

ΤΕΙ ΗΠΕΙΡΟΥ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΙΚΡΟΕΛΕΓΚΤΗ ΚΑΙ Η ΚΑΤΑΣΚΕΥΗ ΤΗΛΕΚΑΤΕΥΘΥΝΟΜΕΝΟΥ ΟΧΗΜΑΤΟΣ (Arduino)



ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:
ΑΝΤΩΝΙΑΔΗΣ ΝΙΚΟΛΑΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ:
ΚΑΣΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΑΡΤΑ 2017

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Αρτα, / /2017

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1.

2.

3.

<< Δηλώνω υπεύθυνα ότι το παρόν κείμενο αποτελεί προϊόν προσωπικής μελέτης και εργασίας και πως όλες οι πηγές που χρησιμοποιήθηκαν για τη συγγραφή της δηλώνονται είτε στις παραπομπές είτε στη βιβλιογραφία. Γνωρίζω πως η λογοκλοπή αποτελεί σοβαρότατο παράπτωμα και είμαι ενήμερος για την επίπτωση των νόμιμων συνεπειών >>

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον εισηγητή της πτυχιακής μου εργασίας Κ. Αντωνιάδη Νικόλαο, ο οποίος με εμπιστεύτηκε πως θα φέρω εις πέρας αυτό το δύσκολο έργο και ανέλαβε την επίβλεψη της εργασίας μου. Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου για την στήριξη καθ' όλη τη διάρκεια των σπουδών μου, τους φίλους και τα αγαπημένα μου πρόσωπα που με στήριξαν κατά τη διάρκεια υλοποίησης της εργασίας αυτής.

ΠΕΡΙΛΗΨΗ

Η πτυχιακή αυτή εργασία έχει σαν στόχο να αναλύσει τα βασικά στοιχεία και τις λειτουργίες των μικροεπεξεργαστών και των μικροελεγκτών και να συγκρίνει τα δυο αυτά κυκλώματα μεταξύ τους. Επιπλέον, αναλύονται τα πλεονεκτήματα και τα μειονεκτήματα των μικροελεγκτών και των ενσωματωμένων συστημάτων. Γίνεται μια εισαγωγή στις βασικές αρχές των ενσωματωμένων συστημάτων, που χρησιμοποιείτε το κάθε ένα και που τα συναντάμε καθημερινά στην ζωή μας. Ακόμα παρατίθενται οι εταιρίες μικροελεγκτών ονομαστικά και αναλυτικά.

Γίνεται μια αναφορά για το Arduino και συγκεκριμένα για το Arduino Uno, τα τεχνικά χαρακτηριστικά και αναφορά για ένα μέρος της οικογένειάς τους, διάφορες πλακέτες μικροελεγκτών και ακόμα πιο αναλυτικά χαρακτηριστικά για τις λειτουργίες που μπορεί να εκτελέσει και τις δυνατότητες που διαθέτει. Γίνεται μεγάλη αναφορά για τους ανταγωνιστές και συγκεκριμένα για τον μεγάλο ανταγωνιστή της την Texas Instruments. Το προγραμματιστικό περιβάλλον που χρησιμοποιεί και πως θα εισάγουμε σε αυτό επιπλέον βιβλιοθήκες με διάφορους τρόπους. Αξίζει να αναφερθεί ότι υπάρχουν και αναλυτικά προγραμματιστικά περιβάλλοντα για την πλακέτα του Arduino και σε πιο λειτουργικά συστήματα είναι συμβατά.

Ακόμα το πιο σημαντικό κομμάτι της εργασίας είναι η κατασκευή και ο προγραμματισμός του οχήματος. Θα δείτε αναλυτικά τα υλικά που θα χρειαστείτε για την δημιουργία του οχήματος αυτού ακόμα φωτογραφίες και διευθύνσεις για το κάθε υλικό. Την κατασκευή του οχήματος αναλυτικά από τις βίδες μέχρι και την πλακέτα του Arduino με αναλυτικά τα βήματα και φωτογραφίες που θα σας βοηθήσουν να δημιουργήσετε το όχημα εύκολα και γρήγορα. Όπως ακόμα διαγράμματα για την συνδεσμολογία του οχήματος με μια αναλυτική αναφορά στις κινήσεις που πρέπει να κάνετε. Κομμάτια κώδικα που θα εισάγετε στο πρόγραμμα σας με τον τρόπο αυτό θα μπορέσετε να κάνετε το αυτοκίνητο να κάνει τις κινήσεις που εσείς θέλετε. Ακόμα και την εισαγωγή ενός τηλεκοντρόλ στο οποίο προγραμματίζουμε το κάθε κουμπί ξεχωριστά με αποτέλεσμα να κάνει τις κινήσεις που κάνει ένα τηλεκατευθυνόμενο όχημα.

Επιπλέον, ένα σημαντικό κεφάλαιο είναι τα προβλήματα που μπορεί να αντιμετωπίσετε και τους τρόπους που μπορείτε να τα αποφύγετε στον προγραμματισμό, στον κώδικα ακόμα και στην εγκατάσταση του προγραμματιστικού περιβάλλοντος και τις συνδεσμολογίες που μπορεί να γίνουν λάθος.

Περιεχόμενα

ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ	7
1.ΕΙΣΑΓΩΓΗ.....	11
2) ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΕΣ ΚΑΙ ΜΙΚΡΟΕΛΕΓΤΕΣ	12
2.1 ΙΣΤΟΡΙΑ ΤΩΝ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ.....	12
2.2 ΜΙΚΡΟΕΛΕΚΤΗΣ.....	13
2.3 ΒΑΣΙΚΑ ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΜΙΚΡΟΕΛΕΓΚΤΩΝ.....	14
2.4 ΕΝΣΩΜΑΤΟΜΕΝΑ ΣΥΣΤΗΜΑΤΑ	15
2.4 ΕΝΣΩΜΑΤΟΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΣΤΗ ΖΩΗ ΜΑΣ.....	16
2.5 ΕΤΑΙΡΙΕΣ ΚΑΙ ΜΟΝΤΕΛΑ.....	16
3 ΑΝΑΦΟΡΑ ΓΙΑ ΤΟ ARDUINO	20
3.1 ΤΙ ΕΙΝΑΙ ΤΟ ARDUINO.....	20
3.2 ARDUINO UNO	21
3.2.1 ΤΕΧΝΙΚΕΣ ΠΡΟΔΙΑΓΡΑΦΕΣ ARDUINO UNO.....	22
3.2.2 ΑΚΡΟΔΕΚΤΕΣ ΤΡΟΦΟΔΟΣΙΑΣ	22
3.3 ΟΙΚΟΓΕΝΕΙΑ ΤΟΥ ARDUINO	25
3.3.1 ARDUINO LEONARDO.....	25
3.3.2 ARDUINO 101.....	25
3.3.3 ARDUINO MICRO.....	26
3.3.4 ARDUINO NANO	27
3.3.5 ARDUINO MEGA	27
3.3.6 ARDUINO LILYPAD.....	28
3.3.7 ARDUINO ETHERNET	28
4 ΠΩΣ ΠΡΟΓΡΑΜΜΑΤΙΖΕΤΑΙ ΤΟ ARDUINO	29
4.1 ARDUINO IDE	30
4.2 ΠΛΑΤΦΟΡΜΕΣ ΜΕ ΤΙΣ ΟΠΟΙΕΣ ΜΠΟΡΟΥΜΕ ΝΑ ΠΡΟΓΡΑΜΑΤΙΣΟΥΜΕ ΤΗΣ ΠΛΑΚΕΤΕΣ.....	32
4.2.1 PROGRAMINO IDE.....	32
4.2.2 ARDUINO ΓΙΑ VISUAL STUDIO.....	34
4.2.3 PLUTO – Πρωτότυπο Arduino στην Python	34
4.2.4 EMBRIO	35
4.2.5 ZERYNTH	36
4.2.6 PlatformIO IDE	37
4.2.7 Deviot	38
4.3 ΒΙΒΛΙΟΘΗΚΕΣ ΓΙΑ ΤΟ ARDUINO IDE ΚΑΙ ΠΩΣ ΓΙΝΕΤΑΙ Η ΕΓΚΑΤΑΣΤΑΣΗ ΑΥΤΩΝ.....	39

5. ΣΧΕΔΙΑΣΜΟΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	46
5.1 ΤΙ ΘΕΛΟΥΜΕ ΝΑ ΦΤΙΑΞΟΥΜΕ.....	46
5.2 ΤΙ ΥΛΙΚΟ ΘΑ ΧΡΕΙΑΣΤΟΥΜΕ	46
5.3 ΤΙ ΕΝΕΡΓΙΕΣ ΠΡΕΠΕΙ ΝΑ ΓΙΝΟΥΝ ΓΙΑ ΒΡΕΘΟΥΝ ΤΑ ΥΛΙΚΑ ΑΥΤΑ	53
6 ΥΛΟΠΟΙΗΣΗ	54
6.1 ΣΥΝΑΡΜΟΛΟΓΙΣΗ ΤΟΥ ΟΧΗΜΑΤΟΣ.....	54
6.2 ΣΥΝΔΕΣΜΟΛΟΓΙΕΣ	59
6.3 ΠΡΟΓΡΑΜΑΤΙΣΜΟΣ ΤΗΣ ΠΛΑΚΕΤΑΣ ΤΟΥ ΟΧΗΜΑΤΟΣ	61
6.3.1 ΠΡΩΤΟ ΠΡΟΓΡΑΜΜΑ	63
6.3.2 ΔΕΥΤΕΡΟ ΠΡΟΓΡΑΜΜΑ (κώδικας του τηλεχειριστηρίου (case))	66
7) ΔΟΚΙΜΕΣ.....	73
8)ΣΥΜΠΕΡΑΣΜΑΤΑ	75
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	77

ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ

Εικόνα 1.1: Μικροεπεξεργαστές	12
Εικόνα1.2: Μικροελεγκτές.....	14
Εικόνα 1.3: Χέρι ρομπότ.....	15
Εικόνα 1.4: Ενσωματωμένα συστήματα στην ζωή μας.....	16
Εικόνα 1.5: Εταιρίες μικροελεγκτών.....	16
Εικόνα 2.1: Απεικόνιση συμβολισμού εταιρίας Arduino.....	19
Εικόνα 2.2: Arduino Uno.....	20
Εικόνα 2.3: Τεχνικές προδιαγραφές.....	21
Εικόνα 2.4: Ακροδέκτες τροφοδοσίας Arduino Uno.....	22
Εικόνα 2.5: Ψηφιακές είσοδοι Arduino Uno.....	23
Εικόνα 2.6: Ακροδέκτες τροφοδοσίας power(αριστερά) αναλογική είσοδοι analog in (δεξιά) στην πλακέτα Arduino Uno.....	23
Εικόνα 2.7: Arduino Leonardo.....	24
Εικόνα 2.8: Arduino 101	24
Εικόνα 2.9: Arduino Micro.....	25
Εικόνα 2.10: Arduino Nano.....	25
Εικόνα 2.11: Arduino Mega.....	26

Εικόνα 2.12: Arduino Lilypad	26
Εικόνα:2.13: Arduino Ethernet.....	27
Εικόνα 3.1: Προγραμματιστικό περιβάλλον Arduino IDE.....	28
Εικόνα 3.2: Programino Ide.....	30
Εικόνα 3.3: Arduino για Visual Studio.....	31
Εικόνα 3.4: Πρωτότυπο Arduino στην Python(PLUTO).....	32
Εικόνα 3.5: Προγραμματιστικό περιβάλλον Embrio.....	33
Εικόνα 3.6: Προγραμματιστικό περιβάλλον Zerunth.....	34
Εικόνα 3.7: Προγραμματιστικό περιβάλλον Zerunth για Smartphones.....	34
Εικόνα 3.8:Προγραμματιστικό περιβάλλον PlatformIO IDE.....	35
Εικόνα 3.9:Προγραμματιστικό περιβάλλον Deviot	36
Εικόνα 3.10: Εισαγωγή Βιβλιοθήκης από υπάρχον βιβλιοθήκη(βήμα 1 ^ο)	37
Εικόνα 3.11: Εισαγωγή Βιβλιοθήκης από υπάρχον βιβλιοθήκη(βήμα 2 ^ο)	38
Εικόνα 3.12: Εισαγωγή Βιβλιοθήκης από ZIP(βήμα 1 ^ο)	38
Εικόνα 3.13: Εισαγωγή Βιβλιοθήκης από ZIP(βήμα 2 ^ο).....	39
Εικόνα 3.14: Εισαγωγή Βιβλιοθήκης από ZIP(βήμα 3 ^ο).....	39
Εικόνα 3.15: Εισαγωγή Βιβλιοθήκης από ZIP χειροκίνητα (βήμα 1 ^ο).....	40
Εικόνα 3.16: Εισαγωγή Βιβλιοθήκης από ZIP χειροκίνητα (βήμα 2 ^ο).....	40
Εικόνα 3.17: Εισαγωγή Βιβλιοθήκης από ZIP χειροκίνητα (βήμα 3 ^ο).....	41
Εικόνα 3.18: Εισαγωγή Βιβλιοθήκης από ZIP χειροκίνητα (βήμα 4 ^ο).....	41
Εικόνα 3.19: Εισαγωγή Βιβλιοθήκης από ZIP χειροκίνητα (βήμα 5 ^ο).....	42
Εικόνα 4.1:Υλικά για την κατασκευή του σκελετού του οχήματος [23].....	44
Εικόνα 4.2: Arduino Uno[24].....	44
Εικόνα 4.3:Πλακέτα L298N [25].....	45
Εικόνα 4.4:Πλακέτα wireless και τηλεχειριστήριο [26].....	46
Εικόνα 4.4.1: Τηλεχειριστήριο.....	65
Εικόνα 4.5: Θήκη μπαταρίας κλιπ για 9V μπαταρία[27].....	46
Εικόνα 4.6: Μπαταρίες ονομαστικά [28].....	47
Εικόνα 4.7: Μπαταρίες 2A [29].....	47
Εικόνα 4.8: Μπαταρία CR2025 [30].....	48

Εικόνα 4.9: Μπαταρία 9V.....	48
Εικόνα 4.10: Καλώδια σύνδεσης [31].....	49
Εικόνα 4.11: Power Bank[32].....	49
Εικόνα 5.1: Βάση οχήματος.....	51
Εικόνα 5.2: Κατευθυντήριο τροχός.....	51
Εικόνα 5.3: Σύνδεση κατευθυντήριου τροχού και βάση οχήματος (Α).....	51
Εικόνα 5.4: Σύνδεση κατευθυντήριου τροχού και βάση οχήματος (Β).....	51
Εικόνα 5.5: Σύνδεση μοτέρ με ρόδες (Α).....	52
Εικόνα 5.6: Σύνδεση μοτέρ με ρόδες (Β).....	52
Εικόνα 5.7: Σύνδεση μοτέρ με ρόδες (Γ).....	52
Εικόνα 5.8: Σύνδεση μοτέρ με ρόδες (Δ).....	52
Εικόνα 5.9: Σύνδεση μοτέρ με ρόδες (Ε).....	52
Εικόνα 5.10: Σύνδεση μοτέρ με ρόδες (Ζ).....	52
Εικόνα 5.11: Σύνδεση μοτέρ με ρόδες στην βάση (Α).....	53
Εικόνα 5.12: Σύνδεση μοτέρ με ρόδες στην βάση (Β).....	53
Εικόνα 5.13: Σύνδεση μοτέρ με ρόδες στην βάση (Γ).....	53
Εικόνα 5.14: Σύνδεση μοτέρ με ρόδες στην βάση (Δ).....	53
Εικόνα 5.15: Σύνδεση βάση μπαταριών με την βάση του οχήματος (Α).....	53
Εικόνα 5.16: Σύνδεση βάση μπαταριών με την βάση του οχήματος (Β).....	53
Εικόνα 5.17: Πλακέτα L298N.....	54
Εικόνα 5.18: Σύνδεση πλακέτας L98N με την βάση του οχήματος.....	54
Εικόνα 5.19: Πλακέτα Arduino Uno.....	54
Εικόνα 5.20: Σύνδεση πλακέτας Arduino Uno με την βάση του οχήματος.....	54
Εικόνα 5.21: Πλακέτα wireless	55
Εικόνα 5.22: Σύνδεση πλακέτας wireless με την βάση του οχήματος.....	55
Εικόνα 6.1: Σύνδεση πλακέτας Arduino Uno με πλακέτα L298N και L298N με τα δύο μοτέρ.....	55
Εικόνα 6.2: Συνδεσμολογία μεταξύ θήκης μπαταριών και πλακέτας L298N.....	56
Εικόνα 6.3: Συνδεσμολογία μεταξύ πλακέτας Arduino Uno και wireless.....	57
Εικόνα 7.1: Πρώτο πρόγραμμα	58
Εικόνα 7.2: Πρώτο πρόγραμμα.....	58

Εικόνα 7.3: Πρώτο πρόγραμμα.....	59
Εικόνα 7.4: Πρώτο πρόγραμμα.....	60
Εικόνα 7.5: Δεύτερο πρόγραμμα straight()	61
Εικόνα 7.6: Δεύτερο πρόγραμμα backOne().....	61
Εικόνα 7.7: Δεύτερο πρόγραμμα leftOne().....	62
Εικόνα 7.8: Δεύτερο πρόγραμμα rightOne().....	62
Εικόνα 7.9: Δεύτερο πρόγραμμα rotation().....	63
Εικόνα 7.10: Δεύτερο πρόγραμμα.....	63
Εικόνα 7.11: Δεύτερο πρόγραμμα.....	64
Εικόνα 7.12: Δεύτερο πρόγραμμα loop().....	65
Εικόνα 7.13: Δεύτερο πρόγραμμα Buttons().....	66

1.ΕΙΣΑΓΩΓΗ

Στη σύγχρονη εποχή οι μικροελεγκτές αποτελούν αναπόσπαστο κομμάτι της καθημερινότητάς μας. Συναντώνται σε πολλές εφαρμογές που χρησιμοποιούμε ανελλιπώς στην καθημερινότητα μας, είτε στο σπίτι, είτε στο γραφείο και βρίσκονται παντού στις ηλεκτρικές συσκευές και όχι μόνο (από το κινητό τηλέφωνο μέχρι το αυτοκίνητο μας). Παρόλο που συνήθως αγνοούμε την ύπαρξή τους, οι μικροελεγκτές έχουν καταφέρει να κάνουν τη ζωή μας ευκολότερη και ευχάριστη.

Ο κύριος λόγος που επέλεξα να ασχοληθώ στην πτυχιακή μου εργασία με τους μικροελεγκτές είναι η ραγδαία παραγωγή μικροελεγκτών που έχουν σαν αποτέλεσμα δημιουργίας νέων συσκευών με μεγαλύτερες δυνατότητες όπως είναι οι τηλεοράσεις νέας τεχνολογίας, τα ρομπότ και πολλά άλλα δημιουργήματα. Ακόμη είναι ένα ενδιαφέρον αντικείμενο που θα είχε σαν αποτέλεσμα την δημιουργία νέων μαθημάτων σε επαγγελματικά λύκεια και πανεπιστήμια. Με τους μικροελεγκτές Arduino μπορούμε να κατασκευάσουμε τηλεκατευθυνόμενα οχήματα και άλλες κατασκευές (δηλαδή να τις προγραμματίσουμε να κάνουν τις εντολές που θέλουμε). Έχουν σαν αποτέλεσμα ενδιαφέρουσες κατασκευές που μπορούν να σας δώσουν άμεσα, γρήγορα και εύκολα το αποτέλεσμα που εσείς περιμένετε. Είναι κάτι πραγματικά πολύ όμορφο διότι βλέπετε άμεσα τα αποτελέσματα του προγραμματισμού μέσω του προγράμματος Arduino IDE.

Οι μικροελεγκτές Arduino είναι από τους πιο φιλικούς μικροελεγκτές και διαθέτουν φιλικό προγραμματιστικό περιβάλλον. Επίσης κάνουν πιο εύκολη στην εκμάθηση του μικροελεγκτή με μερικές γραμμές κώδικα. Με το πάτημα ενός κουμπιού μπορείς να κάνεις ένα μοτέρ να λειτουργήσει και όχι μόνο. Είναι συμβατό με πολλές περιφερικές πλακέτες όπως είναι οι αισθητήρες, τα Bluetooth, οι συσκευές Wi-Fi ακόμα και τα τηλεκοντρόλ όπως θα δείτε και στην συνέχεια της πτυχιακής εργασίας. Είναι ένας εύκολος τρόπος προγραμματισμού για αρχάριους και για τους εμπείρους πολύ πιο δημιουργικός.

Η αίσθηση που αποκτά κάποιος με την δημιουργία μιας κατασκευής είναι ενδιαφέρουσα, ιδιαίτερα όταν κάποιος εκτός από να το κατασκευάσει μπορεί και να το προγραμματίσει αυτή την δυνατότητα μπορεί να σου την δώσει το Arduino. Όποιος είναι πρόθυμος να ασχοληθεί και να αφιερώσει λίγο χρόνο και λίγα χρήματα θα ανταμειφθεί με ένα πολύ ωραίο αποτέλεσμα.

Οι μικροελεγκτές σε συνδυασμό με ένα δημιουργικό μυαλό μπορούν να κατασκευάσουν απίστευτα εντυπωσιακά αντικείμενα. Όπως και την δημιουργία νέων προϊόντων στην αγορά. Ένα ωραίο δημιούργημα είναι τα Drone (ηλεκτρονικά ιπτάμενα αντικείμενα που κινούνται μέσω τηλεχειριστηρίου) που έδωσαν σαν αποτέλεσμα το να μπορείτε να έχετε μια τηλεκατευθυνόμενη κάμερα από την κορυφή ενός βουνού, μέχρι και το χείλος ενός γκρεμού. Στο μέλλον τα Drone θα έχουν την δυνατότητα να μεταφέρουν έναν άνθρωπο όπου αυτός θέλει.

Η εργασία μου πιστεύω είναι ένας καλός λόγος να δούμε άμεσα ένα αποτέλεσμα που μπορεί να μας δώσει ένα ενσωματωμένο σύστημα όπως το μηχανοκίνητο όχημα που έχω κατασκευάσει που μπορεί να μας δώσει πολλές γνώσεις πάνω στα ενσωματωμένα συστήματα και συγκεκριμένα πάνω στο Arduino Uno. Αρχικά, μας

δίνει μαθήματα δημιουργίας για την κατασκευή του μηχανοκίνητου οχήματος και γενικά στην κατασκευή οχημάτων και ρομπότ. Στο δεύτερο μέρος την εκμάθηση και τον τρόπο συνδεσμολογίας που γίνεται ανάμεσα στην πλακέτα και τα μοτέρ του όχημα και τέλος τον προγραμματισμό της πλακέτας και την δυνατότητα του οχήματος να κάνει διάφορες κινήσεις με το πάτημα ενός κουμπιού είτε οι κινήσεις αυτές είναι μια ευθεία είτε ένας κύκλος. Μπορεί κάποιος ακόμα να γράψει τον δικό του κώδικα και να δημιουργήσει το δικό του σχέδιο και να το ενσωματώσει σε ένα κουμπί στο τηλεχειριστήριο. Μπορείτε να δείτε και αναλυτικά τους κώδικες οι οποίοι κάνουν το όχημα σας να πηγαίνει προς όλες τις κατευθύνσεις.

2) ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΕΣ ΚΑΙ ΜΙΚΡΟΕΛΕΓΤΕΣ

2.1 ΙΣΤΟΡΙΑ ΤΩΝ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ

Η επιθυμία που είχαν οι κατασκευαστές συστημάτων να δημιουργήσουν συστήματα με υψηλές δυνατότητες και με μικρότερα μεγέθη είχε ως αποτέλεσμα την ανάπτυξη της τεχνολογίας που χρησιμοποιήθηκε για την κατασκευή των μικροεπεξεργαστών. Οι επεξεργαστές είναι ένα ολοκληρωμένο κύκλωμα (IC) γενικού σκοπού τα οποία μπορούμε να τα προγραμματίσουμε. Η επεξεργασία όλων των δεδομένων γίνεται με μια σειρά εντολών. Η γλώσσα με την οποία εκτελούνται όλες οι εντολές είναι σε γλώσσα μηχανής και είναι αποθηκευμένες στην κύρια μνήμη, η κάθε εντολή είναι μια σειρά από δυαδικά ψηφιά όπου είναι κωδικοποιημένο το είδος της εντολής. Οι εντολές αυτές γίνονται με σκοπό την σύνδεση ανάμεσα στο λογισμικό και το υλικό ή αλλιώς ανάμεσα στο software και το hardware. Ο μικροεπεξεργαστής είναι υπεύθυνος για όλη τη λειτουργία του υπολογιστή. Το εσωτερικό του αποτελείται από έναν μεγάλο αριθμό τρανζίστορ, που κάνουν την δουλειά του διακόπτη δηλαδή είτε αφήνουν είτε δεν αφήνουν την διέλευση του ρεύματος. Επίσης έχουν καταχωριτές και αντιστάτες και πολλά ακόμα. Η λειτουργία ενός επεξεργαστή αποτελείτε πλήρως από το σύνολο εντολών που είναι ικανός να εκτελέσει.[1][2]



Εικόνα 1.1: Μικροεπεξεργαστές

2.2 ΜΙΚΡΟΕΛΕΚΤΗΣ

Μικροελεγκτές ή αλλιώς (microcontroller) είναι το αποτέλεσμα των κατασκευαστών συστημάτων, οι οποίοι στην προσπάθεια τους να δημιουργήσουν συστήματα με μικρότερο μέγεθος ενσωμάτωσαν όλες τις λειτουργίες ενός υπολογιστή σε ένα ολοκληρωμένο κύκλωμα. Θα μπορούσαμε να θεωρήσουμε ότι είναι ένας υπολογιστής μέσα σε ένα ολοκληρωμένο κύκλωμα. Περιέχει κεντρική μονάδα επεξεργασίας (CPU), μνήμη (RAM) για αποθήκευση δεδομένων και μνήμη μόνον ανάγνωσης για αποθήκευση του προγράμματος. Επίσης διαθέτουν μια γεννήτρια ορολογικών παλμών και δυνατότητες εισόδου και εξόδου δεδομένων. Η μνήμη μόνον ανάγνωσης που χρησιμοποιείτε για την αποθήκευση των προγραμμάτων του μικροελεγκτή μπορεί να είναι ROM, EPROM, EEPROM, PROM. Στο τέλος της παραγράφου αναφέρουμε αναλυτικά για την κάθε μνήμη ανάγνωσης. Οι μικροελεγκτές είναι προγραμματισμένοι για συσκευές χαμηλού κόστους μέχρι και πολύ ακριβές συσκευές κόστους ανάλογα με την πολυπλοκότητα της λειτουργίας τους. Είναι διαδεδομένοι στα καταναλωτικά προϊόντα εξαιτίας του χαμηλού. Οι δυνατότητες των μικροελεγκτών διαφέρουν πάρα πολύ ανάλογα με την χρήση που χρειάζονται. Κάποιοι από αυτούς είναι πιο γρήγοροι (ταχύτεροι) κάποιοι έχουν περισσότερη μνήμη άλλοι έχουν περισσότερες θύρες εισόδου-εξόδου κλπ. Μια άλλη ονομασία που έχουν πάρει οι μικροελεγκτές, είναι του μικροϋπολογιστή. Οι μικροελεγκτές αναπτύχθηκαν λίγο αργότερα από τους μεγάλους συγγενείς τους, οι οποίοι είναι οι μικροεπεξεργαστές. Μικροελεγκτές βρίσκουμε συνέχεια μπροστά μας σχεδόν παντού σε πολλά καταναλωτικά προϊόντα όπως π.χ. τα αυτοκίνητα, τα παιχνίδια, οι τηλεοράσεις, τα DVD, τους φούρνους μικροκυμάτων σαν ένα ενσωματωμένο σύστημα.

