TEHNIČKA ŠKOLA RUĐERA BOŠKOVIĆA, ZAGREB

Predmet:	**MIKROUPRAVLJAČI**
Vježba: 07	**Arduino – Sedam segmentni LED indikator**
Ishodi vježbe:	Ispisivati brojeve i slova na 7-segmentni LED indikatoru, usvojiti upravljanje direktnim adresiranjem Port registara, upravljati indikatorima sa zajedničkom anodom i katodom

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) kòd 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

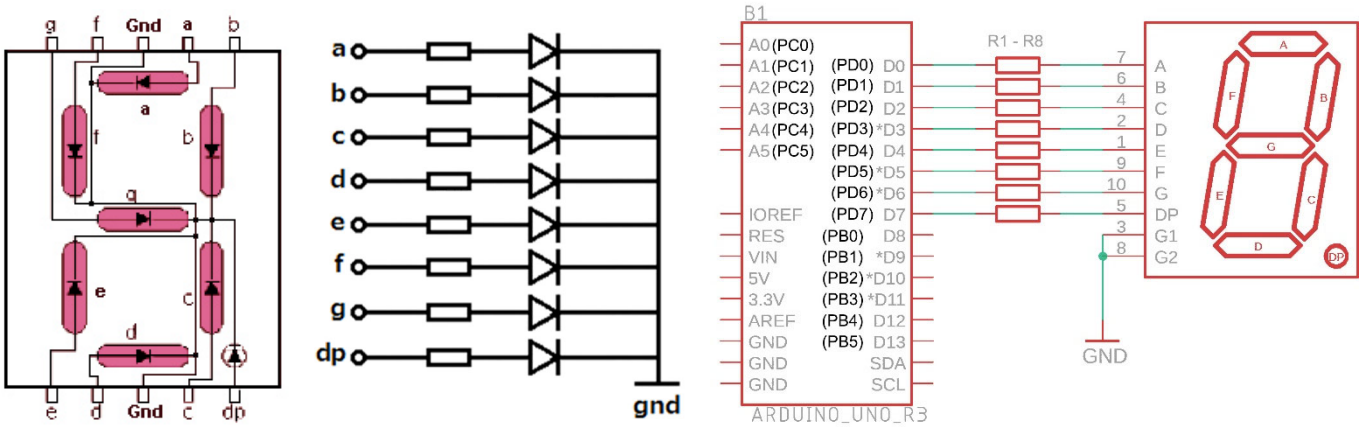
SEDAM SEGMENTNI LED INDIKATOR

Sedam segmentni LED indikator (SSD – od engl. Seven Segment Display) se može svrstati u jednostavnije vizualne indikatore. Sastoji se od 8 LED dioda koje predstavljaju 7 segmenata (A-G) i jedne decimalne točke (DP). Služe za prikaz brojeva i nekih slova. Prednost LED indikatora je bolja uočljivost, pogotovo u tamnoj okolini. Nedostatak je teže razlikovanje brojeva ako imamo više takvih indikatora, pogotovo ako se pomiče i decimalna točka. Uobičajeni način obilježavanja segmenata dan je na slici 1.

Slika 1. Sedam segmentni LED indikator i raspored pinova [1]

