

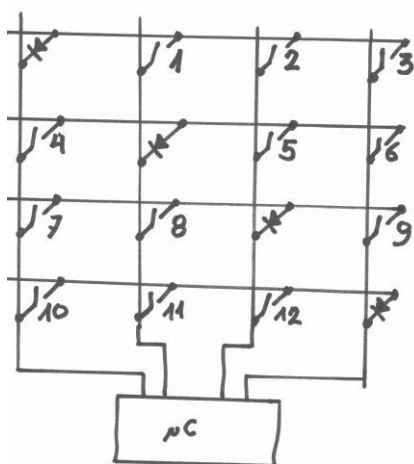
Odtipavanje večih tipk z manjšim številom priključkov mikrokontrolerja

Avtor: Vilko Sustič
E-pošta: vilko@sustic.com

Delam na nekem hobby projektu za sina. Nič posebnega: mikrokontroler, tipke, L298 in dva DC motorja. Naredil sem vezje, ATMEGA8, L298, MAX485, pretvornik napajanja iz 24VDC na 5VDC. Za upravljanje sem predvidel tipke in za vsako tipko po en pin mikrokontrolerja. Potem pa sin pride z idejo, da potrebuje 12 tipk. Zlomka! Sedaj to poveš! Tiskano vezje so Kitajci že naredili! Naj grem predelat tiskano vezje?

Pa sem se spomnil, da sem neke davno v Elektorju videl trik, kako z malo pini odtipavati več tipk. Elektorja seveda več nimam, s spomnim se, da sem trik nekako skopiral in spravil. Sedaj je čas, da ga najdem. In našel sem ga.

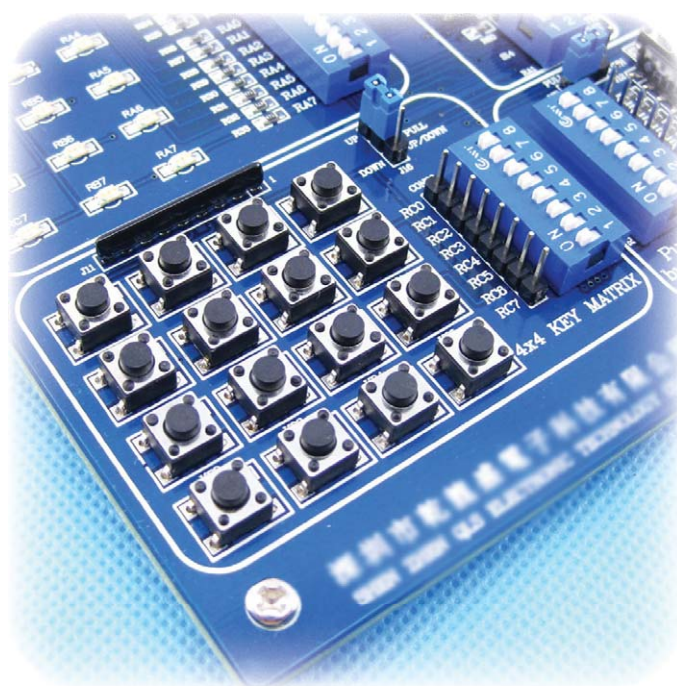
Gre za matrično orientirano tipkovnico, a ne tako običajno, ki jo dobite v trgovinah. Shema tipkovnice je kot sledi na sliki 1.



Slika 1: Shema tipkovnice

Kot vidite je matrika 4 x 4. Tipke so na presečiščih kolon in vrst, razen tam, kjer presečišče pripada diagonali. Matrika 4 x 4 ima 16 presečišč, če odštejemo 4 presečišča diagonale ostane 12 presečišč, kjer so tipke. Kjer presečišče pripada diagonali je namesto tipke dioda obrnjena od vrste k koloni.

Kako sedaj to odtipavati? Če krmilimo port za kolono 1 na izhod/output in damo gor signal 0V, ta signal preko diode, na presečišču 1-1 ki se odpre gre tudi na celo vrsto 1, ki je sedaj na 0V. Če ostale priključke mikrokontrolerja krmilimo za na vhod/input z pullup uporom in ni nobena tipka pritisnjena, bodo pač ti priključki na visokem nivoju +5V. Če pa je ena od tipk sklenjena, bo nivo 0V prešel iz vrste



preko sklenjene tipke na celo kolono in lahko na priključku ugotovimo, katera tipka te vrste je pritisnjena.

Postopek je seveda potrebno ponoviti za vse kolone/vrst, razen, če smo že ugotovili, da je ena od tipk že pritisnjena.

Zadeva ima seveda tudi slabe strani: Če je pritisnjenih več tipk, zaznamo samo eno od tipk. No in druga slaba stran: pri programiranju se je potrebno malo pomujati.

Ker kolikor toliko obvladam le Bascom, bom v tem programerskem jeziku pojasnil primer programiranja.

