



## ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

### **Ασύρματη απομακρυσμένη διαχείριση κινητού μικροελεγκτή και εφαρμογή εντοπισμού μέσω τεχνικών υπολογιστικής όρασης**

#### ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**ΣΤΕΡΓΙΟΣ Δ. ΔΗΜΑΣ**

**Επιβλέπων :** Θεολόγου Μιχαήλ  
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2016





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ  
ΠΛΗΡΟΦΟΡΙΚΗΣ

**Ασύρματη απομακρυσμένη διαχείριση κινητού  
μικροελεγκτή και εφαρμογή εντοπισμού μέσω τεχνικών  
υπολογιστικής όρασης**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**ΣΤΕΡΓΙΟΣ Δ. ΔΗΜΑΣ**

**Επιβλέπων :** Θεολόγου Μιχαήλ  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 4η Ιουλίου 2016.

.....  
Θεολόγου Μιχαήλ  
Καθηγητής Ε.Μ.Π.

.....  
Στασινόπουλος Γεώργιος  
Καθηγητής Ε.Μ.Π.

.....  
Συκάς Ευστάθιος  
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2016

.....  
**ΣΤΕΡΓΙΟΣ Δ. ΔΗΜΑΣ**

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Στέργιος Δ. Δήμας, 2016.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## Περίληψη

Ο σκοπός της διπλωματικής εργασίας ήταν η ανάπτυξη και η κατασκευή ενός συστήματος ικανό να χειριστεί απομακρυσμένα ένα μικρό όχημα (διαστάσεις μικρού τηλεκατευθυνόμενου αυτοκινήτου) χρησιμοποιώντας μεθόδους και αλγορίθμους αναγνώρισης εικόνας. Το βασικό στοιχείο που χρησιμοποιήθηκε για τον χειρισμό του οχήματος ήταν ο μικροελεγκτής Arduino Uno, μία ανοιχτού-κώδικα πλατφόρμα βασισμένη σε ευκολόχρηστα υλικά και λογισμικό. Ο αρχικός στόχος ήταν η ανάπτυξη του λογισμικού το οποίο εκτελείται σε υπολογιστή και έχει ως σκοπό τον απομακρυσμένο έλεγχο του Arduino, με την ελάχιστη δυνατή καθυστέρηση και την μέγιστη απόκριση. Ο δεύτερος στόχος της εργασίας ήταν η εύρεση της κατάλληλης μεθόδου αναγνώρισης εικόνας για τον εντοπισμό του οχήματος σε πραγματικό χρόνο, το οποίο κατέστησε αναγκαία μία εισαγωγική μελέτη και μια σειρά δοκιμών διαφόρων τεχνικών εντοπισμού κινούμενου αντικειμένου. Ο τελικός στόχος ήταν ο συνδυασμός όλων των επιμέρους στοιχείων και η ανάπτυξη ενός ενιαίου προγράμματος το οποίο, εκτελώντας το στον υπολογιστή, να είναι ικανό να ανιχνεύσει σε πραγματικό χρόνο το όχημα μέσω μίας κάμερας, να δημιουργεί την κατάλληλη διεπαφή για το χρήστη καθώς και να είναι υπεύθυνο για την αυτόματη πλοήγηση σύμφωνα με το σημείο προορισμού που έχει οριστεί.

**Λέξεις Κλειδιά:** Arduino, αναγνώριση εικόνας, τηλεκατευθυνόμενο όχημα, εντοπισμός αντικειμένου



## **Abstract**

The scope of this thesis was the development of a system that can remotely control and navigate a small vehicle (rc car model) by using computer vision algorithms and methods. The core component responsible for controlling the vehicle is the Arduino Uno, an open-source prototyping microcontroller platform based on easy-to-use hardware and software. The first goal was to develop the software that runs in our computer and can remotely control the Arduino, and to achieve minimum delay issues and maximum responsiveness. The second goal was to decide the computer vision method for tracking the vehicle, therefore an introductory study on the fields of object tracking algorithms was required. The final goal was to combine each part and build a program that runs in our computer, tracks in real time the vehicle through a camera, implements the appropriate user interface and auto-navigates the vehicle according to the set destination point.

**Keywords:** Arduino, remote control, rc car, computer vision, tracking





## Πίνακας περιεχομένων

<b>1</b>	<b>Εισαγωγή.....</b>	<b>11</b>
1.1	Η εμφάνιση των μικροελεγκτών .....	11
1.2	Εφαρμογές.....	16
1.3	Δυνατότητες – Περιορισμοί.....	17
1.4	Ιστορική εξέλιξη .....	18
1.5	Τρέχοντες μικροελεγκτές και αυξημένες δυνατότητες αυτών .....	26
<b>2</b>	<b>Περιγραφή εφαρμογής .....</b>	<b>29</b>
2.1	Σκοπός.....	29
2.2	Λειτουργίες .....	29
2.3	Δομικές μονάδες .....	30
2.3.1	Υλικό .....	30
2.3.1.1	Εισαγωγή στην πλατφόρμα Arduino .....	30
2.3.1.2	Αναλυτική παρουσίαση του Arduino Uno.....	37
2.3.1.3	Τηλεκατευθυνόμενο όχημα.....	44
2.3.1.4	Κάμερα.....	49
2.3.1.5	Modem router - access point.....	51
2.3.1.6	Προσωπικός υπολογιστής.....	51
2.3.2	Λογισμικό .....	52
<b>3</b>	<b>Μέθοδοι εντοπισμού αντικειμένων σε εσωτερικό χώρο .....</b>	<b>53</b>
3.1	Εισαγωγή.....	53
3.2	Ανίχνευση και παρακολούθηση αντικειμένων .....	55
3.3	Μια πρώτη προσέγγιση στην ανίχνευση χρώματος .....	58
3.4	Ανίχνευση μεγάλου δυαδικού αντικειμένου (Blob tracking) .....	62
<b>4</b>	<b>Αρχιτεκτονική και λεπτομέρειες υλοποίησης .....</b>	<b>70</b>
4.1	Αρχιτεκτονική του συστήματος.....	70
4.2	Λεπτομέρειες συνδεσμολογίας Hardware.....	72
4.3	Λεπτομέρειες και ανάλυση του κώδικα του Arduino Uno .....	74

4.3.1	Λογική υλοποίησης και Βιβλιοθήκες.....	74
4.3.2	Ανάλυση του κώδικα.....	74
4.4	Λεπτομέρειες και ανάλυση του Software του υπολογιστή .....	76
4.4.1	Αρχιτεκτονική της εφαρμογής.....	76
4.4.2	Ανάλυση των πακέτων και των κλάσεων τους.....	78
<b>5</b>	<b>Εγχειρίδιο χρήσης της εφαρμογής.....</b>	<b>83</b>
5.1	Οδηγίες εκκίνησης.....	83
5.2	Οδηγίες χρήσης.....	84
<b>6</b>	<b>Επίλογος.....</b>	<b>87</b>
6.1	Σύνοψη και συμπεράσματα.....	87
6.2	Μελλοντικές επεκτάσεις.....	88
6.2.1	Βελτίωση και αναβάθμιση του συστήματος πλοήγησης του οχήματος.....	88
6.2.2	Βελτίωση του συστήματος διεπαφής του χρήστη.....	88
<b>7</b>	<b>Βιβλιογραφία .....</b>	<b>90</b>

# 1

## *Εισαγωγή*

### *1.1 Η εμφάνιση των μικροελεγκτών*

Ένα από τα σημαντικότερα προβλήματα που παρουσιάστηκαν καθώς οι υπολογιστές εξελίσσονταν, ήταν το πώς θα χρησιμοποιηθούν σε όσο το δυνατόν περισσότερες καθημερινές εφαρμογές. Η «ευφυΐα» και η ευκολία με την οποία κάνουν πράξεις οι υπολογιστές θα μπορούσε να χρησιμοποιηθεί για να γίνουν πολύ πιο λειτουργικές και αποδοτικές πολλές συσκευές (ηλεκτρικές συσκευές, συσκευές ελέγχου μηχανών, συστήματα σηματοδότησης) Φυσικά, ήταν αδύνατο να δεσμευτεί ένας μεγάλος ή έστω ένας μεσαίων δυνατοτήτων υπολογιστής σ' αυτού του είδους τις εφαρμογές κυρίως λόγω του υπερβολικού τους μεγέθους και κόστους. Προέκυψε έτσι η απαίτηση για την κατασκευή ενός συστήματος που να περιέχει τα κυκλώματα όσο και τη βασική λογική έτσι ώστε να μπορεί να αντεπεξέλθει στο σύνολο των απαιτούμενων εφαρμογών . Συνεπώς, η λύση θα ήταν ένας ολοκληρωμένος αυτοδύναμος υπολογιστής μέσα σ' ένα μοναδικό ολοκληρωμένο κύκλωμα, ο οποίος θα έπρεπε να είχε τη δυνατότητα αποθήκευσης προγραμμάτων και εκτέλεσης σύνθετων μαθηματικών πράξεων. Βέβαια, όπως συνήθως γίνεται, οι απαιτήσεις της βιομηχανίας και οι δυνατότητες των σχεδιαστών υπολογιστών ήταν αδύνατο να γεφυρωθούν. Φυσικά ήταν δυνατό να δοθεί λύση σε μερικές από τις απαιτήσεις, αλλά η κατασκευή ενός υπολογιστή πάνω σ' ένα μοναδικό ολοκληρωμένο κύκλωμα φαινόταν σαν μία μακρινή επιδίωξη. Τη λύση φάνηκε να τη δίνει η τεχνολογία ολοκληρωμένων κυκλωμάτων, όταν μπόρεσε να τοποθετήσει χιλιάδες κυκλώματα πάνω σε μια πολύ μικρή επιφάνεια.

Το αποτέλεσμα της εμφάνισης της τεχνολογίας των ολοκληρωμένων κυκλωμάτων ήταν η ενσωμάτωση σε ένα μόνο ολοκληρωμένο κύκλωμα όλης της κεντρικής μονάδας επεξεργασίας, η οποία βέβαια θα έπρεπε να προγραμματίζεται για να περιέχει τις βασικότερες λειτουργίες ενός ψηφιακού υπολογιστή. Το κύκλωμα αυτό ονομάστηκε μικροεπεξεργαστής. Η μνήμη του βρίσκεται σε αρκετά ολοκληρωμένα κυκλώματα περιορισμένων αποθηκευτικών δυνατοτήτων, τα οποία το συνοδεύουν. Επίσης υποστηρίζεται και από μια πλειάδα α) ολοκληρωμένων κυκλωμάτων για να διασυνδέεται κατάλληλα και με τον εξωτερικό κόσμο μια και δεν έχει ενσωματωμένες αυτές τις δυνατότητες και β) ολοκληρωμένων κυκλωμάτων, που επιτελούν τις λειτουργίες χρονισμού και προώθησης δεδομένων στον τελικό τους προορισμό [5]. Η ανάπτυξη της τεχνολογίας των ολοκληρωμένων κυκλωμάτων τις τελευταίες δεκαετίες έδωσε τη δυνατότητα να μπορούν να ενσωματωθούν σε ένα ολοκληρωμένο κύκλωμα όλο και πιο πολύπλοκα κυκλώματα (από τον πρώτο μικροεπεξεργαστή, που είχε 2 χιλιάδες τρανζίστορ, έχουμε φτάσει πλέον σε επεξεργαστές με πάνω από 2 δισεκατομμύρια τρανζίστορ (ο εξαπύρηνος Intel Core i7-E περιέχει 2.600.000.000 τρανζίστορ) σε ένα και μόνο ολοκληρωμένο κύκλωμα) με αποτέλεσμα τη γρήγορη ανάπτυξη των μικροεπεξεργαστών και την ολοένα και πιο συχνή χρήση τους τόσο σε πολύπλοκες υπολογιστικές συσκευές όσο και σε απλές οικιακές συσκευές ή συστήματα ελέγχου.

Ένας μικροεπεξεργαστής περιλαμβάνει τις περισσότερες ή όλες τις λειτουργίες μιας κεντρικής μονάδας επεξεργασίας (ΚΜΕ) ενός ηλεκτρονικού υπολογιστή σε ένα ενιαίο ολοκληρωμένο κύκλωμα (IC). Η κατασκευή ενός τέτοιου υπολογιστικού συστήματος προϋποθέτει την εξωτερική διασύνδεση επιπρόσθετων λειτουργικών τμημάτων με τον μικροεπεξεργαστή, όπως η μνήμη, οι δίαυλοι μεταφοράς δεδομένων και άλλα περιφερειακά, γεγονός που καθιστά αδύνατη και μη συμφέρουσα εμπορικά τη χρήση του σε οικιακές ή βιομηχανικές συσκευές. Αυτό που θα διευκόλυνε το μηχανισμό ελέγχου και λειτουργίας τέτοιου είδους συσκευών είναι ένα ενσωματωμένο σύστημα αυτοματισμού, ικανό να αποθηκεύσει και να επεξεργαστεί πληροφορία. Γεννήθηκε λοιπόν η ανάγκη ύπαρξης μιας συσκευής, γνωστής και ως μικροελεγκτής που θα είναι υπεύθυνη για τον προγραμματισμό λειτουργίας αυτών των μηχανών και ικανή να εκτελεί απλές ή σύνθετες αριθμητικές και λογικές διαδικασίες [1].

Ο μικροελεγκτής είναι ένα αυτόνομο υπολογιστικό σύστημα, με πολύ μικρό μέγεθος, σε ένα και μοναδικό ολοκληρωμένο κύκλωμα (computer on a chip). Όπως και όλα τα VLSI κυκλώματα, αποτελείται από μέρη που κατασκευάζονται με διάφορες λιθογραφικές μεθόδους πάνω σε πλάκες πυριτίου, τα λεγόμενα Silicon Wafers. Πάνω σε αυτά σχηματίζονται χιλιάδες έως εκατομμύρια τρανζίστορ και κατ' επέκταση δημιουργούνται τα λεγόμενα ολοκληρωμένα

κύκλωματα που είναι συνδυασμός λογικών πυλών. Συνδυάζοντας τις λογικές πύλες, δημιουργούνται υπομονάδες που επιτελούν ορισμένες πιο εξειδικευμένες λειτουργίες στον μικροελεγκτή. Είναι ουσιαστικά ένας τύπος επεξεργαστή, μια παραλλαγή μικροεπεξεργαστή, ο οποίος μπορεί να λειτουργήσει με ελάχιστα εξωτερικά εξαρτήματα, λόγω των πολλών ενσωματωμένων υποσυστημάτων που διαθέτει. Μια κύρια διαφορά μεταξύ ενός μικροελεγκτή και ενός μικροεπεξεργαστή είναι πως στον μικροελεγκτή υπάρχει ενσωματωμένη μνήμη και μονάδες ελέγχου περιφερειακών συσκευών. Στον μικροεπεξεργαστή, το ολοκληρωμένο κύκλωμα που τον αποτελεί περιέχει μόνο την Λογική και Αριθμητική Μονάδα (ALU), στοιχειώδεις καταχωρητές (registers), προσωρινή μνήμη RAM πολύ υψηλής ταχύτητας (cache memory) και, κάποιες φορές, τον ελεγκτή μνήμης (memory controller). Ο μικροελεγκτής χρησιμοποιείται ευρύτατα σε όλα τα ενσωματωμένα συστήματα (embedded systems) ελέγχου χαμηλού και μεσαίου κόστους, όπως αυτά που χρησιμοποιούνται σε αυτοματισμούς, ηλεκτρονικά καταναλωτικά προϊόντα (από ψηφιακές φωτογραφικές μηχανές έως παιχνίδια), ηλεκτρικές συσκευές και κάθε είδους αυτοκινούμενα τροχοφόρα οχήματα. Για τη λειτουργία ενός πλήρους ενσωματωμένου υπολογιστικού συστήματος, απαιτούνται πολλά εξωτερικά υποσυστήματα και περιφερειακά [1]. Τέτοια είναι:

- Κύκλωμα συνδετικής λογικής (glue logic) για τη σύνδεση των εξωτερικών μνημών και άλλων περιφερειακών παράλληλης σύνδεσης στην αρτηρία δεδομένων (bus) του επεξεργαστή.
- Μνήμη προγράμματος (τύπου ROM, FLASH, EPROM κλπ) η οποία περιέχει το λογισμικό του συστήματος. Σε κάποια μοντέλα, είναι δυνατό το κλείδωμα αυτής της μνήμης, μετά την εγγραφή της, ώστε να προστατευτεί το περιεχόμενό της από αντιγραφή.
- Μεγάλο μέγεθος μνήμης RAM.
- Μόνιμη μνήμη αποθήκευσης παραμέτρων λειτουργίας (τύπου EEPROM ή NVRAM) η οποία να μπορεί να γράφεται τον πυρήνα του μικροελεγκτή. Αυτή η μνήμη έχει, έναντι της FLASH, το πλεονέκτημα της δυνατότητας διαγραφής και εγγραφής οποιουδήποτε μεμονωμένου byte.
- Κύκλωμα αρχικοποίησης (reset).
- Διαχειριστή αιτήσεων διακοπής (interrupt request controller) από τα περιφερειακά.
- Κύκλωμα επιτήρησης τροφοδοσίας (brown-out detection) το οποίο παρακολουθεί την τροφοδοσία και αρχικοποιεί ολόκληρο το σύστημα όταν αυτή πέσει κάτω από τα ανεκτά όρια, προλαμβάνοντας έτσι την αλλοίωση των δεδομένων.

- Κύκλωμα επιτήρησης λειτουργίας (watchdog timer) το οποίο αρχικοποιεί το σύστημα, αν αυτό εμφανίσει σημάδια δυσλειτουργίας λόγω κολλήματος (hang).
- Τοπικό ταλαντωτή για την παροχή παλμών χρονισμού (clock).
- Έναν ή περισσότερους χρονιστές-απαριθμητές υψηλής ταχύτητας (hardware timer-counter) για τη δημιουργία καθυστερήσεων, μέτρηση διάρκειας γεγονότων, απαρίθμηση γεγονότων και άλλων λειτουργιών ακριβούς χρονισμού.
- Ρολόι πραγματικού χρόνου (Real Time Clock, RTC) το οποίο τροφοδοτείται από ανεξάρτητη μπαταρία και γι αυτό πρέπει να έχει πολύ χαμηλή κατανάλωση ρεύματος.
- Σειρά ανεξάρτητων ψηφιακών εισόδων και εξόδων (Parallel Input-Output, PIO).

Γενικά, όλες οι οικογένειες μικροελεγκτών ενσωματώνουν τα περισσότερα από τα παραπάνω περιφερειακά, με διαφοροποιήσεις κυρίως στην ύπαρξη ή μη εσωτερικής μνήμης προγράμματος και στο είδος της [3]. Έτσι, υπάρχουν:

- Μικροελεγκτές χωρίς μνήμη προγράμματος, οι οποίοι χαρακτηρίζονται ως *ROM-less*. Αυτοί παρέχουν πάντοτε μια παράλληλη αρτηρία (bus) δεδομένων, πάνω στην οποία συνδέονται εξωτερικές μνήμες προγράμματος και RAM. Τέτοιοι τύποι μικροελεγκτών προορίζονται για πιο ισχυρά υπολογιστικά συστήματα ελέγχου, με μεγαλύτερες απαιτήσεις μνήμης.
- Μικροελεγκτές με μνήμη ROM, η οποία κατασκευάζεται με το λογισμικό της (Mask ROM) ή γράφεται μόνο μια φορά (One Time Programmable, OTP). Παρέχουν τη δυνατότητα πολύ χαμηλού κόστους, όταν αγοράζονται σε πολύ μεγάλες ποσότητες.
- Μικροελεγκτές με μνήμη FLASH, οι οποία μπορούν συνήθως να προγραμματιστεί πολλές φορές. Αυτή είναι η πιο διαδεδομένη κατηγορία. Συχνά ο προγραμματισμός της μνήμης μπορεί να γίνει ακόμη και πάνω στο κύκλωμα της ίδιας της ενσωματωμένης (embedded) εφαρμογής (δυνατότητα In Circuit Programming, ISP). Αυτοί οι μικροελεγκτές έχουν ουσιαστικά αντικαταστήσει τους παλαιότερους τύπους EPROM που έσβηναν με υπεριώδη ακτινοβολία (από το ειδικό τζαμάκι).

Ανάλογα με την εφαρμογή για την οποία προορίζεται ένας μικροελεγκτής, μπορεί να περιέχει και:

- Μία ή περισσότερες ασύγχρονες σειριακές θύρες επικοινωνίας (Universal Asynchronous Receiver Transmitter, UART).
- Σύγχρονες σειριακές θύρες επικοινωνίας (πχ I<sup>2</sup>C, SPI, Ethernet).

- Ολόκληρα υποσυστήματα για την άμεση υποστήριξη από υλικολογισμικό (firmware) των πιο σύνθετων πρωτοκόλλων επικοινωνίας όπως CAN, HDLC, ISDN, ADSL.
- Μονάδα άμεσης εκτέλεσης πράξεων κινητής υποδιαστολής (Floating Point Processing Unit, FPU), η οποία είναι πάντοτε πιο γρήγορη από την ALU του επεξεργαστή. Τέτοιες μονάδες χαρακτηρίζουν τους μικροελεγκτές με δυνατότητες ψηφιακής επεξεργασίας σήματος (Digital Signal Processing, DSP). Τα τελευταία χρόνια, με την ευρύτατη διάδοση των φορητών συσκευών ήχου και εικόνας, παρατηρείται μια τάση σύγκλισης των μικροελεγκτών με τους DSP.<sup>[4]</sup>
- Περισσότερες από μία εισόδους για μετατροπή αναλογικού σήματος σε ψηφιακό (Analog to Digital converter, ADC).
- Μετατροπέα ψηφιακού σε αναλογικό σήμα (Digital to Analog converter, DAC).
- Ελεγκτή οθόνης υγρών κρυστάλλων (Liquid Crystal Display, LCD).
- Υποσύστημα προγραμματισμού πάνω στο κύκλωμα (τύπου ISP, βλ. παραπάνω). Χάρη σε αυτό το κύκλωμα, είναι δυνατός ο επαναπρογραμματισμός (αναβάθμιση λογισμικού) της εφαρμογής, συνδέοντας στη συσκευή μια εξωτερική συσκευή προγραμματισμού (συνήθως σε θύρα UART RS-232) ή ακόμη και από το διαδίκτυο. Αυτή η δυνατότητα απαιτεί την προϋπαρξη λογισμικού υποδοχής (bootstrap) μέσα στη μνήμη προγράμματος και επομένως δεν μπορεί να γίνει σε τελείως άδεια μνήμη προγράμματος.
- Υποσύστημα προγραμματισμού (τύπου ISP) και διάγνωσης (συνήθως είναι το καθιερωμένο πρότυπο **JTAG**). Χάρη σε αυτό, είναι δυνατός ο προγραμματισμός της μνήμης προγράμματος χωρίς να προαπαιτείται κάποιο πρόγραμμα υποδοχής. Γι αυτό το λόγο, είναι ιδιαίτερα χρήσιμο στον αρχικό προγραμματισμό, πχ κατά τη συναρμολόγηση, ή σε περίπτωση σφάλματος (bug) στο λογισμικό υποδοχής το οποίο να καθιστά αδύνατη την κανονική αναβάθμιση.

Η γλώσσα που αντιλαμβάνεται ένας μικροελεγκτής όπως και ένας μικροεπεξεργαστής ονομάζεται γλώσσα μηχανής. Στην πράξη ο χρήστης προγραμματίζει τον μικροελεγκτή σε μνημονική γλώσσα Assembly. Η γλώσσα Assembly γενικά θεωρείται χαμηλού επιπέδου καθώς βρίσκεται πολύ κοντά στην γλώσσα μηχανής. Είναι δυνατόν να γράψουμε προγράμματα σε αυτή τη γλώσσα με την βοήθεια εργαλείων που ονομάζονται assemblers. Έτσι, όταν ο προγραμματιστής δώσει εντολές όπως την MOV, ADD, LD, (Μετακίνηση, Πρόσθεση, Φόρτωση) ο assembler αντιστοιχεί την εντολή με μια ακολουθία 0 και 1 που είναι κατανοητή από τον μικροελεγκτή και τα υποσυστήματα του. Ωστόσο, ακόμη και με αυτή την διευκόλυνση που προσφέρουν τα εργαλεία αυτά, είναι αρκετά οδυνηρό από άποψη χρόνου να γραφεί ένα πρόγραμμα τέτοιου επιπέδου. Οι εντολές της γλώσσας μηχανής επιτελούν περιορισμένες διεργασίες και πολλές φορές χρειάζεται εκατοντάδες εντολών για να

εκτελέσουν πράξεις και λειτουργίες πιο σύνθετες από τις συνηθισμένες. Υπάρχει όμως η δυνατότητα να γράψουμε το πρόγραμμα σε γλώσσα προγραμματισμού υψηλού επιπέδου.

Οι γλώσσες υψηλού επιπέδου μπορούν να μεταφραστούν σε γλώσσα χαμηλού επιπέδου με τη βοήθεια μεταφραστικών εργαλείων, όπως είναι ο compiler (μεταγλωττιστής) και ο interpreter (διερμηνέας). Το πρώτο είναι εργαλείο λογισμικού, το οποίο δέχεται τις εντολές υψηλού επιπέδου που δίνει ο χρήστης και τις μετατρέπει σε γλώσσα μηχανής. Ο διερμηνέας από την άλλη, είναι υλικό (hardware) μέσα στον μικροελεγκτή το οποίο αποκωδικοποιεί τη γλώσσα υψηλού επιπέδου άμεσα σε γλώσσα μηχανής. Αυτό απαιτεί βεβαίως επεξεργαστική ισχύ από το μέρος του μικροελεγκτή και έχει την τάση να τρέχει πιο αργά από ένα πρόγραμμα που «μεταγλωττίστηκε» (μέσω compiler) σε γλώσσα μηχανής. Ωστόσο έχει το πλεονέκτημα της αμεσότητας, καθώς ο προγραμματιστής μπορεί να αλλάξει ένα σημείο στον κώδικα του και να δει τα αποτελέσματα χωρίς το βήμα της μεταγλώττισης.

## **1.2 Εφαρμογές**

Οι μικροελεγκτές αποτελούν μέρος ενός εντυπωσιακού αριθμού προϊόντων τα οποία βρίσκονται γύρω μας. Το αυτοκίνητό μας, τα τηλεχειριστήριά μας, η τηλεόρασή μας, οι ψηφιακές κάμερες, τα κινητά τηλέφωνα, τα πλυντήριά μας είναι μερικά από αυτά. Στην ουσία δεν θα ήταν υπερβολή να πούμε ότι η χρήση μικροελεγκτών στις μέρες μας είναι καθολική και γενικά κάθε προϊόν το οποίο αλληλεπιδρά με ένα χρήστη περιλαμβάνει ένα μικροελεγκτή, ο οποίος παίζει το ρόλο του «εγκεφάλου» των ηλεκτρονικών κυκλωμάτων [4]. Δεν είναι τυχαίο πλέον, ότι πολλές βιομηχανίες προσανατολίζονται σε εφαρμογές όπου αποτελούνται από ηλεκτρικά κυκλώματα τα οποία ελέγχονται από μικροελεγκτή σε αντίθεση με την πρακτική του παρελθόντος όπου χρησιμοποιούσαν ογκώδεις ηλεκτρονικούς υπολογιστές (PC) ή ηλεκτρονόμους και σύνθετη αλλά μόνιμη λογική. Όλα τα παραπάνω είναι αποτέλεσμα της ανάπτυξης και του σχεδιασμού των ενσωματωμένων συστημάτων. Ο μικροελεγκτής ουσιαστικά αποτελεί ένα αυτόνομο ολοκληρωμένο σύστημα με επεξεργαστή, μνήμη και περιφερειακά και ως εκ τούτου μπορεί να χρησιμοποιηθεί ως ένα ενσωματωμένο σύστημα. Ενώ κάποια ενσωματωμένα συστήματα είναι αρκετά σύνθετα, τα περισσότερα σχεδιάζονται με ελάχιστη μνήμη και μέγεθος κώδικα, χωρίς την ύπαρξη λειτουργικού συστήματος και με λογισμικό χαμηλής πολυπλοκότητας. Ως συσκευές εισόδου – εξόδου συνήθως χρησιμοποιούνται διακόπτες, ηλεκτρικά χειριζόμενοι διακόπτες (ρελέ), πηνία, LEDs, οθόνες LCD, συσκευές ραδιοσυχνότητας και αισθητήρες θερμοκρασίας, υγρασίας, ισχύος φωτός κλπ. Τα ενσωματωμένα συστήματα συνήθως δεν διαθέτουν συσκευές εισόδου – εξόδου που συναντώνται στον προσωπικό υπολογιστή όπως πληκτρολόγιο, οθόνη, οπτικά μέσα



αποθήκευσης ή εκτυπωτές και τις περισσότερες φορές δεν παρέχουν δυνατότητα αλληλεπίδρασης με το χρήστη [2].

#### **Εφαρμογές στην καθημερινότητα**

- Συσκευές ελέγχου και αίσθησης φωτισμού
- Συσκευές ελέγχου και αίσθησης θερμοκρασίας
- Συσκευές ασφαλείας και ανίχνευσης φωτιάς
- Συσκευές ελέγχου διαδικασιών

#### **Εφαρμογές σε συσκευές βιομηχανικού ελέγχου**

- Βιομηχανικές συσκευές οργάνων μέτρησης
- Συσκευές ελέγχου διαδικασιών

#### **Εφαρμογές σε μετρητικές συσκευές**

- Μέτρηση τάσης και ρεύματος
- Μετρήσεις σε περιστρεφόμενα συστήματα/αντικείμενα
- Φορητές μετρητικές συσκευές

### ***1.3 Δυνατότητες – Περιορισμοί***

Στους σύγχρονους μικροεπεξεργαστές για μη ενσωματωμένα συστήματα (πχ τους μικροεπεξεργαστές των προσωπικών υπολογιστών), δίνεται έμφαση στην υπολογιστική ισχύ. Η ευελιξία ανάπτυξης διαφορετικών εφαρμογών είναι μεγάλη, καθώς η λειτουργικότητα του τελικού συστήματος καθορίζεται από τα εξωτερικά περιφερειακά τα οποία διασυνδέονται με την κεντρική μονάδα (μικροεπεξεργαστή), η οποία δεν είναι εξειδικευμένη. Αντίθετα, στους μικροεπεξεργαστές για ενσωματωμένα συστήματα (μικροελεγκτές), οι οποίοι έχουν μικρότερες ή και μηδαμινές δυνατότητες συνεργασίας με εξωτερικά περιφερειακά, αυτού του είδους, η ευελιξία είναι περιορισμένη, καθώς και η υπολογιστική ισχύς. Οι μικροελεγκτές δίνουν έμφαση στο μικρό αριθμό ολοκληρωμένων κυκλωμάτων που απαιτείται για τη λειτουργία μιας συσκευής, το χαμηλό κόστος και την εξειδίκευση [5].

