

14/12/2017

Σταύρος Παπαντωνάκης, Νικοκλής Αργυρού, Κετσεμενίδης Ελευθέριος, Ahmed Abelsayed

# Arduino Remote Fire Truck

Από Την Ιδέα Στην Υλοποίηση



Ανώτατο Εκπαιδευτικό Ίδρυμα Πειραιά Τεχνολογικού Τομέα  
Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστικών Συστημάτων  
Μάθημα: Μηχατρονικά Συστήματα



## Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Παρόμοια Διανομή 4.0 Διεθνές (CC BY-NC-SA 4.0)

Μπορείτε να δείτε το πλήρες κείμενο της αδείας στην τοποθεσία:

<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.el>

2017-2018

Το Έργο αυτό διατίθεται υπό τους ορούς της Αδείας:



This is a human-readable summary of (and not a substitute for) the [license](#). [Αποτοίηση](#).

### Μπορείτε να:

- **Μοιραστείτε** — αντιγράψετε και αναδιανέμετε το υλικό με κάθε μέσο και τρόπο
- **Προσαρμόστε** — αναμείξτε, τροποποιήστε και δημιουργήστε πάνω στο υλικό

Ο αδειοδότης δεν μπορεί να ανακαλέσει αυτές τις ελευθερίες όσο εσείς ακολουθείτε τους όρους της άδειας.

---

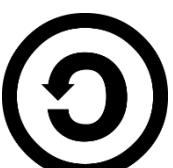
## Υπό τους ακόλουθους όρους:



**Αναφορά Δημιουργού** — Θα πρέπει να καταχωρίσετε [αναφορά στο δημιουργό](#), με σύνδεσμο της άδειας, και [με αναφορά αν έχουν γίνει αλλαγές](#). Μπορείτε να το κάνετε αυτό με οποιονδήποτε εύλογο τρόπο, αλλά όχι με τρόπο που να υπονοεί ότι ο δημιουργός αποδέχεται το έργο σας ή τη χρήση που εσείς κάνετε.



**Μή Εμπορική Χρήση** — Δε μπορείτε να χρησιμοποιήσετε το υλικό για [εμπορικούς σκοπούς](#).



**Παρόμοια Διανομή** — Αν αναμείξετε, τροποποιήσετε, ή δημιουργήσετε πάνω στο υλικό, πρέπει να διανείμετε τις δικές σας συνεισφορές υπό την [ίδια άδεια](#) όπως και το πρωτότυπο.

**Δεν υπάρχουν πρόσθετοι περιορισμοί** — Δε μπορείτε να εφαρμόσετε νομικούς όρους ή [τεχνολογικά μέτρα](#) που να περιορίζουν νομικά τους άλλους από το να κάνουν οτιδήποτε επιτρέπει η άδεια.

---

## Σημειώσεις:

Δεν είστε υποχρεωμένοι να συμμορφωθείτε με τη άδεια για στοιχεία του υλικού που είναι σε Κοινό Κτήμ/ public domain, ή εκεί όπου η χρήση επιτρέπεται στα πλαίσια μιας ισχύουσας εξαίρεσης ή περιορισμού.

Δεν παρέχεται καμία εγγύηση. Η άδεια μπορεί να μην σας δίνει όλα τα απαραίτητα δικαιώματα για τη χρήση που σκοπεύετε. Για παράδειγμα, άλλα δικαιώματα, όπως διαφήμιση, ιδιωτικότητα, ή ηθικά δικαιώματα μπορεί να περιορίσουν το πώς χρησιμοποιείτε το υλικό.

---

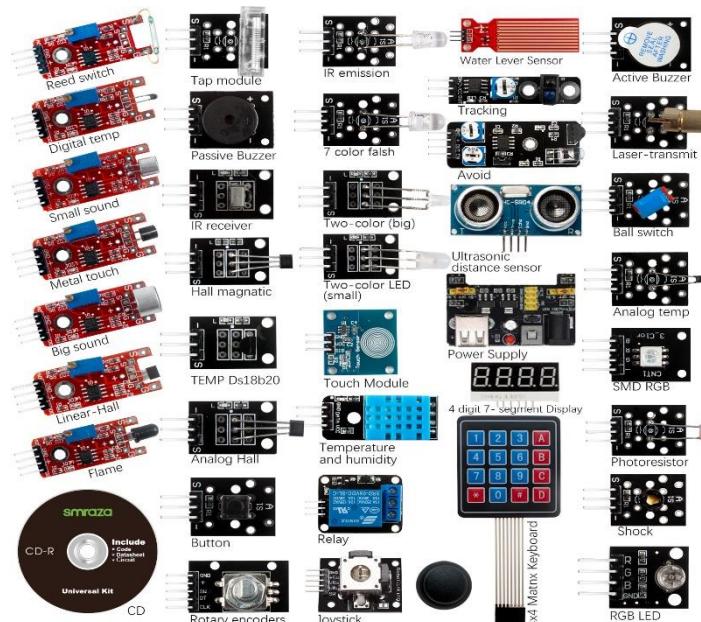
(Κενή Σελίδα).

# Περιεχόμενα

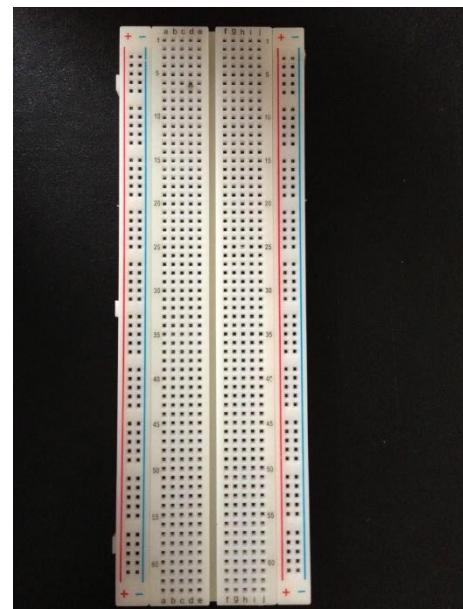
Εισαγωγή .....	6
Η Ιδέα .....	8
Σχεδιασμός .....	9
Λίστα Υλικών.....	9
Εργαλεία.....	11
To Hardware .....	11
Flame Sensor .....	12
KY-028 .....	13
To L293D.....	14
To L298N.....	16
To Software .....	18
Android Code.....	19
Android GUI.....	19
Android Functions .....	19
Arduino Code.....	21
Bluetooth Controller Test .....	21
Flame Sensor Test.....	24
Thermistor KY-028 Sensor Test.....	25
Main code.....	27
Συναρμολόγηση.....	41
Προβλήματα.....	47
Χωρίς Συριακή! .....	47
Πάρτο Αλλιώς .....	48

# Εισαγωγή

Οι αυτοματισμοί και τα συστήματα αυτόματου ελέγχου πριν από μερικά χρόνια χρησιμοποιούνταν κυρίως στην βιομηχανία και στα συστήματα μαζικής παραγωγής, η χρήση τους από τον μέσο άνθρωπο ήταν σχεδόν αδύνατη λόγω της απαγορευτικής τους τιμής. Τον τελευταίο καιρό αυτό έχει αλλάξει λόγω του ξεσπάσματος των microcontrollers στην αγορά και των έτοιμων kit που περιέχουν αισθητήρες θερμοκρασίας, υγρασίας, κεραίες Bluetooth και Wi-Fi, LCD οθόνες και πολλά ακόμα ηλεκτρονικά εξαρτήματα κατασκευασμένα έτσι ώστε να μπορούν να τοποθετηθούν εύκολα σε ένα Breadboard.



Εικόνα 2. Basic Sensor Kit For Arduino



Εικόνα 1. Breadboard

Πλέον με λίγα χρήματα μπορείς να αγοράσεις έναν μικροεπεξεργαστή και μερικούς αισθητήρες και να δημιουργήσεις θαυμάσια πράγματα και όταν λέμε λίγα χρήματα εννοούμε της τάξεως των 10 Ευρώ. Βεβαία το κόστος εξαρτάται από την πολυπλοκότητα και την δυσκολία της υλοποίησης της εφαρμογής μας. Με λίγα λόγια είναι προφανές ότι το

κόστος της τροποποίησης ενός φωτιστικού έτσι ώστε να ανάβει την νύχτα και να σβήνει μόνο του την ημέρα είναι πολύ χαμηλότερο από την κατασκευή ενός τηλεκατευθυνόμενου οχήματος. Ο πιο δημοφιλής μικροεπεξεργαστής και ένας από τους πρώτους που προώθησαν αυτή τη «μόδα» είναι το Arduino. Στην πραγματικότητα το Arduino είναι μια μητρική πλακέτα η οποία περιέχει έναν μικροεπεξεργαστή της εταιρίας Atmel, έναν κρυσταλλικό ταλαντωτή, μερικούς πυκνωτές και αντίστασης, μερικούς σταθεροποιητές τάσης, έναν USB To Serial Converter, και μερικά Socket στα οποία μπορείς να συνδέσεις καλώδια και να τα εκμεταλλευτείς ως εισόδους και εξόδους του μικροεπεξεργαστή.

Το Arduino βγαίνει σε διάφορες εκδόσεις, μερικές αυτών είναι το Arduino Nano, Arduino Micro και Arduino Lilypad τα οποία είναι χαρακτηριστικά για το μικρό τους μέγεθος. Στην συνέχεια έρχεται το πιο δημοφιλές, το Arduino UNO καθώς και το Arduino Mega, Arduino Leonardo και μερικά ακόμα . Οι βασικές διάφορες ανάμεσα σε όλα αυτά τα διαφορετικά μοντέλα είναι το μέγεθος, η μνήμη, η ποσότητα των Input και Output υποδοχών καθώς και η τιμή. Ανάλογα με την πολυπλοκότητα του project που θέλουμε να υλοποιήσουμε καλούμαστε να επιλέξουμε και τον κατάλληλο μικροεπεξεργαστή. Στις περισσότερες περιπτώσεις το Arduino UNO είναι επαρκές εκτός από τις περιπτώσεις που χρειάζονται πολλά Input και Output Signals ή πολύ μικρό μέγεθος. Το Arduino UNO κοστίζει γύρω στα 25 ευρώ. Βέβαια εφόσον είναι open source και open hardware μπορούμε ακόμα και εμείς να φτιάξουμε το δικό μας Arduino UNO αγοράζοντας έναν μικροεπεξεργαστή, συγκεκριμένα τον Atmel ATmega328 ή ακόμα και να αγοράσουμε μια αντιγραφή «γκούχου...» του Arduino UNO από το eBay «γκούχου...» με 5 Ευρώ το οπόιο εμείς ουδέποτε δεν σας το συνιστούμε!! «γκούχου...». Το Arduino προγραμματίζεται με πολύ απλές εντολές όμοιες με αυτές της C/C++.



Εικόνα 3. Arduino Collection

## Η Ιδέα

Στο πλαίσιο του εργαστηρίου του μαθήματος Μηχατρονικά Συστήματα μας ζητήθηκε να επιλέξουμε μια ανάμεσα σε μερικές εργασίες και να την υλοποιήσουμε. Επιλέξαμε αυτή που μας φάνηκε πιο ενδιαφέρον από τις υπόλοιπες καθώς πιθανόν και την πιο περίπλοκη. Ο τίτλος της εργασίας είναι «**Μελέτη και κατασκευή αυτόνομου ρομποτικού οχήματος που να ανιχνεύει εστία φωτιάς μέσα σε κλειστό χώρο, με χρήση Arduino**». Ο έλεγχος του οχήματος θα γίνεται μέσω Android συσκευής με χρήση Bluetooth καθώς και ο εντοπισμός της εστίας φωτιάς θα επιτυγχάνεται από διπλό έλεγχο καθώς θα παρέχει και μια ενδεικτική αντλία έτσι ώστε να μπορεί να ψεκάσει νερό. Ένα από τα

πρώτα πράγματα που κάναμε ήταν να δώσουμε ένα όνομα στο project μας, το οποίο είναι Arduino FireCar.

Έτσι μετά από αρκετή σκέψη για τα τεχνικά μέρη του οχήματος καθώς και το πώς μπορούν να υλοποιηθούν καταλήξαμε και αποφασίσαμε να προχωρήσουμε στον σχεδιασμό του.

## Σχεδιασμός

Σκεφτήκαμε πως το όχημα μας για να εντοπίζει φωτιές ενώρα κίνησης θα ήταν πρακτικό να έχει αισθητήρες περιμετρικά του έτσι ώστε η πιθανότητα εντοπισμού μιας φλόγας να είναι μεγαλύτερη καθώς και να αυξηθεί ο χώρος παρατήρησής του, σε περίπτωση εντοπισμού φωτιάς θα ενεργοποιείται μια σειρήνα και τα δεδομένα των αισθητήρων θα αποστέλλονται στο Android σύστημα, όπου θα τυπώνονται στην οθόνη του ξεχωριστά μηνύματα για κάθε αισθητήρα και θερμοκρασία. Επίσης, στην οθόνη του Android συστήματος θα υπάρχουν τέσσερα(4) κουμπιά για την κίνηση του οχήματος (μπροστά, πίσω, αριστερά, δεξιά) καθώς και έξι(6) κουμπιά για την κίνηση της μάνικας ψεκασμού (δεξιά, αριστερά, πάνω, κάτω, μέσα, έξω).

## Λίστα Υλικών

Από την εκφώνηση της εργασίας καθώς και από τα παραπάνω, μας προέκυψε μια λίστα εξαρτημάτων που χρειαζόμασταν για να υλοποιήσουμε το project.

