

Predmet:	MIKROUPRAVLJAČI
Vježba: 08	Arduino – 4x7 segmentni LED indikator
Ishodi vježbe:	Ispisivati brojeve i slova na 4 X 7 segmentni LED indikator koristeći standardne Arduino naredbe, te pomoću direktnog adresiranja Port registara

Priprema za vježbu:

Budući da je mikroupravljač računalo na čipu koji se programira kako bi upravljao priključenim vanjskim elektroničkim komponentama, priprema za vježbu se sastoji od dva dijela:

1. **Opis elektroničkih komponenti koje će se koristiti na LV** – proučiti tekst u uvodnom dijelu vježbe, proanalizirati i u bilježnicu ispisati najvažnije informacije za elektroničke komponente.
2. **Opis naredbi korištenih u LV** – proanalizirati programski kod za sve zadatke, ispisati nove naredbe i funkcije, objasniti njihovu namjenu i argumente. Ako ne možeš pronaći sve informacije u kodu priloženih zadataka, posluži se internetom npr. www.arduino.cc ...

Radu laboratoriju:

- Svaki zadatak treba prije prevođenja (eng. compile) pohraniti u napravljeni folder na Desktopu, tako da, u slučaju pogreške (HW, SW) imaš sačuvan kod.
- Na kraju LV, sve zadatke spremi na USB ili pošalji na svoj mail.
- Nazivi datoteka, zbog preglednosti, neka budu: LV01_ZAD01, LV01_ZAD02, itd.
- Vježbe se rade u paru, preporuka - jedan učenik spaja komponente, drugi piše programski kod, a na slijedećoj vježbi se uloge zamjenjuju.
- U zadacima koji zahtijevaju samostalno rješavanje, oba učenika sudjeluju u spajanju i programiranju.
- Za pojedini zadatak potrebno je u bilježnicu nacrtati električnu shemu s vidljivim oznakama korištenih pinova i vezu istih s oznakama u programskom kodu.
- Dobiveno rješenje treba komentirati, tj. dati zaključak što je novo u tom zadatku i kako je to riješeno, ukratko ispisati važniji dio koda (ne prepisivati cijeli kod) te navesti eventualne probleme i kako su isti riješeni.
- Ako su uz neki zadatak postoje pitanja, potrebno je u bilježnicu odgovoriti na ista.
- Ako u kodu postoji greška (negdje će biti namjerno stavljena) kod treba korigirati i objasniti!
- Budući da se na vježbama koriste stvarne komponente, postoji mogućnost da je neka komponenta neispravna (pregorena LED, oštećen kontakt tipkala, prekinut vodič...). Ukoliko se sklop ponaša drugačije od očekivanog, predvidjeti i tu mogućnost i pokušati zamijeniti komponentu drugom. **Isto vrijedni za ispitnu vježbu!**
- Prilikom spajanja, za Vcc (+5V) koristi crveni vodič, a za GND (-) crni vodič. Za ostale signale koristiti ostale boje.
- Za zadatke koje nisi stigao odraditi na vježbi, treba kod kuće razmisliti kako bi ih riješio
- Po završetku izvođenja vježbe, na temelju odrađene pripreme te riješenih zadataka, očekuje se da učenik zna odgovoriti na pitanja na kraju ovih materijala.

Pregledavanje priprema i provjeravanje znanja bit će na svakoj LV, uključujući i prethodne vježbe

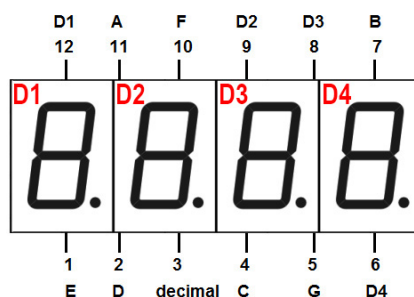
4 X 7 SEGMENTNI LED INDIKATOR

U praksi je vrlo često potrebno prikazati više od jedne znamenke, pa se koristi 4x7 segmentni LED indikator. Dolaze s različitim brojem znamenki i različitim boja znamenki. Prikladni su za prikaz vremena na digitalnim satovima ili na mjestima gdje treba jasno prikazati određenu brojčanu vrijednost npr. temperature. Slika 1.



Slika 1. 4x7 segmentni LED indikator [1]

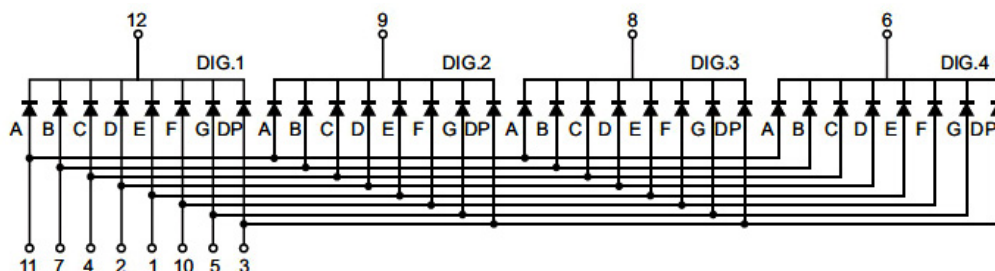
Postoji nekoliko varijanti rasporeda pinova 4x7 segmentnog LED indikatora. Na slici 2 prikazan je raspored pinova indikatora kojeg koristimo na LV.



