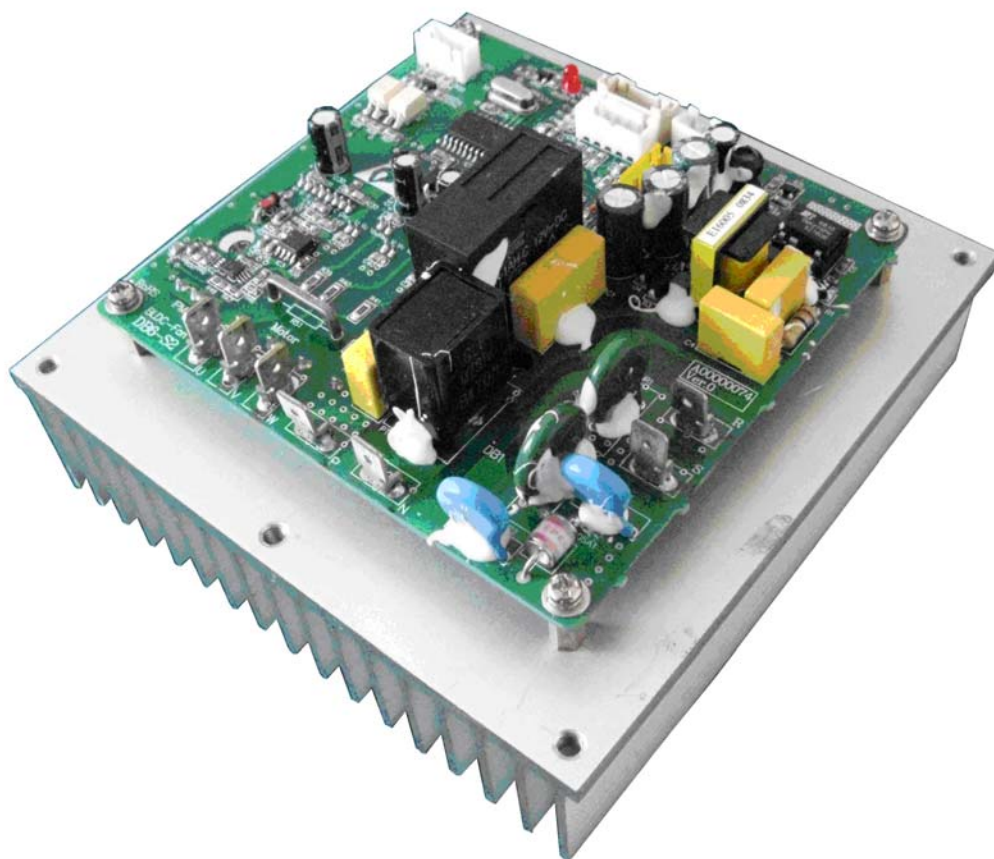




BLDC 驱动模块 产品手册



青岛斑科变频技术有限公司

地址：青岛保税区上海路 34 号 3 段 5 层

电话：0532-86959564 传真：0532-86959563

网址：www.qdbantek.com

e-mail: qdbkxs@126.com

目 录

1 变更记录	2
2 标准规范	3
3 功能框图	4
4 通讯协议	5
5 故障、保护及运行状态	8
6 安装及配线	9
附录 A: CRC16 校验例程	11
附录 B: 可选配件	13

1. 变更记录

[illegible]

