



摘要

TM4C129x MCU 是高度连接的 32 位 Arm® Cortex®-M4F 微控制器，配有集成式以太网 MAC 和 PHY 以及各种有线通信外设。本应用报告展示了基于轻量级 IP (lwIP) 栈的各种以太网应用示例。轻量级 IP (lwIP) 栈是一种为嵌入式系统设计的开源 TCP/IP 栈。这些示例所运行的 EK-TM4C1294XL LaunchPad™ 开发套件是适用于 TM4C129x 以太网 MCU 的评估平台。

本文档所讨论的工程配套资料和源代码可从以下 URL 下载：<https://www.ti.com/cn/lit/zip/spna248>。

内容

1 引言.....	3
1.1 许可.....	3
1.2 lwIP 版本.....	3
1.3 lwIP 定制.....	3
1.4 lwIP API.....	3
1.5 lwIP 开发主页和支持.....	5
2 应用示例.....	6
3 应用设置.....	7
3.1 硬件设置.....	7
3.2 软件工具.....	7
4 下载并导入以太网示例.....	8
5 Enet_tcpecho_server_lwip 示例概述.....	12
5.1 构建和刷写程序.....	12
5.2 对 MAC 地址进行检查和编程.....	12
5.3 配置终端窗口.....	15
5.4 运行 enet_tcpecho_server_lwip 示例.....	15
6 Enet_tcpecho_server_static_ip_lwip 示例概述.....	18
6.1 如何将 lwIP 配置为静态地址.....	18
6.2 运行 enet_tcpecho_server_static_ip_lwip 示例.....	19
7 Enet_udpecho_server_lwip 示例概述.....	21
7.1 运行 enet_udpecho_server_lwip 示例.....	21
8 Enet_dns_lwip 示例概述.....	23
8.1 如何将 lwIP 配置为 DNS.....	23
8.2 如何在 Wireshark 上查看 DNS 流量.....	24
8.3 运行 enet_dns_lwip 示例.....	25
9 Enet_sntp_lwip 示例概述.....	26
9.1 运行 enet_sntp_lwip 示例.....	26
10 Enet_tcpecho_client_lwip 示例概述.....	27
10.1 配置服务器 IP 地址.....	27
10.2 配置 SocketTest 服务器并运行 enet_tcpecho_client_lwip 示例.....	27
10.3 enet_tcpecho_client_lwip 示例的 Wireshark 捕获.....	29
11 Enet_adcensor_client_lwip 示例概述.....	29
11.1 运行 adcensor_client_lwip 示例.....	29
12 Enet_udpecho_client_lwip 示例概述.....	30
12.1 运行 enet_udpecho_client_lwip 示例.....	30
13 参考文献.....	31

插图清单

图 1-1. 使用 Raw API 的 TCP 客户端服务器通信.....	4
图 1-2. 使用 Raw API 的 UDP 客户端服务器通信.....	5
图 3-1. 应用示例的硬件设置.....	7
图 4-1. 导入 CCS 工程步骤 1.....	8
图 4-2. 导入 CCS 工程步骤 2.....	9
图 4-3. 导入 CCS 工程步骤 3.....	10
图 4-4. 导入 CCS 工程步骤 4.....	11
图 5-1. 调试 CCS 工程.....	12
图 5-2. 使用 LM Flash Programmer 进行 MAC 地址编程.....	13
图 5-3. 使用 CCS 对 MAC 地址进行编程.....	14
图 5-4. 使用 Uniflash 对 MAC 地址进行编程.....	14
图 5-5. 串行终端设置.....	15
图 5-6. Enet_tcpecho_server_lwip 输出.....	15
图 5-7. Enet_tcpecho_server_lwip 的 SocketTest 客户端配置.....	16
图 5-8. Enet_tcpecho_server_lwip 的客户端到服务器 Wireshark 捕获.....	16
图 5-9. Enet_tcpecho_server_lwip 中的客户端收到数据确认.....	17
图 5-10. Enet_tcpecho_server_lwip 的服务器到客户端 Wireshark 捕获.....	17
图 6-1. Enet_tcpecho_server_static_ip_lwip 输出.....	19
图 6-2. Enet_tcpecho_server_static_ip_lwip 的 SocketTest 客户端配置.....	19
图 6-3. Enet_tcpecho_server_static_ip_lwip 的客户端到服务器 Wireshark 捕获.....	20
图 6-4. Enet_tcpecho_server_static_ip_lwip 的服务器到客户端 Wireshark 捕获.....	20
图 7-1. 查询 PC 的 IP 地址.....	21
图 7-2. Enet_udpecho_server_lwip 输出.....	21
图 7-3. Enet_udpecho_server_lwip 的客户端到服务器 Wireshark 捕获.....	22
图 7-4. Enet_udpecho_server_lwip 的服务器到客户端 Wireshark 捕获.....	22
图 8-1. 端口镜像.....	24
图 8-2. Enet_dns_lwip 输出.....	25
图 8-3. Enet_dns_lwip 的 Wireshark 捕获.....	25
图 9-1. Enet_sntp_lwip 输出.....	26
图 9-2. Enet_sntp_lwip 的 Wireshark 捕获.....	26
图 10-1. Enet_tcpecho_client_lwip 的 SocketTest 服务器配置.....	28
图 10-2. Enet_tcpecho_client_lwip 输出.....	28
图 10-3. Enet_tcpecho_client_lwip 的客户端到服务器 Wireshark 捕获.....	29
图 11-1. Enet_adcensor_lwip 的客户端到服务器 Wireshark 捕获.....	29
图 12-1. Enet_udpecho_client_lwip 输出.....	30
图 12-2. Enet_udpecho_client_lwip 的客户端到服务器 Wireshark 捕获.....	30

表格清单

表 1-1. Raw TCP API.....	4
表 1-2. Raw UDP API.....	5
表 2-1. 应用示例.....	6

商标

LaunchPad™, TivaWare™, and Code Composer Studio™ are trademarks of Texas Instruments.
 Arm® and Cortex® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.
 所有商标均为其各自所有者的财产。

1 引言

lwIP 是 TCP/IP 协议栈的小型实现。lwIP 的实现允许全面的 TCP 支持，同时使用非常少量的 RAM 和 ROM，这适用于嵌入式系统。lwIP 支持以下协议：

- 地址解析协议 (ARP)
- 自动 IP (AUTOIP)。用于选择 IP 地址的无服务器方法。
- 域名解析程序 (DNS)
- 动态主机配置协议 (DHCP)
- 互联网控制消息协议 (ICMP)
- 互联网组管理协议 (IGMP)
- 互联网协议 (IP)
- 简单网络管理协议 (SNMP)
- 传输控制协议 (TCP)
- 点对点协议 (PPP)
- 用户数据报协议 (UDP)

1.1 许可

通过 BSD 许可证可免费获得 lwIP。

1.2 lwIP 版本

lwIP v1.4.1 已移植到 TM4C129x MCU 上，是 TivaWare™ 库发布的一部分。本应用手册中描述的所有示例都是使用 TivaWare v2.2.0.295 库中的 lwIP v1.4.1 编译的。

1.3 lwIP 定制

Lwipopts.h 是一个用户文件，可用于完全配置 lwIP 及其所有模块。所有选项都设置了默认值。您可以通过仅编译应用真正需要的那些功能来优化代码大小。lwipopts.h 已配置为包含适用于所提供示例的功能和内存管理选项。可在每个示例的工程文件夹中找到 lwipopts.h。

1.4 lwIP API

lwIP 为程序提供了三种应用编程接口 (API) 用于与 TCP/IP 代码通信：

- 低级 Raw API
- 高级 Netconn 序列 API
- BSD 套接字 API。

此应用报告中演示的所有示例仅基于 Raw API 接口。Raw API 为应用提供回调接口。应用首先将回调函数注册到不同的核心事件。典型的 TCP 事件包括：

- 接受新的 TCP 连接。
- 远程主机确认数据发送给自己。
- 有新数据到达。
- 定期间隔事件。

当发生相应的事件时，将从 lwIP 内核调用用户提供的回调函数。使用 Raw API 接口时，TCP/IP 代码和应用程序都可以在非操作系统环境中的同一线程中运行。Raw TCP/IP 接口在代码执行速度上更快，并且为内存分散型（代价是编码复杂）。另外两种 API 目前不受 TivaWare 库支持。但是，基于 BSD 套接字的 API 通过 TI-RTOS NDK 获得支持。从 Code Composer Studio™ (CCS) [Resource Explorer](#) 可下载各种基于套接字的以太网示例。

1.4.1 TCP RAW API

表 1-1 列出了要在应用中使用的典型 lwIP Raw TCP API。**图 1-1** 显示使用 Raw API 的 TCP 客户端-服务器通信的简化流程图。

表 1-1. Raw TCP API

功能类别	API	说明
TCP 连接	tcp_new	创建一个新的 TCP PCB (协议控制块)。
	tcp_bind	将 PCB 绑定到本地 IP 地址和端口。可以将 IP 地址指定为 IP_ADDR_ANY，以便将连接绑定到所有本地 IP 地址。
	tcp_listen	使 PCB 侦听传入连接。
	tcp_accept	设置用于新传入连接的回调。接受一个传入连接时，将调用使用 tcp_accept() 指定的回调函数。
	tcp_connect	打开与远程主机的连接。此函数将立即返回并调用在第四个参数中指定的回调函数。
接收 TCP 数据	tcp_recv	设置在新数据到达时将调用的回调函数。
	tcp_recved	当应用收到数据时，必须调用此函数。
发送 TCP 数据	tcp_write	将数据排入队列以进行传输
	tcp_sent	指定在远程主机成功接收 (即已确认) 数据时应调用的回调函数。
	tcp_output	强制立即发送所有排队数据。
应用轮询	tcp_poll	设置应用轮询回调。当连接空闲时，lwIP 将通过调用指定的回调函数来反复轮询应用。
关闭连接和错误处理	tcp_close	关闭 TCP 连接。
	tcp_err	将回调函数设置为发生连接错误时调用。
	tcp_abort	中止连接。

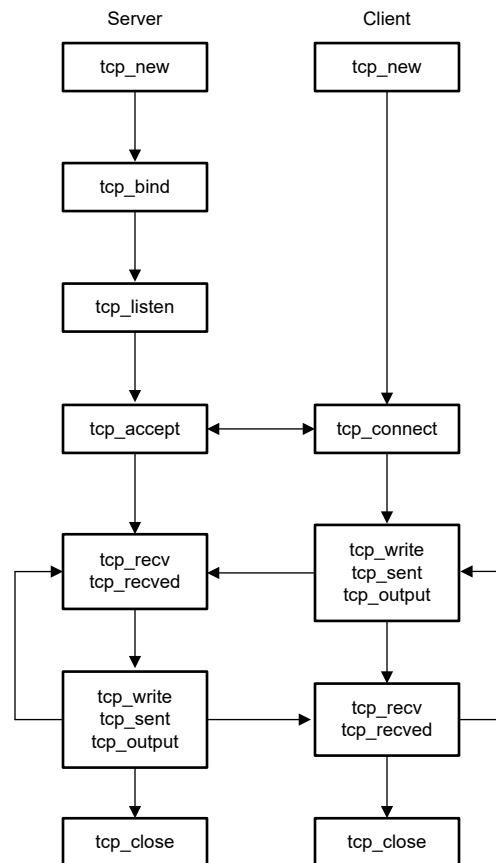


图 1-1. 使用 Raw API 的 TCP 客户端服务器通信

1.4.2 UDP RAW API

表 1-2 列出要在应用中使用 lwIP Raw UDP API。图 1-2 显示使用 Raw API 的 UDP 客户端-服务器通信的简化流程图。

表 1-2. Raw UDP API

功能类别	API	说明
UDP 连接	udp_new	创建一个新的 UDP PCB (协议控制块)。
	udp_bind	将 PCB 绑定到本地 IP 地址和端口。可以将 IP 地址指定为 IP_ADDR_ANY，以便将连接绑定到所有本地 IP 地址。
	udp_connect	将 PCB 与远程 UDP 对等地址相关联。
接收 UDP 数据	udp_recv	设置新数据到达时将调用的回调函数。
发送 UDP 数据	udp_send	将 UDP 数据包发送到与 PCB 关联的当前远程主机。
	udp_sendto	使用 UDP 将数据发送到指定的地址。
关闭连接	udp_disconnect	移除 PCB 的远程端。
	udp_remove	移除并取消分配 PCB。

