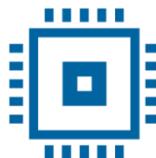


Sisteme cu MicroProcesoare

Cursul 5

Sistemul de Întreruperi

Conf. Gigel Măceșanu

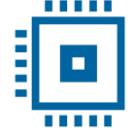


Universitatea
Transilvania
din Brașov

FACULTATEA DE INGINERIE ELECTRICĂ
ȘI ȘTIINȚA CALCULATOARELOR

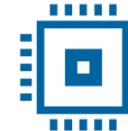


Cuprins



- Prezentare generală
- Caracteristici ale sistemului de întreruperi
- Prințipiu de funcționare
- Surse de întrerupere
- Aplicație cu sistemul de întreruperi

Obiectivul cursului

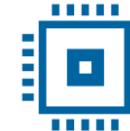


■ Înțelegerea sistemului de funcționare a unei îintreruperi:

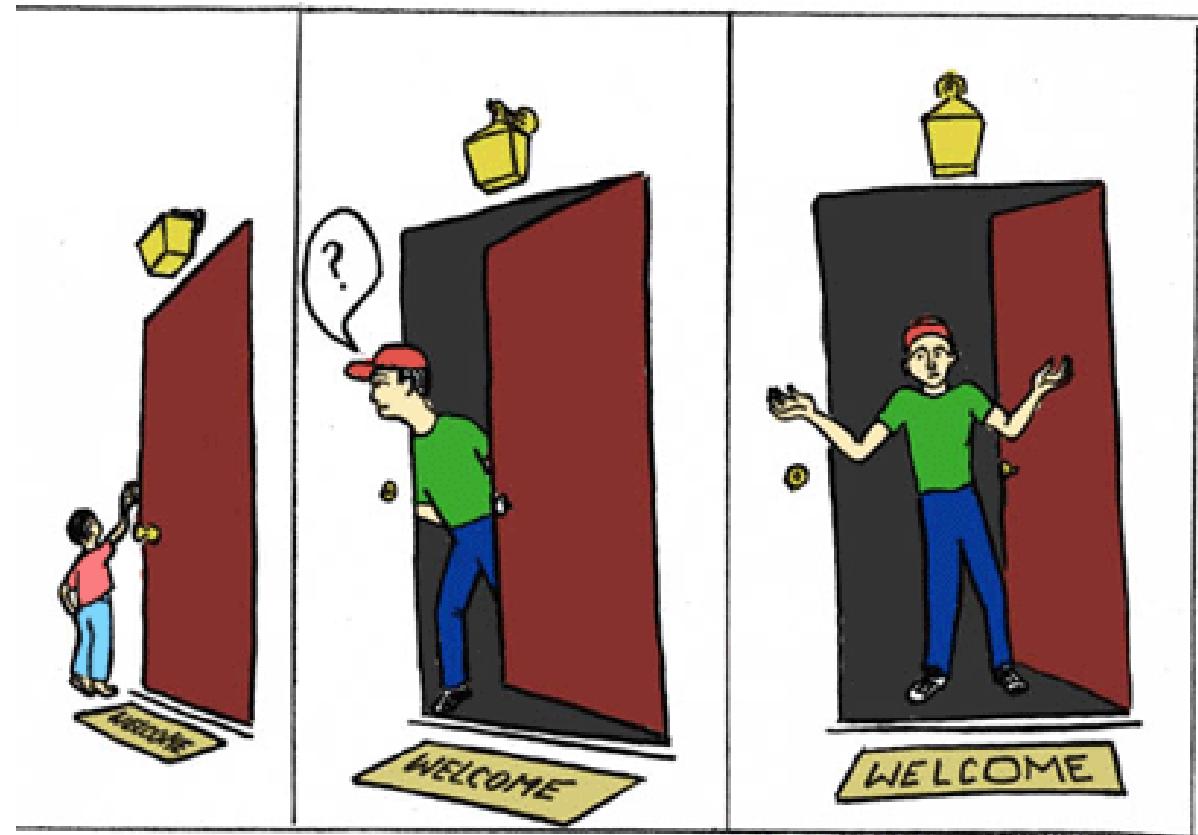
- Structura internă
- Mecanisme de declansare
- Module care pot genera îintreruperi
- Utilizarea în aplicații



