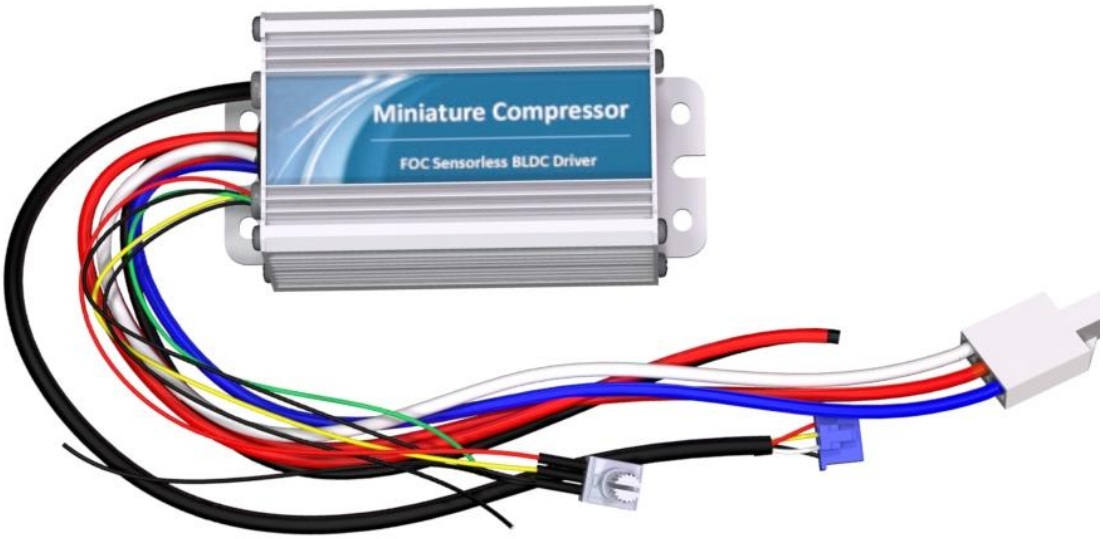
微型压缩机驱动器 GE2117-GP

使用说明书



Rev 2.0

珠海市世马科技有限公司

http://www.seama.com.cn

- 感谢您选择世马科技的产品。
- 使用之前，请仔细阅读本说明书，并请妥善保管。如有任何疑问，请与我司联系。
- 本文档采用 LibreOffice 编写,字体采用开源字体-思源黑体。

产品特点

GE2117-GP 驱动器是一款专为微型压缩机驱动而设计的驱动器，它有如下特点；

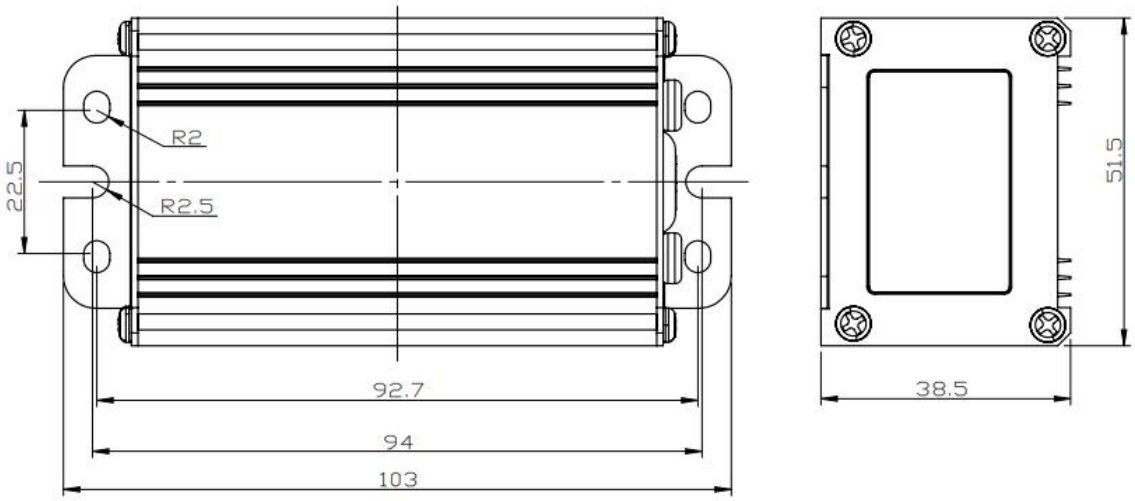
✓ **参数开放，客户可以设定相关参数**
✓ **特别的启动优化，压缩机启动无忧**
✓ **可以在宽电压下工作**
✓ **FOC 正弦波控制，低噪音**
✓ **具有最大功率限制功能，保护压缩机**
✓ **低速力矩补偿，转速平稳**
✓ **结构紧凑，尺寸小巧**

