



ARDUINO TANÁRI TOVÁBBKÉPZÉS

Szakképzésben
Oktatók
Számára

BEMUTATKOZÁS

- Budapesti Műszaki Szakképzési Centrum
- HDidakt Kft.
- Oktatók



TEMATIKA

- Bemutatókozás
- Arduino
- Arduino Education
- Starter KIT
 - Arduino UNO
 - Környezet
 - Projektek
- Explore IoT KIT
 - MKR Wifi 1010
 - MKR IoT Carrier
 - Projekt IoT
- Engineering KIT
- Science KIT
- CTC GO!
- Befejezés

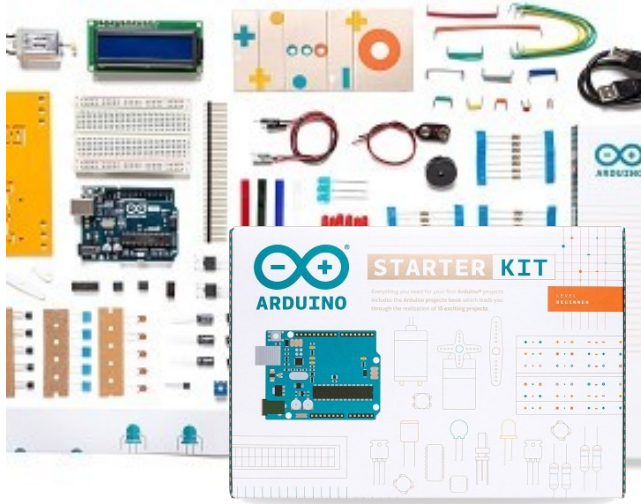
ARDUINO

- Elektronikai Platform, Keretrendszer (open-source)
- Hardver és Szoftver környezet
 - Arduino Board
 - Arduino IDE
- Közösség
- Arduino Education

ARDUINO EDUCATION

- STEAM (Science, Technology, Engineering, Mathematics)
- Oktatási terület (Classroom)
 - Oktatási készletek
- Középiskolától, Egyetemig
- Növekvő komplexitás

Starter Kit



Engineering Kit



Science Kit



CTC GO!

ARDUINO - EDUCATION

Készletek

- **Starter Kit**
- Education Starter Kit
- **Explore IoT Kit**
- Engineering Kit
- Science Kit
- CTC GO!!

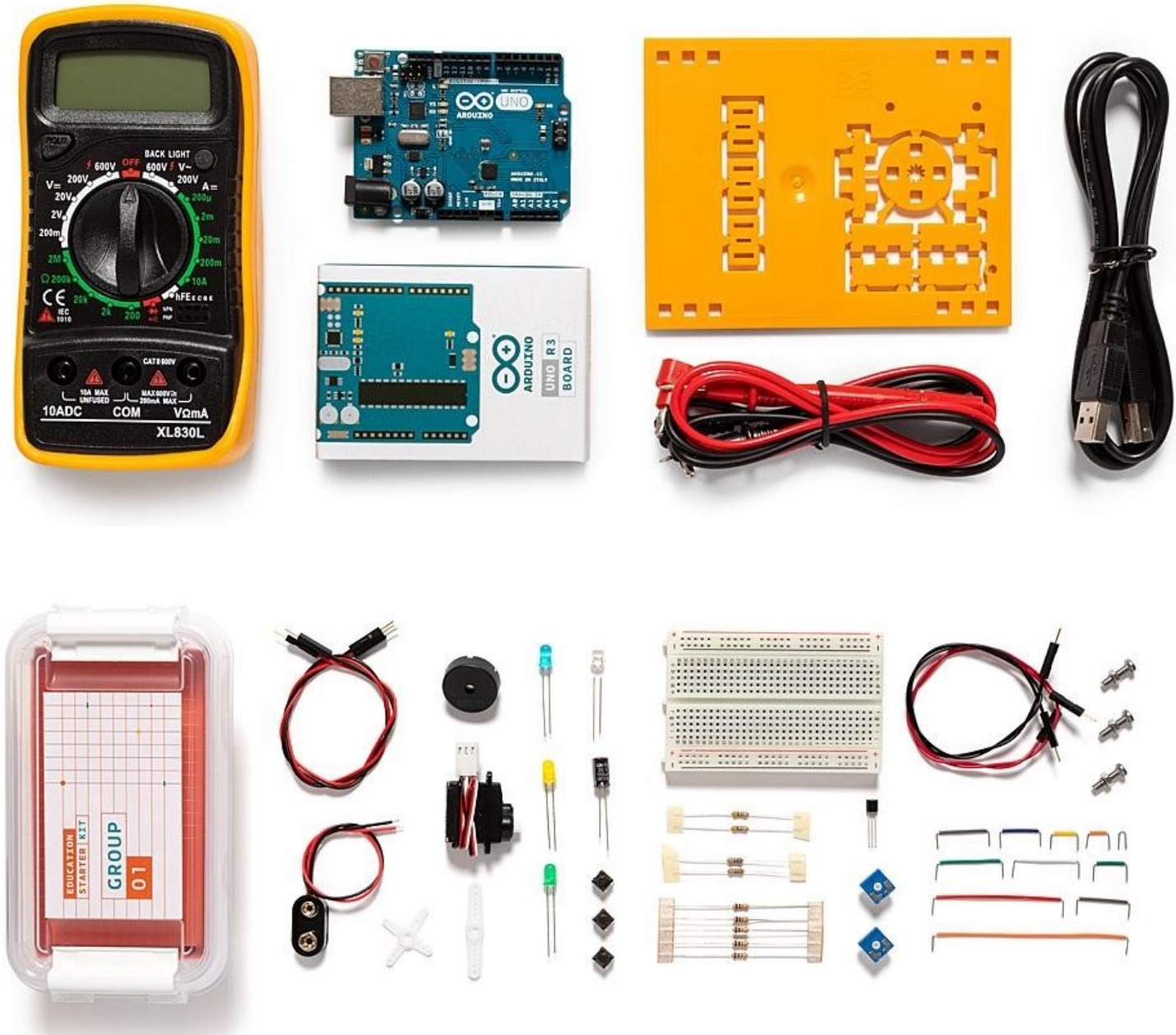
STARTER KIT

- Tartalma
 - Arduino UNO
 - LCD, Szervo
 - Elektronikai alkatrészek
- Oktatási készlet
- Projectbook

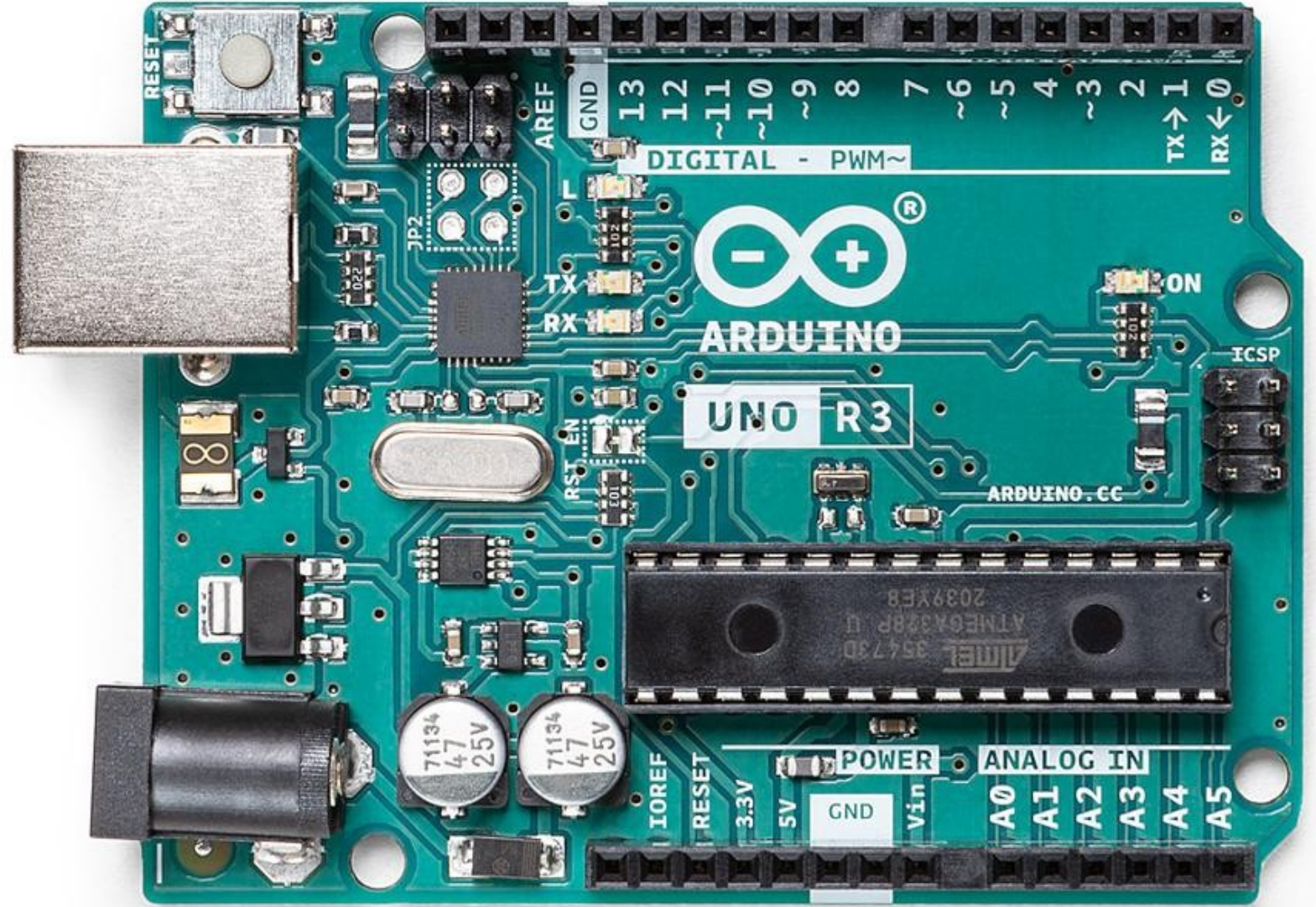


EDUCATION STARTER KIT

- Tartalma
 - 4 - Készlet
 - Arduino UNO
 - Multiméter
 - Szervo, Buzzer
 - Elektronikai alkatrészek
- Online tananyag



ARDUINO UNO



ARDUINO IDE TELEPÍTÉS



Arduino IDE 1.8.16

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

DOWNLOAD OPTIONS

Windows Win 7 and newer

Windows ZIP file

Windows app Win 8.1 or 10 [Get](#) 

Linux 32 bits

Linux 64 bits

Linux ARM 32 bits

Linux ARM 64 bits

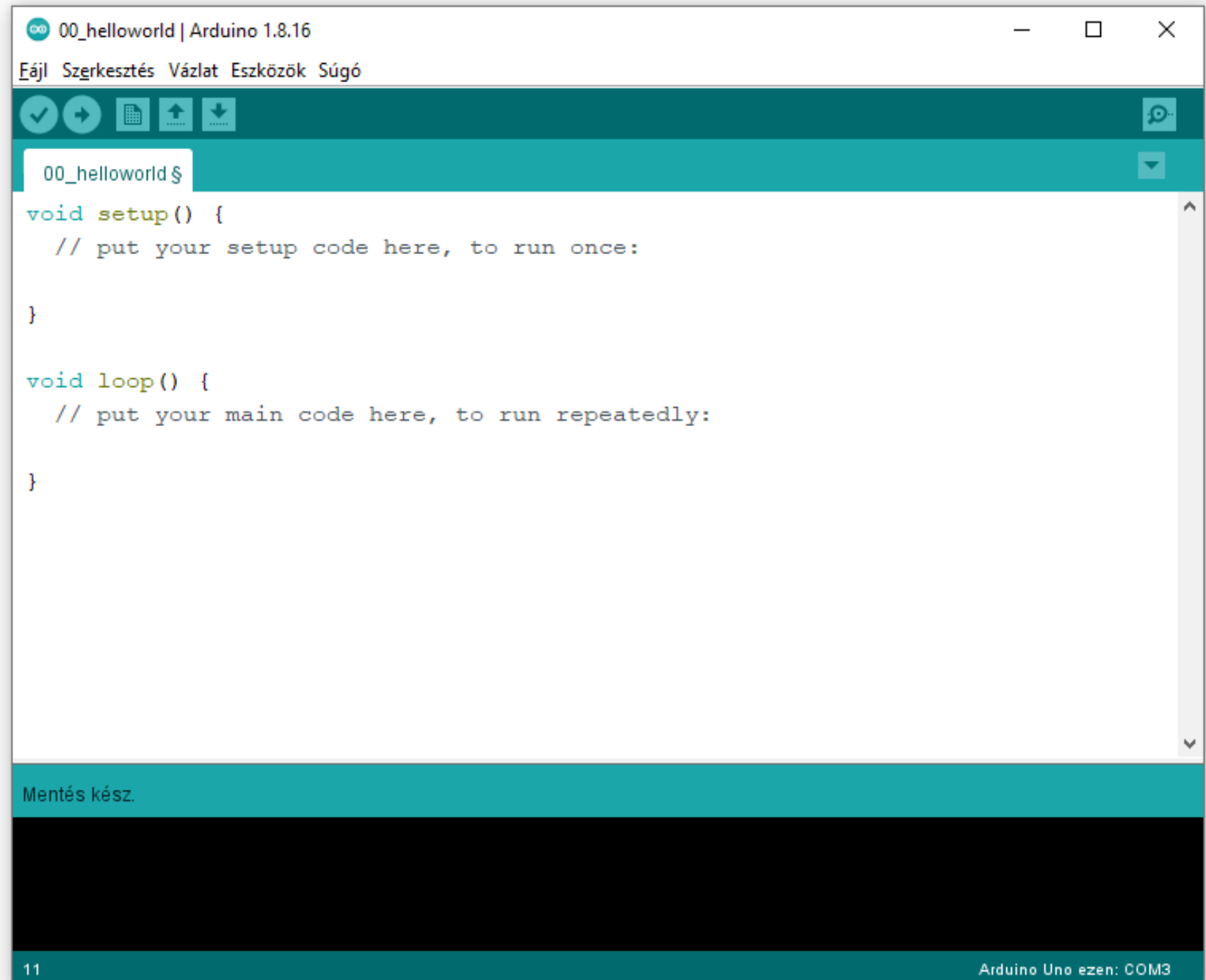
Mac OS X 10.10 or newer

[Release Notes](#) [Checksums \(sha512\)](#)

PROJEKT 1

„HELLO WORLD”

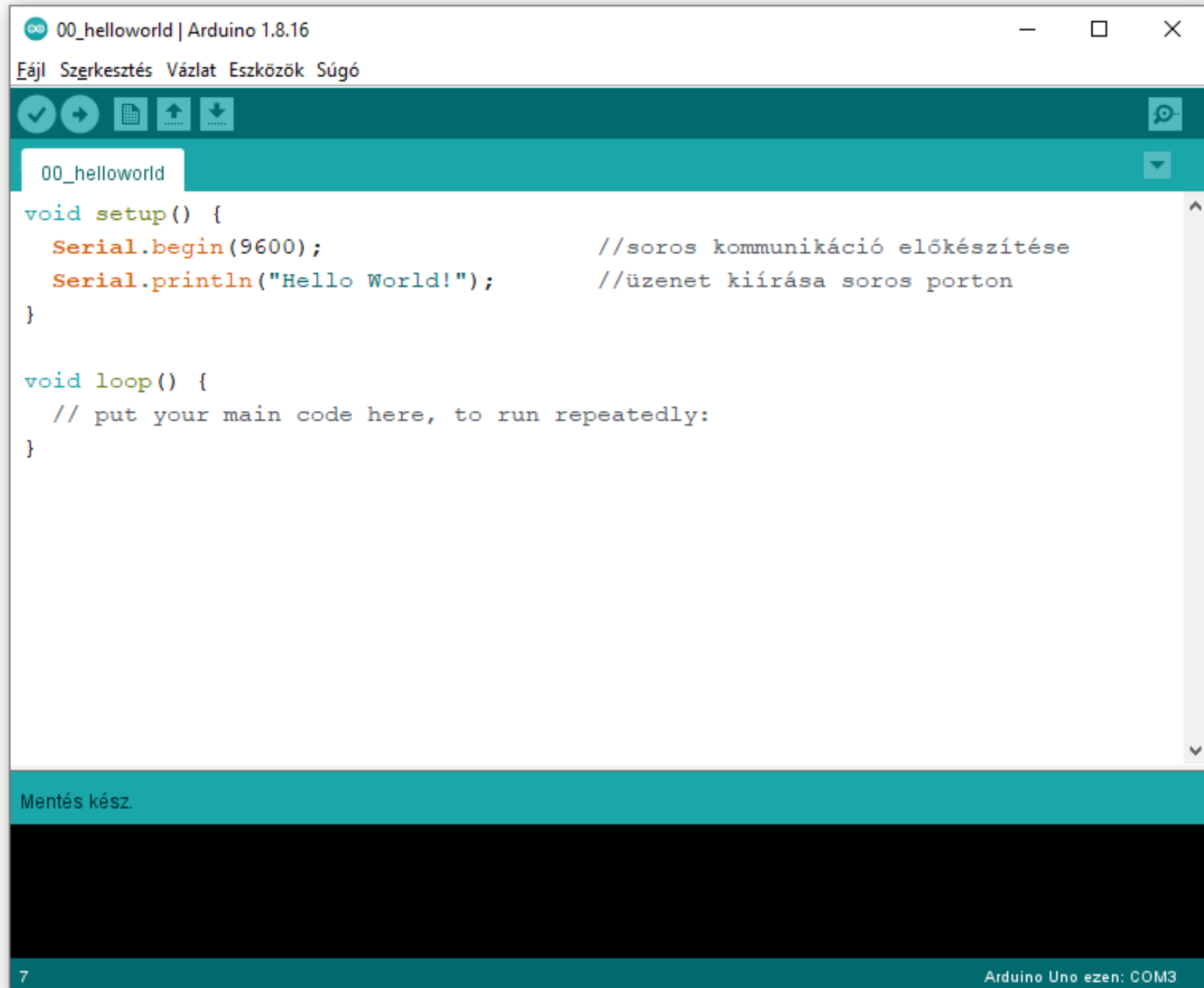
ARDUINO IDE



PROJEKT 1

„HELLO WORLD”