- 1 – Flame Sensor 5 Channels το οποίο θα τοποθετήσουμε στο μπροστινό μέρος του οχήματος.
- 2 – 3x Flame Sensor 1 Chanel οι οποίοι θα τοποθετηθούν αριστερά, δεξιά και πίσω.
- 3 – Thermistor KY-028 NTC για τον έλεγχο της θερμοκρασίας.
- 4 – Αντλία νερού 5v, εκκίνηση 2A.
- 5 – 5x Dc motor για κίνηση μπρος πίσω, αριστερά δεξιά. Κίνηση του βραχίονα πάνω, κάτω, αριστερά, δεξιά, μέσα, έξω.
- 6 – Bluetooth module HC-05 για την επικοινωνία με το Android.
- 7 – Breadboard.
- 8 – Arduino Mega καθώς με τόσους αισθητήρες και κινητήρες ένα UNO δεν μας έφτανε.
- 9 – 3xLM293 Για τον έλεγχο των κινητήρων.
- 10 – LM298 Για τον έλεγχο της αντλίας.
- 11 – Μπαταρίες Li-on 18650.
- 12 – Πολλά καλώδια!!!!(Πάρα Πολλά Καλώδια!).
- 13 – Πολλή Θερμοσυλικόνη.
- 14 – Πολλά Πλαστικά Δεματικά.
- 15 – Καλάι.
- 16 – Μονωτική Ταινία.
- 17 – Μπαταρίες 18650 3.7V 1A

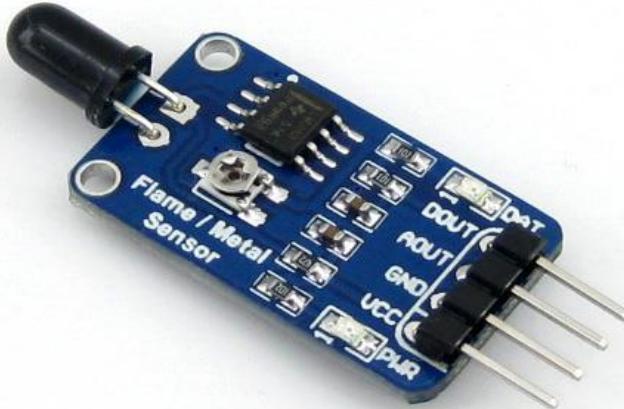
# Εργαλεία

- 1 – Κατσαβίδια.
- 2 – Πολύμερο.
- 3 – Κοπίδι.
- 4 – Κόπτης Καλωδίων.
- 5 – Πολυεργαλείο τύπου “Dremel”.
- 6 – Πιστόλι Θερμοσυλικόνης.
- 7 – Κολλητήρι.

## To Hardware

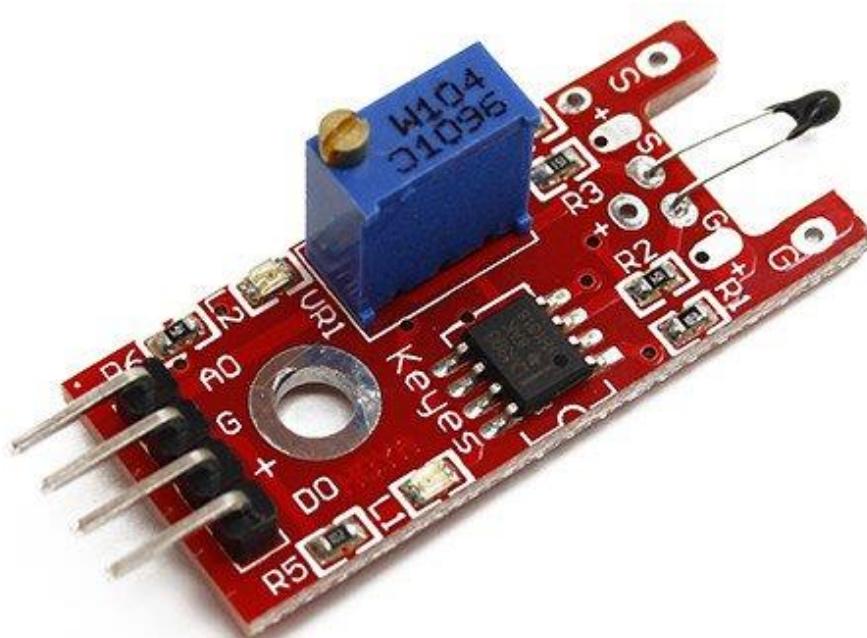
Το πρώτο πράγμα που κάναμε ήταν να βρούμε πώς δουλεύουν οι αισθητήρες και τα υλικά που είχαμε πάρει. Μετά από αναζήτηση και διάβασμα βρήκαμε τα εξής.

## Flame Sensor



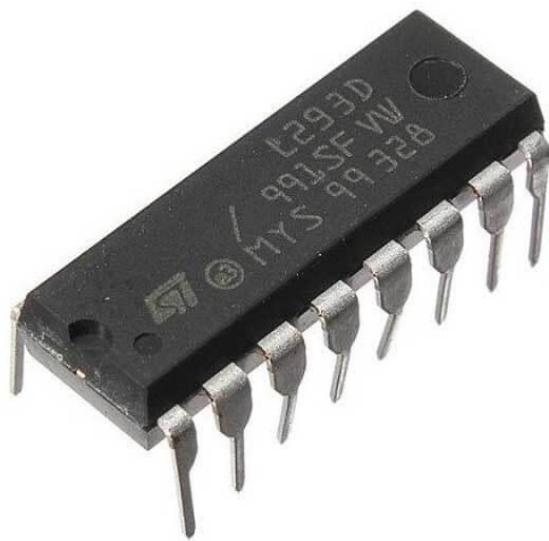
Ο αισθητήρας φωτιάς αποτελείται από ένα IR LED, έναν ενισχυτή, ένα ποτενσιόμετρο και έχει 4 pin Vcc, Gnd, Aout, Dout. Το IR LED παράγει ένα αναλογικό σήμα ανάλογα της ακτινοβολίας που λαμβάνει, το σήμα αυτό περνάει από τον ενισχυτή και ενισχύεται. Στην Αναλογική Έξοδο (Aout) παίρνουμε μια τιμή από 0 έως 5V. ~5V παίρνουμε όταν δεν υπάρχει φωτιά και ~0V παίρνουμε όταν υπάρχει. Χρησιμοποιούμε το σύμβολο “~”(Περίπου ίσο) λόγω του ότι ο αισθητήρας επηρεάζεται ακόμα και από την ακτινοβολία του περιβάλλοντος και του φωτός. Ενώ αντίθετα η έξοδος Dout παίρνει τις τιμές 0V ή 5V ανάλογα με την ακτινοβολία. Η τιμή τροφοδοσίας του Vcc μπορεί να είναι από 3.3V έως 5V.

## KY-028



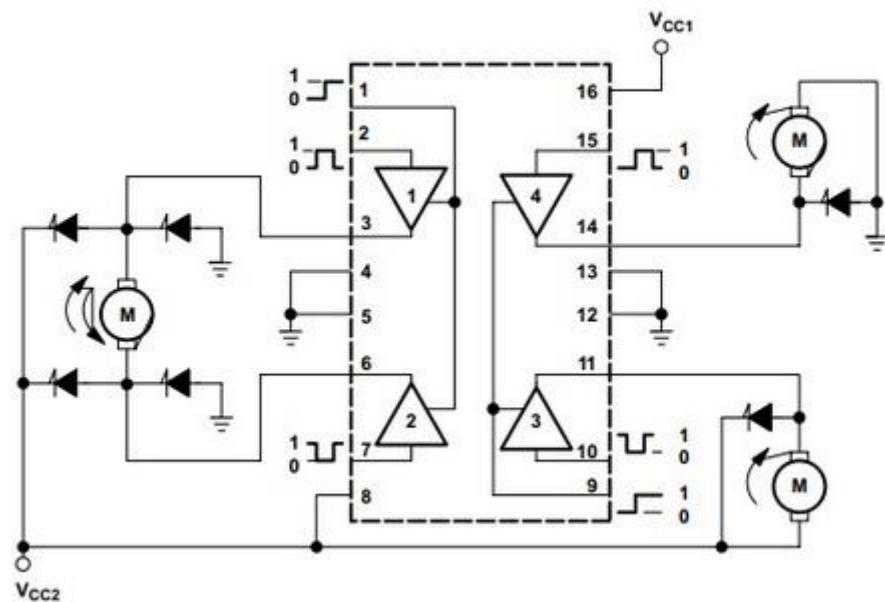
Αντίστοιχα με την ίδια αρχή λειτουργίας δουλεύει και το Thermistor KY-028. Αποτελείται από ένα thermistor, έναν ενισχυτή και ένα ποτενσιόμετρο και έχει 4 pin Vcc, Gnd, Aout, Dout. Το thermistor παράγει ένα αναλογικό σήμα ανάλογο της θερμοκρασίας, το οποίο ενισχύεται από τον ενισχυτή και φτάνει στην έξοδο Dout με μια τιμή 0V ή 5V ανάλογα με το αν έχει περασει το κατώφλι ή όχι. Το κατώφλι μπορεί να ρυθμιστεί από το ποτενσιόμετρο.

## To L293D

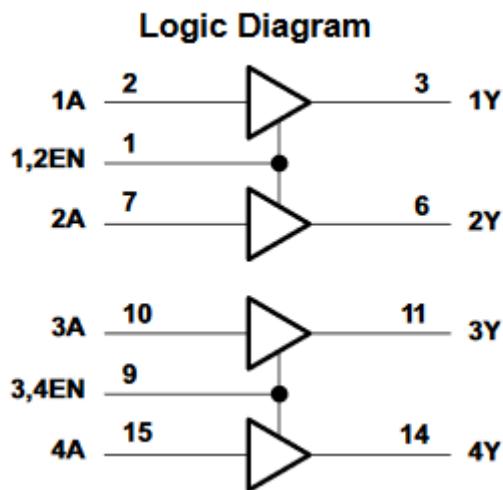


Το L293D είναι ένα ολοκληρωμένο το οποίο αποτελείται από διάφορα ηλεκτρονικά στοιχεία, διόδους κλπ.

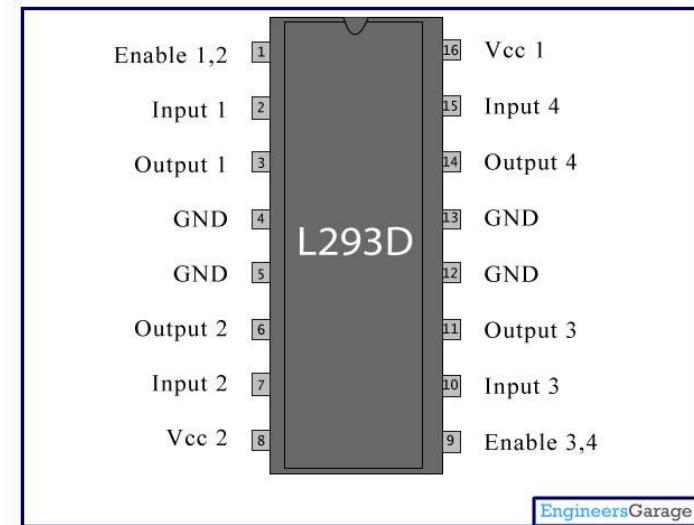
## L293D Functional Block Diagram



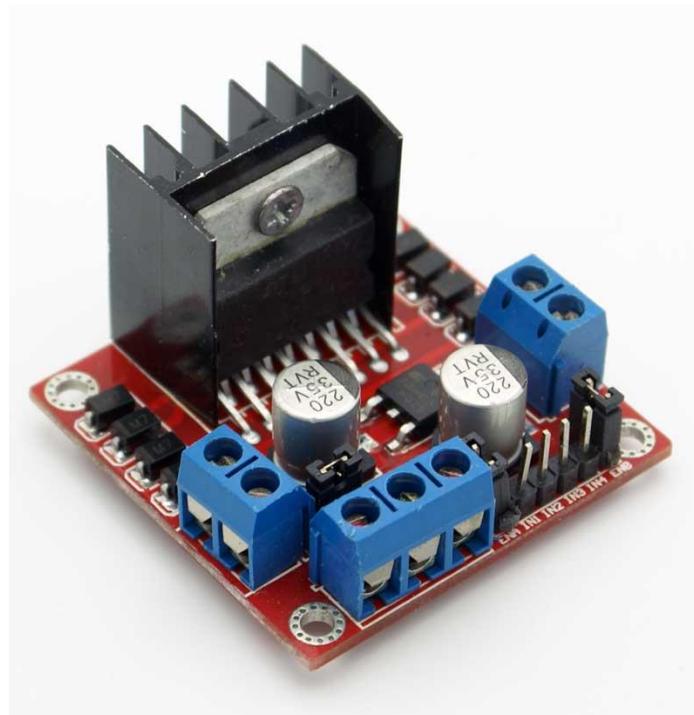
Το βασικό σύστημα όμως είναι ένα δικτύωμα ελεγχόμενων Buffer όπου όταν δώσουμε το σήμα Enable ενεργοποιούμε το κύκλωμα που έχουμε συνδέσει.



To L293D το χρησιμοποιούμε λόγω της αρχής λειτουργίας του για να οδηγήσουμε DC και stepper Motors με τάση 4.5 – 36V και ρεύμα έως 1A. έχει 16 pin και με ένα ολοκληρωμένο μπορούμε να λειτουργήσουμε πλήρως 2 dc motors.

