

AT32 PWM Input Test

前言

这篇应用笔记介绍了如何使用 AT32 定时器进行 PWM 输入测试。

注：本应用笔记对应的代码是基于雅特力提供的V2.x.x 板级支持包（BSP）而开发，对于其他版本BSP，需要注意使用上的区别。

支持型号列表：

支持型号	AT32F 系列 AT32L 系列
------	----------------------

目录

1	AT32 定时器说明	5
1.1	定时器框架介绍	5
1.2	定时器一般配置步骤	5
2	规格说明	7
2.1	AT32 PWM 高频测试说明	7
2.2	AT32 PWM 低频测试说明	8
2.3	AT32 PWM 占空比测试原理说明	9
3	PWM Test 快速使用方法	10
3.1	硬件资源	10
3.2	pwm input test demo 使用	10
4	文档版本历史	13

表目录

表 1. 文档版本历史 13

图目录

图 1. 通用定时器的框图.....	5
图 2. 高频测试原理框图.....	7
图 3. 低频测试原理框图.....	8
图 4. 占空比测试原理框图.....	9
图 5. AT-START-F403A 实验板.....	10
图 6. 测试高频信号，串口打印信息.....	11
图 7. 测试低频信号，串口打印信息.....	11
图 8. 测试 PWM 占空比，串口打印信息.....	12

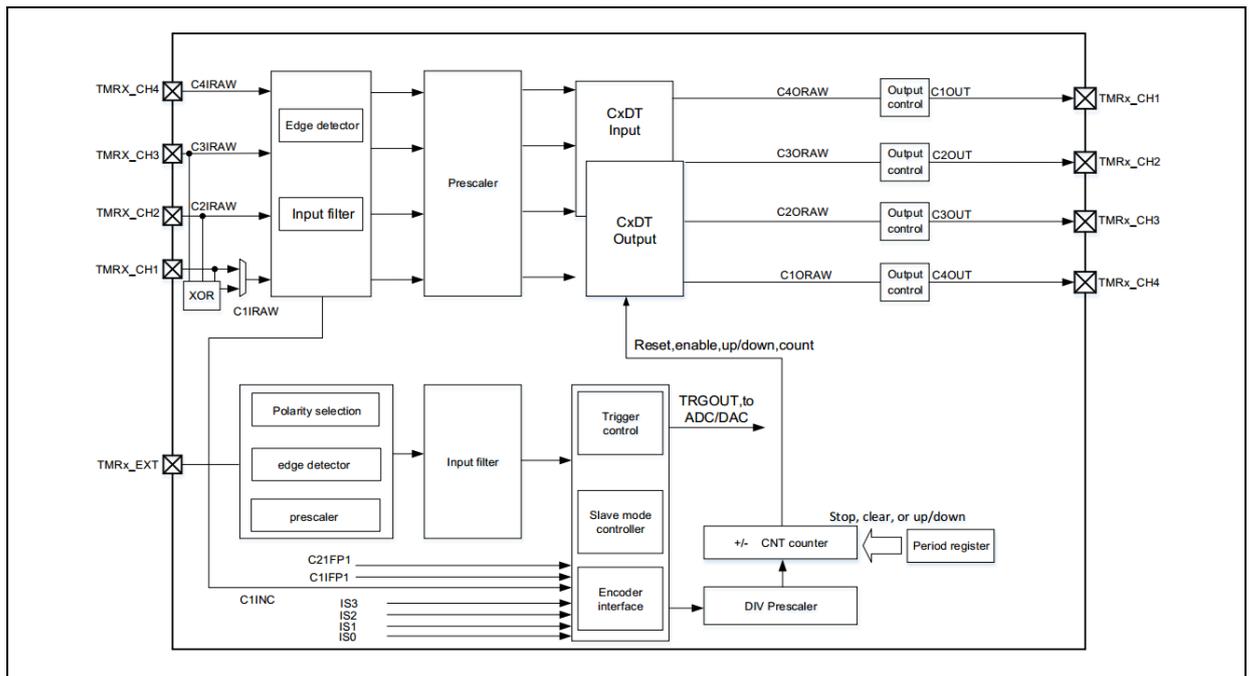
1 AT32 定时器说明

1.1 定时器框架介绍

定时器由一个 16 位的自动装载计数器组成（TMR2 和 TMR5 较为特殊，其自动装载计数器为 32 位），它由一个可编程的预分频器驱动。它适合多种用途，包含测量输入信号的脉冲宽度（输入捕获），或者产生输出波形（输出比较、PWM、嵌入死区时间的互补 PWM 等）。

