

Predmet:	MIKROUPRAVLJAČI
Vježba: 04	Arduino – Ispis na alfanumerički LCD zaslon
Ishodi vježbe:	Ispisati niz znakova na željenu lokaciju LCD-a povezanim s mikroupravljačem paralelnim spojem ili putem I2C komunikacije. Detektirati I2C adresu LCD zaslona, kreirati i ispisati proizvoljni znak na LCD

Upute za izradu pripreme 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 nastavku, 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 ...

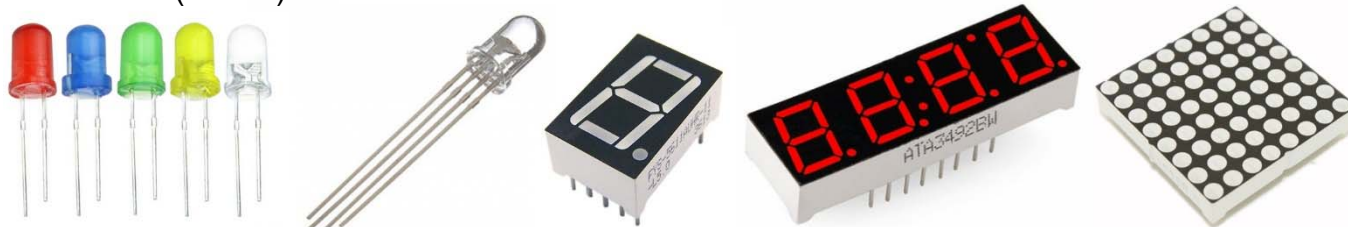
Upute za izradu vježbe:

- 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 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 neispravna (pregorena LED, oštećen kontakt tipkala, prekinut vodič...). Ukoliko se sklop ponaša drugačije od očekivanog, predvidjeti i tu mogućnost i pokušati zamijeniti komponentu drugom.
- 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

INDIKATORI

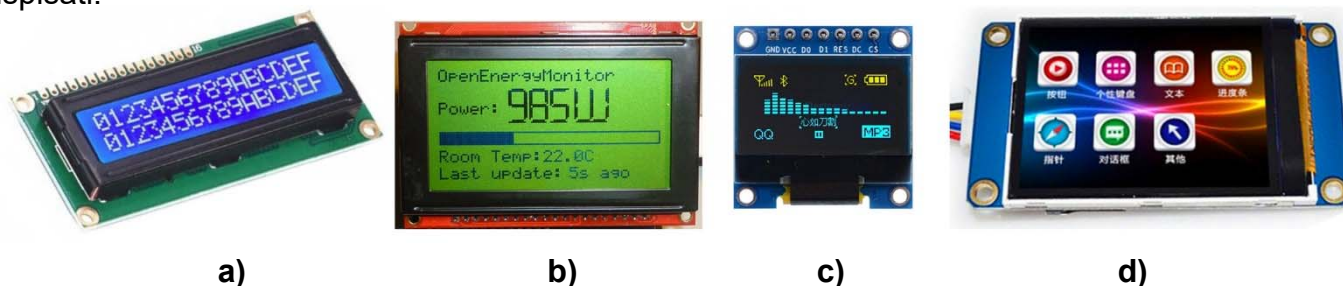
Mikroupravljač prima informacije od različitih senzora iz fizičkog svijeta i na temelju zadane funkcije upravljanja različitim **izvršnim elementima** (aktuatorima) te tako kontrolira proces za koji je namijenjen. Osim upravljanja aktuatorima, sustav često treba korisniku davati informacije o stanju sustava: sustav je ispravan ili neispravan, iznos fizikalne veličine na određenom senzoru (temperatura, tlak,...), upozorenje u slučaju opasnosti i slično. Za tu svrhu, koriste se indikatori kao što su LED ili RGB LED diode, sedam segmentni LED zasloni, 8x8 LED matrice i sl. (slika 1).



Slika 1. Jednostavniji indikatori

Složeniji indikatori, omogućuju prikaz slova i grafike, te prikaz grafike u boji. Najčešće su to LCD indikatori koji se dijele se na alfanumeričke LCD zaslone (slika 2a), grafičke LCD zaslone (slika 2b), OLED LCD zaslone (slika 2c) i TFT LCD zaslone (slika 2d). U najnovije vrijeme popularni su zaslone osjetljivi na dodir. Oni omogućuju podešavanje i upravljanje uređajem pomoću izbornika.

Ovakvi LCD zaslone imaju ugrađen mikroupravljač koji upravlja ispisom, dok mu mikroupravljač iz našeg sustava određenim protokolom šalje informaciju što, kako i gdje treba ispisati.



Slika 2. Složeniji indikatori – varijante LCD zaslona [1], [2]

Alfanumerički LCD zaslon (slika 2a i 3) ograničen je na prikaz alfanumeričkih znakova (slova, brojeva, interpunkcijskih i matematičkih znakova). Omogućuju i kreiranje ograničenog broja znakova od strane korisnika, primjerice simbol za bateriju kojom se može prikazati stanje napunjenosti baterije i slično (slika 3).