SOROS KÖZVETLEN KÖZVETLEN KÖZVETLEN



```
00_helloworld | Arduino 1.8.16
Fájl Szerkesztés Vázlat Eszközök Súgó

00_helloworld

void setup() {
  Serial.begin(9600);           //soros kommunikáció előkészítése
  Serial.println("Hello World!"); //üzenet kiírása soros porton
}

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

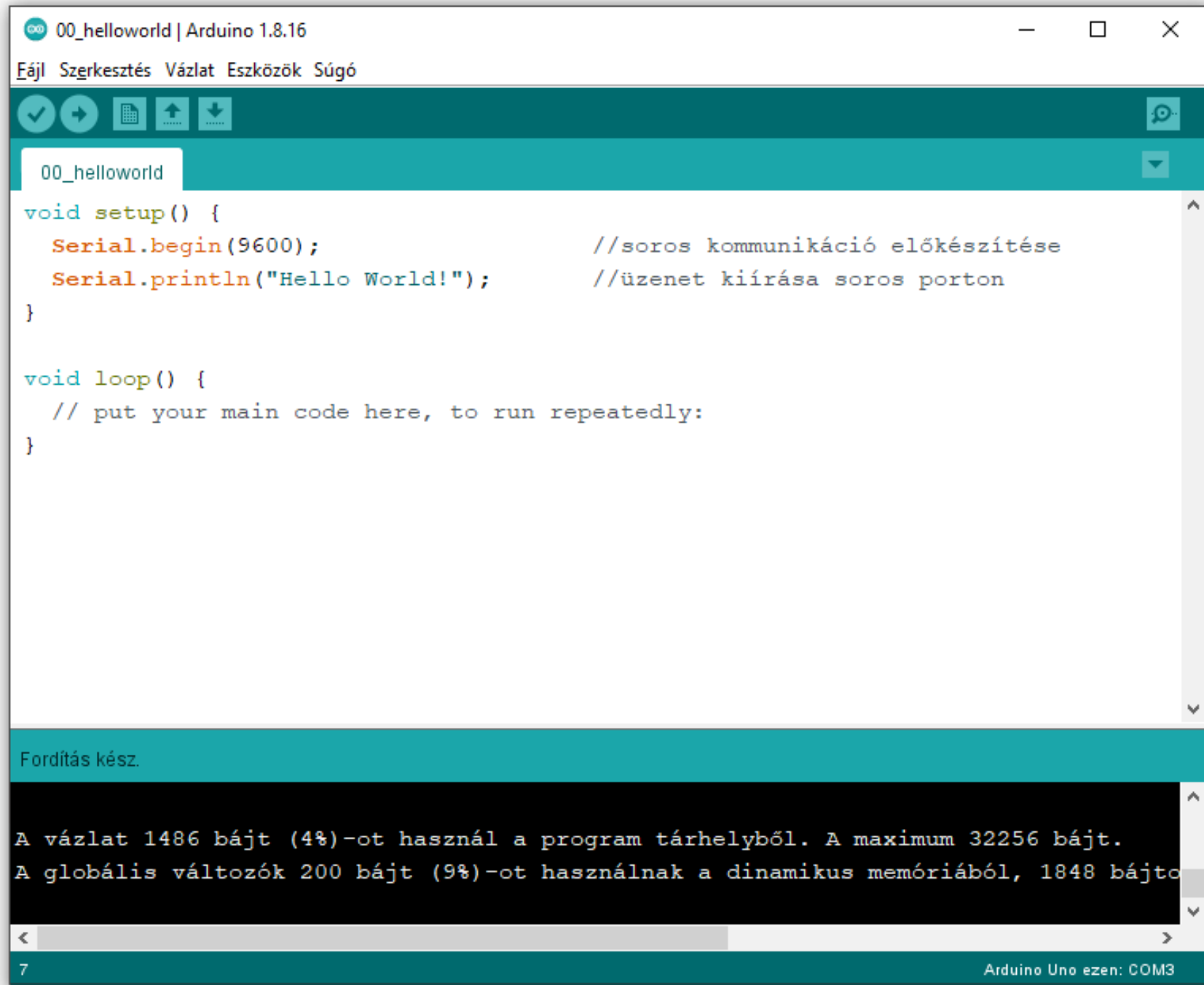
Mentés kész.

7 Arduino Uno ezen: COM3

PROJEKT 1

„HELLO WORLD”

FORDÍTÁS, FELTÖLTÉS



The screenshot shows the Arduino IDE interface. The title bar reads '00_helloworld | Arduino 1.8.16'. The menu bar includes 'Fájl', 'Szerkesztés', 'Vázlat', 'Eszközök', and 'Súgó'. The toolbar contains icons for checking, running, saving, and uploading. The code editor shows the following code:

```
void setup() {  
  Serial.begin(9600);           //soros kommunikáció előkészítése  
  Serial.println("Hello World!"); //üzenet kiírása soros porton  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

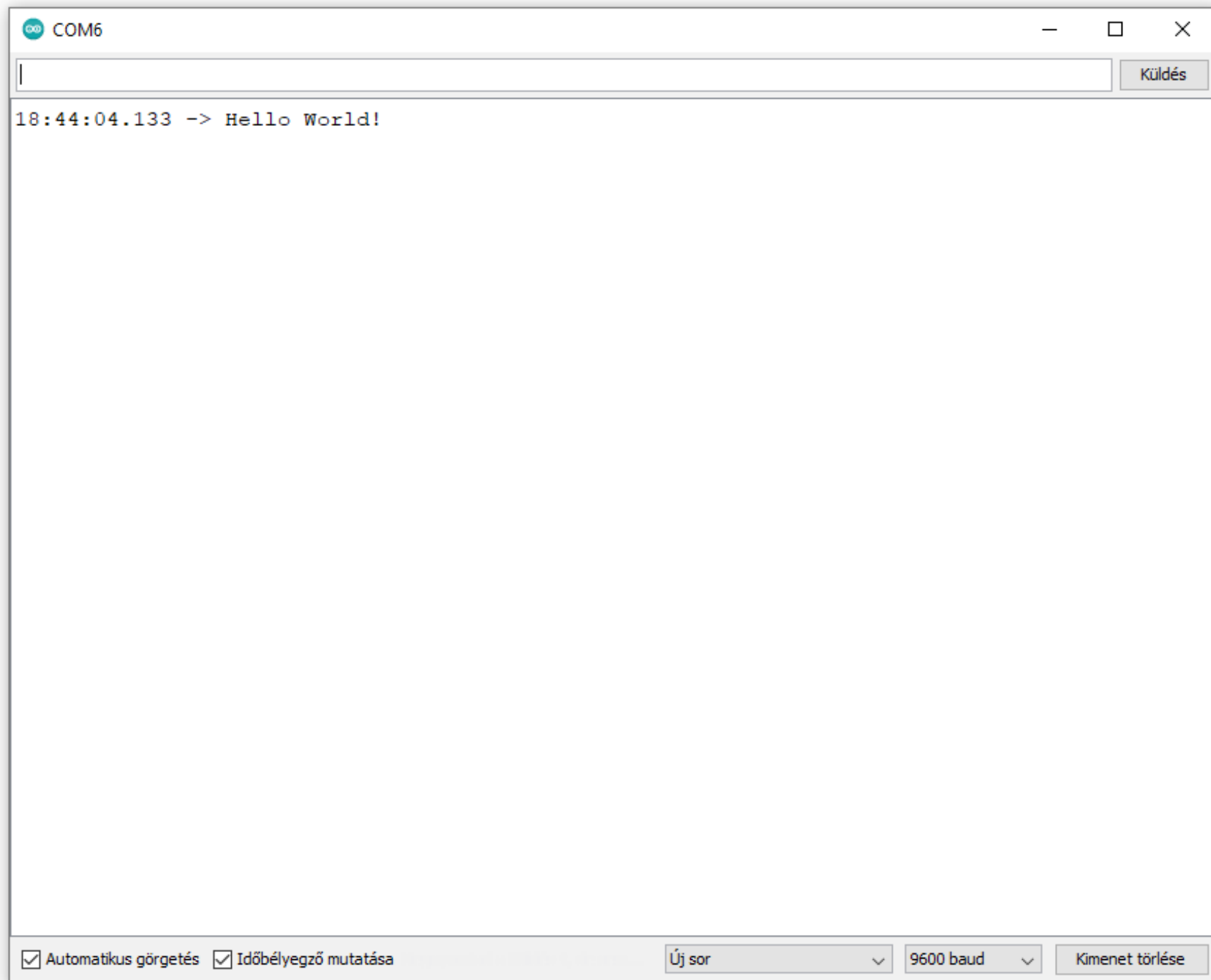
Below the code editor, a status bar indicates 'Fordítás kész.' (Compilation complete). The bottom status bar shows '7' and 'Arduino Uno ezen: COM3'.

A vázlat 1486 bájt (4%)-ot használ a program tárhelyből. A maximum 32256 bájt.
A globális változók 200 bájt (9%)-ot használnak a dinamikus memóriából, 1848 bájt

PROJEKT 1

„HELLO WORLD”

SOROS MONITOR



PROJEKT 1

„HELLO WORLD”

MIT CSINÁLTUNK?

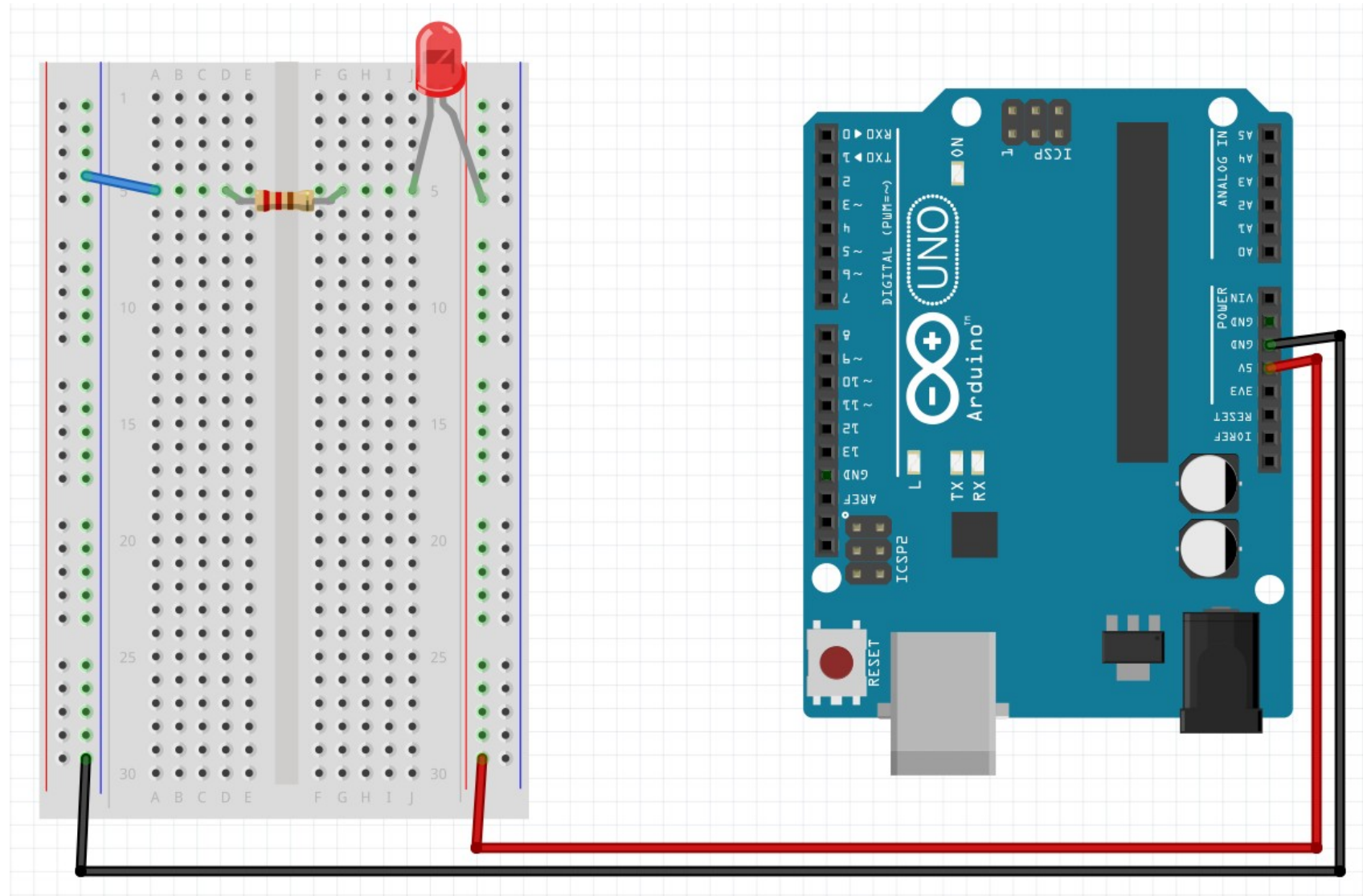
- Program feltöltése
 - Arduino (C++) program fordítása
 - Program feltöltése
- Soros kommunikáció
 - USB
 - Arduino - PC
 - Üzenetküldés

MIVEL CSINÁLTUK?

- Arduino UNO
 - Atmega328
 - 32KB Program memória
 - Atmega16u2
 - Kommunikációs chip
- Arduino IDE
 - Integrált Fejlesztői Környezet
 - Kód szerkesztő
 - Fordító
 - Programozó

PROJEKT 2 „BLINK”

LED BEKÖTÉSE



PROJEKT 2

„BLINK”

MIT CSINÁLTUNK?

- Arduino csatlakoztatása
 - GND
 - 5V
- Elektronikai alkatrészek
 - LED
 - Korlátozó ellenállás

MIVEL CSINÁLTUK?

- Breadboard
 - Vezető sorok
 - Csatlakoztatási pontok
- Arduino UNO

MŰKÖDIK?

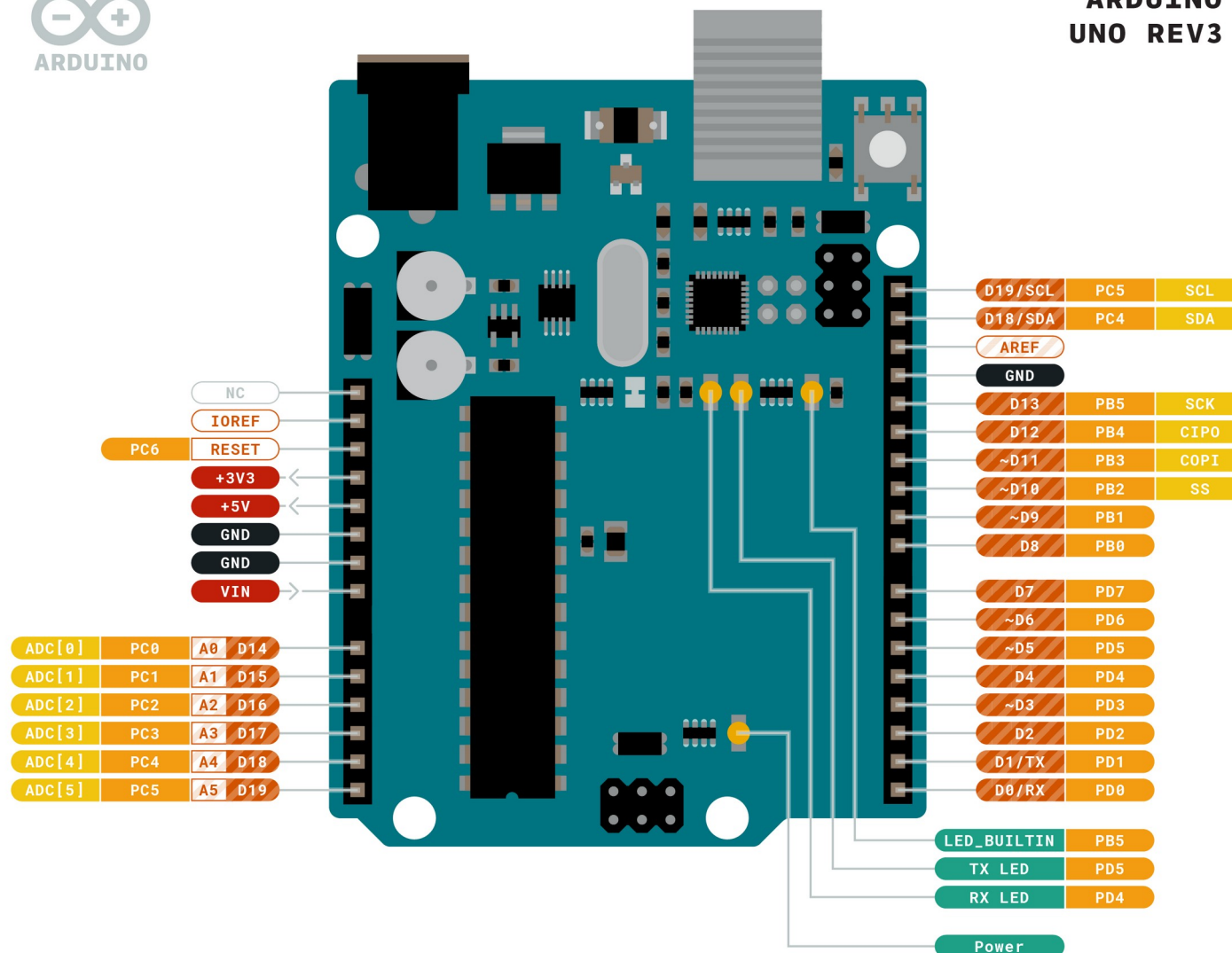
- Világít
- Nem Villog

PROJEKT 2 „BLINK“

ARDUINO UNO



ARDUINO
UNO REV3



Ground	Internal Pin	Digital Pin	Microcontroller's Port
Power	SWD Pin	Analog Pin	
LED	Other Pin	Default	

