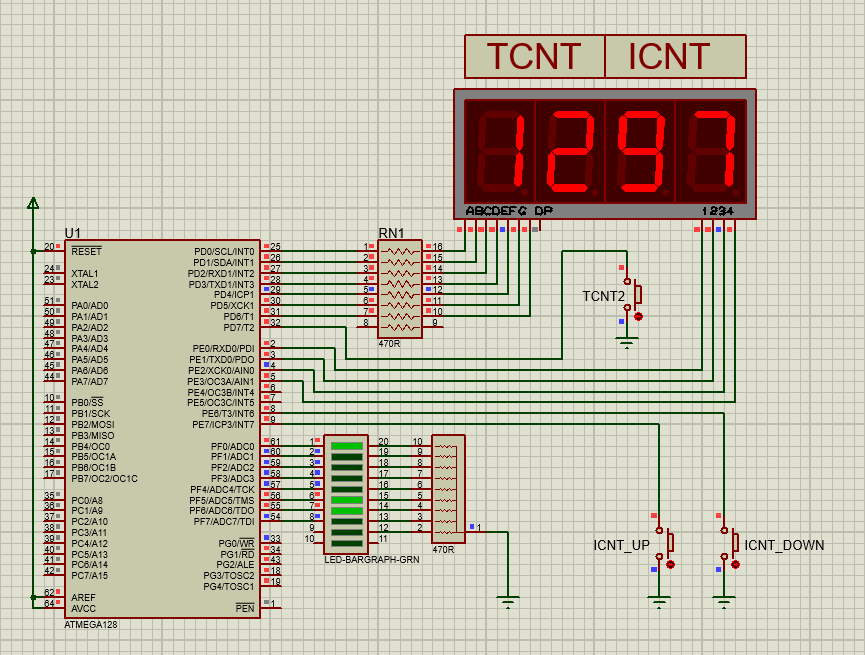
# Laboratorul nr. 3

## Obiective. În urma efectuării lucrării de laborator se învață:

* desenarea circuitelor utilizând programul Proteus;
* plasarea componentelor pe foaia de desenare;
* editarea componentelor (nume, valoare și simbol);
* trasarea liniilor de conexiuni;
* configurarea / accesarea portului D, E, F a microcontroler-lui ATmega128
* configurarea / utilizarea modulului numărător TCNT2
* configurarea / utilizarea întreruperilor externe INT6 / INT7

### **Tema 1**

Să se deseneze cu ajutorul programului *Proteus* circuitul din fig. L1-1.

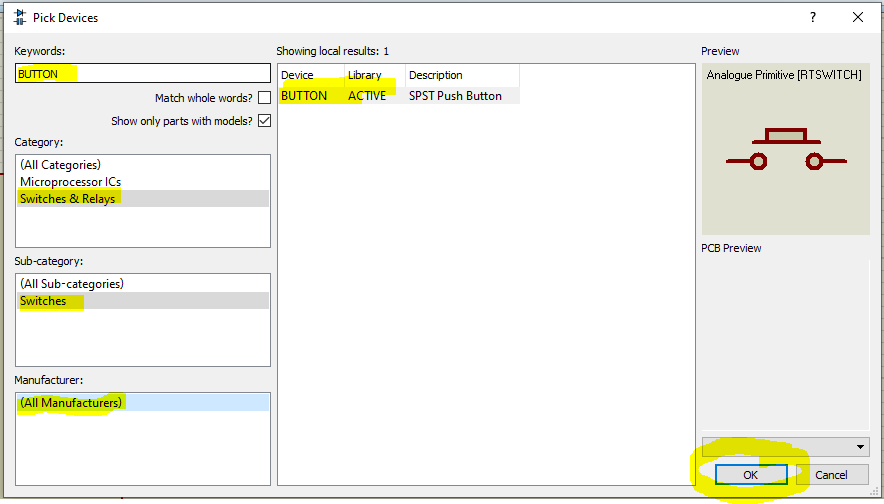


**Modul de lucru :**

**Pașii pentru lansarea programului *Proteus, crearea proiectului, plasarea componentelor pe planșa de lucru au fost prezentați in cadrul laboratorului cu numărul 1.***

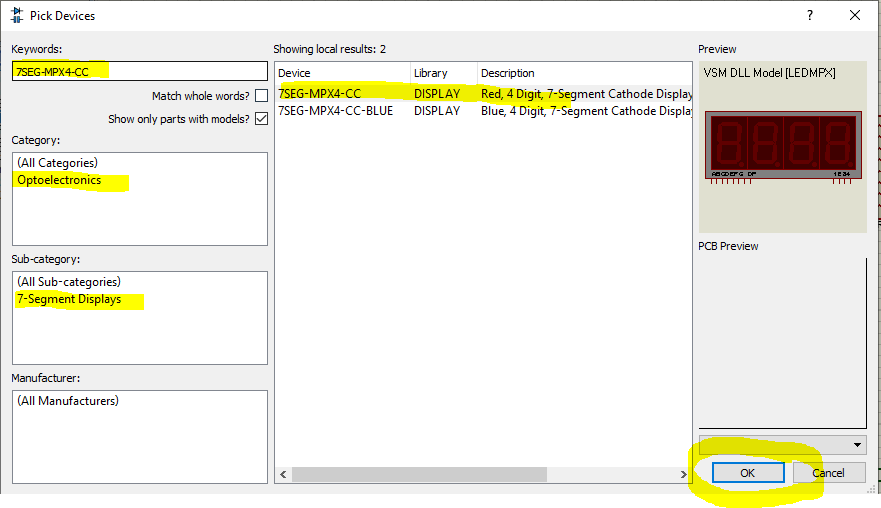
## Adăugarea butoanelor pe planșa de lucru:

ALT+L > Library > Pick Parts



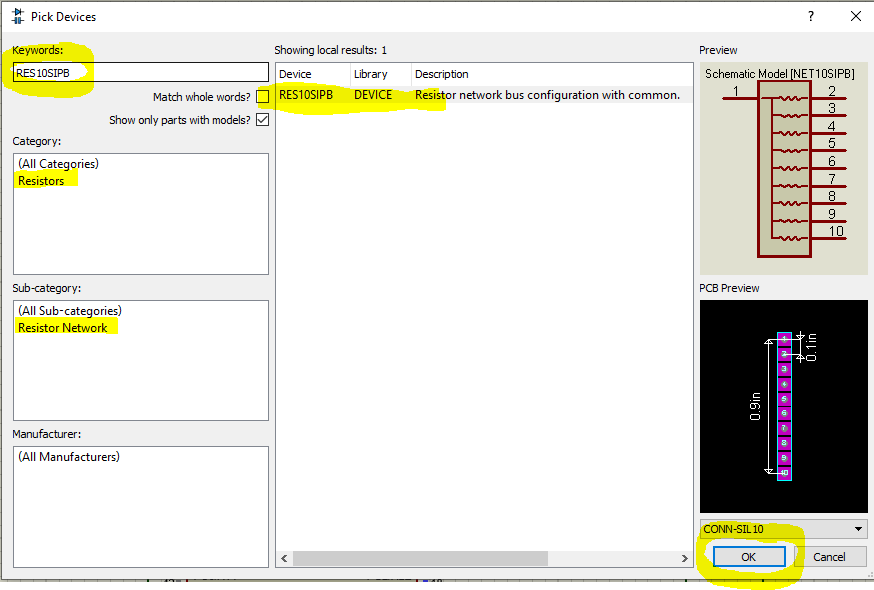
## Adăugarea afișorului cu 7 segmente si 4 digiți pe planșa de lucru:

ALT+L > Library > Pick Parts

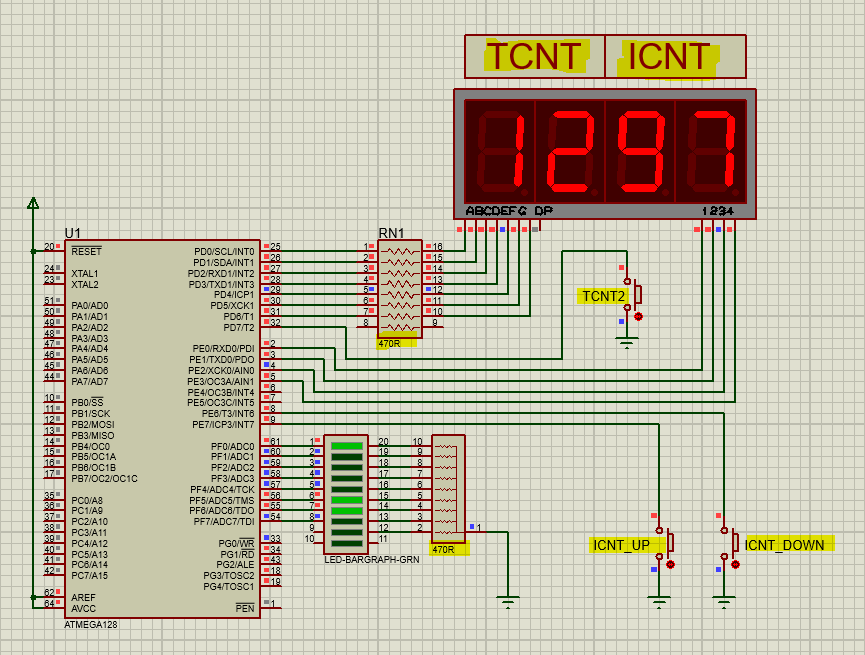


## Adăugarea rezistente de tip rețea pe planșa de lucru:

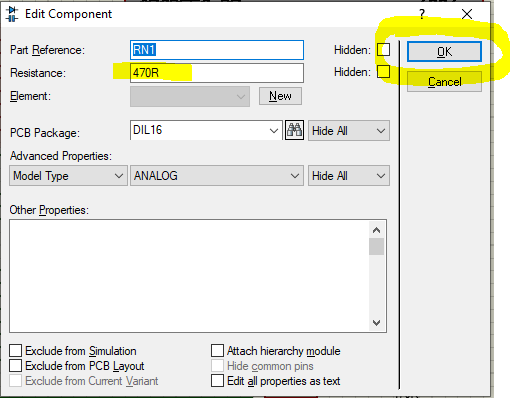
ALT+L > Library > Pick Parts

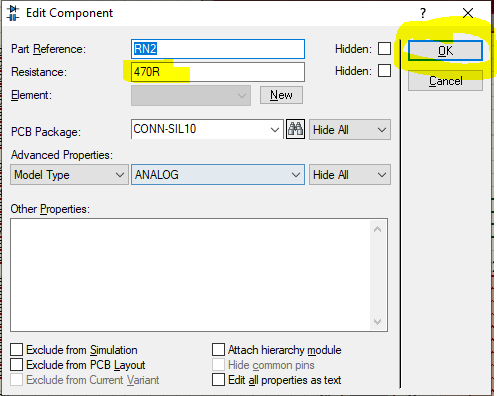


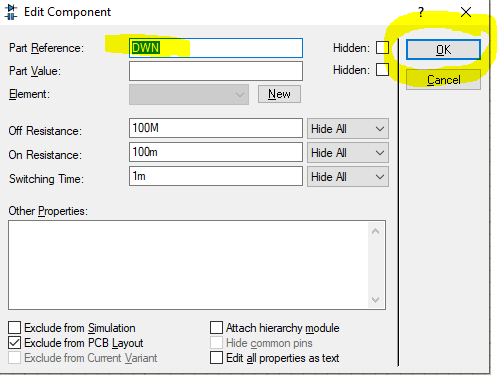
## Ajustarea valorii rezistentelor, redenumirea butoanelor:



Poziționam mouse-ul deasupra componentei pe care dorim sa o re-parametrizam, apăsam tastele CTRL+E.







