ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Σχεδίαση Radar με χρήση εργαλείων ανοιχτού κώδικα

Δανάη Χ. Τζιτζιλέρη
Α.Μ. 40979

Εισηγητές: **Δρ Ιωάννης Έλληνας, Καθηγητής**
κ. Στυλιανός Βουτσινάς, Πανεπιστημιακός Υπότροφος

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Σχεδίαση Radar με χρήση εργαλείων ανοιχτού κώδικα

Δανάη Χ. Τζιτζιλέρη
Α.Μ. 40979

Εισηγητής:

Δρ. Ιωάννης Έλληνας, Καθηγητής
κ. Στυλιανός Βουτσινάς, Πανεπιστημιακός Υπότροφος

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Η κάτωθι υπογεγραμμένη Δανάη Τζιτζιλέρη, του Χαραλάμπους, με αριθμό μητρώου 40979 φοιτήτρια του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασης της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

ΕΥΧΑΡΙΣΤΙΕΣ

Την προσπάθειά μου στην εκπόνηση της πτυχιακής μου εργασίας υποστήριξε ο επιβλέπων καθηγητής μου Δρ. Ιωάννης Έλληνας, τον οποίο θα ήθελα να ευχαριστήσω.

Η παρούσα πτυχιακή εργασία δε θα μπορούσε να ολοκληρωθεί χωρίς την αμέριστη συμπαράσταση του κ. Στέλιου Βουτσινά, τον οποίο ευχαριστώ βαθύτατα.

ΠΕΡΙΛΗΨΗ

Τα οχήματα αυτοματισμού και αυτόνομης κίνησης άπτονται του κλάδου της Μηχατρονικής και ενσωματώνουν πολλούς κλάδους της μηχανικής όπως ηλεκτρολογία, μηχανολογία, προγραμματισμό, ηλεκτρονική και μηχανική υπολογιστών (και τεχνολογίας). Αλλάζουν ριζικά την τεχνολογία αυτοκινήτων μεταφοράς και παραγωγής. Επομένως, αναπτύσσοντας νέα, τεχνολογικά προοδευτικά προγράμματα σπουδών και εργαστηριακής έρευνας, θα μπορούσε να προετοιμαστεί το έδαφος για την κάλυψη αναγκών και τεχνογνωσίας του μελλοντικού εργατικού δυναμικού των αυτόνομων αυτοκινήτων στον κλάδο της αυτοκινητοβιομηχανίας. Η συγκεκριμένη μελέτη σχεδίασης συστήματος ραντάρ για την αυτόματη στάθμευση οχήματος αποτελεί ένα πρώτο βήμα στη μελέτη αυτοματοποίησης οχημάτων του μέλλοντος, η οποία μπορεί να επεκταθεί στο μέλλον σε ερευνητικό επίπεδο και να βελτιωθεί, προωθώντας και εξελίσσοντας την τεχνολογία αυτονομίας οχημάτων. Σε αυτό το έργο, ένα τηλεχειριζόμενο αυτοκίνητο RC (παιχνίδι) έχει τροποποιηθεί με την ενσωμάτωση μικροεπεξεργαστή και Motor Shield για τον έλεγχο του οχήματος στις κινήσεις και τις διαδικασίες στάθμευσης καθώς αισθητήρες υπερήχων για τη λήψη των απαραίτητων χωρικών (αποστάσεις αντικειμένων) πληροφοριών. Κατόπιν δοκιμών, το αυτοκίνητο είναι σε θέση να κινηθεί για να ανιχνεύσει, «μιμούμενο» το «οδικό» περιβάλλον και στη συνέχεια, κρίνοντας αν ένας χώρος είναι κατάλληλος για στάθμευση ή όχι, να παρκάρει αυτόματα ή να συνεχίσει την πορεία του. Η συγκεκριμένη σχεδίαση αναπτύσσεται σε πλατφόρμα Arduino Uno. Η ανάπτυξη του λογισμικού στηρίζεται σε εργαλεία ανοιχτού κώδικα.

ABSTRACT

Vehicle automation, autonomy and connectivity is a subject of mechatronics integrating many engineering disciplines including electrical, mechanical, control, and computer engineering (and technology). It is fundamentally changing the concept of automobile transportation and manufacturing. Therefore, developing new, technologically progressive curricula and hands-on lab as well as student project materials is desired to prepare for the future work-force needs of autonomous cars in the automotive industry. This particular radar system design study for automatic vehicle parking is a research and concept-proving project that will be prepared and extended to develop teaching materials for courses and students project on the subject of vehicle automation, autonomy and connectivity. In this project, an RC (remote-controlled) toy car is modified by integrating ultrasound sensors and Arduino with a high current shield to control the vehicle movements and the parking processes. Parking strategies and the corresponding algorithms are explored and programmed through Arduino. During testing, the car is able to move to detect the imitated “roadside” environment, judge a space suitable for parking or not, and then drive to park automatically. This particular design is developed on an Arduino Uno platform. The software is developed by using open source tools.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Αρχιτεκτονική Ηλεκτρονικών Υπολογιστών, Μηχατρονική

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Ραντάρ, Arduino Uno, ανοιχτού κώδικα, αυτόματο παρκάρισμα, ανίχνευση υπερήχων.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

1. ΕΙΣΑΓΩΓΗ

1.1 Περιγραφή του προβλήματος

Πάνω από το ήμισυ της ανθρωπότητας, ζει σήμερα στις πόλεις. Το 2030 ο αριθμός αυτός θα αυξηθεί, συμφώνα με έρευνες, στο 60%. Οι πόλεις έχουν μεγάλο κινητικό, ενεργειακό πρόβλημα και πρόβλημα ρύπανσης που μειώνει την ποιότητα ζωής. [1]

Το 35% των ανθρώπων που ζουν στις πόλεις μετακινείται κατά τις ώρες αιχμής και ψάχνει για σημεία στάθμευσης, που είναι δύσκολο να βρεθούν. Αυτός είναι ο λόγος για τον οποίο η στάθμευση είναι πολύ σημαντική για τη ρύθμιση της ροής των οχημάτων και τη μείωση της ατμοσφαιρικής ρύπανσης. [1]

Όπως δείχνουν και πολλές άλλες μελέτες, ένα σημαντικό πρόβλημα που αντιμετωπίζουν οι οδηγοί είναι αυτό του παρκαρίσματος. Σε κάποια μοντέλα αυτοκινήτων έχουν ήδη εφαρμοστεί μηχανισμοί υποβοήθησης παρκαρίσματος, γεγονός το οποίο συνέβαλε στην εξεύρεση λύσης (έστω και μερικής) στο πρόβλημα. Μηχανισμοί αυτόματου παρκαρίσματος (που θα οδηγούσαν ίσως στην οριστική λύση) έχουν τοποθετηθεί μόνο σε αυτοκίνητα υψηλών (οικονομικά) κατηγοριών.

1.2 Περιγραφή του αντικειμένου της πτυχιακής εργασίας

Στόχος αυτού του έργου είναι να σχεδιαστεί ένα πρωτότυπο σύστημα ραντάρ που θα καθοδηγήσει ένα όχημα να εκτελέσει αυτόνομους ελιγμούς όταν εντοπίσει έναν επαρκή χώρο στάθμευσης και τελικά να παρκάρει. Το σύστημα αυτό περιλαμβάνει μια σειρά από ηχητικούς αισθητήρες, καθώς και μια κεντρική μονάδα μικροεπεξεργαστή που ελέγχει το αυτοκίνητο. Οι αισθητήρες είναι τοποθετημένοι στην περίμετρο του αυτοκινήτου. Είναι συνδεδεμένοι με τον μικροεπεξεργαστή ενός Arduino Uno, ο οποίος λαμβάνει και επεξεργάζεται τα δεδομένα από τις εξόδους των αισθητήρων.

Το αυτοκίνητο κινείται ευθεία, βασισμένο σε τιμές που δέχεται από τους αισθητήρες. Το αυτοκίνητο έχει οπίσθια κίνηση ενώ οι μπροστινοί τροχοί έχουν το σύστημα διεύθυνσης (στρίβουν το όχημα δεξιά και αριστερά). Ο εμπρόσθιος άξονας είναι συνδεδεμένος με έναν DC κινητήρα. Οι πίσω τροχοί συνδέονται σε έναν άλλο DC κινητήρα και κατευθύνουν το αυτοκίνητο μπροστά και πίσω. Και οι δύο κινητήρες ρυθμίζονται μέσω ενός οδηγού H-Bridge ο οποίος έχει στελεχωθεί

πάνω στον μικροεπεξεργαστή της Arduino UNO. Η H-Bridge επιτρέπει στους τροχούς να αλλάζουν κίνηση μπροστά και πίσω, ενώ μέσω του εμπρόσθιου συστήματος περιστροφής να κινούνται δεξιά και αριστερά.

Βασική λογική του όλου εγχειρήματος είναι να τοποθετηθεί σε ένα όχημα προσομοίωσης ένας μηχανισμός ο οποίος, με βάση τα σήματα που δέχονται οι αισθητήρες, θα οδηγεί το αυτοκίνητο προσομοίωσης μέχρι να βρει θέση παρκαρίσματος και θα το παρκάρει. Σκοπός του όλου εγχειρήματος είναι η κατασκευή ενός μηχανισμού, ο οποίος με μικρές παραλλαγές να μπορεί να εφαρμοστεί αν όχι σε όλα, σε πολλά κανονικά αυτοκίνητα.

Κύριος γνώμονας για την επιλογή των εξαρτημάτων στη συγκεκριμένη σχεδίαση είναι αφ' ενός μεν η ελαχιστοποίηση (όσο το δυνατόν) του κόστους τόσο των υλικών όσο και του λογισμικού και αφ' εταίρου τα υλικά να κυκλοφορούν ευρέως στο εμπόριο, οπότε θα είναι εύκολη η προμήθειά τους.

1.3 Ανάλυση εργασίας

Στο επόμενο ($2^{ο}$) κεφάλαιο γίνεται μνεία άλλων μεθόδων που έχουν αναπτυχθεί για το αυτόματο παρκάρισμα.

Στο $3^{ο}$ κεφάλαιο αναλύεται η λογική του μηχανισμού της συγκεκριμένης μελέτης, και τα επιμέρους εξαρτήματα που χρησιμοποιήθηκαν: ποια είναι, γιατί επιλέχθηκαν και πως συνδέονται μεταξύ τους. Στο ίδιο κεφάλαιο περιγράφεται, επίσης, ο κώδικας για τον μικροεπεξεργαστή του συστήματος.

Στο $4^{ο}$ κεφάλαιο περιγράφεται η διαδικασία που ακολουθήθηκε για την υλοποίηση του project: οι πρώτες προσπάθειες, οι δυσκολίες που αντιμετωπίστηκαν και τελικά η κατασκευή.

Στο $5^{ο}$ κεφάλαιο αναφέρονται τα αποτελέσματα της εργασίας και τα συμπεράσματα που αντλούνται από αυτήν. Επίσης, παρατίθενται ιδέες για τη βελτίωση και την επέκταση παρόμοιων εφαρμογών.

2. ΜΕΘΟΔΟΙ ΠΟΥ ΕΧΟΥΝ ΑΝΑΠΤΥΧΘΕΙ ΓΙΑ ΤΟ ΑΥΤΟΜΑΤΟ ΠΑΡΚΑΡΙΣΜΑ

Μέχρι σήμερα έχουν σχεδιαστεί και αναπτυχθεί πολλά συστήματα για αυτόματο παρκάρισμα. Κάποια από αυτά τα συστήματα είναι βιομηχανικών προδιαγραφών, κάποια άλλα είναι συστήματα για προσομοιώσεις και κάποια θεωρητικά.

2.1 Τα συστήματα βιομηχανικών προδιαγραφών

Τα συστήματα βιομηχανικών προδιαγραφών που χρησιμοποιούν οι αυτοκινητοβιομηχανίες συνήθως το αναφέρουν ως Active ή Intelligent Parking Assist System (IPAS). Το πρώτο IPAS το εφάρμοσε η Toyota Motor Corporation το 1999, αρχικά για τα μοντέλα υβριδικών Prius της Ιαπωνίας και τα μοντέλα της Lexus ώστε να βοηθούνται οι οδηγοί να σταθμεύουν το όχημά τους. Τα οχήματα που ήταν εφοδιασμένα με αυτό το σύστημα, μέσω μιας οθόνης στην κονσόλα και των κουμπιών ελέγχου, μπορούσαν να κατευθυνθούν σε χώρο στάθμευσης με ελάχιστη επέμβαση του οδηγού. Η πρώτη έκδοση του συστήματος αναπτύχθηκε στο Prius Hybrid που πωλήθηκε στην Ιαπωνία το 2003. Το 2006, μια αναβαθμισμένη έκδοση κυκλοφόρησε για πρώτη φορά έξω από την Ιαπωνία στο πολυτελές sedan της Lexus LS, το οποίο εδραίωσε την τεχνολογία αυτόματης στάθμευσης. [2]

Η αρχική έκδοση του συστήματος Intelligent Parking Assist, που ξεκίνησε το 2003, σχεδιάστηκε για αντίστροφη παράλληλη στάθμευση. Η παρέμβαση του οδηγού δεν ήταν απαραίτητη, καθώς το σύστημα εκτιμούσε το μέγεθος του χώρου και έκανε τους κατάλληλους ελιγμούς για τη στάθμευση του οχήματος. Το σύστημα το έλεγχε ένας ενσωματωμένος υπολογιστής που χρησιμοποίησε μια κάμερα ενσωματωμένη στο εμπρός και πίσω μέρος του αυτοκινήτου. Οι αισθητήρες, τοποθετημένοι σε παρόμοιες θέσεις, ανίχνευαν την εγγύτητα των κοντινών οχημάτων. Η οθόνη στην κονσόλα εμφάνιζε μια εικόνα του χώρου στάθμευσης και ο οδηγός έπρεπε στη συνέχεια να καθορίσει την ακριβή θέση που θα έπρεπε να πάρει το όχημα μέσω των βέλων που εμφανίζονταν στην οθόνη. Χρησιμοποιώντας τα βέλη, ο οδηγός καθόριζε τη θέση του οχήματος στο χώρο στάθμευσης και έδινε εντολή (με το κατάλληλο κουμπί) στο IPAS να πάρει τον έλεγχο διεύθυνσης για να ελιχθεί το όχημα και τελικά να παρκάρει στον επιλεγμένο από πριν χώρο. [2]

Εικόνα 42 Η οθόνη στην κονσόλα του Toyota Prius. [3]

Οι πρώιμες εκδόσεις αυτού του συστήματος είχαν δυσκολία στην ανίχνευση αντικειμένων, συμπεριλαμβανομένων των γάτων, των παιδικών καροτσιών και των πεζών. Δεύτερον, όταν ο οδηγός ενεργοποιούσε το σύστημα σε πολύ μικρό χώρο, το σύστημα προειδοποιούσε τον οδηγό για τον κίνδυνο να χτυπήσει το όχημα. Σε αυτές τις περιπτώσεις ήταν απαραίτητη η παρέμβαση του οδηγού. Το 2005, μια αναβαθμισμένη έκδοση προσέθεσε την ικανότητα αναγνώρισης των λωρίδων στάθμευσης. Σε μια μεταγενέστερη έκδοση αυτής της τεχνολογίας στάθμευσης, που ξεκίνησε το 2006, προστέθηκαν αισθητήρες στάθμευσης. Αυτή η τελευταία έκδοση μπορούσε να υπολογίσει τους ελιγμούς διεύθυνσης που απαιτούνταν για παράλληλο ή αντίστροφο παρκάρισμα και να βοηθήσει να διαπιστωθεί αν το αυτοκίνητο έχει αρκετό χώρο για το συγκεκριμένο χώρο με έγχρωμες ενδείξεις στην οθόνη που υποδεικνύουν επαρκή ή ανεπαρκή χώρο. [2]

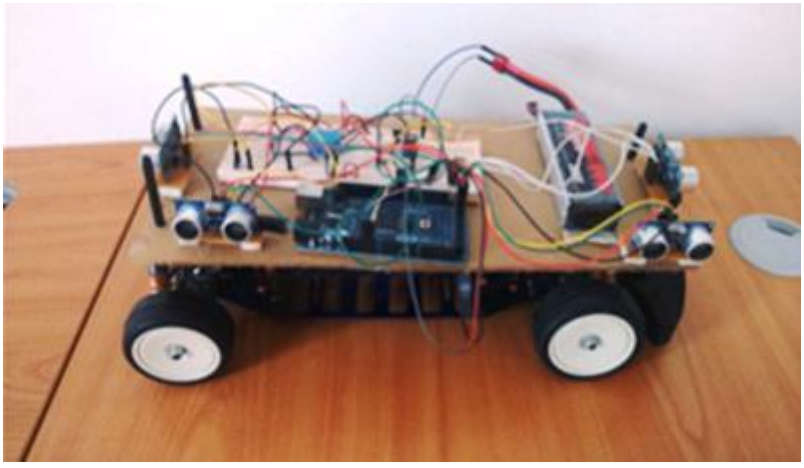
Η Toyota δεν είναι βεβαίως η μόνη αυτοκινητοβιομηχανία που χρησιμοποιεί το σύστημα υποβοήθησης παρκαρίσματος. Ενδεικτικά αναφέρουμε τη Mercedes Benz, τη Ford, τη Volkswagen και πολλές άλλες που έχουν εφοδιάσει κάποια

μοντέλα τους (συνήθως τα πολυτελή) με ένα σύστημα ανάλογων προδιαγραφών. Όπως ήταν αναμενόμενο, τα αποτελέσματα των μελετών των αυτοκινητοβιομηχανιών είναι εμφανή σε κάθε διαφημιστική καμπάνια της εκάστοτε εταιρείας, οι έρευνες όμως και οι μελέτες που οδήγησαν σε αυτά τα αποτελέσματα δεν έχουν κοινοποιηθεί.

2.2 Τα συστήματα προσομοιώσεων

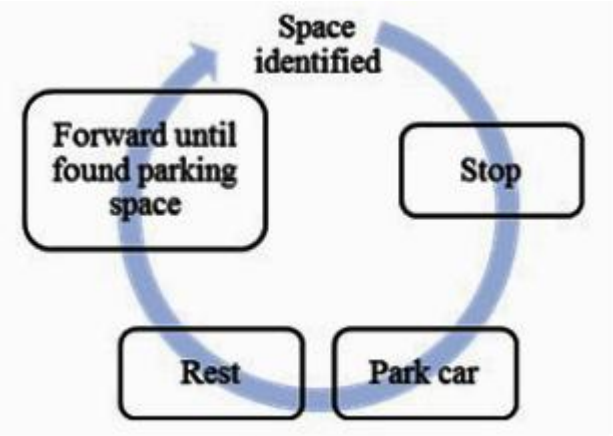
Στον τομέα των προσομοιώσεων υπάρχει πληθώρα μελετών και εργασιών. Η βασική ιδέα παραμένει η ίδια: ένας μικροεπεξεργαστής ελέγχει την κίνηση του οχήματος, λαμβάνοντας σήματα από τους αισθητήρες με τους οποίους αυτός είναι συνδεδεμένος. Αυτά που διαφέρουν είναι το είδος του οχήματος, ο μικροεπεξεργαστής, η πλατφόρμα στην οποία συνδέεται ο μικροεπεξεργαστής, το είδος και το πλήθος των αισθητήρων και σαν συνέπεια όλων των παραπάνω, ο κώδικας για τον έλεγχο του οχήματος. Στις επόμενες παραγράφους αναφέρονται κάποιες από αυτές τις μελέτες.

