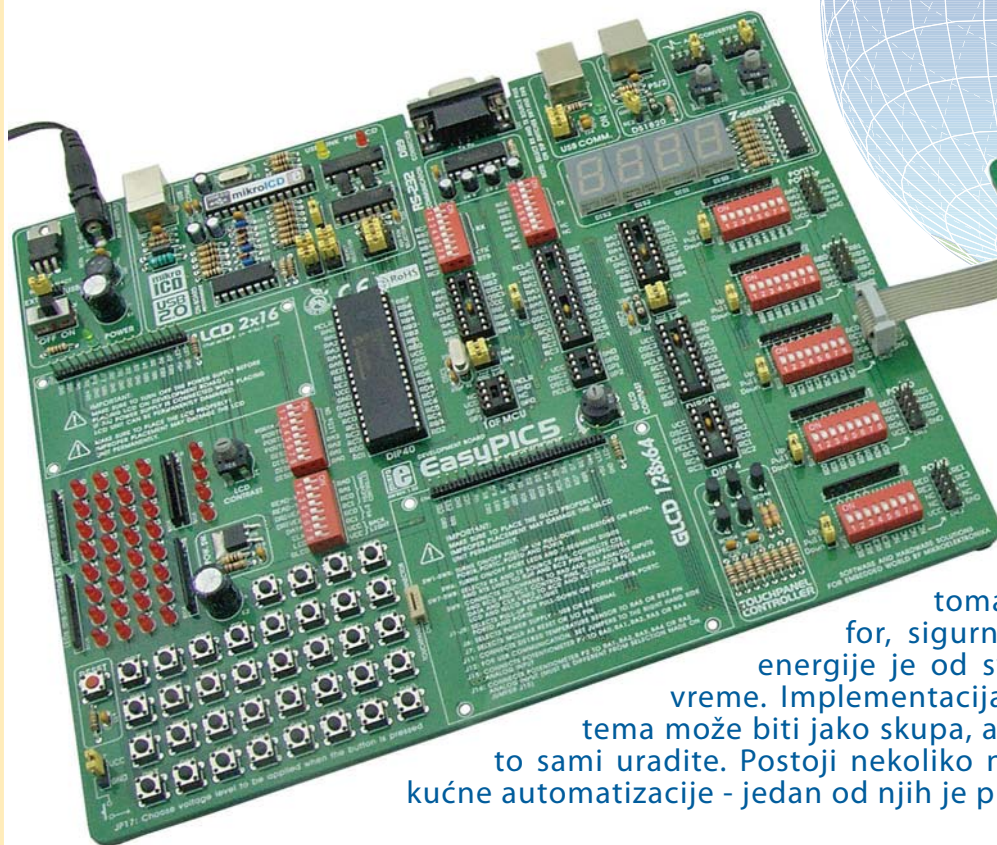


OK. Vama je potreban ... ETHERNET



Ploča serijskog eterneta povezana sa razvojnim sistemom EasyPIC5

Elektronski sistem kućne automatizacije je sinonim za komfor, sigurnost i uštedu energije. Ušteda energije je od sve većeg značaja u poslednje vreme. Implementacija/realizacija/ugradnja ovih sistema može biti jako skupa, ali i jako jeftina ako odlučite da to sami uradite. Postoji nekoliko načina da upravljate sistemom kućne automatizacije - jedan od njih je putem Eterneta.

Srdan Tomić
MikroElektronika - Software Department

Sve što je potrebno je jedan mikrokontroler PIC18F4520 i serijski ethernet čip ENC28J60. Serijska komunikacija čini ovaj čip odličnim rešenjem i za mikrokontrolere iz drugih familija kao što su AVR, dsPIC itd. Za povezivanje na ethernet mrežu korišćen je RJ-45 konektor firme CviLux pod oznakom CJC8 A8HF1Y0. LED dioda povezana na PORTB.0 mikrokontrolera predstavljaće uređaj u kući kojim želimo da upravljamo.

Kompajler *mikroBASIC for PIC* sadrži ethernet biblioteku koja će nam uveliko olakšati pravljenje samog programa za mikrokontroler. Uz pomoć svega par rutina iz ove biblioteke, napravićemo program koji će nam omogućiti paljenje i gašenje uređaja u kući putem web browser-a.

U programu je potrebno uraditi sledeće:

- Korak 1.** Napraviti html stranicu preko koje će se upravljati mikrokontrolerom i ubaciti je u kod u vidu string-a.
- Korak 2.** Setovati IP, DNS, Gateway adrese i Subnet masku dobijenu od strane internet provajdera.

