



Εθνικό και Καποδιστριακό
ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Διπλωματική Εργασία

Τμήμα Φυσικής και Πληροφορικής & Τηλεπικοινωνιών

ΜΔΕ στον Ηλεκτρονικό Αυτοματισμό

ΜΕΛΕΤΗ, ΣΧΕΔΙΑΣΗ, ΑΝΑΠΤΥΞΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΑΠΟΜΑΚΡΥΣΜΕΝΟΥ ΕΛΕΓΧΟΥ ΓΙΑ ΜΙΚΡΟ ΕΠΙΓΕΙΟ ΟΧΗΜΑ ΚΑΙ ΣΧΕΤΙΚΩΝ ΑΥΤΟΜΑΤΙΣΜΩΝ



Μπεκρής Νικόλαος 2019

Πληροφορίες Διπλωματικής Εργασίας

Σχολή : Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Τμήμα : Φυσικής και Πληροφορικής & Τηλεπικοινωνιών

Τίτλος μεταπτυχιακού : Ηλεκτρονικός Αυτοματισμός – ΗΑ

Τίτλος διπλωματικής εργασίας : Μελέτη, Σχεδίαση, Ανάπτυξη και Υλοποίηση
Συστήματος Απομακρυσμένου Ελέγχου για μικρό
επίγειο όχημα και σχετικών Αυτοματισμών

Όνομα συγγραφέα : Μπεκρής Νικόλαος

A.M. : 2016518

Επιβλέπον καθηγητής : Κούτουλας Δημήτρης

Μέλη τριμελούς επιτροπής : Κούτουλας Δημήτρης, Τσίπουρας Αριστείδης,
Αλωνιστιώτη Αθανασία

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ.....	1
1. ΠΡΟΛΟΓΟΣ.....	3
1.1 Σκοπός.....	3
1.2 Περιγραφή.....	3
2. PROLOGUE.....	4
2.1 Purpose.....	4
2.2 Description	4
3. ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ	5
3.1 Stanford Cart 1961	5
3.2 Robot Fire Cart 1962	7
3.3 Lunokhod I 1970	7
3.4 BigDog 2005.....	8
3.5 Curiosity 2012.....	9
3.6 ΡΟΜΠΟΤΙΚΟ ΟΧΗΜΑ PANDORA 2013.....	10
3.7 Ρομποτικό όχημα R1 2018	10
3.8 Almira-class USAR vehicle	11
3.9 R+D Project Robot	12
4. Η ΚΑΤΑΣΚΕΥΗ.....	13
4.1 Arduino	13
4.1.1 Ιστορικό	13
4.1.2 Επίσημες πλακέτες	13
4.1.3 Λογισμικό	15
4.1.4 Ανάπτυξη	18
4.1.5 Τροφοδοσία.....	20
4.1.6 Μνήμη	21
4.1.7 Είσοδοι - Έξοδοι.....	21
4.2 Project Car	24
4.2.1 Κινητήρας DC.....	25
4.2.2 H-Bridge	26
4.2.3 Lights.....	28

4.2.4	DHT11 Humidity Sensor	28
4.2.5	FPV Camera	29
4.2.6	Power Board	30
4.2.7	Building Project Car	31
4.2.8	Controller.....	36
4.3	Code.....	39
4.3.1	Arduino UNO - Car	39
4.3.2	Arduino Nano - Controller	42
4.4	Συνδεσμολογία.....	49
5.	Μελλοντικές Επεκτάσεις	54
6.	Βιβλιογραφία	56

1. ΠΡΟΛΟΓΟΣ

Από παλιά έχουν γίνει προσπάθειες αυτοματισμού λειτουργιών και έχουν δημιουργηθεί πολλά αυτόματα συστήματα (π.χ. αυτόματη ηλεκτρική σκούπα) με σκοπό τη διευκόλυνση των ανθρώπων. Έτσι κι εμείς στη παρούσα εργασία, φτιάξαμε ένα μικρό και ευέλικτο όχημα (βλ. *Κεφάλαιο 4.2*), τοποθετήσαμε σε αυτό μια μικρή ασύρματη κάμερα (βλ. *Κεφάλαιο 4.2.5*).

1.1 Σκοπός

Σκοπός μας είναι η εποπτεία κάποιου χώρου από απόσταση με οπτικά μέσα και ο απομακρυσμένος έλεγχος του οχήματος με νέες τεχνολογίες (π.χ. Wi-Fi communication). Μπορεί να χρησιμοποιηθεί για την παρακολούθηση χώρων, χωρίς να χρειάζεται εκεί η παρουσία ανθρώπου, παρά μόνο ένας χειριστής του οχήματος εντός εμβέλειας (έως 100 μέτρα μακριά), καθώς διαθέτει ασύρματη κάμερα με live feed και δυνατότητα εγγραφής βίντεο. Επίσης το όχημα θα φέρει πάνω του αισθητήρες (π.χ. θερμοκρασίας), τα δεδομένα των οποίων συλλέγονται και αποστέλλονται σε οθόνη πίσω στον χειριστή.

Χρησιμεύει για ερευνητικούς σκοπούς, σε περιπτώσεις όπου δεν έχει πρόσβαση ο άνθρωπος, λόγω περιορισμένου χώρου ή επικίνδυνου – αφιλόξενου περιβάλλοντος (π.χ. σπήλαια). Τέλος, είναι πρακτικό διότι είναι μικρό και ευέλικτο και διαθέτει καλή αυτονομία.

1.2 Περιγραφή

Για την κατασκευή χρησιμοποιήσαμε ένα [όχημα](#) διαστάσεων 260 mm x 150 mm. Η τροφοδοσία του, καθώς της κύριας πλακέτας και των υπόλοιπων κυκλωμάτων γίνεται με μπαταρίες μεγάλης χωρητικότητας, για μεγάλη αυτονομία. Η επικοινωνία του Project για χειρισμό, πραγματοποιείται με τεχνολογία Wi-Fi (NRF24L01 board). Έχουμε τοποθετημένη την κάμερα στο μπροστινό μέρος πάνω σε περιστρεφόμενη βάση για καλύτερη εποπτεία του χώρου. Υπάρχουν 2 κύρια LED μπροστά, για να ανάβουν σε περίπτωση χαμηλού φωτισμού. Επίσης κατασκευάστηκε [χειριστήριο](#) για τον έλεγχο όλων των λειτουργιών. Τέλος, όλα αυτά συνδέονται σε δύο απλές μητρικές πλακέτες ανοικτού κώδικα ([Arduino](#) UNO & Arduino Nano), με ενσωματωμένο μικροελεγκτή (ATmega328P).

Στο Κεφάλαιο 4 περιγράφονται αναλυτικότερα όλα τα υλικά που αναφέρθηκαν παραπάνω. Επίσης, αναλύεται ο κώδικας και η συνδεσμολογία όλων των κυκλωμάτων.

2. PROLOGUE

In the past, attempts have been made to automate functions and many automatic systems (eg, a vacuum cleaner) have been created to help people. So on this research paper, we built a small and flexible vehicle (*see Chapter 4.2*) and placed a small wireless camera on it (*see Chapter 4.2.5*).

2.1 Purpose

Our aim is to monitor a space remotely by optical means and to remotely control the vehicle with new technologies (eg Wi-Fi communication). It can be used to monitor areas without the need for human presence, but only one vehicle operator within range (up to 100 meters away) as it has a wireless live video camera and video recording capability. The vehicle will also carry on it sensors (eg temperature), the data of which are collected and sent to a screen back to the operator.

It serves for research purposes, for places that are not accessible, because of limited space or dangerous - inhospitable environments (eg caves). Finally, it is practical because it is small and flexible and has good autonomy.

2.2 Description

For construction we used a 260mm x 150mm [vehicle](#). The power supply, as well as the main board and the rest of the circuits, are powered with high-capacity batteries for great autonomy. Project communication is handled with Wi-Fi technology (NRF24L01 board). We have the camera positioned at the front on a rotating base for better surveillance of the area. There are 2 main LEDs in the front to light up, in low light cases. A [Controller](#) was also built to control all functions. Finally, all of these are connected to two simple open source motherboards ([Arduino UNO](#) & [Arduino Nano](#)) with a built-in microcontroller (ATmega328P).

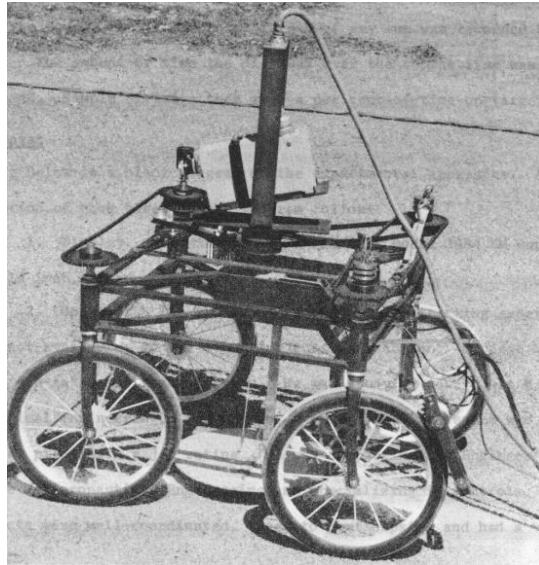
Chapter 4 describes in more detail all the materials mentioned above and analyzes the code and the connection of all circuits.

3. ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

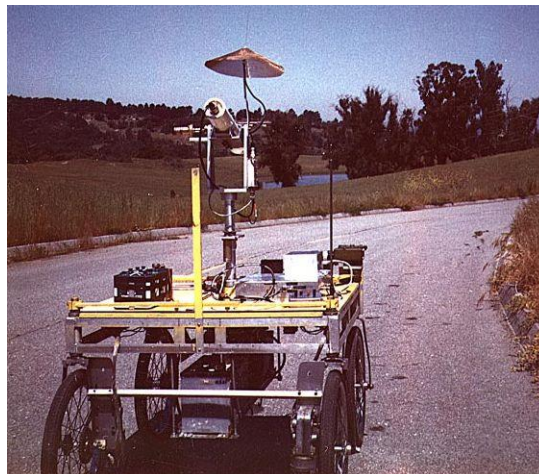
Σε αυτή την ενότητα θα αναφέρουμε παραδείγματα από την ιστορία των κινούμενων ρομπότ. Ο κλάδος της Ρομποτικής παρουσιάζει ραγδαία εξέλιξη τα τελευταία χρόνια, καθώς οι εφαρμογές της συνεισφέρουν στη μείωση του κόστους, την αύξηση της παραγωγικότητας, τη βελτίωση της ποιότητας των παραγόμενων προϊόντων και στην απαλλαγή του ανθρώπου από πολλές επικίνδυνες και ανθυγιεινές εργασίες (π.χ. ορυχεία, πυρηνικοί αντιδραστήρες).

3.1 Stanford Cart 1961

Κατασκευάστηκε αρχικά από τον James L. Adams ως ερευνητικό όχημα για απομακρυσμένες αποστολές στο φεγγάρι. Το Stanford Cart είχε τέσσερις μικρές ρόδες ποδηλάτου που συνδέονταν με ηλεκτρικούς κινητήρες και τροφοδοτούνταν από μια μπαταρία αυτοκινήτου. Επίσης πραγματοποιούσε λήψεις με μια τηλεοπτική κάμερα με σταθερή θέα προς τα εμπρός. Δοκιμές διεξήχθησαν με το Cart να στρίβει τόσο με δύο ρόδες (μπροστινές), όπως ένα αυτοκίνητο, όσο και με τέσσερις, στο οποίο οι τροχοί και η τηλεοπτική κάμερα περιστρέφονταν μαζί. Το Stanford Cart συνδεόταν με ένα πολύ μακρύ καλώδιο σε μια κονσόλα ελέγχου, με οθόνη τηλεόρασης για τον έλεγχο της οδήγησης και της ταχύτητας. Ο Adams διερευνούσε τη δυνατότητα ελέγχου του οχήματος, αποφεύγοντας τα εμπόδια με διάφορους συνδυασμούς τριών συντελεστών: της καθυστέρησης, της επικοινωνίας και της ταχύτητας. Όταν οι εντολές διεύθυνσης καθυστερούσαν στην επικοινωνία, υπήρχε μια τάση για το χειριστή πάνω στο τιμόνι και έχανε τον έλεγχο. Μεταξύ άλλων, ο Adams έδειξε στη διατριβή του ότι με καθυστέρηση επικοινωνίας, που αντιστοιχεί στο ταξίδι μετ'επιστροφής στη Σελήνη (περίπου 2 1/2 δευτερόλεπτα), το όχημα δεν μπορεί να ελεγχθεί με αξιοπιστία εάν ταξιδεύει γρηγορότερα από περίπου 0,3 km/h.



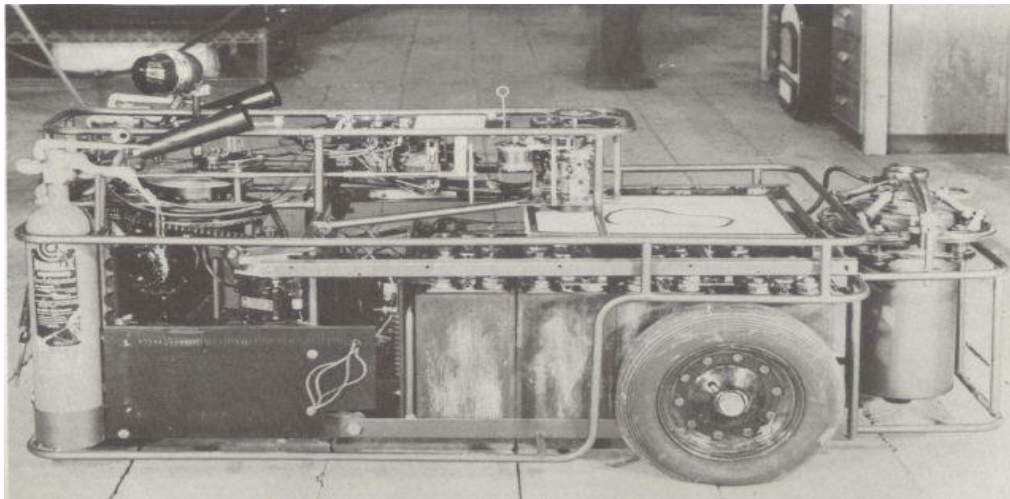
Εικόνα 3.1: Το Stanford Cart 1960-1961



Εικόνα 3.2: Το Stanford Cart 1964-1971 εφοδιασμένο με τεχνική νοημοσύνη

3.2 Robot Fire Cart 1962

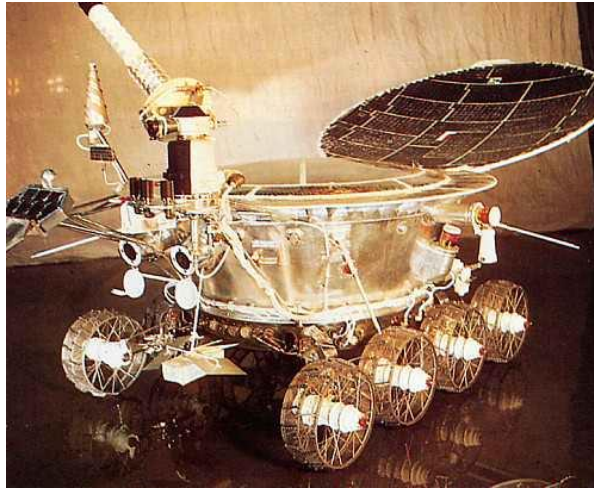
Σχεδιάστηκε και κατασκευάστηκε από τον Meredith Thring στο Queen Mary College. Ήταν ένα πυροσβεστικό ρομπότ που πλοηγούνταν ελεύθερα στο χώρο χρησιμοποιώντας σήματα από μια γυροσκοπική πυξίδα για τη μέτρηση της απόστασης. Όταν εντόπιζε μια εστία φωτιάς, μέσω του φωτοκύτταρου, πήγαινε κατευθείαν προς αυτήν και όταν ο διμεταλλικός διακόπτης ανίχνευε τη θερμότητα της φλόγας, το ρομπότ σταματούσε και τοποθετούσε το ακροφύσιο του πυροσβεστήρα πάνω στη φλόγα για να τη σβήσει. Η ιδέα ήταν να αναπτυχθεί ένα πλήρως αυτόματο ρομπότ "νυχτοφύλακας" που θα μπορούσε να περιπλανιέται γύρω από μια αποθήκη και να κοιτάζει έξω για πυρκαγιά.



