



# 应用笔记

ACM32H5 系列芯片  
以太网应用笔记

版本: V1.0

日期: 2024-11-22

**上海航芯电子科技股份有限公司**

# 1. 概述

本应用手册适用于 ACM32H5 系列芯片。它描述了芯片的以太网使用说明。

本应用说明应与相关的用户手册、数据手册一同阅读。

## 2. 初始化

- 1) 设置所有参数到默认值，调用 HAL\_ETH\_InitDefaultParamter ()。
- 2) 设置一般应用参数值，见 2.1.1~2.1.9。
- 3) 使能中断，使能发送中断、接收中断。
- 4) 调用 HAL\_ETH\_Init () 初始化 ETH。
- 5) 初始化 PHY。

PHY 接口读写函数：

```
HAL_ETH_ReadPHYRegister();
```

```
HAL_ETH_WritePHYRegister();
```

### 2.1.1. MAC 地址

变量：ETH\_Handle.Init.MACAddr

MAC 地址，又称为以太网地址，硬件地址，为网络中唯一标示。MAC 地址长度 6 字节，格式为 XX-XX-XX-XX-XX-XX。

ETH 最多 4 个 MAC 地址，MAC 地址 0 默认始终有效，MAC 地址 1~MAC 地址 3 需要配置后才能使用。

### 2.1.2. Media 接口

变量：ETH\_Handle.Init.MediaInterface

ETH 支持 MII 接口 (ETH\_MEDIA\_INTERFACE\_MII)、RMII 接口 (ETH\_MEDIA\_INTERFACE\_RMII)。

### 2.1.3. PHY 地址

变量：ETH\_Handle.Init.PhyAddress

PHY 通过硬件地址 PHYAD[X:0] 配置地址。

### 2.1.4. 自动协商

变量：ETH\_Handle.Init.AutoNegotiation

PHY 根据网络情况自动调整以太网速度和全/半双工模式。

### 2.1.5. 速度

变量：ETH\_Handle.Init.Speed

ETH 支持 10Mbps (ETH\_SPEED\_10M) /100Mbps (ETH\_SPEED\_100M)。

### 2.1.6. 全/半双工模式

变量：ETH\_Handle.Init.DuplexMode

ETH 支持全双工 (ETH\_MODE\_FULL\_DUPLEX) /半双工 (ETH\_MODE\_HALF\_DUPLEX)。

## 2.1.7. 发送描述符

变量：发送缓冲区：ETH\_Handle.Init.TxBuff  
发送缓冲区数量：ETH\_Handle.Init.TxBuffNbr  
发送缓冲区长度：ETH\_Handle.Init.TxBuffLen  
发送描述符：ETH\_Handle.Init.TxDesc  
发送描述符数量：ETH\_Handle.Init.TxDescNbr  
发送描述符模式：ETH\_Handle.Init.TxDescListMode  
发送缓冲区列表：ETH\_Handle.Init.TxBuffTab  
发送缓冲区列表节点数量：ETH\_Handle.Init.TxBuffNodeNbr

TxBuffTab 中包括发送缓冲区地址、发送数据总长度、下一个发送缓冲区地址。发送缓冲区地址指向 TxBuff。此变量主要为 LWIP 服务。

发送描述符模式：环式 (ETH\_DESC\_LIST\_MODE\_RING) 和链表式 (ETH\_DESC\_LIST\_MODE\_LIST)。环式，一个发送描述符有两个发送缓冲区，下一个发送描述符的发送缓冲区为自动增加相应增量地址，自动增加数量由 ETH\_Handle.Init.DMAConfig.DescriptorSkipLen 指定。链表式，一个发送描述符有一个发送缓冲区，下一个发送描述符的发送缓冲区由描述符中的 DESC2 指定，在初始化发送描述符时设置 DESC2，且保存中发送描述符的 Buff2 中。

