

# Uvod u Arduino programiranje

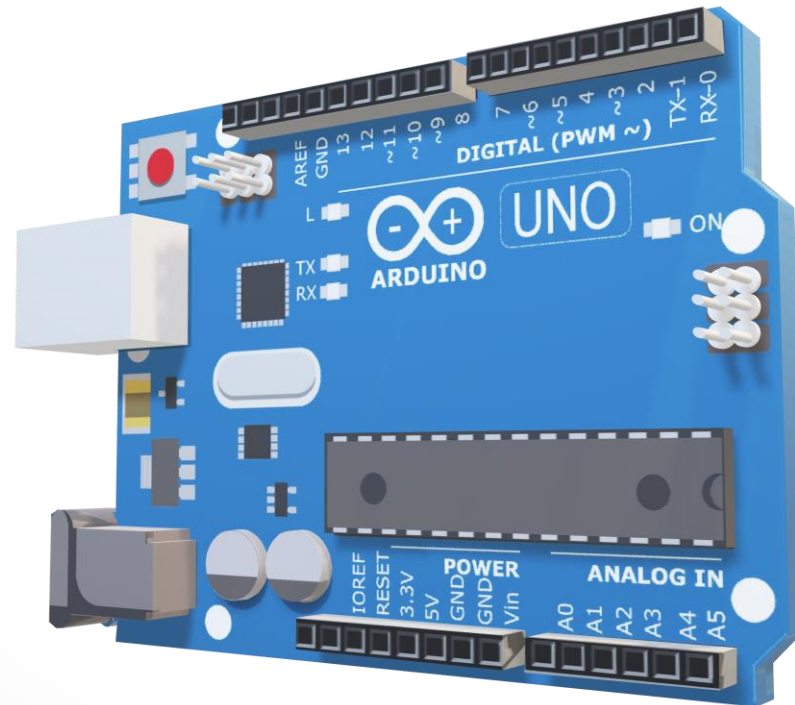
## *Internet stvari 2023. - I termin*



Nenad Petrović

*Univerzitet u Nišu, Elektronski fakultet*

[nenad.petrovic@elfak.ni.ac.rs](mailto:nenad.petrovic@elfak.ni.ac.rs), kancelarija 323



# Materijali

- CS Moodle stranica kursa IoT – prezentacije
  - <https://cs.elfak.ni.ac.rs/nastava/course/view.php?id=149>
- Video materijali
  - <https://www.youtube.com/@penenadpi.science5630>
- GitHub repozitorijum sa primerima
  - [https://github.com/penenadpi/iot\\_course\\_2023](https://github.com/penenadpi/iot_course_2023)

# Uvod

- Arduino je pristupačna razvojna ploča zasnovana na open-source hardveru i softveru
  - Bilo ko može da proizvodi Arduino ploče i softver za njih
- Inicijalni razvoj – Ivrea (Italija) 2003. godine kao master teza, 2005. ozvaničen projekat, a 2008. osnovana kompanija Arduino LLC
  - Inicijalno za ATmega128 mikrokontrolere
  - Od 2005. podrška za jeftinije Atmel 8-bit AVR mikrokontrolere
  - Od 2012. i podrška za 32-bit ARM cortex
  - Veliki broj modela – 2016 godine čak 17 različitih proizvoda
- Primarno namenjen edukaciji i razvoju prototipova
- Arduino Programming Language
  - Zasnovan na C-u
- Interakcija sa spoljašnjim svetom preko ulaza i izlaza (portova)
  - Reaguje na događaje iz spoljašnje sredine - čita vrednosti sa senzora (temperatura, blizina, vlažnost...) i dugmića, a upravlja različitim aktuatorima (svetla, motori...)

