

Arduino alapok

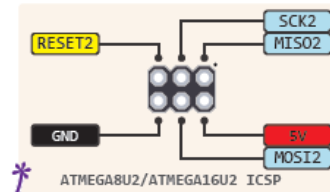
- Sketch ~ Solution
 - Forrás: .ino (1.0 előtt .pde)
 - .C, .CPP, .H
 - Külső könyvtárak (legacy / 3rd party)
 - Mintakódok (example)
- setup()
 - Induláskor fut le, kezdeti értékeket állít be, inicializálja a perifériákat
- loop()
- Referencia:
 - <https://www.arduino.cc/en/Reference/HomePage>

Hello world!

- Led villogtatás

ARDUINO MEGA PINOUT DIAGRAM

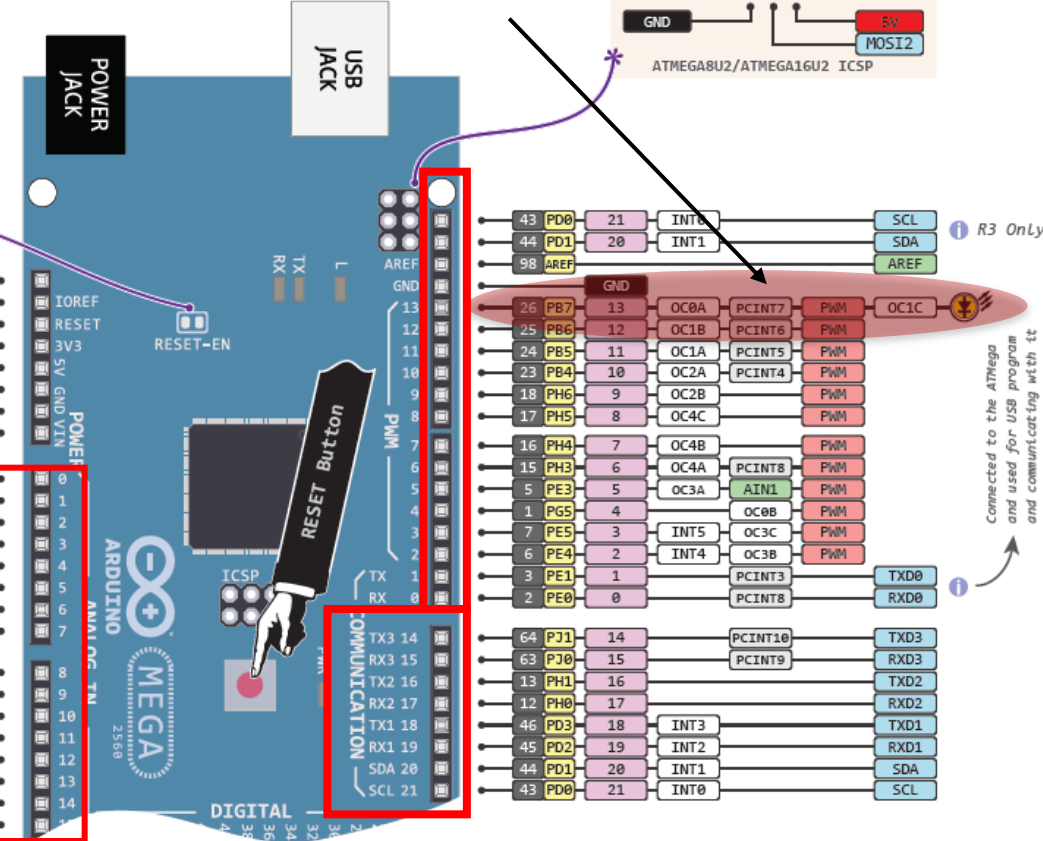
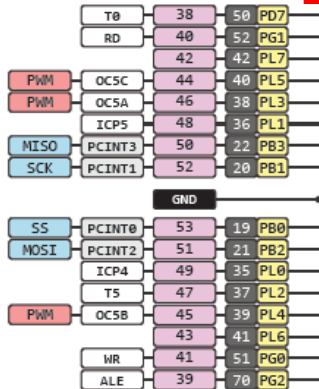
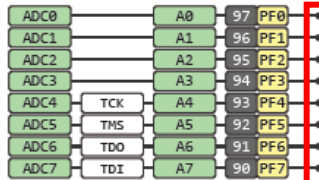
(D)13 = PB7 ≈ = LED



Cut to disable the auto-reset
This provides a Logic reference voltage for shields that use it. It is connected to the 5V bus.

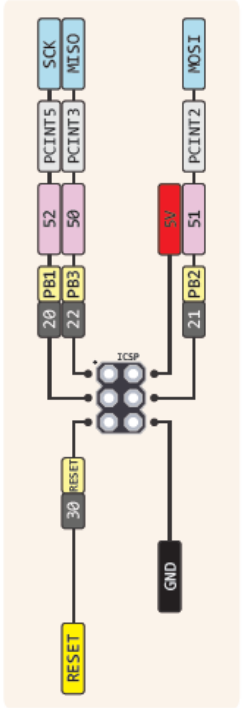
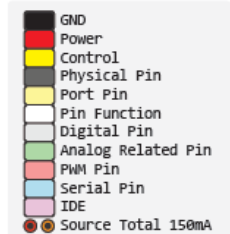
Not Connected
R3 Only

The input voltage to the Arduino board when it is running from external power.
Not USB bus power.

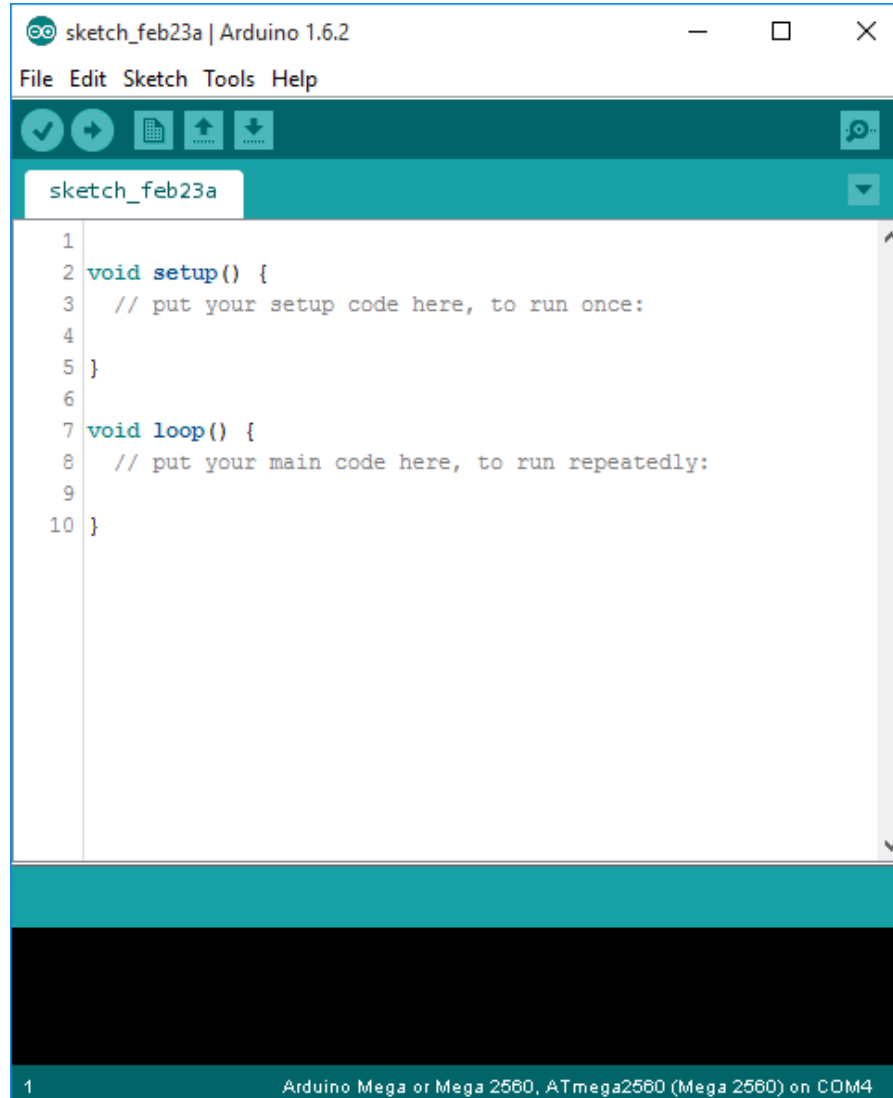


connected to the ATmega and used for USB program and communicating with it

- ⚠ Absolute max per pin 40mA recommended 20mA
- ⚡ Absolute max 200mA for entire package



