ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Πρόγραμμα Σπουδών: Μηχανικών Πληροφορικής Τ.Ε.

Πτυχιακή Εργασία

ΑΝΑΠΤΥΞΗ ΠΡΩΤΟΚΟΛΛΟΥ M2M ΓΙΑ AD HOC ΔΙΚΤΥΑ ΣΕ ΛΕΙΤΟΥΡΓΙΑ ΣΜΗΝΟΥΣ

Τσιχλάκης Βασίλειος (Αμ: ΤΠ4212)

Επιβλέπων καθηγητής : Παναγιωτάκης Σπυρίδων

Επιτροπή Αξιολόγησης :

ΠΑΝΑΓΙΩΤΑΚΗΣ ΣΠΥΡΙΔΩΝ

ΒΑΣΙΛΑΚΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΜΑΡΚΑΚΗΣ ΕΥΑΓΓΕΛΟΣ

ΗΡΑΚΛΕΙΟ, 9/4/2021

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Σύνοψη **4**

1. Ανάλυση τεχνολογιών **5**

1.1 Multi-Robot Systems 5

1.2 Swarm Robotics 5

1.3 M2M Communications 6

1.3.1 MQTT Message 8

1.3.2 Χαρακτηριστικά του MQTT 8

1.3.2.1 Δημοσίευση – Εγγραφή 8

1.3.2.2 Διάγραμμα αλληλουχίας επικοινωνίας 9

1.3.2.3 Ποιότητα υπηρεσιών (QoS) 9

1.3.2.4 Διατήρηση μηνυμάτων 10

1.3.2.5 Τελευταία Επιθυμία (Last Will and Testament) 10

1.4 AD HOC Δίκτυα 10

2. Σχεδιασμός Υλικού και Ανάλυση **12**

2.1 Σχεδιασμός Υλικού 12

2.2 Λίστα Εξαρτημάτων Και Σχεδιάγραμμα Αρχιτεκτονικής 13

2.3 Ανάλυση Και Δυνατότητες Εξαρτημάτων 15

2.3.1 Arduino Uno 15

2.3.2 DC-Motors 17

2.3.3 NRF tranceiver modules 17

2.3.4 H-bridge βασισμένα στο L298N 19

2.3.5 Robot Car 2WD 20

2.3.6 HC-020K Speed Measuring Module with Photoelectric Encoders 21

2.3.7 HC-05 Bluetooth Module 21

2.3.8 HC-SR05 22

3. Δίκτυα και Πρωτόκολλα **24**

3.1 Τι Είναι Δίκτυο Υπολογιστών 24

3.2 Τεχνολογίες και Τοπολογίες 24

3.2.1 Master/Slave Communications 24

3.2.2 Τοπολογίες 24

3.2.2.1 Τι είναι τοπολογίες 24

3.2.2.2 Πλέγμα (Mesh) 25

3.2.2.2.1 Γενικές πληροφορίες 25

3.2.2.2.2 RF24 26

3.2 Τι Είναι Το Πρωτόκολλο 27

3.3 Το ιδιωτικό Πρωτόκολλό Μας 28

4. Υλοποίηση Συστήματος 31

4.1 Arduino IDE 31

4.2 Βιβλιοθήκες 31
4.2.1 Serial Peripheral Interface 31
4.2.2 NRF24L01 31

4.3 Αρχικοποιήσεις 31
4.3.1 #define 32
4.3.2 Void Setup 32

4.4 Συναρτήσεις 33
4.4.1 startMove() 33
4.4.2 moveForward() 33
4.4.3 stopMoving() 34
4.4.4 turnLeft() turnRight() turnLeftBack() turnRightBack() moveBackwards() 34
4.4.5 calcDist() 35
4.4.6 countLeft() countRight() 35
4.4.7 changeD() 36
4.4.8 messageCreation() 36
4.4.9 moveDataFromMessage() msgTypeFromMessege() messengeIdFromMessage() 37
4.4.10 autoPilot() 37
4.4.11 Loop() 39
4.4.12 Loop() reciver 40
4.4.13 autoPilot() reciver 40

5. Λειτουργία και Εφαρμογή Android 41

5.1 Εφαρμογή Android 41

5.2 Λειτουργία 42

6 Συμπεράσματα και Μελλοντικές Επεκτάσεις 47

6.1 Συμπεράσματα 47

6.2 Μελλοντικές Επεκτάσεις 48

ΠΗΓΕΣ 49

Σύνοψη

Στην παρούσα διπλωματική εργασία επιχειρούμε να μελετήσουμε και να υλοποιήσουμε ένα σύστημα δύο ή περισσοτέρων αυτοκινούμενων οχημάτων, τα οποία είναι συνδεδεμένα μεταξύ τους μέσω ενός δικτύου WSN σε τοπολογία πλέγματος και επεξεργάζονται δεδομένα. Αυτά παράγουν χρήσιμη πληροφορία χρησιμοποιώντας μικροελεγκτές τύπου "Arduino" και διάφορους αισθητήρες . Κατ' αυτόν τον τρόπο επιτρέπεται η ανταλλαγή δεδομένων ανάμεσα τους και κατ΄ επέκταση η δυνατότητα συγχρονισμού τους. Προκειμένου να επιτευχθεί η παραγωγή δεδομένων, είναι αναγκαία η χρήση καταλλήλων αισθητήρων, οι οποίοι θα τροφοδοτήσουν με την απαραίτητη πληροφορία τα συστήματα που τα έχουν ανάγκη. Αξιοποιώντας τα κοινοποιημένα δεδομένα, μας δίνεται η δυνατότητα της δημιουργίας αλγορίθμων αποφυγής εμποδίων. Το εν λόγω σύστημα κάνει χρήση και των δύο προαναφερθεισών δυνατοτήτων, αξιοποιώντας έναν αλγόριθμο ο οποίος συνδυάζει συγχρονισμό κινήσεων με αποφυγή εμποδίων. Προκειμένου να δουλέψουν όλα αυτά θα χρειαστεί να δημιουργηθεί ένα πρωτόκολλο επικοινωνίας τύπου M2M (Μηχανής-σε-μηχανή). Αυτό το σύστημα είναι κατά βάση ένα multi robot system με στοιχεία από Swarm Robotics, τεχνολογίες που θα αναλυθούν στο παρόν έγγραφο. Μερικά από τα πλεονεκτήματα αυτών των εφαρμογών είναι το χαμηλό κόστος παραγωγής, το μικρό μέγεθος και η αντικαταστασιμότητα τους. Γεγονός που κάνει τους ειδικούς, σε συνδυασμό με το Internet of Things, να πιστεύουν ότι η χρήση αυτών των τεχνολογιών θα γίνεται όλο και πιο συχνή.

Στα παρακάτω κεφάλαια θα αναλύσουμε περαιτέρω τις τεχνολογίες που χρησιμοποιούνται, τον σχεδιασμό και την ανάλυση του υλικού που χρησιμοποιείται, τον σχεδιασμό και την ανάλυση του πρωτοκόλλου, την ανάλυση του κώδικα , τα συμπεράσματα και τις μελλοντικές επεκτάσεις.

1. Ανάλυση τεχνολογιών

1.1 Multi-Robot Systems

Ο όρος “Multi robot systems” (MRS) στην ουσία αναφέρεται σε συστήματα που αποτελούνται από πολλαπλά ρομπότ, τα οποία μπορούν να συνεργάζονται και να επικοινωνούν μεταξύ τους ώστε να επιτελούν συγκεκριμένα καθήκοντα [1]. Ένα σύνολο από δύο ή περισσότερα αυτόνομα κινητά ρομπότ που αλληλεπιδρούν, ονομάζεται «Κοινωνία» ή «Ομάδα» των κινητών ρομπότ. Στο συγκεκριμένο τύπο συστήματος, τα ρομπότ διαθέτουν πολύ λιγότερες δυνατότητες ως αυτόνομες οντότητες, σε αντίθεση με τις δυνατότητες τις οποίες αποκτούν όταν συνεργάζονται μεταξύ τους. Παραδείγματα εφαρμογής των εν λόγω συστημάτων σε πραγματικές συνθήκες εντοπίζουμε στις στρατιωτικές αποστολές (επιτήρηση του πεδίου μάχης), σε περιπτώσεις καταστροφών κατά την αναζήτηση επιζώντων, στις παράλληλες και ταυτόχρονες μεταφορές οχημάτων, καθώς και στην παράδοση ωφέλιμου φορτίου.

Από την προηγούμενη δεκαετία παρατηρείται ότι τα “MRS” έχουν προσελκύσει ιδιαίτερο ερευνητικό ενδιαφέρον διεθνώς, γεγονός το οποίο δικαιολογείται από τα ουσιαστικά πλεονεκτήματα που φαίνεται να παρέχουν. Καταρχάς, έχει διαπιστωθεί ότι τα MRS είναι πολύ πιο οικονομικά σε αναλογία με την απόδοσή τους, συγκρίνοντάς τα με την κατασκευή ενός ενιαίου δαπανηρού ρομπότ αντίστοιχων δυνατοτήτων. Επιπλέον, η δυναμική (αποκεντρωμένη, διαμελισμένη) δομή των συγκεκριμένων ρομποτικών συστημάτων, τους προσδίδει σημαντική ανεκτικότητα σε μεμονωμένα σφάλματα, με αποτέλεσμα τη βελτίωση της αξιοπιστίας και ανθεκτικότητάς τους. Αντιθέτως, σε περίπτωση σφάλματος σε κάποιο λειτουργικό μέρος ενός ενιαίου συμβατικού ρομπότ, το ενδεχόμενο να τεθεί εκτός λειτουργίας είναι πολύ πιθανότερο. Ωστόσο, η δυναμική δομή και το μικρό μέγεθος των ρομπότ ενός MRS εφιστά την προσοχή στην κατασκευή ανθεκτικότερων μονάδων, ώστε να ελαττώνεται στο μεγαλύτερο δυνατό ποσοστό ο κίνδυνος συντριβής τους, κατά την λειτουργία τους σε αντίξοες συνθήκες. [2]

1.2 Swarm Robotics

Όταν η ομάδα ρομπότ συνεργάζεται για να ολοκληρώσει μια εργασία τότε αυτή η ομάδα ονομάζεται σμήνος ρομπότ (Swarm Robotics). Η ρομποτική σμήνους αντλεί έμπνευση από τους τρόπους με τους οποίους βιολογικοί κοινωνικοί οργανισμοί αλληλεπιδρούν, όπως τα μυρμήγκια, και χρησιμοποιούν συνεργατικές συμπεριφορές για να επιτύχουν πολύπλοκες εργασίες πέρα από την ικανότητα οποιουδήποτε ατόμου. Τα παρακάτω χαρακτηριστικά προέρχονται από τους βασικούς στόχους της ρομποτικής σμήνους [3]:

1. Απλό και κομψό - Ο ελεγκτής της συμπεριφοράς των ρομπότ είναι πολύ απλός. Οι συμπεριφορές των μεμονωμένων ρομπότ μπορούν να αναπαρασταθούν σαν ένα απλό state machine με διαφορετικά states.
2. Κλιμακωτό - Οι αλγόριθμοι των ρομπότ σμήνους είναι σχεδιασμένοι ώστε να λειτουργούν για οποιοδήποτε αριθμό ρομπότ. Επίσης είναι εφικτό να μεγαλώσουν σε αριθμό όσο προστίθενται νέα ρομποτ.
3. Αποκεντρωμένο - Τα ρομπότ σε ένα σμήνος είναι αυτόνομα και δεν ακολουθούν καμία εξωτερική εντολή. Αν και μέλος του σμήνος ρομπότ μπορεί να επηρεαστεί άμεσα και προβλέψιμα από τις συμπεριφορές των άλλων, η επιλογή πάντα θα γίνεται από το ίδιο. Η αποκέντρωση συνηθώς πάει μαζί με την κλιμάκωση.
4. Χρήση τοπικών αλληλεπιδράσεων - Οι τοπικές αλληλεπιδράσεις χρησιμοποιούνται κατά τη μετάδοση μηνυμάτων στην πλειονότητα αυτών των αλγορίθμων. Οι εκπομπές εφαρμόζονται ως πρωτόκολλα μετάδοσης μηνυμάτων. Αυτό το ιδανικό αποτελεί σημαντικό παράγοντα για την επεκτασιμότητα του συστήματος.

Μερικοί αλγόριθμοι που εφαρμόζονται στην ρομποτική σμήνους είναι:

1. Διασπορά σε εσωτερικά περιβάλλοντα.

2. Κατανεμημένη τοπική προσαρμογή και αντιστοιχίσεις.
3. Κινητοί σχηματισμοί.
4. Συνεργατική αποφυγή τρυπών.
5. Διαμόρφωση διαδρομής με βάση την αλυσίδα.
6. Ομαδική μεταφορά

Εφαρμογές

Η ρομποτική σμήνους βρίσκει αρκετές πρακτικές εφαρμογές, όπως για παράδειγμα σε εργασίες όπου απαιτείται μικροσκοπικότητα (νανορομποτική, μικρορομποτική), σε κατανεμημένες εργασίες ανίχνευσης, σε micromachines και στο ανθρώπινο σώμα. Επιπλέον, μία από τις πιο υποσχόμενες χρήσεις της, είναι σε αποστολές διάσωσης καταστροφών. Σμήνη ρομπότ διαφορετικών μεγεθών θα μπορούσαν να σταλούν σε μέρη που οι εργαζόμενοι διάσωσης δεν μπορούν να φτάσουν με ασφάλεια, για να ανιχνεύσουν την παρουσία ζωής μέσω υπέρυθρων αισθητήρων. Από την άλλη πλευρά, η ρομποτική σμήνους ενδέχεται να μπορεί να εφαρμοστεί σε εργασίες που απαιτούν φθηνά χέρια, για παράδειγμα σε εργασίες εξόρυξης ή γεωργικής τροφής.

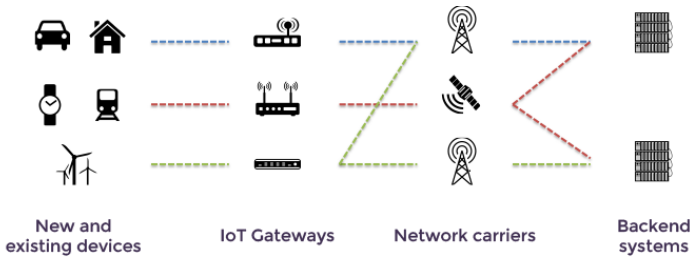
1.3 M2M Communications

Στην εποχή της τεχνολογίας, οι άνθρωποι τείνουν σε τρόπους που διευκολύνουν την εκτέλεση κάποιων εργασιών, με καλύτερο παράδειγμα την αυτοματοποίηση σε κάθε τομέα. Αυτό δεν είναι τίποτα άλλο παρά ένα ρομπότ / μηχανή που συν τοις άλλοις επικοινωνεί με άλλα παρόμοια μηχανήματα. Προκειμένου να επιτευχθεί ένας τέτοιος σκοπός, θα πρέπει να υπάρχει κάποια διεπαφή μεταξύ των μηχανών μέσω της οποίας να μπορούν να στέλνουν και να λαμβάνουν μηνύματα. Σε αυτό ακριβώς συντελούν τα πρωτόκολλα επικοινωνίας μηχανής-μηχανής (M2M) [4]. Η επικοινωνία μηχανής με μηχανή ή αλλιώς M2M είναι μια τεχνολογία ανταλλαγής πληροφοριών και εκτέλεσης ενεργειών, η οποία υλοποιείται μέσω ενσύρματων ή ασύρματων καναλιών για επικοινωνία μεταξύ μηχανών χωρίς την παρέμβαση του ανθρώπινου παράγοντα. Συχνά χρησιμοποιείται για απομακρυσμένη παρακολούθηση αισθητήρων και μετρητών βιομηχανικής χρήσης για να στέλνουν τα δεδομένα που συλλέγουν σε μονάδες επεξεργασίας και ανάλυσης. Για να επιτευχθεί αυτή η επικοινωνία συχνά χρησιμοποιείται ένα ασύρματο δίκτυο από μηχανές που μεταφέρουν την πληροφορία σε ένα κεντρικό κόμβο, ο οποίος με τη σειρά του, ανατροφοδοτεί με την πληροφορία που συλλέγει ένα κεντρικό σύστημα, για παράδειγμα έναν υπολογιστή σε ρόλο server. Οι M2M επικοινωνίες είναι η τελευταία τάση στην βιομηχανική εξέλιξη, η οποία συνδυάζει τις τεχνολογίες πληροφορικής με τις επικοινωνίες από δεδομένα μηχανών μεταξύ συσκευών ή μηχανών.

Εφαρμόζοντας τον ιδεολογικό πυρήνα του Internet of Things (IoT), η M2M επικοινωνία έχει μετατραπεί σε ένα σύστημα από δίκτυα, τα οποία μεταφέρουν πληροφορίες σε προσωπικές εφαρμογές. Σημαντικές εφαρμογές της M2M επικοινωνίας εντοπίζονται στους εξ' αποστάσεως χειρισμούς, στη ρομποτική, στη διαχείριση αποθηκών, στις λογιστικές υπηρεσίες, στον έλεγχο της κυκλοφορίας, στη διαχείριση στόλων και αλυσίδων εφοδιασμού, καθώς και στον τομέα της τηλεϊατρικής. Παράλληλα, η επέκταση των δικτύων IP σε παγκόσμιο επίπεδο, καθιστά εφικτή την καλύτερη και ταχύτερη επικοινωνία M2M με τη χρήση λιγότερης ενέργειας, ανοίγοντας έτσι νέες επιχειρηματικές προοπτικές για τους καταναλωτές και τους προμηθευτές.

Τα πρωτόκολλα M2M αποτελούν το επόμενο σημαντικό σημείο καμπής της χρήσης του διαδικτύου, επιτρέποντας τη διαλειτουργικότητα ανάμεσα σε βασικά στοιχεία όπως RFID, αισθητήρες, Wi-Fi, δίκτυα κινητής τηλεφωνίας και λογισμικό αυτόνομων υπολογιστών, και έχουν σχεδιαστεί για να βοηθούν τις δικτυωμένες συσκευές να λαμβάνουν αποφάσεις και να ερμηνεύουν δεδομένα. Η πιο αναγνωρισμένη μορφή επικοινωνίας M2M είναι η Τηλεμετρία που έχει τεθεί σε ισχύ για τη μετάδοση επιχειρησιακών δεδομένων από τις αρχές του περασμένου αιώνα. Το Διαδίκτυο και οι σύγχρονες ασύρματες τεχνολογίες έχουν επεκτείνει τη λειτουργικότητα της τηλεμετρίας από τη μηχανική, την καθαρή επιστήμη και την κατασκευή, σε προϊόντα που χρησιμοποιούνται καθημερινά, όπως ηλεκτρικοί μετρητές και οικιακές μονάδες θέρμανσης.

Τα πρωτόκολλα M2M είναι μια καλά καθορισμένη αρχιτεκτονική που αποτελείται από παραμέτρους επικοινωνίας και μοντέλα, τα οποία ανταλλάσσουν δεδομένα ή πληροφορίες μέσω δικτύων. Κάθε πρωτόκολλο καθορίζει το μέγεθος του πακέτου, τους κανόνες επικοινωνίας, την ασφάλεια μέσω δικτύου, τις απαιτήσεις επικοινωνίας και άλλες ιδιότητες του δικτύου M2M, έτσι ώστε να βοηθήσει στη σύνδεση συσκευών χαμηλής ισχύος, στον κόσμο του Διαδικτύου. Στο παρακάτω σχήμα (1) απεικονίζονται τα διάφορα επίπεδα επικοινωνίας του Inetrent of Things (IoT), στο οποίο περιλαμβάνεται η έννοια M2M [5].



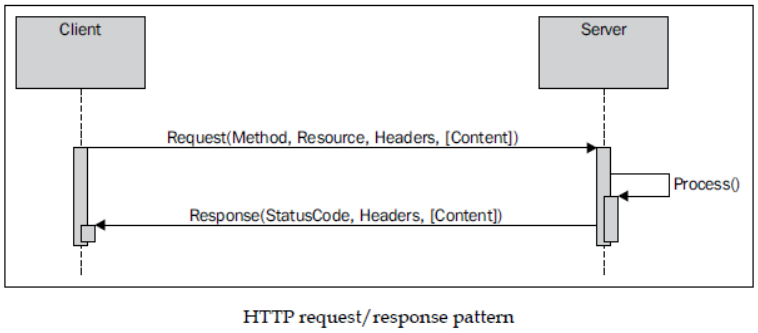
Εικόνα 1. Επικοινωνία M2M εν συντομία

Υπάρχουν τρεις μεγάλες ομάδες πρωτοκόλλων που χρησιμοποιούνται για την επικοινωνία M2M:

1) Οι **Service-Oriented Architectures (SOA)** χρησιμοποιούνται σε βιομηχανικά συστήματα αυτοματισμού για την ανταλλαγή ευαίσθητων δεδομένων σε πραγματικό χρόνο, για παράδειγμα μεταξύ προγραμματιζόμενων ελεγκτών λογικής και συστημάτων εποπτείας, ελέγχου και απόκτησης δεδομένων (supervisory, control and Data Acquisition - SCADA).
2) Στυλ αρχιτεκτονικής **Representational State Transfer (REST)** που καθορίζει τους περιορισμούς στα χρησιμοποιημένα στοιχεία, τους συνδέσμους και τα στοιχεία δεδομένων.
3) Τα **Πρωτόκολλα Προσανατολισμένα στο Μήνυμα (Message Oriented Protocols)** υποστηρίζουν την ασύγχρονη μεταφορά δεδομένων μεταξύ του κατανεμημένου συστήματος.

