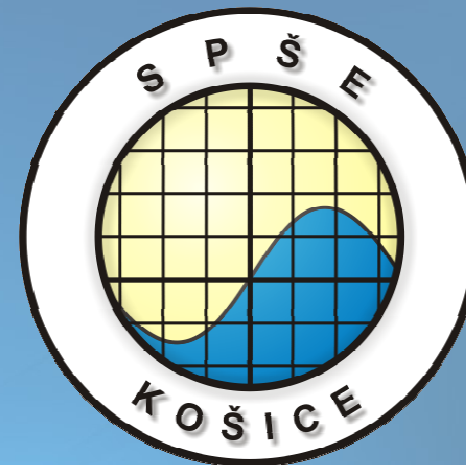




EURÓPSKA ÚNIA

Program  
celoživotného  
vzdelávania



# Programové prostredie mikrokontrolérov PIC

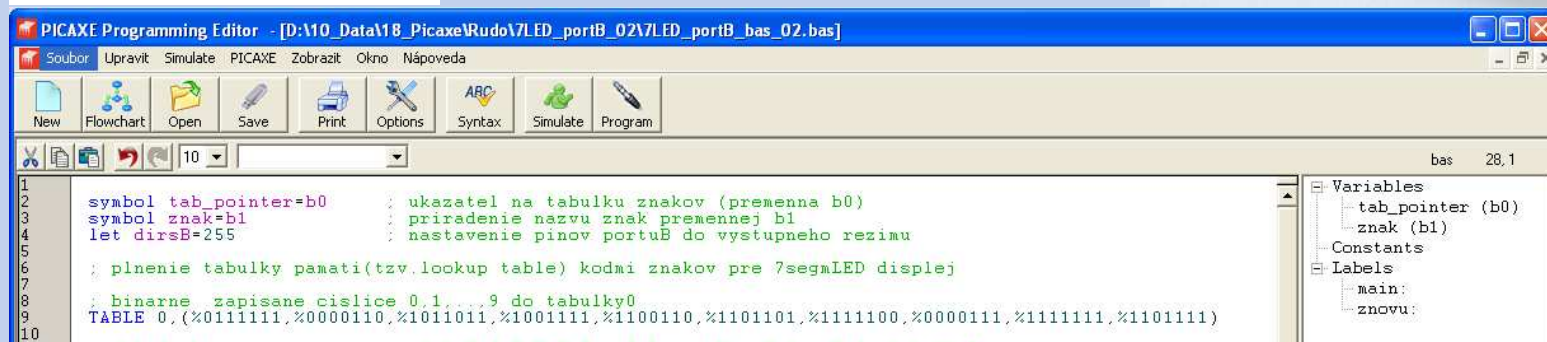
Vypracoval: Ing. Rudolf Sviantek



# Programové prostredie mikrokontrolérov PIC



- PICAXE-08M2
- PICAXE-14M2
- PICAXE-18M2
- PICAXE-20M2
- PICAXE-20X2
- PICAXE-28X2
- PICAXE-40X2



# Programové prostredie PICAXE

## Obsah

- Čo je to systém PICAXE
- Programový editor prostredia PICAXE
- Možnosti tvorby programu pre mikrokontroléry PIC
- Ukážky programu ( vývojový diagram, jazyk Basic)
- Prehľad príkazov jazyka BASIC
- Technické parametre PICAXE-20x2
- Obrazová príloha
- Zdroje informácií

# Čo je to systém PICAXE

- Programové prostredie PICAXE je voľne šíriteľný, otvorený program, ktorý na druhej strane poskytuje aj pomerne slušnú simuláciu a testovanie vyvíjaného programu bez potreby priameho fyzického pripojenia mikrokontroléra ku PC.
- PICAXE čipy sú štandardné PIC mikročipy firmy Microchip, ktoré obsahujú v sebe tzv. bootstrap program, ktorý umožňuje mikrokontroléru ho preprogramovať priamo cez sériovú linku z počítača, kde sa v príslušnom softvérovom prostredí napíše program konkrétnej aplikácie.
- PICAXE mikrokontroléry sú programované pomocou jazyka BASIC. Je však možné vytvárať program tiež graficky pomocou vývojových diagramov, zložených z grafických prvkov v podobe ikoniek. Grafická interpretácia programu sa potom preloží do príkazov jazyka BASIC.

# Programový editor prostredia

The screenshot displays the PICAXE Programming Editor interface. The main window contains a code editor with the following code:

```
1 symbol tab_pointer=b0      ; ukazatel na tabulku znakov (premena b0)
2 symbol znak=b1           ; priradenie nazvu znak premennej b1
3
4 let dirsB=255            ; nastavenie pinov portuB do vystupneho režimu
5
6 ; plnenie tabulky pamati(tzv.lookup table) kodmi znakov pre 7segmLED displej
7
8 ; binarne zapisane cislice 0,1,...9 do tabulky0
9 TABLE 0,(%0111111,%0000110,%1011011,%1001111,%1100110,%1111100,%0000111,%1111111,%1101111)
10
11 ; dekadicky zapisane pismena A,b,C,d,E,F,G,H,i,J,L,n,o,P,r,S,t,u,Y do tabulky1
12 TABLE 10,(119,124,57,94,121,113,125,118,4,31,56,84,92,115,80,109,120,28,114)
13
14 ; hexadecimalne hodnoty textu zapisane (119,118,92,31,124,56,124,121,57) "AHoJ bLbEC" do tabulky29
15 TABLE 29,($77,$76,$5C,$1F,$7C,$38,$7C,$79,$39)
16
17 main:
18 znovu:for tab_pointer = 0 to 9      ; start cyklu (zobrazenie cislic 0...9 nahor )
19     readtable tab_pointer,znak      ; citanie znaku z tabulky
20     pinsB=znak                     ; vyslanie znaku na portB
21     pause 1000                     ; oneskorenie 1 sec
22 next tab_pointer
23 wait 2                             ; pozdrzanie programu 2 sec
24
25 for tab_pointer = 10 to 28        ; start cyklu (zobrazenie pismen A,b,...Y )
26     readtable tab_pointer,znak
27     pinsB=znak
28     pause 1000
29 next tab_pointer
30
31
32 for tab_pointer=29 to 37          ; start cyklu
33     readtable tab_pointer,znak
34     pinsB=znak
35     pause 1000
36 next tab_pointer
37 wait 3
38
39 goto znovu
```

The simulation panel is open, showing the state of the PICAXE-20X2. It includes a pin configuration section with LEDs for pins A.0 through A.7 and B.0 through B.7. The ADC 0-3 section shows values for channels 0, 1, 2, and 3. The memory table shows the following data:

Address	Symbol	Value	Hex
0	tab_pointer	\$14	%00010100
1	znak	\$1F	%00011111
2	b2	\$00	%00000000
3	b3	\$00	%00000000
4	b4	\$00	%00000000
5	b5	\$00	%00000000
6	b6	\$00	%00000000
7	b7	\$00	%00000000
8	b8	\$00	%00000000
9	b9	\$00	%00000000
10	b10	\$00	%00000000
11	b11	\$00	%00000000
12	b12	\$00	%00000000
13	b13	\$00	%00000000
14	pinsA	0	%00000000
15	pinsB	0	%00000000
16	pinsC	0	%00000000
17	pinsD	0	%00000000
18	outpinsA	0	%00000000
19	outpinsB	31	%00011111

The variables list on the right shows:

- Variables: tab\_pointer (b0), znak (b1)
- Constants: none
- Labels: main, znovu

Programové pole

Použitie premenných, konštánt, návěstia

Vstupno-výstupný simulačný panel

# Programový editor prostredia

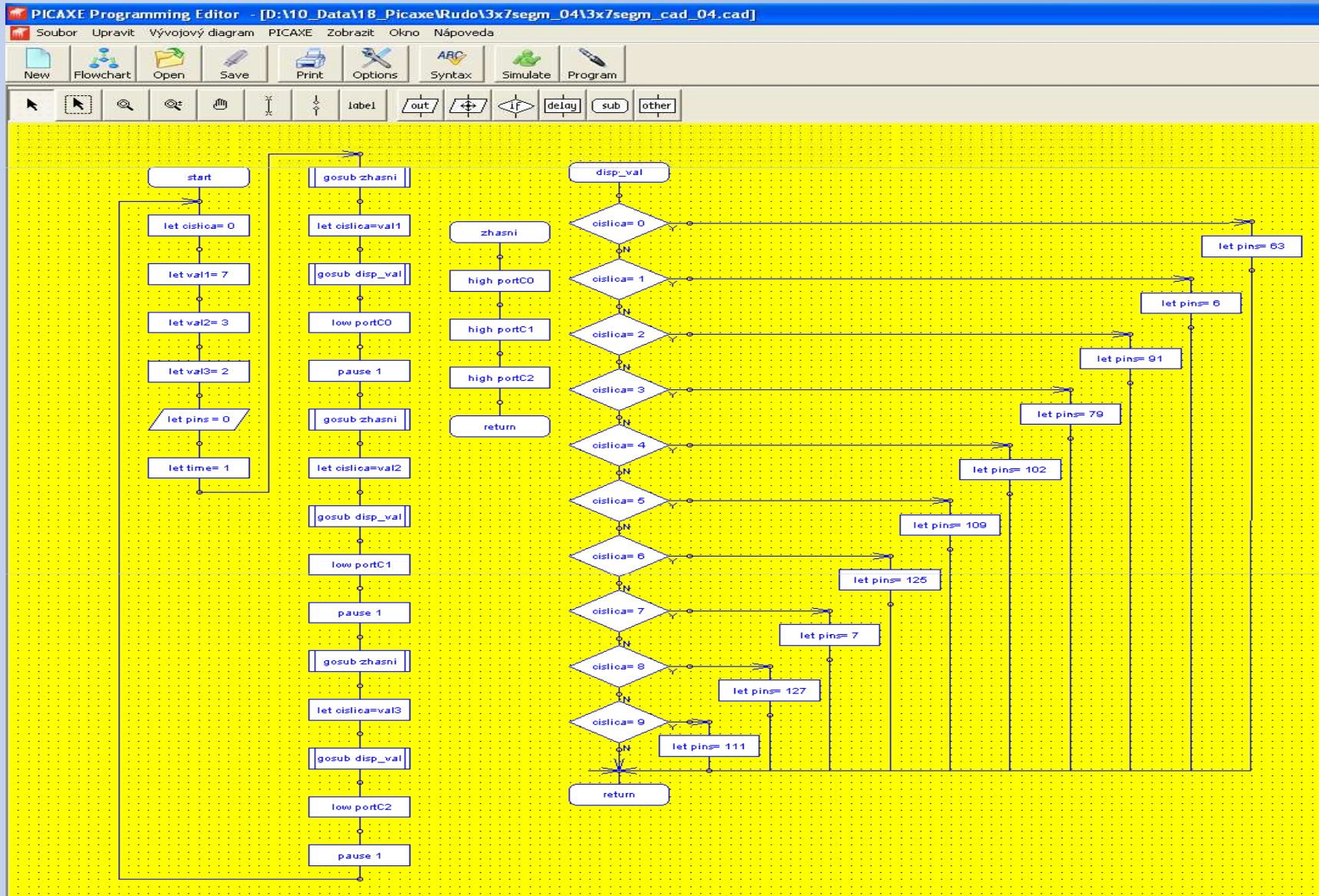
## Umožňuje tvorbu programu :

- priamo príkazmi jazyka Basic
- graficky pomocou značiek vývojového diagramu (Flowchart)
- graficky značkami logických číslicových obvodov (Logicator)

