

# 应用笔记

## Application Note

文档编号: **AN1100**

**APM32F035\_MOTOR EVAL 有感方波方控制方案**

版本: **V1.1**

## 目录

<b>1</b>	<b>整体介绍 .....</b>	<b>3</b>
1.1	项目概述.....	3
1.2	APM32F035 芯片资源.....	3
<b>2</b>	<b>硬件介绍 .....</b>	<b>5</b>
2.1	整体硬件电路.....	5
2.2	接口电路及设置.....	6
2.3	系统硬件实物.....	9
<b>3</b>	<b>软件介绍 .....</b>	<b>10</b>
3.1	整体程序架构.....	10
3.2	状态机介绍 .....	11
3.3	顶层外设配置.....	11
3.4	关键参数设置.....	14
3.5	调试方法.....	15
<b>4</b>	<b>实际测试波形 .....</b>	<b>17</b>
<b>5</b>	<b>修订历史 .....</b>	<b>18</b>

# 1 整体介绍

## 1.1 项目概述

APM32F035 是极海半导体针对电机控制推出的专用芯片，本设计基于 APM32F035 设计了有感方波控制方案。具体设计规格如下表所示：

表格 1 设计规格

控制方式	有感方波六步换相
PWM 调制方式	HPWM_LON
角度获取	HALL Sensor
PWM 频率	20 KHz
极对数	2 对极
电机转速	0~3000RPM
起动方式	6 Step
保护功能	过压，欠压，软件过流，硬件过流
代码大小	<10Kbytes
开发软件	Keil C (V5.23 版本及以上)

## 1.2 APM32F035 芯片资源

APM32F035 是一款基于 Arm Cortex-M0+内核，集成 FOC 算法中常用的数学运算加速器（Cordic, Svpwm, 硬件除法等），并且集成了运放、比较器等模拟外设，以及 CAN 控制器的高性能电机控制专用 MCU。

表格 2 APM32F035 系列芯片功能和外设

产品	APM32F035	
型号	C8T7	K8T7
封装	LQFP48	LQFP32
内核及最大工作频率	Arm® 32-bit Cortex®-M0+@72MHz	
M0CP 协处理器	1	
闪存 (KB)	64	
SRAM(KB)	10	
定时器	32 位/16 位通用	1/2
	16 位高级	1
	16 位基本	2
	24 位计数器	1

产品		APM32F035	
	看门狗 (WDT)	2 (1 个独立看门狗+1 个窗口看门狗)	
	实时时钟	1	
通信接口	USART	2	
	SPI/I2S	1/1	
	I2C	1	
	CAN	1	
12 位 ADC	单元	1	
	外部通道	16	12
	内部通道	3	
比较器		2	
运算放大器		4	2
GPIOs		42	27
工作温度		环境温度: -40°C 至 105°C 结温度: -40°C 至 125°C	
工作电压		2.0~3.6V	