2. 标准规范

驱动模块型号		DB6-S2
电路 拓扑	整流器	单相二极管整流器
	逆变器	三相 IPM 逆变器
	驱动方式	带位置传感器的优化 180° 正弦波驱动
供电 电源	电压	单相 220V ±20%
	频率	50/60Hz ±5%
PWM 控制 方式	运行范围	150rpm~1000rpm
	载波频率	5kHz
	死区时间	2.0us
	输出电流	最大 6Arms
保护		短路、过电流、欠/过电压、电机堵转、电机过速、电机失速、过热 ^[1] 位置信号错、无位置信号、运行方向错、电流检测故障、感温器故障 ^[1] 、通讯故障、
与上 位机 通讯	方式	半双工异步通讯
	波特率	1200 bps
	格式	8-N-1（8 位数据位、无校验位、1 位停止位）
功能框图		图 1
EMC 及 EMI		为了抑制 EMI，提高整机的 EMC，用户应在模块的输入端接入噪声滤波器，并做测试，达到国标要求
结构	散热方式	自带散热器，需配风扇，强制风冷（为保证散热效果，用户应合理设计散热结构）
	尺寸（mm） 长×宽×高	130×130×71
	重量（Kg）	0.62
环境	运行环境温度	-20 到 60℃ 30 到 80%RH（不结露，不结冰）
	存储温度	-20 到 80℃ 25 到 85%RH（不结露，不结冰）
	振动	5.9m/s ² （0.6g）以下（10 ~ 55Hz）
适用马达		DMSB18P80AS1（松下） UGBTEF-13L31X（日电）

