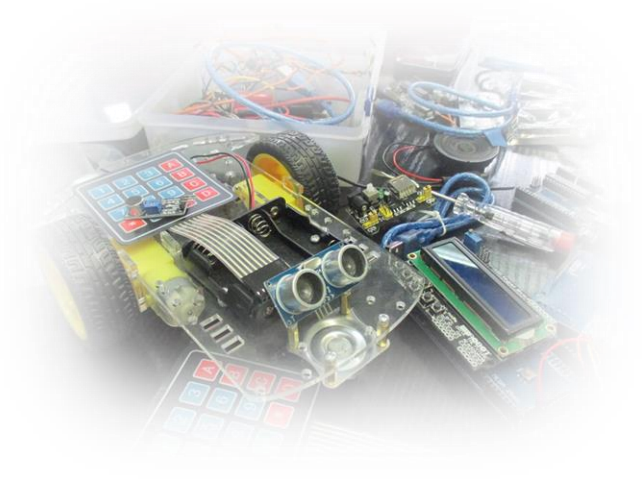
Kreativno programiranje – primena Arduino platforme

Republički seminar Društva matematičara Srbije
Beograd, februar 2018.

Svetlana Radlovački,
Školski centar "Nikola Tesla", Vršac
e-mail: radlovacki@gmail.com



UVOD

Arduino je mikrokontrolerska platforma razvijena u Italiji 2005. godine na Institutu za interaktivni dizajn u Ivreji. Napravljena je od open-source hardverskih komponenti koje su spojene na 8-bitni Atmel AVR mikrokontroler ili 32-bitni ARM procesor. Za sada postoji više od petnaest različitih modela. Trenutno aktuelne verzije ove popularne ploče su Arduino Uno R3, Arduino Leonardo, Duemilanove, Nano, Pro Mini, Arduino MEGA 2560 R3, Arduino Due…

Cena originalnog Arduina trenutno iznosi oko 30€, a broj proizvedenih ploča iz godine u godinu rapidno raste. Pored originalnih verzija, na tržištu se mogu pronaći kineske kopije po znatno povoljnijim cenama (nekoliko stotina dinara), uz istu funkcionalnost i ne mnogo slabiji kvalitet.

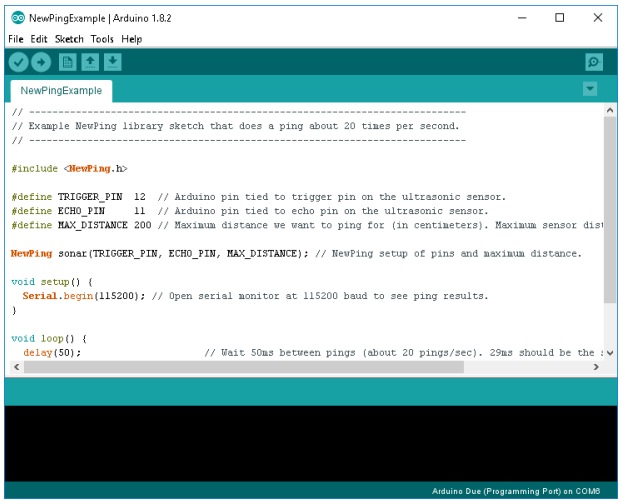
Na ulaze ovog kontrolera moguće je dovesti tastere, prekidače, gotove tastature, različite senzore (temperature, pritiska, protoka, pokreta, ultrazvučni senzor…), dok izlaze možete povezati na širok spektar izvršnih uređaja i komponenata – LE (*light emitting*) dioda, sijalica, zujalica, motora, ekrana za prikaz podataka…

Dodatni, gotovi štitovi (*shields*), zasebne štampane ploče koje se montiraju na osnovnu ploču i proširuju njene osnovne mogućnosti, mogu da obavljaju različite funkcije (npr. kontrola motora, povezivanje senzora, bežična komunikacija itd.) i na taj način učine rad sa ovom platformom još jednostavnijim i lakšim.

Niski naponi napajanja (3.3 i 5V) omogućavaju bezbednu upotrebu Arduina čak i u nižim razredima osnovne škole.

Sve ovo omogućava da učenike na interesantan način uvedemo u svet novih tehnologija. Kroz igru i zanimaciju, kod učenika se budi želja za savlađivanjem i proširivanjem nastavnog gradiva iz oblasti programiranja, ali i oblasti prirodnih i tehničkih nauka.

Arduino se programira preko posebnog editora (*Slika 1*), u koji se unose C/C++ programski kodovi. Odličnu podršku i ideje za dalji rad možete pronaći u velikom broju gotovih, jednostavnih primera koji se dobijaju uz samo okruženje. Na samom početku, mini projekte možete osmišljavati i realizovati tako što ćete vršiti modifikaciju već napisanih kodova. Kada učenici shvate osnovne principe rada, pisanje programa postaje trivijalno.

