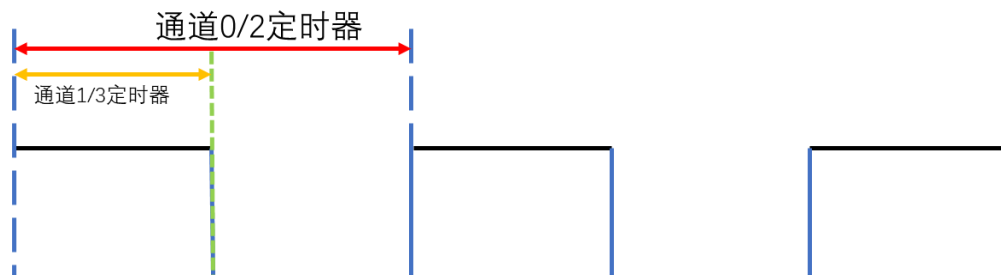
3.1.5. 多通道主从联动输出可调 PWM

对于定时单元 TIMER40，定时单元都有 4 个通道 ch0，ch1，ch2，ch3；其中当 ch0 通道的 16 位定时器用作周期定时器，其余通道定时器用作占空比定时器，如下图



这种情况下，Timer40 可以最多产生 3 路同周期，不同占空比的 PWM；

若 ch0 和 ch2 用作周期定时器，ch1 用作 ch0 的占空比定时器；ch3 用作 ch2 的占空比定时器，如下图：



这种情况下，Timer40 可以产生 2 路不同周期的 PWM；

```
1. void Tim40_PwmOutput1_1(uint16_t period, uint16_t duty)
2. {
3.     TIM_InitTypeDef TIM_InitStructure={0};
4.     GPIO_InitTypeDef GPIO_InitStructure= {0};
5.
6.     GPIO_PinAFConfig(GPIO_PORT1,GPIO_Pin_6,GPIO_P16,GROUP_AF_ODEFAULT);
       //P16 default used as T001 output
7.
8.     GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6;
9.     GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
10.    GPIO_InitStructure.GPIO_Level = GPIO_Level_LOW;
11.    GPIO_InitStructure.GPIO_Ctrl = GPIO_Control_DIG;
12.    GPIO_Init(GPIO_PORT1,&GPIO_InitStructure);
```

```

13.
14. TIM_InitStructure.TIM = TIM40;
15. TIM_InitStructure.TIM_Selection_Master = TTM_Channel_0;//when multi
-tim combination, it can generate pwm wave
16. TIM_InitStructure.TIM_Channel      = TTM_Channel_1; ///TTM_Channel_
1
17. TIM_InitStructure.TIM_ClkDivision = TIM_CLK0_Div1; //specify the
operation clk of tim
18. TIM_InitStructure.TIM_Period[0] = period; //specify the
number of count clock
19. TIM_InitStructure.TIM_Period[1] = duty; //specify duty
20.
21. TIM_InitStructure.TIM_Trigger      = TIM_Trigger_Software; //specify t
he software trigger
22. TIM_InitStructure.TIM_Mode        = TIM_Mode_PWM_Master; // PWM_Mas
ter mode
23. TIM_InitStructure.TIM_StartInt    = TIM_StartInt_Disable; //the relat
ionship between startCount and interrupt setting
24. TIM_Init(&TIM_InitStructure);
25. }

```

- 选择定时器 tim40
- 注意设置主控通道为 TTM_Channel_0
- 从属通道选择 TTM_Channel_1
- 分别设置周期定时器 Period[0]，占空比定时器 Period[1]
- 选择 PWM 模式

3.1.6. 多通道主从联动输出单脉冲

单脉冲模式，将主控通道用作延时定时器，从属通道用作输出脉宽；即在计时器运行时，当计数到主控通道预设的定时器值时，根据主控通道中断触发从属通道定时器计时，从属通道运行完，进行复位；手册说明如下：

