HRVATSKA ZAJEDNICA TEHNIČKE KULTURE

Croatian Society for Educational Technology

ARDUINO KROZ JEDNOSTAVNE PRIMJERE

- pripreme za natjecanja -

PRIPREMA 4 - 2015
KORIŠTENJE EEPROM MEMORIJE

Paolo Zenzerović, mag. ing. el.

Zagreb, 2015.

EEPROM MEMORIJA

U ovoj ćemo se pripremi upoznati sa EEPROM memorijom. Njezin naziv dolazi s engleskog od – Electrically Erasable Programable Read Only Memory. To je memorija koja zadržava svoj sadržaj i nakon isključivanja njezinog napajanja. U takvu je memoriju moguće pisati i čitati vrijednosti. Najjednostavnije ju možete zamisliti kao mali tvrdi disk koji se nalazi unutar vašeg mikrokontrolera. U mikrokontroleru koji imamo na Arduino UNO pločici nalazi se EEPROM memorija veličine 1024 byta. Ovu memoriju koristimo za pohranu vrijednosti koje želimo da ostanu zapamćene i kada isključimo mikrokontroler.

Pristup ovoj memoriji vrlo je jednostavan. Kako bismo mogli koristiti ugrađene naredbe za pisanje i čitanje iz memorije iskoristiti ćemo gotovu knjižnicu naredbi. Ova se knjižnica zove eeprom. Da bi ju mogli koristiti na vrhu Arduino programa dodajemo novu naredbu:

```
#include <EEPROM.h>
```

Naredba #include uključuje korištenje neke od knjižnica za Arduino. Nakon same naredbe u <> zagrade upisuje se ime knjižnice – u ovom slučaju to je EEPROM.h.

Knjižnica ima dvije osnovne naredbe – naredbu za pisanje u memoriju i naredbu za čitanje iz memorije.

EEPROM.write(*adresa, vrijednost*) - naredba za pisanje u memoriju

EEPROM.read(*adresa*) - naredba za pisanje u memoriju

Naredba za upis vrijednosti u EEPROM memoriju ima dva argumenta – adresu i vrijednost. *Adresa* definira gdje ćemo u EEPROM memoriju upisati neku vrijednost, a *vrijednost* definira što ćemo na tu adresu upisati.

Naredba za čitanje vrijednosti iz EEPROM memorije ima samo jedan argument – adresu. Ovdje taj argument definira s koje ćemo adrese EEPROM memorije očitati vrijednost.

Prije nego pristupimo radu s EEPROM memorijom moramo razumijeti njezino adresiranje. Kod ATMEGA328 procesora koji se nalazi na Arduino UNO pločici ova memrija ima 1024 byta. To znači da možemo na svakoj memorijskoj adresi od 0 do 1023 upisati jedan bajt podataka. Jednostavnije – argument *adresa* se kreće od 0 do 1023, a argument podatak od 0 do 255.

Pogledajmo prvi primjer.

Zadatak 1: U EEPROM memoriju mikrokontrolera na adresu 0 upišite vrijednost 100 te nakokon toga isčitajte vrijednost sa te adrese i pošaljite ju serijskim putem na računalo. Pomoću Serial monitor alata pogledajte dobivenu vrijednost.

```
#include <EEPROM.h>

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  EEPROM.write(0,100);
  Serial.println(EEPROM.read(0));
}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Uploadajte ovaj program u Arduino i pokranite Serial monitor alat. Na ekranu bi se trebao ispisati broj 100.

Što se ovdje dogodilo?

Na početku smo inicijalizirali serijsku komunikaciju Serial.begin naredbom. Potom smo pomoću naredbe EEPROM.write upisali na adresu 0 vrijednost 100. Pomoću Serial.println naredbe poslali smo serijskim putem rezultat naredbe EEPROM.read(0) koja je pričitala vrijednost koja je upisana u EEPROM memoriji na adresi 0.

Zadatak 2: U EEPROM memoriju na adrese 0 do 99 upišite vrijednosti koje su dva puta veće od adrese na koje ste ih upisali (na adresu 0 upišite 0, na adresu 1 upišite 2, na adresu 2 upišite 4, na adresu 3 upišite 6 itd). Pomoću serial monitor alata ispišite adrese i pripadajuće vrijednosti na ekranu računala za sve adrese u kojima ste upisali brojeve.