## 2.1.8. 接收描述符

变量：接收缓冲区：ETH\_Handle.Init.RxBuff  
接收缓冲区数量：ETH\_Handle.Init.RxBuffNbr  
接收缓冲区长度：ETH\_Handle.Init.RxBuffLen  
接收描述符：ETH\_Handle.Init.RxDesc  
接收描述符数量：ETH\_Handle.Init.RxDescNbr  
接收描述符模式：ETH\_Handle.Init.RxDescListMode  
接收缓冲区列表：ETH\_Handle.Init.RxBuffTab  
接收缓冲区列表节点数量：ETH\_Handle.Init.RxBuffNodeNbr  
自动 CRC：ETH\_Handle.Init.AutoCRC  
自动 PAD：ETH\_Handle.Init.AutoPad  
校验和插入控制：ETH\_Handle.Init.AutoChecksumInsertion

RxBuffTab 中包括接收缓冲区地址、接收数据总长度、下一个发送缓冲区地址。

接收描述符模式：环式 (ETH\_DESC\_LIST\_MODE\_RING) 和链表式 (ETH\_DESC\_LIST\_MODE\_LIST)。环式，一个接收描述符有两个接收缓冲区，下一个接收描述符的接收缓冲区为自动增加相应增量地址，自动增加数量由 ETH\_Handle.Init.DMAConfig.DescriptorSkipLen 指定。链表式，一个接收描述符有一个接收缓冲区，下一个接收描述符的接收缓冲区由描述符中的 DESC2 指定，在初始化接收描述符时设置 DESC2，且保存中接收描述符的 Buff2 中。

自动 CRC，硬件计算发送数据 CRC，并自动附加在发送帧的末尾。LWIP，可禁止软件 CRC，由硬件自动计算并附加，可节约软件节约时间。Cyclone TCP 强制软件 CRC，不可设置。

自动 PAD，硬件自动将不足 64 字节的帧添加补位项。

校验和插入控制，硬件自动对 IP 报头校验和进行计算和插入、对有效负载 (TCP/UDP/ICMP) 报头校验和进行计算和插入。LWIP，禁止软件对 IP/TCP/UDP/ICMP 报头校验和进行计算和插入，使能硬件自动计算和

插入，可节约软件计算的时间。Cyclone TCP 强制对 IP/TCP/UDP/ICMP 报头进行计算和插入，因此可禁止校验和插入控制。ETH 在对 ICMP 报头进行计算和插入时，校验和初始值必须为 0，而 Cyclone TCP 对 ICMP 报头已强制进行计算和插入，因此，ETH 在 Cyclone TCP 时不能设置对有效负载报头校验和进行计算和插入，否则将会造成 ICMP 报头校验和的错误，导致 PING 协议错误。

## 2.1.9. 以太网 DLYB

变量：单位延迟数：ETH\_Handle.Init.Delay.Uint

延迟线长度：ETH\_Handle.Init.Delay.Len

DLYB 指在 RX 时钟延迟一个时间进行接收数据的采样。DLBY 包括三个时间概念，基本延迟，初始延迟，延迟步长。

基本延迟，指使用 DLYB 有一个外围基本消耗的时间，目前测试计算为 1.7ns。

初始延迟，指每个单位延迟也有一个基本延迟，目前测试计算为 508~572ps，典型值为 550ps。

延迟步长，指每个单位延迟中一个 Uint 的延迟时间，目前测试计算为 50.5~51.7ps，典型值为 50ps。

DLYB 延迟时间计算公式：

$$\text{DLYB} = \text{基本延迟} + (\text{初始延迟} + \text{延迟步长} * \text{Uint}) * \text{Len}$$

### 3. PHY 选择

PHY 接口支持 MII、RMII 两种接口。

MII 接口，18 个引脚，PHY 芯片 LAN8710。

MII 接口 PHY 的时钟，由外部供给，时钟频率 25MHz，四根数据线。PHY 时钟可由独立的外部时钟供给，也可由 ACM32H5 供给。MAC 时钟 TX\_CLK, RX\_CLK 由 PHY 提供。

