Tolmač

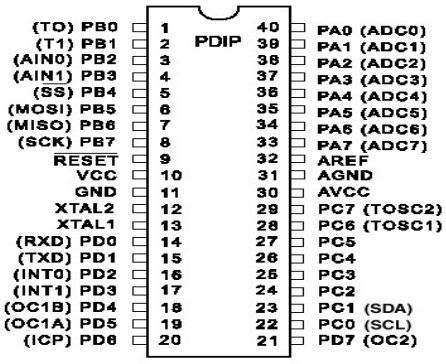
Boštjan Šuhel
Ekonomska Srednja Šola Celje
Vodnikova 10, 3000 Celje
bostjan.suhel@guest.arnes.si

INTERPRETER

Abstract:This paper talks about interpreter softveare named »Tolmač«. It is povered by Atmel AVR controlers. Sintaks is like industry languages, short and effective. It is good platform to develop inteligent sensors, small regulators and it's opened to network connections.

1 Uvod

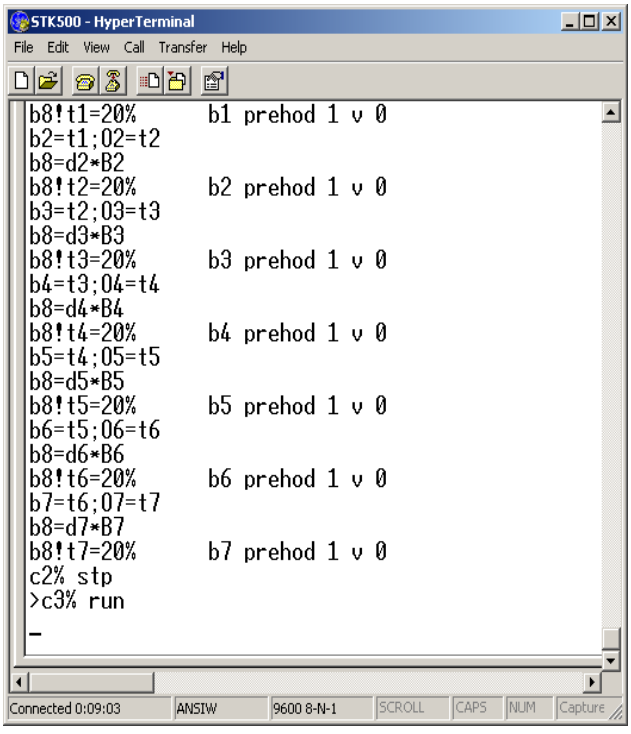
Članek govori o programu »Tolmač«. Napisan je za AVR[1] mikrokontrolerje. Pravopis je, kot se za industrijske jezike spodobi, kratek in učinkovit. Je dobra osnova za inteligentne senzorje in manjše krmilnike in je odprt za mrežno povezovanje.



Slika 1

Mešani razvoj programa s kombinacijo zbirnika in enega od programskih jezikov, je, zdi se tako, danes tista prava pot pri razvoju naprav. Vrednost proizvajalca razvojnega orodja se danes meri tudi po programski podpori, čimveč zunanjih enot in industrijskih standardov komunikacije. Tako se seveda bistveno dvigne storilnost in posledično konkurenčnost programerja. Seveda je pot do trga in uspeha poleg hitrega programiranja, zabeljena še z veliko tehnološkimi, organizacijskimi in drugimi problemi. Če se pa malo ozremo naokoli, pa vidimo, da danes pravzaprav projektanti pri svojem delu uporabljajo naprave, ki jih tudi programirajo. Seveda govorim o industrijskih krmilnikih, ki pa so danes v veliki večini primerov edina pot, za izdelavo malo in srednje serijskih proizvodov, da o avtomatizaciji industrije, ne govorim. Tolmač je torej program, ki sem ga

napisal na Atmelovem razvojnem sistemu STK500[2] in za razvojno orodje uporabil AVR basic. Slika 1 prikazuje shemo kontrolerja Atmega16. Slika 2 prikazuje primer zaslonskega izpisa pri delu s »Tolmačem«.



