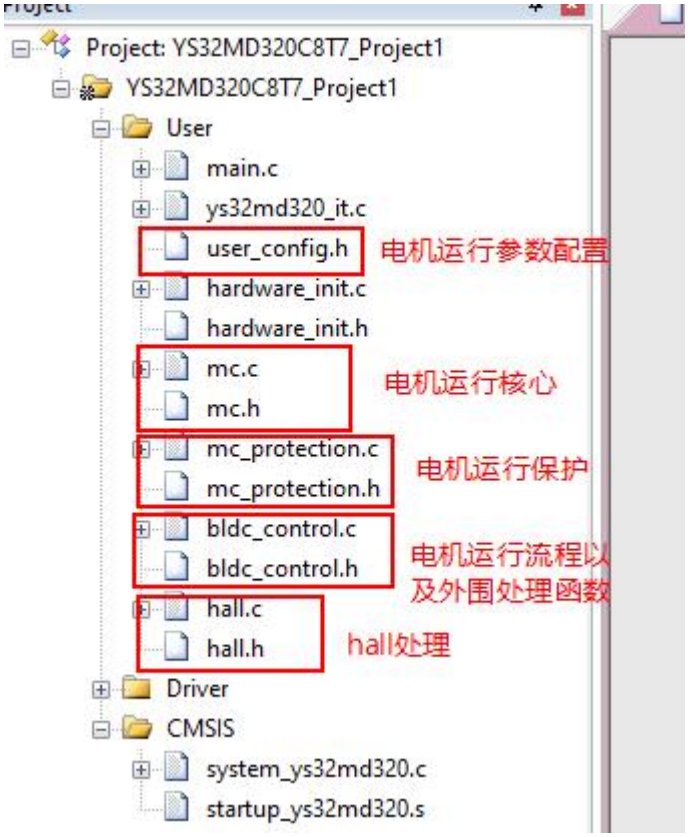
方波调试文档初始版本

1：主要文件概述



2：文件详述

Hardware_init.h：反电动势采样端口在这里修改

hardware_init.h

```
#include "ys32md320.h"


//反电动势端口
//Vdc
#define AN_VDC_CH    ADC_Channel_1
#define AN_VDC_PORT  GPIOF
#define AN_VDC_PIN   GPIO_Pin_4
//U_BEMF
#define AN_U_CH      ADC_Channel_2
#define AN_U_PORT    GPIOA
#define AN_U_PIN     GPIO_Pin_0
//V_BEMF
#define AN_V_CH      ADC_Channel_3
#define AN_V_PORT    GPIOA
#define AN_V_PIN     GPIO_Pin_1
//W_BEMF
#define AN_W_CH      ADC_Channel_7
#define AN_W_PORT    GPIOA
#define AN_W_PIN     GPIO_Pin_5
//I_BUS
#define AN_IBUS_CH   ADC_Channel_14
#define AN_IBUS_PORT GPIOB
#define AN_IBUS_PIN  GPIO_Pin_11

//HO1-UH LO1-UL
//HO2-VH LO2-VL
//HO3-WH LO3-WL
//123对应uvw
#define UVW_OUT    (H123)
#define H123  0
#define H132  1
#define H213  2
#define H231  3
#define H312  4
#define H321  5
#if(UVW_OUT == H123)
  #define UH_ON    {TIM1->CCER |= TIM_CCER_CC1E;}
  #define UH_OFF   {TIM1->CCER &= ~TIM_CCER_CC1E;}
  #define VH_ON    {TIM1->CCER |= TIM_CCER_CC2E;}
  #define VH_OFF   {TIM1->CCER &= ~TIM_CCER_CC2E;}
```

母线电压ad采样端口配置

U相反电动势ad采样端口

V相反电动势ad采样端口

W相反电动势ad采样端口

母线电流ad采样端口

默认引脚

此处针对D320芯片配置的；

如果板子上面没有按照默认引脚接入
则通过这里对应修改引脚引脚顺序

Hall.h：hall 引脚以及顺序修改

hall.h

```
#ifndef __HALL_H__
#define __HALL_H__

#include "ys32md320.h"

#define HA 3
#define HB 1
#define HC 2

#define H1_PORT     GPIOB
#define H1_PIN      GPIO_Pin_5
#define H2_PORT     GPIOB
#define H2_PIN      GPIO_Pin_6
#define H3_PORT     GPIOB
#define H3_PIN      GPIO_Pin_7

#define H1_Read()   GPIO_ReadInputDataBit(H1_PORT,H1_PIN)
#define H2_Read()   GPIO_ReadInputDataBit(H2_PORT,H2_PIN)
#define H3_Read()   GPIO_ReadInputDataBit(H3_PORT,H3_PIN)
```

