
Quick Start Guide

1.0 INTRODUCTION

The ENC624J600 Quick Start Guide provides a visual view of step-by-step procedures used when establishing a TCP/IP connection between the Fast Ethernet PICtail board and the Explorer 8 development board. Detailed information is included in the following sections:

- [Section 2.0, General Considerations](#)
- [Section 3.0, Creating a Project Using MCC and TCP/IP Lite Stack](#)
- [Section 4.0, Sending and Receiving UDP Packets](#)
- [Section 5.0, TCP Client Implementation](#)
- [Section 6.0, TCP Server Implementation](#)
- [Section 7.0, Conclusion](#)

2.0 GENERAL CONSIDERATIONS

2.1 Required References

The ENC624J600 implementor should have the following documents on hand:

- *ENC424J600/624J600 Stand-Alone 10/100 Ethernet Controller with SPI or Parallel Interface Data Sheet*
- *ENC424J600/624J600 ENC424J600/624J600 Silicon Errata and Data Sheet Clarification*
- *Fast 100 Mbps Ethernet PICtail Plus Daughter Board User's Guide*
- *Explorer 8 Development Board User's Guide*
- ENC Board Schematics

2.2 Hardware Requirements and Setup

All the hardware devices mentioned in this quick start guide can be acquired using their part numbers from www.microchipdirect.com. Below is a list of devices:

- Explorer 8 Development Board with power supply (DM160228)
- Fast 100 Mbps Ethernet PICtail Plus Daughter Board (AC164132)
- PIC18F87K22 PIM (MA180028)
- Ethernet cables (RJ45)
- Ethernet switch with a power supply

2.3 Software Requirements

- MPLAB X IDE
- XC8 Compiler
- MPLAB Code Configurator (MCC)
- [Wireshark Packet Analyzer](#)
- Microchip TCP/IP Lite Stack ([AN1921 Source Code](#))

2.4 Jumper Settings

2.4.1 FOR THE EXPLORER 8 DEVELOPMENT BOARD:

- J14 – Connect to 5V power setting.
- J2 – Move to supply 5V power to the board.
- J7 – Connect to provide LED current path.
- J37 – Do not connect a jumper to any of the pins.
- J45 – Connect the microcontroller SDO pin to the J32 mikroBUS MOSI.
- J46 – Connect the microcontroller SDI pin to the J32 mikroBUS MISO.
- J47 – Connect the microcontroller SCK pin to the J32 mikroBUS SCK.
- J31 – Configure RA4 as an I/O port.
- J4 – Configure RA5 as an I/O port.

2.4.2 FOR THE PICTAIL PLUS DAUGHTER BOARD:

- JP2 – Shorted by jumper shunt
- JP9 – Jumper between PSPCFG2 and GND
- JP10 – Jumper between 3V3 and PSPCFG3
- JP11 – Jumper between PSPCFG1&4 and GND

3.0 CREATING A PROJECT USING MCC AND TCP/IP LITE STACK

This chapter demonstrates the generation of a basic project with the TCP/IP Lite library using MCC.

3.1 General Procedure

Follow the steps below to start a basic project:

1. Open the MPLAB X IDE and create a new project.
2. Launch the MCC plug-in using the MCC icon in the top right corner in the MPLAB X IDE. Alternatively, MCC can be launched by selecting *Tools>Embedded>MPLAB Code Configurator v3: Open/ Close*.

3.2 System Module Configuration

1. In the **Project Resources** tab, select **System Module**. Go to Easy Setup view and change the “Oscillator Select” to Internal RC oscillator, CLKOUT function on OSC2. (See [Figure 3-1](#).)

FIGURE 3-1: SYSTEM MODULE

System Module

Easy Setup Registers

▼ INTERNAL OSCILLATOR

Current System clock 8 MHz

Oscillator Select Internal RC oscillator, CLKOUT function on OSC2

System Clock Select FOSC

Internal Clock 8MHz_HF →PLL Capable Frequency

External Clock 1 MHz

☐ PLL Enabled ☐ Software PLL Enabled

▼ WDT

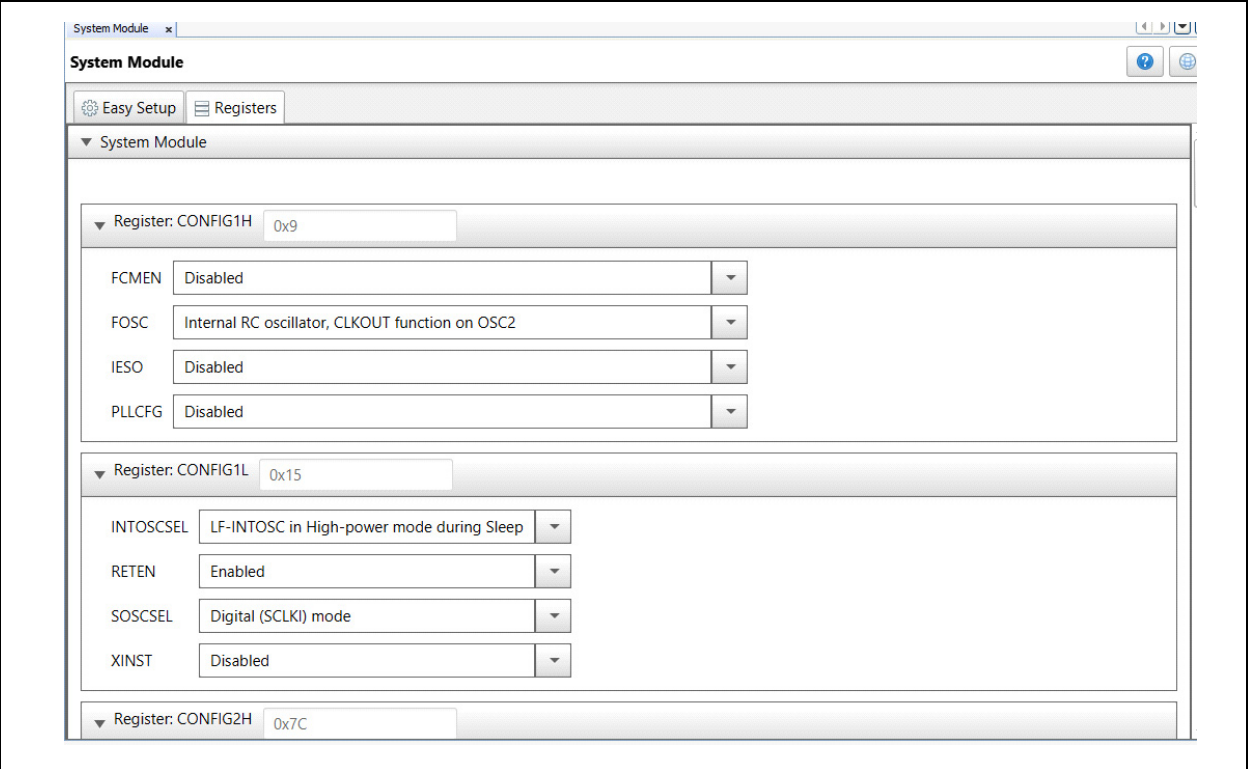
Watchdog Timer Period 33.82503 s

Watchdog Timer Enable WDT disabled in hardware; SWDTEN bit disabled

Watchdog Timer Postscaler 1:1048576

2. Navigate to the **Registers** tab of the System Module in the MCC window. In the CONFIG1L register, disable the “XINST” bit to disable the extended instruction set. (See [Figure 3-2](#).)