Arduino fejlesztő környezet

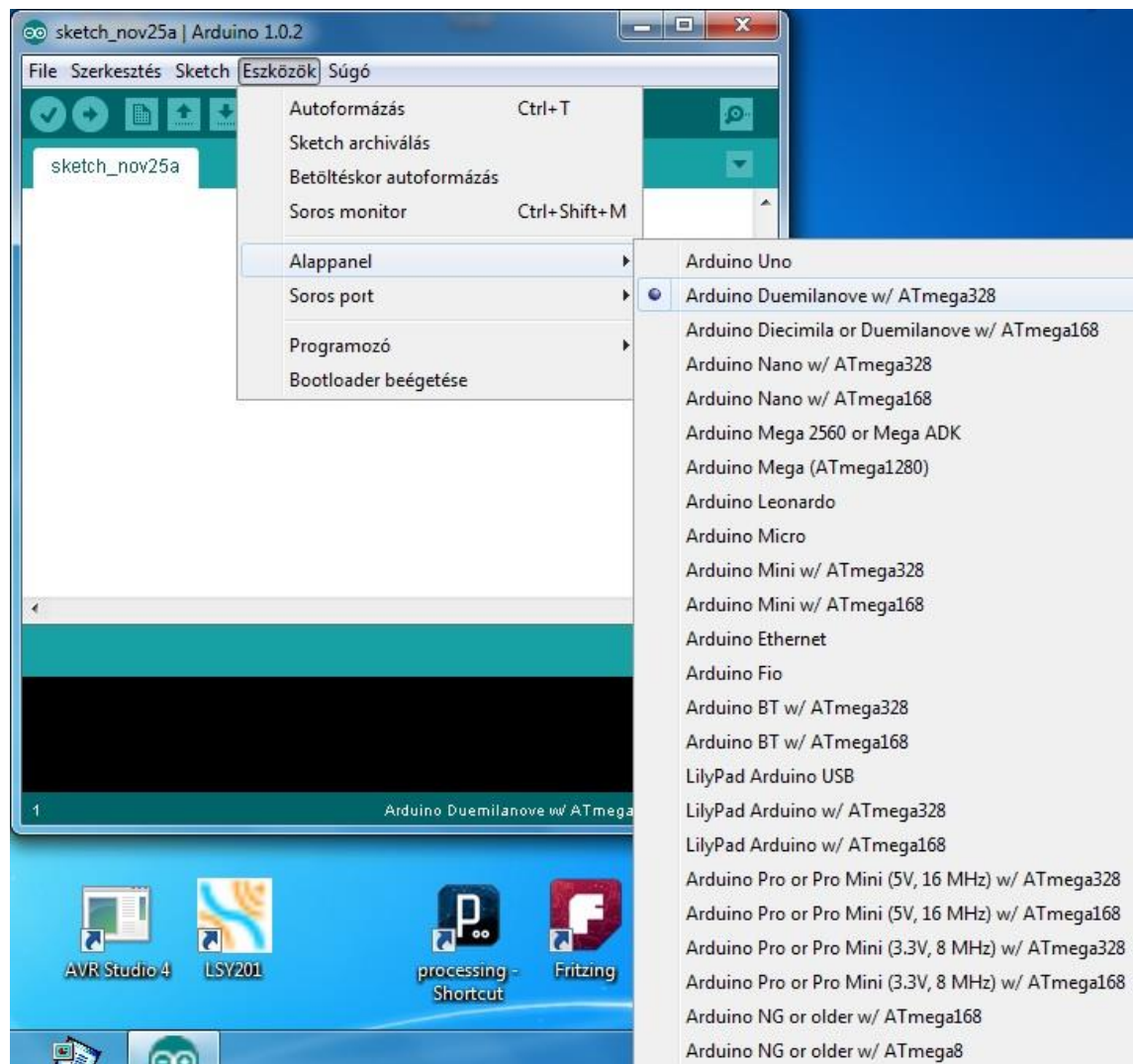


The image shows a screenshot of the Arduino IDE (version 1.6.2) interface. The window title is "sketch_feb23a | Arduino 1.6.2". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for checking, running, uploading, and downloading. The sketch name "sketch_feb23a" is displayed in the top bar. The code editor contains the following C++ code:

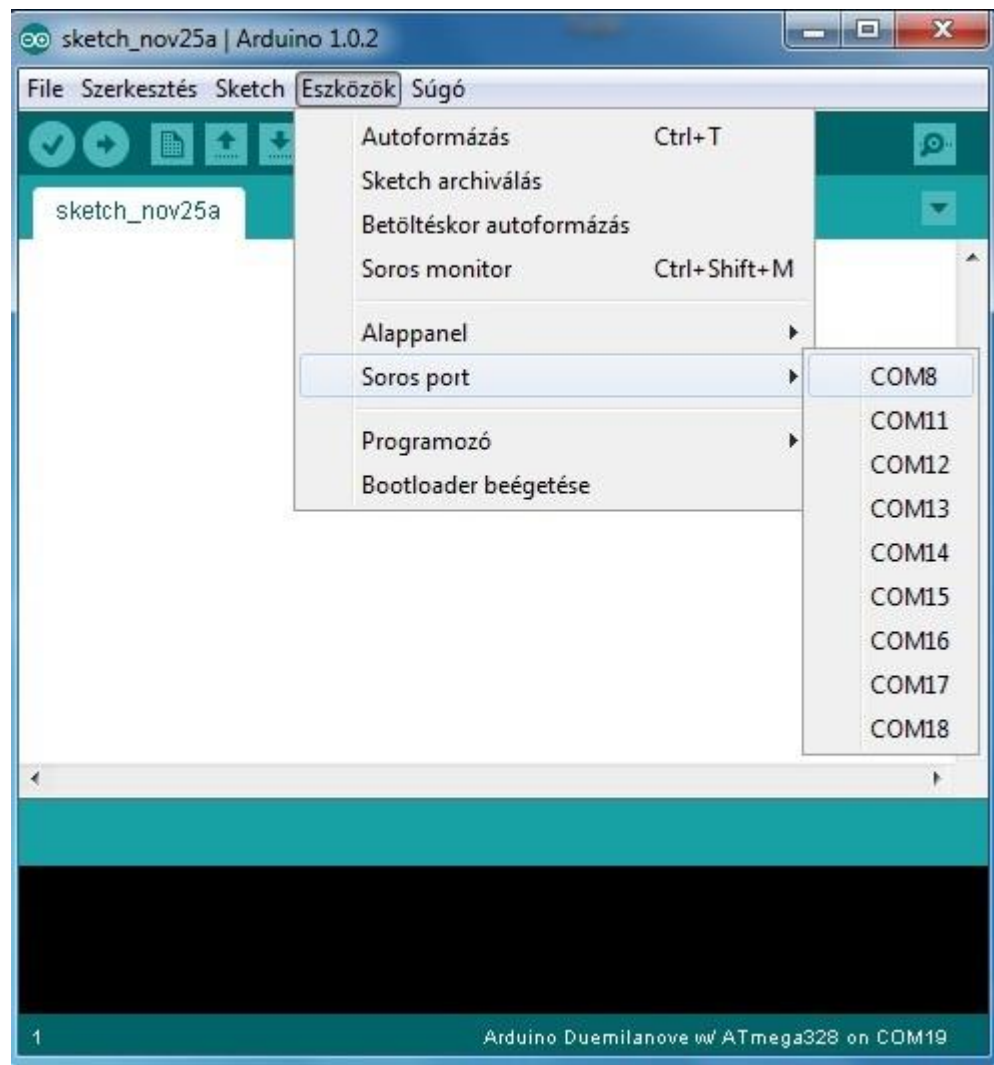
```
1
2 void setup() {
3   // put your setup code here, to run once:
4
5 }
6
7 void loop() {
8   // put your main code here, to run repeatedly:
9
10 }
```

The bottom status bar shows the board and port configuration: "1 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM4".

Arduino fejlesztő környezet



Arduino fejlesztő környezet



Arduino fejlesztő környezet



- **Ellenőrzés:** mielőtt a programot az Arduino panelba töltenénk, le kell fordítanunk. Ezzel a gombbal fordítható le a kód és ellenőrizhető, hogy a programunk hibamentes-e.

Új: új projektet létrehozása

Megnyitás: korábban létrehozott projek megnyitása

Mentés: a jelenlegi projekt elmentése

Feltöltés: a lefordított kód feltöltése az Arduino-ba

Soros Monitor: az Arduino panel által küldött soros adatok megjelenítése egy termináblakban

Program felépítése



```
Fájl Szerkesztés Vázlat Eszközök Súlyó
led_blink$
1 #include <SoftwareSerial.h> //könyvtárra hivatkozás
2
3 //változók deklarációja
4 int pin = 2; // I/O láb száma
5 bool status = false;
6
7 //inicializálás
8 void setup() {
9     pinMode(pin, OUTPUT); // láb konfigurálása
10 }
11
12 //főciklus
13 void loop() {
14     LedBlink();
15     delay(1000); // késleltetés
16 }
17
18 //Villogtató eljárás
19 void LedBlink(){
20     if (status==false){
21         digitalWrite(pin, HIGH);
22         status=true;
23     }else{
24         digitalWrite(pin, LOW);
25         status=false;
26     }
27 }
```

Hello world!

