UTC2208

用户手册

类别	内容
关键词	UTC2208、六叶树、用户手册
摘要	UTC2208 是六叶树一款 USB 接口的 SocketCAN 适配器，集成 8 路 CAN 接口，支持 CAN2.0 模式，波特率范围:5K-1M。

文档记录

版本	日期	说明
V1.00	2025.12	创建文档

目录

1.功能简介

1.1 产品概述

UTC2208 是六叶树一款 USB 接口的 SocketCAN 适配器，集成 8 路 CAN 接口，支持 CAN2.0A 和 CAN2.0B 协议，符合 ISO/DIS 11898-1/2/3 标准，采用 USB2.0 接口，与 USB1.1\USB3.0 总线兼容，支持 5Kbps~1Mbps 的波特率，自带匹配电阻，可以通过开关进行控制，支持主流的 linux 操作系统，支持 X86、ARM 架构。

1.2 产品外观

1.3 参数指标

名称	六叶树 UTC2208 适配器	**尺寸**	17×12×3.0cm
调试软件	can-utils 工具(支持命令安装+源码安装)	**适用系统**	Linux
应用范围	雷达、机器人开发等	**二次开发**	SOCKET 接口(提供 C++代码用例)
功能	SOCKETCAN 适配器	**CAN 波特率**	5K-1M(支持自定义)
应用平台	X86/64/ARM 等	**驱动安装**	Linux
USB 接口	USB2.0 高速 480M/S	**CAN 标准**	CAN2.0
隔离类型	电源和信号双隔离	**通道数量**	8×CAN2.0
应用平台	OrangePi/树莓派/友善之臂/东凌嵌入式/瑞芯微/英伟达等。	**匹配电阻**	内置 120 欧姆(可开关)
		外观材质	铝合金

1.4 主要功能

1.平台通过 USB 接口将 CAN 数据报文发送到 CAN 总线。
2.将 CAN 总线上的 CAN 数据报文通过 USB 接口采集到平台。
3.ROS 机器人系统支持。

1.5 软件支持

● CAN 通讯工具 can-utils,点击查看详情。

apt-get install can-utils

1.6 二次开发

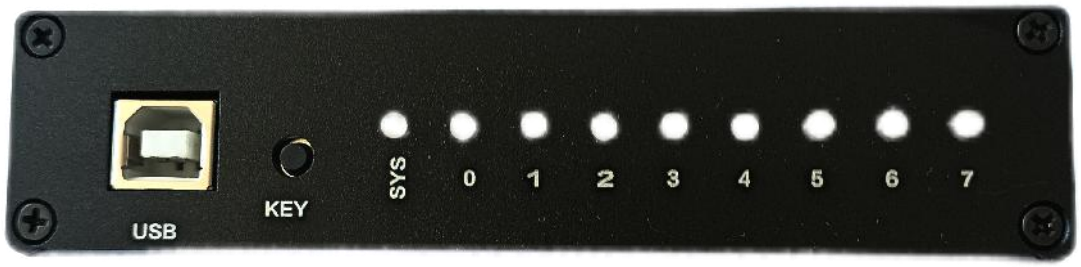
二次开发代码用例，点击下载

1.7 典型应用

● 机器人控制
● 汽车雷达调试

2.接线说明

2.1 前面板接口



2.1.1 USB

通过 USB 线接到电脑 USB 口，起供电和通讯的作用。

2.1.2 KEY

功能键，保留。

2.1.3 指示灯

指示灯	状态	
SYS	设备运行指示灯	红灯: 常亮:设备未连接电脑 闪烁:USB 故障 绿灯: 常亮:设备运行正常
0	指示 CAN0 通道状态	
1	指示 CAN1 通道状态	
2	指示 CAN2 通道状态	
3	指示 CAN3 通道状态	
4	指示 CAN4 通道状态	
5	指示 CAN5 通道状态	
6	指示 CAN6 通道状态	
7	指示 CAN7 通道状态	

CAN0-7 灯状态定义:

绿灯	常亮:通道开启 常灭:通道关闭 闪烁:CAN 总线有数据正在进行	
红灯	常灭:无故障 常亮/闪烁:CAN 总线通讯故障	故障排除: 1.检查波特率是否正确 2.检查 CAN 总线接线是否正确 3.检查 CAN 总线是否具备匹配电阻