hall顺序修改

hall引脚配置修改

User_config.h：主要用于配置电机运行相关参数

```
//主频参数
#define SYS_CLK     (48000000)
#define PWM_CLK     (16000) //(16000)            //电机驱动频率
#define PWM_PERIOD (SYS_CLK/PWM_CLK)            //周期

//有感无感
#define BLDC_MOTOR     (SENSOR_NONE)
#define SENSOR_HALL     (1)                     //有HALL驱动
#define SENSOR_NONE     (0)

//电机转速范围选择
#define MC_SPEED_SELECT   (MC_LOW_SPEED)
#define MC_LOW_SPEED    (0)                     //电周期60000以下
#define MC_HIGH_SPEED   (1)                     //电周期60000-140000, 需要PWM_CLK配置30K以上
```

PWM_CLK: 电机驱动频率，最大可以设置到 40K。

BLDC_MOTOR: 选择电机驱动方式，1 为有 hall 运行，0 为无感运行

MC_SPEED_SELECT: 电机的转速范围选择，具体看注释，若是选 MC_HIGH_SPEED,则建议 PWM_CLK 为 40KHz

```
19 //电机极对数
20 #define POLE        1
21 //外部调速方式
22 #define SPEED_CONTORL (NONE)                    //调速方式选择
23 #define PWM_SPEED      (0)                      //pwm调速
24 #define VOL_SPEED      (1)                      //电压调速
25 #define NONE           (2)                      //无
26
27 //运行模式
28 #define MC_LOOP     (OPEN_LOOP)
29 #define STRONG_PULL (0)                         //强拖运行
30 #define OPEN_LOOP   (1)                         //开环运行
31 #define CLOSE_LOOP  (2)                         //速度环
32 //闭环模式
33 #define MC_CLOSE_LOOP  (SPEED_LOOP)
34 #define SPEED_LOOP      (0)                     //速度环
35 #define CUR_LOOP        (1)
36 #define POWER_LOOP      (2)
```

POLE：电机极对数，问厂家要该参数。

SPEED_CONTORL：电机调速方式，主要有 PWM 调速和模拟电压调速

MC_LOOP：运行方式，强拖、开环、闭环

MC_CLOSE_LOOP：如果上面选择了闭环运行，此处才起作用，常用的有速度环和功率环。

```
37 //预定位
38 #define PRE_POS_CHECK_ENABLE (0)                //预定位测试
39 #define PRE_POS_MODE (STRONG_POS)               //预定位方式选择
40 #define STRONG_POS (0)                          //强制定位
41 #define IPD_POS (1)                             //ipd定位
42 //强制定位
43 #define STRONG_POS_DUTY (0.3)                   //预定位电流
44 #define STRONG_POS_TIME 500                     //预定位时间
45 //ipd定位
46 #define IPD_DUTY_H  (50)                        //ipd脉宽
47 #define IPD_DUTY_L  (150)
48
49 //启动强拖
50 #define START_DUTY            (0.2)             //百分比0.0-1.0
51 #define START_ADD_TIME        (10)              //加速周期
52 #define MIN_COM_TIMES         (2)               //最小拖动周期换相
53 #define MAX_COM_TIMES         (10)              //最大拖动周期换相
54 #define START_SPEED_MAX    (400)                //480启动最大速度2000
55 #define START_SPEED_MIN    (100)                //启动最小速度1000
56 #define START_SPEED_MAX_VALUE  (((PWM_CLK*60)/(START_SPEED_MAX*6*POLE)))
57 #define START_SPEED_MIN_VALUE  (((PWM_CLK*60)/(START_SPEED_MIN*6*POLE)))
```

