

Pico Link II 使用说明

关于

- 作者: 卜林 / QQ: 1626632460
- QQ交流群: 613773009
- 文档修订时间: 2024.1.2
- 适用固件版本: v2.31
- 适用硬件版本: v1.1

模块参数

Pico Link II 模块是一款高速 UART/SPI 2.4G WiFi 透传模块，该模块支持多样化的配置，具有两种输入方式（UART、SPI）和两种输出方式（UDP、TCP），可自由组合成四种工作模式

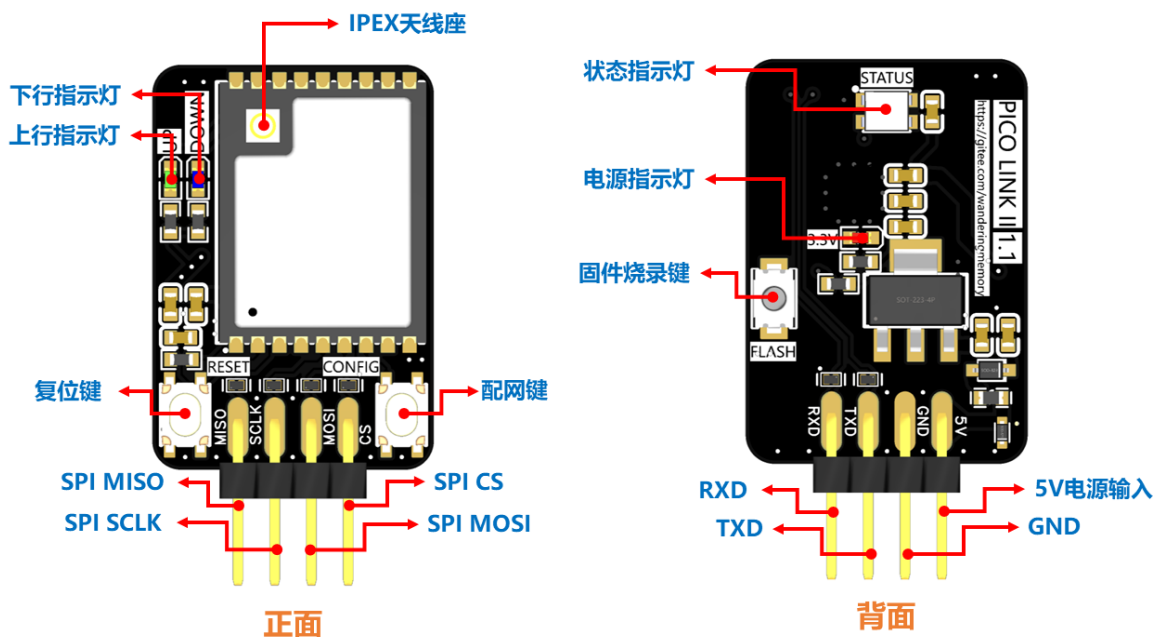
模块参数如下

芯片	esp32-c3
最大 UART 速率	5Mbps
UART 单次接收最大字节数	无限制
最大 SPI 速率	60Mbps (1bit全/半双工) 120Mbps (2bit半双工)
SPI 单次接收最大字节数	16384
支持的 WiFi 模式	STA AP
空中速率	30Mbps (UDP) 20Mbps (TCP)
双向通信支持	支持，可关闭以提高单向发送性能
双向通信模块最大发送字节数	4096
静态 IP 支持	WiFi AP 模式下支持
配置方式	通过按键进入配置模式，WEB 配置
支持的 WiFi 频率	在2.4 GHz 频带支持20 MHz 和40 MHz 频宽
支持 WiFi 协议	IEEE 802.11 b/g/n
天线连接方式	IPEX 1代天线座
与上位机通信方式	UDP TCP, client server
指示灯	电源指示、状态指示、上行指示、下行指示
供电电压	5V DC
工作电流	120mA 左右
硬件保护	过流保护、反接保护、静电保护
尺寸	1.95cm x 2.8cm

