



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΤΜΗΜΑ
ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Απομακρυσμένη Επικοινωνία Οχήματος Μέσω BLUETOOTH & WiFi –
Ορισμός Τοποθεσίας Με Τη Χρήση Δεδομένων GPS Και Δημιουργία
Εφαρμογής Χειρισμού.

Στυλιανός Κασκάνης AM 7147

Ευάγγελος Παππάς AM 6931

Επιβλέπων Καθηγητής: Δημήτριος Καρέλης

Πάτρα 2024

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή
Πάτρα, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Καρέλης Δημήτριος επίκουρος καθηγητής, Υπογραφή
2. Σταθάτος Ηλίας καθηγητής, Υπογραφή
3. Χαραλαμπίκος Βασίλειος καθηγητής, Υπογραφή

Υπεύθυνη Δήλωση Φοιτητή

Βεβαιώνω ότι είμαι συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τη συγκεκριμένη εργασία.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Πελοποννήσου δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία των __φοιτητών__ Κασκάνη Στυλιανού και του Παππά Ευάγγελου _ που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης ο συγγραφέας/δημιουργός εκχωρεί στο Πανεπιστήμιο Πελοποννήσου, μη αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού. Ο συγγραφέας/δημιουργός διατηρεί το σύνολο των ηθικών και περιουσιακών του δικαιωμάτων.

Πρόλογος

Τα τελευταία έτη η ανάπτυξη της τεχνολογίας των ηλεκτρολογικών υπολογιστών και των μικροελεγκτών έχει αυξηθεί σε μεγάλο βαθμό. Αυτό έχει σαν αποτέλεσμα την διευκόλυνση της καθημερινότητας μας σε πολλούς τομείς, απλοποιώντας την χρήση τους, διευκολύνοντας την εκμάθησή τους και μειώνοντας σημαντικά το κόστος.

Συνεπώς η απόκτηση ενός μικροελεγκτή αποτελεί μία διαδικασία που είναι προσιτή προς το ευρύ κοινό. Αυτό συμβάλει στην εύκολη πραγματοποίηση μικροεφαρμογών ή εφαρμογών μεγαλύτερου εύρους και προσελκύει το ενδιαφέρον πολλών ανθρώπων να δημιουργήσουν και να αναπτύξουν νέες γνώσεις στον προγραμματισμό και στους μικροελεγκτές.

Οι μικροελεγκτές είναι ένας τύπος επεξεργαστή βασισμένος στους μικροεπεξεργαστές ο οποίος λειτουργεί με ελάχιστα εξωτερικά εξαρτήματα εξαιτίας των πολλών ενσωματωμένων υποσυστημάτων που διαθέτει. Ακόμα χρησιμοποιείται ευρύτατα σε όλα τα ενσωματωμένα συστήματα ελέγχου χαμηλού και μεσαίου κόστους, όπως αυτά που χρησιμοποιούνται σε αυτοματισμούς, ηλεκτρονικά καταναλωτικά προϊόντα.

Τέλος θα θέλαμε να ευχαριστήσουμε τον καθηγητή και εισηγητή μας Δημήτριο Καρέλη, που συνέβαλε σημαντικά στην εκπόνηση της εργασίας μας, την συνεργασία μας και αποτέλεσε σημαντικό παράγοντα για την ολοκλήρωση της.

Περίληψη

➤ ΕΛΛΗΝΙΚΑ

Με αφορμή την διπλωματική εργασία που έπρεπε να εκπονήσουμε για την ολοκλήρωση του προγράμματος σπουδών του Πανεπιστημίου Πελοποννήσου του τμήματος Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών. Αποφασίσαμε και σε συνεργασία με τον επιβλέποντα Επίκουρο καθηγητή Δημήτριου Καρέλη να ασχοληθούμε και να φέρουμε εις πέρας την δημιουργία και κατασκευή ενός οχήματος με απομακρυσμένο τρόπο χειρισμού.

Με την υποστήριξη και τον συντονισμό από τον κ. Δημήτριο Καρέλη αυτό το έργο πραγματοποιήθηκε και πολύ πιο αναβαθμισμένο από την αρχική ιδέα. Στη συνέχεια δίνεται μια συνοπτική περιγραφή του έργου που υλοποιήσαμε.

Αρχικά όσο αφορά το κατασκευαστικό κομμάτι αγοράσαμε ένα όχημα το οποίο του αφαιρέσαμε από όλους τους τρόπους χειρισμού που είχε μόνο του από τον κατασκευαστή του , το βάψαμε εξωτερικά το ‘κουβούκλιο’ του όπως επίσης και τους τροχούς του. Στη συνέχεια πρέπει να αναφερθούμε στους τρόπους χειρισμού και στις πλακέτες που χρησιμοποιήσαμε για αυτό το σκοπό καθώς και την συνδεσμολογία τους και τις επιμέρους κατασκευές που υλοποιήσαμε για να μπορέσουν αν τοποθετηθούν αυτές οι πλακέτες πάνω στο σασί του οχήματος μας. Οι πλακέτες αυτές είναι δύο μια για το χειρισμό του ρομποτ μας με Bluetooth σύζευξη η οποία είναι η ευρέως γνώστη πλακέτα Arduino με τον Bluetooth δέκτη Bluetooth HC-05 με BT 2.0 και η δεύτερη που είναι υπεύθυνη για το χειρισμό του ρομποτ μας με τη χρήση WiFi είναι η NodeMCU V3. Επιπρόσθετους δέκτες που χρησιμοποιούμε για τον καλύτερο χειρισμό και την όσο τον δυνατόν λιγότερα ατυχήματα είναι ο αισθητήρα υπερήχων HC-SR04 που το χρησιμοποιούμε για την αποφυγή τρακαρίσματος του οχήματος μας. Τελευταία πλακέτα και ίσως η πιο σημαντική είναι η πλακέτα χειρισμού που μας δίνει την κίνηση στους τροχούς του project μας και είναι η L298N Dual Motor Driver Module.

Τέλος η πλακέτα που χρησιμοποιούμε είναι η μονάδα SIM808 GPS/GSM/GPRS με την οποία προσπαθούμε να κάνουμε το tracking του οχήματος μας και να έχουμε το στίγμα κάθε φορά που βρίσκεται.

Καταλήγοντας πρέπει να επισημανθεί ότι η βοήθεια του κ. Δημήτριου Καρέλη ήταν πολύ σημαντική όσο αφορά την υλοποίηση του project στο τεχνικό κομμάτι αλλά και όσο αφορά την εμπύχωση σε δυσκολίες που αντιμετωπίσαμε κατά τη διάρκεια της υλοποίησης του . Στη συνέχεια

θα δούμε πιο αναλυτικά στα επιμέρους κεφάλαια όλα τα παραπάνω καθώς και πολύ πιο επεξηγηματικά τον τρόπο δημιουργίας του project.

➤ ENGLISH

On the occasion of the diploma thesis that we had to prepare for the completion of the study program of the University of Peloponnese in the Department of Electrical Engineering & Computer Engineering. We also decided, in collaboration with the supervising Assistant Professor Dimitrios Karelis, to deal with and carry out the creation and construction of a vehicle with remote control.

With the support and coordination of Mr. Dimitrios Karelis, this project was realized and much more upgraded than the original idea. Below is a brief description of the project we implemented.

First, as far as the construction part is concerned, we bought a vehicle which we stripped of all the handling methods that it had by its manufacturer, we painted the exterior of its 'hood' as well as its wheels. Then we must refer to the handling methods and the boards we used for this purpose as well as their wiring and the individual constructions we implemented to enable these boards to be placed on the chassis of our vehicle. These boards are two one for controlling our robot with Bluetooth pairing which is the well-known Arduino board with Bluetooth receiver Bluetooth HC-05 with BT 2.0 and the second one which is responsible for controlling our robot using WiFi it is NodeMCU V3. Additional receivers that we use for better handling and as few accidents as possible are the HC-SR04 ultrasonic sensor that we use to avoid crashing our vehicle. Last board and perhaps the most important is the control board that gives us the movement to the wheels of our project and is the L298N Dual Motor Driver Module.

Finally, the board we use is the SIM808 GPS/GSM/GPRS unit with which we try to track our vehicle and have a spot every time it is located.

In conclusion, it must be pointed out that the help of Mr. Dimitrios Karelis was very important in terms of the implementation of the project in the technical part but also in terms of animating the difficulties we faced during its implementation. Then we will look more analytically in the individual chapters all the above as well as much more explanatory how to create the project.

Περιεχόμενα

Πρόλογος	3
Περίληψη	4
Κεφάλαιο 1: Εισαγωγή	8
1.1 Ιστορικά.....	8
1.2 Έννοια Του Arduino.....	8
1.3 Πλακέτες Που Χρησιμοποιήσαμε	9
1.4 Μικροελεγκτής Arduino Uno	13
1.5 Μικροελεγκτής Arduino Mega 2560	17
1.6 Μικροελεγκτής NodeMCU ESP8266	20
1.7 Bluetooth Module.....	22
1.8 Λειτουργία αισθητήρα υπερήχων HC-SR04	24
1.8.1 Σύνδεση στο Arduino και προγραμματισμός.	25
1.9 L298N Dual Motor Driver Module	26
1.10 Buzzer.....	27
1.11 Led Λάμπες	28
1.12 Μονάδα sim808.....	28
.....	30
Κεφάλαιο 2: Εργαλεία ανάπτυξης λογισμικού	32
2.1 Fritzing	32
2.2 Notepad++	33
2.4 Arduino IDE.....	33
Κεφάλαιο 3: Δημιουργία Εφαρμογών	35
3.1 Το MIT App Inventor.....	35
3.2 Εφαρμογή για χειρισμό του οχήματος μέσω Bluetooth	36
3.2.1 Ανάλυση Block προγραμματισμού του Bluetooth Application.....	39
3.3 Εφαρμογή για το χειρισμό του οχήματος μέσω WiFi	42
3.3.1 Ανάλυση Block Διαγράμματος WiFi εφαρμογής.....	43
Κεφάλαιο 4: Σχηματικά, ανάλυση και συνδεσμολογίες	46
4.1 Σχηματικό συνδεσμολογίας Bluetooth	46
4.2 Σχηματικό συνδεσμολογίας WiFi.....	47
4.3 Συνδεσμολογία GSM – GPS MODULE SIM 808	49
Κεφάλαιο 5: Ανάλυση του Κώδικα	50
5.1 Προγραμματισμός για χειρισμό μέσω Bluetooth	50
5.2 Προγραμματισμός για χειρισμό μέσω WiFi.....	59

5.3 Προγραμματισμός για το GSM – GPS module	67
Κεφάλαιο 6: Συμπεράσματα και εξέλιξη της εργασίας μελλοντικά.....	73
6.1 Συμπεράσματα.....	73
6.2 Εξέλιξη της εργασίας μελλοντικά	73
Κατάλογος φωτογραφικού υλικού	74
Εικόνα 1.0.....	74
Εικόνα 1.1.....	75
Εικόνα 1.2.....	76
Εικόνα 1.3.....	77
Εικόνα 1.4.....	78
Εικόνα 1.5.....	79
Εικόνα 1.6.....	80
Εικόνα 1.7.....	81
Εικόνα 1.8.....	82
Εικόνα 1.10.....	83
Εικόνα 1.11.....	84
ΒΙΒΛΙΟΓΡΑΦΙΑ	85
Διαδίκτυο.....	85
Συγκεντρωτικά ο κώδικας των μικροελεγκτών :.....	86

Κεφάλαιο 1: Εισαγωγή

Στο πρώτο κεφάλαιο θα αναλυθούν όλα τα απαραίτητα στοιχεία που χρειάζεται να γνωρίζει κανείς ώστε να τεθούν σε λειτουργία οι μικροελεγκτές, οι επεκτάσεις, οι μνήμες καθώς και ο τρόπος προγραμματισμού και η συνδεσμολογία τροφοδοσίας. Επιπροσθέτως, θα αναλυθούν τα περιφερειακά στοιχεία που θα χρησιμοποιηθούν για την υλοποίηση του τηλεκατευθυνόμενου.

1.1 Ιστορικά

Το 2005 ένα σχέδιο ξεκίνησε προκειμένου να φτιαχτεί μία συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν χαμηλότερου κόστους από άλλα πρωτότυπα συστήματα τα οποία ήταν διαθέσιμα εκείνη την περίοδο. Οι ιδρυτές Massimo Banzi και David Cueartielles εμπνεύστηκαν το όνομα του σχεδίου από τον Αρντουίνο της Ιβρέας και ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ιβρέα, κομμόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας - την ίδια περιοχή στην οποία στεγαζόταν η εταιρία υπολογιστών Olivetti.

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

1.2 Έννοια Του Arduino

Το Arduino είναι μια πλακέτα μικροελεγκτή ανοικτού κώδικα, που διαθέτει ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους. Μπορεί να προγραμματιστεί με τη γλώσσα Wiring, η οποία βασίζεται στη γλώσσα προγραμματισμού C++ και περιλαμβάνει διάφορες βιβλιοθήκες γραμμένες επίσης σε C++. Το Arduino είναι κατάλληλο για την ανάπτυξη αυτόνομων διαδραστικών συσκευών και μπορεί να συνδεθεί με υπολογιστή μέσω λογισμικών όπως το Processing, το Max/MSP, το Pure Data και το SuperCollider. Οι περισσότερες εκδόσεις του Arduino διατίθενται προ-

συναρμολογημένες, ενώ τα διαγράμματα και οι πληροφορίες για το υλικό είναι διαθέσιμα δωρεάν για όσους θέλουν να συναρμολογήσουν την πλακέτα μόνοι τους..

1.3 Πλακέτες Που Χρησιμοποιήσαμε



Arduino Uno

Μικροελεγκτής: ATmega328P

Τάση Λειτουργίας : 5V

Τάση Εισόδου(συνιστάται) : 7-12 V

Τάση Εισόδου(Όριο) : 6-20 V

Ψηφιακές ακίδες εισόδου / εξόδου : 14

(με 6 PWM)

Αναλογικοί ακροδέκτες εισόδου : 6

Τάση DC ανά είσοδο I/O : 20mA

DC ρεύμα για 3.3V ακίδων: 50mA

Μνήμη Flash : 32KB

SRAM: 2KB

EEPROM: 1KB

Ταχύτητα Ρολογιού : 16MHz

LED_BUILTIN: 13

Μήκος : 68,6mm

Πλάτος : 53,4mm

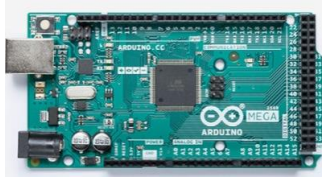
Βάρος : 25g

Μικροελεγκτής: ATmega2560

Τάση Λειτουργίας : 5V

Τάση Εισόδου(συνιστάται) : 7-12 V

Τάση Εισόδου(Όριο) : 6-20 V



Arduino Mega

Ψηφιακές ακίδες εισόδου / εξόδου : 54

(με 15 PWM)

Αναλογικοί ακροδέκτες εισόδου : 16

Τάση DC ανά είσοδο I/O : 20mA

DC ρεύμα για 3.3V ακίδων: 50mA

Μνήμη Flash : 256KB

SRAM: 8KB

EEPROM: 4KB

Ταχύτητα Ρολογιού : 16MHz

LED_BUILTIN: 13

Μήκος : 101.52mm

Πλάτος : 53,3mm

Βάρος : 37g

Υλικό: Πειραματική πλακέτα +

Μικτή αλουμινένιο **χρώμα:**

Μονάδα

BlackChip: CP2102



NodeMCU ESP8266

Θερμοκρασία εργασίας: -40 °C ~ +125 °C

Ισχύς: 4.5V ~ 9V (10VMAX),
USB poweredTrive

Τύπου: ≈70mA (200mA MAX)

Αναμονή: <200uASize: (L) X
(W) 4.8X2.5cm / 1.89"X0.98 "
(appr.)

Χαρακτηριστικά: Ένα μεγάλο σύνολο εργαλείων για την ανάπτυξη ESP8266.ESP8266 έχει Pin IO .ESP8266 πλήρης στόμα IO οδηγεί απευθείας λήψη χωρίς resetting.Open-source, διαδραστικό, προγραμματιζόμενο, χαμηλού κόστους, απλό, έξυπνο, WI-FI enabled.Greatly επιταχύνει τη διαδικασία ανάπτυξης IOT εφαρμογή σας.

Συνδεσιμότητα Arduino, συμβατή με UNO, Leonardo, NUCLEO, XNUCLEO



WAVESHARE SIM808

Ενσωματωμένος μετατροπέας USB TO UART CP2102 για σφάλμα UART

5 x LED για την ένδειξη της κατάστασης λειτουργίας της μονάδας

Ο ενσωματωμένος μετατροπέας επιπέδου τάσης, υποστηρίζει και τα συστήματα 3.3V και 5V

Υποδοχή κάρτας SIM για κάρτα SIM 1.8V / 3V

Ανίχνευση αυτόματης ανίχνευσης (1200bps ~ 115200bps)

Bluetooth 3.0, υποστηρίζει τη μεταφορά δεδομένων μέσω Bluetooth

RTC με διασύνδεση τροφοδοσίας ρεύματος

Το firmware μπορεί να αναβαθμιστεί μέσω USB

Έλεγχος μέσω εντολών AT (3GPP TS 27.007,27.005 και ενισχυμένες εντολές ATC της SIMCOM)

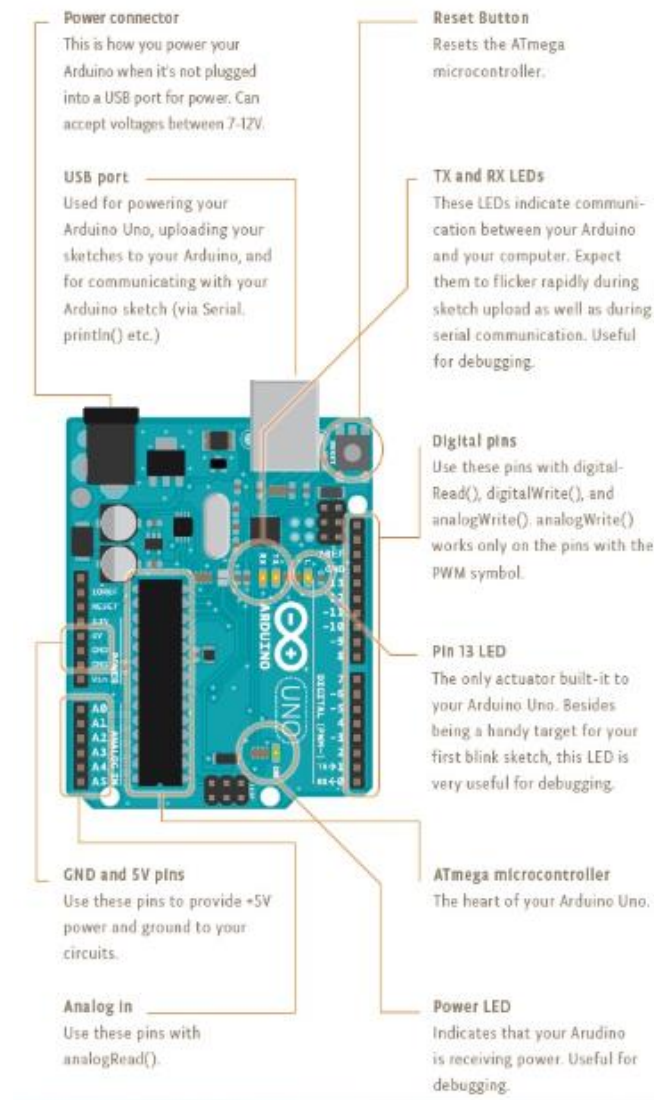
Υποστηρίζει το πακέτο εργαλείων SIM: GSM 11.14 Release 99

Τάση λειτουργίας: 6 ~ 12V

Θερμοκρασία λειτουργίας: -40 ° C ~ +85 ° C

Θερμοκρασία αποθήκευσης: -45 ° C ~ +90 ° C

1.4 Μικροελεγκτής Arduino Uno



Το Arduino Uno αποτελεί την βασική πλακέτα της τεχνολογίας Arduino και προτείνεται για να ξεκινήσει κανείς την εκμάθηση με τα ηλεκτρονικά και τον προγραμματισμό. Στο σημείο αυτό αξίζει να αναφερθεί ότι το μοντέλο Uno της οικογένειας Arduino είναι το πιο διαδεδομένο δεδομένου ότι είναι συμβατό με πλήθος αισθητήρων και επεκτάσεων.

Η έκδοση 3 (Rev3) είναι η τελευταία που κυκλοφορεί και είναι πλήρως συμβατή τόσο με καινούργιες όσο και με παλιές πλακέτες επέκτασης.

Το Arduino Uno Rev3 βασίζεται στον μικροελεγκτή ATmega328 της Atmel. Είναι μια ολοκληρωμένη πλακέτα που περιέχει ό,τι χρειάζεται για να μπορεί να προγραμματιστεί και να λειτουργήσει με μοναδική προϋπόθεση να συνδεθεί με ένα απλό καλώδιο USB στον υπολογιστή είτε με ένα τροφοδοτικό στην πρίζα. Επιπροσθέτως, υπάρχει δυνατότητα να λειτουργήσει με απλή μπαταρία. Αναλυτικά, η πλακέτα διαθέτει 14 ψηφιακές εισόδους ή εξόδους (6 από αυτές μπορεί να

χρησιμοποιηθούν σαν PWM έξοδοι), 6 αναλογικές εισόδους, 1 θύρα USB (τύπου B) για τον προγραμματισμό και την τροφοδοσία της πλακέτας, 1 είσοδο τροφοδοσίας που μπορεί να χρησιμοποιηθεί εναλλακτικά για τροφοδοσία από τροφοδοτικό πρίζας ή από απλή μπαταρία, 1 υποδοχή ICSP και τέλος το κουμπί για την επανεκκίνηση (reset) της πλακέτας. Ο μικροελεγκτής είναι συγχρονισμένος στους 16 μεγακύκλους (Crystal 16MHz). Η συγκεκριμένη έκδοση του Arduino Uno περιλαμβάνει αποσπώμενο μικροελεγκτή (DIP Version) και δίνει τη δυνατότητα στο χρήστη να πειραματιστεί όσες φορές θελήσει. Σε περίπτωση που καεί ο ελεγκτής, μπορεί να αντικατασταθεί με έναν καινούριο με μικρό κόστος.

