



ΡΟΜΠΟΤ ΓΙΑ ΤΟΝ ΚΑΘΑΡΙΣΜΟ ΣΙΝΤΡΙΒΑΝΙΩΝ

Αρχικό πρόβλημα

- Πολλές φορές μερικοί ασυνείδητοι άνθρωποι μολύνουν το περιβάλλον που ζούμε π. χ. εάν δούμε ένα κλασικό σιντριβάνι μιας πόλης έχει αρκετά σκουπίδια. Και για αυτόν τον λόγο έχουμε επιλέξει να λύσουμε αυτό το πρόβλημα.
- Μετά κάναμε ένα πείραμα:
Σκεφτήκαμε τα περισσότερα σκουπίδια που έχει ένα σιντριβάνι. Είδαμε ποια επιπλέουν και ποια όχι. Τελικά τα σκουπίδια που βυθίζονταν ήταν λιγότερα, δεύτερον ήταν δύσκολο να πάμε κάτω από την επιφάνεια του νερού. Έτσι αποκλείσαμε τα σκουπίδια που βρίσκονταν στο βυθό.



Η λύση μας

Η ιδέα μας είναι η κατασκευή ενός αυτόνομου πλωτού ρομποτικού οχήματος το οποίο θα αναλαμβάνει να μαζεύει τα σκουπίδια από τα σιντριβάνια του Πανεπιστημίου, ώστε να διασφαλίζεται η σωστή λειτουργία τους και έτσι να συνεισφέρουν και στην αισθητική αναβάθμιση του χώρου. Η τεχνολογική μας λύση θα παρουσιαστεί στην πανεπιστημιακή κοινότητα και θα δοκιμαστεί σε πραγματικές συνθήκες στα σιντριβάνια του πανεπιστημίου. Το ρομποτικό όχημα θα σαρώνει την επιφάνεια του σιντριβανιού συλλέγοντας τα φύλλα, τα πλαστικά και τα υπόλοιπα σκουπίδια. Έπειτα θα τα συγκεντρώνει σε ειδικά διαμορφωμένο κάδο μέσα στο σώμα του. Το ρομπότ επίσης θα έχει τη δυνατότητα να αναγνωρίζει πότε γέμισε ο κάδος και να τον αδειάζει, ενώ θα ενημερώνει τον χρήστη μέσω Bluetooth για την κατάσταση λειτουργίας του. Το άδειασμα του κάδου θα μπορεί να γίνεται χωρίς να σταματήσει το ρομπότ τη λειτουργία του.

Υλικά που χρησιμοποιήθηκαν:

- Κινητήρα dc (2 φορές)



- Διακόπτες (4 φορές)



- Λαμπτήρες led(2 φορές)



- Ultrasonic (1 φορά)



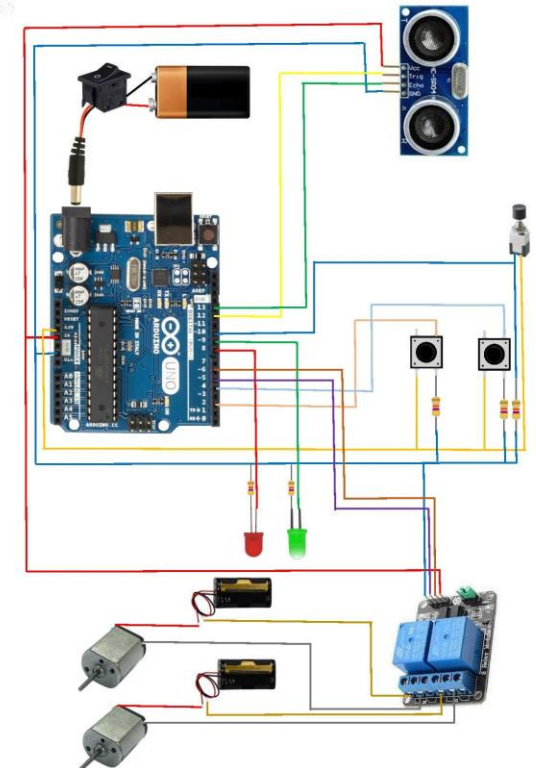
- μπαταρίες 1.5 Volt (η κάθε μια)(10 φορές)



- Arduino UNO



- RELE(1 μία φορά)



Στρατηγική

- Πατάμε το κουμπί on/off
- Κινείται το ρομπότ ελεύθερα στο σιντριβάνι
- Εάν οι διακόπτες που έχει, ακουμπήσουν τοίχο
Τότε: αναστροφή προς τα δεξιά ή τα αριστερά
- Εάν ο αισθητήρας ultrasonic δει όριο
Τότε: πηγαίνει στην άκρη κοντά στον άνθρωπο
Περιμένει ένα λεπτό για να πάρει ο άνθρωπος τα απορρίμματα
Μετά συνεχίζει να μαζεύει τα απορρίμματα

*Αυτό γίνεται συνέχεια

*Εάν οι αισθητήρες δεν βρουν κάτι συνεχίζει η επανάληψη

Ο κώδικας

//Εισαγωγή της κατάλληλης συνάρτησης για να λειτουργήσει ο αισθητήρας

```
#include "Ultrasonic.h"
```

//Μεταβλητές για τα αποτελέσματα του αισθητήρα

```
int Apostasi;
```

```
long Xronos;
```

//Πινάκια που συνδέουμε τον αισθητήρα στο Arduino

```
int EchoPin = 13;
```

```
int TrigPin = 12;
```

//Πινάκια που συνδέουμε τα κουμπάκια στο Arduino

```
int buttonPin1 = 2;
```

```
int buttonPin2 = 10;
```

```
int buttonPin3 = 4;
```

//Πινάκια που συνδέουμε τα ρελέ για τους κινητήρες στο Arduino

int dcPin1 = 5;

int dcPin2 = 6;

//Πινάκια που συνδέουμε τα δύο λαμπάκια στο Arduino

int led1 = 8;

int led2 = 9;

//Μεταβλητές που αποθηκεύουν την κατάσταση των κουμπιών (0 κλειστό - 1 ανοιχτό)

int buttonState1 = 0;

int buttonState2 = 0;

int buttonState3 = 0;

//Μεταβλητές που δείχνουν σε ποιά κατάσταση βρίσκεται το πρόγραμμα (γίνονται 1 όταν ενεργοποιούνται)

int Full = 0;

int Pressed_Left = 0;

int Pressed_Right = 0;

//Μεταβλητή για το κλείσιμο του ρομπότ (όταν πατηθεί το αντίστοιχο κουμπί γίνεται 0)

int start_stop = 1;

```
void setup() {  
    //Ορίζουμε τα πινάκια για τον αισθητήρα  
    pinMode(EchoPin,INPUT);  
    pinMode(TrigPin,OUTPUT);  
  
    //Ορίζουμε τα πινάκια για τα κουμπάκια  
    pinMode(buttonPin1, INPUT);  
    pinMode(buttonPin2, INPUT);  
    pinMode(buttonPin3, INPUT);  
  
    //Ορίζουμε τα πινάκια για τα ρελέ του κινητήρα  
    pinMode(dcPin1, OUTPUT);  
    pinMode(dcPin2, OUTPUT);  
  
    //Ορίζουμε τα πινάκια για τα λαμπάκια  
    pinMode(led1, OUTPUT);  
    pinMode(led2, OUTPUT);  
  
    //Ορίζουμε τον χρόνο ανανέωσης της σειριακής οθόνης  
    Serial.begin(9600);  
}
```


//Αρχικά σταματάμε τα μοτέρ και περιμένουμε μέχρι να πατηθεί το κουμπί για να ξεκινήσει το ρομπότ

```
Serial.println("Waiting...");
```

```
digitalWrite(dcPin1,HIGH);
```

```
digitalWrite(dcPin2,HIGH);
```

```
buttonState2 = digitalRead(buttonPin2);
```

//Διαβάζουμε την κατάσταση του κουμπιού μέχρι να πατηθεί

```
while (buttonState2 == 0){
```

```
    buttonState2 = digitalRead(buttonPin2);
```

```
}
```

//Αφού πατηθεί μετά από 1 δευτερόλεπτο αρχίζουν να κινούνται οι κινητήρες

```
delay(1000);
```

```
}
```

```
void loop() {  
    //Αν πατηθεί το κουμπί κατά τη λειτουργία του ρομπότ το ρομπότ θα  
    σταματήσει  
    buttonState2 = digitalRead(buttonPin2);  
    if (buttonState2 == 1){  
        start_stop = 0;  
    }  
  
    if (start_stop == 1){  
        //Διαβάζουμε τη τιμή του αισθητήρα Ultrasonic  
        //Στέλνουμε έναν παλμό και μετρώντας τον χρόνο μέχρι να  
        ξαναγυρίσει  
        //υπολογίζουμε την απόσταση.
```

```
delayMicroseconds(5);
```

```
digitalWrite(TrigPin,HIGH);
```

```
delayMicroseconds(10);
```

```
digitalWrite(TrigPin,LOW);
```

```
Xronos = pulseIn(EchoPin,HIGH);
```

```
Apostasi = (Xronos/2)/29.1;
```

```
//Διαβάζουμε τα κουμπάκια
```

```
buttonState1 = digitalRead(buttonPin1);
```

```
buttonState2 = digitalRead(buttonPin2);
```

```
buttonState3 = digitalRead(buttonPin3);
```

```
//Τυπώνουμε τη κατάσταση που βρίσκονται τα κουμπάκια 1,2,3
//καθώς και την απόσταση που έχει ο ultrasonic από τα σκουπίδια, σε εκατοστά.
Serial.print("Button 1: ");
Serial.print(buttonState1);
Serial.print(" Button 2: ");
Serial.print(buttonState2);
Serial.print(" Button 3: ");
Serial.print(buttonState3);
Serial.print(" Apostasi: ");
Serial.println(Apostasi);
```

```
//Εάν η απόσταση που έχει ο αισθητήρας ultrasonic από τα σκουπίδια στα
δυχτάκια είναι μικρότερη από 10cm
if (Apostasi < 10){
    //τότε σημαίνει πως ο αποθηκευτικός χώρος έχει γεμίσει
    Full = 1;
}
```

```
else{
    //αλλιώς σημαίνει πως είναι άδειος
    Full = 0;
}
// Αποθη' κεύεται στη μνήμη η κατάσταση για το αριστερό κουμπάκι αν έχει πατηθεί ή
όχι
if (buttonState3 == 1){
    Pressed_Left = 1;
}
else{
    Pressed_Left = 0;
}
//Αποθηκεύεται στη μνήμη η κατάσταση για το δεξί κουμπάκι αν έχει πατηθεί ή όχι
if (buttonState1 == 1){
    Pressed_Right = 1;
}
else{
    Pressed_Right = 0;
}
```