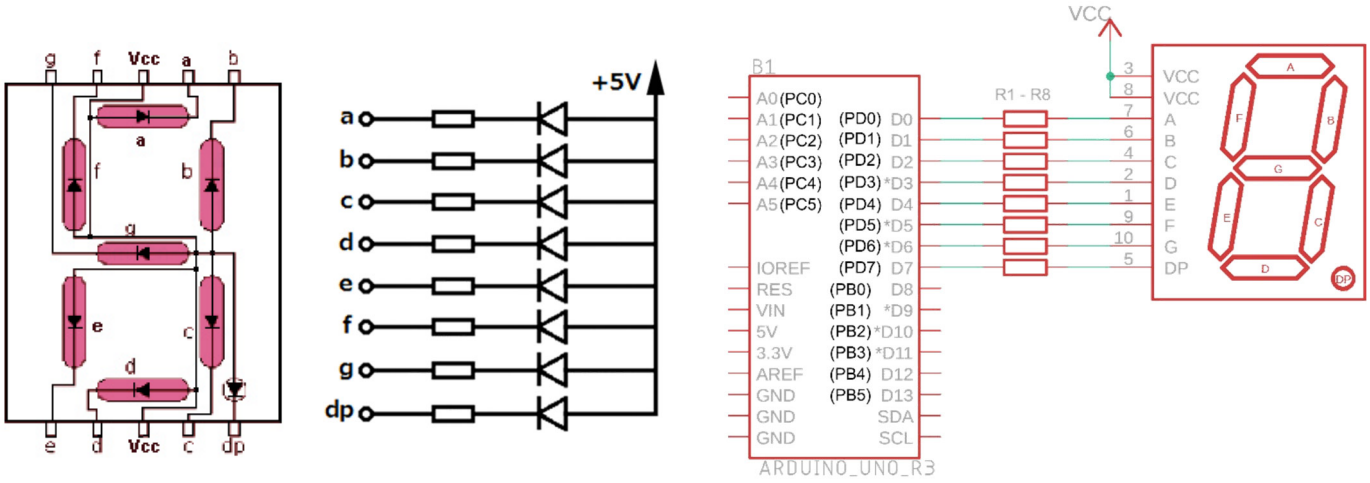
Osam LED dioda ima ukupno 16 izvoda, 8 anoda i 8 katoda. Kako bi se smanjio ukupan broj nožica, na 7 segmentnom LED indikatoru unutar indikatora su međusobno povezane ili sve anode ili sve katode. Budući da za 7 + 1 segment i zajedničku nožicu treba 9 nožica, deseta nožica se iskoristi za još jednu zajedničku nožicu koje su interno spojene, pa je svejedno da li se na GND spaja nožica 3 ili 8 (npr. kod zajedničke katode).

U slučaju da se zajedno povežu sve **katode**, takav se spoj zove spoj sa **zajedničkom katodom**, a pojedinim segmentom se upravlja pomoću svake **zasebne anode**. Da bi segmenti svijetlili, diode moraju biti propusno polarizirane, odnosno zajednička katoda (–) mora biti spojena na GND, a 8 izvoda anoda na 5V, tj. na izlaze mikroupravljača. Segmenti se uključuju logičkom 1 (tj. 5V) na izlazu odgovarajućeg pina (slika 2). Kao i kod obične LED diode, potrebno je u seriju ugraditi otpore.



Slika 2. Sedam segmentni LED indikator sa zajedničkom katodom (–) [2]

U slučaju da se zajedno povežu sve anode, takav se spoj zove spoj sa **zajedničkom anodom**, a pojedinim segmentom se upravlja pomoću svake zasebne katode. Da bi segmenti svijetlili, diode moraju biti propusno polarizirane, odnosno zajednička anoda (+) mora biti spojena na VCC, a 8 izvoda anoda na izlaze mikroupravljača. Segmenti se uključuju logičkom 0 (tj. 0V) na izlazu odgovarajućeg pina (slika 3).



Slika 3. Sedam segmentni LED indikator sa zajedničkom anodom (+) [2]

Obično je moguće naručiti isti model koji ima jednak raspored nožica. Razlikuju se jedino u zajedničkoj nožici koja može biti VCC (zajednička anoda) ili GND (zajednička katoda).

Želimo li na 7-segmentnom indikatoru sa zajedničkom katodom ispisati broj 1, potrebno je zajedničku katodu spojiti na GND, a na pinove anoda postaviti napone prema tablici 1.

Segmenti (logička 1 = 5V)							Znak	Segmenti (logička 1 = 5V)							Znak
G	F	E	D	C	B	A		G	F	E	D	C	B	A	
	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0
				1	1		1	1	1			1	1	1	9
1		1	1		1	1	2	1	1	1		1	1	1	A
1			1	1	1	1	3	1	1	1	1	1			b
1	1			1	1		4		1	1	1			1	C
1	1		1	1		1	5	1		1	1	1	1		d
1	1	1	1	1		1	6	1	1	1	1			1	E
				1	1	1	7	1	1	1				1	F

Tablica 1. Raspored segmenata za zajedničku katodu [3]

Ako koristimo indikator sa zajedničkom anodom, anodu spajamo na Vcc. Pojedine katode uključujemo logičkom 0, a katode na kojima je logička 1 neće svijetliti. U tom slučaju, potrebno je segmente iz tablice invertirati.

