1 网络通信实验

1.1 实验目的

使用 ALIENTEK 阿波罗 STM32F429 开发板自带的网口和 LWIP 实现：TCP 服务器、TCP 客服端、UDP 以及 WEB 服务器等四个功能，熟悉 LWIP 网络协议栈的使用。

1.2 实验原理

STM32F429 芯片自带以太网模块，该模块包括带专用 DMA 控制器的 MAC 802.3（介质访问控制）控制器，支持介质独立接口 (MII) 和简化介质独立接口 (RMII)，并自带了一个用于外部 PHY 通信的 SMI 接口，通过一组配置寄存器，用户可以为 MAC 控制器和 DMA 控制器选择所需模式和功能。STM32F429 以太网功能框图如图所示：

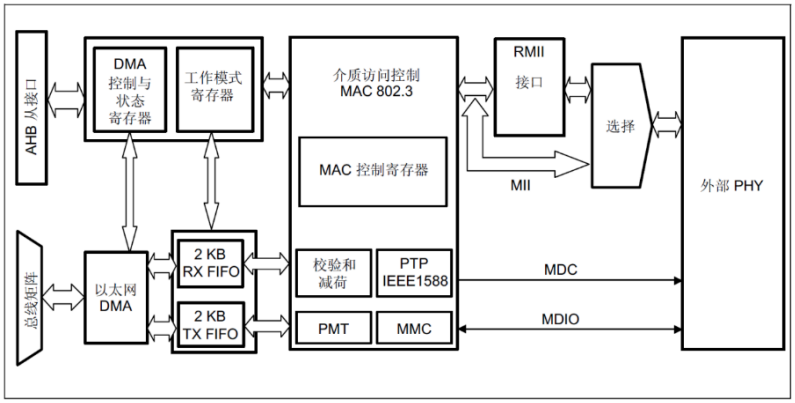

图 **1** **STM32F429** 以太网框图

从上图可以看出，STM32F429 是必须外接 PHY 芯片，才可以完成以太网通信的，外部 PHY 芯片可以通过 MII/RMII 接口与 STM32F429 内部 MAC 连接，并且支持 SMI（MDIO&MDC）接口配置外部以太网 PHY 芯片。

阿波罗 STM32F429 开发板使用的是 LAN8720A 作为 PHY 芯片。LAN8720A 是低功耗的 10/100M 以太网 PHY 层芯片，I/O 引脚电压符合 IEEE802.3-2005 标准，支持通过 RMII 接口与以太网 MAC 层通信，内置 10-BASE-T/100BASE-TX 全双工传输模块，支持 10Mbps 和 100Mbps。

LAN8720A 可以通过自协商的方式与目的主机最佳的连接方式(速度和双工模式)。

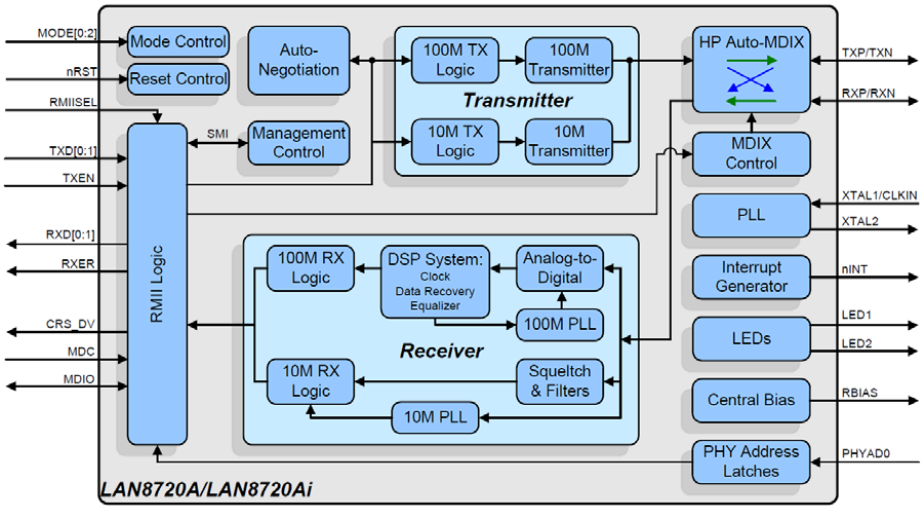

图 **2** **LAN8720A** 功能框图

1.3 代码修改

本实验中，HAL 库模板代码已经实现了 UDP 收发和 TCP 收发、Web 服务器访问功能。我们小组在此基础上进行修改，使开发板能够控制 LCD，根据经由 UDP 包传送的字符串数据进行实时图形显示。以下是修改后的代码：

