

Nastavni predmet:	Ugradbeni računalni sustavi
Vježba: 07	Inkrementalni enkoder
Cilj vježbe:	Upravljanje vanjskim elementima spojenih na Arduino pomoću inkrementalnog enkodera

Upute

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

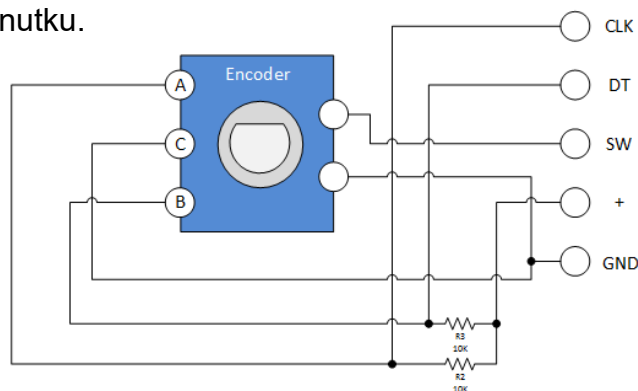
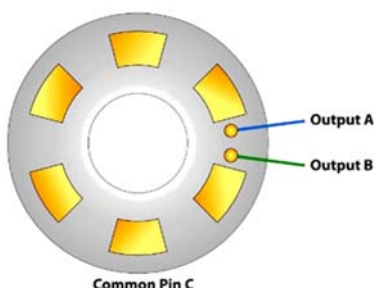
- postupak izrade programa
- objašnjenje korištenih naredbi
- dobivene rezultate po točkama
- odgovoriti u bilježnicu na postavljena pitanja vezana uz ovu vježbu
- Ukoliko u kòdu postoji greška, korigiraj i objasni!

Inkrementalni okretni davači (enkoderi) su senzori koji mogu detektirati kutnu poziciju i brzinu pretvarajući rotacijski mehanički pomak u električne impulse. Moguće je detektirati i **smjer okretanja** uspoređujući dva impulsa koji su fazno pomaknuti za 90°.

Za razliku od potencijometra, imaju **neograničen broj okretaja u oba smjera**. Često se isporučuju s osovnom kao tipkalom koje se može koristiti za potvrdu odabira izbornika, kao reset tipka ili kao bilo koja tipka prikladna u danom trenutku.

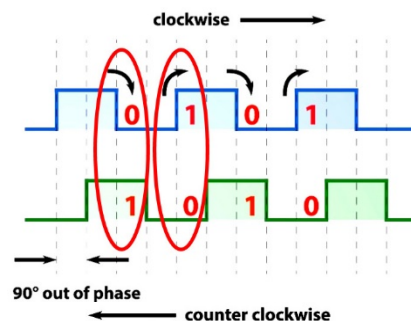
Kad se disk okreće pinovi A i B rade kontakt sa zajedničkim pinom, kao dva tipkala ili dvije sklopke.

U skladu s tim na izlazu A i B će se pojavljivati pravokutni impulsi.

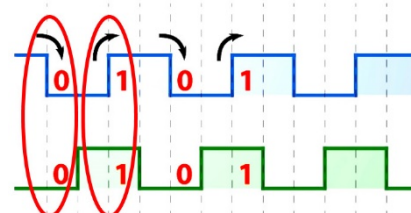


Električna shema KY-040 enkodera

Ako se osovina okreće u smjeru kazaljke na satu (**CW**), impuls na izlazu A će prethoditi izlazu B (za 90° ili T/4). Padajući ili rastući brid impulsa A uvijek prelazi u suprotno stanje nego što je impuls B ($A = \neg B$)



