

目 录

第 1 章 无刷直流电机驱动实验指导.....	2
1.1 概述.....	2
1.2 有感驱动速度开环模式实验	3
1.3 有感驱动速度闭环模式实验	4
1.4 无感驱动速度开环模式实验	6
1.5 无感驱动速度闭环模式实验	7

广州致远 运动控制

第1章 无刷直流电机驱动实验指导

无刷直流电机是在有刷直流电机的基础上发展起来的，它用位置传感器代替了有刷直流电机的电刷和换向器，因此其内部发生的电磁过程和普通的有刷直流电机类似，无刷直流电机既具有直流电机良好的调速性能等特点，又具有交流电机结构简单、无换向火花、运行可靠和易于维护等优点。

本实验指导基于富士通 FM3 微处理器和 BLDC-DK615 驱动板。

在开始实验之前先详细阅读教材——无刷直流电机的原理与驱动。

1.1 概述

BLDC-DK615 是一款无刷电机驱动学习板，支持无刷电机有位置传感器驱动方案（下文简称“有感驱动”），无位置传感器驱动方案（下文简称“无感驱动”）。

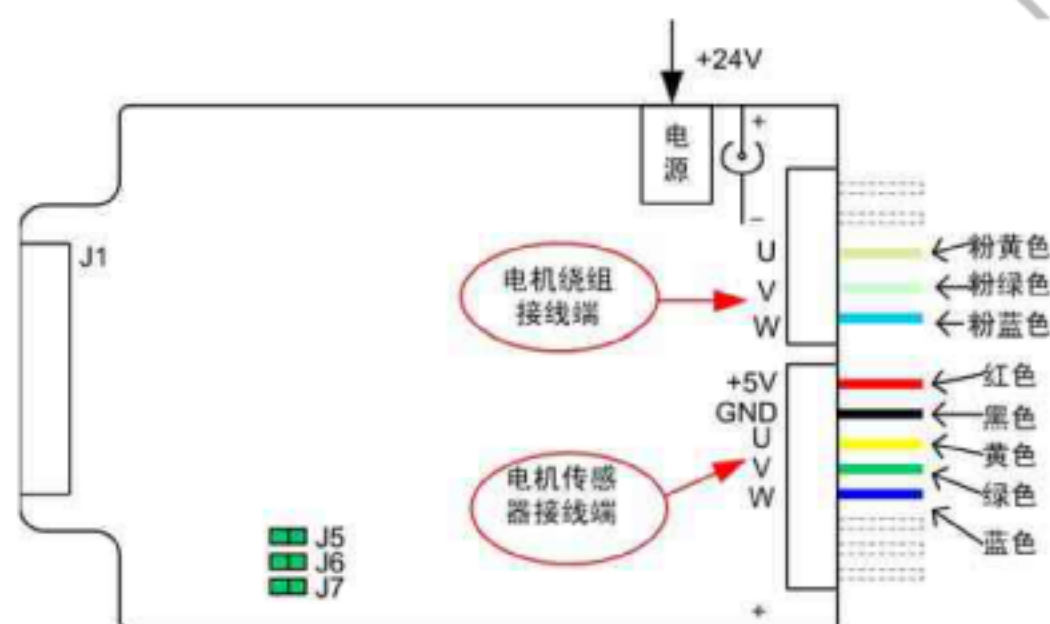


图 1.1 BLDC-DK615 驱动板接线图

FSSDC-9B506-EK 是一款基于富士通 FM3 微处理器（Cortex-M3 内核）的开发板。

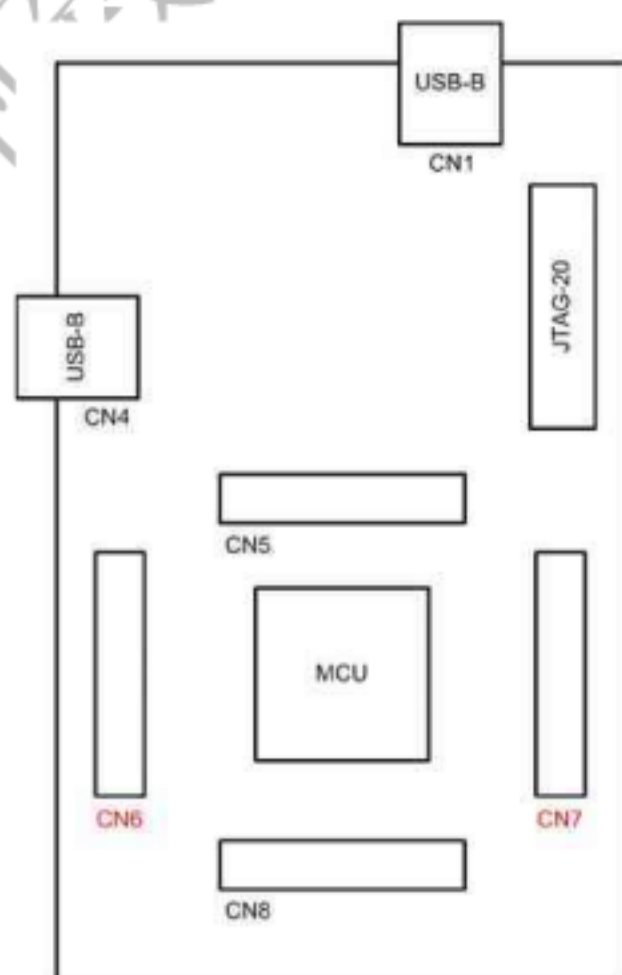


图 1.2 FSSDC-9B506-EK 开发板布局图

表 1.1 详细说明了 BLDC-DK615 与 FSSDC-9B506-EK 接线关系。

表 1.1 BLDC-DK615 与 FSSDC-9B506-EK 接线表

BLDC-DK615		FSSDC-9B506-EK	
J1-1	PWM_UH	CN6-25	P3B
J1-2	PWM_UL	CN6-24	P3A
J1-3	PWM_VH	CN6-27	P3D
J1-4	PWM_VL	CN6-26	P3C
J1-5	PWM_WL	CN6-29	P3F
J1-6	PWM_WH	CN6-28	P3E
J1-11	IO_HA	CN6-22	P38
J1-12	IO_HB	CN6-21	P37
J1-13	IO_HC	CN6-20	P36
J1-21	ADC_CUR	CN7-77	P1D
J1-22	ADC_PHA	CN7-74	P19
J1-23	ADC_PHB	CN7-75	P1A
J1-24	ADC_PHC	CN7-76	P1B
J1-26	GND	CN6-30	GND

注：J1 为 BLDC-DK615 控制信号接口，CN6/7 为 FSSDC-9B506-EK 的 MCU 引脚接口。

1.2 有感驱动速度开环模式实验

1. 实验原理