电气参数

项目	GE2117-GP	GE2117-GP	GE2117-GP
适用对象	48V 压缩机	24V 压缩机	12V 压缩机
输入最大电流	6.5A	13A	17A
最大输入功率	280W	250W	180W
最小转速	1800rpm	1800rpm	1800rpm
最大转速	6000rpm	6000rpm	4800rpm
压缩机转速精度	60rpm	60rpm	60rpm
输入电压（额定）	48VDC	24VDC	12VDC
建议输入电压范围	DC 36V-54V	DC 18V-30V	DC 8V-16V
极限电压	54V		
变频器效率	>90%		
冷却方式	铝壳自然冷却 保持通风		
工作环境	温度：-20℃ ~ 55℃ 湿度：30% ~ 90%		
保护功能	过、欠压保护（可设置） 软件过流保护（可设置） 硬件过流保护 散热器过温保护 堵转保护 缺相保护		

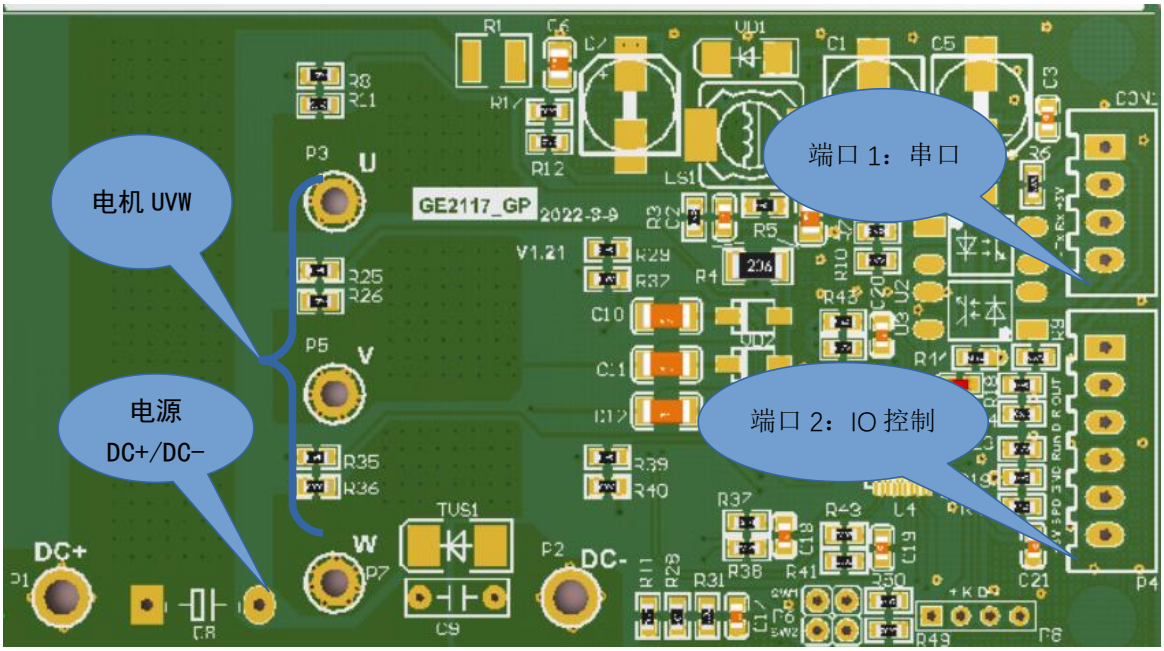
驱动方式	正弦波 FOC 矢量驱动
控制方式	UART 通讯和模拟量 IO 控制
电控盒尺寸	103*52*39mm

