



---

# 量产工具介绍

---

11 FEBRUARY 2019





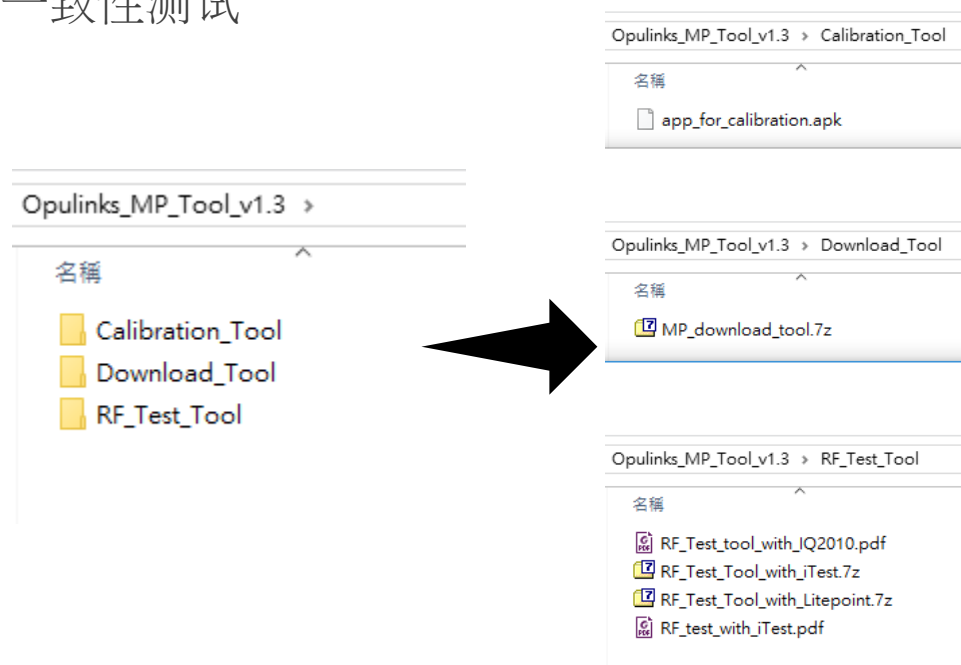
# 大纲

---

- 前言
- 下载工具介绍
  - UART触发的多设备下载方式
- 量产流程介绍
- RF测试工具介绍
  - 使用治具搭配Litepoint仪器的RF测试方式
  - 不使用治具搭配Litepoint仪器的RF测试方式
  - 使用治具搭配iTest仪器的RF测试方式
  - 不使用治具搭配iTest仪器的RF测试方式
- 校准工具介绍

