

Vážení zákazníci,

dovolujeme si Vás upozornit, že na tuto ukázkou knihy se vztahují autorská práva, tzv. copyright.

To znamená, že ukáзка má sloužit výhradně pro osobní potřebu potenciálního kupujícího (aby čtenář viděl, jakým způsobem je titul zpracován a mohl se také podle tohoto, jako jednoho z parametrů, rozhodnout, zda titul koupí či ne).

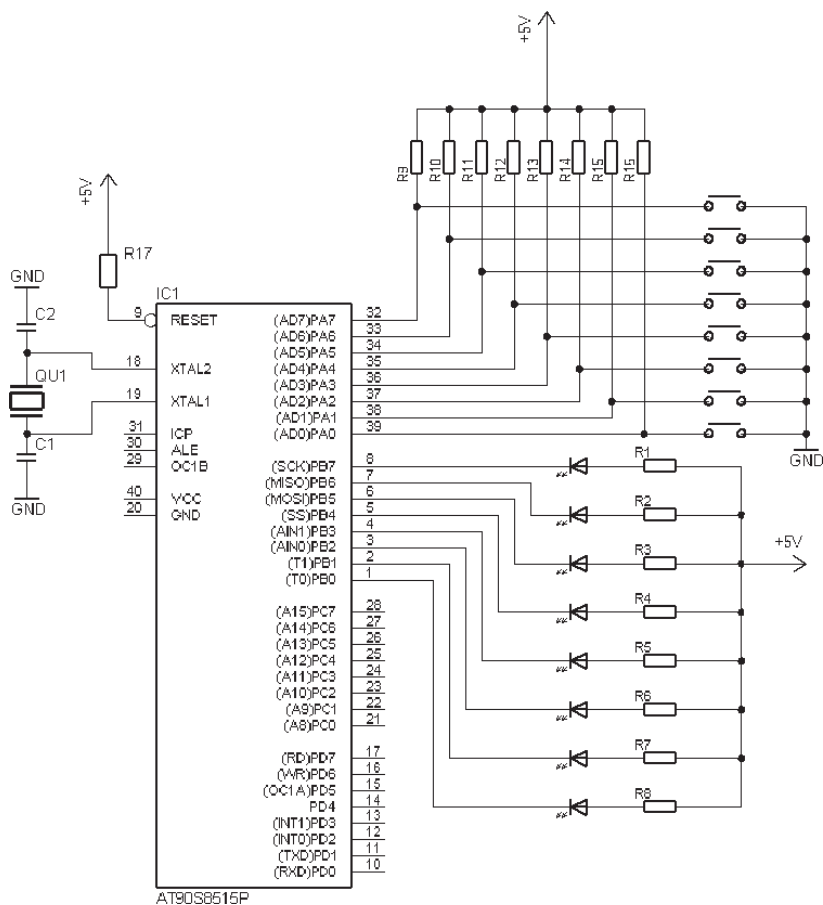
Z toho vyplývá, že není dovoleno tuto ukázkou jakýmkoliv způsobem dále šířit, veřejně či neveřejně např. umístováním na datová média, na jiné internetové stránky (ani prostřednictvím odkazů) apod.

redakce nakladatelství BEN – technická literatura
redakce@ben.cz



6.5 Pátý příklad – ovládání LED pomocí tlačítek

V předchozích programech jsme zatím programovali výstup na PORT, LCD displej či jsme výstupní data vysílali sériově UARTem. V dalších programech si ukážeme, jak naopak provádět vstup. Pro školní účely se obvykle používají různé startkity, jako např. STK500, u kterých vstupem budou tlačítka, výstupem diody LED. V dalším programu si ukážeme jak se dá v jazyce Bascom programově obsloužit takový jednoduchý startkit. Předpokládáme, že k PORTA máme připojena tlačítka tak, že v klidovém stavu jsou na PORTA samé jedničky, po stisknutí některého tlačítka se potom na příslušném pinu tohoto portu objeví nula. Jako výstup slouží LEDky připojené k PORTB tak, že je-li na některém z vývodů tohoto portu nula, tak LED svítí, je-li tam jednička, tak nesvítí. Principiální zapojení ukazuje obr. 6.9.