注：1. 有的模块没有此功能。

3. 功能框图

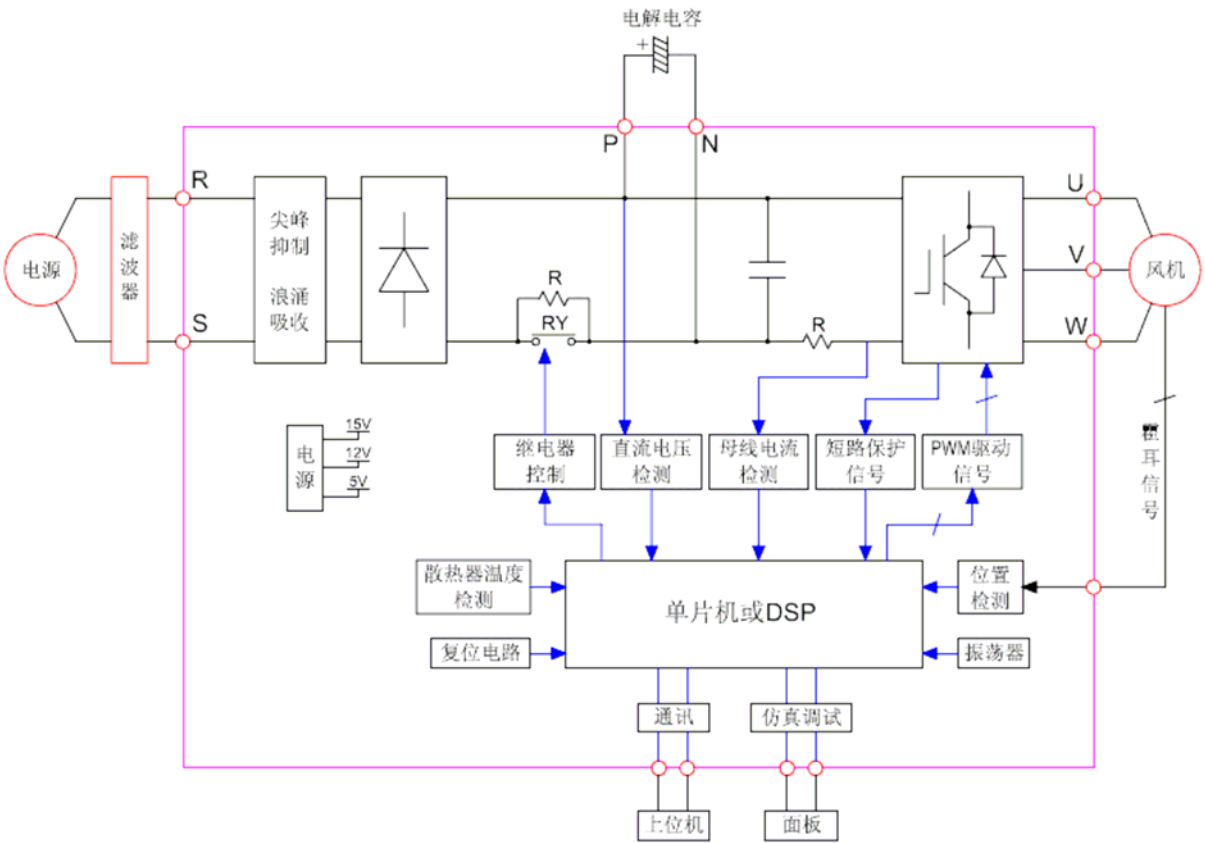
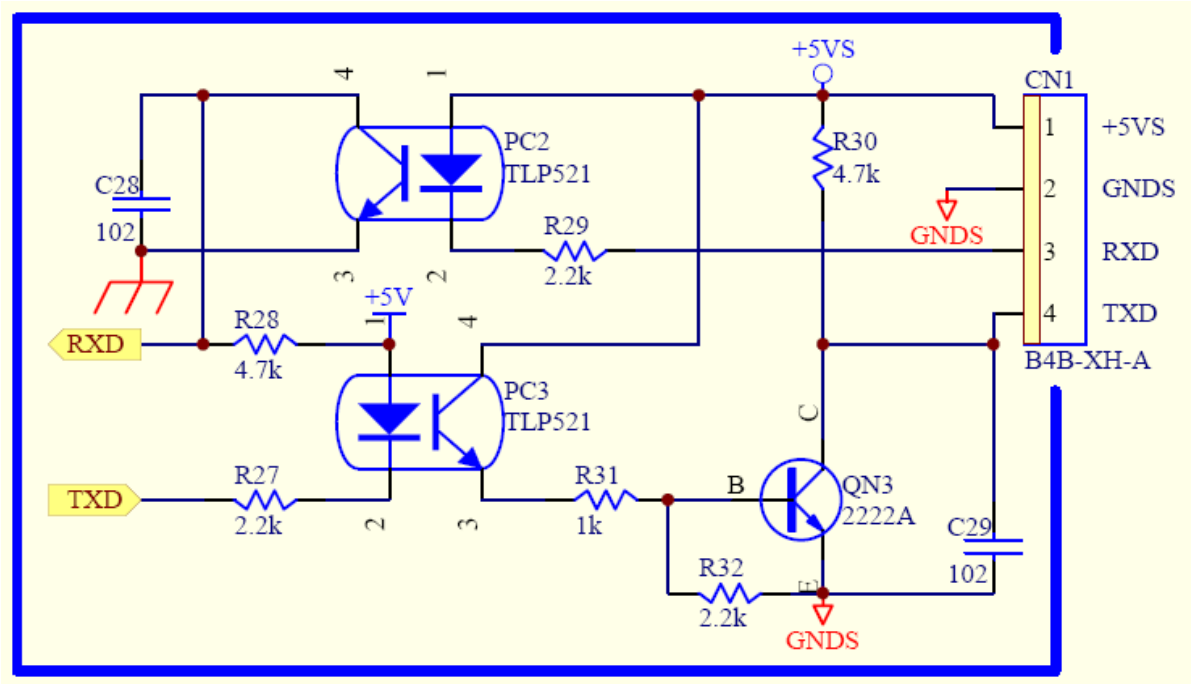


图 1

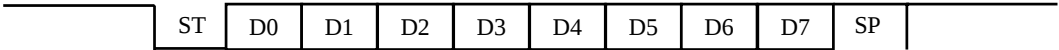
4. 通讯协议

4.1 通讯电路



4.2 通讯类型

条目	内容
方式	半双工异步通讯
波特率	1200 bps
起始位	1 位
数据位	8 位
校验位	无
停止位	1 位



ST : 起始位
D0 - D7 : 数据位
SP : 停止位

4.3 数据帧格式

有两种数据帧格式：上位机→模块，模块→上位机。

4.3.1 上位机→模块

表 1 数据帧格式：上位机→模块

序号	名称	内容
1	识别码	55H: 上位机→模块（固定）
2	风机转速指令 a（低字节）	字节 b 的最高位代表旋转方向（0：逆时针；1：顺时针） 转速 R（rpm）=(b&7FH)*256+a
3	风机转速指令 b（高字节）	
4	应用类型 ID	根据不同厂家应用的不同，确定不同的 ID 号
5	保留	00H
6	保留	00H
7	CRCL	CRC16 校验（低字节）
8	CRCH	CRC16 校验（高字节）

