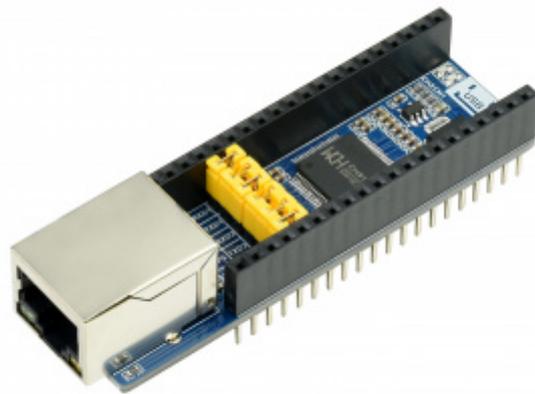


Pico-ETH-CH9121

来自Waveshare Wiki
跳转至: [导航](#)、[搜索](#)



功能简介

74.54mm x 21mm

UART转ETH模块

UART

Ethernet

说明

产品概述

提供Pico C语言例程跟Python例程

产品参数

参数名称	参数
供电电压	5V
工作电流	140mA
工作模式	TCP/UDP
串口波特率	300bps ~ 921600bps
串口 TTL 电平	兼容 3.3V 和 5V
工作温度	-40°C~85°C

产品尺寸

74.54 x21.00(mm)

接口说明

GP0	1	40	VBUS	VSYS	供电电源
GP1	2	39	VSYS	GND	电源地
GND	3	38	GND	GP0	接模块 UART1 的接收引脚 (UART1默认开启)
GP2	4	37	3V3_EN	GP1	接模块 UART1 的发送引脚 (UART1默认开启)
GP3	5	36	3V3(OUT)	GP4	接模块 UART2 的接收引脚 (UART2默认关闭)
GP4	6	35	ADC_VREF	GP5	接模块 UART2 的发送引脚 (UART2默认关闭)
GP5	7	34	GP28	GP14	接模块的串口配置引脚, 低电平进入配置, 高电平退出配置
GND	8	33	GND	GP17	接模块的复位引脚, 外部复位输入, 低电平有效
GP6	9	32	GP27		
GP7	10	31	GP26		
GP8	11	30	RUN		
GP9	12	29	GP22		
GND	13	28	GND		
GP10	14	27	GP21		
GP11	15	26	GP20		
GP12	16	25	GP19		
GP13	17	24	GP18		
GND	18	23	GND		
GP14	19	22	GP17		
GP15	20	21	GP16		

功能说明

CH9121 为网络串口透传芯片, 可实现串口数据与网络数据的双向透明传输, 支持 TCP CLIENT/SERVER,UDP CLIENT/SERVER 4 种工作模式, 串口波特率支持范围为 300bps ~ 921600bps, 使用前需通过上位机软件 NetModuleConfig.exe 或者串口命令配置芯片的网络参数和串口参数, 配置完成后, CH9121 将配置参数保存至内部存储空间, 芯片复位后, CH9121 将按保存的配置值工作。

默认配置

CH9121 出厂时, 端口 2 默认关闭, 端口 1 默认工作在 TCP CLIENT 模式, 网络相关默认参数依次为:

- (1) 设备 IP : 192.168.1.200
- (2) 子网掩码 : 255.255.255.0
- (3) 默认网关 : 192.168.1.1
- (4) 模块端口 : 2000
- (5) 目标 IP : 192.168.1.100

