

单相电力线载波调制解调器 BWP08

产品说明书



通讯速率：100bps-600bps, 工作电压：5-12V

目标应用：智能家居、灯饰控制、智能开关

(试行)

深圳市必威尔科技有限公司

SHENZHEN BITWELL SCIENCE&TECHNOLOGY CO.,LTD

一、产品简介

BWP08 嵌入式电力线调制解调器（电力线载波模块、电力线 MODEM）是深圳市必威尔科技有限公司（www.bitwell.cn）全力打造的专业电力线载波产品，其核心芯片采用国际著名公司的专用电力载波集成电路，配合必威尔科技专门研发的通讯算法及电力线接口信号驱动电路，使得产品具有通信速率高，通讯可靠，抗杂波干扰能力强，通讯距离远等特点，是专门为适应中国国内电力线应用环境而研发的高性能电力线载波通讯产品。

BWP08 是专门针对智能家居及灯饰控制市场研发设计，产品具有体积小，通讯可靠，通讯频点可调，功能可定制特点，可以广泛应用于智能家居，灯饰控制，家电控制等领域。

BWP08 电力载波模块采用 5-12V 宽电压设计，载波波特率 100bps-300bps 可调，有多种接口方式可供选择，包括 UART、SPI 等二种接口，可以方便地与单片连接进行数据通讯，方便用户进行二次开发，串行接口波特率可由用户设定，共有四种波特率可设置：1200bps、2400bps、4800bps、9600bps。

BWP08 电力载波模块提供半双工通信功能，可以在 220/110V，50/60Hz 电力线上实现局域通信，也可以用于直流线路或者无电导体。该模块可以自由配置电力线上数据通讯模式，有两种通讯模式可供用户选择：固定字节长度传输及固定帧长度传输，关于这两种通讯模式的详细介绍，请参照相关章节介绍。该模块为用户提供了透明的数据传输通道，数据传输与用户协议无关，模块采用扩频编码方式，抗干扰能力强，数据传输可靠。通讯过程中，由用户通讯协议验证数据传输的可靠性。在同一台变压器下，多个 BWP08 模块可以连接在同一条电力线上，在主从通信模式下，模块分别单独工作，不会相互影响。

主要性能特点：

- 工作电源：5-12VDC
- 接口类型：TTL 电平串行接口 (UART)，SPI，半双工通讯
- 线上载波速率：100bps、200bps、300bps、400bps、500bps、600bps，由用户设置
- 串行接口速率：1200bps、2400bps、4800bps、9600bps，由用户设置。
- 工作环境：220VAC/110VAC，50/60Hz，直流线路，无电导体
- 通讯距离：≤200m，（轻负载条件或者直流线路情况下，通讯距离可能大于 200m）
- 数据传输类型：固定字节长度传输（1-32 字节）、固定帧长度传输（32-256 字节）
- 电力线载波频率：65KHz、88KHz、113 KHz、140KHz、170KHz、202KHz，可配置
- 调制解调方式：DSSS（直序扩频）
- 工作温度：-20℃~+70℃

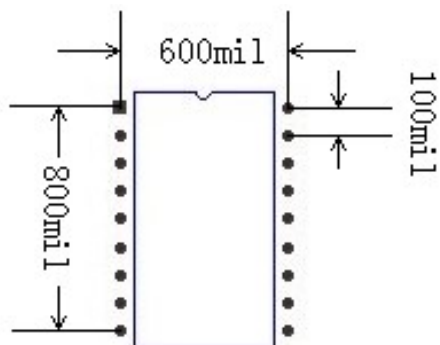
- 外形尺寸：DIP18 双列直插封装(宽度：600mil，脚距：100mil)

主要应用

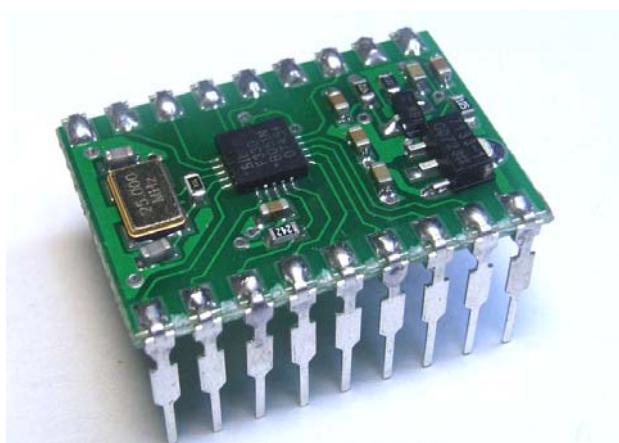
目标应用：短距离电力载波应用，包括智能家电控制，灯饰控制，家庭安防控制，远程灯光控制，空调控制，低速率通信网络，消防及保安系统，舞台灯光音响控制等

二、外形及引脚定义

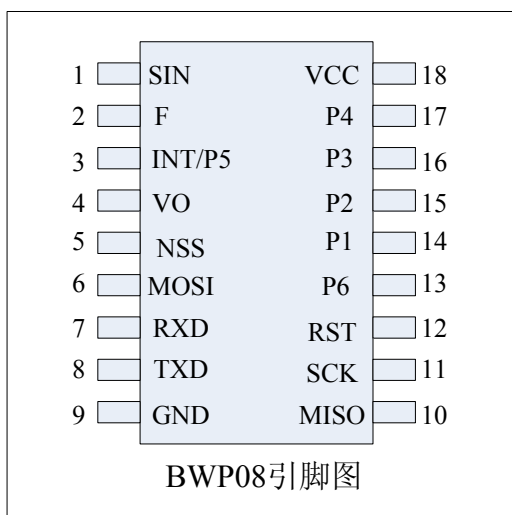
BWP08 电力载波模块采用双列直插式设计，体积小，采用全贴片式元件，可以方便地嵌入用户的应用系统中，外观与普通集成电路一样。具体尺寸如图一所示。(单位：mil)