## Dovoľuje :

- Testovanie a simuláciu vytvoreného programu ( aj bez HW )
- Naprogramovanie mikrokontroléra cez sériový kábel (USB, RS-232) z prostredia programu v PC

# Programovanie vývojovým diagramom

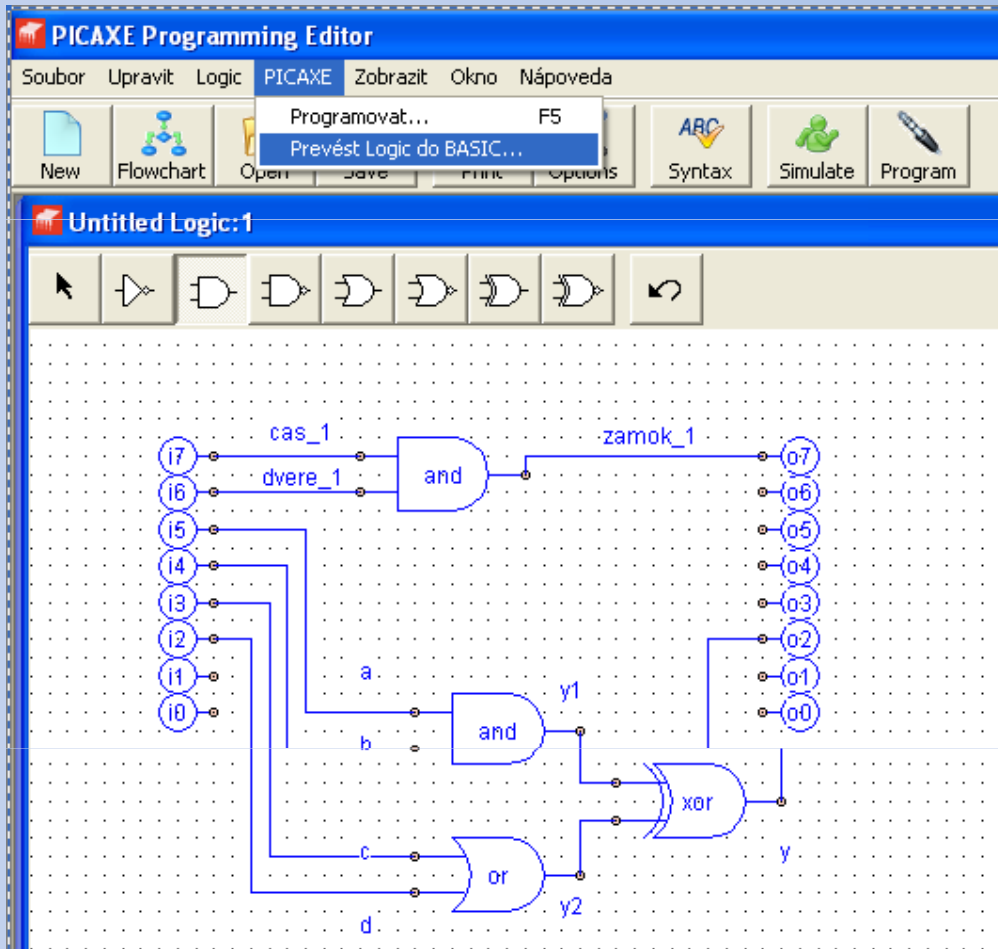


# Programovanie príkazmi jazyka Basic

```
1
2 symbol tab_pointer=b0      ; ukazatel na tabulku znakov (premenna b0)
3 symbol znak=b1            ; priradenie nazvu znak premennej b1
4 let dirsB=255            ; nastavenie pinov portuB do vystupneho rezimu
5
6 ; plnenie tabulky pamati(tzv.lookup table) kodmi znakov pre 7segmLED displej
7
8 ; binarne zapisane cislice 0,1,...,9 do tabulky0
9 TABLE 0,(%0111111,%0000110,%1011011,%1001111,%1100110,%1101101,%1111100,%0000111,%1111111,%1101111)
10
11 ; dekadicky zapisane pismena A,b,C,d,E,F,G,H,i,J,L,n,o,P,r,S,t,u,Y do tabulky1
12 TABLE 10,(119,124,57,94,121,113,125,118,4,31,56,84,92,115,80,109,120,28,114)
13
14 ; hexadecimalne hodnoty textu zapisane (119,118,92,31,124,56,124,121,57) "AhoJ bLbEC" do tabulky29
15 TABLE 29,($77,$76,$5C,$1F,$7C,$38,$7C,$79,$39)
16
17 main:
18 znovu:for tab_pointer = 0 to 9      ; start cyklu (zobrazenie cislic 0...9 nahor )
19     readtable tab_pointer,znak      ; citanie znaku z tabulky
20     pinsB=znak                     ; vyslanie znaku na portB
21     pause 1000                     ; oneskorenie 1 sec
22 next tab_pointer
23 wait 2                             ; pozdrzanie programu 2 sec
24
25     for tab_pointer = 10 to 28     ; start cyklu (zobrazenie pismen A,b,...Y )
26         readtable tab_pointer,znak
27         pinsB=znak
28         pause 1000
29     next tab_pointer
30     wait 2
31
32     for tab_pointer=29 to 37       ; start cyklu (zobrazenie textu "Ahoj blbec" )
33         readtable tab_pointer,znak
34         pinsB=znak
35         pause 1000
36     next tab_pointer
37     wait 3
38
39     goto znovu
```



# Programovanie pomocou logikátoru



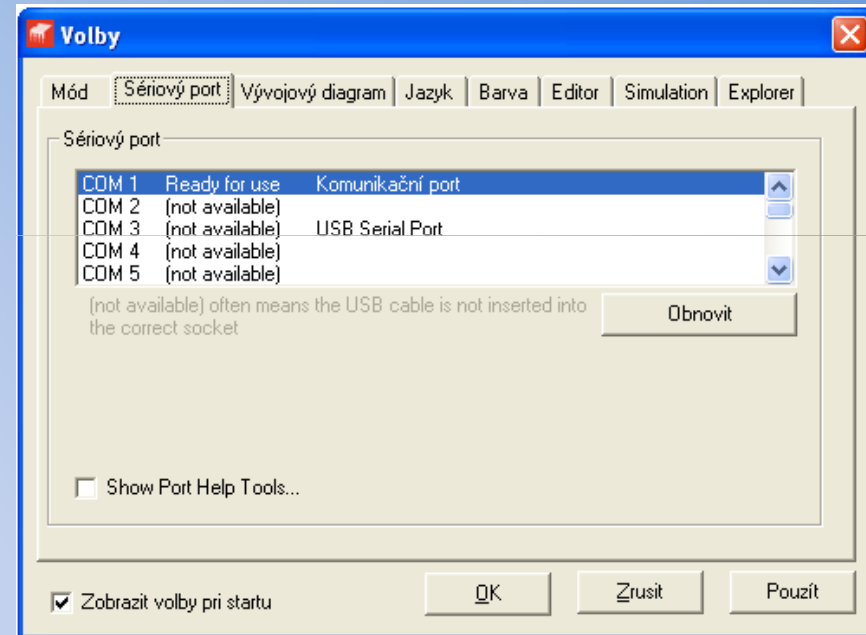
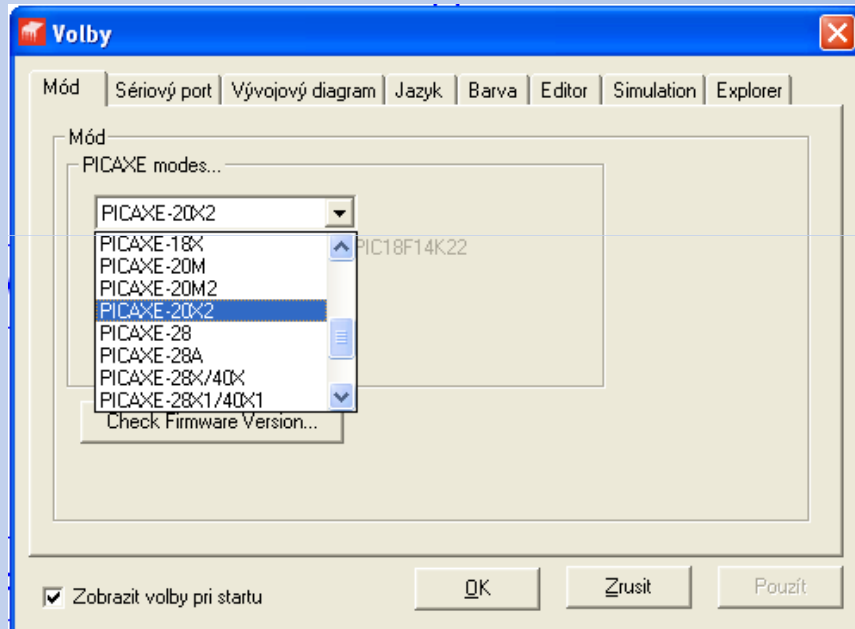
Grafickými značkami

Preklad do Basicu

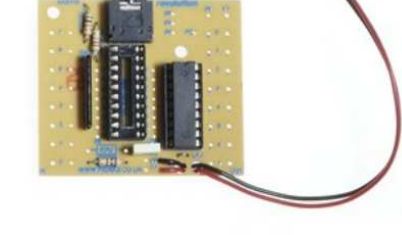
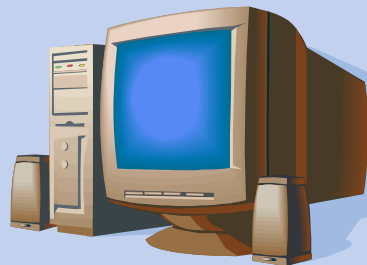
The screenshot shows a text editor window titled 'Untitled:2' containing the BASIC code generated from the logic circuit. The code is as follows:

```
1 'BASIC converted from Logic Circuit:
2 'Untitled Logic:1
3 'Converted on 17.4.2012 at 17:04:17
4
5 symbol pos0 = bit0
6 symbol pos1 = bit1
7 symbol pos2 = bit2
8 symbol pos3 = bit3
9 symbol pos4 = bit4
10 symbol pos5 = bit5
11 symbol pos6 = bit6
12 symbol pos7 = bit7
13 symbol pos8 = bit8
14 symbol pos9 = bit9
15 symbol posA = bit10
16 symbol posB = bit11
17 symbol posC = bit12
18 symbol posD = bit13
19 symbol posE = bit14
20 symbol posF = bit15
21
22 main:
23
24   pinB.7 = input7 and input6
25   pos1 = input5 and input4
26   pos2 = input3 or input2
27   pinB.2 = pos1 xor pos2
28
29   goto main
30
```