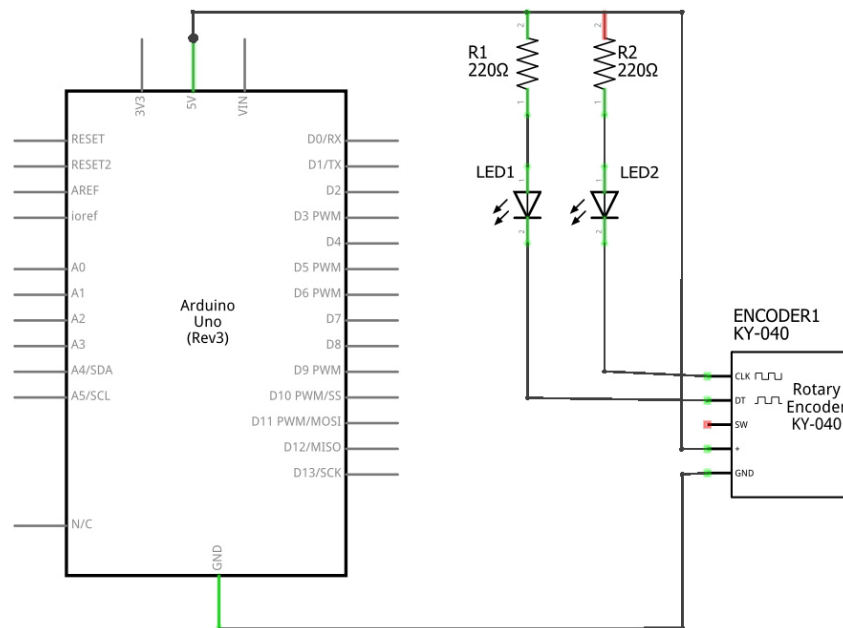
Ako se osovina okreće suprotno od smjera kazaljke na satu (**CCW**), impuls na izlazu B će prethoditi izlazu A (za 90° ili T/4). Padajući ili rastući brid impulsa A uvijek prelazi u isto stanje kao što je i impuls B ($A = B$)



Fotografije: www.howtomechatronics.com

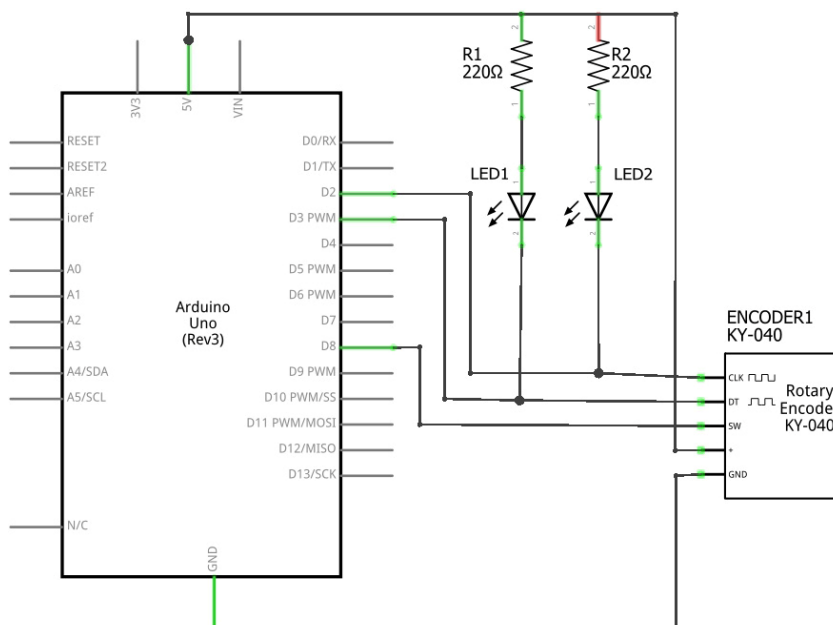
Zadatak 1. Spoji inkrementalni enkoder prema priloženoj shemi i promatraj redoslijed paljenja LED dioda ovisno o smjeru vrtnje. Osovinu okretati sporo i pratiti paljenje LED dioda između dva utora. U ovom zadatku nema koda.

Električna shema



Zadatak 2. Proširi spoj iz prethodnog zadatka povezivanjem inkrementalnog enkodera s mikroupravljačem. Stanje pinova CLK, DT prikazati na Serial monitoru.

Električna shema



Kod zadatka

```
const int PinCLK = 2;           // Izlaz A enkodera - CLK signal
const int PinDT = 3;           // Izlaz B enkodera - DT signal
const int PinSW = 8;           // Tipkalo enkodera
```

```
int counter = 0;
int iCurrentState;
int iLastState;
```

```
String sPrethodni;
```

```
void setup() {
    pinMode(PinCLK, INPUT);
    pinMode(PinDT, INPUT);

    Serial.begin(9600);
    Serial.println("Ready...");
    // očitava inicijalno stanje izlaza A enkodera
    // izvan loop funkcije
    iLastState = digitalRead(PinCLK);
}

void loop() {
    iCurrentState = digitalRead(PinCLK); // očitava trenutno stanje izlaza
                                         // A enkodera
    // ako se prethodno stanje i trenutno stanje izlaza A razlikuje, znači da
    // imamo impuls
    if (iCurrentState != iLastState) {

        // ako je izlaz B enkodera različit od izlaza A,
        // znači da je smjer okretanja CW
        if (digitalRead(PinDT) != iCurrentState) {
            counter++;
            sPrethodni = "++";
        }
        else {
            counter--;
            sPrethodni = "--";
        }
        Serial.print("\nPinCLK = ");
        Serial.print(digitalRead(PinCLK));
        Serial.print("\tPinDT = ");
        Serial.print(digitalRead(PinDT));
        Serial.print("\tPosition: ");
        Serial.print(counter);
        Serial.print("\tCounter");
        Serial.print(sPrethodni);
    }

    iLastState = iCurrentState; // sadašnje stanje varijable iCurrentState
                                // dodjeljujemo varijabli intLastState i
                                // time smo spremni za novo očitavanje
}
```