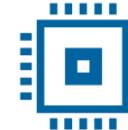
Sistemul de întreruperi



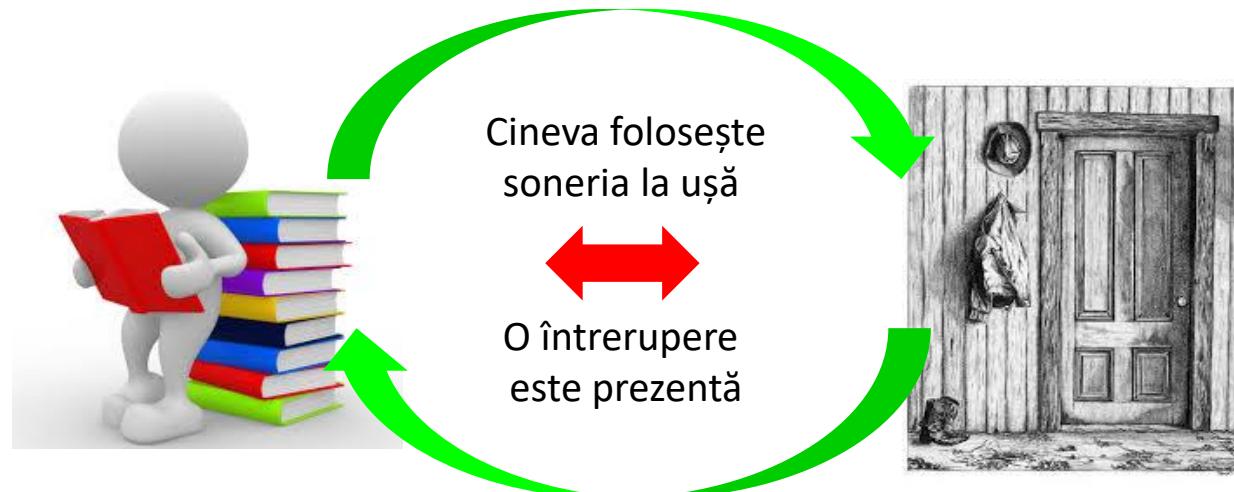
■ Ce este o întrerupere?



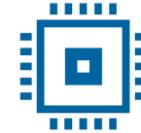
Sistemul de Întreruperi



- 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
 - Ca și răspuns la aceste efecte, CPU oprește execuția codului curent și începe execuția codului de întrerupere
 - CPU nu monitorizează apariția unui întreruperi, ci reacționează doar când întreruperea apare



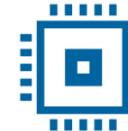
Caracteristici ale sistemului de întreruperi