Αναλυτικά, τα πλεονεκτήματα των μικροελεγκτών είναι:

- Αυτονομία, μέσω της ενσωμάτωσης σύνθετων περιφερειακών υποσυστημάτων όπως μνήμες και θύρες επικοινωνίας. Έτσι πολλοί μικροελεγκτές δεν χρειάζονται κανένα άλλο ολοκληρωμένο κύκλωμα για να λειτουργήσουν.
- Η ενσωμάτωση περιφερειακών σημαίνει ευκολότερη υλοποίηση εφαρμογών λόγω των απλούστερων διασυνδέσεων. Επίσης, οδηγεί σε χαμηλότερη κατανάλωση ισχύος, μεγιστοποιώντας τη φορητότητα και ελαχιστοποιεί το κόστος της συσκευής στην οποία ενσωματώνεται ο μικροελεγκτής.
- Χαμηλό κόστος.
- Μεγαλύτερη αξιοπιστία, και πάλι λόγω των λιγότερων διασυνδέσεων.
- Μειωμένες εκπομπές ηλεκτρομαγνητικών παρεμβολών και μειωμένη ευαισθησία σε αντίστοιχες παρεμβολές από άλλες ηλεκτρικές και ηλεκτρονικές συσκευές. Το πλεονέκτημα αυτό προκύπτει από το μικρότερο αριθμό και μήκος εξωτερικών διασυνδέσεων καθώς και τις χαμηλότερες ταχύτητες λειτουργίας.
- Περισσότεροι διαθέσιμοι ακροδέκτες για ψηφιακές εισόδους-εξόδους (για δεδομένο μέγεθος ολοκληρωμένου κυκλώματος), λόγω της μη δέσμευσής τους για τη σύνδεση εξωτερικών περιφερειακών.
- Μικρό μέγεθος συνολικού υπολογιστικού συστήματος.

## 1.4 Ιστορική εξέλιξη

### Η Εξέλιξη των Μικροεπεξεργαστών

#### 4 bit Μικροεπεξεργαστές

Θεωρείται ότι ο Intel 4004 είναι ο πρώτος μικροεπεξεργαστής, αν και στις αρχές τις δεκαετίας του 1970 τρεις μικροεπεξεργαστές έκαναν την εμφάνιση τους, κατασκευασμένοι από τρεις διαφορετικές εταιρείες [6]. Οι μικροεπεξεργαστές αυτοί ήταν ο 4004 της Intel, ο TMS 1000 της Texas Instruments (TI) και ο Central Air Data Computer (CADC) της Garrett AiResearch. Ο Intel 4004 παρουσιάστηκε το 1971 από τον Ted Hoff και το συνεργάτη του Stan Mazor, μηχανικούς της Intel, στους οποίους ανατέθηκε η ανάπτυξή του. Η Ιαπωνική εταιρεία Busicom είχε ζητήσει από την Intel την ανάπτυξη του για χρήση σε αριθμομηχανές. Ο Intel 4004 ήταν ένας 4bit επεξεργαστής (ο μικροεπεξεργαστής λαμβάνει 4 bit από την μνήμη κάθε φορά με σκοπό να τα επεξεργαστεί), που αποτελούταν από περίπου 2300 τρανζίστορες με συχνότητα ρολογιού 108 KHz. Επιπλέον, εκτελούσε 60000 πράξεις το δευτερόλεπτο και μπορούσε να δει 640 bytes μνήμης. Επιπλέον, το 1971, η Intel ανακοίνωσε

τον πρώτο μικροϋπολογιστή, το σύστημα MCS-4, στο οποίο χρησιμοποιήθηκε ο 4004, το 4001 ROM chip, το 4002 RAM chip και το 4003 shift register chip για σειριακή επικοινωνία. Ο 4004 ήταν ένα τεχνολογικό αποκορύφωμα για την εποχή του παρότι ήταν πολύ περιορισμένων δυνατοτήτων. Σχεδόν ταυτόχρονα το 1971, το The Smithsonian Institution ισχυρίζονταν ότι οι μηχανικοί Gary Boone και Michael Cochran είχαν επίσης ολοκληρώσει τον πρώτο μικροεπεξεργαστή, ο οποίος ονομάστηκε TMS 1000 και διατέθηκε στην αγορά το 1974. Λίγα χρόνια πιο πριν, το 1968, ο Ray Holt, απόφοιτος του California Polytechnical University, είχε φτιάξει τον F14 CADC. Αυτό όμως έγινε γνωστό 30 χρόνια αργότερα, το 1998, όταν το Αμερικανικό Ναυτικό (US Navy) έκανε γνωστά δημοσίως έγγραφα που αποδείκνυαν τη χρήση του μικροεπεξεργαστή σε πολεμικά αεροπλάνα. Ωστόσο επιστημονικές δημοσιεύσεις και βιβλιογραφία αναφέρουν ότι ο MP944 ήταν ο πρώτος μικροεπεξεργαστής, ο οποίος χρησιμοποιούνταν στο F-14 Tomcat πολεμικό αεροσκάφος, αλλά δεν αποτελούνταν από ένα μοναδικό chip και δεν ήταν γενικής χρήσης.

#### 8 bit Μικροεπεξεργαστές

Τους 4 bits επεξεργαστές ακολούθησαν οι 8 bits, με τον σημαντικότερο από αυτούς τον Intel 8008 ο οποίος ήταν ο πρώτος εμπορικός 8 bit μικροεπεξεργαστής. Η ανάπτυξη του ξεκίνησε το 1971 με επικεφαλής τον Federico Faggin και ολοκληρώθηκε το 1972, αφού ξανασχεδιάστηκε καθώς η αρχική σχεδίαση είχε διαρροές ηλεκτρικού φορτίου από τις συσκευές μνήμης. Η συχνότητα ρολογιού ήταν στα 200 KHz, ενώ το chip χρησιμοποιούσε 3500 κρυσταλλολυχνίες και μπορούσε να δει 16 Kbytes μνήμης. Χρησιμοποιήθηκε στον Micral το 1973, ο οποίος ήταν ο πιο σύγχρονος υπολογιστής που τροφοδοτήθηκε από έναν μικροεπεξεργαστή 8008 ενώ ένα χρόνο αργότερα η Scelbi χρησιμοποιεί τον 8008 στον υπολογιστή 8H. Με τον 8008 ξέσπασε μεγάλο ενδιαφέρον για την ανάπτυξη μικροεπεξεργαστών το οποίο είχε αποτέλεσμα να αυξηθούν οι απαιτήσεις στην ταχύτητα, επικοινωνία με το περιβάλλον, και πιο πολλές εντολές και εισόδους δεδομένων.

Το 1974 ακολούθησε η κυκλοφορία του Intel 8080, ο οποίος έτρεχε στα 2 MHz, μπορούσε να δει 64 Kbytes μνήμης και περιείχε 6.000 τρανζίστορ. Σημαντικά γεγονότα που ακολούθησαν ήταν η ανάπτυξη του λειτουργικού συστήματος CP/M για τον Intel 8080 από τον Gary Kildall της εταιρίας Microcomputer Applications Associates, η σχεδίαση του υπολογιστή Altair 8800 ο οποίος χρησιμοποιούσε Intel 8080 με 256 bytes RAM και η ανάπτυξη της γλώσσας Microsoft Basic, από τον Bill Gates και τον Paul Allen, για τους μικροεπεξεργαστές της Intel. Τα γεγονότα αυτά θεωρούνται από πολλούς ότι οδήγησαν στη ραγδαία ανάπτυξη των προσωπικών υπολογιστών.

Σύντομα μετά τη κυκλοφορία του 8080, η Motorola κυκλοφόρησε τον 6800, 8 bit επεξεργαστή. Είχε 4000 τρανζίστορ, 78 εντολές, σήμα χρονισμού στα 1 ή 2 MHz με 16 bit πλάτος διαύλου διευθύνσεων και ήταν ένας από τους πρώτους μικροεπεξεργαστές με καταχωρητή δείκτη (index register). Ο 6800 ήταν καλοσχεδιασμένος, παρόλα τα προβλήματα με την παραγωγή και γι' αυτό και χρησιμοποιήθηκε σε πολλά συστήματα. Έτσι η εταιρία MITS ξεκίνησε την σχεδίαση ενός Altair βασισμένου στον Motorola 6800 και λίγο αργότερα παρουσιάστηκε ο υπολογιστής Sphere I, με επεξεργαστή τον Motorola 6800, 4Kbytes RAM, πρόγραμμα ROM monitor, πληκτρολόγιο και διασύνδεση βίντεο. Προς τα τέλη του 1970 η Tektronix σχεδίασε τον 4051 βασισμένο στον επεξεργαστή της Motorola με 32 Kbytes RAM πληκτρολόγιο και high speed data cartridge. Ο 4051 προοριζόταν για χρήστες που προγραμμάτιζαν σε BASIC.

Τον 6800 ακολούθησε ο 6809, ένα από τα πιο ισχυρά σχέδια μικροεπεξεργαστή και επίσης ένα από τον πιο σύνθετα σχέδια λογικής που έγιναν ποτέ στην παραγωγή οποιουδήποτε μικροεπεξεργαστή. Αρχικά χρησιμοποιούσε σήμα χρονισμού στο 1 MHz στο μοντέλο 68A09, ύστερα στο 1.5 MHz στο μοντέλο 68A09 και στα 2 MHz στο 68B09. Ο 6809 χρησιμοποιήθηκε, μεταξύ άλλων συστημάτων και στη κονσόλα παιχνιδιών Vectrex.

Το 1975, η ανταγωνίστρια εταιρεία MOS Technology κυκλοφορεί τον επεξεργαστή 6502, μια παραλλαγή του 6800, ο οποίος χρησιμοποιούσε δυο 8 bit καταχωρητές. Είχε 5000 τρανζίστορ, 56 εντολές, σήμα χρονισμού αρχικά 20 KHz μέχρι 4 MHz με 16 bit πλάτος διαύλου διευθύνσεων. Χρησιμοποιήθηκε σε πολλά συστήματα, μερικά από αυτά οι κονσόλες ηλεκτρονικών παιχνιδιών Atari 2600, Nintendo Entertainment System/NES και οι πρώτοι οικιακοί υπολογιστές Commodore 64 και Apple II.

Το 1976 φτιάχνεται ο Z80 από την εταιρεία Zilog. Επίσης 8bit μικροεπεξεργαστής, βασισμένος στον 8080, του οποίου η γλώσσα μηχανής είναι υπερσύνολο αυτής του Intel 8080. Είχε σήμα χρονισμού στα 3.5 MHz με 16 bit πλάτος διαύλου διευθύνσεων, ενώ μπορούσε να δει 64 Kbytes μνήμης. Το χαμηλό κόστος, η μικρή συσκευασία καθώς και άλλα χαρακτηριστικά τον έκαναν πολύ δημοφιλή στη δεκαετία του 80.

Τέλος, το 1982, ήδη στην αρχή της εποχής των 16 bit επεξεργαστών κυκλοφορεί ο RCA(CDP)1802 της RCA και χρησιμοποιήθηκε για την κατασκευή των δορυφόρων Voyager,

Viking και του διαστημοπλοίου Γαλιλαίος. Είχε πολύ μικρή κατανάλωση και ήταν ανθεκτικός στην κοσμική ακτινοβολία και τις ηλεκτροστατικές αποφορτίσεις.

### 16 bit Μικροεπεξεργαστές

Ο πρώτος 16bit μικροεπεξεργαστής multi-chip ήταν ο IMP 16 της National, που εισήχθη στις αρχές του 1973. Το 1974 εισήχθη η 8 bit έκδοσή του IMP-8. Δύο χρόνια αργότερα, η National εισήγαγε το πρώτο 16bit single-chip μικροεπεξεργαστή, τον PACE, ο οποίος ακολουθήθηκε αργότερα από μια NMOS έκδοση, το INS8900.

Το 1976, εμφανίζεται ο TMS 9900 της Texas Instruments, ένας από τους πρώτους καθαρά 16bit μικροεπεξεργαστές. Ο TMS 9900 δεν είχε καθόλου εσωτερικούς καταχωρητές, εκτός από έναν που όριζε την θέση των καταχωρητών του στην RAM, όπου αποθηκεύονται. Η σχεδίαση επέτρεπε την ταχύτατη αλλαγή context, αφού για να αλλάξουν όλοι οι καταχωρητές και να κληθεί μια συνάρτηση, πρέπει να αλλάξει ο μοναδικός εσωτερικός καταχωρητής. Η συγκεκριμένη σχεδίαση είχε νόημα για την εποχή της, διότι η εσωτερική μνήμη ήταν πιο αργή από την εξωτερική.

Η Intel επανέρχεται στο προσκήνιο αναβαθμίζοντας το σχέδιο του 8080 στον 16bit Intel 8086, το πρώτο μέλος της x86 οικογένειας που χρησιμοποιούν οι περισσότεροι σύγχρονοι υπολογιστές. Ο 8086 είχε 10 φορές καλύτερη απόδοση από τον 8080. Η Intel εισήγαγε τον 8086 ως οικονομικώς αποδοτικό τρόπο μεταφοράς του λογισμικού από τον 8080, και πέτυχε κερδίζοντας την εμπιστοσύνη πολλών επιχειρήσεων με εκείνη την προϋπόθεση. Ο 8086 είχε 29.000 τρανζίστορ, ταχύτητα λειτουργίας στα 10 MHz, ενώ χρησιμοποιούσε καταχωρητές των 16 bit και δίαυλο δεδομένων των 16 bit. Επιπλέον, μπορούσε να δει 1 Mbyte μνήμης. Τον Ιούνιο του 1979, αποκαλύφθηκε ο μικροεπεξεργαστής 8088, που ήταν μια παραλλαγή του 8086. Πρόκειται για έναν 16bit επεξεργαστή εσωτερικά, του οποίου ο εξωτερικός δίαυλος δεδομένων ήταν των 8 bits. Η σχεδίαση αυτή είχε σκοπό τη χρήση των υπαρχόντων 8bit controller chips για συσκευές. Ο 8088 περιείχε 29000 τρανζίστορς και μπορούσε να δει 1 Mbyte μνήμης. Μετά τον 8088 η Intel απελευθέρωσε τους 16bit μικροεπεξεργαστές 80186 και 80286, παγιώνοντας την κυριαρχία της στην αγορά προσωπικών υπολογιστών. Ο 80286 παρουσιάστηκε το 1982, και είχε συχνότητα λειτουργίας αρχικά στα 6 MHz και έπειτα στα 12 MHz. Ο ιστορικός αυτός μικροεπεξεργαστής ανήκε στην οικογένεια x86 και περιελάμβανε δίαυλο δεδομένων 16 bit, δίαυλο διευθύνσεων 24 bit. Επιπλέον, μπορούσε να δει μέχρι 16 MBytes μνήμης και περιείχε 130000 τρανζίστορ. Ο 80286 αποτελεί τον πρώτο μικροεπεξεργαστή που είχε τη δυνατότητα να λειτουργεί στην κατάσταση Protected Mode (προστατευμένη κατάσταση λειτουργίας).

### 32 bit Μικροεπεξεργαστές

Πολύ σύντομα μετά τη κυκλοφορία των 16 bit επεξεργαστών εμφανίστηκαν οι 32 bit επεξεργαστές.

Το 1979 εισήχθη ο MC68000, Ο διασημότερος 32bit μικροεπεξεργαστής και το πρώτο μέλος της οικογένειας m68k. Είχε 32bit καταχωρητές αλλά χρησιμοποίησε 16bit διαδρομές δεδομένων, καθώς και έναν 16bit εξωτερικό διάυλο δεδομένων. Η Motorola το περιέγραψε γενικά ως 16bit επεξεργαστή, αν και είχε 32bit αρχιτεκτονική. Ο συνδυασμός της υψηλής ταχύτητας, του μεγάλου χώρου αποθήκευσης (16 Mbyte) και του αρκετά χαμηλού κόστους τον έκανε τον δημοφιλέστερο μικροεπεξεργαστή της κατηγορίας του, με αποτέλεσμα να χρησιμοποιηθεί στους υπολογιστές Apple Lisa και η Macintosh. Τον MC68000 ακολούθησε ο MC68010, ο οποίος πρόσθεσε την υποστήριξη της εικονικής μνήμης ενώ το 1985 ακολούθησε ο MC68020. Είχε 200000 τρανζίστορ και συχνότητα λειτουργίας στα 16 MHz. Τα 68020 έγιναν ιδιαίτερα δημοφιλή στη super microcomputer Unix αγορά, ενώ πολλές μικρές επιχειρήσεις, όπως η Altos, παρήγαγαν τα συστήματα desktop. Έπειτα, ακολούθησε ο MC68030, ο οποίος πρόσθεσε τη μονάδα διαχείρισης μνήμης (MMU) στο τσιπ, ο 68040, ο οποίος περιέλαβε τη μονάδα υπολογισμού κινητής υποδιαστολής (FPU) για καλύτερη απόδοση ενώ ο 68050 δεν κατάφερε να επιτύχει τους στόχους απόδοσής του και έτσι δεν απελευθερώθηκε. Το ακόλουθο MC68060 βγήκε στην αγορά την ίδια περίοδο που κυκλοφορούσαν και τα πολύ γρηγορότερα σχέδια RISC όπως οι MIPS R2000 (1984) και R3000 (1989) που ήταν πολύ πετυχημένοι 32-bit RISC επεξεργαστές. Χρησιμοποιήθηκαν σε high-end σταθμούς εργασίας και servers από την SGI, μεταξύ άλλων. Η οικογένεια m68k εξασθένησε από την αγορά υπολογιστή γραφείου στις αρχές της δεκαετίας του 90.

Ο παγκόσμιος πρώτος single-chip 32bit μικροεπεξεργαστής ήταν ο BELLMAC 32A της AT&T Bell Labs, ο οποίος παρουσιάστηκε το 1980 και κυκλοφόρησε το 1982. Μετά από την αποστέριση του AT&T το 1984 και την αλλαγή της επωνυμίας της εταιρίας, ο BELLMAC 32A μετονομάστηκε σε WE 32000 (WE είναι τα αρχικά της Western Electric). Η επόμενη γενιά μικροεπεξεργαστών που στηρίχτηκε στον WE 32000, είναι ο WE 32100 και ο WE 32200. Αυτοί οι μικροεπεξεργαστές χρησιμοποιήθηκαν στους μίνι-υπολογιστές AT&T 3B5, 3B15 στο 3B2, τον πρώτο super microcomputer γραφείου. Όλα αυτά τα συστήματα έτρεξαν το αρχικό λειτουργικό σύστημα Unix των Bell Labs.

Ο πρώτος 32bit μικροεπεξεργαστής της Intel ήταν το iAPX 432, ο οποίος εισήχθη το 1981 αλλά δεν ήταν μια εμπορική επιτυχία. Παρόλο που είχε μια προηγμένη αντικειμενοστραφή

αρχιτεκτονική, είχε κακή απόδοση σε σχέση με άλλες ανταγωνιστικές αρχιτεκτονικές όπως το Motorola 68000.

Ένα άλλο ενδιαφέρον σχέδιο ήταν ο Zilog Z8000, ο οποίος εισήχθη πολύ αργά στην αγορά και εξαφανίστηκε από αυτή πολύ γρήγορα.

Από το 1985 έως το 2003, η 32-bit x86 αρχιτεκτονική έγινε η κυρίαρχη αρχιτεκτονική στην αγορά desktop, laptop, και server και οι επεξεργαστές έγιναν γρηγορότεροι και πιο ισχυροί. Η σειρά Pentium της Intel είναι πιθανότατα η πιο αναγνωρίσιμη σειρά 32 bit επεξεργαστών.

Πριν τη σειρά Pentium προηγήθηκαν αρκετές σειρές επεξεργαστών. Ξεκινώντας τον Οκτώβριο του 1985, η Intel παρουσιάζει τον απόγονο του 80286 τον μικροεπεξεργαστή 80386. Είχε συχνότητα λειτουργίας αρχικά στα 16 MHz. και χρησιμοποιούσε καθαρούς 32bit καταχωρητές και 32bit διαύλους δεδομένων και διευθύνσεων. Περιείχε 275000 τρανζίστορ, υποστήριζε πολυδιεργασία (multitasking) εικονικής μνήμης με δυνατότητα σελιδοποίησης (paging). Το 1989, εμφανίζεται ο μικροεπεξεργαστής Intel 80486, ο οποίος είχε 1200000 τρανζίστορ και συχνότητα λειτουργίας 50 MHz. Τη δεκαετία του 1990 έκαναν την εμφάνισή τους οι μικροεπεξεργαστές Intel Pentium, οι οποίοι αποτελούσαν τη συνέχεια του 80486 ενώ είχαν υπερβαθωπή (superscalar) αρχιτεκτονική και 32bit δίαυλο δεδομένων. Το 1993 εμφανίζεται ο Intel Pentium της οικογένειας P5, ο οποίος περιείχε 3100000 τρανζίστορ και λειτουργούσε στα 60 και 66 MHz. Το 1995, η Intel παρουσιάζει τον Pentium Pro, τον πρώτο στην οικογένεια των P6. Είχε 5500000 τρανζίστορ και ανήκε στην έκτη γενιά των επεξεργαστών της οικογένειας x86. Δύο χρόνια αργότερα, η Intel εισάγει τον μικροεπεξεργαστή Pentium II, έναν Pentium Pro με τεχνολογία MMX (MMX εντολές) για την υποστήριξη πολυμέσων. Ο Pentium II είχε 7500000 τρανζίστορ και η συχνότητα λειτουργίας του βρισκόταν στα 300 MHz. Το 1999, ακολούθησε ο Pentium III με 9500000 τρανζίστορ και συχνότητα λειτουργίας στα 450 MHz. Την επόμενη χρονιά, εμφανίστηκε ο Pentium IV ο οποίος ήταν σχεδιασμένος σύμφωνα με την μικροαρχιτεκτονική NetBurst, η οποία συνεχίζει να αποτελεί την τεχνολογική καρδιά του Pentium IV και των παραλλαγών του κυκλοφόρησαν αργότερα. Ο Pentium D ήταν ο τελευταίος μικροεπεξεργαστής της σειράς Pentium, η οποία σταμάτησε να κυκλοφορεί το 2008. Λόγω του μεγάλου κόστους παραγωγής του Pentium II η Intel ανεπτυξε και κυκλοφόρησε μια νέα σειρά, τη σειρά Celeron. Ο Celeron είχε πιο προσιτή τιμή, αλλά δεν μπορούσε να λειτουργήσει σε πολύ υψηλές συχνότητες.

Η ανταγωνίστρια της Intel, AMD μπήκε δυναμικά στο χώρο το 1997. Αρκετά αργότερα σε σχέση με την Intel αφού η Intel είχε ήδη πάρει τα δικαιώματα της αρχιτεκτονικής x86 οπότε όλες οι άλλες εταιρείες έπρεπε να απαντήσουν τα δικά τους σχέδια. Η αρχή έγινε με τους επεξεργαστές της σειράς K6, οι οποίοι ήταν εφάμιλλοι αυτών της Intel σε τιμή και επιδόσεις. Το 1999, η AMD προώθησε την καινούργια οικογένεια μικροεπεξεργαστών, Athlon. Ο Athlon Classic, που αποτελεί τον πρώτο επεξεργαστή της σειράς και μεγάλο ανταγωνιστή των Pentium, εισήγαγε την έβδομη γενιά επεξεργαστών της οικογένειας x86.

#### 64 bit Μικροεπεξεργαστές

Αν και οι πρώτοι 64bit επεξεργαστές εμφανίστηκαν στις αρχές του 1990, άρχισαν να εφαρμόζονται στους υπολογιστές γραφείου το 2003. Μέχρι το 2003, οι 64bit επεξεργαστές απευθύνονταν αποκλειστικά στην αγορά των ακριβών σταθμών εργασίας και των διακομιστών. Τον Σεπτέμβριο του 2003 η AMD εισήγαγε “την καλύτερη καινοτομία στους επεξεργαστές για το έτος 2003”, τον Athlon 64 για να ακολουθήσει η Intel με τον Intel 64. Και οι 2 επεξεργαστές μπορούσαν να τρέξουν 32 και 64bit εντολές. Με τη κυκλοφορία των επεξεργαστών αυτών, η γνώμη του ευρύτερου χώρου ήταν ότι θα έβρισκαν εφαρμογή μόνο σε ακαδημαϊκά ή ερευνητικά προγράμματα καθώς εκεί χρειαζόνταν μεγάλοι υπολογισμοί, γρήγορες προσβάσεις σε μεγάλες βάσεις δεδομένων αλλά και επίλυση σύνθετων προβλημάτων που μπορούσαν να προσφέρουν οι 64bit επεξεργαστές.

Ο Athlon 64 αποτελεί διαφοροποίηση της αρχιτεκτονικής AMD64 για να μειωθεί το κόστος του. Το 2005, η AMD ανακοίνωσε τους διπλοπύρηνους επεξεργαστές Opteron για servers και workstations, καθώς και τους διπλοπύρηνους επεξεργαστές Athlon 64 για προσωπικούς υπολογιστές. Τον Φεβρουάριο του 2009, η AMD παρουσίασε τον τετραπύρηνο επεξεργαστή Phenom II με το κωδικό όνομα Deneb. Οι πιο πρόσφατοι επεξεργαστές της AMD ανακοινώθηκαν τον Ιούλιο. Συγκεκριμένα η εταιρία ενημέρωσε για τη διάθεση πέντε νέων εξαπύρηνων επεξεργαστών Istanbul, με πολύ χαμηλή κατανάλωση ενέργειας. Όσον αφορά στην απόδοση, τα καινούργια μοντέλα δεν καταφέρνουν να φτάσουν τους Xeon E5504 και E5520 της Intel.

Το 2006, η Intel αλλάζει τα δεδομένα στην απόδοσης και κατανάλωσης των υπολογιστών ανακοινώνοντας τις αρχιτεκτονικές Intel Core 2 Duo και Intel Core 2 Extreme. Οι αρχιτεκτονικές αυτές μπορούν να χρησιμοποιηθούν σε επεξεργαστές για σταθερούς, φορητούς και servers και παρέχουν ισχυρή απόδοση με παράλληλη εξοικονόμηση ενέργειας. Η χρήση των 2 ή 4 πυρήνων διευκολύνει τη ομαλότερη και γρηγορότερη εκτέλεση περισσότερων διεργασιών. Το 2008, κυκλοφόρησε ο Atom, ο μικρότερος σε μέγεθος



επεξεργαστής της Intel που υλοποιήθηκε με τα μικρότερα τρανζίστορ του κόσμου. Δημιουργήθηκε ως μία εντελώς νέα σχεδίαση, ειδικά για φθηνές συσκευές, όπως πολύ μικρά notebooks και φορητές συσκευές με πρόσβαση στο Internet. Για τη σχεδίαση του Atom χρησιμοποιήθηκε η μικροαρχιτεκτονική Core, η ίδια δηλαδή τεχνολογία με την οποία η Intel κατασκευάζει τους γνωστούς Core 2 Duo επεξεργαστές για επιτραπέζιους και φορητούς υπολογιστές. Την ίδια χρονιά η Intel ανακοίνωσε τη νέα σειρά επεξεργαστών της με το όνομα Core i7.

### Η εξέλιξη των μικροελεγκτών

Όπως αναφέρθηκε και παραπάνω ο πρώτος μικροεπεξεργαστής ήταν ο Intel 4004 βασισμένος σε αρχιτεκτονική 4 bit που κυκλοφόρησε το 1971. Ο Intel 8008 όπως και άλλοι ισχυρότεροι μικροεπεξεργαστές έγιναν διαθέσιμοι τα επόμενα χρόνια. Προκειμένου όμως να υλοποιηθεί ένα σύστημα εργασίας οι παραπάνω επεξεργαστές έπρεπε να συνδεθούν με πολλαπλά εξωτερικά ολοκληρωμένα κυκλώματα, αυξάνοντας έτσι το κόστος κατασκευής και καθιστώντας ουσιαστικά αδύνατον τη ενσωμάτωσή τους σε ηλεκτρικές συσκευές οικιακής ή βιομηχανικής χρήσεως.

Βάσει αναφορών του Ιδρύματος Σμιθσόνιαν (Smithsonian Institution) ο πρώτος μικροελεγκτής κατασκευάστηκε από τους μηχανικούς της Texas Instruments Gary Boone και Michael Cochran το 1971. Το αποτέλεσμα της δουλειάς τους ήταν ο TMS 1000 που έγινε εμπορικά διαθέσιμος το 1974. Στο ολοκληρωμένο του κύκλωμα είχαν υλοποιηθεί μνήμη μόνο ανάγνωσης (read-only memory), μνήμη ανάγνωσης-εγγραφής, ο επεξεργαστής και το ρολόι χρονισμού. Ο στόχος ήταν να χρησιμοποιηθεί σε ενσωματωμένα συστήματα.

Η απάντηση ήρθε από την Intel με την κατασκευή του Intel 8048, ένα σύστημα υπολογιστή υλοποιημένο σε ένα μόνο ολοκληρωμένο κύκλωμα που προοριζόταν για χρήση σε συσκευές ελέγχου και κυκλοφόρησε το 1977. Συνδυάζε μνήμη τυχαίας προσπέλασης (RAM) και μνήμη μόνο ανάγνωσης (ROM) στο ίδιο ολοκληρωμένο κύκλωμα. Ο Intel 8048 χρησιμοποιήθηκε ευρέως στην κατασκευή πληκτρολογίων για προσωπικούς υπολογιστές και σε πολυάριθμες άλλες οικιακές ηλεκτρικές συσκευές. Εκείνη τη στιγμή ήταν που ο Luke J. Valenter πρόεδρος της Intel δήλωσε ότι ο μικροελεγκτής ήταν μια τεράστια επιτυχία στην ιστορία της εταιρείας και αύξησε τον προϋπολογισμό του εν λόγω τμήματος κατά 25%.

Οι περισσότεροι μικροελεγκτές την εποχή αυτή έρχονταν σε δύο παραλλαγές. Ο ένας τύπος διέθετε επαναπρογραμματιζόμενη μνήμη (EPROM) με ένα διαφανές παράθυρο από χαλαζία στο καπάκι της συσκευασίας που επέτρεπε τη διαγραφή των περιεχομένων της με έκθεση σε

υπεριώδες φως. Ο δεύτερος τύπος ήταν μια παραλλαγή μνήμης PROM που μπορούσε να προγραμματιστεί μόνο μία φορά (πολλές φορές αυτό δηλώνεται με τον όρο OTP (one time programmable) ). Η PROM ήταν ουσιαστικά η ίδια μνήμη με την EPROM χωρίς όμως τη δυνατότητα να εκτεθεί σε υπεριώδη ακτινοβολία και να διαγραφούν τα περιεχόμενα της. Στις εκδόσεις με EPROM η συσκευασία ήταν από κεραμικό υλικό και σε συνδυασμό με το παράθυρο χαλαζία το κόστος κατασκευής ήταν αρκετά μεγαλύτερο από τις εκδόσεις με PROM που μπορούσαν να υλοποιηθούν σε αδιαφανείς πλαστικές συσκευασίες.

Το 1993 η εμφάνιση της ηλεκτρικά επαναπρογραμματιζόμενης μνήμης (EEPROM) επέτρεψε στους μικροελεγκτές (ξεκινώντας από τον PIC16x84) να διαγράφουν ηλεκτρικά και αρκετά γρήγορα το περιεχόμενο της μνήμης απαλλάσσοντας τους από την ακριβή κεραμική συσκευασία με το παράθυρο χαλαζία. Αυτό είχε ως αποτέλεσμα να καταστούν δυνατά η γρήγορη δημιουργία πρωτοτύπων και ο προγραμματισμός του μικροελεγκτή αφού ενσωματωθεί στο σύστημα (In System Programming). (Η μνήμη EEPROM ήταν διαθέσιμη και πριν το 1993 αλλά οι αρχικές εκδόσεις της ήταν αρκετά ακριβές και λιγότερο ανθεκτικές γεγονός που τις καθιστούσε μη ιδανικές για μαζική παραγωγή μικροελεγκτών χαμηλού κόστους) . Την ίδια χρονιά η Atmel παρουσίασε τον πρώτο μικροελεγκτή που χρησιμοποιούσε μνήμη flash, έναν ειδικό τύπο της EEPROM. Γρήγορα και άλλες εταιρείες ακολούθησαν, κατασκευάζοντας και τους δύο τύπους μνήμης.