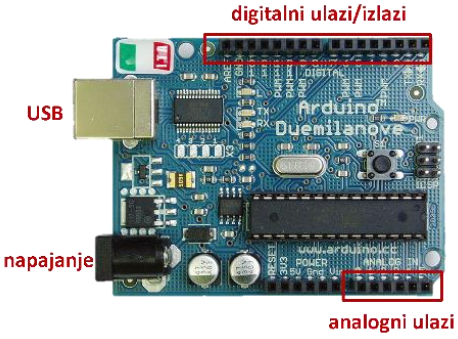


Slika 1

Za rad sa ovom platformom, iako se u prvom momentu tako čini, nisu neophodna velika predznanja iz oblasti elektrotehnike i tehnike uopšte. Velika prednost je što se na internetu mogu pronaći mnogobrojni sajtovi sa detaljnim objašnjenjima vezanim za načine vezivanja komponenata koje se koriste, kao i gotovim bibliotekama koje olakšavaju rad sa istim.

Po detaljnom osmišljavanju projekta, odnosno izgleda uređaja i njegove funkcije, možete izabrati i povezati potrebne komponente prema uputstvima i pristupiti pisanju programa.

Pisanjem programa i posmatranjem njihovih konkretnih efekata, učenici će neprimetno savladati primenu grananja, ciklusa, nizova u programima. Uz pomoć određenih komponenata (LED matrica, Matrix tastatura 4x4,...) lakše će razumeti primenu dvodimenzionalnih nizova, pojmove indeksa elementa, glavne, sporedne dijagonale. S druge strane, čak iako su na samom početku učenja, polako ćete ih uvesti u oblast objektno orijentisanog programiranja.



Sa ukupno 14 digitalnih ulazno/izlaznih pinova i 6 analognih 10-bitnih ulaza, *Arduino Uno* ili *Arduino Duemilanove* ploča je sasvim dovoljna za početak rada. U *Tabeli 1* prikazana je lista nekoliko praktičnih komponenata koje možete povezati na platformu kako biste nastavu učinili interesantnijom:

Komponenta/štit	Kataloški naziv	Preporučena biblioteka
LE dioda	5mm **Light-emitting Diode**	
Ultrazvučni senzor	**HY-SRF05**	**NewPing** (Tim Eckel)
Senzor za temperaturu i vlagu	**DHT11** Temperature/Humidity Sensor	**DHT** (Mark Ruys)
Matrična tastatura 4x4	**4x4 Matrix** 16 Key Membrane Switch Keypad	**Keypad** – standardna
LED matrica 8x8	**MAX7219** Red Dot **Matrix** Module	**Sprite** (David A. Mellis) **Matrix** (Nicholas Zambetti)
Displej sa tasterima	**LCD 1602** Keypad Shield V2.0	**LiquidCrystal** – standardna
Bluetooth modul	**JY-MCU Bluetooth**	
Foto otpornik	**Photo resistor**	
Servo motor	**TowerPro SG90 9g Mini Servo**	**Servo** - standardna
Infra-crveni prijemnik	**3mm Infrared Receiver Diode**	**IRremote** (Ken Shirriff)

Tabela 1

PRIMERI KORIŠĆENJA PLATFORME

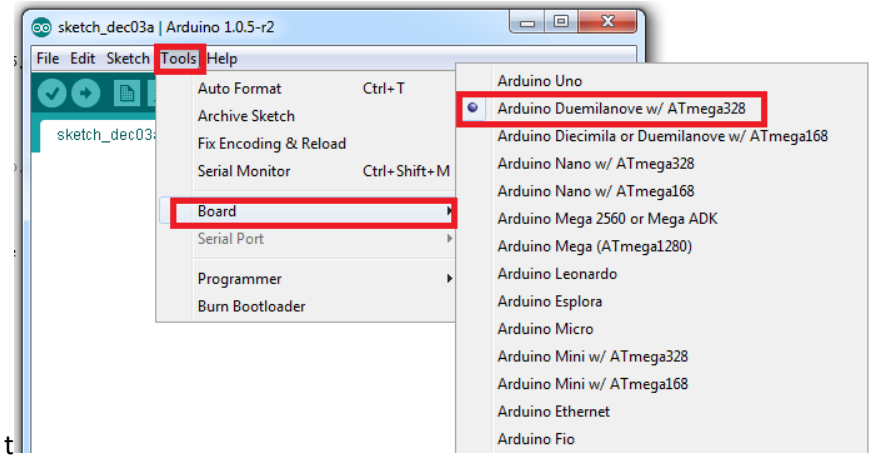
Zavisno od uzrasta, predznanja i interesovanja učenika, programi mogu biti različite težine. Učenici će imati priliku da isti problem reše na više različitih načina, da utvrde razlike, prednosti i mane određenih programskih rešenja, kao i da veoma lako modifikuju postojeća rešenja, predlože nove, slične zadatke i tako kroz vežbu utvrde postojeća znanja i prošire svoja programerska iskustva.

