



# *Sisteme cu MicroProcesoare*

## **Curs 3**

### **Sistemul de întreruperi**

**Gigel Măceşanu**



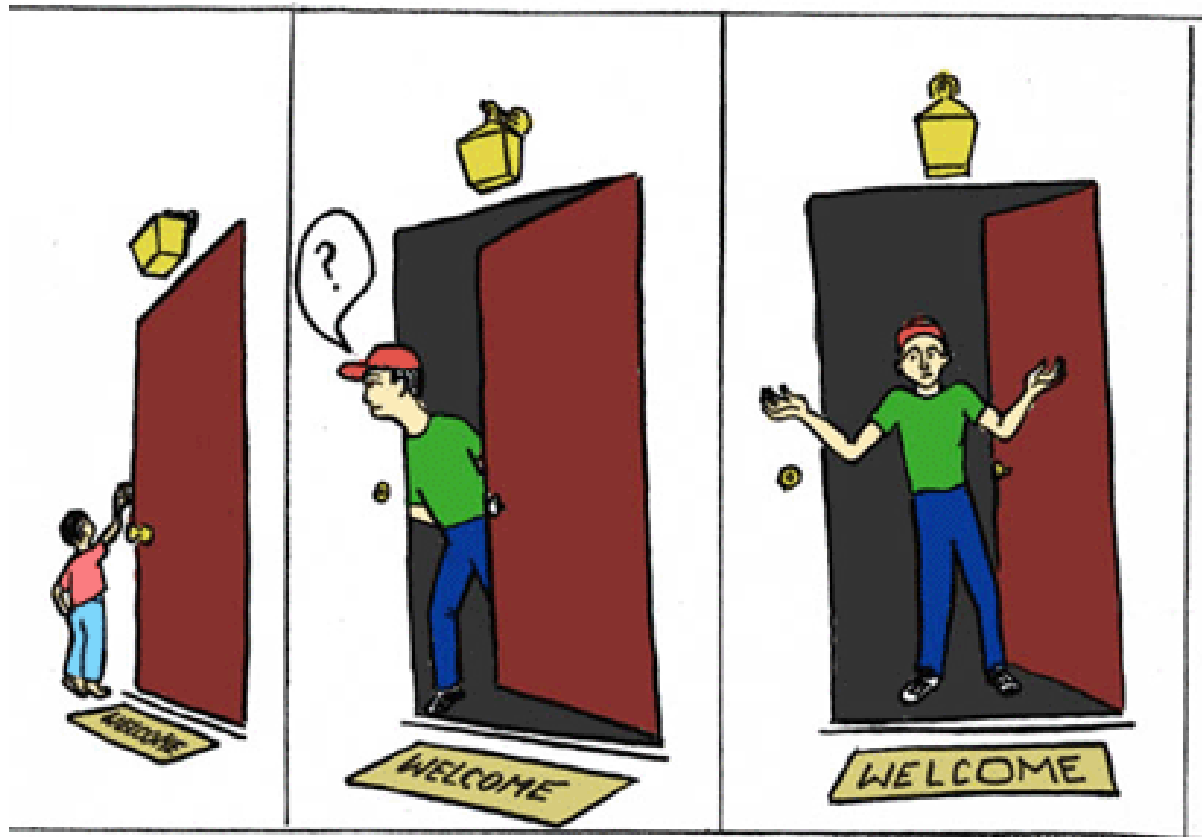
# Cuprins

- Sistemul de întreruperi
- Aplicație



# Sistemul de întreruperi

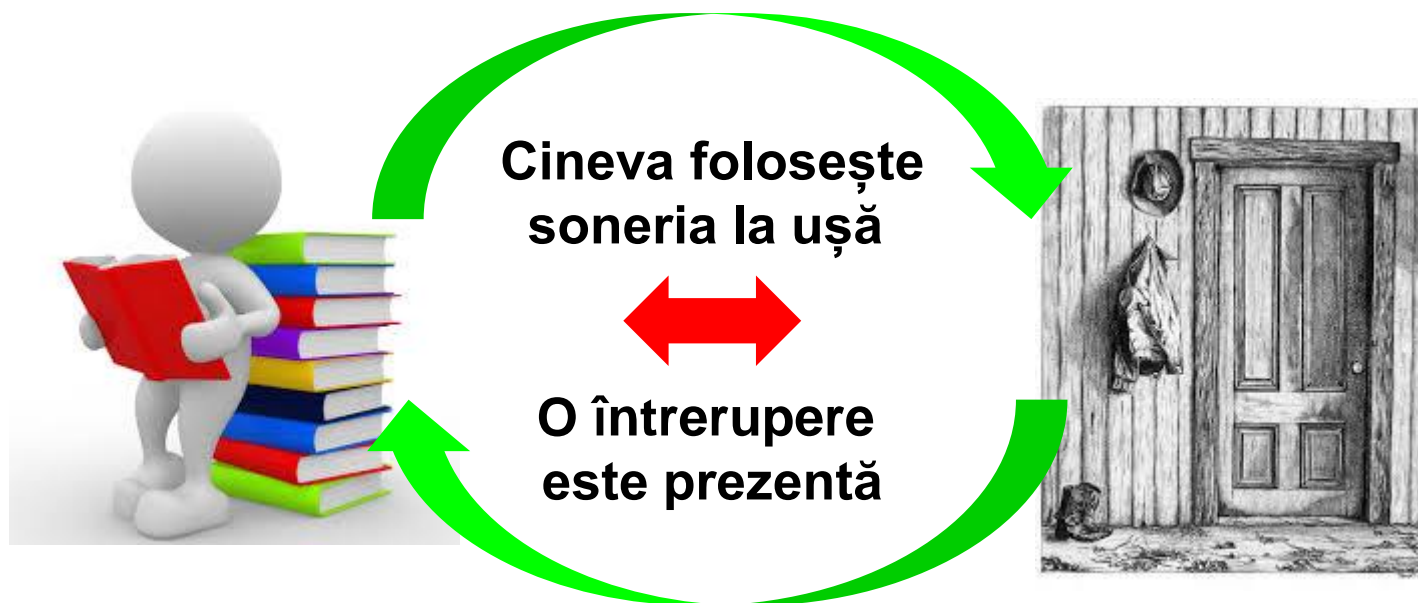
- Ce este o întrerupere?