RMII 接口，9 个引脚，PHY 芯片 LAN8710、LAN8720。

RMII 接口的 MAC/PHY 共用一个时钟 REF\_CLK，时钟频率 50MHz，两根数据线。PHY 的时钟 25MHz，可以由外部独立的时钟供给，也可由 ACM32H5 供给。PHY 在内部倍频，通过 REF\_CLK 输出到 MAC。

如果使用 LPI 功能，可选择支持 “Compliant with Energy Efficient Ethernet IEEE 802.3az” 功能的 PHY 芯片，如 LAN8740。经测试，ETH 进入发送 LPI 状态，PHY 可节约 43mA 电流。ETH 进入接收 LPI 状态，PHY 可节约 4mA 电流。

如果使用 WOL 功能，可选择支持 “Wake on LAN (Wol) support” 功能的芯片，如 LAN8740、LAN8742。

## 4. 发送数据

函数原型:

```
HAL_StatusTypeDef HAL_ETH_Transmit(ETH_HandleTypeDef *heth, ETH_BuffTypeDef *buff,
uint32_t mode, ETH_TxStatusTypeDef *pStatus);
```

参数描述:

heth: ETH 句柄。

buff: 发送数据缓冲区。这是一个 ETH\_BuffTypeDef 类型, 可以输入多个缓冲区, 可以实现大型数据包的发送。

mode: 发送模式。发送模式包括 ETH\_TX\_MODE\_DATA\_COPY、ETH\_TX\_MODE\_TIMESTAMP、ETH\_TX\_MODE\_WAIT\_TX\_COMPLETE 三个模式。

ETH\_TX\_MODE\_DATA\_COPY: 数据 copy 模式。

buff 中的数据存储于多个数据缓冲区中, 而除最后一个缓冲区的其他缓冲区数据的长度有不足内部缓冲区长度的现象, 因此, 必须设置 ETH\_TX\_MODE\_DATA\_COPY, 通过 copy 数据的方式将 buff 中多个缓冲区的数据依次 copy 到内部缓冲区, 从而使所有数据依次填满内部缓冲区。

而 buff 只有一个缓冲区数据, 或除最后一个缓冲区的其他缓冲区数据的长度都等于内部缓冲区长度时, 可以不选择 ETH\_TX\_MODE\_DATA\_COPY, 此时用 buff 中每个缓冲区的地址代替内部缓冲区地址, 直接使用 buff 中的缓冲区发送数据。

使用 ETH\_TX\_MODE\_DATA\_COPY, 发送数据将多出一个数据 copy 的时间, 发送效率较低。但驱动和应用层之间的耦合性较低。

不使用 ETH\_TX\_MODE\_DATA\_COPY, 发送数据无需数据 copy, ETH 直接使用 buff 中缓冲区发送, 发送效率较高, 但驱动和应用之间的耦合性较高, 在驱动发送完成前, 应用不能使用 buff 中的缓冲区。

ETH\_TX\_MODE\_TIMESTAMP: 时间戳模式。

设置 ETH\_TX\_MODE\_TIMESTAMP, 阻塞式发送, 且发送前须在 DESC0 中设置时间戳标志, 发送完成后读取时间戳。

ETH\_TX\_MODE\_WAIT\_TX\_COMPLETE: 等待发送完成模式。

设置 ETH\_TX\_MODE\_WAIT\_TX\_COMPLETE, 阻塞式发送, 发送完成后退出函数。

pStatus: 发送状态。

返回发送状态, DESC0 及时间戳。

## 5. 接收数据

函数原型:

```
HAL_StatusTypeDef HAL_ETH_Receive(ETH_HandleTypeDef *heth, ETH_BuffTypeDef **buff,  
ETH_RxStatusTypeDef *pStatus);
```

```
HAL_StatusTypeDef HAL_ETH_ReleaseRxDescriptors(ETH_HandleTypeDef *heth);
```

接收数据分为两个步骤，第一步，接收数据，第二步，释放接收缓冲区。

接收数据，使用 HAL\_ETH\_Receive ()，ETH 将接收缓冲区组成 ETH\_BuffTypeDef 类型的链表，并将链表首地址返回到 buff，接收状态 (DES0、DES4、时间戳) 返回到 pStatus。