注：CRC16 校验见附录。

4.3.2 模块→上位机

表 2 数据帧格式：模块→上位机

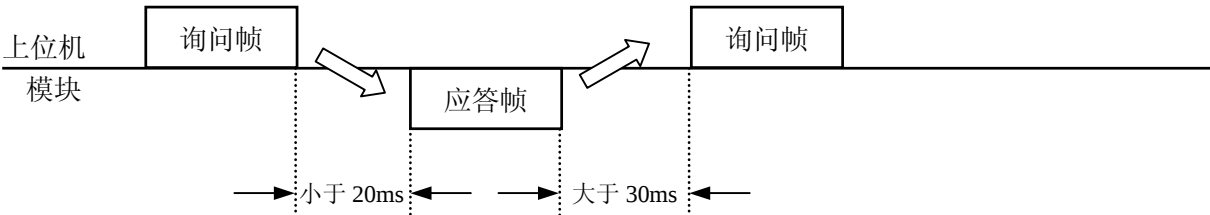
序号	名称	内容
1	识别码	AAH: 模块→上位机（固定）
2	风机实际转速 a（低字节）	字节 b 的最高位代表旋转方向（0：逆时针；1：顺时针） 转速 R（rpm）=(b&7FH)*256+a
3	风机实际转速 b（高字节）	
4	应用类型 ID	驱动器当前应用类型 ID 号
5	输出电流 c	输出电流峰值 $I_{out-peak}$ （A）=c/10
6	故障、保护及运行状态	详见第 5 章
7	CRCL	CRC16 校验（低字节）
8	CRCH	CRC16 校验（高字节）

注：CRC16 校验见附录。

注：应用类型 ID 是根据各个厂家的不同应用，在调试完后，由我公司针对这一应用分配的 ID 号。厂家在正式生产时，必须保证上位机发送的应用类型 ID 与我公司分配的一致。

4.4 传送约定

正常情况下，上位机主发，模块接收到正确的上位机数据后，在 20ms 内应答。上位机在收到模块的应答后，应在 30ms 后进行下一帧询问。



如果模块接收的上位机数据异常，模块将不会应答。此时，上位机应以固定周期进行下一次询问。

如果上位机接收的模块应答数据异常，上位机应以固定周期进行下一次询问。
推荐的最小固定周期为 200ms。

4.5 通讯异常

如果在 30s 内，模块接收的上位机数据持续异常，就判断上位机不正常，模块就会停机。

4.6 异常停机的再启动

异常停机后，模块会把停机原因发送给上位机。上位机在收到停机原因后，发送 10 次转速指令 0 给模块。模块在收到 10 次转速 0 后，会清除故障。此时，上位机可发送新的转速指令给模块，如果保护延时到，模块将重新启动。

