Proiectarea unui ceas de timp real cu afisaj 7 segmente

Colaboratori: Dumitru Andreea

Hamza Sebastian

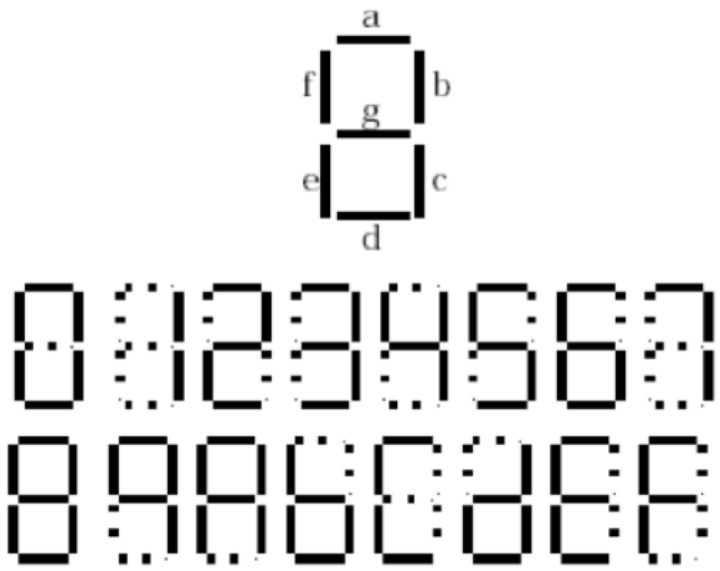
2021

Proiectarea unui ceas de timp real cu afisaj 7 segmente

1. Modul de functionare al display-ului

Afisajele cu 7 segmente sunt utilizate, de obicei, ca afisaje alfanumerice de catre sistemele logice si computerizate. Un afisaj cu 7 segmente este un aranjament de 7 LED-uri sub forma cifrei „8”, care poate fi utilizat pentru a afisa orice numar hexazecimal cuprins intre 0000 si 1111 prin iluminarea combinatiilor acestor LED-uri. Fiecare dintre cele 7 LED-uri se numeste segment, deoarece atunci cand este iluminat, segmentul face parte dintr-o cifra numerica. Un al optulea LED suplimentar este uneori folosit pentru indicarea unui punct zecimal.

Aceste segmente sunt etichetate de la „a” la „g” reprezentand fiecare LED individual. Prin setarea unui segment in starea „High” sau „Low” se genereaza un model de caractere dorit. Acest lucru ne permite sa afisam fiecare dintre cele zece cifre zecimale de la 0 la 9 pe acelasi afisaj cu 7 segmente.



A 7-segment display contains seven light emitting diodes (LEDs)

Emisia acestor fotoni are loc atunci cand jonctiunea diodei este polarizata de o tensiune externa care permite curentului sa circule peste jonctiunea sa. In electronica, acest proces poarta denumirea de electroluminiscenta.

Culoarea reala a luminii vizibile emisa de un LED variaza de la albastru la rosu, pana la portocaliu si este decisa de lungimea de unda spectrala a luminii emise. Aceasta este dependenta

de amestecul diferitelor impuritati adaugate materialelor semiconductoare utilizate pentru a o produce.