Το κόστος των μικροελεγκτών μειώνεται δραματικά με την πάροδο του χρόνου, με τους οικονομικότερους 8-bit μικροελεγκτές να πέφτουν κάτω από τα 0.25 δολάρια όταν διατίθενται σε ποσότητα (1000 κομμάτια), και τους 32-bit περίπου στο 1 δολάριο για αντίστοιχες ποσότητες.

Στις μέρες μας οι μικροελεγκτές είναι αρκετά οικονομικοί και άμεσα διαθέσιμοι για χομπίστες, με μεγάλες διαδικτυακές κοινότητες και υποστήριξη πάνω σε συγκεκριμένα μοντέλα.

## ***1.5 Τρέχοντες μικροελεγκτές και αυξημένες δυνατότητες***

### ***αυτών***

Λόγω του ισχυρότατου ανταγωνισμού αλλά και της τάσης ενσωμάτωσης των μικροελεγκτών σε κάθε ηλεκτρική και ηλεκτρονική συσκευή, η βιομηχανία μικροελεγκτών έχει καταλήξει

στην παραγωγή ανταγωνιστικών μοντέλων μαζικής παραγωγής καθώς και μικροελεγκτών για πιο εξειδικευμένες εφαρμογές [5]. Έτσι διακρίνονται οι εξής κυρίως κατηγορίες:

- Μικροελεγκτές (καμιά φορά 4-bit αλλά συνήθως 8-bit) πολύ χαμηλού κόστους, γενικής χρήσης, με πολύ μικρό αριθμό ακροδεκτών (ακόμη και λιγότερους από 8). Σχεδιάζονται με έμφαση στη χαμηλή κατανάλωση ισχύος και την αυτάρκεια, ώστε να χρειάζονται ελάχιστα ή και καθόλου εξωτερικά εξαρτήματα και να μη μπορεί να αντιγραφεί εύκολα το εσωτερικό λογισμικό τους. Απουσιάζει η δυνατότητα επέκτασης της μνήμης τους. Μερικά μοντέλα είναι ευρέως γνωστά στους ερασιτέχνες ηλεκτρονικούς, όπως πχ οι περισσότεροι μικροελεγκτές των σειρών PIC (Microchip), AVR (Atmel) και 8051 (Intel, Atmel, Dallas κα).
- Μικροελεγκτές (συνήθως 8-bit αλλά και 16 ή 32-bit) χαμηλού κόστους, γενικής χρήσης, με μέτριο έως σχετικά μεγάλο αριθμό ακροδεκτών. Διαθέτουν μεγάλο αριθμό κοινών περιφερειακών, όπως θύρες UART, I<sup>2</sup>C, SPI ή CAN, μετατροπείς αναλογικού σε ψηφιακό και ψηφιακού σε αναλογικό. Στους κατασκευαστές της Άπω Ανατολής (Ιαπωνία, Κορέα), συνηθίζεται η ενσωμάτωση ελεγκτών οθόνης υγρών κρυστάλλων και πληκτρολογίου. Μερικές φορές παρέχουν δυνατότητα εξωτερικής επέκτασης της μνήμης τους.
- Μικροελεγκτές (κυρίως 32-bit) μέσου κόστους, γενικής χρήσης, με μεγάλο αριθμό ακροδεκτών. Χαρακτηρίζονται από έμφαση στην ταχύτητα εκτέλεσης εντολών, υψηλή αυτάρκεια περιφερειακών και μεγάλες δυνατότητες εσωτερικής ή εξωτερικής μνήμης προγράμματος (FLASH) και RAM. Στο χώρο αυτό έχουν ισχυρή παρουσία οι αρχιτεκτονικές με υψηλή μεταφερσιμότητα λογισμικού (portability) από τον ένα στον άλλο κατασκευαστή. Πχ μεταξύ των μικροελεγκτών τύπου ARM ή MIPS, το σύνολο των βασικών εντολών που αναγνωρίζει η ALU είναι ακριβώς το ίδιο, μειώνοντας έτσι τις μεγάλες αλλαγές στο λογισμικό, όταν στο μέλλον ο πελάτης υιοθετήσει ένα μικροελεγκτή άλλου κατασκευαστή (αρκεί, φυσικά, να υποστηρίζει κι αυτός το σύνολο εντολών ARM ή MIPS, αντίστοιχα).
- Μικροελεγκτές εξειδικευμένων εφαρμογών, οι οποίοι ενσωματώνουν συνήθως κάποιο εξειδικευμένο πρωτόκολλο επικοινωνίας το οποίο υλοποιείται πάντοτε σε hardware. Τέτοιοι μικροελεγκτές χρησιμοποιούνται σε τηλεπικοινωνιακές συσκευές όπως τα μόντεμ.

Η μεγάλη μερίδα πωλήσεων των μικροελεγκτών εξακολουθεί να αφορά αυτούς των 8-bit, καθώς είναι η κατηγορία με το χαμηλότερο κόστος και το μικρότερο μέγεθος λογισμικού για

το ίδιο αποτέλεσμα, ιδίως επειδή οι σύγχρονες οικογένειες μικροελεγκτών 8-bit έχουν πολύ βελτιωμένες επιδόσεις σε σχέση με το παρελθόν.

# 2

## *Περιγραφή εφαρμογής*

### *2.1 Σκοπός*

Σκοπός της εργασίας είναι η κατασκευή ενός συστήματος αυτόματης πλοήγησης ασύρματα χειριζόμενου οχήματος το οποίο θα κινείται σε μια επιφάνεια συγκεκριμένων διαστάσεων και θα βρίσκεται πάντα εντός εικονικά χαραγμένων δρόμων. Την παραπάνω διάταξη θα καταγράφει μια κάμερα συνδεδεμένη με τον προσωπικό μας υπολογιστή, μέσω της οποίας θα εντοπίζεται το κινούμενο όχημα σε πραγματικό χρόνο. Ο χρήστης μέσω του περιβάλλοντος της εφαρμογής που θα εκτελείται στον προσωπικό του υπολογιστή, θα ορίζει το σημείο προορισμού και το όχημα θα κινείται ακολουθώντας την κατάλληλη διαδρομή προκειμένου να φτάσει στο επιθυμητό σημείο.

### *2.2 Λειτουργίες*

Για την ανάπτυξη του συστήματος αυτόματης πλοήγησης υλοποιήθηκαν τρεις βασικές λειτουργίες:

- Ασύρματος απομακρυσμένος χειρισμός του οχήματος: Ο έλεγχος του οχήματος πραγματοποιείται μέσω της πλατφόρμας Arduino, πυρήνας της οποίας είναι ο 8-bit

μικροελεγκτής ATmega328 της εταιρείας Atmel. Για την ασύρματη επικοινωνία με τον υπολογιστή χρησιμοποιήθηκε επιπλέον η ασπίδα επέκτασης Wifi Shield της εν λόγω πλατφόρμας. Πρωταρχικός στόχος ήταν η μελέτη του τρόπου λειτουργίας του Arduino, η εύρεση του τρόπου διασύνδεσης του με το κύκλωμα του οχήματος και ο προγραμματισμός του ώστε να επιτευχθεί ο έλεγχος της κίνησης του οχήματος

- Εντοπισμός οχήματος: Για την ανίχνευση της θέσης του οχήματος χρησιμοποιήθηκε μια κάμερα τοποθετημένη σε κατακόρυφη διεύθυνση στο κέντρο της πίστας και σε απόσταση περίπου 2.5 μέτρων από το επίπεδο της. Στη πληροφορία που μας δίνει η κάμερα σε πραγματικό χρόνο γίνεται εφαρμογή αλγορίθμων υπολογιστικής όρασης προκειμένου να επιτευχθεί ο εντοπισμός του οχήματος. Μελετήθηκαν δύο ευρέως γνωστές τεχνικές υπολογιστικής όρασης, η ανίχνευση χρώματος και η ανίχνευση σημείων ιδιοαναφοράς (fiducials markers), έγινε δοκιμή και σύγκριση αυτών και επιλέχθηκε η καταλληλότερη για τις απαιτήσεις του συστήματος.
- Πλοήγηση οχήματος: Υπεύθυνο για την πλοήγηση είναι το λογισμικό που αναπτύχθηκε και υλοποιεί τη λογική της αυτόματης δρομολόγησης του οχήματος. Στο σύνολο της η εν λόγω εφαρμογή υλοποιεί επίσης τον εντοπισμό του οχήματος όπως και το περιβάλλον αλληλεπίδρασης με το χρήστη.

## 2.3 Δομικές μονάδες

Στην παρούσα ενότητα θα παρουσιαστούν οι δομικές μονάδες της συνολικής διάταξης.

### 2.3.1 Υλικό

Το υλικό που χρησιμοποιήθηκε περιλαμβάνει επιγραμματικά τα εξής:

- Πλατφόρμα Arduino
- Τηλεκατευθυνόμενο όχημα
- Κάμερα
- Modem – Router – Access Point
- Προσωπικός υπολογιστής

#### 2.3.1.1 Εισαγωγή στην πλατφόρμα Arduino

##### Τι είναι το Arduino

Όπως το περιγράφει ο δημιουργός του, το Arduino είναι μια «ανοικτού κώδικα» πλατφόρμα «πρωτοτυποποίησης» ηλεκτρονικών βασισμένη σε ευέλικτο και εύκολο στη χρήση hardware και software που προορίζεται για οποιονδήποτε έχει λίγη προγραμματιστική εμπειρία,

στοιχειώδεις γνώσεις ηλεκτρονικών και ενδιαφέρεται να δημιουργήσει διαδραστικά αντικείμενα ή περιβάλλοντα [7][8]. Στην ουσία, πρόκειται για ένα ηλεκτρονικό κύκλωμα που βασίζεται στον μικροελεγκτή ATmega της Atmel και του οποίου όλα τα σχέδια, καθώς και το software που χρειάζεται για την λειτουργία του, διανέμονται ελεύθερα και δωρεάν ώστε να μπορεί να κατασκευαστεί από τον καθένα (απ' όπου και ο περιέργος - για hardware-χαρακτηρισμός «ανοικτού κώδικα»). Αφού κατασκευαστεί, μπορεί να συμπεριφερθεί σαν ένας μικροσκοπικός υπολογιστής, αφού ο χρήστης μπορεί να συνδέσει επάνω του πολλαπλές μονάδες εισόδου/εξόδου και να προγραμματίσει τον μικροελεγκτή να δέχεται δεδομένα από τις μονάδες εισόδου, να τα επεξεργάζεται και να στέλνει κατάλληλες εντολές στις μονάδες εξόδου.

#### Η γλώσσα προγραμματισμού της πλατφόρμας

Η γλώσσα προγραμματισμού που χρησιμοποιείται είναι η Wiring η οποία είναι αρκετά εύκολη στη σύνταξη και διατίθεται σε πλατφόρμες Linux, MAC, Windows με άδεια χρήσης GPL. Αυτό όμως που κάνει το Arduino ακόμα πιο σημαντικό είναι ότι όλο το κύκλωμα της πλακέτας διατίθεται με άδεια χρήσης Creative Commons, πράγμα που σημαίνει ότι ο καθένας μπορεί να κατασκευάσει τη δική του πλακέτα όπως αυτός επιθυμεί. Μάλιστα κάποιος θα μπορούσε να ισχυριστεί – και θα ήταν ένας αρκετά επιτυχημένος παραλληλισμός – ότι λειτουργικά το Arduino μοιάζει πολύ με το NXT των Lego Mindstorms Nxt. Άλλωστε το Arduino διαπρέπει στις εφαρμογές της ρομποτικής. Το Arduino βέβαια δεν είναι ούτε ο μοναδικός, ούτε και ο καλύτερος δυνατός τρόπος για τη δημιουργία μιας οποιασδήποτε διαδραστικής ηλεκτρονικής συσκευής.

#### Πλεονεκτήματα Arduino

Το κύριο πλεονέκτημά του Arduino είναι η τεράστια κοινότητα που το υποστηρίζει και η οποία έχει δημιουργήσει, συντηρεί και επεκτείνει μια ανάλογου μεγέθους online γνωσιακή βάση. Έτσι, παρότι ένας έμπειρος ηλεκτρονικός μπορεί να προτιμήσει διαφορετική πλατφόρμα ή εξαρτήματα ανάλογα με την εφαρμογή που έχει στον νου του, το Arduino, με το εκτενές documentation, καταφέρνει να κερδίσει όλους αυτούς των οποίων οι γνώσεις στα ηλεκτρονικά περιορίζονται στα όσα λίγα έμαθαν στο σχολείο. Επίσης οι περισσότεροι προμηθευτές διαθέτουν Arduino Starter Kit, τα οποία, εκτός από το ίδιο το Arduino, περιέχουν διάφορα άλλα εξαρτήματα και εργαλεία που μπορεί να χρειαστείτε για τις πρώτες σας εφαρμογές (όπως το απαραίτητο καλώδιο USB για την σύνδεση με τον υπολογιστή, ράστερ, καλώδια, LED, διακόπτες, ποτενσιόμετρα, αντιστάσεις, διόδους, τρανζίστορ κ.λπ.). Επιπλέον τα drivers και το περιβάλλον ανάπτυξης της πλατφόρμας είναι συμβατά με πολλαπλά λειτουργικά συστήματα. Οι μηχανικοί λογισμικού ανέπτυξαν το περιβάλλον

προγραμματισμού του Arduino για Windows, Macintosh OS X και για τα λειτουργικά συστήματα Linux, ενώ τα περισσότερα συστήματα ανάπτυξης μικροελεγκτών περιορίζονται στα Windows.

### To hardware

Μία πλακέτα Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR (ATmega328 και ATmega168 στις νεότερες εκδόσεις, ATmega8 στις παλαιότερες) και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωση του σε άλλα κυκλώματα. Μια σημαντική πτυχή του Arduino είναι ο τρόπος που εκτίθενται οι ακροδέκτες, επιτρέποντας στην πλακέτα να συνδεθεί με μια ποικιλία εναλλάξιμων add-on modules γνωστές ως ασπίδες. Μερικές ασπίδες επικοινωνούν με την πλακέτα Arduino πάνω ακριβώς από διάφορους ακροδέκτες εισόδου-εξόδου. Πολλές ωστόσο ασπίδες είναι μεμονωμένα διευθετημένες μέσω ενός I<sup>2</sup>C σειριακού διαύλου γεγονός που τους επιτρέπει να στοιβάζονται και να χρησιμοποιούνται παράλληλα. Το αυθεντικό Arduino χρησιμοποιεί τη σειρά megaAVR των μικροελεγκτών και συγκεκριμένα τους ATmega8, ATmega168, ATmega328, ATmega1280 και ATmega2560. Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz (ή κεραμικό αντηχητή σε κάποιες παραλλαγές). Ο μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα bootloader, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής.

Σε εννοιολογικό επίπεδο, όταν γίνεται χρήση της στοίβας λογισμικού Arduino, όλες οι πλακέτες προγραμματίζονται με μία RS-232 σειριακή σύνδεση, αλλά ο τρόπος που επιτυγχάνεται αυτό διαφέρει ανάλογα με τον τύπο του μικροελεγκτή και το υλικό που χρησιμοποιείται. Οι σειριακές πλακέτες Arduino περιέχουν ένα απλό level shifter κύκλωμα για τη μετατροπή μεταξύ σήματος επιπέδου RS-232 και TTL. Οι σύγχρονες πλακέτες Arduino προγραμματίζονται μέσω USB μέσω της εφαρμογής προσαρμοστικών chip USB-to-Serial όπως το FTDI FT232. Κάποιες παραλλαγές, όπως το Arduino mini και το ανεπίσημο Boarduino, χρησιμοποιούν ένα αφαιρούμενο USB-to-Serial καλώδιο ή board, Bluetooth ή άλλες μεθόδους.

Ο πίνακας Arduino εκθέτει τους περισσότερους ακροδέκτες εισόδου-εξόδου του μικροελεγκτή για χρήση από άλλα κυκλώματα. Τα μοντέλα Diecimila, Duemilanove και το τρέχον Uno παρέχουν 14 ακροδέκτες ψηφιακών εισόδων/εξόδων, έξι από τους οποίους μπορούν να παράγουν pulse-width διαμορφωμένα σήματα, και έξι αναλογικά δεδομένα. Αυτά τα pins βρίσκονται στην κορυφή του πίνακα μέσω female headers 0.1 ιντσών (2,2mm).

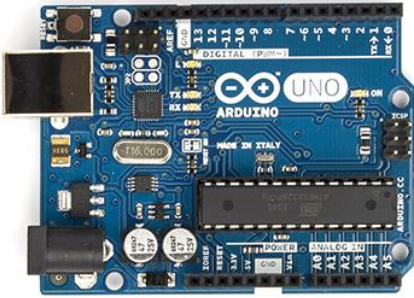
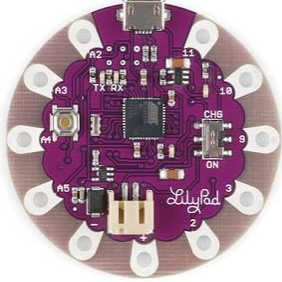
Το Arduino Nano, οι Bare Bones πλακέτες που είναι συμβατές με την πλατφόρμα Arduino, και η πλακέτα Boarduino ενδέχεται να παρέχουν αρσενικούς ακροδέκτες στο κάτω μέρος της

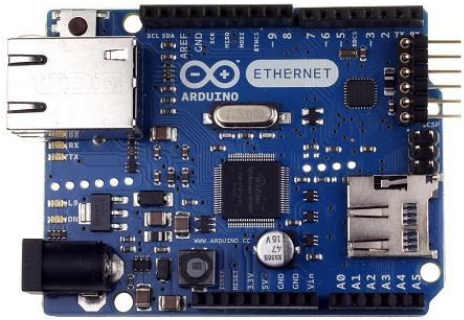


πλακέτας προκειμένου να συνδέονται σε Breadboards [9]. Υπάρχουν πολλές παραλλαγές πλακετών προερχόμενες από Arduino boards που είναι συμβατές με την πλατφόρμα. Κάποιες είναι λειτουργικά ισάξιες με ένα Arduino και μπορεί να χρησιμοποιηθούν εναλλακτικά. Πολλές από αυτές ουσιαστικά αποτελούνται από το βασικό Arduino με την προσθήκη καινοτόμων οδηγών εξόδου, κυρίως για εκπαιδευτική χρήση και απλοποίηση της κατασκευής μικρών οχημάτων και ρομπότ. Άλλες είναι ηλεκτρικά ισάξιες αλλά αλλάζουν τον παράγοντα μορφής, επιτρέποντας κάποιες φορές την συνεχόμενη χρήση των Shields ενώ κάποιες όχι. Κάποιες παραλλαγές είναι τελείως διαφορετικοί επεξεργαστές, με διαφορετικά επίπεδα συμβατότητας.

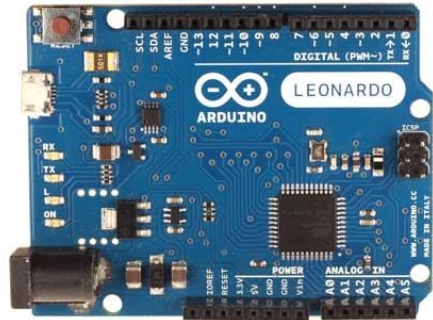
### Μοντέλα Arduino

Τα πιο δημοφιλή μοντέλα Arduino που κυκλοφορούν στην αγορά, παρουσιάζονται στον παρακάτω πίνακα. Στην παρούσα εργασία χρησιμοποιήθηκε το Arduino Uno καθώς καλύπτει όλες τις ανάγκες του συστήματος και οι ακροδέκτες του επαρκούν για όλες τις λειτουργίες.

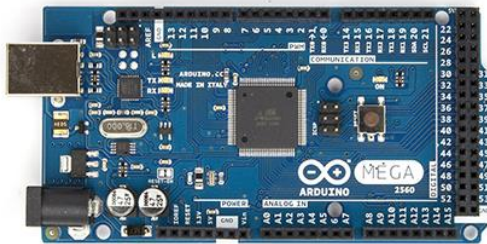
 A photograph of an Arduino Uno R3 microcontroller board. It is a blue PCB with a USB Type-B port on the left, a DC power jack, and a 5-pin header. The board features the ATmega328P microcontroller, a USB-to-UART bridge, and various passive components like resistors and capacitors. The text 'ARDUINO UNO' and the Arduino logo are visible on the board.	Arduino Uno
 A photograph of an Arduino LilyPad microcontroller board. It is a small, circular, purple PCB designed for wearable electronics. It has a USB Type-B port, a DC power jack, and several solder tabs for connections. The board features the ATmega328P microcontroller and various components. The text 'ARDUINO LilyPad' is visible on the board.	Arduino LilyPad



Arduino Ethernet



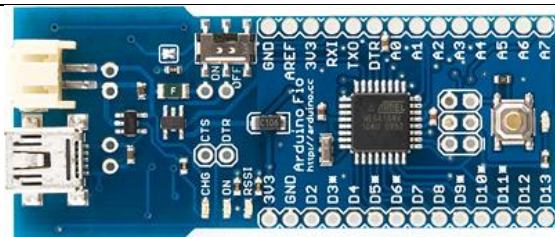
Arduino Leonardo




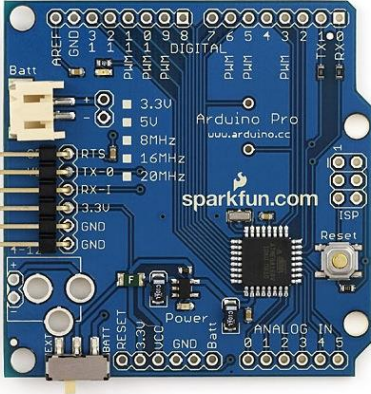

Arduino Mega2560



Arduino Mega ADK



Arduino Fio

	<p>Arduino Nano</p>
	<p>Arduino Pro</p>
	<p>Arduino Pro Mini</p>

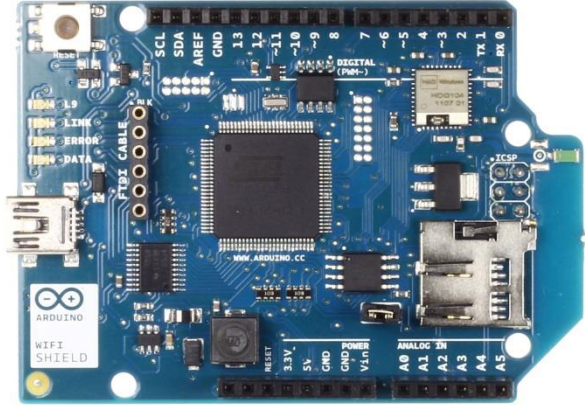
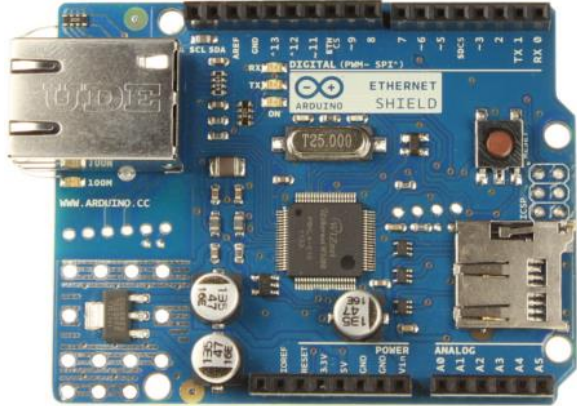
Πίνακας 2.1 : Τα δημοφιλέστερα μοντέλα Arduino

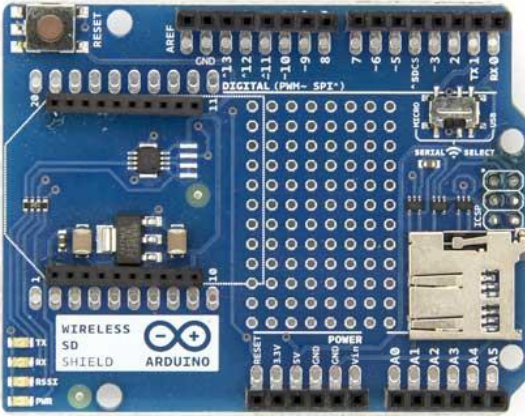
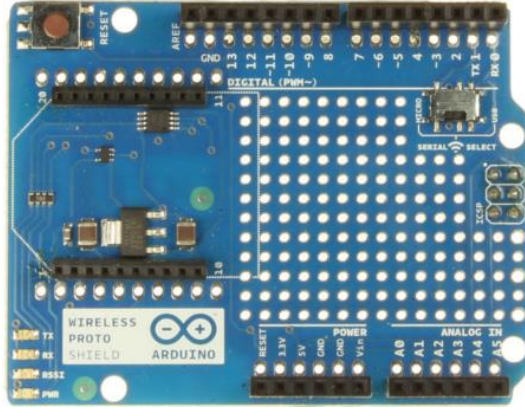
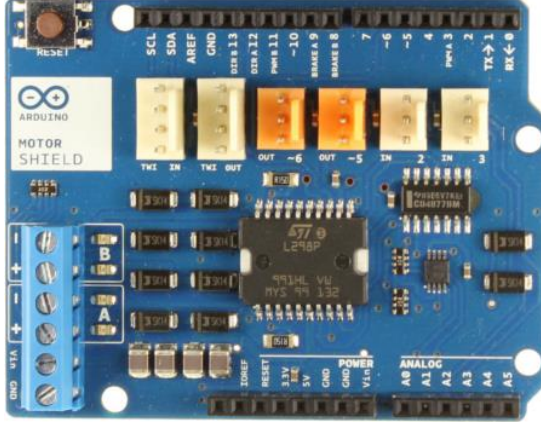
Arduino Shields

Στην παρούσα εργασία χρησιμοποιήθηκε το Wifi Shield που επιτρέπει στην πλατφόρμα Arduino να συνδεθεί στο διαδίκτυο χρησιμοποιώντας τις προδιαγραφές 802.11 ασυρμάτων δικτύων (Wifi). Η Ασπίδα Wi-Fi μπορεί να συνδεθεί με ασύρματα δίκτυα που λειτουργούν σύμφωνα με τις 802.11b και 802.11g προδιαγραφές. Περιέχει τον μικροελεγκτή 32-bit ATmega 32UC3, ο οποίος υποστηρίζει τα πρωτόκολλα UDP (User Datagram Protocol) και TCP (Transmission Control Protocol). Για την εγγραφή του κώδικα είναι απαραίτητη η χρήση της βιβλιοθήκης Wi-Fi library , η οποία συνδέεται στο διαδίκτυο χρησιμοποιώντας την

ασπίδα. Η ασπίδα Wi-Fi συνδέεται στην πλακέτα Arduino χρησιμοποιώντας wire-wrap headers, τα οποία εκτείνονται μέσα στο Wi-fi shield [10]. Αυτό κρατά την διάταξη των pins άθικτη και έτσι επιτρέπεται και σε άλλες ασπίδες να στοιβάζονται στην κορυφή της ασπίδας. Επίσης στην Wi-Fi ασπίδα υπάρχει μια micro – SD υποδοχή κάρτας η οποία χρησιμοποιείται για την αποθήκευση αρχείων για μεταφορά μέσω διαδικτύου. Για την πρόσβαση στα δεδομένα της micro SD κάρτας είναι απαραίτητη η χρήση της βιβλιοθήκης SD. Η ασπίδα Wi-Fi είναι συμβατή με τις πλακέτες Arduino Uno και Mega. Η πλακέτα Arduino επικοινωνεί με την ασπίδα και την κάρτα SD, χρησιμοποιώντας τον διάυλο SPI (Serial Peripheral Interface). Η ασπίδα μπορεί να συνδεθεί σε κρυπτογραφημένα δίκτυα (encrypted networks) που χρησιμοποιούν είτε WPA2 Personal ή WEP κρυπτογράφηση. Επίσης μπορεί να συνδεθεί και σε ανοιχτά δίκτυα.

Στον πίνακα που ακολουθεί παρουσιάζονται οι δημοφιλέστερες ασπίδες επέκτασης της πλατφόρμας Arduino.

	<p style="text-align: center;"><b>Arduino Wifi Shield</b></p> <p><i>Επιτρέπει στην πλατφόρμα Arduino να συνδεθεί στο διαδίκτυο χρησιμοποιώντας τις προδιαγραφές 802.11 ασυρμάτων δικτύων (Wifi)</i></p>
	<p style="text-align: center;"><b>Arduino Ethernet Shield</b></p> <p><i>Επιτρέπει στον Arduino να συνδεθεί στο διαδίκτυο χρησιμοποιώντας τις προδιαγραφές του πρωτοκόλλου 802.3 ενσύρματης τοπικής δικτύωσης υπολογιστών.</i></p>

	<p>Wireless SD Shield</p> <p><i>Συμβατό με Xbee module ασύρματης δικτύωσης που έρχεται από την εταιρεία Digi. Η ασύρματη κάλυψη αγγίζει τα 30,5 μέτρα σε εσωτερικούς χώρους και τα 91,5 μέτρα σε εξωτερικούς χώρους και σε οπτική επαφή.</i></p>
	<p>Wireless Proto Shield</p> <p><i>Επιτρέπει στον Arduino να επικοινωνήσει ασύρματα με χρήση ενός Xbee module.</i></p>
	<p>Motor Shield</p> <p><i>Βασίζεται στο ολοκληρωμένο L298 που χρησιμοποιείται για οδήγηση επαγωγικών φορτίων όπως ηλεκτρονόμους ή DC κινητήρες.</i></p>

Πίνακας 2.2: Arduino Wifi Shields

### 2.3.1.2 Αναλυτική παρουσίαση του Arduino Uno

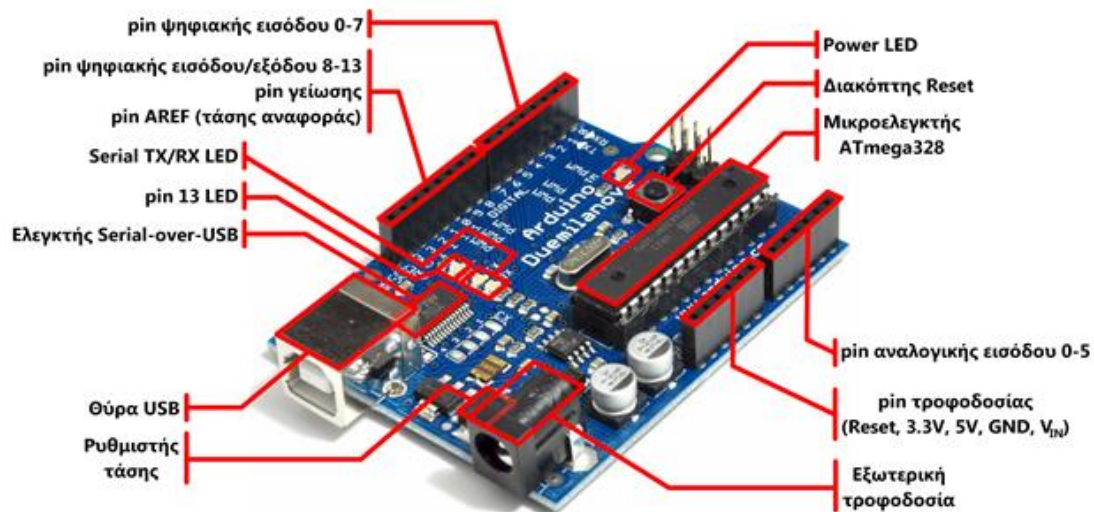
Το Arduino Uno βασίζεται στον ATmega328, έναν 8-bit RISC μικροελεγκτή, τον οποίο χρονίζει στα 16MHz. Σημειώνεται ότι χρησιμοποιεί την ίδια τεχνολογία ATmega328 όπως το τελευταίο μοντέλο Duemilanove, αλλά ενώ το Duemilanove χρησιμοποιεί ένα FTDI chipset

για το USB, το Uno χρησιμοποιεί τεχνολογία ATmega8U2 προγραμματισμένο ως σειριακός μετατροπέας [11]. Ο ATmega328 διαθέτει ενσωματωμένη μνήμη τριών τύπων:

- 2Kb μνήμης SRAM που είναι η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματά σας για να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. κατά το runtime. Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή αν γίνει reset.
- 1Kb μνήμης EEPROM η οποία μπορεί να χρησιμοποιηθεί για «ωμή» εγγραφή/ανάγνωση δεδομένων (χωρίς datatype) ανά byte από τα προγράμματά σας κατά το runtime. Σε αντίθεση με την SRAM, η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset οπότε είναι το ανάλογο του σκληρού δίσκου.
- 32Kb μνήμης Flash, από τα οποία τα 2Kb χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware αυτό που στην ορολογία του Arduino ονομάζεται bootloader είναι αναγκαίο για την εγκατάσταση των δικών σας προγραμμάτων στον μικροελεγκτή μέσω της θύρας USB, χωρίς δηλαδή να χρειάζεται εξωτερικός hardware programmer. Τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν στον υπολογιστή σας. Η μνήμη Flash, όπως και η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset. Επίσης, ενώ η μνήμη Flash υπό κανονικές συνθήκες δεν προορίζεται για χρήση runtime μέσα από τα προγράμματά σας, λόγω της μικρής συνολικής μνήμης που είναι διαθέσιμη σε αυτά (2Kb SRAM + 1Kb EEPROM), έχει σχεδιαστεί μια βιβλιοθήκη που επιτρέπει την χρήση όσου χώρου περισσεύει (30Kb μείον το μέγεθος του προγράμματός σας σε μεταγλωττισμένη μορφή).

