



峰昭科技

FU68xx-有霍尔-FOC

调试说明文档-V1.0.0

峰昭科技(深圳)有限公司

Fortior Technology(Shenzhen) Co.,Ltd.

深圳市南山区科技中二路软件园 11 栋 2 楼 203 室,518057
Room203,2/F,Building No.11,Keji Central Road 2,Software Park,
High-Tech Industrial Park, Shenzhen,P.R.China

Tel: [86-755-26867710](tel:86-755-26867710)

Fax: [86-755-26867715](tel:86-755-26867715)

Contained herein

Copyright by FortiorTechnology(Shenzhen) Co., Ltd all rights reserved.

1. 概述

本文档介绍有感 FOC 原理以及如何调整程序中的相关参数。以 FU6818-DEMO_V2.0 电机控制开发板为例作说明。

2. 有感 FOC 控制原理

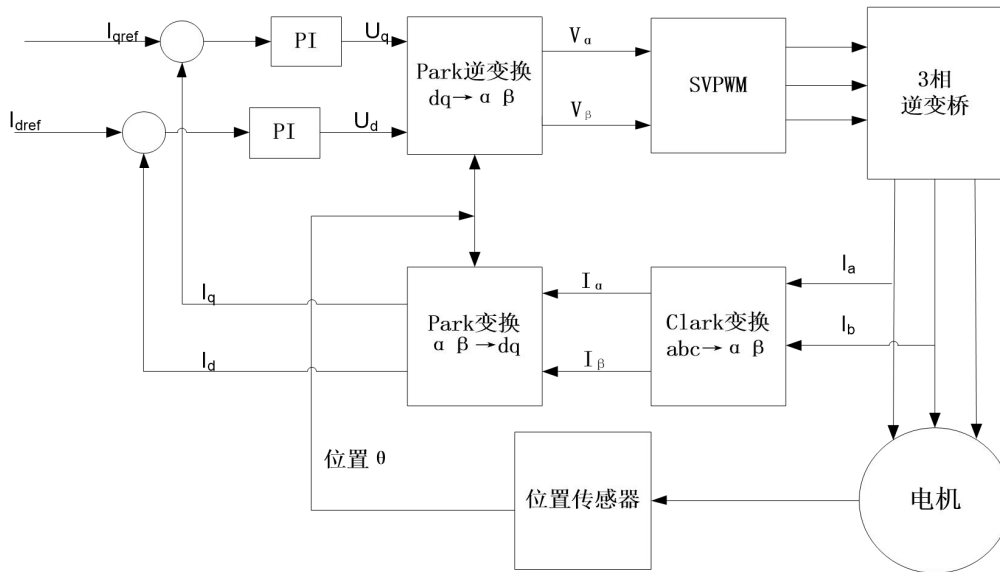


图 2-1 有感 FOC 控制框图

结合有感 OFC 控制框图，系统的控制过程分析如下：

- 1、 测量电机三相定子电流，可得到 I_a 和 I_b 。将三相电流通过 Clark 变换至两相电流 I_α 和 I_β ，其是相互正交的时变电流值。
- 2、 按照控制环上一次迭代计算出的电机角度，通过 Park 变换得到旋转坐标系下相互正交的电流 I_d 和 I_q 。在稳态条件下， I_d 和 I_q 是常量。
- 3、 I_d 的参考值控制转子磁通， I_q 的参考值控制电机的转矩输出，其各自的实际值与参考值进行比较得到电流环 PI 控制器的输入。调节 PI 控制器的参数，得到 V_d 和 V_q ，即要施加到电机上的电压矢量。
- 4、 通过位置传感器得到新的电机位置，从而得到新的角度和转速。新的电机角度可告知 FOC 算法下一个电压矢量在何处。
- 5、 通过使用新的电机角度， V_d 和 V_q 经过 Park 逆变换逆变到两相静止坐标系上。该计算将产生下一个正交电压值 V_α 、 V_β 。再采用 SVPWM 算法判定其合成的电压矢量位于哪个扇区，计算出三相各桥臂开关管的导通时间。最后经过三相逆变器驱动模块输出电机所需的三相电压。