## 2 硬件介绍

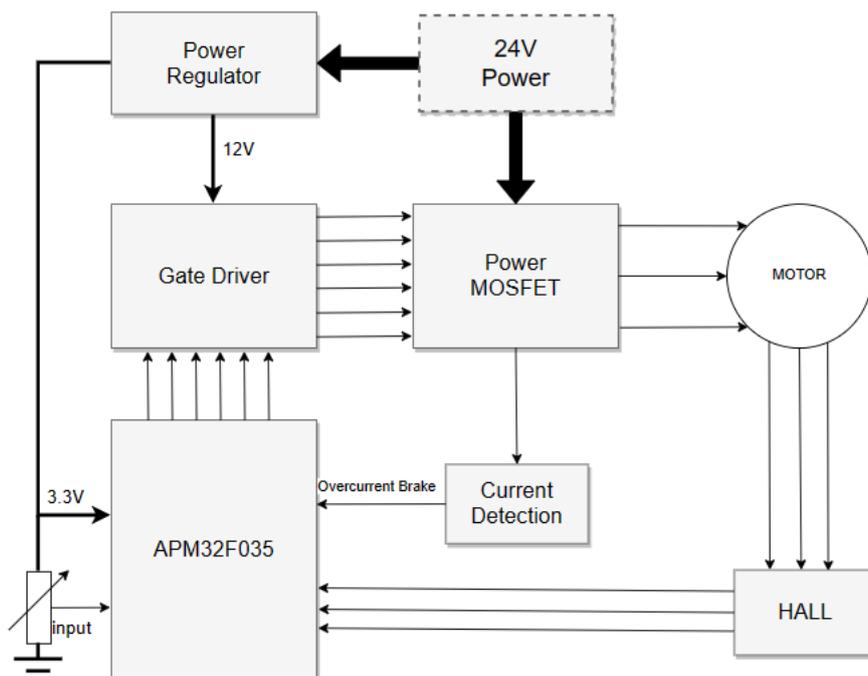
### 2.1 整体硬件电路

整体硬件系统是采用外部 24V 电源供电，并经过相应的电源降压电路转换后输出稳定的 12V、5V、3.3V 电压，其中 12V 电压输出给到 Gate driver IC、3.3V 电压输出给到 APM32F035 系列微处理器，而功率开关管则直接使用 24V 电源。同时，该方案采用可变电阻旋钮调节 0~3.3V 的电压输入作为速度命令的输入端，以此调节电机转速。用户实际使用可直接通过转动可变电阻旋钮以此调节输入电压，同时当输入电压值超过起动阈值时，电机将会启动运行，而当电压值低于阈值时电机将会关闭运行。

当电机启动后，通过捕获 HALL 信号的跳变情况进而确认六个换相点，进行对应六种不同的逆变器 MOS 管驱动顺序实现 BLDC 电机的六步换相。

硬件框图如图所示。

图 1 硬件系统框图









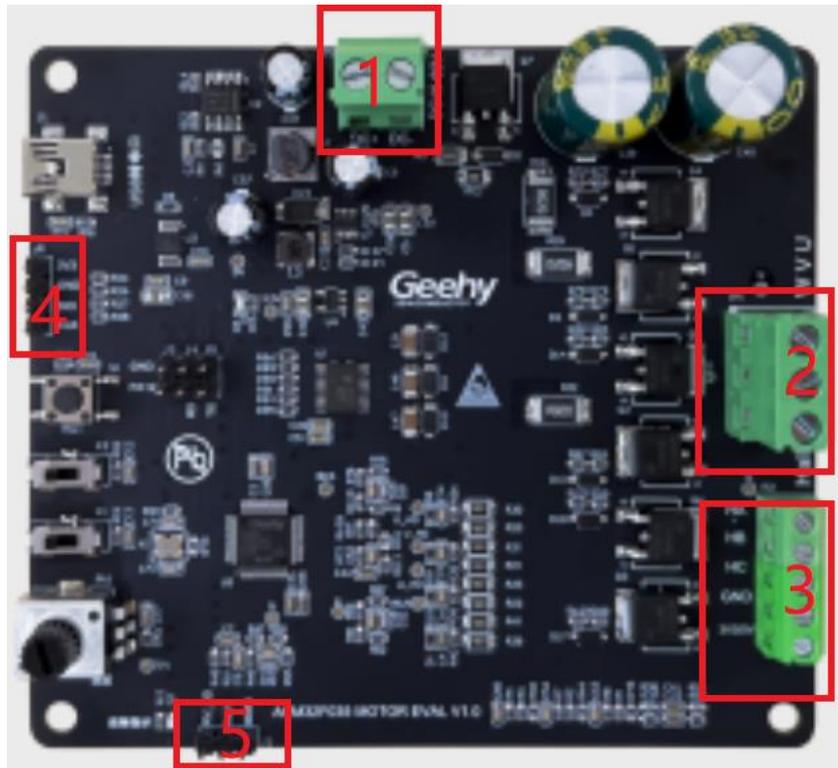
如图所示，APM32F035 MOTOR EVAL V1.0 板级硬件预留有 USB 转串口以及故障指示灯，以便开发者调试使用；两处按键负责控制电机运行方向及锁车功能。

## 2.3 系统硬件实物

系统实物图如图所示，其中主要有以下四个接口：

- (1) 电源输入接口（接入 24V，注意正负）
- (2) 三相电机接口（相序只会影响旋转方向）
- (3) HALL 输入接口
- (4) SWD 调试接口
- (5) 跳线帽端口需接上