### Είσοδοι – Εξοδοι

Καταρχήν το Arduino διαθέτει σειριακό interface. Ο μικροελεγκτής ATmega υποστηρίζει σειριακή επικοινωνία, την οποία το Arduino προωθεί μέσα από έναν ελεγκτή Serial-over-USB ώστε να συνδέεται με τον υπολογιστή μέσω USB. Η σύνδεση αυτή χρησιμοποιείται για την μεταφορά των προγραμμάτων που σχεδιάζονται από τον υπολογιστή στο Arduino αλλά και για αμφίδρομη επικοινωνία του Arduino με τον υπολογιστή μέσα από το πρόγραμμα την ώρα που εκτελείται.



Εικόνα 2.1: Arduino Uno/Duemilanova

Επιπλέον, στην πάνω πλευρά του Arduino βρίσκονται 14 θηλυκά pin, αριθμημένα από 0 ως 13, που μπορούν να λειτουργήσουν ως ψηφιακές εισόδους και εξόδους. Λειτουργούν στα 5V και καθένα μπορεί να παρέχει ή να δεχτεί το πολύ 40mA. Ως ψηφιακή έξοδος, ένα από αυτά τα pin μπορεί να τεθεί από το πρόγραμμά σας σε κατάσταση HIGH ή LOW, οπότε το Arduino θα ξέρει αν πρέπει να διοχετεύσει ή όχι ρεύμα στο συγκεκριμένο pin. Με αυτόν τον τρόπο μπορείτε λόγω χάρη να ανάψετε και να σβήσετε ένα LED που έχετε συνδέσει στο συγκεκριμένο pin. Αν πάλι ρυθμίσετε ένα από αυτά τα pin ως ψηφιακή είσοδο μέσα από το πρόγραμμά σας, μπορείτε με την κατάλληλη εντολή να διαβάσετε την κατάστασή του (HIGH ή LOW) ανάλογα με το αν η εξωτερική συσκευή που έχετε συνδέσει σε αυτό το pin διοχετεύει ή όχι ρεύμα στο pin (με αυτόν τον τρόπο μπορείτε να «διαβάζετε» την κατάσταση ενός διακόπτη). Μερικά από αυτά τα 14 pin, εκτός από ψηφιακές εισόδους/εξόδους έχουν και δεύτερη λειτουργία. Συγκεκριμένα:

- Τα pin 0 και 1 λειτουργούν ως RX και TX της σειριακής όταν το πρόγραμμά σας ενεργοποιεί την σειριακή θύρα. Έτσι, όταν το πρόγραμμά σας στέλνει δεδομένα στην σειριακή, αυτά προωθούνται και στην θύρα USB μέσω του ελεγκτή Serial-Over-USB αλλά και στο pin 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή (π.χ. ένα δεύτερο Arduino στο δικό του pin 1). Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμά σας ενεργοποιήσετε το σειριακό interface, χάνετε 2 ψηφιακές εισόδους/εξόδους.
- Τα pin 2 και 3 λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Με άλλα λόγια, μπορείτε να τα ρυθμίσετε μέσα από το πρόγραμμά σας ώστε να λειτουργούν αποκλειστικά ως ψηφιακές εισόδους στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει άμεσα και

εκτελείται μια συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.

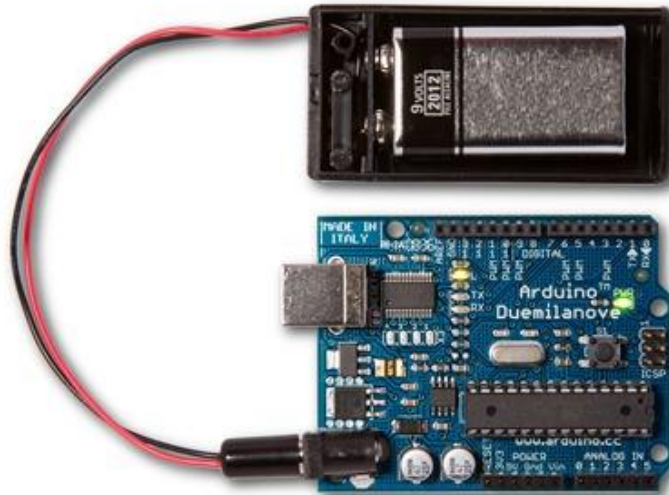
- Τα pin 3, 5, 6, 9, 10 και 11 μπορούν να λειτουργήσουν και ως ψευδοαναλογικές έξοδοι με το σύστημα PWM (Pulse Width Modulation), δηλαδή το ίδιο σύστημα που διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων. Έτσι, μπορείτε να συνδέσετε λόγω χάρη ένα LED σε κάποιο από αυτά τα pin και να ελέγξετε πλήρως την φωτεινότητά του με ανάλυση 8bit (256 καταστάσεις από 0-σβηστό ως 255-πλήρως αναμμένο) αντί να έχετε απλά την δυνατότητα αναμμένο-σβηστό που παρέχουν οι υπόλοιπες ψηφιακές έξοδοι. Είναι σημαντικό να καταλάβετε ότι το PWM δεν είναι πραγματικά αναλογικό σύστημα και ότι θέτοντας στην έξοδο την τιμή 127, δεν σημαίνει ότι η έξοδος θα δίνει 2.5V αντί της κανονικής τιμής των 5V, αλλά ότι θα δίνει ένα παλμό που θα εναλλάσσεται με μεγάλη συχνότητα και για ίσους χρόνους μεταξύ των τιμών 0 και 5V.

Στην κάτω πλευρά του Arduino, με τη σήμανση ANALOG IN, θα βρείτε μια ακόμη σειρά από 6 pin, αριθμημένα από το 0 ως το 5. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος κάνοντας χρήση του ADC (Analog to Digital Converter) που είναι ενσωματωμένο στον μικροελεγκτή. Για παράδειγμα, μπορείτε να τροφοδοτήσετε ένα από αυτά με μια τάση την οποία μπορείτε να κυμάνετε με ένα ποτενσιόμετρο από 0V ως μια τάση αναφοράς Vref η οποία, αν δεν κάνετε κάποια αλλαγή είναι προρυθμισμένη στα 5V. Τότε, μέσα από το πρόγραμμά σας μπορείτε να «διαβάσετε» την τιμή του pin ως ένα ακέραιο αριθμό ανάλυσης 10-bit, από 0 (όταν η τάση στο pin είναι 0V) μέχρι 1023 (όταν η τάση στο pin είναι 5V). Η τάση αναφοράς μπορεί να ρυθμιστεί με μια εντολή στο 1.1V, ή σε όποια τάση επιθυμείτε (μεταξύ 2 και 5V) τροφοδοτώντας εξωτερικά με αυτή την τάση το pin με την σήμανση AREF που βρίσκεται στην απέναντι πλευρά της πλακέτας. Έτσι, αν τροφοδοτήσετε το pin AREF με 3.3V και στην συνέχεια δοκιμάσετε να διαβάσετε κάποιο pin αναλογικής εισόδου στο οποίο εφαρμόζετε τάση 1.65V, το Arduino θα σας επιστρέψει την τιμή 512. Τέλος, καθένα από τα 6 αυτά pin, με κατάλληλη εντολή μέσα από το πρόγραμμα μπορεί να μετατραπεί σε ψηφιακό pin εισόδου/εξόδου όπως τα 14 που βρίσκονται στην απέναντι πλευρά και τα οποία περιγράφηκαν πριν. Σε αυτή την περίπτωση τα pin μετονομάζονται από 0~5 σε 14~19 αντίστοιχα.

### Τροφοδοσία

Το Arduino μπορεί να τροφοδοτηθεί με ρεύμα είτε από τον υπολογιστή μέσω της σύνδεσης USB, είτε από εξωτερική τροφοδοσία που παρέχεται μέσω μιας υποδοχής φισ των 2.1mm (θετικός πόλος στο κέντρο) και βρίσκεται στην κάτω-αριστερή γωνία του Arduino.





Εικόνα 2.2: Σύνδεση Arduino με εξωτερική τροφοδοσία

Για να μην υπάρχουν προβλήματα, η εξωτερική τροφοδοσία πρέπει να είναι από 7 ως 12V και μπορεί να προέρχεται από ένα κοινό μετασχηματιστή του εμπορίου, από μπαταρίες ή οποιαδήποτε άλλη πηγή DC. Δίπλα από τα pin αναλογικής εισόδου, υπάρχει μια ακόμα συστοιχία από 6 pin με την σήμανση POWER. Η λειτουργία του καθενός έχει ως εξής:

- Το πρώτο, με την ένδειξη RESET, όταν γειωθεί (σε οποιοδήποτε από τα 3 pin με την ένδειξη GND που υπάρχουν στο Arduino) έχει ως αποτέλεσμα την επανεκκίνηση του Arduino.
- Το δεύτερο, με την ένδειξη 3.3V, μπορεί να τροφοδοτήσει τα εξαρτήματά σας με τάση 3.3V. Η τάση αυτή δεν προέρχεται από την εξωτερική τροφοδοσία αλλά παράγεται από τον ελεγκτή Serial-over-USB και έτσι η μέγιστη ένταση που μπορεί να παρέχει είναι μόλις 50mA.
- Το τρίτο, με την ένδειξη 5V, μπορεί να τροφοδοτήσει τα εξαρτήματά σας με τάση 5V. Ανάλογα με τον τρόπο τροφοδοσίας του ίδιου του Arduino, η τάση αυτή προέρχεται είτε άμεσα από την θύρα USB (που ούτως ή άλλως λειτουργεί στα 5V), είτε από την εξωτερική τροφοδοσία αφού αυτή περάσει από ένα ρυθμιστή τάσης για να την «φέρει» στα 5V.
- Το τέταρτο και το πέμπτο pin, με την ένδειξη GND, είναι φυσικά γειώσεις.
- Το έκτο και τελευταίο pin, με την ένδειξη Vin έχει διπλό ρόλο. Σε συνδυασμό με το pin γείωσης δίπλα του, μπορεί να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας του Arduino, στην περίπτωση που δεν σας βολεύει να χρησιμοποιήσετε την υποδοχή του φιν των 2.1mm. Αν όμως έχετε ήδη συνδεδεμένη εξωτερική τροφοδοσία μέσω του φιν, μπορείτε να χρησιμοποιήσετε αυτό το pin για να τροφοδοτήσετε εξαρτήματα

με την πλήρη τάση της εξωτερικής τροφοδοσίας (7~12V), πριν αυτή περάσει από τον ρυθμιστή τάσης όπως γίνεται με το pin των 5V.

### Ενσωματωμένα κουμπιά και led

Πάνω στην πλακέτα του Arduino υπάρχει ένας διακόπτης micro-switch και 4 μικροσκοπικά LED επιφανειακής στήριξης. Η λειτουργία του διακόπτη (που έχει την σήμανση RESET) και του ενός LED με την σήμανση POWER είναι μάλλον προφανής. Τα δύο LED με τις σημάνσεις TX και RX, χρησιμοποιούνται ως ένδειξη λειτουργίας του σειριακού interface, καθώς ανάβουν όταν το Arduino στέλνει ή λαμβάνει (αντίστοιχα) δεδομένα μέσω USB. Σημειώστε ότι τα LED αυτά ελέγχονται από τον ελεγκτή Serial-over-USB και συνεπώς δεν λειτουργούν όταν η σειριακή επικοινωνία γίνεται αποκλειστικά μέσω των ψηφιακών pin 0 και 1. Τέλος, υπάρχει το LED με την σήμανση L. Η βασική δοκιμή λειτουργίας του Arduino είναι να του αναθέσετε να αναβοσβήνει ένα LED (θα το δείτε αυτό στην συνέχεια όταν θα φτιάξετε την πρώτη εφαρμογή σας). Για να μπορείτε να το κάνετε αυτό από την πρώτη στιγμή, χωρίς να συνδέσετε τίποτα πάνω στο Arduino, οι κατασκευαστές του σκέφτηκαν να ενσωματώσουν ένα LED στην πλακέτα, το οποίο σύνδεσαν στο ψηφιακό pin 13. Έτσι, ακόμα και αν δεν έχετε συνδέσει τίποτα πάνω στο φυσικό pin 13, αναθέτοντάς του την τιμή HIGH μέσα από το πρόγραμμά σας, θα ανάψει αυτό το ενσωματωμένο LED.

### Arduino IDE

Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java [11], που λειτουργεί σε πολλές πλατφόρμες, και προέρχεται από το IDE για τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring. Έχει σχεδιαστεί για να εισαγάγει τον προγραμματισμό στους καλλιτέχνες και τους νέους που δεν είναι εξοικειωμένοι με την ανάπτυξη λογισμικού. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης και ο συνδυασμός αγκύλων και είναι επίσης σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα με ένα μόνο κλικ. Δεν υπάρχει συνήθως καμία ανάγκη να επεξεργαστείτε αρχεία make ή να τρέξετε προγράμματα σε ένα περιβάλλον γραμμής εντολών. Ένα πρόγραμμα ή κώδικας που γράφτηκε για Arduino ονομάζεται σκίτσο (sketch).

Τα Arduino προγράμματα είναι γραμμένα σε C ή C++. Το Arduino IDE έρχεται με μια βιβλιοθήκη λογισμικού που ονομάζεται "Wiring" από το πρωτότυπο σχέδιο Wiring γεγονός που καθιστά πολλές κοινές λειτουργίες εισόδου/εξόδου πολύ πιο εύκολες. Οι χρήστες πρέπει μόνο να ορίσουν δύο λειτουργίες για να κάνουν ένα πρόγραμμα κυκλικής εκτέλεσης:

-setup(): μία συνάρτηση που τρέχει μία φορά στην αρχή του προγράμματος η οποία αρχικοποιεί τις ρυθμίσεις

-loop(): μία συνάρτηση η οποία καλείται συνέχεια μέχρι η πλακέτα να απενεργοποιηθεί

Ένα τυπικό πρώτο πρόγραμμα για έναν μικροελεγκτή αναβοσβήνει απλά ένα LED. Στο περιβάλλον του Arduino, ο χρήστης μπορεί να γράψει ένα πρόγραμμα σαν αυτό:

```
#define LED_PIN 13

void setup () {
  pinMode (LED_PIN, OUTPUT); // enable pin 13 for digital output
}

void loop () {
  digitalWrite (LED_PIN, HIGH); // turn on the LED
  delay (1000); // wait one second (1000 milliseconds)
  digitalWrite (LED_PIN, LOW); // turn off the LED
  delay (1000); // wait one second
}
```

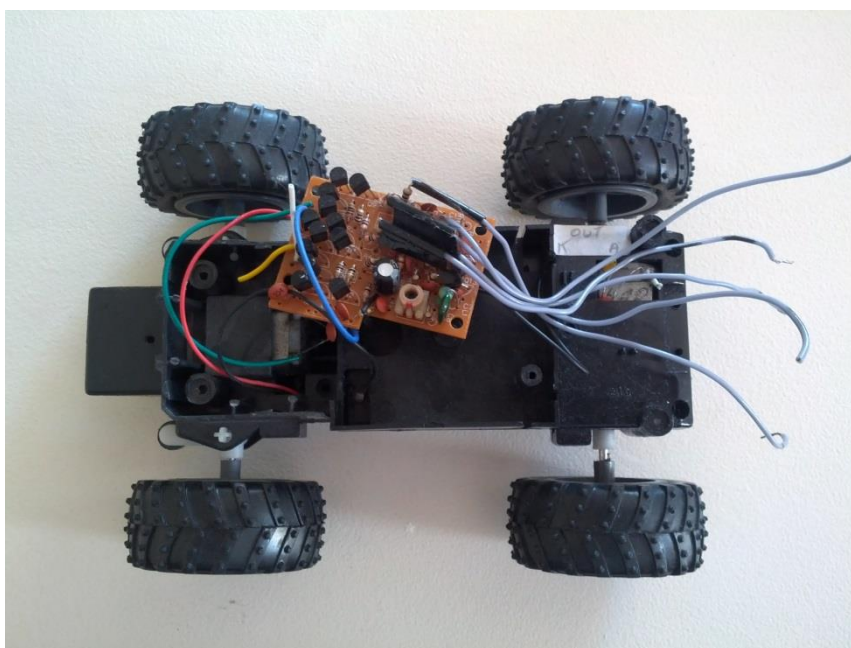
Είναι ένα χαρακτηριστικό των περισσότερων πλακετών Arduino ότι έχουν ένα LED και μία αντίσταση φορτίου που συνδέονται μεταξύ του pin 13 και του εδάφους, ένα βολικό χαρακτηριστικό για πολλά απλά τεστ. Ο προηγούμενος κώδικας δεν θα αναγνωριστεί από ένα κανονικό μεταγλωττιστή C++ ως έγκυρο πρόγραμμα, έτσι ώστε όταν ο χρήστης κάνει κλικ στο κουμπί "Upload to I / O board" στο IDE, ένα αντίγραφο του κώδικα θα γραφτεί σε ένα προσωρινό αρχείο με ένα παραπάνω include στην κορυφή και μία πολύ απλή συνάρτηση main() στο τέλος, για να φτιάξει ένα έγκυρο C++ πρόγραμμα.

Το IDE του Arduino χρησιμοποιεί το GNU toolchain και το AVR Libc για να μεταγλωττίζει προγράμματα και το avrdude για να φορτώνει προγράμματα στην πλακέτα.

Δεδομένου ότι η πλατφόρμα Arduino χρησιμοποιεί Atmel μικροελεγκτές, το περιβάλλον ανάπτυξης της Atmel, το AVR Studio ή η νεότερη έκδοση του Atmel Studio, μπορεί επίσης να χρησιμοποιηθεί για την ανάπτυξη λογισμικού για το Arduino.

### 2.3.1.3 Τηλεκατευθυνόμενο όχημα

Στα πρώτα στάδια ανάπτυξης του συστήματος χρησιμοποιήθηκε ένα τηλεκατευθυνόμενο αυτοκίνητο χαμηλού κόστους που ανήκει ουσιαστικά στην κατηγορία παιχνιδιών όπως φαίνεται και στην φωτογραφία που ακολουθεί. Διαθέτει δύο dc κινητήρες που λειτουργούν σε ονομαστική τάση 4.5V, ο ένας κινεί τους εμπρόσθιους τροχούς και στρίβει το όχημα και ο δεύτερος κινεί τους οπίσθιους τροχούς μετακινώντας το όχημα μπροστά ή πίσω. Οι δυνατές κινήσεις που μπορεί να πραγματοποιήσει είναι πλήρη στροφή δεξιά, πλήρη στροφή αριστερά, πορεία μπροστά και πορεία πίσω.



Εικόνα 2.3: 1<sup>ο</sup> όχημα που χρησιμοποιήθηκε

Όπως θα αναλυθεί και σε επόμενο κεφάλαιο το πρώτο όχημα μετά από δοκιμές που έγιναν δεν κατάφερε να ανταποκριθεί στις απαιτήσεις του συστήματος. Οι βασικοί λόγοι είναι οι εξής:

- Η ταχύτητα του ήταν αρκετά υψηλή για τα δεδομένα της εφαρμογή με αποτέλεσμα να μειώνεται σημαντικά ο βαθμός ελεγχιμότητας του.
- Η υψηλή ταχύτητα επηρεάζει αρνητικά την σταθερότητα και την ακρίβεια του εντοπισμού του καθ'όλη τη διάρκεια λειτουργία του συστήματος
- Όσο πιο γρήγορα κινείται το όχημα τόσο αυξάνεται και ο αριθμός των εντολών που μεταβιβάζονται στον Arduino και πρέπει να επεξεργαστούν από το μικροελεγκτή του. (Στην αρχική υλοποίηση ασύρματης επικοινωνίας με χρήση του πρωτοκόλλου http βγήκαν στην επιφάνεια σημαντικά προβλήματα καθώς το χρονικό παράθυρο

μεταξύ δύο συνεχόμενων GET request που μπορούσε να επεξεργαστεί ο Arduino ήταν κατά μέσο όρο 1,5 δευτερόλεπτα)

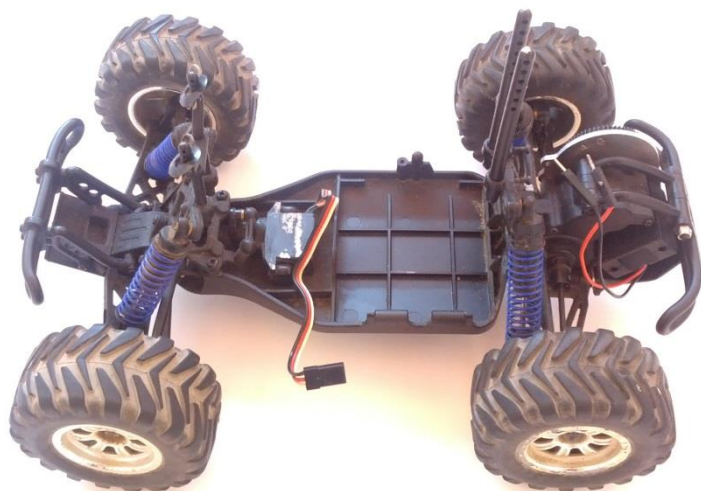
- Ο τρόπος περιστροφής των τροχών του αυτοκινήτου δεν ήταν προβλέψιμος και σταθερός. Λόγω χαμηλής ποιότητας κατασκευής πολλές φορές το όχημα παρουσίαζε μεγάλες αποκλίσεις στην ευθεία πορεία και ειδικότερα στις στροφές.
- Έλλειψη δυνατότητας διόρθωσης πορείας

Να σημειωθεί ότι έγινε δοκιμή να τροφοδοτήσουμε τον ακροδέκτη που ελέγχει το πίσω μοτέρ με PWM σήμα όπως επίσης να μειώσουμε την τάση στο πίσω μοτέρ με έναν ρυθμιστή τάσης (LM2596 Step-down Voltage Regulator) αλλά χωρίς και πάλι ικανοποιητικά αποτελέσματα.

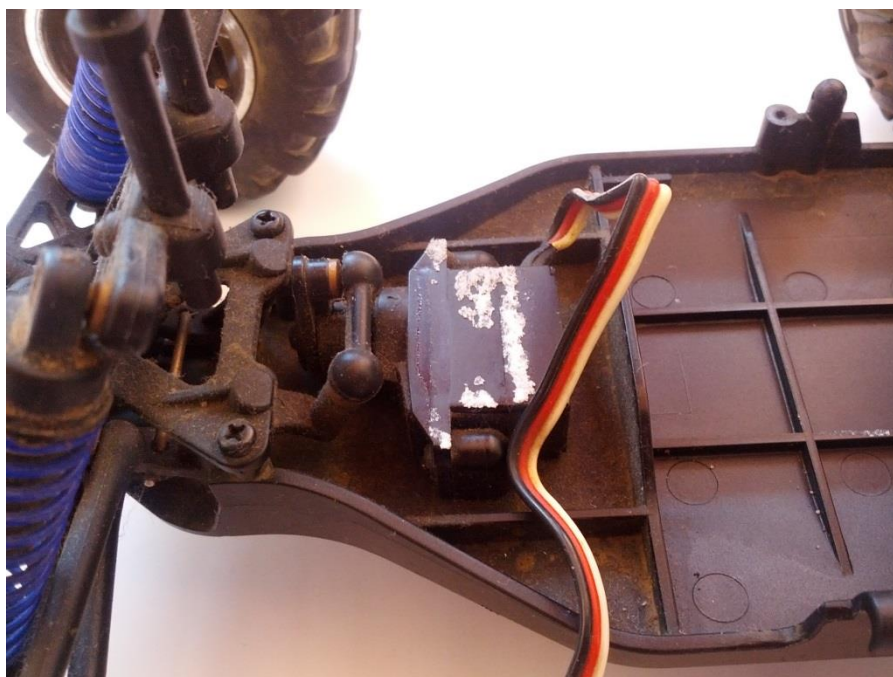
Η μη ανταπόκριση του πρώτου οχήματος μας ανάγκασε να μπούμε στη διαδικασία ανεύρεσης ενός άλλου οχήματος που θα πληρούσε τις παρακάτω προϋποθέσεις:

- Να διαθέτει μοτέρ χαμηλών στροφών ώστε η ταχύτητα του διατηρείται σε χαμηλά επίπεδα
- Η ποιότητα κατασκευής του να είναι σε επίπεδα που θα επιτρέπουν την όσο πιο ομαλή και σταθερή κίνηση του χωρίς να παρουσιάζει απρόβλεπτες αποκλίσεις στην διεύθυνση πορείας
- Οι εμπρόσθιοι τροχοί να ελέγχονται από σερβοκινητήρα ώστε να καταστεί εφικτή η στρέψη του οχήματος κατά όσες μοίρες επιθυμούμε και κατ'επέκταση να είναι υλοποιήσιμη η ενδεχόμενη διόρθωση πορείας του. Επιπλέον μέσω του σερβοκινητήρα η στρέψη του τροχού θα γίνεται με μεγάλη ακρίβεια και σταθερότητα, και η δυνατότητα ρύθμισης του ( calibration) θα εξασφαλίσει την ευθεία πορεία του οχήματος.

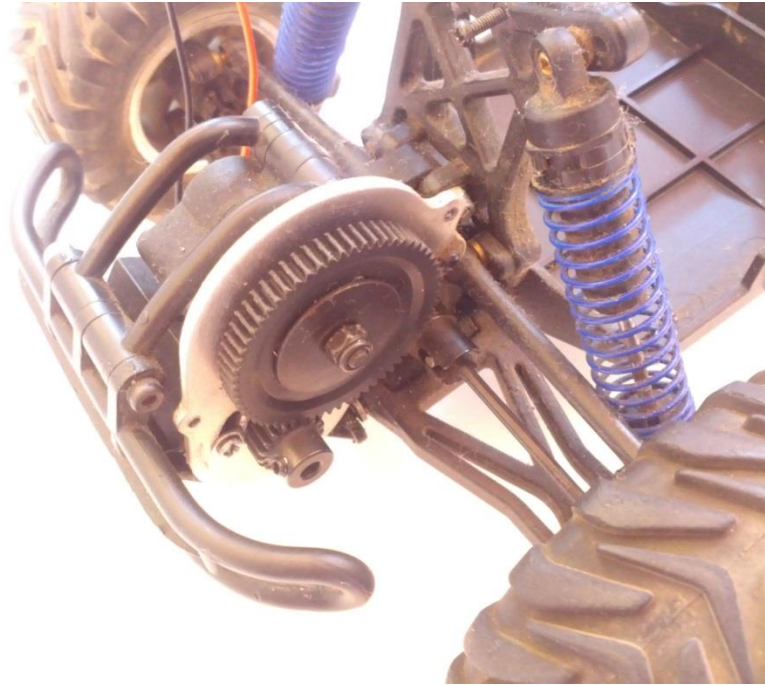
Έτσι λοιπόν μονόδρομος ήταν ένα τηλεκατευθυνόμενο όχημα που ανήκει στην κατηγορία μοντελισμού, το οποίο φαίνεται στην εικόνα που ακολουθεί.



Εικόνα 2.4: 2<sup>ο</sup> τηλεκατευθυνόμενο όχημα



Εικόνα 2.5: Σερβοκινητήρας ελέγχου εμπρόσθιων τροχών

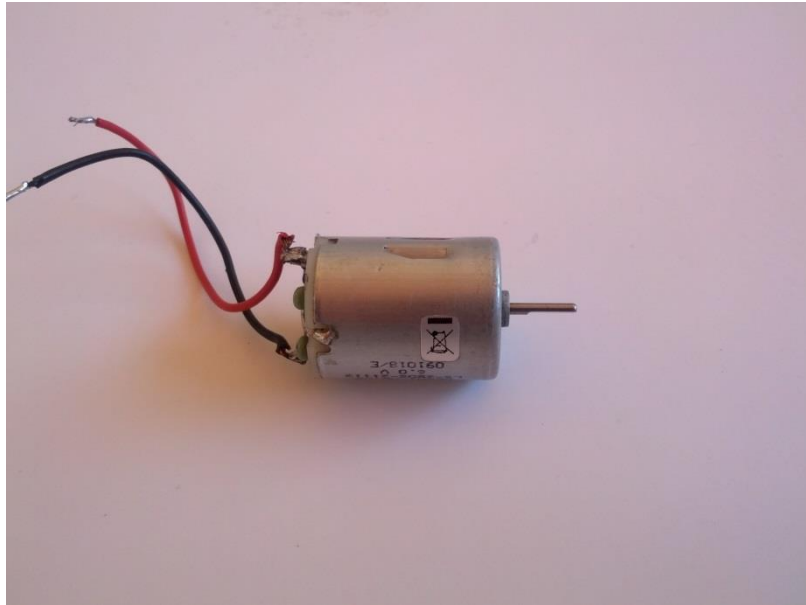


Εικόνα 2.6: Σύστημα μετάδοσης κίνησης / Υποδοχή μοτέρ

Το αρχικό ονομαστικό motor του εν λόγω μοντέλου εγκαταλείφθηκε αμέσως καθώς λειτουργούσε σε αρκετά υψηλές στροφές. Αντικαταστάθηκε αρχικά με το LS-280S-20201 της εταιρείας Leshi Motor με μέγιστη ταχύτητα τις 6300 στροφές/λεπτό. Αναλυτικά τα χαρακτηριστικά του φαίνονται στο σχήμα 2.1.

