

Arduino και Ρομποτική

Fossaegean
Arduino Day
2024

fossaegean

Γιατί μας ενδιαφέρει στη Ρομποτική;

- Ευκολία χρήσης
- Χαμηλό κόστος
- Κοινότητα και Υποστήριξη.
- Επεκτασιμότητα.
- Εκπαιδευτική Χρήση

Βασικά Χαρακτηρίσματα

- Κινητήρες/ Actuators
- Αισθητήρες
- Solenoids

Κινητήρες/Actuators

4 Είδη:

- **Servo**
- **DC**
- **Stepper**
- **Γραμμικοί actuators**

* Έξτρα εξάρτημα ο οδηγός κινητήρων.

Χαρακτηριστικά Servo

- Συγκεκριμένο εύρος κινήσεων
- Ελεγχόμενη ταχύτητα
- Δύναμη
- Ελεγχόμενη κατανάλωση
- Τηλεκατευθυνόμενα, βραχίονες



Χαρακτηριστικά DC

- Ρυθμιζόμενη ταχύτητα
- Ηλεκτροκινητήρας
- Περιστροφική κίνηση
- Ανεμιστήρες, ρόδες
- Κίνηση ανάλογη της τάσης



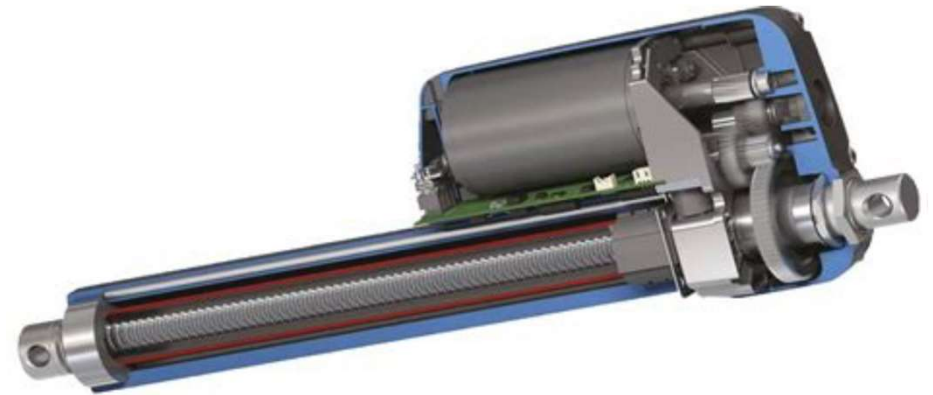
Χαρακτηριστικά Stepper

- Περιορισμένη κίνηση
- Υψηλή ακρίβεια
- Έλεγχος κινήσεων
- 3d Εκτυπωτές, CNC, κάμερες
- Μεγαλύτερη κατανάλωση από servo
- Αργοί



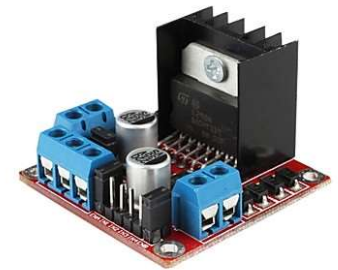
Γραμμικοί actuators

- Κίνηση σε ευθεία γραμμή
 - Σωληνοειδές
 - Υδραυλικοί
 - Πνευματικοί
 - Ηλεκτρικοί
-
- Κίνηση ρομποτικών βραχιόνων, ποδιών καθώς και διαφόρων μερών στα ρομπότ
 - Ιατρική , στα αναπηρικά καθίσματα και ιατρικές συσκευές
 - Βιομηχανικά συστήματα και παραγωγές



Οδηγοί κινητήρων

- Ενισχυτής ρεύματος
- Διαφορετικοί ανάλογα τον κινητήρα
- Αναγκαίοι για την κίνηση
- Ευκολία στην αλλαγή περιστροφής



Αισθητήρες

- Μεταβάλλουν το σήμα εξόδου ανάλογα της εισόδου που λαμβάνουν
 - Αισθητήρες φωτός (LDRs –Light Dependent Resistors)
μεταβάλλονται ανάλογα το φως που δέχονται
- Κατηγορίες
 - Active vs Passive
 - Ψηφιακοί vs Αναλογικοί