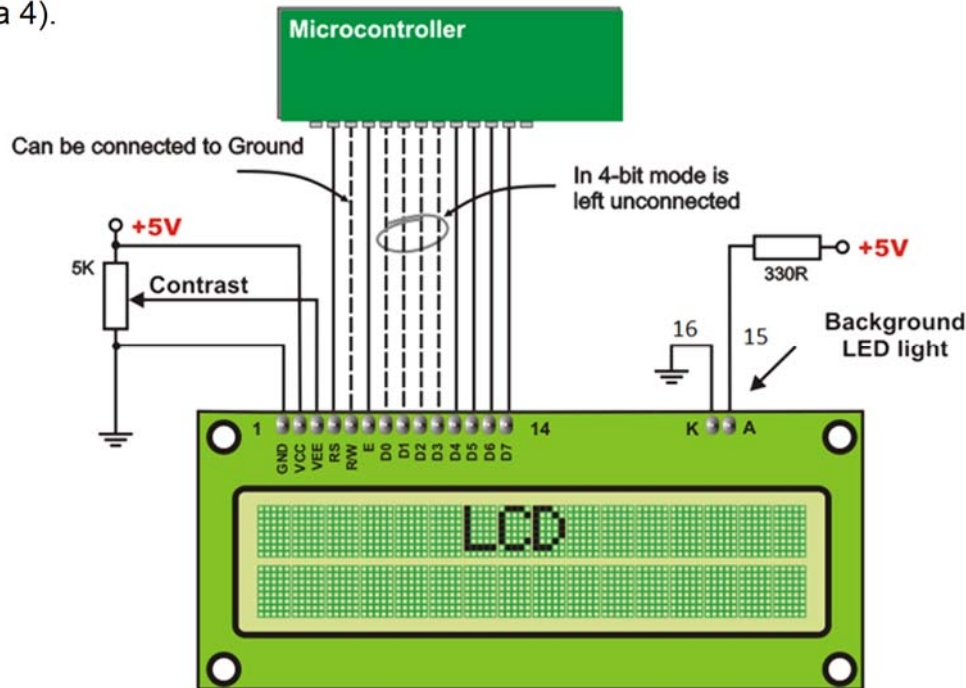
Slika 3. Alfanumerički i korisnički definirani znakovi na alfanumeričkom LCD zaslonu [3]

Ispis i manipulacija znakova izvodi se u redovima i stupcima. Najčešće se koriste zaslone s 2 retka i 16 znakova, iako ima i drugih varijanti 1x16, 2x16, 4x16, 4x20,...

Grafički LCD zaslon (slika 2b), **OLED LCD** (slika 2c) i **TFT LCD zaslon** (slika 2d) sastoje se od matrice - piksela na kojima se može crtati i ispisivati bilo koji oblik (tekst, linije, krugovi, kvadratići, gumbi i sl.) pa je informacija koju pružaju puno bogatija i preglednija. Rezolucije kreću od 128x64 piksela naviše.

ALFANUMERIČKI 16x2 LCD ZASLON

Na vježbama koristimo 16x2 alfanumerički LCD zaslon čiji znakovi imaju matricu 5x8 točaka (slika 4).



Slika 4. Pinovi LCD zaslona i povezivanje na mikroupravljač [4], [5]

Bez obzira na broj redaka i stupaca (1x16, 2x16, 4x16, 4x20), za spajanje na mikroupravljač, koriste se slijedeći pinovi [6]:

DB0 - DB7 Podatkovne linije kojima se obavljaju operacije pisanja i čitanja znakova, te pisanje instrukcije

Vss, Vdd Pinovi za napajanje

R/W (Read/Write) Pin za upravljanje pisanjem (ili čitanjem) u LCD

Uz R/W = 0, pisanje (može se spojiti na GND jer ćemo u LCD samo pisati)

Uz R/W = 1, čitanje

RS (Register select) Pin za odabir upisa u Instrukcijski registar ili Data Registar.

Uz RS = 0, D0 – D7 predstavljaju naredbu (može se samo pisati)

Uz RS = 1, D0 – D7 predstavljaju podatak (može se čitati i pisati)

E (Enable) Pin postavlja LCD u neaktivno stanje.