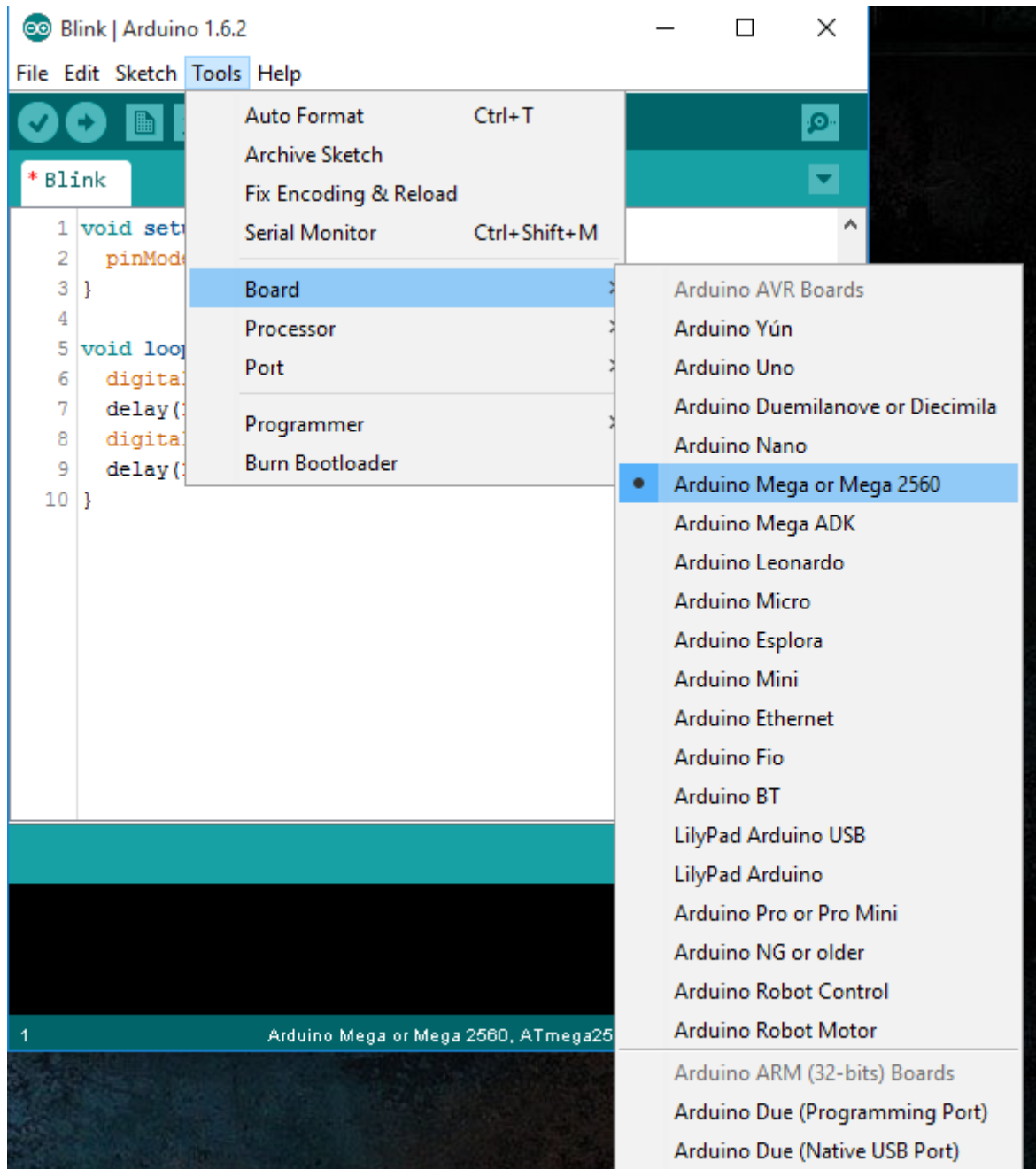
```
1 void setup() {
2   pinMode(13, OUTPUT);
3 }
4
5 void loop() {
6   digitalWrite(13, HIGH);
7   delay(1000);
8   digitalWrite(13, LOW);
9   delay(1000);
10 }
```

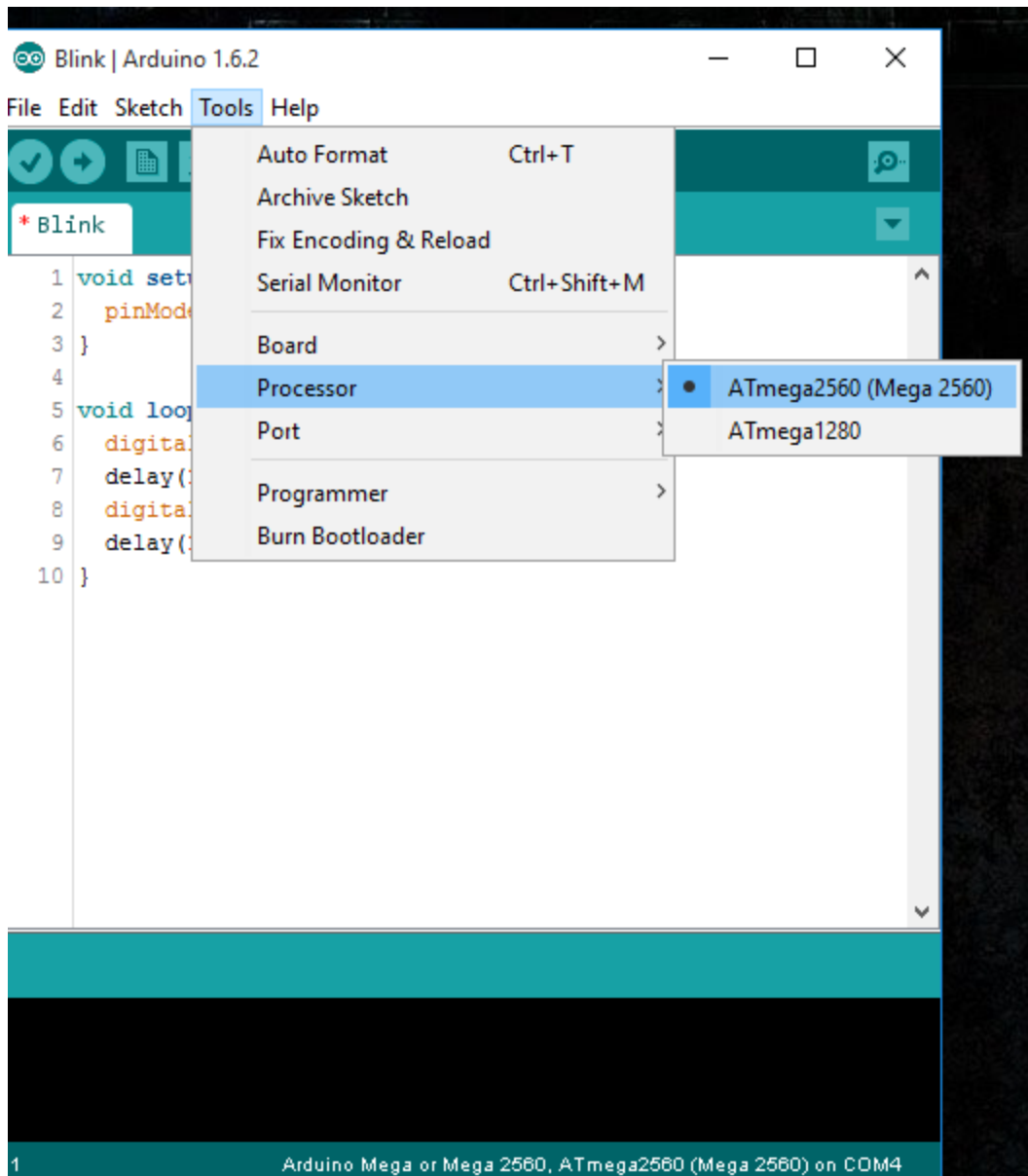
- pinMode(pin, mode)
- Pin: aktuális láb (13)
- Mode:
 - INPUT
 - OUTPUT
 - INPUT_PULLUP
- Arduino (Atmega) lábak alapbeállítása bemenet!
- Kimenet terhelhetősége max 40mA (20mA javasolt!)

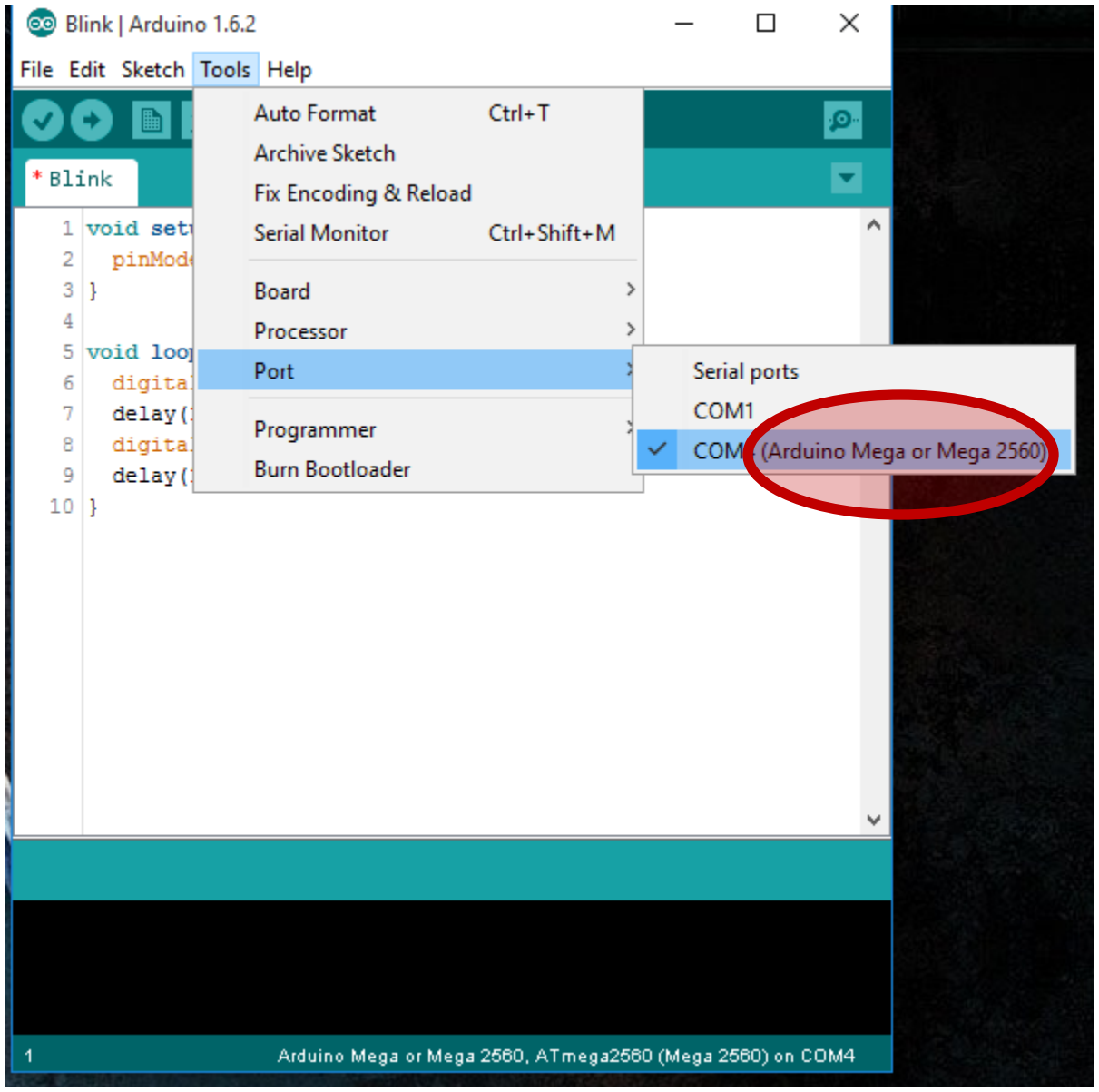
Hello world!

```
1 void setup() {
2   pinMode(13, OUTPUT);
3 }
4
5 void loop() {
6   digitalWrite(13, HIGH);
7   delay(1000);
8   digitalWrite(13, LOW);
9   delay(1000);
10 }
```

