

产测环境搭建说明

1 环境需要

1.1 硬件

产测一体板,

杜邦线,

电脑,

mini 口 USB 线,

阿莫烧录器（如果需要烧录出厂固件）。

1.2 软件

产测上位机软件,

串口调试助手 sscom 等。

1.3 github 资料介绍

产测工具相关的所有资料都可以从 github 上获取：

https://github.com/munanqiu/ING_RF_Tester/tree/main

github 目录如下图所示：

 01-bin for MCU	golden固件	上传downloader工具
 02-sch-pcb	golden原理/pcb	第一次上传
 03-upper software	上位机软件	更新版本号为1.5.2，增加了透传模式，上位机可以通过evl
 04-doc	使用说明	更新产测环境搭建说明

1.4 名词解释

golden：产测一体板的别称

DUT：被测设备，即需要接受监测的芯片/模块


2 测试过程

从上位机发送测试开始，到 golden 反馈测试完成，中间大致分为：烧写测试固件、射频测试、出厂固件烧写三个过程。

2.1 烧写测试固件

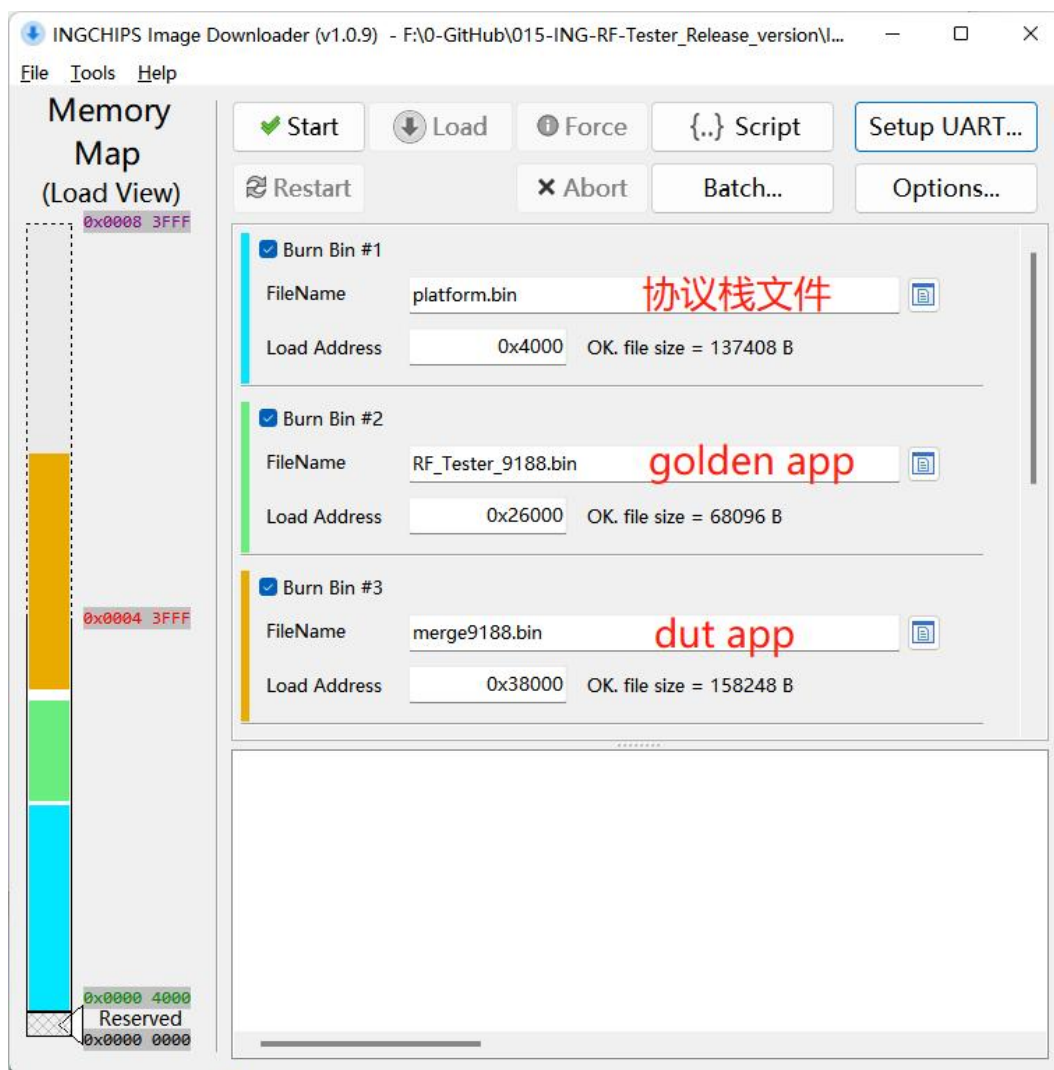
在环境部署阶段，golden 需要烧写三个 bin 文件（此过程可以由原厂工程师辅助完成，也可参照下面步骤操作）。

烧写的固件在 github 上的 `01-bin for MCU` 目录下，目录下的三个文件如下：

[ING_RF_Tester](#) / `01-bin for MCU` / 

 <code>munanqiu</code> 上传downloader工具	
Name	
 ..	
 <code>downloader</code>	烧录工具
 <code>evk-916</code>	量产916系列芯片时的固件
 <code>evk-918</code>	量产918系列芯片时的固件