Slika 2

2 Sintaksa

Sintaksa jezika »Tolmač« je izredno enostavna. Sestavljena je iz dvo znakovnih spremenljivk in osnovnih matematičnih operacij. Spremenljivke so predznačene celoštevilčne, nepredznačene byte in boolean. Predznačene celoštevilčne spremenljivke imajo zalogo vrednosti –32768 do +32767. Nepredznačene byte spremenljivke imajo zalogo vrednosti 0 do 255. Boolean spremenljivke imajo zalogo vrednosti 0 in 1. Ukazovna vrstica je lahko dolga 14 znakov. »Tolmač« pozna dve osnovni stanji: run in stp. Ko je »Tolmač« v stanju run se izvaja program. Na terminalu izgine značilen znak »>«. Med izvajanjem programa lahko povsem brez omejitev izvajamo ročne komande. Ročna komanda ima prednost pred napisanim programom. seveda je problematičen izpis na terminal, če namreč naš program neprestano nekaj izpisuje na terminal, bo mešanica programiranih izpisov in našega tipkanja precej pokvarila zaslonsko sliko. Izvajanje programa zaustavimo s sistemskim ukazom c2(stp). Na terminalu se pojavi značilen znak »>«.

3 Izpis spremenljivk

```
>r0
1234
>
```

Ukaz ki je sestavljen iz imena spremenljivke nam izpiše vrednost le-te. Drugi znak imena spremenljivke je vedno številka od 0 do 9, ki nam označuje zaporedno številko spremenljivke. Za izpis prav vseh spremenljivk imamo na voljo sistemski ukaz c5.

4 Prirejanje

```
r0=1234
```

Ukaz priredi spremenljivki r0 vrednost 1234. Prirejanje celo številčnih spremenljivk je v mejah –32768 do +32767. Pri celoštevilčnem izračunavanju moramo paziti na kolobarjenje celoštevilčnega števila. Spremenljivke tipa boolean imajo vrednost 0 ali ena. Če je rezultat izraza različen od 0 dobi boolean spremenljivka vrednost 1 drugače pa 0.

```
r0=r0+12*r1
```

Zgornja formula se računa od znaka = na desno stran. Prioriteta računanja odstopa od matematičnih pravil in je sledeča. Najprej je izraz vrednost spremenljivke r0, nato se izrazu prišteje vrednost 12 in končno se vsota r0 + 12 pomnoži z vrednostjo r1. Poenostavljeno povedano izraz se računa od znaka = na desno brez upoštevanja matematičnih prioritet.

5 Oklepaji

```
r0=[r0+12]*r1
```

V izrazu lahko uporabimo zaviti oklepaj. Prioriteta računanja je v tem primeru sledeča. Najprej je izraz vrednost spremenljivke r0 , nato se izrazu prišteje vrednost 12. Dobimo vrednost znotraj oklepaja , ki je r0 + 12. Končno to vrednost pomnožimo z vrednostjo r1. Tolmač pozna samo eno globino oklepajev

6 Pogojni stavki

```
r0:r0=100
```

Ukaz ali ukazi za : se izvede(jo) če je vrednost spremenljivke r0 enaka 0.

```
r0!r0=100
```

Ukaz ali ukazi za ! se izvede(jo) če je vrednost spremenljivke r0 različna 0.

7 Primer programa

Napisani program bo prišteval spremenljivko r0, dokler ne bo preklopil histerezni bit h0. To se zgodi, ko je vrednost s0+e0 manjša od spremenljivke r0. s0 je celoštevilčna ram spremenljivka, ki ob resetu dobi vrednost 0. Spodnji preklop se dogodi, ko je s0-e0 večje od spremenljivke r0. Spremenljivka e0 je celoštevilčna eeram spremenljivka, ki obdrži vrednost, ko procesor resetiramo. Posledica izvajanja programa je utripanje digitalnega izhoda o0. Hitrost (frekvenco) utripanja pa lahko spreminjamo, tako , da spremenimo vrednost spremenljivke e0, kar med izvajanjem programa.

```
c2%          *stop*
e0=20
c1%          *prog*
h0:r0=r0+1
h0!r0=r0-1
o0=h0
c2%          *stop*
c3%          *run*
```

