## Αυτοκινούμενο όχημα με αποφυγή εμποδίων

**Η λειτουργία**

Το όχημα κινείται με δύο ανεξάρτητους κινητήρες συνεχούς ρεύματος . Οι τροχοί είναι απευθείας συνδεμένοι στους κινητήρες. Η επιλογή της κατευθύνσεις που θα κινηθεί το όχημα, γίνεται δίνοντας διαφορετική ταχύτητα στους δύο κινητήρες. Όπως κινούνται τα τανκς.

Για την αναγνωρίσει των εμποδίων από το όχημα, χρησιμοποιείται ένας αισθητήρας υπερήχων μέτρησης απόστασης , ώστε όταν ανιχνεύσει μικρότερη απόσταση από αυτήν που θα του ορίσουμε, να αλλάξει συμπεριφορά.

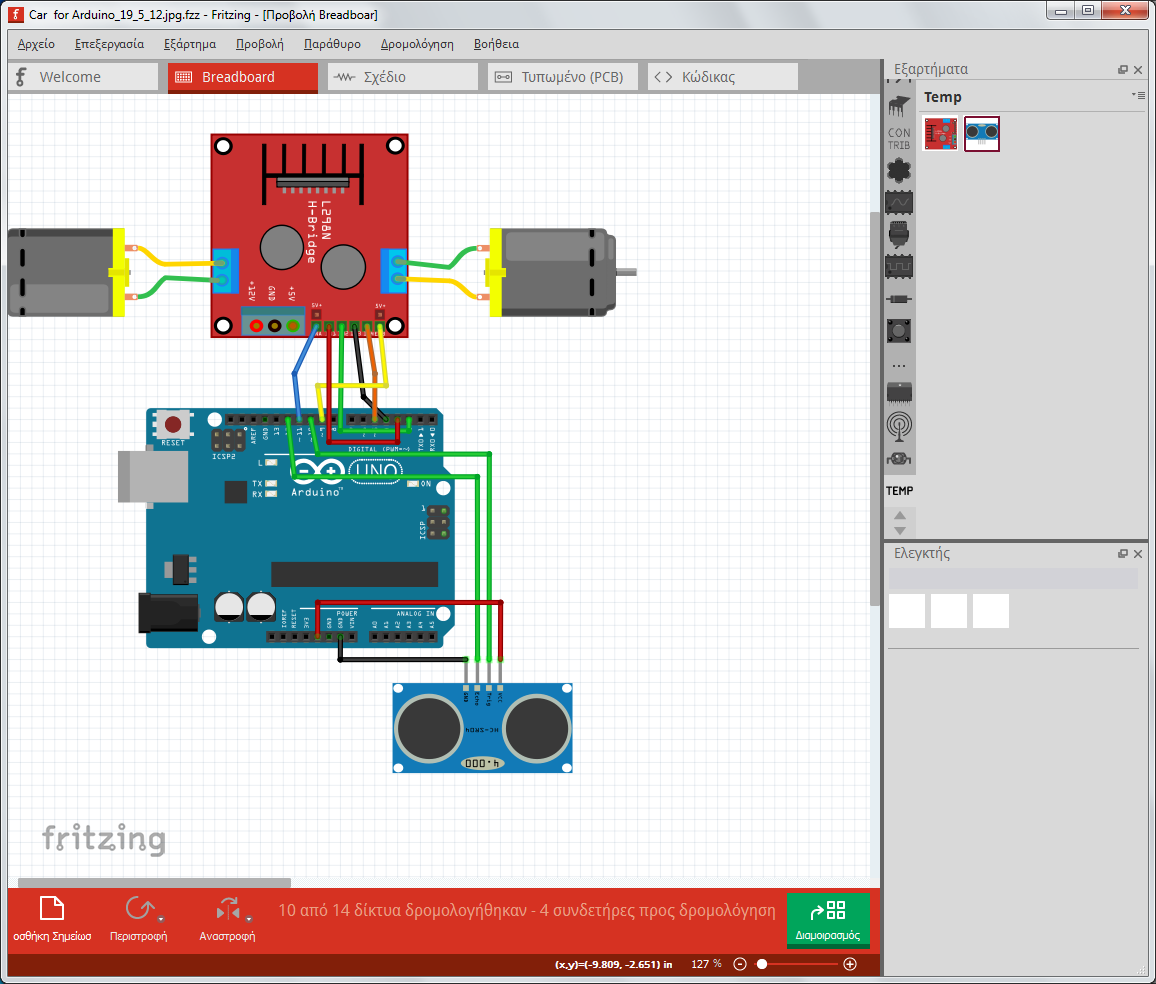