MOJ PRVI PROJEKAT

Jedan od najčešćih i najpraktičnijih primera koji se koristi za prvo testiranje Arduino platforme jeste uključivanje i isključivanje LE diode (sa pauzama u trajanju jedne sekunde) koja je povezana na pin 13. Zašto baš ovaj pin? Arduino platforma na sebi ima ugrađenu svetleću diodicu koja je na njega povezana, tako da za ovo testiranje nije potreban nikakav dodatni hardver.

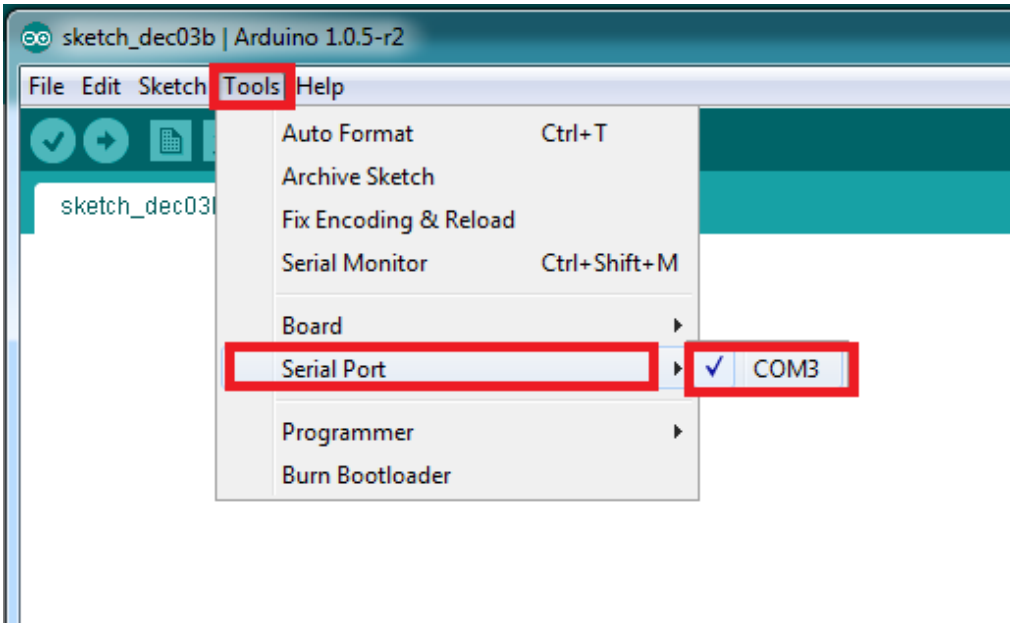
Arduino IDE se može preuzeti sa zvaničnog sajta *www.arduino.cc.* Instalacija je veoma jednostavna. Pre početka pisanja prvog programa neophodno je podesiti dve stvari, kako biste uspešno ostvarili komunikaciju sa kontrolerom i učitali u njega željeni program.

Prvi korak je da izaberete naziv platforme koju koristite (*Duemilanove ATmega 328* na *Slici 2*)



Slika 2

Drugi korak podrazumeva izbor serijskog porta preko kojeg će računar komunicirati sa Arduinom (*Slika 3*)



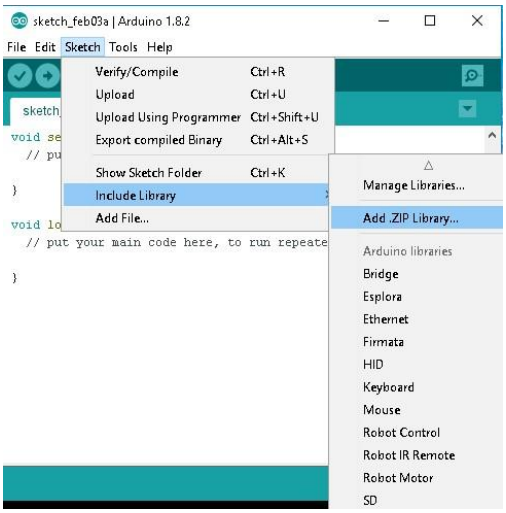
Slika 3

Ako se desi da je ponuđeno više komunikacionih portova i niste sigurni koji da izaberete, sve nedoumice možete razrešiti odlaskom u *Control Panel->Device Manager*. Tu možete proveriti koji se komunikacioni port pojavljuje prilikom priključenja Arduino platforme.