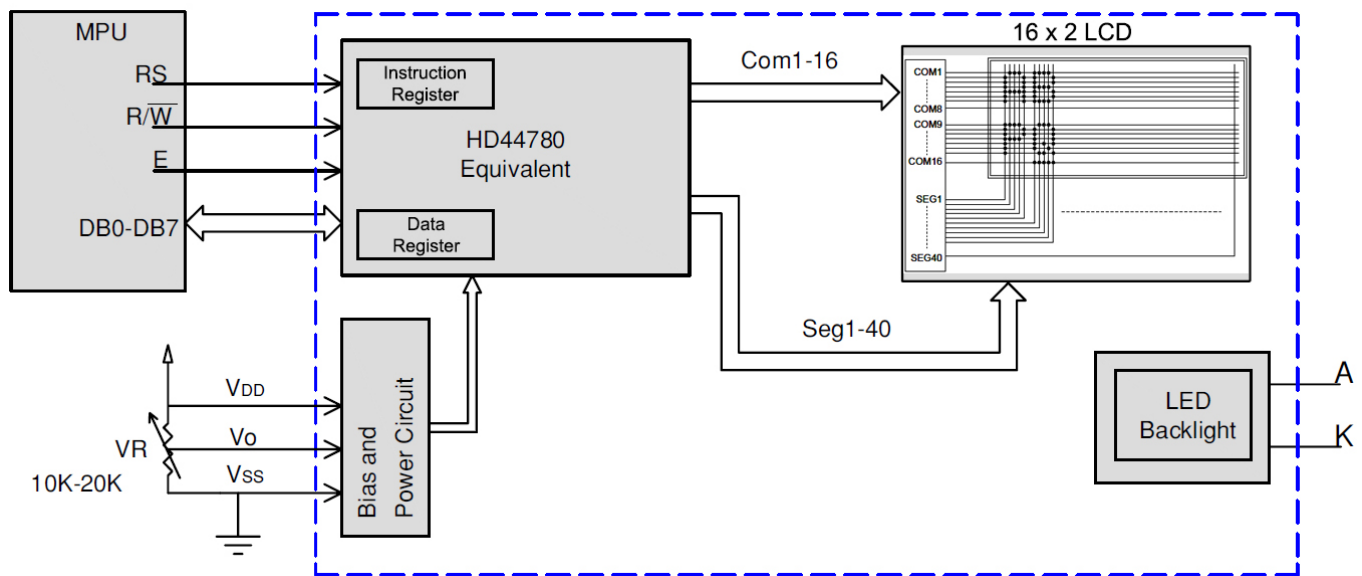
Uz E=0, LCD neaktivan

Uz E=1, LCD procesira ulazne signale

Vee Pin za podešavanje kontrasta, spajanjem srednjeg izvoda potenciometra

Ovo je paralelni način povezivanja LCD-a s mikroupravljačem. Radi uštede pinova i vodiča, umjesto svih 8 podatkovnih linija, moguće je mikroupravljač povezati samo sa 4 linije (D4-D7). Tada se upis naredbi i podataka obavlja u dva koraka, pa je prijenos sporiji.

Zaslonom upravlja Hitachi HD44780 kontroler, koji upravlja u skladu s podacima koje prima u dva registra: **Instrukcijski registar (IR)** i **podatkovni registar (Data register - DR)**.



Slika 5. Blok shema LCD zaslona [7]

Instrukcijski registar (IR) pohranjuje instrukcijski kod, npr: obriši zaslon, pomakni kursor na početnu poziciju, uključi/isključi osvjetljenje, pomicanje kursora ili sadržaja zaslona, uključuje/isključuje kursor, prihvaća kod za inicijalizaciju LCD-a i načina rada (npr. 8-bitna ili 4-bitna komunikacija), sadržava adresu u DDRAM memoriji (Display Data RAM) na koju poziciju želimo upisati znak (0x00 – 0x0F za prvi red i 0x40 – 0x4F za drugi red), itd.

Podatkovni registar - Data registar (DR) privremeno pohranjuje podatak (znak) koji treba pohraniti u DDRAM ili CGRAM (Character Generator RAM).

Da bi ispisali slovo ili znak na LCD, potrebno je slijedeće:

1. Izvršiti inicijalizaciju LCD-a
2. Poslati željenu lokaciju u IR (DDRAM adresu)
3. Poslati ASCII kod željenog znaka

LCD će prikazati slovo koje odgovara poslanom ASCII kodu, a brojač adrese (Address Counter) će se automatski povećati ili umanjiti za 1 (ovisno o inicijalizaciji). Ispis stringa moguć je slanjem znaka za znakom [8].

Upravljanje LCD zaslonom pomoću mikroupravljača odvija se putem funkcija (naredbi) koje uz pomoć odgovarajuće biblioteke (LiquidCrystal.h) pretvaraju naredbe u operacijski kod razumljiv LCD zaslonu.

Primjer ispisa na LCD zaslon uz 4-bitnu paralelnu komunikaciju [9]

```
// Primjer sa https://www.arduino.cc/en/Reference/LiquidCrystal
// dodaj biblioteku za rad s LCD-om
#include <LiquidCrystal.h>

// kreiranje objekta i inicijalizacija komunikacije navođenjem
// koji pin LCD-a je spojen s kojim pinom mikroupravljača - VAŽNO!!!
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

```

void setup() {
    // inicijalizacija broja stupaca i redova LCD-a:
    lcd.begin(16, 2);

