SZÉCHENYI ISTVÁN EGYETEM
GÉPÉSZMÉRNÖKI, INFORMATIKAI ÉS VILLAMOSMÉRNÖKI KAR

Labor gyakorlat – Mikrovezérlők

ATMEL AVR – ARDUINO

1. ELŐADÁS

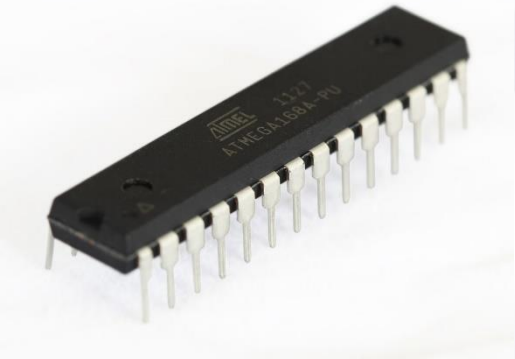
BUDAI TAMÁS

Tartalom

- **Mikrovezérlők**
  - Mikrovezérlők felépítése, működése
  - Mikrovezérlő típusok, gyártók
  - Mikrovezérlők perifériái
- **Mikrovezérlők programozása**
  - A C programozási nyelv (ismétlés)
  - ATMEL AVR mikrovezérlők programozása
  - Feladatmegoldás

Mikrovezérlő:

- Kisméretű, alacsony fogyasztású, programvezérelt digitális hálózat, avagy **számítógép**



Alkalmazási területek:

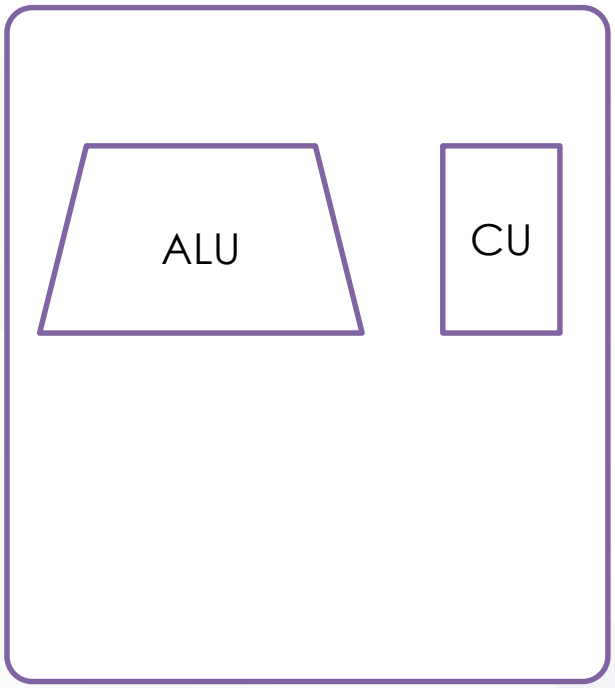
- Háztartási és szórakoztató elektronika
- Ipari elektronika
- Járműipar
- Gyógyászati eszközök
- „SMART" eszközök
- IoT
- …

Tartalom

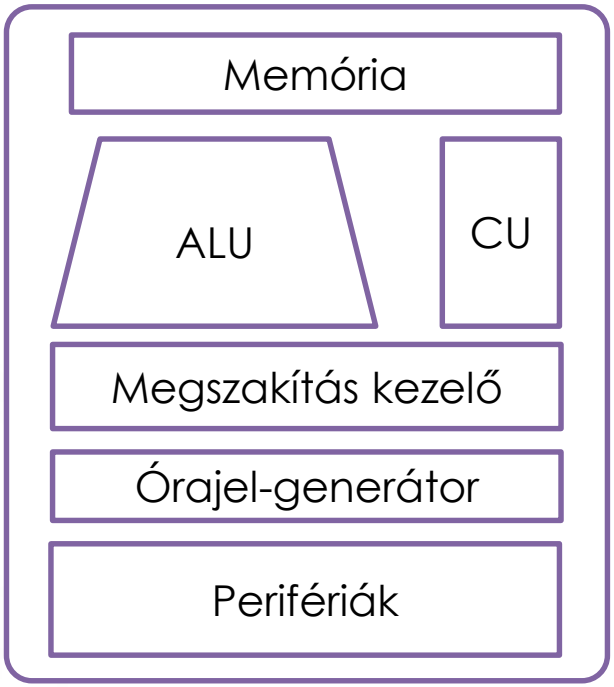
- **Mikrovezérlők**
  - Mikrovezérlők felépítése, működése
  - Mikrovezérlő típusok, gyártók
  - Mikrovezérlők perifériái
- **Mikrovezérlők programozása**
  - A C programozási nyelv (ismétlés)
  - ATMEL AVR mikrovezérlők programozása
  - Feladatmegoldás

Mikrovezérlők felépítése, működése

Mikroprocesszor



Mikrovezérlő



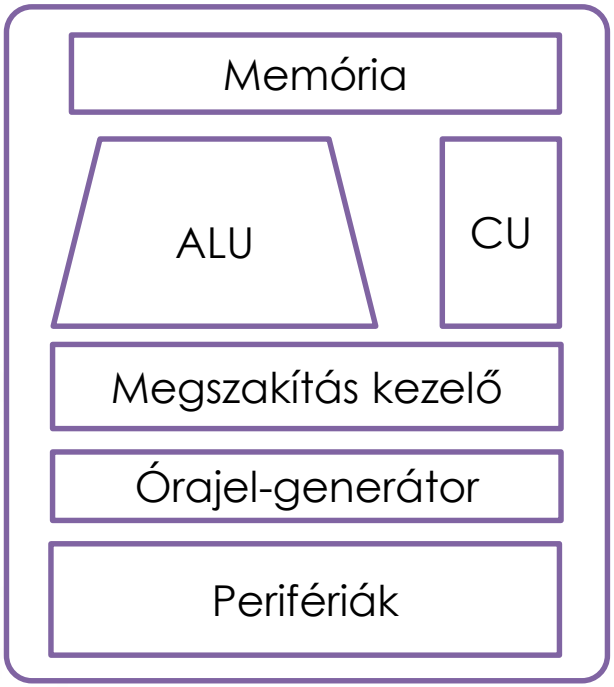
A mikrovezérlő egy teljes számítógép egy tokban

Mikrovezérlők felépítése, működése

Számítógép, tehát érvényesek rá a Neumann alapelvek:

- Elektronikus
- Bináris számrendszer
- Adat és programmemória
- Univerzális Turing-gép

Mikrovezérlő

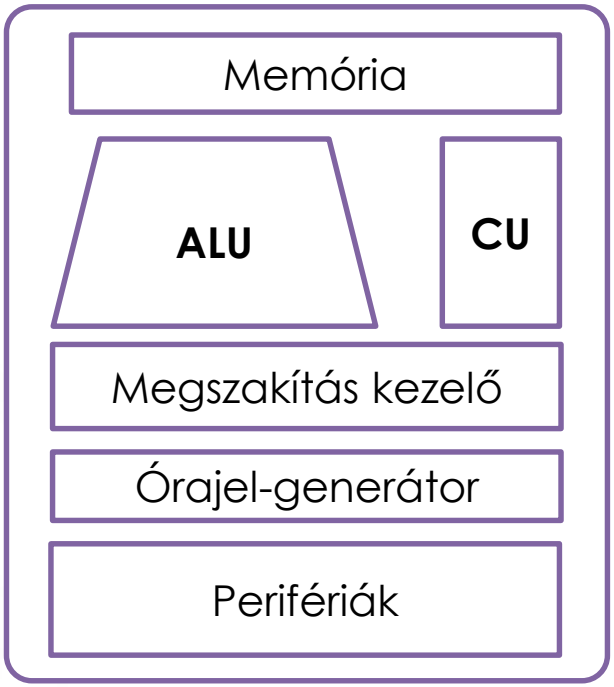


Mikrovezérlők felépítése, működése

CPU (ALU+CU):

