

Nastavni predmet:	Ugradbeni računalni sustavi
Vježba: 10	Upravljanje pomoću IC senzora 2
Cilj vježbe:	Upravljanje vanjskim elementima spojenih na Arduino pomoću IC senzora i daljinskog upravljača

Upute

Sve zadatke spremi na USB, a u bilježnici za sve zadatke napiši:

- postupak izrade programa
- objašnjenje korištenih naredbi
- dobivene rezultate po točkama
- odgovoriti u bilježnicu na postavljena pitanja vezana uz ovu vježbu
- Ukoliko u kòdu postoji greška, korigiraj i objasni!

Ova vježba nastavlja se na LV04 - Upravljanje vanjskim elementima pomoću IC senzora!

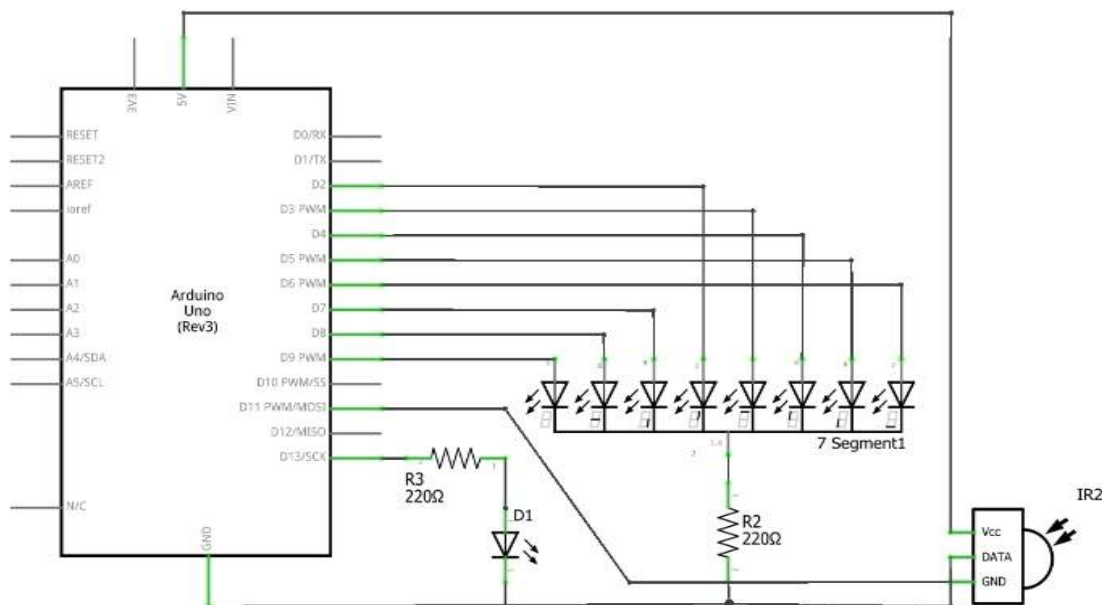
Prije nastavka s izvođenjem ove vježbe potrebno je odraditi sve zadatke iz prethodne vježbe: „URS-LV09 - Upravljanje vanjskim elementima pomoću IC senzora“.

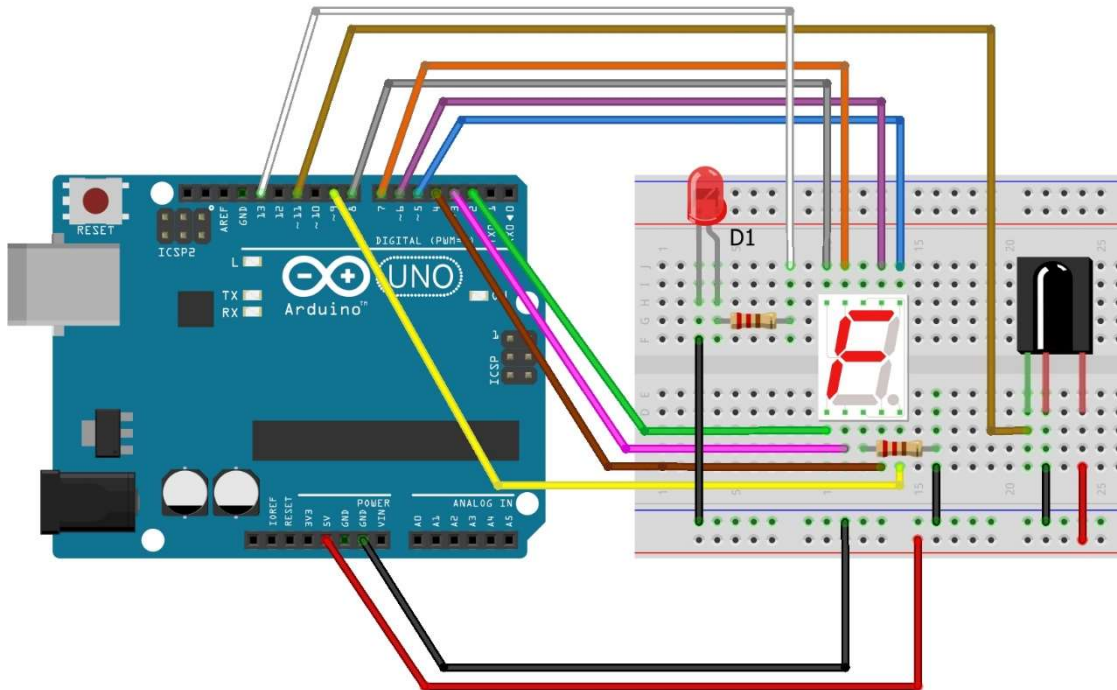
Zadatak 1. Prouči spoj i shemu, te kod zadatka koji je napravljen sa slijedećim funkcijama:

- Pritiskom na tipke 0-9 treba na 7-segmentnom zaslonu prikazati odgovarajuće brojeve.
- U trenutku pritiska LED dioda D1 spojena na pin 13 treba zasvijetliti u trajanju 100 ms kao potvrda primitka koda.
- Pritiskom na tipku CH- treba isključiti display, a pritiskom na tipku CH+ treba uključiti display
- Nakon 5 sekundi automatski isključi display

NAPOMENA: ovaj zadatak nije potrebno realizirati. Treba ga proučiti!

Električna shema:



Grafička shema:**Kòd zadatka**

```
#include <IRremote.h>
```

```
int recvLED = 13;
```

```
int recvPin = 11;
```

```
IRrecv irReceiver(recvPin);
```

```
decode_results results;
```

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

```
byte digits[][12] = { // Definicija bitova za paljenje određenog
segmenta displaya pritiskom na neku tipku
```

```
  { 1, 1, 1, 1, 1, 1, 0, 0}, // broj 0
  { 0, 0, 1, 1, 0, 0, 0, 0}, // broj 1
  { 1, 1, 0, 1, 1, 0, 1, 0}, // broj 2
  { 0, 1, 1, 1, 1, 0, 1, 0}, // broj 3
  { 0, 0, 1, 1, 0, 1, 1, 0}, // broj 4
  { 0, 1, 1, 0, 1, 1, 1, 0}, // broj 5
  { 1, 1, 1, 0, 1, 1, 1, 0}, // broj 6
  { 0, 0, 1, 1, 1, 0, 0, 0}, // broj 7
  { 1, 1, 1, 1, 1, 1, 1, 0}, // broj 8
  { 0, 1, 1, 1, 1, 1, 1, 0}, // broj 9
  { 1, 1, 1, 1, 1, 1, 0, 0}, // Upali
  { 0, 0, 0, 0, 0, 0, 0, 0},}; // Ugasi
```

```
const unsigned long decodeHASH[] = { // Definicija koda za
                                     // određenu tipku
    0xFF6897, // broj 0
    0xFF30CF, // broj 1
    0xFF18E7, // broj 2
    0xFF7A85, // broj 3
    0xFF10EF, // broj 4
    0xFF38C7, // broj 5
    0xFF5AA5, // broj 6
    0xFF42BD, // broj 7
    0xFF4AB5, // broj 8
    0xFF52AD, // broj 9
    0xFFE21D, // Upali
    0xFFA25D,}; // Ugasi

unsigned long lastTime = 0; // Varijabla za brojanje vremena