Primjerice, za broj 1 uz zajedničku katodu segmenti su:

```
hgfedcba
00000110
```

Za broj 1 uz zajedničku anodu, segmenti su invertirani:

```
hgfedcba
11111001
```

Uz pomoć jednostrukog bit komplementa „~“ može se koristiti ista tablica:

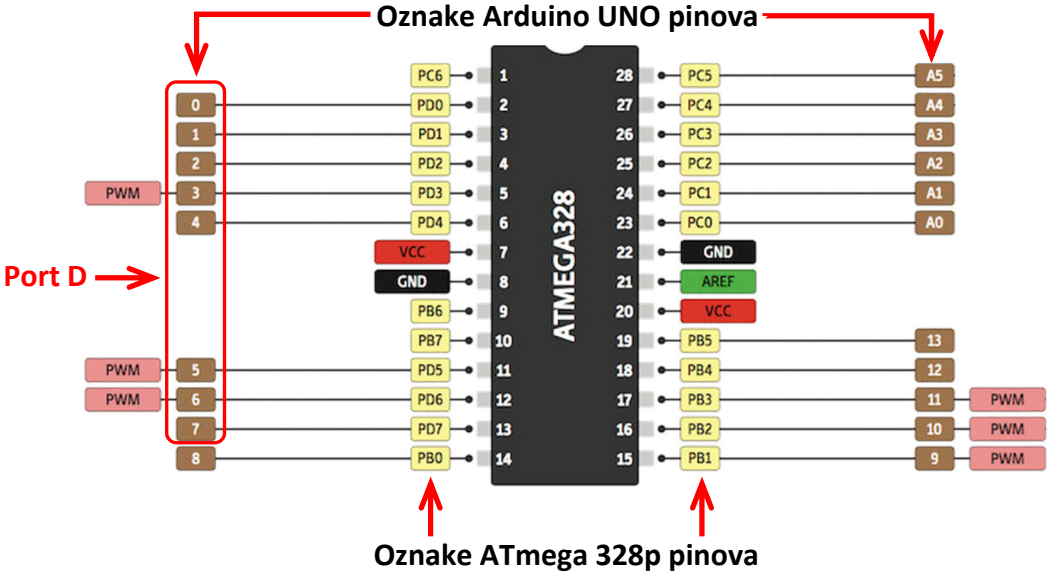
```
(Broj 1 za zajedničku anodu) = ~(broj 1 za zajedničku katodu)

11111001 = ~00000110
```

DIREKTNO ADRESIRANJE REGISTARA PORTOVA MIKROUPRAVLJAČA

Mikroupravljač **ATmega 328p** na kojem je baziran **Arduino UNO**, ima 3 porta (slika :
Port B ima 8 pinova: PB0, PB1,… PB7 (Pinovi Arduina 8 – 13, PB6 i PB7 nisu iskorišteni)
Port C ima iznimno 7 pinova: PC0, PC1,…PC6 (Pinovi Arduina A0 – A5, PC6 nije iskorišten)
Port D ima 8 pinova: PD0, PD1,… PD7 (Pinovi Arduina 0 – 7)

Svakim portom se upravlja pomoću tri 8-bitna registra: **DDR**, **PORT** i **PIN**. Svaki bit pojedinog registra se odnosi na jedan pin. Ti registri imaju specifičnu funkciju:

1. **DDRx** je registar u koji upisujemo binarne vrijednost koje označavaju da li će neki pin biti ulazni ili izlazni (DDRB = b11110000 postavlja pinove B4-B7 kao izlazne, a B0-B3 kao ulazne
2. **PORTx** – je registar u koji upisujemo željeno stanje izlaza tog porta (0 ili 1) – u slučaju da je pojedini pin definiran kao izlazni u DDR registru – **ili**, ako je port definiran kao ulazni, uključujemo Pull-Up otpornik koja je potreban za priključenje tipkala.
3. **PINx** – je registar u kojem pojedini bit označava stanje dovedeno na odgovarajuću nožicu tog porta



Slika 4. Veza oznaka pinova ATmega328p i Arduino UNO pločice [4]

Koristeći direktno adresiranje porta, u jednom koraku možemo upravljati s 8 pinova, što smanjuje broj naredbi, ali i izbjegava korištenje standardnih Arduino naredbi što može značajno ubrzati izvršavanje programa. Više: https://www.arduino.cc/en/Reference/PortManipulation

Primjerice:

```
void setup() {
      pinMode(0, OUTPUT);
      pinMode(1, OUTPUT);
      pinMode(2, OUTPUT);
      pinMode(3, OUTPUT);
      pinMode(4, OUTPUT);
      pinMode(5, OUTPUT);
      pinMode(6, OUTPUT);
      pinMode(7, OUTPUT);
}
```

