YS32MD21x_220_YS32F031 BLDC 比较器 SDK 调试教程 V1.1.x

目录

SDK文件框架及程序流程-占用硬件资源

硬件资源	说明
TIM1模块	6路PWM输出
TIM14模块	相位时间检测
COMP2模块	比较器2检测反电动势
ADC注入、ADC 规则 (DMA-CH1)	ADC 注入（工作电压/相电流）。 ADC 规则 (用户自定义)。
COMP1模块(预留)	硬件过流检测(预留)

SDK文件框架及程序流程-SDK文件说明

文件夹	文件	说明
BLDC	BLDC_ConfigParameter.h	参数配置文件
	BLDC_HardwareInit.c BLDC_HardwareInit.h	硬件配置，比较器、ADC、TIM1等
	BLDC_Motor.c BLDC_Motor.h	逻辑控制(一般不修改)
	BLDC_App.c BLDC_App.h BLDC_AppPwmIN.c BLDC_AppPwmIN.h	应用层(用户功能实现)
CmpLib	BLDC_CmpLib.h	库头文件
	BLDC_CmpLib_MD2xx_F031_V1.1.2.lib	比较器库
Libraries	---	标准芯片驱动
MDK-ARM	---	MDK工程
User	---	用户文件夹

SDK文件框架及程序流程 – 常用变量、宏说明

T_MotorCtrl MotorCtrl; //电机控制结构体

```
MotorCtrl.RunStart              //启动控制，=0停止，=1启动。
MotorCtrl.RunDir                //运行方向，=0正转，=1反转。切换方向必须先停止，改变方向再延时启动。
MotorCtrl.CtrlStatus            //控制状态

MotorCtrl.SpeedValue            //当前速度值

MotorCtrl.RunOpenPwm_Set        //开环 PwmDuty
MotorCtrl.SpeedPI_Set           //速度环，速度设置值
MotorCtrl.SpeedPI               //电流环PI参数宏
MotorCtrl.CurPI_Set             //电流环，电流设置值
MotorCtrl.CurPI                 //速度环PI参数宏

MotorCtrl.VccData_LPF           //VCC电压ADC
MotorCtrl.IbusData_LPF          //相电流ADC

//
RPM_60D(RPM)                    //RPM 转 60度TIME
T60D_RPM(TIME)                  //60度TIME 转 RPM
s16Q12_PWM(DUTY)                //占空比设置宏范围：0.0 ~ 1.0
CONFIG_VCC_TO_ADC_VALUE(VCC)    //设置电压 转12BIT ADC值
```

SDK文件框架及程序流程-程序流程

```c
/**
  * @brief  This function handles TIM6_IRQn.
  * @param  None
  * @retval None
  */
void TIM16_IRQHandler(void)
{
  extern volatile uint16_t MainCLK;
  extern void BLDC_1MS_IRQ(void);

  if(TIM16->SR & TIM_IT_Update)
  {
    MainCLK++;
    BLDC_1MS_IRQ();

    TIM16->SR = ~(TIM_IT_Update);
  }
}

/**
  * @brief  This function handles TIM1_CC_IRQn.
  * @param  None
  * @retval None
  */
void TIM1_CC_IRQHandler(void)      You, 3周前 • V1.1.0 …
{
  if(TIM1->SR & TIM_IT_CC4)
  {
    CmpLib_ForceCtrl();
  }

  TIM1->SR = ~(TIM_IT_CC4);
}

/**
  * @brief  This function handles TIM3_CC_IRQn.
  * @param  None
  * @retval None
  */
void TIM3_IRQHandler(void)
{
  extern void PwmCap_IRQ(void);

  PwmCap_IRQ();

  TIM3->SR=0;
}
```

```c
/**
  * @brief  1ms中断 服务处理。
  * @param  None
  * @retval None
  */
void BLDC_1MS_IRQ(void)
{
  Motor_CheckRestart();
  Motor_CheckVcc();
  Motor_CheckIbus();
  Motor_CheckPhase();
  Motor_CheckSpeed();

  Motor_LoopMode();
  Motor_StartStop();
  Motor_CtrlStatus();
}
```

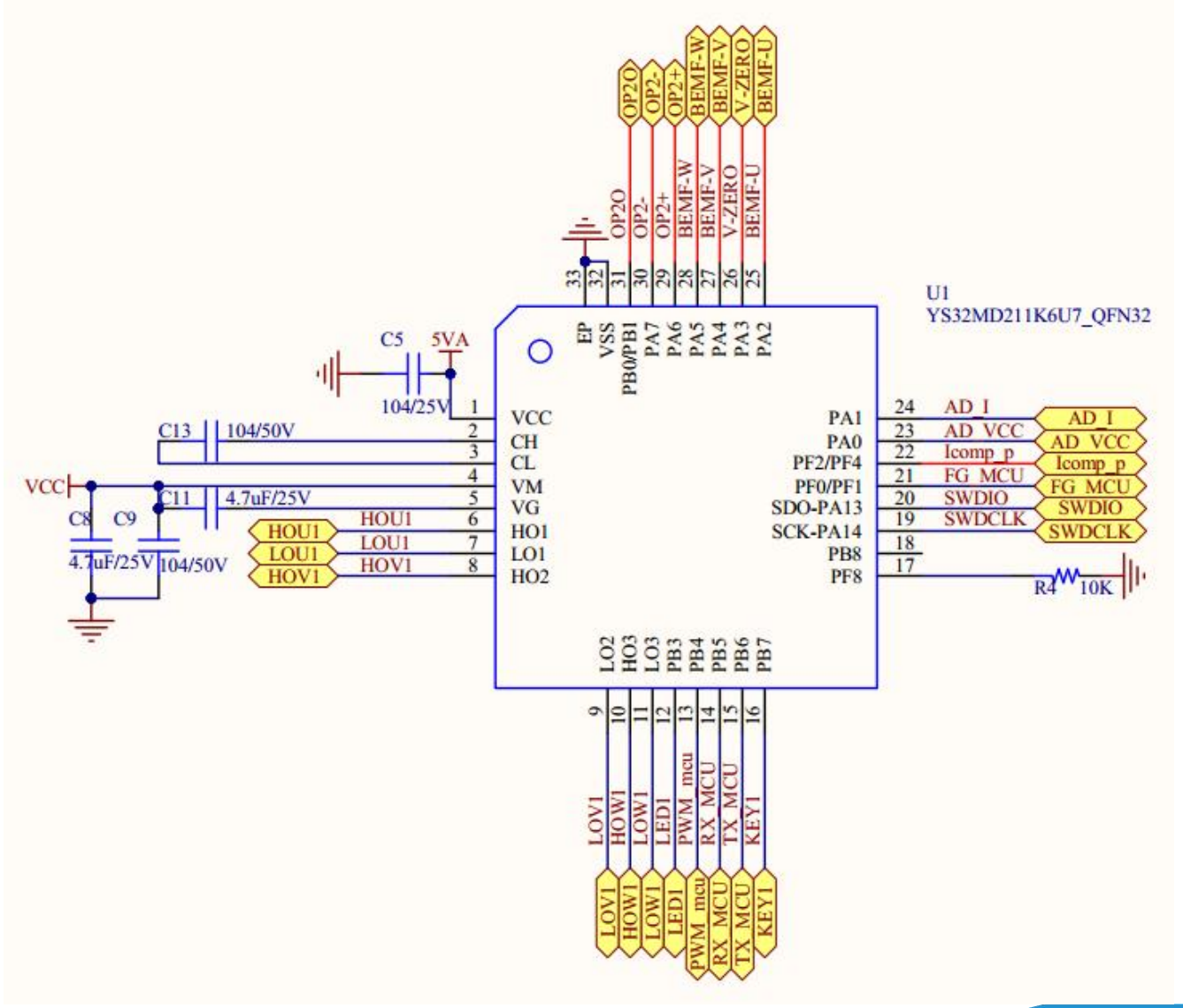
SDK文件框架及程序流程-程序流程

```
void Motor_CtrlStatus(void)
{
  switch(MotorCtrl.CtrlStatus)
  {
    case MOTOR_CTRL_MODE_INIT:
    {
      Motor_Ctrl_Init();
    }
    break;

    case MOTOR_CTRL_MODE_WIND:
    {
      Motor_Ctrl_Wind();
    }
    break;

    case MOTOR_CTRL_MODE_INJECT:
    {
      Motor_Ctrl_Inject();
    }
    break;

    case MOTOR_CTRL_MODE_PRESET:
    {
      Motor_Ctrl_Preset();
    }
    break;

    case MOTOR_CTRL_MODE_FORCE:
    {
      Motor_Ctrl_Force();
    }
    break;

    case MOTOR_CTRL_MODE_RUN:
    {
      Motor_Ctrl_Run();
    }break;

    case MOTOR_CTRL_MODE_STOP:
    {
      Motor_Ctrl_Stop();
    }break;

    default:  // MOTOR_CTRL_MODE_NOP
    {
    }break;
  }
}
```