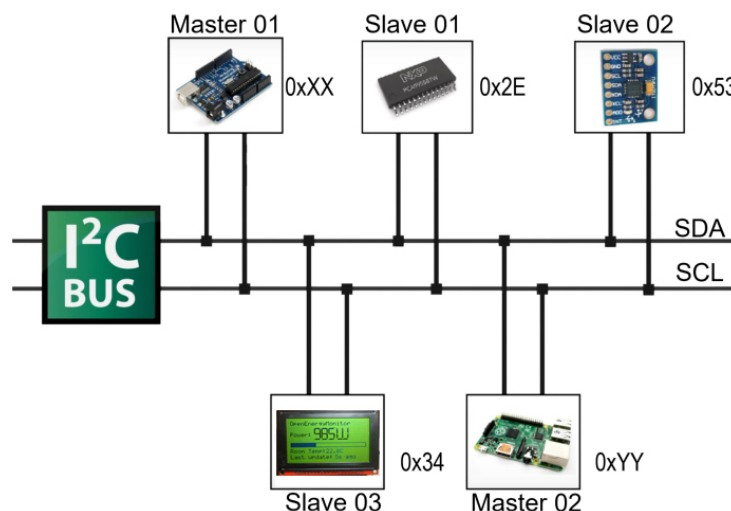
    // Ispiši poruku na LCD.
    lcd.print("hello, world!");
}

