

斐讯 M1 通过 TTL 转 USB 升级固件接入 HA 教程

作者：wzj

1. 前言

2018 年 8 月 28 日,在翰思彼岸发了首贴<https://bbs.hassbian.com/thread-4682-1-1.html>,原本计划 9 月 7 日发刷机教程的,为了保证后续升级可以更加方便快捷,添加了本地化的 tftp OTA 功能,可实现局域网升级。

现有功能:

- 1、可通过 mqtt 直接接入到自建的 homeassistant 服务器;
- 2、可配置 wifi 的 SSID 名称和密码;
- 3、可设置 MQTT 的服务器 IP 地址或是域名、端口号、用户名、密码;
- 4、tftp OTA 升级功能。

存在问题:

- 1、显示面板的 wifi 灯,未能根据联网情况点亮,现在还未弄清楚原理,待后续研究改进;
- 2、对于 wifi、MQTT 的参数配置需要通过命令行的方式,比较不人性化,待后续改进;
- 3、对于网友提出的设置时区,还未弄清楚原理,待后续研究;
- 4、其他未发现的问题;
- 5、升级方式只能本地化,后续会改进,采用连接到指定 OTA 服务器上进行升级。

2. 准备工作

2.1 硬件

序号	工具名称	作用
1	美工刀（或是相应的拆机工具）	拆开斐讯 M1 亚克力板
2	1 个 TTL 转 USB、4 条杜邦线（一头公，一头母；或是针式的杜邦线）	硬件连接升级程序
3	1 把小的十字螺丝刀	用来拆开斐讯 M1 内部主板的螺丝

2.2 软件

序号	软件名称	作用
1	SecureCRT	选择该工具的原因是波特率可以选择 921600，进入 bootloader 升级程序，以及设置参数
2	Tftpd32	用于局域网 OTA 升级使用
3	Notepad++	用于对 bin 文件生成 MD5 码
4	UltraEdit	用于打开 bin 文件添加 16 进制的 MD5 码

3. 刷固件教程

3.1 拆机

操作步骤：

1、做好拆机心里准备!! 拆机! 拆机! 拆机! 重要的事情说三遍!!!

2、拿出你的美工刀，刀片沿着边缘插入，大约插入 2 公分，主要看图 1-1 位置的卡扣，只要把图上几个位置卡扣脱离，后续就不用用很大力气，就可以成为图 1-2 的成果，拆成如图 1-2，那说明你成功啦!! 拆开面盖皆大欢喜，拿起设备好好观摩下，有没有哪里坑碰了，坑碰了没事，仅此一次!!!

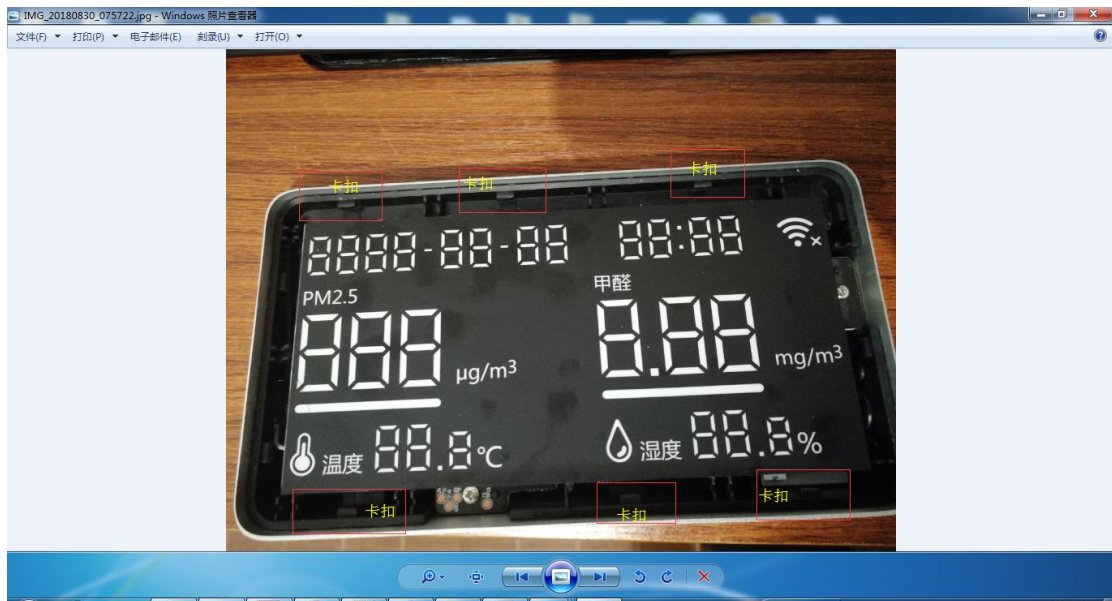


图 1-1

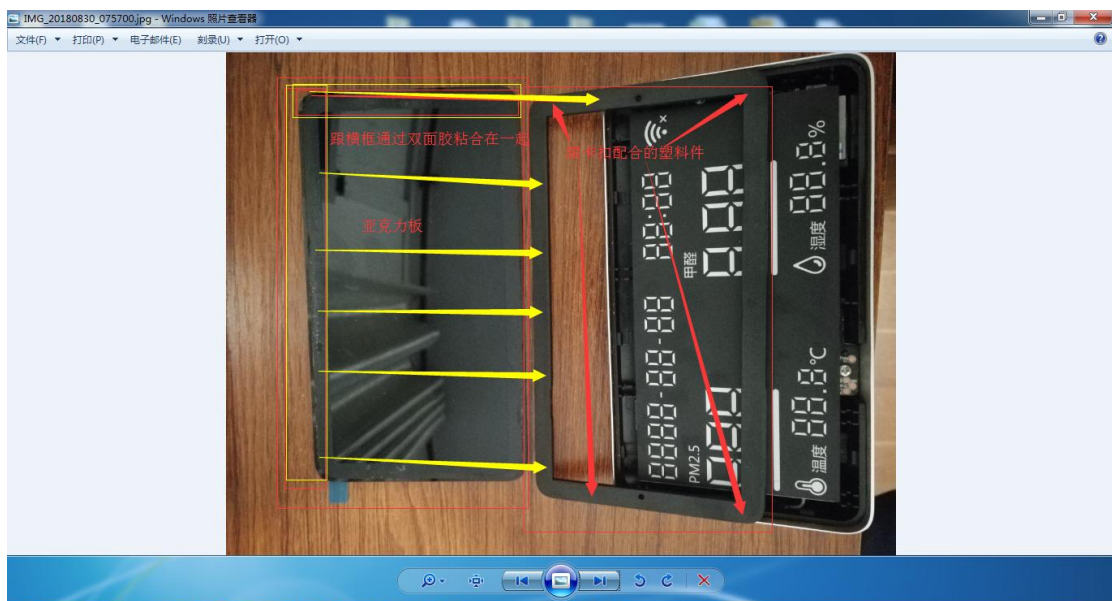


图 1-2

3.2 EMW3080B 模块引脚定义

引脚说明：

- 1、引脚 9 代表 TTL 中的 TX，引脚 10 代表 TTL 中的 RX，作用是用于进入 bootloader 模式，便于升级应用程序
- 2、引脚 16 代表 TTL 中的 3.3V，引脚 17 代表 TTL 的 GND
- 3、引脚 19 代表是进入 B00T 模式，需要低电平才能触发进入 bootloader 模式
- 4、引脚 24 代表 TTL 中的 TX，引脚 25 代表 TTL 中的 RX，作用是设置 wifi、MQTT、TFTP OTA 升级都是在这个串口完成

