

非常感谢购买我们套件的网友，下面是本人针对客户反映的使用 ST 马达库 V2.0 的问题解答如下,稍后时间我会再写个关于整个软件代码的运行分析给大家。

1. 你的代码与 V2.0 库的示例程序有哪些不同？有哪些新的函数定义，等等。

基本框架是一样的，主要是为了让初学者尽快上手，所以没有加过多的软件技巧，主要是为了清晰的实现 FOC+SVPWM 的算法，然后为了这个基本要求，同时也是为了以后开发工作的需要，增加了 TFT 汉字支持，图形显示，FAT 文件支持，以上是为了实现 HMI(人机界面的需要)必须的，同时对于大量数据储存的要求，增加了大容量的串行 FLASH 以及 SD 卡的支持，这是为了扩充以后对于大量输入输出数据的需要增加的。然后增加了第二编码器支持，是为了增加 POSITION(位置)模式的需要，因为现在一个基本的 SERVO，都需要 3-4 个 CONTROL LOOP(控制环)，所以我们预留了一个输入 TIMER 作为该功能扩展需要（该功能硬件资源已经准备好，限于时间软件代码没有完善，需用户自己扩充）。以上功能具体的实现函数，这里不好说，毕竟代码太庞大了，我们主要增加了 4 个源代码文件：SPI_FLASH.C、ILI932x.C、SD.C、tff.c，分别针对上面的 2 个功能模块，您可以参考，其他函数都直接做到代码库里面了。

如果想全面了解 FOC 和 SVPWM 实现细节，以及控制原理的数字化实现需要您通读代码，当然您也可以不用关注太多。主

要是理解整个程序的构架与流程，这些是最基本的，对于您以后把它移植到您自己的系统中绝对是必须要做的，细节可以大概知道就好了，关于软件基本流程我后边再写个详细的说明文档，建议您仔细阅读今年 10 月份 ST 在上海办的 2.0 版本马达库的培训班，我当时报名了，定金也都交了，但是临时有事情没有成行，所以很遗憾，事后主办方给我提供了电子版的讲义，我觉得仔细阅读还是很有帮助的，我会一并把该讲义发给购买我们套件的网友，供您学习参考。

2. 对于给定的一个 PMSM 电机，在与此程序结合的过程中，参数设置以及性能优化的步骤和方法？FOC GUI 的使用注意，那几个 PID 环路的调试步骤和方法。

对于给定的一个 PMSM 电机，首先最需要知道的是该电机的本身结构特点，这里我不多说了，大体提几点供您参考，第一，PMSM 的机械结构分为 3 种，SM-PMSM, I-PMSM, 第二个又分为 2 种，具体机械结构不多说了，主要是它们的磁路设计有区别，从而造成表贴磁钢（SM-PMSM）的其 $L_q=L_d$, 而内嵌磁钢（I-PMSM）的 $L_d < L_q$, 这个特性会造成软件算法的区别，一般的资料上介绍对于单纯的 FOC 算法，需要使 $I_d = 0$, 也就是说直轴电流为 0，这时马达的特性接近直流电机，可以按直流电机算法控制，如果需要减弱磁控制的话，需要附加 $-I_d$ 电流，是电机克服马达本身磁阻扭矩，使得速度达到额定速度以上，当然这些也同时丧失了交轴电流扭矩，从而造成转矩下降。但是这些是理

论的算法，一般的马达本身远比这些复杂的多，所以不可能达到这些条件，所以 ST 的马达驱动代码库里附加了一个 MTPA 的代码模块，该模块没有提供源代码，只有 LIB 文件，但是这些并不影响我们使用该代码段，其主要功能就是针对 I-PMSM 电机结构（其 $I_d < I_q$ ），根据马达本身的电气参数，较为精确的计算需要施加直轴电流分量 I_d 的数值，从而为控制系统本身尽量提供实现 FOC 算法的条件，对于 SM-PMSM 的电机，近似可以使用 $I_d = 0$ 的算法实现。另一个 FLUX WEAPEN 就是经常说的弱磁算法，该算法本身也是为了精确计算 I_d 分量的，如果您的马达本身不需要精确控制以及超过额定速度运行，这些代码块对您没有用处，我们实际测试中把一台额定转速为 3000RPM 的 BLDC 马达超速到 6060RPM，运行非常稳定，但是运行 2、3 分钟后，电机非常热，实际观测 I_d 电流达到 -19000 以上。还有一个就是前馈电流控制 FEED-FORWARD，该模块也是为了尽量提高系统本身的动态性能，简单的说就是在扰动还没有对控制系统作出影响前，直接输出控制分量，把它抵消。上面的这些选项都与 PMSM 马达本身有很大关系，所以要想计算得到理想控制理论参数，对于电机的电气参数知道的越多越好，基本需要知道的有下面这些：相电阻、相电感(L_d, L_q)、电机极对数，额定电流，额定转矩，反电动势常数，转动惯量，额定速度，额定电压等等，代码库里面有个头文件：MC_PMSM_motor_param.h 专门存储这些参数，里面有些数据是可以用 FOC_GUI 软件计算的，有些需要实际调