1. V prvi vrstici imamo sistemski ukaz c2, ki ustavi izvajanje tolmača. % pomeni brezpogojni konec vrstice in ga lahko uporabimo za zelo kratek komentar do 14. znaka.
2. V drugi vrstici smo priredili spremenljivki e0 vrednost 20. e0 je eeram spremenljivka, ki zadrži vrednost, če procesor resetiramo.
3. Sistemski ukaz c1 pomeni začetek programiranja
4. h0: pomeni, da se vrstica izvede, če je h0 enak 0. Torej spremenljivki r0 prištejemo ena, če je h0 enak 0. h0 je pa bitna spremenljivka histereze, ki je vezana na vrednost spremenljivke e0(velikost histereze) in spremenljivko r0 in s0 (celoštevilčne spremenljivke, ki ju primerjamo in preklapljamo ho po histerezi).
5. h0! pomeni, da se vrstica izvede, če je ho različen od 0. Torej spremenljivki r0 odštejemo ena , če je h0 različen od 0.
6. Izhodnemu bitu o0 priredimo vrednost histereznega bita h0
7. V sedmi vrstici prekinemo vnos tolmača
8. V osmi vrstici poženemo tolmač.

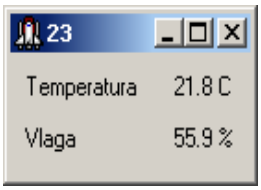
8 Podatki v Access bazo

```
Public retstr As String
Public retbyte
Public retcom
Public retvrsta
Public stevec
Private Sub Form_Load()
   MSComm1.PortOpen = True
End Sub

Private Sub MSComm1_OnComm()
   Select Case MSComm1.CommEvent
      Case comEvReceive   ' Received RThreshold
# of
           While (MSComm1.InBufferCount > 0)
             retbyte = MSComm1.Input
             If retbyte = Chr$(13) Then
               Select Case retcom
                 Case 1:
                    Label1.Caption =
Left(retstring, Len(retstring) - 1) & "." &
Right(retstring, 1) & " C"
                 Case 2:
                    Label2.Caption =
Left(retstring, Len(retstring) - 1) & "." &
Right(retstring, 1) & " %"
               End Select
               Select Case retstring
                 Case "u6": retcom = 1
                 Case "u7": retcom = 2
                 Case Else: retcom = 0
               End Select
               retstring = ""
             Else
               retstring = retstring & retbyte
             End If
           Wend
   End Select
End Sub

Private Sub Timer1_Timer()
    If stevec < 59 Then
      stevec = stevec + 1 ' stej sekunde
    Else
      On Error Resume Next 'Podatek v tabelo
tolmac
      Set povezava =
CreateObject("ADODB.Connection")
      povezava.Provider =
"Microsoft.Jet.OLEDB.4.0;"
      povezava.Open "./tolmac.mdb"
      Set rs = CreateObject("ADODB.Recordset")
      rs.Open "tolmac", povezava, 1, 3, 2
      rs.MoveFirst
      i = 0
      While (Not rs.EOF)
        i = i + 1
        rs.MoveNext
      Wend
      rs.MoveFirst
      For j = 0 To i - 15
        rs.Delete
        rs.MoveNext
      Next
      rs.AddNew
```

```
      rs("vlaga") = Label2.Caption
      rs("temperatura") = Label1.Caption
      rs.Update
      rs.Close
      povezava.Close
      stevec = 0
    End If
    Tolmac.Caption = 60 - stevec
    Select Case retvrsta 'vsako sekundo nov ukaz
        Case 1: MSComm1.Output = "u6" & Chr$(13)
        Case 2: MSComm1.Output = "u7" & Chr$(13)
        Case Else: retvrsta = 0
    End Select
    retvrsta = retvrsta + 1
End Sub
```