图一：BWP08 电力载波模块外形尺寸图



图二：BWP08 电力载波模块产品图片



BWP08 引脚功能定义

序号	名称	输入/输出	电气参数	功能
1	SIN	输入	$\leq 1.4V_{pp}$	接收载波信号输入
2	F	输出	$\leq 3.3V$	模块滤波电容，100Uf
3	INT/P5	通用 I/O	TTL 电平， $\leq 5V$	SPI 接口的中断信号输出，低电平有效/通用 I/O，外加上拉电阻
4	VO	输出	$\geq 3V_{rms}$	载波信号发送输出端口
5	NSS	输入	TTL 电平， $\leq 5V$	SPI 接口的片选引脚，低电平有效，外加上拉电阻
6	MOSI	输入	TTL 电平， $\leq 5V$	SPI 数据输入，外加上拉电阻
7	RXD	输入	TTL 电平， $\leq 5V$	串口数据输入，（模块接收数据）
8	TXD	输出	TTL 电平， $\leq 5V$	串口数据输出，（模块发送数据）
9	GND	电源	电源地	电源地
10	MISO	输出	TTL 电平， $\leq 5V$	SPI 数据输出，外加上拉电阻
11	SCK	输入	TTL 电平， $\leq 5V$	SPI 时钟输入，外加上拉电阻
12	RST	输入	TTL 电平， $\leq 5V$	模块复位引脚，低电平有效。模块内部已有复位电路，用户也可以外接复位，复位时间大于 300ms，此脚悬空时，需外接一只 $\leq 10Uf$ 的电容
13	P6	通用 I/O	TTL 电平， $\leq 5V$	通用 I/O，用户不可用
14	P1	通用 I/O	TTL 电平， $\leq 5V$	通用 I/O，用户不可用

15	P2	通用 I/O	TTL 电平, $\leq 5V$	通用 I/O, 用户不可用
16	P3	通用 I/O	TTL 电平, $\leq 5V$	通用 I/O, 用户不可用
17	P4	通用 I/O	TTL 电平, $\leq 5V$	通用 I/O, 用户不可用
18	VCC	电源	$\geq 5V, \leq 12V$	模块工作电源, 5-12V 直流电源

三、 功能描述

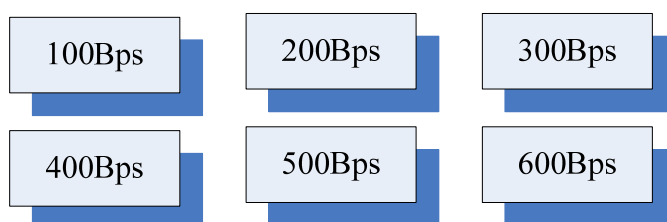
BWP08 主要针对智能家居等需要在小范围内进行电力载波通讯的场合设计, 通讯距离在 200m 以下的电力线载波应用, 采用 BWP08 将非常合适, 或者在直流环境下应用, 通讯距离可以大幅度提高。该模块载波速率、载波频率、串口速率、通讯数据模式等都可以由用户设置调整。在实际应用中, 通讯速率越低, 通讯越可靠, 传输距离越远, 而且可以利用载波频率可设置的特点, 在一条线路上, 可以同时允许多条通讯链路同时工作, 各链路之间互不干扰。

1、载波数据速率可配置

BWP08 电力载波模块的载波数据速率是可配置的, 目前模块支持 100bps、200bps、300bps、400bps、500bps、600bps 共 6 种波特率, 在实际应用中, 用户可根据实际需要及线路负载情况, 灵活地配置线上载波波特率, (可通过我公司提供的专用配置软件进行参数配置)。

在应用中, 线上波特率越低, 则通讯越可靠, 抗干扰能力越强, 通讯距离也越远; 如果用户线路状况比较好, 比如电力线负载比较小、干扰比较轻, 或者是直流线路等, 就可以选择比较高的波特率进行通讯; 当线路负载比较重、干扰比较强、或者想进行更远距离通讯时, 可以选择较低波特率进行通讯。比如在楼宇灯光控制或者断缆监控应用中, 用户更需要通讯可靠及通讯距离更远, 对通讯速率的要求比较低, 就可以选择较低波特率进行通讯; 在通讯距离比较近的情况下, 可以选择较高的通讯速率。

应用提示: 载波波特率可配置,



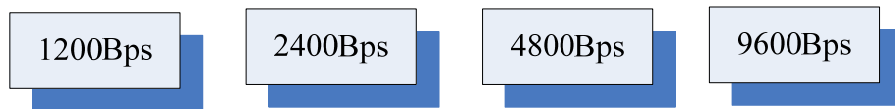
波特率越高, 传输速率越快, 波特率越低, 传输越可靠

2、串行接口速率可配置

BWP08 共有两种接口供用户选择，UART、SPI，其中串口通讯波特率是可以设置的，目前共有 1200bps、2400bps、4800bps、9600bps 共 4 种波特率可由用户自由配置，配置方法同样通过我公司提供的专用配置软件进行参数配置，可在我公司网站上自由下载。有了串口波特率可配置功能后，用户就灵活地根据自身系统的特点配置串行接口通讯速率，在实际应用中，某些高速率单片机系统比较难于产生 1200bps 以下的串口速率，这样就使模块难于与这些单片机系统进行连接，必须使用软件模块串口，造成许多应用上的不便，所以 BWP08 的串口波特率可以灵活配置，极大地提高了模块的易用性，方便用户使用。

串口波特率的高低与载波数据速率无关，在载波速率一定的情况下，用户可以选择不同串口速率与载波模块进行通讯，比如当载波模块的载波数据速率为 300bps 时，用户串口速率可以在四种波特率中任选一种，比如串口速率可以选择 1200bps 或者更高。这完全由用户根据系统需要进行选择。

应用提示：串口波特率可配置，



根据应用需求，灵活配置串口波特率

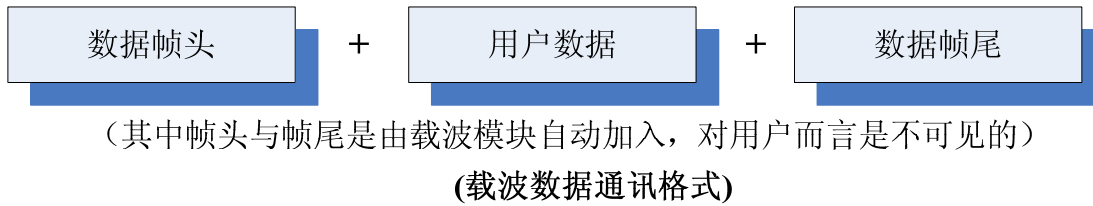
3、数据传输类型可配置

BWP08 数据传输类型有两种，一种是固定字节长度传输（定长传输），一种是固定帧长度传输（定帧传输），这两种传输方式各有优点，用户可根据实际需要灵活选用。

固定字节长度传输（定长传输）：接收模块每次收到数据帧头后，只接收预设长度的用户数据，由于不存在帧尾丢失导致模块一直处于接收状态，所以载波传输时间可以精确计算，提高了载波传输效率。比如用户将模块设置为 6 字节固定长度传输类型，那么在每次接收时，该模块只接收 6 个字节，然后等待发送方进行下一轮传输。