Primeru radi, parametri naše lokalne mreže su:

IP: 192.168.20.60 (adresa kontrolnog sistema)
DNS: 192.168.20.1 (adresa Domain Name Sistema)
GATEWAY: 192.168.20.6 (adresa Gateway-a)
SUBNET: 255.255.255.0 (subnet maska)

Korak 3. Ugasiti analogne ulaze na portu B mikrokontrolera. Postaviti nulu na pin PORTB.0 i setovati ga kao izlazni.

Korak 4. Inicijalizovati SPI modul mikrokontrolera PIC18F4520.

Korak 5. Inicijalizovati serijski ethernet modul ENC28J60.

Korak 6. Unutar funkcije `Spi_Ethernet_userTCP` napisati kod koji će po prijemu komande poslati putem web browser-a upaliti/ugasiti LED diodu.

Korak 7. U neprekidnoj petlji iščitavati primljene podatke.

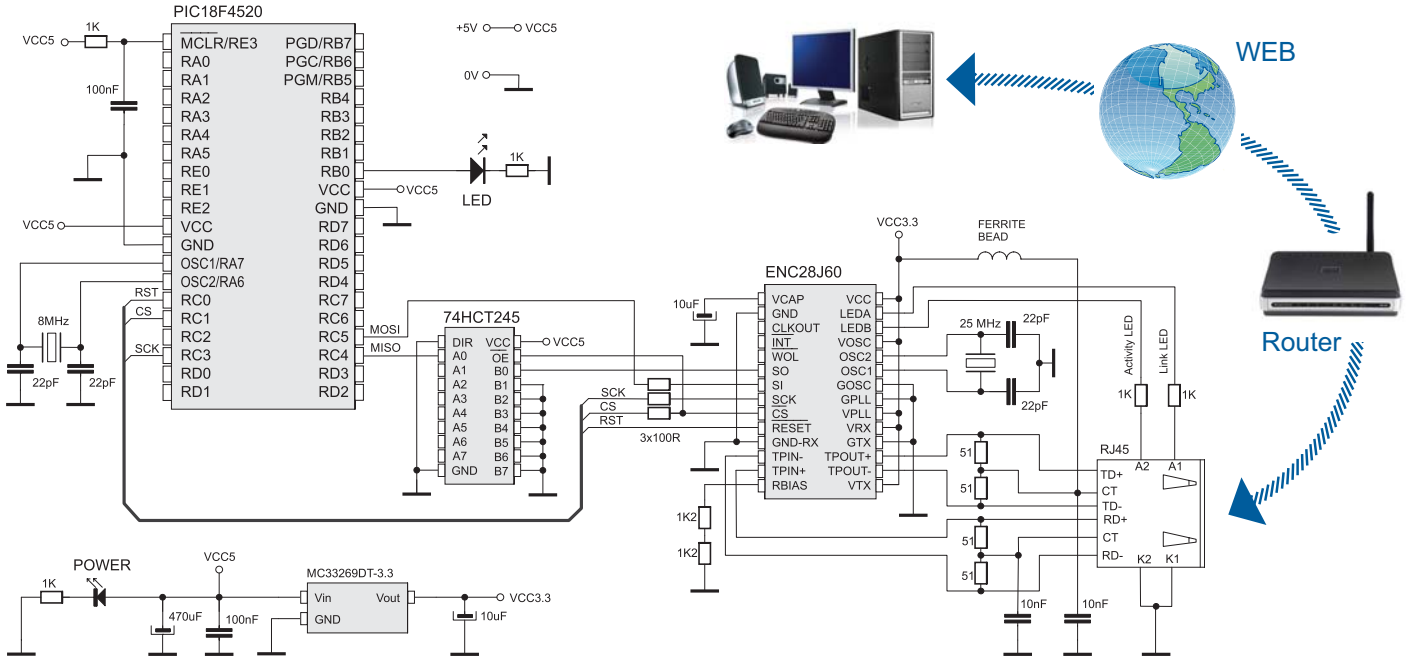
Najvažnija funkcija je `Spi_Ethernet_userTCP` u kojoj se odvija procesiranje svih primljenih komandi. Nakon prijema 'GET' zahteva poslatog sa vašeg računara putem web browser-a na IP adresu kontrolnog sistema, mikrokontroler će

browser-u vratiti web stranicu koja se nalazi u njegovoj memoriji. To je upravo ono što će biti prikazano na monitoru vašeg računara. Po prijemu ON komande LED dioda se pali, a po prijemu OFF komande ista se gasi. Ukoliko se umesto diode stavi relej može se ostvariti kontrola bilo kog uređaja kao što je svetlo, alarm, grejanje itd.