整后，写到里面。我觉得以实际调整为主，FOC_GUI 计算为辅助，实际上 FOC_GUI 使用的算法我觉得局限性也很大，有时也和实际差别不小，不要过多依赖它，ST 的 UM0492 文档附录参考的 APPENDIX.4 里面提到了计算这些参数的基本方法和公式。

下面我们说说具体哪些参数与控制有关，基本上就是 8 个字：由里到外，由快到慢。解释如下：由里到外就是说先调内环，再调外环，一般内环是快环，外环一般是慢环，比如电流环为内环，其频率一般到 10-50KHZ，速度环，转矩环或者位置环为外环一般为 10HZ-10KHZ，因为这是闭环控制特性决定的，自控原理中是这样讲的，内环必须完全响应外环的改变，如果反过来，系统势必不稳定或者较大震荡。对于精确调整各个控制环的 PID 参数是个很复杂的功能，同时也要有些技巧在里面，购买我们套件的网友会得到我们整理的几个专门介绍 PID 参数调整的文件，您可以参考里面的办法，这将极大缩短您调整 PID 参数的时间。针对 ST 的马达库，ST 公司也给出了一个调整电流环 PID 参数的办法，您可以参考 UM0492 文档的 APPENDIX.5 里面提到了一个基本方法，我没有实际测试过，理论上应该是有一定效果的。有一点需要提出的是，针对不同的速度闭环，其 PID 参数都要做相应的调整，应该根据实验结果，做好记录，最后在程序里面体现出来，可以做几个速度段的 PID 参数对应表，当然越详细越好了，针对这一要求，ST 的代码库可以方便的把 Iq、Id、SPEED、FLUX、TORQUE 的 PID 参数分别调整，这给我们的测试工作提

供了很大的帮助，具体的调试步骤建议参考提供的几个 PID 调整的文档。

3. 如何调一个马达使其能达到尽可能低的平稳转速。最低的平稳转速与哪些参数相关。

马达为了获得尽可能低的平稳转速，我个人认为主要有下面几点：1、电流环和速度环的 PID 参数是否合适，电流环的响应速度，速度环的响应速度，位置环的响应速度（如果有）。

2、马达的极对数，决定了马达可以到达的最低物理转速。这个与马达设计生产工艺有关。一般在只是用 60 或 120 度空间排列的 HALL 元件做速度取样的情况下，2 对极的 BLDC，最低速到 300RPM，3 对极到 150RPM，4 对极到 60-100RPM，但是如果使用编码器的情况下，1000PPR 的 ENC 就很容易把速度控制到几转/分钟，2500PPR 的 ENC 可以控制到 0.2 转/分钟。

3、速度传感器的精度和分辨率，一般的话使用旋转编码器可以获得精度很高的速度分量，有了准确的速度信息，就可以实现精确的速度控制，很难想象在不能得到高精度速度信息的情况下，实现稳定的转速控制。当然高精度的 ENC 价格不菲，所以应该综合评价，在成本可以承受的情况下，尽量采用高精度的速度传感器。