注：

1. 模块 SPI 工作在从机模式，双向通信时需要主机端主动发起传输，一次传输中通过 MISO 引脚向主机端最多传输4096字节
2. 模块 UART 双向通信时一次传输中通过 TXD 引脚向主机端最多传输4096字节
3. 对于全双工 1bit SPI，**开启双向通信后**模块单次最大接收字节数降低到**4096**字节（和模块最大发送字节数一致），对于半双工 1/2bit SPI 则不影响
4. 模块 UDP server 双向通信时，模块向最近一次给自己发送数据的上位机网络地址发送数据
5. 单次接收数据长度如果超过4096字节按照最大4096字节为一帧拆分多个发送

模块硬件组成如下



和乐鑫 AT 固件的区别

1. 本模块通信协议为纯透传，对于主机端完全透明，不需要像 AT 固件添加额外的协议帧头
2. 乐鑫 AT 固件需要主机端主动配置。本模块采用 web 配置，配置后的参数掉电不丢失，免去了主机端反复修改代码去配置模块的步骤
3. 本模块 WiFi STA 模式掉线自动重连，TCP client 模式断开自动重连。乐鑫 AT 固件需要主动去查询连接状态然后进行重连
4. 乐鑫 SPI-AT 固件只支持半双工 1bit SPI 通信（协议为乐鑫制定），本模块支持全双工 1bit SPI 透传并且支持 SPI 时钟模式配置
5. 乐鑫 AT 固件使用时只支持1种通信协议，本模块支持多种通信协议

连线

- 5V 供电，地线必接
- UART 通信

模块	下位机
RXD	TXD
TXD	RXD
MOSI	悬空
MISO	悬空
SCLK	悬空
CS	悬空

- SPI 全双工1bit 通信

模块	下位机
RXD	悬空
TXD	悬空
MOSI	MOSI
MISO	MISO
SCLK	SCLK
CS	CS

- SPI 半双工1bit 通信

模块	下位机
RXD	悬空
TXD	悬空
MOSI	MOSI
MISO	MISO
SCLK	SCLK
CS	CS

- SPI 半双工2bit 通信

模块	下位机
RXD	悬空
TXD	悬空
MOSI	D0
MISO	D1
SCLK	SCLK
CS	CS

配置

1. 上电，先按住模块 **CONFIG** 键不放，再按一下模块 **RESET** 键，等待状态指示灯变为**绿色**，成功进入配置模式后，模块 **UP** 指示灯和 **DOWN** 指示灯会**交替闪烁**
2. 手机或电脑打开 WiFi 连接 **Pico Link II Config**，密码：**88888888**
3. 连接成功后，打开浏览器在地址栏输入 **192.168.4.1** 访问即可进入配网界面，配置好参数后按复位键复位，模块会把新配置的参数写进 flash，掉电不丢失

配置界面如下

智能车竞赛的小伙伴们, 垆牛电子祝您比赛成功, 发车 🚗 顺利 🚀

⚡ 请先点击[查询配置信息]按钮, 查询成功后才会显示[提交配置信息]按钮 ⚡
选项右边出现蓝色勾✓代表该参数格式填写正确

模块名称: Pico Link II / 固件版本: v2.31

作者: 卜林 / QQ: 1626632460

QQ交流群: 613773009

输入方式:

UART波特率:

UART数据位:

UART校验位:

UART停止位:

输出方式:

WiFi连接方式:

STA连接的WiFi名称:

STA连接的WiFi密码:

双向通信:

工作方式:

远程主机 IPv4地址:

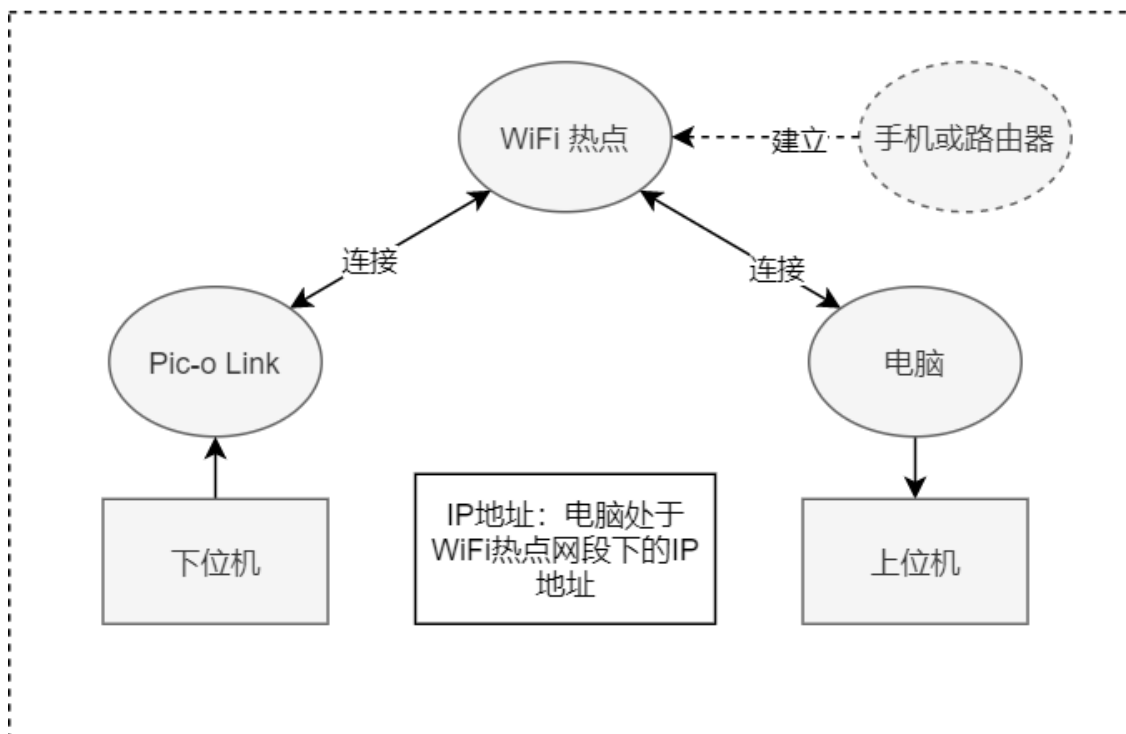
远程主机端口:

查询配置信息

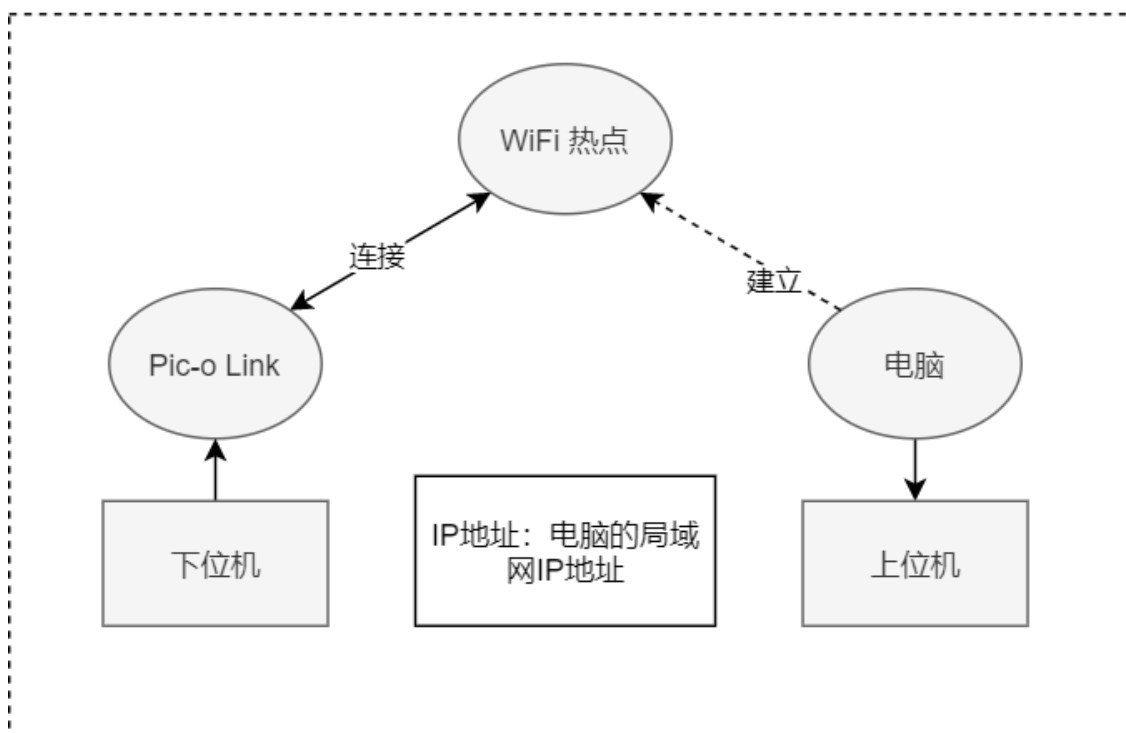
IP 地址配置

- 对于 WiFi STA + client 模式，远程主机 IPv4地址配置见如下示意图

情形一



情形二



状态指示灯

- 红色**: WiFi STA 模式下未连接 WiFi
- 蓝色**: WiFi 正常工作, UDP/ TCP 连接成功
- 紫色**: WiFi 正常工作, UDP / TCP 未连接
- 青色**: WiFi AP 模式下静态 IP 设置失败
- 绿色**: 配置模式

注:

1. UDP server 模式下状态灯指示灯初始为紫色，上位机需要向模块发送一小段数据以和模块建立一次握手从而让模块知道上位机的网络地址，之后状态灯指示灯变为蓝色

驱动

驱动文件见 `driver` 文件夹 `pico_link_ii.c` 和 `pico_link_ii.h`，使用前需要替换宏定义，找到正确的函数来定义，可参考 `driver/example` 文件夹下例程

```
1  //-----用户替换函数需要包含的头文件-----
2
3  //-----UART 宏定义-----
4  // UART发送多个字节，需要自己找函数定义，array:发送数据首地址，length:发送长度
5  #define UART_SEND_BYTES(array, length)
6
7  //-----SPI 宏定义-----
8  // SPI CS引脚拉低，需要自己找函数定义
9  #define SPI_CS_LOW
10 // SPI CS引脚拉高，需要自己找函数定义
11 #define SPI_CS_HIGH
12 // 全双工SPI发送多个字节，需要自己找函数定义，array:发送数据首地址，length:传输长度
13 #define SPI_SEND_BYTES(array, length)
14 // 全双工SPI读取多个字节，需要自己找函数定义，buffer:接收数据首地址，length:传输长度
15 #define SPI_READ_BYTES(buffer, length)
```

UART

通过 UART 透传向上位机发送字符串 "hello world!" 示例

```
1 char *str = "hello world!";
2 pico_send_8bit_array_uart((unsigned char *)str, strlen(str));
```

发送成功后模块 UP 指示灯会亮

通过 UART 透传向上位机发送多字节数组示例

```
1 unsigned char array1[5] = {0x01, 0x02, 0x03, 0x04, 0x05};
2 unsigned char array2[5] = {0x06, 0x07, 0x08, 0x09, 0x0a};
3 unsigned char array3[5] = {0x0b, 0x0c, 0x0d, 0x0e, 0x0f};
4 pico_multi_src_t multi_src;
5 pico_multi_src_clear(&multi_src);
6 pico_multi_src_add(&multi_src, array1, 5);
7 pico_multi_src_add(&multi_src, array2, 5);
8 pico_multi_src_add(&multi_src, array3, 5);
9 pico_send_8bit_array_multi_uart(multi_src);
```

发送成功后模块 UP 指示灯会亮

SPI

使用驱动文件时请以全双工1bit模式初始化 SPI

全双工 1bit spi

通过全双工 1bit SPI 透传向上位机发送字符串 "hello world!" 示例

```
1 char *str = "hello world!";
2 pico_send_8bit_array_fd_spi((unsigned char *)str, strlen(str));
```