Για παράδειγμα ένα σύγχρονο αυτοκίνητο μπορεί να διαθέτει πάνω από 10 μικροελεγκτές. Όσο πιο εξελιγμένη είναι κάθε συσκευή μας τόσοι πιο πολλούς μικροελεγκτές έχει.

Εδώ αναφέρω με λίγα λόγια τις μνήμες ανάγνωσης αναλυτικά τι λειτουργίες που κάνει η κάθε μνήμη:

PROM: Είναι προγραμματισμένη μνήμη μόνο ανάγνωσης, είναι ένα τσιπ μνήμης στην οποία τα δεδομένα μπορούν να εγγραφούν μόνο μια φορά. Μόλις ένα πρόγραμμα έχει εγγραφεί σε έναν χώρο παραμένει εκεί για πάντα έτσι αυτό έχει σαν αποτέλεσμα να μην μπορούν να επαναπρογραμματιστούν.

ROM: Είναι οι πρώτες μνήμες ημιαγωγών που κατασκευάστηκαν και διατηρούν τα περιεχόμενα τους ακόμα και μετά την απενεργοποίηση του υπολογιστή.

EPROM: Οι μνήμες αυτές είναι μια πιο νέα εξέλιξη των μνημών ROM. Διατηρούν τα περιεχόμενα τους και μετά την απενεργοποίηση της τροφοδοσίας. Σε αντίθεση με την ROM μπορούν να διαγραφτούν και να επαναπρογραμματιστούν. Είναι ακόμα ένας ειδικός τύπος PROM που μπορεί να σβήσει τα περιεχόμενα της εάν εκτεθεί σε υπεριώδες φως.

EEPROM: Ηλεκτρική διαγραφόμενη μνήμη μόνο για ανάγνωση. Είναι ένας ειδικός τύπος PROM ο οποίος μπορεί να διαγραφεί και να επαναπρογραμματιστεί πολλές φορές. Μπορεί να σβηστεί εάν εκτεθεί σε ηλεκτρονικό φορτίο.[3][4][5]



Εικόνα 1.2: Μικροελεγκτές

2.3 ΒΑΣΙΚΑ ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΜΙΚΡΟΕΛΕΓΚΤΟΝ

Το βασικό πλεονέκτημα των μικροελεγκτών εναντίον των επεξεργαστών είναι το κόστος σχεδιασμού. Το υλικό είναι πολύ χαμηλό σε κόστος όπως είναι και ο σχεδιασμός. Ένας μικροελεγκτής είναι φθηνός για αντικατάσταση ενώ ένας επεξεργαστής είναι πολύ πιο ακριβός. Ένα ακόμα πλεονέκτημα που έχουν οι μικροελεγκτές είναι η χαμηλή κατανάλωση ρεύματος διότι ο μικροελεγκτής είναι κατασκευασμένος με την λεγόμενη τεχνολογία (CMOS) Metal Oxides

Semiconductor. Ένας μικροελεγκτής αποτελείται από CPU, RAM, ROM και I/O (είσοδο και έξοδο) για να εκτελέσει ένα ολόκληρο έργο. Πολλοί μικροελεγκτές δεν χρειάζονται κανένα άλλο ολοκληρωμένο κύκλωμα για να λειτουργήσουν. Ενώ ο επεξεργαστής δεν διαθέτει RAM, ROM και ούτε είσοδο και έξοδο. Αλλά χρησιμοποιεί τις ακίδες που διαθέτει για διαφορετική χρήση. Άλλη ακίδα έχει για την RAM άλλη για την ROM και άλλη για την είσοδο/έξοδο κλπ. Υπάρχει μειωμένη εκπομπή ηλεκτρομαγνητικών παρεμβολών και μειωμένη ευαισθησία σε αντίστοιχες παρεμβολές από άλλες ηλεκτρικές και ηλεκτρονικές συσκευές αυτό το αποτέλεσμα προέρχεται από το μικρότερο αριθμό και μήκος εξωτερικών διασυνδέσεων και χαμηλότερες ταχύτητες λειτουργίας. Έχουν μικρό μέγεθος συνολικού υπολογιστικού συστήματος και καλύτερη αξιοπιστία λόγω των λιγότερων διασυνδέσεων.[6][7]

2.4 ΕΝΣΩΜΑΤΩΜΕΝΑ ΣΥΣΤΗΜΑΤΑ

Υπάρχει ένας αθέατος κόσμος υπολογιστικών συστημάτων ο οποίος είναι αυτός που δεν γνωρίζουν πολλοί. Ο κόσμος των ενσωματωμένων συστημάτων είναι πολύ μεγαλύτερος από τον κόσμο των τυπικών υπολογιστικών συστημάτων, διότι κατασκευάζονται ετησίως δισεκατομμύρια μονάδες ενσωματωμένων συστημάτων σε σύγκριση με τις απλές τυπικές υπολογιστικές μονάδες που αναφέρονται σε κάποια εκατομμύρια. Σε συγχρόνους ελέγχους και μελέτες που έχουν γίνει σε ένα σύγχρονο σπίτι το οποίο μπορεί να διαθέτει 1 με 2 ηλεκτρονικούς υπολογιστές έχει συνήθως)πάνω από 50 ενωμένα συστήματα. Τα ενσωματωμένα συστήματα διαθέτουν την ίδια αρχιτεκτονική με τα τυπικά υπολογιστικά συστήματα. Όπως κάθε προγραμματισμένο ψηφιακό σύστημα έτσι και τα ενσωματωμένα συστήματα έχουν από έναν ή και περισσότερους επεξεργαστές, μια μνήμη όπως και διεπαφές I/O (Εισόδου και Εξόδου). Τα ενσωματωμένα συστήματα χρησιμοποιούν άλλες κατηγορίες συνήθως με χαμηλότερες δυνατότητες αλλά με περισσότερες δυνατότητες διασύνδεσης εισόδου και εξόδου όμως λιγότερους πόρους όπως είναι η μνήμη. Τα ενσωματωμένα συστήματα θα μπορούσαμε να τα θεωρήσουμε ως έναν μικρό τυπικό υπολογιστικό σύστημα ή αλλιώς μια μικρογραφία ενός τυπικού υπολογιστικού συστήματος.[8]



Εικόνα 1.3: Χέρι
ρομπότ

2.4 ΕΝΣΩΜΑΤΩΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΣΤΗ ΖΩΗ ΜΑΣ

Τα ενσωματωμένα συστήματα έχουν γίνει πλέον μέρος της καθημερινής μας ζωής (στους περισσότερους ανθρώπους). Μάλιστα πολλές φορές τα χρησιμοποιούν χωρίς καν να τα γνωρίζουν. Αρκετοί ερευνητές έχουν χαρακτηρίσει τα ενσωματωμένα



συστήματα ως επανάσταση των ηλεκτρικών συσκευών. Τα ενσωματωμένα συστήματα αρχικά εμφανίστηκαν σαν συσκευές ή προϊόντα χαμηλών επιδόσεων σε αντίθεση με σήμερα που χρησιμοποιούνται σε αρκετά πολύπλοκες συσκευές με αυξημένες απαιτήσεις και επιδόσεις. Όπως αναφέραμε και στο προηγούμενο κεφάλαιο γνωρίζουμε ότι καθημερινά κατασκευάζονται εκατομμύρια υπολογιστές σε όλον τον κόσμο αλλά αυτό που μάλλον δεν γνωρίζουμε είναι ότι κατασκευάζονται δισεκατομμύρια ενσωματωμένα συστήματα. Αυτό οφείλετε στο ότι δεν το βλέπουμε, διότι όπως το λέει και η λέξη είναι ενσωματωμένα στις ηλεκτρικές συσκευές. Τα ενσωματωμένα συστήματα τα συναντάμε σχεδόν παντού και χρησιμοποιούνται για ειδικευμένες εφαρμογές και έχουν περιορισμένη χρήση. Θα μπορούσαμε να δώσουμε πολλούς ορισμούς για τα ενσωματωμένα συστήματα αλλά ο καλύτερος ορισμός είναι τα παραδείγματα. Μερικά παραδείγματα ενσωματωμένων συστημάτων είναι: Η τηλεόραση και τα DVD, Συσκευές τηλε ελέγχου (Τηλεκοντρόλ), Κινητά και ασύρματα τηλεφώνά, Οχήματα (ηλεκτρονικές πλακέτες, ραδιόφωνα, σύστημα φωτισμού), Οικιακές συσκευές (φούρνοι μικροκυμάτων, πλυντήρια, ηλεκτρικές σκούπες κλπ.), Αυτόματοι πολίτες, Εκτυπωτές και φαξ, Φωτογραφικές μηχανές, βιντεοκάμερες κτλ.[9]

Εικόνα 1.4: ενσωματωμένα συστήματα στην ζωή μας

2.5 ΕΤΑΙΡΙΕΣ ΚΑΙ ΜΟΝΤΕΛΑ

Στο κεφάλαιο αυτό θα δούμε μερικές εταιρίες μικροελεγκτών αναλυτικά από το πότε δημιουργήθηκαν και ποιοι ήταν οι δημιουργοί του, ονομαστικά τους δημιουργούς και κάποια μοντέλα. Ακόμα και τους μεγάλους ανταγωνιστές που έχει το Arduino και ονομαστικά κάποια ονόματα εταιριών.



Εικόνα 1.5: Εταιρίες μικροελεγκτών

ATMEL

Η ATMEL ιδρύθηκε το 1984 από τον George Perlegos. Ο George Perlegos είχε εργαστεί στην ομάδα μνήμης της intel την δεκαετία του 1970, είχε συνεργαστεί με την Seeq Technology για την κατασκευή της μνήμης EPROM των μικροελεγκτών. Η εταιρία εστιάζει σε ενσωματωμένα συστήματα που κατασκευάζονται από τους μικροελεγκτές. Οι μικροελεγκτές της ATMEL εξυπηρετούν εφαρμογές όπως είναι οι επικοινωνίες, η δικτύωση μεταξύ υπολογιστών, σε ηλεκτρικές συσκευές, σε βιομηχανίες, σε ιατρικά μηχανήματα, σε αυτοκίνητα, σε αυτοκινητοβιομηχανίες. Ειδικεύεται σε μικροελεγκτές και σε αισθητήρες αφής ειδικά για ενσωματωμένα συστήματα. Η εταιρία έχει έδρα στην Καλιφόρνια και έχει επίσης παραρτήματα σε πολλές πόλεις ακόμα και στην Ελλάδα. Η ATMEL προμηθεύει τις συσκευές της ως τυποποιημένα προϊόντα, ολοκληρωμένα κυκλώματα ειδικά για εφαρμογές και ειδικά πρότυπα προϊόντα ανάλογα με τις απαιτήσεις των πελατών. Η εταιρία ξεκίνησε με ένα κεφάλαιο ύψους 30.000 δολαρίων μόνο. Τα προϊόντα της ATMEL χρησιμοποιούσαν λιγότερη ενέργεια από τα υπόλοιπα ενσωματωμένα συστήματα που υπήρχαν στην αγορά έτσι συνεργάστηκε με πολλές εταιρίες και αναπτύχθηκε πολύ γρήγορα αλλά μέσα σε όλα αυτά ήρθαν και οι αναποδιές με την εταιρία διότι η INTEL έκανε αγωγή κατά της εταιρίας με δικαιολογία την παραβίαση των πνευματικών δικαιωμάτων. Η Atmel δεν το άφησε έτσι αντί να ασχοληθεί να καταπολεμήσει στα δικαστήρια να αποκτήσει τα πνευματικά δικαιώματα. Η Atmel επανασχέδιασε τα προϊόντα της με διαφορετική πνευματική ιδιοκτησία. Η εταιρία έκανε πολύ μεγάλες και επικίνδυνες επενδύσεις που ξεπερνούσαν το ποσό των 100 εκατομμύριον δολαρίων. Η εταιρία μέχρι και το 2016 αναπτύσσετε με σταθερούς ρυθμούς και είναι γνωστή παγκοσμίως για τα προϊόντα της σε όλον τον κόσμο. Στο τέλος του 2016 εξαγοράστηκε από την microchip έναντι του ποσού των 3,6 δισεκατομμυρίων δολαρίων.[10]

MICROCHIP

Η MICROCHIP TECHNOLOGY είναι μια αμερικάνικη εταιρία. Η εταιρία έχει την βάση της στο Τσαντλερ της Αριζόνα κατασκευάζει μικροελεγκτές, μνήμες και αναλογικούς ημιαγωγούς τα προϊόντα της περιέχουν σύγχρονους μικροελεγκτές και συσκευές μνήμης EEPROM. Η Microchip δημιουργήθηκε το 1987 ως ένα τμήμα της General Instrument η Microchip έγινε ανεξάρτητη το 1989 όταν αποκτήθηκε από μια ομάδα επιχειρηματικών κεφαλαίων και το 1993 έγινε δημόσια. Η εταιρία έκανε ραγδαία ανάπτυξη με την δημιουργία ενός μοντέλου το οποίο είχε χαμηλότερη λειτουργία ύπνου παγκοσμίως το μοντέλο αυτό ήταν το nanoWattXLP. Το μοντέλο αυτό έφτασε το υπέρτατο ποσοστό πλύσεων αναφέροντας παραπάνω από 6 δισεκατομμύρια μικροελεγκτές το 2009. Η εταιρία όμως δεν σταμάτησε εκεί, συνέχισε την ραγδαία ανάπτυξη της και την εξαγορά ιδεών και συστημάτων ακόμα και ολόκληρες εταιρίες. Η πιο μεγάλη επένδυση έγινε το 2016 όπως αναφέραμε και παραπάνω με την αγορά της μεγάλης εταιρίας Atmel έναντι του ποσού 3,6 δισεκατομμυρίων δολαρίων.

Είναι μια από τις πιο κερδοφόρες εταιρίες μικροεπεξεργαστών και απασχολεί πάνω από 10.000 εργαζομένους παγκοσμίως [11]

TEXAS INSTRUMENTS

Η Texas Instruments δημιουργήθηκε το 1951 από τους Cecil H. Green, Eugene McDermott και τον Patrick E. Haggerty. Ο McDermott ήταν ένας από τους δημιουργούς της Geophysical Service Inc. Η εταιρία GSI ιδρύθηκε το 1930 για να χρησιμοποιήσει τη διάσπαση και την αντανάκλασή της σεισμολογίας για να διερευνήσει τις καταθέσεις πετρελαίου. Την εταιρία αυτή την αγόρασαν ο Green και οι εργαζόμενοι που ήταν ο Jonsson και ο Green το 1941. Στην πορεία η εταιρία μετονομάστηκε το 1951 σε Texas Instruments. Η Texas Instruments είναι μια αμερικάνικη εταιρία η οποία σχεδιάζει και κατασκευάζει και ημιαγωγούς τους οποίους εξάγει παγκοσμίως. Η έδρα του είναι το Ντάλας στο Τέξας . Επιπλέον, ένα ποσοστό μεγαλύτερο του 85% των εσόδων προέρχεται από τα ενσωματωμένα συστήματα και των αναλογικών καρτών. Η Texas Instruments μέχρι και σήμερα έχει 43.000 χιλιάδες διπλώματα ευρεσιτεχνίας παγκοσμίως. Το 1971 εφευρέθηκε ο μικροϋπολογιστής ενός τσιπ και έλαβε το πρώτο δίπλωμα ευρεσιτεχνίας σε έναν μικροεπεξεργαστή ενός μικροκυκλώματος. Η Texas Instruments προσφέρει επίσης ένα χαρτοφυλάκιο μικροελεγκτών όπως ο MSP430 που έχει χαμηλό κόστος και χαμηλή κατανάλωση ενέργειας και MCU γενικής χρήσης 16bit για χρήση σε ενσωματωμένες εφαρμογές.

Όπως και ο MSP432 ο οποίος έχει τις ίδιες δυνατότητες με τον MSP 430 αλλά είναι 32 bit με καλύτερη απόδοση. Η εταιρία έχει πάρα πολλά είδη και είναι από τις μεγαλύτερες εταιρίες μικροελεγκτών και μικροεπεξεργαστών.[12]

Μερικοί Κατασκευαστές Μικροελεγκτών Ακόμα

SILICON LABS

RENESAS TECHNOLOGY CORP

INTEL

DALLAS SEMICONDUCTOR

FUJITSU SEMICODUCTOR EUROPE

STMICROELECTRONICS

FREESCALE SEMICONDUCTOR

ARM

EPSON

HITACHI

MAXIM

NEC

TOSHIBA

NXP

INFINEON [13] [14] [15]

3 ΑΝΑΦΟΡΑ ΓΙΑ ΤΟ ARDUINO

3.1 ΤΙ ΕΙΝΑΙ ΤΟ ARDUINO



Εικόνα 2.1: Απεικόνιση συμβολισμού εταιρίας Arduino

Το Arduino είναι μια οικογένεια μικροελεγκτών. Τον κάθε μικροελεγκτή μπορούμε να τον χαρακτηρίσουμε και ως μια μητρική πλακέτα ανοικτού κώδικα η οποία έχει ενσωματωμένο έναν μικροεπεξεργαστή, μια μνήμη και μια είσοδο / έξοδο. Με αυτόν τον τρόπο μπορεί να εκτελέσει λειτουργίες όπως ένας μικρός υπολογιστής (για παράδειγμα με το πάτημα ενός κουμπιού μπορεί να κάνει ένα μοτέρ να μπει σε λειτουργία). Τις ενέργειες αυτές το Arduino μπορεί να τις υλοποιήσει μέσω του περιβάλλοντος Arduino IDE αλλά όχι μόνο. Μπορεί να το καταφέρει και μέσω κάποιου άλλου προγραμματιστικού περιβάλλοντος που μπορούμε να δούμε στο ακριβώς επόμενο κεφάλαιο ανάλογα με την επιθυμία του κάθε χρήστη. Το Arduino είναι ένας εύκολος και ευχάριστος τρόπος να μάθει κάποιος να προγραμματίζει, αν δεν γνωρίζει προγραμματισμό και φυσικά κάποιος που γνωρίζει θα μάθει κάποια σημαντικά πράγματα για την γλώσσα wiring και τις βιβλιοθήκες που διαθέτει. Επιπλέον, θα του είναι πολύ πιο εύκολο στην κατανόηση του μικροελεγκτή αναφέροντας πάντα το προγραμματιστικό κομμάτι. Οι μικροελεγκτές Arduino είναι πολύ γνωστοί και πολύ αναγνωρισμένοι στον κλάδο, διότι έχουν δημιουργήσει πολλά διαφορετικά έργα από τα πιο απλά μέχρι και τα πιο σύνθετα. Για παράδειγμα τηλεκατευθυνόμενα οχήματα, ρομπότ και πολλές άλλες κατασκευές. Το πρώτο σχέδιο Arduino δημιουργήθηκε το 2005 με ιδρυτές τον Massimo Banzi και David Cueartielles. Σκοπός ήταν να δημιουργήσουν οι μαθητές πιο φθηνές πλακέτες από αυτές που υπήρχαν στην αγορά. Το σχέδιο αυτό ονομάστηκε Arduinos της Ivrea, όπου Ivrea ήταν η πόλη που υλοποιήθηκε αυτή η δημιουργία και ξεκίνησαν να παράγουν πλακέτες. Συγκεκριμένα σε ένα μικρό εργοστάσιο που βρίσκονταν κοντά στο Τορίνο στην βορειοδυτική Ιταλία. Η εταιρία διαθέτει μια πολύ μεγάλη γκάμα μοντέλων μικροελεγκτών, περιφερειακών χαρακτηριστικών και πλακετών όπως είναι τα Wi-Fi τα Bluetooth τα τηλεχειριστήρια κλπ. Είναι από τις πιο οικονομικές πλακέτες στην αγορά με πολύ μεγάλο εύρος επιλογών έτσι ώστε να μπορεί κάποιος

ευκολά να τις αποκτήσει και να κατασκευάσει αυτό που ακριβώς θέλει. Υπάρχουν ακόμα πάρα πολλές αναλυτικές πληροφορίες στο διαδίκτυο και κατασκευές που μπορεί να μάθει να τις κατασκευάζει ο καθένας. Αρκεί να μπορεί κανείς να αφιερώσει αρκετό χρόνο και να έχει επιμονή. Παρακάτω θα παρουσιαστούν και τρόποι για να αποφευχθούν κάποια προγραμματιστικά προβλήματα όπως και τον τρόπο που θα εισαχθούν βιβλιοθήκες στο πρόγραμμα. Το λογισμικό του Arduino είναι πλήρως συμβατό με όλα τα λειτουργικά συστήματα όπως είναι τα Windows, τα Mac και τα Linux σε αντίθεση με τις άλλες εταιρίες που περιορίζονται μόνο στο λειτουργικό σύστημα των Windows.[16]

3.2 ARDUINO UNO



Εικόνα 2.2: Arduino Uno

Το Arduino UNO είναι μια πλακέτα μικροελεγκτή η οποία είναι βασισμένη στον επεξεργαστή της Atmel ATmega 328P. Το Arduino Uno μπορεί να τροφοδοτηθεί με συνεχές ρεύμα (direct current DC) είτε αυτό το ρεύμα έρθει από τον υπολογιστή μέσω USB είτε από κάποια εξωτερική τροφοδοσία μέσω της μαύρης υποδοχής που βρίσκετε δίπλα ακριβώς από την είσοδο USB. Επίσης, πρέπει να αναφέρουμε ότι η εξωτερική τροφοδοσία πρέπει να έχει εύρος από 7V μέχρι 12V. Τέλος, διαθέτει ακροδέκτες τους οποίους αναλύουμε παρακάτω.

3.2.1 ΤΕΧΝΙΚΕΣ ΠΡΟΔΙΑΓΡΑΦΕΣ ARDUINO UNO

[17]

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

Εικόνα 2.3: Τεχνικές προδιαγραφές

3.2.2 ΑΚΡΟΔΕΚΤΕΣ ΤΡΟΦΟΔΟΣΙΑΣ

Οι ακροδέκτες τροφοδοσίας στο Arduino Uno είναι οι ακόλουθοι:

V_{in}: Η τάση εισόδου της πλακέτας όταν χρησιμοποιείτε εξωτερική πηγή ενέργειας

5V: Η τάση που χρησιμοποιείτε από διάφορα μέρη της πλακέτας και τον μικροελεγκτή είναι 5V. Η τάση η οποία δίνει αυτός ο ακροδέκτης είναι είτε η τάση 5V που δίνει η σύνδεση με USB είτε επίσης η ρυθμιζόμενη τάση που δίνεται μέσω του V_{in}.

GND: είναι οι είσοδοι γείωσης.

Όπως θα δούμε και παρακάτω στην εικόνα (2.4).



Εικόνα 2.4: Ακροδέκτες τροφοδοσίας Arduino Uno

Ο μικροεπεξεργαστής ATmega328 έχει τρεις ομάδες μνήμης. Διαθέτει flash memory, στην οποία αποθηκεύονται τα Arduino sketch τα οποία θα αναλυθούν εκτενέστερα σε επόμενο κεφάλαιο. Ακόμα διαθέτει SRAM(static random access memory), στην οποία δημιουργείται το sketch και χρησιμοποιεί τις μεταβλητές όταν αυτό τρέχει. Επιπλέον διαθέτει και τη μνήμη EEPROM την οποία χρησιμοποιούν οι προγραμματιστές για την αποθήκευση μακροχρόνιων πληροφοριών. Τα προγράμματα για να αποθηκεύουν μεταβλητές είτε πίνακες κλπ, χρησιμοποιούν μνήμη SRAM 2KB. Η μνήμη αυτή χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή πατηθεί το κουμπί της επανεκκίνησης. Ακόμα, το 1KB μνήμης EEPROM μπορεί να χρησιμοποιηθεί για εγγραφή είτε για ανάγνωση δεδομένων από τα προγράμματα. Σε αντίθεση με την μνήμη SRAM η μνήμη EEPROM δεν χάνει τα περιεχόμενα της με απώλεια της τροφοδοσίας είτε με την επανεκκίνηση πατώντας το κουμπί επανεκκίνησης. Το Arduino Uno έχει 32KB μνήμης Flash από τα οποία τα 2KB χρησιμοποιούνται από το firmware του Arduino που είναι ήδη εγκατεστημένο από τον κατασκευαστή του. Το firmware είναι αναγκαίο για την εγκατάσταση προγραμμάτων μέσω της θύρας USB στον μικροελεγκτή. Τα υπόλοιπα 30KB της μνήμης Flash χρησιμοποιούνται για την αποθήκευση των προγραμμάτων αφού βέβαια μεταγλωττιστούν στον υπολογιστή. Η μνήμη Flash δεν χάνει τα περιεχόμενα που διαθέτει από την απώλεια ρεύματος είτε επανεκκίνησης.

Η πλακέτα του Arduino Uno διαθέτει 14 ψηφιακούς ακροδέκτες που μπορούν να χρησιμοποιηθούν για είσοδο και έξοδο ψηφιακών τιμών. Το Arduino όπως αναφέραμε και παραπάνω έχει 5V τάση στους ακροδέκτες με αποτέλεσμα αν ένας ακροδέκτης εισόδου φέρει τάση 5V να διαβαστεί ως λογικό 1 ενώ αν όχι να διαβαστεί ως 0. Έτσι και αντίστοιχα ο ακροδέκτης εξόδου γράφει το λογικό 1 ως τάση των 5V ενώ 0 όταν αντιστοιχεί στην γείωση. Εκτός από τη γενική λειτουργία των ακροδεκτών εισόδου και εξόδου κάποιοι ακροδέκτες έχουν επιπρόσθετες λειτουργίες όπως θα δούμε και παρακάτω στην εικόνα 2.5, καθώς και την λειτουργία του καθενός (αναλυτικά). Οι **ακροδέκτες 0 και 1** λειτουργούν ως RX και TX της σειριακής θύρας όταν το πρόγραμμα ενεργοποιεί τη σειριακή θύρα. Έτσι όταν το πρόγραμμα στέλνει δεδομένα στη σειριακή θύρα αυτά προωθούνται και στην θύρα USB μέσω του ελεγκτή Serial-Over-USB, αλλά και στον ακροδέκτη 0 για να διαβάσει μια άλλη συσκευή. Αν στο πρόγραμμα ενεργοποιηθεί το σειριακό interface αυτό θα έχει ως αποτέλεσμα να χάσει 2 ψηφιακές εισόδους/εξόδους της πλακέτας

μας. Οι ακροδέκτες **2 και 3** λειτουργούν και ως εξωτερικά interrupt. Οι ρυθμίσεις αυτές γίνονται μέσα από το πρόγραμμα ώστε να λειτουργούν αποκλειστικά ως ψηφιακές είσοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές αυτό έχει σαν αποτέλεσμα να σταματάει η κανονική ροή του προγράμματος άμεσα και να εκτελείται μια συγκεκριμένη συνάρτηση. Επίσης, τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα για εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας. Οι ακροδέκτες **3,5,6,10 και 11** μπορούν να λειτουργήσουν και ως ψευδό αναλογικές έξοδοι με το σύστημα PWM(Pulse Width Modulation).



Εικόνα 2.5: Ψηφιακές είσοδοι Arduino Uno

Από την άλλη πλευρά της πλακέτας θα δούμε μια σήμανση η οποία διαθέτει 6 pin εν μονομάτης ANALOG IN όπως θα δούμε και στο σχήμα(2.5) παρακάτω.