SDK文件框架及程序流程-程序流程

```c
/**
  * @brief  开环/闭环模式
  * @param  None
  * @retval None
  */
static void Motor_LoopMode(void)
{
  uint16_t CurA;

  if(MotorCtrl.RunOpenFlag == 0)···

  MotorCtrl.SpeedValue = (MotorCtrl.SpeedValue+T60D_RPM(MotorCtrl.PhaseTime))/2;

  switch(MotorCtrl.RunMode)
  {
    case BLDC_RUN_FORCE:
    {

    }break;

    case BLDC_RUN_OPEN:···
    }break;

    case BLDC_RUN_LOOP_SPEED:···
    }break;

    case BLDC_RUN_LOOP_CUR:···
    }break;

    case BLDC_RUN_LOOP_PWR:
    {

    }break;
  }
}
```

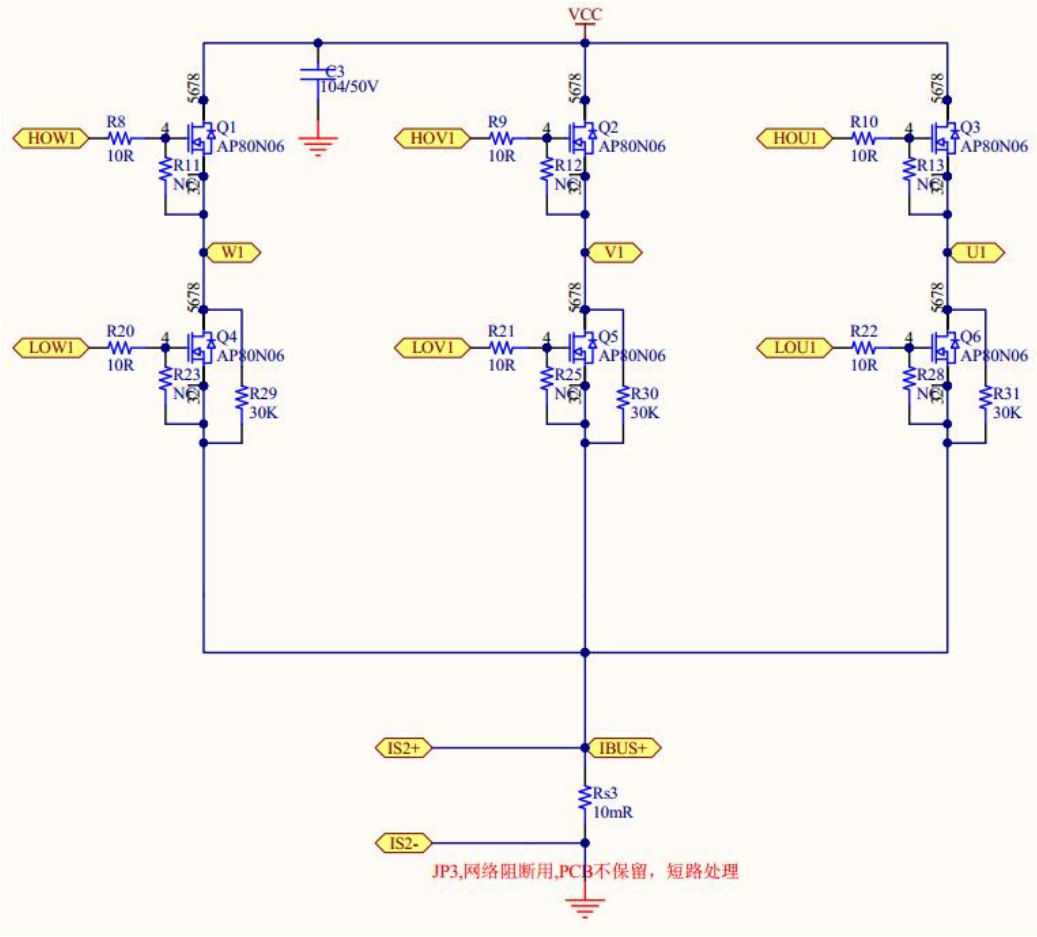
硬件原理图 – MCU主控


U1
YS32MD211K6U7_QFN32
OP2O
OP2-
OP2+
BEMF-W
BEMF-V
V-ZERO
BEMF-U
EP
VSS
PB0/PB1
PA7
PA6
PA5
PA4
PA3
PA2
VCC
CH
CL
VM
VG
HO1
LO1
HO2
PA1
PA0
PF2/PF4
PF0/PF1
SDO-PA13
SCK-PA14
PB8
PF8
LO2
HO3
LO3
PB3
PB4
PB5
PB6
PB7
AD_I
AD_VCC
Icomp_p
FG_MCU
SWDIO
SWDCLK
LOV1
HOW1
LOW1
LED1
PWM_mcu
RX_MCU
TX_MCU
KEY1
C5
5VA
104/25V
C13
104/50V
VCC
C11
4.7uF/25V
C8
C9
4.7uF/25V
104/50V
HOU1
LOU1
HOV1
R4
10K

硬件原理图 – MOS 驱动电路

如果PCB布线不便可把 U相PWM 和 W相PWM 互换

默认配置1UVW	可选配置2WVU
HOU1(CH1)	HOU1(CH3)
LOU1(CH1N)	LOU1(CH3N)
HOV1(CH2)	HOV1(CH2)
LOV1(CH2N)	LOV1(CH2N)
HOW1(CH3)	HOW1(CH1)
LOW1(CH3N)	LOW1(CH1N)



