AVR32817: Getting Started with the 32-bit AVR UC3 Software Framework lwIP TCP/IP Stack

32-bit AVR®
Microcontrollers

Application Note

Features

- **TCP/IP suite description**
- **32-bit AVR® UC3 lwIP port source file architecture**
- **Web, TFTP, SMTP servers:**
  - **Network configuration**
  - **DHCP protocol**

1 Introduction

This document describes the software modules of the Atmel® 32-bit AVR UC3 Software Framework dedicated to the lwIP TCP/IP stack and illustrates how to get started with the lwIP TCP/IP software in the Software Framework.

This document is written for the software developers in order to ease the development of lwIP TCP/IP applications for the 32-bit AVR UC3 series. It assumes that readers are familiar with the 32-bit AVR UC3 series.

2 Overview

Lightweight TCP/IP stack designed for embedded systems. The focus of the lwIP TCP/IP implementation is to reduce resource usage while still having a full scale TCP (cf. http://lwip.wikia.com/wiki/LwIP_Wiki).

lwIP features:

• IP (Internet Protocol) including packet forwarding over multiple network interfaces

•ICMP (Internet Control Message Protocol) for network maintenance and debugging

• UDP (User Datagram Protocol) including experimental UDP-lite extensions

• TCP (Transmission Control Protocol) with congestion control, RTT estimation and fast recovery/fast retransmit

• Specialized raw API for enhanced performance

• Optional Berkeley-alike socket API

• DHCP (Dynamic Host Configuration Protocol)

• PPP (Point-to-Point Protocol)

• ARP (Address Resolution Protocol) for Ethernet

Note: lwIP has been ported on 32-bit AVR UC3 with a MACB controller. The Software Framework 1.7 uses version 1.3.2 which includes support of DHCP.

Rev. 32147A - AVR32-06/10

3 TCP/IP suite

The TCP/IP protocol suite allows systems of all sizes, running different operating systems, to communicate with each other. It forms the basis for what is called the worldwide Internet, a Wide Area Network (WAN) of several million computers.

3.1 TCP/IP Suite Layers

The TCP/IP protocol suite is a combination of different protocols at various layers. TCP/IP is normally considered to be a 4-layer system as shown in Figure 3-1.

Figure 3-1. Four Layers of TCP/IP Protocol Suite

Application	Terminal Emulation Protocol of TCP/IP (Telnet), File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP)
Transport	Transmission Control Protocol (TCP), User Datagram Protocol (UDP)
Network	Internet Protocol (IP), Internet Control Message Protocol (ICMP)
Link	Device Driver and Interface Card

3.1.1 Application Layer

The Application layer handles the details of a particular application. Common TCP/IP

Applications include:

• Telnet for remote login

• Browser support for displaying web pages

• File transfer applications

• E-mail applications

Refer to standard network services as well as communication methods used by various application programs.

3.1.2 Transport Layer

TCP is responsible for a reliable flow of data between two hosts. Typically, TCP divides data passed to it from the application into appropriately sized chunks for the network layer below, acknowledging received packets that are sent and retransmits lost packets. Since this reliable, flow of data is provided by the Transport Layer, the Application Layer above can ignore these details.

UDP is a much simpler service to the Application Layer. It sends packets of data called datagrams from one host to the other, but with no guarantee that the datagrams reach the other end. Desired reliability must be added by the Application Layer.

3.1.3 Network Layer

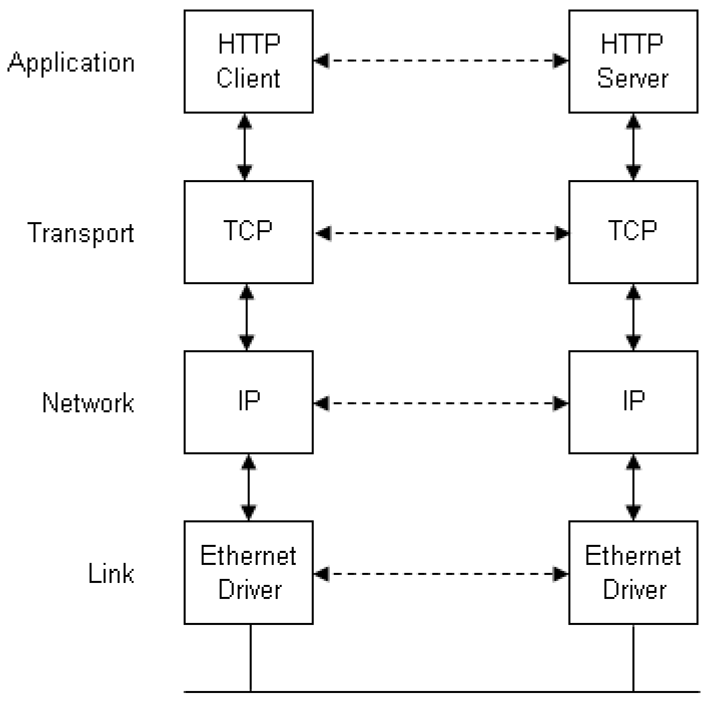
This layer is sometimes called the Internet Layer. It handles the movements of packets around the network. Routing of packets, for example, takes place here. IP (Internet Protocol) and ICMP (Internet Control Message Protocol) provides the Network Layer in the TCP/IP Protocol Suite.

