Computer Systems Engineering Technology
CST 417 – Embedded Networking

Lab 2 – Debug Environment & Ping
Fall 2014
Instructor: Troy Scevers
Possible Points: 10

Name_____
Due Date:10/21/2014 @ start of lab

Instructions

In this lab we will be modifying the web server demo application from lab1. We will cut out the sections of the TCP/IP stack we do not need and enable the ping demo. We will then modify the main program to check for button pushes and perform a predefined task for each button.

Procedure

Open main_demo.h and find the following section and make it look as below.

```
// enable the demo-applications that you want to run
//#define APP_USE_GENERIC_TCP_CLIENT_DEMO
//#define APP_USE_GENERIC_TCP_SERVER_DEMO
//#define APP_USE_GENERIC_SSL_CLIENT_DEMO
//#define APP_USE_SMTP_CLIENT_DEMO
#define APP_USE_PING_DEMO
//#define APP_USE_SNMP_TRAP_DEMO
//#define APP_USE_SNMP_V2_TRAP_DEMO
//#define APP_USE_BERKELEY_API_DEMO
```

Next open tcpip_config.h and make it look like below.

```
// TCPIP Stack Module Selection
//   Uncomment or comment the following lines to enable or
//   disabled the following high-level application modules.

#define TCPIP_STACK_USE_IPV4                // enable IPv4 functionality
#define TCPIP_STACK_USE_ICMP_SERVER         // Ping query and response capability
//#define TCPIP_STACK_USE_HTTP2_SERVER        // New HTTP server with POST,
Cookies, Authentication, etc.
//#define TCPIP_STACK_USE_SSL_SERVER        // SSL server socket support
//#define TCPIP_STACK_USE_SSL_CLIENT        // SSL client socket support
#define TCPIP_STACK_USE_DHCP_CLIENT         // Dynamic Host Configuration Protocol
client for obtaining IP address and other parameters
//#define TCPIP_STACK_USE_SMTP_CLIENT       // Simple Mail Transfer Protocol for
sending email
//#define TCPIP_STACK_USE_TELNET_SERVER     // Telnet server
//#define TCPIP_STACK_USE_ANNOUNCE          // Microchip Embedded Ethernet Device
Discoverer server/client
```

```
#define TCPIP_STACK_USE_DNS                 // Domain Name Service Client for
resolving hostname strings to IP addresses
#define TCPIP_STACK_USE_NBNS              // NetBIOS Name Service Server for
repsonding to NBNS hostname broadcast queries
//#define TCPIP_STACK_USE_REBOOT_SERVER     // Module for resetting this PIC
remotely.  Primarily useful for a Bootloader.
#define TCPIP_STACK_USE_SNTP_CLIENT         // Simple Network Time Protocol for
obtaining current date/time from Internet
//#define TCPIP_STACK_USE_DYNAMICDNS_CLIENT // Dynamic DNS client updater module
//#define TCPIP_STACK_USE_BERKELEY_API      // Berekely Sockets APIs are available
//#define TCPIP_STACK_USE_IPV6              // enable IPv6 functionality
#define TCPIP_STACK_USE_TCP                 // Enable the TCP module
#define TCPIP_STACK_USE_UDP                 // Enable the UDP module
//#define TCPIP_STACK_USE_ZEROCONF_LINK_LOCAL        // Zeroconf IPv4 Link-Local
Addressing;
//#define TCPIP_STACK_USE_ZEROCONF_MDNS_SD           // Zeroconf mDNS and mDNS
service discovery
#define TCPIP_STACK_COMMAND_ENABLE        // TCPIP_COMMANDS for network
configuration or debug
//#define TCPIP_STACK_USE_IPERF             // Enable the iperf module for standard
network benchmarking
//#define TCPIP_STACK_USE_SNMP_SERVER       // Simple Network Management Protocol
v2C Community Agent
//#define TCPIP_STACK_USE_SNMPV3_SERVER     // SNMP v3 agent
//#define TCPIP_STACK_USE_FTP_SERVER        // File Transfer Protocol
//#define TCPIP_STACK_USE_DHCP_SERVER         // DHCP server
#define TCPIP_STACK_USE_ICMP_CLIENT         // Ping transmission capability
```

Also find the following section and comment it out as well....

```
// ============================================================================
//   Event Notifications Options
// ============================================================================
//#define TCPIP_STACK_USE_EVENT_NOTIFICATION

    // The default interrupt priority to use for the TCPIP interrupts
//   #define   TCPIP_EVENT_IPL         5
//   #define   TCPIP_EVENT_SIPL      1
```

Next replace main_demo.c and ping_demo.c with the versions found in lab2.zip. Next in the TODO section of main_demo.c place code to poll the buttons. If button 1 is pressed then call the function `Ping4("www.google.com");`. If button 2 is pressed then call the function `Ping4("www.cset.oit.edu");`. If button three is pressed then use the function DBGETS to get input from the user and then call the Ping4(). An Example of DBGETS is given below.

```
char id_input[100];

DBPRINTF("Please Enter Ping Destination address: ");
DBGETS(id_input, 100);
```

Play around with the DBPRINTF and DBGETS functions to improve the program and get used to how they work. Demo final project to lab instructor. Turn in zip copy of your project.