

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Συγγραφέας: Σάλλαρης Γιάννης (ΑΜ:222017003)

ΤΙΤΛΟΣ

“ΣΧΕΔΙΑΣΗ ΚΙΝΗΣΗΣ ΑΥΤΟΝΟΜΗΣ ΚΙΝΟΥΜΕΝΗΣ ΜΟΝΑΔΑΣ ΣΕ ΠΕΡΙΒΑΛΛΟΝ ΕΜΠΟΔΙΩΝ”

Επιβλέπουσα καθηγήτρια: Δρ. Παρασκευή Ζαχαρία, Επίκουρη Καθηγήτρια

Αθήνα, 2023

Η παρούσα διπλωματική εργασία εγκρίθηκε ομόφωνα από την τριμελή εξεταστική επιτροπή, η οποία ορίστηκε από την Γ.Σ. του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής του Πανεπιστημίου Δυτικής Αττικής, σύμφωνα με το νόμο και τον εγκεκριμένο Οδηγό Σπουδών του τμήματος.

Επιτροπή Αξιολόγησης:

ΟΝΟΜΑΤΕΠΩΝΥΜΟ	ΥΠΟΓΡΑΦΗ
ΖΑΧΑΡΙΑ ΠΑΡΑΣΚΕΥΗ	
ΧΑΜΗΛΟΘΩΡΗΣ ΓΕΩΡΓΙΟΣ	
ΧΑΤΖΟΠΟΥΛΟΣ ΑΒΡΑΑΜ	

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Σάλλαρης Ιωάννης με αριθμό μητρώου 222017003 φοιτητής του Πανεπιστημίου Δυτικής Αττικής, της Σχολής Μηχανικών, του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής, δηλώνω υπεύθυνα ότι:

« Είμαι συγγραφέας αυτής της προπτυχιακής/διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών

Σάλλαρης Ιωάννης

ΕΥΧΑΡΙΣΤΙΕΣ

Ολοκληρώνοντας τη φοίτησή μου στο Τμήμα Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής, της Σχολής Μηχανικών, του Πανεπιστημίου Δυτικής Αττικής, θα ήθελα να εκφράσω ευχαριστίες από καρδιάς σε όλους τους καθηγητές του τμήματος για την μετάδοση των πολύτιμων γνώσεων τους. Ιδιαίτερω, θα ήθελα να ευχαριστήσω την επίκουρη καθηγήτρια Δρ. Παρασκευή Ζαχαρία, ως υπεύθυνη για την εποπτεία της διπλωματικής μου εργασίας.

Τέλος, ένα μεγάλο ευχαριστώ στην οικογένειά μου για την αμέριστη στήριξη κατά τη διάρκεια των σπουδών μου και κυρίως στην μητέρα μου για την ενθάρρυνση και την έμπρακτη συμπαράστασή της.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Περίληψη	8
Abstract	8
Εισαγωγή	9
ΜΕΡΟΣ Α΄ ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ	
Κεφάλαιο 1: Αυτόνομα οχήματα	10
1.1 Ορισμός	10
1.2 Τεχνολογία αυτόνομων οχημάτων	10
1.3 Ιστορική αναδρομή – Εξέλιξη αυτόνομων οχημάτων	11
1.4 Τρόπος λειτουργίας αυτόνομων οχημάτων	13
1.5 Επίπεδα Αυτόνομης Οδήγησης	15
Κεφάλαιο 2: Τεχνολογία Αισθητήρων	18
2.1 Ορισμός	18
2.2 Χαρακτηριστικά Αισθητηρίων – Ιδανικό Αισθητήριο	19
2.3 Ταξινόμηση Αισθητηρίων	21
2.4 Ψηφιακά και Αναλογικά Αισθητήρια	22
2.5 Παραπλανητικά Ψηφιακό Αισθητήριο	24
2.6 Ταξινόμηση Ανάλογα με τη Λειτουργία	24
2.7 Το Αισθητήριο ως Μετατροπέας	28

Κεφάλαιο 3: Μικροελεγκτές- Μικροεπεξεργαστές	28
3.1 Ιστορική Αναδρομή	29
3.2 Σύγκριση Μικροελεγκτών-Μικροεπεξεργαστών	31
3.3 Ο Μικροελεγκτής 8051	31
3.4 Η Τεχνολογία Reduced Instruction Set Computers (RISC)	33
3.5 Οι Μικροελεγκτές AVR-Ο ATmega8515	33
3.6 Η Αρχιτεκτονική του ATmega8515	34
Κεφάλαιο 4: Το Arduino	35
4.1 Η Αναπτυξιακή Πλατφόρμα Arduino	35
4.2 Σύντομη Ιστορική Αναδρομή	36
4.3 Τα Χαρακτηριστικά της Πλατφόρμας Arduino	37
4.4 Η Αρχιτεκτονική του Arduino	37
4.5 Τα βασικά στοιχεία της Πλακέτας Arduino	38
4.6 Το λογισμικό του Arduino	40
ΜΕΡΟΣ Β ' ΕΙΔΙΚΟ ΜΕΡΟΣ	43
Εισαγωγή	43
Κεφάλαιο 5: Σχεδιασμός Κατασκευής - Λειτουργικά μέρη	44
5.1 Περιγραφή της κατασκευής του οχήματος	44
5.2 Σχεδίαση κατασκευής οχήματος στο πρόγραμμα Fritzing	46
5.3 Αισθητήρας Απόστασης HC-SR04	48

5.3.1 Λειτουργία αισθητήρα	49
5.4 Σερβοκινητήρας (micro servo)	50
5.4.1 Εντολές διαχείρισης servo	51
5.5 Κινητήρας DC	52
5.5.1 Γέφυρα H	53
5.5.2 Pulse Width Modulation (PWM)	54
5.5.3 L298N Motor Driver	55
5.6 Φωτοεκπέμπουσα διόδος (LED)	57
5.6.1 Εντολές διαχείρισης LED	58
Κεφάλαιο 6: Συγγραφή του Λογισμικού	59
6.1 Χαρτογράφηση και περιγραφή της κίνησης	59
6.2 Κώδικας	64
6.3 Παρατηρήσεις- Επεξηγήσεις Κώδικα	74
Κεφάλαιο 7: Περιορισμοί- Προτάσεις	76
7.1 Σφάλματα	76
7.2 Προτάσεις για βελτιστοποίηση της λειτουργίας του Οχήματος	77
7.3 Συμπεράσματα	77
Βιβλιογραφία	80

Περίληψη

Η παρούσα διπλωματική εργασία είχε ως στόχο την μελέτη της κίνησης που μπορεί να εκτελέσει ένα αυτόνομο όχημα, όταν βρεθεί σε ένα συγκεκριμένο περιβάλλον εμποδίων. Ειδικότερα, αναζητήθηκε ένας αλγόριθμος, ο οποίος θα είναι ικανός να επιλέγει τη βέλτιστη διαδρομή που θα οδηγεί το αυτόνομο όχημα από έναν προορισμό Α σε έναν προορισμό Β, αποφεύγοντας εμπόδια του περιβάλλοντος χώρου. Για το σκοπό αυτό, επιχειρήθηκε ο σχεδιασμός και η υλοποίηση ενός αυτοκινούμενου μηχανητρονικού οχήματος, με τη δυνατότητα ιχνηλάτησης διαδρομής, αξιοποιώντας τις δυνατότητες ενός μικροελεγκτή.

Η διάθρωση της εργασίας περιλαμβάνει το θεωρητικό υπόβαθρο (1-4 κεφάλαια), όπου αναλύονται οι πτυχές της αυτόνομης κίνησης οχημάτων και η τεχνολογία με την οποία αναπτύσσεται, μέσω της βοήθειας αισθητήρων και μικροελεγκτών. Στο ειδικό μέρος (5-7 κεφάλαια), παρουσιάζεται ο τρόπος που υλοποιήθηκε η κατασκευή του οχήματος και αναπτύσσεται ο κώδικας με τον οποίο επιτυγχάνεται η κίνησή του.

Λέξεις-Κλειδιά: Αυτόνομα Οχήματα, Αισθητήρας, Μικροελεγκτής, Arduino, Αισθητήρας Απόστασης HC-SR04, DC Κινητήρας, Σερβοκινητήρας

ABSTRACT

This thesis aimed to study the movement that can be performed by an autonomous vehicle, when it finds itself in a certain environment of obstacles. In particular, an algorithm was sought, which will be able to choose the optimal route that will drive the autonomous vehicle from a destination A to a destination B, avoiding obstacles in the surrounding area. For this purpose, the design and implementation of a self-propelled mechatronic vehicle, with the ability to track a route, utilizing the capabilities of a microcontroller, was attempted.

The erosion of the thesis includes the theoretical background (chapters 1-4) where the aspects of autonomous vehicle movement and the technology with which it is developed, through the help of sensors and microcontrollers, are analyzed. In the special part (chapters 5-7), the way the construction of the vehicle was implemented and the code that achieves its movement is developed.

Key words: Autonomous Vehicles, Sensor, Microcontroller, Arduino, HC-SR04 Distance Sensor, DC Motor, Servo Motor

Εισαγωγή

Οι αλματώδεις εξελίξεις στον τομέα της τεχνολογίας που χαρακτηρίζουν την εποχή που διανύουμε επηρεάζουν σημαντικά την καθημερινότητά μας. Οι εξελίξεις αυτές δεν θα μπορούσαν να αφήσουν ανεπηρέαστο και τον τομέα της αυτοκίνησης που εξελίσσεται συνεχώς. Η ιδέα για τα αυτόνομα οχήματα πηγάζει από το γεγονός ότι ο ανθρώπινος παράγοντας συνιστά την βασική αιτία των τροχαίων ατυχημάτων. Η τεχνολογία, μέσω των “έξυπνων μετακινήσεων” με οχήματα αυτόνομης οδήγησης που έχουν ως αφετηρία τους συστήματα προηγμένης υποβοήθησης οδηγού, αποτελεί το επικρατέστερο σενάριο που αποβλέπει στην αντιμετώπιση των ανεπιθύμητων επιπτώσεων των μεταφορών.

Τα αυτόνομα αυτοκίνητα, μέσα στην επόμενη δεκαετία, αναμένεται να αυξηθούν στην παγκόσμια αγορά, δημιουργώντας νέες θέσεις εργασίας και αποκομίζοντας τεράστια κέρδη για την αυτοκινητοβιομηχανία. Η κυκλοφορία τους αναμένεται να προκαλέσει τεράστιες αλλαγές στην κοινωνία, όπως τη μείωση του αριθμού των θανάτων ή των τραυματισμών από τα τροχαία ατυχήματα, τον περιορισμό της κυκλοφοριακής συμφόρησης και της εκπομπής ατμοσφαιρικών ρύπων (Διαθέσιμο: <https://www.europarl.europa.eu/news/el/headlines/economy/20190110STO23102/>).

Επιπλέον, τα αυτόνομα οχήματα θα αυξήσουν την πρόσβαση στην κινητικότητα ατόμων που δεν μπορούν να οδηγήσουν, όπως είναι, τα ηλικιωμένα άτομα ή τα άτομα με κινητικές αναπηρίες. Το εγχείρημα αυτό εντυπωσιάζει, ταυτόχρονα όμως δημιουργεί εύλογα ερωτηματικά, καθώς απαιτούνται να γίνουν ακόμα αρκετές δοκιμές, προκειμένου να καταστούν λειτουργικά και ασφαλή για τις μεταφορές.

ΜΕΡΟΣ Α΄ ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

Κεφάλαιο 1: Αυτόνομα οχήματα

1.1. Ορισμός

Αυτόνομα (autonomous ή automated vehicles), αυτοκινούμενα (self-driving), οχήματα χωρίς οδηγό (driverless) ή αλλιώς ρομποτικά (robotic cars) αποκαλούνται τα οχήματα που έχουν την ικανότητα να αλληλεπιδρούν με το περιβάλλον και τα άλλα οχήματα μέσω εξαιρετικά ανεπτυγμένων συστημάτων αισθητήρων, πλοήγησης και ασύρματης διασύνδεσης. Η μετακίνησή τους γίνεται εφικτή μέσω υπολογιστή χωρίς την ανάγκη εμπλοκής του ανθρώπινου παράγοντα (Litman, 2022).

1.2. Τεχνολογία αυτόνομων οχημάτων

Ανεπτυγμένα συστήματα ελέγχου αντιλαμβάνονται πληροφορίες αισθητήρων, εμπόδια και οδική σήμανση ώστε να ανακαλύψουν τις κατάλληλες διαδρομές. Τα αυτόνομα οχήματα χρησιμοποιούν εξοπλισμό όπως αισθητήρες, ραντάρ, κάμερες, τεχνητή νοημοσύνη (AI), συσκευές GPS, υπερήχους και λέιζερ μετρητές απόστασης. Παράλληλα, είναι εξοπλισμένα με συστήματα υποβοήθησης οδηγού, τεχνολογίες συνδεσιμότητας με το διαδίκτυο και άλλα μέσα, τα οποία τους δίνουν τη δυνατότητα να λειτουργούν ως συνδεδεμένα, έξυπνα οχήματα (connected vehicles). Με τον τρόπο αυτό μπορούν να λαμβάνουν τα συγκεκριμένα οχήματα εικόνα του περιβάλλοντος λειτουργίας τους, να την αναλύουν και να παίρνουν αποφάσεις οδήγησης με βάση τα δεδομένα. Για παράδειγμα να κινούνται με ασφάλεια , να αντιδρούν, να ανταλλάσσουν πληροφορίες με άλλα παρόμοια οχήματα, με τις οδικές

υποδομές ή με απομακρυσμένες βάσεις δεδομένων, με σκοπό την παροχή νέων υπηρεσιών κινητικότητας.

Τα αυτόνομα αυτοκίνητα είναι η επόμενη σελίδα της σύγχρονης αυτοκίνησης. Οι εταιρείες κινούμενες προς αυτή την κατεύθυνση, διεξάγουν έρευνες αναφορικά με το ποσοστό του αγοραστικού κοινού που θα στραφεί σε αυτά. Εταιρείες που αναπτύσσουν ή δοκιμάζουν αυτόνομα αυτοκίνητα είναι οι εταιρείες της Audi, BMW, Ford, General Motors, Tesla, Volkswagen και Volvo (Lutkevich, 2019 · RedHat, 2021· RambusPress, 2022).

1.3. Ιστορική αναδρομή – Εξέλιξη αυτόνομων οχημάτων

Η ιστορία των αυτόνομων οχημάτων ξεκινά αμέσως μετά τη γέννηση του συμβατικού αυτοκινήτου. Τη δεκαετία του 1920 αρχίζουν οι πρώτες μελέτες πάνω στα συστήματα αυτόματης οδήγησης, μέσω ραδιοσυχνοτήτων και τηλεκατεύθυνσης. Το 1925, ο Francis Houdina παρουσιάζει ένα ραδιοελεγχόμενο αυτοκίνητο στους δρόμους του Μανχάταν χωρίς οδηγό. Το όχημα εκτελούσε ενέργειες, όπως την εκκίνηση του κινητήρα, τη μετατόπιση του κιβωτίου ταχυτήτων και το πάτημα της κόρνας του, σα να υπήρχε ανθρώπινο χέρι στο τιμόνι (Διαθέσιμο στο: https://en.wikipedia.org/wiki/Houdina_Radio_Control).



Εικόνα 1.3.1: Οχημα Houdina, Ανακτήθηκε από:

https://en.wikipedia.org/wiki/Houdina_Radio_Control

Αρκετά χρόνια αργότερα το 1969, ο John McCarthy, περιέγραψε στο βιβλίο του με τίτλο “Ηλεκτρονικά ελεγχόμενα αυτοκίνητα”, ένα μοντέλο που προσομοίαζε με το σύγχρονο αυτόνομο όχημα. Πρόκειται για ένα αυτόματο αυτοκίνητο - σοφέρ, ικανό να πλοηγείται μέσω τηλεοπτικής κάμερας, η οποία θα χρησιμοποιούσε για είσοδο την ίδια με εκείνη της οπτικής εισόδου που διατίθεται στον οδηγό. Οι χρήστες του οχήματος θα είχαν τη δυνατότητα να εισάγουν αρχικά τον προορισμό, με τη βοήθεια πληκτρολογίου και εν συνέχεια το αυτοκίνητο να αυτό-οδηγείται προς αυτόν. Επιπλέον, περιγράφονταν εντολές που θα επέτρεπαν στους χρήστες να κάνουν στάση ή να αλλάζουν προορισμό, όπως και να επιβραδύνουν ή να επιταχύνουν το όχημα. Ένα τέτοιο όχημα φυσικά δεν ήταν εφικτό να κατασκευαστεί εκείνη την εποχή, ωστόσο ο σχεδιασμός του McCarthy έδωσε την ευκαιρία σε άλλους ερευνητές μελλοντικά να εργαστούν προς την κατεύθυνση αυτή (Glon & Edelstein, 2020) . Το πρώτο ημι-αυτοματοποιημένο αυτοκίνητο αναπτύχθηκε αργότερα, το 1977 από το ιαπωνικό εργαστήριο “Tsukuba” και απαιτούσε ειδικά διαμορφωμένους δρόμους ώστε να τους ερμηνεύει με τις δυο ενσωματωμένες κάμερες και το αναλογικό υπολογιστικό σύστημα που διέθετε.

Το 1986, ομάδες από το ρομποτικό Ινστιτούτο της σχολής Επιστήμης Υπολογιστών του πανεπιστήμιου Carnegie Mellon, , ξεκίνησαν την κατασκευή οχημάτων με την ονομασία “Navlab” . Τα οχήματα εφοδιάστηκαν με μια πρώιμη έκδοση της τεχνολογίας LiDAR (Light Detection And Ranging) που βασίζονταν στην εκπομπή παλμικής ακτινοβολίας λέιζερ για την τρισδιάστατη χαρτογράφηση του περιβάλλοντα χώρου (Ξύδης, 2020). Το 1996 ο Alberto Broggi από το Πανεπιστήμιο της Πάρμα, σχεδίασε το ARGO, ώστε να μπορεί να ακολουθεί τις λωρίδες του αυτοκινητόδρομου, από μία ειδικά τροποποιημένη Lancia Thema. Το όχημα ήταν εξοπλισμένο με δύο ασπρόμαυρες κάμερες χρησιμοποιώντας αλγόριθμους

στερεοσκοπικής όρασης για την αντίληψη του γύρω περιβάλλοντα χώρου (Bertozzi et al., 1998). Τέλος, ανατρέχοντας στη δεκαετία του 2000, εμφανίζονται σιγά-σιγά τα συστήματα υποβοήθησης παρκαρίσματος ή και αυτό-παρκαρίσματος, αποδεικνύοντας έτσι πως οι αισθητήρες θα διαδραματίσουν καίριο ρόλο στο χώρο των αυτόματων τεχνολογιών του συνολικού οδικού δικτύου.

Το 2004, στις ΗΠΑ, ο οργανισμός DARPA (Defense Advanced Research Projects Agency) χρηματοδοτεί το πρόγραμμα “Grand Challenge” με στόχο να επιταχύνει την ανάπτυξη των αυτόνομων οχημάτων. Απώτερος σκοπός του ήταν οι τεχνολογίες αυτές να εφαρμοστούν σε στρατιωτικά προγράμματα των ΗΠΑ (Διαθέσιμο στο: https://en.wikipedia.org/wiki/DARPA_Grand_Challenge).

Η Google το 2010 προχωρά στη δοκιμή ενός συστήματος για αυτό-οδηγούμενα οχήματα με στόχο τη μείωση του αριθμού των αυτοκινητιστικών ατυχημάτων και δύο χρόνια αργότερα ανακοινώνει ότι τα αυτοκίνητά της διένυσαν περίπου 300.000 μίλια, χωρίς να προκληθεί ατύχημα. Το 2014 παρουσιάζει ένα αυτοκίνητο που για πρώτη φορά αποκαλείται 100% αυτόνομο όχημα χωρίς οδηγό, τιμόνι και πεντάλ γκαζιού (Lardinois, 2012) . Παράλληλα, εταιρείες αυτοκινητοβιομηχανίας, όπως η BMW, η General Motors και η Mercedes Benz, εργάζονταν με βάση τη τεχνολογία τους για τα αυτό-οδηγούμενα αυτοκίνητα. Η Tesla Motors από το 2015 έθεσε σε κυκλοφορία αυτοκίνητα με την τεχνολογία αυτόματου πιλότου και τη δυνατότητα να παρκάρουν αυτόματα σε θέσεις στάθμευσης, χωρίς οδηγό.