几何尺寸



驱动器接口定义

■ **内部接口图：**



■ **电源接口**：16AWG 软硅胶线

红色：直流电源正极（DC+）

黑色：直流电源负极（DC-）

■ **压缩机接口**：16AWG PVC 护套镀锡铜线

红色：电机 U 相

白色：电机 V 相

蓝色：电机 W 相

■ **控制接口**：

(1)端口 1 -- 串口控制线 （光电隔离）

串口控制线为 4 芯线，连接上图的 2.54 端口 1，

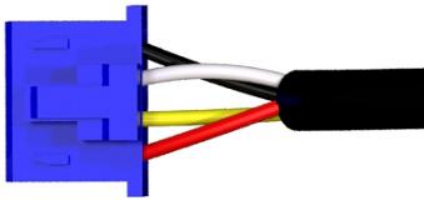
红色： +5V 输入接口

黄色: TTL UART Txd

白色：TTL UART Rxd

黑色：ISO GND

该接口是光电隔离，ISO GND 与内部驱动是不共地的。+5V 是需要外部输入电压



(2)端口 2-- 开关及模拟量 IO 口（非隔离控制）（可选接线）

红色：+5V （信号 5V，无法输出电流）

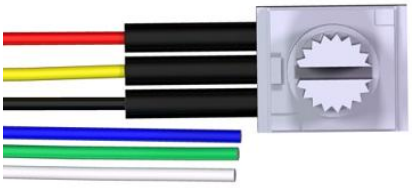
黄色：Sp 速度信号输入 0-5V 。

黑色：Gnd 电源信号地

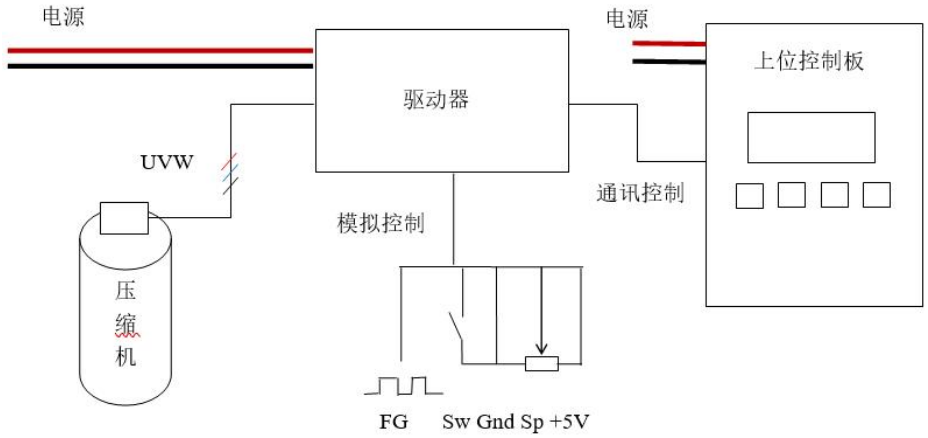
绿色：Run 启停使能

蓝色：DIR 压缩机固定速度使能

白色：FG 压缩机运行速度频率输出



驱动器连接示意图



注意：通讯控制与模拟控制只能二选一，当都连接时，驱动器优先按照通讯控制指令进行工作。

驱动器控制功能

■ **驱动器上电及工作逻辑**

1），驱动器上电等待 10s，然后开始接受指令。

2），驱动器接收到启动信号，无论设定速度多少，驱动器初始化启动的速度为 3000rpm，稳定在 3000rpm 速度 10 秒后，速度再逐步自动闭环到设定速度值。

3），发送停机指令，如果当前速度大于 3000rpm，则驱动器先减速，低于 3000rpm 后压缩机停机工作，如果当前速度小于 3000rpm，则驱动器直接停机。

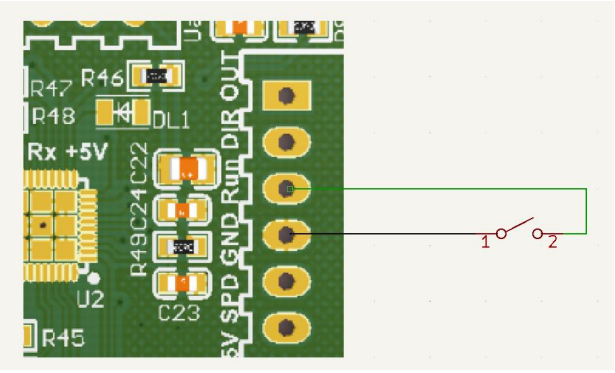
4），再次启动，驱动器的运行间隔是 10s，如果没有间隔小于该时间，则驱动器处于等待状态，到达该间隔时间后再进行启动判断。

