

摘要

本技术文档旨在帮助用户快速配置 MCU CS32F03X 输出多路经过调制的 PWM 波，用于控制直流无刷电机(BLDC)换相。HPWM-LON 是直流无刷电机较为常用的 PWM 调制模式，为了实现该功能，需要正确配置 MCU 的 TIM1 外设和 GPIO 电平。文档中提到了如何使用 TIM1 高级定时器输出经过调制的 PWM 波，以及如何处理直流无刷电机换相的问题。

版本

历史版本	修改内容	日期
V1.0	初版生成	2022-10-05

目录

1 硬件介绍和直流无刷电机原理.....	3
2 基于 SDK 测试.....	错误！未定义书签。
3 注意事项与总结.....	错误！未定义书签。

1 硬件介绍和直流无刷电机原理

1.1 硬件介绍：

电源： 可调直流电源（至少 3A）

调试器： J-Link

示波器： 双通道示波器

评估板： CS32F03x BLDC Driver

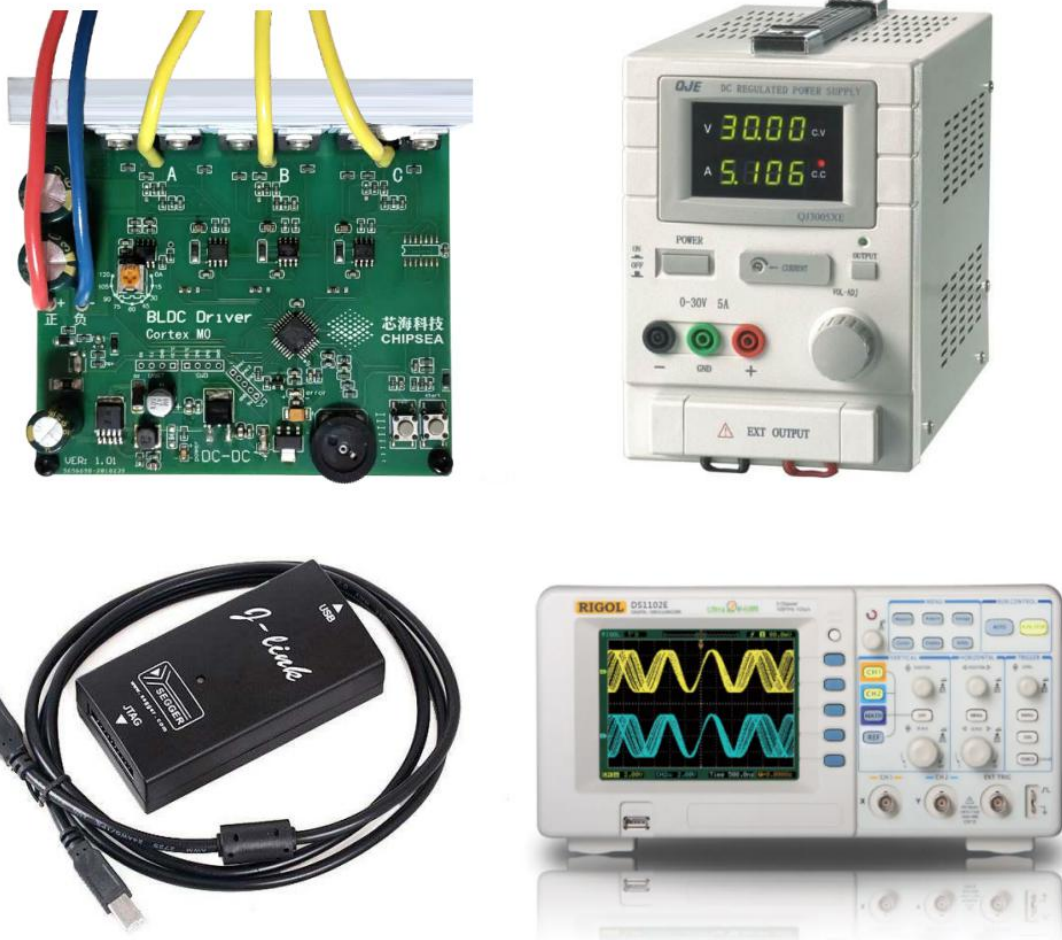


图 1 实验设备

1.2 直流无刷电机原理:

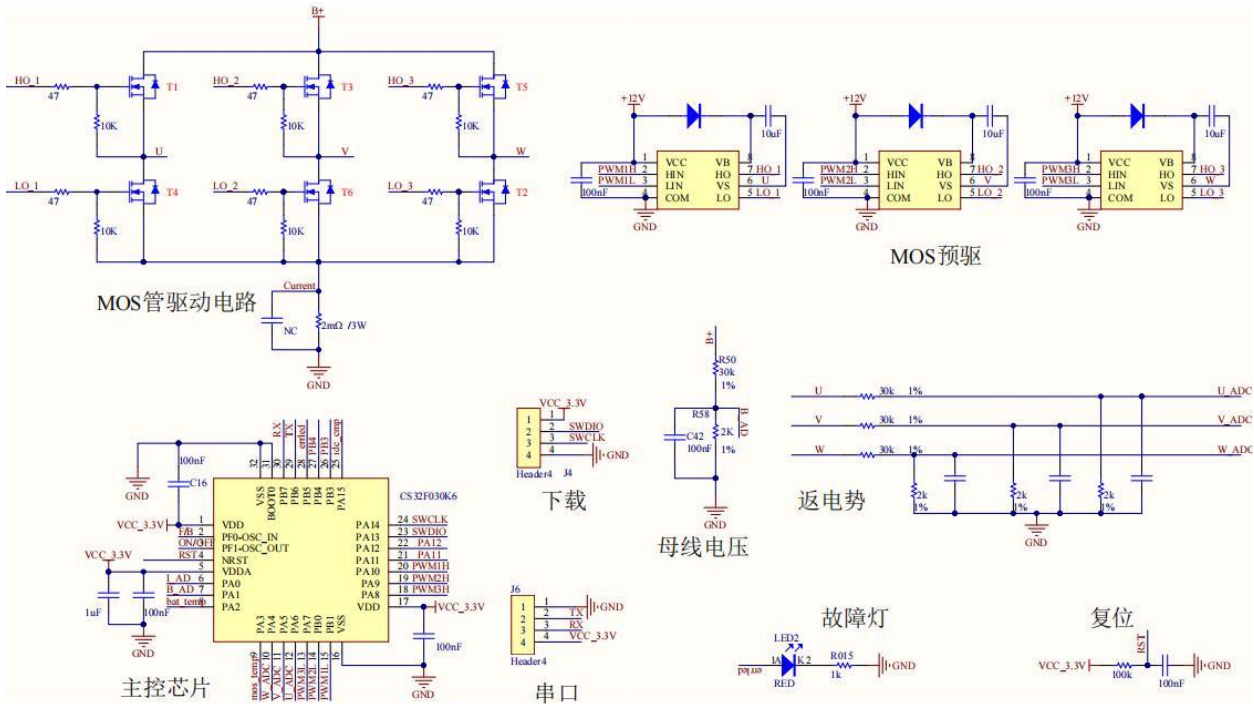


图 2 原理图

直流无刷电机由定子和转子组成，是一种没有电刷和换向器（或集电环）的电机，又称为电子换向式直流电机。该电机具备功率密度高、寿命长、效率高等优点，是一种正在快速普及的电机类型，广泛应用于电动工具、园林工具、家用电器、汽车、航空航天、医疗、工业自动化设备和仪器等领域。