打开 `downloader` 下的 `icsdw918.exe` 工具，通过“file--open .ini file...”操作导入对应芯片（918 则选择 `evk-918` 文件夹，916 则选择 `evk-916` 文件夹）文件夹下的 `flash_download.ini` 文件，以 916 为例，导入后界面如图：



在 Setup UART 中选择对应的串口和波特率（推荐 460800）后，点击 start 按钮，然后让 golden 进入 load 模式，就可以开始下载。

918 和 916 都需要下载 3 个固件，其中协议栈和 golden app 是 golden 运行的固件，dut app（merge9188.bin）是存储在 golden 的 Flash 里，测试时通过串口烧写到 DUT 里的固件。

因此，如果后续有固件更新，要分清更新的是哪个 bin，进行替换即可（当然也可一起全部更新）。

注意：

更新 golden 固件的时候需要断开 amo 和 dut，保证烧录的串口不被干扰，否则有可能握手失败

2.2 射频测试过程

射频测试过程分 5 个环节：

- TX cnt

DUT 发射一定包数（比如 100 包），golden 接收，接收超过一定阈值认为测试通过。

- TX power

DUT 以 0dbm 发射，golden 接收并检测信号强度，强度超过某个阈值（比如 -35dbm）认为测试通过。

- RX cnt

golden 发射一定包数（比如 100 包），DUT 接收，收到包数超过一定阈值认为测试通过。

- RX power

golden 以 0dbm 发射，DUT 接收并检测信号强度，强度超过某个阈值（比如 -40dbm）认为测试通过。

- Freq offset

DUT 发射信号，golden 测试其频偏，如果发现频偏过大，golden 会通过软件进行测试，并把调试值写入 Flash 固定地址。

当前调试的频偏配置存储在如下 Flash 地址处，可以根据需要调整。

```
Plain Text
#define FLASH_FREQ_ADDR 0x0207F000
```

存在一定可能软件无法将频偏调到要求范围内，就需要通过调整 24M 晶振的电容来调节频偏。（918 不支持软件调节频偏，因此只能通过测试结果看到当前频偏值，如果频偏较大需要调整晶振的匹配电容）

注意：烧写到 DUT 的 flash 中的频偏值为 2 个字节，第一个字节是本身的频偏值，第二个字节则是频偏值本身和 0xFF 异或的值，用来给用户提供频偏值是否被篡改的校验，可以自行决定是否使用该频偏值

2.3 出厂固件烧录

测试完成后，通过控制阿莫烧录器烧录出厂固件，此步骤是可选的：

- 如果选择不烧录出厂固件，则测试完成后会直接上传测试结果。
- 如果选择烧录出厂固件，则烧录完成后再上传测试结果。
- 对于 918，golden 不支持出厂烧录固件