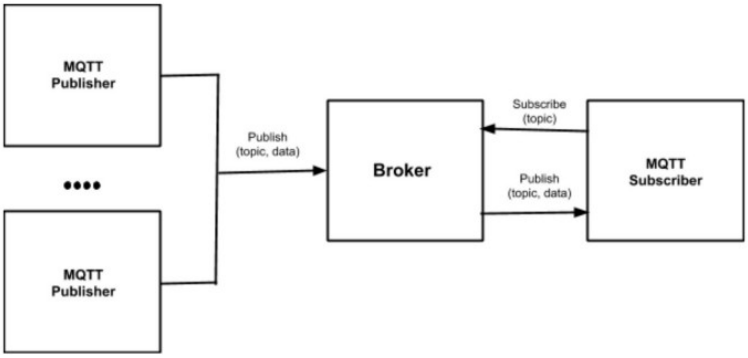
Σύμφωνα με τα μοντέλα επικοινωνίας:

1. Το **μοντέλο Request / Response ή client / server**, που χρησιμοποιείται συνήθως στα κατανεμημένα συστήματα για την ανταλλαγή πληροφοριών μέσω μηνυμάτων, τα οποία περνάνε μεταξύ αποστολέα και παραλήπτη. Εκεί ο client ζητά πληροφορίες από έναν διακομιστή (server) και ο διακομιστής ανταποκρίνεται ανάλογα σε αυτά τα αιτήματα όπως φαίνεται και στην Εικόνα 2. Παραδείγματα: HTTP, CoAP [6].



Εικόνα 2. Πρότυπο αίτησής / απόκρισης μέσω HTTP

2. Το μοντέλο **Publish/Subscribe** στην εικόνα 3, το οποίο βασίζεται σε έναν event broker για την προώθηση ενημερώσεων (ειδοποιήσεις-notifications) σε ενδιαφερόμενους χρήστες (συνδρομητές-subscribers), σχετικά με τις αλλαγές των καταστάσεων των αποστολέων (εκδοτών-publishers). Παράδειγμα: MQTT, AMQP.



Εικόνα 3. Μοντέλο Publish / Subscriber

Το Message Queuing Telemetry Transport (MQTT) είναι ένα πρωτόκολλο σύνδεσης M2M και IoT. Είναι ένα ανοιχτό πρωτόκολλο που προδιαγράφθηκε από την IBM και την Eurotech και πρόσφατα χρησιμοποιείται από το Eclipse σε εφαρμογές M2M.

Πρόκειται για ένα πρωτόκολλο επιπέδου εφαρμογής που ακολουθεί το πρότυπο επικοινωνίας publish-subscribe. Είναι ένα σχετικά απλό και ελαφρύ πρωτόκολλο ανταλλαγής μηνυμάτων, σχεδιασμένο για περιορισμένες συσκευές με περιορισμένη μπαταρία, επεξεργαστή ή / και πόρους μνήμης.

Το MQTT βασίζεται στο TCP / IP, αλλά σε αντίθεση με αυτό, οι απαιτήσεις του σε πόρους είναι σημαντικά μικρότερες. Η γενική μεταφορά μικρών μηνυμάτων (σταθερό μήκος 2 bytes) καθιστά το MQTT μια ενδιαφέρουσα λύση για αναξιόπιστα δίκτυα με περιορισμένους πόρους, όπως χαμηλό εύρος ζώνης και υψηλές καθυστερήσεις.

Το MQTT χρησιμοποιείται επίσης από το Facebook για την υλοποίηση ενός γρήγορου ελαφρού πρωτοκόλλου ασύγχρονης ανταλλαγής μηνυμάτων.

1.3.1 MQTT Message

Το μήνυμα MQTT περιέχει κεφαλίδα σταθερού μήκους 2 byte και προαιρετική κεφαλίδα μεταβλητού μήκους και ωφέλιμου φορτίου. Στο πρώτο byte της σταθερής κεφαλίδας των 2-bytes, περιέχει πεδίο των 4 bits για τον τύπο του μηνύματος που υποδηλώνει μια ποικιλία μηνυμάτων όπως CONNECT, PUBLISH, SUBSCRIBE. Τα υπόλοιπα 4 bits περιέχουν σημαίες (flags) ελέγχου, συμπεριλαμβανομένου αυτού για το επιθυμητό επίπεδο QoS για διασφάλιση παράδοσης των μηνυμάτων δημοσίευσης, DUP και RETAIN. Το δεύτερο byte καθορίζει το μήκος του μηνύματος [7].

1.3.2 Χαρακτηριστικά του MQTT

1. Επικοινωνία Δημοσίευσης – Εγγραφής

2. Ποιότητα υπηρεσιών (QoS)

3. Διατήρηση μηνυμάτων

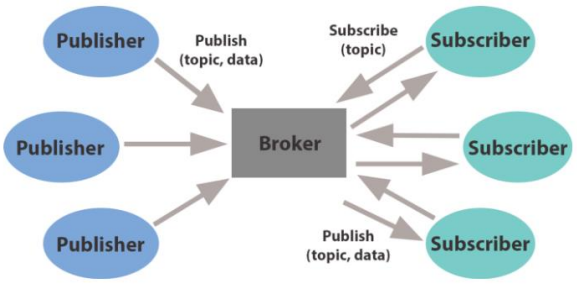
4. Τελευταία Επιθυμία (Last Will and Testament)

1.3.2.1 Δημοσίευση – Εγγραφή

Η αρχιτεκτονική της επικοινωνίας MQTT με βάση τη δημοσίευση-εγγραφή φαίνεται στην Εικόνα 4, όπου:

1. Ο Broker ενεργεί ως μεσάζων για μηνύματα που αποστέλλονται από εκδότη σε συνδρομητές, παρέχοντας διανομή μηνυμάτων από ένα προς πολλά και αποσύνδεση της εφαρμογής σε μερικές περιπτώσεις χρήσης.

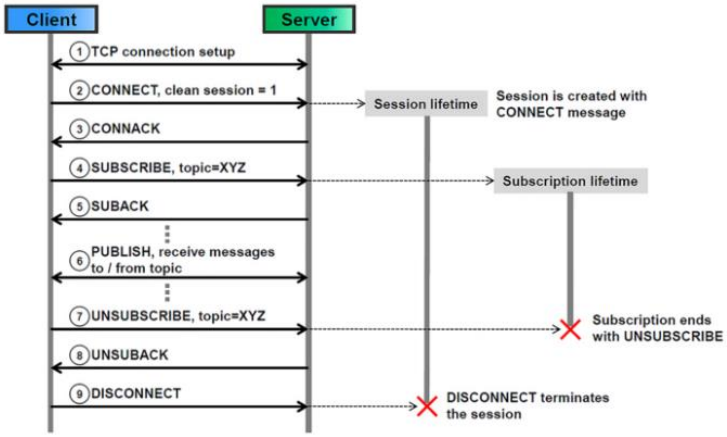
2. Όταν ένας "Εκδότης MQTT" δημοσιεύει ένα μήνυμα "δεδομένα" σε ένα συγκεκριμένο "θέμα", τότε όλοι οι "Συνδρομητές MQTT" που έχουν εγγραφεί στο "θέμα" θα λάβουν το μήνυμα "δεδομένα".



Εικόνα 4. MQTT Μοντέλο Publish / Subscriber

1.3.2.2 Διάγραμμα αλληλουχίας επικοινωνίας

Αρχικά, η σύνδεση TCP δημιουργείται μεταξύ πελάτη και μεσίτη / διακομιστή (broker). Υπάρχουν 16 δυνατοί τύποι μηνυμάτων στους οποίους βασίζεται ο πελάτης-μεσίτης MQTT για την αποστολή και λήψη μηνυμάτων και επιβεβαιώσεων. Η Εικόνα 5 δείχνει τη δημιουργία σύνδεσης, τη διαχείριση συνεδρίας και τον τερματισμό του καναλιού επικοινωνίας.

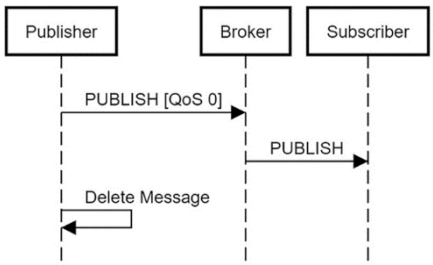


Εικόνα 5. Παρουσίαση μιας MQTT επικοινωνίας μέσω διαγράμματος αλληλουχίας

1.3.2.3 Ποιότητα υπηρεσιών (QoS)

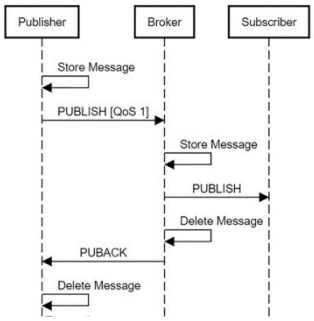
Το MQTT παρέχει άλλο ένα επίπεδο αξιοπιστίας χρησιμοποιώντας τρία επίπεδα QoS που ονομάζονται QoS επίπεδο 0, QoS επίπεδο 1 και QoS επίπεδο 2.

QoS 0 - Το πολύ μία φορά (fire and forget): Ένα απλό αίτημα αποστέλλεται χωρίς να περιμένεται απάντηση, αφήνοντας τη διασφάλιση παράδοσης στο TCP (Εικόνα 6).



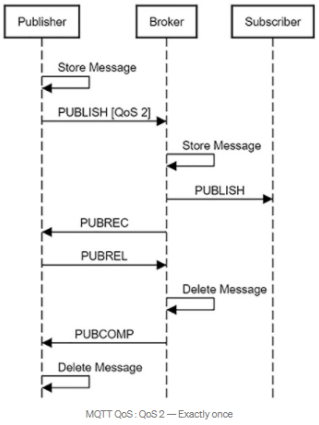
Εικόνα 6. MQTT QoS 0 - το πολύ μία φορά

QoS 1 – Τουλάχιστον μία φορά: Στο QoS επίπεδο 1, το μήνυμα ακολουθείται από απόκριση ACK. Σε περίπτωση απώλειας του ACK, το μήνυμα θα σταλεί ξανά.



Εικόνα 7. MQTT QoS 1 – Τουλάχιστον μια φορά

QoS 2 - Ακριβώς μια φορά: Στο επίπεδο QoS 2, πρέπει να δημιουργηθεί μια χειραψία τεσσάρων κατευθύνσεων για τη διασφάλιση της παράδοσης μηνυμάτων μόνο μία φορά (Εικόνα 8).



Εικόνα 8. MQTT QoS 2 – Ακριβώς μια φορά

1.3.2.4 Διατήρηση μηνυμάτων

Ο διακομιστής MQTT έχει την επιλογή να διατηρήσει το τελευταίο μήνυμα σε ένα θέμα, το οποίο αποστέλλεται σε νέο συνδρομητή. Αυτό το χαρακτηριστικό του πρωτοκόλλου MQTT επιτρέπει στο διακομιστή να ενεργεί ως «πίνακας βαθμολογίας», ο οποίος θα εμφανίζει την τελευταία ενημερωμένη τιμή μιας συγκεκριμένης ανάγνωσης αισθητήρα μόνο εάν δηλωθεί ως θέμα.

1.3.2.5 Τελευταία Επιθυμία (Last Will and Testament)

Το πρωτόκολλο διαθέτει επίσης έναν μηχανισμό διαθηκών, όπου οι εκδότες μπορούν να στείλουν ένα μήνυμα στον μεσίτη που θα σταλεί στους συνδρομητές τους μόνο όταν ο πελάτης αποσυνδεθεί. Μια απροσδόκητη αποσύνδεση από έναν εκδότη έχει ως αποτέλεσμα τη δημοσίευση του μηνύματος που είναι γνωστό ως η τελευταία βούληση, το οποίο υποδηλώνει ένα μη αναμενόμενο κλείσιμο. Αυτή η λειτουργικότητα καθιστά την ανίχνευση σφαλμάτων ευκολότερη, μεταδίδοντας και επιτρέποντας στους πελάτες να εκτελούν τις ρουτίνες σφάλματος ταυτόχρονα, τοποθετώντας το φορτίο της ανίχνευσης στο πρωτόκολλο και όχι στους διακομιστές ή τους πελάτες.

1.4 AD HOC Δίκτυα

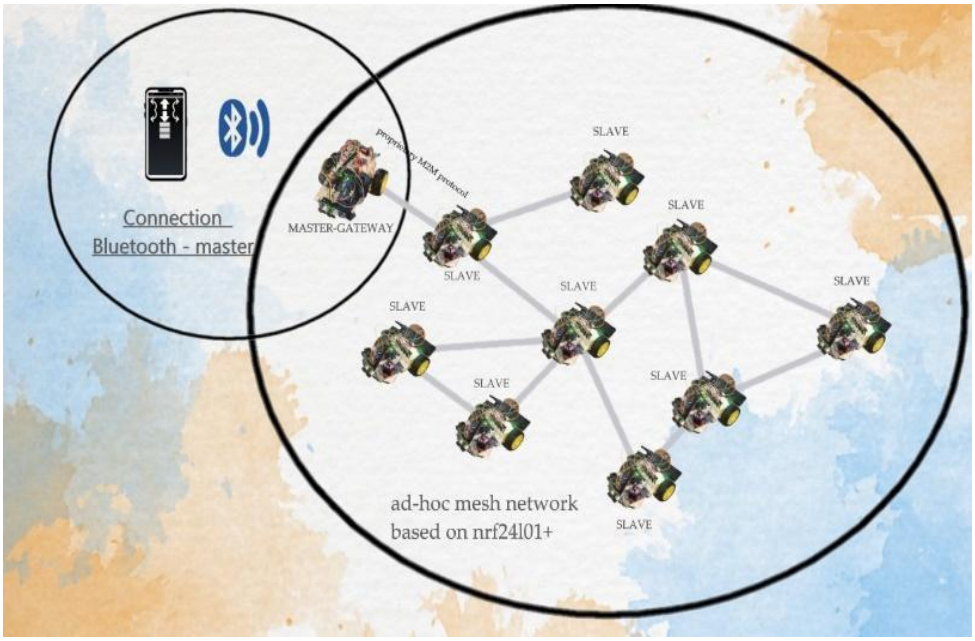
Ένα ασύρματο ad hoc δίκτυο (αυτοοργανωμένο δίκτυο ή δίκτυο κατ' απαίτηση) είναι ένας αποκεντρωμένος τύπος ασύρματου δικτύου. Τα ad hoc δίκτυα δε βασίζονται σε κάποια προϋπάρχουσα υποδομή, όπως οι δρομολογητές στα ενσύρματα δίκτυα ή τα ασύρματα access points στα διαχειριζόμενα ασύρματα δίκτυα. Αντίθετα, κάθε κόμβος λαμβάνει μέρος στη δρομολόγηση προωθώντας τα δεδομένα προς τους άλλους κόμβους, κι έτσι ο καθορισμός του ποιοι κόμβοι προωθούν δεδομένα γίνεται δυναμικά με βάση τη συνδεσιμότητα του δικτύου. Πέρα από την κλασική δρομολόγηση, τα ad hoc δίκτυα μπορούν να χρησιμοποιήσουν την υπερχείλιση για να προωθήσουν τα δεδομένα τους [8].

Τα πρώτα ασύρματα ad hoc δίκτυα ήταν τα PRNETs (δίκτυα "ραδιοφωνικών πακέτων") από τη δεκαετία του '70, υπό την αιγίδα του DARPA (Defense Advanced Research Projects Agency) μετά το ALOHAnet project.

2. Σχεδιασμός Υλικού και Ανάλυση

2.1 Σχεδιασμός Υλικού

Σε αυτό το κεφάλαιο θα αναλύσουμε τα στάδια του σχεδιασμού της διπλωματικής εργασίας από πλευράς υλικών και λειτουργιών. Η κεντρική ιδέα την εργασίας χτίστηκε έχοντας πάρει ως δεδομένο ότι θέλουμε να ασχοληθούμε με αυτοκινούμενα οχήματα. Αρχικά οι βασικότερες λειτουργίες θα είναι η κίνηση των οχημάτων και η μεταφορά δεδομένων μεταξύ τους (Εικόνα 9). Σημαντική προσθήκη αποτελεί η απόφαση μας να υλοποιήσουμε έναν αλγόριθμο αποφυγής εμποδίων. Έχοντας αυτά κατά νου προχωράμε στην εξεύρεση του κατάλληλου εξοπλισμού και εξαρτημάτων ξεκινώντας από τον «σκελετό» της εργασίας.



Εικόνα 9. Οπτικοποίηση Σχεδιασμού

Ο «σκελετός» της εργασίας είναι ένα κλασσικό “kit” με 2 μοτέρ το οποίο είναι πολύ συνηθισμένο σε projects με Arduino. Υπάρχει στην αγορά με διάφορα ονόματα επειδή είναι ένα OEM (Original equipment manufacturer) προϊόν που σημαίνει ότι η κάθε εταιρία μπορεί να αγοράσει ποσότητες από αυτό και να το πουλήσει με την δική της επωνυμία. Το δικό μας ονομάζεται “Robot Car 2WD” και θεωρητικά για την εργασία χρειαζόμαστε αρκετά από αυτά αλλά εν χάρη συντομίας και κόστους μπορούμε να έχουμε τα ίδια αποτελέσματα χρησιμοποιώντας μονάχα δύο.

Συνεχίζουμε με τον «εγκέφαλο» του συστήματος ο οποίος δεν είναι άλλος παρά το Arduino board. Υπάρχουν πολλές διαφορετικές υλοποιήσεις από την εταιρία όσον αφορά τις πλακέτες, αλλά καταλήξαμε στην πλακέτα Uno, η οποία είναι από τις πιο δημοφιλείς. Εκεί θα φιλοξενηθεί ο κώδικας που θα ελέγχει την κίνηση του κάθε οχήματος και θα επεξεργάζεται τα δεδομένα από τους αισθητήρες.

Για να επιτευχθεί η κίνηση των 2 μοτέρ συνεχούς ρεύματος (dc motors) που περιλαμβάνονται στο kit του οχήματος είναι απαραίτητη η χρήση μιας «Η-γέφυρας» για το κάθε μοτέρ. Η “L298N Dual H-Bridge Motor Driver” αποδείχτηκε το καταλληλότερο ολοκληρωμένο κύκλωμα επειδή διαθέτει δύο «Η-γέφυρες», μια για κάθε μοτέρ.

Επειδή η ταχύτητα περιστροφής των μοτέρ συνεχούς ρεύματος δεν είναι πάντα σταθερή και οι παράμετροι που την καθορίζουν ποικίλουν, για το κάθε μοτέρ θα χρειαστούμε έναν “Rotary encoder” ώστε να λύσουμε το πρόβλημα συγχρονισμού μεταξύ των τροχών. Το κύκλωμα που επιλέχθηκε για την μέτρηση της ταχύτητας του κάθε μοτέρ είναι το “HC-020K”. Αυτό το εξάρτημα θα μας φανεί χρήσιμο και στην λειτουργία αυτόματης αποφυγής εμποδίων διότι μας επιτρέπει να κινούμε τους τροχούς με μεγάλη ακρίβεια.

Για την επικοινωνία μεταξύ των οχημάτων θα χρησιμοποιήσουμε ένα επίσης δημοφιλές εξάρτημα το οποίο μας επιτρέπει τη δημιουργία δικτύων. Χρησιμοποιώντας περισσότερα από ένα nRF24L01, τα οποία επικοινωνούν στις ISM συχνότητες 2.4 - 2.5 GHz, μπορούμε να μεταφέρουμε αποτελεσματικά τα δεδομένα του πρωτοκόλλου που θα δημιουργήσουμε.

Τα δεδομένα κατεύθυνσης που θα διανέμονται στο σύστημα αποφασίστηκε να προέρχονται από ένα οποιοδήποτε κινητό με λειτουργικό σύστημα Android. Η διασύνδεση του με το όχημα που θα διαβιβάζει την πληροφορία στα υπόλοιπα οχήματα θα πρέπει να διαθέτει ένα Bluetooth module. Το εξάρτημα που επιλέχθηκε τελικά είναι το HC-05 Bluetooth Module. Η σύνδεση θα επιτευχθεί με τη δημιουργία και χρήση μιας εφαρμογής android.

