

# Bevezetés az Arduino és az elektronika világába I.

Agócs Norbert és Nagy Dániel

2020. február 14.

# Tartalom

- 1 Áram
- 2 Arduinio alapok
- 3 LED
- 4 Programozás alapok
- 5 Feladatok I
- 6 Villogó LED
- 7 Feladatok II

# Tartalom

- 1 Áram
- 2 Arduinio alapok
- 3 LED
- 4 Programozás alapok
- 5 Feladatok I
- 6 Villogó LED
- 7 Feladatok II

# Mi az az áram?

Töltés

# Mi az az áram?

## Töltés

Elektronnak (-) és a protonnak (+) van ilyene. Pozitív vagy negatív lehet.

# Mi az az áram?

## Töltés

Elektronnak (-) és a protonnak (+) van ilyene. Pozitív vagy negatív lehet.

## Áram

# Mi az az áram?

## Töltés

Elektronnak (-) és a protonnak (+) van ilyene. Pozitív vagy negatív lehet.

## Áram

Elektronok mozgása egy vezetőképes közegben, például egy fém vezetékben.

# Mi az az áram?

## Töltés

Elektronnak (-) és a protonnak (+) van ilyene. Pozitív vagy negatív lehet.

## Áram

Elektronok mozgása egy vezetőképes közegben, például egy fém vezetékben.

## Feszültség



# Mi az az áram?

## Töltés

Elektronnak (-) és a protonnak (+) van ilyene. Pozitív vagy negatív lehet.

## Áram

Elektronok mozgása egy vezetőképes közegben, például egy fém vezetékben.

## Feszültség

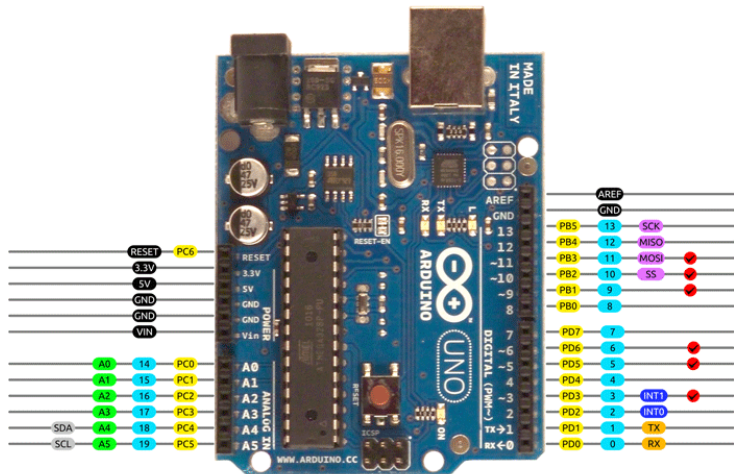
Valami ami hatására az elektronok mozogni kezdenek. Két pont között értelmezzük, minél nagyobb annál jobban át akarnak menni egyik helyről a másira az elektronok.

# Tartalom

- 1 Áram
- 2 **Ardunio alapok**
- 3 LED
- 4 Programozás alapok
- 5 Feladatok I
- 6 Villogó LED
- 7 Feladatok II

# Arduino

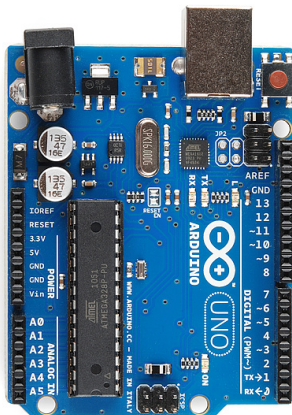
## PIN-ek



AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT

# Arduino

## PIN-ek



### Digitális PIN-ek

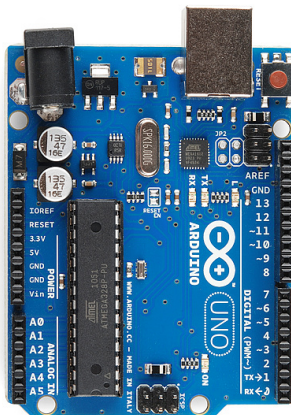
- 0-13 kimenetek
- Beállítás 0V vagy 5V-ra
- Olvasás hogy van-e áram