# Parametre nastavenia prog. prostredia

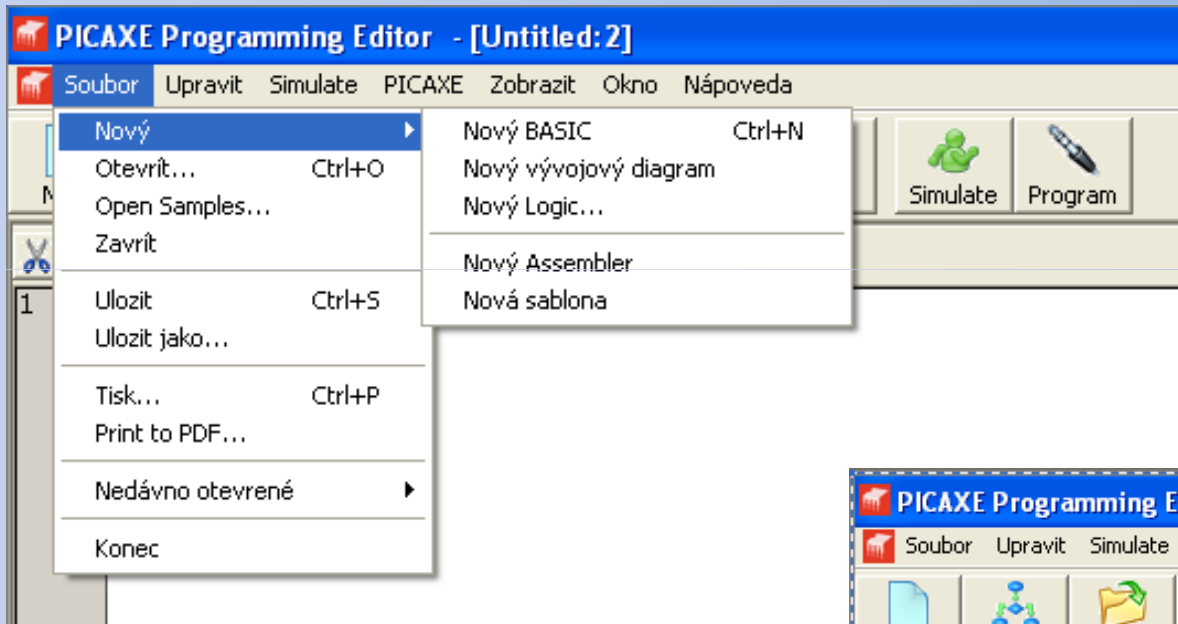


Voľba typu  
obvodu PICAXE



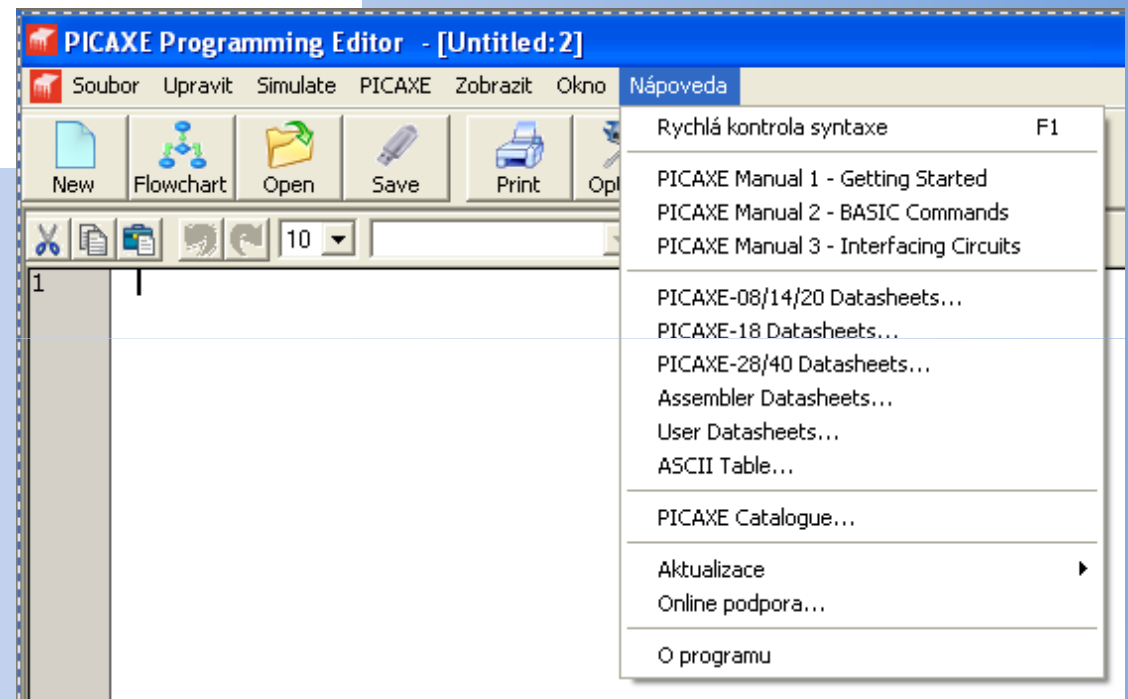
Voľba  
komunikačného  
portu

# Parametre nastavenia prog. prostredia

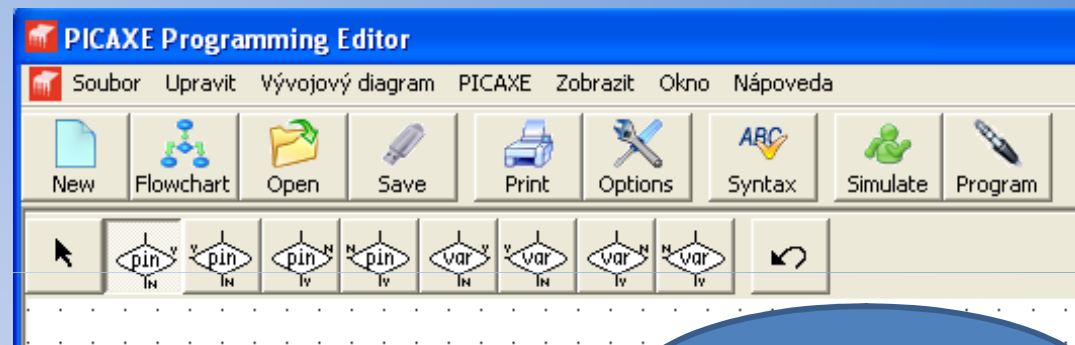


Voľba spôsobu programovania

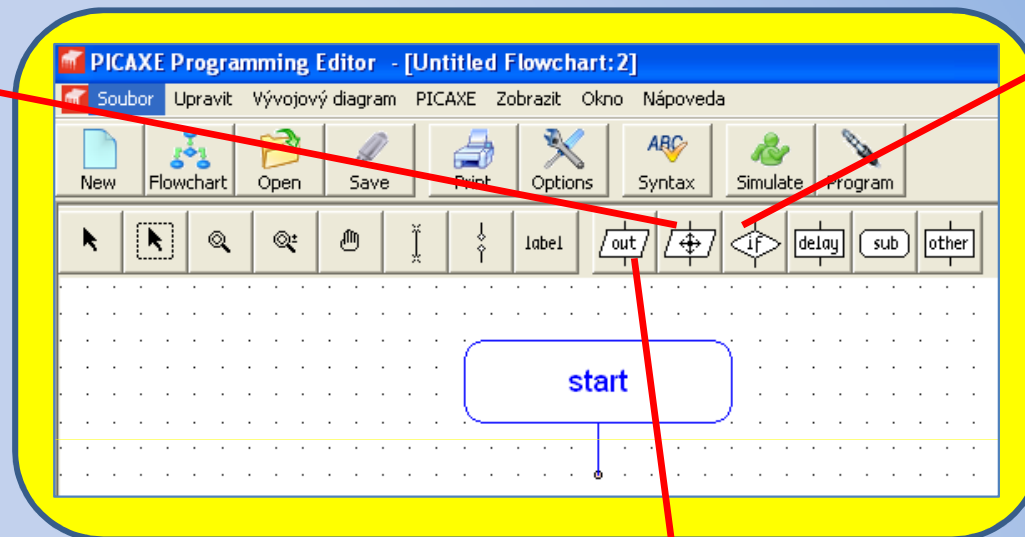
HELP manual  
podpora on-line



# Ponuky v režime Flowchart

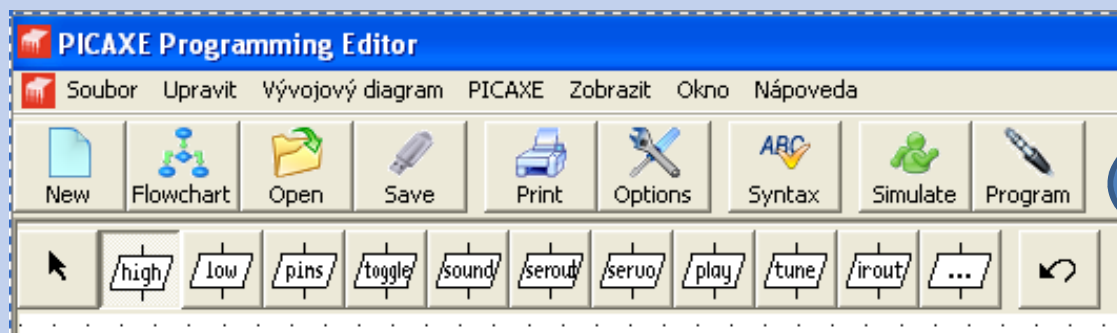


Riadenie pohybu

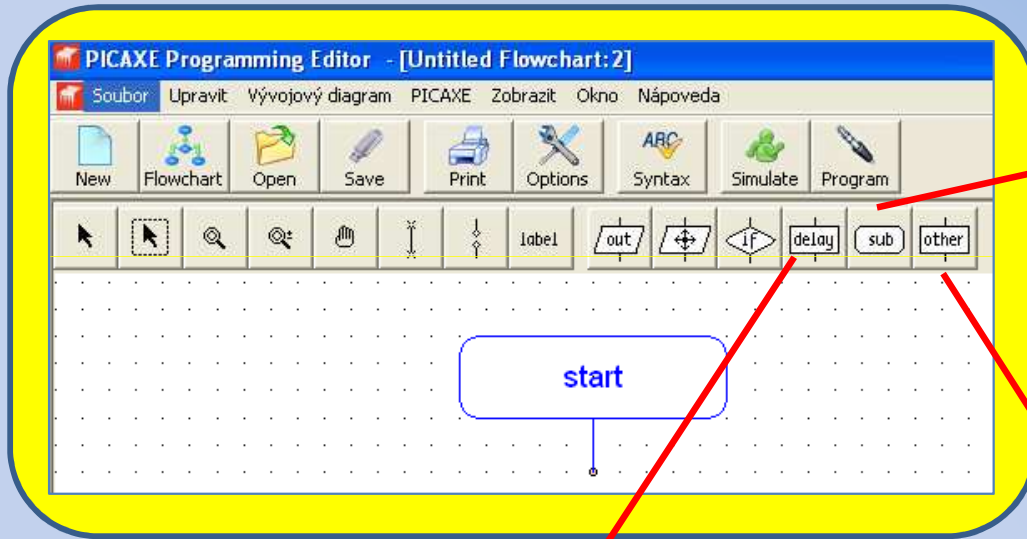


Podmienené vetvenie

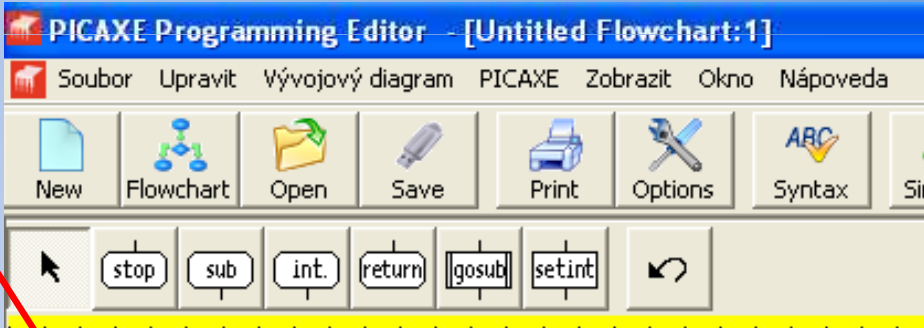
Výstupný režim



# Ponuky v režime Flowchart

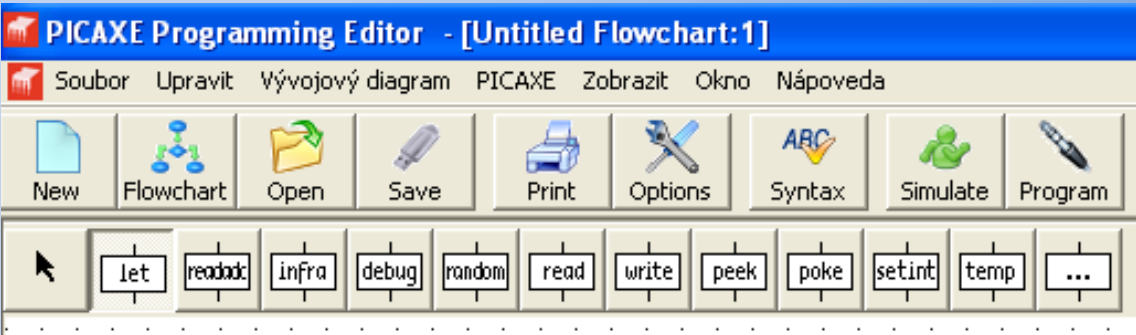


Podprogramy,  
prerušená

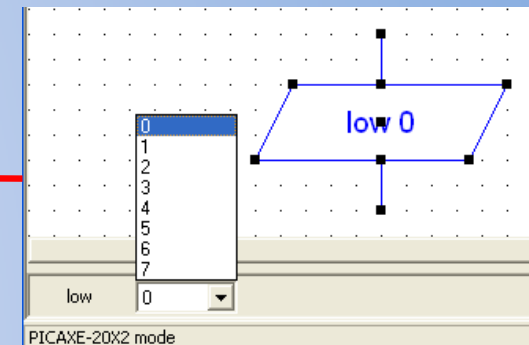
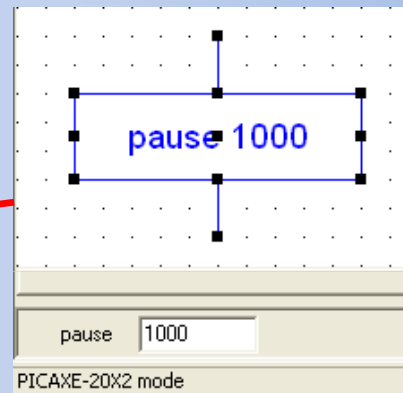
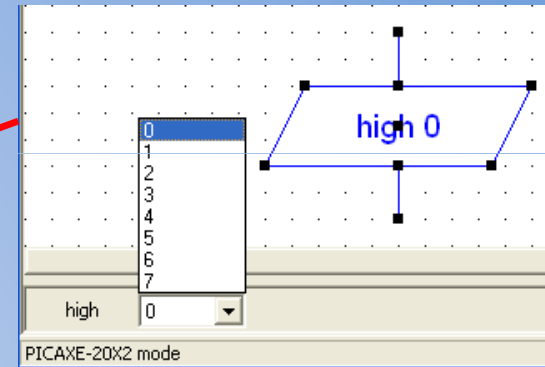
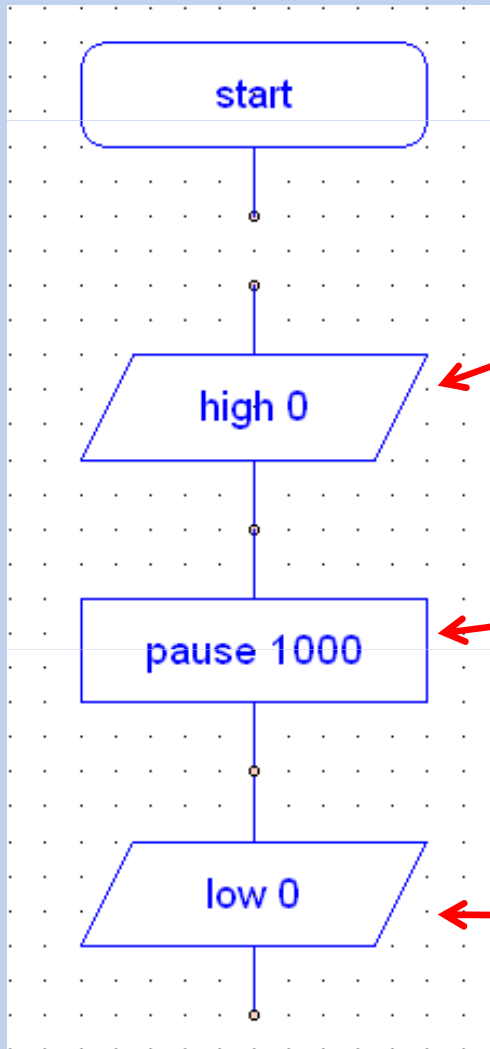


Nastavenie  
času

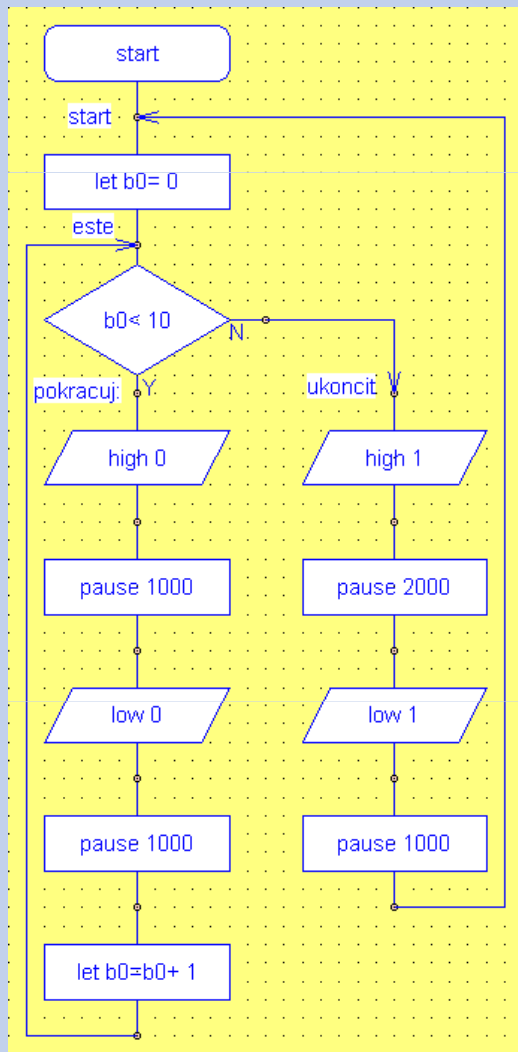
Ostatné príkazy  
Basicu



# Nastavenie parametrov značiek

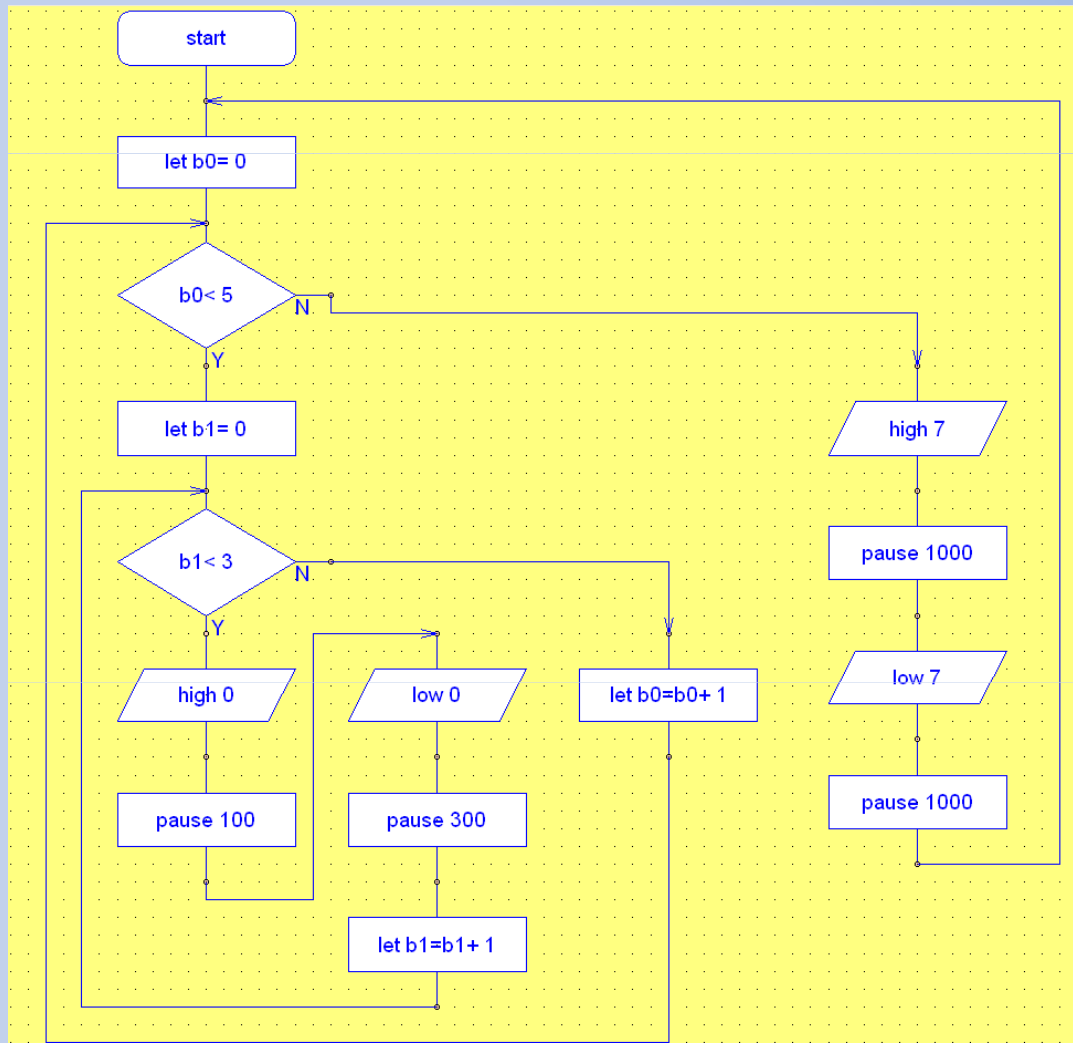


# Ukážky programu (graficky a kódom)



```
PICAXE Programming Editor - [D:\10_Data\18_Picaxe\Rudo\forcyklus_bas_01.bas]
Soubor: Upravit Simulate PICAXE Zobrazit Okno Nápoveda
New Flowchart Open Save Print Options Syntax Simulate Program
10
;BASIC converted from flowchart:
;forcyklus_01.cad
4
symbol pocet= b0
symbol Green_LED= 0
symbol Red_LED= 1
let dirsB = 255
; priradenie nazvu premennej b0
; priradenie nazvu konstante (vystupu 0)
; priradenie nazvu konstante (vystupu 1)
; nastavenie celeho portuB do rezimu vystupu
8
main:
9
start: let pocet= 0
10
este: if pocet < 10 then pokracuj
11
; inicializacia premennej
; testovanie podmienky na zaciatku cyklu
12
ukoncit: high Red_LED
13
; ak nesplnena podmienka, tak ...
14
pause 2000
15
low Red_LED
16
pause 1000
17
goto start
18
; skok na navestie
19
pokracuj: high Green_LED
20
; ak splnena podmienka, tak ...
21
pause 1000
22
low Green_LED
23
pause 1000
24
let pocet=pocet + 1
goto este
; skok na navestie
```