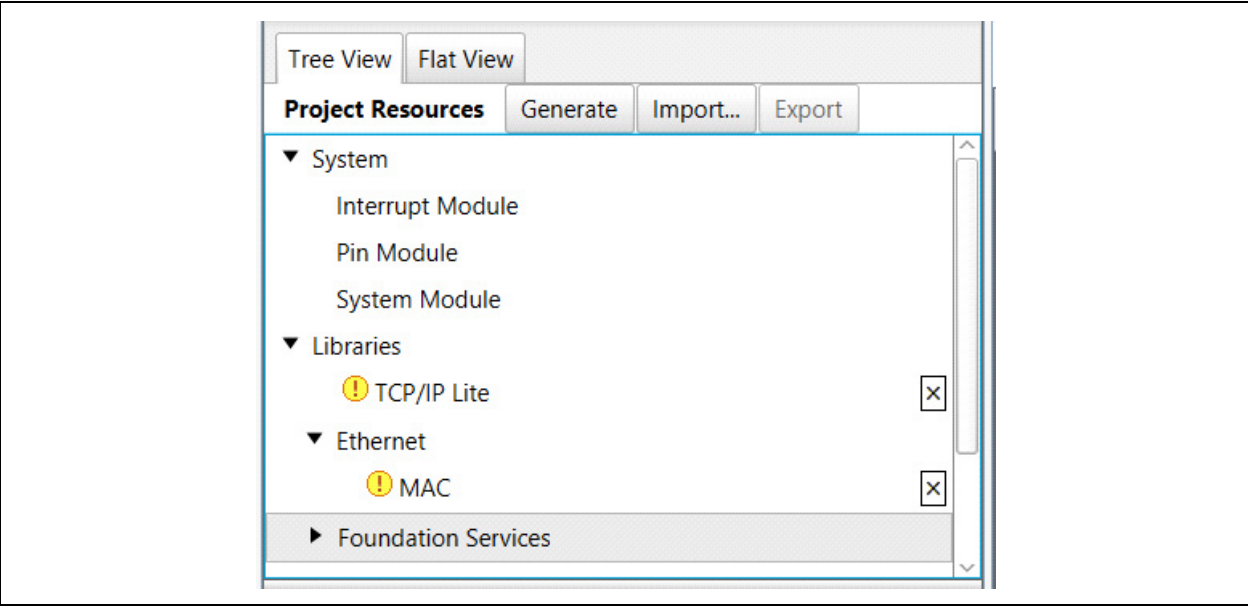
FIGURE 3-2: SYSTEM CONFIGURATION REGISTER



3.3 Ethernet Configuration

- 1. Expand the Ethernet Library listed in the **Device Resources** tab. Double-click on the TCP/IP Lite library to add it to the project. Similarly, add the MAC in the Ethernet library to the project. (See [Figure 3-3](#).)

FIGURE 3-3: PROJECT RESOURCES TCIP MAC



- 2. In the **Project Resources** tab, select **TCP/IP Lite** module. This will display all the protocols supported by the TCP/IP Stack. "UDP," "DHCP," "TCP," and "ICMP" protocols must be checked. "IPV4" and "ARP" are always required and are checked by default. (See [Figure 3-4](#).)

FIGURE 3-4: TCP/IP LITE SETUP

TCP/IP Lite

Easy Setup

Microchip Lightweight TCPIP Stack

☒ UDP
 ☒ DHCP
 ☒ IPV4
 ☒ TCP
 ☒ ICMP
 ☒ ARP
 ☐ TFTP

☐ NTP
 ☐ DNS
 ☐ LLDP

► TCP

▼ DHCP

DHCP Name: PIC18F87K22 Ethernet

► UDP

▼ ICMP

☒ Tx Echo Response
 ☒ Port unreachable

▼ IPV4

Nothing to Configure

▼ ARP

ARP Table Size: 8

- In the **Project Resources** tab, select **MAC** module. This gives the list of available MACs. Select the ENCx24J600 and MSSP1. (See [Figure 3-5](#).)

FIGURE 3-5: MAC SETUP

MAC

Easy Setup

Media Access Controller

ENCx24J600

▼ SPI Configuration

Name	SPI Mode	SPI Data In...	S
MAC	M...	END	S

< >

+ -

3.4 SPI Configuration

The Master Synchronous Serial Port module (MSSP1) needs to be configured as SPI Master for accessing the ENC624J600. Configure the MSSP1 **Easy Setup** as follows:

1. Set “Mode” to SPI Master.
2. Set “Input Data Sampled” to End.
3. Set “Clock Edge” to Active to Idle.
4. Set “Clock Source” to Fosc/4.

See [Figure 3-6](#).

FIGURE 3-6: SPI MASTER

The screenshot shows the 'SPI MASTER' configuration window with the 'Easy Setup' tab selected. Under 'Hardware Settings', there is a table with the following data:

Name	SPI Mode	SPI Data Input ...	Speed (...)	Actual Speed...	SPI
MAC	MODE0	END	8000	1,000	MSSP1

3.5 Timer Configurations

Perform the following steps to set timer configurations:

1. Go to **Peripherals** listed in the **Device Resources** tab. Expand Timer peripheral and double-click on **TMR1** to load the module.
2. In the **Project Resources** tab, select **TMR1** module. Configure the system time to 1s by doing the following:
 - a) Update the “Clock Source” to FOSC/4.
 - b) Update the “Timer Period” to 25 ms.
 - c) Tick “Enable Timer Interrupt.”
 - d) Update the “Callback Function Rate” to 40 times.

This results the timer configuration to interrupt for every 1 second. (See [Figure 3-7](#).)

FIGURE 3-7: TMR1

The screenshot shows the 'TMR1' configuration window with the 'Easy Setup' tab selected. The 'Hardware Settings' section includes:

- ☒ Enable Timer
- Timer Clock:**
 - Clock Source: FOSC/4
 - External Frequency: 32.768 kHz
 - Prescaler: 1:1
 - ☒ Enable Synchronization
- Timer Period:**
 - Timer Period: 500 ns ≤ 25 ms ≤ 32.768 ms
 - Period count: 0x0 ≤ 0x3CB0 ≤ 0xFFFF
 - Calculated Period: 25 ms
- ☐ Enable Gate
 - ☐ Enable Gate Toggle Gate Signal Source: T1G_pin
 - ☐ Enable Gate Single-Pulse mode Gate Polarity: low
- ☒ Enable Timer Interrupt
- ☐ Enable Timer Gate Interrupt

The 'Software Settings' section includes:

- Callback Function Rate: 40 x Time Period = 1 s

3.6 Pin Configuration

Follow these steps to configure the pins:

1. Go to the **Pin Manager: Grid [MCC]** tab. Click on the pins stated in [Table 3-1](#) as illustrated in [Figure 3-8](#) for their configuration.

TABLE 3-1: PIN FUNCTION TABLE

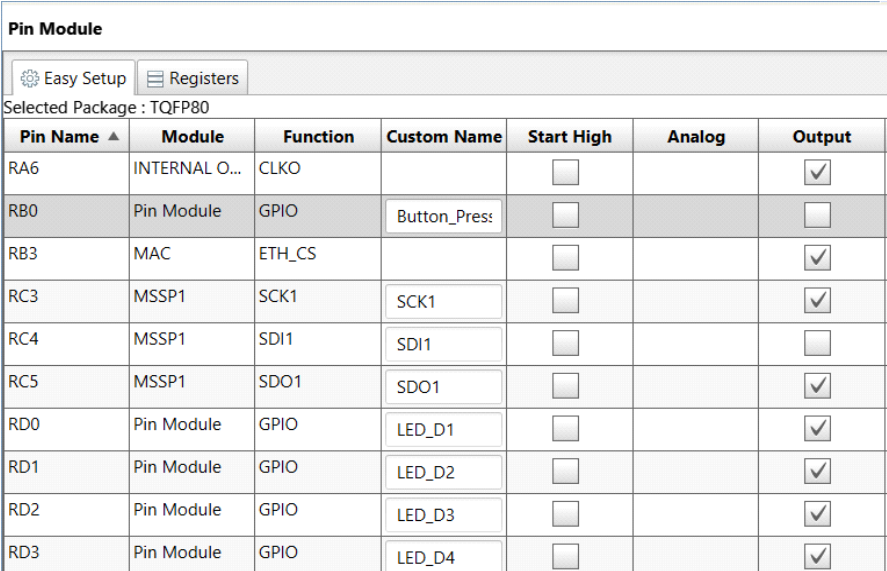
Module in Pin Manager	Pin	Custom Name	Function
MAC	RB3	Not Configurable	Chip select pin for the MAC
MSSP1	RC3	SCK1	Serial Clock
MSSP1	RC4	SDI1	Serial Data In
MSSP1	RC5	SDO1	Serial Data Out
Pin Module	RB0	Button_Press	Push button S1 on the Explorer 8 Board (Input)
Pin Module	RD0	LED_D1	LED D1 on the Explorer 8 board (output)
Pin Module	RD1	LED_D2	LED D2 on the Explorer 8 board (output)
Pin Module	RD2	LED_D3	LED D3 on the Explorer 8 board (output)
Pin Module	RD3	LED_D4	LED D4 on the Explorer 8 board (output)

