

Αυτόνομο όχημα ηχητικής ξενάγησης



2ο Γυμνάσιο Μοσχάτου

Σχολικό έτος 2024 - 25

Περιγραφή του αυτόνομου οχήματος ηχητικής ξενάγησης

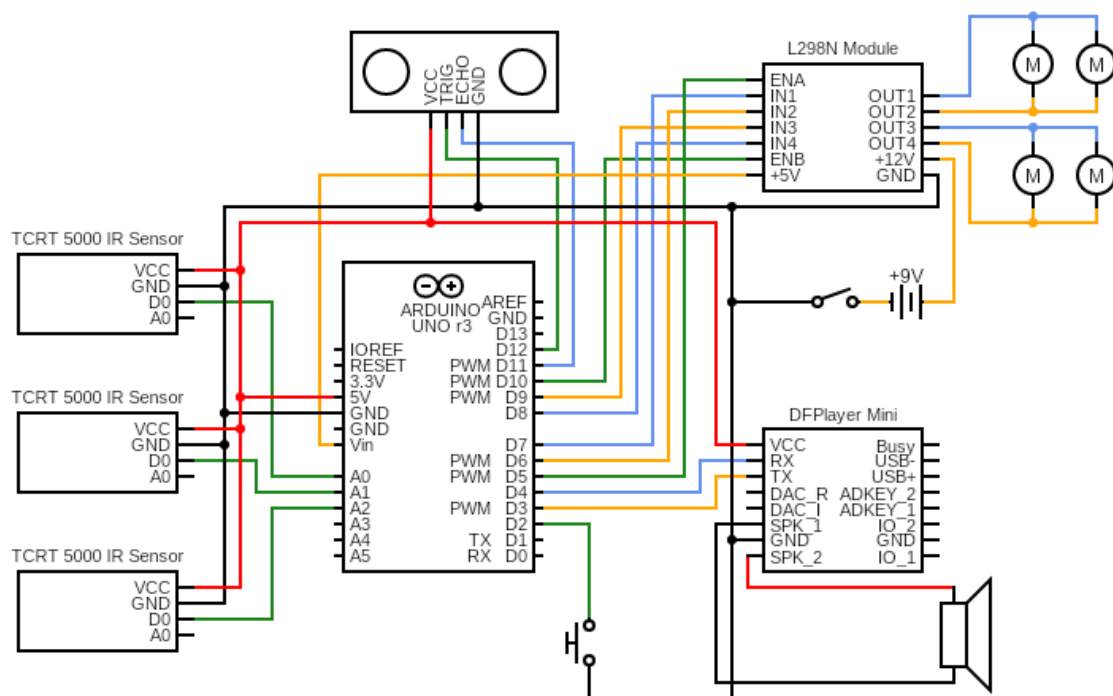
Το όχημα του αυτόνομου ξεναγού που σχεδιάσαμε έχει τη δυνατότητα να ακολουθεί μια μαύρη γραμμή στο δάπεδο του χώρου όπου θα κινηθεί. Για να σταματά στα σημεία που επιθυμούμε μπορεί να ανιχνεύσει τις κάθετες μαύρες γραμμές στη διαδρομή. Το όχημα ξεκινά από την είσοδο του χώρου, και ακολουθώντας τη γραμμή, σταματά διαδοχικά στις θέσεις που έχουμε ορίσει. Σε κάθε θέση αναπαράγει το αντίστοιχο αρχείο ήχου που έχουμε αποθηκεύσει στην κάρτα μνήμης. Για να ξεκινήσει από κάθε θέση και να προχωρήσει στην επόμενη, πρέπει να λάβει την κατάλληλη εντολή με το πάτημα ενός μπουτόν. Αν το όχημα ανιχνεύσει κάποιο εμπόδιο στη διαδρομή, τότε σταματά και περιμένει μέχρι να απομακρυνθεί το εμπόδιο.

Το όχημα θα μπορεί να χρησιμοποιηθεί βοηθητικά σε χώρους εκθέσεων, ιδιαίτερα για άτομα με προβλήματα όρασης.

