

STM32 Three Hall PWM of BLDC

直流无刷三霍尔方波速度闭环控制

硕历电子

USER

文档名称	STM32 Three Hall PWM of BLDC
文档版本	V17_3
制作日期	2017.2. 5
项目名称	

文档历史管理:

版本	日期	修改人	修改说明
V17_1	2017.2. 5	徐影	First issued
V17_2	2017.4.7	徐影	Second issued
V17_3	2017.9.8	徐影	Third issued

硕历电子 FOR USER

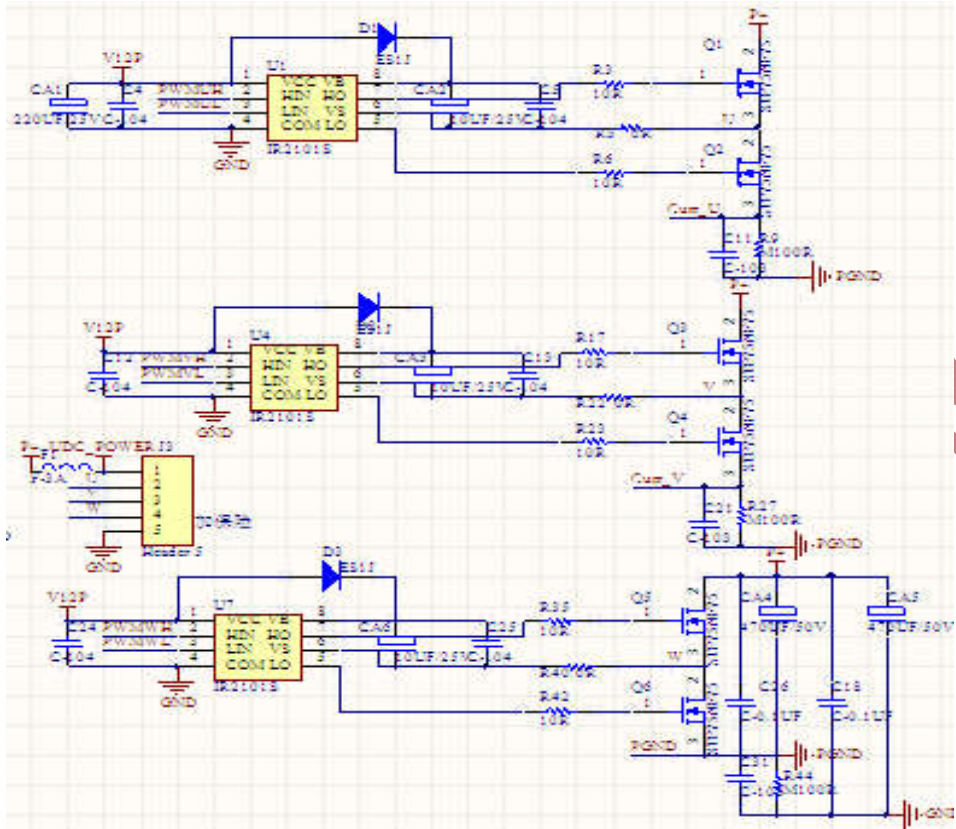
目 录

1.直流无刷电机（三霍尔）控制原理图	1 -
1.1 三相电机主功率驱动电路	1 -
1.2 电源电路	1 -
1.3 三个霍尔接口电路	2 -
1.4 速度控制接口电路.....	2 -
2.直流无刷电机原理	2 -
3 直流无刷换相控制原理	3 -
4. STM32 的电机数字控制技术	7 -
5. STM32 的软件架构	8 -
5.1 STM32 电机控制软件系统	8 -
5.2 数字 PID 算法	9 -
5.3BLDC 霍尔传感器控制程序函数说明.....	10 -
6. 重点核心程序解释	12 -
6.1 电机启动和停止	12 -
6.2 PWM 输出	12 -
6.3 调试参数测试	13 -