3. 有感 FOC 程序调试

3.1. 电机参数配置

➤ 电机参数

有感 FOC 控制，电机参数中，只有极对数有用到计算速度，其电阻，电感以及反电动势相关参数均没用到。所以，电机参数只需要填极对数即可。

速度计算公式：电机运行转速(RPM) = 电机运行电频率(Hz) * 60 /电机极对数（对极）

```
#define Pole_Pairs      (10)           //极对数
#define RS              (0.1)         //相电阻, ohm    (用万用表测量)
#define LD              (0.000080)   //D 轴相电感,H (用电桥测量 频率 1KHZ)
#define LQ              (0.000080)   //Q 轴相电感,H(用电桥测量 频率 1KHZ)
#define Ke_Vpp         (9.4)         //反电动势测量的峰峰值, V
#define Ke_T           (15.0)        //反电动势测量的周期, ms
```

3.2. 硬件参数配置

➤ 电流基准电路参数

```
#define HW_RSHUNT      (0.002)       // 采样电阻, ohm
#define HW_AMPGAIN     (6)           // 运放放大倍数
#define HW_ADC_REF     (5.0)        // ADC 参考电压
```

HW_RSHUNT 是采样电阻阻值。电阻选取需要考虑电机运行最大的电流，防止因为电流过大而烧毁电阻。

HW_AMPGAIN 是运放放大倍数，其选取需要考虑最大运行电流 I_{max} 。

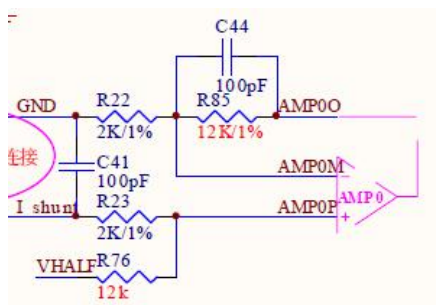


图 3-2-1 运放放大电路图

如没有电流采样基准 V_{half} ，则 $HW_RSHUNT * I_{max} * HW_AMPGAIN < HW_ADC_REF$ 。

如有电流采样基准 V_{half} ，则 $HW_RSHUNT * I_{max} * HW_AMPGAIN < (HW_ADC_REF - V_{half})$ 。

➤ 母线电压采样分压电路参数

```
#define RV1          (30.0)          //单位, K 欧姆
#define RV2          (0.0)          //单位, K 欧姆
#define RV3          (1.5)          //单位, K 欧姆
```

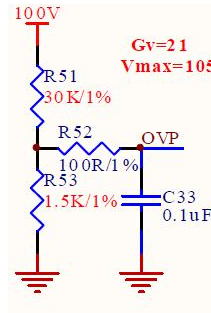


图 3-2-2 母线电压分压电路

根据实际电路填写，选取电阻时，需要考虑阻值功耗以及分压后的电压大小。分压后电压 V 需要小于 5V。

3.3. 软件参数配置 (配置文件: customer.h)

3.3.1. Hall 信号配置

Hall 上升沿与反电动势过零点电角度关系：

当 Hall 上升沿超前反电动势过零点 30 度则配置如下：

```
#define HallAndBEMF          (HallAdvanceBEMF30Degree)
```

当 Hall 上升沿与反电动势过零点重合则配置如下：

```
#define HallAndBEMF          (HallSameWithBEMF)
```

当 Hall 上升沿滞后反电动势过零点 30 度则配置如下：

```
#define HallAndBEMF          (HallDelayBEMF30Degree)
```

```
#define HallStartAngleOffset (20.0)          //启动角度, 只有第一拍有效, 会影响启动动力矩, 范围 0 度~60 度
```