Εικόνα 2.6: Ακροδέκτες τροφοδοσίας power (αριστερά) αναλογική είσοδοι analog in (δεξιά) στην πλακέτα Arduino Uno

Τα pin αυτά είναι αριθμημένα από το 0 μέχρι το 5. Το κάθε κανάλι μπορεί να χρησιμοποιηθεί ξεχώρισα, το κάθε κανάλι έχει την ικανότητα διαχείρισης 10bit δηλαδή 1024 τιμές με λίγα λόγια μπορεί να διαιρεί την τάση αναφοράς σε 1024 εύρη. Η τάση αναφοράς μπορεί να ρυθμιστεί με μια εντολή η οποία είναι η (analogReference()) στο 1.1V, ενώ η προεπιλογή που έχει είναι στα 5V. [18]

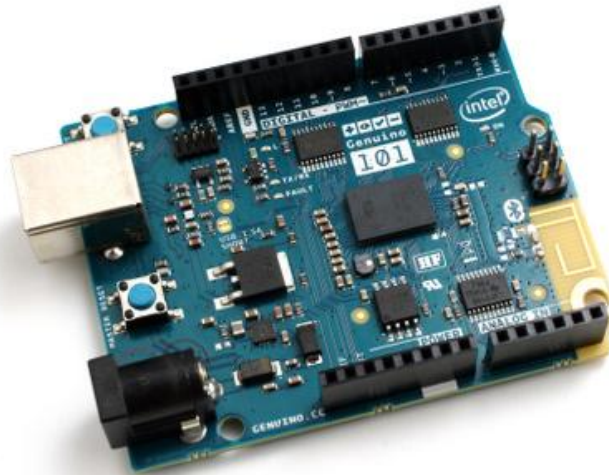
3.3 ΟΙΚΟΓΕΝΕΙΑ ΤΟΥ ARDUINO

3.3.1 ARDUINO LEONARDO



Το Arduino Leonardo είναι μια πλακέτα μικροελεγκτή η οποία είναι βασισμένη στον επεξεργαστή της Atmel ATmega 32u4. Περιέχει 20 ακροδέκτες εισόδου/εξόδου εκ των οποίων 7 από αυτές μπορούν να χρησιμοποιηθούν ως έξοδοι PWM και 12 όπως αναλογικές εισόδους, μια είσοδο micro USB, διαθέτει ακόμα χρονισμένη ταχύτητα μεταδόσεις (clock speed) 16MHz, μια αποδοχή ρεύματος, μια κεφαλίδα ISCP και ένα κουμπάκι επαναφοράς .

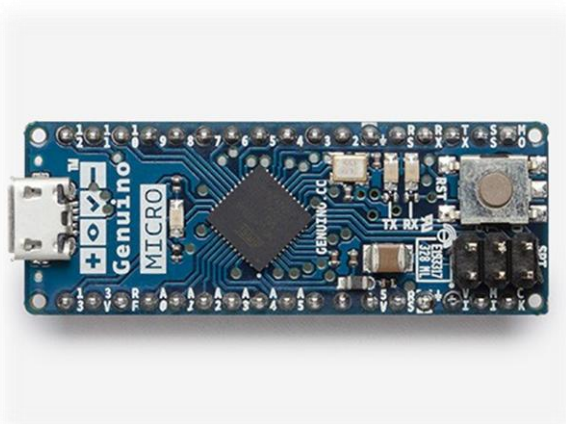
3.3.2 ARDUINO 101



Εικόνα 2.8: Arduino 101

Το Arduino 101 είναι ένας μικροελεγκτής ο οποίος έχει μια βασική διαφορά από τους άλλους της οικογένειάς του, έχει δυο πυρήνες της intel ένα x86(Quark) και έναν 32-bit ARC με χρονισμένη ταχύτητα μεταδόσεις στα 32MHz. Το Arduino 101 περιέχει 14 ακροδέκτες εισόδου-εξόδου, από αυτές οι 4 μπορούν να χρησιμοποιηθούν ως έξοδοι PWM, 6 αναλογικές εισόδους, μια θύρα USB για σειριακή επικοινωνία, μια ICSP κεφαλίδα με σήματα SPI. Η τάση λειτουργίας εισόδου εξόδου είναι 3.3V, αλλά όλοι οι ακροδέκτες προστατεύονται από υπέρταση 5V.

3.3.3 ARDUINO MICRO

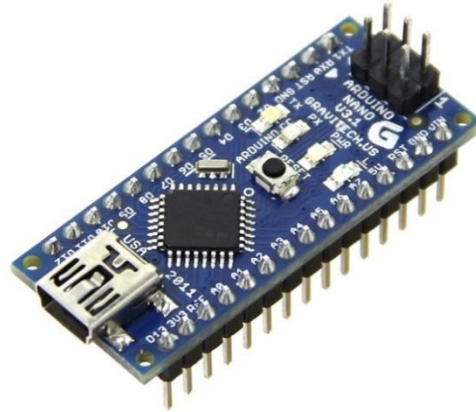


Εικόνα 2.9: Arduino Micro

Το Arduino Micro είναι μια μικρή πλακέτα μικροελεγκτή η οποία διαθέτει έναν επεξεργαστή Atmel τον ATmega32u4. Έχει 20 ακροδέκτες εισόδου-εξόδου από τους οποίους 7 μπορούν να χρησιμοποιηθούν ως έξοδοι PWM και οι 12 για αναλογικές

εισόδους, ακόμα χρονισμένη ταχύτητα μεταδόσεις (clock speed) 16MHz, μια είσοδο micro USB, μια κεφαλίδα ICSP και ένα κουμπί επαναφοράς. Έχει σχεδιαστεί έτσι για να εφαρμόζει άψογα σε breadboard.

3.3.4 ARDUINO NANO



Εικόνα 2.10: Arduino Nano

Το Arduino Nano είναι ένας πλήρης μικροελεγκτής ο οποίος είναι ιδιαίτερα φιλικός στο breadboard διαθέτει έναν επεξεργαστή Atmel τον ATmega328. Έχει λίγο πολύ τα ίδια χαρακτηριστικά που έχει και μικροελεγκτής Arduino Duemilanove αλλά σε διαφορετική συσκευασία. Μια διαφορά είναι η τροφοδοσία του η οποία γίνεται με micro USB.

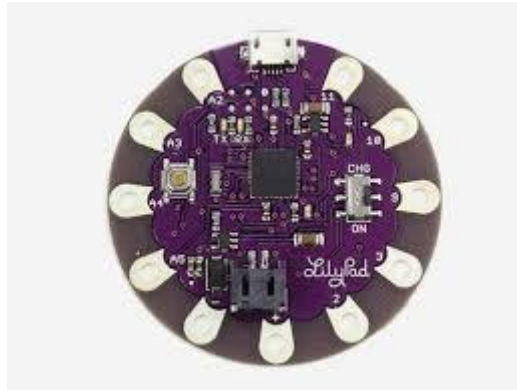
3.3.5 ARDUINO MEGA



Εικόνα 2.11: Arduino Mega

Το Arduino mega έχει σχεδιαστεί για πιο μεγάλα και πιο περίπλοκα έργα διότι διαθέτει 54 ψηφιακούς ακροδέκτες εισόδου και εξόδου και ακόμα 16 αναλογικές εισόδους. Ο μικροελεγκτής αυτός είναι κατασκευασμένος για ρομποτικές εφαρμογές και για 3D εκτυπωτές. Έχει και αυτός ακριβώς τα ίδια χαρακτηριστικά με τους άλλους μικροελεγκτές Arduino (όπως είναι η είσοδος του USB, το κουμπάκι επανεκκίνησης και είσοδο τροφοδοσίας).

3.3.6 ARDUINO LILYPAD



Εικόνα 2.12: Arduino
Lilypad

Το Arduino LilyPad είναι κάτι σε τελείως διαφορετική μορφή όπως βλέπουμε και στην εικόνα έχει μια σφαιρική εικόνα. Διαθέτει έναν μικροεπεξεργαστή ATmega32u4. Ακόμα διαθέτει 9 ψηφιακούς ακροδέκτες από τους οποίους, οι 4 μπορούν να χρησιμοποιηθούν σαν αναλογικές εξόδους. Διαθέτει ακόμα μια σύνδεση microUSB και μια υποδοχή JST για την εισαγωγή μπαταρίας LiPo 3.7V.

3.3.7 ARDUINO ETHERNET



Εικόνα:2.13:Arduino Ethernet

Το Arduino ethernet είναι ένας μικροελεγκτής ο οποίος έχει ενσωματωμένο έναν μικροεπεξεργαστή AT mega 328P ο οποίος όμως έχει κάτι διαφορετικό από τα προηγούμενα που έχουμε αναφέρει και αυτό είναι η είσοδος ethernet ή αλλιώς RJ45. Διαθέτει 14 ψηφιακές ακίδες εισόδου και εξόδου και 6 αναλογικές εισόδους. Ακόμα εισόδους τροφοδοσίας και ένα κουμπάκι επαναφοράς.

Στην σελίδα του Arduino μπορείτε να δείτε πιο αναλυτικά τα μοντέλα και τις δυνατότητες που διαθέτει το κάθε μοντέλο καθώς και που μπορούμε να χρησιμοποιήσουμε τον κάθε μικροελεγκτή. Επιπλέον τις πιο καινούριες εκδόσεις και την επεξεργαστική ισχύ ακόμα και τις βιβλιοθήκες για το κάθε μοντέλο αναλυτικά.[19]

4 ΠΩΣ ΠΡΟΓΡΑΜΜΑΤΙΖΕΤΑΙ ΤΟ ARDUINO

Στο κεφάλαιο αυτό θα δούμε πως μπορούμε να προγραμματίσουμε μια πλακέτα Arduino και πιο συγκεκριμένα το προγραμματιστικό του περιβάλλον (και όχι μόνο). Θα μάθουμε τις γλώσσες με τις οποίες μπορούμε να προγραμματίσουμε την πλακέτα και κάποιες πληροφορίες για τις γλώσσες προγραμματισμού. Επιπλέον, θα αναφερθούμε στις βιβλιοθήκες, τι είναι, που μας χρησιμεύουν αλλά και πως τις κάνουμε εγκατάσταση.

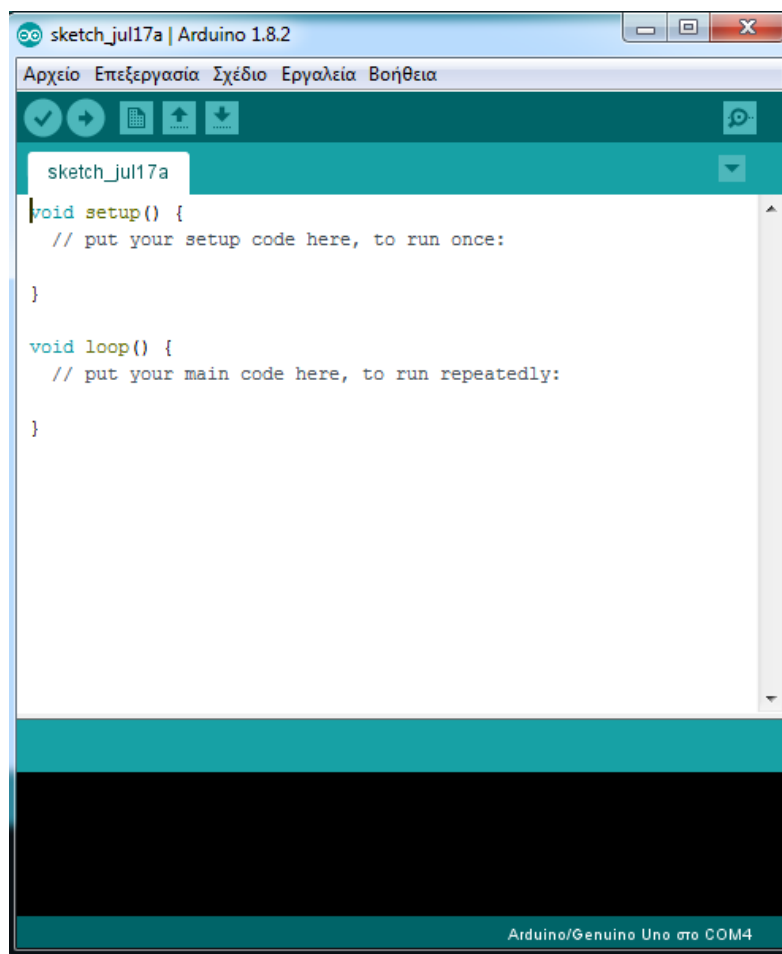
ΠΛΑΤΦΟΡΜΑ ΠΟΥ ΧΡΗΣΗΜΟΠΟΙΟΥΜΕ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ

Εδώ θα δούμε γενικά χαρακτηριστικά για την πλατφόρμα που εμείς χρησιμοποιούμε. Όπως για παράδειγμα: τι γλώσσα χρησιμοποιεί και γενικά χαρακτηριστικά όπως και τις κινήσεις που μπορούμε να κάνουμε στο προγραμματιστικό περιβάλλον.

4.1 ARDUINO IDE

Το ολοκληρωμένο περιβάλλον ανάπτυξης IDE για το Arduino εκτός από ότι είναι το πρόγραμμα που προγραμματίζουμε την πλακέτα μας, είναι ακόμα ένα ευχάριστο περιβάλλον για τους χρήστες και ειδικά τους αρχάριους που δεν είναι εξοικειωμένοι με την ανάπτυξη λογισμικού. Το περιβάλλον ανάπτυξης Arduino περιέχει ένα πρόγραμμα επεξεργασίας κειμένου, για την ανάπτυξη του κώδικα. Επίσης, συνδέεται με το hardware μέρος του Arduino για να φορτώσει προγράμματα και να επικοινωνήσει μαζί τους. Ο κώδικας που έχει γραφεί για το Arduino ονομάζεται sketch. Παρακάτω θα δούμε το πρόγραμμα Arduino IDE.



Εικόνα 3.1: προγραμματιστικό περιβάλλον Arduino IDE

Είναι ένα πρακτικό περιβάλλον για την συγγραφή προγραμμάτων έχοντας χρωματική σήμανση και έτοιμες βιβλιοθήκες για επέκταση τις οποίες θα ερευνήσουμε παρακάτω μαζί με τον τρόπο που μπορούμε να τις εγκαταστήσουμε. Ακόμα, τις λειτουργίες που κάνει ο compiler για την μεταγλώττιση των sketch. Sketch είναι ο προγραμματιστικός κώδικας που έχουμε γράψει σε ένα αρχείο, το οποίο αρχείο είναι η σελίδα που γράφουμε τον κώδικα. Διαθέτει μια σειριακή οθόνη η οποία παρακολουθεί τις επικοινωνίες του USB, αναλαμβάνει να στείλει αλφαριθμητικά στο Arduino μέσω του USB και είναι χρήσιμο για την αποφυγή σφαλμάτων των sketch, όπως επίσης το ανέβασμα των μεταγλωτισμένων sketch στο Arduino. Παρακάτω θα δούμε αναλυτικά τα εργαλεία που έχει το περιβάλλον ανάπτυξης Arduino IDE υπό την μορφή κουμπιών.



(Verify) Ελέγχει για συντακτικά λάθη στον κώδικα.



(Upload) Μεταγλωττίζει τον κώδικα και τον φορτώνει στο Arduino.



(New) Δημιουργεί ένα νέο sketch.



(Open) Παραθέτει ένα μενού με όλα τα sketch για άνοιγμα σε νέο παράθυρο.



(Save) Αποθηκεύει ένα sketch.



(Serial Monitor) Ανοίγει την σειριακή οθόνη.

Το Arduino IDE είναι μια εφαρμογή γραμμένη σε Java, λειτουργεί σε πολλές πλατφόρμες και η γλώσσα προγραμματισμού που χρησιμοποιεί είναι η Wiring. Η γλώσσα Wiring είναι μια παραλλαγή της γλώσσας C και C++ για μικροελεγκτές αρχιτεκτονικής AVR όπως και ο ATmega. Υποστηρίζει όλες τις βασικές δομές της C καθώς και μερικά χαρακτηριστικά της C++ (όπως for, case, if κλπ). Ο compiler που χρησιμοποιείται είναι ο AVR gcc και ως βιβλιοθήκη C χρησιμοποιείτε η AVR libc. Λόγω της καταγωγής της, από την γλώσσα C, στην γλώσσα του Arduino μπορούν να χρησιμοποιηθούν οι ίδιες βασικές εντολές και συναρτήσεις με την ίδια ακριβώς σύνταξη, τους ίδιους τύπους δεδομένων και τους ίδιους τελεστές όπως στην γλώσσα C. Υπάρχουν ακόμα κάποιες ειδικές εντολές συνάρτησης και σταθερές που μας βοηθούν για την διαχείριση του ιδιού hardware του Arduino.

Τα προγράμματα Arduino διαιρούνται σε τρία μέρη τα οποία είναι η δομή (structure), οι τιμές (values) και οι συναρτήσεις (functions). Ένα σημαντικό στοιχείο για την συγγραφή προγραμμάτων στο Arduino αποτελεί η αποσφαλμάτωση του συστήματος είτε αυτό είναι λογισμικό είτε αυτό είναι υλικό. Πολλές φορές θέλουμε να αποσφαλμάτωσης το κύκλωμα μας για να λειτουργήσει το κομμάτι μας σωστά. Για να το καταφέρουμε αυτό χρησιμοποιούμε την σειριακή επικοινωνία σε συνδυασμό με εντολές εκτύπωσης στο σειριακό τερματικό.

Πρώτα εμφανίζουμε την σειριακή οθόνη πατώντας το εικονίδιο με το μεγεθυντικό φακό στην εργαλειοθήκη με τα εικονίδια, το οποίο βρίσκετε τέρμα δεξιά.

Στην συνέχεια ρυθμίζουμε την ταχύτητα σε 9600 bps

Στο sketch του Arduino στο `steup()` τοποθετούμε τη γραμμή `Serial.begin(9600);`

Στο σημείο που θέλουμε να εμφανίσουμε κάποια τιμή δίνουμε `Serial.println(τιμή);` Η σκέτο `Serial.print(τιμή);`

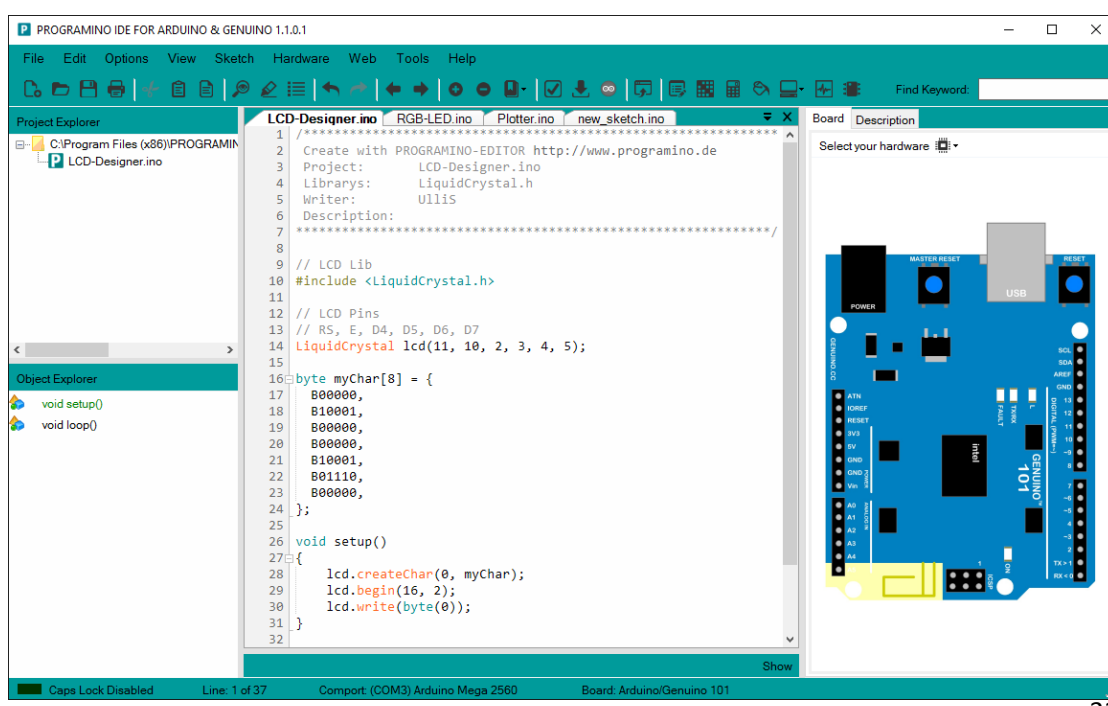
Με τον τρόπο αυτόν μπορούμε να αποσφαλματώσουμε όλο το κύκλωμα. Εκτυπώνουμε τις τιμές από τις εισόδους και τις τιμές που στέλνουμε στις εξόδους και αυτό μας βοηθά να προσδιορίσουμε το πρόβλημα.[20]

4.2 ΠΛΑΤΦΟΡΜΕΣ ΜΕ ΤΙΣ ΟΠΟΙΕΣ ΜΠΟΡΟΥΜΕ ΝΑ ΠΡΟΓΡΑΜΑΤΙΣΟΥΜΕ ΤΗΣ ΠΛΑΚΕΤΕΣ

Σε αυτή την παράγραφο θα δούμε διαφορετικά προγραμματιστικά περιβάλλοντα καθώς και εικόνες αναλυτικά για το κάθε ένα. Επιπλέον, το λειτουργικό σύστημα που μπορεί να χρησιμοποιήσει το κάθε πρόγραμμα όπως και κάποια ιδιαίτερα χαρακτηριστικά.

4.2.1 PROGRAMINO IDE

Το πρόγραμμα IDE PROGRAMINO είναι ένα αναλυτικό IDE χαμηλού επιπέδου για το Arduino και το Genuino. Το IDE είναι σχεδιασμένο για δημιουργούς και επαγγελματίες που εργάζονται πάνω στις πλακέτες αυτές και είναι πλήρως συμβατό με το Arduino. Η HTML5 διευκολύνει την απευθείας σύνδεση και εμφάνιση των αποτελεσμάτων στο Arduino. Δίπλα από το πρόγραμμα υπάρχει μια εικονική πλακέτα που εμφανίζονται οι ενέργειες του κώδικα που γράφουμε. Πέραν του ότι είναι πλήρες συμβατό με το Arduino, είναι πολύ εύχρηστο. Οι υποστηριζόμενες μορφές αρχείων και γλώσσες προγραμματισμού είναι : Arduino, C, C++, headers, HTML, JavaScript, CSS και Text. Το περιβάλλον του IDE είναι μεταφρασμένο στα Αγγλικά και τα Γερμανικά. Υποστηρίζει όλες τις βιβλιοθήκες του Arduino, καθώς και αυτόματης συμπλήρωσης . Ακόμα έχει υποδείξεις και πληροφορίες για τις εντολές Arduino. Διαθέτει επίσης δυο σειριακά τερματικά και πολλά άλλα όπου μπορούμε να τα δούμε

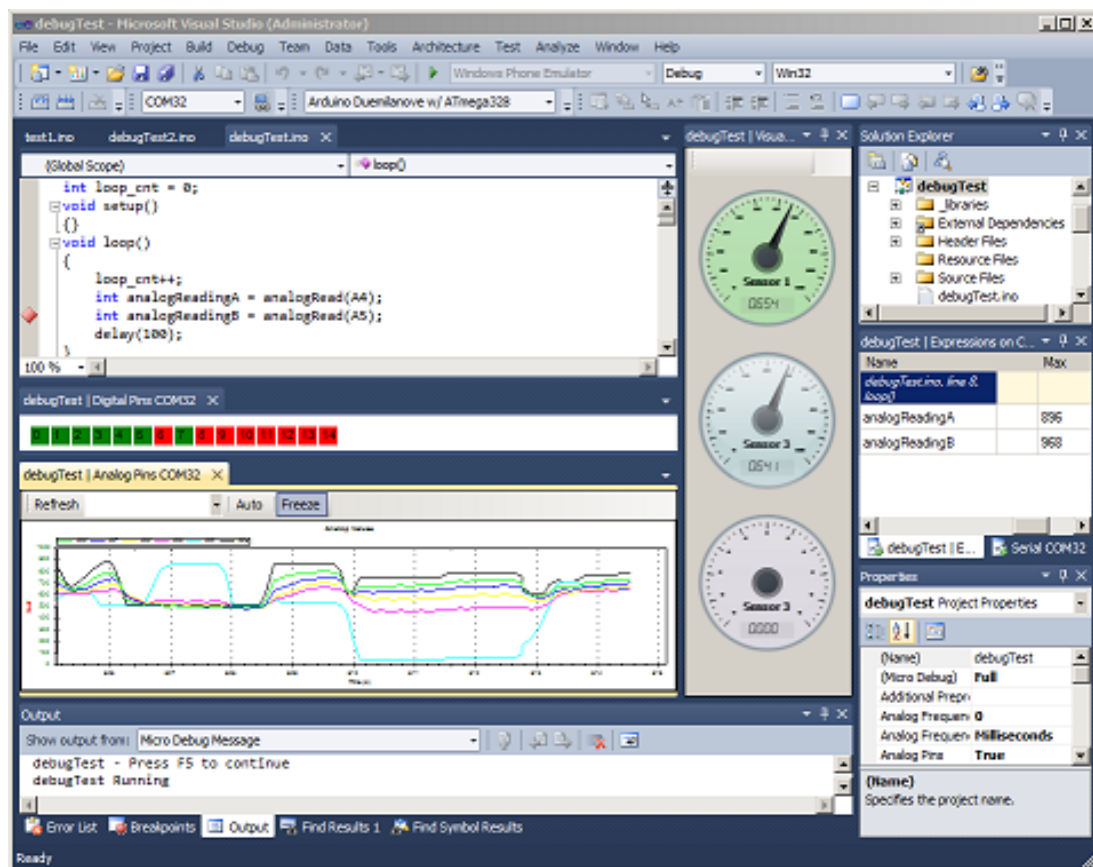


Εικόνα 3.2: Programino Ide

πιο αναλυτικά αν ελέγξετε στην ιστοσελίδα του PROGRAMINO IDE.[21.1]

4.2.2 ARDUINO ΓΙΑ VISUAL STUDIO

Το Visual Micro είναι μια εύχρηστη επέκταση στο Microsoft Visual Studio, από τα κορυφαία IDE στην πλατφόρμα Windows. Έτσι ενσωματώνεται εύκολα στο Visual Studio και είναι έτοιμο για να ξεκινήσει ο προγραμματισμός στο Arduino. Είναι πλήρως συμβατό με το IDE Arduino. Αποτελεί ένα εξαιρετικά προσαρμοσμένο περιβάλλον ανάπτυξης. Υπάρχει υποστήριξη Intellisense, δηλαδή έχει αυτόματη συμπλήρωση και ονόματα κλάσεων. Καθώς κάποιος πληκτρολογεί πραγματοποιείται ένας σύντομος έλεγχος σύνταξης. Έχει την δυνατότητα εργασίας με πολλές πλακέτες ταυτόχρονα ενώ με το πάτημα ενός κουμπιού μπορεί να δημιουργήσει νέες βιβλιοθήκες για το Arduino. Υποστηρίζετε από τις εκδόσεις Visual Studio 2012,2013 και 2015.[21.2]