Εικόνα 3.3: Το Robot Fire Cart απενεργοποιημένο και σε δράση

3.3 Lunokhod I 1970

Η Σοβιετική Ένωση εξερεύνησε την επιφάνεια της Σελήνης, με το Lunokhod 1, ένα σεληνιακό ρόβερ. Το Lunokhod 1 ήταν το πρώτο τηλεκατευθυνόμενο ρομπότ που προσγειώθηκε σε ένα άλλο ουράνιο σώμα. Το διαστημικό σκάφος προσγειώθηκε μαλακά στην επιφάνεια της Σελήνης στη Θάλασσα των Βροχών στις 17 Νοεμβρίου. Το ρόβερ λειτουργούσε κατά τη σεληνιακή ημέρα, σταματώντας περιστασιακά για να επαναφορτίσει τις μπαταρίες του μέσω των ηλιακών συλλεκτών. Τη νύχτα το Lunokhod 1 έκανε αδρανοποίηση μέχρι την επόμενη ανατολή του ηλίου.



Εικόνα 3.4: Το Lunokhod I

3.4 BigDog 2005

Είναι ένα τετράποδο ρομπότ, το οποίο κατασκευάστηκε από τη Boston Dynamics. Είναι ικανό να διασχίζει δύσκολα εδάφη. Η ταχύτητά του φτάνει μέχρι 6.4 km/h και μπορεί να κουβαλήσει εξοπλισμό βάρους 150kg.



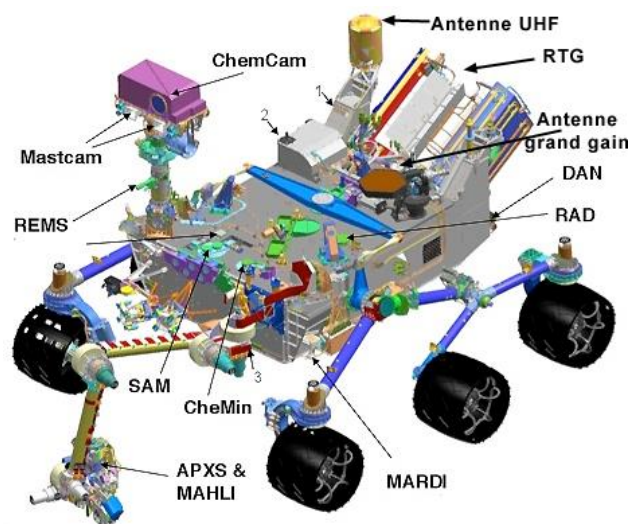
Εικόνα 3.5: BigDog

3.5 Curiosity 2012

Ένα εξάτροχο ρομποτικό όχημα στάλθηκε στον πλανήτη Άρη για να εξετάσει το κλίμα και την γεωλογία του και να διαπιστώσει εάν υπάρχουν ευνοϊκές περιβαλλοντολογικές συνθήκες για την ύπαρξη μικροβιακής ζωής, συμπεριλαμβανομένου του ρόλου του νερού και να ερευνήσει για τον αν ο πλανήτης έχει ευνοϊκές συνθήκες για μελλοντική εξερεύνηση από τους ανθρώπους. Το τροχοφόρο αυτό ρομπότ ζυγίζει περίπου ένα τόνο. Το όχημα εκτοξεύθηκε στις 26 Νοεμβρίου 2011 στις 10:02 EST από το ακρωτήριο Κανάβεραλ και προσεδαφίστηκε στην περιοχή Αιολίς Πάλους, στο κρατήρα Γκέιλ, στις 6 Αυγούστου 2012, 05:17 UTC. Το ρομπότ προσγειώθηκε περίπου 2,4 χιλιόμετρα μακριά από το προκαθορισμένο σημείο, σε μια περιοχή που ονομάστηκε Μπράντμπερι Λάντινγκ, μετά από ταξίδι απόστασης 563 εκατομμυρίων χιλιομέτρων.



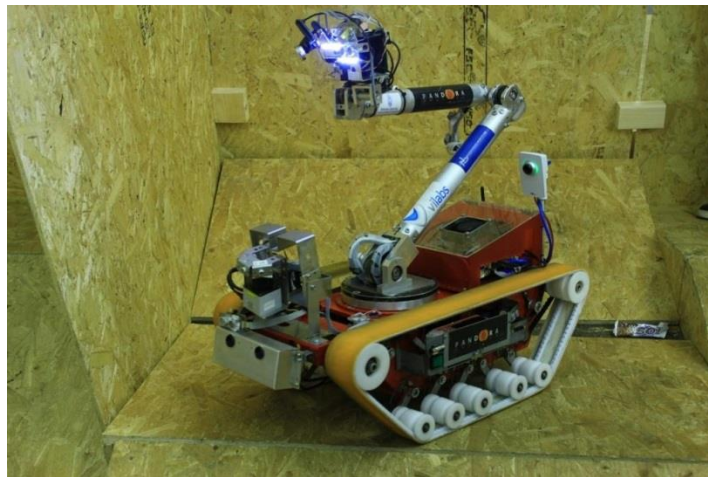
Εικόνες 3.6: Curiosity



3.6 ΡΟΜΠΟΤΙΚΟ ΟΧΗΜΑ PANDORA 2013

Η ερευνητική ομάδα ρομποτικής PANDORA (Program for the Advancement of Non Directed Operating Robotic Agents) του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης (ΤΗΜΜΥ-ΑΠΘ) ανέπτυξε μια πειραματική ρομποτική πλατφόρμα η οποία έχει την δυνατότητα (αυτόνομης και τηλεχειριζόμενης) πλοήγησης σε χώρους οι οποίοι έχουν υποστεί καταστροφή (π.χ. από σεισμό) και ανιχνεύει πιθανά θύματα.

Το ρομποτικό όχημα μπορεί να δώσει τη δυνατότητα σε συνεργεία διάσωσης να χαρτογραφήσουν έναν επικίνδυνο χώρο, να εντοπίσουν επιζώντες, να αναγνωρίσουν την κατάσταση στην οποία βρίσκονται, και να τους μεταφέρουν βασικές προμήθειες.



Εικόνα 3.7: Pandora 2013

3.7 Ρομποτικό όχημα R1 2018

Δύο πρώην μηχανικοί της Google κατασκεύασαν ένα ηλεκτροκίνητο ρομποτικό όχημα που διαθέτει σειρά προηγμένων τεχνολογιών για να μπορεί να κινείται αυτόνομα.

Το R1, όπως ονομάζεται το όχημα αυτό δημιουργήθηκε για να μεταφέρει προϊόντα (τα ψώνια του σούπερμαρκετ, φαγητό, κλπ) στην πόρτα του πελάτη. Το όχημα αυτό εκτός από τα σχετικά συστήματα αυτόνομης κίνησης για να εκτελεί την αποστολή του διαθέτει και μια σειρά από τεχνολογίες και συστήματα για να παραδίδει με ασφάλεια το φορτίο του στον άνθρωπο για τον οποίο προορίζεται.

Το R1 διαθέτει τεχνολογία αναγνώρισης προσώπων για να αναγνωρίζει τον πελάτη ο οποίος θα λαμβάνει ένα κωδικό στο κινητό του τηλέφωνο και μέσω της σχετικής εφαρμογής όταν το όχημα φτάσει στην πόρτα του θα χρησιμοποιεί τον

κωδικό για να ανοίξει η πόρτα του οχήματος και να πάρει τα πράγματα που έχει παραγγείλει. Η εφαρμογή αυτή θα ενημερώνει τον πελάτη για το που βρίσκεται ανά πάσα στιγμή το R1 ώστε να γνωρίζει πότε πλησιάζει στο σπίτι του και να βγει για να παραλάβει την παραγγελία του.

Οι δυο μηχανικοί ίδρυσαν μια start up εταιρεία, την Nuro, η οποία κατάφερε να συγκεντρώσει χρηματοδότηση ύψους περίπου 70 εκατ. δολαρίων με την οποία θα ξεκινήσουν άμεσα την κατασκευή ενός στόλου R1 που θα είναι έτοιμος να αναλάβει υπηρεσία στους δρόμους το 2022.



Εικόνα 3.8: R1 2018

3.8 Almira-class USAR vehicle

Η ρομποτική πλατφόρμα της USAR (Urban Search and Rescue) αποτελεί ένα πρωτότυπο ρομπότ τύπου Almira (Εικόνα 4). Το ιδιαίτερο χαρακτηριστικό του είναι το γεγονός ότι κάθε τροχός του αποτελείται από 3 υπό-τροχούς (12 σε σύνολο) σε τριγωνική διάταξη. Επιπλέον, κάθε σετ τριών τροχών σε κάθε άξονα, δύναται να περιστρέφεται ανεξάρτητα από την περιστροφή του κεντρικού άξονα (που αποτελεί το κέντρο της κάθε τριγωνικής διάταξης). Αυτό επιτρέπει στο όχημα να κινείται χωρίς καμία δυσκολία σε ομαλές επιφάνειες, ενώ παράλληλα έχει τη δυνατότητα να ανέβει σκάλες, καθώς και απότομες επικλινείς επιφάνειες έως και 75 μοιρών.

Η πλατφόρμα είναι εφοδιασμένη με υπερηχητικούς και υπέρυθρους αισθητήρες, καθώς και με μικρόφωνα. Επιπλέον, στο εμπρόσθιο και όπισθεν μέρος του ρομπότ είναι τοποθετημένες δύο ασύρματες βίντεο κάμερες για πιο πλήρη εποπτεία του χώρου, καθώς και για την αναζήτηση επιζώντων.



Εικόνα 3.9: USAR Vehicle

3.9 R+D Project Robot

Το R+D Project (Εικόνα 1) είχε ως στόχο να σχεδιάσει και να εφαρμόσει τις τεχνολογίες ελέγχου και συνεργασίας που απαιτούνται για την ανάπτυξη της πρώτης γενιάς ρομπότ για διάσωση. Προκειμένου να επιτευχθεί ο στόχος αυτός, δημιουργήθηκαν στεγανά ρομποτικά συστήματα για να χρησιμοποιηθούν σε επιχειρήσεις έρευνας και διάσωσης σε ρηχά ή επιφανειακά νερά.

Στην πρώτη κατάσταση λειτουργίας, το αυτόνομο ρομποτικό σύστημα έχει την δυνατότητα να καταγράφει πληροφορίες για την κατάσταση των επιζώντων καθώς και την διαδρομή που έχει εκτελέσει, και μόνο όταν δεν μπορεί να εντοπίσει επιζώντες, επιστρέφει στο κέντρο ελέγχου για επαναφόρτιση.

Στην δεύτερη κατάσταση λειτουργίας, με την βοήθεια ενός GPS και μίας μονάδας ελέγχου, το ρομπότ έχει την δυνατότητα να φιλτράρει τα πλάνα που παίρνει, καθώς και να διατηρεί αυτά που δείχνουν μεγάλη πιθανότητα ότι εντοπίστηκε επιζών, ενώ παράλληλα στο κέντρο ελέγχου, αποστέλλεται και η θέση αυτού. Με αυτόν τον τρόπο το ρομπότ επικεντρώνεται μόνο στα σημεία που είναι πιθανά να βρεθεί εγκλωβισμένος άνθρωπος.



Εικόνα 3.10: R+D Robot

4. Η ΚΑΤΑΣΚΕΥΗ

4.1 Arduino

Το Arduino είναι μια υπολογιστική πλατφόρμα βασισμένη σε μια απλή μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Pure Data, SuperCollider. Οι περισσότερες εκδόσεις του Arduino μπορούν να αγοραστούν προσυναρμολογημένες· το διάγραμμα και πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους.

4.1.1 Ιστορικό

Το 2005, ένα σχέδιο κίνησε προκειμένου να φτιαχτεί μία συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν πιο φθηνή από άλλα πρωτότυπα συστήματα διαθέσιμα εκείνη την περίοδο. Οι ιδρυτές Massimo Banzi και David Cueartielles ονόμασαν το σχέδιο από τον Arduino της Inrea και ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ιβρέα, κωμόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας· την ίδια περιοχή στην οποία στεγαζόταν η εταιρία υπολογιστών Olivetti

Το σχέδιο Arduino είναι μία διακλάδωση της πλατφόρμας Wiring για λογισμικό ανοικτού κώδικα και προγραμματίζεται χρησιμοποιώντας μια γλώσσα βασισμένη στο Wiring (σύνταξη και βιβλιοθήκες), παρόμοια με την C++ με απλοποιήσεις και αλλαγές, καθώς και ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE).

