

Příklad č. 1

Napište podprogram `cekej2ms`, jehož účelem bude realizovat čekací smyčku s dobou trvání 2 ms při frekvenci hodin CPU 12 MHz (dobou trvání se rozumí délka příslušného volání, tedy včetně instrukcí volání a návratu z podprogramu). Podprogram bude napsán tak, aby svou činností neovlivňoval stav registrů (uloží všechny použité na zásobník). SREG ukládat nemusíte.

```
.include    "m32def.inc"

// Konstanty
.equ       N1          = 123
.equ       N2          = 64

// Aliasy registrů

// Vlastní program
.cseg
.org      0x0000
// void Main(void)

// Inicializace zásobníku
ldi      r16, LOW(RAMEND)
out      SPL, r16
ldi      r16, HIGH(RAMEND)
out      SPH, r16

rcall    cekej2ms    // Zavolání podprogramu pro čekání 2ms

end:     rjmp        end

cekej2ms:
// Uložení potřebných registrů na zásobník
push    r16
push    r17

// Vnější cyklus
ldi     r16, N1
Smycka1:
// Vnitřní cyklus
ldi     r17, N2
Smycka2:
dec     r17
brne   Smycka2
// Konec vnitřního cyklu
dec     r16
brne   Smycka1
// Konec vnějšího cyklu

// Vrácení hodnot ze zásobníku
pop     r17
pop     r16
ret                    // Návrat z podprogramu
```

Příklad č. 2

Vyhledejte na [www.fy Atmel](http://www.fy.atmel.com) aplikační poznámku, která obsahuje podprogramy pro násobení a dělení. Napište testovací program pro podprogram 8 b dělení bez znaménka. Pomocí podprogramu vydělte dvě proměnné, výsledky uložte do dalších dvou proměnných (podíl a zbytek). Pokuste se pochopit algoritmus podprogramu a vysvětlete jeho funkci, případné „triky“ v něm použité, atd. Jedná se o algoritmus, probíraný na přednášce?

```
.include "m32def.inc"

// Aliasy registrů podprogramu dělení
.def drem8u =r15 ;remainder
.def dres8u =r16 ;result
.def dd8u =r16 ;dividend
.def dv8u =r17 ;divisor
.def dcnt8u =r18 ;loop counter

.dseg
prom1: .byte 1
prom2: .byte 1
vysl: .byte 1
zbytek: .byte 1

.cseg
.org 0x0000
// Inicializace zásobníku
ldi r16, LOW(RAMEND)
out SPL, r16
ldi r16, HIGH(RAMEND)
out SPH, r16

// vysl = prom1 / prom2
// zbytek = prom1 % prom2
lds dd8u, prom1 // kopie = volání hodnotou
lds dv8u, prom2

rcall div8u

// Uložení návratové hodnoty
sts vysl, dres8u
sts zbytek, drem8u

end: rjmp end

// Podprogram dělení 8u/8u
div8u: sub drem8u,drem8u ;clear remainder and carry
      ldi dcnt8u,9 ;init loop counter
d8u_1: rol dd8u ;shift left dividend
      dec dcnt8u ;decrement counter
      brne d8u_2 ;if done
      ret ;return
d8u_2: rol drem8u ;shift dividend into remainder
      sub drem8u,dv8u ;remainder = remainder - divisor
      brcc d8u_3 ;if result negative
      add drem8u,dv8u ;restore remainder
      clc ;clear carry to be shifted into result
      rjmp d8u_1 ;else
```

```
d8u_3:  sec                ;set carry to be shifted into result
        rjmp            d8u_1
```