2.2 后面板接口

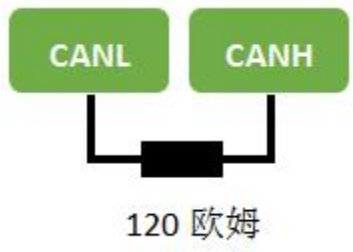
2.2.1 接线端子

名称	定义
CAN0-H	CAN0 通道 H 线
CAN0-L	CAN0 通道 L 线
CAN0-R	CAN0 通道匹配电阻控制线
CAN1-H	CAN1 通道 H 线
CAN1-L	CAN1 通道 L 线
CAN1-R	CAN1 通道匹配电阻控制线
CAN2-H	CAN2 通道 H 线
CAN2-L	CAN2 通道 L 线
CAN2-R	CAN2 通道匹配电阻控制线
CAN3-H	CAN3 通道 H 线
CAN3-L	CAN3 通道 L 线
CAN3-R	CAN3 通道匹配电阻控制线
GND	地
CAN4-H	CAN4 通道 H 线
CAN4-L	CAN4 通道 L 线

CAN4-R	CAN4 通道匹配电阻控制线
CAN5-H	CAN5 通道 H 线
CAN5-L	CAN5 通道 L 线
CAN5-R	CAN5 通道匹配电阻控制线
CAN6-H	CAN6 通道 H 线
CAN6-L	CAN6 通道 L 线
CAN6-R	CAN6 通道匹配电阻控制线
CAN7-H	CAN7 通道 H 线
CAN7-L	CAN7 通道 L 线
CAN7-R	CAN7 通道匹配电阻控制线
GND	地

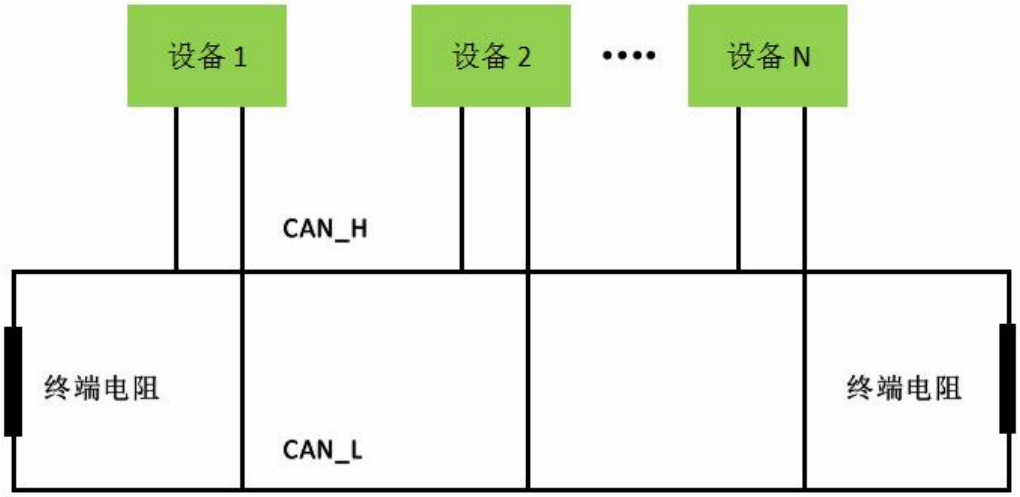
匹配电阻控制线用法:
用导线将 CANx-R 连接到 CANx-H,即可开启 CANx 的匹配电阻，一个阻值 120 欧姆将并联到 CANx 的 CANH 和 CANL 线上。

效果如下图:



2.3 总线连接

UTC2208 适配器和 CAN-bus 总线连接的时候，仅需要将 CAN_L 连 CAN_L,CAN_H 连 CAN_H 信号。 CAN-bus 网络采用直线拓扑结构，总线的 2 个终端需要安装 120 Ω 的终端电阻；如果节点数目大于 2 ，中间节点不需要安装 120 Ω 的终端电阻。对于分支连接，其长度不应超过 3 米。CAN-bus 总线的连接见下图。



注意：CAN-bus 电缆可以使用普通双绞线、屏蔽双绞线。若通讯距离超过 1Km，应保证线的截面积大于Φ 1.0mm2，具体规格应根据距离而定，常规是随距离的加长而适当加大。