Programi napisani u ovom okruženju nazivaju se skečevi (*sketches*). Svaki skeč mora sadržati dve osnovne funkcije (*Slika 4*). To su:

- *setup( )* - funkcija koja se izvršava jednom na početku i služi za početna podešavanja
- *loop( )* - funkcija koja se izvršava u petlji sve vreme dok se ne isključi ploča

Slika 4

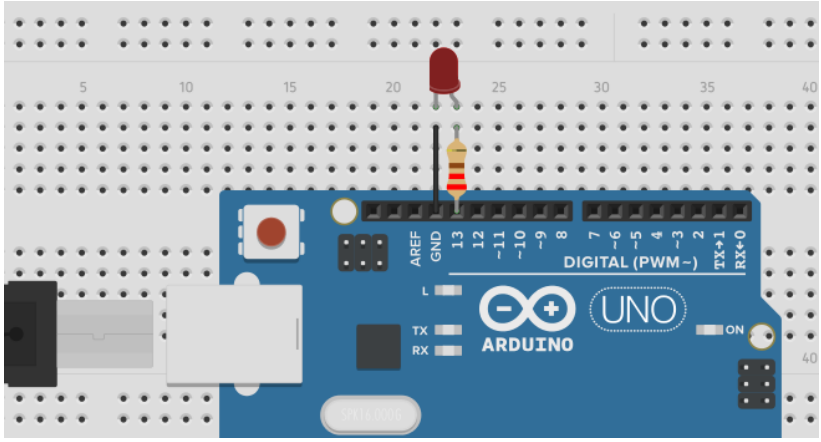


Slika 5

Za korišćenje određenih štitova i komponenti, postoje i adekvatne biblioteke, sa unapred definisanim prototipovima funkcija koje se uz njih koriste. Ove biblioteke pre provere ispravnosti programa treba uključiti u okruženje i pozvati ih na samom početku programa (direktiva *#include*).

Biblioteku možete uključiti direktno iz okruženja (*Sketch-> Include Library-> Add .ZIP Library…*) kao što je prikazano na *Slici 5*. Drugi način da učitate biblioteku jeste da raspakujete arhivu preuzetu sa interneta i kompletan folder iskopirate na mesto gde su i ostale biblioteke instaliranog Arduino okruženja (u većini slučajeva *C:\Program Files (x86)\Arduino\libraries*).

Ukoliko želite, možete (iako postoji ugrađena test diodica) povezati diodu na pin 13 kao na *Slici 6*.



Slika 6

Program je sledeći:

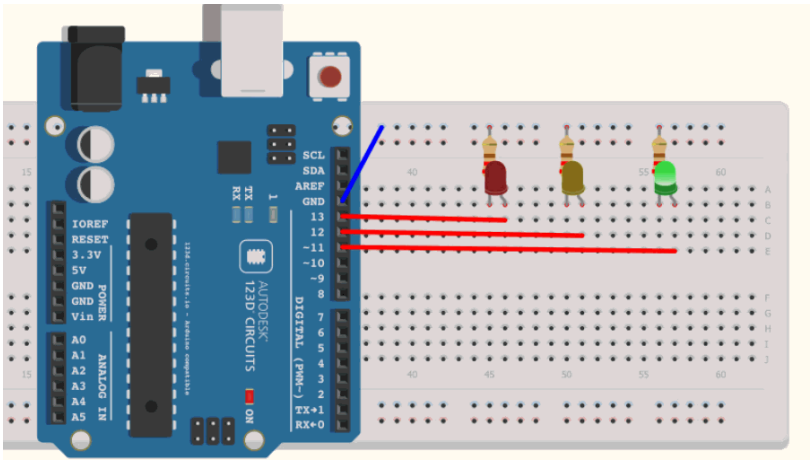
```
int led = 13;  //dioda je povezana na pin 13

void setup() {
  pinMode(led, OUTPUT); // pin 13 je izlaz
}

void loop() {
  digitalWrite(led, HIGH); //dioda uključena
  delay(1000); // pauza 1 sekund
  digitalWrite(led, LOW); // dioda isključena
  delay(1000); // pauza 1 sekund
}
```

Pošto ste napisali program, njegovu ispravnost možete proveriti klikom na dugme *Verify* (u starijim verzijama *Compile*) , a potom ga učitati u mikrokontroler pritiskom na dugme *Upload* .

SEMAFOR – linijska struktura

Kao jedan od prvih primera koje možete uraditi sa učenicima, prosto se nameće izrada malog semafora.

Pored svetleće diode povezane na pin 13 (crvene), na isti način možete povezati i u programu definisati još dve, žutu i zelenu (pinovi 12, 11 respektivno). (*Slika 7*)



Slika 7

Uz objašnjenje da se primenom funkcije *delay(ms)* gde je sa *ms* označeno vreme zadrške u milisekundama, učenici će vrlo brzo sastaviti svoj prvi program.