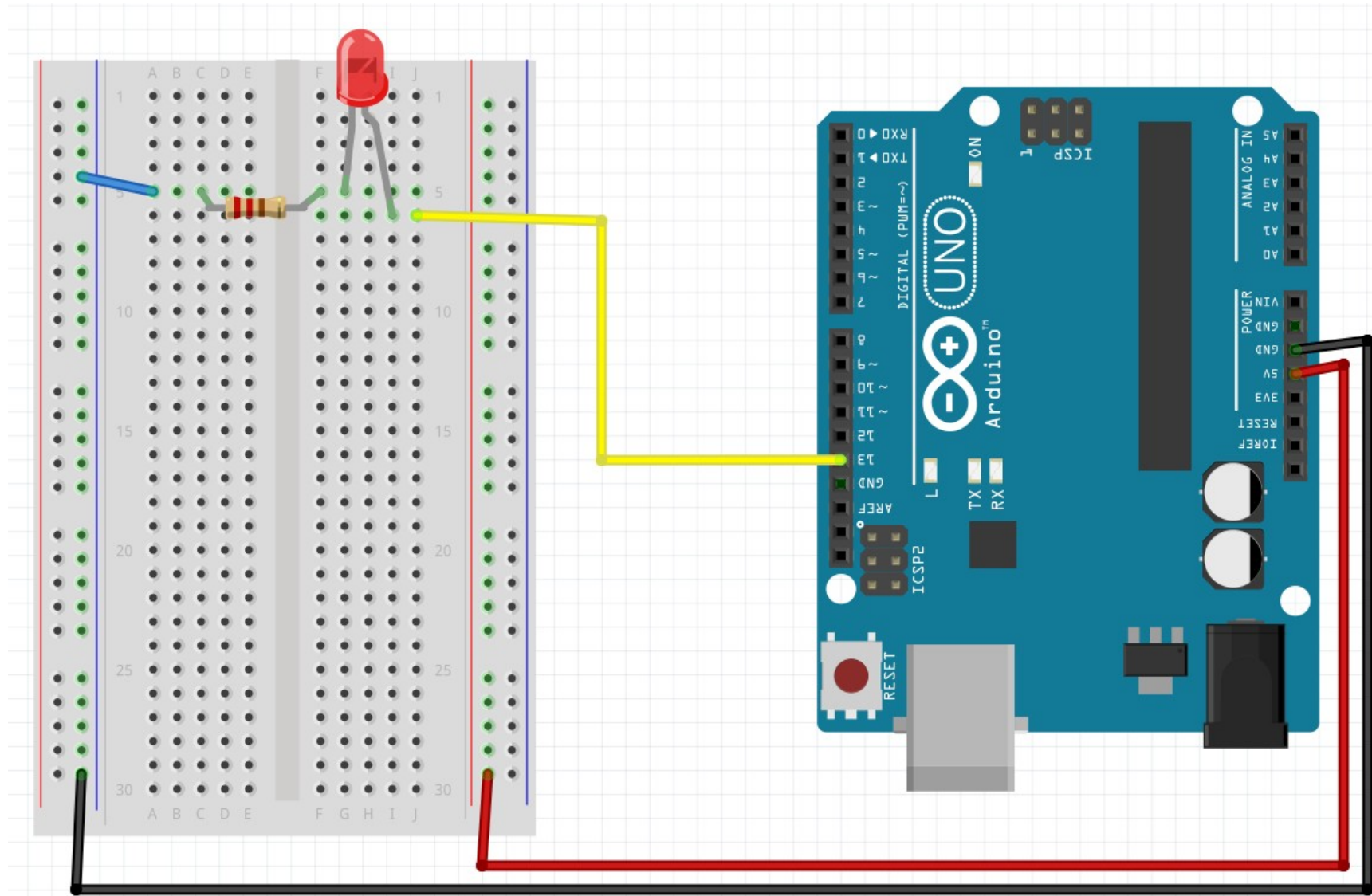
ARDUINO.CC



This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

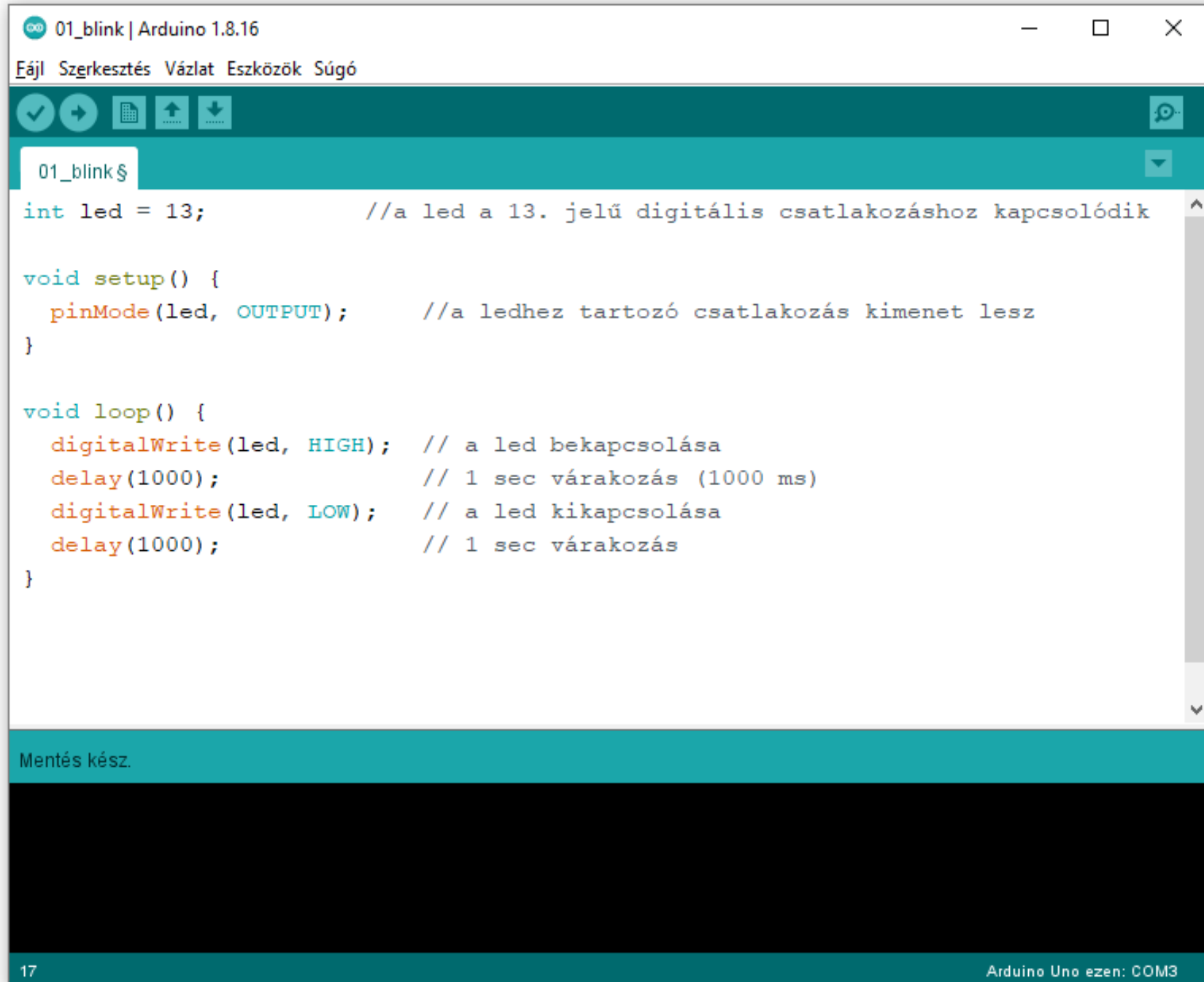
PROJEKT 2 „BLINK”

DIGITÁLIS KIMENET



PROJEKT 2 „BLINK”

KIMENET VEZÉRLÉSE



```
01_blink | Arduino 1.8.16
Fájl Szerkesztés Vázlat Eszközök Súgó

01_blink$

int led = 13;           //a led a 13. jelű digitális csatlakozáshoz kapcsolódik

void setup() {
  pinMode(led, OUTPUT); //a ledhez tartozó csatlakozás kimenet lesz
}

void loop() {
  digitalWrite(led, HIGH); // a led bekapcsolása
  delay(1000);             // 1 sec várakozás (1000 ms)
  digitalWrite(led, LOW);  // a led kikapcsolása
  delay(1000);             // 1 sec várakozás
}

Mentés kész.

17 Arduino Uno ezen: COM3
```


PROJEKT 2 „BLINK”

MIT CSINÁLTUNK?

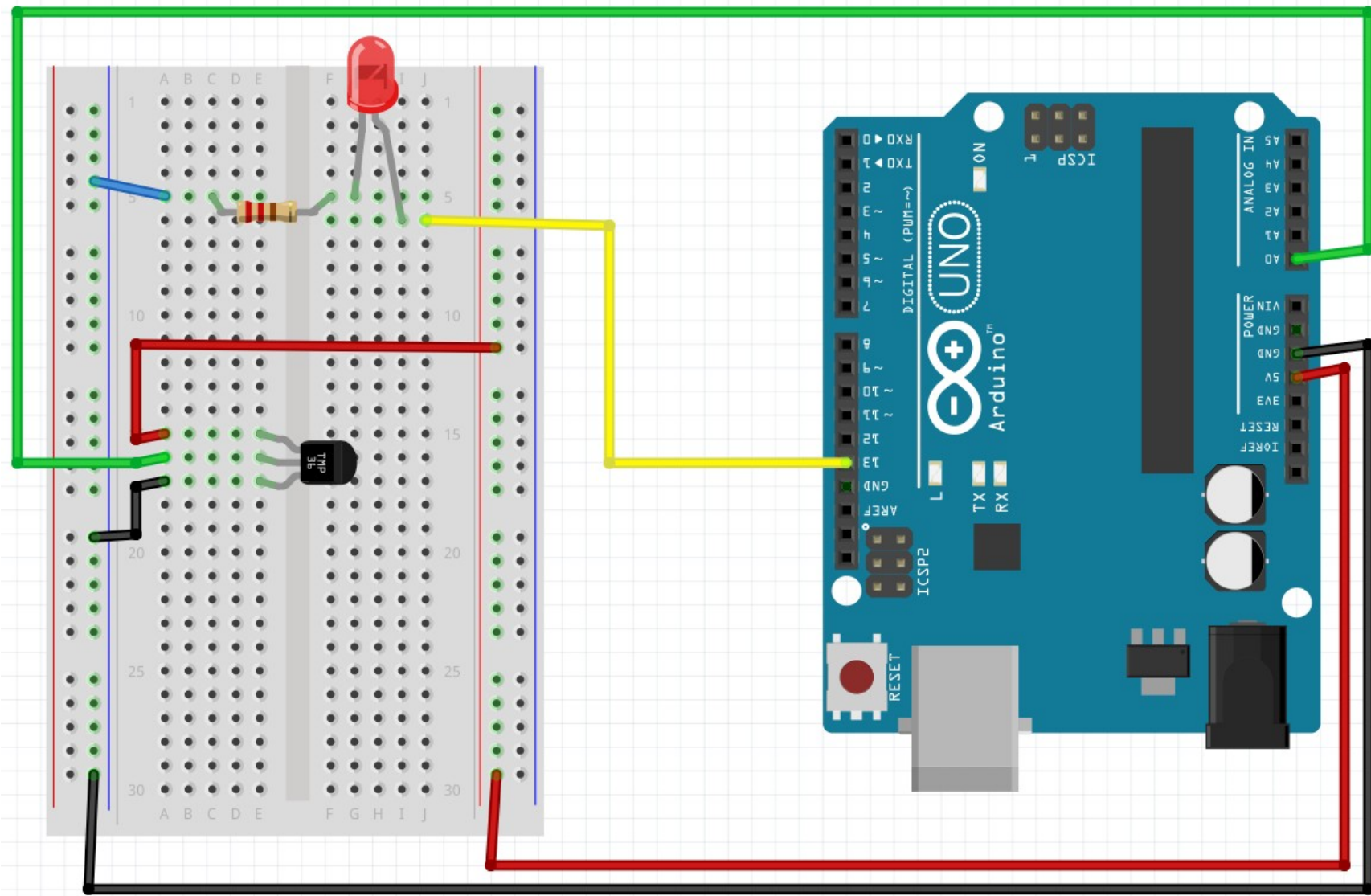
- Elektronika átalakítása
 - LED – Arduino
- Program
 - Digitális láb módjának beállítása
 - Digitális kimenet állapot váltása
 - Késleltetés

MIVEL CSINÁLTUK?

- Arduino UNO
 - Digitális láb
 - Kimenet
 - Korlátozások

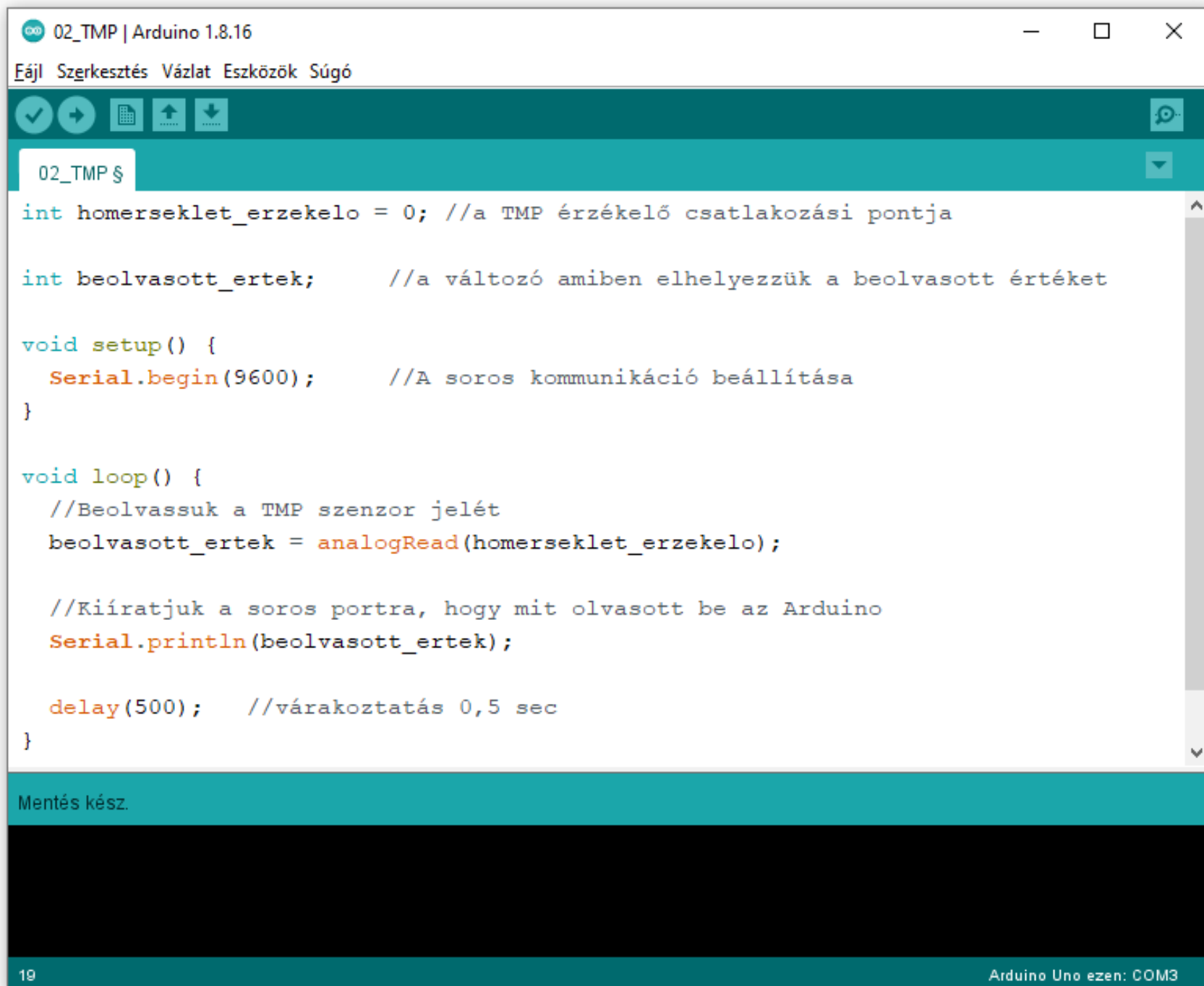
PROJEKT 3 „HŐMÉRŐ”

TMP SZENZOR CSATLAKOZTAT ÁSA



PROJEKT 3 „HŐMÉRŐ”

TMP ÉRTÉKÉNEK OLVASÁSA



```
02_TMP | Arduino 1.8.16
Fájl Szerkesztés Vázlat Eszközök Súgó

02_TMP $

int homerseklet_erzekelo = 0; //a TMP érzékelő csatlakozási pontja

int beolvasott_ertek;          //a változó amiben elhelyezzük a beolvasott értéket

void setup() {
    Serial.begin(9600);        //A soros kommunikáció beállítása
}

void loop() {
    //Beolvassuk a TMP szenzor jelét
    beolvasott_ertek = analogRead(homerseklet_erzekelo);

    //Kiíratjuk a soros portra, hogy mit olvasott be az Arduino
    Serial.println(beolvasott_ertek);

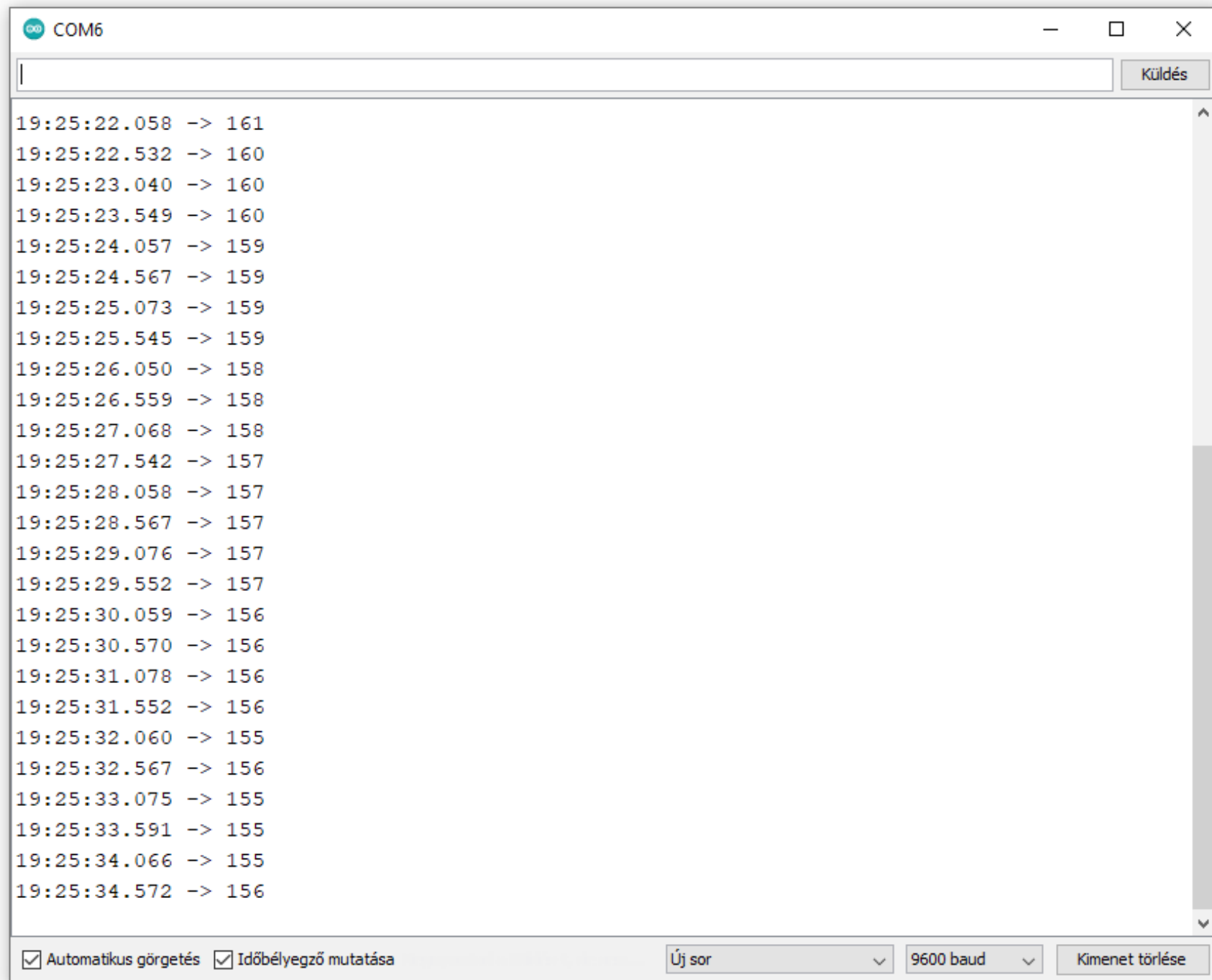
    delay(500);               //várakoztatás 0,5 sec
}

Mentés kész.

19 Arduino Uno ezen: COM3
```

PROJEKT 3 „HŐMÉRŐ”

ÉRTÉKEK MEGJELENÍTÉSE



COM6

Küldés

```
19:25:22.058 -> 161
19:25:22.532 -> 160
19:25:23.040 -> 160
19:25:23.549 -> 160
19:25:24.057 -> 159
19:25:24.567 -> 159
19:25:25.073 -> 159
19:25:25.545 -> 159
19:25:26.050 -> 158
19:25:26.559 -> 158
19:25:27.068 -> 158
19:25:27.542 -> 157
19:25:28.058 -> 157
19:25:28.567 -> 157
19:25:29.076 -> 157
19:25:29.552 -> 157
19:25:30.059 -> 156
19:25:30.570 -> 156
19:25:31.078 -> 156
19:25:31.552 -> 156
19:25:32.060 -> 155
19:25:32.567 -> 156
19:25:33.075 -> 155
19:25:33.591 -> 155
19:25:34.066 -> 155
19:25:34.572 -> 156
```

☒ Automatikusan görgetés ☒ Időbélyegző mutatása

Új sor 9600 baud Kimenet törlése

PROJEKT 3 „HŐMÉRŐ”

MIT CSINÁLTUNK?