3.使用教程

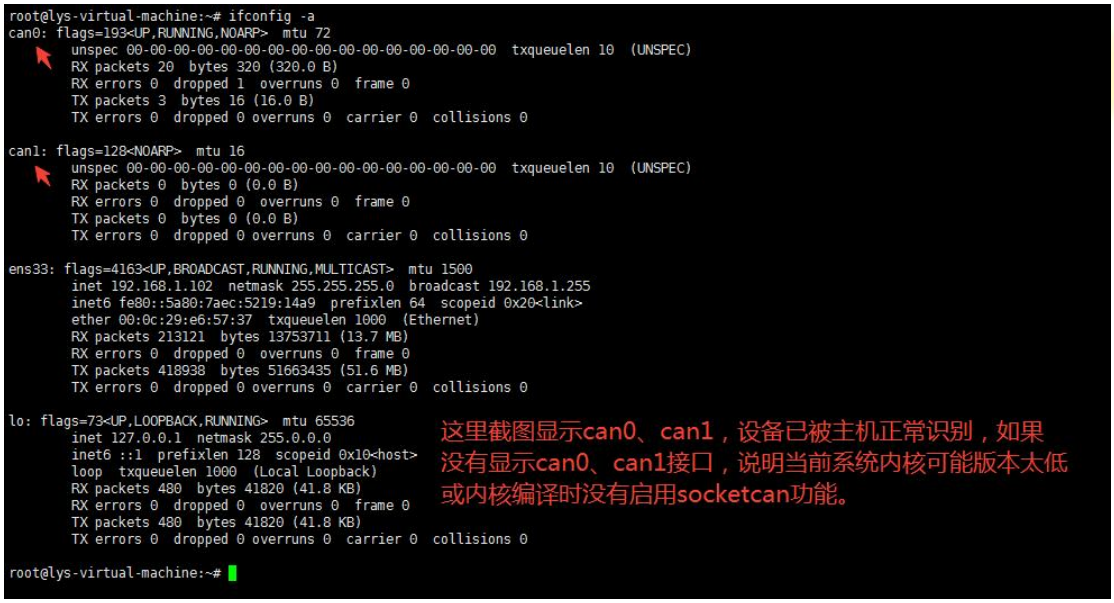
3.1 主机连接及驱动安装

1.通过配套的 USB 线，将 UTC2208 连接到主机(电脑或开发板)。

2.安装好对应平台的 socketcan 驱动。点击下载驱动。

注意事项:所有命令都使用管理员或 root 权限执行。

通过”ifconfig -a”查看设备是否被主机识别，如果能看到 can0、can1 接口代表设备已被主机正确识别。



3.2 软件(can-utils)安装(可选项，非必须)

can-utils 工具非必须安装，该工具只是用来进行简单的 CAN 数据收发测试，自己开发的 socketcan 应用程序不会依赖该工具,如果安装，也只需要安装一次即可，不用每次系统重启都安装。

安装命令:apt-get install can-utils

```
root@ubuntu:/home/zhumiao# apt-get install can-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  can-utils
0 upgraded, 1 newly installed, 0 to remove and 255 not upgraded.
Need to get 0 B/117 kB of archives.
After this operation, 656 kB of additional disk space will be used.
Selecting previously unselected package can-utils.
(Reading database ... 198003 files and directories currently installed.)
Preparing to unpack .../can-utils_2018.02.0-1ubuntu1_amd64.deb ...
Unpacking can-utils (2018.02.0-1ubuntu1) ...
Setting up can-utils (2018.02.0-1ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
```

检查安装结果:

命令:

candump --help

```
root@ubuntu:/home/zhumiao# candump -help
candump - dump CAN bus traffic.

Usage: candump [options] <CAN interface>+
  (use CTRL-C to terminate candump)

Options:                 可以输出这些帮助信息，说明安装成功！
         -t <type>   (timestamp: (a)bsolute/(d)elta/(z)ero/(A)bsolute w date)
         -H          (read hardware timestamps instead of system timestamps)
         -c          (increment color mode level)
         -i          (binary output - may exceed 80 chars/line)
         -a          (enable additional ASCII output)
         -S          (swap byte order in printed CAN data[] - marked with '`' )
         -s <level>  (silent mode - 0: off (default) 1: animation 2: silent)
         -b <can>    (bridge mode - send received frames to <can>)
         -B <can>    (bridge mode - like '-b' with disabled loopback)
         -u <usecs>  (delay bridge forwarding by <usecs> microseconds)
         -l          (log CAN-frames into file. Sets '-s 2' by default)
         -L          (use log file format on stdout)
         -n <count>  (terminate after receiption of <count> CAN frames)
         -r <size>   (set socket receive buffer to <size>)
         -D          (Don't exit if a "detected" can device goes down.
         -d          (monitor dropped CAN frames)
         -e          (dump CAN error frames in human-readable format)
         -x          (print extra message infos, rx/tx brs esi)
         -T <msecs>  (terminate after <msecs> without any reception)

Up to 16 CAN interfaces with optional filter sets can be specified
on the commandline in the form: <ifname>[,filter]*
```