- Egyes családoknál* közös
- Meghatározza a számábrázolást*: 8,16,32bit…

Mikrovezérlő

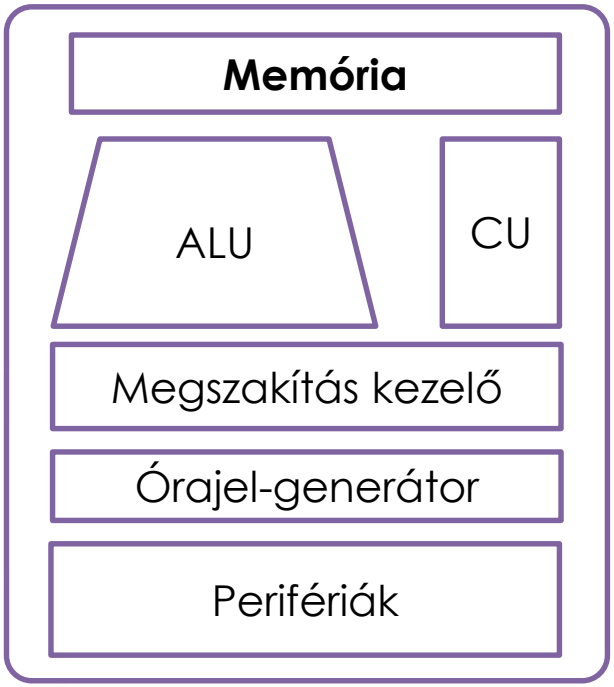


Mikrovezérlők felépítése, működése

Memória típusok:

- Program memória
  - nem újraírható
  - újraírható
- Adatmemória
  - RAM
  - ROM

Mikrovezérlő

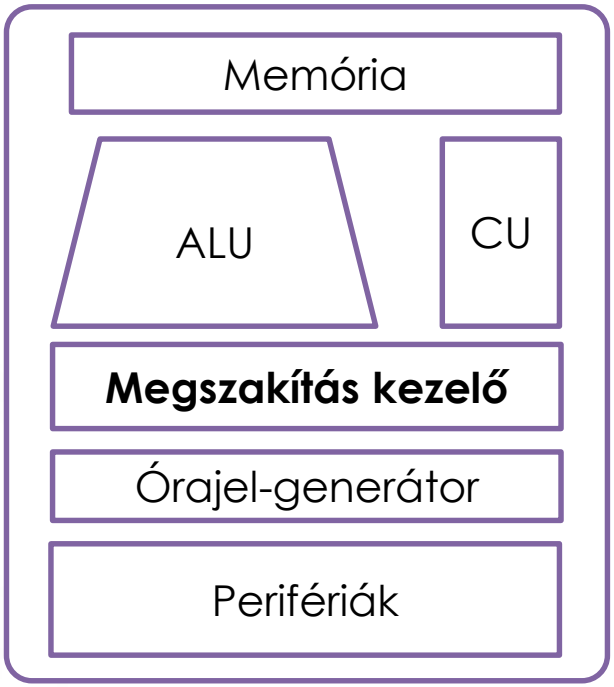


Mikrovezérlők felépítése, működése

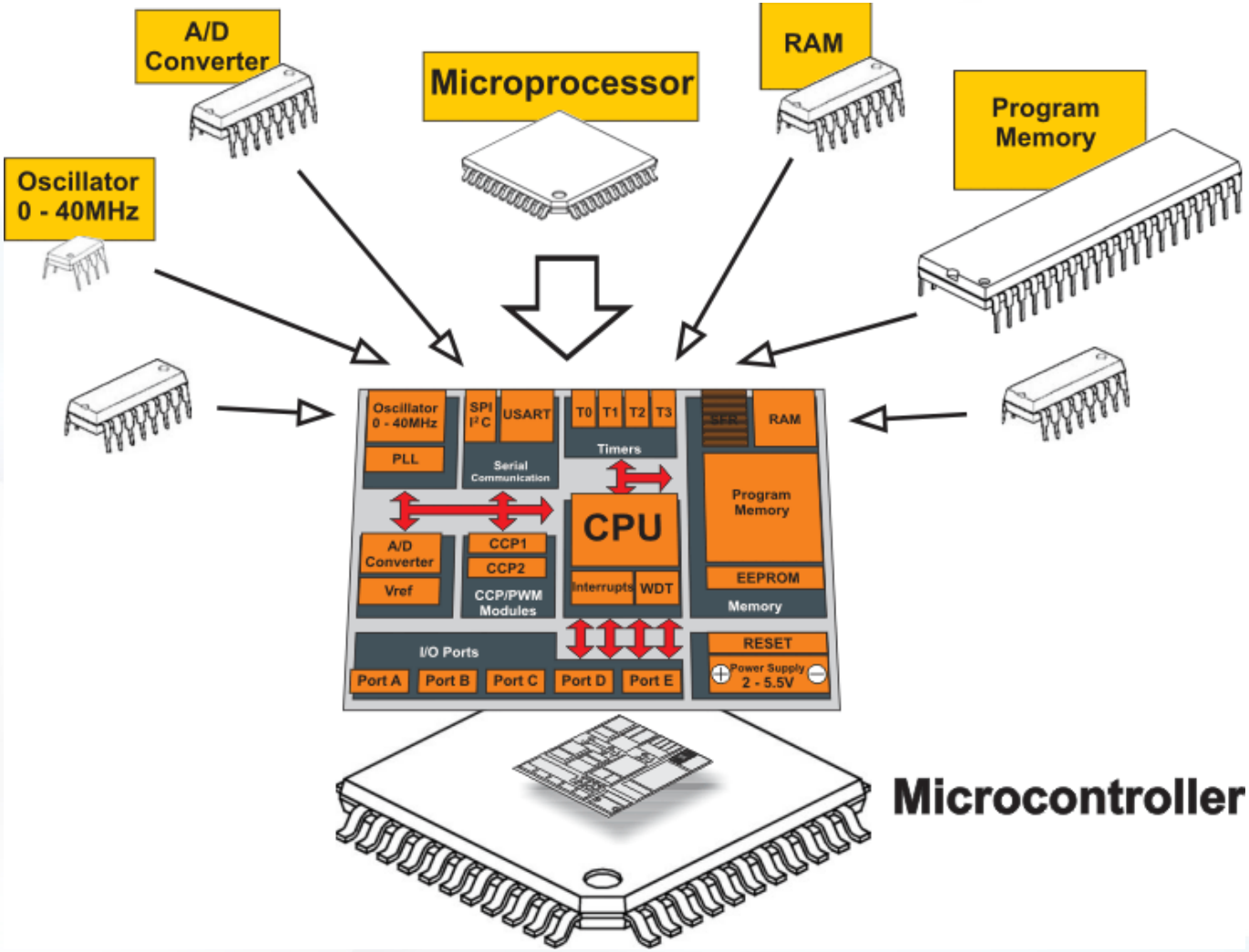
Megszakítás kezelő:

- A normál futást megszakítva rákényszeríti a rendszert, hogy azonnal hajtson végre egy speciális programrészt
- Megszakítás forrásai
  - **Belső**: időzítő, periféria
  - **Külső**: adott lábon felfutó él

Mikrovezérlő



Mikrovezérlők felépítése, működése



Mikrovezérlők felépítése, működése

Működés

Egy általános célú mikrovezérlő a **reset** esemény után a következő lépéseket hajtja végre:

1. Minden periféria alaphelyzetbe állítása
2. A **program** lépéseinek végrehajtása a programmemória elejétől (reset vector) az **órajel** szerint

Mikrovezérlők felépítése, működése

Reset

A **reset** esemény: a mikrovezérlő (újra)indítása

A reset esemény forrásai:

- Power-on reset (bekapcsolás)
- External reset (külső reset láb szintváltozása miatt)
- Watchdog system reset

Mikrovezérlők felépítése, működése

Program (Gépi kód)

- A gép által értelmezhető binárisan kódolt **utasítások** sorozata

Forráskód

- Majd a programozásnál…

Mikrovezérlők felépítése, működése

Órajel

- Mivel a mikrovezérlő egy digitális hálózat ezért a működéséhez szükség van **órajelre.**
- Jelalak: **négyszögjel**
- mértékegysége: 1/s vagyis **Hertz**
- Tipikus nagyságrendje napjainkban: 1MHz-100MHz

Forrás	Pontosság	Költség
RC oszcillátor	Alacsony	Legolcsóbb
Kerámia rezonátor	Közepes	Olcsó
Rezgőkristály	Magas	Drága

Mikrovezérlők felépítése, működése

Órajel ciklus, ciklusidő

- **Ciklusidő**: Az órajel két egymást követő le/felfutó éle közt eltelt idő (vagyis a négyszögjel **periódusideje**)
- Utasítások jellemzése: hány órajelciklus szükséges a végrehajtáshoz

  Pl.:

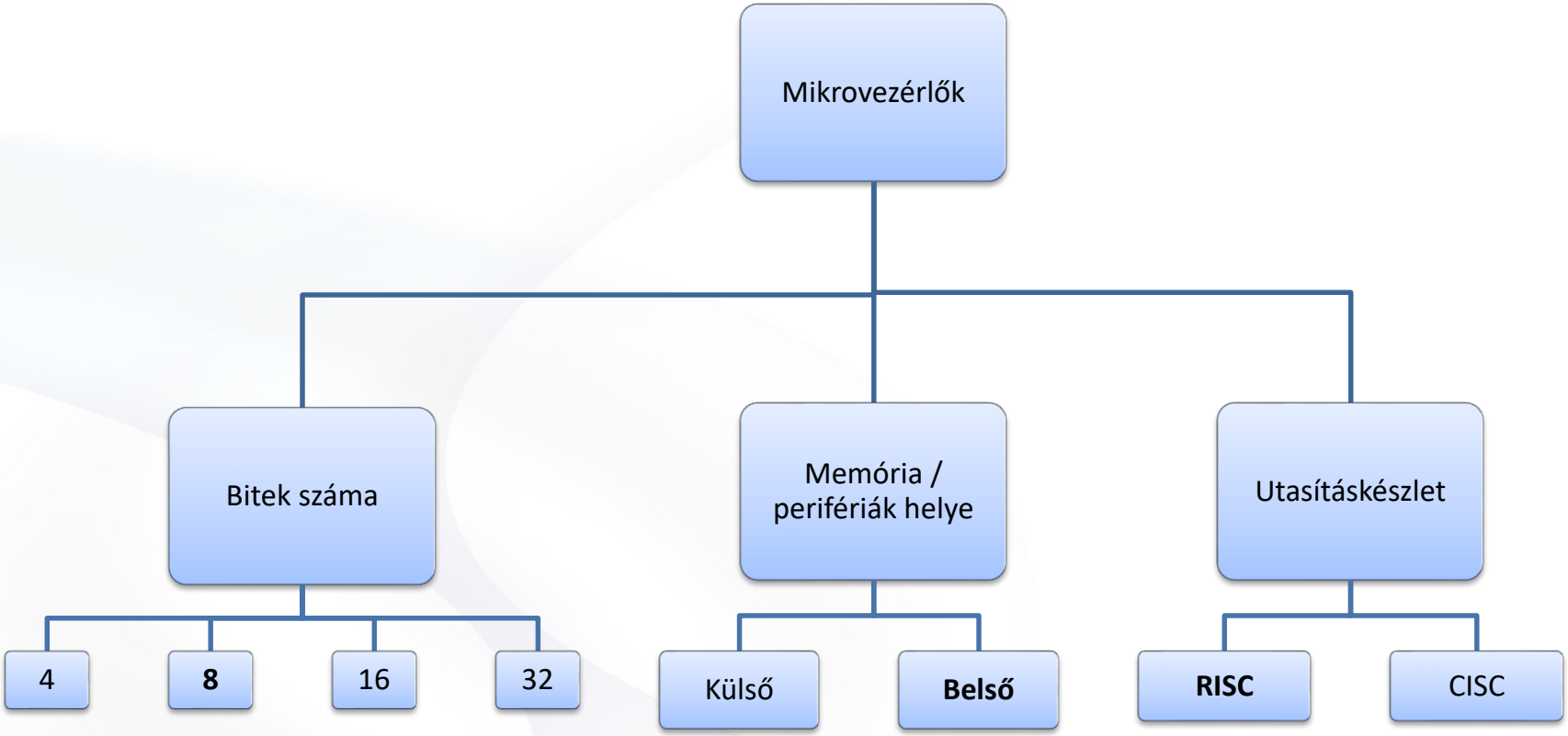
  - Feltétel nélküli ugrás – 1 ciklus
  - Összeadás – 2 ciklus

  (Minél kevesebb, annál gyorsabb)

Tartalom

- **Mikrovezérlők**
  - Mikrovezérlők felépítése, működése
  - Mikrovezérlő típusok, gyártók
  - Mikrovezérlők perifériái
- **Mikrovezérlők programozása**
  - A C programozási nyelv (ismétlés)
  - ATMEL AVR mikrovezérlők programozása
  - Feladatmegoldás

Mikrovezérlő típusok, gyártók



Mikrovezérlő típusok, gyártók

Gyártók

- Atmel
- Infineon
- Intel
- Microchip
- Motorola
- National Instruments
- Parallax
- Texas instruments
- Xilinx
- …

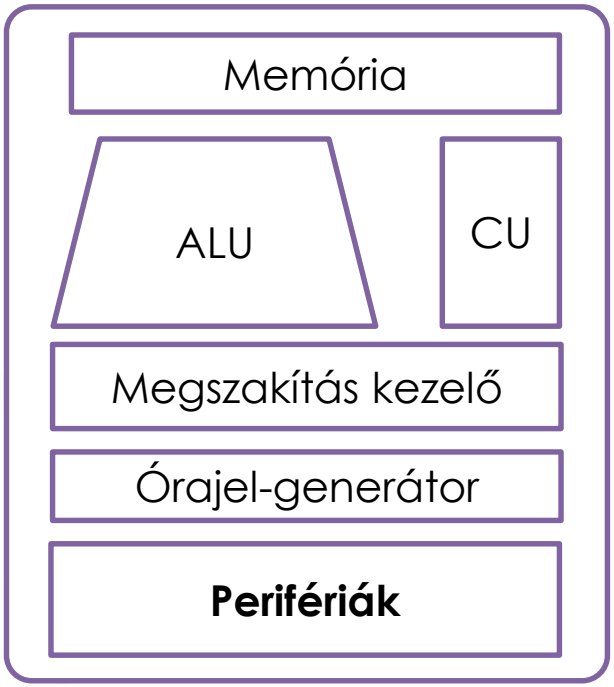
Tartalom

- **Mikrovezérlők**
  - Mikrovezérlők felépítése, működése
  - Mikrovezérlő típusok, gyártók
  - Mikrovezérlők perifériái
- **Mikrovezérlők programozása**
  - A C programozási nyelv (ismétlés)
  - ATMEL AVR mikrovezérlők programozása
  - Feladatmegoldás

Mikrovezérlők perifériái

- **Tipikus perifériák**
  - Digitális be/kimenetek
  - A/D konverter
  - D/A konverter
- **Speciális perifériák**
  - BUS illesztők*:
    - SPI
    - UART (RS232)
    - I^2C
    - CAN
    - …
  - érintés-érzékelő
  - PWM kimenet

Mikrovezérlő



*Nem feltétlenül szükséges minden buszhoz (bitbang)

Mikrovezérlők perifériái

- **Digitális bemenet:**
  - Egy bemenet egy láb
  - Egy bit reprezentálja a logikai szintet:
    - 0 = alacsony
    - 1 = magas
- **Digitális kimenet:**
  - Egy kimenet egy láb
  - Ugyan az, mint a bemenet, csak itt a bitet írjuk és nem olvassuk

Mikrovezérlők perifériái

- **A/D átalakító:**
  - Bemenet
  - Működés:
    - Mintavételezés
    - Kvantálás
  - Jellemzői:
    - Felbontás (hány bites)
    - sebesség
  - Az analóg jel amplitúdója a föld és egy referenciapotenciál közt értelmezett (ez általában a tápfeszültség) és egy változóból kiolvasható

Mikrovezérlők perifériái

- **D/A átalakító:**
  - Kimenet
  - Jellemzői:
    - Felbontás (hány bites)
    - sebesség
  - Az analóg jel amplitúdója a föld és egy referenciapotenciál közt értelmezett (ez általában a tápfeszültség)
  - Logikája azonos az A/D-vel, csak itt beírjuk a kívánt szintet és nem kiolvassuk

Mikrovezérlők perifériái

- **BUSZ illesztők:**
  - Be/kimenet
  - Annyi lábat használ, amennyit az adott BUSZ megkíván:
    - Pl.: CLK, RX, TX
  - Működését az adott BUSZ típusa határozza meg
  - Tartozhat hozzá speciális memóriarész (pl. fogadó buffer)

Tartalom

- **Mikrovezérlők**
  - Mikrovezérlők felépítése, működése
  - Mikrovezérlő típusok, gyártók
  - Mikrovezérlők perifériái
- **Mikrovezérlők programozása**
  - A C programozási nyelv (ismétlés)
  - ATMEL AVR mikrovezérlők programozása
  - Feladatmegoldás