### Analóg PIN-ek

- A0-A5 pinek
- Leolvasható hogy pontosan mekkora a feszültség 0V és 5V között

# Arduino

## PIN-ek

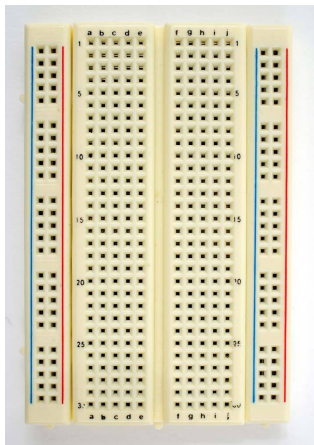


### POWER Pinek

- GND - 0V, földelés
- 5V - mindig 5V-ot ad ki
- 3.3V - mindig 3.3V-ot ad ki
- Vin - Bemeneti feszültség, ha 9V-os elem, akkor 9V lesz. Számítógép esetén 5V.

# Breadboard

## Próbapanel



A lyukak 5-ösével vannak összekötve (soronként). A piros és kék csík melletti lyukak is össze vannak kötve egymással.

# Tartalom

- 1 Áram
- 2 Arduinio alapok
- 3 LED**
- 4 Programozás alapok
- 5 Feladatok I
- 6 Villogó LED
- 7 Feladatok II

# Light Emitting Diode (LED)

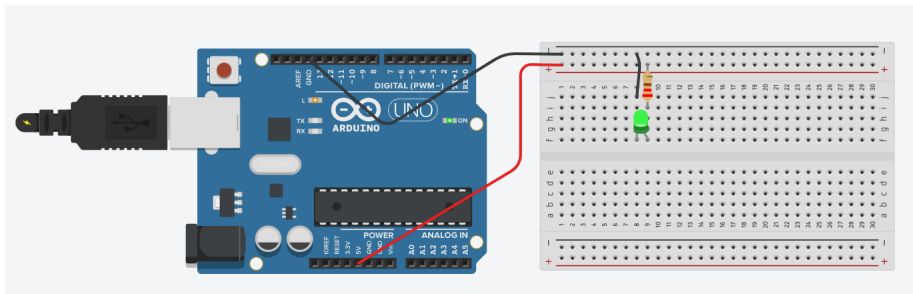
Fényt kibocsátó dióda



- Ha áram folyik rajta keresztül akkor világít
- Hosszabbik láb - Pozitív
- Rövidebbik láb - Negatív
- Max 3.3V-ra szabad kötni



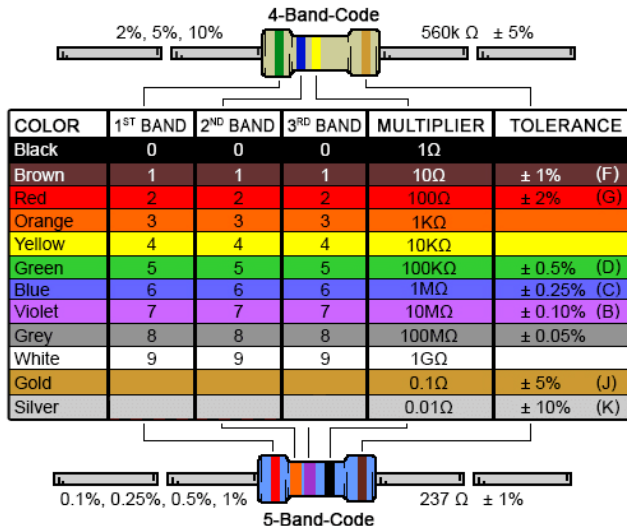
# Csináljunk világító LED-et!



Ellenállás legyen  $100\Omega$  és  $500\Omega$  között.

Ha nem világít a LED akkor fordítva kötöttük be.