Στη μελέτη με τίτλο «Development of automated parallel parking system in small mobile vehicle», που δημοσιεύτηκε στο ARPN Journal of Engineering and Applied Sciences, οι Muhammad Faiz Bin Wahab, Aung Lwin Moe, Aminudin Bin Abu, Zulkifli Bin Yaacob και Ari Legowo, παρουσίασαν ένα σύστημα αυτόματου παρκαρίσματος για ένα αυτοσχέδιο όχημα. [4]



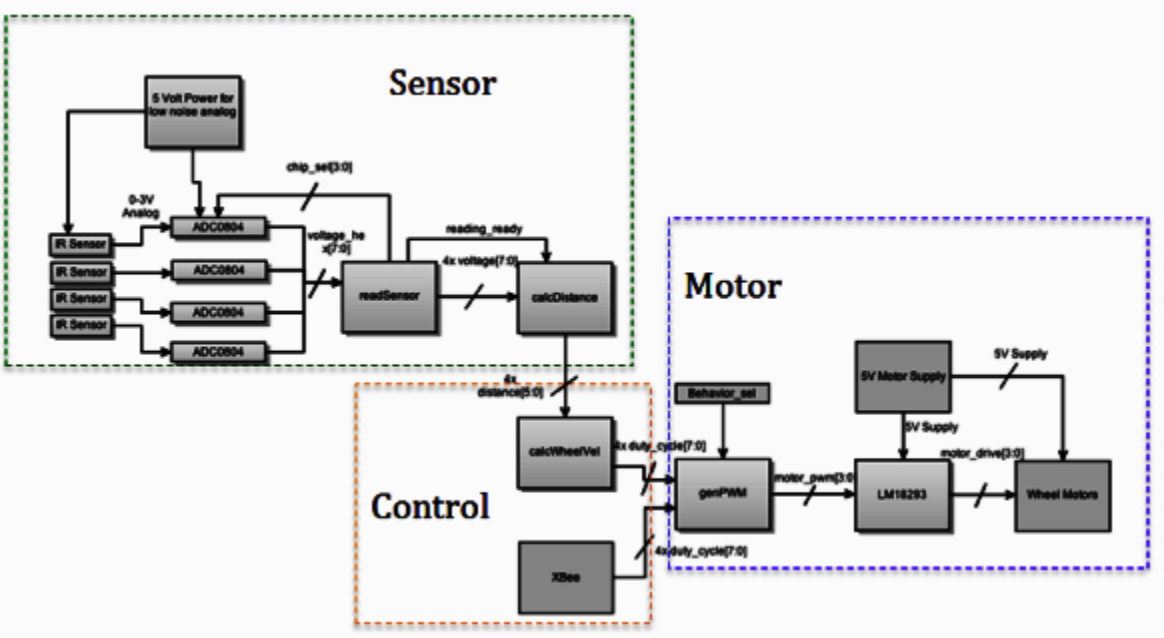
Εικόνα 43 Το αυτοσχέδιο όχημα των Muhammad Faiz Bin Wahab, Aung Lwin Moe, Aminudin Bin Abu, Zulkifli Bin Yaacob και Ari Legowo από το ARPN Journal of Engineering and Applied Sciences. [4]

Την ώθηση στο όχημα έδιναν servo κινητήρες 5V, τους οποίους έλεγχε ο μικροεπεξεργαστής του Arduino Mega. Ο ίδιος μικροεπεξεργαστής έλεγχε και τους αισθητήρες για την εγγύτητα εμποδίων. Η λογική του παρκαρίσματος φαίνεται στο πιο κάτω σχήμα, το οποίο δημοσίευσαν οι συγγραφείς της εργασίας. [4]



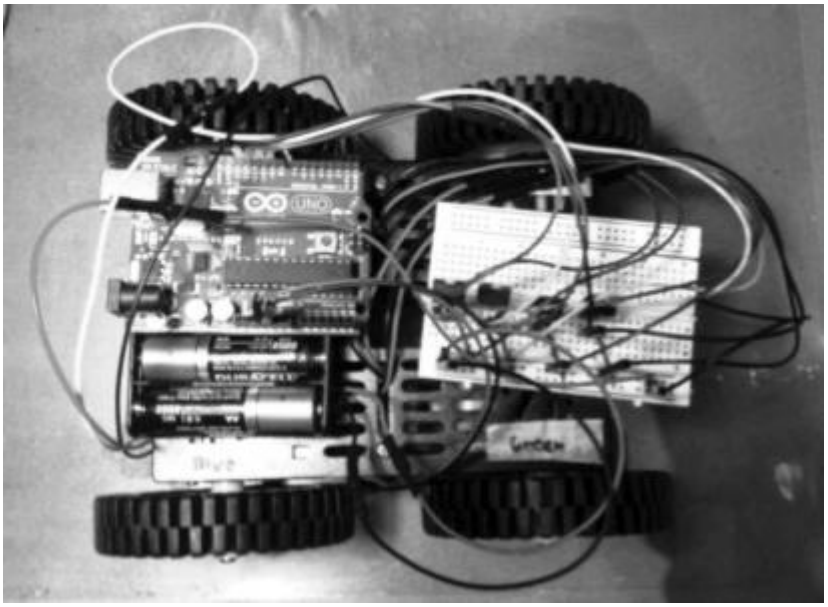
Εικόνα 44 Η λογική του παρκαρίσματος σύμφωνα με τους συγγραφείς της εργασίας. [4]

Μια άλλη εργασία με τίτλο «Self-parking car» των Fei “Frank” Ni και Kevin Hsiue από το M.I.T., φέρνει στο προσκήνιο ένα άλλο σύστημα αυτόματου παρκαρίσματος. Σαν όχημα, οι συγγραφείς της εργασίας «παρκάρουν» το DFRobot 4WD Arduino Mobile Platform, ένα προκατασκευασμένο συναρμολογούμενο αυτοκινητάκι (σασί και τροχοί) που χρησιμοποιείται σε πολλές κατασκευές. «Καρδιά» του όλου συστήματος είναι ο μικροεπεξεργαστής ADC0804 συνδεδεμένος σε ένα FPGA. Ο ADC0804 ελέγχει την κίνηση του οχήματος και τους αισθητήρες υπερύθρων (IR sensors). Η λογική της κατασκευής, σχηματοποιημένη από τους ίδιους τους συγγραφείς της εργασίας**,** φαίνεται στο πιο κάτω σχήμα. [5]



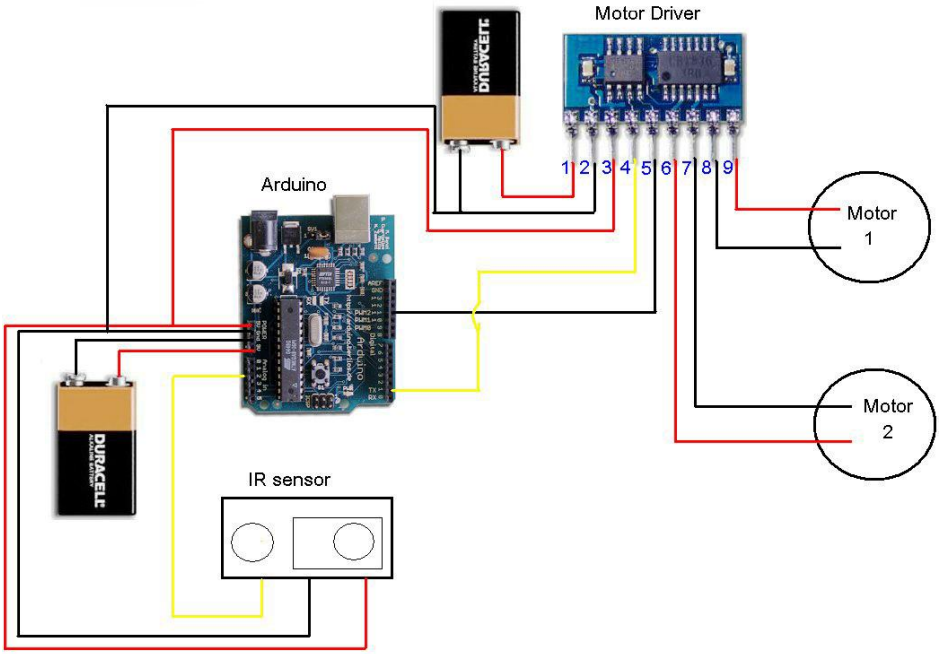
Εικόνα 45 Σχηματικό διάγραμμα των επιμέρους ενοτήτων, από την ίδια την ομάδα του M.I.T. [5]

Στη μελέτη της Claire Chen με τίτλο «Autonomous Parallel Parking Car» το αυτοσχέδιο όχημα έλεγχε η πλακέτα του μικροεπεξεργαστή του Arduino UNO ATmega328. Για τον έλεγχο του χώρου επιλέχθηκαν οι αισθητήρες υπερήχων LV-MaxsSonar-EZ4. Ο κώδικας (sketch) «φορτωνόταν» μέσω της θύρας USB του στο Arduino. [6]



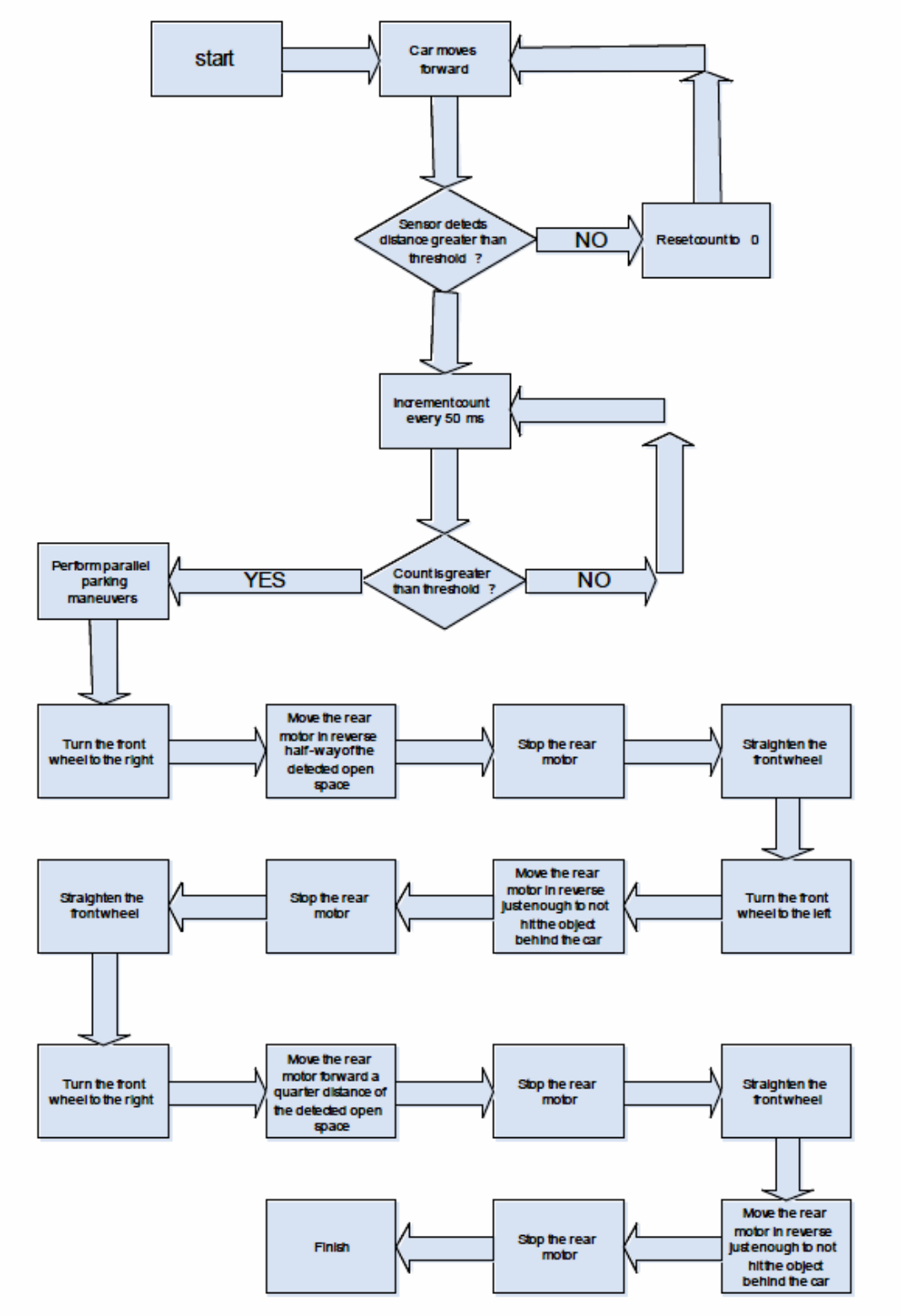
Εικόνα 46 Το όχημα της Claire Chen στη μελέτη της με τίτλο «Autonomous Parallel Parking Car» [6]

Οι Junghoon Ha aka Eric Ha, Stevanes Hermawan και Willie Pramono, στη μελέτη τους με τίτλο «Parallel Parking R/C Car» χρησιμοποίησαν ένα αυτοκινητάκι του εμπορίου, την κίνηση του οποίου έλεγχε το Arduino με σήματα που δέχονταν από έναν αισθητήρα υπερύθρων GP2D15. Η πλακέτα, ο αισθητήρας υπερύθρων και ο κινητήρας τροφοδοτούνταν με δυο μπαταρίες 9V. [7]



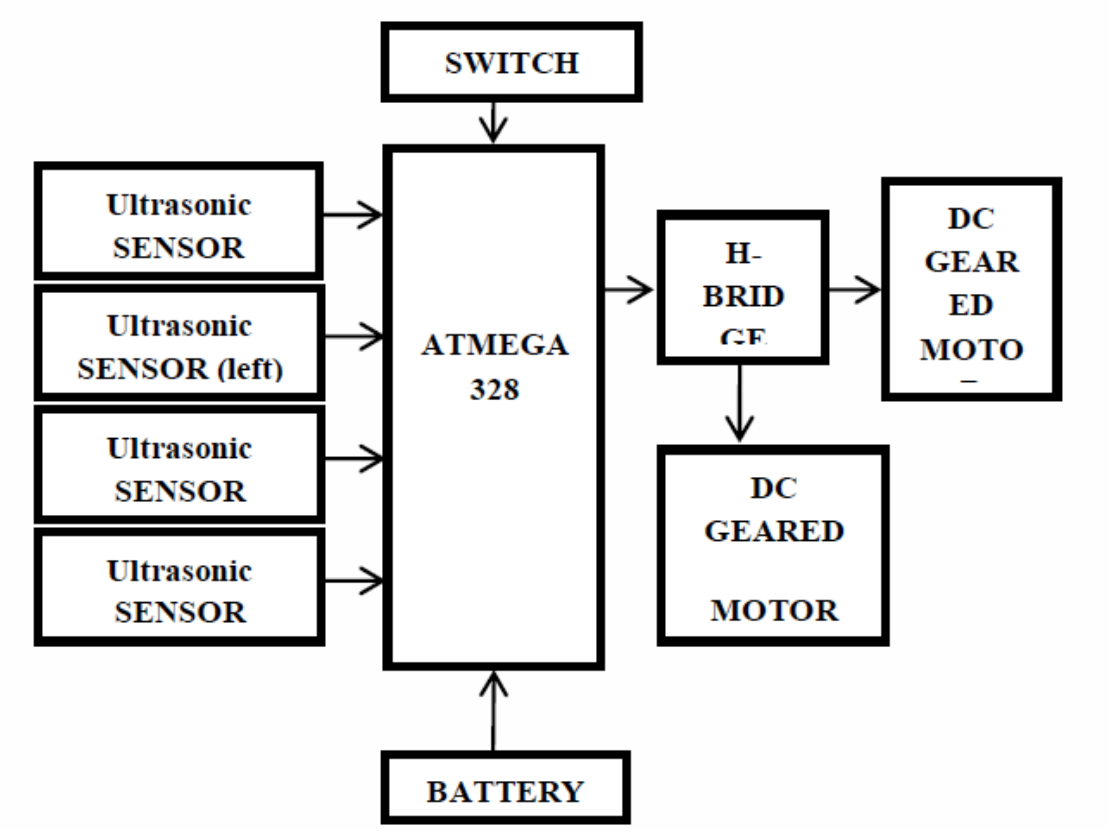
Εικόνα 47 Το κύκλωμα της κατασκευής των Junghoon Ha aka Eric Ha, Stevanes Hermawan και Willie Pramono, στη μελέτη τους με τίτλο «Parallel Parking R/C Car». [7]

Ο αλγόριθμος του αυτόματου παρκαρίσματος φαίνεται στο επόμενο διάγραμμα ροής.



Εικόνα 48 Διάγραμμα ροής της κίνησης του αυτοκινήτου των Junghoon Ha aka Eric Ha, Stevanes Hermawan και Willie Pramono, στη μελέτη τους με τίτλο «Parallel Parking R/C Car» [7]

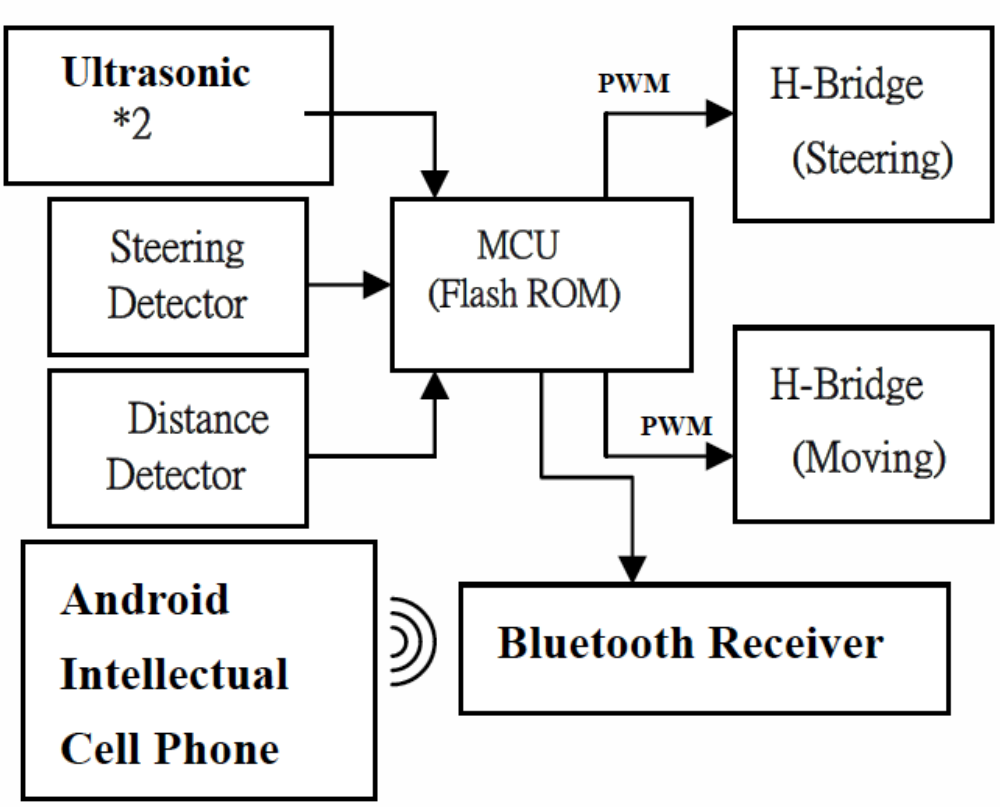
Στη South Asian Journal of Engineering and Technology δημοσιεύτηκε μια μελέτη των I. Pavithra, S. Pradheep Kumar, V. Sharmila, N. K. Senthil και R. Prasanna με τίτλο «An optimal approach for autonomous parallel parking of nonholonomic vehicle», στην οποία παρουσίασαν ένα δικό τους σύστημα αυτόματου παρκαρίσματος. Κεντρικός ελεγκτής αυτού του συστήματος ήταν ο ATmega328 του Arduino UNO. Σχηματικά το κύκλωμα περιγράφεται στην επόμενη εικόνα των ίδιων των συγγραφέων της μελέτης. [8]



Εικόνα 49 Το κύκλωμα της μελέτης των I. Pavithra, S. Pradheep Kumar, V. Sharmila, N. K. Senthil και R. Prasanna με τίτλο «An optimal approach for autonomous parallel parking of nonholonomic vehicle». [8]

Στη μελέτη των Ray_Shine Run, Jui-Cheng Yen και Chihhsuan Chen με τίτλο «A Study of Automatic Parallel Parking System - from the Viewpoints of User and Manufacturer», το κύκλωμα βασίστηκε στον μικροεπεξεργαστή 8-bit 8051, τον Megawin 82G516A με ενσωματωμένο PWM (8 bit) που έλεγχε τους δύο κινητήρες, ένα για την κίνηση και έναν για το τιμόνι. Ο συγκεκριμένος τύπος MCU έχει ενσωματωμένη 64KB Μνήμη Flash, μέρος της οποίας χρησιμοποιείται για την αποθήκευση του κώδικα του προγράμματος του μικροεπεξεργαστή και το άλλο

μέρος της μπορεί να χρησιμοποιηθεί ως μνήμη που απαιτείται από το σύστημα για την αναπαραγωγή της διαδρομής. Στο σύστημα προστέθηκε σύνδεση με μια συσκευή Android. Η συσκευή αυτή έστελνε μέσω Bluetooth την εντολή έναρξης του αυτόματου παρκαρίσματος. Σχηματική παράσταση του κυκλώματος, όπως τη σχεδίασαν οι ίδιοι στη μελέτη τους, παρουσιάζεται στην επόμενη εικόνα. [9]



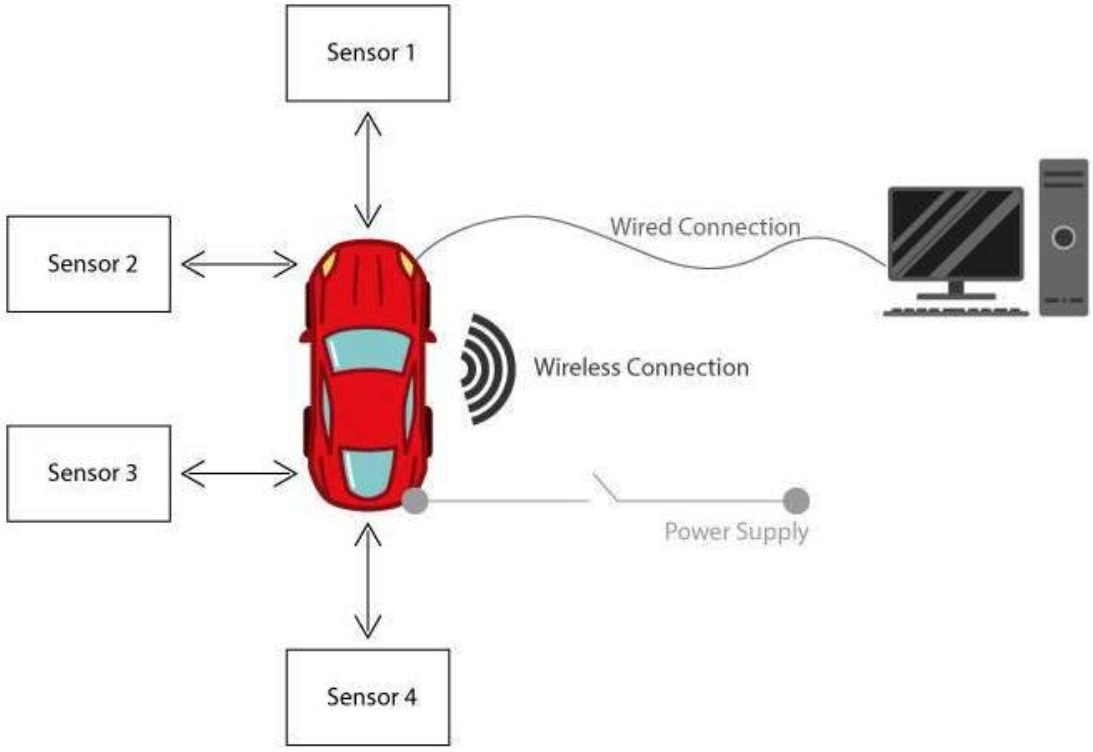
Εικόνα 50 Σχηματικό διάγραμμα του κυκλώματος των Ray_Shine Run, Jui-Cheng Yen και Chihhsuan Chen στην εργασία με τίτλο «A Study of Automatic Parallel Parking System - from the Viewpoints of User and Manufacturer». [9]

2.3 Θεωρητικές μελέτες

Πολλές θεωρητικές μελέτες έχουν εκπονηθεί για τον υπολογισμό των παραμέτρων για το αυτόματο παρκάρισμα. Ενδεικτικά θα αναφέρουμε το «Calculation of optimal path for parallel car parking» των Viktor Zadachyn και Oleksandr Dorokhov, στην οποία, όπως δηλώνει και ο τίτλος, αναπτύσσεται βέλτιστη διαδρομή για το αυτόματο παράλληλο παρκάρισμα [10] και το «Simulate Study of Automatic Parking System», των Tie Wang, Long Qu, Hong Chen και Jing Wu, όπου αναλύονται οι υπολογισμοί για ένα σύστημα αυτόματου παρκαρίσματος. [11]

3. ΥΛΙΚΑ ΚΑΙ ΜΕΘΟΔΟΙ

Η αρχή λειτουργίας του όλου συστήματος είναι (όπως φαίνεται και στην πιο κάτω εικόνα) να τοποθετηθεί σε ένα όχημα προσομοίωσης ένας μικροεπεξεργαστής, ο οποίος είναι συνδεδεμένος με 4 αισθητήρες που θα μεταφέρουν στον μικροεπεξεργαστή σήματα για το αν υπάρχει εμπόδιο και στην περίπτωση που υπάρχει σε ποια απόσταση. Στον μικροεπεξεργαστή είναι, επίσης, συνδεδεμένο το σύστημα ελέγχου των DC κινητήρων (motor shield). Ο μπροστινός κινητήρας διευθύνει τους εμπρός τροχούς. Οι δύο πίσω κινητήρες είναι συνδεδεμένοι σε γέφυρα (H-bridge) (σαν υποτυπώδες διαφορικό) και ρυθμίζουν την κίνηση του αυτοκινήτου. Τέλος, στον μικροεπεξεργαστή είναι ακόμα συνδεδεμένο το εξάρτημα για τη λειτουργία του Bluetooth. Για τη σύνδεση με υπολογιστή, γίνεται χρήση της USB θύρας της κεντρικής πλακέτας.