无刷电机的结构示意图如图 3 所示。

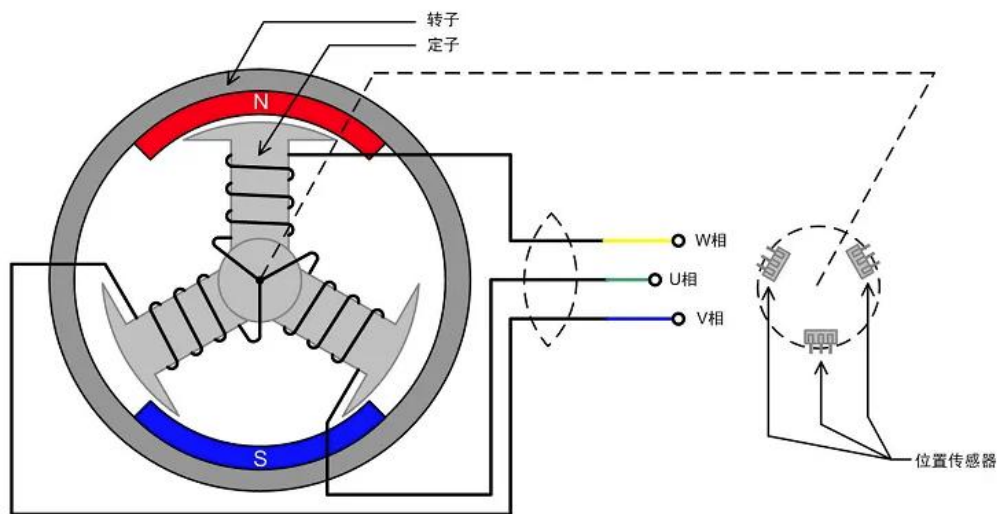


图 3 无刷电机结构示意图

2 基于 SDK 测试

本次实验使用无刷电机评估板驱动无刷电机换相，给评估板供电 16V，电机三相线接到评估板的三相线输出线缆 ABC。连接示意图如图 4 所示。

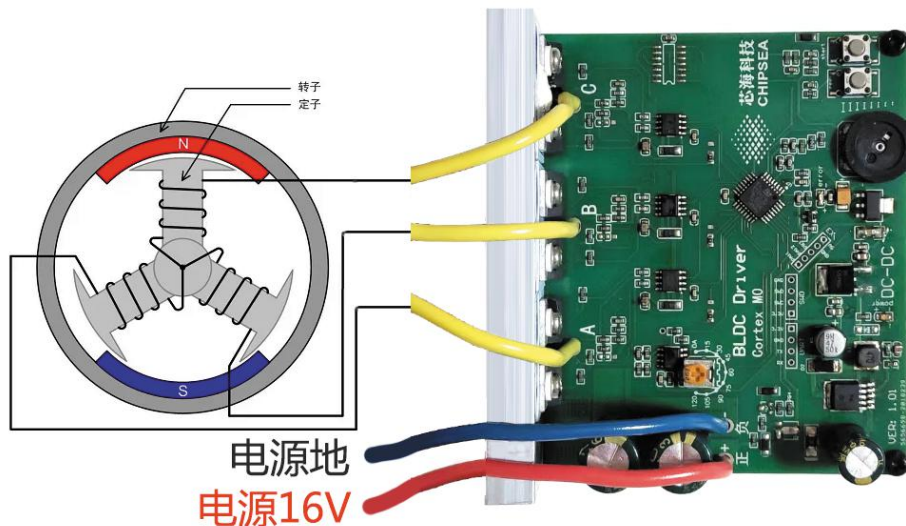


图 4 无刷电机与评估板连线示意图

MCU 的引脚输出能力有限，无法直接驱动无刷电机，此处使用 MCU 产生 PWM 波来驱动 MOS 预驱芯片，再驱动 MOS 管来控制无刷电机换相，本实验采用 HPWM-LON 的调制方式。在这种控制方式下，电机端电压波形为梯形波，因此也称为梯形波控制。PWM 输出存在 6 种状态，对于每种状态，逆变桥的 6 个功率管中仅有 2 个工作。常见的几种 PWM 调制波形如图 5 所示：

