

简介

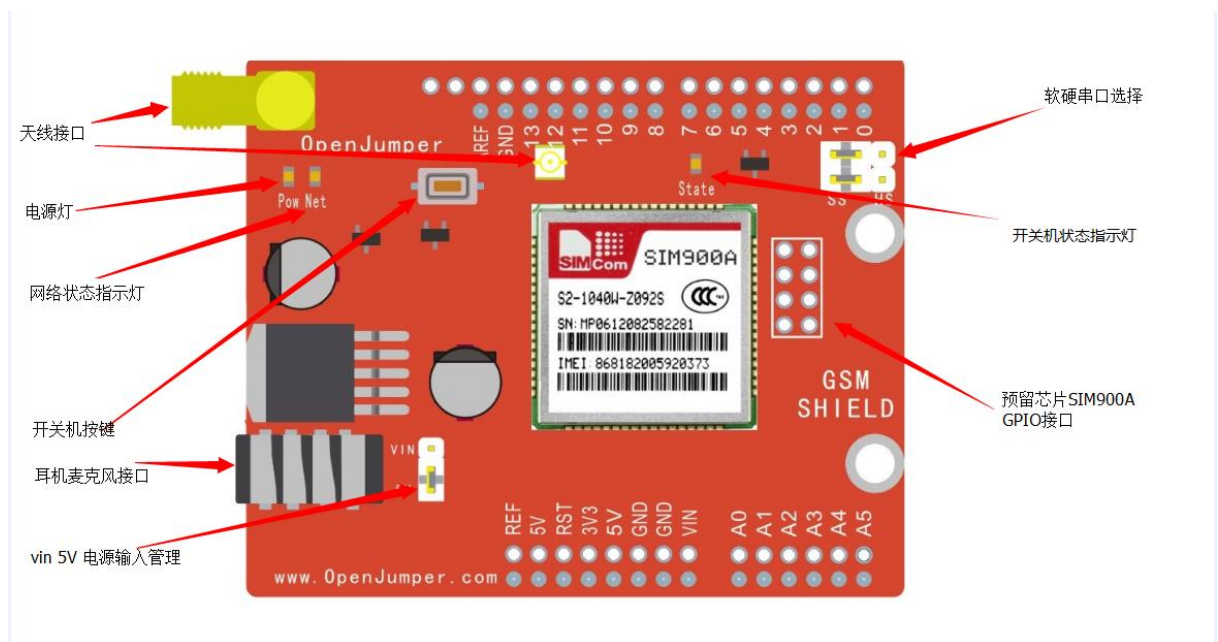
GSM 模块使用 SIMcom 公司的 SIM900 高精度无线 GSMGPRS 完全四频芯片，使用 SMT 封装且融合了高性能的 ARM926EJ-S 内核。可以适应小型设备的高性价比解决方案。模块采用标准工业级接口，SIM900 配备支持 GSM 和 GPRS 的 850MHZ、900MHZ、1800MHZ、1900MHz 四频的语音、短信、数据和传真，高内聚性且低功耗。

产品特点

- 兼容标准的 Arduino 以及 Arduino Mega
- 可自由选择硬串口和软串口
- 支持频段：850/900/1800/1900MHz
- 支持 TCP/UDP 协议
- 通过 AT 指令集控制：标准 – GSM 07.07 & 07.05 和 加强 – SIMCOM AT 指令集
- 二合一耳机接口
- 外部 SIM 卡槽
- 低功耗：1.5mA (Sleep mode)
- 标准化的外形设计