Următorii pașii necesari au fost descriși in cadrul laboratorului cu numărul 1

## Interconectarea componentelor.

## Rularea / oprirea aplicației de tip firmware:

## Modificarea codului sursa (Source Code)

Pornind de la următorul cod sursa rezolvați următoarele doua teme:

#include <inttypes.h>

#include <avr/io.h>

#include <avr/interrupt.h>

#include <avr/sleep.h>

#include <math.h>

// NECESARA PENTRU FUNCTIA \_delay\_ms(10);

#include <util/delay.h>

// TESTAREA VALORII BITULUI DE PE POZITIA [7..0]

// RETURNEAZA 0 DACA BITUL DE PE POZITIA INDICATA ESTE 0

// RETURNEAZA O VALOARE DIFERITA DE 0 DACA BITUL DE PE POZITIA INDICATA ESTE 1

#define TBIT(var,pos) ((var) &   (1<<(pos)))

// SETAREA IN 1 A BITULUI DE PE POZITIA [7..0]

#define SBIT(var,pos) ((var) |=  (1<<(pos)))

// SETAREA IN 0 A BITULUI DE PE POZITIA [7..0]

#define CBIT(var,pos) ((var) &= ~(1<<(pos)))

// NEGAREA BITULUI DE PE POZITIA [7..0]

#define NBIT(var,pos) ((var) ^=  (1<<(pos)))

// CORESPONDENTA SEGMENTELOR DE LED ACTIVE CORESPUNZATOARE NUMARULUI AFISAT

//                    SGFEDCBA

unsigned char hx[10]={0b10111111,  // 0

                      0b10000110,  // 1

                      0b11011011,  // 2

                      0b11001111,  // 3

                      0b11100110,  // 4

                      0b11101101,  // 5

                      0b11111101,  // 6

                      0b10000111,  // 7

                      0b11111111,  // 8

                      0b11101111}; // 9

// VARIABILA ALOCATA AFISORUL CU 4x7SEGMENTE PENTRU AFISAREA VALORII DORITE

// EX. VALOAREA INITIAL 0034

signed int val\_7seg=0034;

// DEFINIREA VARIABILEI GLOBALE ICNT, VALOARE DE START 5

// VARIABILA ICNT ESTE UTILIZATA PENTRU IMPLEMENTAREA NUMARATORULUI

// PRIN INTERMEDIUL EVENIMENTELOR GENERATE DE INTRERUPERILE EXTERNE INT6 / INT7

unsigned char icnt=5;

// RUTINA EXEMPLU PENTRU CREAREA UNEI INTARZIERI PROGRAMABILE

void DLY(volatile long D)

{

    while(D--);

}

// RUTINA RESPONSABILA CU AFISAREA VALORII STOCATE IN VAL\_7SEG PE AFISORUL

// 4x7SEGMENTE

void fUpdate7Seg()

{

  // VARIABIL DE TIP STATIC UTILIZATA PENTRU SELECTAREA DIGITULUI DE AFISAT 0..3

  static char ds=0;

  // SALVAREA VALORII DE AFISAT INTR-UN REGISTRU TEMPORAR (DUBLA BUFFERARE)

  // EVITAREA SCHIMBARII VALORII DE AFISAT IN TIMPUL PROCESULUI DE AFISARE

  unsigned int tmp=val\_7seg;

  // ALOCARE A 4 ZONE DE MEMORIE NECESARE CELOR PATRU DIGITI AI AFISORULUI 4x7SEG

  unsigned char dg[4]={5,6,7,8};

  // RESETAREA VALORII DIGITULUI CURENT (TOATE SEGMENTELE DE LEDURI SUNT STINSE)

  PORTD=0b00000000;

  // CONVERSIA INTEGER LA FORMATUL 4x7SEG

  dg[3]=tmp%10;

  tmp/=10;

  dg[2]=tmp%10;

  tmp/=10;

  dg[1]=tmp%10;

  tmp/=10;

  dg[0]=tmp%10;

  // SELECTAREA DIGIT-LUI ACTIV

  ds=(ds+1)%4;

  PORTE=~(1<<ds);

  // ACTIVAREA LED-URILOR AFERENTE VALORI AFISATE

  PORTD=hx[dg[ds]];

  \_delay\_ms(10);

}

// = TEMA 3.B =

// MODIFICATI RUTINA ISR(INT6\_vect) ASTFEL INCAT:

// - VARIABILA ICNT SA DESCRIE FUNCTIONALITATEA UNUI NUMARATOR IN INEL

// - APASAREA BUTONULUI ICNT\_DOWN (PE6/T3/INT6, PINUL 8) VA DETERMINA DECREMENTAREA VARIABILEI ICNT

// RUTINA DE DESERVIRE A INTRERUPEREI EXTERNE INT6

// INT6\_vect - ADRESA ALOCATA VECTORULUI DE INTRERUPERE INT6

ISR(INT6\_vect)

{

// TODO ...