图 9 硬件实物图



### 3 软件介绍

#### 3.1 整体程序架构

此项目整体代码架构主要可划分为 4 层，用户层、外设驱动层、电机控制驱动层以及电机算法层，具体功能描述如下：

##### 3.1.1 用户层 (USER 层)

**main.c:** 主函数入口，负责初始化电机参数、底层外设、中断优先级、while 循环及低速状态机环路的切换；

**apm32f035\_int.c:** 所有的中断处理函数，重点包括 TMR1 中断函数、ADC 中断处理函数；

**user\_function.c:** 包括电机参数的初始化配置、参数复位等处理函数；

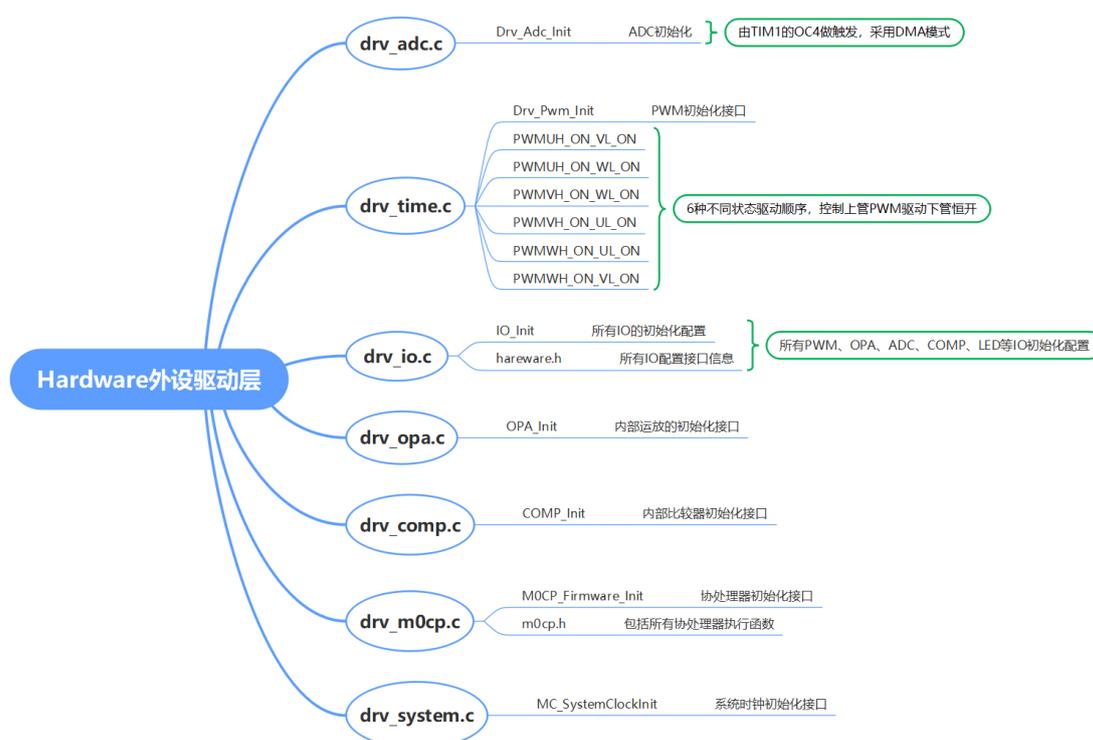
**parameter.h:** 包括了所有的需要配置参数信息；

**board.c:** 包括板级的底层外设初始化配置函数。

##### 3.1.2 外设驱动层 (HARDWARE 层)

外设驱动层主要负责 APM32F035 芯片的外设驱动函数及配置，主要涉略包括 GPIO、PWM、ADC、OPA、COMP、M0CP 协处理器等，具体如下图所示。

