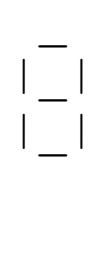
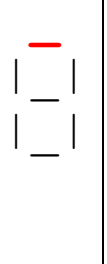
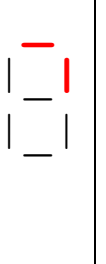
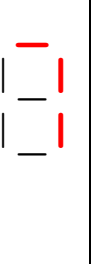
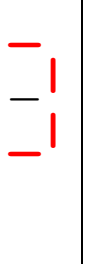
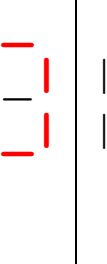
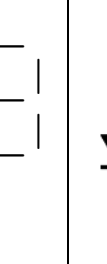
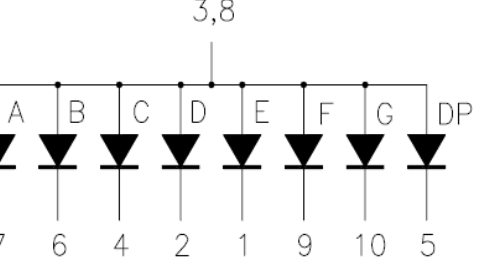


Subiecte.

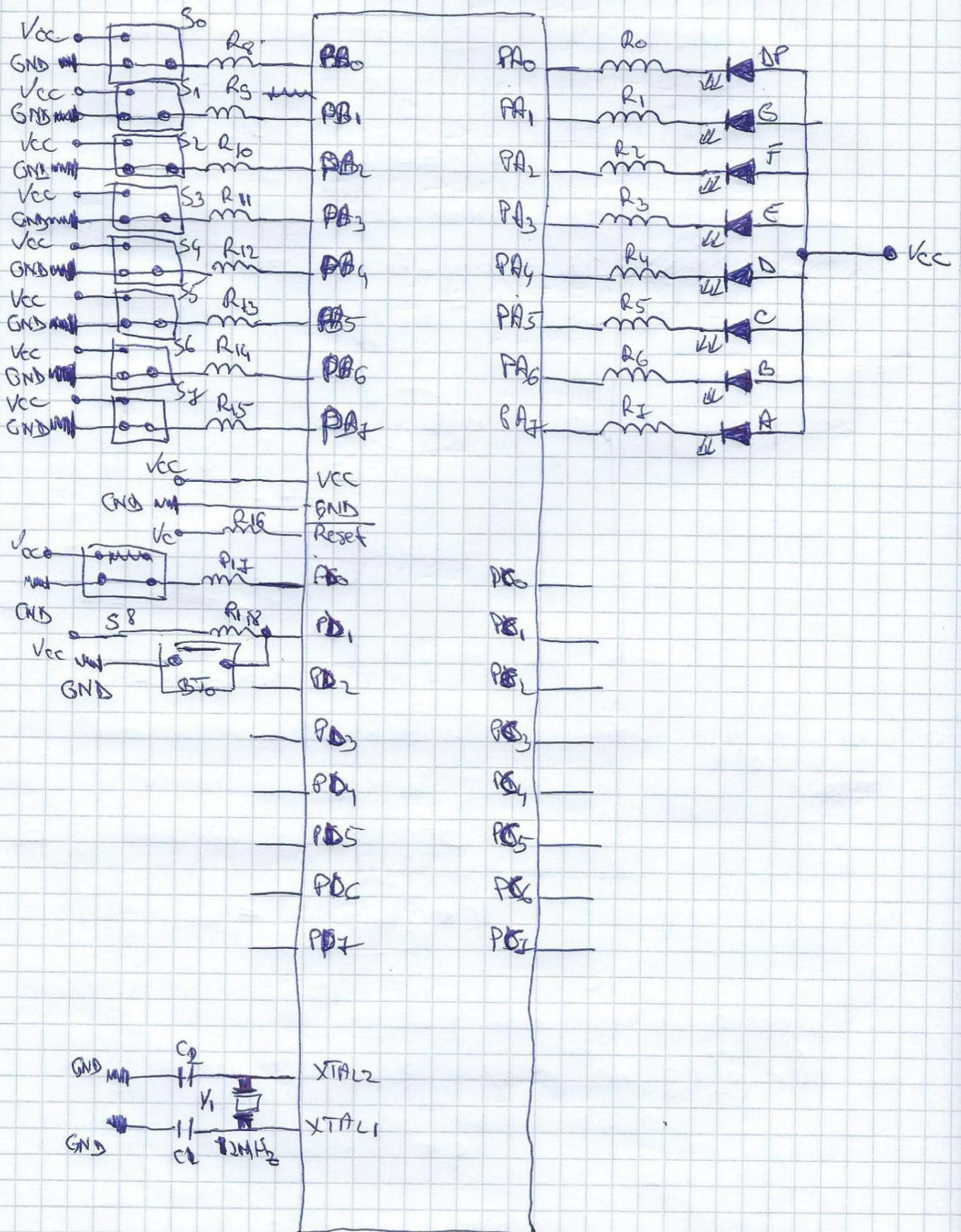
Să se proiecteze un microsistem bazat pe microcontrolerul ATmega16 care gestionează un sistem de votare. Ceasul microcontrolerul este generat cu un cuarț extern cu frecvența de **12MHz**. Sistemul de votare este format din un pushbuton, 9 comutatoare și un afișor cu 7 segmente tip LTS 546AG. Sistemul de votare funcționează după cum urmează:

1. Se apasă pusbutonul.
2. Imediat afișorul 7 segmente se stinge. Pentru a marca scurgerea timpului în care se poate vota, segmentele afișorului se aprind pe rând. Timpul în care se poate vota este de 30 de secunde. Ordinea în care se aprind segmentele este prezentat în tabelul următor:

							
t=0 s	t=5 s	t=10 s	t=15 s	t=20 s	t=25 s	t=30 s	LTS546AG

3. La $t=30$ s se sting toate segmentele și se numără câte comutatoare sunt pe poziția DA. Acest număr se notează cu n . O poziția a comutatorului este considerată DA iar cealaltă, NU. Este la latitudinea proiectantului care poziție este DA și care NU.
 4. La $t=35$ s se afișează n determinat la pasul 3.
- I.** Se cere să se prezinte schema microsistemului format din microcontrolerul ATmega16, comutatoare, pushbuton și afișorul 7 segmente. Se va specifica complet modul de conectare al tuturor componentelor. **1.5 puncte**
- II.** Să se scrie programul C care controlează sistemul de votare ce funcționează conform descrierii anterioare.
- a) Gestionarea push-butonului. **1.5 puncte**
 - b) Calculele necesare pentru utilizarea timerului și codul pentru gestiunea timpului. **2 puncte**
 - c) Gestiunea comutatoarelor. **2 puncte**
 - d) Afișarea trecerii timpului și a lui n . Configurația segmentelor se va calcula numai pentru cifrele 0, 1, 2 și 3. Pentru celelalte configurații se va folosi valoarea 55. **2 puncte**

④ Quant extern 12MHz; 1 pushbutton, 9 comutefone, 7 SEG



a) Gestionare push-buton.

Pentru gestionarea push butonului trebuie aminti în vedere problema contactului mecanic:

- * detectarea flautului colorătar
 - efectuarea a două citiri succesive ale pinului respectiv pentru a vedea dacă există un flaut colorătar, mai exact PB din figura mea. (acolo unde e legat pushbutonul)
- * debouncing-ul: ~~deja~~ curățarea impulsurilor parazite, adică să se pună condiția ca ale două citiri succesive (de unde vei să obții un 1 și 0) să se facă la un interval de timp mai mare ca durata maximă a instabilității.

În curs zice că durata maximă a instabilității este 10 ms și că execuția lui utilă (!) necesită 5 ms și că pt a face o altă nevoie de 3×5 (iteratii) = 15 > 10 ms.

Hai interzic să voi și dai-mi de știre că nu st nici eu prea sigur :)

pag 2 cod.