# Sistemul de întreruperi

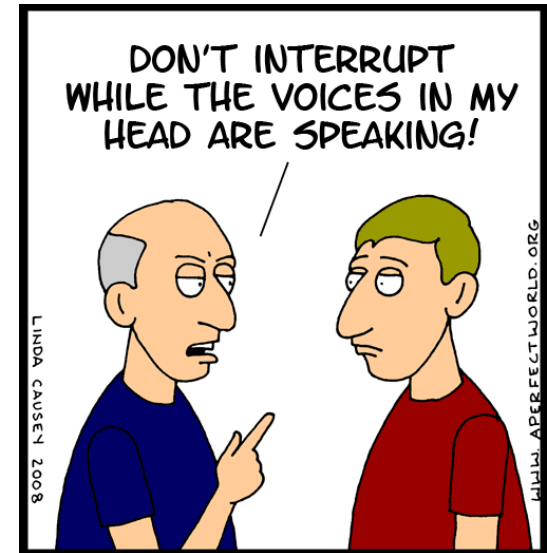
- **Definiție:** *Întreruperea reprezintă suspendarea procesului normal de execuție a programului pentru rezolvarea unei probleme prioritare*
- **Caracteristici:**
  - **Întreruperea, de regulă, este generată ca răspuns la un efect fizic intern sau extern, al unui modul periferic**





