1270 TCP1

Overview of Ethernet and TCP/IP Connectivity Solutions

Objectives

- Learn application possibilities for Ethernet and TCP/IP designs
- Become familiar with Microchip's Ethernet and TCP/IP offerings
- Understand the capabilities of several application protocols

© 2008 Microchip Technology Incorporated. All Rights Reserved.

Agenda

- Introducing Ethernet
- Microchip's Offerings
- Starting Your Design

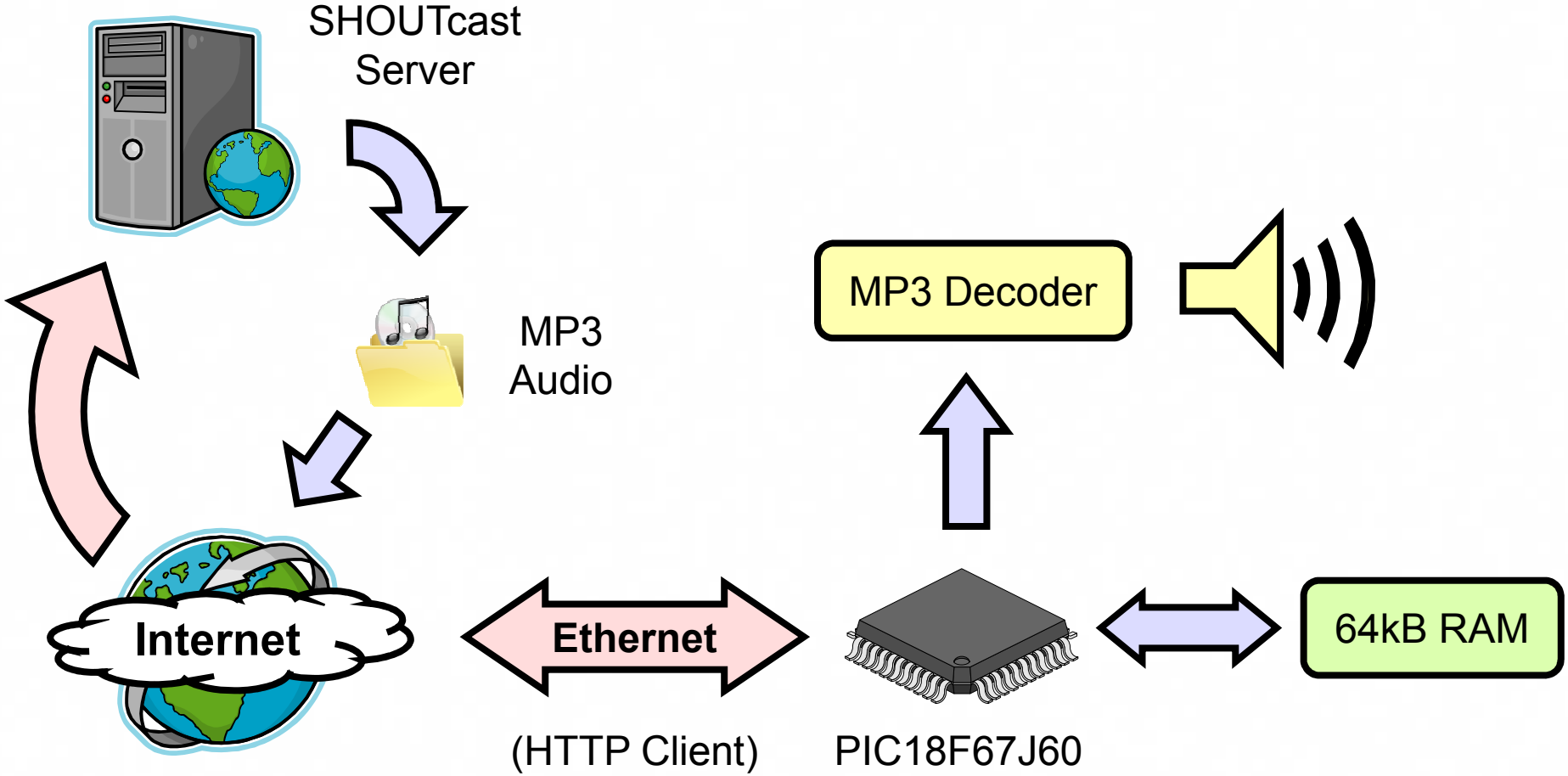
© 2008 Microchip Technology Incorporated. All Rights Reserved.

Internet Radio

- Internet Radio
  - Streams MP3 from SHOUTcast servers
  - Uses PIC18F67J60

© 2008 Microchip Technology Incorporated. All Rights Reserved.