注意：

1、尽量使用默认UVW。只允许U和W互换。

互换后需修改配置

```
#define PWM_CH123_UVW_SYS  (PWM_CH123_WVU_SYS)
```

2、根据电流大小选择合适的MOS管。

3、根据电流大小选择合适的采样电阻（Rs3）和功率。

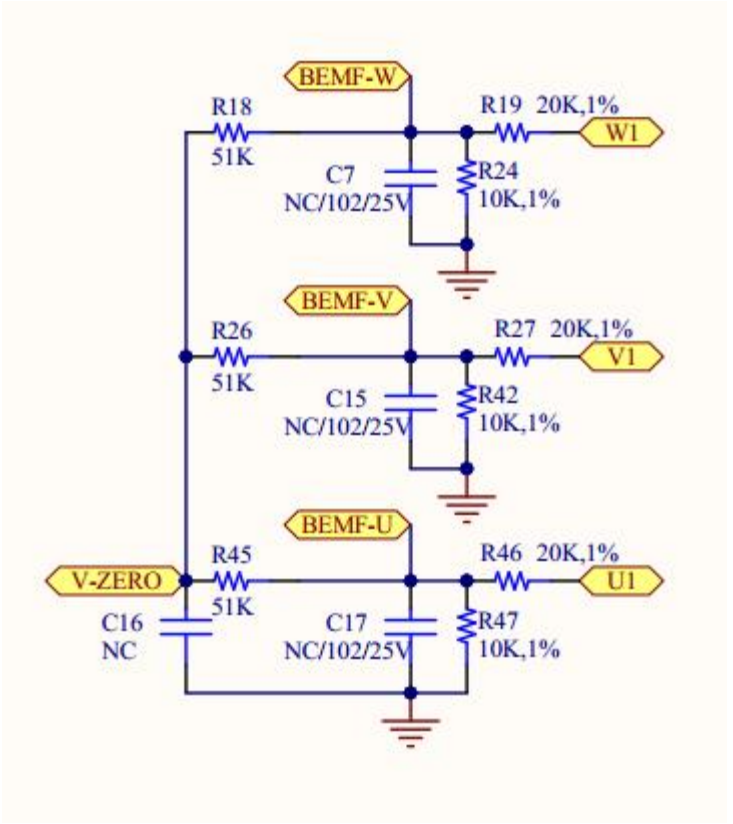
硬件原理图 –反电动势采样（比较器2）

V-ZERO（PA3）：虚拟中心点（COMP_INP）
BEMF-U： U相反电动势
BEMF-V： V相反电动势
BEMF-W： W相反电动势

BEMF-U, BEMF-V, BEMF-W 根据硬件配置选择，可选COMP_INM:
COMP2_InputMinus_PA2, COMP2_InputMinus_PA4, COMP2_InputMinus_PA5

```
//COMP UVW (COMP2_InputMinus_PA2, COMP2_InputMinus_PA4, COMP2_InputMinus_PA5)
#define CONFIG_COMP_PIN_U  (COMP2_InputMinus_PA2)          //COMP_INM_U 对应IO
#define CONFIG_COMP_PIN_V  (COMP2_InputMinus_PA4)          //COMP_INM_V 对应IO
#define CONFIG_COMP_PIN_W  (COMP2_InputMinus_PA5)          //COMP_INM_W 对应IO
```

反电动势UVW分压电阻选择(R19/R24, R27/R42,R46/R47):
1、最高转速分压后的反电动势电压不能超过5V，分压比尽量小。
2、分压比可根据以下公式选择。
分压比 = 最高工作电压 / 5V。
例如最高工作电压12V，12/5=2.4，分压比为3，可选R19/R27/R46=20K，R24/R42/R47=10K。



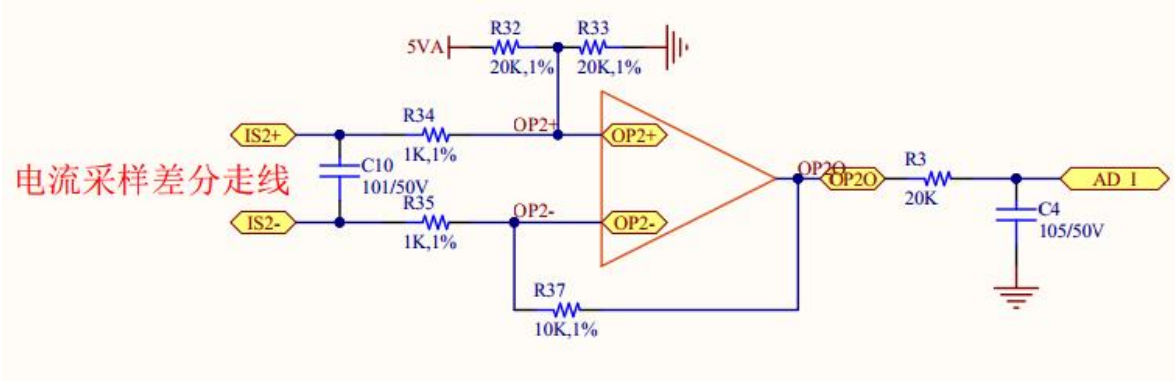
注意：尽量使用默认IO。

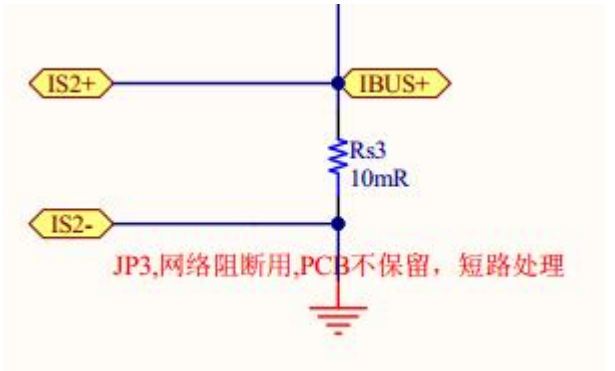
硬件原理图 – 电压/电流采样

2、相电流 --- 主要用于检测过流或电流环，采样电阻参数配置（Rs3）。相电流检测IO默认PB0, 详情见、BLDC_HardwareInit.c文件中的函数 ADC_Channel_Init()。AD_I(PA1) 暂未使用。

根据硬件修改以下参数：

CONFIG_HARD_MCU_VREF = 5000（mv）
CONFIG_HARD_IBUS_R = 0.01 (R)
CONFIG_HARD_IBUS_AMP = 10 (放大倍数)