Εικόνα 3.3:Arduino για Visual Studio

4.2.3 PLUTO – Πρωτότυπο Arduino στην Python

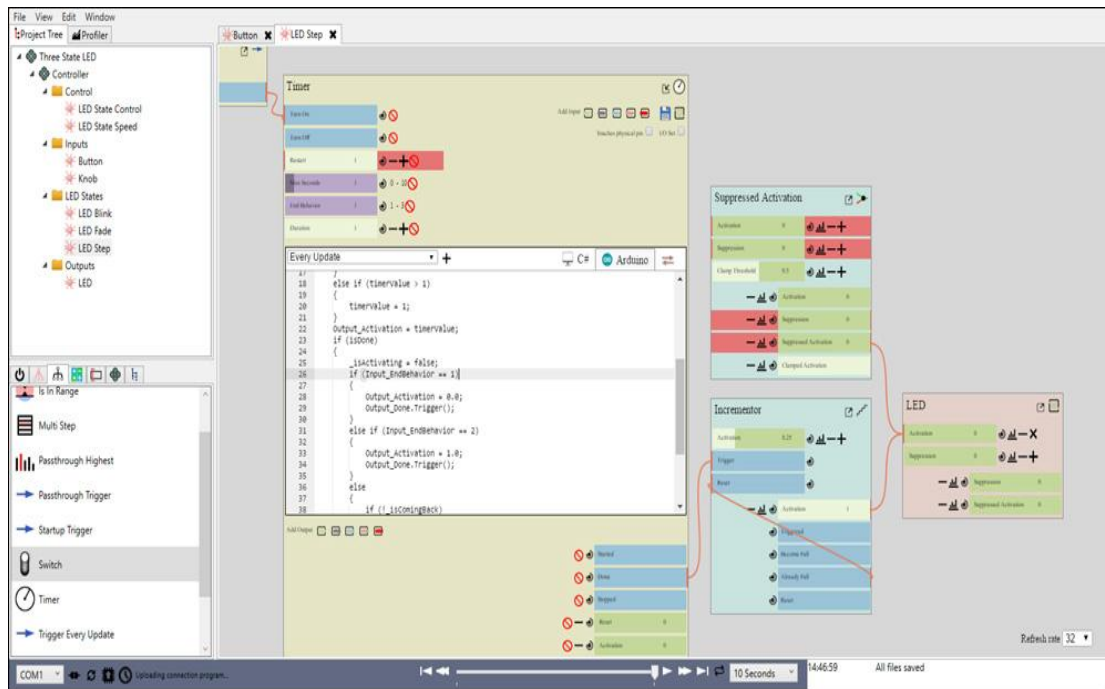
Το Pluto είναι μια βιβλιοθήκη Python χτισμένη στην κορυφή του pyFirmata αλλά με απλούστερα, πιο έξυπνα API και άλλες μεθόδους για γρήγορο πρότυπο. Η Python είναι η απλούστερη γλώσσα που μαθαίνει το Pluto με τον τρόπο αυτό σκοπεύει να βοηθήσει τους αρχάριους, τους σπουδαστές και τους εκπαιδευτικούς να μάθουν τόσο για τον προγραμματισμό αλλά και τον τρόπο με τον οποίο δουλεύει η πλακέτα, καθώς και τους σχεδιαστές που εργάζονται με αυτό και τροποποιούν τις ιδέες τους χωρίς να χρειάζεται να φορτώσουν και να συντάξουν τον κώδικα κάθε φορά.[21.3]

```
>>> from pluto import *
>>> board = Board()
>>> board.led(13).on()
>>> board.led(13).off()
>>> board.pin(13).high()
>>> board.pin(13).low()
>>> board.pin(13).alternate()
```

Εικόνα 3.4: Πρωτότυπο Arduino στην Python(PLUTO)

4.2.4 EMBRIO

Το Embrio είναι ένα οπτικό περιβάλλον ανάπτυξης σε πραγματικό χρόνο για τον προγραμματισμό Arduino. Ο οπτικός προγραμματισμός είναι ιδανικός για ορισμένα πράγματα, όπως η ροή προγραμμάτων και η λογική, σε αντίθεση βέβαια ο κώδικας για τους άλλους αλληλοεπιδρά με το υλικό έτσι βλέπεις άμεσα την κάθε ενέργεια που γίνεται. Το Embrio μπορεί να σου δώσει την δυνατότητα να γράψεις κώδικα απευθείας σε έναν κόμβο και στην συνέχεια χρησιμοποιεί την ισχυρή αρχιτεκτονική για να καθορίσει τον έλεγχο του προγράμματος μας. Γενικά χαρακτηριστικά που έχει το προγραμματιστικό αυτό περιβάλλον είναι ότι τα δεδομένα χρονισμού καταγράφονται για όλους τους κόμβους του έργου. Δίνετε ακριβής εκτίμηση πόρων κατά την εργασία, υπάρχει ακόμα μια αυξανόμενη βιβλιοθήκη προκατασκευασμένων στοιχείων που διευκολύνουν την αλληλεπίδραση με οποιοδήποτε υλικό και εξοικονομούν χρόνο δημιουργώντας έτσι κοινές δομές κόμβων. Αυτός επίσης είναι ένας νέος τρόπος να σχεδιαστεί το ενσωματωμένο λογισμικό σε πραγματικό χρόνο. Αξίζει να αναφέρουμε ότι με την πλήρη έκδοση μπορείτε να ανεβάσετε το τελικό σας πρόγραμμα στο Arduino ώστε να μπορεί να τρέξει χωρίς σύνδεση στον υπολογιστή σας. Για περισσότερες πληροφορίες μπορείτε να απευθυνθείτε στην αντίστοιχη ιστοσελίδα αλλά πιο γενικά στην ιστοσελίδα που έχω βρει την πηγή αυτή, θα την βρείτε στο τέλος του κεφαλαίου αυτού.[21.4]

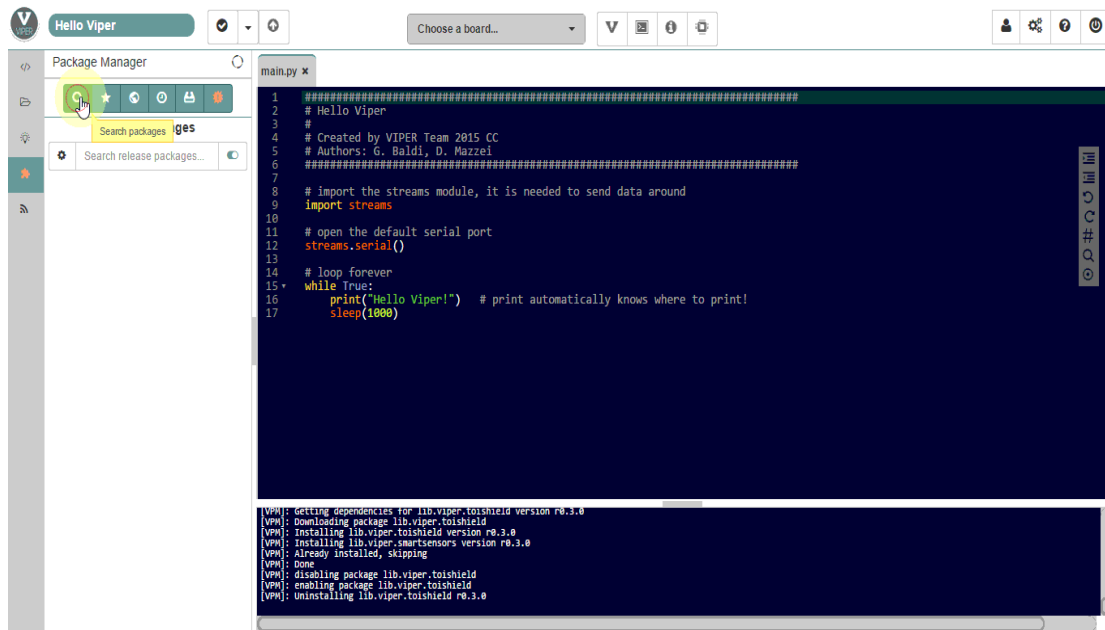


Εικόνα 3.5: Προγραμματιστικό περιβάλλον Embrio

4.2.5 ZERYNTH

Το Zerynth παλαιότερα γνωστό και ως Viper είναι μια εύκολη εφαρμογή για Arduino και άλλες πλακέτες μικροελεγκτή ARM 32. Το Zerynth έχει την δυνατότητα να αναπτυχθεί σε Python για Arduino Due και παρόμοιους πίνακες χρησιμοποιώντας παραδείγματα του υπολογιστή και του κινητού προγραμματισμού όπως θα δούμε και παρακάτω. Το Zerynth είναι συμβατό με όλους τους αισθητήρες και τα kit. Επιπροσθέτως, αξίζει να αναφερθεί ότι το Zerynth είναι ένα εργαλείο ανοικτού κώδικα εύκολο και ευχάριστο στους χρήστες.

Λίγα λόγια για το Zerynth studio τα αρχεία του Zerynth μπορούν να μετατραπούν για Arduino μέσα από το Zerynth studio. Είναι ένα αναπτυξιακό περιβάλλον βασισμένο σε προγράμματα περιήγησης και συμβατό με όλα τα λειτουργικά συστήματα.[21.5]



Εικόνα 3.6: Προγραμματιστικό περιβάλλον Zerynth

Ακόμα, όπως αναφέραμε και παραπάνω το Zerynth περιλαμβάνει μια έτοιμη χρήση εφαρμογής για τα κινητά τηλέφωνα το οποίο μπορούμε να το βρούμε στο store με το όνομα Zerynth App. Το Zerynth App σε πρότυπα HTML5 τα οποία μπορούν να επεξεργαστούν μέσω του Zerynth studio. [21.6]



Εικόνα 3.7: Προγραμματιστικό περιβάλλον Zerynth για Smartphones

4.2.6 PlatformIO IDE

Το PlatformIO IDE είναι ένα ενσωματωμένο περιβάλλον ανάπτυξης νέας γενιάς για το διαδίκτυο. Κάποιες δυνατότητες που έχει είναι το σύστημα μεταξύ πλατφόρμας, χωρίς εξωτερικές εξάρτησης στο λογισμικό έχοντας την δυνατότητα σύνδεσης με πάνω από 200 ενσωματωμένες κάρτες και 15 πλατφόρμες ανάπτυξης. Ένα ευχάριστο

προγραμματιστικό περιβάλλον με διαφορετικά χρώματα και κατανοητή στοίχιση. Κάτι ακόμα που αξίζει να αναφερθεί είναι το ενσωματωμένο τερματικό με το εργαλείο PlatformIO CLI και ισχυρό σύστημα παρακολούθησης σειριακών θυρών. [21.7]

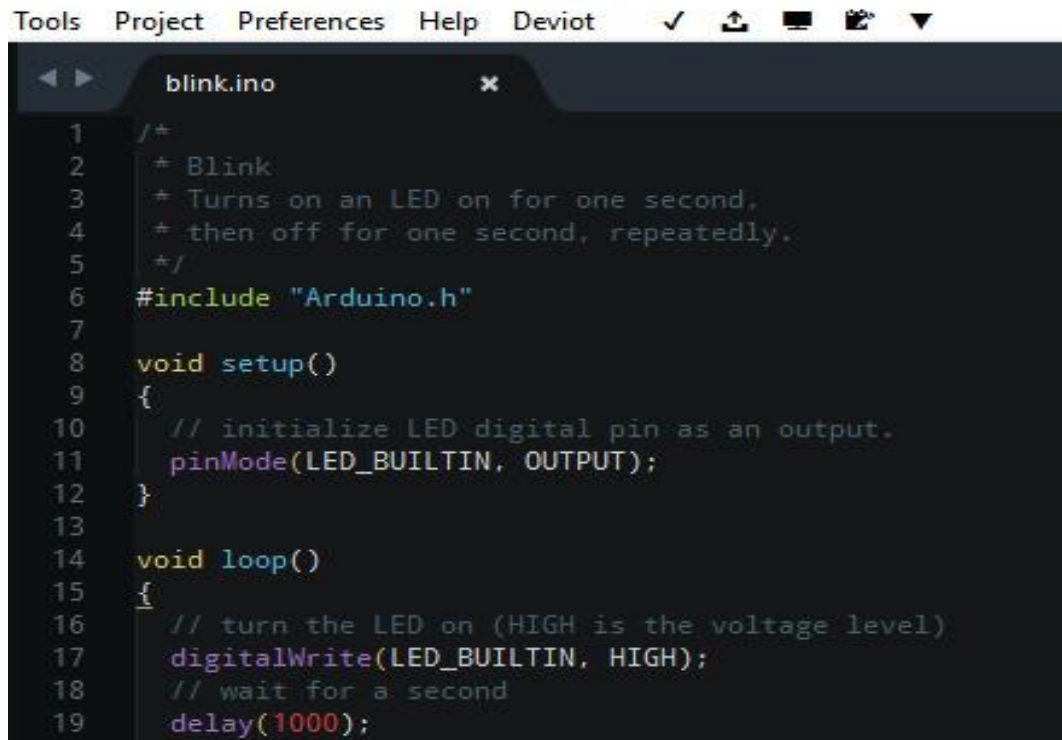


```
1  /**
2   * Test led: Arduino uno
3   */
4
5  #include "Arduino.h"
6
7  int redPin = 9;
8  int greenPin = 10;
9  int bluePin = 11;
10
11 void setColor(int red, int green, int blue)
12 {
13     red = 255 - red;
14     green = 255 - green;
15     blue = 255 - blue;
16 }
17
18 void setup() {
19     Serial.begin(9600);
20     Serial.println("Setup");
21     pinMode(redPin, OUTPUT);
22     pinMode(greenPin, OUTPUT);
23     pinMode(bluePin, OUTPUT);
24 }
```

Εικόνα 3.8: Προγραμματιστικό περιβάλλον PlatformIO IDE

4.2.7 Deviot

Είναι μια έγχρωμη κονσόλα εξόδου. Είναι πολύ εύκολος διαχειριστής βιβλιοθήκης για να αναζητήσετε και να εγκαταστήσετε ή να αφαιρέσετε τις βιβλιοθήκες σας χωρίς καμία προσπάθεια. Δουλεύει σε Arduino projects ή σε PlatformIO.[21.8]



```
Tools Project Preferences Help Deviot ✓ ⬆ ⬇ ⬇ ⬇
blink.ino x
1  /*
2   * Blink
3   * Turns on an LED on for one second,
4   * then off for one second, repeatedly.
5   */
6  #include "Arduino.h"
7
8  void setup()
9  {
10   // initialize LED digital pin as an output.
11   pinMode(LED_BUILTIN, OUTPUT);
12 }
13
14 void loop()
15 {
16   // turn the LED on (HIGH is the voltage level)
17   digitalWrite(LED_BUILTIN, HIGH);
18   // wait for a second
19   delay(1000);
```

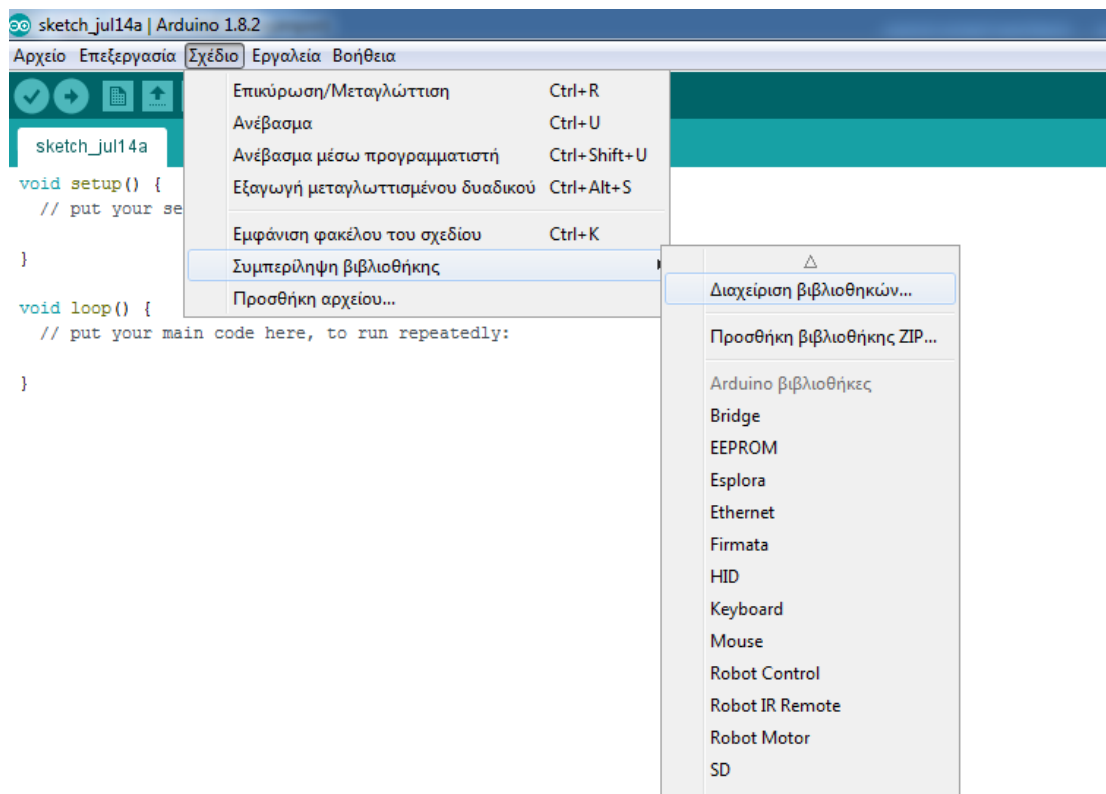
Εικόνα 3.9: Προγραμματιστικό περιβάλλον Deviot

Για ακόμα περισσότερες πληροφορίες μπορείτε να μπείτε στην ιστοσελίδα του Arduino.[21]

4.3 ΒΙΒΛΙΟΘΗΚΕΣ ΓΙΑ ΤΟ ARDUINO IDE ΚΑΙ ΠΩΣ ΓΙΝΕΤΑΙ Η ΕΓΚΑΤΑΣΤΑΣΗ ΑΥΤΩΝ

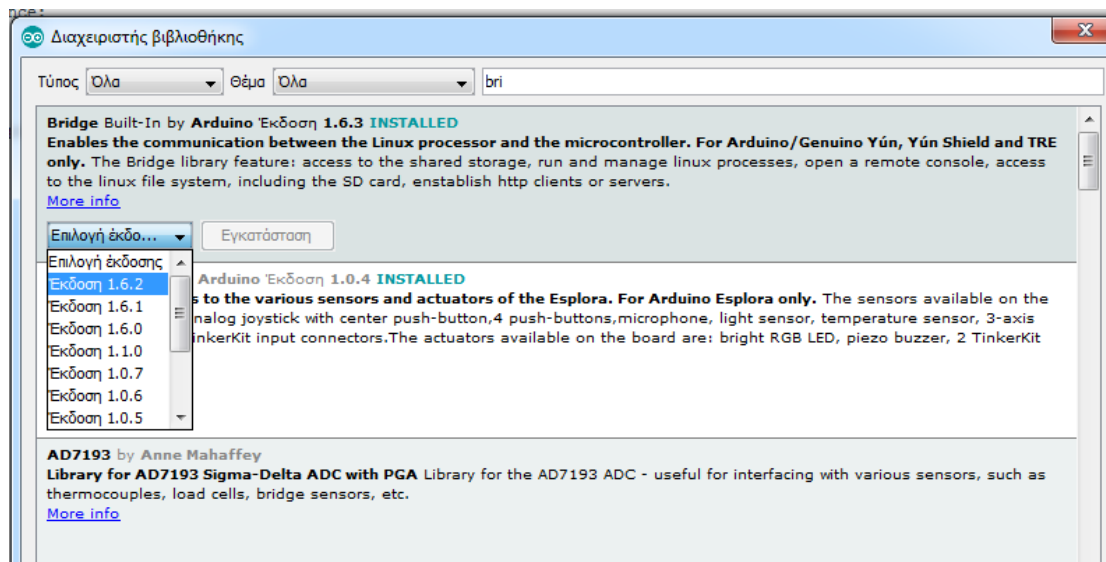
Βιβλιοθήκες είναι η συλλογή κώδικα και συναρτήσεις (functions) που σας διευκολύνει να συνδεθείτε με αισθητήρα, οθόνη και για πολλά αλλά εργαλεία που έρχονται σε επαφή με το Arduino. Για παράδειγμα η ενσωματωμένη βιβλιοθήκη Liquidcrystal διευκολύνει την σύνδεση με τις οθόνες των LCD χαρακτηριστικών. Στο διαδίκτυο θα μπορέσουμε να βρούμε πολλών ειδών βιβλιοθήκες απλά όμως πρέπει να τις εγκαταστήσουμε πρώτα, παρακάτω θα δείτε πως μπορεί να γίνει αυτό.

Για να γίνει μια εγκατάσταση στο Arduino IDE μια βιβλιοθήκη γίνεται με τον παρακάτω τρόπο που θα δείτε. Θα κάνουμε κλικ στο Σχέδιο και στην συνέχεια επιλέγουμε Συμπερίληψη βιβλιοθήκης και επιλέγουμε την Διαχείριση βιβλιοθηκών.



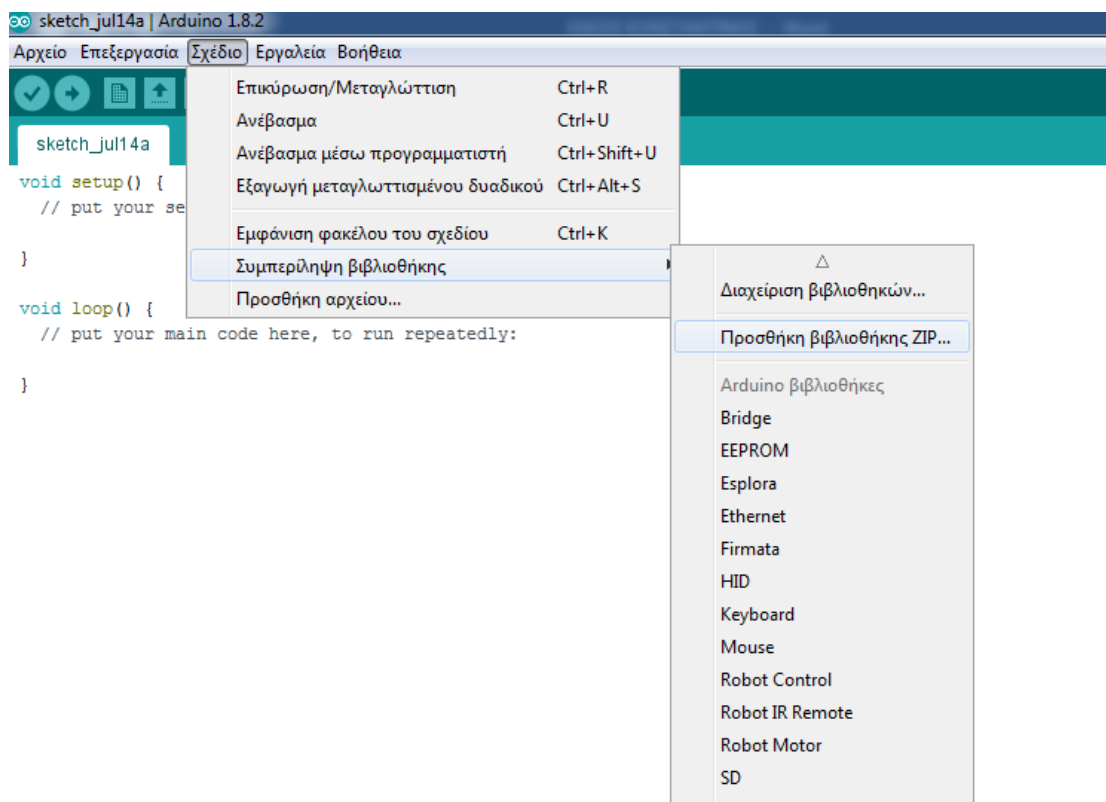
Εικόνα 3.10: Εισαγωγή Βιβλιοθήκης από υπάρχον βιβλιοθήκη(θήμα 1ο)

Αμέσως μετά θα ανοίξει ο διαχειριστής βιβλιοθήκης και θα βρείτε μια βιβλιοθήκη ήδη εγκατεστημένη ή που είναι έτοιμη για εγκατάσταση. Θα πάρουμε για παράδειγμα την βιβλιοθήκη bridge για να κάνουμε τις ενέργειες αυτές. Όταν μεταφερθούμε στη διαχείριση βιβλιοθήκης θα βρούμε την βιβλιοθήκη που θέλουμε και συγκεκριμένα την βιβλιοθήκη bridge και όταν το βρούμε κάτω αριστερά στη κάθε βιβλιοθήκη υπάρχει η έκδοση που θέλουμε να επιλέξουμε και δίπλα ακριβώς η εγκατάσταση αυτού. Κάνουμε την λήψη που μπορεί να διαρκέσει μερικά λεπτά και στην συνέχεια μπορούμε να κλείσουμε το παράθυρο με τις βιβλιοθήκες. Μπορούμε να το δούμε και παρακάτω στην εικόνα.



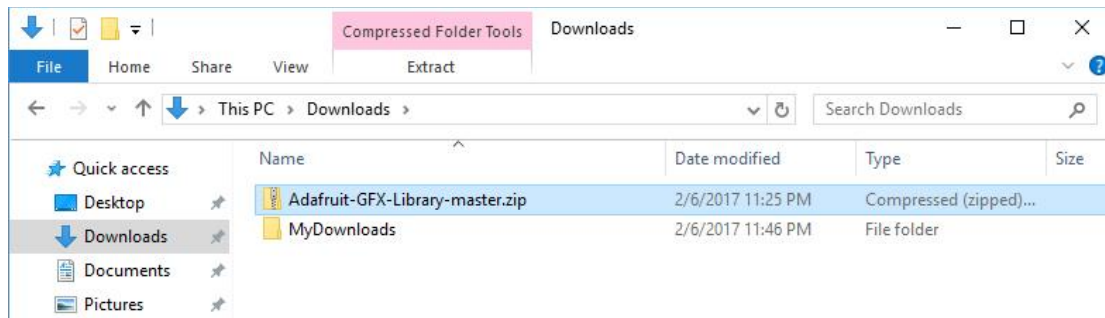
Εικόνα 3.11: Εισαγωγή Βιβλιοθήκης από υπάρχον βιβλιοθήκη(βήμα 2ο)

Για να κάνουμε εγκατάσταση βιβλιοθήκης ZIP γίνεται και αυτό με αρκετά απλό τρόπο πατώντας κλικ στο σκίτσο στην συνέχεια πατάμε στην συμπερίληψη βιβλιοθήκης και επιλέγουμε την επιλογή Προσθήκη βιβλιοθήκης zip αρχείου. Όπως βλέπουμε και στην εικόνα παρακάτω.