# Ukážky programu (graficky a kódem)



```
    let dirsB = 255
main:
label_10:  let b0= 0
label_1B:  if b0< 5 then label_AE
           high 7
           pause 1000
           low 7
           pause 1000
           goto label_10

label_AE:  let b1= 0
label_B9:  if b1< 3 then label_22
           let b0=b0+ 1
           goto label_1B

label_22:  high 0
           pause 100
           low 0
           pause 300
           let b1=b1+ 1
           goto label_B9
```



# Prehľad príkazov jazyka Basic

adcconfig .....  
adcsetup .....  
backward .....  
bcdtoascii .....  
bintoascii .....  
booti2c .....  
branch .....  
button .....  
calibadc (calibadc10) .....  
calibfreq .....  
clearbit .....  
compsetup .....  
count .....  
daclevel .....  
dacsetup .....  
debug .....  
dec .....  
disablebod .....  
disabletime .....  
disconnect .....  
do...loop .....  
doze .....  
eeprom (data) .....  
enablebod .....  
enabletime .....  
end .....  
exit .....

for...next .....  
forward .....  
fvrsetup .....  
get .....  
gosub (call) .....  
goto .....  
hi2cin .....  
hi2cout .....  
hi2csetup .....  
hi2csetup - slave mode (X2) .....  
hi2csetup - master mode .....  
halt .....  
hibernate .....  
high .....  
high portc .....  
hintsetup .....  
hpwm .....  
hpwmduty .....  
hserin .....  
hserout .....  
hsersetup .....  
hspiin (hshin) .....  
hspiout (hshout) .....  
hspisetaup .....  
i2cslave .....  
if...then \ elseif...then \ else \ endif .....  
if...then {goto} .....

if...and/or..then {goto} .....  
if porta...then {goto} .....  
if portc...then {goto} .....  
if...then exit .....  
if...and/or...then exit .....  
if...then gosub .....  
if...and/or...then gosub .....  
inc .....  
infrain .....  
infrain2 .....  
infraout .....  
input .....  
inputtype .....  
irin .....  
irout .....  
kbin .....  
keyin .....  
kbled (keyled) .....  
let .....  
let dirs / dirsc = .....  
let dirsA / dirsB / dirsC / dirsD = .....  
let pins / pinsc = .....  
let pinsA / pinsB / pinsC / pinsD = ..  
lookdown .....  
lookup .....  
low .....  
low portc .....

nap .....  
on...goto .....  
on...gosub .....  
output .....  
owin .....  
owout .....  
pause .....  
pauseus .....  
peek .....  
peeksfr .....  
play .....  
poke .....  
pokesfr .....  
pullup .....  
pulsin .....  
pulsout .....  
put .....  
pwm .....  
pwmduty .....  
pwmout .....  
random .....  
read .....  
readadc .....  
readadc10 .....  
readdac .....  
readdac10 .....  
readi2c .....  
readinternaltemp .....

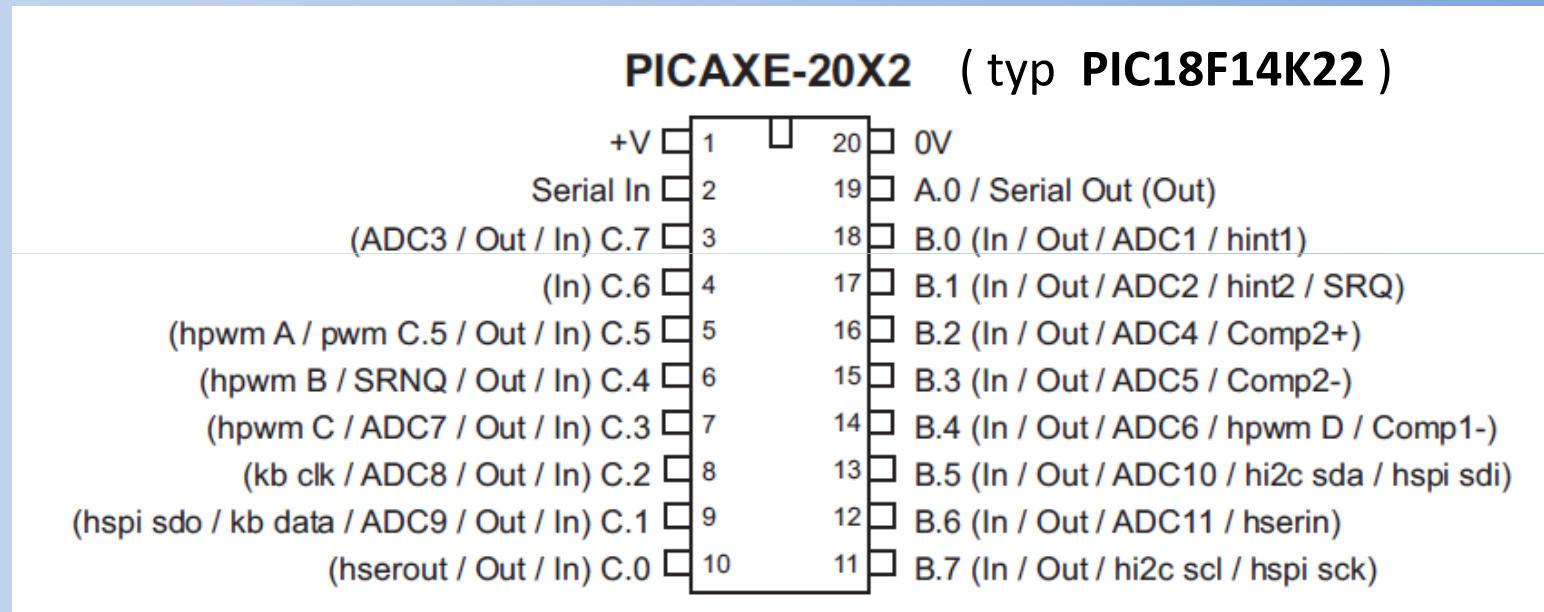
# Prehľad príkazov jazyka Basic

readfirmware .....  
readmem .....  
readtable .....  
readoutputs .....  
readportc .....  
readrevision .....  
readsilicon .....  
readtemp .....  
readtemp12 .....  
readowclk .....  
resetowclk .....  
readownsn .....  
reconnect .....  
reset .....  
restart .....  
resume .....  
return .....  
reverse .....  
rfin .....  
rfout .....  
run .....

select case \ case \ else \ endselect .....  
serin .....  
serrxd .....  
serout .....  
sertxd .....  
servo .....  
servopos .....  
setbit .....  
setint .....  
setintflags .....  
setfreq .....  
settimer .....

shiftin (spiin) .....  
shiftout (spiout) .....  
sleep .....  
sound .....  
srlatch .....  
srset / srreset .....  
stop .....  
suspend .....  
swap .....  
switch on/off .....  
symbol .....  
table .....  
tablecopy .....  
tmr3setup .....  
toggle .....  
togglebit .....  
touch .....  
touch16 .....  
tune .....  
uniin .....  
uniout .....  
wait .....  
write .....  
writemem .....  
writei2c .....

# Technické parametre PICAXE-20X2



- Pamäť programu : 16384 bajtov
- Pamäť dát (SRAM): 512 bajtov
- Pamäť dát (EEPROM) : 256 bajtov
- Napájacie napätie : 2,3 ... 5,5 V
- Počet I/O pinov : 18
- Počet ADC pinov: 11 (8-bit, 10-bit)

# Technické parametre PICAXE-20X2

- I/O prúd : max 25 mA v oboch stavoch
- priorita operátorov neexistuje, vyhodnotenie výrazov zľava doprava
- všetky inštrukcie sú 16-bitové, jednoslovné (word)
- 4 typy premenných v pamäti RAM :
  - všeobecné    b0 ... b55 (56 bajtov)                      56 premenných  
                  w0 ... w28 ( w0= b0:b1, w1=b2:b3, ... )                28 premenných  
                  bit31:bit30: .... bit1:bit0 (bajty b3, b2, b1, b0)        32 premenných
  - storage        (56 ... 127 → 38h ... 7Fh)                      72 premenných
  - scratchpad    pamäťové pole (0 ... 127)                                128 bajtov
  - systémové (špeciálne) premenné

# Technické parametre PICAXE-20X2

- pamäť typu flash preprogramovateľná : až 100 000x
- pamäť programu môže obsahovať : 4 samostatné programy
- dĺžka jedného programu: 1000 riadkov
- frekvencia vnút. oscilátora nastaviteľná : 4, 8, 16, 32, 64 MHz
- defaultná hodnota frekvencie interného rezonátora po zapnutí napájania : 8 MHz
- príkazy pauzy kalibrované na frekvenciu : 8 MHz
- defaultná prenosová rýchlosť sériového kanálu: 9600,n,8,1

# Technické parametre rady PICAXE

Feature	08M2	14M2	18M2	20M2	20X2	28X2	40X2
Memory Capacity (bytes)	2048	2048	2048	2048	4096	4096 x4	4096 x4
RAM (bytes)	128	512	512	512	256	1280	1280
Byte Variables (bytes)	28	28	28	28	56	56	56
Input/Outputs Pins	6	12	16	18	18	22	33
ADC/Touch Pins	3	7	10	11	11	16	27
Max. Freq. (MHz)	32	32	32	32	64	64	64
Serial In/Out	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Infrared In/Out	Yes	Yes	Yes	Yes	Yes	Yes	Yes
I2C	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Tune (ring tones)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Parallel Tasks	4	8	8	8	1	1	1
Program Slots	1	1	1	1	1	4	4

# Domovská stránka PICAXE systému



The screenshot shows the homepage of the PICAXE website. At the top, there is a navigation bar with the PICAXE logo on the left, a search bar, and links for 'Free Software' and 'PICAXE Shop'. Below this is a red navigation menu with links for 'What is PICAXE?', 'Getting Started', 'Hardware', 'Software', 'Manuals', 'Commands', 'Circuit Creator', 'Project Gallery', 'Forums', 'Teaching', 'Distributors', and 'Support'. The main content area features a breadcrumb trail 'Home > What is PICAXE >', a large heading 'What is PICAXE?' with the subtitle 'An introduction to the PICAXE system', and the URL 'www.picaxe.com'. A sidebar on the left contains a list of links: 'What is PICAXE?', 'PICAXE Chip Sizes', 'PICAXE Pinouts', 'PICAXE Chip Labels', and 'Superseded Parts'. The main text area includes a sub-heading 'What Is PICAXE?' followed by a paragraph describing the PICAXE microcontroller as the 'brain of your electronic project'. To the right of this text is an image of a PICAXE microcontroller board connected to a laptop. Below the image is a small logo for 'ohmat-009'. Further down, there are sections for 'PICAXE chips are popular because they are very low-cost, and simple to program using free, easy-to-learn software.', 'The PICAXE chip can react to input sensors and switch outputs on and off accordingly.', and 'The various different PICAXE chip sizes (8, 14, 18, 20, 28 and 40 pins) give great flexibility on how the system can be used - simply select the chip size as required for your project. On PICAXE parts almost all the pins can be user configured to be an output, a digital (on/off) input, an analogue input or a touch sensor. Advanced features such as PWM, I2C, SPI, RS232 and 1-Wire interfacing can also be easily achieved if required.' The page ends with a section for 'Programming Software'.

www.picaxe.com

## What is PICAXE?

An introduction to the PICAXE system

### What Is PICAXE?

A PICAXE microcontroller is designed to be the **brain of your electronic project**. Originally designed as an educational system for schools, the PICAXE system has now also been widely adopted by hundreds of thousands of 'hobbyists' due to its ease of use. Each year thousands of high school students are also introduced to electronics and microcontrollers via building a PICAXE project.