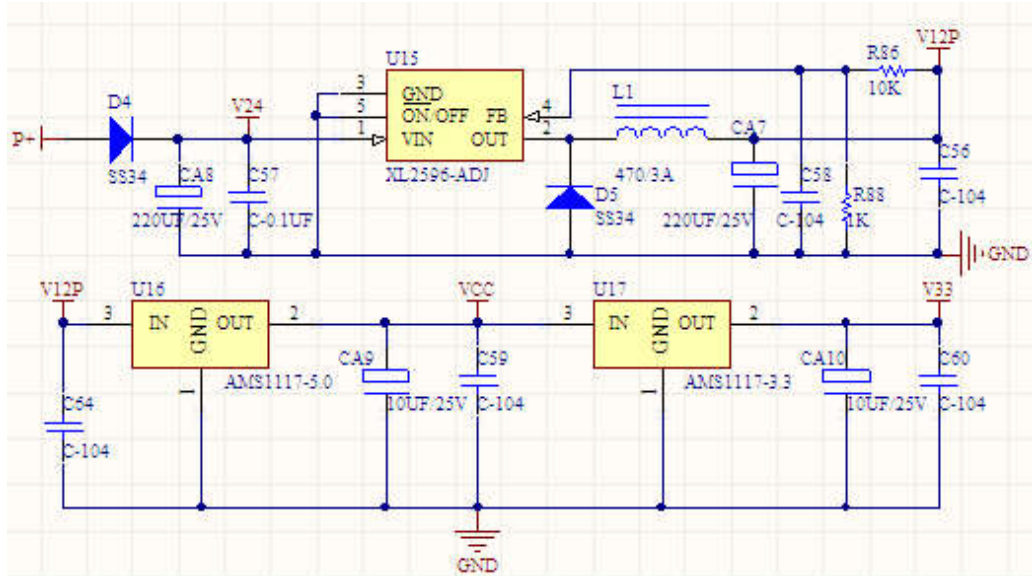
硕 历 电 子 FOR USER

1.直流无刷电机（三霍尔）控制原理图

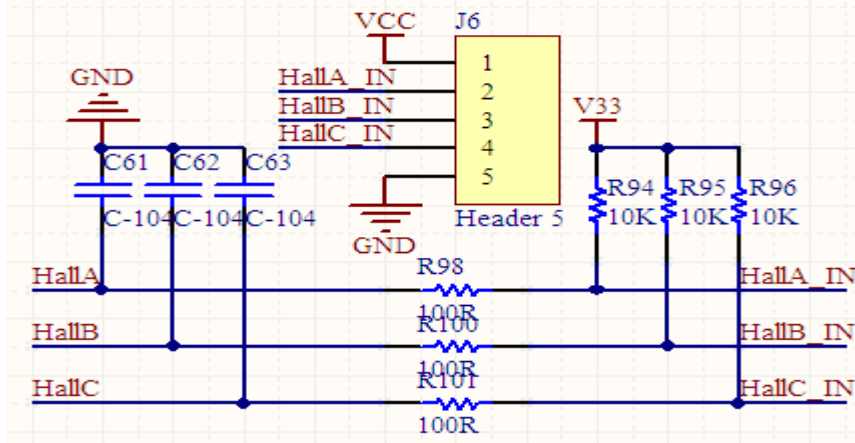
1.1 三相电机主功率驱动电路



1.2 电源电路

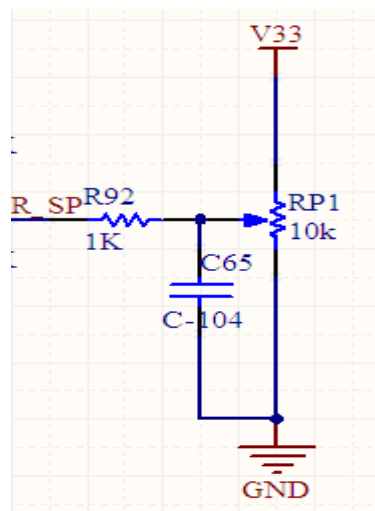


1.3 三个霍尔接口电路



霍尔线：电源地 HallA HallB HallC

1.4 速度控制接口电路



如实物所示电位器的旋转方向按照箭头所示增加转速：RP1 为速度输出给定接口，顺时针旋转速度增大，逆时针旋转到最右边则电机关闭，顺时针旋转电位器输出电压 0.15V 左右电机开始启动。

2.直流无刷电机原理

利用基尔霍夫电压定律 (KVL) 可以得到三相绕组的电压平衡方程：

$$\begin{pmatrix} u_A \\ u_B \\ u_C \end{pmatrix} = \begin{pmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{pmatrix} \begin{pmatrix} i_A \\ i_B \\ i_C \end{pmatrix} + \begin{pmatrix} L-M & & \\ & L-M & \\ & & L-M \end{pmatrix} p \begin{pmatrix} i_A \\ i_B \\ i_C \end{pmatrix} + \begin{pmatrix} e_A \\ e_B \\ e_C \end{pmatrix}$$