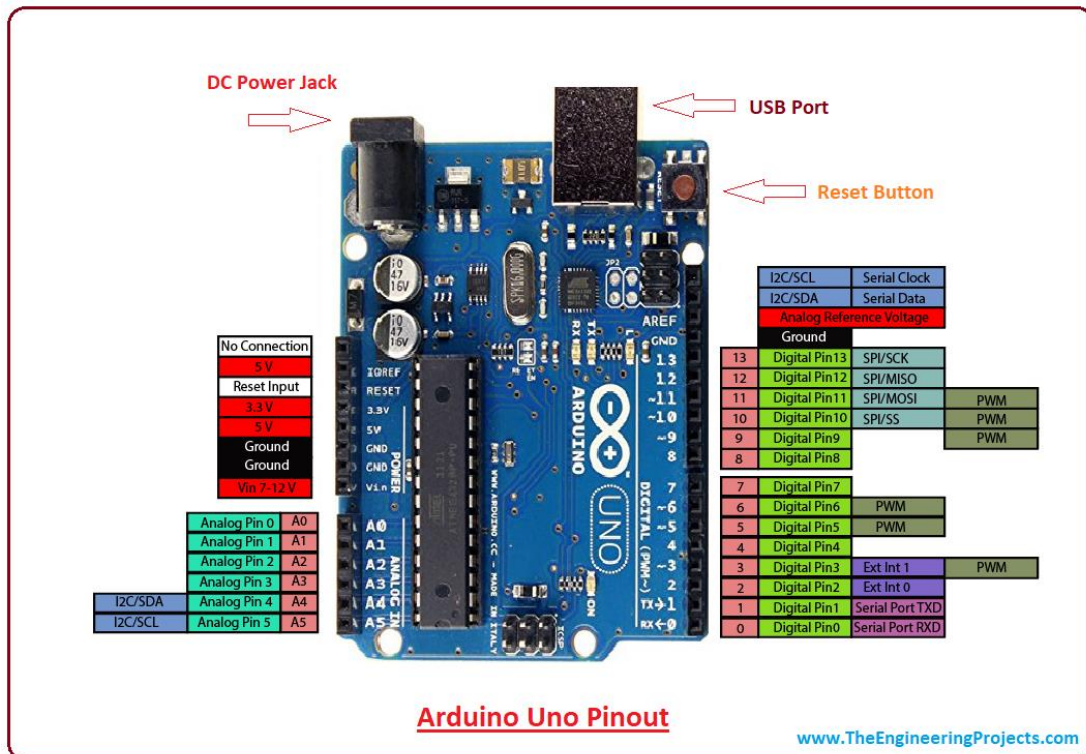
Η μνήμη Flash του Arduino Uno στην οποία μπορεί να αποθηκευτεί το πρόγραμμα σας (sketch) έχει χωρητικότητα 32KB, ικανή να δεχτεί τα περισσότερα απλά προγράμματα. Στην περίπτωση που η συγκεκριμένη χωρητικότητα δεν είναι επαρκής, μπορείτε σαφώς να επιλέξετε μια μεγαλύτερη πλακέτα Arduino όπως η Arduino Mega 2560.

Αξίζει να σημειωθεί μιας και συγκαταλαμβάνεται στα προτερήματά του ότι το Arduino Uno λειτουργεί με τροφοδοσία 5V DC από την είσοδο του USB ή με 9V/12V DC από την είσοδο της τροφοδοσίας χωρίς να υπάρχει κίνδυνος ηλεκτροπληξίας.

Το UNO στα Ιταλικά σημαίνει ENA και το όνομα της πλακέτας προέκυψε χάρη στο γεγονός ότι ήταν η πρώτη της οικογένειας Arduino με δυνατότητα απευθείας επικοινωνίας μέσω θύρας USB σε συνεργασία με το πρόγραμμα Arduino 1.0 (IDE 1.0).

Προκειμένου να ξεκινήσει κανείς με τον προγραμματισμό του Arduino Uno, μπορεί να επισκεφθεί αυτή την σελίδα που περιέχει αναλυτικές πληροφορίες.

Παρακάτω απεικονίζεται η μορφή της πλακέτας. Ωστόσο, θα προχωρήσουμε και σε μία ανάλυση κάθε εισόδου-εξόδου που θα συναντήσει κανείς έχοντας στα χέρια του μια τέτοια πλακέτα.



1. **VIN:** Η τάση εισόδου στην πλακέτα Arduino όταν χρησιμοποιείται εξωτερική πηγή ενέργειας μπορεί να τροφοδοτηθεί είτε μέσω αυτού του pin είτε να χρησιμοποιηθεί για πρόσβαση στην τροφοδοσία που είναι συνδεδεμένη στη θύρα 2.1mm.
2. **5V:** Παρέχει σταθεροποιημένη έξοδο 5V.
3. **3V3:** Παρέχει τάση 3,3V που παράγεται από τον σταθεροποιητή, με μέγιστη κατανάλωση ρεύματος έως 50 mA.
4. **GND:** Pins γείωσης.
5. **IOREF:** Αυτό το pin παρέχει την τάση αναφοράς που χρησιμοποιεί ο μικροελεγκτής. Ένα σωστά ρυθμισμένο shield μπορεί να ανιχνεύσει την τάση στο pin IOREF και να επιλέξει την κατάλληλη πηγή ενέργειας ή να ενεργοποιήσει μετατροπείς τάσης στις εξόδους, επιτρέποντας τη λειτουργία με 5V ή 3.3V.

Ο μικροελεγκτής Arduino uno είναι εξοπλισμένος με τρεις ξεχωριστές τράπεζες μνήμης που παίζουν καθοριστικό ρόλο στη συνολική λειτουργικότητά του. Πρώτον, υπάρχει η μνήμη flash, η οποία χρησιμεύει ως χώρος αποθήκευσης για σκίτσα Arduino. Δεύτερον, έχουμε τη SRAM, γνωστή και ως στατική μνήμη τυχαίας πρόσβασης, η οποία είναι υπεύθυνη για τη δημιουργία και τη χρήση μεταβλητών κατά την εκτέλεση του σκίτσου. Τέλος, υπάρχει η EEPROM, μια εξειδικευμένη μνήμη που χρησιμοποιείται από προγραμματιστές για την αποθήκευση μακροπρόθεσμων πληροφοριών.

Αναλυτικότερα:

1. **2KB μνήμης SRAM:** Τα προγράμματα μπορούν να χρησιμοποιήσουν μια πολύτιμη μορφή μνήμης για να αποθηκεύσουν μεταβλητές, πίνακες και άλλες απαραίτητες πληροφορίες. Ωστόσο, αυτή η μνήμη δεν είναι μόνιμη και θα χάσει τα δεδομένα της εάν διακοπεί η παροχή ρεύματος στο Arduino ή εάν πατηθεί το κουμπί επαναφοράς.
2. **1KB μνήμης EEPROM:** Η δυνατότητα ανάγνωσης και εγγραφής δεδομένων από προγράμματα είναι βασικό χαρακτηριστικό αυτής της τεχνολογίας. Σε αντίθεση με τη SRAM, διατηρεί το περιεχόμενό της ακόμη και σε περίπτωση απώλειας ρεύματος ή επανεκκίνησης.
3. **32KB μνήμης Flash:** Τα 2 KB χρησιμοποιούνται για τον bootloader, προκειμένου ο μικροελεγκτής να είναι σε θέση να προγραμματίζεται μέσω USB. Τα υπόλοιπα 30KB της μνήμης Flash χρησιμοποιούνται για την αποθήκευση των προγραμμάτων του χρήστη, αφού πρώτα μεταγλωττιστούν στον υπολογιστή. Η μνήμη Flash, δε χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή επανεκκίνησης. Τέλος, αναφορικά με τα pins εισόδου και εξόδου, χωρίζονται σε αναλογικά και ψηφιακά. Ακολουθεί η εικόνα με την μορφή και την επεξήγησή τους.

Τα pins από 0 έως 13 μπορούν να χρησιμοποιηθούν ως ψηφιακές έξοδοι. Το pin AREF χρησιμοποιείται για να παρέχεται τάση αναφοράς στα αναλογικά pins που θα αναλυθούν στη συνέχεια. Εκτός από ψηφιακές έξοδοι, ορισμένα pins επιτελούν και άλλες εργασίες. Αναλυτικότερα:

1. **Pins 0 και 1:** λειτουργούν ως RX και TX της σειριακής θύρας όταν το πρόγραμμά ενεργοποιεί τη σειριακή θύρα. Έτσι, όταν το πρόγραμμά στέλνει δεδομένα στη σειριακή θύρα, αυτά προωθούνται και στη θύρα USB μέσω του ελεγκτή USB, αλλά και στον ακροδέκτη 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή. Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμά ενεργοποιήσει το σειριακό interface, χάνει 2 ψηφιακές εισόδους/εξόδους η πλατφόρμα.

2. **Pins 2 και 3:** Λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Ρυθμίζονται μέσα από το πρόγραμμα, ώστε να λειτουργούν αποκλειστικά ως ψηφιακές εισοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει άμεσα και εκτελείται μια συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.

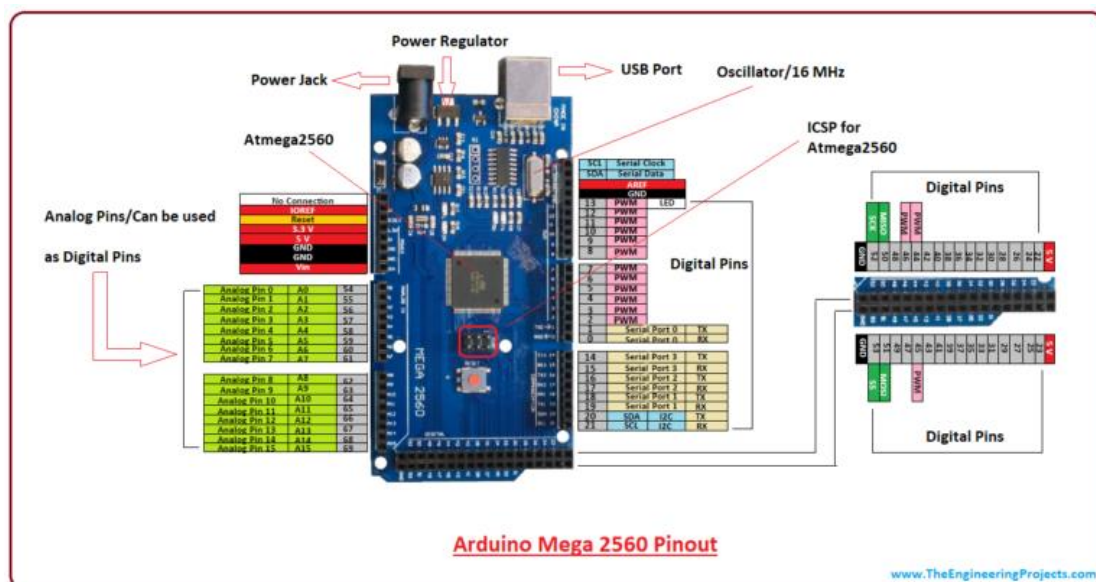
Pins 3, 5, 6, 9, 10 και 11: μπορούν να λειτουργήσουν και ως ψευδό - αναλογικές έξοδοι χρησιμοποιώντας PWM (Pulse Width Modulation).

1.5 Μικροελεγκτής Arduino Mega 2560

Το **Arduino Mega 2560** είναι ένας πίνακας Microcontroller που βασίζεται στο Atmega2560 με την ιδιαιτερότητα ότι διαθέτει περισσότερη χωρητικότητα ως προς τη μνήμη και I / O καρφίτσες σε σύγκριση με άλλες κάρτες που διατίθενται στην αγορά. Υπάρχουν 54 ψηφιακές ακίδες εισόδου / εξόδου και 16 αναλογικές ακίδες ενσωματωμένες στον πίνακα που κάνουν τη συσκευή αυτή μοναδική και ξεχωρίζουν από τους άλλους. Από τις 54 ψηφιακές εισόδους / εξόδους, οι 15 χρησιμοποιούνται για PWM (διαμόρφωση πλάτους παλμών). Ένας κλασματικός ταλαντωτής συχνότητας 16MHz προστίθεται στον πίνακα, ο οποίος διαθέτει θύρα καλωδίου USB που χρησιμοποιείται για τη σύνδεση και τη μεταφορά κώδικα από τον υπολογιστή στον πίνακα. Το βύσμα τροφοδοσίας συνεχούς ρεύματος συνδέεται με την πλακέτα που χρησιμοποιείται για την τροφοδοσία του πίνακα. Αξίζει να σημειωθεί ότι σε κάποια έκδοση του πίνακα Arduino δεν υπάρχει αυτό το χαρακτηριστικό, όπως επίσης ότι το Arduino Pro Mini δεν περιλαμβάνει βύσμα ισχύος DC. Η κεφαλίδα ICSP είναι μια αξιόλογη προσθήκη στο Arduino Mega που χρησιμοποιείται για τον προγραμματισμό του Arduino και τη μεταφόρτωση του κώδικα από τον υπολογιστή. Το Arduino Mega είναι ειδικά σχεδιασμένο για τα έργα που απαιτούν σύνθετα κυκλώματα και περισσότερο χώρο μνήμης. Τα περισσότερα από τα ηλεκτρονικά έργα μπορούν να γίνουν πολύ καλά από άλλες επιτροπές που είναι διαθέσιμες στην αγορά και οι οποίες καθιστούν το Arduino Mega ασυνήθιστο για τα τακτικά έργα. Ωστόσο, υπάρχουν μερικά έργα που πραγματοποιούνται αποκλειστικά από το Arduino Mega, όπως η κατασκευή 3D εκτυπωτών ή ο έλεγχος περισσότερων από ένα μοτέρ, λόγω της ικανότητάς του να αποθηκεύει περισσότερες οδηγίες στη μνήμη κωδικών και μια σειρά ψηφιακών και αναλογικών I / O ακίδων. Υπάρχουν τρεις τρόποι για την τροφοδοσία του σκάφους: είτε η χρήση καλωδίου USB προκειμένου να τροφοδοτηθεί η πλακέτα και να μεταφερθεί ο κωδικός στην πλακέτα είτε υπάρχει δυνατότητα να τροφοδοτηθεί

χρησιμοποιώντας το Vin της κάρτας ή μέσω της υποδοχής Power ή της ζύμης. Οι τελευταίες δύο πηγές για την τροφοδοσία της κάρτας απαιτούνται μόλις έχει ήδη χτιστεί και μεταγλωττιστεί ο κώδικας μέσα στο καλώδιο μέσω καλωδίου USB. Αυτός ο πίνακας επιδέχεται επαναφορά πολλών χρήσεων που εμποδίζει τη θύρα USB του υπολογιστή από υπερθέρμανση παρουσία υψηλού ρεύματος που ρέει μέσω του πίνακα. Οι περισσότεροι από τους υπολογιστές διαθέτουν την ικανότητα να προστατεύονται από τέτοιες συσκευές, ωστόσο, η προσθήκη της ασφάλειας παρέχει ένα επιπλέον επίπεδο προστασίας. Μπορεί να χρησιμοποιηθεί είτε για τη δημιουργία ανεξάρτητων έργων είτε σε συνδυασμό με άλλους πίνακες Arduino. Τα πιο σύνθετα έργα μπορούν να δημιουργηθούν χρησιμοποιώντας αυτό το διοικητικό συμβούλιο.

Παρακάτω απεικονίζεται η μορφή της πλακέτας. Ωστόσο θα προχωρήσουμε και σε μία ανάλυση κάθε εισόδου-εξόδου που θα συναντήσει κανείς έχοντας στα χέρια του μια τέτοια πλακέτα.



5V & 3.3V. Αυτός ο ακροδέκτης χρησιμοποιείται για την παροχή εξόδου ρυθμιζόμενης τάσης περίπου 5V. Αυτή η ρυθμιζόμενη παροχή ρεύματος ενεργοποιεί τον ελεγκτή και άλλα εξαρτήματα του πίνακα. Μπορεί να ληφθεί από το Vin της πλακέτας ή το καλώδιο USB ή άλλη ρυθμισμένη τάση τάσης 5V. Ενώ άλλη ρύθμιση τάσης παρέχεται από τον ακροδέκτη 3.3V. Η μέγιστη ισχύς που μπορεί να αντλήσει είναι 50mA.

GND. Υπάρχουν 5 αλεξίσφαιρες ακίδες που είναι χρήσιμες όταν απαιτούνται περισσότερες από μία ακίδες για το έργο.

Επαναφορά. Αυτός ο ακροδέκτης χρησιμοποιείται για την επαναφορά της πλακέτας. Η ρύθμιση αυτού του ακροδέκτη σε LOW θα επαναφέρει την πλακέτα.

Vin. Είναι η τάση εισόδου που παρέχεται στην πλακέτα και κυμαίνεται από 7V έως 20V. Η τάση που παρέχεται από την υποδοχή τροφοδοσίας μπορεί να προσεγγιστεί μέσω αυτού του πείρου. Ωστόσο, η τάση εξόδου μέσω αυτού του πείρου στην πλακέτα θα ρυθμιστεί αυτόματα σε 5V.

Σειριακή επικοινωνία. Οι RXD και TXD είναι οι σειριακοί ακροδέκτες που χρησιμοποιούνται για τη μετάδοση και λήψη σειριακών δεδομένων, δηλαδή το Rx αντιπροσωπεύει τη μετάδοση των δεδομένων ενώ το Tx χρησιμοποιείται για τη λήψη δεδομένων. Υπάρχουν τέσσερις συνδυασμοί αυτών των σειριακών καρφίτσες που χρησιμοποιούνται όπου το Serial 0 περιέχει RX (0) και TX (1), Serial 1 περιέχει TX (18) και RX (19), Serial 2 περιέχει TX (16) και RX (17) και το Serial 3 περιέχει TX (14) και RX (15).

Εξωτερικές διακοπές. Έξι ακίδες χρησιμοποιούνται για τη δημιουργία εξωτερικών διακοπών, δηλ. Διακοπή 0 (0), διακοπή 1 (3), διακοπή 2 (21), διακοπή 3 (20), διακοπή 4 (19), διακοπή 5 (18). Αυτές οι καρφίτσες παράγουν διακοπές με διάφορους τρόπους, δηλ. Παρέχουν χαμηλή τιμή, ανυψούμενη ή πτώση άκρη ή αλλαγή τιμής στις ακίδες διακοπής.

LED. Αυτός ο πίνακας διαθέτει ενσωματωμένο LED συνδεδεμένο στην ψηφιακή πύλη 13. Η υψηλή τιμή σε αυτήν την ακίδα θα ανάψει η λυχνία LED και η τιμή LOW θα την απενεργοποιήσει. Αυτό σας δίνει την αλλαγή της νοσηλευτικής ικανότητας προγραμματισμού σας σε πραγματικό χρόνο.

AREF. Το AREF αντιπροσωπεύει την αναλογική τάση αναφοράς, η οποία αποτελεί τάση αναφοράς για αναλογικές εισόδους.

Αναλογικές ακίδες. Υπάρχουν 16 αναλογικές ακίδες ενσωματωμένες στον πίνακα που φέρουν την ένδειξη A0 έως A15. Είναι σημαντικό να σημειωθεί ότι όλες αυτές οι αναλογικές ακίδες μπορούν να χρησιμοποιηθούν ως ψηφιακοί ακροδέκτες I / O. Κάθε αναλογική ακίδα διαθέτει ανάλυση 10 bit. Αυτές οι καρφίτσες μπορούν να μετρηθούν από το έδαφος σε 5V. Ωστόσο, η ανώτερη τιμή μπορεί να αλλάξει χρησιμοποιώντας τη λειτουργία AREF και analogReference ().

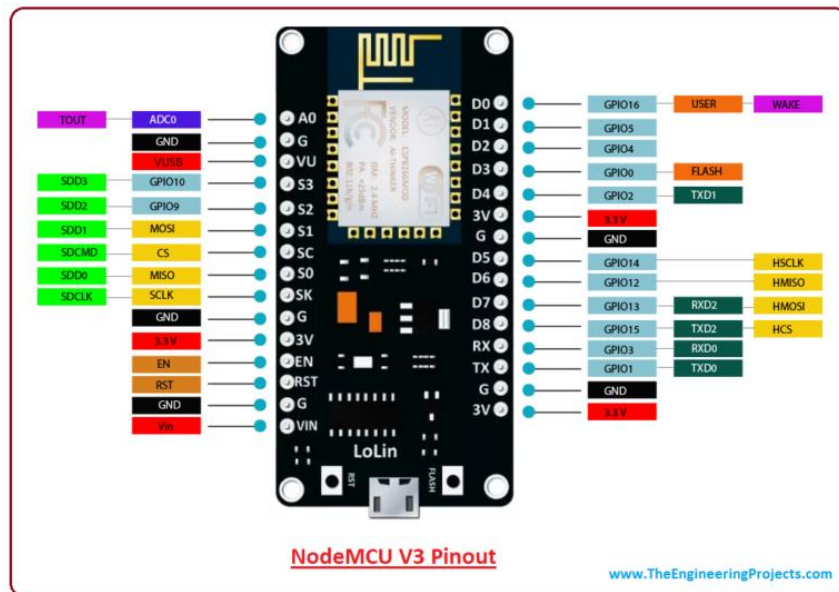
I2C. Δύο ακίδες 20 και 21 υποστηρίζουν επικοινωνία I2C όπου το 20 αντιπροσωπεύει SDA (Σειριακή Γραμμή Δεδομένων που χρησιμοποιείται κυρίως για τη συγκράτηση των δεδομένων) και το 21 αντιπροσωπεύει SCL (Γραμμή σειριακού ρολογιού που χρησιμοποιείται κυρίως για την παροχή συγχρονισμού δεδομένων μεταξύ των συσκευών)

Επικοινωνία SPI. Το SPI σημαίνει Serial Peripheral Interface που χρησιμοποιείται για τη μετάδοση δεδομένων μεταξύ του ελεγκτή και των άλλων περιφερειακών στοιχείων. Για την επικοινωνία SPI χρησιμοποιούνται τέσσερις ακίδες: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).