3.1.4 Link Layer

Data-link or Network Interface Layer is another common name of this layer. The Link Layer normally includes the device driver in the operating system and the corresponding network interface (card) in the computer. Together they handle all the hardware details of physically interfacing with the cable.

Figure 3-2 shows an example that includes two hosts on a Local Area Network (LAN) such as Ethernet, using HTTP.

Figure 3-2. Example with Protocols Involved



One side represents the client, and the other the server. The server provides some services to clients, in this case, access to web pages on the server host. Each layer has one or more protocols for communicating with its peer at the same layer. One protocol, for example, allows the two TCP layers to communicate, and another protocol lets the two IP layers communicate.

The Application Layer is normally a user-process while the lower three layers are usually implemented an operating system.

3.1.5 Port Numbers

Different applications can use TCP or UDP at any time. The Transport layer protocols store an identifier in the headers they generate to identify the application. TCP and UDP store the 16-bit source port number and the 16-bit destination port number in those respective headers.

Servers have standard ports. Every TCP/IP implementation with a FTP server provides that service on TCP port 21. Every Telnet server is on TCP port 23. Services provided by any implementation of TCP/IP have well-known port numbers between 1 and 1023. The well-known ports are managed by the Internet Assigned Numbers Authority (IANA).

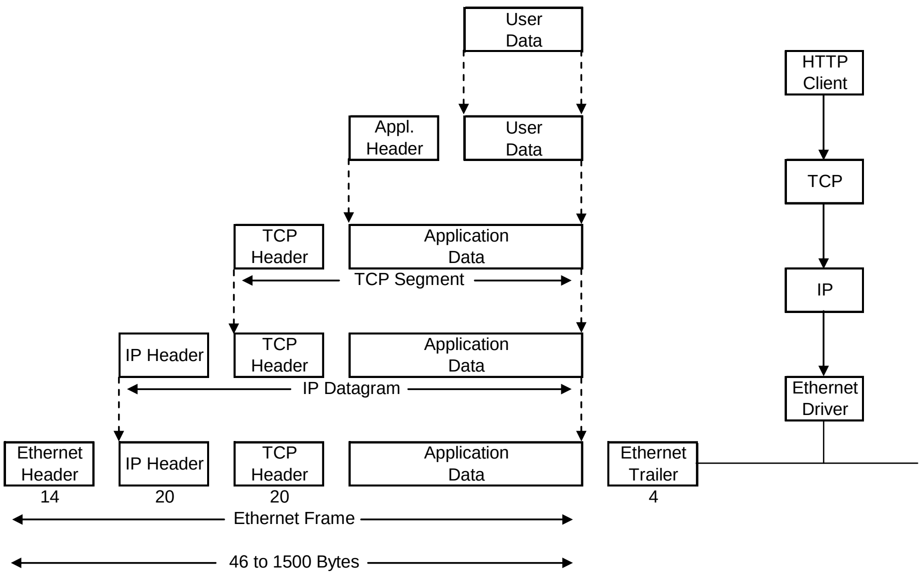
A client usually does not care what port number it uses on its end. All it needs to be certain of is that whatever port number it uses, it must be unique on its host. Client port numbers are called ephemeral ports (i.e., short lived). This is because a client typically exists only as long as the user running the client needs its service, while servers typically run as long as the host is up. Most TCP/IP implementations allocate ephemeral port numbers between 1024 and 5000. The port numbers above 5000 are intended for other services (those that are not well known across the Internet).

The combination of an IP address and a port number is called a socket.

3.1.6 Encapsulation

When an application sends data using TCP, the data is sent down the protocol stack, through each layer, until it is sent as a stream of bits across the network. Each layer adds information to the data by pre-pending headers and adding trailers to the data it receives. Figure 3-3 shows this process.

Figure 3-3. Encapsulation of Data as It Goes Down the Protocol Stack



Some abbreviations:

- TCP segment: The unit of data that TCP sends to IP.
- IP datagram: The unit of data that IP sends to the network interface.
- Frame: The stream of bits that flows across the Ethernet.

IP (Internet Protocol) adds an identifier to the IP header it generates to indicate which layer the data belongs to. IP handles this by storing an 8-bit value in its header called the protocol field. Similarly, many different applications can be using TCP or UDP at any time. The Transport Layer protocol stores an identifier in the header they generate to identify the application. Both TCP and UDP use 16-bit port numbers to identify applications. The TCP and UDP store the source port number and the destination port number in their respective headers. The network interface sends and receives frames on behalf of IP, ARP, RARP. There must be some form of identification in the Ethernet header indicating which network layer protocol generates the data. To handle this, there is a 16-bit frame type field in the Ethernet header.