其中： u_A 、 u_B 、 u_C 为定子相绕组电压 (V) ；

i_A 、 i_B 、 i_C 为定子相绕组电流 (A) ；

e_A 、 e_B 、 e_C 为定子相绕组电动势 (V) ；

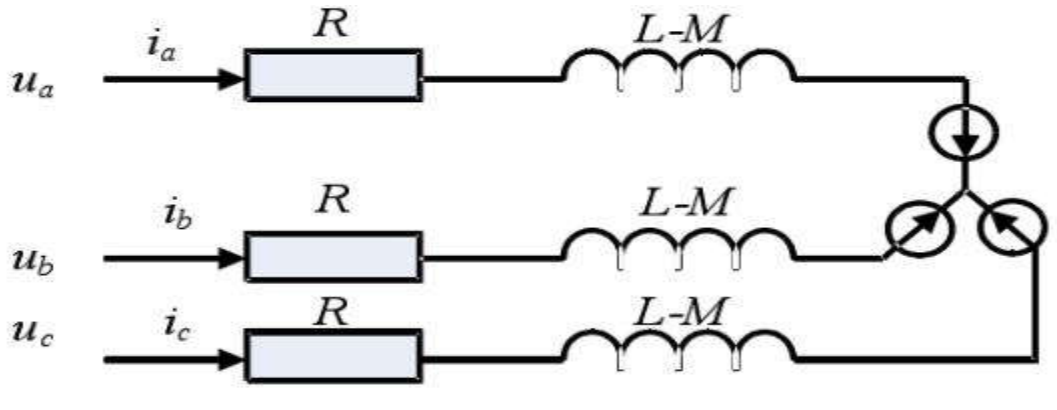
R 为电机相电阻 ；

L 为每相绕组的自感 (H) ；

M 为每两相绕组间的互感 (H) ；

p 为微分算子， $p=d/dt$ 。

由绕组电压方程可知，无刷直流电机的等效电路如图：



(1) 电磁转矩方程

无刷直流电机的定子绕组输入功率与产生的电磁转矩关系为：

$$e_A i_A + e_B i_B + e_C i_C = T \omega$$

故无刷直流电机的电磁转矩为：

$$T = (e_A i_A + e_B i_B + e_C i_C) / \omega$$

其中， ω 为电机的机械角速度 (rad/s)。无刷直流电机的电磁转矩是由定子绕组中的电流与转子磁钢产生的磁场相互作用而产生的，由上式可见，电机电磁转矩与磁场和相电流成正比，与转动角速度成反比。

(2) 电机运动方程

无刷直流电机的运动方程为：

$$T - T_L = J \frac{d\omega}{dt}$$

式中， T_L 为负载转矩， J 为电机转动惯量。

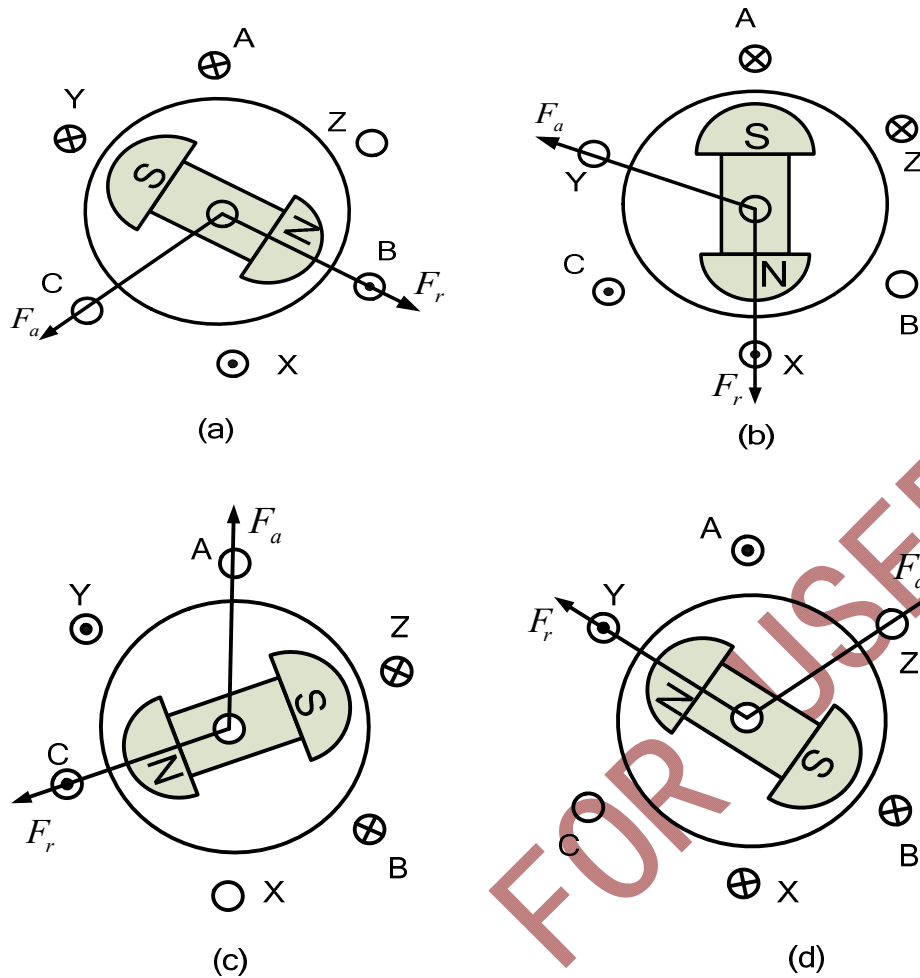
3 直流无刷换相控制原理

永磁无刷直流电机最常用的主电路为星形连接三相桥式主电路（图（c）），这种电路主要有两种导通方式：二二导通方式和三三导通方式。

二二导通方式是指每一瞬间有两个功率管导通，每隔 1/6 周期（60°电角度）换相一次，每次换相一个功率管，桥臂之间左右互换，每个功率管导通 120°电角度。

三三导通方式是在任一瞬间使三个开关管同时导通，同样每隔 60°电角度换相一次，每次换相一个功率管，但换相发生在同一桥臂上下管之间，因而每个功率管导通 180°电角度。

相比较之下，二二导通方式电机出力大，转矩特性较好，且不会发生开关管直通短路的现象，因此最为常用。其工作原理如图所示。



永磁无刷直流电动机工作原理示意图

- (a) VT1,VT2 导通, A,B 相导通 (b) VT1,VT2 导通, A,C 相导通
 (c) VT2,VT3 导通, B,C 相导通 (d) VT3,VT4 导通, B,A 相导通