1.4. Τρόπος λειτουργίας αυτόνομων οχημάτων

Ο τρόπος με τον οποίο λειτουργούν τα αυτόνομα οχήματα μπορεί να διαφέρει σημαντικά μεταξύ των πολλών και διαφορετικών τεχνολογιών που χρησιμοποιούνται

για τη δημιουργία τους. Σε βασικό επίπεδο, το όχημα δημιουργεί έναν χάρτη του περιβάλλοντός του, χρησιμοποιώντας μια σειρά από αισθητήρες, οι οποίοι του δίνουν πληροφορίες για τον δρόμο γύρω από το ίδιο το όχημα και για τα άλλα οχήματα στο πέρασμά του. Οι προγραμματιστές αυτόνομων αυτοκινήτων χρησιμοποιούν τεράστιες ποσότητες δεδομένων από συστήματα αναγνώρισης εικόνας, μαζί με μηχανική μάθηση και νευρωνικά δίκτυα, για να κατασκευάσουν συστήματα που μπορούν να οδηγούν αυτόνομα. Τα δεδομένα αυτά περιλαμβάνουν εικόνες από κάμερες από τις οποίες, το νευρωνικό δίκτυο μαθαίνει να αναγνωρίζει τα φανάρια, τα δέντρα, τα κράσπεδα, τους πεζούς, τις πινακίδες και άλλα μέρη οποιουδήποτε δεδομένου περιβάλλοντος οδήγησης. Οι πληροφορίες αυτές στη συνέχεια ερμηνεύονται από έναν σύνθετο αλγόριθμο μηχανικής μάθησης, ο οποίος εξάγει ένα σύνολο ενεργειών που πρέπει να κάνει το όχημα. Οι συγκεκριμένες ενέργειες αλλάζουν και ενημερώνονται συνεχώς, καθώς ο αλγόριθμος λαμβάνει νέες πληροφορίες για το περιβάλλον του οχήματος.




















Εκτός από τις κάμερες και τους αισθητήρες που χρησιμοποιούνται από τα περισσότερα προηγμένα συστήματα υποβοήθησης οδηγού, για να δημιουργήσουν έναν εσωτερικό χάρτη του περιβάλλοντος χώρου του οχήματος περιλαμβάνονται και τα εξής :



- **Lidar:** Στέλνει χιλιάδες δέσμες υπέρυθρου φωτός λέιζερ σε αντικείμενα με σκοπό να υπολογίσει τις αποστάσεις και να αποφύγει αντικείμενα. Παρέχει αποτελέσματα υψηλής ανάλυσης με βέλτιστη ακρίβεια.
- **Ραντάρ:** Χρησιμοποιεί ραδιοκύματα για τη μέτρηση γωνιών, εμβέλειας και ταχύτητες αντικειμένων στις περισσότερες περιβαλλοντικές συνθήκες. Λειτουργεί με βάση την αρχή της ηλεκτρομαγνητικής ακτινοβολίας.

- **Σόναρ:** Εντοπίζει μεγάλα αντικείμενα κατασκευασμένα από στερεά υλικά, όπως μέταλλο και κεραμικά σε μικρές αποστάσεις.
- **Αδρανειακό σύστημα πλοήγησης:** Βοηθά τα αυτόνομα αυτοκίνητα να σταθεροποιηθούν.
- **GPS:** Εντοπίζει γεωγραφικά στοιχεία με αριθμητικές συντεταγμένες συμπεριλαμβανομένων γεωγραφικού πλάτους και μήκους, ενώ πλοηγείται συνδυάζοντας συντεταγμένες GPS σε πραγματικό χρόνο με άλλες εφαρμογές δεδομένων ψηφιακών χαρτών (Lutkevich, 2019 · RedHat, 2021 · RambusPress, 2022).

1.5. Επίπεδα Αυτόνομης Οδήγησης

Η SAE International (Society of Automotive Engineers) είναι μια παγκόσμια επαγγελματική ένωση και οργανισμός προτύπων που εστιάζει κυρίως σε παγκόσμιες

	Who steers, accelerates, brakes?	Monitoring of the driving situation	Backup system for dynamic driving activities	Driving Mode
0				Driver-controlled
1	 			Assisted driving
2				Partially-automated driving
3				Highly-automated driving
4				Fully-automated driving
5				Autonomous Driving

 Human  Assistance System

Εικόνα 1.5.1: Τα επίπεδα της SAE, Ανακτήθηκε από:

<https://www.dekra-solutions.com/2018/11/leveling-up/?lang=en>

βιομηχανίες μεταφορών, όπως η αεροδιαστημική, η αυτοκινητοβιομηχανία και τα εμπορικά οχήματα. Το 2014 δημιούργησε έξι επίπεδα αυτόνομης οδήγησης τα οποία έκτοτε έχουν υιοθετηθεί από το υπουργείο μεταφορών των Η.Π.Α (SAE, 2021).

Αναλυτικά τα στάδια αυτά παρατίθενται πιο κάτω και είναι τα εξής:

Επίπεδο 0: Χωρίς αυτοματισμό οδήγησης

Στο επίπεδο 0, τα οχήματα στερούνται εντελώς τεχνολογίας αυτοματισμού οδήγησης. Ο οδηγός είναι αφοσιωμένος και πλήρως υπεύθυνος για τη λειτουργία του οχήματος (τιμόνι, επιτάχυνση, φρενάρισμα, στάθμευση κλπ.). Ορισμένα οχήματα επιπέδου 0 μπορούν να προσφέρουν περιορισμένες δυνατότητες υποβοήθησης οδηγού. Τέτοιες μπορεί να είναι προειδοποιήσεις και ειδοποιήσεις ή ακόμα και επείγουσες παρεμβάσεις ασφαλείας που δεν θεωρούνται αυτόνομες λειτουργίες, όπως το αυτόματο φρενάρισμα έκτακτης ανάγκης, η προειδοποίηση εμπρόσθιας σύγκρουσης κλπ (Rambus Press, 2022).

Επίπεδο 1: Υποβοηθούμενη οδήγηση

Στο επίπεδο 1, τα συστήματα αυτοκινήτων παρέχουν συνεχή υποβοήθηση με την επιτάχυνση, το φρενάρισμα ή το τιμόνι. Παράδειγμα τεχνολογίας υποβοήθησης οδηγού επιπέδου 1 αποτελεί το σύστημα υποβοήθησης διατήρησης λωρίδας.

Επίπεδο 2: Μερικός αυτοματισμός οδήγησης

Στο επίπεδο 2, παρέχεται συνεχώς βοήθεια στους οδηγούς από προηγμένα συστήματα υποβοήθησης στην επιτάχυνση, στο φρενάρισμα και στο τιμόνι, τα οποία μπορούν να αναλάβουν τον έλεγχο του οχήματος ανά πάσα ώρα και στιγμή. Τόσο η τεχνολογία επιπέδου 1, όσο και η επιπέδου 2 απαιτούν από τον οδηγό να είναι

προσεχτικός και αφοσιωμένος κατά την οδήγηση του μηχανοκίνητου οχήματος (Rambus Press, 2022).

Επίπεδο 3: Αυτοματισμός οδήγησης υπό όρους

Στο επίπεδο 3, όλες οι εργασίες οδήγησης χειρίζονται αυτόνομα από τα οχήματα. Ωστόσο, εάν το προηγμένο σύστημα υποβοήθησης χρειαστεί βοήθεια ή σταματήσει ξαφνικά να λειτουργεί αποτελεσματικά, ο οδηγός πρέπει πάντα να είναι διαθέσιμος για να αναλάβει το τιμόνι. Συστήματα αυτοματισμού οδήγησης υπό όρους επιπέδου 3 έχουν αναπτυχθεί από πολλές μεγάλες αυτοκινητοβιομηχανίες συμπεριλαμβανομένης της εταιρείας Audi. Στο επίπεδο αυτό εντάσσονται και τα Tesla με τη δυνατότητα του αυτόματου πιλότου (Καμινιάρης, 2020 · Rambus Press, 2022).

Επίπεδο 4: Υψηλός αυτοματισμός οδήγησης

Σε αυτό το επίπεδο, τα αυτόνομα συστήματα οχημάτων είναι πλήρως υπεύθυνα για όλες τις εργασίες οδήγησης και πλοήγησης. Οι οδηγοί δεν χρειάζεται να εμπλακούν ή να είναι έτοιμοι να πάρουν τον έλεγχο του οχήματος. Ωστόσο, τα συστήματα αυτοματισμού οδήγησης επιπέδου 4, περιορίζονται σε συγκεκριμένες γεωγραφικές τοποθεσίες και τα οχήματα που τα διαθέτουν δε μπορούν να ταξιδέψουν εκτός καθορισμένων περιοχών εξυπηρέτησης ή κατά τη διάρκεια επικίνδυνων καιρικών συνθηκών. Ο αυτοματισμός οδήγησης επιπέδου 4 είναι ιδιαίτερα κατάλληλος για ταξί και λεωφορεία χωρίς οδηγό σε καθορισμένες διαδρομές, για φορτηγά που μεταφέρουν εμπορεύματα εντός συγκεκριμένων γεωγραφικών ορίων, καθώς και για λεωφορεία επιβατών και φορτίου αεροδρομίου (Rambus Press, 2022).

Επίπεδο 5: Πλήρης αυτοματισμός οδήγησης

Στο επίπεδο 5, οι επιβάτες απλώς ορίζουν έναν προορισμό και μπορούν να εργαστούν, να κοιμηθούν, να παρακολουθήσουν ταινίες και να παίξουν παιχνίδια, αφήνοντας τα αυτόνομα συστήματα να έχουν τον πλήρη έλεγχο των καθηκόντων της οδήγησης. Στο μέλλον, οχήματα εξοπλισμένα με συστήματα πλήρους αυτοματισμού οδήγησης επιπέδου 5, θα λειτουργούν ανεξάρτητα και καθολικά σε όλες τις καιρικές συνθήκες και δρόμους (Rambus Press, 2022).

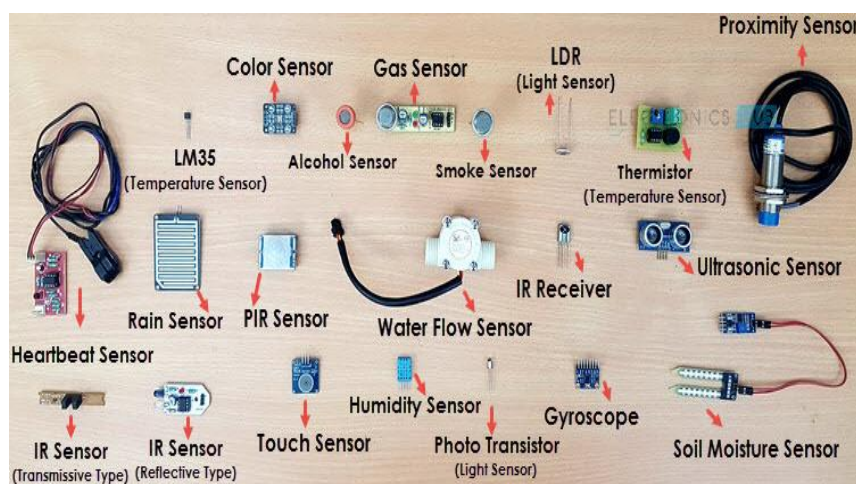
Κεφάλαιο 2: Τεχνολογία Αισθητήρων

Αν παρατηρήσει κανείς τις συσκευές που χρησιμοποιούνται στην καθημερινή μας ζωή, εύλογα θα οδηγηθεί στο συμπέρασμα, πόσο αναπόσπαστο κομμάτι της αποτελούν οι αισθητήρες, οι οποίοι διευκολύνουν την καθημερινότητα όλων. Υπάρχουν διαφορετικά είδη αισθητήρων στο σπίτι, στο γραφείο, στο αυτοκίνητο. Μπορεί να είναι απλοί αισθητήρες, όπως ο οπτικός αισθητήρας που διαθέτει το ποντίκι του ηλεκτρονικού υπολογιστή, ή πιο προηγμένοι, όπως οι αισθητήρες ανίχνευσης εμποδίων στους οποίους βασίζεται η κίνηση των αυτόνομων οχημάτων. Όσο εξελίσσεται η τεχνολογία και αυξάνεται η διαθεσιμότητα των φθηνών μικροεπεξεργαστών, τόσο μεγαλώνει και το ενδιαφέρον για τους αισθητήριους μηχανισμούς. Όμως, πώς μπορούμε να ορίσουμε έναν αισθητήρα;

2.1 Ορισμός

Ο αισθητήρας είναι μια συσκευή, η οποία δέχεται φυσικό σήμα από το περιβάλλον της (συσκευή εισόδου) και το μετατρέπει σε δεδομένα, που μπορούν να

ερμηνευθούν από τον άνθρωπο ή από μια άλλη συσκευή. Μπορούμε να πούμε ότι αποτελεί μέρος ενός μεγαλύτερου συστήματος που παρέχει είσοδο σε ένα κύριο σύστημα ελέγχου, όπως ένας επεξεργαστής ή ένας μικροελεγκτής. Οι αισθητήρες μπορεί να είναι ηλεκτρονικοί. Στην περίπτωση των ηλεκτρονικών αισθητήρων, το φυσικό σήμα μετατρέπεται σε ηλεκτρικό ή μπορεί οι αισθητήρες να είναι τελείως απλοί, όπως το γυάλινο θερμόμετρο, το οποίο παρέχει οπτικά δεδομένα στον άνθρωπο. Οι αισθητήρες χρησιμοποιούνται από τον άνθρωπο για τη μέτρηση της θερμοκρασίας, της πίεσης, για την ανίχνευση καπνού, φωτιάς και σε πολλές άλλες περιπτώσεις (Jost, 2019 · Teja, 2021).



Εικόνα 2.1.1: Αισθητήρια, Ανακτήθηκε από:

<https://www.electronicshub.org/different-types-sensors/>

2.2 Χαρακτηριστικά Αισθητηρίων – Ιδανικό Αισθητήριο

Στην πιο απλή του μορφή, ένα αισθητήριο μπορεί να θεωρηθεί ως σύστημα με είσοδο $x(t)$ και έξοδο $y(t)$. Το αισθητήριο μπορεί να είναι αυτό-διεγερόμενο (self-exciting) αισθητήριο ή τροποποιούμενο (modulating). Βασικό χαρακτηριστικό του αυτό-διεγερόμενου αισθητηρίου είναι ότι η έξοδος είναι συνάρτηση της εισόδου (πχ

θερμοζεύγος ή πιεζοηλεκτρικός κρύσταλλος). Στο τροποποιούμενο αισθητήριο αντίστοιχα, υπάρχει και εξωτερικό σήμα $x_d(t)$ (πχ σταθερό ρεύμα οδήγησης ή τάση αναφοράς) (Τσελές, 2002).

Ένα αισθητήριο χαρακτηρίζεται ως ιδανικό, όταν το σήμα εξόδου είναι σχεδόν ανάλογο με το σήμα εισόδου. Για ένα ιδανικό αισθητήριο ισχύει ότι: $y(t)=S*x(t)$, όπου S , η κλίση της χαρακτηριστικής εξόδου- εισόδου του ιδανικού αισθητηρίου που έχει σταθερή τιμή και ονομάζεται ευαισθησία. Στην πράξη, κανένα αισθητήριο δεν μπορεί να ανταποκριθεί ακαριαία σε κάποια αλλαγή του σήματος εισόδου, αλλά χρειάζεται κάποιο χρονικό διάστημα. Έτσι, μπορούμε να πούμε ότι ένα ιδανικό αισθητήριο έχει τα κάτωθι χαρακτηριστικά (Τσελές, 2002· Πογαρίδης , 2020).

- **Ευαισθησία:** Είναι το μέτρο του βαθμού με τον οποίο μια στοιχειώδης αλλαγή του σήματος εισόδου αλλάζει το σήμα εξόδου. Δίνεται από τον λόγο της μεταβολής της απόλυτης τιμής εξόδου προς τη μεταβολή της απόλυτης τιμής εισόδου. Με τον όρο “διασταυρούμενη ευαισθησία” αναφερόμαστε στην ευαισθησία στις περιβαλλοντικές παραμέτρους. Όταν είναι υψηλή και δεν μπορεί να μοντελοποιηθεί η διασταυρούμενη ευαισθησία είναι γενικά ανεπιθύμητη.
- **Γραμμικότητα:** Είναι το μέτρο της συμπεριφοράς του σήματος εξόδου κατά τη μεταβολή του σήματος εισόδου.
- **Χρόνος απόκρισης:** Είναι ο χρόνος που χρειάζεται το σήμα εξόδου για να φθάσει το 90% της τελικής του τιμής.
- **Δυναμική περιοχή:** Είναι ο λόγος της μέγιστης τιμής εισόδου προς την ελάχιστη μετρούμενη τιμή εισόδου. Χρησιμοποιείται για τη μέτρηση του

εύρους μεταξύ του χαμηλότερου και του υψηλότερου ορίου των τιμών εισόδου σε έναν αισθητήρα. Μετριέται σε decibels.

- **Ανάλυση:** Είναι η ελάχιστη διαφορά μεταξύ δύο τιμών μέτρησης ενός αισθητήρα. Συνήθως, ισούται με το χαμηλότερο όριο της δυναμικής περιοχής.
- **Εύρος ζώνης- συχνότητα:** Αναφέρεται στην ταχύτητα με την οποία ο αισθητήρας παρέχει μία ακολουθία τιμών. Είναι δηλαδή ο αριθμός των μετρήσεων που μπορεί να κάνει σε ένα δευτερόλεπτο. Μετριέται σε Hertz.
- **Ακρίβεια:** Ορίζεται ως ο βαθμός συμμόρφωσης των μετρούμενων τιμών και των πραγματικών τιμών.
- **Σφάλμα:** Ορίζεται ως η διαφορά μεταξύ των μετρούμενων τιμών εξόδου του αισθητήρα και των πραγματικών τιμών που μετρούνται μέσα σε ένα συγκεκριμένο πλαίσιο λειτουργίας. Γενικά, στους αισθητήρες μπορούμε να έχουμε τα συστηματικά σφάλματα που μπορούν να προβλεφθούν και προκαλούνται από παράγοντες και διαδικασίες που εύκολα μοντελοποιούνται και τα τυχαία σφάλματα. Τα τυχαία σφάλματα δεν μπορούν να προβλεφθούν με κάποιο μοντέλο, ούτε μπορούν να μετριάστούν χρησιμοποιώντας ακριβή μηχανήματα για την κατασκευή των αισθητήρων. Ωστόσο μπορούν να περιγραφούν στοχαστικά (Πογαρίδης, 2020).

2.3 Ταξινόμηση Αισθητηρίων

Κατά καιρούς, έχουν διατυπωθεί από αρκετούς συγγραφείς και ειδικούς, διαφορετικές ταξινομήσεις αισθητηρίων. Κάποιες από αυτές μάλιστα είναι απλές και κάποιες άλλες πιο δύσκολες στην κατανόησή τους. Τα αισθητήρια ταξινομούνται ανάλογα με το φυσικό φαινόμενο που χρησιμοποιούν (πχ πιεζοηλεκτρικό,

φωτοβολταϊκό) ή ανάλογα με τη λειτουργία την οποία επιτελούν, όπως για παράδειγμα η μέτρηση μήκους, θερμοκρασίας κ.α (Γαστεράτος και συν., 2013).

Στην πρώτη περίπτωση ταξινόμησης, πρόκειται για μια ολοκληρωμένη παρουσίαση της αισθητήριας διαδικασίας. Η παρουσίαση αυτή, μπορεί να είναι κατά ένα τρόπο παραπλανητική, στην περίπτωση που κάποιος επιθυμεί να συγκρίνει τα προτερήματα δύο διαφορετικών τύπων αισθητηρίων. Ένα χαρακτηριστικό παράδειγμα, αποτελεί η εξέταση και η σύγκριση δύο διαφορετικών τύπων αισθητηρίων θερμοκρασίας σε ξεχωριστούς τομείς μηχανισμών αντίστασης, ημιαγωγών και θερμοηλεκτρικούς μηχανισμούς. Όταν η ταξινόμηση των αισθητηρίων γίνεται με βάση τη λειτουργία τους, καταλήγουμε συχνά σε ένα κατάλογο από αναρίθμητα και άσχετα μεταξύ τους αισθητήρια (Γαστεράτος και συν., 2013).

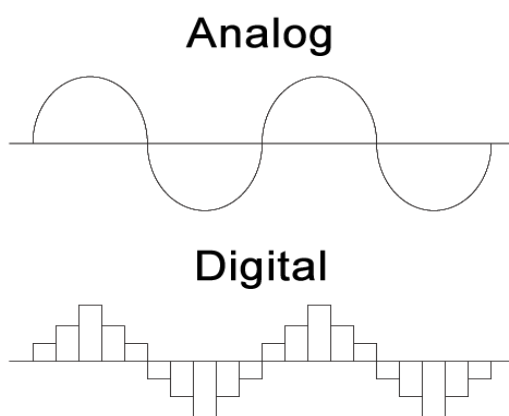
Μια άλλη ταξινόμηση των αισθητηρίων είναι σε ενεργητικά και παθητικά αισθητήρια. Ενεργά αισθητήρια είναι εκείνα που απαιτούν εξωτερικό σήμα διέγερσης ή σήμα ισχύος. Αντίθετα, τα παθητικά αισθητήρια δεν απαιτούν εξωτερικό σήμα ισχύος και παράγουν απευθείας το σήμα εξόδου. Μια τελευταία ταξινόμηση των αισθητηρίων είναι εκείνη που τα ταξινομεί σε αναλογικά και ψηφιακά αισθητήρια.

