

### 3. Sistemul de întreruperi

---

---

Sistemul de întreruperi externe al microcontrolerului la ATmega328

Exemplu de utilizare a sistemului de întreruperi

Aplicație propusă

---

Datorită existenței unei singure unități aritmetice și logice, microcontrolerul poate executa instrucțiuni într-o manieră secvențială. Astfel, codul mașină va rula sub forma unei structuri secvențiale. Totuși, există situații neprevăzute în care trebuie să se execute o altă activitate față de cea care rulează în momentul respectiv. Acest tip de eveniment poate fi executat cu ajutorul sistemului de întreruperi. În acest sens, fiecărui eveniment i se asociază o întrerupere. Apariția evenimentului respectiv va provoca executarea unei rutine de cod special scrisă de programator (rutine de deservire a întreruperii).

Întreruperile sunt dese utilizate în corelare cu îndeplinirea unei condiții (interioare sau exterioare microcontrolerului). Spre deosebire de sistemul de intrare oferit de modulul PORT, întreruperile permit executarea codului indiferent de momentul apariției evenimentului. Întreruperile oferă flexibilitate în scrierea și rularea programelor. Cererea de întrerupere apare la nivel hardware, deci nu este necesară coordonarea rutinei de către programator, microcontrolerul fiind cel care gestionează această cerere.

În cadrul unui sistem cu microcontroler pot exista mai multe surse de întrerupere. Cele mai des întâlnite sunt întreruperile externe, cele provocate de timere sau întreruperile provocate de module de comunicație. Placa de dezvoltare Arduino UNO, ce are în componență un microcontroler ATmega328P, are 26 de surse de întrerupere. Dintre acestea, întreruperile prezentate în tabelul 3.1 prezintă interes în cadrul laboratoarelor.

Drept studiu de caz, lucrarea de fata se va concentra asupra utilizării întreruperilor externe. Placa de dezvoltare Arduino UNO, ce are la baza microcontrolerul ATmega328, detine doi pini pe care se pot genera evenimente externe: INT0 și INT1 (pinii 2 și 3 de pe placa Arduino). Întreruperile de pe acești doi pini pot fi generate pe frontul crescător sau descrescător al semnalului de intrare sau de nivelul de LOW al acestuia. Cererea de întrerupere este implementată în hardware. O altă cerere de întrerupere poate fi dată de schimbarea valorii unui pin din portul MC-ului. Pinii afectați sunt cei prezenți în tabela vectorilor de întrerupere.

Tab. 3.1 Tipuri de îintreruperi disponibile pentru microcontrolerul ATmega328P.

Vector de întrerupere	Tip întrerupere	Sursă	Nume vector
2	Întrerupere externă	INT0	INT0_vect
3	Întrerupere externă	INT1	INT1_vect
4	Întrerupere externă la nivel de port, D8-D13	PCINT0	PCINT0_vect
5	Întrerupere externă la nivel de port, A0-A5	PCINT1	PCINT1_vect
6	Întrerupere externă la nivel de port, D0-D7	PCINT2	PCINT2_vect
10	Timer/Counter2 overflow	TIMER2_OVF	TIMER2_OVF_vect
14	Timer/Counter1 overflow	TIMER1_OVF	TIMER1_OVF_vect
17	Timer/Counter0 overflow	TIMER0_OVF	TIMER0_OVF_vect
18	Transfer complet pe SPI	SPI STC	SPI_STC_vect
19	USART Rx recepție completă	USART_RX	USART_RX_vect
21	USART Tx transfer complet	USART_TX	USART_TX_vect
22	Conversie ADC completă	ADC	ADC_vect
25	Transfer complet pe I2C	TWI	TWI_vect

Pentru cazul îintreruperii externe, registrele EICRA, EIMSK, EIFR și SREG sunt importante. În tabelul 3.1 sunt prezentării biții registrului EICRA (*External Interrupt Control Register A*). Acest regisztr este responsabil pentru controlul îintreruperii. Biții acestui regisztr sunt următorii:



Fig. 3.1 Registrul EICRA.

- Biții 3:2 – ISC1n: Acești biți permit controlul sensului îintreruperii 1 [n=1:0]. O îintrerupere externă pe pinul INT1 este funcțională dacă bitul 7 din regisztrul SREG este activat și masca corespunzătoare îintreruperii este activată. Modurile prin care se poate genera o cerere de îintrerupere pentru această sursă externă sunt date în funcție de valorile biților 3:2 astfel:
  - 00: valoarea de Low a pinului INT1;
  - 01: orice schimbare a valorii logice la pinul INT1;
  - 10: schimbarea din High în Low a valorii pinului INT1;
  - 11: schimbarea din Low în High a valorii pinului INT1.
- Biții 1:0 – ISC0n: Acești biți permit controlul sensului îintreruperii 0 [n=1:0]. O îintrerupere externă pe pinul INT0 este funcțională dacă bitul 7 din regisztrul SREG este activat și masca corespunzătoare îintreruperii este activată. Modurile prin care se poate genera o cerere de îintrerupere pentru această sursă externă sunt date în funcție

de valorile biților 3:2 astfel:

- 00: valoarea de Low a pinului INT0;
- 01: orice schimbare a valorii logice la pinul INT0;
- 10: schimbarea din High în Low a valorii pinului INT0;
- 11: schimbarea din Low în High a valorii pinului INT0.

În tabelul 3.2 sunt prezentării biții registrului EIMSK (*External Interrupt Mask Register*). Acest regisztr este utilizat pentru a activa sau dezactiva masca unuia dintre pinii INT0 și INT1, pentru generarea de înteruperi. Biții acestui regiszru sunt următorii:

Bit	7	6	5	4	3	2	1	0
Access							INT1	INT0

Fig. 3.2 Registrul EIMSK.

- Bitul 1 - INT1: Activare înterupere externă 1. În momentul în care bitul INT1 este setat și bitul 7 din regisztrul de stare SREG este setat, înteruperea pe pin este activată. Generarea înteruperii are în vedere setările efectuate în regisztrul EICRA (tabelul 3.1);
- Bitul 0 - INT0: Activare înterupere externă 0. În momentul în care bitul INT0 este setat și bitul 7 din regisztrul de stare SREG este setat, înteruperea pe pin este activată. Generarea înteruperii are în vedere setările efectuate în regisztrul EICRA (tabelul 3.1); Regisztrul EIFR (*External Interrupt Flag Register*) este responsabil pentru înregistrarea apariției unei înteruperi. La apariția unei înteruperi externe biții din acest regiszru vor fi setati automat de catre microcontroler. Următorii doi biți fac parte din acest regiszru (figura 3.3):

Bit	7	6	5	4	3	2	1	0
Access							INTF1	INTF0

Fig. 3.3 Registrul EIFR.

- Bitul 1 - INTF1: Când o cerere de înterupere este generată pe pinul INT1, bitul INTF1 este setat în mod automat;
- Bitul 0 - INTF0: Când o cerere de înterupere este generată pe pinul INT0, bitul INTF0 este setat în mod automat;

Tratarea unei înteruperi se face în interiorul funcției ISR(). Aceasta este o funcție standard de apel a vectorului de înterupere pentru microprocesoare ATmega. Această funcție poate avea doar un parametru și nu returnează nimic. Parametrii posibili sunt numele vectorilor de înterupere, prezentați în ultima coloană a tabelului 3.1. Sintaxa funcției este:

---

Algoritmul 3.1 Rutina de deservire a înteruperi - sintaxă generală

---

```

1 ISR (isr_vector)
2 {
3     //ISR_code

```

4 }

### 3.1 Exemplu de aplicație

Să se elaboreze și simuleze, utilizându-se Tinkercad, un program care aprinde și sting la fiecare apăsare de buton, un led. Tratarea aprinderii/stingerii ledului trebuie să se realizeze utilizându-se sistemul de întreruperi al microcontrolerului. Schema electrică a schemei propuse este prezentată în figura 3.4. Componentele utilizate în schemă sunt:

- 1x 3 x AA Battery;
  - 1x Arduino Uno R3;
  - 2x 1 kohm Resistor;
  - 1x Yellow LED;
  - 1x Pushbutton;
  - 1x Breadboard Small.

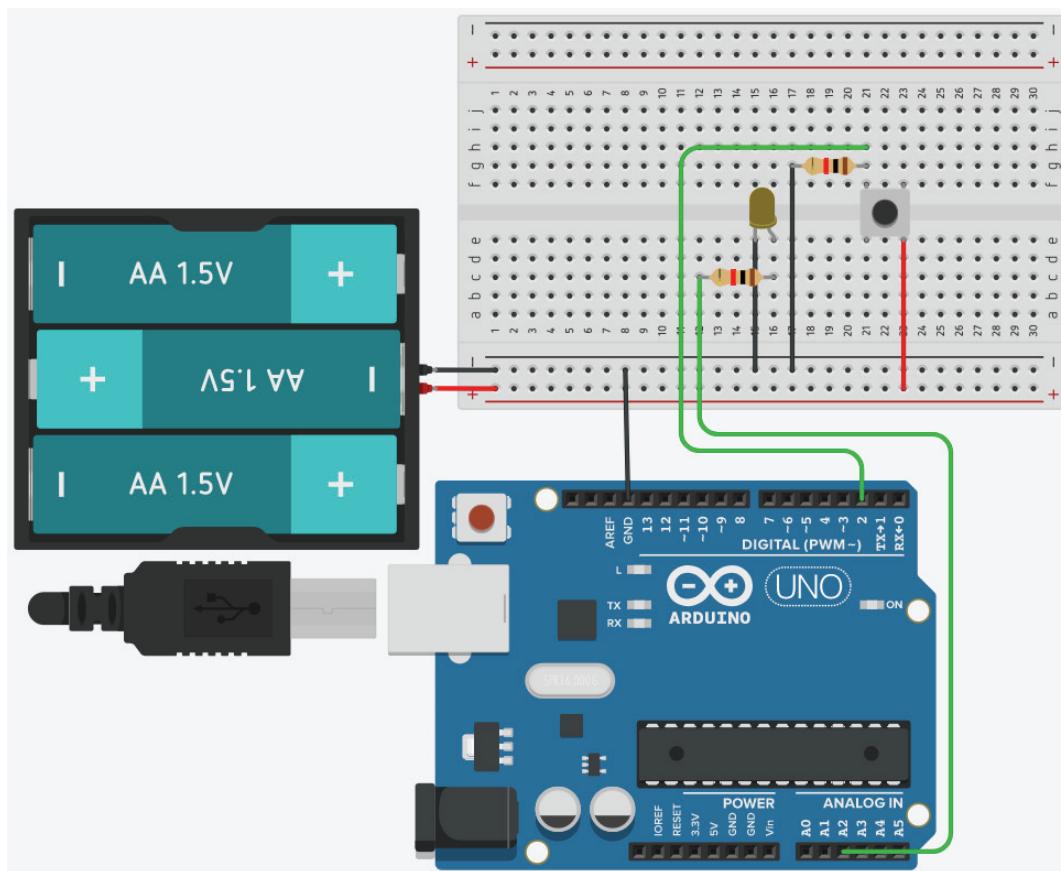


Fig. 3.4 Schema electrică de funcționare a aplicației cu intreruperi.

Programul care realizează cerințele impuse este următorul:

### Algoritmul 3.2 Utilizarea sistemului de întreruperi.

```
1 #include <avr/interrupt.h>
2
3 // definire rutina de tratare a
4 // intreruperii INT0
```

```
5 ISR(INT0_vect)
6 {
7     // dezactivare intreruperi globale
8     SREG &= ~(1 << SREG_I);
9
10    // verific starea curenta a LED-ului
11    // si o schimb in cea opusa
12    if ((PORTC & (1 << PC2)) == 0) // LOW -> HIGH
13    {
14        PORTC |= (1 << PC2);
15    }
16    else // HIGH -> LOW
17    {
18        PORTC &= ~(1 << PC2);
19    }
20
21    // activare intreruperi globale
22    SREG |= 1 << SREG_I;
23 }
24
25 void setup()
26 {
27     // configurare PORTC ca iesire
28     DDRC = 0x00;
29     // setare PORTC pe LOW
30     PORTC = 0x00;
31
32     // activare intreruperi globale
33     SREG |= (1 << SREG_I);
34
35     // configurare intreruperi externe
36     EICRA = (0 << ISC11) | (0 << ISC10) | (1 << ISC01) | (1 <<
37         ↪ ISC00);
38     EIMSK = (0 << INT1) | (1 << INT0);
39     EIFR = (0 << INTF1) | (0 << INTF0);
40     PCICR = (0 << PCIE2) | (0 << PCIE1) | (0 << PCIE0);
41
42 void loop() {
43 }
```

În algoritmul anterior rutina de intrerupere pentru pinul INT0 este compusa din 3 secțiuni. La început sunt dezactivate toate intreruperile. Acest lucru previne apariția unei alte intreruperi în timp ce se tratează întreruperea curentă. În continuare se verifica starea pinui PC2 (la care este conectat LED-ul). În funcție de aceasta se inversează starea pinului pentru a obține un efect vizual pe led (stingere sau aprindere). La final este reactivat sistemul de intreruperi global al microcontroller-ului.

În interiorul funcției setup() se realizează, la început, inițializarea portului C (liniile 23 și 24). Pe pinii acestui port se realizează aprinderea led-ului. La linia 26 este activat sistemul de intreruperi global, prin trecerea bitului 7 (SREG\_I) pe nivelul logic High. Liniile 29:32 permit configurarea intreruperii, astfel:

- linia 36: trecerea Low-High pe pinul INT0 generează intrerupere;
- linia 37: activare mască INT0 (activare intrerupere INT0);
- linia 38: resetarea bitilor ce marchează apariția unei intreruperi;
- linia 39: dezactivare intrerupere la nivel de port.

### 3.2 Aplicație propusă

Să se implementeze o aplicație software pentru microcontroler care să permită numărarea unor piese. Pentru simularea circuitului precum și testarea codului software se poate utiliza platforma on-line Tinkercard. Afisarea rezultatului număririi se va realiza prin intermediul unui afișaj cu 7 segmente. Incrementarea valorii din contor se va realiza prin apăsarea unui buton fără reținere. Acest eveniment rebusie tratat într-o intrerupere externă, pe frontul descrescător al semnalului provenit de la buton. Decrementarea valorii din contor se va realiza prin apăsarea unui al doilea buton tot fără reținere. Acest eveniment rebusie tratat într-o intrerupere externă, pe frontul ascrescător al semnalului provenit de la buton. Resetarea contorului se va face prin intermediul celui de-al 3 buton. Aceasta va fi citit fără a utiliza intreruperi externe. La apasarea acestui buton, valoarea din contor va deveni 0.