Zadatak 3. Program iz prethodnog zadatka modificiraj tako da koristiš prekid INT0 koji se nalazi kao sekundarna funkcija na pinu 2. Ispisivati na Serial monitor. Odgovara li broj impulsa broju koraka enkodera? Objasni!

Cilj je uočiti da ako nema debouncinga, broji više impulsa nego što to stvarno jest!

1. Koja je razlika ako se pri najavi prekida umjesto RISING koristi CHANGE?

Kòd zadatka

```
const int PinCLK = 2;      // Koristimo za generiranje prekida
                           // preko CLK signala (pin A enkodera)
const int PinDT = 3;      // Koristimo za očitavanje DT signala (pin B
                           // enkodera)
const int PinSW = 8;      // Koristi se za tipkalo

volatile long virtualPosition = 0;  // obavezno deklarirati kao
                                   // volatile kad se koristi u isr

void isr0() {
    detachInterrupt(0);
    if (!digitalRead(PinDT) == digitalRead(PinCLK))
        virtualPosition++;
    else
        virtualPosition--;
    attachInterrupt(digitalPinToInterrupt(PinCLK), isr0, RISING);
} // ISR0

void setup() {
    pinMode(PinCLK, INPUT);
    pinMode(PinDT, INPUT);
    pinMode(PinSW, INPUT_PULLUP);
    attachInterrupt(digitalPinToInterrupt(PinCLK), isr0, RISING);
    // Deklariramo prekidnu rutinu ISR isr0 koja se
    // izvršava pojavom rastućeg brida na INT 0
    // Testirati ponašanje sklopa ako se umjesto RISING koristi CHANGE
    // Na Arduino UNO interrupt 0 je uvijek
    // spojen na pin 2
    Serial.begin(9600);
    Serial.println("Start");
}

void loop() {
    int lastCount = 0;
    while (true) {
        if (!(digitalRead(PinSW))) {    // provjera da li je stisnuto
                                         // tipkalo
            virtualPosition = 0;        // ako da, resetiraj
                                         // brojač na 0
            while (!digitalRead(PinSW)) { // čekaj dok tipkalo
                                           // nije otpušteno
            }

            Serial.println("Reset");
        }

        if (virtualPosition != lastCount) {
            lastCount = virtualPosition;
            Serial.print("Count:");
            Serial.println(virtualPosition);
        }
    } // while
}
```

