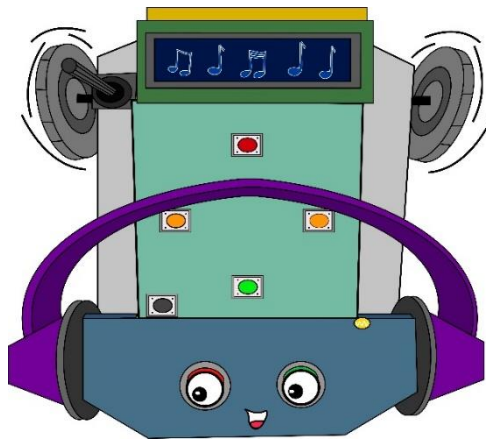




**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**  
**ΤΜΗΜΑ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ**

ΘΕΜΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

**Κατασκευή Εκπαιδευτικού Ρομπότ Δαπέδου για Παιδιά Προσχολικής  
Ηλικίας**  
**Construction of an educational Two-wheel robot for preschool children**



**ΟΝΟΜΑΤΑ ΦΟΙΤΗΤΩΝ:**

ΜΑΥΡΟΒΟΥΝΙΩΤΗ ΒΑΣΙΛΙΚΗ  
ΚΟΤΑΝΙΔΗΣ ΒΑΣΙΛΕΙΟΣ

**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:**

Δρ. ΜΙΧΑΛΗΣ ΠΑΠΟΥΤΣΙΔΑΚΗΣ

ΑΙΓΑΛΕΩ, ΙΟΥΝΙΟΣ 2018

## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο / Η κάτωθι υπογεγραμμένος / η Μαυροβασιλική Βασιλικά,  
του Λευκίδα, με αριθμό μητρώου 49799 φοιτητής / τρια του  
Τμήματος Βιομηχανικής Σχεδίασης και Παραγωγής του Πανεπιστημίου Δυτικής Αττικής  
πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα  
παρακάτω:

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

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

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

Ο Δηλών



Ημερομηνία

23/06/2018

## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο / Η κάτωθι υπογεγραμμένος / η ..... Βαβίλιος Κατανίδης .....

του Θεσσαλίας ....., με αριθμό μητρώου 430240 φοιτητής / τριά του **Τμήματος Βιομηχανικής Σχεδίασης και Παραγωγής του Πανεπιστημίου Δυτικής Αττικής** πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

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

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

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

Ο Δηλών



Ημερομηνία

23-6-2018

## Περίληψη

Η παρούσα πτυχιακή εργασία έχει σαν στόχο την σχεδίαση και την κατασκευή ενός παιχνιδιού για παιδιά προσχολικής ηλικίας, τύπου ρομπότ δαπέδου. Τα παιδιά θα έχουν την δυνατότητα να δίνουν εντολές κίνησης ,στο ρομπότ, μέσω μίας οθόνης και πλήκτρων και να αλληλοεπιδρούν μαζί του μέσω αισθητήριων ήχου και φωτός. Στόχος της κατασκευής είναι η εισαγωγή παιδιών μικρής ηλικίας στις βασικές ιδέες της μηχανικής ,χωρίς να χρειάζονται γνώσεις ανάγνωσης, μέσα από ένα σύνολο εικόνων , φωτεινών και ηχητικών ενδείξεων.

Στα παρακάτω κεφάλαια αναπτύσσεται αρχικά το θεωρητικό υπόβαθρο .Αναφέρεται τι είναι η Μηχανική και στη συνέχεια αναλύονται τα δομικά της στοιχεία. αλλά και η θέση της στην εκπαίδευση. Αναπτύσσοντας την έννοια της μηχανικής αλλά και τα θετικά αποτελέσματα που έχει η χρήση της στην εκπαίδευση, δίνεται έμφαση στο στόχο της εργασίας που αναλύεται στο τέλος του πρώτου κεφαλαίου.

Στο δεύτερο κεφάλαιο αναλύεται το Κατασκευαστικό Μέρος του ρομπότ. Αρχικά παραθέτονται τα εξαρτήματα που χρησιμοποιήθηκαν μαζί με μία περιγραφή. Στη συνέχεια αναλύεται το μηχανολογικό κομμάτι ,δηλαδή η διαδικασία που ακολουθήθηκε για να ενωθούν όλα τα εξαρτήματα σε μία κοινή κατασκευή. Στο τέλος του κεφαλαίου παρουσιάζεται το ηλεκτρονικό κομμάτι ,δηλαδή με ποιο τρόπο ενώθηκαν τα εξαρτήματα μεταξύ τους έτσι ώστε να επιτευχθεί το ζητούμενο της κατασκευής.

Στο τρίτο κεφάλαιο αναλύεται το λογισμικό της κατασκευής. Αρχικά προσδιορίζεται το σενάριο που τρέχει ο κώδικας .Στη περίληψη του κώδικα περιγράφονται τα 4 προγράμματα του ρομπότ ,αλλά και πως γίνεται η πλοήγηση σε αυτά, δηλαδή η χρήση των πλήκτρων. Στη συνέχεια παρουσιάζεται όλη η ροή του προγράμματος με τη μορφή διαγράμματος ροής . Στα διαγράμματα ροής δεν αναλύεται μόνη η βασική ρουτίνα αλλά και όλες η υπορουτίνες που δημιουργήθηκαν. Τέλος δίνεται ο ίδιος ο κώδικας με σχολιασμούς και επεξήγηση.

Στο τελευταίο κεφάλαιο εξηγείται η προσέγγιση που ακολουθήθηκε για τη δημιουργία της κατασκευής αλλά και για την ανάπτυξη του κώδικα. Επίσης παρουσιάζονται τα προβλήματα που προέκυψαν και πως αντιμετωπίστηκαν.

## **Abstract**

The aim of this project lies in the construction of an educational two-wheel robot for preschool children. This construction will allow children to give commands to said robot via a screen and keyboards, as well as to interact with it through sound and light sensors. The objective of this construction is to introduce children to the basic ideas of mechatronics without reading skills as a prerequisite, but instead through images, light and auditory indicators.

In the following chapters, the theoretical framework will be presented first, by analysing the nature of mechatronics and its dimensions, as well as its role in education. The theoretical framework will be concluded with the development of mechatronics and the positive consequences of its use in education, which is the overarching goal of this study.

In the second chapter, the construction of the robot is analyzed. First the components used are cited and described and the engineering process that was followed so as to merge all the components in the final construction is explained. The chapter concludes with the description of the electronic aspect of the construction, explaining the way the components were merged together to build the final product.

In the third chapter, the software used for the construction is presented. The scenario with which the code runs is specified, along with a description of the code, the four programs of the robot and how to navigate through the programs through the use of keys. Next, the whole program flow is presented via flow charts, analysing the basic routine and subroutines created. At the end of the third chapter, the actual code used is presented, along with comments and clarifications.

This study concludes by explaining the approach that was followed during the construction of the hardware and the software. In the end there are mentioned the problems that was faced during this thesis and how they were dealt with.

## Ευχαριστήριο Σημείωμα

Θα θέλαμε να εκφράσουμε τις ευχαριστίες μας στον υπεύθυνο και επιβλέπων καθηγητή της παρούσας πτυχιακής εργασίας, Δρ.Μιχάλη Παπουτσιδάκη, για την ανάθεση του θέματος καθώς και για το χρόνο που μας διέθεσε μέχρι την ολοκλήρωση του.

Ιδιαίτερες ευχαριστίες θα θέλαμε να εκφράσουμε στον κ.Χατζόπουλο για την οδηγίες και τις συμβουλές που μας προσέφερε, καθώς και για την προθυμία που είχε να μας βοηθήσει κάθε φορά που αντιμετωπίζαμε κάποιο πρόβλημα.

Τέλος θα θέλαμε να ευχαριστήσουμε τους φίλους, τους συμφοιτητές και κυρίως τις οικογένειές μας που καθ' όλη την εκπόνηση της πτυχιακής μας εργασίας μας προέτρεπαν και μας βοηθούσαν με υπομονή.

## Περιεχόμενα

<b>1</b>	<b>Θεωρητικό Υπόβαθρο.....</b>	<b>8</b>
1.1	Τεχνολογία - Μηχατρονική .....	8
1.2	Η Τεχνολογία στην εκπαίδευση .....	9
1.3	Σκοπός.....	9
<b>2</b>	<b>Ανάλυση Κατασκευαστικού Μέρους-Hardware.....</b>	<b>10</b>
2.1	Εισαγωγή .....	10
2.2	Εξαρτήματα Κατασκευής.....	10
2.3	Μηχανολογικό Μέρος .....	17
2.4	Ηλεκτρονικό Μέρος .....	21
<b>3</b>	<b>Ανάλυση Λογισμικού – Software .....</b>	<b>24</b>
3.1	Εισαγωγή .....	24
3.2	Ο κώδικας περιληπτικά.....	24
3.3	Διαγράμματα Ροής .....	26
3.4	Κώδικας Προγράμματος .....	40
3.5	Σχόλια - Μεθοδολογία Έρευνας.....	72
<b>4</b>	<b>Βιβλιογραφία .....</b>	<b>73</b>

# 1 Θεωρητικό Υπόβαθρο

## 1.1 Τεχνολογία - Μηχατρονική

Η Τεχνολογία ορίζεται ως η πρακτική εφαρμογή των ιδεών που προκύπτουν από τις θετικές επιστήμες καθώς και το σύνολο των γνώσεων που σχετίζονται με την εφαρμογή αυτή. Προέκυψε, κυρίως, από την ανάγκη του ανθρώπου να προσαρμόζεται στο περιβάλλον και να το ελέγχει, καθώς όμως εξελίσσετε άρχισε να εφαρμόζεται και σε άλλους κλάδους. Πλέον η τεχνολογία δεν χρησιμοποιείται μόνο για την ικανοποίηση των βιοτικών αναγκών του ανθρώπου αλλά και για την επικοινωνία, την ψυχαγωγία και την εκπαίδευση του. [31]

Η τεχνολογία πλέον αλλάζει ταχύτατα. Η καινοτομία της μιας ημέρας μπορεί την επομένη να απαξιωθεί. Με τον ίδιο ρυθμό που αυξάνονται οι τεχνολογικές γνώσεις αυξάνονται και οι κλάδοι της τεχνολογίας. Ένας κλάδος που έχει ιδιαίτερη διάκριση τα τελευταία χρόνια είναι η Μηχατρονική. Η λέξη Μηχατρονική (mechatronics) εμφανίστηκε αρχικά τη δεκαετία του 1970 στην Ιαπωνία. Μηχατρονική είναι η σύνθεση της πληροφορικής με τη μηχανολογία και την ηλεκτρολογία. Αποτέλεσμα της μηχατρονικής είναι μικτά συστήματα με ηλεκτρικά, ηλεκτρονικά και μηχανολογικά μέρη που τα χειρίζεται ένα σύστημα ελέγχου, αυτόματος έλεγχος.

Τα δομικά στοιχεία αυτής της τεχνολογίας πέντε. Πρώτο στοιχείο είναι η μοντελοποίηση φυσικών συστημάτων, κατά αυτή τη διαδικασία μελετάτε το σύστημα πρέπει να ελεγχθεί και στη συνέχεια δημιουργούνται «αντικείμενα» τα οποία μπορούν να αντιγράψουν τη συμπεριφορά του. Με τη διαδικασία της μοντελοποίησης μπορούμε να εξάγουμε συμπεράσματα για το μελετώμενο σύστημα πιο εύκολα και πιο γρήγορα. [20]

Δεύτερο στοιχείο της μηχατρονικής είναι η εφαρμογή της θεωρίας των σημάτων και συστημάτων. Τα σήματα μεταφέρουν πληροφορία και εντολές, ενώ τα συστήματα παράγουν ή επεξεργάζονται τη πληροφορία και εκτελούν τις εντολές.

Τρίτο και εξίσου σημαντικό στοιχείο της μηχατρονικής είναι και η εφαρμογή της τεχνολογίας των Ηλεκτρονικών Υπολογιστών καθώς και των λογικών συστημάτων. Σε αυτό το κομμάτι είναι που ανήκουν και οι Μικροεπεξεργαστές. Οι μικροεπεξεργαστές, ή αλλιώς μικροκοντρόλερς, είναι ο εγκέφαλος του συστήματος, συλλέγουν τα σήματα, τα επεξεργάζονται και αποφασίζουν ποιες θα είναι οι εντολές.

Το τέταρτο στοιχείο είναι το λογισμικό. Λογισμικό (software) είναι το σύνολο των προγραμμάτων και των διαδικασιών που εκτελούνται σε έναν υπολογιστή, μικροεπεξεργαστή. Το λογισμικό είναι η «λογική» που χρησιμοποιεί ο «εγκέφαλος» για να εξάγει συμπεράσματα.

Πέμπτο στοιχείο είναι η Αισθητήρες και οι Ενεργοποιητές. Οι αισθητήρες είναι εξαρτήματα τα οποία μπορούν αντιληφθούν της συνθήκες του περιβάλλοντος. Ένας αισθητήρας μπορεί να «διαβάσει» ένα



φυσικό μέγεθος και στη συνέχεια να το μετατρέψει σε ένα μετρήσιμο μέγεθος, όπως η ηλεκτρική τάση, σε κάτι δηλαδή που να μπορεί να το αντιληφθεί ο μικροεπεξεργαστής. Αντίθετα οι ενεργοποιητές παίρνουν μία εντολή από τον μικροεπεξεργαστή και ενεργοποιούν το μηχανολογικό κομμάτι ,ώστε να εκτελεστεί η εντολή.[21-22]

## **1.2 Η Τεχνολογία στην εκπαίδευση**

Η εκπαίδευση είναι ένας κοινωνικός θεσμός που σαν στόχο έχει τη μετάδοση γνώσεων, προκειμένου το άτομο να μπορεί να είναι αυτόνομο αλλά και να ενταχθεί ομαλά στη κοινωνία. Έχοντας υπόψιν την θέση που κατέχει η τεχνολογία στη σημερινή κοινωνία, είναι φυσικό ενταχθεί και στο κομμάτι της εκπαίδευσης. Η χρήση προϊόντων της σύγχρονης τεχνολογίας στην εκπαιδευτική διαδικασία βοηθάει τα παιδιά να ανταποκριθούν στις σύγχρονες απαιτήσεις. Με τη χρήση της τεχνολογίας η μαθησιακή διαδικασία επιταχύνεται ,υπάρχει καλύτερη εμπέδωση ενώ ταυτόχρονα γίνεται πιο ευχάριστη και δημιουργική. Επιπλέον δίνεται η ευκαιρία και το κίνητρο, σε παιδιά που δυσκολεύονται να λειτουργήσουν με την συμβατική διαδικασία μάθησης, να κατανοήσουν το αντικείμενο του μαθήματος. Η χρήση της Μηχατρονικής ,ειδικότερα , στην εκπαίδευση βοηθάει στην εμπέδωση της γνώσης αφού ο μαθητής θα παρακολουθεί εικονικά πειράματα και εκπαιδευτικά λογισμικά. Η Μηχατρονική, επίσης, δίνει στο παιδί στο παιδί τη δυνατότητα, να έρθει ταυτόχρονα σε επαφή με πολλούς κλάδους της τεχνολογίας και ίσως να ανακαλύψει την κλίση του σε κάποιον από αυτούς.