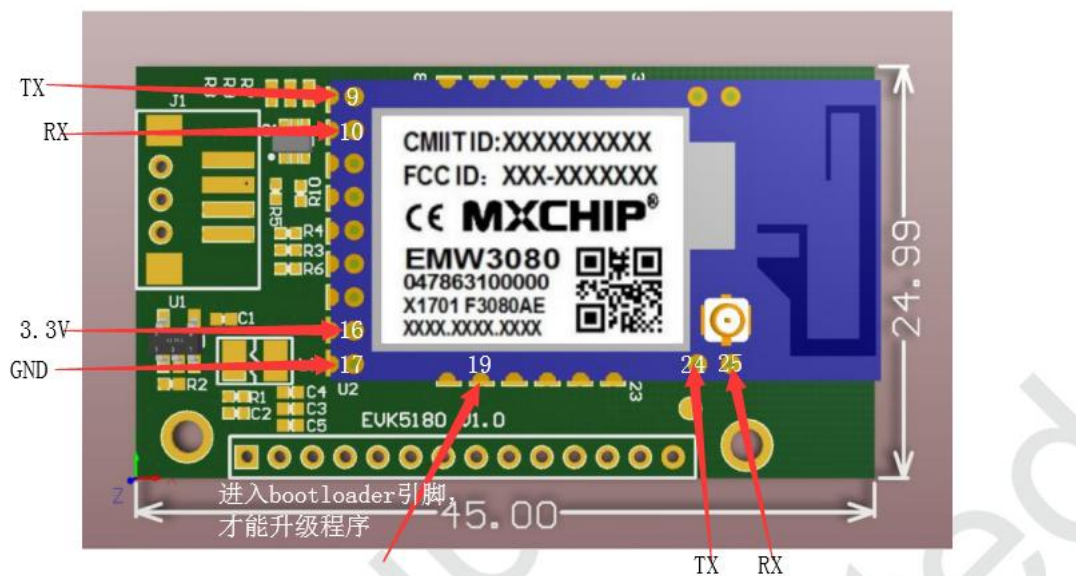


图 1-3

3.3 TTL 转 USB 工具与 EMW3080 硬件连接

重要说明：神器的 TX 与 EMW3080 的 RX 连接在一起，接错不要找我 😱 !!!