3.3 参数配置

注意：不管是使用 can-utils 工具还是自己开发的 socketcan 应用，使用前，参数配置都是必须的,而且是每次主机重启都必须配置，所以建议将参数配置命令放到主机的系统启动脚本里执行。

波特率参数配置:

ip link set can0 type can bitrate 1000000

指令含义:

can0 通道参数配置

波特率:1M 对应命令里的 1000000，单位是 Hz

```
root@lys-virtual-machine:~# ip link set can0 type can bitrate 1000000
root@lys-virtual-machine:~#
```

3.4 开启设备

命令:”ip link set up can0”

指令含义:
使能 can0 通道

```
root@lys-virtual-machine:/home/lys# ip link set up can0
root@lys-virtual-machine:/home/lys#
```

3.5 软件(can-utils)使用

3.5.1 数据发送

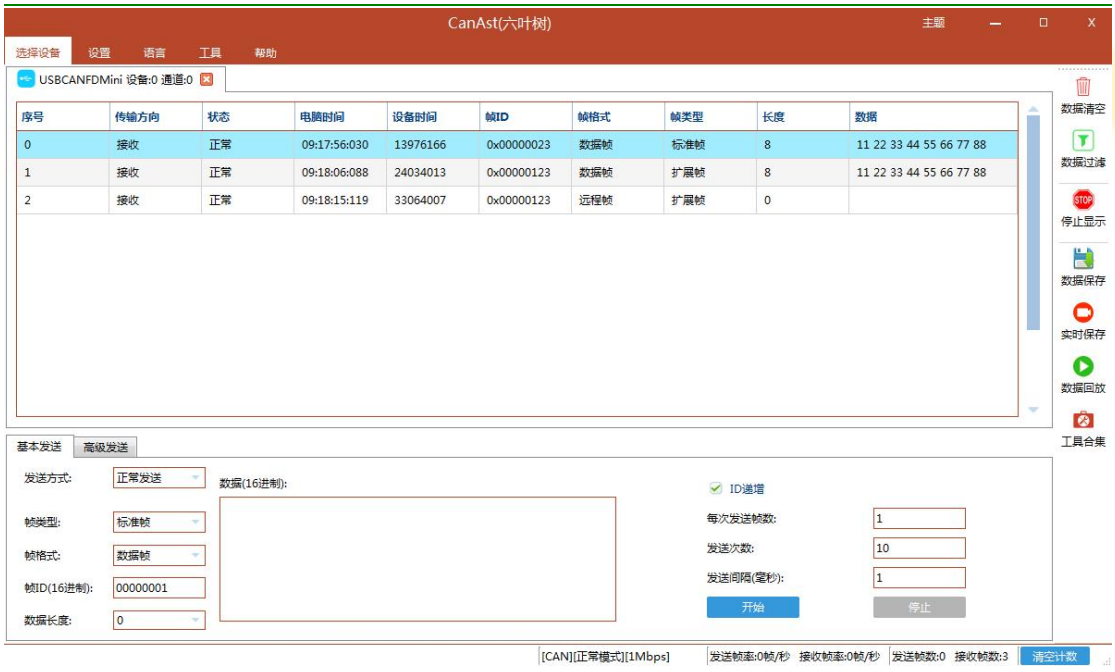
//ID 3 位代表标准帧 8 位代表扩展帧 不足的补零
cansend can0 023#1122334455667788

//扩展数据帧
cansend can0 00000123#1122334455667788

//远程帧
cansend can0 00000123#R

命令截图如下:

```
root@lys-virtual-machine:~# cansend can0 023#1122334455667788
root@lys-virtual-machine:~# cansend can0 00000123#1122334455667788
root@lys-virtual-machine:~# cansend can0 00000123#R
root@lys-virtual-machine:~#
```