Najprej nekaj nujnih definicij:

```
' kolone so privezane na prikljucke recimo
' tako:
Kol1 Alias Pinb.3
Kol2 Alias Pinb.2
Kol3 Alias Pinb.5
Kol4 Alias Pinb.4
```

```
' vrste se krmilijo preko istih, toda
' izhodnih priključkov:
Vrstal Alias Portb.3
Vrsta2 Alias Portb.2
Vrsta3 Alias Portb.5
Vrsta4 Alias Portb.4
Dim Tipka As Byte ' številka pritisnjene
' tipke
```

V Bascom-AVR programsko postavljam izhode na Portx.y, kjer je X oznaka vrat in y oznaka priključka v teh vratih. Kadar pa želimo brati stanje na priključku, ga naslovimo z Pinx.y. Da je programiranje lažje, sem definiral odgovarjajoča Alias imena. Tako bom, če želim Vrsto1 dati na 0V napisal:

```
Vrstal = 0.
```

Kar pomeni isto kot:

```
Portb.3 = 0
```

No in sedaj je že čas, da gremo odtipavat tipke. Prvo vrsto tipk odtipamo tako:

```
Odtipavanje:
Config Vrstal = Output, _
: Config Kol2 = Input
Config Kol3 = Input : Config Kol4 = Input

Vrstal = 0 : Vrsta2 = 1 : Vrsta3 = 1, _
: Vrsta4 = 1

Tipka = 1 : Debounce Kol2 , 0 , Aha , Sub
Tipka = 2 : Debounce Kol3 , 0 , Aha , Sub
Tipka = 3 : Debounce Kol4 , 0 , Aha , Sub

' čakamo, da so vse tipke spuščene
Tipka = 0
Spustil1:
If Kol2 = 0 Or Kol3 = 0 Or Kol4 = 0 Then
Goto Spustil1
End If
```

Prvi dve vrsti gornje kode rekonfigurirata priključke, enega na output, ostale tri na input. Če je priključek definiran kot input in postavimo njegov port na 1, pomeni v Bascom-AVR to, da vključimo interni pullup, zunanji potem ni potreben. V vrstici kode:

```
Vrstal = 0 : Vrsta2 = 1 : Vrsta3 = 1, _
: Vrsta4 = 1
```

Naredimo prav to za vrstice/kolone 2,3 in 4, med tem ko vrstico/kolono 1 postavimo na 0V, saj je ta definirana kot izhod/output. Potem gremo gledat posledice:

```
Tipka = 1 : Debounce Kol2 , 0 , Aha , Sub
Tipka = 2 : Debounce Kol3 , 0 , Aha , Sub
Tipka = 3 : Debounce Kol4 , 0 , Aha , Sub
```



Bascom-AVR ima prav simpatično funkcijo Debounce, ki jo tu uporabimo za ugotavljanje, ali je tipka pritisnjena ali ne. Če je ukaz napisan tako, kot v tej kodi pomeni: Če je signal na koloni padel iz 1 na 0, kar v našem primeru pomeni pritisnjena tipka, in ko očistimo signal odskakovanja kontakta, tedaj skoči na podprogram z imenom Aha. Neposredno pred klicanjem funkcije Debounce smo spremenljivko Tipka postavili na vrednost tipke, ki jo ta Debounce odtipava. Če bo torej Debounce skočil na podprogram Aha, bo ta podprogram v spremenljivki Tipka dobil informacijo, za katero tipko gre.

Če funkcija Debounce ne zazna prehoda iz 1 na 0, ne naredi nič in gre program naprej na sledeči ukaz.

V kolikor je bila katerakoli od treh tipk pritisnjena, se je po obdelavi tipke v podprogramu Aha program vrnil in se ujel v sledeči kodi:

```
Spustil1:
If Kol2 = 0 Or Kol3 = 0 Or Kol4 = 0 Then
Goto Spustil1
End If
```

V zanki Goto Spustil1 se bo program vrtil vse, dokler tipka, ki smo jo pritisnili, ne bo tudi spuščena.

Sedaj smo testirali šele tipke 1 do 3. Kodo je potrebno malo predelano še trikrat ponoviti, da pretestiramo vse tipke do 12.



PROGRAMIRANJE

Podprogram Aha pa je zadolžen, da reagira na pritisk na tipko. On mora vedeti, kaj je potrebno ob pritisku na katero tipko narediti, vklopiti/izklopiti luč, izmeriti in poslati kako meritev, karkoli pač hočemo s temi tipkami krmiliti.

Pričujoči program je samo demonstracija, kako odtipavati tipke, in pričujoči Aha ne naredi nič drugega, kot da številko tipke na UART, ki je vezan na PC in si lahko potem ogledamo na PC-u (Terminal emulatorju) kako smo pritisnili tipke.

