

shiftin (spiin)

Platí pro PICAXE 20X2, 28X1, 28X2, 40X1, 40X2

Syntaxe:

SHIFTIN sclk, sdata, mode, (data {/ bits} {, data {/ bits}, ...})

SPIIN sclk, sdata, mode, (data {/ bits} {, data {/ bits}, ...})

Sclk – je proměnná nebo konstanta, která určuje I/O kontakt pro taktovací signál.

Sdata – je proměnná nebo konstanta která určuje I/O kontakt pro vstup dat.

Mode – je proměnná nebo konstanta (0 až 7), která určuje mód:

0	MSBPre_L	(MSB se načítá první, data před hodinami, klidová úroveň low)
1	LSBPre_L	(LSB se načítá první, data před hodinami, klidová úroveň low)
2	MSBPost_L	(MSB se načítá první, data po hodinách, klidová úroveň low)
3	LSBPost_L	(LSB se načítá první, data po hodinách, klidová úroveň low)
4	MSBPre_H	(MSB se načítá první, data před hodinami, klidová úroveň high)
5	LSBPre_H	(LSB se načítá první, data před hodinami, klidová úroveň high)
6	MSBPost_H	(MSB se načítá první, data po hodinách, klidová úroveň high)
7	LSBPost_H	(LSB se načítá první, data po hodinách, klidová úroveň high)

Data – je proměnná nebo konstanta, do níž se ukládají přijatá data.

Bits – (volitelně) je parametr, určující počet přenášených bitů. Není-li použit, je nastaven počet bitů na 8.

LSB = Least Significant Bit, je nejméně významný bit v binárním vyjádření čísla; v obvyklém dvojkovém zápisu jde o bit nejvíce vpravo (vzhledem k úmluvě psaní méně platných číslic vpravo). LSB může určovat lichost nebo sudost čísla.

MSB = Most Significant Bit je nejvýznamnější bit v binárním vyjádření čísla; v obvyklém dvojkovém zápisu jde o bit nejvíce vlevo. MSB má často zvláštní význam – například ve dvojkovém doplňku určuje znaménko čísla.

Popis:

Příkaz shiftin (alternativní název příkazu je spiin) je u PICAXE řady X1 a X2 určen k čistě programové synchronní sériové (SPI) komunikaci. U těchto PICAXE je ale zároveň možno používat i příkaz hspiout.

Ve výchozím nastavení je posouváno 8 bitů. Odlišný počet posouváných bitů (1 až 8) lze nastavit volitelným parametrem bits. Pokud tedy budete chtít vysunout 12 bitů, rozdělte je na dva bajty, prvním byte se přesune celý (8 bitů) a z druhého jen 4 bity.

Pokud se načítá nejprve MSB bit (mód 0, 2 a 4), bity jsou posouvány doleva, takže pokud chcete posunout jen 4 bity, musí tyto bity být v bajtu umístěny na pozicích 7 až 4 (nikoli 3 až 0). Pokud se načítá nejprve LSB bit (mód 1, 3, 5 a 7) jsou bity posouvány doprava, takže výše zmíněné 4 bity musí být v bajtu naopak umístěny na pozicích 0 až 3.

Při připojování SPI zařízení mějte na paměti, že pin data-in zařízení musíte spojit s pinem data-out PICAXE.

Příkaz shiftin (spiin) je určen jen pro PICAXE řady X1 a X2. U ostatních PICAXE ale můžete nahradit příkaz shiftin níže uvedenými podprogramy.

Efekt zvýšení taktovací frekvence

Zvýšení taktovací frekvence PICAXE zvýší také taktovací frekvenci SPI.

Související příkazy:

- shiftout
- hspissetup
- hspiin

Příklad 1:

Příkazem spiin se vysouvá nejprve LSB, data jsou vystavena před hodinovým pulzem
spiin 2,1,LSB_Pre_H, (b1 / 8) ; clock 8 bits into b1

