

MIKROPROCESORIAI IR JŲ PROGRAMAVIMAS

LABORATORINIS DARBAS NR. 2

MIKROVALDIKLIO PERTRAUKIMAI IR LAIKMAČIAI

DARBO TIKSLAS: Išsiaiškinti kaip yra valdomi ir vykdomi mikrovaldiklio PIC16F1518 pertraukimai. Įsisavinti specialiųjų registrų *INTCON*, *PIE1*, *PIE2*, *PIR1*, *PIR2*, *IOCBP*, *IOCBN* bei *OPTION*, *TMR0*, *TMR1*, *TMR2* veikimą.

DARBO UŽDUOTIS: Pagal savo eilės Nr. sudaryti ir sujungti schemą bei užprogramuoti mikrovaldiklį taip, kad, nuspaudus mygtuką, šviesos diodas nedelsiant užsidegtų, o jei jis šviečia - užgestų. Programa tuo pačiu metu turi valdyti ir 7 segmentų indikatorius – variantų lentelėje nustatytu periodu T paeiliui rodyti užduotyje nurodytus simbolius.

ATASKAITOS TURINYS:

Darbo tikslas;
Varianto duomenys;
Laboratorinio darbo schema pagal užduoties variantą;
Programos tekstas su komentarais (pagal užduoties variantą);
Programos algoritmas pagal užduoties variantą;
Išvados.

VARIANTAI PAGAL GRUPĖS SĄRAŠĄ:

Nr.	7 segm. indikatorius	Mygtukas	LED	T (ms)	simboliai
1.	PORTB.x	PORTC.0	PORTC.1	700	1,2,3,4,5
2.	PORTB.x	PORTC.1	PORTC.2	750	4,3,2,1,0
3.	PORTB.x	PORTC.2	PORTC.3	800	0,2,4,6,8
4.	PORTB.x	PORTC.3	PORTC.4	850	A,b,C,d,E
5.	PORTB.x	PORTC.4	PORTC.5	900	5,4,3,2,1
6.	PORTB.x	PORTC.5	PORTC.6	950	4,3,2,1,0
7.	PORTB.x	PORTC.6	PORTC.7	1000	b,C,d,E,F
8.	PORTB.x	PORTC.7	PORTC.0	1050	A,C,E,F,0
9.	PORTB.x	PORTC.0	PORTC.1	1100	7,8,9,0,A
10.	PORTB.x	PORTC.1	PORTC.2	1150	A,0,1,2,3
11.	PORTB.x	PORTC.2	PORTC.3	1200	3,4,5,6,7
12.	PORTB.x	PORTC.3	PORTC.4	1250	8,7,6,5,4
13.	PORTB.x	PORTC.4	PORTC.5	1300	E,r,r,o,r
14.	PORTB.x	PORTC.5	PORTC.6	1350	5,u,p,E,r
15.	PORTB.x	PORTC.6	PORTC.7	1400	u,o,5,1,5
16.	PORTB.x	PORTC.7	PORTC.0	1450	d,r,E,1,E
17.	PORTB.x	PORTC.0	PORTC.1	1500	F,o,r,p,o
18.	PORTB.x	PORTC.1	PORTC.2	1550	I,A,E,1,0
19.	PORTB.x	PORTC.2	PORTC.3	1600	A,E,1,1,2
20.	PORTB.x	PORTC.3	PORTC.4	1650	I,A,E,1,1
21.	PORTB.x	PORTC.4	PORTC.5	1700	I,A,E,1,2
22.	PORTB.x	PORTC.5	PORTC.6	1750	I,A,E,1,3
23.	PORTB.x	PORTC.6	PORTC.7	1800	d,A,E,1,1
24.	PORTB.x	PORTC.7	PORTC.0	1850	0,A,E,1,2
25.	PORTB.x	PORTC.0	PORTC.1	1900	B,A,E,1,3
26.	PORTB.x	PORTC.1	PORTC.2	1950	D,A,E,1,4
27.	PORTB.x	PORTC.2	PORTC.3	2000	p,o,r,A,5
28.	PORTB.x	PORTC.3	PORTC.4	2050	1,3,5,7,9
29.	PORTB.x	PORTC.4	PORTC.5	2100	0,5,9,b,F
30.	PORTB.x	PORTC.5	PORTC.6	2150	b,A,b,A,0
31.	PORTB.x	PORTC.6	PORTC.7	2200	1,A,b,A,5
32.	PORTB.x	PORTC.7	PORTC.0	2250	5,E,1,1,E