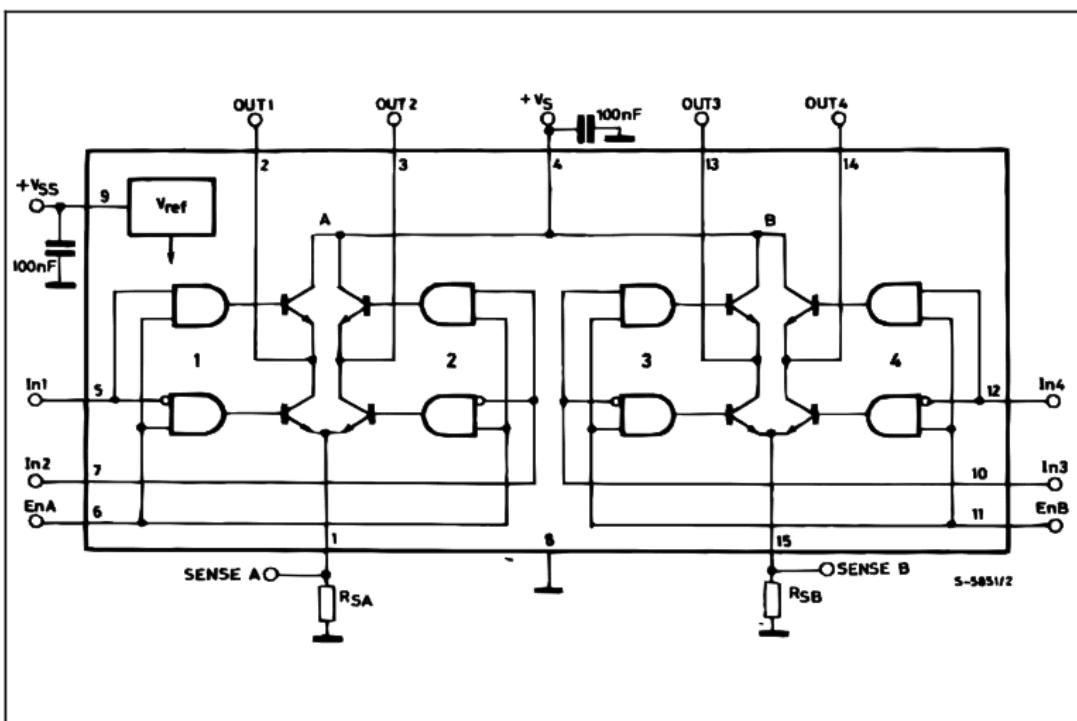


## To L298N

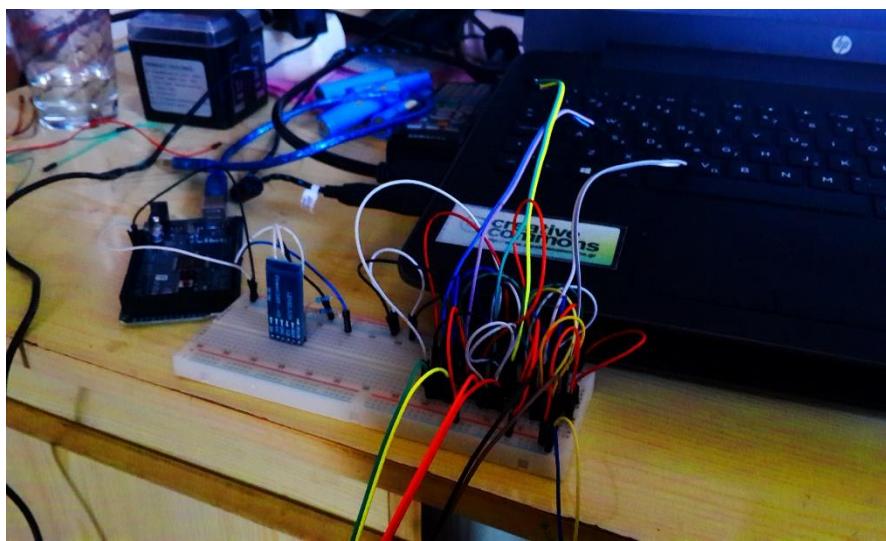


Τέλος το L298N είναι αρκετά πιο σύνθετο αλλά η αρχή λειτουργίας του είναι σχεδόν ίδια με αυτή του L293D. Αντίθετα με το προηγούμενο όμως το συγκεκριμένο μπορεί να αντέξει έως και 46V τάση και 4A ρεύμα.

## BLOCK DIAGRAM

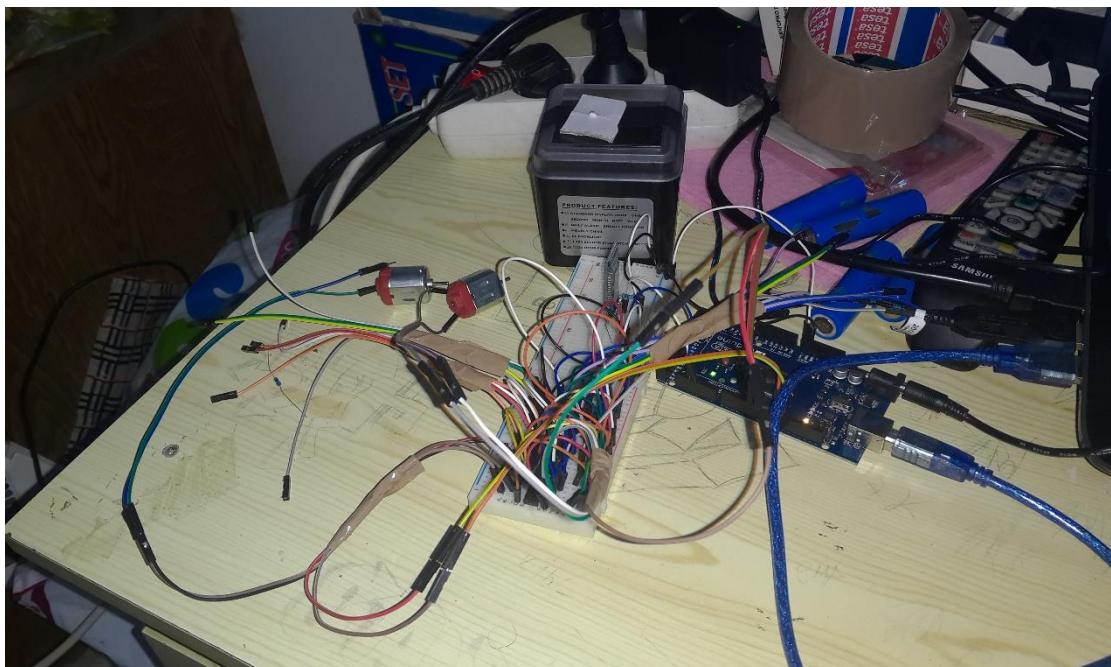


Αρχίσαμε και δοκιμάζαμε το Hardware ξεχωριστά για να δούμε αν δουλεύει σωστά. Γράψαμε μερικά μικρά προγραμματικά και βάλαμε μπροστά τα μοτέρ, μετρήσαμε φωτιά με τους αισθητήρες και αρχίσαμε να σκεπτόμαστε πώς θα το υλοποιήσουμε. Έτσι αρχίσαμε να γράφουμε το βασικό πρόγραμμα του project.



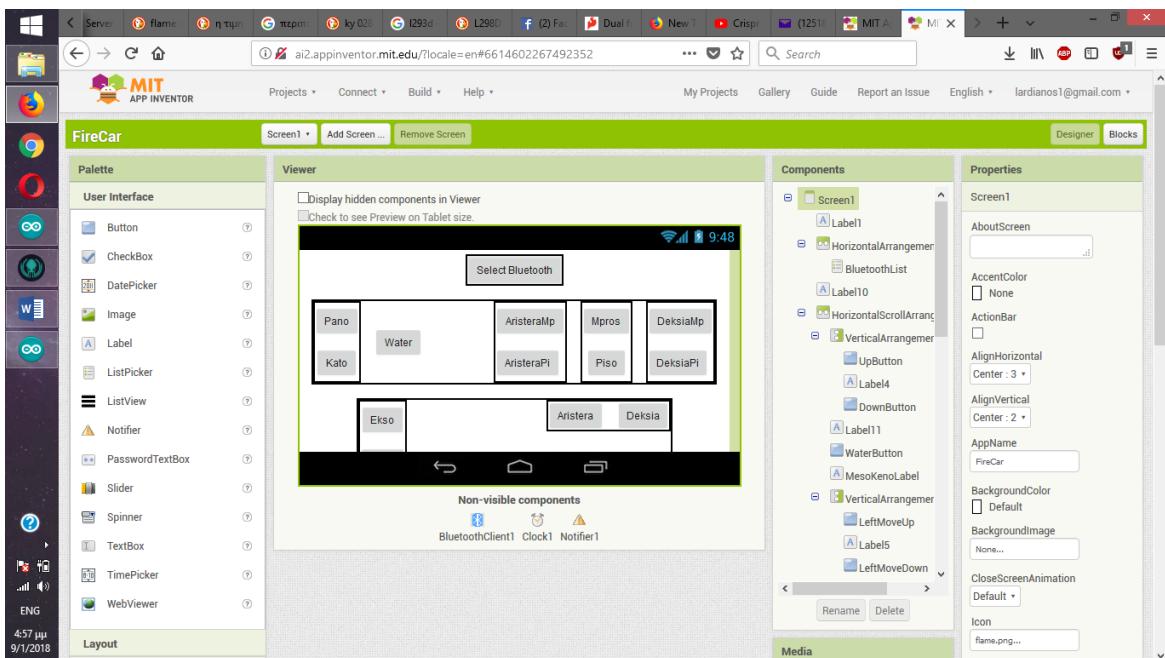
# To Software

To Software το χωρίσαμε σε δυο βασικά κομμάτια: το μέρος του Android Smartphone και το μέρος του Arduino. Το μέρος του android το αναπτύξαμε στο σύστημα MIT App Inventor ενώ το Arduino στο Arduino IDE.

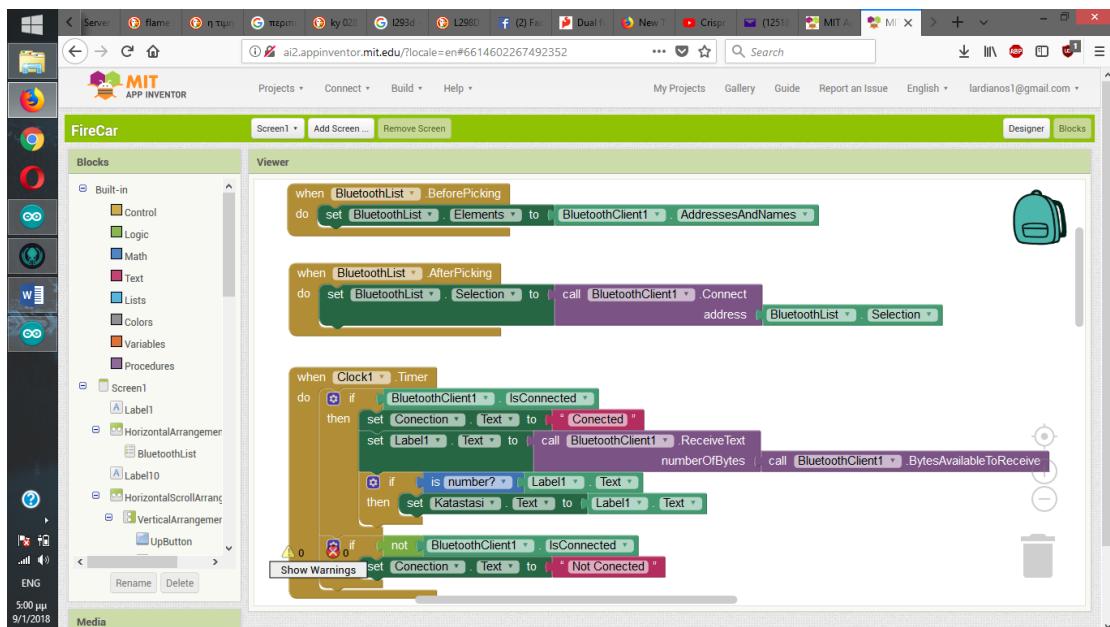


# Android Code

## Android GUI



## Android Functions

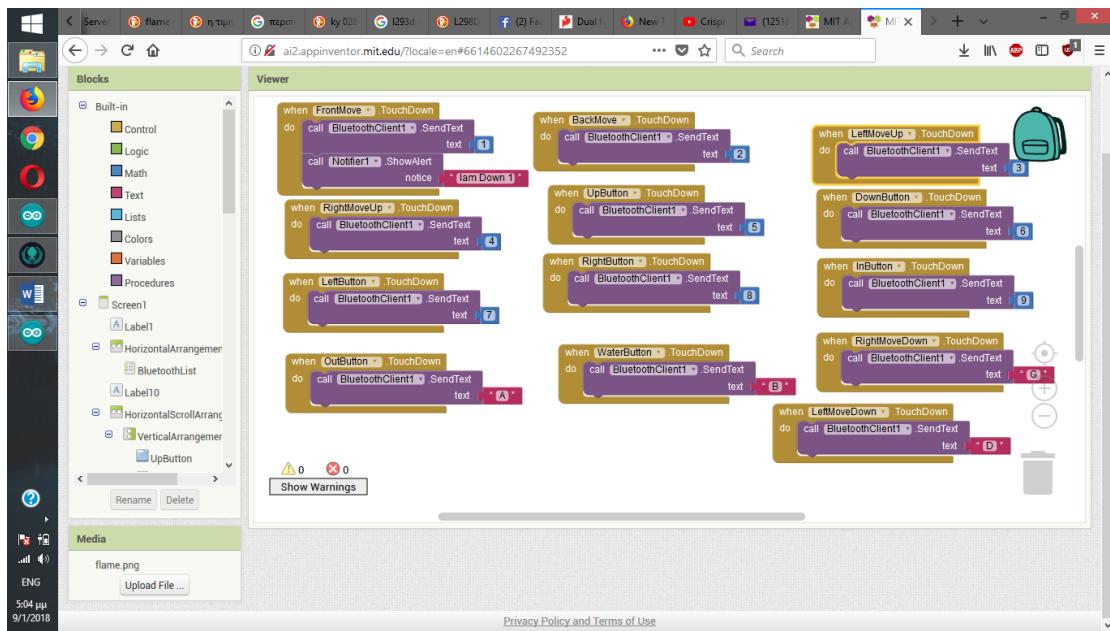


Σε αυτό το κομμάτι γίνεται η σύνδεση με το Arduino μέσω του Bluetooth οπού αν είναι επιτυχής μας εμφανίζει το μήνυμα connected

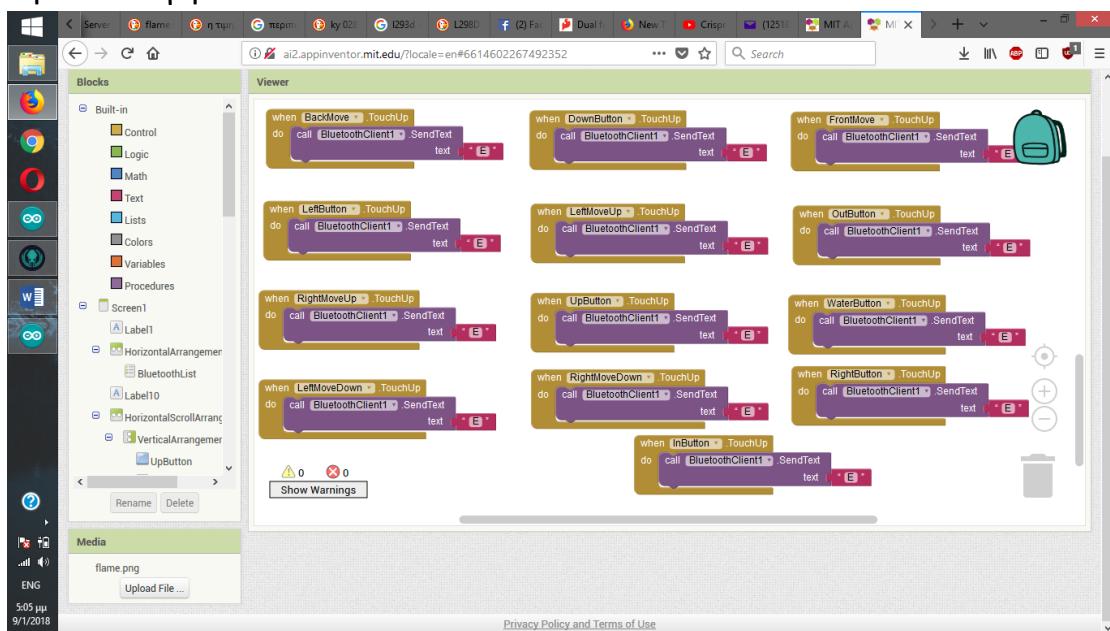
19 | Page

ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ

Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστικών Συστημάτων  
Μηχατρονικά Συστήματα