Σχεδίαση του αυτόνομου οχήματος ηχητικής ξενάγησης

Για την υλοποίηση του οχήματος χρησιμοποιήθηκε μια ρομποτική πλατφόρμα με τέσσερις κινητήρες DC, οι οποίοι ελέγχονται από μια πλακέτα με το ολοκληρωμένο ελέγχου L298N. Η ανίχνευση της διαδρομής του οχήματος πραγματοποιείται με τη βοήθεια τριών αισθητήρων υπερύθρων και η ανίχνευση τυχόν εμποδίων στη διαδρομή με ένα αισθητήρα υπερήχων. Η αναπαραγωγή των αρχείων ήχου γίνεται από μια μονάδα MP3 Player στην οποία συνδέεται ένα ηχείο 3 Watt. Η μονάδα διαθέτει μια υποδοχή για κάρτα μνήμης SD, όπου είναι αποθηκευμένα τα αρχεία ήχου. Οι εντολές για την κίνηση του οχήματος δίνονται από ένα μπουτόν. Το όχημα τροφοδοτείται από μια συστοιχία 6 μπαταριών AA, η οποίες περιέχονται σε μια μπαταριοθήκη με διακόπτη. Για τον έλεγχο του οχήματος χρησιμοποιείται η πλακέτα Arduino Uno.

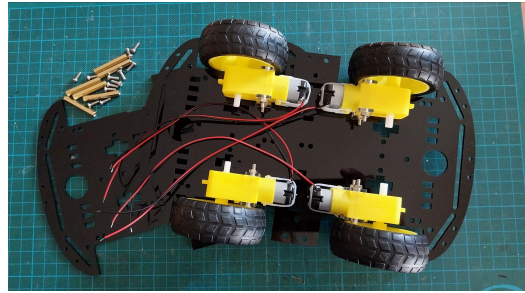
Οι συνδέσεις των στοιχείων του οχήματος φαίνονται στο παρακάτω σχήμα:



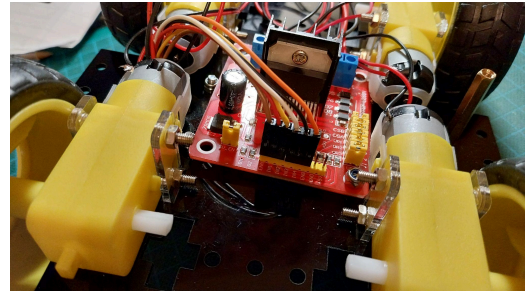
Το διάγραμμα των συνδέσεων δημιουργήθηκε με την δωρεάν online εφαρμογή δημιουργίας διαγραμμάτων ηλεκτρονικών κυκλωμάτων στη διεύθυνση www.circuit-diagram.org

Συναρμολόγηση του αυτόνομου οχήματος ηχητικής ξενάγησης

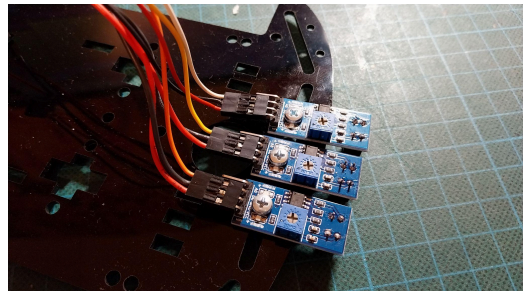
Η κατασκευή του οχήματος ξεκίνησε με την τοποθέτηση των κινητήρων στο ένα τμήμα από το σασί της πλατφόρμας.



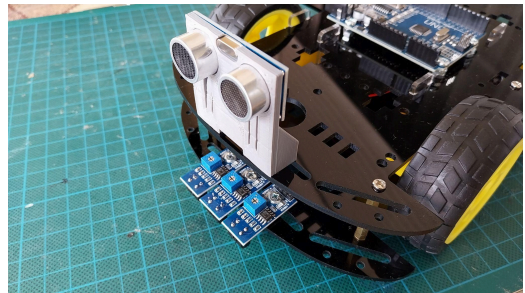
Ανάμεσα στους κινητήρες τοποθετήθηκε η πλακέτα με το ολοκληρωμένο L298N για την οδήγηση των κινητήρων. Στην πλακέτα συνδέθηκαν οι κινητήρες και η μπαταριοθήκη για την τροφοδοσία του οχήματος. Επίσης προετοιμάστηκαν οι καλωδιώσεις για τη σύνδεση με την πλακέτα του Arduino Uno.