//Εάν ο αποθηκευτικός χώρος είναι γεμάτος το κόκκινο λαμπάκι ανάβει και το πράσινο κλείνει'

```
if (Full == 1){
```

```
    Serial.println("Full State is ON");
```

```
    digitalWrite(led1,HIGH);
```

```
    digitalWrite(led2,LOW);
```

// Ενεργοποιούμε τους δύο κινητήρες μέχρι ένα από τα δυο μπροστινά κουμπάκια (1,3) να πατηθεί

```
while ((buttonState1 == 0)&&(buttonState3 == 0)){
```

```
    buttonState1 = digitalRead(buttonPin1);
```

```
    buttonState3 = digitalRead(buttonPin3);
```

```
    digitalWrite(dcPin1,LOW);
```

```
    digitalWrite(dcPin2,LOW);
```

```
}
```

//Εκτυπώνει στην οθόνη Waiting... και απενεργοποιούμε τους κινητήρες

```
Serial.println("Waiting...");
```

```
digitalWrite(dcPin1,HIGH);
```

```
digitalWrite(dcPin2,HIGH);
```

```
buttonState2 = digitalRead(buttonPin2);
```

```
//περιμένει μέχρι να πατήσουμε το κουμπί 2 για να ξαναξεκινήσει
```

```
while (buttonState2 == 0){
```

```
    buttonState2 = digitalRead(buttonPin2);
```

```
}
```

```
//Ο αποθηκευτικός χώρος είναι άδειος και περιμένει 1  
δευτερόλεπτο για να ξαναξεκινήσει
```

```
Full = 0;
```

```
delay(1000);
```

```
}
```

```
if (Pressed_Left == 1){  
    Serial.println("Left Pressed State is ON");  
    // Όσο το κουμπάκι 3 είναι πατημένο  
    while (buttonState3 == 1){  
        buttonState3 = digitalRead(buttonPin3);  
        //στρίβει δεξια  
        digitalWrite(dcPin1,HIGH);  
        digitalWrite(dcPin2,LOW);  
        //ανάβει το κόκκινο κουμπάκι και σβήνει το πράσινο  
        digitalWrite(led1,HIGH);  
        digitalWrite(led2,LOW);  
    }  
    //περιμένει 2 δευτερόλεπτα  
    delay(2000);  
}
```

```
//αλλιώς
else{
  //Όσο το κουμπάκι 1 είναι πατημένο
  if (Pressed_Right == 1){
    Serial.println("Right Pressed State is ON");
    while (buttonState1 == 1){
      buttonState1 = digitalRead(buttonPin1);
      //στρίβει δεξια
      digitalWrite(dcPin1,LOW);
      digitalWrite(dcPin2,HIGH);
      //και ανάβει το κόκκινο κουμπάκι και σβήνει το πράσινο
      digitalWrite(led1,HIGH);
      digitalWrite(led2,LOW);
    }
    //περίμενε 2 δευτερόλεπτα
    delay(2000);
  }
}
```

```
//αλλιώς
else{
    //αν δεν ισχύει τίποτα από αυτά ενεργοποιεί τους κινητήρες
    Serial.println("Zero State is ON");
    digitalWrite(dcPin1,LOW);
    digitalWrite(dcPin2,LOW);
    //και ανάβει το πράσινο κουμπάκι και σβήνει το κόκκινο
    digitalWrite(led1,LOW);
    digitalWrite(led2,HIGH);
}
}
}
//αλλιώς αν πατηθεί το κουμπί απενεργοποιούμε τους κινητήρες
else{
    Serial.println("All OFF");
    digitalWrite(dcPin1,HIGH);
    digitalWrite(dcPin2,HIGH);
}
```