## **1.3 Σκοπός**

Βασικός σκοπός αυτής της πτυχιακής εργασίας είναι η κατασκευή ενός ρομπότ δαπέδου. Το ρομπότ αυτό έχει διπλό ρόλο ,εκπαιδευτικό και ψυχαγωγικό, αφού είναι ένα παιχνίδι για παιδιά προσχολικής ηλικίας με στόχο την εισαγωγή τους σε έννοιες όπως ο προγραμματισμός , ο αυτοματισμός και τα αισθητήρια. Καθώς το παιδί αλληλοεπιδρά με το ρομπότ θα δει πως οι εντολές που δίνει στο ρομπότ και οι συνθήκες στο περιβάλλον του συνδυάζονται και προκύπτει ένα αποτέλεσμα. Η δυνατότητα αυτή δίνει στο παιδί μία αίσθηση ασφάλειας σε ότι αφορά τη τεχνολογία , επιπλέον το βοηθάει να αναπτύξει τη λογική του και τη συνδυαστική του σκέψη. Καθώς η ηλικία στην οποία απευθύνεται η κατασκευή είναι προσχολική και δεν υπάρχει η ικανότητα της ανάγνωσης, θα χρησιμοποιηθούν ένα σύνολο από ηχητικές και φωτεινές ενδείξεις όπως και σύμβολα για την επικοινωνία παιδιού και ρομπότ. Το ρομπότ δεν κατασκευάστηκε για να αντικαταστήσει τον εκπαιδευτικό αλλά για να του προσφέρει ένα νέο εργαλείο.

## 2 Ανάλυση Κατασκευαστικού Μέρους-Hardware

### 2.1 Εισαγωγή

Στο κεφάλαιο αυτό παραθέτονται τα υλικά που χρησιμοποιήθηκαν για τη υλοποίηση της κατασκευής, ηλεκτρικά και ηλεκτρονικά , καθώς και η διαδικασία που ακολουθήθηκε για την συναρμολόγηση . Αφού αποφασίστηκε ο σκοπός της κατασκευής, επιλέχθηκαν τα ανάλογα εξαρτήματα, τα οποία έπρεπε να είναι συμβατά μεταξύ τους στο κατασκευαστικό μέρος αλλά και σε ότι αφορά το λογισμικό, το οποίο θα αναλυθεί στο επόμενο κεφάλαιο.

### 2.2 Εξαρτήματα Κατασκευής

Για την υλοποίηση της κατασκευής χρησιμοποιήθηκαν τα εξής υλικά μία ρομποτική πλατφόρμα (Shadow Chassis) 1 Arduino Mega, 1 LCD(16X2),4 button με ενσωματωμένα led, 2 round Button, 1 push button, 1 sound sensor, 1 LDR, 1 speaker, 2 κινητήρες τύπου DC, 1 driver (L293D), 1 ποτενσιόμετρο, 12 αντιστάσεις, 2 ηλεκτρονικές πλακέτες, 2 breadboard μικρού μεγέθους, 1 μπιλια κίνησης,1 μπαταρία 12 volt και από ένα διακόπτη 2 καταστάσεων.

Πέρα από τα βασικά εξαρτήματα χρησιμοποιήθηκαν για την υλοποίηση της κατασκευής εργαλείο Dremel, ηλεκτρονικό κολλητήρι, πιστόλι θερμαινόμενης κόλλας ,κρυσταλλική σιλικόνη, πολύμετρο, βίδες, αποστάτες, θερμοσυστελλόμενα μονωτικά καλωδίων, μονωτική ταινία, δεματικά καλωδίων και διάφορα ηλεκτρονικά εργαλεία χειρός.

#### 1) Ρομποτική πλατφόρμα (Shadow Chassis)

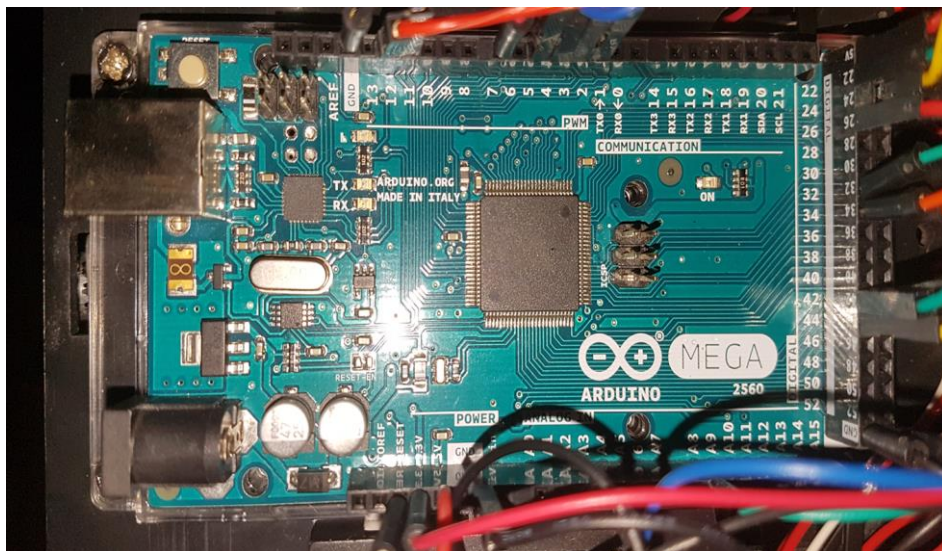
Το Shadow Chassis είναι μία ανθεκτική πλαστική ρομποτική πλατφόρμα η οποία διαθέτει εσοχές και αποσπόμενα στηρίγματα, καθιστώντας την ιδανική βάση για τη συναρμολόγηση μίας κατασκευής[1]



2.2-1 Ρομποτική πλατφόρμα (Shadow Chassis)

## 2) Arduino Mega.

Το Arduino είναι μια ηλεκτρονική πλακέτα ανοικτού κώδικα η οποία μπορεί να προγραμματιστεί με μια ψευδογλώσσα η οποία χρησιμοποιεί στοιχεία της γλώσσας C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++. Το Arduino έχει ενσωματωμένα πάνω του έναν μικροελεγκτή και υποδοχές. Οι υποδοχές είναι αναλογικές ή ψηφιακές ,και ανάλογα τη ρύθμιση η κάθε μία λειτουργεί ως είσοδος ή έξοδος. Το Arduino ,ως μικροεπεξεργαστής, έχει τη δυνατότητα να συνδυάσει το software με το hardware, συλλέγει δεδομένα με τις εισόδους , και ανάλογα με το λογισμικό του ρυθμίζει τις εξόδους, για αυτό και είναι ιδανικό για εφαρμογές σε αυτόματα συστήματα. Ο λόγος για τον οποίο επιλέχθηκε στη συγκεκριμένη κατασκευή το Arduino Mega και όχι το Arduino Uno ,που είναι η βασική εκδοχή της συγκεκριμένης πλατφόρμας, είναι ότι το Arduino Mega , σε σύγκριση με το Arduino Uno, έχει περισσότερες υποδοχές ,αναλογικές και ψηφιακές, συνεπώς μπορεί να χειριστεί περισσότερα εξαρτήματα. [2]

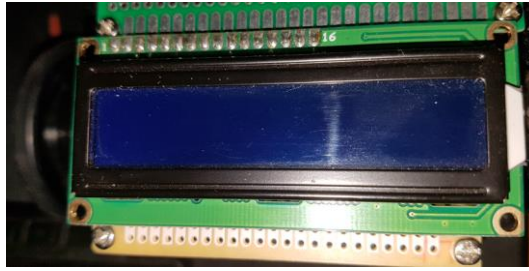


2.2-2 Arduino Mega

## 3) LCD (16x2)

Η οθόνη LCD (liquid crystal display ) είναι ένας τύπος οθόνης η οποίας βασίζεται στην τεχνολογία υγρών κρυστάλλων. Οι διαστάσεις 16x2 σημαίνουν ότι διαθέτει 2 γραμμές όπου η κάθε μία χωράει 16 χαρακτήρες. Διαθέτει 11 ακροδέκτες εισόδου και εξόδου. Όταν υλοποιηθεί η συνδεσμολογία που δίνεται στο φύλλο δεδομένων(datasheet)[16] του κατασκευαστή εμφανίζει τις εντολές που στέλνει ο μικροεπεξεργαστής. Η φωτεινότητα

ρυθμίζεται με τη χρήση ενός ποτενσιόμετρου που έχει συνδεθεί μαζί της, σύμφωνα πάλι με το φύλλο δεδομένων.[3]



2.2-3 LCD (16x2)

#### 4) DC κινητήρες

Ένας Dc κινητήρας είναι ένας από τους τύπους περιστροφικών ηλεκτρικών μηχανών που μετατρέπει την ηλεκτρική ενέργεια συνεχούς ρεύματος σε μηχανική ενέργεια. Η ηλεκτρική ενέργεια που απαιτούν είναι σημαντικά μεγαλύτερη από τα υπόλοιπα υλικά της κατασκευής, και ανάλογή της είναι η ισχύ τους.[4]



2.2-4 DC κινητήρες

#### 5) Button

Ένα button είναι ένας απλός διακόπτης 2 καταστάσεων το οποίο λειτουργεί με ένα πάτημα και επανέρχεται στην αρχική του κατάσταση. Στην εφαρμογή χρησιμοποιήθηκαν απλά button καθώς και button με ενσωματωμένα πάνω τους led.[5]



2.2-5 Button με ενσωματωμένο led [25]



2.2-6 Button [26]

## 6) Δίοδος Εκπομπής Φωτός - Led

Δίοδος Εκπομπής Φωτός, (LED, Light Emitting Diode), αποκαλείται ένας ημιαγωγός ο οποίος εκπέμπει φωτεινή ακτινοβολία όταν του παρέχεται μία ηλεκτρική τάση κατά τη φορά ορθής πόλωσης. Στη κατασκευή τα led που χρησιμοποιήθηκαν είναι ενσωματωμένα σε button. Η συνδεσμολογία που υλοποιήθηκε ωστόσο είναι ίδια με αυτή που θα υλοποιούταν σε ένα αυτόνομο led.[6]



2.2-6 Δίοδος Εκπομπής Φωτός – Led [27]

## 7) Φωτοαντιστάτης LDR

Ένας φωτοαντιστάτης ( LDR) είναι μια μεταβλητή αντίσταση ελεγχόμενη από το φως. Η αντίσταση ενός LDR μειώνεται με την αύξηση της έντασης του προσπίπτοντος φωτός, είναι δηλαδή φωτοαγώγιμο.[7]



2.2-7 Φωτοαντιστάτης LDR [28]

## 8) Αισθητήριο Ήχου(MAX4466)

Το αισθητήριο ήχου είναι ένα εξάρτημα το οποίο διαβάζει τη συχνότητα ήχου και τη μετατρέπει σε τάση. Στη συγκεκριμένη εφαρμογή χρησιμοποιήθηκε ώστε να αντιλαμβάνεται τους έντονους ήχους[8]



2.2-8 Αισθητήριο Ήχου[29]

## 9) Driver L293D

Το L293D είναι ένα ολοκληρωμένο, που χρησιμοποιείται σαν οδηγός κινητήρα και επιτρέπει στον κινητήρα DC να κινείται προς οποιαδήποτε κατεύθυνση. Το L293D είναι ένα ψηφιακό κύκλωμα 16 ακίδων, το οποίο μπορεί ταυτόχρονα να ελέγχει δύο κινητήρες συνεχούς ρεύματος σε οποιαδήποτε κατεύθυνση.[9]



2.2-9 Driver L293D

## 10) Ποτενσιόμετρο

Ένα ποτενσιόμετρο είναι ένας τριφασικός αντιστάτης με μια περιστρεφόμενη επαφή που σχηματίζει ένα ρυθμιζόμενο διαιρέτη τάσης. Το ποτενσιόμετρο είναι ουσιαστικά ένας διαχωριστής τάσης που χρησιμοποιείται για τη μέτρηση του ηλεκτρικού δυναμικού. Στην εφαρμογή χρησιμοποιήθηκε ένα ποτενσιόμετρο 10kΩ για την υλοποίηση της συνδεσμολογίας της οθόνης LCD.[10]



2.2-10 Ποτενσιόμετρο [17]

## 11) Αντιστάτης

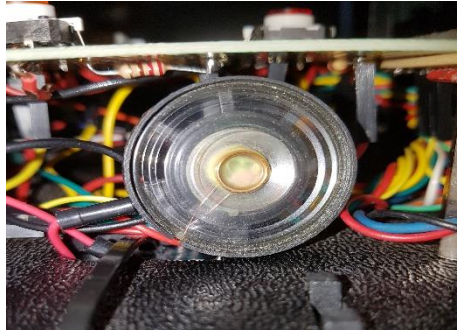
Μια αντίσταση είναι ένα ηλεκτρικό εξάρτημα το οποίο χρησιμοποιείται σε κυκλώματα για τον έλεγχο του ρεύματος. Στη κατασκευή χρησιμοποιήθηκαν αρκετοί και διαφόρων τιμών αντιστάτες ανάλογα με τις ανάγκες του κυκλώματος. [11]