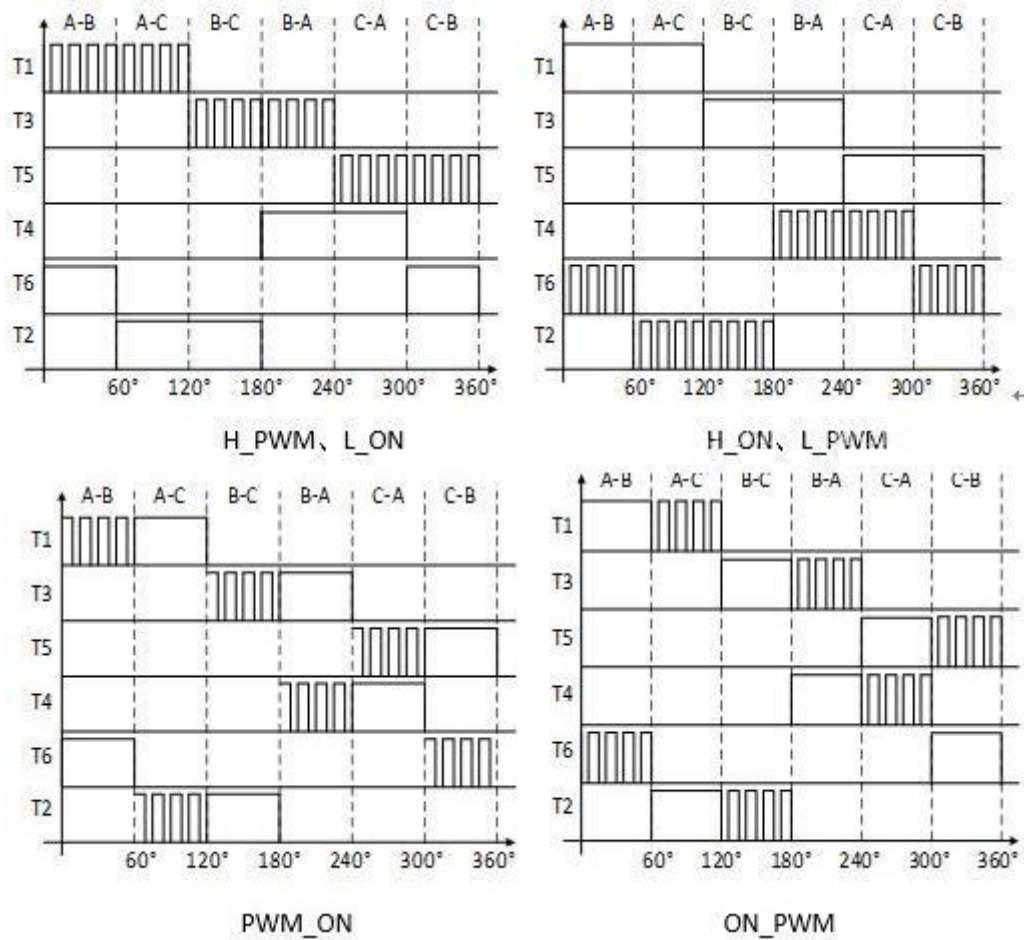


图 5 常见的几种 PWM 调制方式

当需要导通一个上桥臂和一个下桥臂时，如 HPWM-LON 调制方式的第一步 A-B，当 T1 上桥臂以及 T6 下桥臂导通时，其他桥臂的输入电压需为低电平，以使 MOS 管关断。即 MCU 输出的信号应为 PA10 引脚输出 PWM 波以及 PB0 引脚输出高电平，PA7、PB1、PA8 和 PA9 均输出低电平，以关闭另外四个 MOS 管。

TIM1 输出 PWM 原理:

高级定时器包括一个 16 位的预分频器和 16 位的计数器，用来对时钟或外部事件进行计数。高级定时器包括四个捕获比较通道。作为捕获通道时，用来捕获外部触发输入有效沿时刻的计数器值；作为比较通道时，产生输出波形，包括输出比较和 PWM，也可以输出带死区时间控制的互补输出 PWM，高级定时器框图如图 6 所示，PWM 输出原理框图如图 7 所示。