有感驱动速度开环模式，电机需要用有位置传感器电机，实验利用位置传感器的信号反馈位置，根据位置信息对电机进行换向，使电机转动连续平稳。

2. 实验步骤

(1)、断开图 1.1 中的 J5、J6 和 J7 跳线，并用杜邦线将 BLDC-DK615 的 J1 与 FSSDC-9B506-EK 的 CN6/7 按表 1.1 所示的关系连接，确认接线无误后分别给 2 块板子上电。

(2)、用 IAR v6.2 打开光盘目录【单片机程序\FM3_ZMC_Hall】下的【ZMC_Hall_Open】工程，并重新编译。

(3)、用 AK100 仿真器下载程序到 FSSDC-9B506-EK 的 MCU。按复位键程序全速运行，电机恒速运行。

3. 程序解析

有感驱动换相由【MiddleWare\ZLG Motion Control\ZMC_App\BLDC】目录下 ZMC_BldcHall.c 文件的 zmcBldcHallCallback 函数实现，部分代码如下：

程序清单 1.1 有感驱动换相程序

```

/*****

```

此函数为霍尔中断回调函数

```

/*****
void  zmcBldcHallCallback (void)
{
    static INT8U    ucPreSector = 0xff;
    static INT8U    ucSector    = 0;

    /*
     * 自动换相
     */
    ZMC_BLDC_SEC_HALL(&ucSector);          /* 获取当前扇区          */

    if (ucSector != ucPreSector) {          /* 2 次扇区不一致      */
        ZMC_BLDC_PWM_COMMUTATE(GucDir, ucSector); /* 换相                */
        ucPreSector = ucSector;            /* 记录本次扇区        */
    }
}
*****/

```

1.3 有感驱动速度闭环模式实验

1. 实验原理

有感驱动速度闭环模式，电机需要用有位置传感器电机，实验利用位置传感器的信号反馈位置，根据位置信息对电机进行换向，使电机转动连续平稳，并同时利用电机的位置传感器计算电机当前的转速。

2. 实验步骤

(1)、断开图 1.1 中的 J5、J6 和 J7 跳线，并用杜邦线将 BLDC-DK615 的 J1 与 FSSDC-9B506-EK 的 CN6/7 按表 1.1 所示的关系连接，确认接线无误后分别给 2 块板子上电。