注意：尽量使用默认IO。

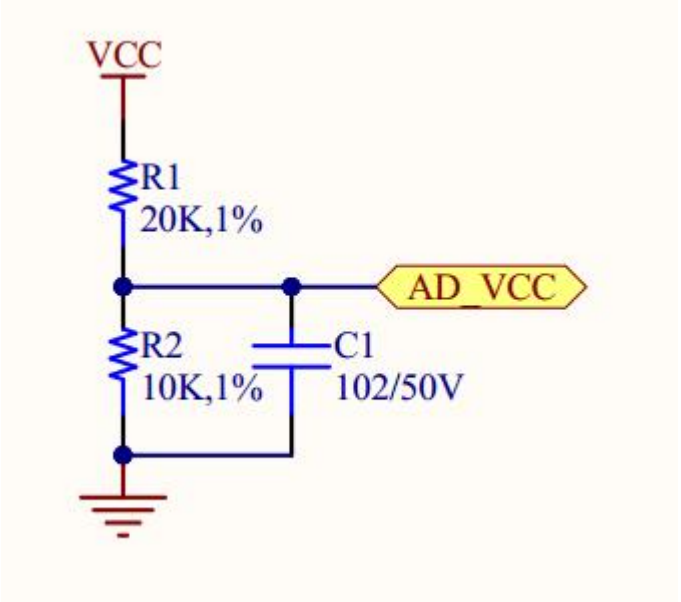
硬件原理图 – 电压/电流采样

SDK内部需要采样两路ADC，分别为：

1、工作电压 --- 主要用于低压报警，分压参数配置。

CONFIG_HARD_VCC_RATIO_R1 = 20

CONFIG_HARD_VCC_RATIO_R2 = 10

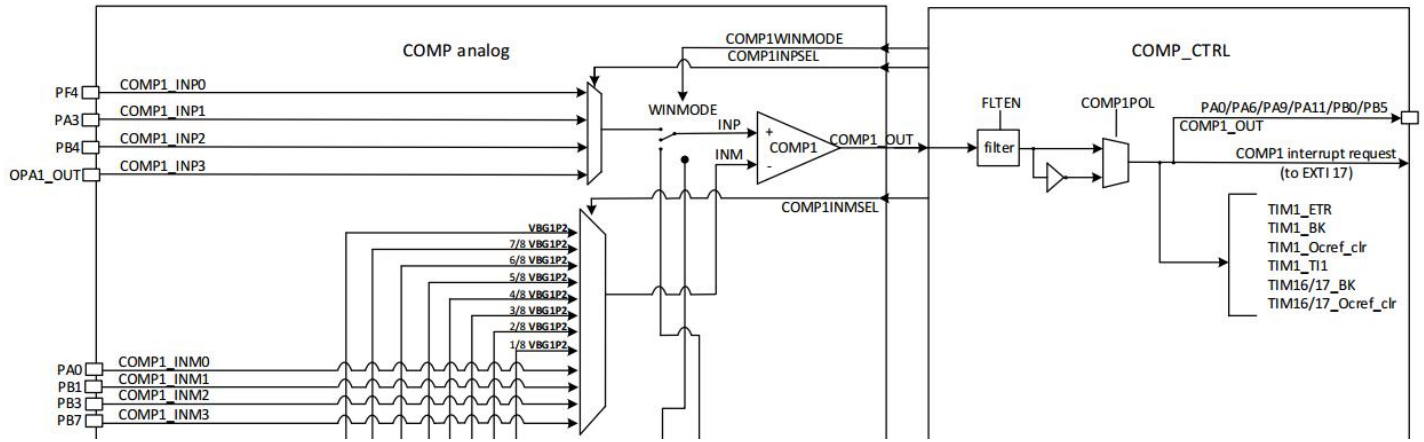


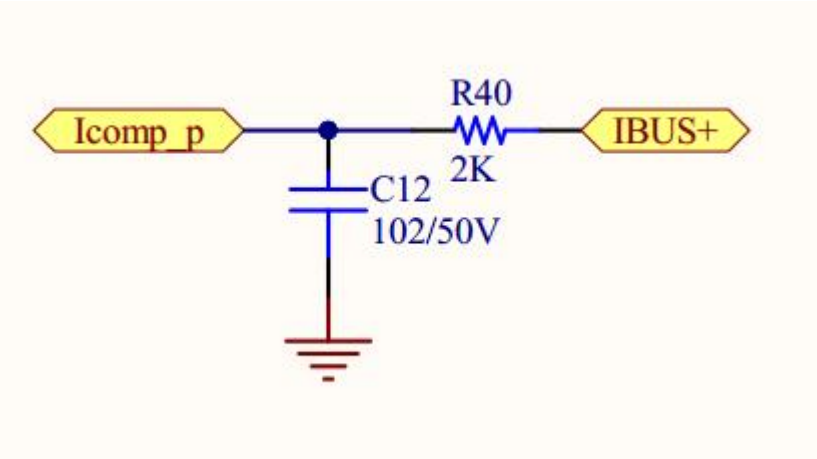
注意：尽量使用默认IO。

硬件原理图 –比较器1 硬件过流检测(预留)

比较器1硬件过流检测(预留)，COMP1_INP使用PF4， COMP1_INM使用芯片内部1.2V的1/8 ~ 8/8，详情查看芯片规格书。

12.3.1 比较器框图





软件SDK – 配置参数

BLDC_ConfigParameter.h文件可配置参数如下：

1、调试参数。 *调试开关同时只能开一个!* 详情见对应参数注释。

2、电机参数 。电机极对数必须正确！

3、硬件功能参数 。 Vcc分压电阻、 ADC基准电压、母线采样电阻、运放放大倍数、硬件比较器滤波时间、 MOS 死区时间、限制PWM最大/最小值（速度环/电流环会用到）、用户ADC采样配置、PWM IN 调速、速度反馈FG_OUT 。详情见对应参数注释。

4、启动参数（调试重点） 。详情见对应参数注释。

5、保护功能参数。 详情见对应参数注释。

软件SDK – 硬件功能(PWM IN 调速)

PWM IN 调速：

1、PWM IN 调速功能实现在“BLDC_AppPwmIN.c”。CONFIG_PWM_CTRL_ENABLE = 1 则启用PWM 调速。

2、用户可配置 启动/停止 电机占空比（xx%）。

3、PWM IN 输入捕获 IO 配置见文件“BLDC_AppPwmIN.c”函数 TIM3_PwmCap_Init（）。

```
#define CONFIG_PWM_CTRL_ENABLE      (1)
```

```
#define CONFIG_PWM_START   25     // x% 启动
#define CONFIG_PWM_STOP    20     // x% 停止
```

```
/**
  * @brief  TIM3 捕获PWM初始化
  * @param  None
  * @retval None
  */
void TIM3_PwmCap_Init(void)      You, 3周前 • V1.1.0 …
{
  GPIO_InitTypeDef GPIO_InitStructure;
  TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;
  NVIC_InitTypeDef NVIC_InitStructure;
  TIM_ICInitTypeDef  TIM_ICInitStructure;

  //========================================
  RCC_AHB2PeriphClockCmd(RCC_AHB2Periph_GPIOB, ENABLE);
  RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM3 , ENABLE);

  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_4;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_VeryHigh;
  GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
  GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_DOWN;
  GPIO_Init(GPIOB, &GPIO_InitStructure);

  GPIO_PinAFConfig(GPIOB, GPIO_PinSource4, GPIO_AF_1);
```

软件SDK – 硬件功能(速度反馈FG_OUT)

速度反馈FG_OUT:

1、 CONFIG_FG_OUT_ENABLE= 1 启用速度反馈。

2、修改 IO 端口配置，打开“BLDC_HardwareInit.h”，修改 FG_OUT_GPIO_RCC，FG_OUT_GPIO_PORT，FG_OUT_PIN。

```
//=============================================
// FG_OUT
//=============================================
#define FG_OUT_GPIO_RCC    RCC_AHB2Periph_GPIOF
#define FG_OUT_GPIO_PORT   GPIOF
#define FG_OUT_PIN         GPIO_Pin_1
```

3、CONFIG_FG_OUT_DIV 配置除频参数，使用默认参数(极对数正确)，当前速度等于FG_OUT频率（HZ）* 60。

软件SDK – 硬件功能(用户自定义ADC配置)

用户自定义ADC通道操作如下：

1、CONFIG_USER_ADC_ENABLE 置1。

2、USER_ADC_CH_SIZE 填写通道数，该配置保存在“BLDC_HardwareInit.h ” 文件中。数组ADC_UserData[]为ADC采样数据，如2个ADC通道时，通道1 为ADC_UserData[0] ， 通道2 为 ADC_UserData[1] 。

3、修改User_ADC_Channel_Init()函数，增加/修改ADC通道初始化 ， 该配置保存在 “BLDC_HardwareInit.c ” 文件中。

4、按照以上步骤修改后，DMA CH1 自动把转换的数据搬到数组ADC_UserData[] ， 用户直接用即可。

```
#define CONFIG_USER_ADC_ENABLE        (1)
```

```
//ADC ========================================================================
#define USER_ADC_CH_SIZE  2
extern volatile uint16_t ADC_UserData[USER_ADC_CH_SIZE];
void ADC_Channel_Init(void);
```

```
static void User_ADC_Channel_Init(void)
{
  DMA_InitTypeDef DMA_InitStructure;
  GPIO_InitTypeDef GPIO_InitStructure;
  NVIC_InitTypeDef NVIC_InitStructure;

  GPIO_StructInit(&GPIO_InitStructure);
  DMA_StructInit(&DMA_InitStructure);

  RCC_AHBPeriphClockCmd(RCC_AHBPeriph_DMA, ENABLE);
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_SYSCFG, ENABLE);
  RCC_AHB2PeriphClockCmd(RCC_AHB2Periph_GPIOA, ENABLE);
  RCC_AHB2PeriphClockCmd(RCC_AHB2Periph_GPIOB, ENABLE);

  //=====================================================
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_12 | GPIO_Pin_13;  // ADC_Channel_15, ADC_Channel_16
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AN;
  GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
  GPIO_Init(GPIOB, &GPIO_InitStructure);

  //=====================================================
  // ADC_TempSensorCmd(ENABLE);  //使能温度传感器
  // VREFBUF_Cmd(ENABLE);        //使能vrefbuf
  ADC_RegularSequencerLengthConfig(ADC, USER_ADC_CH_SIZE);
  ADC_RegularChannelConfig(ADC, ADC_Channel_15, 1,  ADC_SampleTime_63_5Cycles);  // ADC_UserData[0]
  ADC_RegularChannelConfig(ADC, ADC_Channel_16, 2,  ADC_SampleTime_63_5Cycles);  // ADC_UserData[1]
```

软件SDK – 配置参数（*调试开关参数*)

```
//==========================================================================
// 调试参数, 调试开关同时只能开一个!!!
#define CONFIG_DUBUG_AUTO_RUN_TEST   (0)                    //电机自动测试(启动/停止)调试开关.=0 正常工作, =1 电机自动 启动/停止 测试,
#define CONFIG_DUBUG_IPD_TEST        (0)                    //IPD注入调试开关.0-关闭,1-打开IPD注入调试. 注意: 1. 硬件有运放放大电流. 2.
```

注意：调试开关同时只能启用一个。

CONFIG_DUBUG_AUTO_RUN_TEST：电机自动测试(启动/停止)调试开关.
=0 正常工作, =1 电机自动 启动/停止 测试。

CONFIG_DUBUG_IPD_TEST：IPD注入调试开关.
0-关闭,1-打开IPD注入调试.