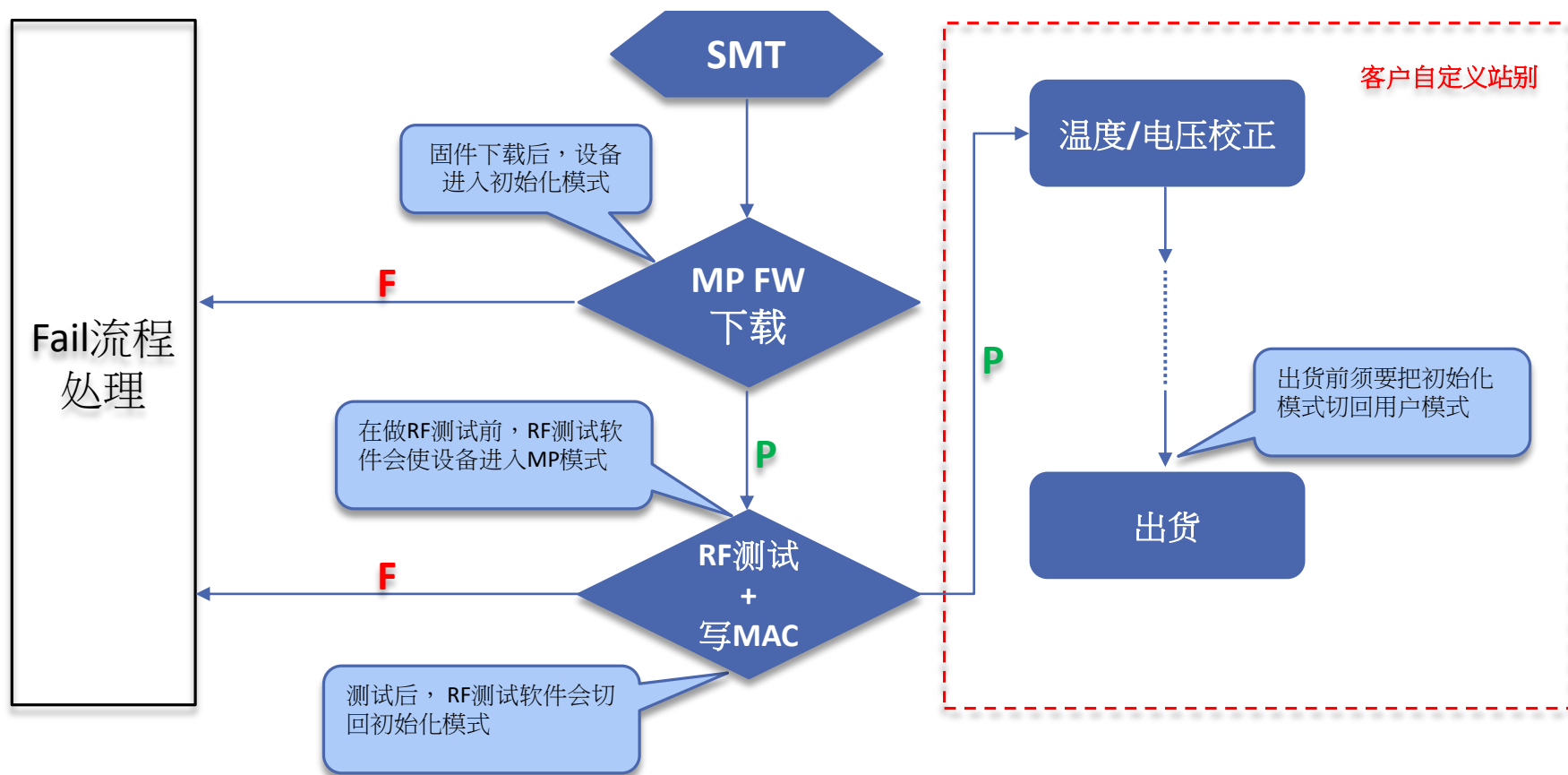
# 前言

- Opulinks提供量产工具来减少生产成本和增加产能
- 量产工具包括
  - 下载工具: 多设备的固件下载和读写BLE和WiFi的MAC地址
  - RF测试工具: BLE和WiFi的RF一致性测试
  - 校准app: 校正电压/温度



# 量产流程介绍

## ● 流程图:





# 量产流程介绍(cont.)

- 模式介绍:
  - 初始化模式: 开启BLE/WiFi功能，而且关闭睡眠功能
  - MP模式: 关闭BLE/WiFi功能
  - 用户模式: 开启BLE/WiFi功能，而且开启睡眠功能
  - 限制情况:
    - 待测物在用户模式中，就**无法下发命令给待测物**，如要下发命令给待测物，需要把待测物切回**初始化模式**或**MP模式**
    - 待测物在MP模式中，就**无法进行BLE/WiFi搜寻和连接**，需要把待测物先切换至**初始化模式**或**用户模式**
- 除固件下载，出货前的各测试站都能依照需求做调整

# 多设备固件下载介绍

- 固件下载可支持波特率最高到921600bps
- 在典型应用中 10 个设备并行下载花费约 10 秒钟
- 工具包提供一个配置文件，用户可以用文本编辑工具根据实际需要和硬件连接方式修改设置下载组合工作模式

模式数值	工作模式
1	仅下载固件
2	仅写入 BLE MAC 地址
3	下载固件并且能写入 BLE MAC 地址
4	仅写入 WIFI MAC 地址
5	下载固件并且写入 WIFI MAC 地址
6	写入 BLE 和 WIFI MAC 地址
7	下载固件 · 写入 BLE 和 WIFI MAC 地址

- 配置文件也定义波特率、固件文件、工作方式、保存 log 文件名、目标设备串口编号和控制通路等信息,让用户自行修改



# Multi-FW Download (cont.)

- Config file (mp\_multi\_download.ini)

```
[GENERAL]
file = opl1000_ota.bin
debug = YES
download_baudrate = 115200
delay_time = 5
trigger = ARDUINO
work_mode = 1
```

```
[BLE_MAC]
begin_addr = 112233445566
end_addr = 11223344557a
curr_addr = 11223344556e
```

```
[WIFI_MAC]
begin_addr = 223344556676
end_addr = 22334455668a
curr_addr = 22334455667e
```

```
[ARDUINO]
control_uart = COM26
control_baudrate = 115200
monitor_uart = COM18
monitor_baudrate = 9600
```

```
[LOG]
log_file = mp_dl_record_3dev.log
log_enable = YES
mac_file = mp_mac_record.csv
```

```
[DEVICES]
number = 2
item1 = COM19, CH0
item2 = COM21, CH1
item3 = COM59, CH2
```

