

版本: V1.04

AFDX 仿真板卡

快速使用手册

CLV-XXA1



成都科洛威尔科技有限公司

地址: 成都市郫都区现代工业港港中路188号

TEL: 1878-0222-336 191-3621-6517

EMAIL: clovertech@163.com 公司官网: www.clvtech.net

声明

本手册中介绍的产品（包括硬件、软件和手册本身）的版权归成都科洛威尔科技有限公司所有，且保留所有权利。用户对产品的购买并不代表用户获得版权的许可，未经成都科洛威尔科技有限公司授权，任何人不得以任何方式复制本手册的任何部分。

本手册所包含的内容如有更改，恕不另行通知。

目录

声明	1
目录	2
图目录	4
表目录	5
版本说明	6
范围	7
1. 模块描述	8
2. 快速开始	8
2.1. 发布包文件说明	8
2.2. 使用说明	9
3. AFDX (ARINC-664) 总线	10
3.1. 概览	10
3.2. 总线特点	10
3.3. 网络拓扑结构	11
3.4. 端系统(ES)	12
3.5. 虚拟链路 (VL)	12
3.6. 冗余网络	13
3.7. AFDX 帧结构	13
3.8. EDE	14
3.9. 仿真卡中使用的术语	14
4. 安装	17
4.1. 安装准备	17
4.2. 硬件安装	17
4.3. 驱动安装	17
4.4. 软件安装	19
4.5. 线缆安装	20
5. 控制台 demo 程序使用	20
5.1. 使用前准备	20
5.2. 控制台 Demo 软件列表	21
5.3. Demo 程序使用	22
6. 总线仿真软件使用	22
6.1. 使用准备	22
6.2. 软件功能	22

6.3. 使用说明23

7. Linux 环境下 libusb 的安装 33

7.1. Libusb 的编译与安装33

7.2. Libusb 库的使用34

8. 故障维护 35

图目录

图 1 Windows 发布文件	9
图 2 Linux 发布文件	9
图 3 Linux 驱动和库文件	9
图 4 AFDX 网络总线拓扑	12
图 5 冗余网络	13
图 6 AFDX 帧格式	14
图 7 通道模式	15
图 8 驱动安装图 1	17
图 9 驱动安装图 2	18
图 10 驱动安装第六步	18
图 11 驱动安装完成	18
图 12 驱动安装完成后显示状态	19
图 13 硬件接口连接示意图	20
图 14 基本操作流程	23
图 15 设备列表	23
图 16 软件主界面	24
图 17 通道控制区界面	24
图 18 工作区界面	25
图 19 配置功能界面	25
图 20 发送管理界面	25
图 21 状态显示区界面	26
图 22 发送端口编辑界面	27
图 23 发送 VL 编辑完成界面	28
图 24 发送数据编辑界面	28
图 25 接收管理界面	29
图 26 接收 VL 编辑界面	29
图 27 接收端口编辑界面	30
图 28 接收 VL 编辑完成界面	31
图 29 接收数据查看界面	31
图 30 状态显示区界面	32
图 31 libusb 的安装	33
图 32 libusb 的使用步骤 1	34
图 33 libusb 的使用步骤 2	35

图 34 libusb 的使用步骤 3	35
---------------------------	----

表目录

表 1 控制台 demo 列表	21
-----------------------	----

版本说明

V1.00

初始版本。

V1.01

1. 更新仿真软件使用说明

V1.02

1. 增加 EDE 功能相关说明。

V1.03

1. 增加独立通道模式说明。
2. 更新仿真软件使用说明

V1.04

- 1、在 4.3 和 4.4 更新 linux 驱动和软件安装说明
- 2、增加第 7 章进行 libusb 和 linux 下使用库的描述
- 3、在 5.2 的 Demo 列表增加 AfdxApi_WriteMultMsgDemo、AfdxApi_XmtCycle500usDemo 描述
- 4、更新发布包 linux 相关描述，更新 linux 下 libusb 编译描述

范围

快速使用手册对 AFDX 仿真卡的基本功能和模式进行描述，并提供控制台 demo 程序以及总线仿真软件的使用说明。用户可以通过本手册了解如初始化硬件，完成参数设置并完成功能启用。控制台 demo 程序提供源码，客户可以根据对应的 demo 程序获取 API 函数的调用方法。

本文档适用与成都科洛威尔科技有限公司生产的 PCI、PCIE、USB、CPCIE/PXIE、CPCI 总线的 AFDX 仿真板卡，即板卡型号为 CLV-30A1、CLV-31A1、CLV-50A1B、CLV-41A1、CLV-40A1。

除《快速使用手册》外，我们还提供《用户手册》、《API 函数使用手册》等相关文档，用户可以通过这些文档了解必要的使用或者开发信息。

1. 模块描述

AFDX 仿真卡是一款基于 AFDX/ARINC-664 Part7（Avionics Full Duplex Switched Ethernet 航空电子全双工交换以太网）的 ES 端系统测试和仿真模块。单个模块最多提供 4 个以太网接口，最大可支持 2 个双冗余 ES 系统或者 4 个单网口 ES 系统。

模块支持符合 AFDX 协议的数据接收和发送功能。为了适应多种仿真和测试工况，仿真卡提供了不同的接收和发送模式。

·接收模式：

仿真卡提供 VL 接收、VL 监控以及顺序监控三种接收模式。

VL 接收模式 (VL-MODE)：按照预先创建的 VL 参数进行接收（即按照 VLID、端口地址进行接收）。该种模式下能够支持采样(sampling)、队列 (Queuing)、服务访问 (SAP) 三种不同的端口接收，队列端口和服务端口能够创建多个 buffer 进行接收，接收的数据会根据端口存放，用户可通过对应的 VL 参数进行数据访问。该模式适用于绝大多数仿真和测试应用。

VL 监控模式 (VL-Monitor-MODE)：按照预先创建的 VL 参数进行接收（仅根据 VLID 值），该模式下将符合 VLID 的数据进行顺序监控和存储。该模式适合用于关心指定 VL 链路的数据，而不关心具体数据目标端口的情况。

顺序监控模式 (SEQ-Monitor-MODE)：无需创建 VL 参数，监控总线上所有 VL 的数据，接收到的数据按照接收的先后顺序进行存放。该模式常作为总线监控功能时使用。

·发送模式：

VL 发送模式 (VL-MODE)：按照预先创建的 VL 参数进行发送。该种模式下能够支持采样 (sampling)、队列 (Queuing)、服务访问 (SAP) 三种不同的发送端口，队列端口和服务访问端口能够创建多个发送 buffer,该模式下会遵循 VL 内的 BAG 时间间隙参数进行发送。该模式适用于绝大多数数据发送测试和仿真。

2. 快速开始

2.1. 发布包文件说明

2.1.1. Windows

Windows 环境下模发布文件目录如下图所示。

名称	修改日期	类型
demo	2024/6/29 17:04	文件夹
doc	2024/6/29 17:05	文件夹
driver	2024/6/29 17:04	文件夹
lib	2024/6/29 17:05	文件夹
总线仿真软件	2024/6/29 17:06	文件夹
readme	2024/6/29 17:05	文本文档

图 1 Windows 发布文件

2.1.2. Linux

Linux 环境下发布文件目录如下图所示。

Demo	2025/10/16 23:44	文件夹	
doc	2025/10/17 12:33	文件夹	
kylinV10_FT2000_Arch64	2025/10/16 23:44	文件夹	
libusb-1.0.28	2025/10/17 12:35	文件夹	
ubuntu20.04_x86	2025/10/16 23:44	文件夹	
note.txt	2025/10/16 23:39	文本文档	1 KB
realese.txt	2025/10/16 23:10	文本文档	1 KB

图 2 Linux 发布文件

发布包默认提供 ubuntu20.04（X86 架构）和麒麟 V10 国防版（Arch64 架构），对应版本下提供驱动文件、库文件以及总线仿真软件。

名称	修改日期	类型
driver	2025/10/16 23:44	文件夹
lib	2025/10/16 23:44	文件夹
总线仿真软件	2025/10/16 23:44	文件夹

图 3 Linux 驱动和库文件

2.2. 使用说明

板卡在使用前需要进行正确的安装，包括硬件板卡、驱动以及软件安装。板卡和驱动安装可参见**第 4 章节-安装**。

AFDX 仿真卡提供两种类型的软件。一种为控制台 demo 软件，一种是总线仿真软件。

控制台 demo 软件存放在发布文件包中的“demo”文件夹，无需安

装，可直接编译后运行。控制台 demo 程序提供软件源代码，提供不同模式下的典型应用例程。如果您是软件开发者需要使用仿真卡进行二次开发或者想要了解 API 函数如何使用，那么请参见**第 5 章节-控制台 demo 程序使用**。

总线仿真软件提供 UI 操作界面，用户可以通过软件实现对硬件板卡的主要功能进行操作，此软件不提供源码，使用前需要安装此软件。如果您对板卡的功能已经有一定了解，想快速的使用板卡对总线进行仿真测试，那么您可以使用总线仿真软件对硬件设备进行快速的部署，实现总线测试的快速激励和仿真。您可以参见第**6 章节-总线仿真软件使用**部分的内容。

3. AFDX（ARINC-664）总线

在使用模块前，您可能需要对 AFDX（ARINC-664）总线知识进行一些基本的了解。如果您已经对这些知识有了相应的了解，那么你可以跳过这个章节。

3.1. 概览

ARINC-664 是一种航空电子数据网络标准，特别是指其第 7 部分，即 Avionics Full Duplex Switched Ethernet (AFDX)，中文名称为航空电子全双工交换以太网。AFDX 是基于以太网技术的航空电子通信协议，旨在为现代飞机提供高带宽、确定性和可靠的数据通信网络。