用户可直接使用 buff 进行数据处理，也可将数据 copy 到应用层进行处理。

当用户处理完数据，或将所有数据 copy 到应用后，调用 HAL\_ETH\_ReleaseRxDescriptors () 释放接收描述符。只有调用 HAL\_ETH\_ReleaseRxDescriptors () 之后，ETH 才能使用其接收描述符进行继续接收数据。



## 6. 中断

函数原型:

```
void HAL_ETH_TxCpltCallback(ETH_HandleTypeDef *heth);
```

```
void HAL_ETH_RxCpltCallback(ETH_HandleTypeDef *heth);
```

使能发送中断, 发送完成后自动调用 HAL\_ETH\_TxCpltCallback (), 应用可以在此处理发送事宜。

使能接收中断, 接收成功后自动调用 HAL\_ETH\_RxCpltCallback (), 应用可以在此处理接收事宜。

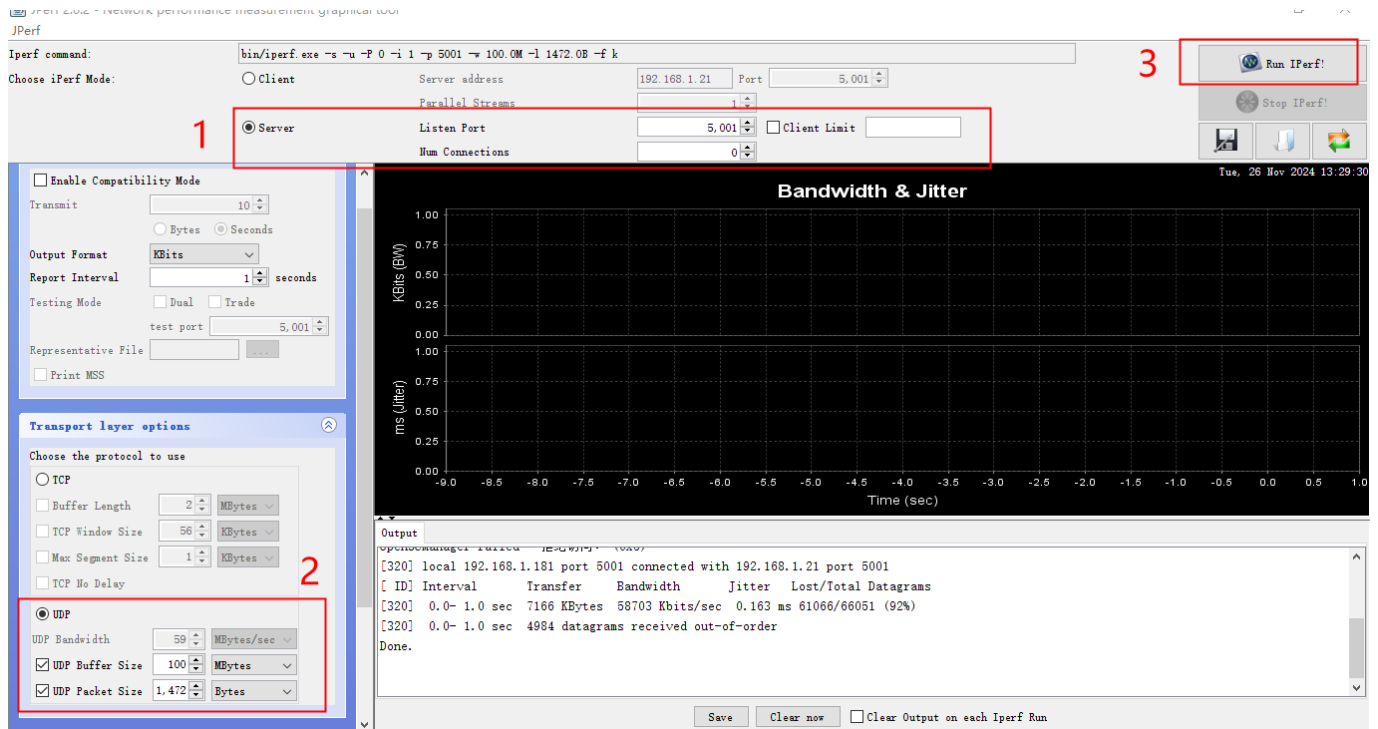
## 7. 测试发送速度

平台：电脑 (Jperft 2.0.2) + 路由器 + EVB 开发板

### 7.1. 设置

#### 7.1.1. JPerf

JPerf 设置：



##### 1) 选择 Server

Listen Port: 5001 (EVB 开发板 PORT)

不选择 Client Limit, 无需输入 EVB 开发板 IP 地址

Num Connections: 0

##### 2) 选择 UDP

选择 UDP Buffer Size: 100 Mbytes

选择 UDP Packet Size: 1472 Bytes

#### 7.1.2. EVB 开发板 ETH 属性

ETH 属性：

1) IPv4 地址: 192.168.1.21

2) IPv4 子网掩码: 255.255.255.0

3) IPv4 默认网关: 192.168.1.1

4) UDP Port: 5001

## 7.2. 操作流程

操作流程：

- 1) 工程: Cyclone\_TCP\_LAN8720\Socket\_UDP\_Recv
- 2) EVB 开发板上电。

EVB 开发板输出：

```
*****
system startup
reset source: 10 0
reset source: sysreq.
HCK: 180000000
PCLK1: 180000000
PCLK2: 180000000
PCLK3: 180000000
PCLK4: 180000000
*****

Press the key to start testing UDP sending speed.
```

- 3) 打开 Jperf 软件，按 7.1.1 设置。
- 4) 点击 Jperf 软件的“Run Iperf!”，启动 UDP 接收。
- 5) EVB 开发板按 KEY，启动 UDP 发送。

UDP 发送完成，EVB 开发板输出：

```
***** test start *****
Detect network connection status.
Network connection succeeded.

remote: 192.168.1.181, 5001
local: 192.168.1.21, 5001

create socket
Successfully created socket.

Send UDP packets, totally 10000 packets, each packet length: 1518 bytes and payload: 1472 bytes.
data length: 15180000 bytes, time: 1937 ms, speed: 7.000000 MB/s 62.000000 Mbps

close socket

***** test end *****
```