4 Source Architecture

4.1 Low level Drivers

The UC3 MACB controller implements a 10/100 Ethernet MAC compatible with the IEEE 802.3 standard using an address checker, statistics and control registers, receive and transmit blocks, and a DMA interface.

The address checker recognizes four specific 48-bit addresses and contains a 64-bit hash register for matching multicast and unicast addresses. It can recognize the broadcast address of every node on the network, copy all frames, and act on an external address match signal.

The MACB software driver provides an API to get access to the main features of the MACB controller.

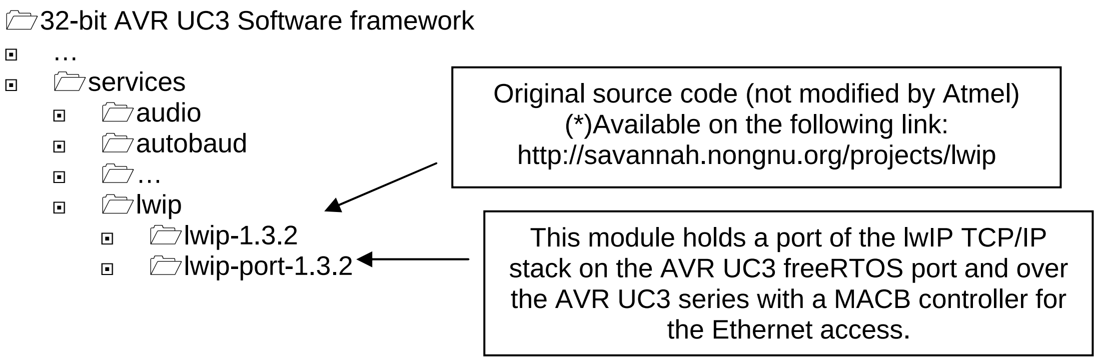
Figure 4-1. MACB low level drivers

- 32-bit AVR UC3 Software framework
  - applications
  - boards
  - components
  - drivers
    - abdac
    - acifb
    - …
    - macb
      - example
      - macb.c
      - macb.h

4.2 lwIP source and port files

lwIP is freely available under a BSD-style license in C source code format and can be downloaded from the development homepage*.