Εικόνα 51 Σχηματική παράσταση της συνδεσμολογίας των εξαρτημάτων.

3.1. Τα υλικά που επιλέχθηκαν

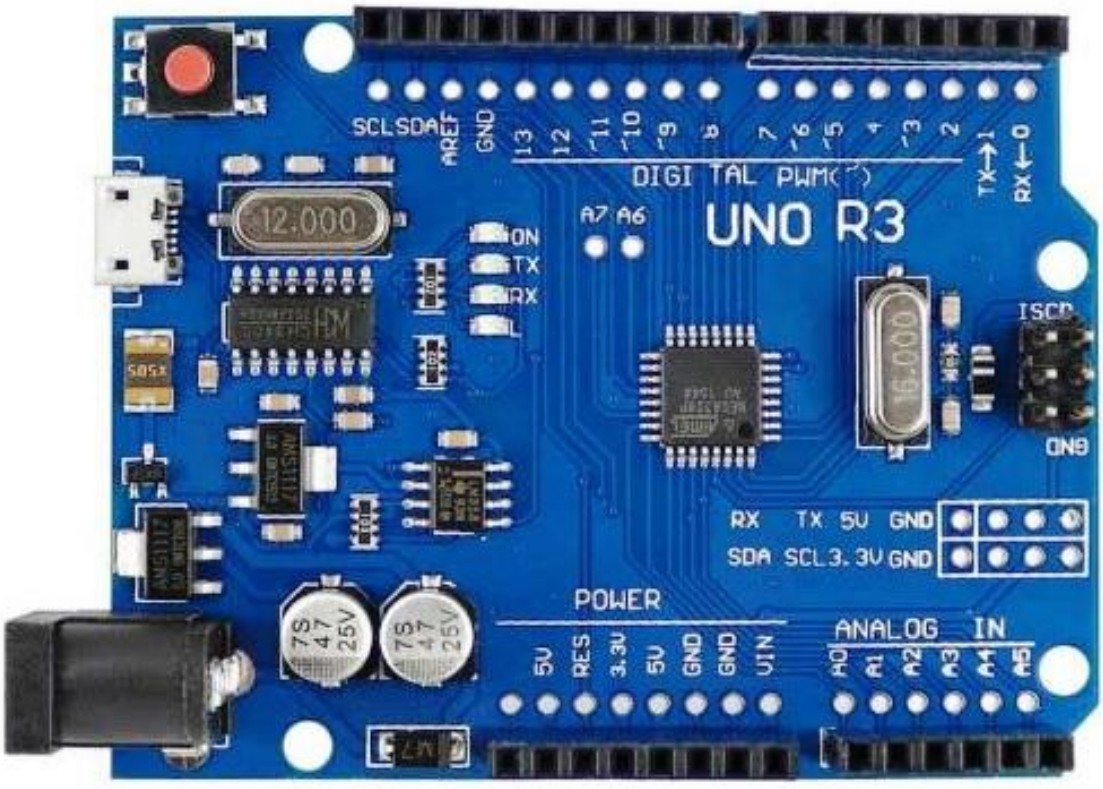
3.1.1 Η κεντρική πλακέτα και ο μικροεπεξεργαστής

Στην επιλογή των εξαρτημάτων, κρίθηκε ότι η κεντρική πλακέτα πρέπει απαραίτητα να διαθέτει θύρα USB. Με αυτόν τον τρόπο διευκολύνεται η διεπαφή με το PC, το οποίο θα εκτελέσει το δυσκολότερο έργο, δηλαδή αυτό της επεξεργασίας των δεδομένων. Για να ελαττωθεί το κόστος του εγχειρήματος, η κεντρική πλακέτα πρέπει να είναι εξοπλισμένη με ενσωματωμένο μετατροπέα ψηφιακού σήματος σε αναλογικό και αναλογικού σε ψηφιακό. Τελευταία απαίτηση για την επιλογή της κεντρικής πλακέτας ήταν να υποστηρίζεται με λογισμικό ανοιχτού κώδικα.

Η λιγότερο δαπανηρή λύση, που να πληροί τις πιο πάνω απαιτήσεις αποδείχτηκε το Arduino Uno.

3.1.2 Το Arduino UNO

«Εγκέφαλος» αυτού του έργου είναι ο μικροεπεξεργαστής ενός Arduino UNO ATmega328.



Εικόνα 52 Η πλακέτα του Arduino UNO ATmega328. [12]

Παρατίθενται εδώ οι προδιαγραφές του κατασκευαστή:

Το Arduino Uno Rev3 βασίζεται στον μικροελεγκτή ATmega328 της Atmel. Είναι μια ολοκληρωμένη πλακέτα που περιέχει ό,τι χρειάζεται για να μπορεί να προγραμματιστεί και να λειτουργήσει συνδέοντάς την με ένα απλό καλώδιο USB σε υπολογιστή ή με ένα τροφοδοτικό στην πρίζα, επίσης μπορεί να λειτουργήσει με απλή μπαταρία. Αναλυτικά, η πλακέτα διαθέτει 14 ψηφιακές εισόδους ή εξόδους (6 από αυτές μπορεί να χρησιμοποιηθούν σαν έξοδοι PWM), 6 αναλογικές εισόδους, 1 θύρα USB (τύπου B) για τον προγραμματισμό και την τροφοδοσία της πλακέτας, 1 είσοδο τροφοδοσίας που μπορεί να χρησιμοποιηθεί εναλλακτικά για τροφοδοσία από τροφοδοτικό πρίζας ή από απλή μπαταρία, 1 υποδοχή ICSP και τέλος κουμπί για το reset της πλακέτας. Ο μικροελεγκτής είναι συγχρονισμένος στους 16 μεγακύκλους (Crystal 16MHz). Σε αυτή την έκδοση του Arduino Uno μικροελεγκτής είναι αποσπώμενος (DIP Version), που δίνει τη δυνατότητα να αντικατασταθεί με έναν καινούργιο με μικρό κόστος σε περίπτωση που καεί. [12]

Η μνήμη Flash του Arduino Uno είναι 32 KB, ικανή να δεχτεί τα περισσότερα απλά προγράμματα. [12]

Το Arduino Uno λειτουργεί με τροφοδοσία 5V DC από την είσοδο του USB ή με 9 V με 12 V DC από την είσοδο της τροφοδοσίας χωρίς να υπάρχει κίνδυνος ηλεκτροπληξίας. [12]

Το Arduino Uno επικοινωνεί απευθείας με τον υπολογιστή μέσω θύρας USB σε συνεργασία με το πρόγραμμα Arduino 1.0 (IDE 1.0). [12]

Τεχνικά Χαρακτηριστικά: [12]

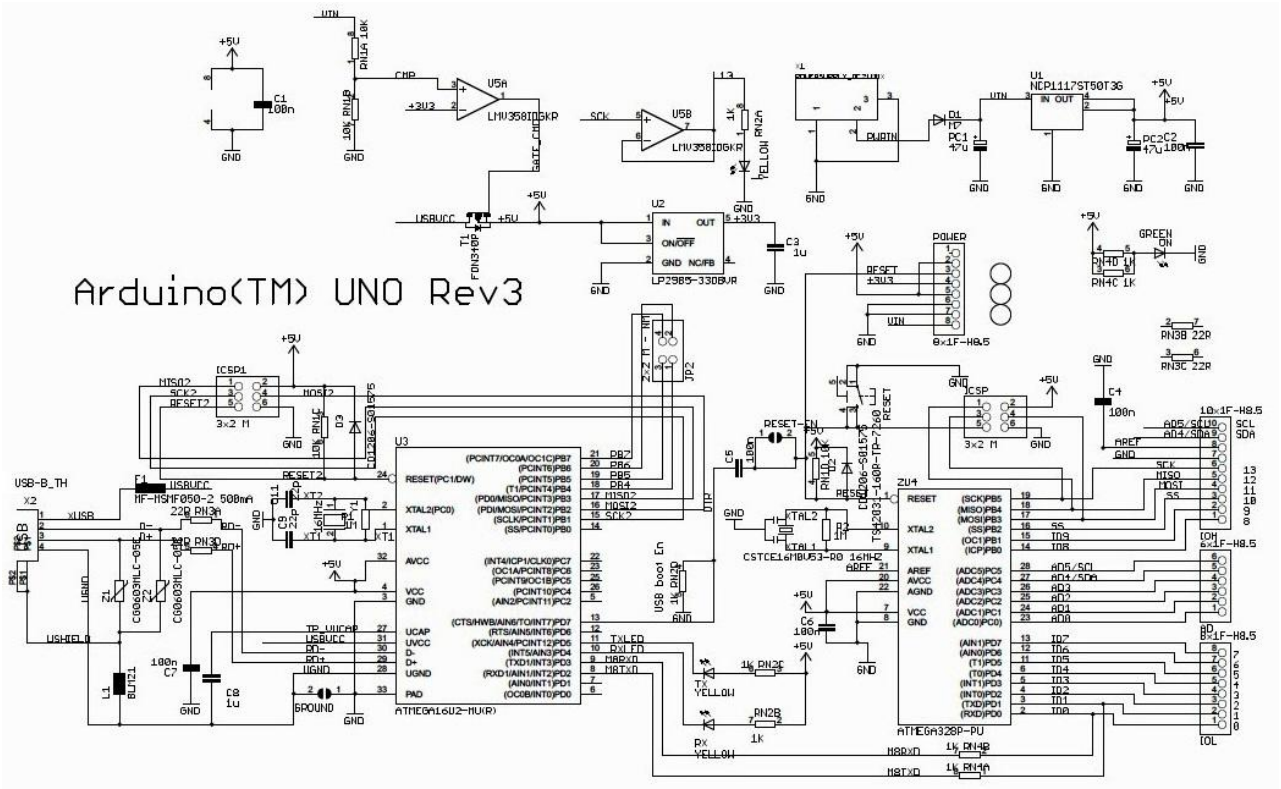
- Μικροελεγκτής ATmega328
- Αρχιτεκτονική ελεγκτή AVR
- Τάση λειτουργίας: 5 VDC
- Τάση εισόδου 7-12 V προτεινόμενη
- Τάση εισόδου 6-20 V limit, min-max
- Ψηφιακές εισόδους/εξόδους 14 (6 εξόδους PWM)
- PWM Ψηφιακές εισόδους/εξόδους 6
- Αναλογικές είσοδοι 6
- Ρεύμα ανά είσοδο / έξοδο 20mA

- Ρεύμα ανά είσοδο / έξοδο 3.3V 50mA
- Μνήμη Flash 32 KB από τα οποία 0.5 KB χρησιμοποιούνται για το σύστημα
- Μνήμη SRAM 2 KB
- Μνήμη EEPROM 1 KB
- Ταχύτητα 16 MHz

Στη παρούσα μελέτη, ο μικροεπεξεργαστής λαμβάνει είσοδο δεδομένων από τους μετρητές υπερήχων (αισθητήρες) και ελέγχει μέσω των κινητήρων τους τέσσερις τροχούς του αυτοκινήτου μέσω διαμόρφωσης πλάτους παλμού.

Ο μικροεπεξεργαστής Arduino «τρέχει» τα λεγόμενα sketch σε γλώσσα επεξεργασίας γραμμένη σε ανοικτό κώδικα «Ολοκληρωμένο περιβάλλον ανάπτυξης». Όταν «κατέβει» για το Arduino, το sketch είναι μεταφρασμένο σε γλώσσα C και περνά από τον avr-gcc compiler, ένα τμήμα λογισμικού που κάνει την τελική μετάφραση σε γλώσσα κατανοητή από τον μικροελεγκτή.

Το Arduino UNO διαθέτει 14 ψηφιακές επαφές εισόδου / εξόδου (pins 0 - 13) που μπορούν να χρησιμοποιηθούν ως είσοδος και ως έξοδος, συμπεριλαμβανομένων των pins για τη διαμόρφωση πλάτους σήματος. Ο μικροελεγκτής τροφοδοτείται είτε μέσω σύνδεσης USB σε έναν υπολογιστή ή με ένα εξωτερικό τροφοδοτικό από 6 έως 20 V. Ωστόσο, περισσότερα από 7 V είναι η βέλτιστη επιλογή για μια σταθερή παροχή 5 V από το pin 5 V. [12]

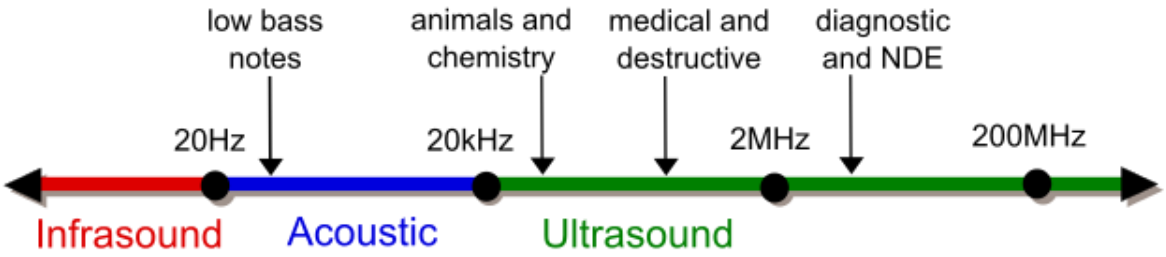


Εικόνα 53 Το PCB Circuit για το Arduino UNO Rev 3. [13]

3.1.3 Οι αισθητήρες υπερήχων

Ο ψηφιακός αισθητήρας υπερήχων παράγει ηχητικά κύματα και διαβάζει την ηχώ τους για την ανίχνευση και τη μέτρηση της απόστασης από τα αντικείμενα. Μπορεί επίσης να στείλει απλά ηχητικά κύματα για να λειτουργήσει σαν sonar ή να ακούσει ένα ηχητικό κύμα που ενεργοποιεί την έναρξη ενός προγράμματος.

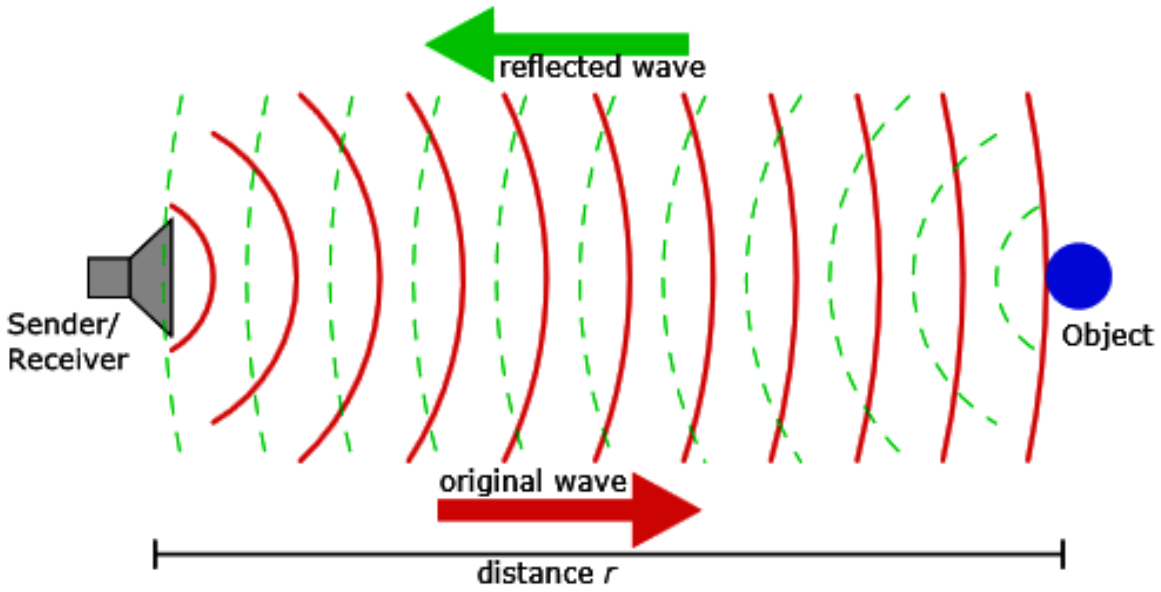
Η επιλογή των αισθητήρων υπερήχων αντί των υπερύθρων βασίστηκε στους εξής λόγους: σε αντίθεση με έναν μετρητή εύρους υπέρυθρων, που λειτουργεί με φως, οι αισθητήρες υπερήχων κάνουν χρήση ήχου, που σημαίνει ότι μπορούν να χρησιμοποιηθούν σε εξωτερικά περιβάλλοντα με έντονο ηλιακό φως. Η διαφορά αυτή γίνεται ιδιαίτερα σημαντική όταν οι αισθητήρες χρησιμοποιηθούν στην πραγματική ζωή για υποβοήθηση στάθμευσης. Το φως αντανακλάται διαφορετικά ανάλογα με την επιφάνεια πρόσπτωσης με αποτέλεσμα η «ανάγνωση» ενός αισθητήρα υπέρυθρων να είναι διαφορετική για τα υλικά, τα οποία διαφέρουν σε ανακλαστικότητα ή/και χρώμα (για παράδειγμα, η περίπτωση ενός λαμπερού αυτοκίνητου) όπως επίσης μιας διαφορετικά φωτιζόμενης επιφάνειας με διαφορετικούς δείκτες απορρόφησης φωτός.



Εικόνα 54 Φάσμα συχνοτήτων ήχου. [14]

Όπως φαίνεται και στην παραπάνω εικόνα, οι υπέρηχοι βρίσκονται πάνω απ' τις ακουστικές συχνότητες, έτσι δεν μπορεί να τους ακούσει το ανθρώπινο αυτί. Παρόλο πάντως που εμείς δεν τους ακούμε κάποια ζώα μπορούν και να τους ακούν αλλά και να τους χρησιμοποιούν. Χαρακτηριστικά παραδείγματα που μας το δείχνουν αυτό είναι η κίνηση των νυχτερίδων και η σφυρίχτρα που χρησιμοποιείται για τους σκύλους. Αισθητήρες υπερήχων συναντάμε σε πολλές εφαρμογές στην ιατρική, στην πλοήγηση σκαφών και πλοίων ακόμα και στα αυτοκίνητα μας στα γνωστά park sensors. [16]

Οι αισθητήρες υπερήχων λειτουργούν με την ίδια αρχή που λειτουργούν τα ραντάρ και τα σόναρ. Εκτιμούν την απόσταση ενός στόχου λαμβάνοντας υπόψη τους την αντανάκλαση ενός ραδιοκύματος ή ενός ηχητικού σήματος πάνω στο στόχο.

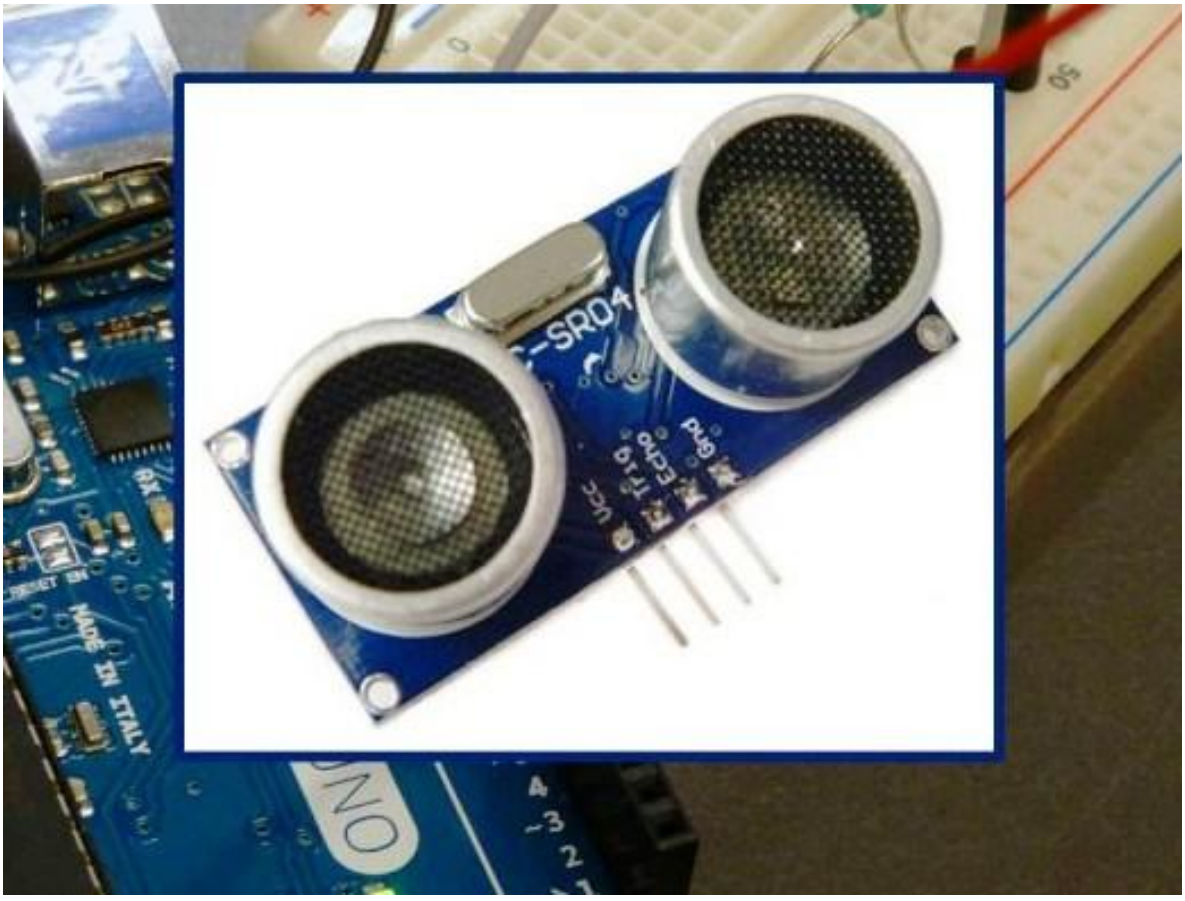


Εικόνα 55 Σχηματική παράσταση της λειτουργίας αισθητήρα υπερήχων. [16]

Οι αισθητήρες υπερήχων δημιουργούν υψηλής συχνότητας κύματα και χρησιμοποιώντας το επιστρεφόμενο σήμα καθορίζουν την απόσταση ή ακόμα και την ταχύτητα του στόχου. Για να το επιτύχουν αυτό χρησιμοποιούν τον χρόνο που έκανε το σήμα για να καλύψει την απόσταση από τον αισθητήρα στο αντικείμενο και πίσω. [16]

Σύμφωνα με το βασικό γνώμονα επιλογής εξαρτημάτων που ορίσαμε για αυτή τη μελέτη, καλύτερη λύση αποτελεί ο αισθητήρας υπερήχων HC-SR04. Ο συγκεκριμένος αισθητήρας πληροί τις απαιτήσεις, και είναι και οικονομικός και ευρείας κατανάλωσης.