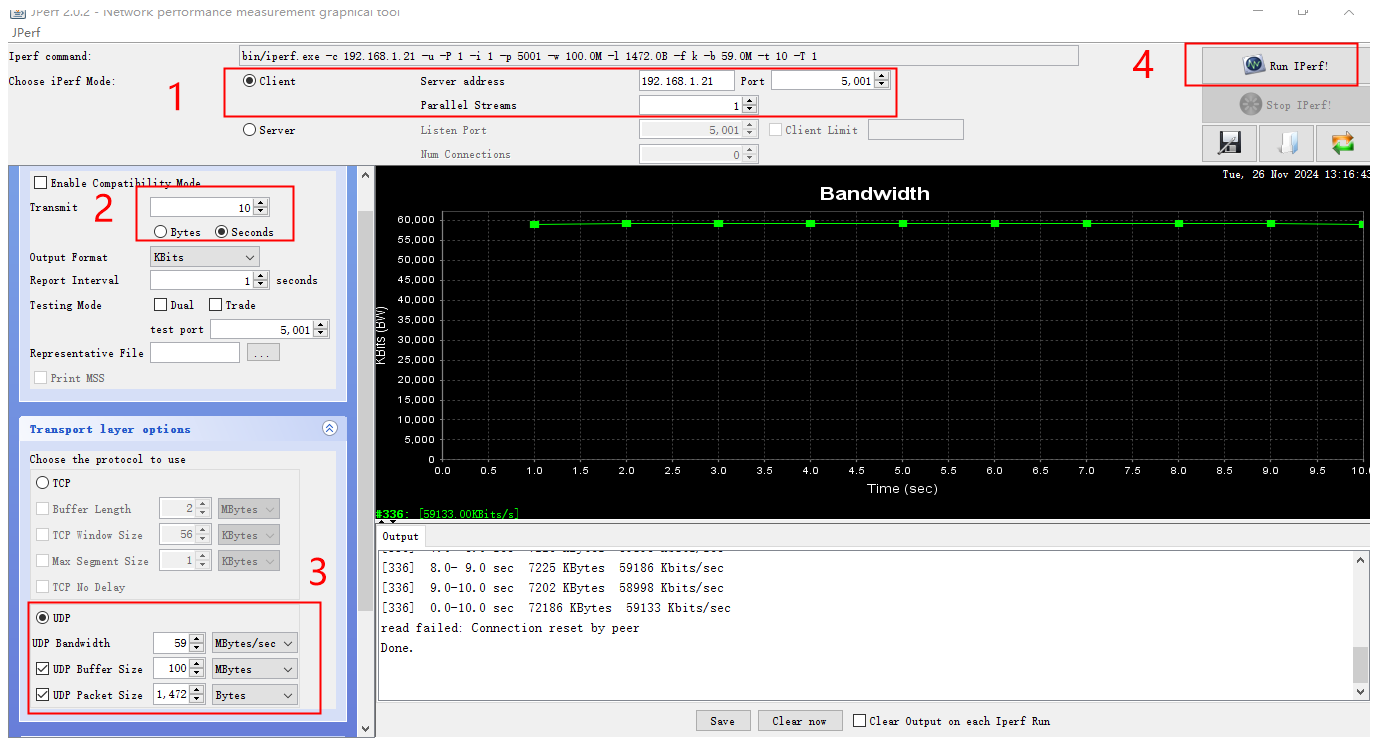
## 8. 测试接收速度

平台：电脑 (Jperf 2.0.2) + 路由器 + EVB 开发板

### 8.1. 设置

#### 8.1.1. JPerf

JPerf 设置如下：



##### 1) 选择 Client

Server address: 192.168.1.21 (EVB 开发板 IP 地址)

Port: 5001 (EVB 开发板 PORT)

Parallel Streams: 1

##### 2) Transmit 选择 10 seconds.

##### 3) 选择 UDP

UDP Bandwidth: 59 Mbytes/sec

选择 UDP Buffer Size: 100 Mbytes

选择 UDP Packet Size: 1472 Bytes

#### 8.1.2. EVB 开发板 ETH 属性

EVB 开发板 ETH 属性：

1) IPv4 地址: 192.168.1.21

2) Ipv4 子网掩码: 255.255.255.0

3) Ipv4 默认网关: 192.168.1.1

4) UDP Port: 5001

## 8.2. 操作流程

操作流程：

- 1) 工程: Cyclone\_TCP\_LAN8720\Socket\_UDP\_Send
- 2) EVB 开发板上电。

EVB 开发板输出如下：

```
*****
system startup
reset source: 10 0
reset source: sysreq.
HCK: 180000000
PCLK1: 180000000
PCLK2: 180000000
PCLK3: 180000000
PCLK4: 180000000
*****

Press the key to start testing UDP sending speed.
```

- 3) 打开 Jperf 软件，按照 8.1.1 进行设置。
- 4) EVB 开发板按下 KEY，启动 UDP 接收。

EVB 开发板输出如下：

```
***** test start *****
Detect network connection status.
Network connection succeeded.

remote: 192.168.1.181, 5001
local: 192.168.1.21, 5001

create socket
Successfully created socket.
```

- 5) Jperf 点击 “Run IPerf!”，启动 UDP 发送。

EVB 开发板收到 10000 个 UDP 数据包，EVB 开发板输出如下：

```
Send UDP packets, totally 10000 packets, each packet length: 1518 bytes and payload: 1472 bytes.
data length: 15180000 bytes, time: 1938 ms, speed: 7.000000 MB/s 62.000000 Mbps

close socket

***** test end *****
```