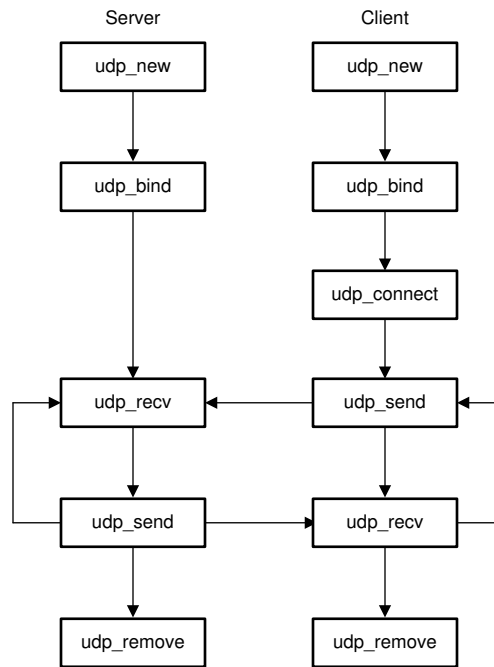


图 1-2. 使用 Raw API 的 UDP 客户端服务器通信

1.5 lwIP 开发主页和支持

有关 lwIP 和支持的其他信息，请参阅以下链接：

- <https://savannah.nongnu.org/projects/lwip>
- https://lwip.fandom.com/wiki/LwIP_Wiki

2 应用示例

TivaWare 库包含演示如何基于 lwIP 栈创建 HTTP Web 服务器应用的以太网示例。本应用报告重点演示了各种回显服务器和客户端应用。此处共提供了八个示例，用来展示运行服务器应用或客户端应用的 TM4C129x MCU。

表 2-1. 应用示例

示例	类型	说明
enet_tcpecho_server_lwip	服务器	使用 TCP 协议的回显服务器应用。服务器回显从客户端接收的数据包。lwIP 配置为从 DHCP 服务器获取 IP 地址。
enet_tcpecho_server_static_ip_lwip	服务器	使用 TCP 协议但配置了 lwIP 以生成静态 IP 地址的回显服务器应用。
enet_udpdecho_server_lwip	服务器	使用 UDP 协议的回显服务器应用。服务器回传从客户端接收的数据报。
enet_dns_lwip	客户端	一个客户端应用，它请求 DNS (域名服务器) 将域名转换为 IP 地址，从而使 DNS 客户端能够访问源服务器。
enet_snmp_lwip	客户端	基于 SNMP (简单网络时间协议) 报告当前网络时间的客户端应用。
enet_tcpecho_client_lwip	客户端	使用 TCP 协议的回显客户端应用。客户端向服务器发送问候消息，并回传从服务器接收的数据包。
enet_adcensor_client_lwip	客户端	一个客户端应用，它使用片上 ADC 定期获取器件的温度读数并将其发送到服务器。
enet_udpdecho_client_lwip	客户端	使用 UDP 协议的回显客户端应用。客户端向服务器发送问候消息，并回传从服务器接收的数据报。

3 应用设置

3.1 硬件设置

- 连接到互联网的网络路由器
- 将设备连接到 LAN 网络的以太网交换机。如果可以将设备直接连接到路由器，则交换机是可选项。大多数家用路由器都有 LAN 端口，用于与设备建立有线连接。
- PC 用于：
 - 调试目标器件。
 - 充当服务器或客户端，以生成或响应来往于目标设备的网络流量。
 - 监控网络流量。
- 运行本文档中所提供应用的 EK-TM4C1294XL LaunchPad 评估套件。

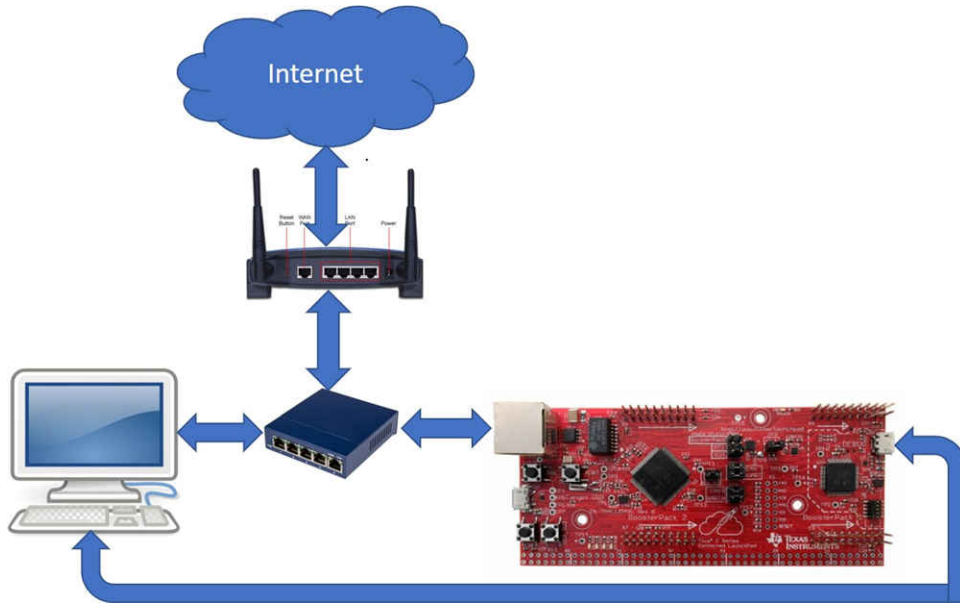


图 3-1. 应用示例的硬件设置

3.2 软件工具

有几种工具可以简化应用示例的调试和测试：

- **CCS**。该 IDE 工具用于调试目标器件和编译应用示例。本应用报告使用 CCS v10.1.1 和 TI v20.2.4.LTS 编译器。
- **SocketTest**。一个免费的小型软件工具，用于测试使用 TCP 或 UDP 协议进行通信的任何服务器或客户端。
- **Wireshark**。此应用报告中用于网络故障排除和分析的免费数据包分析器。
- 终端仿真器。应用示例使用“Terminal”窗口来显示其输出。可以使用任何串行终端仿真器。此应用报告使用 CCS 的一个内置终端仿真器，可以使用“View”->“Terminal”来调用“Terminal”窗口。

4 下载并导入以太网示例

本文件附加了八个 CCS 工程示例作为配套资料。点击以下 URL：<https://www.ti.com/cn/lit/zip/spna248> 来下载示例。您可以解压缩该工程，或将其保留为 zip 格式。两种格式都可以导入 CCS。

1. 若要将工程导入 CCS，请首先选择“File”（文件）->“Import”（导入）。

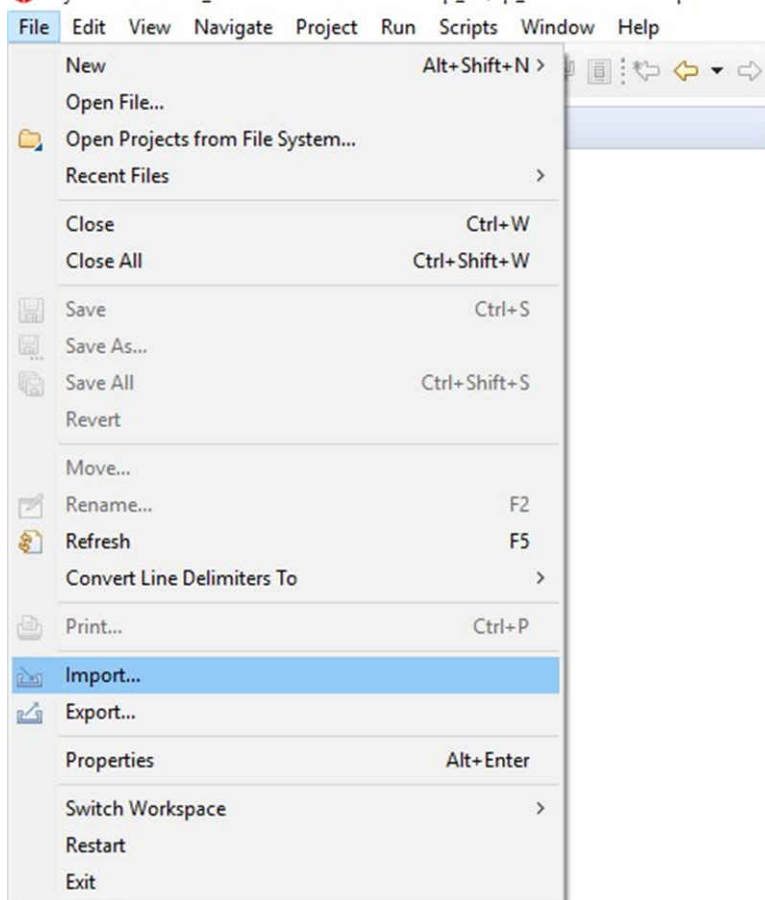


图 4-1. 导入 CCS 工程步骤 1

2. 选择“CCS Projects”导入示例，然后点击“Next”（下一步）。

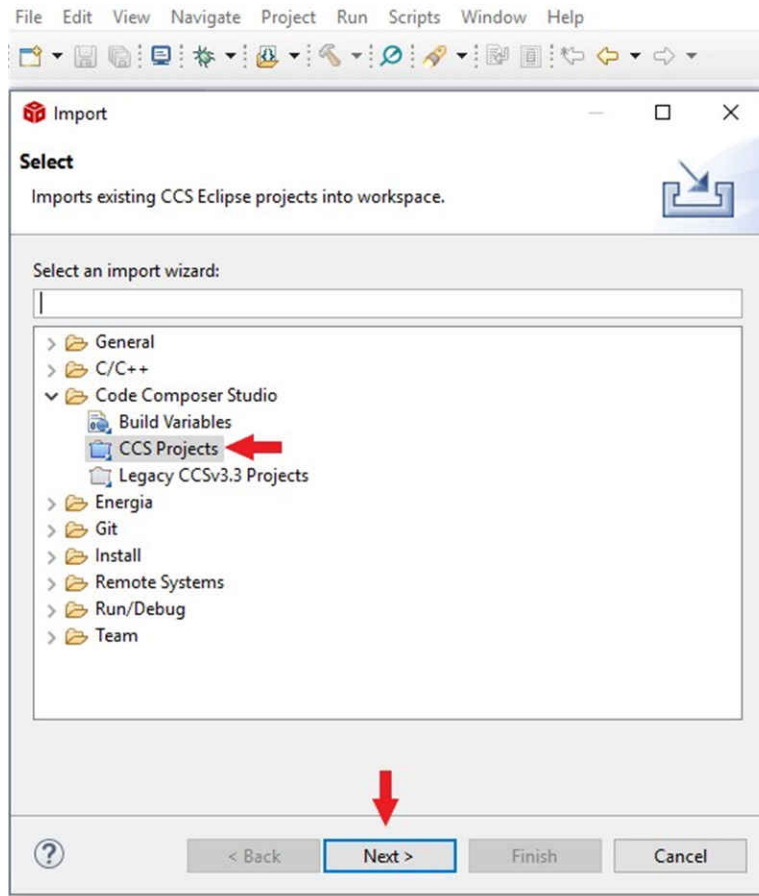


图 4-2. 导入 CCS 工程步骤 2

3. 接下来，提供解压缩工程（选择第一个单选按钮）或直接导入 zip 文件（选择第二个单选按钮）的路径。点击“Copy projects into workspace”（将工程复制到工作区）。