虚线是为了后续使用，主要作用是低电平，保证 EMW3080 模块可以进入 bootloader

模式，我们才可以升级应用程序固件，终于要摆脱斐讯的控制啦!!! 😊 😊

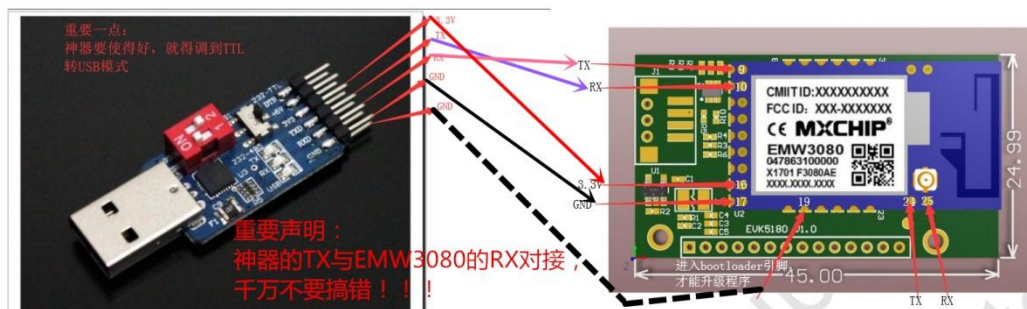


图 1-4

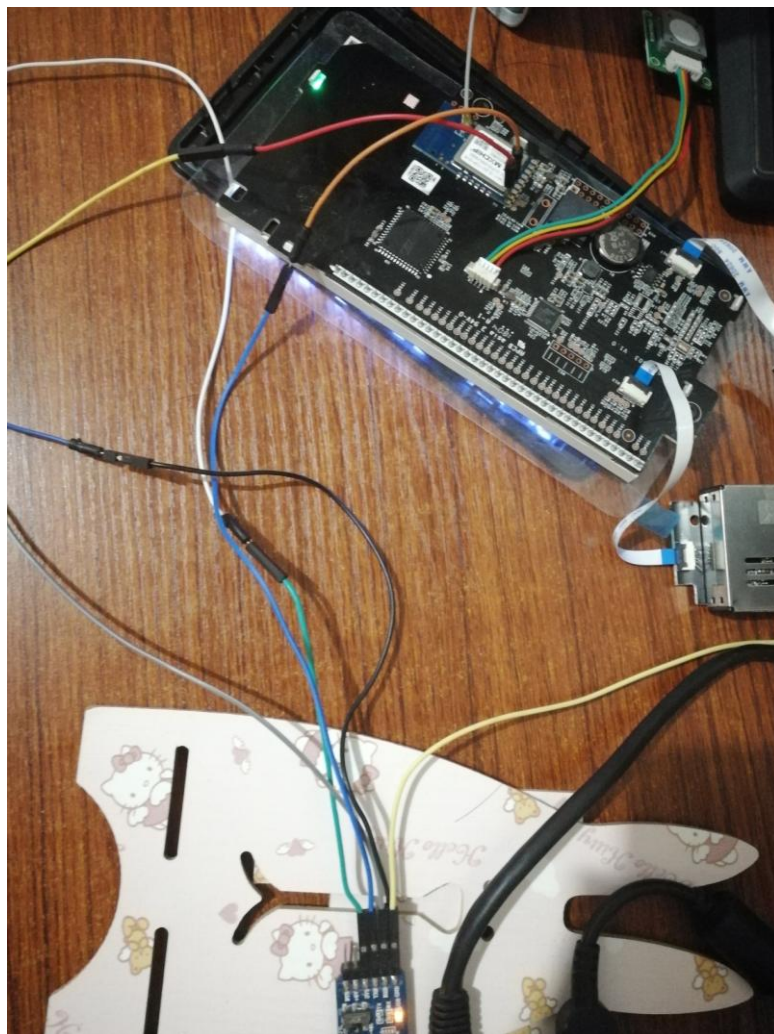


图 1-5

3.4 SecureCRT 软件设置

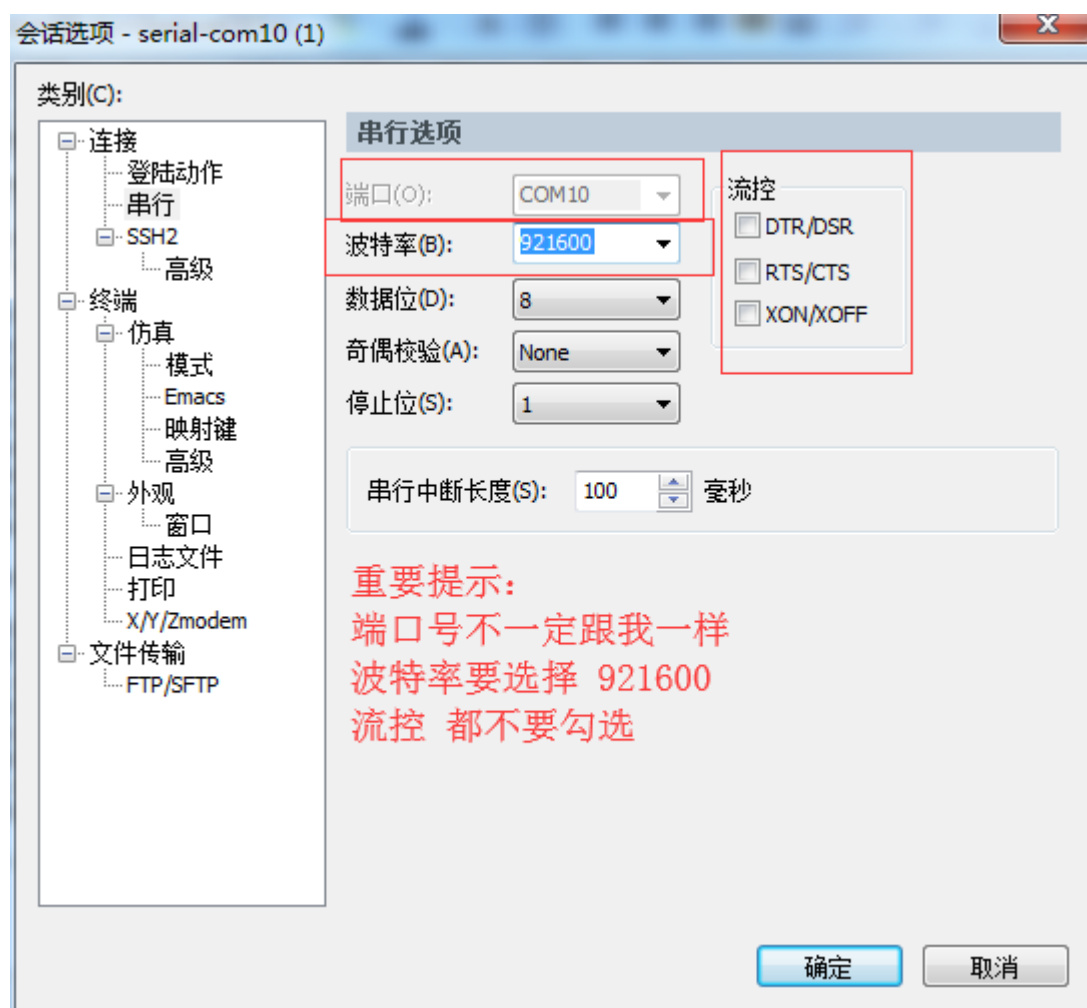


图 1-6

3.5 进入 EMW3080 bootloader 模式

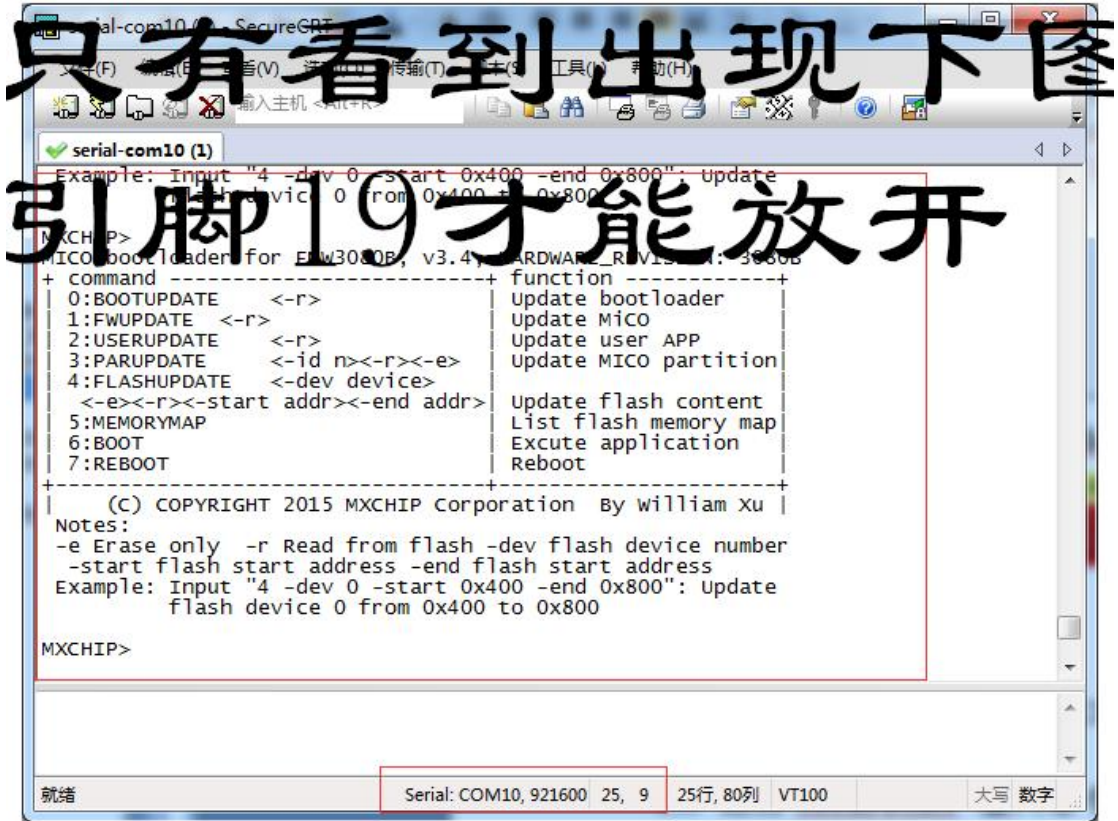
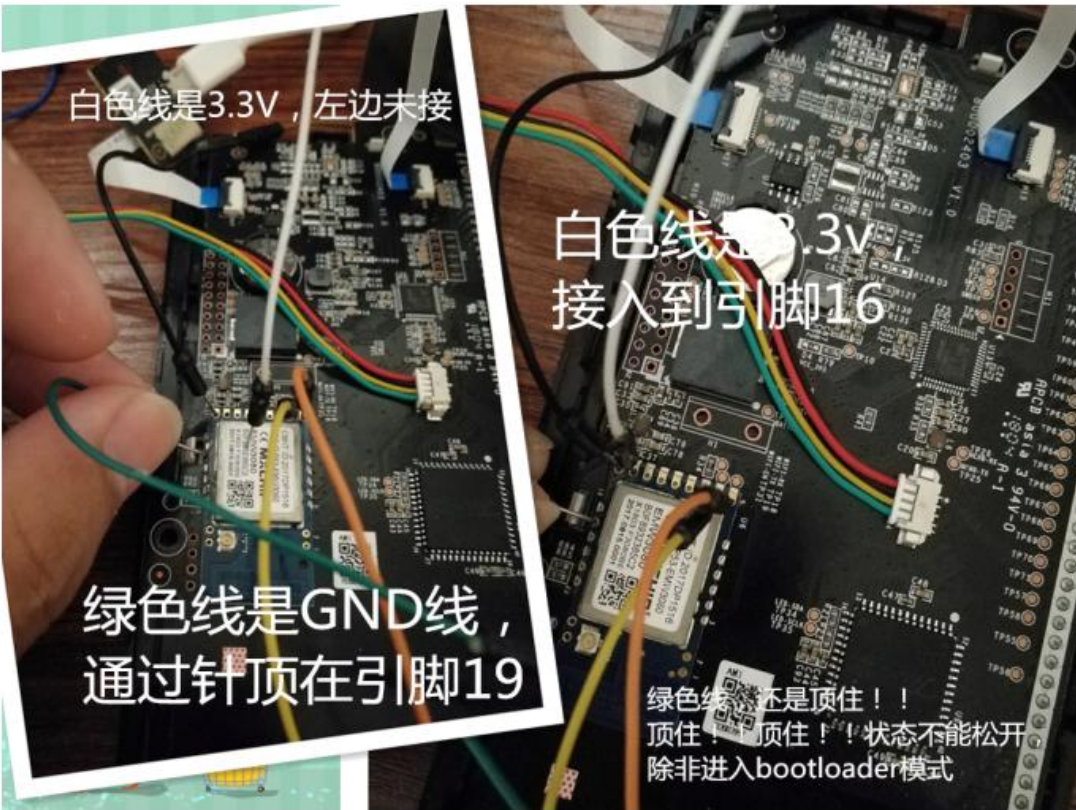