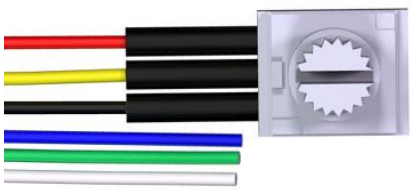
5），驱动器发生故障后，除硬件电流故障外都会自动重启 5 次，如果大于 5 次则停机等待电源复位。重新发送启动指令可以清楚除电流故障外的其他故障。

■ **驱动器 IO 控制(端口 2)**

1），使能控制

端口 2 的 黑色线（GND） 与绿色线（RUN）短接





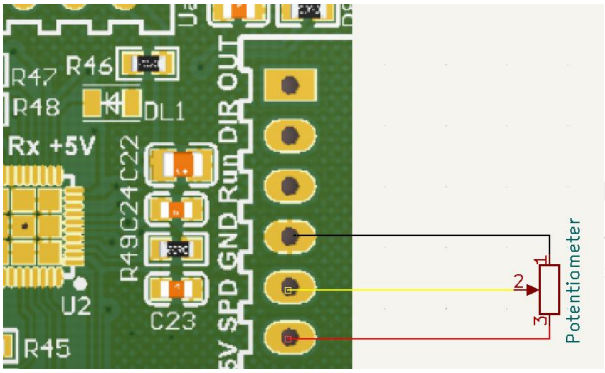
注意：驱动默认设置是高电平有效，即不接任何线，只要给速度信号，则默认是使能状态，如果和地短