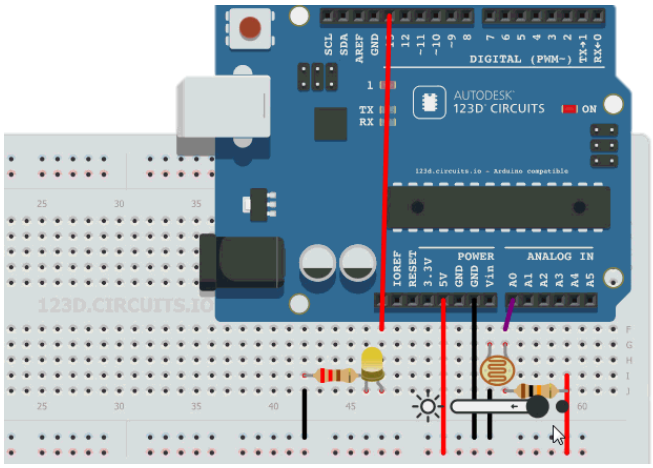
```
void setup() {
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, 1); //crveno
  digitalWrite(12, 0);
  digitalWrite(11, 0);
  delay(3000);          // 3 sekunde
  digitalWrite(12, 1); //crveno i žuto
  delay(1000);          // 1 sekund
  digitalWrite(13, 0);
  digitalWrite(12, 0);
  digitalWrite(11, 1); //zeleno
  delay(3000);
  digitalWrite(12, 1); //žuto
  digitalWrite(11, 0);
  delay(1000);
}
```

Dodavanjem tastera koji bi možda uključio trepćuće žuto ili ubrzao pojavljivanje zelenog svetla (semafori na magistralama), programi mogu postati (blago primetno) složeniji, ali svakako interesantniji.

DAN - NOĆ – grananje u programu

Primenu grananja u programu takođe možemo objasniti učenicima na praktičnom primeru iz svakodnevnog života (možda ih podstaći da razmišljaju i o energetskoj efikasnosti).

Potrebno je samo povezati foto-otpornik (*LDR – light detecting resistor*) na analogni ulaz Arduina (kao na *Slici 8*), a na digitalni izlaz LE diodicu koju treba uključiti kada je u okolini dovoljno mračno, odnosno isključiti kada se uređaj nalazi u osvetljenom prostoru.



Slika 8

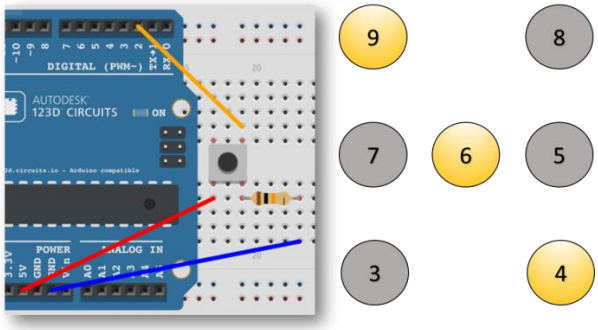
Sa različitim osvetljajem foto-otpornika menja se njegova otpornost, pa samim tim i vrednost koju očitava analogni ulaz Arduino platforme (pošto je analogni ulaz 10-bitni, očitane vrednosti kretaće se u rasponu 0-1023) . Kako biste videli koje su to vrednosti, najbolje je da napišete program koji će te vrednosti ispisivati na serijski port. Izborom *Tools->Serial monitor* možete posmatrati njihovu promenu i odlučiti se koju ćete vrednost postaviti kao granicu za uključivanje/isključivanje diode (u primeru ispod to je 500).

```
int LED = 13;
int LDR = A0;

void setup() {
  pinMode(LED, OUTPUT); // dioda na pinu 13
  pinMode(LDR, INPUT);  // foto-otpornik na pinu A0
 // Serial.begin(9600); // ako želite da pratite stanje
}

void loop() {
  int stanje = analogRead(LDR);
 // Serial.println(stanje);
  if (stanje>500)
   digitalWrite(LED, HIGH);
  else
   digitalWrite(LED, LOW);
  }
```

ELEKTRONSKA KOCKICA ZA „NE LJUTI SE ČOVEČE“

Program generiše slučajan broj 1-6 i zavisno od tog broja diode svetle po rasporedu kao na običnoj kockici. Kombinacije dioda koje svetle brzo se smenjuju, a kada se pritisne taster, trenutna kombinacija se zaustavlja 3 sekunde i to je “dobijeni broj”.

Potrebno je povezati taster na neki od pinova (jedan od načina je prikazan na *Slici 9,* taster je povezan na pin 2) i sedam dioda (preko otpornika vrednosti 220Ω) na pinove 3-9.