## 下载波特率:

支持115200bps, 230400bps, 460800bps, 921600bps

## 触发模式:

UART: 使用 UART RTS信号触发设备的复位信号  
MANUAL: 手动触发设备的复位信号

配置需要固件下载的设备数目





# Multi-FW Download (cont.)

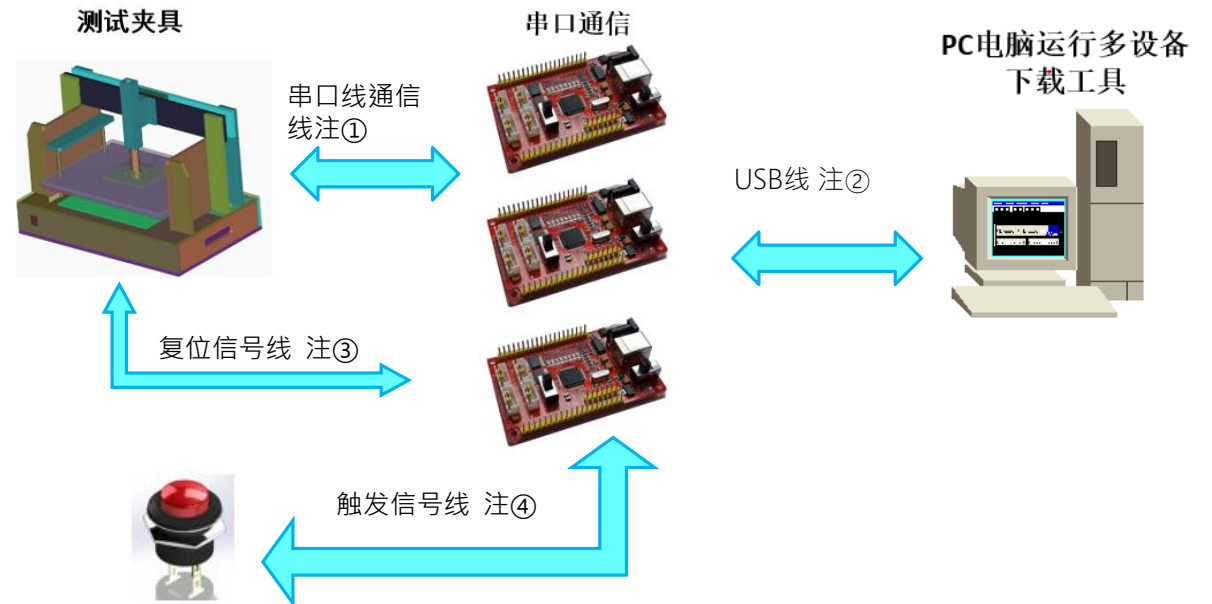
- 使用UART的多设备固件下载的环境架设
  - 设备藉由UART的RTS讯号触发复位
  - UART的CTS讯号主要用于批量生产过程中，板子更换后的开始按钮。