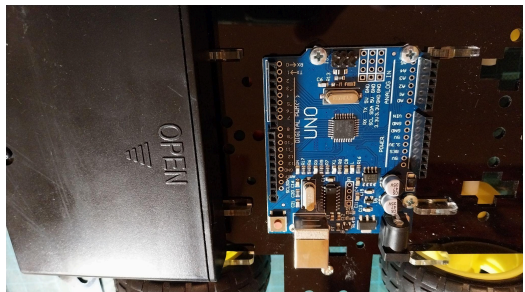
Στο δεύτερο τμήμα του σασί τοποθετήθηκαν οι αισθητήρες υπερύθρων με τις καλωδιώσεις τους.



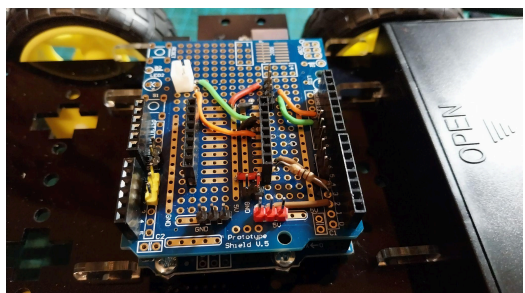
Τα δύο τμήματα του σασί συνδέθηκαν μεταξύ τους και στο πάνω μέρος του οχήματος τοποθετήθηκε και ο αισθητήρας υπερήχων με τη βάση του.



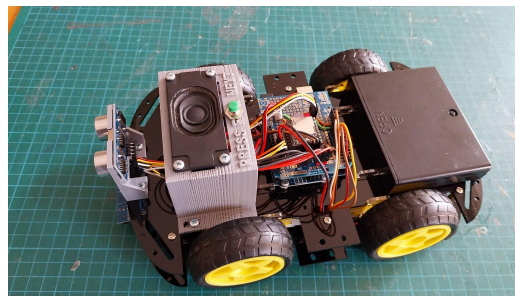
Στο πάνω τμήμα και στο κέντρο του οχήματος τοποθετήθηκε η πλακέτα του Arduino Uno, ενώ πίσω της τοποθετήθηκε η μπαταριοθήκη για την τροφοδοσία του οχήματος.



Στη συνέχεια ετοιμάστηκε η πρωτότυπη πλακέτα με τη βάση για το MP3 Player και τους ακροδέκτες σύνδεσης με τα υπόλοιπα στοιχεία του κυκλώματος και τοποθετήθηκε στο πάνω μέρος της πλακέτας του Arduino Uno.



Η συναρμολόγηση ολοκληρώθηκε με την τοποθέτηση στο μπροστά μέρος του οχήματος της βάσης με το ηχείου και τον διακόπτη εκκίνησης.



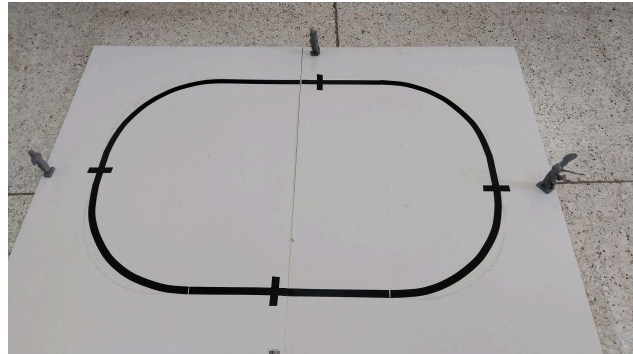
Τα υλικά που χρησιμοποιούμε στην κατασκευή, καθώς και το κόστος τους φαίνονται στον παρακάτω πίνακα:

Περιγραφή	Τεμάχια	Τιμή μονάδας	Κόστος
Arduino Uno SMD Compatible - CH340	1	9,90 €	9,90 €
Ρομποτική πλατφόρμα 4WD - 26 cm	1	16,90 €	16,90 €
Πλακέτα οδήγησης κινητήρων με το L298	1	5,90 €	5,90 €
Αισθητήρας υπερύθρων TCRT5000	3	1,60 €	4,80 €
Αισθητήρας υπερήχων HC-SR04	1	1,80 €	1,80 €
MP3 Player Mini Module	1	6,80 €	6,80 €
Κάρτα μνήμης microSDHC 32 GB	1	6,80 €	6,80 €
Ηχείο 3W 4Ohm	1	2,40 €	2,40 €
Μπουτόν για στήριξη σε πάνελ	1	0,50 €	0,50 €
Πλακέτα πρωτοτύπου για Arduino	1	1,80 €	1,80 €
Arduino Long Headers 6 pin	2	0,25 €	0,50 €
Arduino Long Headers 8 pin	2	0,25 €	0,50 €
Μπαταριοθήκη 6 x AA με διακόπτη	1	1,80 €	1,80 €
Μπαταρίες AA	6	0,40 €	2,40 €
Συνολικό κόστος			62,80 €