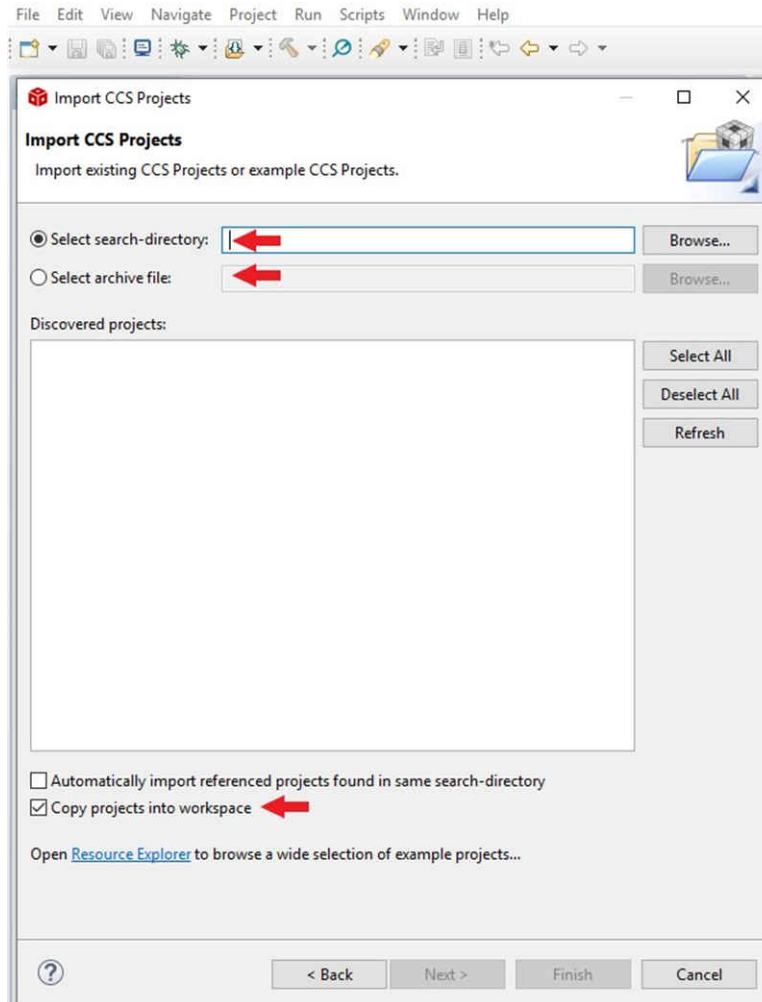


图 4-3. 导入 CCS 工程步骤 3

4. 提供工程路径后，将显示总共八个已发现的工程。首先点击“Select All”（全选）按钮，然后点击“Finish”（结束）按钮完成导入。

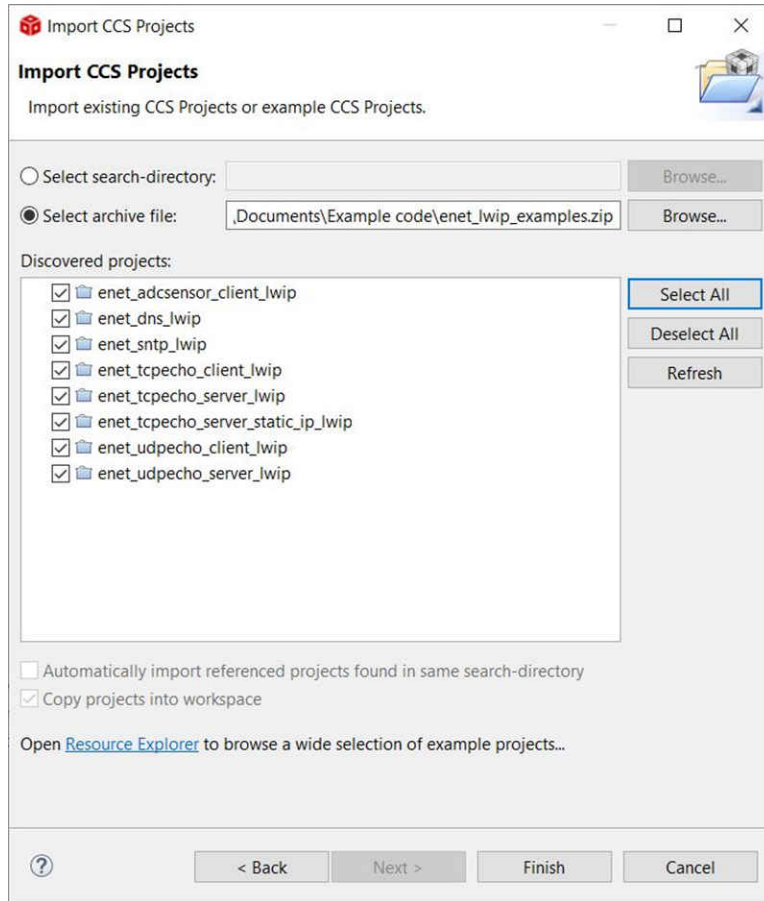


图 4-4. 导入 CCS 工程步骤 4

5 Enet_tcpecho_server_lwip 示例概述

enet_tcpecho_server_lwip 示例演示了一个在 TM4C129x MCU 上运行、使用 TCP 传输控制协议 (TCP) 作为底层传输层协议的回显服务器应用。TCP 是一种面向连接的协议，具有内置的错误恢复和重新传输功能。这种连接协议类似于电话连接。拨打方和接听方都需要握手连接（例如，拨打方拨号，接听方拿起电话）才能交流。连接一直存在，直到一方挂断连接。当需要保证无差错的消息传递时，应用使用 TCP。

在本例中，TM4C129x MCU 用作服务器。lwIP 栈配置为 DHCP，以自动获取 IP 地址。获取后，IP 地址将显示在“Terminal”窗口中。此时，回显服务器准备就绪。服务器将侦听来自客户端的连接。一旦客户端建立起连接，服务器和客户端之间的通信就可以开始。本例中实现的服务器将处理收到的字符，方法是先反转大小写，然后再将反转字符回显到客户端。

5.1 构建和刷写程序

首先选择 enet_tcpecho_server_lwip 作为当前工程。将 LaunchPad 的 ICDI USB 端口通过 USB 电缆连接到 PC，通过点击“Debug”图标编译工程并加载程序。

如果您是 CCS 新手，请点击 [CCS 用户指南](#)。

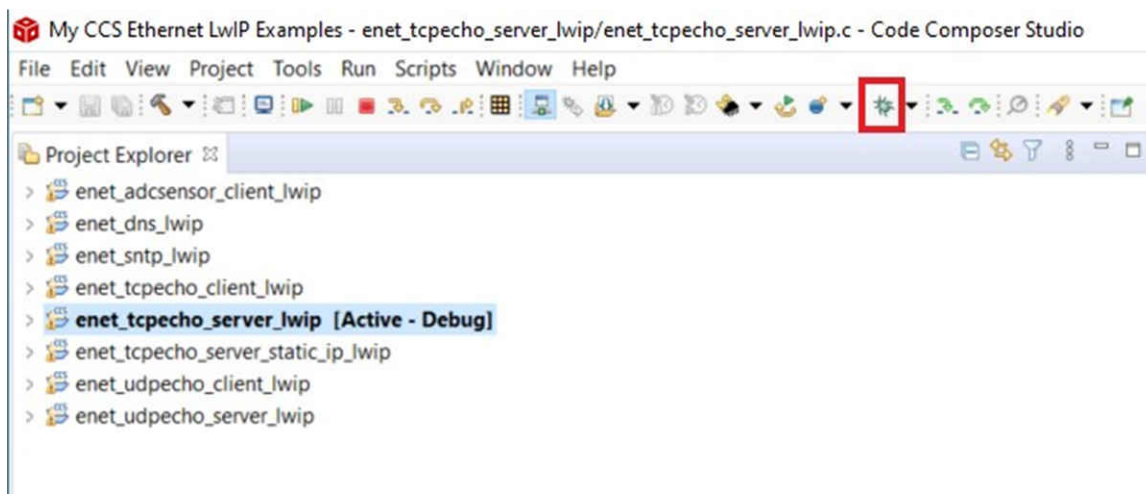


图 5-1. 调试 CCS 工程

5.2 对 MAC 地址进行检查和编程

网络上的每个网络接口控制器 (NIC) 都必须由一个 MAC 地址唯一标识，以便在网段内进行通信。MAC 地址是一个 48 位值，表示为两个十六进制数字的六个八位字节。MAC 地址主要由设备制造商来分配。前三个八位字节是组织唯一标识符 (OUI)。MAC 地址通常在 EK-TM4C1294XL LaunchPad 板上预先编程。LaunchPad 的背面还有一个贴纸，上面写着 MAC 地址。预编程的 MAC 地址的前三个八位字节等于 00:1A:B6，用来唯一标识德州仪器 (TI)。如果您拥有原始状态的设备，则 MAC 地址未预先编程。您必须自行使用分配给贵组织的地址对 MAC 地址进行编程。

提供三种工具可用于对 MAC 地址进行读取和编程。

5.2.1 使用 LM Flash Programmer

如果您在使用 EK-TM4C1294XL LaunchPad，此工具最合适。LM Flash Programmer 仅支持 LaunchPad 上内置的 ICDI 调试探针。若要对 MAC 地址进行检查和编程，须遵循以下步骤。

1. 打开 LM Flash Programmer 并转到“Other Utilities”（其它实用程序）选项卡。
2. 选择“MAC Address Mode”（MAC 地址模式）单选按钮。
3. 若要读取 MAC 地址，请按下“Get Current MAC Address”（获取当前 MAC 地址）按钮。
4. 若要对 MAC 地址进行编程：
 - a. 在“MAC Address”（MAC 地址）字段中键入六个八字字节的 MAC 地址。
 - b. 点击“Commit MAC Address”（提交 MAC 地址）复选框。提交后，将在内部 EEPROM 上永久存储 MAC 地址。如果未选中该提交选项，则刚刚输入的 MAC 地址将在下一次下电上电前临时保存。
 - c. 按“Program MAC Address”（编程 MAC 地址）按钮完成编程。

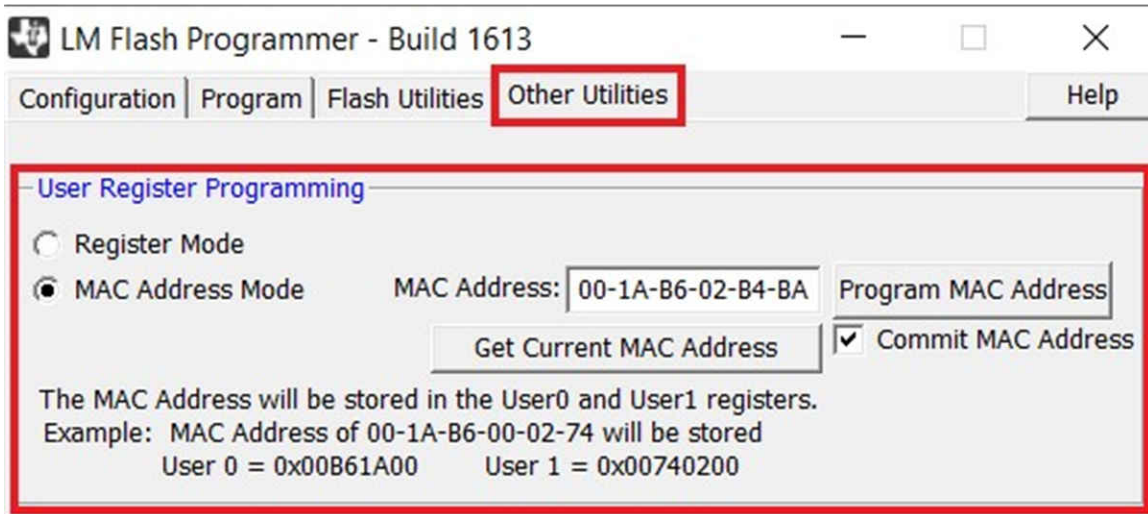


图 5-2. 使用 LM Flash Programmer 进行 MAC 地址编程

5.2.2 使用 CCS

CCS 还具有一个内置实用程序，可用于对 MAC 地址进行编程。如果您选择使用 CCS IDE 进行软件开发，这会非常合适。

若要对 MAC 地址进行读取和编程，请先转到“Tools”->“On-Chip Flash”。对 MAC 地址进行读取和编程的步骤与节 5.2.1 中所述的步骤相同。

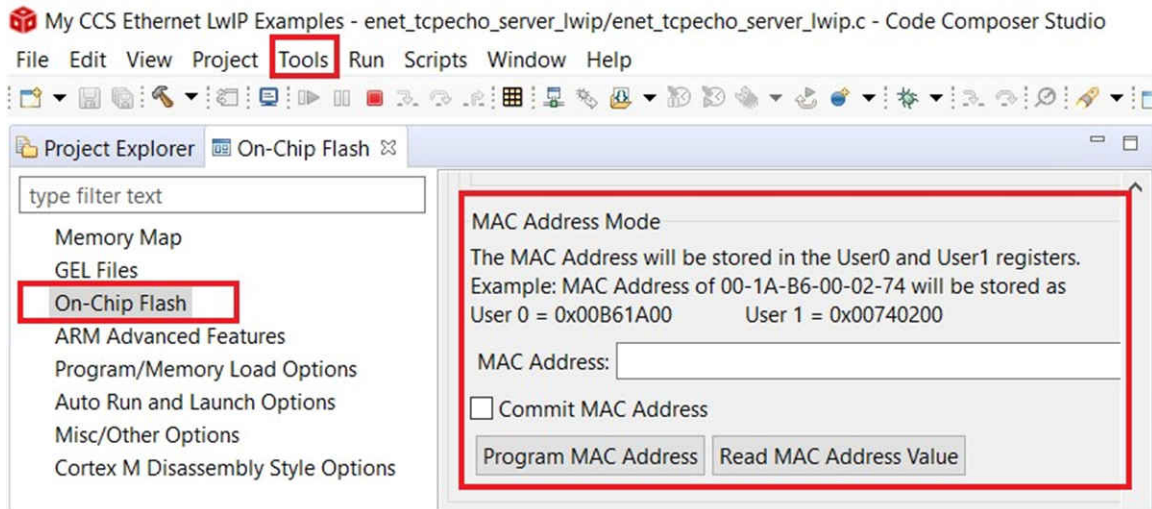


图 5-3. 使用 CCS 对 MAC 地址进行编程

5.2.3 使用 UniFlash

UniFlash 是 TI 的一款独立工具，支持对各种 TI 器件进行编程，包括 TM4C129x MCU 的 MAC 地址。在定制电路板上对 MAC 地址进行编程而调试探针不是 ICD1 时，最适合使用 UniFlash，尽管此工具也支持 ICD1。