使用定时器预分频器和 CRM 时钟控制预分频器，可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。

图 1. 通用定时器的框图



定时器由四个主要部分组成（见图 1）。第一部分时钟单元。此单元提供定时器的时钟驱动。第二部分时钟基单元，此单元提供定时器计数功能。第三部分输入捕获，此单元允许输入信号进入定时器模块。第四部分输出比较，此单元将定时器整合后的 PWM 输出。

1.2 定时器一般配置步骤

- 1) 时钟使能。

```
crm_periph_clock_enable(CRM_TMR2_PERIPH_CLOCK, TRUE);
```

- 2) 初始化定时器参数，设置自动重装值，分频系数，计数方式等。

在库函数中，定时器的初始化参数是通过初始化函数 `tmr_base_init()` 及 `tmr_cnt_dir_set()` 实现的：

```
void tmr_base_init(tmr_type* tmr_x, uint32_t tmr_pr, uint32_t tmr_div);
```

其中，第一个参数是确定是哪个定时器，这个比较容易理解。第二个参数 `tmr_pr` 是定时器计数的周期值。第二个参数 `tmr_div` 是定时器的分频系数。

```
void tmr_cnt_dir_set(tmr_type *tmr_x, tmr_count_mode_type tmr_cnt_dir);
```

其中，第一个参数是确定是哪个定时器，第二个参数 `tmr_cnt_dir` 为定时器的计数模式（向上，

向下，中央对齐）。

特别地，增强模式是 TMR2 和 TMR5 独有的功能。tmr_32_bit_function_enable()为增强模式使能（Plus Mode Enable）函数。开启 TMRx 增强模式，该模式下 TMRx_CVAL，TMRx_PR，TMRx_CxDT 由 16 位扩展为 32 位。

```
void tmr_32_bit_function_enable(tmr_type *tmr_x, confirm_state new_state);
```

当需要配置时钟除频参数时(注意和 TMR_DIV 的区别，配置滤波、死区时间时需配置该参数)，会使用到 tmr_clock_source_div_set()函数；当需要配置重复周期寄存器时（高级定时器 TMR1/TMR8/TMR15 才有），会使用到 tmr_repetition_counter_set()函数。本例程不使用这两个函数，仅做简要介绍。

```
void tmr_clock_source_div_set(tmr_type *tmr_x, tmr_clock_division_type tmr_clock_div);
```

```
void tmr_repetition_counter_set(tmr_type *tmr_x, uint8_t tmr_rpr_value);
```

- 3) 设置 TMRx_IDEN 允许更新中断。

```
void tmr_interrupt_enable(tmr_type *tmr_x, uint32_t tmr_interrupt, confirm_state new_state);
```