Možemo ostvariti pomoću:

```
void setup() {
      DDRD = 0b11111111;
// sve pinove porta D (Arduino pinovi
// 0-7) postavljamo kao izlazne
// '1' je izlaz, '0' je ulaz
}
```

```
void loop() {             // piši '1'
      digitalWrite(0, 0);// segment A
      digitalWrite(1, 1);// segment B
      digitalWrite(2, 1);// segment C
      digitalWrite(3, 0);// segment D
      digitalWrite(4, 0);// segment E
      digitalWrite(5, 0);// segment F
      digitalWrite(6, 0);// segment G
      digitalWrite(7, 0);// segment DP
}
```

Možemo ostvariti pomoću:

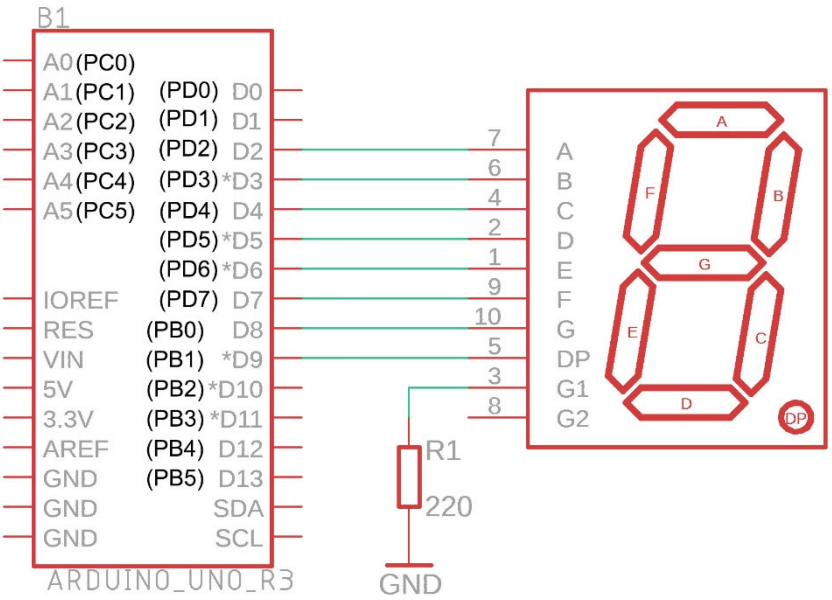
```
void loop() {
      PORTD = 0b00000110;
// u PORTD registar upisujemo stanje
// svih 8 pinova za 8 segmenata
// '1' je 5V, '0' je 0V
}
```

Primjer jedne univerzalne funkcije za ispis odgovarajućih segmenata dan je u nastavku. U funkciju se kao argumenti unose željeni znak (**`unsigned char digit`**) te argument za zajedničku anodu ili katodu (**`unsigned char common`**). [5]

```
/*
* 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 1. Spoji 7-segmentni LED indikator sa zajedničkom katodom (zajednički GND) i napiši program kojim će se na indikatoru odbrojavati od 0-9. Pohrani ovaj zadatak za kasniju uporabu!

Električna shema:



Raspored pinova:

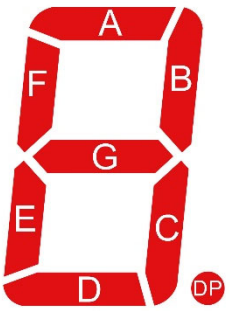
Arduino Pin	7 segmentni LED display Pin
2	7 (A)
3	6 (B)
4	4 (C)
5	2 (D)
6	1 (E)
7	9 (F)
8	10 (G)
9	5 (DP)
GND	3 i/ili 8

NAPOMENA: Radi jednostavnosti spajanja, u ovom zadatku se koristi samo jedan otpornik. Kod ovakvog spoja svjetlina segmenata se mijenja ovisno o tome da li svijetli jedan ili svi segmenti. Ispravan spoj dan je na slici 2 u uvodnom tekstu gdje svaki vod za segmente ima zaseban otpornik.

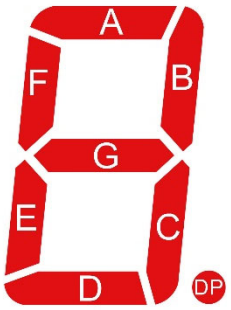
Kòd zadatka