若要对 MAC 地址进行读取和编程，请先转到“Settings and Utilities”选项卡。对 MAC 地址进行读取和编程的步骤与节 5.2.1 中所述的步骤相同。

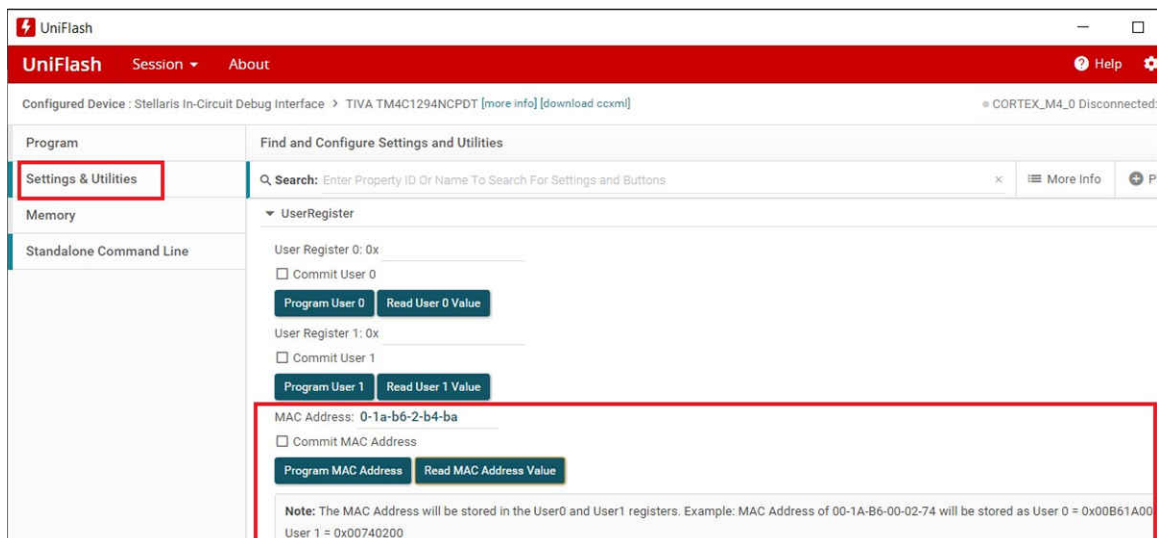


图 5-4. 使用 Uniflash 对 MAC 地址进行编程

5.3 配置终端窗口

该示例显示各种信息，例如获取的 DHCP IP 地址和终端仿真器上的通信进度。您可以使用所选的任何终端仿真器。终端仿真器应配置为 115200 波特和 8-N-1，如图 5-5 所示。

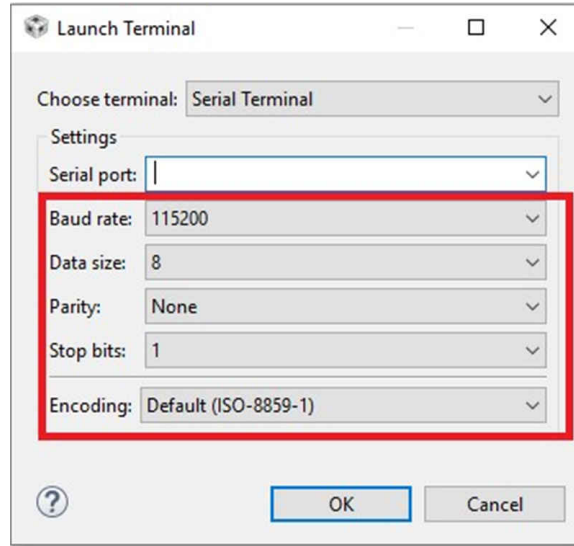


图 5-5. 串行终端设置

5.4 运行 enet_tcpecho_server_lwip 示例

使用以太网电缆将 EK-TM4C1294XL LaunchPad 连接到以太网交换机或路由器，如图 5-6 所示。运行示例。打开“Terminal”窗口后，您应该会看到显示的 IP 地址（箭头 1 所指），并且服务器已准备就绪，如图 5-6 所示。记录此 IP 地址，因为客户端上需要此信息。最初，服务器将处于侦听状态，等待客户端与其连接因此，若要继续运行示例的其余部分，需要设置远程客户端。

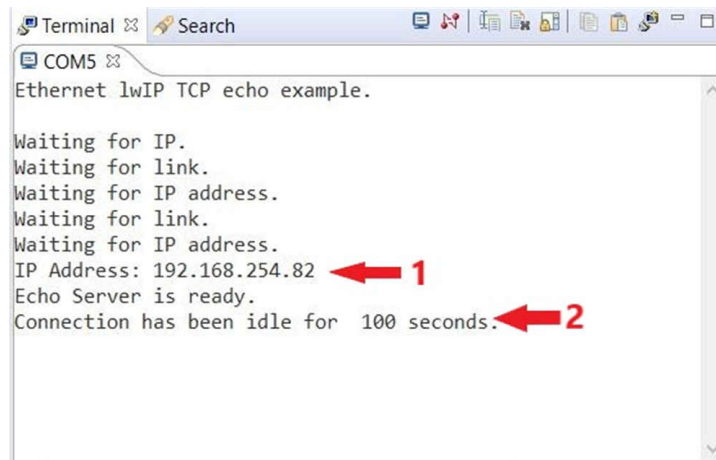


图 5-6. Enet_tcpecho_server_lwip 输出

使用的 SocketTest 工具将充当在 PC 上运行的客户端。确保 PC 连接到与 EK-TM4C1294XL 具有相同子网掩码的网络。

按照图 5-7 中所示的步骤设置客户端：

1. 打开 SocketTest 并输入服务器 IP 地址以及端口号 23。端口 23 是 TCP 和 UDP 协议中的默认 Telnet 端口号。最后，按“Connect”（连接）按钮。与服务器的连接很快就会建立，然后您便可以与服务器进行对话了。
2. 转到“Message”（消息）字段并输入一些消息，然后点击“Send”（发送）按钮。
3. 您输入的消息将显示在对话字段中。当服务器收到消息时，它将反转消息的大小写，然后将消息回显给客户端。服务器还会向客户端回复它收到的字符数。

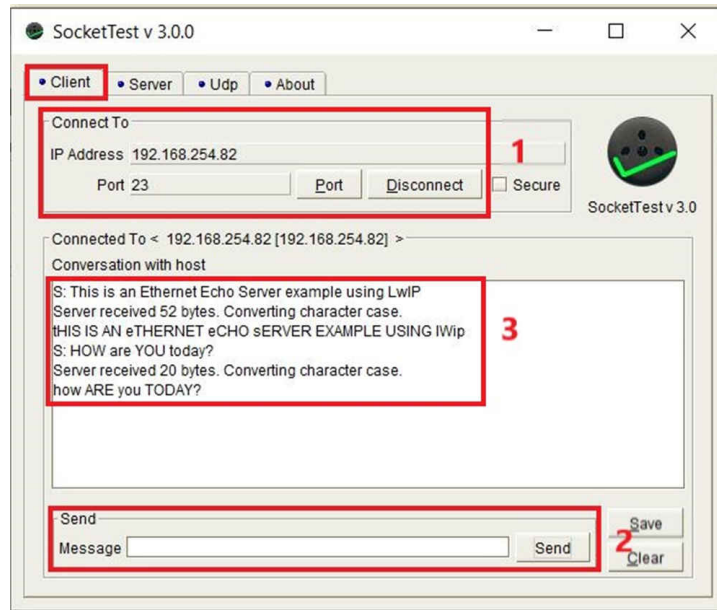


图 5-7. Enet_tcpecho_server_lwip 的 SocketTest 客户端配置

检查客户端发送给服务器的第二条消息“HOW are YOU today?”。如果您选择手动计数，可以计算出总长度为 20 个字节，包括两个 \n\r 转义字符。 \n 是换行符，\r 是 ASCII 表中的回车符。

服务器回复了一条消息，称“Server received 20 bytes. Converting character case. how ARE you TODAY?”。首先，服务器收到的字符数确实是 20。服务器发回的整个消息的总长度为 73 个字节。

No.	Time	Source	Destination	Protocol	Length	Info
702	95.623365	192.168.254.75	192.168.254.82	TELNET	106	Telnet Data ...
703	95.635308	192.168.254.82	192.168.254.75	TELNET	159	Telnet Data ...
704	95.676244	192.168.254.75	192.168.254.82	TCP	54	58696 → 23 [ACK] Seq=53 Ack=106 Win=64135 Len=0
848	110.146923	192.168.254.75	192.168.254.82	TELNET	74	Telnet Data ...
849	110.491747	192.168.254.82	192.168.254.75	TELNET	127	Telnet Data ...
850	110.532014	192.168.254.75	192.168.254.82	TCP	54	58696 → 23 [ACK] Seq=73 Ack=179 Win=64062 Len=0

```

> Frame 848: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{24DD389D-57EF-43B1-8B10-771DF1}
> Ethernet II, Src: IntelCor_65:b6:75 (8c:c6:81:65:b6:75), Dst: TexasIns_02:b4:ba (00:1a:b6:02:b4:ba)
> Internet Protocol Version 4, Src: 192.168.254.75, Dst: 192.168.254.82
> Transmission Control Protocol, Src Port: 58696, Dst Port: 23, Seq: 53, Ack: 106, Len: 20
  ▾ Telnet
    Data: HOW are YOU today?\r\n
  
```

0000	00 1a b6 02 b4 ba 8c c6 81 65 b6 75 08 00 45 00e-u-E-
0010	00 3c 78 5c 40 00 80 06 00 00 c0 a8 fe 4b c0 a8	<x\@-...K-
0020	fe 52 e5 48 00 17 f6 79 b3 5e 00 00 1a 07 50 18	.R.H...y.^...P
0030	fa 87 7e 1e 00 00 48 4f 57 20 61 72 65 20 59 4f	...HO W are YO
0040	55 20 74 6f 64 61 79 3f 0d 0a	U today? ..

图 5-8. Enet_tcpecho_server_lwip 的客户端到服务器 Wireshark 捕获

图 5-9 显示从客户端 (IP 地址 192.168.254.75) 到服务器 (IP 地址 192.168.254.82) 的相应消息。

当服务器发送消息时，它会等待客户端确认数据已成功接收。在 tcp_sent() 调用的回调函数中，服务器只在“Terminal”窗口上显示客户端确认的字节数，如图 5-9 所示。

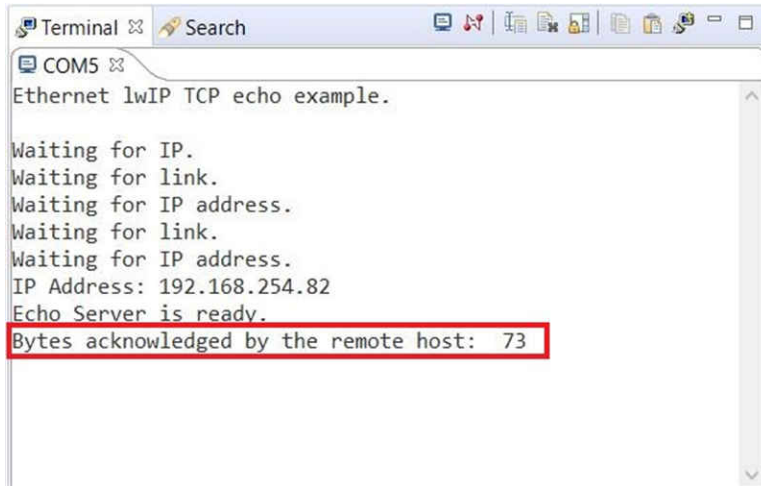


图 5-9. Enet_tcpecho_server_lwip 中的客户端收到数据确认

另外，检查图 5-10 中服务器响应的 Wireshark 捕获。服务器 (IP 地址 192.168.254.82) 向客户端 (192.168.254.75) 发送 73 个字符的消息，客户端确认收到数据。