FIGURE 3-8: PIN MANAGER GRID

Search Results			Action Items		Output		Notifications [MCC]				Pin Manager: Grid View ×																								
Package:		TQFP80	▼		Pin No:		30 29 28 27 34 33 50 49				58 57 56 55 54 53 52 47				36 35 43 44 45 46 37 38				72 69 68 67 66 65 64 63																
			Port A ▼							Port B ▼							Port C ▼							Port D ▼											
Module	Function	Direction	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	
MAC	ETH_CS	output																																	
MSSP1 ▼	SCK1	output																																	
	SDI1	input																																	
	SDO1	output																																	
OSC ▼	CLKO	output																																	
	REFO	output																																	
Pin Module ▼	GPIO	input																																	
	GPIO	output																																	
RESET	MCLR	input																																	
TMR1 ▼	T1CKI	input																																	
	T1G	input																																	

2. Set the custom names for the pins as shown in the **MPLAB Code Configurator** tab. (See [Figure 3-9](#).)

FIGURE 3-9: PIN MODULE CUSTOM NAME



Pin Module

Easy Setup Registers

Selected Package : TQFP80

Pin Name ▲	Module	Function	Custom Name	Start High	Analog	Output
RA6	INTERNAL O...	CLKO		☐		☒
RB0	Pin Module	GPIO	Button_Pres:	☐		☐
RB3	MAC	ETH_CS		☐		☒
RC3	MSSP1	SCK1	SCK1	☐		☒
RC4	MSSP1	SDI1	SDI1	☐		☐
RC5	MSSP1	SDO1	SDO1	☐		☒
RD0	Pin Module	GPIO	LED_D1	☐		☒
RD1	Pin Module	GPIO	LED_D2	☐		☒
RD2	Pin Module	GPIO	LED_D3	☐		☒
RD3	Pin Module	GPIO	LED_D4	☐		☒

- Click on **Generate** to generate the code for the configured modules.

Note: Click on **Generate** even if the MPLAB X shows a warning. These warnings are handled later by modifying the code.

3.7 Programming and Wireshark

- In the **Projects** tab, under sources file, double-click on “main.c”. In the file, enable global and peripheral interrupts by removing the comment declaration “//”. (See [Figure 3-10.](#))

FIGURE 3-10: GLOBAL AND PERIPHERAL INTERRUPTS

```
// Enable the Global Interrupts
INTERRUPT_GlobalInterruptEnable();

// Disable the Global Interrupts
//INTERRUPT_GlobalInterruptDisable();

// Enable the Peripheral Interrupts
INTERRUPT_PeripheralInterruptEnable();
```

- In “main.c”, add **Network_Manage()** to the **while(1)** loop and the corresponding **network.h** header file. This is required to handle the packets in the network. (See [Figure 3-11.](#))

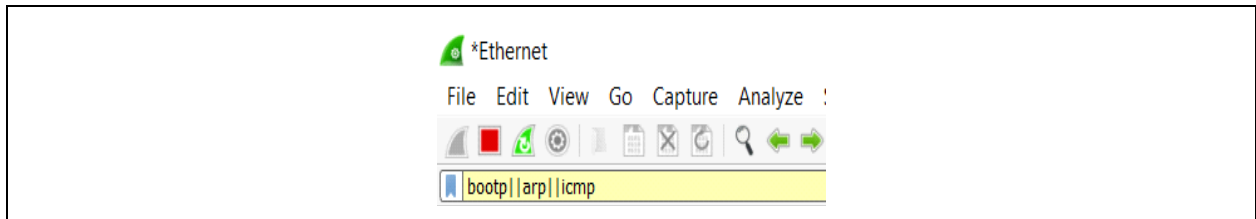
FIGURE 3-11: ADDING NETWORK MANAGE

```

while (1)
{
    // Add your application code
    Network_Manage();
}

```

3. Launch Wireshark. From the Capture menu, click on **Options**. Select an Interface from the list to which your board and PC are connected. Click on **Start** for capturing the packet. (e.g. Ethernet)
4. In Wireshark, set the filter field as "Bootp || arp || icmp." (See [Figure 3-12.](#))

FIGURE 3-12: FILTER SETTINGS

5. In MPLAB X IDE, click on **Clean and Build** project. Select the PICKit Programmer. After a successful build, click on **Run** project to program the code to the device.
6. In Wireshark, notice the DHCP packets handshake to get the device IP address and MAC address. (See [Figure 3-13.](#))

FIGURE 3-13: DHCP OFFER

No.	Time	Source	Destination	Protocol	Length	Info
5	1.922339	Dell_76:1c:7e	Cisco_28:51:0f	ARP	42	Who has 192.168.0.1? Tell 192.168.0.2
6	1.923544	Cisco_28:51:0f	Dell_76:1c:7e	ARP	60	192.168.0.1 is at 2c:0b:e9:28:51:0f
49	57.041412	192.168.0.2	192.168.0.1	DHCP	368	DHCP Request - Transaction ID 0xfb95e34d
50	57.042848	192.168.0.1	192.168.0.2	DHCP	342	DHCP ACK - Transaction ID 0xfb95e34d
75	61.922105	Dell_76:1c:7e	Cisco_28:51:0f	ARP	42	Who has 192.168.0.1? Tell 192.168.0.2
76	61.923112	Cisco_28:51:0f	Dell_76:1c:7e	ARP	60	192.168.0.1 is at 2c:0b:e9:28:51:0f
92	78.422463	0.0.0.0	255.255.255.255	DHCP	333	DHCP Discover - Transaction ID 0x766c6314
93	78.423505	Cisco_28:51:0f	Broadcast	ARP	60	Who has 192.168.0.3? Tell 192.168.0.1
95	80.423666	Cisco_28:51:0f	Broadcast	ARP	60	Who has 192.168.0.3? Tell 192.168.0.1
97	82.423431	Cisco_28:51:0f	Broadcast	ARP	60	Who has 192.168.0.3? Tell 192.168.0.1
99	84.423674	Cisco_28:51:0f	Broadcast	ARP	60	Who has 192.168.0.3? Tell 192.168.0.1
102	85.423701	192.168.0.1	255.255.255.255	DHCP	342	DHCP Offer - Transaction ID 0x766c6314
103	85.449827	0.0.0.0	255.255.255.255	DHCP	345	DHCP Request - Transaction ID 0x766c6314
104	85.451197	192.168.0.1	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0x766c6314
127	117.200203	192.168.0.2	192.168.0.1	DHCP	368	DHCP Request - Transaction ID 0x501b4809

7. In Wireshark, double-click on Offer packet. Expand Bootstrap protocol to get the device IP address. (See [Figure 3-14.](#))

ENC624J600

FIGURE 3-14: OFFER PACKET

```
> Frame 102: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface 0
> Ethernet II, Src: Cisco_28:51:0f (2c:0b:e9:28:51:0f), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 192.168.0.1, Dst: 255.255.255.255
> User Datagram Protocol, Src Port: 67, Dst Port: 68
✓ Dynamic Host Configuration Protocol (Offer)
  Message type: Boot Reply (2)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0x766c6314
  Seconds elapsed: 0
  > Bootp flags: 0x8000, Broadcast flag (Broadcast)
  Client IP address: 0.0.0.0
  Your (client) IP address: 192.168.0.3
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address: Microchi_5b:c6:28 (54:10:ec:5b:c6:28)
  Client hardware address padding: 00000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
```