- Elektronika átalakítása
 - Analóg hőmérséklet szenzor csatlakoztatása
- Program
 - Változó létrehozása
 - Analóg értékek olvasása
 - Értékek továbbítása
 - Várákoztatás

MIVEL CSINÁLTUK?

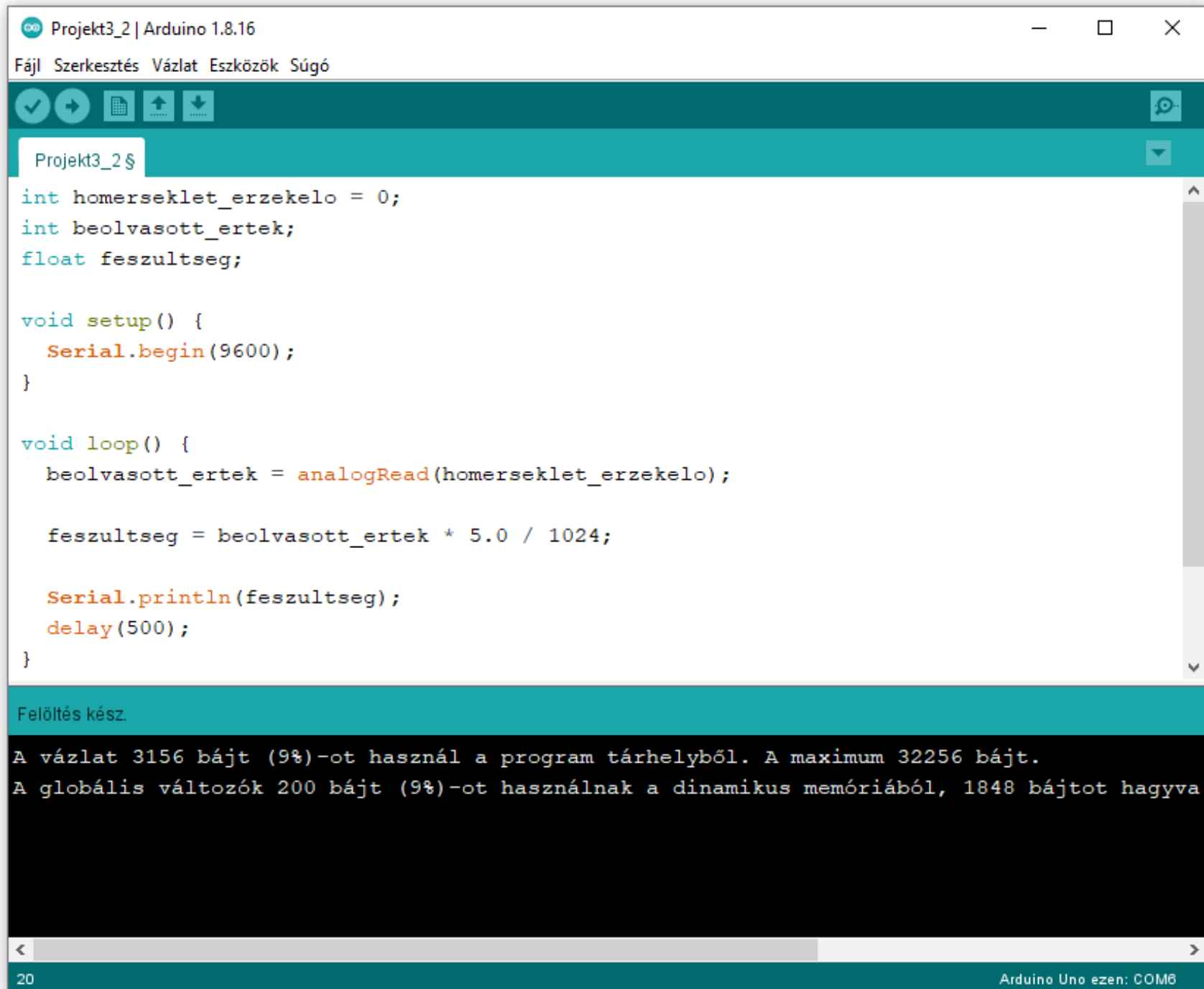
- Arduino UNO
 - Analóg bemenet
 - ADC (10bit , 0 – 1024)
- TMP
 - Analóg hőmérséklet szenzor
 - Pontosság +/- 2 Celsius
 - Tartomány -40 – 125 C

MŰKÖDIK?

- Valótlan értékek?
 - Miért?

PROJEKT 3 „HŐMÉRŐ”

TMP ÉRTÉKEK ÁTALAKÍTÁSA



```
Projekt3_2 | Arduino 1.8.16
Fájl Szerkesztés Vázlat Eszközök Súgó

Projekt3_2 §

int homerseklet_erzekelo = 0;
int beolvasott_ertek;
float feszultseg;

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

void loop() {
  beolvasott_ertek = analogRead(homerseklet_erzekelo);

  feszultseg = beolvasott_ertek * 5.0 / 1024;

  Serial.println(feszultseg);
  delay(500);
}

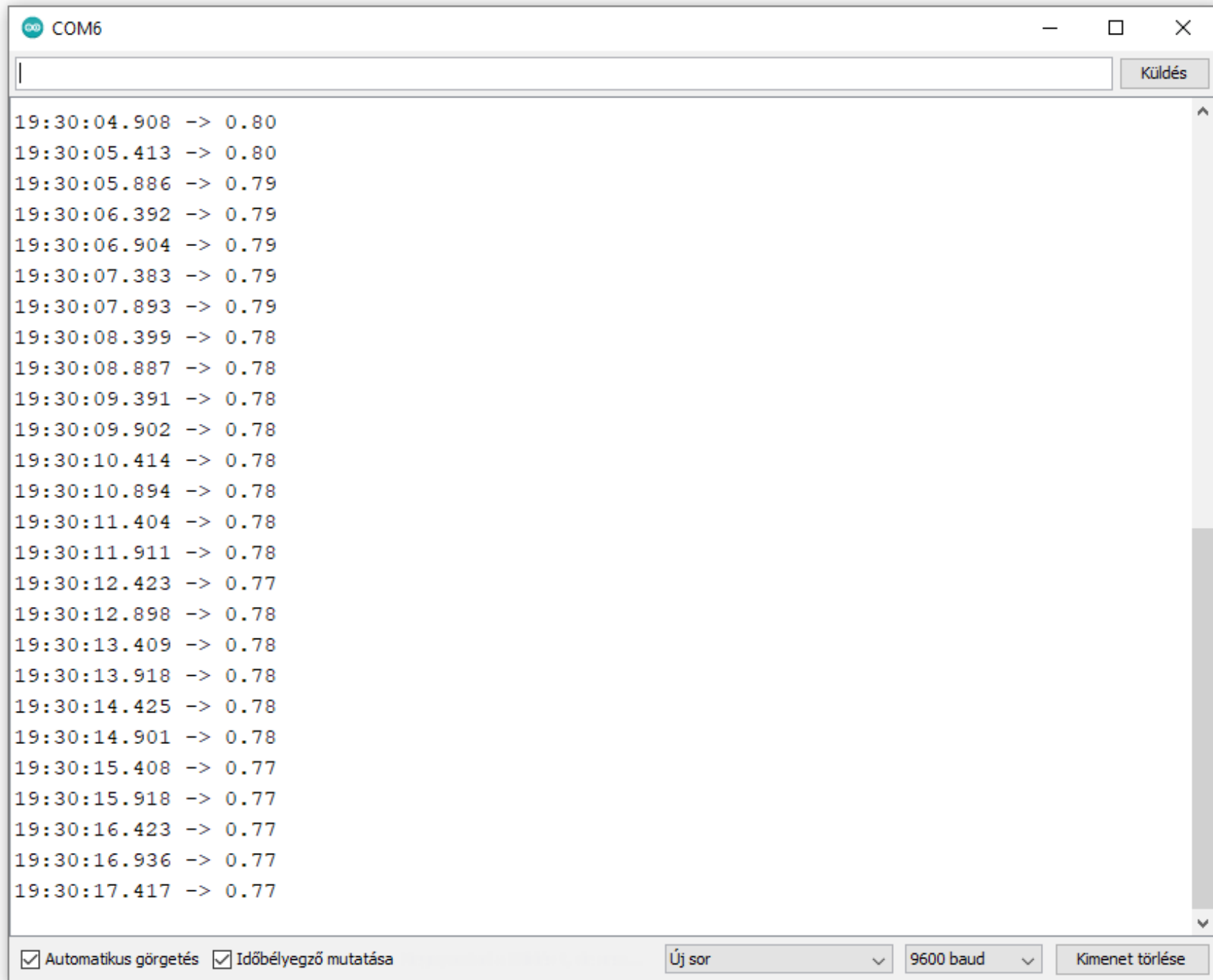
Felöltés kész.

A vázlat 3156 bájt (9%)-ot használ a program tárhelyből. A maximum 32256 bájt.
A globális változók 200 bájt (9%)-ot használnak a dinamikus memóriából, 1848 bájtot hagyva

20 Arduino Uno ezen: COM6
```


PROJEKT 3 „HŐMÉRŐ”

ÉRTÉKEK MEGJELENÍTÉSE



COM6

Küldés

```
19:30:04.908 -> 0.80
19:30:05.413 -> 0.80
19:30:05.886 -> 0.79
19:30:06.392 -> 0.79
19:30:06.904 -> 0.79
19:30:07.383 -> 0.79
19:30:07.893 -> 0.79
19:30:08.399 -> 0.78
19:30:08.887 -> 0.78
19:30:09.391 -> 0.78
19:30:09.902 -> 0.78
19:30:10.414 -> 0.78
19:30:10.894 -> 0.78
19:30:11.404 -> 0.78
19:30:11.911 -> 0.78
19:30:12.423 -> 0.77
19:30:12.898 -> 0.78
19:30:13.409 -> 0.78
19:30:13.918 -> 0.78
19:30:14.425 -> 0.78
19:30:14.901 -> 0.78
19:30:15.408 -> 0.77
19:30:15.918 -> 0.77
19:30:16.423 -> 0.77
19:30:16.936 -> 0.77
19:30:17.417 -> 0.77
```

☒ Automatikus görgetés ☒ Időbélyegző mutatása

Új sor 9600 baud Kimenet törlése

PROJEKT 3 „HŐMÉRŐ”

ÉRTÉKEK HŐMÉRSÉKLET RE ÁLAKÍTÁSA

04_TMP

```
int homerseklet_erzekelo = 0; //a TMP érzékelő csatlakozási pontja

int beolvasott_ertek;          //a változó amiben elhelyezzük a beolvasott értéket
float feszultseg;              //beolvasott érték tárolása feszültségre átszámolva
float homerseklet_C;

void setup() {
    Serial.begin(9600);        //A soros kommunikáció beállítása
}

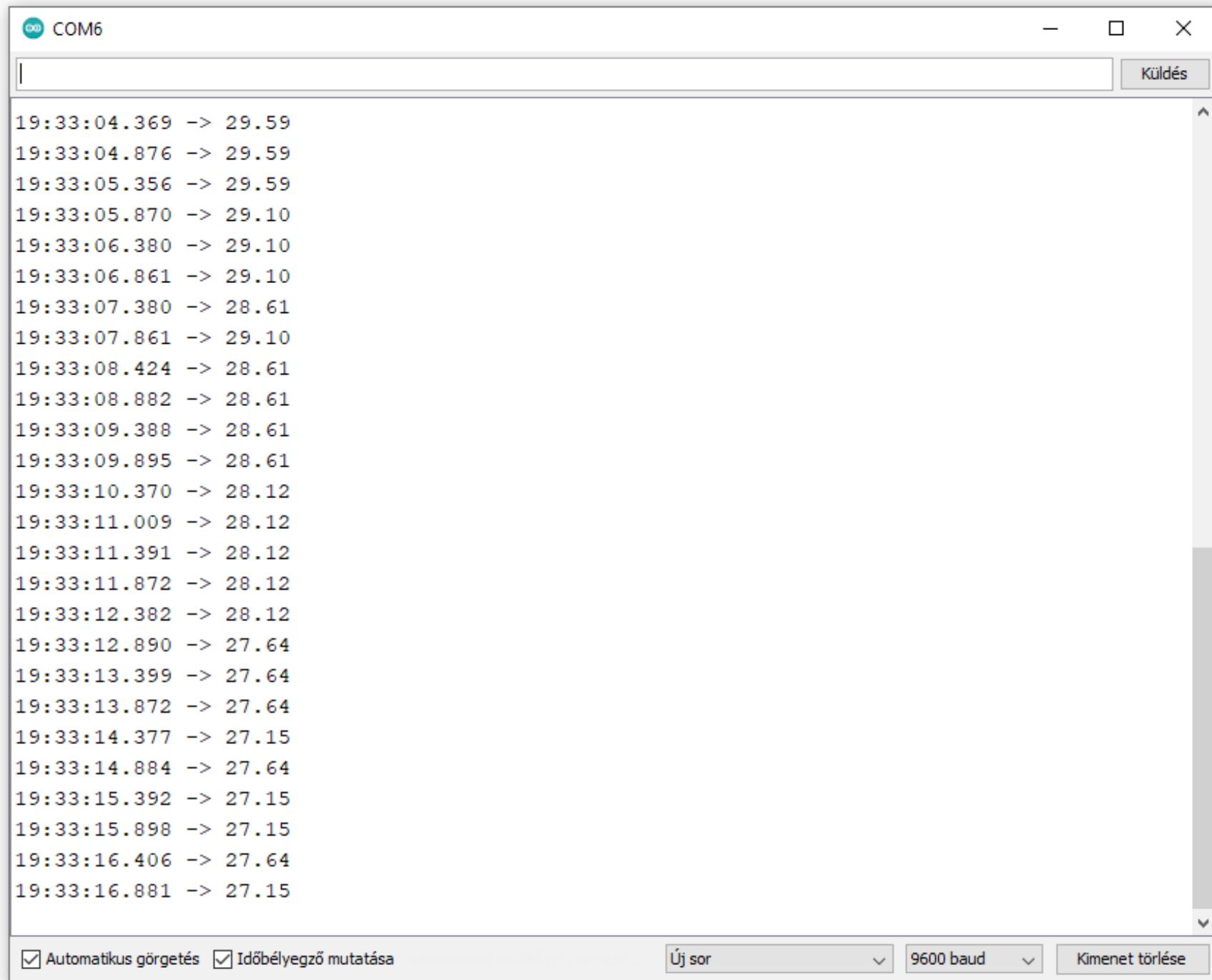
void loop() {
    //Beolvassuk a TMP szenzor jelét
    beolvasott_ertek = analogRead(homerseklet_erzekelo);

    feszultseg = beolvasott_ertek * 5.0 / 1024; //beolvasott érték átszámítása
    homerseklet_C = (feszultseg - 0.5) * 100;   //homerseklet kiszámítása

    //Kiíratjuk a soros portra, a kiszámított hőmérséklet értéket
    Serial.println(homerseklet_C);
    delay(500);               //várakoztatás 0,5 sec
}
```

PROJEKT 3 „HŐMÉRŐ”

HŐMÉRSÉKLET MEGJELENÍTÉSE



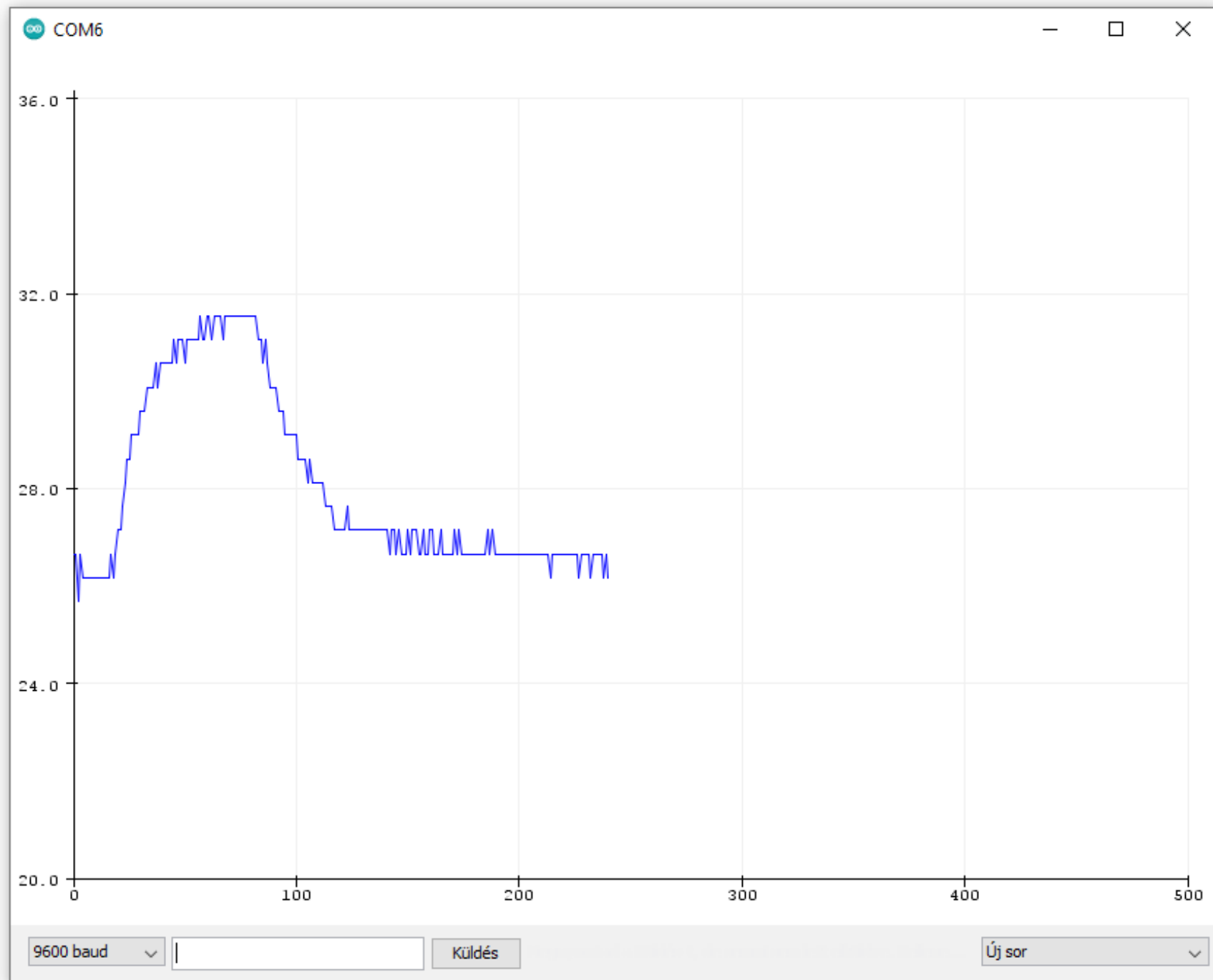
The screenshot shows a serial monitor window titled "COM6". It displays a list of timestamped temperature readings. Each line consists of a timestamp in HH:MM:SS.mmm format, followed by an arrow pointing to the temperature value in degrees Celsius. The data shows a fluctuating temperature that starts at 29.59, drops to 28.61, and then continues to fluctuate between 27.15 and 28.12.