5. 故障、保护及运行状态

如 4.3.2 章所述，模块应答帧的第 6 个字节表示模块的故障、保护及运行状态。

如果 1 小时内故障和保护的次数达到 6 次，模块将被锁定，即使收到转速指令也不会运行。此时，须断电，重新上电。

表 3 模块的故障、保护及运行状态

类别	代码	LED 闪烁次数	描述	原因分析
运行状态	00H	一直闪	模块运行正常	
	01H	一直闪	过压失速	1. 输入电压过高 (>271V)
	02H	一直闪	过流失速	1. 风机负载过重
	05H	一直闪	过流降频	1. 风机负载过重
	07H	一直闪	过热降频	1. 模块散热效果不好; 2. 环境温度太高;
故障	1EH	24	电流检测故障	1. 模块故障
	23H	26	感温器故障	1. 感温器故障
	28H	1	通讯故障	1. 上位机发送故障
	49H	14	位置信号错故障	1. 霍耳传感器故障 2. 风机电源线 UVW 反接
	4AH	15	无位置信号故障	1. 霍耳传感器故障
	52H	28	eprom 故障	1. 模块 EEPROM 故障
	53H	29	ID 错	1. 上位机发送的 ID 不在正常范围之内
保护	34H	5	欠压保护	1. 输入电压过低 (<154V)
	35H	6	过压保护	1. 输入电压过高 (>283V)
	39H	3	过流保护	1. 风机负载过重; 2. 风机加速过快;
	3AH	4	短路保护	1. 风机电源线短路; 2. 风机短路;
	45H	7	过热保护	1. 模块散热效果不好; 2. 环境温度太高;
	2AH	12	电机堵转保护	1. 风机故障
	29H	11	电机失速保护	1. 风机电源线 UVW 反接 2. 霍耳传感器故障
	2BH	13	电机超速保护	1. 霍耳传感器故障
	4BH	16	运转方向错保护	1. 风机电源线 UVW 反接 2. 霍耳传感器线反接