# Megfelelő ellenállás választása

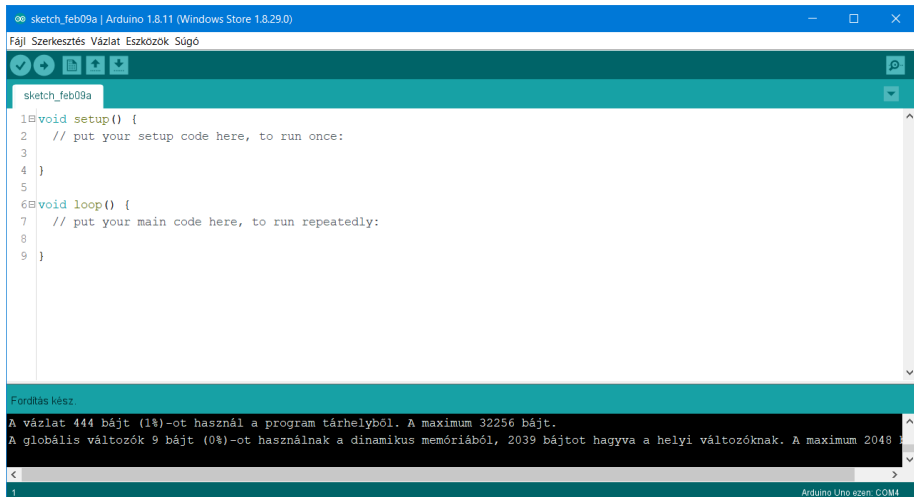


# Tartalom

- 1 Áram
- 2 Arduinio alapok
- 3 LED
- 4 Programozás alapok**
- 5 Feladatok I
- 6 Villogó LED
- 7 Feladatok II

# Arduino IDE

Miután megnyitottuk a programot...



The screenshot shows the Arduino IDE interface. The title bar reads "sketch\_feb09a | Arduino 1.8.11 (Windows Store 1.8.29.0)". The menu bar includes "Fájl", "Szerkesztés", "Vázlat", "Eszközök", and "Súgó". The toolbar contains icons for opening, saving, and running. The main text area shows the following code:

```
1 void setup() {  
2   // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7   // put your main code here, to run repeatedly:  
8  
9 }
```

Below the code editor, a status bar indicates "Fordítás kész." (Compilation ready). At the bottom, a black console window displays memory usage information: "A vázlat 444 bájt (1%-ot használ a program tárhelyből. A maximum 32256 bájt. A globális változók 9 bájt (0%-ot használ a dinamikus memóriából, 2039 bájtot hagyva a helyi változóknak. A maximum 2048 bájt)." (The sketch uses 444 bytes (1% of program memory, max 32256 bytes). Global variables use 9 bytes (0% of dynamic memory, leaving 2039 bytes for local variables). Maximum 2048 bytes). The bottom status bar shows "1" and "Arduino Uno ezen: COM4".

# Arduino kiválasztása

sketch\_feb09a | Arduino 1.8.11 (Windows Store 1.8.29.0)

Fájl Szerkesztés Vázlat Eszközök Súgó

✓ ↻ 📄 ⬆️ ⬇️

sketch\_feb09a

```
1 void setup()  
2   // put you  
3  
4 }  
5  
6 void loop()  
7   // put you  
8  
9 }
```

Automatikus formázás Ctrl+T

Vázlat arhiválása

Kódolás javítása & újratöltés

Könyvtárak kezelése... Ctrl+Shift+I

Soros monitor Ctrl+Shift+M

Soros plotter Ctrl+Shift+L

WiFi101 / WiFiNINA Firmware Updater

Alaplap: "Arduino Uno" >

Port > Soros portok

Alaplap infó beszerzése COM5

Programozó: "AVRISP mkII" >

Bootloader égetése

# Programozás

```
void setup()  
{  
    //ami ide kerül 1x fut le  
}  
  
void loop()  
{  
    //ami ide kerül végtelenségig fut  
}
```

# Hello World program

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    Serial.print("Hello_world");
    delay(10000); //10000ms = 10s várakozás
}
```

- `Serial.begin(9600);` elindítja a kommunikációt az Arduino és a gép között
- `Serial.print("valami");` kiiratja "valami"-t a Soros monitorra
- `delay(1000);` várakozik 1000 ms-t

# Jegyezzük meg

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    Serial.print("Hello_world");
    delay(10000); //10000ms = 10s várakozás
}
```

- Minden cikluson belüli dolog egy tabulátorral (Tab) beljebb van kezdve
- A // mögé írt szöveget nem veszi figyelembe az Arduino (Úgynevezett comment)



# Matekozzunk!

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    int a = 13;
    int b = 7;
    int c = a + b;
    Serial.print(c);
    delay(10000);
}
```

int egy **típus** egy **egész számot** tárol  $-32768$  és  $32767$  között.  
a, b, c úgynevezett **változók** amiknek értékeket is adtunk.

# Matekozzunk!

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  float a,b,c;
  a = 0.5;
  b = 20.2;
  c = a * b;
  Serial.print(c); //c = 10.1
  delay(10000);
}
```

float egy **típus** egy **tört számot** tárol 8 jegy pontossággal.

a, b, c szintén **változók**, azonban itt más a típusok. Most float.