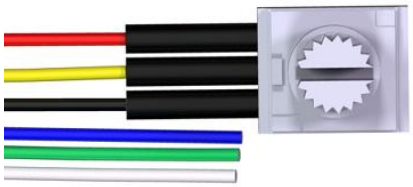
接，则驱动失能，无论速度如何设定，压缩机将不会工作。

2），速度调节控制

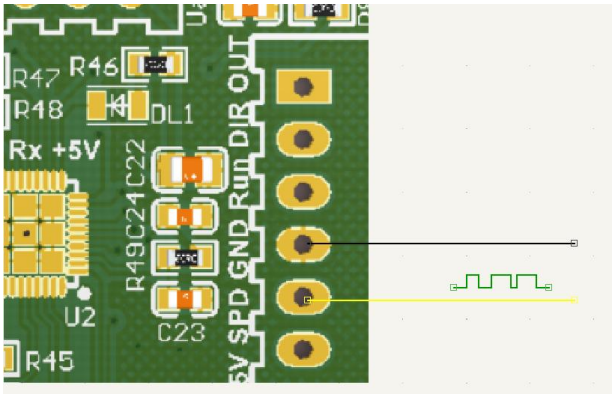
外部信号线定义：红色：+5V；黄色：Spd 速度输入信号；黑色：Gnd。

A：SPD 接口可以接受电位器调节或 0-5V 的模拟量输入





B ： SPD 接受电平是 0-5V 的 200Hz-1KHz 的 PWM 的占空比（0-98%）的控制：

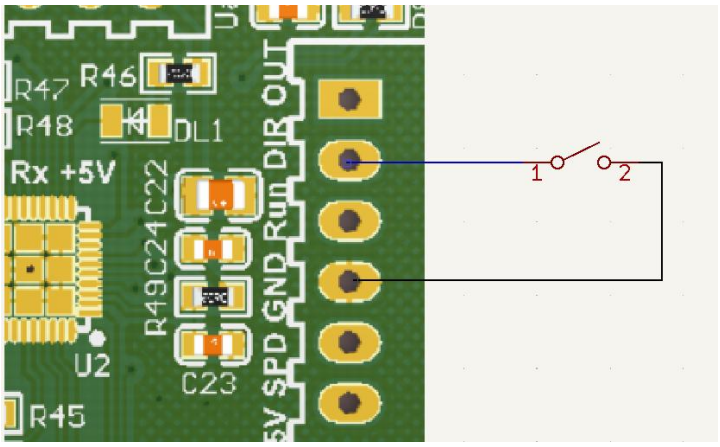


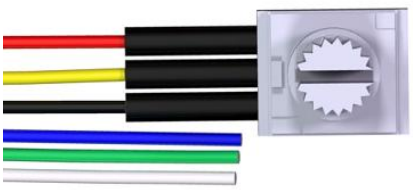
注意：PWM 输入的最高电压是 5V，信号需要和 Gnd 共地。

3），固定速度模式

信号线定义： 蓝色：固定速度使能线；黑色：Gnd

如下图连接，压缩机按照程序设定，进行设定频率运行，系统初始 3600rpm（该速度可以通过更改 0x1023 的值进行更改）

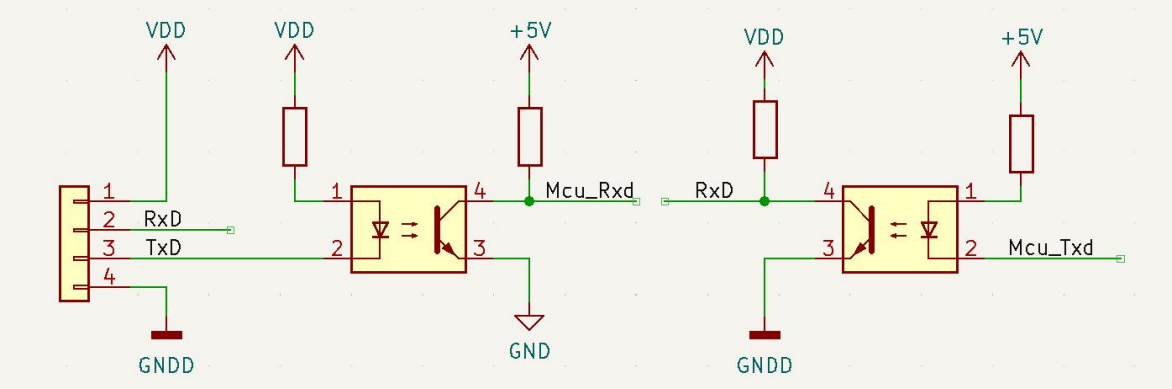




由于系统默认是使能，因此直接短接这两根线就可以驱动压缩机按固定速度工作

■ ***驱动通讯控制***

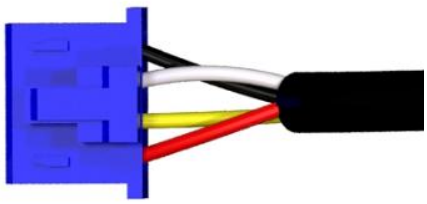
端口 1 原理图：



通信通讯速率是 9600 N 8 1

驱动器可以接收高品压缩机定制协议控制和 modbus 协议控制。

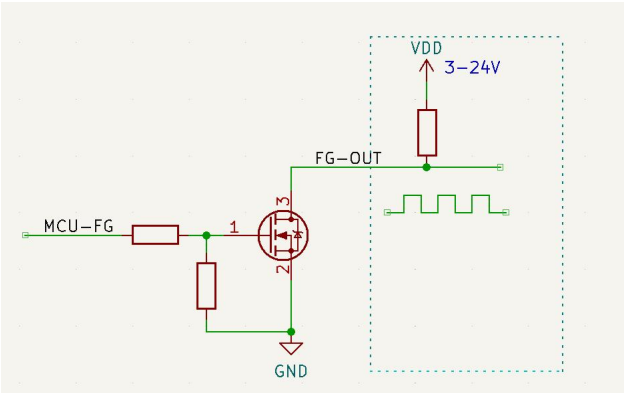
具体协议见后面。串口线如下图。红色：+5V Input； 黄色：Rxd； 白色：Txd；黑色：GNDD.

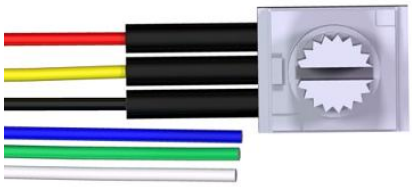


注意：为了光电隔离需要 GND 和 GNDD 内部是不相连的

■ **速度输出 IO**

白色线是速度输出信号。每 1 电角度 1 个脉冲，电机每转输出脉冲是极对数数量，通过测量该输出方波的频率可以得到电机的转速。输出采用集电极开路，因此需要接上拉电阻 1-10K，不可直接接入电源，否则可能永久损坏该功能。