注① 每个设备Device的串口线有三根 TX/RX/GND

注② 推荐使用USB一转四UART转接板。当有N个设备时需要N/4 ( 上取整 ) 个转接板

注③ 每个设备的复位线和对应UART通信口的RTS信号线连接。

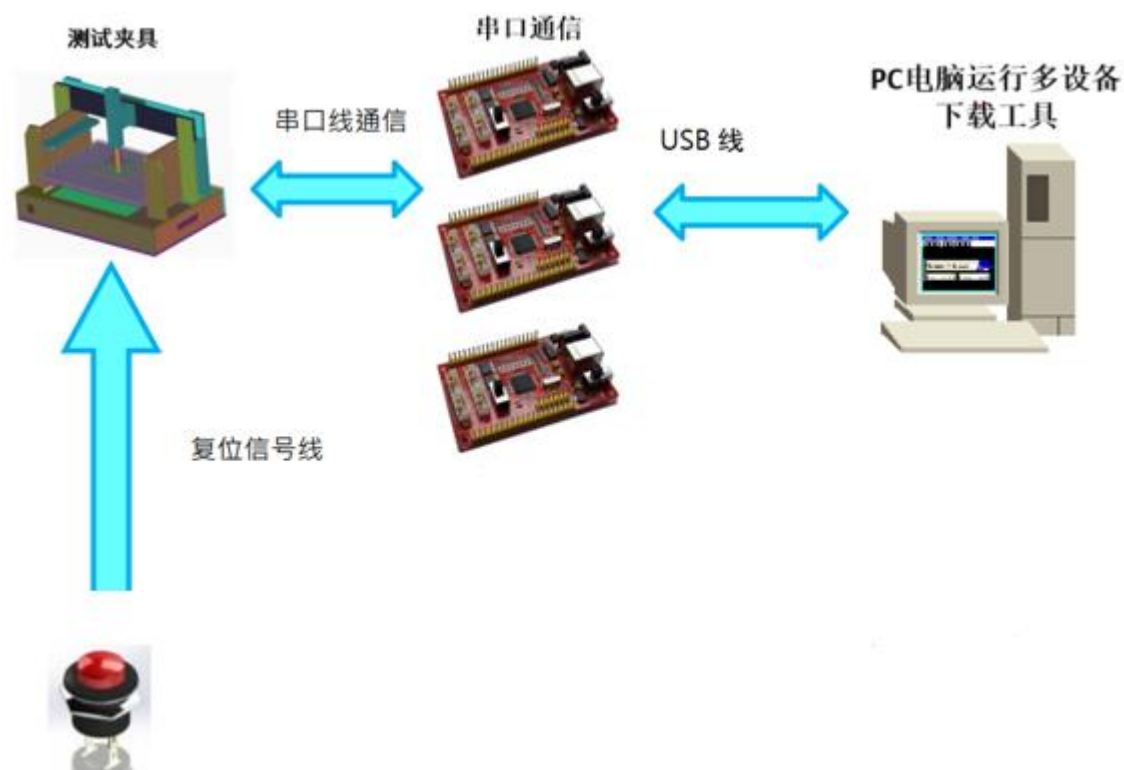
注④ 每个设备的UART通信口CTS线串接在一起和触发按钮连接。当夹具上的设备安装好之后，按下触发按钮，CTS信号在线产生一个低电平脉冲。量产工具监测到此低电平脉冲，开始新一轮的固件下载和MAC地址写入操作。