# Sistemul de întreruperi

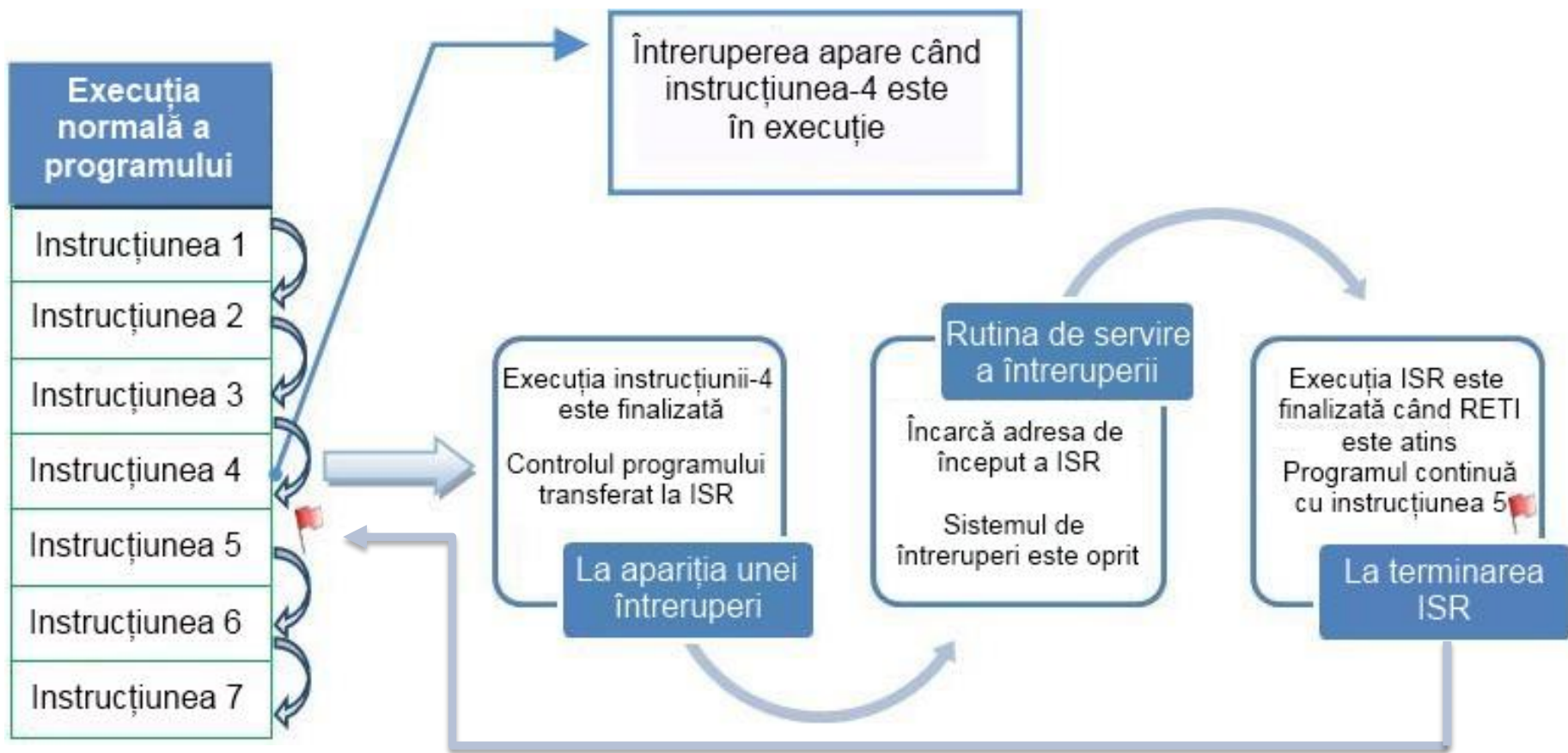
- **Exemple de întreruperi: schimbarea nivelului logic al unui pin, sfârșitul unei perioade de timp, sfârșitul unei operații de transmisie sau recepție de date, sfârșitul unei conversii etc.**
- **Tratarea unei întreruperi presupune existența unei subrutine care să permită executarea acestora**
- **Sistemul poate reacționa real-time la un stimul extern**
- **Sistemul de întreruperi poate fi activat/dezactivat**
- **Apariția unei întreruperi conduce la setarea unor indicatori care marchează acest eveniment**
- **Întreruperile pot avea prioritate diferită**