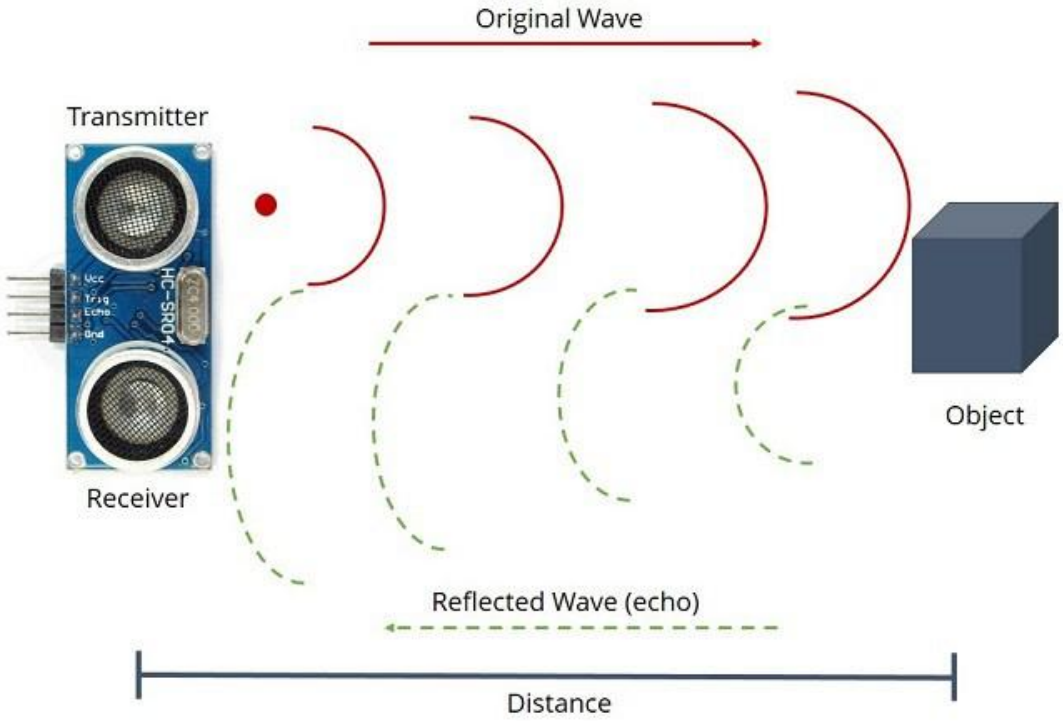
3.1.4 Ο αισθητήρας υπερήχων HC-SR04



Εικόνα 56 Ο αισθητήρας υπερήχων HC-SR04. [15]

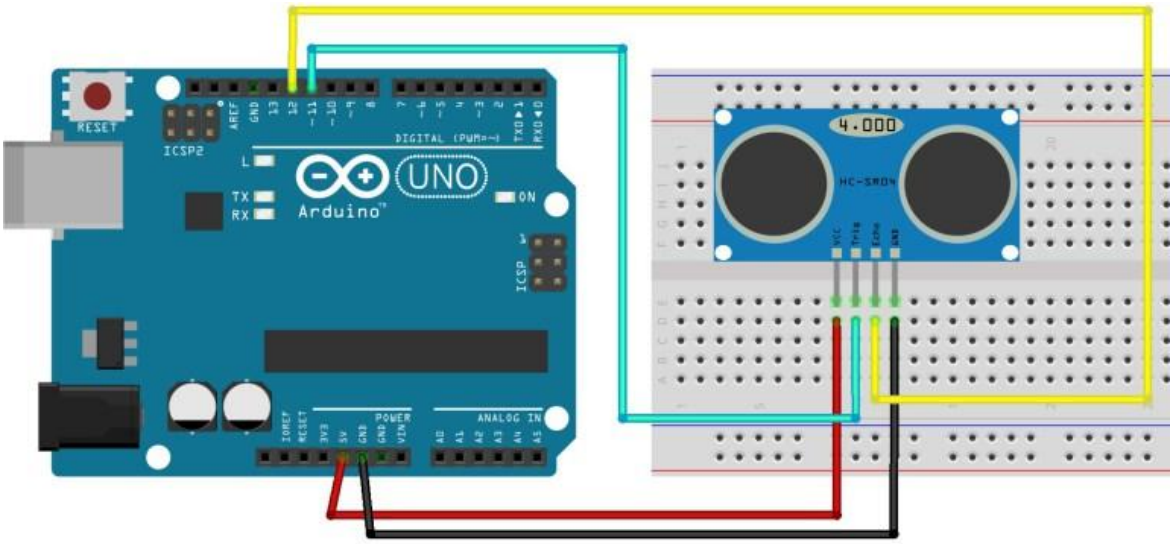
Ο αισθητήρας υπερήχων HC-SR04 παρέχει δυνατότητα κάλυψης 2 - 400 εκατοστών μέτρησης. Η κυμαινόμενη ακρίβεια μπορεί να φτάσει έως 3 mm. Ο αισθητήρας διαθέτει συσκευή αποστολής σημάτων υπερήχων, δέκτη και κύκλωμα ελέγχου. Ο ψηφιακός αισθητήρας υπερήχων παράγει ηχητικά κύματα και διαβάζει

την ηχώ τους για την ανίχνευση και τη μέτρηση της απόστασης από τα αντικείμενα. Μπορεί, επίσης, να στείλει απλά ηχητικά κύματα για να λειτουργήσει σαν sonar ή να ακούσει ένα ηχητικό κύμα που ενεργοποιεί την έναρξη ενός προγράμματος. [15]



Εικόνα 57 Σχηματική παράσταση της λειτουργίας του αισθητήρα υπερήχων HC-SR04. [15]

Στο Arduino η διαδικασία αντιστοίχισης χρόνου και απόστασης είναι αρκετά απλή, διότι υπάρχει συγκεκριμένη βιβλιοθήκη για τον αισθητήρα HC-SR04 που κάνει την "δύσκολη" δουλειά. Στη συγκεκριμένη εφαρμογή αρκεί να προσθέσουμε στον κώδικα μόνο μια εντολή. Η εντολή αυτή θα μας επιστρέψει την απόσταση σε εκατοστά (cm). [16]



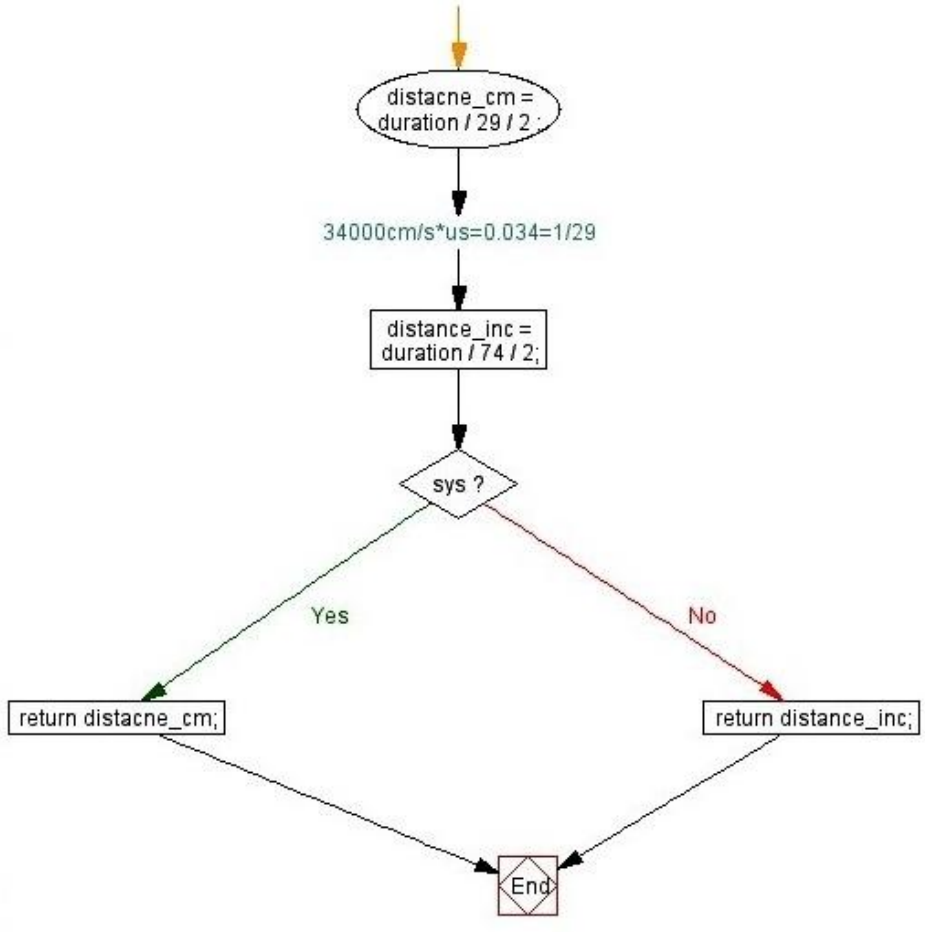
Εικόνα 58 Σύνδεση του αισθητήρα HC-SR04 με το Arduino UNO. [15]

3.1.5 Αρχές λειτουργίας του αισθητήρα HC-SR04

Σύμφωνα με τον κατασκευαστή, [24] οι προδιαγραφές του αισθητήρα είναι οι εξής:

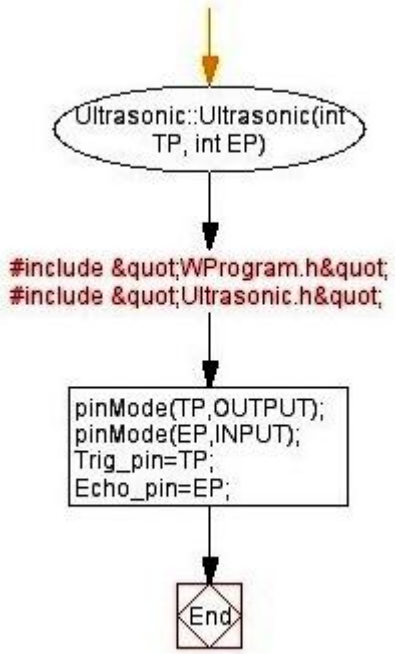
- Τάση λειτουργίας: DC 5V
- Ένταση λειτουργίας: 15mA
- Συχνότητα λειτουργίας: 40Hz
- Μέγιστη εμβέλεια: 4m
- Ελάχιστη εμβέλεια: 2cm
- Γωνία μέτρησης: 15^o
- Σήμα εισόδου ενεργοποίησης: 10μS TTL pulse

Τα ψηφιακά pins του Arduino διαβάζουν και μπορούν να δέχονται είτε υψηλή είτε χαμηλή τάση. Όταν το pin PW του αισθητήρα υπερήχων είναι συνδεδεμένο με έναν ψηφιακό pin στο Arduino, στέλνει πρώτα ένα υψηλής τάσης σήμα και, στη συνέχεια, ένα χαμηλής τάσης μερικά μικροδευτερόλεπτα αργότερα. Η μέθοδος pulseIn() μετρά την ποσότητα του χρόνου σε ένα ψηφιακό pin που είναι ενεργοποιημένο είτε αν είναι υψηλή είτε χαμηλή, ανάλογα με την τιμή που αποστέλλεται.



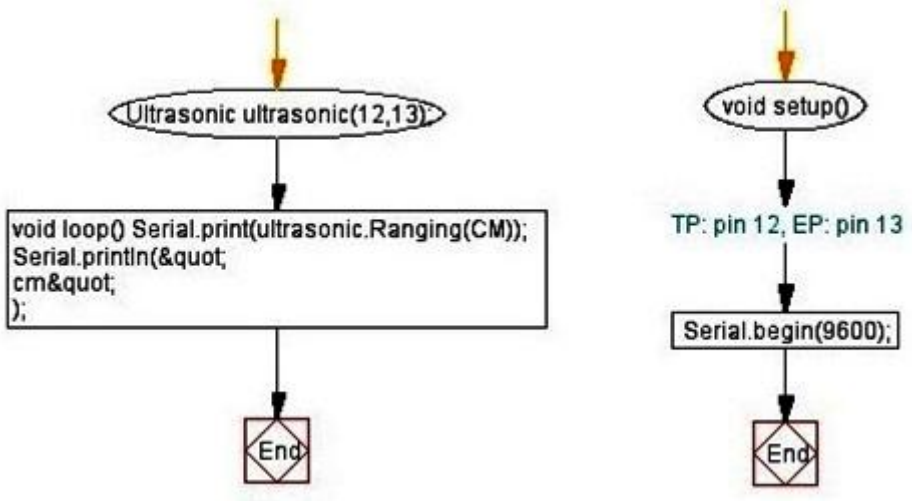
Εικόνα 59 Διάγραμμα ροής για τη λειτουργία του αισθητήρα HC-SR04

Η Βιβλιοθήκη HC-SR04 (Ultrasonic.cpp Ultrasonic.cpp and Ultrasonic.h)

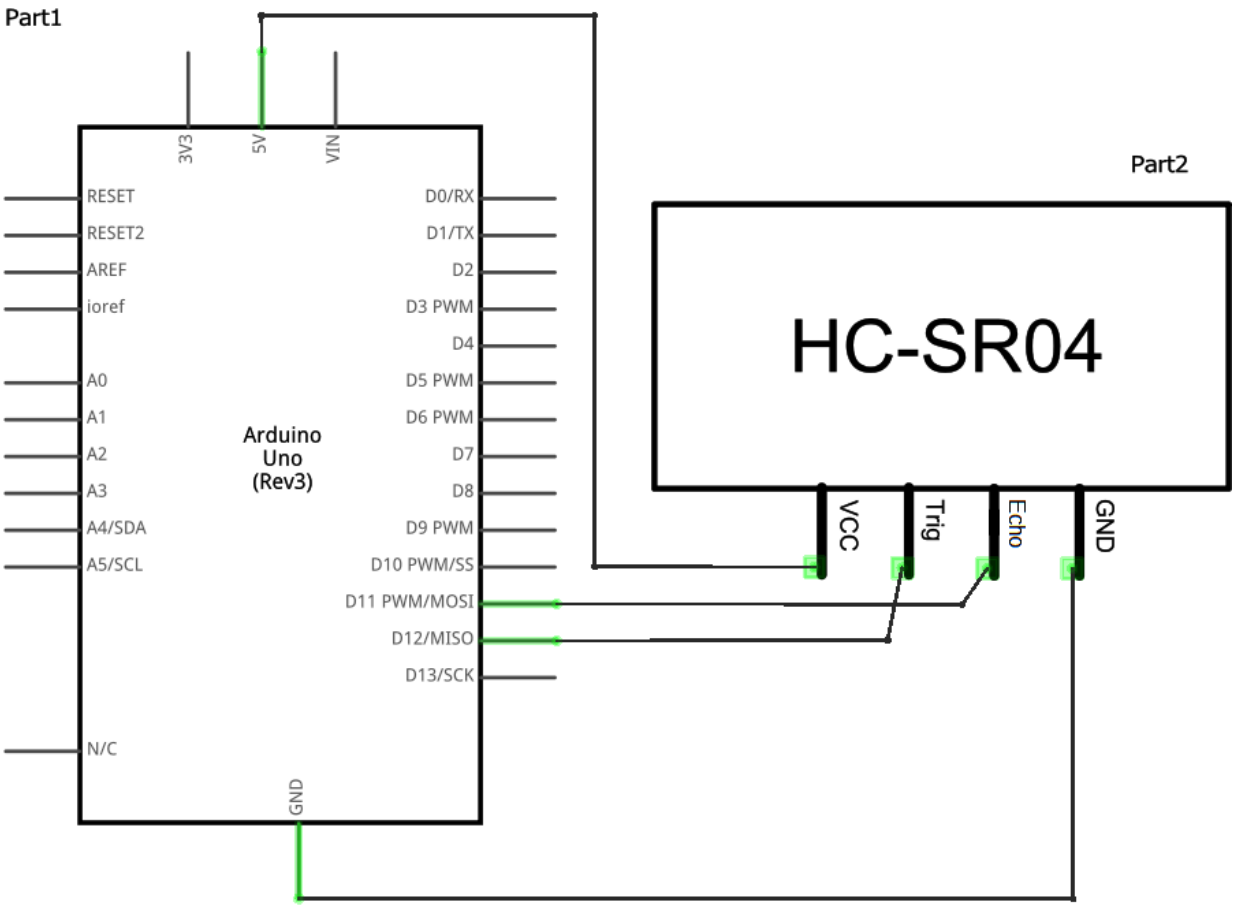


Εικόνα 60 Διάγραμμα ροής για την εισαγωγή της βιβλιοθήκης για τον αισθητήρα HC-SR04 (Ultrasonic.cpp Ultrasonic.cpp and Ultrasonic.h)

HC-SR04 Test Example (Sketch)

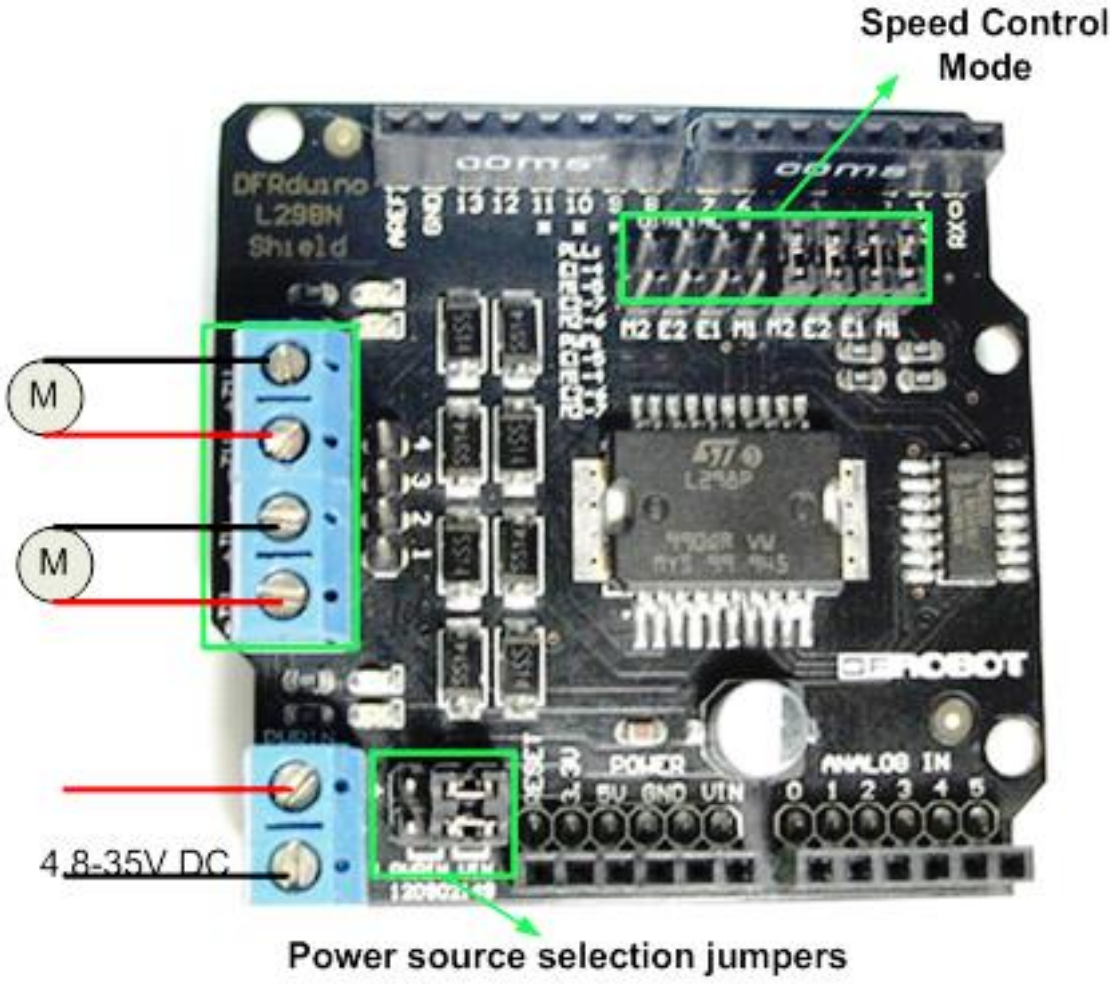


Εικόνα 61 Διάγραμμα ροής για τον έλεγχο της λειτουργίας του αισθητήρα.