当转子旋转到图 (a) 所示的位置时, 位置检测装置输出的转子位置信号经过控制电路逻辑变换后驱动逆变器, 使功率开关 VT1、VT6 导通, A、B 两相绕组通电, 电流从电源的正极流出, 经 VT1 流入 A 相绕组, 再从 B 相绕组流出, 经 VT6 回到电源的负极。电枢绕组在空间产生的磁动势 F_a 如图 (a) 所示, 此时定转子磁场相互作用, 使电机的转子顺时针转动。

当转子在空间转过 60° 电角度, 到达图 (b) 所示位置时, 位置检测器输出的位置信号使 VT1、VT2 导通, A、C 两相绕组通电, 电流从电源的正极流出, 经 VT1 流入 A 相绕组, 再从 C 相绕组流出, 经 VT2 回到电源的负极。电枢绕组在空间产生的磁动势 F_a 如图 (b) 所示, 此时定转子磁场相互作用, 转矩方向不变, 使电机的转子继续顺时针转动。

转子在空间每转过 60° 电角度, 逆变器开关就发生一次切换, 功率开关的导通逻辑为 VT1、VT6→VT1、VT2→VT3、VT2→VT3、VT4→VT5、VT4→VT5、VT6→VT1、VT6。在此期间, 转子始终受到顺时针方向的电磁转矩作用, 沿顺时针方向连续旋转。

在图 (a) 到图 (b) 的 60° 电角度范围内, 转子磁场沿顺时针连续旋转, 而定子合成磁场在空间保持静止。只有当转子转过 60° 电角度到达图 (b) 所示位置时, 定子合成磁场才

从图 (a) 的 Fa 位置跳跃到图 (b) 的 Fa 位置。可见，定子合成磁场在空间不是连续旋转的，而是一种跳跃式的旋转磁场，每个步进角是 60° 电角度。转子在空间每转过 60° 电角度，定子绕组就进行一次换流，定子合成磁场的状态就发生一次跃变。可见，电机有六种磁状态，每一状态有两相导通，每相绕组导通的时间对应着转子旋转 120° 电角度。永磁无刷直流电动机的这种工作方式称为两相导通星形三相六状态，这是无刷直流电动机最常用的一种工作方式。

从永磁无刷直流电动机的工作原理可知，转子位置的正确测量对电动机的正常运行起着十分重要的作用，这与一般的永磁同步电动机是不相同的。

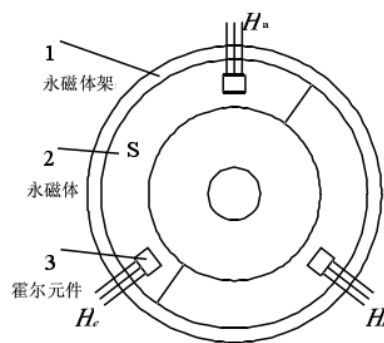
霍尔元件式位置传感器

霍尔元件式位置传感器是磁敏式位置传感器的一种。它是一种半导体器件，是利用霍尔效应制成的。当霍尔元件按要求通以电流并置于外磁场中，即输出霍尔电势信号，当其不受外磁场作用时，其输出端无信号。用霍尔元件作转子位置传感器通常有两种方式。第一种方式是将霍尔元件粘贴于电机

端盖内表面，靠近霍尔元件并与之有一小间隙处，安装在与电机轴同轴的永磁体，如下图所示。对于两相导通星形三相六状态无刷直流电机，三个霍尔元件在空间彼此相隔 120° 电角度，永磁体的极弧宽度为 180° 电角度。这样，当电机转子旋转时，三个霍尔元件便交替输出三个宽度为 180° 电角、相位互差 120° 电角的矩形波信号。

第二种方式是直接将霍尔元件敷贴在定子电枢铁心气隙表面或绕组端部紧靠铁心处，利用电机转子上的永磁体主极作为传感器的永磁体，根据霍尔元件的输出信号即可判断转子磁极位置，将信号放大处理后便可驱动逆变器工作。

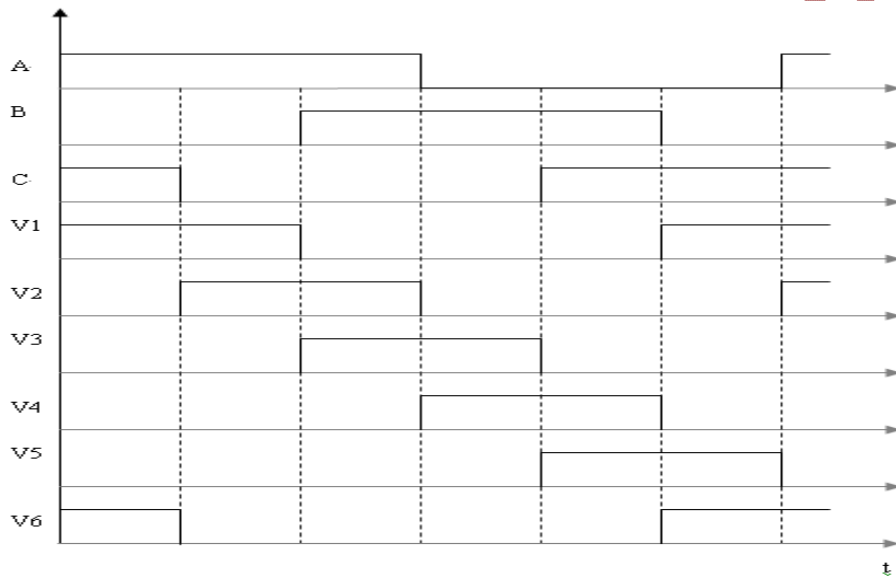
如图所示，霍尔元件式位置传感器结构简单、体积小、价格低、可靠，但对工作温度有一定要求，同时霍尔元件应靠近传感器的永磁体，否则输出信号电平太低，不能正常工作。因此，在对性能和环境要求不是很高的永磁无刷直流电机应用场合大量使用霍尔元件式位置传感器。