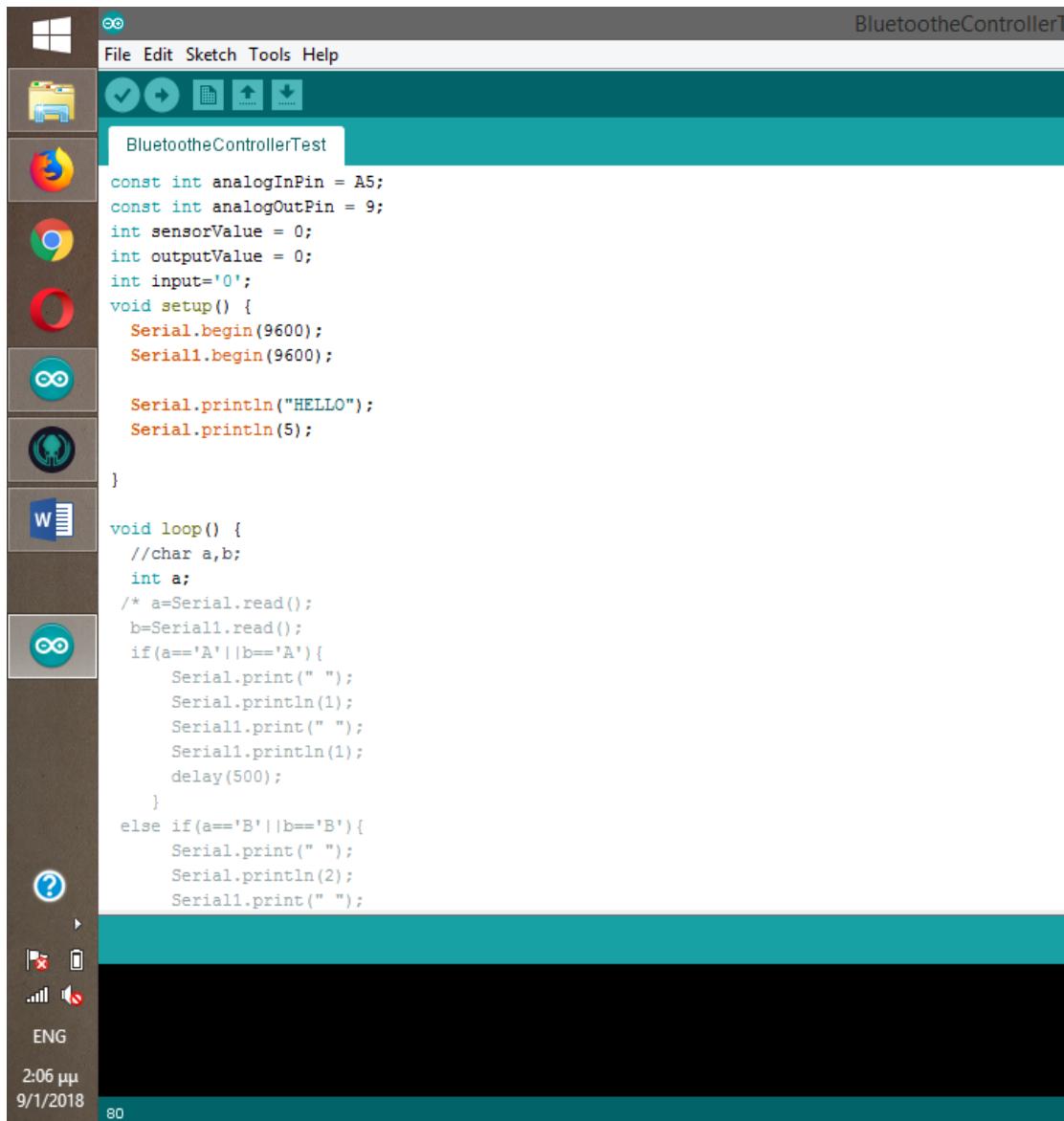
Όταν πατηθεί κάποιο από τα διαθέσιμα πλήκτρα αποστέλλεται ένας κωδικός αριθμός οπού μετέπειτα ελέγχεται από το Arduino και κάνει την ανάλογη δουλειά .



Όταν αφήσουμε το πλήκτρο στο Arduino στέλνεται αυτόματα ο κωδικός “Ε” οπού όταν τον διαβάζει το Arduino τερματίζει όλους τους κινητήρες.

# Arduino Code

## Bluetooth Controller Test



The screenshot shows the Arduino IDE interface. The title bar reads "BluetoothControllerTest". The menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar has icons for Open, Save, Upload, and Download. The left sidebar contains icons for File, Recent, Examples, Libraries, and a question mark. The code editor window displays the following sketch:

```
const int analogInPin = A5;
const int analogOutPin = 9;
int sensorValue = 0;
int outputValue = 0;
int input='0';
void setup() {
    Serial.begin(9600);
    Serial1.begin(9600);

    Serial.println("HELLO");
    Serial.println(5);

}

void loop() {
    //char a,b;
    int a;
    /* a=Serial.read();
    b=Serial1.read();
    if(a=='A'||b=='A') {
        Serial.print(" ");
        Serial.println(1);
        Serial1.print(" ");
        Serial1.println(1);
        delay(500);
    }
    else if(a=='B'||b=='B') {
        Serial.print(" ");
        Serial.println(2);
        Serial1.print(" ");
    }
}

```

The status bar at the bottom shows the language as "ENG", the date and time as "2:06 μμ 9/1/2018", and the page number as "80".

Η επικοινωνία μέσω του Bluetooth επιτεύχθηκε μέσω των συριακών θυρών του Arduino. Τα δεδομένα που έρχονται από το Android περνάνε στην συριακή του Arduino και με ένα Serial.read() και αντίθετα η αποστολή από το Arduino στο Android γίνεται με ένα Serial.write()

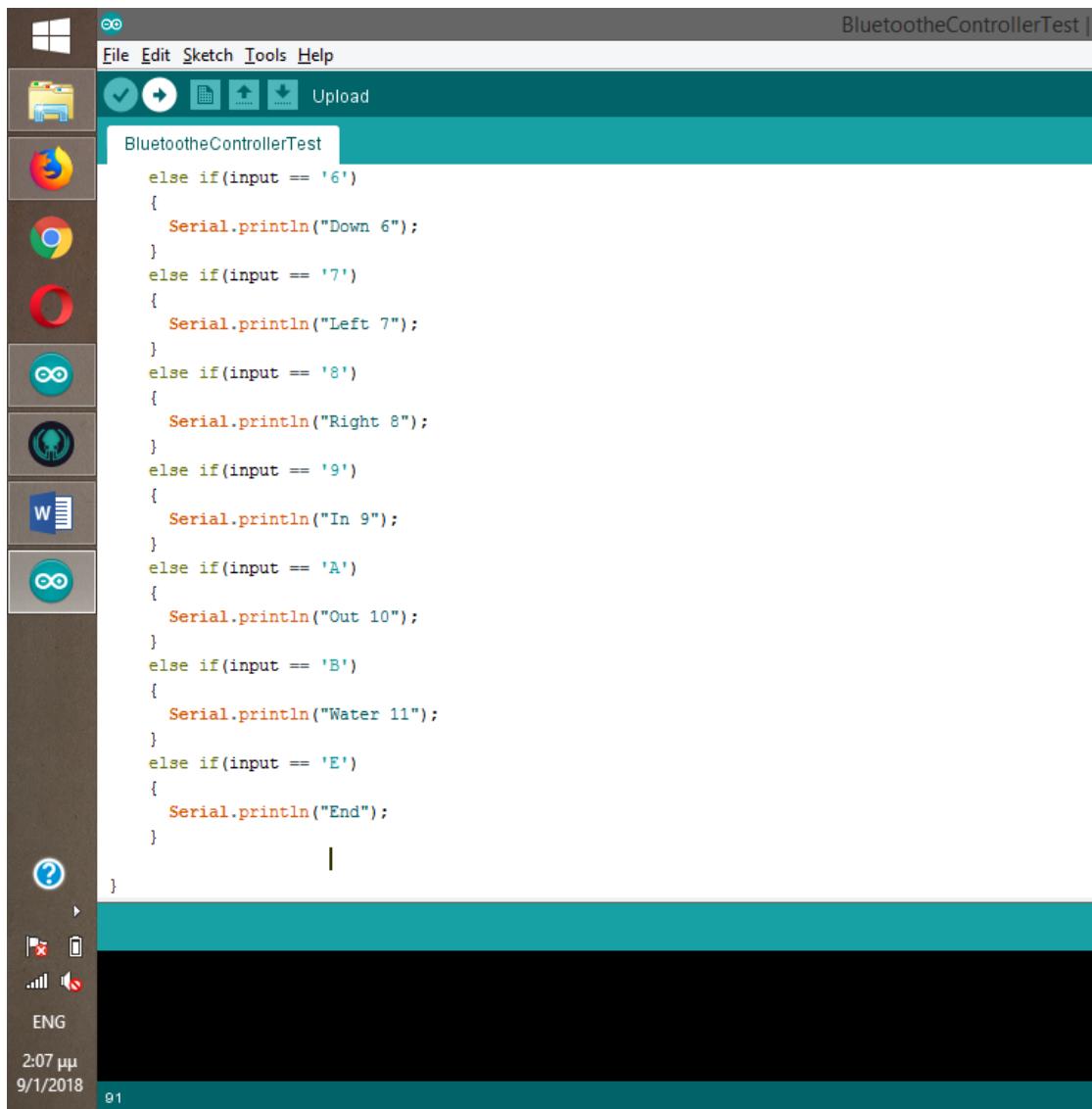
The screenshot shows the Arduino IDE interface. The title bar reads "BluetootheControllerTest | Arduino 1". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for Open, Save, Print, Upload, and Download. The main area displays the following C++ code:

```
if(Serial.available()>0) {
    a = Serial1.read();
    if(a=='1'||a=='2'||a=='3'||a=='4'||a=='5'||a=='6'||a=='7'||a=='8'||a=='9'||a=='A'||a=='B'||a=='E') {
        input = a;
    }
}

if(input == '1')
{
    Serial.println("Front 1");
}
else if(input == '2')
{
    Serial.println("Back 2");
}
else if(input == '3')
{
    Serial.println("Left 3");
}
else if(input == '4')
{
    Serial.println("Right 4");
}
else if(input == '5')
{
    Serial.println("Up ⬆");
}
else if(input == '6')
{
    Serial.println("Down ⬇");
}
```

The left sidebar contains icons for various applications like File Explorer, Task View, Start, Taskbar, and a search bar. The bottom status bar shows "2:06 μμ", "9/1/2018", and "61".

Για να εξαλείψουμε τις πιθανότητες ενεργοποίησης από θορύβους χρησιμοποιήσαμε ένα διπλό if για να ελέγχουμε με την Serial.available() αν υπάρχουν δεδομένα και με ένα σύνθετο if αν από αυτά τα δεδομένα είναι κάτι που μας ενδιαφέρει το αποθηκεύουμε στην μεταβλητή input και μετέπειτα ελέγχουμε αν η τιμή input έχει κάποια χρήσιμη τιμή με ένα πολλαπλό if.

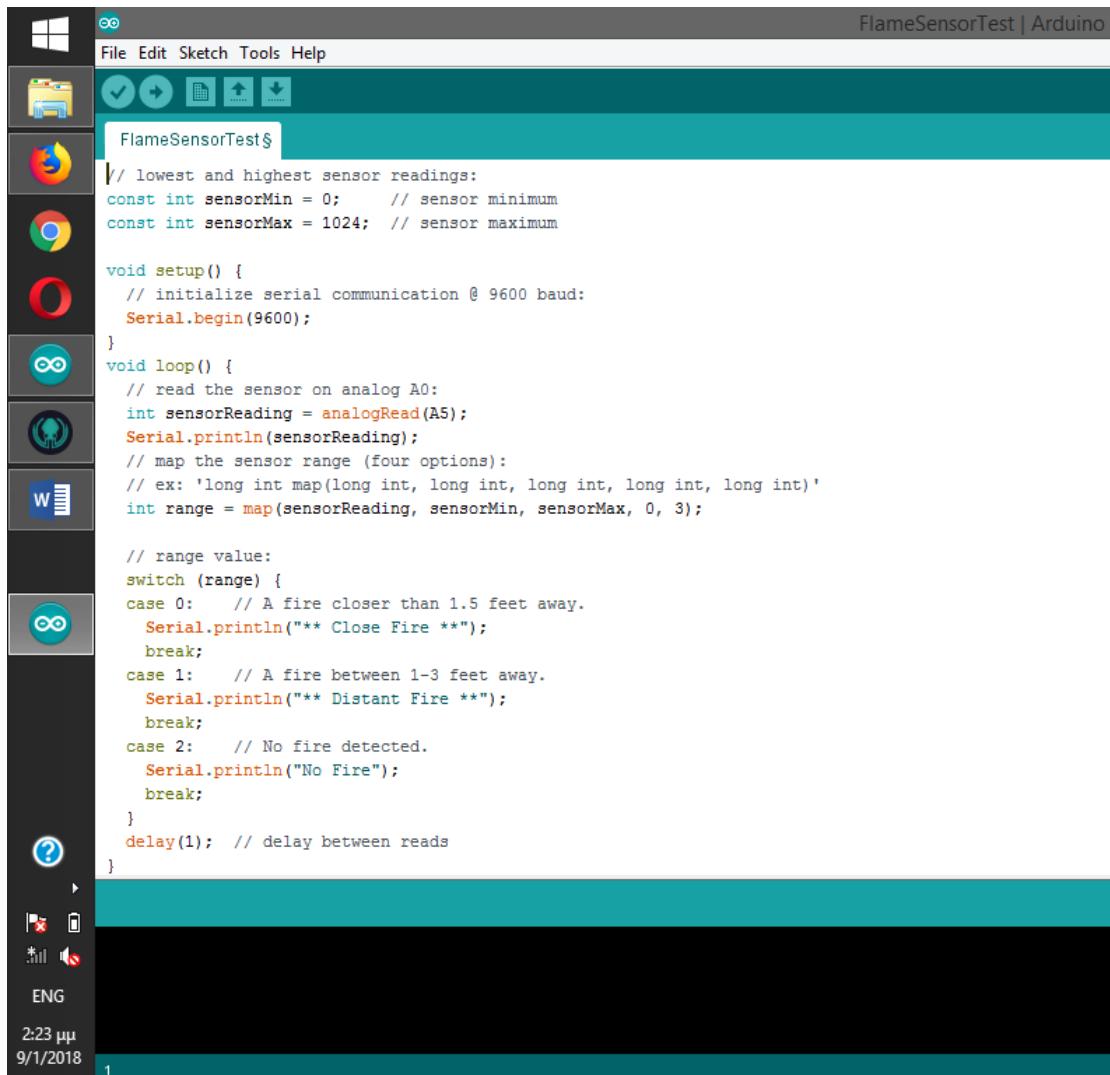


```
BluetootheControllerTest
else if(input == '6')
{
    Serial.println("Down 6");
}
else if(input == '7')
{
    Serial.println("Left 7");
}
else if(input == '8')
{
    Serial.println("Right 8");
}
else if(input == '9')
{
    Serial.println("In 9");
}
else if(input == 'A')
{
    Serial.println("Out 10");
}
else if(input == 'B')
{
    Serial.println("Water 11");
}
else if(input == 'E')
{
    Serial.println("End");
}
```

Τέλος για να βεβαιωθούμε ότι δουλεύει σωστά τυπώνουμε ένα μήνυμα στην συριακή ανάλογο με το κουμπί που πατήθηκε στο android.

