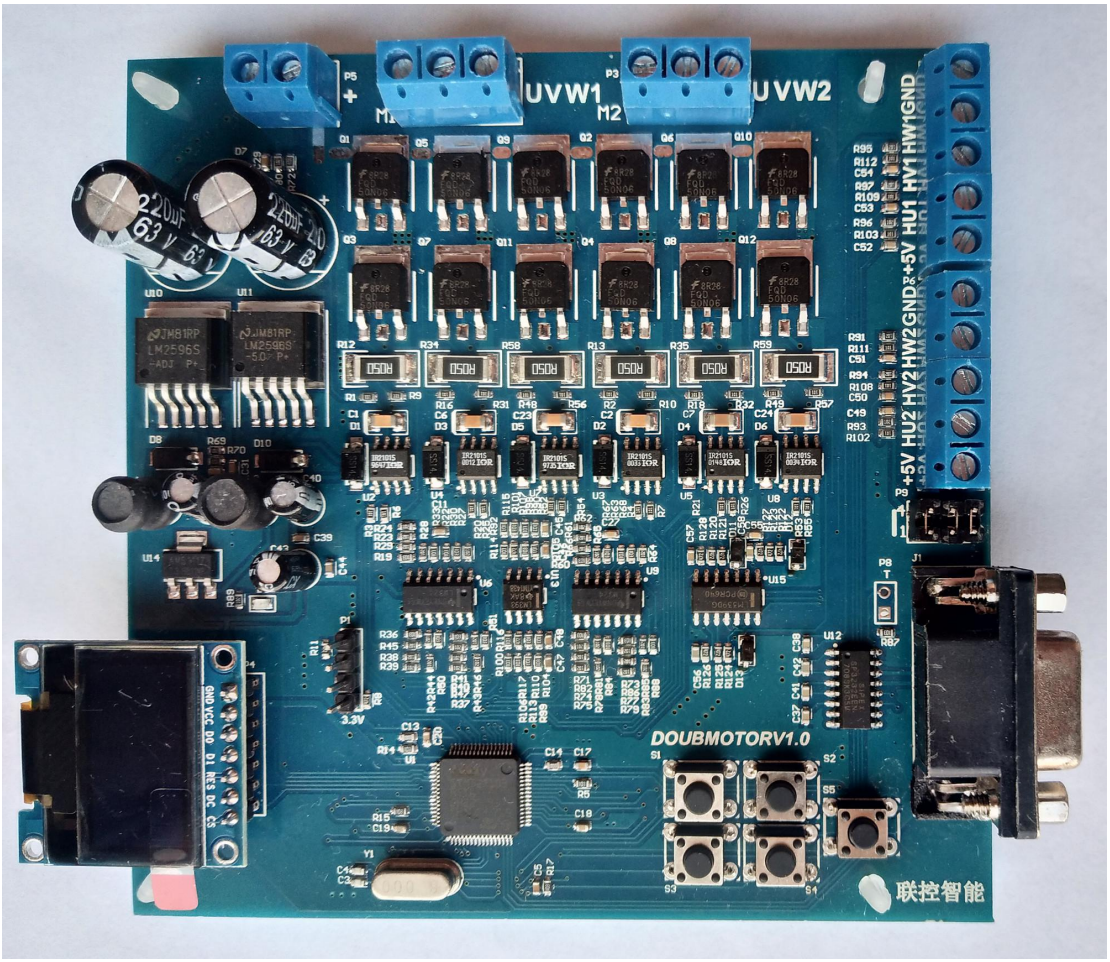

无感方波算法分析 V1.0

看程序要从主函数看起，包过单片机外设初始化

```
/*******************************************************
int main(void)
{
 #ifdef DEBUG
  debug();
 #endif
   int i;
  u8 keytemp=0,bkey;
  u8 flagccw=0;
   NVIC_InitTypeDef NVIC_InitStructure;
  /* System Clocks Configuration */
  RCC_Configuration();//时钟初始化

  /* NVIC configuration */
  NVIC_Configuration();//中断配置
  DMA_Configuration1();
  ADC_Configuration1();

  TIM1_Configuration1();   //定时器初始化
  TIM2_Configuration1();
  TIM4_Configuration1();



  GPIO_Configuration();  //GPIO
  UART3_Init();
  SysTick_Configuration();
```

```
 StartOk=0;//启动成功标志
 Comtime=0;//启动时换相次数
 aim_speed =200;
while (1)
{

 MotorStatus();

 keytemp= key_con(); //按键值

  if(keytemp==0)
  {
    bkey=1;
  }


if(keytemp==1)
{
  if(bkey==1)     //启动
  {
  motorstaus=CHONGD;
  startcnt=0;
  StartOk=0;
  bkey=0;
  Comtime=0;
  }
```

```
  if(keytemp==2)
  {
    if(bkey==1)
    {
       bkey=0;
       TIM_Cmd(TIM1, DISABLE);//停止
       TIM_CtrlPWMOutputs(TIM1, DISABLE);
       aim_speed =200;
    }


  }
  if(keytemp==3)
  {
    if(time >1000)
    {
        if(aim_speed <1200)
         aim_speed+=10;    //速度加
        time =0;
    }
     My_PWM = aim_speed;  //设定PWM值

  }
```