图 1-7

3.6 升级固件

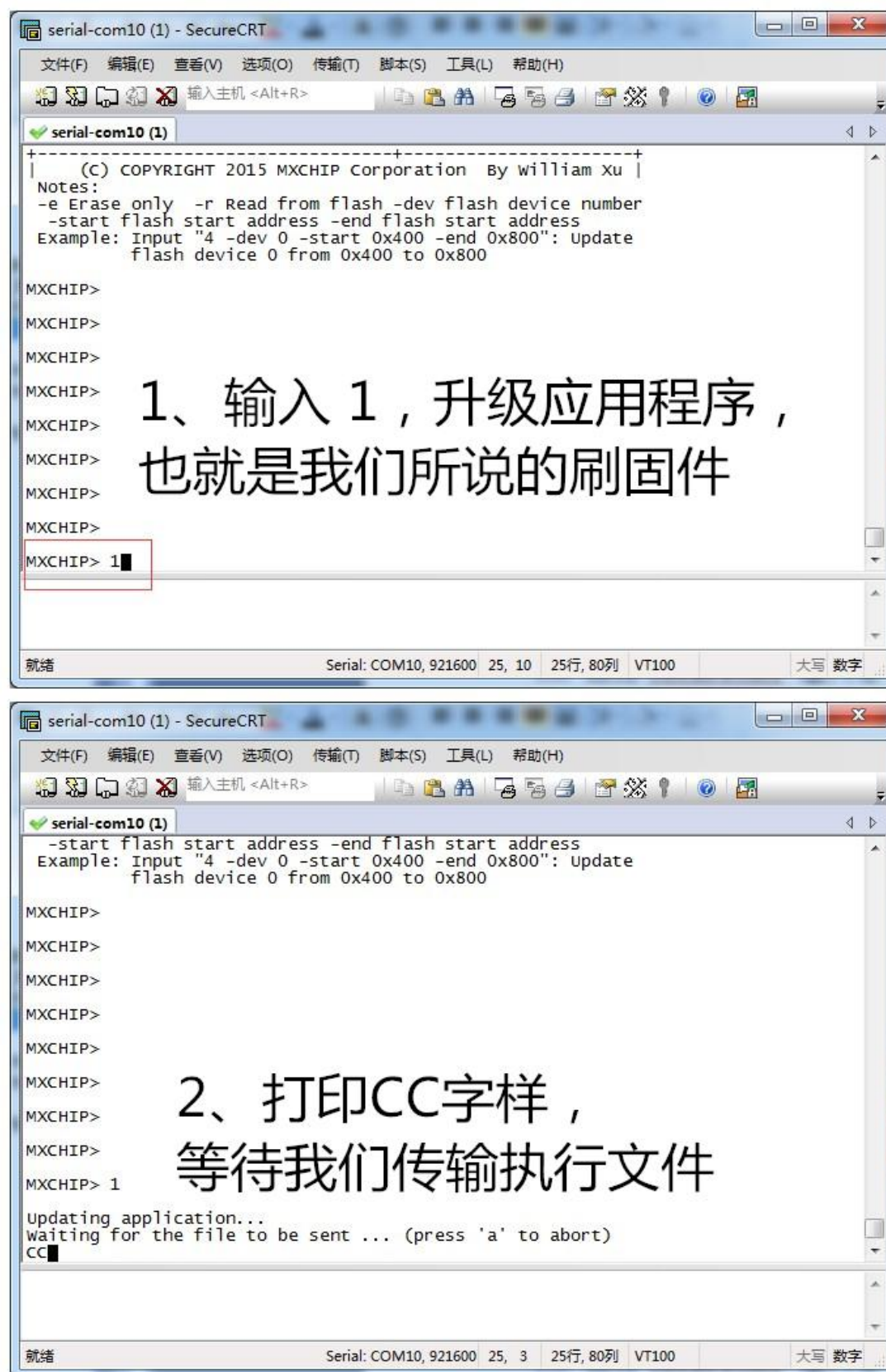


图 1-8

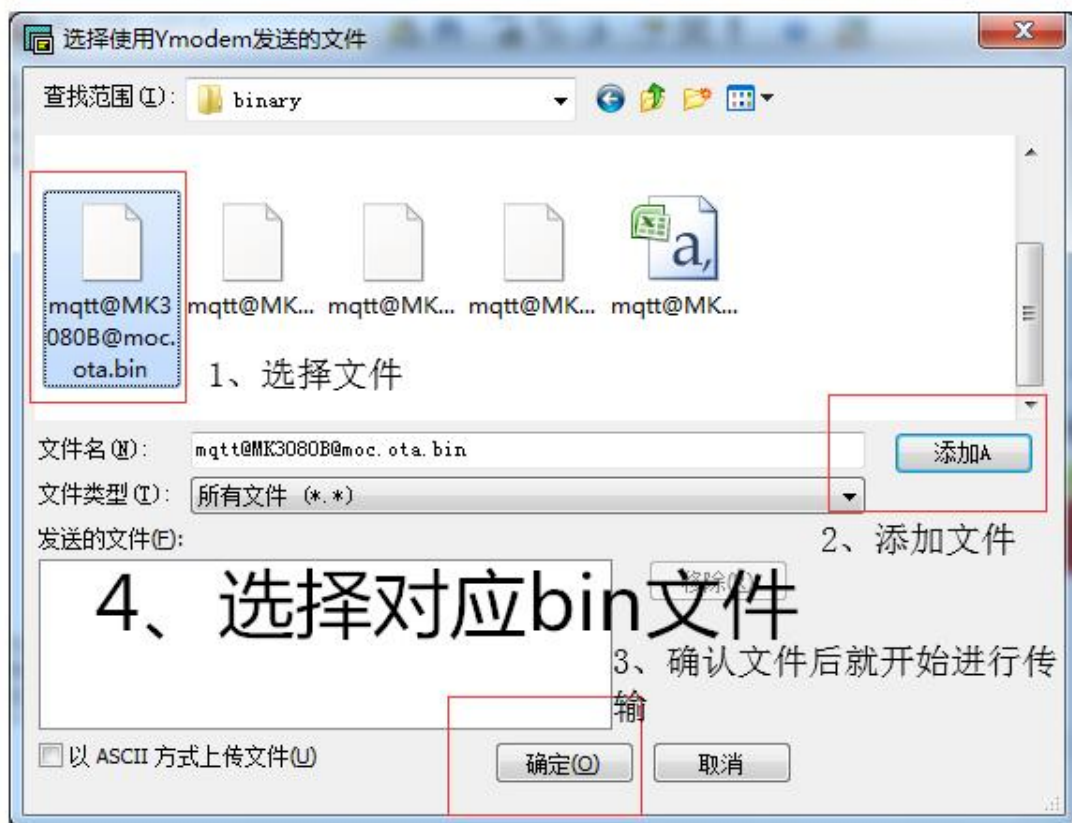