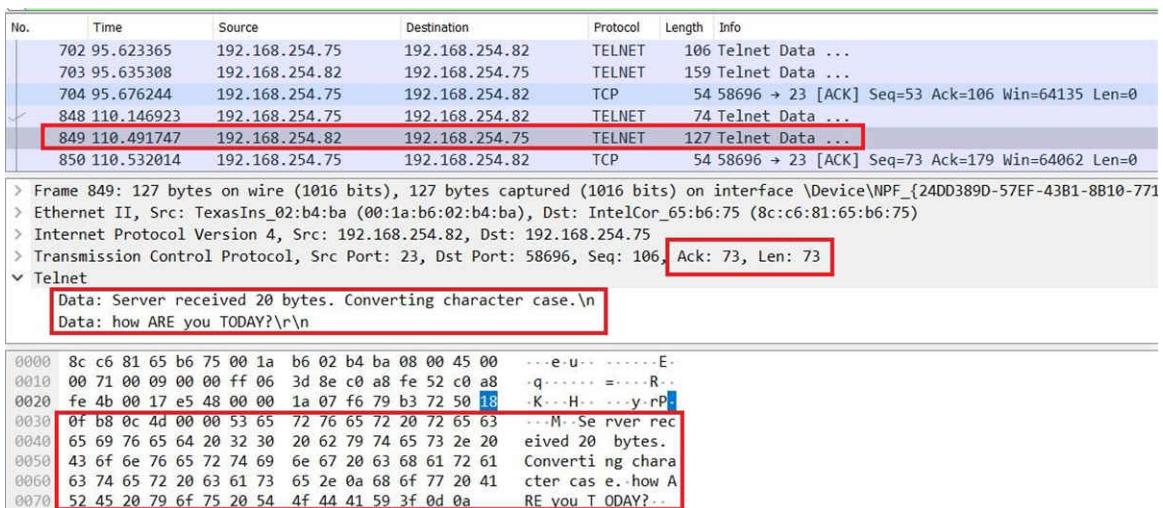


图 5-10. Enet_tcpecho_server_lwip 的服务器到客户端 Wireshark 捕获

假设您让连接空闲，则服务器应用利用 tcp_poll() 每 5 秒定时触发回调函数，以显示服务器空闲了多长时间。在实际应用中，可以在关闭连接之前对内部空闲连接进行编程以节省电量。请参阅图 5-6 中的框 2，其中报告了如下内容：服务器自上个事务后已空闲 100 秒。

6 Enet_tcpecho_server_static_ip_lwip 示例概述

enet_tcpecho_server_static_ip_lwip 示例与 enet_tcpecho_server_lwip 非常相似，只是对服务器使用了静态 IP 地址。需要静态 IP 地址的原因有很多。例如，设备设置为 FTP 或 Web 服务器。您需要确保用户始终可以访问服务器。如果为服务器分配了动态地址，则该地址可能偶尔会更改，这将使路由器无法知道网络上的哪个设备是服务器。

为家庭或专用网络上的本地设备分配静态 IP 地址时，应从 Internet 协议 (IP) 定义的专用 IP 地址范围中选取。静态地址应限制在定义的范围內：

- 10.0.0.0 - 10.255.255.255
- 172.16.0.0 - 172.31.255.255
- 192.168.0.0 - 192.168.255.255

确保所选的静态地址未被专用网络中的任何其他设备使用，否则将导致地址冲突错误。请咨询网络系统管理员来选择静态 IP 地址。

如果要试验静态 IP 地址，但不确定哪个地址当前在网络中未被使用，可以先运行 enet_tcpecho_server_lwip，让 DHCP 选择 IP 地址。此动态地址将租给您的设备一段时间，这将确保在地址过期之前，同一网络上的其他设备都不会获得相同的地址。记录此地址并用其生成静态地址。

6.1 如何将 lwIP 配置为静态地址

lwipopts.h 文件允许启用或禁用不同的功能。若要使用静态 IP 地址，应遵循以下三个步骤：

1. 需要在 lwipopts.h 文件中禁用 DHCP 和 AUTOIP 功能。

```
#define LWIP_DHCP                0
#define LWIP_AUTOIP              0
```

2. 在 enet_tcpecho_server_static_ip_lwip.c 文件中定义静态 IP 地址、子网掩码和网关掩码。请注意，以下地址只是一个示例。用户必须根据自己的网络更改静态地址，否则，该示例将不起作用。

```
#define IPADDR "82.254.168.192"
#define NETMASK "0.0.255.255"
#define GWMASK "0.0.255.255"
#define PORT 23
```

3. 使用 IPADDR_USE_STATIC 标志初始化 lwIP。

```
lwIPInit(g_ui32SysClock, pui8MACArray,
         inet_addr(IPADDR),
         inet_addr(NETMASK),
         inet_addr(GWMASK),
         IPADDR_USE_STATIC);
```

4. 重新编译工程，然后您就可以运行示例了。

6.2 运行 enet_tcpecho_server_static_ip_lwip 示例

因为这个示例与 enet_tcpecho_server_lwip 非常相似，所以它只显示捕获，而不给出详细解释。

图 5-6 和图 6-1 的主要区别是，应用会立即显示静态 IP 地址，因为无需等待 DHCP 服务器返回地址。

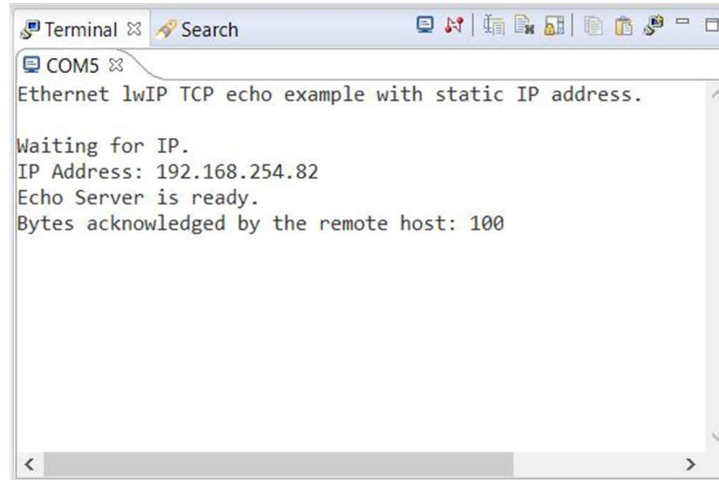


图 6-1. Enet_tcpecho_server_static_ip_lwip 输出

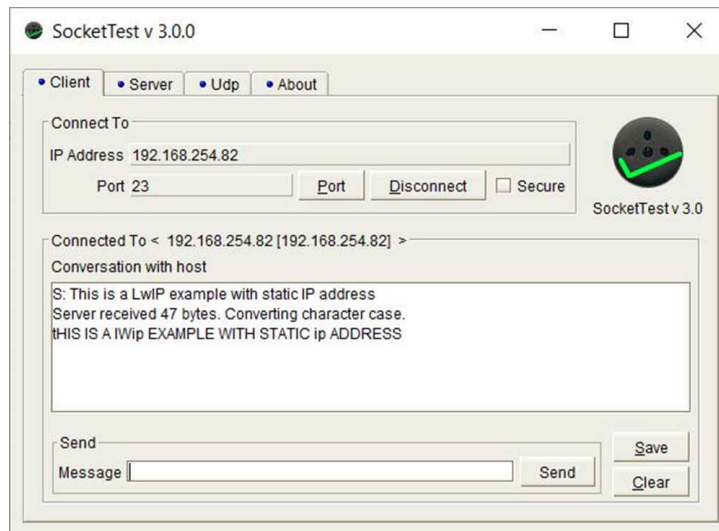


图 6-2. Enet_tcpecho_server_static_ip_lwip 的 SocketTest 客户端配置

No.	Time	Source	Destination	Protocol	Length	Info
203	27.038007	192.168.254.82	192.168.254.75	TCP	60	23 → 52588 [SYN, ACK] Seq=0 Ack=1 Win=4096 Len=0
204	27.038557	192.168.254.75	192.168.254.82	TCP	54	52588 → 23 [ACK] Seq=1 Ack=1 Win=64240 Len=0
207	27.078990	192.168.254.75	192.168.254.82	NBNS	92	Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00>
210	27.082451	192.168.254.82	192.168.254.75	ICMP	70	Destination unreachable (Port unreachable)
220	28.579153	192.168.254.75	192.168.254.82	NBNS	92	Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00>
221	28.585382	192.168.254.82	192.168.254.75	ICMP	70	Destination unreachable (Port unreachable)
225	30.079425	192.168.254.75	192.168.254.82	NBNS	92	Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00>
227	30.085110	192.168.254.82	192.168.254.75	ICMP	70	Destination unreachable (Port unreachable)
403	80.319757	192.168.254.75	192.168.254.82	TELNET	101	Telnet Data ...
404	80.373078	192.168.254.82	192.168.254.75	TELNET	154	Telnet Data ...
405	80.413842	192.168.254.75	192.168.254.82	TCP	54	52588 → 23 [ACK] Seq=48 Ack=101 Win=64140 Len=0
999	231.585712	192.168.254.75	192.168.254.82	TELNET	101	Telnet Data ...
1000	231.626848	192.168.254.82	192.168.254.75	TELNET	154	Telnet Data ...
1002	231.666958	192.168.254.75	192.168.254.82	TCP	54	52588 → 23 [ACK] Seq=95 Ack=201 Win=64040 Len=0

> Frame 403: 101 bytes on wire (808 bits), 101 bytes captured (808 bits) on interface \Device\NPF_{24DD389D-57EF-4381-8B10-771DF} Ethernet II, Src: IntelCor_65:b6:75 (8c:c6:81:65:b6:75), Dst: TexasIns_02:b4:ba (00:1a:b6:02:b4:ba)

Internet Protocol Version 4, Src: 192.168.254.75, Dst: 192.168.254.82

0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
 > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 87
 Identification: 0x7848 (30792)

```

0000 00 1a b6 02 b4 ba 8c c6 81 65 b6 75 08 00 45 00  ....e.u...E-
0010 00 57 78 48 40 00 80 06 00 00 c0 a8 fe 4b c0 a8  ..w.H... ..K..
0020 fe 52 cd 6c 00 17 b3 7b 02 25 00 00 19 7c 50 18  -R.l...{.%...|P-
0030 fa f0 7e 39 00 00 54 68 69 73 20 69 73 20 61 20  --9...th is is a
0040 4c 77 49 50 20 65 78 61 6d 70 6c 65 20 77 69 74  LwIP exam ple wit
0050 68 20 73 74 61 74 69 63 20 49 50 20 61 64 64 72  h static IP addr
0060 65 73 73 0d 0a  ....ess--
  
```

图 6-3. Enet_tcpecho_server_static_ip_lwip 的客户端到服务器 Wireshark 捕获

No.	Time	Source	Destination	Protocol	Length	Info
203	27.038007	192.168.254.82	192.168.254.75	TCP	60	23 → 52588 [SYN, ACK] Seq=0 Ack=1 Win=4096 Len=0
204	27.038557	192.168.254.75	192.168.254.82	TCP	54	52588 → 23 [ACK] Seq=1 Ack=1 Win=64240 Len=0
207	27.078990	192.168.254.75	192.168.254.82	NBNS	92	Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00>
210	27.082451	192.168.254.82	192.168.254.75	ICMP	70	Destination unreachable (Port unreachable)
220	28.579153	192.168.254.75	192.168.254.82	NBNS	92	Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00>
221	28.585382	192.168.254.82	192.168.254.75	ICMP	70	Destination unreachable (Port unreachable)
225	30.079425	192.168.254.75	192.168.254.82	NBNS	92	Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00>
227	30.085110	192.168.254.82	192.168.254.75	ICMP	70	Destination unreachable (Port unreachable)
403	80.319757	192.168.254.75	192.168.254.82	TELNET	101	Telnet Data ...
404	80.373078	192.168.254.82	192.168.254.75	TELNET	154	Telnet Data ...
405	80.413842	192.168.254.75	192.168.254.82	TCP	54	52588 → 23 [ACK] Seq=48 Ack=101 Win=64140 Len=0
999	231.585712	192.168.254.75	192.168.254.82	TELNET	101	Telnet Data ...
1000	231.626848	192.168.254.82	192.168.254.75	TELNET	154	Telnet Data ...
1002	231.666958	192.168.254.75	192.168.254.82	TCP	54	52588 → 23 [ACK] Seq=95 Ack=201 Win=64040 Len=0

Protocol: TCP (6)
 Header Checksum: 0x3d76 [validation disabled]
 [Header checksum status: Unverified]
 Source Address: 192.168.254.82
 Destination Address: 192.168.254.75
 > Transmission Control Protocol, Src Port: 23, Dst Port: 52588, Seq: 1, Ack: 48, Len: 100
 ✓ Telnet
 Data: Server received 47 bytes. Converting character case.\n

```

0000 8c c6 81 65 b6 75 00 1a b6 02 b4 ba 08 00 45 00  ...e.u... ..E-
0010 00 8c 00 06 00 00 ff 06 3d 76 c0 a8 fe 52 c0 a8  ....=v...R..
0020 fe 4b 00 17 cd 6c 00 00 19 7c b3 7b 02 54 50 18  -K...l...|...TP-
0030 0f d1 41 47 00 00 53 65 72 76 65 72 20 72 65 63  --AG...Se rver rec
0040 65 69 76 65 64 20 34 37 20 62 79 74 65 73 2e 20  eived 47 bytes.
0050 43 6f 6e 76 65 72 74 69 6e 67 20 63 68 61 72 61  Converting chara
0060 63 74 65 72 20 63 61 73 65 2e 0a 74 48 49 53 20  cten cas e...THIS
0070 49 53 20 41 20 6c 57 69 70 20 45 58 41 4d 50 4c  IS A lwi p EXAMPL
0080 45 20 57 49 54 48 20 53 54 41 54 49 43 20 69 70  E WITH S TATIC ip
0090 20 41 44 44 52 45 53 53 0d 0a  ....ADDRESS ..
  