2.4 透传模式

测试完成，并且烧录完成出厂固件之后，用户如果想根据自己烧录程序中已经定好的通讯协议读出 DUT 相关信息，

可以使用此模式来使用上位机直接和 DUT 通讯

- 注意，需要事先设置好上位机和 DUT 通讯的波特率、校验方式之后才能进行通讯
- 退出透传模式无需手动操作，golden 会在执行其他功能的时候自动执行退出

3 通讯协议

上位机与 golden 之间设计了一套串口通讯协议，用来配置测试流程和获取测试结果。
协议里上位机的每一条指令从机都需要进行对应的回复，上位机要有超时重发机制。

3.1 协议格式

一条完整的数据包由帧头、类型、命令长度、命令、crc、帧尾 6 个字段构成，如下图所示：



字段说明：

- 类型：产测一体板内部有多种协议，**类型**字段用于区分不同的协议。对于上位机与 golden 的通讯来说，上位机发送给产测一体板的类型固定为 0x4，产测一体板回复给上位机的类型固定为 0xF4。
- 命令长度：后面的**命令**和**结果**两个字段的长度。
- 命令：具体的命令类型，如测试开始、通讯检测等，详见 3.2。
- crc：计算从帧头到 crc 前所有字段的 crc 值，计算参考代码如下：

```
C++
unsigned short check_crc16(char *puchMsg , unsigned short usDataLen)
{
    unsigned char uchCRCHi = 0xFF ; /* high byte of CRC
```

```

initialized */
    unsigned char uchCRCLo = 0xFF ; /* low byte of CRC initialized
*/
    unsigned char uIndex ;          /* will index into CRC lookup
table */

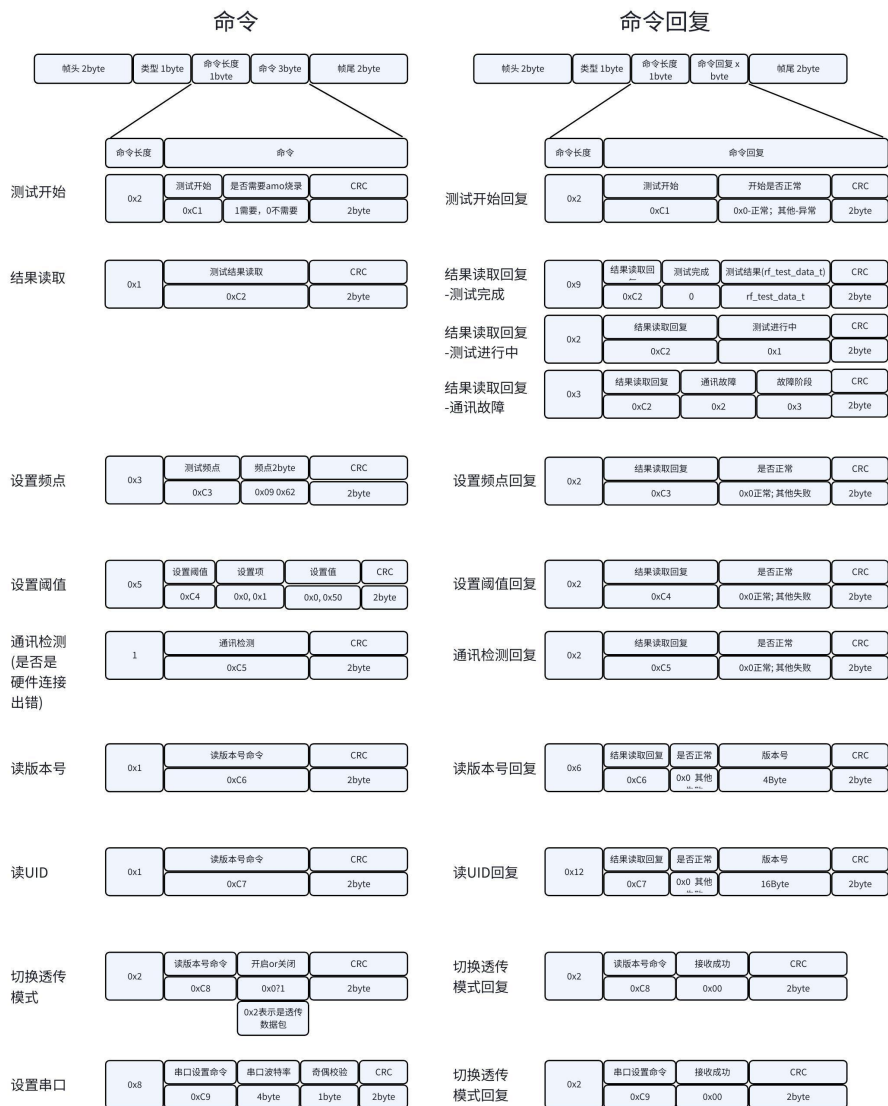
    while (usDataLen--) /* pass through message buffer */
    {
        uIndex = uchCRCHi ^ *puchMsg++ ; /* calculate the CRC */
        uchCRCHi = uchCRCLo ^ auchCRCHi[uIndex];
        uchCRCLo = auchCRCLo[uIndex] ;
    }

    return (uchCRCHi << 8 | uchCRCLo) ;
}

```

3.2 不同的命令

目前实现的命令如下：



3.2.1 测试开始-C1

- 命令

表示上位机开启一轮完整测试，有一个参数：是否需要 amo，长度 1 个字节。

1：带 amo 烧录，测试完成之后使用 amo 烧录出厂固件，需要连接阿莫烧录器。
(918 不支持 amo 烧录，在使用中禁用)

0：不带 amo 烧录，测试完成之后整个流程直接结束。

- 回复

表示 golden 收到了上位机的开始测试命令，有一个参数：是否正常开始。

0：正常地开启了测试。

其他：开启测试失败。

3.2.2 结果读取-C2

- 命令

主动读取测试结果，无参数。

- 回复

回复有三种情况，会有不同的参数。

1) 测试完成，会有两个参数：测试完成（0x0）和测试结果。

测试结果使用下面的数据结构存储，前面 5 个 byte 分别表示测试的各项结果，第 6 个 byte 的 5 个 bit 表示各项测试是否通过。

```
C++
typedef struct {
    uint8_t tx_cnt;
    uint8_t rx_cnt;
    int8_t tx_power;
    int8_t rx_sen;
    int8_t freq_offset;
    struct {
        bool tx_cnt_pass      : 1;  // 1-bit flag for pass/fail
        bool rx_cnt_pass      : 1;
        bool tx_power_pass    : 1;
        bool rx_sen_pass      : 1;
        bool freq_offset_pass : 1;
    } status;
} rf_test_data_t;
```

2) 测试进行中，有一个参数：测试进行中（0x1）。

3) 通讯失败，有两个参数：测试失败（0x2）和失败阶段。

失败阶段通过下列数据结构定义，可以用来定位出问题的位置。

这里的失败通常是硬件线路问题，如串口线松了，供电问题等。如果是发生在 0-5，则 DUT 和 golden 的连接或者 DUT 的供电出了问题。如果发生在 6 阶段（阿莫烧写），则是与阿莫相关的连接、供电出了问题。

```
Bash
typedef enum {
    COMM_ERR_STAGE_BURNING_TEST      = 0,    //burn test image
    COMM_ERR_STAGE_TX_CNT_TEST       = 1,    //tx count test
    COMM_ERR_STAGE_TX_POWER_TEST     = 2,    //tx power test
    COMM_ERR_STAGE_RX_CNT_TEST       = 3,    //rx count test
    COMM_ERR_STAGE_RX_SEN_TEST       = 4,    //rx sensitivity
```



```

test
    COMM_ERR_STAGE_FREQ_OFFSET_TEST    = 5,      //frequency offset
test
    COMM_ERR_STAGE_BURNING_VENDOR      = 6,      //burn vendor test
image
} RF_comm_err_stage_t;