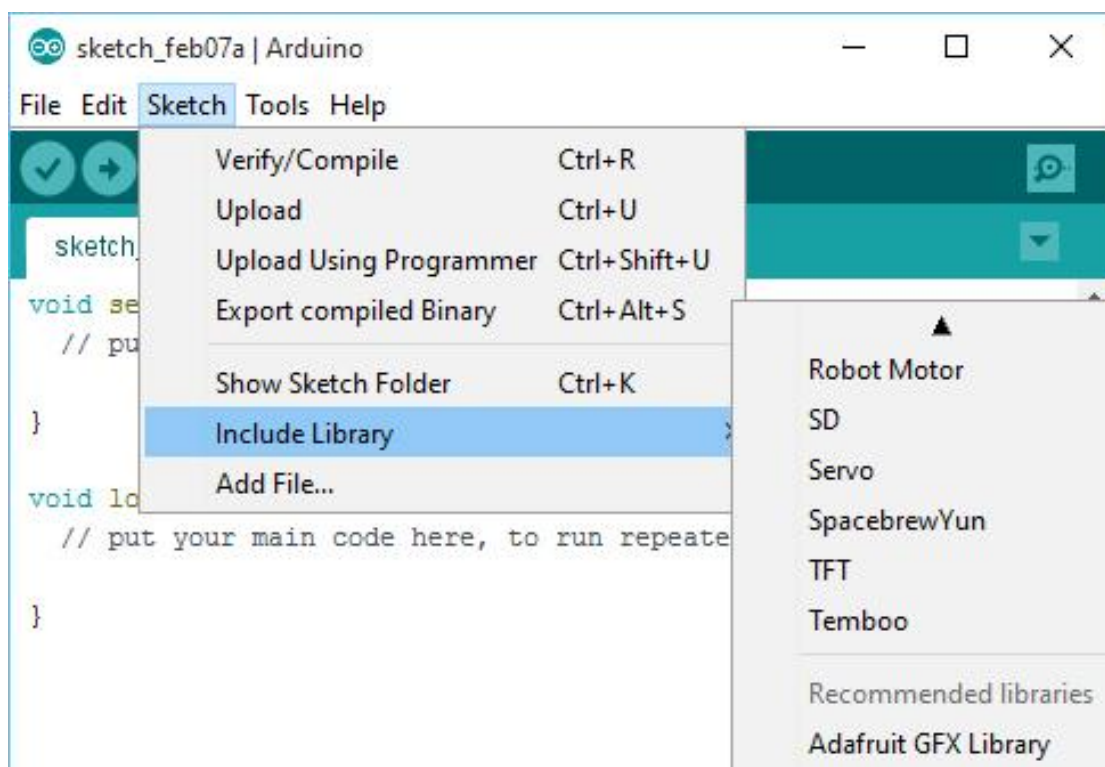
Εικόνα 3.12: Εισαγωγή Βιβλιοθήκης από ZIP(βήμα 1ο)

Στην συνέχεια θα ζητηθεί να βρούμε το αρχείο ZIP και να το ανοίξουμε. Όπως παρακάτω.



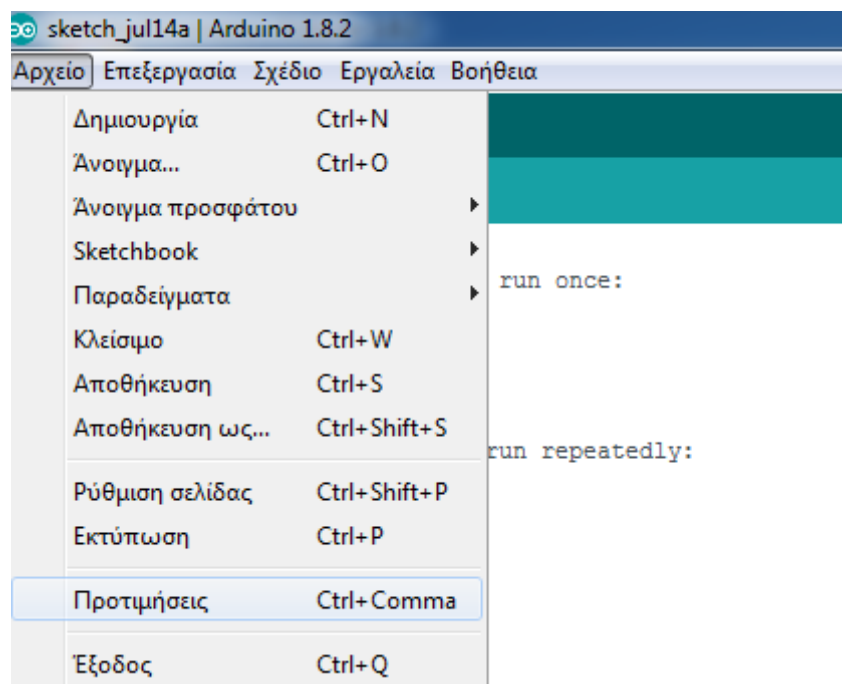
Εικόνα 3.13: Εισαγωγή Βιβλιοθήκης από ZIP(βήμα 2ο)

Στην συνέχεια μπορούμε να πατήσουμε πάλι στο σκίτσο και να επιλέξουμε την εισαγωγή βιβλιοθήκης και εκεί θα δούμε την βιβλιοθήκη που προσθέσαμε.

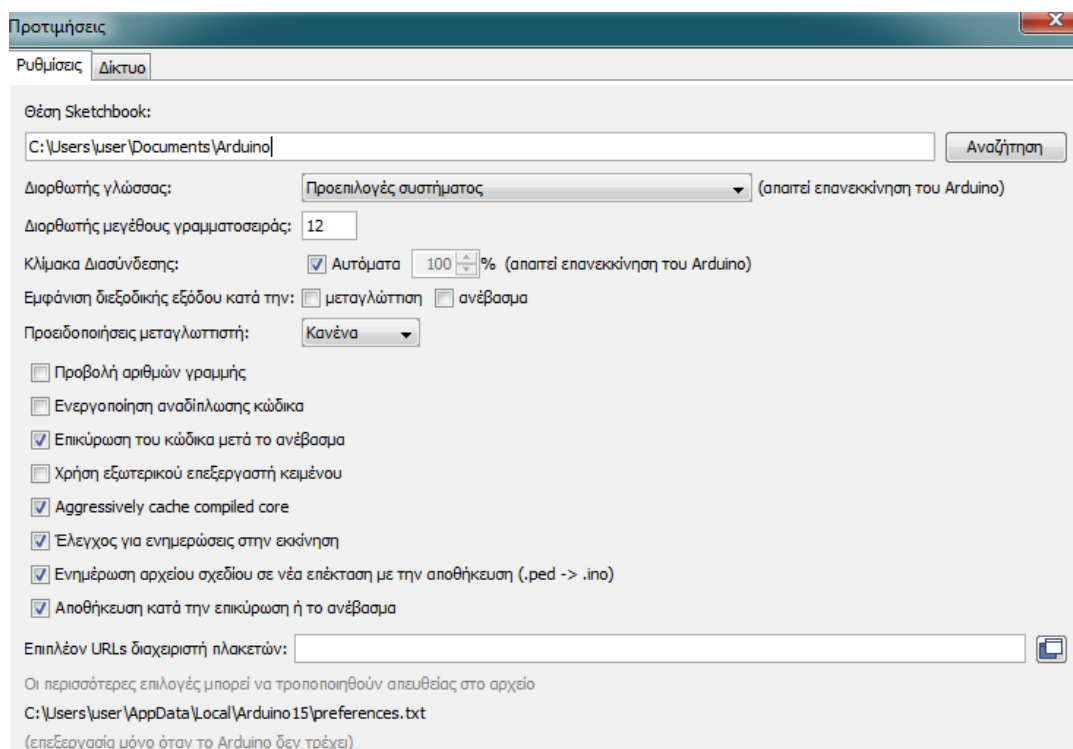


Εικόνα 3.14: Εισαγωγή Βιβλιοθήκης από ZIP(βήμα 3ο)

Υπάρχει ακόμα και ο τρόπος να κάνεις την εγκατάσταση μιας βιβλιοθήκης χειροκίνητα και αυτό μπορούμε να το επιτύχουμε με τον τρόπο που θα δούμε παρακάτω. Θα πατήσουμε κλικ στο Αρχείο και μετά να επιλέξουμε το sketchbook η διαδρομή εμφανίζεται όταν πατάμε κλικ στο αρχείο και μετά προτίμησης

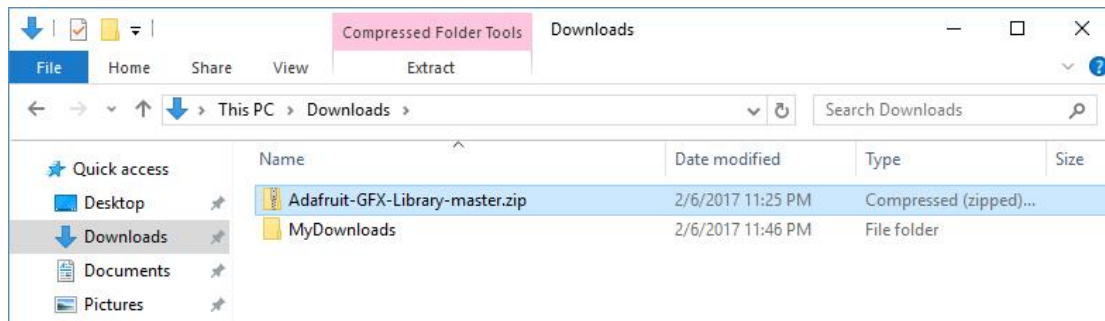


Εικόνα 3.15: Εισαγωγή Βιβλιοθήκης από ZIP χειροκίνητα (βήμα 1ο)



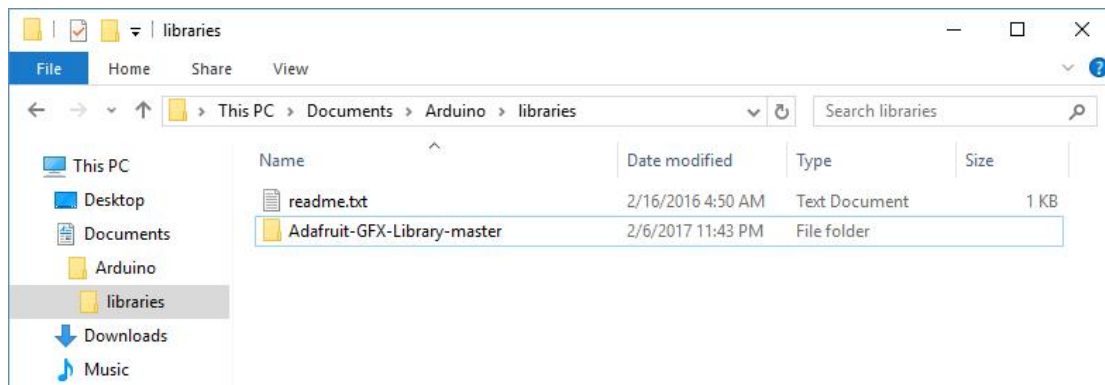
Εικόνα 3.16: Εισαγωγή Βιβλιοθήκης από ZIP χειροκίνητα (βήμα 2ο)

Στην συνέχεια επιλέγουμε τα αρχεία τα οποία έχουμε κατεβάσει και στην συνέχεια βρίσκουμε το αρχείο ZIP.



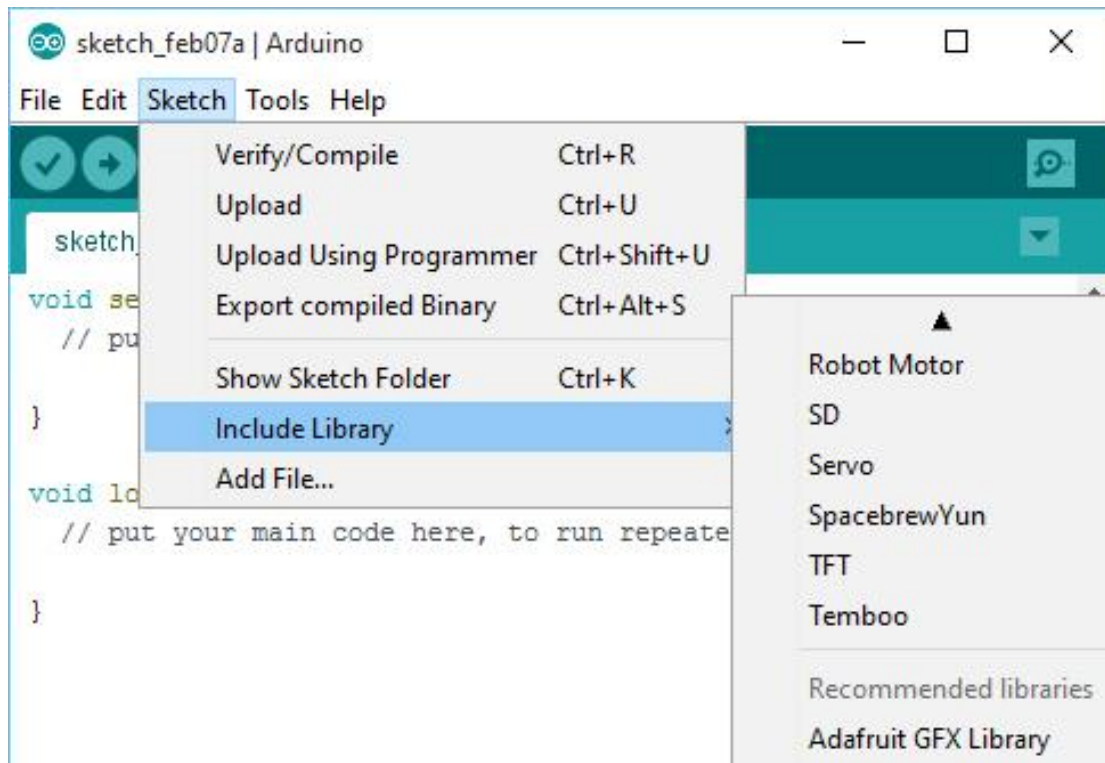
Εικόνα 3.17: Εισαγωγή Βιβλιοθήκης από ZIP χειροκίνητα (βήμα 3ο)

Το εξάγουμε και το μεταφέρουμε στο φάκελο που είναι οι βιβλιοθήκες του Arduino



Εικόνα 3.18: Εισαγωγή Βιβλιοθήκης από ZIP χειροκίνητα (βήμα 4ο)

Στην συνέχεια κάνουμε τον έλεγχο κάνοντας κλικ στο σχέδιο στην συνέχεια Συμπερίληψη βιβλιοθήκης και θα διαπιστώσουμε ότι βρίσκετε μέσα η βιβλιοθήκη που εμείς τοποθετήσαμε.



Εικόνα 3.19: Εισαγωγή Βιβλιοθήκης από ZIP χειροκίνητα (βήμα 5ο)

Όπως βλέπουμε και στις εικόνες η διαδικασία είναι αρκετά απλή και εύκολη. Σε αυτήν την σελίδα μπορούμε να δούμε την εγκατάσταση βιβλιοθηκών.[22]

5. ΣΧΕΔΙΑΣΜΟΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

5.1 ΤΙ ΘΕΛΟΥΜΕ ΝΑ ΦΤΙΑΞΟΥΜΕ

Στο κεφάλαιο αυτό θα δούμε πως με την πλακέτα του μικροελεγκτή Arduino, μπορούμε να φτιάξουμε ένα τηλεκατευθυνόμενο όχημα. Από την κατασκευή του μηχανικά (την συναρμολόγηση που θα γίνει στα κομμάτια που χρειαζόμαστε και έχουμε για να δημιουργήσουμε το αυτοκίνητο), την συναρμολόγηση των πλακετών με το όχημα μας μέχρι και τον προγραμματισμό του οχήματος μας. Το όχημα θα το προγραμματίσουμε με την γλώσσα Wiring η οποία είναι βασισμένη στην γλώσσα προγραμματισμού C και C++. Οι εντολές που θα εκτελεί το όχημα μας θα περαστούν μέσω του υπολογιστή μας στη πλακέτα του Arduino. Το όχημα μας θα κάνει διαφορές εντολές όπως να κάνει κίνηση εμπρός πίσω, θα μπορεί να κάνει και συγκεκριμένες διαδρομές που εμείς θα του δώσουμε εντολή (για παράδειγμα να κάνει ένα γύρο το τετράγωνο ή να κάνει ένα τρίγωνο) και γενικά ό,τι εντολή θέλουμε εμείς να πραγματοποιήσει με το πάτημα ενός κουμπιού. Αυτή η εφαρμογή που κατασκευάστηκε παρακάτω θα μπορούσε στο μέλλον να είναι ένας τρόπος μεταφοράς χωρίς να χρειάζεται κάποιος να κινεί το όχημα απλά και μόνο με το να έχουμε προγραμματίσει το όχημα μας από πριν. Όλα τα υλικά που θα χρειαστούμε θα τα δούμε στην συνέχεια αναλυτικά, όπως και που μπορούμε να βρούμε το καθένα από αυτά. Στο επόμενο κεφάλαιο θα δούμε και την κατασκευή του οχήματος σας με πολύ απλά βήματα . Με τον τρόπο αυτόν θα καταφέρετε να μάθετε να προγραμματίζετε μια πλακέτα Arduino αλλά και τις συνδεσμολογίες καλωδίων.

5.2 ΤΙ ΥΛΙΚΟ ΘΑ ΧΡΕΙΑΣΤΟΥΜΕ

Για την κατασκευή του τηλεκατευθυνόμενου οχήματος χρειάζονται κάποια υλικά . Μερικά από αυτά τα υλικά είναι μηχανικά και κάποια ηλεκτρονικά στη συνέχεια θα τα δούμε αναλυτικά.

Για την κατασκευή του ρομποτικού οχήματος χρειάζονται τα παρακάτω υλικά:

Σασί από plexiglass κομμένο με laser

DC κινητήρες συνεχούς ρεύματος (GEAR MOTORS 1:48 3-6V)

Τροχούς με λάστιχο

1 τροχός που κινείτε προς όλες τις κατευθύνσεις (Universal Wheel)

1 θήκη για 4 μπαταρίες AA

2 ροδέλες για την καταγραφή ταχύτητας



Εικόνα 4.1: Υλικά για την κατασκευή του σκελετού του οχήματος [23]

Βίδες, παξιμάδια

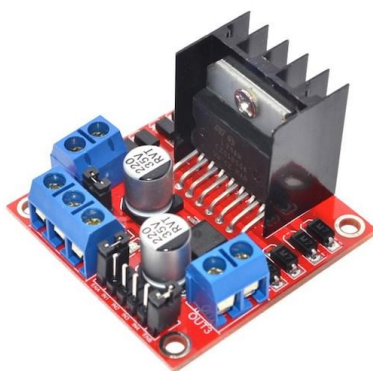
Τα ηλεκτρονικά πράγματα που χρειαζόμαστε ή αλλιώς οι πλακέτες που θα χρειαστούμε είναι:

Μια πλακέτα Arduino Uno



Εικόνα 4.2: Arduino Uno[24]

Το Arduino UNO είναι μια πλακέτα μικροελεγκτή η οποία είναι βασισμένη στον επεξεργαστή της Atmel ATmega 328P έχει 14 ακροδέκτες εισόδου / εξόδου οι 6 από αυτές μπορούν να χρησιμοποιηθούν ως εξόδοι PWM, 6 αναλογικές εισόδους, έναν 16MHz ταλαντωτή κρυστάλλου, μια σύνδεση USB, μια αποδοχή ρεύματος μια κεφαλίδα ICSP και ένα κουμπί επαναφοράς. Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί μέσω προγραμμάτων με Processing, Max/MSP, Pure Data, SuperCollider. Το διάγραμμα και οι πληροφορίες, για το υλικό είναι ελεύθερα για αυτούς που θέλουν να ασχοληθούν να συναρμολογήσουν και να προγραμματίσουν το Arduino μόνοι τους. Καλό θα ήταν να αναφέρω ότι προγραμματίζετε με γλώσσα wiring (είναι μια γλώσσα βασισμένη στη γλώσσα C και C++) όπως αναφέρω και αναλυτικά στα παραπάνω κεφάλαια.



**Εικόνα 4.3: Πλακέτα
L298N [25]**

Πλακέτα L298N Dual H-Bridge Motor Controller

Η πλακέτα L298N είναι ένα κύριο τσιπ της ST εταιρίας. Είναι χαμηλής θερμότητας και εξαιρετικής απόδοσης κατά των παρεμβολών. Επίσης διαθέτει υψηλής ισχύς εργασίας 46V επίσης μεγάλο ρεύμα μπορεί να φτάσει τα 3A MAX και να συνεχίσει με ρεύμα 2A, δύναμης 25W. Μπορεί να οδηγήσει έναν κινητήρα βιωματικής φάσης 2 φάσεων, έναν κινητήρα στάσης 4 φάσεων ή δυο κινητήρες συνεχούς ρεύματος. Ενσωματωμένο 78M05 που παρέχει ισχύ από την ισχύ του μηχανισμού κίνησης. Επίσης όταν τροφοδοτείτε ισχύ άνω των 12V πρέπει να χρησιμοποιείτε η εξωτερική ισχύς 5V. Διαθέτει μεγάλη χωρητικότητα φίλτρου και δίοδο προστασίας μετά από ροή για να είναι πιο σταθερή και αξιόπιστη.

Χαρακτηριστικά - Τεχνικές Λεπτομέρειες

Chip L298N (ST NEW)

Λογική τάση: 5V

Τάση μετάδοσης: 5V-35V

Λογικό ρεύμα: 0mA-36mA

Ρεύμα κίνησης: 2A

Μέγιστη ισχύς: 25W

IR Wireless Remote Control



Εικόνα 4.4: Πλακέτα wireless και τηλεχειριστήριο [26]

Ασύρματο τηλεχειριστήριο για το Arduino με μονάδα λείψεις με υπέρυθρες. Είναι συνεργάσιμο με την πλακέτα του Arduino και διαθέτει 20 πλήκτρα λειτουργιών. Είναι μικρό και ευχάριστο, λαμβάνει σήμα 38KHz η απόσταση μετάδοσης φτάνει τα 8 μετρά ανάλογα τα φυσικά εμπόδια

Μπαταριά CR2025

Χωρητικότητα μπαταρίας: 160mAh

Ρεύμα στατικό: 3uA-5uA

Ρεύμα δυναμικό: 3mA- 5mA

9V Battery Snap Connector to DC Male Power Adapter Cables for Arduino



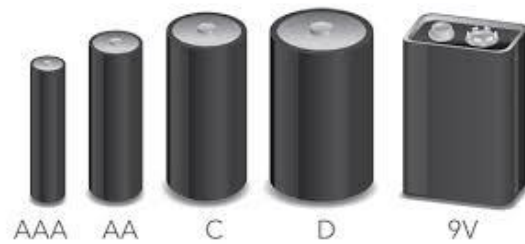
Εικόνα 4.5: Θήκη μπαταρίας κλιπ για 9V μπαταρία[27]

Είναι ιδικό κλιπ μπαταρίας με ένα καλώδιο 150mm

Διαθέτει δύο καλώδια κόκκινο και μαύρο

Ακόμα μια κεφαλή που εφαρμόζει άψογα στην πλακέτα Arduino Uno

Μπαταρίες



Εικόνα 4.6: Μπαταρίες ονομαστικά [28]

Στην παράγραφο αυτήν θα δούμε τι μπαταρίες θα χρειαστούμε για το μηχανοκίνητο όχημα μας. Και αναλυτικά για την πλακέτα Arduino αν δεν χρησιμοποιήσουμε το power bank όπως για το τηλεκοντρόλ και τις μπαταρίες του οχήματος μας.

2 A Μπαταρίες



Εικόνα 4.7:
Μπαταρίες 2A [29]

Για την τροφοδοσία του τηλεκατευθυνόμενου οχήματος σας θα χρειαστείτε επαναφορτιζόμενες μπαταρίες οι οποίες κυκλοφορούν στο εμπόριο. Για την εφαρμογή σας θα χρειαστείτε 4 μπαταρίες 2 A για τους δυο κινητήρες. Οι μπαταρίες αυτές παρέχουν 1,2V με τεχνολογία Νικελίου-Μετάλλου (Ni-Mh) ή Ιόντων Λιθίου

(Li-ion) και έχουν ικανοποιητική διάρκεια ζωής. Πρέπει να αποδίδουν τουλάχιστον 2500mAh και είναι κατά προτίμηση να είναι ιόντων λιθίου.

Μπαταρία CR2025



Εικόνα 4.8:
Μπαταρία CR2025
[30]

Μπαταρία CR2025 είναι μια μικρή μπαταρία λιθίου 3V η οποία χρησιμοποιείτε στο ασύρματο τηλεκοντρόλ.



Εικόνα 4.9: Μπαταρία 9V

Μπαταρία 9V

Μπαταρία ψευδαργύρου - άνθρακα 9V. Ιδανική για χρήση σε συσκευές με μικρή κατανάλωση ενέργειας διαθέτει μια τάση 9V και ο τύπος μπαταρίας είναι 6F22.



Εικόνα 4.10: Καλώδια σύνδεσης [31]

Καλώδια σύνδεσης

Θα χρειαστείτε καλώδια για να συνδέσετε το Arduino με την Πλακέτα L298N και την Πλακέτα L298N με τα μοτέρ και την πλακέτα του ασύρματου τηλεκοντρόλ. Το kit του Arduino περιέχει αρκετά καλώδια (Electrical Cables) και διαφόρων ειδών έτσι ώστε θα σας δώσουν την δυνατότητα να συνδέσετε όλα σχεδόν τα μέρη. Αλλά καλό θα ήταν να αγοράσετε και μερικά διότι δεν είναι ακριβά και σίγουρα κάπου θα χρειαστούν για χρήση.

Power bank



Εικόνα 4.11: Power Bank[32]

Θα χρειαστείτε επίσης ένα καλό power bank για να τροφοδοτεί το Arduino για να δίνει ρεύμα έτσι ώστε να εκτελεί τις ενέργειες που του έχουμε προγραμματίσει να κάνει εάν βέβαια δεν εισάγετε μπαταριά στην είσοδο του Arduino. Καλό θα ήταν ανάλογα με το πόση ώρα θα θελήσετε να κινείτε το όχημα να πάρετε και τον αντίστοιχο μεγέθους power bank, για παράδειγμα στο όχημα σας καλό θα ήταν να είναι τουλάχιστον 2000mAh και να έχει τουλάχιστον μια θήρα USB ακόμα η ένταση του ρεύματος να είναι 2A.

Διάφορα εργαλεία

Για να κατασκευάσουμε το τηλεκατευθυνόμενο όχημα θα χρειαστείτε ένα κατσαβίδι σταυρό και ένα ίσιο , έναν μικρό κοφτάκι, ένα κοπίδι και ένα κολλητήρι αν τυχόν θέλουμε να τα κολλήσουμε τα καλώδια μόνιμα στο όχημα ή να μικρύνουμε τα καλώδια για να είναι πιο ομοιόμορφα και συμμετρικά. Ακόμα και ένα πολύμετρο για να εμφανίζει πόση ενέργεια έχουν οι μπαταρίες.

5.3 ΤΙ ΕΝΕΡΓΙΕΣ ΠΡΕΠΕΙ ΝΑ ΓΙΝΟΥΝ ΓΙΑ ΒΡΕΘΟΥΝ ΤΑ ΥΛΙΚΑ ΑΥΤΑ

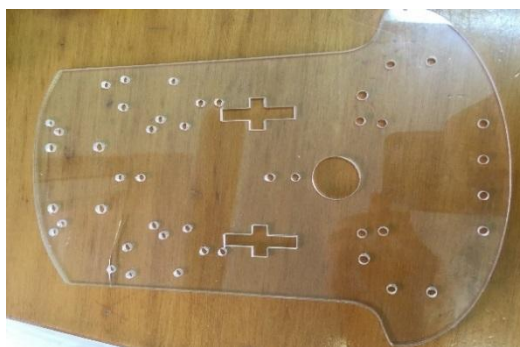
Το κομμάτι της εργασίας που έχει λιγότερες απαιτήσεις είναι τα υλικά, μπορείτε να τα βρείτε παντού πολύ εύκολα και αρκετά οικονομικά. Ανάλογα βέβαια με το ποσά χρήματα και χρόνο έχετε στην διάθεση σας. Αν τα υλικά μας τα αγοράσετε από ελληνικά καταστήματα θα σας κοστίσουν γύρω στα 50 ευρώ και μέσα σε 3 με 4 εργάσιμες μέρες θα τα έχετε στα χέρια σας ενώ αν τα πάρετε από εξωτερικό θα σας κοστίσουν γύρω στα 25 ευρώ αλλά θα αργήσουν περίπου έναν μήνα και θα μπορέσετε να πάρετε και άλλα περιφερειακά κομμάτια ώστε να αναπτύξετε ακόμα το όχημα σας.