固定帧长度传输（定帧传输）：接收模块每次可以接收小于或者等于预设帧长度的数据，但在数据接收时，数据帧尾丢失，那么接收模块必须收满预设最大帧长度为止。比如某模块预设为 32 字节固定帧长度传输类型，数据传输长度可以在 1-32 字节之间变化，但如果数据在传输过程中受得干扰，导致数据帧尾丢失，那么接收模块将无法判断用户数据何时结束而一直处于接收状态，并一直收满 32 字节为止。

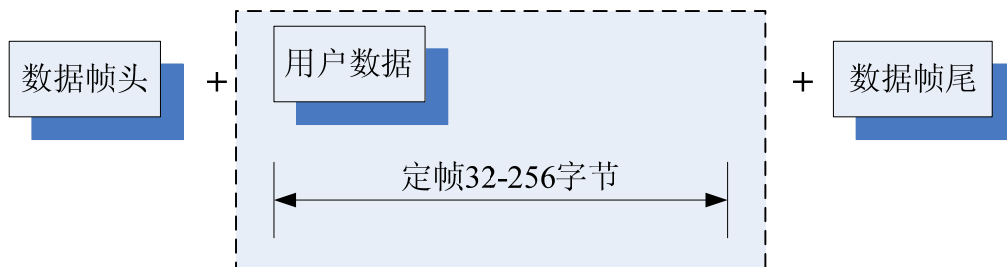
下图为载波数据传输的一般格式：



关于定帧与定长传输的详细说明

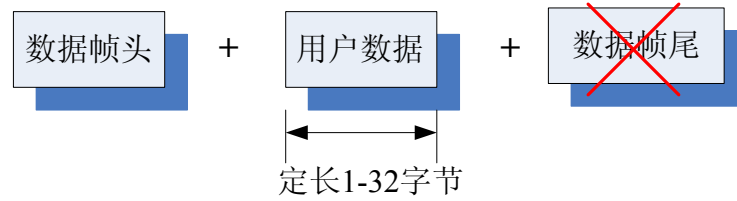
由于电力载波是采用公共信道进行数据通讯，电力线上各种谐波比较强，干扰比较大，通讯过程极易受到各种干扰，虽然 BWP10A 载波模块采用直序扩频方式进行数据编码，以提高数据传输可靠性，但仍然难以避免受到各种干扰源的干扰。

在数据传输过程中，数据帧头、用户数据、数据帧尾等都有可能受到干扰，数据帧尾被干扰而丢失，接收模块将无法判断用户数据何时结束，从而导致接收模块一直处于接收状态，一直收满设定的最大帧长度为止。我们预设了 32、64、96、128、160、192、224、256 共 8 种帧长度供用户选择，用户可根据自身的应用，选择不同的帧长度。在实际应用中，用户根据系统每次传输最长字节数选择相邻的帧长度。



图三：定帧传输示意图

用户还可以通过“定长方式”传输数据，当用户的数据长度比较短，每次传输的字节数比较固定时，就可以使用这种方式。比如一次只需要传输几个字节或者十几个字节，在这种情况下，即使采用定帧 32 字节的传输方式，一旦传输出错，接收模块仍然需要收满 32 字节，比较浪费时间，这样就可以采用定长（定字节长度传输）传输方式。比如系统一次传输的字节最多不超过 8 个字节，一般是 6-8 个字节，在这种情况下，可以将传输类型定义为 8 字节“定长传输”，这样即使传输出错，模块最多只接收 8 个字节，系统对数据进行验证后，如果数据正确则执行，数据错误则丢弃或者通知对方重发，极大地减少了系统资源的浪费，提高了系统执行效率。



图四：定长传输示意图

采用“定长”及“定帧”传输的主要目的是为了提高载波传输效率，降低系统额外开销，从而提高载波应用系统的实时性，由于定长传输无须发送帧尾，用户在单位时间内发送的数据可以更多，相当于加快了用户数据发送速率。在实际应用中，采用“定长”传输可以有效地提高系统的执行效率，所以推荐用户在可能的情况下，尽可能采用“定长”方式传输数据。

4、载波频率可设置

BWP08 共有 6 种频率可供用户设置选用，载波频率的选择是通过设置软件进行的，用户设置好通讯频率以后，外围的滤波电路必须做相应的配合，如果线路板面积或者成本许可，载波输入电路可以采用多级滤波电路，可以显著提高滤波效果，提高接收可靠性。

在一个系统中，用户如果需要多个系统同时工作，或者需要组成全双工的通讯模块，那么就可以采用多频点载波方式来实现，完全由用户根据自己系统及应用的要求，灵活选择。

序号	载波频率 (KHz)	滤波电感(uH)	滤波电容(pF)	谐振频率 (KHz)
1	202.9	270	2200	206.5
2	170.0	270	3300	168.6
3	140.0	270	4700	141.3
4	113.2	270	6800	117.5
5	88.5	270	12000	88.4
6	65.8	270	22000	65.3

表一：BWP08 载波频率及 LC 滤波参数表

为避免不同载波频率通讯时相互干扰，建议在实际使用中，可选择相隔比较远的频率同时工作，比如可以选择 65.8KHz 与 113.2KHz 一起工作，这样可以彻底避免相互干扰。或者通过增加输入滤波级数的方法，提高滤波效果，进一步降低邻频的相互干扰。

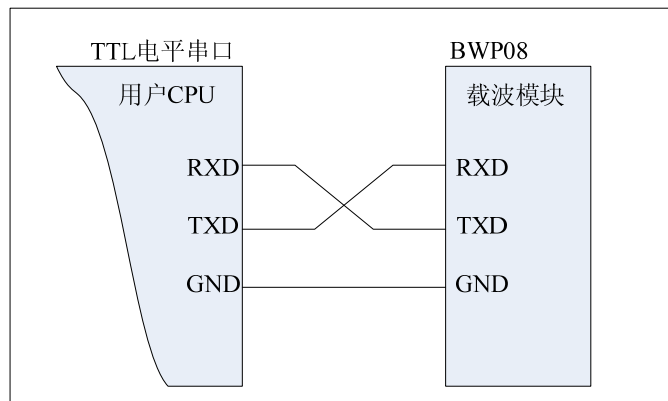
5、宽电压供电 (5-12V)