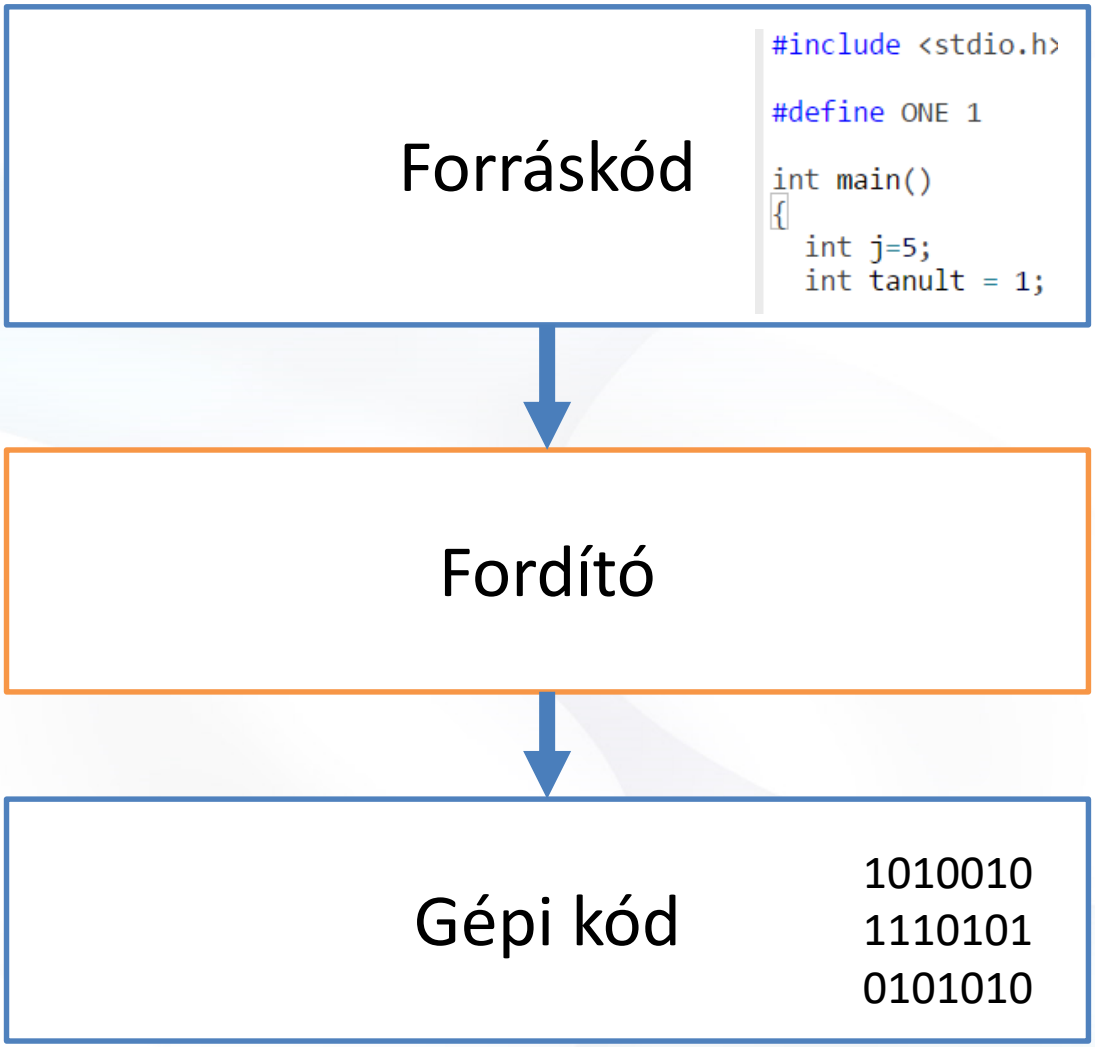
Program (Gépi kód)

- A gép által értelmezhető binárisan kódolt **utasítások** sorozata

Forráskód

- Az egyszerűbb kezelhetőség érdekében adjunk rövid nevet az egyes utasításoknak
- Ember által értelmezhető, gép által **nem**;
- Ahhoz, hogy futtatható legyen **fordításra** van szükség!

Mikrovezérlők programozása



Egy mikrovezérlő bármilyen nyelven programozható, ha van az adott nyelvhez és vezérlőhöz fordítóprogram.

- **Programnyelvek**
  - Alacsony szintű
    - Assembly
  - Magas szintű
    - **C**, C++, Python, Java…

Tartalom

- **Mikrovezérlők**
  - Mikrovezérlők felépítése, működése
  - Mikrovezérlő típusok, gyártók
  - Mikrovezérlők perifériái
- **Mikrovezérlők programozása**
  - A C programozási nyelv (ismétlés)
  - ATMEL AVR mikrovezérlők programozása
  - Feladatmegoldás

A C programozási nyelv (ismétlés) tartalom

- Alapok
- Változók, adattípusok
- Operátorok
- Elágazások, Feltételes utasítások
- Ciklusok
- Függvények

A forráskód:

- Egyszerű szövegfájl
- Ember által értelmezhető
- Utasítások sorozata
- **Kötött formátum**

Példa

```c
#include <stdio.h>

#define ONE 1

int main()
{
  int j=5;
  int tanult = 1;

  if(tanult){
    printf("Az on erdemjegye: %d\n", j);
  }
  else{
    printf("Az on erdemjegye: %d\n", ONE);
  }

  return 0;
}

```

A C programozási nyelv (ismétlés) Alapok

`#include` direktíva:

- más forráskódrészeket tudunk beágyazni
- ezek erősen függenek a használt környezettől

`main()` függvény:

- minden programban pontosan egy darabnak kell lennie
- itt kezdődik a programunk végrehajtása

A C programozási nyelv (ismétlés)
Változók, adattípusok

A legkisebb egység: 1 bit 

8 bit = **1 byte** 

Változótípusok

void	„semmi” (ezt is jelölni kell valahogy)
bool	logikai (igaz/hamis)
char	egy karakter (1 byte)
int	(előjeles) egész szám
unsigned int	egész szám
float	lebegőpontos szám
double	dupla pontosságú float
string	char-okból álló tömb

Példa

```
1   int a;
2   float b;
3   double c;
4
5   a=5;
6   b=3.14;
7   c = 6.53;
8
9   int n = 0;
10  float x = 1.2345;
```

A C programozási nyelv (ismétlés) Változók, adattípusok

Tömbök:

- Egy adott adattípusból álló több elemű halmaz
- A memóriában folytonosan helyezkedik el
- A tömb méretét előre meg kell mondanunk*

A C programozási nyelv (ismétlés)
Változók, adattípusok

Tömbök:

- Egy adott adattípusból álló több elemű halmaz
- A memóriában folytonosan helyezkedik el
- A tömb méretét előre meg kell mondanunk*

```
1  int ints[6];
2  int pins[] = {2, 4, 8, 3, 6};
3  int vals[6] = {2, 4, -8, 3, 2};
4
5  printf("%d", pins[0]);
6  printf("%d", vals[3]);
```

A C programozási nyelv (ismétlés)
Operátorok

Operátorok:

- Aritmetikai: `+ - * / % ( )`
- Összehasonlító: `== != < <= > >=`
- Logikai: `&& || !`
- Bitenkénti: `& | ^ ~ << >>`
- Helyben módosító: `++ -- += -= *= /= &= |=`
- Egyéb: `sizeof`

Példa

```
1   int a,b,c;
2   int d;
3   int e;
4
5   a=5;
6   b=2;
7   c=3;
8
9   d = a+b*c;
10
11  d= (a-b)*c;
12
```

Elágazások:

- if, if-else
- switch case

A C programozási nyelv (ismétlés) Elágazások, feltételes utasítások

If-else

```
1  int x=0;
2  int val = 700;
3
4  if (val < 500)
5  {
6    x = 1;
7  }
8  else
9  {
10   x = 0;
11 }
```

Switch-case

```
1  int kerdes;
2  kerdes = 2;
3
4    switch (kerdes) {
5      case 1:
6        printf("Elso");
7        break;
8      case 2:
9        printf("Masodik");
10       break;
11     default:
12       printf("Egyik sem");
13   }
```

Ciklusok:

- for
- while
- do-while
- Ciklusvezérlés

A C programozási nyelv (ismétlés) Ciklusok

for

```
1  int i;
2  i = 0;
3
4  for (; i <= 255; i++){
5    printf("%d ", i);
6  }
```

Tömörebben:

```
1  for (int i=0; i <= 255; i++){
2    printf("%d ", i);
3  }
```

A C programozási nyelv (ismétlés) Ciklusok

while

```
1  int var = 0;
2  while(var < 200){
3    printf("%d", var);
4    var++;
5  }
```

```
1  int var = 400;
2  while(var > 200){
3    printf("%d", var);
4    var--;
5  }
```

do-while

```
1  int x = 0;
2  do
3  {
4    printf("Ertek: %d",x);
5    x++;
6
7  } while (x < 100);
```

Vezérlő utasítások:

- break
- continue
- return

A C programozási nyelv (ismétlés) Ciklusok

- break

```
1  int var = 0;
2  while(var < 200){
3
4    if(var>10){
5        printf("elertuk a limitet!");
6        break;
7    }
8
9    var++;
10 }
```

A C programozási nyelv (ismétlés) Ciklusok

- break

```
1  int kerdes;
2  kerdes = 2;
3
4    switch (kerdes) {
5      case 1:
6        printf("Elso");
7        break;
8      case 2:
9        printf("Masodik");
10       break;
11     default:
12       printf("Egyik sem");
13   }
```