## Flame Sensor Test

Με τον παρακάτω κώδικα δοκιμάσαμε πώς λειτουργεί ο flame sensor



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** FlameSensorTest | Arduino
- File Menu:** File Edit Sketch Tools Help
- Sketch Area:** The code for "FlameSensorTest" is displayed:

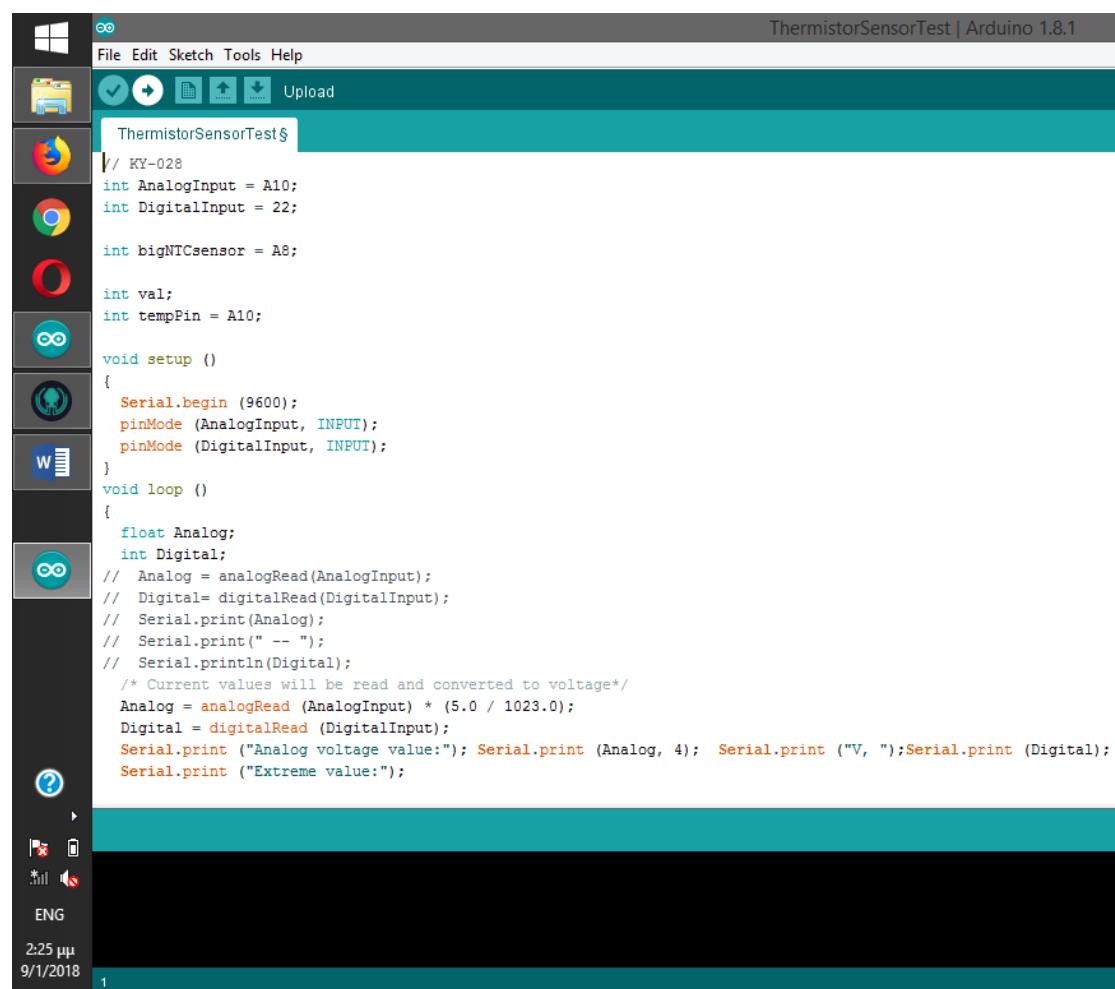
```
// lowest and highest sensor readings:  
const int sensorMin = 0;      // sensor minimum  
const int sensorMax = 1024;    // sensor maximum  
  
void setup() {  
    // initialize serial communication @ 9600 baud:  
    Serial.begin(9600);  
}  
void loop() {  
    // read the sensor on analog A0:  
    int sensorReading = analogRead(A5);  
    Serial.println(sensorReading);  
    // map the sensor range (four options):  
    // ex: 'long int map(long int, long int, long int, long int)'  
    int range = map(sensorReading, sensorMin, sensorMax, 0, 3);  
  
    // range value:  
    switch (range) {  
    case 0:    // A fire closer than 1.5 feet away.  
        Serial.println("** Close Fire **");  
        break;  
    case 1:    // A fire between 1-3 feet away.  
        Serial.println("** Distant Fire **");  
        break;  
    case 2:    // No fire detected.  
        Serial.println("No Fire");  
        break;  
    }  
    delay(1); // delay between reads  
}
```
- Tools Area:** Includes icons for upload, refresh, and other tools.
- Bottom Status Bar:** Shows "ENG", the date "9/1/2018", and the time "2:23 μμ".

Αυτό που κάναμε ήταν ότι απλά διαβάσαμε την τιμή που μας στέλνει ο flame sensor. Η οποία όπως προείπαμε είναι ~5V αν δεν υπάρχει φωτιά και ~0V αν υπάρχει. Αυτό στο Arduino μεταφράζεται σε 0=0V και 1023=5V. Με την χρήση της εντολής `map` αλλάζουμε την κλίμακα της τιμής που διαβάζουμε από 0 – 1023 σε μια τιμή από 0 – 3 και το αποτέλεσμα το αποθηκεύουμε στην ακέραια μεταβλητή `range` το οποίο πρακτικά αυτό που κάνει είναι να μετατρέπει την πιθανότητα σε 3 βασικές τιμές 0 – 1 – 2, οπού 0 σημαίνει φωτιά, 1 σημαίνει μεγάλη

πιθανότητα φωτιάς, 2 όχι φωτιά. Έτσι με μια switch case τυπώνουμε το ανάλογο μήνυμα.

### Thermistor KY-028 Sensor Test

Τέλος με τον KY-028 κάναμε πολλά πειράματα όπως το να μετρήσουμε θερμοκρασία, την τάση που μας επιστρέφει, και διάφορα άλλα πράγματα. Καταλήξαμε ότι η πιο χρήσιμη λειτουργία του για αυτό το project είναι το διάβασμα της ψηφιακής εξόδου του, το οποίο γίνεται 1(ένα) αν η θερμοκρασία ξεπεράσει το κατώφλι που του έχουμε ορίσει



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** ThermistorSensorTest | Arduino 1.8.1
- File Menu:** File Edit Sketch Tools Help
- Upload Buttons:** Checkmark, Refresh, Select Board, Upload, Save As
- Sketch Area:** Contains the C++ code for the Thermistor Sensor Test:

```
// KY-028
int AnalogInput = A10;
int DigitalInput = 22;

int bigNTCsensor = A8;

int val;
int tempPin = A10;

void setup ()
{
    Serial.begin (9600);
    pinMode (AnalogInput, INPUT);
    pinMode (DigitalInput, INPUT);
}

void loop ()
{
    float Analog;
    int Digital;
    // Analog = analogRead(AnalogInput);
    // Digital = digitalRead(DigitalInput);
    // Serial.print(Analog);
    // Serial.print(" -- ");
    // Serial.println(Digital);
    /* Current values will be read and converted to voltage*/
    Analog = analogRead (AnalogInput) * (5.0 / 1023.0);
    Digital = digitalRead (DigitalInput);
    Serial.print ("Analog voltage value:"); Serial.print (Analog, 4); Serial.print ("V, ");Serial.print (Digital);
    Serial.print ("Extreme value:");

    
```
- Toolbox:** Includes icons for File, Open, Save, Print, Find, Copy, Paste, Undo, Redo, Select Board, Upload, Save As, and a question mark icon.
- Bottom Status Bar:** Shows the text "ENG", the date "2:25 μμ 9/1/2018", and the page number "1".

The screenshot shows the Arduino IDE interface. The title bar reads "ThermistorSensorTest | Arduino 1.8.1". The main window displays the following C++ code:

```
File Edit Sketch Tools Help
ThermistorSensorTest
Serial.print ("Analog voltage value:"); Serial.print (Analog, 4); Serial.print ("V, ");Serial.print (Digital);
Serial.print ("Extreme value:");

if(Digital==1)
{
    Serial.println (" reached");
}
else
{
    Serial.println (" not yet reached");
}
delay (1000);

float cel = analogRead(tempPin)/2;
int readVal=analogRead(bigNTCsensor);
double temp = Thermistor(readVal);
Serial.print("Temperature big sensor is = ");
Serial.print(temp);
Serial.println(" Celsius, ");

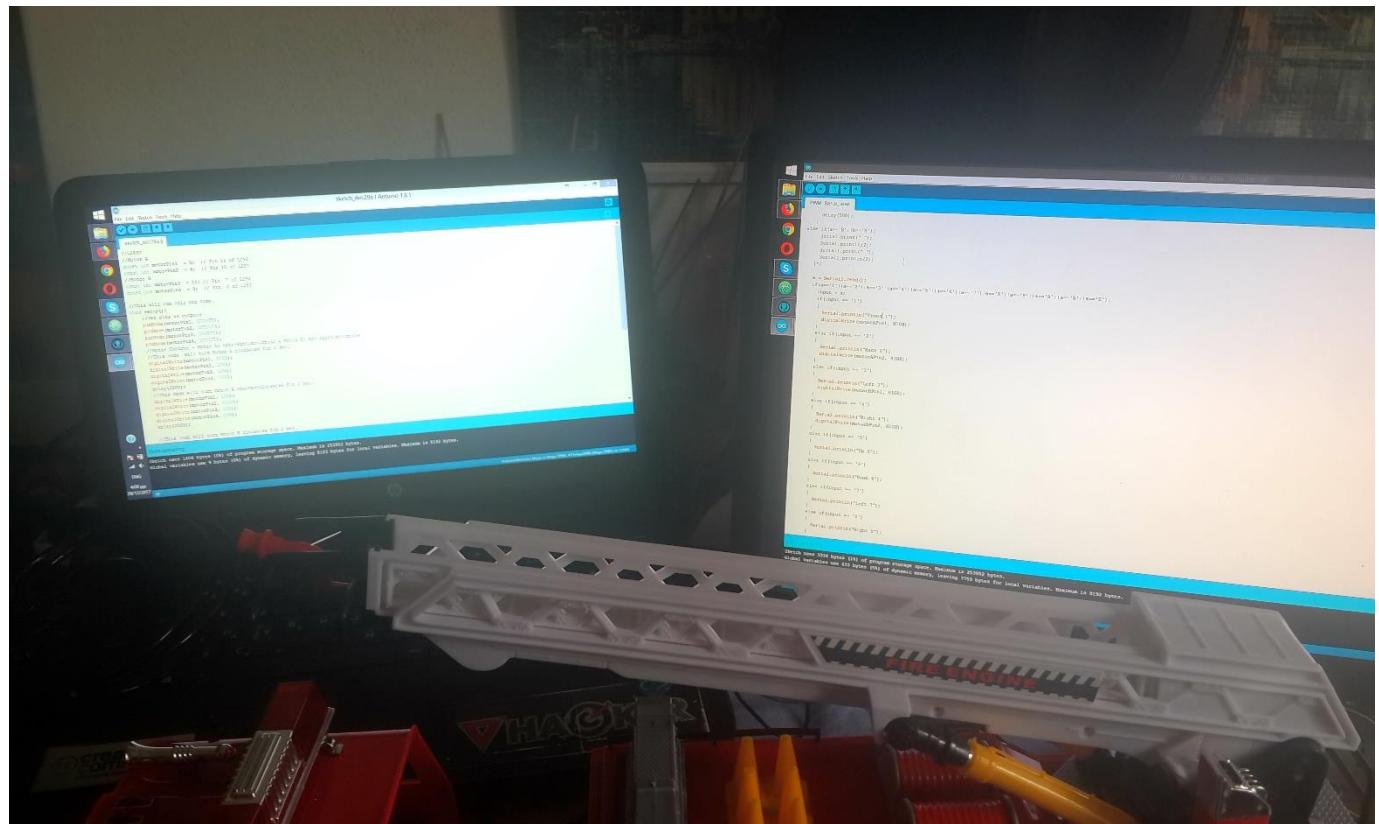
}

double Thermistor(int RawADC) {
    double Temp;
    Temp = log(10000.0*((1024.0/RawADC-1)));
    Temp = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * Temp * Temp ))* Temp );
    Temp = Temp - 273.15;           // Convert Kelvin to Celcius
    //Temp = (Temp * 9.0)/ 5.0 + 32.0; // Convert Celcius to Fahrenheit
    return Temp;
}
```

The code includes a function to calculate temperature from an NTC sensor reading. The IDE's status bar at the bottom shows "2:26 μμ", "9/1/2018", and "23".

