网络透传模块测试流程

整个测试分为以下几类，涵盖了uart,usb,bus这三种接口的ipv4,ipv6各种组合测试，三种接口的测试流程按以下顺序进行

1 被测开发板ipv4,udp测试

2 被测开发板ipv4,tcp,client测试

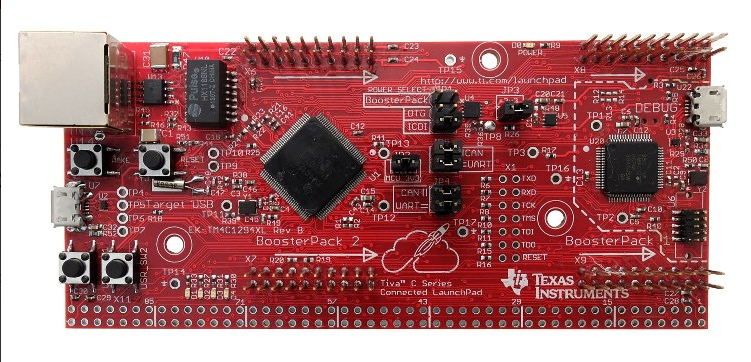
3 被测开发板ipv4,tcp,server测试

4 被测开发板ipv6,tcp,server测试

5 被测开发板ipv6,tcp,client测试

6 被测开发板ipv6,udp测试

开发板启动配置说明



按住上图的左下键上电或者复位（复位脚为右上脚），为串口模式

按住上图的右下键上电或者复位（复位脚为右上脚），为总线模式

什么都不按，上电或者复位，为usb模式

同时按住上图的左下键和右下键，上电或者复位，为总线测试模式的主控模式

在测试总线模式时，注意让主控开发板先上电，被测开发板后上电

测试流程：

1 被测开发板ipv4,udp测试 **串口模式**

将config脚（上图的电容c23向左数第二个脚）和他下面的脚短接，然后在pc a侧执行

Python config.py com5 read 38 8

读回网络连接状态

在pc b侧执行

Python ipv4.py 192.168.1.100 1234 u4 ipv4\_udp\_serial.zip

将config脚（上图的电容c23向左数第二个脚）和他右边的脚短接，然后在pc a侧执行

Python ser.py com5 xxx.zip

观察pc b侧是否打印接收到的字节数，当pc a侧发送完毕后，会打印发送字节数和消耗时间。注意以上com5为开发板的debug口连接pc后所显示的串口号 xxx.zip为你发送的文件名