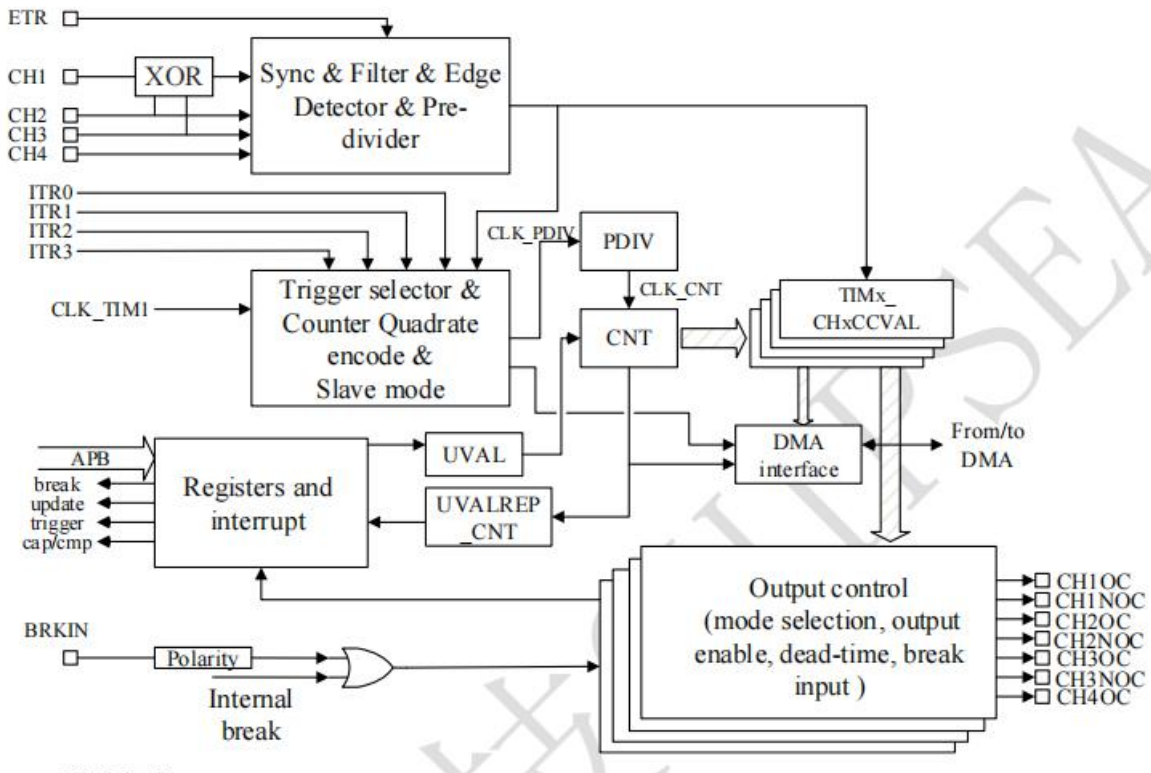


图 6 高级定时器框图

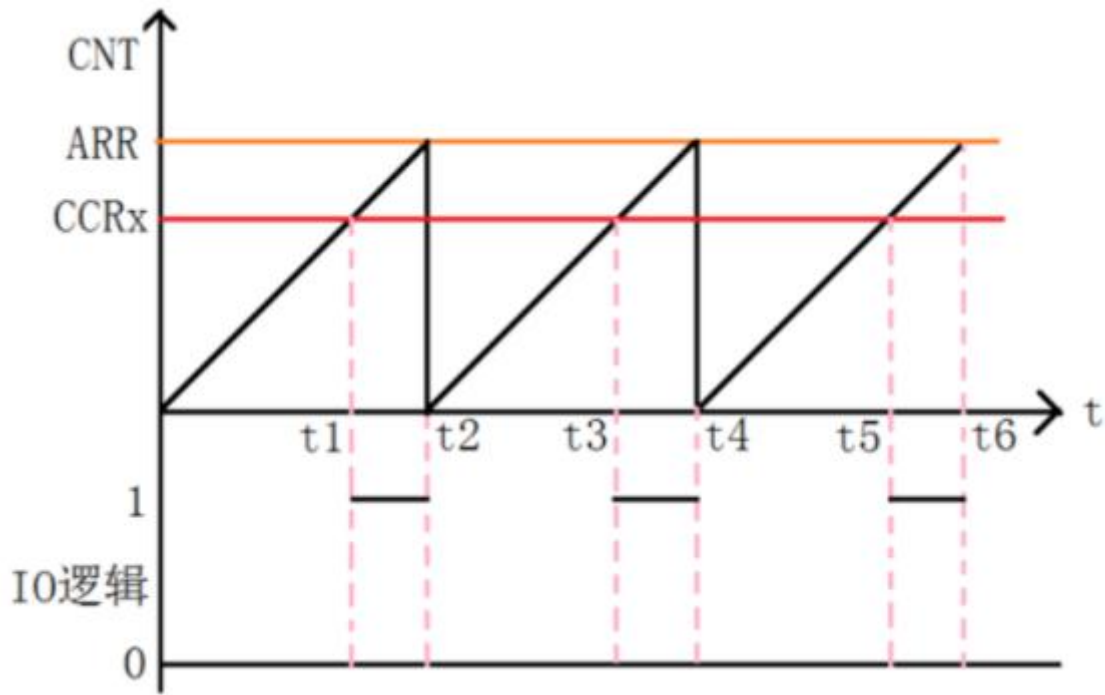


图 7 PWM 输出原理框图

上桥臂的 GPIO 配置为复用推挽输出模式，使用芯海 CS32F03X 的 TIM1 高级定时器外设产生 PWM 输出，下桥臂对应的 GPIO 配置为推挽输出，配置 GPIO 的代码如下：

```

GPIO_InitTypeDef GPIO_InitStructure;
/*****

U:H-PA10-CH3 L-PB1-CH3N
V:H-PA9 -CH2 L-PB0-CH2N
W:H-PA8 -CH1 L-PA7-CH1N
*****/

RCC_AHBPeriphClockCmd( RCC_AHBPeriph_GPIOA|RCC_AHBPeriph_GPIOB
ENABLE);//使能 GPIO 时钟
//配置上管//复用推挽
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8 | GPIO_Pin_9 | GPIO_Pin_10 ;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_DOWN ;
GPIO_Init(GPIOA, &GPIO_InitStructure);
    
```



```
/*端口映射*/
```

```
GPIO_PinAFConfig(GPIOA, GPIO_PinSource8, GPIO_AF_2);  
GPIO_PinAFConfig(GPIOA, GPIO_PinSource9, GPIO_AF_2);  
GPIO_PinAFConfig(GPIOA, GPIO_PinSource10, GPIO_AF_2);
```

```
//配置下管//推挽
```

```
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_7;  
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;  
GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;  
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;  
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;  
GPIO_Init(GPIOA, &GPIO_InitStructure);  
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_1 | GPIO_Pin_0 ;  
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;  
GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;  
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;  
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;  
GPIO_Init(GPIOB, &GPIO_InitStructure);
```

配置 TIM1 高级定时器产生 PWM 的代码如下：

```
TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;  
TIM_OCInitTypeDef TIM_OCInitStructure;  
/* TIM1 Configuration -----*/  
RCC_APB2PeriphClockCmd(RCC_APB2Periph_TIM1 , ENABLE);//使能 TIM1 时钟  
/* Time Base configuration 初始化定时器*/  
TIM_DeInit(TIM1);  
TIM_TimeBaseStructure.TIM_Prescaler =0 ; //预分频  
TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_CenterAligned1 ;  
TIM_TimeBaseStructure.TIM_Period =1499-1; //设置 PWM 周期 24MHz/1500=16kHz  
TIM_TimeBaseStructure.TIM_ClockDivision = 0; //时钟分频系数  
TIM_TimeBaseInit(TIM1, &TIM_TimeBaseStructure);  
/* Channel 1, 2,3 Configuration in PWM mode */  
TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM1; //PWM 模式 1:CNT 与 CCR
```

