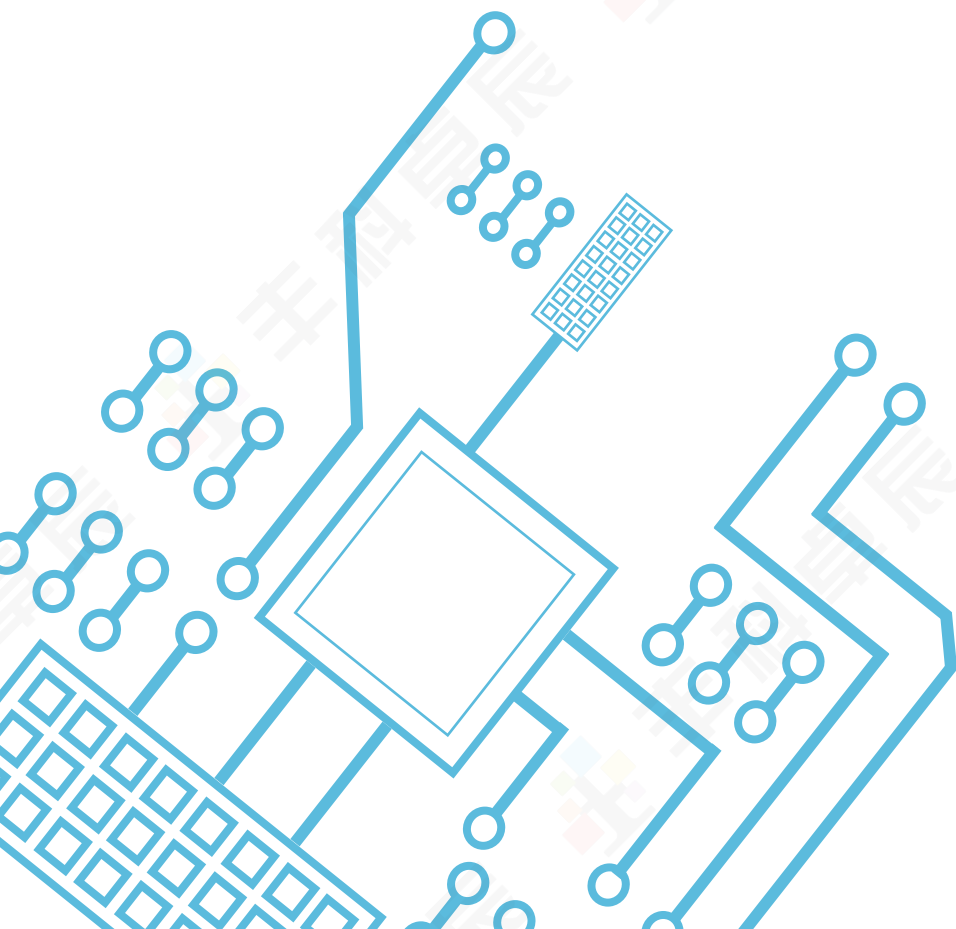




# 使用说明书

丰科卓辰-百G接口



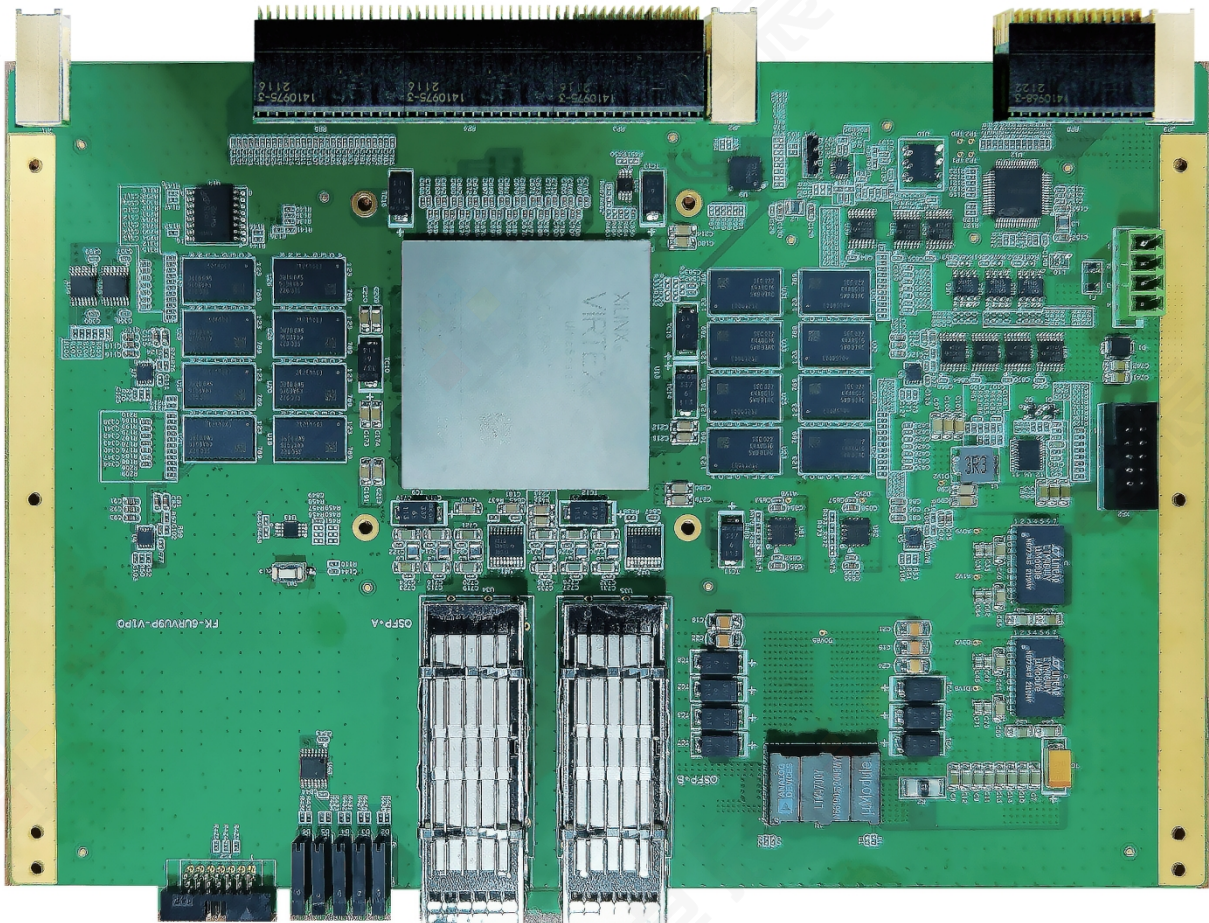
# 目录

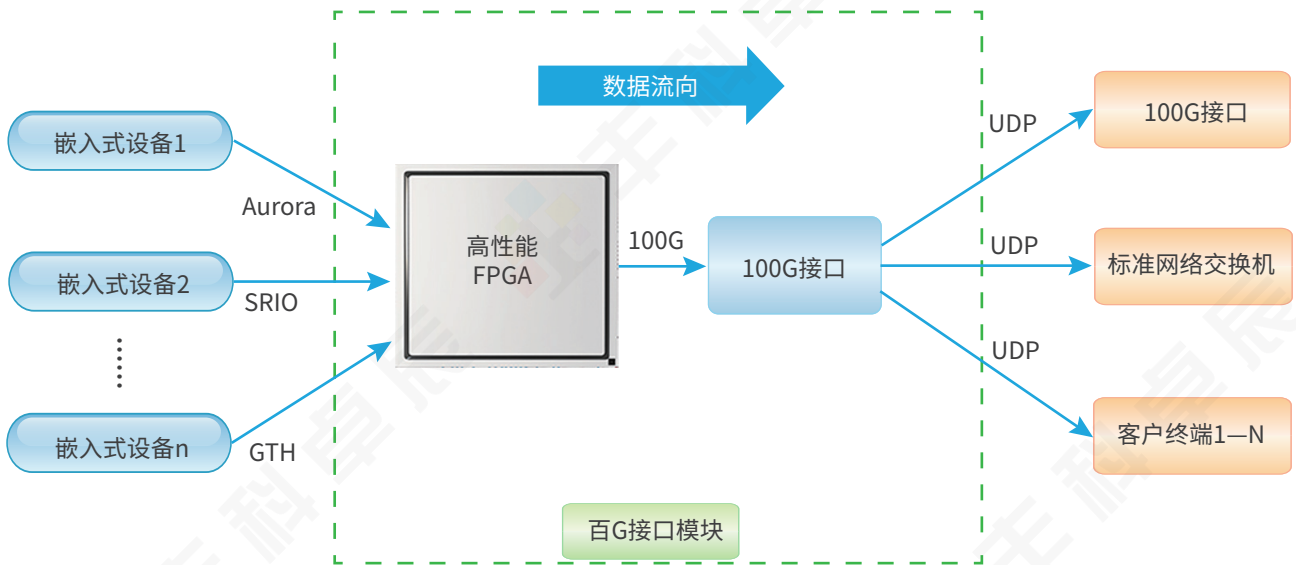
1、接口说明	-----	1
2、传输规则	-----	2
2.1、100G接口IP与端口号	-----	2
2.2、高速通道与低速通道	-----	3
2.3、通道与发送端口对应关系	-----	4
2.4、通道与接收端口对应关系	-----	5
3、接口配置	-----	6
3.1、端口动态绑定	-----	6
3.2、端口静态绑定	-----	7
3.3、设置保存	-----	7
3.4、IP配置	-----	7
4、应用场景使用说明	-----	8
4.1、PC直连模式	-----	8
4.2、交换机连接模式	-----	8
4.3与其他百G接口互联	-----	10
5、100G接口数据帧详细格式	-----	11