➤ Hall 与反电动势对应关系如下图所示：

如下图 Hall 信号与反电动势关系，此时设置

`#define HallAndBEMF (HallAdvanceBEMF30Degree)`

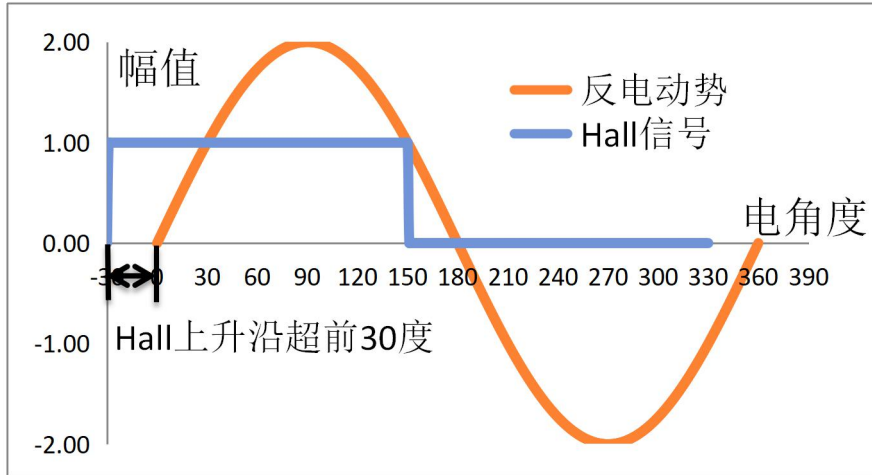


图 3-3-1 Hall 信号超前 BEMF 30 度

如下图 Hall 信号与反电动势关系，此时设置

`#define HallAndBEMF (HallSameWithBEMF)`

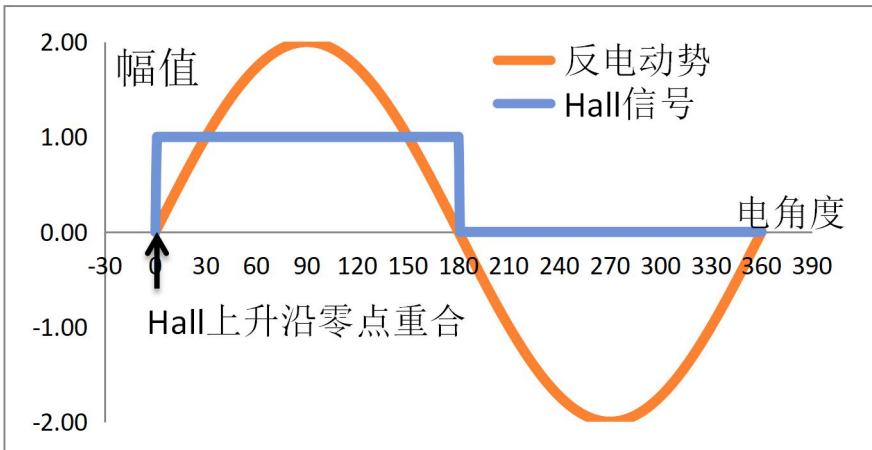


图 3-3-2 Hall 信号与 BEMF 重合

如下图 Hall 信号与反电动势关系，此时设置

```
#define HallAndBEMF (HallDelayBEMF30Degree)
```

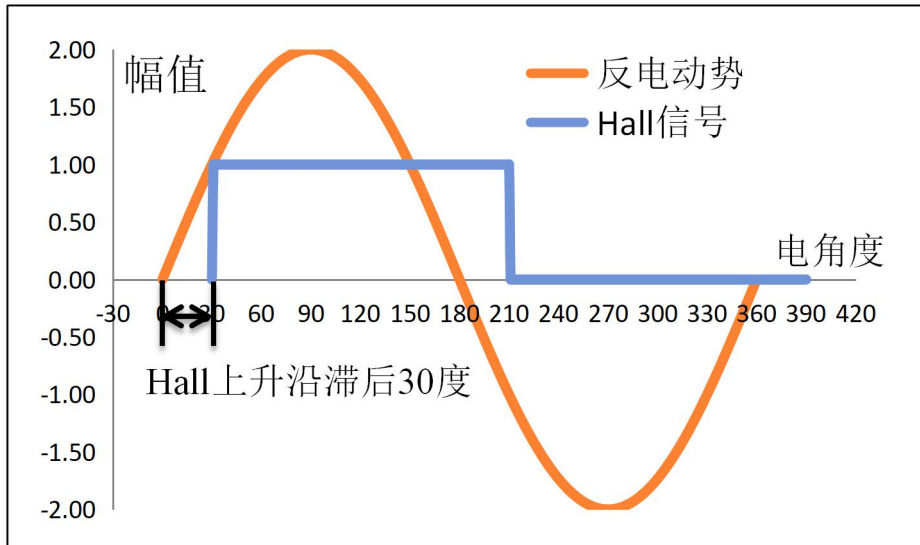


图 3-3-3 Hall 信号与滞后 BEMF 30 度

如不想通过测试 hall 与反电动势关系来确定电机相序，可以通过固定 hall 模式，换电机线序的方式来确定。如设置 `#define HallAndBEMF (HallAdvanceBEMF30Degree)`，最多换 6 次电机线，就能对应出正确的线序，否则换成 `#define HallAndBEMF (HallDelayBEMF30Degree)` 或者 `#define HallAndBEMF (HallSameWithBEMF)`，分别再换 6 次。以往经验，只需要在“hall 超前反电势 30 度”或者“hall 滞后反电动势 30 度”里找相序就可以，即最多换 12 次电机线就能找到正确的线序。

3.3.2. 基本功能配置

➤ 芯片运行频率

```
#define MCU_CLOCK (24.0)
```

程序运行的是芯片内部快时钟 24M。

➤ 载波频率

```
#define PWM_FREQUENCY (20.0)
```

MOSFET 驱动信号载波频率设置为 20kHz。载波频率调制范围建议 10K-25K。