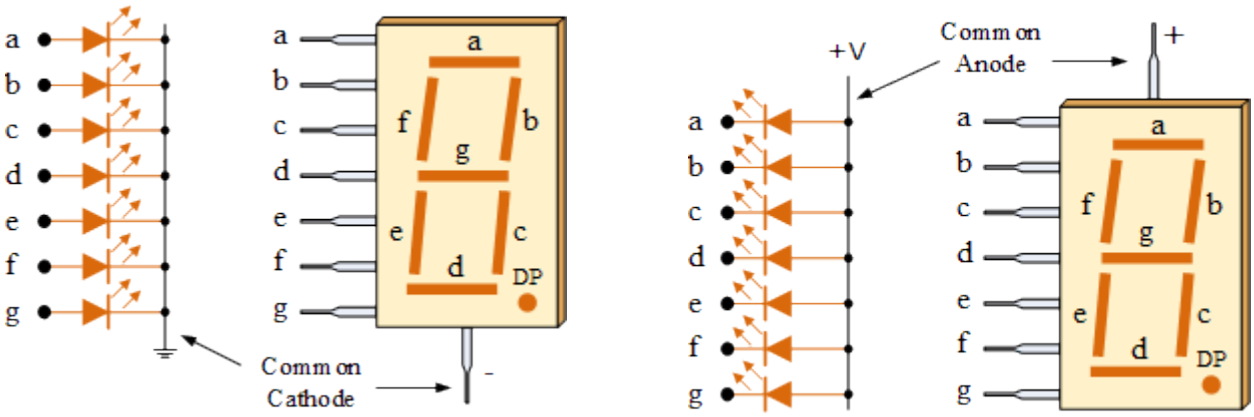
Diodele cu emisie de lumina au multe avantaje fata de becurile si lampile traditionale, principalele fiind dimensiunile lor mici, durata lunga de viata, culorile variate, accesibilitatea si disponibilitatea, precum si faptul ca sunt usor de utilizat. Acestea sunt deseori utilizate in combinatii de componente electronice si circuite digitale.

Principalul avantaj al diodelor emitatoare de lumina este acela ca, din cauza dimensiunilor mici, acestea pot fi conectate impreuna pe o suprafata mica si compacta, producand afisajul cu 7 segmente.

2. Catod comun (CC) vs. Anod comun (AC)

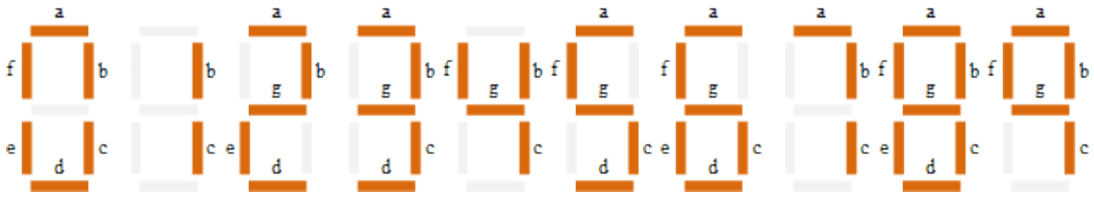
Pinul comun al afisajelor este de obicei utilizat pentru a identifica tipul de afisaj pe 7 segmente. Deoarece fiecare LED are doi pini de conectare, anod si catod, prin urmare exista doua tipuri de afisaj cu 7 segmente:

- anod comun – Afisajul cu 7 segmente are toti anozii legati impreuna la „1 logic”. Segmentele sunt iluminate prin aplicarea unui semnal „0 logic” / „Low” la bornele catodului.
- catod comun – Afisajul cu 7 segmente are toti catozii legati impreuna. Toate conexiunile catodice ale segmentelor LED sunt conectate impreuna la „0 logic” / GND. Segmentele sunt iluminate prin aplicarea semnalului „High” / „1 logic” la bornele anodului.



In general, afisajele anodice comune sunt mai utilizate deoarece multe circuite logice primesc mai mult curent decat pot consuma. De asemenea, un afisaj de tip catod comun nu este o inlocuire directa intr-un circuit pentru un afisaj de anod comun si invers, deoarece este la fel ca si conectarea LED-urilor in sens invers, iar emisia de lumina nu are loc.