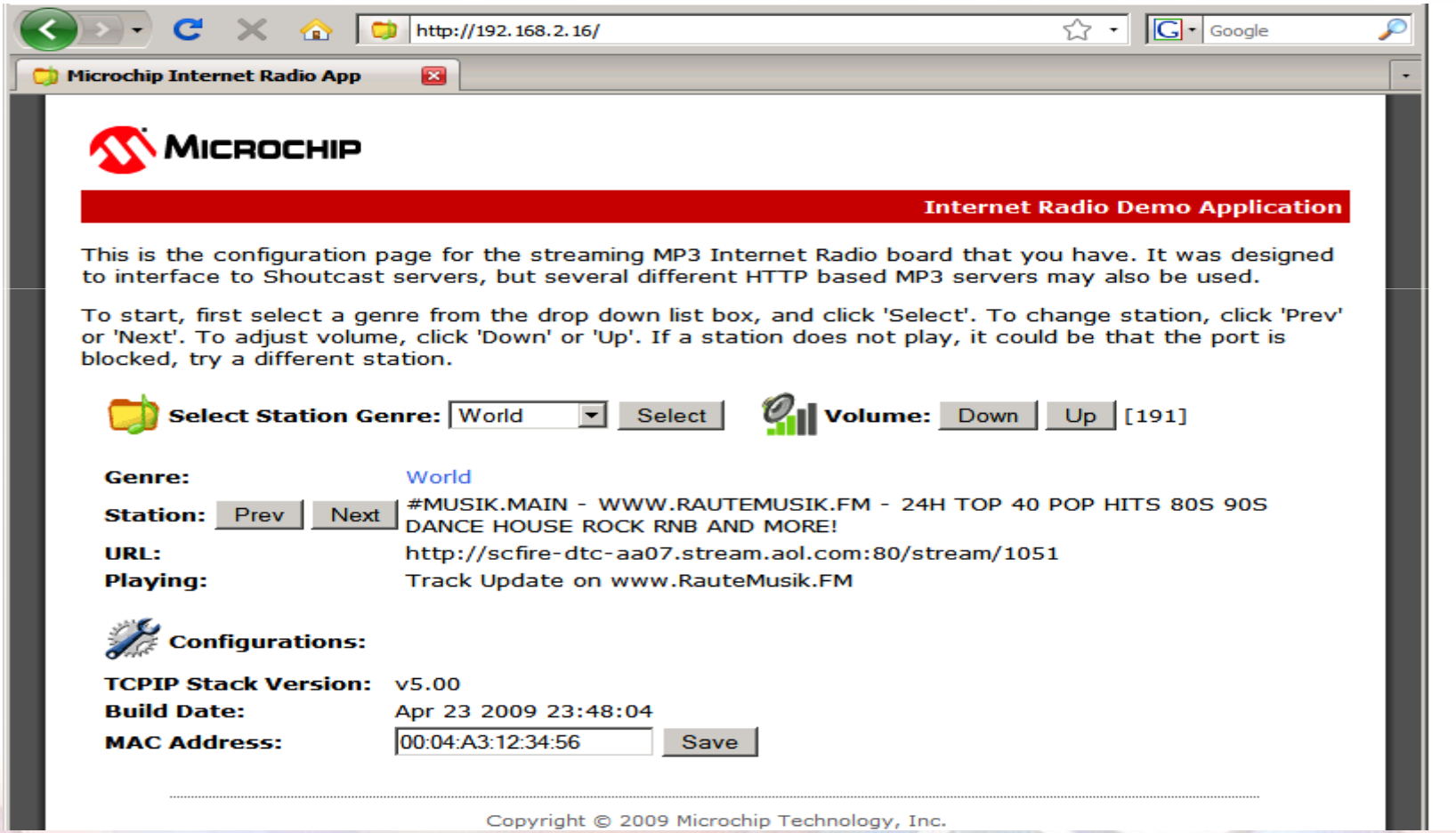
Internet Radio



© 2008 Microchip Technology Incorporated. All Rights Reserved.

Internet Radio

- **The** demo requires the Internet Radio Demonstration board (DM183033) available from www.microchipdirect.com and distributors.



© 2008 Microchip Technology Incorporated. All Rights Reserved.

Internet Bootloader App

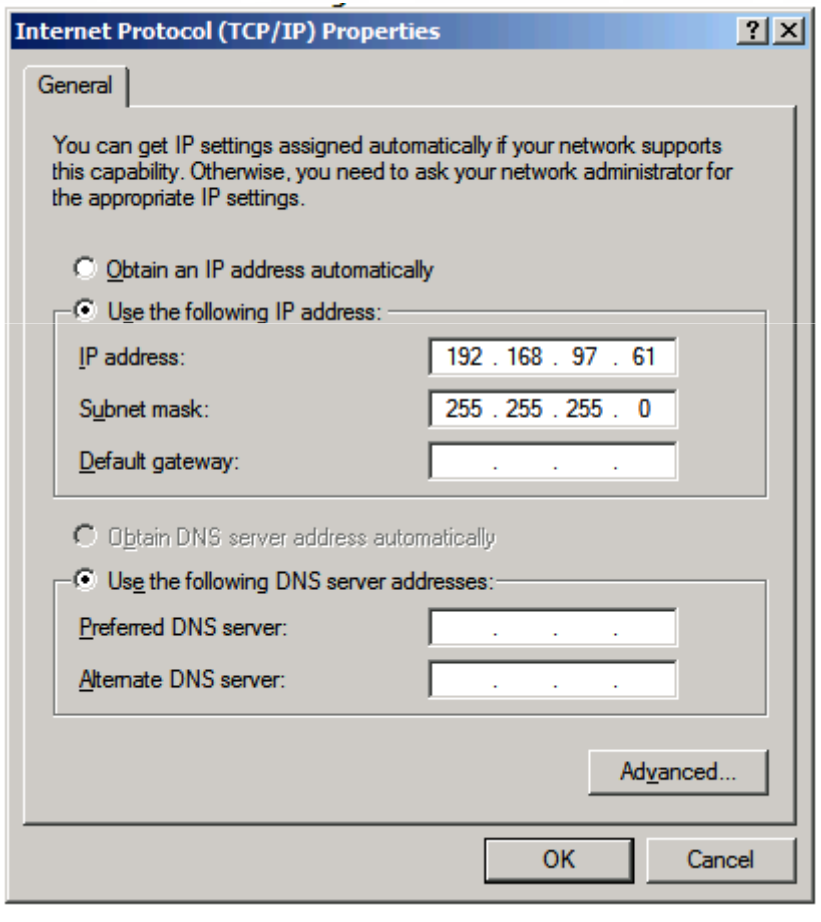
- **Internet Bootloader App**
  - **The Internet Bootloader is a stand alone application allowing new application firmware to be uploaded directly into the Flash memory of an embedded device over an Ethernet network or the Internet. It implements its own private UDP/IP stack as well as a Trivial File Transfer Protocol (TFTP) server. The bootloader operates independently of the main application and cannot update itself. Safeguards are implemented internally to minimize the risk of non-recoverable failed upgrades.**
- **PICDEM.net™**

DM163024

© 2008 Microchip Technology Incorporated. All Rights Reserved.