PRE_POS_MODE：预定位分强制定位和 ipd 定位

PRE_POS_CHECK_ENABLE：设置 1 时用来测试电机定位

STRONG_POS_DUTY：强制定位占空比，范围 0-1，设置越大，定位电流越大。

STRONG_POS_DUTY：强制定位时间，数值越大定位时间越长

IPD_DUTY_H：ipd 脉宽，控制 pid 电流，数值 1-50，凸极性越不明显的电机设置值越大

IPD_DUTY_L：最小取 IPD_DUTY_L 的 3 倍。最大可以根据采样的 ipD 电流调节

START_DUTY：强拖占空比，数值越大，强拖电流越大

START_ADD_TIME：强拖加速度，数值越大加速越慢，数值越小，加速越快

MAX_COM_TIMES：拖动到 START_SPEED_MAX 后，继续运行 MAX_COM_TIMES 后切换到反电动势换相周期。

START_SPEED_MAX：拖动最大速度

START_SPEED_MIN：拖动最小速度

启动强拖的时候需要协调配置 START_DUTY、START_ADD_TIME、MAX_COM_TIMES、START_SPEED_MAX、START_SPEED_MIN 参数，让电机强拖到一个比较合适的速度，以便于能个检测到较好的反电动势信号从而完成启动。

```
59
60  //运行
61  #define RUN_DUTY            (0.2)               //百分比
62  #define ADVANCE_ANGLE       (150)               //超前换相角度0-400
63  #define FLOW_TIMES          (3)                 //续流时间0-10
64  #define OVER_TIME_COM       (8000)              //超时换相时间us
65  #define OPEN_RUN_TIME       (10)                //开环切闭环时间
66  #if(MC_LOOP == CLOSE_LOOP)  //闭环爬坡
67  #define RAMP_PERIOD         (10)                //爬坡周期ms
68  #define RAMP_ADD_SPEED      (500)               //上坡加速度
69  #define RAMP_SUB_SPEED      (500)               //下坡加速度
70  #elif(MC_LOOP == OPEN_LOOP) //开环爬坡
71  #define RAMP_PERIOD         (100)               //爬坡周期ms
72  #define RAMP_ADD_SPEED      (5)                 //上坡加速度
73  #define RAMP_SUB_SPEED      (5)                 //下坡加速度
74  #endif
75
76  #define MAX_SPEED           (100000)            //最大转速
77  #define MIN_SPEED           (200)               //最小转速
78
79  //速度pi
80  #define START_KP    (0.01)                      //速度环kp
81  #define START_KI    (0.001)                     //速度环ki
82  #define SP_KP       (0.8)
83  #define SP_KI       (0.02)
84  #define LIMIT_MAX   1200                        //最大输出
85  #define LIMIT_MIN   200                         //最小输出
86
```

RUN_DUTY：运行占空比，在开环模式下起作用。

ADVANCE_ANGLE：超前角度，由于电路滤波和程序计算的情况，会导致换相会有一定的滞后性，通过这个值进行补偿，范围为 0-400。

FLOW_TIMES：续流屏蔽时间，根据电机不同，时间设置不同

RAMP_PERIOD：爬坡周期，周期越小爬坡速度越快

RAMP_ADD_SPEED：上坡加速度，越大速度上升越快

RAMP_SUB_SPEED：下坡加速度，越大速度下降越快

MAX_SPEED：电机最大转速，

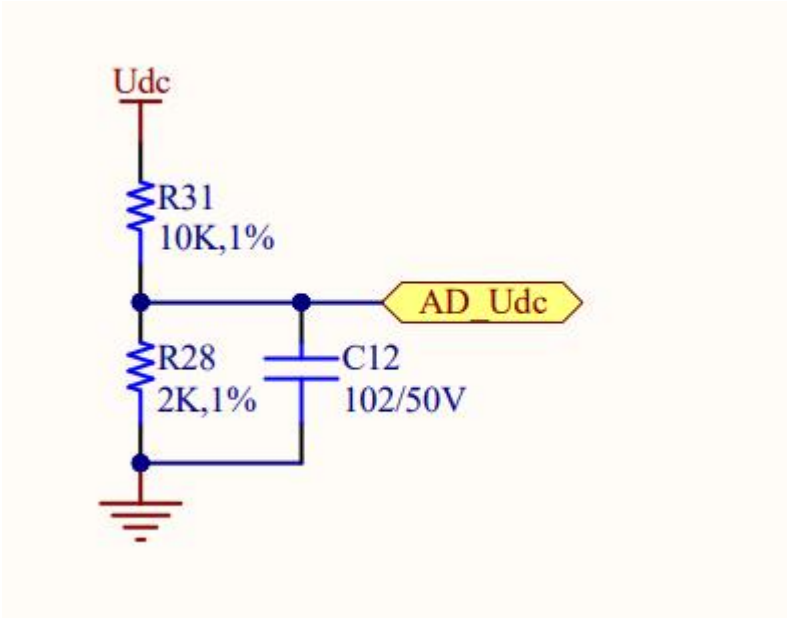
MIN_SPEED：电机最小转速，

START_KP：速度环 kp 参数，kp 越小波动越小

START_KI：速度环 ki 参数，ki 越小速度跟随越慢

LIMIT_MAX：最大限幅输出

```
//母线电压采样参数    反电动势分压电阻和母线电压分压电阻一致
#define AD_VREF     (5.0)                          //ad参考电压
#define UP_R1       (10)                           //上分压电阻
#define DOWN_R2     (2)                            //下分压电阻
```



