

Úvod do programování PICAXE řady M2

Příjemce:

SOŠ a SOU, Uničov,
Moravské nám. 681
793 51 Moravské nám. 681

Doporučeno pro:

1. - 4. ročník
fyzika, elektrotechnika, robotika
Mgr. Radim Děrda

Předmět:

Autor:



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

PICAXE-XXM2

High pin

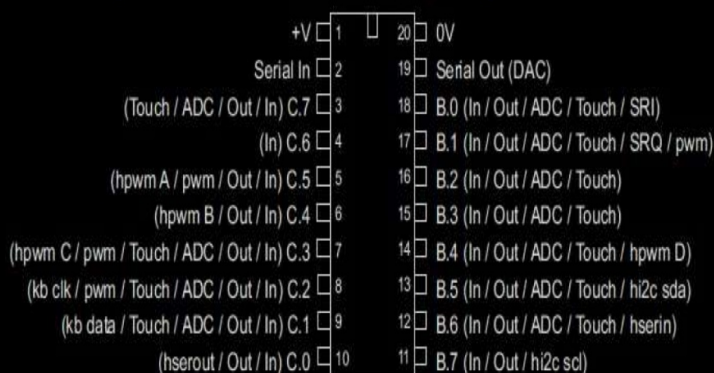
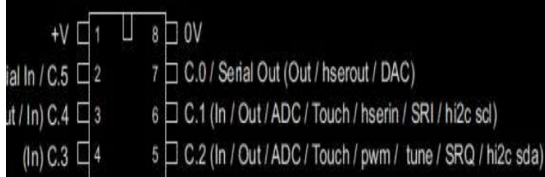
- nastaví pin na logickou 1 = přivede na pin napětí 5V (3,3V)
- *pin* = označení vývodu viz. schéma (B.0 - B.7 a C.0 - C.7)
- příklad rozsvícení LED (připojená k 0V přes rezistor 330 Ohm)

high C.1

Low pin

- nastaví pin na logickou 0 = přivede na pin napětí 0V
- příklad zhasnutí LED (připojená k 0V přes rezistor 330 Ohm)

low C.1



PICAXE-XXM2

Goto návěšti

- skok na určené místo
- *návěšti* - určení místa kam program skočí, při definování místa ukončen ":"

Pause čas

- pozastavení běhu programu
- čas - doba v milisekundách po kterou program čeká
- příklad nekonečného blikání s periodou 1s (500ms + 500ms)

Skok01:

```
high C.1
pause 500
low C.1
pause 500
goto Skok01
```

PICAXE-XXM2

Proměnné

- mohou uložit číslo do paměti
- řada M2 disponuje
 - 28 proměnnými typu Byte s názvy b0, b1, b2 ... b27
 - do těchto proměnných je možné uložit číslo v rozsahu 0 - 255
 - Bytové proměnné lze sloučit do proměnných typu Word
 - do těchto proměnných je možné uložit číslo v rozsahu 0 - 65535
 - jmenují se w0, w1, w2 ... w13
 - pokud použijeme proměnnou w0, změní se hodnoty v b0 a b1



PICAXE-XXM2

Proměnné

- jsou v mikrokontroléru uloženy ve dvojkové soustavě
- = číslo je "vytvořeno" ze součtu mocnin čísla 2

Příklad zápisu čísla 107:

v desítkové soustavě - základ 10

$$107 = 1 \cdot 10^2 + 0 \cdot 10^1 + 7 \cdot 10^0$$

ve dvojkové soustavě - základ 2

$$107 = 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

- jednotlivé řády se ukládají do odpovídajících bitů
(bit - nejmenší paměťové místo v mikrokontroleru)
- => nultý řád do nultého bitu, první řád do prvního bitu ...

$$107 = 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

1	1	0	1	0	1	1
6	5	4	3	2	1	0

PICAXE-XXM2

Proměnné

- typ Byte je tvořen 8 bity => maximální uložitelné číslo je 255

1	1	1	1	1	1	1	1
7	6	5	4	3	2	1	0

 = 255

- typ Word je tvořen 2 proměnnými typu Byte = 16 bity
- => maximální uložitelné číslo je 65535