使用注意事项：
1. 硬件有运放放大电流。
2. 脉冲注入定位使能（BLDC_IPD_ENABLE = 1）。
3. 初始化串口IO。

软件SDK – 配置参数（*电机参数*）

```
//=============================================================================
// 电机参数

#define USER_MOTOR_POLES            (6)             //电机转子磁极对数
```

根据电机配置极对数（重要），此处电机为6对极。

软件SDK – 配置参数（*硬件参数*）

```
//=============================================================================
// 硬件参数

#define MCU_SYSTEM_CLOCK            (48000000)                          //Hz
#define CONFIG_PWM_CLOCK            (20000)  // (40000)                          //Hz

#define CONFIG_PHASE_TIM_PRESCALER  (48)                                //分频

#define CONFIG_COMP_FILTER_TIME     (1)  //(6)                         //us,硬件比较器滤波时间
#define CONFIG_PWM_DEAD_TIME        (0.55)                        //us,MOS 死区时间
#define CONFIG_CC4_TIME             (1.5)                         //us,CC4 时间

#define CONFIG_PWM_CH123_UVW        (PWM_CH123_UVW_SYS)         //CH1/2/3 对应 UVW 选择(T_PWM_CH123).PWM_CH123_UVW_SYS, PWM_CH123_WVU_SYS, PWM_CH123_UVW_IO_3P3N

//COMP UVW (COMP2_InputMinus_PA2, COMP2_InputMinus_PA4, COMP2_InputMinus_PA5)
#define CONFIG_COMP_PIN_U  (COMP2_InputMinus_PA2)          //COMP_INM_U 对应IO
#define CONFIG_COMP_PIN_V  (COMP2_InputMinus_PA4)          //COMP_INM_V 对应IO
#define CONFIG_COMP_PIN_W  (COMP2_InputMinus_PA5)          //COMP_INM_W 对应IO

//
#define CONFIG_USE_VREF_1D2V        (0)             // =1, 使用MCU内部1.2V作为ADC基准电压。=0, 使用VCC作为ADC基准电压。
#define CONFIG_HARD_MCU_VREF        (5000)          // mV, ADC基准电压。 CONFIG_USE_VREF_1D2V=1 必须为 1200mV。

#define CONFIG_HARD_VCC_RATIO_R1    (20)            // K, Vcc分压电阻R1(接 VCC)
#define CONFIG_HARD_VCC_RATIO_R2    (10)            // K, Vcc分压电阻R2(接 GND)
#define CONFIG_HARD_IBUS_R          (0.01)          // R, 母线采样电阻
#define CONFIG_HARD_IBUS_AMP        (10)            // 母线采样 运放放大倍数

#define CONFIG_PWM_CTRL_ENABLE      (0)                                 //PWM调速(功能实现:BLDC_AppPwmIN.c). 0-关闭,1-打开.

#define CONFIG_FG_OUT_ENABLE        (0)                                 //速度反馈(FG_OUT). 0-关闭,1-打开.
#define CONFIG_FG_OUT_DIV           ((USER_MOTOR_POLES/2)*6)            //速度反馈除频, 单位电周期60度

#define CONFIG_USER_ADC_ENABLE      (0)                                 //用户ADC(DMA-CH1)采样配置.0-关闭,1-打开. 详情见 "BLDC_HardwareInit.h"

#define CONFIG_PWM_DUTY_MAX         (s16Q12_PWM(1))             //限制PWM最大占空比
#define CONFIG_PWM_DUTY_MIN         (s16Q12_PWM(0.10))          //限制PWM最小占空比
```

软件SDK – 配置参数（*启动参数*）

```
//==========================================================================
// 启动参数
#define CONFIG_MOTOR_RUN_DIR        (MOTOR_RUN_FORWARD)                    //马达方向(正/反). MOTOR_RUN_FORWARD 正转, MOTOR_RUN_BACKWARD 反转.

#define BLDC_ADVANCE_ANGLE          (18)     //超前角(1~30度)
#define BLDC_CHECK_OPEN_SET         (2)      //延时打开过零检测
#define BLDC_PHASE_TIME_OUT         (10000)  //换相超时(us)

#define BLDC_FORCE_PWM_MIN          (s16Q12_PWM(0.15))                     //强拖初始 PWM MIN
#define BLDC_FORCE_PWM_MAX          (BLDC_FORCE_PWM_MIN+s16Q12_PWM(0.05))  //强拖初始 PWM MAX

#define BLDC_FORCE_TIME_START       (RPM_60D(100))   //强拖开始 转速(1) 或 时间(2). 1: 使用转速时 RPM_60D(150) 转速START.              2: 使用时间时,START时间(us)为(13333).
#define BLDC_FORCE_TIME_END         (RPM_60D(500))   //强拖结束 转速(1) 或 时间(2). 1: 使用转速时 RPM_60D(560) 转速END, END要比START大.  2: 使用时间时,  END时间(us)为(3571), END要比START小.
#define BLDC_FORCE_SPEED_ADD        (3)      //加速周期, 越大越慢
#define BLDC_FORCE_SPEED_RAMP       (1)      //爬坡参数

#define BLDC_FORCE_START_STEP       (3)      // 延迟步数开始过零检测
#define BLDC_FORCE_ZERO_STEP        (1)      // 强拖过零点次数切闭环      You, 3周前 • V1.1.0 …
// #define BLDC_WIND_ENABLE            (0)     //顺风启动(暂不支持).0-关闭,1-打开.
#define BLDC_IPD_ENABLE             (0)      //脉冲注入定位(外转子电机可用).0-关闭,1-打开.
#define BLDC_IPD_ON_TIME            (30)     //注入打开时间(*1us)
#define BLDC_IPD_OFF_TIME           (200)    //注入关闭时间(*1us)

#define BLDC_PRESET_ENABLE          (1)      //预定位.0-关闭,1-打开.
#define BLDC_PRESET_TIME_SET        (30)     //预定位总时间(*1ms) = BLDC_PRESET_TIME_SET * 1MS.
#define BLDC_PRESET_PWM_DUTY        (BLDC_FORCE_PWM_MIN)     //预定位 PWM Duty

#define BLDC_RUN_MODE               (BLDC_RUN_LOOP_SPEED)    // 运行模式(T_BLDC_RUN_MODE): BLDC_RUN_FORCE(强拖), BLDC_RUN_OPEN(开环),
                                                             // BLDC_RUN_LOOP_SPEED(速度闭环),BLDC_RUN_LOOP_CUR(电流闭环),BLDC_RUN_LOOP_PWR(功率闭环),

#define BLDC_PID_CYCLE_TIME             (10)             // 周期调整PWM时间(ms), 开环、速度/电流/功率闭环PID。

//OPEN - CONFIG
#define BLDC_RUN_OPEN_PWM_SET           (s16Q12_PWM(1.0))  // 电压环默认 PWM
#define BLDC_RUN_OPEN_PWM_RAMP          (10)               // 电压环PWM爬坡参数

//SPEED LOOP - CONFIG
#define CONFIG_BLDC_SPEED_RUN           (7200)             // 速度环默认速度7200, 注意USER_MOTOR_POLES必须正确
#define CONFIG_BLDC_SPEED_RAMP          (80)               // 速度环爬坡参数(CONFIG_BLDC_SPEED_RUN)
#define CONFIG_BLDC_SPEED_PI_KP         s16Q14(0.03)
#define CONFIG_BLDC_SPEED_PI_KI         s16Q14(0.003)

//CUR LOOP - CONFIG
#define CONFIG_BLDC_CUR_A_RUN           (4.0)              // 电流环默认电流(A)
#define CONFIG_BLDC_CUR_A_RAMP          (0.2)              // 电流环爬坡参数(A)
#define CONFIG_BLDC_CUR_A_PI_KP         s16Q14(0.08)
#define CONFIG_BLDC_CUR_A_PI_KI         s16Q14(0.006)
```