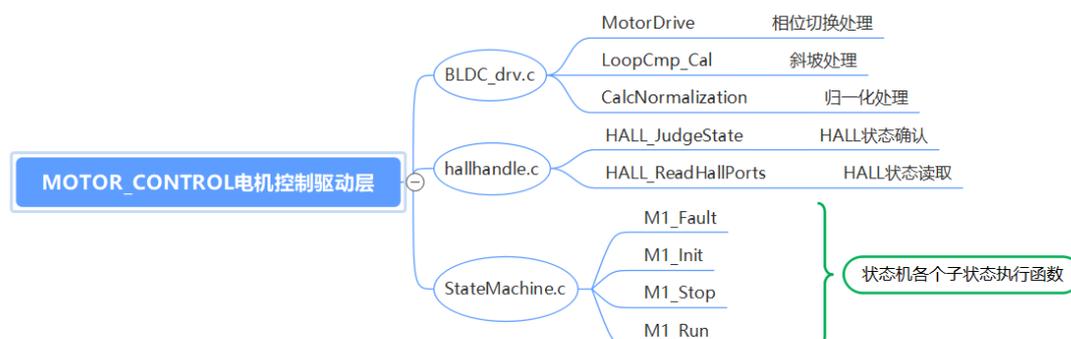
图 10 外设驱动层



### 3.1.3 电机控制驱动层 (MOTOR\_CONTROL 层)

电机控制驱动层主要负责电机的控制运行逻辑及核心处理算法调用, 具体如下图所示。

图 11 电机控制驱动层



### 3.1.4 极海电机算法层 (Geehy\_MCLIB 层)

电机算法层包括过数学库等库函数。

## 3.2 状态机介绍

本案例中采用了主状态机嵌套子状态机的结构, 如下所示:

四个主状态: **INIT**、**STOP**、**FAULT**、**RUN**;

对主状态机的描述如下:

**Fault:** 当系统有错误发生时一直处于此状态, 直到错误的标志位被清除;

而后会延时一段时间, 然后从 **Fault** 状态跳转到 **STOP** 状态等待起动命令

**Init:** 此主状态执行变量初始化;

**Stop:** 系统完成初始化等待速度命令。此状态内下关闭 PWM 输出;

**Run:** 系统处于运行状态, 当有 **Stop** 命令或 **Fault** 异常时会停止系统的运行。

## 3.3 顶层外设配置

### 3.3.1 PWM 输出配置

```
void Drv_Pwm_Init(uint16_t u16_Period,uint16_t u16_DeadTime)
```

(1) PWM 通用配置如下所示:

设置 PWM clock 分频为 1 分频, 选用中心对齐模式 1, 设置重复计数器为 1, 如下图所示。

图 12 PWM 通用配置

```

/* Time Base configuration, init timel freq */
TIM_TimeBaseInitStructure.period           = ul6_Period;
TIM_TimeBaseInitStructure.div              = 0;
TIM_TimeBaseInitStructure.counterMode     = TMR_COUNTER_MODE_CENTERALIGNED1;
TIM_TimeBaseInitStructure.clockDivision   = TMR_CKD_DIV1;
TIM_TimeBaseInitStructure.repetitionCounter = 1;
TMR_ConfigTimeBase(TMRL, &TIM_TimeBaseInitStructure);

```

## (2) PWM 输出状态配置

设置 PWM 上下管输出状态，并使能配置上下管 PWM 输出有效，

配置使能刹车，配置刹车输入极性，关闭刹车硬件自动恢复。

图 13 PWM 输出状态配置

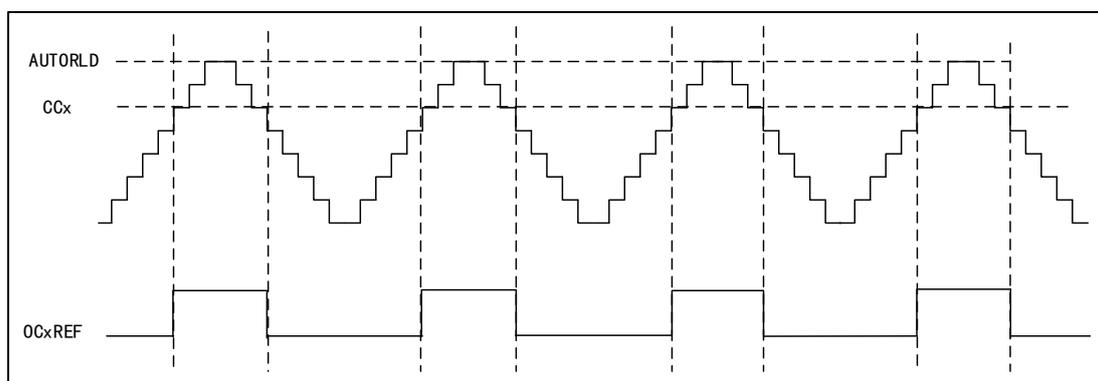
```

/* Automatic Output enable, Break, dead-time and lock configuration */
TIM_BDTRInitStructure.RMOS_State          = TMR_RMOS_STATE_ENABLE;
TIM_BDTRInitStructure.IMOS_State          = TMR_IMOS_STATE_ENABLE;
TIM_BDTRInitStructure.lockLevel           = TMR_LOCK_LEVEL_OFF;
TIM_BDTRInitStructure.deadTime            = ul6_DeadTime; //死区时间
/**
** Brake configuration: enable brake
** Brake input polarity: active in low level
** Auto output enable configuration: Disable MOE bit hardware control
**/
TIM_BDTRInitStructure.breakState          = TMR_BREAK_STATE_ENABLE;
TIM_BDTRInitStructure.breakPolarity      = TMR_BREAK_POLARITY_LOW;
TIM_BDTRInitStructure.automaticOutput    = TMR_AUTOMATIC_OUTPUT_DISABLE;
TMR_ConfigBDT(TMRL, &TIM_BDTRInitStructure);

/* pwm driver set, channel 1,2,3,4 set pwm mode */
TIM_OCInitStructure.OC_Mode               = TMR_OC_MODE_PWM2;
TIM_OCInitStructure.OC_OutputState       = TMR_OUTPUT_STATE_ENABLE; //TMR_OUTPUT_STATE_DISABLE;
TIM_OCInitStructure.OC_OutputNState     = TMR_OUTPUT_NSTATE_ENABLE; //TMR_OUTPUT_NSTATE_DISABLE;
TIM_OCInitStructure.Pulse                = 0;
TIM_OCInitStructure.OC_Polarity          = TMR_OC_POLARITY_HIGH;
TIM_OCInitStructure.OC_NPolarity         = TMR_OC_NPOLARITY_HIGH; //互补输出极性-----
TIM_OCInitStructure.OC_Idlestate         = TMR_OCIDLESTATE_RESET; //TMR_OCIDLESTATE_SET;
TIM_OCInitStructure.OC_NIdlestate        = TMR_OCNIDLESTATE_RESET; //TMR_OCNIDLESTATE_SET;

