



YS32电机SDK应用讲解教程—无感FOC

YS32M系列----SDK版本: V1.0.10



YS32芯片介绍---基本型号



YS32F030x4x6x8x

基于 Arm® Cortex®-M0 的 32 位 MCU, Flash 最高达 64KB, 48 引脚,
1.8V – 5.5 V, 64MHz, 1Msps ADC, UART, I²C, SPI 接口

数据手册

产品特性

- 内核
 - 32 位 Arm® Cortex®-M0 CPU
 - 工作频率最高达 64MHz
- 存储器
 - 64Kbytes Flash 存储器
 - 8Kbytes SRAM
- 时钟系统
 - 内部 24MHz 或 48MHz RC 振荡器(HSI)
 - 内部 32.768KHz RC 振荡器(LSI)
 - 4 ~ 32MHz 晶体振荡器(HSE)
 - 32.768KHz 低速晶体振荡器(LSE)
 - PLL(支持对 HSI 或者 HSE 的 2 倍频或者 4 倍频)
- 电源管理和复位
 - 工作电压: 1.8V ~ 5.5V
 - 低功耗模式: Sleep 和 Stop
 - 上电/掉电复位(POR/PDR)
 - 掉电检测复位(BOR)
 - 可编程的电压检测(PVD)
- 通用输入输出(I/O)
 - 多达 44 个 I/O, 均可作为外部中断
 - 普通驱动电流 8mA
 - 9 个 GPIO 支持超强灌电流, 可配置为 60mA/40mA/20mA
- 5 通道 DMA 控制器
- 1 x 12-bit ADC
 - 支持 18 个外部输入通道
 - 2 个内部通道, 温度传感器通道 (Temperature sensor)和参考电压通道 (VREFINT)
 - 输入电压转换范围: 0 ~ V_{CC}
 - 参考电压可配置为: 1.2V 内部参考电压、V_{CC} 和外部输入电压
- 定时器
 - 1 个 16 位高级控制定时器(TIM1)
 - 5 个通用的 16 位定时器 (TIM3/TIM14/TIM15/TIM16/TIM17)
 - 1 个低功耗定时器(LPTIM), 支持从 Stop 模式唤醒
 - 1 个基本定时器(TIM6)
 - 1 个独立看门狗定时器(IWDT)
 - 1 个窗口看门狗定时器(WWDT)
 - 1 个 SysTick 定时器(SysTick timer)
 - 1 个 IRTIM
- RTC
- 通讯接口
 - 2 个串行外设接口(SPI)
 - 2 个通用同步/异步收发器(UART), 支持自动波特率检测
 - 2 个 I²C 接口, 支持标准模式(100kHz)、快速模式(400kHz), 支持 7 位寻址模式
- LED
 - 支持 4x4、4x5、5x6、6x7、7x7、7x8、8x8 点阵驱动
 - LED0 ~ LED8 扫描序列可配置
- 安全机制
 - 硬件 CRC32 模块
 - 唯一 UID 号
 - 存储器保护(PCROP, RDP, WRP)
- 4 个比较器
- 串行单线调试(SWD)
- 工作温度: -40 ~ 85°C
- 封装形式:
 - LQFP48、LQFP44、LQFP32
 - QFN32
 - TSSOP28、TSSOP24、TSSOP20



YS32F031 系列

基于 Arm® Cortex®-M0 的 32 位 MCU, Flash 最高达 32KB,
1.8V-5.5 V, 48MHz, ADC, LED, OPA, COMP, UART, I²C, SPI

数据手册

产品特性

- 内核
 - 32 位 ARM® Cortex® -M0 CPU
 - 最高 48MHz 工作频率
- 存储器
 - 32Kbytes flash 存储器
 - 4Kbytes SRAM
- 时钟系统
 - 内部 48MHz RC 振荡器(HSI)
 - 内部 32.768KHz RC 振荡器(LSI)
 - 4~32MHz 晶体振荡器(HSE)
- 电源管理和复位
 - 工作电压: 1.8V ~ 5.5V
 - 低功耗模式: Sleep、Stop
 - 上电/掉电复位(POR/PDR)
 - 掉电检测复位(BOR)
 - 可编程的电压检测 (PVD)
- 通用输入输出(I/O)
 - 多达 30 个 I/O, 均可作为外部中断
 - 普通驱动电流 8mA
 - 9 个 GPIO 支持超强灌电流, 可配置为 80mA/60mA/40mA/20mA
- 3 通道 DMA 控制器
- 1 个 12-bit ADC
 - 每个 ADC 支持 13 路外部输入通道
 - 每个 ADC 支持 4 个内部通道(温度传感器、参考电压、OPA1、OPA2)
 - 输入电压转换范围: 0 ~ V_{CC}
 - 参考电压可配置为: 内部参考电压 (1.2V/2.0V/2.5V)、V_{CC} 和外部输入电压
- LED
 - 支持 4*4、4*5、5*6、6*7、7*7、7*8、8*8 点阵驱动
 - LED0~LED8 扫描序列可配置
- 定时器
 - 1 个 16 位高级控制定时器 (TIM1)
 - 4 个通用 16 位定时器(TIM3/14/16/17)
- 1 个低功耗定时器(LPTIM), 支持从 stop 模式唤醒
- 1 个独立看门狗定时器 (IWDT)
- 1 个窗口看门狗定时器 (WWDT)
- 1 个 SysTick timer
- 1 个 IRTIM
- 通讯接口
 - 1 个串行外设接口(SPI)
 - 2 个通用异步收发器(UART),支持自动波特率检测
 - 1 个 I²C 接口
- 2 个比较器
- 2 个 OPA
- RTC
- 安全特性
 - 硬件 CRC32
 - 读保护 RDP
 - 写保护 WRP
 - 对特定程序区的访问保护 PCROP
 - Option byte 写保护
 - 96 位的芯片唯一 ID (UID)
- 串行单线调试(SWD)
- 工作温度:
 - -40 ~ 85°C
 - -40 ~ 105°C
- 封装形式
 - LQFP32
 - QFN32、QFN24、QFN20
 - SSOP28、SSOP24
 - TSSOP20
 - SOP28、SOP20