2.2-11 Αντιστάτης 510Ω [18]

## 12) Ηχείο - Speaker

Το speaker είναι ένα μικρό ηχείο το οποίο δέχεται τάση και τη μετατρέπει σε ήχο, Ρυθμίζεται μέσω κώδικα στον οποίο δηλώνεται η συχνότητα και διάρκεια του ήχου που θέλουμε να παραχθεί.[12]



2.2-12 Mini Metal Speaker 8 ohm 0.5W

## 13) Διακόπτης 2 καταστάσεων - Switch button

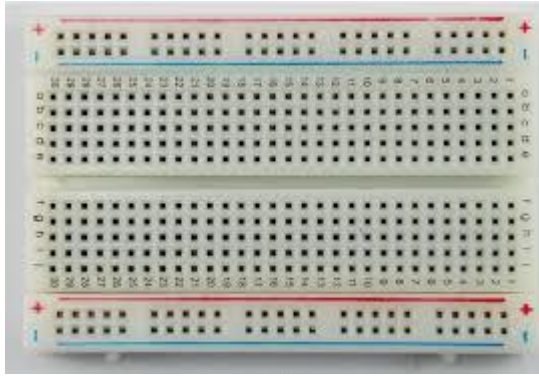
Το switch button είναι ένας διακόπτης 2 καταστάσεων ο οποίος παρέχει θέση μνήμης με τη οποία κρατάει τη τελική του κατάσταση. Στη κατασκευή χρησιμοποιήθηκε μόνο ένας, για την ενεργοποίηση του ρομπότ.[13]



2.2-13 Διακόπτης 2 καταστάσεων - Switch button[30]

## 14) Breadboard

Το Breadboard είναι μία βάση για τη κατασκευή ηλεκτρονικών κυκλωμάτων χωρίς την ανάγκη συγκόλλησης. Διαθέτει έναν αριθμό υποδοχών που συνδέονται εσωτερικά, το πλήθος των υποδοχών είναι ανάλογο του μεγέθους του.[14]



2.2-14 Breadboard [19]

### 15) Μπαταρία Λιθίου

Μπαταρία Λιθίου είναι μία επαναφορτιζόμενη μπαταρία που χρησιμοποιεί ηλεκτρολύτη πολυμερούς λιθίου αντί για υγρό ηλεκτρολύτη, όπως συνηθίζεται. Σε σύγκριση με τις κοινές μπαταρίες προσφέρει υψηλότερη και πιο ακριβή ενέργεια ενώ το μέγεθος και το βάρος της ,την καθιστούν ιδανική για κατασκευές μικρού μεγέθους. Στη κατασκευή χρησιμοποιήθηκε μία μπαταρία λιθίου (11.1V/1300mAh )ως κύρια πηγή τροφοδοσίας. [15]



2.2-15 Μπαταρία λιθίου (11.1V/1300mAh )



## 2.3 Μηχανολογικό Μέρος

Η βάση της κατασκευής είναι ένα σασί με διαστάσεις 196mm L x 126mm W x 44mm H, στο οποίο στερεώθηκαν τα εξής εξαρτήματα.

### 1) Arduino Mega

Το Arduino Mega είναι το βασικό εξάρτημα της κατασκευής πάνω στο οποίο συνδέονται όλα τα εξαρτήματα και σαν μικροελεγκτής διαθέτει το προγραμματισμό και τη διεργασία που πρέπει να εκτελεί κάθε εξάρτημα. Το Arduino έχει τοποθετηθεί στο εσωτερικό της ρομποτικής πλατφόρμας.

### 2) LCD

Το κύκλωμα της LCD (liquid crystal display 16x2) έχει πραγματοποιηθεί με τη χρήση μιας πλακέτας πάνω στην οποία έχουν γίνει οι απαραίτητες συνδεσμολογίες. Η πλακέτα της LCD έχει στερεωθεί στο πάνω μέρος της πλατφόρμας με τη χρήση αποστατών.

### 3) Servo κινητήρες

Στη κατασκευή χρησιμοποιήθηκαν σαν εξάρτημα servo κινητήρες αλλά μέσω διεργασίας έγιναν DC. Οι κινητήρες έχουν εγκατασταθεί στο εσωτερικό της πλατφόρμας με τη χρήση θερμόκολλας.

### 4) Driver L293D

Το driver L293D έχει τοποθετηθεί σε ένα μικρό breadboard στο εσωτερικό της πλατφόρμας μέσω του οποίου γίνονται οι συνδεσμολογίες μεταξύ κινητήρων και Arduino, σύμφωνα με το Datasheet του εξαρτήματος.[32]

### 5) Button κίνησης-Reset button

Τα τέσσερα button κίνησης όπως και το reset button έχουν συνδεθεί στη κεντρική πλακέτα της κατασκευής, μέσω της οποίας έχουν γίνει οι απαραίτητες συνδεσμολογίες μεταξύ των button, των led που είναι ενσωματωμένα πάνω στα button, των αντιστάσεων και του Arduino. Η πλακέτα έχει τοποθετηθεί στο πάνω μέρος της πλατφόρμας με τη χρήση αποστατών.

### 6) Button Start-Stop

Τα button start-stop τοποθετήθηκαν πάνω σε μια πλαστική επιφάνεια, η οποία έχει τροποποιηθεί με τη χρήση του εργαλείου Dremel. Η πλαστική επιφάνεια έχει τοποθετηθεί στο πάνω μέρος της πλατφόρμας με τη χρήση αποστατών.

### 7) Αισθητήριο Ήχου-Sound sensor

Το Αισθητήριο Ήχου συνδέεται απευθείας στο arduino και έχει τοποθετηθεί στο πάνω μέρος της πλατφόρμας με τη χρήση θερμόκολλας.

### **8) Ποτενσιόμετρο**

Το Ποτενσιόμετρο έχει συνδεθεί στη πλακέτα της LCD με ηλεκτρονική κόλληση και στο Arduino. Έχει τοποθετηθεί στο πάνω μέρος της πλακέτας με τη χρήση θερμόκολλας.

### **9) Speaker**

Το speaker είναι συνδεδεμένο απευθείας στο Arduino και έχει τοποθετηθεί στο εσωτερικό της κατασκευής.

### **10) Μπίλια Κίνησης**

Η μπίλια κίνησης χρησιμοποιείται σαν βοηθητικός τροχός για να μην υπάρχουν τριβές στο κάτω μέρος της πλατφόρμας και να μην χάνουν ταχύτητα οι κινητήρες κατά τη διάρκεια της κίνησης. Η μπίλια έχει τοποθετηθεί στο μπροστά-κάτω μέρος της κατασκευής με τη χρήση θερμόκολλας .

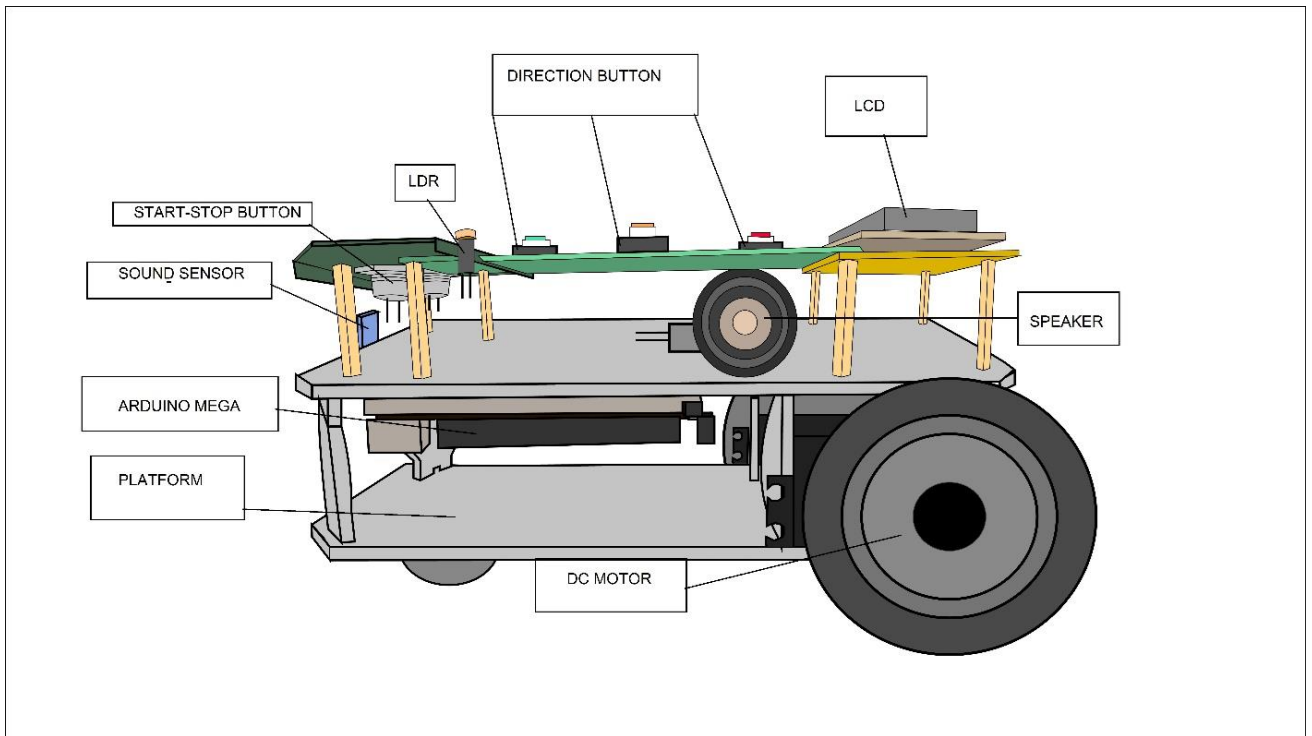
### **11) Breadboard**

Έγινε χρήση δύο breadboard για να υλοποιηθούν οι συνδεσμολογίες για τα ρεύματα και τις γειώσεις. Τα breadboard έχουν στερεωθεί με τη χρήση της κόλλας που είχαν ενσωματωμένη πάνω τους.

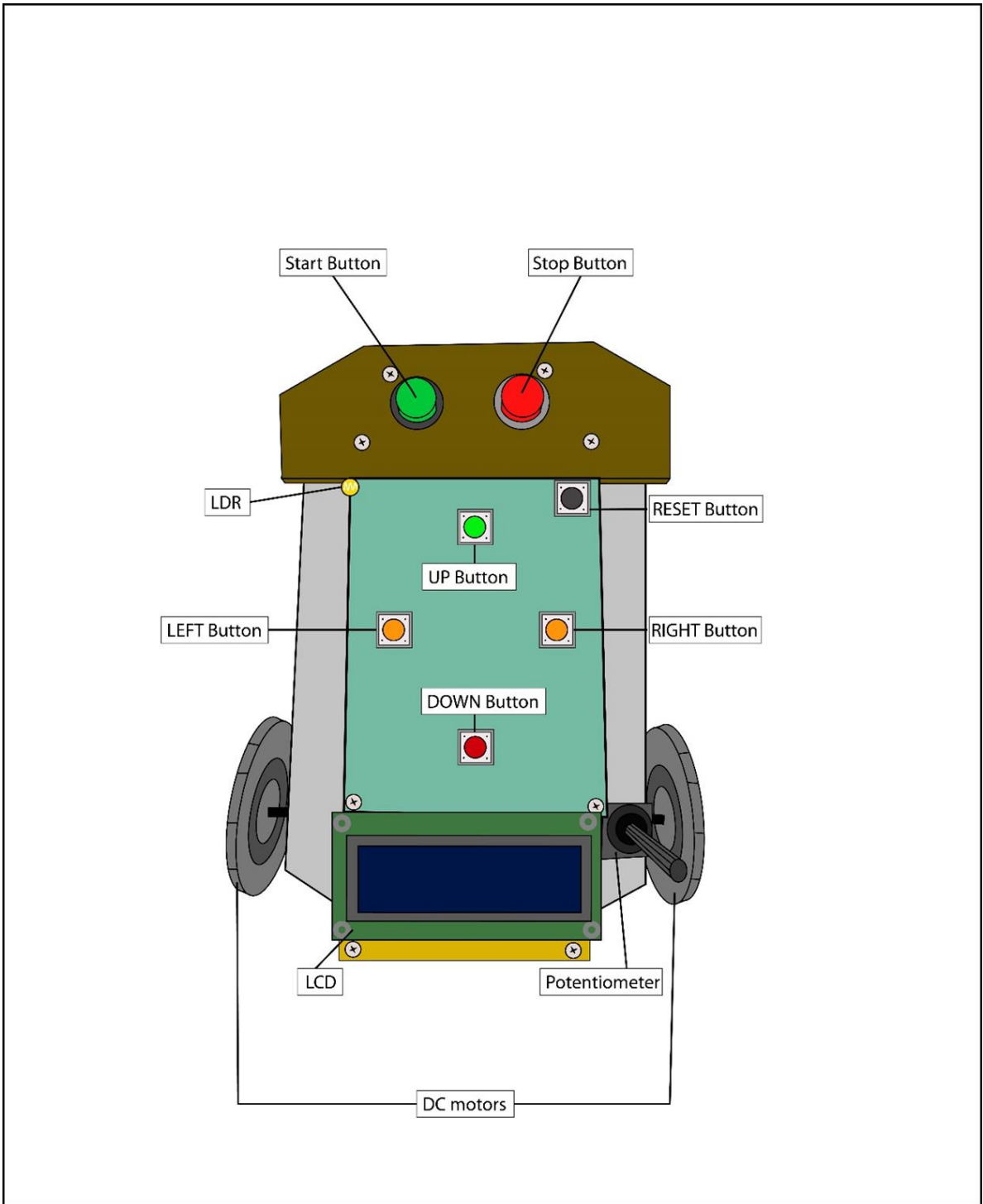
### **12) Φορητή μπαταρία**

Η τροφοδοσία της κατασκευής γίνεται μέσω μιας μπαταρίας 1300 mA η οποία έχει συνδεθεί σε διακόπτη switch και μέσω διεργασίας έχει γίνει διάσπαση των τάσεων της μπαταρίας ώστε να δίνει τάση 5 Volt στο Arduino και 7volt στους DC κινητήρες. Αυτό έγινε διότι οι κινητήρες χρειάζονται μεγαλύτερη τροφοδοσία από τα 5 volt που μπορεί να παρέχει το Arduino.