```
int intDelay = 200;    // vrijeme čekanja između ispisa dviju znamenki u ms
int intPause = 1000;   // vrijeme čekanja između ispisa nove serije
                       // znamenki u ms
void setup() {
      pinMode(2, OUTPUT);
      pinMode(3, OUTPUT);
      pinMode(4, OUTPUT);
      pinMode(5, OUTPUT);
      pinMode(6, OUTPUT);
      pinMode(7, OUTPUT);
      pinMode(8, OUTPUT);
      pinMode(9, OUTPUT);
      digitalWrite(9, 0);          // isključujemo točku decimalnu
}
void loop() {
      // piši '0'
      digitalWrite(2, 1);          // segment A
      digitalWrite(3, 1);          // segment B
      digitalWrite(4, 1);          // segment C
      digitalWrite(5, 1);          // segment D
      digitalWrite(6, 1);          // segment E
      digitalWrite(7, 1);          // segment F
      digitalWrite(8, 0);          // segment G
      delay(intDelay);
```

```
// piši '1'
digitalWrite(2, 0);        // segment A
digitalWrite(3, 1);        // segment B
digitalWrite(4, 1);        // segment C
digitalWrite(5, 0);        // segment D
digitalWrite(6, 0);        // segment E
digitalWrite(7, 0);        // segment F
digitalWrite(8, 0);        // segment G
delay(intDelay);

// piši '2'
digitalWrite(2, 1);        // segment A
digitalWrite(3, 1);        // segment B
digitalWrite(4, 0);        // segment C
digitalWrite(5, 1);        // segment D
digitalWrite(6, 1);        // segment E
digitalWrite(7, 0);        // segment F
digitalWrite(8, 1);        // segment G
delay(intDelay);

// piši '3'
digitalWrite(2, 1);        // segment A
digitalWrite(3, 1);        // segment B
digitalWrite(4, 1);        // segment C
digitalWrite(5, 1);        // segment D
digitalWrite(6, 0);        // segment E
digitalWrite(7, 0);        // segment F
digitalWrite(8, 1);        // segment G
delay(intDelay);

// piši '4'
digitalWrite(2, 0);        // segment A
digitalWrite(3, 1);        // segment B
digitalWrite(4, 1);        // segment C
digitalWrite(5, 0);        // segment D
digitalWrite(6, 0);        // segment E
digitalWrite(7, 1);        // segment F
digitalWrite(8, 1);        // segment G
delay(intDelay);


// piši '5'
digitalWrite(2, 1);        // segment A
digitalWrite(3, 0);        // segment B
digitalWrite(4, 1);        // segment C
digitalWrite(5, 1);        // segment D
digitalWrite(6, 0);        // segment E
digitalWrite(7, 1);        // segment F
digitalWrite(8, 1);        // segment G
delay(intDelay);
```



```
      // piši '6'
      digitalWrite(2, 1);        // segment A
      digitalWrite(3, 0);        // segment B
      digitalWrite(4, 1);        // segment C
      digitalWrite(5, 1);        // segment D
      digitalWrite(6, 1);        // segment E
      digitalWrite(7, 1);        // segment F
      digitalWrite(8, 1);        // segment G
      delay(intDelay);

      // piši '7'
      digitalWrite(2, 1);        // segment A
      digitalWrite(3, 1);        // segment B
      digitalWrite(4, 1);        // segment C
      digitalWrite(5, 0);        // segment D
      digitalWrite(6, 0);        // segment E
      digitalWrite(7, 0);        // segment F
      digitalWrite(8, 0);        // segment G
      delay(intDelay);

      // piši '8'
      digitalWrite(2, 1);        // segment A
      digitalWrite(3, 1);        // segment B
      digitalWrite(4, 1);        // segment C
      digitalWrite(5, 1);        // segment D
      digitalWrite(6, 1);        // segment E
      digitalWrite(7, 1);        // segment F
      digitalWrite(8, 1);        // segment G
      delay(intDelay);