**ŠIAULIŲ VALSTYBINĖ KOLEGIJA
VERSLO IR TECHNOLOGIJŲ FAKULTETAS
TRANSPORTO INŽINERIJOS KATEDRA**

SKAIČIUODAMI LAIKĄ NAUDOKITE TOKIĄ FORMULĘ:

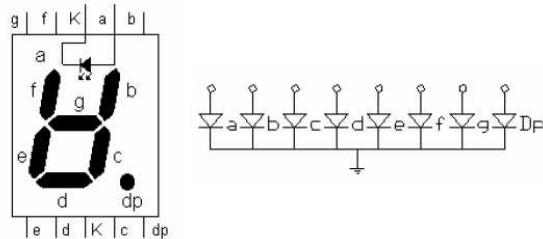
$$T = (255 - TMR0_0) \cdot d \cdot k \text{ (}\mu\text{s)}$$

kur: $TMR0_0$ – pradinė $TMR0$ reikšmė,

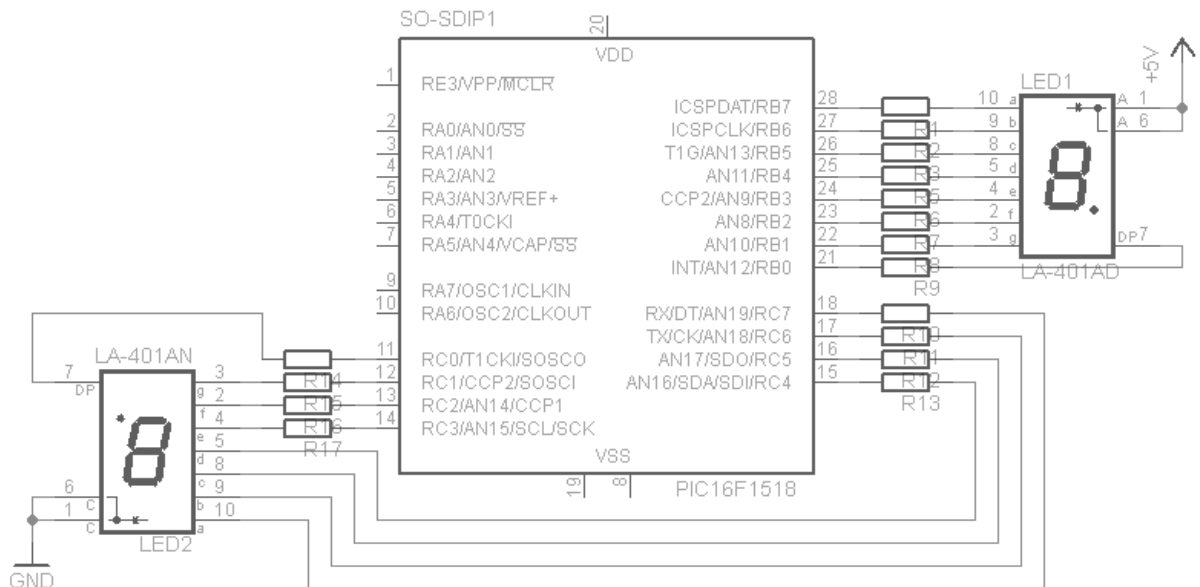
d – daliklio vertė,

k – papildomas kintamasis.

BENDRO KATODO 7 SEGMENTŲ LED INDIKATORIUS:



SCHEMOTECHNINIŲ SPRENDIMŲ PAVYZDŽIAI:



PROGRAMOS KODO PAVYZDŽIAI:

1. PVZ. Pertraukimas įvyks pasikeitus $PORTB.4$ lygiui. Įvykus pertraukimui invertuojamas $PORTC$.

<...>

```
void interrupt isr (void){
    PORTC = ~PORTC; // Invertuojamas PORTC
    IOCIF=0; // Išvaloma PORTB skilčių pasikeitimo pertraukimo IOCIF vėliavėlė
}
```

```
int main() {
    ANSELB=0; //visi PORTB išvadai- skaitmeniniai
    TRISB=0b00010000; //PORTB.4 - įėjimas, kiti išvadai - išėjimai
    TRISC=0; //PORTC – visi išvadai– išėjimai
```

$PORTC=0$; //visi $PORTC$ išvadai - sujungti su GND

$IOCIE = 1$; //Aktyvuojami pertraukimai, kurie įvyks pasikeitus nors vienai iš $PORTB$ skilčių lygių