Timestamp	Temperature (°C)
19:33:04.369	29.59
19:33:04.876	29.59
19:33:05.356	29.59
19:33:05.870	29.10
19:33:06.380	29.10
19:33:06.861	29.10
19:33:07.380	28.61
19:33:07.861	29.10
19:33:08.424	28.61
19:33:08.882	28.61
19:33:09.388	28.61
19:33:09.895	28.61
19:33:10.370	28.12
19:33:11.009	28.12
19:33:11.391	28.12
19:33:11.872	28.12
19:33:12.382	28.12
19:33:12.890	27.64
19:33:13.399	27.64
19:33:13.872	27.64
19:33:14.377	27.15
19:33:14.884	27.64
19:33:15.392	27.15
19:33:15.898	27.15
19:33:16.406	27.64
19:33:16.881	27.15

At the bottom of the window, there are several controls: two checked checkboxes for "Automatikus görgetés" and "Időbélyegző mutatása", a dropdown menu set to "Új sor", another dropdown menu set to "9600 baud", and a button labeled "Kimenet törlése".

PROJEKT 3 „HŐMÉRŐ”

HŐMÉRSÉKLET GRAFIKONON



PROJEKT 3 „HŐMÉRŐ”

MIT CSINÁLTUNK?

- Program
 - Feszültség kiszámítása
 - Hőmérséklet kiszámítása

MŰKÖDIK?!

MIVEL CSINÁLTUK?

- Arduino Környezet
 - Műveletek
 - Lebegőpontos számok
- Arduino IDE
 - Soros adat grafikonon
- TMP
 - Alkatrész adatlap
 - Működési paraméterek

PROJEKT 4 „HŐMÉRŐ”

HŐMÉRSÉKLET ALAPÚ SZABÁLYZÁS

06_TMP

```
int homerseklet_erzekelo = 0;
int led = 13;

int beolvasott_ertek;
float feszultseg;
float homerseklet_C;

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

void loop() {
    beolvasott_ertek = analogRead(homerseklet_erzekelo);
    feszultseg = beolvasott_ertek * 5.0 / 1024;
    homerseklet_C = (feszultseg - 0.5) * 100;

    Serial.println(homerseklet_C);

    if(homerseklet_C > 21){
        digitalWrite(led, HIGH);
    }else{
        digitalWrite(led, LOW);
    }

    delay(500);
}
```


PROJEKT 4 „HŐMÉRŐ”

HŐMÉRSÉKLET ALAPÚ SZABÁLYZÁS

MIT CSINÁLTUNK?

- Program
 - Feltétel alapú LED vezérlés

MIVEL CSINÁLTUK?

- Arduino Környezet
 - Digitális Kimenet
 - Analóg bemenet
 - Várakoztatás
 - Műveletek
 - Elágazás
 - Összehasonlítás, Logikai értékek

MŰKÖDIK?

- Határérték közeli bizonytalanság
 - Miért?

PROJEKT 4 „HŐMÉRŐ”

FELADAT

PROGRAM FELADATA

- Bizonytalanság kiküszöbölése

HOGYAN?

SZÜKSÉGES ELEMEEK

- Változók használata
- Analóg bemenet olvasása
- Digitális láb írása
- Műveletek
- Döntés feltétel alapján
- Várakozás

JÓ MUNKÁT!

PROJEKT 4 „HŐMÉRŐ”

FELADAT

PROGRAM FELADATA

- Bizonytalanság kiküszöbölése

HOGYAN?

- Bizonytalanság kiküszöbölése
 - Ki és Be kapcsolási érték szétválasztása
 - 21.5 C
 - 20.5 C

SZÜKSÉGES ELEMEEK

- Változók használata
- Analóg bemenet olvasása
- Digitális láb írása
- Műveletek
- Döntés feltétel alapján
- Várakozás

JÓ MUNKÁT!

PROJEKT 4 „HŐMÉRŐ”

FELADAT

07_TMP

```
int homerseklet_erzekelo = 0;
int led = 13;

int beolvasott_ertekek;
float feszultseg;
float homerseklet_C;

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

void loop() {
    beolvasott_ertekek = analogRead(homerseklet_erzekelo);
    feszultseg = beolvasott_ertekek * 5.0 / 1024;
    homerseklet_C = (feszultseg - 0.5) * 100;

    Serial.println(homerseklet_C);

    if(homerseklet_C > 21.5){
        digitalWrite(led, HIGH);
    }
    if(homerseklet_C < 20.5){
        digitalWrite(led, LOW);
    }

    delay(500);
}
```

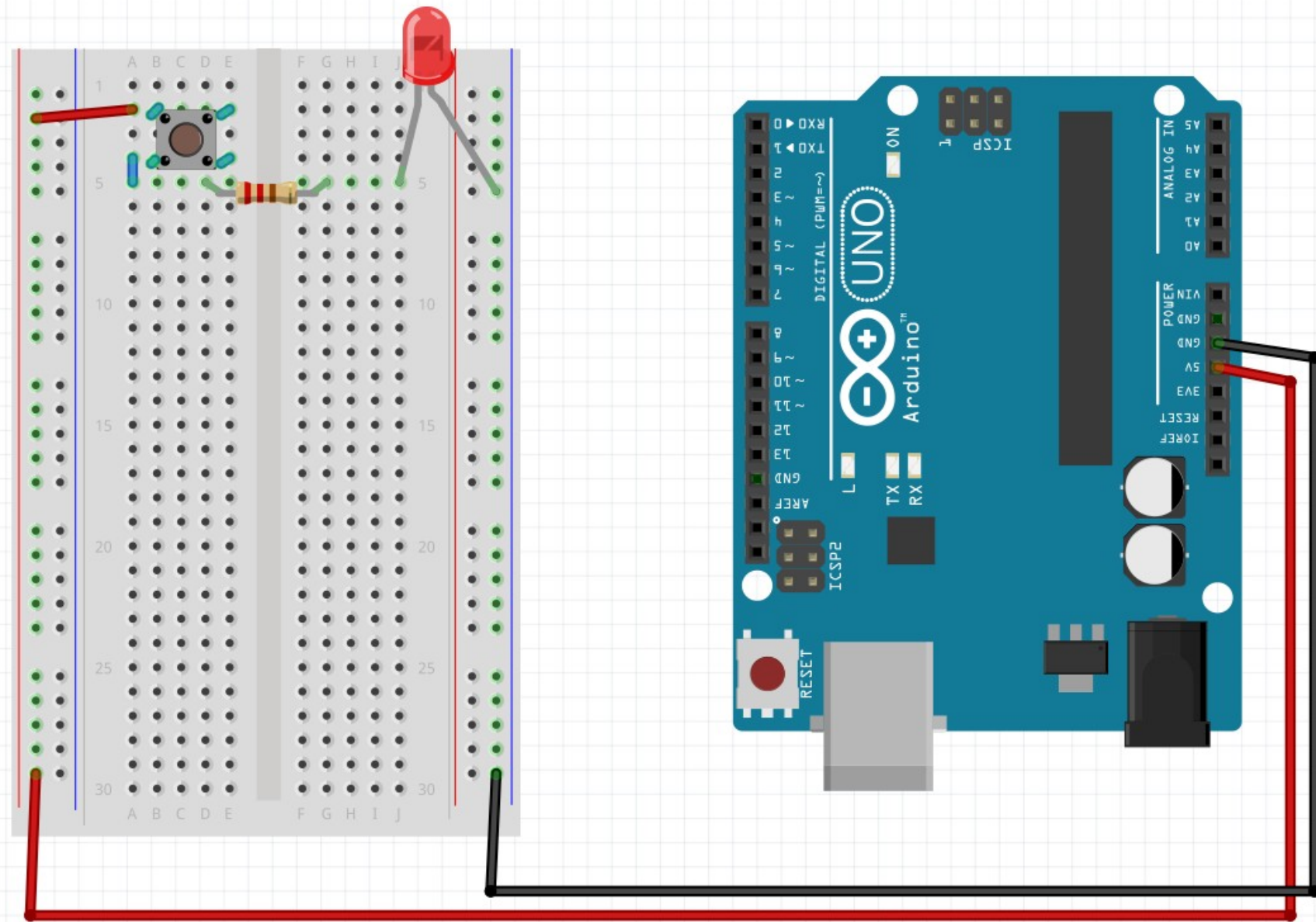
SIKERÜLT?

TAPASZTALATOK?

PROJEKT 5

„NYOMÓGOMB”

NYOMÓGOMB BEKÖTÉSE



PROJEKT 5 „NYOMÓGOMB”

MIT CSINÁLTUNK?

- Arduino csatlakoztatása
 - Nyomógomb
- Elektronikai alkatrészek
 - LED
 - Korlátozó ellenállás
 - Nyomógomb

MIVEL CSINÁLTUK?

- Breadboard, Elektronika
 - Nyomógomb bekötése

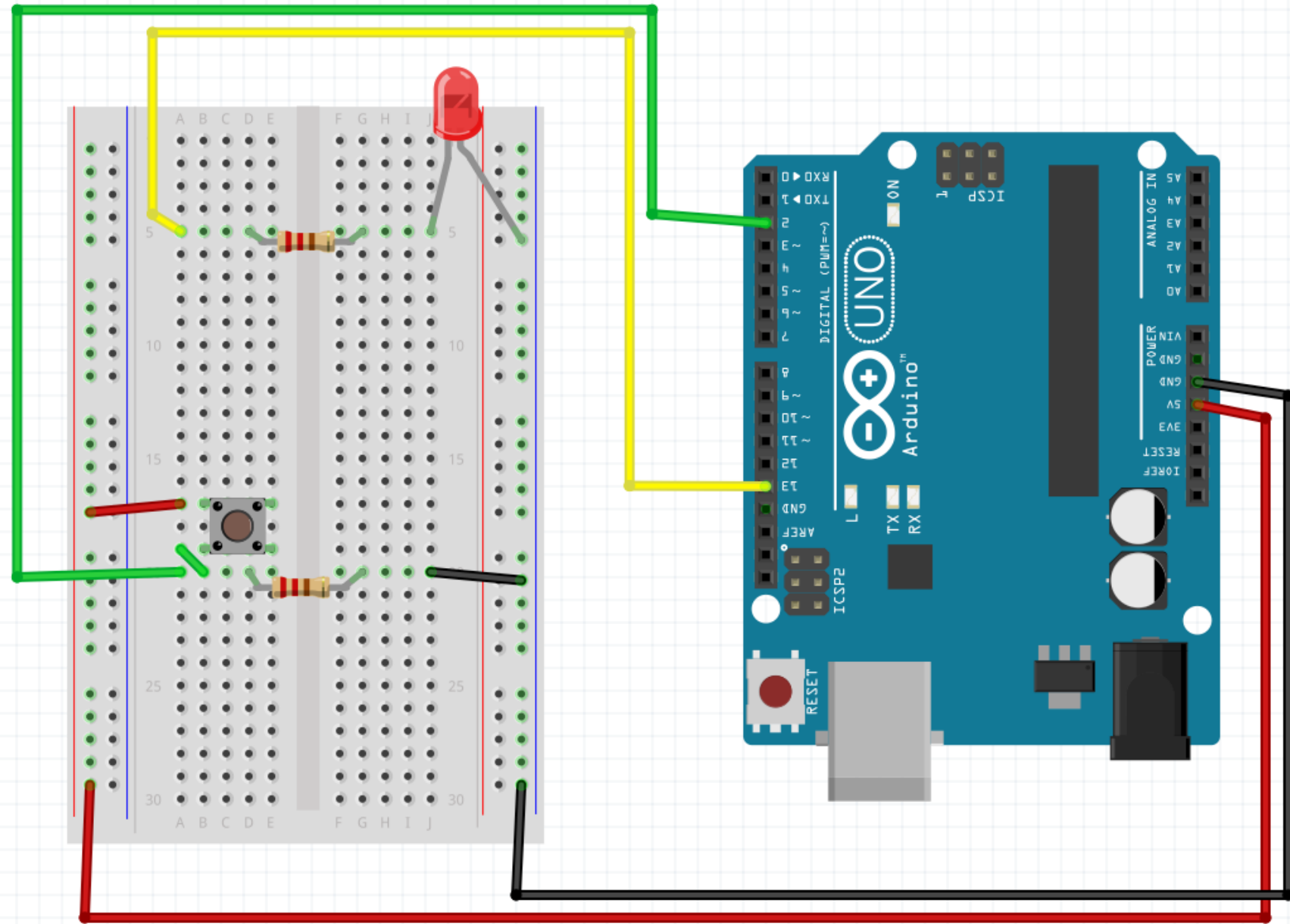
MŰKÖDIK?

- Gombnyomásra világít
- Arduino nem detektálja

PROJEKT 5

„NYOMÓGOMB”

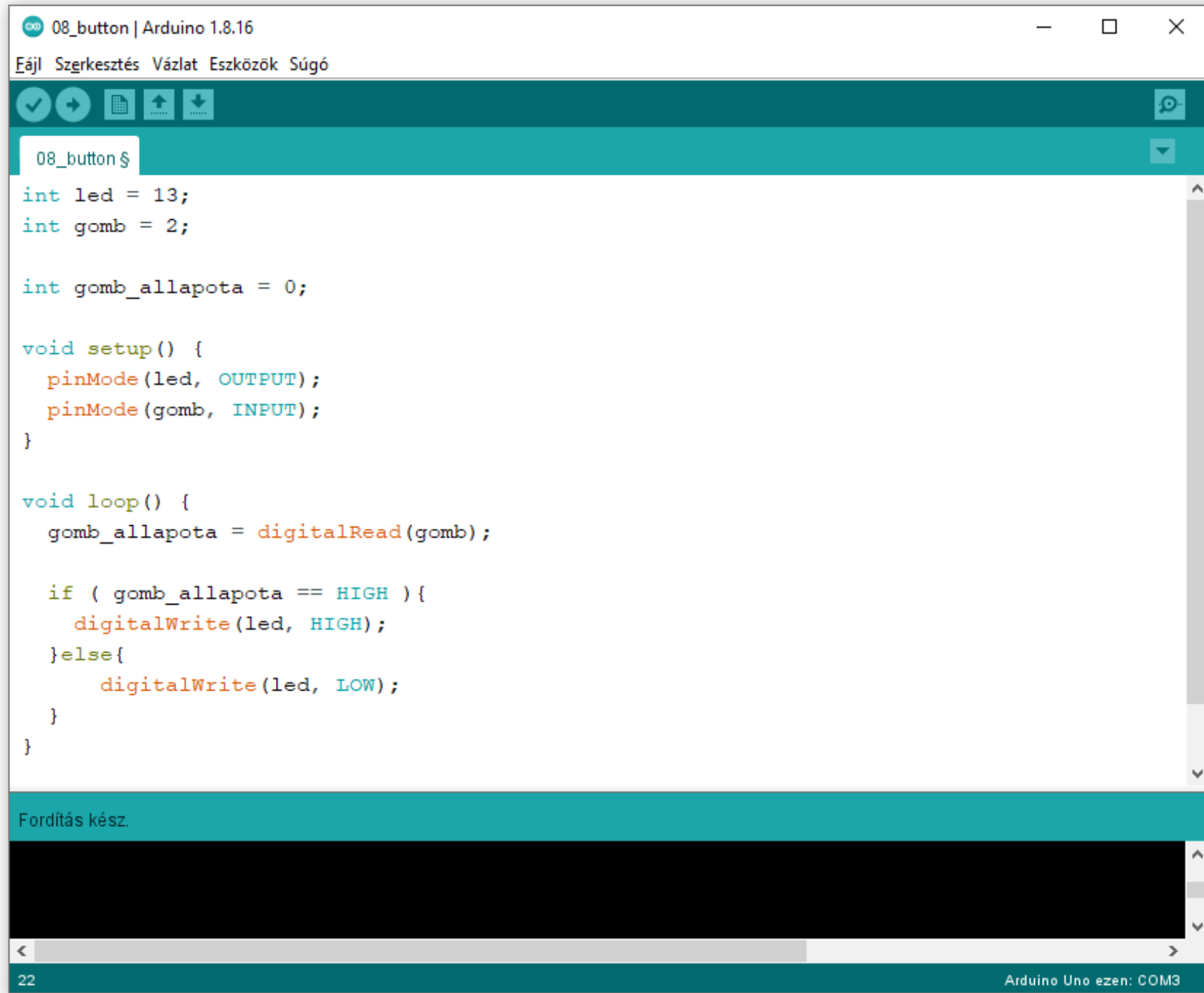
NYOMÓGOMB BEKÖTÉSE



PROJEKT 5

„NYOMÓGOMB”

NYOMÓGOMB VEZÉRLÉSE



The screenshot shows the Arduino IDE interface with the following elements:

- Title Bar:** 08_button | Arduino 1.8.16
- Menu Bar:** Fájl Szerkesztés Vázlat Eszközök Súgó
- Toolbar:** Includes icons for saving, running, uploading, and downloading.
- Code Editor:** Contains the following C++ code:

```
08_button $  
  
int led = 13;  
int gomb = 2;  
  
int gomb_allapota = 0;  
  
void setup() {  
  pinMode(led, OUTPUT);  
  pinMode(gomb, INPUT);  
}  
  
void loop() {  
  gomb_allapota = digitalRead(gomb);  
  
  if ( gomb_allapota == HIGH ) {  
    digitalWrite(led, HIGH);  
  } else {  
    digitalWrite(led, LOW);  
  }  
}
```
- Status Bar:** Fordítás kész. (Compilation complete.)
- Bottom Bar:** 22 Arduino Uno ezen: COM3

PROJEKT 5

„NYOMÓGOMB”

MIT CSINÁLTUNK?

