

let

Platí pro všechny typy PICAXE

Syntaxe:

{LET} variable = {-} value ?? value ...

Variable – proměnná do které bude přiřazováno (pro PICAXE řady M2: b0 až b27, w0 až w14)

Value(s) – proměnná nebo konstanta, která má být uložena

{LET} dirs = value

{LET} dirsA = value

{LET} dirsB = value

{LET} dirsC = value

{LET} dirsD = value

Value – proměnná nebo konstanta, která konfiguruje piny konkrétního portu jako: 1 – vstupní nebo 0 – výstupní

{LET} pins = value

{LET} pinsA = value

{LET} pinsB = value

{LET} pinsC = value

{LET} pinsD = value

Value – proměnné/konstanty, nastavující hodnotu jednotlivých pinů na konkrétním portu

Popis:

Přiřadí do proměnné „variable“ výsledek operace za rovnítkem, nebo přiřadí portu hodnotu určenou hodnotou „value“. Vždy musí být zachován stejný datový typ na obou stranách rovnítko (byte = byte; word = word).

Klíčové slovo „let“ je nepovinné.

POZOR - matematické operace se provádějí zásadně zleva do prava.

Výsledek operace je vždy celočíselný v rozsahu word, nebo byte.

Použitelné matematické operátory a funkce:

+	sčítání
-	odčítání
*	násobení (nižší byty výsledku)
**	násobení (vyšší byty výsledku)
/	celočíselné dělení
// (nebo %)	modulo = zbytek po celočíselném dělení
MAX	omezí výsledek shora na maximální hodnotu
MIN	omezí výsledek zdola na minimální hodnotu
AND (nebo &)	bitový operátor AND, provede log. součin mezi jednotlivými bity
OR (nebo)	bitový operátor OR, provede log. součet mezi jednotlivými bity
XOR	bitový operátor XOR, provede exkluzivní disjunkci mezi jedn. bity
NAND	bitový operátor NAND, provede negovaný součin mezi jedn. bity
NOR	bitový operátor NOR, provede negovaný součet mezi jedn. bity
ANDNOT (nebo &/)	bitový operátor AND NOT, provede logický součin s druhým operátorem negovaným (nerovná se NAND)
ORNOT (nebo /)	bitový operátor OR NOT, provede logický součet s druhým operátorem negovaným (nerovná se NOR)
XNOR (nebo ^/)	bitový operátor XOR NOT, provede exkluzivní disjunkci s druhým operátorem negovaným (nerovná se XOR)

U PICAXE řady X1 a X2 je také možno použít:

<<	bitové posunutí doleva
>>	bitové posunutí doprava
*/	dělení (vrací prostřední část výsledku)
SIN	sinus, argument ve stupních (0 to 65535), vrací výsledek*100
COS	cosinus, argument ve stupních (0 to 65535), vrací výsledek*100
SQR	druhá odmocnina
INV	invertování
NCD	NCD n, je totožné jako - celá část z druhé odmocniny n
DCD	DCD n, je totožné jako - 2n
BINTOBCD	převod z binárního čísla do BCD kódu
BCDTOBIN	převod z BCD kódu na binární číslo
REV	upraví šířku dat
DIG	vrátí BCD číslice

LET dirs / dirsA / dirsB / dirsC / dirsD = value

Pro práci s vnějšími piny je třeba nastavit zda jde o pin vstupní, či výstupní. Je mnoho způsobu (input, output, high, low...) nebo je možné použít příkaz LET dirs.

Platí, že při nastavení je 1 = vstupní pin a 0 = výstupní pin.

LET pins / pinsA / pinsB / pinsC / pinsD = value

Při práci s vnějšími výstupními piny je třeba nastavit jejich výstup. Je možné je nastavovat pomocí high, low, apod. nebo je možné použít příkaz LET pins. Před jeho použitím je třeba nakonfigurovat pin jako výstupní.

Platí, pokud bychom pracovali s portem C (LET pinsC), že:

0.bit = C.0, 1.bit = C.1, 2.bit = C.2, ... , dále platí, že high = 1, low = 0.

Příklad:

Zvyšující se signál, s demonstrací matematických operací.

main:

```
inc b0                ; zvýš b0 o +1
sound B.7,(b0,50)    ; tón na výstup B.7
if b0 > 50 then rest ; když je b0 větší než 50 odskoč na „rest“
goto main            ; zpět na „main“
```

rest:

```
let b0 = b0 max 10   ; omez horní hranici obsahu proměnné b.0
goto main            ; zpět na „main“
```