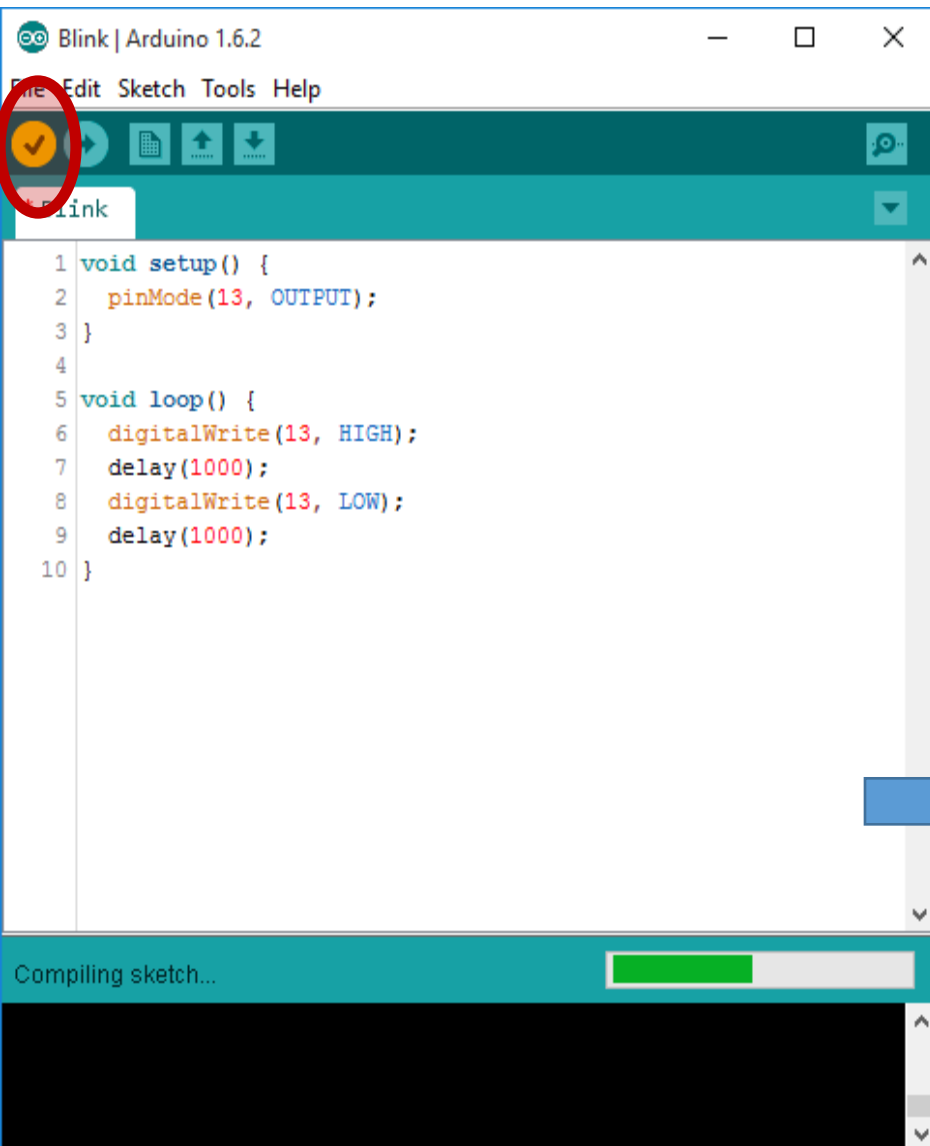
- digitalWrite(pin, value)
 - pin: Láb
 - Value: HIGH, LOW
- delay(ms)
 - Unsigned long ms: millisecc.
 - Nem maszkolja a megszakításokat!







Verify



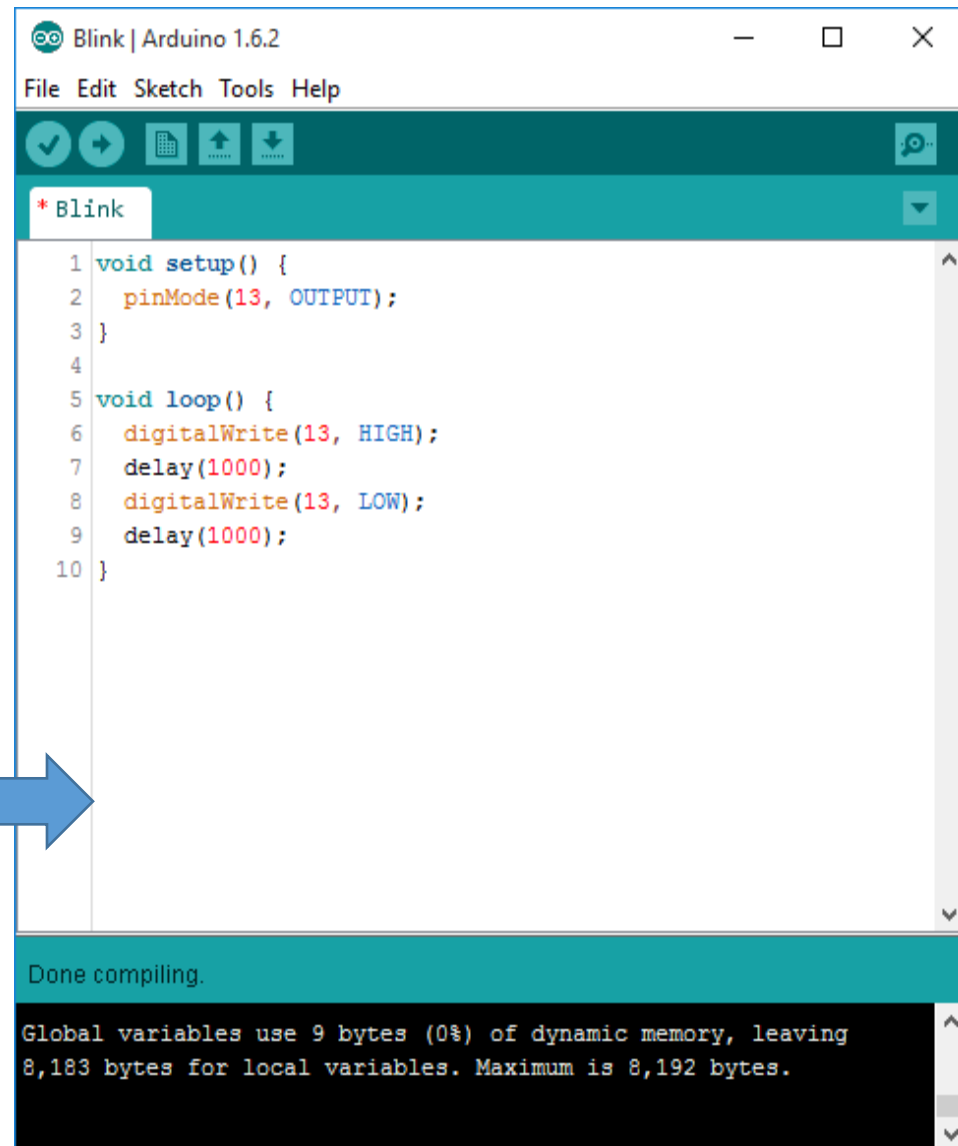
Blink | Arduino 1.6.2

File Edit Sketch Tools Help

Verify

```
1 void setup() {
2   pinMode(13, OUTPUT);
3 }
4
5 void loop() {
6   digitalWrite(13, HIGH);
7   delay(1000);
8   digitalWrite(13, LOW);
9   delay(1000);
10 }
```

Compiling sketch...



Blink | Arduino 1.6.2

File Edit Sketch Tools Help

* Blink

```
1 void setup() {
2   pinMode(13, OUTPUT);
3 }
4
5 void loop() {
6   digitalWrite(13, HIGH);
7   delay(1000);
8   digitalWrite(13, LOW);
9   delay(1000);
10 }
```

Done compiling.