A C programozási nyelv (ismétlés) Ciklusok

- continue

```
1  int var = 0;
2  while(var < 200){
3    var++;
4
5    if((var%2) == 0){
6        continue;
7    }
8
9    printf("%d ",var);
10
11 }
```

A C programozási nyelv (ismétlés) Ciklusok

- return

```c
if(err != 0){
    printf("HIBA!");
    return 1;
}
```

```c
#include <stdio.h>

#define ONE 1

int main()
{
  int j=5;
  int tanult = 1;

  if(tanult){
    printf("Az on erdemjegye: %d\n", j);
  }
  else{
    printf("Az on erdemjegye: %d\n", ONE);
  }

  return 0;
}

```

Függvények

- mint a matematikában:
  - paraméter, visszatérési érték
- programszervezésre használhatók

Példa: y = sin(x)

Példa

```
1  #include <stdio.h>
2  #include <math.h>
3
4  float sinfok(int f){
5      return sin(f*M_PI/180);
6  }
7
8
9  int main(){
10     int x = 30;
11     printf("%f\n", sinfok(x));
12 }
```

A C programozási nyelv (ismétlés) Összefoglalás

Összefoglalás:

```
= + - * /  % ( ) == != < <= > >= ++ -- += -=
*= /= &= |= sizeof bool int float [] {} ;
#include
```

A C programozási nyelv (ismétlés) Összefoglalás

Mit hagytunk ki:

- előfordító direktívák, makrók
- mutatók, mutató aritmetika
- dinamikus memóriakezelés
- saját adattípusok, struktúrák
- függvénykönyvtárak
- hibakeresés
- …

A C programozási nyelv (ismétlés) Összefoglalás

Ajánlott irodalom:

- http://www.tutorialspoint.com/cprogramming/index.htm
- B. W. Kernighan - D. M. Ritchie : A C programozási nyelv
- Takács Gábor: Programozás villamosmérnököknek

- http://www.maxwell.sze.hu/~budait

SZÉCHENYI ISTVÁN EGYETEM
GÉPÉSZMÉRNÖKI, INFORMATIKAI ÉS VILLAMOSMÉRNÖKI KAR

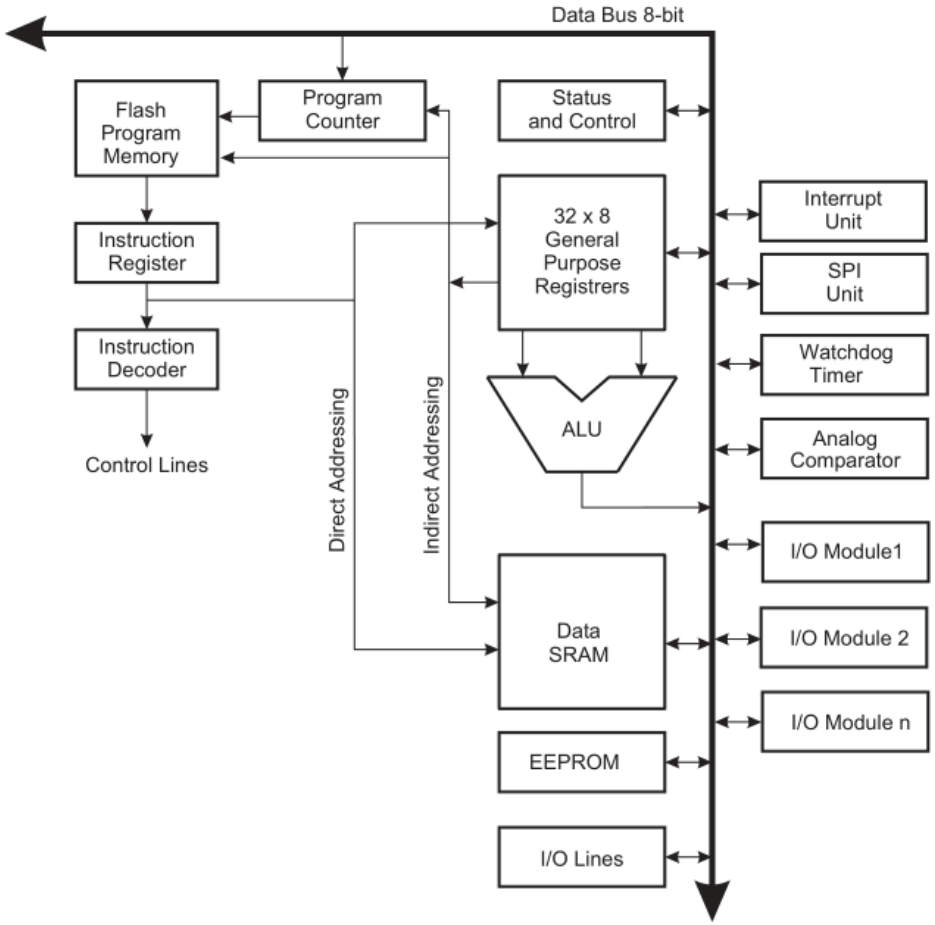
Labor 2 – Mikrovezérlők

ATMEL AVR – ARDUINO

2. FELADATMEGOLDÁS

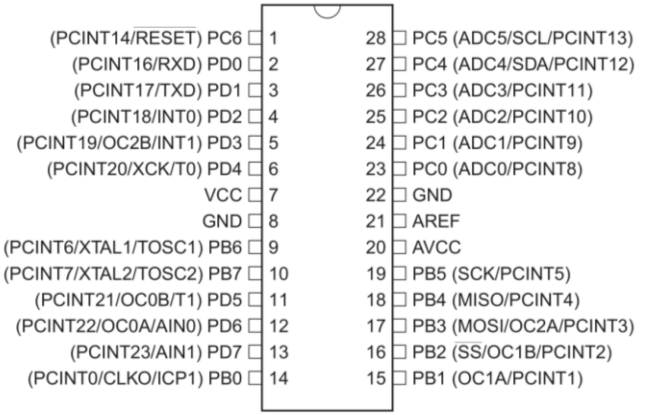
BUDAI TAMÁS

Az AVR architektúra



ATMEGA 328

- 8 bites
- RISC
- 2 db 8 bites időzítő (számláló)
- 1 db 16 bites időzítő
- 6db PWM csatorna
- 6 vagy 8 db 10 bites ADC
- USART, SPI, I^2C interfészek
- Megszakítás és ébresztés lehetőség szintváltozásra
- …



Általános programstruktúra

```
8   #include<avr/io.h>
9
10  int main()
11  {
12      DDRC = 0xFF;     // PortC legyen kimenet
13
14      while(1)
15      {
16          PORTC = 0x00;
17          PORTC = 0x01;
18          PORTC = 0x02;
19          PORTC = 0x04;
20          PORTC = 0x08;
21          PORTC = PORTC;
22      }
23      return 0;
24  }
```

Szükséges header fájlok beszúrása

Inicializálás: Kezdeti értékek, konfigurációs bitek beállítása

Főprogram: Az adott feladat végrehajtása, folyamatosan (végtelen ciklus).

Tartalom

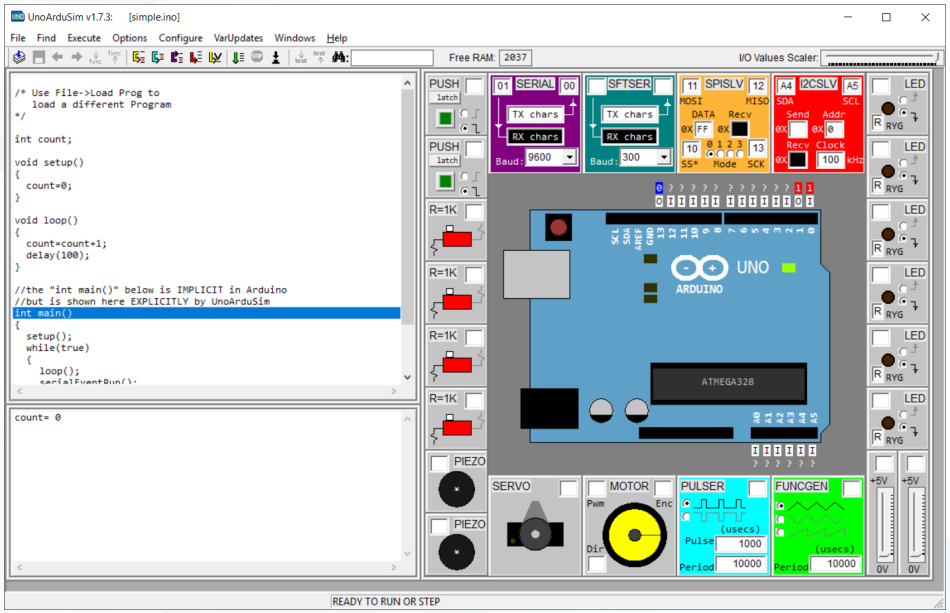
- **Mikrovezérlők**
  - Mikrovezérlők felépítése, működése
  - Mikrovezérlő típusok, gyártók
  - Mikrovezérlők perifériái
- **Mikrovezérlők programozása**
  - A C programozási nyelv (ismétlés)
  - ATMEL AVR mikrovezérlők programozása
  - Az UnoArduSim használata
  - Feladatmegoldás

