

BCM90 蓝牙模块电压支持单电源 3.3V。是一款多功能的蓝牙 4.0 模块，根据内置软件的不同，可以应用于多种做数据传输的蓝牙产品，适合安卓 4.3 系统以上以及 iphone,ipad 等手机或平板电脑(蓝牙 4.0 版本)。

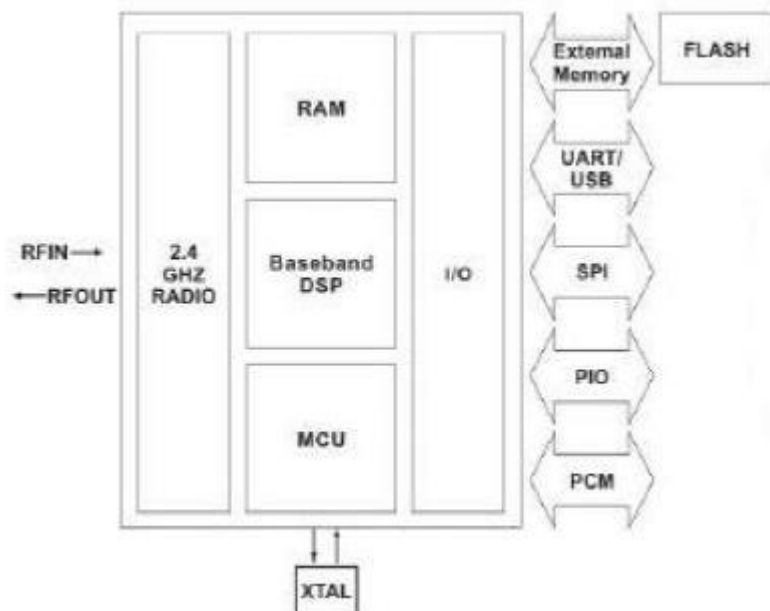
※技术参数

- 主芯片：CSR1010(BlueCore)
- 蓝牙版本：V4.0
- 输出功率：Class II
- Flash 存储容量：6Mbit
- 工作电压：3.3V
- 接口方式：UART
- 外形尺寸：27mm*13mm*2.2mm
- Rohs 标准：满足

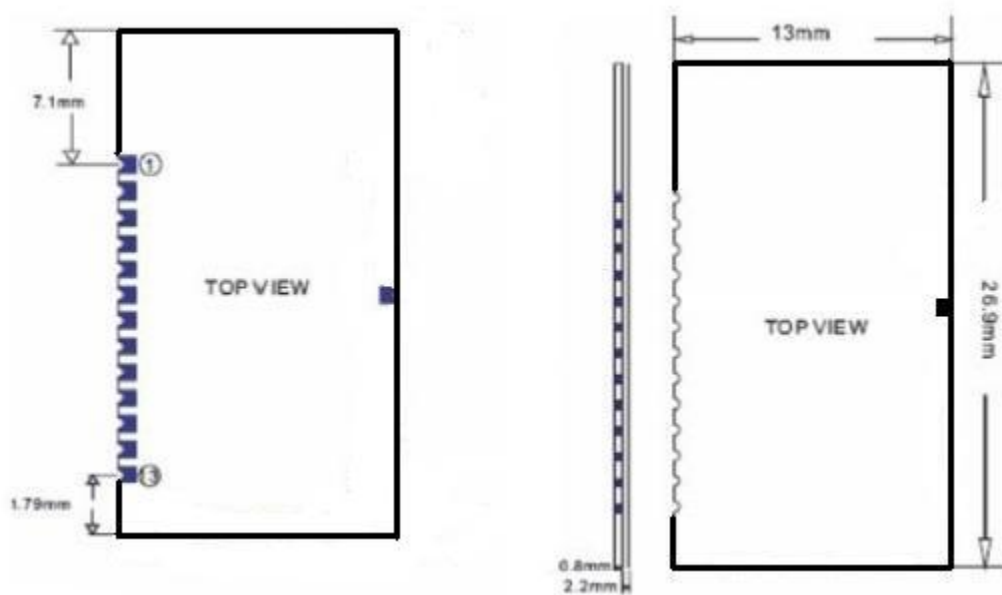
※应用范围

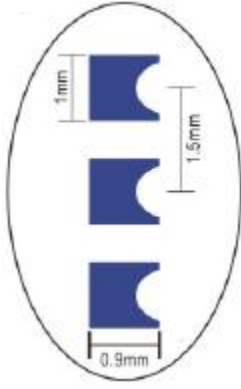
- 蓝牙 GPS
- 汽车诊断仪 OBDII
- 蓝牙打印机
- 蓝牙数码相机
- 蓝牙鼠标，蓝牙键盘等
- PDAS,PND,PCS
- 游戏手柄
- 蓝牙转串口产品