速度计算 Rpm= 测的 FG-OUT 频率（Hz）*60/电机极对数；

驱动器错误指示

故障号	故障说明	备注
0	无故障	
1	硬件检测电流故障	电流过大
2	软件检测电流故障	电流过大
3	驱动器过温保护	散热不良，负载过大
4	输入电压高于设定规范	
5	输入电压低于设定规范	
6	定位启动失败	
7	运行中电机堵转或启动失败	
8	缺相错误	UVW 断线或未接入电机

驱动器参数设定及协议

■ **参数设定寄存器表：**

序号	地址	默认值	取值范围	单位	说明
0	0x1000	1	0-32		Modbus 地址
1	0x1001	0			NC
2	0x1002	0			NC
3	0x1003	0			NC
4	0x1004	0	3-4		通信使能地址 3-压缩机启动，4-压缩机停止
5	0x1005	0			NC
6	0x1006	0	0-7000		通信控制压缩机运行速度
7	0x1007	5	44571	s	电机极对数
8	0x1008	145	1-32767	mΩ	电机相电阻
9	0x1009	75	1-32767	uH	电机相电感 Lq
10	0x100A	75	1-32767	uH	电机相电感 Ld
11	0x100B	300	1-32767	*0.01V/Krpm	电机反电动势常数
12	0x100C	2000	1-2000	*0.01A	电机额定电流 Lq
13	0x100D	0			NC
14	0x100E	1600	1-7000	rpm	电机允许运行的最低转速，用于堵转错误判定

15	0x100F	7000	1-7000	rpm	电机运行运行的最大转速，用于转速异常判定
16	0x1010	0			NC
17	0x1011	1000	ms	ms	电机启动前定位时间
18	0x1012	300	0-1000	*0.01A	电机启动电流 Lq
19	0x1013	300	0-1000	*0.01A	电机启动电流 Ld
20	0x1014	1			NC
21	0x1015	0			NC
22	0x1016	0	0-2		压缩机转速给定方式：0-电位器/固定转速 1-指令控制 2-上电以 0x1023 设置的固定速度运行
23	0x1017	0	0-1		压缩机运行方向 0-CW 1-CCW
24	0x1018	4000	1-32767		电机 q 轴 PID 参数 P
25	0x1019	25	1-32767		电机 q 轴 PID 参数 I
26	0x101A	4000	1-32767		电机 d 轴 PID 参数 P
27	0x101B	25	1-32767		电机 d 轴 PID 参数 I
28	0x101C	4000	1-32767		电机速度 PID 参数 P
29	0x101D	25	1-32767		电机速度 PID 参数 I
30	0x101E	60	1-1024		电机反电动势滤波系数
31	0x101F	50	1-1024		电机转速滤波系数
32	0x1020	1			NC
33	0x1021	50	1-2047	*0.0025V	模拟量设置转速的最低电压值
34	0x1022	1650	1-2047	*0.0025V	模拟量设置转速的最高电压值
35	0x1023	3600	1600-7000	rpm	当（0x1016）= 2 时，压缩机直接以此设置速度运行。当 DIR 压缩机固定速度使能，压缩机也以此速度运行。
36	0x1024	0			NC
37	0x1025	0			NC
38	0x1026	10	8-60	V	欠压保护电压
39	0x1027	55	8-60	V	过压保护电压
40	0x1028	1200	1-2000	*0.01A	过流保护电流
41	0x1029	85	25-100	℃	过温保护温度
42	0x102A	1000	1-7000	rpm	堵转保护转速
43	0x102B	100	1-1000	*0.1S	驱动器停止后，再启动等待时间
44	0x102C	1200	1-32767	*0.01A	驱动器启动时，允许的最大电流
45	0x102D	0	0-1		是否开启过流降低转速功能
46	0x102E	1150	1-32767	*0.01A	认为过流的电流

