

Componente Arduino

Prezentare generală a plăcii, senzorilor și motoarelor folosite frecvent în proiecte de electronica

Autor: Vilcu David – clasa a XI-a B

Profesor coordonator: Melinte Elena-Carmen

1. Plăcuța Arduino



Arduino este o placă de dezvoltare bazată pe un microcontroller. Rolul ei este să citească semnale de la senzori, să proceseze aceste date conform unui program scris de utilizator și să controleze alte componente (LED-uri, motoare, ecrane etc.) pe baza rezultatelor.

Cum funcționează

Placa are pini digitali (folosiți pentru semnale de tip HIGH/LOW, adică 0V sau 5V) și pini analogici (care pot citi valori variabile de tensiune, convertite intern într-un număr între 0 și 1023 printr-un convertor analog-digital). Microcontrolerul execută codul încărcat de la calculator prin portul USB, folosind un bootloader care permite reprogramarea plăcii fără echipament suplimentar.

Structura unui program (sketch)

Codul Arduino este scris în C/C++ și are întotdeauna două funcții obligatorii: `setup()`, care rulează o singură dată la pornire (pentru inițializări, ex: `pinMode()`), și `loop()`, care se repetă continuu cât timp placa este alimentată.

```
void setup() {  
  pinMode(13, OUTPUT); // configurăm pinul 13 ca ieșire  
  Serial.begin(9600); // pornim comunicarea serială  
}
```

```
void loop() {  
  digitalWrite(13, HIGH); // aprindem LED-ul de pe placă
```

```

delay(1000);      // așteptăm 1 secundă
digitalWrite(13, LOW); // stingem LED-ul
delay(1000);
}

```

2. Senzorul ultrasonic (HC-SR04)



HC-SR04 este un senzor folosit pentru măsurarea distanței până la un obstacol, pe baza timpului de propagare a unui semnal ultrasonic (unde sonore cu frecvență peste 20 kHz, inaudibile pentru om).

Cum funcționează

Senzorul are doi pini principali: Trig (declanșare) și Echo (recepție). Placa trimite un impuls scurt pe Trig, senzorul emite un puls ultrasonic, iar acesta se reflectă din obstacol și revine la senzor. Pinul Echo rămâne în starea HIGH exact cât durează dus-întors al semnalului. Din acest timp și viteza sunetului (~343 m/s) se calculează distanța.

```

long durata;
float distanta;

```

```

digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

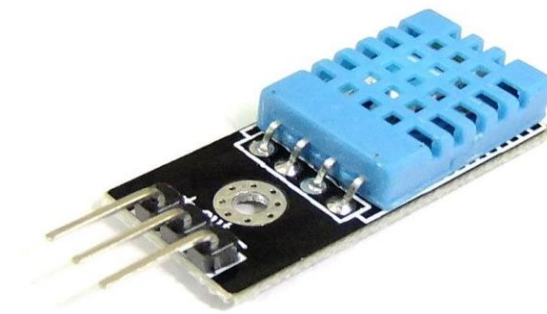
```

```

durata = pulseIn(echoPin, HIGH);
distanta = durata * 0.034 / 2; // cm

```

3. Senzorul de temperatură și umiditate (DHT11)



DHT11 măsoară simultan temperatura și umiditatea relativă a aerului. Este folosit frecvent în stații meteo simple, sisteme de monitorizare a mediului sau proiecte cu alertă la depășirea unui prag de temperatură.

Cum funcționează

Senzorul are în interior un termistor (pentru temperatură) și un senzor capacitiv de umiditate. Datele sunt transmise pe un singur fir digital, printr-un protocol propriu, sub formă de pachete de 40 de biți. Din cauza acestui protocol special, citirea se face de obicei cu ajutorul unei biblioteci dedicate (ex: DHT.h).

```
#include <DHT.h>
DHT dht(2, DHT11); // pin 2, tip DHT11

void setup() { dht.begin(); }

void loop() {
  float temp = dht.readTemperature();
  float umid = dht.readHumidity();
  Serial.print(temp); Serial.print(" C, ");
  Serial.print(umid); Serial.println(" %");
  delay(2000);
}
```

4. Servo motorul



Servo motorul este un motor capabil să se poziționeze la un unghi precis (de obicei între 0° și 180°), spre deosebire de un motor DC obișnuit care doar se rotește continuu. Este folosit pentru mecanisme de direcționare, brațe robotice, sisteme de tip radar (rotirea unui senzor ultrasonic) etc.

Cum funcționează

În interior are un motor DC mic, un set de angrenaje de reducere și un potențiomtru care indică poziția curentă a axului, folosit într-o buclă de control internă. Poziția dorită este transmisă printr-un semnal PWM (Pulse Width Modulation): lățimea impulsului (de obicei între 1 și 2 milisecunde, repetat la fiecare 20 ms) determină unghiul la care se poziționează axul.

```
#include <Servo.h>
Servo myServo;

void setup() {
  myServo.attach(9); // pin PWM
}

void loop() {
  myServo.write(90); // poziționează la 90 grade
  delay(1000);
  myServo.write(0);
  delay(1000);
}
```

5. Motorul pas cu pas (stepper motor)



Motorul pas cu pas (stepper motor) este un tip special de motor care, spre deosebire de motorul DC, se rotește în pași mici și egali (de exemplu 5.625° per pas la modelul 28BYJ-48). Acest lucru permite poziționarea axului cu precizie ridicată, fără a avea nevoie de un senzor de feedback, fiind folosit în imprimante 3D, scanere, sisteme CNC sau mecanisme de focalizare.

Cum funcționează

În interior, motorul are mai multe bobine (înfășurări) dispuse în jurul unui rotor cu magneți permanenți. Alimentând bobinele într-o anumită secvență, câmpul magnetic se deplasează treptat, iar rotorul “urmărește” acest câmp pas cu pas. Deoarece Arduino nu poate furniza direct curentul necesar bobinelor, se folosește o placă driver cu circuitul ULN2003 (un array de tranzistoare Darlington), care amplifică semnalele de control de la placă și alimentează cele patru bobine ale motorului în secvența corectă.

```
#include <Stepper.h>
#define PASI_PE_ROTATIE 2048
Stepper motor(PASI_PE_ROTATIE, 8, 10, 9, 11);

void setup() {
  motor.setSpeed(10); // rotații pe minut
}

void loop() {
  motor.step(PASI_PE_ROTATIE); // o rotație completă
  delay(1000);
}
```