※原理方框图

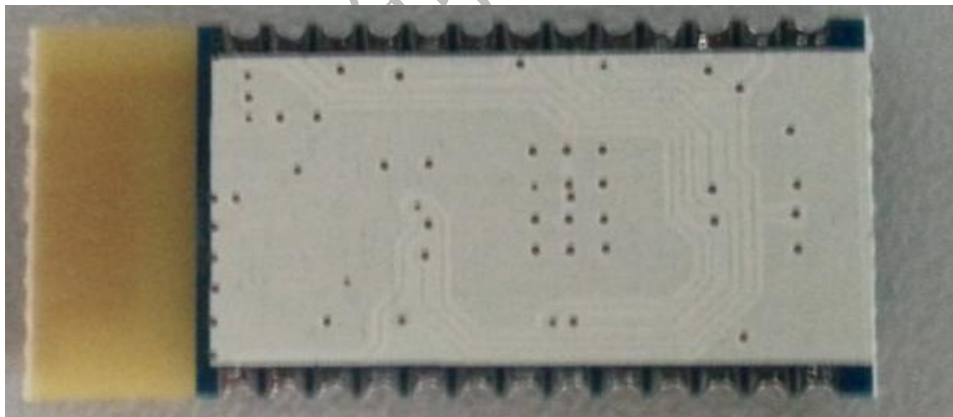
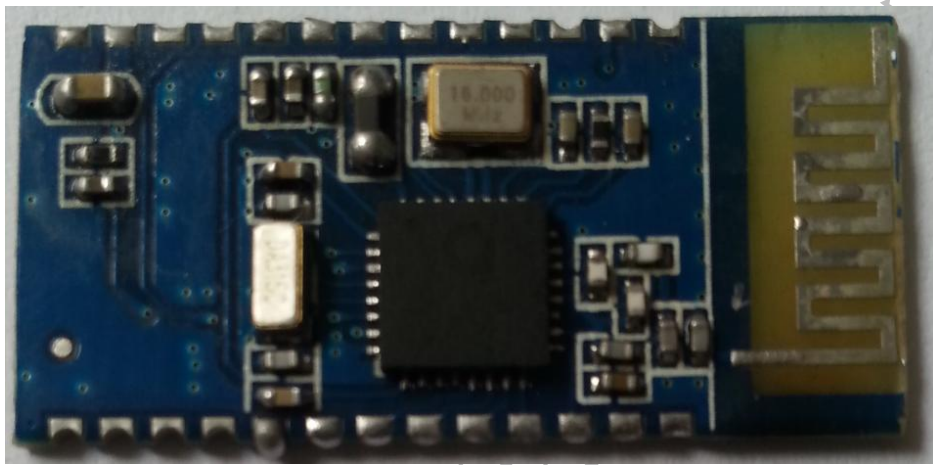