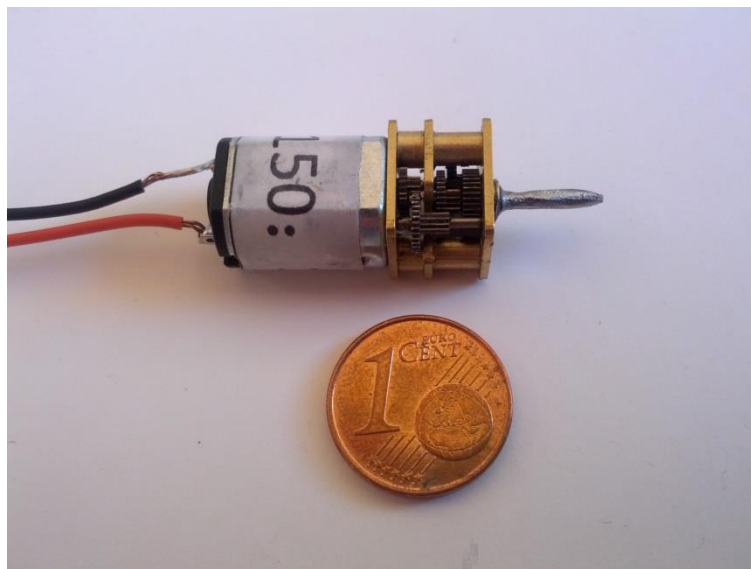
MODEL	VOLTAGE		NO LOAD		AT MAXIMUM EFFICIENCY				STALL			
	OPERATING RANGE	NOMINAL	SPEED	CURRENT	SPEED	CURRENT	TORQUE		OUTPUT	TORQUE		CURRENT
			RPM	A	RPM	A	g.cm	mN.m	W	g.cm	mN.m	A
LS-280SR-20201	3.6-4.5	3.6V	6300	0.16	5257	0.61	30.9	3.03	1.67	186.6	18.29	4.09

Σχήμα 2.1: Προδιαγραφές LS-280S-20201



Εικόνα 2.7: Leshi Motor LS-280S-20201

Η ταχύτητα του LS-280S ήταν μεν σημαντικά μικρότερη από το ονομαστικό motor του μοντέλου αλλά δεν κρίθηκε απόλυτα ιδανική για την εφαρμογή. Έγινε μείωση στροφών και πάλι με χρήση PWM σήματος και ρυθμιστή τάσης, το motor κινούσε το όχημα όμως με μεγάλες διακυμάνσεις στο ρυθμό επιτάχυνσης. Τελικά το motor που χρησιμοποιήθηκε στην τελική υλοποίηση είναι ένα Micro Gearmotor με μέγιστη ταχύτητα τις 270 στροφές/λεπτό. Τα motor αυτά έχουν πολύ σκληρή λειτουργία και ενσωματώνουν ένα μεταλλικό μικρο-γρανάζωμα. Έχουν σχέση ταχύτητας 100:1, λειτουργούν σε ονομαστική τάση 12V, μέγιστη ροπή σε μηδενικές στροφές 25 oz-in. Αναλυτικά τα χαρακτηριστικά του παρουσιάζονται παρακάτω.



Εικόνα 2.8: Micro Gearmotor



Προδιαγραφές:

- Voltage: 6 - 12 Volts
- Gear Ratio: 100:1
- Stall Torque: 12/25 oz-in. (6/12V)
- Speed: 140/270 RPM (6/12V)
- No Load Current: 40/80 mA (6/12V)
- Stall Current: 360/1600 mA (6/12V)
- DC Reversible

Διαστάσεις:

- Motor Size: 26 x 12 x 10mm
- Shaft Size: 3mm Diameter x 10mm Length
- Wire Length: 6"
- Weight: 17g

#### 2.3.1.4 Κάμερα

Η αρχική επιλογή της κάμερας έγινε με τα εξής κριτήρια:

- Να είναι οικονομική
- Να συνδυάζει τα απαιτούμενα χαρακτηριστικά για τη λειτουργία του συστήματος (επαρκής ρυθμός καταγραφής καρέ, επαρκής ανάλυση)
- Να έχει ευρυγώνιο φακό

Έτσι αρχικά χρησιμοποιήθηκε η PS3 EYE CAMERA, τα τεχνικά χαρακτηριστικά της οποίας είναι τα εξής:

Μέγιστη ανάλυση (βίντεο)	640x480
Ρυθμός καταγραφής βίντεο (μέγιστο)	120 καρέ/δευτερόλεπτο
Αισθητήρας εικόνας	CMOS
Ευρυγώνιος φακός	Ναι
Οπτικό πεδίο	75 μοίρες
Ελάχιστη απόσταση εστίασης	9,84 ίντσες
Ακύρωση θορύβου	Ναι
Αναγνώριση προσώπου	Ναι

Συνδεσιμότητα	USB
---------------	-----



Εικόνα 2.9: PS3 EYE CAM

Στις δοκιμές παρατηρήθηκε ότι ο αισθητήρας της PS3 EYE υστερεί στην αποκωδικοποίηση του χρώματος με αποτέλεσμα να υπάρχουν σημαντικές αλλοιώσεις σε σχέση με το πραγματικό χρώμα. Συγκεκριμένα ο αισθητήρας της έχει την τάση να «γκριζάρει» όλα τα χρώματα και προκειμένου να λειτουργήσει ο αλγόριθμος ανίχνευσης χρώματος έπρεπε να εξασφαλίσουμε ότι η πίστα θα φωτίζεται ομοιόμορφα σε όλη την επιφάνεια της και ιδίως στις γωνίες, πράγμα που είναι αρκετά δύσκολο υπό μεταβαλλόμενες συνθήκες. Στην τελική υλοποίηση χρησιμοποιήθηκε η Microsoft LifeCam Cinema η οποία ναι μεν καταγράφει με μικρότερο ρυθμό (μέχρι 30 καρά/δευτερόλεπτο γεγονός που δεν δημιούργησε πρόβλημα στη λειτουργία του συστήματος) ωστόσο ο αισθητήρας της αποκωδικοποιεί το χρώμα με πολύ μεγαλύτερη πιστότητα.



Εικόνα 2.10: Microsoft LifeCam Cinema

Τα τεχνικά χαρακτηριστικά της Microsoft LifeCam Cinema είναι:

Αισθητήρας	CMOS
Καταγραφή βίντεο	720p (1280x720 pixels)
Ρυθμός καταγραφής	30 καρέ/δευτερόλεπτο
Οπτικό πεδίο	73 μοίρες
Αυτόματη εστίαση	Ναι
Χειροκίνητη εστίαση	Ναι

### 2.3.1.5 Modem – Router – Access point

Χρησιμοποιήθηκε το Thomson TG585v8 ως router – access point για τη σύνδεση της πλατφόρμας Arduino στο τοπικό ασύρματο δίκτυο. Η μεταφορά πληροφορίας από προσωπικό υπολογιστή προς και από Arduino γίνεται με χρήση της τεχνολογίας WiFi μέσω του δρομολογητή/σημείου πρόσβασης Thomson.



Εικόνα 2.11: Thomson TG585

### 2.3.1.6 Προσωπικός υπολογιστής

Στον υπολογιστή βρίσκεται η εφαρμογή που εκτελεί ο χρήστης. Στο interface που έχει σχεδιαστεί περιλαμβάνεται (1) η έξοδος του αλγορίθμου εντοπισμού με τη μορφή βίντεο στο οποίο υπάρχει marker γύρω από το αντικείμενο που έχει ανιχνευθεί και (2) ένα εικονικό

σχέδιο της πίστας και των σταθερών σημείων μαζί με τα buttons επιλογής σημείου προορισμού, στο οποίο αποτυπώνεται σε πραγματικό χρόνο η κίνηση του οχήματος.

### **2.3.2 Λογισμικό**

Το λογισμικό που αναπτύχθηκε ουσιαστικά αποτελείται από τον κώδικα προγραμματισμού της πλατφόρμας Arduino και τον κώδικα της εφαρμογής που εκτελείται στον υπολογιστή του χρήστη. Η εν λόγω εφαρμογή αναπτύχθηκε σε γλώσσα προγραμματισμού Java στο ολοκληρωμένο περιβάλλον ανάπτυξης Eclipse IDE Enterprise Edition και υλοποιεί συνοπτικά τις παρακάτω διαδικασίες:

- τον εντοπισμό του οχήματος εφαρμόζοντας στην ακολουθία εικόνων από την κάμερα τον αλγόριθμο εντοπισμού αντικειμένου
- την αποθήκευση αυτής της θέσης σε πραγματικό χρόνο σε ένα web service που εκτελείται σε apache server
- τη λογική της δρομολόγησης του οχήματος
- τη σχεδίαση του περιβάλλοντος αλληλεπίδρασης με το χρήστη

# 3

## *Μέθοδοι εντοπισμού αντικειμένων σε εσωτερικό χώρο*

### *3.1 Εισαγωγή*

Ο εντοπισμός αντικειμένων στον χώρο της υπολογιστικής όρασης αποτελεί τη διαδικασία εύρεσης και αναγνώρισης αντικειμένων σε μια εικόνα ή ένα βίντεο. Ο άνθρωπος έχει την ικανότητα να αναγνωρίζει πλήθος αντικειμένων σε εικόνες καταβάλλοντας πολύ μικρή προσπάθεια παρ'ότι η όψη, το μέγεθος ή η κλίμακα των αντικειμένων μπορεί να διαφοροποιείται ανάλογα με την οπτική γωνία, να περιστρέφονται, ή τμήμα τους να μην ανήκει στο οπτικό του πεδίο. Αν και οι υπολογιστές έχουν σημαντικά αποτελέσματα όσον αφορά την ταχύτητα της επεξεργασίας πολύπλοκων δεδομένων, την αποθήκευση και γρήγορη ανάκτηση μεγάλου όγκου πληροφοριών, αδυνατούν να προσεγγίσουν τα ποιοτικά χαρακτηριστικά που διαθέτει ο ανθρώπινος εγκέφαλος για να εκτελέσει τη διαδικασία της παρακολούθησης και αναγνώρισης της τροχιάς ενός αντικειμένου. Το έργο αυτό αποτελεί ακόμα μια πρόκληση για τα συστήματα υπολογιστικής όρασης και τις τελευταίες δεκαετίες έχουν γίνει πολλές προσπάθειες να αναπτυχθεί η τεχνολογία που θα προσομοιώνει την ανθρώπινη αντίληψη αναγνώρισης της κίνησης [12].

Τα υπολογιστικά συστήματα εξομοιώνουν την αίσθηση της όρασης μέσω της καταγραφής και ανάλυσης ακολουθιών εικόνων που προκύπτουν από μία ή περισσότερες κάμερες. Καλούνται να εκτελέσουν μια σειρά από σύνθετες λειτουργίες, ειδικότερα όταν η πληροφορία προέρχεται από πολλαπλές κάμερες όποτε και ο εντοπισμός είναι ακριβέστερος. Ένα σύνθετο σύστημα καμερών σε σχέση με μία σταθερή κάμερα εξασφαλίζει μεν μεγαλύτερο βάθος πεδίου και ευρύτερο οπτικό πεδίο αλλά η υλοποίηση του δεν είναι πάντα εύκολη υπόθεση.

Για να εξαχθεί κάποιο αποτέλεσμα από το υπολογιστικό σύστημα θα πρέπει να οριοθετηθεί, να κατηγοριοποιηθεί και να χαρακτηριστεί η πληροφορία που λαμβάνει σαν είσοδο. Τα δεδομένα εισόδου είναι ουσιαστικά μήτρες εικονοστοιχείων (pixels) με τιμές στο χώρο των χρωμάτων, και η επεξεργασία που υφίστανται εξαρτάται άμεσα από τα επιθυμητά αποτελέσματα εξόδου. Οι διαδικασίες που εκτελεί ένα σύστημα εντοπισμού αντικειμένων είναι πολλαπλές και διαφοροποιημένες ανάλογα με την εφαρμογή και την τρέχουσα υλοποίηση. Π.χ η διαδικασία που προσδιορίζει την μετατόπιση του αντικειμένου από εικόνα σε εικόνα είναι τελείως διαφορετική από τη διαδικασία που συγκεντρώνει την πληροφορία που καθορίζει και χαρακτηρίζει ένα αντικείμενο. Επίσης διαφορετική διαδικασία θα ακολουθηθεί για τον εντοπισμό της τροχιάς ενός αντικειμένου.

Τα βασικά χαρακτηριστικά μια εικόνας που επεξεργάζονται από εφαρμογές εντοπισμού αντικειμένων είναι τα εξής:

- Χρώμα: Είναι πληροφορία που μπορεί να εξαχθεί άμεσα από την εικόνα και να μοντελοποιηθεί με διάφορους τρόπους.
- Υφή: Είναι ένα μέγεθος που συνεισφέρει κυρίως στους υπολογισμούς εκτίμησης της κίνησης.
- Σχήμα: Θεωρείται χαρακτηριστικό υψηλότερου επιπέδου, ωστόσο μοντελοποιείται σε χαμηλό επίπεδο. Προσδιορίζεται βάσει των ακμών, του εμβαδού, της θέσης κλπ. Μια ακόμα παράμετρος που βοηθάει στην εξαγωγή του σχήματος είναι το περίγραμμα το οποίο όμως μοντελοποιείται σε υψηλότερο επίπεδο.
- Κίνηση: Εξάγεται από δύο ή περισσότερα καρέ ενός βίντεο. Παρέχει πληροφορία που χρησιμοποιείται σαν βάση για πολλές άλλες εφαρμογές όπως ο υπολογισμός της τρισδιάστατης κίνησης ενός αντικειμένου και ο εντοπισμός της τροχιάς του.
- Πρότυπα: Η εξαγωγή προτύπων από ακολουθίες εικόνων είναι μια αρκετά δημοφιλής διαδικασία στο χώρο των εφαρμογών εντοπισμού. Τα πρότυπα αφού εξαχθούν χρησιμοποιούνται αυτούσια, μετασχηματισμένα ή και παραμορφωμένα σε διαδικασίες εντοπισμού και αναγνώρισης.

- Μετασχηματισμοί: Ο μετασχηματισμός των εικόνων ή συγκεκριμένων περιοχών τους παρέχει αρκετά ενδιαφέρουσα πληροφορία . Να αναφέρουμε για παράδειγμα το μετασχηματισμό Fourier που εξάγει το συχνοτικό περιεχόμενο των εικόνων.

### 3.2 Ανίχνευση και παρακολούθηση αντικειμένων

Η παρακολούθηση αντικειμένων(object tracking) αποτελεί μία αρκετά δημοφιλή διαδικασία που χρησιμοποιείται σε πολλές εφαρμογές της πληροφορικής. Η ραγδαία αύξηση των υπολογιστών υψηλής ισχύος, η διαθεσιμότητα οικονομικών και υψηλής ποιότητας καμερών και η όλο και αυξανόμενη ανάγκη για αυτοματοποιημένη ανάλυση βίντεο έχει δημιουργήσει μεγάλο ενδιαφέρον στον τομέα του εντοπισμού αντικειμένων. Η παρακολούθηση αντικειμένων ορίζεται σαν η διαδικασία διαχωρισμού ενός αντικειμένου από μια ακολουθία εικόνων (video) και παρατήρησης της κίνησης και του προσανατολισμού του , με σκοπό την εξαγωγή χρήσιμης πληροφορίας. Η πληροφορία αυτή χρησιμοποιείται κυρίως σε εφαρμογές υπολογιστικής όρασης όπως στην παρακολούθηση της κυκλοφοριακής συμφόρησης, στην απομακρυσμένη αυτοματοποιημένη παρακολούθηση χώρων ή στην παρακολούθηση ανθρώπων. Το πρώτο στάδιο είναι η *ανίχνευση του αντικειμένου*, η οποία περιλαμβάνει την αναγνώριση του αντικειμένου (σιλουέτας του) αλλά και την ιδιότητα του να κινείται μέσα στον παρατηρούμενο χώρο. Το δεύτερο στάδιο είναι η *παρακολούθηση του αντικειμένου* που έχει ανιχνευτεί σε μια ακολουθία εικόνων και η ανάλυση της τροχιάς του με σκοπό τον προσδιορισμό της συμπεριφοράς του [13].

#### Ανίχνευση (detection)

Για να επιτευχθεί η ανίχνευση πρέπει να υλοποιηθούν δύο διαδικασίες, η *κατάτμηση της κίνησης* και η *αναγνώριση του αντικειμένου*. Η κατάτμηση της κίνησης ουσιαστικά εξασφαλίζει το διαχωρισμό του κινούμενου αντικειμένου από την εικόνα του φόντου [14]. Τα αποτελέσματα των αλγορίθμων κατάτμησης της κίνησης δεν πρέπει να επηρεάζονται από το θόρυβο, από αλλαγές στο φόντο ή αλλαγές στο φωτισμό. Παρακάτω παρουσιάζονται συνοπτικά κάποιες σύγχρονες τεχνικές:

- *Αφαίρεση φόντου* (background subtraction): Είναι μια ευρέως χρησιμοποιούμενη τεχνική για τη εξαγωγή της εικόνας (foreground mask) που περιέχει μόνο τα pixels των κινούμενων αντικειμένων. Για τον υπολογισμό του foreground mask αφαιρείται από το τρέχον καρέ το μοντέλο φόντου που περιέχει ουσιαστικά το στατικό μέρος του βίντεο ή γενικότερα οτιδήποτε μπορεί να θεωρηθεί ως φόντο δεδομένου των χαρακτηριστικών της σκηνής.

- *Χρονική διαφορά* (temporal difference): Βασίζεται στη χρονική διαφορά μεταξύ διαδοχικών καρέ για την εξαγωγή του κινούμενου αντικειμένου. Αυτή η μέθοδος είναι εύκολα υλοποιήσιμη και προσαρμόζεται εύκολα σε δυναμικά περιβάλλοντα.
- *Οπτική ροή* (optical flow): Βασική προϋπόθεση για να λειτουργήσει αυτή η τεχνική είναι να μην αλλάζει η ένταση των εικονοστοιχείων σε μια ακολουθία εικόνων και η κίνηση των εσωτερικών χαρακτηριστικών του καρέ να είναι μικρή. Είναι δύσκολα υλοποιήσιμη και συχνά απαιτεί εξειδικευμένο εξοπλισμό. Προσφέρει όμως τη δυνατότητα διαχωρισμού κινούμενων αντικειμένων ακόμα και αν η κάμερα κινείται.
- *Στατιστικές μέθοδοι* (statistical methods): Βασίζονται στον υπολογισμό στατιστικών μεγεθών μεμονωμένων εικονοστοιχείων ή ομάδων αυτών, και στη χρήση αυτής της πληροφορίας για να κατηγοριοποιήσουν περιοχές μιας εικόνας ως περιοχές που ανήκουν στο φόντο ή στο προσκήνιο.

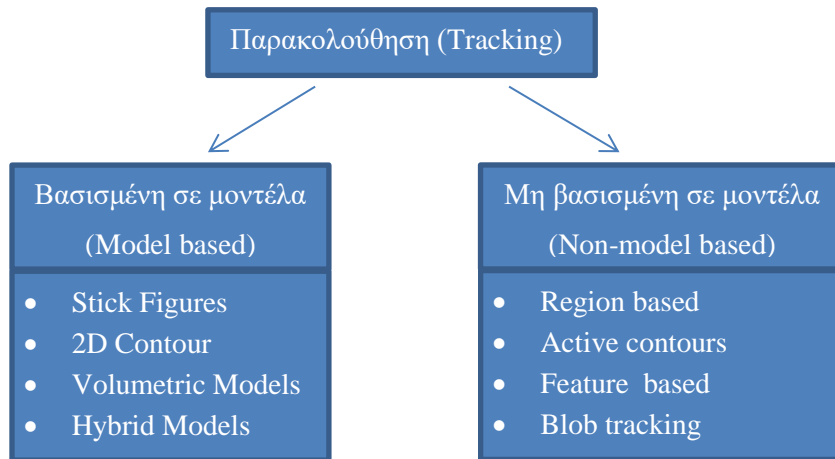
Η αναγνώριση του αντικειμένου είναι χρήσιμη στις περιπτώσεις που έχουμε πολλά προς εντοπισμό αντικείμενα. Τα αντικείμενα αναγνωρίζονται με βάση δύο κριτήρια: το σχήμα και το είδος της κίνησης που εκτελούν. Οι δύο βασικές τεχνικές είναι οι εξής [12]:

- *Αναγνώριση με βάση το σχήμα* (Shape based identification): Χρησιμοποιείται κυρίως για το διαχωρισμό αντικειμένων με διαφορετικό σχήμα. Η ταξινόμηση του αντικειμένου γίνεται με βάση παραμέτρους όπως η αναλογία δύο διαστάσεων ή η θέση των ακραίων σημείων.
- *Αναγνώριση με βάση την περιοδικότητα* (periodicity based identification): Διαφοροποιεί κυρίως αντικείμενα του ίδιου είδους που παρουσιάζουν περιοδικότητα στην κίνηση τους. Για παράδειγμα η περιοδικότητα της κίνησης των χεριών ή των ποδιών μας μπορεί να χρησιμοποιηθεί ως χαρακτηριστικό για τη κατηγοριοποίηση ανθρώπων βάσει του τρόπου βάδισης.

### Παρακολούθηση

Μετά την κατάτμηση και την αναγνώριση του αντικειμένου, είναι συνήθως αναγκαία και η παρακολούθηση του. Η παρακολούθηση ενός αντικειμένου βασίζεται στην εξαγωγή των συναφών σχέσεων μεταξύ χαρακτηριστικών της εικόνας σε συνεχόμενα καρέ, όπως το χρώμα, η υφή, η ταχύτητα και η θέση. Όπως φαίνεται και στο παρακάτω σχήμα οι αλγόριθμοι παρακολούθησης κατηγοριοποιούνται με βάση το αν χρησιμοποιούν μοντέλα σχήματος για την παρακολούθηση του αντικειμένου.





Σχήμα 3.1: Αλγόριθμοι παρακολούθησης αντικειμένων

Οι αλγόριθμοι που είναι βασισμένοι σε μοντέλα χρησιμοποιούν ένα προκαθορισμένο μοντέλο του αντικειμένου για παρακολούθηση. Βρίσκουν χρήση ιδίως σε εφαρμογές που παρακολουθούν τον άνθρωπο ή τμήματα του σώματος όπως τα χέρια ή το πρόσωπο. Η δομή και η κίνηση του μοντέλου προσδιορίζεται βάσει των χαρακτηριστικών της εικόνας. Τέτοιες διαδικασίες ωστόσο είναι υπολογιστικά πολύπλοκες και απαιτούν ιδανική κατάτμηση του αντικειμένου από το φόντο.

Αντίθετα στις μεθόδους που χρησιμοποιούν τη μη βασισμένη σε μοντέλα παρακολούθηση είναι αναγκαία η εξαγωγή μόνο κάποιων σημαντικών χαρακτηριστικών του αντικειμένου και όχι ολόκληρης της δομής του, πράγμα το οποίο μειώνει κατά πολύ την υπολογιστική πολυπλοκότητα. Στον **feature-based αλγόριθμο** ως χαρακτηριστικά για την παρακολούθηση χρησιμοποιούνται παράμετροι όπως η χρωματική πληροφορία, οι γωνίες ή η υφή. Υπάρχουν δύο διαφορετικές υλοποιήσεις του αλγορίθμου: η δυναμική και η στατική παρακολούθηση. Στην στατική παρακολούθηση οι παράμετροι εξάγονται εκ των προτέρων από κάθε καρέ και ο αλγόριθμος υπολογίζει τη βέλτιστη αντιστοιχία μεταξύ τους. Στην δυναμική παρακολούθηση οι παράμετροι προσδιορίζονται δυναμικά στα διαδοχικά καρέ, υπολογίζοντας την κίνηση τους σε ένα καρέ και εκτιμώντας για αυτή στο επόμενο καρέ. Ένας ευρέως γνωστός αλγόριθμος είναι ο KLT αλγόριθμος (Kanade-Lucas Tracking algorithm). Ένα χαρακτηριστικό αυτής της μεθόδου είναι η σχέση μεταξύ της δυσκολίας εξαγωγής των χαρακτηριστικών και της αποδοτικότητας του αλγορίθμου. Για παράδειγμα είναι εύκολο να γίνει εξαγωγή της θέσης των ακμών αλλά δύσκολο να παρακολουθηθούν, καθώς δεν είναι εύκολο να προσδιοριστεί μια αντιστοιχία ένα προς ένα μεταξύ τους.

Στην τελική υλοποίηση έγινε εφαρμογή μιας feature based (βασισμένη σε χαρακτηριστικά) τεχνικής γνωστής ως blob tracking και ως παράμετρος για την παρακολούθηση χρησιμοποιήθηκε η χρωματική πληροφορία. Το προς εντοπισμό αντικείμενο είναι ένα χρωματιστό χαρτόνι που έχει τοποθετηθεί πάνω στο τηλεκατευθυνόμενο όχημα. Η εν λόγω τεχνική θα παρουσιαστεί στην ενότητα που ακολουθεί.

### 3.3 Μια πρώτη προσέγγιση στην ανίχνευση χρώματος

Η πρώτη γνωριμία με το χώρο της υπολογιστικής όρασης πραγματοποιήθηκε μέσω της γλώσσας Processing. Η **Processing** είναι μία ανοικτού κώδικα γλώσσα προγραμματισμού και παρέχει ένα περιβάλλον ανάπτυξης για άτομα που θέλουν να δημιουργήσουν εικόνες, κινούμενα σχέδια και διάφορες αλληλεπιδράσεις [15]. Αρχικά αναπτύχθηκε σαν ένα σχεδιαστικό πρόγραμμα και για να διδάξει βασικές αρχές προγραμματισμού μέσα σε ένα οπτικό πλαίσιο, αλλά στη συνέχεια εξελίχθηκε σε ένα εργαλείο δημιουργίας ολοκληρωμένων επαγγελματικών εργασιών. Αυτή την στιγμή υπάρχουν δεκάδες χιλιάδες σπουδαστές, καλλιτέχνες, σχεδιαστές, ερευνητές και χομπίστες που χρησιμοποιούν τη Processing για διδασκαλία, προτυποποίηση και παραγωγή.

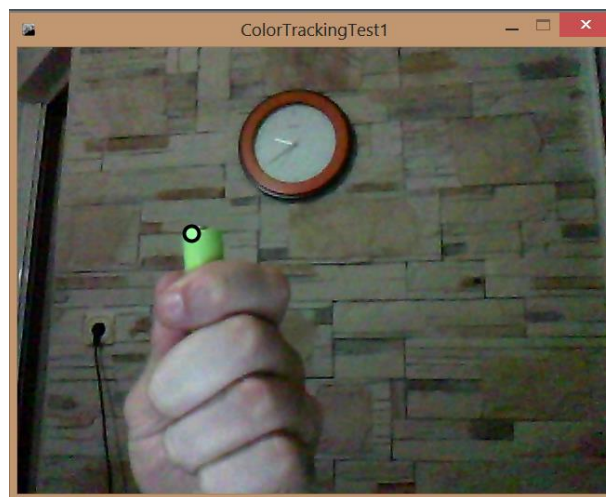
Μερικά από τα χαρακτηριστικά της Processing είναι:

- Είναι ελεύθερο/ανοικτό λογισμικό με άδεια χρήσης GPL/LGPL.
- Είναι πολυπλατφορμική, μπορεί να τρέξει σε λειτουργικά συστήματα GNU/Linux, Mac OS X και Windows.
- Δημιουργεί διαδραστικά προγράμματα χρησιμοποιώντας δισδιάστατα (2D) ή τρισδιάστατα (3D) γραφικά.
- Ενσωμάτωση της OpenGL για επιτάχυνση 3D.
- Δημιουργία stand-alone desktop εφαρμογών και web-based εφαρμογών (applets).
- Υπάρχουν αρκετές βιβλιοθήκες επέκτασης της γλώσσας, για εφαρμογές ήχου, βίντεο, τεχνητής όρασης, κ.α

Η Processing βασίστηκε στη δυνατότητες γραφικών της γλώσσας προγραμματισμού Java, απλοποιώντας την χρήση και δημιουργώντας νέα χαρακτηριστικά. Διαθέτει το δικό της περιβάλλον ανάπτυξης (IDE) το οποίο μάλιστα είναι σχεδόν όμοιο με αυτό της πλατφόρμας Arduino, πράγμα που δεν προκαλεί έκπληξη αφού η Wiring και το Arduino είναι δύο από τα σημαντικότερα έργα που χρησιμοποιούν τη γλώσσα και το περιβάλλον της Processing.

Υπάρχουν δυο βασικές συναρτήσεις σε ένα τυπικό πρόγραμμα της Processing. Η συνάρτηση `setup()` εκτελείται στην αρχή του προγράμματος και για μία μόνο φορά. Χρησιμοποιείται για τις αρχικοποιήσεις ιδιοτήτων και βιβλιοθηκών, όπως για παράδειγμα: το μέγεθος της οθόνης, το χρώμα φόντου, το φόρτωμα εικόνων κ.α. Οι μεταβλητές που δηλώνονται στη `setup()` δεν είναι προσβάσιμες από άλλες συναρτήσεις, συμπεριλαμβανομένης και της `draw()` που θα εξηγήσουμε αμέσως μετά. Η συνάρτηση `draw()` εκτελείται αμέσως μετά την `setup()` και ο κώδικας που γράφεται μέσα στη συνάρτηση αυτή επαναλαμβάνεται συνεχώς μέχρι να τερματιστεί η εφαρμογή ή μέχρι να κληθεί η συνάρτηση `noLoop()`.