以下省略文字部分解释，只按顺序给出执行命令

2被测开发板ipv4,tcp,client测试

设置开发板为串口配置模式（见上）

Python config.py com5 tcp 0 1

在pc b侧执行

Python ipv4.py 192.168.1.100 1234 s4 ipv4\_tcp\_server\_serial.zip

在pc a侧执行

Python config.py com5 read 38 8

判断返回是否为26 02，是的话设置开发板为串口透传模式（见上）

在pc a侧执行

Python ser.py com5 xxx.zip

3被测开发板ipv4,tcp,server测试

设置开发板为串口配置模式（见上）

Python config.py com5 mode 0 1

在pc b侧执行

Python ipv4.py 192.168.1.103 1234 c4 ipv4\_tcp\_client\_serial.zip

在pc a侧执行

Python config.py com5 read 38 8

判断返回是否为26 02，是的话设置开发板为串口透传模式（见上）

在pc a侧执行

Python ser.py com5 xxx.zip

4被测开发板ipv6,tcp,server测试

设置开发板为串口配置模式（见上）

Python config.py com5 protol 0 1

在pc b侧执行

Python ipv6.py fe80::1 1234 c6 ipv6\_tcp\_client\_serial.zip

在pc a侧执行

Python config.py com5 read 38 8

判断返回是否为26 02，是的话设置开发板为串口透传模式（见上）

在pc a侧执行

Python ser.py com5 xxx.zip

5 被测开发板ipv6,tcp,client测试

设置开发板为串口配置模式（见上）

Python config.py com5 mode 0 0

在pc b侧执行

Python ipv6.py pcb的ipv6地址 s6 ipv6\_tcp\_server\_serial.zip

在pc a侧执行

Python config.py com5 ripv6 0 pcb的ipv6地址

Python config.py com5 read 38 8

判断返回是否为26 02，是的话设置开发板为串口透传模式（见上）

在pc a侧执行

Python ser.py com5 xxx.zip

6 被测开发板ipv6,udp测试

设置开发板为串口配置模式（见上）

Python config.py com5 tcp 0 0

在pc b侧执行

Python ipv6.py pcb的ipv6地址 u6 ipv6\_udp\_serial.zip

在pc a侧执行

Python config.py com5 read 38 8

判断返回是否为26 02，是的话设置开发板为串口透传模式（见上）

在pc a侧执行

Python ser.py com5 xxx.zip

1 被测开发板ipv4,udp测试 **usb模式**

Usbpc.exe read 38

读回网络连接状态

在pc b侧执行

Python ipv4.py 192.168.1.100 1234 u4 ipv4\_udp\_usb.pdf

在pc a侧执行

Usb-w.exe 1 xxx.pdf

观察pc b侧是否打印接收到的字节数，当pc a侧发送完毕后，会打印发送字节数和消耗时间。注意以上xxx.pdf为你发送的文件名

以下省略文字部分解释，只按顺序给出执行命令

2被测开发板ipv4,tcp,client测试

Usbpc.exe tcp 0 1

在pc b侧执行

Python ipv4.py 192.168.1.100 1234 s4 ipv4\_tcp\_server\_usb.pdf

在pc a侧执行

Usbpc.exe read 38

判断返回是否为26 02，是的话

在pc a侧执行

Usb-w.exe 1 xxx.pdf

3被测开发板ipv4,tcp,server测试

Usbpc.exe mode 0 1

在pc b侧执行

Python ipv4.py 192.168.1.103 1234 c4 ipv4\_tcp\_client\_usb.pdf

在pc a侧执行

Usbpc.exe read 38

判断返回是否为26 02，是的话

在pc a侧执行

Usb-w.exe 1 xxx.pdf

4被测开发板ipv6,tcp,server测试

Usbpc.exe protol 0 1

在pc b侧执行

Python ipv6.py fe80::1 1234 c6 ipv6\_tcp\_server\_usb.pdf

在pc a侧执行

Usbpc.exe read 38

判断返回是否为26 02，是的话

在pc a侧执行

Usb-w.exe 1 xxx.pdf

5 被测开发板ipv6,tcp,client测试

Usbpc.exe mode 0 0

在pc b侧执行

Python ipv6.py pcb的ipv6地址 s6 ipv6\_tcp\_client\_usb.pdf

在pc a侧执行