## 1、接口说明

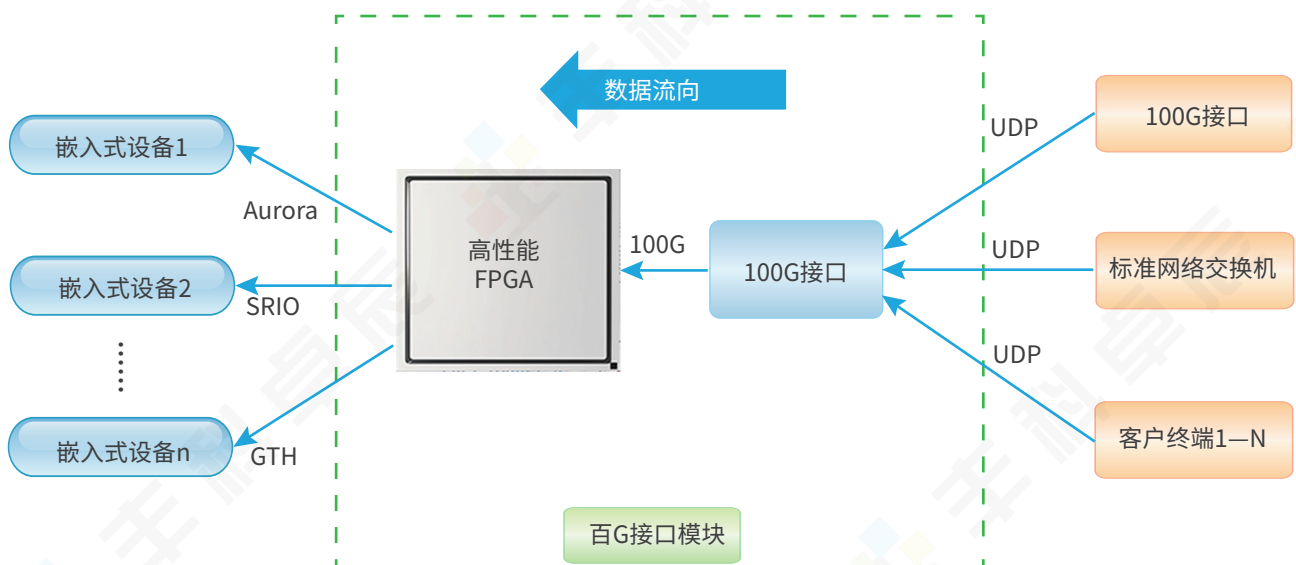
百G接口模块以高性能FPGA为核心，主要功能为可重构型高速协议转换，能够在外部配置下，将嵌入式设备内自定义高速协议数据转成标准网络数据，通过标准网络交换系统，实时传输到多地、多台的终端进行相关数据处理操作，或将多地、多台的终端数据传输到百G接口模块，然后转成嵌入式设备内自定义高速协议发给不同设备，进而发挥嵌入式设备的高速处理或其他特定性能，完成相关任务。百G接口模块可拆分成两个核心功能：数据发送与数据接收。

数据发送模式，是指百G接口模块可将数据通过标准UDP协议数据，根据数据内帧头的不同标识，将数据发送到不同的客户终端；每个百G接口最大可支持2048个客户端。





数据接收模式与数据发送的数据通路是相反的过程，是指百G接口模块可接收外部UDP数据，然后根据数据中携带的帧头信息，将数据以嵌入式高速协议实时分发到不同的嵌入式设备，以完成相应的数据任务。



## 2、传输规则

### 2.1、100G接口IP与端口号

每个100G接口都是单独的网口，有独立的IP地址和MAC地址，IP地址可通过配置端口进行配置。

每个100G接口的使用端口说明如下：

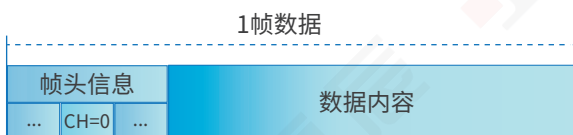
序号	端口号	数量	说明
1	8080	1	配置端口,外部向该端口发送配置信息
2	10000~12047	2048	数据发送模式下,发送数据的目的端口,支持向2048个端口发送数据,发送源端口与目的端口相同
3	32768~32775	8	数据接收模式下,数据接收端口,外部向该端口发送数据

如果百G接口模块包含2个100G接口,则2个100G接口输出的内容相同,IP不同,端口对应关系相同。(2个接口是为了提供更多的输出口,以适应带宽过高,终端需要更多设备接收处理的场景)

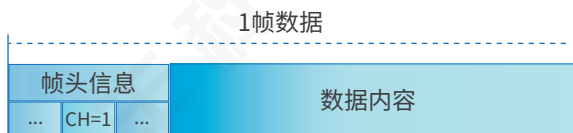
## 2.2、高速通道与低速通道

嵌入式设备在不同的工作模式下,会产生多路信号数据,每路数据都是以数据帧的格式进行打包、传输,在每帧数据的帧头信息中,会有通道号的标识信息,用来标识此帧数据是哪路数据。每路数据可通过一路Aurora接口或者多路数据通过一路Aurora接口将数据发送到百G接口模块。