47	0x102F	3600	1800-6000	RPM	过流之后，转速设定
48	0x1030	30	1-1000	S	降低转速时间
49	0x1031	1000	1-32767	*0.01A	恢复到设定速度的电流
50	0x1032	5	0-32767		错误重启次数，当此值为零时，发生错误不重启
51	0x1033	180	1-32767	S	驱动器正常运行多长时间之后，清除错误累加
52	0x1034	1	0-1		当驱动器用高品协议时，是否开启掉线自停功能
53	0x1035	120	1-32767	S	高品协议检测掉线时间
54	0x1036	500	1-32767		缺相保护值保护阀值
55	0x1037	1000	1	ms	缺相保护时间
56	0x1038	3000		*0.01A	缺相判定电流
57	0x1039	2500		Rpm	判定堵转方式 2 转速
58	0x103A	200			堵转保护方式 2 判定时间
59	0x103B	2000		mS	欠压保护时间
60	0x103C	1000		mS	过压保护时间
61	0x103D	200			过流保护时间
62	0x103E	200			过温保护时间
63	0x103F	1000			堵转保护方式 1 判定时间
注意：除 0x1001-0x1006 因兼容需求，可接受指令，但不会保存； 其他寄存器可读可写可保存。					

状态参数表：

序号	地址	默认值	单位	说明
1	0x2000	0		驱动状态：0-待机，1-工作
2	0x2001	0		*电机当前速度*
3	0x2002	0		保留
4	0x2003	0		当前电压
5	0x2004	0		当前电流
6	0x2005	0		当前错误值
7	0x2006	0		当前驱动器温度值
注意：该组寄存器地址只读				

■ 通讯协议指令说明

1、modbus 协议：

通信速率是 9600 N 8 1； （无特殊说明，以下数字为 16 进制）

压缩机驱动器指令遵循部分 modbus 协议。用于修改驱动器参数，或者控制启停和转速。

为了方便采用电脑串口工具的调试，系统内置了通用的 modbus 的效验码 0xBB 0xAA，建议只是在测试的时候使用，正常程序时，请程序计算 Modbus-CRC16.

Modbus 协议部分说明：

参数修改说明：				
发送：			返回：	
0x01	地址		0x01	地址
0x06	功能码		0x06	功能码
0x01	寄存器地址（高）		0x01	寄存器地址（高）
0xXX	寄存器地址（低）		0xXX	寄存器地址（低）
0xdataH	修改值（高）		0xdataH	修改值（高）
0xdataL	修改值（低）		0xdataL	修改值（低）
0xCRCH	校验码 0xBB		0xXX	校验码
0xCRCL	校验码 0xAA		0xXX	校验码
参数查询说明：				
发送：			返回：	
0x01	地址		0x01	地址
0x03	功能码		0x03	功能码
0x01	寄存器地址（高）		0x02	返回数值个数
0xXX	寄存器地址（低）		0xdataH	查询值（高）
0x00	查询寄存器个数（高）		0xdataL	查询值（低）
0x01	查询寄存器个数（低）		……	……
0xCRCH	校验码 0xBB		0xXX	校验码
0xCRCL	校验码 0xAA		0xXX	校验码

■ 特殊指令

01 06 60 03 00 01 BB AA 保存更改的参数
01 06 60 03 00 03 BB AA 恢复默认的出厂设定值，需上电重启
01 06 60 00 xx xx BB AA 通讯控制启停与速度，当设置值 XX XX 大于或等于寄存器 0x100E（默认 1600rpm）的值时，压缩机启动

V1.0 的兼容指令：

01 06 10 04 00 03 BB AA 启动

01 06 10 04 00 04 BB AA 停机

01 06 10 06 xx xx BB AA 发送 xx 速度显示

V2.0 可以直接用 01 06 60 00 xx xx BB AA 实现启停

2、高品公司的协议(自定义协议)：

驱动板接受 TTL 电平的 RS232（UART）通讯方式。

波特率：9600bps

数据格式：1 位起始位，8 位数据，1 位停止位

驱动器每 1s 钟自动发送一帧数据给上位机，具体的格式如下表。

驱动器在运行状态中会检测是否有指令发送过来，超过 x 分钟没有收到 正确的数据则自动停机，因此在控制驱动器运转时候，最长时间指令间隔 不能少于 x 分钟。超时的时长 x 在寄存器 0x1035 设置，默认是 120 秒

上位机控制发出内容：

0	0xAA	起始码
1	0X00	地址
2	指令	0：关机 1：开机
3	设定转速	低字节
4	设定转速	高字节
5	0x00	
6	0x00	
7	0x00	
8	0x00	
9	0x00	
10	0x00	
11	0x00	
12	0x00	
13	0x00	
14	校验和	
15	0x55	结束码