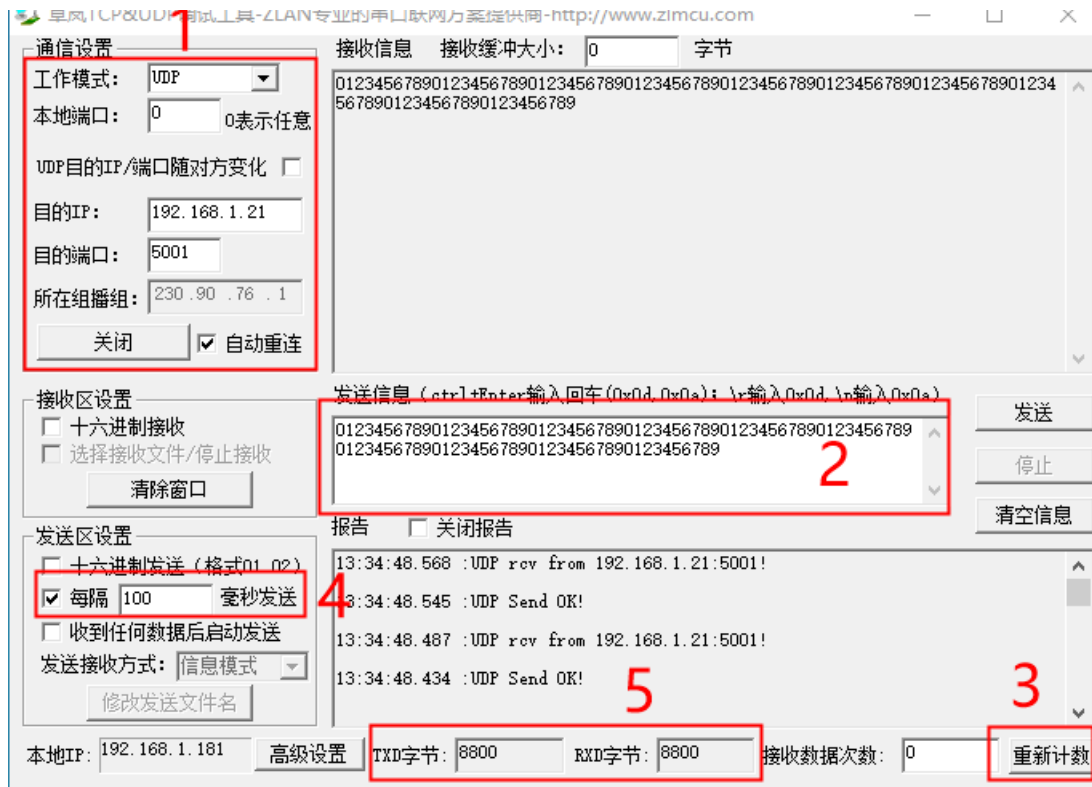
## 9. 测试 UDP 数据收发

平台：电脑 (Jperf 2.0.2) + 路由器 + EVB 开发板

### 9.1. 设置

#### 9.1.1. TCP&UDP 调试工具

TCP&UDP 调试工具设置：



- 1) 工作模式：UDP
- 2) 本地端口：0
- 3) 目的 IP：192.168.1.21
- 4) 目的端口：5001
- 5) 选择自动重连
- 6) 发送信息输入 10 个字符串 “0123456789”，共 100 个字符
- 7) 点击 “重新计数”

#### 9.1.2. EVB 开发板 ETH 属性

EVB 开发板 ETH 属性：

- 1) IPv4 地址：192.168.1.21
- 2) IPv4 子网掩码：255.255.255.0
- 3) IPv4 默认网关：192.168.1.1
- 4) UDP Port：5001

## 9.2. 操作流程

操作流程：

- 1) 工程：Cyclone\_TCP\_LAN8720\Socket\_UDP\_SendRecv
- 2) EVB 开发板上电。

EVB 开发板输出如下：

```
*****
system startup
reset source: 10  0
reset source: sysreq.
HCK: 220200000
PCLK1: 220200000
PCLK2: 220200000
PCLK3: 220200000
PCLK4: 220200000
*****
press the key to start testing UDP, receive data and return the received data.
```