void setup(){
    pinMode(recvLED, OUTPUT); // LED dioda za izlaz
    digitalWrite(recvLED, LOW); // LED dioda na početku ugašena
    Serial.begin(9600);
    irReciver.enableIRIn(); // Pokreni receiver
    for (int i=0; i<8; i++){
        pinMode(s7segment[i], OUTPUT);} // Definiraj sve pinove
                                         // za display kao izlaz

void loop(){
    unsigned long recv_value;
    if (irReciver.decode(&results)) {
        recv_value = results.value;
        if (recv_value != 0xFFFFFFFF) {
            digitalWrite(recvLED, HIGH); // Upali LED diodu svaki put
                                         // kada pritisnemo tipku

            delay(100);
            digitalWrite(recvLED, LOW);
            Serial.println(recv_value, HEX);

            for (int i=0; i<12; i++){ // Idi kroz petlju dok se
                                     // kòd ne poklapa
                if ( recv_value == decodeHASH[i] ){
                    lastTime = millis();
                    for (int j=0; j<=8; j++)
                        digitalWrite(s7segment[j], digits[i][j]);
```

```

        break;}} // Prekini petlju kada se tipka i
                // kòd poklope
    irReciver.resume();} // Očitaj sljedeću vrijednost

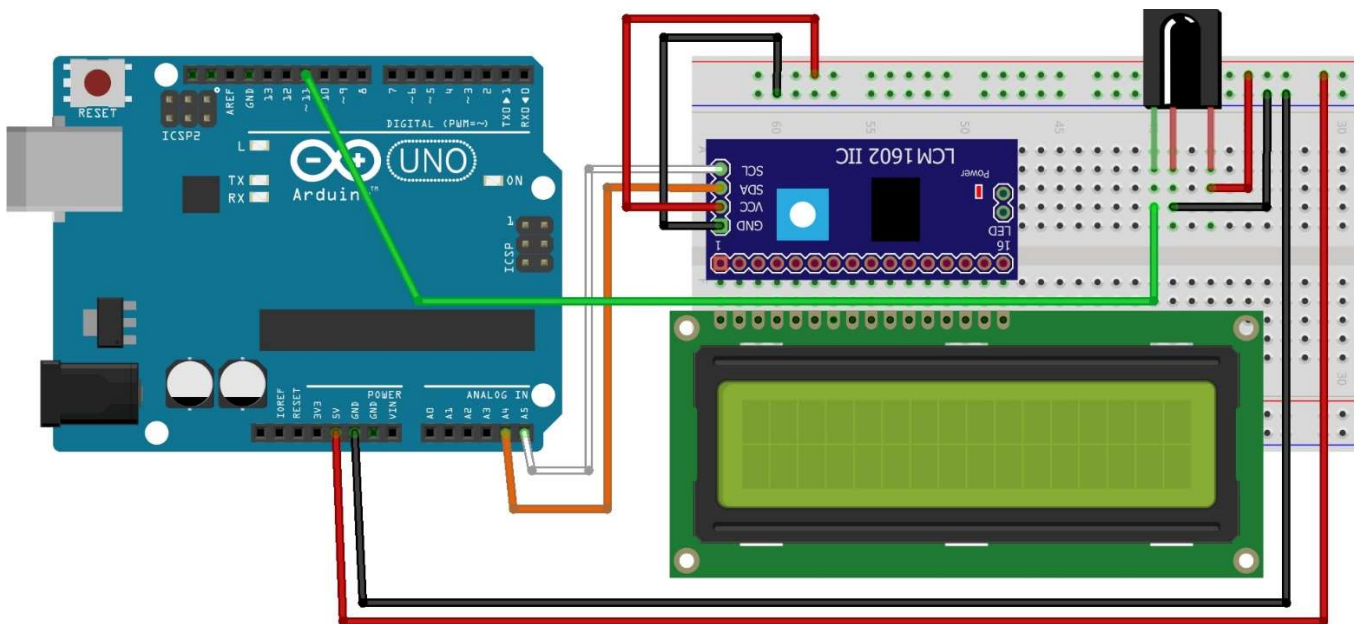
if ( millis() - lastTime >= 5000 ){ // Isključi cijeli display
                                // nakon 5 sekundi
    for (int k=0; k<=8; k++)
        digitalWrite(s7segment[k], LOW);
    lastTime = 0;}} // Resetiraj timer

```

Zadatak 2. Prouči zadatak 6, te doradi spoj i program tako da umjesto ispisa na 7 segmentni display, brojeve ispisuješ na LCD display. Potrebno je zadržati kompletnu funkcionalnost – točke a) do d).