## Main code

Μετά από όλα αυτά αρχίσαμε να γράφουμε το κύριο πρόγραμμα του project μας.



```
const int analogInPin = A5;
const int analogOutPin = 9;
int sensorValue = 0;
int outputValue = 0;
int input='0';

const int Alarm = 3; // Pin 14 of L293

//L293D 1
//Motor A
const int motorAPin1 = 4; // Pin 14 of L293
const int motorAPin2 = 5; // Pin 10 of L293
//L293D 1
//Motor B
const int motorBPin1 = 6; // Pin 7 of L293
const int motorBPin2 = 7; // Pin 2 of L293
//L293D 2
//Motor C
const int motorCPin1 = 8; // Pin 14 of L293
const int motorCPin2 = 9; // Pin 10 of L293
//L293D 2
//Motor D
const int motorDPin1 = 10; // Pin 7 of L293
const int motorDPin2 = 11; // Pin 2 of L293
//L293D 3
//Motor E
const int motorEPin1 = 12; // Pin 14 of L293
const int motorEPin2 = 13; // Pin 10 of L293
```

Αρχικά δηλώσαμε όλα τα pins τα οποία θα οδηγούν τα moter καθώς και στην συνέχεια όλους τους αισθητήρες.

The screenshot shows the Arduino IDE interface. The title bar reads "ArduinoFire | Arduino". The menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar contains icons for Open, Save, Upload, and other functions. The main area displays the following C++ code:

```
//L298
//Motor F
const int motorWaterPin1 = 30; // Pin 30 of L293
const int motorWaterPin2 = 31; // Pin 31 of L293

//Flame sensor Front 5-channels
const int FlamePinFrontL1 = A0; // This is our input pin
const int FlamePinFrontL2 = A1; // This is our input pin
const int FlamePinFrontC1 = A2; // This is our input pin
const int FlamePinFrontR1 = A3; // This is our input pin
const int FlamePinFrontR2 = A4; // This is our input pin

//Flame sensor Right
const int FlamePinRight = A5; // This is our input pin
//Flame sensor Left
const int FlamePinLeft = A6; // This is our input pin
//Flame sensor Back
const int FlamePinBack = A7; // This is our input pin

const int sensorMin = 0;      // sensor minimum
const int sensorMax = 1024; // sensor maximum

void motor_test();

void setup() {
    Serial.begin(9600);
    Serial1.begin(9600);
    Serial.println("HELLO");
    Serial.println(5);
    //Set pins as outputs
```

Στο συγκεκριμένο project χρησιμοποιούμε 4 συριακά pin. 2 για Debugging και 2 για την επικοινωνία μέσω του Bluetooth. Επομένως, χρησιμοποιούμε το Serial.begin(9600) και το Serial1.begin(9600), καθώς το Arduino mega έχει 4 συριακά ζευγάρια.

The screenshot shows the Arduino IDE interface. The title bar reads "ArduinoFire [ ]". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for Save, Undo, Redo, Cut, Copy, Paste, Select All, Find, Replace, and Upload. The main area displays the following C++ code:

```
//Set pins as outputs
pinMode(Alarm, OUTPUT);
pinMode(motorAPin1, OUTPUT);
pinMode(motorBPin1, OUTPUT);
pinMode(motorBPin2, OUTPUT);
pinMode(motorCPin1, OUTPUT);
pinMode(motorCPin2, OUTPUT);
pinMode(motorDPin1, OUTPUT);
pinMode(motorDPin2, OUTPUT);
pinMode(motorWaterPin1, OUTPUT);
pinMode(motorWaterPin2, OUTPUT);
//motor_test();
}

void loop() {
//char a,b;
int a;
int Flame1,Flame2,Flame3,Flame4,Flame5,Flame6,Flame7,Flame8;
int range1=0,range2=0,range3=0,range4=0,range5=0,range6=0,range7=0,range8=0;
/* a=Serial.read();
b=Serial1.read();
if(a=='A'||b=='A') {
    Serial.print(" ");
    Serial.println(1);
    Serial1.print(" ");
    Serial1.println(1);
    delay(500);
}
else if(a=='B'||b=='B') {
```

Σε αυτό το κομμάτι δηλώνουμε κάποιες μεταβλητές καθώς και τα αντίστοιχα ριν για εξόδους και εισόδους.

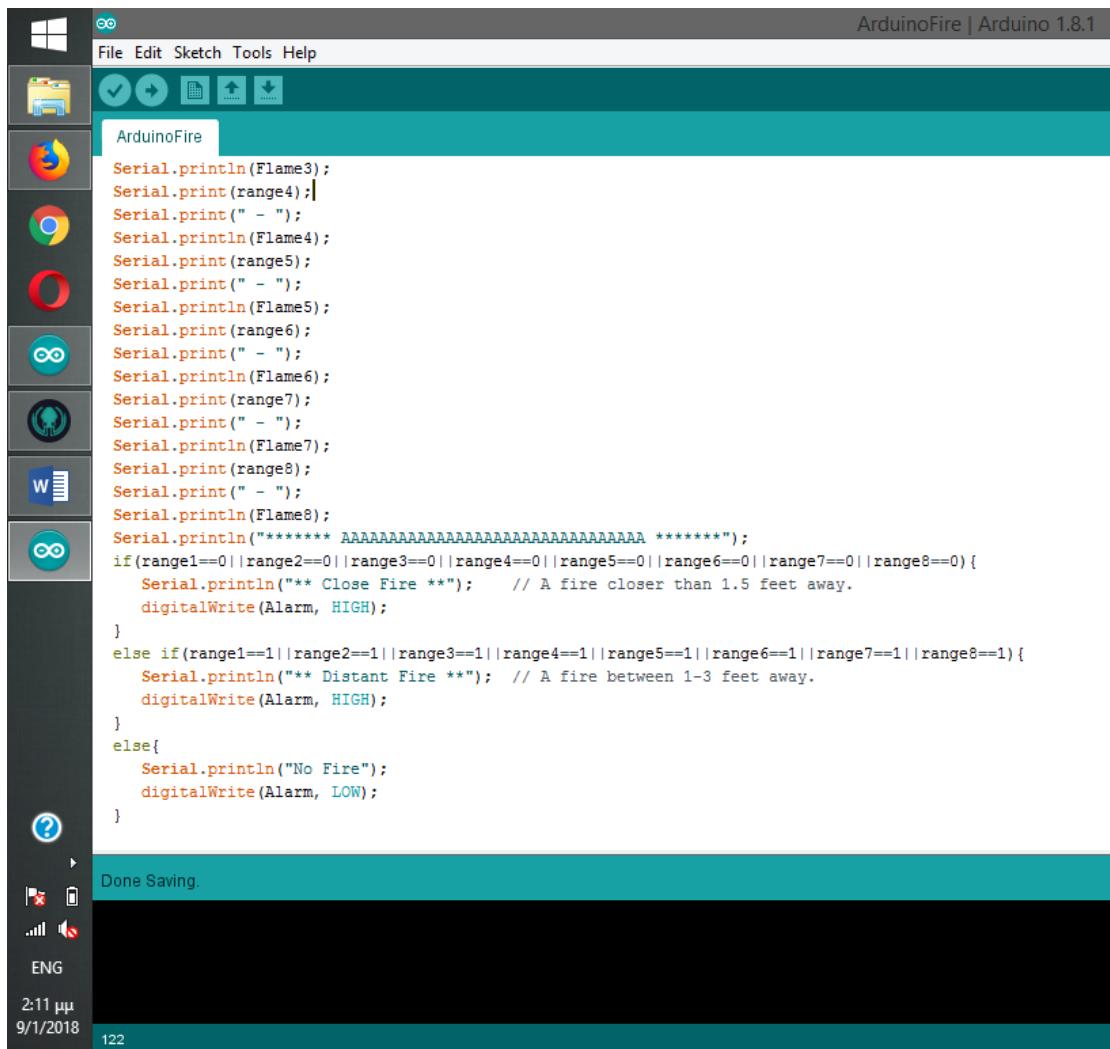
```
File Edit Sketch Tools Help
ArduinoFire
Serial.print(" ");
Serial.println(2);
Serial.print(" ");
Serial.println(2);
} */

Flame1 = analogRead(FlamePinFrontL1);
Flame2 = analogRead(FlamePinFrontL2);
Flame3 = analogRead(FlamePinFrontC1);
Flame4 = analogRead(FlamePinFrontR1);
Flame5 = analogRead(FlamePinFrontR2);
Flame6 = analogRead(FlamePinRight);
Flame7 = analogRead(FlamePinLeft);
Flame8 = analogRead(FlamePinBack);

range1 = map(Flame1, sensorMin, sensorMax, 0, 3);
range2 = map(Flame2, sensorMin, sensorMax, 0, 3);
range3 = map(Flame3, sensorMin, sensorMax, 0, 3);
range4 = map(Flame4, sensorMin, sensorMax, 0, 3);
range5 = map(Flame5, sensorMin, sensorMax, 0, 3);
range6 = map(Flame6, sensorMin, sensorMax, 0, 3);
range7 = map(Flame7, sensorMin, sensorMax, 0, 3);
range8 = map(Flame8, sensorMin, sensorMax, 0, 3);

Serial.print(range1);
Serial.print(" - ");
Serial.println(Flame1);
Serial.print(range2);
Serial.print(" - ");
Serial.println(Flame2);
```

Διαβάζουμε τα περιεχόμενα από τους αισθητήρες και τα κάνουμε μαρόπως είδαμε και στο παράδειγμα με τον αισθητήρα φωτιάς. Και τυπώνουμε τα αποτελέσματα για επαλήθευση στην συριακή.



Με ένα σύνθετο if ελέγχουμε αν κάποιος από τους αισθητήρες έχει εντοπίσει φωτιά. Και αν ναι τότε ενεργοποιούμε ένα Alarm το οποίο σταματάει όταν σταματήσει και η φωτιά.

```
a = Serial1.read();
if(a=='1'||a=='2'||a=='3'||a=='4'||a=='5'||a=='6'||a=='7'||a=='8'||a=='9'||a=='A'||a=='B'||a=='E'||a=='G'||a=='D') {
    input = a;
    if(input == '1')
    {
        Serial.println("Front 1");
        digitalWrite(motorAPin1, HIGH);
    }
    else if(input == '2')
    {
        Serial.println("Back 2");
        digitalWrite(motorAPin2, HIGH);
    }
    else if(input == '3')
    {
        Serial.println("Left Front 3");
        //digitalWrite(motorAPin1, HIGH);
        digitalWrite(motorBPin1, HIGH);
    }
    else if(input == '4')
    {
        Serial.println("Right Front 4");
        //digitalWrite(motorAPin1, HIGH);
        digitalWrite(motorBPin2, HIGH);
    }
    else if(input == '5')
    {
        digitalWrite(motorDPin1, HIGH);
        Serial.println("Up 5");
    }
}
```

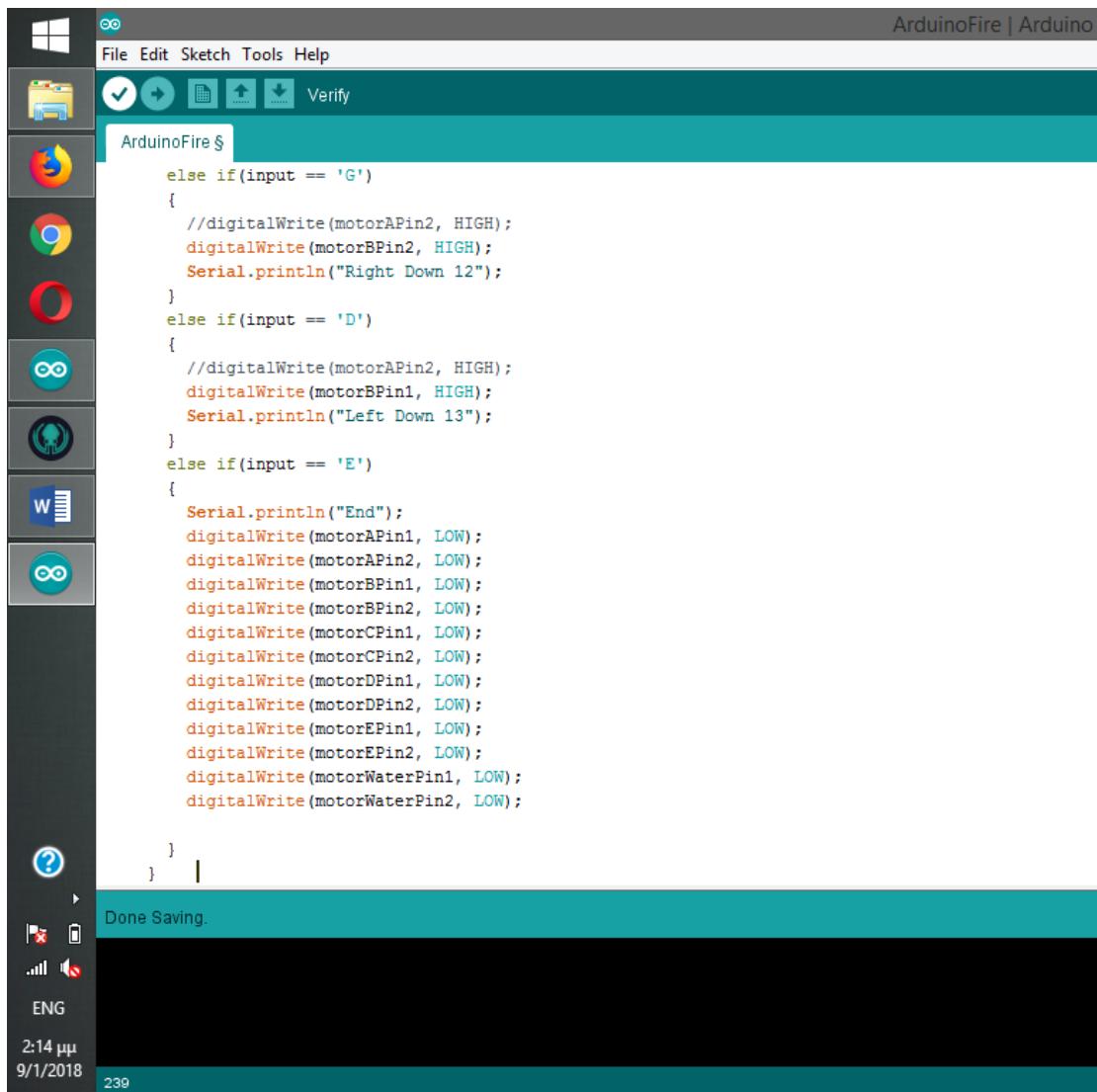
Done Saving.

