

push

Platí pro PICAXE 20X2, 28X2, 40X2

Syntaxe:

PUSH data {, data, data ...}

Data – jeden byte, proměnná nebo konstanta, která bude uložena do zásobníku.

Popis:

Uloží hodnotu proměnné nebo konstantu o velikosti jednoho byte do zásobníku.

Proměnná typu bit bude doplněna na velikost byte. Všechny hodnoty dříve uložené budou posunuty o jedno místo a nová hodnota bude uložena na začátek zásobníku. Zásobník je 32 byte hluboký a je oddělený od zásobníku používaného příkazy pushram a popram.

Pořadí vkládání a odběru proměnných

Proměnné jsou ze zásobníku odebírány v opačném pořadí, než v jakém jsou do něj vkládány. Je-li napřed vložena hodnota b1 a poté hodnota b2, pak se ze zásobníku odebere nejprve b2 a pak b1. Pro zásobník tohoto typu se používá zkratka LIFO. (Last In First Out – první dovnitř poslední ven.)

Jsou-li proměnné b1 a b2 vkládány do zásobníku jedním příkazem push, je nutno je příkazem pop ze zásobníku odebrat v opačném pořadí, než byly vloženy.

Tedy...

```
push b1, b2
gosub myroutine
pop b2, b1
```

... je totéž, jako:

```
push b1
push b2
gosub myroutine
pop b2
pop b1
```

Proměnné datového typu word

Všimněte si, že zásobník má šířku 1 byte, což je třeba brát v úvahu při používání proměnných typu word. Když je v příkazu push uvedena proměnná typu word, uloží se do zásobníku jen nejméně významný byte. Když se použije proměnná typu word v příkazu pop, ze zásobníku se přečte pouze jeden byte (vyšší byte bude nulový).

Pokud je tedy třeba uložit proměnnou typu word do zásobníku, je nutno uložit každý její byte zvlášť. Například proměnná w2 se skládá ze dvou byte: b5 (MSB významnější byte) a b4 (LSB méně významný byte). Zrovna tak při načítání proměnné typu word ze zásobníku je třeba načíst každý yte zvlášť. Mějte prosím na paměti, že byty musí být načteny v opačném pořadí, než v jakém byly ukládány.

Uložení a následující načtení proměnné W2 je v následujícím příkladu:

```
push b4          ; ulož proměnnou w2 do zásobníku  
push b5
```

```
pop b5           ; vyjmi proměnnou w2 ze zásobníku  
pop b4
```

Související příkazy:

- pop
- pushram

Příklad 1:

Příklad ukazuje, jak je proměnná b0 uložena do zásobníku, následně se změní její hodnota, a nakonec je původní hodnota obnovena příkazem pop.

```
main:  
b0 = 0  
sertxd("Before push, b0=", #b0, cr, lf)  
push b0  
sertxd("After push, b0=", #b0, cr, lf)  
b0 = 99  
sertxd("After change, b0=", #b0, cr, lf)  
pop b0  
sertxd("After pop, b0=", #b0, cr, lf)  
end
```

Příklad 2:

```
main:  
b1 = 11  
b2 = 22  
push b1  
push b2  
b1 = 99  
b2 = 99  
pop b2  
pop b1  
sertxd("b1=", #b1, " b2=", #b2, cr, lf)  
end
```

Příklad 3:

```
main:
b1 = 11
b2 = 22
push b1, b2
b1 = 99
b2 = 99
pop b2, b1
sertxd("b1=", #b1, " b2=", #b2, cr, lf)
end
```

Příklad 4:

```
main:
b0 = 11
push 1, b0
gosub mysub

b1 = 22
push 2, b1
gosub mysub
end

mysub:
pop b2
pop b1
sertxd("b1=", #b1, " b2=", #b2, cr, lf)
return
```