Obr. 6.9 Principiální zapojení pro ovládání LED tlačítky

Napišeme jednoduchý program, inspirovaný ukázkovým programem v assembleru u startkitu STK500. Jeho úkolem je v nekonečné smyčce zjišťovat stav tlačítek a na stisknutí některého z tlačítek nějak reagovat, např. inkrementací, dekrementací, rotací apod. údaje zobrazovaného diodami LED. Příklad takového programu je:

```
-----  
' priklad5.BAS  
' ukazka pristupu na porty AVR AT90S8535  
' pripojeni LEDEk na PORTB a tlacitek na PORTA  
-----  
$regfile = "8515def.dat"  
$crystal = 8000000  
  
Dim P As Byte  
Config Portb = Output  
Config Porta = Input  
P = &H00  
Do  
  
    Portb = P ' do PORTB posle obsah P  
    If Pina.0 = 0 Then  
        Incr P  
    End If  
  
    If Pina.1 = 0 Then  
        Decr P  
    End If  
  
    If Pina.2 = 0 Then  
        Shift P, Left  
    End If  
  
    If Pina.3 = 0 Then  
        Shift P, Right  
    End If  
  
    If Pina.4 = 0 Then  
        Rotate P, Left  
    End If  
  
    If Pina.5 = 0 Then  
        Toggle P  
    End If
```

```
If Pina.6 = 0 Then  
P = &HFF  
End If
```

```
If Pina.7 = 0 Then  
P = &H00  
End If
```

```
Waitms 200 ' prodleva 200 msec
```

```
Loop  
End
```

Činnost tohoto programu je velice primitivní. Nejprve se provede inicializace PORTA jako vstupního a PORTB jako výstupního. Dále se nastaví hodnota pomocné proměnné P na nulu. Potom se v nekonečné smyčce Do...Loop příkazem `Portb = P` provede zobrazení této pomocné proměnné pomocí LED diod. Hodnota této pomocné proměnné se mění, pokud je stisknuto některé z tlačítek. K tomu slouží příkazy jako např.

```
If Pina.3 = 0 Then ...
```

Kdy Pina.3 snímá logickou úroveň na pinu 3 portu PORTA a interpretuje ji jako nulu či jedničky. Při stisknutém tlačítku je tato hodnota nula. Podmínka v příkaze `If` je napsána tak, že k jejímu splnění dojde právě při stisknutém tlačítku a provádí se příkazy uvedené za `Then`.

Takto se provádí postupně test na stisknutí u všech tlačítek. Akce prováděné po stisknutí těchto tlačítek jsou inkrementace, dekrementace, rotace apod. obsahu pomocné proměnné P. Na konci těla nekonečné smyčky jsem ještě zařadil příkaz `Waitms` pro vytvoření časové prodlevy 200 ms jako jednoduchý způsob ošetření zátkmitu tlačítek.

6.6 Šestý příklad – připojení maticové klávesnice 4 × 4

V některých aplikacích potřebujeme ke komunikaci s běžící aplikací více než osm tlačítek použitých v předchozím příkladu. Přidání dalších tlačítek by sice bylo možné vyřešit jejich připojením k dalšímu portu mikrokontroléru a každým z tlačítek tak ovládat úroveň signálu pro jeden bit portu, ale zbytečně se tím ochuzujeme o porty umožňující komunikaci mikrokontroléru s okolím. Proto při použití více tlačítek je obvyklé jejich uspořádání do matice a připojení této tlačítkové matice k jedinému portu. Programové obslužení takové klávesnice je však složitější, než obsluha samotných tlačítek. Častým případem je použití 16 tlačítek zapojených do matice 4 × 4. Proto jazyk Bascom obsahuje funkci `Getkbd()` umožňující snadno naprogramovat obsluhu takové klávesnice. Příkladem může být kód:

```

-----
' file: priklad6.BAS
' pripojeni klavesnice 4 x 4 k PORTC
' pripojeni LCD k PORTA
'
'
' 1 PC0 -----0----1----2----3--
' | | | |
' 2 PC1 -----4----5----6----7--
' | | | |
' 3 PC2 -----8----9----10---11-
' | | | |
' 4 PC3 -----12---13---14---15-
' | | | |
' 5 PC4 -----
' | | | |
' 6 PC5 -----
' | | | |
' 7 PC6 ----- R1..R8= 470 ... 1k
' | | | |
' 8 PC7 -----
'
-----

```

```

$regfile = "8515def.dat"
$crystal = 8000000
$lib "lcd4busy.lib"