# Ključne prednosti Arduino platforme

- Open-source dizajn
- Lako rešavanje problema i odlična podrške zajednice
- Jednostavan USB interfejs
  - Ploča sa čipom se direktno može povezati na računar preko USB porta
  - Vidljiv od strane računara kao virtuelni serijski port
  - Omogućena komunikacija kao bilo koji drugi serijski uređaj
- Udobno rukovanje napajanjem zahvaljujući ugrađenoj automatskoj regulaciji napona
  - Može se povezati spoljašnji izvor napajanja do 12v, a regulacija će biti izvršena na 5v i 3.3v
  - Direktno napajanje uz pomoć USB porta
- Pristupačna cena
  - Oficijelni proizvod od oko 20e pa naviše
  - Sam mikrokontroler ATmega328 se može naći online u prodaji za samo 3e
- Širok skup hardverskih mogućnosti
  - Tajmeri, PWM piov, eksterni i interni prekidi, više modova spavanja
- Laka proširivost
  - Dodaci u vidu „shield“-ova
- Dovoljno brz za većinu primena
  - 16MHz u najrasprostranjenijim varijantama

# Primene

- Ogromna zajednica korisnika
  - Oko 10 milina prodatih Arduino Uno uređaja
- Interesantna rešenja
  - Arduboy
    - Ručna konzola za igre
  - Arduinome
    - MIDI kontroler
  - Ardupilot
    - Platforma za dronove
  - C-STEM Studio
    - Platforma za praktičnu integrisanu nastavu iz računarstva, inženjerstva i matematike sa robotikom
  - Naučni radovi
    - Evidencija raznih podataka – COVID-19 provere, rashladni i termički uređaji
  - OBDuino
    - Pokretni dijagnostički interfejs za moderne automobile
  - OpenEVSE
    - Open-source punjač za elekrična vozila

# Pregled modela i varijanti

| Model               | Opis                                                                                                                              | Ključne karakteristike                                                                                                                                                                                                                                       |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Arduino Uno (R3)    | Najrasprostranjenija varijanta – malih dimenzija i niske cene                                                                     | <ul style="list-style-type: none"><li>- USB konekcija + eksterno napajanje</li><li>- 16MHz 8-bit ATmega328P mikrokontroler</li><li>- 32 KB programske memorije</li><li>- 14 digitalnih ulaza/izlaza (6 od njih za PWM)</li><li>- 6 analognih ulaza</li></ul> |
| Arduino Mega 2560   | Standardnih dimenzija, obično se misli na ovaj uređaj kada se kaže Arduino, veća memorija i broj pinova od minijaturnih varijanti | <ul style="list-style-type: none"><li>- ATmega 2560 mikrokontroler</li><li>- 256 KB memorije</li><li>- 54 ulaza/izlaza (15 od njih za PWM)</li><li>- 16 analognih pinova</li></ul>                                                                           |
| Arduino Nano        | Još manji od Uno, namenjen uređajima minijaturnih veličina                                                                        | <ul style="list-style-type: none"><li>- Atmega 328 mikrokontroler</li><li>- 22 digitalna u/i (6 za PWM)</li><li>- 8 analognih ulaza</li></ul>                                                                                                                |
| Arduino Yun         | Mogućnost pokretanja Linux-a pored Arduino sistema, a i komunikacija modula                                                       | <ul style="list-style-type: none"><li>- ATmega32U4 čip uparen sa Atheros AR9331 mikroprocesorom</li><li>- WiFi konekcija</li></ul>                                                                                                                           |
| LilyPad             | Specijalizovana varijanta namenjena primeni u e-tekstilnim proizvodima i kao dodatak odeći, mogućnost ušivanja                    | <ul style="list-style-type: none"><li>- ATmega32u4</li><li>- 9 digitalnih pinova (4 za PWM)</li><li>- 8 MHz resonator</li><li>- micro USB konekcija</li><li>- 3.7V LiPo baterijad.</li></ul>                                                                 |
| Zastarele varijante | Mega ADK – namenjen Android uređajima, NG, Diecimila, Duemilanove                                                                 |                                                                                                                                                                                                                                                              |
| Arduino Due         | Sličan kao Mega, ali 32-bit                                                                                                       | 32bit ARM Cortex-M3, 84Mhz. 512KB programske memorija                                                                                                                                                                                                        |