➤ 死区时间

```
#define PWM_DEADTIME (1.0)
```

死区时间设置为 1us。实际根据 MOSFET 的开关时间设置。

➤ 单电阻最小采样窗口

```
#define MIN_WIND_TIME          (1.8)
```

最小采样窗口设置为 1.8us。采样点设置的时间小于最小采样窗口时间减去死区时间。TRGDLY 寄存器设置采样点是超前还是延迟。比如 MCU 时钟 24M (41.67ns)，TRGDLY=5,则延迟 41.67*5 =208ns。TRGDLY=-5，则提前 208ns。

➤ 驱动模式

```
#define Driver_6N              (0)
#define Driver_3P3N           (1)
#define DriverMode             (Driver_6N)
```

FU6818 选择 6N 输出,FU6811 选择 6N 模式，FU6831 选择 3P3N 模式。

➤ 高低电平有效模式

```
#define High_Level             (0)
#define Low_Level              (1)
#define PWM_Level_Mode         (High_Level) //选择高电平有效
```

FU6818 选择高电平,FU6811 根据外接的预驱动来选择，FU6831 选择高电平。

➤ 旋转方向设定

```
#define CW                      (0)          //正向
#define CCW                     (1)          //反向
#define FRSet                   (CCW)        //选定反向
```

➤ 速度基准设置

```
#define MOTOR_SPEED_BASE       (8000.0)      // 速度基准，需要设置成最高运行转速的 2 倍以上。单位：RPM
```