(6) 目标端口：1000

串口相关默认参数依次为：

(1) 波特率：9600

(2) 超时：0

(3) 数据位：8；停止位：1；校验：无

(4) 清空串口缓冲区：从不清空

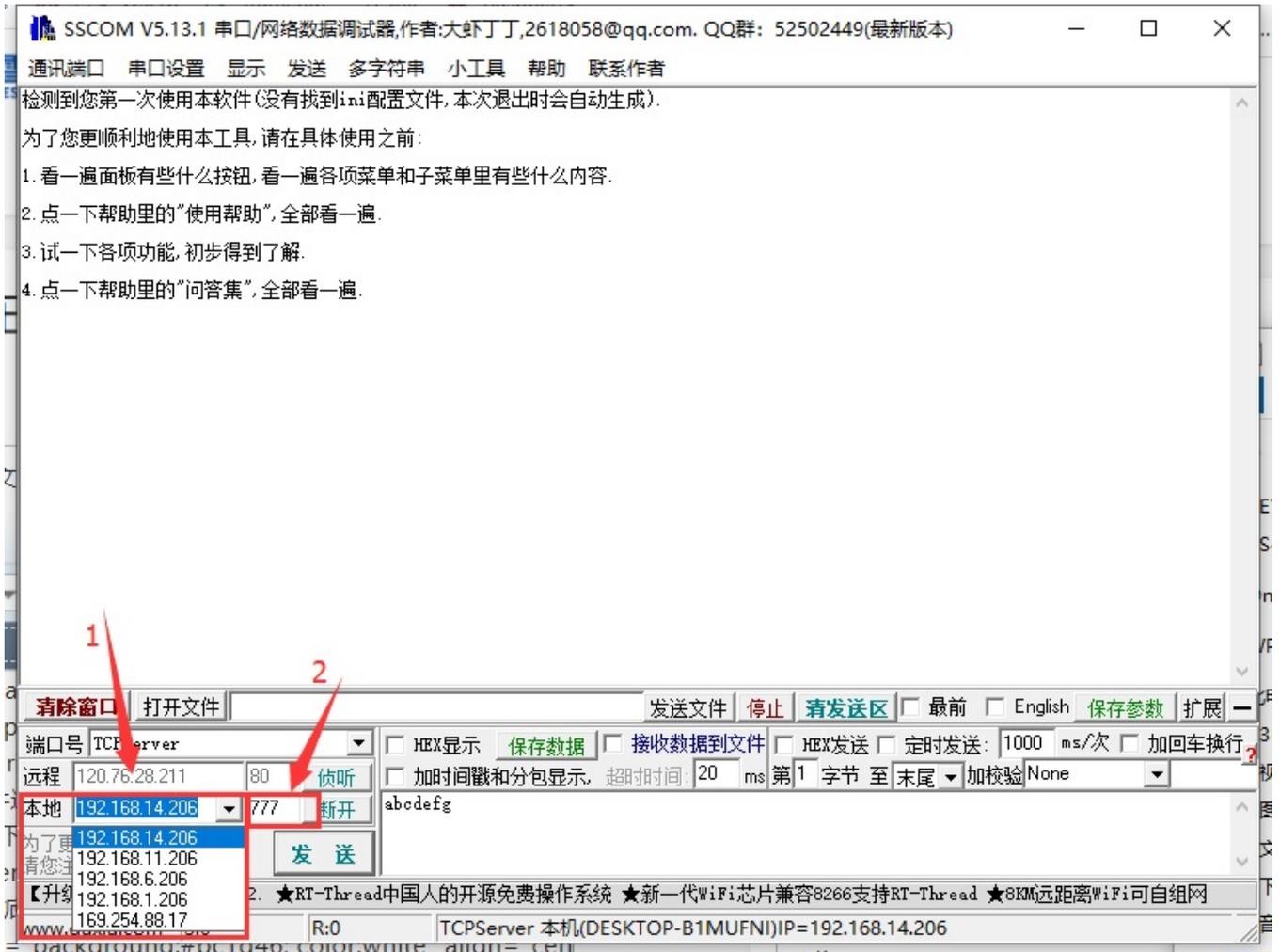
通过上位机打开端口2

通过路由连接模块，再运行NetModuleConfig.exe，搜索到设备后双击设备，将启用端口2勾选上即可



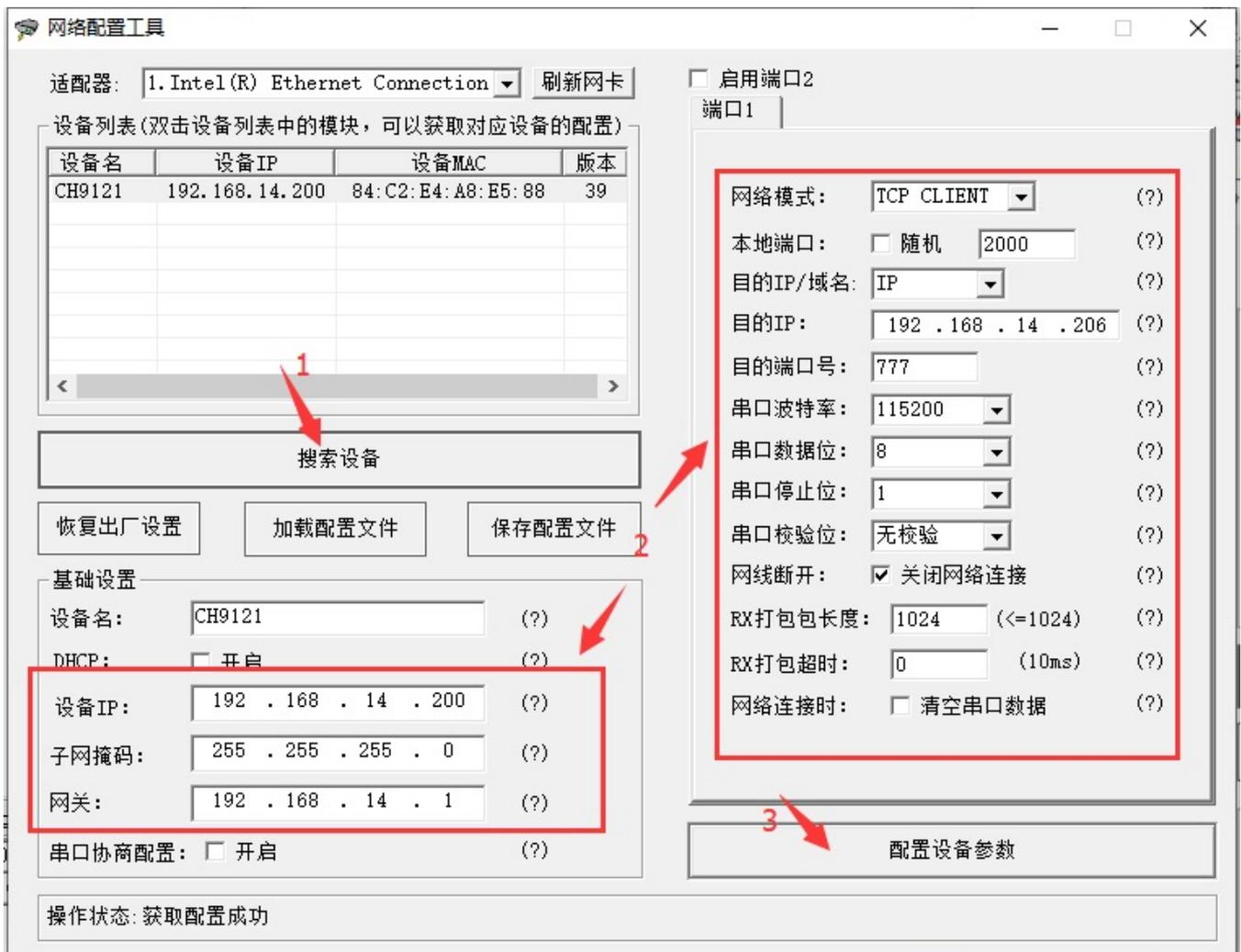
TCP CLIENT

1、选择SSCOM协议类型为TCP Servre,再通过SSCOM查看电脑IP跟端口号（也就是上面的目标IP跟目标端口）