void loop() {
    // postavljanje kursora na lokaciju 0,0 (stupac, redak)
    // (napomena: redak 1 je drugi red jer brojanje počinje od 0:
    lcd.setCursor(0, 1);
    // ispisuje broj sekundi od trenutka pokretanja Arduina:
    lcd.print(millis() / 1000);
}

```

I2C komunikacija

Osim paralelnom vezom, za koju je potrebno minimalno 11 vodiča, LCD zaslonom moguće je upravljati i **serijskom vezom**. Prednost tog načina spajanja je manji broj vodova: **UART i I2C protokol zahtijevaju 2 vodiča**, dok **SPI protokol zahtijeva 3 + 1 vodič**. Pri tom je brzina komunikacije manja, ali je još uvijek dovoljno brza za većinu senzora i za LCD. Svaki od ta tri protokola ima svoje prednosti i nedostatke, a u nastavku će ukratko biti objašnjena I2C komunikacija budući da **na našem LCD-u postoji adapter (*Backpack*) koji pretvara serijsku I2C komunikaciju u paralelnu**. Ostali protokoli nisu tema ove vježbe.



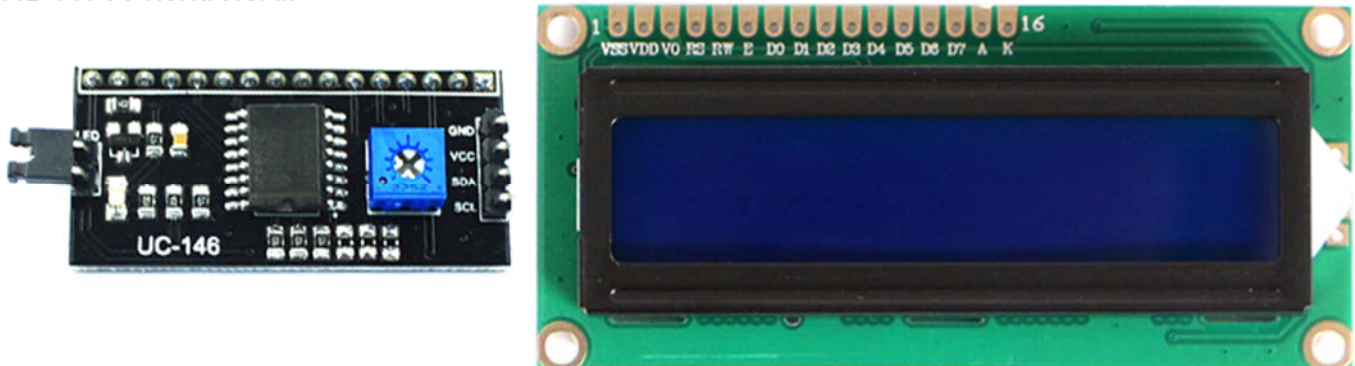
Slika 6. Komunikacija pomoću I2C sabirnice [10]

Svojstva I2C ili I²C (Inter-Integrated Circuit) sabirnice:

- Omogućuje povezivanje više uređaja sa samo dva vodiča **SDA** (Serial Data) i **SCL** (Serial Clock)
- Master - Slave koncept: svaki uređaj na sabirnici ima jedinstvenu adresu tako da Master uređaj može putem adrese odabrati s kojim će Slave uređajem komunicirati (adresa LCD-a na LV je 0x27 ili 0x3F)
- Komunikacija je 8 bitna, serijska, sinkrona i bidirekcijska
- Uz 7-bitno adresiranje moguće do 128 Slave uređaja na sabirnici
- Koristi se za povezivanje sporijih perifernih uređaja na relativno kratkim udaljenostima

PCF8574 8-Bit I/O Expander za I2C sabirnicu

Na LCD se može zalemiti I2C adapter (*Backpack*) koji je baziran na čipu PCF8574. Uloga mu je pretvaranje serijskih podataka poslanih I2C protokolom u paralelnu komunikaciju kojom komunicira LCD. Može se koristiti sa 16x2, 16x4 i 20x4 i drugim LCD-ima baziranim na Hitachi HD44780 kontroleru.



Slika 7. I2C adapter (Backpack) i alfanumerički LCD zaslon

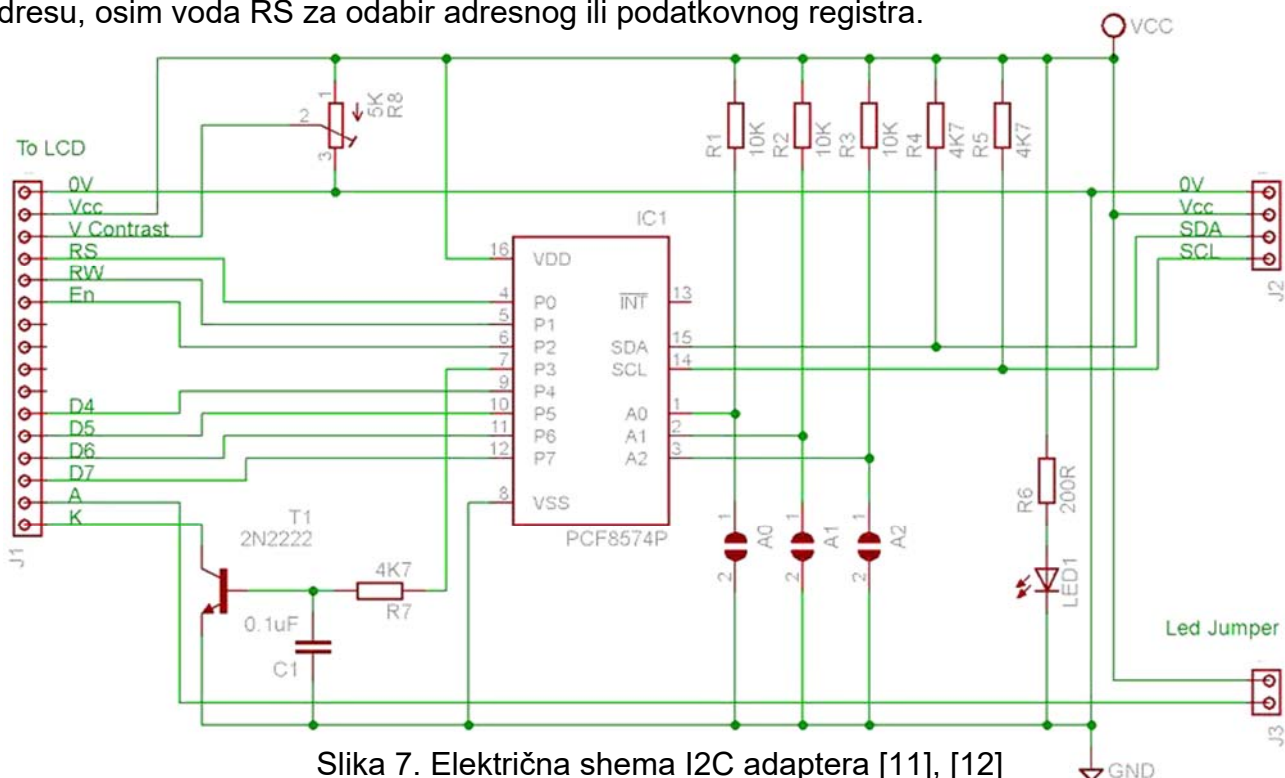
Električna shema dana je na slici 8. Na njoj se mogu vidjeti uloge pojedinih komponenata:

Plavi trimer potencijometar R8 služi za podešavanje kontrasta zaslona. Budući da Vcc može varirati, moguće je da položaj potencijometra na jednom mikroupravljaču neće odgovarati na drugom. Ukoliko nema znakova na LCD-u, pokušati podesiti kontrast.

Kratkospojnik J3 služi za uključivanje pozadinskog osvjetljenja zaslona.

Lemna mjesta A0, A1 i A2 služe za promjenu adrese, u slučaju da priključujemo više LCD zaslona, ili je postojeća 0x27 zauzeta. Prva četiri bita adrese su podešena tvornički 0010 0, a zadnja tri definiramo lemnim mjestima. Ako kratko spojimo lemno mjesto A0, to znači da smo liniju A0 spojili na GND, tj. na logičku 0. Kad su sva tri lemna mjesta odspojena, to znači 111, pa je adresa 0010 0111 = 0x27. Crveno označenu 0 treba ubaciti prilikom kalkulacije adrese.

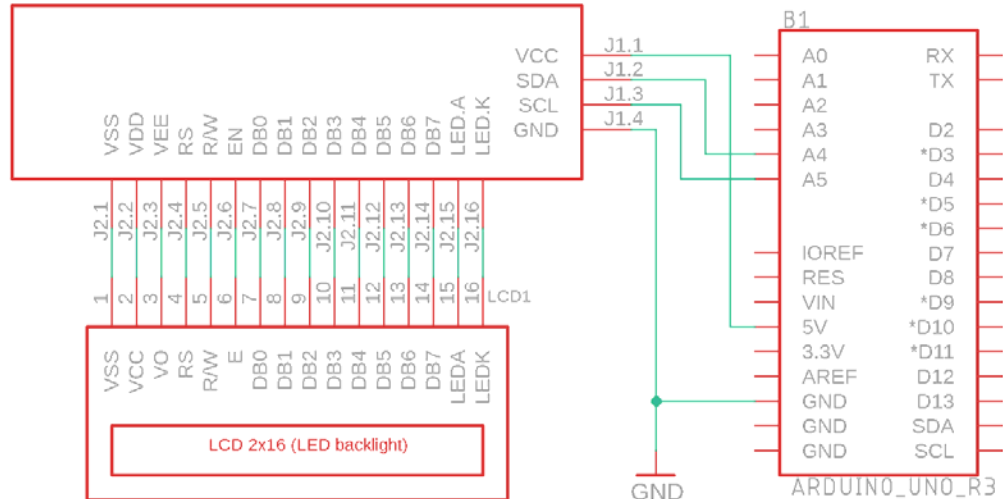
Napomena: iako govorimo o I2C adresi LCD-a, zapravo se adresa odnosi na adresu adaptera (*Backpack-a*) baziranog na PCF8574 integriranom krugu. LCD u paralelnoj komunikaciji nema adresu, osim voda RS za odabir adresnog ili podatkovnog registra.



Slika 7. Električna shema I2C adaptera [11], [12]

Zadatak 1. Spoji LCD zaslon preko I2C sabirnice na Arduino. S adrese <https://playground.arduino.cc/Main/I2cScanner> preuzmi *I2C Scanner* (ne treba prepisivati programski kod u Zadatku 1!). Pomoću njega odredi I2C adresu spojenog LCD zaslona. **I2C Scanner** omogućuje pretraživanje i ispis adresa svih uređaja spojenih na I2C sabirnicu. Ne ispisuje koji je uređaj na kojoj adresi, već samo koje I2C adrese su aktivne na I2C sabirnici.

Električna shema: LCD-I2C-CONVERTER



Kòd zadatka