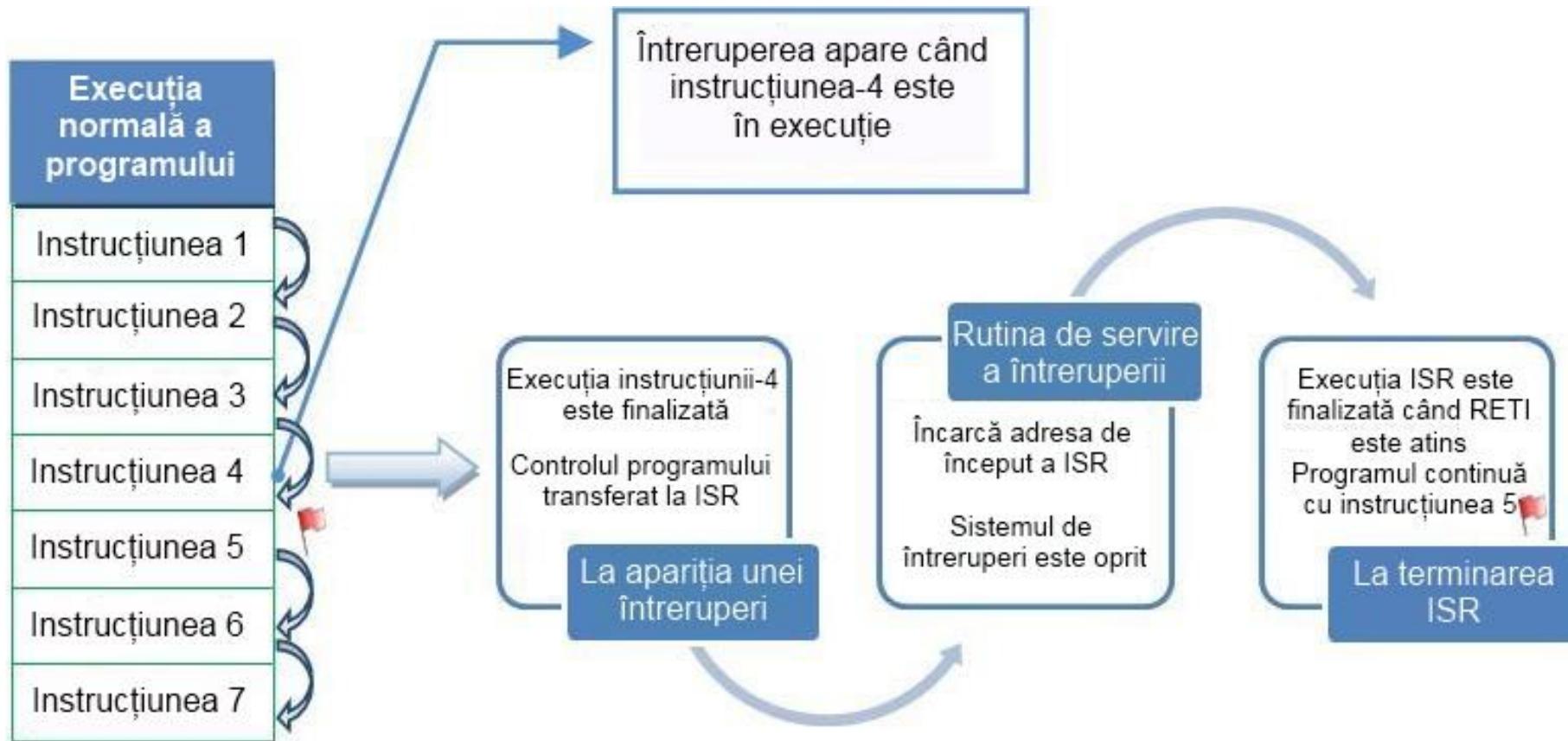
- Exemple de întreruperi: schimbarea nivelului logic al unui pin, sfârșitul unei perioade de timp, sfârșitul unei operații de transmisie sau recepție de date, sfârșitul unei conversii etc.
- Tratarea unei întreruperi presupune existența unei subroutines care să permită executarea acestora
- Sistemul 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
- Crește eficiența programelor, permitând tratarea evenimentelor asincrone



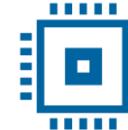
Principiu de funcționare



■ Principiul de funcționare al unei întreruperi

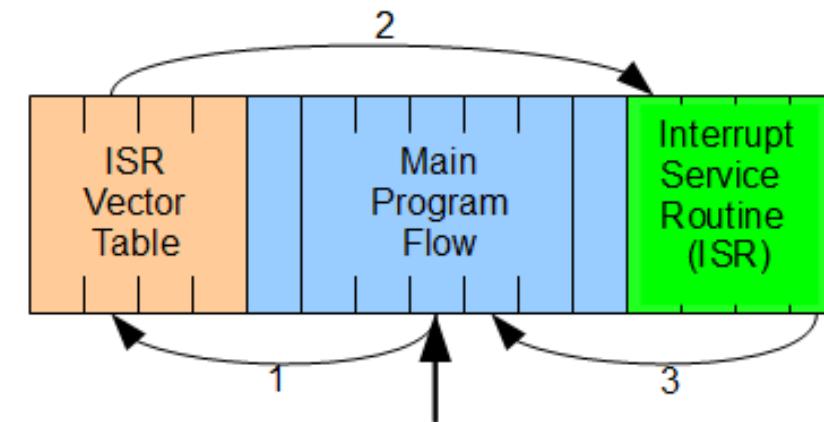
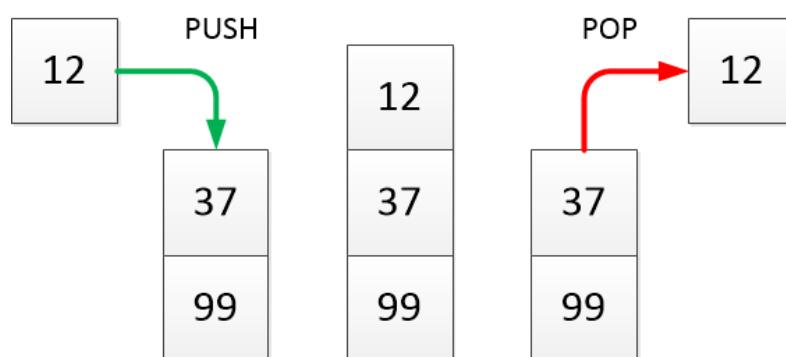


Principiu de funcționare

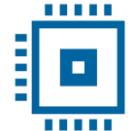


■ Principiul de funcționare al unei îintreruperi:

- Programul execută instrucțiuni în programul principal (MPF)
- Când o îintrerupere apare Program Counter-ul (PC) se salvează în stivă: push (PC)
- În PC este salvată adresa 1 din ISR (din tabela vectorilor de îintrerupere, pe baza sursei de îintrerupere)
- Funcția de tratare a îintreruperii este executată
- După execuție în PC este adusă valoarea instrucțiunii următoare: pop (PC) + 1

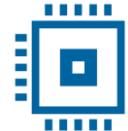


Rutina de tratare a întreruperii



- Rutina de tratare a întreruperii, cunoscută și sub numele de ISR (Interrupt Service Routine), este o funcție specială din cadrul unui program care este invocată automat atunci când apare o întrerupere
- Caracteristicile ISR:
 - Executare automată: când o întrerupere este declanșată, microcontrolerul întrerupe execuția curentă a programului și sare la adresa de memorie a ISR-ului corespunzător acelei întreruperi
 - Salvarea și restaurarea contextului: înainte de a executa ISR-ul, microcontrolerul salvează automat starea curentă a procesorului (de exemplu, registrii de lucru și contorul de program). După ce ISR-ul este finalizat, microcontrolerul restaurează contextul salvat și reia execuția programului de la punctul unde a fost întrerupt
 - Prioritate ridicată: ISR-urile sunt, concepute pentru a fi cât mai rapide și mai eficiente, deoarece în timpul execuției lor, alte activități sunt suspendate
 - returnare de valori: ISR-urile nu returnează valori, spre deosebire de funcțiile obișnuite dintr-un program. Rolul lor este de a trata evenimentul întreruperii și de a pregăti sistemul pentru a continua execuția normală

Tabelul vectorilor de întrerupere

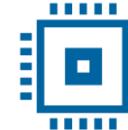


■ Reprezintă legătura între întrerupere și rutina de întrerupere corespunzătoare

■ Caracteristici:

- Secțiune fixă de memorie rezervată într-o locație specifică, definită de arhitectura microcontrolerului. Fiecare intrare din această tabelă corespunde unui tip specific de întrerupere
- Fiecare poziție din tabelă conține o adresă de memorie care indică locația de start a ISR-ului asociat
- Tabela asociază fiecare eveniment de întrerupere posibil cu ISR-ul corespunzător
- Ordinea vectorilor din tabelă este predefinită de producătorul microcontrolerului.
- La unele familii de microcontrolere, adresele stocate în tabela vectorilor de întrerupere sunt configurabile, permitând programatorului să direcționeze întreruperile către diferite ISR-uri în funcție de necesități.
- Ex. vector întrerupere microcontroler ATMega16

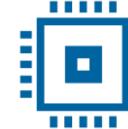
Nr. Vector	Adresă program	Sursă	Informații
1	\$000	RESET	Ext. Pin, Power-on Reset
2	\$004	INT0	Cerere înt. ext. 0
3	\$008	INT1	Cerere înt. ext. 1
4	\$00C	TIMER2/COMP	Evaluare Timer/Comparitor



■ Întreruperi externe

- Intrări digitale externe (pini cu funcții generale, cu capacitate de întrerupere):
 - Active pe nivel de tensiune (Level-triggered Interrupts), întreruperea apare încontinuu atât 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
- Sisteme de Comunicații USART, SPI, I2C, USB, Ethernet):
 - Perifericele de comunicație pot genera întreruperi atunci când datele sunt gata de a fi transferate sau au fost primite, permitând procesorului să proceseze datele imediat

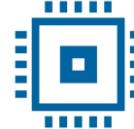
Surse de Întrerupere



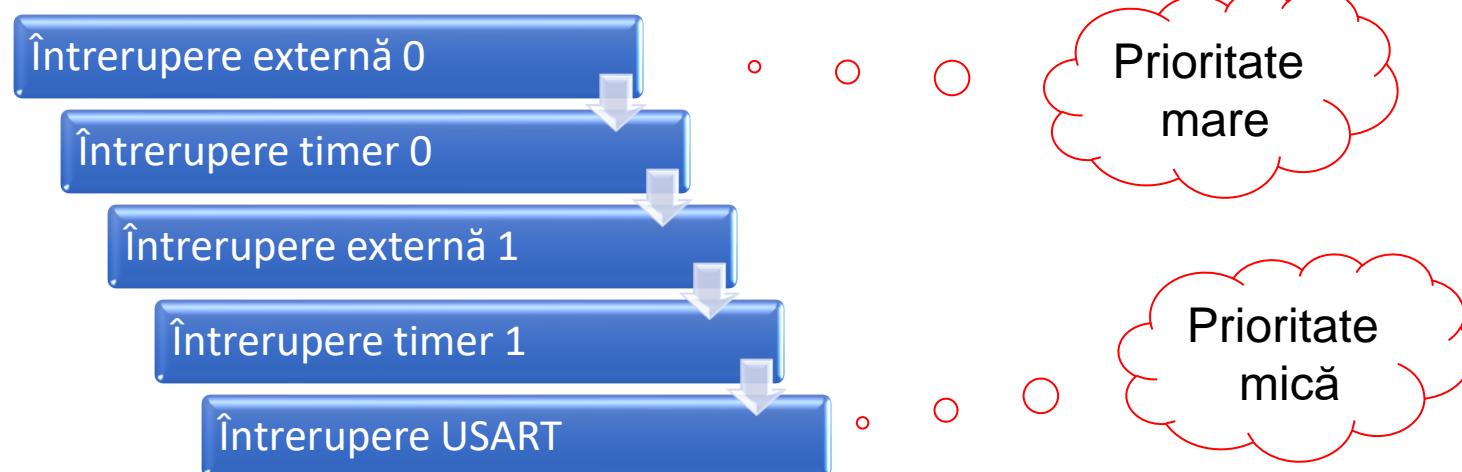
■ Întreruperi interne:

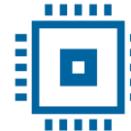
- Întreruperi interne corespunzătoare timer-elor/counter-elor:
 - Când valoarea din timer/counter este resetată (overflow)
 - Când valoarea din timer/counter atinge o valoare prestabilită
- Întreruperi interne corespunzătoare ADC:
 - Apare când este terminată o conversie analog-numerică
- Întreruperi provocate de module specifice unui anumit MC:
 - Oscilator, watchdog-timer, scriere memorie (EEPROM)

Priorități ale sistemului de întreruperi



- Prioritatea întreruperilor determină ordinea în care întreruperile sunt tratate atunci când apar mai multe întreruperi simultan sau când o întrerupere se declanșează în timpul tratării unei alte întreruperi.
 - Prioritățile pot fi HW sau SW (poate decide programatorul)
 - Masca de întrerupere este un mecanism prin care anumite întreruperi pot fi temporar dezactivate, indiferent de prioritatea lor.
 - Pentru 8051 (tip comun de arhitectură pentru MC) este următoarea ordine de priorități:

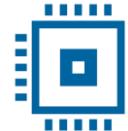




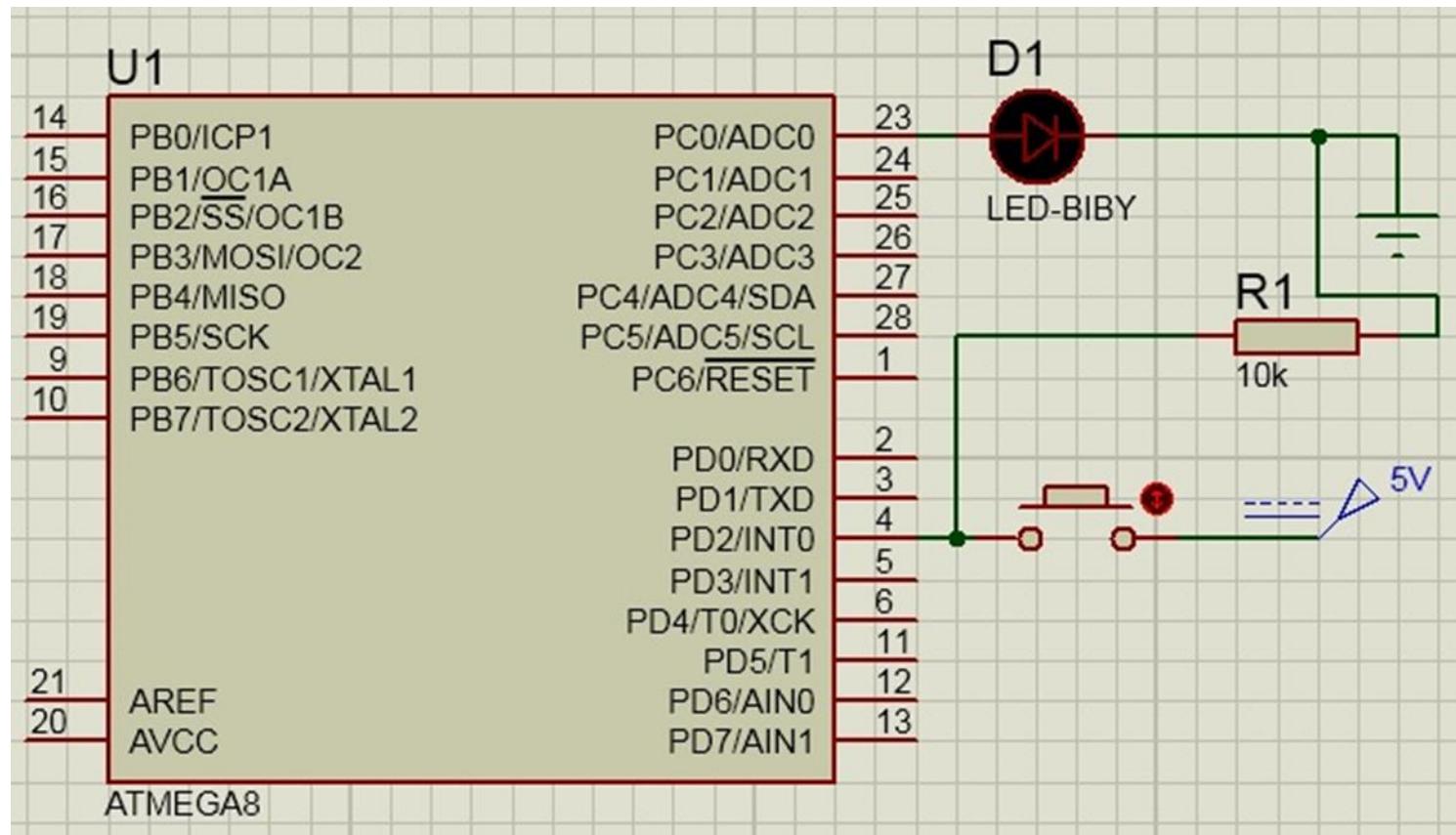
■ Exemplu de utilizare a unei întreruperi:

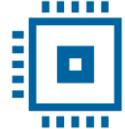
- Cerință: să se implementeze un program care permite aprinderea/stingerea unui LED de la un buton. Aprinderea se va realiza pe frontul crescător al semnalului.
- Funcționalitatea butonului va fi dezvoltată cu ajutorul sistemului de întreruperi externe
- Se va folosi un MC din familia AVR

Aplicație



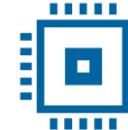
■ Schema electrică a aplicației propuse





■ Initializare porturi:

```
// Port C initialization
// Function: Bit6=I Bit5=I Bit4=I Bit3=I Bit2=I Bit1=In Bit0=O
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);
```



■ Initializare 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:

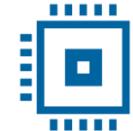
GICR – registru de configurare a întreruperilor

MCUCR – registru de control

ISCxx – Controlul sensului întreruperii

GIFR – registru cu flaguri generale de întrerupere

INTFx – flag întrerupere externă



■ Rutina de întrerupere:

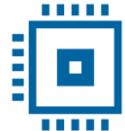
```
// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
    // Place your code here
    PORTC.PORTC0 = ~PORTC.PORTC0; //aprindere sau stingere led
}
```

■ Activarea sistemului de întreruperi global

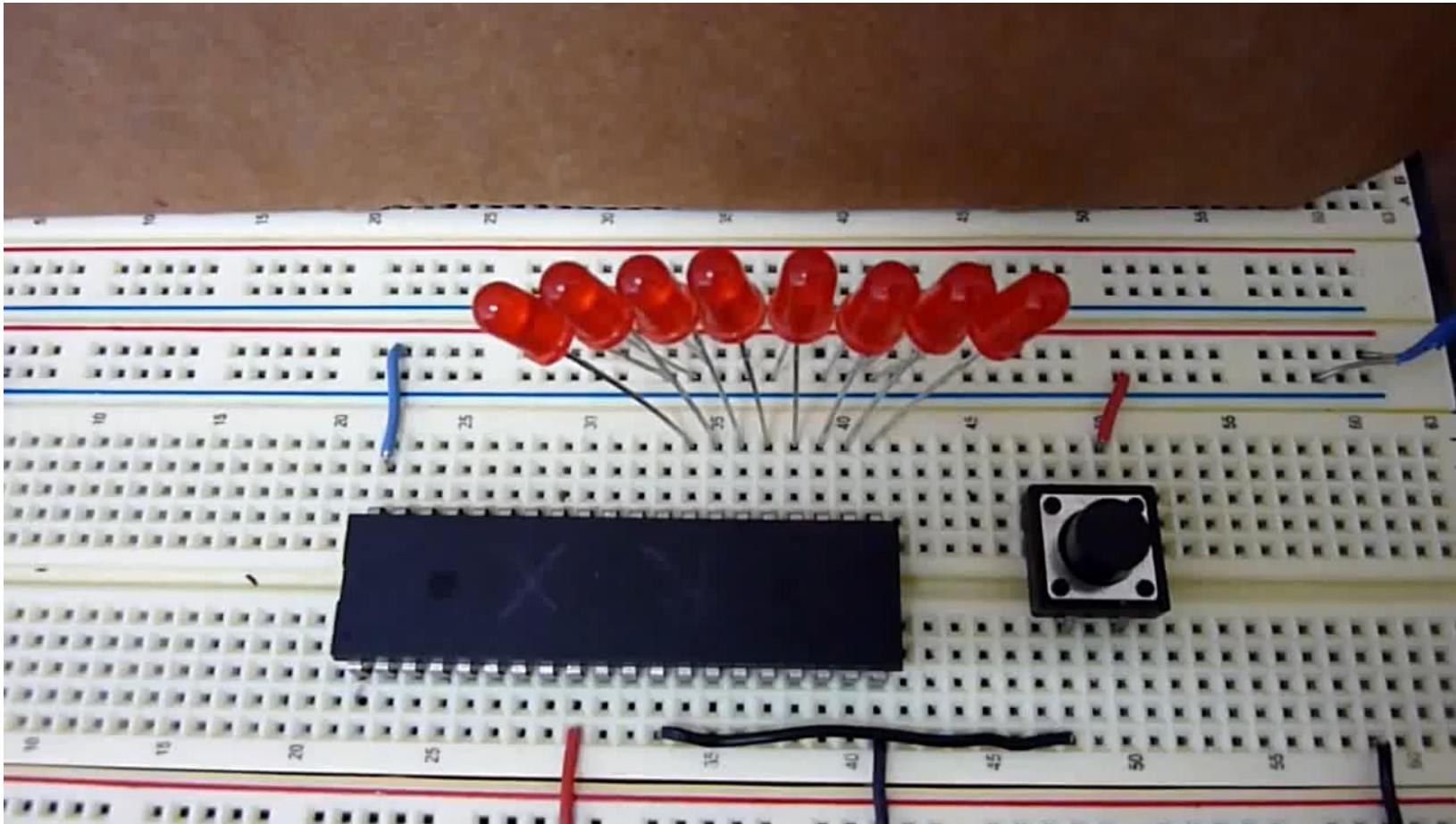
```
// Global enable interrupts
#asm("sei")// sau SREG |= (1 << 7);
```

Obs:

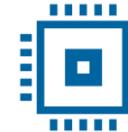
- Regiștrii implicați cât și valorile acestora, pentru o funcționare corectă, se găsesc în fișa de catalog a componentei utilizate.



- Întrerupere ATMega8515 – aplicație

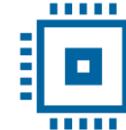


Concluzii



- Sistemul de întreruperi este configurabil, în funcție de sursele de întrerupere pe care dorim să le utilizăm și specific fiecărei familii de MC
- Rutina de tratare a întreruperii și tabela vectorilor de întrerupere sunt extrem de importante în funcționarea acestui sistem
- Fiecare întrerupere corespunde unei locații specifice în memorie
- Întreruperile sunt cu priorități diferite
 - Configurabile în Sw sau în Hw
 - În unele situații, configurabile





Contact:

Email: gigel.macesanu@unitbv.ro
elearning.unitbv.ro - Sisteme cu Microprocesoare