比较

```
TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable; //输出使能
TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High; //设置初始极性 -高电平
TIM_OCInitStructure.TIM_OCIdleState = TIM_OCIdleState_Reset;
TIM_OCInitStructure.TIM_OCNIdleState = TIM_OCIdleState_Reset;
TIM_OCInitStructure.TIM_Pulse = 0; //脉冲宽度
TIM_OC1Init(TIM1, &TIM_OCInitStructure); //使能通道 1
TIM_OC2Init(TIM1, &TIM_OCInitStructure); //使能通道 2
TIM_OC3Init(TIM1, &TIM_OCInitStructure); //使能通道 3
控制无刷电机换相的代码如下：（此处为按照顺序依次使无刷电机换下一相）
Motor.Duty=75; //配置 PWM 占空比为 5%（TIM_Period*5%=75）
while(1)
{
Motor.PhaseCnt++; //无刷电机目标相加 1
if(Motor.PhaseCnt > 6)
Motor.PhaseCnt = 1;
switch(Motor.PhaseCnt) //根据无刷电机目标相位配置 PWM 和 GPIO
{
case 1:
MOS_Q15PWM(); //UV
break;
case 2:
MOS_Q16PWM(); //UW
break;
case 3:
MOS_Q26PWM(); //VW
break;
case 4:
MOS_Q24PWM(); //VU
break;
case 5:
MOS_Q34PWM(); //WU
```

```
break;
case 6:
MOS_Q35PWM();//WV
break;
default:
break;
}
Delay_us(5000);//延迟 5ms 后换下一相
}
void MOS_Q15PWM(void)
{
TIM1->CCR1 = 0;
TIM1->CCR2 = 0;
TIM1->CCR3 = Motor.Duty;
GPIOB->BSRR = U_Mos_L_Pin;
GPIOA->BSRR = W_Mos_L_Pin;
GPIOB->BRR = V_Mos_L_Pin;
}
void MOS_Q16PWM(void)
{
TIM1->CCR1= 0;
TIM1->CCR2 = 0;
TIM1->CCR3 = Motor.Duty;
GPIOB->BSRR = U_Mos_L_Pin | V_Mos_L_Pin;
GPIOA->BRR = W_Mos_L_Pin;
}
void MOS_Q26PWM(void)
{
TIM1->CCR1=0;
TIM1->CCR3=0;
TIM1->CCR2 = Motor.Duty;
GPIOB->BSRR = U_Mos_L_Pin | V_Mos_L_Pin;
GPIOA->BRR = W_Mos_L_Pin;
```

```
}  
void MOS_Q24PWM(void)  
{  
TIM1->CCR1 = 0;  
TIM1->CCR3 = 0;  
TIM1->CCR2 = Motor.Duty;  
GPIOB->BSRR = V_Mos_L_Pin;  
GPIOA->BSRR = W_Mos_L_Pin;  
GPIOB->BRR = U_Mos_L_Pin;  
}  
void MOS_Q34PWM(void)  
{  
TIM1->CCR2 = 0;  
TIM1->CCR3 = 0;  
TIM1->CCR1 = Motor.Duty;  
GPIOB->BSRR = V_Mos_L_Pin;  
GPIOA->BSRR = W_Mos_L_Pin;  
GPIOB->BRR = U_Mos_L_Pin;  
}  
void MOS_Q35PWM(void)  
{  
TIM1->CCR2 = 0;  
TIM1->CCR3 = 0;  
TIM1->CCR1 = Motor.Duty;  
GPIOB->BSRR = U_Mos_L_Pin;  
GPIOA->BSRR = W_Mos_L_Pin;  
GPIOB->BRR = V_Mos_L_Pin;  
}
```