霍尔元件式位置传感器结构

无刷直流电动机的位置一般采用三个在空间上相隔 120° 电角度的霍尔位置传感器进行检测，当位于霍尔传感器位置处的磁场极性发生变化时，传感器的输出电平将发生改变，

由于三个霍尔传感器位检测元件的位置在空间上各差 120 电角度，因此从这三个检测元件输出端可以获得三个在时间上互差 120 度、宽度为 180 度的电平信号，分别用 A、B、C 来表示，如图所示，以信号 A 为例，A 相位置宽度为 180 电角度：在 0-60 度，T1 必须导通，故 T1 状态为 1，而 C 相还剩下 60 度通电宽度，所以此段时间为 T1 和 T6 等于 1，（此时下部可供导通的管子为 T4、T6 和 T2，而为避免桥臂直通，T4 不能导通；T2 的导通时间未到，故只能是 T6 导通）；而在 60 度—120 度，此时只有 A 相通电，B 和 C 相处于非导电期，故导通的开关管为 T1 和 T2（T1 和 T2 等于 1），其中 T2 是为 B 相导电作准备；而在 120 度—180 度时，由于每一相只有 120 电角度导电时间，故此时 T1 关断（T1=0），T2 仍然导通（B 相开始进入导电期），此时可知，T1 关断，T5 不能开通（防止桥臂直通），则此时只能开通 T3，所以 T3 信号此时间段为 1。其他时间段的开关管导通情况与此类似。

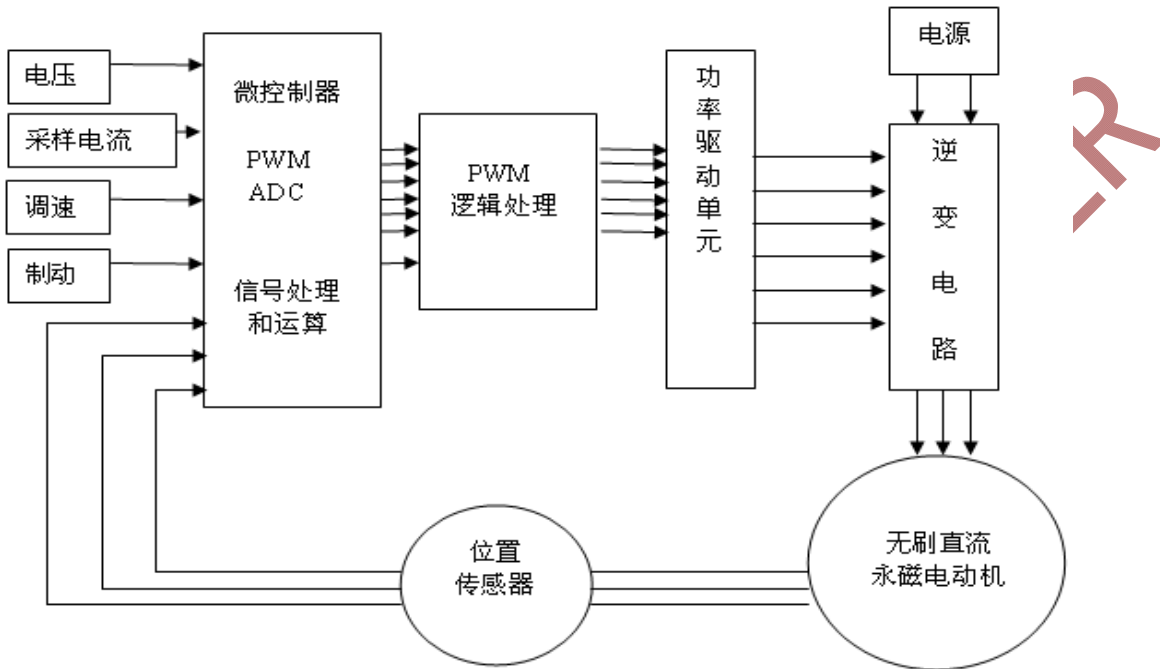


无刷电动机位置检测及开关管驱动信号

表：无刷电动机直流通电控制方式开关切换表

旋转方向	位置传感器			逆变桥开关管驱动信号					
	A	B	C	T1	T2	T3	T4	T5	T6
正转	0	0	1	0	0	0	0	1	1
	0	1	0	0	0	1	1	0	0
	0	1	1	0	0	0	1	1	0
	1	0	0	1	1	0	0	0	0
	1	0	1	1	0	0	0	0	1
	1	1	0	0	0	1	1	0	0
	0	0	1	0	1	1	0	0	0