YS32M系列电机驱动芯片介绍---衍生型号规格

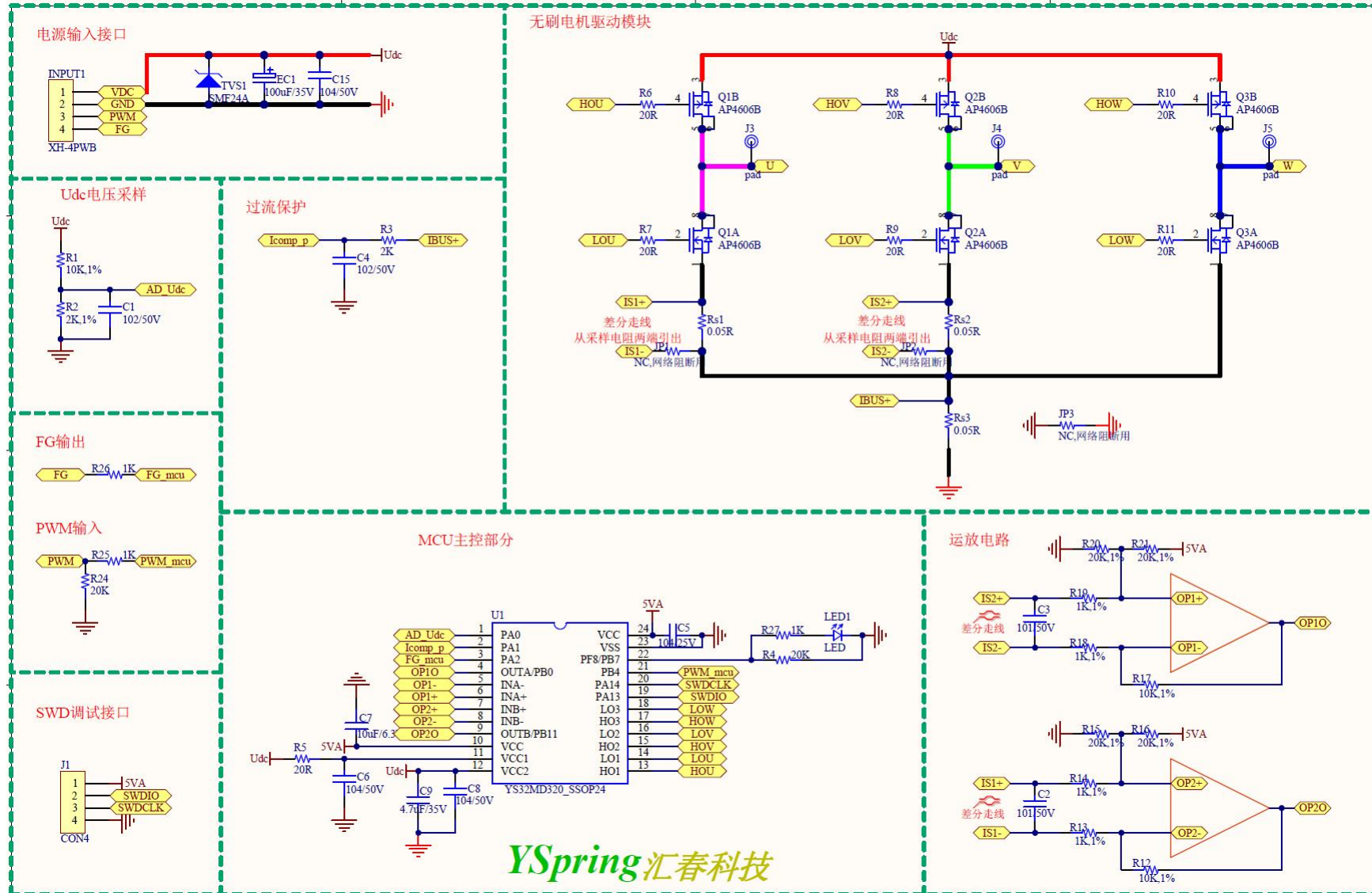
YS32F030衍生型号

型号	规格概述
YS32MD320	集成1路12bit高精度ADC、4路比较器、2路OPA、三相 P/N MOS 管栅极驱动电路，内置5V电压输出的LDO，工作电压范围 6V ~ 36V，工作温度范围-40~105℃，工作频率高达 64MHz，提供64Kbytes Flash、8Kbytes SRAM，适用于多种低压电机应用场景，例如风扇、水泵、排气扇、风扇灯等
YS32MD311	集成1路12bit高精度ADC、4路比较器、2路OPA、高精度低功耗LDO和三相N/N MOS半桥驱动电路，工作电压范围 4.5V ~ 28V，工作温度范围-40~105℃，工作频率高达 64MHz，提供64Kbytes Flash、8Kbytes SRAM，适用于多种低压电机应用场景，例如暴力风扇、水泵、筋膜枪、按摩仪等
YS32M032	集成1路12bit高精度ADC、4路比较器、2路OPA，工作电压范围 1.8V ~ 5.5V，工作温度范围-40~105℃，工作频率高达 64MHz，提供64Kbytes Flash、8Kbytes SRAM，适用于高压电机应用场景，例如油烟机、空调内外风机、水泵、吊扇、工业风扇等

YS32F031衍生型号

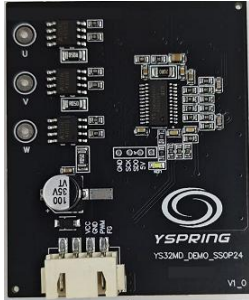
型号	规格概述
YS32MD220	集成1路12bit高精度ADC、2路比较器、2路OPA、三相 P/N MOS 管栅极驱动电路，内置5V电压输出的LDO，工作电压范围 6V ~ 36V，工作温度范围-40~105℃，工作频率高达 48MHz，提供32Kbytes Flash、4Kbytes SRAM，适用于多种低压电机应用场景，例如风扇、水泵、排气扇、风扇灯等
YS32MD211	集成1路12bit高精度ADC、2路比较器、2路OPA、高精度低功耗LDO和三相N/N MOS半桥驱动电路，工作电压范围 4.5V ~ 28V，工作温度范围-40~105℃，工作频率高达 48MHz，提供32Kbytes Flash、4Kbytes SRAM，适用于多种低压电机应用场景，例如暴力风扇、水泵、筋膜枪、按摩仪等
YS32MD210	集成1路12bit高精度ADC、2路比较器、2路OPA、高精度低功耗LDO和三相N/N MOS半桥驱动电路，悬浮电压高达200V，工作电压范围 5V ~ 20V，工作温度范围-40~105℃，工作频率高达 48MHz，提供32Kbytes Flash、4Kbytes SRAM，适用于多种中高压电机应用场景，例如平衡车、电动工具、暴力风扇、工业扇等
YS32F031	集成1路12bit高精度ADC、2路比较器、2路OPA，工作电压范围 1.8V ~ 5.5V，工作温度范围-40~105℃，工作频率高达 48MHz，提供32Kbytes Flash、4Kbytes SRAM，适用于高压电机应用场景，例如油烟机、空调内外风机、水泵、吊扇、工业风扇等

