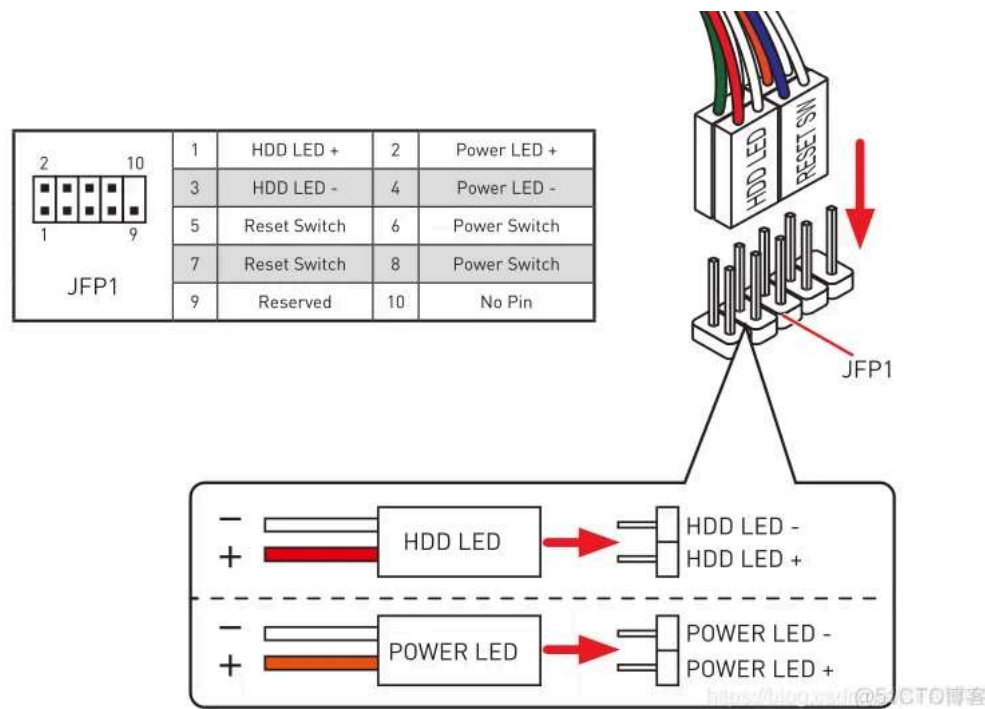


本项目的目标是要实现低成本的远程开机，因此wifi模块是必须的。常见的wifi模块包括esp8266、esp32等，这两种模块有不同的封装形式在不同的应用场景下可以根据需求选用。考虑到电脑开机需要仅需要控制2个io（一个检测按钮输入，一个IO连接A点输出），esp8266的esp-01s型模块引出了IO0和IO2两个IO，当不使用串口通信因此tx、rx也可用作I。esp-01s型模块需要3.3v供电，需要外电压转换模块。esp01-s模块价格低廉，非常适合本项目的使用。如果有其他需求还可以使用具有更多引脚的nodemcu模块，该模块为5v供电可以直接使用主板未使用的USB接口取电。本项目以esp01-s为例进行介绍，因为实验室有现成的nodemcu模块，所以实验以nodemcu进行。

1.3电路连接

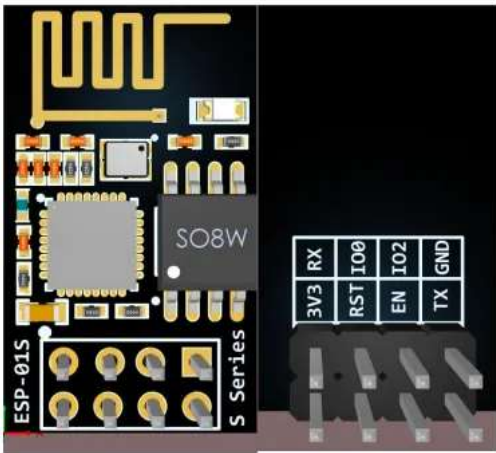
电脑机箱面板一般会引出8根杜邦接头的连接线，以微星主板为例，如下表和图所示，分别为硬盘工作指示灯，电源指示灯，开机按键和重启按键。在电脑处于关机状态下测量开关按键和重启按键引脚电压，发现第5和第6引脚电平为3.3v，第7和第8脚为地。

1	HDD LED+	2	Power LED+
3	HDD LED-	4	Power LED-
5	Reset Switch	6	Power Switch
7	Reset Switch	8	Power Switch
9	Reserved	10	No Pin