1.6 Μικροελεγκτής NodeMCU ESP8266

Το NodeMCU V3 είναι ένα λογισμικό ανοιχτού κώδικα και ένα κιτ ανάπτυξης υψίστης σημασίας για το σχεδιασμό του δικού σας προϊόντος IoT με λίγες γραμμές σεναρίου Lua. Πολλαπλοί ακροδέκτες GPIO στον πίνακα επιτρέπουν τη σύνδεση της κάρτας με άλλα περιφερειακά καθώς επίσης και την παραγωγή σειριακών επικοινωνιών PWM, I2C, SPI και UART. Η διασύνδεση της μονάδας χωρίζεται κυρίως σε δύο μέρη, συμπεριλαμβανομένων τόσο του Firmware όσο και του Hardware, όπου ο πρώην τρέχει στο ESP8266 Wi-Fi SoC και αργότερα βασίζεται στην ηλεκτρονική μονάδα ESP-12. Το υλικολογισμικό βασίζεται στη γλώσσα προγραμματισμού Lua - A που είναι εύκολη στην εκμάθηση, παρέχοντας ένα απλό περιβάλλον προγραμματισμού με μια γρήγορη γλώσσα scripting που επιτρέπει τη σύνδεση με μια γνωστή κοινότητα προγραμματιστών. Παράλληλα, το firmware ανοιχτού κώδικα παρέχει στο χρήστη την ευελιξία για επεξεργασία, τροποποίηση και να ανοικοδόμηση της υπάρχουσας μονάδας και την αλλαγή ολόκληρης της διεπαφής μέχρι να επιτευχθεί η βελτιστοποίηση της μονάδας σύμφωνα με τις απαιτήσεις του εκάστοτε χρήστη. Ο μετατροπέας USB σε UART προστίθεται στην ενότητα που βοηθά στη μετατροπή των δεδομένων USB σε δεδομένα UART, τα οποία κατανοούν κυρίως τη γλώσσα της σειριακής επικοινωνίας. Αντί της κανονικής θύρας USB, η θύρα MicroUSB περιλαμβάνεται στη μονάδα που την συνδέει με τον υπολογιστή για διπλούς σκοπούς: τον προγραμματισμό και ενεργοποίηση της πλακέτας. Η πλακέτα ενσωματώνει LED κατάσταση που αναβοσβήνει και σβήνει αμέσως, δείχνοντας στο χρήστη την τρέχουσα κατάσταση της μονάδας δηλαδή πιο συγκεκριμένα την ένδειξη σωστής λειτουργίας όταν πραγματοποιείται σύνδεση με τον υπολογιστή. Η δυνατότητα της μονάδας να δημιουργήσει μια άψογη σύνδεση WiFi μεταξύ δύο καναλιών την καθιστά ιδανική επιλογή για την ενσωμάτωσή της με άλλες ενσωματωμένες συσκευές όπως το Raspberry Pi.

Το NodeMCU V3 έρχεται με έναν αριθμό ακίδων GPIO. Παρακάτω απεικονίζεται η μορφή της πλακέτας και πραγματοποιείται η ανάλυση των εισόδων αλλά και εξόδων της.



Υπάρχει μια ειλικρινή διαφορά μεταξύ Vin και VU όπου πρώην είναι η ρυθμισμένη τάση που μπορεί να σταθεί κάπου μεταξύ 7 έως 12 V ενώ αργότερα είναι η τάση ρεύματος για USB που πρέπει να διατηρηθεί περίπου 5 V.

- Ανοιχτή πηγή
- Εξοπλισμός που μοιάζει με Arduino
- LED κατάστασης
- Θύρα MicroUSB
- Κουμπιά επαναφοράς / φλας
- Διαδραστική και προγραμματιζόμενη
- Χαμηλό κόστος
- ESP8266 με ενσωματωμένο wifi
- Μετατροπέας USB σε UART
- GPIO ακίδες

Όπως αναφέρθηκε παραπάνω, ένα καλώδιο που υποστηρίζει τη θύρα micro USB χρησιμοποιείται για τη σύνδεση του πίνακα. Καθώς συνδέετε τον πίνακα με έναν υπολογιστή, η λυχνία LED θα αναβοσβήνει. Ενδέχεται να χρειαστεί να εγκατασταθούν στον υπολογιστή σας ορισμένα προγράμματα οδήγησης, αν δεν εντοπίσει την πλακέτα NodeMCU. Μπορείτε να κατεβάσετε το πρόγραμμα οδήγησης από αυτήν τη σελίδα.

Σημείωση: Χρησιμοποιούμε λογισμικό Arduino IDE για τον προγραμματισμό αυτής της μονάδας. Είναι σημαντικό να σημειωθεί ότι η διαμόρφωση των ακίδων που εμφανίζεται στον πίνακα είναι διαφορετική από τη διαμόρφωση που χρησιμοποιούμε για τον προγραμματισμό της πλακέτας στο λογισμικό δηλαδή όταν γράφουμε κώδικα για τον πείρο στόχευσης 16 στο IDE του Arduino, θα βοηθήσει πραγματικά να καθορίσει την επικοινωνία με τον ακροδέκτη D0 της μονάδας.

Το παρακάτω σχήμα δείχνει τη διαμόρφωση των ακίδων που θα χρησιμοποιηθεί στο IDE του Arduino.

Μπορείτε να δείτε από την εικόνα pinout παραπάνω, υπάρχουν πέντε καρφίτσες γείωσης και τρεις 3V3 καρφίτσες στο διοικητικό συμβούλιο. Ο πίνακας μπορεί να τροφοδοτηθεί με τους ακόλουθους τρεις τρόπους.

Τροφοδοσία USB. Αποδεικνύεται μια ιδανική επιλογή για προγράμματα φόρτωσης εκτός αν το έργο που σχεδιάζετε να σχεδιάσετε απαιτεί ξεχωριστή διεπαφή, δηλαδή αποσυνδεδεμένη από τον υπολογιστή.

Παρέχετε 3.3V. Αυτή είναι μια άλλη μεγάλη επιλογή για την ενεργοποίηση της μονάδας. Αν έχετε το δικό σας ρυθμιστή εκτός του σκάφους, μπορείτε να δημιουργήσετε μια άμεση πηγή ενέργειας για το kit ανάπτυξης.

Power Vin. Αυτός είναι ένας ρυθμιστής τάσης που έρχεται με την ικανότητα να υποστηρίξει έως και 800 mA. Μπορεί να χειριστεί κάπου μεταξύ 7 έως 12 V. Δεν μπορείτε να ενεργοποιήσετε τις συσκευές που λειτουργούν με 3,3 V, καθώς αυτός ο ρυθμιστής δεν μπορεί να παράγει τόσο χαμηλά όσο τα 3,3V.

1.7 Bluetooth Module



ALLNET 4duino HC-05 Ασύρματη μονάδα Bluetooth (master και slave)

Η μονάδα Bluetooth HC-05 με BT 2.0 μπορεί να λειτουργήσει τόσο σε master mode όσο και σε λειτουργία slave. Το υλικό υποστηρίζει το σετ εντολών AT για τη ρύθμιση της ταχύτητας μετάδοσης από 2400 σε 1.382.400.

Η προεπιλεγμένη ρύθμιση για τη σειριακή θύρα είναι [9600, N, 8.1] και ο κωδικός πρόσβασης είναι: 1234.

Το σετ εντολών AT μπορεί να χρησιμοποιηθεί για να αλλάξει η ταχύτητα baud, το όνομα της συσκευής, ο κωδικός πρόσβασης και η λειτουργία master ή slave.

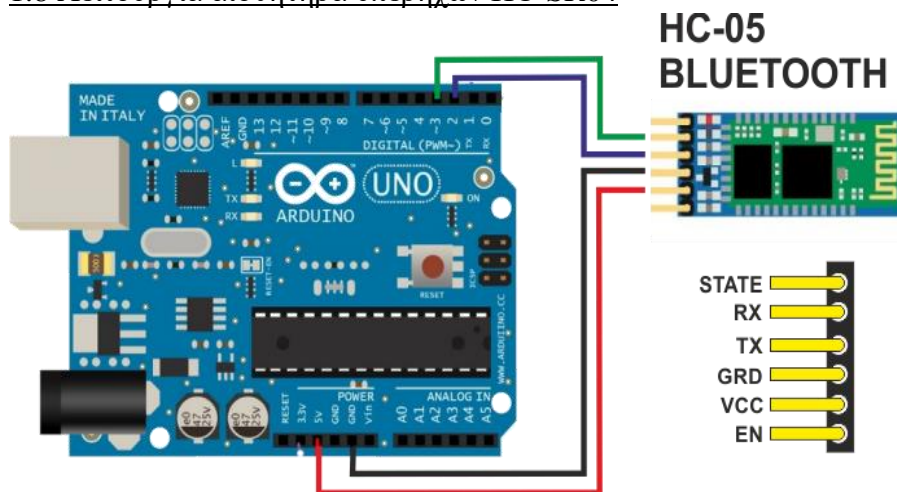
Λεπτομέρειες:

- Πρωτόκολλο Bluetooth: προδιαγραφή Bluetooth v2.0 + EDR
- Πρωτόκολλο USB: USB V1.1 / 2.0
- Συχνότητα: Bandwidth ISM 2.4 GHz
- Διαμόρφωση: GFSK (Gaussian shift keying)
- Ισχύς μετάδοσης: = 4 dBm, κλάση 2
- Ευαισθησία: = -84 dBm σε 0,1% BER
- Βαθμολογία: Ασύγχρονης: 2.1 Mbps (max.) / 160 kbps
- Σύγχρονοι: 1 Mbps / 1 Mbps

διαθέτει • Ασφάλεια: τον έλεγχο ταυτότητας και κρυπτογράφηση

- Προφίλ Υποστήριξη: σειριακή θύρα Bluetooth (Master και Slave)
- Τροφοδοσία: 3.3 V DC, 50 mA
- Θερμοκρασία λειτουργίας: -5 - +45 centig

1.8 Λειτουργία αισθητήρα υπερήχων HC-SR04

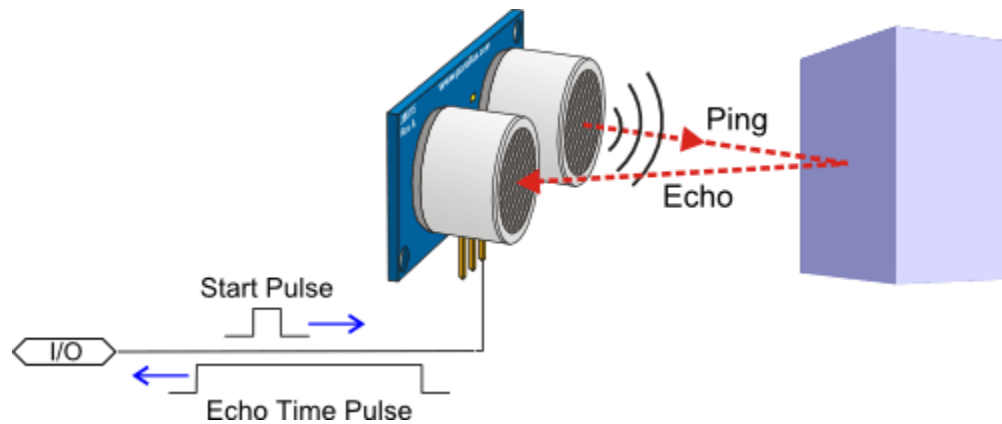


Ο αισθητήρας υπερήχων μπορεί να:

- Ανιχνεύει κοντινά αντικείμενα μέσα σε μία κλίμακα από 2 cm έως 3 μέτρα
- Μέτρα αποστάσεις χρησιμοποιώντας ήχο υψηλής συχνότητας (40 kHz)

Η μονάδα HC-SR04 είναι μια μονάδα όλα-σε-ένα για την ακριβή μέτρηση αποστάσεων μεταξύ της ίδιας και αντικείμενων σε κοντινή απόσταση. Το εύρος λειτουργίας της είναι από **2 cm έως 3 μέτρα**. Ο Αισθητήρας καταναλώνει ελάχιστη ενέργεια και είναι ιδανικός για χρήση σε αυτοκινούμενα ρομπότ, συστήματα ασφαλείας, καθώς και κάθε άλλη περίπτωση για την ανίχνευση κοντινών αντικειμένων, ή τη μέτρηση της απόστασής από τον αισθητήρα.

Η ανίχνευση αντικειμένων και η μέτρηση απόστασης πραγματοποιείται με την αποστολή ενός σύντομου τόνου υψηλής συχνότητας (40 kHz) από την μονάδα (Trigger). Ο χρόνος που χρειάζεται για την ανίχνευση του ήχου επιστροφής (Echo) αντιπροσωπεύει την απόσταση μεταξύ της μονάδας και οποιαδήποτε αντικείμενου που βρίσκεται μπροστά της. Χρησιμοποιώντας βασικές αριθμητικές πράξεις, μπορούμε να μετατρέψετε την καθυστέρηση του ήχου σε ίντσες ή εκατοστά. Οι συχνότητες ήχου που χρησιμοποιούνται στον αισθητήρα βρίσκονται στην υπερηχητική περιοχή, και είναι πέρα από την ανθρώπινη ακοή.



Για να μπορέσουμε να τηλεκατευθύνουμε το ρομποτικό όχημα μέσω μιας εφαρμογής που θα δημιουργήσουμε με τη βοήθεια του MIT App Inventor, θα χρειαστούμε μια μονάδα Bluetooth κατάλληλη για το Arduino Uno. Η μονάδα HC06 αναλαμβάνει την σειριακή επικοινωνία του οχήματος με τη φορητή συσκευή χωρίς να χρειάζονται καλώδια.

Παρέχει:

- Ενσωματωμένη κεραία
- Ταχύτητα μετάδοσης **9600 bps**
- Μέγιστη εμβέλεια **9-10μ**
- Bluetooth έκδοση: **V2.0+EDR**
- Εύκολο προγραμματισμό μέσω των pins **0->RXD** και **1->TXD** του Arduino

1.8.1 Σύνδεση στο Arduino και προγραμματισμός.

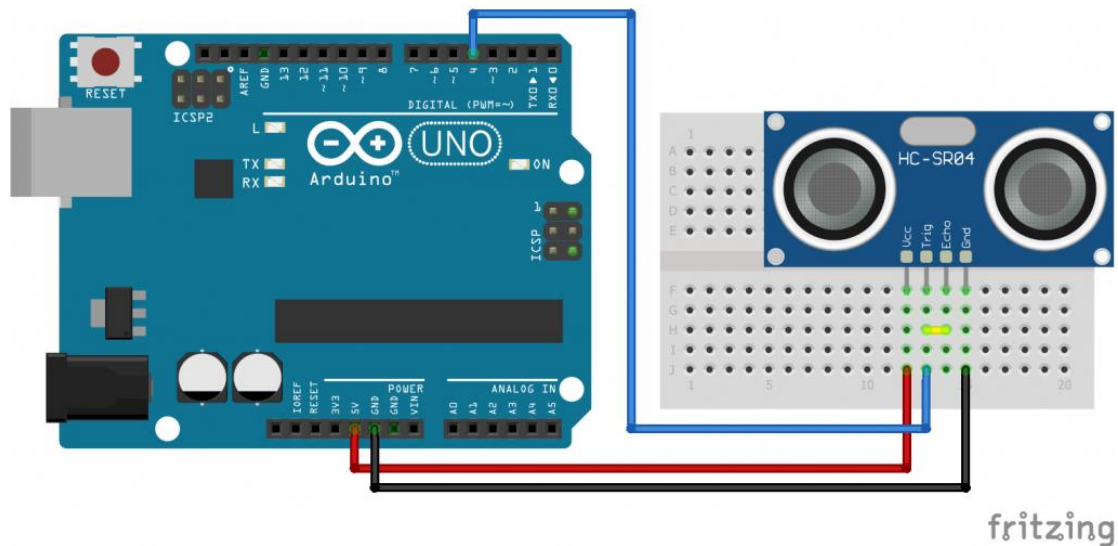
Ο αισθητήρας HC-SR04 διαθέτει 4 pins:

- Το VCC, στο οποίο δέχεται 5V για να μπει σε λειτουργία,
- το GND, το οποίο το συνδέουμε στη γείωση,
- το TRIG (Trigger), το οποίο ενεργοποιεί το ηχείο για να στείλει το υπέρηχο και
- το ECHO, το οποίο είναι συνδεδεμένο με την είσοδο του μικροφώνου και αντιλαμβάνεται την επιστροφή του ήχου.

Υπάρχουν δυο τρόποι για να συνδέσουμε τον αισθητήρα στο Arduino:

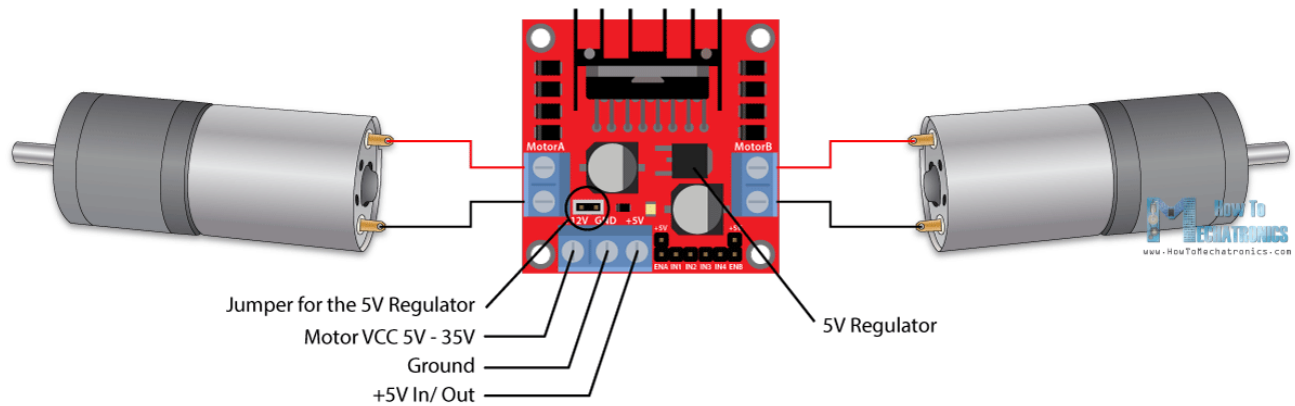
- Είτε συνδέουμε το TRIG και το ECHO σε διαφορετικά digital pins και αναλαμβάνουμε στο πρόγραμμα μας να κάνουμε εμείς όλη την σκληρή δουλειά του υπολογισμού της απόστασης,
- είτε συνδέουμε το TRIG και το ECHO στο ίδιο digital pin και χρησιμοποιούμε μια έτοιμη βιβλιοθήκη που μας απαλλάσσει από αυτό το φορτίο.

Όπως είναι κατανοητό, ο πρώτος τρόπος μας δίνει μεγαλύτερο έλεγχο στο πως θα υπολογίσουμε την απόσταση και έχει μεγαλύτερο εκπαιδευτικό ενδιαφέρον. Θα ξεκινήσω λοιπόν με τον δεύτερο (εύκολο τρόπο) και παρακάτω θα ασχοληθώ με τα πιο ενδιαφέροντα.



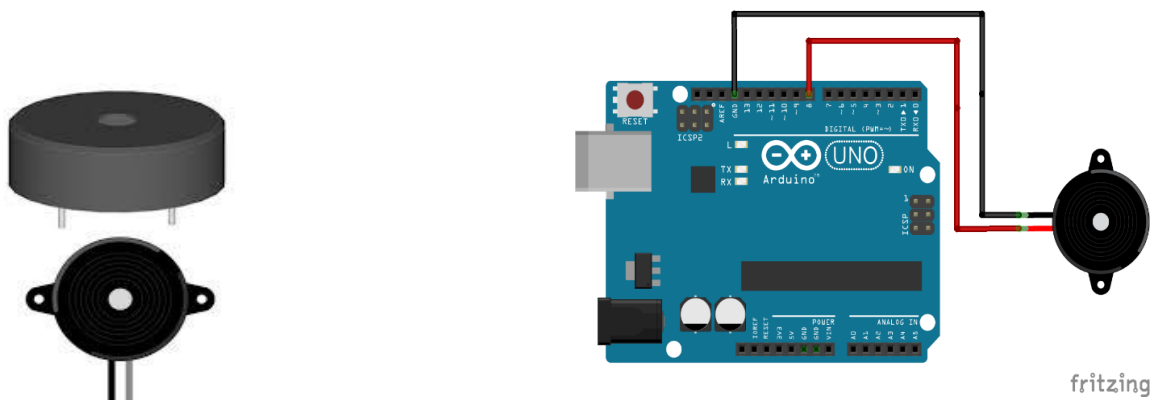
1.9 L298N Dual Motor Driver Module

Η μονάδα L298N Dual Motor Driver Module, μας επιτρέπει να ελέγχουμε την ταχύτητα και την κατεύθυνση δύο κινητήρων συνεχούς ρεύματος ή ένα βηματικό διπολικό μοτέρ. Η μονάδα L298N μπορεί να χρησιμοποιηθεί με κινητήρες που έχουν τάση μεταξύ 5 και 35V DC. Το L298N είναι ένα ολοκληρωμένο μονολιθικό κύκλωμα. Πρόκειται για ένα υψηλής τάσης, διπλό οδηγό πλήρους γέφυρας που έχει σχεδιαστεί για να δέχεται στάνταρ επίπεδα λογικής TTL και να κατευθύνει επαγωγικά φορτία όπως ρελέ, ηλεκτρομαγνητικές βαλβίδες, DC και βηματικούς κινητήρες.



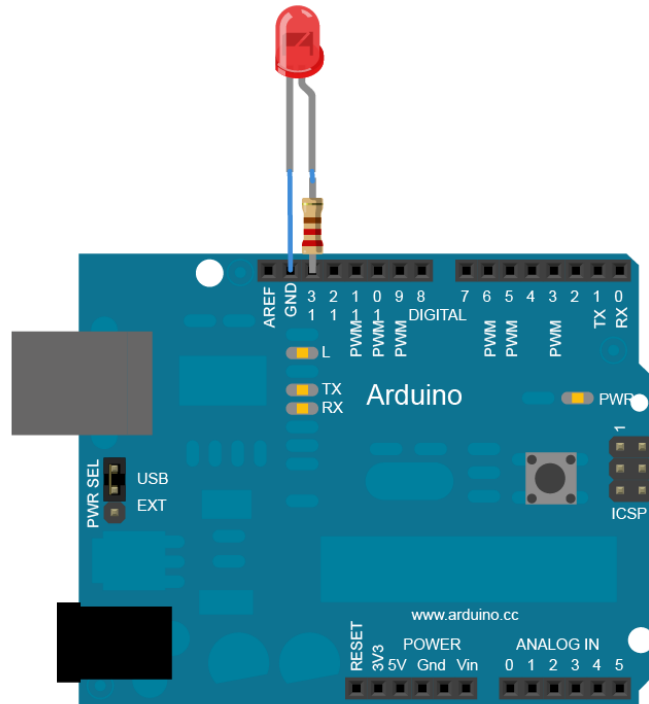
1.10 Buzzer

Το Buzzer είναι ένα μονοτονικό ηχείο. Είναι ευρέως γνωστό και χρησιμοποιείται σε πλήθος εφαρμογών όπως υπολογιστές, συστήματα συναγερμών, ξυπνητήρια ακόμα και στα ηλεκτρολόγια των ATMS. Αυτό που πρέπει να γνωρίζουμε για το Buzzer είναι να χειριζόμαστε τις συναρτήσεις Tone και noTone για τον προγραμματισμό. Η λειτουργία του είναι απλή στέλνοντας ένα σήμα της τάξης του 1KHz αυτό λειτουργεί. Ανάλογα με την συχνότητα που το συντονίζουμε και το χρόνο που σταματάει αλλάζει και ο ήχος που παράγει.