Στον παραπάνω πίνακα δεν περιλαμβάνεται το κόστος της καλωδίωσης και των ακροδεκτών για την πλακέτα πρωτοτύπων.

Κατασκευή μακέτας για την επίδειξη της λειτουργίας

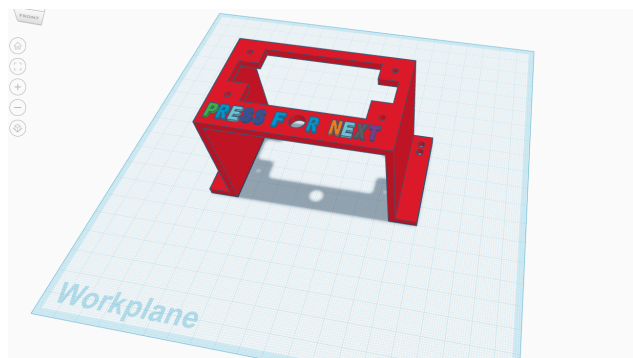
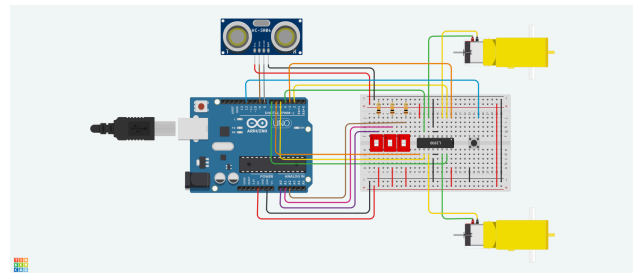
Για να ελέγξουμε τη λειτουργία του οχήματος κατασκευάσαμε μια διαδρομή σχήματος οβάλ πάνω σε χαρτί μακέτας με διαστάσεις 100x140 εκατοστά. Η διαδρομή σημειώθηκε με τη βοήθεια μαύρης μονωτικής ταινίας. Σε τέσσερα σημεία της διαδρομής τοποθετήθηκαν κάθετες γραμμές για τις στάσεις του οχήματος.



Για την αναπαράσταση των εκθεμάτων εκτυπώθηκαν τρία αντίγραφα αρχαίων ελληνικών αγαλμάτων. Από τα κείμενα με τις περιγραφές των αγαλμάτων δημιουργήθηκαν, με τη βοήθεια της πλατφόρμας parakeet.com, τα αρχεία mp3 που αναπαράγονται από τη μονάδα του MP3 Player.



Για την προσομοίωση της λειτουργίας και του προγραμματισμού των επιμέρους στοιχείων, χρησιμοποιήθηκε το ελεύθερο λογισμικό TinkerCAD. Το ίδιο πρόγραμμα χρησιμοποιήσαμε και για τη σχεδίαση της βάσης του ηχείου και του διακόπτη που κατασκευάσαμε με τον 3D printer.



Προγραμματισμός του αυτόνομου οχήματος ηχητικής ξενάγησης

Για τον προγραμματισμό του οχήματος χρησιμοποιήθηκε το λογισμικό Arduino IDE.