※外形尺寸图

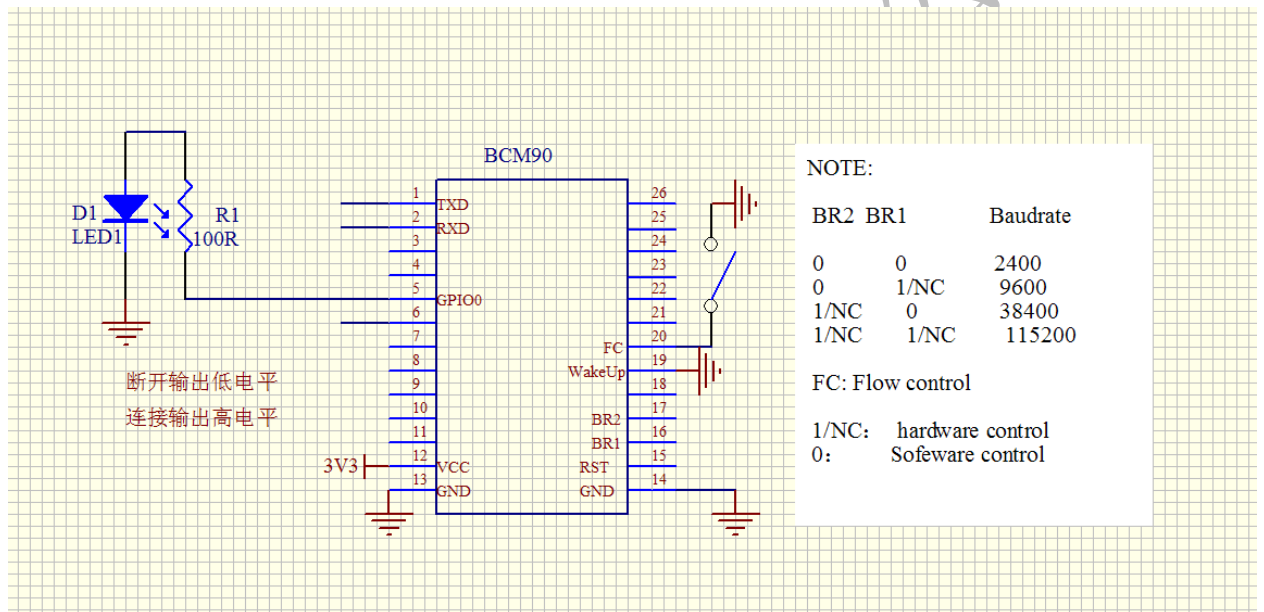




※外形图片



※参考设计原理图



※脚位定义

| | | | | | |
|----|----------------|---------------------------|----|------------------------|---|
| 1 | UART TXD | 数据发送端口 | 14 | GND | 电源地 |
| 2 | UART RXD | 数据接受脚 | 15 | RST | 模块复位，可悬空，模块内部有复位 |
| 3 | NULL | 悬空无连接 | 16 | BR2(内部上拉) | 波特率选择脚 BR2 BR1 波特率 |
| 4 | NULL | 悬空无连接 | 17 | BR1(内部上拉) | 0 0 2400 0 1 9600 1 0 38400 1 1 115200 |
| 5 | BT-STS | 蓝牙状态指示 IO 断开：0 连接：1 | 18 | TX_WAKE | 模块需要发送数，先拉低该 IO 唤醒 |
| 6 | NULL | 悬空无连接 | 19 | RX_WAKE | 模块需要接受数，先拉低该 IO 唤醒 |
| 7 | NULL | 悬空无连接 | 20 | Flow Control (内部上拉) | 数据流控制模式设置 1/NC: 硬件控制 0: 软件控制 |
| 8 | NULL | 悬空无连接 | 21 | GPIO | 通用输入输出端口 |
| 9 | ST_CON/广播模式控制脚 | 拉低进入广播模式 | 22 | GPIO | 通用输入输出端口 |
| 10 | GPIO | 通用输入输出端口 | 23 | GPIO | 通用输入输出端口 |
| 11 | GPIO | 通用输入输出端口 | 24 | GPIO | 通用输入输出端口 |
| 12 | 3.3V | 电源正 | 25 | GPIO | 通用输入输出端口 |
| 13 | GND | 电源地 | 26 | GPIO | 通用输入输出端口 |

※命令列表（硬件控制模式）

| 类型 | 命令 ID | 长度 | 参数 | 注释 |
|-------------------|-------|-----------|--|--|
| 命令(CMD) 0xEA | 0x05 | 0x01-0x10 | Params[]="BTCOM" | 修改蓝牙设备名字, 默认是 BTCOM |
| | 0x08 | 0x04 | Interval Min=param0+(param1<<8) Interval Max=param2+(param3<<8) | 修改连接模式的间隔时间 默认: Min=0x0018(0x0018*1.25ms) Min=0x0028(0x0028*1.25ms) |
| | 0x09 | 0x04 | Interval Min=param0+(param1<<8) Interval Max=param2+(param3<<8) | 修改广播模式的间隔时间 默认: Min=0x0030(0x0018*0.625ms) Min=0x0064(0x0028*0.625ms) |
| | 0x0b | 0x00 | No parameters | 读取蓝牙 MAC 地址 |
| | 0x0c | 0x01 | 0=<param0<=11 Power[param1]={-20,-18,-16,-14,-12,-10,-8,-6,-4,-2,0,2} | 修改射频发射功率 默认: param0=0x0a(0dbm) |
| | 0x0f | 0x01 | 0=<param0<=255(ms) | 修改射频发射唤醒时间 默认: 0ms |
| 事件(EVENT) 0xED | 0x05 | 0x01-0x10 | Params[]="BTCOM" | 设置成功: 返回新设备名字 设置失败: 返回之前设备名字 |
| | 0x08 | 0x04 | Interval Min=param0+(param1<<8) Interval Max=param2+(param3<<8) | 设置成功: 返回新的控制参数 设置失败: 返回之前的控制参数 |
| | 0x09 | 0x04 | Interval Min=param0+(param1<<8) Interval Max=param2+(param3<<8) | 设置成功: 返回新的控制参数 设置失败: 返回之前的控制参数 |