Εικόνα 62 Το PCB Circuit για τον αισθητήρα Ultrasonic HC-SR04 και η σύνδεσή του με το Arduino UNO. [17]

3.1.6 Το Arduino Motor Shield (L298N)



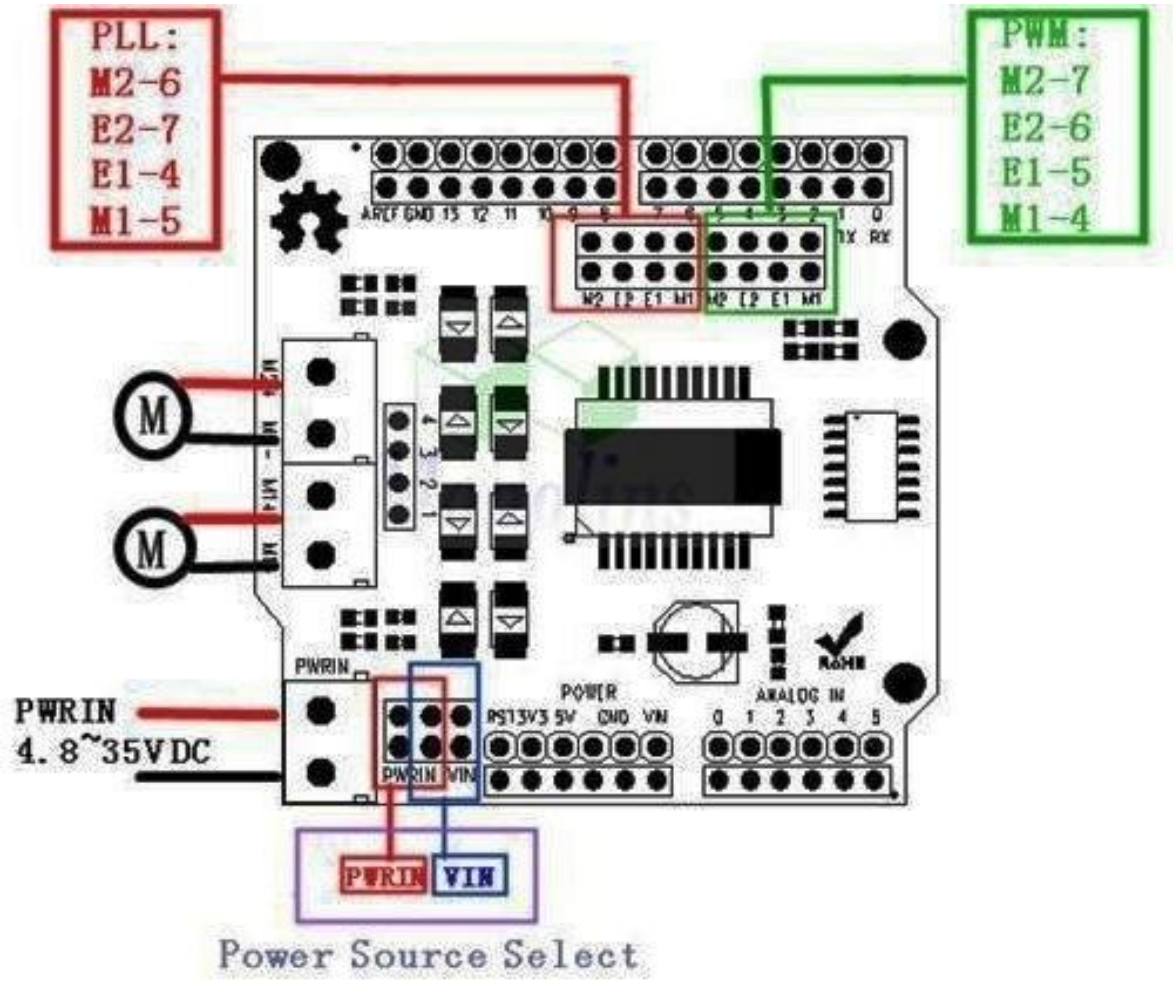
Εικόνα 63 Το Arduino Motor Shield L298N. [18]

Το motor shield επιτρέπει στον μικροελεγκτή του Arduino να «οδηγεί» (ελέγχει) κινητήρες DC δύο καναλιών. Χρησιμοποιεί ένα τσιπ L298N το οποίο έχει δυνατότητα να δέχεται από κάθε κανάλι εξόδου ρεύμα έως 2A. Ο έλεγχος ταχύτητας επιτυγχάνεται μέσω συμβατικών PWM που μπορεί να ληφθεί από το Arduino έξοδος PWM (Pin 5 και 6). Η λειτουργία ενεργοποίησης και απενεργοποίησης του ελέγχου του κινητήρα σηματοδοτείται από το Arduino από τα ψηφιακά Pin 4 και 7. [18]

Το motor shield μπορεί να τροφοδοτηθεί απευθείας από το Arduino ή από εξωτερική πηγή ρεύματος. Η κατασκευάστρια εταιρεία συστήνει στις προδιαγραφές του κινητήρα τη χρήση εξωτερικής τροφοδοσίας, πρόταση που υιοθετήθηκε στην παρούσα μελέτη. [18]

Οι εργοστασιακές προδιαγραφές του L298N motor shield, σύμφωνα με τον κατασκευαστή είναι οι εξής: [18]

- Logic Control Voltage：5V (From Arduino)
- Motor Driven Voltage：4.8～35V (From Arduino or External Power Source)
- Logic supply current Iss：≤36mA
- Motor Driven current Io：≤2A
- Maximum power consumption：25W（T=75℃）
- PWM, PLL Speed control mode
- Control signal level:
  - High：2.3V≤Vin≤5V
  - Low：-0.3V≤Vin≤1.5V



Εικόνα 64 Το Arduino Motor Shield L298N.

3.1.7 Οι λειτουργίες ελέγχου κινητήρων

Το shield υποστηρίζει δύο (PWM και PLL (Phased Locked Loop)) λειτουργίες ελέγχου κινητήρων. Η PWM λειτουργία χρησιμοποιεί τα E1 και E2 για να δημιουργεί σήμα PWM. [18]

Η λειτουργία PLL χρησιμοποιεί τα M1 και M2 για να δημιουργήσει σήμα ελέγχου. [18]

Στη συγκεκριμένη μελέτη έγινε χρήση της PWM.

3.1.8 Οι κινητήρες

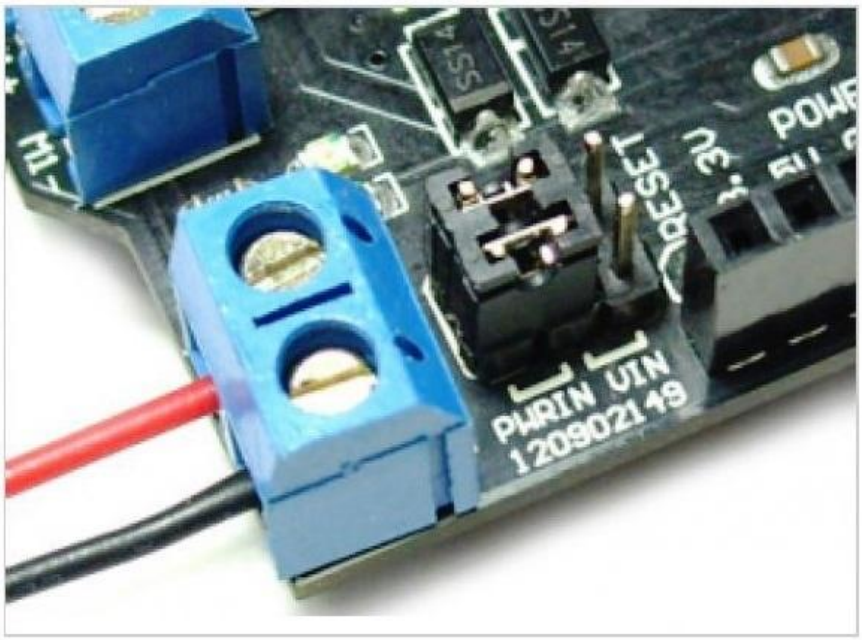
Οι δύο DC κινητήρες συνεχούς ρεύματος συνδέονται στους μπλε ακροδέκτες κινητήρων που διαθέτει το motor shield.

PWRIN: Οι κινητήρες μπορούν να τροφοδοτηθούν με εξωτερική παροχή ηλεκτρικού ρεύματος όταν οι προδιαγραφές των χρησιμοποιούμενων κινητήρων υπερβαίνουν τους περιορισμούς που προβλέπονται από τον Arduino UNO. Ο τρόπος εναλλαγής παροχής ανάμεσα σε εξωτερική πηγή ή από τον Arduino υλοποιείται από δύο γέφυρες βραχυκύκλωσης:

PWRIN: External Power

VIN: Arduino Power

Στην εικόνα φαίνεται η επιλογή μας για εξωτερική πηγή τροφοδοσίας των κινητήρων της μελέτης.



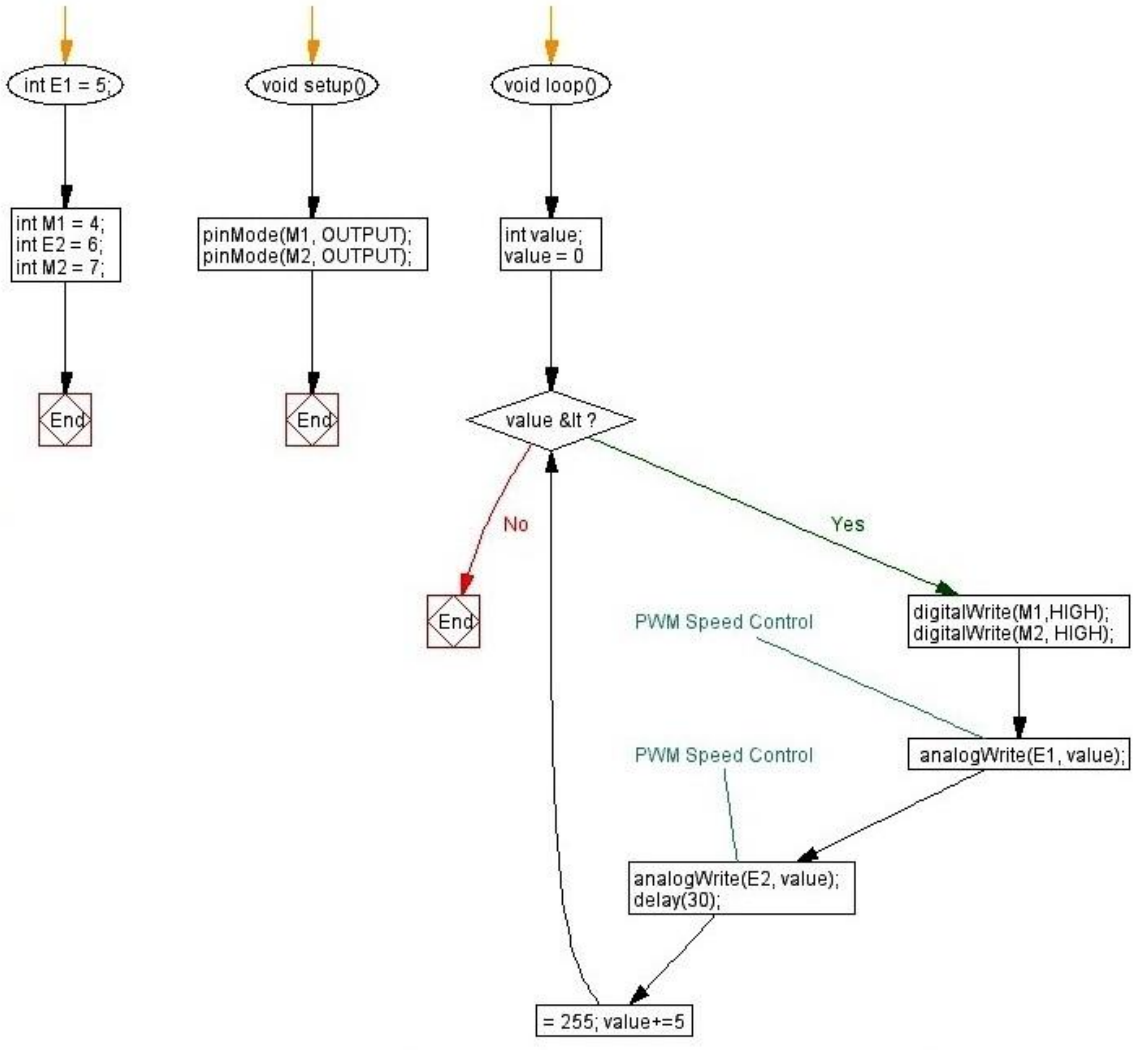
Εικόνα 65 Σύνδεση των κινητήρων με εξωτερική πηγή τροφοδοσίας. [18]

Η διάταξη διασύνδεσης των pin στον Arduino που επιλέχτηκε για το project απεικονίζεται στον παρακάτω πίνακα.

Pin	**Λειτουργία**
Digital 4	Έλεγχος διεύθυνσης κινητήρα 2
Digital 5	Έλεγχος PWM κινητήρα 2
Digital 6	Έλεγχος PWM κινητήρα 1
Digital 7	Έλεγχος διεύθυνσης κινητήρα 1

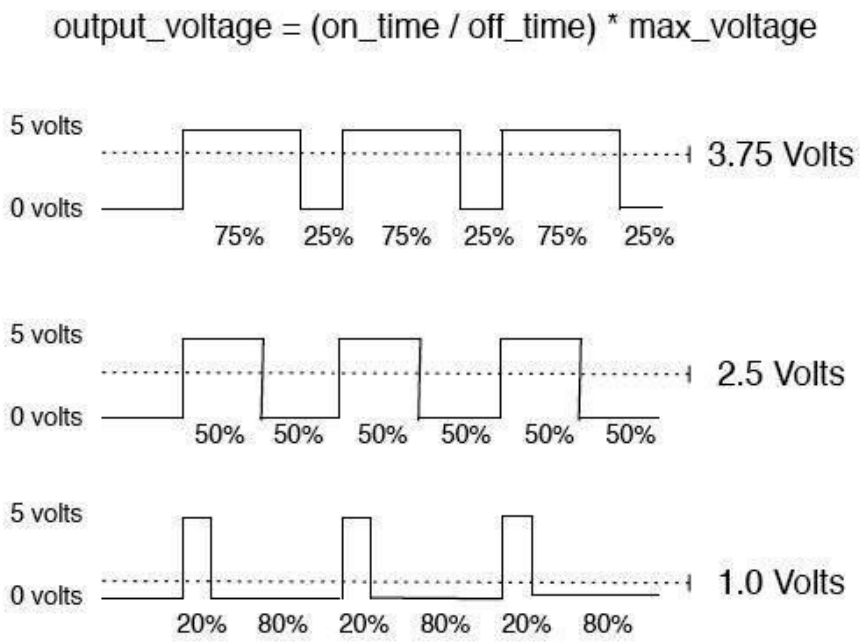
Πίνακας 2 Κατανομή των pins [18]

3.1.9 Έλεγχος ταχύτητας PWM

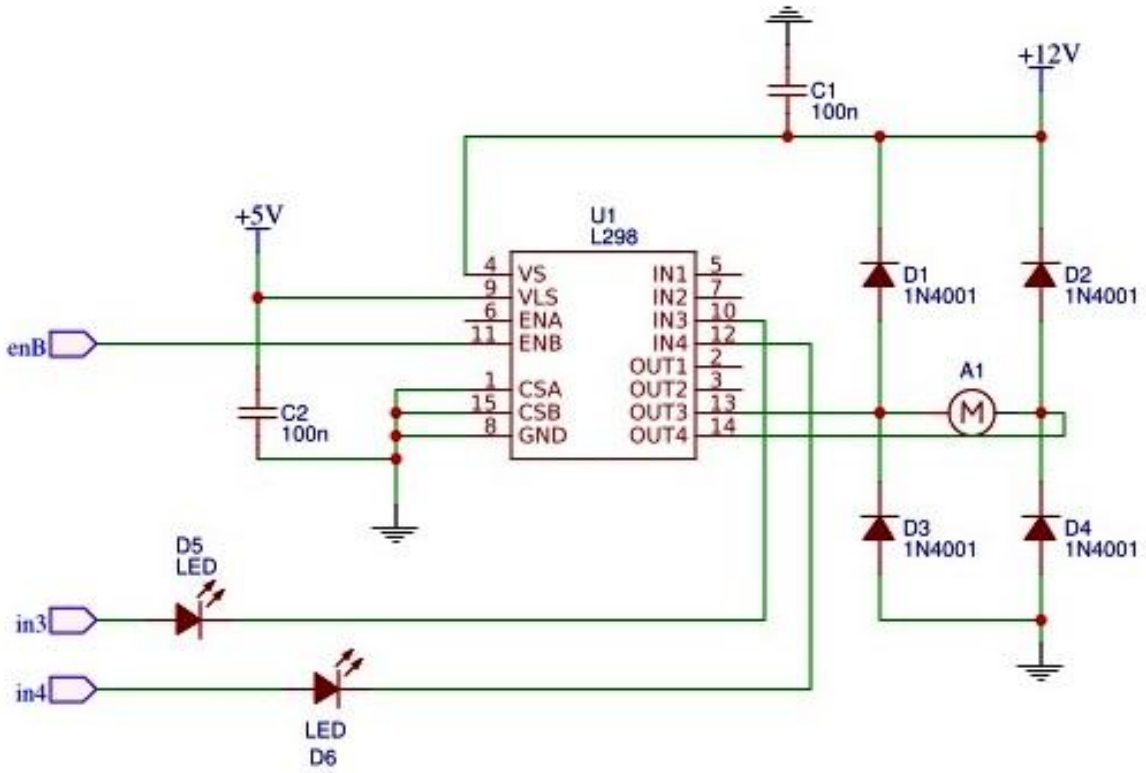


Εικόνα 66 Διάγραμμα ροής για τον έλεγχο ταχύτητας PWM

Η διάταξη PWM ελέγχου ταχύτητας χρησιμοποιείται για την προσομοίωση διαφορετικής εφαρμόζουσας τιμής τάσης (Volt) με τη ρύθμιση του αέρα για τον έλεγχο της τάσης που εφαρμόζεται στον κινητήρα ώστε να επιτευχθεί η επιθυμητή ταχύτητα. Η διαμόρφωση αυτή επιτυγχάνεται στον παλμό του σήματος σύμφωνα με την παρακάτω εικόνα για διαφορετικές τάσεις (V).



Εικόνα 67 Ο παλμός του σήματος για διαφορετικές τάσεις. [19]



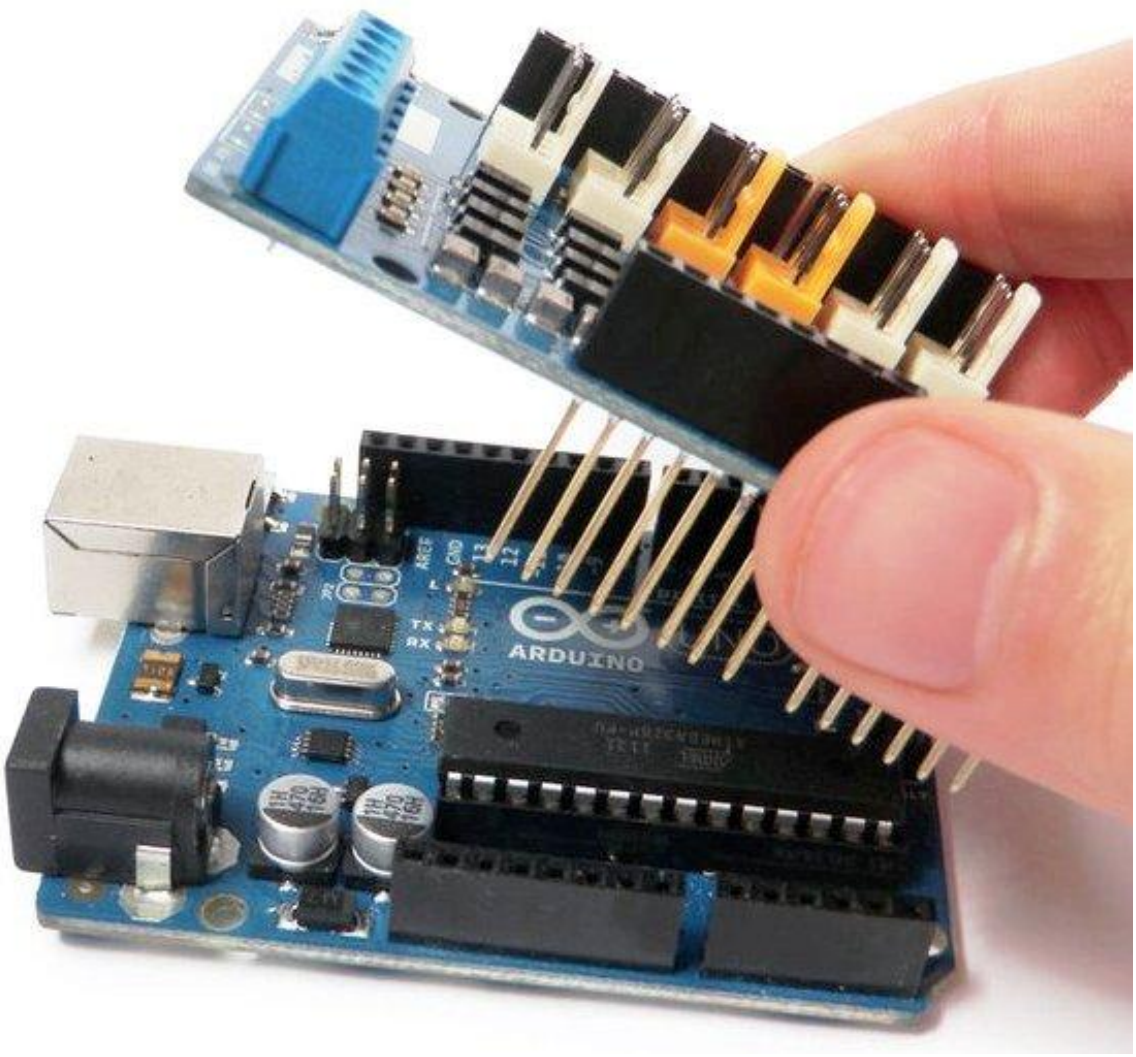
Εικόνα 68 Το PCB Circuit για το Motor Shield L298N.

3.1.10 Για τη λειτουργία των DC κινητήρων

Οι δύο DC κινητήρες συνδέονται στο motor shield και σ' αυτό θα μεταφέρονται οι εντολές για να πηγαίνει το αυτοκινητάκι μπροστά, πίσω και να σταματάει μετά από μερικά δευτερόλεπτα.