2.4 Ψηφιακά και Αναλογικά Αισθητήρια

Υπάρχουν πολλοί τρόποι για να κατηγοριοποιήσουμε τους αισθητήρες. Ένας από τους πιο βασικούς είναι η κατηγοριοποίηση σε αναλογικούς και ψηφιακούς. Οι αναλογικοί αισθητήρες μετατρέπουν το φυσικό σήμα σε αναλογικό. Είναι περισσότερο ακριβείς σε σχέση με τους ψηφιακούς, οι οποίοι είναι περιορισμένοι σε ένα πεπερασμένο σύνολο πιθανών τιμών. Η αιτία για την οποία αποδίδουν τα

αναλογικά σήματα και την παραμικρή αλλαγή σε μια φυσική μεταβλητή, είναι ότι, τα αναλογικά σήματα είναι συνεχή. Σε αντίθεση, τα ψηφιακά σήματα, ενώ ακολουθούν τη γενική τάση διακύμανσης περιορίζονται σε σταθερά δεδομένα.

Γενικώς , η διαφορά ανάμεσα σε αναλογικούς και ψηφιακούς αισθητήρες αφορά στο πως ο αισθητήρας εξάγει τα μετρούμενα δεδομένα. Το αναλογικό σήμα που εξάγει ο αισθητήρας τροφοδοτεί έναν analog to digital converter και μετατρέπεται σε ψηφιακό. Όταν ο analog to digital converter βρίσκεται εντός του αισθητήρα, τότε ο αισθητήρας είναι ψηφιακός. Εάν, ο analog to digital converter βρίσκεται εκτός του αισθητήρα (συνήθως μέσα στον μικροελεγκτή ή σε ένα τσιπ ADC) τότε ο αισθητήρας είναι αναλογικός. Οι ψηφιακοί αισθητήρες συνήθως προτιμούνται ,εάν η τιμή και οι προδιαγραφές τους είναι αποδεκτές. Αυτό συμβαίνει, επειδή οι ψηφιακοί αισθητήρες είναι λιγότερο επιρρεπείς σε ηλεκτρικές παρεμβολές και έχουν μικρότερο κίνδυνο σχεδιασμού (Διαθέσιμο στο: <https://predictabledesigns.com/introduction-to-electronic-sensors/>).



Εικόνα 2.4.1: Αναλογικό και ψηφιακό σήμα, Ανακτήθηκε από:

<https://www.fierceelectronics.com/sensors/what-a-sensor>

2.5 Παραπλανητικά Ψηφιακό Αισθητήριο

Υπάρχουν όντως ψηφιακά αισθητήρια; Ο άνθρωπος έχει προσαρμοστεί στον αναλογικό κόσμο. Οι μετρούμενες ποσότητες γίνονται αντιληπτές σε αναλογική μορφή. Αν και για ένα μηχανικό, θα ήταν μεγάλη διευκόλυνση να χρησιμοποιούσε ψηφιακά αισθητήρια στα συστήματα των υπολογιστών, εντούτοις τα ψηφιακά αισθητήρια στην πραγματικότητα συλλαμβάνουν αναλογικές ποσότητες και χρησιμοποιούν τις ίδιες φυσικές αρχές με τα αντίστοιχα αναλογικά.

Αν ερευνήσει κανείς ένα τυπικό ψηφιακό αισθητήριο, θα παρατηρήσει ότι είναι ένας απολύτως αναλογικός μηχανισμός, ο οποίος λειτουργεί έτσι ώστε να παράγει μια ψηφιακή έξοδο. Ένα αισθητήριο που παράγει μια συχνότητα ή μια ακολουθία παλμών αντίστοιχη με τη μετρούμενη ποσότητα, ονομάζεται “παραπλανητικά ψηφιακό”. Όταν η κυματομορφή που έχει παραχθεί μετατρέπεται σε ψηφία π.χ. από ένα απαριθμητικό κύκλωμα, τότε η μέτρηση γίνεται ψηφιακή (Γαστεράτος και συν., 2013) .

2.6 Ταξινόμηση Ανάλογα με τη Λειτουργία

Ανάλογα με τη λειτουργία που εκτελούν οι αισθητήρες και τη φυσική ιδιότητα που μετρούν έχουμε την ακόλουθη λίστα αισθητήρων: αισθητήρας θερμοκρασίας, επιταχυνσιόμετρο, αισθητήρας υπέρυθρων, πίεσης, φωτοανιχνευτής, αισθητήρας υπερήχων, καπνού, αερίου, αλκοόλ, αφής, χρώματος, υγρασίας, θέσης, μαγνητικός αισθητήρας, μικρόφωνο, αισθητήρας ήχου, ροής, στάθμης. Παρακάτω, θα αναλυθούν μερικοί από τους πιο γνωστούς αισθητήρες. Πρόκειται για τους αισθητήρες που

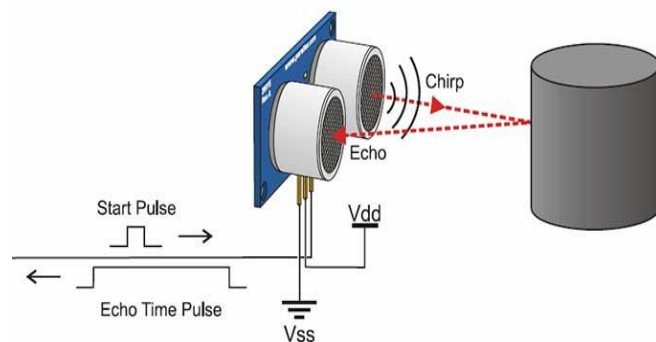
χρησιμοποιούνται κυρίως στην ηλεκτρονική και σε κατασκευές, όπως η κατασκευή της εργασίας.

Αισθητήρας θερμοκρασίας: Ένας από τους πιο κοινούς και δημοφιλείς αισθητήρες είναι ο αισθητήρας θερμοκρασίας. Ένας αισθητήρας θερμοκρασίας, όπως υποδηλώνει και το όνομα του, αισθάνεται τη θερμοκρασία δηλαδή μετράει τις αλλαγές στη θερμοκρασία. Υπάρχουν διαφορετικοί τύποι αισθητήρων θερμοκρασίας, όπως τα ολοκληρωμένα κυκλώματα LM35, DS18B20, τα θερμίστορ και τα θερμοζεύγη. Οι αισθητήρες θερμοκρασίας χωρίζονται σε αναλογικούς και ψηφιακούς. Σε έναν αναλογικό αισθητήρα θερμοκρασίας, οι αλλαγές στη θερμοκρασία αντιστοιχούν σε αλλαγή στη φυσική του ιδιότητα, όπως αντίσταση ή τάση. Ο LM35 είναι ένας κλασικός αναλογικός αισθητήρας θερμοκρασίας. Στον ψηφιακό αισθητήρα θερμοκρασίας, η έξοδος είναι μια διακριτή ψηφιακή τιμή. Τυπικό παράδειγμα αποτελεί ο DS18B20. Οι αισθητήρες θερμοκρασίας συναντώνται στους υπολογιστές, στα κινητά τηλέφωνα, στα αυτοκίνητα, στα air condition και στις βιομηχανίες (Teja, 2021).

Αισθητήρας υπερύθρων: Οι αισθητήρες υπερύθρων είναι αισθητήρες που βασίζονται στο φως και χρησιμοποιούνται σε διάφορες εφαρμογές, όπως είναι η ανίχνευση αντικειμένων. Διακρίνουμε δύο τύπους αισθητήρων υπερύθρων: τους τύπου μετάδοσης (transmissive type) και τους ανακλαστικού τύπου (reflective type). Στον αισθητήρα υπερύθρων τύπου μετάδοσης, ο πομπός υπερύθρων και ο ανιχνευτής υπερύθρων είναι τοποθετημένοι ο ένας απέναντι στον άλλο, έτσι ώστε, όταν ένα αντικείμενο περνά ανάμεσά τους ο αισθητήρας ανιχνεύει το αντικείμενο. Στον ανιχνευτικό τύπο, ο πομπός και ο ανιχνευτής είναι τοποθετημένοι ο ένας δίπλα στον άλλο απέναντι από το αντικείμενο. Όταν ένα αντικείμενο έρχεται μπροστά από τον αισθητήρα, το υπέρυθρο φως από τον πομπό υπερύθρων ανακλάται από το

αντικείμενο και ανιχνεύεται από το δέκτη υπερύθρων και έτσι ο αισθητήρας ανιχνεύει το αντικείμενο. Οι αισθητήρες υπερύθρων χρησιμοποιούνται στα κινητά, στα αυτοκίνητα, στα ρομπότ κτλ (Teja, 2021).

Αισθητήρας υπερήχων: Ο αισθητήρας υπερήχων είναι μια συσκευή αίσθησης τύπου non-contact (χωρίς επαφή) που χρησιμοποιείται για τη μέτρηση της απόστασης καθώς και της ταχύτητας ενός αντικειμένου.



Εικόνα 2.6.1: Λειτουργία HC-SR04 αισθητήρα, Ανακτήθηκε από:

<https://predictabledesigns.com/introduction-to-electronic-sensors/>

Λειτουργεί με βάση τις ιδιότητες των ηχητικών κυμάτων με συχνότητα μεγαλύτερη από αυτή του ακουστικού εύρους. Χρησιμοποιώντας το χρόνο πτήσης του ηχητικού κύματος, ένας αισθητήρας υπερήχων μπορεί να μετρήσει την απόσταση του αντικειμένου. Για τη μέτρηση της ταχύτητας του αντικειμένου χρησιμοποιείται το φαινόμενο Doppler του ηχητικού κύματος.

Φωτοανιχνευτές: Οι φωτοανιχνευτές είναι από τους πιο σημαντικούς αισθητήρες. Ο πιο γνωστός αισθητήρας φωτός είναι το LDR (light dependent resistor). Η ιδιότητά του είναι ότι η αντίστασή του είναι αντιστρόφως ανάλογη με την ένταση του φωτός

δηλαδή, όταν η ένταση του φωτός αυξάνεται η αντίστασή του μειώνεται και αντίστροφα. Εκτός από το LDR, υπάρχει η φωτοδίοδος και το φωτοτρανζίστορ. Όλα αυτά είναι αναλογικοί αισθητήρες. Υπάρχουν επίσης ψηφιακοί αισθητήρες φωτός όπως ο BH1750 και ο TSL2561, οι οποίοι μπορούν να υπολογίσουν την ένταση του φωτός και να παρέχουν μια ισοδύναμη ψηφιακή τιμή.

Αισθητήρες αφής: Αισθητήρες αφής συναντάμε στα κινητά, στα tablet, στους φορητούς υπολογιστές κτλ. Οι αισθητήρες αφής ανιχνεύουν το άγγιγμα ενός δακτύλου ή μιας γραφίδας. Ταξινομούνται σε αντιστασιακούς και χωρητικούς. Οι σύγχρονοι αισθητήρες αφής είναι χωρητικού τύπου ,καθώς είναι πιο ακριβείς και έχουν καλύτερη αναλογία σήματος προς θόρυβο.

Αισθητήρας υγρασίας: Η μέτρηση της υγρασίας είναι πολύ σημαντική σε ορισμένες εφαρμογές και οι αισθητήρες υγρασίας βοηθούν προς το συγκεκριμένο σκοπό. Οι περισσότεροι μετρούν τη σχετική υγρασία (αναλογία της περιεκτικότητας σε νερό στον αέρα προς τη μέγιστη δυνατότητα του αέρα να συγκρατεί νερό). Δεδομένου ότι η σχετική υγρασία εξαρτάται από τη θερμοκρασία του αέρα σχεδόν όλοι οι αισθητήρες υγρασίας μπορούν επίσης να μετρήσουν θερμοκρασία. Οι αισθητήρες υγρασίας ταξινομούνται σε χωρητικούς, αντιστασιακούς και θερμό-αγώγιμους. Οι DHT11(αντιστασιακός) και DHT22(χωρητικός) είναι δύο από τους συχνά χρησιμοποιούμενους αισθητήρες υγρασίας.

Αισθητήρας κλίσης: Χρησιμοποιείται για την ανίχνευση κλίσης ή προσανατολισμού. Είναι από τους απλούστερους και φθηνότερους αισθητήρες. Παλαιότερα αποτελούνταν από υδράργυρο και ως εκ τούτου, πολλές φορές αποκαλείται ως διακόπτης υδραργύρου.

2.7 Το Αισθητήριο ως Μετατροπέας

Το αισθητήριο είναι μία διάταξη που μετατρέπει μια μορφή ενέργειας σε ηλεκτρική. Ο μετατροπέας (transducer) είναι συσκευή που μετατρέπει ισχύ από ένα σύστημα σε ένα άλλο στην ίδια ή διαφορετική μορφή. Ο όρος “μετατροπέας” είναι γενικότερος του αισθητηρίου και περιλαμβάνει και μηχανισμούς κίνησης (actuators) εκτός των αισθητηρίων, τα οποία μετατρέπουν μία μορφή ενέργειας σε ηλεκτρική ενέργεια.

Βασική διαφορά ανάμεσα σε αισθητήριο και μετατροπέα είναι ότι ο όρος “αισθητήριο” χρησιμοποιείται για το αντιλαμβανόμενο καθεαυτό στοιχείο, ενώ ο όρος “μετατροπέας” χρησιμοποιείται ως διάταξη αίσθησης μαζί με ένα σχετικό συνδεδεμένο με αυτήν κύκλωμα. Για παράδειγμα, ένα θερμίστορ αποτελεί αισθητήριο, ενώ ένα θερμίστορ μαζί με μία γέφυρα προκειμένου να μετατραπεί η μεταβολή της ηλεκτρικής αντίστασης σε μεταβολή της ηλεκτρικής τάσης, ονομάζεται “μετατροπέας”. Συνεπώς, συμπεραίνουμε ότι κάποιοι μετατροπέες εμπεριέχουν ένα αισθητήριο, ενώ όλα τα αισθητήρια είναι ταυτόχρονα και μετατροπέες (Γαστεράτος και συν., 2013).

Κεφάλαιο 3: Μικροελεγκτές- Μικροεπεξεργαστές

Στις συσκευές που χρησιμοποιούμε στη καθημερινή μας ζωή, καθώς και στα σύγχρονα τεχνολογικά ανεπτυγμένα αυτόνομα οχήματα, συναντάμε τους λεγόμενους μικροελεγκτές. Ειδικότερα, στο τομέα της αυτοκινητοβιομηχανίας, παράγοντες όπως η αύξηση της τεχνολογικής προόδου, η αύξηση της παραγωγής ηλεκτρικών οχημάτων, η εγκατάσταση πρακτικών εξοικονόμησης ενέργειας, δίνουν ώθηση στην ανάπτυξη της αγοράς μικροελεγκτών αυτοκινήτων. Μάλιστα, σύμφωνα με έρευνα

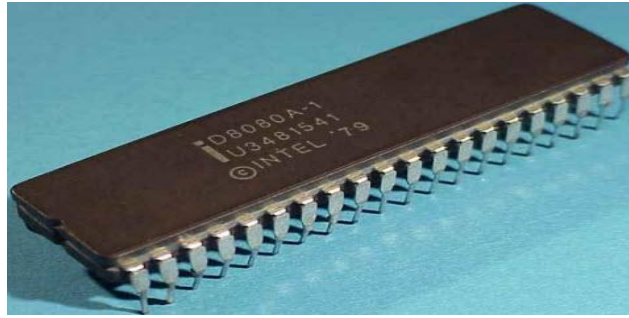
που πραγματοποιήθηκε στη Νέα Υόρκη, η συγκεκριμένη αγορά αναμένεται να φθάσει τα 2,6 εκατομμύρια δολάρια έως το 2023 (Energias Market Research, 2018).

Στα σημερινά αυτόνομα οχήματα, μικροελεγκτές συναντώνται σε συστήματα εκκίνησης, υποβοήθησης στάθμευσης, ελέγχου πέδησης, στους αερόσακους, σε συστήματα παρακολούθησης πίεσης ελαστικών, σε συστήματα ψυχαγωγίας κτλ. Μεταξύ των συνηθισμένων μικροελεγκτών που χρησιμοποιούνται, είναι ο μικροελεγκτής AVR, ο 8051 και ο μικροελεγκτής PIC.

3.1 Ιστορική Αναδρομή

Μέχρι τα μέσα της δεκαετίας του 1980, οι περισσότερες ηλεκτρονικές συσκευές αποτελούνταν από περίπλοκα και συνδυαστικά κυκλώματα, που περιλάμβαναν πολλά τσιπ. Έτσι, σταδιακά οι κατασκευαστές ενσωμάτωσαν στα προϊόντα τους, τους μικροεπεξεργαστές, με σκοπό τη μείωση του κόστους κατασκευής και των τιμών τελικού προϊόντος. Οι πρώτοι μικροεπεξεργαστές 8-bit (οκτώ ψηφίων) που εμφανίστηκαν, ήταν ο Intel 8080 και ο Zilog Z80.

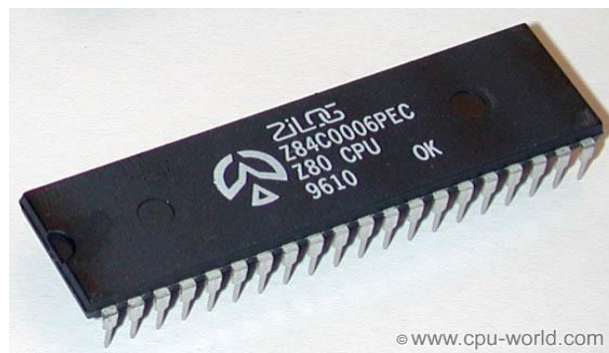
Όμως, οι μικροεπεξεργαστές, δεν ήταν εκείνοι που μείωσαν το κόστος και τη πολυπλοκότητα του σχεδιασμού του προϊόντος. Αυτό συνέβαινε, διότι, ένας μικροεπεξεργαστής έπρεπε να περιβάλλεται από μεγάλο αριθμό πρόσθετων τσιπ για είσοδο-έξοδο και άλλα τσιπ υποστήριξης, όπως για παράδειγμα, τσιπ ρολογιού πραγματικού χρόνου.



Εικόνα 3.1.1: Intel 8080 του 1974, Ανακτήθηκε από:

<https://www.elprocus.com/know-about-architecture-of-the-intel-8080-microprocessor/>

Στις αρχές της δεκαετίας του 1990, οι βελτιωμένες τεχνικές επεξεργασίας πυριτίου και κατασκευής τσιπ, οδήγησαν στο να τοποθετούνται όλο και περισσότερα κυκλώματα σε ένα τσιπ. Κατά συνέπεια, το τσιπ του μικροεπεξεργαστή μπορούσε να εκτελεί επιπλέον λειτουργίες και να έχει περισσότερες δυνατότητες.



Εικόνα 3.1.2: Zilog Z80, Ανακτήθηκε από:

<https://www.cpu-world.com/CPUs/Z80/MANUF-Zilog.html>

Αυτά τα νέα «σούπερ» τσιπ ονομάστηκαν μικροελεγκτές (Trevenor, 2012).

3.2 Σύγκριση Μικροελεγκτών-Μικροεπεξεργαστών

Ο μικροεπεξεργαστής αποτελεί τη καρδιά ενός κλασσικού υπολογιστή, με βασική λειτουργία την εκτέλεση εντολών. Συνήθως, απαιτεί μεγάλο αριθμό υποσυστημάτων και περιφερειακών, όπως το πληκτρολόγιο, την οθόνη και τη μητρική πλακέτα, για να μπορεί να αξιοποιηθεί από τον χρήστη. Συνεπώς, ένας μικροεπεξεργαστής δεν λειτουργεί αυτόνομα.