```

图 6-4. Enet_tcpecho_server_static_ip_lwip 的服务器到客户端 Wireshark 捕获

7 Enet_udpecho_server_lwip 示例概述

UDP (用户数据报协议) 是另一种众所周知的传输层协议。enet_udpecho_server_lwip 示例演示了使用 UDP 协议在 TM4C129x MCU 上运行的一个回显服务器应用。UDP 是一种不涉及错误恢复和重新传输的无连接协议。可以将无连接协议比作邮寄信件。您将信件投入邮局信箱中, 这封信件最终能否送到收件人手里, 以及是否完好无损 (例如, 下雨或处理不当可能导致信件损坏) 都无法保证。

7.1 运行 enet_udpecho_server_lwip 示例

使用的 SocketTest 工具充当在 PC 上运行的客户端。确保 PC 连接到与 EK-TM4C1294XL 具有相同子网掩码的网络。

按照图 7-2 中所示的步骤设置 SocketTest :

1. 转至“Client” (客户端) 选项卡。
2. 输入 PC 的 IP 地址和端口号 23, 然后按下“Start Listening” (开始侦听) 按钮。服务器 IP 地址应为运行 SocketTest 的 PC 的地址。若要查找 PC 在网络中的 IP 地址, 可以使用 Windows 的 ipconfig 命令。打开一个 Windows 命令窗口, 在提示符下键入“ipconfig”, 然后您将看到分配给您的 PC 的 IP 地址。例如, 参阅图 7-1。请注意, 在 SocketTest 中, 无论 PC 是实际的服务器还是客户端, 服务器地址字段都将是 PC 的地址。SocketTest 只侦听指定地址和端口处的任何传入数据。
3. 在“IP Address for Client”中输入 MCU 的 IP 地址。分配给 MCU 的 IP 地址显示在“Terminal”窗口中。
4. 转到“Message” (消息) 字段并输入一些消息, 然后点击“Send” (发送) 按钮。
5. 观察 SocketTest 中的对话字段以及图 7-3 和图 7-4 中的 Wireshark 捕获。

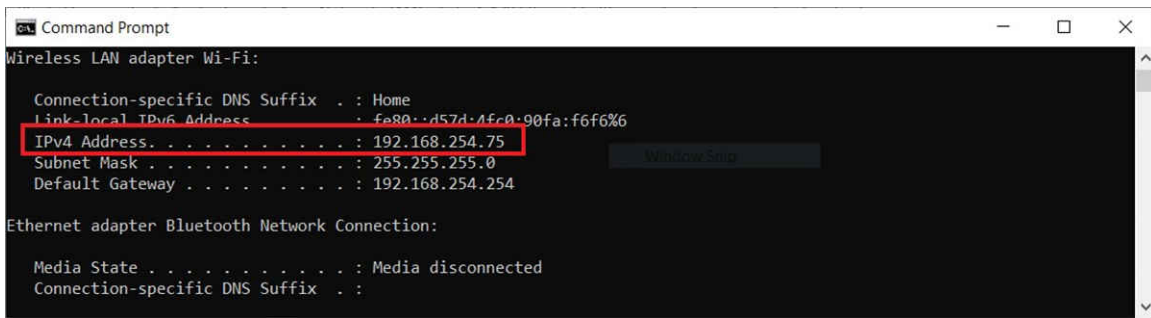


图 7-1. 查询 PC 的 IP 地址

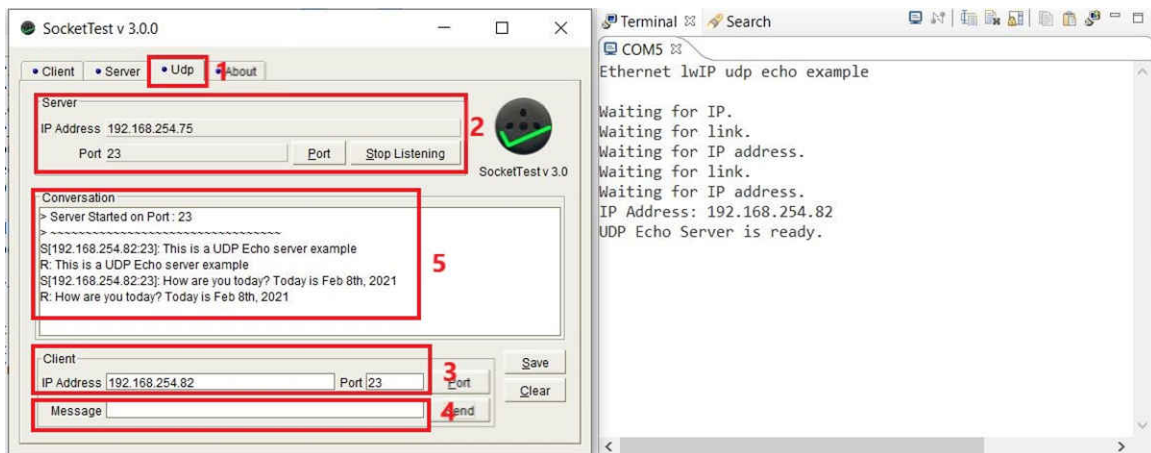


图 7-2. Enet_udpecho_server_lwip 输出

No.	Time	Source	Destination	Protocol	Length	Info
213	45.877681	0.0.0.0	255.255.255.255	DHCP	350	DHCP Discover - Transaction ID 0xabcd0002
214	45.877764	0.0.0.0	255.255.255.255	DHCP	350	DHCP Request - Transaction ID 0xabcd0003
215	45.877792	TexasIns_02:b4:ba	Broadcast	ARP	60	Who has 192.168.254.82? (ARP Probe)
219	46.490029	TexasIns_02:b4:ba	Broadcast	ARP	60	Who has 192.168.254.82? (ARP Probe)
384	71.262898	192.168.254.75	192.168.254.82	UDP	75	57326 → 23 Len=33
385	71.476539	TexasIns_02:b4:ba	Broadcast	ARP	60	Who has 192.168.254.75? Tell 192.168.254.82
386	71.476690	IntelCor_65:b6:75	TexasIns_02:b4:ba	ARP	42	192.168.254.75 is at 8c:c6:81:65:b6:75
387	71.482102	192.168.254.82	192.168.254.75	UDP	75	23 → 57326 Len=33
401	75.923164	IntelCor_65:b6:75	TexasIns_02:b4:ba	ARP	42	Who has 192.168.254.82? Tell 192.168.254.75
402	75.928940	TexasIns_02:b4:ba	IntelCor_65:b6:75	ARP	64	192.168.254.82 is at 00:1a:b6:02:b4:ba

Time to Live: 128
 Protocol: UDP (17)
 Header Checksum: 0x0000 [validation disabled]
 [Header checksum status: Unverified]
 Source Address: 192.168.254.75
 Destination Address: 192.168.254.82

User Datagram Protocol, Src Port: 57326, Dst Port: 23

```

0000  00 1a b6 02 b4 ba 8c c6 81 65 b6 75 08 00 45 00  .....e.u..E-
0010  00 3d 78 53 00 00 80 11 00 00 c0 a8 fe 4b c0 a8  -xS.....K..
0020  fe 52 df ee 00 17 00 29 7e 2a 54 68 69 73 20 69  -R.....)~*This i
0030  73 20 61 20 55 44 50 20 45 63 68 6f 20 73 65 72  s a UDP Echo ser
0040  76 65 72 20 65 78 61 6d 70 6c 65                ver exam ple
  
```

图 7-3. Enet_udpecho_server_lwip 的客户端到服务器 Wireshark 捕获

No.	Time	Source	Destination	Protocol	Length	Info
213	45.877681	0.0.0.0	255.255.255.255	DHCP	350	DHCP Discover - Transaction ID 0xabcd0002
214	45.877764	0.0.0.0	255.255.255.255	DHCP	350	DHCP Request - Transaction ID 0xabcd0003
215	45.877792	TexasIns_02:b4:ba	Broadcast	ARP	60	Who has 192.168.254.82? (ARP Probe)
219	46.490029	TexasIns_02:b4:ba	Broadcast	ARP	60	Who has 192.168.254.82? (ARP Probe)
384	71.262898	192.168.254.75	192.168.254.82	UDP	75	57326 → 23 Len=33
385	71.476539	TexasIns_02:b4:ba	Broadcast	ARP	60	Who has 192.168.254.75? Tell 192.168.254.82
386	71.476690	IntelCor_65:b6:75	TexasIns_02:b4:ba	ARP	42	192.168.254.75 is at 8c:c6:81:65:b6:75
387	71.482102	192.168.254.82	192.168.254.75	UDP	75	23 → 57326 Len=33
401	75.923164	IntelCor_65:b6:75	TexasIns_02:b4:ba	ARP	42	Who has 192.168.254.82? Tell 192.168.254.75
402	75.928940	TexasIns_02:b4:ba	IntelCor_65:b6:75	ARP	64	192.168.254.82 is at 00:1a:b6:02:b4:ba
495	104.820366	192.168.254.75	192.168.254.82	UDP	83	57326 → 23 Len=41
496	104.827160	192.168.254.82	192.168.254.75	UDP	83	23 → 57326 Len=41

Time to Live: 255
 Protocol: UDP (17)
 Header Checksum: 0x3dbc [validation disabled]
 [Header checksum status: Unverified]

```

0000  8c c6 81 65 b6 75 00 1a b6 02 b4 ba 08 00 45 00  ..e.u.. .....E-
0010  00 3d 00 04 00 00 ff 11 3d bc c0 a8 fe 52 c0 a8  -.....=...R..
0020  fe 4b 00 17 df ee 00 29 91 2b 54 68 69 73 20 69  -K.....)~+This i
0030  73 20 61 20 55 44 50 20 45 63 68 6f 20 73 65 72  s a UDP Echo ser
0040  76 65 72 20 65 78 61 6d 70 6c 65                ver exam ple
  
```

图 7-4. Enet_udpecho_server_lwip 的服务器到客户端 Wireshark 捕获

8 Enet_dns_lwip 示例概述

连接到网络的每个设备都将具有唯一的 IP 地址，该地址依赖于 IP (互联网协议) 进行通信。但是，IP 地址要么是 IPv4 32 位地址，要么是 IPv6 128 位地址，很难让人记住。DNS (域名系统) 是一种应用层服务，可将“人性化”的域名转换为 IP 地址。人类很容易记住 www.ti.com 而不是其数字 IP 地址。请注意，DNS 是依赖于 UDP 协议的应用服务，它本身并非是一种协议。

这个简单的示例演示了如何使用 DNS 功能来检索四个不同网站的 IP 地址。

8.1 如何将 lwIP 配置为 DNS

若要使用 DNS，应遵循以下三个步骤：

1. 在 `lwipopts.h` 文件中配置或取消注释以下默认 `#define`。

```
#define LWIP_DNS                1
#define DNS_TABLE_SIZE        4
#define DNS_MAX_NAME_LENGTH   256
#define DNS_MAX_SERVERS      2
#define DNS_DOES_NAME_CHECK   1
#define DNS_USES_STATIC_BUF   1
#define DNS_MSG_SIZE          512
```

2. DNS 依赖于 UDP 协议，所以应用需要包含以下头文件。

```
#include "lwip/udp.h"
#include "lwip/inet.h"
#include "lwip/dns.h"
```

3. 使用 `dns_gethostbyname()` API 转换域名。下面是一个用法示例。查看 `enet_dns_lwip.c` 文件了解详情。

```
err = dns_gethostbyname(host_addresses[current_host], &dns_resolved_address,
                        DnsFound, NULL);
```

4. 重新编译工程，然后您就可以运行示例了。

8.2 如何在 Wireshark 上查看 DNS 流量

如果将 MCU 器件和 PC 连接到以太网交换机，则将无法在 Wireshark 上查看 DNS 流量。原因是 DNS 流量位于 DNS 服务器和器件之间。交换机不会将流量路由到运行 Wireshark 的 PC 上。若想查看 DNS 流量，您要么必须有一个以太网集线器，用于将网络上的所有流量广播到连接网络的所有端口，要么需要将以太网交换机配置为“端口镜像”。如今，很难找到以太网集线器这种设备。容易找到的是智能交换机，可以将其配置为“端口镜像”。图 8-1 显示了在何处将连接交换机端口 2 的 EK-TM4C1294XL 镜像到连接 PC 的端口 1。利用端口镜像，进出端口 2 的所有流量都将被端口 1 上的 Wireshark 监视。请查阅交换机或路由器的数据表，了解如何配置端口镜像。

System

Switching

Monitoring

- Port Statistics
- **Port Mirror**
- Cable Test
- Loop Prevention

VLAN

QoS

Logout

Port Mirror

Port Mirror	Mirroring Port
Enable ▾	Port 1 ▾

Apply

Mirrored Port

Mirrored Port	Ingress	Egress
Port 1 ▲		
Port 2	▾	▾
Port 3		
Port 4		
Port 5 ▾		

Apply Help

Mirrored Port	Ingress	Egress
Port1	Disable	Disable
Port2	Enable	Enable

图 8-1. 端口镜像

8.3 运行 enet_dns_lwip 示例

如预期的那样，运行本示例会在“Terminal”窗口中显示相应网站的四个 IP 地址。更多详细信息，请查看 Wireshark 捕获，特别是关于 www.google.com 的部分。

- 在图 8-3 的 Wireshark 捕获中，从突出显示的框 1 和框 2，可以看到与四个网站对应的四个 DNS 数据包。
- 事务的源地址来自 IP 地址 192.168.254.254。这恰好是运行此应用的家庭网络路由器的 IP 地址。智能路由器通常充当 DNS 服务器，将过去访问过的域名存储在其缓存中以便快速检索。
- 在框 3 中，www.google.com 返回的 IP 地址是 172.217.1.132。您可以通过在浏览器的 URL 字段中输入此地址进行确认，它应该会将其引导至 Google 网站。
- 在框 4 中，IP 地址表示为等于 0xACD90184 的 32 位二进制值。0xAC 等于十进制数 172，0xD9 等于十进制数 217，其余部分也是如此。

```
Terminal
COM5
Ethernet lwIP DNS example

Waiting for IP.
Waiting for link.
Waiting for IP address.
IP Address: 192.168.254.86
Starting DNS Service.
DnsRequest: Waiting for server address to be resolved.
hostname: pool.ntp.org           Resolved IP Address: 138.68.201.49
DnsRequest: Waiting for server address to be resolved.
hostname: www.google.com         Resolved IP Address: 172.217.1.132
DnsRequest: Waiting for server address to be resolved.
hostname: api.openweathermap.org Resolved IP Address: 192.241.245.161
DnsRequest: Waiting for server address to be resolved.
hostname: www.ti.com             Resolved IP Address: 104.100.228.145
```

图 8-2. Enet_dns_lwip 输出

No.	Time	Source	Destination	Protocol	Length	Info
276	25.088102	IntelCor_e2:dc:95	TexasIns_02:b4:ba	ARP	60	192.168.254.84 is at 8c:8d:28:e2:dc:95
277	25.138609	TexasIns_02:b4:ba	Broadcast	ARP	60	Who has 192.168.254.86? (ARP Probe)
278	25.291382	IntelCor_e2:dc:95	TexasIns_02:b4:ba	ARP	60	192.168.254.84 is at 8c:8d:28:e2:dc:95
279	25.741995	TexasIns_02:b4:ba	Broadcast	ARP	60	Who has 192.168.254.254? Tell 192.168.254.86
280	25.742366	Actionte_1a:a5:50	TexasIns_02:b4:ba	ARP	60	192.168.254.254 is at e8:6f:f2:1a:a5:50
281	25.762495	192.168.254.254	192.168.254.86	DNS	136	Standard query response 0x0000 A pool.ntp.org A 138.68.201.49 A
282	25.873454	192.168.254.254	192.168.254.86	DNS	90	Standard query response 0x0001 A www.google.com A 172.217.1.132
283	25.910217	IntelCor_e2:dc:95	TexasIns_02:b4:ba	ARP	60	192.168.254.84 is at 8c:8d:28:e2:dc:95
284	25.977811	192.168.254.254	192.168.254.86	DNS	130	Standard query response 0x0002 A api.openweathermap.org A 192.241.245.161
286	26.098093	192.168.254.254	192.168.254.86	DNS	226	Standard query response 0x0003 A www.ti.com CNAME china.www.ti.com


```
> Frame 282: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface \Device\NPF_{00F83936-2E2B-4DCF-93F4-C535993CB419}, id 0
> Ethernet II, Src: Actionte_1a:a5:50 (e8:6f:f2:1a:a5:50), Dst: TexasIns_02:b4:ba (00:1a:b6:02:b4:ba)
> Internet Protocol Version 4, Src: 192.168.254.254, Dst: 192.168.254.86
> User Datagram Protocol, Src Port: 53, Dst Port: 49153
> Domain Name System (response)

0000  00 1a b6 02 b4 ba e8 6f f2 1a a5 50 08 00 45 00  ....o...P..E.
0010  00 4c 00 00 00 00 40 11 fb fa c0 a8 fe fe c0 a8  .L...@.....
0020  fe 56 00 35 c0 01 00 38 42 b4 00 01 81 80 00 01  .V.5..8B.....
0030  00 01 00 00 00 03 77 77 77 06 67 6f 6f 67 6c    ....w ww-googl
0040  65 03 63 6f 6d 00 00 01 00 01 c0 0c 00 01 00 01  e.com.....
0050  00 00 00 52 00 04 ac d9 01 84 4                ...R...-
```

图 8-3. Enet_dns_lwip 的 Wireshark 捕获

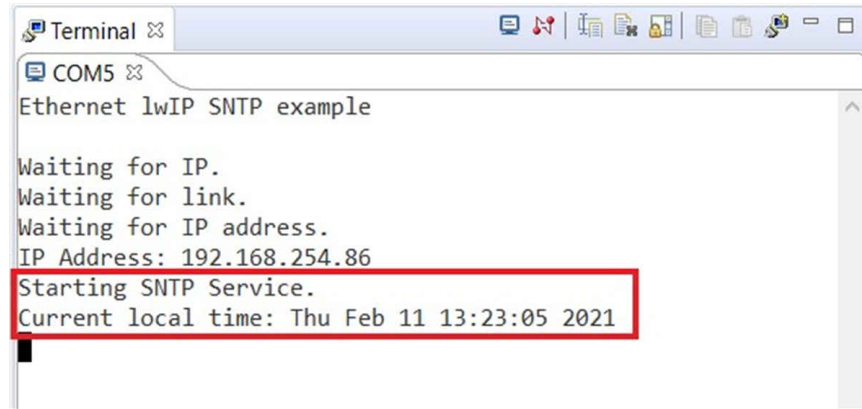
9 Enet_snmp_lwip 示例概述

简单网络时间协议 (SNTP) 是 NTP 的简化版本。SNTP 通常提供精确到 100mS 的时间，无需复杂的 NTP 过滤机制。SNTP 是一种应用层协议，基于 UDP 作为传输层。

本示例使用 SNTP 协议获取网络时间，并在“Terminal”窗口中显示针对北美中部时间 (CT) 区域调整的当前时间。

9.1 运行 enet_snmp_lwip 示例

客户端将请求发送到 pool.ntp.org 网站，从中随机选择一个公共时间服务器来提供网络时间，如图 9-2 所示。获取的网络时间首先针对中央时区进行调整，然后显示在“Terminal”窗口中，如图 9-1 所示。



```

Terminal
COM5
Ethernet lwIP SNTP example

Waiting for IP.
Waiting for link.
Waiting for IP address.
IP Address: 192.168.254.86
Starting SNTP Service.
Current local time: Thu Feb 11 13:23:05 2021
    
```

图 9-1. Enet_snmp_lwip 输出

No.	Time	Source	Destination	Protocol	Length	Info
2816	115.662767	0.0.0.0	255.255.255.255	DHCP	350	DHCP Discover - Transaction ID 0xabcd0001
2817	115.667172	192.168.254.254	192.168.254.86	DHCP	326	DHCP Offer - Transaction ID 0xabcd0001
2818	115.667288	0.0.0.0	255.255.255.255	DHCP	350	DHCP Request - Transaction ID 0xabcd0002
2819	115.675017	192.168.254.254	192.168.254.86	DHCP	326	DHCP ACK - Transaction ID 0xabcd0002
2820	115.675110	TexasIns_02:b4:ba	Broadcast	ARP	60	Who has 192.168.254.86? (ARP Probe)
2823	115.724723	IntelCor_e2:dc:95	TexasIns_02:b4:ba	ARP	60	192.168.254.84 is at 8c:8d:28:e2:dc:95
2825	116.092307	TexasIns_02:b4:ba	Broadcast	ARP	60	Who has 192.168.254.86? (ARP Probe)
2826	116.133094	IntelCor_e2:dc:95	TexasIns_02:b4:ba	ARP	60	192.168.254.84 is at 8c:8d:28:e2:dc:95
2827	116.695674	TexasIns_02:b4:ba	Broadcast	ARP	60	Who has 192.168.254.254? Tell 192.168.254.86
2829	116.696395	Actionte 1a:a5:50	TexasIns_02:b4:ba	ARP	60	192.168.254.254 is at e8:6f:f2:1a:a5:50
2830	116.715972	192.168.254.254	192.168.254.86	DNS	136	Standard query response 0x0000 A pool.ntp.org A 50.205.244.23
2831	116.716105	192.168.254.86	50.205.244.23	NTP	90	NTP Version 4, client
2832	116.747180	IntelCor_e2:dc:95	TexasIns_02:b4:ba	ARP	60	192.168.254.84 is at 8c:8d:28:e2:dc:95

图 9-2. Enet_snmp_lwip 的 Wireshark 捕获

10 Enet_tcpecho_client_lwip 示例概述

enet_tcpecho_client_lwip 示例演示了一个客户端应用，该应用首先连接到服务器并发出问候消息“Hello World!\n\r”，然后回显它从服务器接收到的任何内容。

如图 1-1 中 TCP 的 lwIP 流程图所示，客户端将使用 tcp_connect() 连接到指定的服务器地址和端口。建立连接后，客户端将使用 tcp_recv() 设置回调函数，用于从服务器接收数据，然后回显数据。

流程图中客户端和服务端之间的另一个区别是客户端不需要调用 tcp_bind()。TCP 的客户端通常不需要绑定。某些情况下可能需要绑定客户端，这时可使用 tcp_bind() 来绑定客户端。例如，客户端上的防火墙仅允许在特定端口上进行传出连接。

10.1 配置服务器 IP 地址

若要运行此示例，必须在编译期间知道服务器 IP 地址和端口号，以便客户端与服务器建立连接。

1. 在 enet_tcpecho_client_lwip.c 文件中定义 SERVER_IPADDR 和 SERVER_PORT。请注意，以下地址只是一个示例。您必须将服务器 IP 地址改为要在网络中连接到的地址，否则，该示例将不起作用。

```
#define SERVER_IPADDR "192.168.254.75"  
#define SERVER_PORT 8000
```

2. 重新编译工程，然后您就可以运行示例了。

10.2 配置 SocketTest 服务器并运行 enet_tcpecho_client_lwip 示例

因为这是一个客户端应用，所以必须首先设置服务器，且服务器必须处于侦听状态，这样才能连接客户端。

按照图 10-1 中所示的步骤设置 SocketTest 服务器：

1. 打开 SocketTest 中的“Server”（服务器）选项卡。
2. 输入服务器 IP 地址以及端口号。此 IP 地址是运行 SocketTest 的 PC 的地址。IP 和端口号需要与 enet_tcpecho_client_lwip.c 中定义的设置匹配，如节 10.1 所示。最后，按下“Start Listening”（开始侦听）按钮。等待客户端连接后出现的问候消息对话框。
3. 客户端连接到 SocketTest 服务器后，将发送问候消息“Hello World!\n\r”。服务器接受来自客户端 IP 地址 192.168.254.85 的连接后，图 10-1 中框 3 的对话字段就会显示来自该客户端的消息“Hello World!”。
4. 转到框 4 中所示的 SocketTest 中的 Message 字段并键入一些消息。无论键入什么消息，客户端都会在消息前面加上标题“Client:”进行回显。在本例中，键入的消息是“This is a tcp echo example\n\r”，共 28 个字符。