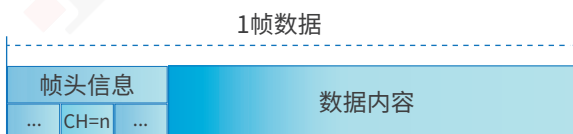
第1路数据



第2路数据



第n路数据

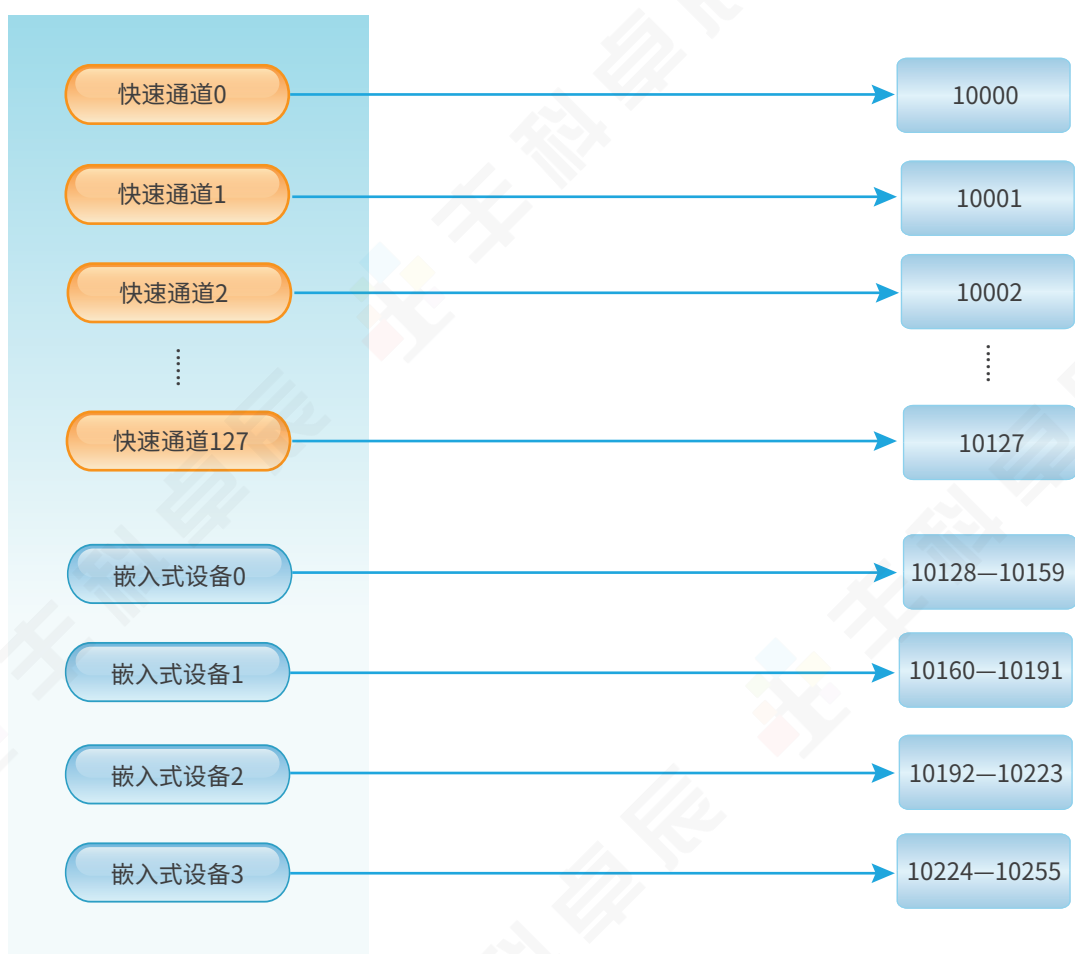


我们将Aurora接口设计成多路并行传输方式,根据传输带宽使用方式的不同,将每路Aurora接口分为高速通道与低速通道两种模式,且每路Aurora接口支持128路低速通道,以及4路高速通道,所有通道数据可并行传输。(注意低速通道与高速通道数据传输总带宽不可超Aurora接口总带宽)

### 2.3、通道与发送端口对应关系

每路Aurora接口分为128路低速通道与4路高速通道,每路Aurora接口共对应256个端口号,通道与端口对应关系为:

- ◇ 每路低速通道固定对应1个端口号;
- ◇ 每路高速通道固定对应32个端口,使用高速通道情况下,32个端口全部使用,轮循发送数据;每个端口轮循发送粒度为1KB。



第一路Aurora通道

对应的发送目的端口号

多路Aurora通道与端口号对应规则如下，最多支持8路Aurora通道。

数据输入接口	网络输出端口
第1路Aurora 8x密码	Aurora通道号:i=0 10000+i*256+0 ~ 10000+i*256+127:对应128路低速通道 10000+i*256+128+0 ~ 10000+i*256+128+31:对应高速通道0 10000+i*256+128+32~ 10000+i*256+128+63:对应高速通道1 10000+i*256+128+64 ~ 10000+i*256+128+95:对应高速通道2 10000+i*256+128+96 ~ 10000+i*256+128+127:对应高速通道3
第2路Aurora 8x	Aurora通道号:i=1 通道号分配规则参考第1路Aurora 8x的对应规则
第3路Aurora 8x	Aurora通道号:i=2 通道号分配规则参考第1路Aurora 8x的对应规则
第4路Aurora 8x	Aurora通道号:i=3 通道号分配规则参考第1路Aurora 8x的对应规则
第5路Aurora 8x	Aurora通道号:i=4 通道号分配规则参考第1路Aurora 8x的对应规则
第6路Aurora 8x	Aurora通道号:i=5 通道号分配规则参考第1路Aurora 8x的对应规则
第7路Aurora 8x	Aurora通道号:i=6 通道号分配规则参考第1路Aurora 8x的对应规则
第8路Aurora 8x	Aurora通道号:i=7 通道号分配规则参考第1路Aurora 8x的对应规则

#### 2.4、通道与接收端口对应关系

100G接口输入,通过不同的端口号区分将数据发向不同的Aurora接口上,端口与Aurora接口的对应关系如下:

从数据的帧头信息中,解析通道号,根据通道号判别通过Aurora接口的哪个通道发送到嵌入式设备。

数据网络输入端口	Aurora输出接口
32768	第1路Aurora 8x
32769	第2路Aurora 8x
32770	第3路Aurora 8x
32771	第4路Aurora 8x
32772	第5路Aurora 8x
32773	第6路Aurora 8x
32774	第7路Aurora 8x
32775	第8路Aurora 8x

### 3、接口配置

100G接口通过8080端口接收外部发送的配置信息，配置发送方可以是单独的配置电脑或者数据接收方都可以，只要能够与该100G接口通过网络连通。

注意：所有配置指令与反馈均以字符串的形式发送或接收。

#### 3.1、端口动态绑定

席位向100G网口的8080地址发送端口绑定配置指令（绑定该席位与端口），来通知该100G网口，对应该端口的数据发送到此席位；

指令内容(席位->板卡)	说明
utrans -ss id:0x0000	utrans bind is begin:绑定成功 utrans already bind:端口被占用
utrans -sp id:0x0000	utrans bind is close:端口绑定关闭

注意指令中的id:0x0000代表端口号，此处为0x0000，代表绑定端口10000到此席位，席位从端口10000接收数据；席位使用的端口与100G网口使用的端口一样；

例如：utrans -ss id:0x0010 对应数据端口10016；



特别注意指令格式,即使通道0,也需要发送0x0000字符串;

如果反馈端口已经被占用,则需要关闭绑定关系后再重新配置,才可以正常接收数据。

### 3.2、端口动态绑定

每个100G网口支持将数据发向另外一块板卡的100G网口,配置立刻生效。配置指令如下:

指令内容(席位->板卡)	反馈内容(板卡->席位)
uhwset -ss id:0x0000 0x00 0xc0a80105 0x000a35000000	utrans bind is begin:绑定成功 utrans already bind:端口被占用

指令说明:

- ◇ id:0x0000:代表通道号;
- ◇ 0x00:指定发给哪路Aurora接口;第1路:0x00,第2路:0x01.....;
- ◇ 0xc0a80105 :发送目的ip地址,发向哪个设备的100G网口;
- ◇ 0x000a35000000:发送目的mac地址;可通过arp指令查询mac地址;

如果已经绑定,还是通过“utrans -sp id:0x0000”指令来先关闭在绑定;

该配置参数可保存在板卡上,需要发送保存指令将参数保存下来。

如果采用UDP组播的方式,把“发送目的ip地址”“发送目的mac地址”设置成组播的ip和mac地址即可。

### 3.3、设置保存

席位向100G网口的8080地址发送保存配置指令,来保存板卡“uhwset”指令的设置参数,参数断电重启生效。

指令内容(席位->板卡)	反馈内容(板卡->席位)
svconfig	hwcfg saved to board

### 3.4、IP配置

席位向100G网口的8080地址发送IP配置指令,来更改板卡的默认IP地址,该配置执行后,需要断电重启才生效。

席位发向哪个100G网口,即更改哪个100G网口的IP。

指令内容 (席位->板卡)	反馈内容 (板卡->席位)
ipconfig 0xc0a80105	ipset up need reboot

0xc0a80105代表192.168.1.5。

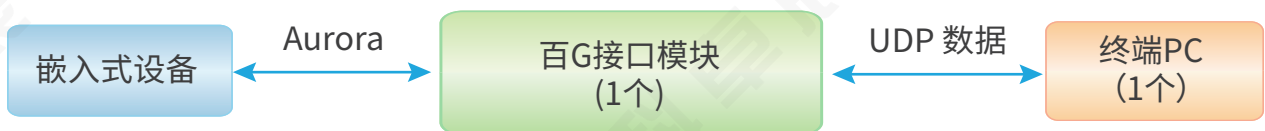
#### 4、应用场景使用说明

根据不同的使用应用场景,百G接口模块的使用方式可分为三种:PC直连模式,交换机连接模式,与其他百G接口模块连接模式。

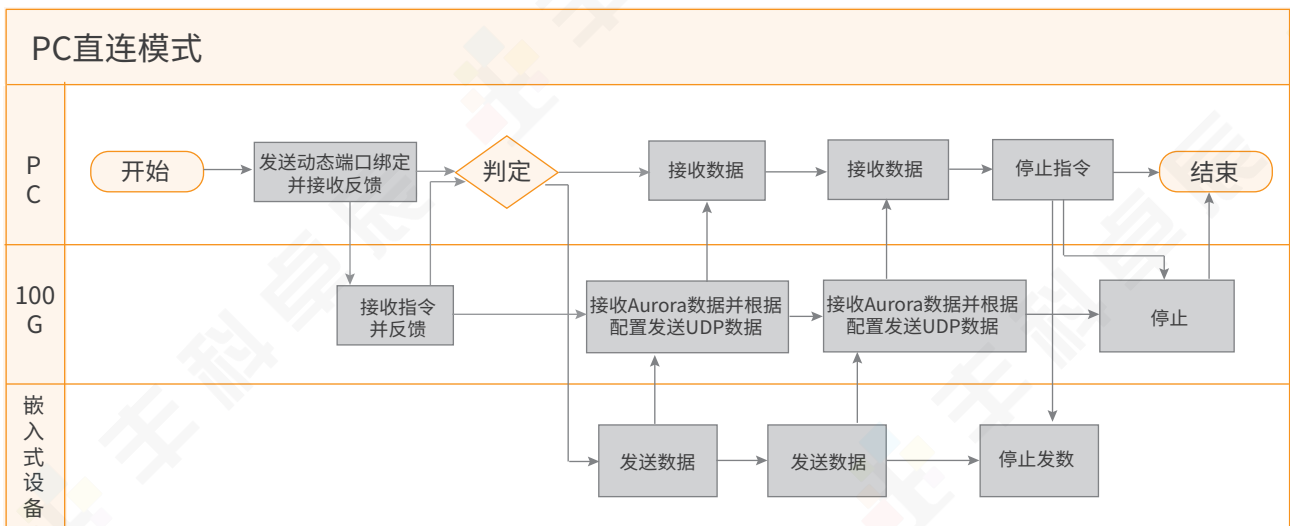
以下使用说明中,百G接口模块与嵌入式设备间的高速自定义协议以Aurora协议为例。