Using the Bootloader

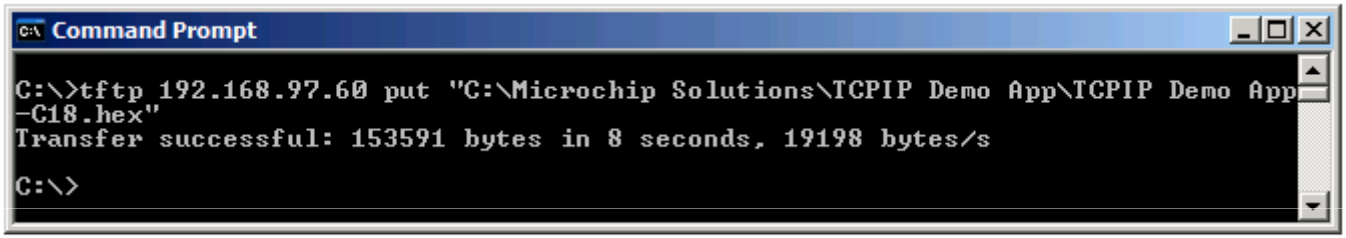
- **Configuring Your PC (Power-on Reset entry)**



© 2008 Microchip Technology Incorporated. All Rights Reserved.

Internet Bootloader App

- TFTP Operation (Power-on Reset entry)

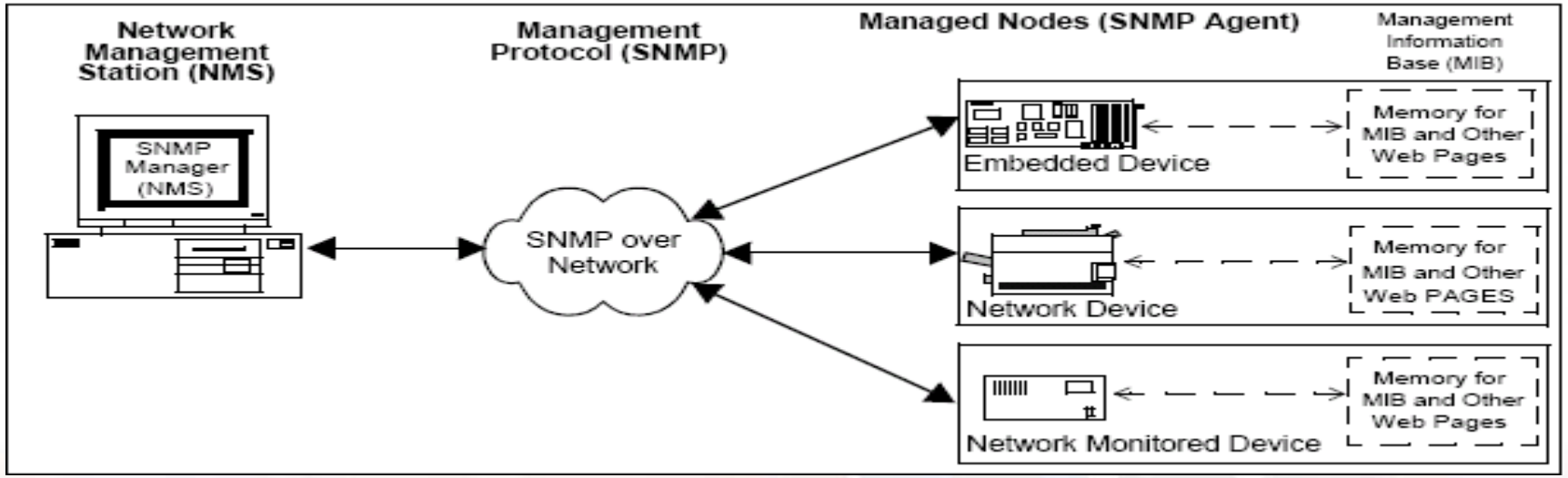


© 2008 Microchip Technology Incorporated. All Rights Reserved.

SNMP V2c Agent

- **Simple Network Management Protocol (SNMP) is one**

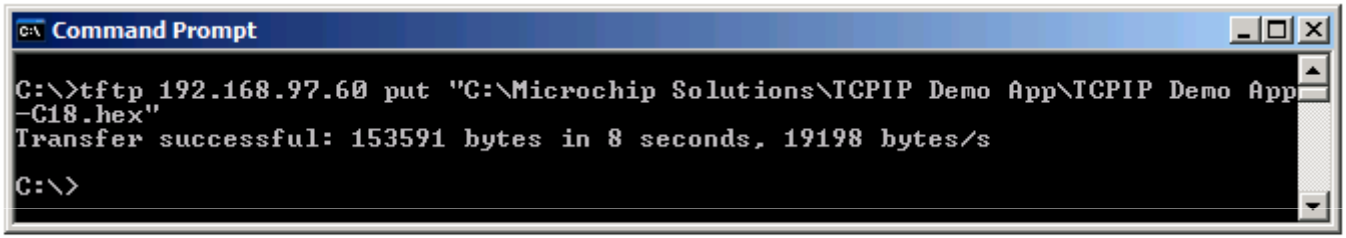
of the key components of a Network Management System (NMS). SNMP is an application layer protocol that facilitates the exchange of management information among network devices.



© 2008 Microchip Technology Incorporated. All Rights Reserved.

Internet Bootloader App

- TFTP Operation (Power-on Reset entry)



© 2008 Microchip Technology Incorporated. All Rights Reserved.

TCP/IP Stack

- Hardware just generates the signals
- Still need to:
  - Find other nodes and addresses
  - Transmit data to specific nodes
  - Manage incoming/outgoing packets
  - Communicate with other software

- Microchip TCP/IP Stack!