```

- 说明

golden 完成整个测试过程后会主动发送测试结果，主机也可以发送结果读取，但是如果测试还没有完成，就需要过一段时间再次读取。

3.2.3 设置频点-C3

- 命令

设置射频测试使用的频点。有一个参数，两个字节，表示频点，范围 2402-2480。

如果上位机进行了设置，就使用配置的频点测试，如果没有调用此命令进行设置，则会使用随机生成的 2402-2480 内的一个频点进行测试。

- 回复

回复设置结果，有一个参数：0-设置成功；其他-设置失败。

- 说明：

产线内有多个测试工位同时执行测试时，则要对使用的频点进行规划，如果两个临近工位同时使用相同频点测试，会互相干扰，导致两边的测试数据同时变差。

如果同一时间只有一个工位进行测试，则要避免环境中干扰比较大的频点。

如果有条件，将 golden 和 DUT 屏蔽在一个空间内可以获得最好的测试效果。

3.2.4 设置阈值-C4

- 命令

设置发送总包数（tx cnt 和 rx cnt 阶段使用）和测试的阈值。有两个参数：

设置项，长两个字节。有 0x1-0x5 五个类别，分别表示总包数，tx_power 阈值，tx_cnt 阈值，rx_sen 阈值，rx_cnt 阈值。

```

Plain Text
typedef enum : uint8_t
{
    PACKET_CNT          = 1,
    TX_POWER             = 2,

```

```
TX_CNT          = 3,  
RX_SEN          = 4,  
RX_CNT          = 5  
}threshold_switch;
```

设置值，实际配制的值，有两个字节。比如总包数设置 100，tx_power 设置-35。

- 回复

回复设置结果：0-设置成功；其他-设置失败。

- 说明

1) TX_CNT 和 R_CNT 的阈值表示收到的包数，收到一定包数以上测试合格，比如设置阈值 80，则 80 包以上合格。

2) TX_POWER 和 RX_SEN 都是 rssi，单位是 dbm，收到的信号强度强于阈值测试合格。

3) PACKET_CNT 是测试使用的总包数，建议设成 100，设置过大会增加测试时间，且需要对应地调整阈值。

3.2.5 通讯检测-C5

- 命令

检测通讯是否正常，无参数。

- 回复

回复通讯检测的结果，带 1 个参数：0-正常；其他-不正常

- 说明：当前版本通讯检测只能检测上位机与 golden 通讯是否正常，无法检测 golden 和 DUT 是否连接正常，以及 golden 和 amo 是否连接正常，后续版本会进行优化。

3.2.5 读版本号-C6

- 命令

读取当前 golden 固件的版本号，不带参数。

- 回复

回复读取到的版本号，有两个参数：

是否正常：0-正常；其他-异常

版本号：golden 固件版本号的 ASCII 码，版本号的格式为 V1.0.0。

3.2.6 读取 UID-C7

- 命令

读取当前测试的 DUT 的 UID，不带参数

- 回复

回复读取到的当前 DUT 的 16byte 的 UID，有两个参数：

是否正常：0-正常；其他-异常

UID：DUT 的 UID，长度 16 字节，HEX 形式

3.2.7 透传模式-C8

- 命令

进入透传模式，evk 仅作数据中转，上位机可与 dut 直接交互。

- 回复

回复设置结果：0-设置成功；其他-设置失败。

说明：此模式用于阿莫烧录出厂固件之后，需要出厂固件支持透传模式。

3.2.8 设置透传模式下 evk 与 dut 交互串口参数-C9

- 命令

更改 evk 和 dut 交互串口的参数，包括波特率以及校验方式

- 回复

回复设置结果：0-设置成功；其他-设置失败。

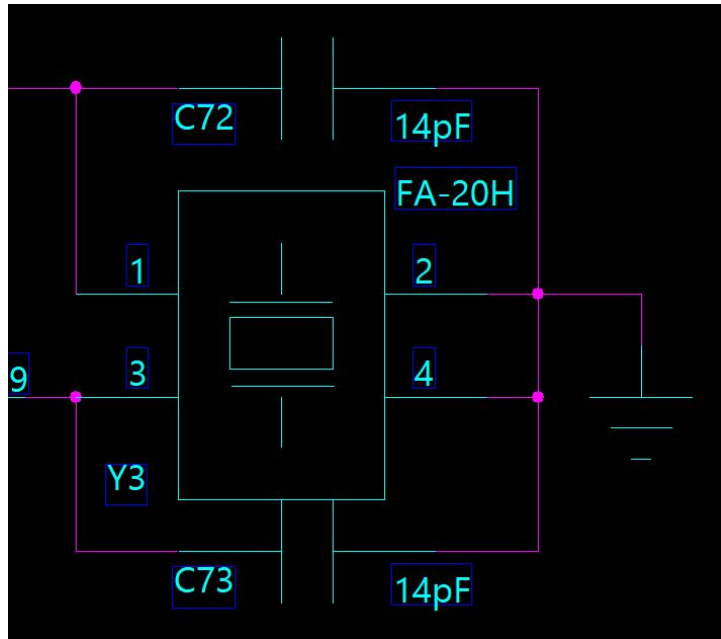
4 环境搭建

4.1 频偏测试

射频测试的最后一个过程——频偏测试过程，golden 通过接收 DUT 的信号来测 DUT 的频偏，需要 golden 本身频偏正常，因此每一块 golden 的频偏都需要单独测试并调整。