3. Combinatii posibile – afisaj 7 segmente



Decimal Digit	Input lines				Output lines							Display pattern
	A	B	C	D	a	b	c	d	e	f	g	
0	0	0	0	0	1	1	1	1	1	1	0	
1	0	0	0	1	0	1	1	0	0	0	0	
2	0	0	1	0	1	1	0	1	1	0	1	
3	0	0	1	1	1	1	1	1	0	0	1	
4	0	1	0	0	0	1	1	0	0	1	1	
5	0	1	0	1	1	0	1	1	0	1	1	
6	0	1	1	0	1	0	1	1	1	1	1	
7	0	1	1	1	1	1	1	0	0	0	0	
8	1	0	0	0	1	1	1	1	1	1	1	
9	1	0	0	1	1	1	1	1	0	1	1	

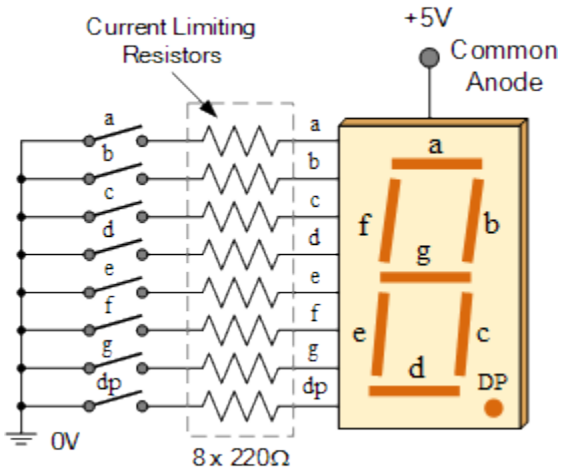
4. Consideratii electronice

Desi un afisaj cu 7 segmente poate fi considerat un singur display, acesta este totusi alcatuit din cele 7 LED-uri individuale. Aceste LED-uri necesita protectie impotriva supracurentului. LED-urile produc lumina numai atunci cand curentul circula direct prin ele. Astfel, cantitatea de lumina este proportionala cu curentul direct.

Acest lucru inseamna ca intensitatea luminii LED-urilor creste intr-o maniera aproximativ liniara cu un curent in crestere. Deci, acest curent trebuie sa fie controlat si limitat la o valoare sigura de catre un rezistor extern pentru a preveni deteriorarea segmentelor LED.

Caderea de tensiune pe un segment de LED rosu este foarte scazuta, la aproximativ 2-2.2V (LED-urile albastre si albe pot ajunge pana la 3.6V). Astfel, pentru a se lumina corect, segmentele LED trebuie conectate la o sursa de tensiune cu valoarea de tensiune mai mare decat cea prevazuta, impreuna cu o rezistenta in serie utilizata pentru a limita curentul la o valoare dorita.

De obicei, la un afisaj standard cu 7 segmente de culoare rosie, fiecare segment LED poate consuma aproximativ 15mA la iluminat corect. Deci, pe un circuit logic digital de 5V, valoarea rezistorului de limitare a curentului ar fi de aproximativ 200Ω.



5. Exemple de implementare

Poate ca afisajele cu 7 segmente nu par suficient de moderne, dar sunt cel mai practic mod de a afisa numerele. Sunt usor de utilizat, rentabile si foarte lizibile, atat in conditii de lumina limitata, cat si in lumina puternica a soarelui.

Un exemplu din lumea reala care foloseste un afisaj cu 7 segmente este celebrul ceas cu numaratoare inversa de la Cape Canaveral, Florida, pe care NASA l-a folosit pentru aterizarea Apollo.

Daca dorim sa proiectam un temporizator, contor sau ceas, vom avea nevoie de un afisaj cu 7 segmente din 4 cifre. Dar, un afisaj cu 4 cifre, cu 7 segmente, necesita de obicei 12 pini de conectare, ceea ce este destul de mult si nu lasa loc pentru alte module si senzori.

Pentru a evita multitudinea de cabluri si pentru a face posibila controlarea afisajului cu 7 segmente din 4 cifre vom utiliza modulul TM1637. Acest modul reduce pinii de conectare la doar patru. Doi pini sunt utilizati pentru conexiunile de alimentare, iar ceilalti doi pini sunt utilizati pentru controlul segmentelor.