Figure 4-2. lwIP source and port files

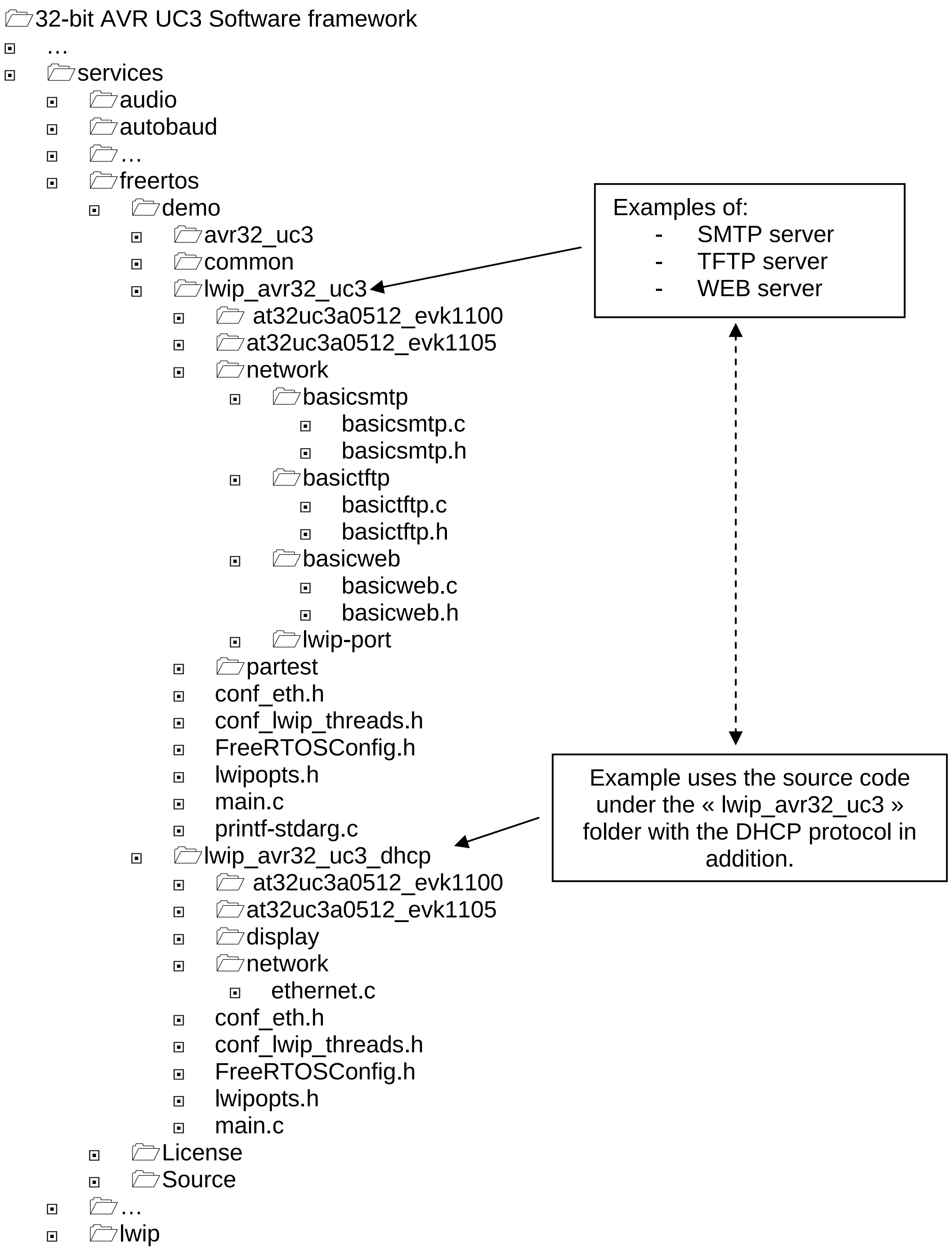

32-bit AVR UC3 Software framework
…
services
audio
autobaud
…
lwip
lwip-1.3.2
lwip-port-1.3.2
Original source code (not modified by Atmel) (*)Available on the following link: http://savannah.nongnu.org/projects/lwip
This module holds a port of the lwIP TCP/IP stack on the AVR UC3 freeRTOS port and over the AVR UC3 series with a MACB controller for the Ethernet access.

Note: In respect to the lwIP original source code, only the last version is supported by the development team. The Atmel port is supported by the standard Atmel support channels (please refer to the section 10).

4.3 32-bit AVR UC3 software framework lwIP examples

The lwIP examples implementation uses the 32-bit AVR UC3 FreeRTOS.org kernel port. FreeRTOS.org is a portable, open source, mini Real Time Kernel - a free to download and royalty free RTOS that can be used in commercial applications. FreeRTOS is licensed under a modified GPL.

Figure 4-3: 32-bit AVR UC3 software framework lwIP examples layout



```
32-bit AVR UC3 Software framework
  …
  services
    audio
    autobaud
    …
    freertos
      demo
        avr32_uc3
        common
        lwip_avr32_uc3
          at32uc3a0512_evk1100
          at32uc3a0512_evk1105
          network
            basicsmtp
              basicsmtp.c
              basicsmtp.h
            basictftp
              basictftp.c
              basictftp.h
            basicweb
              basicweb.c
              basicweb.h
            lwip-port
          partest
          conf_eth.h
          conf_lwip_threads.h
          FreeRTOSConfig.h
          lwipopts.h
          main.c
          printf-stdarg.c
        lwip_avr32_uc3_dhcp
          at32uc3a0512_evk1100
          at32uc3a0512_evk1105
          display
          network
            ethernet.c
          conf_eth.h
          conf_lwip_threads.h
          FreeRTOSConfig.h
          lwipopts.h
          main.c
      License
      Source
    …
    lwip
```

Examples of:
- SMTP server
- TFTP server
- WEB server
Example uses the source code under the « lwip_avr32_uc3 » folder with the DHCP protocol in addition.

5 lwIP_avr32_uc3 example description

5.1 What it does

5.1.1 SMTP server

Implement a simple SMTP server and allow sending an e-mail, without DHCP.

5.1.2 TFTP server

Implement a simplistic TFTP server and allow transferring files up to 2048 Bytes between client and server.

5.1.3 WEB server

Implement a simplistic WEB server.

5.2 How to configure

This application can be customized by changing a few definitions such as the IP address, MAC address...

Note: All definitions are not explained in this section, only those pertinent to this application note.

5.2.1 System

5.2.1.1 src/services/freertos/Demo/lwIP_AVR32_UC3/FreeRTOSConfig.h

This file contains configuration defines for FreeRTOS. There are no lwIP client related defines in this file.

5.2.1.2 src/services/freertos/Demo/lwIP_AVR32_UC3/conf_eth.h

This header file sets all external configurations of the Ethernet module such as the Mac address, server and client IP address.

PHY:

PHY is a common abbreviation for the physical layer of the OSI model. A PHY connects a link layer device (often called a MAC) to a physical medium.

To enable the Reduced Media Independent Interface (RMII), The ETHERNET_CONF_USE_RMII_INTERFACE need to be set to 1 in con_eth.h file. In case of this define is set to 0, MII interface is consequently enabled.

MAC:

Table 5-1. Mac address: this address must be unique, given values are in hexadecimal.

# Define	Value
ETHERNET_CONF_ETHADDR0	0x00
ETHERNET_CONF_ETHADDR1	0x04
ETHERNET_CONF_ETHADDR2	0x25
ETHERNET_CONF_ETHADDR3	0x40
ETHERNET_CONF_ETHADDR4	0x40
ETHERNET_CONF_ETHADDR5	0x40

This configuration will set MAC address to: **00:04:25:40:40:40**

IP:

Table 5-2. Board IP address, required if DHCP is not enabled (see lwipopts.h).