测试过程为通过烧录 SDK 自带的 BQB RF Test App，在频谱仪上查看频偏，并通过调整匹配电容（下图中的 C72，C73）来进行调整。

每一块 golden 的频偏都要调整到 0 附近。



此过程可由原厂协助完成，细节不再赘述。

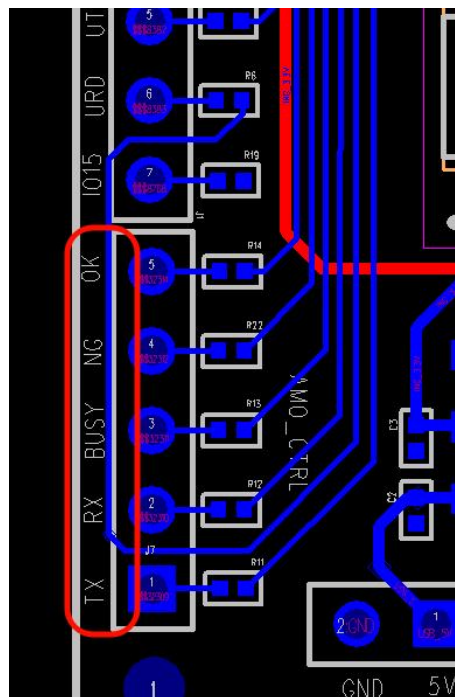
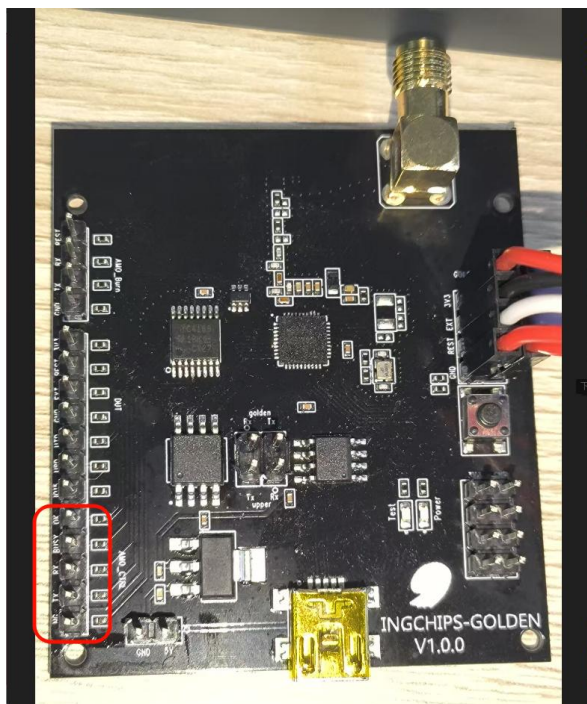
产测一体板的频偏每年都要进行一次测试与校准。

4.2 连线

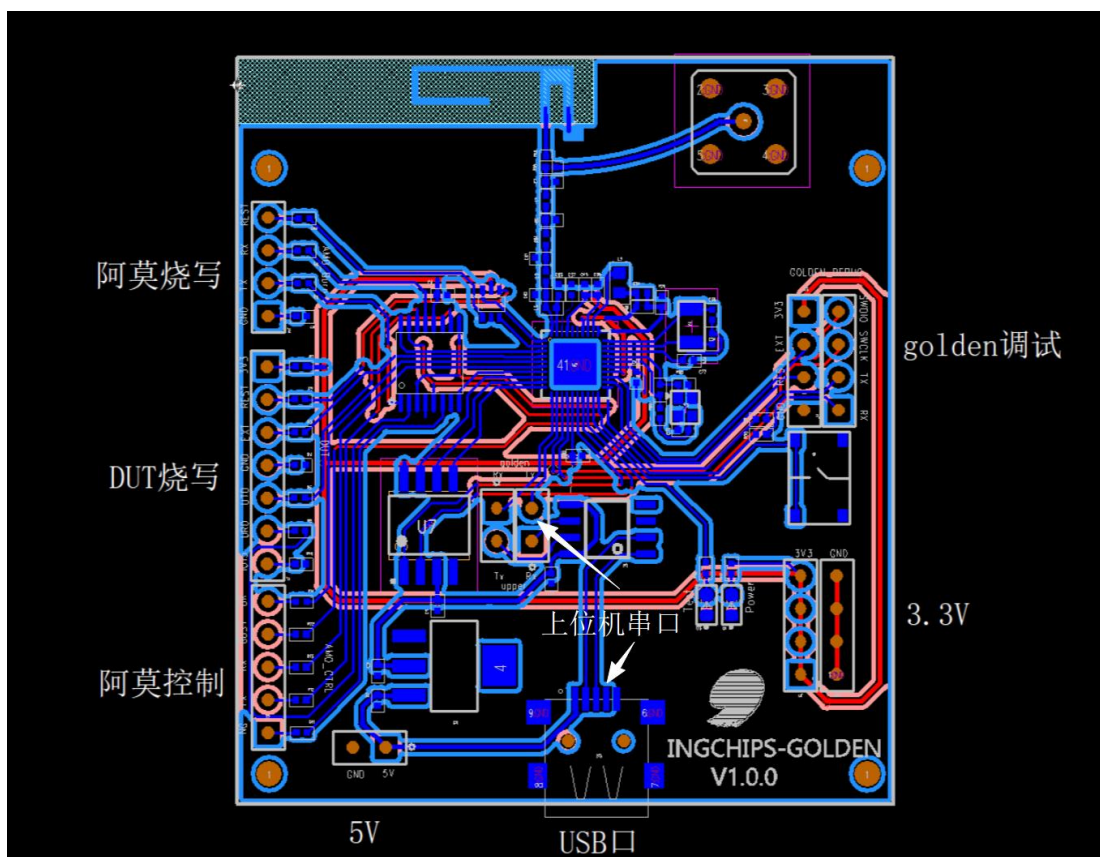
如下图所示，必须连接的包括：

- 阿莫烧写：连接阿莫烧录器的烧写引脚，包 RST，RX，TX，GND，共 4 根线，无出厂固件烧录则不用连。
- DUT 烧写：连接 DUT 的烧写引脚，包括 3V3，RST，EXT，GND，TX，RX，共 6 跟线
- 阿莫控制：连接阿莫烧录器的控制引脚，包括 OK，NG，BUSY，TX，RX，共 5 根线