发送成功后模块 UP 指示灯会亮

通过全双工 1bit SPI 透传向上位机发送多字节数组示例

```
1 unsigned char array1[5] = {0x01, 0x02, 0x03, 0x04, 0x05};
2 unsigned char array2[5] = {0x06, 0x07, 0x08, 0x09, 0x0a};
3 unsigned char array3[5] = {0x0b, 0x0c, 0x0d, 0x0e, 0x0f};
4 pico_multi_src_t multi_src;
5 pico_multi_src_clear(&multi_src);
6 pico_multi_src_add(&multi_src, array1, 5);
7 pico_multi_src_add(&multi_src, array2, 5);
8 pico_multi_src_add(&multi_src, array3, 5);
9 pico_send_8bit_array_multi_fd_spi(multi_src);
```

发送成功后模块 UP 指示灯会亮

半双工 1bit spi

乐鑫官方提供的通信协议：https://docs.espressif.com/projects/esp-idf/zh_CN/latest/esp32c3/api-reference/protocols/esp_spi_slave_protocol.html

通过半双工 1bit SPI 透传向上位机发送字符串 "hello world!" 示例

```
1 char *str = "hello world!";
2 pico_send_8bit_array_hd_spi((unsigned char *)str, strlen(str));
```

发送成功后模块 UP 指示灯会亮

通过半双工 1bit SPI 透传向上位机发送多字节数组示例

```
1 unsigned char array1[5] = {0x01, 0x02, 0x03, 0x04, 0x05};
2 unsigned char array2[5] = {0x06, 0x07, 0x08, 0x09, 0x0a};
3 unsigned char array3[5] = {0x0b, 0x0c, 0x0d, 0x0e, 0x0f};
4 pico_multi_src_t multi_src;
5 pico_multi_src_clear(&multi_src);
6 pico_multi_src_add(&multi_src, array1, 5);
7 pico_multi_src_add(&multi_src, array2, 5);
8 pico_multi_src_add(&multi_src, array3, 5);
9 pico_send_8bit_array_multi_hd_spi(multi_src);
```

发送成功后模块 UP 指示灯会亮

通过半双工 1bit SPI 读取上位机发送的数据示例 (需要开启双向通信)

```
1 unsigned char buffer[128]; // 存放接收数据的缓存
2 unsigned long recv_len = 0;
3 recv_len = pico_read_8bit_array_hd_spi(buffer, 128); // 如果成功收到数据,
recv_len为正数, 否则为0
```

半双工 2bit spi

乐鑫官方提供的通信协议: https://docs.espressif.com/projects/esp-idf/zh_CN/latest/esp32c3/api-reference/protocols/esp_spi_slave_protocol.html

施工中.....

致用上位机图传

使用致用上位机前确保模块的输出方式配置成 `UDP`, 工作方式配置成 `client`, 远程端口配置成 `2333`

致用上位机详细使用见 [致用图传文档.pdf](#)

驱动文件见 `driver` 文件夹 `pico_link_ii.c` 和 `pico_link_ii.h`

通过修改 `pico_link_ii.h` 宏定义值切换发送方式, 1代表使能

```
1 // UART
2 #define PICO_PROTOCOL_UART (0)
3 // 全双工SPI 1bit
4 #define PICO_PROTOCOL_FD_SPI (0)
5 // 半双工SPI 1bit
6 #define PICO_PROTOCOL_HD_SPI (0)
```

向致用上位机发送 60 x 80 灰度图像示例

```
1 unsigned char image[60][80];
2 ipc_send_image(image, 60, 80);
```

发送成功后模块 `UP` 上行灯会亮

GY上位机图传

驱动文件见 `driver` 文件夹 `pico_gy.c` 和 `pico_gy.h`, 使用前需要包含 `pico_link_ii.c` 和 `pico_link_ii.h`

通过修改 `pico_link_ii.h` 宏定义值切换发送方式, 1代表使能

```
1 // UART
2 #define PICO_PROTOCOL_UART (0)
3 // 全双工SPI 1bit
4 #define PICO_PROTOCOL_FD_SPI (0)
5 // 半双工SPI 1bit
6 #define PICO_PROTOCOL_HD_SPI (0)
```


