

Αυτόνομα κινούμενα ρομπότ και εφαρμογές

Μέλη Ομάδας 27:
Μάριος Τσαουσίδης
Κλεονίκη Φραγκάκη
Αναστασία Μαρμάρου



Το έργο αφορά την ανάπτυξη ενός αυτόνομου ρομποτικού οχήματος ικανού να εκτελεί συγκεκριμένες λειτουργίες, όπως αποφυγή εμποδίων, διόρθωση πορείας και σχεδίαση γεωμετρικών σχημάτων, χρησιμοποιώντας αισθητήρες και αλγορίθμους ελέγχου. Η κατασκευή βασίζεται στον μικροελεγκτή **Maker Pi RP2040**, γνωστό για την ισχύ, την ευελιξία και τη χαμηλή κατανάλωση ενέργειας.



Εξαρτήματα συστήματος

1. Ο μικροελεγκτής **Maker Pi RP2040** είναι βασικός εγκέφαλος του συστήματος που εκτελεί τον κώδικα και ελέγχει τους αισθητήρες και τους κινητήρες.
2. Αισθητήρες
 - **MPU6050** είναι γυροσκόπιο και επιταχυνσιόμετρο για μέτρηση κινήσεων και προσανατολισμού.
 - **HC-SR04** είναι αισθητήρας υπερήχων για μέτρηση αποστάσεων και ανίχνευση εμποδίων.
3. **DC motors** είναι κινητήρες συνεχούς ρεύματος για την κίνηση του ρομπότ.
4. **SG90 Servo motor** είναι σερβοκινητήρας για ελεγχόμενη περιστροφή, π.χ., για την κίνηση αισθητήρων.
5. **Μπαταρίες** είναι για την παροχή ενέργειας για τον μικροελεγκτή, τους αισθητήρες και τους κινητήρες.

Αλγόριθμοι και θεωρία

Θεωρία Αισθητήρων και Ελέγχου

- **MPU6050 - Γυροσκόπιο & Επιταχυνσιόμετρο**

Το γυροσκόπιο ανιχνεύει τη γωνιακή ταχύτητα ($^{\circ}/s$), επιτρέποντας την ανίχνευση περιστροφών.

Το επιταχυνσιόμετρο μετρά γραμμικές επιταχύνσεις, παρέχοντας δεδομένα για κλίσεις.

- **HC-SR04 - Υπερήχων**

Λειτουργία βάσει ανακλώμενων κυμάτων. Μετρά την απόσταση από αντικείμενα χρησιμοποιώντας τη φόρμουλα: $\text{Απόσταση} = (\text{Χρόνος μετάβασης} \times \text{Ταχύτητα ήχου}) / 2$

1) Αλγόριθμοι

Ο αλγόριθμος παίρνει δεδομένα από το γυροσκόπιο και υπολογίζει την απόκλιση από την ευθεία.

Ανιχνεύει το σημάδι της απόκλισης (θετικό/αρνητικό) και ρυθμίζει τις ταχύτητες των τροχών:

- Αυξάνει την ταχύτητα στον αργό τροχό.
- Μειώνει την ταχύτητα στον ταχύτερο τροχό.

2) Αποφυγή Εμποδίων

Μετρά αποστάσεις μπροστά, δεξιά και αριστερά:

- Εάν υπάρχει εμπόδιο, επιλέγεται η κατεύθυνση με τη μεγαλύτερη απόσταση.
- Ο σεβροκινητήρας περιστρέφει τον αισθητήρα υπερήχων για σάρωση.

3) Σχεδίαση Τριγώνου

Επαναλαμβάνεται η διαδικασία:

- Ευθεία κίνηση για 2 δευτερόλεπτα.
- Περιστροφή 60° (χρησιμοποιώντας το γυροσκόπιο για ακρίβεια).



Δομή Κώδικα

Τα αρχεία που αποτελούν τη βάση του συστήματος είναι τα εξής:

Main.py

Ο κύριος κώδικας που περιλαμβάνει τη λογική λειτουργίας του ρομπότ, τη διόρθωση ευθυγράμμισης, και τις βασικές λειτουργίες όπως η αποφυγή εμποδίων και η σχεδίαση τριγώνου.

Imu.py

Η βιβλιοθήκη για την αλληλεπίδραση με τον αισθητήρα **MPU6050**.

Περιλαμβάνει μεθόδους για την ανάγνωση δεδομένων γυροσκοπίου και επιταχυνσιόμετρου.

Vector3d.py

Ένα εργαλείο που χρησιμοποιείται για τον χειρισμό τρισδιάστατων δεδομένων (x, y, z) που επιστρέφει το **MPU6050**.