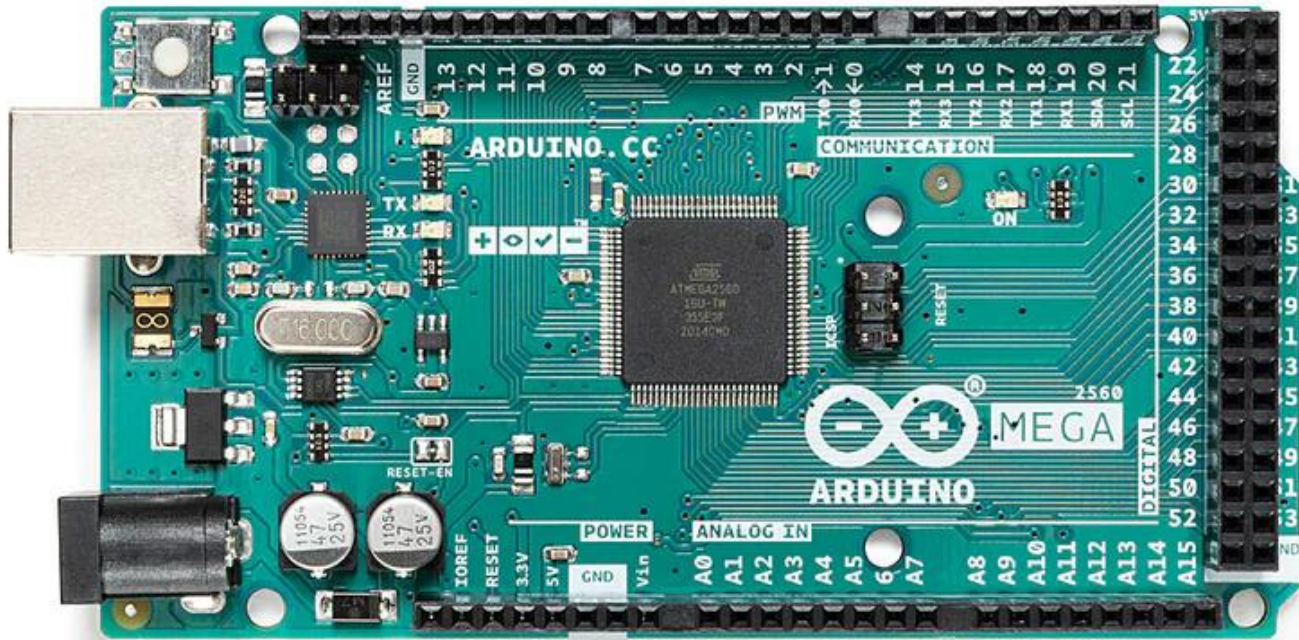
# Različiti Arduino modeli





# Arduino Mega 2560

- Ploča standardnih dimenzija, veći broj pinova od minijaturnih (Uno, Nano)
- Kompatibilian sa shieldovima legacy verzija (Duemilanove, Diecimila)



|                            |                                                |
|----------------------------|------------------------------------------------|
| Mikrokontroler             | <a href="#">ATmega2560</a>                     |
| Operativni napon           | 5V                                             |
| Ulazni (preporučeni) napon | 7-12V                                          |
| Limit napona               | 6-20V                                          |
| Digitalni U/I pinovi       | 54 (od kojih 15 imaju PWM izlaz)               |
| Analogni ulazi             | 16                                             |
| DC po I/O pinu             | 20 mA                                          |
| DC za 3.3V Pin             | 50 mA                                          |
| Flash programska memorija  | 256 KB od kojih je 8 KB rezervisano bootloader |
| SRAM                       | 8 KB                                           |
| EEPROM                     | 4 KB                                           |
| Takt                       | 16 MHz                                         |
| Ugrađeni LED               | 13                                             |
| Dužina                     | 101.52 mm                                      |
| Širina                     | 53.3 mm                                        |
| Masa                       | 37 g                                           |
| Zvanična cena              | 42 eur                                         |