```

图 14 中央对齐模式的时序图



在递增计数时，当  $TMR1\_CNT < TMR1\_CCR1$  时通道 1 为无效电平，否则为有效电平；  
在递减计数时，当  $TMR1\_CNT > TMR1\_CCR1$  时通道 1 为有效电平，否则为无效电平。

### 3.3.2 ADC 配置

void Drv\_Adc\_Init(void)

#### (1) ADC 底层配置

采用 DMA 模式, ADC 量化后的数据直接搬运到 ADC\_ConvertedValue 数组中存储, ADC 触发条件采用 TMR1 的 CC4 作为触发源, 开启 ADC 使能及配置 ADC 中断优先级及其使能。具体如下图所示。

图 15 ADC 底层配置

```
void Drv_Adc_Init(void)
{
    ...ADC_Config_T...ADC_InitStructure;
    ...DMA_Config_T...DMA_InitStructure;
    ...DMA_InitStructure.peripheralAddress...=(uint32_t)&(ADC->DATA); //ADC地址
    ...DMA_InitStructure.memoryAddress...=(uint32_t)&ADC_ConvertedValue[0]; //内存地址
    ...DMA_InitStructure.direction...=DMA_DIR_PERIPHERAL; //方向(从外设到内存)
    ...DMA_InitStructure.bufferSize...=3; //TOTAL_CHANNEL; //传输内容的大小---传输次数
    ...DMA_InitStructure.peripheralInc...=DMA_PERIPHERAL_INC_DISABLE; //外设地址固定
    ...DMA_InitStructure.memoryInc...=DMA_MEMORY_INC_ENABLE; //DMA_MEMORY_INC_ENABLE; //内存地址固定
    ...DMA_InitStructure.peripheralDataSize...=DMA_PERIPHERAL_DATASIZE_HALFWORD; //外设数据单位
    ...DMA_InitStructure.memoryDataSize...=DMA_MEMORY_DATASIZE_HALFWORD; //内存数据单位
    ...DMA_InitStructure.circular...=DMA_CIRCULAR_ENABLE; //DMA模式: 循环传输
    ...DMA_InitStructure.priority...=DMA_PRIORITY_LEVEL_VERYHIGH; //优先级: 高
    ...DMA_InitStructure.memoryToMemory...=DMA_M2M_DISABLE; //禁止内存到内存的传输

    ...DMA_Config(DMA_CHANNEL_1, &DMA_InitStructure); //配置DMA的1通道
    ...DMA_Enable(DMA_CHANNEL_1);
    ...ADC_ClockMode(ADC_CLOCK_MODE_ASYNCCLK); //48M/4=12mADC_CLOCK_MODE_SYNCLKDIV4...
    ...ADC_ConfigStructInit(&ADC_InitStructure);
    ...ADC_InitStructure.convMode...=ADC_CONVERSION_SINGLE;
    ...ADC_InitStructure.scanDir...=ADC_SCAN_DIR_UPWARD;
    ...ADC_InitStructure.extTrigConv1...=ADC_EXT_TRIG_CONV_TRG1; //...timer1-CC4
    ...ADC_InitStructure.extTrigEdgel...=ADC_EXT_TRIG_EDGE_RISING;
    ...ADC_InitStructure.dataAlign...=ADC_DATA_ALIGN_RIGHT;
    ...ADC_InitStructure.resolution...=ADC_RESOLUTION_12B; ...
    ...ADC_Config(&ADC_InitStructure);
    ...ADC_ConfigChannel(ADC_CHANNEL_7 | ADC_CHANNEL_9 | ADC_CHANNEL_12, ADC_SAMPLE_TIME_1_5);
    ...ADC->CFG1_B.OVRMAG = 1; ...
    ...ADC_EnableInterrupt(ADC_INT_CS);
    //=====ADC中断使用=====
    ...NVIC_EnableIRQ(ADC_COMP_IRQn);
    ...NVIC_SetPriority(ADC_COMP_IRQn, 0); ...
    ...ADC_DMAResquestMode(ADC_DMA_MODE_CIRCULAR);
    ...ADC_EnableDMA();
    ...ADC_Enable(); ...
    ...ADC_StartConversion(); //必须要启动一下
}
}
```