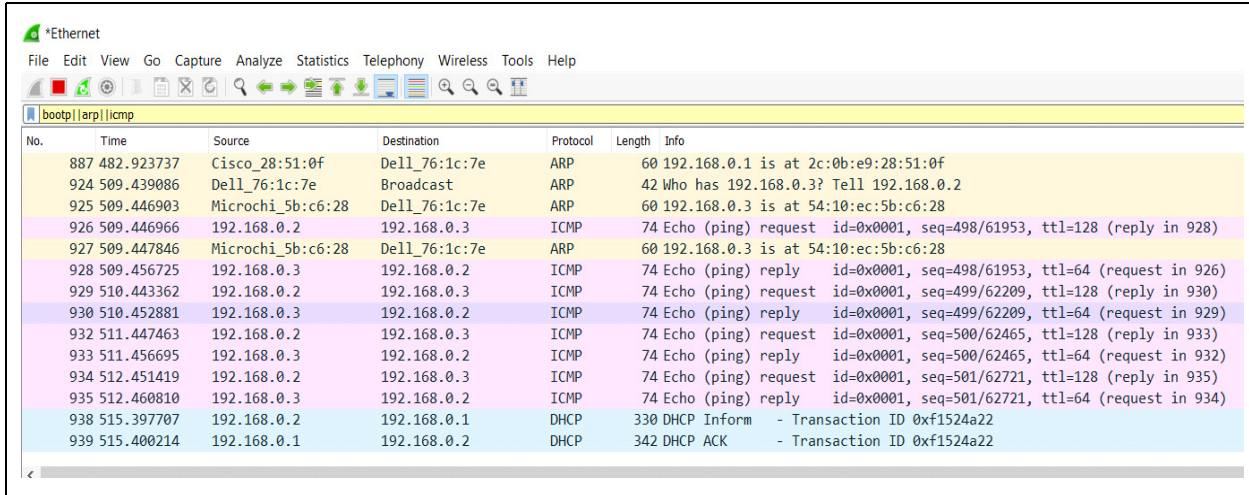
8. On your PC, open the Command Prompt (cmd.exe) window and ping the device using its IP address with the command.

ping IP Address

e.g.: ping 19.168.0.3

9. Check the Wireshark window to see the ICMP & ARP Request/Reply packets. (See [Figure 3-15](#).)

FIGURE 3-15: ICMP & ARP REQUEST/REPLY PACKETS



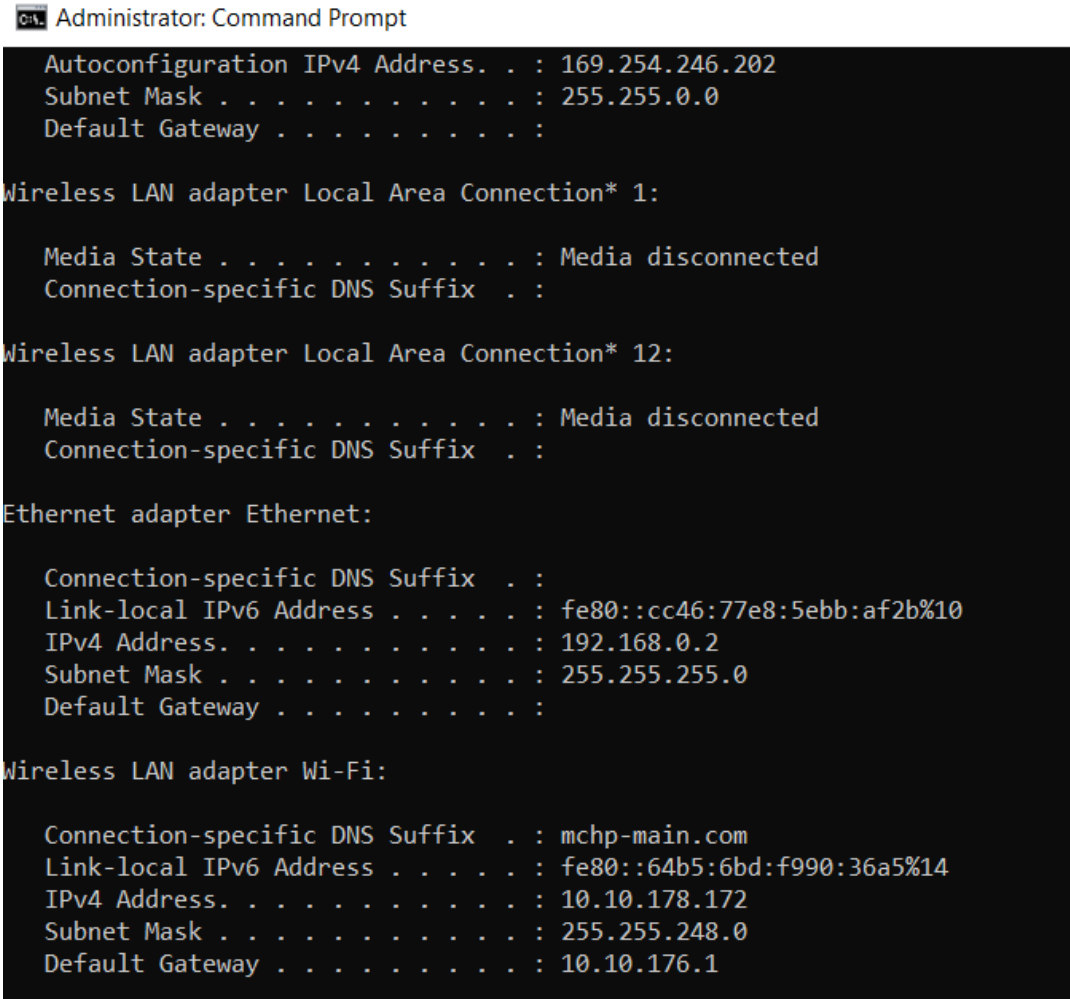
No.	Time	Source	Destination	Protocol	Length	Info
887	482.923737	Cisco_28:51:0f	Dell_76:1c:7e	ARP	60	192.168.0.1 is at 2c:0b:e9:28:51:0f
924	509.439086	Dell_76:1c:7e	Broadcast	ARP	42	Who has 192.168.0.3? Tell 192.168.0.2
925	509.446903	Microchi_5b:c6:28	Dell_76:1c:7e	ARP	60	192.168.0.3 is at 54:10:ec:5b:c6:28
926	509.446966	192.168.0.2	192.168.0.3	ICMP	74	Echo (ping) request id=0x0001, seq=498/61953, ttl=128 (reply in 928)
927	509.447846	Microchi_5b:c6:28	Dell_76:1c:7e	ARP	60	192.168.0.3 is at 54:10:ec:5b:c6:28
928	509.456725	192.168.0.3	192.168.0.2	ICMP	74	Echo (ping) reply id=0x0001, seq=498/61953, ttl=64 (request in 926)
929	510.443362	192.168.0.2	192.168.0.3	ICMP	74	Echo (ping) request id=0x0001, seq=499/62209, ttl=128 (reply in 930)
930	510.452881	192.168.0.3	192.168.0.2	ICMP	74	Echo (ping) reply id=0x0001, seq=499/62209, ttl=64 (request in 929)
932	511.447463	192.168.0.2	192.168.0.3	ICMP	74	Echo (ping) request id=0x0001, seq=500/62465, ttl=128 (reply in 933)
933	511.456695	192.168.0.3	192.168.0.2	ICMP	74	Echo (ping) reply id=0x0001, seq=500/62465, ttl=64 (request in 932)
934	512.451419	192.168.0.2	192.168.0.3	ICMP	74	Echo (ping) request id=0x0001, seq=501/62721, ttl=128 (reply in 935)
935	512.460810	192.168.0.3	192.168.0.2	ICMP	74	Echo (ping) reply id=0x0001, seq=501/62721, ttl=64 (request in 934)
938	515.397707	192.168.0.2	192.168.0.1	DHCP	330	DHCP Inform - Transaction ID 0xf1524a22
939	515.400214	192.168.0.1	192.168.0.2	DHCP	342	DHCP ACK - Transaction ID 0xf1524a22