Μια από τις λειτουργίες του συστήματος μας συμφωνήθηκε να είναι η αποφυγή εμποδίων. Για να γίνει αυτό εφικτό θα χρειαστούμε κάποιου είδους εξάρτημα το οποίο θα μας προειδοποιεί όταν κάτι εμποδίζει την πορεία του συστήματος. Αυτό ακριβώς βρήκαμε στον αισθητήρα υπερήχων HC-SR05, ο οποίος μπορεί και μετράει ακατάπαυστα την απόσταση του από τα αντικείμενα μπροστά του.

2.2 Λίστα Εξαρτημάτων Και Σχεδιάγραμμα Αρχιτεκτονικής

Σε αυτό το μέρος του κεφαλαίου θα συνοψίσουμε σε δύο πίνακες ότι ακριβώς χρησιμοποιήθηκε για την κατασκευή του εν λόγω συστήματος. Ο πρώτος πίνακας περιέχει τα εξαρτήματα που έχουμε χρησιμοποιήσει για να κατασκευάσουμε το σύστημα της παρούσας διπλωματικής το οποίο περιέχει δύο οχήματα. Ο δεύτερος πίνακας περιέχει τα εξαρτήματα που πρέπει να περιλαμβάνει το κάθε όχημα μεμονωμένα, ακολουθούμενος από τα σχεδιαγράμματα συναρμολόγησης.

Πίνακας 1

Πίνακας που περιέχει συνολικά ότι χρειάστηκε για την διπλωματική εργασία.

Υλικά	Ποσότητα
Arduino Uno	2
HC-020K	4
L298N	2
nRF24L01	2
Robot Car 2WD	2
HC-SR05	2
HC-05 Bluetooth Module	1
Μπαταριά	2
Android με Bluetooth	1

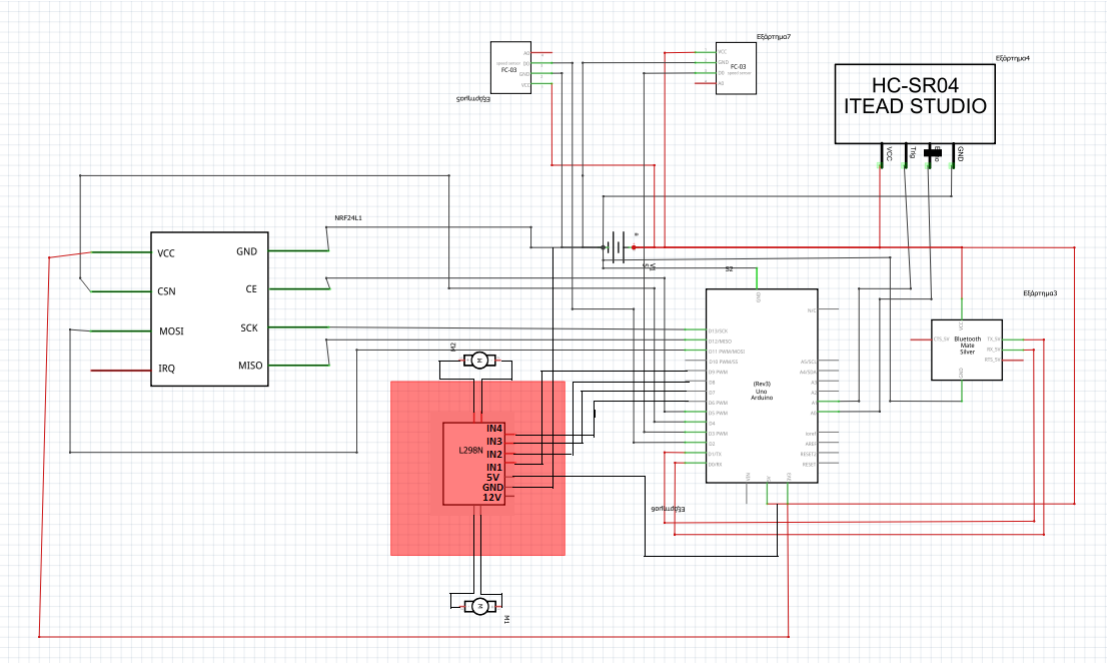
Πίνακας 2

Πίνακας που περιέχει τα εξαρτήματα κάθε οχήματος.

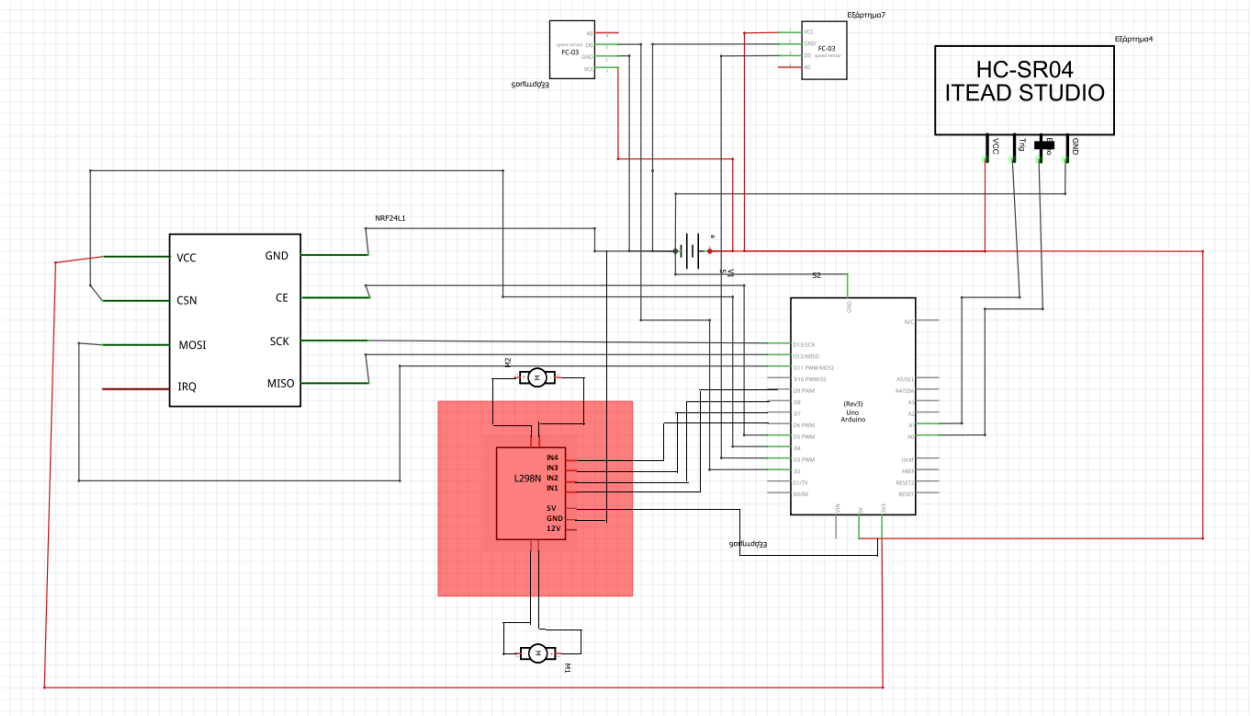
Υλικά	Ποσότητα
Arduino Uno	1
HC-020K	2
L298N	1
nRF24L01	1
Robot Smart Car 2WD	1
HC-SR05	1
HC-05 Bluetooth Module	1 (προαιρετικό)
Μπαταριά	1

Απαιτείται τουλάχιστον ένα όχημα με το εξάρτημα HC-05 για να είναι εφικτός ο χειρισμός του συστήματος μέσω κινητού τηλεφώνου. Στα υπόλοιπα οχήματα είναι προαιρετική προσθήκη και θα ήταν χρήσιμη σε περίπτωση που θέλουμε να αλλάξουμε το όχημα που επιλέγουμε να διανέμει την πληροφορία στα υπόλοιπα, με κάποιο άλλο.

Στις Εικόνες 10 και 11 βλέπουμε την ακριβή αναπαράσταση της συνδεσμολογίας μεταξύ των εξαρτημάτων με και χωρίς HC-05.



Εικόνα 10. Σχεδιάγραμμα οχήματος με HC-05.



Εικόνα 11. Σχεδιάγραμμα οχήματος ΧΩΡΙΣ HC-05.

2.3 Ανάλυση Και Δυνατότητες Εξαρτημάτων

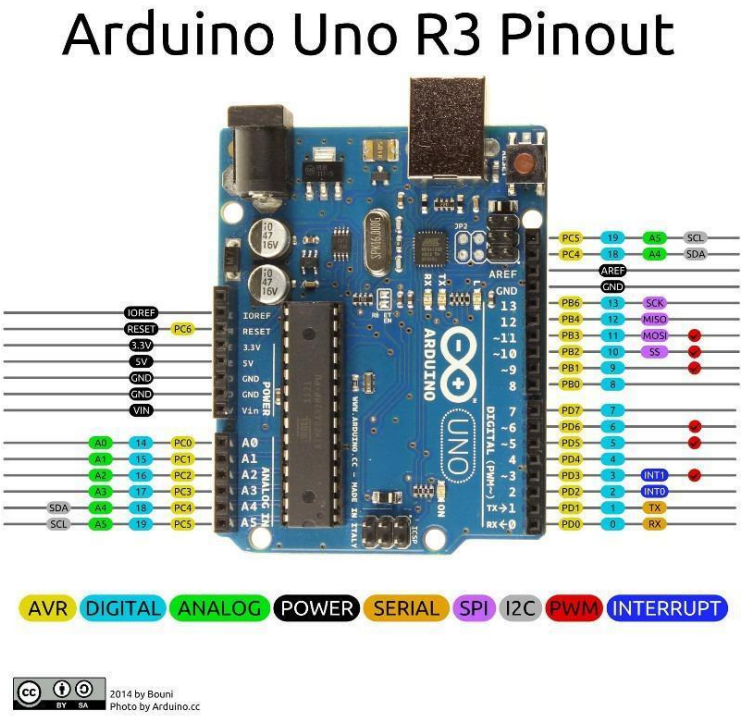
2.3.1 Arduino Uno

Με την ονομασία Arduino αναφερόμαστε στην αναπτυξιακή πλατφόρμα ανοιχτού κώδικα που δημιούργησαν οι Massimo Banzi και David Cuartielles. Σε ένα γενικό πλαίσιο αποτελείται από ένα ηλεκτρονικό κύκλωμα χτισμένο γύρω από κάποιον μικροεπεξεργαστή καθώς και ένα IDE (αναπτυξιακό περιβάλλον) που απλοποιεί τη διεπαφή μας με τον μικροεπεξεργαστή. Προγραμματίζεται σε επίπεδο χρήστη με μία ψευδογλώσσα που είναι βασισμένη στη C++ και κατ' επέκταση της μοιάζει πολύ. Δημιουργήθηκε σα λύση για το πρόβλημα των δύσχρηστων και ακριβών πλατφορμών που υπήρχαν έως τότε. Μέχρι σήμερα έχει διαμορφωθεί, έχει μετεξελιχθεί και έχει εμφανιστεί σε πολλές παραλλαγές ώστε να ικανοποιήσει τις ανάγκες των χρηστών.

Στην πλειοψηφία τους είναι βασισμένα στην οικογένεια μικροεπεξεργαστών ATmega της Atmel, την οποία εξαγόρασε η Microchip το 2016, κυρίως λόγω χαμηλού κόστους σε σχέση με τις δυνατότητές τους, μεγάλης διαθεσιμότητας και ευκολίας στην πρόσβασή τους. Ο πιο διαδεδομένος μικροεπεξεργαστής είναι ο ATmega328 λόγω της χρήσης του στις πιο δημοφιλείς πλακέτες Arduino.

Η πλακέτα μας είναι, στην ουσία, ένα breakout board ενός μικροεπεξεργαστή έτσι ώστε να έχει ο χρήστης εύκολη πρόσβαση στα I/O pins του, συν τα υποστηρικτικά κυκλώματα που είναι απαραίτητα για τη λειτουργία του, τη σύνδεση με τον υπολογιστή (συνήθως μέσω serial USB ή ICSP), κλπ.

Ας δούμε λίγο αναλυτικότερα την ανατομία του Arduino UNO (Εικόνα 12) που είναι και η δημοφιλέστερη αναπτυξιακή πλατφόρμα.



Εικόνα 12. Arduino Uno R3

- ***ΚΥΚΛΩΜΑ ΤΡΟΦΟΔΟΣΙΑΣ.***

Το κύκλωμα τροφοδοσίαςείναι βασισμένο σε ένα γραμμικό regulator με ηλεκτρολυτικούς πυκνωτές εξομάλυνσης στην είσοδο και την έξοδο της τροφοδοσίας καθώς και μια δίοδο για προστασία από την ανάστροφη πολικότητα. Είναι υπεύθυνο να μετατρέπει τα volt εισόδου σε 5V για την τροφοδοσία του μικροεπεξεργαστή και σε 3.3V για την τροφοδοσία εξαρτημάτων που το απαιτούν.

- ***Η ΠΡΩΤΗ ΣΕΙΡΑ I/O ΘΥΡΩΝ ΚΑΙ PINS ΤΡΟΦΟΔΟΣΙΑΣ.***

Έχουμε κατά σειρά:

- RESET pin που επανεκκινεί τον Μ/Ε μας μόλις γειωθεί στιγμιαία.
- 3.3V-5V- GND pins τα οποία μας προσφέρουν τις ομώνυμες τάσεις και γείωση.
- Vin pin που μας δίνει εναλλακτικό τρόπο τροφοδοσίας της πλακέτας.

- ***ΔΕΥΤΕΡΗ ΣΕΙΡΑ I/O ΘΥΡΩΝ.***

Έξι συνολικά θύρες που είναι ικανές να "διαβάσουν" ή να "γράψουν" αναλογικές τάσεις από 0V έως και 5V. Μπορούν να παίξουν διττό ρόλο αναλογικών και ψηφιακών I/O θυρών.

- ***Ο ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΗΣ.***

Στην περίπτωση του Arduino UNO πρόκειται για έναν 8-bit επεξεργαστή ATmega328P κατασκευασμένο από την ATMEL. Είναι βασισμένος σε αρχιτεκτονική RISC και είναι ικανός για 20 MIPS (εκατομμύρια οδηγίες ανά δευτερόλεπτο) όταν είναι χρονισμένος στα 20MHz που είναι και η μέγιστη ταχύτητά του. Να σημειώσουμε πως κατασκευαστικά έχει εσωτερικό κρύσταλλο που τον οδηγεί στα 8MHz αλλά είθισται να οδηγούμε τον Μ/Ε στα

16MHz που είναι ένας ικανοποιητικός συμβιβασμός . Έχει μνήμη Flash μεγέθους 32kb, και μνήμη SRAM της τάξεως των 2kb. Παράγεται σε πακέτο ολοκληρωμένων τύπου PDIF, MLF και TQFP με τα δυο πρώτα να αξιοποιούνται στις αναπτυξιακές πλακέτες arduino. Τέλος από τα 28 συνολικά pins του ολοκληρωμένου πακέτου, τα 23 μπορούν να χρησιμοποιηθούν σαν είσοδοι - έξοδοι (I/O pins). Διπλά στον μικροεπεξεργαστή μπορούμε να διακρίνουμε και τα pins για προγραμματισμό με χρήση ICSP μεθόδου.

- ***ΚΥΚΛΩΜΑ USB - UART(UNIVERSAL ASYNCHRONOUS RECEIVER/TRANSMITTER).***

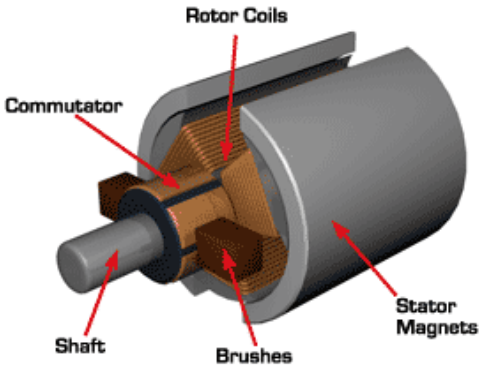
Είναι το κομμάτι που αναλαμβάνει την επικοινωνία του Arduino με τον υπολογιστή μας μέσω USB. Το ρόλο αυτό μπορεί να τον παίζει ένα FTDI ολοκληρωμένο, κάποιος κλώνος του, ή ακόμα και ένας δεύτερος μικροεπεξεργαστής, συνήθως ο ATmega16u2. Στην τελευταία περίπτωση προσφέρεται ακόμα ένα σετ pins για ICSP προγραμματισμό του δεύτερου microcontroller.

- ***Η ΤΕΛΕΥΤΑΙΑ ΣΕΙΡΑ ΜΕ PINS ΕΙΣΟΔΟΥ/ΕΞΟΔΟΥ.***

Πρόκειται για τα ψηφιακά I/Os που μπορούν να λειτουργήσουν μόνο με λογική 0 και 1. Κάποια από αυτά, που είναι μαρκαρισμένα με το σύμβολο "~" είναι ικανά να παράσχουν pwm παλμό.

2.3.2 DC-Motors

Στην κατασκευή μας αξιοποιούμε 4 συνολικά μοτέρ συνεχούς ρεύματος. Είναι ηλεκτροκινητήρες που μετατρέπουν την ηλεκτρική ενέργεια σε κινητική (Εικόνα 13). Στη μορφή που χρησιμοποιούμε εμείς, έχουμε 2 μαγνήτες πακτωμένους στο εσωτερικό του κελύφους με ανάποδη πολικότητα μεταξύ τους και έναν άξονα με ένα δαχτυλίδι συλλέκτη που συνδέεται με δύο τυλίγματα στο δρομέα και που έρχεται σε μόνιμη επαφή με δύο ψήκτρες (καρβουνάκια). Ελέγχοντας την τάση, ελέγχουμε και την ταχύτητα περιστροφής του άξονα και αλλάζοντας την πολικότητα, αλλάζουμε τη φορά περιστροφής. Τα μοτέρ που διατίθενται στο kit με το όχημα έχουν ονομαστική τάση λειτουργίας τα 5V αλλά είναι πολύ ελαστικά στις ανοχές τους.

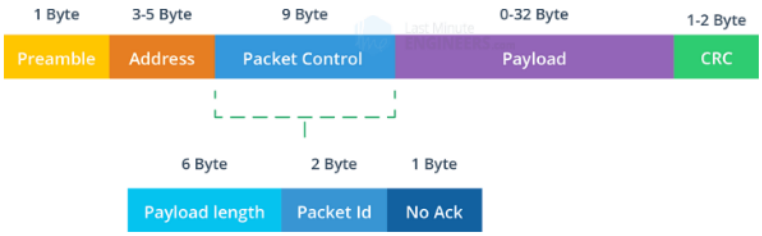


Εικόνα 13. Αρχιτεκτονική Dc-Motor

2.3.3 NRF tranceiver modules

Στην υλοποίησή μας, χρησιμοποιήσαμε δύο modules πομποδέκτες που είναι βασισμένα στο ολοκληρωμένο nRF24L01. Επικοινωνούν μεταξύ τους μέσω κατά GFSK διαμορφωμένων πακέτων στη συχνότητα των 2.4GHz. Χρησιμοποιούν μια εσωτερική μηχανή από τη Nordik, που είναι η κατασκευάστρια εταιρία, την ShockBurst. Τα nRF24L01 χρησιμοποιούν κάποιο κανάλι από 2400 MHz έως 2525 MHz με βήμα 1MHz, συνολικά δηλαδή έχουμε

125 διαθέσιμα κανάλια και για να επικοινωνήσουν 2 NRF πρέπει να εκπέμπουν στο ίδιο κανάλι. Αν χρειαστεί να χρησιμοποιήσουμε μεγαλύτερο data rate στα 2Mbs, τα βήματα θα πρέπει να είναι της τάξεως των 2MHz τουλάχιστον, για να αποφύγουμε το crosstalk σε γειτονικές συχνότητες. Προσφέρει τη δυνατότητα σύνδεσης έξι διαφορετικών NRFmodules σε ένα, μέσω έξι λογικά χωρισμένων διαύλων (pipes) ανά κανάλι, φτιάχνοντας έτσι ένα δίκτυο με διακριτά nodes. Στην Εικόνα 14, διακρίνουμε τη δομή ενός πακέτου έτσι όπως δημιουργείται από τη μηχανή ενός Nordik [9].



Εικόνα 14. Βελτιωμένη δομή πακέτων ShockBurst.

Έχουμε αρχικά ένα πρόθεμα μεγέθους 1byte. Έπειτα τη διεύθυνση αποστολής με μέγεθος τουλάχιστον 3bytes και έως 5bytes. Το επόμενο ονομάζεται PCF ή Packet Control Field και πρόκειται για πρόσφατη προσθήκη στη δομή του πακέτου. Το PCF α) περιέχει πληροφορίες για το μέγεθος του payload ενός πακέτου, καθιστώντας το έτσι δυναμικό, β) αναθέτει ένα ID στο πακέτο για αποφυγή διπλότυπων και εντοπισμό σφαλμάτων και τέλος γ) δίνει πλέον την επιλογή να ζητάμε ACK (acknowledgement), επιβεβαίωση δηλαδή από τον δέκτη για τη λήψη του πακέτου, κάτι που στο προηγούμενο πρότυπο δομής ήταν αυτόματο.