YS32M系列电机驱动DEMO介绍---3P+3N电路原理图

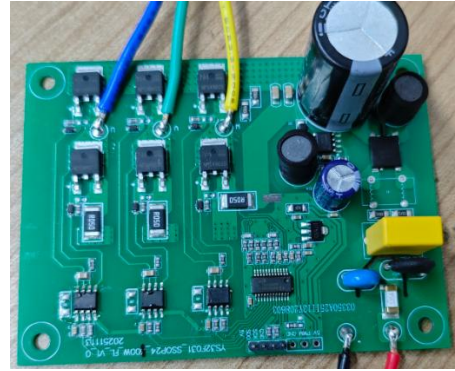




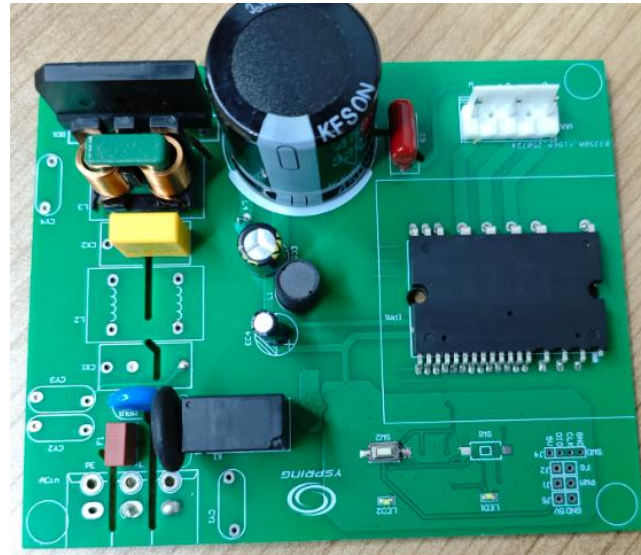
YS32M系列电机驱动DEMO介绍---板子实物



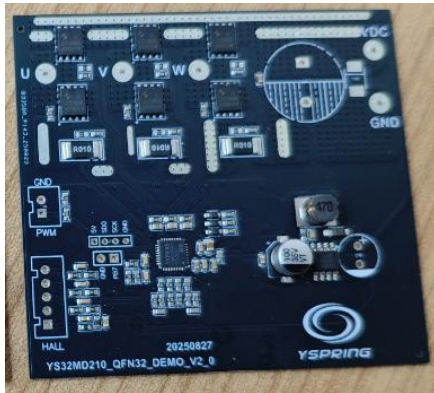
MD320/MD220 低压N+P



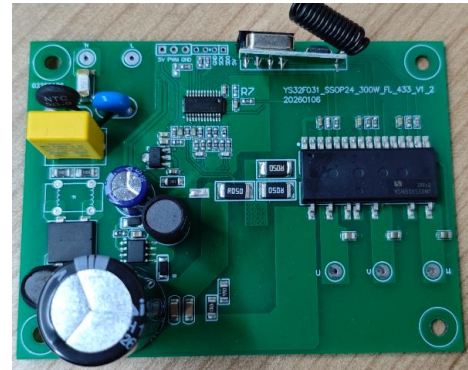
M032/F031高压分立器件



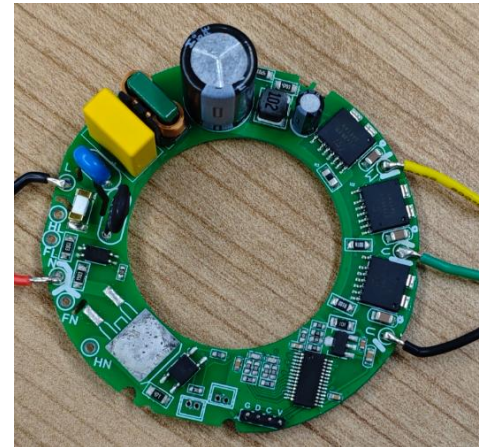
M032/F031高压大功率IPM



MD210 中压双N大电流



M032/F031高压中小功率IPM



M032/F031高压半桥IPM高速风筒



YS32M系列电机驱动DEMO介绍---驱动电流波形

逆风启动波形



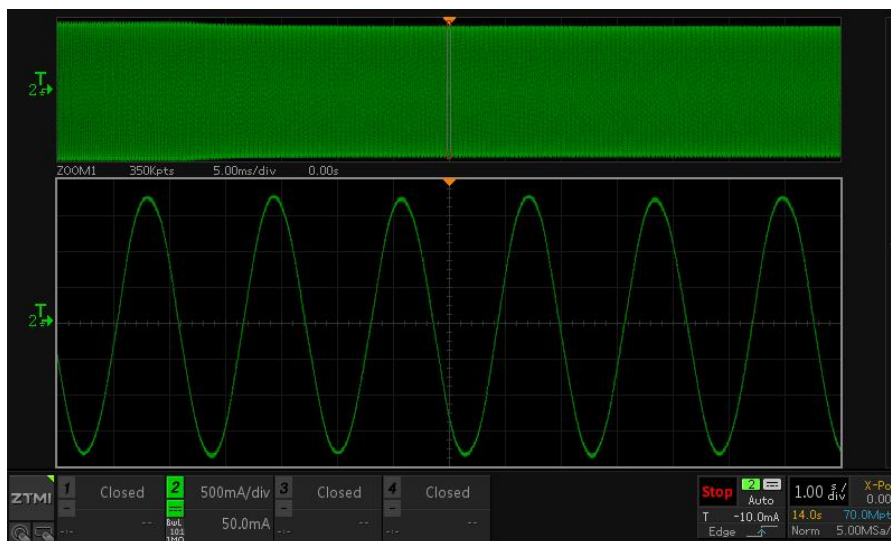
顺风启动波形



静止启动波形



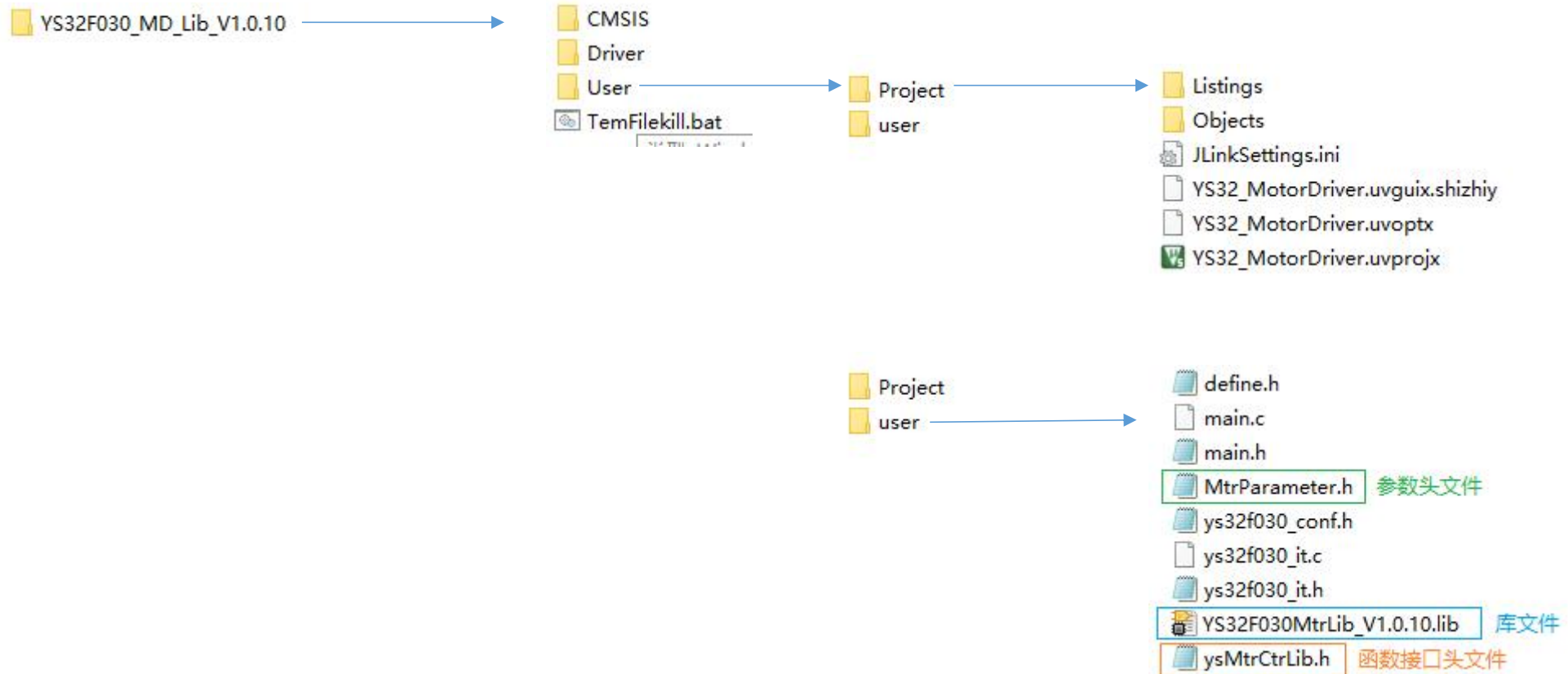
运行波形



特点：平滑、稳定，非常正弦，平滑过度



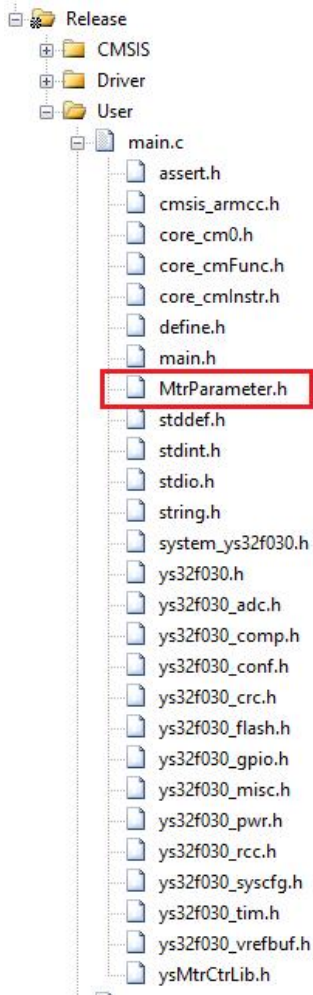
认识YS32M系列电机SDK---文件夹及相关文件



```
19 // =====
20 extern void fmm_LPF(LPF_T *p); // 限幅滤波函数
21
22 // 硬件初始化及中断函数接口 =====
23 extern void fYS_PLL_Init(void);
24 extern void fYSMD_HardwareInit(void);
25 extern void fYSMD_ParsInit(void);
26 extern void fYSMD_ADCDMA_INT(void);
27 extern void fYSMD_lmsProcess(void);
28 extern void fYSMD_HardOCP(void);
29
30 // 电机SDK相关参数设置接口 =====
31 extern void fYSMD_SaveNewPWM(s16 period, s16 pluse);
32 extern void fYSMD_SetPars_PWM(s16 Frq, s16 dtim, s16 tsmin, s16 tdly);
33 extern void fYSMD_SetPars_Motor1(s16 Ls, s16 Rs, s16 kvol);
34 extern void fYSMD_SetPars_Motor2(u8 kLs, u8 kRs, u8 kkvol);
35 extern void fYSMD_SetPars_Motor3(s16 kp1lmin, s16 kp1lmax, s8 kElpfmin, s8 kElpfmax);
36 extern void fYSMD_SetPars_DriverHard1(u8 pwmpin, u8 ExtPwmEn, u8 FgEn);
37 extern void fYSMD_SetPars_DriverHard2(u8 direct, u8 workmode);
38 extern void fYSMD_SetPars_Current(s16 imin, s16 imax, s16 kp, s16 ki);