Usbpc.exe ripv6 0 pcb的ipv6地址

Usbpc.exe read 38

判断返回是否为26 02，是的话

在pc a侧执行

Usb-w.exe 1 xxx.pdf

6 被测开发板ipv6,udp测试

Usbpc.exe mode 0 0

在pc b侧执行

Python ipv6.py pcb的ipv6地址 u6 ipv6\_udp\_usb.pdf

在pc a侧执行

Usbpc.exe read 38

判断返回是否为26 02，是的话

在pc a侧执行

Usb-w.exe 1 xxx.pdf

总线模式（pc a侧只需要打开串口终端观察提示，并依次在pcb执行下述命令）

1 被测开发板ipv4,udp测试

Python ipv4.py 192.168.1.100 1234 u4 ipv4\_udp\_bus.txt

2 被测开发板ipv4,tcp,client测试

Python ipv4.py 192.168.1.100 1234 s4 ipv4\_tcp\_server\_bus.txt

3 被测开发板ipv4,tcp,server测试

Python ipv4.py 192.168.1.103 1234 c4 ipv4\_tcp\_client\_bus.txt

4 被测开发板ipv6,tcp,server测试

Python ipv6.py fe80::1 1234 c6 ipv6\_tcp\_server\_bus.txt

5 被测开发板ipv6,tcp,client测试

Python ipv6.py pcb的ipv6地址 s6 ipv6\_tcp\_client\_bus.txt

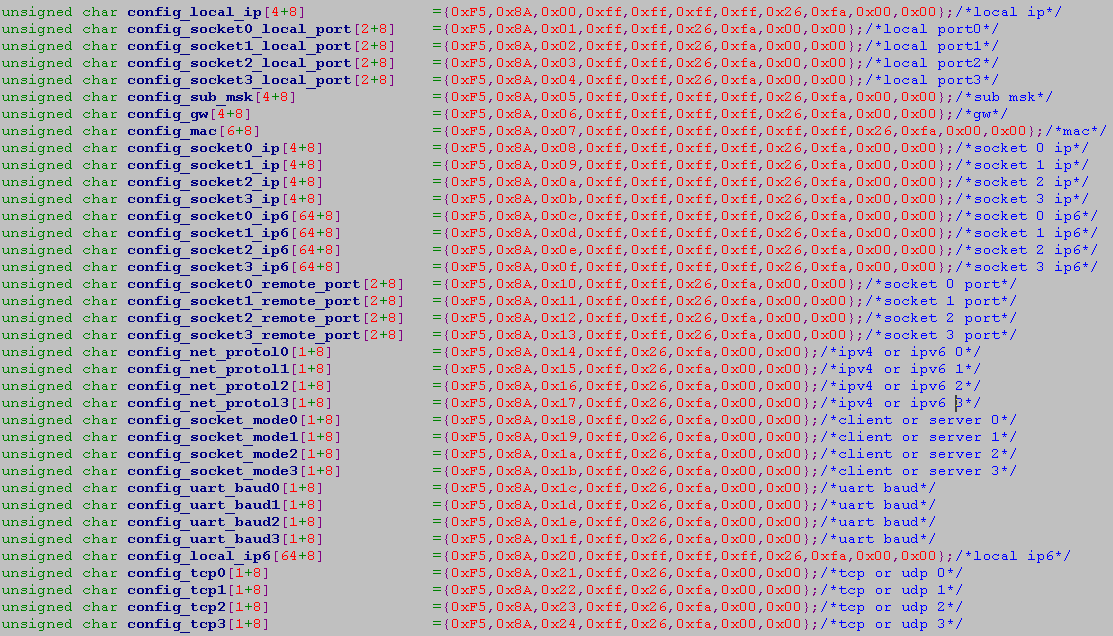
6 被测开发板ipv6,udp测试

Python ipv6.py pcb的ipv6地址 u6 ipv6\_udp\_bus.txt

附录：

一 串口配置测试说明

为方便测试，开发了一个pc上python脚本方便执行配置命令。支持所有的参数配置



**注com5为我这边的串口号，在设备管理器里查询。测试时注意修改为你实际的板子串口号**

1 读取参数

Configure com5 read addr len

Addr为数组里的0xf5,0x8a,后的那个字节

Len 为接收多少个字节，例如要读取本地ip地址则addr为0，len为4+8-1=11

举例：

config.py com5 read 0 11

Excute read

f58b00c0a8016726fa0470

则读到的本地ip为 c0a80167 换算为字符则是192.168.1.103

config.py com5 read 32 71

Excute read

f58b20666538303a3a3300000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000026fa049a

则读到的本地ipv6为666538303a3a33 换算为字符则是fe80::3

2 修改远端ipv4地址

Configure com5 ripaddr socketNo socketIp

socketNo为修改哪个socket的ip地址，如果是串口模式则为0-3 总线和usb模式为0

socketIp为需要设置的ip地址 如192.168.1.123

举例：

config.py com5 ripaddr 0 192.168.1.123

Excute ripaddr

[245, 138, 8, 192, 168, 1, 123, 38, 250, 4, 139]

f58c00

以上修改了远端socket0的ipv4地址为192.168.1.123，**然后模块返回执行成功f58c00便是命令执行成功，如果出现crc错误则会返回f58c01**

3 修改本地ipv4地址

Configure com5 lipaddr socketIp

与修改远端ip地址的区别是ripaddr改为lipaddr,socketNo这个参数没有

举例：

config.py com5 lipaddr 192.168.1.123

Excute lipaddr

[245, 138, 0, 192, 168, 1, 123, 38, 250, 4, 131]

f58c00

4 修改远端端口号

Configure com5 rport socketNo socketPort

socketNo为修改哪个socket的端口号，如果是串口模式则为0-3 总线和usb模式为0

socketPort为需要设置的端口号，如1234

举例：

config.py com5 rport 0 1234

Excute rport

[245, 138, 16, 4, 210, 38, 250, 3, 133]

f58c00

5 修改本地端口号

Configure com5 lport socketNo socketPort

socketNo为修改哪个socket的端口号，如果是串口模式则为0-3 总线和usb模式为0

socketPort为需要设置的端口号，如1234

举例：

config.py com5 lport 0 1234

Excute lport

[245, 138, 1, 4, 210, 38, 250, 3, 118]