Το πακέτο ACK είναι μια εξαιρετικά χρήσιμη λειτουργία η οποία αφενός δίνει τη δυνατότητα να εισάγουμε δεδομένα στο payload του πακέτου επιβεβαίωσης τα οποία λαμβάνει ο πομπός χωρίς να χρειαστεί να σταματήσει τη διαδικασία αποστολής, όπως κανονικά θα συνέβαινε σε μια τυπική αμφίδρομη σύνδεση δύο nRF24L01. Τα δεδομένα αυτά είναι διαθέσιμα μόνο για ανάγνωση από τον πομπό και πρέπει να είναι pre-injected στο payload από το δέκτη, κάτι που σημαίνει πως δε μπορούμε για παράδειγμα να αποστείλουμε δεδομένα και να τα παραλάβουμε επεξεργασμένα στο ACK payload. Η λειτουργία αυτή δυστυχώς είναι ελλιπέστατα βιβλιογραφημένη λόγω των ελάχιστων χρηστών που την αξιοποιούν, καθώς και λόγω των σχετικών ελλείψεων στις NRF βιβλιοθήκες του Arduino. Ο μόνος λόγος να την απενεργοποιήσουμε είναι η περίπτωση που ζητάμε το μέγιστο δυνατό ρυθμό μετάδοσης δεδομένων, χωρίς να μας ενδιαφέρει ο έλεγχος σφαλμάτων.

Επόμενο κατά σειρά κομμάτι στη δομή του πακέτου, είναι το payload, τα δεδομένα δηλαδή που θέλουμε να επικοινωνήσουμε. Το payload έχει πλέον δυναμικό μέγεθος από 0bytes έως 32bytes, ενώ στη προηγούμενη δομή είχε στατικό μέγεθος 32bytes.

Το τελευταίο κομμάτι της δομής του πακέτου περιέχει πληροφορίες για το CRC πολυώνυμο που χρησιμοποιείται για τον έλεγχο της ακεραιότητας των δεδομένων.

Σε μια επιτυχημένη λοιπόν μετάδοση δεδομένων γίνονται τα εξής βήματα:

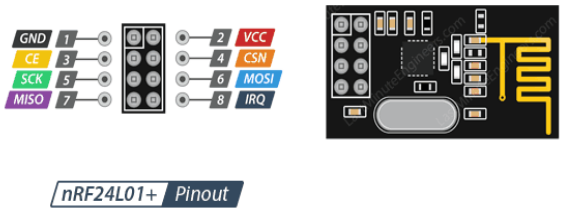
- Σχηματίζεται το πακέτο σύμφωνα με τα δεδομένα του πομπού.
- Ο πομπός αποστέλλει το πακέτο και εισέρχεται σε κατάσταση αναμονής για 130μs.
- Ο δέκτης λαμβάνει το πακέτο και με τη σειρά του απαντάει στον πομπό στέλνοντας το ACK πακέτο.
- Ο πομπός λαμβάνει το ACK πακέτο και συνεχίζει με την αποστολή των επόμενων.

Σε περίπτωση που ο πομπός δε λάβει απάντηση στο προκαθορισμένο χρονικό διάστημα, αποστέλλει ξανά το πακέτο. Τον αριθμό επαναπροσπαθειών και το χρόνο αναμονής του πομπού μπορούμε να τον παραμετροποιήσουμε εμείς σαν χρήστες μέσω του κώδικα. Μια ακόμα περίπτωση είναι να λάβει ο δέκτης το πακέτο αλλά το ACK πακέτο να μη φτάσει έγκαιρα στον πομπό. Σε αυτή την περίπτωση, ο δέκτης μόλις πάρει για δεύτερη φορά το πακέτο, το απορρίπτει μετά από έλεγχο του ID καθώς το έχει ήδη λάβει και στέλνει ξανά το ACK package.

Εμείς, χρησιμοποιήσαμε μια εναλλακτική μορφή του module που μπορεί να επιτύχει μετάδοση σε μεγαλύτερες αποστάσεις. Οι κύριες διαφορές με το απλό module είναι πως έχει ενσωματωμένο βύσμα SMA για εξωτερική κεραία (Εικόνα 15) αντί για τυπωμένη, καθώς και έναν ενισχυτή σήματος PA/LNA ανάμεσα στο ολοκληρωμένο nRF24L01 και την κεραία.

Εικόνα 15. NRF24L01 Module με βύσμα SMA

Στην Εικόνα 16 μπορούμε να δούμε τη δομή ενός συμβατικού NRF module.

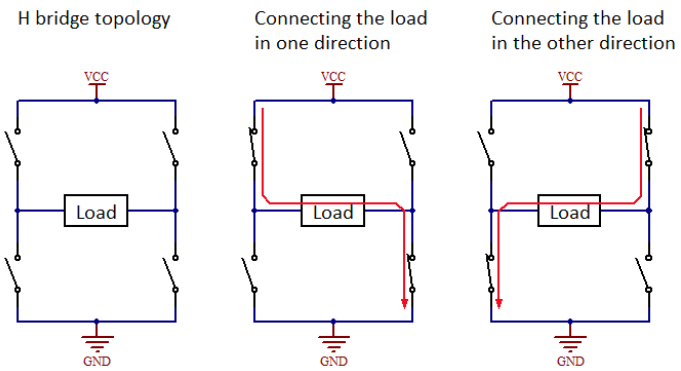


Εικόνα 16. Δομή ενός NRF Module.

Στο αριστερό κομμάτι βλέπουμε τα pins για ICSP συνδεσιμότητα με τον μικροελεγκτή μας, στο κέντρο βρίσκεται το ολοκληρωμένο nrf24l01 περιστοιχισμένο από τα παθητικά εξαρτήματα που χρειάζεται για τη λειτουργία του και έναν εξωτερικό ταλαντωτή συχνότητας 16MHz. Στο δεξιό κομμάτι φαίνεται καθαρά η διπολική κεραία που είναι τυπωμένη στο χαλκό της πλακέτας. Τα pins του module με εξαίρεση αυτά της τροφοδοσίας είναι ανθεκτικά σε τάση έως 5V. Η ονομαστική τάση τροφοδοσίας του ολοκληρωμένου είναι τα 3.3V.

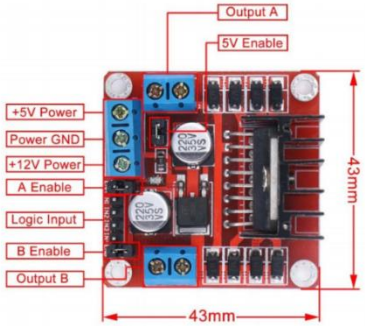
2.3.4 H-bridge βασισμένα στο L298N

Με τον όρο H-bridge αναφερόμαστε σε ένα συγκεκριμένο τύπο κυκλώματος που μας δίνει τη δυνατότητα να ελέγχουμε την πολικότητα του ρεύματος που καταλήγει σε μια έξοδο, ελέγχοντας τέσσερις συνολικά διακόπτες που ανοιγοκλείνουν ανά ζεύγη. Στην έξοδο συνήθως συνδέεται κάποιο DC μοτέρ, ώστε να ελέγχεται η φορά περιστροφής του ανάλογα με την πολικότητα του ρεύματος (Εικόνα 17). Στις συνηθέστερες εφαρμογές, ο έλεγχος του κυκλώματος θέλουμε να γίνεται από κάποιο μικροελεγκτή οπότε, φυσικό είναι να επιλέγουμε τρανζίστορ στο ρόλο των διακοπτών για να είναι εφικτός και σχετικά εύκολος ο έλεγχός τους από τον μικροελεγκτή μέσω παλμών ρεύματος μικρής έντασης. Θα μπορούσαμε να κάνουμε την ίδια υλοποίηση με ρελέ στη θέση των διακοπτών αλλά τα τρανζίστορ είναι προτιμότερα λόγω μεγέθους και ταχύτητας, εκτός αν έχουμε απαίτηση για ρεύμα μεγάλης έντασης.



Εικόνα 17. Καταστάσεις λειτουργίας Η-γέφυρας

Για τη διευκόλυνσή μας επιλέξαμε ένα module βασισμένο στο ολοκληρωμένο L298N. Πρόκειται για ένα IC που ενσωματώνει δύο H-bridge κυκλώματα ο έλεγχος των οποίων γίνεται με δύο ζεύγη λογικών inputs. Είναι ικανό να υποστηρίξει ρεύμα μέγιστης έντασης 2A και μέγιστης τάσης 46V και αξιοποιώντας τα δύο enable pins του, μπορούμε να ελέγξουμε μέσω pwm παλμών την τάση του ρεύματος στις τέσσερις εξόδους του, οπότε κατ' επέκταση και την ταχύτητα περιστροφής των DC κινητήρων. Η τροφοδότηση του ολοκληρωμένου γίνεται με τάση 5-12V. Το module αποτελείται από το ολοκληρωμένο L298N (Εικόνα 18) με την απαραίτητη παθητική ψύξη του, έναν γραμμικό ρυθμιστή τάσης 7805 για την τροφοδότηση του ολοκληρωμένου και στρατηγικά τοποθετημένους συνδέσμους για τη διευκόλυνσή μας στην ένωση με τον μικροελεγκτή. Κάναμε χρήση δύο ίδιων module για τον έλεγχο τεσσάρων συνολικά κινητήρων, ένα για κάθε όχημα.



Εικόνα 18. Ολοκληρωμένο L298N

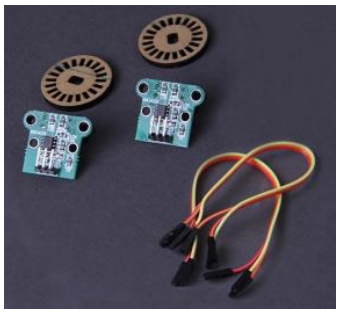
2.3.5 Robot Car 2WD

Το kit αυτό είναι σχεδιασμένο για project βασισμένα σε Arduino και περιέχει 2 μοτέρ τα οποία λειτουργούν στα 5V αλλά η ελαστικότητα τους τους επιτρέπει να φτάσουν από 3V μέχρι και τα 6V (εικόνα 19). Η αναλογία των περίστροφων του μοτέρ σε σχέση με τις ρόδες που περιλαμβάνονται στο kit είναι 1:48. Αυτό σημαίνει ότι για μια πλήρη περιστροφή της ρόδας το μοτέρ πρέπει να κάνει 48. Απαιτείται μια στοιχειώδης συναρμολόγηση του προϊόντος επειδή έρχεται σε κομμάτια.

Εικόνα 19. Robot Car 2WD kit

2.3.6 HC-020K Speed Measuring Module with Photoelectric Encoders

Όπως είπαμε και παραπάνω τα μοτέρ συνεχούς ρεύματος δεν μπορούν να λειτουργήσουν με ακρίβεια και πολλές φορές η ταχύτητα περιστροφής τους εξαρτάται από την τάση της μπαταρίας. Αυτό το πρόβλημα έρχεται να λύσει το HC-020K (Εικόνα 20). Ουσιαστικά αποτελείται από δύο μέρη. Το πρώτο είναι ο δίσκος με τις οπές ο οποίος είναι συνδεδεμένος στον άξονα περιστροφής της ρόδας. Το δεύτερο μέρος ουσιαστικά περιέχει έναν πομπό και ένα δεκτή, τον ένα απέναντι από τον άλλο, δίνοντας έτσι μια μορφή σαν Π στο εξάρτημα. Το εξάρτημα πρέπει να τοποθετηθεί έτσι ώστε ο δίσκος να παρεμβάλλεται ανάμεσα στον πομπό και στον δέκτη. Ο πομπός στέλνει συνεχώς σήμα προς τον δεκτή και έτσι ο δέκτης μπορεί να ξέρει αν κάτι παρεμβάλλεται μεταξύ του ιδίου και του πομπού. Άρα το εξάρτημα μπορεί να καταλάβει δύο κατάστασεις. Όταν δεν παρεμβάλλεται κάτι μεταξύ πομπού και δέκτη ξέρουμε ότι βρίσκεται σε οπή του δίσκου. Με την περιστροφή της ρόδας, άρα και του δίσκου, ο δέκτης αντιλαμβάνεται ένα πλήθος αλλαγών στις καταστάσεις του. Μετρώντας τις μεταβολές και ξέροντας πόσες οπές έχει ο δίσκος, μπορούμε να υπολογίσουμε την γωνιά που διέσχισε η ρόδα. Στην συγκεκριμένη περίπτωση δίσκος μας έχει είκοσι οπές. Αν γνωρίζουμε και την διάμετρο του τροχού μπορούμε να υπολογίσουμε και την απόσταση που διένυσε. Με μια απλή συνάρτηση χρονομέτρησης και ξέροντας την απόσταση που διανύει το όχημα μπορούμε να υπολογίσουμε και την ταχύτητα του. ΠΡΟΣΟΧΗ!! Το εξάρτημα αυτό δεν μπορεί να καταλάβει την κατεύθυνση του τροχού άρα καλό θα ήταν κάθε φορά που θέλουμε να αλλάξουμε την κατεύθυνση του οχήματος να ξεκινάμε καινούριες μετρήσεις. [10]

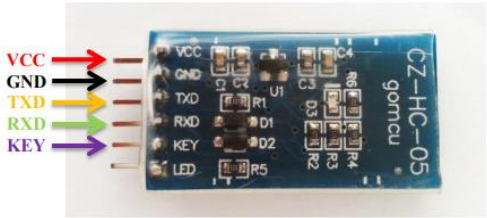


Εικόνα 20. Encoders HC-020K.

Εμείς αξιοποιούμε τις πληροφορίες του HC-020K με διάφορους τρόπους. Δεδομένης της αναξιοπιστίας των μοτέρ, τα οχήματα μας όταν κινούνταν σε ευθεία κατεύθυνση παρέκκλιναν λίγο από αυτή λόγω των διαφορετικών ταχυτήτων που παρήγαγαν. Έχοντας συνεχώς μια εικόνα για την ταχύτητα και την απόσταση που διανύει η κάθε ρόδα μπορέσαμε να κάνουμε μικροδιορθώσεις όπου χρειαζόταν σε πραγματικό χρόνο ώστε να πέτυχουμε μια ομαλή ευθεία πορεία. Μια ακόμα πολύ χρήσιμη και σημαντική λειτουργία είναι ο λεπτομερής έλεγχος του οχήματος. Λειτουργία που είναι απαραίτητη στην αποφυγή εμποδίων. Μας δίνει την δυνατότητα για παράδειγμα να αλλάξουμε με ακρίβεια τη γωνία κατεύθυνσης του οχήματος κατά ενενήντα μοίρες.

2.3.7 HC-05 Bluetooth Module

Το HC-05 είναι ένα Bluetooth Module (Εικόνα 21), το οποίο μας επιτρέπει τον έλεγχο των οχημάτων δημιουργώντας ένα δίαυλο επικοινωνίας μεταξύ ενός από αυτά και μιας android συσκευής. Χρησιμοποιεί το SPP (Serial Port Protocol), δηλαδή το πρωτόκολλο σειριακής θύρας, το οποίο καθιστά την σύνδεση με υπολογιστή ή Android πολύ εύκολη και υποστηρίζει αμφίδρομη επικοινωνία. Παρακάτω βλέπουμε τη δομή των pins και στο που χρησιμεύουν [11].



Εικόνα 21. HC-05 Bluetooth Module με σχεδιάγραμμα ακροδεκτών.

Pin	Λειτουργεία	Περιγραφή
VCC	+5V	Σύνδεση στα 5V
GND	Ground	Σύνδεση στην γείωση
TXD	UART_TXD, Pin για σήμα σειριακή αποστολή	Σύνδεση με RXD PIN στον μικροελεγκτή
RXD	UART_RXD, Pin για σήμα σειριακή απολαβή	Σύνδεση με TXD PIN στον μικροελεγκτή
KEY	Διακόπτης λειτουργίας	High= διακοπή λειτουργίας low= κανονική λειτουργία

Πίνακας 3. Πίνακας λειτουργιών και περιγραφής Bluetooth Module HC-05

2.3.8 HC-SR05

Το HC-SR05 (Εικόνα 22) είναι ένα εξάρτημα το οποίο επιτρέπει την εκπομπή και λήψη υπερηχητικών κυμάτων. Έτσι μπορεί να εκπέμψει ένα παλμό και στην συνέχεια να περιμένει να ανακλαστεί κάπου για να επιστρέψει και να ανιχνευθεί. Αν υπολογίσουμε τον χρόνο που πάει και έρχεται το κύμα και ξέροντας την ταχύτητα του ήχου, μπορούμε να βρούμε την απόσταση που διένυσε το κύμα από τη γέννησή του μέχρι και την ανίχνευσή του. Αν διαιρέσουμε αυτή την απόσταση δια δυο μπορούμε να καταλάβουμε σε τι απόσταση βρίσκεται το module από την επιφάνια που ανάκλασε το κύμα [12].

Εικόνα 22. Module HC-SR05.

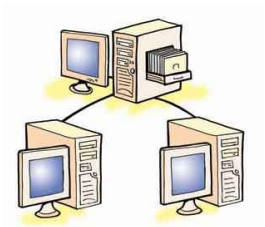
Pin	**Λειτουργία**	**Περιγραφή**
VCC	+5V	Σύνδεση στα 5V
GND	Ground	Σύνδεση στην γείωση
TRIG	10us TTL impulse	Ενεργοποιεί την συσκευή για να αρχίσει να ανιχνεύει
ECHO	Επιστέφει high	Όταν ανιχνεύσει πίσω το σήμα αποστρέφει ένα λογικό high.

Πίνακας 4. Πίνακας λειτουργιών και περιγραφής Module HC-SR05

3. Δίκτυα και Πρωτόκολλα

3.1 Τι Είναι Δίκτυο Υπολογιστών

Η ιστορία των δικτύων υπολογιστών ήρθε μαζί με την ανάπτυξη τους την δεκαετία του 1950. Η ανάγκη της επικοινωνίας μεταξύ των υπολογιστών οδήγησε στη δημιουργία τους. Ένα δίκτυο υπολογιστών είναι ένα σύστημα επικοινωνίας δεδομένων που συνδέει δύο ή περισσότερους αυτόνομους και ανεξάρτητους υπολογιστές και περιφερειακές συσκευές (Εικόνα 23). Δύο υπολογιστές θεωρούνται συνδεδεμένοι όταν μπορούν να ανταλλάξουν μεταξύ τους πληροφορίες [13]. Τα όρια των πρώτων δικτύων δεν ξεπερνούσαν το μέγεθος ενός δωματίου. Αυτού του είδους τα δίκτυα ονομάζονται προσωπικά δίκτυα. Οι πρώτες έννοιες του δικτύου ευρείας περιοχής προήλθαν από διάφορα εργαστήρια πληροφορικής στις Ηνωμένες Πολιτείες, το Ηνωμένο Βασίλειο και τη Γαλλία. Το 1969 στάλθηκε το πρώτο μήνυμα μέσω του ARPANET, από το εργαστήριο του Καθηγητή πληροφορικής Λέναρντ Κλέινροκ στο Πανεπιστήμιο της Καλιφόρνια, Λος Άντζελες (UCLA) προς το δεύτερο κόμβο του δικτύου στο Ερευνητικό Ίδρυμα του Στάνφορντ (SRI). Μαζί με τα δίκτυα έκαναν την εμφάνιση τους και τα πρωτόκολλα που ήταν αναγκαία για την επικοινωνία μεταξύ τον υπολογιστών. Αυτά ήταν τα πρώτα βήματα που στην συνέχεια οδήγησαν στο διαδίκτυο.



Εικόνα 23. Δίκτυο Υπολογιστών

3.2 Τεχνολογίες και Τοπολογίες

3.2.1 Master/Slave Communications

Το Master / Slave είναι ένα μοντέλο ασύμμετρης επικοινωνίας ή ελέγχου όπου μία συσκευή ή διαδικασία (ο «Master») ελέγχει μία ή περισσότερες άλλες συσκευές ή διαδικασίες (οι «Slave») και χρησιμεύει ως κόμβος επικοινωνίας. Σε ορισμένα συστήματα, ένας Master επιλέγεται από μια ομάδα κατάλληλων συσκευών, με τις άλλες συσκευές να λειτουργούν ως υποτελείς. Ιστορικά, η ορολογία του Master / Slave υπάρχει εδώ και δεκαετίες, αν και τον 21ο αιώνα αποτελεί αντικείμενο διαμάχης λόγω της σχέσης της με τη δουλεία και από τότε ορισμένοι οργανισμοί και προϊόντα το έχουν αντικαταστήσει με εναλλακτικούς όρους [14].