Az UnoArduSim használata

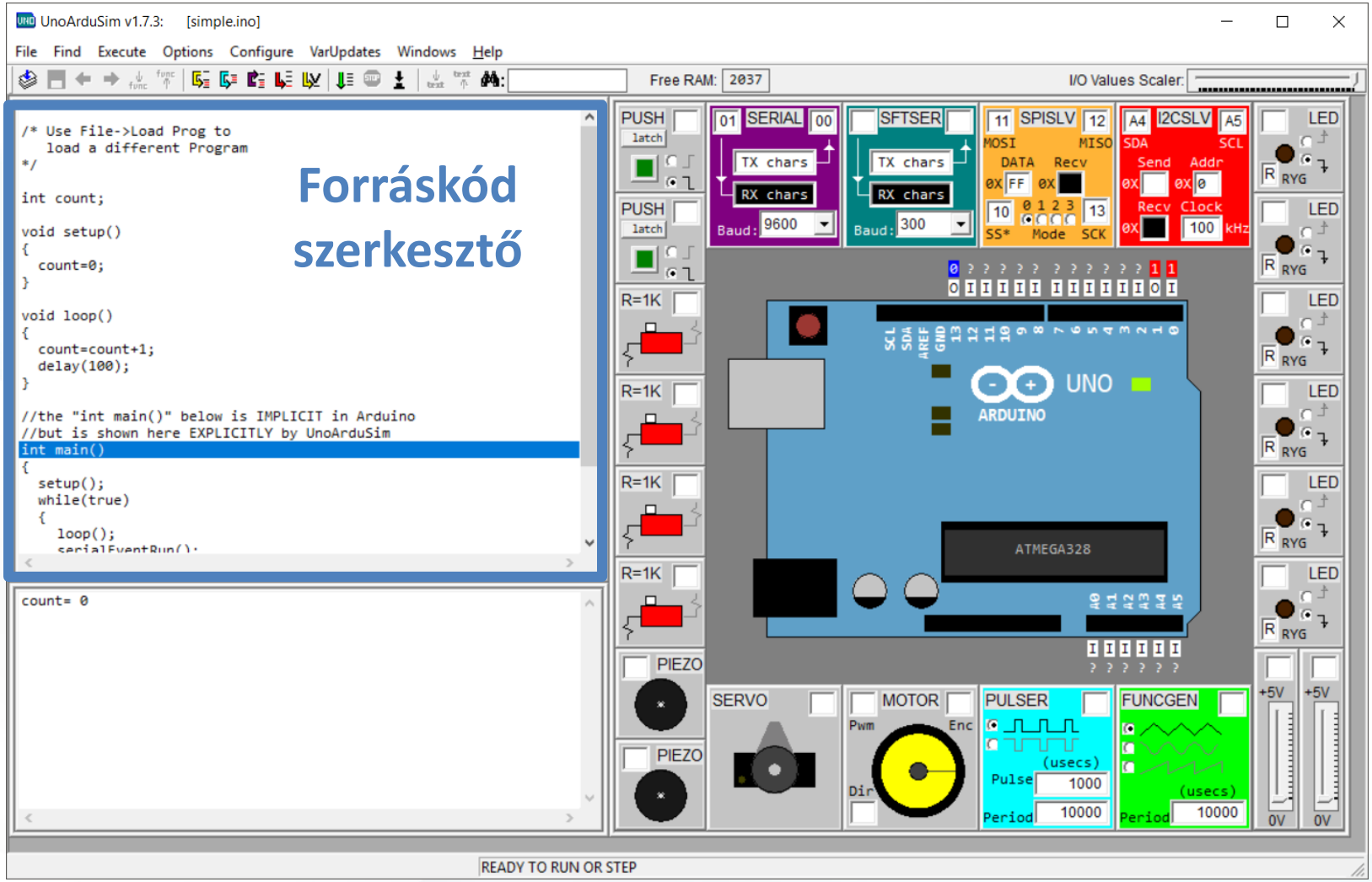
UnoArduSim

https://www.sites.google.com/site/unoardusim/home

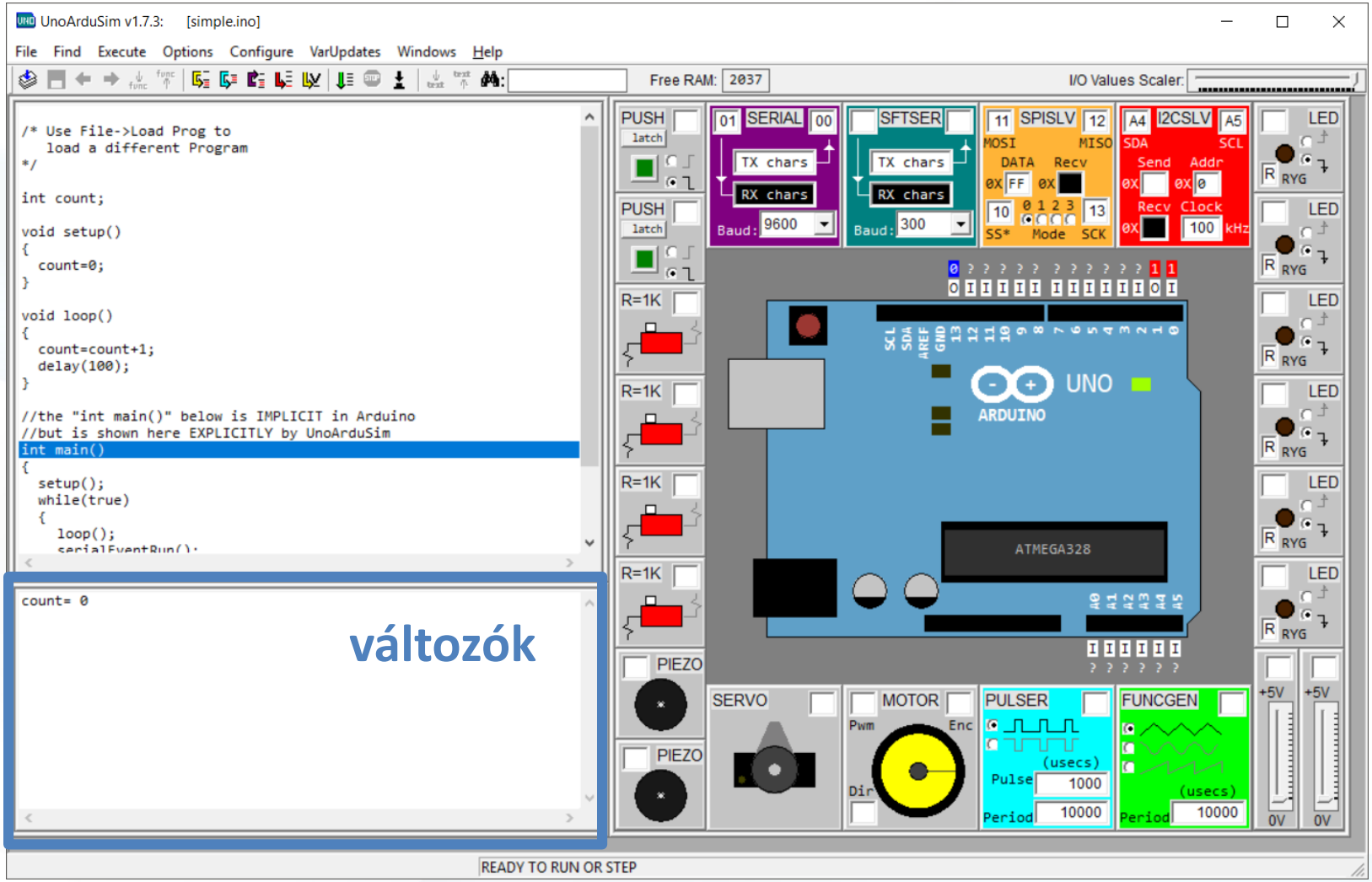
- Ingyenesen használható
- Virtuális Arduino UNO