# Define	Value
ETHERNET_CONF_IPADDR0	192
ETHERNET_CONF_IPADDR1	168
ETHERNET_CONF_IPADDR2	0
ETHERNET_CONF_IPADDR4	2

This configuration will set client IP address to: **192.168.0.2**

5.2.1.3 src/services/freertos/Demo/lwIP_AVR32_UC3_DHCP/lwipopts.h

Table 5-3. LwIP 1.3.2 configuration.

# Define	Value
LWIP_DHCP	0: (disable, default value) the client file uses the fixed IP address set in conf_eth.h

5.2.2 Project definition

Definitions used to enable or disable applications for GCC and IAR^{TM} compiler are located in the following folders of the Software Framework:

- src/services/freertos/Demo/lwIP_AVR32_UC3/AT32UC3A0512_EVK110x

- src/services/freertos/Demo/Demo/lwIP_AVR32_UC3_DHCP/AT32UC3A0512_EVK110x

5.2.2.1 SMTP server

Table 5-4. SMTP definition.

Definition	**Value**
SMTP_USED	1: Build the application with a simplistic SMTP. 0: (default) Build the application without SMTP.

5.2.2.2 TFTP server

Table 5-5. TFTP definition.

Definition	**Value**
TFTP_USED	1: (default) Build the application with the TFTP server. 0: Build the application without TFTP server.

5.2.2.3 WEB server

Table 5-6. HTTP definition.

Definition	**Value**
HTTP_USED	1: (default) Build the application with the web server task. Connecting to this web server, returns stacks memory status. This is updated every second on a web browser. 0: Build the application without web server.

5.2.3 Hardware and Network connections

5.2.3.1 SMTP server

If there is an Ethernet network available, the toolkit can be connected to any Ethernet connection or hub belonging to the network. If the PC is connected to a network, there is a strong possibility that the default IP address of the Toolkit is outside the range of the network (the address doesn't belong to the IP subset of the network). If the Ethernet network is connected to the Internet, this is certain.

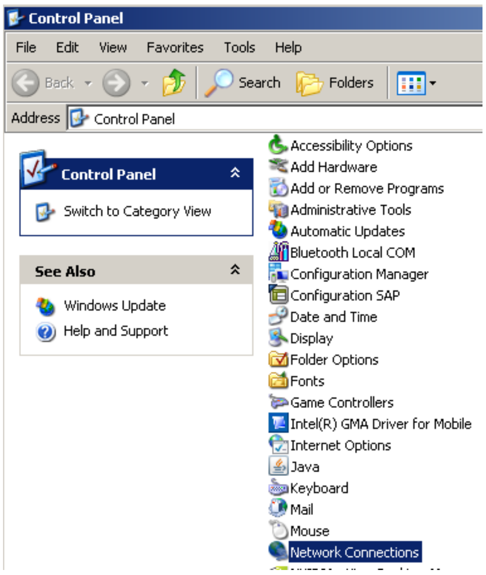
In this case and if DHCP is disabled, a new IP address for the Toolkit is required. Contact the local network administrator to be assigned a free IP address for the Toolkit, then See the section 5.2.1.2 in order to modify the IP configuration value.

5.2.3.2 TFTP server

The Ethernet interface connector is of standard RJ-45 type. Plug the Ethernet cable directly into the Toolkit to connect to the PC.

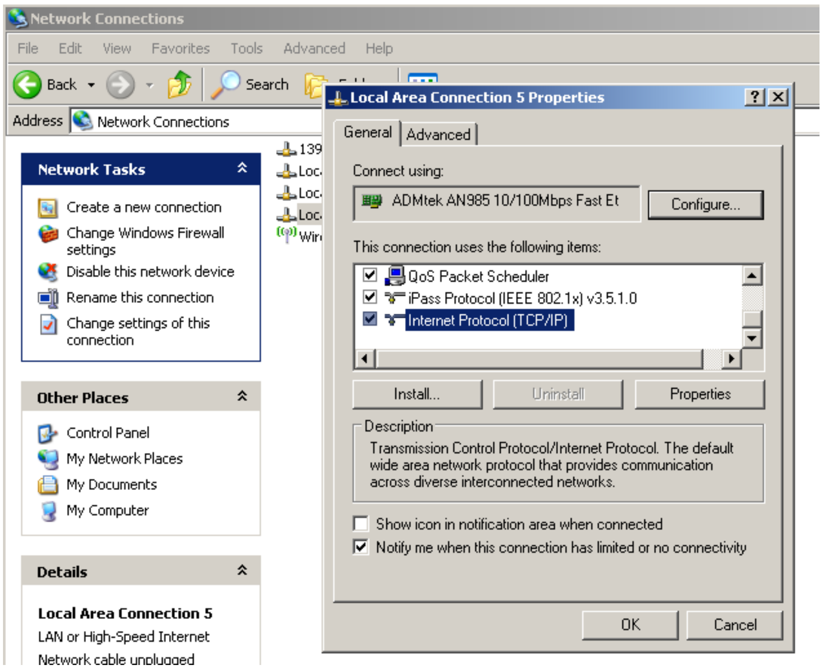
Open the Microsoft® Windows® control panel, and select network setting.