1.11 Led Λάμπες

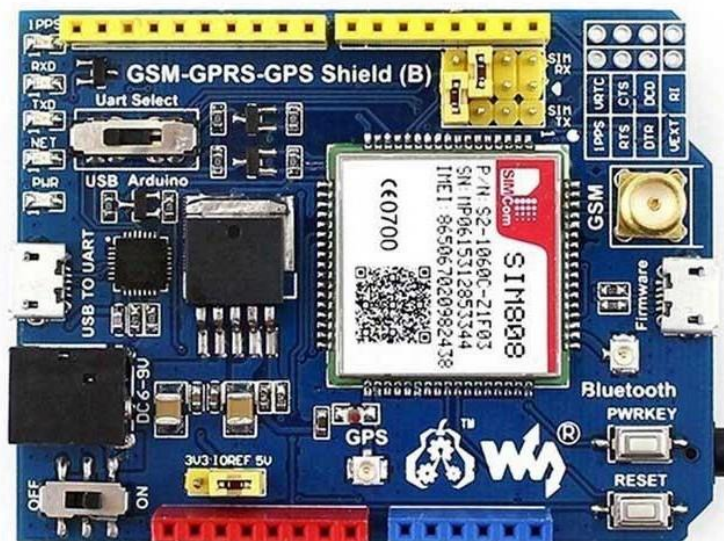
Οι led λάμπες είναι το πιο συνηθισμένη «επέκταση» για Arduino και είναι το βασικό βήμα για να ξεκινήσεις την εκμάθηση του Arduino και τον τρόπο λειτουργίας του. Μια σημαντική σημείωση είναι πως για την σωστή λειτουργία των λαμπτήρων χρειάζεται πάντα μία αντίσταση σε σειρά με την λάμπα.



1.12 Μονάδα sim808

Η μονάδα SIM808 σχεδιασμένη για την παγκόσμια αγορά, είναι ενσωματωμένη σε μία πλακέτα που περιέχει τις τεχνολογίες GSM / GPRS/GPS και την λειτουργία BT(Bluetooth). Οι τεχνολογίες GSM / GPRS αποτελούνται από μια μονάδα τετραπλής ζώνης που λειτουργεί σε συχνότητες GSM 850MHz, EGSM 900MHz, DCS 1800MHz και PCS 1900MHz. Το SIM808 διαθέτει GPRS πολλαπλών υποδοχών κατηγορίας 12, κλάσης 10 και υποστηρίζει τα προγράμματα κωδικοποίησης GPRS CS-1, CS-2, CS-3 και CS-4. Το μοντέλο που έχει χρησιμοποιηθεί αποτελεί

διαμόρφωση της εταιρίας Waveshare.

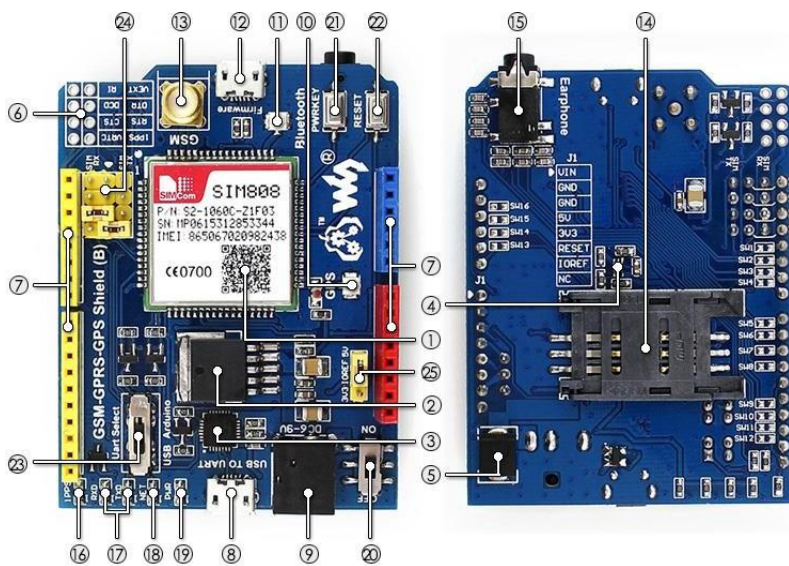


Οι διασυνδέσεις υλικού της μονάδας είναι οι εξής:

- Υποστήριξη 4 *4*2 πηκτρολογίων.
- Μια σειριακή θύρα (micro usb)
- Ένα USB, οι διασυνδέσεις USB μπορούν να εντοπίσουν σφάλματα, να μεταφορτώσουν λογισμικό(firmware).
- Τα κανάλια ήχου που περιλαμβάνουν είσοδο μικροφώνου και έξοδο δέκτη.
- Μια διεπαφή κάρτας SIM που υποστηρίζει κάρτα 3.3 Volts.
- Διεπαφή φόρτισης.
- Προγραμματιζόμενες είσοδοι και έξοδοι γενικού σκοπού.
- Υποστήριξη της λειτουργίας Bluetooth.
- Υποστήριξη PWM και ADC.
- Διεπαφή κάρτας PCM/SPI/SD, μόνο μία λειτουργία είναι προσβάσιμη συγχρονισμένα(Η προεπιλεγμένη λειτουργία είναι PCM).

Στο παρακάτω σχήμα θα παρουσιαστούν όλες οι βασικές λειτουργίες όπως :

- Σύστημα GSM
- Σύστημα GPS
- Σύστημα GPRS
- Ενδεικτικά LED
- Τη μονάδα ρύθμισης της συχνότητας του GSM
- Διασύνδεση κεραίας
- Άλλες διασυνδέσεις



Τα βασικά στοιχεία της επιφάνειας της πλακέτας είναι:

1. SIM808 :Κεντρικός επεξεργαστής
2. MIC29302 power chip.
3. CP2102: USB TO UART converter.
4. SMF05C: δίοδος TVS.

5. 1N5408: ανορθωτής.
6. Υποδεικνύει την λειτουργία της κάθε ακίδας με κίτρινο χρώμα στα αριστερά του.
7. Λειτουργικές ακίδες που μεταφέρονται από την πλακέτα που είναι συνδεδεμένη κάτω από το sim808.
8. USB σε UART επικοινωνία.
9. Σημείο τροφοδοσίας συνεχούς ρεύματος.
10. Σημείο σύνδεσης gprs κεραίας.
11. Σημείο σύνδεσης bluetooth κεραίας.
12. Υποδοχή αναβάθμισης firmware.
13. Σημείο σύνδεσης gsm κεραίας.
14. Σημείο εισαγωγής κάρτας SIM.
15. Σημείο εισαγωγής ακουστικών ή μικροφώνου 3.5mm.
16. Led που κάνει γνωστή τη λειτουργία του GPS.
17. Led που κάνει γνωστή την κατάσταση στα pin ανάγνωσης και μετάδοσης δεδομένων tx/rx.
18. NET led : ενεργοποιείτε όταν πατηθεί το νούμερο 8 και αναβοσβήνει γρήγορα όταν δεν έχει φορτωθεί κάποιο πρόγραμμα, ενώ αναβοσβήνει ανά 3 δεύτερα όταν εκτελεί κάποιο πρόγραμμα.
19. Power led
20. On/Off διακόπτης
21. Κουμπί ενεργοποίησης
22. Κουμπί επανεκκίνησης
23. Διακόπτης που ορίζει το είδος της επικοινωνίας. Δεξιά για σύνδεση με το arduino και άλλες πλακέτες και αριστερά για σειριακή επικοινωνία .
24. Ακίδες μετάδοσης και ανάγνωσης δεδομένων.
25. Επιλογέας volt λειτουργίας σειριακής.

Κεφάλαιο 2: Εργαλεία ανάπτυξης λογισμικού

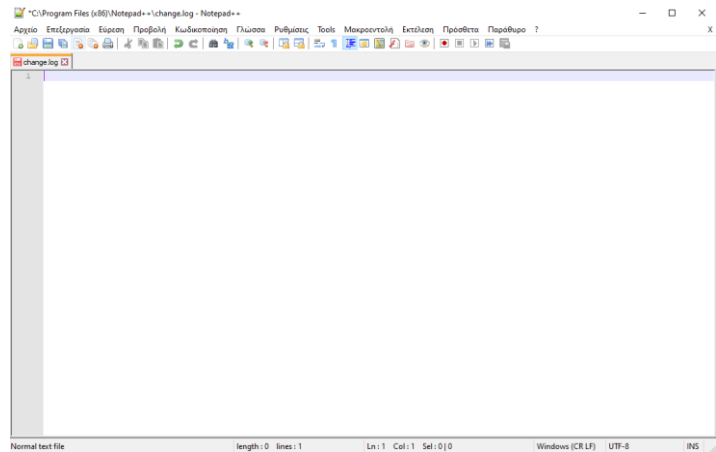
2.1 Fritzing

Το Fritzing είναι ένα πρόγραμμα ανοιχτού κώδικα για την ανάπτυξη ερασιτεχνικού σχεδιασμού (CAD) για το σχεδιασμό ηλεκτρονικού εξοπλισμού. Υποστηρίζει τους σχεδιαστές να σχεδιάζουν ένα πιο «πραγματικό» κύκλωμα και να ξεφεύγουν από τα πειραματικά. Αναπτύχθηκε στο Πανεπιστήμιο των εφαρμοσμένων επιστημών του Πότσταμ. Το λογισμικό δημιουργήθηκε στο πνεύμα της γλώσσας προγραμματισμού επεξεργασίας και του μικροελεγκτή Arduino και επιτρέπει σε χομπίστες σχεδιαστές να τεκμηριώνουν τα πρωτότυπα με βάση το Arduino και να δημιουργούν μια διάταξη PCB για την κατασκευή. Ο συνδεδεμένος ιστότοπος βοηθά τους χρήστες να μοιράζονται και να συζητούν τα σχέδια και τις εμπειρίες τους καθώς και να μειώνουν το κόστος κατασκευής.



2.2 Notepad++

Το Notepad++ είναι ένας δωρεάν και ανοιχτού κώδικα επεξεργαστής κειμένου κυρίως για προγραμματιστές αφού προσφέρει πολλές επιλογές και δυνατότητες. Είναι ένα «ελαφρύ» πρόγραμμα που δεν ξεπερνά τα 3MB. Το πρόγραμμα προσφέρει και δυνατότητα αυτόματης συμπλήρωσης auto-completion και φυσικά να υπογραμμίζει τα διάφορα κομμάτια του κώδικα ανάλογα με το συντακτικό της εκάστοτε γλώσσας.



2.4 Arduino IDE

Το arduino και άλλες πλακέτες έχουν την ανάγκη επικοινωνίας και προγραμματισμού μέσω ενός υπολογιστή, αυτή η ανάγκη είναι δυνατή μέσω της USB θύρας τους Drivers αλλά και του προγράμματος Arduino IDE. Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java, που λειτουργεί σε πολλές πλατφόρμες και προέρχεται από το IDE για τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring. Έχει σχεδιαστεί για να εισαγάγει στον προγραμματισμό τους καλλιτέχνες και τους νέους που δεν είναι εξοικειωμένοι με την ανάπτυξη λογισμικού. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης και ο συνδυασμός αγκύλων και είναι επίσης σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα με ένα μόνο κλικ. Δεν υπάρχει συνήθως καμία ανάγκη να επεξεργαστείτε αρχεία make ή να τρέξετε προγράμματα σε ένα περιβάλλον γραμμής εντολών. Ένα πρόγραμμα ή κώδικας που γράφτηκε για Arduino ονομάζεται σκίτσο (sketch).

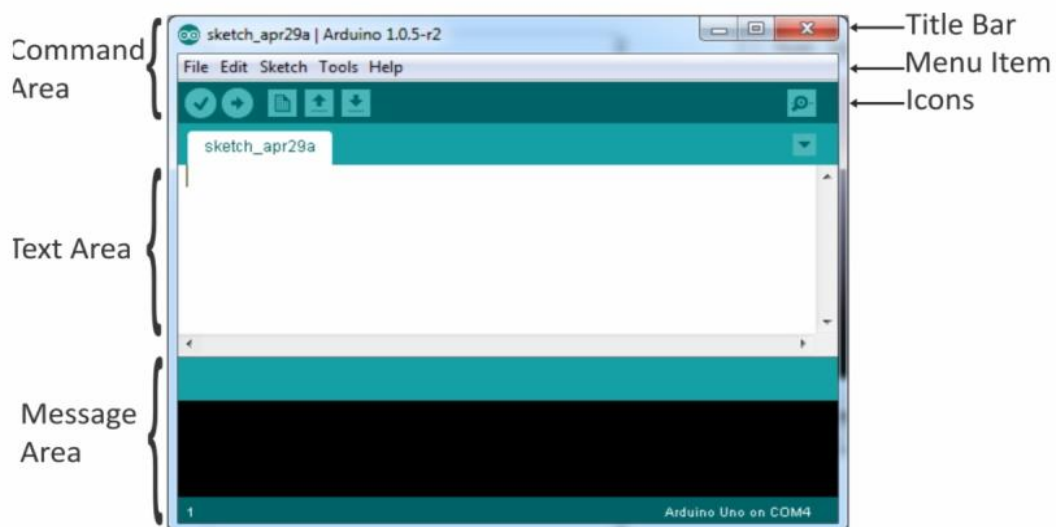
Τα Arduino προγράμματα είναι γραμμένα σε C ή C++. Το Arduino IDE έρχεται με μια βιβλιοθήκη λογισμικού που ονομάζεται "Wiring", από το πρωτότυπο σχέδιο Wiring, γεγονός που καθιστά πολλές κοινές λειτουργίες εισόδου/εξόδου πολύ πιο εύκολες.

Arduino IDE παρέχει:

- Ένα πρακτικό περιβάλλον για τη συγγραφή των προγραμμάτων, με συντακτική χρωματική σήμανση ανάλογα τον τύπο της εντολής.
- Μερικές έτοιμες βιβλιοθήκες αλλά και πολλές διαθέσιμες για λήψη από το διαδίκτυο.
- Τον compiler για τη μεταγλώττιση των sketch.
- Μία σειριακή οθόνη (serial monitor) που παρακολουθεί τις επικοινωνίες της σειριακής (USB), αναλαμβάνει να στείλει αλφαριθμητικά στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για την αποσφαλμάτωση των sketch.
- Την επιλογή για ανέγερση των μεταγλωττισμένων sketch στο Arduino.

Και διαιρείται σε 3 μέρη

- Δομή (structure)
- Τιμές (values)
- Συναρτήσεις (functions)



Κεφάλαιο 3: Δημιουργία Εφαρμογών

Σε αυτό το κεφάλαιο θα αναλύσουμε το περιβάλλον ανάπτυξης εφαρμογών που χρησιμοποιήσαμε για να φτιάξουμε τις δικιές μας εφαρμογές τον τρόπο δημιουργίας των εφαρμογών, να αναλύσουμε τις εντολές που χρησιμοποιούμε και τον λόγο που χρησιμοποιούμε την κάθε μία εντολή.

3.1 Το MIT App Inventor

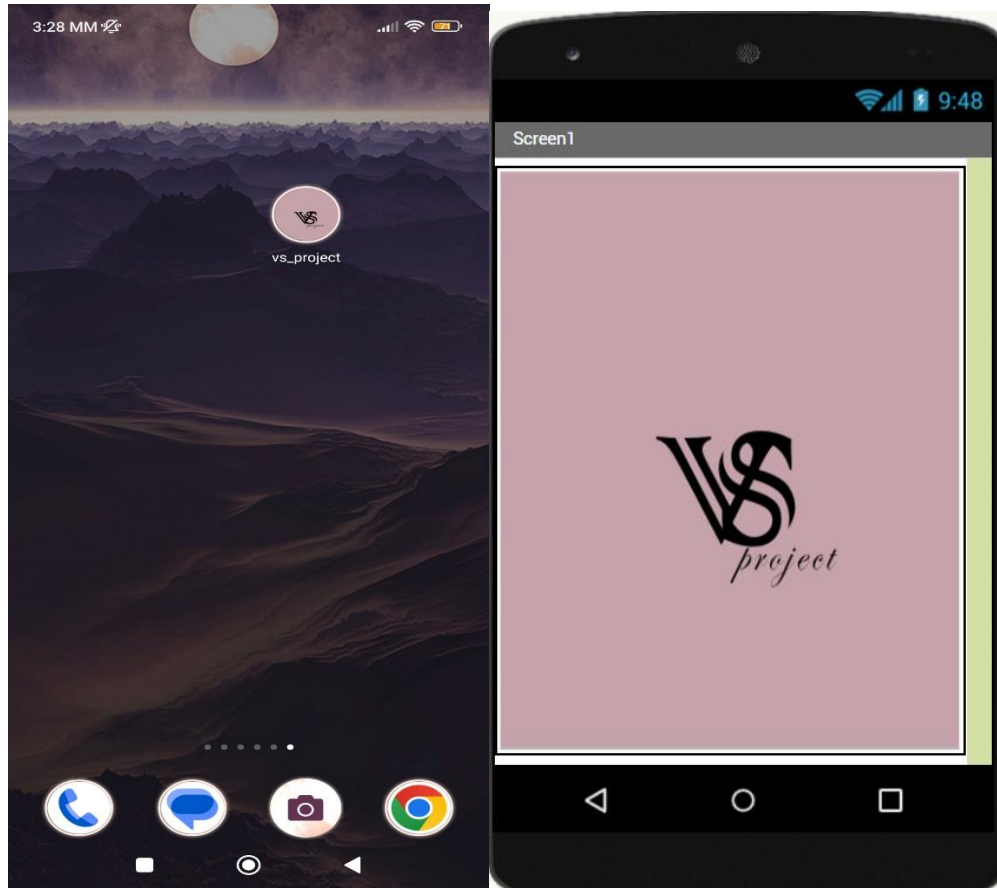
Για το χειρισμό του οχήματος μας επιδιώξαμε τα υλοποιήσουμε μια εφαρμογή μέσω του MIT App Inventor. Ο χειρισμός οπότε του project μας είναι μια εφαρμογή κατασκευασμένη εξολοκλήρου από εμάς. Η εφαρμογή αυτή φτιάχτηκε μέσω του MIT App Inventor, το οποίο είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών που προσφέρθηκε αρχικά από την Google και τώρα από το Ινστιτούτο Τεχνολογίας της Μασαχουσέτης (MIT). Το πρόγραμμα αυτό είναι πολύ φιλικό στους αρχάριους προγραμματιστές καθώς πρόκειται για ένα δωρεάν λογισμικό ανοικτού κώδικα που εκδίδεται με διπλή άδεια χρήσης πολλαπλών αδειών χρήσης: μια άδεια Creative Commons Attribution ShareAlike 3.0 Unported και μια άδεια Apache 2.0. Χρησιμοποιεί ένα γραφικό περιβάλλον εργασίας (GUI) παρόμοιο με τις γλώσσες προγραμματισμού Scratch (γλώσσα προγραμματισμού) και το StarLogo, το οποίο επιτρέπει στους χρήστες να μεταφέρουν και να αποθέτουν οπτικά αντικείμενα για να δημιουργήσουν μια εφαρμογή που μπορεί να τρέξει σε κινητές συσκευές. Επίσης είναι ένα χρήσιμο εργαλείο και για τους έμπειρους προγραμματιστές και τους καθηγητές προγραμματισμούς διότι μπορούν μέσω αυτού να μεταλαμπαδεύσουν τις γνώσεις τους στους υποψήφιους προγραμματιστές με εύγλωττο τρόπο. Το MIT App Inventor αρχικά δημιουργήθηκε και προσφέρθηκε από τη Google αλλά στη συνέχεια πέρασε στα χέρια του Ινστιτούτο Τεχνολογίας της Μασαχουσέτης (MIT), το οποίο το ανέπτυξε περαιτέρω και το έκανε να παράγει λογισμικό εφαρμογών για τα δυο μεγάλα λειτουργικά το Android & Iphone. Ο App Inventor και τα έργα στα οποία βασίζεται ενημερώνονται από τις θεωρίες μάθησης κατασκευαστών, οι οποίες τονίζουν ότι ο προγραμματισμός μπορεί να αποτελέσει μέσο για την εμπλοκή ισχυρών ιδεών μέσω της ενεργού μάθησης. Ως εκ τούτου, αποτελεί μέρος μιας συνεχιζόμενης κίνησης στον τομέα των ηλεκτρονικών υπολογιστών και της εκπαίδευσης που

ξεκίνησε με το έργο του Seymour Papert και της ομάδας λογοτύπου MIT στη δεκαετία του 1960 και επίσης εκδηλώθηκε με το έργο του Mitchel Resnick για τα Lego Mindstorms και StarLogo.

3.2 Εφαρμογή για χειρισμό του οχήματος μέσω Bluetooth

Για το χειρισμό του οχήματος μας χρησιμοποιούμε δύο μέσα και έτσι προκύπτει και η δημιουργία δυο εφαρμογών για τη για τον απομακρυσμένο χειρισμό του project μας. Αρχικά σε αυτή τη παράγραφο θα εξετάσουμε την επικοινωνία και χειρισμό μέσω Bluetooth.