```
// i2c_scanner
//
// This program (or code that looks like it)
// can be found in many places.
// For example on the Arduino.cc forum.
// The original author is not know.
//
// This sketch tests the standard 7-bit addresses
// from 0 to 127. Devices with higher bit address
// might not be seen properly.
//
// Adapted to be as simple as possible by Arduino.cc user Krodal
//
// June 2012
// Using Arduino 1.0.1
```

```
#include <Wire.h>
```

```
void setup()
{
    Wire.begin();

    Serial.begin(115200);
    Serial.println("\nI2C Scanner");
}
```

```
void loop()
{
    byte error, address;
    int nDevices;

    Serial.println("Scanning...");
```

```

nDevices = 0;
for (address = 0; address <= 127; address++)
{
    // The i2c_scanner uses the return value of
    // the Write.endTransmission to see if
    // a device did acknowledge to the address.
    Wire.beginTransmission(address);
    error = Wire.endTransmission();

    if (error == 0)
    {
        Serial.print("I2C device found at address 0x");
        if (address<16)
            Serial.print("0");
        Serial.print(address, HEX);
        Serial.println(" !");
        delay(1000);

        nDevices++;
    }
    else if (error == 4)
    {
        Serial.print("Unknow error at address 0x");
        if (address<16)
            Serial.print("0");
        Serial.println(address, HEX);
    }
}
}

```

Zadatak 2. Ne mijenjajući prethodni spoj, na LCD zaslonu ispiši „Hello world“.

Kòd zadatka

```

#include <Wire.h>           // dodaj biblioteku funkcija za I2C komunikaciju
#include <LiquidCrystal_I2C.h> // dodaj biblioteku funkcija za LCD s I2C
                             // komunikacijom
LiquidCrystal_I2C lcd(0x27, 16, 2); // definiranje I2C adrese LCD-a (0x27) i
                                     // i veličine LCD-a 16 znakova u 2 retka
void setup() {
    // ovdje se upisuje kòd za inicijalizaciju koji se izvršava samo jednom:
    lcd.begin();           // inicijaliziranje LCD-a
    lcd.backlight();       // uključivanje pozadinskog osvjetljenja LCD-a
    lcd.setCursor(0, 0);   // postavljanje kursora na lokaciju 0,0
                             // (stupac,redak)
    lcd.print("Hello World"); // ispisivanje poruke
}

void loop() {
    // ovdje se upisuje glavni kòd programa koji se kontinuirano izvršava u
    // petlji.
}

```


Zadatak 3. Proširi kod iz prethodnog zadatka tako da na LCD u drugi red ispišeš svoje ime i prezime bez hrvatskih znakova.

Kòd zadatka

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
    lcd.begin();
    lcd.backlight();
    lcd.setCursor(0, 0);
    lcd.print("Hello World"); // ispisivanje poruke
    lcd.setCursor(0, 1);      // postavljanje kursora na lokaciju 0,1
                              // (stupac, redak)
    lcd.print("Hello World");
}

void loop() {
    // ovdje se upisuje glavni kod programa koji se kontinuirano izvršava u
    // petlji:
}
```

Zadatak 4. Proširi program iz prethodnog zadatka tako da poruke „Hello world“ i tvoje Ime i prezime trepću svake sekunde.