AD_VREF：ad 采样参考电压默认 5V

UP_R1：如上图 R31

DOWN_R2：如上图 R28

母线电压分压电阻和母线电压分压电阻一致。

```
//电压保护
#define VOL_PROTECT_ENABLE     (0)          //电压保护使能
#define VOL_RESTART_ENABLE     (1)          //电压保护恢复
#define OVER_VOL               (9.0)        //过压保护值V
#define REV_OVER_VOL           (8.4)        //过压恢复值V
#define UNDER_VOL              (6.5)        //欠压保护值V
#define REV_UNDER_VOL          (7.0)        //欠压保护恢复值V
#define OVER_VOL_CHECK_TIME    (100)        //过压检测时间ms
#define UNDER_VOL_CHECK_TIME   (100)        //欠压检测时间ms
#define REV_VOL_CHECK_TIME     (100)        //电压恢复检测时间ms
#define REV_OVER_VOL_VALUE     (REV_OVER_VOL/(UP_R1+DOWN_R2)*DOWN_R2/AD_VREF*4096)
#define REV_UNDER_VOL_VALUE    (REV_UNDER_VOL/(UP_R1+DOWN_R2)*DOWN_R2/AD_VREF*4096)
#define OVER_VOL_VALUE         (OVER_VOL/(UP_R1+DOWN_R2)*DOWN_R2/AD_VREF*4096)
#define UNDER_VOL_VALUE        (UNDER_VOL/(UP_R1+DOWN_R2)*DOWN_R2/AD_VREF*4096)

//堵转保护
#define STALL_PROTECT_ENABLE   (1)          //堵转使能
#define STALL_RESTART_ENABLE   (1)          //堵转恢复使能
#define STALL_CUR              (8.0)        //堵转电流A
#define STALL_CHECK_TIMES       8000         //堵转检测次数
#define STALL_CUR_TIME          2000         //堵转电流检测时间ms
#define STALL_RESTART_TIMES     3           //堵转重启动次数
#define STALL_CUR_VALUE        (STALL_CUR*I_SHUT*OPA_AMP/AD_REF*4096)
```

```
//电流硬件参数
#define I_SHUT                      (0.001)   //电流采样电阻Ω
#define AD_REF                      (5.0)     //ad电压参考值
#define OPA_AMP                     (11)      //放大倍数
#define OPA_BIAS                    (2.5)     //采样偏置

//限流
#define LIMIT_CUR_ENABLE            (0)       //限流使能位
#define LIMIT_CURRENT               (2.0)    //限流电流   //14A
#define LIMIT_ADD_TIME              (100)      //限流加速度
#define LIMIT_CUR_RAMP              (100)// (10)       //限流稳定度
#define LIMIT_CUR_VALUE             (LIMIT_CURRENT*I_SHUT*OPA_AMP*4096.0/AD_REF)
//硬件过流处理
#define HARD_CUR_PROTECT_ENABLE     (1)
//软件过流保护
#define CUR_PROTECT_ENABLE          (1)         //过流保护使能
#define CUR_RESTART_ENABLE          (0)         //过流保护恢复使能
#define OVER_CURRENT1               (25.0)       //A 1级保护
#define OVER_CURRENT_CHECK_TIMES1   (2000)       //ms
#define OVER_CURRENT2               (30.0)       //A 2级保护
#define OVER_CURRENT_CHECK_TIMES2   (1000)       //ms
#define OVER_CURRENT3               (40.0)       //A 3级保护
#define OVER_CURRENT_CHECK_TIMES3   (500)        //ms
#define OVER_CUR_SHIFT              (0)         //A
#define OVER_CURRENT_VALUE1         ((OVER_CURRENT1*I_SHUT*OPA_AMP)*4096/AD_REF)
#define OVER_CURRENT_VALUE2         ((OVER_CURRENT2*I_SHUT*OPA_AMP)*4096/AD_REF)
#define OVER_CURRENT_VALUE3         ((OVER_CURRENT3*I_SHUT*OPA_AMP)*4096/AD_REF)
#define OVER_CUR_SHIFT_VALUE        ((OVER_CUR_SHIFT*I_SHUT*OPA_AMP)*4096/AD_REF)
#define MAX_CUR                     ((AD_REF-OPA_BIAS)/OPA_AMP/I_SHUT)      //1A-3A
```

```
//过温保护
#define TEMP_R_UP                   (10000.0) //R
#define TEMP_R_DOWN_NTC             (1450.0)  //R  85度-1.45K
#define TEMP_PROTECT_ENABLE         (1)      //过温保护使能
#define TEMP_RESTART_ENABLE         (0)      //过温保护恢复
#define OVER_TEMP                   (85.0)   //温度保护值
#define TEMP_CHECK_TIME             (1000)   //温度检测时间
#define OVER_TEMP_VALUE             (TEMP_R_DOWN_NTC*4096/(TEMP_R_UP+TEMP_R_DOWN_NTC))
//缺相保护
#define LOSS_FAULT_ENABLE           (1)    //缺相使能
#define LOSS_RESTART_ENABLE         (0)    //缺相重启
#define LOSS_RESTART_TIMES          (3)    //缺相重启次数
#define LOSS_CHECK_TIMES            (300)  //缺相检测次数
```