### Μια απλή εφαρμογή ανίχνευσης χρώματος ανεπτυγμένη σε γλώσσα processing



Εικόνα 3.1: Υλοποίηση ανίχνευσης χρώματος σε γλώσσα Processing

Ο αλγόριθμος ουσιαστικά αναλύει ένα προς ένα τα εικονοστοιχεία (pixels) του κάθε καρέ του βίντεο που καταγράφει η κάμερα και υπολογίζει τη διαφορά της του κάθε εικονοστοιχείου στο χώρο RGB (Red,Green,Blue) σε σχέση με το προς ανίχνευση χρώμα. Κατά τη διάρκεια του κύκλου σάρωσης του καρέ η ελάχιστη τιμή αυτής της διαφοράς όπως και η θέση του αντίστοιχου εικονοστοιχείου αποθηκεύονται στις μεταβλητές `minD` και `cX`, `cY` αντίστοιχα. Οι τιμές των μεταβλητών θέσης του εν λόγω εικονοστοιχείου (`cX`, `cY`) όπως έχουν διαμορφωθεί μετά την ολοκλήρωση του κύκλου σάρωσης του καρέ αντιπροσωπεύουν τη θέση του προς ανίχνευση χρώματος. Αν η εν λόγω διαφορά είναι μικρότερη από ένα κατώφλι που καθορίζουμε εμείς θεωρούμε ότι ο εντοπισμός του χρώματος έχει πετύχει. Η εφαρμογή αναπτύχθηκε στο περιβάλλον Processing 2.0b8 32bit. Ακολουθεί ο κώδικας της εμπλουτισμένος με τα απαραίτητα σχόλια.

```

//Το απαραίτητο import της βιβλιοθήκης video
import processing.video.*;

//Η συσκευή καταγραφής video
Capture video;

//Το χρώμα που θέλουμε να ανιχνεύσουμε
color trackColor;

void setup() {
  size(640,480);
  video = new Capture(this,width,height,"Chicony USB 2.0 Camera",30);
  //Αρχικοποίηση του προς ανίχνευση χρώματος
  trackColor = color(255,0,0);
  smooth();
  video.start();
}

void draw() {
  //Καταγραφή και προβολή video
  if (video.available()) {
    video.read();
  }
  video.loadPixels();
  image(video,0,0);

  //Πριν ξεκινήσει η σάρωση του κάθε καρέ η μεταβλητή της
  //ελάχιστης διαφοράς με το προς ανίχνευση χρώμα
  //αρχικοποιείται σε τιμή 500 (αρκετά μεγάλη ώστε να
  //βρεθεί εύκολα το πρώτο pixel)
  float minD = 500;

  //Οι συντεταγμένες (x,y) του εικονοστοιχείου που πλησιάζει
  //περισσότερο το προς ανίχνευση χρώμα

```

```

int cX = 0;
int cY = 0;

//Ο βρόχος σάρωσης του καρέ pixel προς pixel
for (int x = 0; x < video.width; x ++ ) {
    for (int y = 0; y < video.height; y ++ ) {
        int loc = x + y*video.width;
        //Υπολογισμός της τιμής του χρώματος
        //στο χώρο RGB για το τρέχον pixel
        color currentColor = video.pixels[loc];
        float r1 = red(currentColor);
        float g1 = green(currentColor);
        float b1 = blue(currentColor);
        float r2 = red(trackColor);
        float g2 = green(trackColor);
        float b2 = blue(trackColor);

        //Χρήση της ευκλείδειας απόστασης για τον
        //υπολογισμό της διαφοράς δύο χρωμάτων
        //Χρησιμοποιούμε τη συνάρτηση dist() για να
        //συγκρίνουμε το τρέχον χρώμα με το χρώμα που
        //θέλουμε να εντοπίσουμε
        float d = dist(r1,g1,b1,r2,g2,b2);

        //Αν το τρέχον χρώμα είναι πλησιέστερο στο
        //προς ανίχνευση χρώμα σε σχέση με το ήδη
        //αποθηκευμένο, αποθήκευσε θέση τρέχοντος
        //pixel και τρέχουσα διαφορά
        if (d < minD) {
            minD = d;
            cX = x;
            cY = y;
        }
    }
}

```

```

//Θεωρούμε πως έχουμε πετύχει τον εντοπισμό
//αν η διαφορά του πλησιέστερου εικονοστοιχείου
//είναι μικρότερη από 10. Το κατώφλι αυτό μπορεί
//να προσαρμοστεί ανάλογα με το πόσο ακριβές
//θέλουμε να είναι ο εντοπισμός.
if (worldRecord < 10) {
    // Σχεδιασμός κύκλου στο σημείο που έχει
    // εντοπιστεί το χρώμα
    fill(trackColor);
    strokeWeight(4.0);
    stroke(0);
    ellipse(closestX,closestY,16,16);
}
}

void mousePressed() {
    // Ο event handler που αποθηκεύει τη τιμή
    // του χρώματος του σημείου που κλικάρουμε
    // με το ποντίκι στη μεταβλητή trackColor
    int loc = mouseX + mouseY*video.width;
    trackColor = video.pixels[loc];
}

```

### ***3.4 Ανίχνευση μεγάλου δυαδικού αντικειμένου (Blob tracking)***

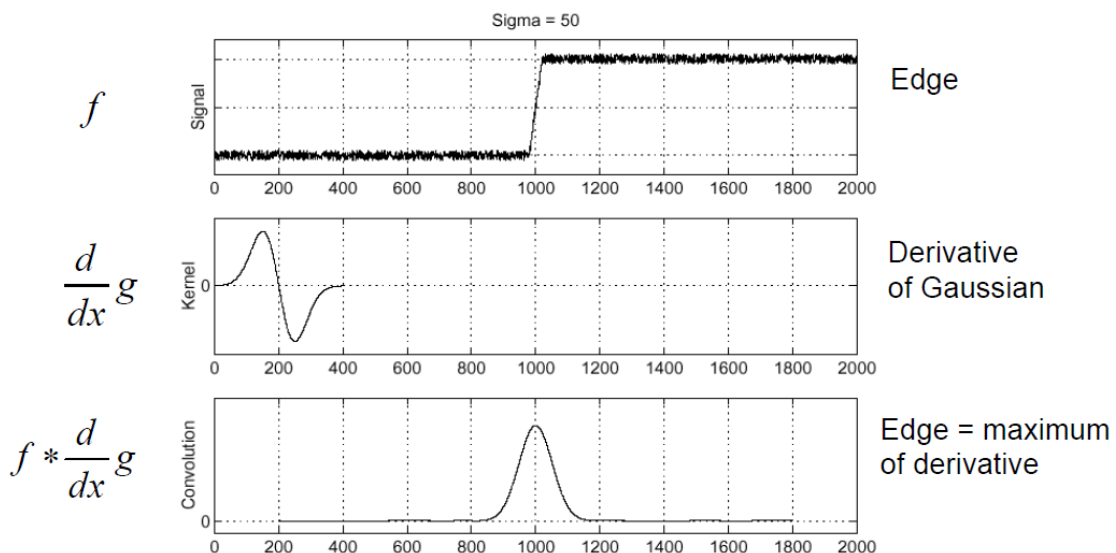
Στον χώρο της υπολογιστικής όρασης, η ανίχνευση blob αναφέρεται στις μεθόδους που αποσκοπούν στον εντοπισμό περιοχών σε μια ψηφιακή εικόνα που διαφέρουν ως προς τα χαρακτηριστικά (όπως το χρώμα ή η φωτεινότητα) σε σχέση με περιοχές που τις περιβάλλουν. Ουσιαστικά το blob είναι μια περιοχή της εικόνας της οποίας τα χαρακτηριστικά είναι σταθερά ή λαμβάνουν ένα προκαθορισμένο εύρος τιμών. Όλα τα

σημεία από τα οποία αποτελείται ένα blob θεωρούνται όμοια ως προς ένα ή περισσότερα χαρακτηριστικά. Θεωρώντας ότι η παράμετρος που μας ενδιαφέρει είναι εκφρασμένη ως συνάρτηση της θέσης στην ψηφιακή εικόνα, υπάρχουν δύο βασικές προσεγγίσεις στον εντοπισμό του blob [16][17]:

- Διαφορικές μέθοδοι: Βασίζονται σε παραγώγους της συνάρτησης του χαρακτηριστικού ως προς τη θέση.
- Μέθοδοι που βασίζονται σε τοπικά ακρότατα: Βασίζονται στην εύρεση του μεγίστου και του ελαχίστου της συνάρτησης του χαρακτηριστικού.

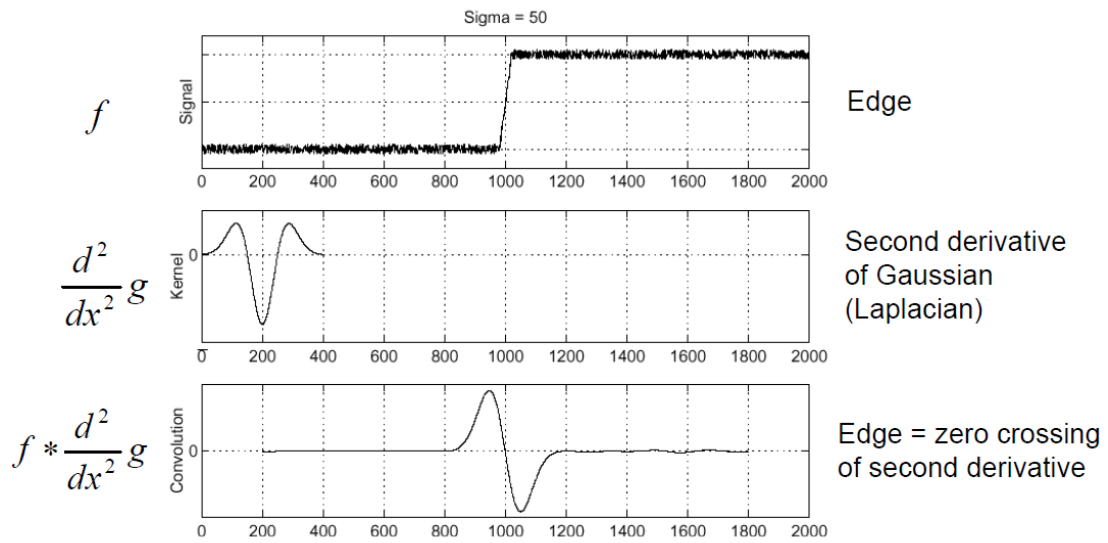
Θεωρητική βάση της μεθόδου εντοπισμού blob (μεγάλου δυαδικού αντικειμένου)

Μία από τις πρώτες και βασικές μεθόδους εντοπισμού blob βασίζεται στο μετασχηματισμό Laplace της κατανομής Gauss. Η λογική ανάπτυξης της τεχνικής ξεκινάει από τον εντοπισμό ακμών. Έστω  $f$  η συνάρτηση που εκφράζει το χαρακτηριστικό που μας ενδιαφέρει. Εντός της περιοχής ενδιαφέροντος η  $f$  λαμβάνει ένα προκαθορισμένο εύρος τιμών ανάλογα με τα κριτήρια εντοπισμού. Στο σημείο που παρουσιάζει μέγιστο η συνάρτηση που προκύπτει από τη συνέλιξη της  $f$  με την παράγωγο της κατανομής Gauss εντοπίζεται η ακμή [18].



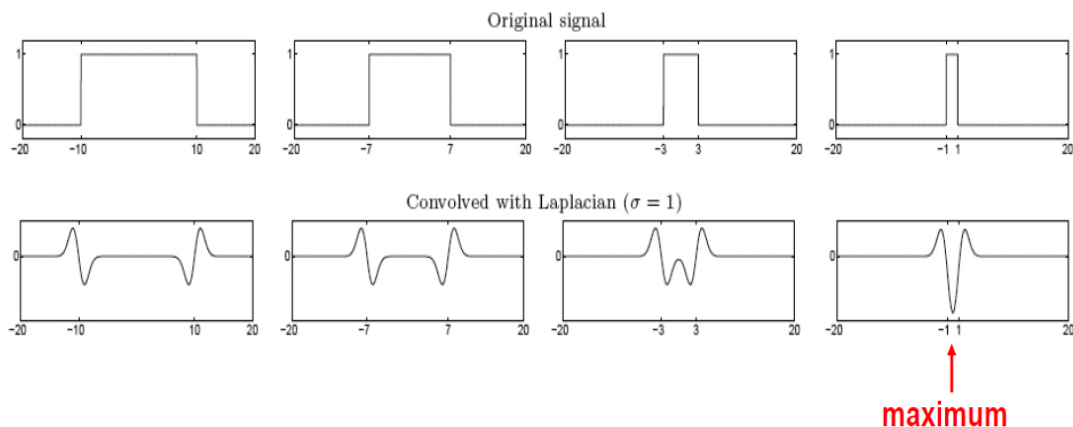
Σχήμα 3.2: Εντοπισμός ακμής (1<sup>ος</sup> τρόπος)

Μία δεύτερη εκδοχή του τρόπου εντοπισμού ακμής είναι να υπολογίσουμε τη συνέλιξη της συνάρτησης  $f$  με τη δεύτερη παράγωγο (Laplace) της συνάρτησης κατανομής Gauss. Η ακμή έτσι εντοπίζεται στο σημείο διέλευσης από το μηδέν της συνάρτησης που θα προκύψει.



Σχήμα 3.3: Εντοπισμός ακμής(2<sup>η</sup> εκδοχή)

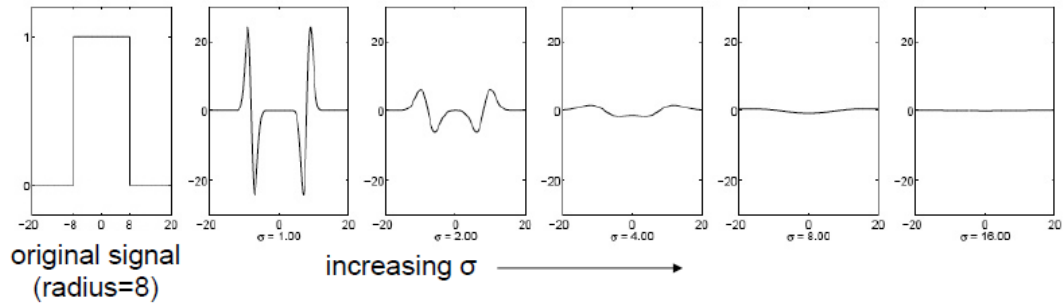
Ουσιαστικά η ακμή αντιστοιχεί σε μια κυμάτωση της συνάρτησης  $f$ , και το blob προκύπτει από την υπέρθεση δύο κυματώσεων. Στη δεύτερη σειρά γραφημάτων του σχήματος 3.4 έχει σχεδιαστεί το αποτέλεσμα της συνέλιξης της Λαπλασιανής της κατανομής Gauss με τη συνάρτηση του χαρακτηριστικού κατά μήκος μιας διάστασης ενός blob. Η εν λόγω απόκριση λαμβάνει μέγιστη τιμή στο κέντρο του blob με την προϋπόθεση η κλίμακα της Λαπλασιανής να είναι «συγχρονισμένη» με την κλίμακα του blob.



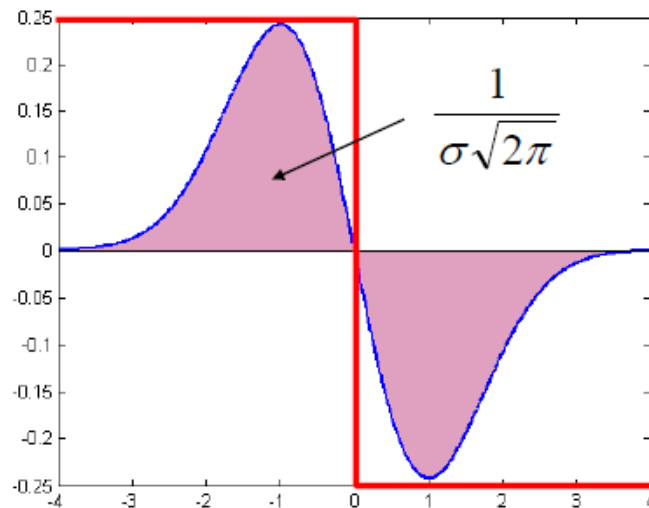
Σχήμα 3.4: Από τις ακμές στα blobs



Σκοπός είναι να βρούμε τη χαρακτηριστική κλίμακα του blob υπολογίζοντας τη συνέλιξη της συνάρτησης χαρακτηριστικού με τον Laplace της κατανομής Gauss σε διάφορες κλίμακες και παράλληλα αναζητώντας το μέγιστο της απόκρισης.

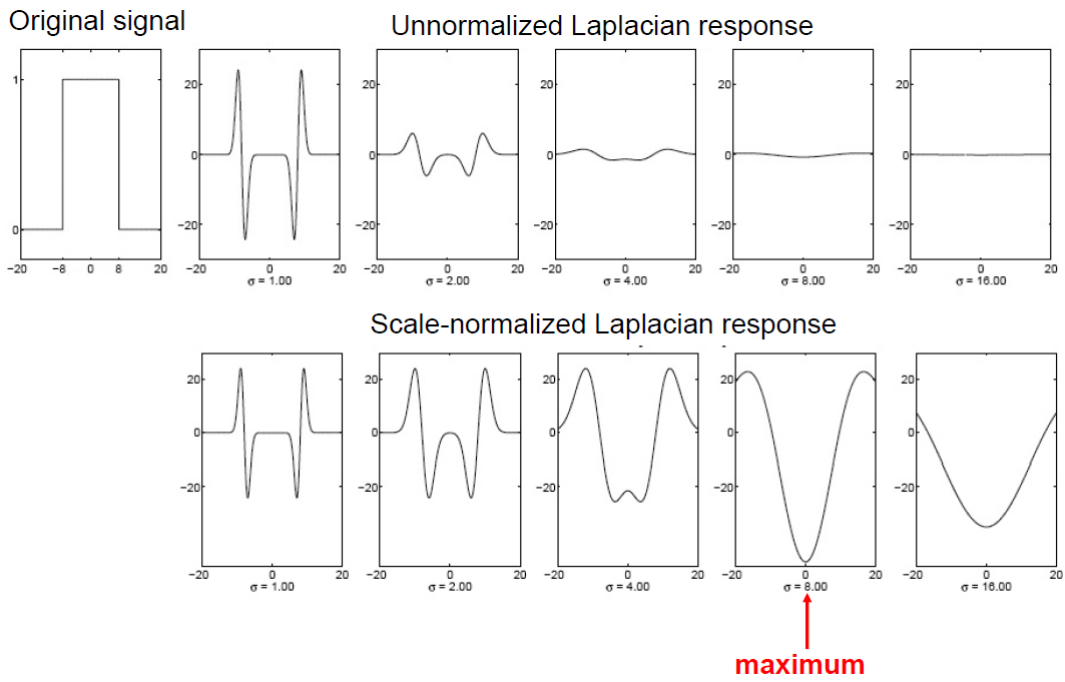


Σχήμα 3.5: Μη κανονικοποιημένη Λαπλασιανή απόκριση



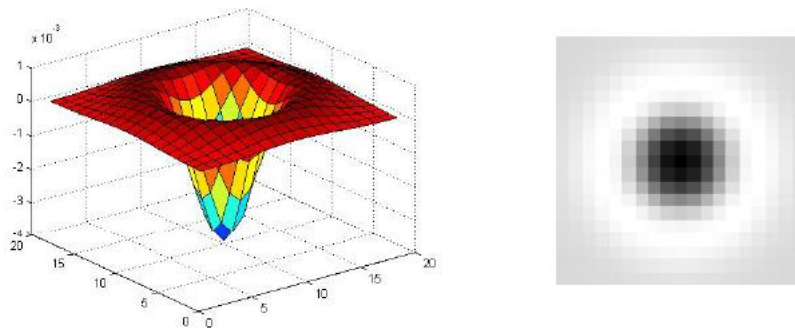
Σχήμα 3.6: Απόκριση του φίλτρου που υλοποιεί τη Λαπλασιανή της κατανομής Gauss

Όπως παρατηρείται από τα παραπάνω σχήματα η απόκριση του εν λόγω φίλτρου μειώνεται καθώς αυξάνεται η κλίμακα. Για να προσδώσουμε αμεταβλητότητα ως προς την κλίμακα θα πρέπει να πολλαπλασιάσουμε με τον συντελεστή  $\sigma^2$ . Παρακάτω φαίνεται το αποτέλεσμα της Λαπλασιανής απόκρισης.



Σχήμα 3.7: Κανονικοποιημένη ως προς την κλίμακα απόκριση

Αν μεταφέρουμε το πρόβλημα στις δύο διαστάσεις προκύπτει ο τελεστής εντοπισμού blob:



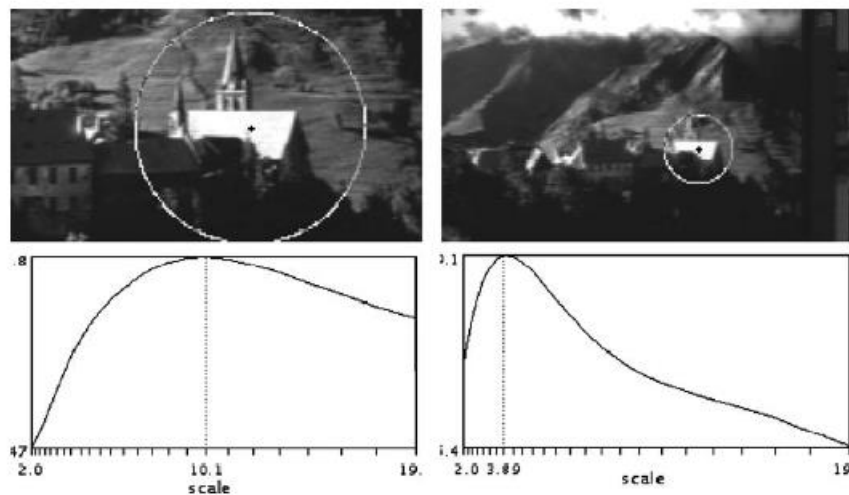
$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

Σχήμα 3.8: Τελεστής εντοπισμού blob στις δύο διαστάσεις

Και μετά την κανονικοποίηση ως προς την κλίμακα:

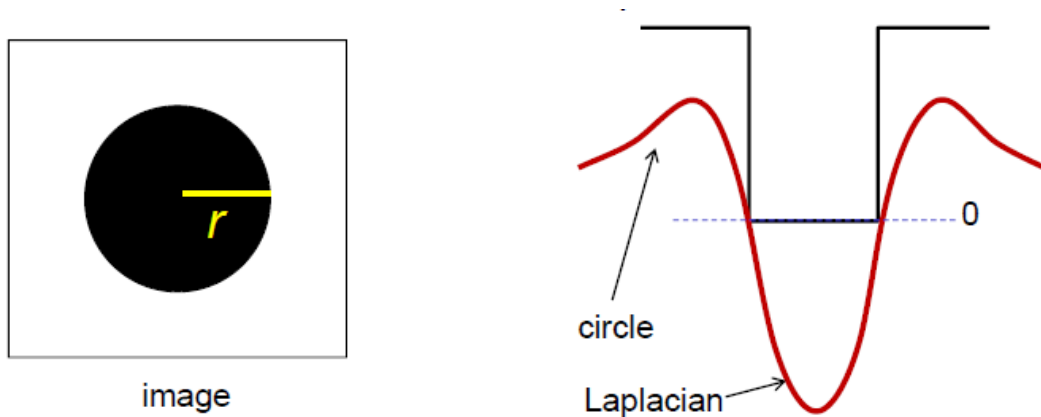
$$\nabla_{\text{norm}}^2 g = \sigma^2 \left( \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)$$

Ένας βασικός στόχος της τεχνικής εντοπισμού blob είναι να εξασφαλίσει τη συµµεταβλητότητα µε την κλίµακα, δηλαδή να µπορεί να εντοπίσει περιοχές µε κοινά χαρακτηριστικά ανεξαρτήτως της κλίµακας της εικόνας. Χρειάζεται λοιπόν ένας µηχανισµός επιλογής της κλίµακας που θα υπολογίζει το µέγεθος της περιοχής ενδιαφέροντος που είναι συναλλοίωτο µε το µετασχηµατισµό της εικόνας.



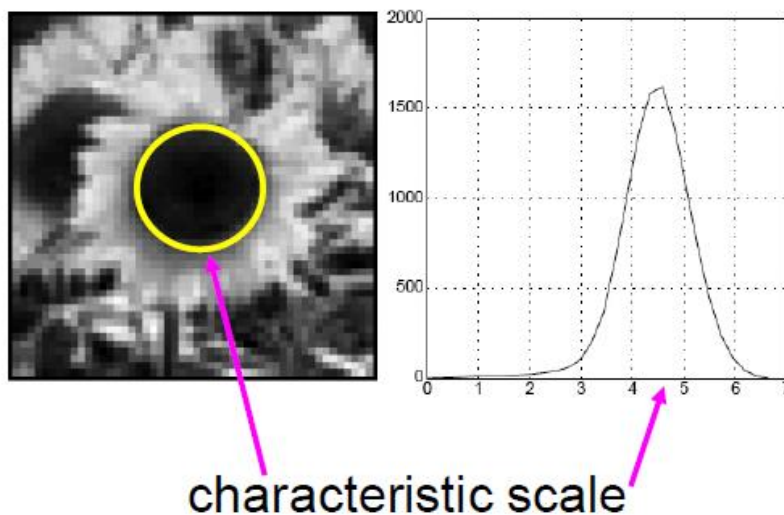
Σχήµα 3.9: Συµµεταβλητότητα τεχνικής εντοπισµού blob ως προς την κλίµακα

Θα υπολογίσουµε την κλίµακα στην οποία µεγιστοποιείται η Λαπλασιανή απόκριση όταν το φίλτρο εφαρµόζεται σε ένα δυαδικό κύκλο ακτίνας  $R$ . Αυτό συµβαίνει όταν η συνάρτηση της Λαπλασιανής κατανοµής Gauss µηδενίζεται στα σηµεία του κύκλου και η µέγιστη απόκριση λαµβάνεται για:  $\sigma = R/\sqrt{2}$



Σχήμα 3.10: Μεγιστοποίηση Λαπλασιανής απόκρισης

Ορίζουμε σαν χαρακτηριστική κλίμακα του blob την κλίμακα στην για την οποία μεγιστοποιείται η Λαπλασιανή απόκριση στο κέντρο του blob.

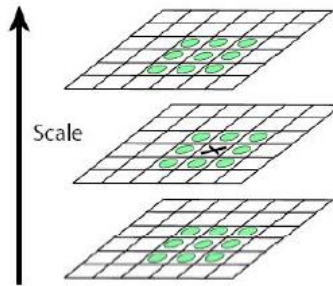


Σχήμα 3.11: Χαρακτηριστική κλίμακα του blob

Ο αλγόριθμος εντοπισμού blob που βασίζεται σε υπολογισμούς στο χώρο της κλίμακας (scale-space blob detector) περιλαμβάνει δύο βασικά στάδια:

- Υπολογισμός της συνέλιξης της συνάρτησης του χαρακτηριστικού της εικόνας με τον κανονικοποιημένο ως προς την κλίμακα τελεστή blob (ο Laplace της κατανομής Gauss) σε διάφορα επίπεδα κλίμακας
- Εύρεση του μεγίστου της Λαπλασιανής απόκρισης σε διάφορα επίπεδα κλίμακας:

$$f(x, y) \times \nabla_{norm}^2 g = f(x, y) \times \sigma^2 \left( \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right)$$



Σχήμα 3.12: Αλγόριθμος εντοπισμού blob στο χώρο της κλίμακας

### Εφαρμογές της μεθόδου εντοπισμού blob

Υπάρχουν πολλοί λόγοι για να μελετήσει ή να αναπτύξει κάποιος τεχνικές εντοπισμού blob. Μία βασική λειτουργικότητα της μεθόδου είναι η παροχή συμπληρωματικής πληροφορίας για περιοχές μιας εικόνας, η οποία δεν είναι διαθέσιμη από μεθόδους εντοπισμού ακμών ή γωνιών. Οι πρώτες εφαρμογές αναπτύχθηκαν με σκοπό να προσδιορίσουν περιοχές ενδιαφέροντος σε μια εικόνα για περαιτέρω επεξεργασία. Αυτές οι περιοχές μπορούν να σηματοδοτήσουν την ύπαρξη αντικειμένων και ως εκ τούτου μπορούν να χρησιμοποιηθούν σε εφαρμογές αναγνώρισης ή παρακολούθησης αυτών. Σε άλλους κλάδους εφαρμογών όπως η ανάλυση ιστογράμματος, ο εντοπισμός blob μπορεί να χρησιμοποιηθεί στην ανάπτυξη συστημάτων που διαχωρίζουν περιοχές σε μία εικόνα. Μια άλλη κοινή εφαρμογή αυτής της τεχνικής συναντάται σε συστήματα ανάλυσης και αναγνώρισης υφής. Σε πιο σύγχρονες εφαρμογές ο εντοπισμός blob χρησιμοποιείται για τον εντοπισμό σημείων ενδιαφέροντος σε τεχνικές επεξεργασίας εικόνας που υλοποιούν την υπέρθεση πολλαπλών σκηνών σε μία ολοκληρωμένη εικόνα.

### Υλοποίηση της τεχνικής εντοπισμού blob μέσω της βιβλιοθήκης JMyron

Στην παρούσα εργασία υλοποιήθηκε ο εντοπισμός χρωματιστού αντικειμένου μέσω της βιβλιοθήκης ανοιχτού κώδικα JMyron [19]. Πρόκειται για έναν wrapper (σαν “wrapper” κλάση θεωρούμε μια οποιαδήποτε κλάση που «περιτυλίσσει» ή «ενθυλακώνει» την λειτουργικότητα μιας άλλης κλάσης ή δομής) για την Java ή την Processing, ο οποίος περιλαμβάνει όλα τα αντικείμενα και μεθόδους (ρουτίνες) της τεχνικής εντοπισμού blob και τα οποία έχουν αναπτυχθεί σε γλώσσα C++. Η βιβλιοθήκη JMyron αποτελεί ουσιαστικά ένα πρόσθετο (plugin) συμβατό με όλες τις πλατφόρμες και τις γλώσσες προγραμματισμού που χρησιμοποιείται σε εφαρμογές καταγραφής video και υπολογιστικής όρασης.

# 4

## *Αρχιτεκτονική και λεπτομέρειες υλοποίησης*

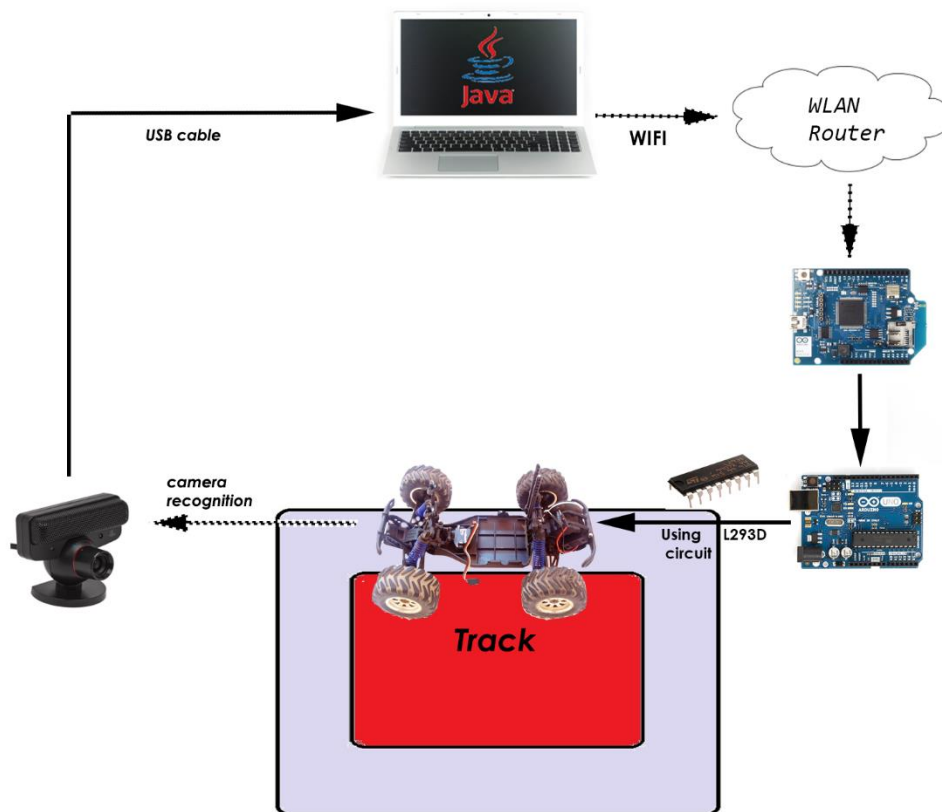
Σε αυτό το κεφάλαιο θα αναλυθεί και θα περιγραφεί η αρχιτεκτονική του συστήματος, και έπειτα το hardware και το software που το απαρτίζουν.

### *4.1 Αρχιτεκτονική του συστήματος*

Το σύστημα αυτό κατασκευάστηκε, για να προωθήσει αλλά και για να επιβεβαιώσει την αναγνώριση ενός κινούμενου οχήματος σε πραγματικό χρόνο με εξαρτήματα και υλικά του εμπορίου, τα οποία συνεργάζονται κάτω από την αιγίδα ενός προγράμματος που εκμεταλλεύεται την τεράστια υπολογιστική δύναμη ενός υπολογιστή. Χωρίζουμε την αρχιτεκτονική του συστήματος σε δύο σκέλη:

- Hardware, στο οποίο συμπεριλαμβάνεται όποια συσκευή ή τεχνολογικό υλικό χρησιμοποιήθηκε και επεξεργάστηκε, καθώς και η συνδεσμολογία μεταξύ αυτών.
- Software, στο οποίο συμπεριλαμβάνεται όποιο λογισμικό δημιουργήθηκε, καθώς και οποιοδήποτε έτοιμο λογισμικό χρησιμοποιήθηκε και πως.

Τα παραπάνω, αλληλοεπιδρούν μεταξύ τους σε έναν κύκλο ροής δεδομένων, σε ένα δηλαδή, feedback system. Αυτή η γενική εικόνα του συστήματος φαίνεται στο σχήμα 4.1 και περιγράφεται παρακάτω.