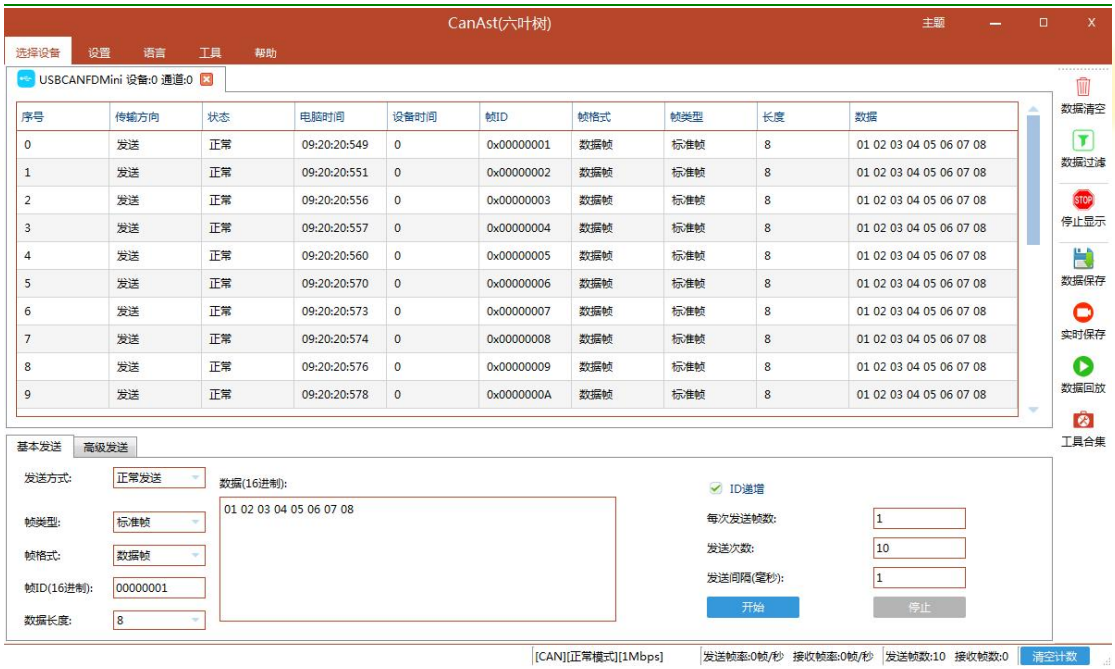


3.5.2 数据接收

//接收数据并打印在窗口

candump can0

```
root@lys-virtual-machine:~# candump can0
  can0  001   [8]  01 02 03 04 05 06 07 08
  can0  002   [8]  01 02 03 04 05 06 07 08
  can0  003   [8]  01 02 03 04 05 06 07 08
  can0  004   [8]  01 02 03 04 05 06 07 08
  can0  005   [8]  01 02 03 04 05 06 07 08
  can0  006   [8]  01 02 03 04 05 06 07 08
  can0  007   [8]  01 02 03 04 05 06 07 08
  can0  008   [8]  01 02 03 04 05 06 07 08
  can0  009   [8]  01 02 03 04 05 06 07 08
  can0  00A   [8]  01 02 03 04 05 06 07 08
```



3.6 关闭设备

命令:”ip link set down can0”
指令含义:
关闭 can0 通道

```
root@lys-virtual-machine:~# ip link set down can0
root@lys-virtual-machine:~#
```

4.代码接口调用

应用代码可以使用 SocketCAN 套接字对设备进行控制，包括数据的接收和发送以及滤波。这里要注意，代码里的 socketCAN 套接字是不能控制 CAN 卡的开启和关闭以及波特率的设置，这些都必须通过上面说的“参数配置”配置好。具体调用逻辑流程，可以查看六叶树官方的 socketCAN 专栏:http://www.liuyeshu.cn/?page_id=1256,里面有代码用例。