Στο πρώτο μέρος του προγράμματός εισάγουμε τις απαραίτητες βιβλιοθήκες, δημιουργούμε το αντικείμενο myDFPlayer για τον MP3 Player και ορίζουμε τη σειριακή μέσω software για την επικοινωνία του Arduino με το MP3 Player.

```
#include <DFRobotDFPlayerMini.h>
#include <SoftwareSerial.h>
SoftwareSerial softSerial(4, 3); //Rx,Tx
DFRobotDFPlayerMini myDFPlayer;
```

Στη συνέχεια ορίζουμε τους ακροδέκτες του Arduino που θα συνδεθούν με την πλακέτα οδήγησης των κινητήρων.

```
//Motor A
const int motorPin1 = 7;
const int motorPin2 = 6;
const int enA = 5;
//Motor B
const int motorPin3 = 9;
const int motorPin4 = 8;
const int enB = 10;
```

Κατόπιν ορίζουμε τους ακροδέκτες, όπου θα συνδεθεί ο αισθητήρας υπερήχων του οχήματος, καθώς και οι μεταβλητές duration, distance και cm τύπου long που θα χρησιμοποιηθούν για τον υπολογισμό της απόστασης από το πλησιέστερο εμπόδιο και θα μεταφέρουν την τιμή στο κυρίως πρόγραμμα.

```
//Ultrasonic Sensor
const int echoPin = 11;
const int trigPin = 12;
long duration, distance, cm;
```

Επίσης ορίζουμε τους ακροδέκτες, όπου θα συνδεθούν οι αισθητήρες υπερύθρων και οι μεταβλητές leftID, middleID και rightID για των αποθήκευση των τιμών τους.

```
//IR sensors
const int leftIR = A2;
const int middleIR = A1;
const int rightIR = A0;
int leftID, middleID, rightID;
```

Τέλος ορίζουμε τον ακροδέκτη σύνδεσης του διακόπτη εκκίνησης και τη μεταβλητή που αποθηκεύει την τιμή του διακόπτη. Επίσης ορίζουμε τη μεταβλητή moveOn για να

αποθηκεύουμε την τιμή 0 ή 1 για τον έλεγχο της κίνησης και τη μεταβλητή track για τον έλεγχο του αρχείου mp3 που θα αναπαράγεται από τη μονάδα του MP3 Player.

```
//Button
const int buttonPin = 2;
int buttonState = 1;
int moveOn = 1;
int track = 0;
```

Στην ενότητα setup, πρώτα καθορίζουμε τον τρόπο λειτουργίας των ακροδεκτών του Arduino με τις εντολές pinMode και την ταχύτητα λειτουργίας των κινητήρων με τις εντολές analogWrite.

```
void setup() {
    //Set pins as outputs
    pinMode(motorPin1, OUTPUT);
    pinMode(motorPin2, OUTPUT);
    pinMode(motorPin3, OUTPUT);
    pinMode(motorPin4, OUTPUT);
    pinMode(enA, OUTPUT);
    pinMode(enB, OUTPUT);
    analogWrite(enA,140);
    analogWrite(enB,140);
    //Set pins for ultrasonic sensor
    pinMode(echoPin, INPUT);
    pinMode(trigPin, OUTPUT);
    //Set pins as inputs for IR sensors
    pinMode(leftIR, INPUT);
    pinMode(middleIR, INPUT);
    pinMode(rightIR, INPUT);
    //Set pin as inputs for button
    pinMode(buttonPin, INPUT_PULLUP);
```

Στη συνέχεια ενεργοποιούμε τη σειριακή του Arduino για την επικοινωνία με τον υπολογιστή και την σειριακή μέσω software για την επικοινωνία με τη μονάδα του MP3 Player.

```
// initialize serial communication at 9600 bits per second:
Serial.begin(9600);
softSerial.begin(9600);
```

Στη συνέχεια ελέγχουμε τη σύνδεση και τη λειτουργία της μονάδας του MP3 Player. Αν η μονάδα δεν λειτουργεί σωστά, η εκτέλεση του προγράμματος σταματά και εμφανίζονται τα κατάλληλα μηνύματα στη σειριακή. Αν η μονάδα λειτουργεί σωστά, τότε, μετά την εμφάνιση των αντίστοιχων μηνυμάτων, ρυθμίζονται και οι παράμετροι λειτουργίας της.

```
Serial.println();
Serial.println(F("DFRobot DFPlayer Mini Demo"));
Serial.println(F("Initializing DFPlayer ... (May take 3~5 seconds)"));
```



```

if (!myDFPlayer.begin(softSerial, /*isACK = */true, /*doReset = */true)) {
//Use serial to communicate with mp3.
    Serial.println(F("Unable to begin:"));
    Serial.println(F("1.Please recheck the connection!"));
    Serial.println(F("2.Please insert the SD card!"));
    while(true);
}
Serial.println(F("DFPlayer Mini online."));
myDFPlayer.setTimeout(500); //Set serial communication time out 500ms
myDFPlayer.volume(30); //Set volume value (0~30)
myDFPlayer.EQ(DFPLAYER_EQ_NORMAL); //Set different EQ
myDFPlayer.outputDevice(DFPLAYER_DEVICE_SD); //Set device we use SD as
default
}