**Η κατασκευή**

Η κατασκευή ενός αυτοκινούμενου οχήματος στο χώρο που αποφεύγει τα εμπόδια θα χρειαστεί τα παρακάτω υλικά:

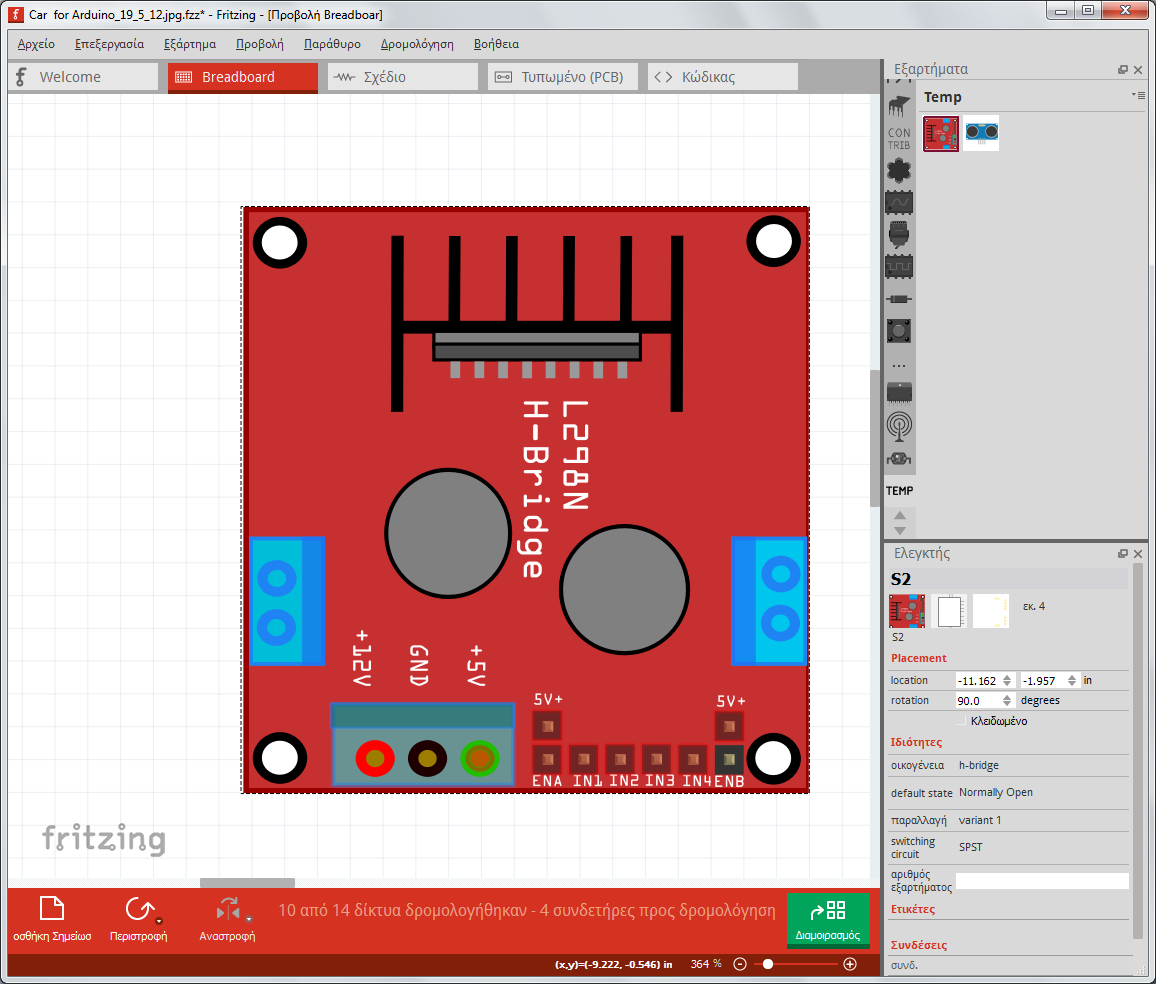
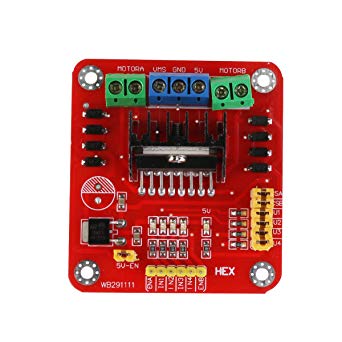
|  |  |
| --- | --- |
| **Περιγραφή** | |
| ARDUINO UNO ή συμβατό |  |
| L298N Dual H-Bridge |  |
| Αισθητήρας  ultrasonic-sonar-distance-sensor  HC-SR04 |  |
| 2WD Motor Robot Car Chassis  Kit for Arduino |  |
| 2 μπαταρίες 9 Volt |  |
| Connector 9 Volt μπαταρία |  |
| Διακόπτης ON-OFF |  |
| Μερικά καλώδια για τις συνδεσμολογίες |  |
| mini breadboard |  |