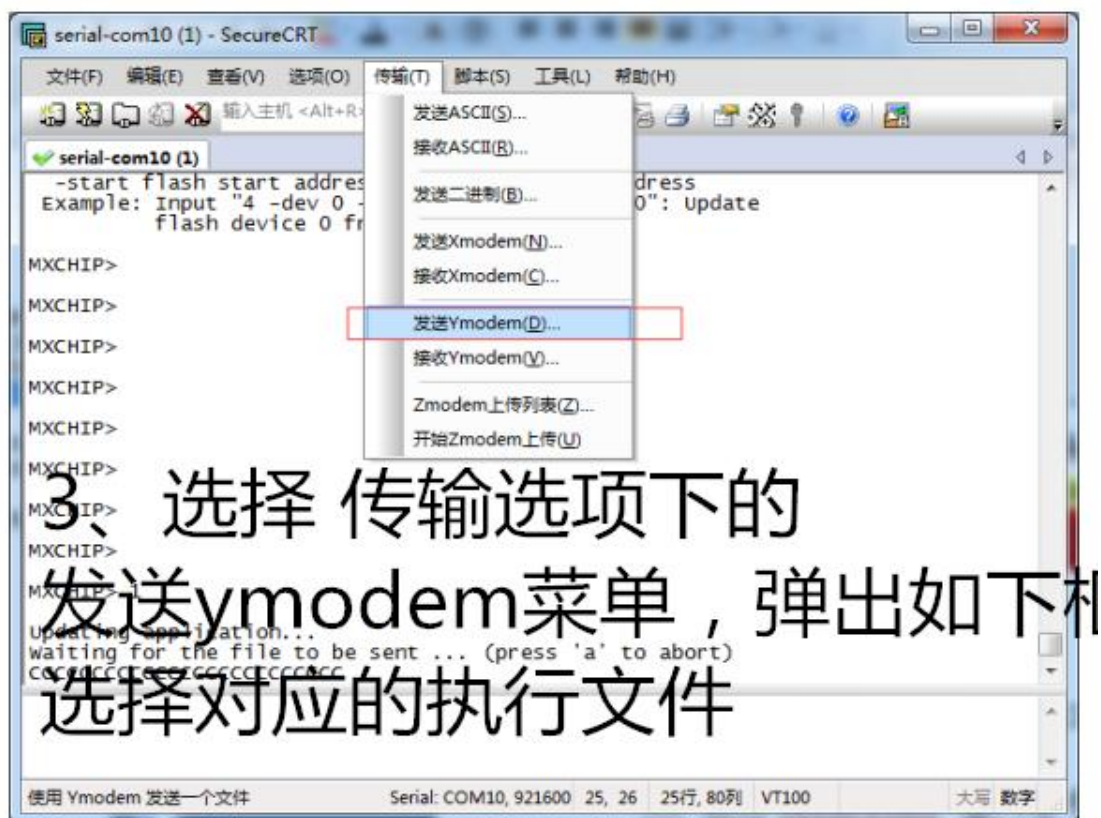
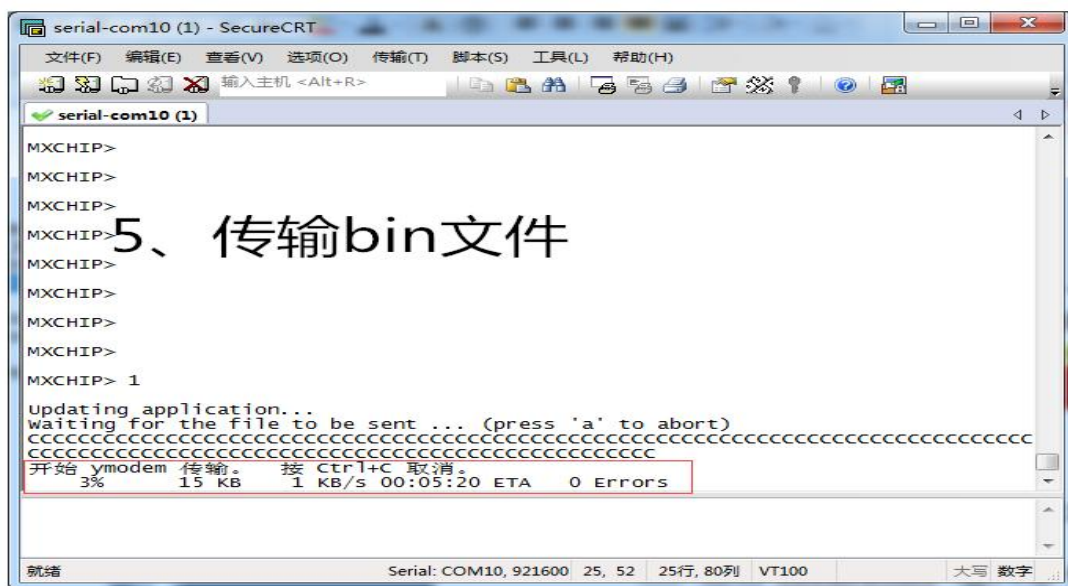
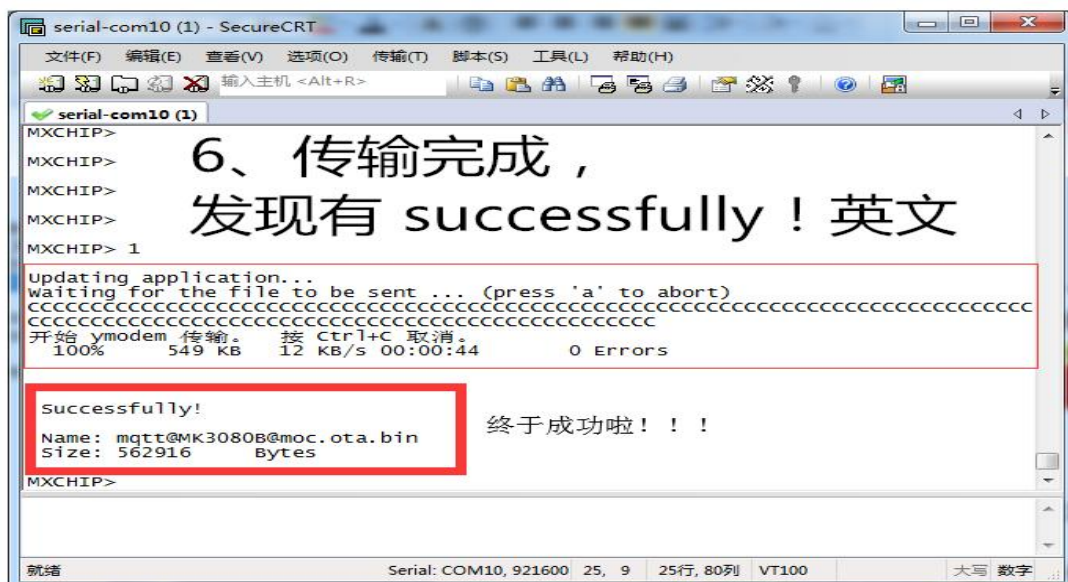