3.2.2 Τοπολογίες

3.2.2.1 Τι είναι τοπολογίες

Η τοπολογία δικτύου είναι η διάταξη των στοιχείων (σύνδεσμοι, κόμβοι, κ.λπ.) ενός δικτύου επικοινωνίας. Η τοπολογία δικτύου μπορεί να χρησιμοποιηθεί για τον καθορισμό ή την περιγραφή της διάταξης διαφόρων τύπων

δικτύων τηλεπικοινωνιών, συμπεριλαμβανομένων των δικτύων ραδιοεπικοινωνιών εντολών και ελέγχου και των δικτύων υπολογιστών [15].

Η τοπολογία δικτύου είναι η τοπολογική δομή ενός δικτύου και μπορεί να απεικονίζεται φυσικά ή λογικά. Είναι μια εφαρμογή της θεωρίας γράφων όπου οι συσκευές επικοινωνίας μοντελοποιούνται ως κόμβοι και οι συνδέσεις μεταξύ των συσκευών μοντελοποιούνται ως συνδέσεις ή γραμμές μεταξύ των κόμβων. Η φυσική τοπολογία είναι η τοποθέτηση των διαφόρων συνιστωσών ενός δικτύου (π.χ. θέση συσκευής και εγκατάσταση καλωδίου), ενώ η λογική τοπολογία απεικονίζει τον τρόπο ροής των δεδομένων μέσα σε ένα δίκτυο. Οι αποστάσεις μεταξύ κόμβων, φυσικών διασυνδέσεων, ποσοστών μετάδοσης ή τύπων σήματος μπορεί να διαφέρουν μεταξύ δύο διαφορετικών δικτύων, ωστόσο οι λογικές τοπολογίες τους μπορεί να είναι ίδιες. Η φυσική τοπολογία ενός δικτύου εξετάζεται από το φυσικό επίπεδο του μοντέλου OSI.

Παραδείγματα τοπολογιών δικτύου συναντώνται στα τοπικά δίκτυα (LAN), την πιο κοινή εγκατάσταση δικτύου υπολογιστών. Κάθε δεδομένος κόμβος στο LAN έχει έναν ή περισσότερους φυσικούς συνδέσμους με άλλες συσκευές στο δίκτυο. Η χαρτογράφηση αυτών των συνδέσμων έχει ως αποτέλεσμα ένα γεωμετρικό σχήμα που μπορεί να χρησιμοποιηθεί για την περιγραφή της φυσικής τοπολογίας του δικτύου. Μια μεγάλη ποικιλία φυσικών τοπολογιών έχει χρησιμοποιηθεί σε LAN, όπως δακτύλιος (Ring) , δίαυλος (Bus), πλέγμα (Mesh) και αστέρας (Star). Αντιστρόφως, η χαρτογράφηση της ροής δεδομένων μεταξύ των στοιχείων καθορίζει τη λογική τοπολογία του δικτύου. Συγκριτικά, τα Δίκτυα Περιοχής Ελεγκτή, κοινά στα οχήματα, είναι πρωτίστως κατανεμημένα δίκτυα συστήματος ελέγχου ενός ή περισσοτέρων ελεγκτών που διασυνδέονται με αισθητήρες και ενεργοποιητές, αναμφισβήτητα, μια φυσική τοπολογία διαύλου.

3.2.2.2 Πλέγμα (Mesh)

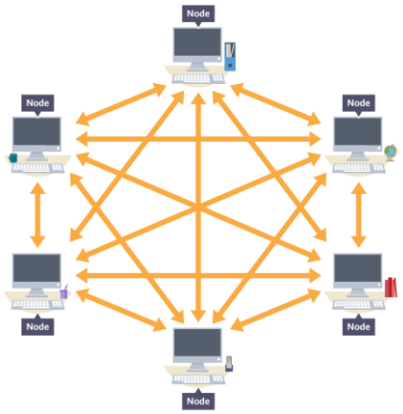
3.2.2.2.1 Γενικές πληροφορίες

Σε τοπολογία πλέγματος δεν υπάρχει κεντρικό σημείο σύνδεσης. Αντ 'αυτού, κάθε κόμβος συνδέεται με τουλάχιστον έναν άλλο κόμβο και συνήθως με περισσότερους από έναν. Κάθε κόμβος μπορεί να στέλνει μηνύματα και να λαμβάνει μηνύματα από άλλους κόμβους. Οι κόμβοι λειτουργούν ως προωθητές, μεταδίδοντας ένα μήνυμα προς τον τελικό προορισμό του. Τα δίκτυα πλέγματος γίνονται όλο και πιο δημοφιλή λόγω της αποτελεσματικότητάς τους.

Υπάρχουν δύο τύποι τοπολογίας πλέγματος:

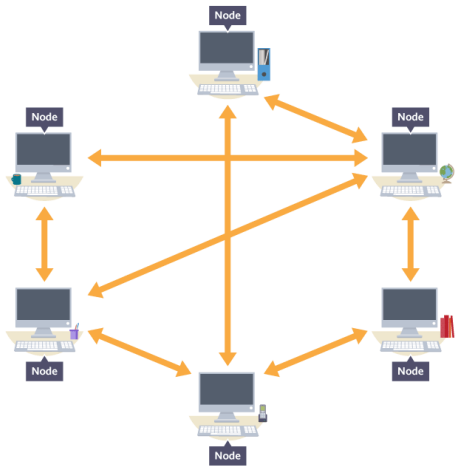
- Πλήρης τοπολογία πλέγματος
- Μερική τοπολογία πλέγματος

Σε ένα πλήρες πλέγμα (Εικόνα 24), κάθε κόμβος συνδέεται άμεσα με κάθε άλλο κόμβο. Αυτό επιτρέπει την αποστολή ενός μηνύματος σε πολλές μεμονωμένες διαδρομές [16].



Εικόνα 24. Πλήρες πλέγμα.

Σε ένα μερικό πλέγμα (Εικόνα 25), δεν συνδέονται όλοι οι κόμβοι απευθείας μεταξύ τους. Επομένως, ένα μερικό πλέγμα έχει λιγότερες διαδρομές για να ταξιδέψει ένα μήνυμα από ένα πλήρες πλέγμα, αλλά είναι πιο δύσκολο να εφαρμοστεί.



Εικόνα 25. Μερικό πλέγμα.

Τα ενσύρματα δίκτυα με τοπολογία πλέγματος τείνουν να είναι ασυνήθιστα, κυρίως επειδή η σύνδεση όλων των κόμβων με όλους τους άλλους κόμβους είναι δαπανηρή και μη πρακτική. Ωστόσο, τα ασύρματα δίκτυα πλέγματος χρησιμοποιούνται όλο και περισσότερο, καθώς είναι πολύ απλούστερο και φθηνότερο να υλοποιηθούν χρησιμοποιώντας ραδιοσήματα.

Πλεονεκτήματα της χρήσης τοπολογίας πλέγματος:

- Τα μηνύματα μπορούν να λαμβάνονται πιο γρήγορα εάν η διαδρομή προς τον παραλήπτη που προορίζεται είναι μικρή.
- Τα μηνύματα πρέπει πάντα να περνούν καθώς έχουν πολλές πιθανές διαδρομές για να ταξιδέψουν.
- πολλαπλές συνδέσεις σημαίνουν (θεωρητικά) ότι κανένας κόμβος δεν πρέπει να απομονωθεί.
- πολλαπλές συνδέσεις σημαίνουν ότι κάθε κόμβος μπορεί να μεταδίδει και να λαμβάνει από περισσότερους από έναν κόμβους ταυτόχρονα.
- νέοι κόμβοι μπορούν να προστεθούν χωρίς διακοπή ή παρέμβαση σε άλλους κόμβους.

Μειονεκτήματα της χρήσης τοπολογίας πλέγματος:

- Τα δίκτυα πλήρους πλέγματος μπορεί να μην είναι πρακτικά εφικτά λόγω του μεγάλου αριθμού συνδέσεων που απαιτούνται.
- πολλές συνδέσεις απαιτούν πολλή συντήρηση.

Αυτή η τοπολογία υποστηρίζεται και από το nRF24 μέσω της βιβλιοθήκης RF24.

3.2.2.2.2 RF24

Η βιβλιοθήκη αυτή είναι σχεδιασμένη να :

- Είναι συμβατή με την καθορισμένη από τον κατασκευαστή λειτουργία του τσιπ, ενώ παράλληλα επιτρέπει στους πιο προχωρημένους χρήστες να εργάζονται εκτός της προτεινόμενης λειτουργίας.
- Χρησιμοποιεί τις δυνατότητες του Radio modem στο μέγιστο των δυνατοτήτων τους μέσω του Arduino.
- Είναι αξιόπιστη, ανταποκρίσιμη, χωρίς σφάλματα και πλούσια σε χαρακτηριστικά.
- Εύκολη στη χρήση από αρχάριους, με καλά τεκμηριωμένα παραδείγματα και χαρακτηριστικά.

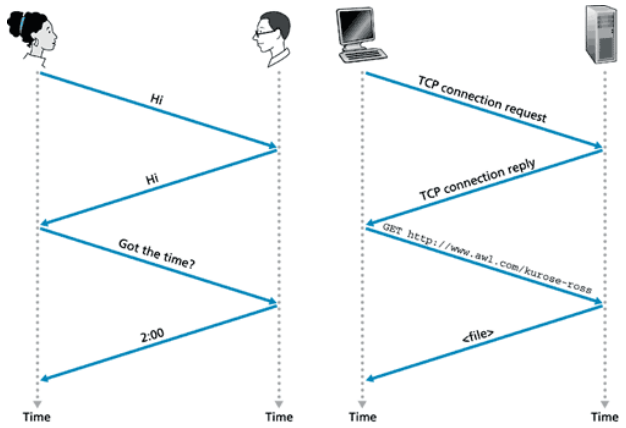
Συγκεκριμένα οι κλάση RF24Mesh είναι σχεδιασμένη να παρέχει ένα απλό και αδιάλειπτο "πλέγμα" για δίκτυα αισθητήρων επιτρέποντας αυτόματη και δυναμική διαμόρφωση που μπορεί να προσαρμοστεί ώστε να ταιριάζει σε πολλά σενάρια. Έχει σχεδιαστεί για άμεση διασύνδεση με τη βιβλιοθήκη RF24Network, ένα OSI Network Layer χρησιμοποιώντας nRF24L01 (+) που οδηγούνται από την RF24. Ο σκοπός είναι να παρέχεται μια απλή διεπαφή χρήστη για τη δημιουργία δυναμικών δικτύων αισθητήρων με τις βιβλιοθήκες RF24 και RF24Network και με την χρήση της να δημιουργούνται σταθερά, πλήρως αυτοματοποιημένα / αυτοδιαχειριζόμενα δίκτυα. Η RF24Mesh παρέχει εκτεταμένες δυνατότητες, όπως αυτόματη διεύθυνση και δυναμική διαμόρφωση ασύρματων αισθητήρων.

Ο τρόπος λειτουργίας της είναι ο εξής:

- Στους κόμβους εκχωρείται ένας μοναδικός αριθμός που κυμαίνεται από 1 έως 253, και η διαχείριση της διεύθυνσης, η δρομολόγηση κ.λπ. γίνεται από τη βιβλιοθήκη.
- Το μοναδικό αναγνωριστικό είναι σαν μια διεύθυνση IP, που χρησιμοποιείται για επικοινωνία σε υψηλό επίπεδο εντός της στοίβας επικοινωνίας RF24 και γενικά παραμένει στατική. Στο επίπεδο δικτύου, οι φυσικές διευθύνσεις, παρόμοιες με τις διευθύνσεις MAC, κατανέμονται καθώς οι κόμβοι κινούνται και δημιουργούν συνδέσεις εντός του δικτύου.
- Ο κόμβος «master» παρακολουθεί τα μοναδικά nodeID και τις διευθύνσεις RF24Network που έχουν εκχωρηθεί. Όταν ένας κόμβος μετακινείται φυσικά, ή απλώς χάνει τη σύνδεσή του με το δίκτυο, μπορεί αυτόματα να ξανασυνδεθεί με το πλέγμα και να αναπροσαρμοστεί εντός του δικτύου.
- Στο πλέγμα διαμόρφωσης οι αισθητήρες / κόμβοι μπορούν να κινούνται φυσικά, μακριά από τον «κύριο κόμβο» χρησιμοποιώντας άλλους κόμβους για να δρομολογήσουν την κίνηση σε εκτεταμένες αποστάσεις. Η τοπολογία αναδιαμορφώνεται καθώς οι συνδέσεις διακόπτονται και αποκαθίστανται σε διαφορετικές περιοχές του δικτύου.
- Το RF24Mesh εκμεταλλεύεται τη λειτουργικότητα και τις λειτουργίες των βιβλιοθηκών RF24 και RF24Network, οπότε οτιδήποτε από την αντιμετώπιση, τη δρομολόγηση, τον κατακερματισμό / επανασυναρμολόγηση αντιμετωπίζεται αυτόματα με διαδικασίες σχεδιασμένες για την υποστήριξη ενός ραδιοδικτύου πολλαπλών κόμβων.

3.2 Τι είναι ένα Πρωτόκολλο

Όλες οι δραστηριότητες στο διαδίκτυο, που εμπλέκουν δύο ή περισσότερες απομακρυσμένες μεταξύ τους οντότητες που επικοινωνούν, διέπονται από ένα πρωτόκολλο. Πρωτόκολλο επικοινωνίας ορίζεται ένα σύνολο κανόνων συμφωνημένων και από τα δυο επικοινωνούντα μέρη και που εξυπηρετούν τη μεταξύ τους ανταλλαγή πληροφοριών (Εικόνα 26). Το πρωτόκολλο επικοινωνίας είναι μια δέσμη κανόνων στους οποίους στηρίζεται η επικοινωνία των συσκευών (συνήθως, αλλά όχι πάντα, υπολογιστών) σε ένα δίκτυο. Οι κανόνες αυτοί καθορίζουν το χρόνο, τη μορφή και τη σειρά μετάδοσης των πληροφοριών στο δίκτυο. Εκτελούν, επίσης, έλεγχο και διόρθωση σφαλμάτων στη διάρκεια μετάδοσης των πληροφοριών. Το διαδίκτυο και τα δίκτυα υπολογιστών γενικά, κάνουν εκτεταμένη χρήση πρωτοκόλλων. Διαφορετικά πρωτόκολλα χρησιμοποιούνται ώστε να γίνονται διαφορετικές εργασίες επικοινωνίας [17].

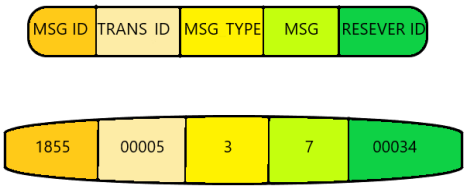


Εικόνα 26. Παράδειγμα πρωτοκόλλου επικοινωνίας μεταξύ ανθρώπων και μεταξύ μηχανών

3.3 Το ιδιωτικό Πρωτόκολλό Μας

Η ανάγκη ανάπτυξης ενός πρωτοκόλλου για τις ανάγκες της πτυχιακής μας εργασίας, έγκειται στο ότι οι έτοιμες λύσεις για mesh που προσφέρονται από τις βιβλιοθήκες του nRF δεν επαρκούν για τις απαιτήσεις της παρούσας εργασίας. Αρχικά η βιβλιοθήκη RFMesh δεν επιτρέπει την αποστολή μηνυμάτων σε μη καθορισμένο ID κόμβου. Στην δική μας περίπτωση είναι πολύ σημαντική η επιλογή άμεσης επικοινωνίας με όλους τους διαθέσιμους κόμβους. Άλλη μια διαφορά της βιβλιοθήκης σε σχέση με το πρωτόκολλο που έχουμε σχεδιάσει, είναι ότι στην πληροφορία που στέλνεται μέσω της συνάρτησης write της βιβλιοθήκης δεν συμπεριλαμβάνεται από που προέρχεται το μήνυμα. Πληροφορία που θα μπορούσε να είναι εξαιρετικά χρήσιμη σε μερικές περιπτώσεις εφαρμογής του συστήματος. Μέσω του RFMesh μπορείς να στείλεις πολλών ειδών πληροφορία σαν μήνυμα και σε διαφορετικά μεγέθη, σε αντίθεση με το πρωτόκολλο μου όπου η κάθε ομάδα πληροφοριών πακετάρεται με συγκεκριμένο τρόπο και έχει πάντα συγκεκριμένο μέγεθος.

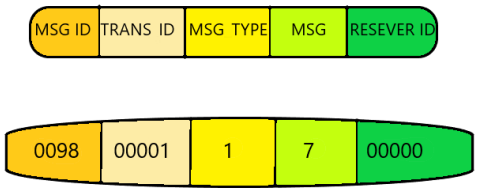
Το πρωτόκολλο που σχεδιάσαμε για να ικανοποιηθούν οι απαιτήσεις της εργασίας είναι αρκετά απλό. Οι απαιτήσεις όμως μπορεί να αυξηθούν κατά την ανάπτυξη και την εξέλιξή του συστήματός μας. Για αυτό υπάρχει η δυνατότητα να είναι επεκτάσιμο ώστε να περιέχει διαφορετικού τύπου πληροφορία κάθε φορά. Αρχικά χωρίζεται σε πέντε κομμάτια, εκ των οποίων το πρώτο είναι ο μοναδικός αριθμός του μηνύματος. Αυτό μας επιτρέπει να ξεχωρίζουμε τα μηνύματα μεταξύ τους καθώς και τη σειρά με την οποία θα πρέπει να εκτελεστούν. Έπειτα εμπεριέχεται ο μοναδικός αριθμός του κόμβου αποστολής. Σε περίπτωση που ο στόλος αποτελείται από παραπάνω από ένα οχήματα μας βοηθάει να καταλάβουμε από ποιο όχημα έρχεται το μήνυμα μιας και η επικοινωνία είναι αμφίδρομη. Το επόμενο κομμάτι του πρωτοκόλλου καθορίζει το είδος του μηνύματος. Όπως είπαμε και παραπάνω υπάρχει η δυνατότητα αναβάθμισης της εργασίας, είτε σε κομμάτι λογισμικού και αλγορίθμων, είτε σε κομμάτι προσθήκης αισθητήρων. Έχοντας παραπάνω από ένα τύπο μηνύματος έχουμε την δυνατότητα να διαχειριζόμαστε για διαφορετικούς κάθε φορά σκοπούς και λόγους, την πληροφορία που δεχόμαστε. Ο τέταρτος συντελεστής αυτού του πρωτοκόλλου είναι η ίδια η πληροφορία. Μέσω αυτής πραγματοποιούνται οι ενέργειες που είναι απαραίτητες για τη λειτουργία του κώδικα και συνεπώς και όλης της εργασίας. Τέλος, το πέμπτο κομμάτι είναι το id του κόμβου που προορίζεται το μήνυμα. Εάν το μήνυμα απευθύνεται στον κόμβο με id “00000” ο οποίος δεν υπάρχει καθώς ξεκινάμε από το 00001, τότε αυτό σημαίνει ότι το μήνυμα απευθύνεται σε όλους του κόμβους (μήνυμα broadcast). Στην Εικόνα 27 βλέπουμε την προδιαγραφή του πρωτοκόλλου μας και ένα παράδειγμα τιμών.