这里着重描述参数 tmr_interrupt，它是用来指明我们使能的定时器中断的类型，定时器中断的类型有很多种，包括更新中断，触发中断，以及输入捕获中断等等。

- 4) TMRx 中断优先级设置。

调用 nvic_irq_enable()函数即可。

- 5) 允许 TMRx 工作，也就是使能 TMRx。

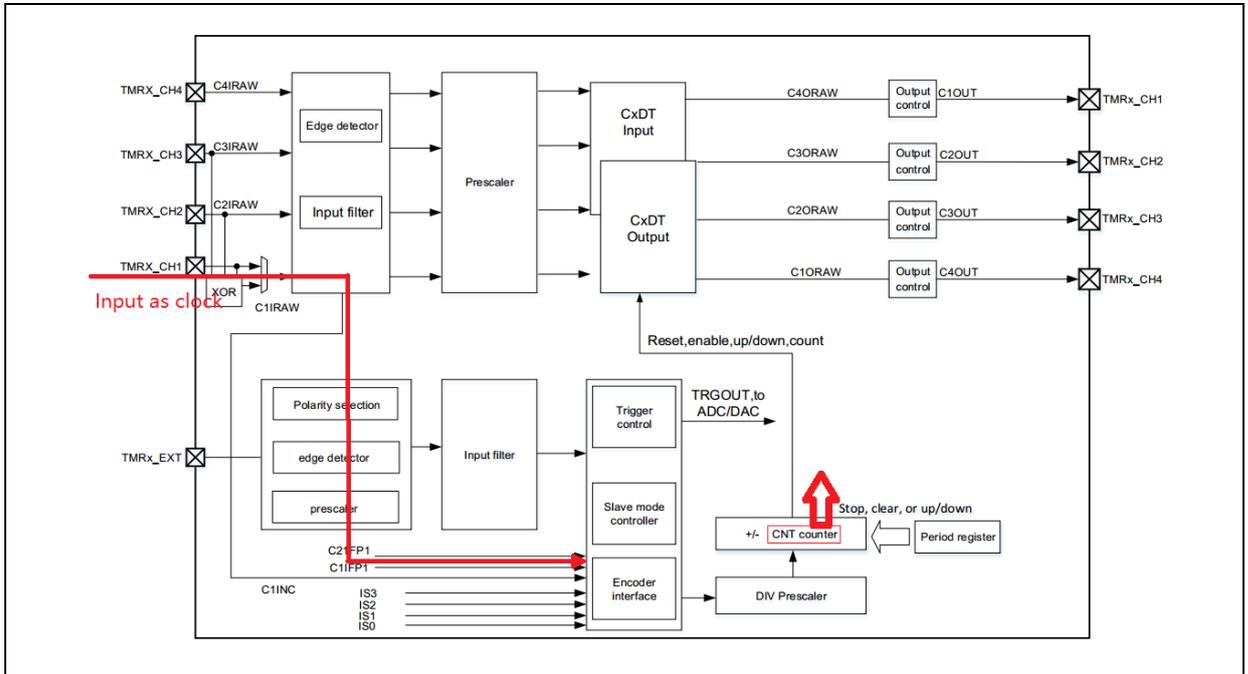
```
void tmr_counter_enable(tmr_type *tmr_x, confirm_state new_state);
```

- 6) 编写中断服务函数。

2 规格说明

2.1 AT32 PWM 高频测试说明

图 2. 高频测试原理框图



测试高频信号时，将高频信号输入作为定时器 TMR2 的时钟源（如上图所示），驱动定时器 TMR2 的 Counter 计数，使用另一个定时器做时钟基准，例如每隔 1s，获取 TMR2 的 Counter 变化值，则 TMR2 的变化值即为高频信号的频率值。

使用两个定时器，其中一个定时器为 TMR2（这里选取 TMR2 的原因在于其可以通过设置 TMRx_CTRL1 中的 PMEN 位，开启 TMRx 增强模式，该模式下 TMRx_CVAL，TMRx_PR，TMRx_CxDT 由 16 位扩展为 32 位），测试高频时，有利于 Counter 计数，不容易产生溢出。

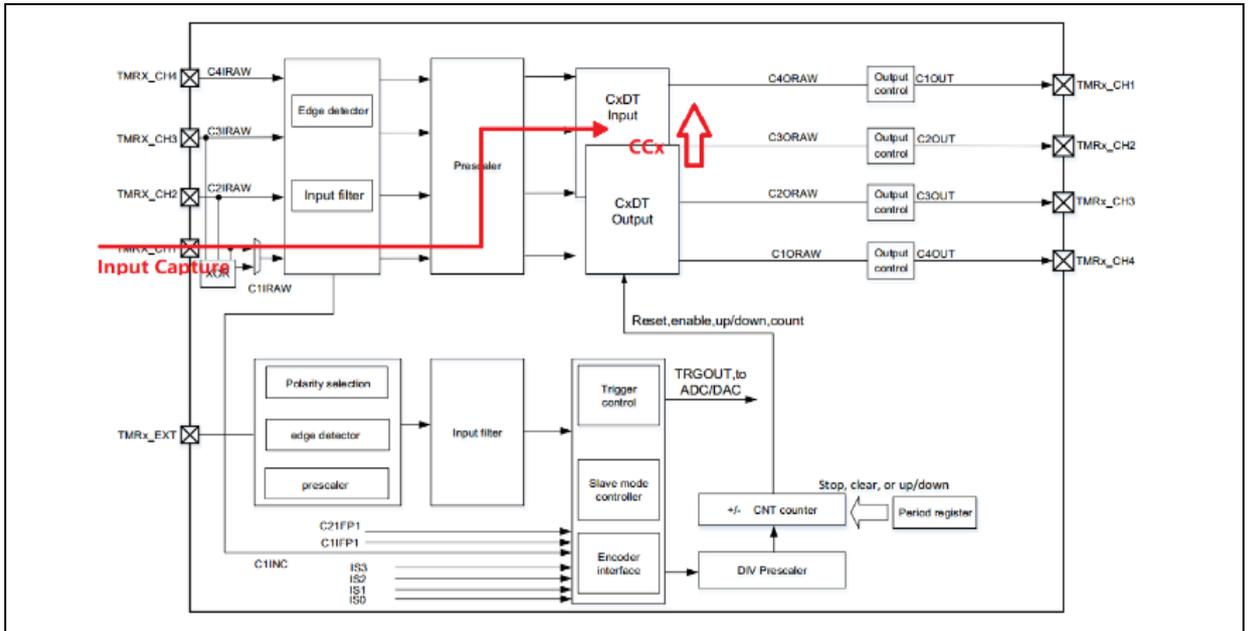
这样做的好处在于，既可以测试高频信号：最高 50MHz（受限于 I/O 口的最高频率），且没有频繁产生中断，代码也有冗余，去处理客户任务。