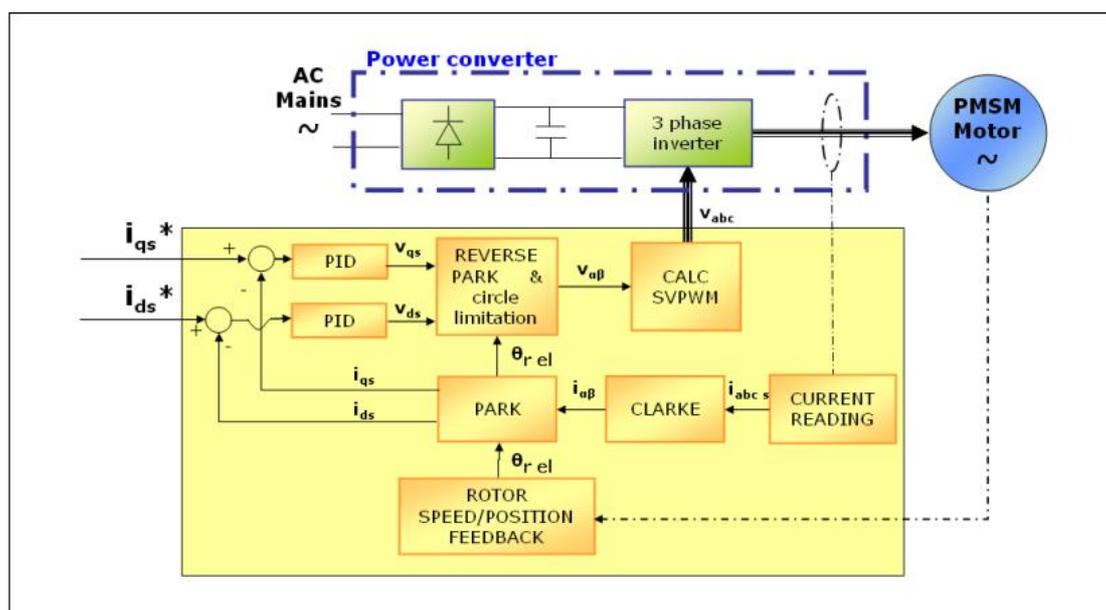
综上所述，在电机物理参数确定的情况下，尽最大可能提供合适的控制参数，可以达到较好的控制效果。所以应该尽量调整好各个控制环的 PID 参数。

4. 程序中有没有驱动电路工作状态检测？比如程序可以探测到 MOS 管损坏或电机短路？

基于使用第一版的客户反馈的信息，我们在第二版上增加了以下硬件保护电路：6 个硬件保护功能。分别是单相过流（正反向电流）保护、母线过流（正反向电流）保护、刹车过流（正反向电流）保护、母线过压保护，母线欠压保护，温度过高保护。每路保护都有 LED 报警显示，同时为了增强可靠性，保护发生后，都会分别给 MCU 和栅极驱动器发送关断信号，保护功率单元 MOSFET，防止元件的损坏，而且该信号必须要求单独的复位信号才能取消，否则保护电路会一直起作用，防止误操作损坏设备和影响人身安全。

5. 程序中，电流环，转矩环和速度环的关系。

请 参 考 下 面 的 图 片 ：



上面的这个图片中黄色方框内部位于闭环控制系统最内层，

它就代表了电流环，根据 FOC 的原理，三相电流 I_a, I_b, I_c 经过各种数学变换最终被转换为 I_d, I_q 电流分量，然后根据参考值 I_{qs} 和 I_{ds} 来计算 e ，最后由 SVPWM 模块实现最终对三相定子电压的矢量控制。在 ST 的代码库里面，转矩环和速度环现在是分开的，这是为了使得程序便于理解做的，也就是说现在您只可以选择 2 种模式控制：速度环+电流环或者转矩环+电流环。但是一般的伺服控制系统可以做到 3 环甚至 4 环控制，就是：速度环+转矩环+电流环或者位置环+速度环+转矩环+电流环。一般把有 3 环或 4 环的系统分成 2 个大环，FAST LOOP 和 LOW LOOP，就是快环和慢环。位置环和速度环属于慢环，它们的响应频率一般在几十 HZ-1KHZ 之间，快环的响应频率一般在几 KHZ-几十 KHZ，根据控制要求不同有差别，不过原则是最内环的频率最高，然后往外变小，最外环的频率最低，否则系统控制结果恶化或者震荡。

针对 ST 的这套代码库，我下面讲讲具体实现这几个控制环的函数。实现源文件为：MC_FOC_Drive.c

- (1) 电流环，实现函数是 FOC_Model 函数，该函数首先通过 A/D 转换取得三相电流值，然后经过 2 个数学变换，得到 I_d, I_q ，然后经过电流环的 PID 调节，输出 V_q, V_d ，再经过 2 个反向数学变换，最后通过调用使用 SVPWM 算法函数，计算出三个半桥的占空