反转	0	1	0	1	0	0	0	0	1
	0	1	1	1	1	0	0	0	0
	1	0	0	0	0	0	1	1	0
	1	0	1	0	0	1	1	0	0
	1	1	0	0	0	0	0	1	1



4. STM32 的电机数字控制技术

STM32F103 增强型系列由意法半导体集团设计，使用高性能的 ARMCortex-M332 位的 RISC 内核，工作频率为 72MHz，内置高速存储器(高达 128K 字节的闪存和 20K 字节的 SRAM)，丰富的增强 I/O 端口和联接到两条 APB 总线的外设。所有型号的器件都包含 2 个 12 位的 ADC、3 个通用 16 位定时器和一个 PWM 定时器，还包含标准和先进的通信接口：多达 2 个 I2C 和 SPI、3 个 USART、一个 USB 和一个 CAN。

STM32 作为电机控制，具有高级定时器 1 驱动三相电机，2 个 12 位高精度 AD 作为相电流采样，高达 72M 时钟可以为 FOC 提供高效运行速率。支持单周期乘法 and 高速硬件除法器，更好的是价格优势，STM32F103C8T6 以 1 美金内的价格优势。

资源如下表：

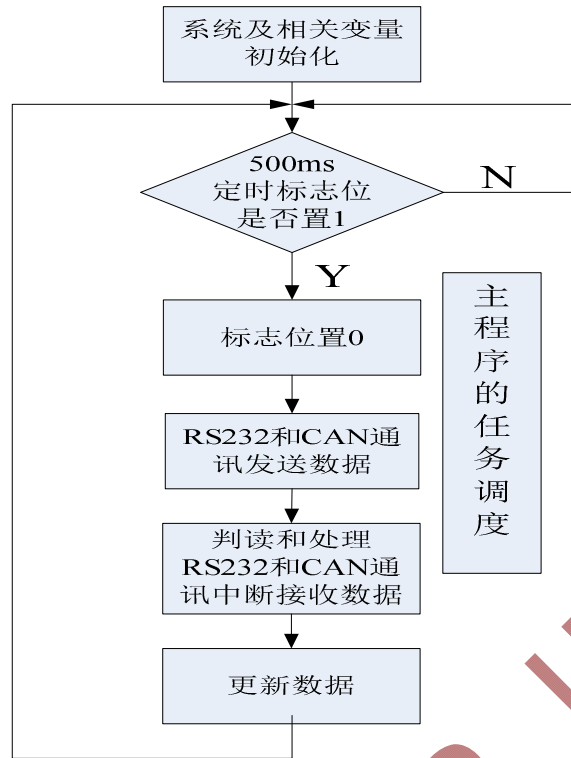
性能	内容	备注
CPU	32 位 ARM/M3 处理器	
系统频率	72MHz	
片内 FLSH	64KB	

系统外部接口 XNTF	有	
通用 IO	37 个	
AD 转换	12 位 8 个通 1us	
电机控制外设	高级定时器 1 : 6 路对称互补 PWM , 2 路独立 PWM QEP : 组正交编码器通道 (定时器)	
SPI	2 个	
USART	3 个	
CAN	1 个	
IIC	2 个	
外部中断	8 个	
USB2.0 全速接口	1 个	
DMA	7 通道	

5 . STM32 的软件架构

5.1 STM32 电机控制软件系统

整个系统由软件、硬件两大部分组成。系统软件主要功能有:转速调节、电流调节、转子换相位置计算、与上位机通讯等。STM32它集成了相当多的电机控制外围电路,这使得系统硬件设计变得非常简单。硬件部分主要包括:DSP最小系统、人机接口、整流逆变装置、电流检测和电平变换、系统保护等。如下图所示电机控制软件基本架构。程序采用的KEIL4,程序底层文件是ST的3.5.0的头文件,文件名字为STM32_BLDC_Ha111_17xxxx,工程名字为STM32_BLDC_Ha111。



RS232 和 CAN 通讯接收和发送程序，500MS 循环发送和查询是否有中断接收数据，主程序还有与 PC 通讯的子程序，SCI_RS232TX_sen()，500ms 循环发送一次数据，主要发送，当前电压 (TestPare.fact_BUS_Voil)；

电流 (TestPare.fact_BUS_Curr)；

目标转速 (TestPare.Speed_target)；

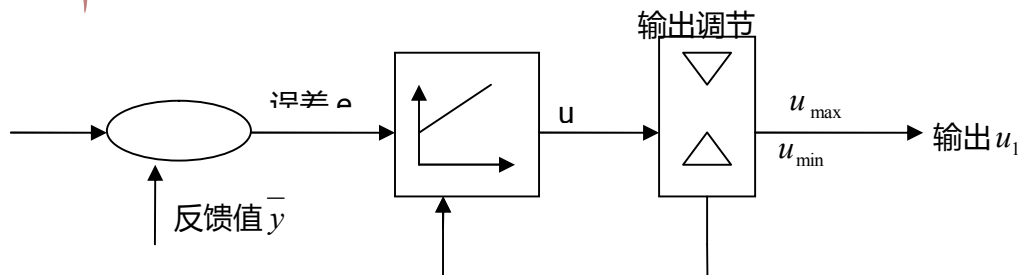
当前实际转速 (TestPare.Speed_fact)；

旋转状态 (TestPare.Run_mode)；

旋转方式 (TestPare.Start_order)；

5.2 数字 PID 算法

对于调速系统，要求每个调节器的输出都应该加限幅。为避免突加给定和工况突然变化时调节器输出值变化过于剧烈导致系统定子过压、过流，本系统应用的数字PI调节器具有输出限幅功能，结构如图所示。