4.0 SENDING AND RECEIVING UDP PACKETS

This chapter illustrates the following steps for implementing UDP packet communication:

1. Open the MPLAB X IDE and load the `UDP_DEMO.X` file (Refer to TCP/IP Lite Stack.).
2. Open the Windows Command Prompt on your PC and type “`ipconfig`” and press enter to find the IP address of your PC. Se. (See [Figure 4-1.](#))

FIGURE 4-1: IPCONFIG



```

Administrator: Command Prompt

Autoconfiguration IPv4 Address. . : 169.254.246.202
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . :

Wireless LAN adapter Local Area Connection* 1:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 12:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet:

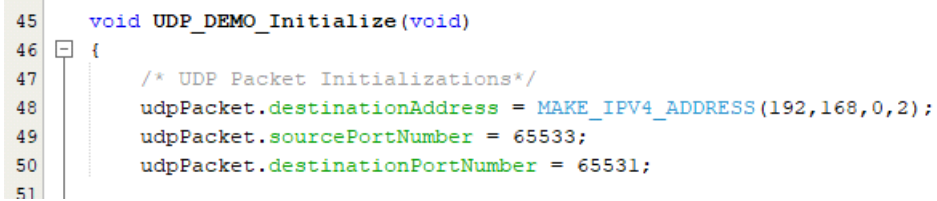
    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::cc46:77e8:5ebb:af2b%10
    IPv4 Address. . . . . : 192.168.0.2
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . : mchp-main.com
    Link-local IPv6 Address . . . . . : fe80::64b5:6bd:f990:36a5%14
    IPv4 Address. . . . . : 10.10.178.172
    Subnet Mask . . . . . : 255.255.248.0
    Default Gateway . . . . . : 10.10.176.1
  
```

3. Open and scroll down in the file `udp_demo.c`. In the `UDP_DEMO_Initialization()` API function, update destination IP address (value found in Step 2) and **Port Number** (custom value within Dynamic/Private range). (See [Figure 4-2.](#))

FIGURE 4-2: INITIALIZING UDP DEMO



```

45 void UDP_DEMO_Initialize(void)
46 {
47     /* UDP Packet Initializations*/
48     udpPacket.destinationAddress = MAKE_IPv4_ADDRESS(192,168,0,2);
49     udpPacket.sourcePortNumber = 65533;
50     udpPacket.destinationPortNumber = 65531;
51 }
  
```

4. In the `main.c` file from the *Project>Source Files*, call `UDP_DEMO_Initialize()`. (See [Figure 4-3](#).)

FIGURE 4-3: CALL UDP INITIALIZE API

```
// Enable the Peripheral Interrupts
INTERRUPT_PeripheralInterruptEnable();

// Disable the Peripheral Interrupts
//INTERRUPT_PeripheralInterruptDisable();
UDP_DEMO_Initialize();

while (1)
{
```

5. Open and scroll down the `main.c` file from *Project>Source Files*. Scroll down to `while(1)` loop, call the function `UDP_DEMO_Send()` to demonstrate the sending UDP packets.
6. Send the UDP packet through the following steps:
- Start the UDP packet.
 - Write the UDP packet.
 - Send the UDP packet.

See [Figure 4-4](#).

FIGURE 4-4: UDP START WRITE SEND

```
/* ***** Step 1 - Start UDP Packet ***** */
* @Param1 - Destination Address
* @Param2 - Source Port Number
* @Param3 - Destination Port Number
*****/
ret = UDP_Start(udpPacket.destinationAddress, udpPacket.sourcePortNumber, udpPacket.destinationPortNumber);

/* ***** Step 2 - Write UDP Packet ***** */
* @Param1 - Data to write
*****/
if(ret == SUCCESS)
{
    UDP_WriteString(text);

    /* ***** Step 3 - Send UDP Packet ***** */
    UDP_Send();
}
```

- Navigate through the *Projects>Source Files>MCC Generated Files>TCPIP Library>udpv4 port handler table.c file*. Add the "UDP_DEMO_Recv" handler to receive UDP packets on desired port, '65531,' in the **UDP_CallbackTable[]**. (See [Figure 4-5](#).)

FIGURE 4-5: UDP PORT NUMBER

```

46  const udp_handler_t UDP_CallbackTable[] = \
47  {
48      {68, DHCP_Handler},
49      {65531, UDP_DEMO_Recv}
50  };

```

In the same file include the UDP demo application header file `udp_demo.h`.

- Open the `udp_demo.c` file. Scroll down to **UDP_DEMO_Recv()** function, add UDP library API to read the receive UDP packet. (See [Figure 4-6](#).)

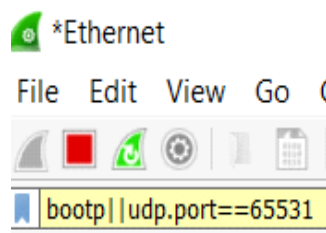
FIGURE 4-6: UDP DEMO RECEIVE

```

17  void UDP_DEMO_Recv(void)
18  {
19      udpDemoRecv_t udpRecv;
20
21      UDP_ReadBlock(&udpRecv, sizeof(udpDemoRecv_t));
22

```

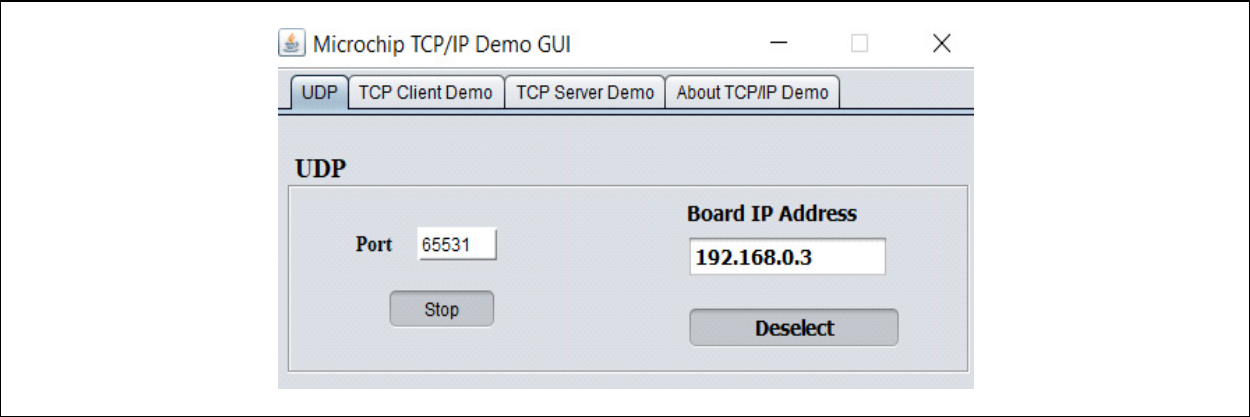
- Launch Wireshark. From the Capture menu, click on **Options**.
- Select an interface from the list to which your board and PC are connected, click on **Start** for capturing the packets. e.g.: Ethernet
- In Wireshark, set the filter field as "bootp||udp.port == 65531." (See [Figure 4-7](#).)

FIGURE 4-7: BOOTP_UDP_PORT_65531

- In MPLAB IDE, from the Production menu, click on **Clean and Build Project**. After successful build, click on **Run** from the tool bar to program the code to the board.
- In Wireshark, notice the DHCP packets handshake to get the device IP address.
- In Wireshark, double-click on Offer packet. Expand Bootstrap protocol to get the device IP address.
- Open the Java application **TCPIP_Demo GUI** (Refer TCP/IP Lite Stack).
- Go to **UDP** tab and assign the same port number as "Port", which was assigned in the UDP library (in step 4). Click on the **Listen** button, and click on **Allow Access** if warning occurs. (See [Figure 4-8](#).)