结合开机原理图可以看到，第6脚对应就是A点，开机按钮按下等价于将6脚电平拉低然后拉高。

esp01-s模块的引脚如下图所示，引出了IO0与IO2，查询手册可以知道模块的LED灯由IO2进行控制，因此首选IO2作为输出的控制引脚。将2脚与主板上JFP1的第6脚连接IO2，第8脚连接GND，开关按键通过杜邦线连接到IO0和GND。因为主板上未连接的USB接口，因此可以将开关一端接在USB接口的GND另一端接在esp01-s的IO0。



https://blog.csdn@51CTO博客

esp01-s的电气特性如图所示，供电电压为3.3v需要特别注意。使用烧写器可以使用充电宝等具有USB输出接口的设备进行独立供电也可以购买USB延长线进行改造，使用未使用的USB接口供电。实验中主板USB2接口同时为无线网卡和nodemcu供电。

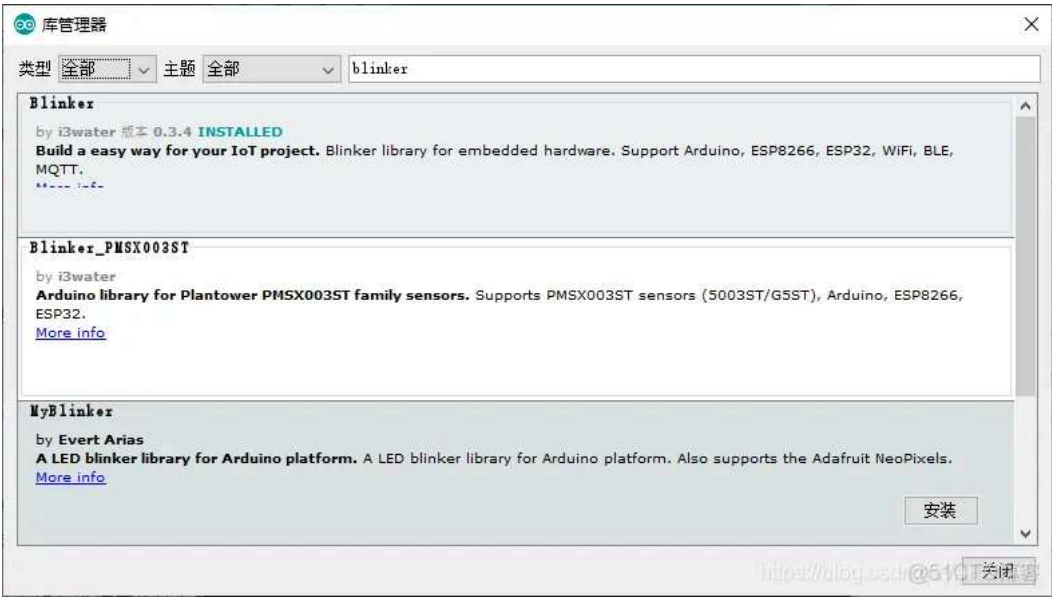
电气特性

参数		条件	最小值	典型值	最大值	单位
供电电压		VDD	3.0	3.3	3.6	V
I/O	$V_{IL}/V_{IH}$	-	-0.3/0.75VIO	-	0.25VIO/3.6	V
	$V_{OL}/V_{OH}$	-	N/0.8VIO	-	0.1VIO/N	V
	$I_{MAX}$	-	-	-	12	mA

https://blog.csdn@51CTO博客

1.4 物联网平台选择

要实现远程开机需要将设备接入物联网，常见的免费物联网平台包括 阿里云、贝壳物联、点灯Blink等等。直接接入云平台需要使用云平台对应的SDK，编程过程复杂。贝壳物联平台和电灯都可以使用arduino进行编程，示例程序只需要修改少量代码即可。相比于贝壳物联，电灯使用手机app进行交互页面设计控件类型丰富界面也比较好看。对比之后决定选择接入电灯Blink平台。  
在手机应用商店下载电灯blink的app，添加独立设备选择网络接入阿里云，添加入门示例。在arduino中安装esp8266和blinker软件库，并打开Blinker\_hello/Hello\_Wifi.ino示例，如图所示。



https://blog.csdn@51CTO博客

修改授权码auth为设备的授权码，ssid为要接入的wifi名，pswd为对应的密码。修改按钮的回调函数并增加按钮按下时的开机实现，然后上传到esp01-s。修改后的代码如下：

