

pushram

Platí pro PICAXE 20X2, 28X2, 40X2

Syntaxe:

PUSHRAM

PUSHRAM CLEAR

Popis:

Uloží obsah paměti RAM z adresy 0 až 15 (proměnné b0 až b15) do 4 úrovněového zásobníku. Chcete-li použít proměnné b0 až b15 uvnitř makra nebo podprogramu, může být užitečné uschovat původní hodnoty, aby proměnné mohly být použity jako dočasné proměnné v rámci makra nebo podprogramu. Příkaz pushram uloží obsah proměnných b0 až b15 do zásobníku tak, aby se těchto 16 proměnných mohlo dočasně používat pro jiné účely. Příkaz popram pak obnoví původní hodnoty všech proměnných na konci makra nebo podprogramu. Volbou clear se hodnoty proměnných b0 až b15 při uložení do zásobníku navíc ještě vynulují. Pokud není parametr clear použit, hodnoty ukládaných proměnných zůstanou zachovány. Zásobník použitý pro příkazy pushram a popram má 4 úrovně a je oddělený od zásobníku použitého příkazy push a pop.

Související příkazy:

- popram
- push

Příklad 1:

Zachování hodnot proměnných

Před voláním podprogramu je vykonán příkaz pushram, aby proměnné mohly být obnoveny, pokud je podprogram používá a změní jejich hodnoty. Po dokončení podprogramu musí být proveden příkaz popram, který obnoví původní hodnoty proměnných.

```
main:
for b1 = 1 to 5
sertxd("Before gosub, b1=", #b1, cr, lf)
pushram
gosub mysub
popram
sertxd("After gosub, b1=", #b1, cr, lf, cr, lf)
next
end
mysub:
b1 = 99
sertxd("  Inside mysub, b1=", #b1, cr, lf)
return
```

Příklad 2:

```
main:
for b1 = 1 to 5
sertxd("Before gosub, b1=", #b1, cr, lf)
gosub mysub
sertxd("After gosub, b1=", #b1, cr, lf, cr, lf)
next
end
```

```
mysub:
pushram
b1 = 99
sertxd("  Inside mysub, b1=", #b1, cr, lf)
popram
return
```

Příklad 3:

```
main:
for b1 = 1 to 5
sertxd("Before gosub, b1=", #b1, cr, lf)
gosub mysub1
gosub mysub2
sertxd("After gosub, b1=", #b1, cr, lf, cr, lf)
next
end
```

```
mysub1:
pushram
b1 = b1 + 1
sertxd("  Inside mysub1, b1=", #b1, cr, lf)
popram
return
```

```
mysub2:
pushram clear
b1 = b1 + 1
sertxd("  Inside mysub2, b1=", #b1, cr, lf)
popram
return
```