# Digitalni interfejs

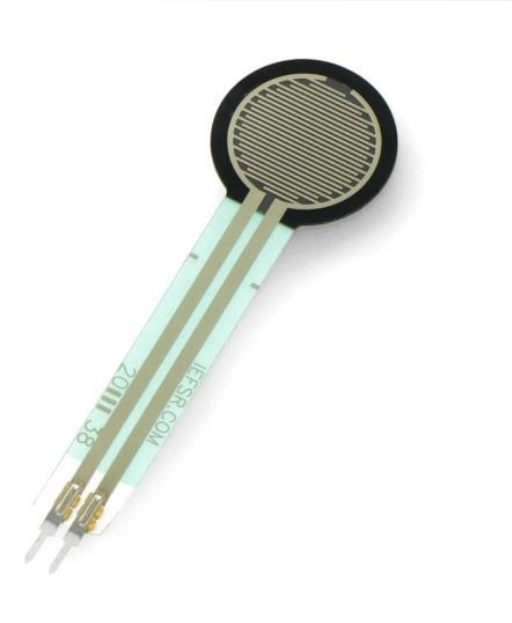
- Digitalni pinovi
  - Imaju dva moguća stanja: HIGH/LOW, 1/0 ili 5V/0V
- Ulazni pinovi
  - Ulaz se obično koristi za osluškivanje prisustva napona kada se prekidač otvori ili zatvori
  - Digitalni ulazi kao osnova za nebrojivo mnogo digitalnih komunikacionih protokola
  - Binarni signal se može kreirati kao kombinacija HIGH ili LOW impulsa, što se može primeniti za čitanje digitalnih senzora, poput ultrazvučnog
- Izlazni pinovi
  - Primena za paljenje i gašenje uređaja (poput treptanja LED dioda)
  - Omogućava digitalnu komunikaciju paljenjem/gašenjem pinova
  - Prepoznat binarni signal koji se koristi u komunikacionim protokolima sa drugim uređajima



# Analogni interfejs

- Analogni ulaz

- Osim digitalnih, Arduino poseduje i analogne ulazne pinove
- Ovi pinovi primaju analogni signal i vrše 10-bitnu analogno-digitalnu konverziju (ADC) sa ciljem da ga pretvore u broj između 0 i 1023 (koraci od po 4.9mV)
- Ovakav tip ulaza je pogodan za čitanje rezistivnih senzora
  - Rezistivni senzori funkcionišu tako što pružaju otpornost u kolu
- Analogni ulazi se mogu koristiti za čitanje promenljivog signala napona u rasponu od 0 do 5V
  - Koristi se kod interakciju sa različitim analognim uređajima



- Analogni izlaz

- Jedna od specijalnih funkcionalnosti Arduina jeste modulacija širine (trajanja) impulsa – Pulse Width Modulation (PWM)
- PWM predstavlja način kako Arduino zapravo može generisati izlazni signal koji nalikuje na analogni
- PWM radi tako što se naizmenično smenjuju HIGH i LOW (visok i nizan naponski nivo) sa ciljem simulacije analognog signala
  - Recimo ako 1 ms upaljen LED, a 9 ms ugašen, onda će delovati kao da je jačina svetlosti 1/10 (isto i za napon važi)
- PWM je jedna od ključnih tehnika u pozadini mnogih primena, kao što su generisanje zvuka, kontrola jačine osvetljenja i brzina motora



# Kostur Arduino programa

- Arduino program se naziva i sketch (nacrt)
- Obično ima 5 delova
  - Zaglavlje
  - Definicija globalih promenljivih
    - Obično inicijalizacija pomoćnih promenljivih
    - Dodeljivanje konstanti različitim Arduino pinovima
  - Setup rutina
    - Postavljamo inicijalne uslove promenljivih
    - Inicijalizacioni kod
    - Želimo da se izvrši samo jednom
  - Loop rutina
    - Glavna rutina sketch-a
    - Ne samo što je ovo glavni program, već se i ponavlja neprestano dok se skeč izvršava
  - Pomoćne funkcije
    - Korisnički efinisane
    - Mogu se izvršavati isključivo unutar setup i loop rutina

```
// globalne promenljive, konstante
const int LEDPin = 9;

void setup() {
    // inicijalizacija, jednom se izvršava
    f1();
}

void loop() {
    // glavni program, ponavlja se
    fN();
}

//funkcije
void f1(){
    ...
}
...
void fN(){
    ...
}
```

# Karakteristike Arduino programa

- Samo dva dela programa su obavezna
  - setup i loop rutine
- Kod se piše u Arduino jeziku koji liči na C
- Skoro sve naredbe se završavaju sa ;
  - Uslovne/grananja su izuzetak
- Promenljive moraju biti definisane (navedene u kodu) i imati eksplicitno naznačen tip pre nego što mogu da se koriste