Εικόνα 27. Παράδειγμα τιμών πρωτοκόλλου

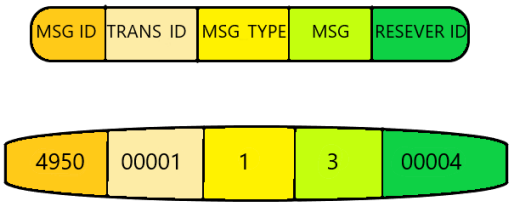
Το Message_ID μπορεί να πάρει τιμές από το 0000 έως το 9999. Το κάθε μήνυμα έχει διαφορετικό id από το προηγούμενο και το επόμενο κατά ένα. Κάθε φορά που φτάνει στο 9999 μηδενίζει και ξεκινάει από την αρχή. Το Transmitter_ID παίρνει σαν τιμή το ID του οχήματος που στέλνει το μήνυμα και μπορεί να πάρει τιμές από το 00001 μέχρι και το 99999. Το Message_Type στην συγκεκριμένη υλοποίηση μπορεί να πάρει την τιμή 1 ή την τιμή 2. Η τιμή 1 χρησιμοποιείται όταν βρισκόμαστε σε λειτουργία τηλεχειρισμού του στόλου οχημάτων από το κινητό μέσω Bluetooth. Η τιμή 2 χρησιμοποιείται αποκλειστικά σε περίπτωση που τα οχήματα έχουν μπει σε λειτουργία αυτόματης αποφυγής εμποδίων (auto pilot). Σε αυτή τη λειτουργία έχουμε προδιαγράψει όλα τα οχήματα να κινούνται ευθεία μέχρι κάποιο από αυτά να αντιληφθεί κάποιο εμπόδιο σε απόσταση 10 εκατοστών. Όταν συμβεί αυτό, στέλνει μήνυμα στα υπόλοιπα οχήματα με Message_Type=2 για να δράσουν ανάλογα. Συγκεκριμένα όλα σταματάνε και κάνουν στροφή 90 μοιρών αριστερά για να αποφύγουν το εμπόδιο. Σε περίπτωση μελλοντικής επέκτασης το Message_Type μπορεί να πάρει και κάποιον άλλον αριθμό για κάποια άλλη λειτουργία. Το πεδίο «μήνυμα (MSG)» εξαρτάται από τον τύπο του. Σε περίπτωση που ο τύπος είναι 2 τότε η μοναδική τιμή MSG που μπορεί να πάρει είναι το 1, αφού μόνο την παραπάνω λειτουργία αποφυγής εμποδίων έχουμε προδιαγράψει. Σε περίπτωση που Message_Type=1 τότε οι τιμές που μπορεί να πάρει είναι από το 1 μέχρι το 8. Κάθε μια από τις τιμές είναι μοναδική και υπεύθυνη για κάτι διαφορετικό. Οι τιμές 1 μέχρι 6 χρησιμοποιούνται για την κίνηση του οχήματος. Το 1 για μπροστά-αριστερά, το 2 για μπροστά, το 3 για μπροστά-δεξιά, το 4 για πίσω-αριστερά, το 5 για πίσω, το 6 για πίσω-δεξιά. Η τιμή 7 είναι για την αδρανή κατάσταση που σταματάει τους κινητήρες. Η αδρανής κατάσταση είναι η default κατάσταση που ισχύει μέχρι να πατηθεί κάποιο άλλο κουμπί από το χειριστή της κινητής εφαρμογής. Τέλος, η τιμή MSG=8 αποστέλλεται από το μάστερ όχημα στα slaves όταν έχει ζητηθεί από τον χειριστή της κινητής εφαρμογής η είσοδος στη λειτουργία auto pilot, προκειμένου να ξεκινήσει αυτή η λειτουργία. Από το σημείο αυτό και μετά, και μέχρι την έξοδο από τη λειτουργία auto pilot, όλα τα μηνύματα που ανταλλάσσονται ανάμεσα στα οχήματα είναι με Message_Type=2 και MSG=1. Τελευταίο πεδίο του πρωτοκόλλου μας είναι το Receiver_ID, που περιέχει το ID του κόμβου προορισμού του μηνύματος. Μπορεί να πάρει σαν όρισμα οποιοδήποτε ID κόμβου. Δηλαδή μπορεί να πάρει τιμές από 00001 μέχρι 99999. Σε περίπτωση όμως που είναι αναγκαία η προώθηση του μηνύματος σε όλους τους κόμβους τότε το Receiver_ID θα πάρει την τιμή 00000 η οποία δεν αντιστοιχεί σε κάποιο ID.

Έστω ότι σε μια στιγμή ηρεμίας του οχήματος επιλέγουμε ένα τυχαίο μήνυμα που αποστέλλεται από το όχημα που είναι συνδεδεμένο με το Bluetooth προς τα υπόλοιπα οχήματα όλου του στόλου. Όπως βλέπουμε στην Εικόνα 28, η τιμή του πρώτου πεδίου είναι απρόβλεπτη επειδή η κατάσταση του οχήματος μπορεί να αλλάξει ανά πάσα στιγμή, καθώς τα μηνύματα παράγονται και στέλνονται με πολύ μικρά χρονικά κενά μεταξύ τους ακόμα και αν το όχημα δεν αλλάζει κατάσταση. Το όχημα που είναι συνδεδεμένο με το Bluetooth στη συγκεκριμένη στιγμή είναι το όχημα με ID 00001. Επειδή δεν είμαστε στην διαδικασία αποφυγής εμποδίων το Message_Type είναι 1. Η τιμή του μηνύματος για την ήσυχη κατάσταση του οχήματος είναι η τιμή 7 και τέλος το Receiver_ID έχει την τιμή 00000 επειδή το μήνυμα αναφέρεται σε όλα τα οχήματα.



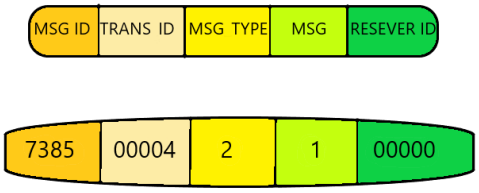
Εικόνα 28. Παράδειγμα μηνύματος σε κατάσταση ηρεμίας

Σε διαφορετική περίπτωση που θέλουμε το όχημα να στείλει το μήνυμα σε ένα μόνο όχημα και θέλουμε να στρίψει δεξιά τότε θα αλλάξουν μερικά πράγματα. Όπως αποτυπώνεται στην Εικόνα 29, πιθανότατα ο αριθμός ID του μηνύματος δε θα είναι ίδιος γιατί αναφερόμαστε σε διαφορετική χρονική στιγμή από το προηγούμενο παράδειγμα. Το Transmitter_ID και το Message_Type είναι το ίδιο αλλά η τιμή που ενημερώνει τον παραλήπτη να στρίψει δεξιά είναι το 3. Το Receiver_ID έχει αλλάξει επειδή απευθυνόμαστε σε συγκεκριμένο όχημα και έχει πάρει την τιμή του ID εκείνου.



Εικόνα 29. Παράδειγμα μηνύματος όπου το όχημα στρίβει δεξιά.

Στην περίπτωση όπου κάποιο όχημα βρίσκεται στη λειτουργία αποφυγής εμποδίων και βρει κάποιο εμπόδιο μπροστά του, στέλνει ένα μήνυμα για να ειδοποιήσει και τα υπόλοιπα οχήματα. Ένα παράδειγμα της παραπάνω υπόθεσης βλέπετε στην Εικόνα 30. Σε αντίθεση με τα προηγούμενα παραδείγματα σε αυτό παρατηρούμε αλλαγές σε όλα τα πεδία του μηνύματος. Ξεκινώντας από την προβλεπόμενη αλλαγή στο ID του μηνύματος συνεχίζουμε στο ID του Transmitter. Αυτό το μήνυμα προέρχεται από ένα όχημα που δεν είναι συνδεδεμένο με Bluetooth με την συσκευή χειρισμού και έχει διαφορετικό ID. Το μήνυμα δεν αφορά την αλλαγή κατάστασης ή κατεύθυνσης οπότε αλλάζει και το Message_Type σε 2. Η τιμή που μπορεί να πάρει το μήνυμα αυτού του τύπου είναι το 1 αρά αλλάζει και αυτό και τελικά το Receiver_ID παίρνει την τιμή 00000 επειδή το μήνυμα απευθύνεται σε όλους τους κόμβους.



Εικόνα 30. Παράδειγμα μηνύματος σε κατάσταση αποστολής ανίχνευσης εμποδίου.

4. Υλοποίηση Συστήματος

4.1 Arduino IDE

Το Arduino IDE είναι μια εφαρμογή για πλατφόρμες Windows, macOS, Linux όπου είναι γραμμένο από συναρτήσεις C και C++. Χρησιμοποιείται για τη σύνταξη και τη μεταφόρτωση προγραμμάτων σε πλακέτες συμβατές με Arduino άλλα και σε άλλες συμβατές πλακέτες τρίτων. Το Arduino IDE υποστηρίζει τις γλώσσες C και C ++ χρησιμοποιώντας ειδικούς κανόνες δομής κώδικα. Επίσης περιέχει βιβλιοθήκη από το Project Wiring και για αυτό τον λόγο υπάρχουν πολλές κοινές διαδικασίες εισόδου και εξόδου.

4.2 Βιβλιοθήκες

4.2.1 Serial Peripheral Interface

Η βιβλιοθήκη Serial Peripheral Interface(SPI) που περιέχει ένα σύγχρονο πρωτόκολλο σειριακών δεδομένων που χρησιμοποιείται από μικροελεγκτές για επικοινωνία με μία ή περισσότερες περιφερειακές συσκευές γρήγορα, σε μικρές αποστάσεις. Μπορεί επίσης να χρησιμοποιηθεί για επικοινωνία μεταξύ δύο μικροελεγκτών. Σε μια σύνδεση SPI υπάρχει πάντα μια κύρια συσκευή. Αυτή η συσκευή συνήθως είναι ένας μικροελεγκτής και ελέγχει τις περιφερειακές συσκευές. Τυπικά υπάρχουν τρεις ίδιες γραμμές για όλες τις συσκευές. Αυτές είναι:

- MISO (Master In Slave Out) η γραμμή Slave για να στέλνει δεδομένα στον Master .
- MOSI (Master Out Slave In) η γραμμή Master για να στέλνει δεδομένα στις συσκευές.
- SCK (Serial Clock) οι παλμοί ρολογιού που συγχρονίζουν τη μετάδοση δεδομένων που δημιουργείται από τον Master.

Και μια γραμμή συγκεκριμένη για κάθε συσκευή:

- SS (Slave Select) Ο ακροδέκτης της κάθε συσκευής με τον οποίο ο Master μπορεί να ενεργοποιήσει ή να απενεργοποιήσει συγκεκριμένες συσκευές. Όταν ο ακροδέκτης είναι low τότε η συσκευή υπακούει τον Master. Όταν είναι high αγνοεί τον Master.

4.2.2 NRF24L01

Είναι ουσιαστικά ένας οδηγός για τη μονάδα nrf24l01 και η βιβλιοθήκη αυτή είναι σχεδιασμένη να:

- Έχει την καλύτερη συμβατότητα με το arduino
- Προσφέρει εύκολη χρήση για αρχάριους
- Μοιάζει με τις ενσωματωμένες βιβλιοθήκες του arduino
- Επικοινωνεί με το Arduino μέσα από τη βιβλιοθήκη SPI

Μερικές από τις πιο χρήσιμες εντολές που προφέρονται είναι οι radio.write και radio.read που χρησιμοποιούνται για την αποστολή και παραλαβή μηνυμάτων μέσω nRF.

4.3 Αρχικοποιήσεις

4.3.1 #define

Με τα #define (Εικόνα 31) ορίζουμε τις σταθερές του προγράμματος. Σε αυτή την περίπτωση ορίζουμε τα pins του Arduino με ονόματα που αντιστοιχούν στις εκάστοτε λειτουργίες.

```
#define SCK  13 //PIN NRF
#define MOSI  11 //PIN NRF
#define MISO  12 //PIN NRF
#define foraL1  9 //pin parametrou 1 gia fora aristeris rodas
#define foraL2  8 //pin parametrou 2 gia fora aristeris rodas
#define foraR3  7 //pin parametrou 3 gia fora deksias rodas
#define foraR4  6 //pin parametrou 4 gia fora deksias rodas
#define CE  5 //PIN NRF
#define CSN  4 //PIN NRF
#define trigPin 15 //pin gia es8itira apostasis
#define echoPin 14 //pin gia es8itira apostasis
#define enc_r_pin 3 // pin de3ioy encoder
#define enc_l_pin 2 // pin aristeroy encoder
```

Εικόνα 31. Screenshot κώδικα define

Συνεχίζουμε με μεταβλητές που πρέπει να φαίνονται σε πολλές διαφορετικές συναρτήσεις και σε interrupts όπως φαίνεται στην Εικόνα 32.

```
RF24 radio(CE, CSN);
const byte addresses[][6] = {"00001,00002"};
long duration;
const String carID = "00001";
char received = '0';
int messengeID=0;
volatile unsigned long enc_l = 0;
volatile unsigned long enc_r = 0;
String history[10];
int pointer=0;
```

Εικόνα 32.. Screenshot κώδικα μεταβλητών.

4.3.2 Void Setup

Άνοιγμα δίαυλου αμφίδρομης επικοινωνίας στην διεύθυνση 00001 και ορισμός εμβέλειας στο nrf24.(Εικόνα 33)

```
radio.begin();
radio.openWritingPipe(addresses[0]); //00001
radio.openReadingPipe(1, addresses[0]); //00001
radio.setPALevel(RF24_PA_MIN); // envelia
```

Εικόνα 33. Screenshot κώδικα nrf24.

Δήλωση του χαρακτηριστικού interrupt στις μεταβλητές που ορίστηκαν για την καταγραφή των encoders των τροχών (Εικόνα 34).

```
attachInterrupt(digitalPinToInterrupt(enc_l_pin), countLeft, CHANGE);
attachInterrupt(digitalPinToInterrupt(enc_r_pin), countRight, CHANGE);
```

Εικόνα 34. Screenshot κώδικα

Η δήλωση του χαρακτήρα των pins φαίνεται στη Εικόνα 35.

```
pinMode(enc_l_pin, INPUT_PULLUP);
pinMode(enc_r_pin, INPUT_PULLUP);
pinMode(foraL1, OUTPUT);
pinMode(foraL2, OUTPUT);
pinMode(foraR3, OUTPUT);
pinMode(foraR4, OUTPUT);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
```

Εικόνα 35. Screenshot κώδικα δήλωσης χαρακτήρα των pins.

4.4 Συναρτήσεις

4.4.1 startMove()

```
void startMove(char received) {
    if (received == '1') {
        turnLeft();
    }
    if (received == '2') {
        moveForward();
    }
    if (received == '3') {
        turnRight();
    }
    if (received == '4') {
        turnRightBack();
    }
    if (received == '5') {
        moveBackwards();
    }
    if (received == '6') {
        turnLeftBack();
    }
    if (received == '7') {
        stopMoving();
        enc_l=0;
        enc_r=0;
    }
    if (received == '8') {
        autoPilot();
    }
}
```

Εικόνα 36. Screenshot κώδικα

Η startMove παίρνει ένα όρισμα τύπου χαρακτήρα και βάσει αυτού καλεί με τη σειρά της την ανάλογη συνάρτηση για να εκκινήσει ή να σταματήσει το όχημα (Εικόνα 36). Οι οκτώ λαμβανόμενες τιμές στέλνονται από την κινητή εφαρμογή μέσω bluetooth στο μάστερ όχημά μας. Το μάστερ όχημα, με τη σειρά του, αναλαμβάνει να στείλει στο slave όχημα τις αντίστοιχες εντολές, μέσω του ιδιωτικού πρωτοκόλλου που έχουμε προδιαγράψει.

4.4.2 moveForward()

```
void moveForward() {
    if(enc_r>enc_l){
        digitalWrite(foraL1, HIGH);
        digitalWrite(foraL2, LOW);
        digitalWrite(foraR3, HIGH);
        digitalWrite(foraR4, HIGH);
    }
    else if(enc_l>enc_r){
        digitalWrite(foraL1, HIGH);
        digitalWrite(foraL2, HIGH);
        digitalWrite(foraR3, LOW);
        digitalWrite(foraR4, HIGH);
    }
    else{
        digitalWrite(foraL1, HIGH);
        digitalWrite(foraL2, LOW);
        digitalWrite(foraR3, LOW);
        digitalWrite(foraR4, HIGH);
    }
}
```

Εικόνα 37. Screenshot κώδικα συνάρτησης moveForward.

Αυτή η συνάρτηση κινεί το όχημα προς τα εμπρός. Όμως επειδή η φύση των dc μοτέρ δεν δίνει ακρίβεια στην ταχύτητα, μετράμε με τα encoders τις στροφές των τροχών του οχήματος και όταν η μια ρόδα ξεπεράσει την άλλη σε στροφές σταματάει μέχρι να την προλάβει η άλλη (Εικόνα 37). Σε περίπτωση που έχουν διανύσει την ίδια απόσταση κινούνται ταυτόχρονα. Αυτές οι μικροδιορθώσεις κάνουν το όχημα να κινείται ευθεία.

4.4.3 stopMoving()

```
void stopMoving() {
    digitalWrite(foraL1, HIGH);
    digitalWrite(foraL2, HIGH);
    digitalWrite(foraR3, HIGH);
    digitalWrite(foraR4, HIGH);
}
```

Εικόνα 38. Screenshot κώδικα συνάρτησης stopMove

Σταματάει την κίνηση του οχήματος. Συνηθώς όταν το κουμπί από την εφαρμογή σηκώνεται ή όταν θέλουμε να σταματήσει το όχημα για να γίνει αποφυγή εμποδίων. (Εικόνα 38)

4.4.4 turnLeft() turnRight() turnLeftBack() turnRightBack() moveBackwards()

```
void turnLeft() {
    digitalWrite(foraR3, LOW);
    digitalWrite(foraR4, HIGH);
    digitalWrite(foraL1, HIGH);
    digitalWrite(foraL2, HIGH);
}

void turnRight() {
    digitalWrite(foraL1, HIGH);
    digitalWrite(foraL2, LOW);
    digitalWrite(foraR3, HIGH);
    digitalWrite(foraR4, HIGH);
}
void turnLeftBack() {
    digitalWrite(foraL1, LOW);
    digitalWrite(foraL2, HIGH);
    digitalWrite(foraR3, HIGH);
    digitalWrite(foraR4, HIGH);
}
void turnRightBack() {
    digitalWrite(foraR3, HIGH);
    digitalWrite(foraR4, LOW);
    digitalWrite(foraL1, HIGH);
    digitalWrite(foraL2, HIGH);
}
void moveBackwards() {
    digitalWrite(foraL1, LOW);
    digitalWrite(foraL2, HIGH);
    digitalWrite(foraR3, HIGH);
    digitalWrite(foraR4, LOW);
}
```

Εικόνα 39. Screenshot κώδικα συναρτήσεων turnLeft() turnRight() turnLeftBack() turnRightBack() moveBackwards()

Σε αυτές τις συναρτήσεις κάνουμε τις υπόλοιπες κινήσεις (Εικόνα 39). Για παράδειγμα με την turnLeft ενεργοποιούμε την δεξιά ρόδα, ώστε να κινηθεί μπροστά και να στρίψει το όχημα αριστερά. Η turnRight κάνει κάτι αντίστοιχο, μόνο που το κάνει με την αριστερή ρόδα για να στρίψει δεξιά. Οι συναρτήσεις με την κατάληξη Back στρίβουν το όχημα αριστερά ή δεξιά με λίγο διαφορετικό τρόπο, χρησιμοποιώντας την όπισθεν. Για παράδειγμα η turnLeftBack στρίβει αριστερά χρησιμοποιώντας την αριστερή ρόδα. Για να το καταφέρει της δίνει οπίσθια φορά με αποτέλεσμα το όχημα να μεταβάλει τον προσανατολισμό του προς τα αριστερά. Το ίδιο κάνει η turnRightBack για να στρίψει το όχημα πίσω δεξιά. Η moveBackwards γυρίζει τους τροχούς προς τα πίσω με αποτέλεσμα το όχημα να κινείται προς τα πίσω.

4.4.5 calcDist()

```
int calcDist() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    return duration * 0.034 / 2;
}
```

Εικόνα 40. Screenshot κώδικα συνάρτησης calcDist

Υπολογίζουμε την απόσταση από ένα εμπόδιο με βάση την ταχύτητα του ήχου και τον χρόνο που έκανε το κύμα να ανιχνευτεί μετά την γέννηση του και το διαιρούμε δια 2 για να βρούμε την απόσταση από το εμπόδιο (Εικόνα 40).

4.4.6 countLeft() countRight()

```
void countLeft() {
    enc_l++;
}

void countRight() {
  enc_r++;
}
```

Κάθε φορά που καλείται, αυξάνει τον μετρητή για τον κάθε encoder (Εικόνα 41).

Εικόνα 41. Screenshot κώδικα συναρτήσεων countLeft() countRight()

4.4.7 changeD()

```
void changeD(){
    enc_r=0;
    enc_l=0;
    while(enc_r<31 and enc_l<31){
        digitalWrite(foraL1, HIGH);
        digitalWrite(foraL2, LOW);
        digitalWrite(foraR3, HIGH);
        digitalWrite(foraR4, LOW);
    }
    stopMoving();

}
```

Αλλάζει κατεύθυνση στο όχημα και το σταματάει (Εικόνα 42).

Εικόνα 42. Screenshot κώδικα συνάρτησης changed.

4.4.8 messageCreation()

```
String messageCreation(String messenge,int messengeType, String ReseverID){
    messengeID++;
    String message;
    if (messengeID==10000)
        messengeID=1;
    else if(messengeID<10)
        message = "000"+String(messengeID) + "_" + String(carID) + "_" + String(messengeType) + "_" + messenge + "_" + ReseverID;
    else if(messengeID<100)
        message = "00"+String(messengeID) + "_" + String(carID) + "_" + String(messengeType) + "_" + messenge + "_" + ReseverID;
    else if(messengeID<1000)
        message = "0"+String(messengeID) + "_" + String(carID) + "_" + String(messengeType) + "_" + messenge + "_" + ReseverID;
    else
        message = String(messengeID) + "_" + String(carID) + "_" + String(messengeType) + "_" + messenge + "_" + ReseverID;
    return message;
  }
```

Εικόνα 43. Screenshot κώδικα συνάρτησης messageCreation().

Παίρνει σαν όρισμα του την πληροφορία που θέλει να μεταφέρει, τον τύπο της και το Id προορισμού. Μετά αυξάνει το messageID κατά 1 ώστε να είναι μοναδικό σε κάθε μήνυμα (Εικόνα 43). Ελέγχει αν είναι μεγαλύτερο από το 10000. Εάν είναι, το κάνει πάλι 1. Θέλοντας να κρατήσουμε το μέγεθος του ID σταθερό, προσθέτουμε μηδενικά ώστε να είναι πάντα 4 οι χαρακτήρες. Όταν ολοκληρωθεί η σύνταξη γυρίζει πίσω ένα string που περιέχει το μήνυμα.