驱动器发送出内容：

0	0xAA	起始码
1	0X01	从机地址
2	压缩机转速	低字节
3	压缩机转速	高字节
4	压缩机电流	低字节，精度为 0.1A
5	压缩机电流	高字节
6	母线电压	低字节，精度为 0.1v
7	母线电压	高字节
8	驱动器温度	20
9	故障代码	
10	工作状态	1：工作中；0：待机中
11	0x00	
12	0x00	
13	0x00	
14	校验和	
15	0x55	结束码

其中校验和为：chck=sum（字节 0， 1，…..，字节 13）；取低 2 位值，最大 255

程序控制代码示例

```
//适用于高品压缩机通讯协议
unsigned                                                                    char
GPSend_data[16]={0xAA,0x0,0x00,0xB8,0x0B,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xDA,0x55};    //发
送变量
unsigned                                                                    char
GPReceive_data[16]={0xAA,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x55}; //
接受变量
unsigned char
GPReceive_data_temp[16]={0xAA,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x55}
;
 //临时保持接受的变量,未计算校验和

void GPUartComFun(void){
  static byte Receive_item=0;
  char inByte, temp;

  //发送控制命令
   if(SysWK.Send_TimeCnt>UART_UPDATE_TIME)  //50*10ms 500ms 更新命令
     {
       //GPSend_data[2]=SysWK.Run;
       GPSend_data[2]=BldcCtl.WorkStatus;
       GPSend_data[3]= BldcCtl.SetSpeed&0x00FF;
       GPSend_data[4]= BldcCtl.SetSpeed>>8;
```

```
        GPSend_data[14]=0;

       for(int i=0;i<14;i++){

         if((GPSend_data[i]==0xAA)&&(i>0))GPSend_data[i] += 1;  //如果中间有 0xAA，则不要发 0xAA 过去，而是+1 防止驱动器以为是头信号出错
         if(GPSend_data[i]==0x55)GPSend_data[i] += 1;//如果中间有 0x55，则不要发 0x55 过去，而是+1，防止驱动器以为是尾信号出错
          GPSend_data[14] += GPSend_data[i];  //累计和
       }

       Serial.write(GPSend_data,16);  //发送高品协议
       SysWK.Send_TimeCnt=0;
       delay(10);
   }

   while(Serial.available() > 0)  {
       //判断是否是一帧新数据
       if(SysWK.Uart_Re_FreeTimeCnt>10){  //大于 100ms 间隔，一定是新的一帧。
          Receive_item=0;
        }
       SysWK.Uart_Re_FreeTimeCnt=0;
       SysWK.Uart0_NoRevCnt=0;

       inByte = Serial.read();    // 收到的数据

       GPReceive_data_temp[Receive_item]=inByte;
       Receive_item++;

      //如果一帧数据完成，则处理切割数据。
       if(inByte==0x55&&Receive_item>14){  //要判断是第 15 位的值,否则中间位也有可能有 0x55 的值,从而判断失败!!
          temp=0;
          Receive_item=0;
         //对接受的数据进行累计校验
          for(int i=0;i<14;i++){
            temp += GPReceive_data_temp[i];
          }

         if(temp==GPReceive_data_temp[14]){//如果接受到的数据正确则 copy 到 Receive_data 数组中

           for(int j=0;j<16;j++){
             GPReceive_data[j]=GPReceive_data_temp[j];
           }   //把收到的正确值 copy 到显示状态数组中

           //切片，把值传给相关变量
```

```
            BldcCtl.ReadRuningSpeed=GPReceive_data[3]*0x100+GPReceive_data[2];
            BldcCtl.ReadCurrent=GPReceive_data[5]*0x100+GPReceive_data[4];
            BldcCtl.ReadVoltage=GPReceive_data[7]*0x100+GPReceive_data[6];
            BldcCtl.ReadTemperature=GPReceive_data[8];
            BldcCtl.ReadError=GPReceive_data[9]; //,
            BldcCtl.ReadBldcStatus=GPReceive_data[10];

          }
        }
      }

      //--------------错误检测--------------------------------------------------
       if(SysWK.Uart0_NoRevCnt>3000){  //SysWK.Uart0_NoRevCnt 在 10ms 中断中++; 3000x10ms=30s 还没有数据则报
错
          SysWK.ErrorCode=ERR_UART0;
      }else{
          if(SysWK.ErrorCode==ERR_UART0)SysWK.ErrorCode=ERR_NONE;
      }

    //收一下数据

    //----------------------
}
```