Sama kontrola se svodi na kucanje IP adrese kontrolnog sistema koji upravlja uređajima u kući u web browser i zadanje željenih komandi. Naravno, kako se kontroliše jedan pin mikrokontrolera tako se može kontrolisati i više pinova čime se dolazi do kompleksne kontrole većeg broja kućnih uređaja, odnosno, do sistema kućne automatizacije.

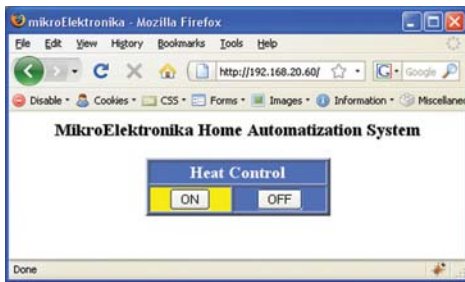


Slika 1. Pločica serijskog eterneta sa ENC28J60 čipom



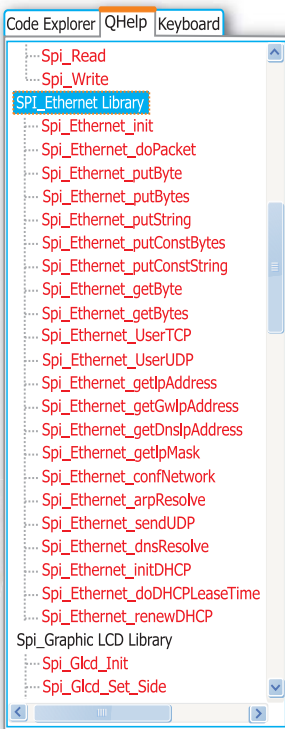
Šema 1. Povezivanje pločice serijskog ethernet i mikrokontrolera PIC18F4520

Na slici 2 je prikazana stranica koja se dobija u web browser-u nakon unošenja IP adrese kontrolnog sistema. U opisanom primeru, klikom na tastere ON i OFF dolazi do paljenja i gašenja LED diode čime se simulira upravljanje sistemom za grejanje.



Slika 2. Stranica web browser-a

SPI Ethernet Library biblioteka sa gotovim funkcijama sadržana u kompajleru mikroBASIC for PIC.



Spi_Ethernet_Init()	inicijalizuje ENC28J60 kontroler
Spi_Ethernet_Enable()	Omoгуuje mrežni saobraćaj
Spi_Ethernet_Disable()	Onemogućuje mrežni saobraćaj
Spi_Ethernet_doPacket()	Obraduje dobijeni paket podataka
Spi_Ethernet_putByte()	Upisuje bajt
Spi_Ethernet_putBytes()	Upisuje zahtevane bajtove
Spi_Ethernet_putConstBytes()	Upisuje zahtevane const bajtove
Spi_Ethernet_putString()	Upisuje string
Spi_Ethernet_putConstString()	Upisuje const string
Spi_Ethernet_getByte()	Uzima bajt
Spi_Ethernet_getBytes()	Uzima zahtevane bajtove
Spi_Ethernet_UserTCP()	Obraduje TCP
Spi_Ethernet_UserUDP()	Obraduje UDP
Spi_Ethernet_getIpAddress()	Uzimanje IP adrese
Spi_Ethernet_getGwIpAddress()	Uzimanje Gateway adrese
Spi_Ethernet_getDnsIpAddress()	Uzimanje DNS adrese
Spi_Ethernet_getIpMask()	Uzimanje IP maske
Spi_Ethernet_confNetwork()	Postavlja mrežne parametre
Spi_Ethernet_arpResolve()	Šalje zahtev za ARP
Spi_Ethernet_sendUDP()	Šalje UDP
Spi_Ethernet_dnsResolve()	Šalje zahtev za DNS
Spi_Ethernet_initDHCP()	Šalje zahtev za DHCP
Spi_Ethernet_doDHCPLeaseTime()	Obrada DHCP validnosti
Spi_Ethernet_renewDHCP()	Zahtev za DHCP obnovu

Spisak dodatnih funkcija korišćenih u programu:

- Spi_Init()** Inicijalizacija modula na mikrokontroleru
- memcpy()** Kopira RAM lokacije mikrokontrolera
- memcmp()** Upoređuje RAM lokacije mikrokontrolera