6 ΥΛΟΠΟΙΗΣΗ

Στο κεφάλαιο αυτό θα δείτε αναλυτικά τι πρέπει να κάνετε για να κατασκευάσετε το όχημα σας από τις βίδες μέχρι και τα καλώδια αναλυτικά. Πως πρέπει να τοποθετήσετε τα αντικείμενα στην πλαστική βάση του οχήματος. Επίσης, θα αναλυθούν οι συνδεσμολογίες που πρέπει να γίνουν μεταξύ του οχήματος και των πλακετών που έχετε στην διάθεση σας. Το πιο σημαντικό κομμάτι από όλα είναι ο προγραμματισμός που θα κάνετε στο όχημα σας και οι εντολές που θα του δώσετε. Με ή χωρίς την χρήση του ασύρματου τηλεχειριστήριου. Με την βοήθεια που δίνετε παρακάτω και με τον έτοιμο κώδικα μπορείτε εύκολα να προγραμματίσετε εσείς το όχημα που θα έχετε είδη κατασκευάσει.

6.1 ΣΥΝΑΡΜΟΛΟΓΙΣΗ ΤΟΥ ΟΧΗΜΑΤΟΣ

Το πρώτο πράγμα που θα κάνετε για να κατασκευάσουμε το έργο είναι η κατασκευή του οχήματος μας για να το κάνουμε αυτό θα χρειαστούμε ένα κατσαβίδι (σταυρό) και μια μικρή πένσα για να σφίξουμε τις βίδες.

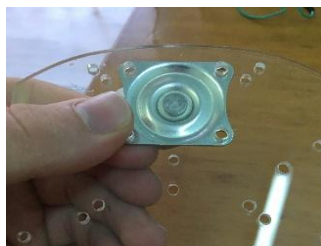


Εικόνα 5.1: Βάση οχήματος

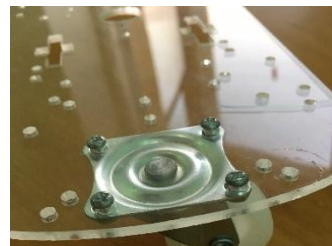
Αρχικά θα πάρουμε την βάση του οχήματος μας και θα τοποθετήσουμε τον τροχό ο οποίος κινείται προς όλες τις κατευθύνσεις. Θα τα ενώσουμε μεταξύ τους με τέσσερις μικρές βίδες όπως μπορούμε να δούμε πάρα κάτω.



Εικόνα 5.2:
Κατευθυντήριο
τροχός



Εικόνα 5.3:
Σύνδεση
κατευθυντήριου
τροχού και βάση
οχήματος (A)



Εικόνα 5.4:
Σύνδεση
κατευθυντήριου
τροχού και βάση
οχήματος (B)

Στην συνέχεια θα συναρμολογήσουμε τους κινητήρες και τα ροδάκια κίνησης. Στο σακουλάκι με τις βίδες θα βρούμε ένα μικρό ορθογώνιο σιδεράκι που μπορούμε να το



Εικόνα 5.5: Σύνδεση
μοτέρ με ρόδες (Α)



Εικόνα 5.6:
Σύνδεση μοτέρ
με ρόδες (Β)



Εικόνα 5.7:
Σύνδεση μοτέρ με
ρόδες (Γ)

τοποθετήσουμε πάνω στο μοτέρ. Θα περάσουμε αναμεσά τους τις 2 μεγαλύτερες βίδες τις οποίες θα τις βρούμε μέσα στο σακουλάκι που παραλάβαμε το όχημα σας.

Στην συνέχεια θα σφίξουμε τις βίδες με το κατσαβίδι και την πένσα έτσι ώστε να μην τρεμοπαίζει. Θα τοποθετήσουμε το ροδάκι από την εξωτερική πλευρά δηλαδή από την πλευρά που μπαίνουν τα καλώδια δηλαδή από την αντίθετη πλευρά που έχουμε τοποθέτηση την ορθογώνια βάση. Στην συνέχεια θα τοποθετήσουμε τις δυο ροδέλες για την καταγραφή της ταχύτητας από την αντίθετη πλευρά της ρόδας όπως ακριβώς βλέπουμε και παρακάτω στις εικόνες.

Θα κάνουμε ακριβώς το ίδιο και στον άλλον μηχανισμό και στην συνέχεια θα τα



τοπο

Εικόνα 5.8:
Σύνδεση μοτέρ
με ρόδες (Δ)



βά

Εικόνα 5.9:
Σύνδεση μοτέρ
με ρόδες (Ε)



γα

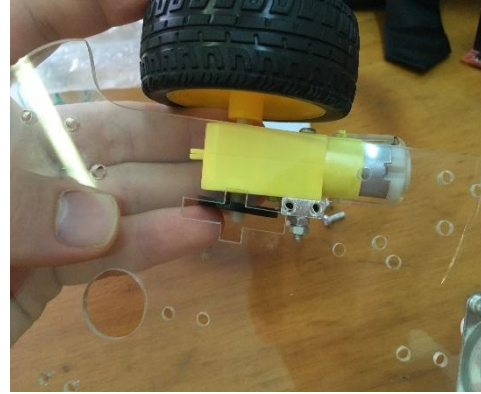
Εικόνα 5.10:
Σύνδεση μοτέρ
με ρόδες (Ζ)

που

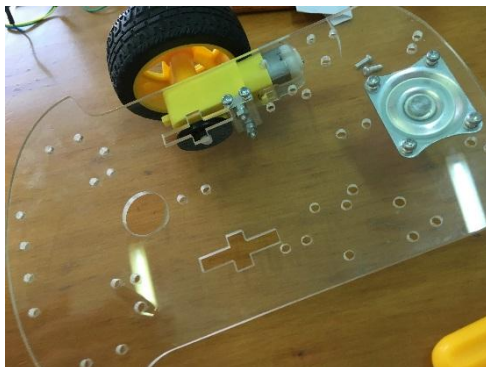
κινείτε προς όλες τις κατευθύνσεις. Υπάρχει συγκεκριμένη θέση στην οποία μπορούμε να βάλουμε τον μηχανισμό.



Εικόνα 5.11: Σύνδεση
μοτέρ με ρόδες στην
βάση (Α)



Εικόνα 5.12: Σύνδεση
μοτέρ με ρόδες στην
βάση (Β)



Εικόνα 5.13: Σύνδεση
μοτέρ με ρόδες στην
βάση (Γ)



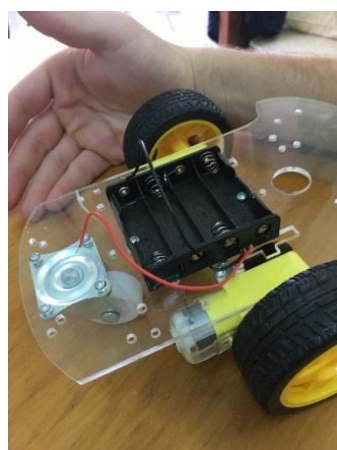
Εικόνα 5.14: Σύνδεση
μοτέρ με ρόδες στην
βάση (Δ)

Στην συνέχεια θα σφίξουμε με δυο μικρές βίδες την βάση της ρόδας πάνω στην πλαστική επιφάνεια. Το ίδιο ακριβώς θα κάνουμε και από την άλλη πλευρά. Έτσι πλέον το όχημα μας μπορεί να στέκετε μόνο του. Όπως θα δούμε και παρακάτω στις εικόνες.

Στην συνέχεια θα τοποθετήσουμε την βάση των μπαταριών και θα στερεωθεί με δυο βίδες όπως βλέπουμε στην εικόνα από κάτω (εικόνες 5.15, 5.16).



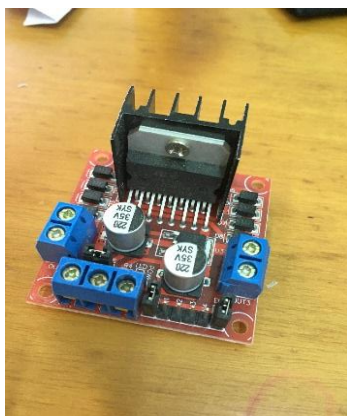
Εικόνα 5.15:



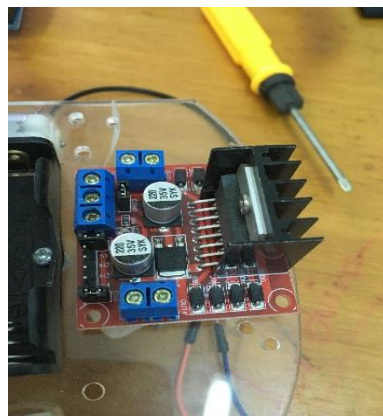
Εικόνα 5.16:

Έχουμε τελειώσει με την συναρμολόγηση του οχήματος και τώρα είναι η σειρά της συναρμολόγησης των πλακετών πάνω στο όχημα.

Θα ξεκινήσουμε με την πλακέτα L298N. Θα ήταν προτιμότερο να την τοποθετήσουμε στο μπροστά μέρος του οχήματος για να είναι κοντά με τα μοτέρ κίνησης και επειδή μπορεί να μην έχει τις τρύπες ακριβώς για στήριγμα καλό θα ήταν να βάζαμε τουλάχιστον μια βίδα για να στηρίζεται.



Εικόνα 5.17:
Πλακέτα L298N



Εικόνα 5.18:
Σύνδεση πλακέτας
L298N με την βάση
του οχήματος

Στην συνέχεια, θα τοποθετήσουμε την πλακέτα του Arduino Uno αυτήν επειδή είναι λίγο μεγαλύτερη θα την βάλουμε στο πίσω μέρος. Καλό θα ήταν να τοποθετήσουμε με δυο βίδες και την πλακέτα του Arduino διότι με την κίνηση μπορεί να πέσει.

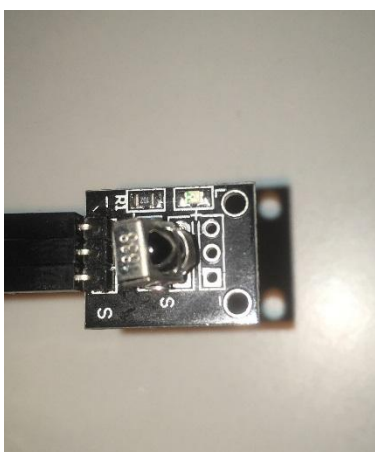


Εικόνα 5.19: πλακέτα
Arduino Uno

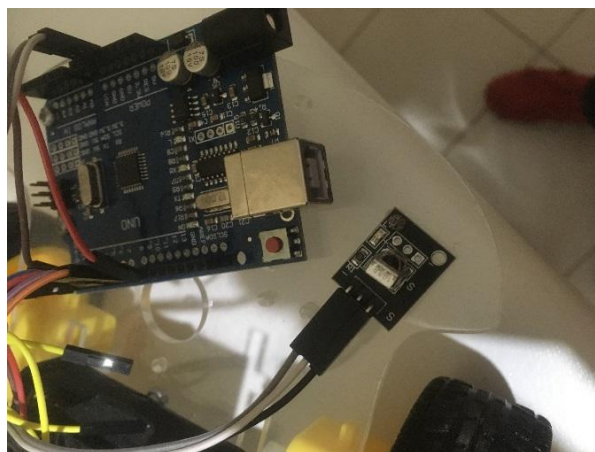


Εικόνα 5.20:
Σύνδεση πλακέτας
Arduino Uno με
την βάση του
οχήματος

Στην συνέχεια μπορείτε να δείτε πως μπορούμε να εισάγουμε την πλακέτα του ασύρματου τηλεχειριστήριου.



Εικόνα 5.21:
Πλακέτα Wireless

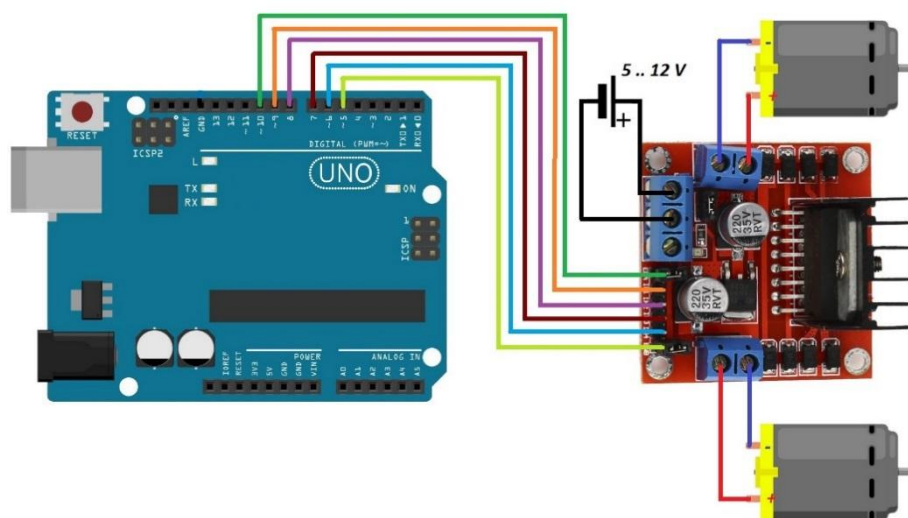


Εικόνα 5.22: Σύνδεση
πλακέτας Wireless με την βάση
του οχήματος

Στο επόμενο κεφάλαιο θα δείτε πως θα μπορέσετε να κάνετε τις συνδεσμολογίες μεταξύ των πλακετών και του οχήματος σας.

6.2 ΣΥΝΔΕΣΜΟΛΟΓΙΕΣ

Σε αυτό κεφάλαιο θα δείτε πως θα μπορέσετε να κάνετε τις συνδεσμολογίες μεταξύ της πλακέτας L298N με το Arduino Uno και της πλακέτας του τηλεχειριστηρίου με του οχήματος σας.

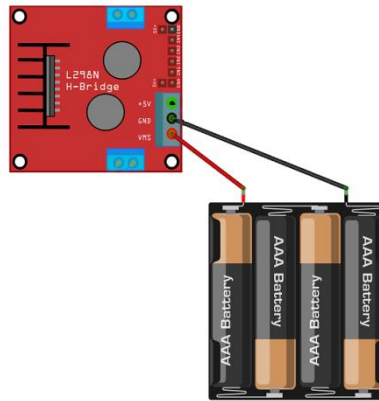


Εικόνα 6.1: Σύνδεση πλακέτας Arduino Uno με πλακέτα L298N και L298N με τα δύο μοτέρ

Όπως βλέπετε και στην εικόνα 6.1 η συνδεσμολογία δεν είναι τόσο δύσκολη (το αντίθετο θα έλεγα είναι αρκετά εύκολη) για κάποιον ο οποίος δεν γνωρίζει τι είναι το καλώδιο.

Όταν θα κατασκευάσετε το όχημα σας θα έχετε και 4 καλώδια για τα μοτέρ το μόνο που πρέπει να κάνετε είναι όταν συνδέσετε τα μοτέρ να τους κάνετε την αντιστροφή καλωδίων όπως φαίνεται και στην εικόνα. Όπως βλέπετε και στην εικόνα 6.1 το L298N. Η πλευρά του πρώτου μοτέρ θα συνδεθεί με το OUT1 και OUT2 (Motor A) και το δεύτερο μοτέρ με το OUT3 και OUT4 (Motor B). Στην θέση OUT1 θα τοποθετήσετε το κόκκινο καλώδιο του πρώτου μοτέρ και στο OUT2 το μαύρο καλώδιο του πρώτου μοτέρ. Όπως αντίστοιχα και στο δεύτερο μοτέρ που απλά θα κάνουμε μια εναλλαγή στο OUT3 θα τοποθετήσετε το μαύρο καλώδιο του δεύτερου μοτέρ και στο OUT4 το κόκκινο καλώδιο του δεύτερου μοτέρ. Όπως μπορείτε να δείτε και αναλυτικά στις εικόνες παρακάτω.

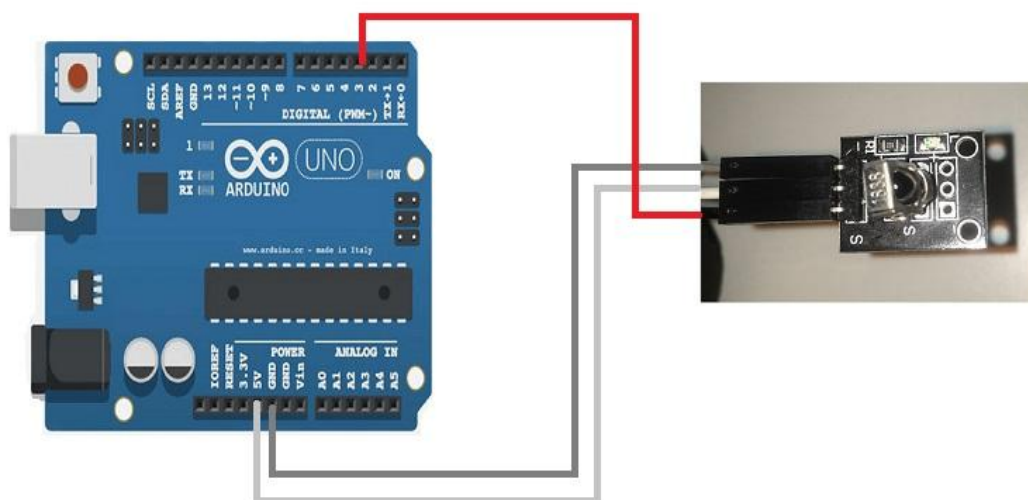
Το δεύτερο κομμάτι συνδεσμολογίας είναι αυτό μεταξύ της πλακέτας L298N και της θήκης των τεσσάρων μπαταριών. Θα συνδέσετε τα δυο καλώδια το μαύρο και το κόκκινο με το L298N. Το κόκκινο καλώδιο με την θέση 12V και το μαύρο με την



Εικόνα 6.2:
Συνδεσμολογία μεταξύ
θήκης μπαταριών και
πλακέτας L298N

GND όπως βλέπετε και στην εικόνα 6.2 παρακάτω.

Το τρίτο κομμάτι συνδεσμολογίας είναι αυτό μεταξύ του Arduino Uno και της πλακέτας L298N. Αν δούμε πάνω στην πλακέτα L298N δίπλα από την εισαγωγή του ρεύματος υπάρχουν κάποιες υποδοχές η ENA, IN1, IN2, IN3, IN4 και ENB. Οι



Εικόνα 6.3: Συνδεσμολογία μεταξύ πλακέτας Arduino Uno και Wireless

λειτουργίες που κάνουν οι υποδοχές ENA, IN1 και IN2 είναι για το πρώτο μοτέρ ή αλλιώς Motor A και οι υποδοχές IN3, IN4 και ENB για το δεύτερο μοτέρ ή αλλιώς Motor B. Από την πλευρά του Arduino Uno οι θέσεις που θα μας χρειαστούν είναι οι θέσεις 10, 9, 8, 7, 6, 5. Η θέση 10 του Arduino Uno με την θέση ENA του L298N. Η θέση 9 του Arduino Uno με την θέση IN1 του L298N. Η θέση 8 του Arduino Uno με την θέση IN2 του L298N. Η θέση 7 του Arduino Uno με την θέση IN3 του L298N. Η θέση 6 του Arduino Uno με την θέση IN4 του L298N και Η θέση 5 του Arduino Uno με την θέση ENB του L298N

Τέλος, το τέταρτο κομμάτι είναι η σύνδεση της πλακέτας τηλεκοντρόλ με την πλακέτα Arduino Uno. Το πρώτο καλώδιο της πλακέτας wireless συνδέετε με την υποδοχή GND στο Arduino Uno το δεύτερο καλώδιο με την υποδοχή 5V στο Arduino και το τρίτο και τελευταίο με την υποδοχή 3V στην ψηφιακή είσοδο του Arduino όπως βλέπουμε και στην εικόνα 6.3. Διότι αυτό έχει δηλωθεί στο πρόγραμμα σαν εισαγωγή του τηλεχειριστηρίου αυτό θα το δούμε και στο πρόγραμμα στο επόμενο κεφάλαιο.

6.3 ΠΡΟΓΡΑΜΑΤΙΣΜΟΣ ΤΗΣ ΠΛΑΚΕΤΑΣ ΤΟΥ ΟΧΗΜΑΤΟΣ

Στο κεφάλαιο αυτό θα δούμε δυο διαφορετικά προγράμματα. Στο πρώτο πρόγραμμα μπορούμε να δούμε τον κώδικα που δίνουμε στο όχημα χωρίς την χρήση του τηλεκοντρόλ. Σε αυτό το πρόγραμμα απλά εισάγουμε στην πλακέτα τη διαδρομή που θέλουμε να κάνει και πιο συγκεκριμένα αναλύουμε τον κώδικα. Στο δεύτερο

πρόγραμμα μπορούμε να δούμε πως θα εισάγουμε το τηλεχειριστήριο στο πρόγραμμα μας και αναλυτικά πως βάζουμε σε λειτουργία τα κουμπιά όπως και πως προγραμματίζουμε το κάθε κουμπί αναλυτικά. Στην συνέχεια του κεφαλαίου θα μπορέσετε να δείτε πως θα το καταφέρετε αυτό.

6.3.1 ΠΡΩΤΟ ΠΡΟΓΡΑΜΜΑ

Όπως αναφέραμε και παραπάνω στο κεφάλαιο αυτό θα δούμε πως με απλά βασικά βήματα θα μπορέσετε να προγραμματίσετε την πλακέτα Arduino Uno όπως φυσικά θα μπορέσετε να δείτε το πρόγραμμα να κάνει την ενέργεια που εμείς του δώσαμε. Με αυτόν τον κώδικα η πλακέτα Arduino UNO πιο συγκεκριμένα το όχημα κινείται σχηματίζοντας ένα τετράγωνο χωρίς κάποιο όριο

```
1 // motor one
2 int enA = 10;
3 int in1 = 9;
4 int in2 = 8;
5 // motor two
6 int enB = 5;
7 int in3 = 7;
8 int in4 = 6;
9
```

Εικόνα 7.1: Πρώτο πρόγραμμα

Αρχικά όπως βλέπετε στην εικόνα 7.1 στις γραμμές από 2 μέχρι 4 και 6 μέχρι 8 δηλώνουμε έξι ακαίρους αριθμούς με τις αντίστοιχες ονομασίες. Αν παρατηρήσετε καλύτερα θα δείτε ότι στην δήλωση των μεταβλητών οι ονομασίες που υπάρχουν είναι αυτές της εισόδου που έχει η πλακέτα L298N όπως έχουμε αναφέρει και στο προηγούμενο κεφάλαιο, αυτό το έκανα για προσωπική χρήση δεν υπάρχει κανένα απολύτως πρόβλημα στην ονομασία των μεταβλητών (για παράδειγμα θα μπορούσατε να τις ονομάσετε όπως εσείς θελήσετε και φυσικά σας το επιτρέπει και το πρόγραμμα).

Στην συνέχεια όπως βλέπετε και στην εικόνα 7.2 δηλώνετε τις λειτουργίες pin ως εξόδους, τις μεταβλητές που δηλώσαμε νωρίτερα αυτό γίνεται μέσα στην συνάρτηση setup() όπως βλέπουμε και παρακάτω στην εικόνα 7.2. Στις γραμμές 12 μέχρι και 17 δηλώνονται οι λειτουργίες pin ως εξοδοί.

```
10 void setup() {
11   // set all the motor control pins to outputs
12   pinMode(enA, OUTPUT);
13   pinMode(enB, OUTPUT);
14   pinMode(in1, OUTPUT);
15   pinMode(in2, OUTPUT);
16   pinMode(in3, OUTPUT);
17   pinMode(in4, OUTPUT);
18 }
```

Εικόνα 7.2: Πρώτο πρόγραμμα

Στην συνέχεια θα δείτε στην εικόνα 7.3 την συνάρτηση με την οποία θα εισάγετε των κώδικα της επιθυμίας σας. Στο παράδειγμα θα δούμε πως το όχημα μας θα κάνει εάν

τετράγωνο χωρίς σταματημό μέχρι να τελειώσει η ενέργεια των μπαταριών (αυτή είναι η συνάρτηση με το όνομα demoOne()). Όπως θα δούμε και στην παρακάτω εικόνα 7.3 τη λειτουργία που κάνει (αναλυτικά).

```
19 void demoOne()
20 {
21   // turn on motor A
22   digitalWrite(in1, LOW);
23   digitalWrite(in2, HIGH);
24   // set speed to 200 out of possible range 0~255
25   analogWrite(enA, 100);
26   // turn on motor B
27   digitalWrite(in3, LOW);
28   digitalWrite(in4, HIGH);
29   analogWrite(enB, 100); // set speed to 200 out of possible range 0~255
30   delay(1000);           //delay
31   // now turn off motors
32   digitalWrite(in1, LOW);
33   digitalWrite(in2, LOW);
34   digitalWrite(in3, LOW);
35   digitalWrite(in4, LOW);
36   delay(1000);           //delay
37
38   // turn on motor A
39   digitalWrite(in1, LOW);
40   digitalWrite(in2, HIGH);
41   analogWrite(enA, 200); // set speed to 200 out of possible range 0~255
42   delay(510);
43   digitalWrite(in1, LOW);
44   digitalWrite(in2, LOW);
45 }
```

Εικόνα 7.3: Πρώτο πρόγραμμα

Στις γραμμές 22,23 και 25 ενεργοποιούμε τον πρώτο κινητήρα και κάνοντας μπροστά κίνηση βάζοντας το in1 σε χαμηλή λειτουργία ή αλλιώς τερματική και την έξοδο in2 σε υψηλή με λίγα λογία με αυτόν τον τρόπο το όχημα πηγαίνει μπροστά. Στην γραμμή 25 δηλώνουμε την ταχύτητα που θέλουμε να έχει το μοτέρ μας, η ταχύτητα πρέπει να κυμαίνεται από 0 μέχρι και 255. Αντίστοιχα στις γραμμές 27,28 και 29 βλέπετε ότι γίνεται ακριβώς το ίδιο πράγμα και με τον δεύτερο κινητήρα (δηλαδή επαναλαμβάνεται σχεδόν ο ίδιος κώδικας).

Στην συνέχεια, στην γραμμή 30 βλέπουμε την εντολή delay(1000); Ανάλογα με το χρονικό διάστημα που βρίσκετε μέσα στην παρένθεση αυτή η εντολή καθυστερεί την ενέργεια που γίνεται. Σε αυτή την περίπτωση η καθυστέρηση είναι 1 sec.

Στις γραμμές 32 μέχρι και 35 βλέπετε ότι ακινητοποιεί και τα δυο μοτέρ βάζοντας τα σε χαμηλή λειτουργία LOW.

Οπότε από την γραμμή 22 μέχρι και 36 το πρόγραμμα πηγαίνει μπροστά το όχημα για ένα διάστημα του 1sec και σταματάει για ακόμα 1sec.

Στην συνέχεια βλέπετε ότι ενεργοποιείτε το ένα μοτέρ και ο συνδυασμός ταχύτητας και καθυστέρησης στρίβει το όχημα 90° μοίρες και στην συνέχεια απενεργοποιεί το μοτέρ.

Στην εικόνα 7.4 παρακάτω θα δείτε την συνάρτηση η οποία επαναλαμβάνει την συνάρτηση demoOne() χωρίς κάποιο όριο

```
47 void loop()  
48 {  
49     demoOne();  
50  
51 }
```

Εικόνα 7.4: Πρώτο πρόγραμμα

Το πρόγραμμα θα έχει σαν αποτέλεσμα όταν εκτελεσθεί, το όχημα να κάνει την διαδρομή ενός τετραγώνου. Θα μπορέσετε να βρείτε ολόκληρο το πρόγραμμα στην τελευταία σελίδα της εργασίας.