Kòd zadatka

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
    lcd.begin();
    lcd.backlight(); // uključivanje pozadinskog osvjetljenja LCD-a
    lcd.setCursor(0, 0);
}

void loop() {
    lcd.backlight(); // uključivanje pozadinskog osvjetljenja LCD-a
    lcd.print("Hello World");
    delay(1000);      // pauza 1000 ms
    //lcd.noBacklight(); // isključivanje pozadinskog osvjetljenja LCD-a
    lcd.clear();      // brisanje zaslona
    delay(1000);      // pauza 1000 ms
}
```

Zadatak 5. Testiraj kako bi na LCD zaslonu, u prvom retku ispisao decimalni broj 1023 u binarnom obliku, a u drugom retku u heksadecimalnom obliku.

Kòd zadatka

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
    int zz = 1023;
    lcd.begin();
    lcd.backlight();
    lcd.setCursor(0, 0);

    lcd.print(zz, BIN);      // ispisivanje broja u binarnom obliku
    lcd.setCursor(0, 1);    // postavljanje kursora na lokaciju 0,1
                             // (stupac, redak)
    lcd.print(zz, HEX);     // ispisivanje broja u heksadekadskom obliku
}

void loop() {
}
```

Zadatak 6. Na LCD zaslonu ispiši „Mi ♥ Arduino“. Riječ „Arduino“ ispiši u drugom retku. Simbol srca treba kreirati kao posebni znak budući da ne postoji u tablici znakova LCD zaslona. Kodove za kreiranje simbola srca moguće je provjeriti na <http://maxpomer.github.io/LCD-Character-Creator/> ili sličnoj.

Kòd zadatka

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// kreiranje simbola srce (♥)
uint8_t heart[8] = { 0x0, 0xa, 0x1f, 0x1f, 0xe, 0x4, 0x0 };

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
    lcd.begin();
    lcd.backlight();
    lcd.setCursor(0, 0);

    lcd.createChar(1, heart); // pozivanje simbola srce (♥)
    lcd.home();

    lcd.print("Mi");
    lcd.setCursor(3, 0);
    lcd.write(1);             // ispisivanje simbola srce (♥)
    lcd.setCursor(0, 1);
    lcd.print("Arduino");
}

void loop() {
}
```

Pixels						
0	0	0	0	0	0	= 0x00
0	1	0	1	0	0	= 0x0a
1	1	1	1	1	1	= 0x1f
1	1	1	1	1	1	= 0x1f
0	1	1	1	0	0	= 0x0e
0	0	1	0	0	0	= 0x04
0	0	0	0	0	0	= 0x00
0	0	0	0	0	0	= 0x00

Output

```
byte customChar[8] = {
    0b00000,
    0b01010,
    0b11111,
    0b11111,
    0b01110,
    0b00100,
    0b00000,
    0b00000
}
```

Zadatak 7. Na LCD zaslonu, u prvom retku, ispiši „šareno lišće“.

Kod zadatka

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

uint8_t slovo1[8] = { 0xa,0x4,0xe,0x10,0xe,0x1,0x1e };// kreiranje simbola š
uint8_t slovo2[8] = { 0x2,0x4,0xe,0x10,0x10,0x11,0xe };// kreiranje simbola ć

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
    lcd.begin();
    lcd.backlight();
    lcd.setCursor(0, 0);

    lcd.createChar(1, slovo1);      // kreiranje simbola š
    lcd.createChar(2, slovo2);      // kreiranje simbola ć
    lcd.home();

    lcd.write(1);                   // ispis simbola š
    lcd.print("areno li");
    lcd.write(1);                   // ispis simbola š
    lcd.write(2);                   // ispis simbola ć
    lcd.print("e");
}

void loop() {
}
```

Zadatak 8. Proširi spoj dodavanjem potencijometra na bilo koji pin osim A0, A4 i A5. Napiši program koji će u prvom retku, u lijevom kutu ispisivati očitane A/D vrijednost kao na slici ispod. Naputak1: funkciju za ispis znakova na LCD: `lcd.print()` i funkciju za pozicioniranje ispisa `lcd.setCursor()` treba iz funkcije `setup()` premjestiti u funkciju `loop()`. Naputak2: u funkciji `lcd.print(AD_value)` treba kao argument staviti varijablu `AD_value` koja sadrži A/D vrijednost.

Zadatak 9. Proširi program tako da izračunatu vrijednost napona ispisuješ u gornjem redu u desnom kutu prema slici ispod. A/D vrijednost 0 odgovara 0V, a 1023 odgovara naponu 5V. Za računanje vrijednosti napona koristiti funkciju `map()`.

Zadatak 10. Proširi program tako da stanje baterije vizualno prikazuješ pomoću simbola napunjenosti baterije. U tu svrhu potrebno je kreirati 7 simbola za bateriju. U donjem retku ispisati tekst BATTERY i na temelju A/D vrijednosti iz potencijometra, tj. ulaznog napona, simulirati stanje napunjenosti baterije – točno prema slici (punu bateriju prikazat sa svih sedam simbola). Koristiti funkciju `map()`, IF, ELSE IF, ELSE ili SWITCH – CASE. **NAPOMENA:** čitanje i ispis na LCD se treba odvijati u funkciji `loop()`.