udp_demo.c 片段

```
void draw(u16 center, u16 length) {
  // 根据中心点与边长绘制矩形
  u16 temp = length >> 1;
  LCD_DrawRectangle(center - temp, center - temp, center + temp, center + temp);
}
u16 stoi(const u8 *s) {
  // 字符串转整数
  u16 temp = 0;
  for (u8 i = 0; s[i] ≠ '\0'; ++i) {
    temp += s[i] - '0';
    temp *= 10;
  }
  return temp;
}
// UDP 测试
void udp_demo_test(void) {
  // ... 省略了创建 UDP 客户端的代码
  while (res == 0) {
    key = KEY_Scan(0);
    if (key == WKUP_PRES)
      break;
    if (key == KEY0_PRES) // KEY0 按下了,发送数据
    {
      udp_demo_senddata(udppcb);
    }
    if (udp_demo_flag & 1 << 6) // 是否收到数据?
    {
      LCD_Clear(99999);                    // 清屏
      draw(230, stoi(udp_demo_recvbuf)); // 根据收到的数据画不同边长的矩形
      udp_demo_flag &= ~(1 << 6); // 标记数据已经被处理了.
    }
    lwip_periodic_handle();
    delay_ms(2);
  }
  udp_demo_connection_close(udppcb);
  myfree(SRAMIN, tbuf);
}
```

在 `udp_demo.c` 中，创建了 `draw` 与 `stoi` 函数用来转换与绘制接收的信息，并更改 `udp_demo_test` 函数，使其能够接收 UDP 包，并调用 `draw` 函数进行实时显示。

lwip_comm.c 片段

```c
void lwip_comm_default_ip_set(__lwip_dev *lwipx) {
  // ...
  lwipx->remoteip[0] = 192;
  lwipx->remoteip[1] = 168;
  lwipx->remoteip[2] = 1;
  lwipx->remoteip[3] = 130;
  // ...
}
```

在 lwip_comm.c 中，可以调整默认的 remote IP 值，使开发板能够快速连接到调试设备的 IP 地址，无需手动选择。

由于实验中开发板无法通过 DHCP 获取 IP 地址，我们还在 lwipopts.h 中设置了 #define LWIP_DHCP 0，使 DHCP 默认禁用，节省了开发板等待 DHCP 任务直到循环变量自然溢出的时间。

1.4 实验结果

首先编译并下载模板代码，测试网口收发功能，并观察开发板与调试设备之间的通信。设置好开发板与网络调试助手的 IP 与端口后，按动按键，可以观察到开发板与调试设备之间通信正常。

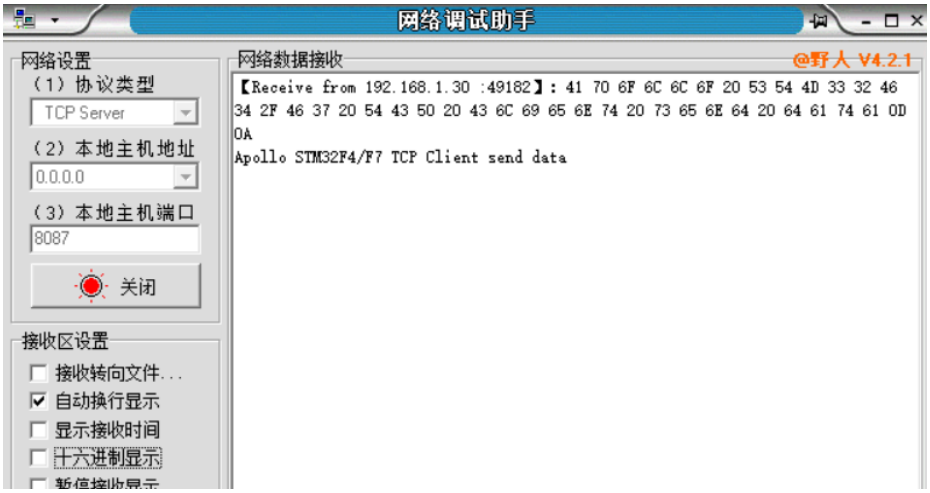

图 3　网络调试助手

然后编译下载修改后的代码，设置开发板参数，当其显示 Connected 后，在网络调试助手发送数字字符串，观察到开发板能够显示矩形，并随着输入数字的变大而变大。

1.5 心得体会

在本次实验中，我们小组完成了对开发板与调试设备的通信，并修改代码，使用 UDP 包进行图案的实时显示。调试过程中，我们遇到了一些问题，例如因为调试设备的网络配置不正确，开发板无法连接到其 IP 地址。通过查阅网络资料，我们解决了这些问题，并加深了对网络通信的理解。

使用开发板发送字符串时，在网络调试助手中默认显示十六进制数字，需要取消勾选 十六进制显示，才可使消息以字符串形式显示在开发板屏幕上。