这个主函数主要 是单片机外设初始化，逻辑功能，包过电机启动，停止，速度加，速度减功能。

无刷电机方波无感配置，其实跟方波 HALL 配置一样，都是利用霍尔线中断去判断转子位置去实现换相，只不过无感方波采用硬件实现霍尔线中断。下面说一下如何配置和实现方法。

```
  /* 配置Hall接口IO */
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6|GPIO_Pin_7|GPIO_Pin_8;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOB, &GPIO_InitStructure);

/*霍尔信号线中断配置*/
GPIO_EXTILineConfig(GPIO_PortSourceGPIOB,GPIO_PinSource6);
GPIO_EXTILineConfig(GPIO_PortSourceGPIOB,GPIO_PinSource7);
GPIO_EXTILineConfig(GPIO_PortSourceGPIOB,GPIO_PinSource8);

EXTI_InitStructure.EXTI_Line = EXTI_Line6|EXTI_Line7|EXTI_Line8;
EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;
EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Rising_Falling;
EXTI_InitStructure.EXTI_LineCmd = ENABLE;
EXTI_Init(&EXTI_InitStructure);
```

这个是配置霍尔线引脚及中断。

```
NVIC_InitStructure.NVIC_IRQChannel = EXTI9_5_IRQChannel;
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 1;
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1;
NVIC_InitStructure.NVIC_IRQChannelCmd = DISABLE;//ENABLE;
NVIC_Init(&NVIC_InitStructure);
```

中断函数实现在这里

```
void EXTI9_5_IRQHandler(void)    //霍尔线中断判断
{

  u8 bHall=0,bHallstemp;

    if(EXTI_GetITStatus(EXTI_Line6)!= RESET)
  {
    EXTI_ClearITPendingBit(EXTI_Line6);
  }
    if(EXTI_GetITStatus(EXTI_Line7)!= RESET)
  {
    EXTI_ClearITPendingBit(EXTI_Line7);
  }
    if(EXTI_GetITStatus(EXTI_Line8)!= RESET)
  {
    EXTI_ClearITPendingBit(EXTI_Line8);
  }
  bHall=GPIO_ReadInputData(GPIOB);
  delays(2);

  bHallstemp=GPIO_ReadInputData(GPIOB);
```

```
if(bHall!=bHallstemp) return;
bHall=0;

//A相反电动势过零点检测
if(GPIO_ReadInputDataBit(GPIOB, GPIO_Pin_6)==1)
{
  bHall|= BIT0;
}
//B相反电动势过零点检测
if(GPIO_ReadInputDataBit(GPIOB, GPIO_Pin_7)==1)
{
  bHall|= BIT1;
}
//C相反电动势过零点检测
if(GPIO_ReadInputDataBit(GPIOB, GPIO_Pin_8)==1)
{
  bHall|= BIT2;
}
```

```
if(StartOk==0) //启动
{
    if(bHall == CheckBemf[bhallstep])
    {
      stComcount++;

    }
    else
    {
      stComcount=0;
    }
   if(stComcount>=ST_CNT)//连续检测到一定数量过零点切换到过零换相
    {
      StartOk=1;

    }
}
if(StartOk==1)
{

Hall_SW(bHall);
}
```

这个函数是实现无感方波启动到启动成功函数，先采集霍尔线，然后根据采集的霍尔值去跟实际换相值比较，连续采集一定数量都一致后，认为启动成功，切入霍尔换相，实现无感方波换相。

下面说一下无感方波启动过程，我们采用三段式无感启动方法，分为预充电，定位，拖动，然后运行。

```
void MotorStatus(void)  //电机状态
{

    switch(motorstaus)
    {
      case CHONGD:
          PrechargeD(50);      //预充电
          motorstaus= ALIGNED;
        break;
      case ALIGNED:
          Positioning(5,1000,50);  //定位
          motorstaus= STARRT;
        break;
      case STARRT:
          if(MotorStart(500,28)==1)  //电机无霍尔运行
          {
              motorstaus= RUN;
          }
          else
          {
              motorstaus= FAULT;
          }
        break;
      case RUN:
        break;
      case FAULT:
         MotorStop();
        break;
      default:
```