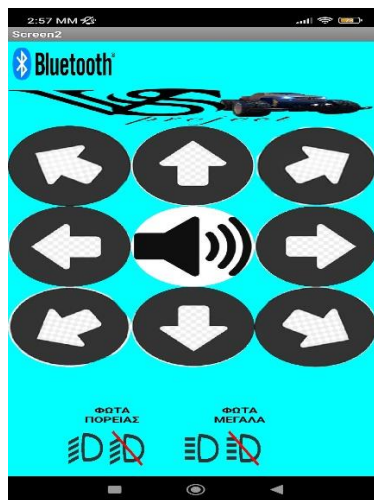
Αρχικά για την δημιουργία ενός ωραίου αισθητικού περιβάλλοντος τονίζοντας και την προσπάθεια της ομάδας μας χρησιμοποιώντας το Adobe Photoshop φτιάξαμε το λογότυπο της ομάδας μας το οποίο είναι και η πρώτη εικόνα που αντικρίζουμε όταν ανοίγουμε το application στο κινητό μας τηλέφωνο, ταυτόχρονα λειτουργεί και σαν start button καθώς με ένα κλικ στην οθόνη σε οδηγεί στην κύρια εφαρμογή χειρισμού του οχήματος.



Για την επίτευξη του προαναφερθέντος έγιναν η ακόλουθες ενέργειες από άποψη προγραμματισμού.



Με το πάτημα του λογότυπου στην αρχή της εφαρμογής μας εμφανίζει τη δεύτερη οθόνη όπως προαναφέρθηκε. Η δεύτερη της εφαρμογής αφορά τον κύριο χειρισμό του οχήματος μας και μέσα από τον οποίο θα γίνεται η κίνηση και ο ορισμό κατεύθυνσης, η ενεργοποίηση των ληχνιών βραδινής πορείας του οχήματος μας , η ενεργοποίηση της ηχητικής σήμανσης ‘‘κόρνα’’ καθώς επίσης και το σημαντικότερο κομμάτι για τη λειτουργία όλου αυτού που είναι η σύζευξη του οχήματος με το application.

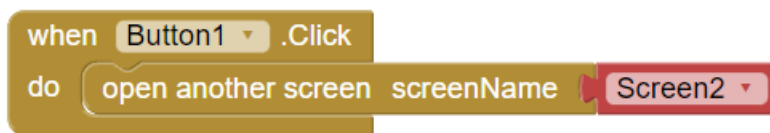


1. Το Bluetooth button είναι για τις ρυθμίσεις σύνδεσης του Bluetooth της συσκευής με το ρομπότ.
2. Το button on για τα μεγάλα φώτα είναι για την ενεργοποίηση των μεγάλων φωτών μπροστά και για τα φώτα πορείας πίσω.
3. Το button off για τα μεγάλα φώτα είναι για την απενεργοποίηση των μεγάλων φωτών μπροστά και για τα φώτα πορείας πίσω.
4. Το button on για τα προβολάκια είναι για την ενεργοποίηση των μικρών φωτών μπροστά.
5. Το button off για τα προβολάκια είναι για την απενεργοποίηση των μικρών φωτών μπροστά.
6. Το button του ηχείου είναι για την ενεργοποίηση της κόρνας - buzzer
7. Το button Up and Left είναι για την εμπρόσθια και αριστερή κίνηση του ρομπότ.

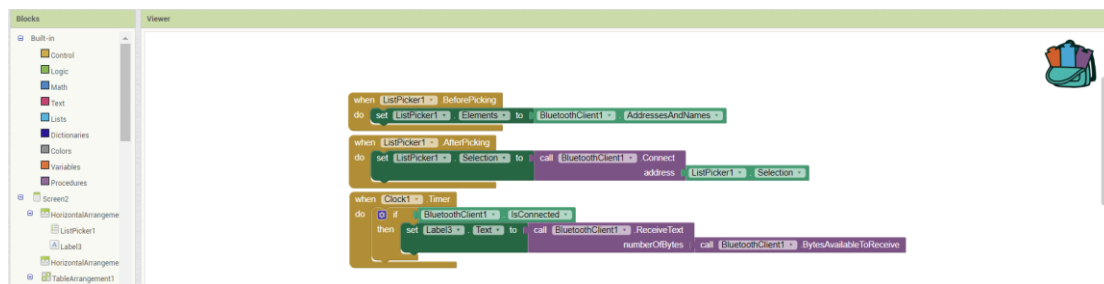
8. Το button Up and Right είναι για την εμπρόσθια και δεξιά κίνηση του ρομπότ.
9. Το button Left είναι για την αριστερόστροφη περιστροφή των εμπρόσθιων τροχών.
10. Το button Right είναι για την δεξιόστροφη περιστροφή των εμπρόσθιων τροχών.
11. Το button Back and Left είναι για την οπίσθια και αριστερή κίνηση του ρομπότ.
12. Το button Back and Right είναι για την οπίσθια και δεξιά κίνηση του ρομπότ.

3.2.1 Ανάλυση Block προγραμματισμού του Bluetooth Application

Ανοίγοντας την εφαρμογή όπως προαναφέραμε και παραπάνω μας εμφανίζει το λογότυπο της ομάδας μας στο οποίο πατώντας πάνω του μας οδηγεί στο κύριο application για το χειρισμό του οχήματος μας, χρησιμοποιώντας την παρακάτω εντολή στο block.



Για να επιτευχθεί ο χειρισμός του ρομπότ μας με την εφαρμογή θα πρέπει να γίνει σύζευξη του app με το ρομπότ. Για να επιτευχθεί κάτι τέτοιο υλοποιούμε το παρακάτω block προγραμματισμού.

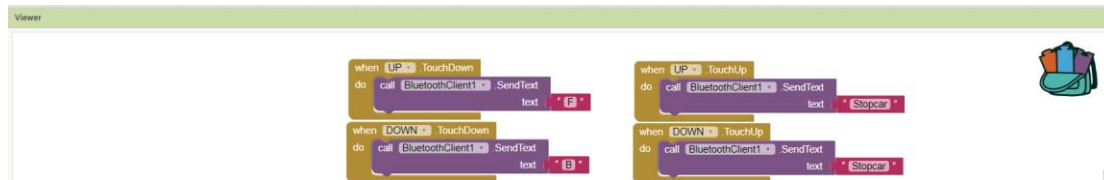


Πριν το πάτημα του κουμπιού του bluetooth η κατάσταση της επικοινωνίας βρίσκεται σε αδράνεια. Όταν πατηθεί καλείται το bluetooth Client 1 για να γίνει σύνδεση με μία διεύθυνση. Μετά από κάποιο χρόνο που θα περάσει και θα μετρήσει το Clock 1, εάν συνδεθεί το bluetooth τότε μπορούν

να καλεστούν οι επόμενες εντολές αν όμως δεν γίνει σύνδεση τότε ξανά καλείται το bluetooth client 1 για να ξανά γίνει προσπάθεια σύνδεσης.

Όσο αφορά την κίνηση του ρομπότ μας αυτή επιτυγχάνετε με τις εντολές Touch Up και Touch Down. Οι εντολές αυτές ενεργοποιούνται με το πάτημα του ανάλογου κουμπιού είτε κίνησης είτε κατεύθυνσης. Οπότε αυτό σημαίνει ότι για να γίνει οποιαδήποτε ενέργεια θα πρέπει το ανάλογο button να είναι μόνιμα πατημένο ενώ άμα το αφήσουμε τότε σταματάει να υπάρχει είτε κίνηση είτε ορισμός κατεύθυνσης. Η επιλογή των εντολών αυτών έγινε για τον καλύτερο χειρισμό και όσο γίνεται μεγαλύτερη ακρίβεια των κινήσεων μας.

Για την κίνηση μπροστά και πίσω του ρομπότ είναι αντιστοιχεί το παρακάτω block προγραμματισμού.

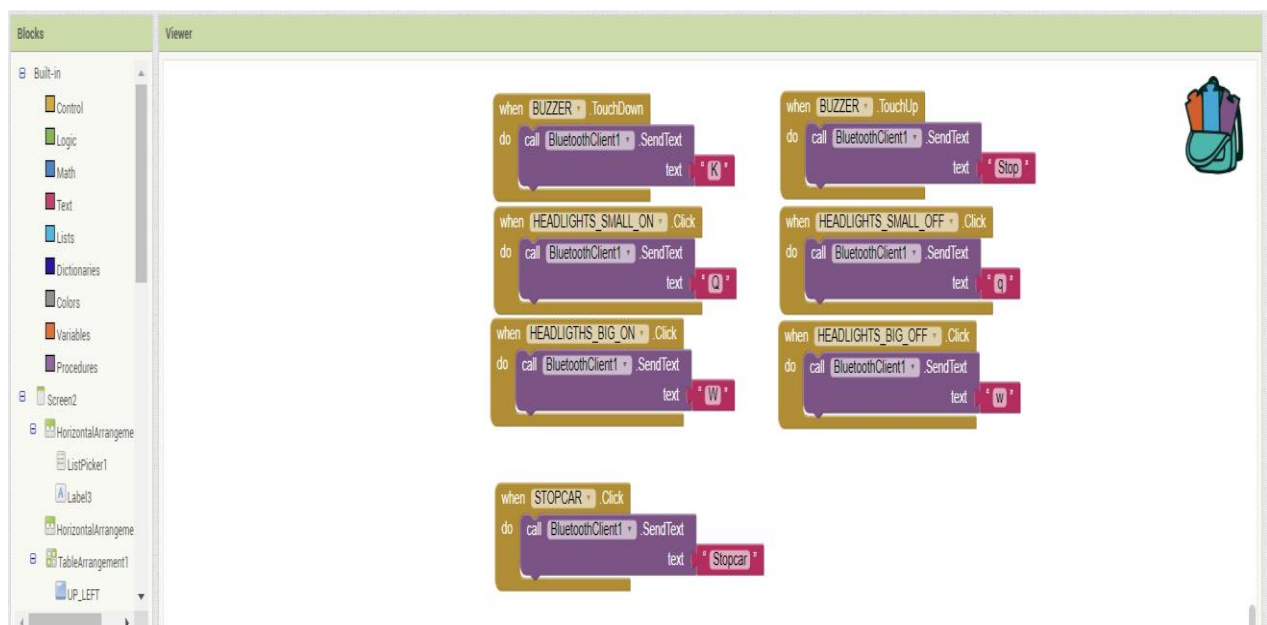


Ενώ για τον ορισμό κατεύθυνση και επι της ουσίας την κίνηση δεξιά και αριστερά των μπροστινών τροχών είναι οι επόμενες εντολές.



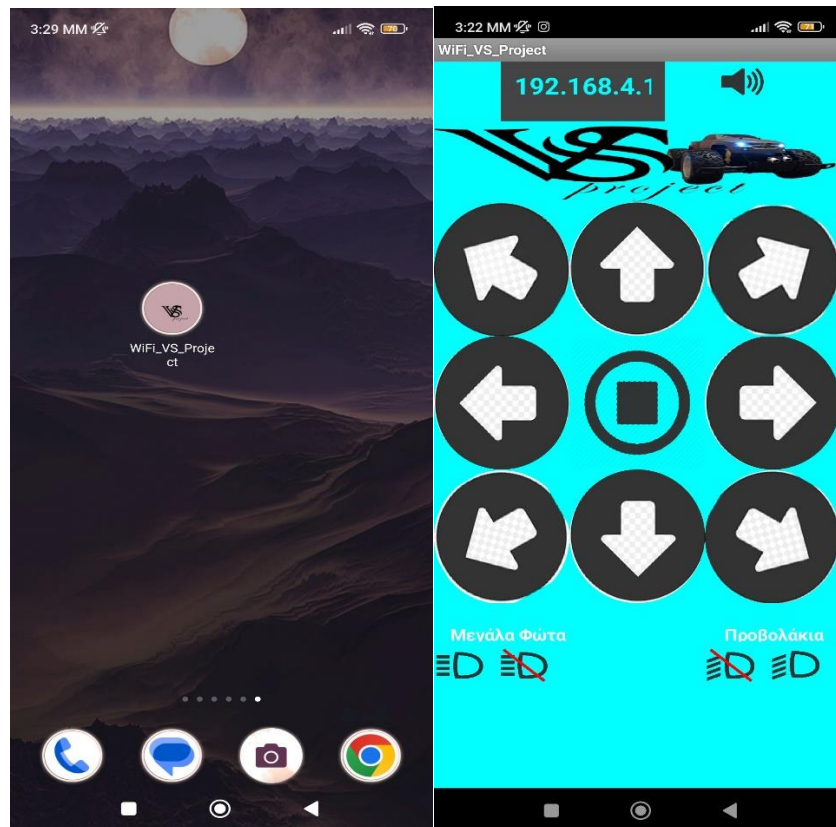
Η επόμενη περίπτωση είναι ο έλεγχος των λαμπτήρων και της 'κόρνας'. Για αυτή τη περίπτωση

χρησιμοποιούμε τις εντολές click δημιουργώντας διαφορετικό button κάθε φορά για την ενεργοποίηση και την απενεργοποίηση του κάθε αντικειμένου. Έτσι όταν πατάμε το button on ενεργοποιούνται οι καθορισμένες λάμπες και όταν αφήνουμε το χέρι από το button συνεχίζουν να είναι στην On κατάστασης και για να σβήσουν πατάμε το Off button. Μετά το πάτημα του κουμπιού παραμένουν σβηστά. Ακόμα και στο stop χρησιμοποιούμε click button, διότι θέλουμε την απενεργοποίηση όλων των εντολών με ένα άγγιγμα. Ενώ πάλι η κόρνα χρειάζεται να δουλεύει όσο κρατάμε το κουμπί και να απενεργοποιείται όταν το αφήνουμε. Το παρακάτω block μας δείχνει το προγραμματισμό που χρησιμοποιήσαμε για την υλοποίηση του.



3.3 Εφαρμογή για το χειρισμό του οχήματος μέσω WiFi

Όπως και στο Bluetooth apk έτσι και στο WiFi κατά την δημιουργία των εφαρμογών επιλέξαμε για αισθητικούς λόγους να δημιουργήσουμε το δικά μας γραφικά στοιχεία μέσω του Photoshop, καθώς και το δικό μας λογότυπο που εκπροσωπεί την ομάδα μας.



1. Η IP που εμφανίζεται είναι η IP του δικτύου μας.
2. Το button on για τα μεγάλα φώτα είναι για την ενεργοποίηση των μεγάλων φωτών μπροστά και για τα φώτα πορείας πίσω.
3. Το button off για τα μεγάλα φώτα είναι για την απενεργοποίηση των μεγάλων φωτών μπροστά και για τα φώτα πορείας πίσω.
4. Το button on για τα προβολάκια είναι για την ενεργοποίηση των μικρών φωτών μπροστά.
5. Το button off για τα προβολάκια είναι για την απενεργοποίηση των μικρών φωτών μπροστά.
6. Το button του ηχείου είναι για την ενεργοποίηση της κόρνας - buzzer
7. Το button Forward είναι για την κίνηση των τροχών (γκάζι) του ρομπότ.

8. Το button Backward είναι για το φρενάρισμα των τροχών.
9. Το button Go and Left είναι για την εμπρόσθια και αριστερή κίνηση του ρομπότ.
10. Το button Go and Right είναι για την εμπρόσθια και δεξιά κίνηση του ρομπότ.
11. Το button Left είναι για την αριστερόστροφη περιστροφή των εμπρόσθιων τροχών.
12. Το button Right είναι για την δεξιόστροφη περιστροφή των εμπρόσθιων τροχών.
13. Το button Back and Left είναι για την οπίσθια και αριστερή κίνηση του ρομπότ.
14. Το button Back and Right είναι για την οπίσθια και δεξιά κίνηση του ρομπότ.
15. Το Button Stop είναι για απενεργοποίηση όλων των ενεργειών του ρομπότ.

3.3.1 Ανάλυση Block Διαγράμματος WiFi εφαρμογής

Και στην περίπτωση του WiFi ο τρόπος χειρισμού και ελέγχου του ρομπότ μας γίνεται με παρόμοιο τρόπο με το Bluetooth. Έτσι ανοίγοντας την εφαρμογή όπως προαναφέραμε και στο Bluetooth ark μας εμφανίζει το λογότυπο της ομάδας μας στο οποίο πατώντας πάνω του μας οδηγεί στο κύριο application για το χειρισμό του οχήματος μας, χρησιμοποιώντας την παρακάτω εντολή στο block. Πριν την σύνδεση του κινητού με το WiFi του ρομπότ είναι σε μηδενικές καταστάσεις. Όταν συνδεθεί εμφανίζεται και η διεύθυνση IP.

Όσο αφορά την κίνηση του ρομπότ μας και σε αυτή την περίπτωση επιτυγχάνετε με τις εντολές Touch Up και Touch Down. Οι εντολές αυτές ενεργοποιούνται με το πάτημα του ανάλογου κουμπιού είτε κίνησης είτε κατεύθυνσης. Οπότε αυτό σημαίνει ότι για να γίνει οποιαδήποτε ενέργεια θα πρέπει το ανάλογο button να είναι μόνιμα πατημένο ενώ άμα το αφήσουμε τότε σταματάει να υπάρχει είτε κίνηση είτε ορισμός κατεύθυνσης. Η επιλογή των εντολών αυτών έγινε για τον καλύτερο χειρισμό και όσο γίνεται μεγαλύτερη ακρίβεια των κινήσεων μας.

Για την κίνηση μπροστά και πίσω του ρομπότ είναι αντιστοιχεί το παρακάτω block προγραμματισμού.



```
when forward TouchDown
do set Web1 Uri to http://
join IP_address Text
call Web1 Get

when right TouchDown
do set Web1 Uri to http://
join IP_address Text
call Web1 Get

when forward TouchUp
do set Web1 Uri to http://
join IP_address Text
call Web1 Get

when right TouchUp
do set Web1 Uri to http://
join IP_address Text
call Web1 Get
```

Ενώ για τον ορισμό κατεύθυνση χρησιμοποιούμε το παρακάτω block.

```
when left TouchDown
do set Web1 Uri to http://
join IP_address Text
call Web1 Get

when backward TouchDown
do set Web1 Uri to http://
join IP_address Text
call Web1 Get

when forward_right TouchDown
do set Web1 Uri to http://
join IP_address Text
call Web1 Get

when left TouchUp
do set Web1 Uri to http://
join IP_address Text
call Web1 Get

when backward TouchUp
do set Web1 Uri to http://
join IP_address Text
call Web1 Get

when forward_right TouchUp
do set Web1 Uri to http://
join IP_address Text
call Web1 Get

when right_backward TouchDown
do set Web1 Uri to http://
join IP_address Text
call Web1 Get

when right_backward TouchUp
do set Web1 Uri to http://
join IP_address Text
call Web1 Get

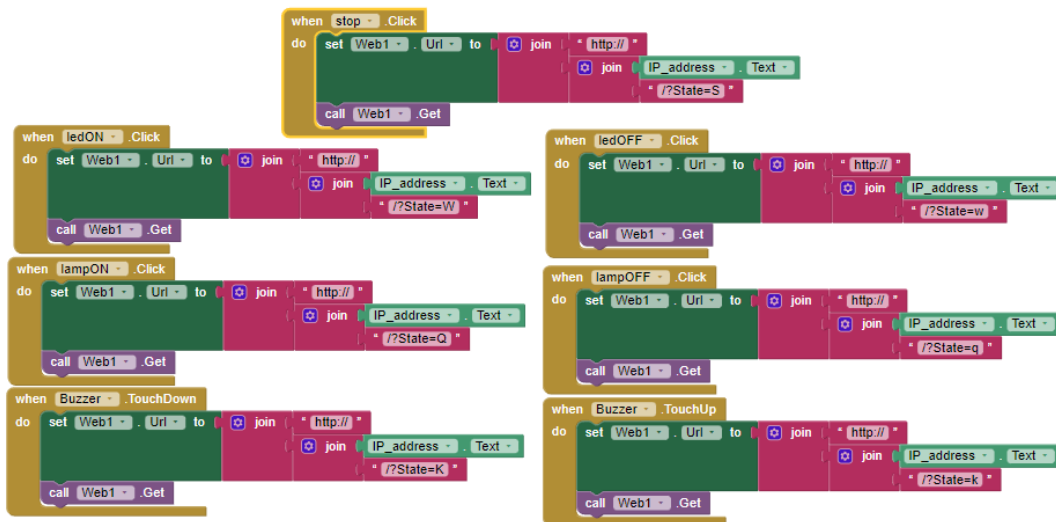
when left_forward TouchDown
do set Web1 Uri to http://
join IP_address Text
call Web1 Get

when left_forward TouchUp
do set Web1 Uri to http://
join IP_address Text
call Web1 Get

when left_backward TouchDown
do set Web1 Uri to http://
join IP_address Text
call Web1 Get

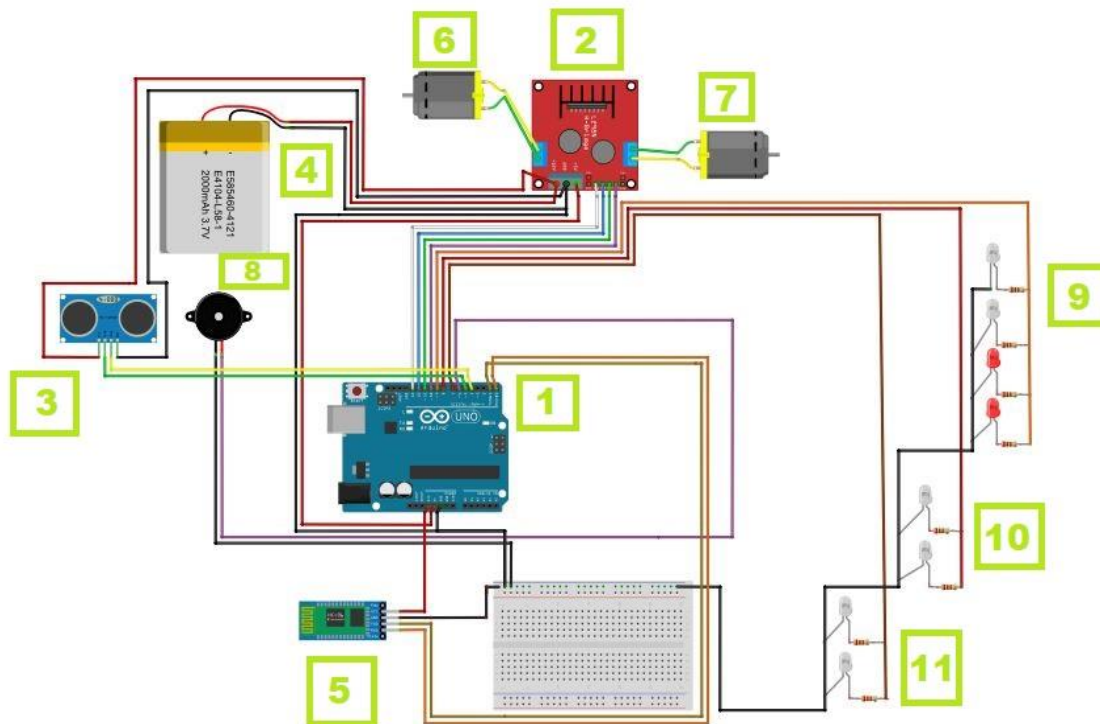
when left_backward TouchUp
do set Web1 Uri to http://
join IP_address Text
call Web1 Get
```

Η επόμενη περίπτωση είναι ο έλεγχος των λαμπτήρων και της ‘κόρνας’. Για αυτή τη περίπτωση χρησιμοποιούμε τις εντολές click δημιουργώντας διαφορετικό button κάθε φορά για την ενεργοποίηση και την απενεργοποίηση του κάθε αντικειμένου. Έτσι όταν πατάμε το button on ενεργοποιούνται οι καθορισμένες λάμπες και όταν αφήνουμε το χέρι από το button συνεχίζουν να είναι στην On κατάσταση και για να σβήσουν πατάμε το Off button. Μετά το πάτημα του κουμπιού παραμένουν σβηστά. Ακόμα και στο stop χρησιμοποιούμε click button, διότι θέλουμε την απενεργοποίηση όλων των εντολών με ένα άγγιγμα. Ενώ πάλι η κόρνα χρειάζεται να δουλεύει όσο κρατάμε το κουμπί και να απενεργοποιείται όταν το αφήνουμε. Το παρακάτω block μας δείχνει το προγραμματισμό που χρησιμοποιήσαμε για την υλοποίηση του.