4.1 C++

```
/**
*****************************************************************************
* @file    socketCanTest.cpp
* @author  lys
* @version V1.1.0
* @date    2023
* @brief
*
* @note
* SOCKET CAN收发使用用例
* 准备条件:
* 1.设备驱动已安装好,ifconfig -a 能够查看can0设备
* 2.已通过命令ip link配置好波特率并开启
    ip link set can0 type can bitrate 500000
    ip link set up can0
* 用法:
* write:发送can数据,使用结构体can_frame
* read:接收can数据,使用结构体can_frame
* 代码功能:
* 程序运行,初始化了socket,设置了过滤器功能(setsockopt), 只接收0x590的报文, 发送了1帧0x610
* 报文, 接收到0x590的报文接收线程会自动退出, 整个程序退出。
* @endverbatim
*****************************************************************************
* 版权:长沙六叶树教育科技有限公司
* 官网:www.liuyeshu.cn
* 更多资料教程下载见官网
*****************************************************************************
修改日期     版本号    修改者    功能描述
2018.11.10 V1.1.0  LYS
-----------------------------------------------------------------------------
*****************************************************************************
*/
#include <iostream>
#include <string.h>
#include <unistd.h>
#include <net/if.h>
#include <sys/ioctl.h>
#include <sys/socket.h>
#include <linux/can.h>
#include <linux/can/raw.h>
#include <thread>
```

```
/**
* @输入参数: 无
* @输出参数: 无
* @返 回 值: 无
* @说    明:接收线程函数
* 接收到1帧数据自动退出
*/
void receiveThread(int socket)
{
    struct can_frame frame;

    std::cout << "Info: read start...\n";

    int nbytes = read(socket, &frame, sizeof(struct can_frame));

    if (nbytes < 0)
    {
        std::cout << "Error: can raw socket read.\n";
        return;
    }

    if (nbytes < sizeof(struct can_frame))
    {
        std::cout << "Error: read incomplete CAN frame.\n";
        return;
    }

    printf("Receive OK\n");

    // 打印接收到的报文数据
    printf("ID=0x%X DLC=%d Data: ", frame.can_id, frame.can_dlc);
    for (uint8_t i = 0; i < frame.can_dlc; ++i)
    {
        printf("0x%02X ", frame.data[i]);
    }
    printf("\n");
}
```

```c
int main()
{
    int nbytes;
    struct sockaddr_can socketCan0Addr;
    struct ifreq ifr;
    struct can_frame frame = {0};

    int socketFd = socket(PF_CAN, SOCK_RAW, CAN_RAW);

    // 获取can0的接口索引
    strcpy(ifr.ifr_name, "can0");
    ioctl(socketFd, SIOCGIFINDEX, &ifr);

    socketCan0Addr.can_family = AF_CAN;
    //把can0的接口索引赋值给socketCan0Addr
    socketCan0Addr.can_ifindex = ifr.ifr_ifindex;

    // 把socketCan0Addr绑定到socket上
    bind(socketFd, (struct sockaddr *)&socketCan0Addr, sizeof(socketCan0Addr));

    /*设置过滤，只接收id是0x590的CAN报文,如果需要接收所有数据
    ,不执行CAN_RAW_FILTER操作即可*/
    struct can_filter rfilter;
    rfilter.can_id = 0x590;
    rfilter.can_mask = CAN_SFF_MASK;
    setsockopt(socketFd, SOL_CAN_RAW, CAN_RAW_FILTER, &rfilter, sizeof(rfilter));

    // 开启接收回复的线程
    std::thread t(receiveThread, socketFd);

    // 准备数据
    frame.can_id = 0x610;
    //扩展帧
    frame.can_id |= CAN_EFF_FLAG;
    frame.can_dlc = 8;
    frame.data[0] = 0x40;
    frame.data[1] = 0x01;
    frame.data[2] = 0x10;
    frame.data[3] = 0x00;
    frame.data[4] = 0x00;
    frame.data[5] = 0x00;
    frame.data[6] = 0x00;
    frame.data[7] = 0x00;

    // 发送数据
    nbytes = write(socketFd, &frame, sizeof(frame));
    if (nbytes != sizeof(frame))
    {
        std::cout << "Send Error\n";
    }

    //等待接收线程结束
    t.join();

    //程序退出
    return 0;
}
```

这里做了数据过滤，如果需要接收所有数据，不执行该步骤即可。

点击进入下载界面

5.免责说明

本文档提供有关长沙六叶树教育科技有限公司产品的信息。本文档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其它方式授予任何知识产权许可。除六叶树教育科技在其产品的销售条款和条件中声明的责任之外，六叶树概不承担任何其它责任。并且，六叶树电子产品的销售使用不作任何明示或暗示的担保，包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等，均不作担保。六叶树可能随时对产品规格及产品描述做出修改，不另行通知。

6.联系我们

电话:15211065817(业务合作咨询)

邮箱:798746621@qq.com(业务咨询+技术支持)

微信:18673379565(技术支持)

官网:www.liuyeshu.cn(资料下载)

网上商城:https://shop112408209.taobao.com(产品购买)

淘宝店铺搜索:“六叶树教育科技”