- 1) TCP&UDP 调试工具，按照 9.1.1 进行设置。
- 2) 点击通信设置中的“打开”，创建 UDP socket。
- 5) EVB 开发板按下 KEY，启动 UDP 接收。

EVB 开发板输出如下：

```
***** test start *****
detect network connection status.
network connection succeeded.
local: 192.168.1.21, 5001
create recv socket
ok
```

- 3) 选择“每隔 100 毫秒”发送，启动自动发送 UDP 包。

发送 10 秒后取消“每隔 100 毫秒发送”，停止自动发送。

观察 TXD 字节、RXD 字节，这两个数据应该一致，说明 EVB 板收到一包数据发送一包数据。

## 10. 性能比较

ETH 收发速度与 STM32F429 比较

ACM32H5 对标 STM32F4，分别从以下几个方面对以太网收发速度进行性能测试。

1) 程序存放于 SPI FLASH、SRAM1、SDRAM 中进行比较。

当程序存放于 SPI FLASH 时，SPI FLASH 接口速度分为 2 分频 (XIP-2)、4 分频 (XIP-4)。

2) 数据存放于 SRAM1、SRAM2、SDRAM。

3) 频率分为 180MHz、220MHz。



	程序	数据	频率	芯片	接收速度(MBps)	接收速度(Mbps)	发送速度(MBps)	发送速度(Mbps)
BGA240	XIP-4	SRAM1	180MHz	H5	6.02	48.15	8.69	69.55
	XIP-4	SRAM2	180MHz	H5	6.54	52.34	9.54	76.33
	XIP-2	SRAM1	180MHz	H5	7.6	60.84	9.76	78.14
	XIP-2	SRAM2	180MHz	H5	8.36	66.91	10.88	87.05
	SRAM1	SRAM2	180MHz	H5	10.54	84.33	12.35	98.81
	SDRAM	SRAM1	180MHz	H5	9.1	72.85	10.78	86.25
	SDRAM	SRAM2	180MHz	H5	9.84	78.7	12.24	97.94
	SDRAM	SDRAM	180MHz	H5	2.49	19.94	3.09	24.75
	XIP-4	SRAM1	220MHz	H5	7.44	59.58	10.63	85.1
	XIP-4	SRAM2	220MHz	H5	8.12	65.01	11.67	93.41
	XIP-2	SRAM1	220MHz	H5	9.11	72.93	11.95	95.62
	XIP-2	SRAM2	220MHz	H5	9.37	75.01	12.35	98.81
	SRAM1	SRAM2	220MHz	H5	12.34	98.73	12.35	98.81
	SDRAM	SRAM1	220MHz	H5	10.62	84.98	12.31	98.49
	SDRAM	SRAM2	220MHz	H5	11.77	94.13	12.35	98.81
	SDRAM	SDRAM	220MHz	H5	3.02	24.17	3.81	30.45
F429	EFLASH	SRAM	180MHz	F429	11.33	90.63	12.32	98.57

## 11. 版本历史

版本	日期	作者	描述
V1.0	2024-11-22	Aisinochip	初始版

### 版权声明

本文档的所有部分，其著作权归上海航芯电子科技股份有限公司（简称航芯科技）所有，未经航芯科技授权许可，任何个人及组织不得复制、转载、仿制本文档的全部或部分组件。本文档没有任何形式的担保、立场表达或其他暗示，若有任何因本文档或其中提及的产品所有资讯所引起的直接或间接损失，航芯科技及所属员工恕不为其担保任何责任。除此以外，本文档所提到的产品规格及资讯仅供参考，内容亦会随时更新，恕不另行通知。

### 联系我们

公司：上海航芯电子科技股份有限公司

地址：上海市闵行区合川路 2570 号科技绿洲三期 2 号楼 702 室

邮编：200241

电话：+86-21-6125 9080

传真：+86-21-6125 9080-830

Email: [service@aisinochip.com](mailto:service@aisinochip.com)

Website: [www.aisinochip.com](http://www.aisinochip.com)