Σε αυτό το μέρος του κώδικα με ένα σύνθετο if ελέγχουμε το τί έρχεται από την συριακή, δηλαδή το Bluetooth και αν είναι κάτι χρήσιμο, δηλαδή έχει πατηθεί κάποιο πλήκτρο στο Android, ενεργοποιούμε τον αντίστοιχο κινητήρα μέχρις ότου έρθει το σήμα τερματισμού που όπως προείπαμε, είναι το “Ε” και αποστέλλεται όταν αφήσουμε το αντίστοιχο πλήκτρο από την συσκευή android.

```
ArduinoFire §

else if(input == '6')
{
    digitalWrite(motorDPin2, HIGH);
    Serial.println("Down 6");
}
else if(input == '7')
{
    digitalWrite(motorCPin1, HIGH);
    Serial.println("Left 7");
}
else if(input == '8')
{
    digitalWrite(motorCPin2, HIGH);
    Serial.println("Right 8");
}
else if(input == '9')
{
    digitalWrite(motorEPin1, HIGH);
    Serial.println("In 9");
}
else if(input == 'A')
{
    digitalWrite(motorEPin2, HIGH);
    Serial.println("Out 10");
}
else if(input == 'B')
{
    digitalWrite(motorWaterPin1, HIGH);
    Serial.println("Water 11");
}

Done Saving.
```



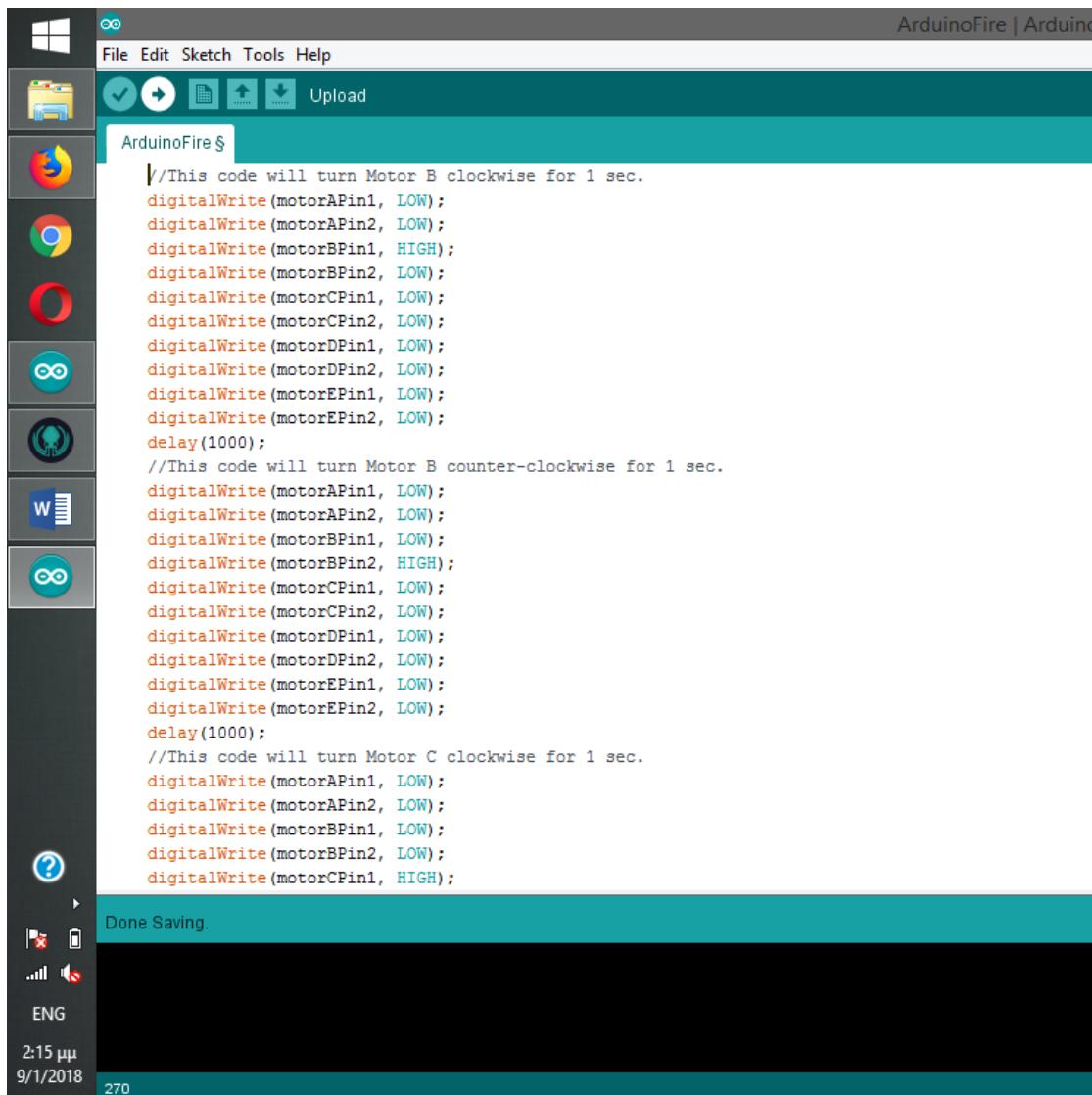
Τέλος έχουμε γράψει ένα μικρό test για τα moter και την σειρήνα τα οποία ενεργοποιούνται στο setup του προγράμματος για να μας δείξουν ότι όλα λειτουργούν.

The screenshot shows the Arduino IDE interface. The title bar reads "ArduinoFire | Arduino". The menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar contains icons for Open, Save, Print, and Upload. The sketch window displays the following code:

```
void motor_test(){
    //Motor Control - Motor A: motorPin1,motorpin2 & Motor B: motorpin3,motorpin4
    //This code will turn Motor A clockwise for 2 sec.
    digitalWrite(motorAPin1, HIGH);
    digitalWrite(motorAPin2, LOW);
    digitalWrite(motorBPin1, LOW);
    digitalWrite(motorBPin2, LOW);
    digitalWrite(motorCPin1, LOW);
    digitalWrite(motorCPin2, LOW);
    digitalWrite(motorDPin1, LOW);
    digitalWrite(motorDPin2, LOW);
    digitalWrite(motorEPin1, LOW);
    digitalWrite(motorEPin2, LOW);

    delay(1000);
    //This code will turn Motor A counter-clockwise for 1 sec.
    digitalWrite(motorAPin1, LOW);
    digitalWrite(motorAPin2, HIGH);
    digitalWrite(motorBPin1, LOW);
    digitalWrite(motorBPin2, LOW);
    digitalWrite(motorCPin1, LOW);
    digitalWrite(motorCPin2, LOW);
    digitalWrite(motorDPin1, LOW);
    digitalWrite(motorDPin2, LOW);
    digitalWrite(motorEPin1, LOW);
    digitalWrite(motorEPin2, LOW);
    delay(1000);
```

The status bar at the bottom shows "Done Saving.", "ENG", "2:14 μμ", "9/1/2018", and "240".



The screenshot shows the Arduino IDE interface. The title bar reads "ArduinoFire | Arduino 1". The menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar contains icons for Open, Save, Upload, and other functions. The main window displays a sketch named "ArduinoFire.ino" with the following code:

```
digitalWrite(motorCPin2, LOW);
digitalWrite(motorDPin1, LOW);
digitalWrite(motorDPin2, LOW);
digitalWrite(motorEPin1, LOW);
digitalWrite(motorEPin2, LOW);
delay(1000);
//This code will turn Motor C counter-clockwise for 1 sec.
digitalWrite(motorAPin1, LOW);
digitalWrite(motorAPin2, LOW);
digitalWrite(motorBPin1, LOW);
digitalWrite(motorBPin2, LOW);
digitalWrite(motorCPin1, LOW);
digitalWrite(motorCPin2, HIGH);
digitalWrite(motorDPin1, LOW);
digitalWrite(motorDPin2, LOW);
digitalWrite(motorEPin1, LOW);
digitalWrite(motorEPin2, LOW);
delay(1000);
//This code will turn Motor D clockwise for 1 sec.
digitalWrite(motorAPin1, LOW);
digitalWrite(motorAPin2, LOW);
digitalWrite(motorBPin1, LOW);
digitalWrite(motorBPin2, LOW);
digitalWrite(motorCPin1, LOW);
digitalWrite(motorCPin2, LOW);
digitalWrite(motorDPin1, HIGH);
digitalWrite(motorDPin2, LOW);
digitalWrite(motorEPin1, LOW);
digitalWrite(motorEPin2, LOW);
delay(1000);
```

The status bar at the bottom shows "Done Saving.", "ENG", "2:15 μμ", "9/1/2018", and "320".

The screenshot shows the Arduino IDE interface. The title bar reads "ArduinoFire | Arduino". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for Open, Save, Verify, and others. The main area displays a sketch titled "ArduinoFire §". The code in the sketch is as follows:

```
//This code will turn Motor D counter-clockwise for 1 sec.  
digitalWrite(motorAPin1, LOW);  
digitalWrite(motorAPin2, LOW);  
digitalWrite(motorBPin1, LOW);  
digitalWrite(motorBPin2, LOW);  
digitalWrite(motorCPin1, LOW);  
digitalWrite(motorCPin2, LOW);  
digitalWrite(motorDPin1, LOW);  
digitalWrite(motorDPin2, HIGH);  
digitalWrite(motorEPin1, LOW);  
digitalWrite(motorEPin2, LOW);  
delay(1000);  
  
//This code will turn Motor E clockwise for 1 sec.  
digitalWrite(motorAPin1, LOW);  
digitalWrite(motorAPin2, LOW);  
digitalWrite(motorBPin1, LOW);  
digitalWrite(motorBPin2, LOW);  
digitalWrite(motorCPin1, LOW);  
digitalWrite(motorCPin2, LOW);  
digitalWrite(motorDPin1, LOW);  
digitalWrite(motorDPin2, HIGH);  
digitalWrite(motorEPin1, LOW);  
digitalWrite(motorEPin2, LOW);  
delay(1000);  
  
//This code will turn Motor E counter-clockwise for 1 sec.  
digitalWrite(motorAPin1, LOW);  
digitalWrite(motorAPin2, LOW);  
digitalWrite(motorBPin1, LOW);  
digitalWrite(motorBPin2, LOW);  
digitalWrite(motorCPin1, LOW);
```

A message "Done Saving." is displayed in the status bar at the bottom.

The screenshot shows the Arduino IDE interface. The title bar reads "ArduinoFire | Arduino". The menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar has icons for Open, Save, Print, and others. The left sidebar contains icons for Windows, File, Sketch, Tools, Help, and a question mark. The main area displays the following C++ code:

```
digitalWrite(motorDPin2, LOW);
digitalWrite(motorEPin1, HIGH);
digitalWrite(motorEPin2, LOW);
delay(1000);
//This code will turn Motor E counter-clockwise for 1 sec.
digitalWrite(motorAPin1, LOW);
digitalWrite(motorAPin2, LOW);
digitalWrite(motorBPin1, LOW);
digitalWrite(motorBPin2, LOW);
digitalWrite(motorCPin1, LOW);
digitalWrite(motorCPin2, LOW);
digitalWrite(motorDPin1, LOW);
digitalWrite(motorDPin2, LOW);
digitalWrite(motorEPin1, LOW);
digitalWrite(motorEPin2, HIGH);
delay(1000);

//And this code will stop motors
digitalWrite(motorAPin1, LOW);
digitalWrite(motorAPin2, LOW);
digitalWrite(motorBPin1, LOW);
digitalWrite(motorBPin2, LOW);
digitalWrite(motorCPin1, LOW);
digitalWrite(motorCPin2, LOW);
digitalWrite(motorDPin1, LOW);
digitalWrite(motorDPin2, LOW);
digitalWrite(motorEPin1, LOW);
digitalWrite(motorEPin2, LOW);

}

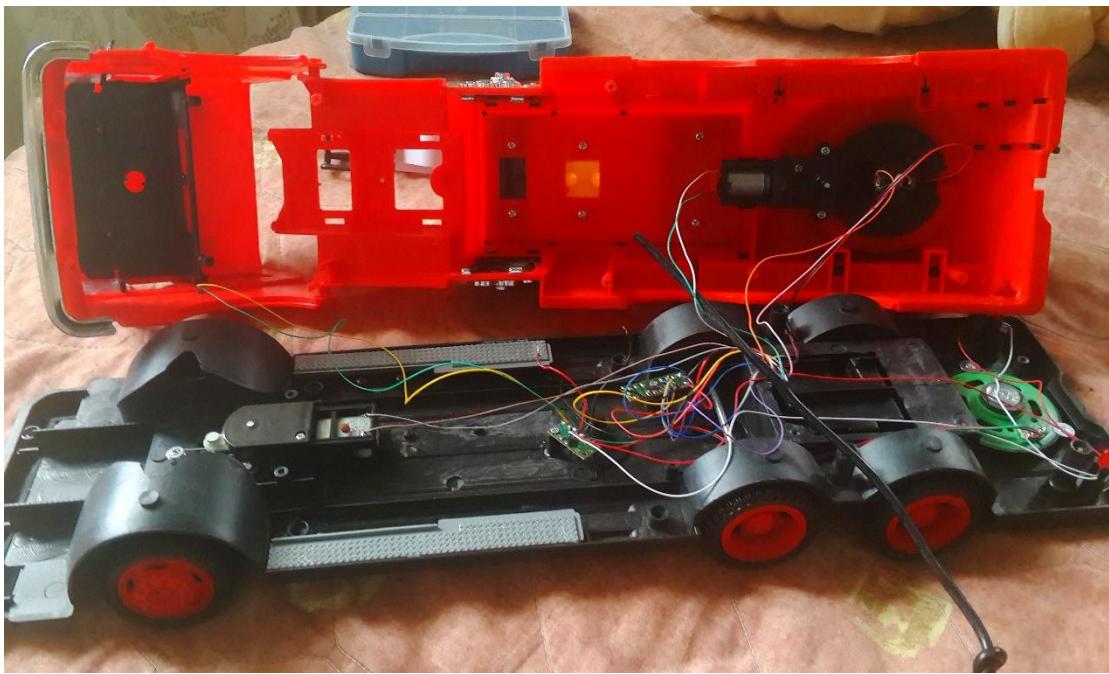
Done Saving.
```

The status bar at the bottom shows "ENG", "2:16 μμ", "9/1/2018", and "379".