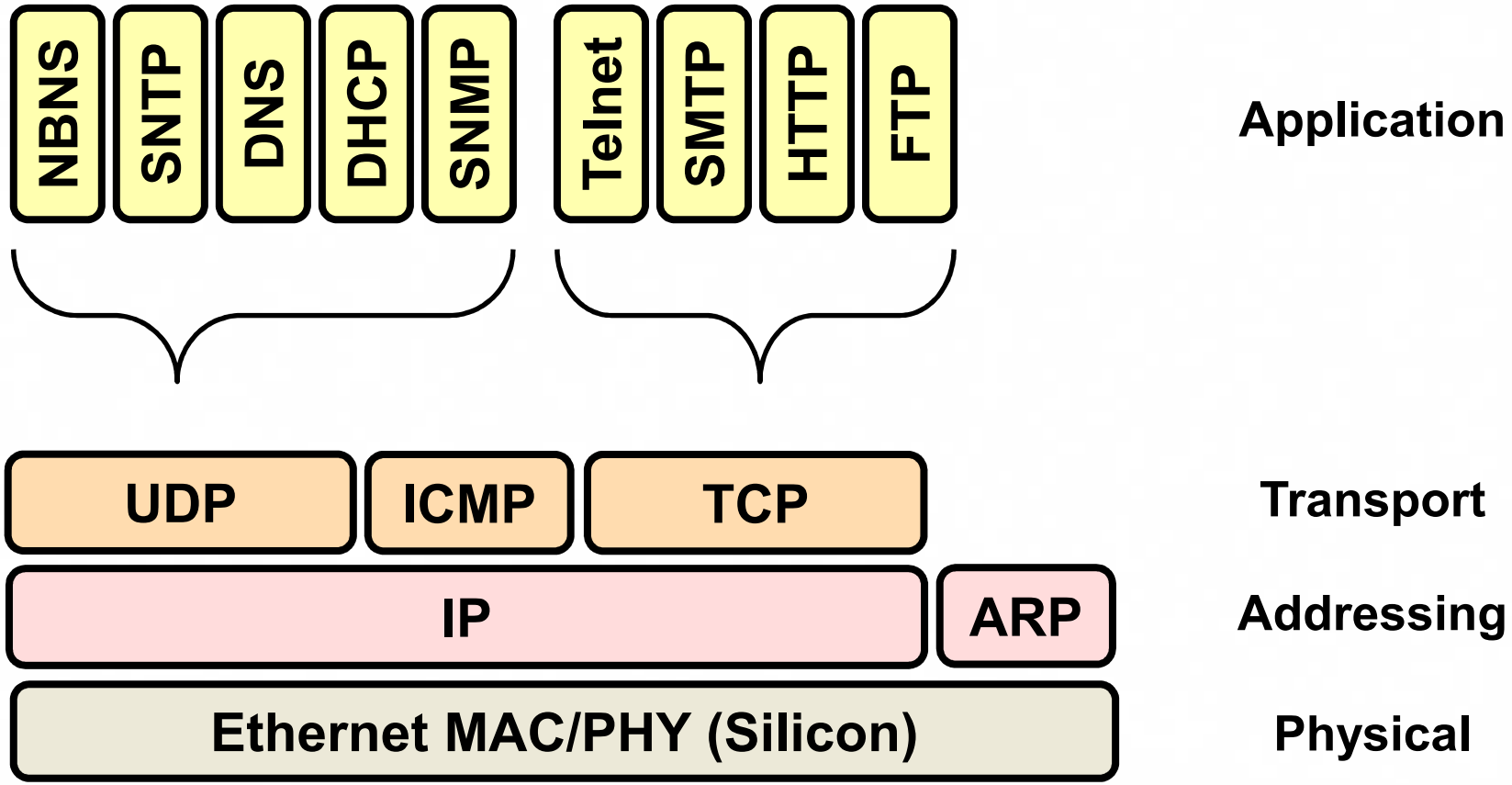
© 2008 Microchip Technology Incorporated. All Rights Reserved.

TCP/IP Stack

- C source code provided
  - No-fee license agreement
  - Use Microchip PIC® MCU or dsPIC® DSC
  - Only driver for ENC28J60 can be ported
  - Download: www.microchip.com/tcpip
- PIC18, PIC24, dsPIC DSC, PIC32
- RTOS Independent & Modular
- Supports multiple connections
- Example projects
- Standard Microchip technical support

© 2008 Microchip Technology Incorporated. All Rights Reserved.

What's Included?



© 2008 Microchip Technology Incorporated. All Rights Reserved.

What's Included?

HTTP	**Web Server** Serve web pages and process web form input
SMTP	**E-mail Client** Send e-mail or SMS messages
Telnet	**Command-Line Interface Server** Allow simple text-based monitoring and control
SNMP	**Simple Network Management Protocol Server** Aggregate enterprise monitor/control capabilities
TCP/UDP	**Generic Data Transports** Retrieve data from servers, or serve data to clients
SSL	**Secure Socket Layer** Encrypt communications over untrusted networks
Bootload	**TFTP Bootloader** Update embedded firmware remotely
DDNS	**Dynamic Domain Name Service Client** Associate host names with dynamic addresses

© 2008 Microchip Technology Incorporated. All Rights Reserved.

Secure Sockets Layer – SSL

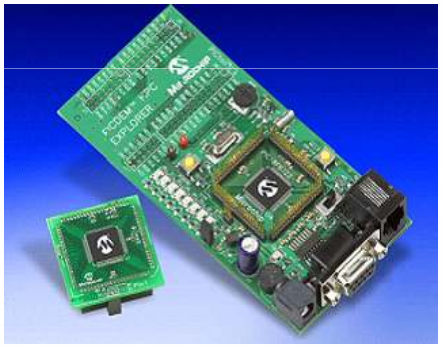
- SSL Module Available with Stack v4.50
- Required Encryption Routines are provided separately and require purchase
- www.microchipdirect.com
- SW300052 – Download or CD
- Subjected to Export Regulations

© 2008 Microchip Technology Incorporated. All Rights Reserved.