注意：此次硬件丝印有误，下一版会改为正确(右侧为正确的 amo 控制引脚连接顺序)



- 上位机串口：连接上位机，则可以直接连接 USB 口（此时两个端子需要插上跳线），如果连接其他 MCU，则去掉跳线，直接接上面的 TX 与 RX 两根线。



其他非必要引脚包括：

- golden 调试：用于调试阶段进行烧录及 log 打印，或者更新 golden 的固件，正式

量产时不需要。

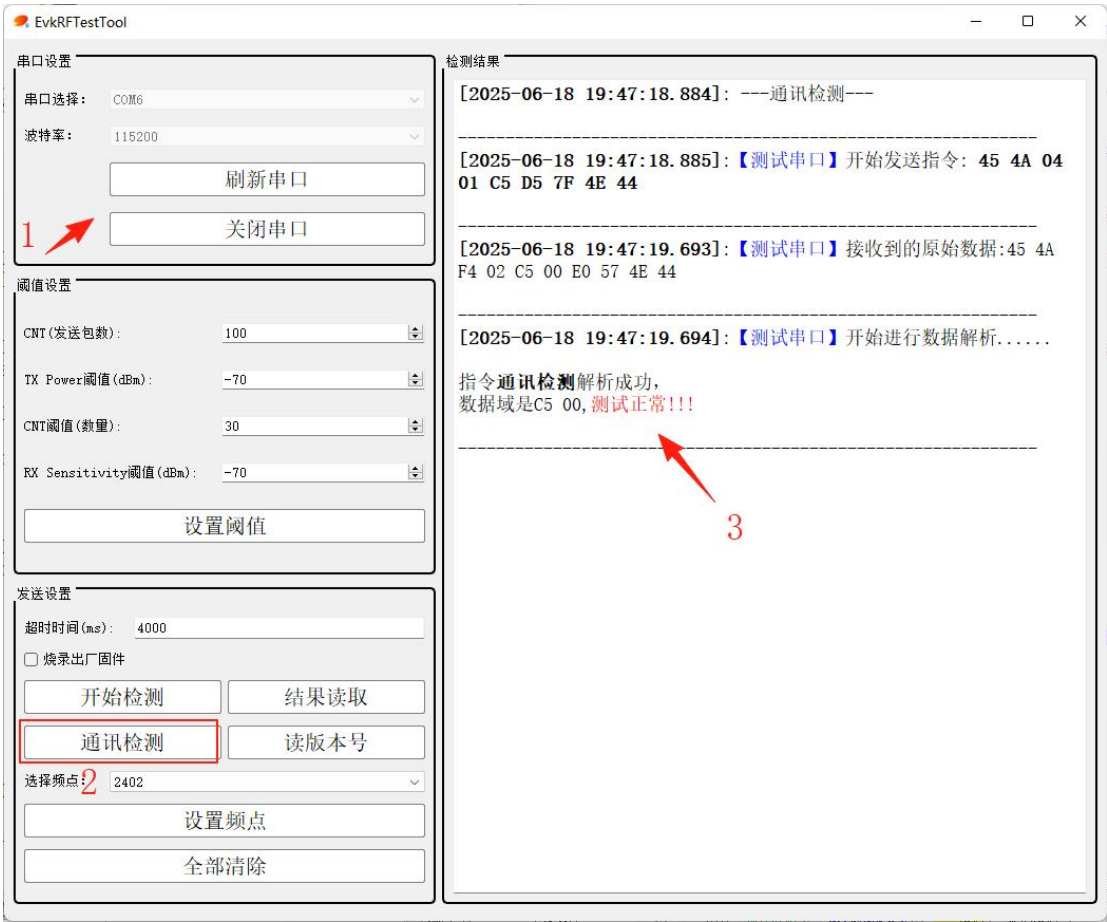
- 3.3V：一组 3.3V 电源和地，调试用。
- 5V：可以通过外部电源给线路板供电。

4.3 连接检测

将所有线路连接好并上电后，可以通过上位机检测系统是否工作正常。

此时通过 USB 接口连接上位机，首先刷新并连接串口，然后点击“**通讯检测**”按钮，可以在右侧窗口查看检测结果。

下图所示表示连接正常，表示 golden 已供电正常，golden 与上位机串口连接正常。



目前软件还未提供其他接口的连接检测，暂时需要人工检测，后续版本会增加。

4.4 阈值调试

实际搭建产测环境时，需要根据场地的实际情况来确定测试的各项阈值，包括：

- **cnt 包数**：此次测试发送的包数基准，范围是 1~255 包，以 100 为例说明。
- **TX cnt**：DUT 发送 100 包，golden 接收，收到的包数高于某个阈值（比如

80)，认为合格。

- **TX power:** DUT 发送，golden 接收，如果接收到的信号强度**强于**某个阈值（比如-30dbm），认为合格。
- **RX cnt:** golden 发送 100 包，DUT 接收，收到的包数**高于**某个阈值（比如 80），认为合格。
- **RX sen:** golden 以 0dbm 发送，DUT 接收，接收到的信号强度**超过**某个阈值（比如-35dbm），认为合格。