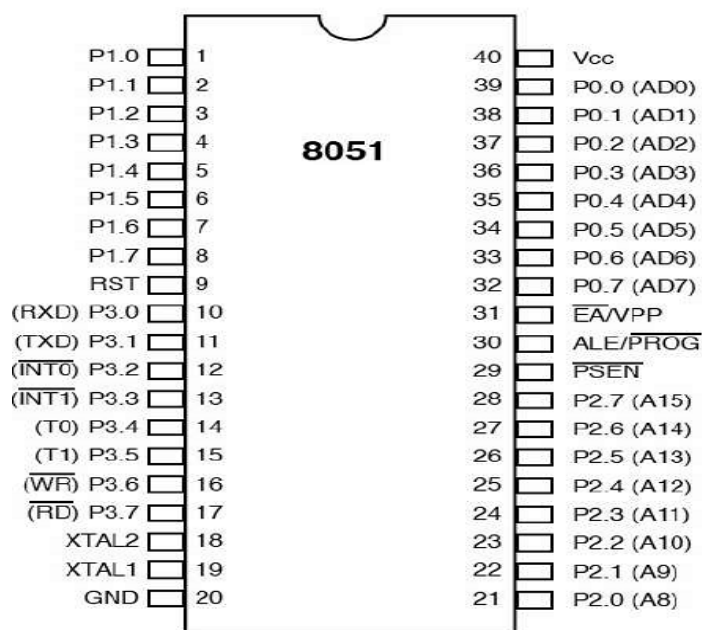
Από την άλλη μεριά, ένας μικροελεγκτής αποτελεί έναν μικρό υπολογιστή, αφού διαθέτει εσωτερική μνήμη για τα προγράμματα, εισόδους-εξόδους για ανταλλαγή σημάτων κτλ. Αντίθετα από τον μικροεπεξεργαστή, ο μικροελεγκτής μπορεί να αλληλεπιδρά με το φυσικό κόσμο παίρνοντας πληροφορίες μέσα από αισθητήρες και διακόπτες καθώς και να ελέγχει φυσικά αντικείμενα, όπως led, μοτέρ κτλ (Παπάζογλου & Λιωνής, 2018).

3.3 Ο Μικροελεγκτής 8051

Ο 8051 είναι ανεπτυγμένος σε ένα chip των 40 ακροδεκτών. Αποτελείται από μονάδα επεξεργασίας δεδομένων, μονάδα ελέγχου, εσωτερική μνήμη RAM και εσωτερική ROM. Έχει επίσης τη δυνατότητα σύνδεσης και με εξωτερικές μνήμες ROM και RAM, εφόσον δεν επαρκούν οι εσωτερικές. Τόσο για την εσωτερική και την εξωτερική ROM, όσο και για την εξωτερική μνήμη RAM ,ο μικροελεγκτής 8051 χρησιμοποιεί 16-bit διευθύνσεις. Επειδή με 16 bit γράφουμε 2^{16} δυαδικούς αριθμούς, με διευθύνσεις των 16 bit, ο 8051 μπορεί να προσπελάσει μνήμες μεγέθους μέχρι 2^{16} =64KB (kilobytes). Η εσωτερική στον 8051 μνήμη ROM έχει μέγεθος 4k. Τα υπόλοιπα 60k μπορούν να καταληφθούν από εξωτερικό chip μνήμης ROM. Η

εσωτερική στον 8051 μνήμη RAM έχει διευθύνσεις των 8 bit και έχει μέγεθος 128 θέσεις.

Η σύνδεση του με εξωτερικές συσκευές γίνεται μέσω τεσσάρων παράλληλων πορτών των 8 bit (P0-P3). Επίσης, ο 8051 περιλαμβάνει δύο χρονιστές των 16 bit (T0 και T1), σειριακό δέκτη/πομπό, απαριθμητή προγράμματος (PC) και δείκτη δεδομένων (DPTR) των 16 bit. Ακόμη, διαθέτει Program Status Word (PSW) των 8 bit, δείκτη σωρού (SP) των 8 bit, καταχωρητές ελέγχου (TCON, TMOD, SCON, PCON, IP και IE), δύο εξωτερικές και τρεις εσωτερικές πηγές διακοπών- interrupt και κυκλώματα ταλαντωτή και χρονιστή.



Εικόνα 3.3.1: Ο 8051, Ανακτήθηκε από:

https://www.tutorialspoint.com/microprocessor/microcontrollers_8051_pin_description.htm

Η ενασχόληση με τον 8051 ως αρχικό μικροελεγκτή, καθιστά εύκολη την εκμάθηση στη συνέχεια οποιουδήποτε άλλου μικροελεγκτή- μικροεπεξεργαστή. Αν και

δύσκολα συναντάται στην αγορά, αποτελεί τον πυρήνα για πολλούς από τους ευρέως χρησιμοποιούμενους σήμερα μικροελεγκτές (Καραίσκος, 2010).

3.4 Η Τεχνολογία Reduced Instruction Set Computers (RISC)

Στους υπολογιστές μειωμένου συνόλου εντολών, Reduced Instruction Set Computers (RISC), ο επεξεργαστής έχει τη δυνατότητα να εκτελεί μία τουλάχιστον εντολή ανά κύκλο ρολογιού. Η τεχνολογία αυτή, αντιτίθεται στους υπολογιστές τεχνολογίας Complex Instruction Set Computers (CISC), που χρειάζονται πολλούς κύκλους για την εκτέλεση μιας εντολής. Οι επεξεργαστές RISC έχουν επίσης αυξημένη κατά πολύ την απόδοσή τους σε σχέση με τους επεξεργαστές CISC. Τέλος, δύο επιπλέον χαρακτηριστικά τους είναι οι πολλοί καταχωρητές που διαθέτουν και η γρήγορη προσπέλαση στη συστοιχία καταχωρητών (Πογαρίδης, 2020).

3.5 Οι Μικροελεγκτές AVR-O ATmega8515

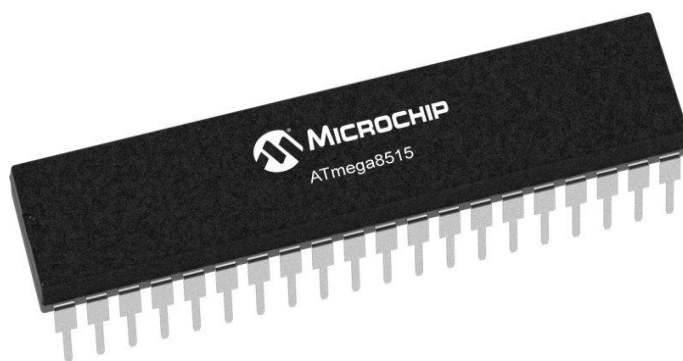
Με χρήση της τεχνολογίας RISC, η εταιρεία ATMEL κατασκεύασε μια οικογένεια μικροελεγκτών, τους AVR. Αυτοί παρέχουν τη δυνατότητα εκτέλεσης σύνθετων εντολών σε ένα κύκλο ρολογιού. Κάθε επεξεργαστής πρέπει να εγγυάται την ορθή εκτέλεση του προγράμματος, προσπελαύνοντας μνήμες, κάνοντας υπολογισμούς, ελέγχοντας τις περιφερειακές συσκευές και χειρίζοντας διακοπές (εσωτερικές ή εξωτερικές).

Οι μικροελεγκτές AVR, χρησιμοποιούν αρχιτεκτονική RISC, παραπλήσια της αρχιτεκτονικής Harvard, με ξεχωριστές μνήμες και διαδρόμους για το πρόγραμμα και τα δεδομένα. Οι εντολές του προγράμματος εκτελούνται ως εξής: μία εντολή

εκτελείται και η επόμενη προ-ανακαλείται από τη μνήμη προγράμματος. Η μνήμη προγράμματος που προγραμματίζεται μέσα στο σύστημα είναι τύπου flash. Η αρχιτεκτονική των AVR επιτρέπει την εκτέλεση κώδικα έως 10 φορές ταχύτερα από τους επεξεργαστές CISC. Ο ATmega8515 είναι ένας 8-bit μικροελεγκτής κατασκευασμένος σύμφωνα με τη τεχνολογία CMOS και την αρχιτεκτονική AVR RISC. Ο συνδυασμός του 8-bit επεξεργαστή του και της ISP flash μνήμης του σε ένα ολοκληρωμένο κύκλωμα, καθιστούν αυτόν μια φθηνή λύση στις εφαρμογές ενσωματωμένων συστημάτων (Πογαρίδης, 2020).

3.6 Η Αρχιτεκτονική του ATmega8515

Ο επεξεργαστής του ATmega8515 αποτελείται από αριθμητική- λογική μονάδα συνδεδεμένη με 32 ενεργούς καταχωρητές γενικού σκοπού, καταχωρητή κατάστασης, καταχωρητή εντολών και αποκωδικοποιητή εντολών. Επίσης, περιλαμβάνει μετρητή προγράμματος (PC), δείκτη σωρού (SP) και μονάδα ελέγχου.



Εικόνα 3.6.1: Ο ATmega8515, Ανακτήθηκε από:

<https://www.microchip.com/en-us/product/ATmega8515>

Το υποσύστημα μνήμης του ATmega8515 αποτελείται από 8KB In System Programmable Flash Memory (ISP flash) για ανάγνωση αλλά και ταυτόχρονα εγγραφή σε αυτήν, 512B (byte) στατικής RAM (Static RAM) και 512B μνήμη EEPROM. Ακόμη διαθέτει 64 καταχωρητές εισόδου-εξόδου (I/O) και 32 καταχωρητές γενικού σκοπού.

Σχετικά με την επικοινωνία του με τον έξω κόσμο, ο ATmega8515 διαθέτει υποσύστημα σειριακής επικοινωνίας διαμέσου ενός σειριακού προγραμματιζόμενου USART και μιας σειριακής διασύνδεσης περιφερειακών. Επιπλέον, περιλαμβάνει υποσύστημα παράλληλης επικοινωνίας διαμέσου 5 πορτών εισόδου-εξόδου πολλαπλών λειτουργιών (Πογαρίδης, 2020).

Τέλος, ο ATmega8515 αποτελείται από υποσύστημα χρονιστών-μετρητών (2 ευέλικτοι χρονιστές-μετρητές), έναν προγραμματιζόμενο επιτηρητή-χρονιστή οδηγούμενο από ταλαντωτή και έναν αναλογικό συγκριτή. Επίσης, έχει τη δυνατότητα εσωτερικών και εξωτερικών διακοπών (Interrupt).

Κεφάλαιο 4: Το Arduino

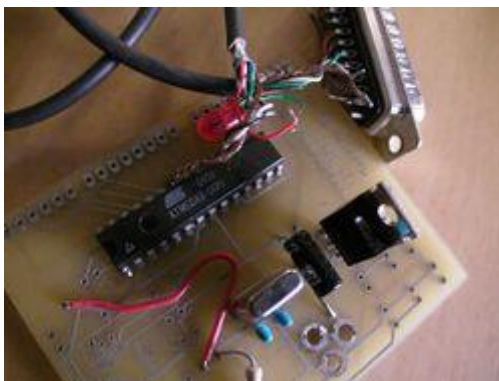
4.1 Η Αναπτυξιακή Πλατφόρμα Arduino

Το Arduino αποτελεί μια μητρική πλατφόρμα ανάπτυξης έργων ανοιχτού κώδικα. Αυτή περιλαμβάνει τη πλακέτα του μικροελεγκτή AVR ATmega328 μαζί με το περιβάλλον ανάπτυξης του κώδικα προγράμματος για τον προγραμματισμό της πλακέτας. Η ανάπτυξη κώδικα γίνεται σε γλώσσα Wiring C, η οποία ακολουθεί τους συντακτικούς κανόνες της γλώσσας C. Ωστόσο, παρέχει πολλές αυτοματοποιημένες

διαδικασίες ακόμη και για τα άτομα που δεν έχουν προηγούμενη εμπειρία με το αντικείμενο (Καλοφωλιάς, 2017).

4.2 Σύντομη Ιστορική Αναδρομή

Η ιδέα για το Arduino προήλθε από το Wiring project που περιλάμβανε την ανάπτυξη ενός συστήματος βασισμένο σε μικροελεγκτή, με σκοπό την εύκολη, γρήγορη και με χαμηλό κόστος ανάπτυξη εφαρμογών. Το συγκεκριμένο project αναπτύχθηκε από τον φοιτητή Hernando Barragan, το 2003 στην Ιβρέα της Ιταλίας. Η πλατφόρμα καλωδίωσης αποτελούνταν από μία πλακέτα τυπωμένου κυκλώματος (PCB) με έναν μικροελεγκτή ATmega128. Το Arduino αποτέλεσε προέκταση του Wiring Project και προοριζόταν για χρήση από άτομα μη εξοικειωμένα με την τεχνολογία. Η ονομασία του προήλθε από τον στρατιωτικό διοικητή της Ιβρέας και αργότερα βασιλιά της Ιταλίας Arduin (Παπάζογλου & Λιωνής, 2018).



Εικόνα 4.2.1 : Το πρώτο Arduino, Ανακτήθηκε από:

<https://handwiki.org/wiki/index.php?curid=1204364>

4.3 Τα Χαρακτηριστικά της Πλατφόρμας Arduino

Η πλατφόρμα Arduino αποτελεί τη δημοφιλέστερη τεχνολογία ανάπτυξης εφαρμογών στο κόσμο. Αυτό συμβαίνει διότι προσφέρει γρήγορη ανάπτυξη με κώδικα υψηλού επιπέδου, σε χαμηλό κόστος και ανοιχτή αρχιτεκτονική. Κάποιος, μπορεί να αγοράσει το Arduino Uno R3, με κόστος μικρότερο από 12 ευρώ. Το περιβάλλον ανάπτυξης του Arduino είναι ανοιχτού κώδικα και διατίθεται δωρεάν από το επίσημο site του Arduino (<https://www.arduino.cc/>). Γι αυτό, και αποτελεί την ιδανική λύση για τα άτομα που αποφασίζουν να ασχοληθούν με την ανάπτυξη εφαρμογών, καθώς μπορούν να βρουν τμήματα πηγαίου κώδικα, για να τα μελετήσουν και να τα τροποποιήσουν. Επίσης, η προσθήκη βιβλιοθηκών C++ προσθέτει δυνατότητες στο περιβάλλον ανάπτυξης. Ο προγραμματισμός σε γλώσσα παραπλήσιας της C, Wiring προσφέρει την ανάπτυξη εφαρμογών που εκτείνονται από τη διαχείριση ενός LED μέχρι την ανάπτυξη συστημάτων ρομποτικής τηλεμετρίας και αυτοματισμού. Τέλος, η πλατφόρμα Arduino χαρακτηρίζεται για τη συμβατότητά της, διότι το λογισμικό της εκτελείται σε περιβάλλοντα Windows, Macintosh και Linux (Παπάζογλου & Λιωνής, 2018 · Πογαρίδης, 2020) .

4.4 Η Αρχιτεκτονική του Arduino

Η πλατφόρμα Arduino περιλαμβάνει τον μικροελεγκτή ATmega328. Ο μικροελεγκτής ATmega328 έχει την ίδια αρχιτεκτονική με τον ATmega8515 και επιπλέον διαθέτει (Πογαρίδης, 2020):

- 32 KB μνήμη flash
- 2 KB μνήμη static RAM

- 1 KB μνήμη EEPROM
- Τρεις χρονιστές
- Μετρητή πραγματικού χρόνου με ξεχωριστό ταλαντωτή
- Έξι κανάλια μετατροπής 10 ψηφίων από αναλογικό σε ψηφιακό
- Έξι κανάλια PWM
- Είκοσι τρεις προγραμματιζόμενες γραμμές I/O

4.5 Τα βασικά στοιχεία της Πλακέτας Arduino

Μικροελεγκτής

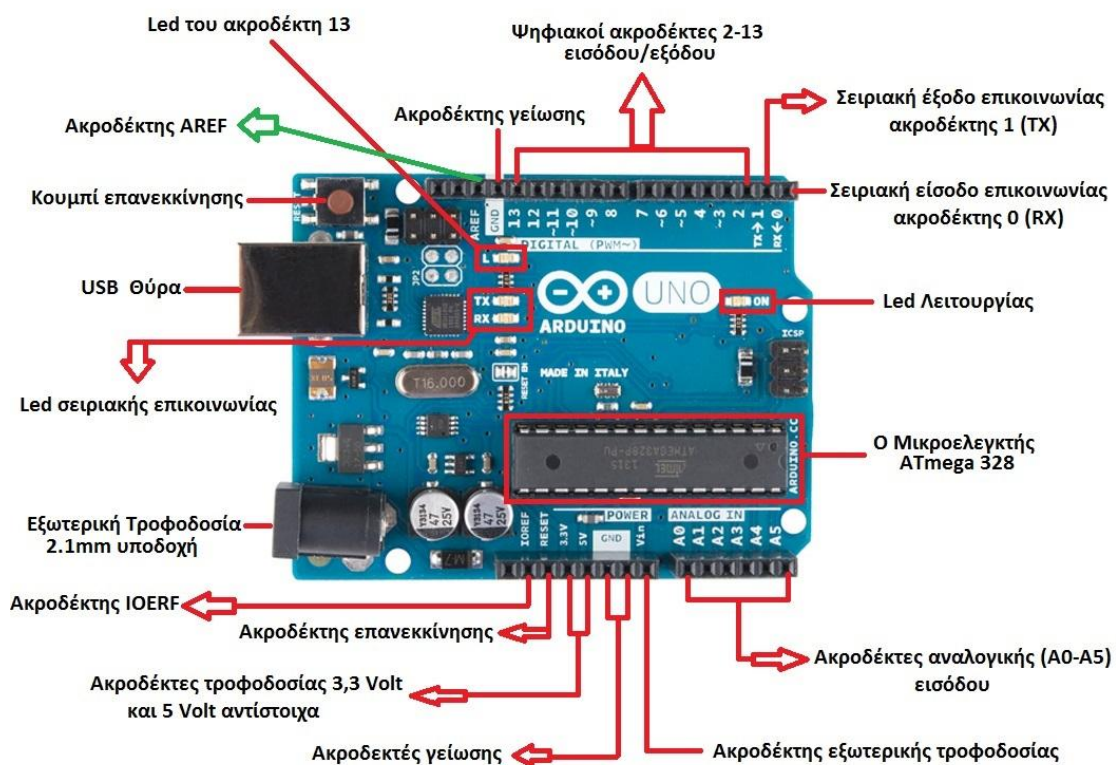
Η ‘οικογένεια’ Arduino περιλαμβάνει πολλές εκδόσεις και παραλλαγές. Κάθε board (πλακέτα) διαθέτει συγκεκριμένο τύπο μικροελεγκτή. Ο κύριος μικροελεγκτής είναι ο ATmega328 ο οποίος, προγραμματίζεται μέσω του λογισμικού Arduino IDE, για το οποίο θα γίνει εκτενέστερη αναφορά στην ενότητα 4.6 της εργασίας. Ο AVR ATmega328 είναι ένας 8-bit ελεγκτής με τάση λειτουργίας τα 5V(Καλοφωλιάς, 2017).

Ακροδέκτες

Κάθε board (πλακέτα) Arduino υποστηρίζει συγκεκριμένο τύπο μικροελεγκτή και διαθέτει ακροδέκτες που χωρίζονται σε ακροδέκτες εισόδου-εξόδου και ακροδέκτες τροφοδοσίας. Το Arduino μπορεί να διαβάσει αναλογικά σήματα 0-5V μέσω των pin A0-A5. Επίσης, μπορεί να διαβάσει ψηφιακά σήματα 0 ή 5V. Αυτό γίνεται μέσω των pin 0-13 του μικροελεγκτή (Καλοφωλιάς, 2017· Παπάζογλου & Λιωνής, 2018).

Το Arduino μπορεί να προσφέρει τροφοδοσία σε εξωτερικές συσκευές, όπως αισθητήρες και ολοκληρωμένα κυκλώματα με τους αντίστοιχους ακροδέκτες 3,3 V

και 5V. Επιπλέον, μπορεί να δεχθεί τροφοδοσία απευθείας με 5V (ακροδέκτης Vin), όπως ακόμη να δεχθεί εξωτερική τάση αναφοράς <5V (ακροδέκτης AREF) για αύξηση της ακρίβειας της μέτρησης, κατά την εισαγωγή αναλογικού σήματος 0-5V. Τέλος, το Arduino διαθέτει τρεις συνήθως ακροδέκτες γείωσης GND (μηδενικής τάσης) (Καλοφωλιάς, 2017 · Παπάζογλου & Λιωνής, 2018).



Εικόνα 4.5.1: Τα βασικά στοιχεία του Arduino, Ανακτήθηκε από:

<https://projectmaniacs.wordpress.com/2014/11/29>

Θύρες τροφοδοσίας και USB

Στη θύρα τροφοδοσίας του Arduino μπορεί να συνδεθεί με καλώδιο power jack μπαταρία 9V ή ένα τροφοδοτικό τάσης έως +12V. Όπως έχει αναφερθεί, η τάση

λειτουργίας της πλακέτας είναι 5V. Αυτό επιτυγχάνεται μέσω κατάλληλου κυκλώματος που μειώνει την τάση τροφοδοσίας.

Η θύρα USB του Arduino συνδέεται μέσω καλωδίου με τη θύρα USB του υπολογιστή, για τη μεταφορά και τη λήψη δεδομένων και για το ανέβασμα του κώδικα προγράμματος από το Arduino IDE στον μικροελεγκτή. Ταυτόχρονα λειτουργεί και ως τροφοδοσία. Άπαξ και φορτωθεί ο κώδικας προγράμματος στον μικροελεγκτή, η θύρα USB λειτουργεί ως τροφοδοσία με τη χρήση ενός power bank (Καλοφωλιάς, 2017).