BWP08 采用宽电压供电，用户可以输入 5—12V 直流电压，静态情况下，工作电流为 30mA，在载波发送情况下，模块工作电流正常为 30mA。

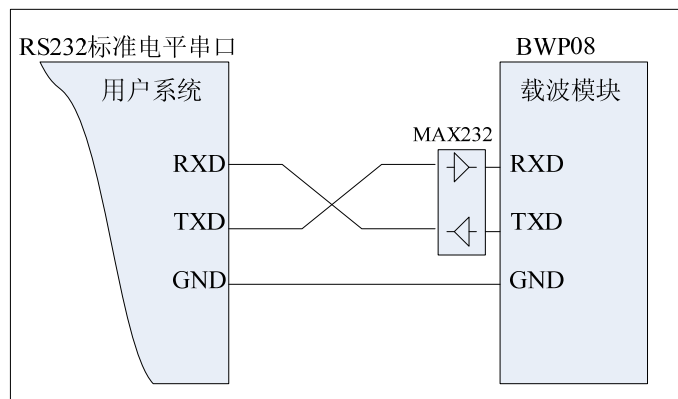
五、 应用指南

1、采用串行接口与用户系统连接

BWP08 电力载波模块使用 TTL 电平串口与用户系统进行连接，并使用交叉连接方式进行连接，通讯采用收、发、地三线制方式。当用户系统为 TTL 电平串口时，可以直接与模块进行交叉连接通讯，无须 RS232 电平转换，所以用户可以直接使用单片机的串行接口（UART）与载波模块进行连接通讯，当用户系统为标准 RS232 接口时，需要增加串口电平转换芯片进行电平转换，比如 MAX232 等芯片进行串口电平转换。



图六：TTL 电平串口连接示意图



图七：RS232 串口连接示意图

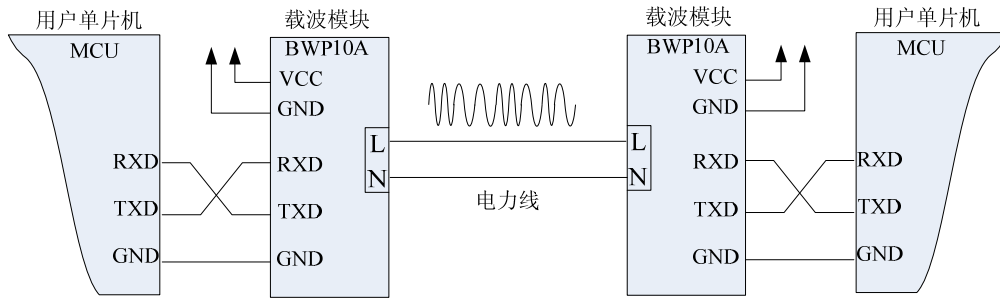
如下图所示，BWP08 与主控设备通过电力线相互通讯，由于 BWP08 为透明传输电力载波模块，它将串口收到时的数据实时向电力线转发，同时将电力线上收到的数据实时通过串口发送给用户系统，模块为单工通讯，串口波特率及载波波特率都可调，下面假设模块的串口波特率设置为 1200bps、载波波特率设置为 300bps，传输类型为定帧 32 字节传输，那么传输过程如下：

第一步：主机设置串口波特率为 1200bps，并通过串口将要发送的数据发送给 BWP08，模块收到数据后，将实时地以 300bps 的速率调制到载波信号上，然后再把载波信号耦合到电力线上进行传输。

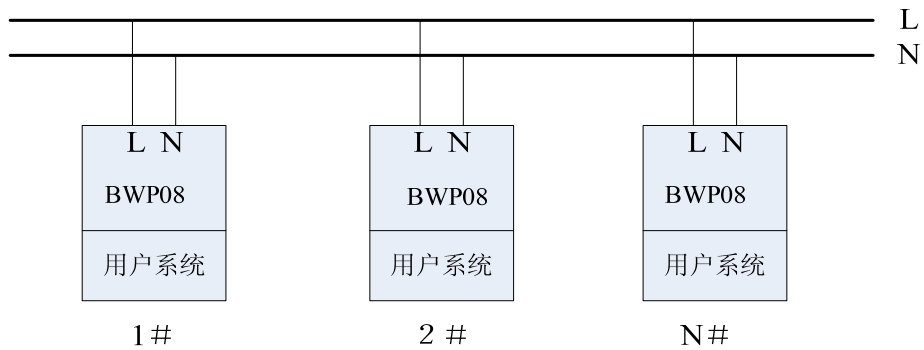
第二步：从机的 BWP08 载波模块将电力上的载波信号解调成用户数据，然后通过 TTL 电平串口

发送给从机设备。

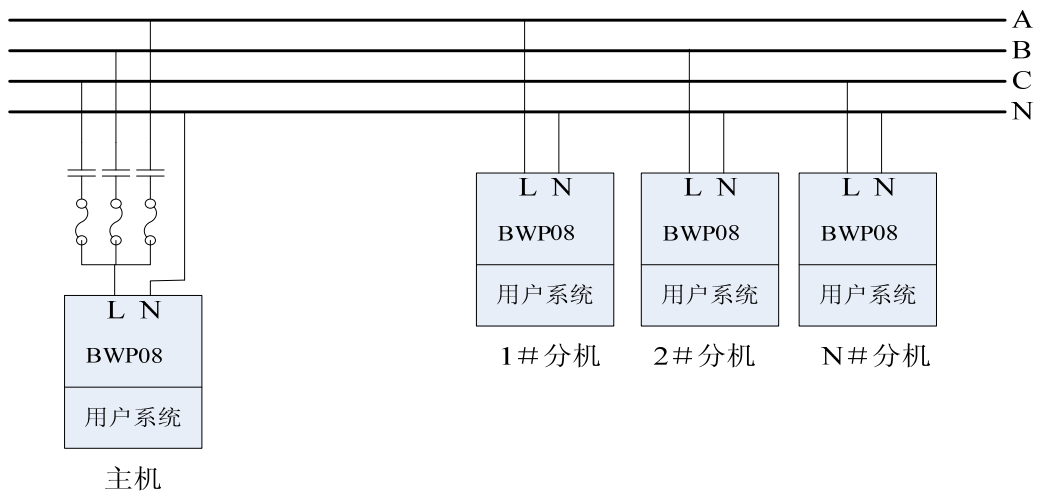
相反，从机设备也可以通过 BWP08 将数据通过电力线发送给主机端设备，对用户而言，BWP08 为用户建立起透明的数据传输通道，用户可以将主从设备当成电缆直接相连，而不需要考虑中间的电力载波实现的过程。用户按照主从通讯方式，正确将从机设备进行编址，就可以将多个以 BWP08 为核心的电力载波设备安装在同一条电力线上，它们可以分别工作，而不会相互干扰。用户通讯的可靠性由用户的通讯协议保证，BWP08 模块只提供无协议的数据传输通道。