```

Στην ενότητα loop διαβάζουμε τη θέση του διακόπτη εκκίνησης και την απόσταση του οχήματος από το πλησιέστερο εμπόδιο. Όταν στο διακόπτη ανιχνεύσουμε χαμηλό δυναμικό, τότε εκτελείται η υπορουτίνα go() και το όχημα κινείται για λίγο ώστε να απομακρυνθεί από την κάθετη γραμμή της διαδρομής. Στη συνέχεια ελέγχει την απόσταση του πλησιέστερου εμποδίου και αν είναι μικρότερη από όσο έχουμε ορίσει το όχημα σταματά, αλλιώς καλεί την υπορουτίνα για την κίνηση του οχήματος. Όσο στον διακόπτη εκκίνησης ανιχνεύουμε υψηλό δυναμικό το όχημα παραμένει ακίνητο.

```

void loop() {
    buttonState = digitalRead(buttonPin);
    Serial.println(buttonState);
    cm = getDistance();
    Serial.print (cm);
    Serial.println(" cm");
    if (buttonState == LOW){
        moveOn = 0;
        delay(100);
        go();
    }
    if (moveOn == LOW) {
        if (cm <= 15){
            stop();
            Serial.println("stop at object");
        }
        else {
            move();
        }
    }
    else {
        Serial.println("Ready");
        delay(100);
    }
}
}

```

Με την υπορουτίνα move() ελέγχουμε την κίνηση του οχήματος. Με την έναρξη της υπορουτίνας διαβάζουμε τις τιμές των αισθητήρων υπερύθρων και ανάλογα με τις τιμές τους καλούμε τις υπορουτίνες για τις διακριτές κινήσεις της πλατφόρμας. Όταν οι αισθητήρες ανιχνεύσουν κάθετη γραμμή στην διαδρομή, τότε το όχημα σταματά, αλλάζει η κατάσταση κίνησης και αυξάνεται ο αριθμός του αρχείου mp3 που θα αναπαραχθεί. Τέλος καλείται η υπορουτίνα για την αναπαραγωγή του αρχείου mp3. Αν το όχημα βγει εκτός γραμμής τότε κινείται προς τα πίσω μέχρι να ανιχνεύσει ξανά τη γραμμή της διαδρομής στο δάπεδο.

```
void move(){
    leftID = digitalRead(leftIR);
    middleID = digitalRead(middleIR);
    rightID = digitalRead(rightIR);
    if (leftID==HIGH && middleID==HIGH && rightID==HIGH){
        stop();
        Serial.println("stop");
        moveOn = 1;
        track = track + 1;
        Serial.print("track=");
        Serial.println(track);
        playSong(); //Play mp3
        delay(1000);
    }
    else if (leftID==LOW && middleID==HIGH && rightID==LOW){
        forward();
        Serial.println("forward");
    }
    else if (leftID==HIGH && middleID==HIGH && rightID==LOW){
        turnLeft();
        Serial.println("turnLeft");
    }
    else if (leftID==LOW && middleID==HIGH && rightID==HIGH){
        turnRight();
        Serial.println("turnRight");
    }
    else if (leftID==HIGH && middleID==LOW && rightID==LOW){
        sharpTurnLeft();
        Serial.println("sharpTurnLeft");
    }
    else if (leftID==LOW && middleID==LOW && rightID==HIGH){
        sharpTurnRight();
        Serial.println("sharpTurnRight");
    }
    else{
        backward();
        Serial.println("backward");
    }
    Serial.print(leftID);
```

```

        Serial.print(middleID);
        Serial.println(rightID);
    }