Az UnoArduSim használata

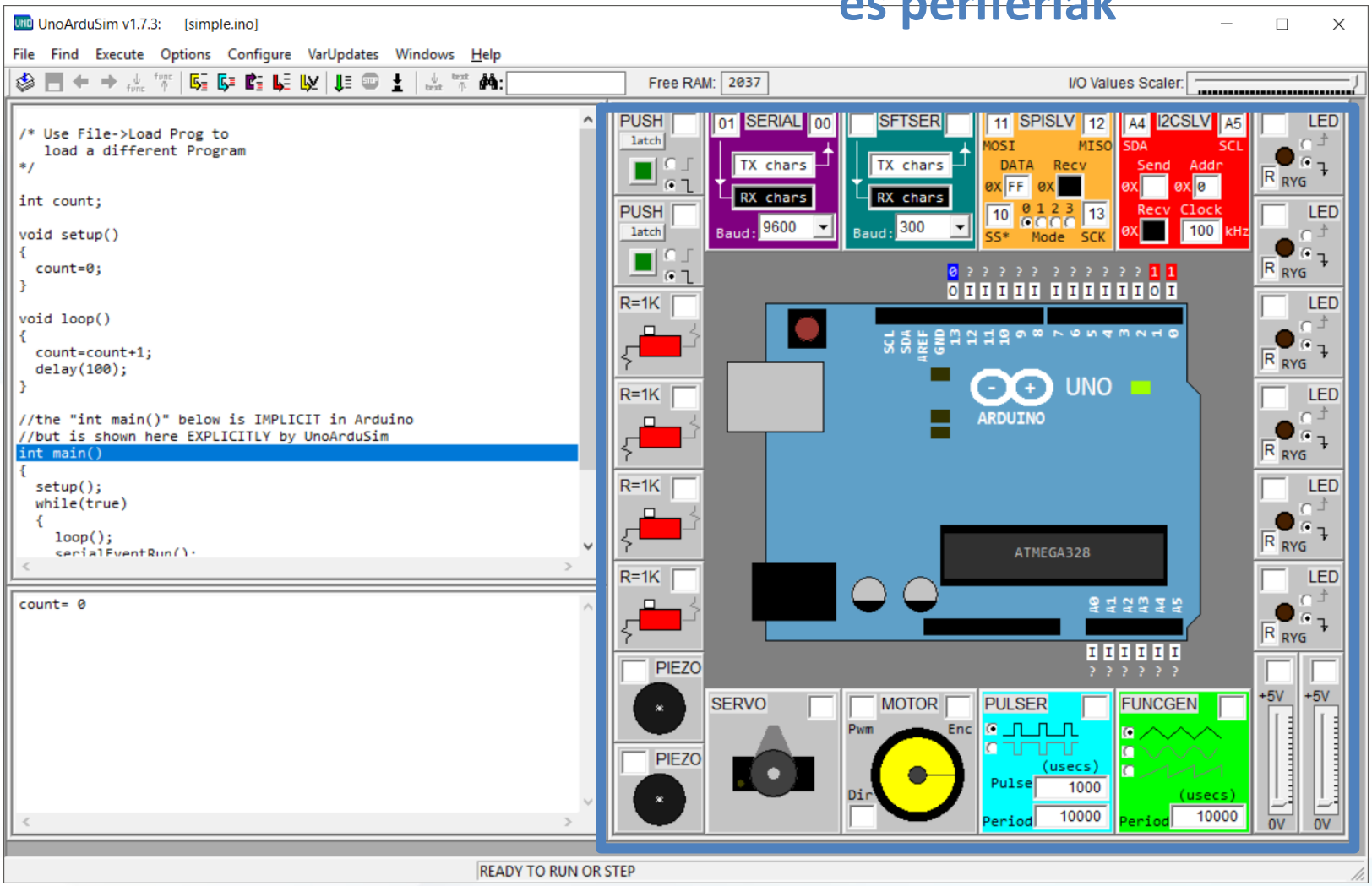


Az UnoArduSim használata



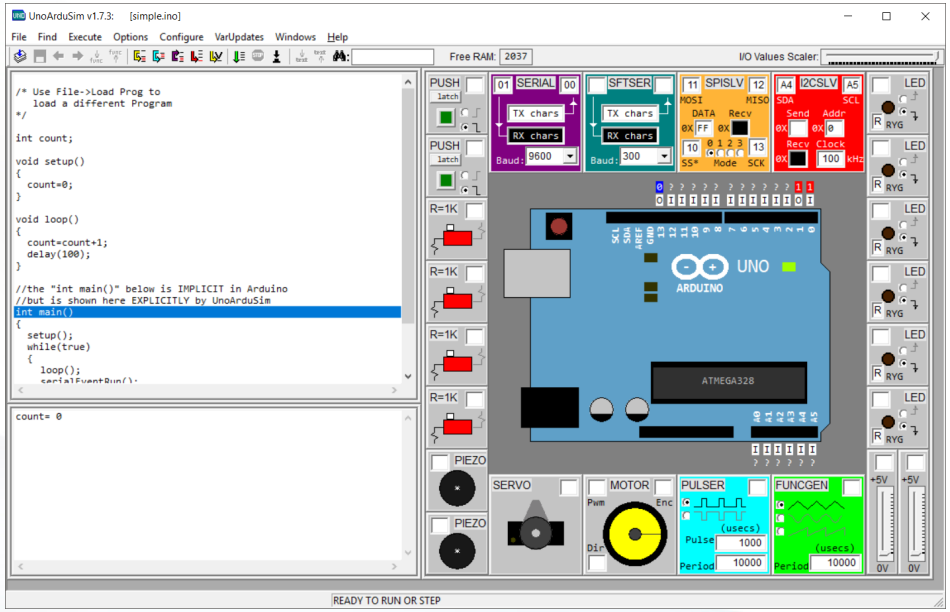
Az UnoArduSim használata

Virtuális Arduino és perifériák



A program használata

- 1. forráskód szerkesztése
- 2. virtuális perifériák behuzalozása
- 3. futtatás / hibakeresés
- 4. GOTO 1.



1. forráskód szerkesztése

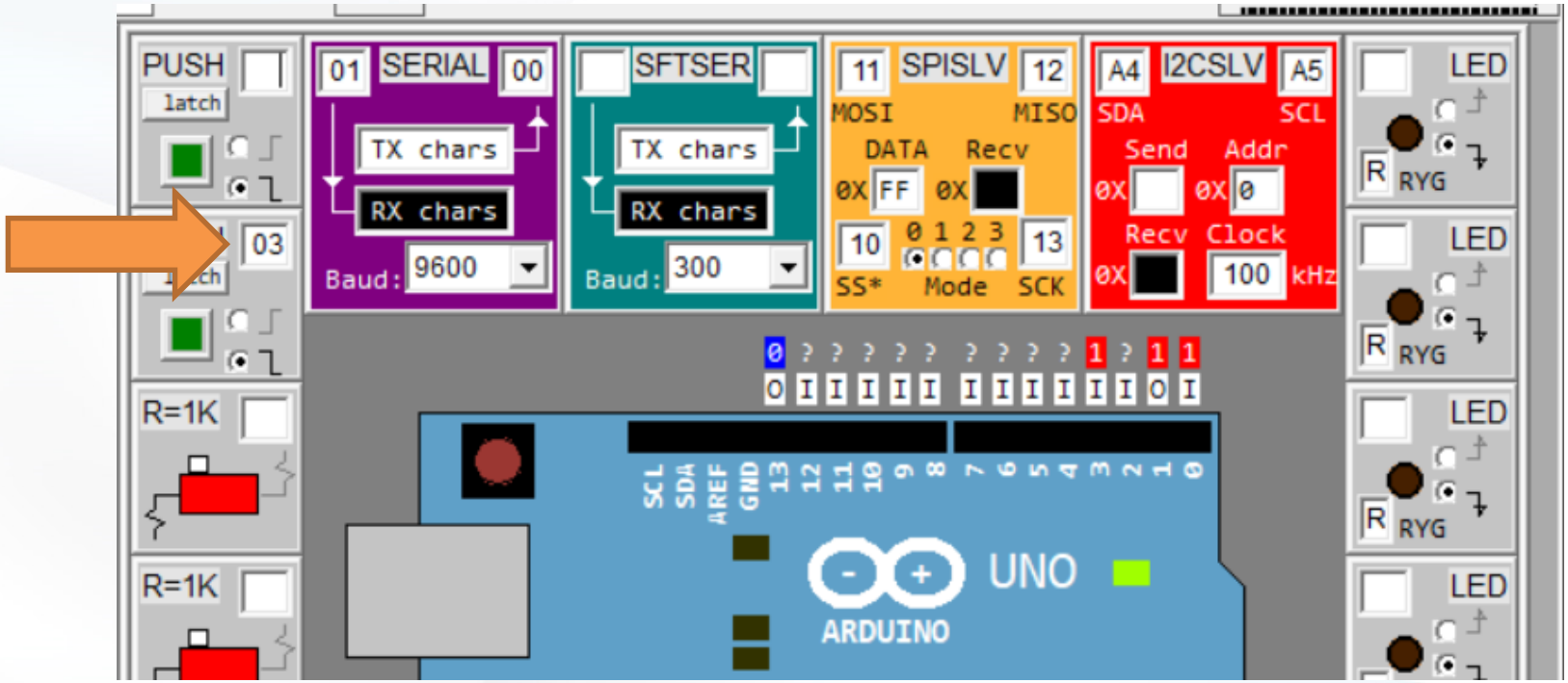
- a főablakban
- a C nyelv szabályait betartva