      // piši '9'
      digitalWrite(2, 1);        // segment A
      digitalWrite(3, 1);        // segment B
      digitalWrite(4, 1);        // segment C
      digitalWrite(5, 1);        // segment D
      digitalWrite(6, 0);        // segment E
      digitalWrite(7, 1);        // segment F
      digitalWrite(8, 1);        // segment G
      delay(intPause);
}
```



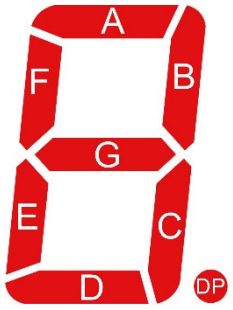
Zadatak 2. Proširi program iz prethodnog zadatka novom serijom znakova, tako da DODATNO slijedno prikazuje i slova od A do F.

Zadatak 3. Proširi program iz prethodnog zadatka novom serijom znakova, tako da DODATNO nakon znakova 0 – F na 7-segmentni LED indikator ispisuješ i „PULA“ ili „HELP.

Zadatak 4. Prouči skraćenu verziju programa kojim će indikator odbrojavati od 0-F. **Ovaj zadatak ne treba realizirati praktično!**

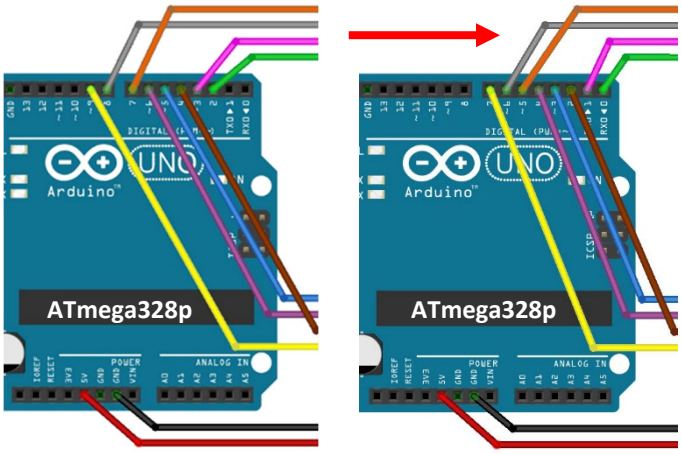
Kòd zadatka

```
int intDelay = 500;    // vrijeme čekanja između ispisa dviju znamenki u ms
int intPause = 4000;   // vrijeme čekanja između ispisa nove serije

// Deklaracija pinova Arduina za spoj na 7 segmentni LCD display
byte s7segment[] = { 2, 3, 4, 5, 6, 7, 8, 9 };

byte digits[][17] = { // Definicija bitova za paljenje određenog segmenta
displaya pritiskom na neku tipku
 //   { A, B, C, D, E, F, G, dp}, // segmenti
      { 1, 1, 1, 1, 1, 1, 0, 0 }, // broj 0
      { 0, 1, 1, 0, 0, 0, 0, 0 }, // broj 1
      { 1, 1, 0, 1, 1, 0, 1, 0 }, // broj 2
      { 1, 1, 1, 1, 0, 0, 1, 0 }, // broj 3
      { 0, 1, 1, 0, 0, 1, 1, 0 }, // broj 4
      { 1, 0, 1, 1, 0, 1, 1, 0 }, // broj 5
      { 1, 0, 1, 1, 1, 1, 1, 0 }, // broj 6
      { 1, 1, 1, 0, 0, 0, 0, 0 }, // broj 7
      { 1, 1, 1, 1, 1, 1, 1, 0 }, // broj 8
      { 1, 1, 1, 1, 0, 1, 1, 0 }, // broj 9
      { 1, 1, 1, 0, 1, 1, 1, 0 }, // slovo A
      { 0, 0, 1, 1, 1, 1, 1, 0 }, // slovo B
      { 1, 0, 0, 1, 1, 1, 0, 0 }, // slovo C
      { 0, 1, 1, 1, 1, 0, 1, 0 }, // slovo D
      { 1, 0, 0, 1, 1, 1, 1, 0 }, // slovo E
      { 1, 0, 0, 0, 1, 1, 1, 0 }, // slovo F
      { 0, 0, 0, 0, 0, 0, 0, 0 }, }; // Ugasi

void setup() {

      for (int i = 0; i<7; i++) {           // Definiraj sve pinove za
            pinMode(s7segment[i], OUTPUT);  // kao display izlazne
      }
}