向 GY 上位机发送 60 x 80 灰度图像示例

```
1 unsigned char image[60][80];
2 sending(image, 80, 60);
```

发送成功后模块 UP 上行灯会亮

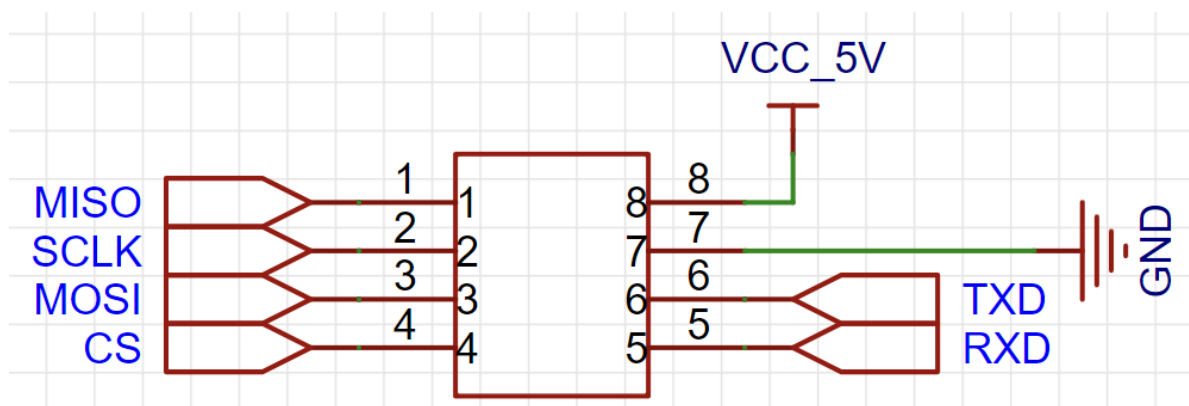
逐飞助手

适配见 [Pico Link II 适配逐飞助手.pdf](#)，在逐飞框架下只需改动两个函数即可使用

硬件接口

- 接口：2.54mm 2x4p 排母
- 5V 电源至少要保证 500mA 的输出电流

模块接口原理图如下



图中 TXD、RXD 为模块的 TXD、RXD

固件版本说明

V1.0 (硬件版本V1.0)

UART一次最多接收25000字节，SPI一次最多接收25000字节

V1.1

1. 增加 TCP 通信模式，可在配网界面设置
2. TCP 通信支持双向，通过 UART 转发上位机发送的数据，一次最多转发128字节
3. 配网界面设置的 ip 格式正确与否使用算法检验
4. 优化 UDP 发送
5. SPI 最大接收字节数提升到30000字节

⚡ V1.1 及之前软硬件已开源：https://gitee.com/wanderingmemory/pico_link_ii ⚡

V2.0 (硬件版本V1.1)

1. 采用 esp-idf5 框架
2. 支持 SPI 模式配置
3. 支持 WiFi AP 模式

V2.1

1. 支持 UART 更多参数配置
2. 修复部分浏览器刷新配置界面异常的问题
3. 用 vue 重构了配网界面
4. 删除 UART 硬件流控

V2.2

1. WiFi STA 模式支持 WiFi 掉线自动重连
2. WiFi AP 模式支持静态 IP 设置
3. 支持 server 模式
4. 支持 UDP 双向
5. 支持 SPI 双向
6. 配置界面增加恢复出厂设置按键
7. 优化了配置界面操作逻辑

V2.3

1. 修改全双工1bit SPI 接收，不需要额外协议，纯透传
2. 支持半双工1bit SPI 通信，理论上兼容逐飞 SPI-WiFi 模块驱动（其实就是乐鑫 SPI-AT 固件）
3. 支持半双工2bit SPI 通信
4. 模块双向通信一次最多发送字节数提升到4096字节
5. WiFi AP 模式支持最大连接设备数和信道配置

V2.31

1. 修复 WiFi AP 无法设置12, 13信道的 bug
2. 降低状态灯亮度
3. 优化性能