(1) 单触发脉冲输出

将2个通道成对使用，生成能任意设定输出时序和脉宽的单触发脉冲。



```

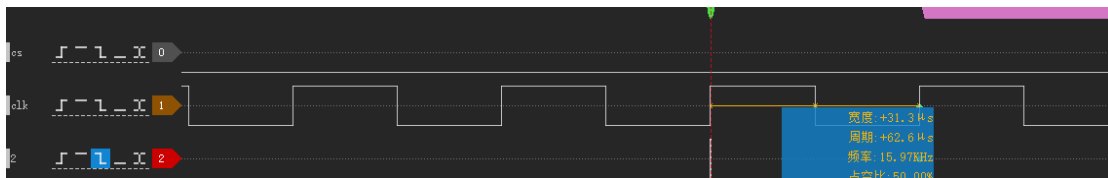
1. void Tim40_Single_PWM(uint16_t delay, uint16_t period)
2. {
3.     TIM_InitTypeDef TIM_InitStructure={0};
4.     GPIO_InitTypeDef GPIO_InitStructure= {0};
5.
6.     GPIO_PinAFConfig(GPIO_PORT1,GPIO_Pin_6,GPIO_P16,GROUP_AF_ODEFAULT);
7.     //P16 default used as T001 output
8.
9.     GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6;
10.    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
11.    GPIO_InitStructure.GPIO_Level = GPIO_Level_LOW;
12.    GPIO_InitStructure.GPIO_Ctrl = GPIO_Control_DIG;
13.    GPIO_Init(GPIO_PORT1,&GPIO_InitStructure);
14.
15.    TIM_InitStructure.TIM = TIM40;
16.    TIM_InitStructure.TIM_Selection_Master = TTM_Channel_0;//when multi
17.    -tim combination,it can generate pwm wave
18.    TIM_InitStructure.TIM_Channel      = TTM_Channel_1;  //|TTM_Channel_
19.    2|TTM_Channel_3
20.
21.    TIM_InitStructure.TIM_ClkDivision = TIM_CLK0_Div32768;  //specify
22.    the operation clk of tim
23.    TIM_InitStructure.TIM_Period[0]    = delay;             //specif
24.    y the number of count clock
25.    TIM_InitStructure.TIM_Period[1]    = period;           //speci
26.    fy duty
27.
28.    TIM_InitStructure.TIM_Trigger      = TIM_Trigger_IputEdge; //specify
29.    the software trigger
30.
31.    TIM_InitStructure.TIM_Mode        = TIM_Mode_SinglePulse;  // PWM_
32.    Master mode
33.
34.    TIM_InitStructure.TIM_StartInt     = TIM_StartInt_Disable; //the rel
35.    ationship between startCount and interrupt setting
36.
37.    TIM_Init(&TIM_InitStructure);
38.
39.}

```

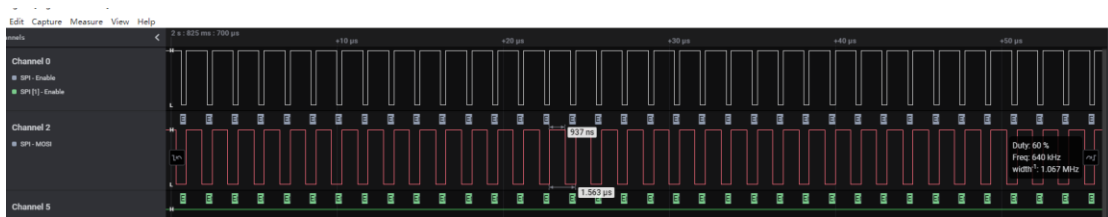
```
26. ISR_Register(TM00_IRQn,tim40_channel0_interrupt);
27. }
```

- 选择定时器 tim40
- 选择 T01，通道 1 作为从属通道输出单脉冲
- 分别设置主控通道周期定时器 Period[0]，占空比定时器 Period[1]
- 选择单脉冲模式
- 注册中断函数，开启中断函数

4. 示例演示



图一 方波时序图



图二 PWM（占空比 80 60）



图三 单脉冲

从复位开始，延时 10000clock，从机计数 10000 个 clk