LEDS

Στην πλακέτα Arduino συναντάμε το LED λειτουργίας της πλακέτας που παραμένει αναμμένο κατά τη διάρκεια της τροφοδοσίας της, το LED του ακροδέκτη 13 του Arduino που ανάβει κατά την ενεργοποίησή του ακροδέκτη και τα LED των ακροδεκτών TX και RX που αναβοσβήνουν, κατά τη σειριακή επικοινωνία του Arduino με τον 'έξω κόσμο'(Καλοφωλιάς, 2017).

Κουμπί Reset

Με το πάτημα του κουμπιού reset της πλακέτας Arduino, επανεκτελείται ο κώδικας προγράμματος από την αρχή.

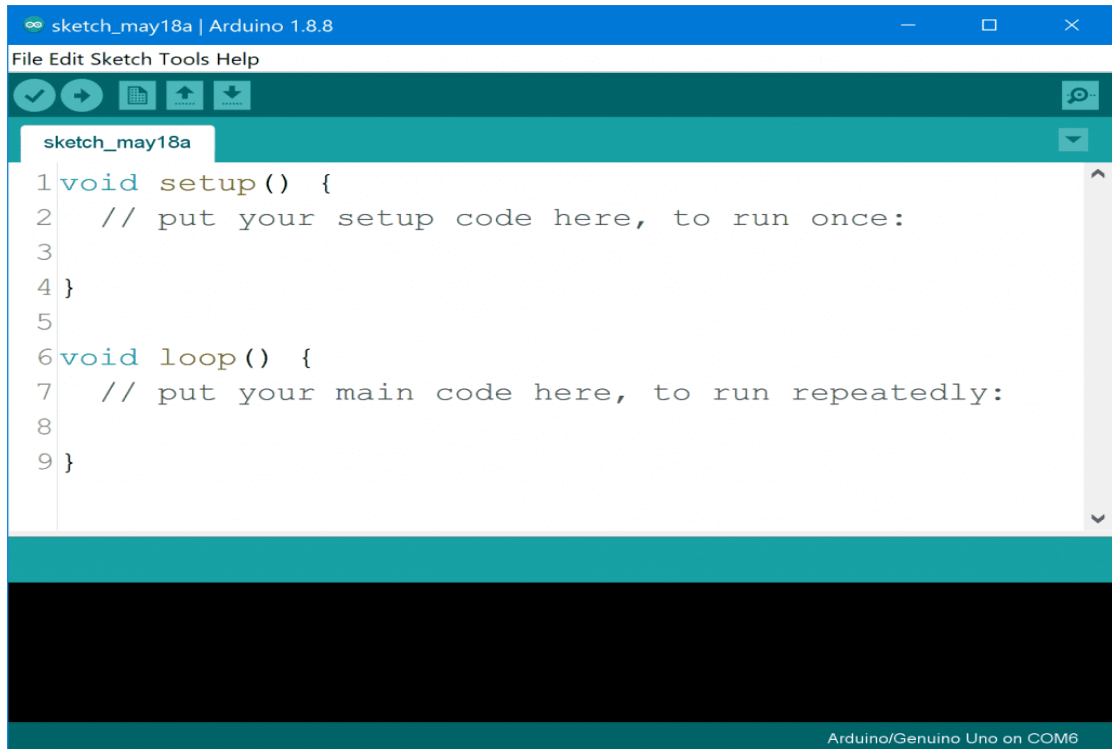
4.6 Το λογισμικό του Arduino

Το ολοκληρωμένο περιβάλλον ανάπτυξης Integrated Development Environment (IDE) του Arduino αποτελεί μία εφαρμογή που λειτουργεί σε πλατφόρμες Windows, Linux, Os. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα. Το πρόγραμμα, αφού

γραφεί από τον χρήστη αποσφαλματοποιείται, γίνεται εκτελέσιμο και μεταφορτώνεται στον ελεγκτή. Το λογισμικό του Arduino είναι ελεύθερο και μπορεί να βρεθεί στην επίσημη ιστοσελίδα του Arduino (<https://www.arduino.cc/>).

Μετά την εκκίνηση του IDE του Arduino, συνδέουμε την πλακέτα με τον υπολογιστή μέσω καλωδίου USB και εν συνέχεια γράφουμε τον κώδικα που επιθυμούμε. Κάθε πρόγραμμα που αναπτύσσεται σε Wiring C αποτελείται από δύο συναρτήσεις. Η μία είναι η συνάρτηση **setup** () που εκτελείται μία φορά μετά την εφαρμογή της τροφοδοσίας ή μετά την ενεργοποίηση του κουμπιού reset. Η δεύτερη είναι η συνάρτηση **loop** () που εκτελείται, όσο το Arduino βρίσκεται σε κατάσταση λειτουργίας (Πογαρίδης, 2020).

Πριν μεταφορτώσουμε τον κώδικα στο Arduino, απαραίτητη προϋπόθεση είναι να επιλέξουμε την πλακέτα που χρησιμοποιούμε μέσω της επιλογής Tools → Board → Arduino Uno και τη θύρα επικοινωνίας μέσω του Tools → Serial Port. Ανάλογα με το λογισμικό που χρησιμοποιούμε, έχουμε επιλογές τύπου COM (Arduino/ Genuino Uno) σε Windows ή tty/ACM (Arduino/ Genuino Uno) σε Linux. Στη συνέχεια, ελέγχουμε το πρόγραμμα για τυχόν συντακτικά λάθη και μεταφορτώνουμε τον κώδικα (Καλοφωλιάς, 2017).



Εικόνα 4.6.1: Το Arduino IDE, Ανακτήθηκε από: <https://www.circuitbasics.com/how-to-install-and-configure-the-arduino-ide/>

ΜΕΡΟΣ Β ' ΕΙΔΙΚΟ ΜΕΡΟΣ

Εισαγωγή

Στόχος της συγκεκριμένης διπλωματικής εργασίας ήταν η μελέτη της κίνησης που μπορεί να εκτελέσει ένα αυτόνομο όχημα, όταν βρεθεί σε ένα συγκεκριμένο περιβάλλον εμποδίων. Έχοντας κατανοήσει πλήρως τη φιλοσοφία του αυτόνομου οχήματος και τη τεχνολογία με την οποία σχεδιάζεται και κατασκευάζεται, επιλέχθηκε η μελέτη, ο σχεδιασμός, η κατασκευή και ο προγραμματισμός της κίνησης ενός ρομποτικού αυτοκινούμενου οχήματος. Πρόκειται για ένα έντροχο όχημα με δυνατότητα λειτουργίας κυρίως σε εσωτερικούς χώρους. Ο μικροελεγκτής που χρησιμοποιήθηκε για την κατασκευή του συγκεκριμένου οχήματος, ήταν ο AVR ATmega328, ο οποίος είναι ενσωματωμένος στην πλατφόρμα Arduino Uno. Ο προγραμματισμός του μικροελεγκτή έγινε μέσω του ολοκληρωμένου περιβάλλοντος ανάπτυξης, Integrated Development Environment(IDE) του Arduino.

Το ρομποτικό όχημα είναι αυτόνομο, δηλαδή αποφασίζει μόνο του πως θα κινηθεί, βάσει του προγράμματος που περιλαμβάνεται στο σύστημα ελέγχου του, χωρίς να απαιτείται η παρέμβαση κάποιου χρήστη. Επίσης, είναι ένα όχημα που υπάρχει στο φυσικό κόσμο και δεν αποτελεί εξομοίωση κάποιου συστήματος σε πρόγραμμα. Μέσω του αισθητήρα που διαθέτει αφουγκράζεται το περιβάλλον, ανιχνεύει δηλαδή τα εμπόδια και δρα ρυθμίζοντας την κίνησή του, με στόχο την αποφυγή εμποδίων.

Κεφάλαιο 5: Σχεδιασμός Κατασκευής - Λειτουργικά μέρη

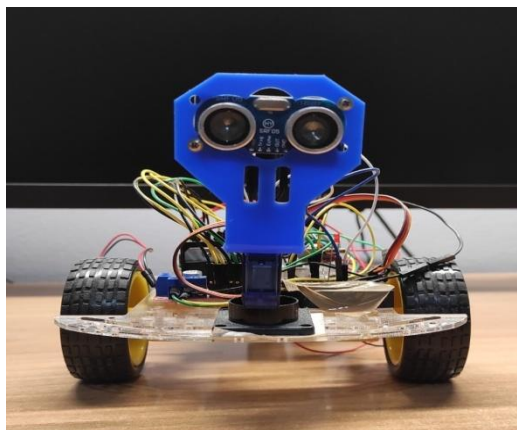
5.1 Περιγραφή της κατασκευής του οχήματος

Για την κατασκευή του ρομποτικού οχήματος της εργασίας είναι απαραίτητο αρχικά να καθοριστούν τα υλικά κατασκευής και οι διαστάσεις του. Το συγκεκριμένο όχημα δεν αποτελεί μια έτοιμη για χρήση κατασκευή αλλά υλοποιήθηκε με την συναρμολόγηση των υλικών που απαριθμούνται πιο κάτω:

- Arduino UNO
- BreadBoard
- Robotic Car Kit (πλατφόρμα, δύο μοτέρ με τροχούς, διακόπτης, AA Battery Case)
- 1 x 9V Μπαταρία
- 4 x AA Μπαταρίες
- 1 x LED
- L298N Motor Driver
- HC-SR04 Ultrasonic Distance Sensor
- 1 x 330Ω Resistor
- Micro Servo
- Καλώδια

Η ρομποτική πλατφόρμα που χρησιμοποιήθηκε έχει διαστάσεις 22 x 14,7 cm. Το υλικό κατασκευής της είναι το plexi-glass και μπορεί να μεταφέρει φορτίο έως 1 kg. Στο εμπρόσθιο τμήμα της (**Εικόνα 5.1.1**), είναι προσαρτημένος ο σερβοκινητήρας και πάνω από αυτόν έχει τοποθετηθεί ο αισθητήρας απόστασης HC-SR04.

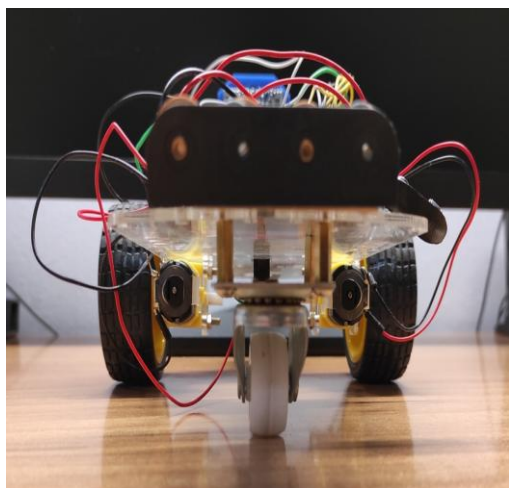
Στο οπίσθιο τμήμα της πλατφόρμας (**Εικόνα 5.1.2**), έχει προσαρτηθεί μια ρόδα



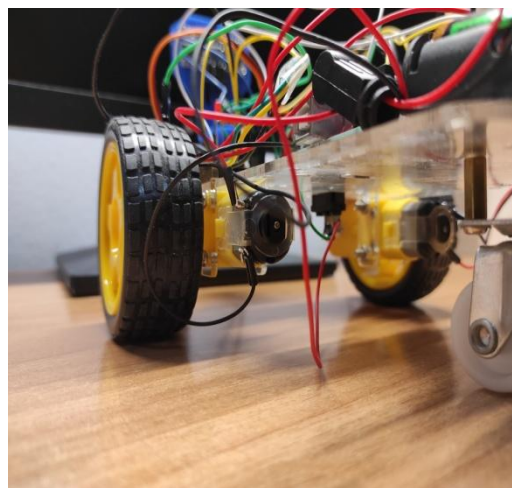
στήριξης ή αλλιώς «καθολικός τροχός» (universal wheel) που υποστηρίζει την κίνηση του οχήματος. Στο κάτω τμήμα της πλατφόρμας (**Εικόνα 5.1.3**), έχουν τοποθετηθεί με τη βοήθεια στηριγμάτων από plexi-glass και με βίδες, δύο μικροί

Εικόνα 5.1.1 Εμπρόσθια όψη οχήματος

κινητήρες (gear motor) συνεχούς ρεύματος (DC) με αποδεκτή τάση λειτουργίας 3-6V και 130 στροφές rpm. Στους DC κινητήρες εφαρμόζουν δύο πλαστικοί τροχοί, με διάμετρο 66 mm και πλάτος 26 mm.



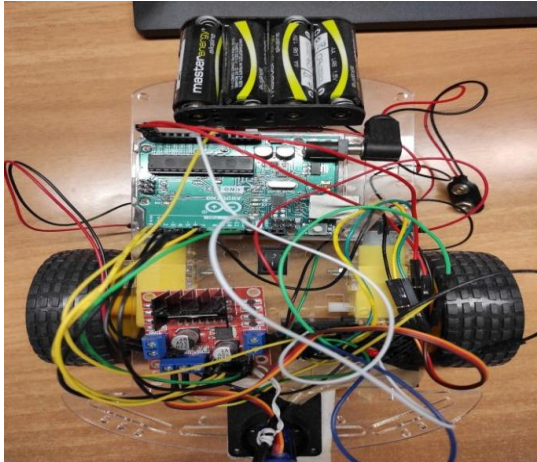
Εικόνα 5.1.2 Οπίσθια όψη οχήματος



Εικόνα 5.1.3 Κάτω όψη οχήματος

Στο επάνω τμήμα της πλατφόρμας (**Εικόνα 5.1.4**), βρίσκονται το Arduino, ο οδηγός L298N (motor driver) των δύο μικρών κινητήρων συνεχούς ρεύματος και μία mini πλακέτα δοκιμών (breadboard) για τη διευκόλυνση των καλωδιώσεων και την τοποθέτηση αντιστάσεων και LED (φωτοεκπεμπουσών διόδων).

Στο πίσω μέρος του έχει προσαρτηθεί η μπαταριοθήκη για τη τροφοδοσία του οχήματος, χωρητικότητας τεσσάρων μπαταριών τύπου AA. Τέλος, στο κέντρο του επάνω τμήματος της πλατφόρμας είναι τοποθετημένος ένας διακόπτης δύο θέσεων on/off (Rocker Switch) για την ενεργοποίηση και την απενεργοποίηση του συνολικού κυκλώματος.

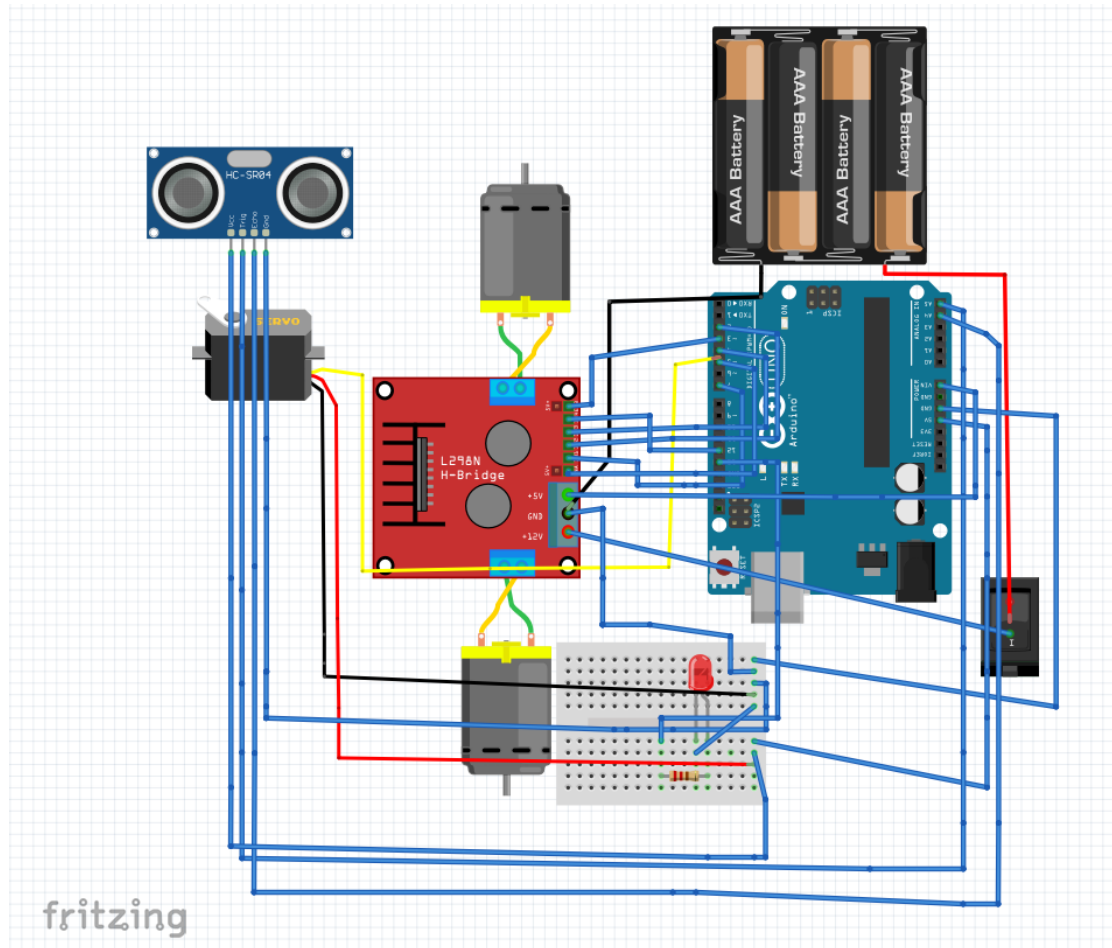


Εικόνα 5.1.4 Επάνω όψη οχήματος

5.2 Σχεδίαση κατασκευής οχήματος στο πρόγραμμα Fritzing

Η συνδεσμολογία των ηλεκτρονικών εξαρτημάτων της κατασκευής, των μηχανολογικών μερών και τμημάτων hardware υλοποιήθηκε παράλληλα και στο σχεδιαστικό πρόγραμμα fritzing. Πρόκειται για ένα λογισμικό ανοιχτού κώδικα, γραμμένο σε γλώσσα προγραμματισμού C++, ειδικά σχεδιασμένο για όσους χρειάζονται να δημιουργήσουν project ηλεκτρονικής. Περιλαμβάνει μια μεγάλη ποικιλία ηλεκτρονικών εξαρτημάτων και υλικών hardware, διαθέσιμη στο χρήστη για σχεδίαση.

Η απεικόνιση της συνδεσμολογίας στο πρόγραμμα fritzing παρατίθεται στην **εικόνα 5.2.1**.

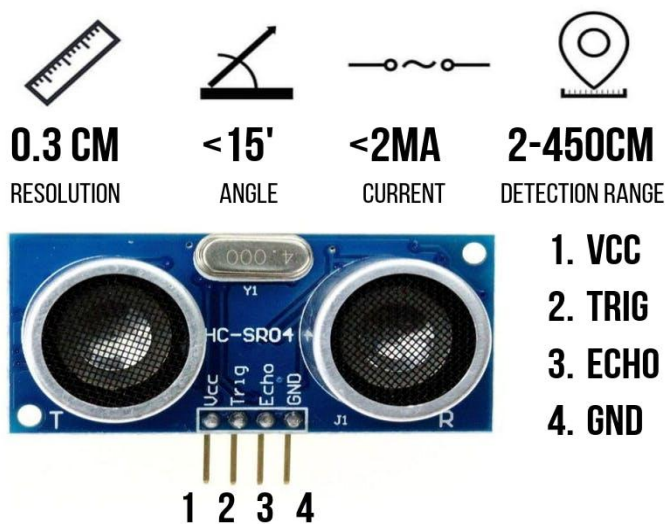


Εικόνα 5.2.1: Η συνδεσμολογία στο πρόγραμμα fritzing, Φωτογραφία προσωπικού αρχείου

5.3 Αισθητήρας Απόστασης HC-SR04

Ο αισθητήρας απόστασης HC-SR04 χρησιμοποιείται για τη μέτρηση απόστασης από κάποιο εμπόδιο. Εκπέμπει στο φάσμα των υπερήχων και ανήκει στους ενεργούς αισθητήρες. Οι υπερήχοι που εκπέμπονται βρίσκονται σε συχνότητες μεγαλύτερες από 20 KHz. Ενδεικτικά, αναφέρεται ότι το ακουστικό φάσμα εκτείνεται περίπου μεταξύ 20 Hz- 20KHz .

Ο HC-SR04 μετράει αποστάσεις από 2cm μέχρι 400 cm, με ακρίβεια 0,3cm. Η τάση τροφοδοσίας του είναι 5 Volt. Διαθέτει επίσης τέσσερις ακροδέκτες. Οι δύο ακραίοι αφορούν την τροφοδοσία (VCC) και τη γείωση (GND). Από τους δύο κεντρικούς ακροδέκτες, ο αριστερός είναι ο ακροδέκτης ενεργοποίησης του αισθητήρα (Trig) και ο δεξιός είναι ο ακροδέκτης εξόδου του (Echo) (Παπάζογλου & Λιωνής, 2018).