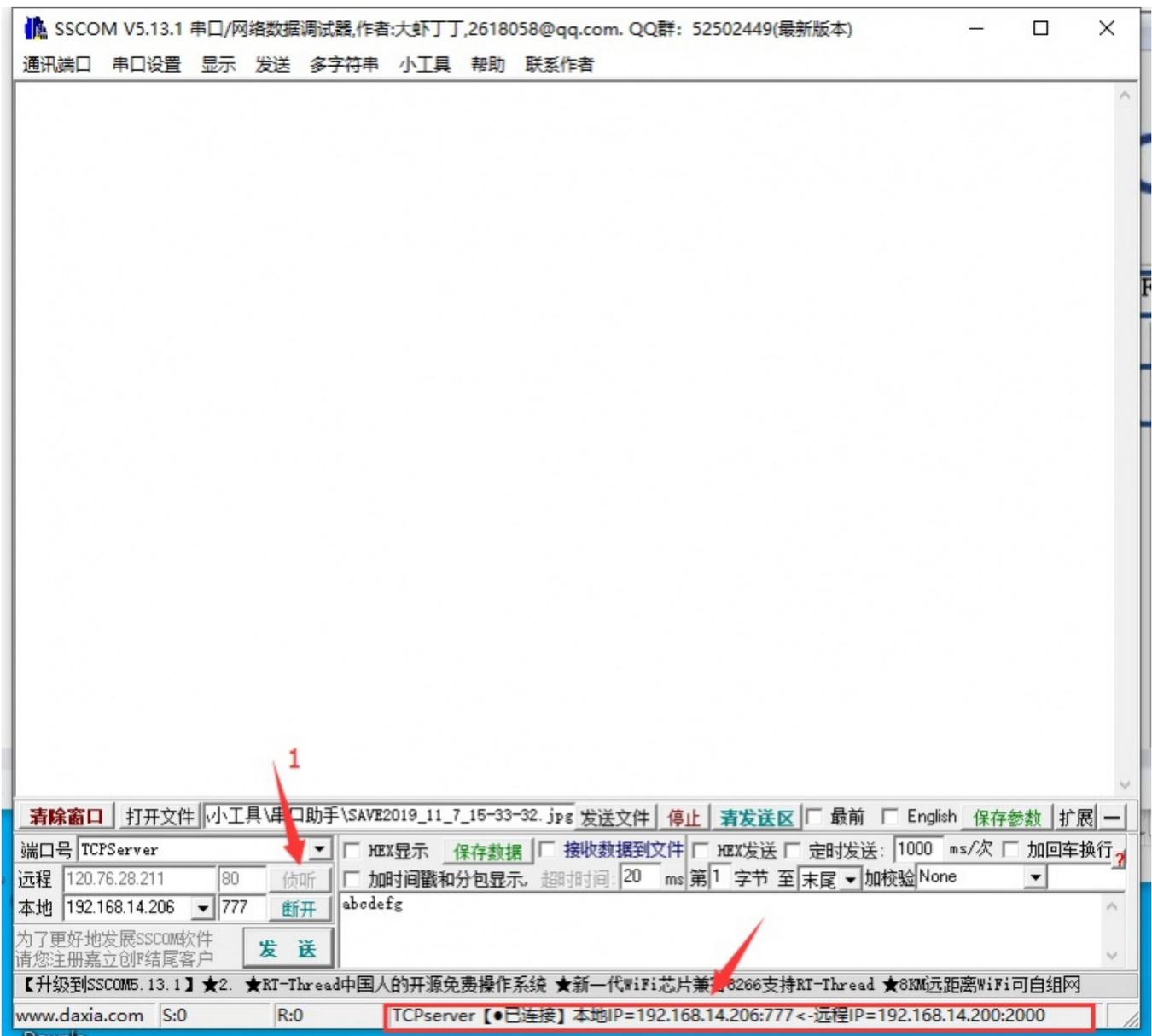


2、通过网线连接模块，再运行NetModuleConfig.exe设置模块参数，如下操作：

- (1).点击“搜索设备”，设备列表会显示出子网内模块；
- (2).双击设备列表里面的模块，在左侧修改参数，然后点击“配置设备参数”；
- (3).配置完成后，模块会重启，稍等一会，点击“搜索设备”，找到模块，查看配置结果。
- (4).根据实际需求修改模块参数，例如：设置网络方式为 TCP CLIENT，目的 IP，目的端口与 TCP SERVER 的 IP 和端口一致，再根据需求修改波特率等参数。然后点击配置设备参数。

