RT8001 高频脉冲触发实现

Date: August 21, 2023

Project Number: AT-xx-xxxxxx

I Versions

Version	Date	Comment	Edited by
1.0	Dec 16, 2022	First Edition	张锋

Table 1: Versions

II Distribution

Name	Company, Department	Amount	Remarks

Table 2: Distribution

III Safety Notices

Safety notices in this document are organized as follows:

Safety notice	Description
Danger!	Disregarding the safety regulations and guidelines can be life-threatening.
Warning!	Disregarding the safety regulations and guidelines can result in severe injury or heavy damage to material.
Caution!	Disregarding the safety regulations and guidelines can result in injury or damage to material.
Information:	Important information used to prevent errors.

Table 3: Safety notices

IV Table of Contents

1 Introduction

1.1 需求

客户需要一个 4 路的 PWM 控制输出，触发 IGBT 脉冲形成 4 路单向脉冲电源，要求主要如下：

1. 占空比可调 0-30%，调节精度(分辨率)0.1%；
2. 频率可调 1Hz-1500Hz，调节精度(分辨率)1Hz；
3. 输出 4 路 PWM，4 路 PWM 的占空比、频率相同（即 4 路 PWM 只设定 1 个占空比和频率）；
4. 1#、4#通道输出相同，2#、3#通道相同（相同的含义为脉冲同时触发，同时消失）
5. 2#、3#通道触发比 1#、4#通道触发滞后半个周期，4#、3#在关闭时分别比 1#、2#延迟关闭数十微妙，500Hz 延迟 156us，1000Hz 延迟 78us，1500Hz 延迟 52us。

1、当电压幅值=1000V、频率=200HZ、占空比=50%时的波形：（如下图1）

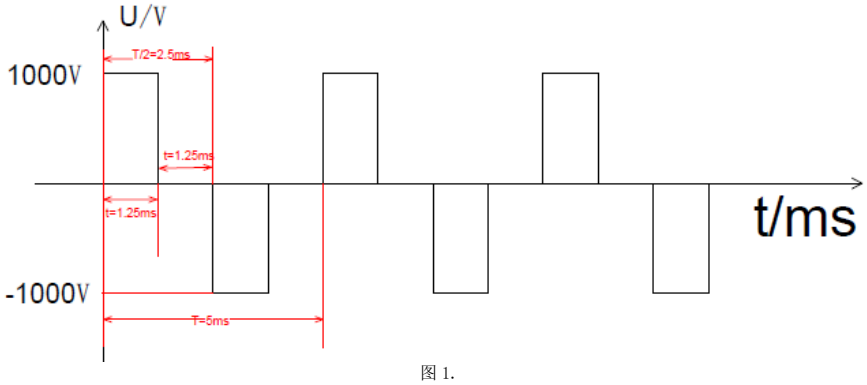


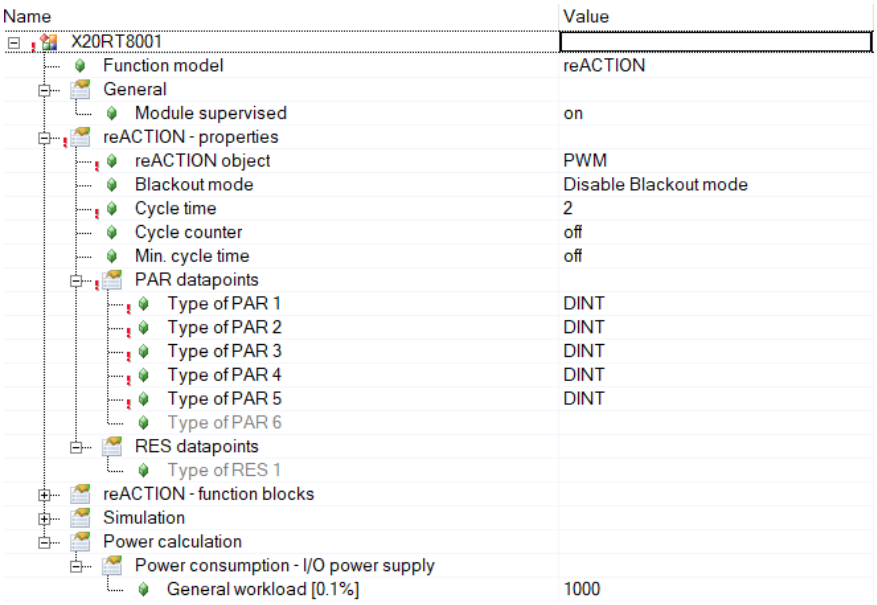
图 1.