Figure 5-7. Windows control Panel



Right click on "Local Area Connection" and select the properties menu.

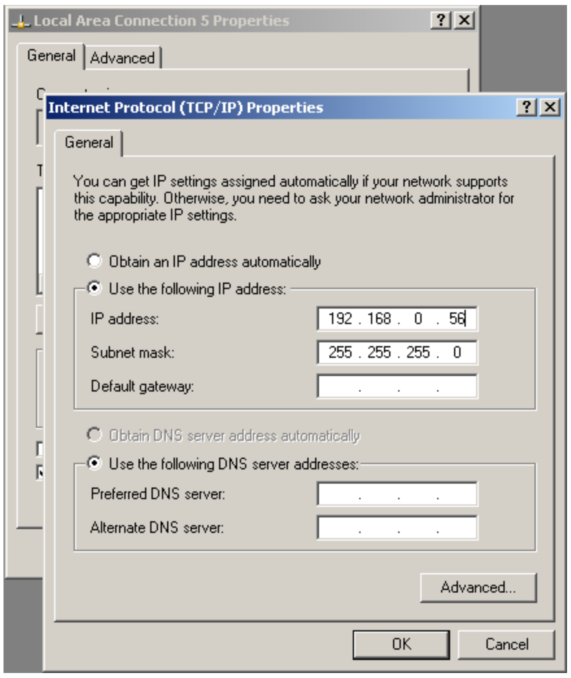
Figure 5-8. Properties menu


Network Connections
File Edit View Favorites Tools Advanced Help
Back Search
Address Network Connections
Network Tasks
Create a new connection
Change Windows Firewall settings
Disable this network device
Rename this connection
Change settings of this connection
Other Places
Control Panel
My Network Places
My Documents
My Computer
Details
Local Area Connection 5
LAN or High-Speed Internet
Network cable unplugged
Local Area Connection 5 Properties
General Advanced
Connect using:
ADMtek AN985 10/100Mbps Fast Et
Configure...
This connection uses the following items:
QoS Packet Scheduler
iPass Protocol (IEEE 802.1x) v3.5.1.0
Internet Protocol (TCP/IP)
Install... Uninstall Properties
Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.
Show icon in notification area when connected
Notify me when this connection has limited or no connectivity
OK Cancel

Select "Internet Protocol (TCP/IP)" and press "Properties".

There is a new dialog box, which must be filled.

Figure 5-9. as defined in the following screen shot.


Local Area Connection 5 Properties

Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

( ) Obtain an IP address automatically

(●) Use the following IP address:

IP address: 192 . 168 . 0 . 56

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

( ) Obtain DNS server address automatically

(●) Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

Once this is configured, press OK. Now configuration is ready.

The default server IP address can be changed (see conf_eth.h).

5.2.3.3 WEB server

Same as TFTP server.

5.3 How to run it

5.3.1 SMTP server

Before starting the application, some configurations must be applied to the src/services/freertos/Demo/lwIP_AVR32_UC3/network/BasicSMTP.c source file:

- SMTP server address : Contact the local network administrator to get this address.

- Mail sender : used in the mail from field, default is sender@domain.com.

- Mail recipient : used in the rcpt to field, default is receiver@domain.com.

- Mail content : default is Subject: *** SPAM ***\r\nFROM: \"Your Name here\" \r\nTO: \"Your Contact here\" \r\n\r\nSay what you want here.

Once all fields are configured, remove the #error lines in the BasicSMTP.c source file to allow compilation.

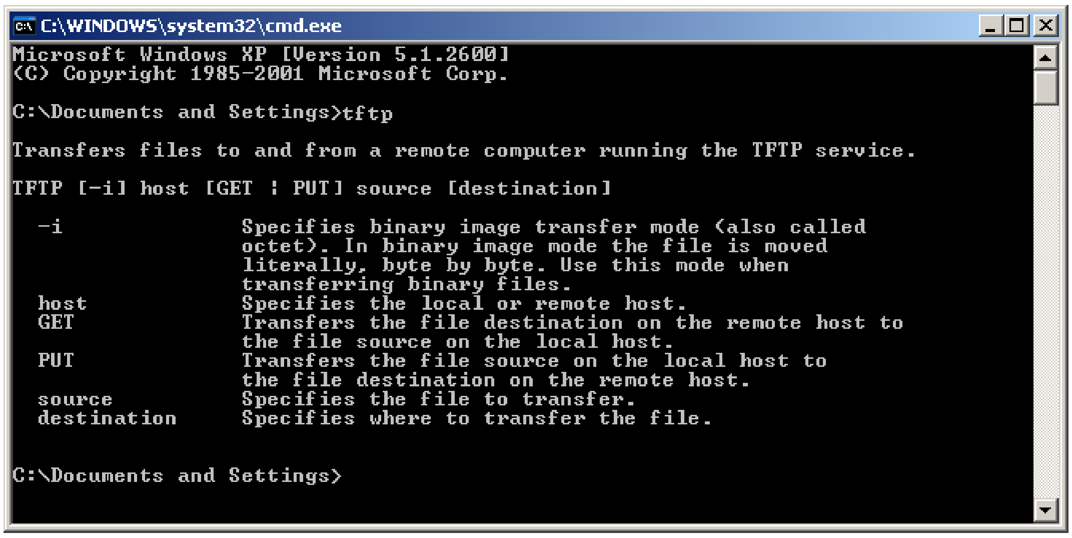
Run the software and press Push Button 0 (on the EK1100) to send an email.