Slika 2. Raspored pinova na 4x7 segmentnom LED indikatoru [2]

4x7 segmentni indikator ima 8 pinova za 8 standardnih segmenata (A – H) + 4 pina za 'odabir' jedne od 4 znamenke. Dakle, treba razlikovati termin 'segment' i termin 'znamenka'. Znamenke se naizmjenice uključuju i ispisuju se odgovarajući segmenti za tu znamenku. Ako se znamenka za znamenkom ispisuju dovoljno velikom brzinom, ljudsko oko će prepoznati kao da sve 4 znamenke svijetle istovremeno.

Na slici 3. je prikazana unutarnja shema 4x7 segmentnog LED indikatora sa **zajedničkom katodom**.

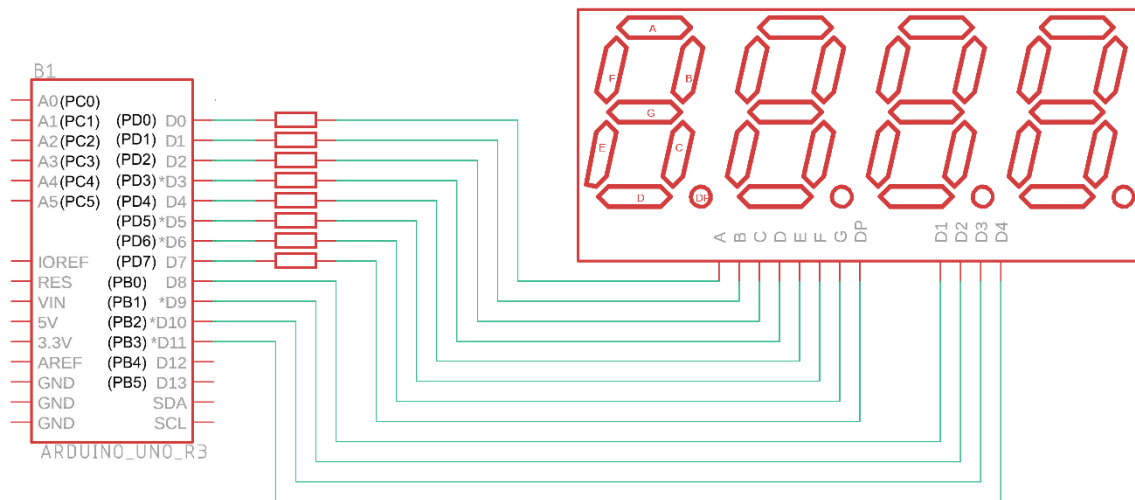


Slika 3. Unutarnja shema 4x7 segmentnog LED indikatora sa **zajedničkom katodom**

Odabir znamenke radi se pomoću 4 zajedničke katode (pinovi 12, 9, 8 i 6). Da bi neka od znamenki svijetlila, potrebno je postaviti logičku 1 (5V) na odgovarajuće pinove segmenata (A-H) i logičku 0 (GND) na željenu zajedničku katodu. Na ostale zajedničke katode treba dovesti logičku 1 (5V) kako te znamenke ne bi svijetlile.

- Dioda na čiji je segment dovedena logička 0, a na zajedničku katodu 1, bit će zaporno polarizirana pa neće svijetliti
- Dioda na čiji je segment dovedena logička 1 i na zajedničku katodu logička 1, anoda i katoda će biti na istom potencijalu (obje na 5V), pa opet neće svijetliti.

U ovom spoju, cijeli port D (8 pinova) koristi se za uključenje segmenata (pinovi 0-7), a 4 zajedničke katode se umjesto na GND, spajaju na 4 pina porta B (pinovi 8-11). Ti pinovi će na temelju logičke 0 (GND) ili logičke 1 (5V) određivati koja će znamenka svijetliti (slika 4).



Slika 4. Povezivanje 4x7 segmentnog LED indikatora na mikroupravljač

Primjer: samo na **prvu** znamenku želimo ispisati broj 1.

```
// SEGMENTI      H G F E D C B A
PORTD = 0b 0 0 0 0 0 1 1 0; // uključeni segmenti C i B za broj 1
// ZNAMENKE      D4 D3 D2 D1
PORTB = 0b x x x x 1 1 1 0; // uključena samo znamenka D1 jer
                             // je spojena na GND (logička 0)
```

Primjer: želimo ispisati broj 1 na **drugu** i **prvu** znamenku

```
// SEGMENTI      H G F E D C B A
PORTD = 0b 0 0 0 0 0 1 1 0; // uključeni segmenti C i B za broj 1
// ZNAMENKE      D4 D3 D2 D1
PORTB = 0b x x x x 1 1 0 0; // uključene znamenke D1 i D2 jer
                             // su spojene na GND (logička 0)
```