f58c00

6 修改本地ipv6地址

Configure com5 lipv6 ipv6addr

Ipv6addr为你要设置的ipv6地址，如fe80::5

举例：

config.py com5 lipv6 fe80::5

Excute lipv6

[245, 138, 32, 7, 102, 101, 56, 48, 58, 58, 53, 255, 255, 255, 255, 255, 255, 25

5, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 25

5, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 25

5, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 25

5, 255, 38, 250, 60, 106]

f58c00

7 修改远端ipv6地址

Configure com5 ripv6 socketNo ipv6addr

socketNo为修改哪个socket的ip地址，如果是串口模式则为0-3 总线和usb模式为0

Ipv6addr为你要设置的ipv6地址，如fe80::5

举例：

config.py com5 ripv6 0 fe80::5

Excute ripv6

[245, 138, 12, 7, 102, 101, 56, 48, 58, 58, 53, 255, 255, 255, 255, 255, 255, 25

5, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 25

5, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 25

5, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 25

5, 255, 38, 250, 60, 86]

f58c00

8 修改子网掩码

Configure com5 submsk msk

Msk为需要修改的子网掩码，如255.255.255.0

举例

config.py com5 submsk 255.255.255.0

Excute submsk

[245, 138, 5, 255, 255, 255, 0, 38, 250, 5, 161]

f58c00

9 修改网关

Configure com5 gw gateway

Gateway为要修改的网关，如192.168.1.1

举例

config.py com5 gw 192.168.1.1

[245, 138, 6, 192, 168, 1, 1, 38, 250, 4, 15]

f58c00

10 修改mac地址

Configure com5 mac macaddr

Macaddr为要修改的mac地址，如12-34-56-78-12-32

举例

config.py com5 mac 12-34-56-78-12-32

Excute mac

[245, 138, 7, 12, 34, 56, 78, 12, 32, 38, 250, 3, 134]

f58c00

11 修改ipv4或者ipv6模式

Configure com5 protol socketNo mode

socketNo为需要修改哪个socket的协议

Mode为1则是ipv6，为0则是ipv4

举例

config.py com5 protol 0 0

Excute protol

[245, 138, 20, 0, 38, 250, 2, 179]

f58c00

12 修改tcp或者udp模式

Configure com5 tcp socketNo mode

socketNo为需要修改哪个socket的协议

mode为1则是tcp，为0是udp

举例

config.py com5 tcp 0 1

Excute tcp

[245, 138, 33, 1, 38, 250, 2, 193]

f58c00

13 修改server或者client模式

Configure com5 mode socketNo client

socketNo为需要修改哪个socket的协议

client为1是server模式，为0是client模式

举例

config.py com5 mode 0 1

Excute mode

[245, 138, 24, 1, 38, 250, 2, 184]

f58c00

14 修改波特率

Configure com5 baud socketNo baudrate

Baudrate为0-8

0—115200，1---128000，2---256000，3---460800，4---921600，5---1000000

6---2000000，7---4000000，8---6000000

举例

config.py com5 baud 0 0

Excute baud

[245, 138, 28, 0, 38, 250, 2, 187]

f58c00

15 获得网络连接状态

Configure com5 read 38 8

38为网络状态的地址，8是读回8个字节

举例

config.py com5 read 38 8

Excute read

f58b260526fa02cb

所有状态如下

#define NETWORK\_CNN\_OK 0x02

#define NETWORK\_ADDR\_CONFLICT 0x03

#define NETWORK\_INVALID\_ADDR 0x04

#define NETWORK\_CNN\_TIMEOUT 0x05

#define NETWORK\_WIRE\_DISCONNECT 0x06

#define NETWORK\_REMOTE\_DISCONNECT 0x07

#define NETWORK\_LOCAL\_DISCONNECT 0x08

二 usb配置测试说明

Usb的配置和串口的配置基本一致，使用程序为usbc.exe,不需要输入com5，在read时最后一个需要读回多少字节不需要输入

举例

usbc.exe mode 0 1

CMD:8

f5 8a 18 01 26 fa 02 b8

ACK:

F5 8C 00

以上为将usb模式下的socket0的client/server修改为server模式，模块返回f5 8c 00表示设置成功