### 3.3.3 OPA 与 COMP 底层配置

#### (1) OPA 底层配置

配置 OPA 底层配置需先配置 OPA 管脚, 先 DISABLE 运算放大器 OPA, 配置为使用外部电阻网络, 而后再 ENABLE。如下图所示;

图 16 OPA 底层配置

```
void OPA_Init(void)
{
    ...OPA_Disable(OPA1);
    ...OPA_Disable(OPA2);
    ...OPA_Disable(OPA3);
    ...OPA_Disable(OPA4);
    ...OPA_SelectGainFactor(OPA1,OPA_GAIN_FACTOR_0);
    ...OPA_SelectGainFactor(OPA2,OPA_GAIN_FACTOR_0);
    ...OPA_SelectGainFactor(OPA3,OPA_GAIN_FACTOR_0);
    ...OPA_SelectGainFactor(OPA4,OPA_GAIN_FACTOR_0);
    ...OPA_Enable(OPA1);
    ...OPA_Enable(OPA2);
    ...OPA_Enable(OPA3);
    ...OPA_Enable(OPA4);
}
```

(1) COMP 底层配置

COMP 用于过流异常检测，配置 COMP 底层配置需先配置 COMP 管脚，设置 COMP 输出为接入 TMR1 的 BKIN，设置输出反向，低电平触发 TMR1 的 BKIN，如下图所示：

图 17 COMP 底层配置

```
void COMP_Init(void)
{
    ...COMP_Config_T compConfig;
    .../* Configure COMP1 */
    ...COMP_ConfigStructInit(&compConfig);
    ...compConfig.invertingInput = COMP_INVERTING_INPUT_PA1;
    ...compConfig.output = COMP_OUTPUT_TIM1BKIN;
    ...compConfig.outputPol = COMP_OUTPUTPOL_NONINVERTED;
    ...compConfig.hysterrsis = COMP_HYSTERRSIS_NO;
    ...compConfig.mode = COMP_MODE_HIGHSPEED;
    ...COMP_Config(COMP_SELECT_COMP1,&compConfig);
    .../* Enable COMP2 */
    ...COMP_Enable(COMP_SELECT_COMP1);
}
```

### 3.4 关键参数设置

此系统中所有参数均在用户层的 parameter.h 中进行配置，主要分为系统参数、底板相关参数、状态机相关参数、电机相关参数等。如下所述：

#### 3.4.1 系统参数

表格 3 系统参数

参数名称	参数说明	设定值
SYS_REFV	系统供电电压	3.3 (V)

参数名称	参数说明	设定值
SYSCCLK_HSE_72MHz	系统主频	72000000 (Hz)
PWMFREQ	PWM 频率	20000 (Hz)
DEAD_TIME	PWM 死区时间	1.0 (us)
SLOWLOOP_FREQ	慢速环控制频率	1000 (Hz)
HALL_TCY	捕获时钟定时精度	1 (μs)
HALL_PSC_LOAD	捕获时钟预分频值	71
HALL_PRIOD	捕获时钟定时计数值	50000 (μs)
ECW	direction 方向设定	1
ECCW	direction 方向设定	0