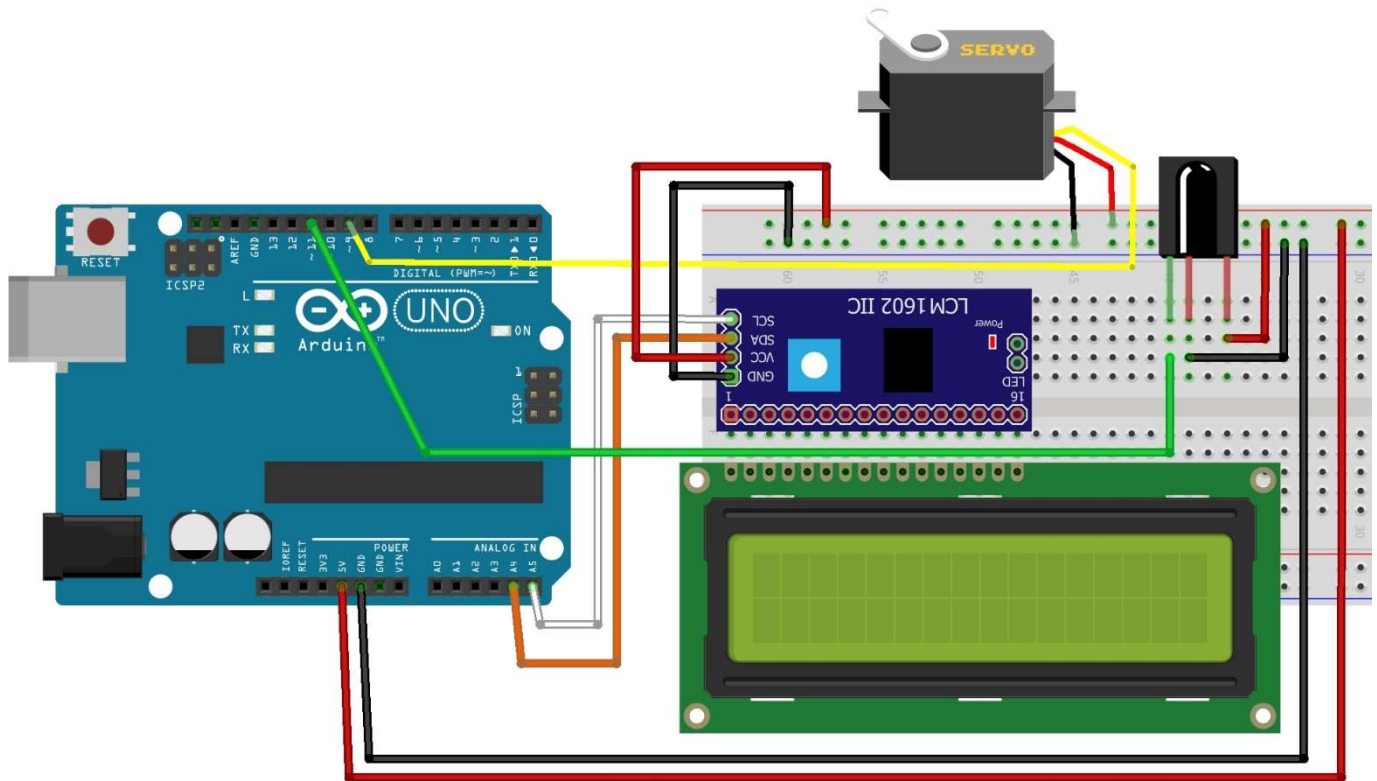
Proširi program tako da:

- pritiskom na tipku CH- treba isključiti display, a pritiskom na tipku CH+ treba uključiti display
- u trenutku pritiska LED dioda D1 spojena na pin 13 treba zasvijetliti u trajanju 100 ms kao potvrda primitka koda – diodu ne treba spajati jer je integrirana na Arduino pločici.
- pritiskom na tipku + povećavaš broj, a pritiskom na tipku – smanjuješ broj.
- pritiskom na tipku << smanjivati broj za 10, a pritiskom na tipku >> povećavati broj za 10
- ograničiti vrijednost broja 0 – 180, a u slučaj da je broj izvan granica, na display-u napisati odgovarajuću poruku
- treba omogućiti unos 1-znamenkastog, 2-znamenkastog i 3-znamenkastog broja. U slučaju 2- i 3-znamenkastog broja, vrijeme između pritiska dviju tipki mora biti kraće od 2 sekunde.



Zadatak 3. Proširi spoj i kôd iz prethodnog zadatka tako da priključiš servo motor, a brojeve koje uneseš daljinskim upravljačem, koristiš za pozicioniranje motora.

Servo motor obrađen je u predmetu Mikroupravljači, „MU-LV14-DC i Servo motori“



Kôd zadatka za servo motor

```
#include <Servo.h>

Servo myservo; // kreiraj servo objekt za kontroliranje servo motora
int pos = 0;    // varijabla za spremanje pozicije servo motora

void setup() {
  myservo.attach(9); // servo postavlja na digitalni pin 9
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // servo kreće od 0 do 180
    //stupnjeva u koracima po 1 stupanj
    myservo.write(pos);                // postavljanje servo motora na
    //poziciju pos (varijabla)
    delay(15);                          // čekanje 15ms za pozicioniranje motora
  }
  for (pos = 180; pos >= 0; pos -= 1) { // servo kreće od 180 do 0
    //u koracima po 1 stupanj
    myservo.write(pos);                // postavljanje servo motora na
    //poziciju pos (varijabla)
    delay(15);                          // čekanje 15ms za pozicioniranje motora
  }
}
```