四项参数中 TX cnt 和 TX power 为对射频发射部分的评估，RX cnt 和 RX sen 为对射频接收部分的评估。

测试环境不同，正常设备测试的数据会有不同，需要根据实际情况设定合适的阈值，以实现既不会把正常设备判定成损坏，也可以把损坏的设备检测出来。

5 协议数据举例

5.1 开始检测

```
Plain Text
//上位机
45 4A 04 02 C1 00 20 66 4E 44
//golden 回复
45 4A F4 02 C1 00 20 55 4E 44
```

5.2 结果读取

```
Plain Text
//上位机
45 4A 04 01 C2 17 3E 4E 44
//golden 回复-测试完成以及测试结果
45 4A F4 08 C2 00 0A 1E E2 BA FD 17 DC 0C 4E 44
//golden 回复-测试进行中
45 4A F4 02 C2 01 10 94 4E 44
//golden 回复-通讯故障，发生在 tx cnt 测试阶段
45 4A F4 03 C2 02 02 62 91 4E 44
```

5.3 设置频点

Plain Text

//上位机

45 4A 04 03 C3 62 09 70 E9 4E 44

//golden 回复

45 4A F4 02 C3 00 40 54 4E 44

5.4 设置阈值

按下设置阈值按钮后，会依次进行测试包数、tx power，tx cnt，rx sen，rx cnt 五个阈值的设置。

Plain Text

//上位机设置测试包数

45 4A 04 05 C4 01 00 64 00 54 1F 4E 44

//golden 回复

45 4A F4 02 C4 00 70 56 4E 44

//上位机设置 tx power 的阈值

45 4A 04 05 C4 02 00 BA FF F0 06 4E 44

//golden 回复

45 4A F4 02 C4 00 70 56 4E 44

//上位机设置 tx cnt 的阈值

45 4A 04 05 C4 03 00 1E 00 8C 3D 4E 44

//golden 回复

45 4A F4 02 C4 00 70 56 4E 44

//上位机设置 rx sen（灵敏度）的阈值

45 4A 04 05 C4 04 00 BA FF 78 06 4E 44

//golden 回复

45 4A F4 02 C4 00 70 56 4E 44

//上位机设置 rx cnt 的阈值

45 4A 04 05 C4 05 00 50 00 A4 08 4E 44

//golden 回复

45 4A F4 02 C4 00 70 56 4E 44

5.5 通讯检测

Plain Text

//上位机

45 4A 04 01 C5 D5 7F 4E 44

//golden 回复

45 4A F4 02 C5 00 E0 57 4E 44

todo..需要补充测试失败和测试成功的数据

5.6 读版本号

```
Plain Text
//上位机
45 4A 04 01 C6 D4 3F 4E 44
//golden 回复
45 4A F4 06 C6 00 00 00 00 01 B1 95 4E 44
```

5.7 读取 UID

```
Plain Text
//上位机
45 4A 04 01 C7 14 FE 4E 44
//golden 回复
45 4A F4 12 C7 16ByteUID CRC CRC 4E 44
```

5.8 设置透传模式

```
Plain Text
//上位机
45 4A 04 02 C8 01 B0 A1 4E 44
//golden 回复
45 4A 04 02 C8 00 70 60 4E 44
```

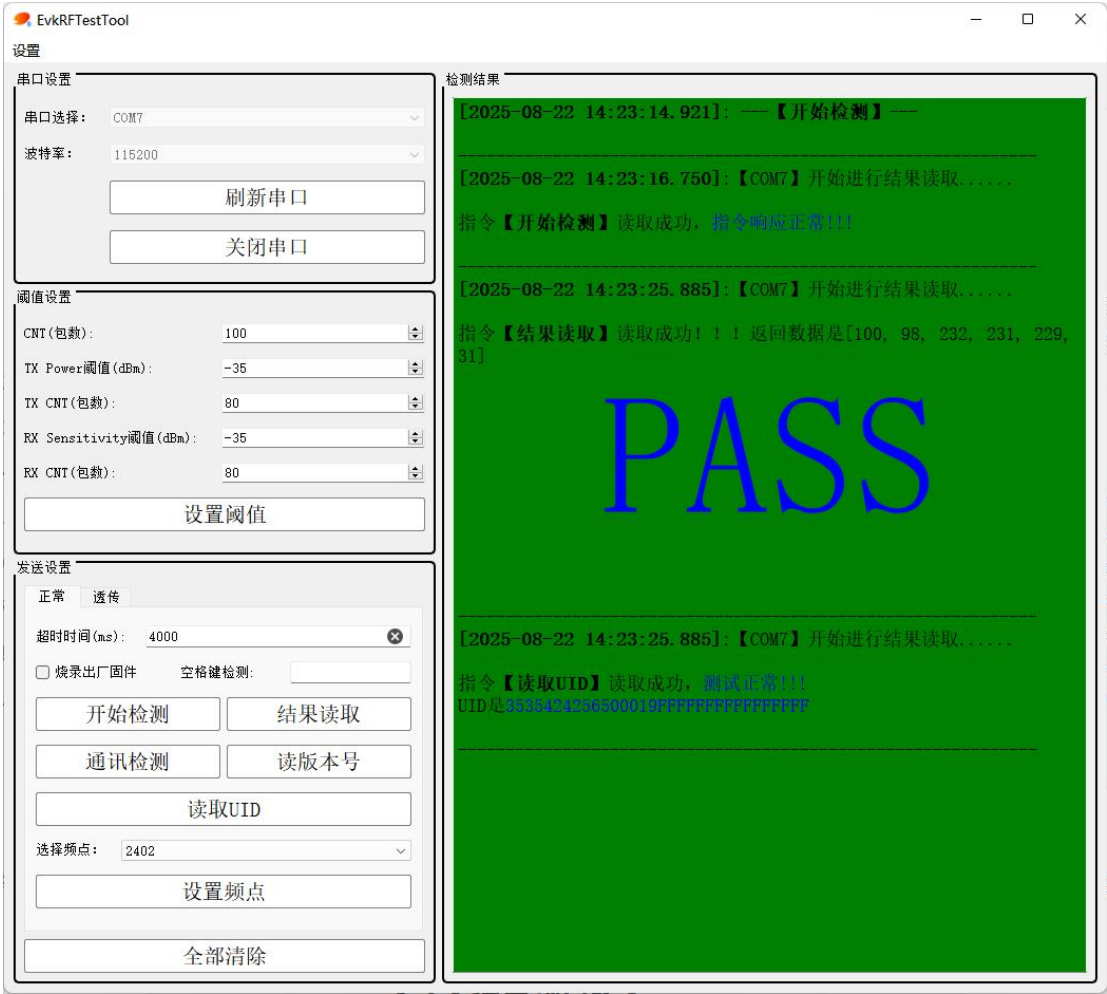
5.9 设置 evk 和 dut 通讯串口参数

```
Plain Text
//上位机
45 4A 04 06 C9 80 25 00 00 00 18 51 4E 44
//golde 回复
45 4A F4 02 C9 00 E0 52 4E 44
```