PI调节器工作原理如下：

$$e(K) = y_r(K) - y(K)$$

$$u(K) = x_i(K-1) + k_p e(K)$$

$$u_l(K) = u(K)$$

$$u(K) \begin{cases} u_{\max}, u(K) > u_{\max} \\ u_l(K), u_{\min} < u(K) < u_{\max} \\ u_{\min}, u(K) < u_{\min} \end{cases}$$

$$e_l(K) = u(K) - u_l(K)$$

$$x_i(K) = x_i(K-1) + K_i e(K) + K_{cor} e_l(K)$$

其中 K_p 是比例系数, K_i 是积分系数, e_l 是 PI 调节器的限幅误差, 在计算中反馈给控制调节器积分量 x_i , 使调节器及时退饱和, $K_{cor} = K_i / K_p$.

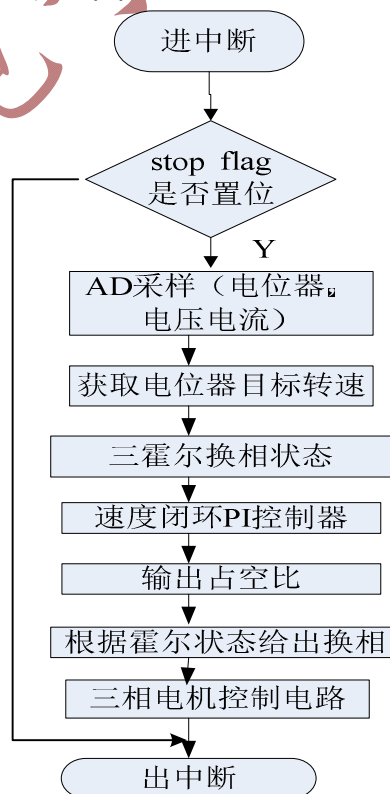
5.3 BLDC 霍尔传感器控制程序函数说明

函数名称	函数说明
SysTickConfig();	系统滴答时钟初始化函数
Delay(10000);	延时函数
GPIO_LED485RE_int();	RS485 的 IO 初始化
Init_Gpio_ADC();	ADC 采样的 IO 初始化函数
InitUSART3_Gpio();	串口 IO 的初始化
InitCAN_Gpio();	CAN 通讯 IO 的初始化
Init_PWMDAC_Gpio();	TIM4 的 PWM 作为 DAC 输出的 IO 初始化
Init_Gpio_TIM1_PWM();	高级定时器的 PWM 输出的 IO 初始化
void InitThreeHallGpio(void);	三个霍尔的 GPIO 检测初始化
Usart3_RS232_init();	串口 3 的底层初始化
CAN_Config();	CAN 通讯的底层初始化
DMA_Configuration();	配置 DMA 通道 1 作为 ADC 数据的初始化
ADC1_Configuration();	配置 ADC1 初始化
Tim1_PWM_Init();	高级定时器的 PWM 输出的初始化
TIM4PWMDAC_Config();	TIM4 的 PWM 作为 DAC 输出初始化
Offset_CurrentReading();	母线电流和相电流偏置读取和计算
VFAngle_init();	VF 自产生转子位置角度参数初始化
RunSystemer();	时间任务标志读取
CAN_Sendlen();	CAN 通讯发送函数
Uart3_RS232TX_sen();	串口 3 通讯发送函数
ReceiveData_chuli();	串口通讯接收处理函数 (1S 执行一次)
CAN_Receivechuli();	CAN 通讯接收处理函数
CLEAR_flag();	清除时间任务管理标志
ThreeHall_huanxkz();	霍尔换相控制
ADC_Sample();	ADC 采样 (电压、电流和电位器)

<code>PI_Controller((p_PI_Control)&pi_spd);</code>	转速闭环 PID 控制
<code>void MOS_Q41PWM(void)</code>	换相函数导通 Q4, Q1PWM 斩波
<code>void MOS_Q16PWM(void);</code>	换相函数导通 Q1, Q6PWM 斩波
<code>void MOS_Q63PWM(void);</code>	换相函数导通 Q6, Q3PWM 斩波
<code>void MOS_Q32PWM(void);</code>	换相函数导通 Q3, Q2PWM 斩波
<code>void MOS_Q25PWM(void);</code>	换相函数导通 Q2, Q5PWM 斩波
<code>void MOS_Q54PWM(void);</code>	换相函数导通 Q5, Q4PWM 斩波
<code>Svpwm_Outpwm();</code>	SVPWM 的输出
<code>VoidTIM1_UP_IRQHandler(void)</code>	电机控制主中断函数
<code>void USART3_IRQHandler(void);</code>	串口通讯的中断接收函数
<code>uint32_t IQSqrt(uint32_t M);</code>	IQ32 格式的开根号函数
<code>int32_t IQsat(int32_t Uint,int32_t U_max, int32_t U_min);</code>	限幅函数
<code>voidIQSin_Cos_Cale(p_IQSin_Cos pV);</code>	IQ16 格式的正余弦函数
<code>voidTIM1_BRK_IRQHandler(void);</code>	过流保护中断函数
<code>voidUSB_LP_CAN1_RX0_IRQHandler(void);</code>	CAN 通讯的中断接收函数