图 1-9

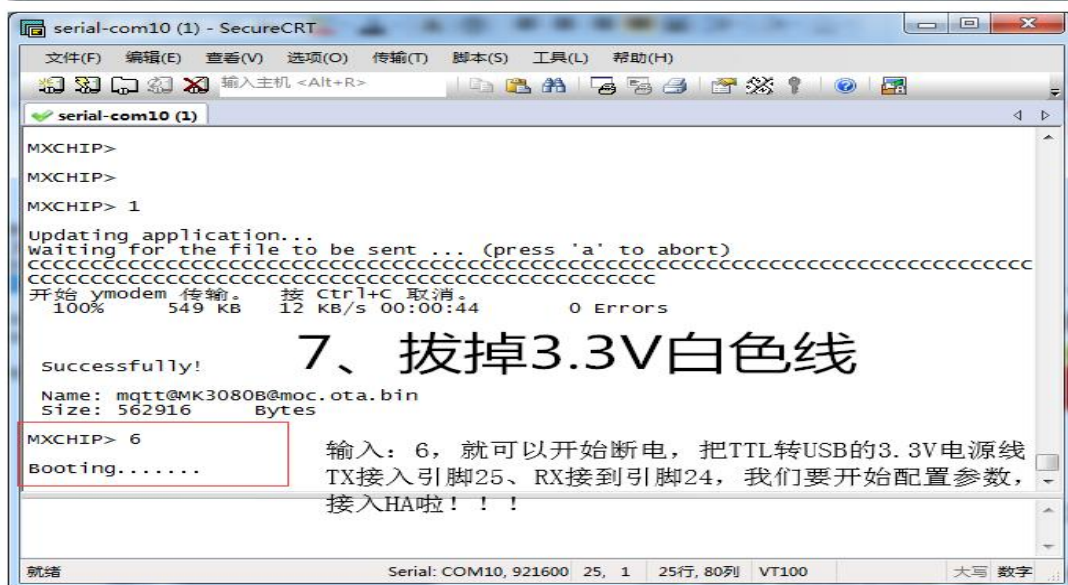


5、传输bin文件



6、传输完成，发现有 successfully！英文

终于成功啦！！



7、拔掉3.3V白色线

输入：6，就可以开始断电，把TTL转USB的3.3V电源线TX接入引脚25、RX接到引脚24，我们要开始配置参数，接入HA啦！！

图 1-10

3.7 修改 SecureCRT 软件串口参数设置

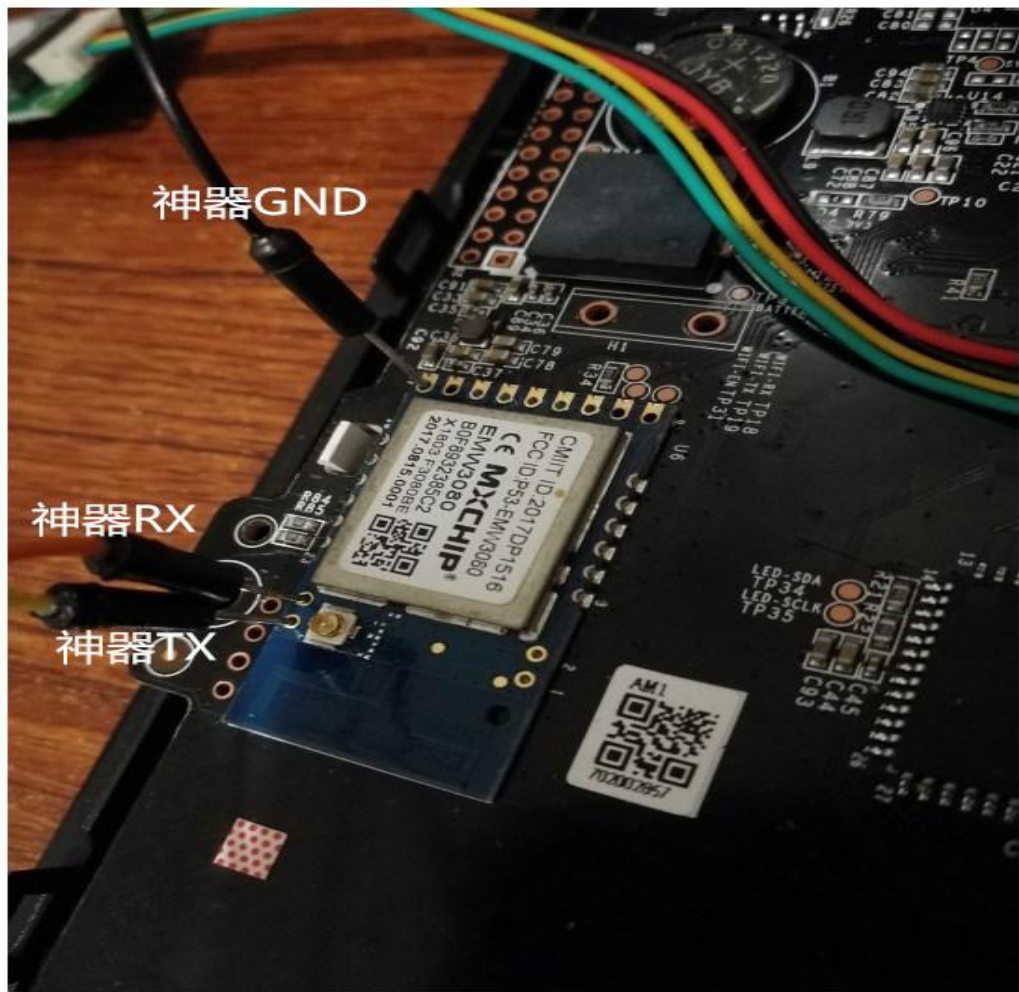
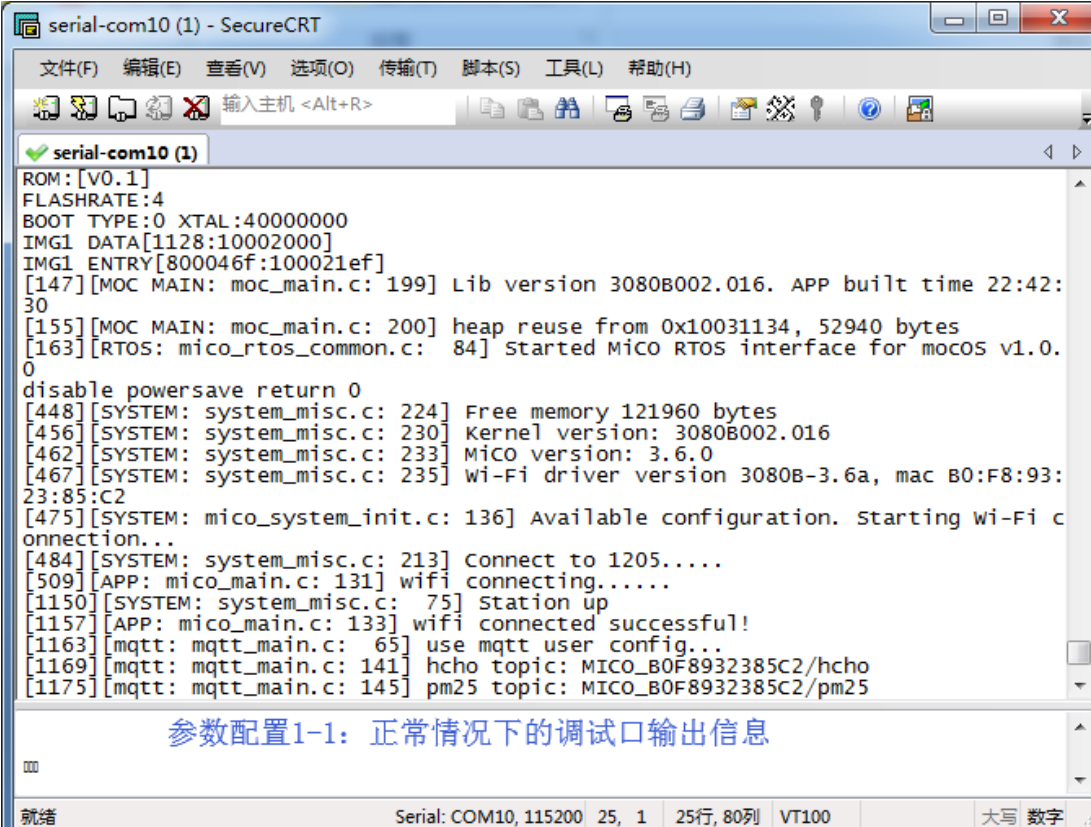


图 1-10

3.8 通过 SecureCRT 输入命令行设置 wifi、mqtt 参数



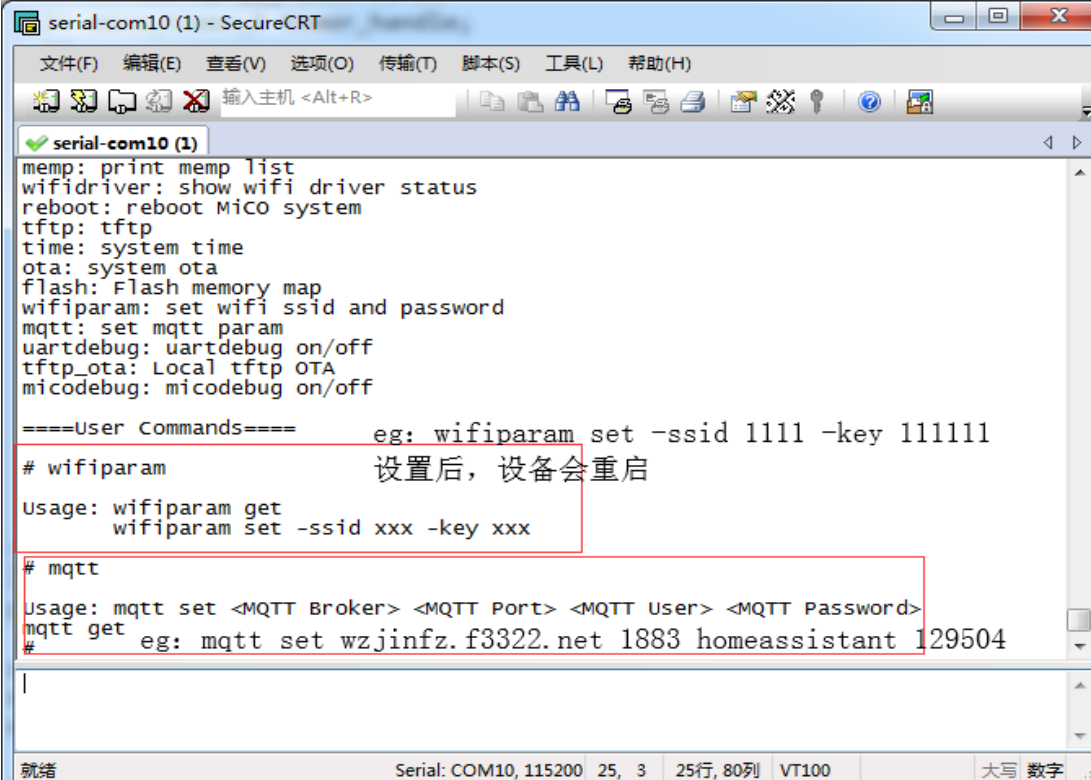
The image shows a SecureCRT terminal window titled 'serial-com10 (1)'. The terminal displays the boot process of a device, including ROM version, flash rate, boot type, and various system initialization messages. Key messages include 'Lib version 3080B002.016', 'heap reuse from 0x10031134, 52940 bytes', 'Started MICO RTOS interface for mocOS v1.0', 'Free memory 121960 bytes', 'Kernel version: 3080B002.016', 'MICO version: 3.6.0', 'Wi-Fi driver version 3080B-3.6a, mac B0:F8:93:23:85:C2', and 'Available configuration. starting wi-Fi connection...'. The terminal also shows the Wi-Fi connection process, including 'Connect to 1205....', 'wifi connecting.....', 'Station up', and 'wifi connected successful!'. Finally, it shows the MQTT configuration process, including 'use mqtt user config...', 'hcho topic: MICO_B0F8932385C2/hcho', and 'pm25 topic: MICO_B0F8932385C2/pm25'. Below the terminal output, there is a blue text label '参数配置1-1: 正常情况下的调试口输出信息'. The status bar at the bottom indicates '就绪', 'Serial: COM10, 115200', '25, 1', '25行, 80列', 'VT100', '大写', and '数字'.