### 3.4.2 底板硬件参数

表格 4 底板硬件参数

参数名称	参数说明	设定值
ADC_REFV	ADC 参考电压	3.3 (V)
R_SHUNT	采样电阻阻值	0.02 (Ω)
CURRENT_OPA_GAIN	运放放大倍数	5.0
I_MAX	电流标么化基准值	16.5 (A)
UDC_MAX	电压标么化基准值	69.0 (V)
U_MAX	相电压标么化基准值	39.83 (V)

### 3.4.3 电机相关参数

表格 5 电机相关参数

参数名称	参数说明	设定值
POLEPAIRS	电机极对数	2 (unit)
RAMP_UP	速度上升斜率	1000 (rpm/s)
RAMP_DOWN	速度下降斜率	1000 (rpm/s)
M1_SPEED_KP_Q10	速度环 KP 参数 Q10 格式	512
M1_SPEED_KI_Q10	速度环 KI 参数 Q10 格式	2

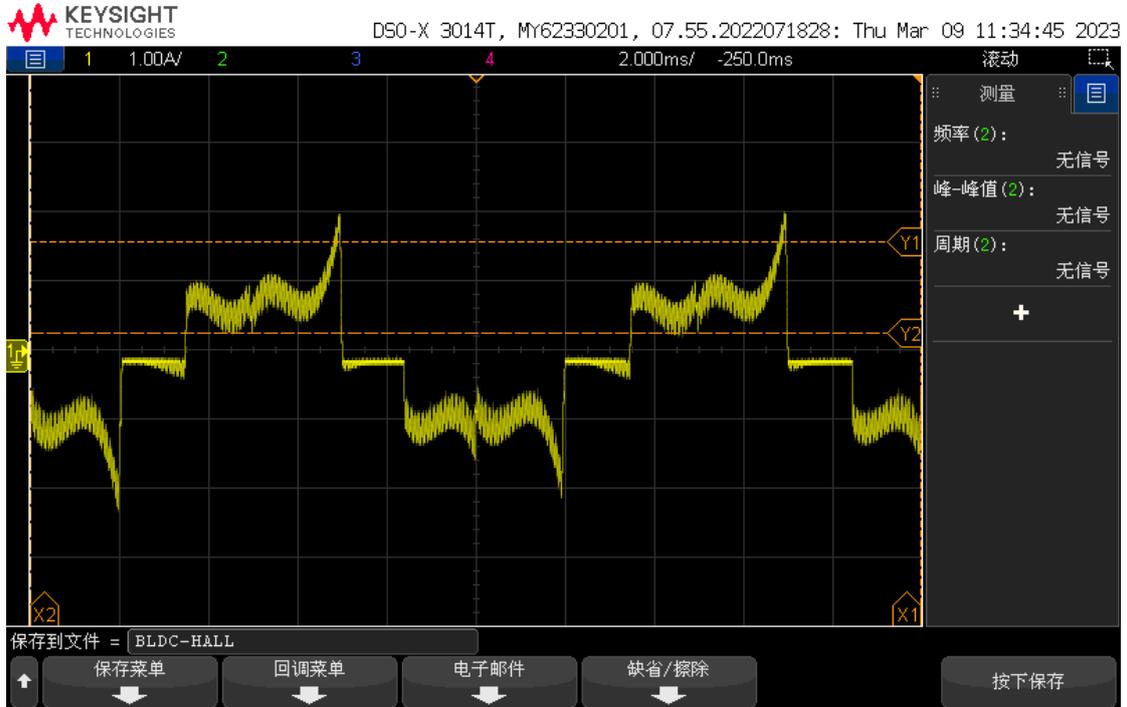
## 3.5 调试方法

- (1) 检查硬件连接: 检查电机、电机驱动器和控制器之间的连接是否正确, 以及电源是否稳定, 确保所有接口都插好并紧固, 没有损坏或短路。

- (2) 配置参数：根据电机和驱动器的规格，配置控制器的参数，例如电机额定电流、电机参数、电机的极对数、电机的最大转速等。确保所有参数正确设置。
- (3) 进行空载测试运行：在未安装电机负载的情况下，进行空载测试，检查电机是否可以正常启动和转动，以及转速是否符合要求。注：需核对电机的 HALL 运行顺序是否正确，是采用示波器采集 HALL 信号端口，手动转动电机一圈确认 HALL 正反转运行的顺序。
- (4) 进行负载测试运行：在安装电机负载的情况下，进行负载测试，检查电机在负载下的性能表现，例如转速、扭矩等。
- (5) 进行速度 PID 调试：使用 PID 调节器来控制电机的响应速度。可以通过改变 PID 参数来优化电机的响应和稳定性。调试时应该记录实验结果，以便后续参考。
- (6) 进行性能测试：在完成上述步骤后，可以进行一些性能测试，例如测量电机的最大速度、最大扭矩和效率等。可以使用一些测试设备来进行测量，例如转速表、负载测试器、功率计等。

## 4 实际测试波形

图 18 实际波形图



## 5 修订历史

表格 6 文件修订历史

日期	修订	变更历史
2023.07.26	1.0	新建
2023.08.14	1.1	(1) 修改产品信息表 (2) 修改格式

## 声明

本手册由珠海极海半导体有限公司（以下简称“极海”）制订并发布，所列内容均受商标、著作权、软件著作权相关法律法规保护，极海保留随时更正、修改本手册的权利。使用极海产品前请仔细阅读本手册，一旦使用产品则表明您（以下称“用户”）已知悉并接受本手册的所有内容。用户必须按照相关法律法规和本手册的要求使用极海产品。