}

// = TEMA 3.C =

// MODIFICATI RUTINA ISR(INT7\_vect) ASTFEL INCAT:

// - VARIABILA ICNT SA DESCRIE FUNCTIONALITATEA UNUI NUMARATOR IN INEL

// - APASAREA BUTONULUI ICNT\_UP (PE7/ICP3/INT7, PINUL 9) VA DETERMINA INCREMENTAREA VARIABILEI ICNT

// RUTINA DE DESERVIRE A INTRERUPEREI EXTERNE INT7

// INT6\_vect - ADRESA ALOCATA VECTORULUI DE INTRERUPERE INT7

ISR(INT7\_vect)

{

// TODO ...

}

int main()

 {

  // VARIABILE LOCALE UTILIZATE PENTRU IDENTIFICAREA

  // NUMARATORULUI CARE A SUFERIT SCHIMBARI ALE VALORII

  unsigned char otcnt, oicnt;

  // DEZACTIVAREA TUTUROR INTRERUPERILOR PENTRU PROGRAMAREA INTERFETELOR

  // ( ARE ROLUL DE A PREVINE PROGRAMAREA DEFECTUOASA SAU PARTIALA A MODULELOR / INTERFETELOR )

  // SE REALIZEAZA PRIN SETAREA IN 0 A BITULUI 7 IN CADRUL REGISTRULUI SREG

  // A BITUL 7 DIN SREG ESTE RESPONSABIL CU ACTIVAREA / DEZACTIVAREA GLOBALA A INTRERUPERILOR

  cli();

  // SETAREA PORTULUI "E" (PORTE ALOCAT SELECTARI DIGITILOR DG3..DG0)

  // PINII PE3..PE0 CONFIGURATI CA PINI DE IESIRE

  DDRE=0x0F;

  // PORTE PE3..PE0 PINI SETATI IN 0 LOGIC -> 0V

  PORTE=0xC0;

  // CONFIGURAREA PORTULUI "D",

  // PINII PORTD[6..0] CONFIGURATI CA PINI DE IESIRE

  //(PORTD[6..0] ALOCAT CONTROLULUI SEGMENTELOR DE LED A..G)

  // PINUL PD7/T2 PIN DE INTRARE

  DDRD=0x7F;

  // PINII PD6..PD0 SETATI IN 0 LOGIC -> 0V (TOATE SEGMENTELE DE LEDURI STINSE)

  // ACTIVAREA REZISTENTEI DE PULL-UP PENTRU PINUL PD7/T2

  PORTD=0x80;

  // SETAREA PORTULUI "F" (PORTF ALOCAT BARGRAPHULUI)

  // PINII PF7..PF0 CONFIGURATI CA PINI DE IESIRE

  DDRF=0xFF;

  PORTF=0xAA;

  // = TEMA2 =

  // IDENTIFICATI SI REPROGRAMATI REGISTRII MODULULUI TIMER / COUNTER2 (VEZI TIMER2.PDF)

  // ASTFEL INCAT:

  // - NUMARATORUL (TCNT2) SA NUMERE CRESCATOR IN INEL DE LA 0 PANA LA VALOAREA 99.

  // - NUMARATORUL SA NUMERE PE FRONTUL (POZITIV SAU NEGATIV) AL SEMNALULUI APLICAT PINULUI PD7/T2.

  // - SEMNALUL APLICAT PINULUI PD7/T2 VA FI GENERAT DE APASAREA BUTONULUI TCNT2.

// TODO ...

  TCNT2 = 0x00;

  OCR2  = 0x00;

  TCCR2 = 0x00;

  // = TEMA 3.A =

  // IDENTIFICATI REGISTRII AFERENTI INTRERUPERILOR EXTERNE INT6 / INT7 (VEZI EXTERNAL\_INTERUPTS.PDF)

  // REPROGRAMATI REGISTRII IDENTIFICATI ASTFEL INCAT:

  // - APASAREA BUTONULUI ICNT\_UP VA DETERMINA PE FRONT POZITIV GENERAREA INTRERUPERII INT7

  // - APASAREA BUTONULUI ICNT\_DOWN VA DETERMINA PE FRONT NEGATIV GENERAREA INTRERUPERII INT6

// TODO ...

  EICRB = 0x00;

  EIMSK = 0x00;

  // ACTIVAREA TUTUROR INTRERUPERILOR (SETAREA IN 1 IN CADRUL REGISTRULUI SREG

  // A BITULUI RESPONSABIL CU ACTIVAREA / DEZACTIVAREA GLOBALA A INTRERUPERILOR

  sei();

   // BUCLA INFINITA

   while (1)

    {

    // APELAREA RUTINEI PENTRU ACTUALIZAREA VALORII PE AFISORUL 4x7SEG

    // VALOAREA DE AFISAT SE AFLA IN STOCATA IN VARIABILA VAL\_7SEG

    fUpdate7Seg();

    // AFISAREA PE BARGRAPH IN FORMAT BINAR A VALORII NUMARATORULUI TCNT2

    // DOAR DACA ACESTA SI-A SCHIMBAT VALOAREA

    if(otcnt != TCNT2)

     {

       otcnt = TCNT2;

       PORTF = otcnt;

     }

    // AFISAREA PE BARGRAPH IN FORMAT BINAR A VALORII NUMARATORULUI ICNT

    // DOAR DACA ACESTA SI-A SCHIMBAT VALOAREA

    if(oicnt != icnt)

     {

       oicnt = icnt;

       PORTF = oicnt;

     }

    // AFISAREA PE AFISORUL CU 7 SEGMENTE A VALORII

    // CELOR DOUA NUMARATOARE

    // (PRIMI 2 DIGITI DIN PARTEA STANGA SUNT ALOCATI PENTRU TCNT)

    // (PRIMI 2 DIGITI DIN PARTEA DREAPTA SUNT REZERVATI PENTRU ICNT)

    val\_7seg = TCNT2\*100+icnt;

    }

 }

## Tema 2 (vezi codul sursa de mai sus, secțiunea // = TEMA 2 =  )

Identificați si reprogramați regiștrii modulului timer / counter2 (vezi TUNER2.pdf) astfel încât:

- numărătorul (TCNT2) sa numere crescător in inel de la 0 pana la valoarea 99.