过压欠压保护、堵转保护、过流保护、过温保护、缺相保护等等。都在上面设置各项参数。具体各项都有详细的注释。

```
//刹车处理
#define BRAKE_ENABLE                (1)
#define BRAKE_SPEED                 (2000)    //rpm
#define BRAKE_TIME                  (1000)     //ms
```

BRAKE_ENBLE：刹车使能

BRAKE_SPEED：刹车速度

BRAKE_TIME：刹车时间，这个参数预留

主要在 mc_run 函数和 mc_brake 函数中处理，具体可以去参考这两个

函数

```
//fg信号输出
#define FG_OUT_ENABLE     (0)       //FG信号输出使能
#define FG_MUL   (1)                //FG输出 转速倍数(1~POLE)
#define FG_PORT   GPIOF             //FG引脚配置
#define FG_PIN    GPIO_Pin_6
//pwm调速信号输入
#define PWM_IN     (HIGH)            //pwm有效电平
#define LOW        (0)
#define HIGH       (1)

//正反转
#define MOTOR_DIR    0
//启动测试
#define MC_START_TEST (0)         //启动测试
#define MC_HARDWARE_TEST (0)      //电机电路测试运行
```

FG 信号输出

FG_MUL: 主要用于设置输出频率

Pwm 调速信号输入

HIGH：输入高电平有效

LOW：输入低电平有效

MOTOR_DIR:用于设置初始方向

MC_START_TEST: 启动测试，用于调试好电机后，自动测试启动是否会出现失败，具体参考启停函数

MC_HARDWARE_TEST: 用于测试硬件电路输出是否正常。

```
/*******************************************************
函数名称：isr_1ms(void)
功能：    1ms中断函数
备注：
*******************************************************/
unsigned char isr_1ms_cnt = 0;
void isr_1ms(void)
{
  if(++isr_1ms_cnt>=1)
  {
    isr_1ms_cnt = 0;
    mc_main();
    pwm_sign_calc();          //pwm输入采集
    mc_on_off_control();      //电机启动停止函数
    mc_speed_calc();          //速度计算
    loop_control();
    loss_phase();             //缺相故障保护
    vol_protection();         //电压故障检测
    stall_protection();       //堵转故障检测
    temp_protection();        //温度检测
    hall_protection();        //hall故障
    cur_protection();         //过流故障检测
    fg_out();                 //FG信号输出函数
    led_fault();              //led闪灯提示错误
    mc_start_test();          //启停测试
    //led_vol_display();
  }

}
```

1ms 中断函数中处理非电机运行的各个函数。

Mc_main()：电机运行状态函数

其他的都有注释；

2：调试说明

1）配置好电压、电流、反电动势采样引脚。

2）正确配置 uvw 三相输出引脚

3）使能 MC_HARDWARE_TEST 位，下载程序，连接电源，测试 uvw 三相是否有相同载波输出。如果三相都有正常输出，则驱动部分电路没有问题。否则有问题。

4）配置好电机极对数，使用 open_loop 模式，设置好合适的启动电流和强拖参数，对电机进行强拖，找到合适的换相波形，得到准确的换相速度。

5）通过波形找到合适的换相速度后，把参数重新配置到强拖参数中，多次运行，看电机是否能进入运行模式正确换相。

通过上面调试，电机能正确换相后，第一阶段电机运行调试完成。