Příklad 2:

```
; ~~~~~ SYMBOL DEFINITIONS ~~~~~  
; Required for all routines. Change pin numbers/bits as required.  
; Uses variables b7-b13 (i.e. b7,w4,w5,w6). If only using 8 bits  
; all the word variables can be safely changed to byte variables.  
  
;***** Sample symbol definitions *****  
symbol sclk = 5 ; taktování (výstup)  
symbol sdata = 7 ; data (výstupní pin pro data)  
symbol serdata = input7 ; data (vstupní pin pro data, note input7)  
symbol counter = b7 ; počítadlo smyčky  
symbol mask = w4 ; bit masking variable  
symbol var_in = w5 ; data variable used during shiftin  
symbol var_out = w6 ; data variable used during shiftout  
symbol bits = 8 ; number of bits  
symbol MSBvalue = 128 ; MSBvalue =128 for 8 bits, 512 for 10 bits, 2048  
for 12 bits)  
  
shiftin_LSB_Pre:  
let var_in = 0  
for counter = 1 to bits ; number of bits  
var_in = var_in / 2 ; shift right as LSB first  
if serdata <> 0 then  
var_in = var_in + MSBvalue ; set MSB if serdata = 1  
end if  
pulsout sclk,1 ; pulse clock to get next data bit  
next counter  
return
```

Příklad 3:

```
; ~~~~ SYMBOL DEFINITIONS ~~~~
; Required for all routines. Change pin numbers/bits as required.
; Uses variables b7-b13 (i.e. b7,w4,w5,w6). If only using 8 bits
; all the word variables can be safely changed to byte variables.

;***** Sample symbol definitions *****
symbol sclk = 5           ; clock (output pin)
symbol sdata = 7         ; data (output pin for shiftout)
symbol serdata = input7  ; data (input pin for shiftin, note input7)
symbol counter = b7      ; variable used during loop
symbol mask = w4         ; bit masking variable
symbol var_in = w5       ; data variable used durig shiftin
symbol var_out = w6      ; data variable used during shiftout
symbol bits = 8          ; number of bits
symbol MSBvalue = 128    ; MSBvalue =128 for 8 bits, 512 for 10 bits, 2048
for 12 bits)

shiftin_MSB_Pre:

let var_in = 0
for counter = 1 to bits ; number of bits
var_in = var_in * 2     ; shift left as MSB first
if serdata <> 0 then
var_in = var_in + 1    ; set LSB if serdata = 1
end if
pulsout sclk,1         ; pulse clock to get next data bit
next counter
return
```

Příklad 4:

```
; ~~~~ SYMBOL DEFINITIONS ~~~~
; Required for all routines. Change pin numbers/bits as required.
; Uses variables b7-b13 (i.e. b7,w4,w5,w6). If only using 8 bits
; all the word variables can be safely changed to byte variables.

;***** Sample symbol definitions *****
symbol sclk = 5           ; clock (output pin)
symbol sdata = 7         ; data (output pin for shiftout)
symbol serdata = input7  ; data (input pin for shiftin, note input7)
symbol counter = b7      ; variable used during loop
symbol mask = w4         ; bit masking variable
symbol var_in = w5       ; data variable used during shiftin
symbol var_out = w6      ; data variable used during shiftout
symbol bits = 8          ; number of bits
symbol MSBvalue = 128    ; MSBvalue =128 for 8 bits, 512 for 10 bits, 2048
for 12 bits)

shiftin_LSB_Post:

let var_in = 0
for counter = 1 to bits ; number of bits
var_in = var_in / 2     ; shift right as LSB first
pulsout sclk,1         ; pulse clock to get next data bit
if serdata <> 0 then
var_in = var_in + MSBvalue ; set MSB if serdata = 1
end if
next counter
return
```

Příklad 5:

```
; ~~~~ SYMBOL DEFINITIONS ~~~~
; Required for all routines. Change pin numbers/bits as required.
; Uses variables b7-b13 (i.e. b7,w4,w5,w6). If only using 8 bits
; all the word variables can be safely changed to byte variables.

;***** Sample symbol definitions *****
symbol sclk = 5           ; clock (output pin)
symbol sdata = 7         ; data (output pin for shiftout)
symbol serdata = input7  ; data (input pin for shiftin, note input7)
symbol counter = b7      ; variable used during loop
symbol mask = w4         ; bit masking variable
symbol var_in = w5       ; data variable used durig shiftin
symbol var_out = w6      ; data variable used during shiftout
symbol bits = 8          ; number of bits
symbol MSBvalue = 128    ; MSBvalue =128 for 8 bits, 512 for 10 bits, 2048
for 12 bits)

shiftin_MSB_Post:

let var_in = 0
for counter = 1 to bits ; number of bits
var_in = var_in * 2     ; shift left as MSB first
pulsout sclk,1         ; pulse clock to get next data bit
if serdata <> 0 then
var_in = var_in + 1    ; set LSB if serdata = 1
end if
next counter
return
```