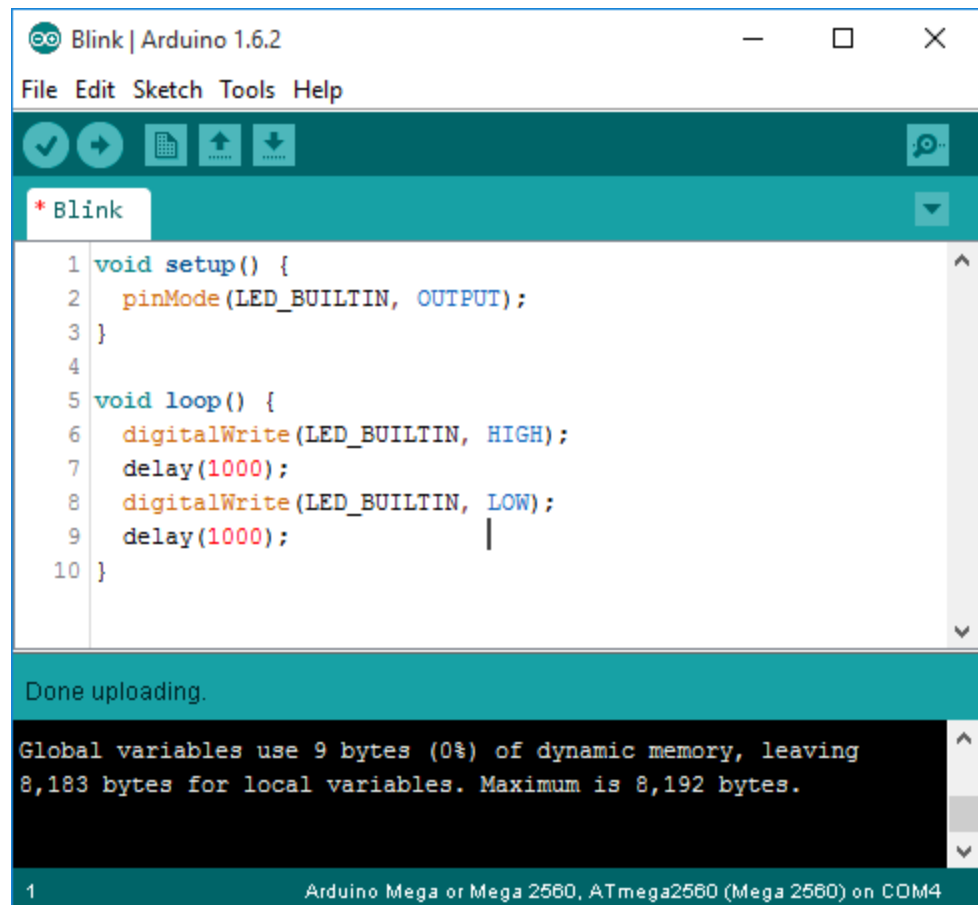
Global variables use 9 bytes (0%) of dynamic memory, leaving 8,183 bytes for local variables. Maximum is 8,192 bytes.

Upload

```
Blink | Arduino 1.6.2
File Edit Sketch Tools Help
* Blink
1 void setup() {
2   pinMode(13, OUTPUT);
3 }
4
5 void loop() {
6   digitalWrite(13, HIGH);
7   delay(1000);
8   digitalWrite(13, LOW);
9   delay(1000);
10 }
Done uploading.
Global variables use 9 bytes (0%) of dynamic memory, leaving
8,183 bytes for local variables. Maximum is 8,192 bytes.
1 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM4
```


Konstansok

- LED_BUILTIN



The screenshot shows the Arduino IDE interface for a sketch named "Blink". The code in the editor is as follows:

```
1 void setup() {  
2   pinMode(LED_BUILTIN, OUTPUT);  
3 }  
4  
5 void loop() {  
6   digitalWrite(LED_BUILTIN, HIGH);  
7   delay(1000);  
8   digitalWrite(LED_BUILTIN, LOW);  
9   delay(1000);  
10 }
```

Below the code editor, the status bar indicates "Done uploading." and the serial monitor shows the following output:

```
Global variables use 9 bytes (0%) of dynamic memory, leaving  
8,183 bytes for local variables. Maximum is 8,192 bytes.
```

The status bar at the bottom of the IDE shows "1" on the left and "Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM4" on the right.

USART/UART kommunikáció

- **USART: Universal Synchronous Asynchronous Receiver Transmitter** („Univerzális Szinkron és Aszinkron Küldő és Fogadó”-nak lehetne lefordítani).
- Külön hardverként van megvalósítva (szoftveres megvalósításra is van lehetőség!)
- Segítségével lehetőség van más mikro vezérlők, eszközök számára adatot küldeni, fogadni.
- Két lábon keresztül megy a kommunikáció: RX/TX (+ közös FÖLD!)
- TTL jeleket használ (RS-232 és USB ettől különbözik!)

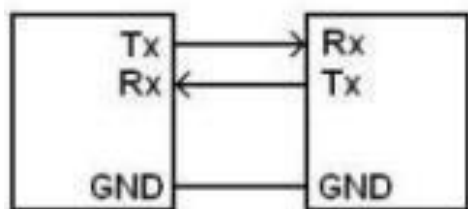
Soros kommunikáció

- A kommunikáció 5 rétegre bontható:
 - **Fizikai réteg:** ezen a csatlakozók kialakítását, tényleges fizikai paramétereit értjük (méret, forma, stb.).
 - **Elektromos réteg:** milyen feszültségszinteket használunk a kommunikációhoz.
 - **Logikai réteg:** melyik feszültségszint jelenti a logikai 1-et és melyik a logikai 0-át.
 - **Adat réteg:** definiálja a kommunikáció sebességét, azt, hogy a biteket 8, 9 vagy 10 bitből álló csoportokban küldjük-e. A csoportok előtt és után használunk-e speciális jelzőbiteket az adatcsoportok elkülönítésére.
 - **Alkalmazási réteg:** azt adja meg, hogy milyen sorrendben kell az egyes adatcsomagokat küldeni, hogyan szerveződnek az adatcsomagok üzenetké.

Soros kommunikáció

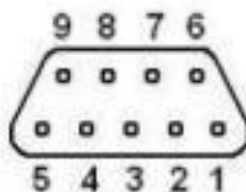
	TTL	RS-232	USB
FIZIKAI réteg	3 vezeték (Rx, Tx és GND)	9 vezeték, de csak 3-at használunk (Rx, Tx és GND)	4 vezeték (DATA+ , DATA- , +5V és GND)
ELEKTROMOS réteg	+5V és 0V	+5 – +25V és -5 – -25V	+5V és 0V
LOGIKAI réteg	Logikai 1: +5V Logikai 0: 0V	Logikai 1: -12V Logikai 0: +12V	<u>Logikai 1:</u> DATA+ > +2.8V és DATA- < 0.3V <u>Logikai 0:</u> DATA+ < 0.3V és DATA- > +2.8V
ADAT réteg	9600 bps, 8 adat bit, 1 START és 1 STOP bit	9600 bps, 8 adat bit, 1 START és 1 STOP bit	Bonyolult, részletesen nem térek ki rá.
ALKALMAZÁSI réteg	Az adatokat bájtanként, a jelzőbitekkel együtt, 10 bites csoportokban küldjük/fogadjuk	Az adatokat bájtanként, a jelzőbitekkel együtt, 10 bites csoportokban küldjük/fogadjuk	Bonyolult, részletesen nem térek ki rá.

Soros kommunikáció fizikai réteg



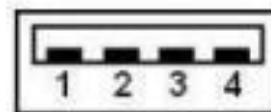
TTL - UART

- Rx
- Tx
- GND

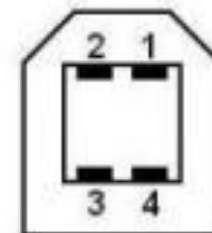


RS-232

- 2 - PC Rx
- 3 - PC Tx
- 5 - PC GND



USB A



USB B

- 1 - +5V
- 2 - DATA +
- 3 - DATA -
- 4 - GND

HOBBI
ELEKTRONIKA

Soros kommunikáció

- **TTL:**

- UART esetén 3 vezeték kell. Egy adó (TX-Transmit) egy vevő (RX-Receive) és egy KÖZÖS FÖLD (GND)! Az adó a TX vonalon küldi az adatot, a vevő pedig az RX-vonalon keresztül fogadja.

- **RS-232:**

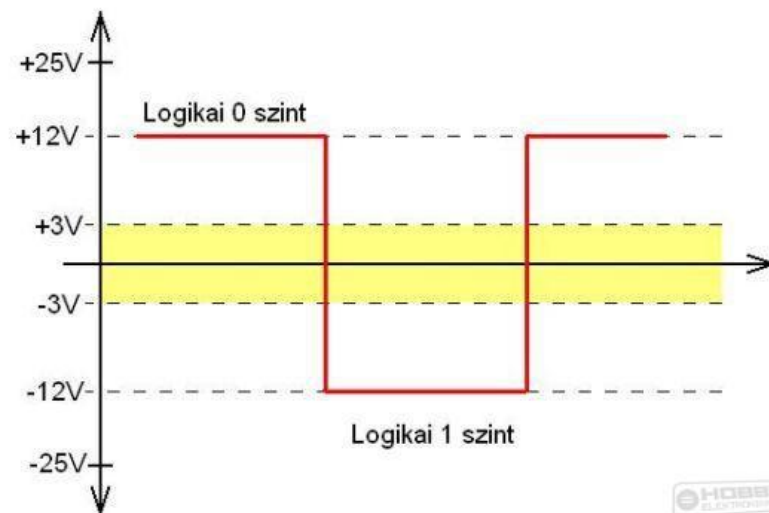
- 9 pólusú ún. D-Sub csatlakozót használ, azonban az aszinkron soros kommunikáció során szintén csak három vezetékre van szükség: RxD, TxD, GND

- **USB:**

- Két adatvonalat (DATA+ és DATA-), egy közös GND és egy +5V (VCC) vezeték.

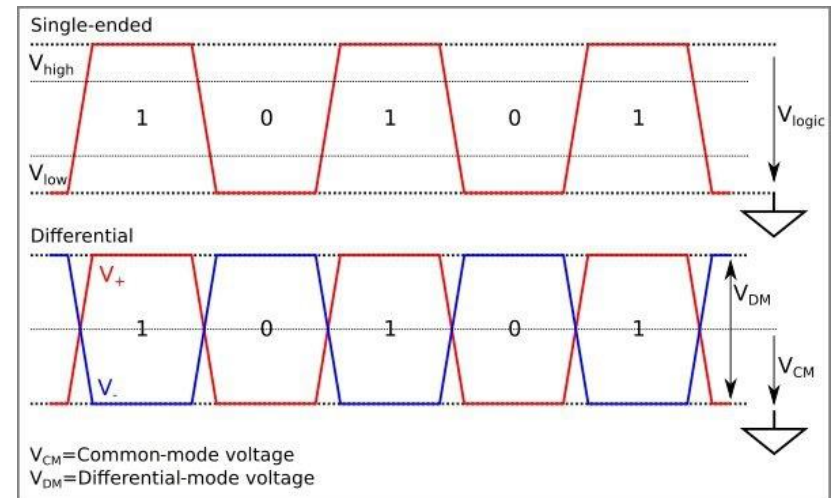
ELEKTROMOS és LOGIKAI réteg

- TTL logikai szintek:
 - Logikai 1 szint: 5V (3V)
 - Logikai 0 szint: 0V
- RS-232 logikai szintek:
 - -3V és -25V között logikai 1
 - +3V és +25V között logikai 0
 - PC-kben ez általában +-12V
 - -3V és +3V közötti rész tiltott!