Zadatak 4. Modificiraj program iz prethodnog zadatka tako da inkrementalni enkoder broji impulse u skladu sa stvarnim zakretom osovine enkodera. Funkcija za provjeru tipke na enkoderu modificirana je da radi softverski debounce. Pratiti sada na Serial monitoru.

```
const int PinCLK = 2;      // Koristimo za generiranje prekida
                           // preko CLK signala (pin A enkodera)
const int PinDT = 3;      // Koristimo za očitavanje DT signala (pin B
                           // enkodera)
const int PIN_SW_ENC = 8;  // Koristi se za tipkalo

// varijable dodane za softverski DEBOUNCE
int nSW_ENC_STATE = 0;     // trenutno stanje tipkala
byte oldSwitchState_SW_ENC = HIGH; // prethodno stanje tipkala
const unsigned long debounceTime_SW_ENC = 10; // ms
unsigned long switchPressTime_SW_ENC; // kad se stanje tipkala
                                     // zadnji put promijenilo

volatile long virtualPosition = 0; // obavezno deklarirati kao
                                     // volatile kad se koristi u isr

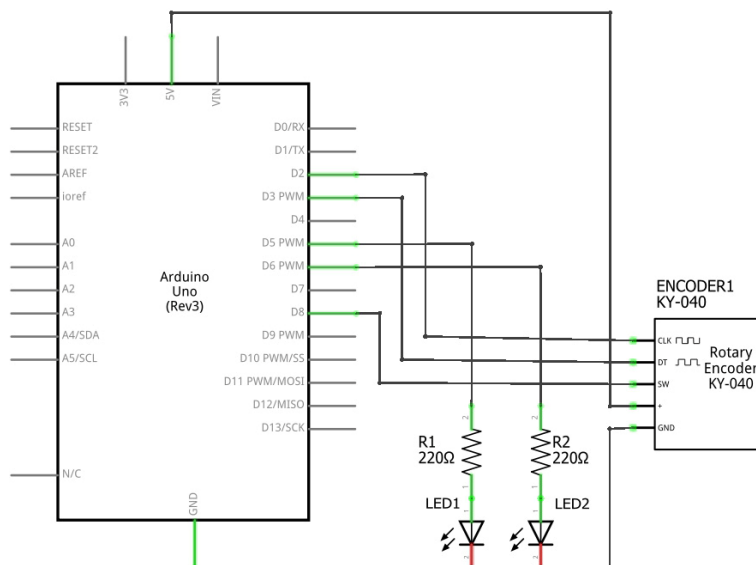
void isr0() {
    detachInterrupt(digitalPinToInterrupt(PinCLK));
    if (!digitalRead(PinDT) == digitalRead(PinCLK)) {
        virtualPosition++;
        // umjesto IF-a, za limitiranje (podešavanje) gornje vrijednosti
        (virtualPosition < 255) ? virtualPosition++ : virtualPosition = 0;
    }
    else {
        virtualPosition--;
        // umjesto IF-a, za limitiranje (podešavanje) donje vrijednosti
        (virtualPosition > 0) ? virtualPosition-- : virtualPosition = 255;
    }

    attachInterrupt(digitalPinToInterrupt(PinCLK), isr0, CHANGE);
} // završetak isr0

void setup() {
    pinMode(PinCLK, INPUT);
    pinMode(PinDT, INPUT);
    pinMode(PinSW, INPUT_PULLUP);
    attachInterrupt(digitalPinToInterrupt(PinCLK), isr0, CHANGE);
    // Deklariramo prekidnu rutinu ISR isr0 koja se okida
    // pojavom rastućeg ili padajućeg brida na pinu PinCLK = 2 (INT 0)
    // Na Arduino UNO interrupt 0 je uvijek
    // spojen na pin 2
    Serial.begin(9600);
    Serial.println("Start");
}
```

```
void loop() {  
  
    // kod koji provjerava da li je pritisnuta tipka  
    // dodan dio koda za eliminaciju Switch Bouncing-a  
  
    int lastCount = 0;  
    nSW_ENC_STATE = digitalRead(PIN_SW_ENC);  
  
    if (nSW_ENC_STATE != oldSwitchState_SW_ENC) {  
  
        //provjera je li vrijeme od pritiska tipke jednako 10ms  
        if (millis() - switchPresTime_SW_ENC >= debounceTime_SW_ENC)  
        {  
            switchPresTime_SW_ENC = millis();// kad pritisnemo prekidač  
            oldSwitchState_SW_ENC = nSW_ENC_STATE;// zapamti za sljedeći  
put  
            if (nSW_ENC_STATE == LOW) {        // provjera da li je stisnuto  
  
                virtualPosition = 0;           // ako da, resetiraj  
                                                // brojač na 0  
                while (!digitalRead(PinSW)) {   // čekaj dok tipkalo  
                }                               // nije otpušteno  
  
                Serial.println("Reset");  
                tone(PIN_BUZZER, 1000, 1000);  
  
                }// ako je stanje na nožici PIN_SW_ENC LOW  
                else  
                {  
                    //Serial.println("prekidač nije stisnut.");  
                } // ako je stanje na nožici PIN_SW_ENC HIGH  
  
            } // ako je debounce vrijeme isteklo završi  
  
        } // kraj provjere tipkala  
  
        if (virtualPosition != lastCount) {  
            lastCount = virtualPosition;  
            Serial.print("Count:");  
            Serial.println(virtualPosition);  
        }  
    } // while  
}
```

Zadatak 5. Modificiraj spoj i program tako da pomoću inkrementalnog enkodera mijenjaš svjetlinu dviju dioda. Svjetlina dioda treba biti suprotna. Kad jedna svijetli najjače, druga svijetli najslabije. Svjetlinu podešavati pomoću PWM. Provjeri koji pinovi omogućuju PWM. Vrijednost svjetline dioda ispisuj na LCD zaslon.



Zadatak 6. Modificiraj program tako da u EEPROM upišeš zadnju vrijednost svjetline diode prije isključenja mikroupravljača. Prilikom uključenja Arduina program treba učitati pohranjenu vrijednost svjetline i omogućiti daljnju promjenu svjetline pomoću rotary enkodera. Koristiti EEPROM.h biblioteku.