速度基准，用于把标么化的速度转化为 RPM 为单位的速度。

如速度 = mcHall.HallSpeed * MOTOR_SPEED_BASE / 32767。

3.3.3. 电流参数配置

➤ 电流采样方案选择

```
#define Single_Resistor         (0)          // 单电阻
#define Double_Resistor        (1)          // 双电阻
#define Shunt_Resistor_Mode     (Single_Resistor) // 选择单电阻
```

➤ 运行电流设置

```
#define ID_Start_CURRENT        I_Value(0.0) // D 轴启动电流
#define IQ_Start_CURRENT        I_Value(1.0) // Q 轴启动电流
#define IQ_RUN_MIN_CURRENT      I_Value(0.1) // 最小运行电流
#define IQ_RUN_MAX_CURRENT      I_Value(43.0) // 最大运行电流，峰值
```

ID_Start_CURRENT D 轴启动电流，设置为 0，整个运行过程 IDref =0，实现最大转矩控制。

IQ_Start_CURRENT Q 轴启动电流，根据实际启动过程调试，电流小，启动缓慢，电流大，启动快速。

IQ_RUN_MIN_CURRENT Q 轴运行最小电流，即启动完后，运行过程的最小 IQ 电流。

IQ_RUN_MAX_CURRENT Q 轴运行的最大电流，根据实际电机运行电流填写。

➤ 均值电流限流

当 RMSCurrentLimitEnable 为 1 时使能限流功能，程序中 TIM3 溢出中断没触发一次，则运行一次限流判断并对输出电压进行调整。参数配置如下：

```
#define RMSCurrentLimitEnable      (0)           //使能限流功能，0 不使能，1 使能
#define IQ_RUN_MAX_RMS_CURRENT    I_Value(8.0) //限流参考值 0，单位 A
```

使用限流功能时外围电路需要如下配置：

AMP0 运放(电流采样)输出经过 10K 10nF/16V RC 滤波后输入至管脚 P34 ADC7，此时 ADC7 采样为平均电流，程序中变量为 mcFocCtrl.mcADCCurrentbus，AMP0 输出滤波电路如下图。(DEMO 板没有这部分电路，需要外接)

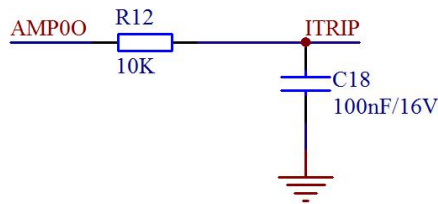


图 3-3-1 均值限流采样电路

➤ FOC 电流环参数设置

✓ D 轴电流环参数

```
#define DKP          _Q12(0.8)           //D 轴电流环 KP
#define DKI          _Q12(0.04)          //D 轴电流 KI
#define DOUTMAX      _Q15(0.19)          //D 轴电流环最大输出
#define DOUTMIN      _Q15(-0.19)         //D 轴电流环最小输出
```

✓ Q 轴电流环参数

```
#define DKP          _Q12(0.8)           //D 轴电流环 KP
#define DKI          _Q12(0.04)          //D 轴电流 KI
#define DOUTMAX      _Q15(0.93)          //D 轴电流环最大输出
#define DOUTMIN      _Q15(-0.93)         //D 轴电流环最小输出
```

一般 DQ 轴电流环的 KPKI 参数不需要调整。

单电阻方案时，D 轴最大输出和 Q 最大输出需要满足 $((DOUTMAX/32768) \text{的平方} + (QOUTMAX/32768) \text{的平方}) \text{取根号要} < 0.97$ 。

双电阻方案时，D 轴最大输出和 Q 最大输出需要满足 $((DOUTMAX/32768)$ 的平方 + $(QOUTMAX/32768)$ 的平方) 取根号要 <0.95 。

3.3.4. 其他功能配置

➤ 油门和刹车保护功能

```
#define mcSpeedVSPBrakeProtectEnable (0) //调速和刹车电压保护使能，0 不使能，1 使能。
```