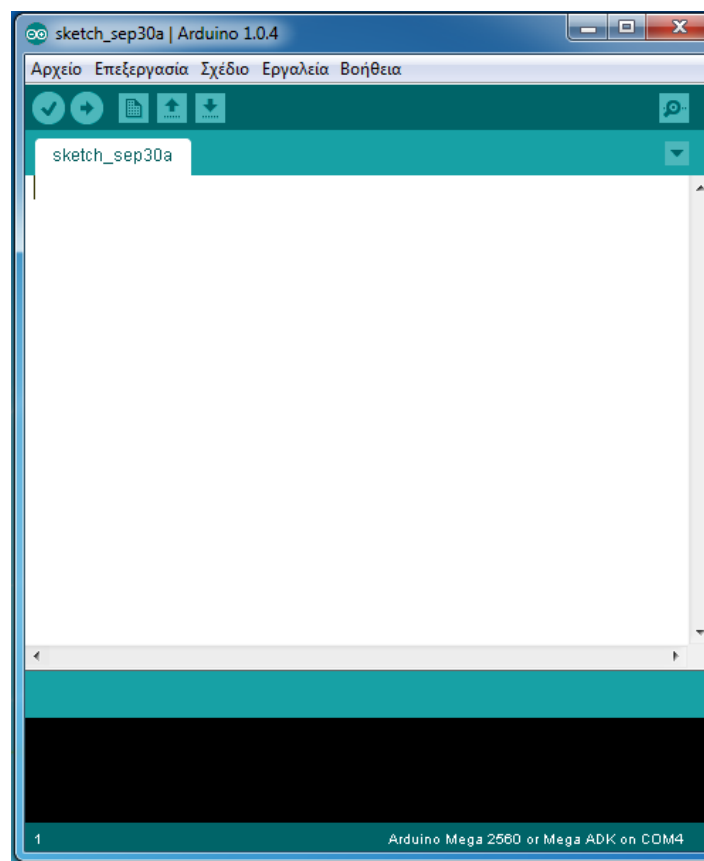
4.1.2 Επίσημες πλακέτες

Η πρωτότυπη πλακέτα του Arduino κατασκευάζεται από την Ιταλική εταιρία Smart Projects. Κάποιες πλακέτες με την μάρκα του Arduino έχουν σχεδιαστεί από την Αμερικάνικη εταιρία SparkFun Electronics .Δεκαέξι εκδοχές του Arduino Hardware έχουν χρησιμοποιηθεί εμπορικά μέχρι τώρα:

- Το Serial Arduino, προγραμματισμένο με μία σειριακή DE-9 σύνδεση χρησιμοποιώντας τεχνολογία ATmega8
- Το Arduino Extreme, με ένα USB interface για προγραμματισμό χρησιμοποιώντας τεχνολογία ATmega8
- Το Arduino Mini, μία έκδοση μινιατούρας του Arduino χρησιμοποιώντας τεχνολογία surface-mounted ATmega168
- Το Arduino Nano, ένα ακόμα πιο μικρό, USB τροφοδοτούμενη εκδοχή του Arduino χρησιμοποιώντας τεχνολογία surface-mounted ATmega168 (ATmega328 για την νεότερη έκδοση)
- Το LilyPad Arduino, ένα μινιμαλιστικό σχέδιο για εφαρμογές ένδυσης και E-textiles χρησιμοποιώντας τεχνολογία surface-mounted ATmega328
- Το Arduino NG, με ένα USB interface για προγραμματισμό και χρησιμοποιώντας τεχνολογία ATmega8
- Το Arduino NG plus, με ένα USB interface για προγραμματισμό και χρησιμοποιώντας τεχνολογία atmega168
- Το Arduino Bluetooth, με Bluetooth interface για προγραμματισμό χρησιμοποιώντας τεχνολογία ATmega168
- Το Arduino Diecimila, με ένα USB interface και χρησιμοποιεί τεχνολογία ATmega168 σε ένα DIP28 πακέτο
- Το Arduino Duemilanove (2009), χρησιμοποιεί τεχνολογία ATmega168 (ATmega328 για την καινούργια έκδοση) και τροφοδοτείται μέσω ενέργειας USB/DC, αυτόματα εναλλασσόμενης
- Το Arduino Mega, χρησιμοποιώντας τεχνολογία surface-mounted ATmega1280 για περαιτέρω I/O και μνήμη
- Το Arduino Uno, χρησιμοποιώντας 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
- Το Arduino Mega2560, χρησιμοποιεί τεχνολογία surface-mounted ATmega2560 φέρνοντας την ολική μνήμη στα 256kB. Επίσης ενσωματώνει την νέα τεχνολογία ATmega8U2 (ATmega16U2 σε αναθεώρηση τύπου 3) USB chipset.
- Το Arduino Leonardo, με ένα ATmega32U4 chip που εξαλείφει την ανάγκη για συνδεσιμότητα μέσω USB και μπορεί να χρησιμοποιηθεί ως ψηφιακό πληκτρολόγιο ή ποντίκι. Κυκλοφόρησε στο Maker Faire Bay Area το 2012
- Το Arduino Esplora, με εμφάνιση που παραπέμπει σε χειριστήριο κονσόλας βιντεοπαιχνιδιών με joystick και ενσωματωμένους αισθητήρες για ήχο, φως, θερμοκρασία και επιτάχυνση
- Το Arduino Due είναι ένα μικροχειριστήριο board βασισμένο στην τεχνολογία Atmel SAM3X8E ARM Cortex-M3 CPU. Είναι το πρώτο board της Arduino βασισμένη σε επεξεργαστή 32-bit ARM microcontroller.

4.1.3 Λογισμικό

Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java, 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).