6.3.2 ΔΕΥΤΕΡΟ ΠΡΟΓΡΑΜΜΑ (κώδικας του τηλεχειριστηρίου (case))

Στο κεφάλαιο αυτό θα δείτε τον κώδικα που έχει το τηλεκατευθυνόμενο όχημα. Στο πρόγραμμα αυτό θα δείτε αναλυτικά την κάθε κίνηση που μπορεί να κάνει το όχημα, όπως για παράδειγμα να πηγαίνει μπροστά, πίσω αλλά και να στριβεί δεξιά και αριστερά. Στο πρόγραμμα αυτό θα δείτε ακόμα εντολές επανάληψης και case όπως και το κλήση διαφορετικών συναρτήσεων που έχουμε δημιουργήσει για τις ενέργειες του τηλεκατευθυνόμενου οχήματος. Ακόμα έχετε διαφορετικούς κώδικες στα κουμπιά όπως για παράδειγμα όταν πατάτε το 1 το όχημα κάνει ένα τετράγωνο ή αν πατήσετε το κουμπί “ok” να κάνει περιστροφή. Όλα τα παραπάνω θα αναλυθούν εκτενέστερα στην συνέχεια.

Το πρόγραμμα που βλέπουμε στην εικόνα παρακάτω είναι αυτό που κάνει το όχημα να κινείται μπροστά για το διάστημα των 2 sec με ταχύτητα στα μοτέρ 200.

```
29 void straight()
30 {
31
32     digitalWrite(in1, LOW);           //      /\      //
33     digitalWrite(in2, HIGH);          //      /  \      //
34     digitalWrite(in3, LOW);           //      /   \      //
35     digitalWrite(in4, HIGH);          //      |      //
36     analogWrite(enA, 200);            //      |      //
37     analogWrite(enB, 200);            //      |      //
38     delay(2000);
39
40     digitalWrite(in1, LOW);           //      _ _ _ _ _ //
41     digitalWrite(in2, LOW);           // | | | | | _ _ _ _ _ //
42     digitalWrite(in3, LOW);           // | | | | | _ _ _ _ _ //
43     digitalWrite(in4, LOW);           // | _ _ _ _ _ | _ _ _ _ _ //
44
45 }
```

Εικόνα 7.5: Δεύτερο πρόγραμμα straight()

Το πρόγραμμα που βλέπουμε στην εικόνα παρακάτω είναι αυτό που κάνει το όχημα να κινείται για 2 sec προς τα πίσω και αυτό με ταχύτητα στα μοτέρ 200.

```
47 void backOne()
48 {
49
50     digitalWrite(in1, HIGH);          //      |      //
51     digitalWrite(in2, LOW);            //      |      //
52     digitalWrite(in3, HIGH);           //      |      //
53     digitalWrite(in4, LOW);            //      \  /      //
54     analogWrite(enA, 200);             //      \  /      //
55     analogWrite(enB, 200);             //      \|/      //
56     delay(2000);
57
58     digitalWrite(in1, LOW);            //      _ _ _ _ _ //
59     digitalWrite(in2, LOW);            // | | | | | _ _ _ _ _ //
60     digitalWrite(in3, LOW);            // | | | | | _ _ _ _ _ //
61     digitalWrite(in4, LOW);            // | _ _ _ _ _ | _ _ _ _ _ //
62
63 }
```

Εικόνα 7.6: Δεύτερο πρόγραμμα backOne()

Στην συνέχεια, στην εικόνα που ακολουθεί βλέπουμε τον κώδικα που κάνει το όχημα να στρίβει από την αριστερή πλευρά δηλαδή να κάνει μια στροφή 90° προς τα αριστερά.

```

78 void leftOne()
79 {
80     digitalWrite(in3, LOW);           //      <           //
81     digitalWrite(in4, HIGH);          //      <           //
82     analogWrite(enA, 200);            // < < < < < < < //
83     delay(510);                       //      <           //
84                                     //      <           //
85     delay(510);
86     digitalWrite(in3, LOW);
87     digitalWrite(in4, LOW);
88 }

```

Εικόνα 7.7: Δεύτερο πρόγραμμα leftOne()

Στην εικόνα που ακολουθεί βλέπουμε τον κώδικα που κάνει το όχημα μας να στρίβει από την δεξιά πλευρά (πιο απλά να κινηθεί το ένα μοτέρ έτσι ώστε να κάνει μια στροφή 90° προς τα δεξιά).

```

65 void rightOne()
66 {
67                                     //          >          //
68     digitalWrite(in1, LOW);          //          >          //
69     digitalWrite(in2, HIGH);          // > > > > > > > //
70     analogWrite(enA, 200);            //          >          //
71                                     //          >          //
72     delay(510);
73     digitalWrite(in1, LOW);
74     digitalWrite(in2, LOW);
75 }
76 }

```

Εικόνα 7.8: Δεύτερο πρόγραμμα rightOne()

Στην συνέχεια θα δούμε λίγο πιο περίπλοκα προγράμματα που μέσα σε αυτά υπάρχουν και επανάληψης. Στην εικόνα που υπάρχει στην συνέχεια θα δείτε την συνάρτηση της περιστροφής. Η διαφορά που έχει από την συνάρτηση της ευθείας (straight()) είναι ότι τα μοτέρ κάνουν αντίθετη κίνηση για ένα διάστημα μικρότερου του 1 sec σχεδόν του μισού.

```

90 void rotation()
91 {
92
93     digitalWrite(in1, LOW);           //          //
94     digitalWrite(in2, HIGH);          //  <-----  //
95     digitalWrite(in3, HIGH);          //  |          /\  //
96     digitalWrite(in4, LOW);           //  |          |  //
97     analogWrite(enA, 200);             //  |          |  //
98     analogWrite(enB, 200);            //  \  /          |  //
99     delay(520);                        //  ----->    //
100
101     digitalWrite(in1, LOW);            //  _ _ _ _ _  //
102     digitalWrite(in2, LOW);            //  | | | | |  //
103     digitalWrite(in3, LOW);            //  | | | | |  //
104     digitalWrite(in4, LOW);            //  | | | | |  //
105
106 }

```

Εικόνα 7.9: Δεύτερο πρόγραμμα rotation()

Στην εικόνα που ακολουθεί θα δείτε την συνάρτηση με το όνομα square() η οποία κάνει ένα τετράγωνο. Στην συνάρτηση αυτή βλέπουμε μια επανάληψη (για τις 4 επαναλήψεις που χρειάζονται) όπως και μέσα στην επανάληψη την κλήση δυο συναρτήσεων της straight() όπως και της leftOne() (για τις κινήσεις που

```

108 void squareOne()
109 {
110     for(int i=0; i<4;i++)
111     {
112         straight();
113         delay(800);
114         leftOne();
115         delay(500);
116     }
117 }

```

εναλλάσσονται).

Εικόνα 7.10: Δεύτερο πρόγραμμα

```

1 #include <IRremote.h>
2 int CONDROL_PIN = 3;
3 // motor one
4 int enA = 10;
5 int in1 = 9;
6 int in2 = 8;
7 // motor two
8 int enB = 5;
9 int in3 = 7;
10 int in4 = 6;
11
12 IRrecv irrecv(CONDROL_PIN);
13 decode_results results;
14
15 void setup() {
16   pinMode(CONDROL_PIN, INPUT); // initialize the digital pin as an output.
17
18   pinMode(enA, OUTPUT);
19   pinMode(enB, OUTPUT);
20   pinMode(in1, OUTPUT); // set all the motor control pins to outputs
21   pinMode(in2, OUTPUT);
22   pinMode(in3, OUTPUT);
23   pinMode(in4, OUTPUT);
24
25   irrecv.enableIRIn(); // Start the receiver
26   Serial.begin(9600);
27 }

```

Εικόνα 7.11: Δεύτερο πρόγραμμα

Στην επόμενη εικόνα θα δούμε της δηλώσεις που γίνονται στο πρόγραμμα.

Στην γραμμή 1 εισάγουμε την βιβλιοθήκη <IRremote.h> η οποία είναι για το ασύρματο τηλεχειριστήριο. Στην γραμμή 2 έχουμε δήλωση μιας μεταβλητής τύπου int (δηλαδή μια μεταβλητή που αποθηκεύει ακέραιες τιμές) που της εκχωρούμε την τιμή 3, δηλαδή ως ψηφιακή πόρτα του Arduino για την εισαγωγή εντολών από το τηλεχειριστήριο στην θέση 3. Στην γραμμή 12 δηλώνουμε ένα αντικείμενο τύπου IRrecv και της εκχωρούμε την τιμή της CONTROL_PIN στην συγκεκριμένη περίπτωση την τιμή 3. Στην γραμμή 13 δηλώνουμε μια μεταβλητή τύπου decode_results η οποία θα κρατάει το αποτέλεσμα της εντολής που δέχεται από το τηλεχειριστήριο αποκωδικοποιημένα. Στην γραμμή 25 ενεργοποιείτε ο αποδέκτης (receiver). Στην γραμμή 26 ενεργοποιείτε η συριακή θήρα επικοινωνίας. Αυτή η εντολή δέχεται μια παράμετρο η οποία εκφράζει τον ρυθμό με τον οποίο θα μεταδίδοντας τα bits(μια τιμή στα 9600 είναι συνηθώς αρκετή)

Στην συνάρτηση επανάληψης Loop βλέπουμε στην γραμμή 161 ελέγχει αν έχει λάβει εντολή από το τηλεχειριστήριο και αν ναι τότε τρέχει τις γραμμές 163 και 165. Η

γραμμή 163 καλεί την συνάρτηση buttons και η γραμμή 165 περιμένει μέχρι να δεχτεί την επόμενη εντολή.

```
159 void loop() {  
160     int i=0;  
161     if (irrecv.decode(&results)) {  
162  
163         buttons();  
164  
165         irrecv.resume();           // Receive the next value  
166  
167     }  
168 }  
169
```

Εικόνα 7.12: Δεύτερο πρόγραμμα loop()

παρακάτω θα δείτε την δήλωση των κουμπιών δηλαδή την ονομασία που διαθέτει το κάθε ένα.

Μπροστά: **0xFF629D**
Αριστερά: **0xFF22DD**
Δεξιά: **0xFFC23D**
Κάτω: **0xFFA857**
OK: **0xFF02FD**
Ένα(1): **0xFF6897**
Δύο(2): **0xFF9867**
Τρία(3): **0xFFB04F**
Τέσσερα(4): **0xFF30CF**
Πέντε(5): **0xFF18E7**
Έξι(6): **0xFF7A85**
Επτά(7): **0xFF10EF**
Οκτώ(8): **0xFF38C7**
Εννέα(9): **0xFF5AA5**



Εικόνα 4.4.1:
Τηλεχειριστήριο

Η λογική αυτής της συνάρτησης είναι να ελέγξει πιο κουμπί πατήθηκε και να εκτελέσει την αντίστοιχη λειτουργία.

```

558 void buttons() // takes action based on IR code received describing Car MP3 IR codes
559 {
560
561     switch(results.value){
562         case 0xFF629D: //koumbi mprosta
563             straight();
564             Serial.println("pige mbrosta ");
565
566             break;
567         case 0xFF22DD: // aristero koumbi
568
569             leftOne();
570             Serial.println("pige aristera ");
571             break;
572
573         case 0xFF02FD: // koumbi ok
574             rotation();
575             Serial.println("ekane anastrofi ");
576             break;
577
578         case 0xFFC23D: //deksi koumbi
579             rightOne();
580             Serial.println("estripse deksia ");
581             break;
582
583         case 0xFFA857: // kato koumbi
584             backOne();
585             Serial.println("pige piso ");
586             break;
587
588         case 0xFF6897: // koumpaki 1
589             squareTwo();
590             Serial.println(" tetragono me deksia peristrofi ");
591             break;
592
593         case 0xFF9867: // koumbi 2
594             updown();
595             Serial.println(" pano kato ");
596             break;
597
598         case 0xFFB04F: // koumbaki 3
599             squareOne();
600             Serial.println(" tetragono me aristeri peristrofi ");
601             break;
602
603         case 0xFF30CF: //koumbaki 4
604             updown();
605             Serial.println(" pano kato");
606             break;
607
608         case 0xFF18E7: // koumbaki 5
609             octapus();
610             Serial.println(" sxediash stayroy ");

```

```

610     Serial.println(" sxediash stayroy ");
611     break;
612
613     case 0xFF7A85:                                     //koumbaki 6
614         downup();
615         Serial.println(" pano kato ");
616         break;
617
618     case 0xFF10EF:                                     //koumbaki 7
619         house();
620         Serial.println(" sxediazei spiti ");
621         break;
622
623     case 0xFF38C7:                                     //koumbaki 8
624         eight();
625         Serial.println(" sxediash okto ");
626         break;
627
628     case 0xFF5AA5:                                     // koumbaki 9
629         functioneight();
630         Serial.println(" kanei to 8 me sinartisis ");
631         break;
632     default:
633         Serial.println();
634 }
635
636 }

```

Εικόνα 7.13: Δεύτερο πρόγραμμα Buttons()

7) ΔΟΚΙΜΕΣ

Στο κεφάλαιο αυτό θα δούμε τι θα κάνει το όχημα μας, τα αποτελέσματα όλων των ενεργειών και πόσο καλά κάνει τις ενέργειες αυτές. Θα δούμε αναλυτικά τα προβλήματα που θα αντιμετωπίσουμε στην πραγματοποίηση του έργου αυτού τόσο στο πρακτικό κομμάτι όσο και στο προγραμματιστικό.

Εγκατάσταση της πλακέτας Arduino

Ένα σημαντικό πρόβλημα που θα συναντήσετε όταν κάνετε την εγκατάσταση του Arduino και θέλετε να βάλετε μια πλακέτα που είναι απομίμηση Arduino θα χρειαστεί να κατεβάσετε ένα αρχείο που θα το βρείτε στο διαδίκτυο και είναι κάτι ιδιαίτερα εφικτό. Αμέσως μετά θα μπορέσετε να συνδέσετε την πλακέτα σας με τον υπολογιστή.

Πόρτες υποδοχής Arduino

Πολλές φορές ξεχνάμε να κάνουμε δήλωση της πόρτας στο πρόγραμμα ή κάποιες φορές το προγραμματιστικό περιβάλλον μπορεί να μην κρατάει πάντα δηλωμένη την πόρτα στο πρόγραμμα. Είναι καλό αν δείτε ότι το πρόγραμμα είναι σωστό και κάνει compile αλλά όταν το ανεβάζετε σας βγάζει λάθος, τις περισσότερες φορές είναι επειδή δεν έχει δηλωθεί η πόρτα της εξόδου του Arduino. Το μόνο που έχετε να κάνετε είναι απλά να το συνδέσετε ξανά.

Βιβλιοθήκες

Το φαινόμενο να μην μπορούμε να κάνουμε compile, στο πρόγραμμά μας βγάζει ότι η βιβλιοθήκη που έχουμε δεν υπάρχει ή ότι δεν την τρέχει το πρόγραμμα. Το πρόβλημα αυτό το λύνουμε και παραπάνω εισάγοντας την βιβλιοθήκη που χρειαζόμαστε. Είναι πολύ εύκολη η εγκατάσταση της βιβλιοθήκης και για περισσότερες πληροφορίες μπορείτε να απευθυνθείτε στο 4^ο κεφάλαιο στην τελευταία κατηγορία που είναι οι βιβλιοθήκες.

Συντακτικά λάθη

Πολλές φορές όλοι μας κάνουμε σημαντικά λάθη καλό είναι να ελέγχουμε το πρόγραμμά μας. Σίγουρα αν δούμε το πρόγραμμα να μην μας τρέχει και να μας βγάζει λάθος τότε πιθανό είναι να έχουμε κάνει κάποιο λάθος στην σύνταξη του προγράμματος ή ακόμα και στον κώδικα που έχουμε γράψει.

Συναρμολόγηση οχήματος

Στο 6^ο κεφάλαιο βλέπουμε πως γίνεται η συναρμολόγηση του οχήματος. Πολλές φορές δημιουργούνται προβλήματα γιατί είτε έχετε μπερδευτεί είτε έχετε βάλει λάθος τις βίδες είτε ακόμα έχετε βάλει πολύ κοντά τις πλακέτες και ακουμπάνε με τις ρόδες με τις άλλες πλακέτες κλπ. Καλό είναι για να αποφύγετε αυτά τα βασικά προβλήματα να ακολουθήσετε αναλυτικά τις εντολές του 6^{ου} κεφαλαίου ή αλλιώς να δείτε κάποιο βίντεο στο YouTube.

Συναρμολόγηση οχήματος L298N

Πολλές φορές μπορεί να κάνετε λάθος στην συνδεσμολογία του μοτέρ με την πλακέτα L298N διότι δεν γίνεται η εναλλαγή καλωδίων. Είτε στην εισαγωγή του

ρεύματος να έχετε βάλει λάθος τα καλώδια. Κάλο θα ήταν να εκτελέσετε τις εντολές αναλυτικά χωρίς να βιάζεστε όπως βλέπετε στο 6^ο κεφάλαιο με αυτόν τον τρόπο θα μπορέσετε να αντιμετωπίσετε τα προβλήματα αυτά διότι οι ενέργειες δεν θα γίνουν σωστά.

Συναρμολόγηση ασύρματου τηλεκοντρόλ

Μεγάλη προσοχή στην συνδεσμολογία της πλακέτας του ασύρματου τηλεκοντρόλ. Πολλές φορές γίνετε λάθος στην συνδεσμολογία μεταξύ των δυο καλωδίων του ρεύματος. Καλό είναι να δείτε ξανά την συνδεσμολογία μεταξύ των δυο αυτών πλακετών. Όπως και πολλές φορές την δήλωση της υποδοχής στο πρόγραμμα μας έχουμε δηλώσει την υποδοχή 3 στο πρόγραμμα μας οπότε πρέπει και στην πλακέτα του Arduino Uno να είναι το τρίτο καλώδιο συνδεδεμένο στην θύρα με τον αριθμό 3.

Συνδεσμολογίες καλωδίων

Πολλές φορές συναντάμε πρόβλημα με τις συνδεσμολογίες και φυσικό είναι κάποιες ενέργειες να μην γίνονται. Κάλο θα είναι να ελέγξετε ξανά τις εντολές και να δείτε μήπως υπάρχει κάποιο πρόβλημα εκεί, επίσης στις θέσεις ENA και ENB της πλακέτας L298N είναι δύσκολος ο τρόπος να βάλετε το καλώδιο αλλά όχι ακατόρθωτος. Αν τελικά δεν μπορέσετε, προτιμότερο είναι να απευθυνθείτε σε έναν ηλεκτρολογικό κατάστημα είτε σε κάποιον ηλεκτρολόγο να σας βοηθήσει.

Μπαταρίες

Καλό θα ήταν πριν από κάθε ενέργεια, να ελέγξετε τις μπαταρίες (είτε αυτή είναι του τηλεκοντρόλ, είτε της πλακέτας Arduino Uno, το USB, είτε την μπαταρία της υποδοχής ρεύματος). Καθώς επίσης και τις τέσσερις κεντρικές που συνδέονται με το L298N που μεταβάλλει την ενεργεία για να κινήσει τα δυο μοτέρ.

Τι λειτουργίες θέλει να κάνει

Στο όχημα μας μπορούμε να δώσουμε μεγάλη γκάμα προγραμμάτων έτσι ώστε να κάνει και τις αντίστοιχες ενέργειες που εμείς θέλουμε. Όπως είδαμε και παραπάνω στον κώδικα στο πρώτο πρόγραμμα δίνουμε εντολές στο αυτοκίνητο να κάνει ένα τετράγωνο ή ένα τρίγωνο ή να κάνει αναστροφή και πολλές άλλες κινήσεις. Οι δυνατότητες που έχει είναι πάρα πολλές και είναι αρκετά εύκολες για να πραγματοποιηθούν. Στο δεύτερο πρόγραμμα με το τηλεχειριστήριο βλέπετε ακόμα περισσότερες επιλογές, κάνοντας μια μεγάλη γκάμα κινήσεων και αυτό το επιτυγχάνει βάζοντας σε ένα κουμπί τις εντολές που θέλετε να κάνει (πιο απλά τον κώδικα που θέλετε χρησιμοποιήσει).

Πόσο καλά κάνει τις λειτουργίες αυτές

Για να πραγματοποιηθούν οι εντολές αυτές είναι αρκετά εύκολο αλλά υπάρχει ένα βασικό πρόβλημα. Αυτό είναι τα φυσικά εμπόδια όπως το έδαφος, δηλαδή άλλη κίνηση θέλει το όχημα στο πάτωμα, άλλη για το χώμα και άλλη για την άσφαλτο. Το όχημα μας είναι προγραμματισμένο σε κλειστό χώρο, με αποτέλεσμα το πείραμα να μην έχει κάποια άλλη φυσική πίεση και επίσης όπως ξέρουμε το αυτοκίνητο το προγραμματίσαμε ανάλογα με τα αποτελέσματα που βλέπουμε και τις κινήσεις που

αυτό κάνει. Δεν υπάρχει πρόγραμμα που θα εμφανίσει ακριβώς τις κινήσεις του οχήματος.

ΠΡΟΣΟΧΗ!!!

Θα πρέπει οι πλακέτες να μην έρχονται σε επαφή, γιατί υπάρχει κίνδυνος να βραχυκυκλωθούν μεταξύ τους και να καταστραφούν.

8)ΣΥΜΠΕΡΑΣΜΑΤΑ

Σε γενικές γραμμές με όλα τα παραπάνω στοιχεία που έχετε στην διάθεση σας είναι εφικτό για τον καθένα χρήστη με μια πολύ μικρή εμπειρία στον προγραμματισμό είτε ακόμα και καθόλου να μπορέσει να κατασκευάσει το όχημα αυτό. Από την εγκατάσταση του λογισμικού του Arduino ακόμα και την έξοδο της πόρτας που θα χρησιμοποιήσουμε. Τις βιβλιοθήκες που πρέπει να χρησιμοποιήσουμε και που θα μας βοηθήσουν στον προγραμματισμό του οχήματος σας. Στην εργασία θα βρείτε τον κώδικα που θα χρειαστείτε ακόμα τις ενέργειες που πρέπει να κάνετε στο πρόγραμμα έτσι ώστε να περαστεί στην πλακέτα του Arduino Uno, για να εκτελέσει τις ενέργειες. Ακόμα οι συνδεσμολογίες μεταξύ των πλακετών που έχουμε όπως, μπορείτε να δείτε και στο 6^ο κεφάλαιο αλλά και ένα ακόμη πιο σημαντικό κομμάτι, το κομμάτι που είναι η κατασκευή του οχήματος μας από την αρχή μέχρι το τέλος. Τα εργαλεία και τα πράγματα που θα χρειαστούμε και το που θα απευθυνθείτε να τα αγοράσετε μπορείτε να τα βρείτε όλα στο 5^ο κεφάλαιο.

Το συμπέρασμα μου είναι ότι είναι κάτι πολύ ενδιαφέρον για όλους όσους ενδιαφέρονται να ασχοληθούν με τον προγραμματισμό ενσωματωμένων συστημάτων και ιδιαίτερα με την πλακέτα Arduino Uno. Όπως επίσης και αυτών που δεν θέλουν να ασχοληθούν τόσο πολύ με τον προγραμματισμό πλακετών αλλά θα ήθελαν να φτιάξουν ένα ωραίο έργο για χόμπι, είναι αρκετά εύκολο και δεν θα χρειαστεί μεγάλη διάρκεια χρόνου για την κατασκευή του έργου αυτού. Σίγουρα θα βρεθούμε αντιμέτωποι με κάποια εμπόδια τα οποία μπορείτε να τα αντιμετωπίσετε με την εκτέλεση αναλυτικά των βημάτων αυτών. Πολλές λύσεις στα προβλήματα μας μπορούμε να τα βρούμε διαβάζοντας το 7^ο κεφάλαιο. Μια καλή βοήθεια είναι και το διαδίκτυο, όπως δηλαδή αν το πρόβλημα είναι μηχανικό, να δούμε διάφορα άρθρα στο διαδίκτυο ενώ αν είναι κάποιο προϊόν ελαττωματικό, μπορούμε να απευθυνθούμε στην εταιρία που το αγοράσαμε. Επίσης υπάρχουν και πολλά βίντεο που μπορούσαν να μας βοηθήσουν. Αν αντιμετωπίσουμε πρόβλημα στο πρόγραμμα, τότε θα απευθυνθούμε στο αντίστοιχο κεντρικό site του Arduino ή και σε άλλες σχετικές ιστοσελίδες.

Σε όλα τα προβλήματα μπορείτε να βρείτε τη λύση από τα πιο μεγάλα μέχρι και τα πιο μικρά. Διαθέτοντας λίγη θέληση και προσπάθεια θα μπορέσετε να τα καταφέρετε να κατασκευάσετε το όχημα το οποίο είναι κάτι ενδιαφέρον και δημιουργικά. Όποιος έχει την θέληση να ασχοληθεί, θα καταφέρει να κατασκευάσει το έργο αυτό. Είναι ένας καλός τρόπος και ένα καλό μάθημα να ξέρετε πως κατασκευάζονται σήμερα τα ρομπότ και τα τηλεκατευθυνόμενα οχήματα. Όπως επίσης και να έχουν μια ιδέα πως λειτουργούν τα ενσωματωμένα συστήματα γύρω τους που μέχρι και χθες οι περισσότεροι άνθρωποι φυσικά δεν γνώριζαν ότι υπάρχουν.

Όλο αυτό το project είναι πραγματικά ευχάριστο και σας δίνει την δυνατότητα να πειραματιστείτε με την συσκευή σας (δηλαδή το τηλεκατευθυνόμενο όχημα που εσείς φτιάξατε και να κάνετε πολλές διαφορετικές λειτουργίες). Όπως να προγραμματίζετε το όχημα σας ώστε να κάνει μια συγκεκριμένη κατεύθυνση με το πάτημα ενός κουμπιού ή να κάνει όλες τις λειτουργίες που κάνει ένα τηλεκατευθυνόμενο όχημα αλλά και να εκτελεί προγραμματισμένες κινήσεις. Επίσης, μπορούμε να ρυθμίσουμε την ταχύτητα που θέλουμε να έχει το όχημα μας και την οικονομία ενεργείας που θέλουμε να κάνει.