- numărătorul sa numere pe frontul (pozitiv sau negativ) al semnalului aplicat pinului PD7/T2.

- semnalul aplicat pinului PD7/T2 va fi generat de apăsarea butonului TCNT2.

## Tema 3

TEMA 3.A (vezi codul sursa de mai sus, secțiunea // = TEMA 3.A =  )

Identificați regiștrii aferenți întreruperilor externe INT6 / INT7 (vezi ***external\_interupts.pdf***)

Reprogramați regiștrii identificați astfel încât:

- apăsarea butonului **ICNT\_UP** va determina pe front pozitiv generarea întreruperii INT7.

- apăsarea butonului **ICNT\_DOWN** va determina pe front negativ generarea întreruperii INT6.

TEMA 3.B ( vezi codul sursa de mai sus, secțiunea // = TEMA 3.B = )

Modificați rutina ISR(INT6\_vect) astfel încât:

- variabila **icnt** sa descrie funcționalitatea unui numărător in inel.

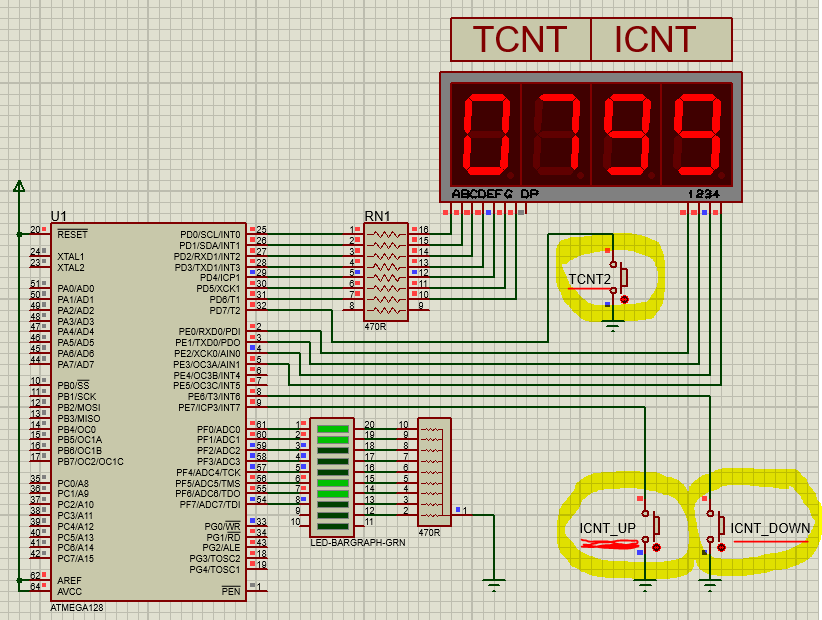
- apăsarea butonului ICNT\_DOWN (PE6/T3/INT6, pinul 8) va determina decrementarea variabilei **icnt**.

TEMA 3.C (vezi codul sursa de mai sus, secțiunea // = TEMA 3.C = )

Modificați rutina ISR(INT7\_vect) astfel încât:

- variabila **icnt** să descrie funcționalitatea unui numărător in inel

- apăsarea butonului **ICNT\_UP** (PE7/ICP3/INT7, pinul 9) va determina incrementarea variabilei **icnt.**