Testni program v celoti izgleda tako:

```
$regfile = »m8def.dat«
$crystal = 11059200
$baud = 9600
$hwstack = 48
$swstack = 48
$framesize = 48

Open »comb.1:9600,8,n,1« For Output As #2           'swuart
Print #2 , » »
Print #2 , » Reset.Keypad.bas«
Print #2 , » »

' Primer bascom odtipavanja 12 tipkovne
' tipkovnice z 4 portpini
'
' Tipke so razvrščene v vrste in kolone tako:
'   (D pomeni dioda, odprta smer od vrste v kolono)
'
'           D   1   2   3
'           4   D   5   6
'           7   8   D   9
'           #   0   *   D
Config Debounce = 40


' kolone so privezane na portpine recimo tako:
Kol1 Alias Pinb.3 : Kol2 Alias Pinb.2 : Kol3 Alias Pinb.5 : Kol4 Alias
Pinb.4           ' vrste se krmilijo preko istih,
'               ' toda izhodnih portpinov:
Vrstal Alias Portb.3 : Vrsta2 Alias Portb.2 :Vrsta3 Alias Portb.5 :Vrsta4 Alias Portb.4

Dim Tipka As Byte           ' vrednost pritisnjene tipke

Odtipavanje:
Config Vrstal = Output : Config Kol2 = Input
Config Kol3 = Input : Config Kol4 = Input
Vrstal = 0 : Vrsta2 = 1 : Vrsta3 = 1 : Vrsta4 = 1   ' Damo 1. vrsto na 0
Tipka = 1 : Debounce Kol2 , 0 , Aha , Sub         ' in gledamo, če se bo ničla
Tipka = 2 : Debounce Kol3 , 0 , Aha , Sub         ' pojavila na kateri koloni
Tipka = 3 : Debounce Kol4 , 0 , Aha , Sub         ' Če se, potem je bila ta tipka
' pritisnjena

' čakamo, da so vse tipke spuščene
Tipka = 0
Spustill:
' čakamo, da bo oseba popustila tipko
' prej ne smemo dati signal na sledečo vrsto
If Kol2 = 0 Or Kol3 = 0 Or Kol4 = 0 Then : Goto Spustill : End If

'.....
Config Vrsta2 = Output : Config Kol1 = Input
```



Lingva
Podskrajnik 35, 1380 Cerkljeva
www.lingva.si
info@lingva.si
t: 01 705 05 00
f: 01 70 50 505

The image shows a vertical banner for 'Lingva'. At the top is the company name 'Lingva' in a large, stylized font. Below it is the address 'Podskrajnik 35, 1380 Cerkljeva'. The main part of the banner features a vertical stack of four small images: a microcontroller chip, a laptop, a printed circuit board (PCB), and a keyboard. To the right of these images is the website address 'www.lingva.si' written vertically. At the bottom right, there is contact information: 'info@lingva.si', a telephone number 't: 01 705 05 00', and a fax number 'f: 01 70 50 505'.

```

Config Kol3 = Input : Config Kol4 = Input
Vrstal = 1 : Vrsta2 = 0 : Vrsta3 = 1 : Vrsta4 = 1           ' In enako 2. vrsto
Tipka = 4 : Debounce Kol1 , 0 , Aha , Sub
Tipka = 5 : Debounce Kol3 , 0 , Aha , Sub
Tipka = 6 : Debounce Kol4 , 0 , Aha , Sub
Tipka = 0
Spustil2:
If Kol1 = 0 Or Kol3 = 0 Or Kol4 = 0 Then : Goto Spustil2 : End If

.....

Config Vrsta3 = Output : Config Koll = Input
Config Kol2 = Input : Config Kol4 = Input
Vrstal = 1 : Vrsta2 = 1 : Vrsta3 = 0 : Vrsta4 = 1           ' in tako naprej
Tipka = 7 : Debounce Koll , 0 , Aha , Sub
Tipka = 8 : Debounce Kol2 , 0 , Aha , Sub
Tipka = 9 : Debounce Kol4 , 0 , Aha , Sub
Tipka = 0
Spustil3:
If Koll = 0 Or Kol2 = 0 Or Kol4 = 0then : Goto Spustil3 : End If
'.....

Config Vrsta4 = Output : Config Koll = Input
Config Kol2 = Input : Config Kol3 = Input
Vrstal = 1 : Vrsta2 = 1 : Vrsta3 = 1 : Vrsta4 = 0

Tipka = 10 : Debounce Koll , 0 , Aha , Sub
Tipka = 11 : Debounce Kol2 , 0 , Aha , Sub
Tipka = 12 : Debounce Kol3 , 0 , Aha , Sub
Tipka = 0
Spustil4:
If Koll = 0 Or Kol2 = 0 Or Kol3 = 0then : Goto Spustil4 : End If
'.....

Goto Odtipavanje                                           ' in gremo spet v odtipavanje

Aha:
' Aha, ena od tipk je pritisnjena njena vrednost je v bytu tipka
Print #2 , »Pritisnil tipko » ; Tipka
Return

Close #2
    
```

www.svet-el.si

RAZVOJNO ORODJE PROGRAM IZDELKI

Programator PROGGY AVR
MegaPin

1Wire
AT 90USB1287
JTAG razhroščevanje
LED signalizacija

Programirajmo z
BASCOM AVR

Priročnik za ljudi z domišljijo in željo za ustvarjanje programov v **BASCOM AVR**