Primjer: želimo ispisati broj 7 na **drugu** znamenku i broj 1 na **prvu** znamenku. **To nije moguće istovremeno, nego jedno za drugim - multipleksiranje!**

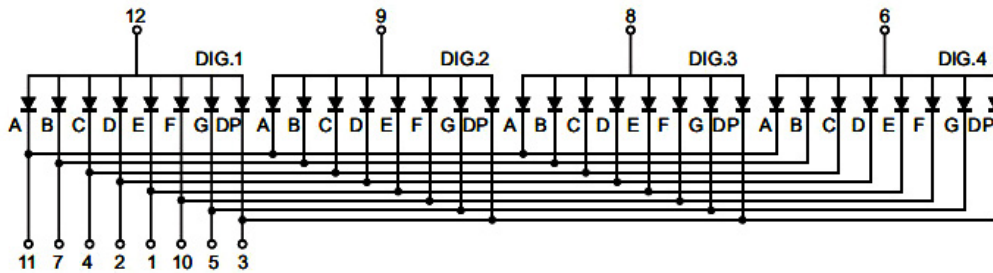
```
// SEGMENTI      H G F E D C B A
PORTD = 0b 0 0 0 0 0 1 1 0; // segmenti za broj 1
// ZNAMENKE      D4 D3 D2 D1
PORTB = 0b x x x x 1 1 1 0; // zajednička katoda D1 na GND
delay(100);
PORTB = 0b x x x x 1 1 1 1; // isključi sve 4 znamenke

// SEGMENTI      H G F E D C B A
PORTD = 0b 0 0 0 0 0 1 1 1; // segmenti za broj 7
// ZNAMENKE      D4 D3 D2 D1
PORTB = 0b x x x x 1 1 0 1; // zajednička katoda D2 na GND
delay(100);
PORTB = 0b x x x x 1 1 1 1; // isključi sve 4 znamenke

// PONAVLJAJ SVE OVO
```

Ako dovoljno brzo izvršavamo prethodni kod, na zaslonu će pisati broj 17.

Na slici 5. je prikazana unutarnja shema 4x7 segmentnog LED indikatora sa **zajedničkom anodom**.



Slika 5. Unutarnja shema 4x7 segmentnog LED indikatora sa **zajedničkom anodom** [4]

Odabir znamenke radi se pomoću 4 **zajedničke anode** (pinovi 12, 9, 8 i 6). Da bi neka od znamenki svijetlila, potrebno je postaviti logičku 0 (GND) na odgovarajuće pinove segmenata i logičku 1 (5V) na zajedničku anodu. Na ostale zajedničke anode treba dovesti logičku 0 (GND) kako te znamenke ne bi svijetlile. Dakle, suprotno tj. invertirane vrijednosti nego kod zajedničke katode.

Povezivanje 4x7 segmentnog LED indikatora na mikroupravljač izvodi se jednako kao kod zajedničke katode (slika 4), jedino su u programskom kodu sve vrijednosti za segmente i znamenke invertirane.

Ponovljen primjer za **zajedničku katodu**: samo na **prvu** znamenku želimo ispisati broj 1.

```
// SEGMENTI      H G F E D C B A
PORTD = 0b 0 0 0 0 0 1 1 0; // uključeni segmenti C i B za broj 1
// ZNAMENKE      D4 D3 D2 D1
PORTB = 0b x x x x 1 1 1 0; // uključena samo znamenka D1 jer
                             // je spojena na GND (logička 0)
```

Isti primjer za **zajedničku anodu**: samo na **prvu** znamenku želimo ispisati broj 1.

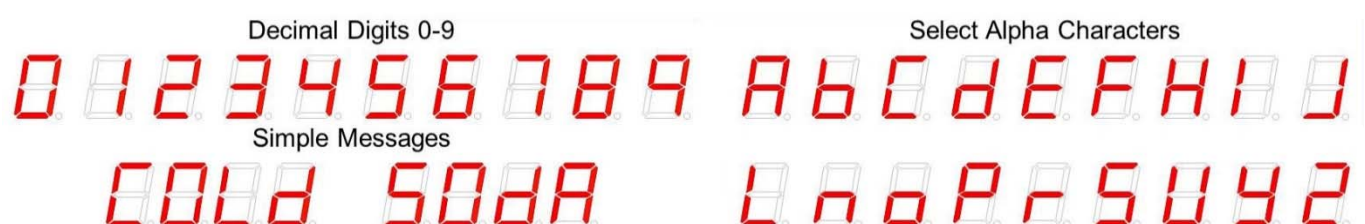
```
// SEGMENTI      H G F E D C B A
PORTD = 0b 1 1 1 1 1 0 0 1; // uključeni segmenti C i B za broj 1
// ZNAMENKE      D4 D3 D2 D1
PORTB = 0b x x x x 0 0 0 1; // uključena samo znamenka D1 jer
                             // je spojena na 5V (logička 1)
```

I ovdje se može koristiti jednostruki binarni komplement „~“:

Ponovljen primjer za **zajedničku anodu**: samo na **prvu** znamenku želimo ispisati broj 1.

```
// SEGMENTI      H G F E D C B A
PORTD = ~ 0b 0 0 0 0 0 1 1 0; // uključeni segmenti C i B za broj 1
// ZNAMENKE      D4 D3 D2 D1
PORTB = ~ 0b x x x x 1 1 1 0; // uključena samo znamenka D1 jer
                             // je spojena na GND (logička 0)
```

Na slici 6. prikazani su znakovi koji se mogu prikazati pomoću 7 segmentnog LED indikatora:



Slika 6. Unutarnja shema 4x7 segmentnog LED indikatora sa zajedničkom anodom [5]

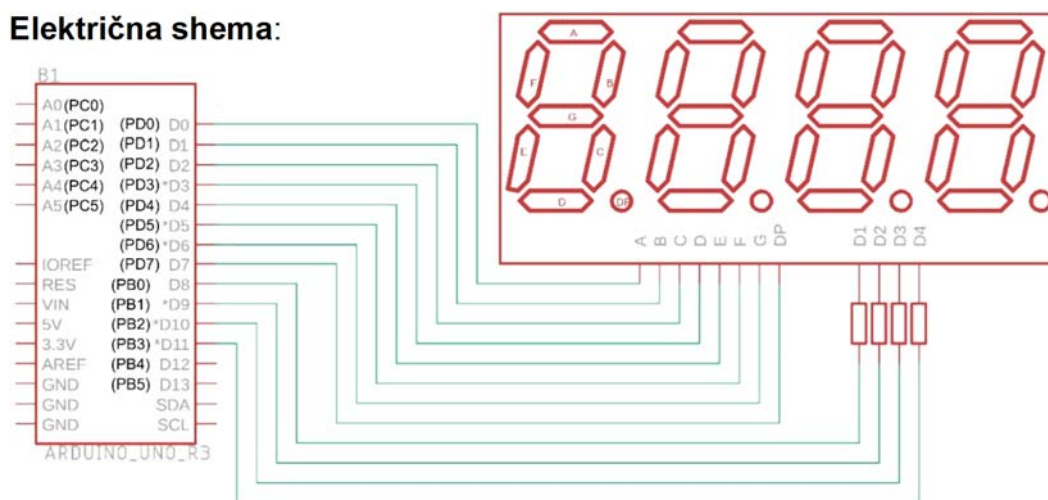
NAPOMENA: Prije izvođenja ove vježbe, potrebno je odraditi sve zadatke iz prethodne vježbe „MU-LV07 - Sedam segmentni LED indikator“ i potrebno je uočiti razliku između označavanja pinova kako ih obilježava proizvođač (PD0-PD7), a kako Arduino (0-7).

Zadatak 1. Spoji 4 X 7 segmentni LED indikator prema priloženoj shemi. Napiši program koji će napisati slovo P na poziciji prve znamenke s lijeve strane (D1). Pinovi 0 – 7 upravljaju pojedinim segmentom kao i u prethodnoj vježbi, a pinovi 8-11 upravljaju zajedničkim katodama, tj. koja od 4 znamenke svijetli.

NAPOMENA: Radi jednostavnosti spajanja, u ovom zadatku se koristi samo 4 otpornika od $220\ \Omega$ u krugu zajedničkih katoda (D1, D2, D3 i D4). Kod ovakvog spoja svjetlina segmenata se može mijenjati ovisno svijetli li jedan ili svi segmenti.

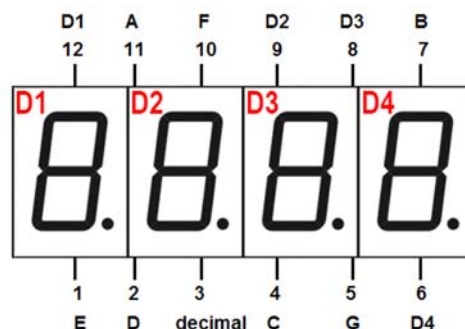
Ispravan spoj dan je na slici 4 u uvodnom tekstu gdje svaki vod za segmente ima zaseban otpornik.

Električna shema:

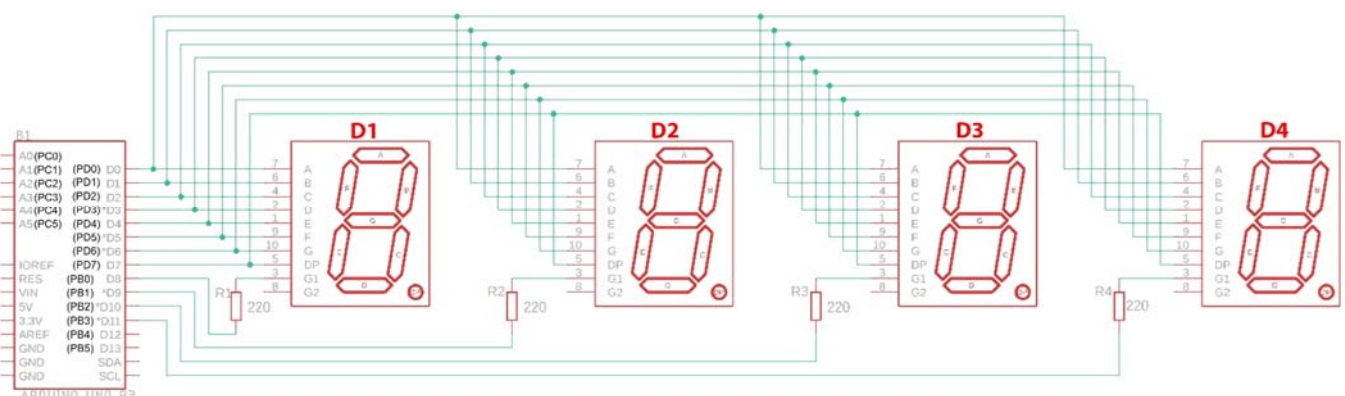


Arduino Pin	4x7 segment. LED display Pin
0	11 (A)
1	7 (B)
2	4 (C)
3	2 (D)
4	1 (E)
5	10 (F)
6	5 (G)
7	3 (DP)
8	12 (D1)
9	8 (D2)
10	9 (D3)
11	6 (D4)

Raspored izvoda:



NAPOMENA: u slučaju da nemaš 4x7 segmentni indikator ili radiš vježbu online, možeš koristiti 4 obična 7 segmentna indikatora prema shemi ispod.



NAPOMENA2: Budući da prilikom učitavanja programa na Arduino UNO pločicu pin 0 koristi kao RX i pin 1 kao TX, potrebno je privremeno odspojiti vodič RX sa Arduina. U suprotnom program se neće moći učitati na Arduino.

Kòd zadatka

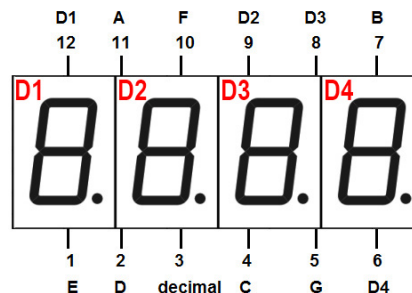
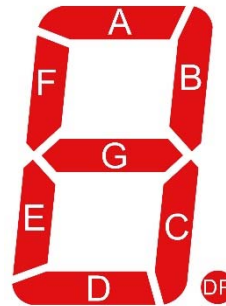
```
int intDelay = 500; // vrijeme čekanja između ispisa dviju znamenki
                    // u ovom zadatku, nigdje se ne isključuje prva
                    // znamenka, tako da slovo P svijetli cijelo vrijeme

void setup() {
  pinMode(0, OUTPUT); // segment A
  pinMode(1, OUTPUT); // segment B
  pinMode(2, OUTPUT); // segment C
  pinMode(3, OUTPUT); // segment D
  pinMode(4, OUTPUT); // segment E
  pinMode(5, OUTPUT); // segment F
  pinMode(6, OUTPUT); // segment G
  pinMode(7, OUTPUT); // decimalna točka

  pinMode(8, OUTPUT); // Pin za katodu prve znamenke D1
  pinMode(9, OUTPUT); // Pin za katodu druge znamenke D2
  pinMode(10, OUTPUT); // Pin za katodu treće znamenke D3
  pinMode(11, OUTPUT); // Pin za katodu četvrte znamenke D4
}

void loop() {
  // piši 'P'
  digitalWrite(0, 1);
  digitalWrite(1, 1);
  digitalWrite(2, 0);
  digitalWrite(3, 0);
  digitalWrite(4, 1);
  digitalWrite(5, 1);
  digitalWrite(6, 1);
  digitalWrite(7, 0); // Isključujemo decimalnu točku

  digitalWrite(8, 0); // Aktivna samo prva znamenka D1
  digitalWrite(9, 1);
  digitalWrite(10, 1);
  digitalWrite(11, 1);
  delay(intDelay);
}
```



Zadatak 2. Doradi program tako da slovo P trepće na poziciji D1 s intervalom 500 ms.

Zadatak 3. Doradi program tako da se slovo P ispisuje na poziciju D1 kao i do sada, a zatim isto slovo P na poziciji D2. Obje znamenke se trebaju ispisivati naizmjenice u intervalu 500 ms. Što treba napraviti da se oba slova P ispisuju istovremeno?

Umjesto 8 naredbi `pinMode()`, koristiti direktno adresiranje registara porta naredbom `DDRD=b11111111` i `DDRB = b11111111`.