Η τοποθέτηση των εξαρτημάτων έγινε όπως φαίνεται στις εικόνες 2.3-1 και 2.3—2.

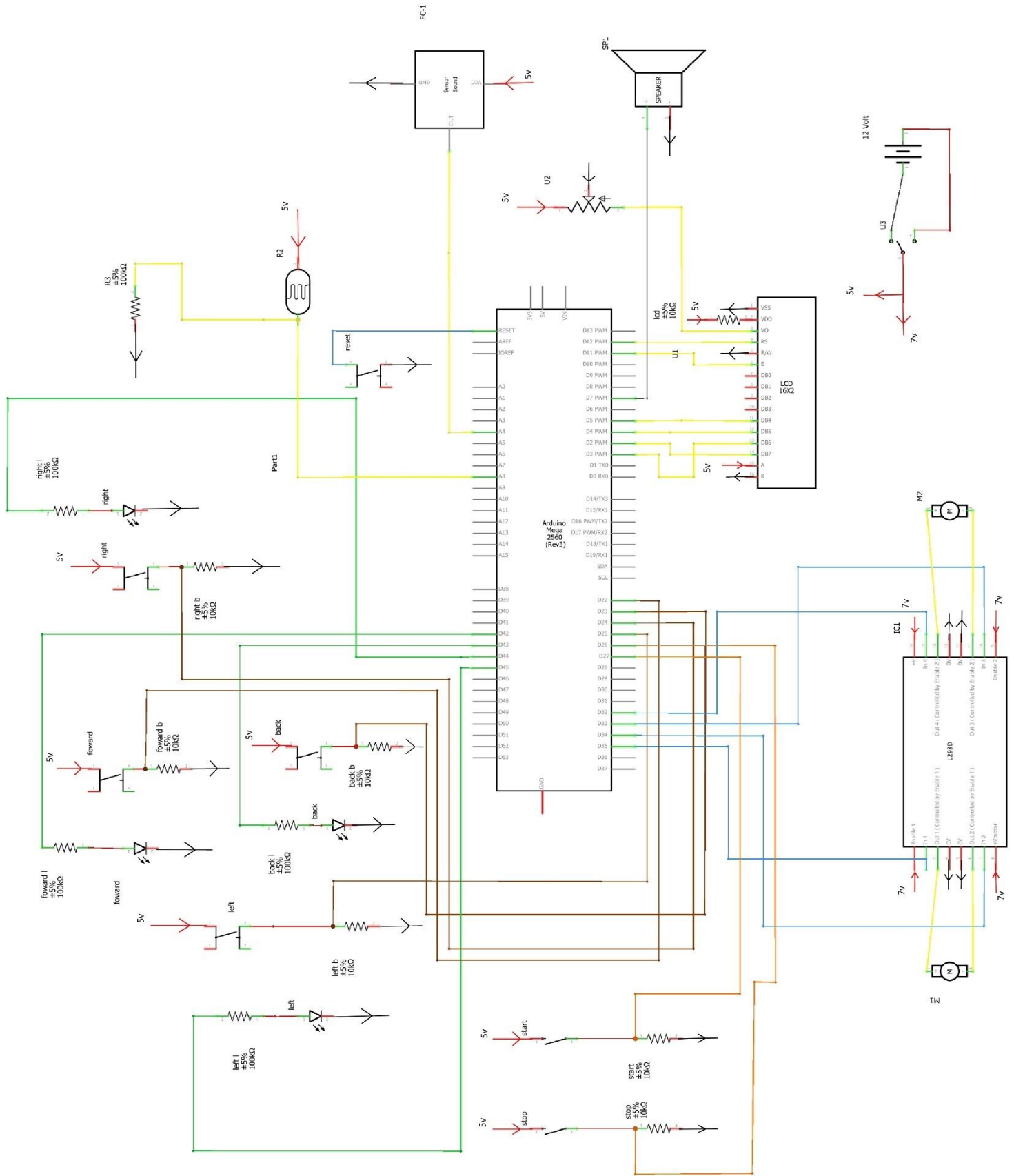


Εικόνα 2.3-1 Αριστερή πλάγια όψη

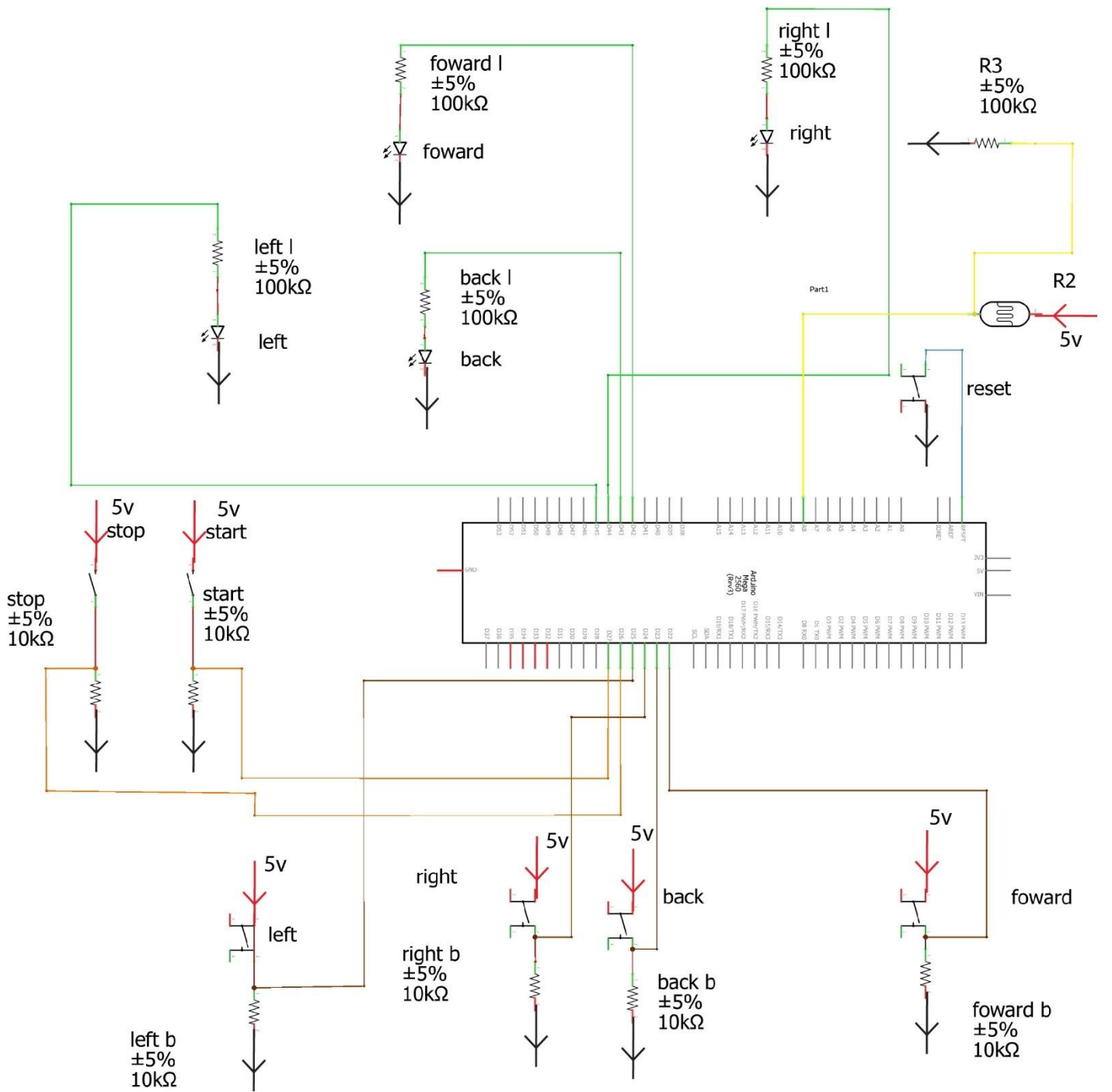


2.3—2 Άνοψη κατασκευής

## 2.4 Ηλεκτρονικό Μέρος

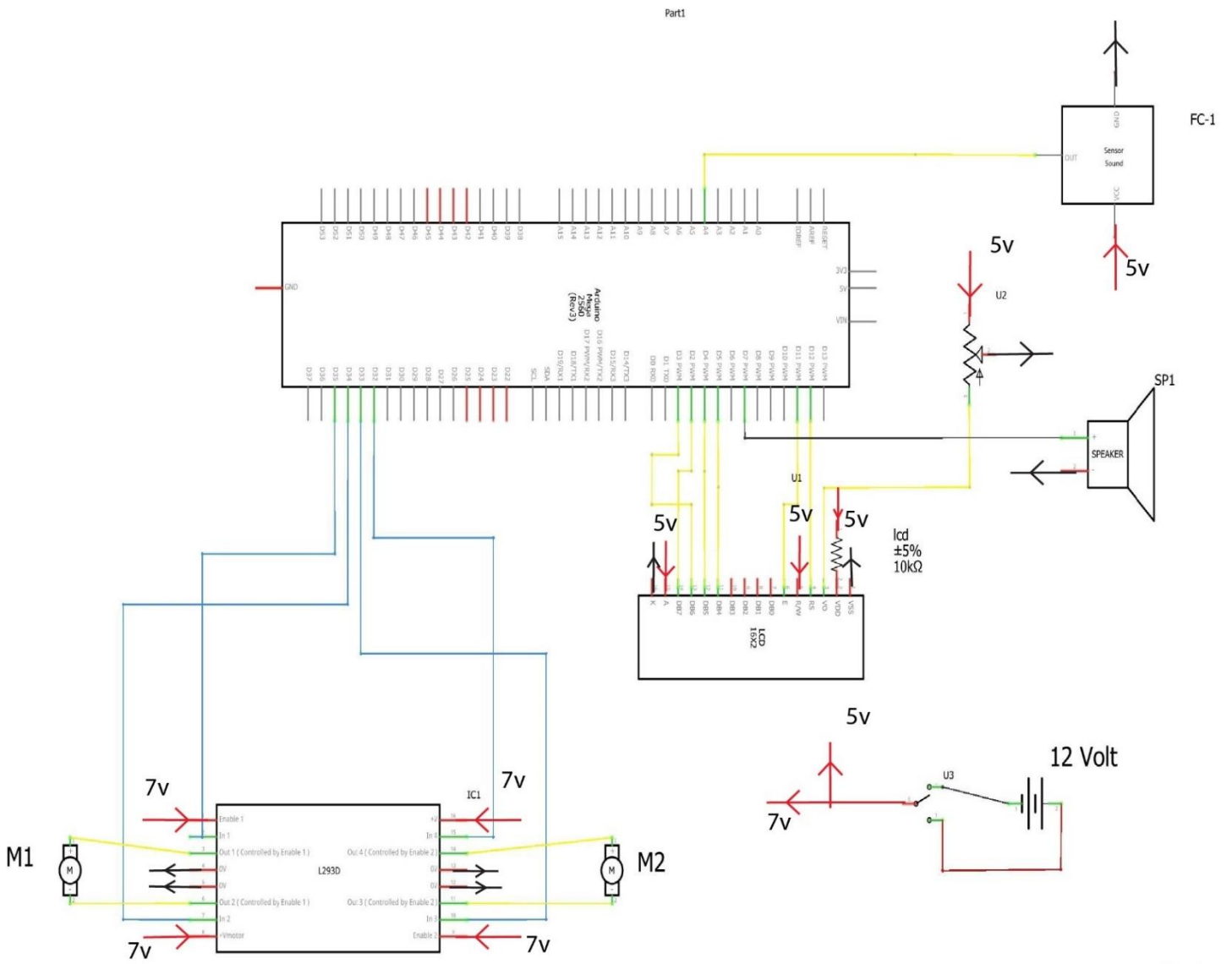


Εικόνα 2.3-1 Ηλεκτρονικό Σχέδιο



fritzing

Εικόνα 2.4-2 Ηλεκτρονικό σχέδιο μέρος 1



Εικόνα 2.4-3 Ηλεκτρονικό σχέδιο μέρος 2

fritzing

## **3 Ανάλυση Λογισμικού – Software**

### **3.1 Εισαγωγή**

Στο κεφάλαιο αυτό αναλύεται το λογισμικό της κατασκευής. Αρχικά θα γίνει μία περιληπτική περιγραφή του κώδικα ,στη συνέχεια θα αποτυπωθεί με τη μορφή διαγράμματος ροής και τέλος αναλυτική αποτύπωση του κώδικα με σχολιασμό.

### **3.2 Ο κώδικας περιληπτικά**

Το ρομπότ έχει τέσσερα προγράμματα λειτουργίας. Αρχικά το κάθε πρόγραμμα υλοποιήθηκε ξεχωριστά και στη συνέχεια ενοποιήθηκαν όλα μαζί σε ένα κώδικα.

Ο χρήστης πλοηγείται ανάμεσα στα προγράμματα με ένα ενιαίο μενού κάνοντας χρήση των πλήκτρων, η επιλογή ενός μενού γίνεται πατώντας το κουμπί Start, μέσα στο μενού περιλαμβάνονται και οδηγίες για το κάθε πρόγραμμα. Όταν ο χρήστης έχει επιλέξει κάποιο πρόγραμμα, η επιστροφή στο αρχικό μενού γίνεται πατώντας το κουμπί Reset.

Σε όλα τα προγράμματα γίνεται χρήση της οθόνης LCD καθώς και των κινητήρων. Ο χρήστης χρησιμοποιώντας τα πλήκτρα κινήσεων εισάγει εντολές οι οποίες εμφανίζονται στην οθόνη LCD, με τη μορφή συμβόλων. Πατώντας το κουμπί Play γίνεται εκκίνηση του προγράμματος. Πατώντας το κουμπί Delete, πριν την εκκίνηση του προγράμματος ο χρήστης μπορεί να διαγράψει μία κίνηση, αν ο χρήστης πατήσει το κουμπί Delete κατά την εκτέλεση ενός προγράμματος, το πρόγραμμα σταματάει. Όταν το ρομπότ εκτελεί μία κίνηση υπάρχει και η αντίστοιχη φωτεινή ένδειξη, από τα LED που είναι ενσωματωμένα στα button κίνησης.