Έλεγχος Μέσω Πληκτρολογίου

```
python
# -- Παράμετροι σχεδίασης --

def main():
    try:
        print("Καλιμπράρισμα...")

        gyro_zero = calibrate_gyro()
        print("Πατήστε 'A', 'B', ή 'D'.")

        while True:
            key = read_key()

            if key == 'A':
                modify_parameter("side_time")
            elif key == 'B':
                modify_parameter("turn_time")
            elif key == 'D':
                draw_triangle(side_time, turn_time, gyro_zero)

    except KeyboardInterrupt:
        stop()

        print("Ρομπότ σταμάτησε.")
```

Η συνάρτηση `def main()` περιλαμβάνει τις ρυθμίσεις για τον χειρισμό και την πλοήγηση του ρομπότ μέσω του πληκτρολογίου. Ο χρήστης μπορεί να πατήσει ένα πλήκτρο ανάμεσα στα γράμματα A, B και D για να επιλέξει την ρύθμιση.

Για κάθε γράμμα:

- A: Ρύθμιση απόστασης της πλευράς του τριγώνου σε εκατοστά μέσω του χρόνου που θα κάνει σε δευτερόλεπτα. Ισχύει ότι:
Απόσταση σε εκ. = δευτερόλεπτα / 2
- B: Ρύθμιση του χρόνου στροφής για τον υπολογισμό των μοιρών του τριγώνου.
- D: Ξεκινάει τον σχηματισμό του τριγώνου.

Σχεδίαση Τριγώνου

```
python
# -- Σχεδίαση τριγώνου --

def draw_triangle(side_length_cm, gyro_zero):
    side_duration = side_length_cm / 20
    for _ in range(3):
        start_time = utime.ticks_ms()
        while utime.ticks_ms() - start_time < side_duration * 1100:
            correct_alignment(gyro_zero) # Συνεχής διόρθωση ευθυγράμμισης
            if ultra() < 10:
                stop()
                avoid_obstacle()
                continue
            time.sleep(0.05)
        stop()
        time.sleep(0.5)
        turn_right(0.525)
    stop()
    time.sleep(0.5)
```

Η συνάρτηση `draw_triangle` κινεί ένα ρομπότ σε πορεία που σχηματίζει τρίγωνο, διορθώνοντας συνεχώς την ευθυγράμμιση και αποφεύγοντας εμπόδια. Δέχεται το μήκος πλευράς του τριγώνου και την τιμή αναφοράς του γυροσκοπίου.

Για κάθε πλευρά:

- Υπολογίζεται ο χρόνος κίνησης με βάση την ταχύτητα του ρομπότ.
- Το ρομπότ κινείται ευθεία, διορθώνει πορεία με το γυροσκόπιο και σταματά αν ανιχνεύσει εμπόδιο, καλώντας τη συνάρτηση αποφυγής.
- Μετά την κίνηση, κάνει παύση, περιστρέφεται δεξιά για να σχηματίσει γωνία 60° και συνεχίζει.

Με τρεις επαναλήψεις, το ρομπότ ολοκληρώνει την πορεία του, διατηρώντας ευθυγράμμιση και αποφεύγοντας εμπόδια, σχηματίζοντας ένα ισόπλευρο τρίγωνο.

Αποφυγή Εμποδίων

```
python
# -- Λογική αποφυγής εμποδίων --
def avoid_obstacle():
    stop()
    backward(0.2)
    move_servo(0)
    distance_right = ultra()
    move_servo(180)
    distance_left = ultra()
    move_servo(90)
    distance_center = ultra()
    if distance_right > distance_left and distance_right > 10:
        print("Αποφυγή δεξιά...")
        turn_right(0.5)
    elif distance_left > 10:
        print("Αποφυγή αριστερά...")
        turn_left(0.5)
    else:
        print("Όπισθεν για επανεκτίμηση...")
        backward(0.5)
```

Η συνάρτηση `avoid_obstacle` διαχειρίζεται την αποφυγή εμποδίων χρησιμοποιώντας αισθητήρες απόστασης και σερβοκινητήρα για εκτίμηση των δυνατών κατευθύνσεων.

Αναλυτικά:

1. Το ρομπότ σταματά και κάνει μικρή όπισθεν για να δημιουργήσει χώρο.
2. Στρέφει τον σερβοκινητήρα προς τα δεξιά, αριστερά και στο κέντρο, καταγράφοντας τις αντίστοιχες αποστάσεις με τον αισθητήρα υπερήχων.
3. Συγκρίνει τις αποστάσεις:
 - Αν η δεξιά πλευρά έχει τη μεγαλύτερη απόσταση (>10 cm), στρέφεται δεξιά.
 - Αν η αριστερή πλευρά έχει απόσταση >10 cm, στρέφεται αριστερά.
 - Αν καμία πλευρά δεν είναι εφικτή, κάνει όπισθεν για να επανεκτιμήσει.