Primer 1: Program demonstracije rada serijskog ethernet

```

program home_auto
include "home_auto_utils"

dim myMacAddr as byte[6]
myIpAddr as byte[4]
gwIpAddr as byte[4]
ipMask as byte[4]
dnsIpAddr as byte[4]

main:
ADCON1 = 0x0F
CMCON = 0x07

PORTB.0 = 0
TRISB.0 = 0

indexPage =
"<html><head><title>mikroElektronika</title></head><body>"+
"<h3 align=center>MikroElektronika Home Automation System</h3>"+
"<form name="+Chr(34)+"input"+Chr(34)+" action="+Chr(34)+"&"+Chr(34)+" method="+
Chr(34)+" get"+Chr(34)+"><table align=center width=200 bgcolor=#4974E2 border=2><tr>"+
"<td align=center colspan=2><font size=4 color=white><b>Heat Control</b></font>"+
"</td></tr><tr><td align=center bgcolor=#4974E2><input name="+Chr(34)+"tst1"+
Chr(34)+" width=60 type="+Chr(34)+" submit"+Chr(34)+" value="+Chr(34)+" ON"+
Chr(34)+"></td><td align=center bgcolor=#FFFF00><input name="+Chr(34)+"tst2"+
Chr(34)+" type="+Chr(34)+" submit"+Chr(34)+" value="+Chr(34)+" OFF"+Chr(34)+">"+
"</td></tr></table></form></body></html>"

myMacAddr[0] = 0x00 myMacAddr[1] = 0x14 myMacAddr[2] = 0xA5
myMacAddr[3] = 0x76 myMacAddr[4] = 0x19 myMacAddr[5] = 0x3F

ipMask[0]=255 ipMask[1]=255 ipMask[2]=255 ipMask[3]=0
myIpAddr[0]=192 myIpAddr[1]=168 myIpAddr[2]=20 myIpAddr[3]=60
gwIpAddr[0]=192 gwIpAddr[1]=168 gwIpAddr[2]=20 gwIpAddr[3]=6
dnsIpAddr[0]=192 dnsIpAddr[1]=168 dnsIpAddr[2]=20 dnsIpAddr[3]=1

' starts ENC28J60 with: reset bit on PORTC.F0, CS bit on PORTC.F1,
my MAC & IP address, full duplex
Spi_Init()
' full duplex, CRC & MAC Unicast + MAC Broadcast filtering
Spi_Ethernet_Init(PORTC.0, PORTC.1,
myMacAddr, myIpAddr, Spi_Ethernet_FULLDUPLEX)

' dhcp will not be used here, so use preconfigured addresses
Spi_Ethernet_confNetwork(ipMask, gwIpAddr, dnsIpAddr)

while true
Spi_Ethernet_doPacket()
wend
end.
  
```

```

module home_auto_utils
const httpHeader as string[31] = "HTTP/1.1 200 OK"+chr(10)+"Content-type:"
const httpMimeTypeHTML as string[13] = "text/html"+chr(10)+chr(10)
const httpMimeTypeScript as string[14] = "text/plain"+chr(10)+chr(10)
' default html page
dim indexPage as string[523]
dim getRequest as byte[20]
' HTTP request buffer
implements
sub function Spi_Ethernet_UserTCP(dim byref remoteHost as byte[4],
dim remotePort, localPort, reqLength as word) as word
dim cnt as word
tmp as string[10]
' my reply length
if(localPort < 80) then
' listen only to web request on port 80
exit
end if
' get 10 first bytes only of the request, the rest does not matter here
for cnt = 0 to 14
getRequest[cnt] = Spi_Ethernet_getByte()
next cnt
tmp = "GET "
if(memcmp(getRequest, @tmp, 5) < 0) then
' only GET method
result = 0
exit
end if
tmp = "ON"
if(memcmp(getRequest+11, @tmp, 2) = 0) then
' do we have ON command
PORTB.0 = 1
set PORTB bit 0
else
tmp = "OFF"
if(memcmp(getRequest+11, @tmp, 3) = 0) then
' do we have OFF command
PORTB.0 = 0
clear PORTB bit 0
end if
end if
if(PORTB.0) then
tmp = "#FFFF00" memcpy(@indexPage+340, @tmp, 6)
tmp = "#4974E2" memcpy(@indexPage+431, @tmp, 6)
else
tmp = "#4974E2" memcpy(@indexPage+340, @tmp, 6)
tmp = "#FFFF00" memcpy(@indexPage+431, @tmp, 6)
' highlight (yellow) OFF
end if
cnt = Spi_Ethernet_putConstString(@httpHeader)
cnt = cnt + Spi_Ethernet_putConstString(@httpMimeTypeHTML)
cnt = cnt + Spi_Ethernet_putString(@indexPage)
' HTML page first part
' with HTML MIME type
result = cnt
' return to the library with the number of bytes to transmit
end sub
sub function Spi_Ethernet_UserUDP(dim byref remoteHost as byte[4],
dim remotePort, destPort, reqLength as word) as word
end sub
end.
  
```

NOTE: Pored proširene verzije ovog programa koji je napisan za PIC mikrokontrolere, sa našeg sajta možete preuzeti i verzije pisane za AVR i dsPIC mikrokontrolere. www.mikroe.com/en/article/