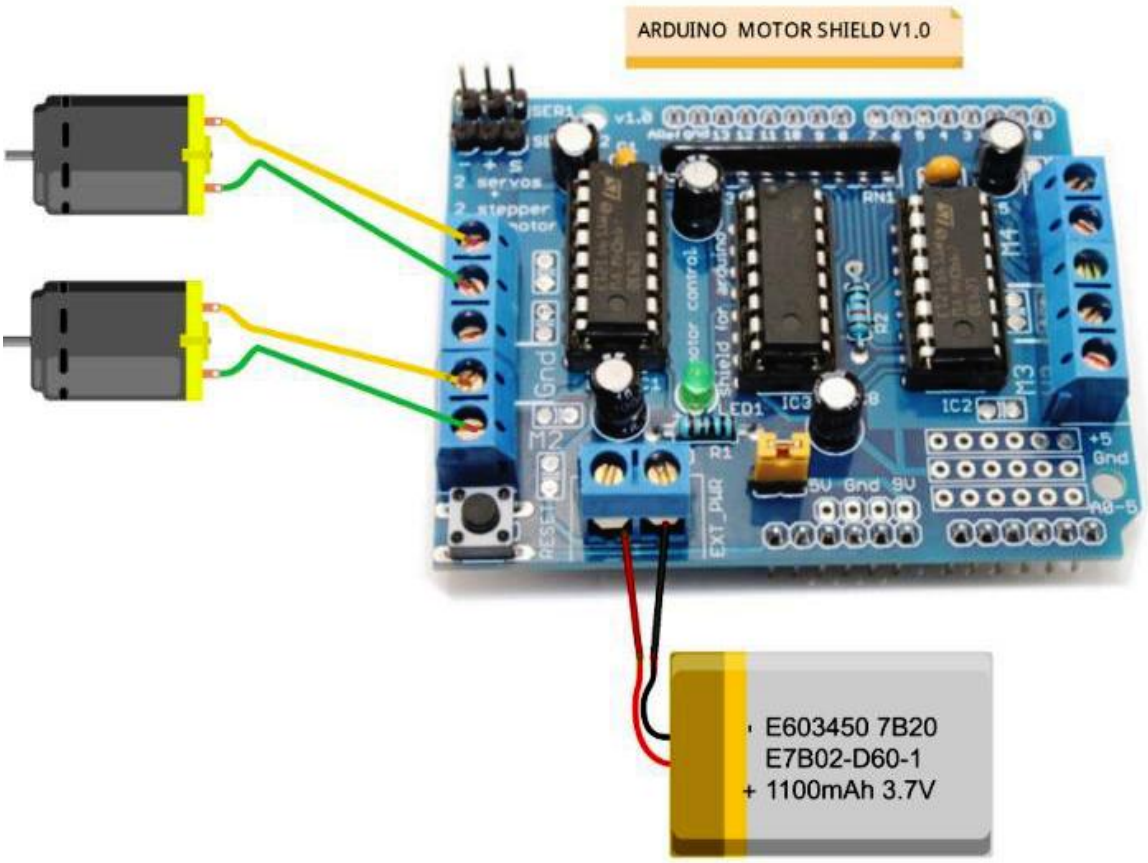
Το motor shield είναι μία ασπίδα που χρησιμοποιείται για να ελέγξει μέχρι 2 DC μοτέρ, 2 servos ή 1 βηματικό μοτέρ. Τροφοδοτείται με ρεύμα από το Arduino αλλά και με σύνδεση στους ακροδέκτες +M (θετικό) και GND (γείωση).

Ο ελεγκτής κινητήρων L298N τοποθετείται πάνω στο Arduino, σύμφωνα με την παρακάτω εικόνα και πάνω του συνδέονται οι 2 κινητήρες.



Εικόνα 69 Σύνδεση του ελεγκτή κινητήρων L298N πάνω στο Arduino. [20]

Οι ηλεκτροκινητήρες συνεχόμενης τάσης απαιτούν τάση της 3 – 6 V με αντίστοιχο φορτίο ρεύματος 200 mA και μέγιστη ταχύτητα ανά τροχό 65 rpm.



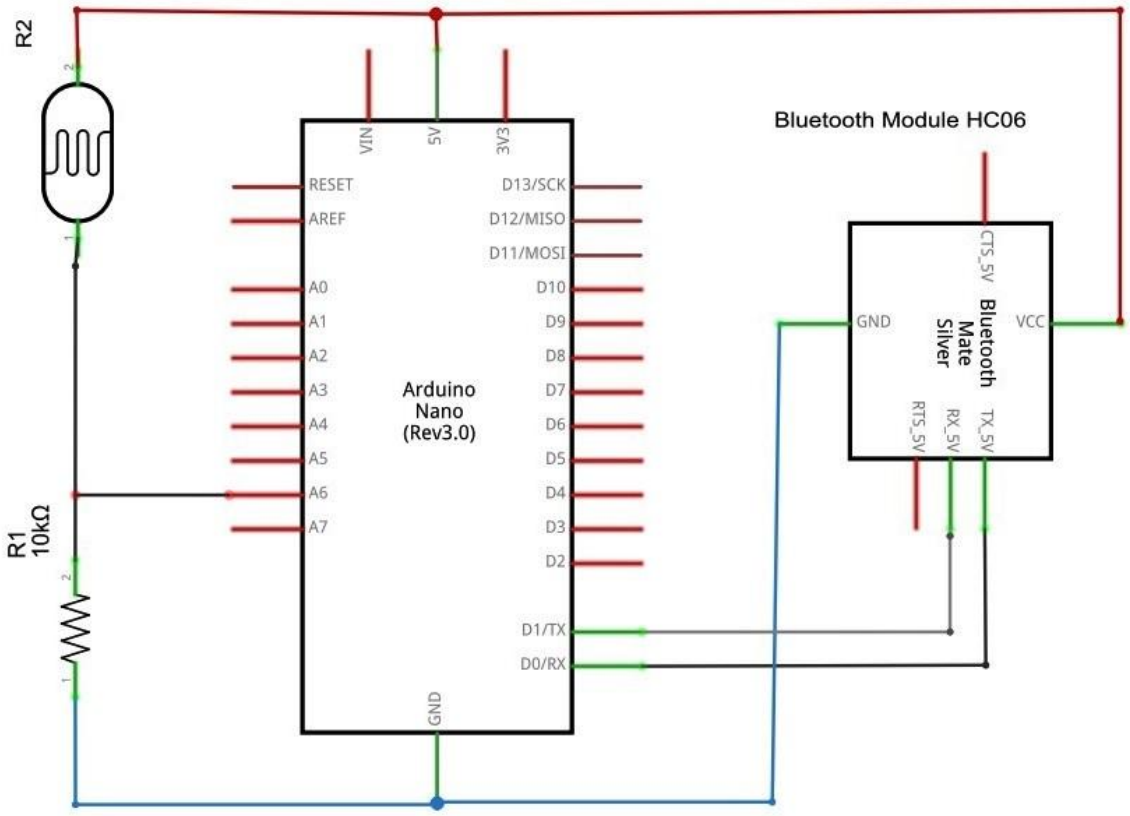
Εικόνα 70 Σύνδεση των κινητήρων και της τροφοδοσίας με το Arduino UNO. [21]

3.1.11 Η συσκευή Bluetooth HC-06

Η συσκευή Bluetooth επιτρέπει να αποστέλλονται ή να λαμβάνονται δεδομένα TTL μέσω τεχνολογίας Bluetooth στον υπολογιστή ασύρματα, χωρίς τη χρήση σειριακού καλωδίου.

Τα χαρακτηριστικά της συσκευής που χρησιμοποιήθηκε στο αυτοκίνητό της μελέτης είναι τα εξής: [23]

- Προκαθορισμένο Baud Rate: 9600,8,1,n.
- Built in antenna.
- Coverage up to 30ft.
- Bluetooth version: V2.0+EDR
- Τάση λειτουργίας: 3.3V



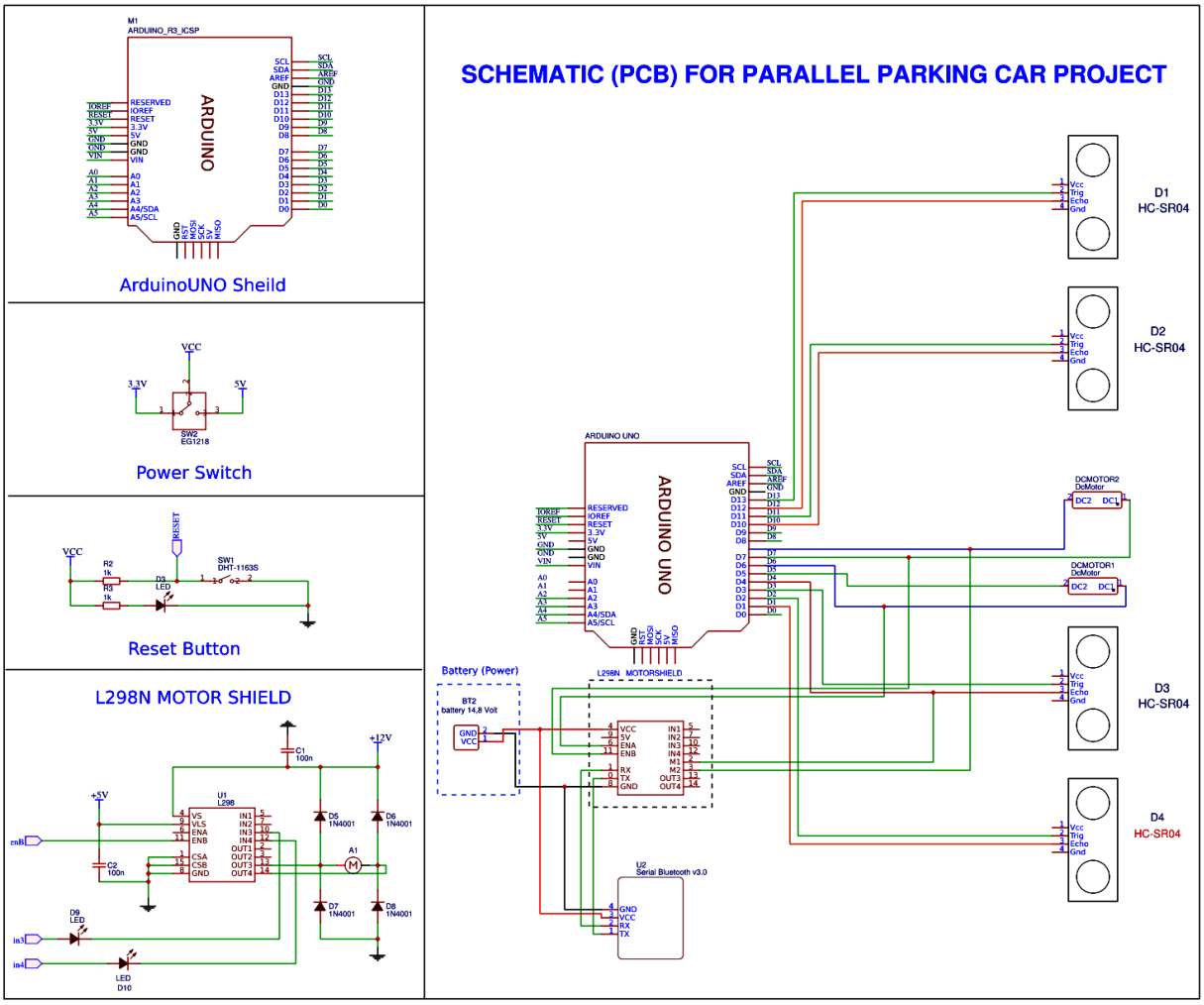
Εικόνα 71 Το PCB Circuit για το συσκευή Bluetooth HC-06 και η σύνδεσή της με το Arduino UNO.

4. ΥΛΟΠΟΙΗΣΗ ΤΟΥ PROJECT

Οι πρώτες δοκιμές για να κατασκευαστεί ή να βρεθεί ένα έτοιμο αυτοκινητάκι δεν απέδωσαν καρπούς. Το βασικό πρόβλημα που συνεχώς παρουσιαζόταν ήταν στους κατευθυντήριους τροχούς: δεν έστριβαν αρκετά με αποτέλεσμα η απόσταση που απαιτούσε το παρκάρισμα να είναι τεράστια. Τελικά, για την συγκεκριμένη κατασκευή, τη λύση έδωσε ένα τηλεκατευθυνόμενο αυτοκίνητο της αγοράς, το οποίο διαθέτει 2 κινητήρες DC με κίνηση κατεύθυνσης εμπρός και πίσω από τον οπίσθιο κινητήρα και δυνατότητα περιστροφικής κίνησης δεξιά-αριστερά από τον εμπρόσθιο κινητήρα, με την βοήθεια ενσωματωμένου συστήματος περιστροφής. Η πρώτη μας κίνηση ήταν να αφαιρεθεί ο μηχανισμός τηλεκατεύθυνσης. Επόμενη κίνηση ήταν η εγκατάσταση των εξαρτημάτων για το αυτόματο παρκάρισμα.

Τα υλικά που χρησιμοποιήθηκαν ήταν:

- Ένα (1) Arduino Uno V3
- Ένα (1) Motor Shield L298N
- Μία (1) Μπαταρία Lip ο 14,8 V, 1400 mhA
- Τέσσερεις (4) Ultrasonic HC-04 αισθητήρες υπέρηχων
- Μία (1) συσκευή Bluetooth HC-06
- Jumpers καλώδια
- Ένα (1) Breadboard για την διεπαφή του κυκλώματος με εξωτερική πηγή ρεύματος
- Ένα (1) Power bank 6000 mhA για τροφοδοσία του Arduino



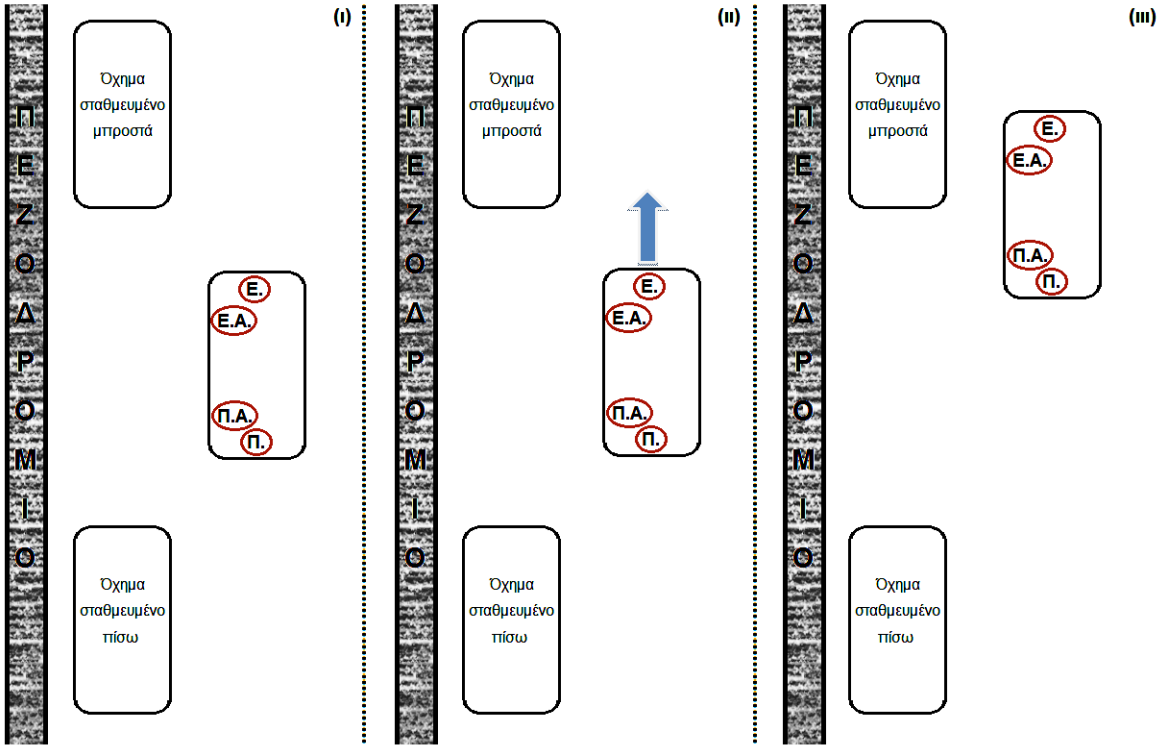
Εικόνα 72 Το PCB Circuit για το σύστημα αυτόματου παρκαρίσματος.

Μετά από πολλές δοκιμές, το αυτοκινητάκι απέδειξε ότι ανταποκρίνεται κατάλληλα στις διάφορες εξωτερικές συνθήκες. Στη συνέχεια, «φορτώσαμε» τον αλγόριθμο (sketch). Ο αλγόριθμος αυτόματου παρκαρίσματος είναι γραμμένος σε περιβάλλον Arduino IDE. Στο επόμενο κεφάλαιο περιγράφονται τα δεδομένα με βάση τα οποία γράφηκε ο αλγόριθμος.

4.1 Δεδομένα για τη συγγραφή του αλγόριθμου

Κίνηση 1: Αναζητώντας θέση παρκαρίσματος

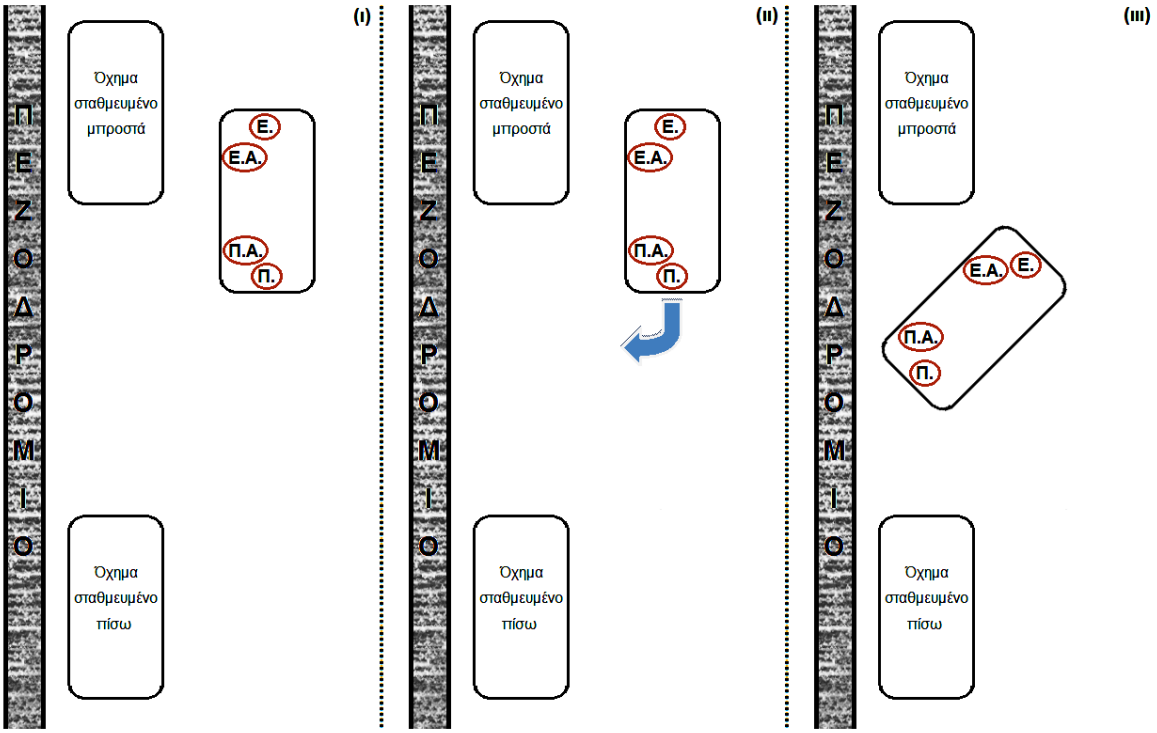
- Προϋπόθεση: Το αυτοκίνητο «ενεργοποιείται» έχοντας από την αριστερή πλευρά του και παράλληλα με το πεζοδρόμιο σταθμευμένα αυτοκίνητα. Το αυτοκίνητο βρίσκεται μπροστά από το χώρο στάθμευσης, ανάμεσα σε δύο σταθμευμένα αυτοκίνητα.
- Εκτέλεση κώδικα: Το αυτοκίνητο αρχίζει να κινείται ευθεία προς τα εμπρός. Ο εμπρός αριστερά αισθητήρας (Ε.Α.) καθώς και ο πίσω αριστερά αισθητήρας (Π.Α.) ανιχνεύουν αποστάσεις λιγότερο από 30 cm. Το αυτοκίνητο κινείται, μέχρι ο εμπρός αριστερά αισθητήρας να βρει απόσταση μικρότερη των 30 cm.



Εικόνα 73 Σχηματική παράσταση της αναζήτησης θέσης παρκαρίσματος.