Kòd zadatka

```
int intDelay = 500; // vrijeme čekanja između ispisa dviju znamenki

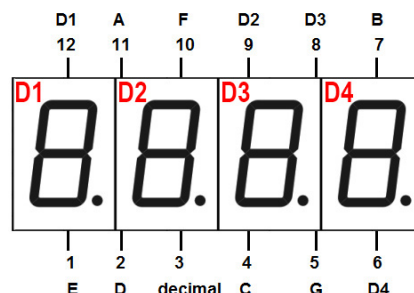
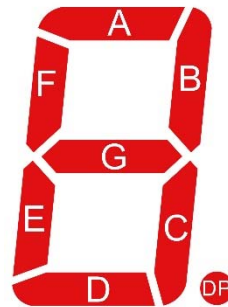
void setup() {
    // segmenti HGFEDCBA izlazni (1 = izlazni pin; 0 = ulazni pin)
    DDRD = B11111111; // postavlja pinove 0-7 kao izlazne
                      // to odgovara portu D (ATmega 328p)
                      // služe za upravljanje 7 segmenata + točka
                      // umjesto 8 naredbi pinMode()

    DDRB = B11111111; // postavlja pinove 8-13 kao izlazne
                      // to odgovara portu B (ATmega 328p)
                      // služe za upravljanje znamenkama
}

void loop() {
    // piši 'P'
    digitalWrite(0, 1);
    digitalWrite(1, 1);
    digitalWrite(2, 0);
    digitalWrite(3, 0);
    digitalWrite(4, 1);
    digitalWrite(5, 1);
    digitalWrite(6, 1);
    digitalWrite(7, 0); // Isključujemo decimalnu točku

    digitalWrite(8, 0); // Aktivna samo prva znamenka D1
    digitalWrite(9, 1);
    digitalWrite(10, 1);
    digitalWrite(11, 1);
    delay(intDelay);

    digitalWrite(8, 1);
    digitalWrite(9, 0); // Aktivna samo druga znamenka D2
    digitalWrite(10, 1);
    digitalWrite(11, 1);
    delay(intDelay);
}
```



Zadatak 4. Doradi prethodni program tako da na poziciji znamenke D2 umjesto slova P ispisuješ slovo U. Sada bi se trebale naizmjenice ispisivati slovo P na poziciju D1 i slovo U na poziciji D2.

Kòd zadatka

```
int intDelay = 500; // vrijeme čekanja između ispisa dviju znamenki

void setup() {
    // segmenti HGFEDCBA izlazni (1 = izlazni pin; 0 = ulazni pin)
    DDRD = B11111111; // postavlja pinove 0-7 kao izlazne

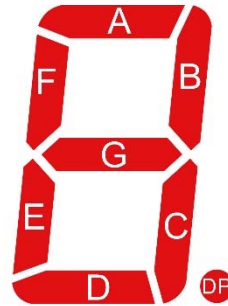
    DDRB = B11111111; // postavlja pinove 8-13 kao izlazne
}

void loop() {
    // piši 'P'
    digitalWrite(0, 1);
    digitalWrite(1, 1);
    digitalWrite(2, 0);
    digitalWrite(3, 0);
    digitalWrite(4, 1);
    digitalWrite(5, 1);
    digitalWrite(6, 1);
    digitalWrite(7, 0); // Isključujemo decimalnu točku

    digitalWrite(8, 1); // Aktivna samo prva znamenka D1
    digitalWrite(9, 1);
    digitalWrite(10, 1);
    digitalWrite(11, 0);
    delay(intDelay);

    // piši 'U'
    digitalWrite(0, 0);
    digitalWrite(1, 1);
    digitalWrite(2, 1);
    digitalWrite(3, 1);
    digitalWrite(4, 1);
    digitalWrite(5, 1);
    digitalWrite(6, 0);
    digitalWrite(7, 0); // Isključujemo decimalnu točku

    digitalWrite(8, 1);
    digitalWrite(9, 0); // Aktivna samo druga znamenka D2
    digitalWrite(10, 1);
    digitalWrite(11, 1);
    delay(intDelay);
}
```



Zadatak 5. Doradi prethodni program tako da na poziciju D3 pišeš slovo „L“ i na poziciju D4 slovo „A“. Sada bi se trebala naizmjence ispisivati slova P U L A.

Zadatak 6. Prethodni zadatak modificiraj tako da dodaš potencijometar na ulaz A0 Arduina pomoću kojeg ćeš regulirati brzinu ispisa slova. Brzinu ispisa treba regulirati u granicama 1 ms do 250 ms.

1. Koje modifikacije u kodu treba napraviti da se dobije čist i stabilan prikaz znamenki? Napravi potrebne modifikacije!

Zadatak 7. Modificiraj kod iz prethodnog zadatka tako da u potpunosti izbaciš ugrađene Arduino naredbe: pinMode() i digitalWrite(). To se odnosi i na pinove 0 – 7 (port D) koji upravljaju segmentima, te pinove 8, 9, 10 i 11 (port B) kojima se upravlja pojedinom znamenkom. Koristi naredbe PORTD = bxxxxxxx i PORTB = bxxxxxxx.