□ Active vs Passive

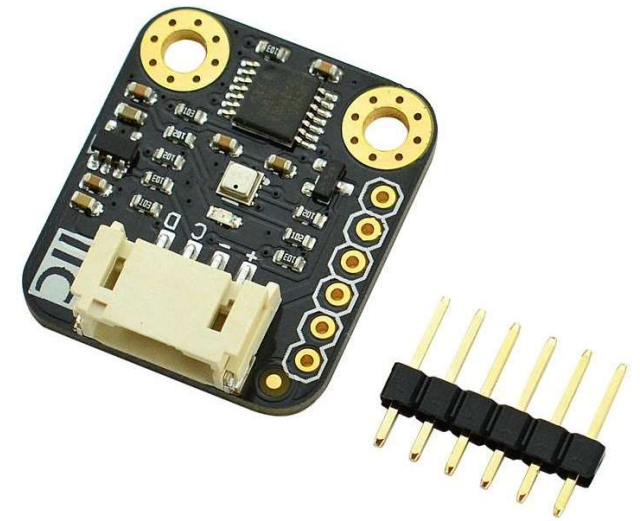
- Active: Απαιτούν εξωτερική τροφοδοσία για την παραγωγή σήματος
- Passive: Η ενέργεια για την παραγωγή του σήματος εξόδου παρέχεται από το μετρούμενο σήμα χωρίς τη χρήση τάσης διέγερσης

□ Ψηφιακοί vs. Αναλογικοί

- Ψηφιακοί αισθητήρες: το σήμα εξόδου του αισθητήρα είναι δυαδικό
- Αναλογικοί αισθητήρες: το σήμα εξόδου του αισθητήρα είναι συνεχές και ανάλογο του μετρούμενου σήματος

Αισθητήρες Περιβάλλοντος

- Ανιχνεύουν στοιχεία του περιβάλλοντος
 - Εργοστάσια, αποθήκες, εφοδιαστική αλυσίδα, γεωργία, υγεία
 - Υγρασίας
 - Θερμοκρασίας
 - Ατμοσφαιρική πίεση
 - Υγρασία Εδάφους
 - Βροχής



Χώρου/ Απόστασης

- Ανίχνευση κοντινών στόχων χωρίς φυσική επαφή
- Αποφυγή εμποδίων

Είδη:

- Υπέρυθρες
- Υπέρηχων
- Λείζερ



Αισθητήρες κίνησης

- Γυροσκόπιο
- Πυξίδα
- Επιταχυνσιόμετρο



Αισθητήρες Υπέρυθρων

- Ανίχνευση αλλαγών στο περιβάλλον τους
- Χρώμα
- Θερμοκρασία
- Ήχος
- Θερμότητα

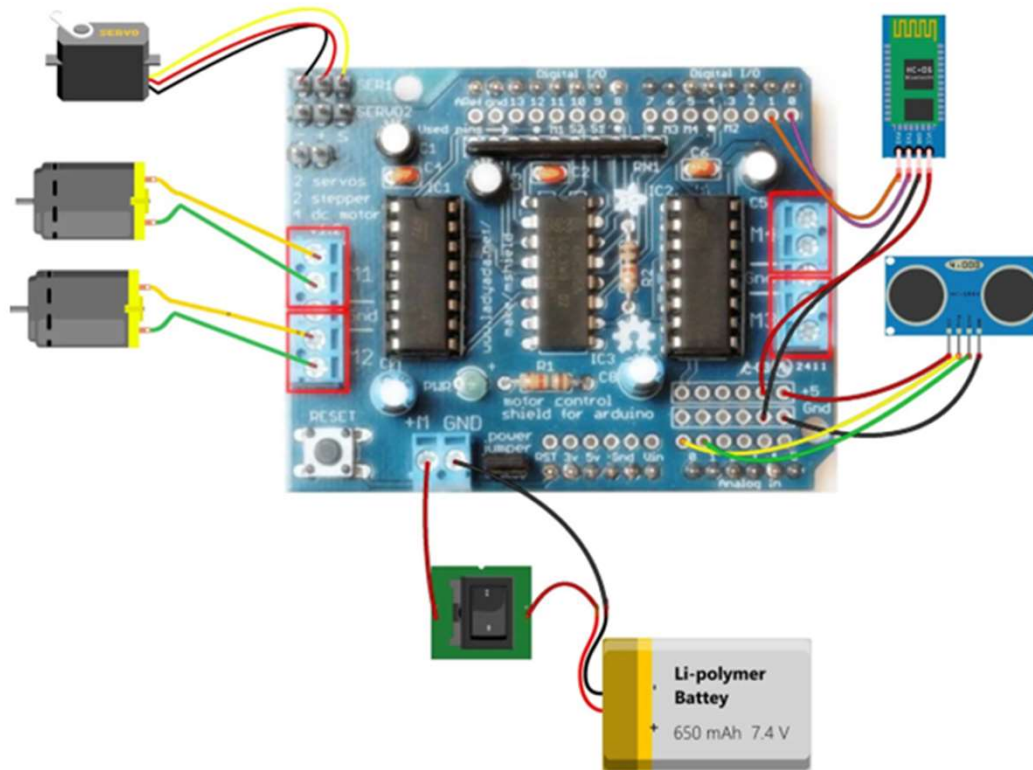


Άλλα είδη αισθητήρων

- Βιομετρικοί
- Υγρών/Στάθμης
- Πίεσης
- Μαγνητικού πεδίου
- Αερίων



Παράδειγμα ενός ρομποτικού οχήματος



Εξαρτήματα Υλοποίησης του προηγούμενου

- Arduino uno
- Arduino motor Shield
- 2 DC κινητήρες
- 1 Servo κινητήρας
- Διακόπτης on/off
- Ultrasonic Sensor
- Bluetooth module HC-06
- Μπαταρία

Ενδεικτικός κωδικας

```
#include <AFMotor.h>    // Library for controlling DC motors
#include <Servo.h>       // Library for controlling servo motors
#include <NewPing.h>     // Library for ultrasonic sensor

#define TRIGGER_PIN 12 // Pin connected to the trigger pin of the ultrasonic sensor
#define ECHO_PIN    11 // Pin connected to the echo pin of the ultrasonic sensor
#define MAX_DISTANCE 200 // Maximum distance in centimeters for the ultrasonic sensor

AF_DCMotor motor1(1); // Motor 1 connected to M1 output on the motor shield
AF_DCMotor motor2(2); // Motor 2 connected to M2 output on the motor shield
Servo myservo;        // Servo motor object
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // Ultrasonic sensor object
```

```
void setup() {
  Serial.begin(9600); // Initialize serial communication at 9600 baud

  motor1.setSpeed(255); // Set the speed of motor 1 to maximum
  motor2.setSpeed(255); // Set the speed of motor 2 to maximum

  myservo.attach(9); // Attach the servo motor to pin 9
}

void loop() {
  int distance = readDistance(); // Read distance from ultrasonic sensor

  if (distance > 0 && distance <= 20) { // If obstacle detected within 20cm
    stopMotors(); // Stop motors
    delay(500); // Delay for stability
    reverse(); // Move backward
  }
}
```

```
delay(1000);    // Delay for 1 second
  stopMotors(); // Stop motors
  delay(500);   // Delay for stability
  turn();      // Turn
} else {
  forward();   // Move forward
}
```

```
  delay(100); // Small delay for stability
}
```

```
int readDistance() {
  delay(50); // Small delay for stability
  int distance = sonar.ping_cm(); // Send ultrasonic pulse and get distance in cm
  Serial.print("Distance: "); // Print distance label
  Serial.println(distance); // Print distance value
  return distance; // Return distance value
}
```

```
void forward() {  
  motor1.run(FORWARD); // Set motor 1 to move forward  
  motor2.run(FORWARD); // Set motor 2 to move forward  
}
```

```
void reverse() {  
  motor1.run(BACKWARD); // Set motor 1 to move backward  
  motor2.run(BACKWARD); // Set motor 2 to move backward  
}
```

```
void turn() {  
  myservo.write(180); // Turn servo motor to 180 degrees  
  delay(1000); // Delay for 1 second  
  myservo.write(90); // Turn servo motor to 90 degrees  
  delay(1000); // Delay for 1 second  
}
```

```
void stopMotors() {  
  motor1.run(RELEASE); // Release motor 1 (stop)  
  motor2.run(RELEASE); // Release motor 2 (stop)  
}
```



Απορίες;