比，实际输出的是电压信号，也就是说通过计算不同 PWM 通道的 DUTY 来实现不同的电压幅值，最后在每个半桥的输出上模拟出类似正弦波的电压。

(2) 转矩环的实现函数是：

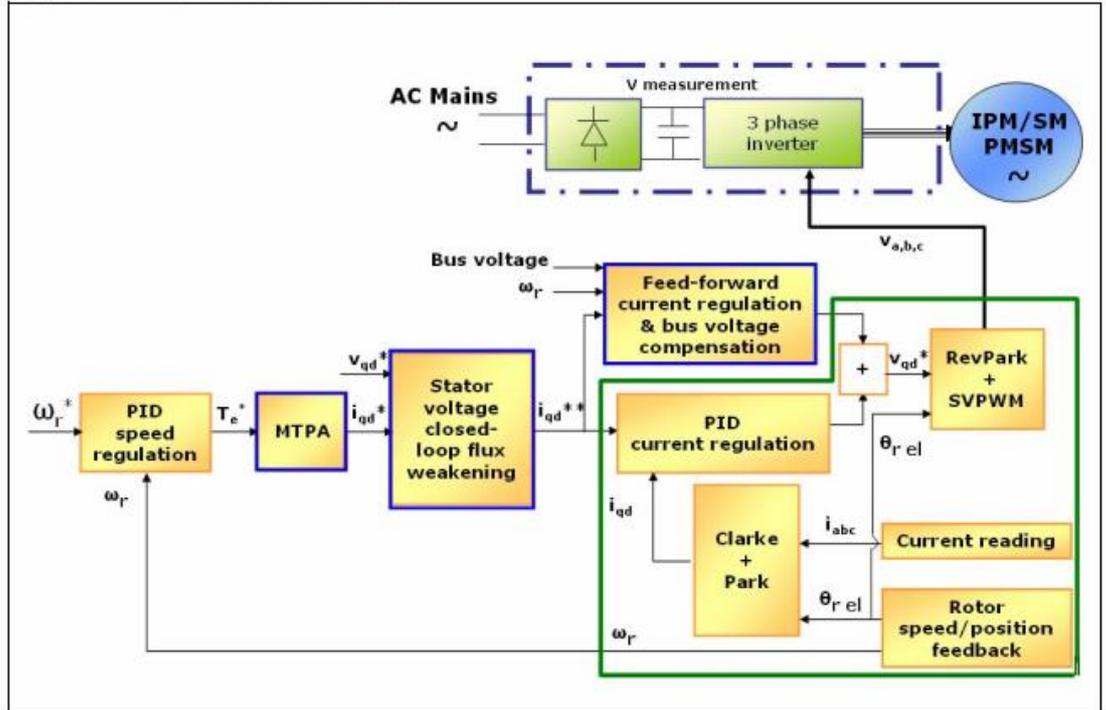
```
void FOC_TorqueCtrl(void)
{
    Stat_Curr_q_d_ref_ref.qI_Component1 = hTorque_Reference;
    Stat_Curr_q_d_ref_ref.qI_Component2 = hFlux_Reference;
#ifdef FEED_FORWARD_CURRENT_REGULATION
    Stat_Volt_q_d_3 =
    FOC_FF_CurrReg(Stat_Curr_q_d_ref_ref,Stat_Volt_q_d_2,

    GET_SPEED_DPP,MCL_Get_BusVolt());
#endif
}
```

可以看到该函数的内容很少，基本就是开始的 2 行代码，把用户设定的扭矩和磁通值直接输入到电流环的 I_q, I_d 的上，这样也是和系统图是对应的，可以看到，如果使用了前馈电流控制，这里有一个函数是可以计算 V_d 的参考值的。

(3) 请看下图：

Figure 5. Speed control loop



The i_{qs} and i_{ds} current components can be selected to perform electromagnetic torque and flux control.

速度环的实现函数是：`void FOC_CalcFluxTorqueRef(void)`。该函数较复杂，不过仔细分析可以看到，如果只是做基本的控制，不加MTPA和FLUX WEAKEN的情况下，该函数还是挺简单的，第一行就计算了

`Stat_Curr_q_d_ref.qI_Component1` 变量，该变量是速度环的输出值，如果没有MTPA和FLUX WEAKEN，那么它就直接赋值给：

`Stat_Curr_q_d_ref_ref = Stat_Curr_q_d_ref` 了。

`Stat_Curr_q_d_ref_ref` 前面讲了又是电流环的 I_q 参考分量。这样就把速度环和电流环连接起来了。该ST马达库没有位置环，实际上根据目前我们的硬件配置，很容

易在速度环的外面增加位置环，这样就可以实现一个全数字控制的闭环伺服控制系统。

稍后时间，我会再写一个整个代码系统的运行分析，相信会给您理解和修改代码很大的帮助。以上文字由于个人水平所限，难免有片面和错误的地方，希望您多提宝贵意见，共同探讨，欢迎交流。

天安电子科技

2010-04-06 深夜

[E-MAIL:DZLJ95@TOM.COM](mailto:DZLJ95@TOM.COM)

QQ:916359820