```

Οι υπορουτίνες forward(), backward(), turnLeft(), turnRight(), sharpTurnLeft(), sharpTurnRight() και stop() εκτελούν τις στοιχειώδεις κινήσεις της πλατφόρμας, ελέγχοντας την φορά περιστροφής κάθε πλευράς κινητήρων. Οι υπορουτίνες αυτές καλούνται από την υπορουτίνα move() που ελέγχει την κίνηση ανάλογα με τις ενδείξεις των αισθητήρων υπερύθρων.

```

void forward(){
    // This code will move the platform forward for 10ms.
    digitalWrite(motorPin1, HIGH);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, HIGH);
    digitalWrite(motorPin4, LOW);
    delay(10);
}

```

```

void backward(){
    // This code will move the platform backward for 10ms.
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, HIGH);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, HIGH);
    delay(10);
}

```

```

void turnLeft(){
    //This code will turn the platform left for 10ms.
    digitalWrite(motorPin1, HIGH);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, LOW);
    delay(10);
}

```

```

void turnRight(){
    //This code will turn the platform right for 10ms.
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, HIGH);
    digitalWrite(motorPin4, LOW);
    delay(10);
}

```

```

void sharpTurnLeft(){
    //This code will turn the platform sharp left for 10ms.

```

```

        digitalWrite(motorPin1, HIGH);
        digitalWrite(motorPin2, LOW);
        digitalWrite(motorPin3, LOW);
        digitalWrite(motorPin4, HIGH);
        delay(10);
    }

    void sharpTurnRight(){
        //This code will turn the platform sharp right for 10ms.
        digitalWrite(motorPin1, LOW);
        digitalWrite(motorPin2, HIGH);
        digitalWrite(motorPin3, HIGH);
        digitalWrite(motorPin4, LOW);
        delay(10);
    }

    void stop(){
        //This code will stop the platform.
        digitalWrite(motorPin1, LOW);
        digitalWrite(motorPin2, LOW);
        digitalWrite(motorPin3, LOW);
        digitalWrite(motorPin4, LOW);
    }

```

Με την υπορουτίνα `getDistance()`, υπολογίζουμε την απόσταση του οχήματος από το πλησιέστερο εμπόδιο. Εισάγουμε την υπορουτίνα με τη λέξη `long` για να δηλώσουμε ότι θα επιστρέψει μια τιμή αυτού του τύπου. Η τιμή επιστρέφεται από την υπορουτίνα, μετά τον υπολογισμό της, με την εντολή `return`.

```

long getDistance(){
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance = (duration/2) / 29.1;
    return distance;
}

```

Με την υπορουτίνα `go()` ξεκινά το όχημα μετά από κάθε στάση. Το όχημα κινείται προς τα μπρος για 200 ms, ώστε να υπερπηδήσει τη κάθετη μαύρη γραμμή στη διαδρομή στην οποία σταμάτησε. Ο χρόνος αυτός εξαρτάται από το πάχος της κάθετης γραμμής που χρησιμοποιούμε.

```

void go(){
    digitalWrite(motorPin1, HIGH);
    digitalWrite(motorPin2, LOW);

```

```
    digitalWrite(motorPin3, HIGH);  
    digitalWrite(motorPin4, LOW);  
    delay(200);  
}
```

Με την υπορουτίνα playSong() καλούμε το αντικείμενο του MP3 Player και αναπαράγουμε το αρχείο mp3 που μας δείχνει η μεταβλητή track. Όταν η μεταβλητή μας πάρει την μέγιστη τιμή, ανάλογα με τον αριθμό των στάσεων και των αρχείων που αναπαράγουμε, τότε τη μηδενίζουμε για να επαναληφθεί η διαδικασία.

```
void playSong(){  
    myDFPlayer.play(track);  
    if (track == 4){  
        track = 0;  
    }  
}
```

Βιβλιογραφία

1. [Βασίλης Νούσης, “ARDUINO για αρχάριους”, 2019](#)
2. [Εμμανουήλ Πουλάκης, “Προγραμματίζοντας με τον μικροελεγκτή Arduino”, 2015](#)
3. [Γεώργιος Γίδας, Ευάγγελος Μαντζάνας, “Εφαρμογές Arduino στο σχολικό εργαστήριο για το περιβάλλον”, 2021](#)
4. [Νικόλαος Φανουράκης, “Εκπαιδευτική Ρομποτική με τον μικροελεγκτή Arduino”, 2019](#)

Ιστοσελίδες

1. <https://github.com/DFRobot/DFRobotDFPlayerMini>
2. https://wiki.dfrobot.com/DFPlayer_Mini_SKU_DFR0299