Εικόνα 4.1: Περιβάλλον IDE

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
}
```

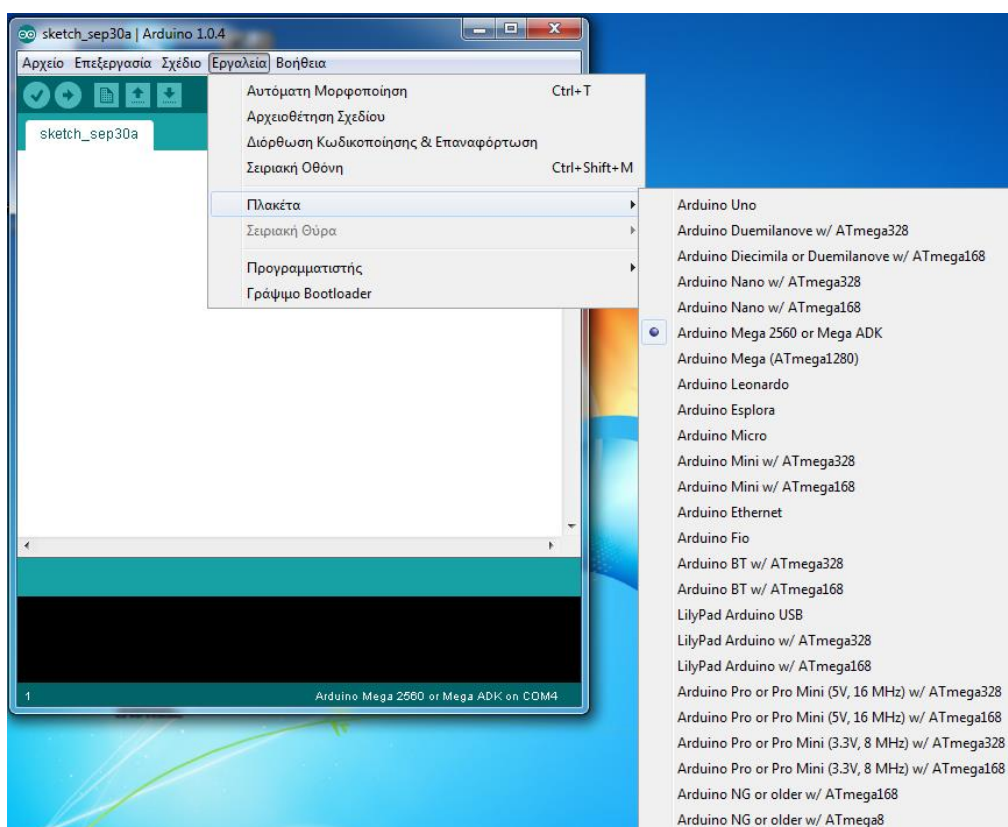
Είναι ένα χαρακτηριστικό των περισσότερων πλακετών Arduino ότι έχουν ένα LED και μία αντίσταση φορτίου που συνδέονται μεταξύ του pin 13 και του εδάφους, ένα βολικό χαρακτηριστικό για πολλά απλά τεστ. Ο προηγούμενος κώδικας δεν θα αναγνωριστεί από ένα κανονικό μεταγλωττιστή C++ ως έγκυρο πρόγραμμα, έτσι 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.

Διαδικασία φόρτωσης πηγαίου κώδικα σε πλακέτα Arduino

Αφού γράψουμε τον κώδικα C στην εφαρμογή, στη συνέχεια πηγαίνουμε στη Main Toolbar και χρησιμοποιούμε το κουμπί  έτσι ώστε το Arduino IDE να ελέγξει τον κώδικα για πιθανά σφάλματα.

Έπειτα, από το μενού **Εργαλεία** → **Πλακέτα**, επιλέγουμε τον τύπο πλακέτας που έχουμε. (βλ. *Εικόνα 2.2*)

Τέλος, συνδέουμε το Arduino με USB στον υπολογιστή και πατάμε το κουμπί  για να γίνει η μεταγλώττιση και η φόρτωση στην πλακέτα.



Εικόνα 4.2: Επιλογή από τη λίστα με τις διάφορες πλακέτες Arduino

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 το νεότερη έκδοση του Atmel Studio, μπορεί επίσης να χρησιμοποιηθεί για την ανάπτυξη λογισμικού για το Arduino.

4.1.4 Ανάπτυξη

Η κύρια ομάδα ανάπτυξης του Arduino αποτελείται από τους: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis και Nicholas Zambetti. Ο Massimo Banzi έδωσε συνέντευξη στις 21 Μαρτίου του 2009 στο επεισόδιο 61 του FLOSS Weekly στο κανάλι TWiT.tv, στο οποίο συζήτησε την ιστορία και τους στόχους του προγράμματος Arduino. Επίσης, έδωσε μια ομιλία στο TEDGlobal 2012 Conference, όπου περιέγραψε διάφορες χρήσεις των πλακετών Arduino σε όλο τον κόσμο.

Το Arduino είναι software ανοιχτού λογισμικού: τα σχέδια αναφοράς του software του Arduino διανέμονται υπό την Creative Commons Attribution Share-Alike 2.5 άδεια και είναι διαθέσιμα στην ιστοσελίδα του Arduino. Ο σχεδιασμός και η παραγωγή αρχείων για κάποιες εκδόσεις του software Arduino είναι επίσης διαθέσιμοι. Ο πηγαίος κώδικας για το IDE είναι διαθέσιμος και διανέμεται υπό την GNU General Public License, έκδοση 2. Παρά το γεγονός ότι το software και τα σχέδια του λογισμικού είναι διαθέσιμα ελεύθερα υπό άδειες πνευματικών δικαιωμάτων, οι προγραμματιστές έχουν ζητήσει η ονομασία "Arduino" είναι αποκλειστική για το επίσημο προϊόν και δεν πρέπει να χρησιμοποιείται για τις εργασίες χωρίς άδεια. Το επίσημο έγγραφο πολιτικής σχετικά με τη χρήση του ονόματος Arduino τονίζει ότι το πρόγραμμα είναι ανοιχτό στη συνεργασία με άλλους στο επίσημο προϊόν. Αρκετά προϊόντα συμβατά με Arduino που κυκλοφορούν στο εμπόριο έχουν αποφύγει το όνομα "Arduino" χρησιμοποιώντας την κατάληξη "-duino" με παραλλαγές στο όνομα.

Arduino Uno

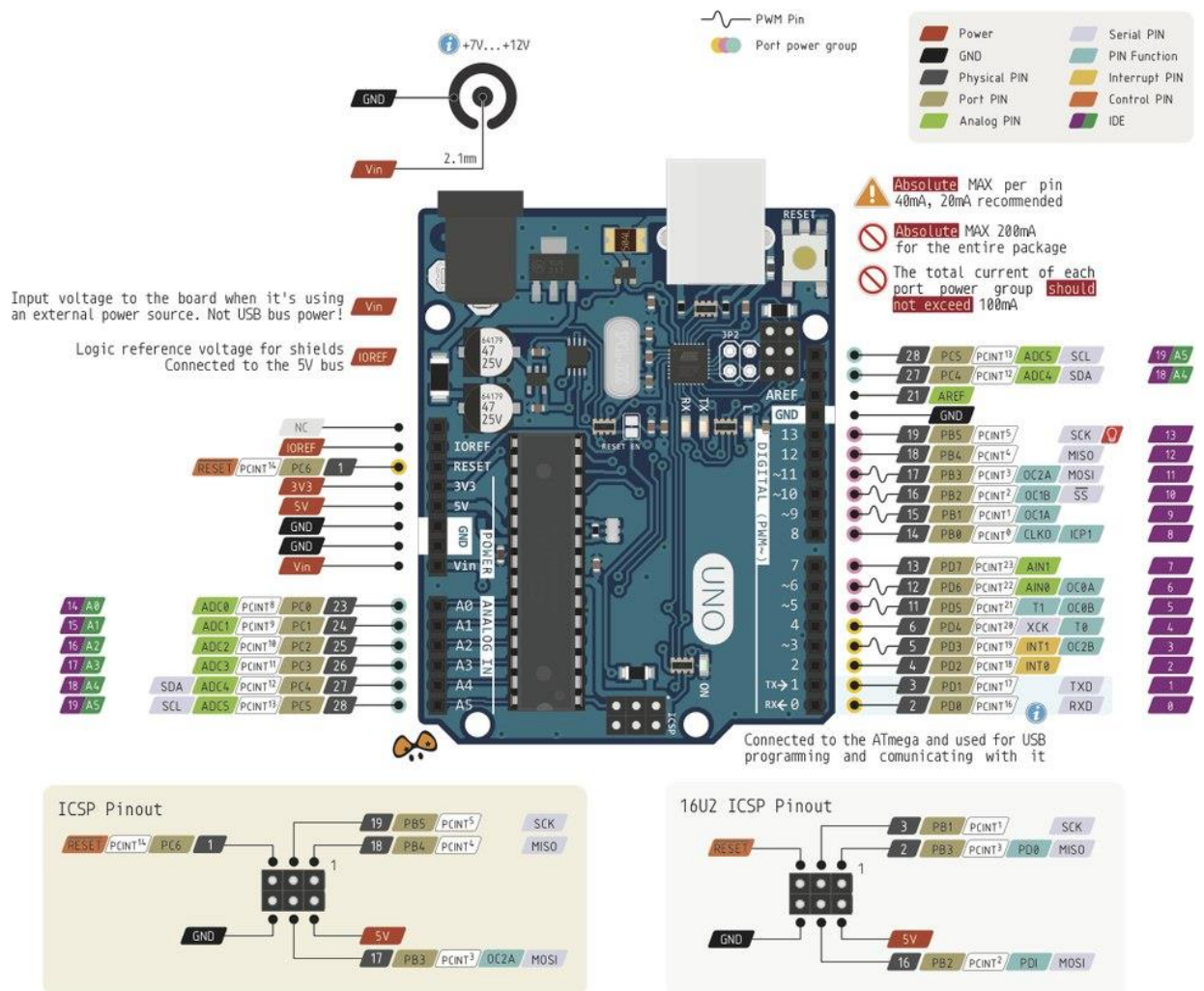
Είναι μία υπολογιστική πλατφόρμα βασισμένη στον μικροελεγκτή ATmega328P. Διαθέτει 14 ψηφιακές υποδοχές εισόδου/εξόδου (από τις οποίες 6 μπορούν να χρησιμοποιηθούν ως PWM έξοδοι), 6 αναλογικές εισόδους, 1 κρυσταλλικό ταλαντωτή 16 MHz, υποδοχή USB, υποδοχή τροφοδοσίας ρεύματος(power jack), 1 ICSP header και ένα κουμπί reset.



Εικόνα 4.3 : Arduino Uno

Τεχνικά χαρακτηριστικά

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g



Εικόνα 4.4 : Arduino Uno Schematic

4.1.5 Τροφοδοσία

Περιέχει ό,τι χρειάζεται για την λειτουργία του microcontroller, συνδέεται απλά με τον υπολογιστή μέσω USB ή εναλλακτικά μπορεί να τροφοδοτηθεί με μετασχηματιστή AC/DC ή μπαταρία 9V. Η πηγή ρεύματος επιλέγεται αυτόματα και το συνιστώμενο εύρος τάσης είναι 7-12V.

Οι ακροδέκτες τροφοδοσίας είναι:

Vin: Είναι η είσοδος τάσης της πλακέτας όταν χρησιμοποιείται μια εξωτερική πηγή ρεύματος (5V από σύνδεση με USB ή άλλη ρυθμιζόμενη πηγή ρεύματος). Όταν υπάρχει συνδεδεμένη εξωτερική τροφοδοσία μέσω του ακροδέκτη ρεύματος (power

jack), τότε το pin Vin μπορεί να χρησιμοποιηθεί για την τροφοδοσία διαφόρων εξαρτημάτων και συσκευών παρέχοντας απευθείας την τάση εξωτερικής τροφοδοσίας (7-12V).

GND: Τα Pins της γείωσης.

5V: Το Pin αυτό είναι έξοδος και παρέχει τάση 5V. Η τάση αυτή προέρχεται είτε από τη θύρα USB, είτε από την εξωτερική τροφοδοσία που μέσω ενός ρυθμιστή τάσης την 'κατεβάζει' στα 5V.

3.3V: Αυτό το Pin έχει τάση 3.3V μέσω του ρυθμιστή τάσης της πλακέτας και η μέγιστη ένταση ρεύματός του είναι 50mA.

RESET: Όταν αυτός ο ακροδέκτης γειωθεί με κάποια από τα Pins GND, γίνεται επανεκκίνηση του Arduino.

