**/\***

**\* Project name: Web-Based Control and Monitoring**

**\* Copyright:**

**\* (c) www.studentcompanion.co.za, 2014.**

**\* Test configuration:**

**\* MCU: PIC18F4520**

**\* Oscillator: 8.000 MHz**

**\* Compiler mikroC Pro for PIC v6.5.0**

**\*/**

**const char HTTPheader[] = "HTTP/1.1 200 OKnContent-type:";**

**const char HTTPMimeTypeHTML[] = "text/htmlnn";**

**const char HTTPMimeTypeScript[] = "text/plainnn";**

**//**

**// Define the HTML page to be sent to the PC**

**//**

**char StrtWebPage[] =**

**"**<html><body>

<form name=**"input"** method=**"get"**>

<table align=center width=500 bgcolor= LightGray border=2>

<tr><td align=center colspan=2><font size=5 color=RoyalBlue face=**"verdana"**>

<b>**Web-Based Control and Monitoring with PIC Microcontroller**</br></font></td>

</tr><tr><td height=**"200"** align=center bgcolor=LightGray><input name=**"TA"**

type=**"submit"**value=**"TOGGLE RELAY A"**></td>

<td height=**"200"** align=center bgcolor=LightGray><input name=**"TB"**

type=**"submit"**value=**"TOGGLE RELAY B"**></td></tr><tr>

<td align=center colspan=2><font size=2 color=RoyalBlue face=**"verdana"**>

</br></font></td></tr>

</table></form></body></html>**" ;**

**// Ethernet NIC interface definitions**

**//**

**sfr sbit SPI\_Ethernet\_Rst at RC0\_bit;**

**sfr sbit SPI\_Ethernet\_CS at RC1\_bit;**

**sfr sbit SPI\_Ethernet\_Rst\_Direction at TRISC0\_bit;**

**sfr sbit SPI\_Ethernet\_CS\_Direction at TRISC1\_bit;**

**//**

**// Define Serial Ethernet Board MAC Address, and IP address to be used for the communication**

**//**

**unsigned char MACAddr[6] = {0x00, 0x14, 0xA5, 0x76, 0x19, 0x3f} ;**

**unsigned char IPAddr[4] = {192,168,0,5};**

**unsigned char getRequest[10];**

**typedef struct**

**{**

**unsigned canCloseTCP:1;**

**unsigned isBroadcast:1;**

**}TethPktFlags;**

**//**

**// TCP routine. This is where the user request to toggle Realy 1 and Relaye 2 are processed**

**//**

**unsigned int SPI\_Ethernet\_UserTCP(unsigned char \*remoteHost,**

**unsigned int remotePort, unsigned int localPort,**

**unsigned int reqLength, TEthPktFlags \*flags)**

**{**

**unsigned int Len;**

**for(len=0; len**<10; len++)getRequest[len]=SPI\_Ethernet\_getByte();

getRequest[len]=0;

if(memcmp(getRequest,"GET /",5))return(0);

if(!memcmp(getRequest+6,"TA",2))RD0\_bit = ~ RD0\_bit;

else if(!memcmp(getRequest+6,"TB",2))RD1\_bit = ~ RD1\_bit;

if(localPort != 80)return(0);

Len = SPI\_Ethernet\_putConstString(HTTPheader);

Len += SPI\_Ethernet\_putConstString(HTTPMimeTypeHTML);

Len += SPI\_Ethernet\_putString(StrtWebPage);

return Len;

}

//

// UDP routine. Must be declared even though it is not used

//

unsigned int SPI\_Ethernet\_UserUDP(unsigned char \*remoteHost,

unsigned int remotePort, unsigned int destPort,

unsigned int reqLength, TEthPktFlags \*flags)

{

return(0);

}

//

// Start of MAIN program

//

void main()

{

ADCON1 = 0x0F; // All AN pins as digital

CMCON = 0x07; // Turn off comparators

TRISD = 0; // Configure PORTD as output

PORTD = 0;

SPI1\_Init(); // Initialize SPI module

SPI\_Ethernet\_Init(MACAddr, IPAddr, 0x01); // Initialize Ethernet module

while(1) // Do forever

{

SPI\_Ethernet\_doPacket(); // Process next received packet

}

}