```
ROM:[V0.1]
FLASHRATE:4
BOOT TYPE:0 XTAL:40000000
IMG1 DATA[1128:10002000]
IMG1 ENTRY[800046f:100021ef]
[147][MOC MAIN: moc_main.c: 199] Lib version 3080B002.016. APP built time 22:42:30
[155][MOC MAIN: moc_main.c: 200] heap reuse from 0x10031134, 52940 bytes
[163][RTOS: mico_rtos_common.c: 84] Started MICO RTOS interface for mocOS v1.0.
0
disable powersave return 0
[448][SYSTEM: system_misc.c: 224] Free memory 121960 bytes
[456][SYSTEM: system_misc.c: 230] Kernel version: 3080B002.016
[462][SYSTEM: system_misc.c: 233] MICO version: 3.6.0
[467][SYSTEM: system_misc.c: 235] Wi-Fi driver version 3080B-3.6a, mac B0:F8:93:23:85:C2
[475][SYSTEM: mico_system_init.c: 136] Available configuration. starting wi-Fi connection...
[484][SYSTEM: system_misc.c: 213] Connect to 1205....
[509][APP: mico_main.c: 131] wifi connecting.....
[1150][SYSTEM: system_misc.c: 75] Station up
[1157][APP: mico_main.c: 133] wifi connected successful!
[1163][mqtt: mqtt_main.c: 65] use mqtt user config...
[1169][mqtt: mqtt_main.c: 141] hcho topic: MICO_B0F8932385C2/hcho
[1175][mqtt: mqtt_main.c: 145] pm25 topic: MICO_B0F8932385C2/pm25
```