图八：载波模块通讯连接示意图



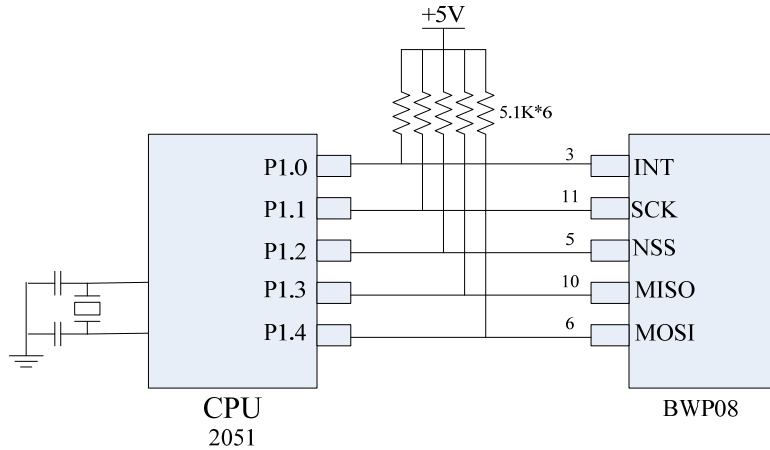
图九：电力载波单相连接示意图



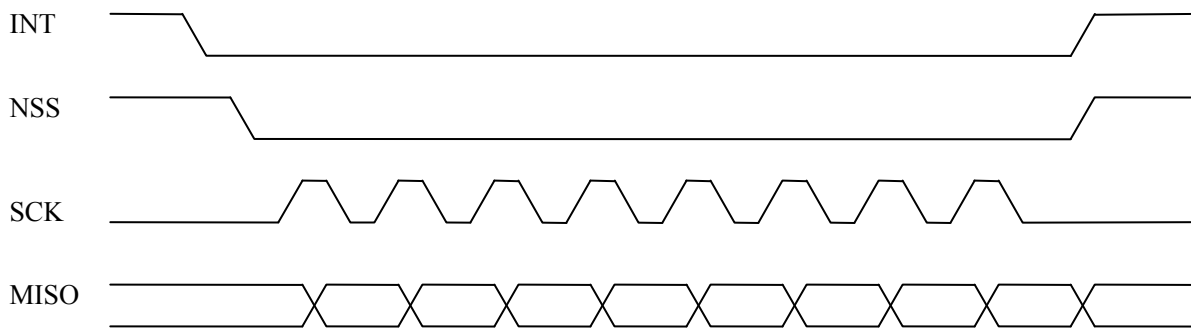
图十：电力载波三相连接示意图

2、采用 SPI 接口与用户系统连接

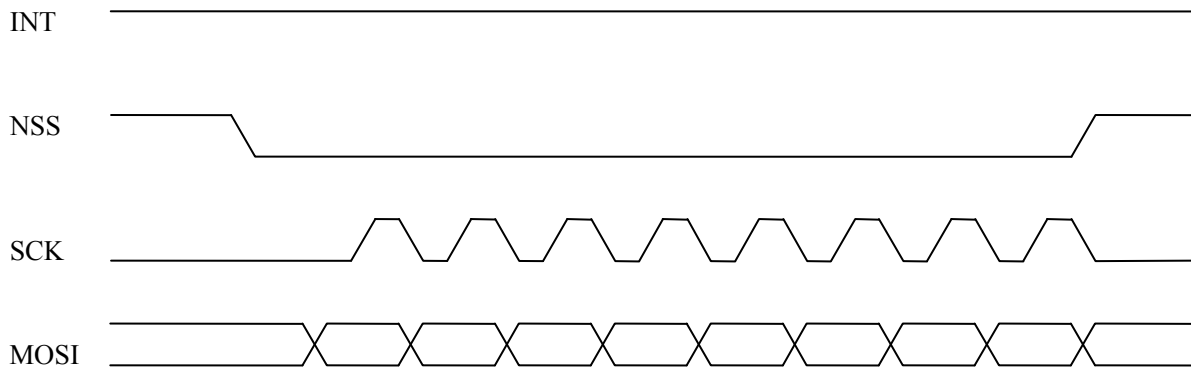
BWP08 模块采用标准 SPI 接口与用户系统交换数据，用户可以使用具有 SPI 接口的控制器与模块连接，进行数据交换，也可以用采用 CPU 的 I/O 模拟 SPI 接口的时序，系统应用拓扑图与串口应用一致，下面仅对 SPI 接口部分做详细说明。



SPI 连接示意图



SPI 读时序



SPI 写时序

SPI 接口示范例程

```
/******
```

此程序为单片机通过 SPI 接口与 BWP08 通信的程序.

单片机的 SPI 时序为软件模拟, 单片机为 STC12C2052

```
*****/
```

```
#include <STC12C2052AD.H>
```

```
#include <intrins.h>
```

```
#define uchar unsigned char
```

```
#define uint unsigned int
```

```
sfr  WD_CONTR=0x0E1;
```

```
sbit RLED = P1^5;
```

```
sbit TLED = P1^6;
```

```
sbit RUN  =  P1^7;
```

```
//下面引脚为 SPI 通信的必备引脚
```

```
sbit SCK  =  P3^3;
```

```
sbit MISO = P3^2;
```

```
sbit  MOSI = P3^4;
```

```
sbit  NSS  = P3^5;
```

```
sbit INT  = P3^7;
```

```
//下面为单片机串口的接收/发送缓冲区定义
```

```
static uchar data RcvComDataBuff[4];
```

```
static uchar data SndComDataBuff[4];
```

```
static uchar data * data rcvp = RcvComDataBuff;           //接收数据指针
```

```
static uchar data * data sdp = SndComDataBuff;           //发送数据指针
```

```
//下面为系统运行的标志
```

```
struct SystemF
```

```
{
```

```
    uchar BootF:1;           //Boot 完成标志
```

```
    uint RunCunt;           //运行指示灯闪烁计数器,800mS
```

```
    uchar SdF:1;
```

```
}System;

*****/

//这是一个延时程序

void Delay(uint j) reentrant

{
    for(;j>0;j--)
        {};
}

*****/

    定时管理函数

    void Time0(void) interrupt 1 using 1

    管理定时事件

*****/

void Time0(void) interrupt 1 using 1

{
    TL0 = 0x0CC;
    TH0 = 0x0F8;
    TR0 = 1;
    WD_CONTR=0x34;
    /*****/
    System.RunCunt++; //这是系统运行指示灯,不是必要的程序
    if(System.RunCunt == 2000)
    {
        RUN = ~RUN;
        //System.SdF = 1;
        System.RunCunt = 0;
    }
}

*****/
```