**IMPORTANT**

**Buna practică inginerească cere ca desenul să fie foarte clar,**

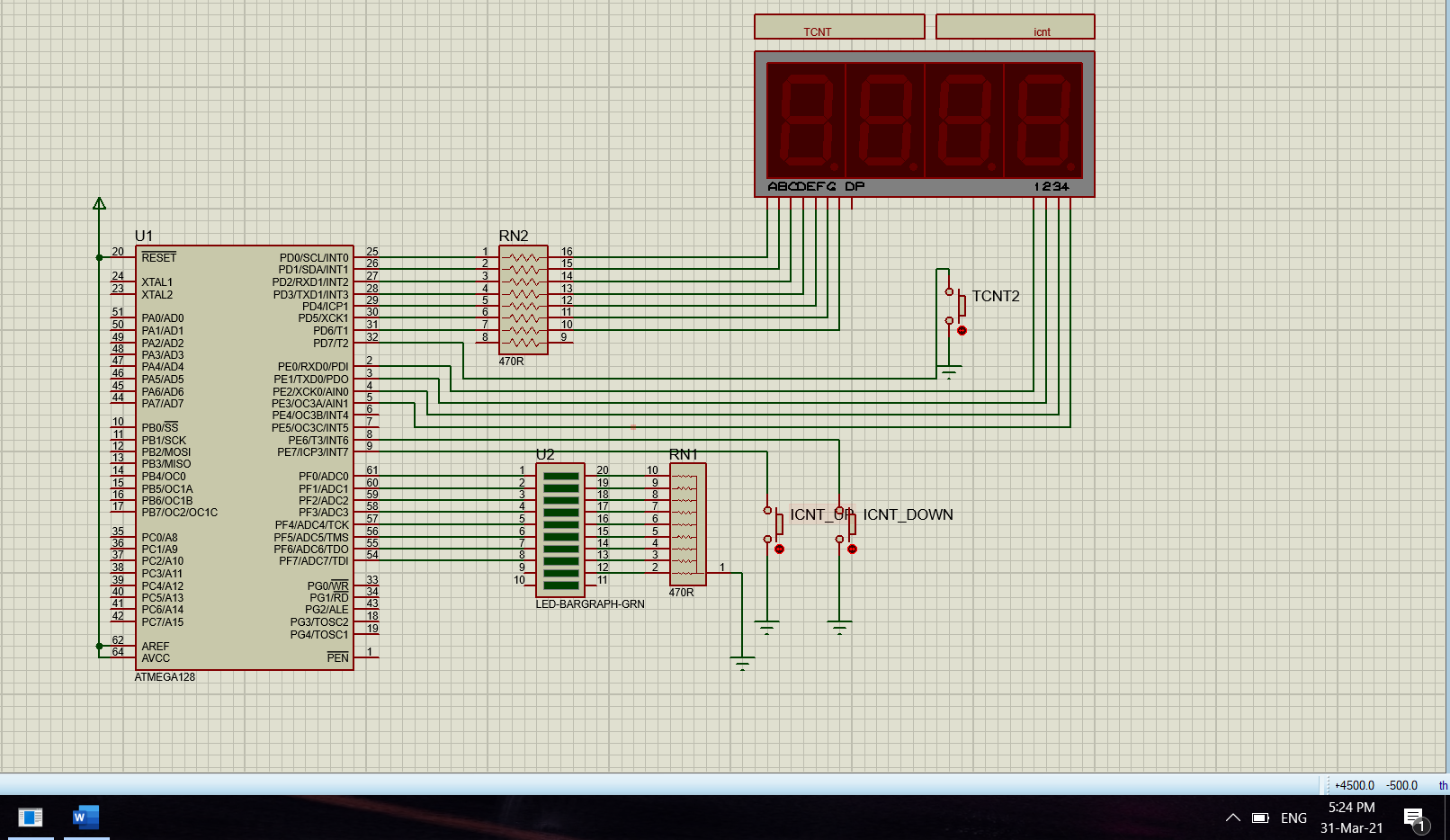
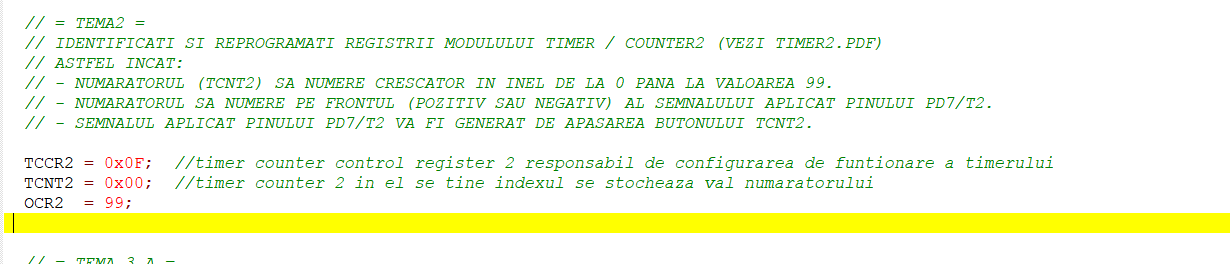
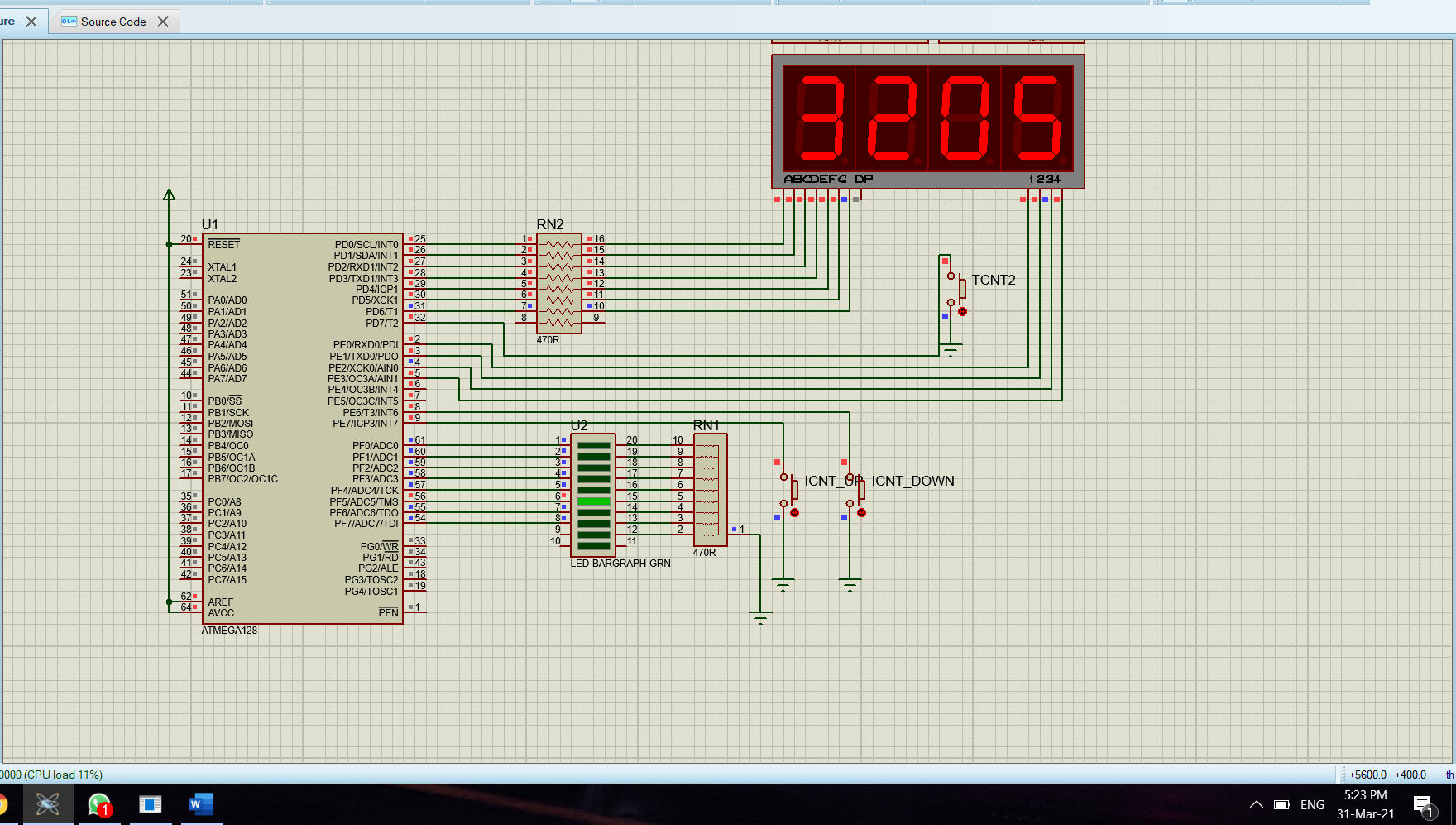
**să nu existe suprapuneri între înscrisuri şi elementele de circuit.**