Slika 3

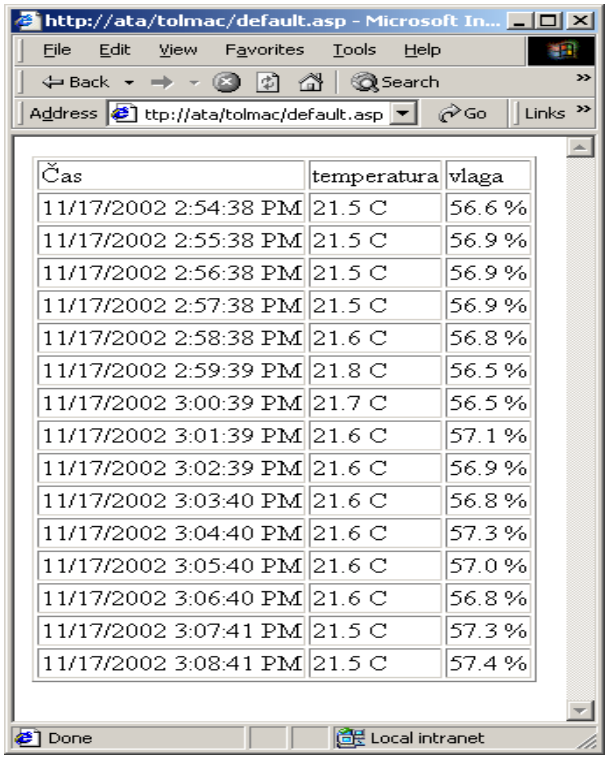
Program(Slika 3), ki prebere podatke iz Tolmača in jih spravi v tabelo podatkovne baze Access, je pravzaprav kratek in ga lahko dobite na moji spletni strani. Sestavljen je iz dogodka MSComm1_OnComm in Timer1_Timer .Prvi dogodek se proži, ko prispe znak iz tolmača v PC preko RS232 vodila, drugi dogodek se proži vsako sekundo. Časovni dogodek izmenično pošilja ukaze u6 in u7 vsako sekundo in dodatno vsake 60 sekund zapiše rezultate v podatkovno bazo. Odgovore Tolmača lovi dogodek prispetja znaka v RS232 vodilo.

9 Objava podatkov v svetovnem spletu

Tu nastopi poleg AVR Basica, Visual Basica še VBScript[5] in ADO[5] ob sočasni uporabi tehnologije HTML[5][6]. Torej do sedaj imamo program Tolmač na ATmega16 in Windows program Tolmac, dodajmo še Strežniški ASP[6] program zopet napisan v Basicu. Namestitveni program namesti bazo in program na \Program Files\Tolmac. Sedaj pa omenjeno mesto razglasimo za navidezno mapo IIServerja. Dajmo ji ime »tolmac« . Več o sistemskih nastavitvah si lahko preberete v navodilih programa. Sam izgled programa, kakor vidimo na sliki je precej špartanski. Napišem naj, da se s tolmačem pogovarja preko COM1 vodila.

```
<%
Dim i
On Error Resume Next
set
povezava=Server.CreateObject("ADODB.Connection")
povezava.provider="Microsoft.Jet.OLEDB.4.0;"
povezava.open server.mappath("./tolmac.mdb")
Set rs = Server.CreateObject("ADODB.Recordset")
rs.Open "tolmac", povezava
rs.MoveFirst
Response.Write "<HTML>" & vbCrLf
Response.Write "<HEAD>" & vbCrLf
Response.write "<META HTTP-EQUIV=""refresh""
CONTENT=""3;URL=./default.asp"">"
Response.Write "<META HTTP-EQUIV=""Content-
Type"" CONTENT=""text/html; charset=UTF-8"">" &
vbCrLf
Response.Write "</HEAD>" & vbCrLf
Response.Write "<BODY>" & vbCrLf
Response.Write "<TABLE border=1>"& vbCrLf
Response.write "<TR><TD>&#268;as</TD>" & vbCrLf
Response.write "<TD>temperatura</TD>" & vbCrLf
Response.write "<TD>vlaga</TD></TR>" & vbCrLf
i = 0
while (not rs.EOF)
      i = i + 1
      Response.write "<TR><TD>" &
rs("time_tolmac") & "</TD>" & vbCrLf
      Response.write "<TD>" & rs("temperatura")
& "</TD>" & vbCrLf
      Response.write "<TD>" & rs("vlaga") &
"</TD></TR>" & vbCrLf
      rs.MoveNext
wend
Response.Write "</TABLE></BODY></HTML>" & vbCrLf
Response.End
rs.close
povezava.close
%>
```