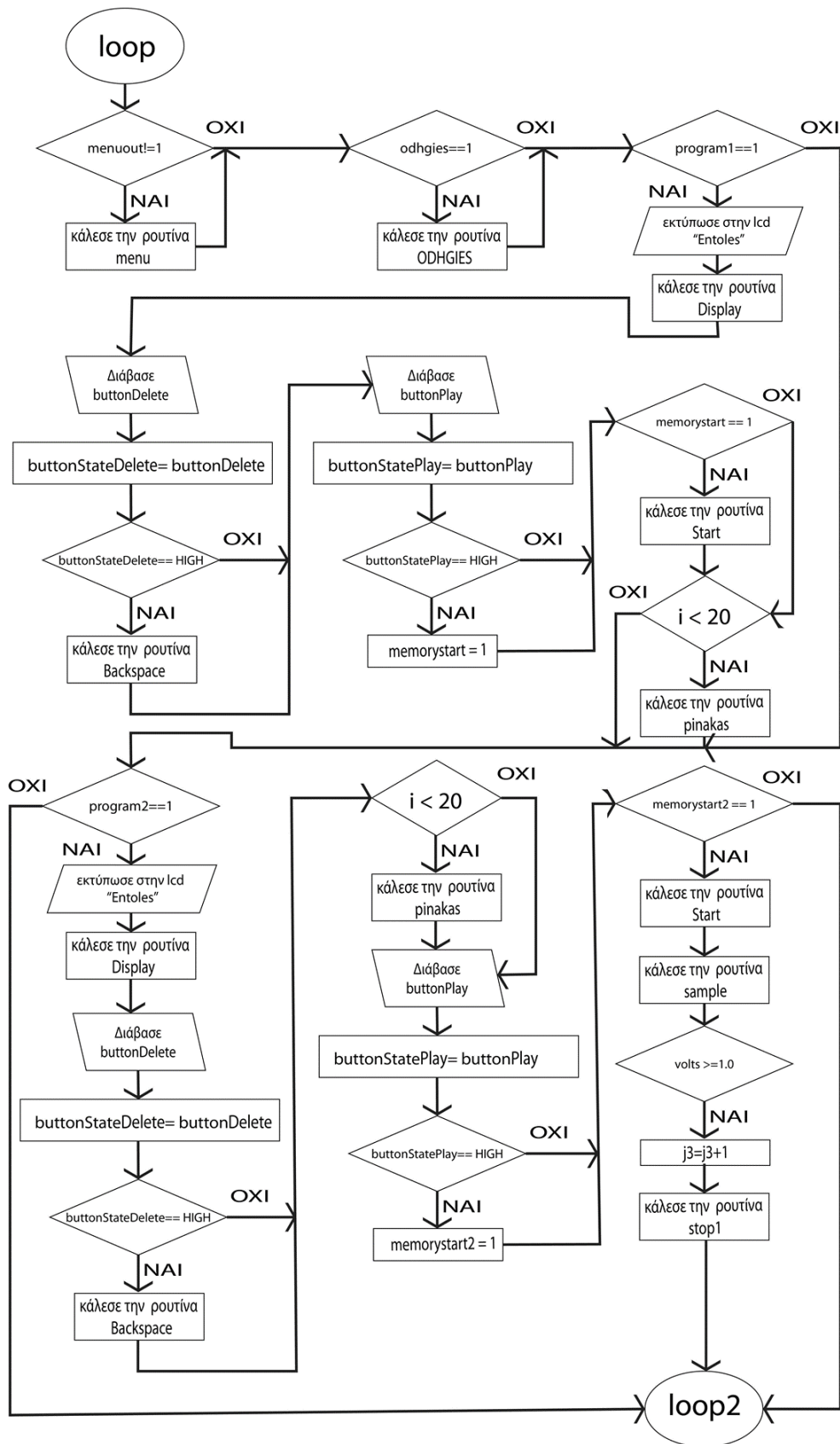
Στο πρώτο πρόγραμμα ο χρήστης εισάγει απλά τις κινήσεις και πατώντας το κουμπί Play το ρομπότ τις εκτελεί, παράγοντας ταυτόχρονα μία νότα. Κάθε κίνηση διαρκεί ένα δευτερόλεπτο ,όταν εκτελεστούν όλες οι κινήσεις το ρομπότ σταματάει.

Στο δεύτερο πρόγραμμα ο χρήστης εισάγει τις κινήσεις με τη χρήση των πλήκτρων και πατώντας το Play γίνεται εκκίνηση. Το Arduino διαβάζει συνεχόμενα τις τιμές που στέλνει το αισθητήριο ήχου και με τη χρήση ενός αλγόριθμου εντοπίζει αν υπάρχει κάποια μεγάλη διακύμανση, δηλαδή ένας έντονος ήχος. Κατά την εκκίνηση του προγράμματος το ρομπότ εκτελεί την πρώτη εντολή κίνησης, όταν εντοπιστεί κάποιος έντονος ήχος σταματάει και εκτελεί την επόμενη έως ότου να εκτελεστούν όλες τις κινήσεις.

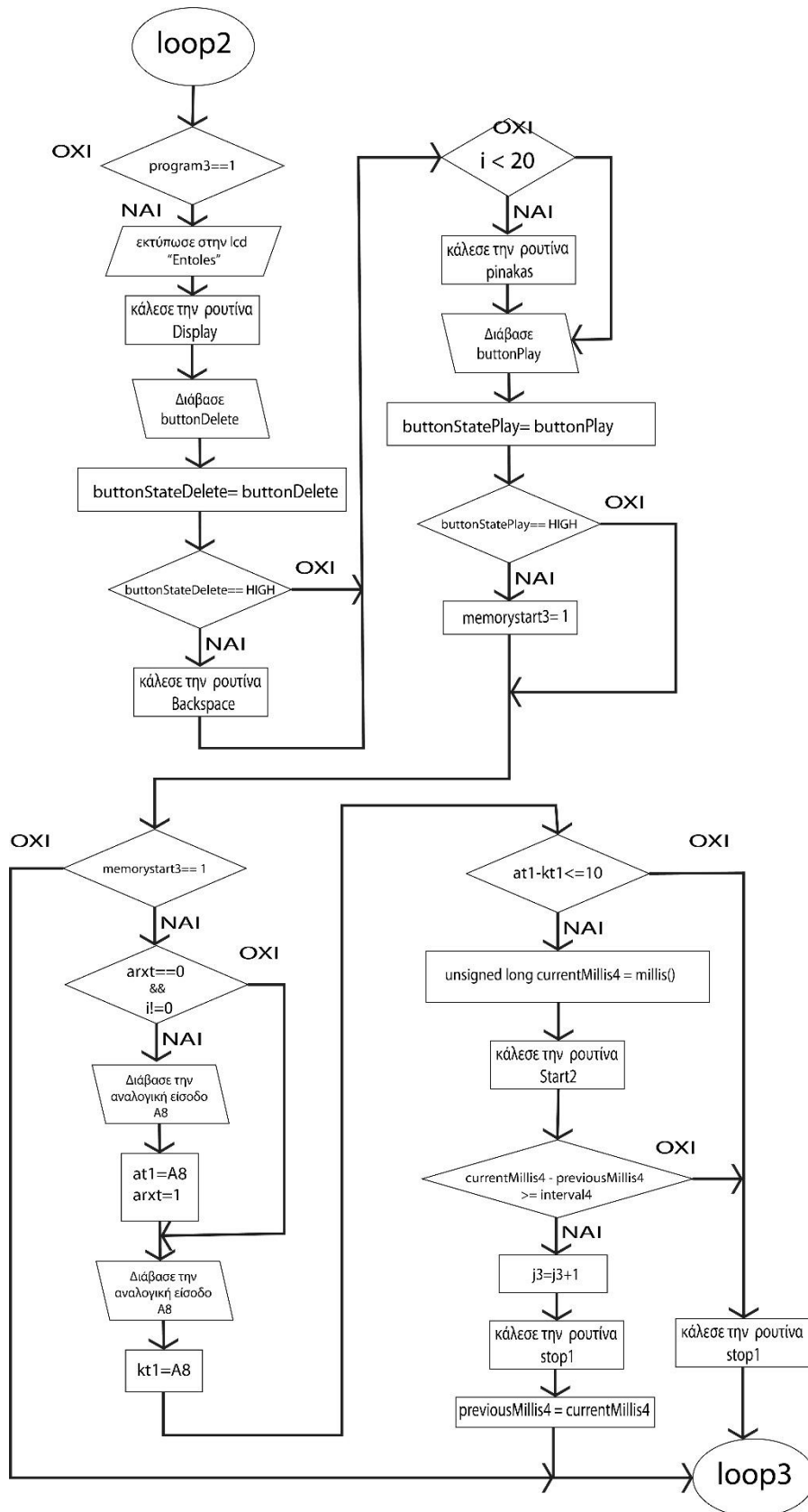


Το τρίτο και το τέταρτο πρόγραμμα είναι παρόμοια, ο χρήστης εισάγει με τη χρήση των πλήκτρων τις κινήσεις και πατώντας το Play γίνεται η εκκίνηση. Σε αυτά τα προγράμματα χρησιμοποιείται το αισθητήριο φωτός LDR, το οποίο κατά την εκκίνηση διαβάζει την ένταση του φωτός στο δωμάτιο και το αποθηκεύει ως αρχική τιμή. Αφού αποθηκευτεί η αρχική τιμή του φωτός ακολουθεί μία μικρή καθυστέρηση και στη συνέχεια ξεκινάει το ρομπότ να εκτελεί τις κινήσεις. Στο ένα πρόγραμμα μεν μόνο όταν βλέπει φως δηλαδή ένταση μεγαλύτερη της αρχικής, στο άλλο δε όταν είναι στο σκοτάδι δηλαδή βλέπει ένταση μικρότερη της αρχικής. Η συνολική διάρκεια της κάθε κίνησης είναι ένα δευτερόλεπτο.