### 1、权利所有

本手册仅应当被用于与极海所提供的对应型号的芯片产品、软件产品搭配使用，未经极海许可，任何单位或个人均不得以任何理由或方式对本手册的全部或部分内容进行复制、抄录、修改、编辑或传播。本手册中所列带有“®”或“TM”的“极海”或“Geehy”字样或图形均为极海的商标，其他在极海产品上显示的产品或服务名称均为其各自所有者的财产。

### 2、无知识产权许可

极海拥有本手册所涉及的全部权利、所有权及知识产权。

极海不应因销售、分发极海产品及本手册而被视为将任何知识产权的许可或权利明示或默示地授予用户。

如果本手册中涉及任何第三方的产品、服务或知识产权，不应被视为极海授权用户使用前述第三方产品、服务或知识产权，除非在极海销售订单或销售合同中另有约定。

### 3、版本更新

用户在下单购买极海产品时可获取相应产品的最新版的手册。

如果本手册中所述的内容与极海产品不一致的，应以极海销售订单或销售合同中的约定为准。

### 4、信息可靠性

本手册相关数据经极海实验室或合作的第三方测试机构批量测试获得，但本手册相关数据难免会出现校正笔误或因测试环境差异所导致的误差，因此用户应当理解，极海对本手册中可能出现的该等错误无需承担任何责任。本手册相关数据仅用于指导用户作为性能参数参照，不构成极海对任何产品性能方面的保证。

用户应根据自身需求选择合适的极海产品，并对极海产品的应用适用性进行有效验证和测试，以确认极海产品满足用户自身的需求、相应标准、安全或其它可靠性要求；若因用户未充分对极海产品进行有效验证和测试而致使用户损失的，极海不承担任何责任。

### 5、合规要求

用户在使用本手册及所搭配的极海产品时，应遵守当地所适用的所有法律法规。用户应了解产品可能受到产品供应商、极海、极海经销商及用户所在地等各国有关出口、再出口或其它法律的限制，用户（代表其本身、子公司及关联企业）应同意并保证遵守所有关于取得极海产品及 / 或技术与直接产品的出口和再出口适用法律与法规。

## 6、免责声明

本手册由极海“按原样”（as is）提供，在适用法律所允许的范围内，极海不提供任何形式的明示或暗示担保，包括但不限于对产品适销性和特定用途适用性的担保。

对于用户后续在针对极海产品进行设计、使用的过程中所引起的任何纠纷，极海概不承担责任。

## 7、责任限制

在任何情况下，除非适用法律要求或书面同意，否则极海和/或以“按原样”形式提供本手册的任何第三方均不承担损害赔偿责任，包括任何一般、特殊因使用或无法使用本手册相关信息而产生的直接、间接或附带损害（包括但不限于数据丢失或数据不准确，或用户或第三方遭受的损失）。

## 8、适用范围

本手册的信息用以取代本手册所有早期版本所提供的信息。

©2023 珠海极海半导体有限公司 - 保留所有权利