

run

Platí pro PICAXE 14M2, 18M2, 20M2, 20X2, 28X2, 40X2

Syntaxe:

RUN slot

Slot – je proměnná nebo konstanta určující, který program bude spuštěn

Popis:

PICAXE 08M2 podporuje pouze slot 0.

PICAXE 14M2, 18M2 a 20M2 mají dva samostatné vnitřní programové sloty.

PICAXE 20X2 podporuje pouze slot 0.

PICAXE 28X2 a 40X2 mají čtyři samostatné vnitřní programové sloty.

Po resetu je vždy spuštěn program ze slotu 0.

Nový program se uloží do slotu určeného direktivou #slot, která se vkládá do zdrojového kódu. Pokud tato direktiva není použita, program se uloží do slotu 0. Do jednoho slotu je možno vložit jen jeden program. Uložení programu do vybraného slotu nejsou programy v ostatních slotech ovlivněny.

Chcete-li spustit program z jiného slotu, například ze slotu 1 (v kódu programu určeno direktivou #slot 1), použijte příkaz run1. Tento příkaz zastaví běh aktuálně prováděného programu a spustí program, uložený ve slotu 1. Nastavení pinů a obsah proměnných zůstane zachován, takže mohou být mezi programy sdíleny. Všechny ostatní programové funkce, například gosub a return jsou ale při spuštění druhého programu vynulovány. Program ve slotu 1 lze tedy použít jen jako náhradu za odskok na dané návěští a nikoli jako podprogram.

V programu ve slotu 1 můžete také použít příkaz run 0 a restartovat tak program ve slotu 0. Pokud chcete navíc vynulovat proměnné, musíte ke spuštění programu ve slotu 0 použít příkaz reset, což je totéž, jako run 0 + reset proměnných.

PICAXE má pouze jednu vnitřní oblast dat EEPROM (používanou příkazy eeprom, read a write), kterou sdílí všechny programové sloty. Nahrání programu do libovolného slotu bude vždy aktualizovat tutéž (sdílenou) paměť EEPROM. Chcete-li tuto aktualizaci zakázat, je možné v zaváděném programu použít direktivu #no_data. Tím se zabrání aktualizaci datové oblasti EEPROM (tj. případný příkaz EEPROM je ignorován).

Externí programové sloty

Stejně jako interní paměťové sloty, u PICAXE řady X2 můžete použít další 4 sloty připojením externí I2C paměti EEPROM (24LC128). Protože na sběrnici I2C může být použito až 8 pamětí typu 24LC128, dostáváme teoreticky až 32 (8x4) dalších programových slotů.

Pro paměť 24LC128 na adrese 0 (tj. piny A0, A1, A2 všechny připojen k 0V) jsou externí programové sloty číslovány 4 až 7, pro další paměti 24LC128 je číslo slotu vypočítáno následovně:

Bit7	24LC128 adresovací pin A2
Bit6	24LC128 adresovací pin A1
Bit5	24LC128 adresovací pin A0
Bit4	vyhrazeno pro budoucí použití, nastavte na 0
Bit3	vyhrazeno pro budoucí použití, nastavte na 0
Bit2	1 = I2C 0 = internal
Bit1 až 0	kombinace vybere jeden ze 4 možných slotů

Spuštění programu z externí I2C paměti má následující omezení:

- I2C sběrnice je vyhrazena pouze pro čtení programu
- Výstupy použité pro I2C nemohou být použity pro žádný jiný účel
- Veškeré příkazy pro hardwarové I2C a SPI jsou zcela ignorovány
- Rychlost provádění programu je snížena vzhledem k relativně pomalé rychlosti čtení dat z externí paměti 24LC128
- Externí paměť 24LC128 slouží pouze k uložení programu. Veškerá data, nahraná příkazem eeprom, nejsou uložena externě. Příkazy read a write i nadále používají vnitřní EEPROM PICAXE.

Viz také příkaz booti2c, což může být vhodnější pro použití slotů 4 až 7.

Další informace

Obvyklým způsobem využití programových slotů je otestovat při startu logické kombinace na určených vstupech (například připojených k přepínači nebo zkratovacím propojkám) a pak spustit program ze slotu, určeného touto kombinací vstupů.

Programové sloty také mohou být použity v rámci jednoho programu, ale musíte vzít v úvahu následující:

- Mezi sloty nemohou být sdíleny žádné podprogramy ani přerušení.
- Gosub / return stack se při přechodu z jednoho slotu do druhého resetuje.
- Výstupy a proměnné ani zápisníková paměť resetovány nejsou.
- Příkaz run X je třeba považovat za goto na začátek programu X
- Run 0 není totéž jako příkaz reset, protože příkaz reset také nuluje všechny proměnné a převede všechny piny na vstupy.

Související příkazy:

- booti2c
- #slot

Příklad:

Tento program spustí kód ve slotu 1 nebo 2 v závislosti na stavu dvou vstupních pinů při spuštění programu.

```
#slot 0

init:
if pinC.1 = 1 then      ; test vstupního pinu po resetu
run 1                  ; je li pin C.1 roven 1, spustí se program se slotu 1
endif

if pinC.2 = 1 then      ; test vstupního pinu po resetu
run 2                  ; je li pin C.2 roven 1, spustí se program se slotu 2
endif

main:
high B.1              ; jinak se spustí program ze slotu 0
etc                   ; ... pokračování kódu
```