# Tipovi podataka

- boolean (8 bit)
  - jednostavno true/false
- byte (8 bit)
  - neoznačeni broj u opsegu 0-255
- char (8 bit)
  - označeni broj od -128 do 127
- unsigned char (8 bit)
  - isto kao bajt, preferira se upotreba 'byte'
- word (16 bit)
  - neoznačeni broj 0-65535
- unsigned int (16 bit)
  - isto kao 'word', preferirano koristiti 'word'
- int (16 bit)
  - označeni brojevi od -32768 to 32767, najčešće se koriste za globalne pomenljive opšte namene
- unsigned long (32 bit)
  - Neoznačeni broj u opsegu 0-4,294,967,295.
  - Najčešće se koristi za čuvanje rezultata koji vraća millis() funkcija – broj milisekundi koliko se kod izvršava
- long (32 bit)
  - označenu broj od -2,147,483,648 do 2,147,483,647
- float (32 bit)
  - označeni broj od -3.4028235E38 do 3.4028235E38
  - nije direktno podržan od Arduina, zahtevno za kompajler
  - ne preporučuje se upotreba



# Korisne funkcije – rad sa vremenom

- Unošenje čekanja/kašnjenja
  - **delay(ms:int)**
  - Čekanje pre nego što se izvrši sledeća linija koda
  - Argument ceo broj koji predstavlja milisekunde
- Pribavljanje trenutnog vremena
  - **milis()** – u milisekundama
  - **micros()** – u mikrosekundama
  - Meri koliko je prošlo od pokretanja ploče
  - Korisno za merenje vremena izvršenja, koliko vremena prošlo

# Rad sa digitalnim pinovima

- Upis vrednosti
  - digitalWrite(pin, value)
    - pin – redni broj digitalnog pina (0...broj-1)
    - Value – HIGH ili LOW
- Čitanje vrednosti
  - digitalRead(pin)
    - pin – redni broj pina koji čitamo
    - Povratna vrednost: HIGH ili LOW
- Postavljanje režima
  - pinMode(pin, mode)
    - Pin – broj pina čiji smer postavljamo
    - Mode
      - OUTPUT
        - izlaz
      - INPUT
        - ulaz
      - INPUT\_PULLUP
        - Ulaz sa omogućenim pullup provodnikom

```
int ledPin = 13; // LED na digitalnom 13
int inPin = 7;   // dugme na digitalnom 7
int val = 0;     // pomoćna promenljiva

void setup() {
    pinMode(ledPin, OUTPUT); // postavlja 13 za izlaz
    pinMode(inPin, INPUT);   // postavlja 7 za ulaz
}

void loop() {
    val = digitalRead(inPin); // čita ulazni pin
    digitalWrite(ledPin, val); // postavlja LED zavisno od dugmeta
}
```

# Rad sa analognim pinovima – čitanje vrednosti

- Čitanje vrednosti
  - **analogRead(pin)**
    - pin – redni broj pina koji čitamo
      - **A0-A5 na većini ploča**
      - A0-A7 na Nano
      - **A0-A15 na Mega 2560**
  - Povratna vrednost je celobrojnog tipa
    - Zavisi od rezolucije ADC
    - 0-1023 za 10bit, 0-4095 za 12bit

```
int analogPin = A0; // potenciometar ovde povezan
                    // izvan vodi na uzemljenje i 5V
int val = 0;        // pomoćna promenljiva

void setup() {
  Serial.begin(9600); // pokreni serijski monitor zbog debugiranja
}

void loop() {
  val = analogRead(analogPin); // pročitaj ulazni pin
  Serial.println(val);          // štampaj vrednost
}
```

# Rad sa analognim pinovima – upis vrednosti

- **analogWrite(pin, value)**

- pin: pin na koji se upisuje
- value: vreme ciklusa – vrednost između 0 (uvek ugašen) i 255 (uvek upaljen)
- Upisuje analognu vrednost (PWM talas) na pin
- Recimo, može se primeniti za promenu jačina osvetljenja LED ili varijacije brzine motora
- Po pozivu analogWrite(), kreira se kontinualni kvadratni talas sa specificiranim ciklusom, sve dok se pozove neka od operacije za isti pin: analogWrite(), digitalWrite() ili digitalWrite()