Slika 9

I rešenje: jednostavnije rešenje, pogodno za učenike osnovnih škola (primena višestrukog grananja):

```
int taster = 2;

setup() {
 pinMode(taster, INPUT);
 pinMode(3, OUTPUT);
 pinMode(4, OUTPUT);
 pinMode(5, OUTPUT);
 pinMode(6, OUTPUT);
 pinMode(7, OUTPUT);
 pinMode(8, OUTPUT);
 pinMode(9, OUTPUT);
}

void loop() {
 int broj = random(1, 7); //generisanje slučajnog broja 1-6
 bool stop = digitalRead(taster);
 switch(broj)
 {
  case 1: digitalWrite(3, 0); digitalWrite(4, 0);
          digitalWrite(5, 0); digitalWrite(6, 1);
          digitalWrite(7, 0); digitalWrite(8, 0);
          digitalWrite(9, 0); break;
  case 2: digitalWrite(3, 1); digitalWrite(4, 0);
          digitalWrite(5, 0); digitalWrite(6, 0);
          digitalWrite(7, 0); digitalWrite(8, 1);
          digitalWrite(9, 0); break;
. . . . . .//ispisati kombinacije za ostale brojeve
 }
 delay(300);
 if(stop == HIGH)//ako je pritisnut taster pauza 3 sekunde
  delay(3000);
}
```

Definisanje ulaza/izlaza moguće je učiniti na više načina (npr. primenom ciklusa i niza) i tako smanjiti broj programskih linija.

II rešenje: za malo starije – primena dvodimenzionalnih nizova

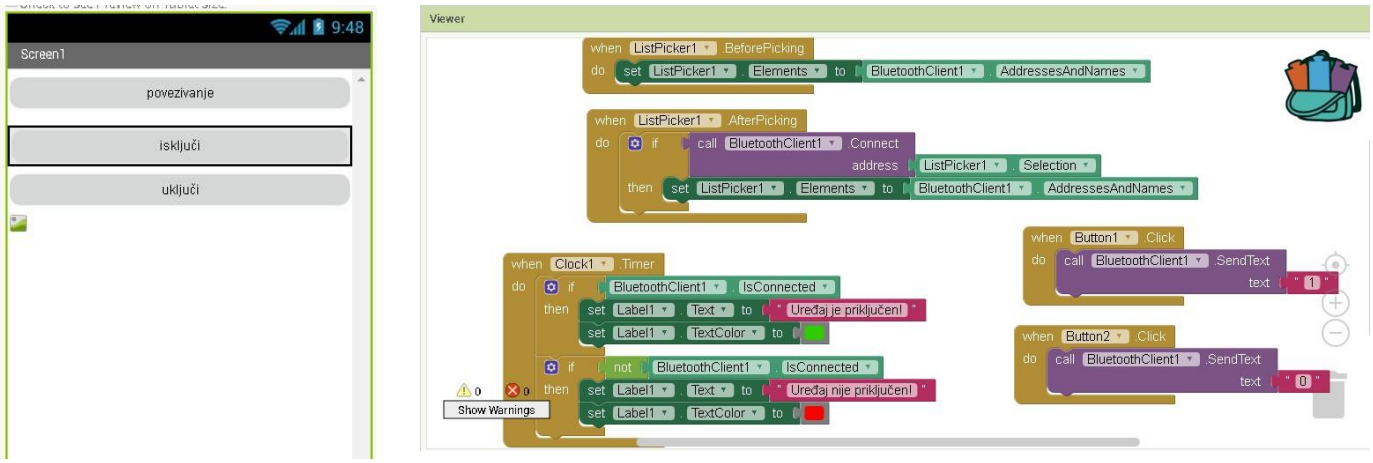
Na samom početku definisali smo kombinacije uključenih/isključenih dioda za određene cifre.

```
int taster = 2;      pinovi 3 4 5 6 7 8 9
byte brojevi[6][7] = { { 0,0,0,1,0,0,0 },  // = 1
                       { 1,0,0,0,0,1,0 },  // = 2
                       { 0,1,0,1,0,0,1 },  // = 3
                       { 1,1,0,0,0,1,1 },  // = 4
                       { 1,1,0,1,0,1,1 },  // = 5
                       { 1,1,1,0,1,1,1 },  // = 6
                     }; //brojevi 1-6 prikazani pomoću sedam dioda

void setup() {
  pinMode(taster,INPUT);
  for(int i=3; i<=9; i++) //diode su povezane na pinove 3-9
   pinMode(i,OUTPUT);
  }

void loop() {
  bool stop = digitalRead(taster);
  int broj = random(1,7);
  for(int pin=3; pin<=9; pin++)
    digitalWrite(pin,brojevi[broj-1][pin]); //[broj-1] zato što indeks u nizu kreće od 0
  delay(200);
  if(stop == HIGH) //ako je pritisnut taster, pauza 3 sekunde
   delay(3000);
}
```

DALJINSKO UPRAVLJANJE

U skladu sa vremenom u kojem živimo, učenici su posebno zainteresovani za daljinsko upravljanje. Neki od primera koje lako mogu savladati jesu promena boje lampe, upravljanje vozilom, upravljanje pokretima napravljenog robota. Za manje domete, uz optičku vidljivost, na jednostavan način platformom možete upravljati i pomoću bilo kojeg daljinskog upravljača.