39 extern void fYSMD_SetPars_OLVP(s16 lvpon, s16 lvprec, s16 ovprec, s16 ovpon);
40 extern void fYSMD_SetPars_Ome(s16 runmin, s16 runmax, s16 startome, s16 omeok);
41 extern void fYSMD_SetPars_StallTim(s16 stalltim, s16 stallrectim, s16 nostalltim, u8 stalltimes);
42 extern void fYSMD_SetPars_ExtPwmCtrl(s16 dtymn, s16 dtymax, s16 outmin, s16 outmax);
43 extern void fYSMD_SetPars_ExtPwmCtr2(s16 pwmon, s16 pwmoft, u8 pwrang);
44 extern void fYSMD_SetPars_CurOther(s16 ibrake, u16 braketime, s8 kcur, u8 KcurEn);
45 extern void fYSMD_SetPars_StartCharge(u8 ChrgeEn, u8 RevStartEn, u8 omeinc, u8 flglinecmd);
46 extern void fYSMD_SetPars_DutySet(s16 pwwmin, s16 pwwmax, s16 qpwwmax);
47 extern void fYSMD_SetPars_LossPhase(s16 llossphase, u16 lossphasetime);
48 extern void fYSMD_SetXCtrPars(s16 Kp, s16 Ki, u8 period, s16 xctrinc);
49
50 // 电机SDK外部控制输入接口 =====
51 extern void fYSMD_SetDir(u8 dirt); // 设置电机运行方向, CW / CCW
52 extern void fYSMD_SetId(s16 idcur); // 设置D轴电流, 0~32767
53 extern void fYSMD_SetPLL(s16 kp1l); // 设置锁相环Kp1l参数, 0~32767
54 extern void fYSMD_SetExtMtrCtr(s16 setdty); // 外部控制量输入, 0~16383
55 extern void fYSMD_SetMtrSpeedErr(void); // 设置电机失速状态
56
57 // 电机SDK内部参数对外输出接口 =====
58 extern s16 fYSMD_GetUdc(void); // 读取母线电压
59 extern u8 fYSMD_GetFault(void); // 读取错误类型
60 extern s16 fYSMD_GetOme(void); // 读取角速度ome, 0~360°对应0~65536
61 extern s16 fYSMD_GetPwr(void); // 读取功率表征值, 乘上一个系数得到真实功率, 该系数需要测试计算得到
62 extern s16 fYSMD_GetExtPwmDuty(void); // 读取外部控制输入PWM的占空比
63 extern s16 fYSMD_GetIq(void); // 读取Q轴电流
```