- Pregled PWM pinova po modelima

- Kao dodatak, MKR i Zero za razliku od ostalih imaju pravi analogni izlaz kada se analogWrite koristi za DAC0() pin
- Dodatno, Arduino Due ima pravi analogni izlaz za pin A0

| PLOČA                      | PWM PINOVI                      | PWM FREKVENCIJA                               |
|----------------------------|---------------------------------|-----------------------------------------------|
| Uno, Nano, Mini            | 3, 5, 6, 9, 10, 11              | 490 Hz (pinovi 5 i 6: 980 Hz)                 |
| Mega                       | 2-13, 44-46                     | 490 Hz (pinovi 4 i 13: 980 Hz)                |
| Leonardo, Micro, Yun       | 3, 5, 6, 9, 10, 11, 13          | 490 Hz (pinovi 3 i 11: 980 Hz)                |
| Uno WiFi Rev.2, Nano Every | 3, 5, 6, 9, 10                  | 976Hz                                         |
| MKR boards *               | 0 - 8, 10, A3 (18), A4 (19)     | 732Hz                                         |
| MKR1000 WiFi *             | 0 - 8, 10, 11, A3 (18), A4 (19) | 732Hz                                         |
| Zero *                     | 3 - 13, A0 (14), A1 (15)        | 732Hz                                         |
| Nano 33 IoT *              | 2, 3, 5, 6, 9 - 12, A2, A3, A5  | 732Hz                                         |
| Nano 33 BLE/BLE Sense      | 1 - 13, A0 - A7                 | 500Hz                                         |
| Due **                     | 2-13                            | 1000Hz                                        |
| 101                        | 3, 5, 6, 9                      | Pinovi 3 i 9: 490 Hz,<br>Pinovi 5 i 6: 980 Hz |