Microchip Development Tools

- **PICDEM.net™ 2 Board**
  (DM163024)
  - **PIC18F97J60**
  - **ENC28J60**

- **Ethernet PICtail™ Board**
  (AC164121)
  - **PICDEM™ HPC Explorer (DM183022)**
  - **PIC18 Explorer (DM183032)**



- **Ethernet PICtail Plus Board**
  (AC164123)
  - **Explorer 16 (DM240001)**
  - **PIC24, dsPIC® DSC, PIC32**

© 2008 Microchip Technology Incorporated. All Rights Reserved.

Starting Your Design

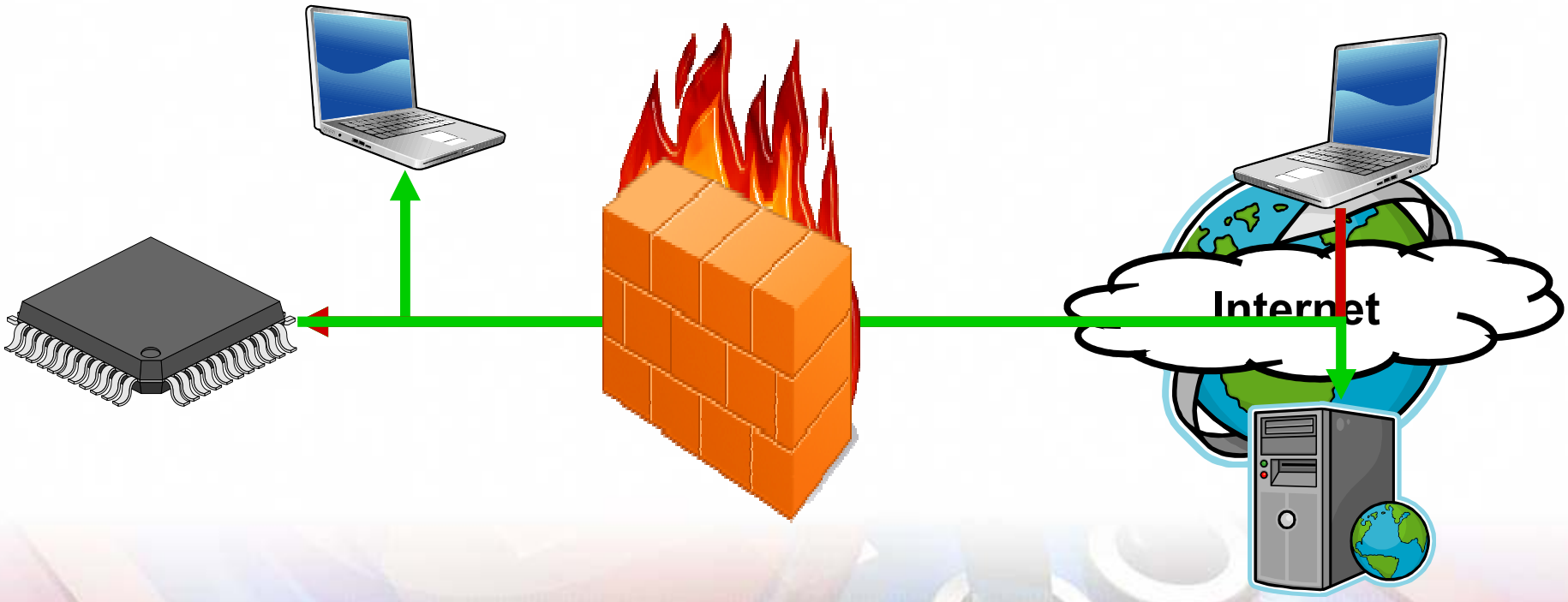
Server or Client?

- Server
  - Listens for connections from remote nodes
  - Always on, always waiting
  - Fixed location or address
- Client
  - Makes connection to remote node
  - Occasionally active – event driven
  - Unknown location or address
- Traffic flows both ways for either!
  - Difference is who initiates the connection

© 2008 Microchip Technology Incorporated. All Rights Reserved.

Server or Client?

- Firewalls & Routers
  - Block most incoming traffic
  - Allow most outgoing traffic



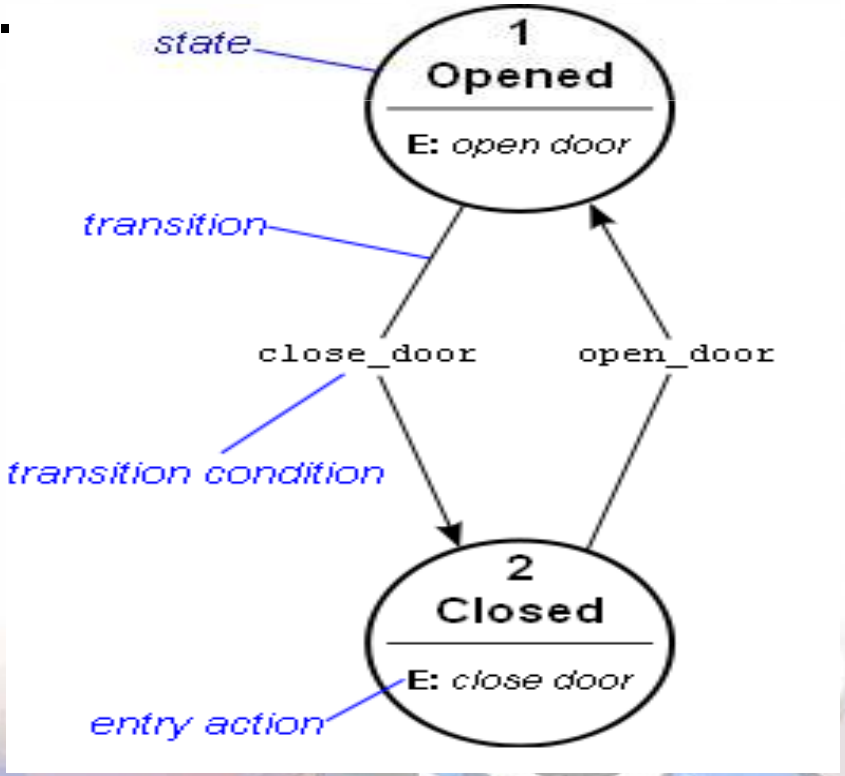
© 2008 Microchip Technology Incorporated. All Rights Reserved.

