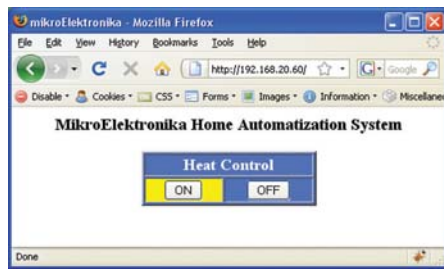


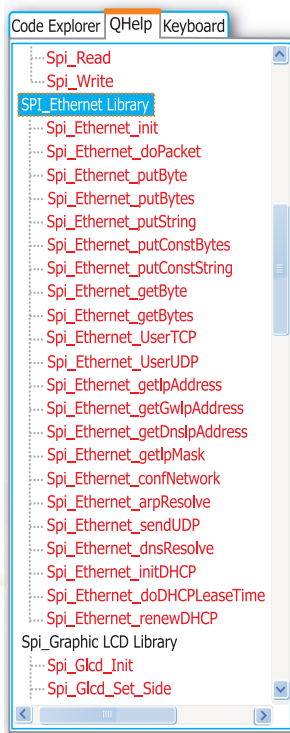
Schema 1. Aansluiten van de Seriële Ethernet-module op een PIC18F4520

Het besturen van een of ander huishoudelijk apparaat bestaat uit het in de webbrowser invoeren van het IP-adres van het controlesysteem en het specificeren van de gewenste opdrachten. Uiteraard is het mogelijk meer dan een microcontrollerpen aan te sturen, zodat u ook een groot aantal huishoudelijke apparaten als een compleet automatiseringssysteem kunt aansturen.



De schermafdruk illustreert de door de webbrowser geïmplementeerde webpagina als het IP-adres van het controlesysteem in ons voorbeeld wordt ingevoerd. Op de ON- of OFF-knop klikken schakelt de LED in en uit en simuleert op die manier het verwarming-sregelsysteem.

Onderstaande lijst van kant en klare functies is opgenomen in de SPI Ethernet Library. Deze bibliotheek maakt deel uit van de *microBASIC for PIC* compiler.



Spi_Ethernet_Init()	ENC28J60-controller initialiseren
Spi_Ethernet_Enable()	Netwerkverkeer inschakelen
Spi_Ethernet_Disable()	Netwerkverkeer uitschakelen
Spi_Ethernet_doPacket()	Ontvangen pakket verwerken
Spi_Ethernet_putByte()	Een byte opslaan
Spi_Ethernet_putBytes()	Bytes opslaan
Spi_Ethernet_putConstBytes()	Bytes continu opslaan
Spi_Ethernet_putString()	Tekenreeks opslaan
Spi_Ethernet_putConstString()	Tekenreeks continu opslaan
Spi_Ethernet_getByte()	Een byte ophalen
Spi_Ethernet_getBytes()	Bytes ophalen
Spi_Ethernet_UserTCP()	TCP-code afhandelen
Spi_Ethernet_UserUDP()	UDP-code afhandelen
Spi_Ethernet_getIPAddress()	IP-adres ophalen
Spi_Ethernet_getGwIPAddress()	Gateway-adres ophalen
Spi_Ethernet_getDnsIPAddress()	DNS-adres ophalen
Spi_Ethernet_getIPMask()	IP-masker ophalen
Spi_Ethernet_confNetwork()	Netwerkparameters instellen
Spi_Ethernet_arpResolve()	ARP-verzoek verzenden
Spi_Ethernet_sendUDP()	UDP-pakket verzenden
Spi_Ethernet_dnsResolve()	DNS-verzoek verzenden
Spi_Ethernet_initDHCP()	DHCP-verzoek verzenden
Spi_Ethernet_doDHCPLeaseTime()	Verwerk lease time
Spi_Ethernet_renewDHCP()	Verzoek tot DHCP-vernieuwing

In het programma gebruikte *.SPI Ethernet Library-functies.

Andere in het programma gebruikte *microBASIC for PIC*-functies.

- Spi_Init()** Microcontroller SPI-module initialiseren
- memcpy()** Microcontroller RAM-geheugenplaatsen kopiëren
- memcmp()** Microcontroller RAM-geheugenplaatsen vergelijken

Voorbeeld 1: Programma om sturing via Ethernet te demonstreren (two files)

```

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 'no analog inputs
CMCON = 0x07 'turn off comparators

PORTB.0 = 0
TRISB.0 = 0 'set PORTB.0 as output (rele control pin)

indexPage =
"<html><head><title>mikroElektronika</title></head><body>"+
"<h3 align=center>MikroElektronika Home Automatization System</h3>"+
"<form name="+Chr(34)+"input"+Chr(34)+" action="+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>Heat Control</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] = 0x5A
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() 'do forever
'process incoming Ethernet packets
wend
end.

module home_auto_utils
const httpHeader as string[31] = "HTTP/1.1 200 OK"+chr(10)+"Content-type:" 'HTTP header
const httpMimeTypeHTML as string[13] = "text/html"+chr(10)+chr(10) 'HTML MIME type
const httpMimeTypeScript as string[14] = "text/plain"+chr(10)+chr(10) 'TEXT MIME type

'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 ' highlight (yellow) ON
tmp = "#4974E2" memcpy@indexPage+431, @tmp, 6 ' clear OFF
else
tmp = "#4974E2" memcpy@indexPage+340, @tmp, 6 ' clear ON
tmp = "#FFFF00" memcpy@indexPage+431, @tmp, 6 ' highlight (yellow) OFF
end if

cnt = Spi_Ethernet_putConstString(@httpHeader) 'HTTP header
cnt = cnt + Spi_Ethernet_putConstString(@httpMimeTypeHTML) 'with HTML MIME type
cnt = cnt + Spi_Ethernet_putString(@indexPage) 'HTML page first part

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
result = 0
' back to the library with the length of the UDP reply
end sub
end.

```

Opn.: De voor dit voorbeeld in C, Basic en Pascal voor PIC® microcontrollers geschreven code staan, evenals de voor dsPIC® en AVR® microcontrollers geschreven programma's, op onze website: www.mikroe.com/en/article/.