软件SDK – 配置参数（*启动参数*）

1、脉冲注入(BLDC_IPD_ENABLE)，需要有运放放大电流才能使用，根据实际情况选择。

2、预定位(BLDC_PRESET_ENABLE)，根据实际情况选择。

3、最重要启动参数：
BLDC_FORCE_PWM_MIN 强拖PWM 最小值。
BLDC_FORCE_PWM_MAX 强拖PWM 最大值。

BLDC_FORCE_TIME_START 强拖开始转速(时间) 。
BLDC_FORCE_TIME_END 强拖结束(时间) 。
强拖时间可用时间（us）或 转/分钟表示。可使用该宏把转速转为时间 RPM_60D（560），560 转/分钟。

BLDC_FORCE_SPEED_ADD 加速周期, 越大越慢。

4、运行模式选择(BLDC_RUN_MODE)：
BLDC_RUN_FORCE(强拖),
BLDC_RUN_OPEN(开环),
BLDC_RUN_LOOP_SPEED(速度闭环),
BLDC_RUN_LOOP_CUR(电流闭环),

软件SDK – 配置参数（*保护功能参数*）

```
//==========================================================================
// 保护功能参数
#define USER_MOTOR_FORCE_RESTART_ENABLE     (1)                 //强拖、运行异常重试(过流,电压保护不重试). 0-关闭, 1-打开
#define USER_MOTOR_FORCE_RESTART_CNT        (3)                 //强拖、运行异常重试次数(1~65535), 0-为重试无限次

#define USER_MOTOR_BRAKE_ENABLE             (0)                 //电机快速刹车(快速停止).0-关闭,1-打开. 注意:快速停止, 3个下臂NMOS电流大, 请按照实际情况选择MOS.
#define USER_MOTOR_BRAKE_DELAY              (10)                //电机停止后延时刹车(ms)

#define USER_MOTOR_IBUS_ENABLE              (1)                 //相电流保护. 0-关闭, 1-打开
#define USER_MOTOR_IBUS_A                   (6.0)               // A, 母线最大电流
#define USER_MOTOR_IBUS_MAX_CNT             (500)               //达到最大电流计数值(*1ms)

#define USER_MOTOR_PHASE_ERROR_ENABLE       (1)                 //堵转相位异常(堵转,缺相). 0-关闭,1-打开.
#define USER_MOTOR_PHASE_ERROR_CNT          (24)                //堵转相位异常步数

#define USER_MOTOR_RPM_ERR_ENABLE           (1)                 //转速异常检测. 0-关闭,1-打开
#define USER_MOTOR_RPM_ERR_CNT              (500)               //转速异常计数(*1ms)
#define USER_MOTOR_MAX_RPM_TIME             (15000)             //正常运行的最高转速

#define USER_MOTOR_VCC_PROTECT_ENABLE       (0)                 //过压, 低压保护. 0-关闭, 1-打开
#define USER_MOTOR_VCC_PROTECT_TIME         (1000)              //保护延时(ms)
#define USER_MOTOR_VCC_HIGH_PROTECT         (9000)              //高压-保护电压(mv)
#define USER_MOTOR_VCC_HIGH_RECOVER         (USER_MOTOR_VCC_HIGH_PROTECT-2000)          //高压-保护后恢复电压(mv)
#define USER_MOTOR_VCC_LOW_RECOVER          (USER_MOTOR_VCC_LOW_PROTECT +2000)          //低压-保护后恢复电压(mv)
#define USER_MOTOR_VCC_LOW_PROTECT          (6200)              //低压-保护电压(mv)
```

软件SDK – 配置参数（*保护功能参数*）

1、保护电流(USER_MOTOR_IBUS_ENABLE)。
参数USER_MOTOR_IBUS_A 设置母线电流(A)，参数USER_MOTOR_IBUS_MAX_CNT 设置计数时间(ms)。

2、堵转相位异常(USER_MOTOR_PHASE_ERROR_ENABLE)。

3、转速保护(USER_MOTOR_RPM_ERR_ENABLE)。
参数 USER_MOTOR_MAX_RPM_TIME 设置最高转速。如电机速度7200转/分钟，参数USER_MOTOR_MAX_RPM_TIME 设置为20000，电机转速高于20000会触发保护。

4、电机快速刹车(USER_MOTOR_BRAKE_ENABLE)。

5、过压，低压保护(USER_MOTOR_VCC_PROTECT_ENABLE)。

电机异常停机，Debug可查看错误码 **MotorCtrl.ErrorCode** 。

软件SDK – IPD注入调试

脉冲注入定位，外转子电机可用。内转子电机根据实测试效果选择是否启用。

IPD注入调试步骤如下：

1、BLDC_IPD_ENABLE = 1，
CONFIG_DUBUG_IPD_TEST = 1，
配置串口驱动*UART1_Configuration*()。

2、用USB转串口工具连 串口TX引脚。打开上位机调试软件TouchTest1.3.2.exe，选择对应端口，波特率2M。
点击开始，打印检测相位0~5。

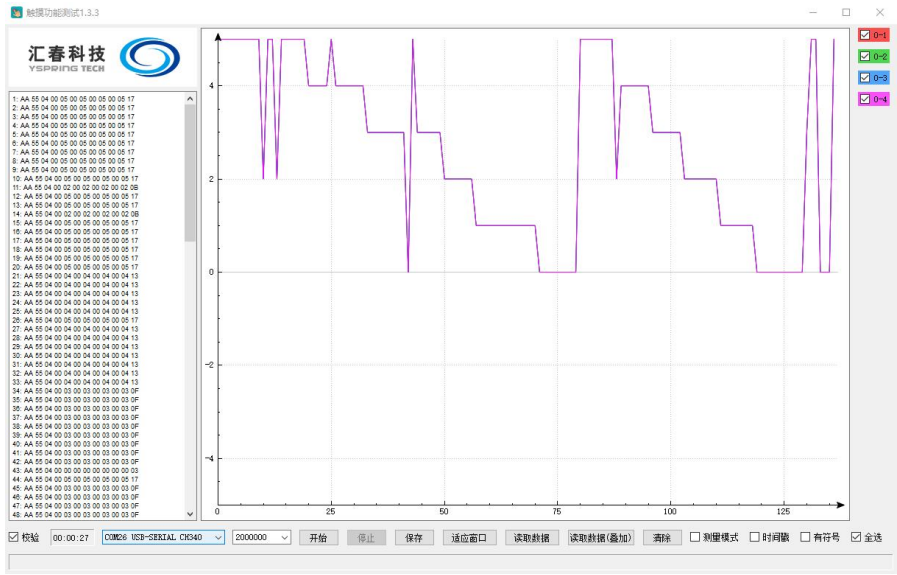
3、调试修改BLDC_IPD_ON_TIME、
BLDC_IPD_OFF_TIME 参数，微转动电机直到能检测到相位并且电机不能转动。