Software Architecture

© 2008 Microchip Technology Incorporated. All Rights Reserved.

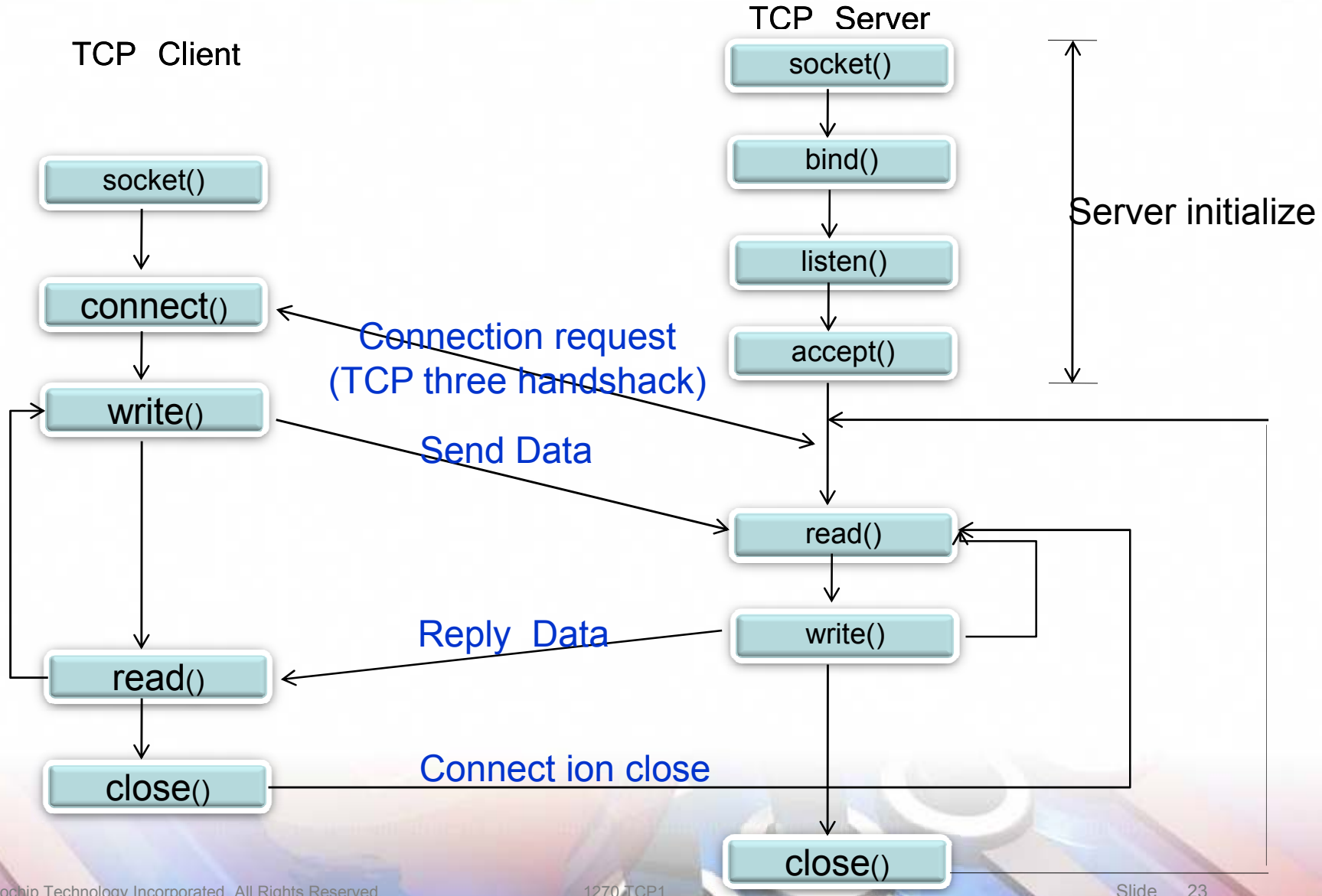
FSM(Finite-state machine)

- FSM is a model of behavior composed of a finite number of states, transitions between those states, and actions. A finite state machine is an abstract model of a machine with a primitive internal memory.