认识YS32M系列电机SDK---参数配置文件“MtrParameter.h”



```
10 // PWM Parameter =====
11 #define uPWM_Freq (16) // kHz 载波频率 1~16
12
13 #define uPWM_DTim (0.35) // us 死区 0.25~1.0
14 #define uTSMIN (1.4) // us 采样窗口 1.4~2.0
15 #define uTdly (0.4) // us 采样延时 0~0.6
16
17
18 // Motor Parameter =====
19 #define uLs (1.0/2) // (mH) 相电感 (线电感/2)
20 #define uRs (1.0/2) // (ohm) 相电阻 (线电阻/2)
21
22 #define uPoles (5) // 电机转子磁极对数
23
24 #define uKPl1 s16Q8(0.7) // 观测器PLL参数 // 0.005~128.0
25
26 // HardSource select or enable =====
27
28 #define uHWDIRECT (CW) // CW / CCW 正/反转
29
30 #define uPWMPIN (UVW) // UVW / WVU 相顺序
31
32 #define uEXTPWMEN (ENABLE) // ENABLE / DISABLE 外部PWM调速使能
33
34 #define uFGOUTEN (DISABLE) // ENABLE / DISABLE 速度反馈输出使能
35
36 // ADC Vref =====
37 #define uVref (5.0) // ADC基准电压
38
39 // Current Parameter =====
40 #define uRshunt (0.05) // (ohm) 采样电阻
41 #define uIGain (10) // 放大倍数
42
43 #define CurKp (0.07) // 电流环KP参数
44 #define CurKi (0.004) // 电流环KI参数
45
46 // Udc R1/R2 Parameter =====
47 #define uDcR1 (10) // (Kohm) 电压采样电阻R1 (连接到Udc的电阻)
48 #define uDcR2 (2) // (Kohm) 电压采样电阻R2 (连接到GND的电阻)
49
50 // Udc Protect Parameter =====
51 #define uUDCOVP_ON (15) // (V) 过压保护值
52 #define uUDCLVP_ON (9) // (V) 欠压保护值
53
54 #define uUDCOVP_REC (uUDCOVP_ON-1) // (V) 过压恢复值
55 #define uUDCLVP_REC (uUDCLVP_ON+1) // (V) 欠压恢复值
56
57 // RPM Parameter =====
58 #define uRPM_Min (200) // 正常运行的最低转速
59 #define uRPM_Max (3000) // 正常运行的最高转速
60
61 #define uRPM_Start (uRPM_Min*0.06) // 启动速度
62 #define uRPM_OK (uRPM_Min*0.6) // 启动完成速度
63
64 #define uOmeInc (3) // 启动Ome加速度, 1~128
```

```
73 // Stall Parameter =====
74 #define uStallTimes (3) // 堵转保护次数, 0为无限次数
75 #define uStallTim (4000) //ms, 堵转时间, 执行堵转保护
76 #define uStallRecTim (1000) //ms, 堵转保护后再次启动的时间
77 #define uNoStallTim (6000) //ms, 无堵转时间, 清零堵转次数
78
79 // Work mode =====
80 #define uWM_Select (uWM_VLoop) // uWM_ILoop--电流环模式,
81
82 // 各模式下的参数设置 =====
83 #if (uWM_Select==uWM_ILoop) // 电流环 =====
84
85 #define uIsetMin (0.6) // (A) 最小相电流
86 #define uIsetMax (2.0) // (A) 最大相电流
87
88 #define uPWMMin (0.15) // 电机输出最小占空比
89 #define uPWMMax (0.93) // 电机输出最大占空比
90
91 #define uXCtrInc (0.1) // 外部控制爬坡大小
92
93 #elif (uWM_Select==uWM_VLoop) // 电压环 =====
94
95 #define uIsetMin (0.6) // (A) 最小相电流
96 #define uIsetMax (2.0) // (A) 最大相电流
97
98 #define uPWMMin (0.25) // 电机输出最小占空比
99 #define uPWMMax (0.93) // 电机输出最大占空比
100
101 #define uXCtrInc (0.001) // 外部控制爬坡大小
102
103 #elif (uWM_Select==uWM_FLoop) // 功率环 =====
104
105 #define uIsetMin (0.6) // (A) 最小相电流
106 #define uIsetMax (2.0) // (A) 最大相电流
107
108 #define uPWMMin (0.25) // 电机输出最小占空比
109 #define uPWMMax (0.93) // 电机输出最大占空比
110
111 #define uPowerMin (500) // 最小功率
112 #define uPowerMax (1200) // 最大功率
113
114 #define xKp (0.12) // 功率环KP参数
115 #define xKi (0.01) // 功率环KI参数
116
117 #define uXCtrInc (5) // 外部控制爬坡大小
118
119 #elif (uWM_Select==uWM_SLoop) // 速度环 =====
120
121 #define uIsetMin (0.6) // (A) 最小相电流
122 #define uIsetMax (3.4) // (A) 最大相电流
123
124 #define uPWMMin (0.20) // 电机输出最小占空比
125 #define uPWMMax (0.93) // 电机输出最大占空比
126
127 #define uSpeedMin (400) // 最低转速
128 #define uSpeedMax (2500) // 最高转速
129
130 #define xKp (0.5) // 速度环KP参数
131 #define xKi (0.005) // 速度环KI参数
132
133 #define uXCtrInc (2) // 外部控制爬坡大小
134
135 #define xPeriod (1) //ms, 速度环执行周期
```

```
143 // ExtCtr Parameter =====
144 #define uExtPwmRANGE (uEPM_200_10K) // 外部PWM频率范围选: uEPM_20_1K / uEPM_200
145
146 #define uExtPwmDcy_Min (0.20) // 外部PWM控制最小占空比 (大于该值启动)
147 #define uExtPwmDcy_Max (0.90) // 外部PWM控制最大占空比
148
149 #define uExtPwmDcy_On (0.18) // 外部启动占空比
150 #define uExtPwmDcy_Off (0.16) // 外部关闭占空比
151
152 #define uExtCmdLineEn (ENABLE) // ENABLE / DISABLE // 内部线性处理使能
153
154 // Brake Parameter =====
155 #define uBrake (0.3) // (A) 刹车电流 !!! 不要太大, 避免电压冲高
156
157 #define uBrakeTime (300) // (ms) 刹车时间, 根据负载惯性设置
158
159 // ome电流系数 =====
160 #define uKcur (6) // ome电流系数, 1~100
161 #define uOmeKcurEN (ENABLE) // ENABLE / DISABLE // Ome(转速)电流系数使能
162
163 // =====
164 #define uChargeEn (ENABLE) // ENABLE / DISABLE // 预充电使能
165
166 // =====
167 #define uReverseStartEn (ENABLE) // ENABLE / DISABLE // 逆风启动使能
168
169 // =====
170 #define uQPWMMax (0.35) // D轴输出最大占空比
171
172 // =====
173 #define uLossPhaseCur (uIsetMin*0.3) // 缺相保护电流阈值 (A)
174 #define uLossPhaseTim (1200) //ms, 缺相保护的时间
175
176 // 短路保护 =====
177 #define uShortVrefSelt (uSHORTP_1_4_VREF) // uSHORTP_1_4_VREF, uSHORTP_1_2_VREF,
178 #define uShortFiltNum (96) // 0~65535
```



YS32M系列电机SDK调试---电机参数测量

```
// Motor Parameter
#define uLs          (1.0/2) // (mH) 相电感 (线电感/2)
#define uRs          (1.0/2) // (ohm) 相电阻 (线电阻/2)

#define uPoles       (5) //电机转子磁极对数

#define uKP11        s16Q8(0.4) //观测器PLL参数 // 0.005~128.0
```

电机相电感及相电阻参数测试方法:

使用LCR数字电桥进行测量，测量条件通常设置为：电压250mV，频率1KHz，测量得到电机的线电感和线电阻，直接填入，如上，测量的线电感是10uH，线电阻是0.18Ω；

电机转子磁极对数测试方法:

使用直流电源，通常设置电流为1A左右，电压5V左右，电机上做好标识，用于确认转动位置，电源连接电机的其中2相，通电后用手转动电机转子，每转过一定角度，转子会停在一个固定位置，依次转动转子一圈，停顿的次数即是电机转子的磁极对数。



YS32M系列电机SDK调试---电流环调节

```
#define uIsetMin      (1.0) // (A) 最小相电流  
#define uIsetMax      (2.0) // (A) 最大相电流
```

```
#define CurKp         (0.07) // 电流环KP参数  
#define CurKi         (0.005) // 电流环KI参数
```

电流环参数调试方法:

根据电机的相电感和相电阻，以及驱动板的电压，进行调节，在调节测试过程中不断优化，使得电流波形和效率达到较好的状态。

调节规律:

先小后大，小步调节，边调边看电流波形和效率



YS32M系列电机SDK调试---电压保护及转速保护

```
55 // Udc Protect Parameter =====
56 #define uUDCOVP_ON          (15) // (V) 过压保护值
57 #define uUDCLVP_ON          (9)  // (V) 欠压保护值
58
59 #define uUDCOVP_REC          (uUDCOVP_ON-1) // (V) 过压恢复值
60 #define uUDCLVP_REC          (uUDCLVP_ON+1) // (V) 欠压恢复值
61
62
63 // RPM Parameter =====
64 #define uRPM_Min             (200) //正常运行的最低转速
65 #define uRPM_Max             (3000) //正常运行的最高转速
66
67 #define uRPM_Start           (uRPM_Min*0.06) // 启动速度
68 #define uRPM_OK              (uRPM_Min*0.6) // 启动完成速度
69
70 #define uOmeInc               (3) // 启动Ome加速度, 1~128
71
72
73 // Stall Parameter =====
74 #define uStallTimes           (3) // 堵转保护次数, 0为无限次数
75 #define uStallTim             (4000) //ms, 堵转时间, 执行堵转保护
76 #define uStallRecTim         (1000) //ms, 堵转保护后再次启动的时间
77 #define uNoStallTim           (6000) //ms, 无堵转时间, 清零堵转次数
```

电压保护参数设置方法:

根据驱动板的供电电压, 通常情况下, 取 $\pm 3V$, 做过压和欠压保护的阈值, 当然在实际的项目中要根据客户的实际需求来进行设定。

电机转速参数设置方法:

根据电机在实际应用中的转速设置, 最大转速需要考虑电机空载的情况, 避免电机在空载时触发转速保护。



YS32M系列电机SDK调试---工作模式选择及各模式下的参数设置

```
75 // Work mode =====
76 #define uWM_Select      (uWM_VLoop) // uWM_ILoop--电流环模式,   uWM_VLoop--电压环模式, uWM_PLoop--功率环模式, uWM_SLoop--速度环模式
77
78
79 // 各模式下的参数设置 =====
80 #if (uWM_Select==uWM_ILoop) // 电流环 =====
81
82 #define uIsetMin        (0.2) // (A) 最小相电流
83 #define uIsetMax        (1.0) // (A) 最大相电流
84
85 #define uPWWMin         (0.15) // 电机输出最小占空比
86 #define uPWWMax         (0.93) // 电机输出最大占空比
87
88 #define uXCtrInc         (0.05) // 外部控制爬坡大小
89
90 #elif (uWM_Select==uWM_VLoop) // 电压环 =====
91
92 #define uIsetMin        (0.2) // (A) 最小相电流
93 #define uIsetMax        (1.0) // (A) 最大相电流
94
95 #define uPWWMin         (0.15) // 电机输出最小占空比
96 #define uPWWMax         (0.93) // 电机输出最大占空比
97
98 #define uXCtrInc         (0.001) // 外部控制爬坡大小
99
100 #elif (uWM_Select==uWM_PLoop) // 功率环 =====
101
102 #define uIsetMin        (0.6) // (A) 最小相电流
103 #define uIsetMax        (2.0) // (A) 最大相电流
104
105 #define uPWWMin         (0.25) // 电机输出最小占空比
106 #define uPWWMax         (0.93) // 电机输出最大占空比
107
108 #define uPowerMin       (500) // 最小功率
109 #define uPowerMax       (1200) // 最大功率
110
111 #define xKp              (0.12) // 功率环kp参数
112 #define xKi              (0.01) // 功率环ki参数
113
114 #define uXCtrInc         (5) // 外部控制爬坡大小
115
116 #elif (uWM_Select==uWM_SLoop) // 速度环 =====
117
118 #define uIsetMin        (0.6) // (A) 最小相电流
119 #define uIsetMax        (2.0) // (A) 最大相电流
120
121 #define uPWWMin         (0.25) // 电机输出最小占空比
122 #define uPWWMax         (0.93) // 电机输出最大占空比
123
124 #define uSpeedMin       (3000) // 最低转速
125 #define uSpeedMax       (6000) // 最高转速
126
127 #define xKp              (0.99) // 速度环kp参数
128 #define xKi              (0.005) // 速度环ki参数
129
130 #define uXCtrInc         (2) // 外部控制爬坡大小
131
132 #define xPeriod          (1) //ms, 速度环执行周期
133
134 #endif
```

四种工作模式可选

电流环模式参数

电压环模式参数

功率环模式参数

速度环模式参数

四种模式可选：
在实际项目中，根据项目需求，进行选择



YS32M系列电机SDK调试---运行效率优化

```
#define uKPl1          s16Q8(0.4) //观测器PLL参数 // 0.005~128.0  
  
#define CurKp          (0.07) // 电流环KP参数  
#define CurKi          (0.005) // 电流环KI参数
```

影响效率的参数主要是电流环参数和观测器PLL参数，两者密切配合，是整个系统最关键的参数，决定了电机运行的性能。

uKPII的调节规律：

先小后大，电机启动有卡顿则加大，电机出现震动则减小

电流环调节规律：

先小后大，小步调节，边调边看电流波形和效率

协同调节，效率优化：

在电机启动及运行平稳情况下，加大uKPII和电流环参数，可以提高运行效率，如果电机电流震荡，则需要降低，并保留一定的裕度。



YS32M系列电机SDK调试---母线电压及相电流接口及ADC通道配置

电机驱动相关的硬件配置，包括：母线电压、U相电流、V相电流 以及 用于短路保护的比较器输入，全部包含在main.c的初始化函数里面，在调试的最开始，优先确认修改正确这些硬件相关配置，确认好后，在进行下一步的调试。

```
106 // 注入通道配置 =====
107
108 // GPIO config =====
109 GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AN;
110 GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
111 GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
112 GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
113
114 GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0; // 正确设置 电压 电流采样通道的IO
115 GPIO_Init(GPIOA, &GPIO_InitStructure);
116
117 GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0 | GPIO_Pin_11;
118 GPIO_Init(GPIOB, &GPIO_InitStructure);
119
120
121 // ADC config =====
122 ADC_InitStruct.ClockMode = ADC_ClockMode_HSI;
123 ADC_InitStruct.ADC_DataAlign = ADC_DataAlign_Right;
124 ADC_InitStruct.ADC_ScanMode = ADC_ScanMode_ON;
125
126
127 ADC_InitStruct.ADC_ExternalTrigInjectedConvEdge = ADC_ExternalTrigInjectedConvEdge_EN;
128 ADC_InitStruct.ADC_ExternalTrigInjectedConv = ADC_ExternalTrigInjectedConv_T1_TRGO;
129
130
131 ADC_Init(ADC, &ADC_InitStruct);
132
133
134 ADC->JSQR = ((3-1)<<25) // 正确设置 电压 电流采样通道
135 | (ADC_Channel_2<<10) // 母线电压
136 | (ADC_Channel_14<<15) // U相电流
137 | (ADC_Channel_10<<20); // V相电流
138 ADC->SMPR = ADC_SampleTime_3_5Cycles;
```

正确设置母线电压，U、V相电流的IO口

正确设置母线电压，U、V相电流的ADC输入通道

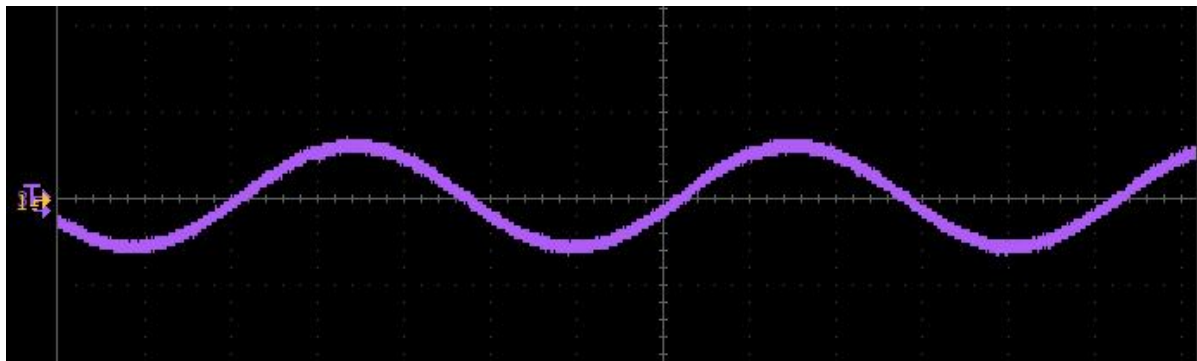
在不确定电机驱动相关的硬件配置是否正确的时候，为避免较大的电流冲击，建议根据电机的电感电阻大小，设置合适的较小的占空比输出，且KpII也设置一个较小的值，通过示波器查看输出的电流波形，修改UV相的ADC采样通道配置及 UVW 输出相顺序，直到出现如下图的电流波形，硬件配置就配置OK了。然后再把KpII和占空比设置回正常大小。

```
#define uKPII                s16Q8(0.04) //观测器PLL参数

#elif (uWM_Select==uWM_VLoop) // 电压环 =====
#define uIsetMin             (0.6) // (A) 最小相电流
#define uIsetMax             (2.0) // (A) 最大相电流

#define uPWMMin              (0.15) //电机输出最小占空比
#define uPWMMax              (0.23) //电机输出最大占空比

#define uXCtrInc             (0.001) //外部控制爬坡大小
```