| | | | | |
|--|------|------|--|-----------------------------------|
| | | | | |
| | 0x0b | 0x06 | | 返回设备的 MAC 地址 |
| | 0x0c | 0x01 | 0=<param0<=11 Power[param1]={-20,-18,-16,-14,-12,-10,-8,-6,-4,-2,0,2} | 设置成功: 返回新的控制参数 设置失败: 返回之前的控制参数 |
| | 0x0f | 0x01 | 0=<param0<=255(ms) | 设置成功: 返回新的控制参数 设置失败: 返回之前的控制参数 |

※命令列表（软件控制模式）

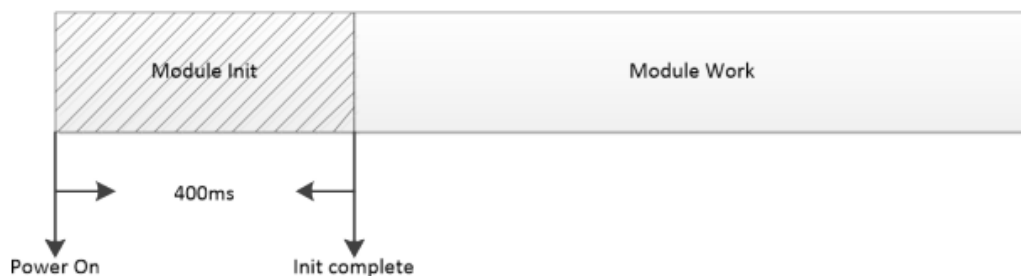
| 类型 | 命令 ID | 长度 | 参数 | 注释 |
|-------------------|-------|-----------|--|--|
| 命令(CMD) 0xEA | 0x01 | 0x01 | Params1=0 stop advertise Params1=1 start advertise | 启动和关闭广播模式 |
| | 0x04 | 0x01 | Params1=0 disconnect command | 停止蓝牙连接 |
| | 0x05 | 0x01-0x10 | Params[]="BTCOM" | 修改设备名字, 默认是 BTCOM |
| | 0x08 | 0x04 | Interval Min=param0+(param1<<8) Interval Max=param2+(param3<<8) | 修改连接模式的间隔时间 默认: Min=0x0018(0x0018*1.25ms) Min=0x0028(0x0028*1.25ms) |
| | 0x09 | 0x04 | Interval Min=param0+(param1<<8) Interval Max=param2+(param3<<8) | 修改广播模式的间隔时间 默认: Min=0x0030(0x0018*0.625ms) Min=0x0064(0x0028*0.625ms) |
| | 0x0b | 0x00 | No parameters | 读取蓝牙 MAC 地址 |
| | 0x0c | 0x01 | 0=<param0<=11 Power[param1]={-20,-18,-16,-14,-12,-10,-8,-6,-4,-2,0,2} | 修改射频发射功率 默认: param0=0x0a(0dbm) |
| | 0x0d | 0x00 | No parameters | 读取当前蓝牙状态 |
| | 0x0e | 0x00 | No parameters | 读取蓝牙 RSSI 信号强度 |
| 事件(EVENT) 0xED | 0x05 | 0x01-0x10 | Params[]="BTCOM" | 设置成功: 返回新设备名字 设置失败: 返回之前设备名字 |
| | 0x08 | 0x04 | Interval Min=param0+(param1<<8) Interval Max=param2+(param3<<8) | 设置成功: 返回新的控制参数 设置失败: 返回之前的控制参数 |