ARINC-664 标准分为多个部分，其中第 7 部分（AFDX）是最核心的部分，定义了航空电子全双工交换以太网的网络架构和协议。其他部分包括：

- Part 1: 系统概念和概述
- Part 2: 以太网物理层和数据链路层规范
- Part 3: 基于互联网的协议和服务
- Part 4: 基于互联网的地址结构和分配编号
- Part 5: 网络互连服务和功能元素
- Part 6: 保留
- Part 7: AFDX 网络
- Part 8: 上层服务

3.2. 总线特点

相对于传统的大飞机的飞行总线，它能提供更高的传输速率（基

于 10Mb/100Mb 网络传输)、更高的可靠性(虚拟链路)、高确定性(双冗余通道、确定的延迟和抖动)、更方便的接入方式(星形结构)以及更好的便捷性(基于成熟的以太网协议)。

基于以太网: ARINC-664 使用 IEEE 802.3 以太网标准作为其物理层和数据链路层的基础,但对其进行了扩展和修改,以满足航空电子系统的高可靠性和确定性要求。

全双工交换网络: AFDX 使用全双工交换以太网,避免了传统以太网中的碰撞问题,并通过交换机实现点对点通信。

虚拟链路 (Virtual Links, VLs): AFDX 引入了虚拟链路的概念,虚拟链路是网络中预定义的、具有确定带宽和延迟的点对点或组播通信通道。每个虚拟链路都有其特定的带宽和延迟要求,确保数据传输的确定性。

冗余设计: AFDX 网络通常由两个独立的冗余网络(网络 A 和网络 B)组成,每个数据包同时在两个网络上传输,接收端系统负责管理冗余帧,确保数据的可靠性和顺序。

确定性通信: AFDX 提供了确定的带宽、延迟和抖动控制,确保关键航空电子系统的实时通信需求。

网络配置: AFDX 网络是预配置的,所有的连接、地址和带宽需求在系统设计时就已经确定,网络配置在飞机电子系统升级或更改时更新。

兼容性: AFDX 可以与其他航空电子协议(如 ARINC 429)兼容,通过 SAP 端口实现不同协议之间的通信。

3.3. 网络拓扑结构

AFDX 网络采用星型拓扑方式,基本拓扑结构如下图所示。该图展示了通过交换机互联多个 ES 端系统的拓扑结构,应当注意的是该图并没有展示出冗余网络。

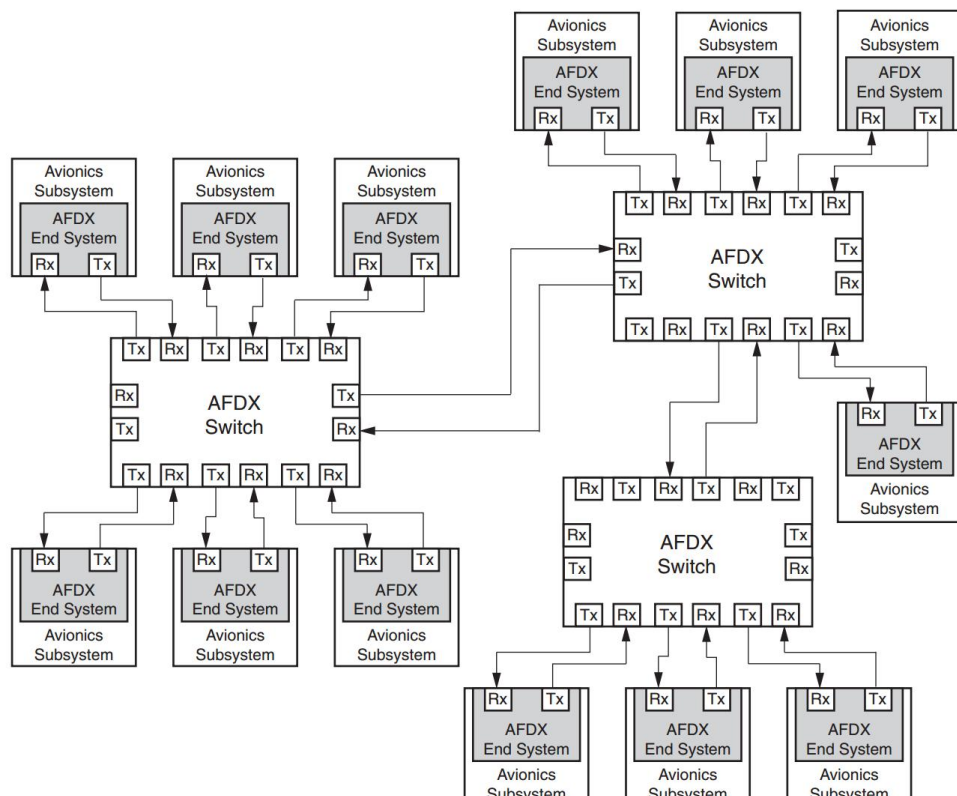


图 4 AFDX 网络总线拓扑

这种结构下，端系统与交换机之间形成星型连接，每个端系统唯一地接入到某一交换机的端口。例如，在实际的航空电子系统中，各个航空电子设备作为端系统，通过 AFDX 端口与交换机相连，实现数据的传输和交互。

3.4. 端系统(ES)

ES 是 End System（端系统）的缩写。它是 AFDX 网络的核心组件之一，负责与航空电子子系统（如传感器、控制器、执行器等）交互，并通过 AFDX 交换机实现可靠的数据通信。每个终端系统连接到一个 AFDX 交换机。在一个系统中可能会有多于一个 AFDX 交换机存在，且这些交换机彼此相连。不同于标准以太网，在 AFDX 中数据发送源端口是确定且唯一的，而以太网可以有多个发送源端口。

3.5. 虚拟链路（VL）

虚拟链路（Virtual Link, VL）是 AFDX 网络中的一个核心概念，用于在航空电子系统中实现确定性、可靠的点对点或组播通信。虚拟链路是 AFDX 网络中的逻辑通信通道，每个 VL 都有预定义的带宽、延迟和抖动要求，确保数据传输的实时性和可靠性。

虚拟链路可以是点对点的（一个发送端到一个接收端），也可以是

组播的（一个发送端到多个接收端）。这种设计取代了传统航空电子系统中复杂的点对点布线，简化了网络架构。

每个虚拟链路都有固定的带宽分配（通过 Bandwidth Allocation Gap, BAG）。延迟和抖动是预先确定的，确保数据传输的确定性。

每个虚拟链路都有一个唯一的标识符（VL ID），用于在 AFDX 网络中识别和路由数据帧。

虚拟链路可以在冗余网络（NETA 和 NETB）上同时传输数据，确保高可靠性。

每个虚拟链路的帧都包含一个序列号（Sequence Number, SN），用于检测丢帧和确保数据顺序。

3.6. 冗余网络

为了保障系统可靠性和数据传输连续性的关键机制，AFDX 网络采用双冗余网络设计。即，AFDX 网络包含两个完全独立的物理网络（NETA 和 NETB），每个终端系统（End System）通过双以太网端口（NETA 和 NETB）同时连接这两个网络。数据帧在发送时会被复制并同时通过两个网络传输，确保即使一个网络发生故障，数据仍可通过另一网络到达接收端。

AFDX 在工作时两条网络 NETA 和 NETB 都会收发数据，数据接收端在接收到数据后通过完整性检测和冗余判断后收取需要的数据（保留一份有效数据）。

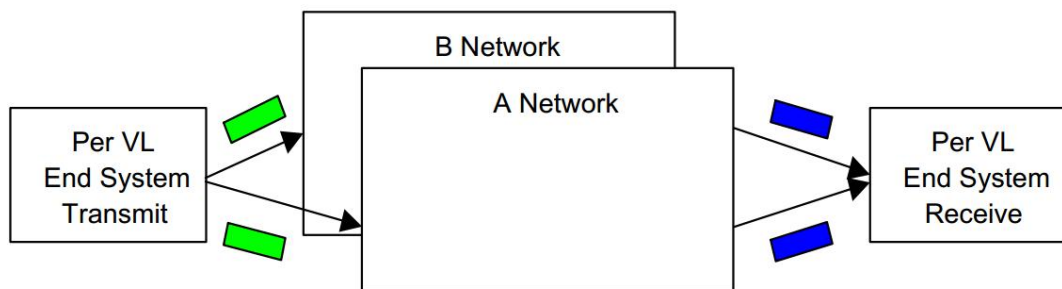


图 5 冗余网络

3.7. AFDX 帧结构

AFDX 的数据帧遵循 IEEE 802.3（以太网）标准。每一个 AFDX 数据帧都包含由虚拟链路确定的目标地址和源地址，与标准以太网帧不同的是在每一个 AFDX 的帧尾部（帧校验序列前）都有一个 SN 序号字节表示帧的序列，并且每一条 VL 都有自己独立的 SN，SN 值从 1 到 255 循环，0 表示复位。