串口发送函数（非中断）

```
void SndComData(uchar *p,uchar len)
```

*p 为发送数据的地址指针，len 为
长度

```
*****/
```

```
void SndComData(uchar *p,uchar len) //串口发送数据的函数
```

```
{
```

```
ES = 0;
```

```
while(len) //发送
```

```
{
```

```
SBUF = *p;
```

```
p++;
```

```
len--;
```

```
while(!TI)
```

```
{
```

```
_nop_();
```

```
}
```

```
TI = 0; //清发送中断标志
```

```
}
```

```
ES = 1;
```

```
}
```

```
*****/
```

串行口接收数据的函数（中断）

```
void RcvComData(void) interrupt 4 using 1
```

接收的数据存放在 RcvComDataBuff 中

并且将 Com.RcvF = 1;

```
*****/
```

```
void RcvComData(void) interrupt 4 using 2 //接收 PLC 数据的函数
```

```
{
```

```
*rcvp = SBUF;
```

```
System.SdF = 1;

RI = 0;

}

/*****

    初始化函数

    void Boot(void)

    初始化端口，寄存器

*****/

void Boot(void)
{
    SCON = 0x50;          // SCON: mode 1, 8-bit UART, enable rcvr
    TMOD = 0x21;         // TMOD: timer 1, mode 2, 8-bit reload
    PCON = 0x00;
    IE = 0x92;
    IP = 0x10;
    TL1 = 0x0B8;
    TH1 = 0x0B8;
    TL0 = 0x042;
    TH0 = 0x0F5;
    P3M0 = 0x084;        //这是 STC 单片机的引脚模式定义,为的是符合 SPI 通信标准
    P3M1 = 0x038;
    TR1 = 1;            // TR1: timer 1 run
    TR0 = 1;            // TR0: timer 0 run
    SCK = 0;
    NSS = 1;
    System.BootF = 1; //Boot 完成标志
    System.SdF = 0;
}

/*****

    主函数
```

```
main()
*****/

main()
{
    uchar temp,i;
    Boot();
    while(1) //主循环
    {
        if(!INT) //INT=0,表明 BWP08 模块从电力线上接收到数据,INT=0 是
        通知单片机的信号
        {
            RLED = 0; //这是接收指示灯,非必要
            /*****关键代码,接收*****/
            temp = 0;
            i = 8; //一共 8 位数据
            NSS = 0; //这是 SPI 接口读数据的标准:先"片选"
            while(i)
            {
                SCK = 1;
                temp = (temp<<1)|MISO;
                Delay(10);
                SCK = 0;
                Delay(10);
                i--;
            }
            NSS = 1; //数据读取完毕,数据存在 temp 中
            /*****关键代码,接收 END*****/
            *sdp = temp;
            SndComData(sdp,1); //这是调试代码,将这个数据发送到串口,非必要的代码
            RLED = 1;
```



```
}  
if(System.SdF&(INT))           //这是单片机向 SPI 接口写数据的程序  
{  
    System.SdF = 0;  
    TLED = 0;  
    /*****关键代码,发送*****/  
    temp = *rcvp;               //待发送数据存在 temp 中  
    i = 8;  
    NSS = 0;  
    while(i)  
    {  
        if(temp&0x80)MOSI = 1;   //发送最高位  
        else MOSI = 0;  
        SCK = 1;  
        Delay(10);  
        SCK = 0;  
        Delay(10);  
        temp = temp<<1;         //左移  
        i--;  
    }  
    SCK = 0;  
    Delay(10);  
    NSS = 1;  
    /*****关键代码,发送 END*****/  
    TLED = 1;  
}  
}  
}
```

3、应用要点说明

- BWP08 的串行接口与 SPI 接口不可以同时使用。
- 载波模块出厂默认配置为：串口 1200bps、载波波特率 300bps、定帧 32 字节。
- 载波波特率越低，传输越可靠，传输距离越远。
- 串口波特率选择与载波波特率选择无关，两者没有任何关系。
- SPI 接口的连接方法，请参照附录的参考电路图及示范例程。

4、模块参数配置软件



图十一：模块配置软件界面

软件操作步骤

- 选择串口，然后点击“打开串口”按钮，如果串口没有被占用，那么会提示串口已打开提示窗口。
- 如果不知道模块当前的串口波特率设置，可以点击“自动搜索”按钮，软件开始从 1200bps、2400bps、4800bps、9600bps 依次进行探索，如果探索成功，那么串口波特率栏会显示当前模块通讯波特率，并在“当前配置”框中显示模块的当前参数配置，包括串口波特率、载波波特率、传输类型等，如果搜索不成功，那么在监视窗口提示不成功信息
- 如果知道当前模块的串口波特率，那么可以直接选择串口波特率，然后点击“读取参数”按钮，那么也可以读取当前模块的参数配置，信息提示与“自动搜索”结果相同
- 读取了模块的当前配置后，如果希望在当前的基础上进行修改，可以点击“参数导入”按钮，那么当前模块的配置信息会转入“参数设置”框中的对应参数区，这样就可以在原参数基础上进行修改了。
- 如果已知当前模块串口速率，那么也可以直接选择好串口并设置串口速率后，直接在“参数设

置”框中选择参数，然后点击“写器件”将参数通过当前串口下载至模块中

- 选择好参数后，点击“锁定”按钮，“参数设置”框中所有的参数都不可改变，直至点击“解锁”为止，这样就可以防止在批量烧录时误改参数的可能。
- 可以点击“文件导入”将当前配置保存起来，点击“打开设置”调用已保存的配置信息。
- 该软件可以直接从我公司网站下载，或向相关业务人员咨询。

六、技术参数

序号	项目内容	参数指标
1	电源电压	5-12V
2	静态电流	30mA
3	数据输入/输出电平	TTL
4	接口方式	TTL 电平串行接口、SPI 接口
5	串口波特率	1200bps-9600bps
6	电力线载波频率	65KHz—202KHz
7	调制解调技术	DSSS
8	载波波特率	100bps -600bps
9	数 位	8
10	起 始 位	1
11	停 止 位	1
12	每帧数据长度	32byte-256byte
13	通信距离	200m
14	存储温度	-40℃~+85℃
15	工作温度	-20℃~+70℃
16	相对湿度	5%~95%不凝结