# Multi-FW Download (cont.)

- 手动复位的多固件下载的环境架设
  - 设备将手动触发复位





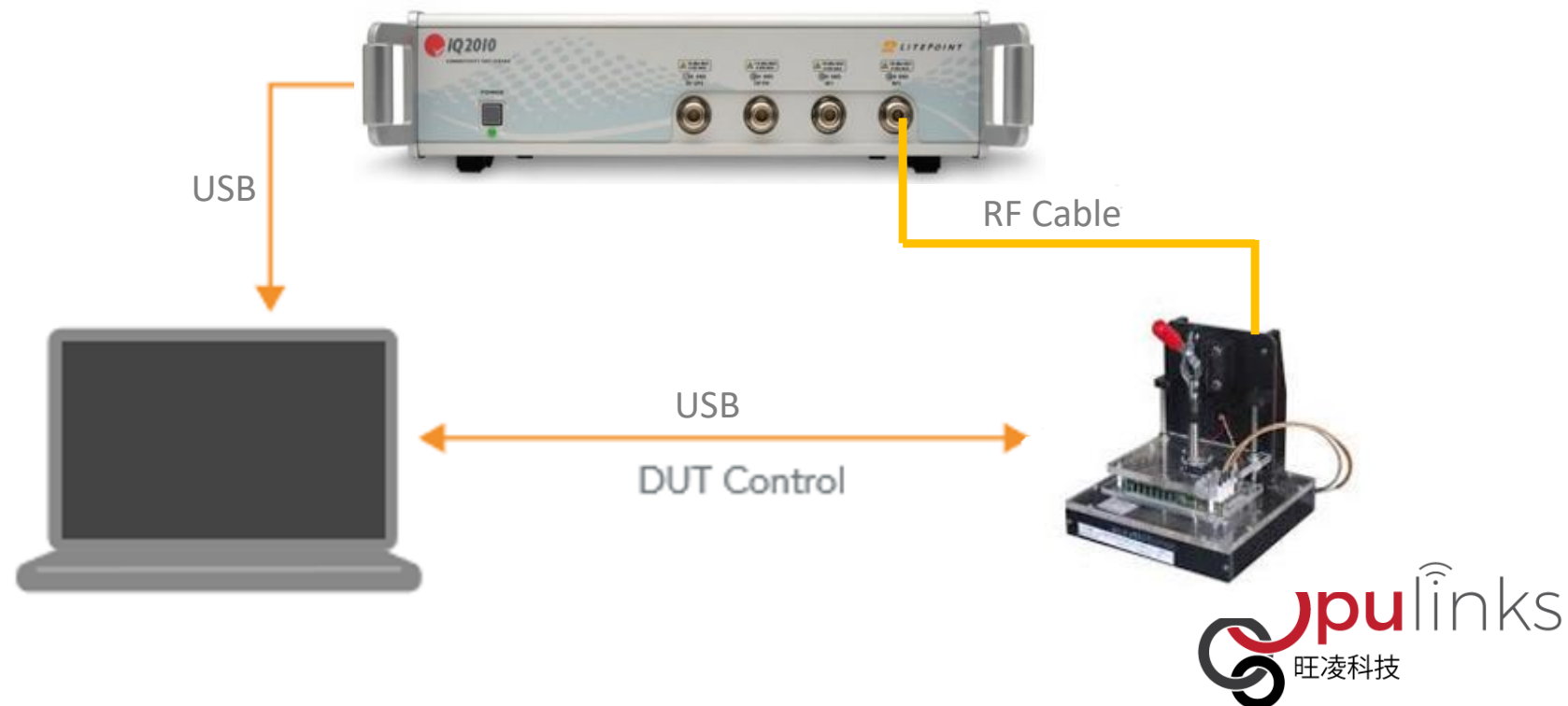
# RF测试工具介绍

---

- OPL RF测试工具目的是做RF的一致性测试
- 在生产过程中，RF测试软件以优化的方式找出生产缺陷
- RF测试软件是一个包含WiFi/BLE的生产需求的测试套件
- RF测试软件也支持WiFi/BLE地址读写

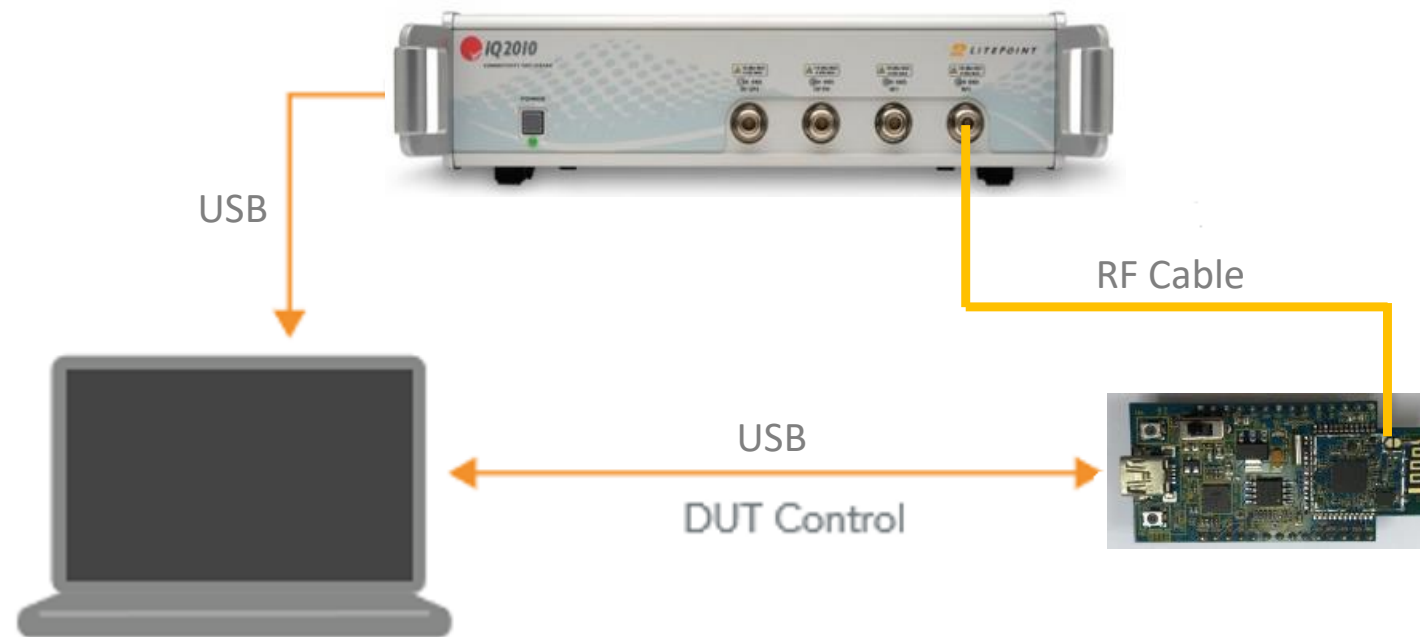
# RF测试工具(cont.)