RT8001 刚好有 4 路可以实现高速输入/输出，因此选用 RT8001 模块实现。

1.2 模块配置

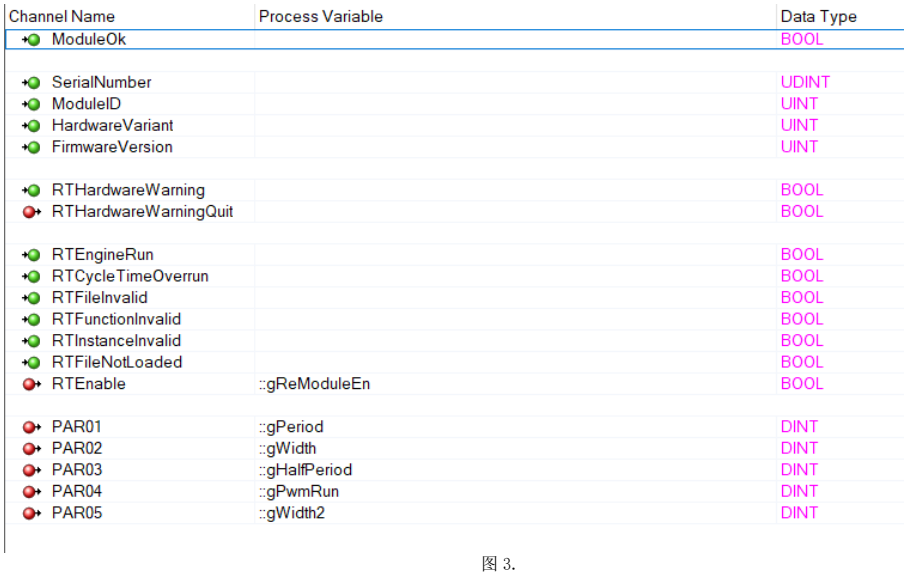
如下图 2 所示，RT8001 Cycle Time 配置为 2us，跟 CPU 通讯参数配置 5 路，主要为 CPU 发送给 RT8001 模块的数据，模块 IO Map 表中的数据分别为：

gPeriod：脉冲调制周期，由调制频率换算。
gWidth：脉宽，由占空比换算得出，为 1#、2#通道脉宽。
gWidth2：脉宽，gWidth+延迟关闭，为 4#、3#通道脉宽。
gHalfPeriod：脉冲周期/2，程序计算得出。
gPwmRun：开启 PWM 控制。



Name	Value
X20RT8001	
Function model	reACTION
General	
Module supervised	on
reACTION - properties	
reACTION object	PWM
Blackout mode	Disable Blackout mode
Cycle time	2
Cycle counter	off
Min. cycle time	off
PAR datapoints	
Type of PAR 1	DINT
Type of PAR 2	DINT
Type of PAR 3	DINT
Type of PAR 4	DINT
Type of PAR 5	DINT
Type of PAR 6	
RES datapoints	
Type of RES 1	
reACTION - function blocks	
Simulation	
Power calculation	
Power consumption - I/O power supply	
General workload [0.1%]	1000

图 2



Channel Name	Process Variable	Data Type
ModuleOk		BOOL
SerialNumber		UDINT
ModuleID		UINT
HardwareVariant		UINT
FirmwareVersion		UINT
RTHardwareWarning		BOOL
RTHardwareWarningQuit		BOOL
RTEngineRun		BOOL
RTCycleTimeOverrun		BOOL
RTFileInvalid		BOOL
RTFunctionInvalid		BOOL
RTInstanceInvalid		BOOL
RTFileNotLoaded		BOOL
RTEnable	::gReModuleEn	BOOL
PAR01	::gPeriod	DINT
PAR02	::gWidth	DINT
PAR03	::gHalfPeriod	DINT
PAR04	::gPwmRun	DINT
PAR05	::gWidth2	DINT

图 3.

1.3 软件实现

1.3.1 CPU 计算周期、脉宽实现

如下图 4、5、6 所示，程序中 gSetFrequency 为触摸屏设定的脉冲频率，gSetWidth 为设定的占空比。此段程序目的主要是根据设定的频率和占空比计算的出 Reaction 程序 PWM 功能块需要的脉冲周期和脉宽占比。

需要注意的是，程序中为了防止连续调节频率时误触发损坏硬件，频率每改变一次主动关闭 PWM 1s 时间，用以延迟防止误触发。

```
if(gHmiStart != 0)
{
    gPwmRun = 1;
}
else if(gHmiStop != 0)
{
    gPwmRun = 0;
}

// 计算PWM周期
if(gSetFrequency >= 1 && gSetFrequency <= 1500)
{
    gPeriod = 1000000000 / gSetFrequency;              // 单位：转换为1ns
    gPeriod         = gPeriod / 10;                    // 单位：转换为模块需要的单位10ns
    gHalfPeriod     = gPeriod / 400;                   // 单位：转换为模块需要的单位us
}
else
{
    gPeriod         = 0;
    gHalfPeriod     = 0;
    gWidth          = 0;
    gWidth2         = 0;
    gWidth_us       = 0;
    gWidth2_us      = 0;
    gPwmRun         = 0;
    return;
}
```