当油门或者刹车的电压分别大于 ProtecPWMDuty 或者 BrakeProtecPWMDuty，则关闭输出。

➤ 油门调速参数

```
#define OFFPWMDuty (0.05) //关机电平，当调速电平小于 (HW_ADC_REF*OFFPWMDuty) 关机
```

```
#define ONPWMDuty (0.20) //开机电平，当调速电平大于 (HW_ADC_REF*ONPWMDuty) 开机
```

```
#define MAXPWMDuty (0.90) //最大电平，当调速电平大于 (HW_ADC_REF*MAXPWMDuty) 维持最大输出
```

```
#define ProtecPWMDuty (1.0) //保护电平，当调速电平大于 (HW_ADC_REF*ProtecPWMDuty) 关闭输出
```

➤ 刹车调速参数

```
#define MCBrakeEnable (1) //刹车功能使能
```

```
#define BrakeOFFPWMDuty (0.28) //关闭刹车，当刹车电平小于 (HW_ADC_REF*BrakeOFFPWMDuty) 关闭刹车
```

```
#define BrakeONPWMDuty (0.50) //开启刹车，当刹车电平大于 (HW_ADC_REF*BrakeONPWMDuty) 开启刹车
```

```
#define BrakeMAXPWMDuty (1.0) //刹车的最大有效电平，当刹车电平大于 (HW_ADC_REF*BrakeMAXPWMDuty) 维持最大刹车力度
```

```
#define BrakeProtecPWMDuty (1.0) //刹车保护电平，当刹车电平大于 (HW_ADC_REF*BrakeProtecPWMDuty) 关闭输出
```

➤ 倒车模式

```
#define FRControlEnable (1) //倒车功能使能，需要配置 IO 口，0 不使能，1 使能
```

```
#define RunningReverse (0) //延迟运行倒车，即触发倒车，油门不需要复位，车停即可倒车，速度需要低于阈值
```

```
#define RunningReverseRNow (1) //立即运行倒车，即触发倒车，油门不需要复位，车停即可倒车，与速度无关
```

```
#define StopTORReverse (2) //停止倒车，即触发倒车，油门需要复位，并且车停止才能倒车
```

```
#define ReverseMode          (RunningReverse) //倒车模式选择延迟运行倒车
```

➤ **巡航控制**

巡航控制，当调速信号固定不变的时间大于或者等于时间 `mcCruiseControlCntTime` ，则维持当前输出不变，直到油门复位再调速。

```
#define mcCruiseControlEnable    (0)          //巡航使能  
#define mcCruiseControlCntTime  (5.0)       //巡航判断时间, s
```

➤ 关机可配置

MotorStopMode 选择自由停机 FreeStop、刹车停机 BrakeStop。选择自由停机 FreeStop 模式时关机后先减 IQ 电流，直至达到阈值，再关闭输出；选择刹车停机 BrakeStop 模式时关机后，先减 IQ 电流，直至达到阈值，关闭上桥，同时开通下桥刹车。

```
#define FreeStop          (0)          //自由停机
#define BrakeStop        (1)          //刹车停机
#define MotorStopMode    (FreeStop)  //停机模式选择
```

➤ 外环选择

```
#define IQ_Loop_Enable    (0)          // IQ 内环控制
#define Voltage_Open_Loop_Enable (1)    // 电压环
#define Speed_Loop_Mode  (Voltage_Open_Loop_Enable) // 选择电压环
```

➤ 电压环参数设置

```
#define VKP               _Q12(0.1)    // KP
#define VKI               _Q12(0.01)   // KI
#define VOUTMAX           IQ_RUN_MAX_CURRENT // PI 最大输出电流
#define VOUTMIN           I_Value(0.15) // PI 最小输出电流
#define SPEEDRampTIME     25           //电压爬坡周期,1->1ms
#define Voltage_INC       100          //电压爬坡增量
#define Voltage_DEC       100          //电压爬坡减量
```