Za malo veće domete (10m) možete upravljati uređajima preko *Bluetooth*-a. Veoma praktična, besplatna web aplikacija otvorenog kôda, koju možete koristiti za tu svrhu, jeste *MIT App Inventor*. Slaganjem određenih blokova, moguće je, na veoma jednostavan način, sastaviti program koji će omogućiti bežično slanje odgovarajućih podataka (brojeva, karaktera i sl.) na Arduino.



Slika 10

Na *Slici 10* prikazana je blokovska šema programa koji omogućava slanje vrednosti 1 ili 0 sa mobilnog uređaja preko *Bluetooth*-a.

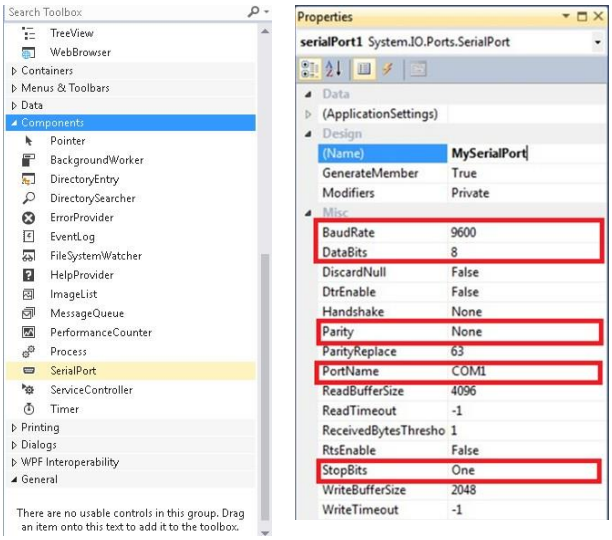
Ako na Arduino platformu postavite npr. *JY-MCU Bluetooth* modul (osim napajanja modula, povezuju se samo dva pina Rx(*receive*) i Tx(*transmit*)), ti podaci mogu biti primljeni i obrađeni. Ovaj tip bežične komunikacije svešće se na klasičnu serijsku komunikaciju. Neophodno je samo pre upravljanja „upariti“ uređaje, za šta se koristi standardan kôd **1234** (isti kôd se koristi i za uparivanje Arduina sa računarom).

Zavisno od vrednosti koje se prosleđuju, moguće je programski definisati akcije koje treba da se izvrše nad komponentama koje su povezane na izlaze platforme. U primeru ispod, ako preko serijskog porta primi karakter ‘1’, kontroler će uključiti LED na pinu 13, ako primi vrednost ‘0’ isključiće diodu.

```
 void setup()
{
  pinMode(13, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  if(Serial.available())
  {
    char c = Serial.read();
    if (c=='1')
     digitalWrite(13,HIGH);
    else if (c=='0')
     digitalWrite(13,LOW);
   }
}
```

Zahvaljujući jednostavnosti rada u *App Inventor*-u, već i učenici viših razreda osnovne škole moći će, bez problema, da naprave svoje prve projekte sa primenom daljinskog upravljanja.

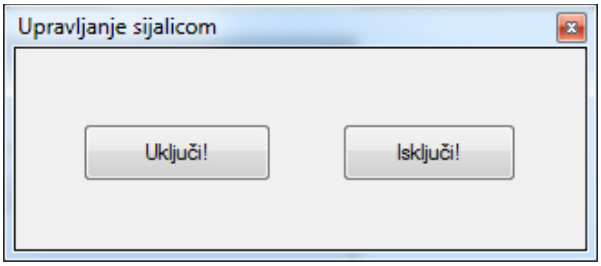
Arduino platformu veoma je lako povezati i sa programima pisanim u *C#* programskom jeziku. *C#* aplikacijom se definiše “grafičko okruženje” i omogućava programsko upravljanje platformom na način sličan malopre opisanom: podaci se preko serijskog porta (komponenta *SerialPort*) šalju na platformu, a program pisan u Arduino okruženju prima te podatke i sprovodi određene akcije u zavisnosti od primljenih vrednosti. Neophodno je samo podesiti parametre porta kao što je prikazano na *Slici 11.* Pored brzine protoka podataka (baudrate 9600b/s je standardan za rad sa platformom Arduino Uno), neophodno je navesti ime porta preko kojeg se komunicira (proveriti na koji je port povezan Arduino).



Slika 11

Evo i jednostavnog primera napisanog u C# programu koji će omogućiti uključivanje/isključivanje svetleće diode:

Na formu smo postavili dva dugmeta. (*Slika 12*)



Slika 12

```
// OTVARANJE PORTA (npr. kada se učitava glavna forma)

private void Form1_Load(object sender, EventArgs e)
  {
    MySerialPort.Open();
  }

// ZATVARANJE PORTA (npr. kada se zatvara aplikacija)

private void Form1_FormClosing(objectsender,FormClosingEventArgs e)
  {
 if (MySerialPort.IsOpen) MySerialPort.Close();
  }

// SLANJE PODATAKA na serijski port (tipa string)
private void btUkljuci_Click(object sender, EventArgs e)
 {
    mySerialPort.Write("1");
 }

private void btIskljuci_Click(object sender, EventArgs e)
 {
    serialPort1.Write("0");
 }
```

Arduino program, koji prima te podatke identičan je programu iz prethodnog primera sa *App Inventorom*.