# Mik is ezek a változók?

Változó

# Mik is ezek a változók?

## Változó

Valamilyen adatot tárol. Ezt a tárolt adatot meg is tudjuk változtatni, vagy ki tudjuk iratni. Műveleteket tudunk végezni velük.

# Mik is ezek a változók?

## Változó

Valamilyen adatot tárol. Ezt a tárolt adatot meg is tudjuk változtatni, vagy ki tudjuk iratni. Műveleteket tudunk végezni velük.

## Típus

# Mik is ezek a változók?

## Változó

Valamilyen adatot tárol. Ezt a tárolt adatot meg is tudjuk változtatni, vagy ki tudjuk iratni. Műveleteket tudunk végezni velük.

## Típus

A változó tulajdonsága. Megadja hogy a változó mit tárol, ezt csak egyszer kell megadni minden változó esetén. Lehet egész számot tároló típus `int` vagy `long`. Illetve tört számot tároló `float` vagy `double`.

# Mik is ezek a változók?

## Változó

Valamilyen adatot tárol. Ezt a tárolt adatot meg is tudjuk változtatni, vagy ki tudjuk iratni. Műveleteket tudunk végezni velük.

## Típus

A változó tulajdonsága. Megadja hogy a változó mit tárol, ezt csak egyszer kell megadni minden változó esetén. Lehet egész számot tároló típus `int` vagy `long`. Illetve tört számot tároló `float` vagy `double`.

## Deklaráció

Változók létrehozása. Először megadjuk a típust, aztán nevet adunk a változónak, majd a változónak értéket adunk.

## Példa

```
int alma = 123; //típus: int // név: alma //érték: 123
```

# Tartalom

- 1 Áram
- 2 Arduinio alapok
- 3 LED
- 4 Programozás alapok
- 5 Feladatok I**
- 6 Villogó LED
- 7 Feladatok II



# Feladat I

Mit ír ki a kód?

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    int a = 25;
    int b = 5;
    int c = a / b;
    int d = c + b + a;
    Serial.print(d);
    delay(10000);
}
```

# Feladat I

Mit ír ki a kód?

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    int a = 25;
    int b = 5;
    int c = a / b;
    int d = c + b + a;
    Serial.print(d);
    delay(10000);
}
```

Megoldás:  $c = 5$ ;  $d = 35$ ;

# Feladat II

Mit ír ki a kód?

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    double alma,korte,eper,cseresznye;
    alma = 10.0;
    korte = 2.5;
    eper = alma/korte + alma;
    cseresznye = (eper - 4)*eper;
    Serial.print(cseresznye);
    delay(10000);
}
```

# Feladat II

Mit ír ki a kód?

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    double alma,korte,eper,cseresznye;
    alma = 10.0;
    korte = 2.5;
    eper = alma/korte + alma;
    cseresznye = (eper - 4)*eper;
    Serial.print(cseresznye);
    delay(10000);
}
```

Megoldás:

$eper = 4 + 10 = 14$ ;  $cseresznye = (14-4)*14 = 140$ ;

# Feladat III

Mit ír ki a kód?

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    int a = 2;
    float b = 1.5;
    float c = a*b + 2*2 - 4;
    Serial.print(c);
    delay(10000);
}
```

# Feladat III

Mit ír ki a kód?

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    int a = 2;
    float b = 1.5;
    float c = a*b + 2*2 - 4;
    Serial.print(c);
    delay(10000);
}
```