```
#include <EEPROM.h>

int brojac;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);

  for(brojac=0; brojac<100; brojac++){
  EEPROM.write(brojac, 2*brojac);
  }

  for(brojac=0; brojac<100; brojac++){
  Serial.print("Adresa:\t");
  Serial.print(brojac);
  Serial.print("\t");
  Serial.print("Vrijednost:\t");
  Serial.println(EEPROM.read(brojac));
  }

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Za ovaj zadatak iskoristili smo for petlju za prolazak kroz adrese i upis u EEPROM memoriju. Prva for petlja broji od 0 do 99. Naredba EEPROM.write(brojac, 2*brojac) će na adresu na kojoj se trenutno nalazi brojač upisati dvostruku vrijednost brojača.

Druga for petlja služi za isčitavanje vrijednosti i njihov ispis pomoću Serial monitor alata. Na početku ispisujemo string “Adresa:” i znak tabulatora nakon toga. Potom ispisujemo trebutnu adresu i nakon nje ponovno znak za tabulator. Sljedeće ispisujemo string “Vrijednost:” pa još jedan tabulator i na kraju vrijednost isčitanu iz EEPROM memorije pomoću EEPROM.read naredbe. Kao zadnju naredbu iskoristili smo Serial.println kako bi sljedeći ispis bio u novom redu.

Ove nam funkcije daju jednostavna rješenja dok god su vrijednosti koje želimo upisati u pojedinu adresu moemorije manje od 255. No što ako želimo trajno memorirati više od jednoj bajta? Recimo da želimo pohraniti očitanu vrijednost analognog ulaza koja varira od 0 do 1023, što onda?

Obično u Arduino koristimo integer varijable u ovim pripremama. To znači da one zauzimaju dva bajta odnosno 16 bitova memorije. Takve varijable možemo onda pohraniti u dva bajta EEPROM memorije. Kako ćemo to učiniti?

Odgovor leži u razdvajanju integer varijable na dvije bajt varijable. Pogledajmo primjer:

```
#include <EEPROM.h>

int podatak = 1000;

int podatak1, podatak2;

int podatakeeprom;

int podatakeeprom1, podatakeeprom2;

int BitShiftCombine( unsigned char x_high, unsigned char x_low)
{
  int combined;
  combined = x_high;
  combined = combined<<8;
  combined |= x_low;
  return combined;
}

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);

  podatak1=highByte(podatak);
  podatak2=lowByte(podatak);

  EEPROM.write(0,podatak1);
  EEPROM.write(1, podatak2);

  Serial.print("Pohranjeni podatak: ");
  Serial.println(podatak);

  podatakeeprom1=EEPROM.read(0);
  podatakeeprom2=EEPROM.read(1);


  podatakeeprom=BitShiftCombine(podatakeeprom1, podatakeeprom2);

  Serial.print("Dobiveni podatak: ");
  Serial.println(podatakeeprom);
}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Unutar koda definirali smo nekoliko varijabli. Varijabla podatak nam je početni podatak koji želimo upisati u EEPROM memoriju. Varijable podatak1 i podatak2 služe za rasčlaniti varijablu podatak na dva dijela. Varijable podatakeeprom1 i podatakeeprom2 služe za isčitavanje podataka pohranjenih u EEPROM memoriju a podatakeeprom varijabla služi za kranje kombiniranje dvaju isčitanih varijabli iz EEPROMa i njihovo kombiniranje u jednu varijablu.

Funkcija BitShiftCombine služi za spajanje dvaju bajtova u jedan integer.

Naredbe highByte i lowByte služe za rasčlanjivanje jednog integera na dva bajta. Nakon rasčalanjivanja integera njegovu vrijednost moramo pohraniti u dva bajta u EEPROM memoriju. U ovome programu to smo učinili na adresama 0 i 1.

Na kraju programa ispisujemo serijskim putem podata koji smo isčitali iz dva bajta EEPROMa i spojili ga u jedan integer.

Ovo možete iskoristiti kod pohrane bilo koji integer vrijednosti u EEPROM memoriju. Pritom morate paziti da ne upišete vrijednosti na adrese koji su vam već zauzete u EEPROMu. Ne zaboravite, jedan integer zauzima dva bajta :)