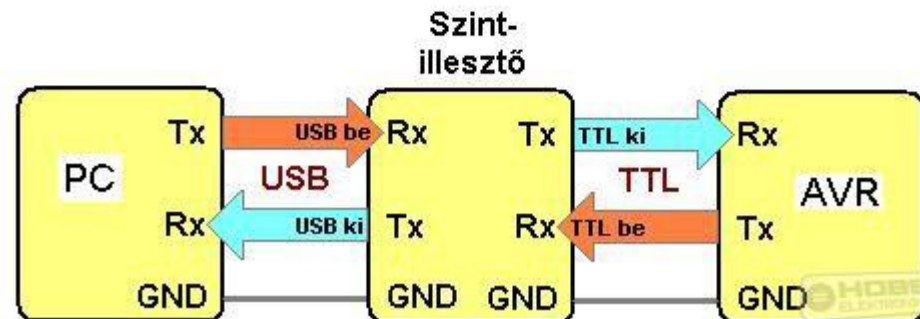
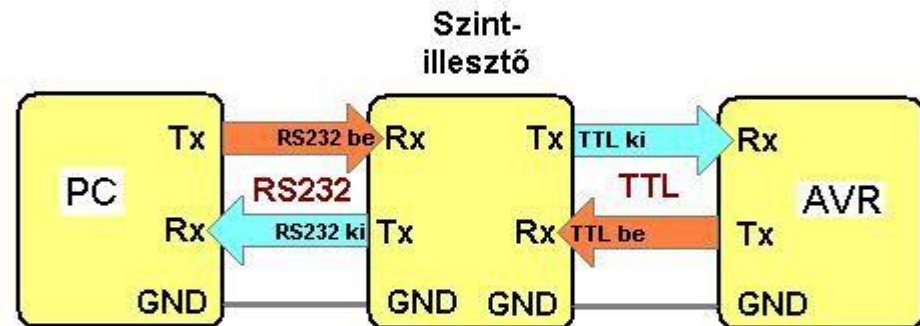
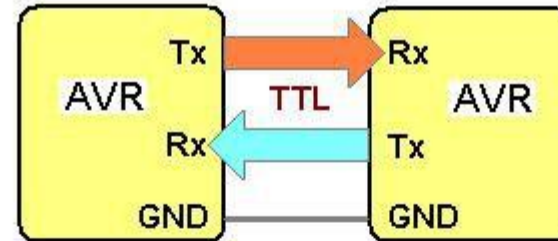
ELEKTROMOS és LOGIKAI réteg

- TTL logikai szintek:
 - Logikai 1 szint: 5V (3V)
 - Logikai 0 szint: 0V
- RS-232 logikai szintek:
 - -3V és -25V között logikai 1
 - +3V és +25V között logikai 0
 - PC-kben ez általában +-12V
 - -3V és +3V közötti rész tiltott!
- USB:
 - Logikai 1 szint: ha DATA+ 2,8 V fölötti értékű és ezzel egyidejűleg a DATA- pedig 0,3 V alatti
 - Logikai 0 szint: ha DATA- 2,8 V fölötti értékű és ezzel egyidejűleg a DATA+ pedig 0,3 V alatti



Szintillesztés

- Amit lehet:
 - TTI->TTL
 - RS-232->RS-232
 - USB->USB
- Ezek kombinációjához szintillesztőt kell használni!

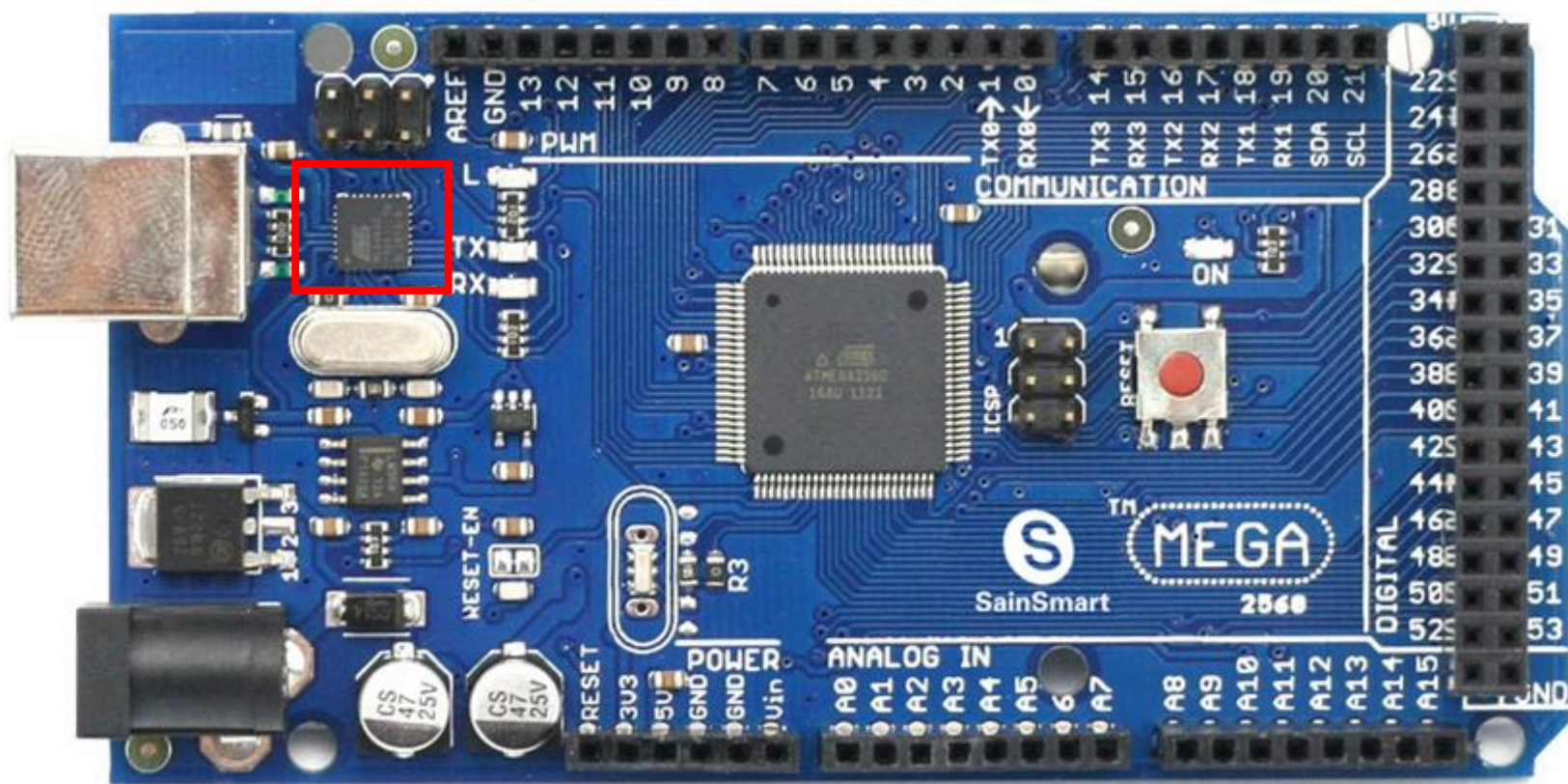


Szintillesztés

- TTL-USB illesztés pl. az FTDI cég által gyártott TTL-USB jelszint illesztő modullal.
- Vagy CP-2102
- Van 3V és 5V verzió!!!



Programozó

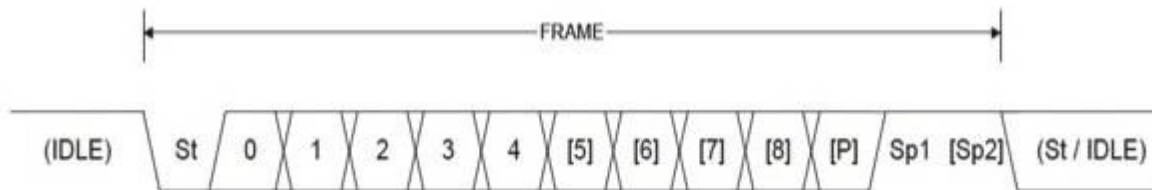


ADAT és ALKALMAZÁSI réteg

- Adat küldése egymás után bitről-bitre.
- UART esetén nem kell az órajelet átvinni egy közös vezetéken, hanem előre meg kell adni mind a két oldalon az átviteli sebességet! (baud-rate Pl. 9600 bps [bit-per-second])
- $9600\text{bps} \Rightarrow 1\text{bit } 1/9600\text{s} = 0.000104\text{s} = 104\mu\text{s}$
- A fogadó eszköz újra byte-okká alakítja

Kommunikációs protokoll

- **1 bit**: Kommunikáció kezdetét jelző START bit a vevő számára
- **8 bit**: adat
- 1 paritás bit
- **1** vagy 2 bit: kommunikáció végét jelző STOP bot



St Start bit, always low.

(n) Data bits (0 to 8).

P Parity bit. Can be odd or even.

Sp Stop bit, always high.

IDLE No transfers on the communication line (RxD or TxD). An IDLE line must be high.