4.4.9 moveDataFromMessage() msgTypeFromMessege() messengeIdFromMessage()

```
}
char moveDataFromMessage(String message){
    int y = message.lastIndexOf('_');
    char moveData = message.charAt(y+1);
    return moveData;
}
String msgTypeFromMessage(String message){
    int y = message.lastIndexOf('_');
    int z = message.lastIndexOf('_',y-1);
    String msgType=message.substring(z+1,y);
    return msgType;
}
String messengerIdFromMessage(String message){
    int y=message.indexOf('_');
    int z=message.indexOf('_',y+1);
    String msngerId=message.substring(y+1,z);
    return msngerId;
}
```

Εικόνα 44. Screenshot κώδικα συναρτήσεων moveDataFromMessage() msgTypeFromMessege() messengeIdFromMessage()

Αυτές οι συναρτήσεις στην Εικόνα 44 επιστρέφουν την πληροφορία του μηνύματος, τον τύπο του μηνύματος και το Id του μηνύματος αντίστοιχα. Λόγω της ιδιαίτερης σύνταξης, γνωρίζουμε πού είναι η πληροφορία που αναζητάμε στην κάθε περίπτωση.

Οι παραπάνω συναντήσεις είναι κοινές και για Master και για Slave. Ακολουθούν οι συναρτήσεις, που αφορούν μόνο το όχημα που ελέγχεται απευθείας από τον χειριστή μέσω Bluetooth.

4.4.10 autoPilot()

AutoPilot ονομάζω την συνάρτηση που κινεί όλα τα οχήματα ευθεία μέχρι κάποιο από αυτά να αντιληφθεί κάποιο εμπόδιο σε απόσταση 10 εκατοστών. Όταν συμβεί αυτό, στέλνει μήνυμα και στα υπόλοιπα οχήματα για να δράσουν ανάλογα. Σταματάνε και κάνουν μια στροφή 90 μοιρών για να αποφύγουν το εμπόδιο.

```
void autoPilot() {
    String i = "1";
    String ReseverID = "00000";
    int stopM='7';
    int distance;
    String message;
    int len;
    int messengeType = 2;
    int j=0;
    String type;
```

Εικόνα 45. Screenshot κώδικα συνάρτησης autoPilot() για αρχικοποιήσεις.

Ξεκινάμε με μερικές αρχικοποιήσεις (Εικόνα 45).

Ακολουθεί ένας επαναληπτικός βρόγχος while, ο οποίος σταματάει όταν κάποιο από τα οχήματα βρει εμπόδιο. Μέσα από αυτόν τον βρόγχο ελέγχουμε αν υπάρχει κάποιο νέο μήνυμα από το Bluetooth για να διακόψει την επανάληψη (Εικόνα 46).

```
while (received != '7') {
    if (Serial.available() > 0) {
        received = Serial.read();//diavazei apo bluetooth
    }
    distance = calcDist();
```

Εικόνα 46. Screenshot κώδικα

Από κάτω βάζουμε στην Εικόνα 47 τη συνάρτησης που μας επιστρέφει την απόσταση που θα χρειαστούμε αργότερα.

```
radio.startListening();
```

Εικόνα 47. Screenshot κώδικα

Αλλάζουμε την κατάσταση του nrf ώστε να γίνει δέκτης. Και στην συνέχεια ελέγχουμε αν υπάρχει διαθέσιμο από κάποιο άλλο όχημα. Εάν υπάρχει, κάνουμε την απαραίτητη μετατροπή του μηνύματος από πίνακα χαρακτήρων σε string. Αντλούμε δεδομένα μέσω των συναρτήσεων, που είπαμε παραπάνω, για τον τύπο και το id του αποστολέα του μηνύματος. Είναι σημαντικό να ξέρουμε το Id του αποστολέα γιατί υπάρχει περίπτωση να πάρουμε πίσω, μέσω της αναμετάδοσης, ένα μήνυμα που παρήγαγε και έστειλε το ίδιο το όχημα. Ο τύπος μας ενδιαφέρει σε περίπτωση που το μήνυμα δεν αφορά τη συγκεκριμένη κατάσταση του οχήματος (Εικόνα 48).

```
radio.startListening();
if (radio.available()){
      len = radio.getDynamicPayloadSize();
      char gotmsg[len]="";
      radio.read( &gotmsg, len );
      message= String(gotmsg);
      radio.stopListening();
      radio.write(&gotmsg, strlen(gotmsg));
      String type=msgTypeFromMessage(message);
      String messengerId=messengerIdFromMessage(message);
      bool found=false;
```

Εικόνα 48. Screenshot κώδικα αναμετάδοση του μηνύματος.

Επίσης κάνουμε και αναμετάδοση του μηνύματος όπως φαίνεται στην Εικόνα 49.

```
for(int h=0;h<10;h++){
        if(message==history[h])
            found=true;
    }
```

Εικόνα 49. Screenshot κώδικα.

Παραπάνω ελέγχουμε αν έχουμε ξαναπάρει το μήνυμα που πήραμε.(εικόνα 50)

```
    }
if(type=="2" && messengerId!="carID" && found!=true){
    if(pointer==10)
          pointer=0;
    history[pointer]=message;
    pointer++;
    i = moveDataFromMessage(message);
  }
```

Εικόνα 50. Screenshot κώδικα.

Εάν το μήνυμα είναι ενός συγκεκριμένου τύπου, δεν έχει εκδοθεί από το ίδιο το όχημα και δεν έχει εκτελεστεί ξανά, μπαίνει στον πίνακα με τα εκτελεσμένα και βάζουμε σε μια μεταβλητή την πληροφορία που περιέχει. Μετά κάνουμε αναπαραγωγή το μήνυμα που μας έστειλαν (Εικόνα 51).

```
if (distance < 10 or i=="0"){
    if (i=="1"){
        String str = messageCreation(String(j),messengeType,ReseverID);
        int str_len =str.length()+1;
        const char messages[str_len];
        str.toCharArray(messages,str_len);
        radio.stopListening();
        radio.write(&messages, strlen(messages));
    }
    i = "0";
    stopMoving();
    received = '7';
    changeD();

}
else {
    i = "1";
    moveForward();
}
```

Εικόνα 51. Screenshot κώδικα για έλεγχο απόστασης ή μηνύματος από άλλο όχημα.

Στην συνθήκη που φαίνεται στην Εικόνα 50 ελέγχουμε αν η απόσταση είναι μικρότερη από το 10 ή αν το μήνυμα είχε την πληροφορία που σηματοδοτεί ότι κάποιο όχημα έχει βρει κάποιο εμπόδιο. Εάν δεν έχουμε πάρει μήνυμα από κάποιο όχημα στέλνουμε εμείς ότι βρήκαμε εμπόδιο. Αλλάζουμε την μεταβλητή για να βγούμε από την επανάληψη, σταματάμε το όχημα και κάνουμε στροφή 90 μοιρών. Αλλιώς συνεχίζει να πηγαίνει μπροστά.

4.4.11 Loop()

Loop ονομάζεται η κύρια συνάρτηση του κώδικα και κάνει αυτό λέει, κάθε φορά που τελειώνει ξεκινάει από την αρχή (Εικόνα 52).

```
void loop() {
    String ReseverID = "00000";
    int messengeType = 1;
    if (Serial.available() > 0) {
        received = Serial.read(); //diavazei apo bluetooth
    }
    String str = messageCreation(String(received),messengeType,ReseverID);
    int str_len =str.length()+1;
    const char messages[str_len];
    str.toCharArray(messages,str_len);

    radio.stopListening();
    radio.write(&messages, strlen(messages));
    startMove(received);
}
```

Εικόνα 52. Screenshot κώδικα κεντρικής συνάρτησης Loop.

Ξεκινάμε ελέγχοντας αν υπάρχει διαθέσιμο μήνυμα από το Bluetooth. Εάν υπάρχει αλλάζει μπαίνει σε μια μεταβλητή και στέλνεται μεσώ nrf, αλλιώς στέλνεται η μη αλλαγμένη τιμή, μετά από τις απαραίτητες μετατροπές του μηνύματος.

Συνεχίζουμε με τις συναρτήσεις που είναι μοναδικές σε οχήματα που δεν ελέγχονται απευθείας από το Bluetooth (Εικόνα 53).

```
void loop() {
    String message;
    int len=0;
    int messengeType = 1;
    while (!radio.available());
    len = radio.getDynamicPayloadSize();
    char gotmsg[len]="";
    radio.read( &gotmsg, len );
    message= String(gotmsg);
    radio.stopListening();
    radio.write(&gotmsg, strlen(gotmsg));
    radio.startListening();
    String type=msgTypeFromMessage(message);
    String messengerId=messengerIdFromMessage(message);
    bool found=false;
    for(int h=0;h<10;h++){
        if(message==history[h])
            found=true;
    }
    if(type=="1" && messengerId!=carID && found!=true){
        if(pointer==10)
            pointer=0;
        history[pointer]=message;
        pointer++;
        received = moveDataFromMessage(message);
        startMove(received);
    }
```

Εικόνα 53. Screenshot κώδικα Loop slave.

4.4.12 Loop() receiver

Συνεχίζοντας σε παρόμοια μοτίβα, αρχικά ελέγχουμε αν υπάρχει διαθέσιμο μήνυμα με τη διαφορά ότι το πρόγραμμα δεν τρέχει συνεχώς αλλά σταματάει στη while μέχρι να υπάρξει διαθέσιμο μήνυμα. Όταν βρεθεί το μήνυμα το αναπαράγει και αποσπάει τα απαραίτητα δεδομένα. Στη συνέχεια ελέγχουμε αν έχουμε ξαναπάρει το μήνυμα, αν είναι ο τύπος μηνύματος που περιμένουμε και αν δεν το έχει πράξει το ίδιο το όχημα. Αν τηρούνται οι προϋποθέσεις βάζουμε το μήνυμα στον πίνακα με τα μυνήματα που έχουν εκτελεστεί και αποσπάμε τα δεδομένα του μηνύματος που μας αφορούν για τη συνέχεια. Με αυτά τα δεδομένα καλούμε τη συνάρτηση που είναι υπεύθυνη για την κίνηση του οχήματος.

4.4.13 autoPilot() receiver

Η μονή διάφορα στο όχημα που δε διαθέτει Bluetooth είναι ότι πρέπει να λαμβάνει και μηνύματα που δεν αφορούν μόνο το autopilot για την περίπτωση της ματαίωσης του (Εικόνα 54).

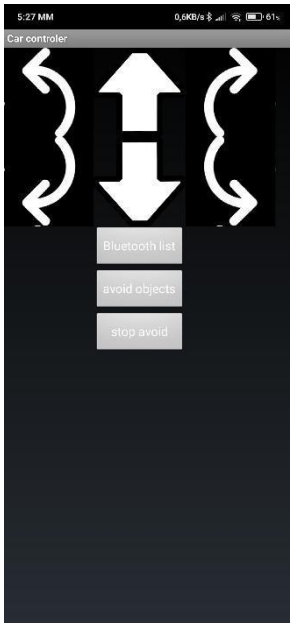
```
else if(type=="1" && messengerId!="carID" && found!=true){
    if(pointer==10)
        pointer=0;
    history[pointer]=message;
    pointer++;
    received = moveDataFromMessage(message);
}
```

Εικόνα 54. Screenshot κώδικα autoPilot() του Slave

5. Λειτουργία και Εφαρμογή Android

5.1 Εφαρμογή Android

Για να χειριστούμε τον στόλο χρησιμοποιούμε μια εφαρμογή που δημιουργήσαμε για συσκευές android με πολύ απλές και συγκεκριμένες λειτουργίες. Όπως βλέπουμε στην Εικόνα 55 η εφαρμογή αποτελείται από εννέα κουμπιά εκ των οποίων τα 6 πάνω είναι τα κουμπιά χειρισμού. Όταν πατάμε το κουμπί με το βελάκι προς τα πάνω και για όσο το πατάμε, οι 2 τροχοί του οχήματος γυρνάνε προς τα μπροστά και το όχημα κινείται ευθεία. Το αντίθετο γίνεται για το βελάκι που δείχνει προς τα κάτω. Όταν πατάμε αυτό το βελάκι και για όσο το πατάμε, το όχημα θα κινείται προς τα πίσω. Για το βελάκι που δείχνει αριστερά και πάνω η κίνηση περιορίζεται στον δεξί τροχό. Ο δεξί τροχός κινείται προς τα εμπρός και το οχιά στριβεί προς τα αριστερά. Το κουμπί με το βελάκι που δείχνει κάτω αριστερά κάνει το ίδιο με την διαφορά ότι ο δεξιός τροχός κινείται προς τα πίσω. Το αντίστοιχο ισχύει και τα κουμπιά που δείχνουν μπροστά-δεξιά και πίσω-δεξιά.



Εικόνα 55. Screenshot κυρίας σελίδας εφαρμογής.

Τα υπολείπα τρία κουμπιά χρησιμοποιούνται για διαφορετικούς σκοπούς το κάθε ένα. Το πρώτο κουμπί που ονομάζεται Bluetooth list σε μεταφέρει σε μια σελίδα (Εικόνα 55) με όλες τις διαθέσιμες συσκευές για να επιλέξεις σε ποια συσκευή θες να συνδεθείς. Σε περίπτωση που παραπάνω από ένα όχημα είναι εξοπλισμένο με Bluetooth module τότε δίνεται η δυνατότητα στον χρήστη να επιλέξει εκείνος το όχημα που θέλει να συνδεθεί. Το όνομα του module είναι HC-05 και το ID του στο συγκεκριμένο παράδειγμα είναι: 00:21:13:00:02:CB. Πριν οποιαδήποτε προσπάθεια επικοινωνίας με τα οχήματα πρέπει να πατήσουμε το κουμπί Bluetooth list και να επιλέξουμε το όχημα που θέλουμε να συνδεθούμε. Μετά τη σύνδεση θα μεταφερθούμε αυτόματα στην αρχική σελίδα της εφαρμογής (Εικόνα 56).

Εικόνα 56. Σελίδα επιλογής και σύνδεσης συσκευής Bluetooth.

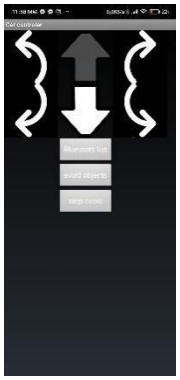
Το δεύτερο και το τρίτο κουμπί είναι κουμπιά λειτουργιών. Το πρώτο ονομάζεται avoid objects και βάζει τα οχήματα σε λειτουργία αυτόματης αποφυγής εμποδίων. Το τρίτο κουμπί που ονομάζεται stop avoid χρησιμοποιείται για να βγάζει τα οχήματα από τη λειτουργία αυτόματης αποφυγής και να τα βάζει σε κατάσταση αδρανείας.

Ο τρόπος λειτουργίας του apk είναι σχετικά απλός. Κάθε φορά που πατιέται ένα κουμπί (Button Push) τότε στέλνεται από το κινητό μέσω Bluetooth στο όχημα ένας κωδικός σε μορφή String. Για παράδειγμα το μπροστά αριστερά στέλνει το “1” το μπροστά στέλνει το “2”, κ.ο.κ. Αυτός ο κωδικός ενημερώνει το όχημα σχετικά με τι κουμπί έχει πατηθεί για να κάνει τις αντίστοιχες ενέργειες Κάθε φορά που αφήνουμε κάποιο κουμπί στέλνεται ο κωδικός “7”. Ο κωδικός του κουμπιού για την αποφυγή εμπόδιων (avoid objects) είναι το “8”. Το stop avoid στέλνει, επίσης, το “7” όταν πατηθεί.

5.2 Λειτουργία

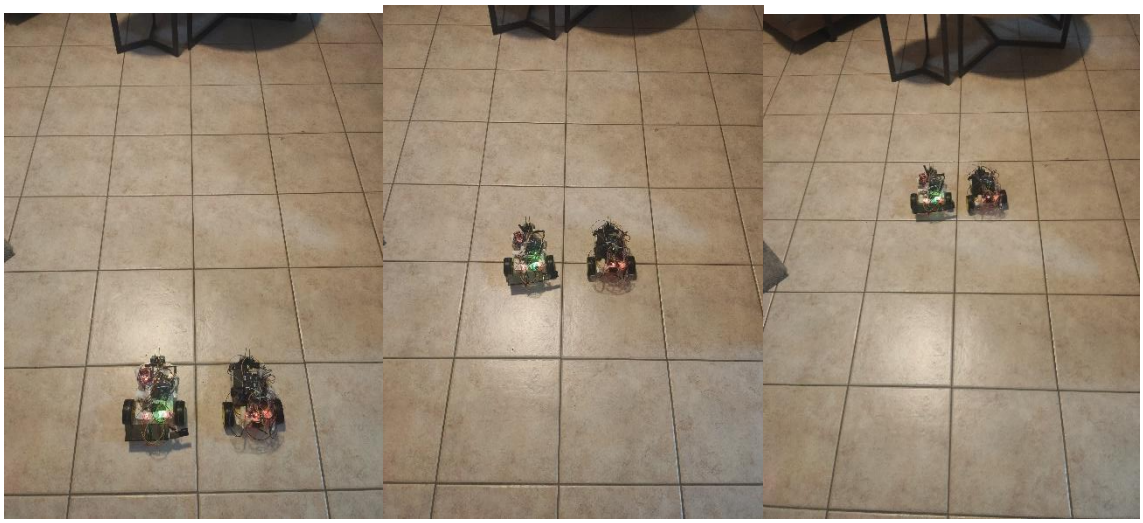
Σε αυτό το σημείο θα δούμε μερικά παραδείγματα μέσω φωτογραφιών για να καταλάβουμε καλύτερα το πως δουλεύουν όλες αυτές οι διαφορετικές τεχνολογίες. Για να μη δείχνουμε σε κάθε παράδειγμα το πως συνδέεται το όχημα μέσω Bluetooth με το κινητό, θεωρούμε ότι η σύνδεση έχει γίνει για τα παρακάτω παραδείγματα.

Αρχικά ας δείξουμε τι θα γίνει αν πατήσουμε παρατεταμένα το κουμπί στην εφαρμογή που έχει ένα βελάκι και δείχνει προς τα πάνω (Εικόνα 57).



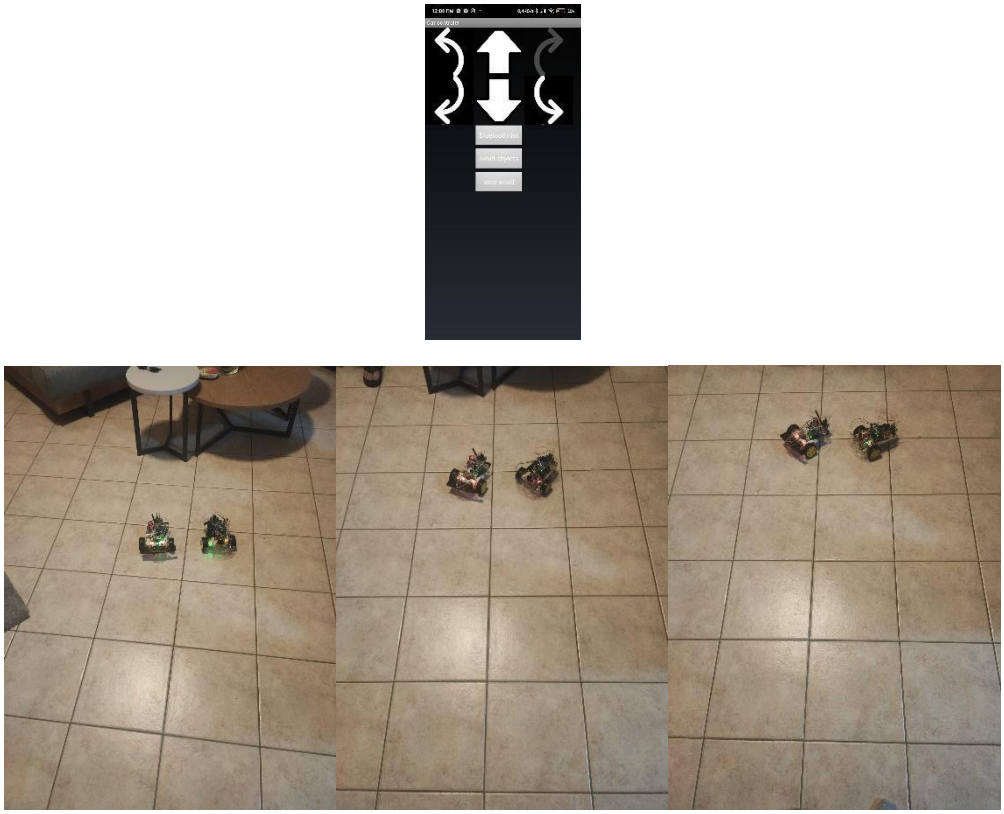
Εικόνα 57. Screenshot εφαρμογής την ώρα που το πλήκτρο «μπροστά» πατιέται.