- Litepoint RF测试环境配置
  - 测试搭配治具
    - RF射频线连接Litepoint IQ2010的RF2
    - PC使用USB分别控制IQ2010和待测物



# RF测试工具(cont.)

- Litepoint RF测试环境配置
  - 测试不搭配治具
    - RF射频线连接Litepoint IQ2010的RF2
    - PC使用USB分别控制IQ2010和待测物

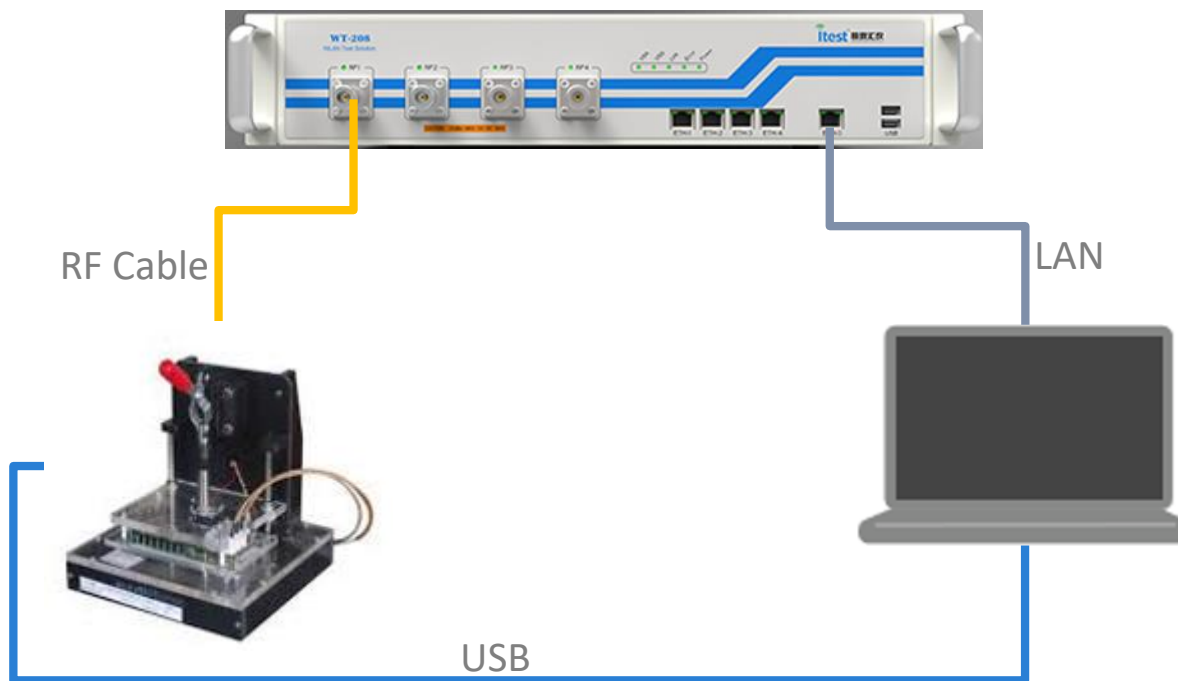


# RF测试工具(cont.)

- iTest RF环境配置

- RF测试搭配治具

- RF射频线连接iTest WT208的RF port
    - PC使用LAN和USB分别控制iTest和待测物

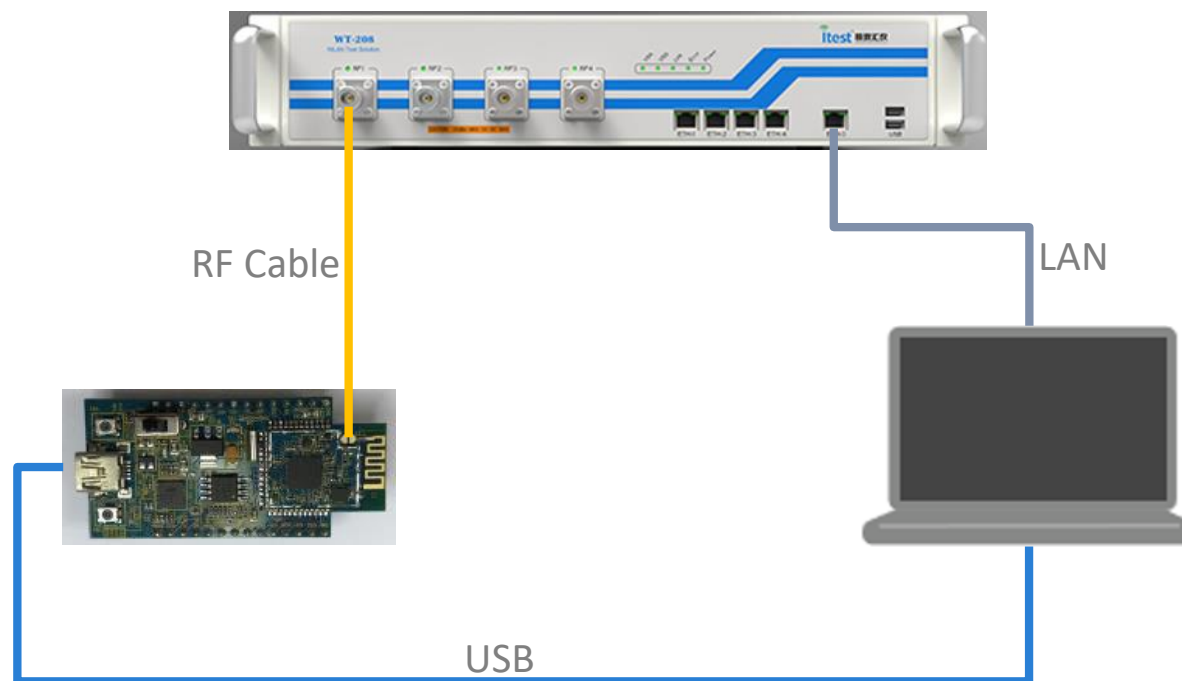


# RF测试工具(cont.)

- iTTest RF环境配置

- RF测试不搭配治具

- RF射频线连接iTTest WT208的RF port
    - PC使用LAN和USB分别控制iTTest和待测物



# RF测试工具(cont.)

- 配置文文件(IQ2010\_RF\_Test.py)
- 用户可以配置测试Channel和测试项

```
##WiFi Tx=====
CH_START = 1
CH_STOP = 13
CH_STEP = 6
rate_start = 3      #// 0:1Mbps, 1:2Mbps, 2:5.5Mbps, 3:11Mbps
rate_stop = 3
rate_step = 1
WiFi_High_Power_Tx_Test_Enable = 1
WiFi_Low_Power_Tx_Test_Enable = 1
##BLE Tx=====
BT_start_freq = 2402
BT_stop_freq = 2480
BT_step = 38
BLE_High_Power_Tx_Test_Enable = 1
BLE_Low_Power_Tx_Test_Enable = 1
##WiFi Rx=====
Wframe_no = 1000
Rx_rate_start = 3    #// 0:1Mbps, 1:2Mbps, 2:5.5Mbps, 3:11Mbps
Rx_rate_stop = 3
Rx_rate_step = 1
Wrfinmax = -15
Wrfinmin = -70
Wrfinstep = 55
CH_List = [1,7,13]
WiFi_Rx_Test_Enable = 1
##BLE Rx=====
Bframe_no = 1000
BR_start_freq = 2402
BR_stop_freq = 2480
BR_step = 38
BR_rate_start = 5    #// 4:LETestRun1a.mod, 5:LETestRun6a, 6:LETestRun10a.mod
BR_rate_stop = 5
BR_rate_step = 1
Brfinmax = -15
Brfinmin = -70
Brfinstep = 55
BLE_Rx_Test_Enable = 1
```