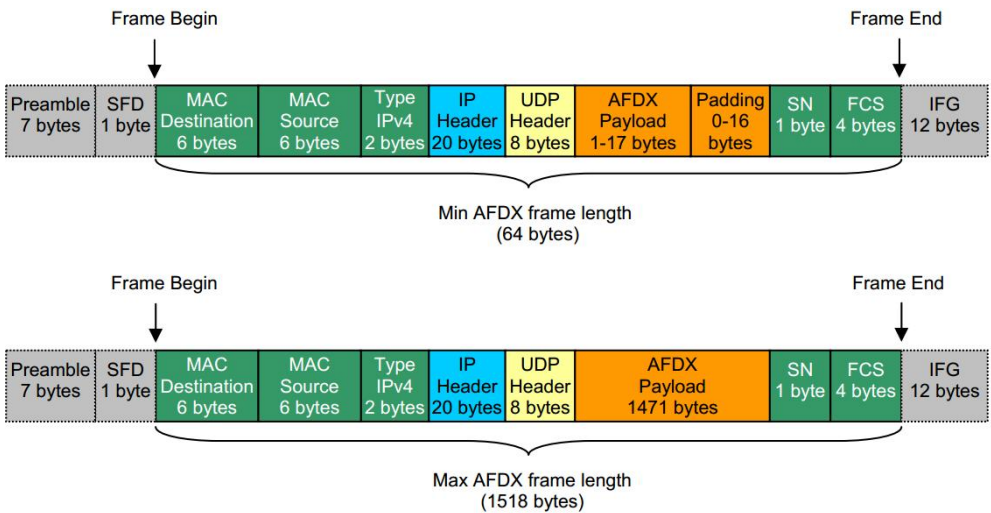


图 6 AFDX 帧格式

AFDX 的帧长度范围可以从 64~1518 字节，对应的有效载荷范围为 1~1471 字节。

3.8. EDE

EDE（Error Detection Encoding，错误侦测编码）是在 AFDX 负载内容的基础上，通过添加 EDE 头和 EDE 尾部，来对 AFDX 的数据进行校验的一种方法。

AFDX 负载内容		
EDE 头 (SN+时标)	负载	EDE 尾部 (CRCX+CRCY)

EDE 校验仅对负载内容本身进行校对，不对其他内容进行校对。仿真卡的 EDE 默认使用 Boeing 的 EDE 多项式。如果使能了 EDE 功能，仿真卡将自动为传输的内容添加 EDE 首部和尾部校验。在非 IP 分包时 EDE 的首部和尾部使用硬件自动产生，在需要进行 IP 分包时 EDE 校验则由 API 层软件进行自动产生。

3.9. 仿真卡中使用的术语

AFDX 仿真卡基于 AFDX 标准进行开发作为 ES 进行数据收发仿真，在使用仿真卡进行功能仿真前，需要明确以下基本定义。

3.9.1. 通道与 NETA/B 网络

仿真卡中提及的通道即为 AFDX 中的 ES 端系统收发接口，即 ES 通道。

仿真卡最多支持 4 个以太网接口（分别为 NET0~3），支持冗余工

作模式和独立工作模式。如果工作在冗余模式时，将由两个以太网接口组成一个冗余 ES 通道。如果工作在独立工作模式时，那么单个网口也可以成为一个 ES 通道（此时该通道将不具备冗余功能）。

仿真卡可最多组成 4 个独立 ES 通道、2 个冗余 ES 通道或者 1 个冗余通道加 2 个独立通道的组合方式。

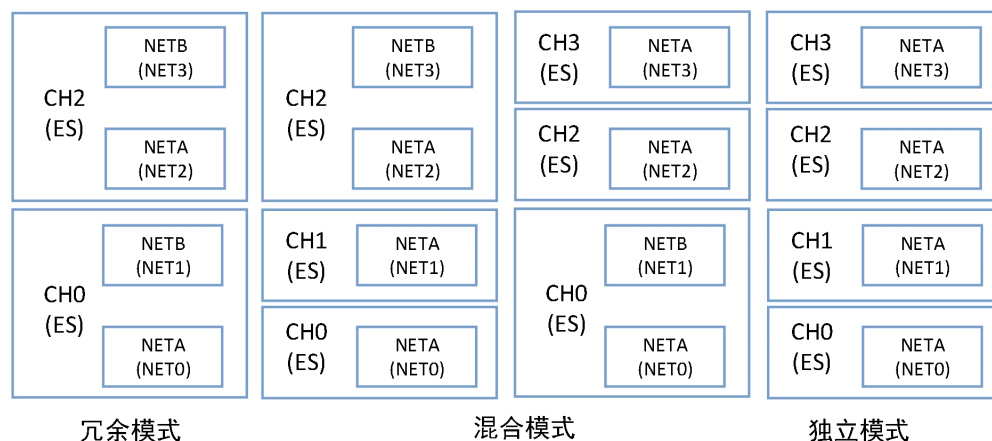


图 7 通道模式

当工作在冗余模式时，由两个网络接口（NETA 和 NETB）组成一个冗余通道。如上图所示的冗余模式的 CH0，在该模式下网口 NET0/NET1 将组成通道 0，其中 NET0 作为 NETA，NET1 作为 NETB。

冗余工作模式下，通道内的 NETA 和 NETB 都会进行数据收发，数据接收端在接收到数据后通过完整性检测和冗余判断后收取需要的数据。仿真卡由硬件实现完整性检测和冗余判断。

在冗余工作模式下，除了冗余接收外还支持“仅 NETA 接收”、“仅 NETB 接收”、以及“同时接收 NETA 和 NETB”的数据。同样发送端也支持“仅 NETA 发送”、“仅 NETB 发送”以及“同时发送”三种模式，在进行同时发送模式时可以通过 skew 参数模拟哪个网络优先发送，以及优先发送的时间参数。

如果工作在独立工作模式时，那么只能选择当前的网口进行数据收发操作。

3.9.2. VL 和子虚拟链路（subVL）

虚拟链路（Virtual Links）定义了一帧数据的源和目标地址。也就是数据的通信总是以虚拟链路的方式进行点对点，或者多播的方式进行通信。在 AFDX 网络中 VL ID 不能重复，且 BAG、SN 序号参数都是以 VL 为对象进行约束的。

子虚拟链路（subVL）是虚拟链路（VL）的内部逻辑划分，用于

区分同一 VL 内不同优先级或类型的数据。子 VL 不独立占用带宽资源，其带宽分配仍受父 VL 的总带宽限制。每个 VL 可以定义最多 4 个子链路。在每个 VL 内的数据发送采用子链路轮询的方式。

每一个子链路由不相等的 UDP 源和目标端口号定义，并且分配一个子链路类型（即端口类型）。AFDX 规范定义了三种类型的子链路类型，分别为：

采样端口(Sampling)：数据拥有固定格式并且通常为重复发送。例如：高度、维度、经度；

队列端口（Queuing）：数据没有固定格式，并且每次发送的数据可能会不同。例如：数据库、视频数据。

服务访问端口（SAP）：与非航空电子设备的网络兼容使用，支持 UDP 和 ICMP 协议。

注意：由于在实际使用子 VL 进行传输时，总是要进行端口类型和端口号进行分配，因此在实际使用仿真卡时，总是以端口的方式对子 VL 进行使用 and 描述。

3.9.3. BAG

BAG（Bandwidth Allocation Gap，带宽分配间隔）是控制虚拟链路（VL）数据传输的关键参数，用于确保每个 VL 的带宽使用符合预定义的确定性要求。BAG 约束了同一虚拟链路（VL）中连续两帧数据之间的最小时间间隔，单位是毫秒（ms）。例如定义了一个 VLID=22 的 BAG=32，那么在 VLID=22 的链路中传输数据包的速率不能比 32ms 快。

在 AFDX 网络中每条 VL 必须要有确定的 BAG 值。根据规范，BAG 的值必须为 1~128 毫秒之间，且为 2 的幂。允许的 BAG 的值可以用公式表示： $BAG = 2^K (k \in \{0, 1, 2, \dots, 7\})$ 。

应当注意的是，在使用仿真卡进行数据发送时，每个 VL 的实际带宽不仅只由 BAG 确定，还和发送的最大帧长度（LMAX）相关。

即， $\text{带宽} = \frac{20\text{BYTE} + \text{LMAX}}{\text{BAG}}$ ，单位为 Kbyte/s。用户在创建 VL 时应当注意创建的所有的 VL 的带宽不能超过当前网络的带宽的总和。

在标准规范中最小的 BAG 时间为 1ms。仿真卡除了提供标准的 BAG 参数外，为了实现一些特定的仿真测试需求，额外提供了扩展的可设置的 BAG 时间，支持 100us~1ms。设置非标准的最小 BAG 时间时会由相应的前置条件，具体的设置方法和条件可参见《API 函数使用手册》。

4. 安装

使用 AFDX 仿真卡，需要将模块正确的安装到用户的计算机或指定设备中，然后正确安装模块对应的驱动或者软件。

4.1. 安装准备

- 1) 在操作硬件模块前应当避免静电放电损害；
- 2) 在第一次使用前，应检查模块外观是否正常，确保运输过程未对设备或板卡造成损伤，如果发现损伤请立即联系我们；
- 3) 用于应请根据模块的总线类型提供相应的安装环境。

4.2. 硬件安装

- 1) 硬件安装前应当确保计算机或计算机设备的电源处于关闭状态；
- 2) 将板卡插入对应的槽位，并通过前面板的螺钉进行固定；
- 3) 待板卡固定到位后，再将配套的电缆插入到板卡的外部连接器接口并完成锁紧固定；
- 4) 开启计算机，等待计算机自动发现设备，首次使用时需要进行驱动安装。

4.3. 驱动安装

4.3.1. Windows 系统

- 1) 在进行驱动安装前应当确保硬件板卡已被正确安装。
- 2) 在板卡配套的发布包中“driver”文件中可以找到对应系统的驱动文件。
- 3) 可以用过设备管理器方式进行手动安装，手动安装过程如下：