Note: Step 4 and Step 16 should use the same UDP port number. The board will be sending UDP packets to Java application on that port.

FIGURE 4-8: UDP GUI

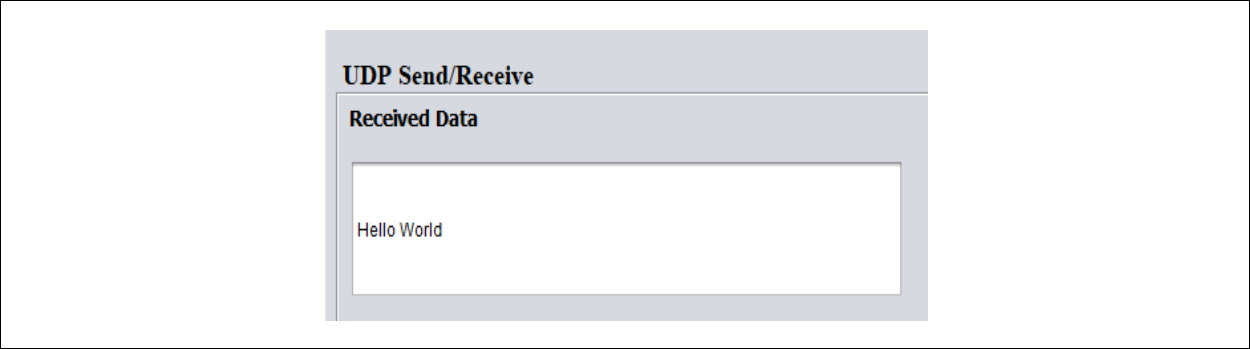


17. Click the push button **S1** on your board to send “Hello World” in the UDP packet. The first click issues an ARP request to get the MAC Address of the PC. In Wireshark, open the filter field “arp,” and click on **Apply** to set the filter.

Note: Step 17 needs to be done more than twice.

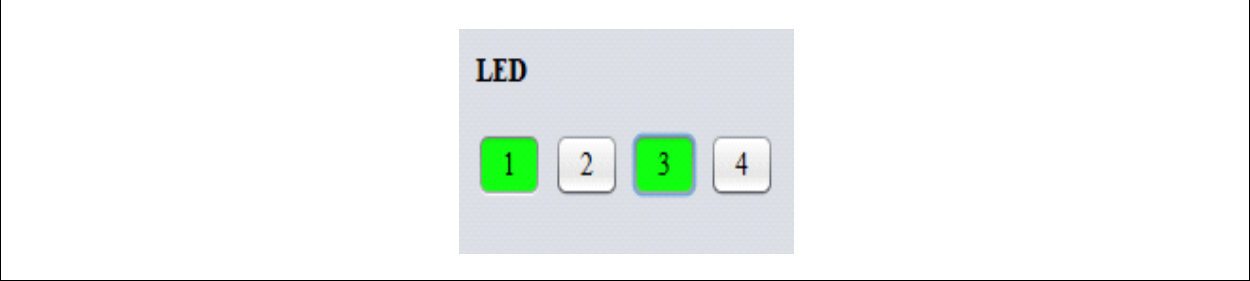
18. The TCP/IP Demo GUI prints the received data. (See [Figure 4-9.](#))

FIGURE 4-9: UDP SEND AND RECEIVE



19. In TCP/IP Demo GUI, click on any **LED** buttons (1, 2, 3, or 4). Observe the corresponding LED's (D1, D2, D3, or D4) toggle ON/OFF on the board. (See [Figure 4-10.](#))

FIGURE 4-10: UDP LED DEMO



20. In Wireshark, observe the UDP packets sent on the port numbers from step 2 to 14. (See [Figure 4-11.](#))

FIGURE 4-11: UDP SENT PORT NUMBER

No.	Time	Source	Destination	Protocol	Length	Info
671	280.956230	192.168.0.2	192.168.0.1	DHCP	368	DHCP Request - Transaction ID 0xefc1c68f
672	280.957399	192.168.0.1	192.168.0.2	DHCP	342	DHCP ACK - Transaction ID 0xefc1c68f
746	340.996158	192.168.0.2	192.168.0.1	DHCP	368	DHCP Request - Transaction ID 0x2a093651
747	340.998142	192.168.0.1	192.168.0.2	DHCP	342	DHCP ACK - Transaction ID 0x2a093651
826	401.032731	192.168.0.2	192.168.0.1	DHCP	368	DHCP Request - Transaction ID 0xc4b57a8a
827	401.037780	192.168.0.1	192.168.0.2	DHCP	342	DHCP ACK - Transaction ID 0xc4b57a8a
883	433.181760	192.168.0.2	192.168.0.3	UDP	44	62139 → 65531 Len=2
885	435.390606	192.168.0.2	192.168.0.3	UDP	44	62140 → 65531 Len=2

5.0 TCP CLIENT IMPLEMENTATION

This chapter shows how TCP Client communication can be implemented with the following procedure:

1. Open the MPLAB X IDE and load the `TCPClientDemo.X` file (Refer TCP/IP Lite Stack.).
2. Open the Windows Command Prompt on your PC and type "ipconfig" and press <Enter> to find the IP address of your PC.
3. Open and scroll down in the file `tcp_client_demo.c`. Scroll down in the source file to modify the `UDP_DEMO_Initialize()` function. Update the server IP address (PC's) and port number. (See [Figure 5-1.](#))

FIGURE 5-1: TCP CLIENT INITIALIZE

```
void TCP_Client_Initialize(){  
    // TODO[2] - Initialize the server IP address with your PC's IP address  
  
    remoteSocket.addr.s_addr = MAKE_IPv4_ADDRESS(192, 168, 0, 2);  
    remoteSocket.port = 65534;  
}
```

4. In the `main.c` file from *Projects>Source Files*, call `UDP_DEMO_Initialize()`. (See [Figure 5-2.](#))

FIGURE 5-2: CALL TCP CLIENT INITIALIZE

```
TCP_Client_Initialize();  
  
while (1)  
{  
    .  
    .  
    .  
}
```

5. Initialize socket by modifying function `DEMO_TCP_Client()` in file `tcp_client_demo.c`. (See [Figure 5-3.](#))

FIGURE 5-3: INSERT SOCKET

```
case NOT_A_SOCKET:  
    //TODO[4] - Inserting and Initializing the socket  
  
    TCP_SocketInit(&port65534TCB);  
    break;  
case SOCKET_CLOSED:
```

6. Call `TCP_Connect()` server on port '65534' once at least 2 seconds. (See [Figure 5-4.](#))

FIGURE 5-4: CLIENT CALL TCP SOCKET

```

case SOCKET_CLOSED:
    // if the socket is closed we will try to connect again
    // try to connect once at 2 seconds
    socketTimeout = t_client + 2;
    TCP_InsertRxBuffer(&port65534TCB, rxdataPort65534, sizeof(rxdataPort65534));
    TCP_Connect(&port65534TCB, &remoteSocket);
    //TODO[5] - Connect to the Server