Ακολουθεί η παρατεταμένη πορεία τον οχημάτων για όσο το πλήκτρο είναι πατημένο. Η κατεύθυνση τον οχημάτων θα είναι προς τα εμπρός.(Εικόνα 58)



Εικόνα 58. Οπτικοποιημένο παράδειγμα πορείας προς τα εμπρός.

Αφήνοντας το κουμπί τα οχήματα σταματάνε. Μετά πατάμε το πάνω-δεξιά και το όχημα στρίβει προς τα δεξιά γυρνώντας την αριστερή του ρόδα προς τα μπροστά (Εικόνα 59).

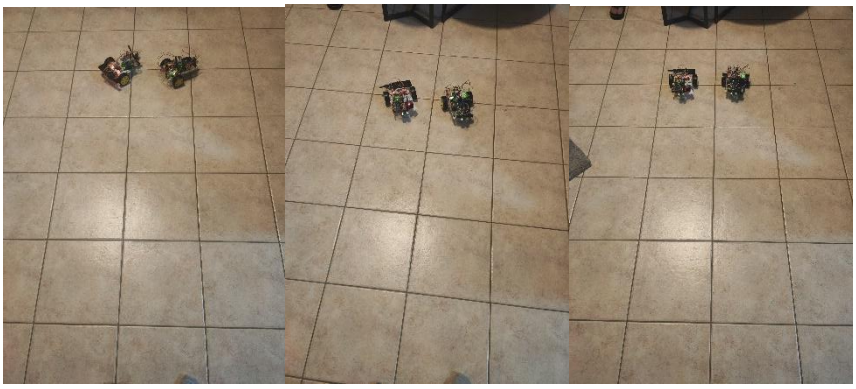


Εικόνα 59. Screenshot εφαρμογής και Οπτικοποιημένο παράδειγμα δεξιάς στροφής.

Για να αποκτήσουμε μια καλή εικόνα για όλους τους βασικούς χειρισμούς αρκεί να δείξουμε το πίσω-αριστερά (Εικόνα 60)

Εικόνα 60. Screenshot εφαρμογής πατώντας το κουμπί «πίσω-αριστερά»

Πατώντας παρατεταμένα το κουπί αυτό, γυρίζει ο δεξιός τροχός προς τα πίσω με αποτέλεσμα το όχημα να προσανατολίζει το μπροστινό του μέρος προς τα δεξιά. Όπως παρατηρούμε στην εικόνα 61 η αριστερή ρόδα κάθε οχήματος δεν έχει κουνηθεί από την θέση της καθώς το όχημα στρίβει.

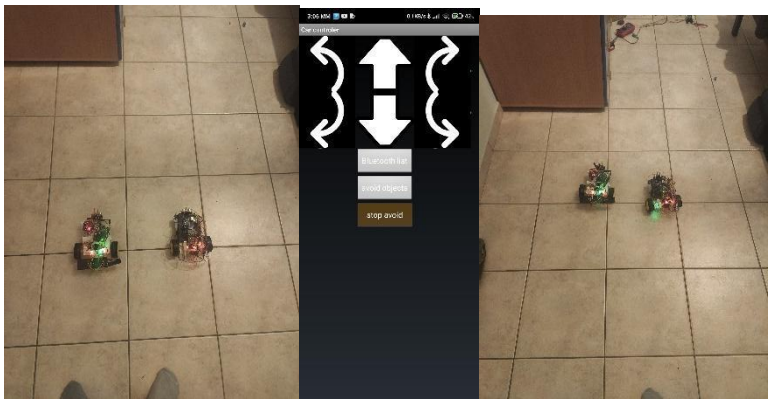


Εικόνα 61. Οπτικοποιημένο παράδειγμα οπισθογωνίας.

Το επόμενο σενάριο αφορά τον την λειτουργία αποφυγής εμποδίων και μπορούμε να δούμε τη συμπεριφορά τον οχημάτων όταν βρουν ένα εμπόδιο μπροστά τους. Πατώντας το κουμπί όπως την εικόνα 62 ξεκινάμε τη διαδικασία.

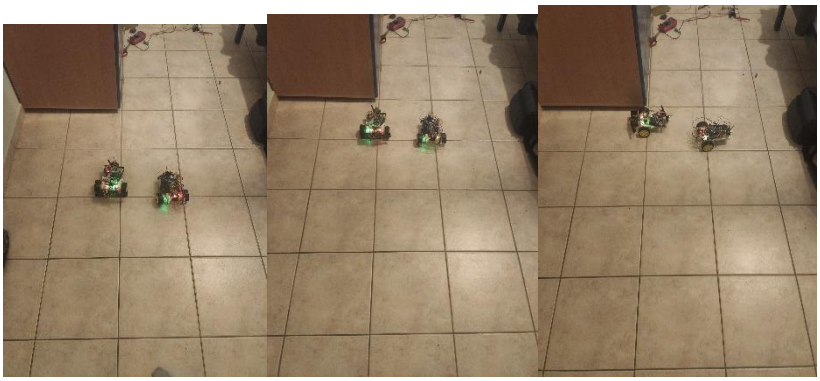
Εικόνα 62. Screenshot εφαρμογής πατώντας το κουμπί «Avoid objects»

Το κουμπί για αυτή την λειτουργία δεν χρειάζεται να είναι παρατεταμένα πατημένο. Το πατάμε μια φορά και αν χρειαστεί πατάμε το κουμπί Stop Avoid για να διακόψουμε τη λειτουργία αποφυγής. Στις παρακάτω εικόνες βλέπουμε τη λειτουργία και τη διακοπή της από το Stop avoid (Εικόνα 63).



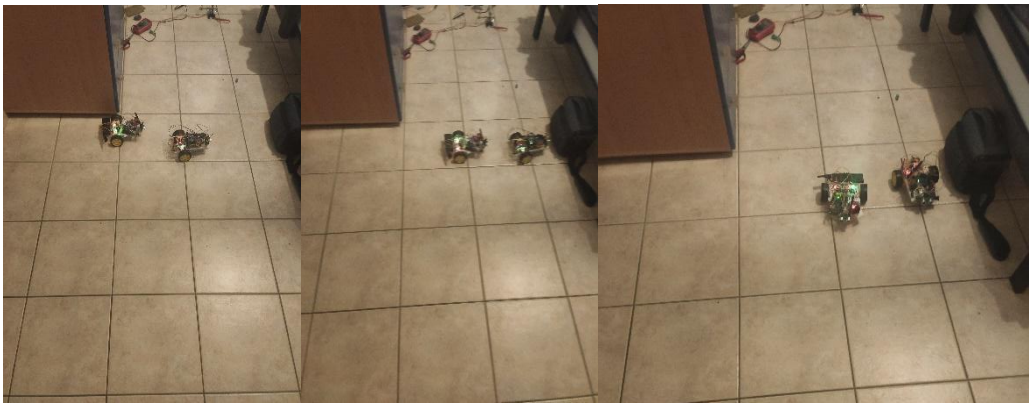
Εικόνα 63. Οπτικοποιημένο παράδειγμα διακοπής λειτουργίας αποφυγής εμποδίων.

Πατώντας ξανά το κουμπί avoid objects ξεκινάμε ξανά τη λειτουργία αποφυγής. Η πορεία των οχημάτων φαίνεται στην Εικόνα 64.



Εικόνα 64. Οπτικοποιημένο παράδειγμα λειτουργίας αποφυγής εμποδίων.

Στην παραπάνω εικόνα βλέπουμε ότι το όχημα αριστερά βρήκε μπροστά του ένα εμπόδιο οπότε έδωσε εντολή και έστριψαν τα οχήματα για να το αποφύγουν. Στην συνέχεια προχώρισαν ευθεία όπως φαίνεται στην εικόνα 65.



Εικόνα 65. Οπτικοποιημένο παράδειγμα λειτουργίας αποφυγής εμποδίων.

Όπως φαίνεται στη εικόνα το δεξί όχημα αυτή την φορά βρίσκει κάποιο εμπόδιο οπότε αυτό δίνει την εντολή για να στρίψουν και να το αποφύγουν.

6 Συμπεράσματα και Μελλοντικές Επεκτάσεις

6.1 Συμπεράσματα

Σε αυτή την πτυχιακή εργασία τέθηκε υπό εξέταση μια τεχνολογική πρόκληση με πολλές διαφορετικές πτυχές. Η πρόκληση αφορά τη δημιουργία ενός συστήματος πολλαπλών οχημάτων που ενεργούν σε λειτουργία σμήνους. Τη μεταξύ τους επικοινωνία αναλαμβάνει ένα ad hoc mesh δίκτυο και την ανταλλαγή δεδομένων αναλαμβάνει ένα πρωτόκολλο που κατασκευάστηκε ειδικά για αυτόν τον λόγο. Επομένως πρόκειται για μια εργασία που περιλαμβάνει τα παρακάτω:

- Ανάπτυξη πρωτοκόλλου M2M
- Ανάπτυξη δικτύου
- Ανάπτυξη κώδικα λειτουργίας
- Ανάπτυξη εφαρμογής χειρισμού
- Ανάπτυξη αλγορίθμου αποφυγής εμποδίων
- Σχεδιασμό και συνδυασμό υλικού
- Ανάπτυξη οχήματος

Η ανάπτυξη του πρωτοκόλλου έγινε με βάση τις τρέχουσες ανάγκες της εργασίας. Το πρωτόκολλο επικοινωνίας μεταξύ μηχανών είναι πολύ σημαντικό και αναγκαίο για τη μεταφορά πληροφορίας από μηχανή σε μηχανή . Ένα πρωτόκολλο μας επιτρέπει να ελέγχουμε την πορεία ενός μηνύματος, τη μοναδικότητά του και από που προέρχεται. Κατά την εφαρμογή του παρατηρήθηκε σημαντική βελτίωση στην ταχύτητα και στην ακρίβεια μετάδοσης της πληροφορίας.

Το δίκτυο αναπτύχθηκε χρησιμοποιώντας το nrf24 και η λειτουργία του είναι αρκετά απλή. Ουσιαστικά τη διασύνδεση την αναλαμβάνει η βιβλιοθήκη που μας επιτρέπει να χρησιμοποιήσουμε τις συναρτήσεις της.

Η λειτουργία του οχήματός βασίζεται σε κώδικα που γράφτηκε στο arduino IDE και είναι υπεύθυνος για τις βασικές λειτουργίες του οχήματος. Μερικές από τις λειτουργίες του οχήματος είναι το να πηγαίνει εμπρός, πίσω, δεξιά, αριστερά. Σε αυτή τη λειτουργία υπάρχει επίσης η μεταφορά των δεδομένων κίνησης του οχήματος μέσω πρωτοκόλλου από το δίκτυο για να εκτελεστούν και από άλλα μέλη του σμήνους.

Οι εντολές για τον χειρισμό του σμήνους δίνονται από μια εφαρμογή που δημιουργήθηκε για android συσκευές και μέσω σύνδεσης bluetooth μεταφέρει τις εντολές στο σμήνος.

Ο αλγόριθμος αποφυγής εμποδίων αποδείχτηκε ένα από τα δυσκολότερα και πιο περίπλοκα κομμάτια της εργασίας. Σημαντικό ρόλο έπαιξε η αναγκαιότητα του συστήματος να ελέγχει για εμπόδια ενώ παράλληλα να είναι σε επικοινωνία για τυχόν εμπόδια σε κάποιο μέλος του σμήνους. Σε αυτή τη λειτουργία του σμήνους όλα τα συστήματα είναι σε λειτουργία και πρέπει να λειτουργούν αρμονικά και παράλληλα.

Το πρώτο βήμα για την κατασκευή του οχήματος είναι η ανάλυση των αναγκών του για την συγκεκριμένη υλοποίηση. Έπειτα ακολουθεί η συλλογή των κομματιών και η σύνδεση τους. Η επιλογή των συγκεκριμένων εξαρτημάτων δεν είναι απόλυτη. Ο κύριος λόγος που χρησιμοποιούνται σε αυτή την εργασία είναι για λόγους προσβασιμότητας. Ενδέχεται να χρειαστούν αλλαγές στον κώδικα έναν χρησιμοποιηθούν άλλου είδους εξαρτήματα.

Όλα αυτά μας δίνουν μια καλή βάση για αρκετές μελλοντικές χρησιμότητες. Για παράδειγμα οι συγχρονισμένες κινήσεις των οχημάτων αποτελούν αρκετά καλή λύση για μεταφορά μεγάλων και βαρέων αντικειμένων με πολλαπλά μικρά οχήματα. Ο αριθμός τον οχημάτων θα μεταβάλλεται ανάλογα τον όγκο και το βάρος τωνν αντικείμενων που μεταφέρονται. Για παράδειγμα σε μια οικοδομή θα μπορούσαν να εφαρμοστούν για τη

μεταφορά αντικείμενων στον τόπο χρήσης τους, διαδικασία που κανονικά γίνεται από ανθρώπους και σε πολλές περιπτώσεις οδηγεί σε προβλήματα υγείας.

6.2 Μελλοντικές Επεκτάσεις

Ο τομέας της ρομποτικής σμήνους και γενικότερα των πολλαπλών ρομπότ ανήκει σε μια κατηγορία που η δημιουργικότητα και η φαντασία μπορούν να φέρουν στην επιφάνεια δημιουργίες που αξιοποιούν στο έπακρο τις δυνατότητες τους. Για αυτό πιστεύεται ότι η εφαρμογή αυτών τον τεχνολογιών έχει τη δυνατότητα και την προοπτική να εξελιχθεί και να χρησιμοποιηθεί για πολλά και διαφορετικά πράγματα [17-22].

Αρχικά θα μπορούσε να εμπλουτιστεί το σύστημα αισθητήρων απόστασης προσφέροντας δεδομένα και από τις 4 πλευρές του οχήματος για την ακριβέστερη αναγνώριση του περιβάλλοντα χώρου. Σε συνδυασμό με την ανάπτυξη ενός νέου αλγορίθμου αποφυγής εμποδίων, η αποτελεσματικότητα, η ταχύτητα και η ακρίβεια θα αυξηθούν.

Η χρήση πιο σκληρών υλικών και μεγαλύτερων μοτέρ θα μπορούσαν να επιτρέψουν στο σμήνος να δραστηριοποιείται κάτω από πιο αντίξοες συνθήκες.

Οι ρόδες είναι μια ανακάλυψη που αδιαμφισβήτητα έχει αλλάξει τον κόσμο. Δεν είναι όμως κατάλληλη για κάθε περίπτωση. Τα μηχανικά πόδια είναι πολύ πιο περίπλοκα στη λειτουργία και στην κατασκευή τους. Προσφέρουν όμως πρόσβαση σε σημεία που δεν μπορούμε να φτάσουμε με άλλα μέσα. Μερικά από αυτά τα σημεία θα μπορούσαν να είναι σπηλιές ή συντρίμμια. Τα ρομπότ θα μπορούν να περνούν από μικρά περάσματα αποτελεσματικά και με τους καταλλήλους αισθητήρες να εξερευνούν σημεία που δεν μπορούν να φτάσουν άνθρωποι.

Ο ρόλος της τεχνολογίας είναι να δίνει λύσεις σε προβλήματα των ανθρώπων και να κάνει την καθημερινότητά τους πιο εύκολη. Είναι δικός μας ρόλος να αναγνωρίζουμε αυτά τα προβλήματα και τις καθημερινές προκλήσεις, καθώς επίσης να εφευρίσκουμε νέα τεχνολογικά μέσα ή να εξελίσσουμε τα υπάρχοντα.

ΠΗΓΕΣ

[1] Y. Tan, Handbook of Research on Design, Control, and Modeling of Swarm Robotics, Peking University, China: Igi Global, December, 2015.

[2] S. M. Avinash Gautam, A Review of Research in Multi-Robot Systems, Pilani, India, August 2012.

[3] Akhileshkhot, "Swarm Robotics," 10 May 2020. [Online]. Available: https://medium.com/@akhileshkhot9007/swarm-robotics-d3fe103fdc50.

[4] s. jacob, What is Machine to Machine Communication (M2M)?, available from https://medium.com, Jan 24, 2018.

[5] K. —. A. N. Reader, An era of IoT — Machine-to-Machine Communication (M2M), medium.com, Sep 9, 2018.

a. T. ,. M. A. E.-d. Ahmed Salah Rozik, Design and Implementation of the Sense Egypt Platform for Real-Time Analysis of IoT Data Streams, Mansoura, Egypt: Scientific Research Publishing, September 2016.

[6] K. —. A. N. Reader, "An era of IoT — M2M communication protocols — MQTT," Jul 12, 2019.

[7] Wikipedia, "Ad hoc δίκτυο," 7 Φεβρουαρίου 2020.. [Online]. Available: https://el.wikipedia.org/wiki/Ad_hoc_δίκτυο.

[8] LastMinuteEngineers, "How nRF24L01+ Wireless Module Works & Interface with Arduino," [Online]. Available: https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/.

[9] B. SCHWEBER, "Rotary encoder basics and applications, Part 1: Optical encoders," analogictips, FEBRUARY 13, 2018.

[10] "HC-05 Bluetooth Module," [Online]. Available: https://www.gme.cz/data/attachments/dsh.772-148.1.pdf.

[11] randomnerdtutorials.com, "Complete Guide for Ultrasonic Sensor HC-SR04 with Arduino," [Online]. Available: https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/.

[12] "Τι είναι ένα δίκτυο," [Online]. Available: https://sites.google.com/site/efaliagka/diktio.

[13] "Master/slave (technology)," 26 March 2021. [Online]. Available: https://en.wikipedia.org/wiki/Master/slave_(technology).

[14] V. Beal, "What Are Network Topologies? Webopedia Study Guide," Wedopedia, 31 Οκτωβρίου 2017.

[15] Bitesize, "Network topologies, protocols and layers," [Online]. Available: https://www.bbc.co.uk/bitesize/guides/zr3yb82/revision/2.

[16] Wikipedia, "Πρωτόκολλο επικοινωνίας," 26 Σεπτεμβρίου 2019. [Online]. Available: https://el.wikipedia.org/wiki/Πρωτόκολλο_επικοινωνίας.

[17] Y. Nikoloudakis ; S. Panagiotakis ; E. Markakis ; G. Mastorakis ; C. X. Mavromoustakis ; E. Pallis, “Towards a FOG-enabled navigation system with advanced cross-layer management features and IoT equipment”, chapter contribution in the “Cloud and Fog Computing in 5G Mobile Networks: Emerging advances and applications”, Editors: Evangelos Markakis ; George Mastorakis ; Constandinos X. Mavromoustakis ; Evangelos Pallis, IET, March 2017, pp. 171 –191.

[18] Prokopis Vavouranakis, Spyros Panagiotakis, George Mastorakis, Constandinos X. Mavromoustakis and Jordi Mongay Batalla, “Recognizing Driving Behaviour Using Smartphones”, chapter contribution in the “Beyond the Internet of Things, Everything Interconnected”, Editors: Jordi Mongay Batalla, George Mastorakis, Constandinos X. Mavromoustakis, Evangelos Pallis, Springer-Verlag (2017), pp. 269-299.

[19] Prokopis Vavouranakis, Spyros Panagiotakis, George Mastorakis, Constandinos X. Mavromoustakis, “Smartphone-Based Telematics for Usage Based Insurance”, chapter contribution in the “Advances in Mobile Cloud Computing and Big Data in the 5G Era”, Editors: Mavromoustakis Constandinos X., Mastorakis George, Dobre Ciprian, Springer-Verlag (2017), pp. 309-339.

[20] Koralia Papadokostaki, George Mastorakis, Spyros Panagiotakis, “Handling Big Data in the era of IoT”, chapter contribution in the “Advances in Mobile Cloud Computing and Big Data in the 5G Era”, Editors: Mavromoustakis Constandinos X., Mastorakis George, Dobre Ciprian, Springer-Verlag (2017), pp. 3-22.

[21] Georgios Skourletopoulos, Constandinos X. Mavromoustakis, George Mastorakis, Jordi Mongay Batalla, Ciprian Dobre, Spyros Panagiotakis, Evangelos Pallis, “Big Data and Cloud Computing: A Survey of the State-of-the-

Art and Research Challenges", chapter contribution in the "Advances in Mobile Cloud Computing and Big Data in the 5G Era", Editors: Mavromoustakis Constandinos X., Mastorakis George, Dobre Ciprian, Springer-Verlag (2017), pp. 23-41.

[22] Despina E. Athanasaki, Spyros Panagiotakis, Nikos Pinikas, Zacharias G. Sifakis, Andreas Vlisidis, "Real-time obstacle avoidance navigation strategy in unknown environments", in proceedings of the 15th International Symposium on Ambient Intelligence and Embedded Systems (AMIES 2016), 22 - 24 September, 2016, Heraklion, Crete, Greece.