6 上位机软件

上位机界面如下图所示，分别包括：串口设置、阈值设置、发送设置和检测结果 4 个

部分，其中发送设置部分分为正常检测和透传功能两个选项卡。



6.1 串口

刷新串口可以枚举当前电脑连接的虚拟串口，选择一体板对应的串口号连接。

波特率当前使用的 115200。

6.2 设置阈值

当前设置阈值分为五条指令分别设置，交互时间需要 10s 左右。

对于同一批产品在同样的环境中进行测试，每次上电后只要配置一次即可，配置会保存在产测一体板内。

但是如果产测一体板断电后重新上电，则需要重新配置。

6.3 发送设置

发送设置下有正常（正常测试）和透传两个选项卡。

6.3.1 正常测试

发送设置

正常 透传

超时时间(ms): 4000

☐ 烧录出厂固件 空格键检测:

开始检测 结果读取

通讯检测 读版本号

读取UID

选择频点: 2402

设置频点

全部清除

- 超时时间：上位机和 golden 通讯的超时时间，即 golden 如果在多长时间内没有回复，上位机认为本次通讯失败。
- 烧录出厂固件（复选框）：如果连接了阿莫烧录器用于出厂固件烧录，则勾选此复选框，否则不要选。
- 空格键检测：将鼠标放在后面的输入框内，之后点击空格就会开启检测，和按下“开始检测”按钮效果相同，目的是方便产线操作。
- 结果读取：主动读取上一次的检测结果。
- 通讯检测：检测上位机与 golden 是否通讯正常。
- 读版本号：读取 golden 固件的版本号。
- 读取 UID：读取 dut 芯片的 UID。
- 设置频点：设置当前检测使用的频点，如果多个工位同时测试，使用同一个频点很可能形成干扰，要选择不同的频点。

6.3.2 透传

发送设置

正常

透传

波特率：

9600

▼

校验位：

N

▼

DUT数据：

☐ HEX发送

发送配置

注：发送DUT数据之前先发送配置才生效!!!

发送数据

全部清除

透传模式时上位机和 dut 直接通讯，golden 充当透传的通道。

透传模式的存在，是为了上位机和 dut 的出厂固件进行交互，比如读取出厂固件的版本号等。因此想要使用透传模式，需要满足前提条件：DUT 烧写了出厂固件，且出厂固件支持透传相关协议。

- 发送配置：使用透传模式前需要先将与 DUT 连接的串口进行配置，配置项包括波特率和校验选择两个。
- 发送数据：在 DUT 数据框内输入想要透传给 DUT 的数据，点击发送数据即可。HEX 发送复选框可以选择数据的格式，为 ASCII 还是 HEX 格式。

7 重要说明

7.1 频偏校准

硬件的频偏可能会随着时间推移发生偏移，因此 4.1 提到的频偏测试与校准，最好每年进行一次，否则可能导致产品的系统性频偏问题，如某些手机连接不上，容易断开等。

版本记录

文件版本	发布时间	说明
V1.0.0	2025/6/20	第一版发布
V1.1.0	2025/7/2	1) 增加了第 6 部分重要说明 2) 增加了 uid 读取协议 3) 增加了 2.1 烧写测试固件的注意事项
v1.2.0	2025/7/5	增加了 UID 读取的流程
v1.3.1	2025/7/7	将频偏值烧录到 DUT 放到了 vendor 之后
v1.4.1	2025/6/9	取消烧录出厂固件之后再烧录频偏修改值，变为原始在测试完成之后就烧录频偏修改建议值
v1.4.2	2025/7/17	增加了接入检测的超时流程，让接入检测更快
v1.5.2	2025/8/6	增加了 DUT 和上位机之间直接交互的透传模式
V1.6.1	2025/8/22	增加了 918 的适配，增加了上位机的说明