Zadatak 8. Modificiraj program iz prethodnog zadatka tako da na 4 X 7 segmentnom zaslonu ispisuješ očitane A/D vrijednost potencijometra priključenog na A0 ulaz. Prati stanje znamenki na Serial monitoru. [6]

Kòd zadatka

```

/*****
File name: 14_aSimpleCounter.ino
Description:when you are pressing two buttons, four-digit
segment display values will change.
Website: www.adeept.com
E-mail: support@adeept.com
Author: Tom
Date: 2015/05/05

http://www.avr-tutorials.com/projects/atmega16-based-digital-clock

Program djelomično modificiran za potrebe LV!!!
*****/

int nDelayMicroseconds = 50;

int d1 = 8;
int d2 = 9;
int d3 = 10;
int d4 = 11;

// int a = 0;
// int b = 1;
// int c = 2;
// int d = 3;
// int e = 4;
// int f = 5;
// int g = 6;
// int p = 7;

int data;
int segData;

void setup()
{
    pinMode(A0, INPUT);

    // ARDUINO:      13 12 11 10 9 8    7 6 5 4 3 2 1 0 --> pinovi Arduina
    //ATmega328p: B7 B6 B5 B4 B3 B2 B1 B0 D7 D6 D5 D4 D3 D2 D1 D0 --> nožice ATmega328p
    // 7 segm:      d4 d3 d2 d1    h g f e d c b a --> segmenti 4x7 segm.LED
    // 7 segm:      6 9 8 12    3 5 10 1 2 4 7 11 --> nožice 4x7 segm.LED

```

```

// 0          0 0 1 1 1 1 1
// 1          0 0 0 0 0 1 1
// 2          0 1 0 1 1 0 1
// 3          0 1 0 0 1 1 1
// 4          0 1 1 0 0 1 1
// 5          0 1 1 0 1 1 0
// 6          0 1 1 1 1 1 0
// 7          0 0 0 0 0 1 1
// 8          0 1 1 1 1 1 1
// 9          0 1 1 0 1 1 1
// A
// B
// C
// D
// E
// F

DDRD = DDRD | B11111111; // postavlja pinove porta D (ATmega 328p) 2-7 kao izlazne
DDRB = DDRB | B00111111; // postavlja pinove 8-13 kao izlazne
//Serial.begin(9600);
}

void loop()
{
    int d1, d2, d3, d4;

    data = analogRead(A0);

    //data = 1023;

    d1 = data / 1000 % 10; //Znamenka tisućice
    d2 = data / 100 % 10; //Znamenka stotice
    d3 = data / 10 % 10; //Znamenka desetice
    d4 = data % 10; //Znamenka jedinice

    for (int m = 0; m<80; m++) {
        clearLEDs(); //Turn off all LED lights
        segData = DigitTo7SegEncoder(d1, 1);
        pickDigit(1); //Selection of a digital display
        PORTD = segData;
        delayMicroseconds(nDelayMicroseconds);

        clearLEDs(); //Turn off all LED lights
        segData = DigitTo7SegEncoder(d2, 1);
        pickDigit(2); //Select the first two digital display
        PORTD = segData;
        delayMicroseconds(nDelayMicroseconds);

        clearLEDs(); //Turn off all LED lights
        segData = DigitTo7SegEncoder(d3, 1);
        pickDigit(3); //Select the first three digital display
        PORTD = segData;
        delayMicroseconds(nDelayMicroseconds);

        clearLEDs(); //Turn off all LED lights
        segData = DigitTo7SegEncoder(d4, 1);
        pickDigit(4); //Select the first four digital display
        PORTD = segData;
        delayMicroseconds(nDelayMicroseconds);
    }
}

void pickDigit(int x) //Defined pickDigit (x), whose role is to open the port dx
{
    digitalWrite(d1, HIGH);

```

```

    digitalWrite(d2, HIGH);
    digitalWrite(d3, HIGH);
    digitalWrite(d4, HIGH);

    switch (x)
    {
    case 1:
        digitalWrite(d1, LOW);
        break;
    case 2:
        digitalWrite(d2, LOW);
        break;
    case 3:
        digitalWrite(d3, LOW);
        break;
    default:
        digitalWrite(d4, LOW);
        break;
    }
}

void clearLEDs() //Clear screen
{
    PORTD = B00000000;
    //digitalWrite(a, LOW);
    //digitalWrite(b, LOW);
    //digitalWrite(c, LOW);
    //digitalWrite(d, LOW);
    //digitalWrite(e, LOW);
    //digitalWrite(f, LOW);
    //digitalWrite(g, LOW);
    //digitalWrite(p, LOW);
}

/*
 * Function Description:
 * Encode a Decimal Digit 0-9 to its Seven Segment Equivalent.
 *
 * Function Arguments:
 * digit - Decimal Digit to be Encoded
 * common - Common Anode (0), Common Cathode(1)
 * SegVal - Encoded Seven Segment Value
 *
 * Connections:
 * Encoded SegVal is return in the other G-F-E-D-C-B-A that is A is the least
 * significant bit (bit 0) and G bit 6.
 */

unsigned char DigitTo7SegEncoder(unsigned char digit, unsigned char common)
{
    unsigned char SegVal;

    switch (digit)
    {
    case 0: if (common == 1) SegVal = 0b00111111; // broj 0
            else SegVal = ~0b00111111;
            break;
    case 1: if (common == 1) SegVal = 0b00000110; // broj 1
            else SegVal = ~0b00000110;
            break;
    case 2: if (common == 1) SegVal = 0b01011011; // broj 2
            else SegVal = ~0b01011011;
            break;
    case 3: if (common == 1) SegVal = 0b01001111; // broj 3
            else SegVal = ~0b01001111;
            break;
    }
}