# Sistemul de întreruperi

## Principiul de funcționare al unei întreruperi





# Sistemul de întreruperi

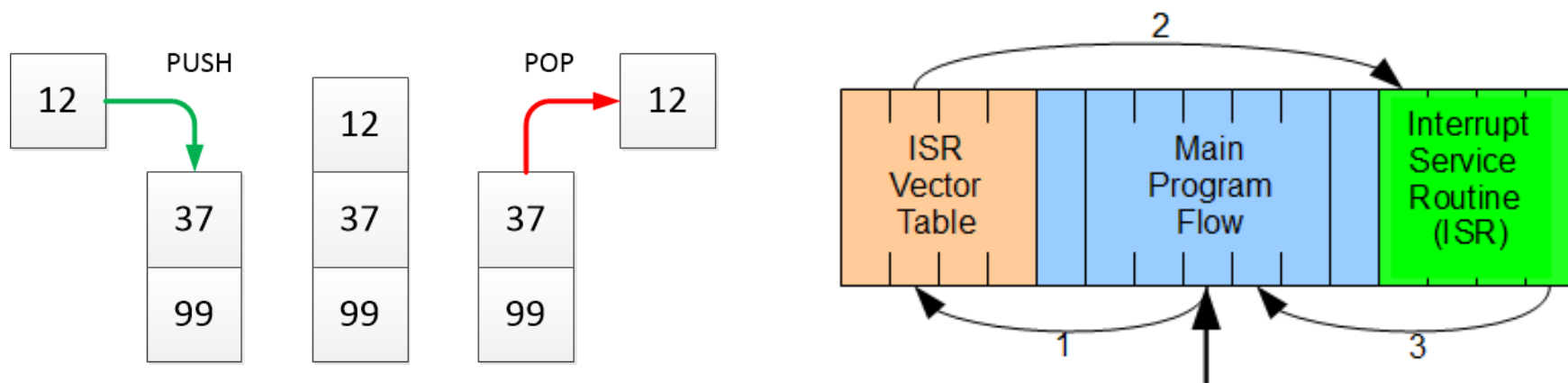
- Tabelul vectorilor de întrerupere
  - Reprezintă legătura între întreruperi și rutina de întrerupere
  - Este reprezentat sub forma unui tabel care conține informații despre fiecare vector de întrerupere
  - Vectorul de întrerupere este un număr asociat cu o întrerupere
  - Ex. vector întrerupere microcontroler ATmega16

Nr. Vector	Adresă program	Sursă	Informații
1	\$000	RESET	Ext. Pin, Power-on Reset
2	\$004	INT0	Cerere înt. ext. 0
3	\$008	INT1	Cerere înt. ext. 1
4	\$00C	TIMER2/COMP	Evaluare Timer/Comparator



# Sistemul de întreruperi

- Principiul de funcționare al unei întreruperi
  - Programul execută instrucțiuni în MPF
  - Când o întrerupere apare PC → stivă: push(PC)
  - În PC este salvată adresa 1 din ISR (din tabela vectorilor de întrerupere)
  - Funcția de tratare a întreruperii este executată
  - În PC este adusă valoarea instrucțiunii următoare: pop(PC) + 1







# Sistemul de întreruperi

- Surse de întrerupere
  - **Întreruperi externe. Sunt de două tipuri:**
    - Active pe nivel de tensiune (Level-triggered Interrupts), întreruperea apare încontinuu atâta timp cât linia are valoarea considerată (low sau high)
    - Active pe schimbare de nivel (Edge-triggered) întreruperea apare doar când este o schimbarea de nivel pe linia respectivă
  - **Întreruperi interne corespunzătoare timerelor:**
    - Când valoarea din timer este resetată (overflow)
    - Când valoarea din timer atinge o valoare prestabilită