Κίνηση 2: Εκτέλεση κίνησης οπισθογωνίας

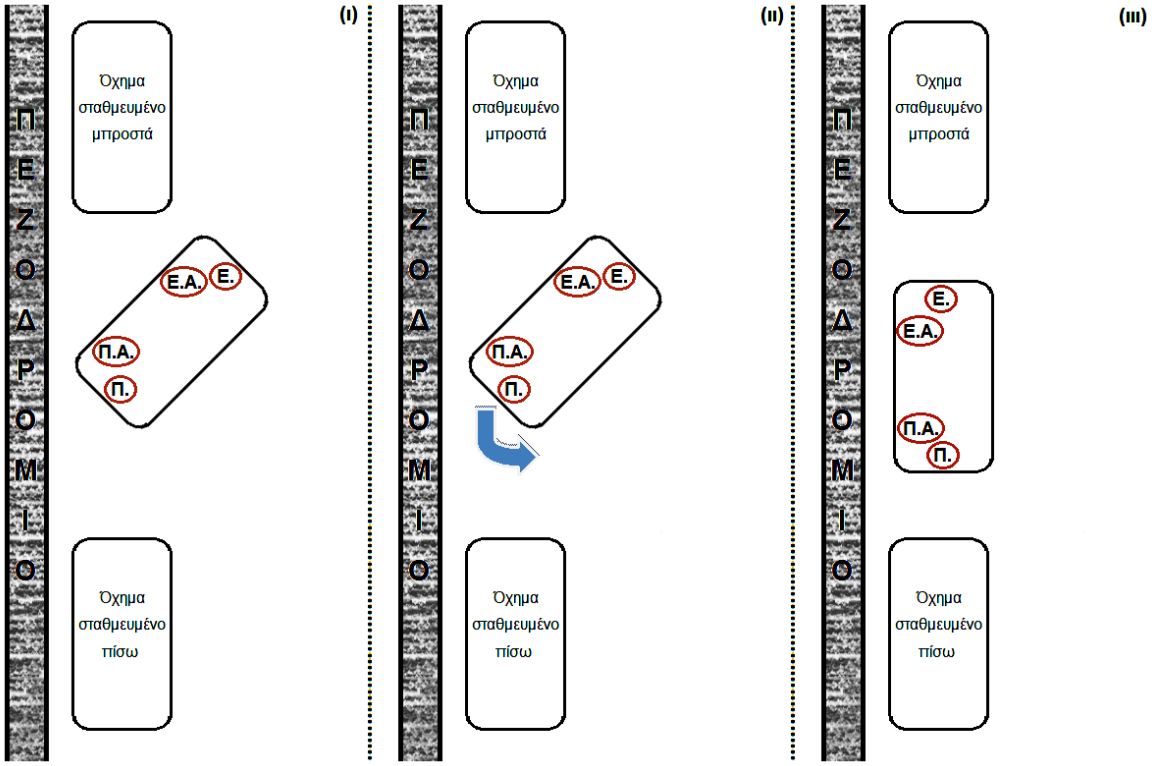
- Εκτέλεση κώδικα: Οι μπροστινοί τροχοί του αυτοκινήτου στρίβουν όλη τη διαδρομή του «τιμονιού» προς τα αριστερά. Το όχημα κινείται προς τα πίσω με αυτήν την γωνία μέχρι η πίσω αριστερά γωνία του να βρεθεί σε απόσταση 10cm από την άκρη του πεζοδρομίου ή του εμποδίου.
- Τελική θέση: Το αυτοκίνητο σταματά με την πίσω αριστερά γωνία κοντά στο πεζοδρόμιο. Ακολουθεί άλλη υπο-ρουτίνα με σκοπό την ολοκλήρωση του παρκαρίσματος.



Εικόνα 74 Σχηματική παράσταση της εκτέλεσης κίνησης οπισθογωνίας.

Κίνηση 3: Επαναφορά και τελική θέση παρκαρίσματος

- Εκτέλεση κώδικα: Οι μπροστινοί τροχοί του αυτοκινήτου στρίβουν όλη τη διαδρομή του «τιμονιού» προς τα δεξιά. Το όχημα αντιστρέφει την κίνησή του με κατεύθυνση μπροστά υπό αυτήν την γωνία, μέχρι ο εμπρός αισθητήρας να ανιχνεύσει απόσταση μικρότερη των 10 cm από το αυτοκίνητο, το οποίο είναι παρκαρισμένο εμπρός του. Παράλληλα, κατά τη κίνηση αυτή, οι τροχοί επανέρχονται σε ευθεία θέση. Με αυτή τη διαδικασία ολοκληρώνεται το παρκάρισμα, έτσι το αυτοκίνητο ακινητοποιείται.

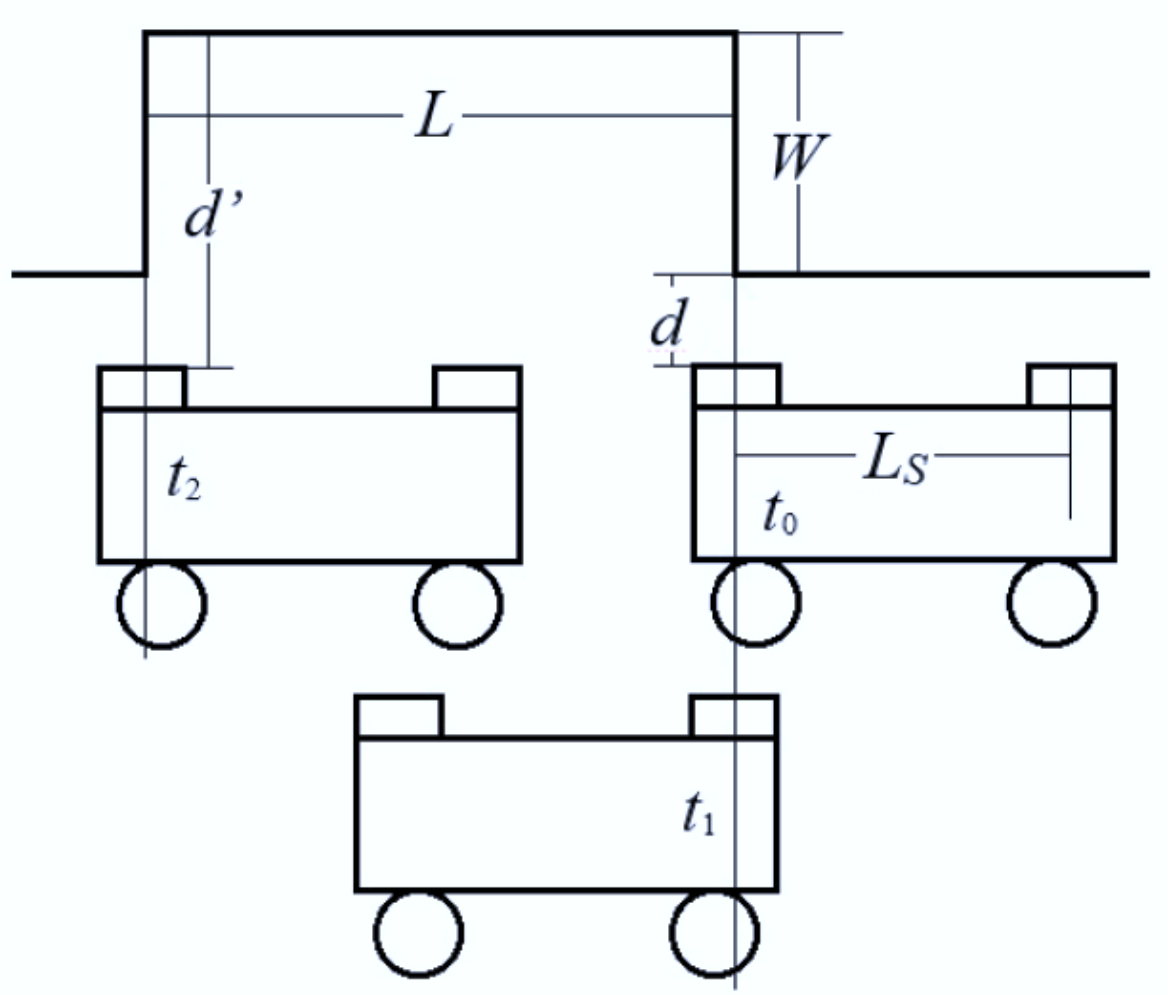


Εικόνα 75 Σχηματική παράσταση της επαναφοράς και τελικής θέσης παρκαρίσματος.

4.2 Ανάλυση της διαδικασίας παρκαρίσματος

Μόλις οι πλευρικοί αισθητήρες ανιχνεύσουν την κατάλληλη απόσταση ώστε να τηρείται η συνθήκη παρκαρίσματος, υπολογίζεται ο αντίστοιχος χρόνος καταγραφής και αποστολής δεδομένων προς το μικροελεγκτή Arduino ώστε να υπολογιστεί το μήκος της κενής θέση στάθμευσης L, που δίνεται από τον τύπο:

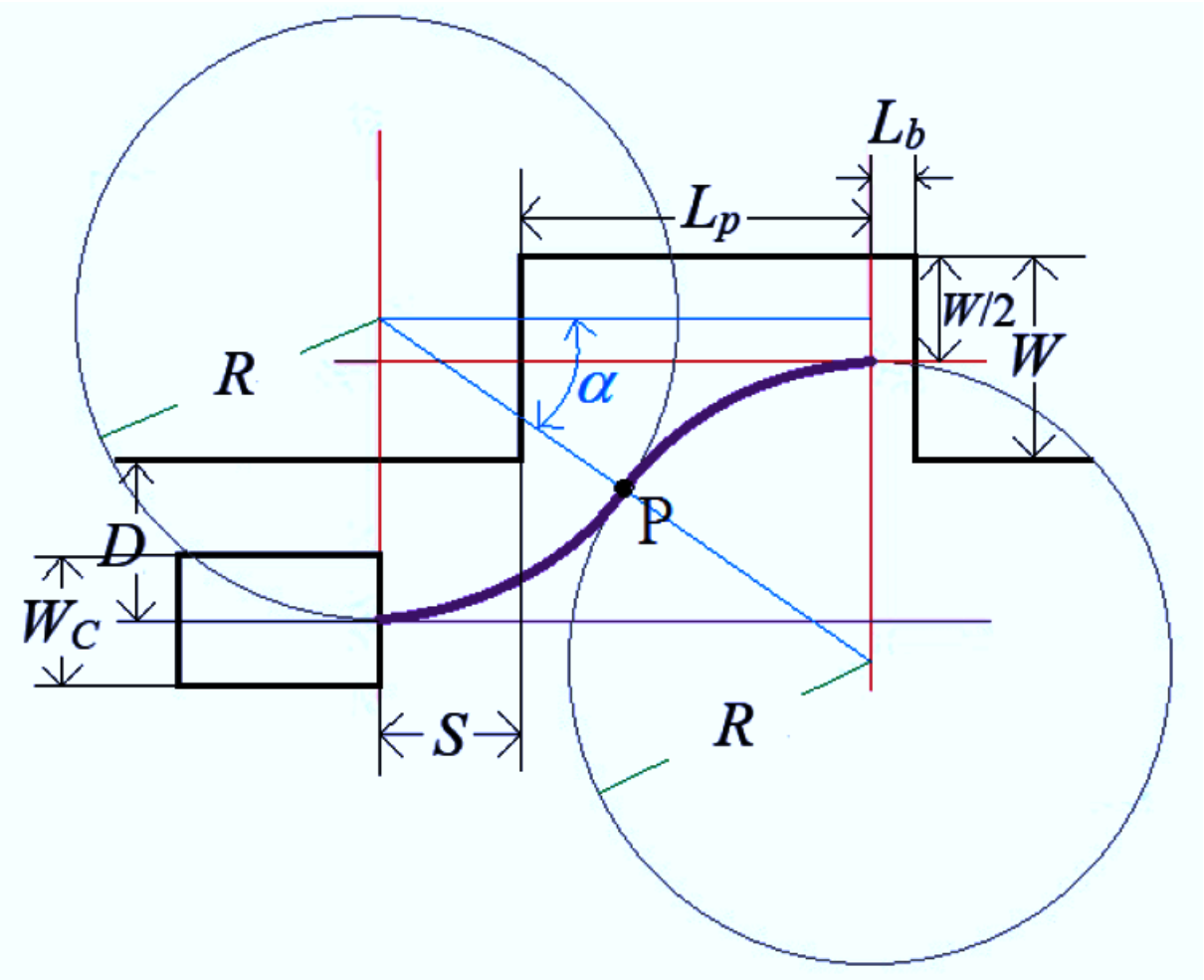
$$\boldsymbol{L = \frac{t_2 - t_0}{t_1 - t_0}\, Ls} \quad [22]$$



Εικόνα 76 Ανάλυση της διαδικασίας παρκαρίσματος. Μέρος 1^{ο}. [22]

όπου Ls είναι η απόσταση ανάμεσα στον μπροστινό πλευρικό αισθητήρα και τον πίσω πλευρικό, t_0 είναι ο χρόνος όπου ο μπροστινός πλευρικός αισθητήρας ανιχνεύει την πρώτη κατάλληλη (βάσει συνθήκης) θέση παρκαρίσματος, t_1 είναι ο χρόνος που ο πίσω πλευρικός αισθητήρας ανιχνεύει την πρώτη κατάλληλη θέση (απόσταση) παρκαρίσματος και t_2 είναι η στιγμή όπου ο μπροστινός πλευρικός αισθητήρας ανιχνεύει την δεύτερη αλλαγή στη θέση σε σχέση με την πρώτη κατάλληλη θέση παρκαρίσματος. Η μέση ταχύτητα του αυτοκινήτου μπορεί να υπολογιστεί με βάση τον τύπο:

$$\boldsymbol{u = \frac{L}{t_2 - t_0} = \frac{Ls}{t_1 - t_0}} \quad [22]$$



Εικόνα 77 Ανάλυση της διαδικασίας παρκαρίσματος. Μέρος 2^{ο}. [22]

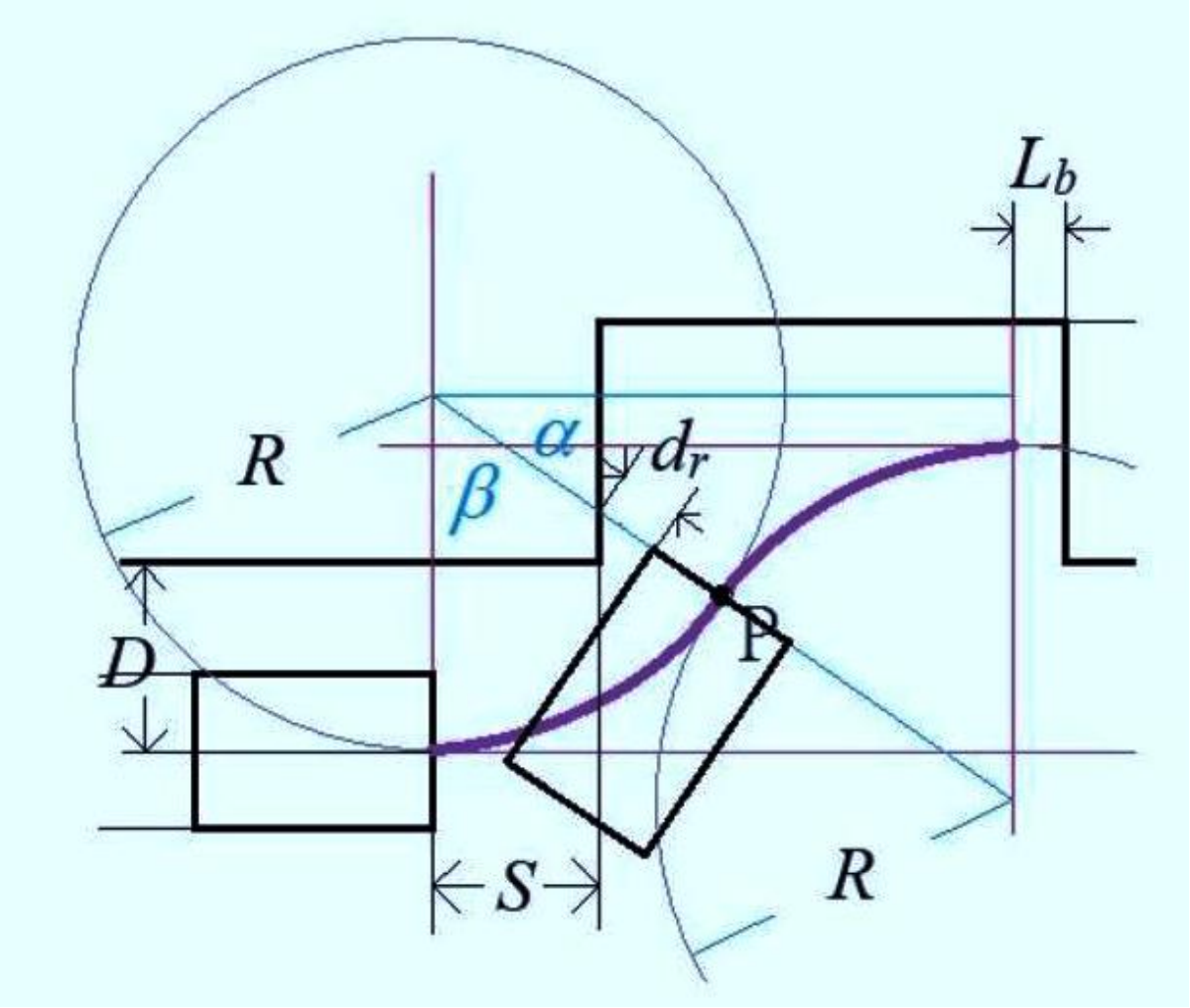
<u>Κίνηση του αυτοκινήτου στη διαδικασία παρκαρίσματος</u>

Στο Σχήμα 37 εμφανίζεται η καμπύλη διαδρομή της οπισθογωνίας την οποία πρέπει να ακολουθήσει το αυτοκίνητο για να πάρει την ορθή θέση παρκαρίσματος. Βάσει του σχήματος, η γωνία a που πρέπει να στρίψει αρχικά το αυτοκίνητο για να λάβει ορθή θέση κατά την οπισθογωνία υπολογίζεται από τον τύπο:

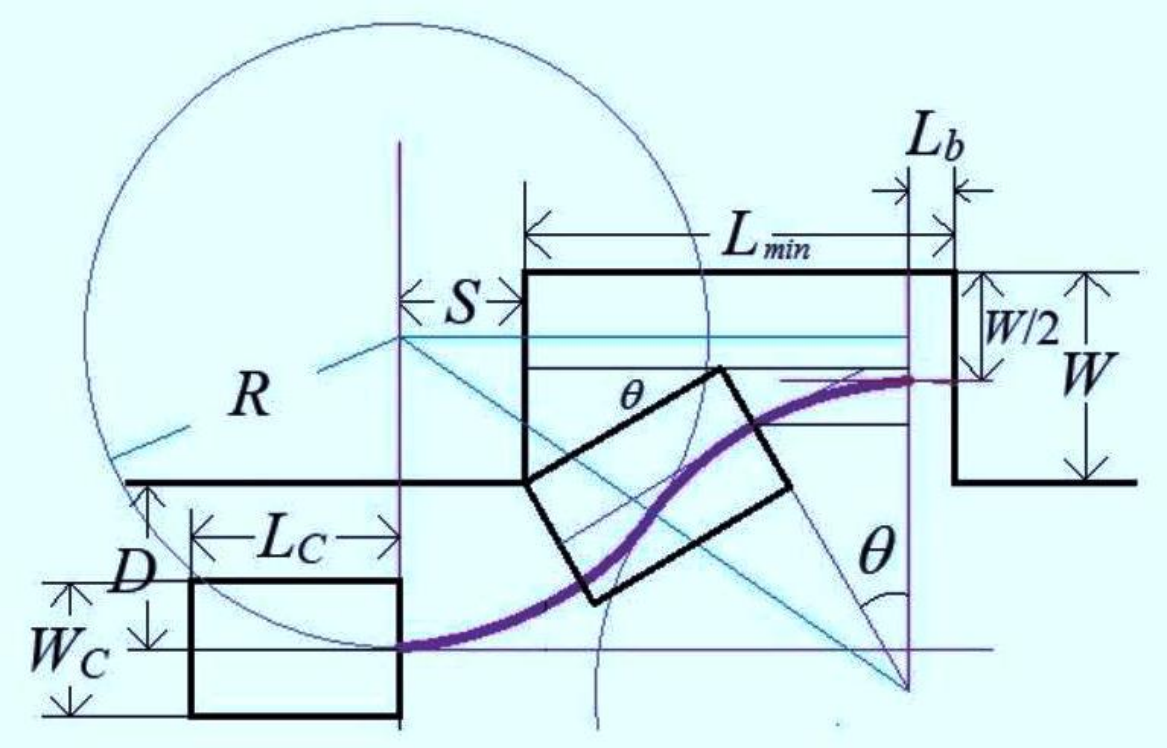
$$\boldsymbol{a} = \mathbf{sin^{-1}}\left(\frac{\boldsymbol{2R-D-\frac{w}{2}}}{\boldsymbol{2R}}\right),$$ [22]

όπου το D υπολογίζεται από τον τύπο: $\boldsymbol{D} = \boldsymbol{d} + \frac{\boldsymbol{W_c}}{\boldsymbol{2}}$ [22]

Στο σημείο P, οι μπροστινοί τροχοί πρέπει να στραφούν προς την άλλη κατεύθυνση έτσι ώστε το αυτοκίνητο να κινείται σε τροχιά του τόξου R β όπου $\beta = \frac{\pi}{2-\alpha}$, με την παραδοχή ότι εκτελεί μια σταθερή ταχύτητα. Ένας πιο ακριβής τρόπος για να διαπιστώσουμε αν το αυτοκίνητο έχει φτάσει στο σημείο P είναι να χρησιμοποιηθεί ο πίσω αισθητήρας, ο οποίος θα ελέγξει την απόσταση dr σύμφωνα με το πιο κάτω σχήμα. [22]



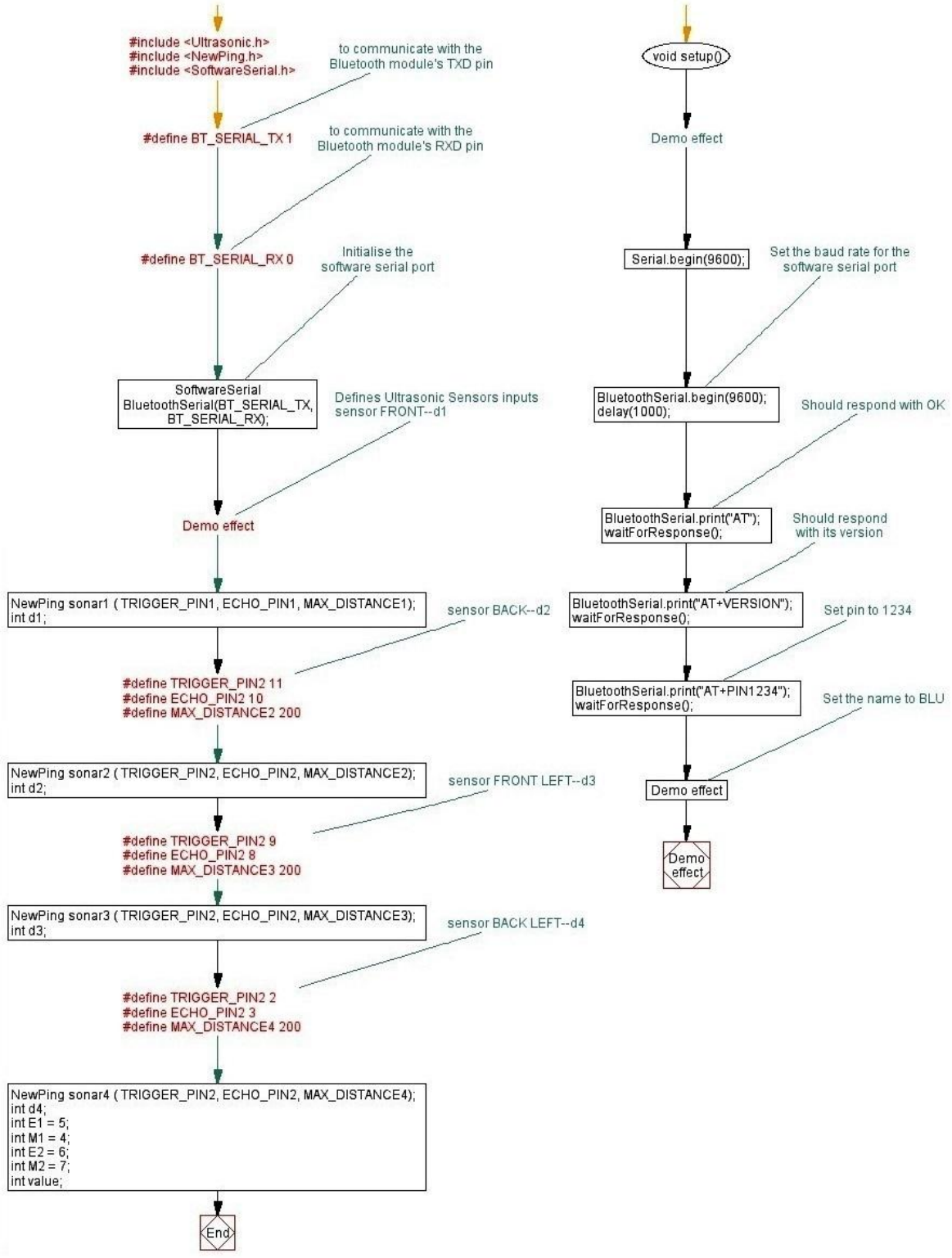
Εικόνα 78 Κίνηση του αυτοκινήτου στη διαδικασία παρκαρίσματος. Μέρος 1^{o}. [22]