使用此方法测试的频率范围在：50MHz 到 1Hz（TMR2 的工作频率为 240MHz）。

注意：增强模式是TMR2和TMR5独有的功能，使用不支持增强模式的其他TMR，或使用不含有增强模式TMR的AT32时，测试频率会受限。

2.2 AT32 PWM 低频测试说明

图 3. 低频测试原理框图



测试低频信号时，将低频信号输入作为定时器TMR2的捕获输入（如上图所示），触发TMR2的输入捕获中断，利用TMR2工作时钟除以两次输入捕获之间Counter变化值，即可得到低频信号频率值。这里定时器为TMR2（这里选取TMR2的原因在于其可以通过设置TMRx_CTRL1中的PMEN位，开启TMRx增强模式，该模式下TMRx_CVAL，TMRx_PR，TMRx_CxDT由16位扩展为32位），有利于低频测试。

使用此方法测试的最低频率为：56mHz。（TMR2的工作频率在240MHz）。

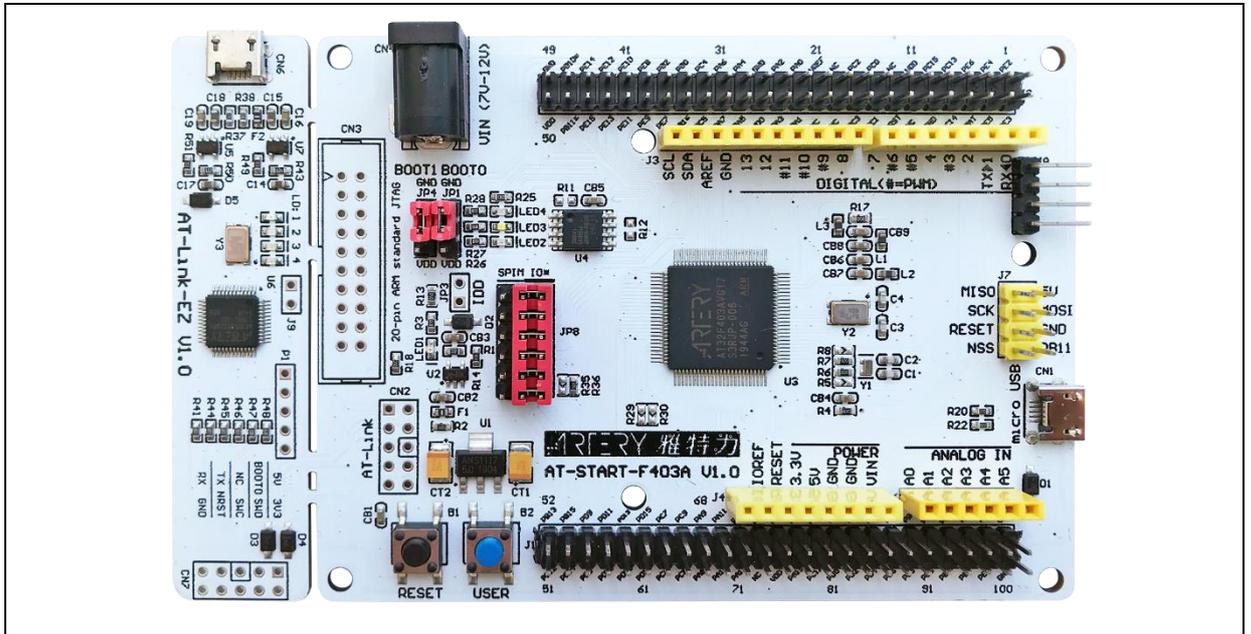
注意：增强模式是TMR2和TMR5独有的功能，使用不支持增强模式的其他TMR，或使用不含有增强模式TMR的AT32时，测试频率会受限。