# Sistemul de întreruperi

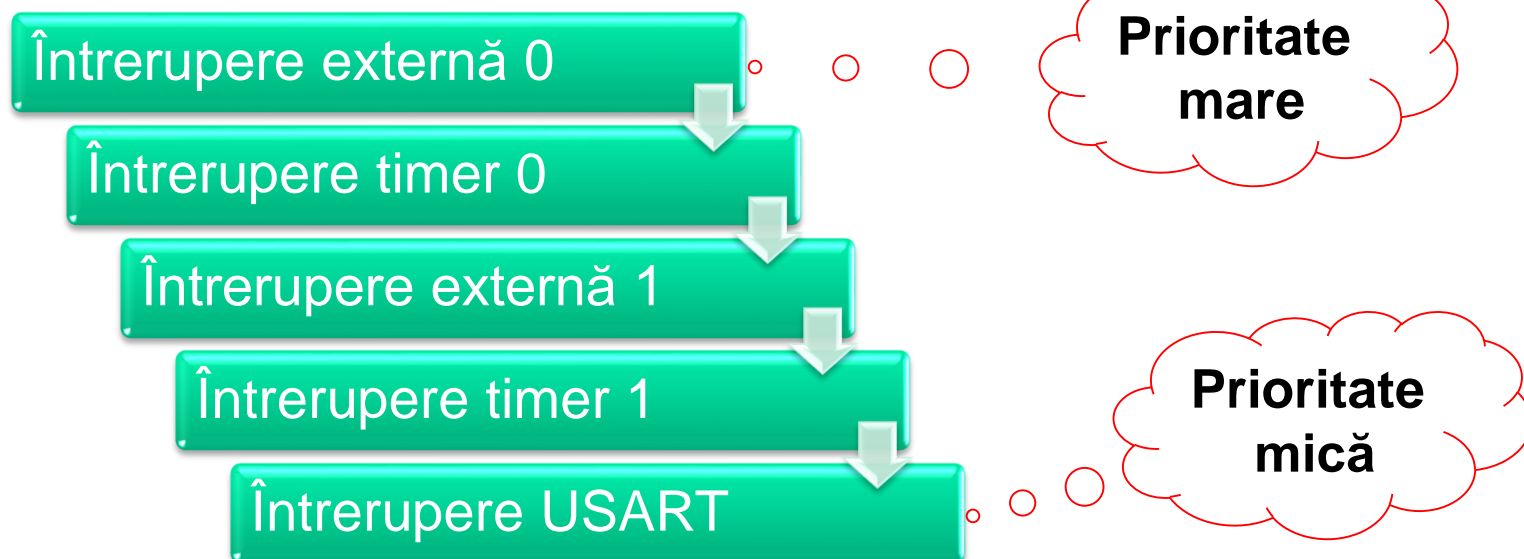
- Surse de întrerupere
  - **Întreruperi interne corespunzătoare ADC:**
    - Apare când este terminată o conversie analog-numerică
  - **Întreruperi cauzate de sistemele de comunicație (ex. USART, I2C, SPI):**
    - Provocate de disponibilitatea emitorului sau receptorului
    - Provocate de finalizarea transmisiei sau recepției de date
  - **Întreruperi provocate de module specifice unui anumit MC:**
    - Oscilator, USB, scriere memorie (EEPROM)



# Sistemul de întreruperi

## ▪ Prioritatea întreruperilor

- În același moment pot apărea două cereri diferite de întrerupere
- Este folosit un tabel cu priorități
- Prioritățile pot fi HW sau SW (poate decide programatorul)
- Pentru 8051 (tip comun de arhitectură pentru MC) este următoarea ordine de priorități:





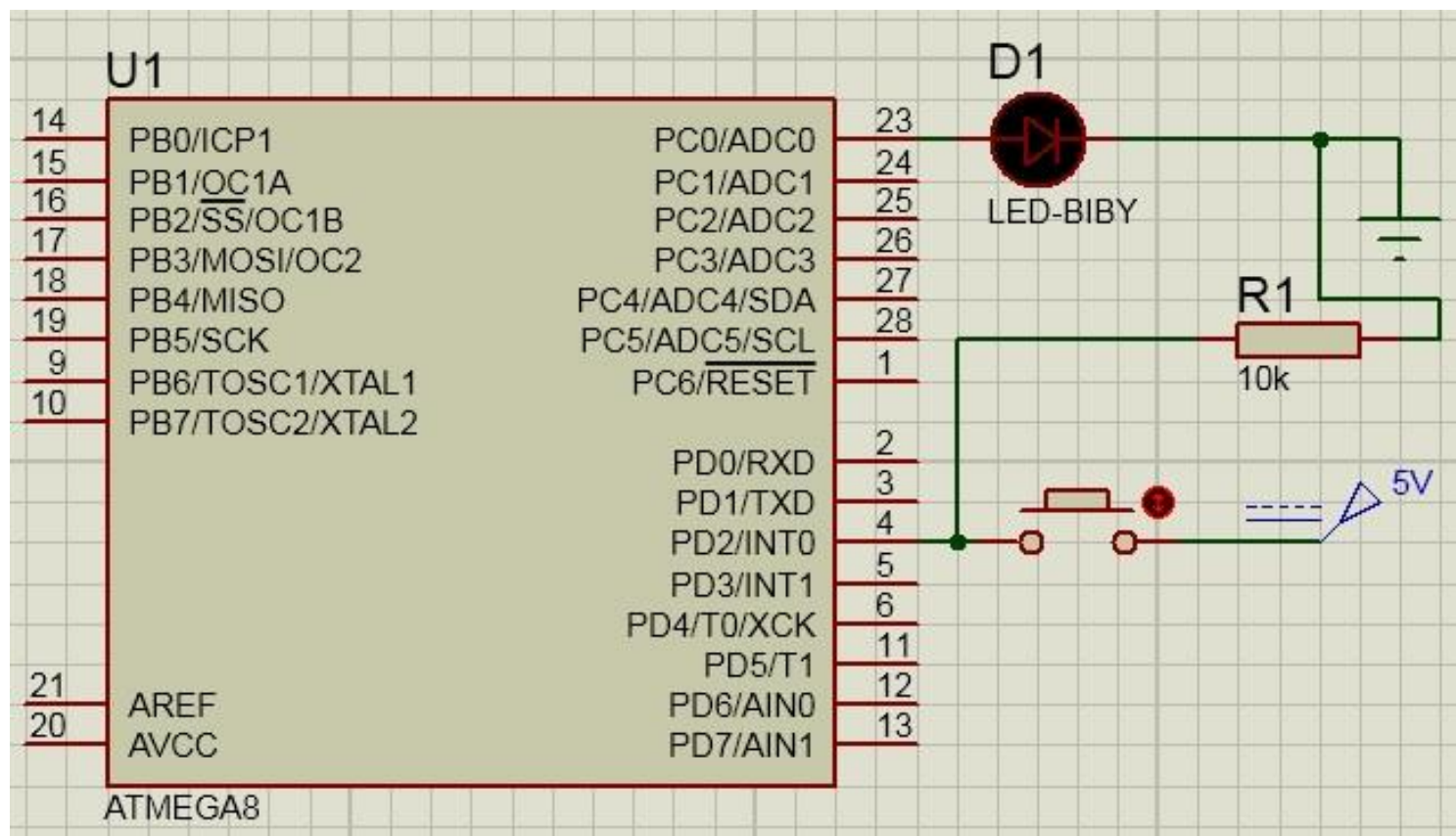
# Aplicație

- **Exemplu de utilizare a unei întreruperi**
  - **Cerință: să se implementeze un program care permite aprinderea/stingerea unui LED de la un buton. Aprinderea se va realiza pe frontul crescător al semnalului.**
  - **Funcționalitatea butonului va fi dezvoltată cu ajutorul sistemului de întreruperi externe**
  - **Se va folosi un MC ATmega8**
  - **Simularea va fi realizată utilizându-se Proteus**



# Aplicație

- Schema electrică a aplicației propuse





# Aplicație

## ▪ Inițializare porturi:

```
// Port C initialization
// Function: Bit6=I Bit5=I Bit4=I Bit3=I Bit2=I Bit1=In Bit0=0
DDRC=(0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2)
| (0<<DDC1) | (1<<DDC0);
// State: Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=0
PORTC=(0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) |
(0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);
// Port D initialization
//Function: Bit7=I Bit6=I Bit5=I Bit4=I Bit3=I Bit2=I Bit1=I
Bit0=I
DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3)
| (0<<DDD2) | (0<<DDD1) | (0<<DDD0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T
Bit0=T
PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) |
(0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);
```



# Aplicație

## ▪ Inițializare sistem întreruperi:

```
// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Rising Edge
// INT1: Off

GICR|=(0<<INT1) | (1<<INT0);
MCUCR=(0<<ISC11) | (0<<ISC10) | (1<<ISC01) | (1<<ISC00);
GIFR=(0<<INTF1) | (1<<INTF0);
```

### Legendă:

**GICR** – registru de configurare a întreruperilor

**MCUCR** – registru de control

**ISCxx** – Controlul sensului întreruperii

**GIFR** – registru cu flaguri generale de întrerupere

**INTFx** – flag întrerupere externă



# Aplicație

## ▪ Rutina de întrerupere:

```
// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
    // Place your code here
    PORTC.PORTC0 = ~PORTC.PORTC0; //aprindere sau stingere led
}
```