void loop() {
      for (int i = 0; i<17; i++) {          // Petlja za ispis svih znamenki
            for (int j = 0; j <= 7; j++) {  // Petlja za ispis svih segmenata
                        digitalWrite(s7segment[j], digits[i][j]);
            }
            delay(intDelay);
      }
      delay(intPause);
}
```



Zadatak 5. Modificiraj program iz 1. zadatka gdje 7-segmentni LED indikator odbrojava od 0-9 tako da umjesto naredbi digitalWrite() i pinMode(), koristiš direktno adresiranje porta D. Potrebno je modificirati spoj tako da signalne vodiče za segmente A – H premjestiš za dva pina udesno tako da kreću od pina 0, umjesto pina 2 Arduina. Vidi priložene slike.

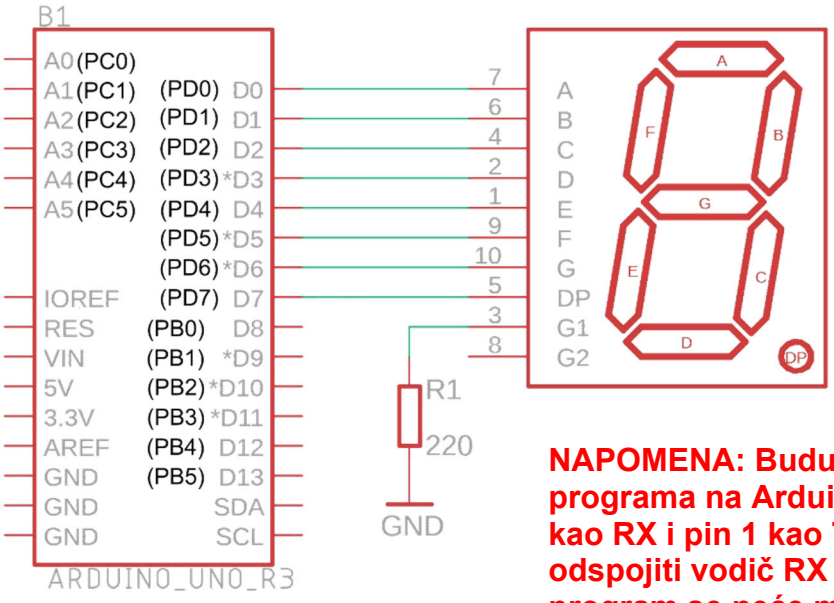
Grafička shema:



Novi raspored pinova:

Arduino Pin	Pin Mikroupravljača ATmega 328p	7 segmentni LED display Pin
0	PD0	7 (A)
1	PD1	6 (B)
2	PD2	4 (C)
3	PD3	2 (D)
4	PD4	1 (E)
5	PD5	9 (F)
6	PD6	10 (G)
7	PD7	5 (DP)
GND	GND	3 i/ili 8

Električna shema:

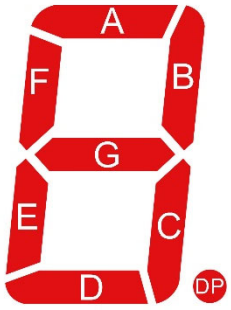


NAPOMENA: Budući da prilikom učitavanja programa na Arduino UNO pločicu pin 0 koristi kao RX i pin 1 kao TX, potrebno je privremeno odspojiti vodič RX sa Arduina. U suprotnom program se neće moći u čitati na Arduino.

Kòd zadatka

```
int intTime = 200;
int intPause = 1000;

void setup() {
      DDRD = 0b11111111;    // sve pinove porta D, što odgovara pinovima
                            // Arduino 0-7 postavljamo kao izlazne
      Serial.begin(9600);
      // Zbog Serial.begin(9600); segment a svijetli slabije jer se njom
      // inicijalizira serijska komunikacija koja koristi pinove 0 i 1
      // Probaj isključiti funkciju Serial.begin(9600) tako da ju
      // komentiraš pomoću dvije kose crte (//) ispred funkcije
}
```

```
void loop() {
      // piši '0'
      //segment hgfedcba
      PORTD = 0b00111111;   // u PORTD registar upisujemo vrijednosti
                            // svih 8 pinova za 8 segmenata
      delay(intTime);

      PORTD = 0b00000110;   // piši '1';
      delay(intTime);       // ako imaš indikator sa zajedničkom anodom
                            // koristiti ~0b00000110

      PORTD = 0b01011011;   // piši '2'
      delay(intTime);

      PORTD = 0b01001111;   // piši '3'
      delay(intTime);

      PORTD = 0b01100110;   // piši '4'
      delay(intTime);

      PORTD = 0b01101101;   // piši '5'
      delay(intTime);

      PORTD = 0b01111101;   // piši '6'
      delay(intTime);

      PORTD = 0b00000111;   // piši '7'
      delay(intTime);

      PORTD = 0b01111111;   // piši '8'
      delay(intTime);