图 8 驱动安装图 1

第一步：找到设备，右键选择“更新驱动程序”，选择“浏览计算

机以查找并安装驱动程序软件”。

第二步：选择 “让我从计算机上的可用驱动程序列表中选择”。

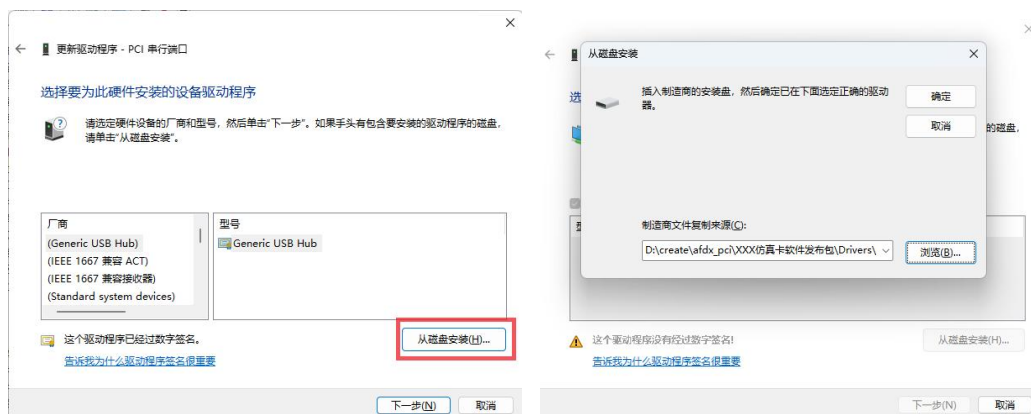


图 9 驱动安装图 2

第三步：选择 “从磁盘安装”。

第四步：点击 “浏览” 找到驱动程序存放路径，并选择驱动程序 “xxxx.inf” 后点击 “确定” 按钮。

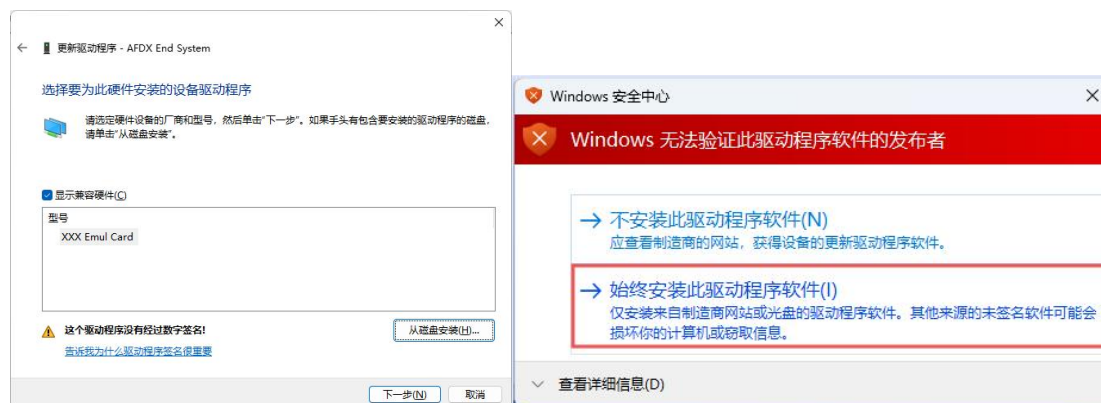


图 10 驱动安装第六步

第五步：选择 “AFDX-xxxx” 后点击 “下一步”。

第六步：选择 “始终安装次驱动程序软件”。根据提示提示驱动程序更新/安装完成。



图 11 驱动安装完成

- 4) 驱动正常安装后可在设备管理器列表中查看到相应的板卡信息。



图 12 驱动安装完成后显示状态

4.3.2. Linux 系统

Linux 系统下，USB 总线的 AFDX 仿真卡一般无需安装驱动。如果再使用时提示未找到设备，可按照对应 Linux 版本路径下的“/driver/USB/readme”文件中的说明进行文件复制。

PCI/CPCI/PXI 或 PCIe/CPCIe/PXle 总线的仿真卡需要安装 bus_afdx.ko 驱动，文件存放在对应 Linux 版本路径下的“/driver/Pcie”文件夹中。单次生效可以使用 `sudo insmod bus_afdx.ko`；需要永久生效时，用户可以自行将驱动添加到设备开始启动的脚本中。

注：在 Linux 环境下开发包提供的库为通用库，同时兼容 USB 和 PCIE、PCI 总线类型。因此，在 linux 环境下 PCI、PCIe 总线类型的 AFDX 仿真卡也需要安装 libusb 库，具体说明可参见第 7 节的内容。

4.4. 软件安装

4.4.1. Windows 系统

AFDX 仿真卡提供两种类型的软件，一种为控制台 demo 程序，另外一种为总线仿真软件。

控制台 demo 软件无需安装，存放在发布文件包中的“demo”文件夹，可直接运行。控制台 demo 程序提供软件源代码，提供板卡操作的多种 demo 例程，用户可以用过对应的 demo 软件实现硬件的快速操作以及作为软件设计的参考用例。

总线仿真软件提供 UI 界面，用户可以通过软件实现对硬件板卡功能进行操作，此软件不提供源码，直接点击“总线仿真软件”文件夹中的“AFDX 仿真软件安装程序 Vx.xx(64bit/32bit).exe”可自动完成软件安装。

4.4.2. Linux 系统

控制台 demo 程序提供软件源代码，提供板卡操作的多种 demo 例程，用户可以用过对应的 demo 软件实现硬件的快速操作以及作为软件设计的参考用例。

总线仿真软件提供 UI 界面，用户可以通过软件实现对硬件板卡功能进行操作，此软件不提供源码，在终端中使用” sudo ./MIL1394B_WARE” 指令进行运行，运行需要管理员权限。

仿真软件需要 libusb 库的支持，libusb 库的安装流程见第 7 章 linux 系统运行环境搭建。

4.5. 线缆安装

AFDX 仿真卡在正常通信前需要确保线缆正确接入，在接入线缆时需要注意按照 NETA/B 网络标识。

当仿真卡工作在冗余模式时需要将冗余通道的 NETA(NET0/NET2) 和 NETB (NET1/NET3) 网络分别接入对应设备的 NETA 和 NETB 网络;当仿真卡工作在独立模式时将通道接入对应设备的 NETA 或 NETB 网络。

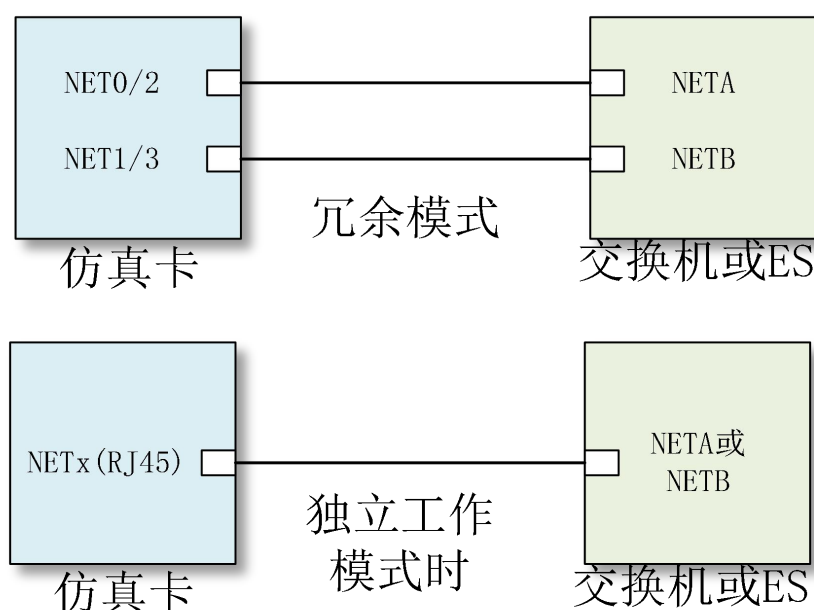


图 13 硬件接口连接示意图

5. 控制台 demo 程序使用

5.1. 使用前准备

在使用控制台 demo 之前请确保板卡硬件模块已正确插入到对应槽位，固定到位，且能够正确上电并完成驱动安装。

由于某些 demo 用例会使用到多个通道，且需要外部线缆接入，

请确保 demo 运行前线缆已正确接入。

5.2. 控制台 Demo 软件列表

表 1 控制台 demo 列表

序号	文件名	描述
1	AfdxApi_ConstructDemo	超长包发送和接收例程
2	AfdxApi_EdeDemo	AFDX EDE 功能收发例
3	AfdxApi_ICMP_BusMonDemo	ICMP 发送, 顺序监控方式接收例程
4	AfdxApi_ICMPXmt_VIRcvDemo	ICMP 发送, VL 方式接收例程
5	AfdxApi_Int_Bus_Demo	顺序监控中断例程
6	AfdxApi_Int_VLMon_Demo	VL 监控中断例程
7	AfdxApi_Loop2048VICycleSche1BusRcvDemo	2048 条 VL 循环发送, 调度表采用调度方式 1, 顺序监控方式接收例程
8	AfdxApi_Loop2048VICycleSche4Demo	2048 条 VL 循环发送, 调度表采用调度方式 4, 顺序监控方式接收例程
9	AfdxApi_LoopDualModeDemo	冗余通道模式收发例程
10	AfdxApi_LoopErrInsertDemo	错误注入和软件生成指定数据例程
11	AfdxApi_LoopMulPortSendLastDemo	单 VL 多端口 SENDLAST(发送最后一帧)发送例程
12	AfdxApi_LoopMulVLSendLastDemo	多 VL 单端口 SENDLAST(发送最后一帧)发送例程
13	AfdxApi_LoopMulVLSendonceDemo	多 VL SENDONCE 例程
14	AfdxApi_LoopReadMsgDemo	使用消息读取函数读取数据的例程
15	AfdxApi_VLIntDemo	VL 模式中断接收和发送例程
16	AfdxApi_WriteMultMsgD	VL 发送模式一次性写入多条数据