$IOCBP4 = 1$; //Aktyvuojami pertraukimai, kurie įvyks pasikeitus $PORTB.4$ skilties lygiui

$GIE = 1$; //Aktyvuojami pertraukimai

```
while(1){
}
}
```

**ŠIAULIŲ VALSTYBINĖ KOLEGIJA
VERSLO IR TECHNOLOGIJŲ FAKULTETAS
TRANSPORTO INŽINERIJOS KATEDRA**

2. PVZ. TMR0 didinamas nuo išorinio signalo esančio RA4/T0CK1 išvade. Didinimas vyksta kas ketvirtą impulsą, nuo krentančio fronto. TMR0 reikšmė atvaizduojama PORTB.

<...>

```
void main() {
    ANSELA = 0;
    ANSELB = 0; // Visi I/O išvadai yra skaitmeniniai
    TRISB = 0; // PORTB skirtas išėjimui
    PORTB = 0x00; // Inicializuojamas PORTB
    T0CS = 1; // TMR0 didinimo šaltinis RA4/T0CK1 išvadas
    T0SE = 1; // Didinimas vyks nuo krentančio fronto (1 -> 0)
    PSA = 0; // Daliklis priskirtas TMR0
    PS2 = 0; // Daliklio vertė 1:4
    PS1 = 0;
    PS0 = 1;
    TMR0 = 0; // Pradinė TMR0 reikšmė lygi nuliui
    while(1){
        PORTB = TMR0; // Į PORTB išvedama TMR0 reikšmė
    }
}
```

3 PVZ. Daliklio vertė nustatyta 1:32, t.y. TMR0 reikšmė bus didinama vienetu kas 32μs. Pradinė TMR0 reikšmė yra 96. Vadinasi pertraukimas, persipildžius TMR0 reikšmei įvyks kas $(255 - 96) \cdot 32 \mu s = 5088 \mu s$. Kintamojo k reikšmė didinama kiekvieno pertraukimo metu. Kai ši reikšmė tampa lygi 196 gaunamas laiko intervalas $5088 \mu s \cdot 196 = 1\,002\,336 \mu s \approx 1s$.

<...>

unsigned k; //sukuriamas globalus kintamasis k

```
void interrupt isr (void) { //pertraukimo paprogramė
    k=k+1; //su kiekvienu pertraukimu k reikšmė padidinama vienetu
    TMR0 = 96; // Atstatoma pradinė TMR0 reikšmė
    INTCON = 0b10100000; // Nustatoma GIE, T0IE, išvaloma T0IF
    if (k == 196) { // Jeigu k reikšmė pasiekė 196, t.y. praėjo 1s
        PORTB = ~PORTB; // Invertuojama PORTB reikšmė
        k = 0; // skaičiuosime laiką iš naujo iki 1s.
    }
}
```

```
void main() {
    OPTION_REG = 0b000000100; //parenkame skaitikliui TMR0 vidinį taktinių impulsų šaltinį (OPTION_REG
//5 bitas T0CS →0), daliklį priskiriame skaitikliui TMR0 (OPTION_REG 3 bitas PSA →0), nustatome daliklio
//vertę 1:32 (OPTION_REG 2-0 bitai PS2 →1, PS1→0, PS0→0), t.y. TMR0 reikšmė bus padidinama vienetu
//kas 32 μs.
    OSCCON = 0b01101010; //Naudosime vidinį taktinio dažnio generatorių. Nustatome jo dažnį 4MHz.
    ANSELA = 0;
    ANSELB = 0; // Visi I/O išvadai yra skaitmeniniai
    TRISB = 0; // PORTB skirtas išvedimui
    PORTB = 0xFF; //Į PORTB nusiunčiama 0xFF
    TMR0 = 96; // Nustatoma pradinė TMR0 reikšmė
    INTCON = 0b10100000; // Aktyvuojami pertraukimai
    k = 0; // Inicializuojamas k
    while(1);
}
```

4 PVZ.

char lentele[4]={0b11111100, 0b01100000, 0b11011010, 0b11110010}; //sukuriamas globalus masyvas, kurio kiekvienas narys atitinka savo eilės numerio 7 segmentų indikatorius kodą.

```
void main() {
    PORTB = lentele[2]; //Į PORTB nusiunčiama 0b11011010 – užsidega „2“
}
```