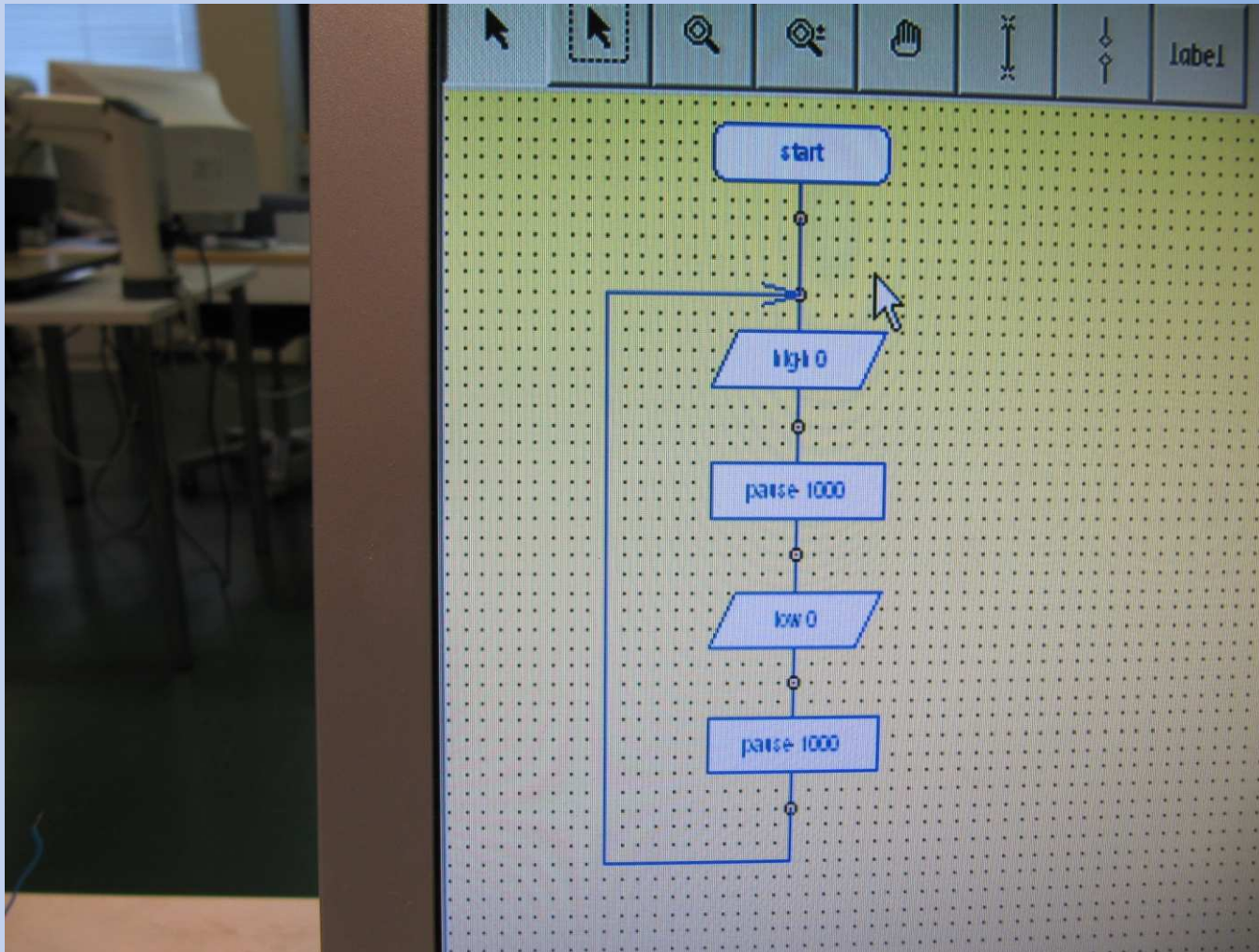
PICAXE chips are popular because they are very **low-cost**, and **simple to program** using free, easy-to-learn software.

The PICAXE chip can react to input sensors and switch outputs on and off accordingly.

The various different PICAXE **chip sizes** (8, 14, 18, 20, 28 and 40 pins) give great flexibility on how the system can be used - simply select the chip size as required for your project. On PICAXE parts almost all the pins can be user configured to be an output, a digital (on/off) input, an analogue input or a touch sensor. Advanced features such as PWM, I2C, SPI, RS232 and 1-Wire interfacing can also be easily achieved if required.

