Electronica Digitala
Counter cu afisaj 2 cifre cu 7-segment display

IONESCU MIHAI si TACHE ALEXANDRU-CĂTĂLIN

325CD

30.05.2021

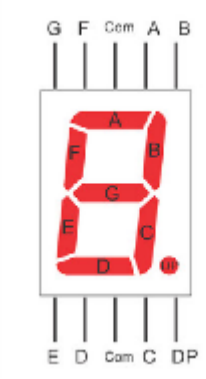
1 Introducerea proiectului

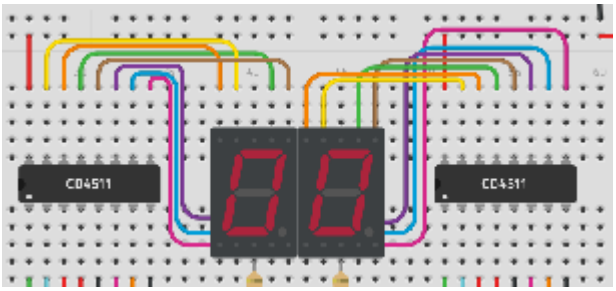
Ideea principala a proiectului este realizarea unui counter care incepe de la 0, iar cand ajunge la 99 reia numaratoarea. Numaratoarea este controlata de 2 butoane, unul de START si unul de STOP. Pentru a accentua numaratoarea de fiecare data cand counter-ul ajunge la o valoare rotunda (10, 20, 30, 40, 50, 60, 70, 80, 90) avem o alarma sonora implementata.

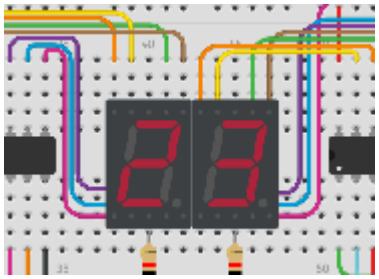
2 Schema circuitului

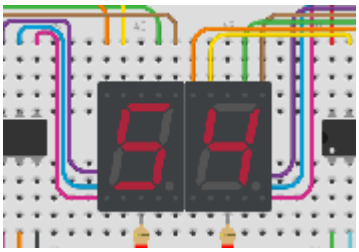
Componentele utilizate pentru implementarea circuitului sunt:

- O placuta Arduino de pe care am folosit pinii: 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 si cei de Ground si 5V pentru alimentarea breadboard-ului
- Un small breadboard, pe care am proiectat si interconectat elementele circuitului
- Doua elemente 7-segment display, pentru afisajul numerelor de la 0 la 99
- Doua aparate 7-segment decoder, conectate fiecare la cate un 7-segment display si la cate 4 pini de pe placa Arduino: 3, 4, 5 si 6 pentru cifra unitatilor, respectiv 8, 9, 10 si 11 pentru cifra zecilor. Fiecare placuta de 7-segment display are o conexiune cu 7-segment display prin pinii A, B, C, D, E, F, G fiecare reprezentand unul dintre cele 7 segmente ce poate fi setat pe HIGH, respectiv LOW la un moment de timp, dupa cum este ilustrat in pozele urmatoare:





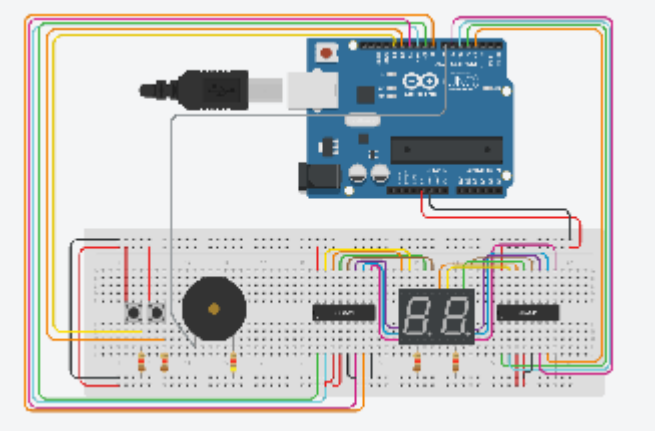




- Doua pushbuttons utilizate pentru a porni si a opri numaratoarea. Cel din dreapta este cel de start, iar cel din stanga de stop. Acestea sunt conectate la placuta Arduino la pinii 12 si 13.
- Un buzzer (piezoelectric) conectat la pinul 7 al placutei Arduino. Rolul acestuia este de a oferi un semnal sonor
- 5 rezistoare conectate la butoane, buzzer si 7-segment display. Rolul acestora este de a micsora cantitatea de curent survenita fiecarui element, astfel incat componentele sa nu fie suprasolicitate si afectate
- Firele de legatura a componentelor de pe breadboard intre ele, a breadboard-ului cu placa Arduino si a componentelor cu pinii asociati de pe Arduino. Singura regula general valabila sunt culorile rosu si negru asociate cu polii pozitiv(+) si negativ(-). In rest am incercat sa folosim toata paleta de culori pentru a fii cat mai exacti si a nu confunda firele. Exemplu: pentru buzzer am folosit gri, iar pinii la care sunt conectati 7-segment display-erele sunt portocaliu, verde, turcoaz si roz.

Un link catre proiectul in Tinkercad este acesta: Proiect-Tinkercad.

O imagine cu intreg circuitul este:



3 Implemetarea codului si funtionalitatea proiectului

Codul poate fii impartit impartit in 5 zone:

- Prima parte este cea de definire a pinilor folositi. Avem pini pentru butoanele de START si STOP. Pinul BUZZ reprezinta piezoelctric-ul. Pentru cele 2 display-uri avem pinii A1, B1, C1, D1 si A2, B2, C2, D2. Avem nevoie de 4 pini pentru ca numarul pe care il reprezentam intr-un display va fi oferit sub forma binara. Iar pentru a reprezenta cifrele de la 0 la 9 avem nevoie de 4 biti. Apoi definim 2 index cu valoarea 0, respectiv 1, care ajuta la implementarea butoanelor. Si initializam variabilele isrunning, value1, value2 utilizate in functia de loop si variabilele buttonpin, buttonstate, laststate, lastdebouncetime si debouncedelay pentru a folosi tehnica de debounce pentru butoane. De asemenea, in aceasta parte a programului avem si functia de setup unde am setat pinii de INPUT si de OUTPUT si am initializat variabilele value1, value2 si isrunning cu 0. O imagine a codului este aceasta:

```
#define A1      11
#define B1      10
#define C1      9
#define D1      8

#define A2      6
#define B2      5
#define C2      4
#define D2      3

#define START       12
#define STOP  13
#define BUZZ  7

#define STOP_INDEX  0
#define START_INDEX 1

// A B C D - reprez. in binar

int is_running;
int value1, value2;

// 0 = STOP
// 1 = START
int button_pin[2] = {STOP, START};
int button_state[2];
int last_state[2] = {LOW, LOW};
unsigned long last_debounce_time[2] = {0, 0};
unsigned long debounce_delay = 50;
```

```
void setup(void)
{
  Serial.begin(9600);
  pinMode(A1, OUTPUT);
  pinMode(B1, OUTPUT);
  pinMode(C1, OUTPUT);
  pinMode(D1, OUTPUT);
  pinMode(A2, OUTPUT);
  pinMode(B2, OUTPUT);
  pinMode(C2, OUTPUT);
  pinMode(D2, OUTPUT);
  pinMode(START, INPUT);
  pinMode(STOP, INPUT);

  is_running = 0;
  value1 = value2 = 0;
}
```