注意事项

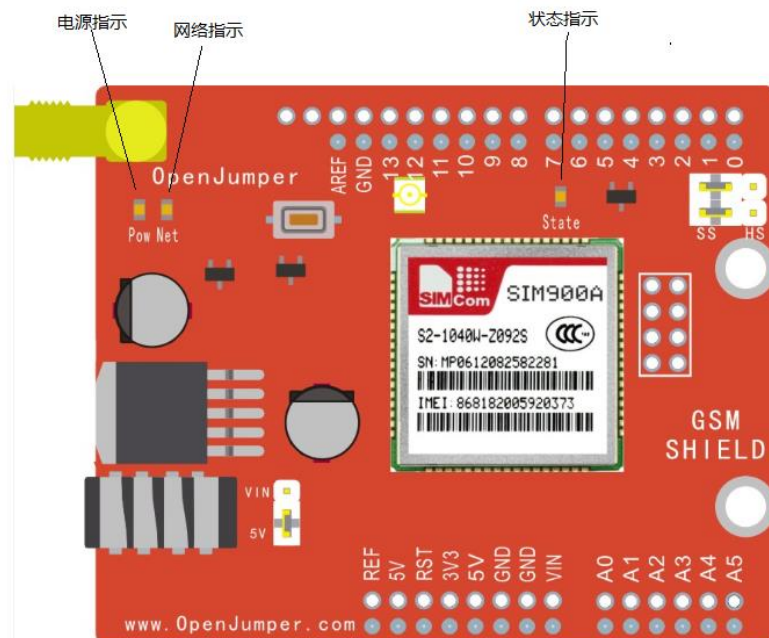
- 确保你的 SIM 卡解锁。
- GSM SHIELD 进行了 ESD 保护。可在干燥天气使用，但也请小心使用。
- GSM SHIELD 出厂设置的 UART 是 115200 bps 8-N-1(可以使用 AT 命令来改变)。



- 天线接口：接插外部天线
- 串口选择:选择软串口或硬串口作为与 Arduino 的接口
- 硬串口：Arduino/GSM shield 的 D0 和 D1(跳冒选择 gs 端)
- 软串口： Arduino/GSM SHIELD 的 D3 和 D4(跳冒选择 SS 端)
- 状态指示：指示 SIM900 是否启动
- 电源指示：指示 GSM SHIELD 电源状况
- 网络指示：指示 SIM900 连接网络的状态
- SIM900 的 GPIO, PWM 和 ADC：SIM900 的 GPIO,PWM 和 ADC 的管脚引出口
- 耳机麦克风接口：麦克风和耳机二合一接口
- 电源开关：开关 SIM900 A
- Vin、5V 电源管理：通过跳冒选择供电方式

Arduino 的接口用法

- D0: 被用于硬串口的数据接收 (RX)
- D1: 被用于硬串口的数据发送 (TX)
- D3: 被用于软串口的数据接收 (RX)
- D4: 被用于软串口的数据发送 (TX)
- D2: 被用于 SIM900A 的软电源开关
- VIN: 当你选择 6.5V~12V 的外部电源作为 Arduino 和 GSM SHIELD 的供电时被使用



LED 状态描述

LED	状态	功能
电源指示	灭	GSM Shield 的电源关闭
	亮	GSM Shield 的电源接通
状态指示	灭	SIM900A 电源关闭
	亮	SIM900A 电源接通
网络指示	灭	SIM900A 没有工作
	64ms 亮/800ms 灭	SIM900A 没有找到网络
	64ms 亮/3000ms 灭	SIM900A 找到网络
	64ms 亮/300ms 灭	GPRS 正在通信

用法

当你拿到 GSM Shield 的时候，你最想用它来做什么了？发一条短信（SMS）？或者打电话给朋友（需要耳机）？通过对 AT 命令的操作，你就能实现这些功能。AT 命令是一种简单文本的指令。你可以通过 GSM Shield 模块（SIM900A）的外部串口（UART）将 AT 命令发送到模块，从而实现对模块的控制。也就是说你可以使用任何的串口终端软件来对模块进行通信。

硬件安装

按照以下步骤来搭建你的硬件系统。

1. 在插入一张解锁的 SIM 卡

GSM Shield 使用的是一个 6 Pin 的 SIM 卡卡槽。SIM900 同时支持 1.8 伏 3.0 伏的 SIM 卡，它能自动识别 SIM 卡的电压类型。