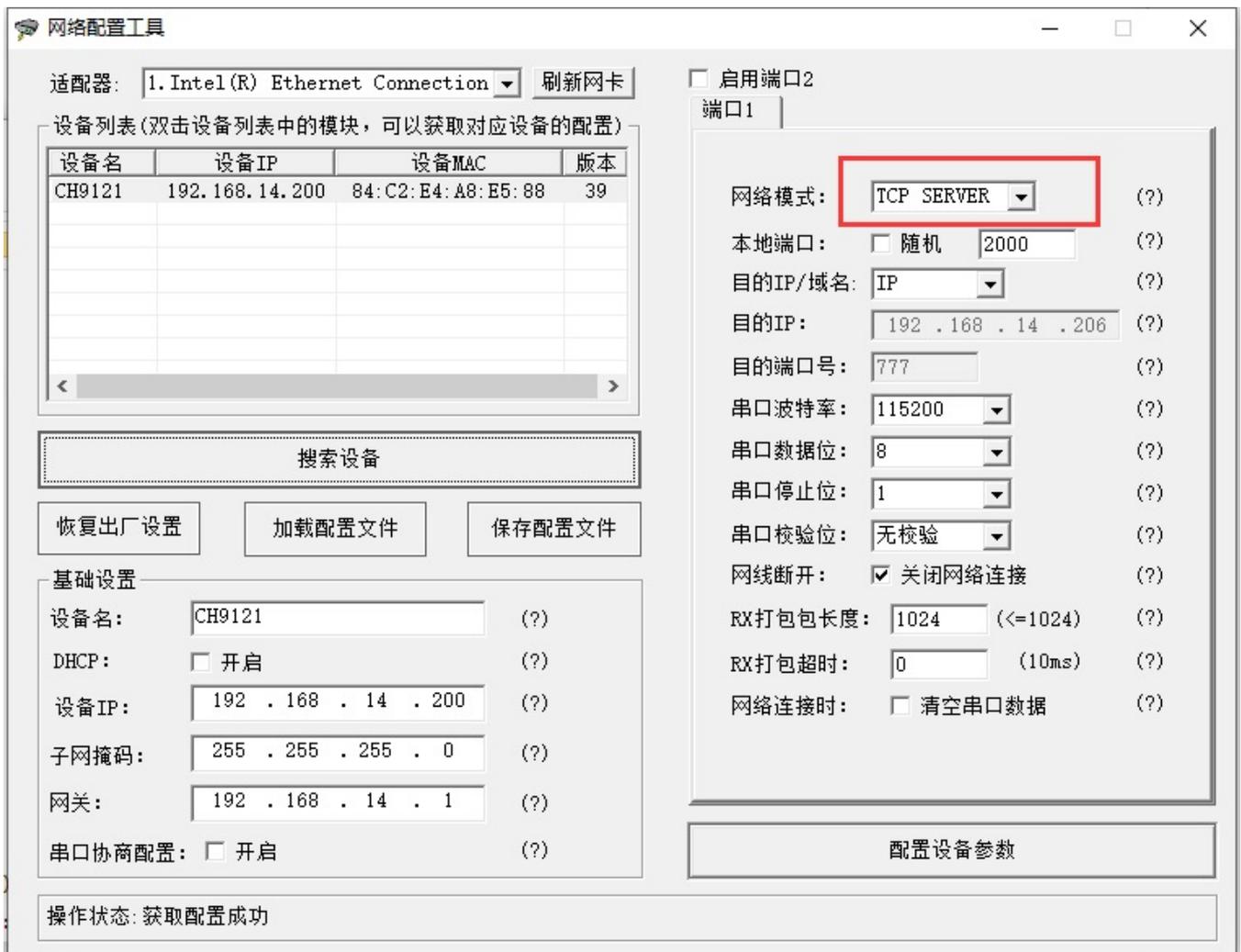


3、配置完之后，点击SSCOM的侦听，就连接起来了。

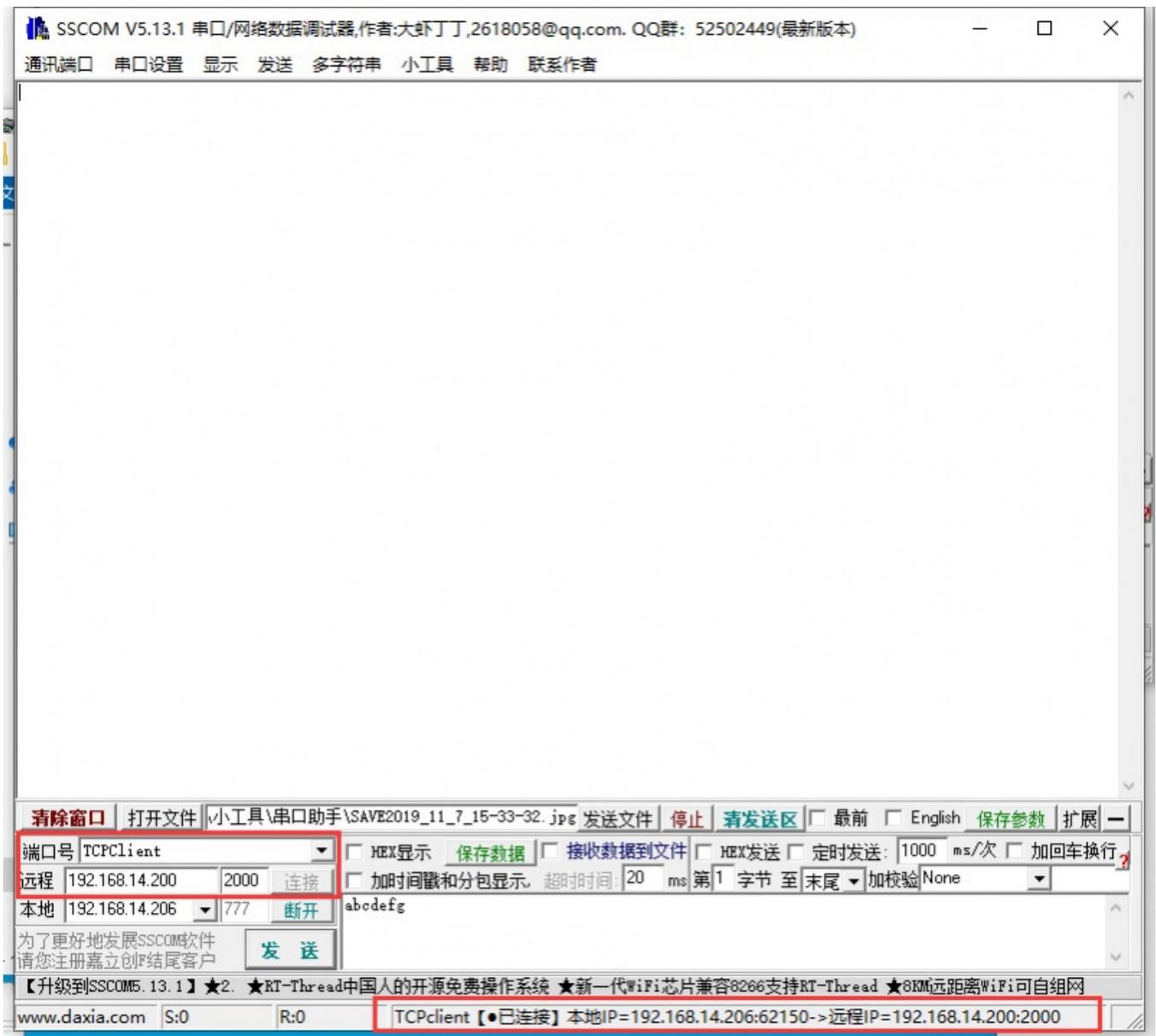


TCP SERVER

- 1、通过网线连接模块，再运行NetModuleConfig.exe设置模块参数，如下操作：
 - (1).点击“搜索设备”，设备列表会显示出子网内模块；
 - (2).双击设备列表里面的模块，在左侧修改网络模式为TCPServe，再根据需求修改模块参数，最后点击“配置设备参数”；



2、通过SSCOM去连接配置成TCP SERVER的模块, 连接起来会如下图显示:

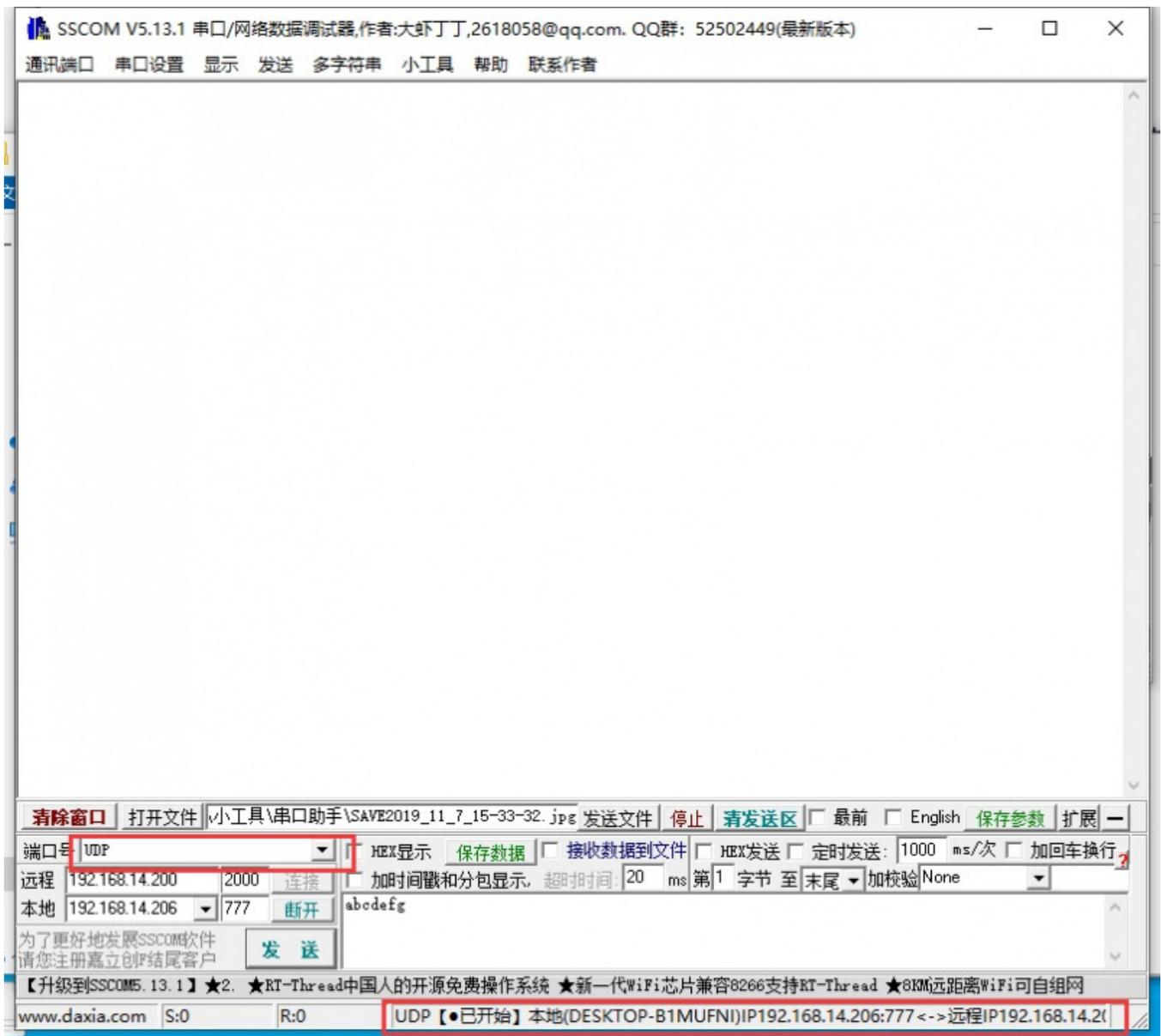


UDP CLIENT/SERVER

- 1、通过网线连接模块，再运行NetModuleConfig.exe设置模块参数，如下操作：
 - (1).点击“搜索设备”，设备列表会显示出子网内模块；
 - (2).双击设备列表里面的模块，在左侧修改网络模式为UDP CLIENT/Servre，再根据需求修改模块参数，最后点击“配置设备参数”；



2、通过SSCOM去连接配置成UDP CLIENT/Servre的模块 (SSCOM的UDP模式会自动连接, 不管你的模式配置成CLIENT/Servre) , 连接起来会如下图显示:



Pico快速上手

Pico百科

- 树莓派Pico百科 (墙裂推荐)

固件下载



C_Blink固件下载 [展开](#)

视频教程 (更新中)

PICO系列教程1——基础介绍 [展开](#)

PICO系列教程2——外设GPIO [展开](#)

PICO系列教程3——PWM(脉冲宽度调制) [展开](#)

PICO系列教程4——ADC (模拟数字转换器) [展开](#)

PICO系列教程5——UART (异步收发传输器) [展开](#)

PICO系列教程6——I2C (集成电路总线) [展开](#)

PICO系列教程7——SPI (串行外设接口) [展开](#)

文字教程 (更新中)

基础介绍

Raspberry Pi Pico的基础介绍

MicroPython系列 [展开](#)

C/C++系列 [展开](#)

开源例程

MircoPython视频例程(github)

MicroPython固件/Blink例程 (C)

树莓派官方C/C++示例程序 (github)

树莓派官方micropython示例程序 (github)

Pico使用教程

硬件连接

您可以对照以下表格连线。

Pico连接引脚对应关系

ETH	Pico	功能
5V	VSYS	电源输入
GND	GND	电源地
RXD1	GP0	串行数据输入
TXD1	GP1	串行数据输出
RXD2	GP4	串行数据输入
TXD2	GP5	串行数据输出
CFG0	GP14	网络配置使能脚
RST1	GP17	复位

直连



如需使用屏幕，请将Pico-ETH-CH9121上的CFG0短路帽拔掉。

基础介绍

MicroPython系列 [展开](#)

C/C++系列 [展开](#)

程序下载

打开树莓派终端，执行：
从我们官网下载。

```
sudo apt-get install p7zip-full
cd ~
sudo wget https://www.waveshare.net/w/upload/a/a4/Pico_ETH_CH9121_CODE.7z
7z x Pico_ETH_CH9121_CODE.7z -o./Pico_ETH_CH9121_CODE
cd ~/Pico_ETH_CH9121_CODE
cd Pico/c/build/
```

例程使用

C部分

- 以下教程为在树莓派上操作，但由于cmake的多平台、可移植的特点，在PC上也是能成功编译，但操作略有不同，需要您自行判断。

进行编译，请确保在c目录：

```
cd ~/Pico_ETH_CH9121_CODE/Pico/C/
```

Pico_ETH_CH9121_CODE/Pico/C/Serial Port Parameter Configuration:用来通过串口配置模式。

Pico_ETH_CH9121_CODE/Pico/C/RX_TX:用来进行信息收发，收到什么返回什么。

进入其中一个文件夹中创建并进入build目录,并添加SDK: 其中 ../../pico-sdk 是你的SDK的目录。我们示例程序中是有build的，直接进入即可

```
cd build
export PICO_SDK_PATH=../../pico-sdk
(注意：务必写对你自己的SDK所在路径)
```

