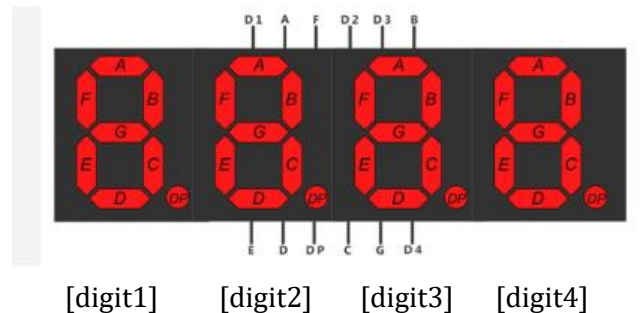


7-segmentni LED display s 4 znamenke (multiplexiranje)

1. Što je 4-znamenkasti display?

To je LED modul koji ima **četiri 7-segmentne znamenke** i često jednu decimalnu točku za svaku znamenku.

Izgleda ovako:



Svaka znamenka ima segmente A–G + DP, ali **segmenti su zajednički** za sve četiri znamenke.

To znači:

- segment A za sve četiri znamenke povezan je na isti izvod
- segment B također
- itd.

Zbog toga ih **ne možemo paliti odvojeno** bez posebne tehnike.

Ta tehnika zove se **multiplexiranje**:

Što je multiplexiranje?

Multiplexiranje znači:

Arduino pali samo jednu znamenku u isto vrijeme, ali tako brzo (oko 100–1000 puta u sekundi) da ljudsko oko vidi kao da sve svijetle istovremeno.

Postupak:

1. Uključimo segmente za znamenku 1
2. Uključimo samo digit-izvod za znamenku 1
3. Čekamo npr. 1 ms
4. Ugasimo digit 1
5. Uključimo segmente za znamenku 2
6. Upalimo digit 2
7. itd...

Ako se ponavlja dovoljno brzo, sve izgledaju stalno osvijetljene.

3. Vrste displaya

Postoje 2 glavne vrste:

COMMON CATHODE (CC)

- sve katode znamenke idu na GND kada je znamenka aktivna
- segmenti se pale HIGH signalom

COMMON ANODE (CA)

- sve anode znamenke idu na +5 V kada je aktivna
- segmenti se pale LOW signalom

U natjecanjima se najčešće koristi **common cathode**.

4. Pinovi 4-znamenkastog displaya

Tipičan 12-pinski raspored:

D1	A	F	D2	D3	B
E	D	DP	C	G	D4

Segmentni izvodi: A, B, C, D, E, F, G, DP

Digit-select izvodi: D1, D2, D3, D4

5. Kako spojiti na Arduino?

Primjer raspodjele izvoda:

Segmenti (8 izvoda):

7-segment LED izvodi	Arduino izvodi
A	D2
B	D3
C	D4
D	D5
E	D6
F	D7
G	D8
DP	D9

Digit-select (4 izvoda):

7-segment LED izvodi	Arduino izvodi
D1	D10
D2	D11
D3	D12
D4	D13

Digit-izvodi moraju imati tranzistore ili rade neposredno

Za niske struje (do ~20 mA po segmentu) može i direktno iz Arduina

6. Potrebno:

- 1× 4-znamenasti LED display
- 8× otpornika 220 Ω (jedan za svaki segment!)
- Arduino UNO / MKR
- Spojni vodiči

Otpornici idu **u seriju sa segmentima**, ne s digitima.

7. Kako radi multiplex u programu?

Korak 1: definirati sve segmente i znamenke

```
int segPins[8] = {2,3,4,5,6,7,8,9}; // A B C D E F G DP
int digitPins[4] = {10,11,12,13}; // DIG1-DIG4
```

Korak 2: tablica segmenata (brojevi 0-9)

```
byte digits[10] = {
  0b00111111, // 0
  0b00000110, // 1
  0b01011011, // 2
  0b01001111, // 3
  0b01100110, // 4
  0b01101101, // 5
  0b01111101, // 6
  0b00000111, // 7
  0b01111111, // 8
  0b01101111 // 9
};
```

Korak 3: funkcija za prikaz jedne znamenke

```
void showDigit(int digit, int number) {
  // ugasi sve znamenke
  for(int d=0; d<4; d++)
    digitalWrite(digitPins[d], LOW);

  // postavi segmente
  byte mask = digits[number];
  for(int i=0; i<7; i++)
    digitalWrite(segPins[i], (mask >> i) & 1);

  // upali traženu znamenku
  digitalWrite(digitPins[digit], HIGH);
}
```

Korak 4: u loop() brzo vrtimo sve četiri

```
void loop() {
  showDigit(0, 1); delay(2); // prikazuje 1 u prvoj znamenki
  showDigit(1, 2); delay(2); // 2 u drugoj
  showDigit(2, 3); delay(2); // 3 u trećoj
  showDigit(3, 4); delay(2); // 4 u četvrtoj
}
```

Ljudsko oko vidi 1234.

8. Najvažnije napomene za učenike

Bez otpornika segmenti će pregorjeti!

Ne smije se istovremeno paliti više znamenki.

Multiplex mora biti brži od ~ 200 Hz da ne titra.

Digit-izvodi se povremeno moraju "ugasiti" prije paljenja druge znamenke.

Kod mora biti čist i optimiziran — nema `delay(100)`!

9. Zaključak

4-znamenkasti 7-segmentni display važan je element za učenje:

- timera i prekida
- multiplexiranja
- bit-maskiranja
- prikaza brojčanih vrijednosti
- integracije senzora i korisničkog sučelja

Odličan je kao natjecateljski zadatak.