```

```

Const _lcdport = Porta
Const _lcdddr = Ddra
Const _lcdin = Pina
Const _lcd_e = 2
Const _lcd_rw = 1
Const _lcd_rs = 0

```

```

Dim A As Byte
Dim I As Byte
Config Lcd = 16 * 2 'nastaveni lcd
Config Kbd = Portc
Cls
Lcd "klavesnice 4 x 4"
Wait 1
Lowerline
Cursor Off
I = 0

```

```

Do
  A = Getkbd()
  If A <> 16 Then
    Incr I
    If I = 8 Then
      I = 0
      Locate 2, 1
      Lcd " "
      Locate 2, 1
    End If
    Lcd Hex(a)
  End If
  Waitms 20          ' prodleva 20 msec
Loop
End

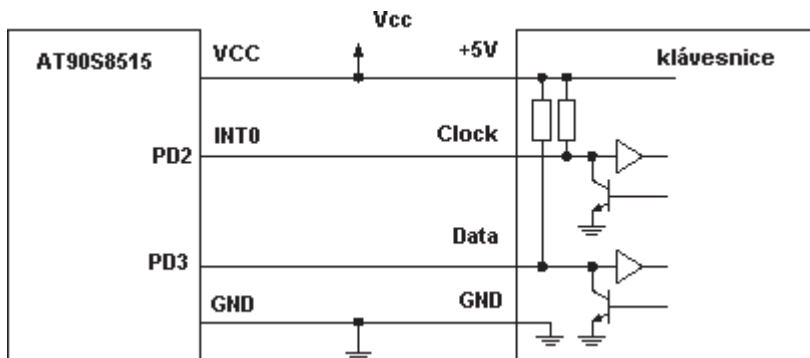
```

Jde opět o školní, ukázkový program. V komentáři na jeho začátku je uvedeno i připojení klávesnice k mikrokontroléru a kód kláves, vrácený po jejich stisknutí. Po počáteční inicializaci běží program v nekonečné smyčce. Je založen na použití příkazu `A = Getkbd()`, který při každém průchodu nekonečnou smyčkou do proměnné `A` posílá celočíselnou hodnotu v rozmezí 0, 1 ... 16. Je-li tato hodnota rovna 16, znamená to, že není stisknuta žádná klávesa. Je-li klávesa stisknuta, vrací tato funkce číselný kód této klávesy. Každá z 16 kláves má přiřazeno jednoznačně číslo v rozmezí 0, 1 ... 15. Toto číslo zobrazené jako hexaznak necháme zobrazovat na LCD displeji. Ve skutečné aplikaci se v závislosti na vrácené hodnotě bude provádět nějaká užitečná činnost, popř. se předtím ještě provede překódování vrácené hodnoty do nějakého jiného kódu.

6.7 Sedmý příklad – připojení klávesnice PC

Na předchozím programu jsme si ukázali princip připojení a programového obslužení maticové klávesnice. V případě, kdy potřebujeme klávesnici s větším počtem kláves může být výhodné použít běžnou klávesnici používanou u počítačů PC. Důsledkem jejího rozšíření je nízká cena a snadnost jejího připojení k mikropočítači či mikrokontroléru. Klávesnice PC AT tvoří samostatný konstrukční celek, obsahující obvykle 102 kláves a mikrokontrolér, který vykonává řadu funkcí. Tou hlavní je neustálé sledování stavu matice spínačů. V případě jakékoli změny zjistí mikrokontrolér souřadnice spínače, který ji způsobil, vybere v této situaci odpovídající SCAN kód, a ten po sériové lince DATA vyšle směrem k zařízení, s nímž klávesnice komunikuje. Je-li některá klávesa stisknuta déle než 0,5 s, vysílá její kód automaticky znovu. Kromě vlastních dat vysílá klávesnice při stisku klávesy i hodinový signál

CLK o kmitočtet 10 až 20 kHz. Při sestupné hraně hodinového signálu CLK jsou vysílána DATA platná. Toho můžeme využít při komunikaci klávesnice s nějakým systémem jako PC, mikrokontrolér AVR apod. Tuto komunikaci včetně obslužného programu napsaného v jazyce Bascom si ukážeme na komunikaci PC klávesnice s mikrokontrolérem ATMEL AT90S8515, který s klávesnicí spojíme podle obr. 6.10.