2. virtuális perifériák behuzalozása

- a periféria melletti szövegmezőbe beírva az arduino be/kimenet számát
- **FONTOS**: mindig 2 számjegyet kell beírni (ha a láb száma egy számjegyű, egészítsük ki egy 0-val)

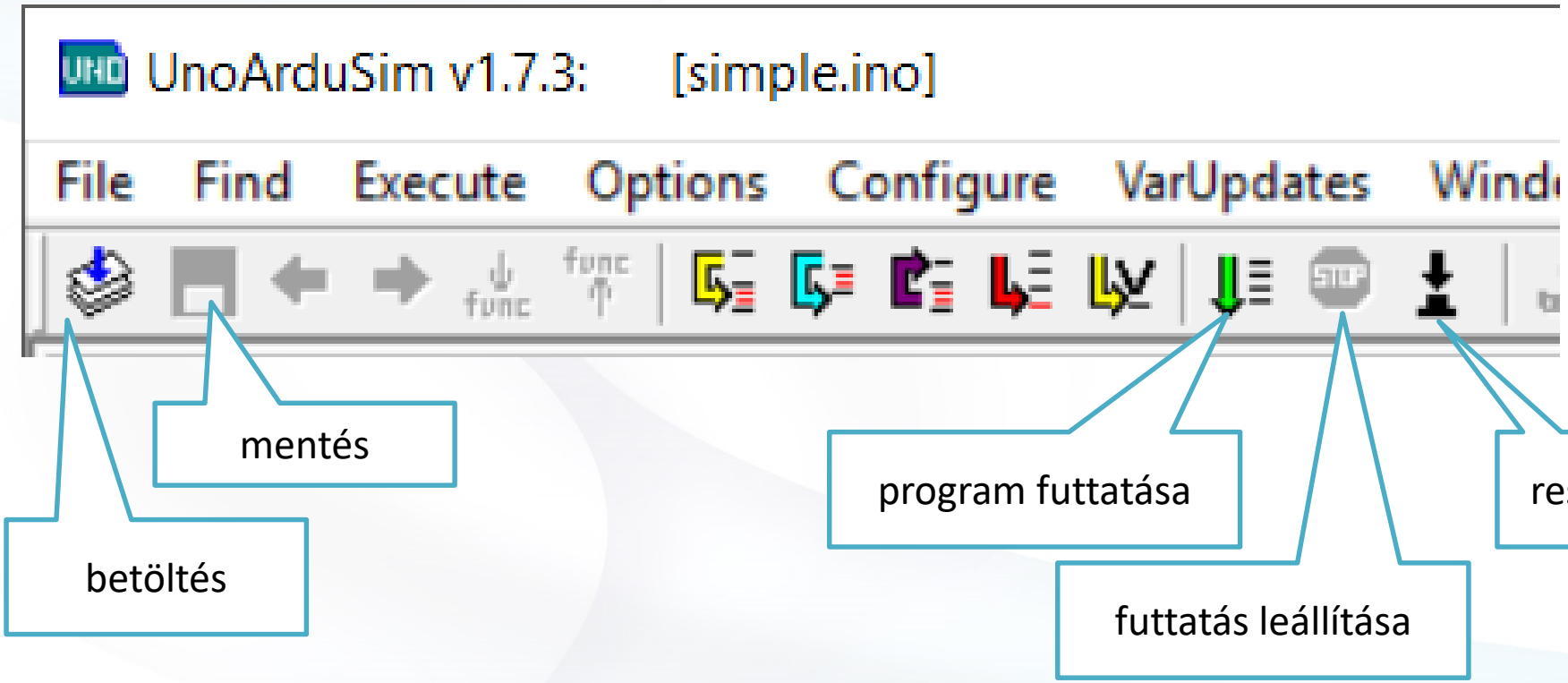
2. virtuális perifériák behuzalozása

- példa: a 2. nyomógombot a 3-as lábra:



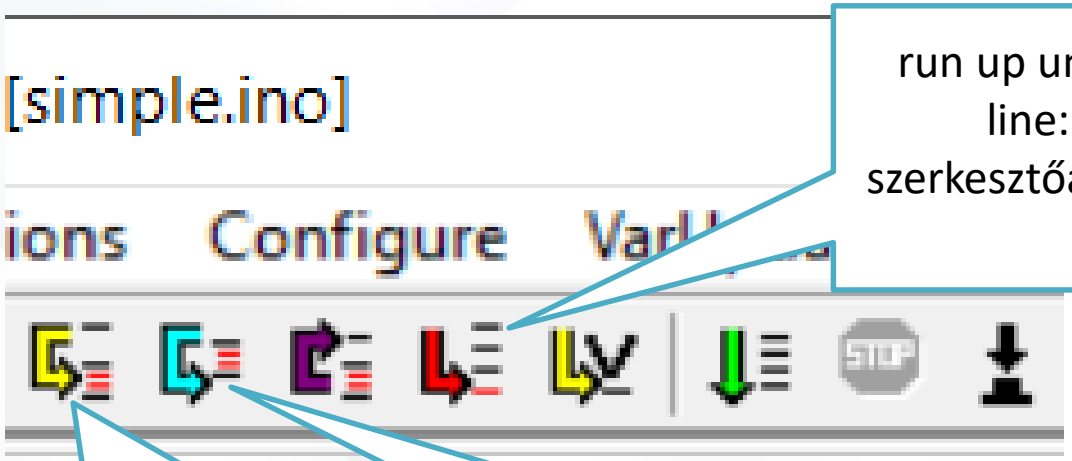
3. futtatás / hibakeresés

a megszokott IDE eszközökkel:



3. futtatás / hibakeresés

hibakereséshez:



run up until highlighted line: futtatás a szerkesztőablakban kijelölt sorig

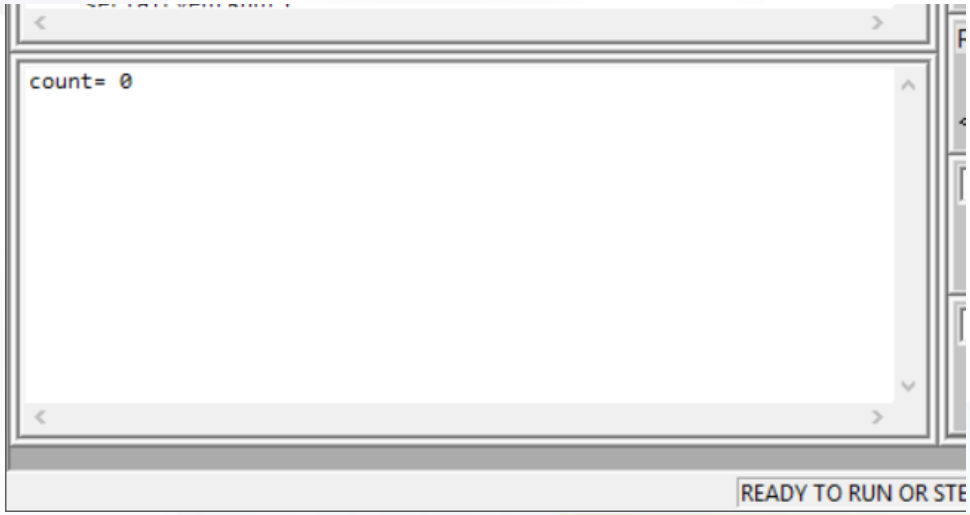
step into: program léptetése (belépés a függvényekbe)

step over: program léptetése (függvények átlépése egy lépésben)

3. futtatás / hibakeresés

hibakereséshez:

az alsó, „változók” ablakban a definiált változók értéke látható, amely futás közben folyamatosan frissül



FELADATMEGOLDÁS

1. led villogtatás
2. logikai operátorok
3. analóg bemeneti érték beolvasása (A/D)
4. dc motor fordulatszám vezérlése

…