(2)、用 IAR v6.2 打开光盘目录【单片机程序\FM3_ZMC_Hall】下的【ZMC_Hall_PID】工程，并重新编译。

(3)、用 AK100 仿真器下载程序到 FSSDC-9B506-EK 的 MCU。按复位键程序全速运行，电机恒速运行。

3. 程序解析

有感驱动换相及测速由【MiddleWare\ZLG Motion Control\ZMC_App\BLDC】目录下 ZMC_BldcHall.c 文件的 zmcBldcHallCallback 函数实现，部分代码如下：

程序清单 1.2 有感驱动换相及测速程序

```

/*****
此函数为霍尔中断回调函数
*****/
void  zmcBldcHallCallback (void)
{
    static INT8U    ucSectorCnt = 0;
    static INT8U    ucPreSector = 0xff;

```

```

static INT8U   ucSector   = 0;
               INT32U   ulTimeCnt = 0;

ZMC_BLDC_SEC_HALL(&ucSector);           /* 获取当前扇区 */

/*
 * 自动换相
 */
if (ucSector != ucPreSector) {          /* 2 次扇区不一致 */
    ZMC_BLDC_PWM_COMMUTATE(GucDir, ucSector); /* 换相 */
    ucPreSector = ucSector;               /* 记录本次扇区 */
    ucSectorCnt++;
}

/*
 * 速度计算
 */
if (ucSectorCnt >= 3) {                  /* 换相 3 次计算 1 次速度 */
    ucSectorCnt = 0;

    ulTimeCnt = zmdTimerInitValueCntGet(ZMD_16BIT_TIMER6) -
                zmdTimerCurValueCntGet(ZMD_16BIT_TIMER6);
    zmdTimerResetCnt(ZMD_16BIT_TIMER6); /* 复位定时器，重新计时 */

    if (!ulTimeCnt) {                    /* 防止除数为 0 */
        return;
    }

    /* 计算速度，电机极对数 2 */
    GulSpeed = ((zmdTimerClkGet(ZMD_16BIT_TIMER6) * 30) / 2) / ulTimeCnt;
}
}

```

速度 PID 调节由 ZMC_BldcHall.c 文件的 zmcBldcPwmCallback 函数实现，部分代码如下：

程序清单 1.3 速度 PID 调节程序

```

ZMC_PID_DATA   GtPid       = ZMC_PID_DATA_DEF; /* PID 结构体变量 */
/*****
  此函数为 PWM 中断回调函数
 *****/
void zmcBldcPwmCallback (void)
{
    static INT32U   ulPwmTick = 0;

    if (GucStart != 2) { /* 电机未启动 */
        return;
    }
}

```

```

}

if ( ulPwmTick++ >= (ZMC_BLDC_PID_PERIOD * 20) ) {          /* PID 调节周期          */
    ulPwmTick = 0;

    GtPid.qRef = ZMC_Q16(GulTarSpeed);                        /* PID 参考值          */
    ZMC_PID(&GtPid, ZMC_Q16(GulSpeed));                       /* PID 的 PID 调节器    */
    ZMC_BLDC_PWM_DUTY(GtPid.qOut);                            /* 设置 PWM 占空比     */
}
}

```

1.4 无感驱动速度开环模式实验

1. 实验原理

无感驱动速度开环模式，利用采样电机的反电势，并检测反电势的过零点来判断电机转子的位置，根据电机转子的位置来实现换向，使电机能平稳连续转动。

2. 实验步骤

(1)、短接图 1.1 中的 J5、J6 和 J7 跳线，并用杜邦线将 BLDC-DK615 的 J1 与 FSSDC-9B506-EK 的 CN6/7 按表 1.1 所示的方法连接，确认接线无误后分别给 2 块板子上电。

(2)、用 IAR v6.2 打开光盘目录【单片机程序\FM3_ZMC_Bemf】下的【ZMC_Bemf_Open】工程，并重新编译。

(3)、用 AK100 仿真器下载程序到 FSSDC-9B506-EK 的 MCU。按复位键程序全速运行，电机恒速运行。

3. 程序解析

无感驱动换相由【MiddleWare\ZLG Motion Control\ZMC_App\BLDC】目录下 ZMC_BldcBemf.c 文件的 zmcBldcPwmCallback 函数实现，部分代码如下：