IOREF: Το Pin IOREF αναφέρεται στην τάση την οποία διαχειρίζεται ο μικροελεγκτής της πλακέτας. Μία κατάλληλα διαμορφωμένη shield μπορεί να διαβάσει αυτό το Pin τάσης και να διαλέξει την κατάλληλη πηγή ρεύματος.

4.1.6 Μνήμη

Το ATmega328P έχει μνήμη 32 KB για να αποθηκεύει τον κώδικα, 2 KB για την SRAM και 1 KB για την EEPROM(η οποία μπορεί να διαβαστεί και να γραφτεί με τη βιβλιοθήκη EEPROM).

4.1.7 Είσοδοι - Έξοδοι

Καθένας από τους 14 ψηφιακούς ακροδέκτες του μπορεί να χρησιμοποιηθεί σαν είσοδος και έξοδος, με τη βοήθεια των συναρτήσεων: [pinMode\(\)](#), [digitalWrite\(\)](#), και [digitalRead\(\)](#). Ενεργοποιούνται στα 5 volts. Κάθε pin μπορεί να παρέχει ή να λαμβάνει μέχρι 20 mA και επίσης έχει μια εσωτερική αντίσταση (αποσυνδεδεμένη από προεπιλογή) των 20-50 kOhms. Επιπλέον, μερικά pins έχουν εξειδικευμένες συναρτήσεις:

Serial: 0 (RX) και 1 (TX); Serial 1: 19 (RX) και 18 (TX); Serial 2: 17 (RX) και 16 (TX); Serial 3: 15 (RX) και 14 (TX). Χρησιμοποιούνται για να λάβει (RX) και να μεταδώσει (TX) TTL σειριακά δεδομένα. Τα pins 0 and 1 συνδέονται επίσης με τα αντίστοιχα pins του ATmega8U2 USB-to-TTL Serial chip.

External Interrupts: 2 (interrupt 0), 3 (interrupt 1). Αυτοί οι ακροδέκτες μπορούν να ρυθμιστούν ώστε να προκαλέσουν μία διακοπή σε μία χαμηλή τιμή, αύξηση, πτώση ή ακόμα και αλλαγή στη τιμή. Χρησιμοποιείται η συνάρτηση [attachInterrupt\(\)](#).

PWM: 3, 5, 6, 9, 10 και 11. Παρέχει 8-bit PWM εξόδους με συνάρτηση την [analogWrite\(\)](#).

SPI: 10 (SS), 11 (MOSI), 12 (MISO) 13 (SCK). Υποστηρίζουν SPI επικοινωνία χρησιμοποιώντας την SPI library.

LED: 13. Στο ψηφιακό pin 13 είναι συνδεδεμένο ένα LED. Όταν το pin είναι HIGH, το LED είναι αναμμένο, ενώ όταν το pin είναι LOW, είναι σβηστό.

TWI: A4 (SDA) και A5 (SCL). Υποστηρίζουν TWI επικοινωνία χρησιμοποιώντας τη Wire Library.

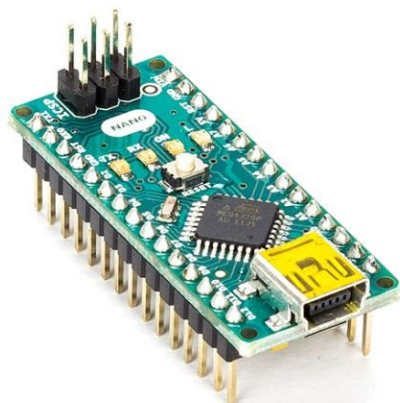
Το Uno έχει 6 αναλογικές εισόδους, εκ των οποίων η κάθε μία παρέχει 10 bits of resolution (π.χ. 1024 διαφορετικές τιμές). Από προεπιλογή, μετρούν από ground έως 5V, ωστόσο μπορεί να αλλάξει το ανώτατο όριό τους χρησιμοποιώντας το pin AREF και τη συνάρτηση [analogReference\(\)](#).

Υπάρχουν δύο άλλα pins στη πλακέτα:

AREF: Reference voltage για τις αναλογικές εισόδους. Χρησιμοποιείται με τη συνάρτηση [analogReference\(\)](#).

RESET: Πατώντας το γίνεται reset στον μικροελεγκτή. Στην ουσία χρησιμοποιείται για να προσθέσει ένα κουμπί reset στις shields το οποίο συνδέεται με την πλακέτα.

Arduino Nano



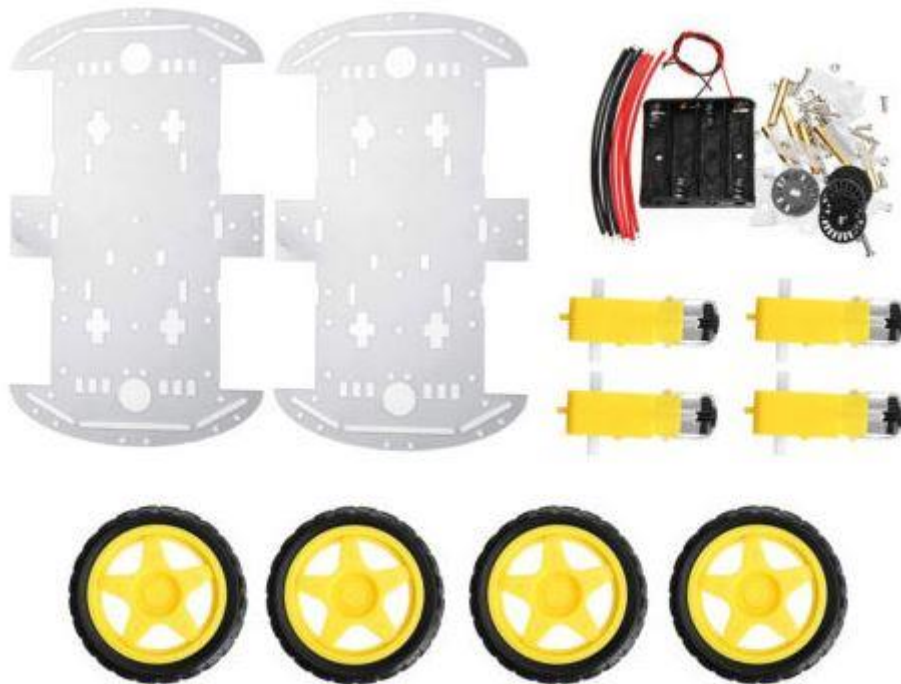
Τεχνικά χαρακτηριστικά

Microcontroller	ATmega328
Architecture	AVR
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Digital I/O Pins	22 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	32 KB of which 2 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz
Power Consumption	19 mA
PCB Size	18 x 45 mm
Weight	7 g

4.2 Project Car



Εικόνα 4.5 : Project Car 1



Εικόνα 4.6 : Project Car 2

4.2.1 Κινητήρας DC

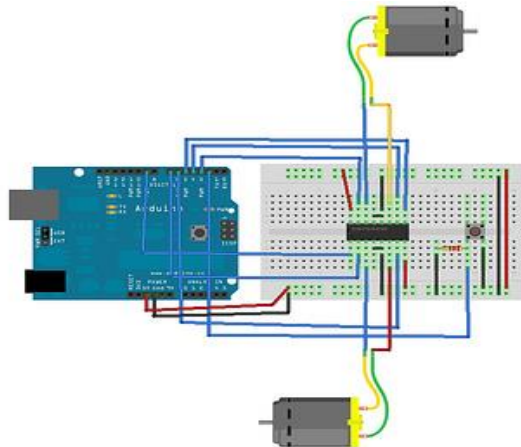
Οι κίνηση των τροχών πραγματοποιείται με τη βοήθεια 4 κινητήρων DC με τάση λειτουργίας 6V.



Χαρακτηριστικά:

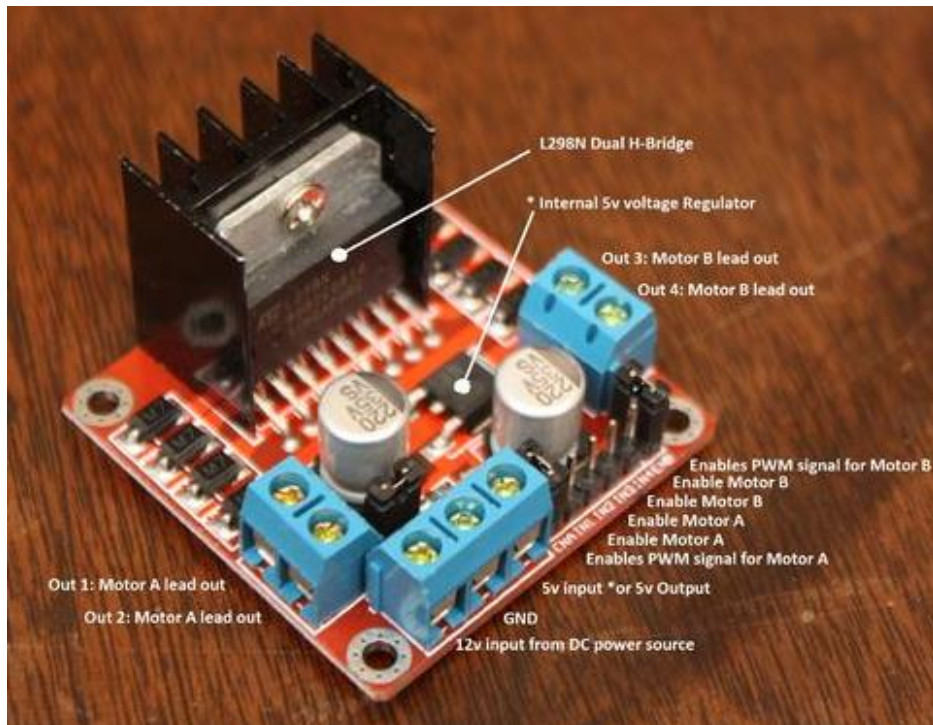
- Motor Voltage: 4.5-6 VDC
- No Load Speed: 90 ± 10 rpm
- No Load Current: 190mA(max.250mA)
- Torque: 800gf.cm

Εικόνα 4.7: Κινητήρας DC και τροχός



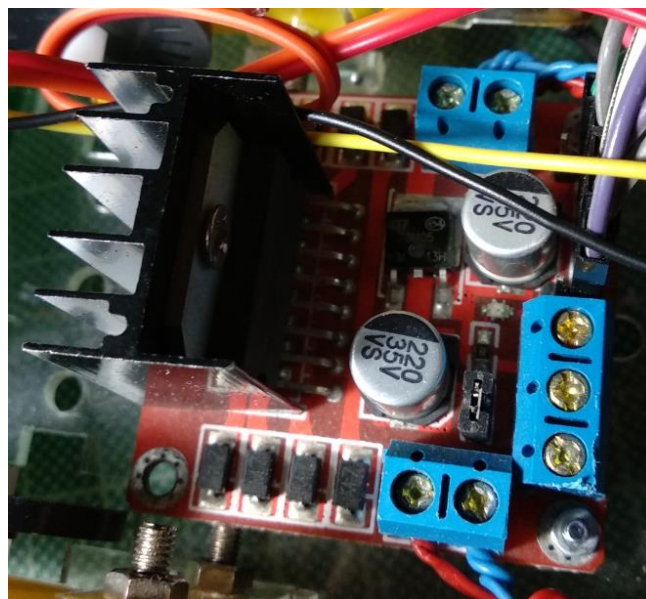
Εικόνα 4.8: Σύνδεσμολογία Arduino με κινητήρες DC με τη βοήθεια κατάλληλου μικροελεγκτή H-Bridge.

4.2.2 H-Bridge



Εικόνα 4.9: Motor Drive

H-Bridge: Χρησιμοποιείται για τον χειρισμό της ταχύτητας και της κατεύθυνσης των κινητήρων. Είναι ένα κύκλωμα που μπορεί να ελέγχει ένα ρεύμα σε κάθε πολικότητα και μπορεί να ελέγχεται από Ρύθμιση Πλάτους Παλμού (PWM).



Εικόνα 4.10: Motor Drive on Car

Pins

Out 1: Motor A lead out

Out 2: Motor A lead out

Out 3: Motor B lead out

Out 4: Mo (Can actually be from 5v-35v, just marked as 12v)

GND: Ground

5v: 5v input (unnecessary if your power source is 7v-35v, if the power source is 7v-35v then it can act as a 5v out)

EnA: Enables PWM signal for Motor A (Please see the "Arduino Sketch Considerations" section)

In1: Enable Motor A

In2: Enable Motor A

In3: Enable Motor B

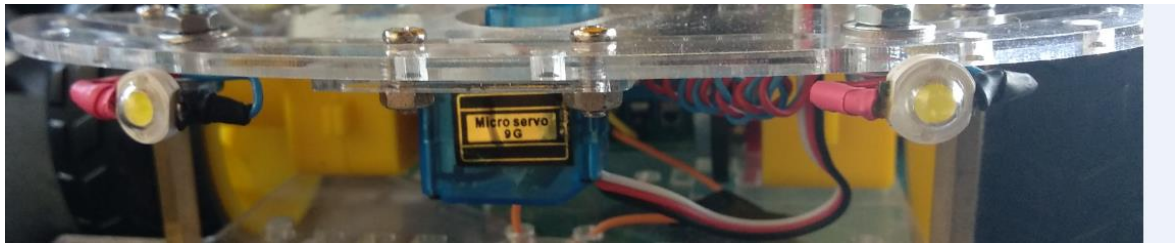
In4: Enable Motor B

EnB: Enables PWM signal for Motor B (Please see the "Arduino Sketch Considerations" section)

Specifications

- Double H bridge Drive Chip: L298N
