if

Platí pro všechny typy PICAXE

Syntaxe při použití jednoho řádku:

IF condition THEN label

IF condition THEN GOTO label

IF condition THEN GOSUB label

IF condition THEN EXIT

Syntaxe při použití více řádků:

IF condition THEN

{kód}

ELSEIF condition THEN

{kód}

ELSE

{kód}

ENDIF

Condition – podmínka je logický výraz, při jejímž splnění budou vykonány příkazy umístěné za then. Při nesplnění podmínky budou vykonány příkazy za else (je-li použito), jinak program pokračuje dalším řádkem

Kód – posloupnost příkazů která bude provedena v závislosti na vyhodnocení podmínky – condition. Posloupnost může obsahovat výkonné příkazy, nebo další podmíněné příkazy if

Popis:

Vykoná část programového kódu v závislosti na splnění podmínky.

Pokud je podmínka splněna, budou vykonány příkazy umístěné za příkazem then, v případě, že podmínka není splněna a je přítomen příkaz else, budou vykonány příkazy umístěné za tímto příkazem. Není-li příkaz else použit, program pokračuje následujícím řádkem.

Příkaz if musí být ukončen příkazem endif.

Podmínka může obsahovat nejen proměnné, ale také může testovat stav pinů portů.

Podmínku testující vnější piny je vhodné vložit do těla nekonečné smyčky a pro její ukončení použít příkaz exit.

Podmínky - conditions

Pro porovnání obsahu proměnné s konkrétní hodnotou:

IF b1 >= 10 THEN …

Pro porovnání dvou proměnných:

IF b2 < b3 THEN …

Použitelná porovnávací pravidla:

=	rovná se
is	rovná se
<>	nerovná se
!=	nerovná se
>	větší než
<	menší než
>=	větší nebo rovno
<=	menší nebo rovno

Pro zjištění stavu vstupního pinu:

IF pin2 = 1 THEN GOSUB buttonpushed

IF pinC.0 = 0 THEN GOSUB buttonnotpushed

Vícenásobná podmínka cyklu (složená podmínka)

Pro vytvoření složitější podmínky cyklu se používají logické operátory:

logický součin - AND (a zároveň) - musí platit obě podmínky a platila i výsledná

logický součet - OR (nebo) - platí-li jedna z podmínek, pak platí celá podmínka

IF b0 > 10 AND b0 < 20 THEN GOTO
between

je-li b0>10 a zároveň b0<20, pak skoč na „between" (musí platit obě dvě podmínky současně).

```
IF pinC.0 = 1 OR b1 = 1 OR b2 = 3 THEN
EXIT
```

je-li C.0=1 nebo b1=1 nebo b2=3, pak ukonči tuto část programu (stačí, aby platila aspoň jedna z podmínek).

```
IF condition THEN label
```

Je-li podmínka splněna, program přejde (skočí) na místo specifikované label (návěští). Příkaz je zkráceným zápisem příkazu IF condition THEN GOTO label.

```
IF condition THEN GOTO label
```

Je-li podmínka splněna, program přejde (skočí) na místo specifikované label (návěští).

```
IF condition THEN GOSUB label
```

Je-li podmínka splněna, program spustí podprogram (odskočí si do podprogramu) na místo specifikované label (návěští). Pro návrat z podprogramu se použije příkaz return, po návratu pokračuje program následujícím příkazem.

IF condition THEN EXIT

Je-li podmínka splněna, program ukončí aktuální cyklus (do .. loop nebo for .. next) a přejde na následující příkaz za cyklem.

```
IF condition THEN
{kód}
ENDIF
```

Je-li podmínka splněna, program provede příkazy, které se nacházejí mezi slovy then a endif.

```
IF condition THEN
{kód}
ELSE
{kód}
ENDIF
```

Je-li podmínka splněna, program provede příkazy, které se nacházejí mezi slovy then a else.

Jinak (= podmínka není splněna) provede příkazy mezi slovy else a endif.

```
IF condition THEN
{kód}
ELSEIF condition THEN
{kód}
ENDIF
```

Je-li podmínka splněna, program provede příkazy, které se nacházejí mezi slovy then a else.

Jinak (= podmínka není splněna) vyhodnotí 2. podmínku uvedenou za elseif a pokud bude splněna, provede příkazy mezi slovy then (2. then) a endif. Pokud ani 2. podmínka není splněna, přejde na další příkaz za endif.

```
IF condition THEN
{kód}
ELSEIF condition THEN
{kód}
ELSEIF condition THEN
{kód}
ENDIF
```

Použití stejné jako v přecházejícím případě, pouze testováno více podmínek. Pokud je to možné, je vhodné raději použít příkaz select case.

```
IF condition THEN
{kód}
ELSEIF condition THEN
{kód}
ELSE
{kód}
ENDIF
```

Je-li podmínka splněna, program provede příkazy, které se nacházejí mezi slovy then a else.

Jinak (= podmínka není splněna) vyhodnotí 2. podmínku uvedenou za elseif a pokud bude splněna, provede příkazy mezi slovy then (2. then) a else. Pokud ani 2. podmínka není splněna, provede příkazy mezi slovy else a endif.

Související příkazy:

- if pin
- else
- endif

Příklad:

Po přivedení log.1 na C.0 se LED na B.1 rozsvítí na 5 s.

```
main:
if pinC.0 = 1 then      ; hlídej úroveň na C.0
goto flsh               ; v případě stisku (C.0=1) skoč na „flsh“
endif
goto main               ; zpět na „main“ (nekonečná smyčka)

flsh:
high B.1                ; rozsviť LED na B.1
pause 5000              ; čekej 5 s
low B.1                 ; zhasni LED na B.1
goto main               ; zpět na „main“
```