Εικόνα 5.3.1: Αισθητήρας Απόστασης HC-SR04, Ανακτήθηκε από:

<https://www.daraz.pk/products/>

5.3.1 Λειτουργία αισθητήρα

Η λειτουργία του αισθητήρα σε επίπεδο σημάτων είναι η εξής: Το Arduino στέλνει αρχικά έναν παλμό ενεργοποίησης του αισθητήρα. Εν συνεχεία, ο αισθητήρας εκπέμπει μια ριπή και δέχεται μια ανάκλαση από το εμπόδιο. Μετά την ανάκλαση, ο αισθητήρας στέλνει έναν παλμό Echo, η διάρκεια του οποίου είναι ανάλογη της απόστασης του εμποδίου η οποία προσδιορίζεται με το χρόνο επιστροφής του σήματος.

Η διαδικασία δημιουργίας του παλμού ενεργοποίησης που στέλνεται από το Arduino στον αισθητήρα περιλαμβάνει τις παρακάτω εντολές:

digitalWrite (trig, LOW);

delayMicroseconds (2) ;

digitalWrite (trig, HIGH);

delayMicroseconds (10) ;

digitalWrite (trig, LOW) ;

delayMicroseconds (2) ;

Με τη συνάρτηση **pulseIn ()** διαβάζεται η έξοδος του HC- SR04. Η συνάρτηση δέχεται ως ορίσματα το pin λήψης και το επίπεδο του παλμού (HIGH ή LOW). Αυτό που κάνει είναι να ανιχνεύει το αντίστοιχο pin μέχρι να λάβει έναν παλμό με το επιθυμητό επίπεδο. Τελικά, επιστρέφει μια ακέραια τιμή του τύπου Long που δείχνει τη διάρκεια του παλμού σε msec. Η διάρκεια αυτή ταυτίζεται με το χρόνο που μεσολαβεί από την αποστολή ακολουθίας υπερήχων μέχρι την ανάκλασή τους (Παπάζογλου & Λιωνής, 2018).

Πρόκειται για τον χρόνο που χρειάζονται οι υπέρηχοι για να διανύσουν δύο φορές την απόσταση ανάμεσα στον αισθητήρα και κοντινότερου αντικειμένου. Λαμβάνοντας υπόψη τη ταχύτητα του ήχου στον αέρα ($340 \text{ m/sec}=0,034\text{cm}/\mu\text{sec}$), από τον τύπο της ταχύτητας ($u=s/t$) παίρνουμε τη σχέση:

$$\text{Distance}=(0,034*\text{duration})/2$$

όπου μας δίνει την απόσταση σε συνδυασμό με τη τιμή που επιστρέφει ο αισθητήρας (διάρκεια αποστολής ακολουθίας υπερήχων και ανάκλασής τους).

5.4 Σερβοκινητήρας (micro servo)

Το micro servo ή σερβοκινητήρας που χρησιμοποιήθηκε είναι μία συσκευή που αποτελείται από ένα μοτέρ DC, ένα σύστημα με γρανάζια, ένα ποτενσιόμετρο, ένα κύκλωμα ελέγχου και έναν άξονα. Η περιστροφή του άξονα του servo επιτυγχάνεται με τη χρήση του ποτενσιόμετρου και του κυκλώματος ελέγχου. Με το σήμα Pulse Width Modulation (PWM) ορίζεται η γωνιακή θέση του άξονα. Το σήμα αυτό παραμένει ενεργό διατηρώντας το servo στην επιθυμητή γωνία, ενώ έχει ολοκληρωθεί η περιστροφή του άξονα.



Τα περισσότερα servo υποστηρίζουν γωνία περιστροφής έως 180° . Ωστόσο, υπάρχουν και servo με γωνία περιστροφής που φθάνει τις 360° και servo συνεχούς περιστροφής,

Εικόνα 5.4.1: Micro Servo, Ανακτήθηκε από:

<https://www.skroutz.gr/s/>

στα οποία το σήμα PWM καθορίζει την ταχύτητα περιστροφής του άξονα.

Τα micro servo μπορούν να τροφοδοτηθούν κατευθείαν από το Arduino. Διαθέτουν τρία καλώδια από τα οποία, το καφέ αφορά τη γείωση, το κόκκινο αφορά την τροφοδοσία του servo με 5 V και το πορτοκαλί αποτελεί είσοδο για το σήμα ελέγχου. Για την αποφυγή δονήσεων ή περιστασιακών κολλημάτων του servo, όπως για μοτέρ με μεγαλύτερες απαιτήσεις ρεύματος, πρέπει να παρέχεται εξωτερική τροφοδοσία. Στην περίπτωση αυτή, το καφέ και το κόκκινο καλώδιο που αναφέρθηκαν παραπάνω πρέπει να συνδεθούν με το τροφοδοτικό, ενώ η γείωση του τροφοδοτικού πρέπει να συνδεθεί και με τη γείωση του Arduino (Καλοφωλιάς, 2017).

5.4.1 Εντολές διαχείρισης servo

Ο έλεγχος των μοτέρ servo γίνεται μέσω της βιβλιοθήκης servo που υπάρχει προεγκατεστημένη στο Arduino IDE. Η βιβλιοθήκη υποστηρίζει την ταυτόχρονη σύνδεση μέχρι και 12 servo. Σε όλες τις πλακέτες, πλην της πλακέτας Arduino Mega, η χρήση της βιβλιοθήκης καθιστά αδύνατη τη χρήση της λειτουργίας PWM στους ακροδέκτες 9 και 10. Στο Arduino Mega όπου υποστηρίζει έως 48 servo, το PWM απενεργοποιείται στους ακροδέκτες 11 και 12, αν χρησιμοποιηθούν περισσότερα από 12 servo (Καλοφωλιάς, 2017).

Εντολή Servo myservo;

Με τη συγκεκριμένη εντολή, η κλάση Servo χρησιμοποιείται για τη δημιουργία του αντικειμένου myservo που θα αντιπροσωπεύει το servo της εργασίας.

Εντολή `myservo.attach` (αριθμός_ pin);

Η εντολή χρησιμοποιείται για τη σύνδεση του αντικειμένου `myservo` με το μοτέρ `servo`. Ο αριθμός_ pin είναι ο αριθμός του ακροδέκτη του Arduino στον οποίο έχει συνδεθεί το καλώδιο ελέγχου του `servo`.

Εντολή `myservo.write` (γωνία);

Με τη μέθοδο `write` () της κλάσης `servo` κάνουμε τον άξονα του `servo` να περιστραφεί σε μια συγκεκριμένη γωνία. Η γωνία αυτή κυμαίνεται από 0° έως 180°.

Εντολή `myservo.detach` ();

Με την συγκεκριμένη εντολή μπορεί να αποσυνδεθεί ένα αντικείμενο από το pin με το οποίο έχει αντιστοιχιστεί. Με την εκτέλεση της εντολής προκαλείται διακοπή του σήματος PWM που στέλνεται στον ακροδέκτη ελέγχου του `servo`. Για την αποφυγή των δονήσεων και των θορύβων στα μοτέρ `servo`, μπορεί να χρησιμοποιηθεί η μέθοδος `detach` (), μετά από κάθε μετακίνηση του `servo`. Φυσικά, πριν από κάθε νέα μετακίνηση, θα πρέπει να εκτελείται εκ νέου η αντίστοιχη `attach` ().

5.5 Κινητήρας DC

Για να ελέγξουμε πλήρως έναν κινητήρα, θα πρέπει να μπορούμε να καθορίσουμε τη φορά περιστροφής του, καθώς και τη ταχύτητα περιστροφής του. Το ρεύμα που απαιτεί ένας κινητήρας είναι ανάλογο του φορτίου του. Συνεπώς, είναι ιδιαίτερα εύκολο να ξεπεράσει τα 40 mA που είναι το μέγιστο ρεύμα με το οποίο μπορεί μία θύρα του ελεγκτή να τροφοδοτήσει μία συσκευή. Επομένως, ένας κινητήρας δεν

επιτρέπεται να τροφοδοτείται άμεσα από τον ελεγκτή, διότι είναι πιθανό να ζητήσει υψηλή τιμή ρεύματος από τον τελευταίο, με συνέπεια να τον καταστρέψει.

Στην προκειμένη περίπτωση επιβάλλεται η χρήση ενός motor driver (οδηγού κινητήρα). Το motor driver είναι ένα ηλεκτρονικό κύκλωμα το οποίο παρέχει στον κινητήρα ξεχωριστή πηγή τροφοδοσίας. Επιπλέον, δίνει τη δυνατότητα στον ελεγκτή να ορίζει την ταχύτητα περιστροφής του κινητήρα, μέσω των ψευδο-αναλογικών εξόδων τύπου Pulse Width Modulation (PWM), αλλά και τη φορά περιστροφής, μέσω της τεχνικής της γέφυρας H (H bridge).

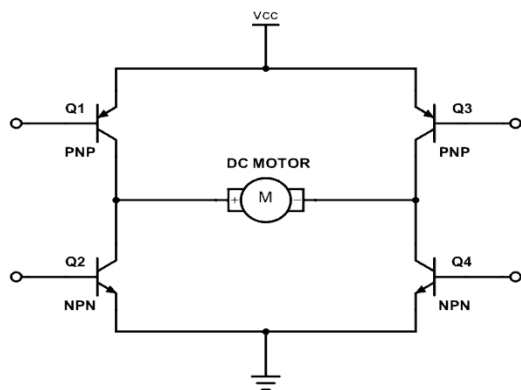


Εικόνα 5.5.1: DC Gear Motor TT 130 RPM (With Wire) and Wheel, Ανακτήθηκε από:

<https://www.joom.com/el/products/5d78931936b54d0101da3a81>

5.5.1 Γέφυρα H

Για τον έλεγχο της φοράς περιστροφής ενός κινητήρα, χρησιμοποιούμε την τεχνική της γέφυρας H, με την οποία αλλάζουμε την κατεύθυνση της ροής του ρεύματος σε έναν κινητήρα.



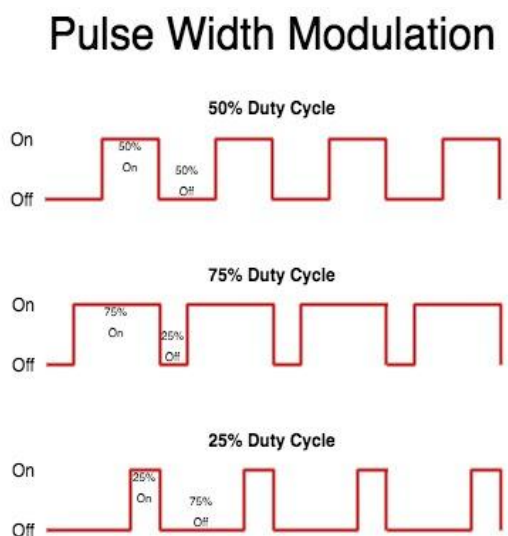
Εικόνα 5.5.1.1: Κύκλωμα Γέφυρας H, Ανακτήθηκε από:

<https://www.build-electronic-circuits.com/h-bridge/>

Η γέφυρα H είναι ένα κύκλωμα που αποτελείται από τέσσερις διακόπτες τρανζίστορ ή MOSFETS με τον κινητήρα στο κέντρο σχηματίζοντας το γράμμα H. Ενδεικτικά, αναφέρεται ότι το L298N περιέχει δύο κυκλώματα γέφυρας H, ένα για κάθε DC κινητήρα.

5.5.2 Pulse Width Modulation (PWM)

Το Arduino Uno δεν διαθέτει αμιγώς αναλογικές εξόδους. Ωστόσο, μπορούμε να παράξουμε ένα ψευδό-αναλογικό σήμα με την τεχνική PWM. Οι ψηφιακές έξοδοι 3, 5, 6, 9, 10, 11 του Arduino συνοδεύονται από το σύμβολο ~ που δηλώνει πως πρόκειται για θύρες PWM.



Εικόνα 5.5.2.1: Η τεχνική PWM, Ανακτήθηκε από:

<https://mynewmicrophone.com/>

Η συχνότητα ενός παλμού PWM είναι 500 Hz και άρα η περίοδος που παράγει μία θύρα είναι $1/500 = 0,002$ sec. Η διάρκεια του παλμού που παράγει μία θύρα PWM κάθε $1/500$ sec ελέγχεται από την εντολή analogWrite (). Ο παλμός PWM έχει ανάλυση 8 bit, άρα μπορεί να αναλυθεί σε $2^8=256$ διαστήματα με ελάχιστο το 0 και μέγιστο το 255.

Η εντολή analogWrite () συντάσσεται ως εξής: analogWrite (θύρα PWM, ακέραια τιμή 0-255). Εκτελώντας, για παράδειγμα την εντολή analogWrite (51), θα παραχθεί ένας παλμός με διάρκεια 20% της συνολικής ($51/255=0,2$). Επίσης, με μία τροφοδοσία των 5V, η εντολή analogWrite (128) δε θα βγάλει στην έξοδο 2,5V, αλλά κάθε $0,002$ sec, θα βγάζει έναν παλμό των 5V διάρκειας $0,001$ sec. Εάν επιθυμούμε να δημιουργήσουμε ένα αμιγώς αναλογικό σήμα, θα πρέπει να χρησιμοποιήσουμε ένα Digital To Analog Converter (DAC) που συνδέεται με τις ψηφιακές θύρες εξόδου του ελεγκτή (Καλοφωλιάς, 2017).

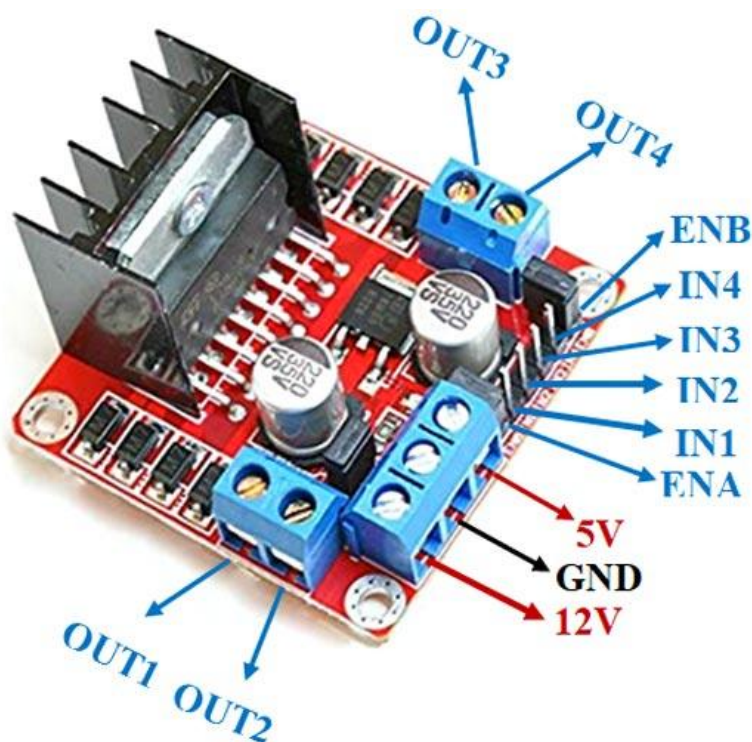
5.5.3 L298N Motor Driver

Το L298N είναι ένα ολοκληρωμένο κύκλωμα διαστάσεων $43,6 \times 43,6 \times 25,5$ mm και βάρους 26 g. Το κύκλωμα αυτό μας παρέχει τη δυνατότητα να ελέγξουμε την ταχύτητα και την κατεύθυνση δύο κινητήρων συνεχούς ρεύματος ή να ελέγξουμε έναν διπολικό βηματικό κινητήρα. Το L298N μπορεί να οδηγήσει κινητήρες με τάση λειτουργίας 5V έως 35V, ενώ παρέχει ρεύμα ανά κανάλι έως 2 A.

Κατασκευαστικά, το L298N αποτελείται από δύο μπλοκ βιδωτών επαφών (δύο επαφές σε κάθε μπλοκ), ένα για κάθε κινητήρα, όπου συνδέονται τα + και τα - του τελευταίου. Επιπλέον, υπάρχει ένα μπλοκ τριών επαφών: μία για τη γείωση του

L298N, μία δεύτερη για την τροφοδοσία των κινητήρων και τέλος μία τρίτη για τα 5V ,η οποία μπορεί να χρησιμοποιηθεί ως είσοδος ή ως έξοδος.

Ας σημειωθεί εδώ, ότι από τα Volt που θα χρησιμοποιηθούν για την τροφοδοσία των DC κινητήρων, θα εξαρτηθεί, εάν η επαφή των 5V που αναφέρθηκε πιο πάνω, θα λειτουργεί ως έξοδος ή ως είσοδος. Το L298N περιέχει έναν ρυθμιστή τάσης (voltage regulator) των 5V. Εάν η τροφοδοσία των κινητήρων είναι έως 12V , τότε ενεργοποιείται ο ρυθμιστής τάσης και καθορίζει την επαφή των 5V ως έξοδο, με την οποία μπορούμε για παράδειγμα να τροφοδοτήσουμε το Arduino. Εάν η τροφοδοσία υπερβαίνει τα 12V, τότε ο ρυθμιστής απενεργοποιείται προκειμένου να μην καταστραφεί , λόγω αυξημένης υπερθέρμανσης. Στην περίπτωση αυτή, η επαφή των 5V θα λειτουργεί ως είσοδος και θα πρέπει να συνδεθεί με εξωτερική τροφοδοσία των 5 V για να λειτουργεί σωστά το L298N.



Εικόνα 5.5.3.1: Οι ακροδέκτες και οι επαφές του L298N, Ανακτήθηκε από:

<https://components101.com/modules/l293n-motor-driver-module>

Το L298N διαθέτει επίσης μία πτώση τάσης της τάξεως των 1,5- 2V. Για το λόγο αυτό, εάν θέλουμε για παράδειγμα, ένας κινητήρας των 12V να λειτουργεί στη μέγιστη ταχύτητά του, θα πρέπει το L298N να τροφοδοτηθεί με τάση υψηλότερη κατά 1,5V τουλάχιστον.

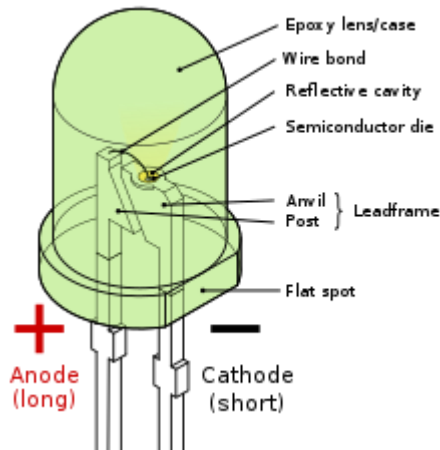
Εκτός από τις προαναφερθείσες επαφές, το L298N διαθέτει τους ακροδέκτες ENABLE A (ENA) και ENABLE B (ENB), οι οποίοι συνδέονται στα pin PWM του Arduino και με αυτά ελέγχεται η ταχύτητα των κινητήρων. Ακόμη, οι ακροδέκτες Input 1 (IN1) και Input 2 (IN2) χρησιμοποιούνται για έλεγχο της φοράς περιστροφής του ενός κινητήρα και οι Input 3 (IN3) και Input 4 (IN4) για τον έλεγχο του άλλου κινητήρα. Χρησιμοποιώντας τους συγκεκριμένους τέσσερις ακροδέκτες , στην ουσία ελέγχουμε τους διακόπτες των γεφυρών H που βρίσκονται στο εσωτερικό του L298N. Οι ακροδέκτες Input συνδέονται σε ψηφιακούς ακροδέκτες του Arduino. Θέτοντας το IN1 ως LOW και το IN2 ως HIGH, ο κινητήρας θα κινηθεί προς τη μία φορά και θέτοντας IN1 ως HIGH και IN2 ως LOW θα κινηθεί προς την αντίθετη φορά.

5.6 Φωτοεκπέμπουσα διόδος (LED)

Η διόδος εκπομπής φωτός, Light Emitting Diode (LED) κατασκευάζεται έτσι, ώστε κάθε φορά που διαρρέεται από ρεύμα να φωτίζει. Η φωτεινότητά της είναι ανάλογη προς το ρεύμα που την διαρρέει (Πογαρίδης, 2020).

Όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο του μέρους A, οι ακροδέκτες 0-13 του Arduino υποστηρίζουν την ανταλλαγή ψηφιακών σημάτων στάθμης 0V ή 5V. Για

να έχουμε αναμμένο ένα LED θα πρέπει να δώσουμε στον ακροδέκτη τάση 5V και για να το σβήσουμε, θα πρέπει να του δώσουμε τάση 0V.