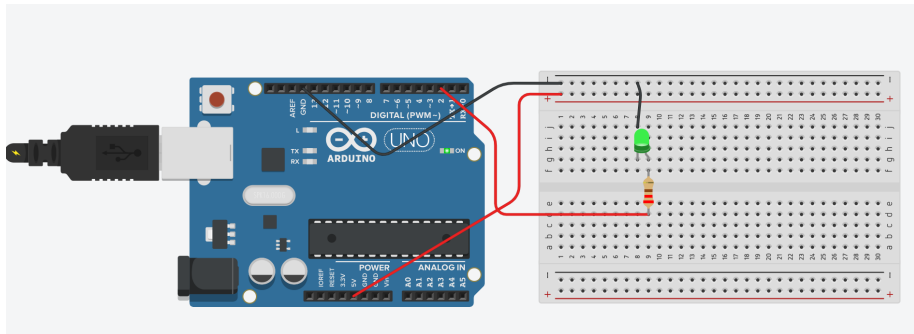
Megoldás:

$c = 3;$

# Tartalom

- 1 Áram
- 2 Arduino alapok
- 3 LED
- 4 Programozás alapok
- 5 Feladatok I
- 6 Villogó LED**
- 7 Feladatok II

# Csináljunk villogó LED-et!



Válasszunk  $220\Omega$ -os ellenállást.  
Ez most nem fog még világítani!



# Írjuk meg a kódot!

```
int LED = 2;

void setup()
{
    pinMode(LED, OUTPUT); //beállítja a LED pint (2-es) kimenetre
}

void loop()
{
    digitalWrite(LED, HIGH); //kiad 5V-ot a 2-es pinen
}
```

- `pinMode(pin, OUTPUT/INPUT)` beállítja a megadott számú pint kimenetre (OUTPUT) vagy bemenetre (INPUT)
- `digitalWrite(pin, HIGH/LOW)` beállítja a megadott számú pin-en hogy folyjon-e áram HIGH folyik áram, LOW nem folyik áram

# És most villogjon is!

```
int LED = 2;

void setup()
{
    pinMode(LED, OUTPUT); //beállítja a LED pint (2-es) kimenetre
}

void loop()
{
    digitalWrite(LED, HIGH); //kiad 5V-ot a 2-es pinen
    delay(1000);
    digitalWrite(LED, LOW); //kikapcsolja a 2-es pinen az áramot
    delay(1000);
}
```

# Tartalom

- 1 Áram
- 2 Arduinio alapok
- 3 LED
- 4 Programozás alapok
- 5 Feladatok I
- 6 Villogó LED
- 7 Feladatok II**

# Feladatok

Kicsit fejlesszük tovább!

- Tároljátok el egy változóba a várakozási időt **másodpercben**.
- Állítsátok be, hogy LED nagyon gyorsan, vagy nagyon lassan villogjon.
- Írjátok ki a soros monitorra, hogy jelenleg a LED ki van kapcsolva, vagy be van kapcsolva. (Segítség:  
`Serial.println("szoveg")` mindig új sorba ír)

# Megoldás

## Továbbfejlesztett verzió

```
int LED = 2;
float varakozas = 0.3; //0.3 másodpercet vár
int varakozas_ms = varakozas*1000; //0.3*1000 = 300 ms

void setup()
{
    Serial.begin(9600);
    pinMode(LED, OUTPUT); //beállítja a LED pint (2-es) kimenetre
}

void loop()
{
    Serial.println("BE");
    digitalWrite(LED, HIGH); //kiad 5V-ot a 2-es pinen
    delay(varakozas_ms);
    Serial.println("KI");
    digitalWrite(LED, LOW); //kikapcsolja a 2-es pinen az áramot
    delay(varakozas_ms);
}
```

## Szavak I

e Ladung /-en	töltés
s Elektron /-en	elektron
s Proton /-en	proton
r Strom /e Ströme	áram
e Bewegung /-en	mozgás
s Medium / e Medien	közeg
s Metall /-e	fém
e Leitung /-en	vezeték
e Spannung /-en	feszültség
r Ausgang /e Ausgänge	kimenet
r Eingang /e Eingänge	bement
e Erdung	GND avagy földelés
s Breadboard / s Steckbrett	próbapanel

# Szavak II

etw. mit +D verbinden

e Leuchtdiode (LED)

leuchten

r Widerstand / e Widerstände

e Variable /-n

r Typ

r Programmcode

blinkend

vmi vmivel összekötni

led

világítani

ellenállás

változó

típus

programkód

villogó

# Szavak III