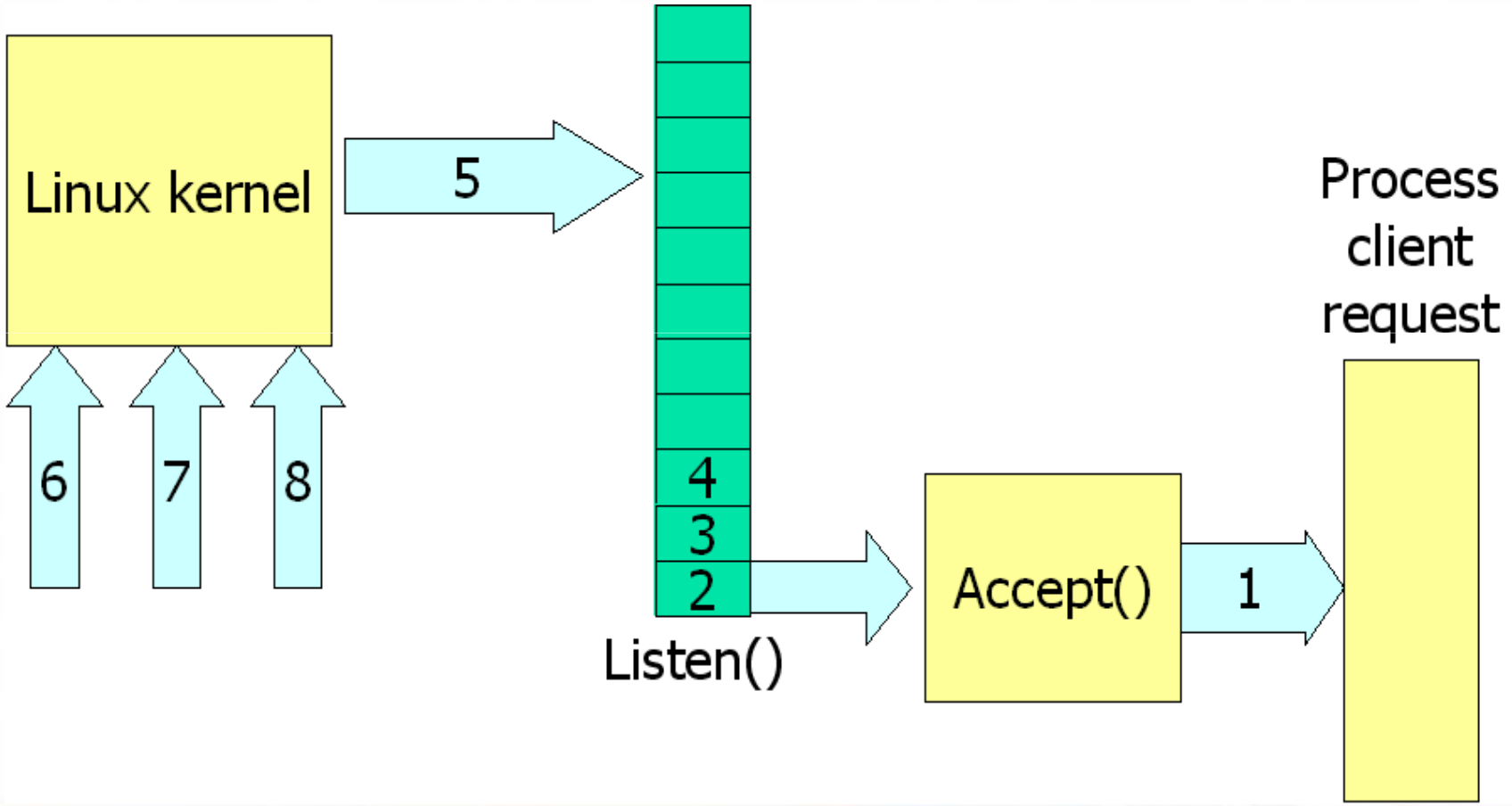
© 2008 Microchip Technology Incorporated. All Rights Reserved.

TCP Client-Server



© 2008 Microchip Technology Incorporated. All Rights Reserved.

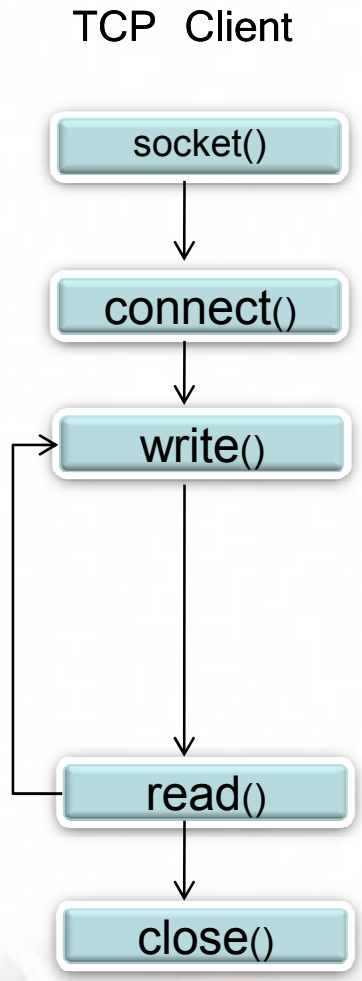
Linux TCP/IP



© 2008 Microchip Technology Incorporated. All Rights Reserved.

Linux Client Source Code

```
int main(int argc,char **argv)
{
    struct sockaddr_in addr_svr;
    int sockfd;
    char buffer[1024];
         // create server address
    memset(&addr_svr, 0, sizeof(addr_svr));
    addr_svr.sin_family= AF_INET;
    addr_svr.sin_port= htons(1234);
    addr_svr.sin_addr.s_addr = inet_addr("xxx.xxx.xxx.1");
    // create client socket
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    // connect
    connect(sockfd, (struct sockaddr *)&addr_svr, sizeof(addr_svr));
    // write to and read from server
    write(sockfd, buffer, strlen(buffer)+1);
    for(int len=0; ; ){
          len +=read(sockfd, buffer+len, 1024);
          if (len == 0)       break;
    }
    close(sockfd);
    return 0;
}
```



© 2008 Microchip Technology Incorporated. All Rights Reserved.

Microchip-- Main Code

```
int main(void)
{
    // Initialize application specific hardware
    InitializeBoard();
    // Initialize Stack and application related NV variables into AppConfig.
    InitAppConfig();
    StackInit();
    ZGConsoleInit( p_cmdStringPtrList, kZGNumCmdsInList);
    while(1)
    {
        // This task performs normal stack task including checking
        // for incoming packet, type of packet and calling
        // appropriate stack entity to process it.
        StackTask();

        // This tasks invokes each of the core stack application tasks
        StackApplications();
        ZGConsoleProcess();
        PingAppCall();
        BarCodeReaderDemo();
        ProcessIO();
    }
}
```

LOOP

© 2008 Microchip Technology Incorporated. All Rights Reserved.