```

7. In `main.c`, call the **DEMO_TCO_Client()** function to start the device as a TCP Client. (See [Figure 5-5](#).)

FIGURE 5-5: CALL DEMO TCP CLIENT

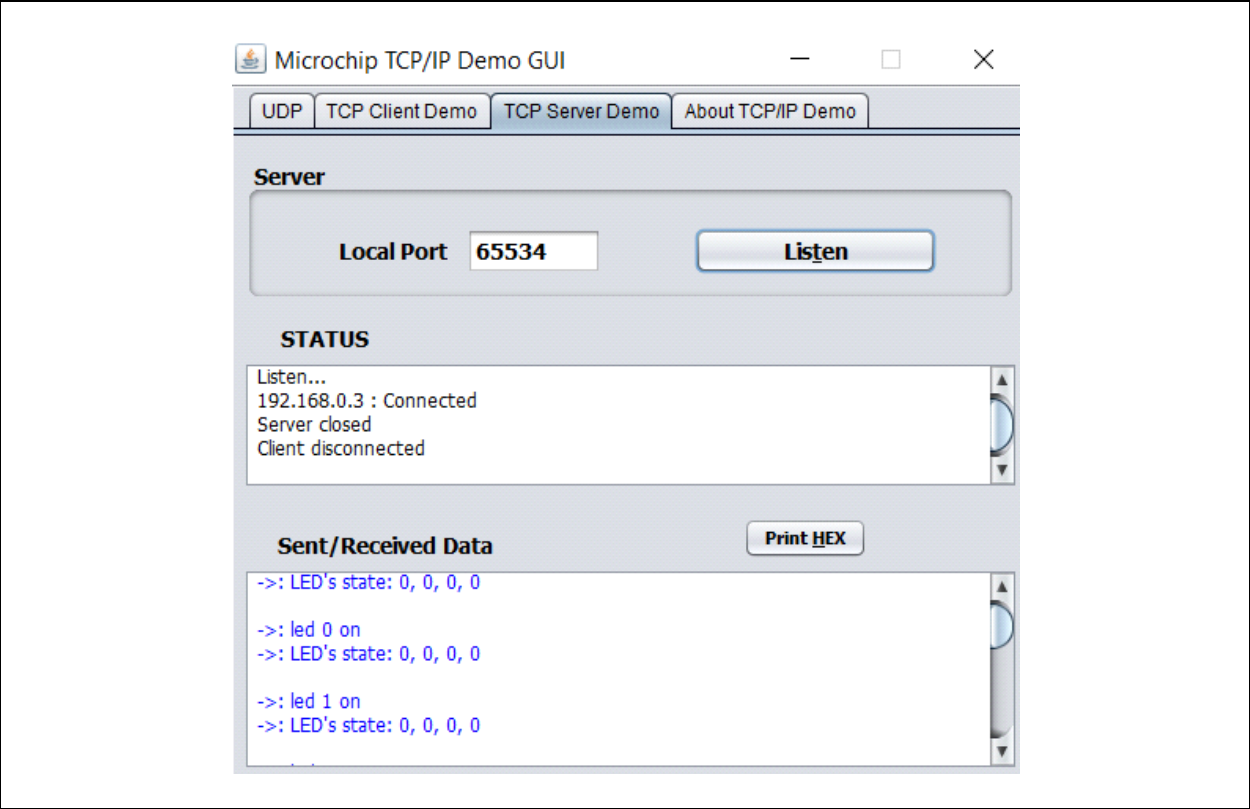
```

Network_Manage();
ip = ipdb_getAddress();
if(ip)
{
    DEMO_TCP_Client();
}

```

8. From the Production menu, click on **Clean and Build Project**. After a successful build, click on **Run** from the tool bar to program the device.
9. Open the Java application from the TCP/IP Demo GUI folder.
10. Go to the **TCP Server Demo** tab, assign the same port number as in Step 4; 65534. Click on the **Listen** button. The status of the TCP connection is printed inside the "STATUS" text box.
11. After the connection is established, the Explorer board sends the status of the LED's in a TCP packet. Click on the **LED** buttons (**0**, **1**, **2**, or **3**). This will toggle LEDs (**D1**, **D2**, **D3**, or **D4**). Board STATUS can be observed on the "Sent/Received Data" box in the GUI.
12. Push the **Disconnect** button, to close the TCP Connection. A client disconnected message will appear on the "STATUS" text box. (See [Figure 5-6](#).)

FIGURE 5-6: TCP CLIENT DEMO



6.0 TCP SERVER IMPLEMENTATION

This chapter details the following steps for implementing TCP server communication:

1. Open the MPLAB X IDE and load the `tcpServerDemo.X` file (Refer TCP/IP Lite Stack.).
2. Insert and initialize the Socket in `tcp_server_demo.c`. (See [Figure 6-1.](#))

FIGURE 6-1: TCP SERVER SOCKET INITIALIZE

```
case NOT_A_SOCKET:

//TODO[2] - Inserting and Initializing the socket
TCP_SocketInit(&port7TCB);
```

3. Bind the local port of the TCP Server to **Port 7**. (See [Figure 6-2.](#))

FIGURE 6-2: TCP SOCKET SERVER BIND

```
case SOCKET_CLOSED:

//TODO[3] - Configure the local port
TCP_Bind(&port7TCB, 7);
```

4. Start the **TCP Server**. (See [Figure 6-3.](#))

FIGURE 6-3: TCP SERVER LISTEN

```
// TODO[4] - Start the TCP server: Listen on port
TCP_Listen(&port7TCB);
```

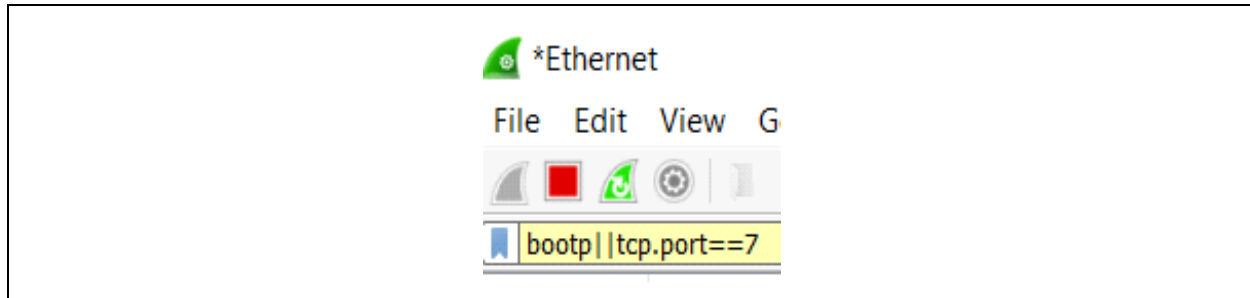
5. Call the **DEMO_TCP_EchoServer()** function within the `while(1)` loop in `main.c`. This allows the device to run as a TCP Server. (See [Figure 6-4.](#))

FIGURE 6-4: CALL TCP ECHO SERVER

```
if(ip)
{
    DEMO_TCP_EchoServer();
}
```

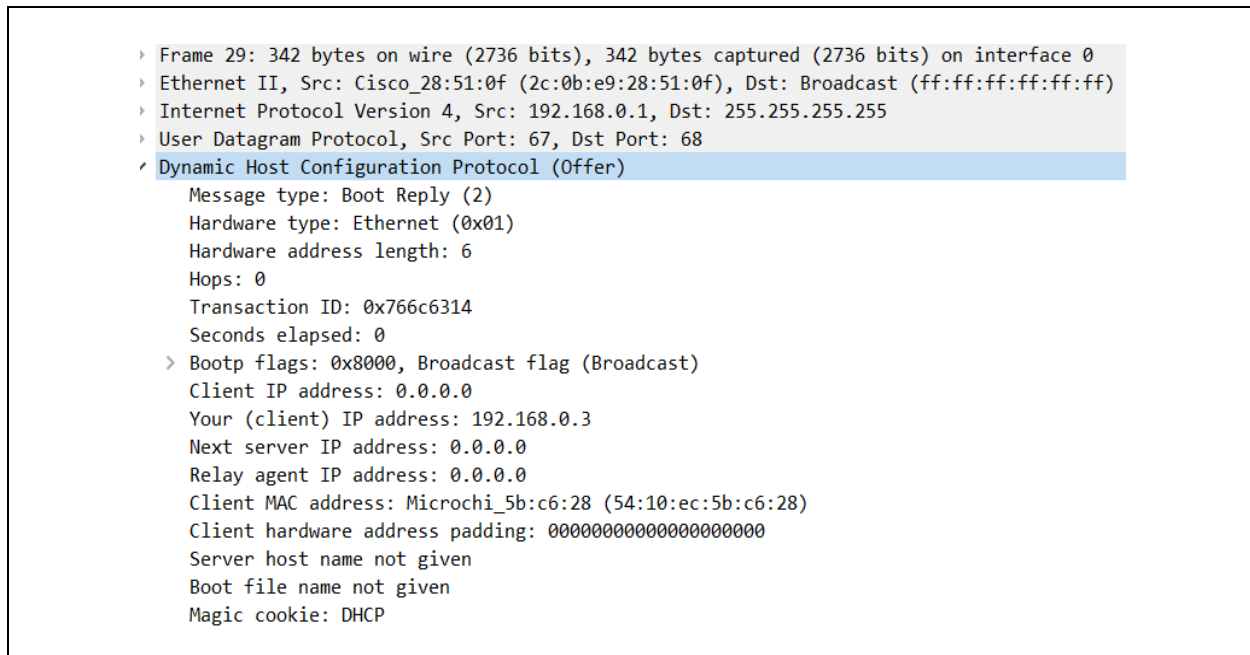
6. Launch Wireshark. From the Capture menu, click on **Options**.
7. Select an interface from the list to which your PC and the board are connected. Click on **Start** to capture packets.
8. In Wireshark, set the filter field as "bootp || tcp.port==7." (See [Figure 6-5.](#))