4、参数调试完成后，关闭IPD调试
(CONFIG_DUBUG_IPD_TEST = 0)。

TouchTest1.3.2.exe下载地址：https://yspringtech.com/uploads/file/TouchTest1.3.2.zip

```
#define CONFIG_DUBUG_IPD_TEST        (1)

#define BLDC_IPD_ENABLE              (1)
#define BLDC_IPD_ON_TIME             (10)
#define BLDC_IPD_OFF_TIME            (100)
```

```
int main(void)
{
  RCC_Configuration();
  GPIO_Configuration();

  TIM16_Configuration();

  UART1_Configuration();  // Debug_DataLine(), Test IPD

  BLDC_Motor_Init();

  #if ((CONFIG_PWM_CTRL_ENABLE == 1) & (BLDC_RUN_MODE == BLDC_RUN_LOOP_VOL))
  TIM3_PwmCap_Configuration();  // PWM Rev Init
  #endif

  LEDx_Init();
  KEYx_Init();
```



软件SDK – 预定位

```
#define BLDC_PRESET_ENABLE          (1)      // 预定位.0-关闭,1-打开.
#define BLDC_PRESET_TIME_SET        (30)     // 预定位总时间(*1ms) = BLDC_PRESET_TIME_SET * 1MS.
#define BLDC_PRESET_PWM_DUTY        (s16Q12_PWM(0.15))     // 预定位 PWM Duty
```

使用注意事项：

根据电机启动速度，选择合适的预定位时间和占空比，调整参数使电机启动顺畅，没有卡顿感。

SDK调试步骤实践

1、使用强拖模式，测试硬件是否正常。用示波器测试UVW相波形，调试启动参数。

#define CONFIG_DUBUG_AUTO_RUN_TEST **(1)**
#define BLDC_RUN_MODE **(**BLDC_RUN_FORCE**)**

BLDC_FORCE_PWM_MIN 根据负载选择，电机能正常启动，不能出现失步振动。

```
#define BLDC_FORCE_PWM_MIN          (s16Q12_PWM(0.15))                          //强拖初始 PWM MIN
#define BLDC_FORCE_PWM_MAX          (BLDC_FORCE_PWM_MIN+s16Q12_PWM(0.05))       //强拖初始 PWM MAX

#define BLDC_FORCE_TIME_START       (RPM_60D(100))    //强拖开始 转速(1) 或 时间(2)。1: 使用转速时 R
#define BLDC_FORCE_TIME_END         (RPM_60D(500))    //强拖结束 转速(1) 或 时间(2)。1: 使用转速时 R
#define BLDC_FORCE_SPEED_ADD        (3)       //加速周期，越大越慢
#define BLDC_FORCE_SPEED_RAMP       (1)       //爬坡参数
```

SDK调试步骤实践

用示波器测量 UVW某一相波形和 调试取反IO。如下图，调试取反IO 有3个以上连续信号 且 强拖波形较好。

```
C BLDC_Motor.c ×
BLDC > C BLDC_Motor.c > CmpLib_UVW_IRQ(void)
366  /**
367   * @brief   相位检测中断
368   * @param   None
369   * @retval  None
370   */
371  void CmpLib_UVW_IRQ(void)        You, 3周前 • V1.1.0 …
372  {
373      TEST1_CPL();
374  }
375
```

SDK调试步骤实践

2、调试优化启动参数（调频、调频调压），运行模式选择开环，启动电流保护功能。

#define BLDC_RUN_MODE (BLDC_RUN_OPEN)

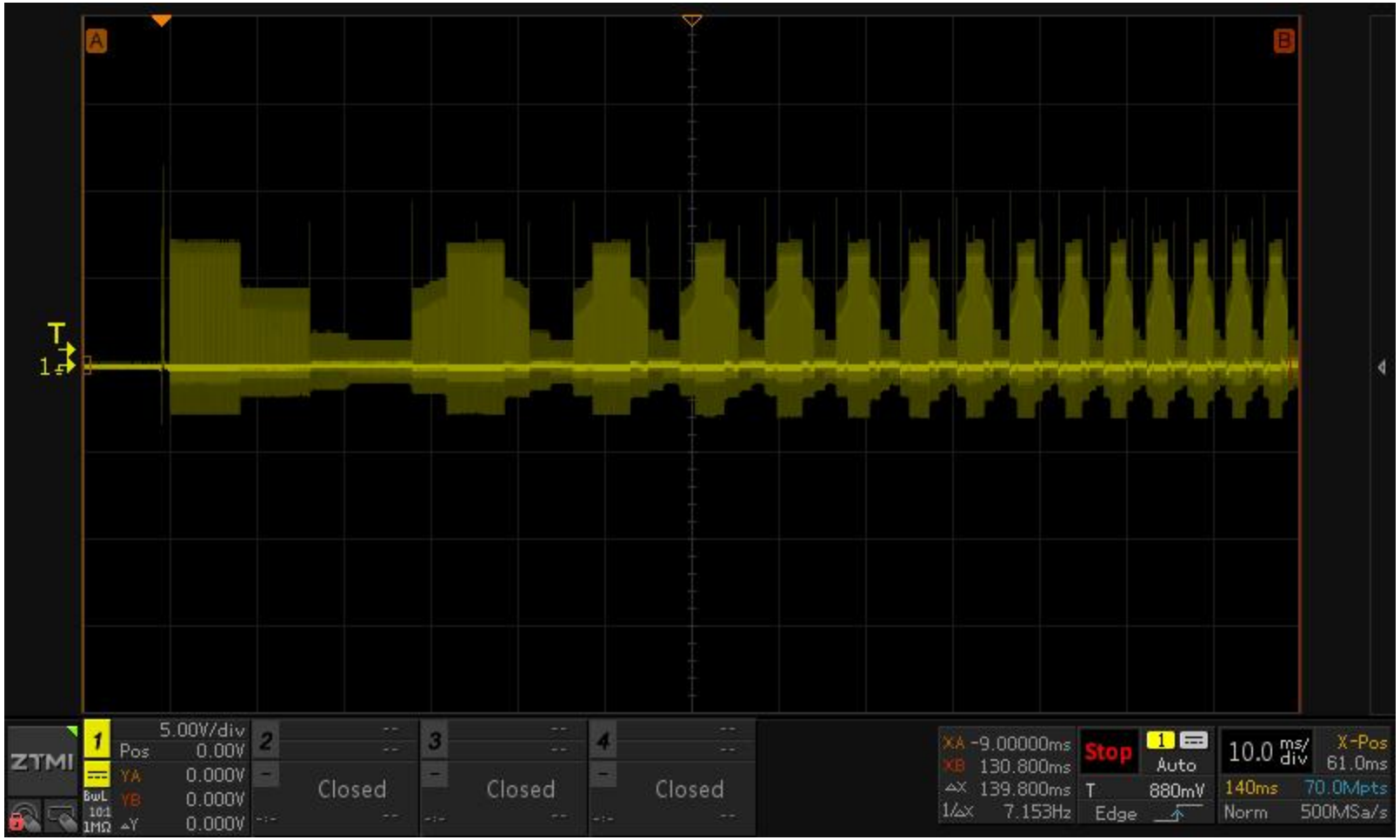
调频方式：
BLDC_FORCE_PWM_MIN、 BLDC_FORCE_PWM_MAX 相同，同时A速度强拖B速度。

调频调压：
PWM从BLDC_FORCE_PWM_MIN 加速到 BLDC_FORCE_PWM_MAX，同时A速度强拖B速度。

根据负载情况选择合理的PWM占空比，强拖速度(BLDC_FORCE_TIME_START、BLDC_FORCE_TIME_END、BLDC_FORCE_SPEED_ADD)。