# Primer za analogni upis

- Postavljanje vrednosti jačine osvetljenja LED diode proporcionalno potencijometru
- Za ovaj slučaj nema potrebe podešavati smer pinova uz pomoć pinMode() ako je GND pin povezan
  - analogWrite već u sebi zove podešavanje smera kao izlaz
  - Ako GND nije povezan, onda mora da se ručno postavi smer kao izlazni u setup

```
int ledPin = 9; // LED spojen na digitalni pin 9
int analogPin = 3; // potencijometar na pinu 3
int val = 0; // promenljiva za čuvanje pročitane vrednosti

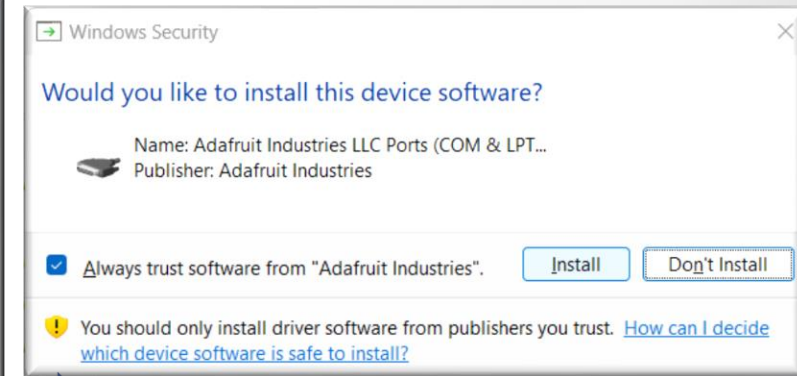
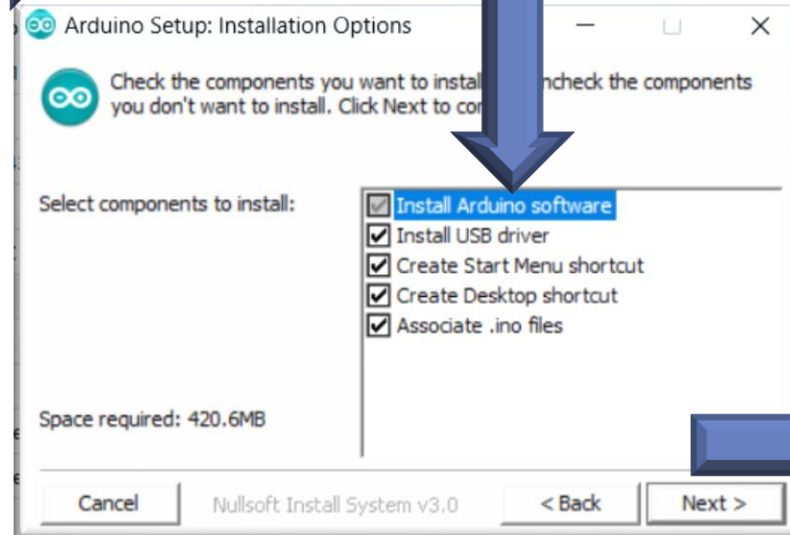
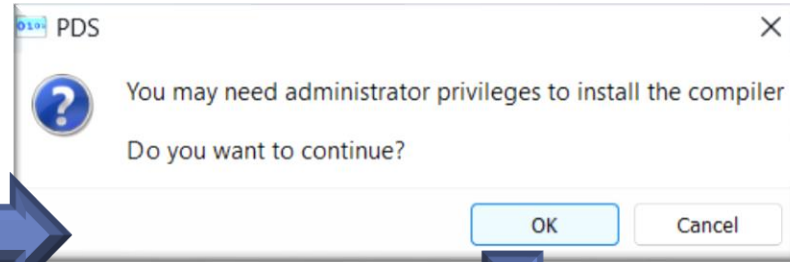
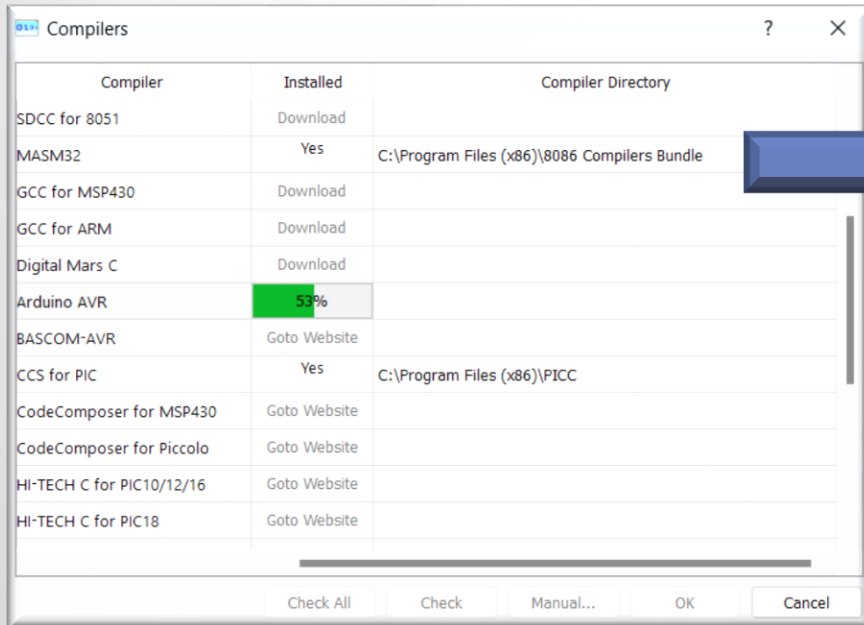
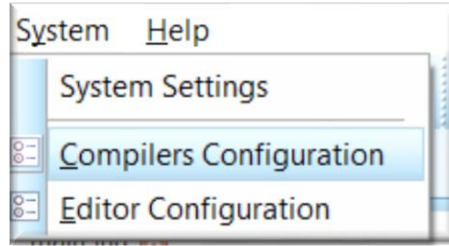
void setup() {
    pinMode(ledPin, OUTPUT); // postavi pin kao izlaz
}

void loop() {
    val = analogRead(analogPin); // čitanje vrednosti pina potencijometra
    analogWrite(ledPin, val / 4); // delimo sa 4, jer analogRead vraća 0-1023, a analogWrite očekuje 0-255
}
```