Εικόνα 5.6.1: Φωτοεκπέμπουσα διάδος, Ανακτήθηκε από:

https://en.wikipedia.org/wiki/Light-emitting_diode

5.6.1 Εντολές διαχείρισης LED

Ο έλεγχος του LED περιλαμβάνει τόσο τον καθορισμό του αντίστοιχου ψηφιακού ακροδέκτη ως ακροδέκτη εξόδου, όσο και τη ρύθμιση της στάθμης του σήματος (Παπάζογλου & Λιωνής, 2018).

Εντολή `pinMode` (αριθμός _ ακροδέκτη, τύπος)

Με την παραπάνω εντολή, καθορίζεται ένας ακροδέκτης ως είσοδος ή ως έξοδος. Ο αριθμός _ ακροδέκτη περιλαμβάνει έναν από τους ακροδέκτες 0-13 του Arduino, ενώ ο τύπος μπορεί να είναι INPUT ή OUTPUT.

Εντολή `digitalWrite` (ακροδέκτης, στάθμη)

Με την εντολή αυτή, καθορίζεται η στάθμη σήματος σε έναν ψηφιακό ακροδέκτη εξόδου, η οποία παίρνει τιμές HIGH (5V) ή LOW (0V).

Εντολή delay (πλήθος ms)

Με την εντολή αυτή, δημιουργείται μια χρονική καθυστέρηση έτσι ,ώστε να μπορούμε να δούμε το LED να ανάβει και να σβήνει.

Κεφάλαιο 6: Συγγραφή του Λογισμικού

6.1 Χαρτογράφηση και περιγραφή της κίνησης

Ο σχεδιασμός της κίνησης του οχήματος έλαβε υπόψη του, τις απαιτήσεις ενός ρομποτικού οχήματος που προορίζεται για εκπαιδευτικές και ερευνητικές ανάγκες. Το όχημα έχει μικρό μέγεθος 23x16 cm και προορίζεται να κινείται σε επίπεδες επιφάνειες, με ταχύτητες κίνησης σχετικά μικρές και χωρίς να απαιτούνται απότομες επιταχύνσεις ή επιβραδύνσεις. Ο χώρος στον οποίο πραγματοποιηθήκαν οι δοκιμές της κίνησης του, είναι ένα χαρτόνι διαστάσεων 98,2x138 cm, το οποίο χωρίστηκε σε τετράγωνα με πλευρά 23 cm το καθένα. Επάνω στο χώρο, τοποθετήθηκαν τα εμπόδια που ήταν κουτιά από χαρτόνι, ύψους 12,1 cm και μήκους 19,3 cm. Το ύψος των εμποδίων διαμορφώθηκε με τέτοιο τρόπο, ώστε να είναι αναγνωρίσιμα από τον αισθητήρα HC-SR04.

Η κίνηση πραγματοποιείται με δύο κινητήριους τροχούς που έχουν τοποθετηθεί εκατέρωθεν του εμπρόσθιου τμήματος του οχήματος, κάθε ένας από τους οποίους διαθέτει το δικό του ηλεκτροκινητήρα. Ο τρίτος τροχός χρησιμοποιείται μόνο για τη στήριξη του οχήματος, ωστόσο μπορεί να αλλάζει κατεύθυνση, ακολουθώντας την πορεία που καθορίζουν οι ταχύτητες περιστροφής των δύο κινητήριων τροχών.

Στις εικόνες 6.1.1 έως 6.1.9, απεικονίζεται η κίνηση που προγραμματίστηκε να εκτελεί το ρομποτικό αυτοκινούμενο όχημα της κατασκευής.

Το όχημα, αφού αρχικά τεθεί σε λειτουργία, παραμένει ακινητοποιημένο για 7 δευτερόλεπτα (**Εικόνα 6.1.1**). Εν συνεχεία, ξεκινά να κινείται προς τα εμπρός, εφόσον η απόσταση από το εμπόδιο που μετρά ο αισθητήρας HC-SR04 είναι μεγαλύτερη από πέντε ίντσες. Όταν ο αισθητήρας του οχήματος βρεθεί σε απόσταση μικρότερη ή ίση από 8,5 ίντσες από κάποιο εμπόδιο, τότε το όχημα ακινητοποιείται για τρία δευτερόλεπτα (**Εικόνα 6.1.2**).



Εικόνα 6.1.1



Εικόνα 6.1.2

Έπειτα, ο αισθητήρας HC-SR04 περιστρέφεται με τη βοήθεια του micro servo, πρώτα προς τα αριστερά έως τις 140° . Με την ολοκλήρωση της περιστροφής, μετριέται ξανά η απόσταση από το εμπόδιο που συναντά ο αισθητήρας και εάν αυτή, είναι μεγαλύτερη των 8 ιντσών, το όχημα, μετά από δύο δευτερόλεπτα ακινητοποίησής του, κάνει στροφή 90° , προς τα αριστερά, με σκοπό να το αποφύγει (**Εικόνες 6.1.3, 6.1.4**). Ο αισθητήρας επιστρέφει στην αρχική του θέση, μετά από τρία δευτερόλεπτα

ακινήτοποίησής του. Τέλος, η κίνηση συνεχίζεται μέχρι να εντοπιστεί από τον αισθητήρα το επόμενο εμπόδιο.

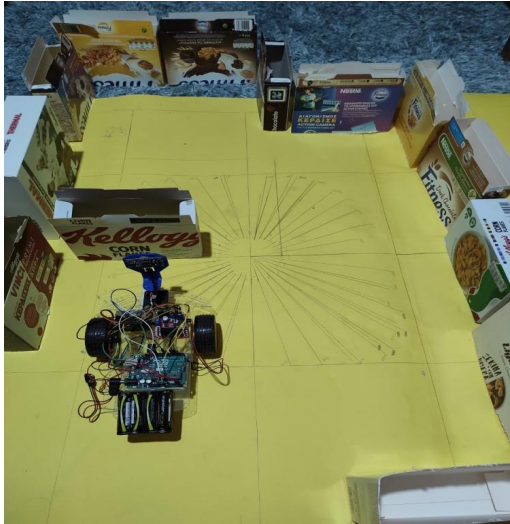


Εικόνα 6.1.3



Εικόνα 6.1.4

Στην περίπτωση που η απόσταση από το εμπόδιο που συναντά ο αισθητήρας δεν είναι μεγαλύτερη των 8 ιντσών, ο αισθητήρας HC-SR04 περιστρέφεται με τη βοήθεια του micro servo από τις 140° που βρίσκεται, προς τα δεξιά έως τις 20° . Μετά το πέρας της περιστροφής, μετριέται η απόσταση από το εμπόδιο που συναντά ο αισθητήρας, και εάν αυτή είναι μεγαλύτερη των 8 ιντσών, το όχημα, μετά από δύο δευτερόλεπτα ακινήτοποίησής του, κάνει στροφή 90° , προς τα δεξιά, με σκοπό να το αποφύγει (Εικόνες 6.1.5, 6.1.6). Ο αισθητήρας επιστρέφει στην αρχική του θέση, μετά από τρία δευτερόλεπτα ακινήτοποίησής του. Στη συνέχεια, η κίνηση του οχήματος συνεχίζεται μέχρι να εντοπιστεί από τον αισθητήρα το επόμενο εμπόδιο.

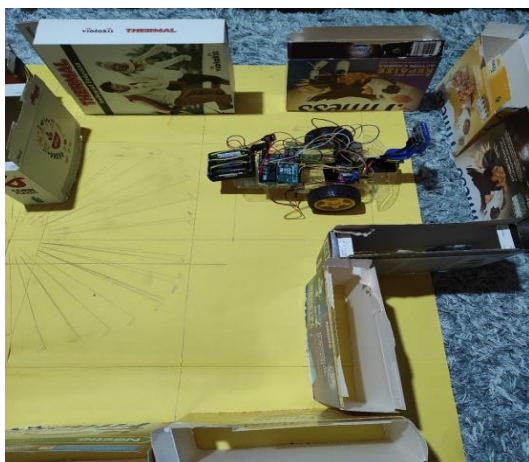


Εικόνα 6.1.5



Εικόνα 6.1.6

Οι κινήσεις που περιγράφηκαν πιο πάνω συνεχίζονται έως ότου, η απόσταση από το εμπόδιο που συναντά ο αισθητήρας κατά την περιστροφή του προς τα αριστερά και προς τα δεξιά, είναι μικρότερη των 8 ιντσών (Εικόνα 6.1.7). Στη συγκεκριμένη περίπτωση, αφού ο αισθητήρας επιστρέψει στην αρχική του θέση, το όχημα θα περιστραφεί κατά 180° και η κίνησή του θα ολοκληρωθεί (Εικόνα 6.1.8).



Εικόνα 6.1.7



Εικόνα 6.1.8

Τέλος, ένα κόκκινο LED, θα αρχίσει να αναβοσβήνει με συχνότητα 500 millisecond, υποδεικνύοντας την ολοκλήρωση της κίνησης.

Ο τρόπος της κίνησης του οχήματος που αναλύθηκε πιο πάνω αποτελεί τον κώδικα που συντάχθηκε και μεταφορτώθηκε στην πλακέτα Arduino του οχήματος.



Εικόνα 6.1.9

6.2 Κώδικας

```
#include <Servo.h> // Συμπερίληψη της βιβλιοθήκης servo.

int IN1=7; // Δήλωση των ακέραιου τύπου integer (int), μεταβλητών
int IN2=2; // IN1-IN4 των αντίστοιχων ακροδεκτών του L298N
int IN3=4; // και εκχώρηση σε αυτές ως τιμές των ψηφιακών ακροδεκτών 7,2,4,12
int IN4=12; // του Arduino.

int ENA=5; // Δήλωση των ακέραιου τύπου integer (int), μεταβλητών ENA και ENB των
αντίστοιχων ακροδεκτών του L298N.

int ENB=3; // και εκχώρηση σε αυτές ως τιμές των ψηφιακών ακροδεκτών 5,3 του Arduino.

float deg; // Δήλωση της πραγματικού τύπου floating point (float), μεταβλητής deg για την
εκχώρηση σε αυτήν, του αριθμού των μοιρών με τις οποίες οι συναρτήσεις turnleft και
turnright κάνουν τον υπολογισμό του χρόνου t.

int Trig=A5; // Δήλωση των ακέραιου τύπου integer (int), μεταβλητών.

int Echo=A4; // Trig και Echo των αντίστοιχων ακροδεκτών του HC-SR04 και εκχώρηση σε
αυτές ως τιμές των αναλογικών ακροδεκτών A5, A4 του Arduino.

int servoPin=8; // Δήλωση της ακέραιου τύπου integer (int), μεταβλητής servoPin για έλεγχο
του servo και εκχώρηση σε αυτή ως τιμή του ψηφιακού ακροδέκτη 8 του Arduino.

int servoPos=80; // Δήλωση της ακέραιου τύπου integer (int), μεταβλητής servoPos για
καθορισμό της θέσης (σε μοίρες) του servo και εκχώρηση σε αυτή της αρχικής τιμής 80.

Servo myServo; // Δημιουργία του αντικειμένου myservo από τη κλάση Servo.

int redLED=13; // Δήλωση του ακέραιου τύπου integer (int), μεταβλητής redLED για
έλεγχο του κόκκινου LED και εκχώρηση σε αυτή ως τιμή του ψηφιακού ακροδεκτή 13 του
Arduino.

void setup() {

pinMode(ENA,OUTPUT); // Καθορισμός των ψηφιακών ακροδεκτών 5, 3

pinMode(ENB,OUTPUT); // ως εξόδων.

digitalWrite(ENA,HIGH); // Απόδοση στους ακροδέκτες 5, 3 της τιμής των 5V

digitalWrite(ENB,HIGH); // ενεργοποίηση δηλαδή των δύο DC gearmotor TT, έτσι ώστε να
καταστούν ελέγξιμα.

pinMode(IN1,OUTPUT); // Καθορισμός των ψηφιακών ακροδεκτών 7

pinMode(IN2,OUTPUT); // 2
```



```

pinMode(IN3,OUTPUT); // 4

pinMode(IN4,OUTPUT); // και 12 ως εξόδων.

pinMode(Trig,OUTPUT); // Καθορισμός των αναλογικών ακροδεκτών A5, A4

pinMode(Echo,INPUT); // ως έξοδος και είσοδος αντίστοιχα.

pinMode(redLED,OUTPUT); // Καθορισμός του ψηφιακού ακροδέκτη 13 ως έξοδος.

pinMode(servoPin,OUTPUT); // Καθορισμός του ψηφιακού ακροδέκτη 8 ως έξοδος.

myServo.attach(servoPin); // Σύνδεση του αντικειμένου myservo με το μοτέρ servo.

myServo.write(servoPos); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos.

}

unsigned int pingTime() // Συνάρτηση υπολογισμού του χρόνου pingTravelTime σε microseconds από την εκπομπή του σήματος Trig του HC-SR04 έως τη λήψη του σήματος ανάκλασης.

unsigned int pingTravelTime; // Δήλωση της μη προσημασμένου ακέραιου τύπου integer (int), μεταβλητής pingTravelTime.

digitalWrite(Trig,LOW); //Δημιουργία παλμού ενεργοποίησης Trig.

delayMicroseconds(2); //Δημιουργία παλμού ενεργοποίησης Trig.

digitalWrite(Trig,HIGH); //Δημιουργία παλμού ενεργοποίησης Trig.

delayMicroseconds(10); //Δημιουργία παλμού ενεργοποίησης Trig.

digitalWrite(Trig,LOW); //Δημιουργία παλμού ενεργοποίησης Trig.

delayMicroseconds(2); //Δημιουργία παλμού ενεργοποίησης Trig.

pingTravelTime=pulseIn(Echo,HIGH); // Ανιχνεύει το αντίστοιχο pin, μέχρι να λάβει έναν παλμό με το επιθυμητό επίπεδο και επιστρέφει μια ακέραια τιμή του τύπου Long που δείχνει τη διάρκεια του παλμού σε msec.

return pingTravelTime; // Επιστροφή της τιμής της μεταβλητής pingTravelTime από την συνάρτηση pingTime.

}

float measDist() // Συνάρτηση μέτρησης απόστασης από τον HC-SR04 έως το εμπόδιο μέσω του χρόνου pingTravelTime και της ταχύτητας του ήχου στον αέρα.

```

unsigned int pingTravelTime; // Δήλωση της μη προσημασμένου ακέραιου τύπου integer (int), μεταβλητής pingTravelTime.

float Dist; // Δήλωση της πραγματικού τύπου floating point (float), μεταβλητής Dist που αντιστοιχεί στην μετατροπή του pingTravelTime σε ίντσες.

float Dist1; // Δήλωση της πραγματικού τύπου floating point (float), μεταβλητής Dist1 που αντιστοιχεί στην απόσταση του εμποδίου από τον HC-SR04 σε ίντσες.

pingTravelTime=pingTime(); // Κλήση της συνάρτησης pingTime και εκχώρηση της τιμής που επιστρέφει στην μεταβλητή pingTravelTime.

Dist=(pingTravelTime*761.*5280.*12.)/(1000000.*3600.); // Υπολογισμός του Dist μέσω της ταχύτητας του ήχου στον αέρα.

Dist1=Dist/2; // Διαίρεση του Dist δια 2 για λήψη της απόστασης του εμποδίου από τον HC-SR04. Εκχώρηση της τιμής στην Dist1.

return Dist1; // Επιστροφή της τιμής της μεταβλητής Dist από την συνάρτηση measDist.

}

void forward(){ // Συνάρτηση που θέτει το όχημα σε κίνηση προς τα εμπρός. Δεν επιστρέφει κάποια τιμή.

digitalWrite(IN1,LOW); // Απόδοση στους ακροδέκτες 7 και 4

digitalWrite(IN2,HIGH); // της τιμής των 0V

digitalWrite(IN3,LOW); // και στους ακροδέκτες 2 και 12

digitalWrite(IN4,HIGH); // της τιμής των 5V.

}

void stopcar(){ // Συνάρτηση ακινητοποίησης του οχήματος. Δεν επιστρέφει κάποια τιμή.

digitalWrite(IN1,LOW); // Απόδοση στους ακροδέκτες 7

digitalWrite(IN2,LOW); // 2

digitalWrite(IN3,LOW); // 4

digitalWrite(IN4,LOW); // 12 της τιμής των 0V.

}

void turnleft(float deg){ // Συνάρτηση με παράμετρο τη μεταβλητή deg που θέτει το όχημα σε αριστερή στροφή. Δεν επιστρέφει κάποια τιμή.

float t; // Δήλωση της πραγματικού τύπου floating point (float), μεταβλητής t που αντιστοιχεί στον χρόνο που απαιτείται για την εκτέλεση της στροφής.

digitalWrite(IN1,LOW); // Απόδοση στους ακροδέκτες 7 και 12

digitalWrite(IN2,HIGH); // της τιμής των 0V

digitalWrite(IN3,HIGH); // και στους ακροδέκτες 2 και 4

digitalWrite(IN4,LOW); // της τιμής των 5V.

t=(deg+14.205)/188.13*1000; // Υπολογισμός της t μέσω μετρήσεων με εκτέλεση στροφών σε διάφορα χρονικά διαστήματα (millisecond) και εξαγωγή της ευθείας μέσω γραμμικής παλινδρόμησης σε Excel.

delay(t); // Εκτέλεση των εντολών digitalWrite με καθυστέρηση όσο η τιμή της t.

stopcar(); // Κλήση της συνάρτησης stopcar για αποφυγή επανάληψης της συνάρτησης turnleft.

}

void turnright(float deg){ // Συνάρτηση με παράμετρο τη μεταβλητή deg που θέτει το όχημα σε δεξιά στροφή. Δεν επιστρέφει κάποια τιμή.

float t; // Δήλωση της πραγματικού τύπου floating point (float), μεταβλητής t που αντιστοιχεί στον χρόνο που απαιτείται για την εκτέλεση της στροφής.

digitalWrite(IN1,HIGH); // Απόδοση στους ακροδέκτες 7 και 12

digitalWrite(IN2,LOW); // της τιμής των 0V

digitalWrite(IN3,LOW); // και στους ακροδέκτες 2 και 4

digitalWrite(IN4,HIGH); // της τιμής των 5V.

t=(deg+14.205)/188.13*1000; // Υπολογισμός της t μέσω μετρήσεων με εκτέλεση στροφών σε διάφορα χρονικά διαστήματα (millisecond) και εξαγωγή της ευθείας μέσω γραμμικής παλινδρόμησης σε Excel.

delay(t); // Εκτέλεση των εντολών digitalWrite με καθυστέρηση όσο η τιμή της t.

stopcar(); // Κλήση της συνάρτησης stopcar για αποφυγή επανάληψης της συνάρτησης turnright.