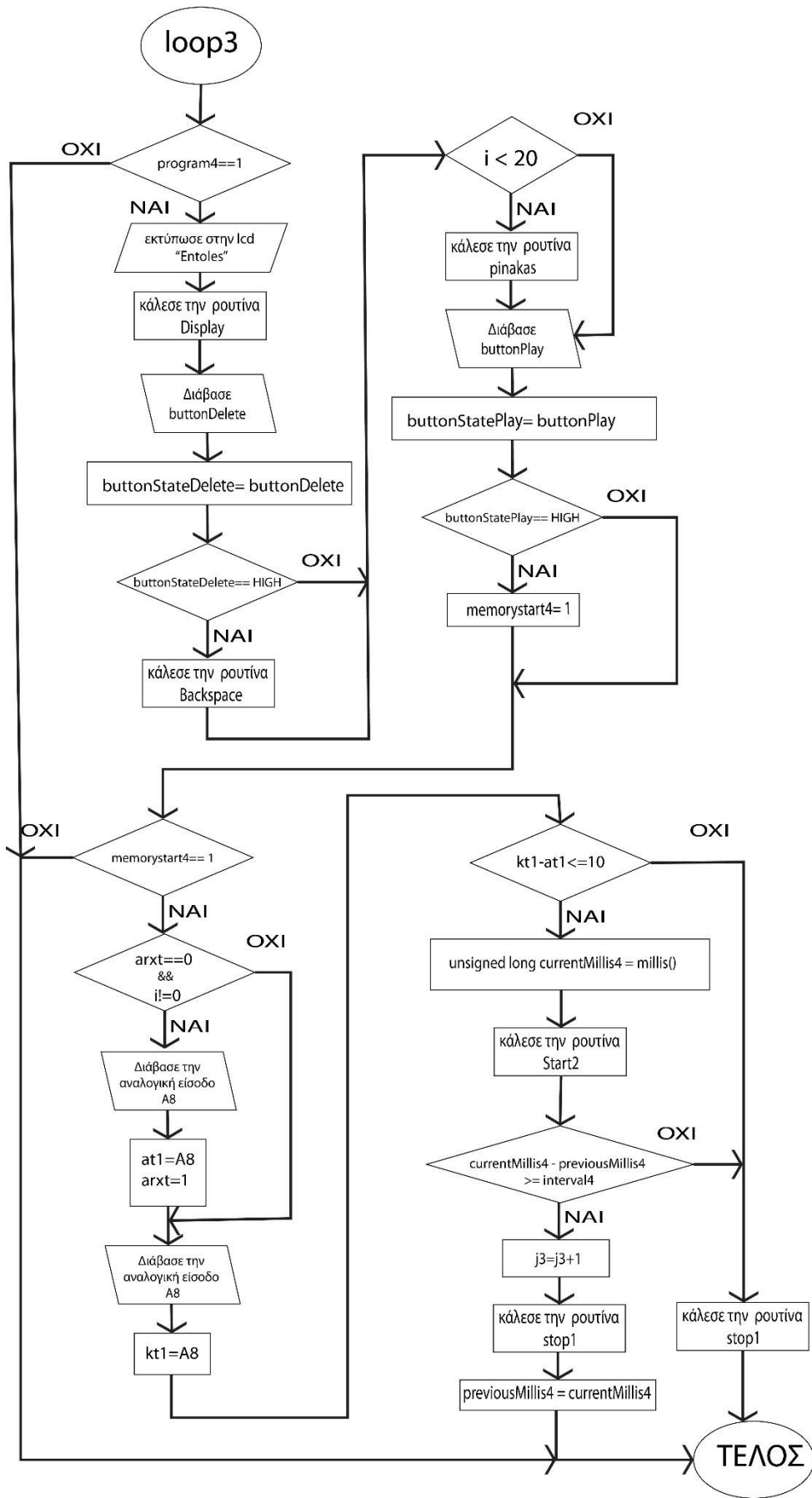
### 3.3 Διαγράμματα Ροής



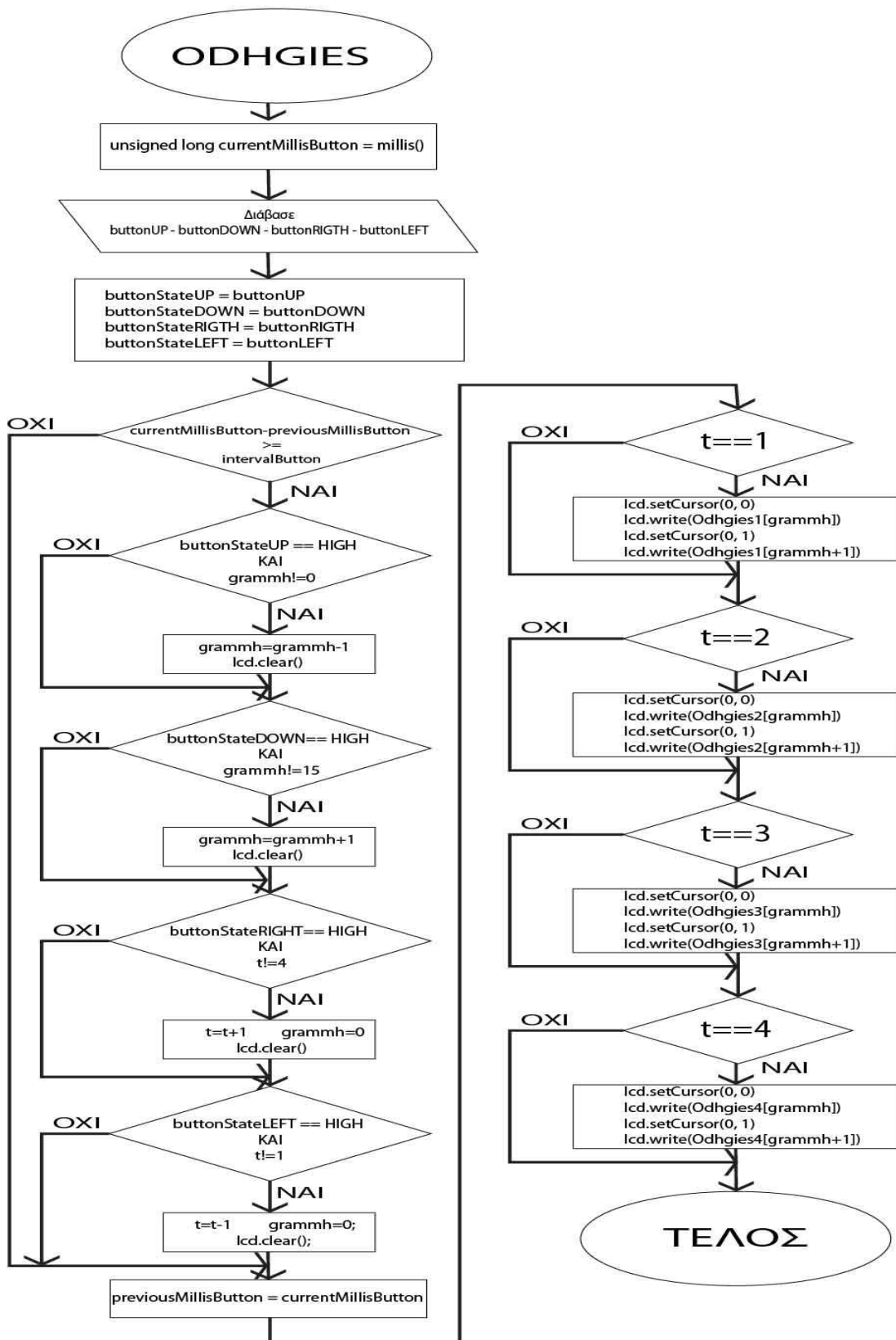
3.2-1 Διάγραμμα ροής ρουτίνα loop μέρος 1



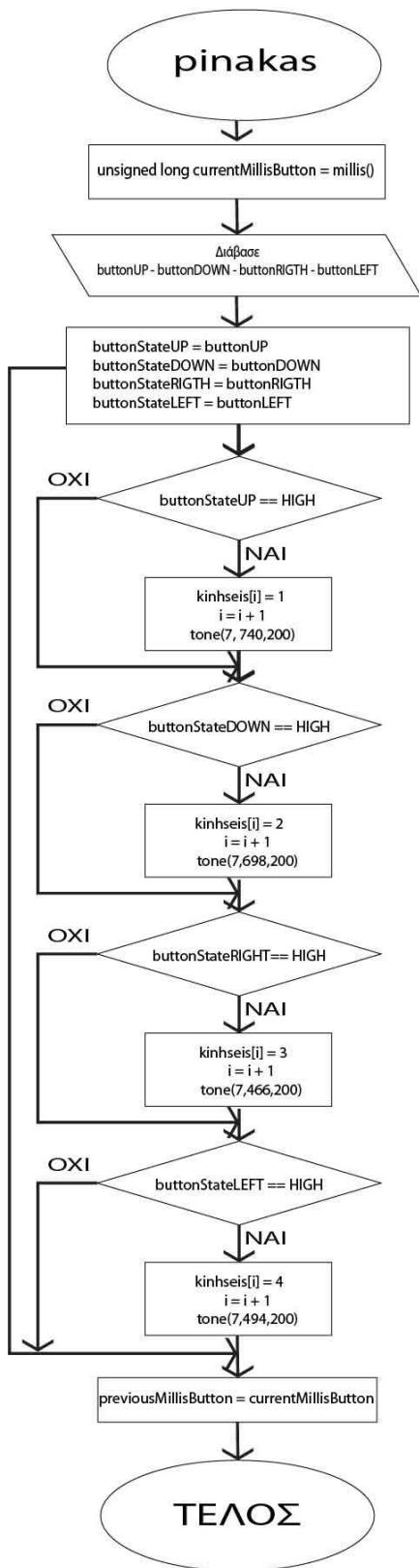
3.3-2 Διάγραμμα ροής ρουτίνα loop μέρος 2



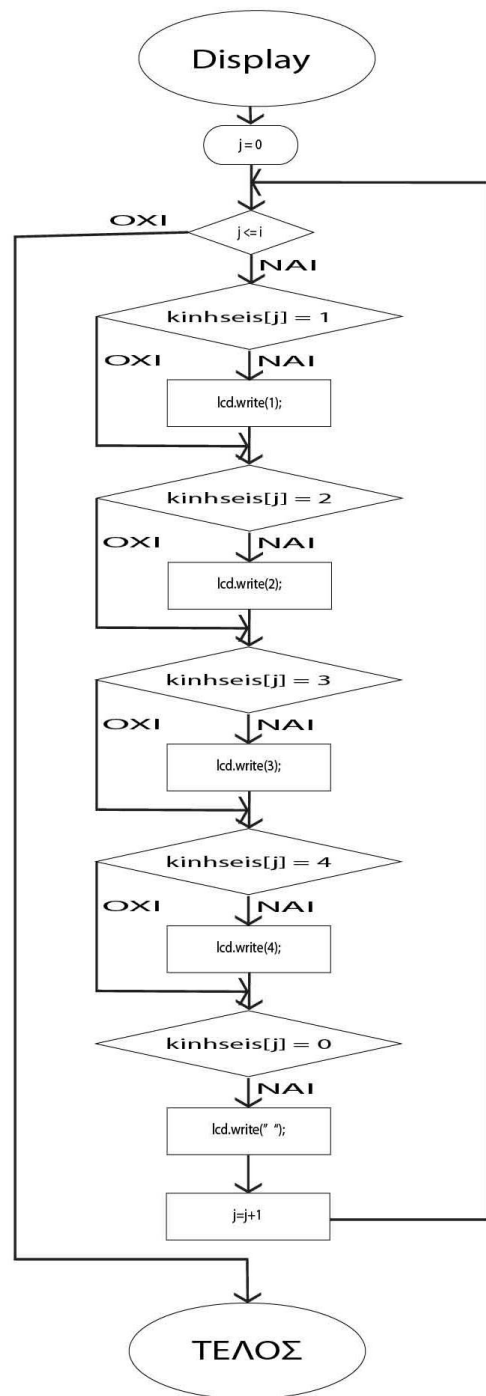
3.3-3 Διάγραμμα ροής ρουτίνα loop μέρος 3