## WiFi Tx配置项:

1. 配置测试channel
2. 配置data rate
3. 开启高功率或低功率测试

## BLE Tx配置项:

1. 配置测试channel
2. 开启高功率或低功率测试

## WIFI Rx配置项:

1. 配置测试channel
2. 配置Rx功率
3. 开启或关闭Rx测试

## BLE Rx配置项:

1. 配置测试channel
2. 配置Rx功率
3. 开启或关闭Rx测试





# RF测试工具(cont.)

- 配置文文件(IQ2010\_RF\_Test.py)
- 上下限设定

```
#=====WiFi Tx Limit=====
wifi_low_target_power=-5.0
wifi_high_target_power=10.0
wifi_Tx_ULimit=[2.0,2.0,2.0,2.0]
wifi_Tx_LLimit=[-2.0,-2.0,-2.0,-2.0]
wifi_EVM_ULimit=[-10.0,-10.0,-10.0,-10.0]
wifi_FreqErr_Limit=20.0
wifi_Clk_ULimit=20.0
wifi_Clk_LLimit=-20.0
wifi_IQamp_ULimit=99.0
wifi_IQamp_LLimit=-99.0
wifi_IQPhase_ULimit=99.0
wifi_IQPhase_LLimit=-99.0
wifi_RmsPhase_ULimit=99.0
wifi_RmsPhase_LLimit=-99.0
#=====
#=====BLE Tx Limit=====
ble_low_target_power=0.0
ble_high_target_power=10.0
ble_Tx_power_ULimit =2
ble_Tx_power_LLimit =-2
ble_Init_Freq_Err_Limit=75
ble_Delta_F1_Avg_ULimit=275
ble_Delta_F1_Avg_LLimit=225
ble_Delta_F2_Avg_ULimit=500
ble_Delta_F2_Avg_LLimit=180
ble_Delta_F2_Max_ULimit=500
ble_Delta_F2_Max_LLimit=185
ble_Fn_Max_Limit=150
ble_Delta_F0Fn_Max_Limit=50
ble_F1F0_Delta_Limit=20
ble_Delta_FnFn_5Max_Limit=20
#=====
```

WiFi Tx 上下限

BLE Tx上下限

# RF测试工具(cont.)

- 配置文文件(IQ2010\_RF\_Test.py)
  - 线损和MAC配置

```
##==path loss=====
ble_path_loss=[0.8, 0.8, 0.8] #depend on num. of test channels
wifi_path_loss=[0.8, 0.8, 0.8] #depend on num. of test channels
#=====MAC Setting=====
#0:Scanner, 1:from file
MAC_Input=1
Write_MAC_Enable=0
#=====
ISN_Length = 6
#=====
```

## ● 报告格式

A	B	C	D	E	F	G	H	I
Channel	I_VBAT	Rate(Mbps)	PeakPWR(dBm)	RMSMaxAvgPWR(dBm)	EVM(pk)(dB)	EVM(all)(db)	Freq Err(Hz)	CLK Err(ppm)
1	0	11	-4.569	-5.631	-18.904	-28.704	17313.423	5.563
7	0	11	-4.03	-5.095	-17.708	-26.905	17342.304	8.205
13	0	11	-4.754	-5.835	-13.787	-24.094	17721.959	4.993

## ● 执行画面

```
Tx Delta F2 Avg = 288.40 kHz ( 180.00, 500.00) -----> PASS
Tx Delta F2 Max = 248.73 kHz ( 185.00, 500.00) -----> PASS
Tx Fm Max      = 16.65 kHz (-150.00, 150.00) -----> PASS
Tx F0Fn Max    = 5.02 kHz (  0.00,  50.00) -----> PASS
Tx F1F0 Delta  = 1.62 kHz ( 20.00,   0.00) -----> PASS
Tx FmFn 5Max   = 4.96 kHz (  0.00,  20.00) -----> PASS
BLE Low Power mode Test complete!!
BLE Low Power mode Tx Test elapsed time:15.0179998875
=> Starting to Write MAC!

Wrt WiFi MAC->8c:88:2b:00:00:12
ret for wrt WiFi MAC:CRC OK: 0000

ret for wrt BLE init:OK

Wrt BLE MAC->8C:88:2B:00:00:12
ret for wrt BLE MAC:OK

Rd WiFi MAC:8c:88:2b:00:00:12
ret for wrt BLE init:OK

Rd BLE MAC:8C:88:2B:00:00:12
#####
##
##          #####      ##          #####      ##
##          ##  ##  ##  ##  ##  ##  ##  ##  ##  ##
##          ##  ##  ##  ##  ##  ##  ##  ##  ##  ##
##          #####      ##  ##          #####      ##
##          ##          #####      ##          ##
##          ##          ##  ##  ##  ##  ##  ##  ##
##          ##          ##  ##  ##  ##  ##  ##  ##
#####
```

# OBWapp介绍

- OBWapp是一个做BLE联机和做温度和电压校准的app
  - 电压校准: 校准电池电压
  - I/O 电压校准: 校准 I/O Pin的电压
  - 温度校准:
    - 关于温度校准是针对一些温度的IoT产品
    - 温度校准时需要把待测物方在温箱中
  - 校准完成后校准结果会自动写入flash