七、要点说明

1、工作电源需在规定的要求范围内

BWP08 使用 5V-12VDC 供电，电压不能高于 12V，最低不能低于 4.5V，用户必须保证模块的供电电源在规定的范围内。

2、正确进行通讯端口的连接

BWP08 模块提供 TTL 电平的通讯口，串口波特率共有四种进行选择（1200bps、2400bps、4800bps、9600bps），用户在正常使用前必须正确地配置好模块的参数，如果模块的串口波特率与用户系统的串口波特率不一致，那么将导致模块无法接收用户数据，还无法将电力线上的数据发送给用户。

3、电力线上载波波特率必须保持一致

由于 BWP08 电力载波模块的载波波特率可以设置，所以用户在实际使用前必须根据自身系统所处实际应用环境选择好线上通讯波特率，不同波特率的模块无法相互通讯。按照正常情况，载波波特率越低，数据传输越可靠、传输距离也越远。请注意：载波波特率与载波频率是两个不同的概念，载波波特率是指数据传输的速率，载波频率是指 FSK 调制时所用的调制频率。BWP08 共有 6 种载波波特率供用户选择，用户可以从 100bsp-600bps 自由选择进行通讯，同时用户也可以选择载波频率，BWP08 共有 6 种载波频率可供用户选择。

4、载波数据传输类型必须一致

在应用中，BWP08 的载波波特率及载波频率一致的情况下，就可以相互通讯，即使串口波特率不一致也可以，但前提条件是：载波数据的传输类型必须一致。目前 BWP08 载波模块有两种传输类型，一种是定长传输，另一种是定帧传输，关于这两种传输类型的定义，请参看相关章节。如果定长类型的模块与定帧类型的模块相互混用，那么将导致定帧模块每次都会接收到最大帧长度的字节数，因为定长模块在发送时，不发送帧尾，定帧类型的模块由于收不到帧尾，所以无法判断用户数据何时结束，将一直处于接收状态，直到收满最大帧长度为止。如果定帧类型的模块向定长类型的模块发送数据，那么定长模块只接收预设的字节数后便停止接收，如果发送的字节数小于预设的字节长度，那么模块将接收线上噪音数据，只到收满预设长度为止。所以在实际应用中，用户可以灵活地把握模块的传输类型，在正常情况下，一个网络中，必须将所有模块的传输类型保持一致。

5、如果通讯距离达不到要求，应使用软中继功能

用户在使用中，实际的通讯距离达不到要求，可以采用两种方式，一种是缩短两个模块的安装

距离，另一个就是采用软中继功能，以此增加通讯距离。

6、保证相互通信的模块处于电力线的同一相中

在首次使用时，需要确认相互通讯的模块处于电力线的同一相中，由于 BWP08 电力载波模块不能跨相传输数据，所以如果相互通讯的模块如果不处于同一相中，那么有可能导致通讯失败，如果用户无法确认相互通讯的模块是否处于同一相中，那么可以主控模块处增加一个三相藕合器，那么就可以确保主控模块与三相下的任意一个模块进行通讯。