LED 指示灯说明如下：

停机模式：常亮；

运行模式：灭 0.1s，亮 0.1s；

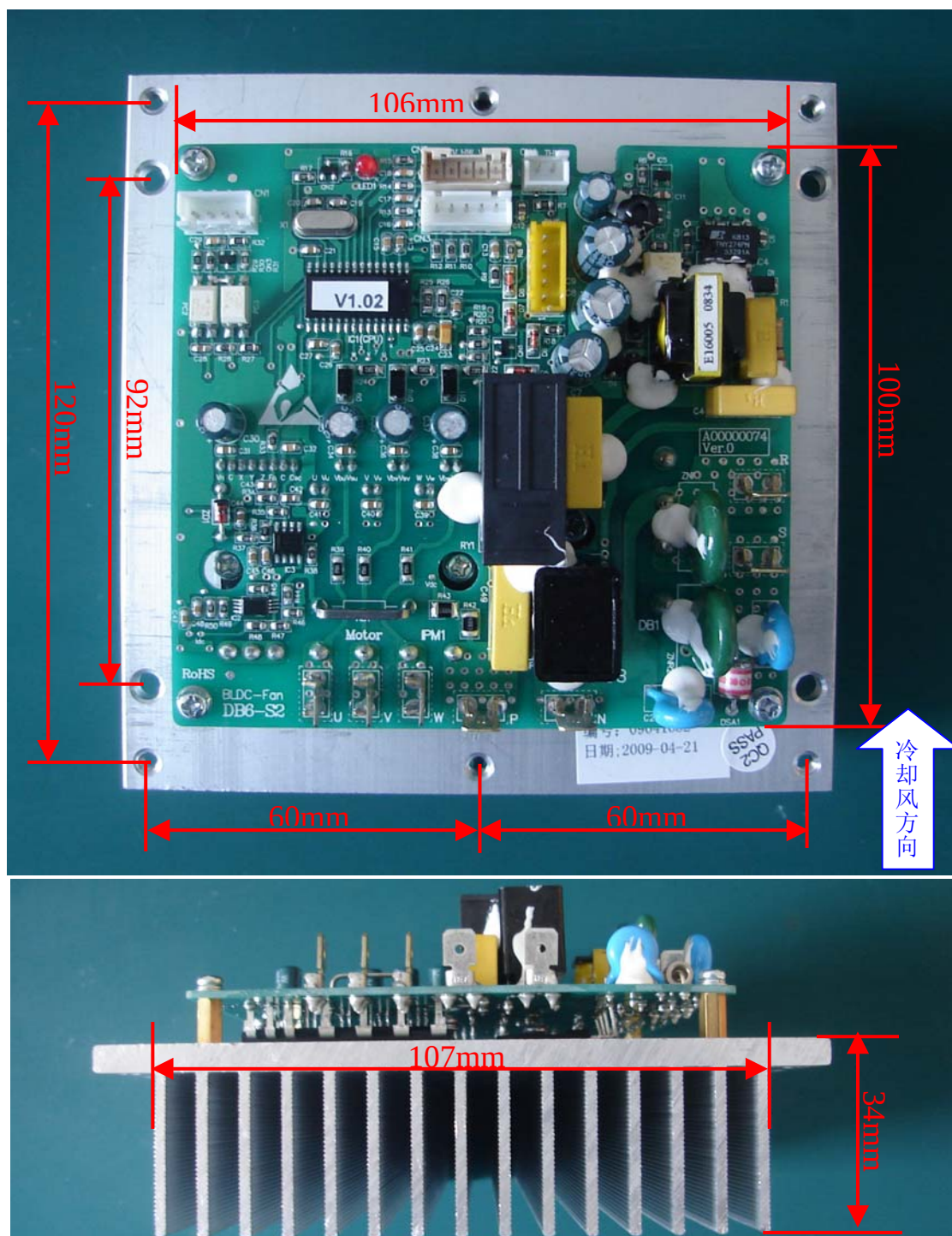
故障模式：灭 0.3s，亮 0.3s。

6. 安装及配线

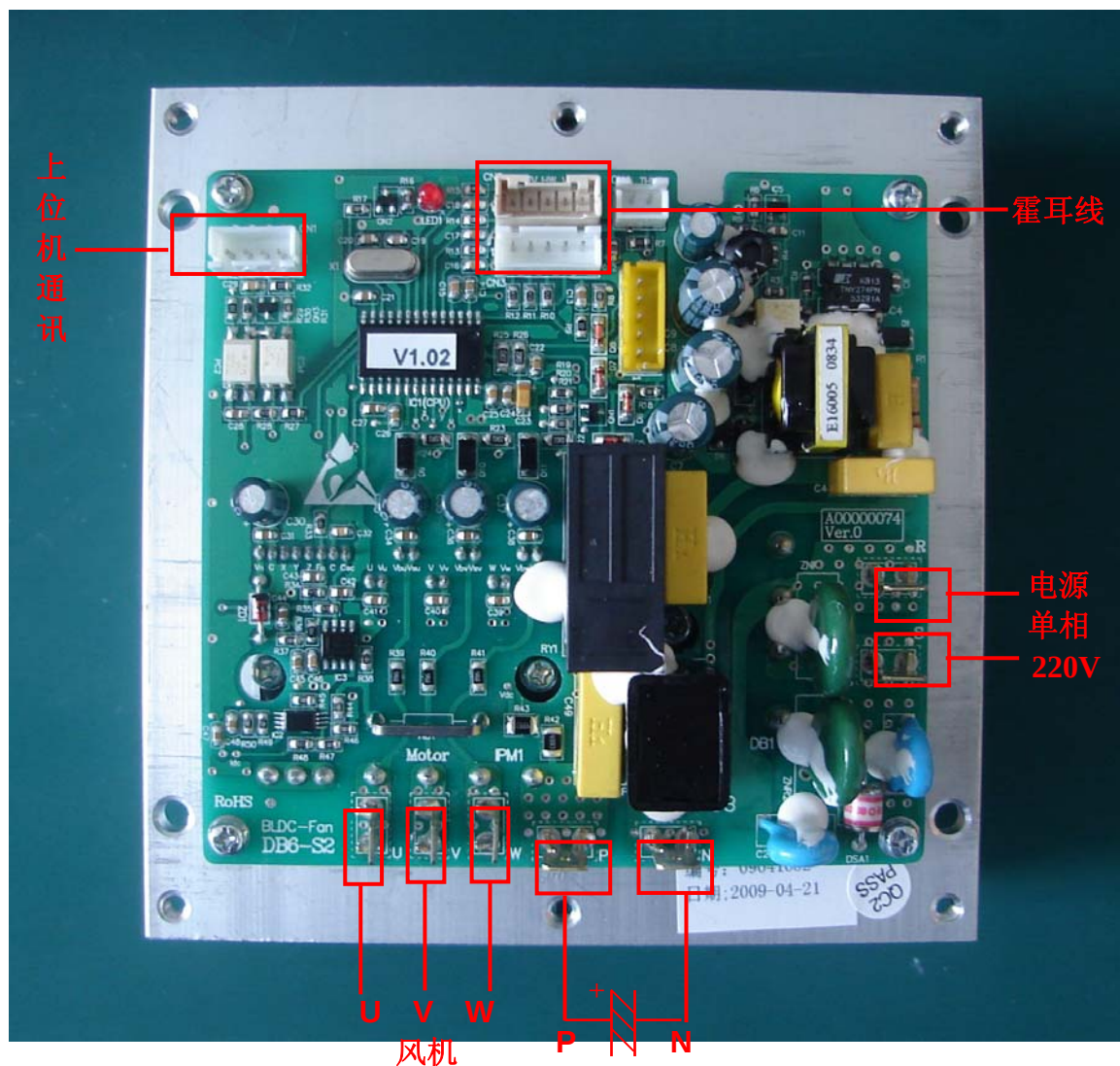
6.1 DB6-S2 安装及配线

6.1.1 尺寸及安装

模块采用 6 颗 M4 螺钉或 4 颗 M4 螺钉固定，具体尺寸见下图。



6.1.2 配线



铝电解电容推荐：高温 105℃品，两个 560 μ F 或 680 μ F 并联。

附录 A: CRC16 校验例程

例程 1: (直接生成法)

```
unsigned int CRC16(unsigned char *puchmsg, unsigned int usDataLen) //进行CRC校验的程序
{
    unsigned int uchCRC;
    unsigned int uIndex_x, uIndex_y;

    uchCRC=0xFFFF; //CRC寄存器初始化

    if (usDataLen>0)
    {
        for (uIndex_x=0; uIndex_x<=(usDataLen-1); uIndex_x++) //访问所有要校验的数据
        {
            uchCRC=uchCRC^(unsigned int) (*puchmsg);
            puchmsg++;
            for (uIndex_y=0; uIndex_y<=7; uIndex_y++)
            {
                if ((uchCRC&0x0001) !=0 )
                {
                    uchCRC=(uchCRC>>1)^0xA001;
                }
                else
                {
                    uchCRC=uchCRC>>1;
                }
            }
        }
    }
    else
    {
        uchCRC = 0;
    }

    return uchCRC;
}
//
```

例程 2: (查表法)

```
unsigned int CRC16(unsigned char *puchmsg, unsigned int usDataLen) //进行CRC校验的程序
{
    unsigned char uchCRCLo=0xFF;
    unsigned char uchCRHi=0xFF;
    unsigned char uIndex;

    while(usDataLen--)
    {
        uIndex=uchCRCLo^( *puchmsg++ );
        uchCRCLo=uchCRHi^auchCRCLo[uIndex];
        uchCRHi=auchCRHi[uIndex];
    }

    return ((uchCRHi<<8) | uchCRCLo);
}
```

```

/* Table of CRC values for low-order byte */
const unsigned char auchCRCLo[] =
{ 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
  0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
  0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
  0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
  0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
  0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
  0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
  0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
  0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
  0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
  0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
  0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
  0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
  0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
  0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
  0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
  0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
  0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
  0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
  0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
  0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
  0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
  0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
  0x80, 0x41, 0x00, 0xC1, 0x81, 0x40 } ;

/* Table of CRC values for high-order byte */
const unsigned char auchCRCHi[] =
{ 0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06,
  0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD,
  0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
  0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A,
  0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC, 0x14, 0xD4,
  0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
  0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3,
  0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,
  0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
  0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29,
  0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED,
  0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
  0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60,
  0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67,
  0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
  0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,
  0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E,
  0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
  0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71,
  0x70, 0xB0, 0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92,
  0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
  0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B,
  0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B,
  0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
  0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42,
  0x43, 0x83, 0x41, 0x81, 0x80, 0x40 } ;

```

附录 B：可选配件

1. 外接电容



2. 通讯适配器

通讯适配器可实现模块与上位机之间的通讯转换，通讯格式有I²C、RS485、RS232。

通讯适配器最多可连接 4 个模块。

有关适配器的详细说明，请详见《通讯适配器使用手册》