	emo	的例程
17	AfdxApi_XmtCycle500us Demo	VL 发送时，更改最小时间间隔为 500us 的例程

5.3. Demo 程序使用

用户直接选择相应的 demo 例程编译执行即可，根据不同的 demo 程序打印相应的信息。用户可以根据打印信息判断执行的过程和结果。

6. 总线仿真软件使用

6.1. 使用准备

在使用仿真软件之前请确保板卡硬件已正确插入到指定槽位，固定到位，且能够正确上电并完成驱动安装。

在运行软件前请确保外部线缆已正确接入。

6.2. 软件功能

- 通用功能
 - 网络链接状态
 - 网络速度设置
 - VL 收发消息统计
 - 接收/发送启动
- 消息发送功能
 - 发送 ES 目的地址设置
 - 发送 VL 创建和删除
 - 发送端口创建功能
 - 发送模式选择
 - 发送 VL 参数设置
 - 发送 VL 错误注入
 - 发送端口错误注入
 - 发送数据设置，支持多 buffer
- 消息接收功能
 - 接收 ES 目的地址设置
 - 接收 VL 创建和删除
 - 接收模式选择
 - 接收 VL 参数设置
 - 接收错误检测

■ 接收端口创建功能

6.3. 使用说明

6.3.1. 基本流程

仿真卡的使用需要遵循一些基本且必要的流程，基本的操作流程如下如所示。

板卡连接-通道模式设置-创建 VL-编辑 VL 参数-消息编辑（需要发送的话）-启动接收/发送-处理接收数据（如果需要的话）-更新发送数据（如果需要的话）-操作完成停止。

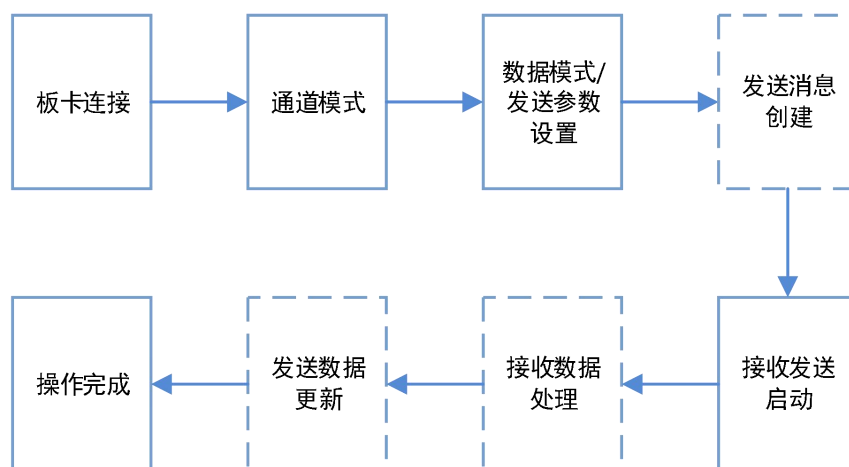


图 14 基本操作流程

6.3.2. 板卡连接

软件运行后将自动识别出系统中的 AFDX 仿真卡，选择需要连接的设备并点击“OK”连接设备。

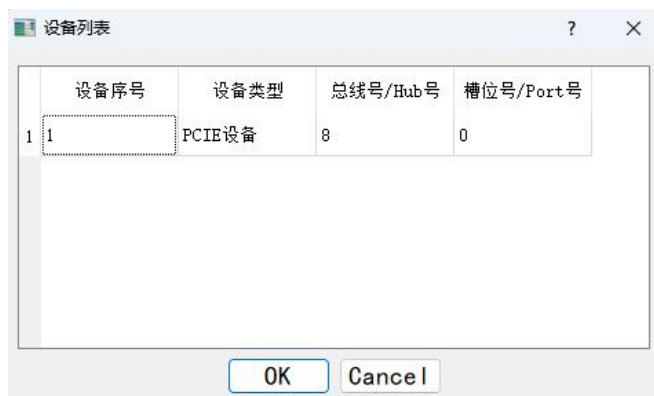


图 15 设备列表

设备连接成功后将进入程序主界面。程序主界面可显示各通道链接状态，用户可以在主界面中通过对应的操作界面进行功能设置和数据收发操作。

6.3.3. 仿真软件主界面

软件主界面可分为三个个区域，分别为通道控制区、工作区和状态显示区。



图 16 软件主界面

6.3.3.1.通道控制区

通道控制区可以对通道进行选择同时查看网络状态。



图 17 通道控制区界面

点击指定通道后将会激活对应通道的工作区。

在冗余通道模式下会显示网络 NETA 和 NETB 的状态；在独立通道模式下，则只显示单个网络状态。

通道号与网络接口对应关系如下表所示。

通道模式与通道号关系		
网络接口	冗余模式	独立模式
NET0	CH0-NETA	CH0
NET1	CH0-NETB	CH1
NET2	CH2-NETA	CH2

NET3	CH2-NETB	CH3
------	----------	-----

6.3.3.2.工作区

工作区界面如下图所示，可进行通道参数配置、发送和接收功能操作。

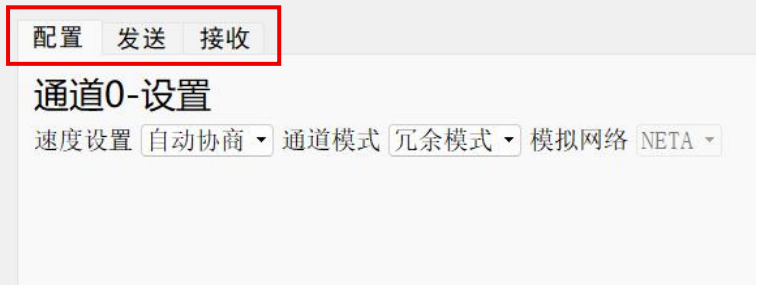


图 18 工作区界面

配置界面

配置界面可以进行速度和通道模式配置。

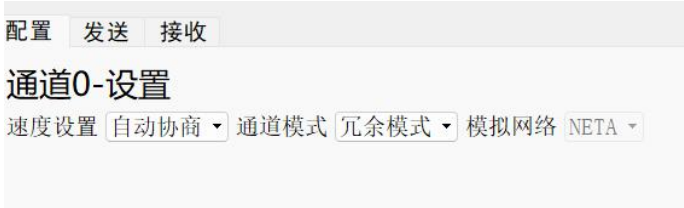


图 19 配置功能界面

速度设置可以选择 10M、100M、1000M 和自动协商；

通道模式可以选择独立通道模式和冗余通道模式。如果工作在独立通道模式时，可以选择模拟网络 A 或模拟网络 B。在通道模式切换以后，左侧的通道状态将随之匹配。

发送界面

发送界面可以进行发送 VL、PORT 的创建和管理，可以对 ES 目的地址设置，可以选择发送调度模式。



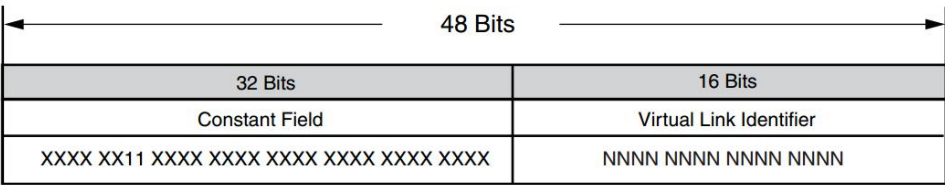
图 20 发送管理界面

发送管理界面可以创建发送 VL 和 PORT（即子 VL），可以对 ES 目的地址设置和调度模式进行设置，同时可以查看网络发送数据统计

以及发送控制。

1) MAC 目的地址设置

MAC 目的地址(即 ES 目的地址)由用户设置。该处设置的是 48 位 MAC 目的地址中的高 32 位。根据 AFDX 协议规定，48 位 ES 目的地址中的第 40 和 41 位必须为 1。



2) 调度模式设置

可以设置 4 种 BAG 发送调度模式。分别为调度模式 1~4。其中调度模式 1 为突发模式，调度模式 4 为均衡模式。调度模式 2 和调度模式 3 在两者之间。

一般建议使用调度模式 4 和调度模式 3, 如果发送的 VL 数量较多时请避免使用调度模式 0。

4 种调度的具体介绍和区别可以参见《API 软件参考手册》种的详细说明。

3) 创建和编辑发送 VL

当点击“创建 VL”时会弹出 VL 编辑界面。该界面可以配置 VL 的相关参数，包括 MAC 源地址、VLID、IP 源地址、冗余模式、BAG 值、LMax、Skew。

注意：仿真软件界面最多支持 64 条接收 VL 的创建。



图 21 状态显示区界面

MAC 源地址控件可以设置 MAC 源地址中的用户设置域(共 16bit, 16 进制设置)。

VLID 和 IP 源地址采用十进制设置方式，其中 VLID 设置范围为 0~65535。

发送冗余模式可以选择仅 NETA、仅 NETB 或冗余发送。

BAG 可以选择 1~128，单位为 ms。

LMax 为该 VL 最大发送的以太网帧长度可设置 256~1518.当用户

发送的数据超过此长度后，将进行 IP 分包传输。

4) 创建和编辑发送 PORT

在 VL 创建完成后，需要对该 VL 创建至少一个 PORT 才能进行发送。在发送界面选中 VL 后，可点击“创建 PORT”按钮进行端口的创建。单个 VL 最多可创建 4 个 PORT。