##### 4.1、PC直连模式

该模式使用框图如下所示,每个百G接口模块通过线缆与PC终端直连(PC终端一般是安装有100G接口网卡,为PC扩展标准100G接口)。该模式下,配置与数据接收均为终端PC。



工作流程说明如下:



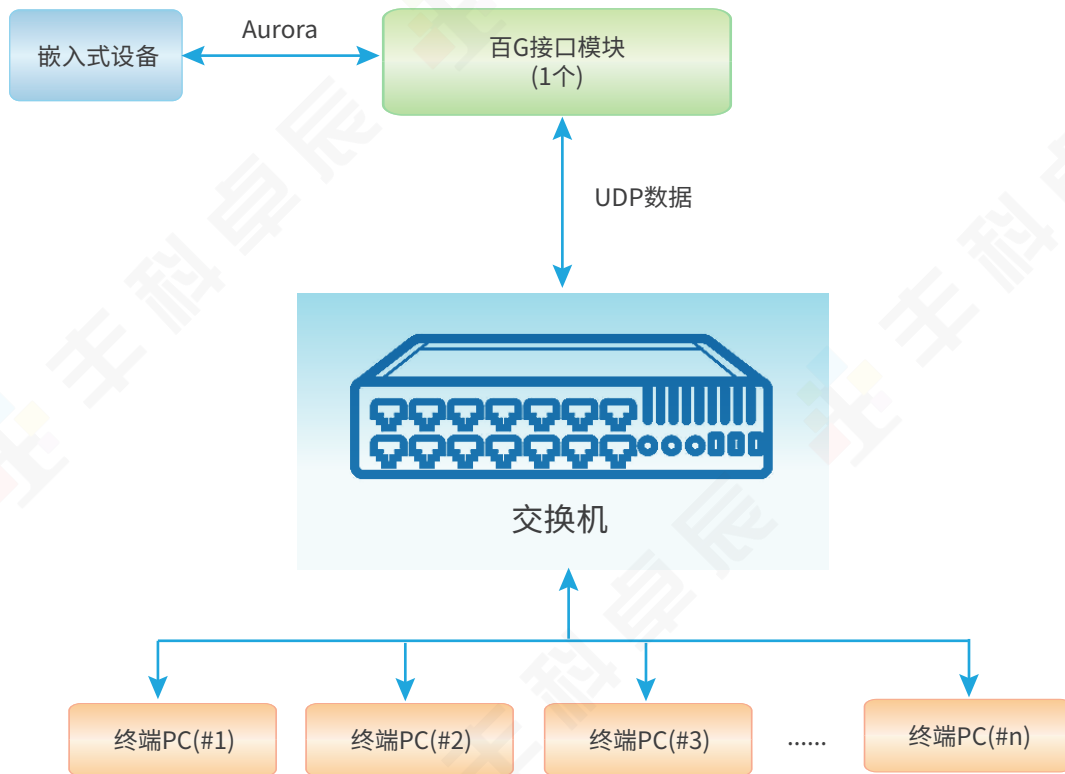
一台终端PC接收多路数据,根据端口对应关系,打开多个端口进行数据的接收即可。

注意,如果是高速通道,则每路高速通道对应的32个端口都需要打开,接收数据。

##### 4.2交换机连接模式

该模式使用框图如下所示,每个百G接口模块通过线缆与交换机上的100G接口连接,多台PC终端通过百G,10G,1G接口与交换机连接。

支持1台设备进行配置,多台数据只负责接收数据。或者每台设备进行配置,进行数据接收。该模式可将数据分散到多台设备上进行处理。数据接收方式,按照PC直连进行数据接收即可。



假设百G接口 IP为:192.168.1.7;以PC (#1) 接收第一路Aurora的第1路低速通道数据,PC (#2) 接收第一路Aurora的第2、3路低速通道数据,PC(#3)接收第一路Aurora的第0路高速通道数据为例,工作流程描述如下:

- 1、PC (#1) 打开本地端口8080,向192.168.1.7的8080端口发送“utrans -ss id:0x0001” (意思是通知100G接口,PC(#1)要接收端口10001对应通道的数据);
- 2、PC (#1) 进行监听10001的端口是否有UDP数据到来;
- 3、PC (#2) 打开本地端口8080,向192.168.1.7的8080端口发送“utrans -ss id:0x0002”和“utrans -ss id:0x0003”;
- 4、PC (#2) 进行监听10002、10003的端口是否有UDP数据到来;
- 5、PC (#3) 打开本地端口8080,向192.168.1.7的8080端口发送“utrans -ss id:0x0080”、“utrans -ss id:0x0081”、“utrans -ss id:0x0082”,以此类推32个指令;

6、PC (#2) 进行监听10128~10159的端口是否有UDP数据到来;端口128来的第1KB数据为数据流第1包, 端口129来的1KB为数据流第2包, 以此类推, 端口159来的数据为数据流第128包, 然后端口128来的第2个1KB数据, 则是数据流的第129包, 以此类推;

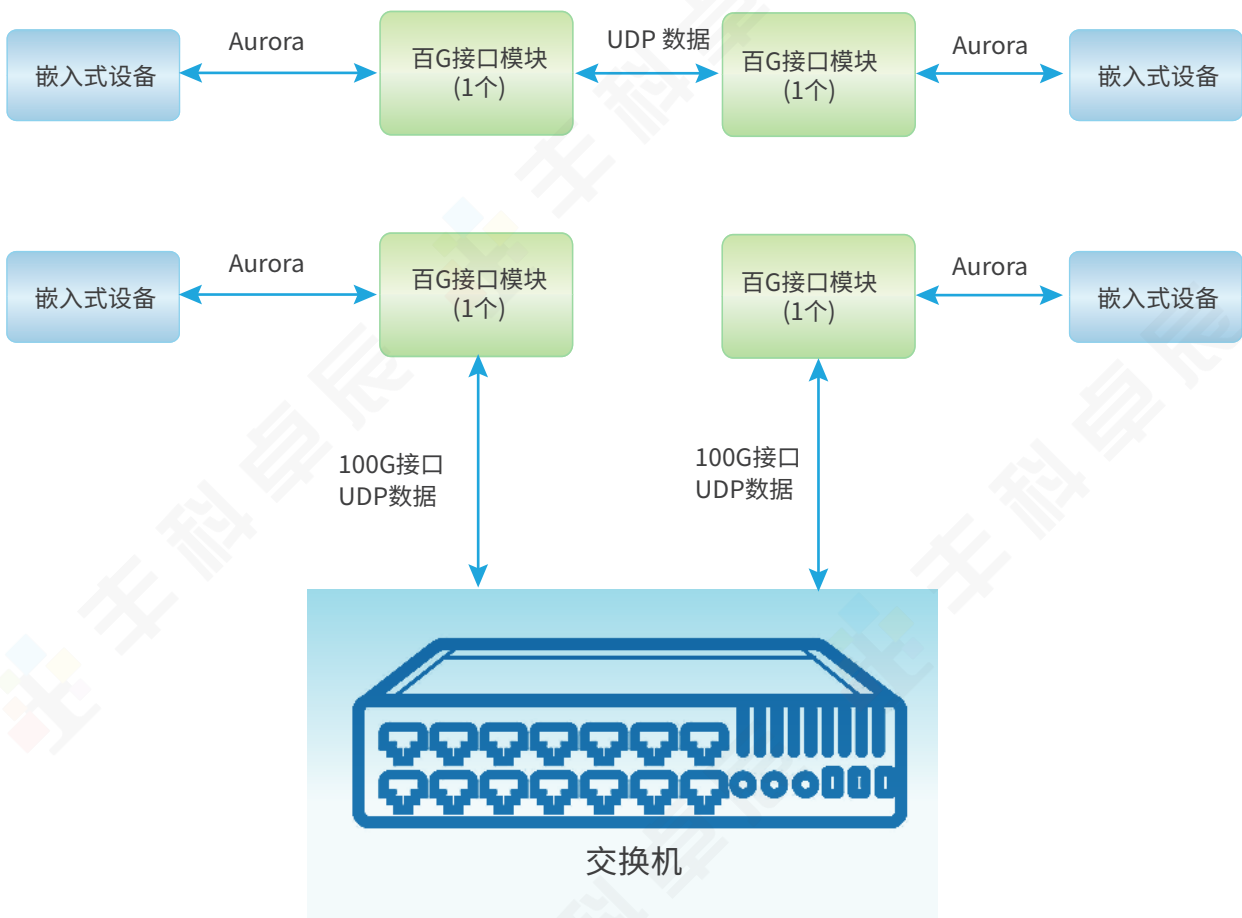
7、三个PC各自收取各自通道的数据进行相关处理即可

8、停止时, PC (#1)、PC (#2)、PC (#3) 按照发送开始指令一样, 发送“utrans -sp id:0x0001”.....等指令用来关闭端口。

#### 4.3与其他百G接口互联

该模式使用框图如下所示, 每个百G接口模块通过交换机或者直连的方式与其他百G接口进行连接。

按照静态绑定的方式, 可设置板卡发送数据到其他百G接口, 支持保存配置, 上电自动进行数据发送, 不再需要额外配置操作。



## 5、100G接口数据帧详细格式

### 数据帧格式

MAC_Header (14字节)	IP_Header (20字节)	UDP_Header (8字节)	TxBlock (8292字节)		
			TxBlock_Header (100字节)		TxBlock_Data (8192字节)
			Frame_Header (24字节)	Data_Header (76字节)	Data_Block_Body (IQ-Data) :IQ-samples