下面分析下预充电函数

```
void PrechargeD(u16 ctime)  //预充电
{
     TIM_Cmd(TIM8, ENABLE);//启动
      TIM_CtrlPWMOutputs(TIM8, ENABLE);
    time=0;

    PWM_UL_ON();  //U下管打开
    PWM_VL_ON();  //V下管打开
    PWM_WL_ON();  //W下管打开

   while(time<ctime)
   {
      time=time;
      time=time;
   }
    PWM_UL_OFF();  //U下管关闭
    PWM_VL_OFF();  //V下管关闭
    PWM_WL_OFF();  //W下管关闭
}
```

实现原理是 打开 PWM 把 MOS 下管打开，经过一定时间再把下管关掉，实现预充电功能。

下面说一下电机定位功能

```
void Positioning(u8 bhallstep ,u16 bpwm,u16 ctime)  //电机定位
{

    My_PWM = bpwm;  //定位pwm值
    Hall_SW(bhallstep);//定位电机
    time=0;
    while(time<ctime)  //定位时间
    {
        time=time;
        time=time;
    }

}
```

实现原理是可以设定 PWM 值，定位相线值和相应的定位时间

然后就是电机拖动

```
u8 MotorStart( u16 bpwm,u16 ctime)  //电机启动
{
  NVIC_InitTypeDef NVIC_InitStructure;
  NVIC_InitStructure.NVIC_IRQChannel = EXTI9_5_IRQChannel;
  NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
  NVIC_InitStructure.NVIC_IRQChannelSubPriority = 3;
  NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;//ENABLE;
  NVIC_Init(&NVIC_InitStructure);
   My_PWM = bpwm;  //启动pwm值

  do{

     bhtime=0;
     bhallstep++;
   if(bhallstep>5) bhallstep=0;
     Hall_SW(CheckBemf[bhallstep]);//启动电机
   while(bhtime<ctime)  //启动时间
    {
      bhtime=bhtime;
    }
    Comtime++;
}while(StartOk==0&&Comtime<COM_CNT);  //启动成功标志判断

if(Comtime>=COM_CNT)
{
      TIM_Cmd(TIM1, DISABLE);//停止
      TIM_CtrlPWMOutputs(TIM1, DISABLE);
      PWM_UL_OFF();  //U下管关闭
      PWM_VL_OFF();  //V下管关闭
      PWM_WL_OFF();   //W下管关闭
      return 0;
}

 return 1;   //小于COM_CNT次认为启动成功

}
```

实现原理是 打开霍尔线中断，设定 PWM 值，按照一定时间去拖动电机，当没达到启动次数启动标志为 1 就认为启动成功，当启动次数超过设定值认为启动失败，这里面的设定 PWM 值和间隔时间根据不同电机去调，调到这个值电机转动平稳，无换相抖动为止。为了达到这个效果，调试时可以先把启动次数设置大点。

这个是电机停止，原理是把上桥臂 PWM 关闭，下桥臂引脚输出低电平关闭。

```
/*PC6,PC7,PC8 为上半桥臂*/
 GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6| GPIO_Pin_7 | GPIO_Pin_8 ;
 GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
 GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
 GPIO_Init(GPIOC, &GPIO_InitStructure);

 /*PB0,PB1,PA7 为下半桥臂*/
 GPIO_InitStructure.GPIO_Pin =   GPIO_Pin_0 | GPIO_Pin_1;
 GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
 GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
 GPIO_Init(GPIOB, &GPIO_InitStructure);

 GPIO_InitStructure.GPIO_Pin =   GPIO_Pin_7;
 GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
 GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
 GPIO_Init(GPIOA, &GPIO_InitStructure);
```

这里配置 MOS 驱动引脚包过上下桥臂。

```
void MotorStop(void)  //电机停止
{
        TIM_Cmd(TIM8, DISABLE);//停止
        TIM_CtrlPWMOutputs(TIM8, DISABLE);
        PWM_UL_OFF();  //U下管关闭
        PWM_VL_OFF();  //V下管关闭
        PWM_WL_OFF();   //W下管关闭
}
```

以我的电机来说，用这个方法启动每次都能成功，而且力矩很大，希望大家有更好的想法加进去完善这个开源程序。