# Συναρμολόγηση



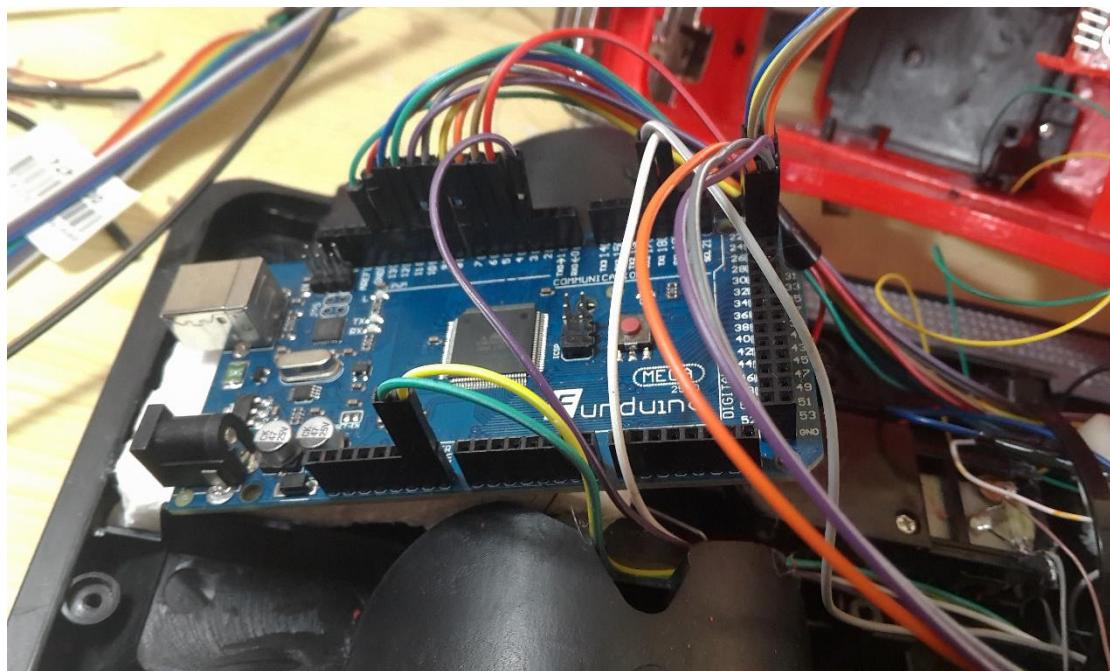
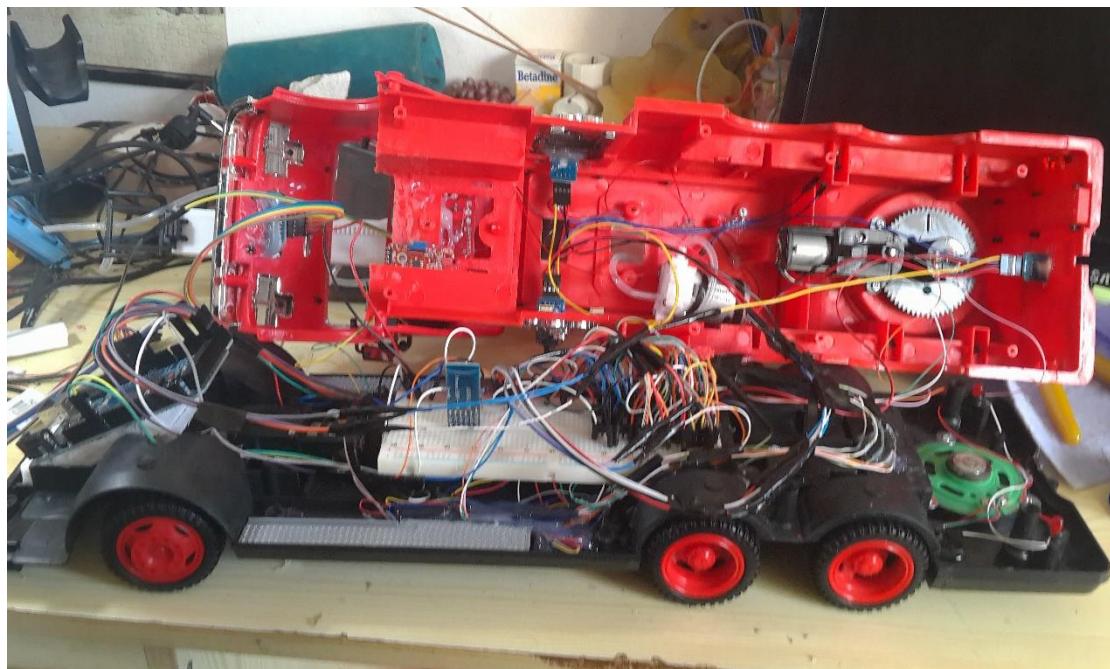
Το όχημα το οποίο θα διαμορφώναμε ήταν ένα πυροσβεστικό που εκινείτο με ένα ενσύρματο χειριστήριο. Το ξεβιδώσαμε και αρχίσαμε να παρατηρούμε τους μηχανισμούς που είχε για να κινείται και για να στρίβει καθώς είχε ένα πρόβλημα με το στρίψιμο. Το σύστημα στροφής του λειτουργούσε με έναν dc κινητήρα ο οποίος για κάποιο λόγο κόλλαγε και δεν έστριβε. Έτσι σκεφτήκαμε να αλλάξουμε dc κινητήρα ή και να αλλάξουμε το σύστημα με ένα Servo. Το επόμενο βήμα που κάναμε ήταν να κάνουμε χώρο έτσι ώστε να χωρέσουν όλα τα εξαρτήματα μέσα στο εσωτερικό του οχήματος.

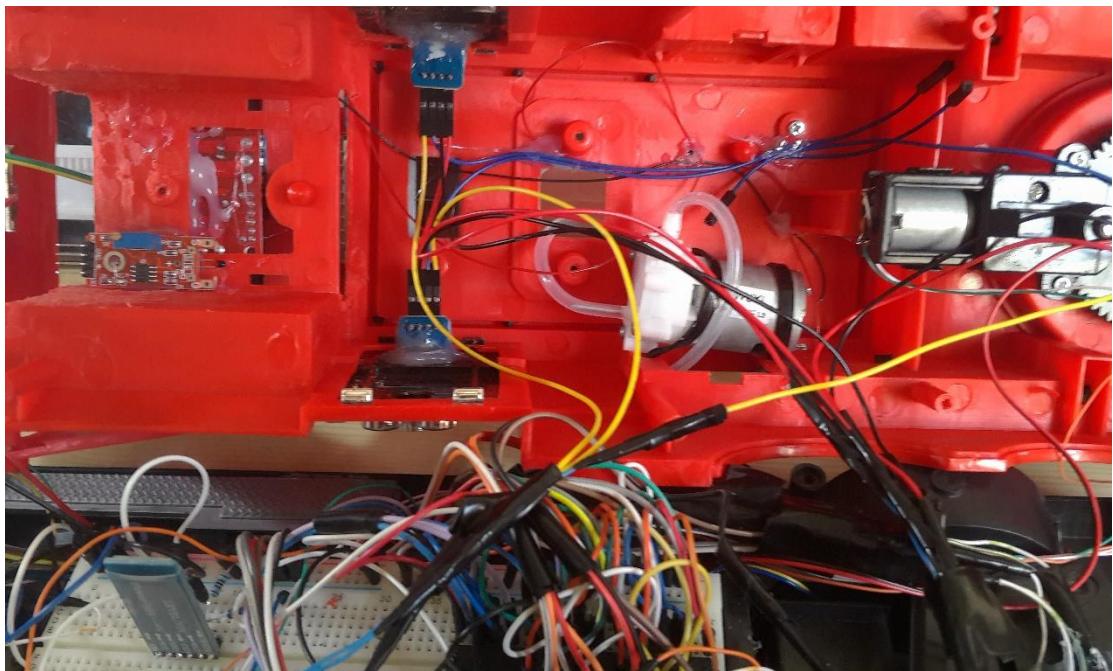
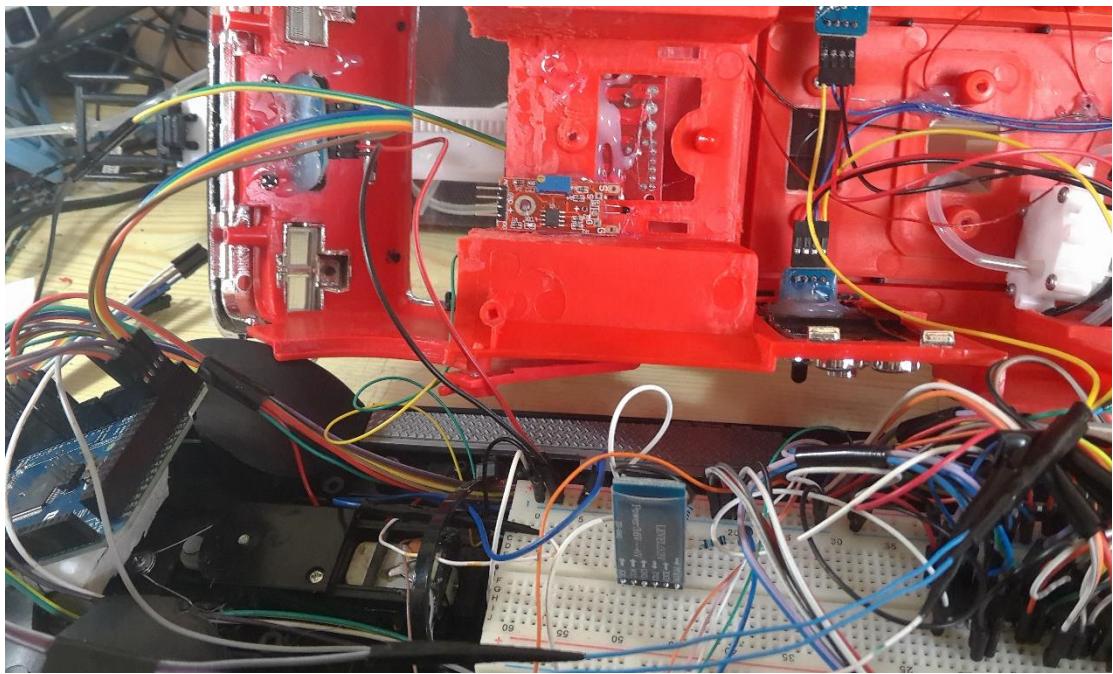


Μετά από αρκετές δοκιμές βρήκαμε το ιδανικό μέρος για να τα τοποθετήσουμε το Arduino καθώς και το breadboard με τους 3 drivers και το Bluetooth.



Ανοίξαμε τρύπες όπου χρειαζόταν για να περάσουμε τους αισθητήρες, τους σταθεροποιήσαμε με Θερμοσυλικόνη και πλαστικά Δεματικά και τους συνδέσαμε πάνω στο Arduino.





Για μπαταρίες ανοίξαμε μια παλιά μπαταρία από laptop, και μετρήσαμε τις μπαταρίες 18650 που δεν ήταν εξαντλημένες.



Τις φορτίσαμε και τις συνδέσαμε σε συστοιχία των 12v και περίπου 3-3.5Ah



Τέλος τις τοποθετήσαμε στο όχημα και αρχίσαμε τις δοκιμές



## Προβλήματα...

Συναντήσαμε πάρα πολλά προβλήματα όπως αστοχία υλικών, εργοστασιακά λάθη-προβλήματα, προβλήματα χώρου. Ενώ τα περισσότερα από αυτά καταφέραμε και τα επιλύσαμε, κάποια άλλα πάλι όχι και μάθαμε να ζούμε με αυτά!!! Ας αναφέρουμε μερικά από αυτά που έκαναν την ζωή μας λίγο πιο δύσκολη.

## Χωρίς Συριακή!

Το πιο παράξενο πρόβλημα ήταν αυτό της συριακής επικοινωνίας μέσω του Bluetooth στο android. Για κάποιο περίεργο λόγο όταν συνδέαμε το Bluetooth στην πρώτη συριακή του Arduino, το Arduino IDE δεν μπορούσε να στείλει το πρόγραμμα στο Arduino. Όταν αποσυνέδεες το Bluetooth, το πρόγραμμα δεν περνούσε κανονικά και αν το συνέδεες, μετέπειτα λειτουργούσε μια χαρά. Αν προσπαθούσες να το

ξαναπεράσεις δεν το έπαιρνε για κανέναν λόγο. Το πιο πιθανό είναι λόγω του ότι η USB είναι συνδεμένη μέσω του USB to Serial με τις συριακές RX TX το Bluetooth το μπέρδευε και δεν μπορούσε να βρει το Arduino. Αυτό το λύσαμε πολύ απλά συνδέοντας το Bluetooth στην συριακή θύρα 1 του Arduino, δηλαδή RX1 TX1.

## Πάρτο Αλλιώς

Ένα πρόβλημα αρκετά περίεργο επίσης που συναντήσαμε ήταν το πρόβλημα με το στρίψιμο οπού χρειάστηκε να αλλάξουμε 3 dc κινητήρες και ένα κομμάτι πλαστικό για να καταφέρουμε να το επιλύσουμε. Το πρόβλημα ήταν το εξής: Το σύστημα στροφής κάποια στιγμή κολλούσε και έπρεπε να το κουνήσεις με το χέρι για να ξεκολλήσει. Αυτό συμπεράναμε τελικά ότι οφειλόταν σε εργοστασιακό λάθος και λάθος σχεδιασμό του συστήματος στροφής. Το Dc motor δεν πάταγε καλά στα γρανάζια στροφής με αποτέλεσμα να μπλοκάρει. Υπήρχε μια βίδα η οποία έσφιγγε το πλαστικό βάσης με το κάλυμμα η οποία όμως δεν έσφιγγε σωστά. Έτσι τοποθετήσαμε ένα κομμάτι μαλακό πλαστικό κάτω από το motor για να το πιέζει προς τα πάνω και να κάθεται στην θέση του.