端口编辑对话框可以设置 PORT 的相关参数，包括端口类型、UDP 源地址、UDP 目标地址、Buff 大小（深度）、Buff 个数和发送模式。



图 22 发送端口编辑界面

端口类型：支持采样端口（Sampling）、队列端口（Queuing）、服务器端口（SAP）。

UDP 源地址：十进制方式设置，在 SAP 端口下无效。

UDP 目标地址：十进制方式设置。

Buff 大小：可设置 256~8192，单位为字节。用户发送的数据长度不应超过该值。

Buffer 个数：设置发送的 buff 个数，可设置 1~64；

发送方式：支持循环（sendcycle）、发送最新数据（sendonce）以及重复发送上一次的数据（sendlast）。**注意：在仿真软件中如果发送模式为 sendonce 的 port，在启动发送后该 PORT 仅会发送单次数据。**

用户设置完成后点击“OK”按钮进行端口创建。

在端口创建完成后，需要对发送的数据进行编辑。首次创建 PORT 时将自动弹出送数据编辑界面，对已创建的 PORT 进行数据更改时可在主界面中点击对应 PORT 位置的“点击编辑数据”按钮进行数据编辑。



图 23 发送 VL 编辑完成界面

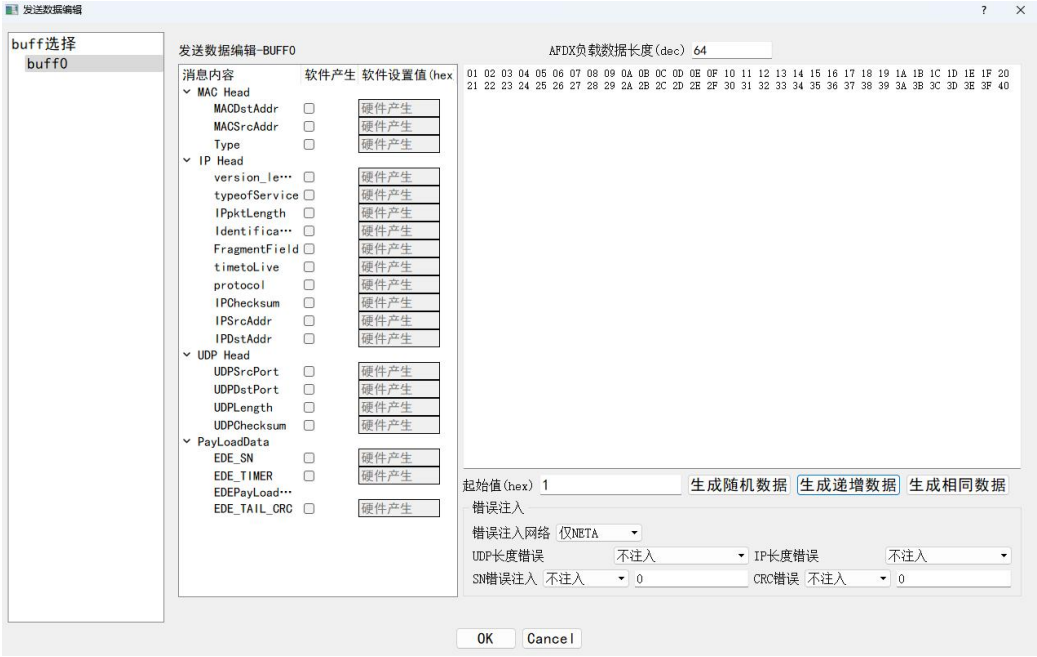


图 24 发送数据编辑界面

在发送数据编辑界面中，若有多个 buffer，需要选中对应的 buffer 后再进行数据标记。

发送数据可以对 AFDX 帧的全域进行设置，包括 MAC、IP、UDP、SN、CRC 以及 AFDX 负载数据。

除 AFDX 负载外，其余区域默认均由硬件自动产生。如果需要对指定的数据域进行编辑可以勾选“软件产生”复选框，然后输入需要设置的值即。设置值为 16 进制。

AFDX 负载数据在确认好长度后（不能大于 buffersize），可以选择生成随机数据、递增数据和相同数据。用户自行在编辑框中更改的数据不会生效，此编辑框的数据仅进行展示。

故障注入可以进行注入网络的选择、UDP 长度、IP 长度、SN 和 CRC 的错误注入。其中 SN 和 CRC 可以选择使用软件值，此时在后续的编辑框中输入需要设置的值即可。

5) 网络发送数据统计

启动发送后可以查看 NETA 和 NETB 发送的数据统计。在通道独立工作模式时，将只会通过 NETA 的发送数据统计进行显示。

6) 发送控制

用户点击开始发送将进行数据发送，再次点击即可停止数据发送。

接收界面

接收界面与发送界面类似，可以进行接收模式的选择，接收 VL、PORT 的创建和管理。



图 25 接收管理界面

1) 接收模式选择

接收界面可以进行接收模式的选择，支持 VL 接收模式、VL 监控模式和顺序监控模式。

VL 模式和 VL 监控模式，都需要创建接收 VL 和 PORT 后才能进行数据接收。

顺序监控模式下，可以直接启动数据接收。若用户需要指定一些 VL 进行监控时，可以点击“创建 VL”，在弹出的对话框中输入想要监控的 VLID 即可。

2) 创建和编辑接收 VL

接收 VL 的创建与发送相似，点击“创建 VL”按钮可以进行 VL 的创建。

注意：仿真软件界面最多支持 64 条接收 VL 的创建。



图 26 接收 VL 编辑界面

MAC 源地址控件可以设置 MAC 源地址中的用户设置域(共 16bit, 16 进制设置)。

VLID 和 IP 源地址采用十进制设置方式, 其中 VLID 设置范围为 0~65535。

接收冗余模式可以选择冗余接收、仅 NETA 接收、仅 NETB 接收以及 NETA 和 NETB 均接收。

SkewMax: 单位 ms。设置 VL 的 skewmax 参数。Skewmax 的参数说明可参见《API 软件参考手册》中的相关说明, 该值必须大于 bag 值, 若不清楚该参数含义建议设置为 1000ms 或设置为 0。若设置为 0 将不启用硬件 skewmax 判断。

刷新率: 单位 us。用于判断 VL 的两个数据帧之间的刷新率是否符合条件的判断条件。若接收到的 VL 刷新率大于或低于该值, 再接收数据中会置起相应标记, 仿真软件默认设置为 1000us。

容错使能: 可以设置接收错误帧, 打开指定的错误容错使能, 接收对应的错误帧并再接收数据区中进行显示。

3) 创建和编辑接收 PORT

在 VL 创建完成后, 需要对该 VL 创建至少一个 PORT 才能进行数据接收。在主面选中 VL 后, 可点击“创建 PORT”按钮进行端口的创建。单个 VL 最多可创建 4 个 PORT。

端口编辑对话框可以设置 PORT 的相关参数, 包括端口类型、UDP 源地址、UDP 目标地址、Buff 个数。

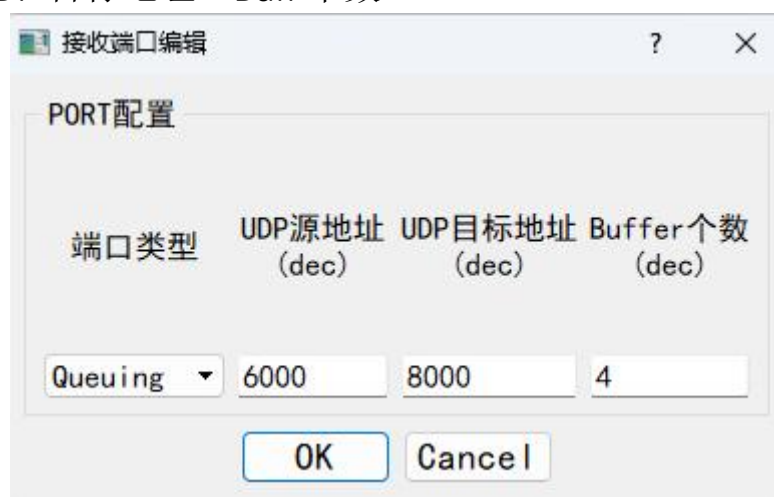


图 27 接收端口编辑界面

端口类型: 可以选择采样端口 (Sampling)、队列端口 (Queuing) 和服务端口 (SAP);

UDP 源地址: 采用十进制编辑方式, **SAP 端口下无效**;