➤ 过/欠压保护

```
#define VoltageProtectEnable (1)        //电压保护，0 不使能，1 使能
#define Over_Protect_Voltage (80.0)     //过压保护，单位 V
#define Over_Recover_Voltage (40.0)     //过压保护恢复值，单位 V
#define Under_Protect_Voltage (20.0)    //欠压保护，单位 V
#define Under_Recover_Voltage (22.0)    //欠压保护恢复值，单位 V
#define Voltage_Detect_Time1 (20)       //第一次欠压，过压后再触发欠压或过压
得检测时间，单位，5ms
#define Voltage_Detect_Time2 (400)      //第一次欠压检测时间，单位，5ms
```

➤ 堵转参数

```
#define StallProtectEnable (1)          //堵转保护使能，0 不使能，1 不使能
#define StallDetectTime    (2000)      //堵转检测时间，单位 ms
```

4. 程序结构说明

4.1. 电机运行状态机

如下表为 main()函数中电机控制状态机 MC_Control()函数各状态说明

序号	程序执行状态	函数说明
0	mcInIt	程序参数初始化、电机运行方向初始化、闭环运算参数初始化
1	mcCharge	MOSFET 预充电控制,三相下桥依次打开,使用 FU6811 和 FU6818 时需要在 customer.h 中设置足够的 PrechargeTime 预充电时间。一般低压应用可以不用预充电,比如车类,如果使能预充电,会造成顺风启动刹车效果。
4	mcStart	电机启动控制
5	mcRun	电机运行状态
6	mcStop	电机关机控制,包括自有停机、刹车停机
8	mcFault	驱动异常处理,包括驱动关闭输出、减速输出及自动重启控制

表 4-4-1

4.2. 中断函数

void CMP_ISR(void) interrupt 7

- 硬件过流保护

void FOC_INT(void) interrupt 3

包含如下函数或功能:

- 峰值限流
- 刹车处理

void TIM0_INT(void) interrupt 4

- Hall 换相处理
- 角度限制处理

void TIM23_ISR(void) interrupt 9

包含如下函数或功能:

- 平均限流处理
- 倒车处理
- 巡航时间累计

void TIM45_ISR(void) interrupt 10

TIM4 1ms 溢出中断，周期执行，包含以下功能：

- ADC 采样
- 欠压保护、启动保护、缺相、堵转等保护处理
- 开关机处理
- 调速处理
- 速度跟随处理
- 巡航条件判断

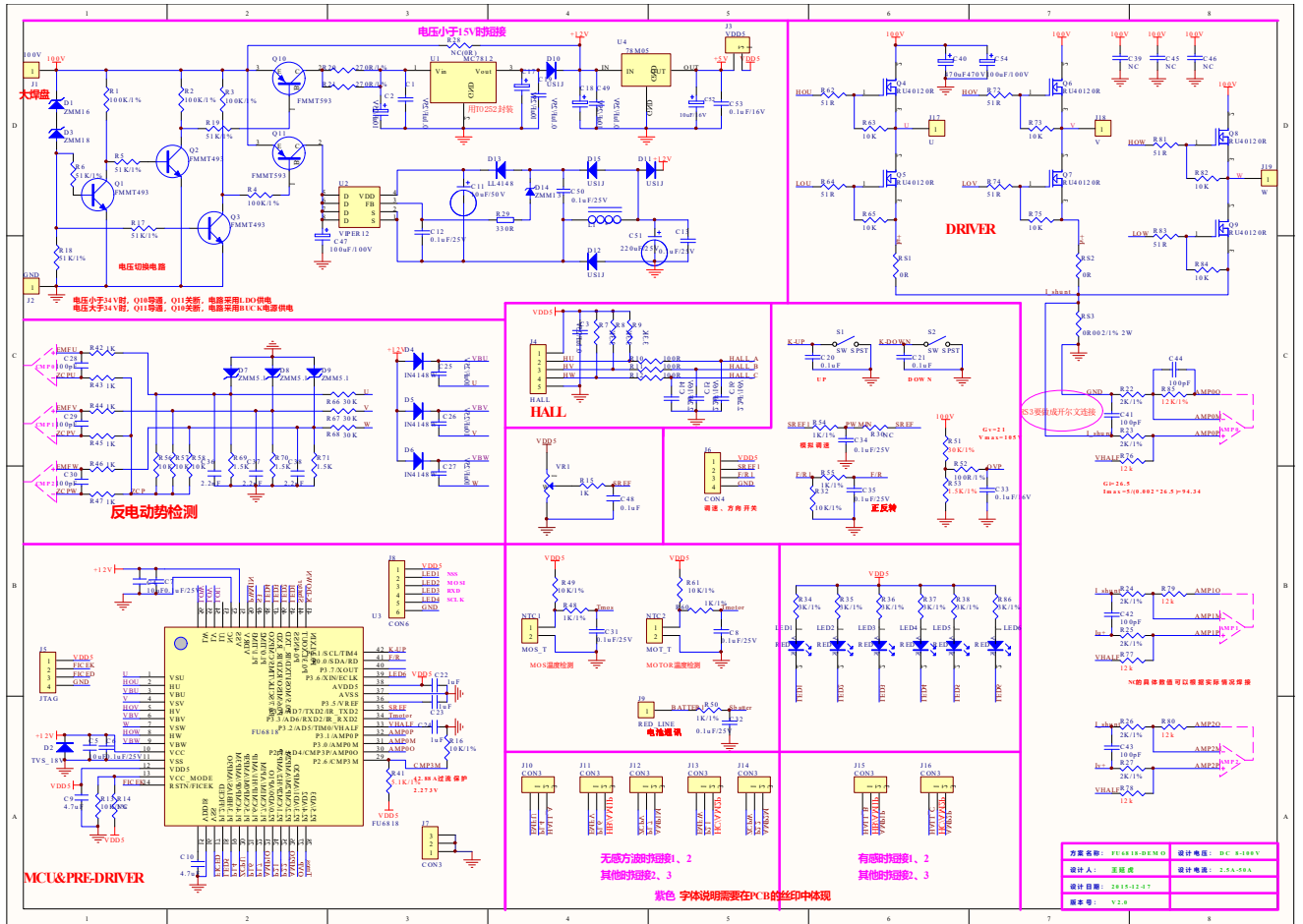
void UART_ISR(void) interrupt 12

UART 发送/接收中断处理

FU6818 DEMO 中压板应用于有感 FOC 单电阻方案需要修改的硬件参数

- 接口 J10、J11、J12、J13、J14、的 2 脚和 3 脚短接；
- 接口 J15、J16 的 1 脚和 2 脚短接；
- 去掉 C44；
- R85 = R76 = 12K ; R41 = 5.1K；
- C14 = C15 = C16 = 10nf；

5. 附录 1: 原理图



版权说明

该版权由峰昭科技（深圳）有限公司所有。为提高产品性能，峰昭科技有权对产品电路、标准、软件等做出修改。本手册中信息是为广大使用客户提供的。客户应确保采取适当措施以便其在使用我们产品过程中不侵犯任何专利。峰昭科技（深圳）有限公司原则上尊重有效的第三方专利权，不侵犯或协助他人侵犯这些权利。本手册版权由峰昭科技（深圳）有限公司所有。未经峰昭科技（深圳）有限公司书面许可，任何单位及个人不得以任何方式或理由对该出版物的任何部分进行复制、传播、修改、抄录或储存在检索系统中，或翻译成任何语言。凡侵犯本公司版权的，我司必依法追究其法律责任。

峰昭科技（深圳）有限公司

地址：深圳市南山区科技中二路深圳软件园2期（11栋）203

电话：0755-86181158

传真：0755-26867715

网址：www.fortiortech.com