执行cmake自动生成Makefile文件

```
cmake ..
```

执行make生成可执行文件，第一次编译时间比较久

```
make -j9
```

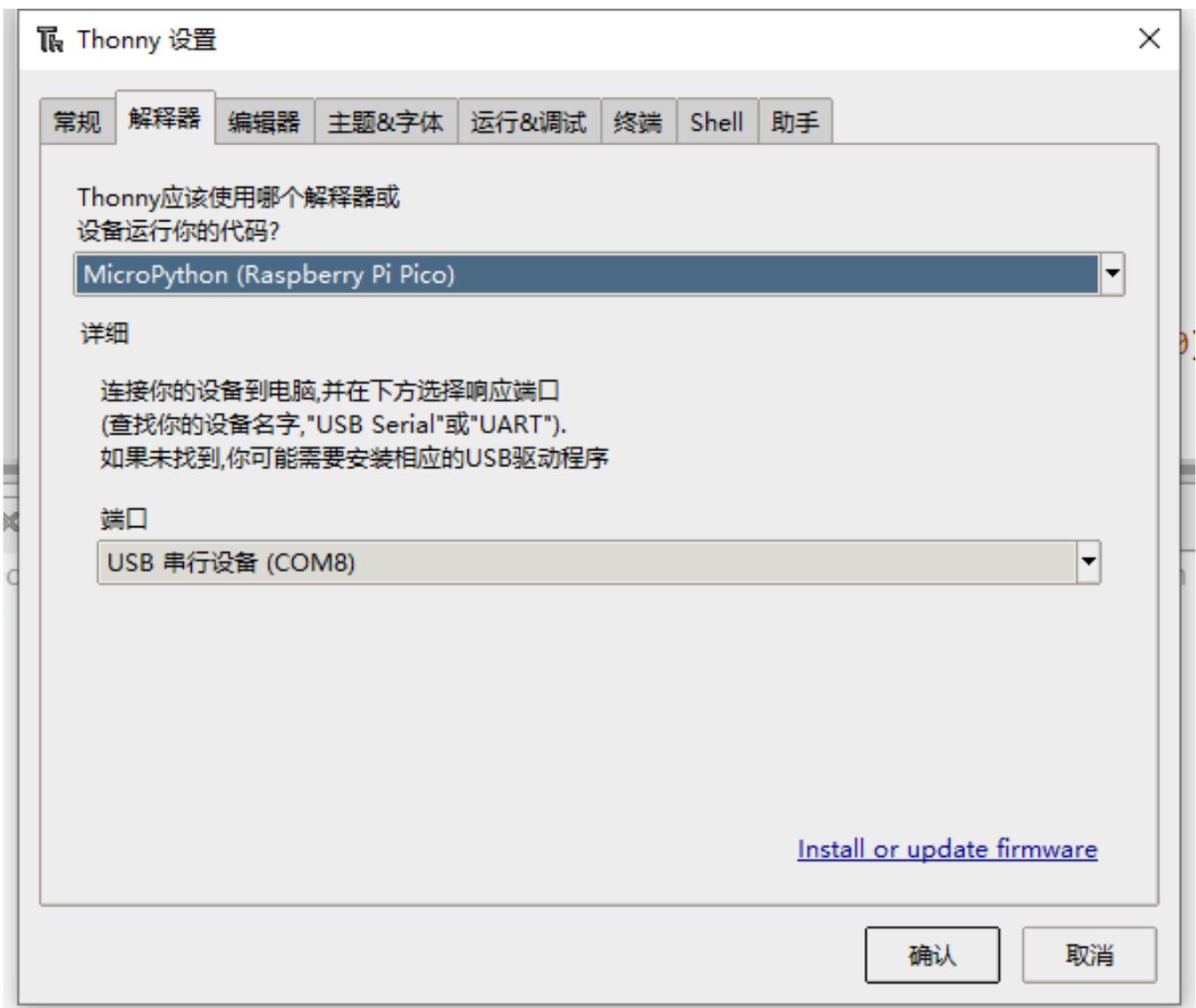
编译完成，会生成uf2文件。按住Pico板上的按键，将pico通过Micro USB线接到树莓派的USB接口，然后松开按键。接入之后，树莓派会自动识别到一个可移动盘（RPI-RP2），将build文件夹下的main.uf2 文件复制到识别的可移动盘（RPI-RP2）中即可。

```
cp main.uf2 /media/pi/RPI-RP2/
```

Python部分

windows环境下的使用

- 1.按住Pico板上的BOOTSET按键，将pico通过Micro USB线接到电脑的USB接口，待电脑识别出一个可移动硬盘（RPI-RP2）后，松开按键。
- 2.将python目录中rp2-pico-20210418-v1.15.uf2 文件复制到识别的可移动盘（RPI-RP2）中
- 3.打开Thonny IDE（注意：要使用最新版本的Thonny，否则是没有Pico的支持包的，当前Windows下的最新版本为v3.3.3）
- 4.点击工具->设置->解释器，如图所示选择Pico及对应的端口