| | | | | |
|--|------|------|--|---|
| | 0x09 | 0x04 | Interval Min=param0+(param1<<8) Interval Max=param2+(param3<<8) | 设置成功: 返回新的控制参数 设置失败: 返回之前的控制参数 |
| | 0x0b | 0x06 | | 返回设备的 MAC 地址 |
| | 0x0c | 0x01 | 0=<param0<=11 Power[param1]={-20,-18,-16,-14,-12,-10,-8,-6,-4,-2,0,2} | 设置成功: 返回新的控制参数 设置失败: 返回之前的控制参数 |
| | 0x0d | 0x01 | param0=0x00 sleep param0=0x01 advertise param0=0x02 connecte full param0=0x03 advertise empty | 有三种情况, 状态更新送出 1,接收到启动和广播模式命令 2 接收到停止连接命令 3 模块状态有变化 |
| | 0x0e | 0x01 | | |
| | 0x0f | 0x01 | 0=<param0<=255(ms) | 设置成功: 返回新的控制参数 设置失败: 返回之前的控制参数 |

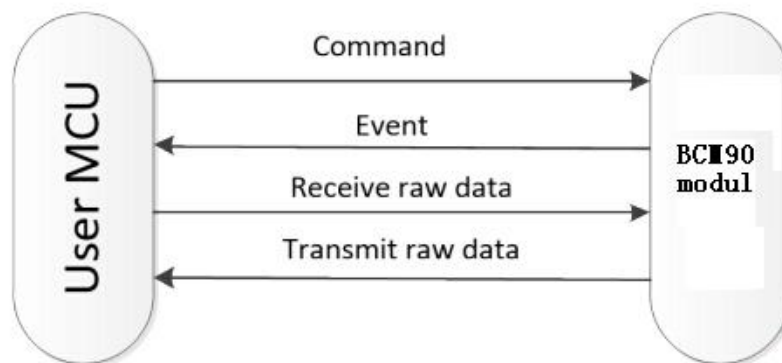
※传输数据

直接将数据发送给模块, 数据会透传给上位机蓝牙(手机, PC,iPAD)

一, 模块上电后, 启动蓝牙协议栈初始化程序时间是 400ms。



Communication



Speed Test

| Operating system | Baud Rate | Connect interval(ms) | SPEED | Direction |
|------------------|-----------|----------------------|---------|-----------|
| IOS | 115200 | 18.75 | 6.4KBps | To iPhone |
| IOS | 38400 | 18.75 | 3KBps | To iPhone |
| IOS | 9600 | 18.75 | 0.9KBps | To iPhone |
| IOS | 2400 | 18.75 | 0.2KBps | To iPhone |

二参考代码

```

Void main (void)
{
    Uint8_t module_state = 0;
    gpio_write_pin ( MODULE_VCC_EN,HIGH); //Power on the Module.
    Delay_ms(400); // wait 400ms for module power on init.
    uart_rx_enable(); //enable uart receive.( RX interrupt enable )
                    //before TX command to
                    //module, should wakeup it, and delay 5ms

    gpio_write_pin(RX_WAKEUP , LOW );
    Delay_ms(5);
    //(1)transmit command--- Set TX wakeup timer
    uart_tx( &CMD_ Set TX wakeup timer[0], sizeof(CMD_ Set TX wakeup timer));
    //(2)transmit command--- updata adv param
    uart_tx( &CMD_ updata_adv_param[0], sizeof(CMD_ updata_adv_param));
    //start advertising
    uart_tx( &CMD_ Advertising[0], sizeof(CMD_ Advertising));
    // after command send, stop wakeup
    gpio_write_pin(RX_WAKEUP , HIGH );
    while(1)
    {
        switch(ble_state)
        {
            case SLEEP:
            case ADVERTITSE:
            case CON_BUSY: break;
            Case CON_IDLE: //in this state, user can send data to module.
            { //note:
            //before send data,module_state's value should be set CON_BUSY manually
            module_state = CON_BUSY; // before send data, wake up module.
            gpio_write_pin(RX_WAKEUP , LOW );

```

```
Delay_ms(5); //send data to module
uart_tx( &user_data [0], sizeof(user_data));
//len_max = 120byte // stop wakeup
gpio_write_pin(RX_WAKEUP , HIGH );
}break;
}
}
```

```
Void uart_rx_process(void)
```

```
{ //receive event
If(event_type == 0xED)
{ Switch(event_id)
{ //receive the module state
Case (Module state):
{ module_state = module_st_value; }
break;
}
}
}
```

深圳蓝创科技有限公司

电话: 0755-26508810-888

传真: 0755-26508830

QQ: 411894339

技术支持: 15914038540 唐生