Buffer 个数: 用户可以设置接收 Buffer 个数。

设置完成端口后点击“OK”按钮完成编辑，接收 VL 编辑完成界面如下图所示：

4) 数据查看界面

完成 VL 和 PORT 的创建，启动并接收到数据后。可以通过主界面的“查看所有数据”和对应 PORT 的“查看 PORT 数据”进行数据查看。

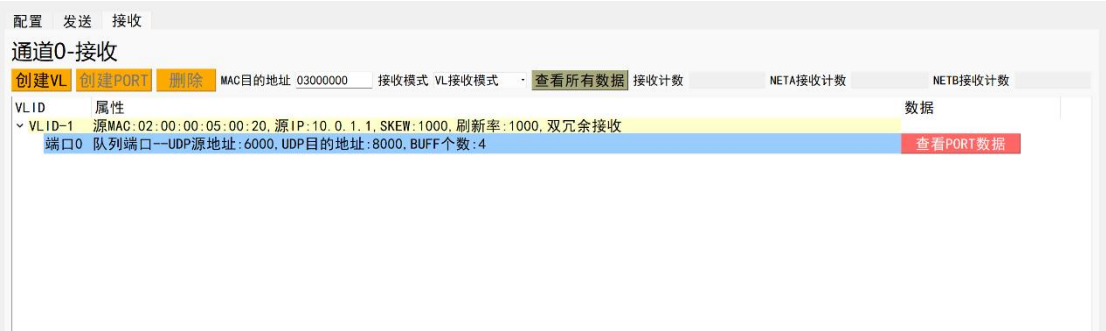


图 28 接收 VL 编辑完成界面

“查看所有数据”按钮可以查看所有接收到的数据，对应 PORT 的“查看 PORT 数据”按钮则可以查看对应 PORT 的接收数据。

每次点击“查看所有数据”或“查看 PORT 数据”按钮将自动清除缓存中的数据。如果需要检查启动开始时刻的数据，可以先点击“查看数据”或“查看 PORT 数据”按钮打开监控数据界面后，再开始接收数据。

在接收数据查看界面，左侧会显示所有数据的概要信息。点击指定信息后，可以查看数据的详细内容，并进行协议解析。协议解析只针对 UDP 协议，其它协议仅解析到 IP 协段。

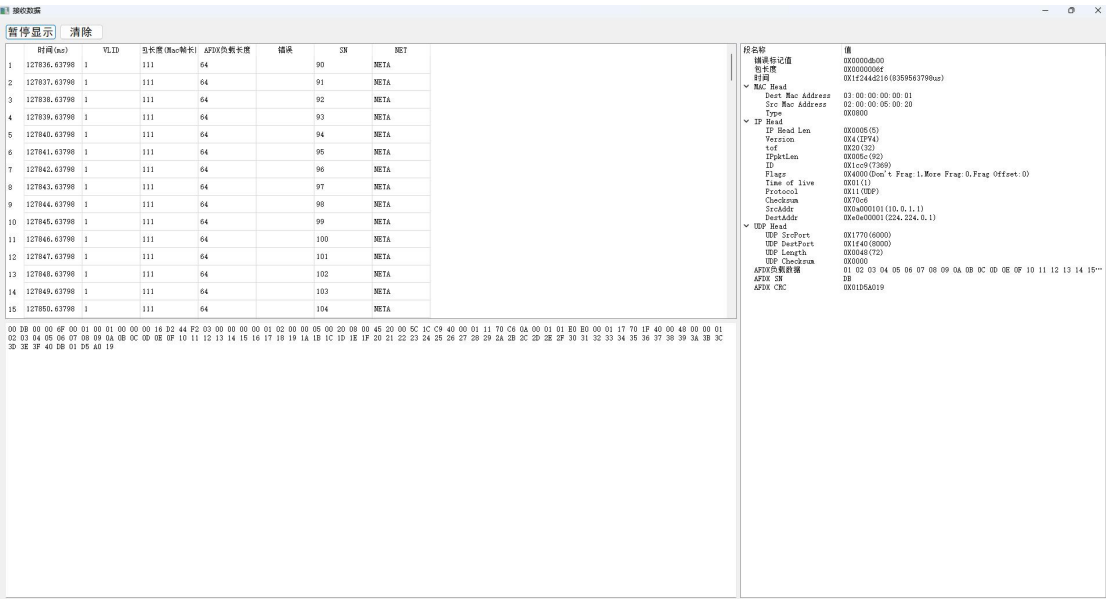


图 29 接收数据查看界面

6.3.3.3. 状态显示区

状态显示区主要进行当前操作状态的异常状态显示，当有错误时，将在此提示，请用户注意进行查看以便进行故障排查。



图 30 状态显示区界面

6.3.4. 数据发送使用说明

仿真卡的发送基本使用流程如下。

- 1) 从主界面选择“发送”
- 2) 在界面选择发送调度表模式
- 3) 在界面选择“创建 VL”，完成 VL 的创建
- 4) 点击“创建 PORT”，完成 PORT 的创建
- 5) 编辑发送数据
- 6) 开始发送

6.3.5. VL 接收模式接收使用说明

VL 接收模式下的基本使用流程如下。

- 1) 从主界面选择“接收”
- 2) 选择 VL 接收模式
- 3) 点击“创建 VL”，完成 VL 的创建
- 4) 点击“创建 PORT”，完成 PORT 的创建
- 5) 启动接收

6.3.6. VL 监控模式接收使用说明

VL 监控模式使用流程与 VL 接收模式基本一致。

- 1) 从主界面选择“接收”
- 2) 选择 VL 监控模式
- 3) “创建 VL”，完成 VL 的创建
- 4) 启动接收

6.3.7. 顺序监控模式接收使用说明

顺序监控接收模式下的基本使用流程如下。

- 1) 从主界面选择“接收”
- 2) 选择顺序监控模式

3) 启动接收

注意：在启动接收前，如果需要的话可以点击“创建 VL”选择想要指定监控的 VL，否则默认为全监控。

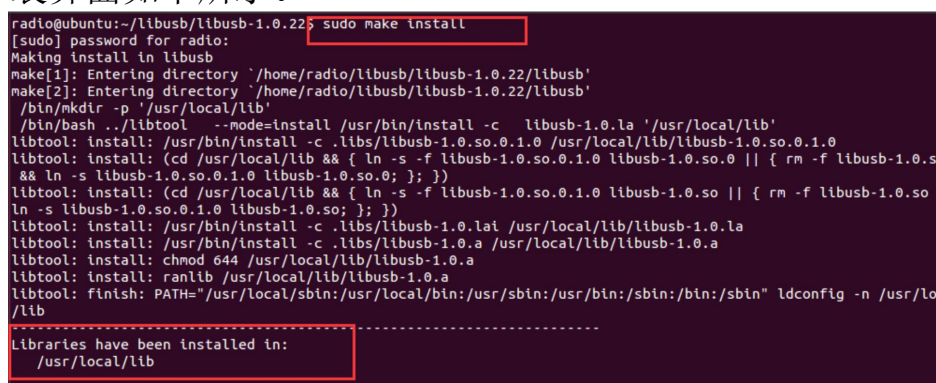
7. Linux 环境下 libusb 的安装

注意：在 Linux 环境下仿真卡提供的库为通用库，可同时兼容 USB 和 PCIE、PCI 总线类型。因此，在 linux 环境下 PCI、PCle 总线类型的 AFDX 仿真卡也需要安装 libusb 库。

在 linux 环境下(例如 ubuntu、麒麟(kylin)等系统)，总线仿真卡需要 libusb 库才能正常使用。建议 libusb 版本为 libusb-1.0.28 或以上。若用户系统中已经安装或者集成了 libusb 库，则无需再次进行添加，可直接使用仿真软件或使用 AFDX 库文件进行二次开发。如果用户的使用环境中没有安装该库，那么可以从发布包中的 <libusb-1.0.28> 文件夹中找到该库的压缩包，按照后续程进行编译和安装。

7.1. Libusb 的编译与安装

1. 在“软件发布包_linux_Vx.xx / libusb-1.0.28”文件夹中解压 libusb-1.0.28.tar.gz；
2. 使用终端，进入“软件发布包_linux_Vx.xx”文件夹路径下，在“软件发布包_linux_Vx.xx/libusb-1.0.28/libusb-1.0.28”文件夹中打开终端；
3. 在终端中输入“./configure --disable-udev”进行工程配置；
4. 配置完成后使用“make”指令进行工程编译；
5. 待编译完成后，使用“sudo make install”进行库的安装，安装界面如下所示。



```
radio@ubuntu:~/libusb/libusb-1.0.22$ sudo make install
[sudo] password for radio:
Making install in libusb
make[1]: Entering directory '/home/radio/libusb/libusb-1.0.22/libusb'
make[2]: Entering directory '/home/radio/libusb/libusb-1.0.22/libusb'
/bin/mkdir -p '/usr/local/lib'
/bin/bash ./libtool --mode=install /usr/bin/install -c libusb-1.0.la '/usr/local/lib'
libtool: install: /usr/bin/install -c .libs/libusb-1.0.so.0.1.0 /usr/local/lib/libusb-1.0.so.0.1.0
libtool: install: (cd /usr/local/lib && { ln -s -f libusb-1.0.so.0.1.0 libusb-1.0.so.0 || { rm -f libusb-1.0.s
&& ln -s libusb-1.0.so.0.1.0 libusb-1.0.so.0; }; })
libtool: install: (cd /usr/local/lib && { ln -s -f libusb-1.0.so.0.1.0 libusb-1.0.so || { rm -f libusb-1.0.so
ln -s libusb-1.0.so.0.1.0 libusb-1.0.so; }; })
libtool: install: /usr/bin/install -c .libs/libusb-1.0.lai /usr/local/lib/libusb-1.0.la
libtool: install: /usr/bin/install -c .libs/libusb-1.0.a /usr/local/lib/libusb-1.0.a
libtool: install: chmod 644 /usr/local/lib/libusb-1.0.a
libtool: install: ranlib /usr/local/lib/libusb-1.0.a
libtool: finish: PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin" ldconfig -n /usr/lo
/lib
-----
Libraries have been installed in:
  /usr/local/lib
```