电机缓慢转动，出现比较正弦的电流波形，证明硬件配置是正确的



YS32M系列电机SDK调试--- ome电流系数

```
161 // ome电流系数-----  
162 #define uKcur          (6) // ome电流系数, 1~100  
163 #define uOmeKcurEN    (ENABLE) // ENABLE /DISABLE Ome(转速)电流系数使能  
164
```

ome电流系数主要作用是当电机转速很低或堵转时保护电机，避免低转速时，输出大电流，导致电机发热温度升高甚至烧毁起火。所以正常情况下，都建议开启该功能；

在开启该功能情况下，如果电机需要低转速大电流运行，可以适当调大uKcur。如果调试过程中发现电机转速上不去，电流也上不去，有可能是因为uKcur太小所致。但如果增大了uKcur，也没有变化，则为其他原因，比如uKPII 太小，或者最大输出占空比太小。



YS32M系列电机SDK调试---逆风启动使能

```
169 // =====  
170 #define uReverseStartEn    (ENABLE) //ENABLE /DISABLE // 逆风启动使能  
171  
4 // Brake Parameter =====  
5 #define uIbrake            (0.3) // (A) 刹车电流 !!! 不要太大, 避免电压冲高  
7  
8 #define uBrakeTime        (300) // (ms) 刹车时间, 根据负载惯性设置  
9
```

使能后，在逆风出现电机反转的时候启动电机，会先进行刹车，刹车电流大小可通uIbrake设置，同时刹车时间也可以设置。

但需要注意，在非电池供电情况下，较大的刹车电流，会出现反灌导致母线电压升高，导致过压烧坏器件的问题。



YS32M系列电机SDK调试---缺相保护

```
175 // =====  
176 #define uLossPhaseCur      (uIsetMin*0.3) //缺相保护电流阈值 (A)  
177 #define uLossPhaseTim      (1200) //ms, 缺相保护的时间  
178
```

电机缺相时，至少会有其中一相出现电流接近为0的情况，缺相保护就是利用这个原理进行检测的；但由于实际电流信号采样有波动，所以实际情况一般都不会使0，而是在0附近波动，因此需要设置一定的范围，以便能正确检测缺相的发生。

正常情况下，相电流也有过零的情况出现，所以需要设置适当的缺相保护的时间，以避免误判。



YS32M系列电机SDK调试---需要电机反转要怎么设置

```
extern void fYSMD_SetDirt(u8 dirt); // 设置电机运行方向, CW / CCW
```

通过调用 fYSMD_SetDirt(u8 dirt); 函数设置电机的运行方向。

例如:

```
302
303 while (1)
304 {
305 // Delay_lms(8000);
306 // fYSMD_SetExtMtrCtr(s16Q14(0.0));
307 // Delay_lms(8000);
308 // fYSMD_SetExtMtrCtr(s16Q14(0.2));
309
310 Delay_lms(5000);
311 fYSMD_SetDirt(CW);
312 Delay_lms(5000);
313 fYSMD_SetDirt(CCW);
314 }
315
```

正反转各5秒



YS32M系列电机SDK调试---需要开弱磁，该如何设置

```
extern void fYSMD_SetId(s16 idcur); // 设置D轴电流, 0~32767
```

通过调用 `fYSMD_SetId(s16 idcur);` 函数设置D轴的电流，便可进入弱磁运行。
正常项目一般不建议开弱磁，因为弱磁会增加损耗，降低效率，电机会发热。
如果可以先建议客户修改电机绕组参数，比如减少线圈匝数，增加线径等，
还可以节省材料和加工时长。

例如：

```
fYSMD_SetId(nlset(1.0)); // 设置弱磁电流1A
```



YS32M系列电机SDK调试---如何增加和修改失速保护

用户可以在main_ext void fMain_UserSpeedErrCheck(s16 ome, s16 Eq) 函数完成实数保护的功能，且已有程序框架，只需要通过调试测试得到正常运行时的ome—Eq和失速时的ome—Eq关系，然后设置一个合理的阈值，就可以了。

```
219 // 失速保护功能实现，用户根据实际项目修改。判断方法：相同ome下，失速时的Eq比正常运行时Eq的小，具体大小关系每个电机不一样，需  
220 main_ext void fMain_UserSpeedErrCheck(s16 ome, s16 Eq)  
221 {  
222 #if 1 // 不需要时可以屏蔽  
223 // 请在此处添加用户代码  
224 static u8 SpeedErrCnt=0;  
225  
226 if((ome>nRpm2Ome(uRPM_Min)) && (Eq<DTs16(ome*3))) // 此处需要根据实际项目调整!!!  
227 {  
228     if(++SpeedErrCnt>30)  
229     {  
230         SpeedErrCnt=0;  
231         fYSMD_SetMtrSpeedErr(); //确认发生失速时，调用该函数设置电机状态和错误标志  
232     }  
233 }  
234 else  
235 {  
236     if(SpeedErrCnt>2)  
237     {  
238         SpeedErrCnt-=2;  
239     }  
240 }  
241 #endif  
242 }
```



YS32M系列电机SDK调试---内部状态及信息的获取

SDK库提供了一系列的对外输出接口，用户可以通过调用以获取对应的信息。

当前SDK库提供的对外输出接口如下：

```
57 // 电机SDK内部参数对外输出接口 -----  
58 extern s16 fYSMD_GetUdc(void); // 读取母线电压  
59 extern u8 fYSMD_GetFault(void); // 读取错误类型  
60 extern s16 fYSMD_GetOme(void); // 读取角速度ome, 0~360°对应0~65536  
61 extern s16 fYSMD_GetPwr(void); // 读取功率表征值, 乘上一个系数得到真实功率, 该系数需要测试计算得到  
62 extern s16 fYSMD_GetExtPwmDuty(void); // 读取外部控制输入PWM的占空比  
63 extern s16 fYSMD_GetIq(void); // 读取Q轴电流  
..
```

```
59 | extern u8 fYSMD_GetFault(void); // 读取错误类型
```

通过调用 fYSMD_GetFault() 函数获取到当前的错误类型。

错误类型如下：

```
typedef enum
{
    mcfNone = 0, // 0--无错误
    mcfSelfChkErr, // 1--上电自检错误
    mcfLVP, // 2--低压保护
    mcfOVP, // 3--过压保护
    mcfOCP, // 4--过流保护
    mcfStall, // 5--堵转保护
    mcfShort, // 6--短路保护
    mcfLossPhase, // 7--缺相保护
    mcfSpeedErr, // 8--失速保护
} mcFaultType;
```



YS32M系列电机SDK调试---速度获取及转化

```
extern s16 fYSMD_GetOme(void); // 读取角速度ome, 0~360°对应0~65536
```