Η εργασία αυτή σας δίνει το δικαίωμα να αναπτύξετε και άλλο το έργο σας εισάγοντας αισθητήρες ή Wi-Fi και πολλά άλλα. Είναι μια πλακέτα την οποία μπορείς να την επεκτείνεις πάρα πολύ και είναι όπως είπαμε και πριν συμβατή με πολλές άλλες πλακέτες. Βέβαια όσο εισάγεις πράγματα σε μια πλακέτα τόσο πιο δύσκολή είναι και στην κατασκευή αλλά και πιο δυσανάγνωστος ο προγραμματιστικός κώδικας. Άλλο ένα προνόμιο είναι ότι μαθαίνεις να προγραμματίζεις πολύ καλύτερα και πιο συγκεκριμένα στο προγραμματιστικό περιβάλλον του Arduino IDE.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Βιβλίο ψηφιακά ηλεκτρονικά Roger L.Tokheim σελίδα 352
- [2] <http://cgi.di.uoa.gr/~std04013/Welcome.html>
- [3] Βιβλίο ψηφιακά ηλεκτρονικά Roger L.Tokheim σελίδα 406
- [4] <https://el.wikipedia.org/wiki/EEPROM>
- [5] <http://tymkrs.tumblr.com/post/12328620000/rom-prom-eprom-eprom>
- [6] <https://www.engineersgarage.com/tutorials/difference-between-microprocessor-and-microcontroller>
- [7] Βιβλίο ψηφιακά ηλεκτρονικά Roger L.Tokheim σελίδα 407
- [8] Βιβλίο ενσωματωμένα συστήματα “ο αθέατος ψηφιακός κόσμος ” του Μηνά Δασυγένη και Δημήτριου Σουντρή σελίδα 35 μέχρι 36.
<https://repository.kallipos.gr/pdfviewer/web/viewer.html?file=/bitstream/11419/2247/1/book.compress.pdf>
- [9] Βιβλίο ενσωματωμένα συστήματα “ο αθέατος ψηφιακός κόσμος ” του Μηνά Δασυγένη και Δημήτριου Σουντρή σελίδα 37 μέχρι 38.
<https://repository.kallipos.gr/pdfviewer/web/viewer.html?file=/bitstream/11419/2247/1/book.compress.pdf>
- [10] <https://en.wikipedia.org/wiki/Atmel>
- [11] https://en.wikipedia.org/wiki/Microchip_Technology
- [12] https://en.wikipedia.org/wiki/Texas_Instruments
- [13] <http://www.techyv.com/article/top-10-microcontroller-manufacturing-companies/>
- [14] <https://microcontrollermanufacturer.wordpress.com/2013/09/25/top-ten-microcontroller-manufacturing-companies-across-the-globe-use-the-companies-listed-below/>
- [15] https://en.wikipedia.org/wiki/List_of_common_microcontrollers
- [16] <https://el.wikipedia.org/wiki/Arduino>
- [17] <http://mcuhq.com/17/getting-started-arduino>
- [18] Βιβλίο ενσωματωμένα συστήματα “ο αθέατος ψηφιακός κόσμος ” του Μηνά Δασυγένη και Δημήτριου Σουντρή σελίδα 383 μέχρι 393.
<https://repository.kallipos.gr/pdfviewer/web/viewer.html?file=/bitstream/11419/2247/1/book.compress.pdf>
- [19] <https://www.arduino.cc/en/Main/Products>

[20] Βιβλίο ενσωματωμένα συστήματα “ο αθέατος ψηφιακός κόσμος ” του Μηνά Δασυγένη και Δημήτριου Σουντή σελίδα 389 μέχρι 393.

<https://repository.kallipos.gr/pdfviewer/web/viewer.html?file=/bitstream/11419/2247/1/book.compress.pdf>

[21] <https://playground.arduino.cc/Main/DevelopmentTools>

[21.1]https://www.google.gr/search?q=PROGRAMINO+IDE&rlz=1C1AVFA_enGR736GR737&espv=2&source=lnms&tbm=isch&sa=X&ved=0ahUKEwig2ITNjYjVAhWBKMAKHRTBD9YQ_AUIBigB&biw=1366&bih=662#imgsrc=UpgeLHc_pIeeDM
:

[21.2] <http://forum.arduino.cc/index.php?topic=128023.0>

[21.3] <https://playground.arduino.cc/Main/DevelopmentTools>

[21.4] <http://www.embrio.io/>

[21.5] <https://playground.arduino.cc/Main/DevelopmentTools>

[21.6]<https://create.arduino.cc/projecthub/luigifcerfeda/iot-ambient-light-zerynth-lamp-d09ea3>

[21.7] <https://dzone.com/articles/platformio-iot-integrated-platform>

[21.8] <http://www.instructables.com/id/Deviot-the-IoT-Plugin-for-Sublime-Text/>

[22] <https://www.arduino.cc/en/Guide/Libraries>

ΠΗΓΕΣ ΓΙΑ ΥΛΙΚΑ ΠΟΥ ΘΑ ΧΡΕΙΑΣΤΟΥΜΕ

[23] <https://www.aliexpress.com/cp/arduino-car-kit-online-shopping.html>

[24]https://www.aliexpress.com/wholesale?catId=0&initiative_id=SB_20170725012303&SearchText=arduino+uno

[25]https://www.aliexpress.com/wholesale?catId=400103&initiative_id=AS_20170724233531&SearchText=l298n

[26]<http://www.ebay.com/itm/1Pcs-Infrared-IR-Wireless-Remote-Control-Module-Kits-for-Arduino-NEW-/141976901262?hash=item210e7b968e:g:CiAAAOSw-7RVC3ii>

[27]<https://www.ebay.com/sch/i.html?from=R40&trksid=p2047675.m570.l1313.TR0.TRC0.H0.X9V+Battery+Snap+Connector.TRS0&nkw=9V+Battery+Snap+Connector&sacat=0>

[28]https://www.google.gr/search?q=battery+9v+no+name&tbm=isch&tbs=rimg:Cb7zLNSHrwyCljhKnUg1eCmpHReedy8-RwSNDolcZFbI48bgjXGWOe0WbNyfaJ6HJt3ruN3fAwZrXlRoNqEfA-ULCoSCUqdSDV4KakdESLGFZbaOX-iKhIJF553Lz5HBI0RYrHXZfQTSCAqEgkOhqVxkVsjiXGRAiDSW9-EhyoSCRuCNcZagTRZEaoPb7_14LjIUkhIJs3J9onocm3cR8P2BM5XFGzYqEgmu43d8DBmteREjNzGb_1j-

[WNSoSCVGg2oR8D5QsEUfAT3Hq6gMI&tbo=u&sa=X&ved=0ahUKEwjP5J_kkczVAhXRERQKHQAqDyQQ9C8IHw&biw=1366&bih=613&dpr=1#imgrc=vvMs1IevDJwYFM:](https://www.aliexpress.com/wholesale?catId=0&initiative_id=SB_20170725012504&SearchText=BATTERY+AA)

[29]https://www.aliexpress.com/wholesale?catId=0&initiative_id=SB_20170725012504&SearchText=BATTERY+AA

[30]<https://www.ebay.com/sch/i.html?from=R40&trksid=p2047675.m570.l1313.TR0.TRC0.H0.XCR2025+mpateri.TRS0&nkw=CR2025+mpateri&sacat=0>

[31]https://www.aliexpress.com/wholesale?catId=0&initiative_id=SB_20170725012749&SearchText=cable+ARDUINO+PINS

[32]https://www.aliexpress.com/wholesale?catId=0&initiative_id=AS_20170725012750&SearchText=power+bank

Προγράμματα

```
// motor one
int enA = 10;
int in1 = 9;
int in2 = 8;

// motor two
int enB = 5;
int in3 = 7;
int in4 = 6;

void setup() {
  // set all the motor control pins to outputs
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
}

void demoOne()
{
  // turn on motor A
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  // set speed to 200 out of possible range 0~255
  analogWrite(enA, 100);

  // turn on motor B
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  analogWrite(enB, 100); // set speed to 200 out of possible range 0~255
```

```

delay(1000);      //delay
// now turn off motors
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
delay(1000);      //delay

                // turn on motor A
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
analogWrite(enA, 200); // set speed to 200 out of possible range 0~255
delay(510);
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
}

void loop()
{
    demoOne();

}

```

Δεύτερο πρόγραμμα

```

#include <IRremote.h>

int CONTROL_PIN = 3;

// motor one
int enA = 10;
int in1 = 9;

```

```

int in2 = 8;
// motor two
int enB = 5;
int in3 = 7;
int in4 = 6;

IRrecv irrecv(CONTROL_PIN);
decode_results results;

void setup() {
  pinMode(CONTROL_PIN, INPUT); // initialize the digital pin as an output.

  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT); // set all the motor control pins to outputs
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  irrecv.enableIRIn(); // Start the receiver
  Serial.begin(9600);
}

void straight()
{

  digitalWrite(in1, LOW);           //      /\      //
  digitalWrite(in2, HIGH);          //      /|\      //
  digitalWrite(in3, LOW);           //      / | \      //
  digitalWrite(in4, HIGH);          //      |      //

```

```

analogWrite(enA, 200);           //      |      //
analogWrite(enB, 200);           //      |      //
delay(2000);

digitalWrite(in1, LOW);           //  ___  ____  ____ //
digitalWrite(in2, LOW);           // |  |  |  |  //
digitalWrite(in3, LOW);           // |  |--- |--- //
digitalWrite(in4, LOW);           // |___| |  |  //

}

void backOne()
{

digitalWrite(in1, HIGH);           //      |      //
digitalWrite(in2, LOW);           //      |      //
digitalWrite(in3, HIGH);           //      |      //
digitalWrite(in4, LOW);           //      \ | /      //
analogWrite(enA, 200);           //      \| /      //
analogWrite(enB, 200);           //      \| /      //
delay(2000);

digitalWrite(in1, LOW);           //  ___  ____  ____ //
digitalWrite(in2, LOW);           // |  |  |  |  //
digitalWrite(in3, LOW);           // |  |--- |--- //
digitalWrite(in4, LOW);           // |___| |  |  //

}

void rightOne()

```

```

{
    //      >      //
    digitalWrite(in1, LOW);      //      >      //
    digitalWrite(in2, HIGH);      //  >>>>>>>>> //
    analogWrite(enA, 200);      //      >      //
    //      >      //

    delay(510);
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);

}

```

void leftOne()

```

{
    digitalWrite(in3, LOW);      //  <      //
    digitalWrite(in4, HIGH);      //  <      //
    analogWrite(enA, 200);      // <<<<<<<<<< //
    delay(510);      //  <      //
    //  <      //

    delay(510);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
}

```

void rotation()

```

{

    digitalWrite(in1, LOW);      //      //
    digitalWrite(in2, HIGH);      //  <-----  //
    digitalWrite(in3, HIGH);      //  |      /\  //

```

```

digitalWrite(in4, LOW);           //  |      |  //
analogWrite(enA, 200);           //  |      |  //
analogWrite(enB, 200);           //  \|      |  //
delay(520);                       //  ----->  //

digitalWrite(in1, LOW);           //  ____  ____  ____  //
digitalWrite(in2, LOW);           //  |  |  |  |  //
digitalWrite(in3, LOW);           //  |  |  |--- |---  //
digitalWrite(in4, LOW);           //  |____|  |  |  //

}

```

```

void squareOne()
{
  for(int i=0; i<4;i++)
  {
    straight();
    delay(800);
    leftOne();
    delay(500);
  }
}

```

```

void squareTwo()
{
  for(int i=0; i<4;i++)
  {
    straight();
    delay(800);
    rightOne();
    delay(500);
  }
}

```

```

    }
}

void octopus()
{

    digitalWrite(in1, LOW);           //      /\      //
    digitalWrite(in2, HIGH);          //      /|\      //
    digitalWrite(in3, LOW);           //      / |\      //
    digitalWrite(in4, HIGH);          //      |      //
    analogWrite(enA, 200);             //      |      //
    analogWrite(enB, 200);            //      |      //
    delay(1000);
    off();

                                //      >      //

    digitalWrite(in1, LOW);           //              >  //
    digitalWrite(in2, HIGH);          //  >>>>>>>>> //
    analogWrite(enA, 200);             //              >  //
    delay(510);                       //      >      //
    off();

    digitalWrite(in1, LOW);           //      /\      //
    digitalWrite(in2, HIGH);          //      /|\      //
    digitalWrite(in3, LOW);           //      / |\      //
    digitalWrite(in4, HIGH);          //      |      //
    analogWrite(enA, 200);             //      |      //
    analogWrite(enB, 200);            //      |      //
    delay(1000);
    off();

    digitalWrite(in1, HIGH);          //      |      //

```

```

digitalWrite(in2, LOW);           //      |      //
digitalWrite(in3, HIGH);          //      |      //
digitalWrite(in4, LOW);           //      \ | /      //
analogWrite(enA, 200);            //      \| /      //
analogWrite(enB, 200);            //      \|/      //
delay(1000);
off();

digitalWrite(in1, LOW);           //              //
digitalWrite(in2, HIGH);          //  <-----  //
digitalWrite(in3, HIGH);          //  |      /\  //
digitalWrite(in4, LOW);           //  |      |  //
analogWrite(enA, 200);            //  |      |  //
analogWrite(enB, 200);            //  \|/      |  //
delay(520);                       //  ----->  //
off();

```

```

digitalWrite(in1, LOW);           //      /\      //
digitalWrite(in2, HIGH);          //      / |\      //
digitalWrite(in3, LOW);           //      / | \      //
digitalWrite(in4, HIGH);          //      |      //
analogWrite(enA, 200);            //      |      //
analogWrite(enB, 200);            //      |      //
delay(1000);
off();

digitalWrite(in1, HIGH);          //      |      //
digitalWrite(in2, LOW);           //      |      //
digitalWrite(in3, HIGH);          //      |      //
digitalWrite(in4, LOW);           //      \ | /      //
analogWrite(enA, 200);            //      \| /      //

```

```

analogWrite(enB, 200);           //  \//  //
delay(1000);

off();

//      >  //

digitalWrite(in1, LOW);          //      >  //
digitalWrite(in2, HIGH);         //  >>>>>>>>> //
analogWrite(enA, 200);           //      >  //
delay(510);                      //      >  //
off();

digitalWrite(in1, LOW);          //      /\    //
digitalWrite(in2, HIGH);         //      /|\   //
digitalWrite(in3, LOW);          //      /|\   //
digitalWrite(in4, HIGH);         //      |    //
analogWrite(enA, 200);           //      |    //
analogWrite(enB, 200);           //      |    //
delay(1000);

off();

digitalWrite(in1, HIGH);         //      |    //
digitalWrite(in2, LOW);          //      |    //
digitalWrite(in3, HIGH);         //      |    //
digitalWrite(in4, LOW);          //      \ | /   //
analogWrite(enA, 200);           //      \ | /   //
analogWrite(enB, 200);           //      \//    //
delay(2000);

digitalWrite(in1, LOW);          //  ____  ____  ____ //
digitalWrite(in2, LOW);          // |  |  |  |  //
digitalWrite(in3, LOW);          // |  |  |--- |--- //
digitalWrite(in4, LOW);          // |____| |  |  //

```

```
}
```

```
void eight()
```

```
{
```

```
    digitalWrite(in1, LOW);           //      /\      //
```

```
    digitalWrite(in2, HIGH);          //      /|\      //
```

```
    digitalWrite(in3, LOW);           //      / |\      //
```

```
    digitalWrite(in4, HIGH);          //      |      //
```

```
    analogWrite(enA, 200);             //      |      //
```

```
    analogWrite(enB, 200);            //      |      //
```

```
    delay(1000);
```

```
    off();
```

```
                                //      >      //
```

```
    digitalWrite(in1, LOW);           //      >      //
```

```
    digitalWrite(in2, HIGH);          // >>>>>>>>> //
```

```
    analogWrite(enA, 200);             //      >      //
```

```
    delay(510);                       //      >      //
```

```
    off();
```

```
    digitalWrite(in1, LOW);           //      /\      //
```

```
    digitalWrite(in2, HIGH);          //      /|\      //
```

```
    digitalWrite(in3, LOW);           //      / |\      //
```

```
    digitalWrite(in4, HIGH);          //      |      //
```

```
    analogWrite(enA, 200);             //      |      //
```

```
    analogWrite(enB, 200);            //      |      //
```

```
    delay(1000);
```

```
    off();
```

```
    digitalWrite(in3, LOW);           //      <      //
```

```

digitalWrite(in4, HIGH);           //  <           //
analogWrite(enA, 200);             //  <<<<<<<<<<  //
delay(510);                        //  <           //
                                   //  <           //

off();

digitalWrite(in1, LOW);            //      /\      //
digitalWrite(in2, HIGH);           //      /|\     //
digitalWrite(in3, LOW);            //      / |\    //
digitalWrite(in4, HIGH);           //      |      //
analogWrite(enA, 200);             //      |      //
analogWrite(enB, 200);            //      |      //
delay(1000);

off();

digitalWrite(in3, LOW);            //  <           //
digitalWrite(in4, HIGH);           //  <           //
analogWrite(enA, 200);             //  <<<<<<<<<<  //
delay(510);                        //  <           //
                                   //  <           //

off();

digitalWrite(in1, LOW);            //      /\      //
digitalWrite(in2, HIGH);           //      /|\     //
digitalWrite(in3, LOW);            //      / |\    //
digitalWrite(in4, HIGH);           //      |      //
analogWrite(enA, 200);             //      |      //
analogWrite(enB, 200);            //      |      //
delay(1000);

off();

digitalWrite(in3, LOW);            //  <           //
digitalWrite(in4, HIGH);           //  <           //
analogWrite(enA, 200);            //  <<<<<<<<<<  //

```

```

delay(510);           //  <           //

//  <           //

off();

digitalWrite(in1, LOW);           //      /\      //
digitalWrite(in2, HIGH);          //      /|\      //
digitalWrite(in3, LOW);           //      / |\      //
digitalWrite(in4, HIGH);          //      |      //
analogWrite(enA, 200);            //      |      //
analogWrite(enB, 200);            //      |      //
delay(1000);
off();
digitalWrite(in3, LOW);           //  <           //
digitalWrite(in4, HIGH);          //  <           //
analogWrite(enA, 200);            // <<<<<<<<< //
delay(510);           //  <           //

//  <           //

off();

digitalWrite(in1, LOW);           //      /\      //
digitalWrite(in2, HIGH);          //      /|\      //
digitalWrite(in3, LOW);           //      / |\      //
digitalWrite(in4, HIGH);          //      |      //
analogWrite(enA, 200);            //      |      //
analogWrite(enB, 200);            //      |      //
delay(1000);
off();

//      >      //

digitalWrite(in1, LOW);           //      >      //
digitalWrite(in2, HIGH);          //  >>>>>>>> //
analogWrite(enA, 200);            //      >      //

```

```

delay(510);           //      >    //
off();
digitalWrite(in1, LOW);      //      /\      //
digitalWrite(in2, HIGH);     //      /|\      //
digitalWrite(in3, LOW);      //      / |\      //
digitalWrite(in4, HIGH);     //      |      //
analogWrite(enA, 200);       //      |      //
analogWrite(enB, 200);       //      |      //
delay(1000);
off();

//      >    //

digitalWrite(in1, LOW);      //      >    //
digitalWrite(in2, HIGH);     // >>>>>>>>> //
analogWrite(enA, 200);       //      >    //
delay(510);           //      >    //
off();
digitalWrite(in1, LOW);      //      /\      //
digitalWrite(in2, HIGH);     //      /|\      //
digitalWrite(in3, LOW);      //      / |\      //
digitalWrite(in4, HIGH);     //      |      //
analogWrite(enA, 200);       //      |      //
analogWrite(enB, 200);       //      |      //
delay(1000);
off();

//      >    //

digitalWrite(in1, LOW);      //      >    //
digitalWrite(in2, HIGH);     // >>>>>>>>> //
analogWrite(enA, 200);       //      >    //
delay(510);           //      >    //

```

```

digitalWrite(in1, LOW);           //  _  _  _  _  //
digitalWrite(in2, LOW);           // |  |  |  //
digitalWrite(in3, LOW);           // |  |--- |--- //
digitalWrite(in4, LOW);           // |__| |  |  //

}

void functioneight()
{
    straight();
    off();
    rightOne();
    off();
    straight();
    off();
    leftOne();
    off();
    straight();
    off();
    leftOne();
    off();
    straight();
    off();
    leftOne();
    off();
    straight();
    off();
    leftOne();
    off();
    straight();
    off();
}

```

```

rightOne();
off();
straight();
off();
rightOne();
off();
straight();
off();
rightOne();
}

```

```

void off()
{
    digitalWrite(in1, LOW);           //  _  _  _  //
    digitalWrite(in2, LOW);           // |  |  |  //
    digitalWrite(in3, LOW);           // |  |--- |--- //
    digitalWrite(in4, LOW);           // |__| |  |  //
    delay(50);
}

```

```

void house()
{
    digitalWrite(in1, LOW);           //      /\      //
    digitalWrite(in2, HIGH);          //      /|\      //
    digitalWrite(in3, LOW);           //      / | \      //
    digitalWrite(in4, HIGH);          //      |      //
    analogWrite(enA, 200);             //      |      //
    analogWrite(enB, 200);            //      |      //
    delay(1000);
    off();
}

```

```

//      >      //
digitalWrite(in1, LOW);      //      >      //
digitalWrite(in2, HIGH);      //  >>>>>>>>> //
analogWrite(enA, 200);      //      >      //
delay(360);      //      >      //
off();

digitalWrite(in1, LOW);      //      /\      //
digitalWrite(in2, HIGH);      //      /|\      //
digitalWrite(in3, LOW);      //      / |\      //
digitalWrite(in4, HIGH);      //      |      //
analogWrite(enA, 200);      //      |      //
analogWrite(enB, 200);      //      |      //
delay(510);
off();

//      >      //
digitalWrite(in1, LOW);      //      >      //
digitalWrite(in2, HIGH);      //  >>>>>>>>> //
analogWrite(enA, 200);      //      >      //
delay(510);      //      >      //
off();

digitalWrite(in1, LOW);      //      /\      //
digitalWrite(in2, HIGH);      //      /|\      //
digitalWrite(in3, LOW);      //      / |\      //
digitalWrite(in4, HIGH);      //      |      //
analogWrite(enA, 200);      //      |      //
analogWrite(enB, 200);      //      |      //
delay(1000);
off();      //      >      //
digitalWrite(in1, LOW);      //      >      //
digitalWrite(in2, HIGH);      //  >>>>>>>>> //

```

```

analogWrite(enA, 200);           //      >  //
delay(360);                      //      >  //
off();
digitalWrite(in1, LOW);          //      /\      //
digitalWrite(in2, HIGH);         //      /|\      //
digitalWrite(in3, LOW);          //      / |\      //
digitalWrite(in4, HIGH);         //      |      //
analogWrite(enA, 200);           //      |      //
analogWrite(enB, 200);           //      |      //
delay(1000);
off();
digitalWrite(in1, LOW);          //      <      //
digitalWrite(in2, HIGH);         //      <      //
analogWrite(enA, 200);           // <<<<<<<<<  //
delay(360);                      //      <      //
//      <      //

off();
digitalWrite(in1, LOW);          //      /\      //
digitalWrite(in2, HIGH);         //      /|\      //
digitalWrite(in3, LOW);          //      / |\      //
digitalWrite(in4, HIGH);         //      |      //
analogWrite(enA, 200);           //      |      //
analogWrite(enB, 200);           //      |      //
delay(1000);
off();
digitalWrite(in1, LOW);          //      <      //
digitalWrite(in2, HIGH);         //      <      //
analogWrite(enA, 200);           // <<<<<<<<<  //
delay(360);                      //      <      //
//      <      //

```

```

off();

digitalWrite(in1, LOW);           //      /\      //
digitalWrite(in2, HIGH);          //      /|\      //
digitalWrite(in3, LOW);           //      / |\      //
digitalWrite(in4, HIGH);          //      |      //
analogWrite(enA, 200);             //      |      //
analogWrite(enB, 200);            //      |      //
delay(510);

off();

digitalWrite(in1, LOW);           //      <      //
digitalWrite(in2, HIGH);          //      <      //
analogWrite(enA, 200);            // <<<<<<<<<< //
delay(360);                       //      <      //
                                   //      <      //

off();

digitalWrite(in1, LOW);           //      /\      //
digitalWrite(in2, HIGH);          //      /|\      //
digitalWrite(in3, LOW);           //      / |\      //
digitalWrite(in4, HIGH);          //      |      //
analogWrite(enA, 200);             //      |      //
analogWrite(enB, 200);            //      |      //
delay(1000);

off();

digitalWrite(in1, LOW);           //      <      //
digitalWrite(in2, HIGH);          //      <      //
analogWrite(enA, 200);            // <<<<<<<<<< //
delay(360);                       //      <      //
                                   //      <      //

off();

digitalWrite(in1, LOW);           //      /\      //

```

```

digitalWrite(in2, HIGH);           //      /|\      //
digitalWrite(in3, LOW);            //      / |\      //
digitalWrite(in4, HIGH);           //      |      //
analogWrite(enA, 200);              //      |      //
analogWrite(enB, 200);             //      |      //
delay(1000);

digitalWrite(in1, LOW);            //  ____  ____  ____  //
digitalWrite(in2, LOW);            // |  ||   |   //
digitalWrite(in3, LOW);            // |  ||--- |--- //
digitalWrite(in4, LOW);            // |____| |   |   //

}

void downup()
{
  digitalWrite(in1, HIGH);          //      |      //
  digitalWrite(in2, LOW);           //      |      //
  digitalWrite(in3, HIGH);          //      |      //
  digitalWrite(in4, LOW);           //      \ | /      //
  delay(2000);                      //      \|/      //
                                   //      \//      //

  digitalWrite(in1, LOW);           //      /\      //
  digitalWrite(in2, HIGH);          //      /|\      //
  digitalWrite(in3, LOW);           //      / |\      //
  digitalWrite(in4, HIGH);          //      |      //
  analogWrite(enA, 200);             //      |      //
  analogWrite(enB, 200);            //      |      //
  delay(2000);

  digitalWrite(in1, LOW);            //  ____  ____  ____  //

```

```

digitalWrite(in2, LOW);           // | | | //
digitalWrite(in3, LOW);           // | |--- |--- //
digitalWrite(in4, LOW);           // |___| | | //
}

```

```

void updown()
{

```

```

digitalWrite(in1, LOW);           //      /\      //
digitalWrite(in2, HIGH);          //      / | \      //
digitalWrite(in3, LOW);           //      / | \      //
digitalWrite(in4, HIGH);          //      |      //
analogWrite(enA, 200);            //      |      //
analogWrite(enB, 200);            //      |      //
delay(2000);

```

```

digitalWrite(in1, HIGH);          //      |      //
digitalWrite(in2, LOW);           //      |      //
digitalWrite(in3, HIGH);          //      |      //
digitalWrite(in4, LOW);           //      \ | /      //
delay(2000);                      //      \ | /      //
//      \//      //

```

```

digitalWrite(in1, LOW);           //  ____  ____  ____ //
digitalWrite(in2, LOW);           // | | | | //
digitalWrite(in3, LOW);           // | |--- |--- //
digitalWrite(in4, LOW);           // |___| | | //

}

```

```

void loop() {

    if (irrecv.decode(&results)) {

        buttons();

        irrecv.resume();    // Receive the next value

    }
}

void buttons() // takes action based on IR code received describing Car MP3 IR codes
{

    switch(results.value){
    case 0xFF629D:                //koumbi mprosta
        straight();
        Serial.println("pige mbrosta ");

        break;
    case 0xFF22DD:                // aristero koumbi

        leftOne();
        Serial.println("pige aristera ");
        break;

    case 0xFF02FD:                // koumbi ok
        rotation();
        Serial.println("ekane anastrofi ");
    }
}

```

```

break;

case 0xFFC23D:                                //deksi koumbi
    rightOne();
    Serial.println("estripse deksia ");
    break;

case 0xFFA857:                                // kato koumbi
    backOne();
    Serial.println("pige piso ");
    break;

case 0xFF6897:                                // koumpaki 1
    squareTwo();
    Serial.println(" tetragono me deksia peristrofi ");
    break;

case 0xFF9867:                                // koumbi 2
    updown();
    Serial.println(" pano kato ");
    break;

case 0xFFB04F:                                // koumbaki 3
    squareOne();
    Serial.println(" tetragono me aristeri peristrofi ");
    break;

case 0xFF30CF:                                //koumbaki 4
    updown();
    Serial.println(" pano kato");

```

```

break;

case 0xFF18E7:                                     // koumbaki 5
    octapus();
    Serial.println(" sxediash stayroy ");
    break;

case 0xFF7A85:                                     //koumbaki 6
    downup();
    Serial.println(" pano kato ");
    break;

case 0xFF10EF:                                     //koumbaki 7
    house();
    Serial.println(" sxediazei spiti ");
    break;

case 0xFF38C7:                                     //koumbaki 8
    eight();
    Serial.println(" sxediash okto ");
    break;

case 0xFF5AA5:                                     // koumbaki 9
    functioneight();
    Serial.println(" kanei to 8 me sinartisis ");
    break;
default:
    Serial.println();
}

```

}