Σχήμα 4.1: Εξαρτήματα του συστήματος και γενική εικόνα της ροής δεδομένων.

Σημείωση: το Arduino UNO και οι υπόλοιπες πλακέτες και κυκλώματα βρίσκονται προσκολλημένα πάνω στο όχημα, κάτι το οποίο δεν φαίνεται στο παραπάνω σχεδιάγραμμα. Δίνεται βάση κυρίως στην επικοινωνία όλων των επιμέρους κομματιών.

Μέσω της κάμερας, το πρόγραμμα στον υπολογιστή βρίσκει την θέση του οχήματος και μέσω της επιλογής του χρήστη το πρόγραμμα στέλνει εντολή κίνησης στο αμαξάκι μέσω του δικτύου, του WiFi shield και του Arduino UNO. Έπειτα το αμαξάκι κινείται, επανατροφοδοτώντας την κάμερα με νέα πληροφορία θέσης, ξεκινώντας ξανά τον κύκλο του προγράμματος.

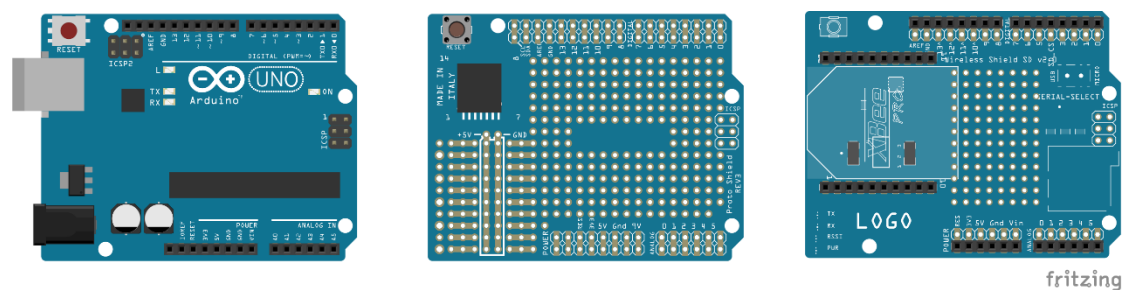
## 4.2 Λεπτομέρειες συνδεσμολογίας Hardware

Παρακάτω αναλύεται το ηλεκτρολογικό σύστημα που αναπτύχθηκε μεταξύ των εξαρτημάτων του οχήματος.

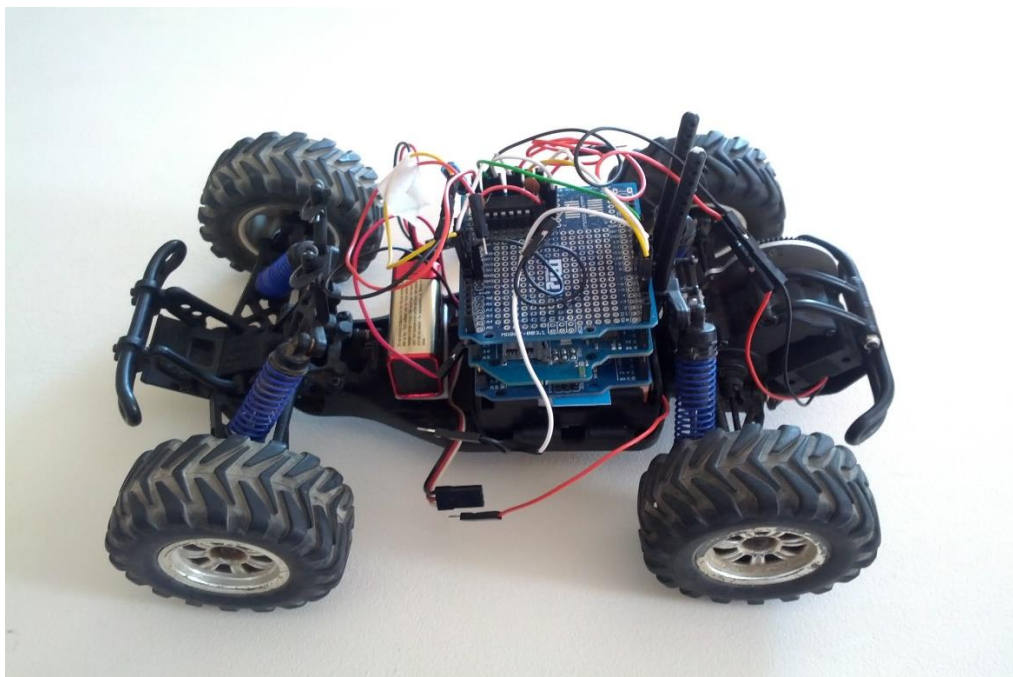
Πέρα από τον σκελετό του οχήματος με τα μοτέρ που έχει πάνω του, τοποθετήθηκαν 2 σειρές τεσσάρων AAA μπαταριών για την τροφοδοσία τους, και μία μπαταρία 9V για την τροφοδοσία του Arduino UNO.

Πάνω σε αυτά τοποθετήθηκε μία συστοιχία του Arduino UNO, του WiFi shield, και του Proto shield (βλ. εικόνα 4.2). Το Proto Shield τοποθετήθηκε προφανώς πάνω πάνω, ώστε να τοποθετηθεί και να συνδεσμολογηθεί το κύκλωμα ελέγχου του DC motor.

Η τελική μορφή του οχήματος μπορεί να φανεί στην εικόνα 4.3.



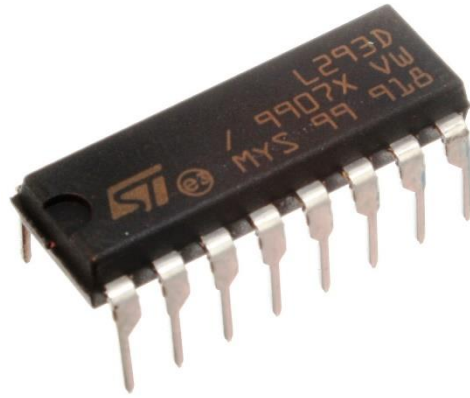
Σχήμα 4.2: Το Arduino Uno, και τα 2 shield που τοποθετήθηκαν πάνω σε αυτό.



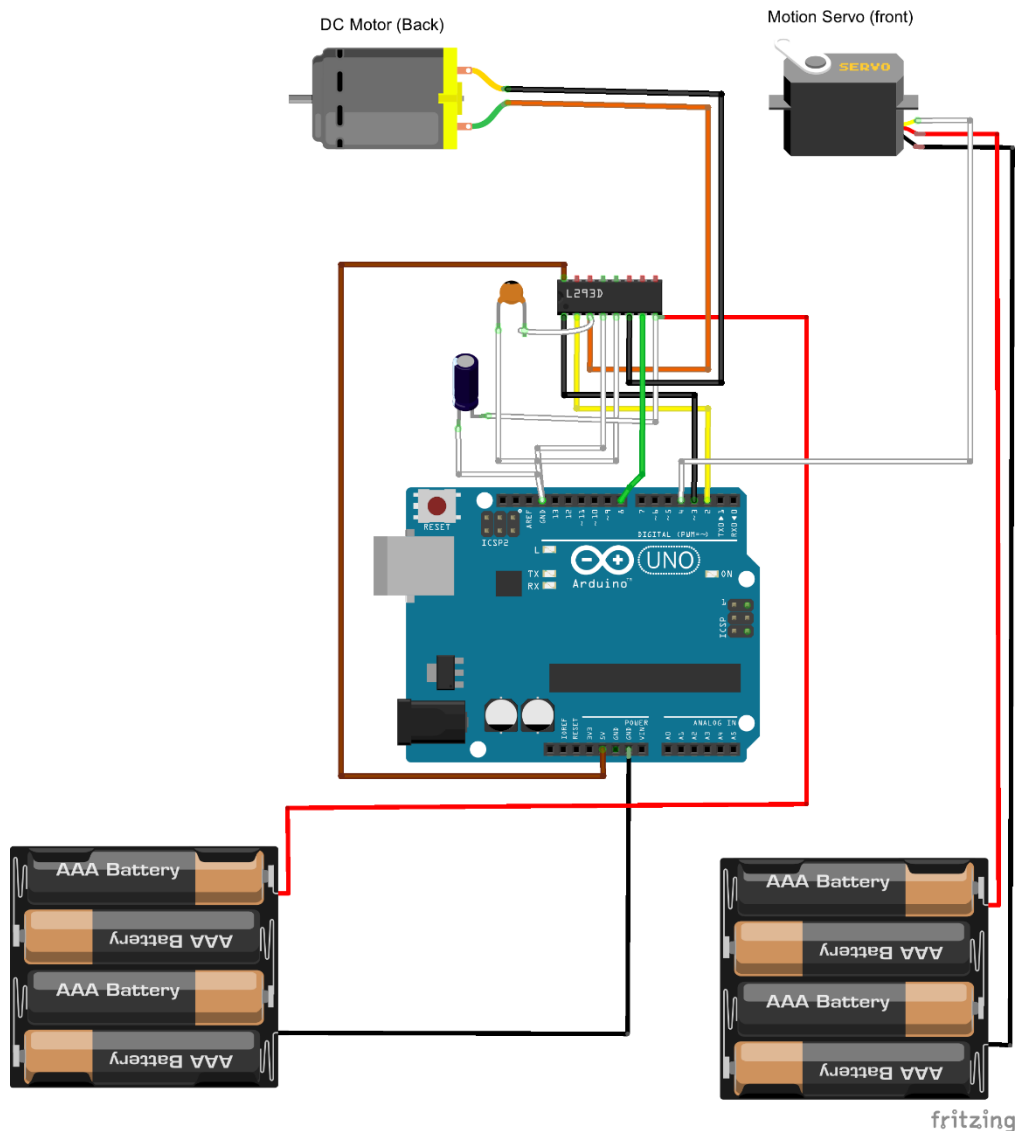
Εικόνα 4.3: Το όχημα μας στην τελική του μορφή.



Το κύκλωμα ελέγχου, βασίζεται πάνω στο ολοκληρωμένο κύκλωμα L293D (βλ. εικόνα 4.3- [https://www.arduino.cc/documents/datasheets/H-bridge\\_motor\\_driver.PDF](https://www.arduino.cc/documents/datasheets/H-bridge_motor_driver.PDF)), το οποίο έχει ως σκοπό τον έλεγχο της μπροστά αλλά και πίσω κίνησης του DC motor, στέλνοντας απλές εντολές από το Arduino UNO [20] [21] [22]. Επίσης χρησιμοποιήθηκαν 2 capacitors και διάφορα jumper καλώδια για την σύνδεση του συστήματος. Η σχηματική απεικόνιση (schematics) του κυκλώματος ελέγχου του μοτέρ, αλλά και του κυκλώματος και των υπόλοιπων στοιχείων, απεικονίζεται στο Σχήμα 4.5.



Εικόνα 4.4: Το ολοκληρωμένο κύκλωμα L293D



Σχήμα 4.5: Schematics του οχήματος και των στοιχείων του.

## 4.3 Λεπτομέρειες και ανάλυση του κώδικα του Arduino Uno

Ο κώδικας που φορτώθηκε στο Arduino Uno σχετίζεται άμεσα με το Hardware, καθώς το Arduino έχει τον πιο σημαντικό ρόλο στην λειτουργία του κυκλώματος που αναφέρθηκε προηγουμένως στην ενότητα 4.2.

### 4.3.1 Λογική υλοποίησης και βιβλιοθήκες

Η επικοινωνία του Arduino με το λογισμικό που εκτελείται στον υπολογιστή μας και ο απομακρυσμένος χειρισμός του, βασίστηκε στη λογική του μοντέλου επικοινωνίας client – server. Ουσιαστικά αναπτύχθηκε ένας web server στην πλατφόρμα του Arduino, ο οποίος δέχεται tcp μηνύματα (tcp requests) από τον client μέσω wlan διεπαφής, το ρόλο του οποίου ανέλαβε το λογισμικό στον υπολογιστή μας. Το κάθε tcp μήνυμα που λαμβάνεται από τον server αποτελεί μια συμβολοσειρά η οποία επεξεργάζεται περαιτέρω για να ενεργοποιηθούν οι κατάλληλες έξοδοι του Arduino και να εκτελεστεί η επιθυμητή κίνηση του οχήματος. Αξίζει να αναφέρουμε ότι κατά τις πρώτες δοκιμές η υλοποίηση βασίστηκε στο πρωτόκολλο http. Λόγω όμως της περιττής πληροφορίας που περιλαμβάνεται σε ένα http μήνυμα (πληροφορίες επικεφαλίδας) η απόκριση του οχήματος στις εντολές που λογισμικού καθυστερούσε και το πρωτόκολλο http κρίθηκε μη επαρκής για τις ανάγκες της εφαρμογής.

Στον κώδικα αυτό χρησιμοποιούνται 3 έτοιμες βιβλιοθήκες οι οποίες παρέχονται από την εταιρία κατασκευής του:

- SPI.h(Serial Peripheral Interface): Αυτή η βιβλιοθήκη χρησιμοποιείται για την ταχεία επικοινωνία μεταξύ διάφορων περιφερειακών συσκευών, σε μικρές αποστάσεις. Μπορεί επίσης να χρησιμοποιηθεί για την σειριακή απικοινωνία μεταξύ δύο μικροελεγκτών.
- WiFi.h: Η βιβλιοθήκη αυτή με την χρήση του WiFi shield, επιτρέπει στο Arduino να συνδεθεί στο τοπικό δίκτυο και κατ' επέκταση στο ίντερνετ. Στην περίπτωση μας χρησιμοποιείται σαν server, ο οποίος μπορεί να δεχτεί εισερχόμενες συνδέσεις. Η βιβλιοθήκη υποστηρίζει κρυπτογ'ραφηση WEP και WPA2 Personal.
- Servo.h: Αυτή η βιβλιοθήκη επιτρέπει στην πλακέτα Arduino να ελέγξει RC servo motors. Χρησιμοποιώντας την, μπορούμε να ελέγξουμε το servo motor του οχήματος που αφορά την πλάγια κίνηση, με μεγάλη ακρίβεια. Το μοτέρ με μία εντολή της βιβλιοθήκης μπορεί να περιστραφεί από 0 μέχρι 180 μοίρες, κάτι παραπάνω από αρκετό για τον χειρισμό του οχήματος.

### 4.3.2 Ανάλυση του κώδικα

Ο κώδικας αυτός, όπως και κάθε άλλος κώδικας Arduino αποτελείται από 4 κομμάτια:

1. Οι δηλώσεις των μεταβλητών, σταθερών και των βιβλιοθηκών. Εδώ δηλώθηκαν οι 3 παραπάνω βιβλιοθήκες, τα pins που χρησιμοποιούνται (βλ. εικόνα 4.5) καθώς και κάποιες σταθερές που περιέχουν κωδικοποιημένα τις εντολές που μπορούν να σταλούν στο Arduino.
2. Η μέθοδος setup(), η οποία τρέχει μία φορά στην εκκίνηση του κώδικα ώστε να αρχικοποιήσει και να εκκινήσει κάποια πράγματα. Στην περίπτωση μας, καθορίζεται η φύση των pins(OUTPUT ή INPUT) και γίνεται προσπάθεια σύνδεσης του Arduino μέσω του WiFi shield στο τοπικό δίκτυο. Μετά από την επιτυχία σύνδεσης, ενεργοποιείται ο server και είναι έτοιμος να λάβει συνδέσεις.
3. Η μέθοδος loop(), η οποία θα εκτελείται σε επανάληψη όσο το Arduino είναι ενεργό. Εδώ, ο server μόλις αποκτήσει μία σύνδεση, διαβάζει τα πρώτα 2 γράμματα του string που του στάλθηκε, και αποκωδικοποιώντας το, ενεργοποιεί την αντίστοιχη μέθοδο κίνησης (που βρίσκεται στο επόμενο κομμάτι).
4. Οι υπόλοιπες βοηθητικές μέθοδοι, οι οποίες παρατίθενται μετά την μέθοδο loop(), και χρησιμοποιούνται από αυτή κατά την εκτέλεση του κώδικα. Εδώ περιλαμβάνονται κάποιες μέθοδοι που έχουν ως σκοπό την εμφάνιση πληροφοριών της σύνδεσης WiFi (σε περίπτωση που συνδεθεί στο Arduino κάποια οθόνη ή με τον υπολογιστή). Επιπλέον, περιλαμβάνονται οι πιο σημαντικές μέθοδοι, κάθε μία από τις οποίες περιέχει τις εντολές για κάθε μία πιθανή κίνηση του οχήματος. Για παράδειγμα η leftSeven αλλάζει την επιθυμητή γωνία περιστροφής, την οποία μεταφέρει σαν εντολή στον σερβοκινητήρα. Η centerCar από την άλλη, στέλνει εντολή στον πίσω κινητήρα να σταματήσει, καθώς και αρχικοποιεί την θέση του σερβοκινητήρα.

## **4.4 Λεπτομέρειες και ανάλυση του Software του υπολογιστή**

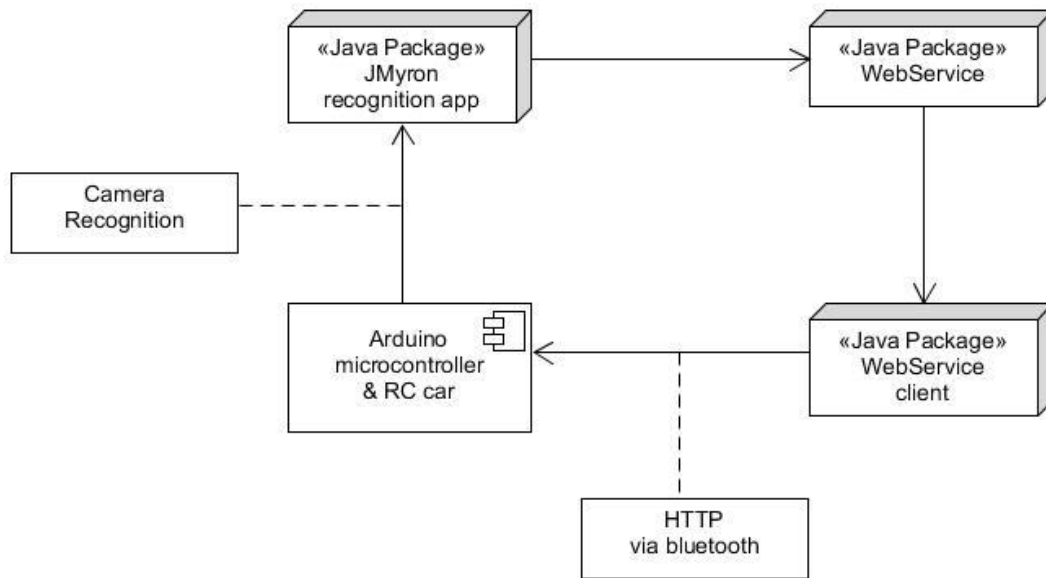
### **4.4.1 Αρχιτεκτονική της εφαρμογής**

Η εφαρμογή αυτή αποτελείται από τρία διαφορετικά πακέτα κώδικα και ένα ακόμα υποπακέτο, τα οποία υλοποιούν τις τέσσερις λειτουργίες που περιεγραφήκαν στην παράγραφο 2.3.2.

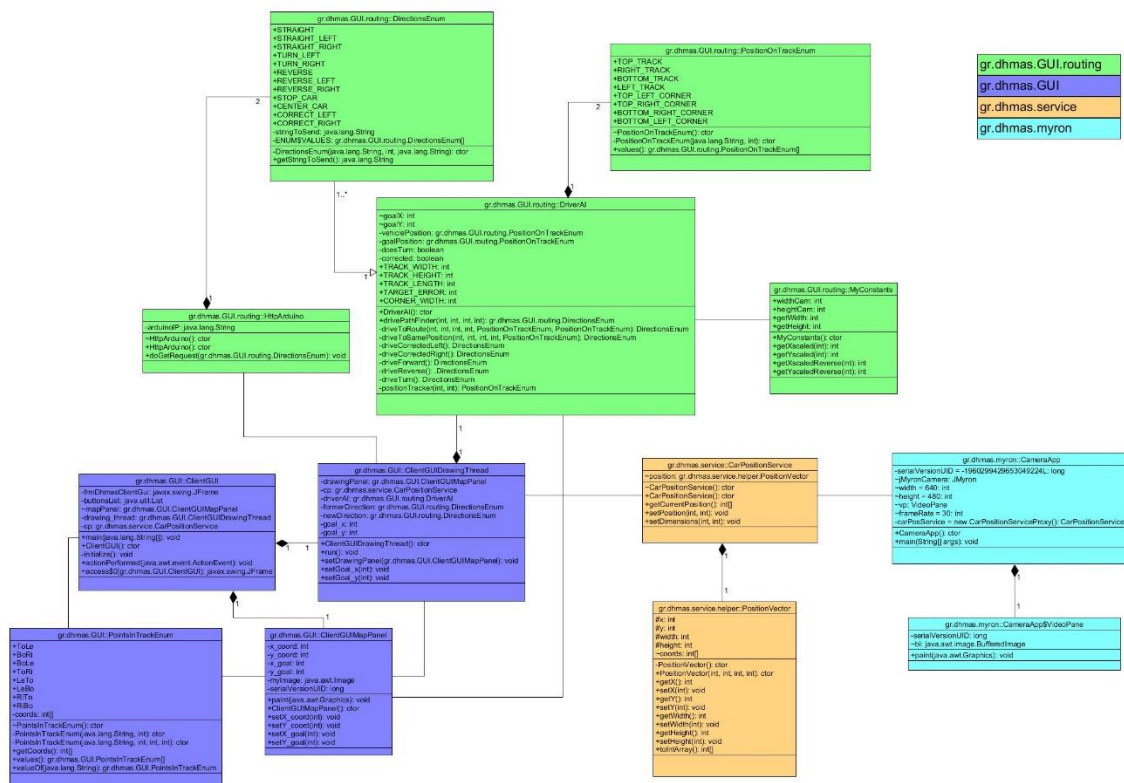
Το πρώτο πακέτο αφορά την ανίχνευση της τρέχουσας θέσης οχήματος, το δεύτερο υλοποιεί την αποθήκευση και ανάκτηση της τρέχουσας θέσης οχήματος, το τρίτο (υποπακέτο του τέταρτου) αφορά την υποστήριξη πλοήγησης του οχήματος, και το τέταρτο υλοποιεί το διαδραστικό περιβάλλον της εφαρμογής.

Αυτά τα τέσσερα πακέτα έχουν δύο εκτελέσιμες μεθόδους “main”, όπου χρειάζεται η εκτέλεση και των 2 για να λειτουργήσει σωστά η εφαρμογή. Τα πακέτα αυτά βρίσκονται στο ίδιο project καθώς και αλληλοεπιδρούν μεταξύ τους.

Η σύνδεσή τους και η ροή δεδομένων μπορούν να φανούν συνοπτικά στο παρακάτω διάγραμμα (εικόνα 4.6) καθώς και πιο αναλυτικά οι κλάσεις των κάθε πακέτων, στο διάγραμμα της εικόνας (4.7), όπου και αναλύονται στην συνέχεια.



Εικόνα 4.6 : Διάγραμμα εξαρτημάτων



Εικόνα 4.7 : Διάγραμμα κλάσεων όλου του προγράμματος. Τα διαφορετικά πακέτα περιγράφονται με διαφορετικό χρώμα.

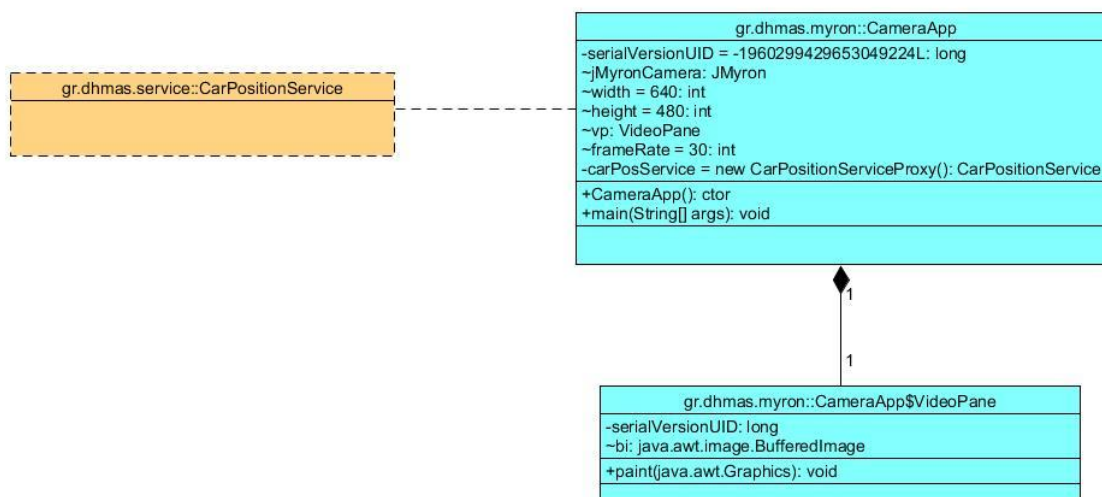
## 4.4.2 Ανάλυση των πακέτων και των κλάσεων τους

### 4.4.2.1 Πακέτο *gr.dhmas.myron* – Υποσύστημα ανίχνευσης τρέχουσας θέσης

#### οχήματος

Ο σκοπός του πακέτου αυτού είναι η ανίχνευση της θέσης του αμαξίου μέσω της κάμερας, και η αποστολή της για αποθήκευση στο web Service.

Το πακέτο αυτό περιέχει την βιβλιοθήκη JMyron, η οποία είναι μία βιβλιοθήκη Java ανοιχτού κώδικα, βασισμένη στην γλώσσα Processing, και χρησιμοποιείται για την αναγνώριση βίντεο. Στην συγκεκριμένη εφαρμογή, χρησιμοποιείται από την συμπεριλαμβανόμενη κλάση CameraApp για την ανίχνευση της τοποθεσίας του αμαξίου μέσω του χρώματός του και επιλέγοντας το κεντρικό του σημείο. Στο παρακάτω διάγραμμα κλάσεων βλέπουμε αναλυτικά αυτή την κλάση καθώς και την σύνδεση της με μια κλάση του πακέτου του Service.



Εικόνα 4.8: Διάγραμμα κλάσεων του πακέτου *gr.dhmas.myron*

Η κλάση *CameraApp* περιέχει την συνάρτηση *main*, οποία δημιουργεί ένα αντικείμενο της κλάσης, κατά την εκτέλεση της. Το αντικείμενο αυτό, δημιουργεί ένα παράθυρο καθώς είναι τύπου *JFrame*, παραμετροποιεί και ενεργοποιεί ένα αντικείμενο της βιβλιοθήκης *JMyron*, καθώς και δημιουργεί ένα αντικείμενο της περιλαμβανόμενης κλάσης *VideoPane* το οποίο εμφανίζεται μέσα στο παράθυρο. Έπειτα δημιουργείται ένας *timer* για την συστηματική ανανέωση (συνάρτηση *paint*) του *VideoPane*.

Το αντικείμενο *VideoPane* έχει ως σκοπό την αναπαραγωγή του βίντεο που συνεχώς τραβάει η κάμερα μέσα στο προηγούμενος αναφερόμενο παράθυρο, καθώς και την διαρκή ανίχνευση ενός αντικειμένου συγκεκριμένου χρώματος (στην περίπτωση της εργασίας είναι το

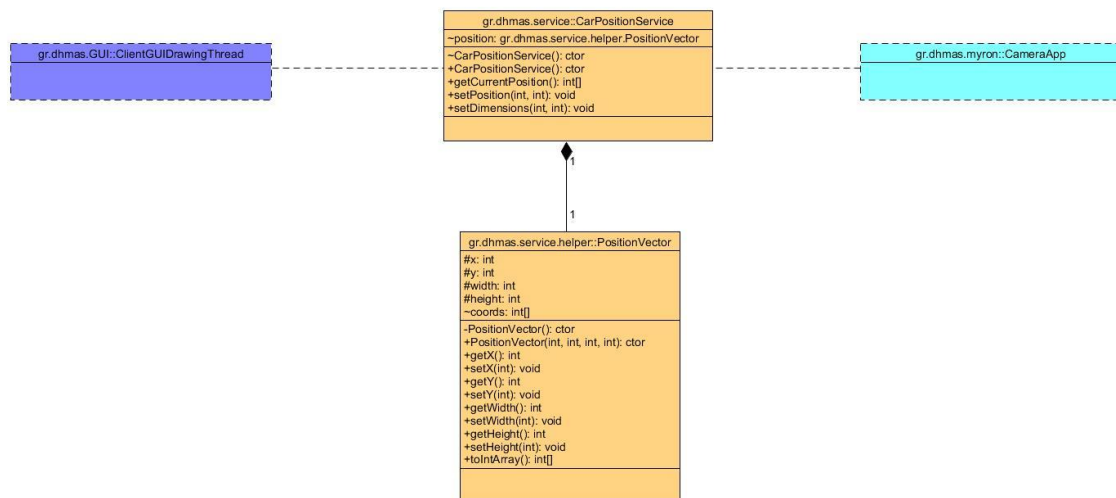
αμαξάκι). Πέρα από την εμφάνιση των αποτελεσμάτων, το αντικείμενο στέλνει την θέση του αυτοκινήτου στο web service μέσω ενός proxy (βλ. εικόνα 4.8).

#### 4.4.2.2 Πακέτο *gr.dhmas.service* – Υποσύστημα αποθήκευσης τρέχουσας θέσης

##### οχήματος

Το πακέτο αυτό αποτελεί ένα Web Service με σκοπό την διαρκή αποθήκευση της ζωντανής θέσης του οχήματος και τις διαστάσεις του χάρτη, καθώς και την δυνατότητα ανάκτησης αυτής της πληροφορίας.

Σε αυτό το πακέτο περιλαμβάνονται οι δύο κλάσεις χρήσης (βλ. εικόνα 4.9) αλλά και οι κλάσεις για την λειτουργία του web service, οι οποίες έχουν παραχθεί αυτόματα από το Eclipse, κατά την δημιουργία του( οπότε και δεν παρουσιάζονται).



Εικόνα 4.9: Διάγραμμα κλάσεων του πακέτου *gr.dhmas.service*

Η βοηθητική κλάση *PositionVector* έχει τέσσερις μεταβλητές για την οριζόντια και κάθετη θέση, καθώς και για το ύψος και το πλάτος. Επίσης περιλαμβάνει συναρτήσεις για την αποθήκευση και λήψη των τιμών των παραπάνω μεταβλητών.

Η κλάση *CarPositionService* δημιουργεί ένα αντικείμενο της παραπάνω κλάσης, και δίνει την δυνατότητα σε άλλες κλάσεις (*ClientGUIDrawingThread*, *CameraApp*) μέσω συναρτήσεων, να αποθηκεύσουν μια θέση (την ανιχνευμένη θέση του αμαξιού), να πάρουν αυτή την θέση, η να θέσουν τις διαστάσεις του χάρτη.

#### 4.4.2.3 Πακέτο *gr.dhmas.GUI.routing* – Υποσύστημα πλοήγησης οχήματος

Το πακέτο αυτό είναι υποπακέτο του *webServiceClient*. Οι κλάσεις που περιέχονται σε αυτό το πακέτο είναι υπεύθυνες για τον έλεγχο και για την δρομολόγηση του αμαξιού σύμφωνα με τα δεδομένα που λήφθηκαν, δηλαδή το AI (Artificial Intelligence) της εργασίας. Η βασική κλάση ονομάζεται *DriverAI* και υποστηρίζεται άλλες τρεις κλάσεις που συμβάλουν στην λειτουργία της. Επιπρόσθετα στο πακέτο αυτό εμπεριέχεται η κλάση *HttpArduino* η οποία είναι υπεύθυνη για την επικοινωνία με το αμαξάκι/Arduino. (βλ. εικόνα 4.10).