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

 = 65535

PICAXE-XXM2

For *prom* = *start* to *cíl* step \pm *krok* ... next *prom*

- cyklus s předem známým počtem opakování
- příkazy mezi *for* a *next* se provedou při každém průchodu
- *prom* = proměnná cyklu, obsahuje aktuální hodnotu průchodu
- *start* - počáteční hodnota cyklu
- *cíl* - koncová hodnota cyklu
- \pm *krok* - hodnota, která se přičte/odečte k aktuální hodnotě proměnné cyklu - *prom*
 - nemusí být uvedena, pak se použije +1
 - pokud je záporná, musí platit, že *start* > *cíl*
- cyklus se provádí dokud platí, že hodnota *prom* náleží do intervalu <*start*,*cíl*>

PICAXE-XXM2

- příklad ver. 1 - zpomalující se blikání

```
for b0 = 1 to 20
  b1 = b0 * 10
  high C.1
  pause b1
  low C.1
  pause b1
next b0
```

- příklad ver. 2 - zpomalující se blikání

```
for b0 = 10 to 200 step 10
  high C.1
  pause b0
  low C.1
  pause b0
next b0
```

PICAXE-XXM2

- příklad ver. 3 - zrychlující se blikání

```
for b0 = 200 to 10 step -10
  high C.1
  pause b0
  low C.1
  pause b0
next b0
```

PICAXE-XXM2

PinX.x

- zjištění logické hodnoty na pinu X.x
- X.x - jméno konkrétního pinu = jméno portu + číslo pinu
(C.1, B.2,...)
- je-li na pin(nožičku), přivedeno
 - napětí < 0,7V => log. 0
 - napětí > 1,4V => log. 1

PICAXE-XXM2

If podmínka

- příkaz podmíněného skoku
- provede část programu, jen když je *podmínka* splněna
- *podmínka* - logický výraz, který nabývá hodnoty pravda (true), nebo nepravda (false)
- složená podmínka = více podmínek spojených logickou spojkou
AND => pravda, když platí obě podmínky zároveň
OR => pravda, když platí aspoň jedna z podmínek
- příkaz je možné použít ve třech základních variantách

var. 1 - v případě splnění podmínky skoč na návěští "Start",
- čekací smyčka - čeká tak dlouho, dokud je na pinu C.1 napětí
- obě možnosti zápisu jsou správné

Start:

```
if pinC.1 = 1 then Start
```

Start:

```
if pinC.1 = 1 then goto Start
```

PICAXE-XXM2

var. 2 - v případě splnění podmínky proved' příkazy mezi "if" a "endif"
- když bude na C.1 a zároveň na C.2 napětí, tak "rozsvit" B.1 a B.0

```
if pinC.1 = 1 AND pinC.2 = 1 then  
  high B.1  
  high B.0  
endif
```

var. 3 - v případě splnění podmínky proved' příkazy mezi "if" a "else",
jinak není-li splněna, proved' příkazy mezi "else" a "endif"
- když bude na C.1, nebo na C.2 napětí, tak "rozsvit" B.1 a B.0,
jinak je "zhasni"

```
if pinC.1 = 1 OR pinC.2 = 1 then  
  high B.1  
  high B.0  
else  
  low B.1  
  low B.0  
endif
```

PICAXE-XXM2

Pullup on/off

- připojí vstupní piny přes interní rezistory 10k, ke kladnému napětí, takže na volném pinu je log. 1 a pro log. 0 stačí jen pin spojit s 0 V
- příklad použití

```
pullup on
```

```
...
```

```
pullup off
```

Enabletime / disabletime

- nastaví / zastaví přičítání času do proměnné TIME
- proměnná TIME obsahuje čas v sekundách od posledního zapnutí
- proměnnou TIME je možné vynulovat programově

```
TIME = 0
```

PICAXE-XXM2

Sertxd(*data*)

- pošle po programovacím kabelu *data* do PC
- možno načíst v Picaxe->Terminál, nutno nastavit 4800Bd, 8N1, případně je možné použít Picaxe->Datové spojení
- *data* jsou ASCII kody znaků =>
 - text se uzavírá do úvozovek: "Picaxe"
 - výpis obsahu proměnné pomocí mřížky: #b0
 - odřádkování, na konci dat použít: (... , 13, 10)

- příklad

```
enabletime  
time = 0  
sertxd("Picaxe běží:", 13, 10)
```

Start:

```
b0 = time % 60 ;zbytek po celočíselném dělení  
w1 = time / 60 ;celočíselné dělení  
sertxd(#w1," minut a ", #b0," sekund ", 13, 10)  
pause 1000  
goto Start
```