Microchip—Client Source Code(1)

```
#define BARCodePORTNUM 1234
BYTE    sendBarCodeRequest[20] = "\0";
void BarCodeReaderDemo(void)
    {
    #if defined(STACK_USE_DNS)
          static SOCKET bsdClientSocket = INVALID_SOCKET;
          static struct sockaddr_in addr;
          char recvBuffer[9];
          int i;
          int addrlen;
          IP_ADDR RemoteIP;
          static enum _BARCodeServerState
          {
                    BARDCode_START,
                    BARDCode_CONNECT,
                    BARDCode_SEND,
                    BARDCode_OPERATION,
                    BARDCode_CLOSE,
                    BARDCode_DONE,
          } BARCodeClientState = BARDCode_DONE;

          switch(BARCodeClientState)
          {
```

© 2008 Microchip Technology Incorporated. All Rights Reserved.

Microchip—Client Source Code(2)

```
case BARDCode_START:
if (ZGConsoleIsConsoleMsgReceived() == kZGBoolTrue)
{
   if (isIPAddress(argv[1], address))
   {
      RemoteIP.v[0] = address[0]; RemoteIP.v[1] = address[1];
      RemoteIP.v[2] = address[2]; RemoteIP.v[3] = address[3];
      addr.sin_addr.S_un.S_addr=RemoteIP.Val;
      // Create a socket for this client to connect with
      if((bsdClientSocket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP))
                                          == INVALID_SOCKET )
           return;

        memset(sendBarCodeRequest,0,strlen((char*) sendBarCodeRequest));
        strcpy((char*) sendBarCodeRequest,(char*) argv[2]);
        BARCodeClientState = BARDCode_CONNECT;
    }
    else
    {
        ZG_PUTSUART( (char *) "Error sendbarcode command\n\r");
        ZGConsoleReleaseConsoleMsg();
    }
 }

 break;
```

© 2008 Microchip Technology Incorporated. All Rights Reserved.

Microchip-- Main Code

```
int main(void)
{
    // Initialize application specific hardware
    InitializeBoard();
    // Initialize Stack and application related NV variables into AppConfig.
    InitAppConfig();
    StackInit();
    ZGConsoleInit( p_cmdStringPtrList, kZGNumCmdsInList);
    while(1)
    {
        // This task performs normal stack task including checking
        // for incoming packet, type of packet and calling
        // appropriate stack entity to process it.
        StackTask();

        // This tasks invokes each of the core stack application tasks
        StackApplications();
        ZGConsoleProcess();
        PingAppCall();
        BarCodeReaderDemo();
        ProcessIO();
    }
}
```

LOOP

© 2008 Microchip Technology Incorporated. All Rights Reserved.

Microchip—Client Source Code(3)

```
case BARDCode_CONNECT:
 // addr.sin_addr.S_un.S_addr destination IP address was set earlier in DNS step
 addr.sin_port = BARCodePORTNUM;  //PORTNUM;
 addrlen = sizeof(struct sockaddr);
 if(connect( bsdClientSocket, (struct sockaddr*)&addr, addrlen) < 0)
       return;

 BARCodeClientState = BARDCode_SEND;
 break;

case BARDCode_SEND:
  //send TCP data
  send(bsdClientSocket, (const char*)sendBarCodeRequest, strlen((char*)
                                                        sendBarCodeRequest), 0);
  BARCodeClientState = BARDCode_OPERATION;
  break;
```

© 2008 Microchip Technology Incorporated. All Rights Reserved.

Microchip-- Main Code

```
int main(void)
{
    // Initialize application specific hardware
    InitializeBoard();
    // Initialize Stack and application related NV variables into AppConfig.
    InitAppConfig();
    StackInit();
    ZGConsoleInit( p_cmdStringPtrList, kZGNumCmdsInList);
    while(1)
    {
        // This task performs normal stack task including checking
        // for incoming packet, type of packet and calling
        // appropriate stack entity to process it.
        StackTask();

        // This tasks invokes each of the core stack application tasks
        StackApplications();
        ZGConsoleProcess();
        PingAppCall();
        BarCodeReaderDemo();
        ProcessIO();
    }
}
```

LOOP

© 2008 Microchip Technology Incorporated. All Rights Reserved.

Microchip—Client Source Code(4)

```
    case BARDCode_OPERATION:
        if(recv(bsdClientSocket, recvBuffer, 0, 2) < 0) //get the connection status
            BARCodeClientState = BARDCode_CLOSE;
        while(1)
        {
          i = recv(bsdClientSocket, recvBuffer, sizeof(recvBuffer)-1, 0);
          //get the data from the recv queue
          if(i <= 0)
          {
            //putsUART((char*) "receive null data\r\n");
            break;
          }
          putsUART((char*)recvBuffer);
          BARCodeClientState = BARDCode_CLOSE;
          break;
    case BARDCode_CLOSE:
          closesocket(bsdClientSocket);
          BARCodeClientState = BARDCode_DONE;
          break;
    case BARDCode_DONE:
          ZGConsoleReleaseConsoleMsg();
          BARCodeClientState = BARDCode_START;
          break;
    default:
          return;
}
```

© 2008 Microchip Technology Incorporated. All Rights Reserved.


Microchip Technology
MASTERS 08
CONFERENCE
YOU + MICROCHIP
ENGINEERING THE FUTURE TOGETHER

Thank You!

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KeeLoq, KeeLoq logo, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, PICkit, PICDEM, PICDEM.net, PICtail, PIC32 logo, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, Select Mode, Total Endurance, UNI/O, WiperLock 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.

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

© 2008, Microchip Technology Incorporated. All Rights Reserved.