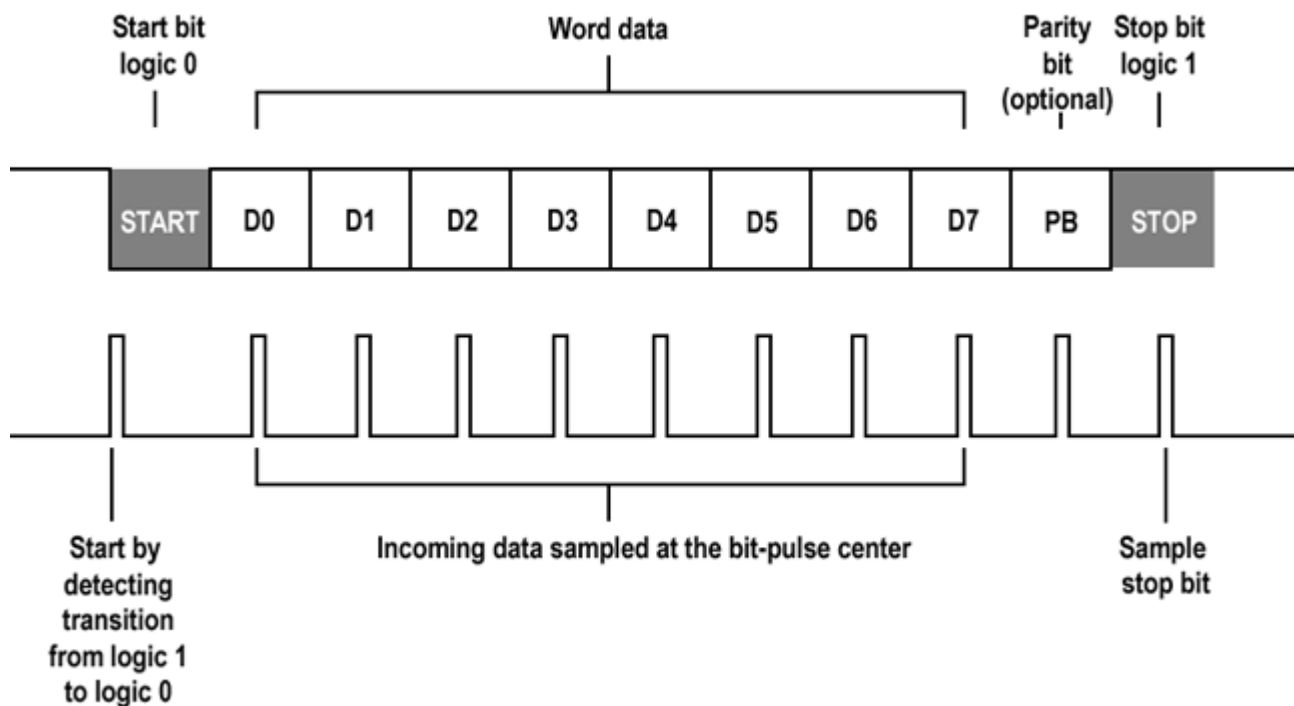
Kommunikációs protokoll/Küldés

- Adásszünetben a TX vonal logikai 1 értéken van
- Küldés esetén a TX vonalon kiküldésre kerül a START bit (104 us-ig logikai **0 szint** (9600bps esetén))
- Kiküldésre kerülnek az adatbitek
- Ha nincs paritás akkor a STOP bit kerül kiküldésre (TX vonal legalább 104 us –ig **logikai 1**)
- Utána vagy következik egy újabb START bit, vagy marad a TX logikai **1 szinten** (várakozás)

Kommunikációs protokoll/Fogadás

- A fogadó várakozik a START bitre (RX vonal logikai 0 állapotba kerüljön)
- START bit esetén, várakozik „fél bitnyi” időt ($104/2=52\mu\text{s}$ -ig) majd újra leolvassa az RX állapotát, hogy biztos START bit érkezett-e, nem pedig valami zaj.
- Ha START bit érkezett, akkor egy újabb periódusnyi időt vár ($104\mu\text{s}$), ebben az esetben az első adatbit közepénél tart, majd mintavételezi az RX vonal értékét.
- Ez addig tart, amíg a STOP bit meg nem érkezik.

Kommunikációs protokoll/Fogadás



U(S)ART

- Atmega: 4 uart
- Serial (0)
 - 0 (RX)
 - 1 (TX)
- Gyárilag bekötve VCP-re (virtual com port) (FTDI, CN340, cp-2102)
- Arduino IDE serial monitorral megnyitható
- Serial.begin(baud)
 - 8n1 (8 adat bit, nincs paritás, 1 stop bit)
- Serial.begin(baud,config)
 - Config: pl. SERIAL_8N1 (the default)

File Edit Sketch Tools Help



* Blink



```
1 void setup() {  
2   pinMode(LED_BUILTIN, unsigned long baud)  
3   Serial.begin(baud)  
4 }
```

```
5  
6 void loop() {  
7   digitalWrite(LED_BUILTIN, HIGH);  
8   delay(1000);  
9   digitalWrite(LED_BUILTIN, LOW);  
10  delay(1000);  
11 }
```

1200

2400

4800

9600

14400

19200

28800

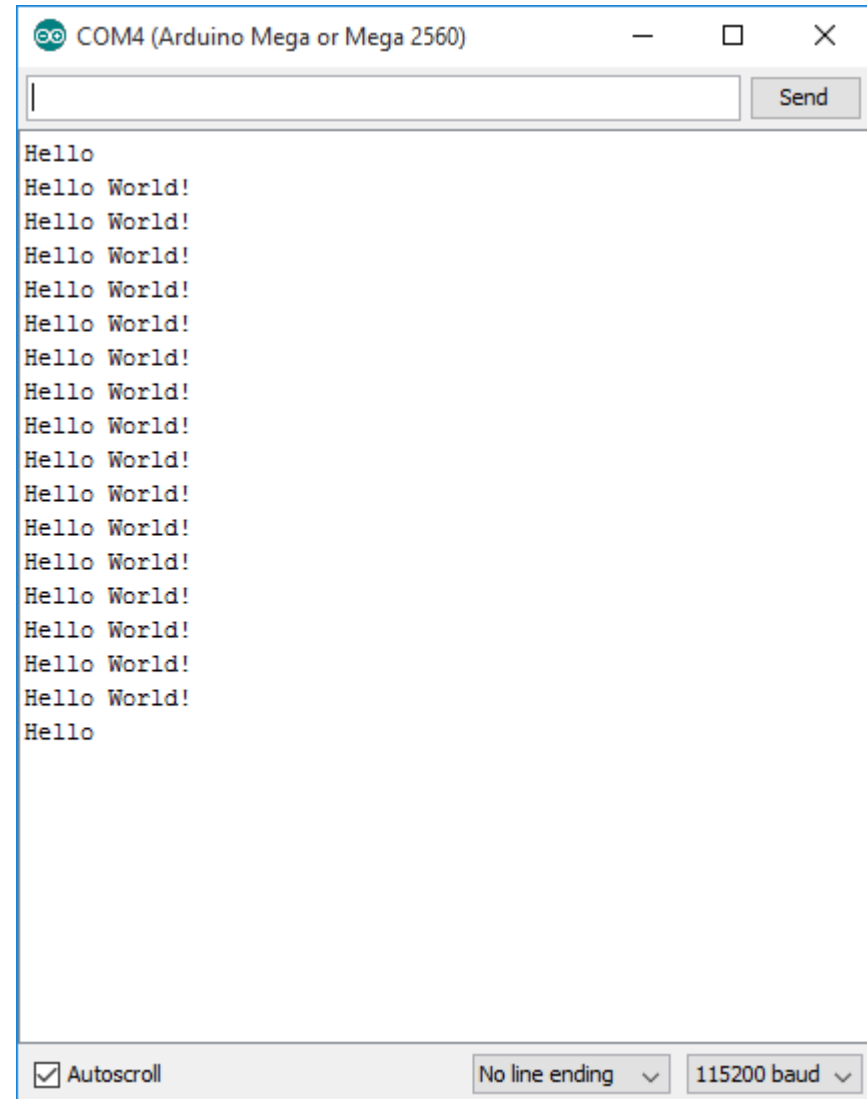
38400

57600

115200

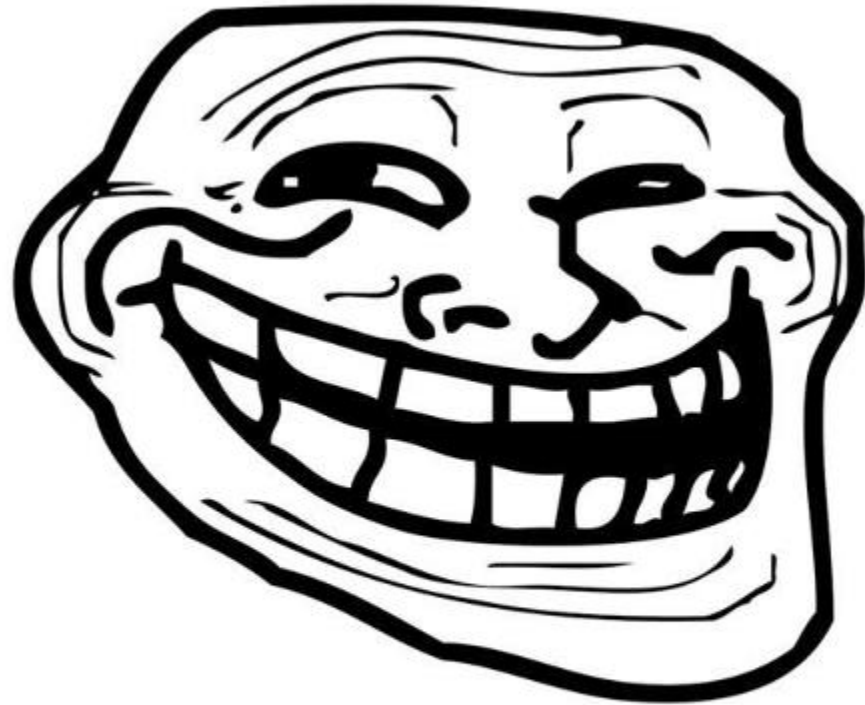
Tools / Serial monitor

```
1 void setup() {  
2   pinMode(LED_BUILTIN, OUTPUT);  
3   Serial.begin(115200);  
4 }  
5  
6 void loop() {  
7   digitalWrite(LED_BUILTIN, HIGH);  
8   Serial.print("Hello");  
9   delay(1000);  
10  digitalWrite(LED_BUILTIN, LOW);  
11  Serial.println(" World!");  
12  delay(1000);  
13 }
```