### Programming Software

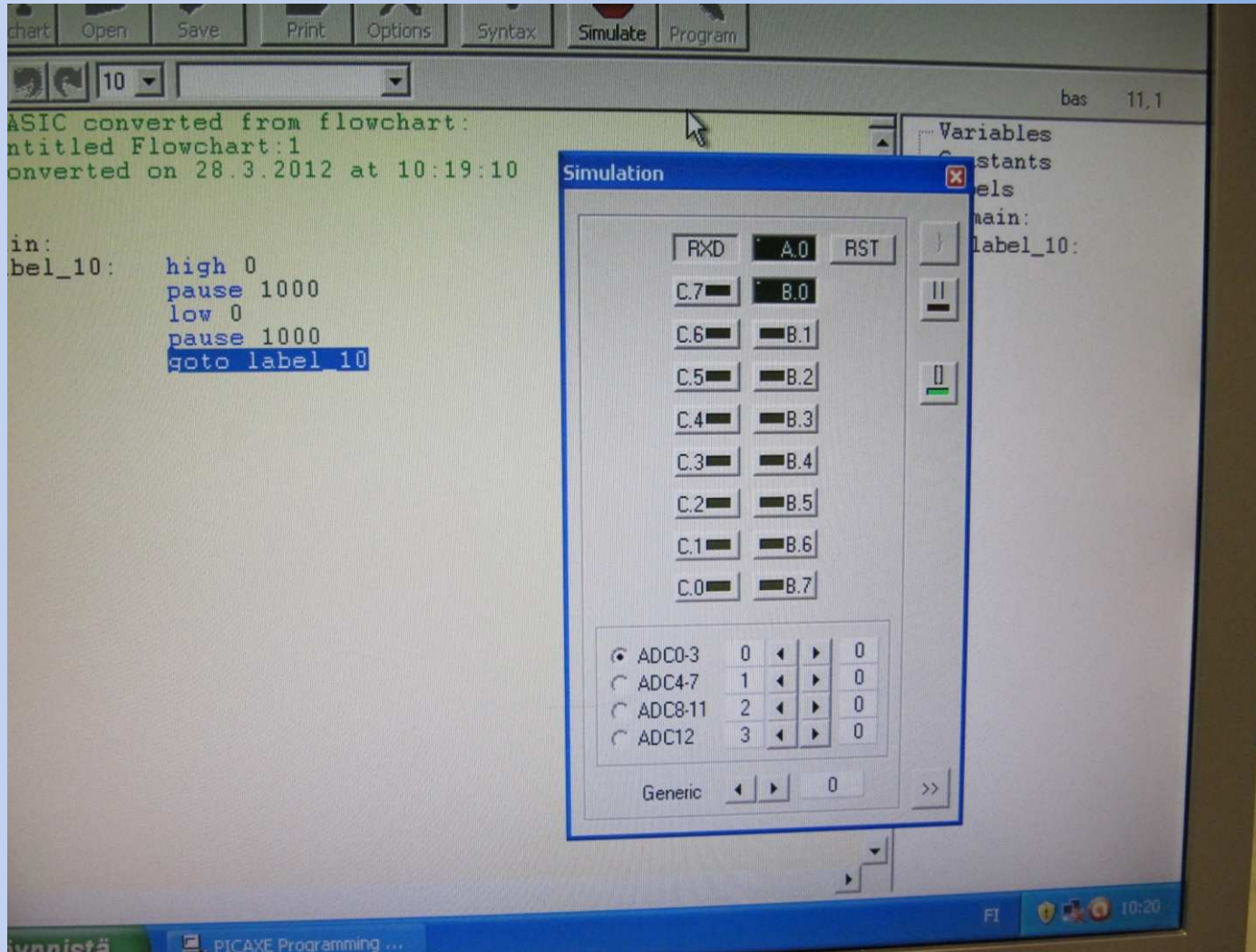
# Obrázová príloha



Obr.1 Jednoduchý vývojový diagram riadenia LED diódy

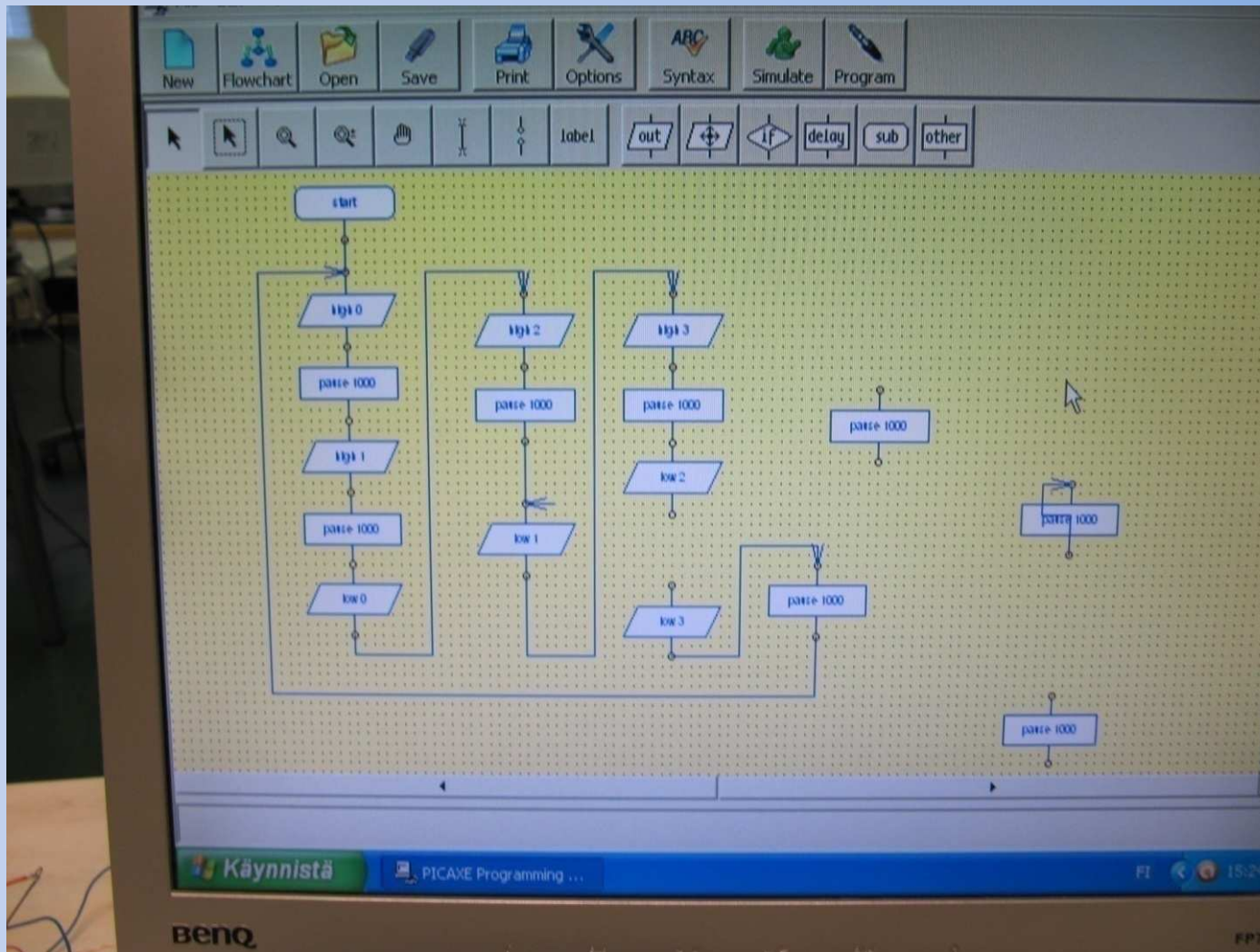


# Obrazová príloha



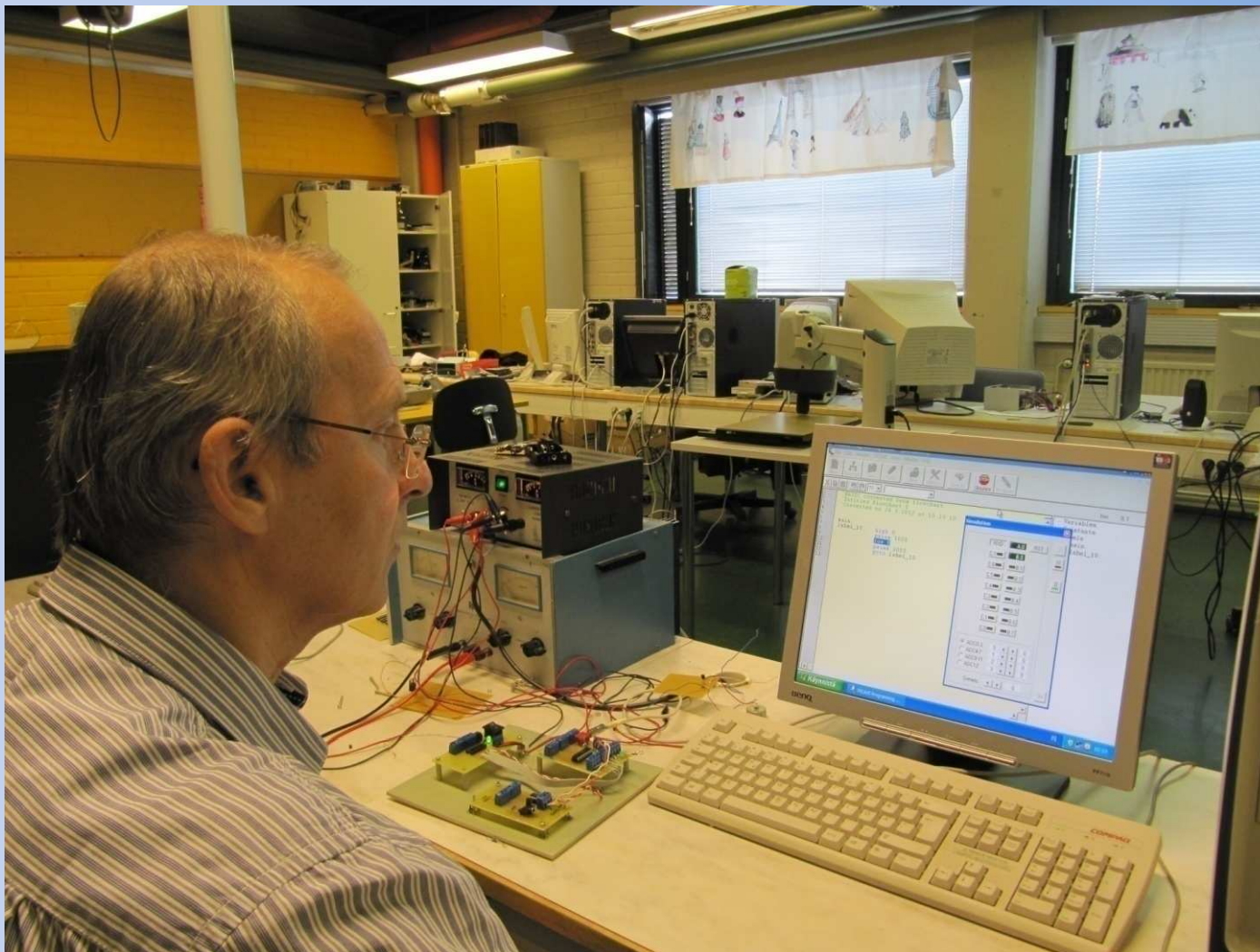
Obr.2 Testovanie programu v Basicu pomocou simulátora

# Obrázová príloha



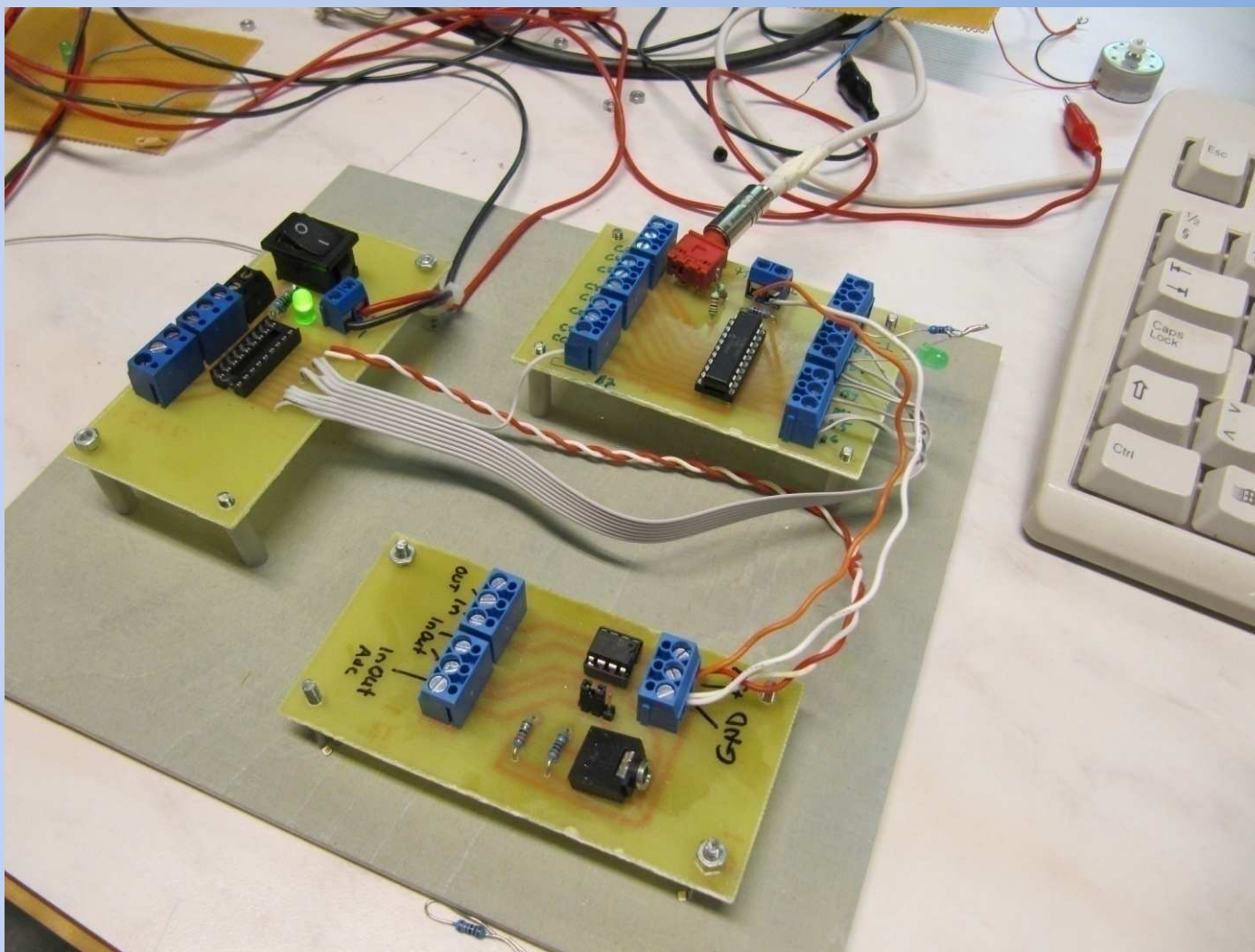
Obr.3 Tvorba zložitejšieho vývojového diagramu

# Obrazová příloha



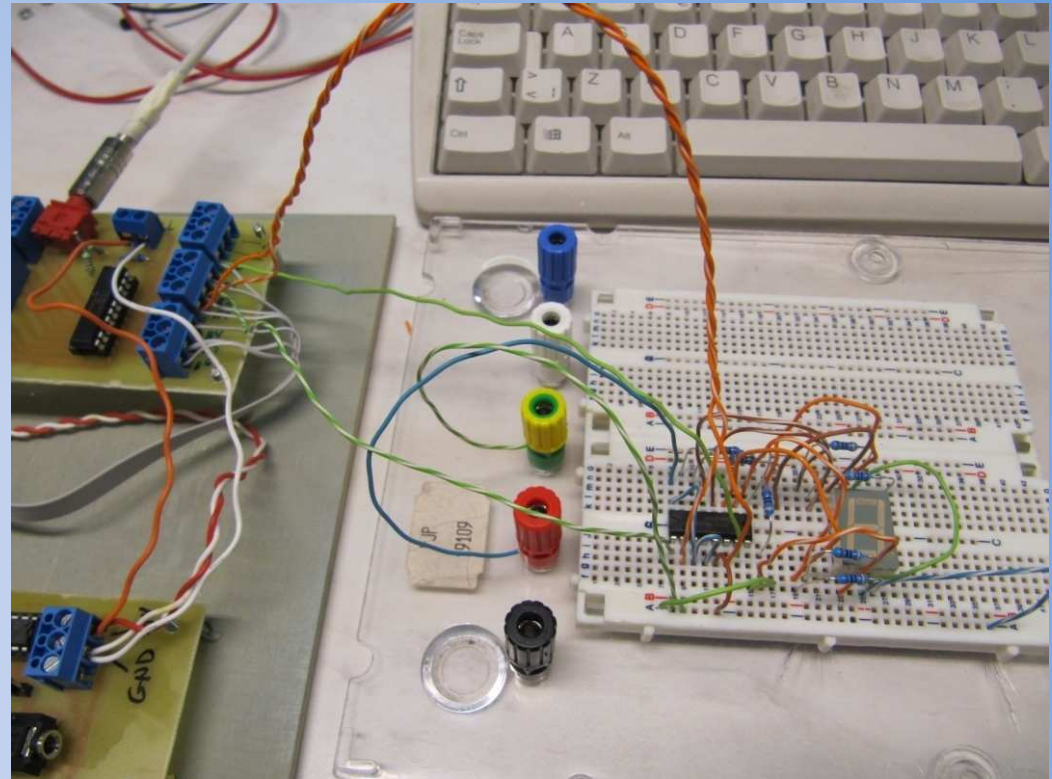
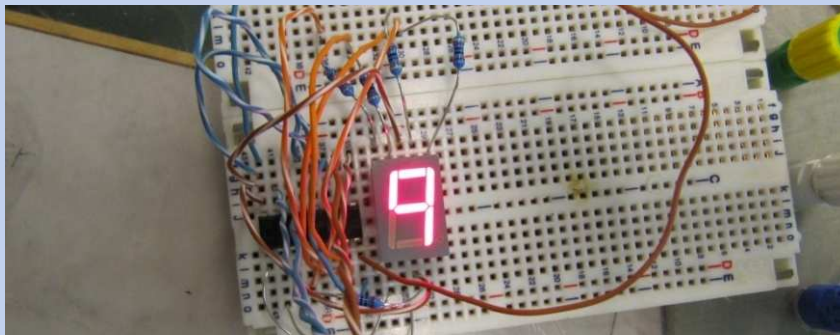
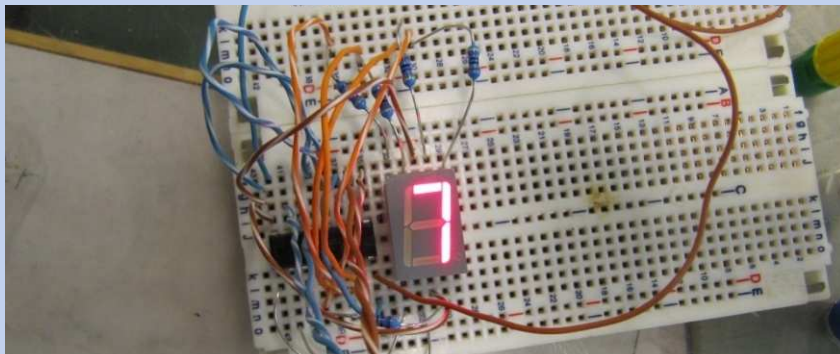
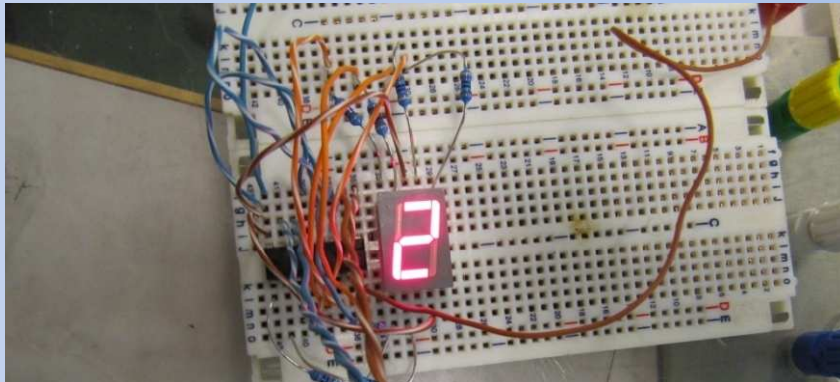
Obr.4 Práce v programovém prostředí editora PICAXE