参数配置1-1: 正常情况下的调试口输出信息

就绪 Serial: COM10, 115200 25, 1 25行, 80列 VT100 大写 数字

图 1-11



The image shows a SecureCRT terminal window titled 'serial-com10 (1)'. The terminal displays a list of user commands and their usage. The commands include 'memp: print mempool list', 'wifidriver: show wifi driver status', 'reboot: reboot MICO system', 'tftp: tftp', 'time: system time', 'ota: system ota', 'flash: Flash memory map', 'wifiparam: set wifi ssid and password', 'mqtt: set mqtt param', 'uartdebug: uartdebug on/off', 'tftp_ota: Local tftp OTA', and 'micodebug: micodebug on/off'. Below the list, there is a section titled '====User Commands====' which contains the following commands and their usage: 'eg: wifiparam set -ssid 1111 -key 111111', '# wifiparam', 'Usage: wifiparam get', 'wifiparam set -ssid xxx -key xxx', '# mqtt', 'Usage: mqtt set <MQTT Broker> <MQTT Port> <MQTT User> <MQTT Password>', 'mqtt get', and 'eg: mqtt set wzjinfz.f3322.net 1883 homeassistant 129504'. The status bar at the bottom indicates '就绪', 'Serial: COM10, 115200', '25, 3', '25行, 80列', 'VT100', '大写', and '数字'.

```
memp: print mempool list
wifidriver: show wifi driver status
reboot: reboot MICO system
tftp: tftp
time: system time
ota: system ota
flash: Flash memory map
wifiparam: set wifi ssid and password
mqtt: set mqtt param
uartdebug: uartdebug on/off
tftp_ota: Local tftp OTA
micodebug: micodebug on/off

====User Commands====
eg: wifiparam set -ssid 1111 -key 111111
# wifiparam
Usage: wifiparam get
wifiparam set -ssid xxx -key xxx

# mqtt
Usage: mqtt set <MQTT Broker> <MQTT Port> <MQTT User> <MQTT Password>
mqtt get
eg: mqtt set wzjinfz.f3322.net 1883 homeassistant 129504
```

就绪 Serial: COM10, 115200 25, 3 25行, 80列 VT100 大写 数字

图 1-12

3.9 HA 配置

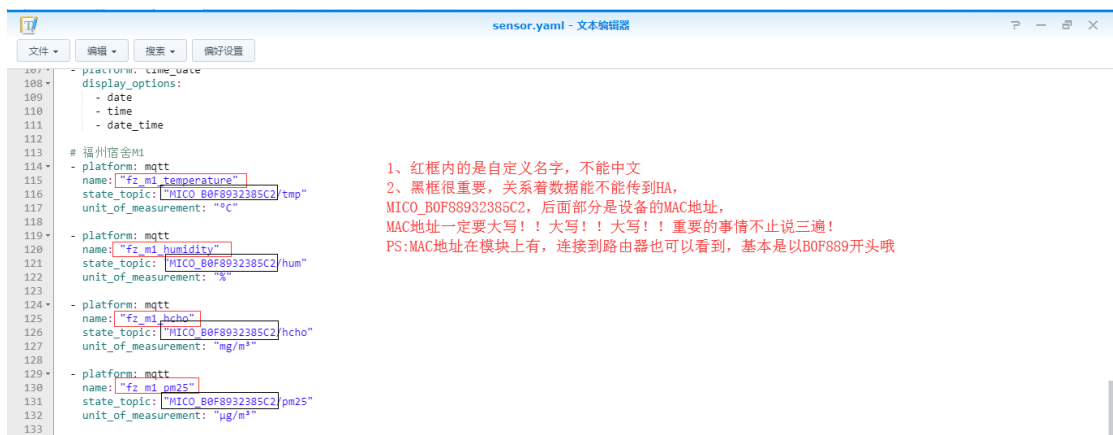


图 1-13

3.10 tftp OTA 升级固件操作

未完待续.....

3.11 http OTA 升级固件

未完待续.....

4. 结束语

首先，感谢自己爱折腾，让自己可以在折腾中不断学习新东西，8月初萌生了开发斐讯M1固件的想法，查阅了相关资料，最后发现我有3GB的内容跟开发固件有关系，开发可以让我有动力以及不眠不休，虽然现在处于管理岗位，还是喜欢利用业余时间开发点东西，让自己保持活性以及更好的思维方式，接着，特别感谢我家领导的支持，毕竟一个人带着2个多月的宝宝，酸甜苦辣不是我们男人都能体会到的，然后，感谢我们翰思彼岸提供了一个平台让我们可以共享经验、知识，最后，也要感谢翰思彼岸的网友们的支持！

对于现在固件存在的一些缺陷，也会慢慢完善，尤其需要完成http OTA功能，这将是广大网友的福音，存在bug只要提交对应的现象，后续都可以自动升级固件了。

未完待续.....