3 PWM Test 快速使用方法

3.1 硬件资源

- 1) AT-START-F403A 实验板

图 5. AT-START-F403A 实验板



注意：该 Demo 是基于 AT32F403A 的硬件条件，若使用者需要在 AT32 其他型号上使用，请修改相应配置即可。

3.2 pwm input test demo 使用

- 打开pwm input test project源程序，其中，在at32f403a_407_clock.h里面有三个宏定义：

```
#define high_frequency_test
#define low_frequency_test
#define duty_ration_test
```

分别用于测试高频信号，低频信号，PWM占空比，打开需要测试的宏【注意：每次只开一个宏】。

打开pwm output源程序，其中，在at32f403a_407_clock.h中有三个宏定义：

```
#define Output_High_Frequency
#define Output_Low_Frequency
#define Output_PWM_Duty_Ration_10
```

分别产生高频信号，低频信号，PWM占空比用于测试。

AT-START板载的AT-LINK-EZ自带串口输出功能，它可以将USART1_TX口PA9输出至PC。也可使用其他串口工具进行测试结果的输出。

- 若测试高频信号时：

- 1) 打开pwm output源程序宏定义：`#define Output_High_Frequency`，PA8产生60MHz PWM（I/O口已超频工作，可适当降低主频）。编译下载到实验板1。

- 2) 打开pwm input test程序宏定义：`#define high_frequency_test`，编译下载到实验板2。
- 3) 将实验板1的PA8接入到实验板2的PA0，USART1通过PA9输出当前的PWM频率信息。
串口打印信息如下：

图 6. 测试高频信号，串口打印信息

```
Make sure connected with external signal
High Frequency : 60000002 Hz
High Frequency : 60000001 Hz
High Frequency : 60000001 Hz
High Frequency : 60000001 Hz
High Frequency : 60000002 Hz
High Frequency : 60000001 Hz
High Frequency : 60000001 Hz
High Frequency : 60000001 Hz
```

■ 若测试低频信号时：

- 1) 打开pwm output源程序宏定义：`#define Output_Low_Frequency`，PA8产生500mHz PWM。编译下载到实验板1。
- 2) 打开pwm input test程序宏定义：`#define low_frequency_test`，编译下载到实验板2。
- 3) 将实验板1的PA8接入到实验板2的PA0，USART1通过PA9输出当前的PWM频率信息。
串口打印信息如下（应将第一个数据舍弃）：

图 7. 测试低频信号，串口打印信息

```
Make sure connected with external signal
Low Frequency : 499.999969 mhz
```

■ 若测试PWM占空比时，

- 1) 打开pwm output源程序宏定义：`#define Output_PWM_Duty_Ration_10`，PA8产生6MHz PWM，占空比为10%。编译下载到实验板1。
- 2) 打开pwm input test程序宏定义：`#define duty_ration_test`，编译下载到实验板2。
- 3) 将实验板1的PA8接入到实验板2的PA0，USART1通过PA9输出当前的PWM占空比信息。
串口打印信息如下：

图 8. 测试 PWM 占空比, 串口打印信息

```
Make sure connected with external signal
Duty Ration : 10.791441 %
Duty Ration : 10.793387 %
Duty Ration : 10.791404 %
Duty Ration : 10.792070 %
Duty Ration : 10.792516 %
Duty Ration : 10.792645 %
Duty Ration : 10.791965 %
Duty Ration : 10.793122 %
```

4 文档版本历史

表 1. 文档版本历史

日期	版本	变更
2021.12.30	2.0.0	最初版本

重要通知 - 请仔细阅读

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和/或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途（及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力产品并非设计或专门用于下列用途的产品：（A）对安全性有特别要求的应用，例如：生命支持、主动植入设备或对产品功能安全有要求的系统；（B）航空应用；（C）航天应用或航天环境；（D）武器，且/或（E）其他可能导致人身伤害、死亡及财产损害的应用。如果采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，风险及法律责任仍将由采购商单独承担，且采购商应独立负责在前述应用中满足所有法律和法规要求。

经销的雅特力产品如有不同于本文档中提出的声明和/或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。

© 2021 雅特力科技 保留所有权利