Pitanja za provjeru znanja:

1. Nabroji nekoliko jednostavnijih vizualnih indikatora.
2. Nabroj vrste LCD zaslona.
3. Objasni razliku između alfanumeričkog i grafičkog LCD zaslona.
4. Na koji način se stvara slika jednog znaka na alfanumeričkom zaslonu i koja je rezolucija znaka na LCD-u koji koristimo na vježbama?
5. Navedi različite varijante komunikacije s LCD zaslonima.
6. Navedi koje pinove ima alfanumerički LCD zaslon i čemu služi pojedini pin.
7. Što se pohranjuje i instrukcijski registar, a što u podatkovni registar?
8. Koje korake treba napraviti da bi se na zaslonu ispisalo jedno slovo/znak?
9. Koja je uloga biblioteke LiquidCrystal.h?
10. Koje pinove koristimo kod 4-bitne paralelne komunikacije s LCD-om? Koju naredbu koristimo za inicijalizaciju, te što predstavlja pojedini argument?
11. Navedi tri vrste serijske komunikacije koje podržava mikroupravljač u Arduino UNO pločici u komunikaciji s raznim vrstama LCD-a.
12. Skiciraj i objasni I2C komunikaciju te navedi osnovna svojstva.
13. Po čemu se razlikuju uređaji spojeni na I2C sabirnicu, budući da su svi spojeni paralelno na ista dva vodiča?
14. Koje sve vodove (signale) spajamo na I2C LCD, odnosno na I2C adapter (*Backpack*)?
15. Koji pinovi na Arduino Uno i na mikroupravljaču ATmega 328p omogućuju I2C komunikaciju?
16. Iz kojeg razloga smo na alfanumerički LCD zaslon dodali PCF8574 adapter (*Backpack*)?
17. Objasni što se sve može podešavati na I2C adapteru?
18. Na što treba obratiti pažnju kad se LCD zaslon s jednom mikroupravljača premjesti na drugi?
19. Kako se može podešavati adresa LCD-a (I2C adaptera)?
20. Čemu služi biblioteka Wire.h, a čemu LiquidCrystal_I2C.h?
21. Navedi osnovne naredbe za inicijalizaciju LCD-a pomoću LiquidCrystal_I2C.h, te kako bi ispisao jednostavan tekst na zaslonu LCD-a?
22. Što radi funkcija lcd.begin(), a što lcd.setCursor()?
23. Na koji način bi ispisao neki broj u binarnom ili heksadecimalnom obliku na LCD zaslon?
24. Objasni postupak kreiranja znakova kojih nema u tablici znakova LCD zaslona?
25. U kojem modu s obzirom na broj podatkovnih linija mora raditi LCD kad se koristi I2C adapter (*Backpack*)?

LITERATURA:

1. Nathan.chantrell.net, Building a graphical display for OpenEnergyMonitor, <https://nathan.chantrell.net/20111015/building-a-graphical-display-for-openenergymonitor/>, (pregledano 21.10.2018.)
2. Nextion, <https://nextion.itead.cc/>, (pregledano 21.10.2018.)
3. ElectronicWings, AVR ATmega Controllers, LCD16x2 custom character display using AVR ATmega16/ATmega32, <http://www.electronicwings.com/avr-atmega/lcd-custom-character-display-using-atmega-16-32->, (pregledano 21.10.2018.)
4. Mikroelektronika, <https://www.mikroe.com/ebooks/pic-microcontrollers-programming-in-basic/additional-components> , (pregledano 21.10.2018.)
5. Complete LCD interfacing with C-code, <http://lcdinterface.blogspot.com/>, (pregledano 21.10.2018.)
6. Innovation off Engineers, LCD 16x2 (LM016L), <https://embeddedcenter.wordpress.com/ece-study-centre/display-module/lcd-16x2-lm016l/>, (pregledano 21.10.2018.)
7. Crystalfontz America, Inc., CFAH1602X-YYH-JP Character LCD Module Data Sheet, August 2008, www.crystalfontz.com
8. MicrocontrollerBoard.com, LCD - Liquid Crystal Display tutorial, <http://www.microcontrollerboard.com/lcd.html> , (pregledano 21.10.2018.)
9. Arduino.cc, LiquidCrystal Library, <https://www.arduino.cc/en/Reference/LiquidCrystal>, (pregledano 21.10.2018.)
10. Howtomechatronics, Dejan Nedelkovski, How I2C Communication Works and How To Use It with Arduino, <https://howtomechatronics.com/tutorials/arduino/how-i2c-communication-works-and-how-to-use-it-with-arduino/>, (pregledano 21.10.2018.)
11. Youtube: Serial LCD - I2C Backpack, Saravanan AL, https://www.youtube.com/watch?v=Wj_BBSsfNRg, (pregledano 21.10.2018.)
12. Instructables, Arduino I2C LCD Driver Library and PackMan, <https://www.instructables.com/id/Arduino-I2C-LCD-Driver-Library-and-PackMan/>, (pregledano 21.10.2018.)