6. Implementare

Pentru partea practica a proiectului am folosit un microcontroller NodeMCU care dispune de un modul de Wi-Fi (ESP8266), la care am conectat un display cu 7 segmente de tip TM1637.

Cand microcontrollerul NodeMCU este conectat la o sursa de curent prin portul micro-USB, acesta incepe procesul de conectare la reteaua de Wi-Fi ale caror date de autentificare sunt incluse in fisierul credentials.h. Odata ce acesta este conectat, se trimite un request catre un server public, care trimite inapoi timpul universal coordonat (UTC). La acesta se adauga 7200 de secunde (2 ore) pentru a obtine fusul orar al Romaniei. Mai departe, microcontrollerul trimite display-ului valoarea orei si a minutelor care este afisata.

7. Cod sursa (C)

```
//Programul incarcat pe NodeMCU

#include <NTPClient.h>
#include <ESP8266WiFi.h>
#include <WiFiUdp.h>
#include <TM1637Display.h>
#include "credentials.h"
#define CLK D2
#define DIO D3

TM1637Display display = TM1637Display(CLK, DIO);

const long utcOffsetInSeconds = 7200; //Offset Local Time ( RO +2:00 / 7200 sec)

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org", utcOffsetInSeconds);

void setup(){
  Serial.begin(57600);
  display.clear();

  WiFi.begin(user, password);

  while ( WiFi.status() != WL_CONNECTED ) {
    delay ( 500 );
    Serial.print ( "." );
  }
  timeClient.begin();
}

void loop() {
  int A,B;
  timeClient.update();
  display.setBrightness(7);
```

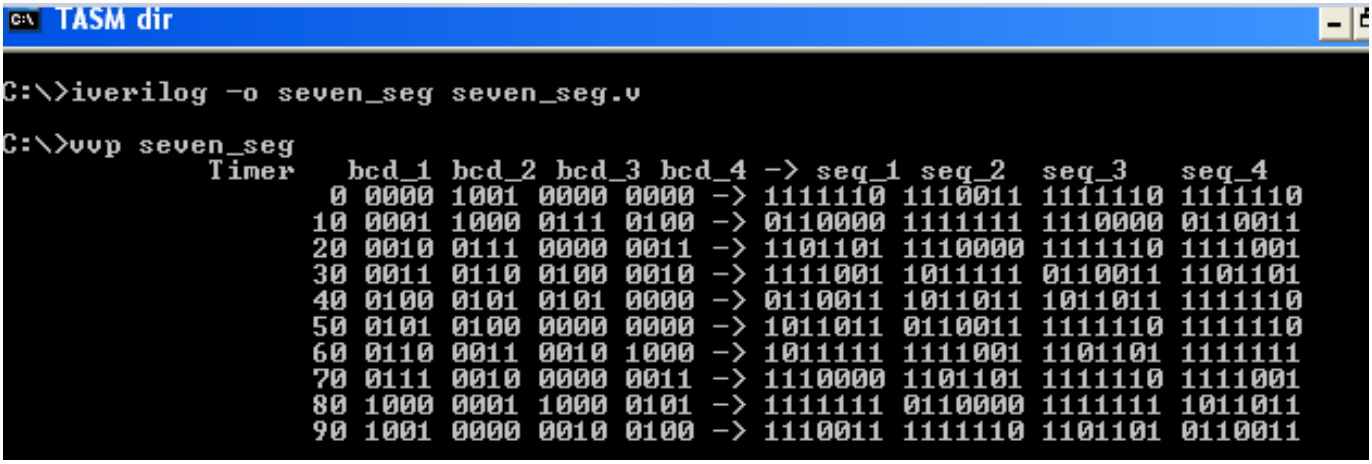
```
  A = timeClient.getHours() * 100 + timeClient.getMinutes();
  B = timeClient.getSeconds();

//  Serial.print (A);
//  Serial.print ("\n");

  if((B % 2) == 0)
  {
    display.showNumberDecEx(A, 0b01000000 , false, 4, 0);
  }
  else
  {
    display.showNumberDecEx(A, 0b00000000 , false, 4, 0);
  }
}
```