电机控制主中断程序:

本主中断 12.5Khz 频率，在中断内电机环路控制，速度闭环，速度换采用 80us * 25=2ms。电机控制主中断环路控制流程图如下。



```
if(logicContr.Control_Mode==1)
{
    DUTY = 2*pi_spd.Ref; // 开环}
else if(logicContr.Control_Mode==2)
{DUTY = pi_spd.OutF; // 闭环}
```

6. 重点核心程序解释

STM32 在 Keil 开发环境下进行的电机控制软件调试，测试流程和步骤见《STM32 电机控制开发板测试步骤.pdf》

6.1 电机启动和停止

通过旋转电位器，当电位器的电压值 ADCSampPare.RP_speed_Voltage 大于 200，电压值约: 0.15V,在 Task_function.C 文件里，knob_control 函数根据电位器判断启动和停止。

```
if( ADCSampPare.RP3_Voltage<=200)
```

```
//电机启动的电位器信号
```

在电位器大于 200 时，第一个扫描循环 logicContr.olddrive_car==1，之后 logicContr.olddrive_car==1 然后开启 6 个 PWM，即执行 Start_Motor() 函数

在电位器小于 200，之后 logicContr.olddrive_car==0，或者

logicContr.olddrive_car==0，此时清零占空比，同时 Stop_Motor()，关闭 6 个 PWM 口，启动电机控制需要初始化参数，三个 PI 环路参数。

6.2 PWM 输出

通过 IO 读取霍尔的状态，在根据速度 PI 输出的占空比，最终输出 PWM 驱动 MOS，控制电机。占空比为 duty 等于 0 到 6000 范围左右。

```
if ( Hall_Three.Hall_State!=Hall_Three.OldHall_State )
{ switch (Hall_Three.Hall_State )
{
    case Hall_num1: // 0x5
    { if( logicContr.Run_mode==1)
      MOS_Q63PWM();
      if( logicContr.Run_mode==2)
      MOS_Q54PWM();}
    break;
    case Hall_num2: // 0x1
    {if( logicContr.Run_mode==1)
      MOS_Q32PWM();
      if( logicContr.Run_mode==2)
      MOS_Q41PWM();}
    break;
    case Hall_num3: //0x3
    { if( logicContr.Run_mode==1)
      MOS_Q25PWM();
      if( logicContr.Run_mode==2)
      MOS_Q16PWM();}
    break;
    case Hall_num4: // 0x2
    { if( logicContr.Run_mode==1)
```



```
        MOS_Q54PWM();
        if( logicContr.Run_mode==2)
            MOS_Q63PWM();}
break;
case Hall_num5:    // 0x6
{ if( logicContr.Run_mode==1)
    MOS_Q41PWM();
    if( logicContr.Run_mode==2)
        MOS_Q32PWM();}
break;
case Hall_num6:    // 0x4
    {if( logicContr.Run_mode==1)
        MOS_Q16PWM();
        if( logicContr.Run_mode==2)
            MOS_Q25PWM();}
break;
default:
{ Stop_Motor();
    Hall_Three.Speed_RPM=0;}
break; }
```

6.3 调试参数测试

通过 PWM 输出通过 RC 滤波形成直流电压的 DAC (数字量转换模拟电压) , 用定时器 4 的第三路和第四路输出 PWM 实现。将其变量改变, 可以测试其他参数。观测参数最好在 0---2048 范围。

```
TIM_SetCompare3(TIM4 , (Hall_Three.Speed_RPMF));
TIM_SetCompare4(TIM4 , PWM_DUTY[0]);
```

在 Usart_RS232.C 的文件中 Uart3_RS232TX_sen 函数中 1S 的通讯发送一次, TestPare.Uart_txdr [8]变量的内容, 可以根据自行修改。同时客户可以根据自己需要可以连接上位机。

在电路板上丝印上部分测试点, IV_V 和 IV_U 的 VU 二相电流, I_DC 的母线电流。同时电路板上还有电源电压 V12P、V33 和 VCC 等测试点。

- ◆ 欢迎提出文档编写错误 , [投稿邮箱 616264123@qq.com](mailto:616264123@qq.com)
- ◆ 盗版举报电话 13816643017
- ◆ 电机控制交流 QQ 群 : 314306105

购买网址 :

https://item.taobao.com/item.htm?spm=2013.1.20141001.3.9b0e22bAUG3ML&id=558376790442&scm=1007.12144.81309.42296_42296&pvid=c593173d-b6dc-4476-80c7-7e80ae078e67

<https://item.taobao.com/item.htm?spm=a1z0d.6639537.1997196601.16.f939a42LP5omf&id=557626592872>