Η λογική εξασφαλίζει ότι το ρομπότ επιλέγει την πιο ασφαλή κατεύθυνση ή κάνει όπισθεν αν χρειάζεται.

Ευθυγράμμιση κίνησης

```
python
def correct_alignment(gyro_zero):
    gyro_data = imu.gyro.xyz
    gyro_x = gyro_data[0] - gyro_zero[0]
    gyro_y = gyro_data[1] - gyro_zero[1]
    gyro_z = gyro_data[2] - gyro_zero[2]
    print("Gyro Data (degrees/s): X: {:.2f}, Y: {:.2f}, Z: {:.2f}".format(gyro_x, gyro_y, gyro_z))

    alignment_threshold = 0.5 # Όριο για την ανίχνευση εκτροπής
    if gyro_z > alignment_threshold:
        print("Η εκτροπή είναι δεξιά. Διορθώνω...")
        set_motor_speed(50000, 60000)
    elif gyro_z < -alignment_threshold:
        print("Η εκτροπή είναι αριστερά. Διορθώνω...")
        set_motor_speed(60000, 50000)
    else:
        set_motor_speed(60000, 60000) # Ισορροπία ταχύτητας
```

Η συνάρτηση `correct_alignment` διορθώνει την πορεία του ρομπότ χρησιμοποιώντας δεδομένα από το γυροσκόπιο για να ανιχνεύσει εκτροπές.

Αναλυτικά:

1. Ο υπολογισμός εκτροπής αφαιρεί τις αρχικές τιμές του γυροσκοπίου (`gyro_zero`) από τα τρέχοντα δεδομένα και υπολογίζει την εκτροπή στις τρεις διαστάσεις.
2. Η ανίχνευση εκτροπής ελέγχει την τιμή της εκτροπής γύρω από τον άξονα Z:
 - Αν είναι μεγαλύτερη από το όριο 0.5, θεωρεί ότι το ρομπότ στρέφεται δεξιά και μειώνει την ταχύτητα του αριστερού κινητήρα.
 - Αν είναι μικρότερη από -0.5, θεωρεί ότι στρέφεται αριστερά και μειώνει την ταχύτητα του δεξιού κινητήρα.
3. Στη διατήρηση ισορροπίας, αν η εκτροπή είναι εντός ορίου, διατηρεί ίσες ταχύτητες στους κινητήρες.

Κόστος ολικής κατασκευής

	Προϊόν/Εξάρτημα	Ηλεκτρονικό Κατάστημα	Ποσότητα x Τιμή
1.	SG90 Servo Micro Motor 9G	Μέσω E-shop Skroutz: Radel S.A.	1 x 2,90 €
2.	Raspberry Pi Maker Pi (RP2040 Microcontroller)	Nettop	1 x 12,90 €
3.	ElecFreaks HC-SR04 Ultrasonic Distance Module για Arduino	Μέσω E-shop Skroutz: Nettop	1 x 1,50 €
4.	Haitronic 2WD Smart Robot Car Chassis kit for Arduino <ul style="list-style-type: none">Διαστάσεις 22 x 14.7 cmΜεγιστο φορτίο: 1kg2 x κινητήρες(1:48)2 x 65mm τροχοίΡεύμα λειτουργίας 80-100 mA	Μέσω E-shop Skroutz: Top Electronics	1 x 8,68 €
5.	Optum Acrylic Ultrasonic Sensor Mounting Bracket for HC-SR04 Module	Μέσω E-shop Skroutz: Nettop	1 x 0,70 €
6.	4x4 Matrix 16 Buttons Keypad/Keyboard για Arduino	Μέσω E-shop Skroutz: Radel S.A.	1 x 1,60 €
7.	30cm Digital Sensor Cable 3p Ph2.0 To 3p Dupont	Μέσω E-shop Skroutz: Nettop	3 x 0,40 €
8.	GY-521 MPU-6050 Triple Axis Gyroscope & Accelerometer	Μέσω E-shop Skroutz: Nettop	1 x 4,60 €
9.	40P 10cm dupont wire male to female Jumper Breadboard Wires	Μέσω E-shop Skroutz: Nettop	1 x 2,00 €
10.	Μπαταρίες	—	1 x 3,00 €
11.	Μαρκαδόρος	—	1 x 2,00 €
12.	Μεταφορικά	Μέσω E-shop Skroutz	1 x 6,00 €
	Συνολική Αξία προϊόντων:		47,08 €

Σας ευχαριστούμε!