通过调用 fYSMD_GetOme(void); 函数获取到当前的角速度ome，电机的转速需要经过转化才可以得到。

ome转化到rpm的公式如下：

$$n = \frac{60 * \omega * f_{pwm}}{P * 65536}$$

其中：P为电机极对数

f_{pwm}为载波频率



YS32M系列电机SDK调试---功率获取及转化

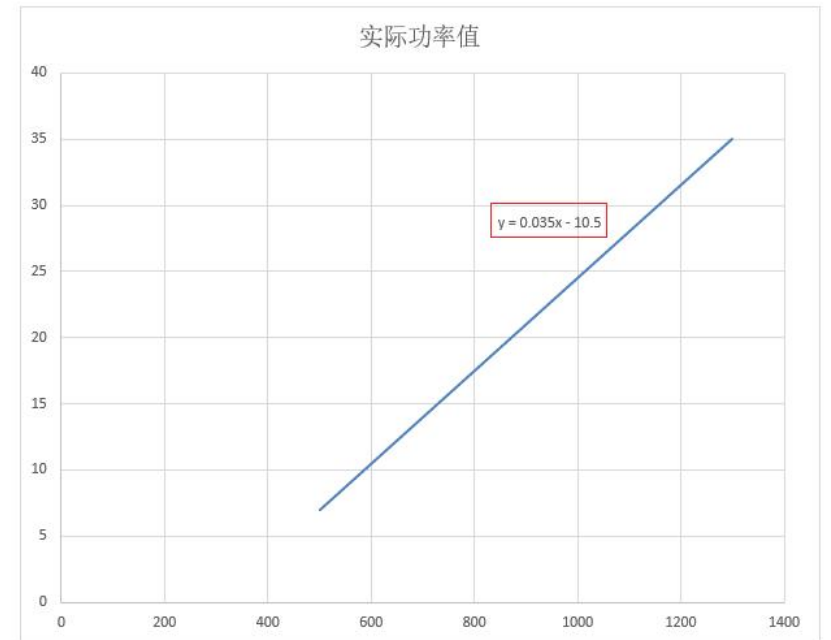
```
extern s16 fYSMD_GetPwr(void); // 读取功率表征值，乘上一个系数得到真实功率，该系数需要测试计算得到
```

通过调用 `fYSMD_GetPwr(void)`; 函数得到功率的表征值，并非实际的功率值，需要乘上一个系数才得到实际功率值，该系数需要测试计算得到。

大概方法：

- 1、测试几个点的功率表征值和实际功率值;
- 2、在EXCEL表格中进行处理得到公式;
- 3、根据生产的公式，修改到程序中即可;

表征值	实际功率值
500	7
700	14
900	21
1100	28
1300	35





YS32M系列电机SDK调试---母线电压获取及转化

```
extern s16 fYSMD_GetUdc(void); // 读取母线电压
```

通过调用 fYSMD_GetUdc(void); 函数获取母线电压，得到的是母线电压的ADC值。

如需得到实际的是母线电压需要进行转化。

```
// Udc R1/R2 Parameter =====  
#define uDcR1          (10) // (Kohm) 电压采样电阻R1 (连接到Udc的电阻)  
#define uDcR2          (2)  // (Kohm) 电压采样电阻R2 (连接到GND的电阻)  
  
// ADC Vref =====  
#define uVref          (5.0) // ADC基准电压
```

根据分压比和基准电压，12位的ADC最大值是4095，进行换算即可。

实际电压值=获取到的母线电压AD值* 基准电压* 分压系数 / 4095



YS32M系列电机SDK调试--- Q轴电流获取及转化

```
extern s16 fYSMD_GetIq(void); // 读取Q轴电流
```

通过调用 fYSMD_GetIq(void);函数获取Q轴电流，得到的同样也是ADC值。

如需得到实际的Q轴电流需要进行转化。

```
207 // =====  
208 //  
209 #define uImax          (uVref/2/uIGain/uRshunt)  
210 #define nIset(cur)    s16Q11((cur)/uImax)  
211  
212
```

```
37 // =====  
38 // ADC Vref =====  
39 #define uVref          (5.0) // ADC基准电压  
40  
41  
42 // Current Parameter =====  
43 #define uRshunt        (0.05) // (ohm) 采样电阻  
44 #define uIGain         (10) // 放大倍数
```

根据采样电阻值、放大倍数和基准电压，12位的ADC最大值是4095，进行换算即可。

实际电流值=获取到的电流AD值* 基准电压 / 放大倍数 / 4095 / 采样电阻值



YS32M系列电机SDK调试---低压高速电机调试

低压高速电机的特点是电感量小，电阻小，而且往往需要较大的相电流驱动。

因此，对应会有部分的参数设置不一样。

```
246 int main(void)
247 {
248     // 此部分为电机控制涉及的所有参数及硬件资源初始化，不可修改>>>>>:
249     fYS_PLL_Init();
250
251     // 电机控制参数初始化 =====
252
253     fYSMD_SetPars_PWM(uPWM_Freq, nPWM_DTim, nTSMIN, nTdly);
254
255     fYSMD_SetPars_Motor1(s16Q10(uLs), s16Q10(uRs), s16Q10(uKVol));
256     fYSMD_SetPars_Motor2(10, 10, 10);
257     fYSMD_SetPars_Motor3(uKPII*0.6, uKPII, s8Q7(0.25), s8Q7(0.45));
258
259
260
261
262
263
264
265
266
267
268 #define uKVol
269     (uIGain*uKV*uRshunt)
```

根据电机参数调整Q格式，尽量保证精度，同时不要溢出，结果在2048~4096为佳。对于s16Q7(uKVol)，需要保证结果在16384~32767为佳

另外，低压高速电机，一般uKPII需要设置大些，经验值在 2.0 ~ 20.0 左右。



YS32M系列电机SDK调试---高压电机调试

高压电机的特点是电感量大，电阻大，而且往往需要较小的相电流驱动。

因此，对应会有部分的参数设置不一样。

```
246 int main(void)
247 {
248     // 此部分为电机控制涉及的所有参数及硬件资源初始化，不可修改>>>>>
249     fYS_PLL_Init();
250
251     // 电机控制参数初始化 =====
252
253     fYSMD_SetPars_PWM(uPWM_Freq, nPWM_DTim, nTSMIN, nTdly);
254
255     fYSMD_SetPars_Motor1(s16Q5(uLs), s16Q5(uRs), s16Q7(uKVol));
256     fYSMD_SetPars_Motor2(7, 7, 9);
257     fYSMD_SetPars_Motor3(uKPl1*0.6, uKPl1, s8Q7(0.25), s8Q7(0.45));
...
218 #define uKVol          (uIGain*uKV*uRshunt)
219
```

根据电机参数调整Q格式，尽量保证精度，同时不要溢出，结果在2048~4096为佳。对于s16Q7(uKVol)，需要保证结果在16384~32767为佳

另外，高压电机，一般uKpl1需要设置小些，经验值在 0.01 ~ 0.1 左右。

```
256 fYSMD_SetPars_Motor2(7, 7, 9);
```

这个设置比前面的Q格式数多2，要不uKpl1要设置很小



YS32M系列电机SDK调试---最大电流和最大电压设置的注意事项

由于运放偏置有偏差，及ADC采样在最低和最高电压处转换精度会有偏差，再加上需要预留一定的冗余量，所以建议设置的电流最大值不大于理论最大值的90%。

内部预留150的AD值，当AD值低于145或高于（4095-145）时，会触发过流保护。

```
// ADC Vref =====  
#define uVref          (5.0) // ADC基准电压  
  
#define uIsetMin      (0.6) // (A) 最小相电流  
#define uIsetMax      (2.0) // (A) 最大相电流
```

同理，电压需要留的冗余量通常要求更多一些，最高电压也尽量小于理论最大电压的80%。

```
// Udc R1/R2 Parameter =====  
#define uDcR1          (10) // (Kohm) 电压采样电阻R1（连接到Udc的电阻）  
#define uDcR2          (2) // (Kohm) 电压采样电阻R2（连接到GND的电阻）  
  
// Udc Protect Parameter =====  
#define uUDCOVP_ON     (15) // (V) 过压保护值  
#define uUDCLVP_ON     (9) // (V) 欠压保护值
```



感谢聆听!

扎根春天向阳生，用芯共创美好未来