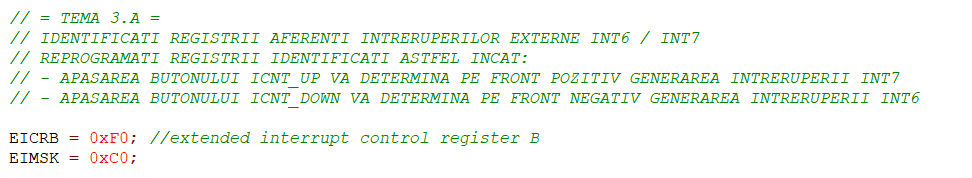
**Toate înscrisurile (nume, valori, parametri) se deplasează până când se văd clar atât componentele cât şi înscrisurile.**

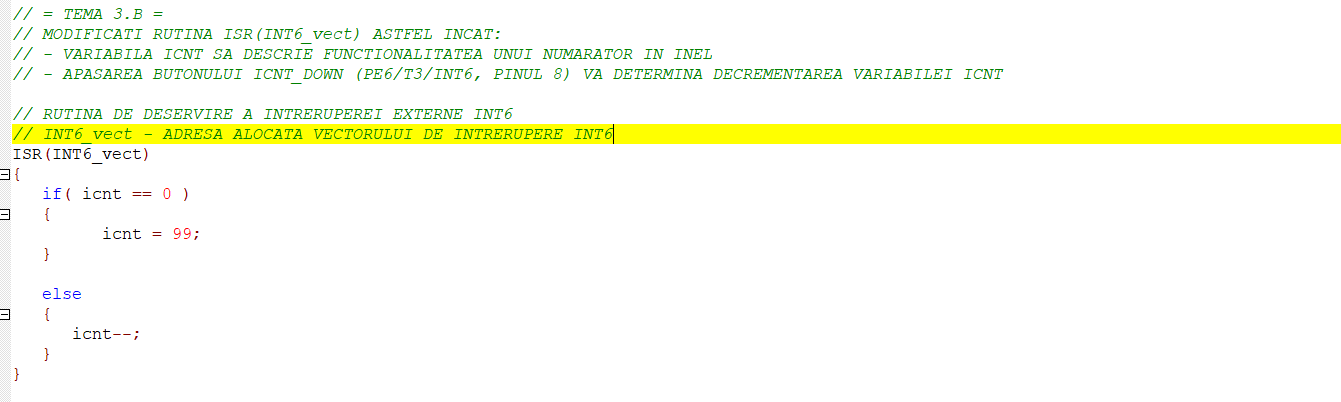
## Cerințe

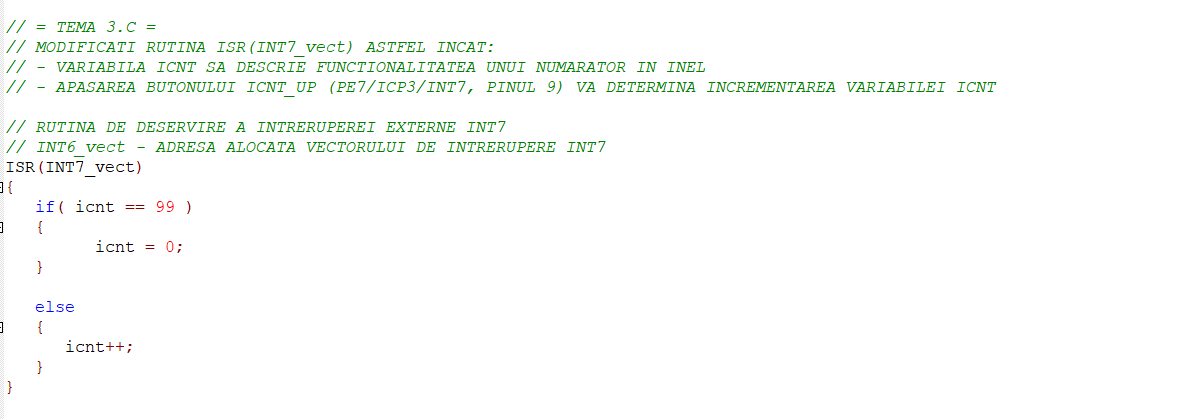
Lucrarea trebuie sa cuprindă rezolvarea temelor:

( **funcționalitatea descrisa mai sus poate fi vizualizata in videoclipul L3\_AC.mp4** )

* Rezolvarea temei 1;
* 
* Rezolvarea temei 2;
* 
* 
* Rezolvarea temei 3;





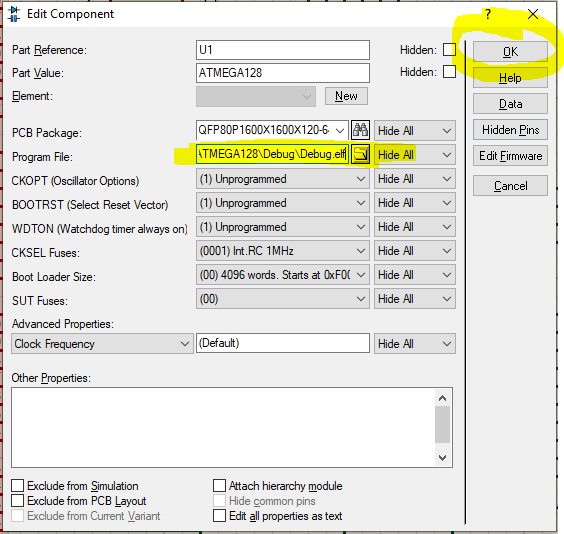


## CODUL HEX DEMO:

Încărcarea codului HEX DEMO presupune executarea următorilor pași:

* salvarea într-un fișier cu extensia hex a codului in format Intel HEX (<https://en.wikipedia.org/wiki/Intel_HEX>) de mai jos.
* încărcarea in microcontroler a codului hex ( poziționam mouse-ul deasupra microcontrolerului,

apăsam tastele CTRL+E, selectam fișierul cu extensia .hex salvat )



:100000000C9446000C9465000C9465000C946500FB

:100010000C9465000C9465000C9465000C94000130

:100020000C9417010C9465000C9465000C94650009

:100030000C9465000C9465000C9465000C946500AC

:100040000C9465000C9465000C9465000C9465009C

:100050000C9465000C9465000C9465000C9465008C

:100060000C9465000C9465000C9465000C9465007C

:100070000C9465000C9465000C9465000C9465006C

:100080000C9465000C9465000C94650011241FBE4F

:10009000CFEFD0E1DEBFCDBF11E0A0E0B1E0E6E4FC

:1000A000F3E000E00BBF02C007900D92AE30B10745

:1000B000D9F711E0AEE0B1E001C01D92AF30B10759

:1000C000E1F70E942D010C94A1010C940000DF9334

:1000D000CF9300D000D0CDB7DEB769837A838B830E

:1000E0009C8329813A814B815C81DA01C9010197A6

:1000F000A109B10989839A83AB83BC83211531059A

:100100004105510571F70F900F900F900F90CF910F

:10011000DF910895FF920F931F93DF93CF9300D049

:1001200000D0CDB7DEB720910A0130910B0102E378

:1001300010E0F8011082C9016AE070E00E947A01C3

:10014000F82EC9016AE070E00E947A01262F372F4D

:10015000862F972F6AE070E00E947A01F82F822F95

:10016000932F6AE070E00E947A01262F372F862FA6

:10017000972F6AE070E00E947A01E82F822F932F78

:100180006AE070E00E947A01CB016AE070E00E94B0

:100190007A01482F80910E01992787FD909501964D

:1001A00064E070E00E948E0180930E01992787FD24

:1001B000909521E030E0082E02C0220F331F0A94F0

:1001C000E2F7209523B94983EA83FB83FC82FE0191

:1001D000E80FF91F8181E0E0F1E0E80FF11D808177

:1001E000F801808384EC99E00197F1F70F900F906C

:1001F0000F900F90CF91DF911F910F91FF90089575

:100200001F920F920FB60F9211248F9380910C01C1

:10021000882321F483E680930C0103C081508093EE

:100220000C018F910F900FBE0F901F9018951F9289

:100230000F920FB60F9211248F9380910C01833689

:1002400019F410920C0103C08F5F80930C018F9101

:100250000F900FBE0F901F901895DF92EF92FF92B4

:100260000F931F93CF93DF93F8948FE082B990ECB4

:1002700093B98FE781BB12BA81EE809364008EE15F

:10028000809365008FEF809361008AEA809362001B

:10029000E5E4F0E0108283E084BD83E683BD8EE276

:1002A000808380EE8ABF99BF7894C4E4D0E00F2E9B

:1002B000F2E6EF2EFF24F02D0F2EF4E6DF2EF02DC8

:1002C0000E948A008881811719F01881F701108334

:1002D00020910C01021719F0F7012083022F888169

:1002E0008D9DC0011124820F911D90930B0180936D

:1002F0000A01E6CFAA1BBB1B51E107C0AA1FBB1F07

:10030000A617B70710F0A61BB70B881F991F5A95A1

:10031000A9F780959095BC01CD01089597FB092E12

:1003200007260AD077FD04D0E5DF06D000201AF4B6

:10033000709561957F4F0895F6F7909581959F4F41

:060340000895F894FFCFC0

:0E034600BF86DBCFE6EDFD87FFEF1C00050054

:00000001FF