Note: The EVK1105 Evaluation Kit has been developed around QTouch® features, therefore no push button has been implemented. Nevertheless, some GPIOs are free and can be used in replacement of the push button.

5.3.2 TFTP server: src\services\freertos\Demo\lwIP_AVR32_UC3\network\BasicTFTP\BasicTFTP.c

To view help at the command-line, at the command prompt, type the following:

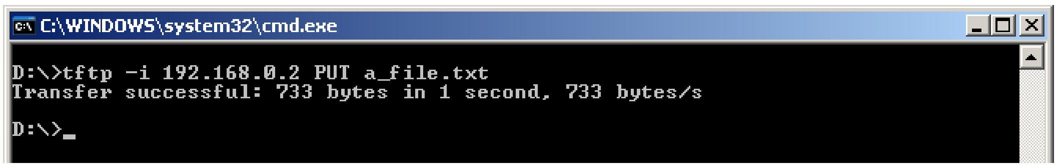
Figure 5-10. TFTP commands help



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings>tftp

Transfers files to and from a remote computer running the TFTP service.

TFTP [-i] host [GET | PUT] source [destination]

  -i              Specifies binary image transfer mode (also called
                  octet). In binary image mode the file is moved
                  literally, byte by byte. Use this mode when
                  transferring binary files.
  host            Specifies the local or remote host.
  GET             Transfers the file destination on the remote host to
                  the file source on the local host.
  PUT             Transfers the file source on the local host to
                  the file destination on the remote host.
  source          Specifies the file to transfer.
  destination     Specifies where to transfer the file.


C:\Documents and Settings>
```

To put a file onto the TFTP server (Supported file size < 2048 bytes), on a PC command line, type put "a_file": this will copy a_file from your hard drive to a RAM buffer of the demo (see figure 6-7).

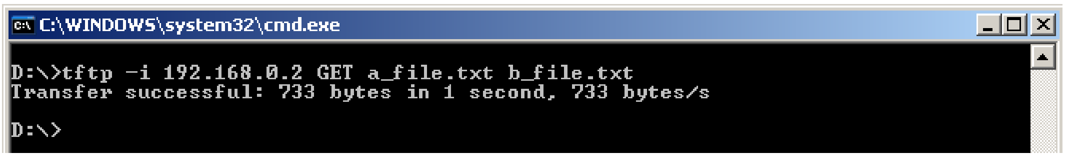
Figure 5-11. PUT command line example



```
C:\WINDOWS\system32\cmd.exe

D:\>tftp -i 192.168.0.2 PUT a_file.txt
Transfer successful: 733 bytes in 1 second, 733 bytes/s

D:\>_
```

To get a file from the TFTP server, on a PC command line, type get "a_file": this will copy a_file from the RAM buffer of the application to the PC's hard drive (see figure 6-8).

Figure 5-12. GET command file example



```
C:\WINDOWS\system32\cmd.exe

D:\>tftp -i 192.168.0.2 GET a_file.txt b_file.txt
Transfer successful: 733 bytes in 1 second, 733 bytes/s

D:\>
```

Note 1: only one file at a time is supported on this TFTP server. This is because the TFTP server being a simplistic example, it does not use a file system to store files but a predefined RAM area of 2048 Bytes.

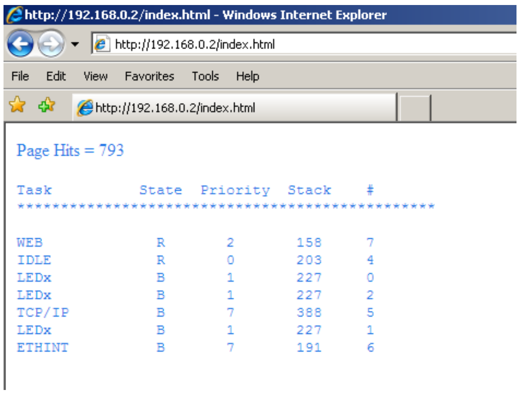
Note 2: The LAN security configuration may halt the file transfer.

5.3.3 WEB server: src\services\freertos\Demo\lwIP_AVR32_UC3\network\BasicWEB\Basicweb.c

Launch your favorite web browser. Type the WEB server example IP address in your browser's address bar.

Every time a connection is made and data is received, a dynamic page that shows the current FreeRTOS.org kernel statistics is generated and returned (see figure 5-13). The connection is then closed.

Figure 5-13. Basic Web server


http://192.168.0.2/index.html - Windows Internet Explorer
http://192.168.0.2/index.html
File
Edit
View
Favorites
Tools
Help
http://192.168.0.2/index.html