# Obrazová príloha



Obr.5 Zapojenie montážnych skúšobných dosiek PCB

# Obrázová príloha



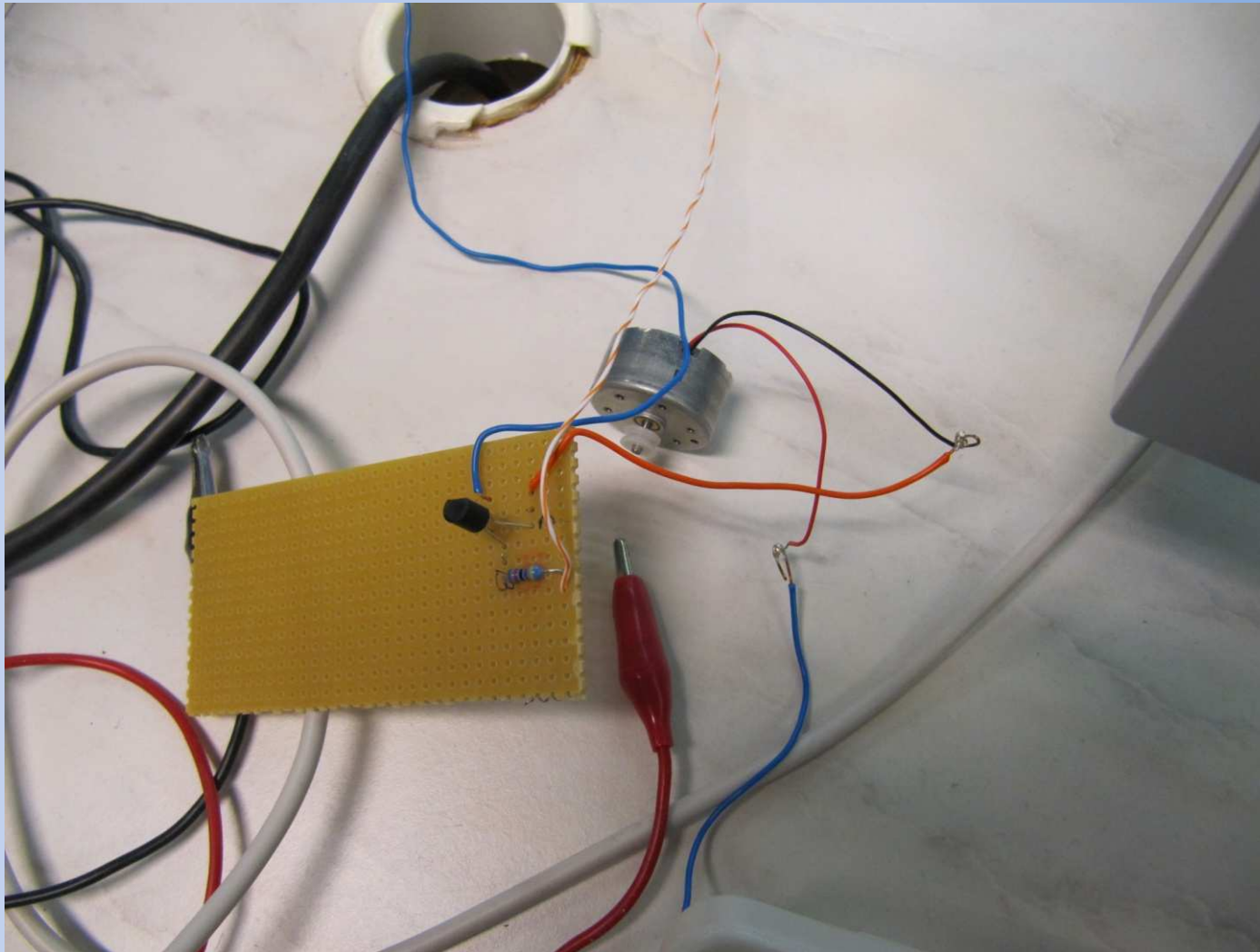
Obr.6 Ukážka činnosti riadenia 7-segment LED displeja s PICAXE

# Obrazová príloha



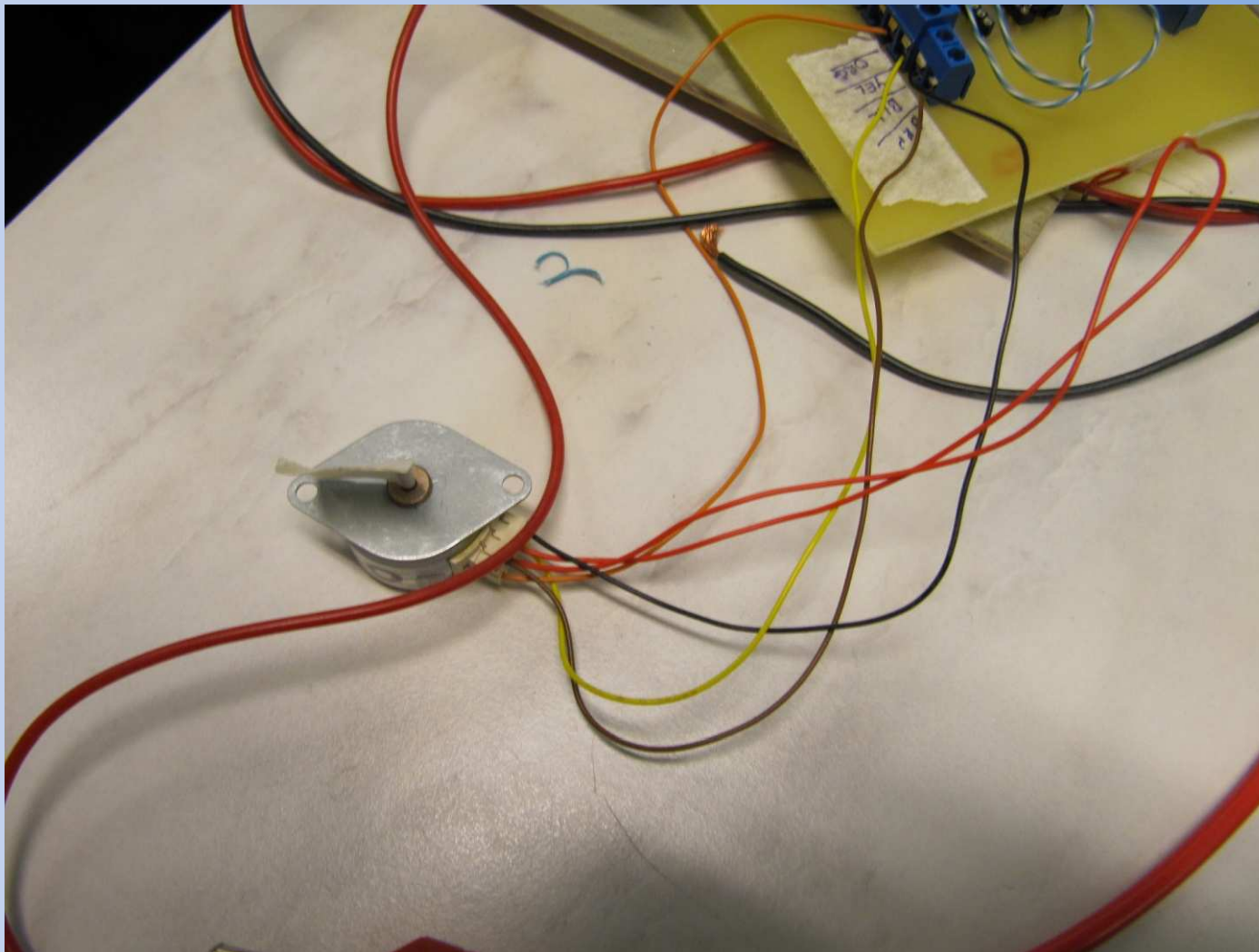
Obr.7 Práca pri odlad'ovaní programu mikrokontroléra

# Obrázová príloha



Obr.8 Riadenie rýchlosti otáčok jednosmerného motorčeka

# Obrázová príloha



Obr.9 Riadenie polohy hriadeľa krokového motorčeka



# Obrázová príloha



Obr.10 Spolupráca pri programovom riešení aplikácie

# Zdroje informácií

- [www.picaxe.com](http://www.picaxe.com)
- [ww1.microchip.com/downloads/en/DeviceDoc/41365E.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/41365E.pdf)
- [www.picaxe.com/docs/picaxex2.pdf](http://www.picaxe.com/docs/picaxex2.pdf)
- [www.freeinfosociety.com/media/pdf/3134.pdf](http://www.freeinfosociety.com/media/pdf/3134.pdf)
- <http://www.ermicro.com/blog/>
- [www.picaxe.com/docs/picaxe\\_manual1.pdf](http://www.picaxe.com/docs/picaxe_manual1.pdf)
- [www.picaxe.com/docs/picaxe\\_manual2.pdf](http://www.picaxe.com/docs/picaxe_manual2.pdf)
- [www.picaxe.com/docs/picaxe\\_manual3.pdf](http://www.picaxe.com/docs/picaxe_manual3.pdf)
- [www.hobbyrobot.cz/download/bas805.exe](http://www.hobbyrobot.cz/download/bas805.exe)
- [www.hobbyrobot.cz/PDF/PICAXE\\_basic\\_man\\_cz.pdf](http://www.hobbyrobot.cz/PDF/PICAXE_basic_man_cz.pdf)