```

```
case 4: if (common == 1) SegVal = 0b01100110; // broj 4
        else SegVal = ~0b01100110;
        break;
case 5: if (common == 1) SegVal = 0b01101101; // broj 5
        else SegVal = ~0b01101101;
        break;
case 6: if (common == 1) SegVal = 0b01111101; // broj 6
        else SegVal = ~0b01111101;
        break;
case 7: if (common == 1) SegVal = 0b00000111; // broj 7
        else SegVal = ~0b00000111;
        break;
case 8: if (common == 1) SegVal = 0b01111111; // broj 8
        else SegVal = ~0b01111111;
        break;
case 9: if (common == 1) SegVal = 0b01101111; // broj 9
        else SegVal = ~0b01101111;
    }
    return SegVal;
}
```

Zadatak 9. Modificiraj spoj iz prethodnog zadatka tako da dodaš dva tipkala. Modificiraj program tako da pritiskom na jednu tipku povećavaš broj prikazan na 4 X 7 segmentnom indikatoru, a pritiskom na drugu tipku smanjuješ. Inicijalno, na indikatoru treba napisati broj 500.

1. Da li svakim pritiskom broj povećavaš samo za 1?

Zadatak 10. Modificiraj kod tako da se brojanje povećava ili smanjuje samo na pritisak tipke i da ne broji dalje dok je tipka pritisnuta. Dakle samo na promjenu.

Pitanja za provjeru znanja:

1. Koje sve pinove ima 4x7 segmentni LED indikator i čemu služi pojedini pin?
2. Skiciraj 4x7 segmentni LED indikator (i njegove pinove) koji smo koristili na LV!
3. Skiciraj unutarnju shemu 2x7 segmentnog LED indikatora sa zajedničkom katodom!
4. Skiciraj unutarnju shemu 2x7 segmentnog LED indikatora sa zajedničkom anodom!
5. Skiciraj kako bi povezoao 4x7 segmentni LED indikator na mikroupravljač!
6. Objasni na koji način se ispisiuje određeni broj ili tekst na 4x7 segmentni LED indikator u slučaju zajedničke katode!
7. Objasni na koji način se ispisiuje određeni broj ili tekst na 4x7 segmentni LED indikator u slučaju zajedničke anode!
8. Može li se na 4x7 segmentni indikator istovremeno ispisiati 1111? Može li se istovremeno ispisiati broj 1234? Objasni!
9. Kako smo na LV spojili otpornike za zaštitu od prevelike struje kroz 4x7-segmentni LED indikator i zašto? Što je nedostatak takvog spoja?
10. Uoči i objasni problem/nedosljednost u programskom kodu po pitanju uključivanja i isključivanja segmenata i znamenki u 4. zadatku ove vježbe?
11. Objasni izvadak koda iz zadatka 8 pomoću kojeg se određuje koji broj se ispisiuje na pojedinu znamenku 4x7 segmentnog indikatora!

```
data = 1023;  
d1 = data / 1000 % 10;      //Znamenka tisućice  
d2 = data / 100 % 10;       //Znamenka stotice  
d3 = data / 10 % 10;        //Znamenka desetice  
d4 = data % 10;             //Znamenka jedinice
```

12. Napiši program koji će pomoću direktnog adresiranja registara porta na 4x7 segmentnom LED indikatoru sa zajedničkom katodom ispisiati HELP?
13. Napiši program koji će pomoću direktnog adresiranja registara porta na 4x7 segmentnom LED indikatoru sa zajedničkom anodom ispisiati HELP?
14. Navedi vezu pinova ATmega328p i Arduino UNO pločice!
15. Što treba napraviti prilikom učitavanja programa na Arduino UNO kad se za spoj na 7-segmentni LED koristi čitav Port D, tj. pinovi 0 – 7 i zašto?

LITERATURA:

1. HobbyTronics, 4-Digit 7-Segment Display - Red, <http://www.hobbytronics.co.uk/4digit-7segment-red>, (pregledano 19.01.2019.)
2. Learning about Electronics, How to Control a 4-digit 7-segment LED Display with an Arduino, <http://www.learningaboutelectronics.com/Articles/4-digit-7-segment-LED-circuit-with-an-arduino.php>, (pregledano 19.01.2019.)
3. Nuts and Volts, Using Seven-Segment Displays — Part 1, <https://www.nutsvolts.com/magazine/article/using-seven-segment-displays-part-1>, (pregledano 19.01.2019.)
4. Electrical Engineering Stack Exchange, Using 4 digit 7 segment LED, <https://electronics.stackexchange.com/questions/34815/using-4-digit-7-segment-led>, (pregledano 20.01.2019.)
5. Seven Segment Displays, Project Lead The Way, Dana Watts, <https://slideplayer.com/slide/8938795/>, (pregledano 20.01.2019.)
6. AVR tutorials, ATmega16 AVR Microcontroller Seven Segment Digital Clock , <http://www.avr-tutorials.com/projects/atmega16-based-digital-clock>, (pregledano 20.01.2019.)