如果您返回到“Terminal”窗口，它也指示从远程主机接收的相同数量的字符。

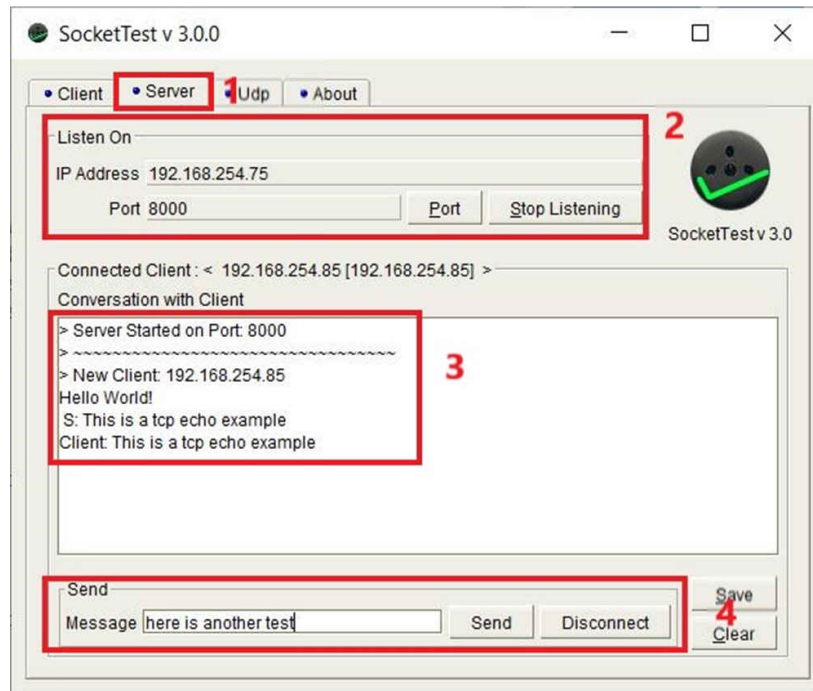


图 10-1. Enet_tcpecho_client_lwip 的 SocketTest 服务器配置

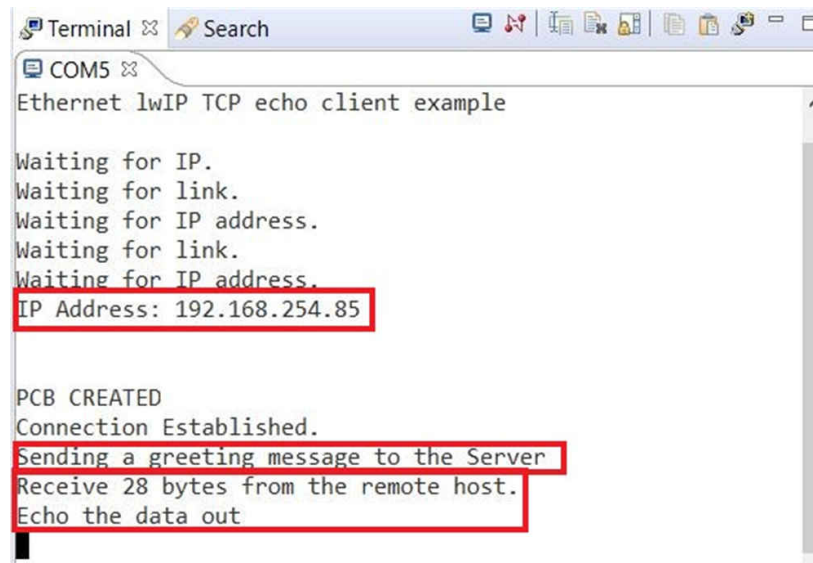


图 10-2. Enet_tcpecho_client_lwip 输出

10.3 enet_tcpecho_client_lwip 示例的 Wireshark 捕获

图 10-3 所示为 Wireshark 捕获。

1. Wireshark 提供各种过滤功能。在本例中，使用 tcp 端口 8000 (这是为 SocketTest 服务器配置的端口号) 过滤事务。
2. 如前所述，TCP 是基于连接的协议。在这里，您会看到客户端 192.168.254.85 发送到服务器 192.168.254.75 以建立连接的 SYN 段，以及来自服务器的用来接受连接的 ACK 段。
3. 连接被接受后，客户端发送消息 “Hello World!\n\r”。请参阅捕获顶部和底部的框 3。请注意，消息的长度为 16 个字节，与消息中的总字符数相匹配。
4. 服务器发送消息 “This is a tcp echo example\n\r”，总长度 28 个字节。
5. 客户端回显消息 “Client: This is a tcp echo example\n\r”，总长度 36 个字节。

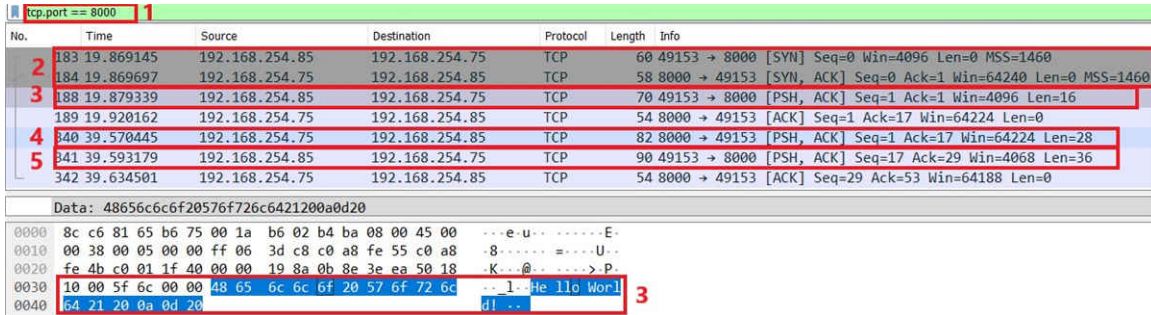


图 10-3. Enet_tcpecho_client_lwip 的客户端到服务器 Wireshark 捕获

11 Enet_adcSENSOR_client_lwip 示例概述

enet_adcSENSOR_client_lwip 示例类似于 enet_tcpecho_client_lwip 客户端应用示例。

在本例中，目的是使用 ADC 模块测量片上温度，并定期将测量数据发送到服务器。

11.1 运行 adcSENSOR_client_lwip 示例

运行该示例将在 “Terminal” 窗口上显示测得的片上温度，并将测得的温度发送到服务器。

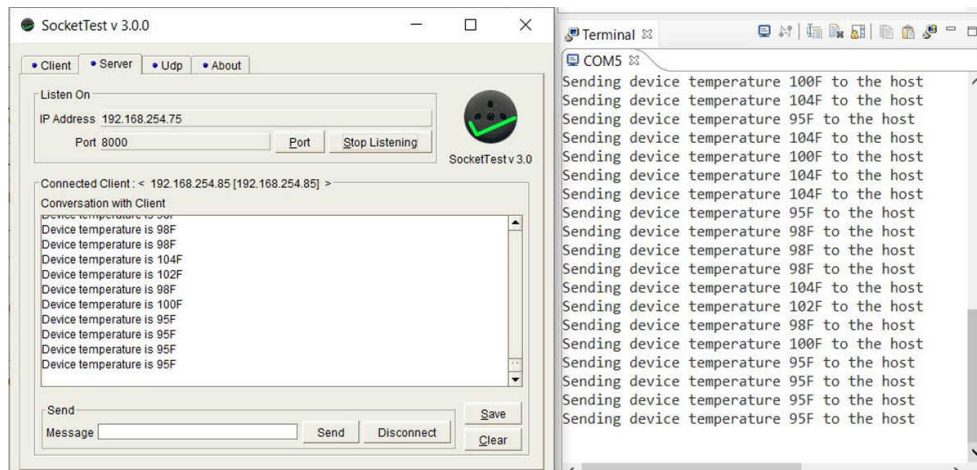


图 11-1. Enet_adcSENSOR_client_lwip 的客户端到服务器 Wireshark 捕获

12 Enet_udpecho_client_lwip 示例概述

enet_udpecho_client_lwip 示例类似于使用 UDP 协议的客户端应用。就 API 而言，它与图 1-2 的 UDP 流程图中描述的服务器示例类似。不同之处在于，客户端需要调用 `udp_connect()` 以首先将 PCB 与远程地址相关联。本示例将使用 `udp_send()` 将问候消息 “Hello! Greeting from EK-TM4C1294XL LaunchPad\n\r” 发送到远程服务器。然后，它将回显从服务器接收到的任何内容。

12.1 运行 enet_udpecho_client_lwip 示例

有关如何设置 SocketTest 以进行 UDP 测试的更多信息，请参阅节 7.1。图 12-1 中的 SocketTest 显示服务器发送和接收的两条消息。两条消息的长度在 “Terminal” 窗口中显示。您可以交叉参考 Wireshark 在图 12-2 中捕获的消息。首先，发送问候消息。然后，客户端发送两条回显消息及其接收的长度。

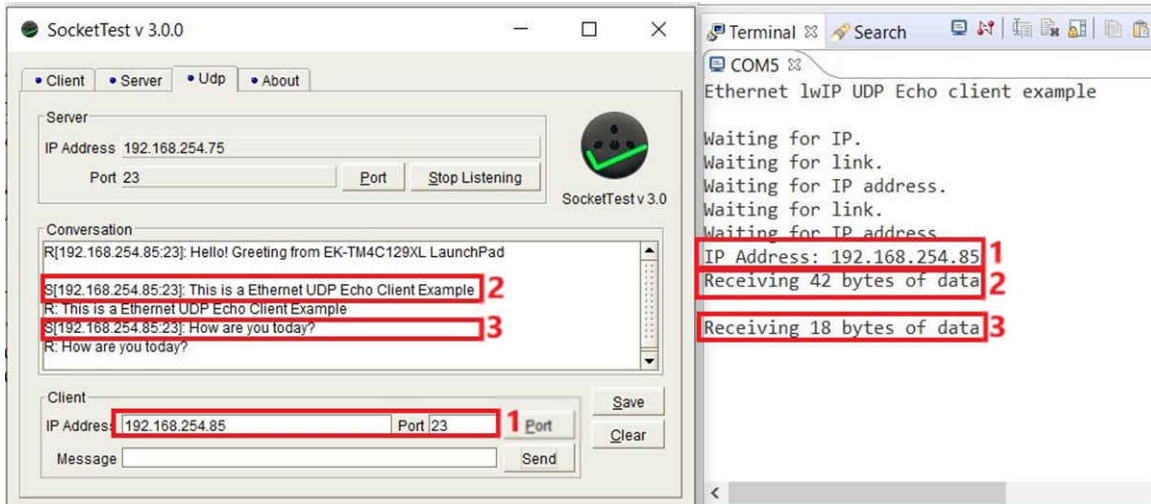


图 12-1. Enet_udpecho_client_lwip 输出

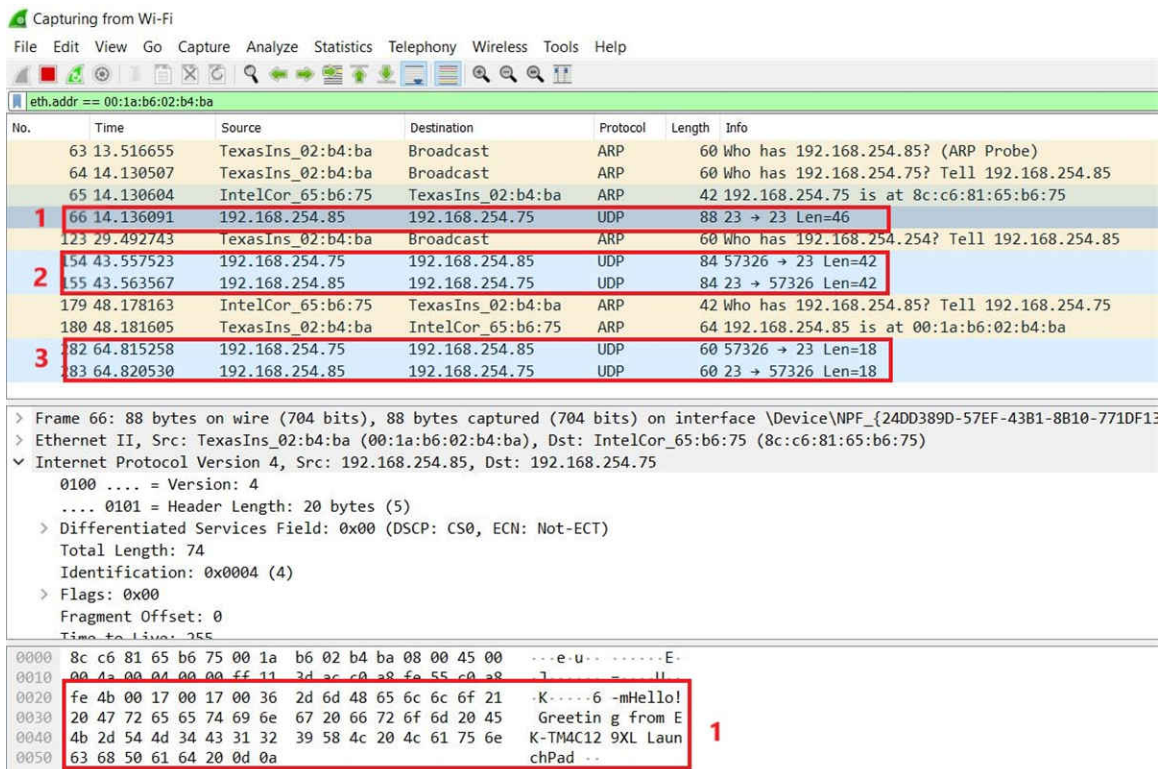


图 12-2. Enet_udpecho_client_lwip 的客户端到服务器 Wireshark 捕获

13 参考文献

- 德州仪器 (TI) : [Tiva C 系列 TM4C1294 互联 LaunchPad 评估套件用户指南](#)
- 德州仪器 (TI) : [TM4C1294NCPDT Snowflake 数据表](#)
- 德州仪器 (TI) : [TivaWare™ 外设驱动程序库用户指南](#)

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2022，德州仪器 (TI) 公司