**Η συνδεσμολογία**

Η συνδεσμολογία των εξαρτημάτων δημιουργήθηκε με το πρόγραμμα σχεδίασης Fritzing. Αν και τα εξαρτήματά δεν έχουν ακριβώς την ίδια εικόνα, οι ονομασίες των ακροδεκτών είναι ίδιες και κάνουν την ίδια ακριβώς λειτουργία (Σχ 2).



**Σχ. 1** Συνδεσμολογία των εξαρτημάτων βάση του προγράμματος



**Σχ 2** Η λειτουργίατης Η-γέφυρας είναι ίδια όπου και να εμφανίζονται οι ακροδέκτες στην πλακέτα.

**Ο προγραμματισμός**

#define pin\_IN1 3

#define pin\_IN2 2

#define ENABLE\_A 11

#define pin\_IN3 4

#define pin\_IN4 5

#define ENABLE\_B 9

int trigPin = 10;

int echoPin = 12;

int duration;

int distance;

void setup() {

// put your setup code here, to run once:

pinMode(pin\_IN1, OUTPUT);

pinMode(pin\_IN2, OUTPUT);

pinMode(ENABLE\_A, OUTPUT);

pinMode(pin\_IN3, OUTPUT);

pinMode(pin\_IN4, OUTPUT);

pinMode(ENABLE\_B, OUTPUT);

pinMode(trigPin, HIGH);

pinMode (echoPin, INPUT);

}

void forward (int a, int b)

{

digitalWrite(pin\_IN1, LOW);

digitalWrite(pin\_IN2, HIGH);

analogWrite(ENABLE\_A, a);

digitalWrite(pin\_IN3, HIGH);

digitalWrite(pin\_IN4, LOW);

analogWrite(ENABLE\_B, b);

}

void backwards (int c, int d)

{

digitalWrite(pin\_IN1, HIGH);

digitalWrite(pin\_IN2, LOW);

analogWrite(ENABLE\_A, c);

digitalWrite(pin\_IN3, LOW);

digitalWrite(pin\_IN4, HIGH);

analogWrite(ENABLE\_B, d);

}

int get\_distance(int trigPin, int echoPin)

{

digitalWrite(trigPin, HIGH);

delayMicroseconds(2000);

forward (130, 200);

digitalWrite ( trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = (duration / 2) / 29.1;

}

void loop()

{

get\_distance(trigPin, echoPin);

if ( distance > 25)

forward (130, 255);

if (distance<25, distance>0)

backwards (20, 200);

}