

bit 4 u variabli test postavljati na 1
#define testbit(var, bit) ((var) & (1<<(bit)))
if (testbit(1<<4)) {

}

Elma Šeremet, 18318

1.2. Prekid na ulaznu i silaznu ivicu
na pinu RBO.

Da bi se omogućio prekid na silaznu
ivicu signala na ulazu RBO potrebno
je postaviti slj.

OPTION - REG bits, INTEDG = 0. → prekid na silaznu se odabire

INTF = 0 → briše se flag INTF

INTE = 1 → aktivira se prekid na RBO

GIE = 1 → aktivira se sistem prekida

na ulaznu jedina izmjenjiva je
u prvoj naredbi gdje imamo
→ za ulaznu

OPTION - REG bits, INTEDG = 1

ostalo isto

1.3

Frekvencija = 32 MHz

BITNO!

brzina komunikacije = 1225ps

podešavaju se

BRG16=0

misu
jerprihvatljiv
8-bitnite register
vrijednosti
veće od 255ne
može
se
upisati

BRG16	BRGH	SPBRG	Zaokruživanje SPBRG	Greška (%)
0	0	4097.36	X X	
0	1	16391.44262	X X	
1	0	16391.44262	16391 16392	-0.0087999 -0.0027
1	1	65572.8	65572 65573	



više od
16 bita
pa ne može
mi to
u BRG16

za 0 0:

$$X = \frac{32 \cdot 10^6}{64 \cdot 122} - 1 = 4097.36$$

za 0 1:

$$X = \frac{32 \cdot 10^6}{16 \cdot 122} - 1 = 16391.44262$$

za 1 1:

$$X = \frac{32 \cdot 10^6}{4 \cdot 122} - 1 = 65572.77049$$

Greške:

16391
16392

$$\frac{32 \cdot 10^6}{16 \cdot 16392} = 122.0107369$$

$$\frac{122}{122.0107369} = 0.999912 \quad / -1$$

$$- 0.000087999 \quad \%$$

$$= - 0.0087999 \%$$

$$\frac{32 \cdot 10^6}{16 \cdot 16393} = 122.0032941$$

$$\frac{122}{122.0032941} = 0.999972999 \quad / -1$$

$$- 0.000027 \quad \%$$

$$= - 0.0027 \%$$

Minimalna vrijedost greške se
postiže za BRG16=1;

BRGH=0

kada se upiše
16392

Kod za inic. :

BRGH = 0
BRG16 = 1 } brzina 1225

SPBRGH = 0x40

SPBRGL = 0x08

SYNC = 0

SPEN = 1

CPEN = 1

TXEN = 1

SP12C

→ asinkrona kom.

→ ser. port ukłj.

→ ukłj. přijemnik
ukłj. přijemnik

1C3B2

= 0x4008

② Ideja:

Napraviti prekid tajmera svako ~~0.1ms~~
0.1ms, jer je najkraće trajanje impulsa
1ms. U prekidačkoj rutini je potrebno
onda proveravati da li je

pim RBG na visokom ili
niskom naponskom nivou. Ako je
visoki, počinje merenje vremena
trajanja impulsa, pri čemu se
dodaje svaki prekid tajmera

0.1ms tj. cmo ~~1~~ vrijeme = 0.001

To je potrebno raditi sve dok je

$RBG = 1$, u suprotnom, prije se
vršilo merenje nivoa, te merenje staje
te je potrebno ispisati taj rez
u BCD formatu svakih 0.1ms.

```
void main (void) {
```

```
    init_pins();  
    init_timer();
```

```
    while(1);  
}
```

```
init_timer() {
```

```
    THROCS=0;
```

```
    PSA=1 ←
```

```
    TMROIF=0;
```

```
    TMROIE=1;
```

```
    GIE=1;
```

```
    TMRO=6;
```

```
}
```

✓

$pr = 1 \cdot 4$

$vrijeme = (0.5 * pr) * (256 - TMRO)$

$vrijeme = (0.5 * 4) * (256 - TMRO)$

$vrijeme = 2 * 250$

$vrijeme = 500 = 0.5ms$

$pr = 1 \cdot 4$


```
void __interrupt isr() {
```

```
    if (TMROIE && TMROIF) {
```

```
        if (!stajje && RBO) {
```

```
            stajje = 1;
```

```
        }
```

```
    else if (stajje && RBO) {
```

```
        vrijeme += 0.0001;
```

```
    }  
    TMRO = 6;
```

```
    else if (stajje && !RBO) {
```

```
        stajje = 0;
```

```
        PORTD = to_bcd(vrijeme);
```

```
        vrijeme = 0;
```

```
        TMRO = 0;
```

```
    }  
    TMRO = 6;  
    TMROIF = 0;
```

```
}
```

```
}
```



```
#include <xc.h>
```

```
double vrijeme=0, stage=0;
```

```
void init_pins() {
```

```
    TRISB = 0x00;
```

```
    ANSELB = 0x00;
```

```
    TRISD = 0xFF;
```

```
    ANSELD = 0x00;
```

```
}
```

```
char to_bed (double bimar) {
```

```
    char d, j, bed;
```

```
    j = bimar;
```

```
    d = 0
```

```
    while (j >= 10) {
```

```
        j = 10;
```

```
        d++;
```

```
    }
```

```
    d *= 16;
```

```
    bed = d/j;
```

```
    return bed;
```

```
}
```

Ulazi:

P₁ - analogni ulaz

P₂ - analogni ulaz

M - digitalni izlaz (motor)

V - digitalni izlaz (ventil)

10 digitalnih izlaza \Rightarrow 10 led dioda

Šema:

