

## Laborator 3. Întreruperi

Întreruperile sunt semnale care dau comanda procesorului să se oprească din ce face în acel moment și să realizeze un proces cu prioritate mai mare. Acest proces cu prioritate mai mare se numește Interrupt Handler. Acest Interrupt Handler este ca și oricare funcție. Dacă este atașată o întrerupere atunci poate fi chemată când este activat un semnal de întrerupere. După această întrerupere procesul initial poate fi reluat.

Întreruperile pot fi generate de următoarele scenarii:

- Întreruperile externe pentru schimbarea unei stări de unul din pinii de întreruperi externi
- Întreruperi de timp de către unul din timerele Arduino

Beneficiul folosirii întreruperilor este că nu mai este nevoie să se scrie cod de repetare care verifică încontinuu dacă a aparut o condiție de întrerupere prioritară.

Arduino UNO folosește pinii 2 și 3 digitali pentru întrerupere.

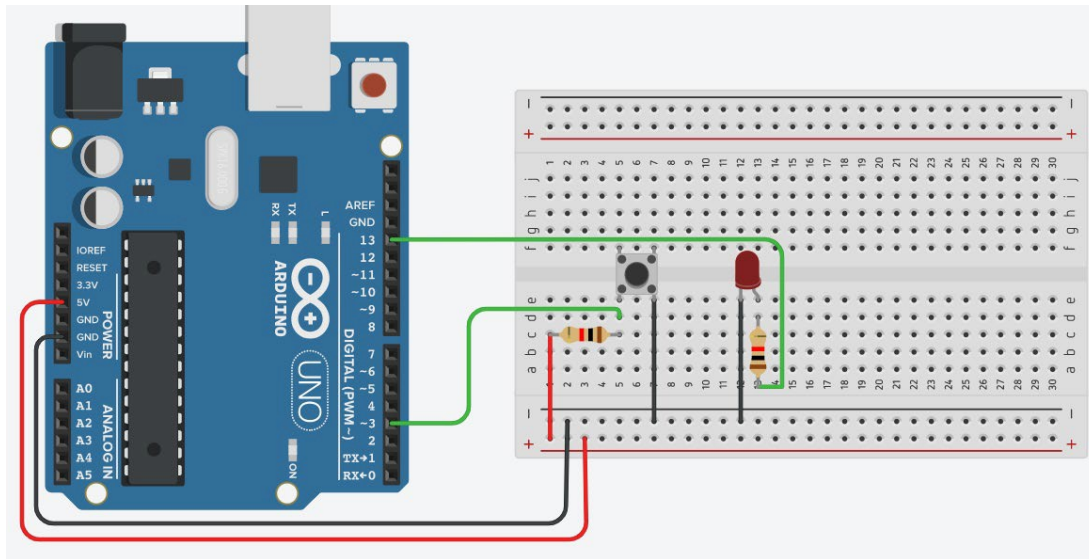
Mai multe informații pot fi găsite următorul link:

<https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/>

### Exemplu

#### Componente:

- Arduino UNO
- Buton
- LED



### Cod sursă fără întreruperi

```
int buttonPin = 3;    // the number of the pushbutton pin
int ledPin = 13;      // the number of the LED pin

int buttonState = 0;  // variable for reading the button status

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop() {
  buttonState = digitalRead(buttonPin);

  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

## Cod sursă cu întreruperi

```
//acum ne folosim de attachInterrupt()
int buttonPin = 3;
int ledPin = 13;

volatile int buttonState = 0; // variable for reading the button status
void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
  // Attach an interrupt to the ISR vector
  attachInterrupt(digitalPinToInterrupt(buttonPin), pin_ISR, CHANGE);
}

void pin_ISR() {
  buttonState = !buttonState;
}

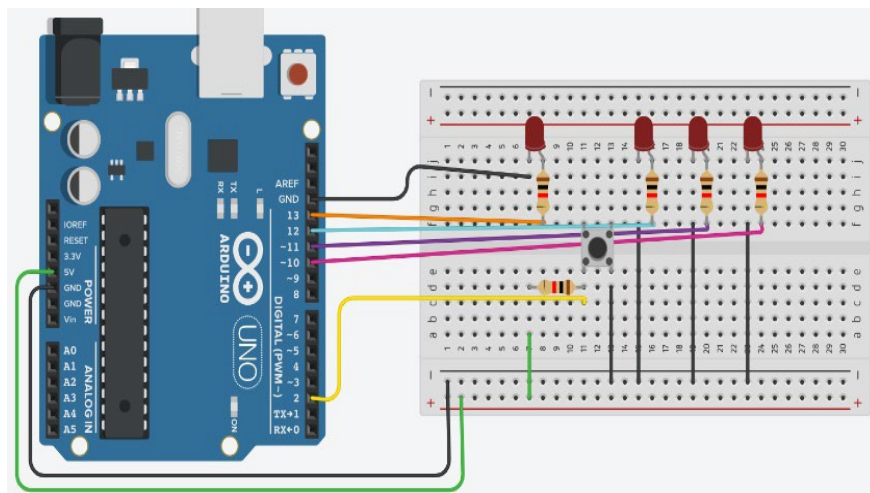
void loop() {
  digitalWrite(ledPin, buttonState);
}
```

## Exemplu

### Componente:

- Arduino UNO
- Buton
- LED-uri

Programul va folosi întreruperi pentru a aprinde cele 4 LED-uri.



## Codul sursă

```
int button1Pin=2;
int led1Pin=13;
int led2Pin=12;
int led3Pin=11;
int led4Pin=10;

int onoff=0;//0 lights are off 1 light are on
volatile byte state = HIGH;

//the setup function runs once when you press reset or power the board
void setup()
{
  pinMode(led1Pin, OUTPUT);
  pinMode(led2Pin, OUTPUT);
  pinMode(led3Pin, OUTPUT);
  pinMode(led4Pin, OUTPUT);
  pinMode(button1Pin, INPUT_PULLUP);
  Serial.begin(115200);
  attachInterrupt(digitalPinToInterrupt (button1Pin), blink1, CHANGE);
}

void blink1()
{
  state = !state;
}

//the loop function runs over and over again forever
void loop()
{
  if (state==LOW) { //using the state variable that gets changed in the
    //interrupt function instead of a digitalRead ()
    //turn LEDs on :
    Serial.println("on");
    digitalWrite (led1Pin, HIGH);
    digitalWrite (led2Pin, HIGH);
    digitalWrite (led3Pin, HIGH);
    digitalWrite (led4Pin, HIGH);
    onoff=1;
  }

  else if (state==HIGH){
    //using the state variable that gets changed in the interrupt function instead of a digitalRead ()
    //turn LEDs off:
    if(onoff=1){
      digitalWrite(led1Pin, LOW);
      digitalWrite(led2Pin, LOW);
      digitalWrite(led3Pin, LOW);
      digitalWrite(led4Pin, LOW);
      onoff=0;
    }
  }
}
```

```

digitalWrite(led1Pin, HIGH); // turn the LED on (HIGH is the voltage level)
delay (1000); // wait for a second

if(state==LOW)return; //returns back to the top of the main loop
    digitalWrite(led1Pin, LOW); // turn the LED off by making the voltage LOW
    digitalWrite(led2Pin, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second

if(state==LOW )return ; //returns back to the top of the main loop
    digitalWrite (led2Pin, LOW); // turn the LED off by making the voltage LOW
    digitalWrite (led3Pin, HIGH); // turn the LED on (HIGH is the voltage level)
    delay (1000); // wait or a second

if(state==LOW)return;
digitalWrite(led3Pin, LOW); // turn the LED off by making the voltage LOW
digitalWrite(led4Pin, HIGH); // turn the LED on (HIGH is the voltage level)
delay (1000); // wait for a second"

if(state=LOW) return; //returns back to the top of the main loop
digitalWrite(led4Pin, LOW); // turn the LED off by making the voltage LOW

}

}

```

---

## Exercitiu

### Componente:

- Arduino Uno
- 2 LED-uri RGB sau 4 LED-uri de diferite culori (pentru exterior și interior)
- 2 senzori pentru temperatura
- 1 buton

### Cerințe

#### Ansamblul Arduino:

- Când sistemul este alimentat de la portul USB și utilizatorul va apăsa pe buton (se va folosi intrerupere), sistemul va porni și se vor realiza următoarele acțiuni:
  - va măsura temperatura din exteriorul serei și va compara cu valorile normale:
    - dacă valorile citite sunt normale atunci culoarea verde a LED-ului RGB de exterior se va aprinde
    - dacă valorile citite nu sunt normale atunci se va aprinde culoarea albastru a LED-ului RGB de exterior dacă este prea rece sau culoarea rosu dacă este prea cald
  - va măsura temperatura din interiorul serei și va compara cu valorile normale:
    - dacă valorile citite sunt normale atunci culoarea verde a LED-ului RGB de interior se va aprinde
    - dacă valorile citite nu sunt normale atunci se va aprinde culoarea albastru a LED-ului RGB de interior dacă este prea rece sau culoarea rosu dacă este prea cald.