Κεφάλαιο 4: Σχηματικά, ανάλυση και συνδεσμολογίες

4.1 Σχηματικό συνδεσμολογίας Bluetooth

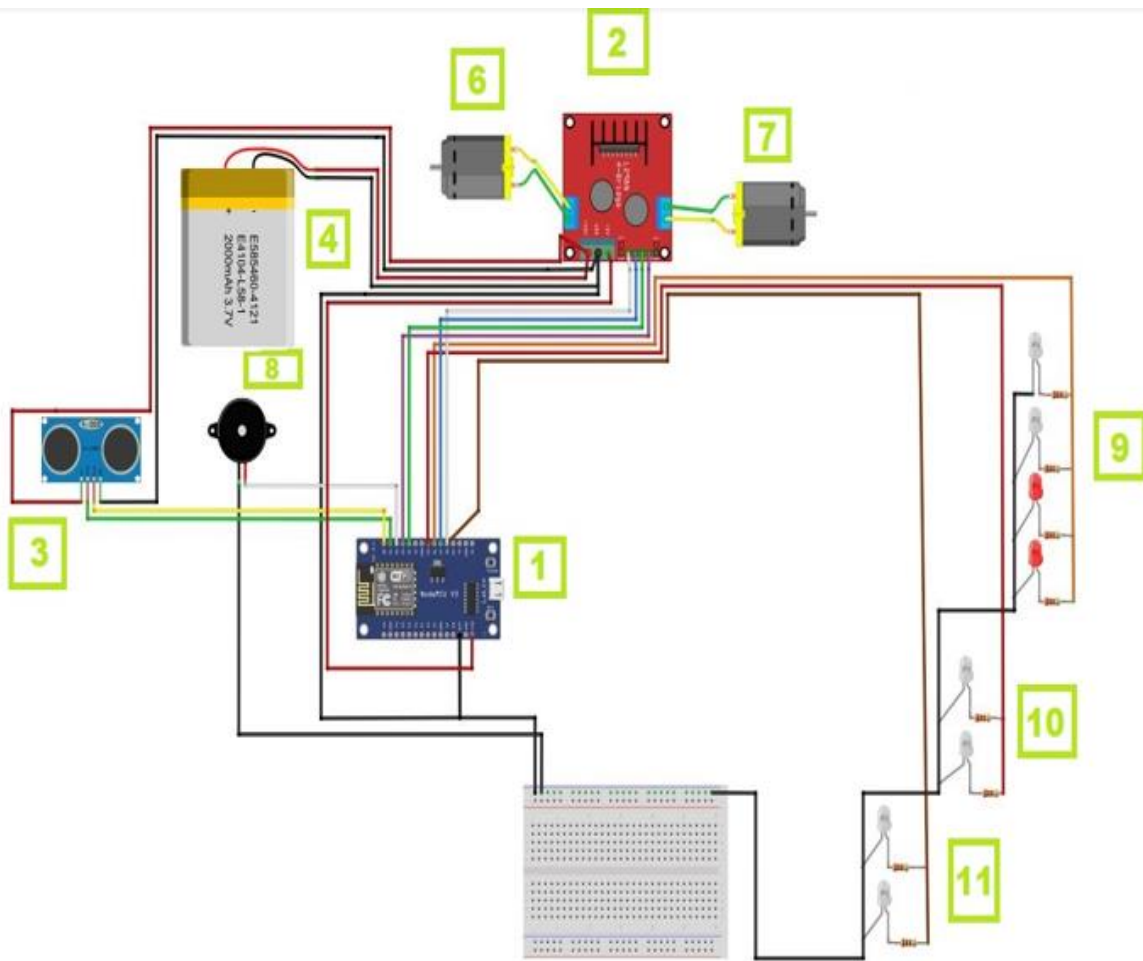


1. Arduino
2. L298N (Σε αυτό συνδέονται τα δύο μοτέρ, Το συν πλην της μπαταρίας αλλά και του arduino σε κοινούς κόμβους)
3. Ultrasonic Sensor (Παίρνει τάση και Com από την μπαταρία και συνδέεται σε δύο αναλογικές εξόδους του arduino για triggerPin και echoPin)
4. Επαναφορτιζόμενη Μπαταρία
5. Bluetooth Module (Παίρνει από το Arduino 3volt και COM και συνδέεται RX Bluetooth σε TX Arduino καθώς επίσης TX Bluetooth σε RX Arduino)
6. Κινητήρας κίνησης εμπρόσθιων τροχών τηλεκατευθυνόμενου
7. Κινητήρας οδήγησης εμπρόσθιων τροχών τηλεκατευθυνόμενου
8. Buzzer – Κόρνα (Αναλογικό PIN και COM)
9. Φώτα Πορείας (Αναλογικό PIN και COM)

10. Προβολάκια (Αναλογικό PIN και COM)

11. Φώτα Όπισθεν (Αναλογικό PIN και COM)

4.2 Σχηματικό συνδεσμολογίας WiFi



1. NodeMCU ESP8266
2. L298N (Σε αυτό συνδέονται τα δύο μοτέρ, Το συν πλην της μπαταρίας αλλά και του esp8266 σε κοινούς κόμβους)
3. Ultrasonic Sensor (Παίρνει τάση και Com από την μπαταρία και συνδέεται σε δύο αναλογικές εξόδους του esp8266 για triggerPin και echoPin)
4. Επαναφορτιζόμενη Μπαταρία
5. Κινητήρας κίνησης εμπρόσθιων τροχών τηλεκατευθυνόμενου
6. Κινητήρας οδήγησης εμπρόσθιων τροχών τηλεκατευθυνόμενου
7. Buzzer – Κόρνα (Αναλογικό PIN και COM)
8. Φώτα Πορείας (Αναλογικό PIN και COM)
9. Προβολάκια (Αναλογικό PIN και COM)
10. Φώτα Όπισθεν (Αναλογικό PIN και COM)
11. Φώτα Όπισθεν (Αναλογικό PIN και COM)

4.3 Συνδεσμολογία GSM – GPS MODULE SIM 808



Η συνδεσμολογία της είναι απλή, τοποθετούμε το gsm module πάνω στο arduino εφαρμόζοντας όλα τα pin στις αντίστοιχες υποδοχές του arduino. Τοποθετούμε και τις κεραίες gsm και gps στις θέσεις που αναγράφονται στην πλακέτα.

Κεφάλαιο 5: Ανάλυση του Κώδικα

Στο κεφάλαιο αυτό θα αναλύσουμε τους κώδικες τους οποίους γράψαμε και φορτώσαμε στους μικροελεγκτές για την λειτουργία του τηλεκατευθυνόμενου και του gps

5.1 Προγραμματισμός για χειρισμό μέσω Bluetooth

Σε αυτή την ενότητα θα δείξουμε τον κώδικα που δημιουργήσαμε στο περιβάλλον Arduino IDE ώστε να λειτουργήσει ορθά το όχημα μας, ενώ ταυτόχρονα θα αναλυθούν οι εντολές και η σύνταξή τους.

Αρχικοποιούμε τα pin πάνω στα οποία είναι συνδεδεμένος ο αισθητήρας υπερήχων, καθώς επίσης και την φωνητική εντολή. Οι αρχικοποιήσεις γίνονται με την εντολή “int” ενώ στη συνέχεια της σύνταξης μπορούμε να προσθέσουμε τις μεταβλητές και τη αρχική τους τιμή.

```
// ΑΡΧΙΚΟΠΟΙΗΣΕΙΣ
int triggerPin = 5;
int echoPin = 4;
int val;
int nb;
String voice;
```

Στη συνέχεια δηλώνουμε την λειτουργία των pin ανάλογα με την κατεύθυνση του σήματος τους (είσοδος – έξοδος). Με την εντολή “pinMode” συνοδευόμενη από τον αριθμό του pin που θέλουμε να προγραμματίσουμε, μας δίνει την δυνατότητα να θέσουμε την ιδιότητα που θέλουμε (output or input) στο συγκεκριμένο pin.

```

// ΟΡΙΣΜΟΣ PINS ΣΕ OUTPUT ΚΑΙ INPUT
void setup() {
  Serial.begin(9600);
  pinMode(13,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(11,OUTPUT);
  pinMode(10,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(8,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(1,OUTPUT);
  pinMode(0,INPUT);
  pinMode(triggerPin,OUTPUT);
  pinMode(echoPin,INPUT);
}

```

Σε αυτό το σημείο καλούνται οι συναρτήσεις που έχουμε δημιουργήσει παρακάτω. Η εντολή “void” χρησιμοποιείται για να δηλώσει την απουσία τυπικών παραμέτρων σε μια συνάρτηση αλλά και το ότι μια συνάρτηση δεν επιστρέφει τιμή.

```

// ΜΕ ΑΥΤΕΣ ΤΙΣ ΕΝΤΟΛΕΣ ΚΑΝΟΥΜΕ ΝΑ ΕΠΙΚΟΙΝΩΝΟΥΝ ΤΑ 3 VOID ΜΕΤΑΞΥ ΤΟΥΣ
void loop(){
  { bluetooth();
    sensor();}

    { speak();          //ΜΕ ΤΙΣ ΑΓΚΙΛΕΣ ΛΕΙΤΟΥΡΓΕΙ ΚΑΙ Η ΕΦΑΡΜΟΓΗ VOICE ΚΑΙ ΤΟ BLUETOOTH CONTROL APP
    sensor();}

}

```

Δημιουργούμε την συνάρτηση του Bluetooth που έχουμε καλέσει παραπάνω η οποία δηλώνει την λειτουργία του κάθε pin και κατ’ επέκταση την λειτουργία του οχήματος μας.

Οι λειτουργίες του οχήματος πραγματοποιούνται μέσω κάποιων εξόδων του Arduino μας. Στην ουσία θέτουμε “LOW” τις εξόδους που δεν θέλουμε να ενεργοποιούνται και “HIGH” την έξοδο που θα μας δώσει την συγκεκριμένη λειτουργία που ζητάμε μέσω του χειριστηρίου μας. Αυτό πραγματοποιείται με την εντολή “digitalWrite”.

Επίσης στον κώδικα φαίνεται η καθυστέρηση που έχουμε ορίσει ώστε να λειτουργεί η κόρνα με συγκεκριμένο τόνο. Με τον αριθμό που θέτουμε στην εντολή “tone” και στην εντολή delay επεμβαίνουμε στη συχνότητα του buzzer και κάνουμε τον ήχο που εκπέμπει “λεπτό” ή πιο “βαρύ”.

```

// ΜΕ ΤΙΣ ΕΝΤΟΛΕΣ ΑΥΤΕΣ ΛΕΙΤΟΥΡΓΕΙ ΜΕΣΩ ΧΡΕΙΣΙΕΤΗΡΙΟΥ
void bluetooth()
{
if(Serial.available()>0)
{
int data= Serial.read();
Stop();
if(data=='R')
{
digitalWrite(13,LOW);
digitalWrite(12,LOW);
digitalWrite(11,HIGH);
digitalWrite(10,LOW);
}
else if(data=='L')
{
digitalWrite(13,LOW);
digitalWrite(12,LOW);
digitalWrite(11,LOW);
digitalWrite(10,HIGH);
}
else if(data=='F')
{
digitalWrite(13,LOW);
digitalWrite(12,HIGH);
digitalWrite(11,LOW);
digitalWrite(10,LOW);
}
else if(data=='B')
{
digitalWrite(13,HIGH);
digitalWrite(12,LOW);
digitalWrite(11,LOW);
digitalWrite(10,LOW);
digitalWrite(7,HIGH);
}
}
}
}

```

```
tone(6,450);
delay(500);
noTone(6);
delay(500);
}
else if (data=='G')
{
digitalWrite(13, LOW);
digitalWrite(12, HIGH);
digitalWrite(11, LOW);
digitalWrite(10, HIGH);
}
else if (data=='I')
{
digitalWrite(13, LOW);
digitalWrite(12, HIGH);
digitalWrite(11, HIGH);
digitalWrite(10, LOW);
}
else if (data=='H')
{
digitalWrite(13, HIGH);
digitalWrite(12, LOW);
digitalWrite(11, LOW);
digitalWrite(10, HIGH);
digitalWrite(7,HIGH);

tone(6,450);
delay(500);
noTone(6);
delay(500);
}
else if (data == 'J')
{
digitalWrite(13, HIGH);
digitalWrite(12, LOW);
```

```

digitalWrite(10, LOW);
digitalWrite(7,HIGH);

tone(6,450);
delay(500);
noTone(6);
delay(500);
}
else if (data== 'W')
{
digitalWrite(9, HIGH);
}

else if (data=='w')
{
digitalWrite(9, LOW);
}

else if (data== 'Q')
{
digitalWrite(8, HIGH);
}
else if (data=='q')
{
digitalWrite(8, LOW);
}
else if(data=='K')
{
digitalWrite(6,HIGH);
}
}
}

void Stop()
{
digitalWrite(13,LOW);

digitalWrite(12,LOW);
digitalWrite(11,LOW);
digitalWrite(10,LOW);
digitalWrite(7,LOW);
digitalWrite(6,LOW);
}
void Stopcar()
{
digitalWrite(13,LOW);
digitalWrite(12,LOW);
digitalWrite(11,LOW);
digitalWrite(10,LOW);
}
}

```

Στην συνέχεια κωδικοποιούμε την συνάρτηση του σένσορα υπερήχων που έχουμε καλέσει στην αρχή, ακόμα δηλώνουμε τις κατάλληλες αποστάσεις για την λειτουργία αποφυγής σύγκρουσης. Στην αρχή του προγραμματισμού μας είχαμε δηλώσει μικρή απόσταση και ο σένσορας μας έδινε εντολή φρεναρίσματος, όμως η απόκριση του κινητήρα ήταν καθυστερημένη με αποτέλεσμα την σύγκρουση. Με αρκετές προσπάθειες βρήκαμε τα ιδανικά εκατοστά που έπρεπε να θέσουμε στον αισθητήρα ώστε να ανταποκρίνεται σωστά στο έργο μας.

```
// ΕΝΤΟΛΕΣ ΓΙΑ ΤΟΝ ΣΕΝΣΟΡΑ ΑΠΟΣΤΑΣΗΣ
void sensor(){
  int duration, distance; //ΔΗΛΩΣΗ ΑΠΟΣΤΑΣΗΣ
  digitalWrite(triggerPin, HIGH);
  delay(10);
  digitalWrite(triggerPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1; //ΜΕ ΑΥΤΗ ΤΗΝ ΕΝΤΟΛΗ ΤΗΝ ΧΡΗΣΙΜΟΠΟΙΟΥΜΕ ΓΙΑ ΜΕΤΑΤΡΟΠΗ ΣΕ ΙΝΤΣΕΣ
  Serial.print(distance);
  Serial.print("cm");
  Serial.println(" ");

// ΜΕ ΑΥΤΕΣ ΤΙΣ ΕΝΤΟΛΕΣ ΦΤΕΙΑΧΝΟΥΜΕ ΤΗΝ ΑΠΟΣΤΑΣΗ ΠΟΥ ΘΕΛΟΥΜΕ ΝΑ ΣΤΑΜΑΤΑΕΙ ΤΟ ΑΥΤΟΚΙΝΗΤΑΚΙ
if(distance <= 55)
{
  digitalWrite(13,HIGH);
  digitalWrite(12,LOW);
  digitalWrite(11,LOW);
  digitalWrite(10,LOW);
  digitalWrite(7,HIGH);

  Serial.println("ALERT");
  delay(250);
  digitalWrite(13,LOW);
  digitalWrite(12,LOW);
  digitalWrite(11,LOW);
  digitalWrite(10,LOW);
  digitalWrite(7,LOW);
  delay(2000);
}
}
```


Τέλος ορίζουμε την συνάρτηση του voice control μέσα από την οποία το όχημα μας υπακούει στις φωνητικές μας εντολές.

```
//ΕΝΤΟΛΕΣ ΓΙΑ ΤΟ VOICE CONTROL
void speak ()
{
  while (Serial.available()){
    delay(10);
    char c = Serial.read();
    if (c == '#') {break;}
    voice += c;
  }
  if (voice.length() > 0){

    if(voice == "*forward")
    {
      digitalWrite(13,LOW);
      digitalWrite(12,HIGH);
      digitalWrite(11,LOW);
      digitalWrite(10,LOW);
      delay (1000);
    }
    else if(voice == "*backward")
    {
      digitalWrite(13,HIGH);
      digitalWrite(12,LOW);
      digitalWrite(11,LOW);
      digitalWrite(10,LOW);
      digitalWrite(7,HIGH);|
    }
  }
}
```

```

    delay (1000);
  }
  else if(voice == "*right")
  {
    digitalWrite(13,LOW);
    digitalWrite(12,LOW);
    digitalWrite(11,HIGH);
    digitalWrite(10,LOW);
    delay(800);
  }
  else if(voice == "*left")
  {
    digitalWrite(13,LOW);
    digitalWrite(12,LOW);
    digitalWrite(11,LOW);

    digitalWrite(10,HIGH);
    delay(800);
  }
  else if(voice == "*light on")
  {
    digitalWrite(9,HIGH);
  }
  else if(voice == "*light off")
  {
    digitalWrite(9,LOW);
  }
  else if(voice == "*lamp on")
  {
    digitalWrite(8,HIGH);
  }
  else if(voice == "*lamp off")
  {
    digitalWrite(8,LOW);
  }
  else if(voice == "*stop")
  {
    digitalWrite(10,LOW );
    digitalWrite(11,LOW);
    digitalWrite(12,LOW );
    digitalWrite(13,LOW);
    digitalWrite(7,LOW);
  }

  voice=""; //ΕΑΝ ΔΕΝ ΔΟΘΕΙ ΤΙΠΟΤΑ ΤΟΤΕ ΕΠΑΝΑΛΑΒΕΤΑΙ ΤΗ ΔΕΞΗ
}
}

```

5.2 Προγραμματισμός για χειρισμό μέσω WiFi

Αρχικοποιούμε τα pin πάνω στα οποία είναι συνδεδεμένος ο αισθητήρας υπερήχων, καθώς και τις βιβλιοθήκες. Επίσης δημιουργούμε τοπικό δίκτυο WiFi με ονομασία VS. Αυτή η διαδικασία πραγματοποιείται με την εντολή “define” και στη συνέχεια το όνομα της εξόδου-εισόδου που θέλουμε να αρχικοποιήσουμε. Φαίνεται επίσης και η εντολή “include”, με τη οποία επιτυγχάνουμε την ανάκληση βιβλιοθηκών που έχουμε κατεβάσει και στη συνέχεια τοποθετήσει στην εφαρμογή στην οποία γράφουμε τον κώδικα μας. Οι βιβλιοθήκες αυτές είναι απαραίτητες για την κίνηση με WiFi.

```
#define A1      15      // L298N in1 Πίσω Κίνηση Μοτέρ          GPIO15 (D8)
// Μαζί με το A1 Ενεργοποιούνται τα φώτα τις όπισθεν
#define A2      13      // L298N in2 Μπροστά Κίνηση Μοτέρ          GPIO13 (D7)
#define B1       2      // L298N in3 Αριστερή Κίνηση              GPIO2 (D4)
#define B2       0      // L298N in4 Δεξιά Κίνηση                 GPIO0 (D3)
#define triggerPin  5      // ultrasonic sensor Trigger              GPIO5 (D1)
#define echoPin    16     // ultrasonic sensor Echo                 GPIO16 (D0)
#define led        14     // Φώτα μπροστά μεγάλα και πίσω         GPIO14 (D5)
#define lamp       12     // Φώτα μικρά                             GPIO12 (D6)
#define buzzer     4      // Κόρνα                                   GPIO4 (D2)

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

String command;

const char* ssid = "vs" ;
ESP8266WebServer server(80);
```

Έπειτα δηλώνουμε την λειτουργία των pin ανάλογα με την κατεύθυνση του σήματος τους (είσοδος – έξοδος).

```

void setup() {

  pinMode (A1, OUTPUT);
  pinMode (A2, OUTPUT);
  pinMode (B1, OUTPUT);
  pinMode (B2, OUTPUT);
  pinMode (led,OUTPUT);
  pinMode (lamp,OUTPUT);
  pinMode (buzzer,OUTPUT);
  pinMode (triggerPin, OUTPUT);
  pinMode (echoPin, INPUT);

  Serial.begin(9600);

```

Οι συναρτήσεις που δημιουργήσαμε για την λειτουργία WiFi φαίνονται παρά κάτω και είναι για κάθε κίνηση ξεχώριστες.

```

// Σύνδεση WiFi

WiFi.mode(WIFI_AP);
WiFi.softAP(ssid);

IPAddress myIP = WiFi.softAPIP();
Serial.print("AP IP address: ");
Serial.println(myIP);

// Starting WEB-server
server.on ( "/", HTTP_handleRoot );
server.onNotFound ( HTTP_handleRoot );
server.begin();
}