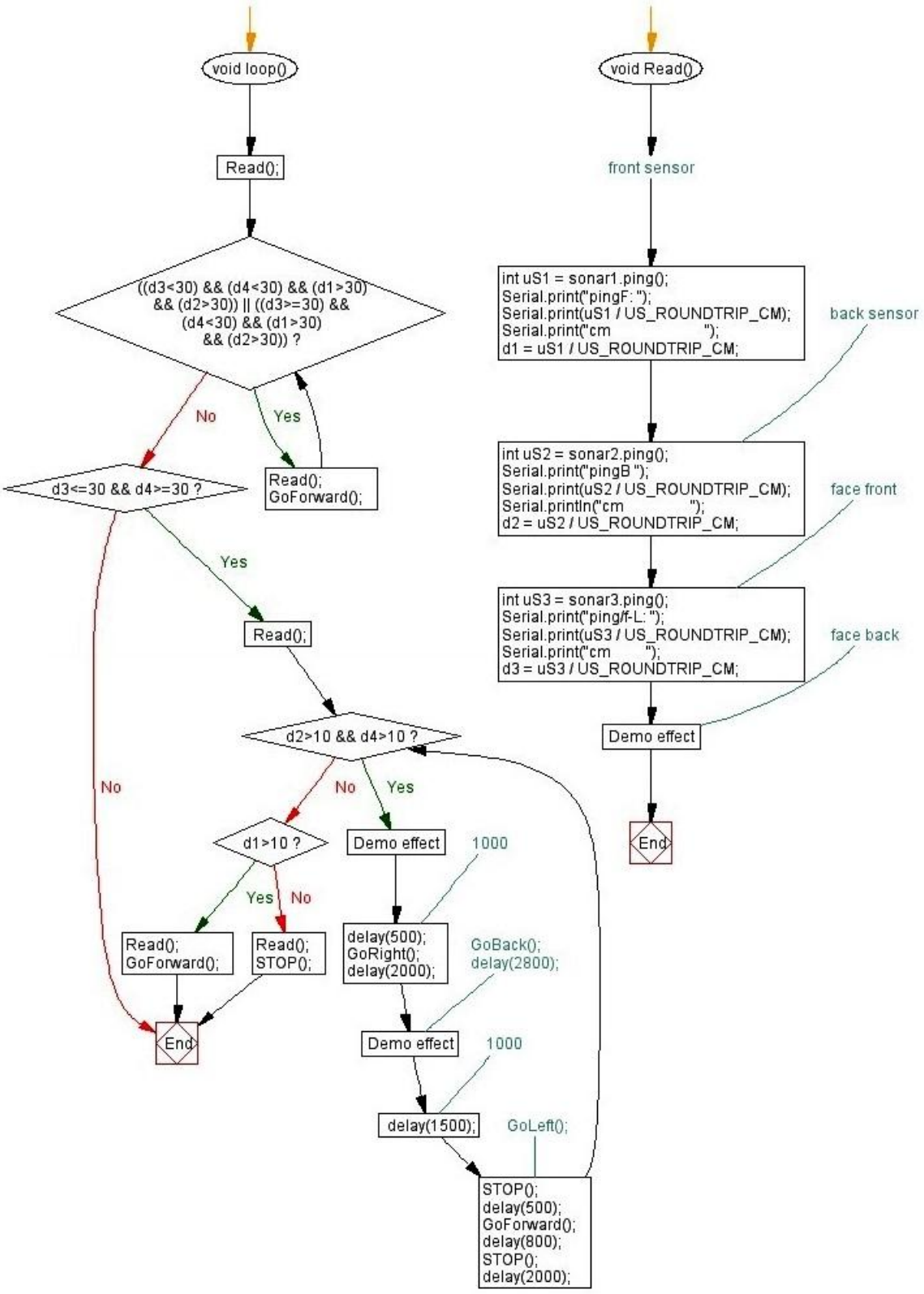
Εικόνα 79 Κίνηση του αυτοκινήτου στη διαδικασία παρκαρίσματος. Μέρος 2^o. [22]

Στα δοκιμαστικά που ακολούθησαν, υπολογίστηκαν όλες οι παραπάνω παράμετροι παρκαρίσματος και δομήθηκε αντίστοιχος κώδικας όπου σε τακτά delays δοκιμάστηκαν και επιλέχτηκαν οι - κατά προσέγγιση των υπαρκτών αντικειμενικών συνθηκών - χρονισμοί που πρέπει να τηρούνται κατά τη διάρκεια της διαδικασίας παρκαρίσματος.

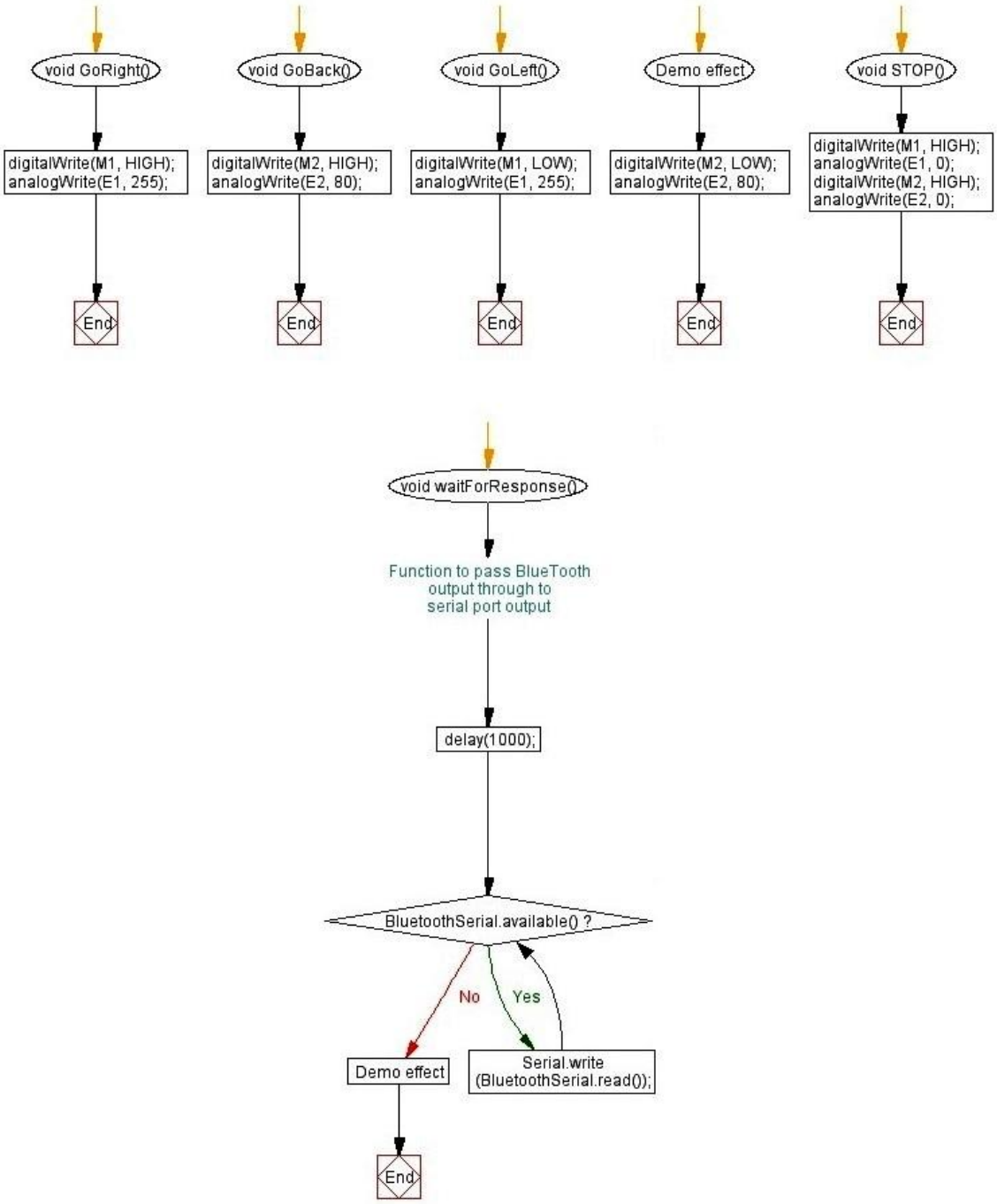
4.3 Διαγράμματα ροής



Εικόνα 80 Διάγραμμα ροής του κώδικα για το παρκάρισμα. Μέρος 1^{ο}.



Εικόνα 81 Διάγραμμα ροής του κώδικα για το παρκάρισμα. Μέρος 2^{ο}.



Εικόνα 82 Διάγραμμα ροής του κώδικα για το παρκάρισμα. Μέρος 3^{ο}.

5. ΣΧΟΛΙΑ, ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Μια μεγάλη σειρά δοκιμών έδειξε ότι ο σκοπός επιτεύχθηκε: το αυτοκινητάκι έβρισκε το χώρο και πάρκαρε κανονικά και απρόσκοπτα.

5.1 Αντικειμενικές δυσκολίες

Κατασκευαστικά το αυτοκίνητο του εμπορίου, που τελικά χρησιμοποιήθηκε στην παρούσα εργασία, ανταποκρινόταν καλύτερα στις στροφές απ' ότι άλλα προηγούμενων δοκιμών. Και αυτό όμως, παρουσίαζε κάποια ελαττώματα, που δυσχέραιναν την εκτέλεση παρκαρίσματος.

i. Το διαφορικό σύστημα περιστροφής των εμπρόσθιων τροχών παρουσιάζει διαφορετικές αποκρίσεις στις συγκεκριμένες εντολές αλλαγής κατεύθυνσης (στροφή δεξιά ή αριστερά ή επαναφορά τροχών). Η έλλειψη σωστής συμπεριφοράς οφείλεται στα κακής ποιότητας πλαστικά που διαθέτει και στην ελαττωματική συνάρθρωση όλου του διαφορικού συστήματος περιστροφής. Οι ελλείψεις αυτές δημιουργούν ενίοτε λανθασμένη γωνία στην στροφή των εμπρόσθιων τροχών.
ii. Η πρόσφυση των τροχών σε διαφορετικά εδάφη αλλάζει την ταχύτητα των τροχών (διαφορετική τριβή ανάλογα με το έδαφος κίνησης).
iii. Η έλλειψη συστήματος πέδησης εμποδίζει την έγκαιρη ακινητοποίηση του αυτοκινήτου σε προεπιλεγμένους χρόνους λόγω της αδράνειας (ορμή) κίνησης του αυτοκινήτου στις ανάλογες στιγμές.
iv. Οι DC κινητήρες διαθέτουν μια ελλειμματική σταθερή απόδοση υπό την ίδια τάση (V), αποδίδοντας κάθε φορά διαφορετικό αριθμό στροφών.
v. Η τροφοδοσία του κυκλώματος γίνεται με μπαταρία Lipo 14,8 V και 13000 mh, ωστόσο σε κάθε επαναφόρτισή της δεν μπορεί να αποδώσει το μέγιστο της ισχύος της.
vi. Η διασύνδεση του Arduino Uno με τον υπολογιστή μέσω Bluetooth είναι μια διαδικασία που απαιτεί πολλά βήματα, τα οποία είναι πιθανόν να χρειαστεί να επαναληφθούν για την εδραίωση της επικοινωνίας του Arduino Uno με τον υπολογιστή.

5.2 Προτάσεις βελτιστοποίησης

Στη συγκεκριμένη κατασκευή, με βάση τις υπάρχουσες συνθήκες και προκλήσεις που παρουσιάστηκαν μέχρι να επιτευχθεί ικανοποιητικό παρκάρισμα του αυτοκινήτου, μπορούν να πραγματοποιηθούν μελλοντικές αλλαγές, οι οποίες θα επιτύχουν τη βελτιστοποίηση παρκαρίσματος.

Για τον έλεγχο στροφών που θα αποδίδουν οι DC κινητήρες μπορεί να χρησιμοποιηθούν DC Speed controllers. Ωστόσο, θα μπορούσε να αντικατασταθεί ο εμπρός DC κινητήρας με Servo Motor. Έτσι, επιτυγχάνεται η ευκολότερη και ακριβέστερη στροφή των εμπρόσθιων τροχών. Επίσης, ο οπίσθιος κινητήρας θα μπορούσε να αντικατασταθεί με έναν Stepper Motor, που μέσω ελεγκτή Stepper Motor Driver Module, να αποδώσει τη βέλτιστη περιστροφική απόδοση του συστήματος κίνησης.

Επίσης, με τη χρήση ενός Altitude Gyro Accelerator Magnetometer Module, θα ήταν εφικτό να γίνουν διορθώσεις πορείας του αυτοκινήτου και να αμβλυνθούν οι εξωγενείς παράγοντες, οι οποίοι δυσχεραίνουν το αποτελεσματικό παρκάρισμα.

ΠΑΡΑΡΤΗΜΑ

Παρατίθεται εδώ ο κώδικας για τον που «φορτώσαμε» στο Arduino Uno.

```
#include <Ultrasonic.h>
#include <NewPing.h>
#include <SoftwareSerial.h>

/* to communicate with the Bluetooth module's TXD pin */
#define BT_SERIAL_TX 1
/* to communicate with the Bluetooth module's RXD pin */
#define BT_SERIAL_RX 0
/* Initialise the software serial port */
SoftwareSerial BluetoothSerial(BT_SERIAL_TX, BT_SERIAL_RX);

// Defines Ultrasonic Sensors inputs
//sensor FRONT--d1
#define TRIGGER_PIN1 13
#define ECHO_PIN1 12
#define MAX_DISTANCE1 200
NewPing sonar1 ( TRIGGER_PIN1, ECHO_PIN1, MAX_DISTANCE1);
int d1;

//sensor BACK--d2
#define TRIGGER_PIN2 11
#define ECHO_PIN2 10
#define MAX_DISTANCE2 200
NewPing sonar2 ( TRIGGER_PIN2, ECHO_PIN2, MAX_DISTANCE2);
int d2;

//sensor FRONT LEFT--d3
#define TRIGGER_PIN2 9
#define ECHO_PIN2 8
#define MAX_DISTANCE3 200
NewPing sonar3 ( TRIGGER_PIN2, ECHO_PIN2, MAX_DISTANCE3);
int d3;

//sensor BACK LEFT--d4
#define TRIGGER_PIN2 2
#define ECHO_PIN2 3
#define MAX_DISTANCE4 200
NewPing sonar4 ( TRIGGER_PIN2, ECHO_PIN2, MAX_DISTANCE4);
int d4;

int E1 = 5;
int M1 = 4;
int E2 = 6;
int M2 = 7;
int value;
void setup()
```

```
{
  /* Set the baud rate for the hardware serial port */
  Serial.begin(9600);
  /* Set the baud rate for the software serial port */
    BluetoothSerial.begin(9600);
    delay(1000);
    // Should respond with OK
    BluetoothSerial.print("AT");
    waitForResponse();

    // Should respond with its version
    BluetoothSerial.print("AT+VERSION");
    waitForResponse();

    // Set pin to 1234
    BluetoothSerial.print("AT+PIN1234");
    waitForResponse();

    // Set the name to BLU
     BluetoothSerial.print("AT+NAMEBLU");
    waitForResponse();
    Serial.println("Finished!");

    pinMode(M1, OUTPUT);
    pinMode(M2, OUTPUT);
    STOP();
    GoForward();
    Read();
}

void loop()
{
  Read();
  while (((d3<30) && (d4<30) && (d1>30) && (d2>30)) || ((d3>=30) && (d4<30) && (d1>30) &&
(d2>30)))
  {
    Read();
    GoForward();
  }
  if (d3<=30 && d4>=30)
  {
    Read();
    while (d2>10 && d4>10)
    {
      Read();
 GoBack();
 delay(500);//1000
  GoRight();
  delay(2000);
//  GoBack();
//  delay(2800);
```

```
  GoLeft();
  delay(100);
             GoBack();
            delay(1500); //1000
             //GoLeft();
             STOP();
             delay(500);
  GoForward();
  delay(800);
            STOP();
             delay(2000);
    }
    if (d1>10)
    {
      Read();
    GoForward();
    }
    else
    {
      Read();
     STOP();
    }
  }
}

void Read()
{
  //front sensor
  int uS1 = sonar1.ping();
  Serial.print("pingF: ");
  Serial.print(uS1 / US_ROUNDTRIP_CM);
  Serial.print("cm                    ");
  d1 = uS1 / US_ROUNDTRIP_CM;

   //back sensor
  int uS2 = sonar2.ping();
  Serial.print("pingB ");
  Serial.print(uS2 / US_ROUNDTRIP_CM);
  Serial.println("cm               ");
  d2 = uS2 / US_ROUNDTRIP_CM;
//
  //face front
  int uS3 = sonar3.ping();
  Serial.print("ping/f-L: ");
  Serial.print(uS3 / US_ROUNDTRIP_CM);
  Serial.print("cm        ");
  d3 = uS3 / US_ROUNDTRIP_CM;

//face back
  int uS4 = sonar4.ping();
  Serial.print("ping/f-f: ");
```

```
  Serial.print(uS4 / US_ROUNDTRIP_CM);
  Serial.print("cm        ");
  d4 = uS4 / US_ROUNDTRIP_CM;
  }

void GoRight()
{
  digitalWrite(M1, HIGH);
  analogWrite(E1, 255);
  }
void GoBack()
{
  digitalWrite(M2, HIGH);
  analogWrite(E2, 80);
  }
void GoLeft()
{
  digitalWrite(M1, LOW);
  analogWrite(E1, 255);
 }
void GoForward()
{
  digitalWrite(M2, LOW);
  analogWrite(E2, 80);
 }

void STOP()
{
  digitalWrite(M1, HIGH);
  analogWrite(E1, 0);
  digitalWrite(M2, HIGH);
  analogWrite(E2, 0);
 }

// Function to pass BlueTooth output through to serial port output
void waitForResponse() {
delay(1000);
while (BluetoothSerial.available()) {
Serial.write(BluetoothSerial.read());
}
Serial.write("\n");
}
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. http://circontrol.com/blog-posts/the-parking-of-the-future-problems-challenges-and-solutions/
2. https://en.wikipedia.org/wiki/Intelligent_Parking_Assist_System
3. https://www.thecarconnection.com/news/1039030_does-the-2010-lexus-ls-600h-park-itself-as-well-as-a-ford
4. https://www.researchgate.net/publication/281812528_Development_of_automated_parallel_parking_system_in_small_mobile_vehicle
5. http://web.mit.edu/6.111/www/f2012/projects/fni_Project_Final_Report.pdf
6. http://roundtable.menloschool.org/issue20/6_Chen_MS_Roundtable20_Winter_2015.pdf
7. http://eecs.richardnelson.com/2006-07/CpE_Projects/RC%20Car%20Parallel%20Parking/Parallel_Parking_RC_Car.pdf
8. http://ishitvtech.in/pdf/sajet-vol-3-no2-16.pdf
9. https://www.researchgate.net/publication/238522350_A_Study_of_Automatic_Parallel_Parking_System_-_from_the_Viewpoints_of_User_and_Manufacturer
10. http://www.tsi.lv/sites/default/files/editor/science/Research_journals/Tr_Tel/2012/V4/zadachyn_dorokhov.pdf
11. https://www.researchgate.net/publication/274478739_Simulate_Study_of_Automatic_Parking_System
12. https://store.arduino.cc/arduino-uno-rev3
13. https://www.electroschematics.com/10955/build-arduino-bootload-atmega-microcontroller-part-1/
14. http://www.wikiwand.com/en/Sound#/Perception_of_sound
15. https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/
16. http://www.ardumotive.com/greco-robotgr.html
17. https://www.maxphi.com/ultrasonic-sensor-interfacing-arduino-tutorial
18. https://www.dfrobot.com/wiki/index.php/Arduino_Motor_Shield_(L298N)_(SKU:DRI0009)
19. https://www.dfrobot.com/wiki/index.php/MD1.3_2A_Dual_Motor_Controller_SKU_DRI0002
20. http://www.instructables.com/id/Arduino-Motor-Shield-Tutorial/
21. https://arduinobots.wordpress.com/%CE%BA%CE%B1%CF%84%CE%B1%CF%83%CE%BA%CE%B5%CF%85%CE%AE-%CF%81%CE%BF%CE%BC%CF%80%CF%8C%CF%84/%CF%85%CE%BB%CE%B9%CE%BA%CE%AC/
22. https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0ahUKEwjekN6D7JHbAhXFbVAKHfONBCcQFghYMAY&url=https%3A%2F%2Fpeer.asee.org%2Fautomatic-parking-vehicle-system.pdf&usg=AOvVaw2zI0Go-N3FU1gdP0TJAinP
23. https://grobotronics.com/bluetooth-module-for-arduino-hc06.html?sl=en
24. https://www.electroschematics.com/8902/hc-sr04-datasheet/