八、技术支持

有任何技术及应用问题，请及时联系本公司技术工程师

联系人：高先生 陈先生

电话：0755-25587679 0755-25587677

传真：0755-25587858

地址：深圳市罗湖区笋岗东路 2121 号华凯大厦 2202 室 邮编：518008

电邮：sales@bitwell.cn 网址：<http://www.bitwell.cn>



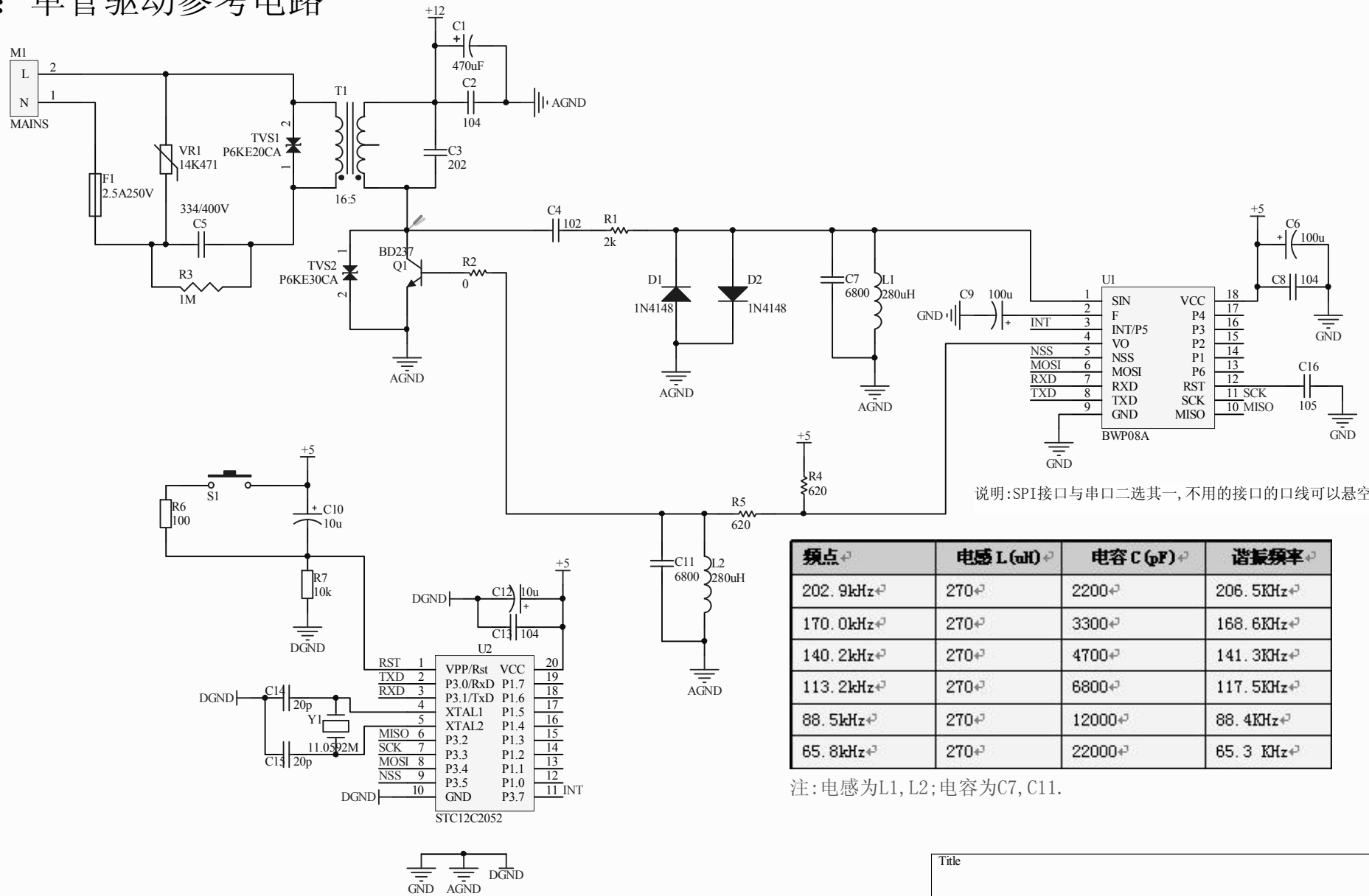
交流高压 注意安全

版权声明：本档版权属于深圳市必威尔科技有限公司所有，如有更改，恕不另行通知

版本号：1.0 文档类型：说明文档

发行日期：2007-12

附录一：单管驱动参考电路

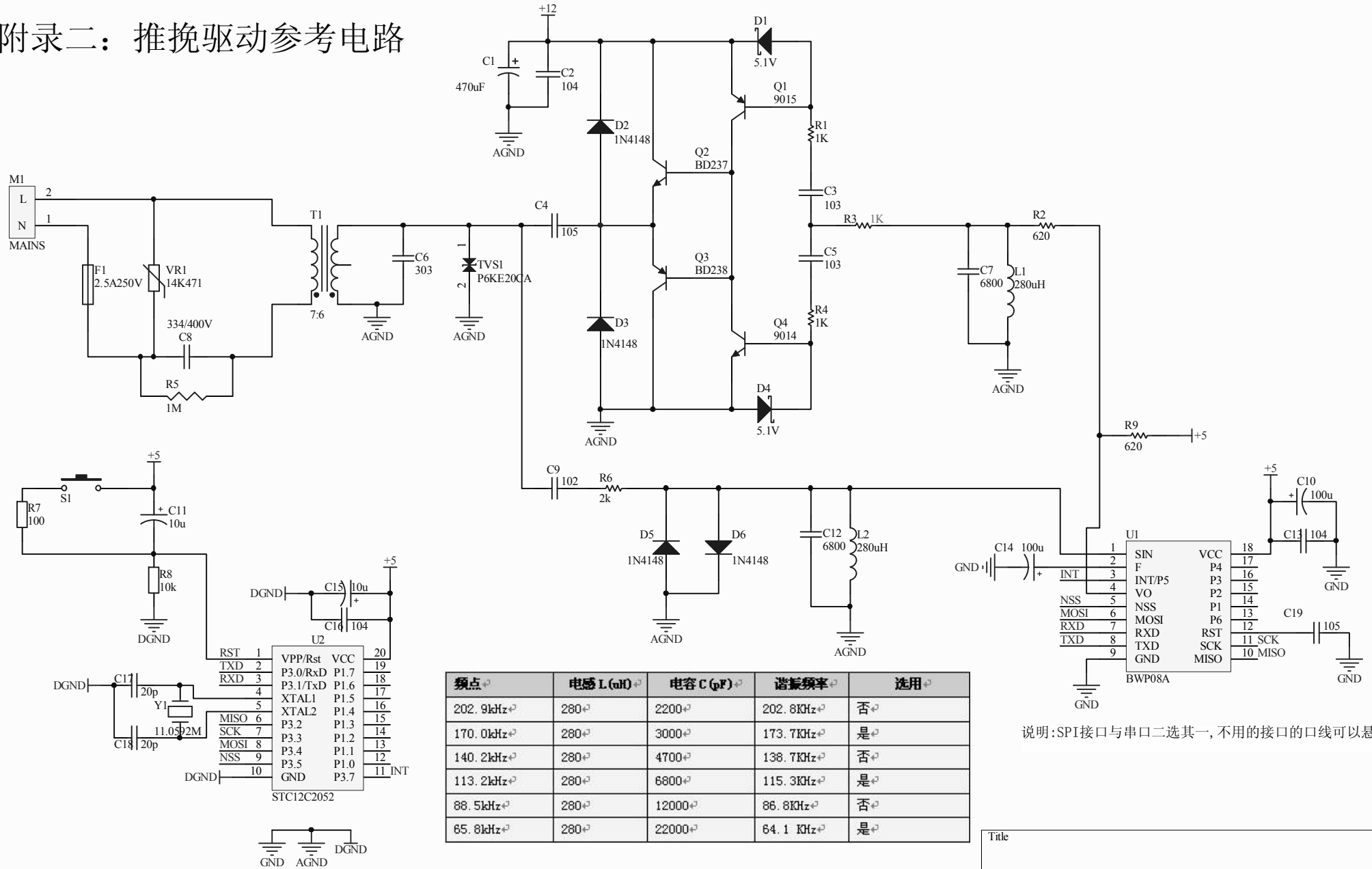


频点	电感 L (uH)	电容 C (pF)	谐振频率
202.9kHz	270	2200	206.5KHz
170.0kHz	270	3300	168.6KHz
140.2kHz	270	4700	141.3KHz
113.2kHz	270	6800	117.5KHz
88.5kHz	270	12000	88.4KHz
65.8kHz	270	22000	65.3 KHz

注：电感为L1, L2；电容为C7, C11.

Title		
Size	Number	Revision
A4		
Date:	2008-4-2	Sheet of
File:	J:\Wrok_Group\..BWP08单管驱动.SchDoc Drawn By:	

附录二：推挽驱动参考电路



频点 [□]	电感 L(mH) [□]	电容 C(μF) [□]	谐振频率 [□]	选用 [□]
202.9kHz [□]	280 [□]	2200 [□]	202.8KHz [□]	否 [□]
170.0kHz [□]	280 [□]	3000 [□]	173.7KHz [□]	是 [□]
140.2kHz [□]	280 [□]	4700 [□]	138.7KHz [□]	否 [□]
113.2kHz [□]	280 [□]	6800 [□]	115.3KHz [□]	是 [□]
88.5kHz [□]	280 [□]	12000 [□]	86.8KHz [□]	否 [□]
65.8kHz [□]	280 [□]	22000 [□]	64.1 KHz [□]	是 [□]

注：电感为L1, L2; 电容为C7, C12.

说明：SPI接口与串口二选一，不用的接口的口线可以悬空。

Title		
Size	Number	Revision
A4		
Date:	2008-8-14	Sheet of
File:	L:\Wrok_Group\..BWP08推挽驱动.SchDoc	
	Drawn By:	