{ // de la main
// undeva in main

unsigned char cod-antior = 1;

unsigned char cod-nou = 1;

unsigned char loop-cut = 0;

DDRD = 0b00000000^{7 6 5 4 3 2 1 0}0;

↳ aici am pus automul iar pt ca eu tiint

un impuls la ~~pin~~ si se fie in modul IN sirul respectiv.

while (1) {

if (loop-cut == 3) {

loop-cut = 0; // if faci 0 ca dupa aceea sa
// faci iar 3 itari

cod-antior = cod-nou;

(10)

cod-nou = PINB;

if ((cod-antior & 1 << 1) == 1)

(cod-nou & 1 << 1) == 0

{

// inseamna ca ~~automul e gresit~~
sa gasim automul

}

}

loop-cut++;

}

} // de la main

5) $F = 12 \text{ MHz}$

Eu am ales Numărătorul 0 în modul CTC, deși nu sunt sigur dacă e bine.

Am ales așa pt că ts numărat 35 de sec, iar în curs știi că în modul CTC numărătorul e setat ca un numărător modulo N programabil.

La rezolvarea timpului am ales metoda 2 din curs.

Numărătorul 0 este pe 8 biți \Rightarrow nu poate număra decât 256 de impulsuri \Rightarrow ts să se facă o stare

Modul CTC $\Rightarrow T_{ciclului} = T_{imp_{CCT}} * N$

$T_{imp_{CCT}} * P$

$T_{imp_{ciclului}} = T_{imp_{CCT}} * P * N$

$1 \text{ sec} = K * T_{ciclului}$

\downarrow pt că ts o sec obținuta din mai multe cicluri

$1 \text{ sec} = K * T_{imp_{CCT}} * P * N \Rightarrow \frac{1}{T_{imp_{CCT}}} = K * P * N$

$\frac{1}{\frac{1}{f}}$
11
 12 MHz

$12 \text{ MHz} = \frac{12.000.000}{2^2 \cdot 3 \cdot 2^6 \cdot 5^6} = 2^8 \cdot 3 \cdot 5^6 = K * P * N$

$N = 5^3$

1, 8, 64, 256, 1024

$P = 256$

$N = 125$

$K = 375 (3 \cdot 5^3)$

$N =$ cel mai mare n care poate fi scris ≤ 256

$N = 125$

unsigned long sec
unsigned K;

TCCR0 = 0b00001100; // modul CTC

OCR0 = 125; // valoarea lui N

while(1) {

if (TIFR & 1 << OCR0) {

 // bitul n 1 din TIFR

 TIFR = TIFR | 1 << OCR0;

 K++;

 if (K == 375) {

 K = 0;

 sec++; // aici a trecut o secunda

 // aici se introduce codul cu pt

 // subprogramul c si d

 }

}

}

c) gestiunea comutatoarelor

Am pus 8 comutatoare pe Portul B și un comutator pe portul D (PDO)

Tb să am grija ca acestea să fie de tip IN

Deoarece pe portul A am pus ofisorul tb să fie de tip OUT

```
DDRB = 0b00000000;
```

```
DDRA = 0b11111111;
```

```
DDRD = 0b00000000; // asta îl mai declarai la (bta)
```

```
....
```

```
sec++;
```

```
if (sec == 5) { // dacă am trecut 5 secunde
```

```
    PORTA = 0b01111011;
    // squies
    // staus
```

```
if (sec == 10) {
```

```
    PORTA = 0000; // se face pt scara de
    // ofisori la t=10s
```

```
....
```

```
if (sec == 30) {
```

```
    // se sterg ofisori si se face subiectul d
```

```
    // staus
```

```
}
```


d)

....

```
if(sec == 30) {
```

```
    PORTA = 0x00000000;
```

```
    for(int i=0; i<8; i++){
```

```
        if(PINB & 1<<i)
```

```
            n++; //afie numărul de comuta  
                //sau oprișe
```

```
    }
```

```
    if (PINB & 1) u++;
```

```
    for(int i=0; i<3; i++){
```

```
        if(i==u){
```

```
            PORTA = pgm_read_byte(&seg7[u]);
```

```
        } else
```

```
        { PORTA = pgm_read_byte(&seg7[5]);
```

```
        }
```

pt ce este subliniat se ~~declara~~ definesc
variabile de main()

```
const char seg7[] PROGMEM = { 0x00000001,  
                                0x10011111,  
                                0x00100101,  
                                0x00001101,  
                                0x01010101.
```

```
};  
si se include <avr/pgmspace.h>;
```

Cum asa as fi facut eu problema, Nu stiu cât
de aproape sunt de adevar!

Meri reînviu de pus cap la cap fiecare subpunct.