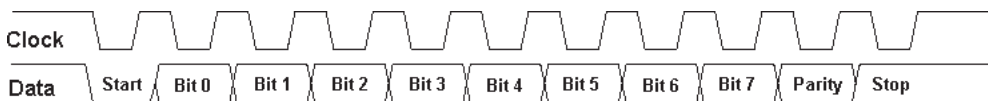
Obr. 6.10 Připojení klávesnice PC AT k mikrokontroléru Atmel AVR

Signál CLK z klávesnice přitom bude vyvolávat vnější přerušování mikrokontroléru AVR. Přerušování vyvolané sestupnou hranou hodinového signálu CLK bude obsluhováno funkcí, která bude snímat data vysílána klávesnicí. Klávesnice bude k mikrokontroléru připojena přes její konektor. Zapojení konektoru klávesnice PC AT popisuje tab. 6.1.

Tab. 6.1 Zapojení konektoru klávesnice PC AT

klávesnice PC AT		
Signál	DIN41524, zásuvka 5 pinů	Mini DIN PS2, zásuvka 6 pinů
Clock	1	5
Data	2	1
nezapojen	3	2, 6
GND	4	3
+5V	5	4
stínění	stínění	stínění

Klávesnice vysílá data sériově vždy po 11 bitech. Nejprve je vyslán startbit (logická nula), poté 8 datových bitů (první je LSB, poslední MSB), dále parita a nakonec stopbit (logická jednička) *obr. 6.11*.



Obr. 6.11 Časování klávesnice PC AT

Je zřejmé, že napsání programu pro obsluhu PC klávesnice není jednoduchou záležitostí. Proto součástí Bascomu je i funkce `GetATkbd()`, která za nás tuto práci udělá, takže napsat program využívající PC klávesnici může být jednoduché. Příkladem školního programu, který bude na LCD displeji zobrazovat znaky odpovídající stisknutým klávesám, je:

```

'-----
' file: prikklad7.BAS
' pripojeni PC AT klavesnice k PORTD PD2 .. CLK PD3 .. DATA
' pripojeni LCD k PORTA
'
'-----
$regfile = "8515def.dat"
$crystal = 8000000
$lib "lcd4busy.lib"

Const _lcdport = Porta
Const _lcdaddr = Ddra
Const _lcdin = Pina
Const _lcd_e = 2
Const _lcd_rw = 1
Const _lcd_rs = 0

Dim A As Byte
Dim I As Byte
Config Lcd = 16 * 2          'nastaveni lcd
Config Keyboard = Pind.2, Data = Pind.3, Keydata = Keydata

Cls
Lcd "PC klavesnice"
Wait 1
Lowerline

```



```

Cursor Off
I = 0
Do
  A = Getatkbd()

  If A > 0 Then
    Incr I
    If I = 16 Then
      I = 0
      Locate 2, 1
      Lcd " "
      Locate 2, 1
    End If
    Lcd Chr(a)
  End If

Loop
End

```

Keydata:

'normal keys lower case

Data 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, &H5E, 0

Data 0, 0, 0, 0, 0, 113, 49, 0, 0, 0, 122, 115, 97, 119, 50, 0

Data 0, 99, 120, 100, 101, 52, 51, 0, 0, 32, 118, 102, 116, 114, 53, 0

Data 0, 110, 98, 104, 103, 121, 54, 7, 8, 44, 109, 106, 117, 55, 56, 0

Data 0, 44, 107, 105, 111, 48, 57, 0, 0, 46, 45, 108, 48, 112, 43, 0

Data 0, 0, 0, 0, 0, 92, 0, 0, 0, 0, 13, 0, 0, 92, 0, 0

Data 0, 60, 0, 0, 0, 0, 8, 0, 0, 49, 0, 52, 55, 0, 0, 0

Data 48, 44, 50, 53, 54, 56, 0, 0, 0, 43, 51, 45, 42, 57, 0, 0

'shifted keys UPPER case

Data 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

Data 0, 0, 0, 0, 0, 81, 33, 0, 0, 0, 90, 83, 65, 87, 34, 0

Data 0, 67, 88, 68, 69, 0, 35, 0, 0, 32, 86, 70, 84, 82, 37, 0

Data 0, 78, 66, 72, 71, 89, 38, 0, 0, 76, 77, 74, 85, 47, 40, 0

Data 0, 59, 75, 73, 79, 61, 41, 0, 0, 58, 95, 76, 48, 80, 63, 0