表2 群路DDC数据帧字段定义

帧格式	bytes length	Byte Number	字段长度	字段名称	字段标识	字段描述
MAC_Header	14	0-5	6	MAC目的地址	MAC-DA	MAC目的地址 (“little endian”),可配置
		6-11	6	MAC源地址	MAC-SA	MAC源地址 (“little endian”),可自动读取,也可人工配置
		12-13	2	MAC帧类型	MAC-LT	MAC type/length (“little endian”),默认取值:0x0800
IP_Header	20	0	1	版本+首部长度	VER_IHL	固定为0x45
		1	1	区分服务	TOS	固定为0
		2-3	2	总长	TLEN	总长=数据长度+28,最大为9000 如上述格式此处为8292+28=8320
		4-5	2	标识	IDENTIFY	标识 (Identify, 16 bits),帧计数器,每产生一个数据包,计数器就加1(不是序号)
		6-7	2	标志位+段偏移量	FLAGS_FO	固定为0
		8	1	存活时间	TTL	固定为0x80
		9	1	协议	PROTO	协议 (Protocol, 8 bits),指出此数据报携带的数据是何种协议,以便目的主机IP层知道应将数据交由哪个协议进行处理;固定为0x11
		10-11	2	首部校验和	IP_CRC	首部校验和 (Header checkSum, 16 bits),此处若使用现行IP网络,应正确计算该32位crc
		12-15	4	源地址	SRC_ADDR	源IP地址 (Source Address, 32 bits) 设备IP地址,可配置
16-19	4	目的地址	DES_ADDR	目的IP地址 (Destination Address, 32 bits) 发送IP地址,可配置		
UDP_Header	8	0-1	2	源端口	SRC_PORT	源端口 (Source Port, 16 bits) 设备源端口,可配置,用于多路数据通道号区分
		2-3	2	目的端口	DES_PORT	目的端口 (Destination Port, 16 bits),发送目的端口,用于多路数据通道号区分
		4-5	2	UDP数据包长度	UDP_LEN	UDP数据包长度,应为数据长度+8字节 UDP包头:如上述格式此处为8292+8=8300
		6-7	2	UDP校验和	UDP_CRC	不使用 固定位0

MAC帧头部结构体定义：

```
1.// u_char: 1字节,8位
2.// u_short: 2字节,16位
3.// u_int: 4字节,32位
4.
5.// 以太网帧数据头部 - 固定 14 个字节
6.typedef struct ethernet_header
7.{
8. u_char dest_mac[6]; // 目的地址(MAC地址)
9. u_char src_mac[6]; // 源地址(MAC地址)
10. u_short type;// 类型字段,用来标志上一层使用的协议,以便把MAC帧数据交给上一层
    的这个协议(IP:0x0800;)
11.}ETHERNET_HEADER;
```

IP数据包头部结构体定义：

```
1.// 网络层 IP数据包 首部 - 固定 20 个字节
2.typedef struct ip_header
3.{
4. u_char ver_ihl;// 版本(4 bits)+首部长度(4 bits),首部长度单位是32位字,所以需要
    *4 计算有多少个字节
5. u_char tos;// 区分服务(Type of Service, 8 bits)
6. u_short tlen;// 总长(Total Length, 16 bits),=首部长度+数据长度,最大为65535字节
    (2^16-1)
7. u_short identify;// 标识(Identify, 16 bits),IP软件在存储器中维持的一个计数器,每
    产生一个数据包,计数器就加1(不是序号)
8. u_short flags_fo;// 标志位(Flags 3 bits)+段偏移量(Fragment offset 13 bits)
9. u_char ttl;// 存活时间(Time to Live, 8 bits)
```

10. u\_char proto;// 协议 (Protocol, 8 bits),指出此数据报携带的数据是何种协议,以便目的主机IP层知道应将数据交由哪个协议进行处理
11. u\_short crc;// 首部校验和 (Header checkSum, 16 bits)
12. u\_long src\_addr;// 源地址 (Source Address, 32 bits)
13. u\_long des\_addr;// 目的地址 (Destination Address, 32 bits)
14. //u\_int op\_pad;// 选项与填充 (Option \* Padding, 32 bits)
- 15.}IP\_HEADER;

UDP数据包头部结构体定义:

- 1.// 传输层 UDP数据包 首部 - 固定首部 8 个字节
- 2.typedef struct udp\_header
- 3.{
4. u\_short src\_port; // 源端口 (Source Port, 16 bits)
5. u\_short des\_port; // 目的端口 (Destination Port, 16 bits)
6. u\_short len; // UDP数据包长度 (Datagram Length, UDP用户数据包的长度, 单位为字节)
7. u\_short crc; // 校验和 (CheckSum, 16 bits)
- 8.}UDP\_HEADER;



# 北京丰科卓辰电子技术有限公司

v1.0 2022.5

 联系  
热线 | 010-57325880

 联系  
邮箱 | [guangrui.liu@fkzctech.com](mailto:guangrui.liu@fkzctech.com)

 首页  
网站 | [www.fkzctech.com](http://www.fkzctech.com)

 公司  
地址 | 北京市昌平区科技园区超前路甲1号6号楼308室