Pored opisane realizacije upravljanja preko *Bluetooth*-a, koje je veoma pogodno za upravljanje platformom sa određenog uređaja (npr. mobilnog telefona, *PC*-a), bežičnu komunikaciju moguće je uspostaviti i preko *XBee*-a (pogodno za realizaciju mreže različitih senzora), ili preko *WiFi*-ja (uz upotrebu npr. *ESP8266* modula). I u ovoj oblasti (IoT – *Internet of Things*) Arduino može naći primenu u školama, nadovezujući se na poznavanje *HTML*-a i *CSS*-a.

Ideje su nebrojene. Da li ćete postaviti nekoliko dugmića i upravljati robot vozilom, postaviti klizače pa njihove vrednosti (0-255) prosleđivati pinovima RGB diode i upravljati njenom bojom, povezati preko releja pravu sijalicu (230V) pa smanjivati intenzitet svetlosti u prostoriji sedeći za računarom… prepuštam vama. Ono što je dobro jeste da, kao što smo već pomenuli, Arduino ima sve veću primenu u svetu, kako u oblasti učenja programiranja, tako i za izradu praktičnih projekata i uređaja korisnih i primenljivih u svakodnevnom životu. Podrška zajednice je velika i skoro da ne postoji zamisao koju ne možete ostvariti. Najbolje ideje za realizaciju projekata biće one koje samostalno predlože vaši učenici.

Sve navedeno doprinosi osmišljavanju ideja za projekte koji će povećati interesovanje učenika za nastavu programiranja, njegovo povezivanje sa prirodnim naukama i primenu u novim oblastima, koje nas očekuju u neposrednoj budućnosti.

ZAKLJUČAK

Arduino platformu možete koristiti bez ikakve izmene nastavnih programa. Osnovne prednosti njene primene u nastavi su:

- platforma otvorenog kôda
- jednostavna i bezbedna za korišćenje
- povoljna cena
- široka mogućnost primene
- odlična podrška zajednice
- veliki broj gotovih primera i projekata
- omogućava da korisnici osete vizuelne i audio efekte napisanih aplikacija

Efekti primene ove platforme u nastavi programiranja:

- povećava motivaciju i interesovanje učenika za učenje programiranja i primenu informacionih tehnologija
- pospešuje kreativnost učenika
- pospešuje saradnju učenika i razmenu ideja
- razvija istraživački duh
- podstiče učenike na izradu konkretnih projektnih zadataka
- povećava nivo aspiracije učenika

Arduino može poslužiti kao odlično sredstvo za prelazak sa struktuiranog na objektno-orijentisano programiranje, za razumevanje principa daljinskog upravljanja, interneta stvari, ali i povezivanje različitih naučnih činjenica i primenu istih u izradi interdisciplinarnih projekata.

Ako niste u prilici da nabavite Arduino platformu, možete koristiti i odličan, besplatan *Autodesk 123D Circuits* simulator, realizovan u formi web aplikacije zatvorenog kôda na adresi *http://www.123d.circuits*

Još pre više od deset godina, *Kevin Ashton* je osmislio i upotrebio termin "Internet stvari" (*Internet of Things, IoT*), kako bi opisao koncept po kome će u budućnosti svakodnevni fizički objekti, stvari, biti povezane na internet i moći da komuniciraju.

Prema njemu se stvari, po prirodi pasivne, transformišu u inteligentne, "pametne" aktere koji putem interneta mogu razmenjivati informacije o sebi i svom okruženju sa drugim stvarima ili ljudima. Za razliku od personalnih računara i mobilnih telefona, koji predstavljaju nervne završetke današnjeg interneta, nervni završeci interneta biće mali, jeftini kontroleri i procesori ugrađeni u svakodnevne stvari.

"Internet stvari" nije utopijska vizija budućnosti. Prevazilaženje programa čija je svrha samo izračunavanje i ispis određenih podataka na računaru, izlazak iz "monitorskih" okvira i postepeno uvođenje programiranja kontrolera koji upravljaju stvarima u realnom svetu jedan su od načina da učenike pripremimo za primenu budućih tehnologija.

REFERENCE

[1] Jeremy Blum, Exploring Arduino – Tools and Techniques for Engineering Wizardry, Wiley

[2] Brian Evans, Beginning Arduino Programming – Writing Code for the Most Popular Microcontroller Platform in The World, Apress

[3] M.McRoberts, Arduino Starters Kit Manual - A Complete Beginners Guide to the Arduino, Earthshine Design

[4] https://www.arduino.cc/

[5] http://www.instructables.com/howto/arduino/

[6] http://appinventor.mit.edu/explore/

[7] https://cecaradlovacki.wordpress.com/