- Logical voltage: 5V Drive voltage: 5V-35V
- Logical current: 0-36mA Drive current: 2A (MAX single bridge)
- Max power: 25W
- Dimensions: 43 x 43 x 26mm
- Weight: 26g

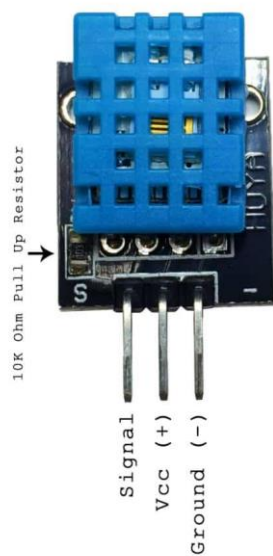
4.2.3 Lights



1-Watt White LED
6000-6500K
3.2-3.6 V
350mA

Εικόνα 4.11: LED Lights

4.2.4 DHT11 Humidity Sensor



DHT11

0 - 50°C / ± 2°C

20 - 80% / ± 5%

1Hz (one reading every second)

15.5mm x 12mm x 5.5mm

3 - 5V

2.5mA

Εικόνα 4.12: DHT11 sensor

Σύνδεση Αισθητήρα Θερμοκρασίας και Υγρασίας για ενδείξεις περιβάλλοντος στο οποίο κινείται το όχημα.

4.2.5 FPV Camera



600TVL M7 1/4" Cmos FPV Camera

Lens Angle	120° FOV 150° DFOV
Format	NTSC
Resolution	720x480
Focal length	F=2.0mm f=0.95mm
Power consumption	500mA
Power	3.3-5V

5.8Ghz 40CH 200mW FPV Transmitter

Format	NTSC
Output impedance	50Ω
Output signal strength	13±1 dbm dBm
Output Power	200mW
Wireless Channel	40CH
Operating voltage	3.3-5V
Current	480mA
Audio carrier	None



Εικόνα 4.23: Specs & FPV camera

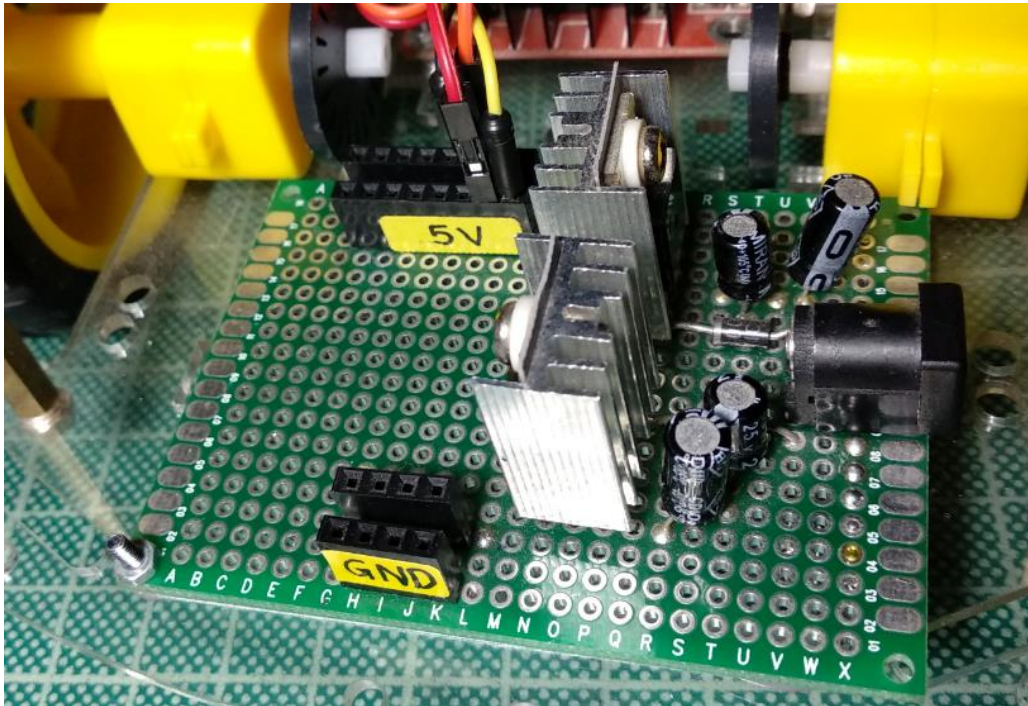
Η κάμερα βρίσκεται πάνω στη βάση που ελέγχει ο κινητήρας (servo). Ο χειριστής έχει τη δυνατότητα, μέσω του Joystick, να περιστρέφει την κάμερα(0-180°) για την καλύτερη εποπτεία του χώρου.

4.2.6 Power Board

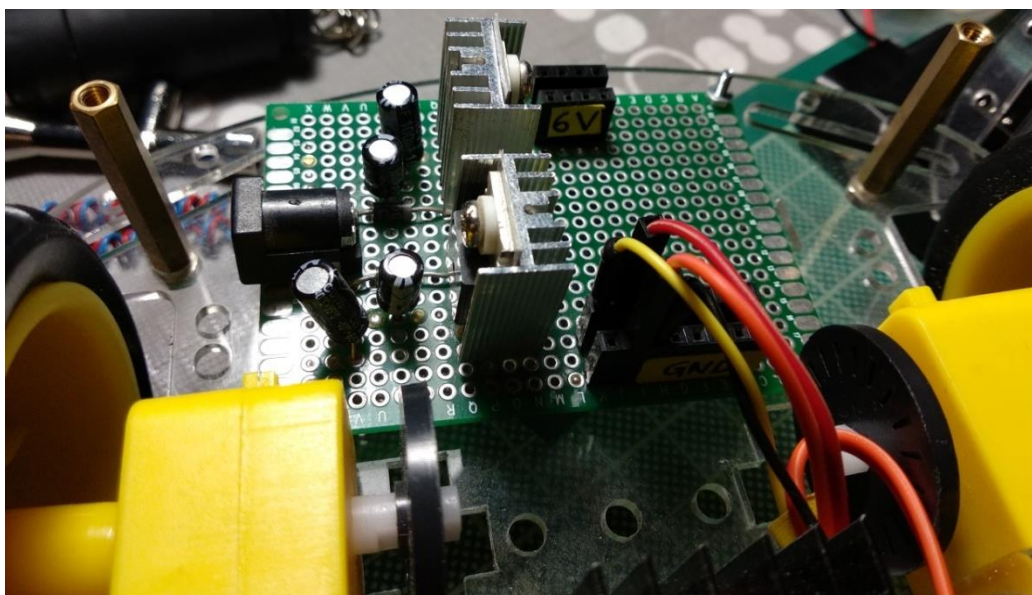
Χρήση voltage regulator για πτώση τάσης στα 5 και 6 Volt(Τάση εισόδου 7-12V).

Voltage Regulator for 5V: **LM7805C**

Voltage Regulator for 6V: **LM7806CV**



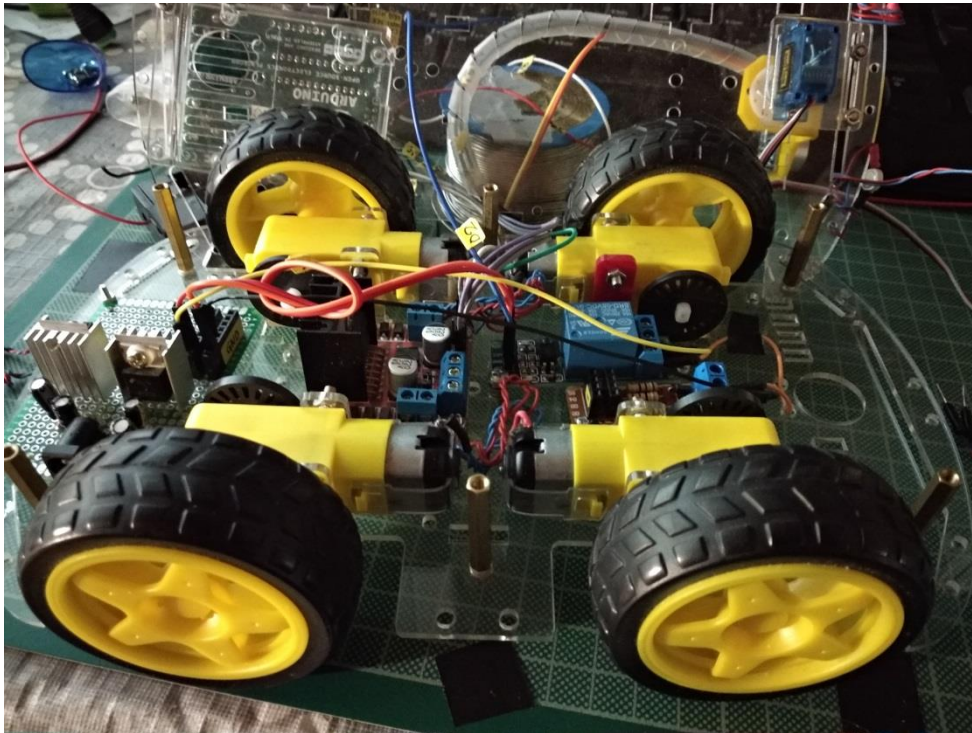
Εικόνα 4.13: Power board view 1



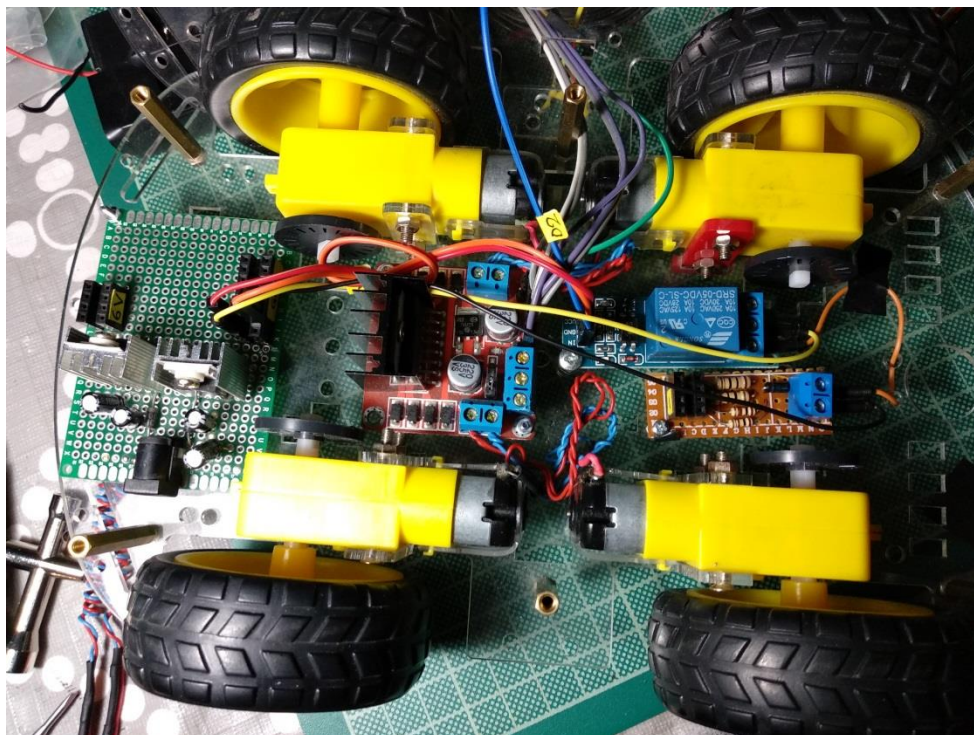
Εικόνα 4.14: Power board view 2

4.2.7 Building Project Car

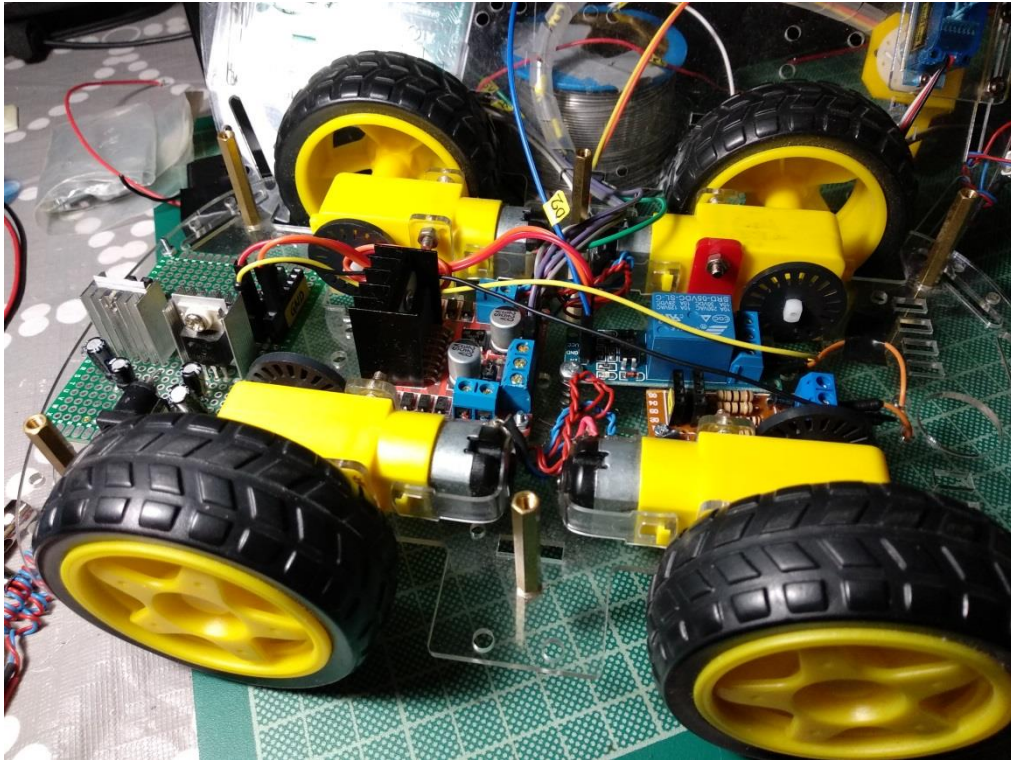
Στο 1^ο επίπεδο έχουμε: Motor Drive, PCB Power board, Relay Module και PCB board για τα φώτα.



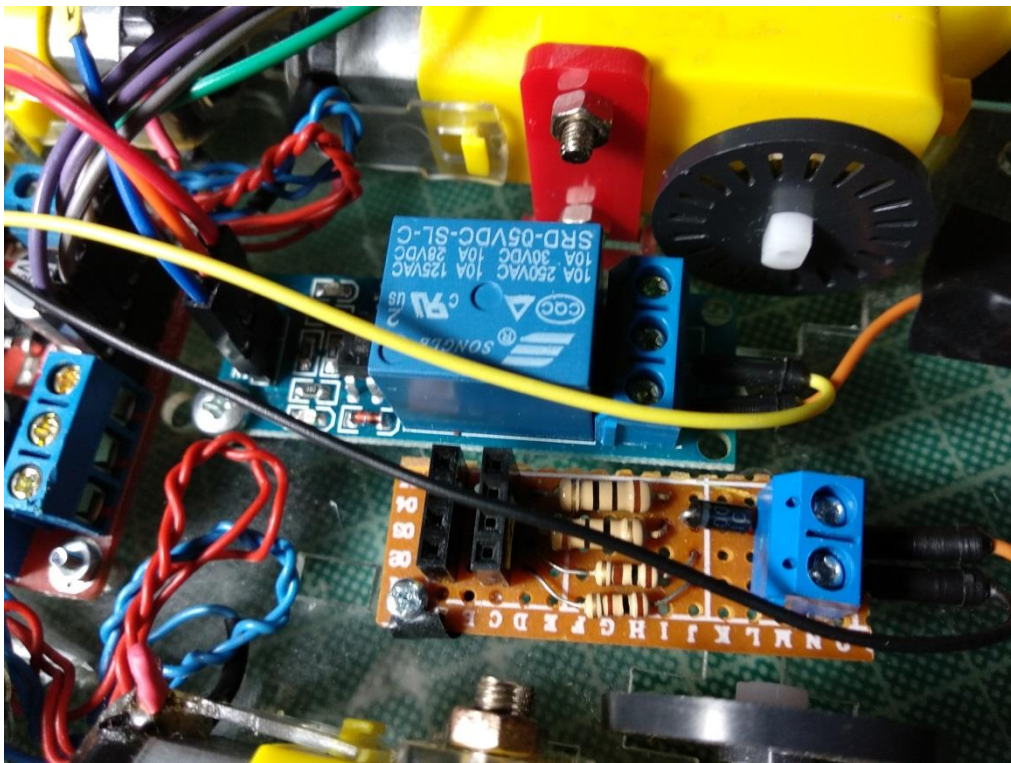
Εικόνα 4.15: 1st Level connections



Εικόνα 4.16: 1st Level connections 2



Εικόνα 4.17: 1st Level connections³

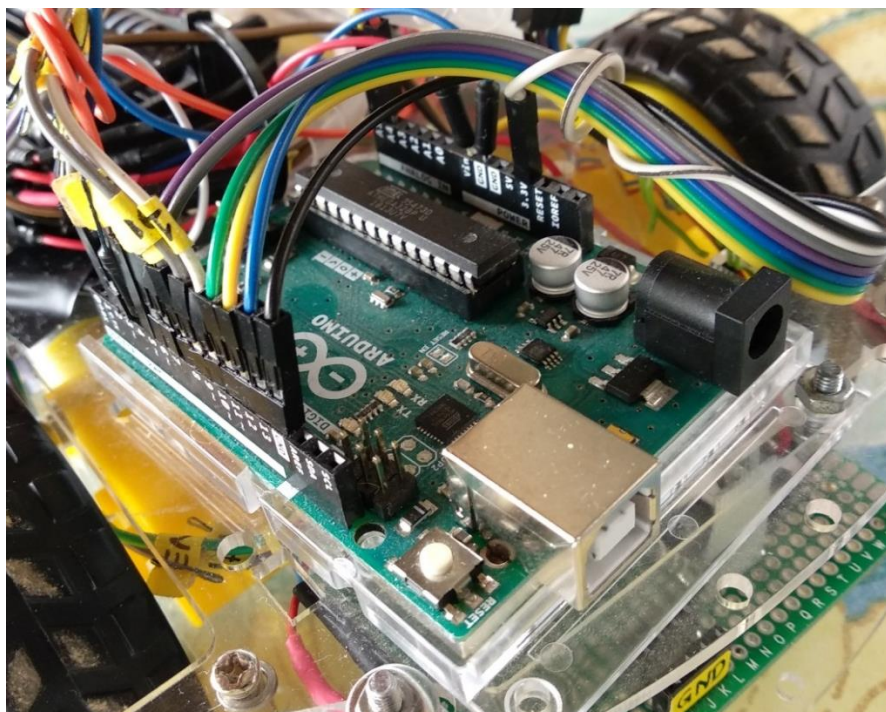


Εικόνα 4.18: Relay & PCB Lights

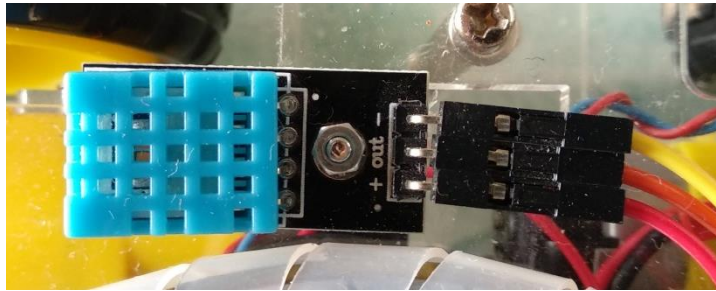
Στο 2^ο επίπεδο έχουμε: Power Supply(2x18650: 7.4Volt), 3.7V Lithium Battery, Arduino Uno, DHT11 Humidity Sensor, Servo Motor, NRF24L01, FPV Camera.



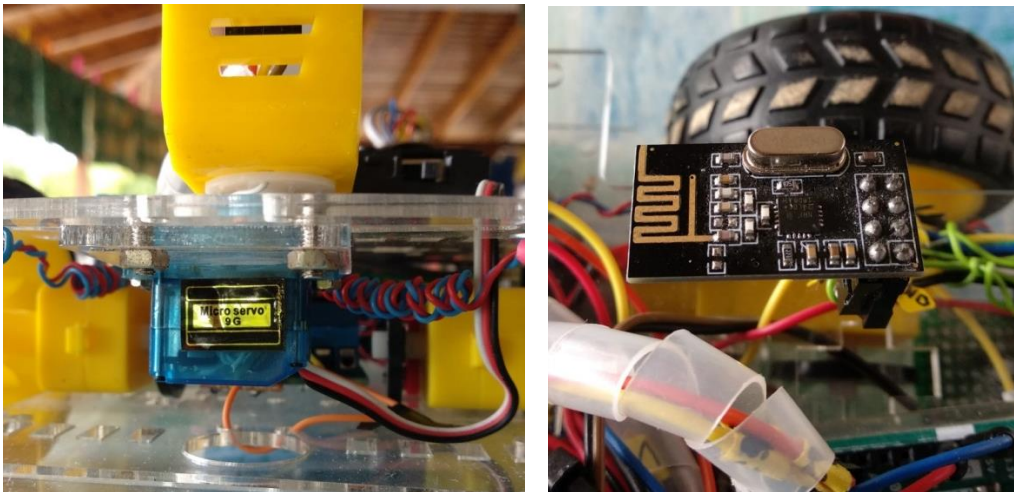
Εικόνα 4.19: Power supply



Εικόνα 4.20: Arduino Uno view



Εικόνα 4.21: DHT11 Humidity Sensor

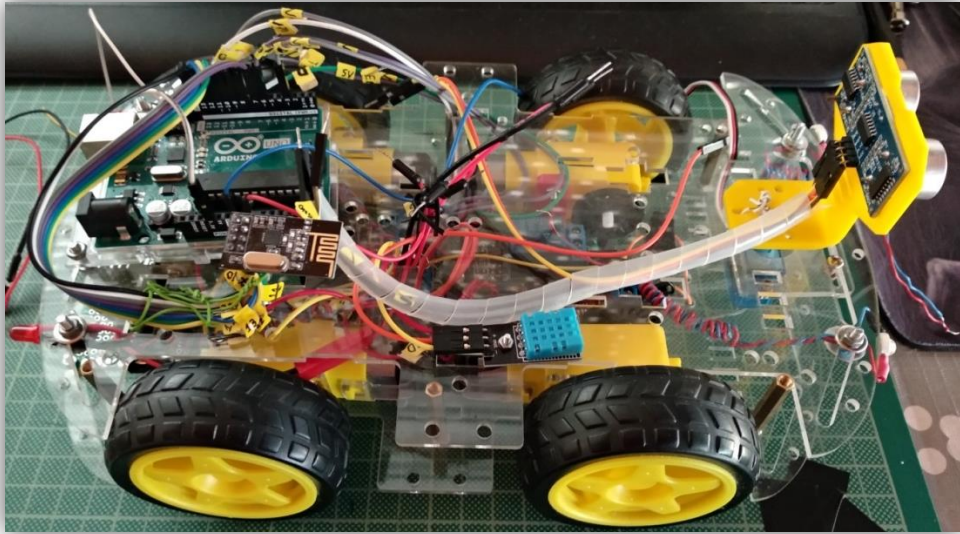


Εικόνα 4.22: Servo Motor & NRF24L01Module

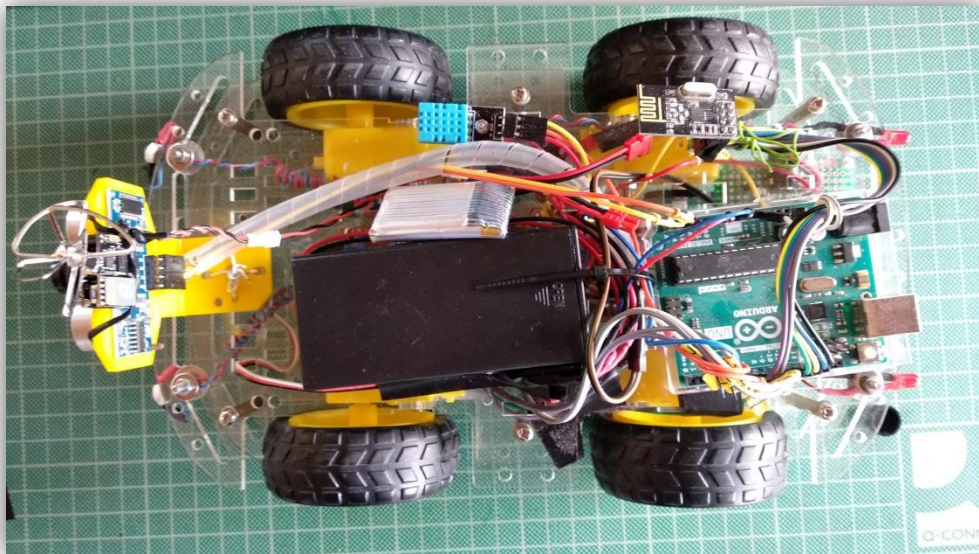
Ο αισθητήρας θερμοκρασίας, υγρασίας τοποθετήθηκε στη μέση του οχήματος, στη δεξιά μεριά.

Ο κινητήρας (servo) βρίσκεται στο μπροστινό μέρος και ελέγχει τη βάση πάνω στην οποία βρίσκεται η κάμερα.

Η ασύρματη επικοινωνία (NRF24L01) τοποθετήθηκε δίπλα στην πλακέτα Arduino με τέτοιο τρόπο ώστε να διευκολύνεται η μετάδοση του σήματος.



Εικόνα 4.24: Car in progress



Εικόνα 4.25: Almost Ready

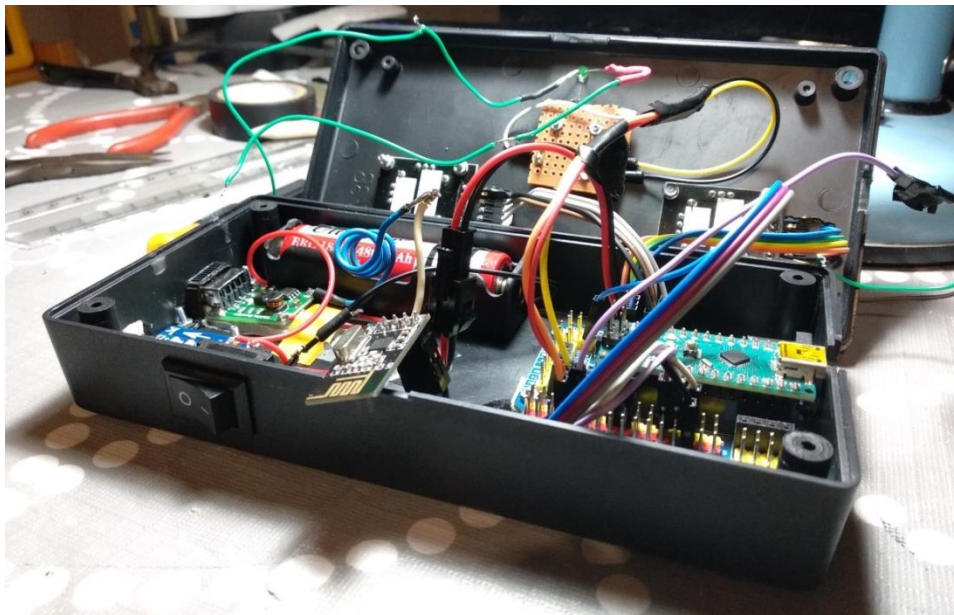
Έχουν τοποθετηθεί όλα τα υλικά και το όχημα είναι σχεδόν έτοιμο. Παρακάτω ακολουθεί η διαδικασία συναρμολόγησης του χειριστήριου.

4.2.8 Controller

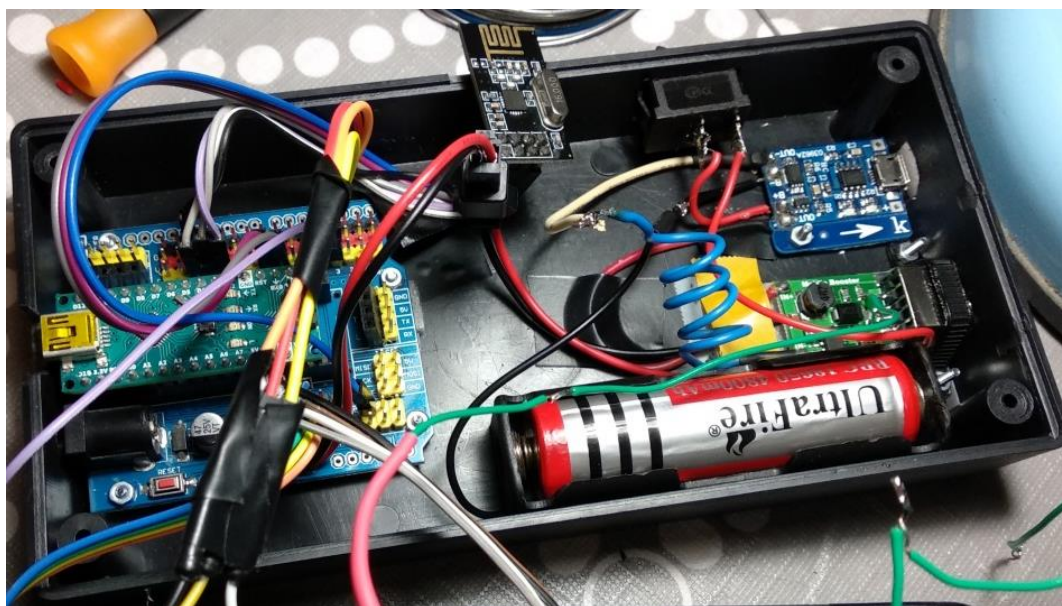
18650 Battery(3.7V), Charging PCB for 18650, step-up board(3.7 to 5V), Arduino Nano with Sensor shield, 2x Joystick, Power led, Function led, Function button, NRF24L01, ON-OFF switch.



Εικόνα 4.26: Controller front view



Εικόνα 4.27: Building Controller

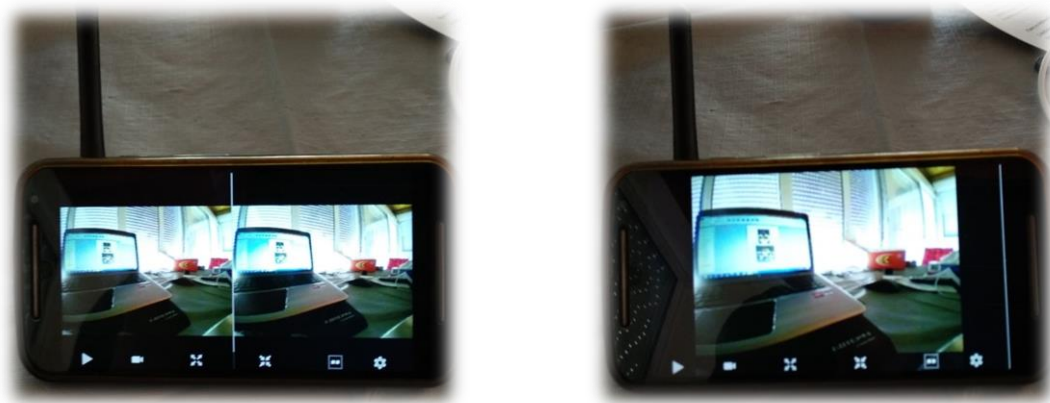


Εικόνα 4.28: Building Controller 2



Εικόνα 4.29: FPV Receiver

Συνδέουμε τον δέκτη της κάμερας με ένα κινητό, το οποίο υποστηρίζει UVC MPEG και συνδεόμαστε στην κατάλληλη εφαρμογή. Η εφαρμογή μας δίνει εικόνα και δυνατότητα εγγραφής βίντεο. Η εικόνα μπορεί να εμφανίζεται και σε διπλό παράθυρο κι έτσι έχουμε τη δυνατότητα χρήσης VR Glasses.



Εικόνα 4.30: Camera – single and double screen



Εικόνα 4.31: VR Glasses & FPV Receiver



Εικόνα 4.32: Controller & FPV Receiver with mobile

4.3 Code

Στην κατασκευή υπάρχουν δύο κώδικες, ένας για κάθε πλακέτα (2x Arduino Board). Μέσω των NRF24L01, με συγκεκριμένες αρχικοποιήσεις στην αρχή του κάθε κώδικα και μέσα στις συναρτήσεις `void setup()`, καθώς και με τη βοήθεια της δομής `struct myData`, επιτυγχάνεται η αμφίδρομη επικοινωνία μεταξύ των δύο microcontroller.

Ο κώδικας που χρησιμοποιείτε στο Arduino UNO και στο Arduino Nano αναφέρεται παρακάτω.

4.3.1 Arduino UNO - Car

Χρησιμοποιούνται οι εξής βιβλιοθήκες:

```
#include <SPI.h>
#include "nRF24L01.h"
#include <RF24.h>
#include <Servo.h>
#include <dht.h>
```

Δήλωση ενός `unsigned integer` για να επιτύχουμε την αμφίδρομη επικοινωνία των δύο NRF24L01 Boards και δήλωση `digital pin` για CE & CSN.

```
const uint64_t addresses[] = {0xF0F0F0F0E1LL, 0xF0F0F0F0D2LL};
RF24 radio(7, 8); // CE, CSN
```

Δημιουργία αντικειμένου και βοηθητικής μεταβλητής καθώς και ορισμός του `digital pin 5` για τη χρήση αισθητηρίου Θερμοκρασίας και Υγρασίας (DHT11).

```
dht DHT;
#define DHT11_PIN 5
int temp_hum_values;
```

Δηλώσεις μεταβλητών για τους κινητήρες και τα φώτα και δημιουργία αντικειμένου για κινητήρα Servo.

```
const int motorA10= 3;
const int motorA20= 6;
const int motorB10= 10;
const int motorB20= 9;

Servo myServo;

int lights = 2;
int status_lights = 0;
int lost_signal_count = 0;
```

Για την μεταφορά δεδομένων μεταξύ των NRF24 χρησιμοποιείτε μια δομή, η οποία περιέχει όλες τις μεταβλητές που χρειαζόμαστε για να επικοινωνεί το όχημα με το χειριστήριο. Οι μεταβλητές αυτές ανανεώνονται συνεχώς και στέλνονται από το χειριστήριο στο όχημα και αντίστροφα (αμφίδρομη επικοινωνία). Στη συνέχεια, οι μεταβλητές χρησιμοποιούνται για τις διάφορες λειτουργίες.

```
struct dataStruct{
  int JoylposX;
  int JoylposY;
  int motorA1;//PISW DEKSIA
  int motorA2;//MPROSTA DEKSIA
  int motorB1;//MPROSTA ARISTERA
  int motorB2;//PISW ARISTERA
  int auto_mode_enabler;
  int sensor;
  float humidity;
  float temperature;
} myData;
```

Συνάρτηση `void setup()`

Σε αυτή την συνάρτηση γίνονται όλες οι αρχικοποιήσεις. Καθορίζονται τα Pin τα οποία είναι είσοδοι/έξοδοι του συστήματος (Digital I/O). Επίσης, ανοίγει ο δίαυλος επικοινωνίας μεταξύ των NRF24 για την αμφίδρομη επικοινωνία τους σε συγκεκριμένο κανάλι Wi-Fi και με σταθερό ρυθμό μεταφοράς δεδομένων.

```
void setup() {

  Serial.begin(9600);
  pinMode(lights, OUTPUT);

  pinMode(motorA1,OUTPUT);
  pinMode(motorA2,OUTPUT);
  pinMode(motorB1,OUTPUT);
  pinMode(motorB2,OUTPUT);

  myServo.attach(4);

  radio.begin();
  radio.setChannel(108); // Above most WiFi frequencies
  radio.setDataRate(RF24_250KBPS); // Fast enough.. Better range
  radio.openWritingPipe(addresses[0]); // 00002
  radio.openReadingPipe(1, addresses[1]); // 00001

  //radio.setPALevel(RF24_PA_LOW);
  //radio.printDetails();

}
```

Συνάρτηση `void loop()`

Εδώ βρίσκεται το κύριο κομμάτι του κώδικα το οποίο ' τρέχει ' συνεχώς (συνάρτηση επανάληψης). Το όχημα συνεχώς διαβάζει και στέλνει δεδομένα από και προς το χειριστήριο. Ανάλογα με τις συνθήκες που έχουν οριστεί, πραγματοποιούνται οι εκάστοτε λειτουργίες (π.χ. εκκίνηση κινητήρων, έλεγχος κινητήρα servo για την κάμερα, έλεγχος για τα φώτα – ON/OFF).

Το NRF24 πάνω στο όχημα ξεκινάει να διαβάζει δεδομένα από το Controller.

```
radio.startListening();
if ( radio.available()) {

    lost_signal_count = 0;

while (radio.available()) {
    radio.read( &myData, sizeof(myData) );
}

//KINISH SERVO-AISTHITHRA

myServo.write(myData.JoylposX);

//KINHSH AYTOKINHTOY

analogWrite(motorA10, myData.motorA1);
analogWrite(motorA20, myData.motorA2);
analogWrite(motorB10, myData.motorB1);
analogWrite(motorB20, myData.motorB2);
```

Ελέγχοντας τις τιμές στον άξονα ψ του Joystick_1 είτε ανάβω τα φώτα (με κίνηση του Joystick προς τα πάνω), είτε στέλνω μέτρηση θερμοκρασίας και υγρασίας στο Controller (με κίνηση του Joystick προς τα κάτω).

```
if (myData.JoylposY < 300) //Open Lights
{
    if (status_lights == 0){
        digitalWrite(lights, HIGH);
        status_lights = 1;
    }
    else{
        digitalWrite(lights, LOW);
        status_lights = 0;
    }
}
else if (myData.JoylposY > 720) //Send Temperature-Humidity
{
    temp_hum_values = DHT.read11(DHT11_PIN);
    myData.sensor=1;
    myData.humidity=DHT.humidity;
    myData.temperature=DHT.temperature;
```

```

    delay(5);
    radio.stopListening();
    radio.write( &myData, sizeof(myData) );
    Serial.print("Current humidity = ");
    Serial.print(DHT.humidity);
    Serial.print("% ");
    Serial.print("temperature = ");
    Serial.print(DHT.temperature);
    Serial.println("C ");
    radio.startListening();
}
else
{
    myData.sensor=0;
    delay(5);
    radio.stopListening();
    radio.write( &myData, sizeof(myData) );
    radio.startListening();
}

} // TELOS RADIO AVAILABLE

```

Εάν χαθεί η επικοινωνία σταματάω τους κινητήρες του αυτοκινήτου.

```

else {

    lost_signal_count++;
    //An xasw sima 5 fores - Stop Car
    if (lost_signal_count == 5){
        analogWrite(motorA10, 0);
        analogWrite(motorA20, 0);
        analogWrite(motorB10, 0);
        analogWrite(motorB20, 0);
    }

}

} //TELOS VOID LOOP

```

4.3.2 Arduino Nano - Controller

Χρησιμοποιούνται οι εξής βιβλιοθήκες:

```

#include <SPI.h>
#include "nRF24L01.h"
#include <RF24.h>

```

Οι δηλώσεις για το NRF24L01 είναι ίδιες.

```

RF24 radio(7, 8); // CE, CSN
const uint64_t addresses[] = {0xF0F0F0F0E1LL, 0xF0F0F0F0D2LL};

```

Δήλωση τοπικών μεταβλητών για αισθητήρα θερμοκρασίας, υγρασίας (DHT11).

```
int sensor1 = 0;
float humidity1;
float temperature1;
```

Δηλώσεις μεταβλητών για κίνηση κινητήρων.

```
int potValue =0;
int angleValue =0;
int potValue1 =0;
int angleValue1 =0;
int potValue3 =0;
int angleValue3 =0;
int angleValue3L =0;
int potValue4 =0;
int angleValue4 =0;
int angleValue4L =0;
```

Η συνάρτηση `dataStruct()` είναι ακριβώς η ίδια και στο χειριστήριο για να συγχρονίζονται τα δεδομένα.

```
struct dataStruct{
  int Joy1posX;
  int Joy1posY;
  int motorA1;//PISW DEKSIA
  int motorA2;//MPROSTA DEKSIA
  int motorB1;//MPROSTA ARISTERA
  int motorB2;//PISW ARISTERA
  int auto_mode_enabler;
  int sensor;
  float humidity;
  float temperature;
} myData;
```

Συνάρτηση `void setup()`

Σε αυτή την συνάρτηση γίνονται όλες οι αρχικοποιήσεις. Καθορίζονται τα Pin τα οποία είναι είσοδοι/έξοδοι του συστήματος (Digital I/O). Επίσης, ανοίγει ο διάλογος επικοινωνίας μεταξύ των NRF24 για την αμφίδρομη επικοινωνία τους σε συγκεκριμένο κανάλι Wi-Fi και με σταθερό ρυθμό μεταφοράς δεδομένων.

```
void setup() {
  Serial.begin(9600);
  radio.begin();
  radio.setDataRate(RF24_250KBPS);
  radio.setChannel(108);
  radio.openWritingPipe(addresses[1]); // 00001
  radio.openReadingPipe(1, addresses[0]); // 00002
  //radio.setPALevel(RF24_PA_LOW);
}
```

Συνάρτηση `void loop()`

Εδώ βρίσκεται το κύριο κομμάτι του κώδικα το οποίο 'τρέχει' συνεχώς (συνάρτηση επανάληψης). Το χειριστήριο συνεχώς διαβάζει και στέλνει δεδομένα από και προς το όχημα. Υπάρχουν δύο Joystick, ένα για την κίνηση του οχήματος και ένα για τα φώτα, τον αισθητήρα (DHT11) και το servo, για τον έλεγχο της κάμερας (0 – 180°). Διαβάζει λοιπόν τις τιμές των αξόνων x, y των Joystick.

```
void loop() {  
  
    delay(5);  
  
    // LEITOURGIA APOSTOLHS ENTOLWN STO AYTOKINITO  
    radio.stopListening();  
  
    //----- Joystick 1 - DEKSI -----//  
  
    // X axonas tou Joystick 1 --> SERVO  
    potValue = analogRead(A0);  
    angleValue = map(potValue, 0, 1023, 20, 160);  
    myData.Joy1posX = angleValue;  
  
    // Y axonas tou Joystick 1 --> KATW Temperature , EPANW Lighs  
    potValue1 = analogRead(A1);  
    myData.Joy1posY = potValue1;  
  
    //--- Joystick 2 - ARISTERO -----//  
  
    // X axonas tou Joystick 2 --> KINSH DEKSIA - ARISTERA  
    potValue3 = analogRead(A3);  
  
    // Y axonas tou Joystick 2 --> KINSH MPROSTA - PISW  
    potValue4 = analogRead(A4);
```

Το παρακάτω κομμάτι του κώδικα χρησιμοποιείται για την κίνηση του οχήματος. Λαμβάνοντας τις τιμές από τους άξονες x, y του Joystick 2, με τις παρακάτω συνθήκες και εντολές, καλύπτουμε όλες τις περιπτώσεις κίνησης του οχήματος (κίνηση προς όλες τις κατευθύνσεις και με 'αναλογική' ταχύτητα - PWM). Η συνθήκες για κάθε κίνηση (αριστερά, δεξιά, μπροστά, πίσω) έχουν τις δικές τους υποπεριπτώσεις (Στα σχόλια του κώδικα φαίνονται λεπτομερώς οι περιπτώσεις της κίνησης). Έτσι, η κίνηση γίνεται ομαλή σε κάθε αλλαγή κατεύθυνσης.