图 31 libusb 的安装

出现如上图中的提示，表明 libusb 的库已被正确安装。用户需

要留意安装路径，以备使用。

7.2. Libusb 库的使用

在没有使用第三方开发环境时，用户可直接链接 libAFDX_API.a 和 libusb-1.0.so 库进行编译即可。

注意： Demo 下的默认 Makefile 进行编译要求 libusb-1.0.so 库需要在 `usr/local/lib` 目录或 `/usr/lib` 目录下，libAFDX_API.a 在 Demo 上一级的 AFDX_API 目录下才能编译成功，否则用户需要根据自己的路径对环境变量和路径进行链接后才能编译。

如果在使用了第三方开发环境时，用户需要手动进行库的添加，以下以 Qt 工程为例。

用户在新建工程之后。如下图所示，通过右键工程，选择“添加库…”。

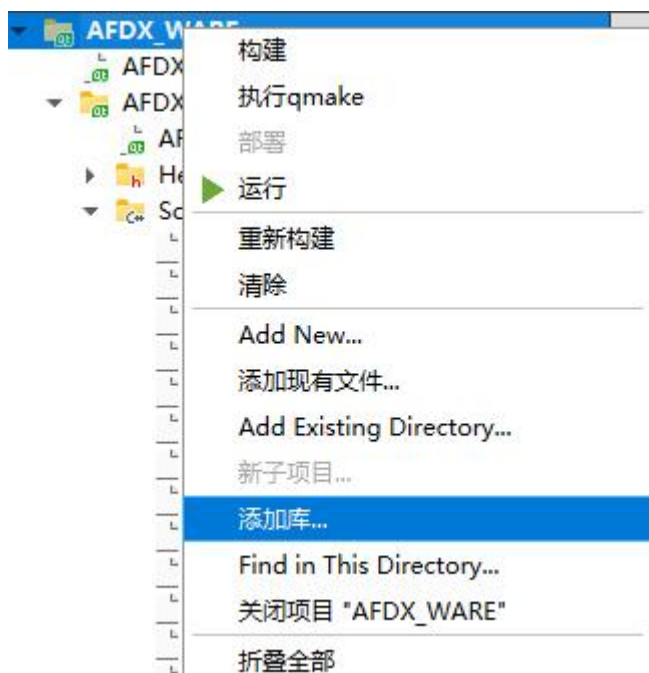


图 32 libusb 的使用步骤 1

选择“外部库”如下所示；

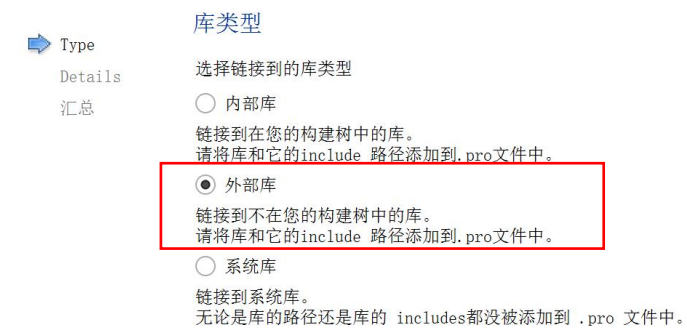


图 33 libusb 的使用步骤 2

然后选择“发布文件”夹中的“libAFDX_API.a”文件。直到完成。

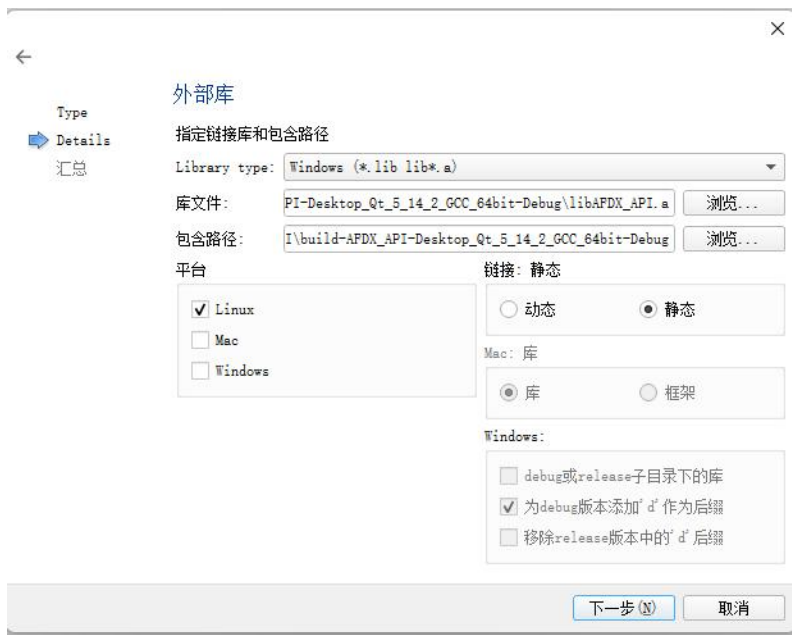


图 34 libusb 的使用步骤 3

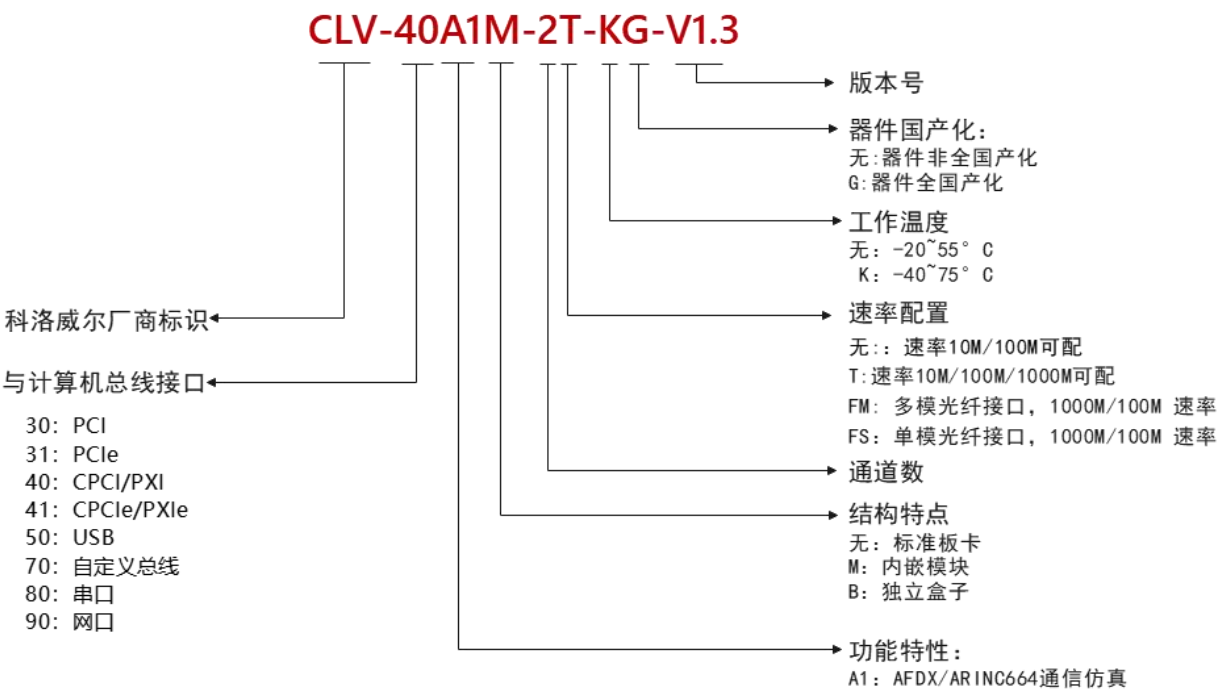
添加成功后，可按照上述步骤继续添加 libusb.so 动态库，libusb 路径为安装时选择的路径。

8. 故障维护

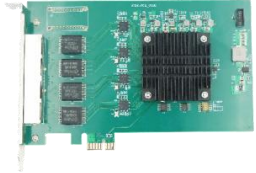
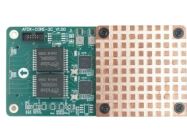
模块无法正常工作时，按照以下步骤进行检测。

- 1) 检查电源指示灯是否正常点亮；
- 2) 查看各网口状态指示灯是否异常；
- 3) 检查线缆连接状态是否正常；
- 4) 若以上检测均无异常则联系售后人员处理。

9. 选型指南



10. 选型示例

板卡型号	总线类型	参数配置	实物图
CLV-31A1-4T-V1.00	PCIe 总线	4 个网口，速率支持 10M，100M，1000M，3U 标准卡，工作温度 -20~55℃，器件非国产化，版本 V1.00	
CLV-41A1-4T-KG-V1.00	CPCle 总线	4 个网口，速率支持 10M，100M，1000M，3U 标准卡，工作温度 -40~75℃，器件国产化，版本 V1.00	
CLV-40A1-4-V1.00	CPCI 总线	4 个网口，速率支持 10M，100M，3U 标准卡，工作温度 -20~55℃，器件非国产化，版本 V1.00	
CLV-50A1B-4T-V1.00	USB 总线	4 个网口，速率支持 10M，100M，1000M，独立盒子，工作温度 -20~55℃，器件非国产化，版本 V1.00	
CLV-31A1M-2T-KG-V1.00	PCIe	2 个网口，速率支持 10M，100M，1000M，内嵌模块，工作温度 -40~75℃，器件国产化，版本 V1.00	
CLV-40A1-4FM-V1.00	CPCI 总线 光口	4 个多模光纤接口，速率支持 100M，1000M，内嵌模块，工作温度 -20~55℃，器件非国产化，版本 V1.00	