```

Δημιουργούμε συναρτήσεις για κάθε λειτουργία του WiFi ξεχωριστά.

```
void forward() {  
  
    digitalWrite(A1, LOW);  
    digitalWrite(A2, HIGH);  
  
    digitalWrite(B1, LOW);  
    digitalWrite(B2, LOW);  
  
}  
  
void backward() {  
  
    digitalWrite(A1, HIGH);  
    digitalWrite(A2, LOW);  
  
    digitalWrite(B1, LOW);  
    digitalWrite(B2, LOW);  
  
}
```

```
tone(buzzer,450); //Ενεργοποιείται η κόρνα για να ακούγεται πως το όχημα κάνει όπισθεν
delay(500);
noTone(buzzer);
delay(500);

}

void Right(){

digitalWrite(A1, LOW);
digitalWrite(A2, LOW);

digitalWrite(B1, HIGH);
digitalWrite(B2, LOW);

}

void Left(){

digitalWrite(A1, LOW);
digitalWrite(A2, LOW);

digitalWrite(B1, LOW);
digitalWrite(B2, HIGH);

}

void goRight(){

digitalWrite(A1, LOW);
digitalWrite(A2, HIGH);
```

```
        digitalWrite (B1, HIGH);
        digitalWrite (B2, LOW);

    }

void goLeft() {

    digitalWrite (A1, LOW);
    digitalWrite (A2, HIGH);

    digitalWrite (B1, LOW);
    digitalWrite (B2, HIGH);

}

void BackRight() {

    digitalWrite (A1, HIGH);
    digitalWrite (A2, LOW);

    digitalWrite (B1, HIGH);
    digitalWrite (B2, LOW);

    tone (buzzer, 450);
    delay (500);
    noTone (buzzer);
    delay (500);

}
```

```
void BackLeft() {  
  
    digitalWrite(A1, HIGH);  
    digitalWrite(A2, LOW);  
  
    digitalWrite(B1, LOW);  
    digitalWrite(B2, HIGH);  
  
    tone(buzzer, 450);  
    delay(500);  
    noTone(buzzer);  
    delay(500);  
  
}  
  
void stopCar() {  
  
    digitalWrite(A1, LOW);  
    digitalWrite(A2, LOW);  
  
    digitalWrite(B1, LOW);  
    digitalWrite(B2, LOW);  
  
}  
void ledON() {  
    digitalWrite(led, HIGH);  
}  
  
void ledOFF() {  
    digitalWrite(led, LOW);  
}
```



```

void lampON() {
digitalWrite(lamp,HIGH);
}

void lampOFF() {
digitalWrite(lamp,LOW);
}

void buzzerON() {
tone(buzzer,500);
}

void buzzerOFF() {
digitalWrite(buzzer,LOW);
}

```

Στην συνέχεια εκτελούμε την συνάρτηση του σένσορα υπερήχων που έχουμε καλέσει στην αρχικοποίηση, ακόμα δηλώνουμε τις κατάλληλες αποστάσεις για την λειτουργία αποφυγής σύγκρουσης.

```

void loop () {
  controller();

  sensor();
}

void controller() {
  server.handleClient();

  command = server.arg("State");
  if (command == "F") forward();
  else if (command == "B") backward();
  else if (command == "L") Left();
  else if (command == "R") Right();
  else if (command == "I") goRight();
  else if (command == "G") goLeft();
  else if (command == "J") BackRight();
  else if (command == "H") BackLeft();
}

```

```

        else if (command == "S") stopCar();
        else if (command == "W") ledON();
        else if (command == "w") ledOFF();
        else if (command == "Q") lampON();
        else if (command == "q") lampOFF();
        else if (command == "K") buzzerON();
        else if (command == "k") buzzerOFF();

    }

void HTTP_handleRoot(void) {

if( server.hasArg("State") ){
    Serial.println(server.arg("State"));
}
server.send ( 200, "text/html", "" );
delay(1);
}

void sensor(){
    int duration, distance;

    digitalWrite(triggerPin, HIGH);
    delay(10);
    digitalWrite(triggerPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance = (duration/2) / 29.1; //ΜΕ ΑΥΤΗ ΤΗΝ ΕΝΤΟΛΗ ΤΗΝ ΧΡΗΣΙΜΟΠΟΙΟΥΜΕ ΓΙΑ ΜΕΤΑΤΡΟΠΗ ΣΕ ΙΝΤΣΕΣ
    Serial.print(distance);

    Serial.print("cm");
    Serial.println(" ");

// ΜΕ ΑΥΤΕΣ ΤΙΣ ΕΝΤΟΛΕΣ ΦΤΕΙΑΧΝΟΥΜΕ ΤΗΝ ΑΠΟΣΤΑΣΗ ΠΟΥ ΘΕΛΟΥΜΕ ΝΑ ΣΤΑΜΑΤΑΕΙ ΤΟ ΑΥΤΟΚΙΝΗΤΑΚΙ
if (distance <= 40)
{
    backward ();
    delay(250);

    stopCar();
    delay(2000);
}
}
}

```

5.3 Προγραμματισμός για το GSM – GPS module

Οι απαραίτητοι υπολογισμοί για τον καθορισμό της θέσης αλλά και διάφορων άλλων χρήσιμων τιμών θα γίνουν αποκλειστικά από την βιβλιοθήκη `tinygps` σε συνδυασμό με την `Software Serial`.

Αρχικά θα συμπεριληφθεί η βιβλιοθήκη και θα οριστεί με το όνομα «`gps`».

```
#include <TinyGPS.h>
```

```
TinyGPS gps;
```

Όπως και η βιβλιοθήκη για την σειριακή επικοινωνία αλλά και θα οριστούν οι ακίδες στις οποίες θα στέλνει και θα δέχεται δεδομένα.

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

```
SoftwareSerial ss(2, 3);
```

Θα γίνει δήλωση των μεταβλητών που είναι απαραίτητες για τον υπολογισμό των δεδομένων που χρειάζονται (χρόνου απόκρισης, χρόνου, ημερομηνίας, ταχύτητας και πορείας). Επίσης δηλώνεται η συνάρτηση `printFloat` που θα αναλυθεί στη συνέχεια.

```
Unsigned long fix_age, time, date, speed, course;
```

```
Void printFloat(double f, int digits = 2);
```

Θα εκκινήσει η σειριακή με ταχύτητα 9600 bps.

```
ss.begin(9600);
```

Πλέον μέσα στην συνάρτηση της επανάληψης θα δημιουργηθεί συνθήκη που θα ελέγχει αν η πλακέτα δέχεται σωστού τύπου δεδομένα. Ξεκινά η επικοινωνία με την πλακέτα `sim808` με τις `AT-Commands` που έχουν αναλυθεί σε προηγούμενη ενότητα όπως και η συνάρτηση `millis`. Τέλος

όταν μεταδοθούν σωστά δεδομένα η μεταβλητή newData θα γίνει αληθής και θα αρχίσουν οι υπολογισμοί.

```
bool newData = false;
ss.println("AT");
ss.println("AT+CGNSPWR=1");
ss.println("AT+CGNSTST=1");

for (unsigned long start = millis(); millis() - start < 1000;)
{
    while (ss.available())
    {
        char c = ss.read();

        //Serial.write(c); //gps nmea string
        if (gps.encode(c))
            newData = true;
    }
}
```

Για να παρουσιάζονται όλες οι τιμές στην σειριακή ανεξαρτήτως αν είναι σωστές ή όχι θα χρειαστεί να αφαιρεθούν οι δύο «//» που ορίζουν την εντολή σαν σχόλιο.

Εφόσον η πλακέτα δεχθεί σωστά δεδομένα ορίζονται οι μεταβλητές στις οποίες θα αποθηκευτούν οι τιμές των υπολογισμών.

```
if (newData)
{
```

```
float latitude, longitude;  
unsigned long age;  
  
float fkmph;  
int year;  
  
byte month, day, hour, minute, second, hundredths;
```

Στις επόμενες γραμμές υπολογίζονται οι τιμές που εισέρχονται από την κεραία του gps. Αυτές είναι το γεωγραφικό πλάτος και μήκος, η ταχύτητα, οι δορυφόροι, το υψόμετρο, η ώρα και η ημερομηνία. Όλες τυπώνονται στην σειριακή του ide για ευκολότερη απεικόνιση.

```
gps.f_speed_kmph();

gps.f_get_position(&latitude, &longitude, &age);

gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths, &age);

Serial.print("LAT=");

Serial.print(latitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : latitude,

Serial.print(" LON=");

Serial.print(longtitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 :
```

```

longitude, 6);

    Serial.print(" SAT=");

    Serial.print(gps.satellites() == TinyGPS::GPS_INVALID_SATELLITES ? 0 :
gps.satellites());

    Serial.print("Date: "); Serial.print(static_cast<int>(month));

Serial.print("/");

    Serial.print(static_cast<int>(day)); Serial.print("/");
Serial.print(year);

    Serial.print("  Time: "); Serial.print(static_cast<int>(hour+3));
Serial.print(":");

    Serial.print(static_cast<int>(minute)); Serial.print(":");
Serial.print(static_cast<int>(second));

    Serial.print("."); Serial.print(static_cast<int>(hundredths));
Serial.print("Alt: "); Serial.print(gps.f_altitude());
Serial.print(" (kmph): "); printFloat(gps.f_speed_kmph());
Serial.println();

String Text1 = "LAT=";

String Text2 = "LON=";

String Text3 = "-";

```

Τέλος για την σωστή απεικόνιση των τιμών θα χρησιμοποιηθεί μία συνάρτηση που θα επεξεργάζεται τους αρνητικούς αριθμούς.

```

void printFloat(double number, int digits)

{

    if (number < 0.0) {
        Serial.write('-');
        number = -number;
    }
}

```

////////Διαχείριση αρνητικών αριθμών //////

```
double rounding = 0.5;

for (uint8_t i=0; i<digits; ++i)
    rounding /= 10.0;

number += rounding;
```

////////Σωστή Αποκόνιση Στρογγυλοποίησης////////

```
unsigned long int_part = (unsigned long)number;
double remainder = number - (double)int_part;
Serial.print(int_part);
```

////////Τύπωσε το Ακέραιο Κομμάτι////////

```
if (digits > 0)

    Serial.print(".");
```

////////Τύπωσε τα ψηφία ένα τη φορά////////

```
while (digits-- > 0) {
    remainder *= 10.0;

    int toPrint = int(remainder);
    Serial.print(toPrint);
    remainder -= toPrint;
```


Κεφάλαιο 6: Συμπεράσματα και εξέλιξη της εργασίας μελλοντικά

6.1 Συμπεράσματα

Η διπλωματική εργασία που παρουσιάστηκε είχε σκοπό την υλοποίηση μιας μικρογραφίας ενός αυτοκινήτου που να μπορεί να εκτελεί τις λειτουργίες ενός πρότυπου συμβατικού οχήματος. Το όχημα αυτό έχει σχεδιαστεί με τέτοιο τρόπο έτσι ώστε να μπορεί ο οδηγός να έχει ιδιαίτερους χειρισμούς που θα του κάνουν πολύ πιο εύκολη την καθημερινότητα του.

Αρχικά η ιδέα για αυτή τη διπλωματική εργασία προέκυψε από το μεγαλύτερο πρόβλημα που αντιμετωπίζει ο Έλληνας οδηγός στους δρόμους της χώρας μας που είναι το παρκάρισμα. Εμείς σκεφτήκαμε να κάνουμε το αυτοκίνητο να έχει τηλεχειρισμό για να μπορεί ο οδηγός να σταθμεύει και στο πιο δύσκολο σημείο με κάθε ασφάλεια. Επίσης θέλαμε να προσφέρουμε στον οδηγό τη μέγιστη ασφάλεια και έτσι δημιουργήσαμε συστήματα αυτόματης πέδηση και αποφύγει εμποδίων. Τέλος έχουμε υλοποιήσει και ένα σύστημα tracking με το οποίο θα μπορεί ο οδηγός να παρκάρει και να ξέρει κάθε στιγμή που είναι το όχημα του χωρίς να ψάχνει να το βρεί, το σύστημα αυτό βοηθάει πολύ όταν κάποιος παρκάρει σε μέρη άγνωστα σε αυτόν ή σε μεγάλους χώρους στάθμευσης με πολλά οχήματα.

Τέλος πρέπει να επισημάνουμε ότι και η διπλωματική εργασία αυτή αντιμετώπισε πολλά προβλήματα και δυσκολίες κατά την υλοποίηση της όπως επίσης και έχει και πολλά περιθώρια βελτίωσης.

6.2 Εξέλιξη της εργασίας μελλοντικά

Η διπλωματική εργασία αυτή όπως προαναφέραμε έχει πολλά περιθώρια βελτίωσης σε διάφορα επίπεδα, είτε όσο αφορά το χειρισμό, είτε όσο αφορά το gps tracking είτε ακόμα και την ασφάλεια.

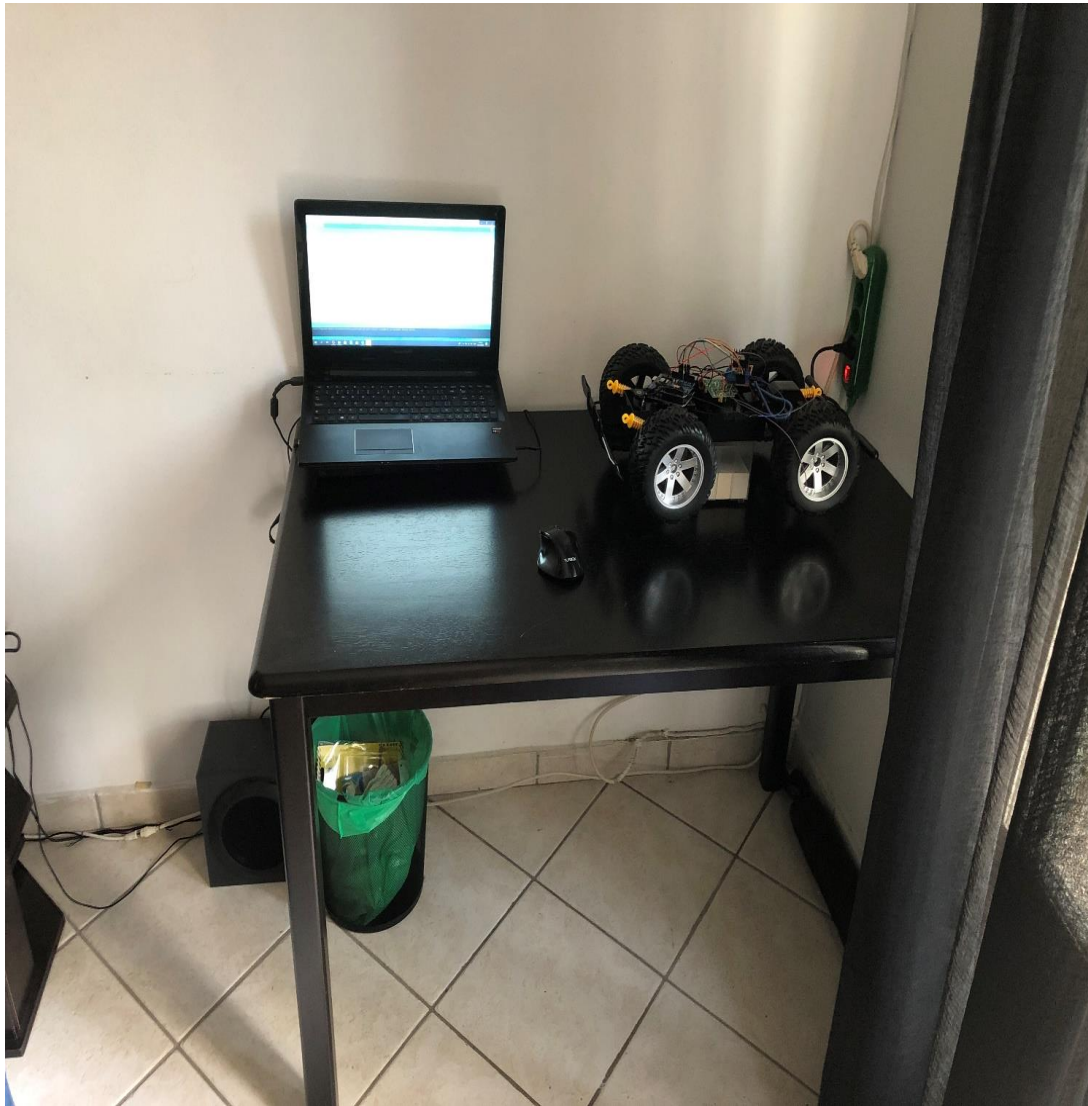
Αρχικά όσο αφορά το χειρισμό μπορεί μελλοντικά να υλοποιηθεί ένα σύστημα τηλεματικής που να γίνεται ο έλεγχος και η παρακολούθηση του οχήματος από εκεί. Επίσης έχει πολλά περιθώρια βελτίωσης και τη τροφοδοσία του οχήματος έτσι ώστε να μπορεί να χρησιμοποιείται για περισσότερη ώρα και να διανύει περισσότερα χιλιόμετρα. Όσο αφορά το GPS tracking μπορεί να εξελιχθεί πολύ περισσότερο φτιάχνοντας μια βάση δεδομένων όπου το GPS να στέλνει συνεχώς δεδομένα και να τα αποθηκεύει στη βάση και να μπορεί να μας δείχνει συνεχώς που βρίσκεται το όχημα και όχι να παίρνει μια φορά το στίγμα και αν κάποιος κλέψει ή για κάποιο λόγο το όχημα μετακινηθεί να μπορεί να ξέρει ο οδηγός που βρίσκεται.

Κατάλογος φωτογραφικού υλικού

Εικόνα 1.0



Εικόνα 1.1



Εικόνα 1.2



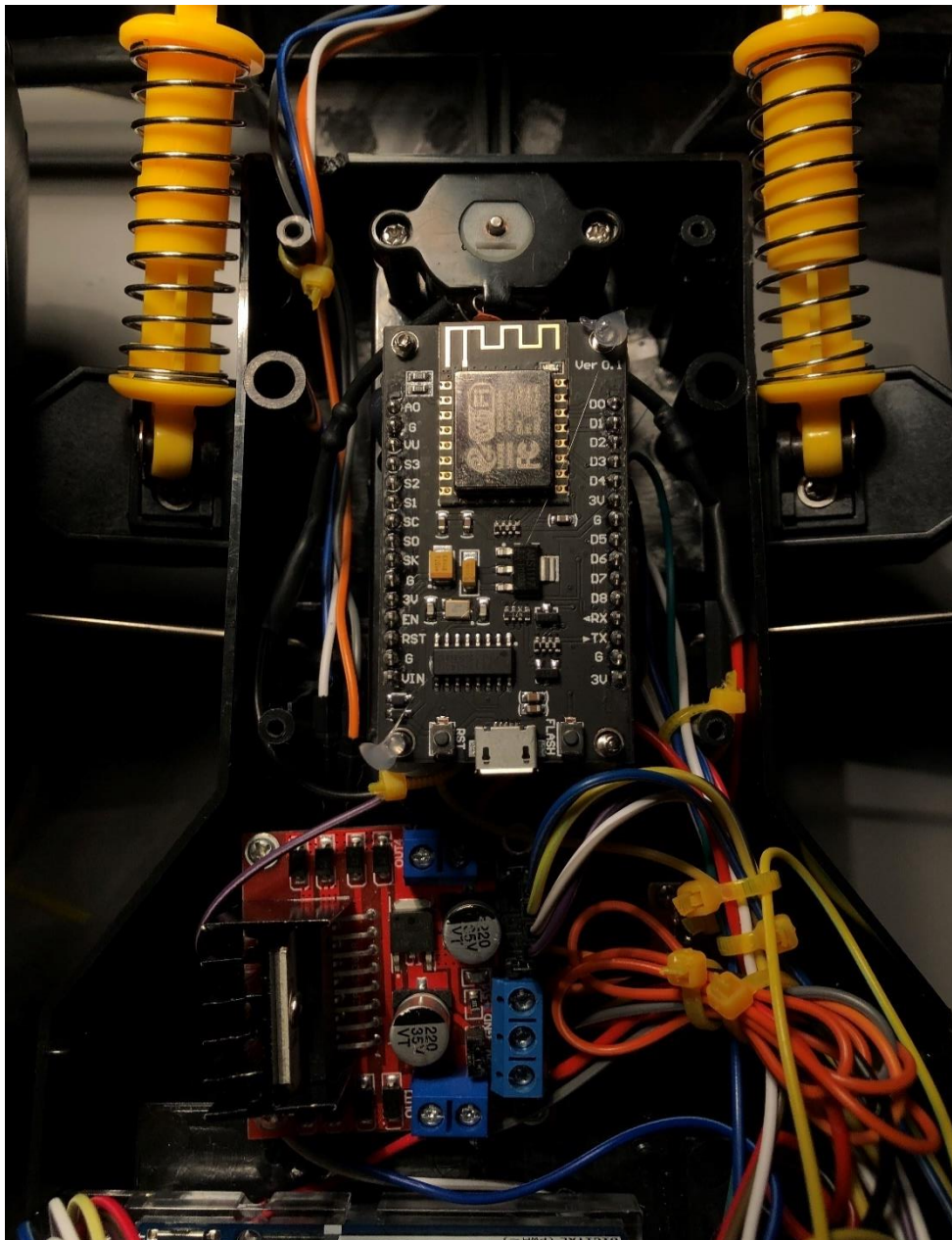
Εικόνα 1.3



Εικόνα 1.4



Εικόνα 1.5



Εικόνα 1.6



Εικόνα 1.7



Εικόνα 1.8



Εικόνα 1.10



Εικόνα 1.11



ΒΙΒΛΙΟΓΡΑΦΙΑ

Διαδίκτυο

<https://create.arduino.cc/projecthub/samanfern/bluetooth-controlled-car-d5d9ca>

<https://www.ardumotive.com/bluetooth-rc-car.html>

<https://www.instructables.com/Arduino-Bluetooth-RC-Car-Android-Controlled/>

https://create.arduino.cc/projecthub/mindhe_aniket/bluetooth-controlled-car-e696f3

<https://www.hackster.io/alankrantas/simple-nodemcu-wifi-controlled-car-esp8266-c5491e>

<https://www.instructables.com/WiFi-Car-Using-NodeMCU/>

<https://www.instructables.com/NodeMCU-ESP8266-WiFi-Robot-Car-Controlled-by-Appli/>