}

```

void loop() {

    float dObs; // Δήλωση της πραγματικού τύπου floating point (float), μεταβλητής dObs που αντιστοιχεί στην απόσταση του HC-SR04 από το εμπόδιο.

    float dObs1; // Δήλωση της πραγματικού τύπου floating point (float), μεταβλητής dObs1 που αντιστοιχεί στην απόσταση του HC-SR04 από το εμπόδιο κατά την περιστροφή του servo κατά 140°.

    dObs1=0; // Εκχώρηση στην μεταβλητή dObs1 της τιμής 0.

    stopcar(); // Κλήση της συνάρτησης stopcar.

    delay(7000); // Εκτέλεση της συνάρτησης stopcar με καθυστέρηση 7000 millisecond.

    dObs=measDist(); // Κλήση της συνάρτησης measDist. Εκχώρηση της τιμής στη μεταβλητή dObs.

    while(dObs>5){ // Εκτέλεση των εντολών της δομής επανάληψης while όσο η τιμή της dObs είναι μεγαλύτερη των 5 ιντσών.

        dObs=measDist(); // Κλήση της συνάρτησης measDist. Εκχώρηση της τιμής στη μεταβλητή dObs.

        forward(); // Κλήση της συνάρτησης forward.

        while(dObs<=8.5){ // Εκτέλεση των εντολών της δομής επανάληψης while όσο η τιμή της dObs είναι μικρότερη ή ίση των 8,5 ιντσών.

            stopcar(); // Κλήση της συνάρτησης stopcar.

            delay(3000); // Εκτέλεση της συνάρτησης stopcar με καθυστέρηση 3000 millisecond.

            myServo.write(servoPos=95); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (95°).

            delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

            myServo.write(servoPos=110); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (110°).

            delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

            myServo.write(servoPos=125); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (125°).

            delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

            myServo.write(servoPos=140); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (140°).

            delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

```

dObs=measDist(); // Κλήση της συνάρτησης measDist. Εκχώρηση της τιμής στη μεταβλητή dObs.

dObs1=dObs; // Εκχώρηση στην μεταβλητή dObs1 της τιμής της μεταβλητής dObs.

if(dObs>8 && dObs<16){ // Εκτέλεση των εντολών της δομής επανάληψης if όσο η τιμή της dObs είναι μεγαλύτερη των 8 ιντσών και μικρότερη των 16 ιντσών.

dObs1=dObs; // Εκχώρηση στην μεταβλητή dObs1 της τιμής της μεταβλητής dObs.

turnleft(49.1); //Κλήση της συνάρτησης turnleft.

delay(3000); // Εκτέλεση της συνάρτησης turnleft με καθυστέρηση 3000 millisecond.

myServo.write(servoPos=125); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (125⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=110); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (110⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=95); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (95⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=80); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (80⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

}

else if(dObs>8 && dObs<18){ // Εκτέλεση των εντολών της δομής επανάληψης if όσο η τιμή της dObs είναι μεγαλύτερη των 8 ιντσών και μικρότερη των 18 ιντσών.

dObs1=dObs; // Εκχώρηση στην μεταβλητή dObs1 της τιμής της μεταβλητής dObs.

turnleft(63.5); // Κλήση της συνάρτησης turnleft.

delay(3000); // Εκτέλεση της συνάρτησης turnleft με καθυστέρηση 3000 millisecond.

myServo.write(servoPos=125); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (125⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=110); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (110⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=95); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (95⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=80); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (80⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

}

else{

myServo.write(servoPos=125); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (125⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=110); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (110⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=95); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (95⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=80); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (80⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=65); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (65⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=50); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (50⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=35); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (35⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=20); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (20⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

dObs=measDist(); // Κλήση της συνάρτησης measDist. Εκχώρηση της τιμής στη μεταβλητή dObs.

if(dObs>8 && dObs1<=5.5){ // Εκτέλεση των εντολών της δομής επανάληψης if όσο η τιμή της dObs είναι μεγαλύτερη των 8 ιντσών και όσο η τιμή της dObs1 είναι μικρότερη ή ίση των 5,5 ιντσών.

turnright(54.8); // Κλήση της συνάρτησης turnright.

delay(3000); // Εκτέλεση της συνάρτησης turnright με καθυστέρηση 3000 millisecond.

myServo.write(servoPos=35); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (35⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=50); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (50⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=65); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (65⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=80); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (80⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

}

else if(dObs>8 && dObs1>5.5){ // Εκτέλεση των εντολών της δομής επανάληψης if όσο η τιμή της dObs είναι μεγαλύτερη των 8 ιντσών και όσο η τιμή της dObs1 είναι μεγαλύτερη των 5,5 ιντσών.

turnright(40.4); // Κλήση της συνάρτησης turnright.

delay(3000); // Εκτέλεση της συνάρτησης turnright με καθυστέρηση 3000 millisecond.

myServo.write(servoPos=35); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (35⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=50); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (50⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=65); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (65⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=80); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (80⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

}

else{

myServo.write(servoPos=35); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (35⁰).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=50); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (50°).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=65); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (65°).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

myServo.write(servoPos=80); // Περιστροφή του servo σε γωνία όσο η τιμή της μεταβλητής servoPos (80°).

delay(2000); // Εκτέλεση της εντολής myServo.write με καθυστέρηση 2000 millisecond.

turnleft(140); // Κλήση της συνάρτησης turnleft.

delay(2000); // Εκτέλεση της συνάρτησης turnleft με καθυστέρηση 2000 millisecond.

stopcar(); // Κλήση της συνάρτησης stopcar.

while(1==1){ // Εκτέλεση των εντολών της δομής επανάληψης while όσο ισχύει η ισότητα 1==1, που ισχύει πάντα.

digitalWrite(redLED,HIGH); // Απόδοση στον ακροδέκτη 13 της τιμής των 5V.

delay(500); // Εκτέλεση της συνάρτησης digitalWrite με καθυστέρηση 500 millisecond.

digitalWrite(redLED,LOW); // Απόδοση στον ακροδέκτη 13 της τιμής των 0V.

delay(500); // Εκτέλεση της συνάρτησης digitalWrite με καθυστέρηση 500 millisecond.

}

}

}

dObs=measDist(); // Κλήση της συνάρτησης measDist. Εκχώρηση της τιμής στη μεταβλητή dObs.

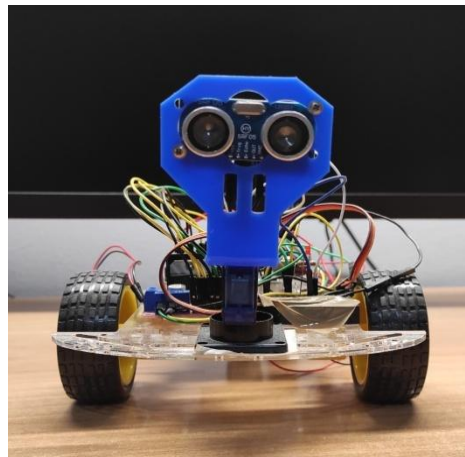
}

}

}

6.3 Παρατηρήσεις- Επεξηγήσεις Κώδικα

1. Όπως έχει αναφερθεί στο Κεφάλαιο 1 του μέρους Β' της εργασίας, το micro servo έχει εύρος περιστροφής από 0° - 180° . Στην πραγματικότητα όμως, η περιστροφή του servo αγγίζει περίπου τις 170° . Συνεπώς, με την εντολή του κώδικα **int servoPos=80;** το servo τίθεται σε αρχική θέση 80° αντί για την αναμενόμενη θέση των 90° .



Εικόνα 6.3.1: Αρχική θέση servo, Φωτογραφία προσωπικού αρχείου

2. Η βαθμονόμηση του αυτοκινούμενου ρομποτικού οχήματος ως προς τις στροφές που αυτό εκτελεί με τις συναρτήσεις turnleft και turnright έγινε με τη βοήθεια της μεθόδου της γραμμικής παλινδρόμησης, μέσω του προγράμματος Excel στον υπολογιστή. Ειδικότερα, πραγματοποιήθηκαν κάποιες μετρήσεις που περιλάμβαναν στροφές του οχήματος σε κάποια δευτερόλεπτα. Οι μετρήσεις αυτές δίνονται στον πίνακα 1:

Πίνακας 1. Μετρήσεις Στροφών Οχήματος

Time(sec.)	Degrees(⁰)
0,3	40
0,5	87
1	165
1,5	272

Το πρόγραμμα Excel με τις επιλογές **Εισαγωγή-> Διασπορά** ,μας δίνει την ευθεία και την εξίσωση αυτής **$y=188,13x-14,205$** που αντιστοιχεί στην εντολή **$t=(deg+14.205)/188.13*1000$** ; Η εξίσωση έχει πολλαπλασιαστεί με το 1000 για μετατροπή των seconds σε milliseconds, καθώς η εντολή **delay** δέχεται ως όρισμα, χρόνο σε milliseconds.

3. Λαμβάνοντας υπόψη τη ταχύτητα του ήχου στον αέρα (340 m/sec ή 761 miles/hour), από τον τύπο της ταχύτητας ($u=d/t$ με $t= pingTravelTime$) παίρνουμε τη σχέση:

$$d(\text{Inches})=761(\text{miles/hour}) * pingTravelTime(\text{mseconds}) *$$

$$1(\text{sec})/1.000.000(\text{mseconds}) * 1(\text{hour})/3600(\text{seconds}) * 5280(\text{ft})/1(\text{mile}) * 12(\text{inches})/1$$

$$(\text{ft})=761 * 5280 * 12 * pingTravelTime / 1.000.000 * 3600.$$

Η σχέση αυτή, αν διαιρεθεί με το 2, μας δίνει την απόσταση του εμποδίου από τον αισθητήρα HC-SR04. Στον κώδικα, αποτυπώνεται με την εντολή **$Dist=(pingTravelTime*761.*5280.*12.)/(1000000.*3600.);$**

4. Η μεταβλητή **pingTravelTime** έχει δηλωθεί ως **unsigned int** στον κώδικα, με σκοπό να αποφευχθεί η υπερχείλιση της. Όταν δεν υπάρχουν εμπόδια σε απόσταση τέτοια ώστε να είναι ανιχνεύσιμα από τον αισθητήρα HC-SR04, η μεταβλητή θα

μπορούσε να υπερχειλίσει προκαλώντας ενδεχόμενα προβλήματα στην εκτέλεση του κώδικα.

Κεφάλαιο 7: Περιορισμοί- Προτάσεις

7.1 Σφάλματα

Κατά τη διάρκεια των δοκιμαστικών τεστ που πραγματοποιήθηκαν με σκοπό τον έλεγχο της κίνησης του οχήματος, αναγνωρίστηκαν κάποιοι βασικοί περιορισμοί - σφάλματα που εμπόδισαν την ομαλή μετάβασή του οχήματος από τον προορισμό Α στον προορισμό Β. Ειδικότερα, παρατηρήθηκε ελάχιστη απόκλιση του οχήματος δεξιά ή αριστερά, κατά την εμπρόσθια κίνησή του. Έτσι, κατά την ακινητοποίησή του μπροστά στα εμπόδια (στις 8,5 ίντσες περίπου), εμφάνιζε μια ελαφριά απόκλιση στην διεύθυνσή του είτε προς τα δεξιά, είτε προς τα αριστερά.

Το σφάλμα αυτό μπορεί να αποδοθεί στο γεγονός ότι πρόκειται για μια κατασκευή που περιλαμβάνει εξαρτήματα χαμηλού κόστους. Συγκεκριμένα, οι τροχοί του παρουσιάζουν μία ελαφριά ολίσθηση και η κίνησή τους δεν είναι απολύτως κυκλική, με απόκλιση στην ευθυγράμμισή τους. Ένας ακόμη λόγος που οδήγησε στην εμφάνιση του παραπάνω σφάλματος, χωρίς ωστόσο να παρατηρηθεί συχνά κατά τις δοκιμές του οχήματος, ήταν η μη ταυτόχρονη εκκίνηση των δύο κινητήρων και συνεπώς των τροχών του οχήματος. Αυτό οφείλονταν στη πιθανή μειωμένη παροχή volt από την μπαταριοθήκη.

Η εμφάνιση του παραπάνω σφάλματος είχε ως αποτέλεσμα, το όχημα να παρεκκλίνει από τον αρχικό σχεδιασμό του που ήταν η αποφυγή των εμποδίων, εκτελώντας περιστροφή αριστερά ή δεξιά κατά 90^0 . Για την αντιμετώπιση του προβλήματος, ορίστηκε η περιστροφή του οχήματος να είναι συγκεκριμένες μοίρες κάθε φορά,

ανάλογα με την απόσταση που μετρούσε ο αισθητήρας HC-SR04 από το εμπόδιο.

Αυτό επιτεύχθηκε με τις εντολές του κώδικα: **turnright(40.4);** **turnright(54.8);**
turnleft(63.5); **turnleft(49.1);**.

Σχετικά με τη βαθμονόμηση των στροφών του οχήματος με τη μέθοδο της γραμμικής παλινδρόμησης, μπορούμε να πούμε ότι προσέφερε ικανοποιητικά αποτελέσματα, παρά την μικρή απόκλιση μοιρών σε σχέση με τις επιθυμητές.

7.2 Προτάσεις για βελτιστοποίηση της λειτουργίας του Οχήματος

Το σφάλμα της απόκλισης από τη διεύθυνση της κίνησης, του οχήματος, θα μπορούσε να αποφευχθεί στο μέλλον με μια κατασκευή η οποία θα περιλάμβανε δύο επιπλέον αισθητήρες HC-SR04 αριστερά και δεξιά του οχήματος. Οι αισθητήρες αυτοί θα έδιναν τη δυνατότητα μέτρησης της απόστασης από τα εμπόδια εκατέρωθεν του οχήματος, διατηρώντας την ευθεία κίνησή του, χωρίς αποκλίσεις. Επίσης, θα μπορούσε η κατασκευή να διαθέτει θήκη με 5 μπαταρίες AA που θα παρείχαν στο όχημα συνολική τάση ίση με 7,5 volt. Έτσι, θα τροφοδοτούνταν οι δύο κινητήρες χωρίς απώλειες, δεδομένης και της δέσμμευσης των περίπου 2V από το L298N. Με τον τρόπο αυτό, μπορεί να αποφευχθεί επιπλέον και το σφάλμα της μη ταυτόχρονης εκκίνησης των δύο ηλεκτροκινητήρων DC.

7.3 Συμπεράσματα

Η τεχνολογική εξέλιξη στον τομέα της ρομποτικής και της αυτόνομης κίνησης έχει σημειώσει τεράστια πρόοδο την τελευταία δεκαετία, έχοντας ως αποτέλεσμα την κυκλοφορία αυτοκινούμενων ρομποτικών οχημάτων στην ευρεία αγορά. Τα

ρομποτικά οχήματα περιλαμβάνουν τόσο προϊόντα που προορίζονται για οικιακή χρήση, όπως για παράδειγμα οι ρομποτικές σκούπες, όσο και τα σημερινά τεχνολογικά ανεπτυγμένα αυτόνομα οχήματα των αυτοκινητοβιομηχανιών.

Σκοπός της παρούσας εργασίας ήταν η μελέτη της κίνησης που εκτελεί ένα αυτόνομο όχημα, όταν βρεθεί σε συγκεκριμένο περιβάλλον εμποδίων. Κύριος στόχος είναι η ασφαλή λειτουργία και πλοήγηση του οχήματος με τη σύνθεση των πληροφοριών που δέχεται από τον αισθητήρα HC-SR04.

Το ρομποτικό όχημα που κατασκευάστηκε συνδυάζει το χαμηλό κόστος με τις δυνατότητες που καλύπτουν τις σύγχρονες ερευνητικές ανάγκες, όπως η χρήση του μικροελεγκτή AVR και του αισθητήρα HC-SR04 που συναντώνται και στα σημερινά τεχνολογικά ανεπτυγμένα αυτόνομα οχήματα. Η αρχική σχεδίαση του οχήματος στο πρόγραμμα Fritzing, συνέβαλλε στην ορθή συναρμολόγηση των επιμέρους τμημάτων του για την υλοποίηση της κατασκευής του σε φυσική μορφή.

Διαπιστώνεται ότι, παρά τους περιορισμούς που αναπτύχθηκαν κατά το σχεδιασμό της κίνησής του, η αρχή λειτουργίας των αυτόνομων οχημάτων επαληθεύτηκε. Το όχημα είναι ικανό να πλοηγείται σε ένα χώρο με εμπόδια και να φθάνει στο προορισμό του, χωρίς την παρέμβαση από κάποιο χρήστη. Αυτό επιτυγχάνεται με την ανίχνευση των εμποδίων από τον αισθητήρα υπερήχων που διαθέτει, ο οποίος μετρά την απόσταση του οχήματος από αυτά. Αφού μετρηθεί η απόσταση από το εμπόδιο, το όχημα μπορεί να ακινητοποιείται σε συγκεκριμένη απόσταση από αυτό. Έπειτα, επιλέγει την πορεία που θέλει να ακολουθήσει πραγματοποιώντας ελιγμό προς την κατεύθυνση αυτή και αποφεύγοντας έτσι το συγκεκριμένο εμπόδιο.

Εν κατακλείδι, πρόκειται για μία κατασκευή που υλοποιήθηκε στα πλαίσια διπλωματικής εργασίας και απευθύνεται κυρίως σε φοιτητές. Παρ' όλα αυτά, αποτελεί

μια ενδιαφέρουσα μελέτη που παρέχει αξιόπιστες πληροφορίες για την τεχνολογία των αυτόνομων οχημάτων και την προοπτική τους για το μέλλον. Τέλος, θα μπορούσε να αποτελέσει πηγή έμπνευσης για τον σχεδιασμό παρόμοιων ρομποτικών οχημάτων από άλλα άτομα.

Βιβλιογραφία

Arduino Home: Available at: <https://www.arduino.cc/>

Bertozzi, M., Broggi, A., Conte, G., & Fascioli, A. (1998). *Experience of the ARGO autonomous vehicle*, Proc. SPIE (3364), 218-229. *Enhanced and Synthetic Vision*

Available at: <https://doi.org/10.1117/12.317473>

Γαστεράτος, Α., Μουρούτσος, Σ., & Ανδρεάδης, Ι. (2013). *Τεχνολογία Μετρήσεων. Αισθητήρια*, εκδ. Τσότρας, 43-50.

DARPA Grand_Challenge (2004)

Available at: https://en.wikipedia.org/wiki/DARPA_Grand_Challenge

Energias Market Research. (2018). *Microcontrollers Performing High Standards in automobiles industry, to Witness a CAGR of 17.2% during 2017 – 2023.*

Available at: <https://www.globenewswire.com/en/news-release/2018/09/13/1570305/0/en/Microcontrollers-Performing-High-Standards-in-automobiles-industry-to-Witness-a-CAG>

Ευρωπαϊκό Κοινοβούλιο ,Οικονομία (2019). *Αυτόνομα αυτοκίνητα στην ΕΕ: από επιστημονική φαντασία...σε απτή πραγματικότητα. Επικαιρότητα*, Available at:

<https://www.europarl.europa.eu/news/el/headlines/economy/20190110STO23102/>

Glou, R. & Edelstein, S. (2020). *The history of self-driving car. Digital trends daily.*

Available at: www.digitaltrends.com/cars/history-of-self-driving-cars-milestones/

Houdina Radio Control – Wikipedia (2022)

Available at: https://en.wikipedia.org/wiki/Houdina_Radio_Control

Introduction to electronic Sensors. Available at:

<https://predictabledesigns.com/introduction-to-electronic-sensors/>

Jost, D. (2019). *What is a Sensor?* Available at:

<https://www.fierceelectronics.com/sensors/what-a-sensor>

Καλοφωλιάς, Δ. (2017). *Ο προγραμματισμός του μικροελεγκτή AVR ATmega328 με τη χρήση της πλατφόρμας Arduino*, εκδ. Τζιόλα, 48-212.

Καμινιάρης, Σ. (2020). *Μάθε τα πάντα για την αυτόνομη οδήγηση και τα επίπεδα*

αυτονομίας. Available at: <https://www.autonomous.gr/mathe-ta-panta-tin-aytonomi-odigisi-ta-epipeda-aytonomias/>

Καραϊσκος, Χ.(2010). *Ο Μικροελεγκτής 8051*. Σύγχρονη Εκδοτική, 11-29

Lardinois, F. (2012). *Google's Self-Driving Cars Complete 300K Miles Without Accident, Deemed Ready For Commuting*. Available at:

<https://techcrunch.com/2012/08/07/google-cars-300000-miles-without-accident/>

Litman, T. (2022). *Autonomous Vehicle Implementation Predictions Implications for Transport Planning*. Victoria Transport Policy Institute. 3-42. Available at:

<https://www.vtpi.org/avip.pdf>

Lutkevich, B.(2019). *Self-driving car (autonomous car or driverless car)*, Available at:

https://www.techtarget.com/searchenterpriseai/definition/driverless-car?fbclid=IwAR0gweE6Sg-C25IYG3LXEGdGfLMooVEVyXXQ7fgAsMxcClru-t3pR4vY8E_c

- Ξύδης, Θ. (2020). *LiDAR: Η συσκευή που έκανε τα αυτόνομα αυτοκίνητα πραγματικότητα*. Available at: <https://traction.gr/lidar-i-syskevi-pou-ekane-ta-aftonoma-aftokinita-pragmatikotita/>
- Rambus Press. (2022). *Autonomous Vehicles: Everything about self-driving cars explained*. Available at: https://www.rambus.com/blogs/autonomous-vehicles-explained/?fbclid=IwAR2ki2ltEVZ7qAhDX25G7u6_15k8ijXc8HZVzVN4OMBo4kDbMduVUTKsODM
- RedHat. (2021). *What is an autonomous vehicle?* Available at: <https://www.redhat.com/en/topics/edge-computing/what-is-an-autonomous-vehicle?fbclid=IwAR3ZKhMrQezp4j6R9XCUn1WK-X5oJzsel3L-K0XLftFOcLVszTDf2cdCbU>
- Παπάζογλου, Π., & Λιωνής, Σ. (2018). *Ανάπτυξη εφαρμογών με το Arduino*, 2^η έκδοση, εκδ.Τζιόλα, 1-304.
- Πογαρίδης, Δ. (2020). *Ενσωματωμένα Συστήματα. Οι μικροελεγκτές AVR και Arduino*, εκδ. Δίσιγμα, 13-412.
- SAE (2021). *Levels of Driving Automation Refined for Clarity and International Audience*. Available at: <https://www.sae.org/blog/sae-j3016-update>
- Teja, R., (2021). *What is a Sensor? Different Types of Sensors and their Applications*. *Electronics Hub*. Available at: <https://www.electronicshub.org/different-types-sensors/>
- Trevennor, A. (2012). *A Brief History of Microcontrollers*. In: *Practical AVR Microcontrollers*. Apress, Berkeley, CA. Available at: https://doi.org/10.1007/978-1-4302-4447-9_1

Τσελές, Δ. (2002). *Συλλογή και επεξεργασία δεδομένων*. Σύγχρονη Εκδοτική, 140-142.