- A doua parte a codului reprezinta functiile ce se ocupa de afisarea cifrelor. Fiind reprezentate binar functia primeste ca argumente cei 4 pini si in caz ca pentru o anumita cifra bitul de la o anumita pozitie este 0, atunci functia seteaza pinul LOW, altfel il seteaza HIGH. Fiecare functie este denumita in conformitate cu valoarea pe care o afiseaza pe display. Acestea sunt pozele cu functiile:

```
// Function to print digits on the 7-segment display
void zero(int A, int B, int C, int D) {
  digitalWrite(A, LOW);
  digitalWrite(B, LOW);
  digitalWrite(C, LOW);
  digitalWrite(D, LOW);
}

void one(int A, int B, int C, int D) {
  digitalWrite(A, LOW);
  digitalWrite(B, LOW);
  digitalWrite(C, LOW);
  digitalWrite(D, HIGH);
}

void two(int A, int B, int C, int D) {
  digitalWrite(A, LOW);
  digitalWrite(B, LOW);
  digitalWrite(C, HIGH);
  digitalWrite(D, LOW);
}

void three(int A, int B, int C, int D) {
  digitalWrite(A, LOW);
  digitalWrite(B, LOW);
  digitalWrite(C, HIGH);
  digitalWrite(D, HIGH);
}

void four(int A, int B, int C, int D) {
  digitalWrite(A, LOW);
  digitalWrite(B, HIGH);
  digitalWrite(C, LOW);
  digitalWrite(D, LOW);
}
```

```
void five(int A, int B, int C, int D) {
  digitalWrite(A, LOW);
  digitalWrite(B, HIGH);
  digitalWrite(C, LOW);
  digitalWrite(D, HIGH);
}

void six(int A, int B, int C, int D) {
  digitalWrite(A, LOW);
  digitalWrite(B, HIGH);
  digitalWrite(C, HIGH);
  digitalWrite(D, LOW);
}

void seven(int A, int B, int C, int D) {
  digitalWrite(A, LOW);
  digitalWrite(B, HIGH);
  digitalWrite(C, HIGH);
  digitalWrite(D, HIGH);
}

void eight(int A, int B, int C, int D) {
  digitalWrite(A, HIGH);
  digitalWrite(B, LOW);
  digitalWrite(C, LOW);
  digitalWrite(D, LOW);
}

void nine(int A, int B, int C, int D) {
  digitalWrite(A, HIGH);
  digitalWrite(B, LOW);
  digitalWrite(C, LOW);
  digitalWrite(D, HIGH);
}
```

- Urmatoarea parte a codului o reprezinta implementarea functiei choosenumber. Aceasta primeste in variabila x, una dintre cele 2 valori(value1/value2), care desemneaza al catelea numar trebuie sa fie afisat. Functia apeleaza una din functiile de mai sus.

```
void choose_number(int x, int A, int B, int C, int D) {

  if (x == 0)
     zero(A,B,C,D);
   else if (x == 1)
     one(A,B,C,D);
   else if (x == 2)
     two(A,B,C,D);
   else if (x == 3)
     three(A,B,C,D);
   else if (x == 4)
     four(A,B,C,D);
   else if (x == 5)
     five(A,B,C,D);
   else if (x == 6)
     six(A,B,C,D);
   else if (x == 7)
     seven(A,B,C,D);
   else if (x == 8)
     eight(A,B,C,D);
   else if (x == 9)
     nine(A,B,C,D);
}
```

- Functia readbutton se ocupa cu realizarea de debounce astfel incat oprirea si pornirea counter-ului sa fie cat mai exacte, iar intarzierea sa fie micsorata.

```
void read_button(int id)
{
  // Citire valori butoane
  int reading = digitalRead(button_pin[id]);

  if (reading != last_state[id])
    last_debounce_time[id] = millis();

  if((millis() - last_debounce_time[id]) > debounce_delay)
  {
    if (reading != button_state[id])
    {
      button_state[id] = reading;

      if (button_state[id] == HIGH)
        is_running = id; // 0 for STOP, 1 for START
    }
  }

  last_state[id] = reading;
}
```

- Ultima parte contine functi loop. In aceasta am transformat o forma de for in for care afecta procesul intr-un if in if. Astfel la fiecare pas ne asiguram ca este pornit counter-ul, apoi verificam daca cifra unitatilor ajunge la 10, caz in care aceasta trebuie sa ia valoarea 0 si pornim buzzer-ul. La fel verificam si pentru cifra zecilor. Apelam functia choosenumber mai intai cu prima valoare si dupa cu value2. Astfel completam cele doua 7-sgment display. Incrementam valorile si realizam un delay in numaratoare.

```
void loop(void){
  read_button(START_INDEX);
  read_button(STOP_INDEX);

  if (is_running)
  {
    // Incrementari & Afisari cifre
    if (value2 == 10)
    {
      tone(BUZZ, 500, 500);

      value2 = 0;
      value1++;
      if (value1 == 10)
      {
        value1 = 0;
      }
      choose_number(value1, A1, B1, C1, D1);
    }

    choose_number(value2, A2, B2, C2, D2);
    value2++;

    delay(300);
  }
}
```

4 Bibliografie

Referintele ce ne-au ajutat in proiectarea circuitului sunt:

$\Longrightarrow$ laboratorul 4, aflat la adresa LABORATOR 4 [accesat ultima oara la data de 29 Mai 2021, ora 21:50].

$\Longrightarrow$ site-ul Debounce button [accesat ultima oara la data de 29 Mai 2021, ora 21:50].

$\Longrightarrow$ site-ul Two Digit Timer [accesat ultima oara la data de 29 Mai 2021, ora 21:50].