编译代码，下载程序运行可直以看到电机成功转动起来，用示波器可以看到三相线上的输出电压波形图如下：

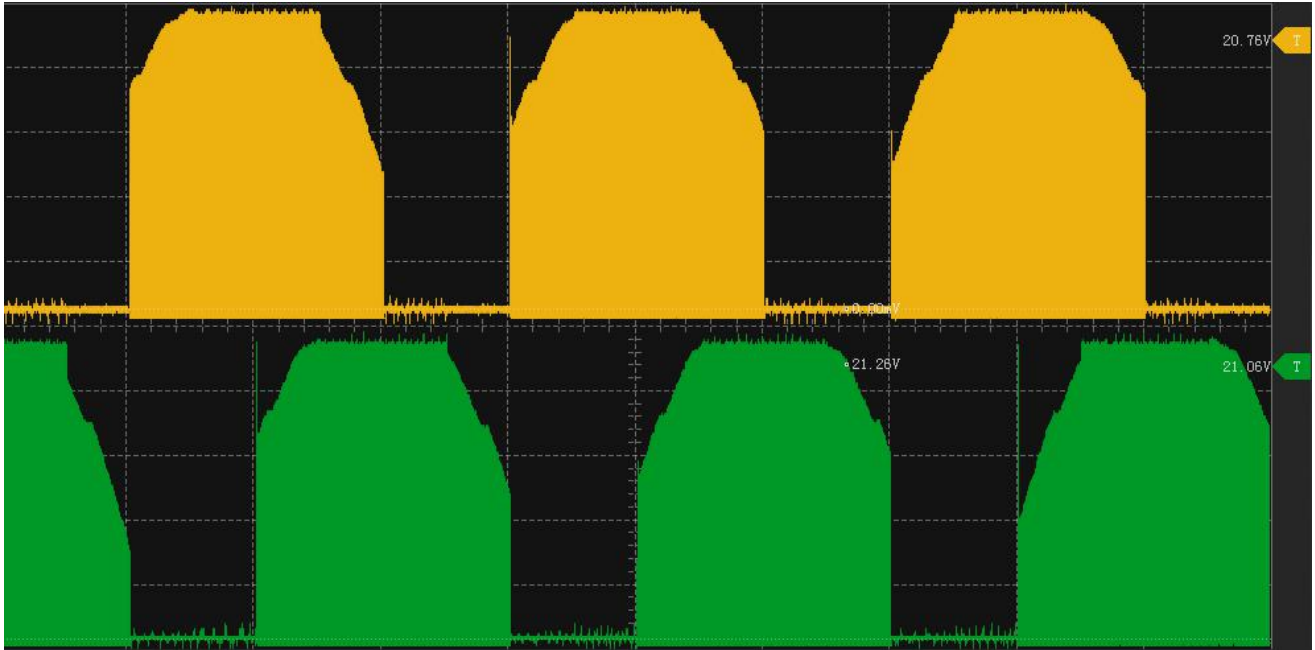


图 8 无刷电机相电压波形

3 注意事项

- 1.本次实验为直流无刷电机的开环拖动换相实验，占空比不宜过高，推荐 10%以内。
- 2.MOS 预驱芯片一般具备防止上下 MOS 管直通的功能，某些驱动板可能不具备该功能，建议先用示波器观察上下管控制波形不会同时输出高电平，再给驱动板供电。
- 3.MOS 管耐压值应大于无刷电机的额定电压。
- 4.不同的电机额定转速不同，换相间隔时间需要根据实际情况调整。

免责声明和版权公告

本文档中的信息，包括供参考的 URL 地址，如有变更，恕不另行通知。

本文档可能引用了第三方的信息，所有引用的信息均为“按现状”提供，芯海科技不对信息的准确性、真实性做任何保证。

芯海科技不对本文档的内容做任何保证，包括内容的适销性、是否适用于特定用途，也不提供任何其他芯海科技提案、规格书或样品在他处提到的任何保证。

芯海科技不对本文档是否侵犯第三方权利做任何保证，也不对使用本文档内信息导致的任何侵犯知识产权的行为负责。本文档在此未以禁止反言或其他方式授予任何知识产权许可，不管是明示许可还是暗示许可。

Wi-Fi 联盟成员标志归 Wi-Fi 联盟所有。蓝牙标志是 Bluetooth SIG 的注册商标。

文档中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

版权归 © 2022 芯海科技（深圳）股份有限公司，保留所有权利。



芯海科技
CHIPSEA

股票代码:688595