Strežniški ASP(Active Server Pages)[5] ima končnico (klaso) .asp kar za strežnik pomeni to, da ob klicu datoteke, izvede program in vrne uporabniku rezultat programa. Pri vračanju lahko programer uporabi katero od številnih internetnih tehnologij. Zgornji program zaradi čim večje razumljivosti uporablja HTML[6]. Delo programa je enostavno, odpre podatkovno bazo, izpiše petnajst podatkov v tabelo in zapre podatkovno bazo. Aplikacije, ki smo si jih odprli s takim pristopom so kar številne. Če odštejemo ceno računalnika, je zgoraj opisana pot dobra za nadzor hišnih aparatov, daljinsko spremljanje parametrov in še bi se našlo. Vsekakor pa bo moja pozornost poslej usmerjena v sisteme z neposrednim komuniciranjem preko RJ48 in IP protokola[8] z obvezno nadgradnjo v internetni strežnik. Ne rabimo ne računalnika, ne licenc, rabimo samo priklop na medmrežje in že smo dosegljivi od povsod. Slika 4 nam prikazuje izpis podatkov v internetnem brkljalniku.



Čas	temperatura	vlaga
11/17/2002 2:54:38 PM	21.5 C	56.6 %
11/17/2002 2:55:38 PM	21.5 C	56.9 %
11/17/2002 2:56:38 PM	21.5 C	56.9 %
11/17/2002 2:57:38 PM	21.5 C	56.9 %
11/17/2002 2:58:38 PM	21.6 C	56.8 %
11/17/2002 2:59:39 PM	21.8 C	56.5 %
11/17/2002 3:00:39 PM	21.7 C	56.5 %
11/17/2002 3:01:39 PM	21.6 C	57.1 %
11/17/2002 3:02:39 PM	21.6 C	56.9 %
11/17/2002 3:03:40 PM	21.6 C	56.8 %
11/17/2002 3:04:40 PM	21.6 C	57.3 %
11/17/2002 3:05:40 PM	21.6 C	57.0 %
11/17/2002 3:06:40 PM	21.6 C	56.8 %
11/17/2002 3:07:41 PM	21.5 C	57.3 %
11/17/2002 3:08:41 PM	21.5 C	57.4 %

Slika 4

10 Praksa

Kako vse skupaj izgleda si lahko pogledate na mojem domačem strežniku http://suhel.homeip.net/ [4], kjer lahko na gumbu »Tolmač« odčitate temperato in vlago v naši kleti zadnjih 15 minut, na gumbu »V živo« pa vidite STK500 razvojni komplet z naloženim programom Tolmač. Za boljši pogled sem vpisal program »leteča luč«(Slika 5). Na Arnes naslovu http://www2.arnes.si/~bsuhel/[3] si lahko preberete vse o tolmaču in si snamete željene programe.

```
c2% stp
c1% prg
t0:t0=16
o0=t0<16
o1=t0<14
o2=t0<12
o3=t0<10
o4=t0<8
o5=t0<6
o6=t0<4
o7=t0<2
c2% stp
c3% run
```

Slika 5

11 Zaključek

Kakršno koli razmišljanje o opravljenem delu ni na mestu. Članek opiše teško vendar zanimivo pot , k avtomatizaciji. Čaka me še veliko dela. Pot do uporabnih izdelkov je znana in tu ni iluzij. Kakor tudi čakanje na odziv trga, ki bo pograbil »perspektivne izdelke« spada v preživelo razmišljanje. Za resnejše konkuriranje moramo imeti kritično maso razvijalcev in zagotovljene stabilne in mirne pogoje za delo. To je pa že stvar politike.

Literatura

[1]http://www.mcselec.com/bascom-avr.htm – proizvajalec AVR basica
[2]http://www.atmel.com/dyn/resources/prod_documents/doc1925.pdf – Razvojni komplet STK500
[3]http://www2.arnes.si/~bsuhel/hoby/tolmac_guide.htm – Opis Tolmača
[4]http://suhel.homeip.net/tolmac/default.asp – Temperatura in vlaga na svetovnem spletu
[5]http://www.w3schools.com – Šola mrežnih tehnologij
[6]http://www.w3.org/ - Mrežni standardi
[7]http://www.sensirion.com/ - Proizvajalec senzorja SHT11
[8]Internetworking with TCP/IP Voulume 1 Principles, protocols, and architecture; Douglas E. Comer, Department of Computer Scients Purdue University, West Lafayette, IN 47907