Εικόνα 4.10: Διάγραμμα κλάσεων του πακέτου *gr.dhmas.GUI.routing*

Οι κλάσεις *DirectionsEnum*, *PositionOnTrackEnum* έχουν δημιουργηθεί για να παρέχουν αντικείμενα σε άλλες κλάσεις (όπως φαίνεται στην εικόνα 4.10), τα οποία περιέχουν κάποιες κωδικοποιημένες σταθερές (και μεθόδους), για την ευκολότερη και πιο αποτελεσματική υλοποίηση του προγράμματος. Πιο συγκεκριμένα η κλάση *DirectionsEnum* περιέχει τις



οδηγίες (λειτουργικές αλλά και δοκιμαστικές) που μπορεί να πάρει το αμάξι και η `PositionOnTrack` περιέχει τις 8 θέσεις (λειτουργικές αλλά και δοκιμαστικές) που μπορεί το αμάξι πάνω στον χάρτη.

Για παρόμοιο λόγο κατασκευάστηκε η κλάση `MyConstants`, η οποία περιέχει κάποιες σταθερές τιμές τις συγκεκριμένης υλοποίησης (ύψος - πλάτος κάμερας) καθώς και μεθόδους που λαμβάνουν μία θέση, και την επιστρέφουν υπό κλίμακα σύμφωνα με αυτές τις σταθερές.

Η κλάση `DriverAI` είναι η βασικότερη κλάση αυτού του πακέτου η οποία έχει ως σκοπό την εύρεση της κατάλληλης εντολής που πρέπει να λάβει το αμαξάκι για να κινηθεί από την θέση του (`vehiclePosition`) στην στοχευμένη θέση (`goalPosition`). Αυτό επιτυγχάνεται με τις μεταβλητές και μεθόδους που περιέχει η κλάση, αρχίζοντας με την εκτέλεση της μεθόδου `drivePathFinder`, δίνοντας της τα 2 διανύσματα θέσης. Το τελικό αποτέλεσμα που θα επιστραφεί είναι μία εντολή της κλάσης `DirectionsEnum`.

#### Αναλυτικά οι μέθοδοι:

`drivePathFinder`: Η μέθοδος αυτή δέχεται το διάνυσμα θέσης (x, y) του αυτοκινήτου και του στόχου, και ανάλογα με αυτές επιλέγει ανάμεσα από τις μεθόδους `driveToRoute` και `driveToSamePosition`. Αυτή είναι η εναρκτήρια μέθοδος που πρέπει να επικαλεστεί για να εκτελεστούν και όλες οι υπόλοιπες ώστε να επιστραφεί το τελικό αποτέλεσμα.

`positionTracker`: Δέχεται ένα διάνυσμα θέσης (x, y) και επιστρέφει την κατά προσέγγιση θέση του πάνω στον χάρτη ανάμεσα από έξι(6) επιλογές. Αυτό επιτυγχάνεται με τη σύγκριση του διανύσματος με κάποιες σταθερές των αποστάσεων του χάρτη.

`driveToSamePosition`: Αυτή η μέθοδος ελέγχοντας τις ακριβείς συντεταγμένες θέσης του αυτοκινήτου και του στόχου, και ανάλογα με την θέση τους στον χάρτη, υπολογίζει αν, και πως, πρέπει να κινηθεί το αμαξάκι για να διορθωθεί η θέση του.

`driveToRoute`: Ανάλογα αν το αμαξάκι βρίσκεται σε στροφή, ή σε ευθεία, καλεί την αντίστοιχη μέθοδο οδήγησης (`driveForward`, `driveTurn`).

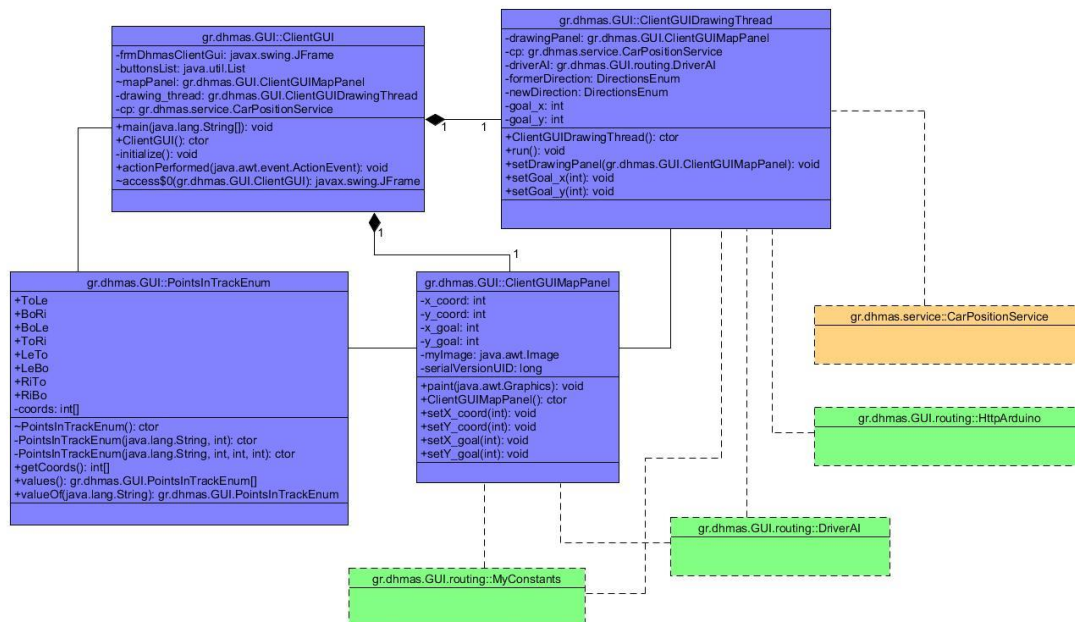
`driveForward`, `driveTurn`: αυτές οι 2 μέθοδοι, επιστρέφουν την τελική εντολή που θα δοθεί στο αμάξι, λαμβάνοντας υπόψη κάποια flags ελέγχου.

`driveReverse`, `driveCorrectedLeft`, `driveCorrectedRight`: Αυτές είναι μέθοδοι αντίστοιχες με τις αμέσως προηγούμενες, αλλά σε δοκιμαστική μορφή και δεν έχουν ακόμα εφαρμοστεί.

Τέλος, η κλάση `HttpArduino` περιέχει τα στοιχεία που χρειάζονται για την σύνδεση με το Arduino μέσω HTTP (και WiFi) καθώς και την μέθοδο `doGetRequest` η οποία κάνει ακριβώς αυτή την δουλειά. Δέχεται ως όρισμα την εντολή κατεύθυνσης, και στέλνει το κωδικοποιημένο string της εντολής στο Arduino, μέσω ενός socket.

#### 4.4.2.4 Πακέτο `gr.dhmas.GUI` – Διαδραστικό περιβάλλον εφαρμογής

Οι κλάσεις αυτού του πακέτου είναι υπεύθυνες για την δημιουργία του παραθυρικού περιβάλλοντος της εφαρμογής, καθώς και για την σχεδίαση και έλεγχο του χάρτη και τη θέση του αμαξίου πάνω σε αυτό. Επιπρόσθετα, μέσω κουμπιών που προσφέρει η διεπαφή (και με την χρήση της κλάσης `HttpArduino` στο πακέτο `GUI.routing`), δίνεται η δυνατότητα στον χρήστη να στείλει στο αμαξάκι εντολές κατεύθυνσης, όπως περιγράφεται στο διάγραμμα της εικόνας 4.11.



Εικόνα 4.11: Διάγραμμα κλάσεων του πακέτου `gr.dhmas.GUI`

Αρχικά, η κλάση `PointsInTrackEnum` έχει δημιουργηθεί για την ευκολία υλοποίησης των άλλων λειτουργικών κλάσεων. Περιέχει, σε μορφή διανύσματος, όλες τις πιθανές θέσεις στις οποίες μπορεί να κατευθυνθεί το αμαξάκι πάνω στον χάρτη. Το αντικείμενο της είναι και αυτό ένα διάνυσμα, το οποίο περιέχει την θέση του αμαξιού πάνω στον χάρτη.

Η κλάση `clientGUI` είναι η βασική κλάση δημιουργίας της διεπαφής του χρήστη και περιέχει και την `main` του πακέτου (φαίνεται και από την εικόνα 4.11 ότι αποτελείται από τις υπόλοιπες κλάσεις). Κατασκευάζει και αρχικοποιεί το παράθυρο, τα κουμπιά, τον χάρτη τύπου `ClientGUIMapPanel`, και ξεκινάει το νήμα τύπου `ClientGUIDrawingThread` που θα διαχειρίζεται τον χάρτη και τις εντολές.

Η κλάση `ClientGUIMapPanel` είναι προέκταση της κλάσης `javax.swing.JPanel`. Μέσα λοιπόν σε αυτό το `panel`, έχει ως σκοπό την σχεδίαση και αρχικοποίηση του χάρτη (το `background` είναι `jpg` από το πακέτο), των θέσεων στις οποίες μπορεί να κατευθυνθεί το αμαξάκι, την ζωντανή θέση του, καθώς και κάποιες γραμμές `auto correction`. Αυτά τα στοιχεία τα λαμβάνει από άλλες κλάσεις όπως την `pointsInTrackEnum`, `DriverAI`, `MyConstants`, ή αν αλλάξουν σύμφωνα με την `ClientGUIDrawingThread`. Τα παραπάνω επιτυγχάνονται σε επανάληψη με την εκτέλεση της κληροδοτημένης μεθόδου του `JPanel`, `paint`.

Τέλος, η κλάση `ClientGUIDrawingThread` κληρονομεί την κλάση `Thread` της Java. Όπως αναφέρθηκε προηγουμένως, ένα νήμα αυτής της κλάσης εκτελείται παράλληλα με την διεπαφή και έχει ως σκοπό την ανανέωση του `map panel` σε ζωντανό χρόνο καθώς και την αποστολή των εντολών οδήγησης. Αυτό υλοποιείται στην κληροδοτημένη μέθοδο `run`, η οποία τρέχει από την στιγμή που δοθεί σήμα εκκίνησης (`.start`) από την κλάση `clientGUI`. Προτού όμως γίνει αυτό, με την μέθοδο `setDrawingPanel` ορίζεται το `map panel`, ώστε να μπορεί να το μεταχειριστεί το νήμα. Μετά την εκκίνησή του, διαρκώς ενημερώνεται με την καινούργια θέση του αμαξιού, χρησιμοποιώντας ένα `proxy` του `CarPositionService`. Στην συνέχεια, ανανεώνει το `map panel` με την καινούργια θέση του αυτοκινήτου και την θέση στόχου που του έχει οριστεί. Αν αυτή είναι μια καινούργια θέση, καλεί την `driverAI.drivePathFinder`, ώστε να βρει την κατάλληλη εντολή για να την στείλει στο αμαξάκι μέσω της κλάσης `HTTPArduino`. Όπως φαίνεται αυτό το νήμα ενοποιεί τις περισσότερες κλάσεις, ώστε να εκτελέσουν έναν κοινό σκοπό.



# 5

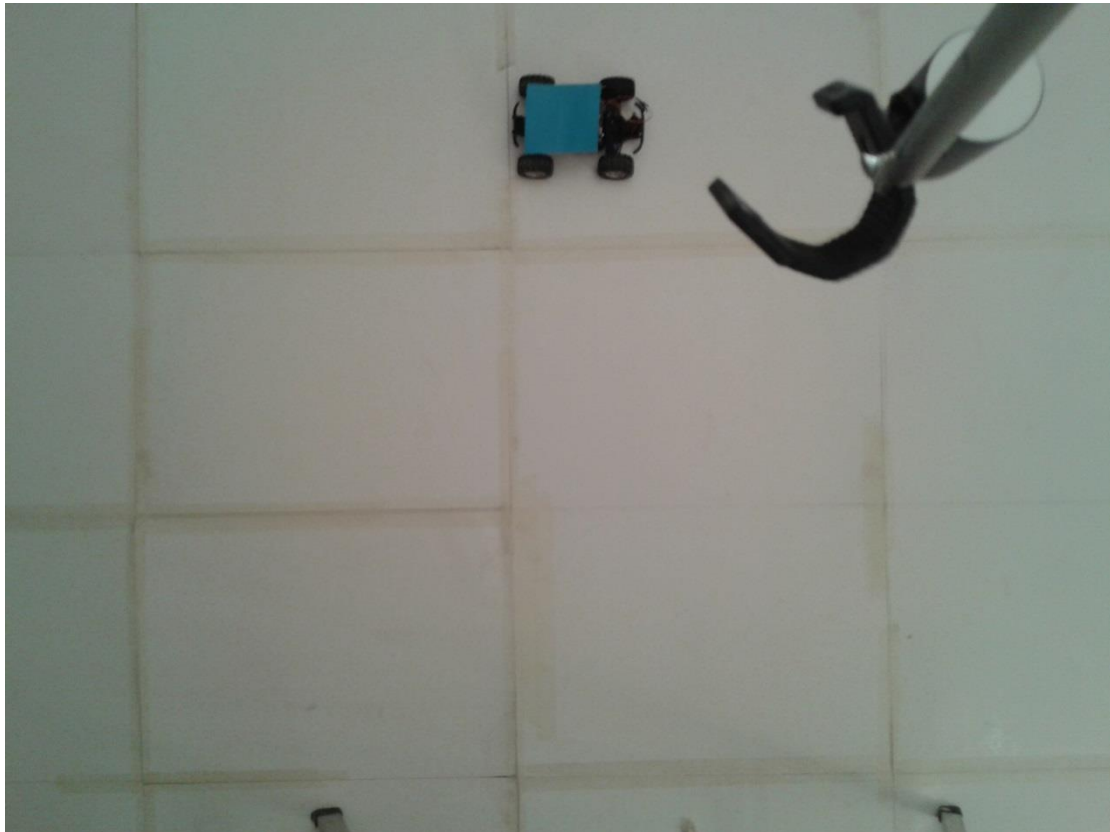
## *Εγχειρίδιο χρήσης της εφαρμογής*

### *5.1 Οδηγίες εκκίνησης*

Προτού ο χρήστης εκτελέσει την εφαρμογή πρέπει να σιγουρευτεί ότι η διάταξη του συστήματος είναι σωστή, αλλιώς μπορεί να εμφανιστούν μηνύματα λάθους. Η κάμερα της εργασίας πρέπει να τοποθετηθεί πάνω από μία άσπρη «πίστα», στο κέντρο περίπου 2.5 μέτρα ψηλά, ώστε να δείχνει ολόκληρη την πίστα και μόνο αυτή(βλ. εικόνα 5.1). Πρέπει έπειτα να συνδεθεί μέσω USB στον ηλεκτρονικό υπολογιστή στον οποίο θα εκτελέσουμε την εφαρμογή, καθώς και αυτός πρέπει να είναι συνδεδεμένος στο ίδιο τοπικό δίκτυο που συνδέεται το όχημα. Τέλος, ο χρήστης πρέπει να σιγουρευτεί ότι το όχημα είναι ενεργοποιημένο και έτοιμο για λειτουργία και να επιλέξει πριν την εκτέλεση του προγράμματος το χρώμα το οποίο επιθυμεί να ανιχνεύσει η εφαρμογή. Αυτό πραγματοποιείται αλλάζοντας τα ορίσματα στην κλήση της συνάρτησης `jMyronCamera.trackColor(r, g, b, s)` [project: JMyron java file: CameraApp.java line 46] όπου “r,g,b” οι συνιστώσες του χρώματος στο χώρο RGB και “s” η ευαισθησία της ανίχνευσης.

Όταν ο χρήστης έχει ολοκληρώσει τα παραπάνω βήματα, μπορεί να προχωρήσει στην εκτέλεση και χρήση της εφαρμογής. Αυτό το επιτυγχάνει εκκινώντας πρώτα τον Apache

server και εκτελώντας έπειτα το CameraApp.jar ώστε να αρχίσει η αναγνώριση θέσης του αμαξιού και έπειτα, το DhmasClientGUI.jar.

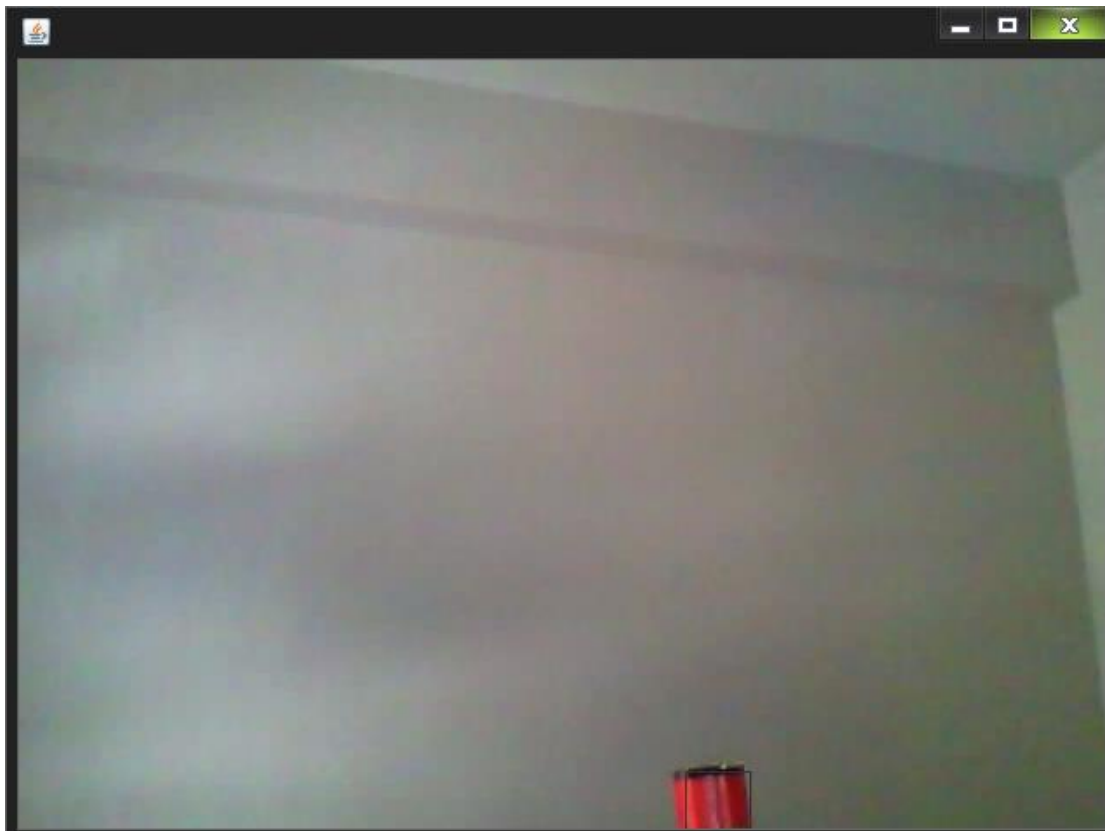


Εικόνα 5.1: Η πίστα για το όχημα, και η κάμερα από πάνω της.

## ***5.2 Οδηγίες χρήσης***

### CameraApp

Η χρήση αυτού του παραθύρου είναι μόνο για να μπορεί να ελέγξει ο χρήστης αν η κάμερα δουλεύει και αν βρίσκεται στην σωστή θέση και για να βλέπει ζωντανά την τρέχον αναγνώριση του οχήματος. Το παράθυρο αυτό περιέχει μόνο το stream της συνδεδεμένης κάμερας, στο οποίο το ζητούμενο αντικείμενο (το αμαξάκι στην περίπτωση αυτή) παρουσιάζεται με ένα γκρι παραλληλόγραμμο γύρω του, όπως και φαίνεται στη εικόνα 5.2.



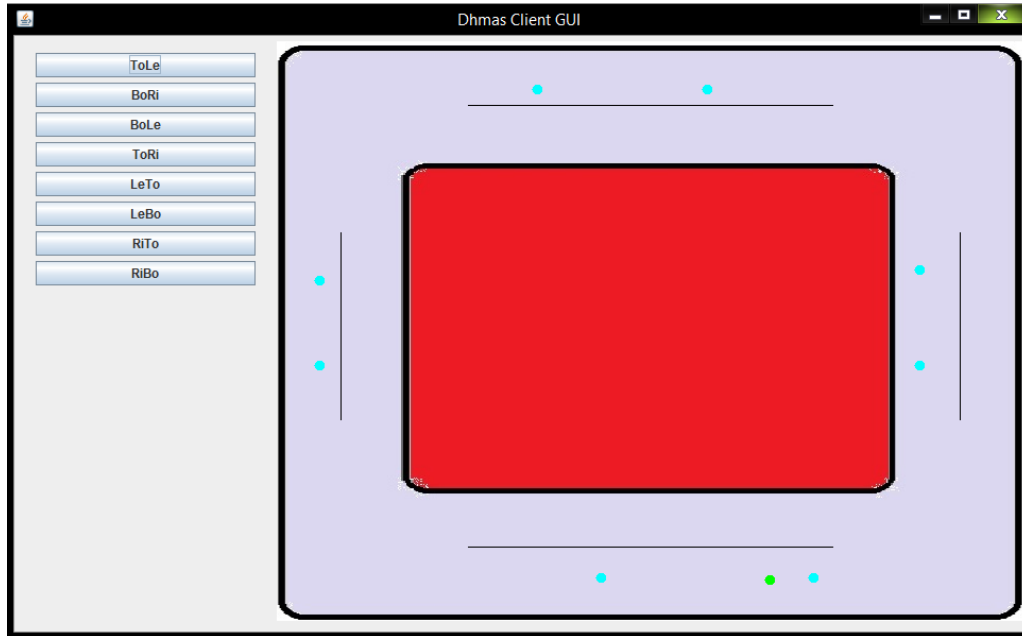
Εικόνα 5.2: screenshot κατά την εκτέλεση του CameraApp ενώ φαίνεται η ανίχνευση ενός κόκκινου αντικείμενου

### Dhmas Client GUI

Αυτό το παράθυρο αφορά την αλληλεπίδραση του χρήστη με το σύστημα. όπως φαίνεται και στην εικόνα 5.3, το παράθυρο χωρίζεται σε δύο μέρη:

- τον χάρτη στα δεξιά, ο οποίος δείχνει την νοητή πίστα και το μονοπάτι που μπορεί να ακολουθήσει το αμαξάκι (γκρι κομμάτι). Δείχνει επίσης τα σημεία προς τα οποία μπορεί να κατευθυνθεί το αμαξάκι από εντολή του χρήστη, με τουρκουάζ τελεία. Με την πράσινη τελεία, περιγράφεται η θέση του αμαξιού κάθε χρονική στιγμή. Τέλος παρουσιάζονται και τέσσερις γραμμές διόρθωσης για μελλοντική χρήση.
- τα κουμπιά στα αριστερά, με το όνομα των οποίων περιγράφεται η θέση στην οποία μπορεί να πάει το αμαξάκι, αυτές που φαίνονται δηλαδή στον χάρτη με τουρκουάζ χρώμα. Για παράδειγμα το κουμπί με περιγραφή “ToLe”, περιγράφει την πάνω αριστερή θέση/κουκίδα καθώς το ToLe είναι συντομογραφία του λατινικού Top Left.

Έτσι ο χρήστης πατώντας ένα από αυτά τα κουμπιά, δίνει την εντολή στο αμαξάκι να κινηθεί σε αυτή την θέση σύμφωνα με τον αλγόριθμο του προγράμματος, παρακολουθώντας το ζωντανά.



Εικόνα 5.3: screenshot κατά την εκτέλεση του Client GUI κατά την διάρκεια ανίχνευσης της θέσης ενός αντικειμένου (πράσινη τελεία).



# 6

## *Επίλογος*

### *6.1 Σύνοψη και συμπεράσματα*

Ο στόχος της εργασίας ολοκληρώθηκε επιτυχώς και αυτός πρωτίστως ήταν η μελέτη διαφόρων τεχνολογιών και η διασύνδεση τους ώστε να καταφέρουμε να κατασκευάσουμε ένα σύμπλεγμα υποσυστημάτων που θα λειτουργούν αρμονικά και θα υλοποιούν σε πραγματικό χρόνο την λογική απομακρυσμένης πλοήγησης κινητού οχήματος. Προκειμένου να φτάσουμε στο επιθυμό αποτέλεσμα έγιναν πολλαπλές δοκιμές και τροποποιήσεις ώστε να πετύχουμε την καλύτερο δυνατό αποτέλεσμα, χωρίς αυτό να σημαίνει ότι οι δυνατότητες για μελλοντικές επεκτάσεις και βελτιώσεις είναι λίγες. Αξιο αναφοράς είναι ότι τα παραπάνω έγιναν εφικτά με χρήση εργαλείων ανοιχτού κώδικα διαθέσιμα σε όλους μας και με μικρό έως μηδενικό κόστος.

## **6.2 Μελλοντικές επεκτάσεις**

Η εργασία αυτή έχει το περιθώριο για κάποιες βελτιώσεις και επεκτάσεις οι οποίες παρατίθενται παρακάτω:

### **6.2.1 Βελτίωση και αναβάθμιση του συστήματος πλοήγησης του οχήματος**

Με περαιτέρω μελέτη, ο αλγόριθμος επιλογής των οδηγιών του οχήματος μπορεί να βελτιωθεί και να επεκταθεί έτσι ώστε να δίνεται η δυνατότητα στο όχημα να κινηθεί προς τα πίσω(κάτι το οποίο είναι ήδη υλοποιημένο από πλευράς hardware) σε περίπτωση που αυτή η διαδρομή είναι βέλτιστη. Οι μέθοδοι αυτές έχουν περιγραφεί στον κώδικα, αλλά δεν έχουν υλοποιηθεί ακόμα.

Επιπρόσθετα, καθώς μερικές φορές εμφανίζεται ένα μικρό σφάλμα στην κατεύθυνση του αυτοκινήτου, μια σημαντική επέκταση θα ήταν η υλοποίηση κάποιου είδους σταθεροποιητή, ο οποίος να αναγνωρίζει την ύπαρξη σφάλματος, και να το διορθώνει σε πραγματικό χρόνο.

Μια προσπάθεια υλοποίησης του έχει είδη γίνει. Το πρόγραμμα δημιουργεί και κάποιες γραμμές διόρθωσης της πορείας, οι οποίες όμως δεν έχουν χρησιμοποιηθεί σωστά.

Επίσης θα μπορούσαν να προστεθούν εμπόδια στην πίστα και να γίνει χρήση αισθητήρων για την επίτευξη της αυτόματης πλοήγησης.

### **6.2.2 Βελτίωση του συστήματος διεπαφής του χρήστη**

Η διεπαφή χρήστη να μπορούσε να δεχτεί αρκετές βελτιώσεις όσον αφορά την ευχρηστία και τις δυνατότητες παραμετροποίησης/αρχικοποίησης της εφαρμογής. Θα μπορούσε ο χρήστης να επιλέγει όποιο σημείο προορισμού αυτός επιθυμεί και να αφαιρεθούν τα προκαθορισμένα σημεία, όπως επίσης να επιλέγει και το χρώμα προς ανίχνευση μέσω παλέτας ή του παραθύρου του stream της κάμερας. Επιδέχεται επίσης αρκετές αισθητικές βελτιώσεις, καθώς σχεδιάστηκε αρκετά μινιμαλιστικά για να εξυπηρετήσει τις δοκιμές και τους ελέγχους καλής λειτουργίας του συστήματος.

# 7

## *Βιβλιογραφία*

[1] Γεώργιος Αλεξίου Καθηγητής Τμήματος Μηχανικών Η/Υ & Πληροφορικής Πανεπιστημίου Πατρών, Μικροεπεξεργαστές, Ελληνικό Ανοικτό Πανεπιστήμιο, Σχολή Θετικών Επιστημών και Τεχνολογίας, 2001

[2] Μικροελεγκτές. Website: <http://www.eln.teilam.gr/sites/default/files/Lesson03.pdf>

[3] Microcontroller. Website: <https://en.wikipedia.org/wiki/Microcontroller>

[4] Μικροελεγκτές στο βιομηχανία: Δυνατότητες και εφαρμογές [www.plant-management.gr](http://www.plant-management.gr)

Μιχαήλ Μακροδημήτρης: Υποψήφιος Διδάκτορας ΕΜΠ

Ευάγγελος Παπαδόπουλος: Αναπληρωτής Καθηγητής ΕΜΠ

Link: [https://nereus.mech.ntua.gr/Documents/pdf\\_ps/pm4.pdf](https://nereus.mech.ntua.gr/Documents/pdf_ps/pm4.pdf)

[5] Μικροελεγκτής. Website: <https://el.wikipedia.org/wiki/Μικροελεγκτής>

[6] Η ιστορία των μικροεπεξεργαστών. Website:

<http://cgi.di.uoa.gr/~std03006/Welcome.html>

[7] Μικροελεγκτής Arduino | MicroPlanet. Website:

[www.microplanet.gr/tutorials/microcontrollers/arduino](http://www.microplanet.gr/tutorials/microcontrollers/arduino)

[8] Εισαγωγή στο Arduino – Το απόλυτο geek toy. Website:

<https://deltahacker.gr/arduino-intro/>

[9] Εισαγωγή στο Arduino. Website:

<http://hlektrologia.gr/%CE%B5%CE%B9%CF%83%CE%B1%CE%B3%CF%89%CE%B3%CE%AE-%CF%83%CF%84%CE%BF-arduino/>

[10] Μαργαρίτα Παπαευθυμίου. Διπλωματική «Ανάπτυξη τηλεχειριστηρίου σε συσκευή ipad η οποία ελέγχει ελικόπτερο με μικροελεγκτή Arduino». Μάϊος 2013

[11] Arduino. Website: <https://el.wikipedia.org/wiki/Arduino>

[12] Αρβανίτη Γερασίμου. Διπλωματική «Ανίχνευση και παρακολούθηση κίνησης (motion detection and tracking)». Ιανουάριος 2012

[13] Video tracking. Website: [https://en.wikipedia.org/wiki/Video\\_tracking](https://en.wikipedia.org/wiki/Video_tracking)

[14] Σταμούλη Μαριάννα. Διπλωματική «Εξαγωγή κινούμενου αντικειμένου σε βίντεο». 2008

[15] Processing. Website: <https://processing.org/>

[16] Blob detection. Website: [https://en.wikipedia.org/wiki/Blob\\_detection](https://en.wikipedia.org/wiki/Blob_detection)

[17] What is blob tracking? Website: <http://www.wisegeek.com/what-is-blob-tracking.htm>

[18] Blob detection. University of Maryland. Website:

[www.cs.umd.edu/~djacobs/CMSC426/Blob.pdf](http://www.cs.umd.edu/~djacobs/CMSC426/Blob.pdf)

[19] Jmyron (Myron WebcamXtra). Website: <http://webcamxtra.sourceforge.net/>

[20] Lab: DC Motor Control Using an H-Bridge. Website:

<http://itp.nyu.edu/physcomp/labs/motors-and-transistors/dc-motor-control-using-an-h-bridge/>

[21] Basics of DC Motor Drive and Speed Control for Robots. Website:

<http://embedjournal.com/basics-of-dc-motor-drive-and-speed-control-for-robots/>

[22] Προδιαγραφές ολοκληρωμένου κυκλώματος L293D

[https://www.arduino.cc/documents/datasheets/H-bridge\\_motor\\_driver.PDF](https://www.arduino.cc/documents/datasheets/H-bridge_motor_driver.PDF)