## ▪ Activarea sistemului de întreruperi global

```
// Global enable interrupts
#asm("sei")
```

### Obs:

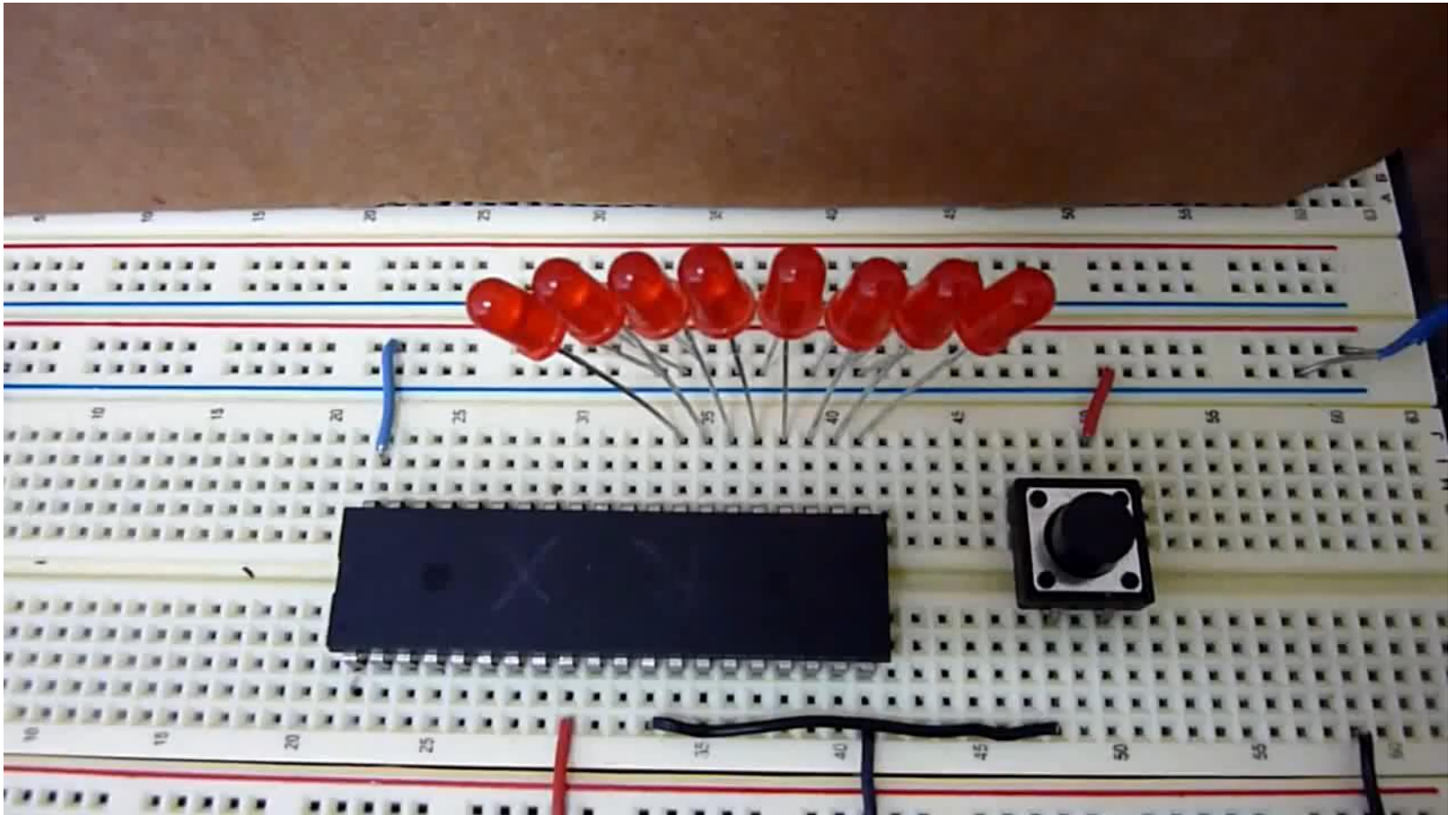
- Regiștrii implicați cât și valorile acestora, pentru o funcționare corectă, se găsesc în fișa de catalog a componentei utilizate.





# Aplicație

- Întrerupere ATmega8515 – aplicație





**Contact:**

**Email: [gigel.macesanu@unitbv.ro](mailto:gigel.macesanu@unitbv.ro)**

**Web: [rovis.unitbv.ro](http://rovis.unitbv.ro)**