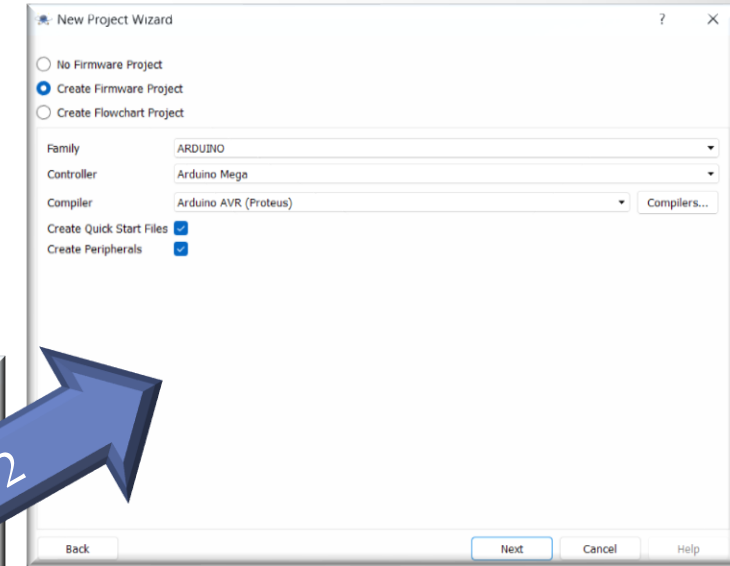
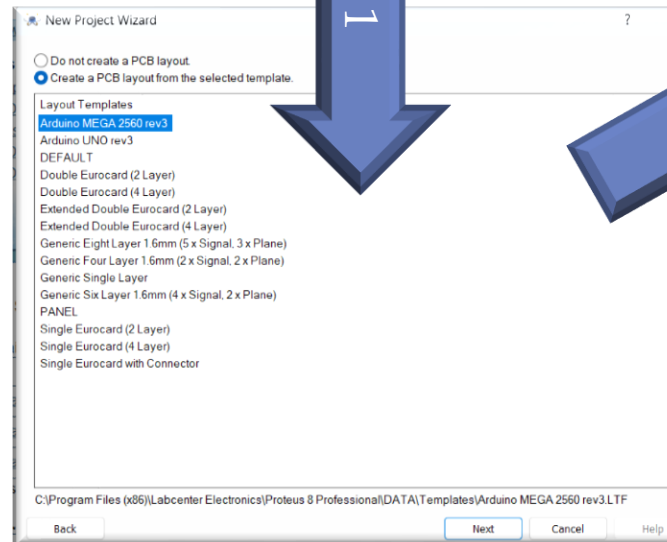
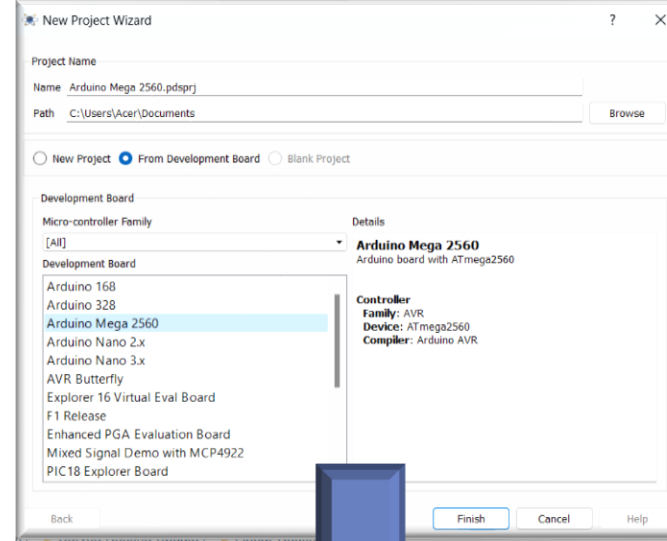
# Instalacija AVR kompajlera za Proteus

- U editor koda
  - System->Compilers Configuration
- Pronaći Arduino AVR
  - Preuzeti i instalirati



# Mega 2560 projekat u Proteusu

- Prilikom kreiranja novog projekta odabrati
  - From Development Board
  - Selektovati **Arduino Mega 2560**
- Odabrati Create PCB layout fromm...
  - **Arduino MEGA 2560 rev3**
- Kreirati Firmware project
  - **Family: Mega**
  - **Controller: Arduino Mega**
  - **Compiler: Arduino AVR (Proteus)**



# Zadatak 1

- Napisati Arduino program za ploču Mega 2560 koji svake sekunde naizmenično gasi i pali svake sekunde žutu LED diodu poezanu preko pina 13.

- Setup
  - Pin 13 kao izlazni
- Loop
  - Gasimo LED na pinu 13
  - Pauza 1s
  - Palimo LED na pinu 13
  - Pauza 1s

```
void setup()
{
  // put your setup code here, to run once:
  pinMode(13, OUTPUT);
}

void loop()
{
  // put your main code here, to run repeatedly:
  digitalWrite(13, LOW);
  delay(1000);
  digitalWrite(13, HIGH);
  delay(1000);
}
```