程序清单 1.4 无感驱动换相程序

```

ZMC_BEMF_DATA  GtBemf = ZMC_BEMF_DATA_DEF;
/*****
此函数为 PWM 中断回调函数
*****/
void  zmcBldcPwmCallback (void)
{
    static INT8U  ucPreSector = 0xff;
    static INT8U  ucSector    = 0;

    zmdAdcDataGet(ZMD_ADC0, &GulAdcBemf[0]);          /* 读取反向电动势 ADC 转换值  */

    ZMC_BLDC_UPHA_PU(&GtBemf.qVa);                    /* 读取 A 相反向电动势，Q24    */
    ZMC_BLDC_UPHB_PU(&GtBemf.qVb);                    /* 读取 B 相反向电动势，Q24    */
    ZMC_BLDC_UPHC_PU(&GtBemf.qVc);                    /* 读取 C 相反向电动势，Q24    */
}

```

```

/*
 * 自动换相
 */
ZMC_BLDC_SEC_BEMF(GtBemf, &ucSector);          /* 根据反向电动势计算换相扇区 */

if (ucPreSector != ucSector) {                    /* 2 次扇区不一致 */
                                                    /* 换相 */
    ZMC_BLDC_PWM_COMMUTATE(GtBemf.ucCmtDir, ucSector);
    ucPreSector = ucSector;                        /* 记录本次扇区 */
}
}

```

1.5 无感驱动速度闭环模式实验

1. 实验原理

无感驱动速度闭环模式，利用采样电机的反电势，并检测反电势的过零点来判断电机转子的位置，根据电机转子的位置来实现换向，使电机能平稳连续转动，同时通过计算过零点之间的时间差来换算出电机的转速。

2. 实验步骤

(1)、短接图 1.1 中的 J5、J6 和 J7 跳线，并用杜邦线将 BLDC-DK615 的 J1 与 FSSDC-9B506-EK 的 CN6/7 按表 1.1 所示的方法连接，确认接线无误后分别给 2 块板子上电。

(2)、用 IAR v6.2 打开光盘目录【单片机程序\FM3_ZMC_Bemf】下的【ZMC_Bemf_PID】工程，并重新编译。

(3)、用 AK100 仿真器下载程序到 FSSDC-9B506-EK 的 MCU。按复位键程序全速运行，电机恒速运行。

3. 程序解析

无感驱动换相、测速及速度 PID 调节由【MiddleWare\ZLG Motion Control\ZMC_App\BLDC】目录下 ZMC_BldcBemf.c 文件的 zmcBldcPwmCallback 函数实现，部分代码如下：

程序清单 1.5 无感驱动换相、测速及速度 PID 调节程序

```

ZMC_PID_DATA  GtPid      = ZMC_PID_DATA_DEF;          /* PID 结构体变量 */
/*****
此函数为 PWM 中断回调函数
*****/
void  zmcBldcPwmCallback (void)
{
    static INT8U  ucPreSector = 0xff;
    static INT8U  ucSector    = 0;
    static INT32U  ulPwmTick  = 0;

    /*
     * 自动换相
     */

```

```

zmdAdcDataGet(ZMD_ADC0, &GulAdcBemf[0]);          /* 读取反向电动势 ADC 转换值 */

ZMC_BLDC_UPHA_PU(&GtBemf.qVa);                     /* 读取 A 相反向电动势, Q24 */
ZMC_BLDC_UPHB_PU(&GtBemf.qVb);                     /* 读取 B 相反向电动势, Q24 */
ZMC_BLDC_UPHC_PU(&GtBemf.qVc);                     /* 读取 C 相反向电动势, Q24 */

ZMC_BLDC_SEC_BEMF(GtBemf, &ucSector);              /* 根据反向电动势计算换相扇区 */

if (ucPreSector != ucSector) {                      /* 2 次扇区不一致 */
                                                    /* 换相 */
    ZMC_BLDC_PWM_COMMUTATE(GtBemf.ucCmtDir, ucSector);
    ucPreSector = ucSector;                          /* 记录本次扇区 */
}

/*
 * 计算速度
 */
if (GtBemf.ulCmtDlyCnt > 200000) {                  /* 转速过低 */
    GtBemf.ulCmtDlyCnt = 0;
    GulSpeed = 0;
} else {
    GulSpeed = 50000 / GtBemf.ulAvrgCmtDly;         /* 计算转速, 电机极对数 2 */
}

/*
 * 速度 PID 调节
 */
if (!GucStart) {                                    /* 电机未启动 */
    return;
}

if ( ulPwmTick++ >= (ZMC_BLDC_PID_PERIOD * 20) ) { /* PID 调节周期 */
    ulPwmTick = 0;

    GtPid.qRef = ZMC_Q16(GulTarSpeed);               /* PID 参考值 */
    ZMC_PID(&GtPid, ZMC_Q16(GulSpeed));              /* PID 的 PID 调节器 */
    ZMC_BLDC_PWM_DUTY(GtPid.qOut);                  /* 设置 PWM 占空比 */
}
}

```