      PORTD = 0b01101111;   // piši '9'
      delay(intPause);
}
```



Zadatak 6. Proširi program iz prethodnog zadatka novom serijom znakova, tako da DODATNO slijedno prikazuje i slova od A do F.

Zadatak 7. Proširi program iz prethodnog zadatka novom serijom znakova, tako da DODATNO nakon znakova 0 – F na 7-segmentni LED indikator ispisuješ i „PULA“ ili „HELP“.

Zadatak 8. Proširi spoj tako da dodaš potenciometar. Pomoću njega treba regulirati brzinu ispisa znakova u intervalu 100 ms do 1000 ms. Pauza između serija treba biti dvostruko dulja nego između pojedinih znakova u seriji.

Zadatak 9. Zamijeni sedam segmentni indikator sa zajedničkom katodom novim indikatorom sa zajedničkom anodom (zajednički Vcc). Umjesto na zajednički GND, indikator treba također preko otpornika spojiti na zajednički VCC (slika 3 u uvodu vježbe). Modificiraj program tako da se znakovi korektno ispisuju.

Zadatak 10. Proširi spoj dodavanjem jednog tipkala (ili prekidača ukoliko se vježba radi online). Doradi program tako da bude univerzalan za korištenje i sedam segmentnog indikatora sa zajedničkom katodom i indikatora sa zajedničkom anodom, tako da koristiš jednu varijablu `unsigned char common` kojom će definirati koji indikator koristiš, a kojom možeš upravljati pomoću tipkala. Moguće upotrijebiti funkciju danu u uvodu vježbe: `unsigned char DigitTo7SegEncoder(unsigned char digit, unsigned char common)` .

Pitanja za provjeru znanja:

1. Od čega se sastoji 7-segmentni LED indikator?
2. Koje su prednosti, a koji nedostaci 7-segmentnog LED indikatora.
3. Kakvi mogu biti 7-segmentni LED indikatori s obzirom na zajedničku elektrodu?
4. Objasni koje pinove ima 7-segmentni indikator sa zajedničkom anodom? Skiciraj električku shemu takvog indikatora.
5. Objasni koje pinove ima 7-segmentni indikator sa zajedničkom katodom? Skiciraj električku shemu takvog indikatora.
6. Kako treba spojiti pinove 7-segmentnog indikatora sa zajedničkom anodom na mikroupravljač da bi segmenti mogli svijetliti?
7. Kako treba spojiti pinove 7-segmentnog indikatora sa zajedničkom katodom na mikroupravljač da bi segmenti mogli svijetliti?
8. Zašto je potrebno ugraditi otpornike između pinova mikroupravljača i pinova 7-segmentnog indikatora i kako izračunati njihovu vrijednost?
9. Kako smo na LV spojili otpornik za zaštitu od prevelike struje kroz 7-segmentni LED indikator i zašto? Što je nedostatak takvog spoja?
10. Koji 7-segmentni LED indikator si koristio na LV, sa zajedničkom anodom ili katodom? Po čemu se to može prepoznati?
11. Kako bi napisao programski kod da na 7-segmentnom LED indikatoru sa zajedničkom katodom ispišeš broj 3?
12. Kako bi napisao programski kod da na 7-segmentnom LED indikatoru sa zajedničkom anodom ispišeš broj 7?
13. Navedi vezu pinova ATmega328p i Arduino UNO pločice!
14. Čemu služi DDR registar? Objasni što znače upisane vrijednosti 0 i 1?
15. Čemu služi PORT registar? Objasni što znače upisane vrijednosti 0 i 1?
16. Čemu služi PIN registar? Objasni!
17. Napiši program koji će pomoću direktnog adresiranja registara porta na 7-segmentnom LED indikatoru sa zajedničkom katodom ispisati broj 7?
18. Napiši program koji će pomoću direktnog adresiranja registara porta na 7-segmentnom LED indikatoru sa zajedničkom anodom ispisati broj 3?
19. Kako se jednostavno može koristiti ista binarna kombinacija za ispis nekog broja i na 7-segmentni indikator sa zajedničkom anodom i zajedničkom katodom?
20. 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. CircuitsToday, 7-segment LED display, http://www.circuitstoday.com (pregledano 19.01.2019.)
2. El-Pro-Cus, Types of 7 Segment Displays and Controlling Ways, https://www.elprocus.com/types-of-7-segment-displays-and-controlling-ways/ (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. TR3SDLAND, Diagramas de pines de los micros usados en Arduino, https://www.tr3sdland.com/2013/02/arduino-pinout/, (pregledano 19.01.2019.)
5. AVR tutorials, ATMega16 AVR Microcontroller Seven Segment Digital Clock , http://www.avr-tutorials.com/projects/atmega16-based-digital-clock, (pregledano 20.01.2019.)