<https://github.com/lacour-vincent/wifi-car-esp8266>

<https://create.arduino.cc/projecthub/projects/tags/wifi>

<https://create.arduino.cc/projecthub/andriy-baranov/from-bt-to-wifi-creating-wifi-controlled-arduino-robot-car-09b7c1>

<https://www.instructables.com/Android-controlled-car-for-beginners-MIT-app-inven/>

<https://create.arduino.cc/projecthub/embeddedlab786/arduino-robot-car-control-using-hc-05-bluetooth-f13ae1>

<https://appinventor.mit.edu/>

[https://www.waveshare.com/wiki/GSM/GPRS/GPS_Shield_\(B\)](https://www.waveshare.com/wiki/GSM/GPRS/GPS_Shield_(B))

<https://www.e-wireless.gr/sim808-module-gsm-gprs-gps-for-arduino-with-gps-gsm-antenna/>

<https://grobotronics.com/waveshare-gsm-gprs-gps-shield-b.html>

<https://www.dfrobot.com/product-1469.html>

<https://simcom.ee/modules/gsm-gprs-gnss/sim808/>

<https://www.robotshop.com/media/files/pdf/sim808-gps-module-datasheet.pdf>

<https://wikifactory.com/@makerfabs/sim808-gps-tracker>

<https://el.wikipedia.org/wiki/%CE%9C%CE%B9%CE%BA%CF%81%CE%BF%CE%B5%CE%B%CE%B5%CE%B3%CE%BA%CF%84%CE%AE%CF%82>

<https://www.ece.uop.gr/diplomatiki-ergasia>

<https://grobotronics.com/arduino-uno-rev3.html?sl=el>

<https://arduinobots.wordpress.com/2015/03/09/προγραμματισμός-αισθητήρα-υπερήχων/>

<http://repository.library.teimes.gr/xmlui/bitstream/handle/123456789/10717/ΔΙΠΛΩΜΑΤΙΚΗ%20ΕΡΓΑΣΙΑ-ΑΠΟΣΤΟΛΙΔΗΣ%20ΧΡΗΣΤΟΣ.pdf?isAllowed=y&sequence=1>

<http://repository.library.teiwest.gr/xmlui/bitstream/handle/123456789/6450/ΚΑΤΑΣΚΕΥΗ%20ΙΣΤΟΣΕΛΙΔΑΣ%20ΥΠΟΣΤΗΡΙΞΗΣ%20ΤΗΣ%20ΠΛΑΤΦΟΡΜΑΣ%20ARDUINO%20ΜΕ%20ΠΑΡΑΔΕΙΓΜΑΤΑ%20ΚΑΙ%20ΕΦΑΡΜΟΓΕΣ..pdf?isAllowed=y&sequence=1>

Συγκεντρωτικά ο κώδικας των μικροελεγκτών :

Software Bluetooth

```
// ΑΡΧΙΚΟΠΟΙΗΣΕΙΣ
```

```
int triggerPin = 5;
```

```
int echoPin = 4;
```

```
int val;
```

```
int nb;
```

```
String voice;
```

```
// ΟΡΙΣΜΟΣ PINS ΣΕ OUTPUT ΚΑΙ INPUT
```

```
void setup() {
```

```
Serial.begin(9600);
```

```
pinMode(13,OUTPUT);  
pinMode(12,OUTPUT);  
pinMode(11,OUTPUT);  
pinMode(10,OUTPUT);  
pinMode(9,OUTPUT);  
pinMode(8,OUTPUT);  
pinMode(7,OUTPUT);  
pinMode(6,OUTPUT);  
pinMode(1,OUTPUT);  
pinMode(0,INPUT);  
pinMode(triggerPin,OUTPUT);  
pinMode(echoPin,INPUT);  
}
```

```
// ΜΕ ΑΥΤΕΣ ΤΙΣ ΕΝΤΟΛΕΣ ΚΑΝΟΥΜΕ ΝΑ ΕΠΙΚΟΙΝΩΝΟΥΝ ΤΑ 3 VOID ΜΕΤΑΞΥ ΤΟΥΣ
```

```
void loop(){  
  { bluetooth();  
    sensor();}
```

```
  { speak(); //ΜΕ ΤΙΣ ΑΓΚΙΛΕΣ ΛΕΙΤΟΥΡΓΕΙ ΚΑΙ Η ΕΦΑΡΜΟΓΗ VOICE ΚΑΙ ΤΟ  
    BLUETOOTH CONTROL APP
```

```
    sensor();}
```

```

}

// ΜΕ ΤΙΣ ΕΝΤΟΛΕΣ ΑΥΤΕΣ ΛΕΙΤΟΥΡΓΕΙ ΜΕΣΩ ΧΡΕΙΡΙΣΤΗΡΙΟΥ

void bluetooth()

{

if(Serial.available()>0)

{

int data= Serial.read();

Stop();

if(data=='R')

{

digitalWrite(13,LOW);

digitalWrite(12,LOW);

digitalWrite(11,HIGH);

digitalWrite(10,LOW);

}

else if(data=='L')

{

digitalWrite(13,LOW);

digitalWrite(12,LOW);

digitalWrite(11,LOW);

digitalWrite(10,HIGH);

}

}

```



```
else if(data=='F')  
  
{  
  
digitalWrite(13,LOW);  
  
digitalWrite(12,HIGH);  
  
digitalWrite(11,LOW);  
  
digitalWrite(10,LOW);  
  
}
```

```
else if(data=='B')  
  
{  
  
digitalWrite(13,HIGH);  
  
digitalWrite(12,LOW);  
  
digitalWrite(11,LOW);  
  
digitalWrite(10,LOW);  
  
digitalWrite(7,HIGH);
```

```
tone(6,450);  
  
delay(500);  
  
noTone(6);  
  
delay(500);  
  
}
```

```
else if (data=='G')
```

```
{  
digitalWrite(13, LOW);  
digitalWrite(12, HIGH);  
digitalWrite(11, LOW);  
digitalWrite(10, HIGH);  
}
```

```
else if (data=='I')
```

```
{  
digitalWrite(13, LOW);  
digitalWrite(12, HIGH);  
digitalWrite(11, HIGH);  
digitalWrite(10, LOW);  
}
```

```
else if (data=='H')
```

```
{  
digitalWrite(13, HIGH);  
digitalWrite(12, LOW);  
digitalWrite(11, LOW);  
digitalWrite(10, HIGH);  
digitalWrite(7,HIGH);
```

```
tone(6,450);
```

```
delay(500);

noTone(6);

delay(500);

}

else if (data == 'J')

{

tone(6,450);

    delay(500);

    noTone(6);

    delay(500);

    digitalWrite(13, HIGH);

    digitalWrite(12, LOW);

    digitalWrite(11, HIGH);

    digitalWrite(10, LOW);

    digitalWrite(7,HIGH);

}

else if (data== 'W')

{

    digitalWrite(9, HIGH);

}

}
```

```
else if (data=='w')  
{  
    digitalWrite(9, LOW);  
}
```

```
else if (data== 'Q')  
{  
    digitalWrite(8, HIGH);  
}
```

```
else if (data=='q')  
{  
    digitalWrite(8, LOW);  
}
```

```
else if(data=='K')  
{  
    digitalWrite(6,HIGH);  
}
```

```
}
```

```
}
```

```
void Stop()
```

```
{
```

```

digitalWrite(13,LOW);

digitalWrite(12,LOW);

digitalWrite(11,LOW);

digitalWrite(10,LOW);

digitalWrite(7,LOW);

digitalWrite(6,LOW);

}

void Stopcar()

{

    digitalWrite(13,LOW);

    digitalWrite(12,LOW);

    digitalWrite(11,LOW);

    digitalWrite(10,LOW);

}

// ΕΝΤΟΛΕΣ ΓΙΑ ΤΟΝ ΣΕΝΣΟΡΑ ΑΠΟΣΤΑΣΗΣ

void sensor(){

    int duration, distance; //ΔΗΛΩΣΗ ΑΠΟΣΤΑΣΗΣ

    digitalWrite(triggerPin, HIGH);

    delay(10);

    digitalWrite(triggerPin, LOW);

    duration = pulseIn(echoPin, HIGH);

```

```
distance = (duration/2) / 29.1; //ΜΕ ΑΥΤΗ ΤΗΝ ΕΝΤΟΛΗ ΤΗΝ ΧΡΗΣΙΜΟΠΟΙΟΥΜΕ ΓΙΑ  
ΜΕΤΑΤΡΟΠΗ ΣΕ ΙΝΤΣΕΣ
```

```
Serial.print(distance);
```

```
Serial.print("cm");
```

```
Serial.println(" ");
```

```
// ΜΕ ΑΥΤΕΣ ΤΙΣ ΕΝΤΟΛΕΣ ΦΤΕΙΑΧΝΟΥΜΕ ΤΗΝ ΑΠΟΣΤΑΣΗ ΠΟΥ ΘΕΛΟΥΜΕ ΝΑ  
ΣΤΑΜΑΤΑΕΙ ΤΟ ΑΥΤΟΚΙΝΗΤΑΚΙ
```

```
if(distance <= 55)
```

```
{
```

```
digitalWrite(13,HIGH);
```

```
digitalWrite(12,LOW);
```

```
digitalWrite(11,LOW);
```

```
digitalWrite(10,LOW);
```

```
digitalWrite(7,HIGH);
```

```
Serial.println("ALERT");
```

```
delay(250);
```

```
digitalWrite(13,LOW);
```

```
digitalWrite(12,LOW);
```

```
digitalWrite(11,LOW);
```

```
digitalWrite(10,LOW);
```

```
digitalWrite(7,LOW);
```

```

    delay(2000);

}

}

//ΕΝΤΟΛΕΣ ΓΙΑ ΤΟ VOICE CONTROL

void speak ()

{

while (Serial.available()){

    delay(10);

    char c = Serial.read();

    if (c == '#') {break;}

    voice += c;

}

if (voice.length() > 0){

    if(voice == "*forward")

    {

digitalWrite(13,LOW);

digitalWrite(12,HIGH);

digitalWrite(11,LOW);

digitalWrite(10,LOW);

    delay (1000);

    }

}

```

```
else if(voice == "*backward")
```

```
{
```

```
digitalWrite(13,HIGH);
```

```
digitalWrite(12,LOW);
```

```
digitalWrite(11,LOW);
```

```
digitalWrite(10,LOW);
```

```
digitalWrite(7,HIGH);
```

```
delay (1000);
```

```
}
```

```
else if(voice == "*right")
```

```
{
```

```
digitalWrite(13,LOW);
```

```
digitalWrite(12,LOW);
```

```
digitalWrite(11,HIGH);
```

```
digitalWrite(10,LOW);
```

```
delay(800);
```

```
}
```

```
else if(voice == "*left")
```

```
{
```

```
digitalWrite(13,LOW);
```

```
digitalWrite(12,LOW);
```

```
digitalWrite(11,LOW);
```

```
digitalWrite(10,HIGH);
```



```
delay(800);

}

else if(voice == "*light on")

{

digitalWrite(9,HIGH);

}

else if(voice == "*light off")

{

digitalWrite(9,LOW);

}

else if(voice == "*lamp on")

{

digitalWrite(8,HIGH);

}

else if(voice == "*lamp off")

{

digitalWrite(8,LOW);

}

else if(voice == "*stop")

{

digitalWrite(10,LOW );

digitalWrite(11,LOW);

digitalWrite(12,LOW );
```

```

digitalWrite(13,LOW);

digitalWrite(7,LOW);

}

voice=""; //ΕΑΝ ΔΕΝ ΔΟΘΕΙ ΤΙΠΟΤΑ ΤΟΤΕ ΕΠΑΝΑΛΑΒΕΤΑΙ ΤΗ ΛΕΞΗ
}
}

```

Software WiFi

```

#define A1      15    // L298N in1 Πίσω Κίνηση Μοτέρ      GPIO15(D8)

                // Μαζί με το A1 Ενεργοποιούνται τα φώτα τις όπισθεν

#define A2      13    // L298N in2 Μπροστά Κίνηση Μοτέρ      GPIO13(D7)

#define B1      2     // L298N in3 Αριστερή Κίνηση      GPIO2(D4)

#define B2      0     // L298N in4 Δεξιά Κίνηση      GPIO0(D3)

#define triggerPin  5    // ultrasonic sensor Trigger      GPIO5(D1)

#define echoPin    16    // ultrasonic sensor Echo      GPIO16(D0)

#define led        14    // Φώτα μπροστά μεγάλα και πίσω    GPIO14(D5)

#define lamp       12    // Φώτα μικρά                      GPIO12(D6)

#define buzzer     4     // Κόρνα                            GPIO4(D2)

```

```
#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <ESP8266WebServer.h>
```

```
String command;
```

```
const char* ssid = "VMS";

ESP8266WebServer server(80);
```

```
void setup() {
```

```
    pinMode(A1, OUTPUT);

    pinMode(A2, OUTPUT);

    pinMode(B1, OUTPUT);

    pinMode(B2, OUTPUT);

    pinMode(led,OUTPUT);

    pinMode(lamp,OUTPUT);

    pinMode(buzzer,OUTPUT);

    pinMode(triggerPin, OUTPUT);
```

```
pinMode(echoPin,INPUT);

Serial.begin(9600);

// Σύνδεση WiFi

WiFi.mode(WIFI_AP);

WiFi.softAP(ssid);

IPAddress myIP = WiFi.softAPIP();

Serial.print("AP IP address: ");

Serial.println(myIP);

// Starting WEB-server

server.on ( "/", HTTP_handleRoot );

server.onNotFound ( HTTP_handleRoot );

server.begin();

}

void forward(){

    digitalWrite(A1, LOW);

    digitalWrite(A2, HIGH);
```

```
digitalWrite(B1, LOW);
```

```
digitalWrite(B2, LOW);
```

```
}
```

```
void backward(){
```

```
digitalWrite(A1, HIGH);
```

```
digitalWrite(A2, LOW);
```

```
digitalWrite(B1, LOW);
```

```
digitalWrite(B2, LOW);
```

```
tone(buzzer,450);
```

```
delay(500);
```

```
noTone(buzzer);
```

```
delay(500);
```

```
}
```

```
void Right(){
```

```
    digitalWrite(A1, LOW);
```

```
    digitalWrite(A2, LOW);
```

```
    digitalWrite(B1, HIGH);
```

```
    digitalWrite(B2, LOW);
```

```
}
```

```
void Left(){
```

```
    digitalWrite(A1, LOW);
```

```
    digitalWrite(A2, LOW);
```

```
    digitalWrite(B1, LOW);
```

```
    digitalWrite(B2, HIGH);
```

```
}
```

```
void goRight(){
```

```
    digitalWrite(A1, LOW);
```

```
    digitalWrite(A2, HIGH);
```

```
    digitalWrite(B1, HIGH);
```

```
    digitalWrite(B2, LOW);
```

```
}
```

```
void goLeft(){
```

```
    digitalWrite(A1, LOW);
```

```
    digitalWrite(A2, HIGH);
```

```
    digitalWrite(B1, LOW);
```

```
    digitalWrite(B2, HIGH);
```

```
}
```

```
void BackRight(){

    digitalWrite(A1, HIGH);

    digitalWrite(A2, LOW);

    digitalWrite(B1, HIGH);

    digitalWrite(B2, LOW);

    tone(buzzer,450);

    delay(500);

    noTone(buzzer);

    delay(500);

}
```

```
void BackLeft(){

    digitalWrite(A1, HIGH);
```



```
digitalWrite(A2, LOW);
```

```
digitalWrite(B1, LOW);
```

```
digitalWrite(B2, HIGH);
```

```
tone(buzzer,450);
```

```
delay(500);
```

```
noTone(buzzer);
```

```
delay(500);
```

```
}
```

```
void stopCar(){
```

```
digitalWrite(A1, LOW);
```

```
digitalWrite(A2, LOW);
```

```
digitalWrite(B1, LOW);
```

```
digitalWrite(B2, LOW);
```

```
}
```

```
void ledON() {
```

```
digitalWrite(led,HIGH);
```

```
}
```

```
void ledOFF() {
```

```
digitalWrite(led,LOW);
```

```
}
```

```
void lampON() {
```

```
digitalWrite(lamp,HIGH);
```

```
}
```

```
void lampOFF() {
```

```
digitalWrite(lamp,LOW);
```

```
}
```

```
void buzzerON() {
```

```
tone(buzzer,500);
```

```
}
```

```
void buzzerOFF() {
```

```
digitalWrite(buzzer,LOW);
```

```
}
```

```
void loop () {
```

```
  controller();
```

```
  sensor();
```

```
}
```

```
void controller() {
```

```
  server.handleClient();
```

```
  command = server.arg("State");
```

```
  if (command == "F") forward();
```

```
  else if (command == "B") backward();
```

```
  else if (command == "L") Left();
```

```
  else if (command == "R") Right();
```

```
  else if (command == "I") goRight();
```

```
  else if (command == "G") goLeft();
```

```
  else if (command == "J") BackRight();
```

```
  else if (command == "H") BackLeft();
```

```

else if (command == "S") stopCar();

else if (command == "W") ledON();

else if (command == "w") ledOFF();

else if (command == "Q") lampON();

else if (command == "q") lampOFF();

else if (command == "K") buzzerON();

else if (command == "k") buzzerOFF();

}

void HTTP_handleRoot(void) {

if( server.hasArg("State") ){

    Serial.println(server.arg("State"));

}

server.send ( 200, "text/html", "" );

delay(1);

}

```

```

void sensor(){

```

```

int duration, distance;

digitalWrite(triggerPin, HIGH);

delay(10);

digitalWrite(triggerPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = (duration/2) / 29.1; //ΜΕ ΑΥΤΗ ΤΗΝ ΕΝΤΟΛΗ ΤΗΝ ΧΡΗΣΙΜΟΠΟΙΟΥΜΕ ΓΙΑ
ΜΕΤΑΤΡΟΠΗ ΣΕ ΙΝΤΣΕΣ

Serial.print(distance);

Serial.print("cm");

Serial.println(" ");

// ΜΕ ΑΥΤΕΣ ΤΙΣ ΕΝΤΟΛΕΣ ΦΤΕΙΑΧΝΟΥΜΕ ΤΗΝ ΑΠΟΣΤΑΣΗ ΠΟΥ ΘΕΛΟΥΜΕ ΝΑ
ΣΤΑΜΑΤΑΕΙ ΤΟ ΑΥΤΟΚΙΝΗΤΑΚΙ

if (distance <= 40)
{
    backward ();

    delay(250);

    stopCar();

    delay(2000);

}

```

```
}
```

Software GPS

```
#include <TinyGPS.h>
```

```
TinyGPS gps;
```

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

```
SoftwareSerial ss(2, 3);
```

```
unsigned long fix_age, time, date, speed, course;
```

```
void printFloat(double f, int digits = 2);
```

```
void setup()
```

```
{
```

```
Serial.begin(115200);
```

```
ss.begin(9600);
```

```
}
```

```
//////////OTAN DOTHOUN SWSTA TA STOIXEIA TOTE ENERGOPOIHTE TO GPG//////////
```

```
void loop()
```

```

{
bool newData = false;

ss.println("AT");

ss.println("AT+CGNSPWR=1");

ss.println("AT+CGNSTST=1");

for (unsigned long start = millis(); millis() - start < 1000;)
{
while (ss.available())
{

char c = ss.read();

Serial.write(c);

if (gps.encode(c))

newData = true;

}

}

//////////////////O KWDIKAS GIA TI LEITOURGEIA TOU GPS//////////////////

if (newData)
{

float latitude, longitude;

unsigned long age;

float fkmph;

```

```

int year;

byte month, day, hour, minute, second, hundredths;

gps.f_speed_kmph();

gps.f_get_position(&latitude, &longitude, &age);

gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths, &age);

Serial.print("LAT=");

Serial.print(latitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : latitude, 6);

Serial.print(" LON=");

Serial.print(longitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : longitude, 6);

Serial.print(" SAT=");

Serial.print(gps.satellites() == TinyGPS::GPS_INVALID_SATELLITES ? 0 : gps.satellites());
/////APO EDW KAI KATW BLEPOUME TA STOIXEIA POU EXRONTAI APO THN
KAIREA/////

Serial.print("Date: "); Serial.print(static_cast<int>(month));

Serial.print("/");

Serial.print(static_cast<int>(day)); Serial.print("/");

Serial.print(year);

Serial.print(" Time: "); Serial.print(static_cast<int>(hour+3));

Serial.print(":");

Serial.print(static_cast<int>(minute)); Serial.print(":");

Serial.print(static_cast<int>(second));

Serial.print("."); Serial.print(static_cast<int>(hundredths));

Serial.print("Alt: "); Serial.print(gps.f_altitude());

```



```
Serial.print(" (kmph): "); printFloat(gps.f_speed_kmph());
```

```
Serial.println();
```

```
//////EDW KANOUME THN METATROPH SE STRING ETSI WSTE NA MPOROUN NA  
STALOUN STH BASH////////
```

```
String mystring1=String(year);
```

```
String mystring2=String(month);
```

```
String mystring3=String(day);
```

```
String mystring4=String(hour+3);
```

```
String mystring5=String(minute);
```

```
String mystring6=String(second);
```

```
String datetime= mystring1 + "-" + mystring2 + "-" + mystring3 + "/" + mystring4 + ":" + mystring5  
+ ":" + mystring6;
```

```
Serial.println(datetime);
```

```
//////EDW THA VLEPOUME STO XARTI TOUS PARAKATW XARASKTHRES //////////
```

```
String Text1 = "LAT=";
```

```
String Text2 = "LON=";
```

```
String Text3 = "-";
```

```
//////KANW TON DIAXORISMO TWN ARNHTIKWN ARITHON////////
```

```
void printFloat(double number, int digits)
```

```
{
```

```

if (number < 0.0) {

    Serial.write('-');

    number = -number;

}

```

////////KANW STROGGYLOPOIISI TWN ARITHMON////////

```

double rounding = 0.5;

for (uint8_t i=0; i<digits; ++i)

    rounding /= 10.0;

number += rounding;

```

////////EMFANIZW TOUS AKERAIIOUS ARITHMOUS////////

```

unsigned long int_part = (unsigned long)number;

double remainder = number - (double)int_part;

Serial.print(int_part);

```

////////EMFANIZW TOUS DEKADIKOUS ARITHMOUS////////

```

if (digits > 0)

    Serial.print(".");

```

//////////EMFANIZW ENA ENA TA PSIFIA//////////

```

while (digits-- > 0) {

    remainder *= 10.0;

    int toPrint = int(remainder);

```

```
Serial.print(toPrint);  
  
remainder -= toPrint;  
  
}  
  
}
```