- Arduino Program
 - Digitális bemenet olvasása

MIVEL CSINÁLTUK?

- Breadboard, Elektronika
 - Nyomógomb bekötése
 - Lehúzó ellenállás
- Arduino UNO
 - Digitális bemenet

MŰKÖDIK!

- Gombnyomásra világít
- Arduino vezérli

PROJEKT 5

„NYOMOGOMB” FELADAT

PROGRAM FELADATA

- Gomb állapotának bekérése
- LED villogtatása
- Feltételes működés
 - Lenyomott gomb esetén a villanások között 0.1 másodperc várakozás
 - Felengedett gomb esetén a villanások között 0.5 másodperc várakozás

SZÜKSÉGES ELEMÉK

Szükséges elemek

- Változók használata
- Digitális láb olvasása
- Digitális láb írása
- Döntés feltétel alapján
- Várakozás

JÓ MUNKÁT!

PROJEKT 5

„NYOMÓGOMB” FELADAT

09_button

```
int led = 13;
int gomb = 2;

int gomb_allapota = 0;

int varakozas;

void setup() {
    pinMode(led, OUTPUT);
    pinMode(gomb, INPUT);
}

void loop() {
    gomb_allapota = digitalRead(gomb);

    if ( gomb_allapota == HIGH ){
        varakozas = 100;
    }else{
        varakozas = 500;
    }

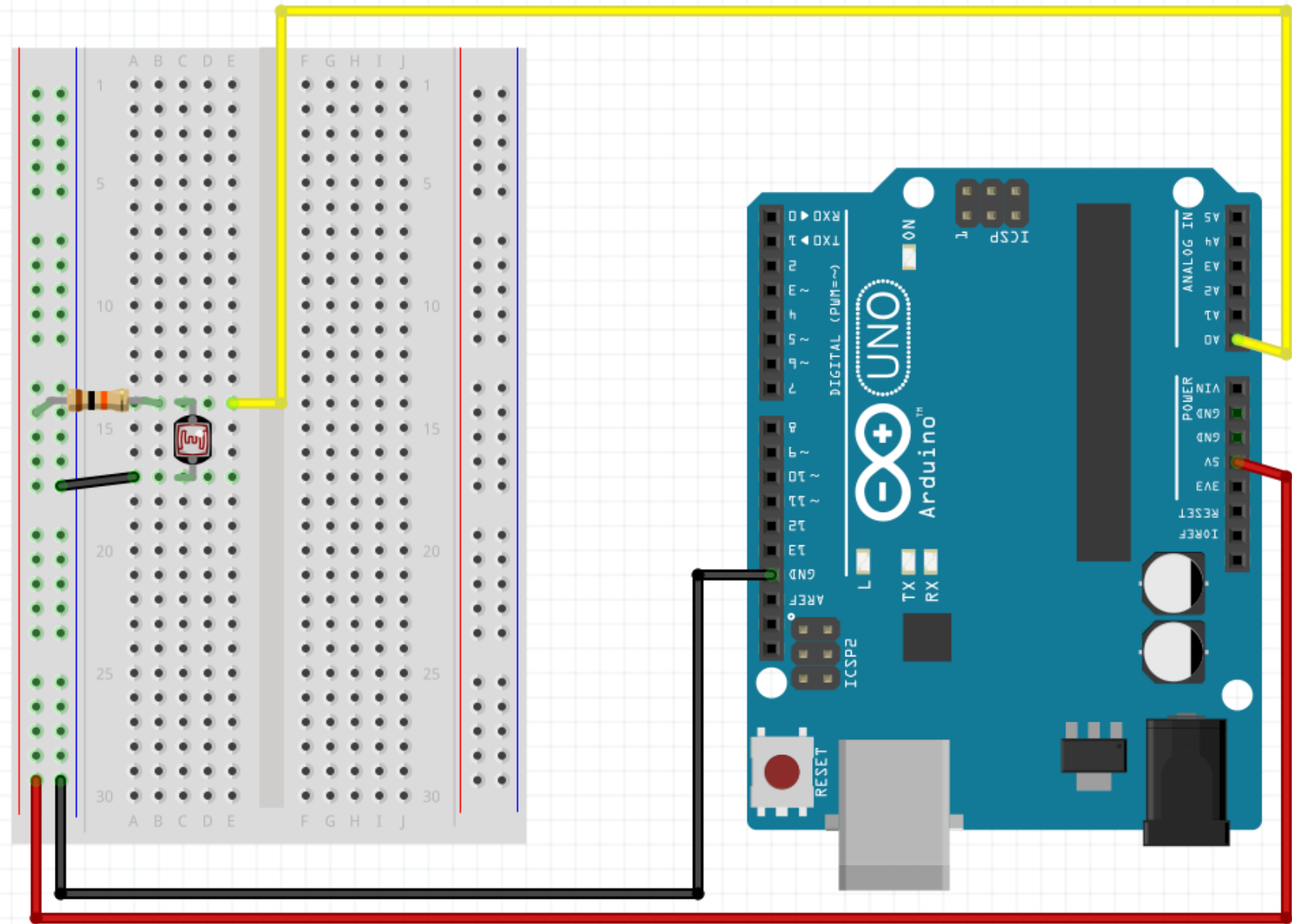
    digitalWrite(led, HIGH);
    delay(varakozas);
    digitalWrite(led, LOW);
    delay(varakozas);
}
```

SIKERÜLT?

TAPASZTALATOK?

PROJEKT 6

„ZAJ SZŰRÉS” LDR BEKÖTÉSE



PROJEKT 6

„ZAJ SZŰRÉS” BEMENET VEZÉRLÉSE

10_idr

```
int feny_erzekelo = 0;

int beolvasott_ertek;
float feszultseg;

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

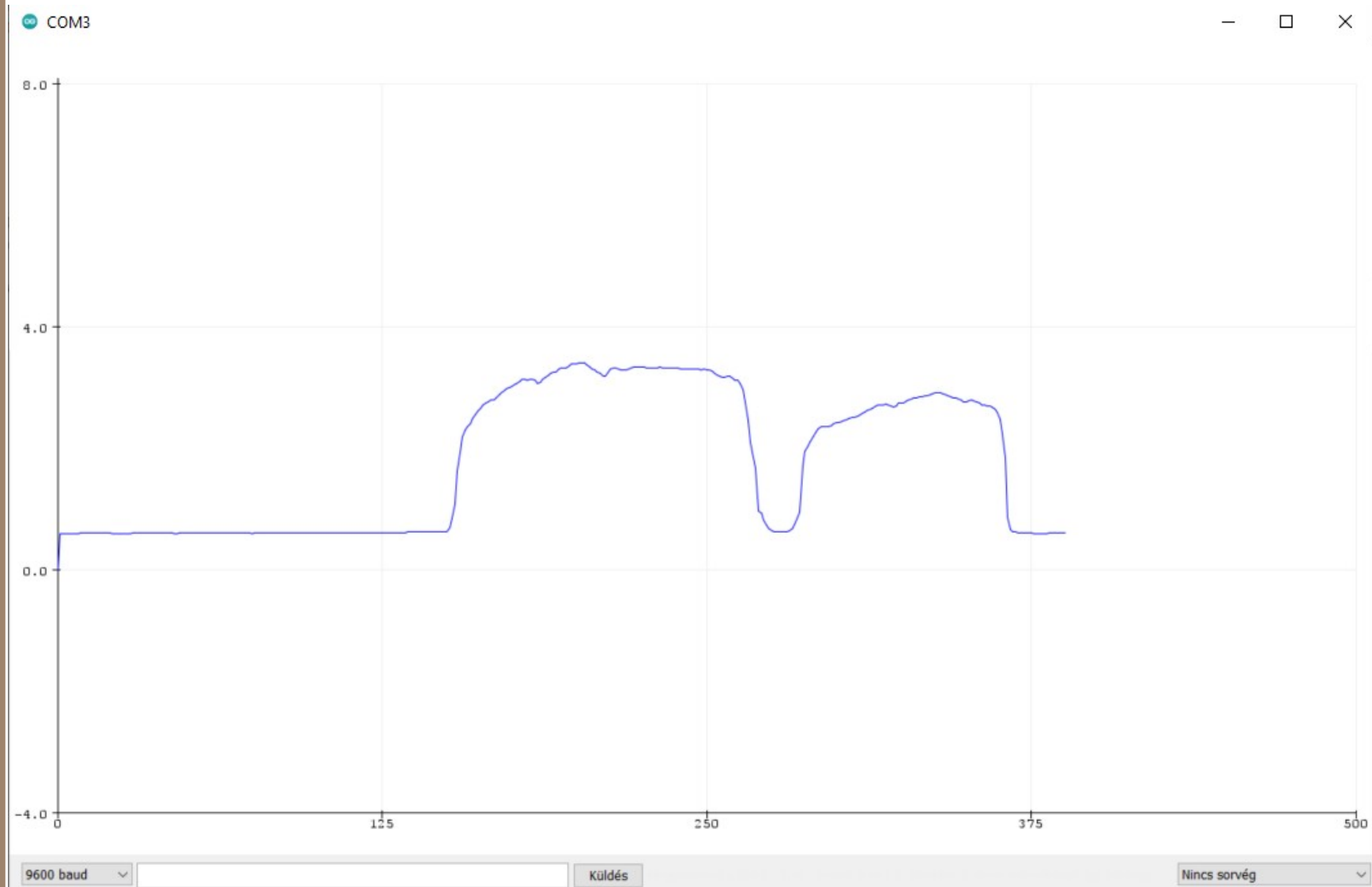
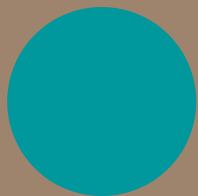
void loop() {
    beolvasott_ertek = analogRead(feny_erzekelo);
    feszultseg = beolvasott_ertek * 5.0 / 1024;

    Serial.println(feszultseg);

    delay(50);
}
```

PROJEKT 6

„ZAJ SZŰRÉS” MEGJELENÍTÉS



PROJEKT 6

„ZAJ SZŰRÉS”

MIT CSINÁLTUNK?

- Arduino Program
 - Analóg bemenet olvasása
 - Átszámolás feszültségre
- Arduino IDE
 - Grafikonos megjelenítés

MIVEL CSINÁLTUK?

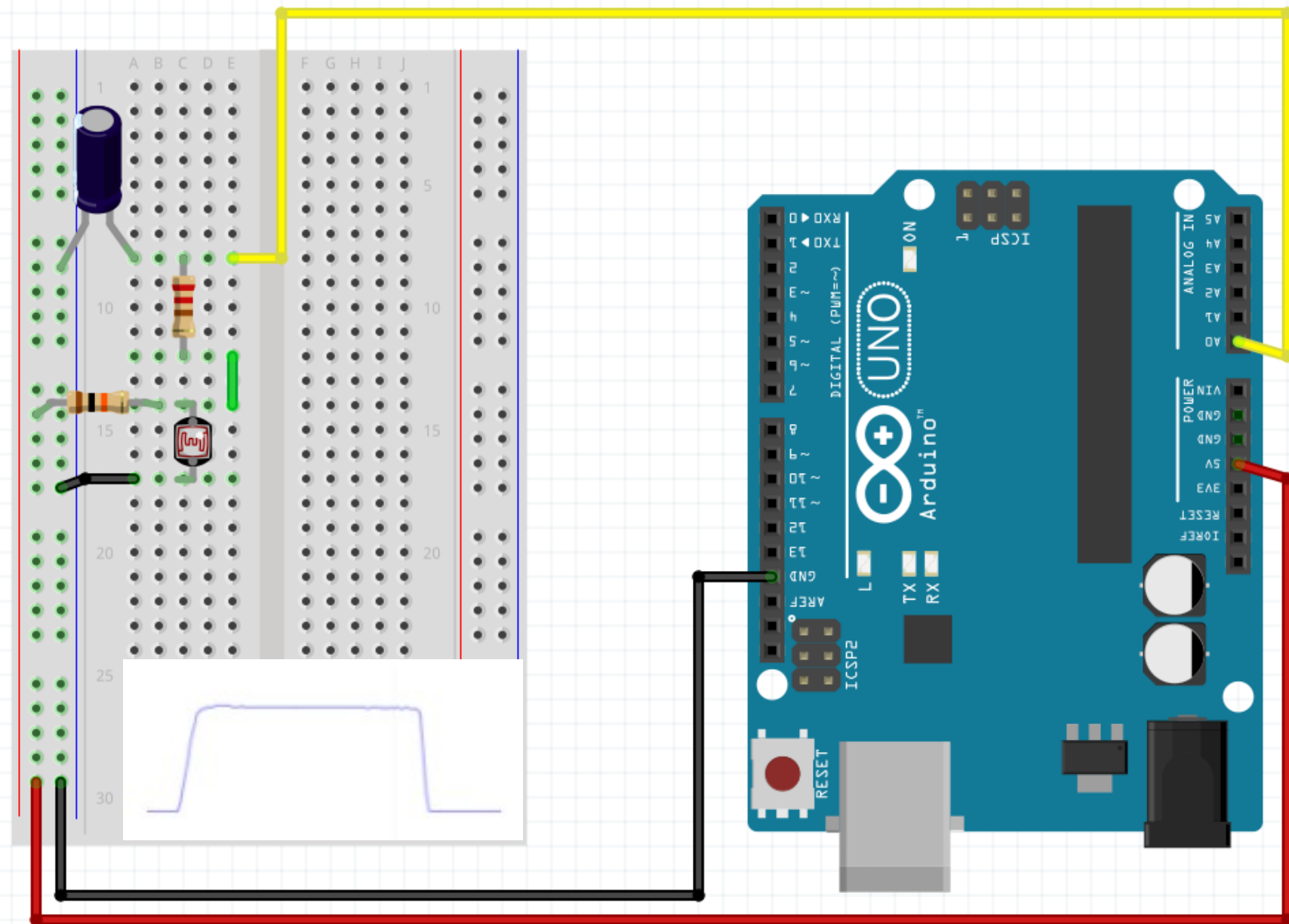
- Breadboard, Elektronika
 - LDR
 - Feszültség osztó
- Arduino UNO
 - Analóg bemenet

MŰKÖDIK?

- Sikeres értékbeolvasás
- Zaj, Zavar

PROJEKT 6

„ZAJ SZŰRÉS” SZŰRŐ ÁRAMKÖR ÉPÍTÉSE



PROJEKT 6

„ZAJ SZŰRÉS”

MIT CSINÁLTUNK?

- Arduino Program
 - Analóg bemenet olvasása
 - Átszámolás feszültségre
- Arduino IDE
 - Grafikonos megjelenítés

MIVEL CSINÁLTUK?

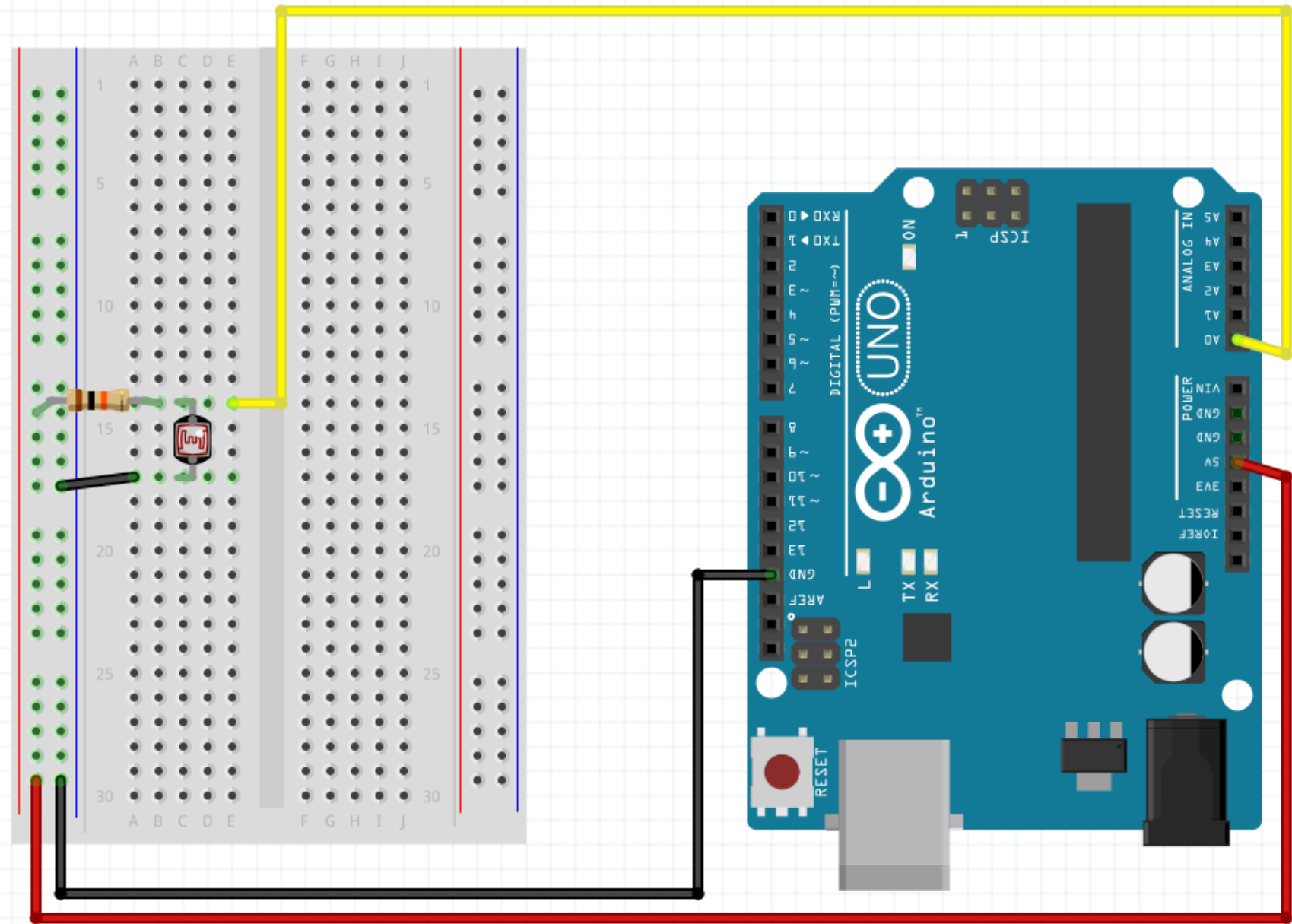
- Breadboard, Elektronika
 - LDR
 - Feszültség osztó
 - Szűrő áramkör
- Arduino UNO
 - Analóg bemenet

MŰKÖDIK?