图 4.

```
// 计算PWM脉宽占比，系统规定8192为100%，本系统占空比指的是一个周期内正向脉冲和反向脉冲加起来之和
if(gSetFrequency >= 500 && gSetFrequency <= 1500)
    gDelayOff_us = 156 * 500 / gSetFrequency;
else
    gDelayOff_us = 0;

if(gSetWidth >= 0 && gSetWidth <= 30)
{
    gWidth_us   = gSetWidth / 100.0 * gPeriod / 100 / 2;
    gWidth2_us  = gWidth_us + gDelayOff_us;
    gWidth      = gSetWidth * 4096 / 100;
    if(gPeriod > 0)
        gWidth2     = gWidth2_us * 100.0 / gPeriod * 8192;
}
else
{
    gWidth      = 0;
    gWidth2     = 0;
    gWidth_us   = 0;
    gWidth2_us  = 0;
}
```

图 5.

```
// 连续设定频率，需要主动关闭一次PWM调制
if(OldSetFrequency != gSetFrequency)
{
    TempStopFlag = 1;
    TimeDelay = 0;
}
if(TempStopFlag != 0)
{
    if((TimeDelay +=Cyct) >= 1000)
    {
        TimeDelay       = 0;
        TempStopFlag    = 0;
    }
    gPwmRun = 0;
}

OldSetFrequency = gSetFrequency;
```

图 6.