根据转速选择合适PWM频率（CONFIG_PWM_CLOCK）。
电周期小于8万转，选择20K或更小。电周期大于8万转，选择20K~50K。

SDK调试步骤实践


A
B
T
1
ZTMI
1
5.00V/div
Pos 0.00V
YA 0.000V
YB 0.000V
-Y 0.000V
2
Closed
3
Closed
4
Closed
XA -9.00000ms
XB 130.800ms
ΔX 139.800ms
1/ΔX 7.153Hz
Stop
1
Auto
T 880mV
Edge
10.0 ms/div
X-Pos 61.0ms
140ms 70.0Mpts
Norm 500MSa/s

SDK调试步骤实践

3、分别在最高、最低工作电压测试启动正常。关闭调试开关（调试参数**），根据需要求选择运行模式（**BLDC_RUN_MODE**），配置默认参数。**

A、开环（BLDC_RUN_OPEN）配置参数。

```
//OPEN - CONFIG
#define BLDC_RUN_OPEN_PWM_SET           (s16Q12_PWM(1.0))  // 电压环默认 PWM
#define BLDC_RUN_OPEN_PWM_RAMP          (10)               // 电压环PWM爬坡参数
```

B、速度闭环（BLDC_RUN_LOOP_SPEED）配置参数。

```
//SPEED LOOP - CONFIG
#define CONFIG_BLDC_SPEED_RUN           (7200)             // 速度环默认速度7200，注意USER_MOTOR_POLES必须正确
#define CONFIG_BLDC_SPEED_RAMP          (80)               // 速度环爬坡参数(CONFIG_BLDC_SPEED_RUN)
#define CONFIG_BLDC_SPEED_PI_KP         s16Q14(0.03)
#define CONFIG_BLDC_SPEED_PI_KI         s16Q14(0.003)
```

C、电流闭环（BLDC_RUN_LOOP_CUR）配置参数。

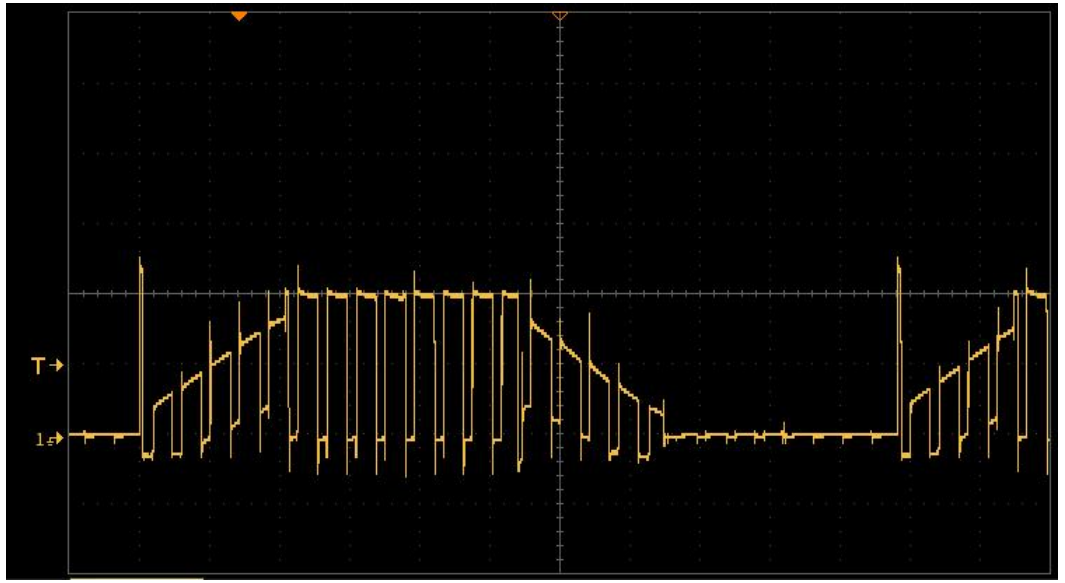
```
//CUR LOOP - CONFIG
#define CONFIG_BLDC_CUR_A_RUN           (4.0)              // 电流环默认电流(A)
#define CONFIG_BLDC_CUR_A_RAMP          (0.2)              // 电流环爬坡参数(A)
#define CONFIG_BLDC_CUR_A_PI_KP         s16Q14(0.08)
#define CONFIG_BLDC_CUR_A_PI_KI         s16Q14(0.006)
```

SDK调试步骤实践

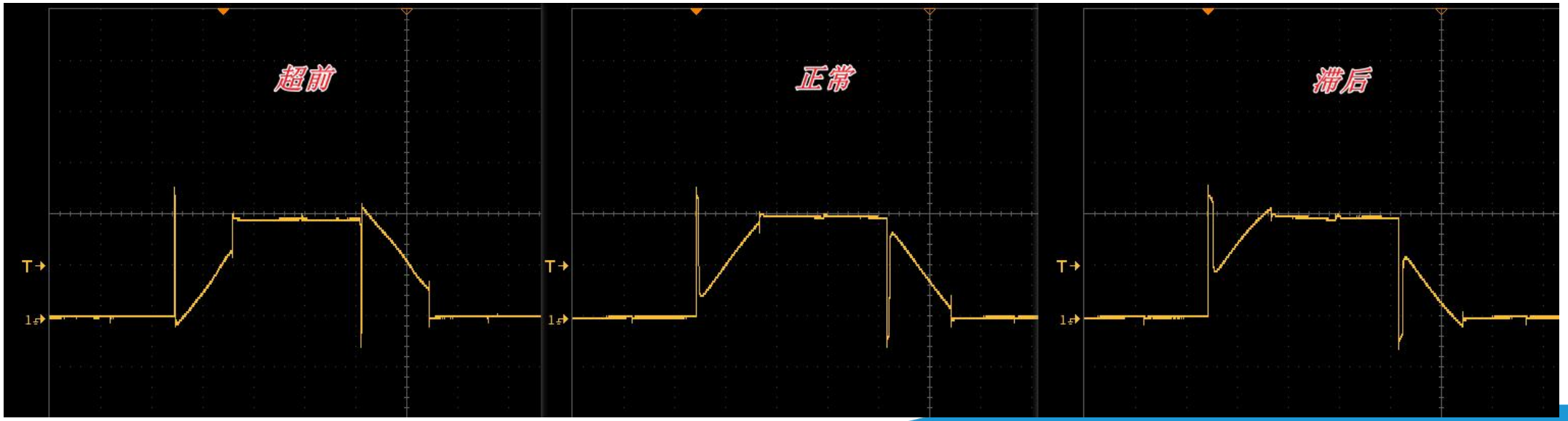
4、超前角、屏蔽续流参数。

BLDC_ADVANCE_ANGLE：参考图片2
超前：超前角增加
滞后：超前角减小

BLDC_CHECK_OPEN_SET：参考图片1
观察右图二极管续流用多少个PWM周期



图片1



图片2

SDK调试步骤实践

5、调试常见问题分析。

5.1、电机启动一会就停止。查看错误状态位(**MotorCtrl.ErrorCode**)，解决方法：是否触发过流保护、转速异常保护、检查比较器过零检测电路。

5.2、电机能转动相波形不正常。解决方法：检测MOS管、MOS 栅极6个电阻是否有虚焊。

5.3、使用速度环时，速度达不到设定转速。解决方法1，调整PI参数。解决方法2，速度补偿。

5.4、中断函数优先级别：用户新增中断函数优先级别不能为0，因为中断优级别0 SDK使用。

5.5、YS32F031 BLDC_SDK库包含有YS32MD210 的 Driver 驱动文件 这是怎么回事?
YS32MD210(N+N 外置电荷泵)、YS32MD211(N+N)、YS32MD220 (P+N) 、YS32F031 MCU内核是一样的，只是外挂预驱动不一样。YS32MD210 外设功能最全，选择YS32MD210 的 Driver 驱动文件完成兼容其它型号。

扎根春天向阳生长

用芯共创美好未来