- Sikeres értékbeolvasás
- Csökkent a zaj
- Elektronikai átépítést igényel

PROJEKT 6

„ZAJ SZŰRÉS” EGYSZERŰ LDR KAPCSOLÁS



PROJEKT 6

„ZAJ SZŰRÉS” SZOFTVERES SZŰRÉS MEGVALÓSÍTÁSA

11_ldr

```
int feny_erzekelo = 0;

int beolvasott_ertek;
float feszultseg;

float a;
float b;
float eloza_ertek;
float aktualis_ertek;
float uj_ertek;

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

    a = 0.2;
    b = 1.0 - a;
    eloza_ertek = 0.0;
}

void loop() {
    beolvasott_ertek = analogRead(feny_erzekelo);
    feszultseg = beolvasott_ertek * 5.0 / 1024;

    aktualis_ertek = feszultseg;
    uj_ertek = aktualis_ertek * a + eloza_ertek * b;
    eloza_ertek = uj_ertek;

    Serial.println(feszultseg);
    Serial.println(" ");
    Serial.println(uj_ertek);

    delay(50);
}
```



PROJEKT 6

„ZAJ SZŰRÉS”

MIT CSINÁLTUNK?

- Arduino Program
 - Analóg bemenet olvasása
 - Átszámolás feszültségre
 - Eltérés súlyozáson alapuló szűrés
- Arduino IDE
 - Grafikonos megjelenítés
 - Több adatsorral

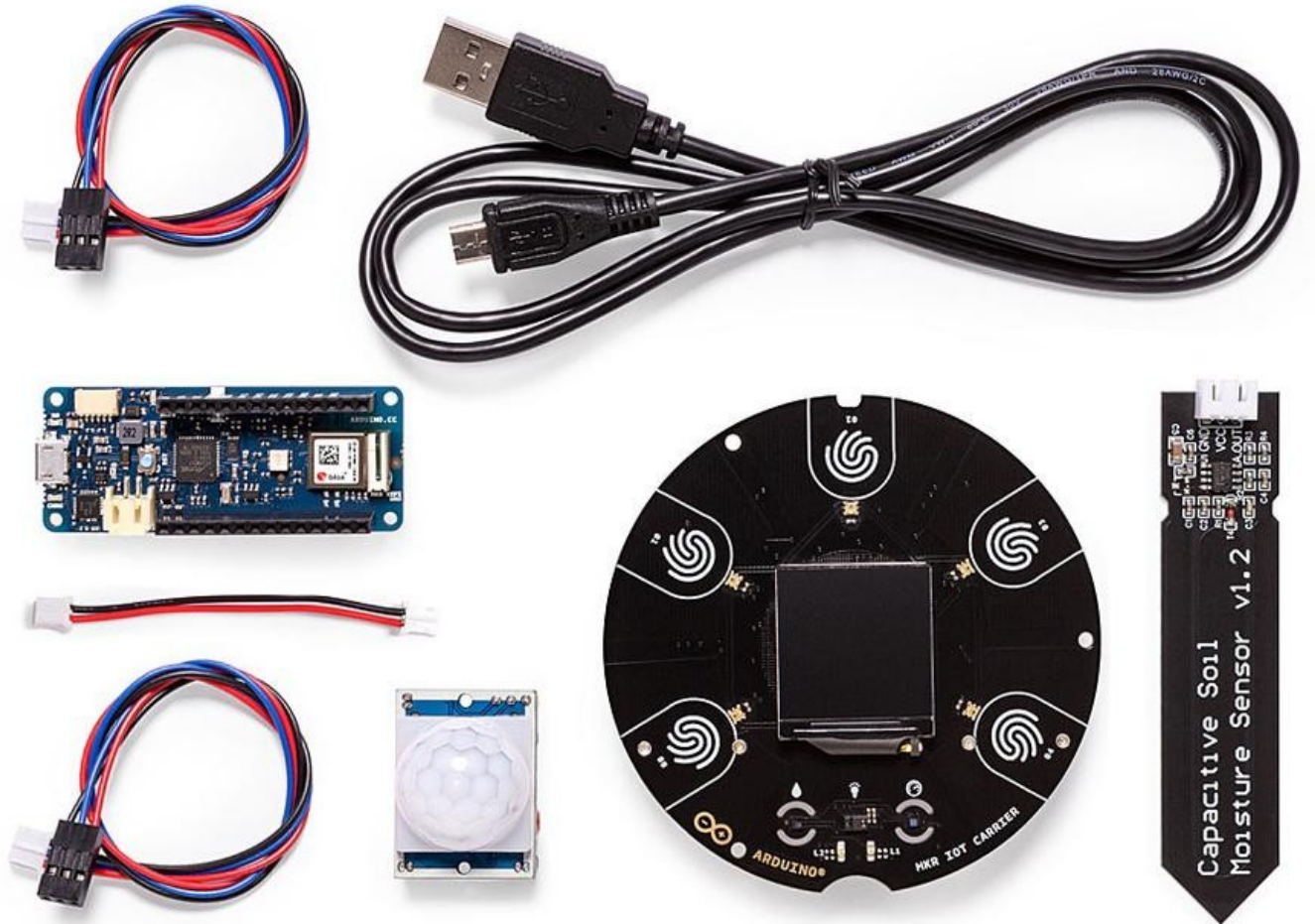
MIVEL CSINÁLTUK?

- Breadboard, Elektronika
 - LDR
 - Feszültség osztó
 - (Szűrő áramkör)
 - Szűrő logika

MŰKÖDIK?

- Sikeres értékbeolvasás
- Csökkent a zaj
- Komplexebb program

EXPLORE IOT KIT

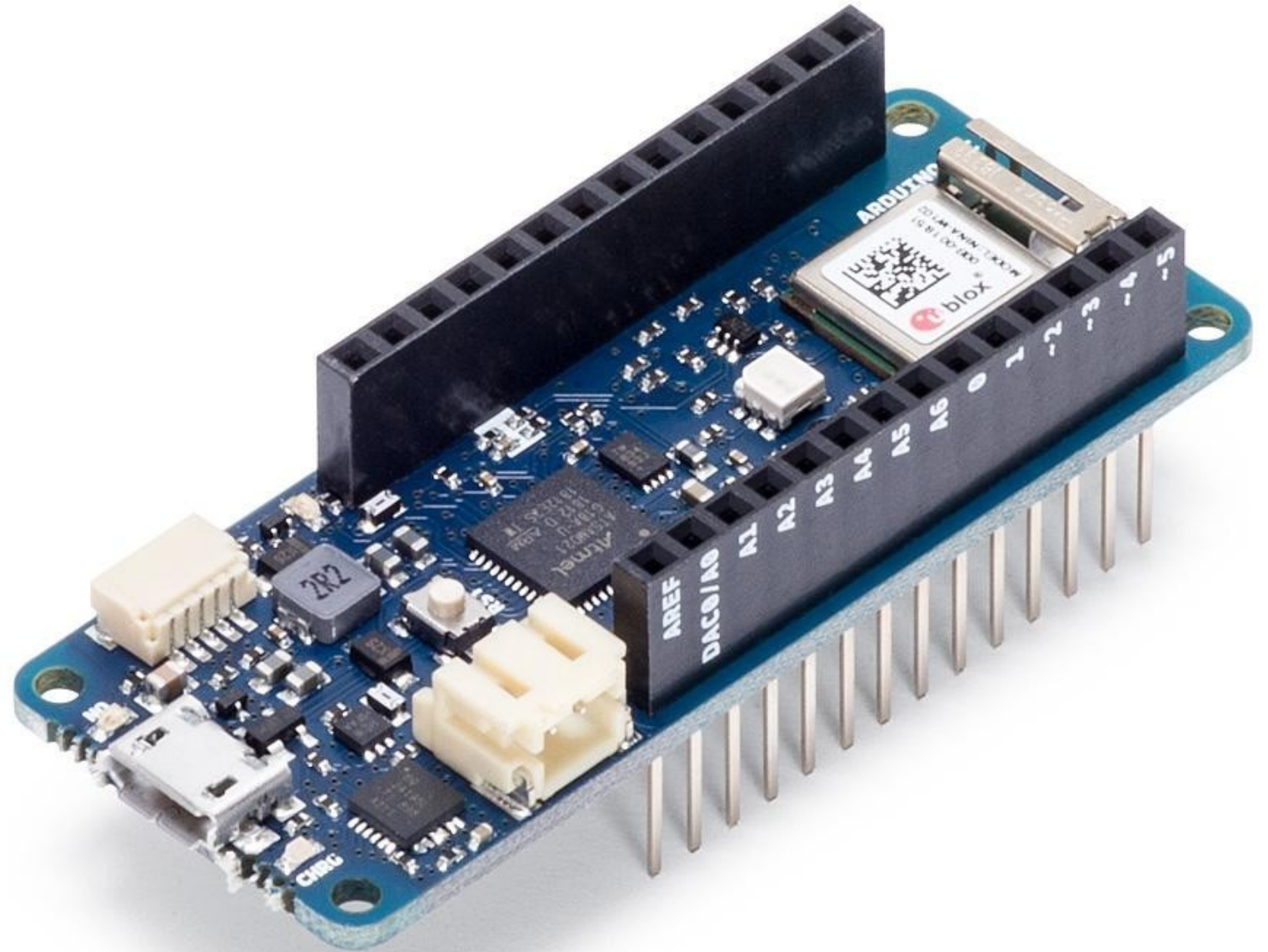


EXPLORE IOT KIT

- Tartalma
 - 4 - Készlet
 - Arduino MKR Wifi 1010
 - Integrált modul
 - Mozgás, Nedvesség érzékelő
- Online tananyag

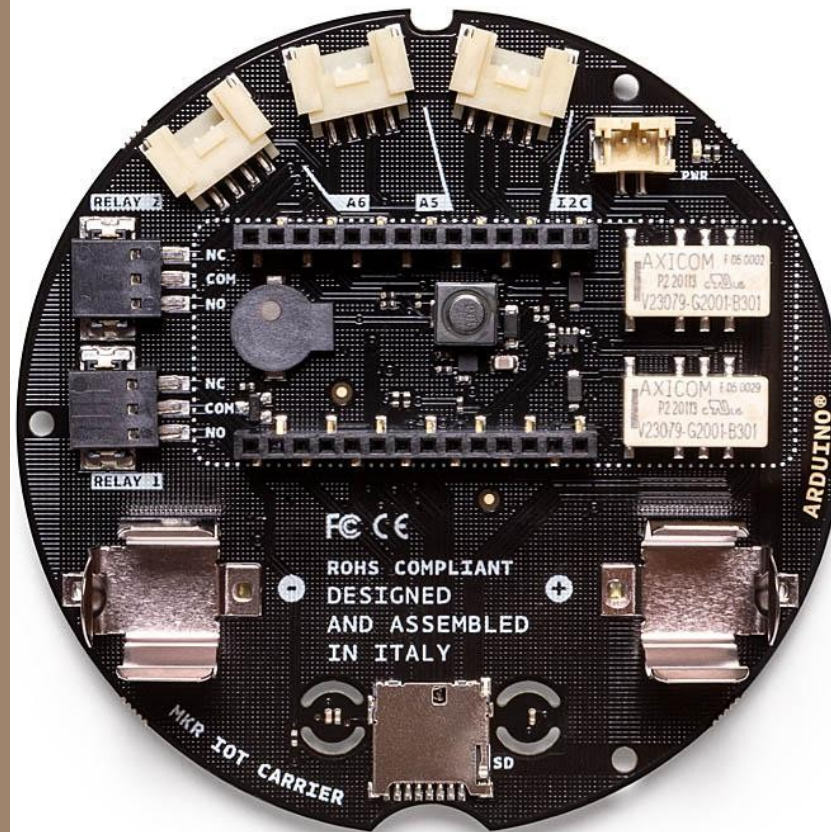


ARDUINO MKR WIFI 1010

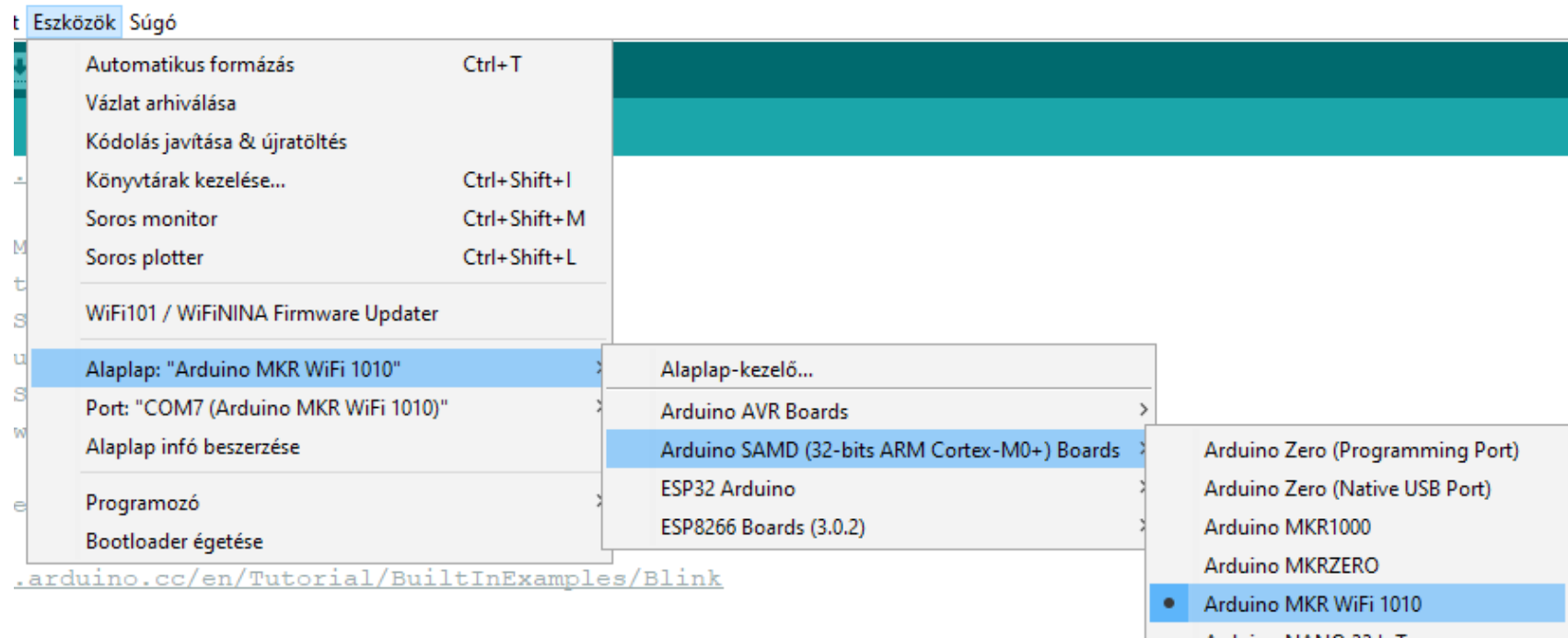
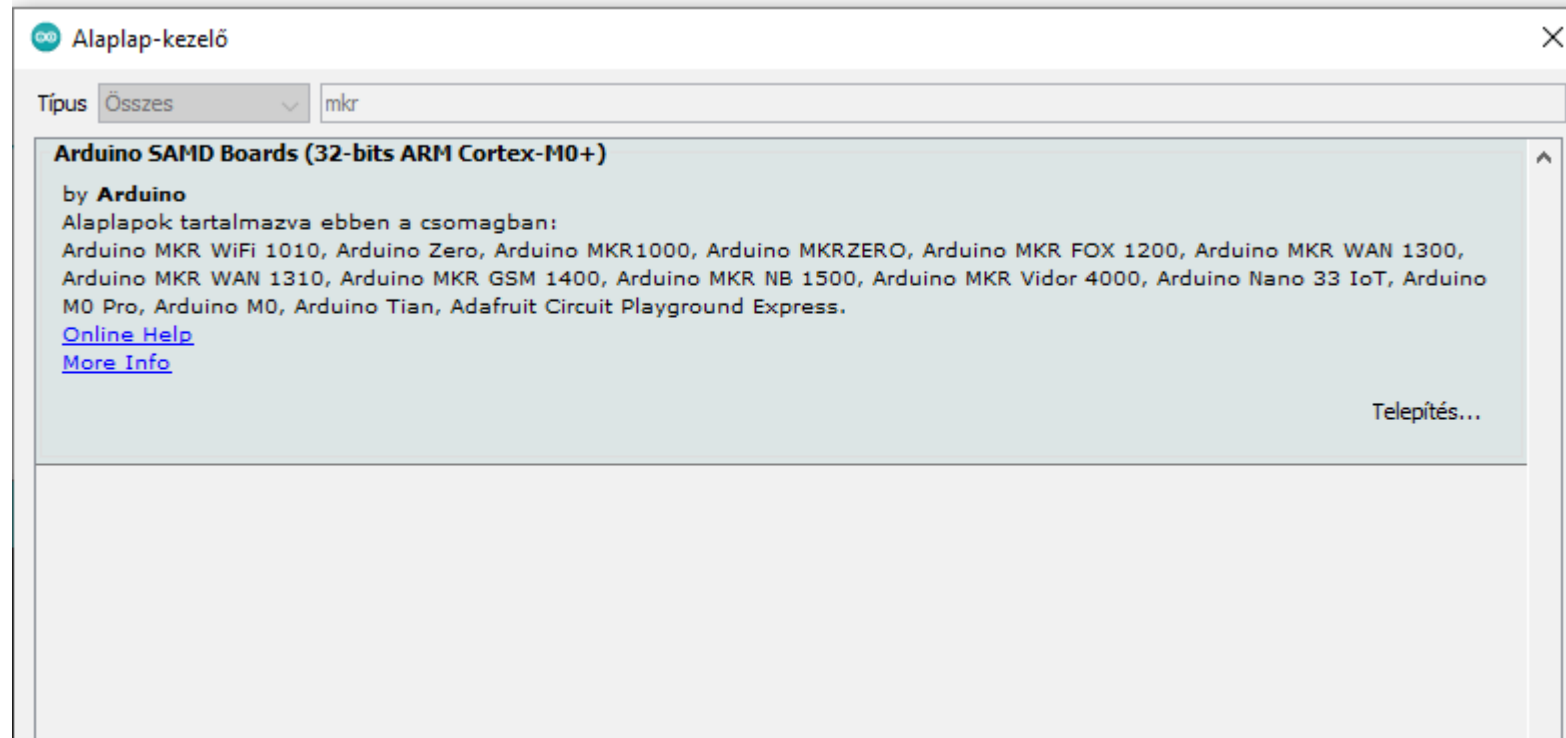