2. 检查天线

确保天线正确接到天线接口。

3. 通讯接口配置

GSM Shield 能通过 Arduino 的硬戒软串口来控制。我们默认使用硬串口，

4. 插上 Arduino

将 GSM Shield 插到 Arduino Uno 上。

5. 给 Arduino 供电

通过 USB 口或者 DC 接口给 Arduino 供电。一旦通电后，电源指示灯就会亮起。

6. GSM Shield 开机

硬件开机：在通电后按住开关机按键，等待状态指示灯长亮即可使用。

软件开机：将 GSM 900a Shield 兼容板与 Uno 插上后利用代码控制 (D2) 口输出控制信号，等待状态指示灯长亮即可使用。 软启动控制时序图参考硬件设计手册中的启动时许图

Uno 实例测试程序

```
/*
*****

/* Openjumper

1.本段程序选择的是硬件串口通信，使用前请将软硬串口跳冒选择到硬串口端。

2.本段实例程序通过的是按键控制启动，请在插上 GSM Shield 兼容板子的时候先将板子使用按键启动。

具体参考 GSM Shield 开机中的硬件开机。

*/

*****

int val;          //定义变量 val

#define MAXCHAR 81 //recive buff max char
```

```

char aa[MAXCHAR]; //recive gsm model back signal

int j=0;          //recive index

int g_timeout=0; //handing time

////////////////////////////////////

char ATE0[]="ATE0";

char CREG_CMD[]="AT+CREG?";

char SMS_send[]="AT+CMGS=18";

char ATCN[]="AT+CNMI=2,1";

char CMGF0[]="AT+CMGF=0";

char CMGF1[]="AT+CMGF=1";

char CMGR[12]="AT+CMGR=1";

char CMGD[12]="AT+CMGD=1";

/*****/

#define SEND_MESSA_TO_YOUR "at+cmgs=\"18382117086\"\r\n" //发短信接收的手机号码

#define SEND_MESSA_CONTENT "http://shop69712475.taobao.com \n openjumper " //短信
内容

#define CALL_YOU_TELNUMBER "ATD18382117086;\r\n" //打电话接受的手机号码

/*****/

int readSerial(char result[])

{

    int i = 0;

```

```

while (Serial.available() > 0)

    {

        char inChar = Serial.read();

        if (inChar == '\n')

            {

                result[i] = '\0';

                Serial.flush();

                return 0;

            }

        if(inChar!='\r')

            {

                result[i] = inChar;

                i++;

            }

    }

}

void clearBuff(void)//打电话发短信清除指令

{

    for(j=0;j<MAXCHAR;j++)

        {

```

```
        aa[j]=0x00;

    }

    j=0;

}

int Hand(char *s)//发短信打电话发短信调用指令

{

    delay(200);

    clearBuff();

    delay(300);

    readSerial(aa);

    if(strstr(aa,s)!=NULL)    //检测单片机和模块的连接

    {

        g_timeout=0;

        clearBuff();

        return 1;

    }

    if(g_timeout>50)

    {

        g_timeout=0;

        return -1;

    }

}
```

```
        g_timeout++;

        return 0;

    }

void send_english(void) //发短信

{

    clearBuff();

    Serial.println(CMGF1);

    delay(500);

    while (Hand("OK")==0);

    clearBuff();

    Serial.println(SEND_MESSA_TO_YOUR);

    delay(500);

    while (Hand(">")==0);

    Serial.println(SEND_MESSA_CONTENT); //发短信内容

    delay(100);

    Serial.print("\x01A"); //发送结束符号

    delay(10);

    delay(1000);
```



```
        while (Hand("OK")==0);

    }

void send_call(void) //打电话

{

    clearBuff();

    Serial.println(CALL_YOU_TELNUMBER); //打电话

    delay(2000);

    while (Hand("OK")==0);

}

void setup (void)

{

    Serial.begin(115200); //设置波特率为 115200

    // Serial.println("serial port [ok]!");

}

void loop (void)

{

    send_english(); //发短信

    delay(1000);

    send_call(); //打电话

    while(1);

}
```

```
}
```

注意：sim900 的模块如果设置过波特率 需要设置为自适应波特率才能进行上面的程序操作。