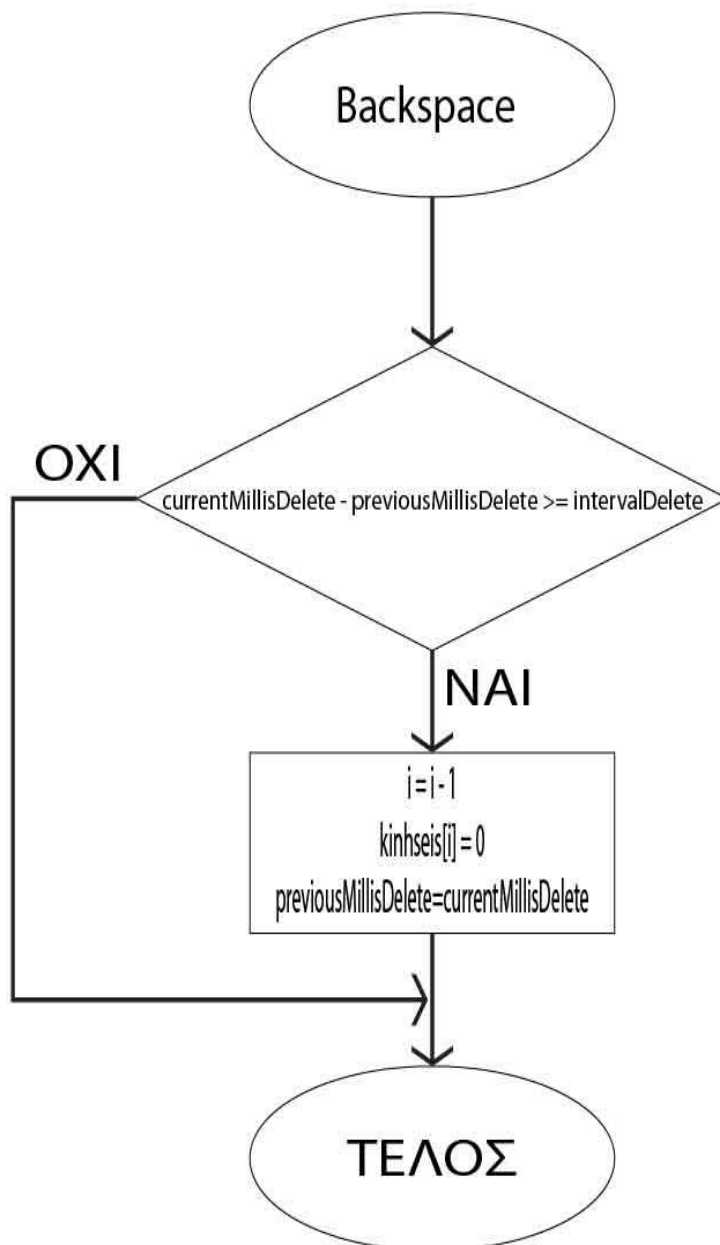
3.3-4 Διάγραμμα ροής ρουτίνας ODHGIES



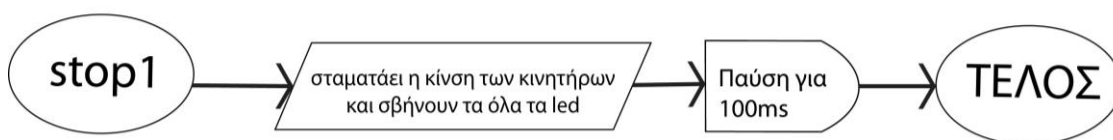
3.3—5 Διάγραμμα ροής ρουτίνας pinakas



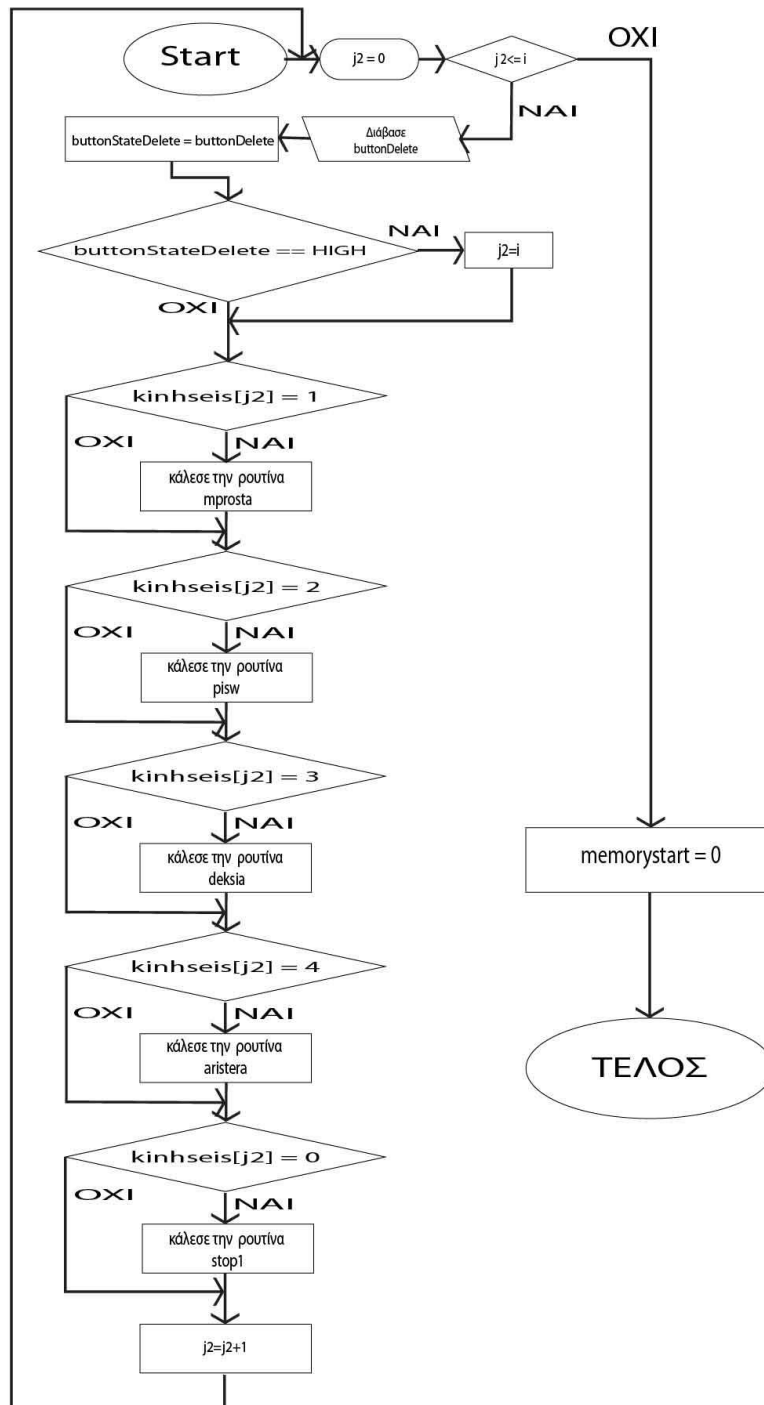
3.3—6 Διάγραμμα ροής ρουτίνας Display



3.3—7 Διάγραμμα ροής ρουτίνας Backspace

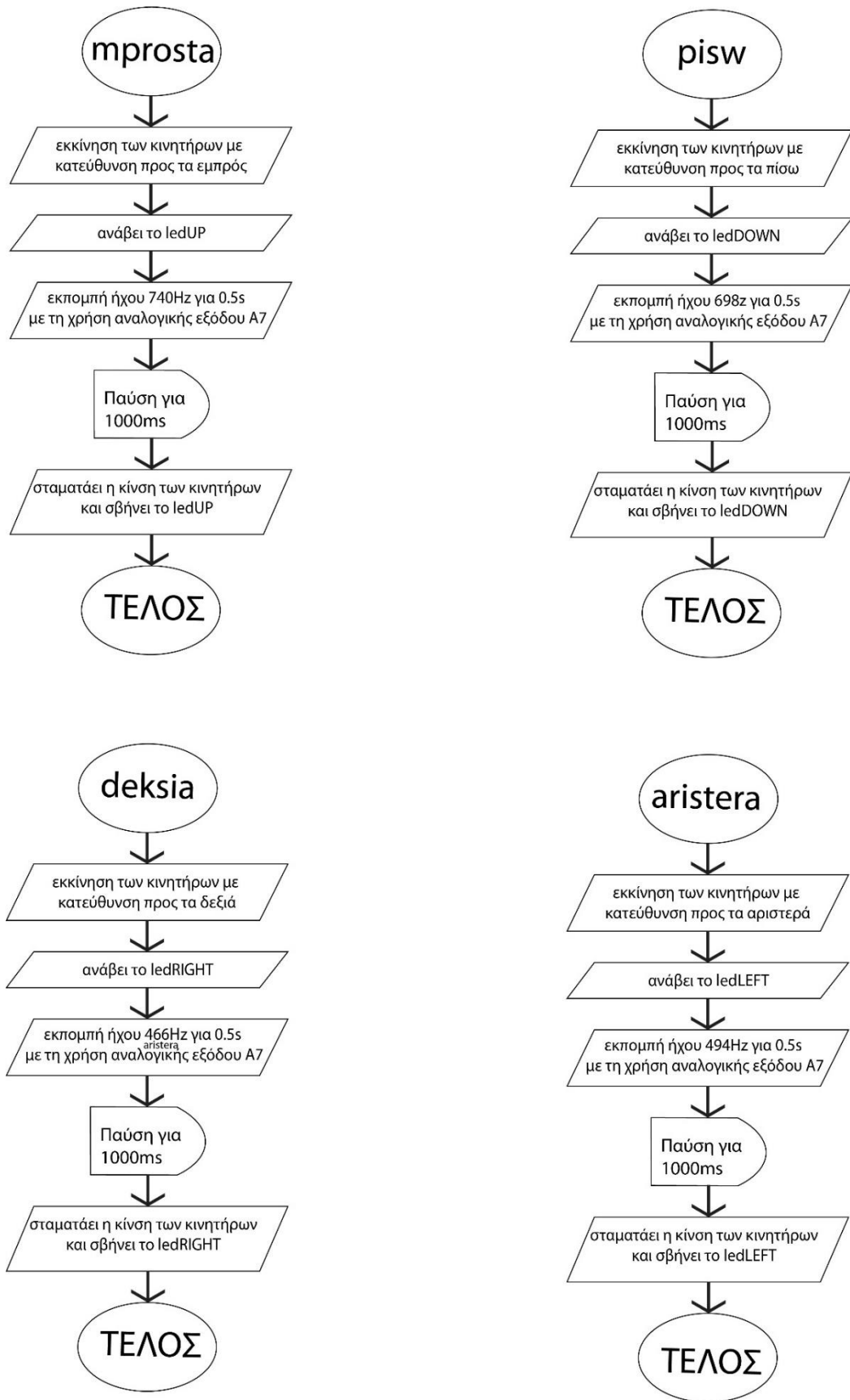


3.3—8 Διάγραμμα ροής ρουτίνας stop1

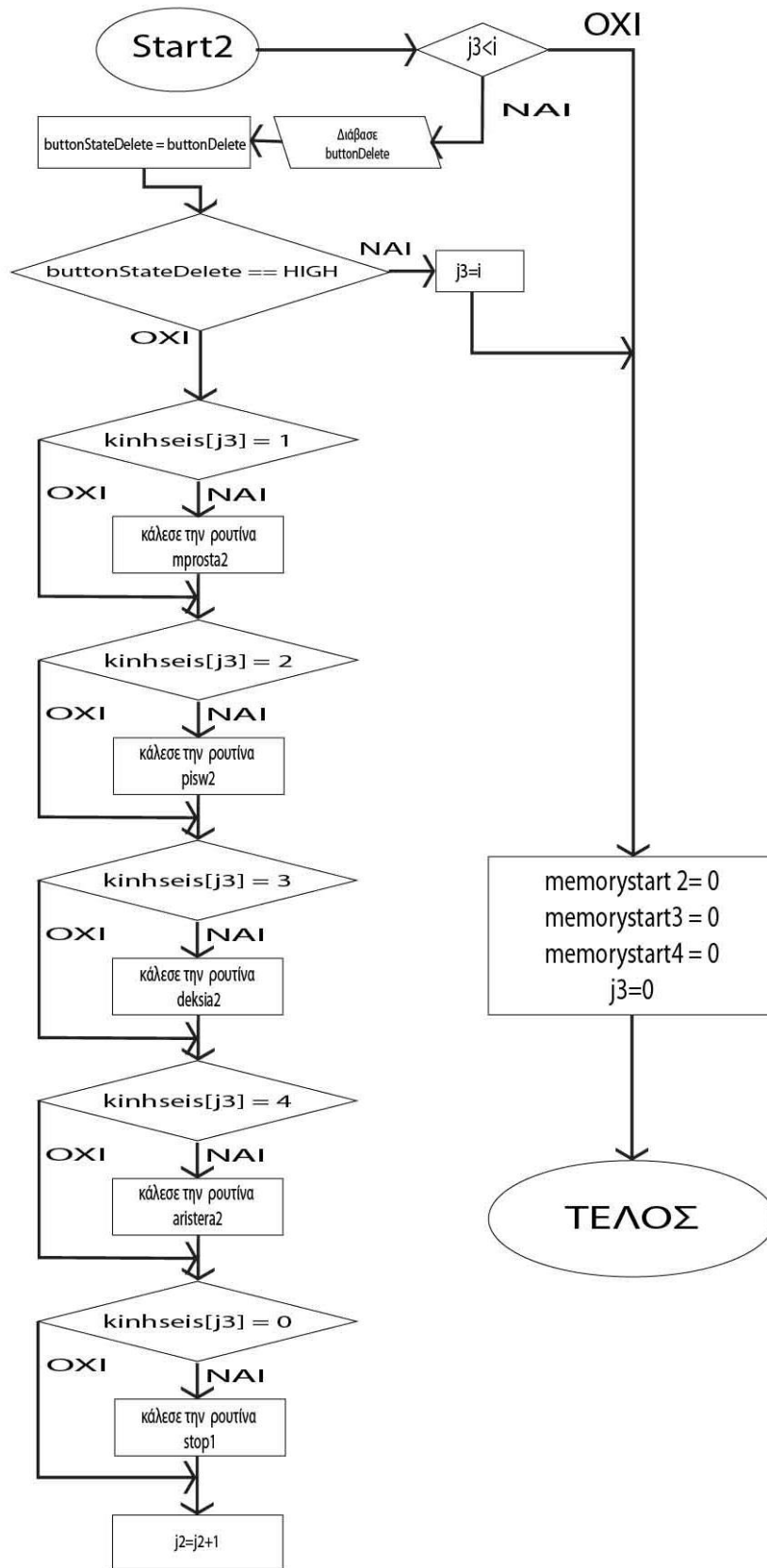


3.3—9 Διάγραμμα ροής ρουτίνας Start

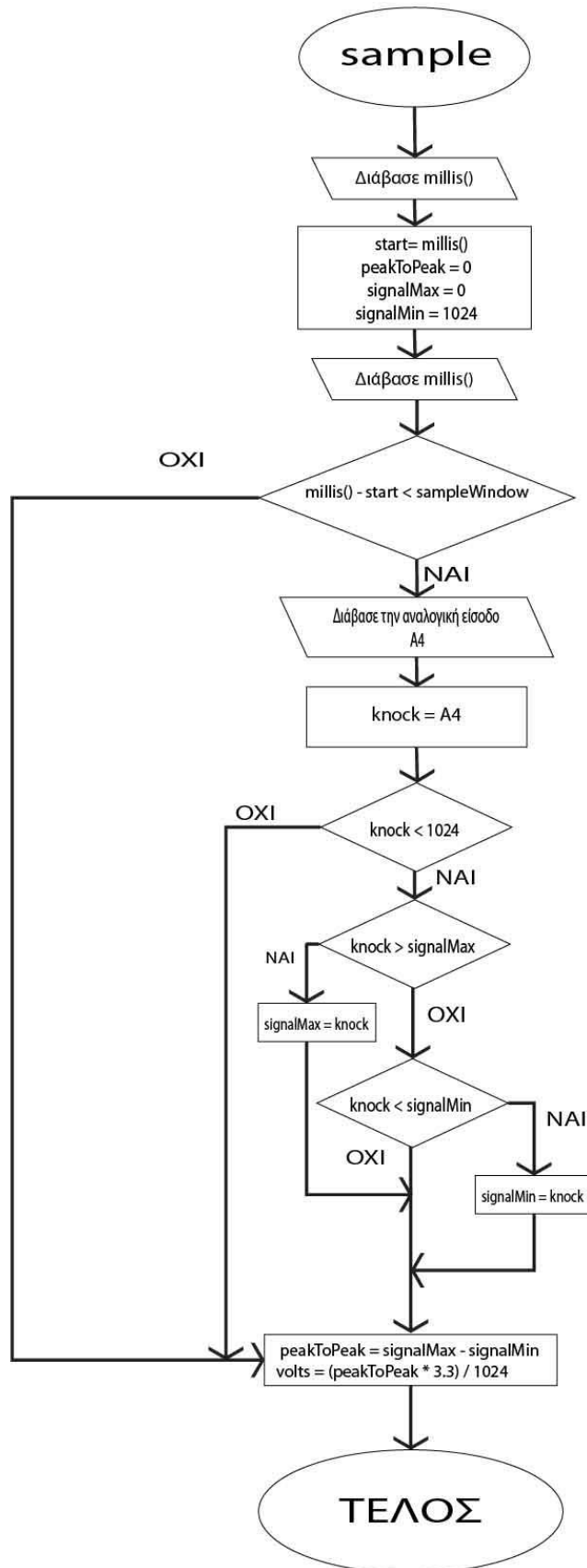




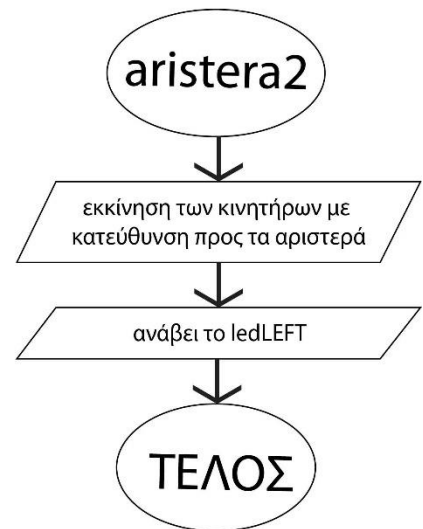
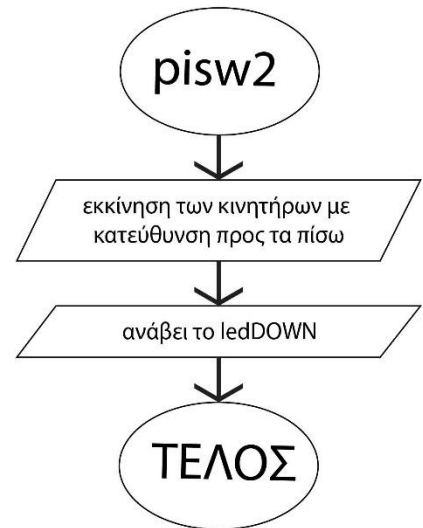
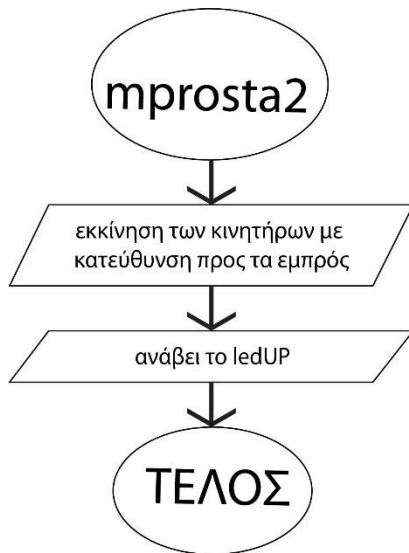
Εικόνα 3.3-10 Διαγράμματα ροής για τις ρουτίνες mprosta, pisw, deksia και aristera