```
Page Hits = 793

Task          State  Priority  Stack    #
************************************************

WEB             R       2       158     7
IDLE            R       0       203     4
LEDx            B       1       227     0
LEDx            B       1       227     2
TCP/IP          B       7       388     5
LEDx            B       1       227     1
ETHINT          B       7       191     6
```

6 lwIP_avr32_uc3_dhcp example description

6.1 What it does

Same as the simplistic SMTP, TFTP and WEB server from lwIP_avr32_uc3 example (src\services\freertos\Demo\lwIP_AVR32_UC3\network\...) with DHCP enabled.

6.2 How to configure

6.2.1 System

6.2.1.1 src/services/freertos/ Demo/lwIP_AVR32_UC3_dhcp/FreeRTOSConfig.h

Same as in lwIP_avr32_uc3 example, section 5.2.1.1.

6.2.1.2 src/services/freertos/Demo/lwIP_AVR32_UC3_dhcp/conf_eth.h

PHY and MAC:

Same as in lwIP_avr32_uc3 example, section 5.2.1.2.

IP:

As the DHCP is enabled, no IP define from this file is required. The network will automatically provide an available IP address.

6.2.2 Project definition

Same as in lwIP_avr32_uc3 example, section 5.2.2.

6.2.3 Hardware and Network connections

6.2.3.1 SMTP server

Plug one side of the Ethernet cable to the toolkit connector named “eth” or “Ethernet” and the other side on an Ethernet switch, hub or router (not directly to a PC).

6.2.3.2 TFTP server

Same as SMTP server.

6.2.3.3 WEB server

Same as SMTP server.

6.3 How to run it

6.3.1 SMTP server: src/services/freertos/Demo/lwIP_AVR32_UC3/network/BasicSMTP.c

Same as the SMTP server from lwIP_avr32_uc3 example, section 5.3.1.

6.3.2 TFTP server: src\services\freertos\Demo\lwIP_AVR32_UC3\network\BasicTFTP\BasicTFTP.c

Start up the application and get the IP address from the toolkit display in order to use it for the transfer command-line.

6.3.3 WEB server: src\services\freertos\Demo\lwIP_AVR32_UC3\network\BasicWEB\Basicweb.c

Start up the application and get the IP address from the toolkit display in order to use it in your browser's address bar.

7 Other avr32 uc3 examples using the lwIP stack

7.1 EVK1100 Control Panel example

The EVK1100 Control Panel application is a demonstration application. Its purpose is to log onboard sensors and actuators data and events (data acquisition through ADC channels) and make these available through the various connectivity channels supported by the AT32UC3A microcontroller series.

The logs are accessible locally through USART or USB (Mass Storage class), and/or remotely through the Internet (Web server).

EVK1110 Getting Started Application related to the Control Panel example can be found here:

http://www.atmel.com/dyn/resources/prod_documents/EVK1100_Getting_Started.pdf

7.2 SSL server

Based on PolarSSL light-weight open source cryptographic, this example allow to the application to exchange messages with a server over a TCP/IP connectivity through a secure socket layer connection.

Application note can be found here:

http://www.atmel.com/dyn/resources/prod_documents/doc32111.pdf

Source Code “AVR32753 .zip” can be found on the Atmel 32-bit AVR UC3 Application note web page:

http://www.atmel.com/dyn/products/app_notes_v2.asp?family_id=607

8 Frequently Asked Questions

Q: I need an example of a WEB server for the AVR32 UC3A0/1 but not using freeRTOS.

A: The Web servers implemented in the Software Framework (under applications/EVK1100-control-panel/ folder or under services/freertos/Demo/lwIP_AVR32_UC3/ folder) are using the lwIP TCP/IP stack and FreeRTOS.

You can find an implementation of the lwIP TCP/IP stack without RTOS in the WiFi stack available under components/wifi/hd/ folder of the Software Framework package.

There is also some words about this subject here:

http://lwip.wikia.com/wiki/LwIP_with_or_without_an_operating_system

9 Reference

lwIP source code development home page can be found here:

http://savannah.nongnu.org/projects/lwip

lwIP wikia link:

http://lwip.wikia.com/wiki/LwIP_Wiki

lwIP from Wikipedia:

http://en.wikipedia.org/wiki/LwIP

lwIp optimization:

http://lwip.wikia.com/wiki/Maximizing_throughput

FreeRTOS homepage:

http://www.freertos.org/

10 Support

Atmel has several support channels available:

• Web portal:	http://support.atmel.no/	All Atmel microcontrollers
• Email:	avr@atmel.com	All AVR products
• Email:	avr32@atmel.com	All 32-bit AVR products

Please register on the web portal to gain access to the following services:

- Access to a rich FAQ database
- Easy submission of technical support requests
- History of all your past support requests
- Register to receive Atmel microcontrollers' newsletters

Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
avr32@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Request
www.atmel.com/literature