FIGURE 6-5: BOOTP TCP PORT 7



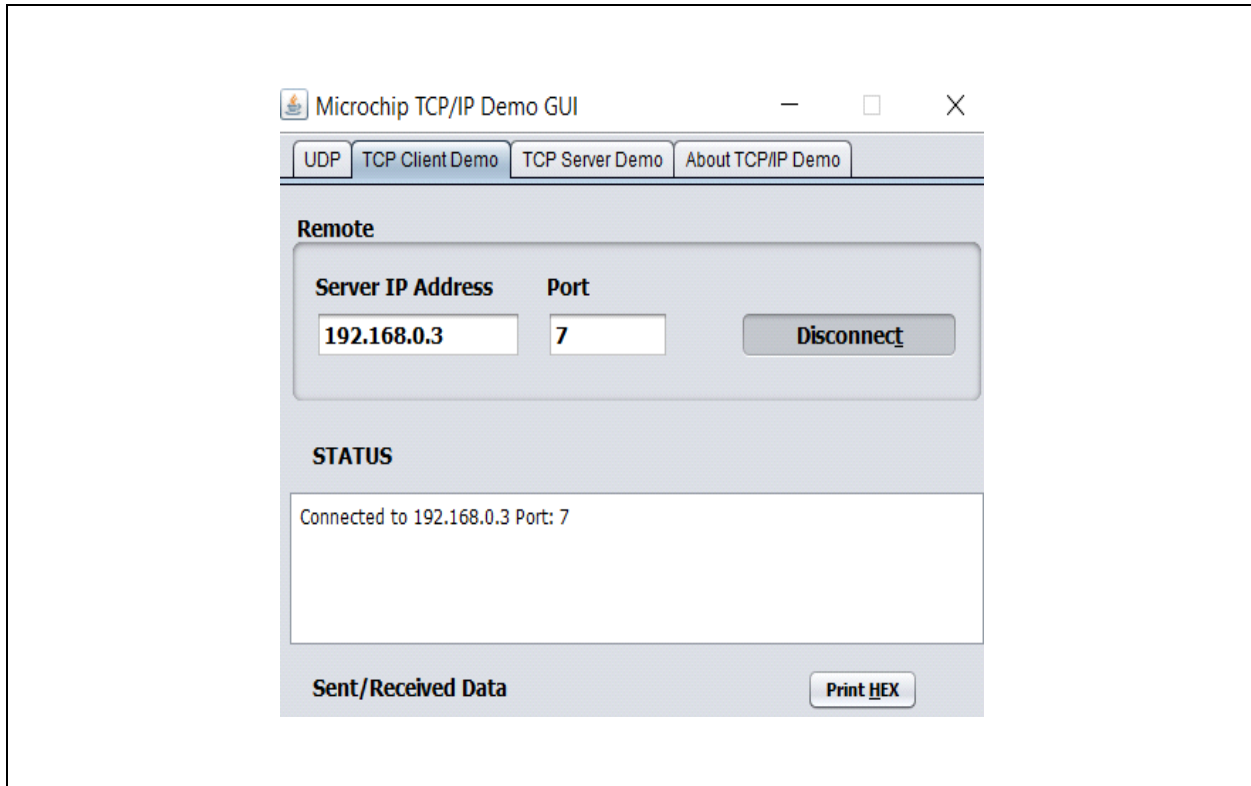
9. From the MPLAB IDE Production menu, click on **Clean and Build Project**. After successful build, click on **Run** from the tool bar to program the code to device.
10. In Wireshark, notice the DHCP packet handshake to get the device IP address.
11. Double-click on Offer packet. Expand Bootstrap protocol to get the device IP address. (See [Figure 6-6.](#))

FIGURE 6-6: TCP SERVER OFFER PACKET



12. Open the Java application from the TCP/IP Demo GUI folder.
13. Go to the **TCP Client Demo** tab, enter the IP address from step 12. Assign it as the Server IP address in the GUI. Assign the same port number in step 4. Click on the **Connect** button.
14. The status of the TCP Connection is printed inside the "STATUS" text box. (See [Figure 6-7.](#))

FIGURE 6-7: TCP CLIENT DEMO



15. After the connection is established, type text inside the "Send Text" box. Click on the **Send** button. The text will be echoed and displayed inside the "Sent/Received Data" box. Notice the ECHO message is divided and sent into 32 Bytes packets by the device.
16. Push the **Disconnect** button to close the TCP Connection. The "STATUS" text box indicates the connection message.

7.0 CONCLUSION

In summary, the basic steps for implementing UDP and TCP communication have been demonstrated in this guide. The user can build upon this application to extend the capabilities of the ENC624J600 controller with the help of the driver provided by the MCC.

NOTES:

APPENDIX A: DOCUMENTATION CONVENTIONS

This quick start guide uses the following conventions:

TABLE 7-1: DOCUMENTATION CONVENTIONS

Description	Represents	Examples
Arial Font:		
Italic characters	Referenced books	<i>MPLAB[®] IDE User's Guide</i>
	Emphasized text	...is the <i>only</i> compiler...
Initial caps	A window	the Output window
	A dialog	the Settings dialog
	A menu selection	select Enable Programmer
Quotes	A field name in a window or dialog	"Save project before build"
Underlined, italic text with right angle bracket	A menu path	<u><i>File>Save</i></u>
Bold characters	A dialog button	Click OK
	A tab	Click the Power tab
N'Rnnnn	A number in verilog format, where N is the total number of digits, R is the radix and n is a digit.	4'b0010, 2'hF1
Text in angle brackets < >	A key on the keyboard	Press <Enter>, <F1>
Courier New Font:		
Plain Courier New	Sample source code	#define START
	Filenames	autoexec.bat
	File paths	c:\mcc18\h
	Keywords	_asm, _endasm, static
	Command-line options	-Opa+, -Opa-
	Bit values	0, 1
	Constants	0xFF, 'A'
Italic Courier New	A variable argument	<i>file.o</i> , where <i>file</i> can be any valid filename
Square brackets []	Optional arguments	mcc18 [options] <i>file</i> [options]
Curly brackets and pipe character: { }	Choice of mutually exclusive arguments; an OR selection	errorlevel {0 1}
Ellipses...	Replaces repeated text	var_name [, var_name...]
	Represents code supplied by user	void main (void) { ... }

APPENDIX B: REVISION HISTORY

TABLE B-1: REVISION HISTORY

Revision Level & Date	Section/Figure/Entry	Correction
DS50002901A (07-16-19)	Initial release	

THE MICROCHIP WEBSITE

Microchip provides online support via our WWW site at www.microchip.com. This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip website at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the website at: <http://microchip.com/support>

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Klear, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TempTrackr, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, Vite, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KlearNet, KlearNet logo, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICKit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQL, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2019, Microchip Technology Incorporated, All Rights Reserved.

ISBN: 978-1-5224-4783-2

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
<http://www.microchip.com/support>
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880-3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7288-4388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820