本例程提供了两个程序：

Serial Port Parameter Configuration.py: 该程序用来通过串口配置模式。

RX_TX.py:这个用来进行信息收发，收到什么返回什么。

- 5.文件->打开->RX_TX.py,点击运行即可，如下图所示：

注意：如果想自启动，请将RX_TX.py重命名为main.py后保存到Pico里面。

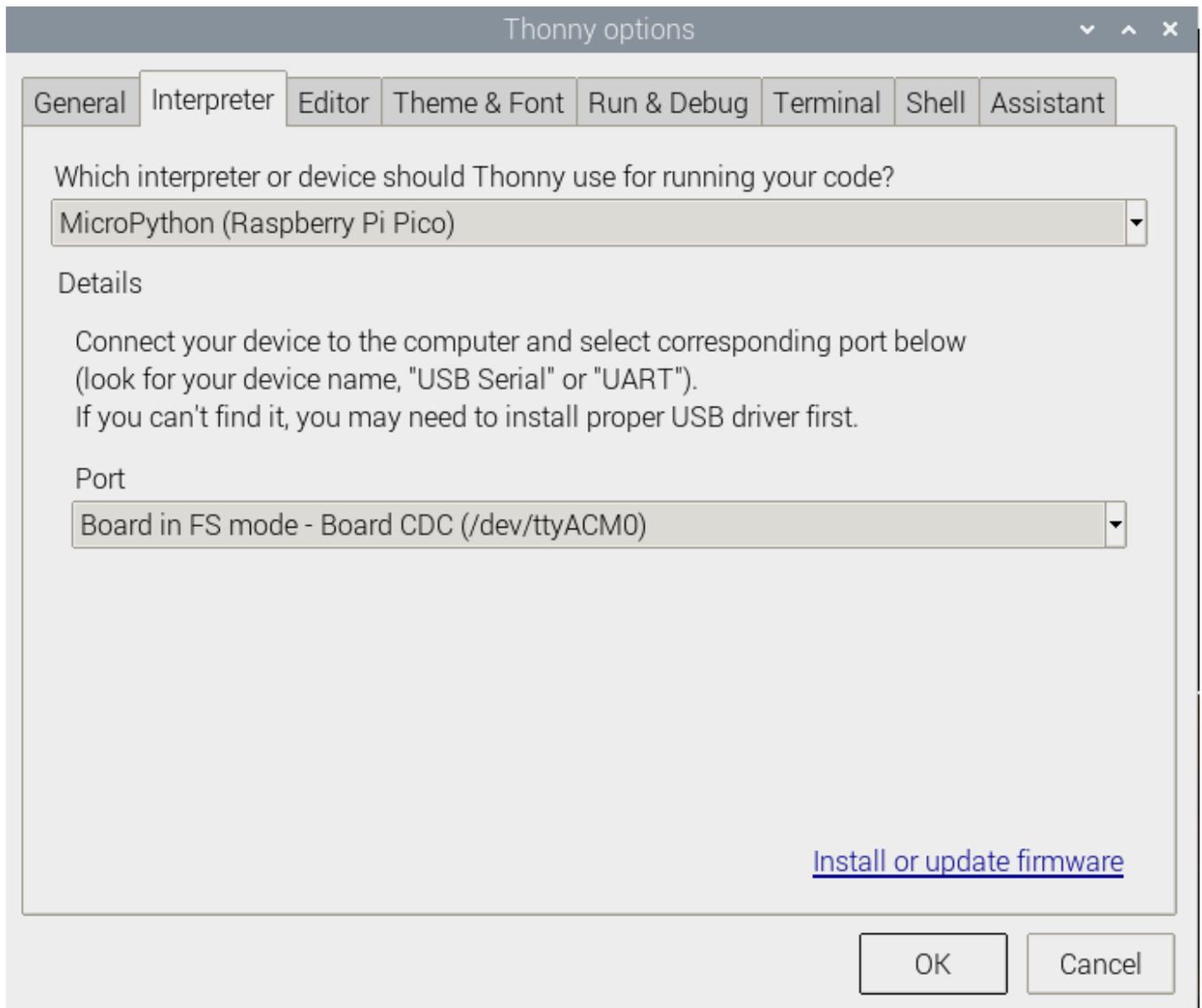
```
Shell X
MicroPython v1.13-290-g556ae7914 on 2021-01-21; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
```

树莓派环境下的使用

- 1.刷固件的过程与Windows上一致，你可以选择在PC或者树莓派上将rp2-pico-20210418-v1.15.uf2 文件拷入pico中。
- 2.在树莓派上打开Thonny IDE（点击树莓logo -> Programming -> Thonny Python IDE），你可以在Help->About Thonny查看版本信息

以确保你的版本是有Pico支持包的，同样你可以点击Tools -> Options... -> Interpreter选择MicroPython(Raspberry Pi Pico 和ttyACM0端口

如图所示：



如果你当前的Thonny版本没有pico支持包，输入以下指令来更新Thonny IDE

```
sudo apt upgrade thonny
```

3.点击File->Open...->python/RX_TX.py，运行脚本即可

注意：如果想自启动，请将RX_TX.py重命名为main.py后保存到Pico里面。

代码简析

C部分

通过串口配置参数(根据你的需求进行修改即可):

- 数据类型

```
#define UCHAR unsigned char
#define UBYTE uint8_t
#define UWORD uint16_t
#define UDOUBLE uint32_t
```

■ 模块初始化:

```
void CH9121_init(void);
```

■ 本程序只进行了简单的配置, 如需配置其他功能可查阅串口控制命令, 自行配置即可:

```
UCHAR CH9121_Mode           //模式选择
UCHAR CH9121_LOCAL_IP[4]   //本地IP
UCHAR CH9121_GATEWAY[4]    //网关
UCHAR CH9121_SUBNET_MASK[4] //子网掩码
UCHAR CH9121_TARGET_IP[4]  //目标IP
UWORD CH9121_PORT1         //本地端口
UWORD CH9121_TARGET_PORT   //目标端口
UDOUBLE CH9121_BAUD_RATE   //串口波特率
```

■ 根据串口控制命令合理利用下面函数可进行参数配置:

```
void CH9121_TX_4_bytes(UCHAR data, int command); //用于模式、是否随机端口、端口是否断开
网络连接、是否清空串口数据、是否开启DHCP、是否开启串口2
void CH9121_TX_5_bytes(UWORD data, int command); //用于设置串口的端口号
void CH9121_TX_7_bytes(UCHAR data[], int command); //用于设置IP、子网掩码、网关
void CH9121_TX_BAUD(UDOUBLE data, int command); //用于设置串口的波特率
void CH9121_Eed(); //更新配置参数至 EEPROM、执行配置, 复位 9121、离开配置模式
```

Python部分

用户只需修改Serial Port Parameter Configuration.py里面如下所示的值, 就可以对模块进行串口参数配置:

```
MODE = 1 #0:TCP Server 1:TCP Client 2:UDP Server 3:UDP Client
GATEWAY = (169, 254, 88, 1) # GATEWAY
TARGET_IP = (169, 254, 88, 17)# TARGET_IP
LOCAL_IP = (169,254,88,70) # LOCAL_IP
SUBNET_MASK = (255,255,255,0) # SUBNET_MASK
LOCAL_PORT1 = 5000 # LOCAL_PORT1
LOCAL_PORT2 = 4000 # LOCAL_PORT2
TARGET_PORT = 3000 # TARGET_PORT
BAUD_RATE = 115200 # BAUD_RATE
```

资料

文档

- 原理图

程序

- 示例程序

数据手册

- 芯片 CH9121数据手册
- CH9121 串口控制指令集

配置跟调试软件

- 上位机跟SSCOM