```
// METATROPH X kai Y tou JOY 2 se KINSH AYTOKINHTOY  
  
if (potValue3<=450) //KINSH ARISTERA  
{  
    if (potValue4<=450) //KINSH ARISTERA kai ligo MPROSTA  
    {  
        angleValue3 = map(potValue3, 0, 450, 255, 0);
```



```

    angleValue3L = angleValue3/2;
    angleValue4 = 0;
//    Serial.print("ARISTERA - ligo MPROSTA: ");
myData.motorA1=angleValue4;
myData.motorA2=angleValue3;
myData.motorB1=angleValue3L;
myData.motorB2=angleValue4;
}
else if (potValue4>=570) //KINHSH ARISTERA kai ligo PISW
{
    angleValue3 = map(potValue3, 0, 450, 255, 0);
    angleValue3L = angleValue3/2;
    angleValue4 = 0;
//    Serial.print("ARISTERA - ligo PISW: ");
myData.motorA1=angleValue3;
myData.motorA2=angleValue4;
myData.motorB1=angleValue4;
myData.motorB2=angleValue3L;
}
else //KINHSH ARISTERA FULL
{
    angleValue3 = map(potValue3, 0, 450, 255, 0);
    angleValue4 = 0;
//    Serial.print("ARISTERA FULL: ");
myData.motorA1=angleValue4;
myData.motorA2=angleValue3;
myData.motorB1=angleValue4;
myData.motorB2=angleValue3;
}
}

if (potValue3>=570) //KINHSH DEKSIA
{
    if (potValue4<=450) //KINHSH DEKSIA kai ligo MPROSTA
    {
        angleValue3 = map(potValue3, 570, 1023, 0, 255);
        angleValue3L = angleValue3/2;
        angleValue4 = 0;
//        Serial.print("DEKSIA - ligo MPROSTA: ");
myData.motorA1=angleValue4;
myData.motorA2=angleValue3L;
myData.motorB1=angleValue3;
myData.motorB2=angleValue4;
    }
    else if (potValue4>=570) //KINHSH DEKSIA kai ligo PISW
    {
        angleValue3 = map(potValue3, 570, 1023, 0, 255);
        angleValue3L = angleValue3/2;
        angleValue4 = 0;
//        Serial.print("DEKSIA - ligo PISW: ");
myData.motorA1=angleValue3L;
myData.motorA2=angleValue4;
myData.motorB1=angleValue4;
myData.motorB2=angleValue3;
    }
    else //KINHSH DEKSIA FULL
    {
        angleValue3 = map(potValue3, 570, 1023, 0, 255);
        angleValue4 = 0;
//        Serial.print("DEKSIA FULL: ");

```

```

        myData.motorA1=angleValue3;
        myData.motorA2=angleValue4;
        myData.motorB1=angleValue3;
        myData.motorB2=angleValue4;
    }
}

if (potValue4<=450) //KINHSH MPROSTA
{
    if (potValue3<=450) //KINHSH MPROSTA kai ligo ARISTERA
    {
        angleValue4 = map(potValue4, 0, 450, 255, 0);
        angleValue4L = angleValue4/2;
        angleValue3 = 0;
        // Serial.print("MPROSTA - ligo ARISTERA: ");
        myData.motorA1=angleValue3;
        myData.motorA2=angleValue4;
        myData.motorB1=angleValue4L;
        myData.motorB2=angleValue3;
    }
    else if (potValue3>=570) //KINHSH MPROSTA kai ligo DEKSIA
    {
        angleValue4 = map(potValue4, 0, 450, 255, 0);
        angleValue4L = angleValue4/2;
        angleValue3 = 0;
        // Serial.print("MPROSTA - ligo DEKSIA: ");
        myData.motorA1=angleValue3;
        myData.motorA2=angleValue4L;
        myData.motorB1=angleValue4;
        myData.motorB2=angleValue3;
    }
    else //KINHSH MPROSTA FULL
    {
        angleValue4 = map(potValue4, 0, 450, 255, 0);
        angleValue3 = 0;
        // Serial.print("MPROSTA FULL: ");
        myData.motorA1=angleValue3;
        myData.motorA2=angleValue4;
        myData.motorB1=angleValue4;
        myData.motorB2=angleValue3;
    }
}

if (potValue4>=570) //KINHSH PISW
{
    if (potValue3<=450) //KINHSH PISW kai ligo ARISTERA
    {
        angleValue4 = map(potValue4, 570, 1023, 0, 255);
        angleValue4L = angleValue4/2;
        angleValue3 = 0;
        // Serial.print("PISW - ligo ARISTERA: ");
        myData.motorA1=angleValue4;
        myData.motorA2=angleValue3;
        myData.motorB1=angleValue3;
        myData.motorB2=angleValue4L;
    }
    else if (potValue3>=570) //KINHSH PISW kai ligo DEKSIA
    {

```

```

    angleValue4 = map(potValue4, 570, 1023, 0, 255);
    angleValue4L = angleValue4/2;
    angleValue3 = 0;
//    Serial.print("PISW - ligo DEKSIA: ");
    myData.motorA1=angleValue4L;
    myData.motorA2=angleValue3;
    myData.motorB1=angleValue3;
    myData.motorB2=angleValue4;
}
else //KINHSH PISW FULL
{
    angleValue4 = map(potValue4, 570, 1023, 0, 255);
    angleValue3 = 0;
//    Serial.print("PISW FULL: ");
    myData.motorA1=angleValue4;
    myData.motorA2=angleValue3;
    myData.motorB1=angleValue3;
    myData.motorB2=angleValue4;
}
}
}

```

Όταν δεν κινείται το Joystick 2, αλλάζει σε 0 τις τέσσερις αντίστοιχες μεταβλητές της struct, έτσι ώστε να σταματά η κίνηση των DC κινητήρων.

```

if ( (potValue3>450) && (potValue3<570) && (potValue4>450) &&
(potValue4<570) )//STOP CAR
{
//    Serial.print("STOP!!!");
    myData.motorA1=0;
    myData.motorA2=0;
    myData.motorB1=0;
    myData.motorB2=0;
}
}

```

Αποστολή δεδομένων μέσω της δομής myData.

```

radio.write(&myData, sizeof(myData));
delay(5);

if (potValue1 < 300)
{
    delay(500); //Delay gia na prolavainw na afinw to xeri mou
}
else if (potValue1 > 720)
{
    delay(200); //Delay mikro gia na min stelnv polles entoles
}
}

```

Οι παρακάτω εντολές αναφέρονται στην λειτουργία λήψης δεδομένων από το όχημα. Το χειριστήριο σταματάει να στέλνει δεδομένα και τώρα δέχεται τις μετρήσεις από τον αισθητήρα (DHT11) μόνο στην περίπτωση όπου `sensor1 == 1` , δηλαδή μόνο όταν έχει δοθεί εντολή από το χειριστήριο για να λάβω νέες τιμές από τον αισθητήρα.

```
// LEITOURGIA LIPSIS APO AYTOKINITO
radio.startListening();
while (radio.available()){

  radio.read(&myData, sizeof(myData));
  sensor1 = myData.sensor;
  humidity1 = myData.humidity;
  temperature1 = myData.temperature;

//Temperature and Humidity
  if (sensor1 == 1)
  {
    Serial.print("Current humidity = ");
    Serial.print(humidity1);
    Serial.print("% ");
    Serial.print("temperature = ");
    Serial.print(temperature1);
    Serial.println("C ");
  } //end if

} //end while RADIO AVAILABLE

} //Telos void loop
```

4.4 Συνδεσμολογία

Car connections

ARDUINO UNO

Board voltage Vin -> 7.4V (2x 18650 Battery)

3.3V -> VCC pin NRF24L01

GND -> GND pin NRF24L01

GND -> GND pin Input 7.4V

Digital 2 -> Input Relay Module

Digital 3 -> IN1 Drive chip L298N

Digital 4 -> Servo digital input (PWM)

Digital 5 -> DHT11 signal pin

Digital 6 -> IN2 Drive chip L298N

Digital 7 -> CE pin

Digital 8 -> CSN pin

Digital 9 -> IN4 Drive chip L298N

Digital 10 -> IN3 Drive chip L298N

Digital 11 -> MOSI pin

Digital 12 -> MISO pin

Digital 13 -> SCK pin

DHT11 Humidity Sensor

VCC -> 5V Power Board

GND -> GND Power Board

Signal pin -> Digital 5

Motor drive chip

Board voltage - > *7.4V (2x 18650 Battery)*

VCC - > VCC Input 7.4V

GND - > GND Input 7.4V

ENA - > 5V

ENB - > 5V

IN1 - > Digital 3 Arduino pin

IN2 - > Digital 6 Arduino pin

IN3 - > Digital 10 Arduino pin

IN4 - > Digital 9 Arduino pin

Relay Module

Board voltage - > *5V Power Board*

GND - > GND Power Board

Trigger pin - > Digital 2 Arduino pin

COM - > 5V Power Board

NO - > 5V Vout for Lights PCB

Controller connections

NRF24L01 , ίδια συνδεσμολογία και στις δύο πλακέτες

<i>CE</i>	->	Digital 7 Arduino pin
<i>CSN</i>	->	Digital 8 Arduino pin
<i>MOSI</i>	->	Digital 11 Arduino pin
<i>MISO</i>	->	Digital 12 Arduino pin
<i>SCK</i>	->	Digital 13 Arduino pin
<i>VCC</i>	->	3V3 Arduino pin
<i>GND</i>	->	GND Arduino pin

ARDUINO NANO

<i>Board voltage Vin</i>	->	5V (18650 + Step up module)
3.3V	->	VCC pin NRF24L01
GND	->	GND pin NRF24L01
GND	->	GND pin Input 7.4V
Digital 7	->	CE pin
Digital 8	->	CSN pin
Digital 11	->	MOSI pin
Digital 12	->	MISO pin
Digital 13	->	SCK pin

JOYSTICK 1 (Servo motor, DHT11, Lights)

VCC	->	5V Power Board
GND	->	GND Power Board
VRx	->	analog pin A0
VRy	->	analog pin A1
SW	->	-----

JOYSTICK 2 (Κίνηση οχήματος)

VCC	->	5V Power Board
GND	->	GND Power Board
VRx	->	analog pin A3
VRy	->	analog pin A4
SW	->	-----

Completed Project - Photos



Εικόνα 4.33: Car – front view



Εικόνα 4.34: Car – top view

5. Μελλοντικές Επεκτάσεις

- ✚ Αυτόνομη λειτουργία Lights χρησιμοποιώντας LDR(Φωτοαντίσταση). Αυτό μπορεί να επιτευχθεί με δύο τρόπους: είτε με προσθήκη κώδικα στο όχημα, ή με αυτόνομο κύκλωμα με τη χρήση τρανζίστορ.
- ✚ Επέκταση εμβέλειας στο NRF24L01 με μεγαλύτερη κεραία (εμβέλεια έως 1000 μέτρα σε ανοικτό πεδίο).
- ✚ Ισχυρότερος μικροελεγκτής. Μπορούμε να χρησιμοποιήσουμε ένα Arduino Mega 2560 το οποίο διαθέτει περισσότερες αναλογικές και ψηφιακές εισόδους, εξόδους. Έτσι, μπορούμε να χρησιμοποιήσουμε πολύ μεγαλύτερο εύρος αισθητήρων. Διαθέτοντας περισσότερα Digital I/O από αυτά που έχει το Arduino UNO, μπορούμε να χρησιμοποιήσουμε αισθητήρες αποφυγής εμποδίων (π.χ. obstacle avoidance, Ultrasonic sensors), καθώς και άλλα αισθητήρια για μετρήσεις του περιβάλλοντος (π.χ. smoke/gas sensor, rain sensor).
- ✚ Χρήση buzzer module στο χειριστήριο για ηχητικές ενδείξεις (π.χ. όταν βρεθούν εμπόδια παντού γύρω από το όχημα, σε περίπτωση που χαθεί η επικοινωνία μεταξύ οχήματος και χειριστηρίου κλπ.).
- ✚ Προσθήκη Gripper στη μπροστά μεριά του οχήματος χρησιμοποιώντας servo motors.
- ✚ Solar panel για την φόρτιση των μπαταριών του οχήματος και συνεπώς την πλήρη αυτονομία αυτού.
- ✚ OLED (0.91 inch) Screen. Προσθήκη οθόνης στο χειριστήριο, για να βλέπουμε τις μετρήσεις που λαμβάνουμε από το όχημα.
- ✚ Auto Mode: Προσθήκη sub program στο όχημα για αυτόνομη λειτουργία. Σκοπός είναι με ένα κουμπί από το χειριστήριο να στέλνεται η αντίστοιχη εντολή και το όχημα να μπαίνει σε auto mode. Στη λειτουργία αυτή θα κινείται αυτόνομα, αποφεύγοντας εμπόδια με τη βοήθεια αισθητήρων όπως Ultrasonic και Obstacle avoidance sensors. Επίσης, ανά τακτά χρονικά διαστήματα θα λαμβάνει τιμές από τους αισθητήρες που διαθέτει (π.χ. θερμοκρασίας, υγρασίας, καπνού, φωτός κλπ.) και θα τα στέλνει στο χειριστήριο για εμφάνιση στην οθόνη.
- ✚ Data Storage to SD card: Με την προσθήκη ενός SD card module στο χειριστήριο, οι τιμές που λαμβάνονται θα αποθηκεύονται σε κάρτα SD.
- ✚ Raspberry Pi: Το όλο project μπορεί να αναβαθμιστεί για να έχει περισσότερες δυνατότητες χρησιμοποιώντας διαφορετική πλακέτα από το Arduino board (more flexible, additional functions, internet connection, graphics).

- ✚ Χρησιμοποιώντας πλακέτα Raspberry Pi, υπάρχουν δυνατότητες χαρτογράφησης του χώρου, το οποίο μπορεί να βοηθήσει στο να προστεθεί λειτουργία το όχημα να θυμάται την αρχική του θέση και να μπορεί να επιστρέψει σε αυτή εάν του ζητηθεί. Επίσης, με κατάλληλους αισθητήρες μπορεί να υπάρξει λειτουργία έτσι ώστε κατόπιν αιτήματος από το χειριστήριο, το όχημα να πηγαίνει προς αυτό (λειτουργία go-home or come-here).
- ✚ Προστατευτικό κάλυμμα: Το όχημα θα μπορούσε να περικλείεται από ένα κάλυμμα, για προστασία από ήλιο, υγρασία, σκόνη και χτυπήματα. Η προστασία μπορεί να είναι από πλαστικό ή κάποιο άλλο ελαφρύ υλικό και επίσης μπορεί να σχεδιαστεί και να φτιαχτεί με 3D printer.

6. Βιβλιογραφία

LINKS

1. R1 --> <https://www.in.gr/2018/01/31/tech/rompotiko-oxima-kanei-delivery-binteo/>
2. Pandora --> <http://excellence.minedu.gov.gr/draseis/listing/194-pandora>
3. USAR vehicle --> http://www.latsis-foundation.org/content/actions/action_245/inline/ell/actionInline_245_5550_70719bc2d.pdf
4. R+D Project Robot --> http://www.latsis-foundation.org/content/actions/action_245/inline/ell/actionInline_245_5550_70719bc2d.pdf
5. Curiosity --> <https://el.wikipedia.org/wiki/Curiosity>
6. Arduino Boards --> <https://www.arduino.cc/>
7. Arduino Details --> <https://el.wikipedia.org/wiki/Arduino>
8. <https://stackoverflow.com/>
9. <http://www.circuitbasics.com/>
10. <https://github.com/>
11. <https://www.brainy-bits.com/>
12. <https://www.instructables.com/>

BOOKS

1. **ΕΙΣΑΓΩΓΗ ΣΤΗ ΡΟΜΠΟΤΙΚΗ**, Φώτης Ν. Κουμπουλής & Βασίλης Γ. Μέρτζιος, Παπασωτηρίου 2002
2. **ΗΛΕΚΤΡΟΝΙΚΗ**, Albert Malvino & David J. Bates, Εκδόσεις Τζιόλα 2011