ARDUINO MKR IOT CARRIER

- Szenzorok
 - Nyomás
 - Hő, Páratartalom
 - Fény, Közelség
 - Gyorsulás
 - Érintő felületek
- Akkumlátor tartó
- További csatlakozók



PROJEKT IOT „BOARD TELEPÍTÉSE”

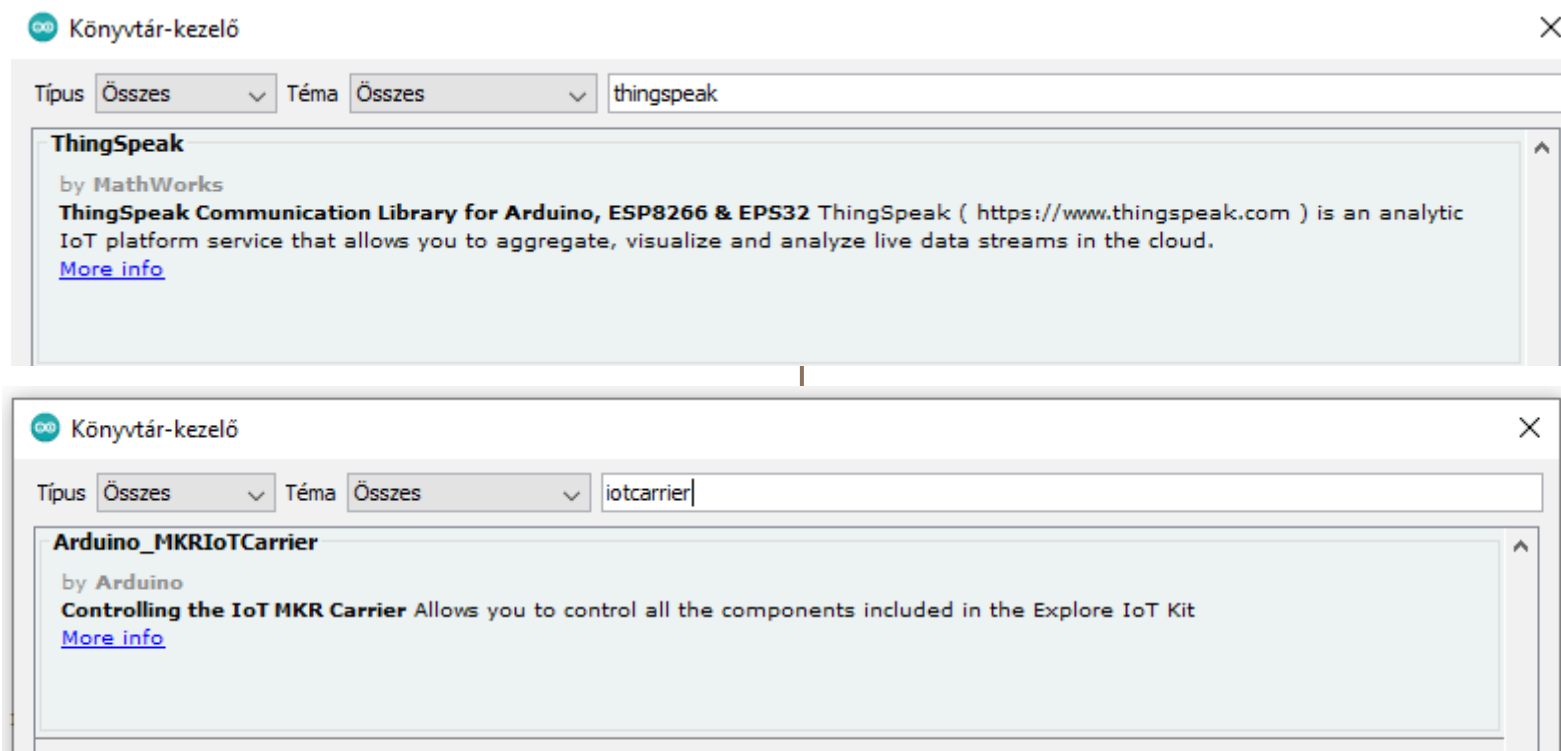


PROJEKT IOT „KÖRNYEZET ELŐKÉSZÍTÉSE”



- Regisztráció
- Bejelentkezés
- Csatorna létrehozása

- API Kulcsok beszerzése
- Könyvtár telepítése
- Példa megnyitása



PROJEKT IOT „PARAMÉTEREK BEVITELE”

- WiFi

- SSID: „”
- Pass: „”

ProjektIoT

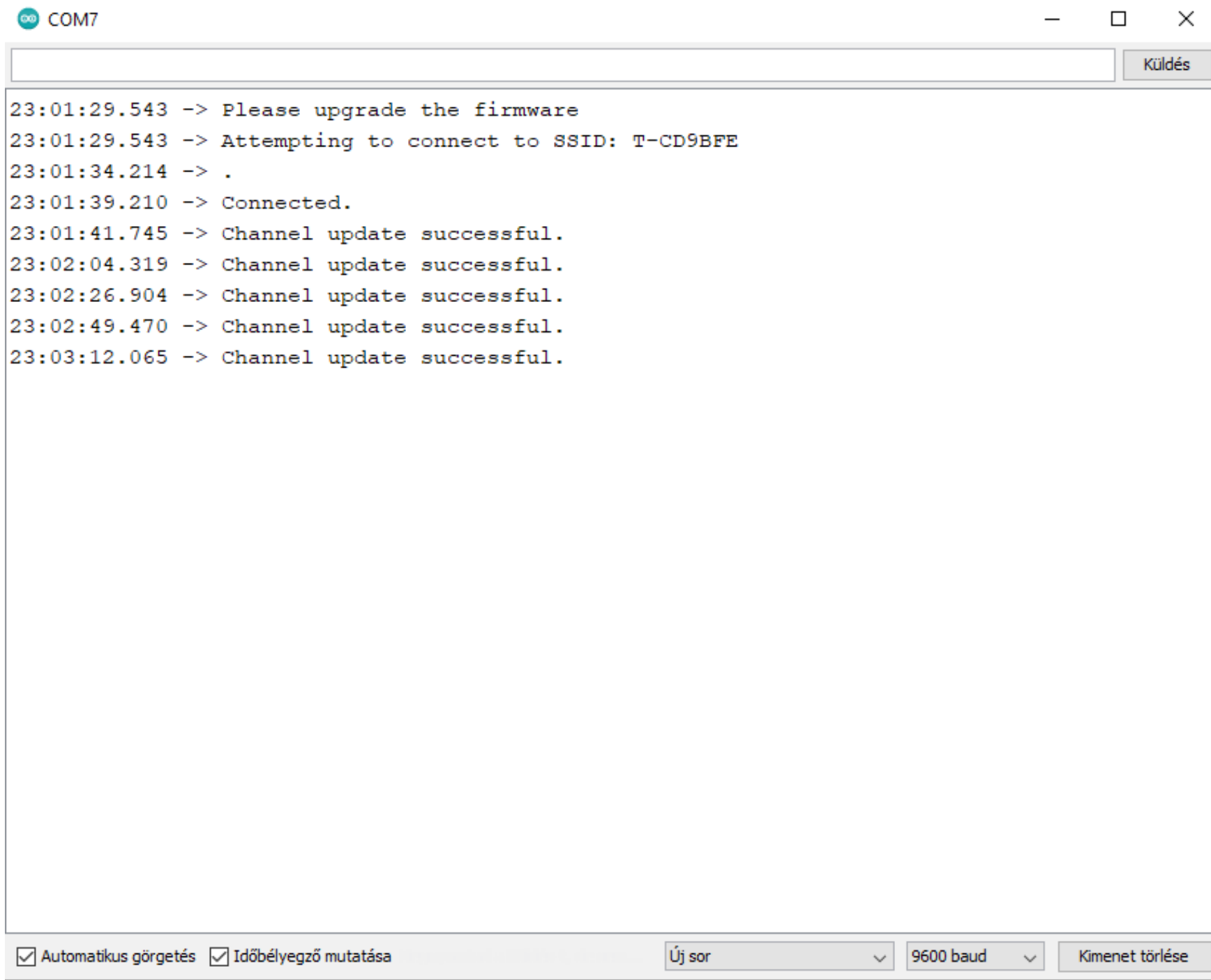
secrets.h §

```
// Use this file to store all of the private credentials  
// and connection details
```

```
#define SECRET_SSID "MySSID"    // replace MySSID with your WiFi network name  
#define SECRET_PASS "MyPassword" // replace MyPassword with your WiFi password
```

```
#define SECRET_CH_ID 000000    // replace 0000000 with your channel number  
#define SECRET_WRITE_APIKEY "XYZ" // replace XYZ with your channel write API Key
```

PROJEKT IOT „PÉLDA KÓD KIMENETE”



The screenshot shows a serial terminal window titled 'COM7'. It features a text input field at the top right with a 'Küldés' (Send) button. The main area displays a series of log messages in a monospaced font. At the bottom, there are checkboxes for 'Automatikus görgetés' (Auto scroll) and 'Időbélyegző mutatója' (Timestamp), along with dropdown menus for 'Új sor' (New line) and '9600 baud', and a 'Kimenet törlése' (Clear output) button.

```
COM7

23:01:29.543 -> Please upgrade the firmware
23:01:29.543 -> Attempting to connect to SSID: T-CD9BFE
23:01:34.214 -> .
23:01:39.210 -> Connected.
23:01:41.745 -> Channel update successful.
23:02:04.319 -> Channel update successful.
23:02:26.904 -> Channel update successful.
23:02:49.470 -> Channel update successful.
23:03:12.065 -> Channel update successful.
```

☒ Automatikus görgetés ☒ Időbélyegző mutatója Új sor 9600 baud Kimenet törlése

PROJEKT IOT „ADATOK A FELHŐBEN”

IOT test

Channel ID: **1505120**

Author: **mwa0000023864985**

Access: Private

Hőmérséklet értékek a felhőben

Private View

Public View

Channel Settings

Sharing

API Keys

Data Import / Export

+ Add Visualizations

+ Add Widgets

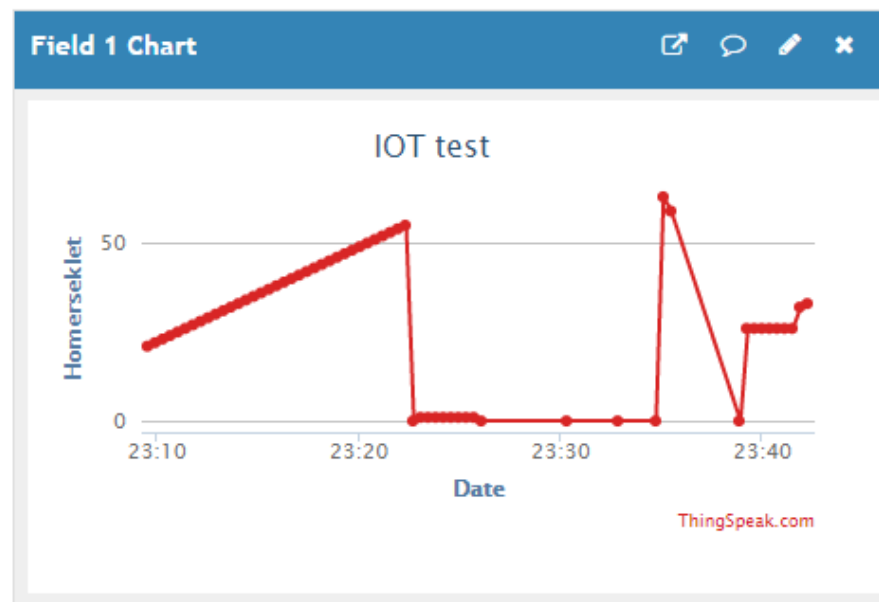
Export recent data

Channel Stats

Created: a day ago

Last entry: less than a minute ago

Entries: 112



PROJEKT IOT „CARRIER HASZNÁLATA”

Carrier Könyvtár behívása

```
#include <WiFiNINA.h>
#include "secrets.h"
#include "ThingSpeak.h" // always include thingspeak header file after other header files and custom macros

//IoTCarrier könyvtár behívása, inicializálása
#include <Arduino_MKRIoTCarrier.h>
MKRIoTCarrier carrier;

char ssid[] = SECRET_SSID;    // your network SSID (name)
char pass[] = SECRET_PASS;    // your network password
```

Inicializálás, Indítás jelzése

```
ThingSpeak.begin(client); //Initialize ThingSpeak

//Carrier inicializálása
carrier.begin();
//Bekapcsolás jelzése 2 sípolással
carrier.Buzzer.sound(300);
delay(300);
carrier.Buzzer.noSound();
delay(300);
carrier.Buzzer.sound(300);
delay(300);
carrier.Buzzer.noSound();
}
```

Hőmérséklet lekérése

```
// Hőmérséklet lekérése
//number = carrier.Env.readHumidity();
number = carrier.Env.readTemperature();
Serial.println(number);
```

PROJEKT IOT „ZAJ SZŰRÉS”

MIT CSINÁLTUNK?

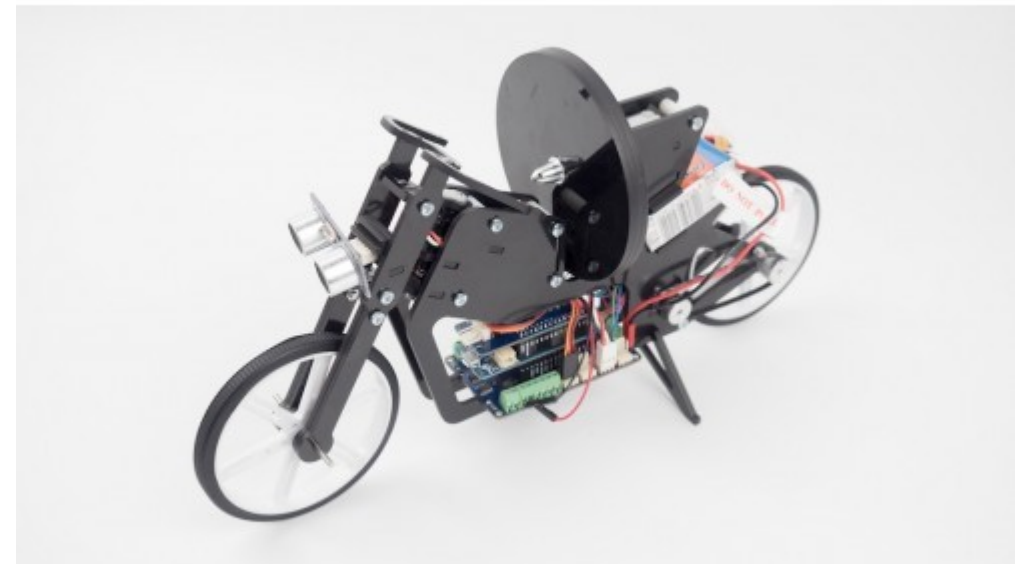
- Arduino Program
 - Kommunikáció a felhővel
 - Adatok küldése
 - Carrier használata
- Thingspeak
 - API kérések küldése
 - Adatok megjelenítése

MIVEL CSINÁLTUK?

- Arduino MKR Wifi 1010
- IoTCarrier
- Internet

ENGINEERING KIT

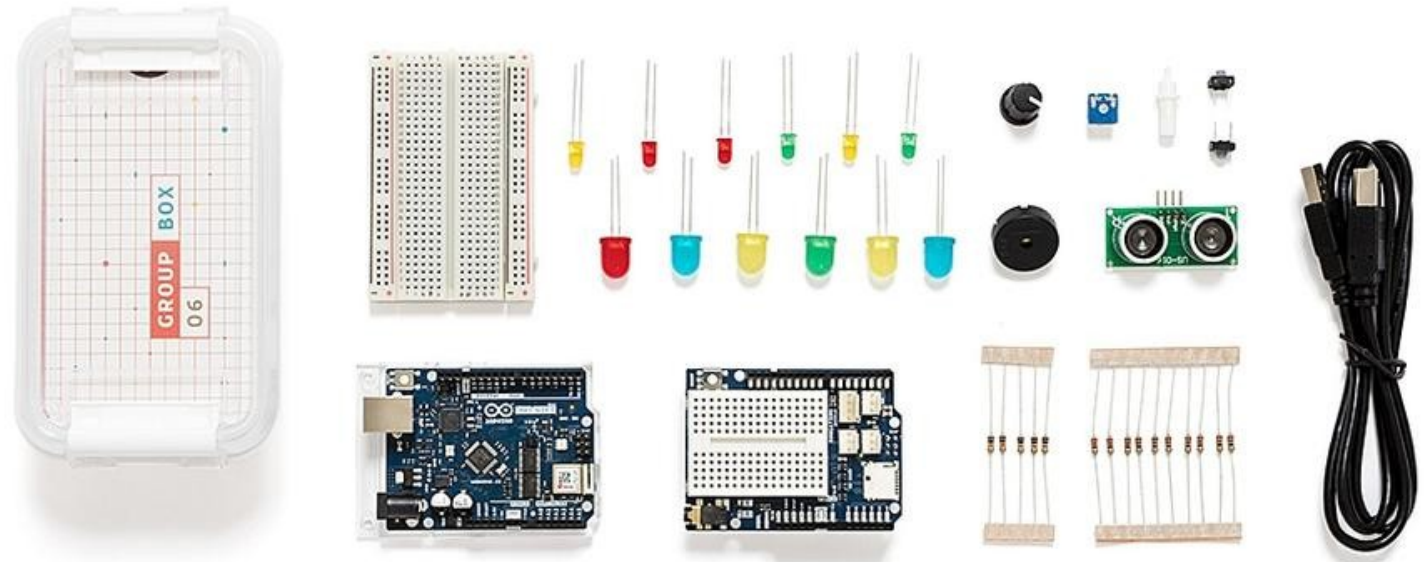
- Tartalma
 - Arduino MKR1000
 - Motor, IMU Shield
 - Érzékelők
 - Akkumulátor
 - Elektronikai alkatrészek
 - Robot alkatrészek
- Online tananyag
 - Egyetem





CTC GO!

- Tartalma
 - Arduino UNO Wifi
 - Education Shield
 - Elektronikai alkatrészek
 - 8 Készlet
- Online tananyag
 - 3 Tanár
 - 24 Diák
 - Elektronika



KÖSZÖNJÜK A
FIGYEELMET!