1.3.2 Reaction PWM 实现

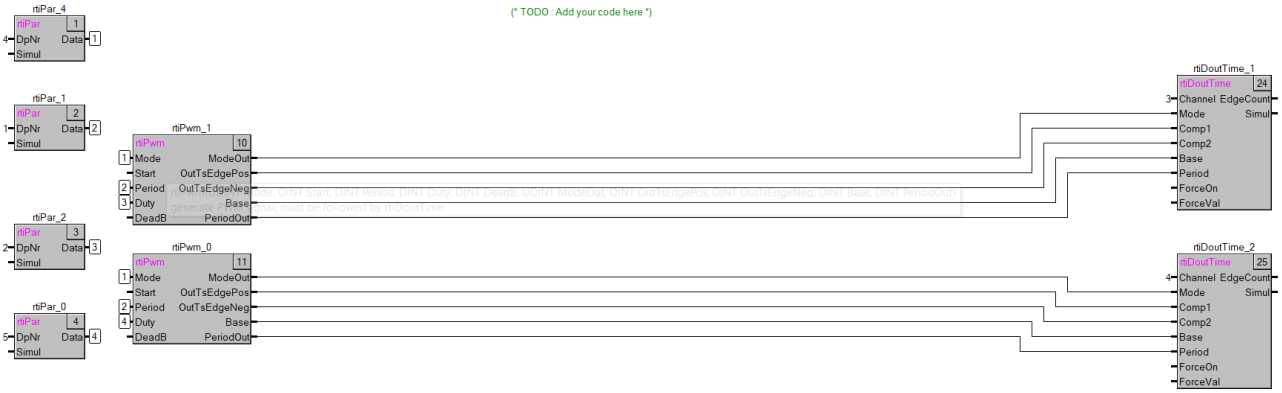


图 6. 1#和 4#通道输出

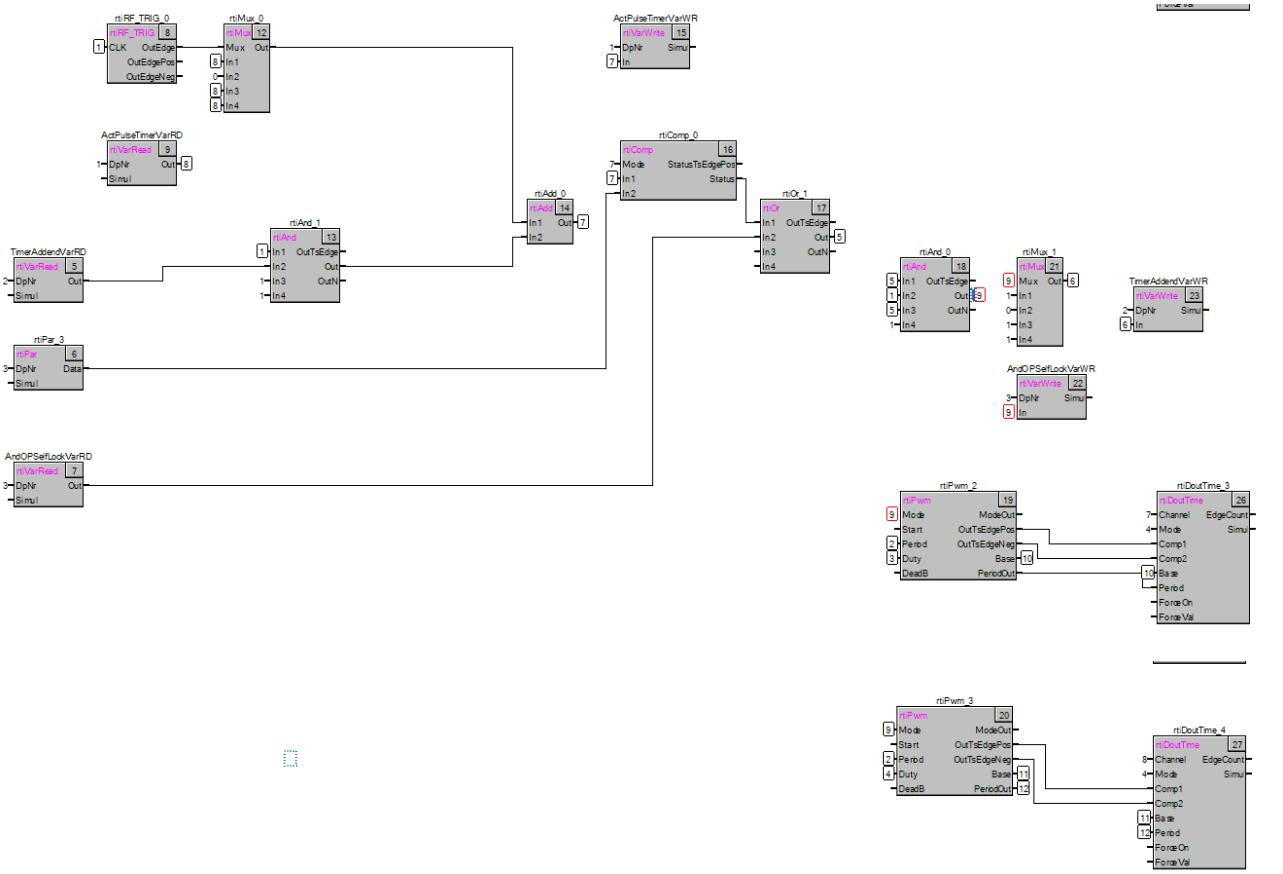


图 7 和图 8. 2#和 3#通道输出

实现 PWM 输出需要用到 AsIORTI 库当中的 rtiPwm()和 rtiDoutTime()功能块，这两个功能块必须配合使用，具体参数以及使用详见 help 中的说明，使用 rtiPwm 功能块的输入 Mode 来控制 PWM 的开启和关断。1#和 4#通道 PWM 输出比较简单，直接调用 rtiPwm()和 rtiDoutTime()功能块即可实现，这里不在叙述，重点是 2#和 3# PWM 输出控制。

由于 2#和 3#通道总是滞后 1#和 4#通道半个周期，我们需要精确控制 2#和 3# PWM 相对 1#、4#通道的开启时间，查看了 AsIORTI 库的所有功能块，发现没有可以用到的延迟功能块，因此需要用多个功能块逻辑互相配合来实现延迟计时。

gHalfPeriod 参数为设定脉冲周期一半，单位为 us，由于 PWM Reaction 程序执行周期设定为 2us，因此，在程序里将 gHalfPeriod 实际设定为脉冲周期的一半再除以 2,得到实际需要计时的周期次数。

程序设计思路：当开启 PWM 调制时，立刻启动 1#和 4#通道 PWM 调制，2#和 3#通道开始计时，每个周期进行加 1 操作，同时比较功能块（工作模式设定为“>=”）一直进行比较功能，比较阈值为脉冲周期一半（gHalfPeriod），当计时值大于等于 gHalfPeriod 时，比较功能块输出状态 1 触发逻辑操作开始 2#和 3#通道的 PWM 调制，同时为了防止计数值累加溢出，需要将加 1 操作进行暂停，同时将比较输出的状态 1 进行锁存。当关闭 PWM 调制时，立刻关闭 1#和 4#通道的 PWM 调制，同时也关闭 2#和 3#通道的 PWM 调制，需要将计数值复位为 0，安全期间，当开启时，也同样可以做一次复位计数值，为下一次改变频率或者关闭之后在开启做准备。

Figure Index

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

2 Table Index

3 Listing Index

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

4 Index