

# Labor 2 – Mikrovezérlők

## ATMEL AVR - ARDUINO

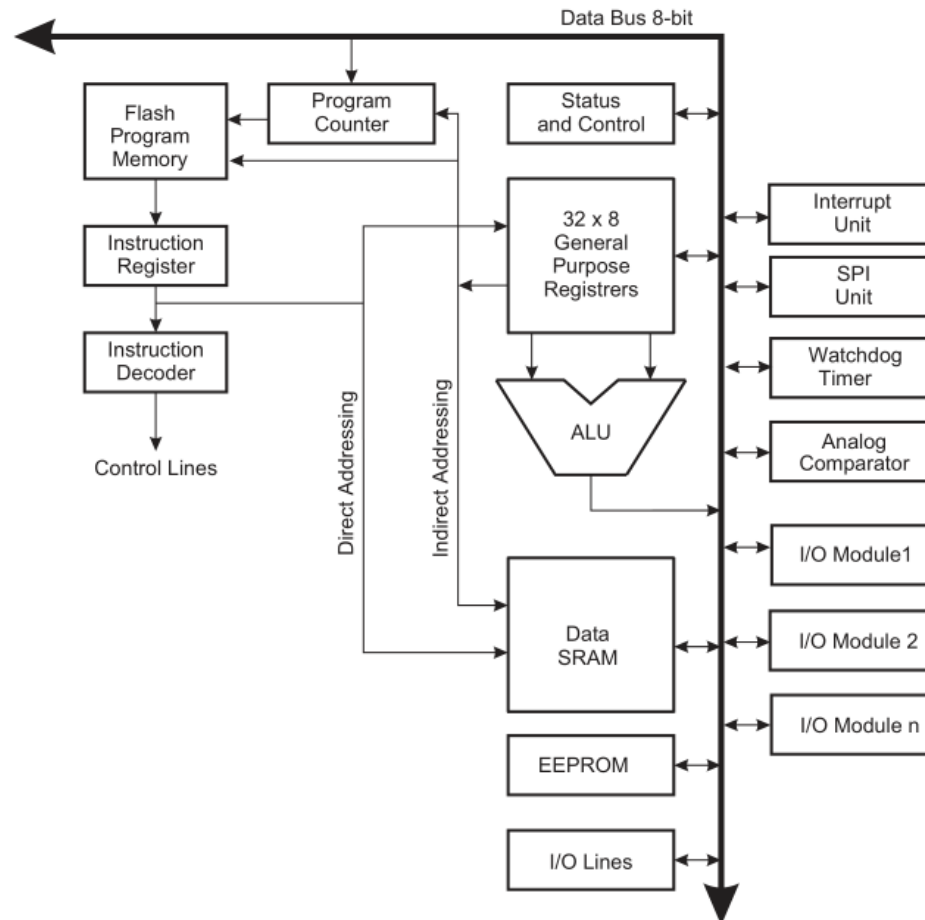
---

BUDAI TAMÁS

2015. 09. 06.

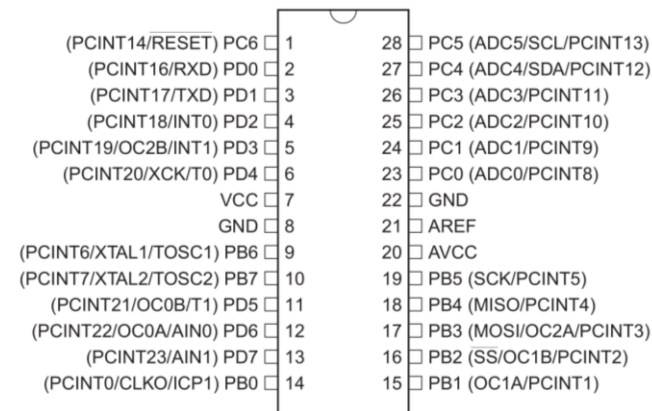
- **Mikrovezérlők**
  - Mikrovezérlők felépítése, működése
  - Mikrovezérlő típusok, gyártók
  - Mikrovezérlők perifériái
- **Mikrovezérlők programozása**
  - A C programozási nyelv (ismétlés)
  - **ATMEL AVR mikrovezérlők programozása**
  - Az ATMEL studio használata
  - Feladatmegoldás

## 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<sup>2</sup>C interfészek
- Megszakítás és ébresztés lehetőség szintváltózásra
- ...



## 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).

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

## ATMEL Studio

[http://www.atmel.com/microsite/atmel\\_studio6/](http://www.atmel.com/microsite/atmel_studio6/)

- Ingyenesen használható
- Microsoft Visual Studio alapokon
- Fordító: AVR GCC toolchain
- Támogatott nyelvek: Assembly, C/C++



## A program használata

1. Új projekt létrehozása
  1. Forrásnyelv kiválasztása
  2. Mikrovezérlő típusának kiválasztása
2. Forrásprogram megírása
3. Fordítás
4. Beégetés
5. Futtatás, hibakeresés

## A program használata

1. Új projekt létrehozása
    1. Forrásnyelv kiválasztása
    2. Mikrovezérlő típusának kiválasztása
  2. Forrásprogram megírása
  3. Fordítás
  4. Beégetés
  5. Futtatás, hibakeresés
- } Hardver-specifikus  
(Hardverigény)

## A program használata

1. Új projekt létrehozása
  1. Forrásnyelv kiválasztása
  2. Mikrovezérlő típusának kiválasztása
2. Forrásprogram megírása
3. Fordítás
4. Szimulátor használata
  1. Futtatás, hibakeresés

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

## 1. Feladat: Inicializálás

Állítsa be, hogy a B port 1. és 2. lába kimenet legyen!

Állítsa be, hogy a C port összes lába bemenet legyen!

## 2. Feladat: Digitális írás olvasás

Olvassa be a C port összes lábán jelenlévő értékeket!

Olvassa be a C port 3. lábán jelenlévő értéket!

Állítsa be a B port 1. és 2. lábát magas szintre!

## 3. Feladat: LED villogtatás

A B port 1. lábára egy LED-et (és soros ellenállást) csatlakoztattunk. Oldja meg, hogy a LED folyamatosan villogjon! A villogás periódusideje 1s legyen!

## 4. Feladat: Megszakítás

Állítsa be, hogy INT0 láb állapotváltozása megszakítást okozzon!

Állítsa be, hogy a TIMER0 időzítő (számláló) túlcsordulása megszakítást okozzon!

## 5. Feladat: Időzítés

Használja fel a TIMER0 által okozott megszakítást a LED villogtatásához!  
Változtassa az előosztót és figyelje meg mi történik!

## 6. Feladat: Watchdog

Engedélyezze a Watchdog modult a korábbi LED villogtató programban!

Futtassa a programot, majd figyelje meg mi történik!

## 7. Feladat: Watchdog

Engedélyezze a Watchdog modult a korábbi LED villogtató programban!

Kezelje a programban a watchdog számlálóját (megfelelő időközönként nullázza le)!