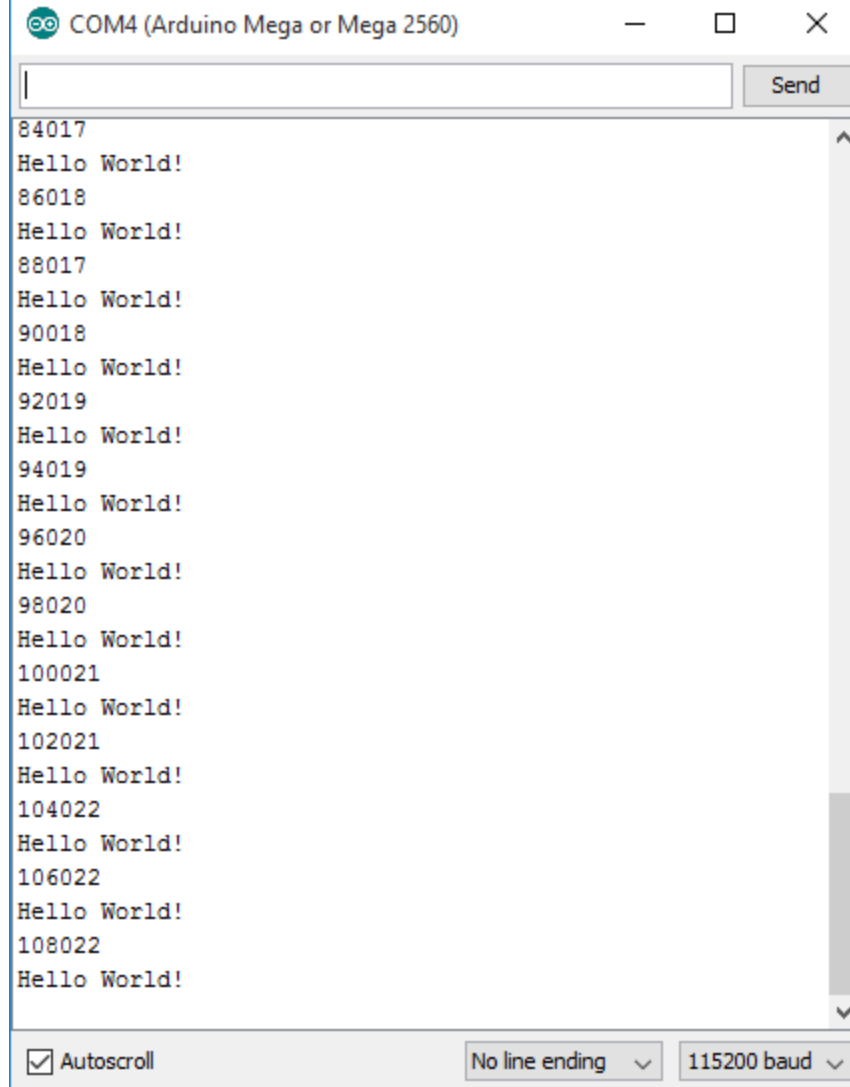
Debug

Debug?



problem?

```
1 unsigned long time;
2
3 void setup() {
4     pinMode(LED_BUILTIN, OUTPUT);
5     Serial.begin(115200);
6 }
7
8 void loop() {
9     digitalWrite(LED_BUILTIN, HIGH);
10    Serial.print("Hello");
11    delay(1000);
12    digitalWrite(LED_BUILTIN, LOW);
13    Serial.println(" World!");
14    delay(1000);
15    time = millis();
16    Serial.println(time);
17 }
```



COM4 (Arduino Mega or Mega 2560)

Send

84017
Hello World!
86018
Hello World!
88017
Hello World!
90018
Hello World!
92019
Hello World!
94019
Hello World!
96020
Hello World!
98020
Hello World!
100021
Hello World!
102021
Hello World!
104022
Hello World!
106022
Hello World!
108022
Hello World!

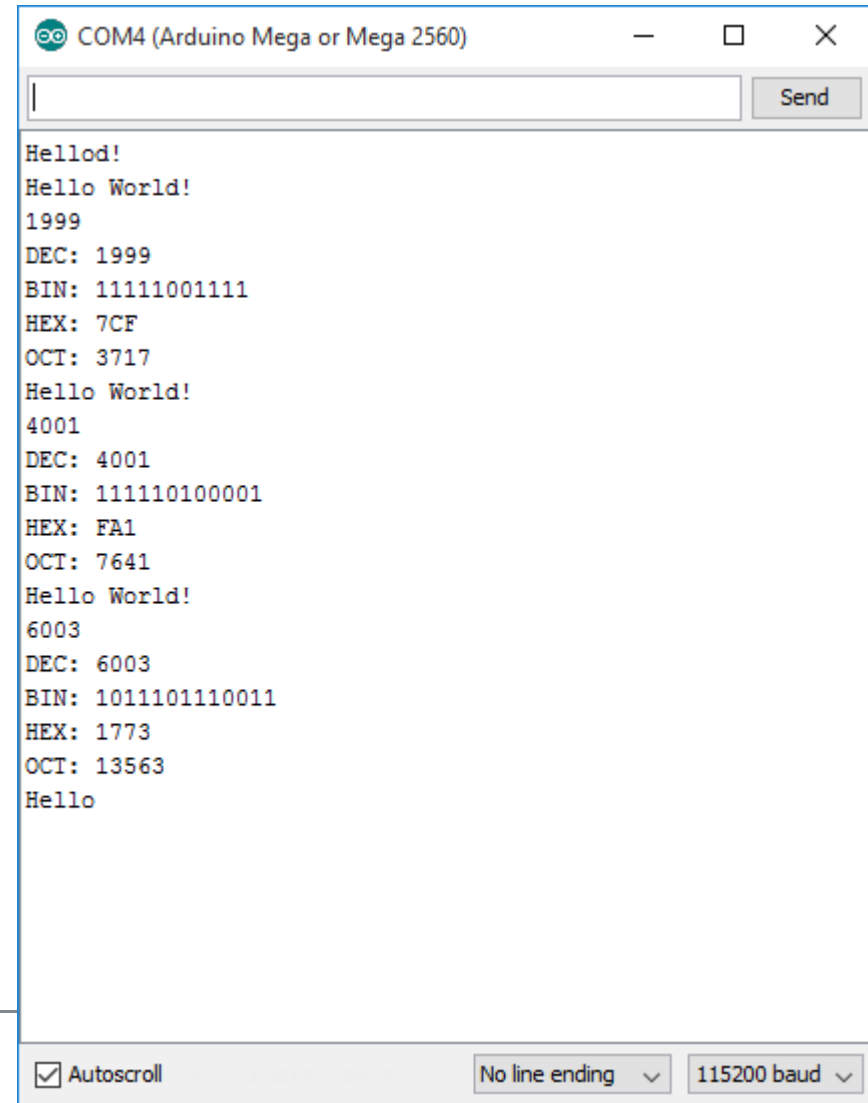
Autoscroll No line ending 115200 baud

millis()

- Visszatér a futásidővel (ms – unsigned long)
- Kb 50 nap után overflow

println()

```
15  time = millis();
16  Serial.println(time);
17  Serial.print("DEC: ");
18  Serial.println(time, DEC);
19  Serial.print("BIN: ");
20  Serial.println(time, BIN);
21  Serial.print("HEX: ");
22  Serial.println(time, HEX);
23  Serial.print("OCT: ");
24  Serial.println(time, OCT);
25 }
```



COM4 (Arduino Mega or Mega 2560)

```

Hello!
Hello World!
1999
DEC: 1999
BIN: 11111001111
HEX: 7CF
OCT: 3717
Hello World!
4001
DEC: 4001
BIN: 111110100001
HEX: FA1
OCT: 7641
Hello World!
6003
DEC: 6003
BIN: 1011101110011
HEX: 1773
OCT: 13563
Hello
```

Autoscroll No line ending 115200 baud


```
1 unsigned long time;
2
3 void setup() {
4     pinMode(LED_BUILTIN, OUTPUT);
5     Serial.begin(115200);
6 }
7
8 void loop() {
9     time=100;
10    while (Serial.available() > 0) {
11        time=Serial.parseInt();
12        Serial.println("Delay time: "+String(time));
13        if(Serial.read()=='\n'){
14            digitalWrite(LED_BUILTIN,1-digitalRead(LED_BUILTIN));
15        }
16    }
17    Serial.println(String(millis()));
18    delay(time);
19 }
```

int Serial.available()

- Visszatér a soros porton olvasásra elérhető byte-ok számával
 - Soros RX puffer
 - Max. 64 byte

long Serial.parseInt()

- 0...9, '-' karaktereken kívül kihagy mindent
- Hibás karakterek esetén 0-t ad vissza

```
1 String s;
2
3 void setup() {
4     pinMode(LED_BUILTIN, OUTPUT);
5     Serial.begin(115200);
6 }
7
8 void loop() {}
9     while (Serial.available() > 0) {
10         s=Serial.readStringUntil('\n');
11         Serial.println("");
12         Serial.println("Simon says: "+s);
13
14
15     }
16     Serial.print(".");
17     delay(250);
18 }
```