8. Program (Verilog)

```
module seven_seg_decoder(bcd,seven_seg);
        input [3:0] bcd;
        output reg[6:0] seven_seg;
        always@(*)
                begin
                  case (bcd)
                   4'b0000 : begin seven_seg = 7'b1111110; end
                   4'b0001 : begin seven_seg = 7'b0110000; end
                   4'b0010 : begin seven_seg = 7'b1101101; end
                   4'b0011 : begin seven_seg = 7'b1111001; end
                   4'b0100 : begin seven_seg = 7'b0110011; end
                   4'b0101 : begin seven_seg = 7'b1011011; end
                   4'b0110 : begin seven_seg = 7'b1011111; end
                   4'b0111 : begin seven_seg = 7'b1110000; end
```

```
                4'b1000 : begin seven_seg = 7'b1111111; end
                4'b1001 : begin seven_seg = 7'b1110011; end
                default : begin seven_seg = 7'b0000000; end
              endcase
            end
endmodule

module simulare;
        reg [3:0] bcd_1;
        reg [3:0] bcd_2;
        reg [3:0] bcd_3;
        reg [3:0] bcd_4;
        wire [6:0] seven_seg_1;
        wire [6:0] seven_seg_2;
        wire [6:0] seven_seg_3;
        wire [6:0] seven_seg_4;

        seven_seg_decoder seven_seg_decoder1(bcd_1,seven_seg_1);
        seven_seg_decoder seven_seg_decoder2(bcd_2,seven_seg_2);
        seven_seg_decoder seven_seg_decoder3(bcd_3,seven_seg_3);
        seven_seg_decoder seven_seg_decoder4(bcd_4,seven_seg_4);



   initial begin
                #0 bcd_1=4'b0000; bcd_2=4'b1001; bcd_3=4'b0000; bcd_4=4'b0000;
                #10 bcd_1=4'b0001; bcd_2=4'b1000; bcd_3=4'b0111; bcd_4=4'b0100;
                #10 bcd_1=4'b0010; bcd_2=4'b0111; bcd_3=4'b0000; bcd_4=4'b0011;
```

```
        #10 bcd_1=4'b0011; bcd_2=4'b0110; bcd_3=4'b0100; bcd_4=4'b0010;
        #10 bcd_1=4'b0100; bcd_2=4'b0101; bcd_3=4'b0101; bcd_4=4'b0000;
        #10 bcd_1=4'b0101; bcd_2=4'b0100; bcd_3=4'b0000; bcd_4=4'b0000;
        #10 bcd_1=4'b0110; bcd_2=4'b0011; bcd_3=4'b0010; bcd_4=4'b1000;
        #10 bcd_1=4'b0111; bcd_2=4'b0010; bcd_3=4'b0000; bcd_4=4'b0011;
        #10 bcd_1=4'b1000; bcd_2=4'b0001; bcd_3=4'b1000; bcd_4=4'b0101;
        #10 bcd_1=4'b1001; bcd_2=4'b0000; bcd_3=4'b0010; bcd_4=4'b0100;
    end

  initial begin
    $monitor($time, " %b %b %b %b -> %b %b %b %b", bcd_1[3:0], bcd_2[3:0], bcd_3[3:0],
bcd_4[3:0] ,seven_seg_1[6:0], seven_seg_2[6:0], seven_seg_3[6:0], seven_seg_4[6:0]);
    $display("\t    Timer   bcd_1 bcd_2 bcd_3 bcd_4 -> seq_1 seq_2  seq_3   seq_4 ");
  end
endmodule
```



9. Bibliografie

1. https://www.geekering.com/
2. https://www.instructables.com/
3. https://www.makerguides.com/
4. https://lastminuteengineers.com/
5. https://www.electronics-tutorials.ws/
6. https://www.fpga4student.com/
7. http://classweb.ece.umd.edu/