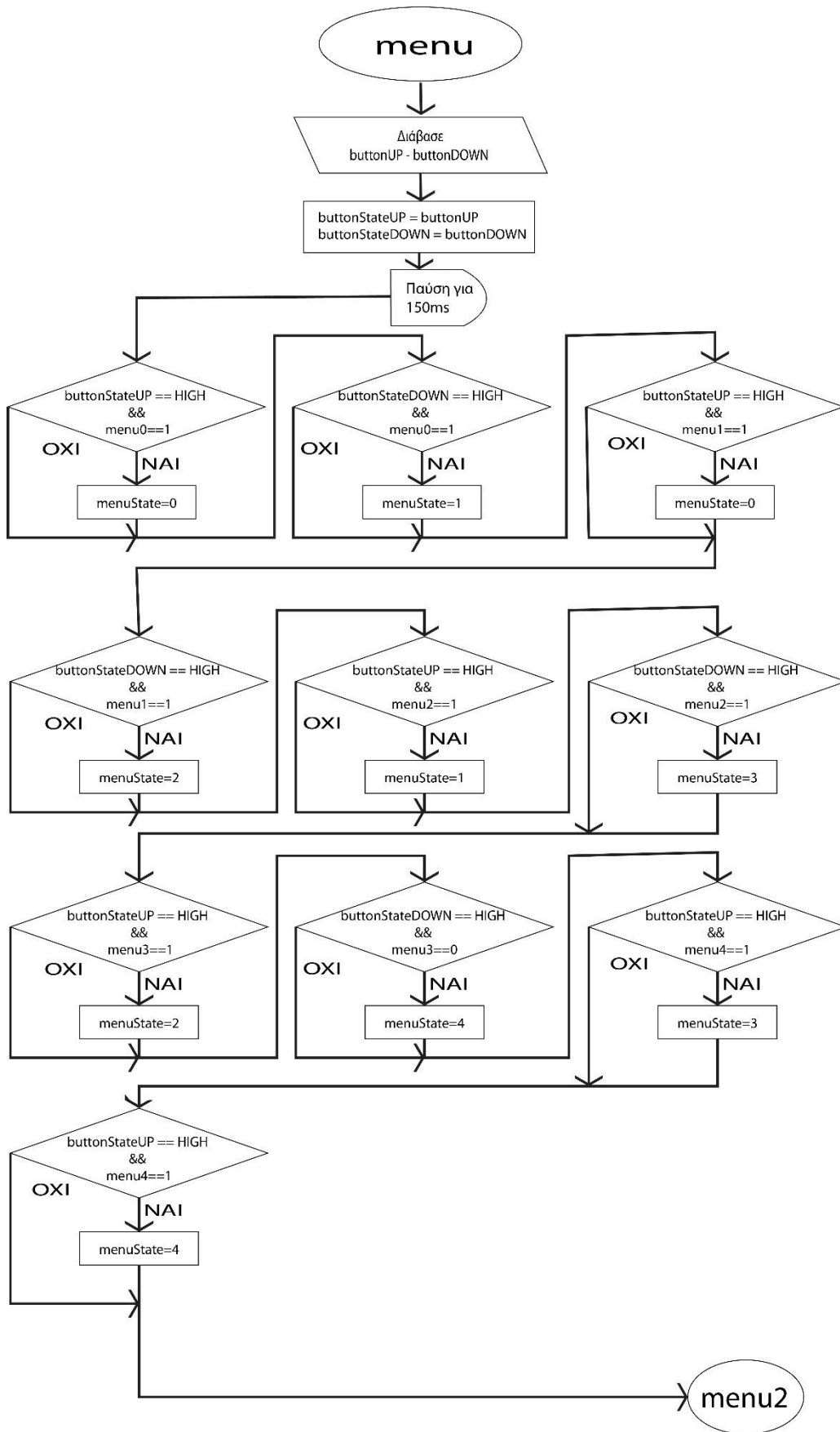
Εικόνα 3.3-11 Διάγραμμα ροής για τη ρουτίνα Start2



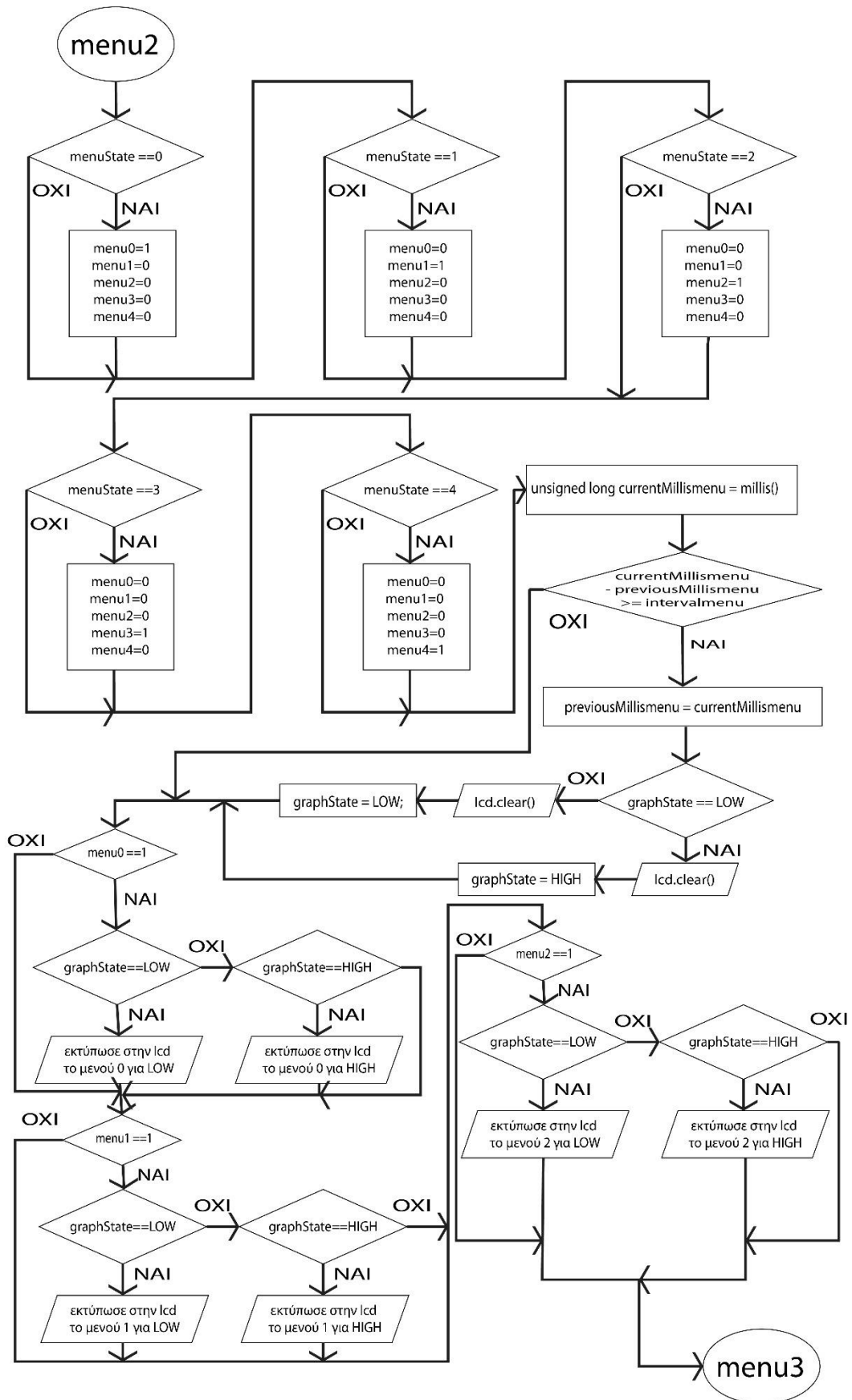
Εικόνα 3.3-12 Διάγραμμα ροής για τη ρουτίνα sample



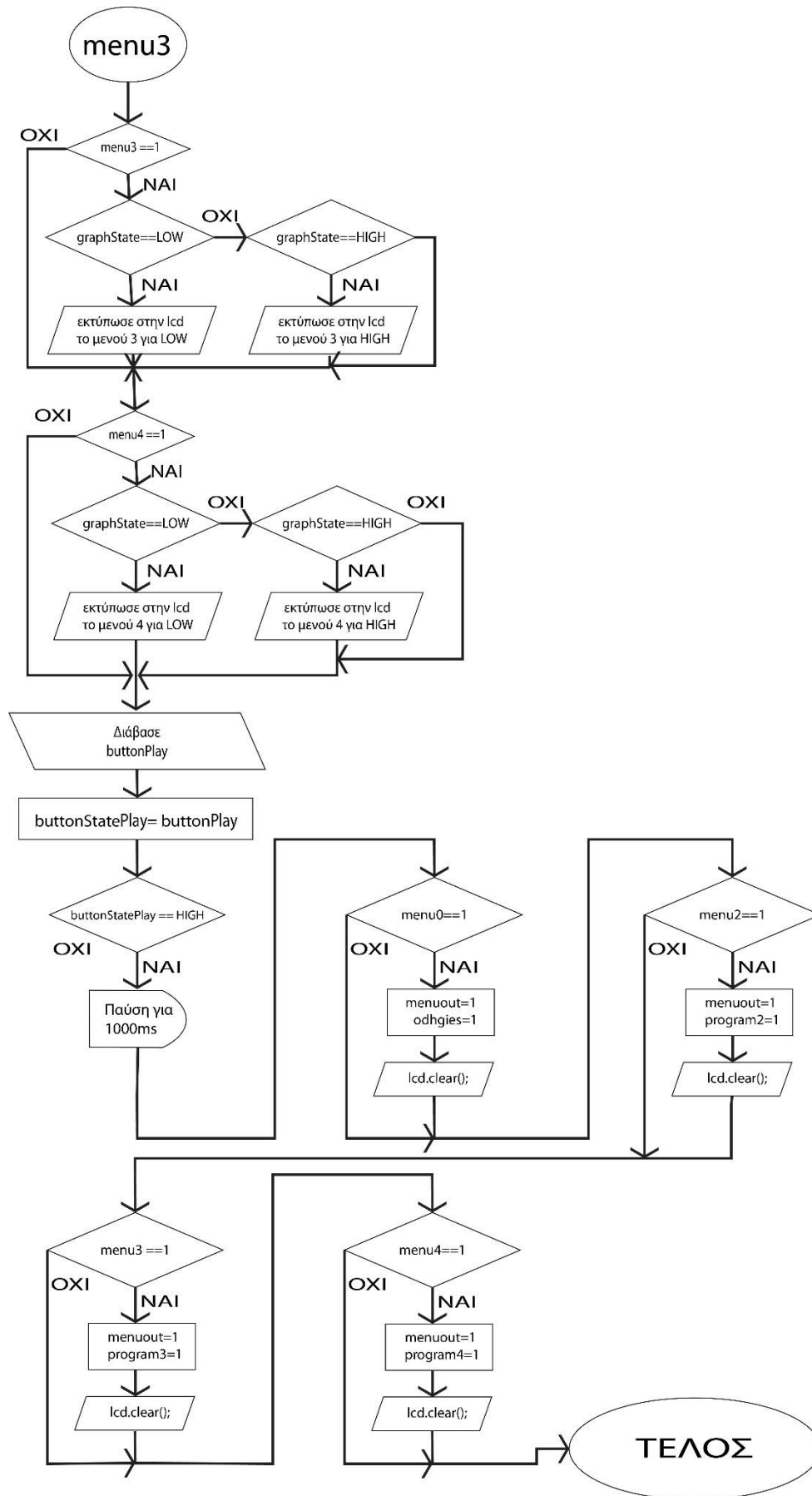
Εικόνα 3.3-13 Διαγράμματα ροής για τις ρουτίνες mprosta2, pisw2, deksia2 και aristera2



Εικόνα 3.3-14 Διάγραμμα ροής ρουτίνα menu μέρος 1



Εικόνα 3.3-15 Διάγραμμα ροής ρουτίνα menu μέρος 2



Εικόνα 3.3-16 Διάγραμμα ροής ρουτίνα menu μέρος 3

### 3.4 Κώδικας Προγράμματος

- Στην αρχή του κώδικα αρχικοποιούνται οι διάφορες μεταβλητές που χρησιμοποιούνται μέσα στο κώδικα και εισάγονται όλες οι βιβλιοθήκες που χρειάζονται.

```
int kinhseis[16] = {};
```

```
int i = 0;
```

```
int grammh;
```

```
int t=1;
```

```
int memorystart = 0;
```

```
int memorystart2 =0 ;
```

```
int memorystart3 =0 ;
```

```
int memorystart4 =0 ;
```

```
int clap =0 ;
```

```
int memorystartonce = 0;
```

```
int odhgies=0;
```

```
int program1=0;
```

```
int program2=0;
```

```
int program3=0;
```

```
int program4=0;
```

```
char* Odhgies1[]={ "Aplh kinhsh", "Se auto to", "programma to", "robot ektelei", "tis kinhseis  
pou", "eisagoume gia 1s", "Afou eisagetai", "tis kinhseis",  
"xrhsimopointas ", "ta plhktra ", "pathste PLAY" , "gia na arxisei", "h kinhsh,pathste", "STOP  
gia na", "stamathsei h'gia", "na diagrapsete", "mia kinhsh."};
```

```
char* Odhgies2[]={ "Kinhsh me Hxo", "Se auto to", "programma to", "robot allazei", "kinhsh  
otan", "akouei thorubo", "Afou eisagetai", "tis kinhseis",  
"xrhsimopointas ", "ta plhktra ", "pathste PLAY" , "gia na arxisei", "h kinhsh,pathste", "STOP  
gia na", "stamathsei h'gia", "na diagrapsete", "mia kinhsh."};
```

```
char* Odhgies3[]={ "Kinhsh sto fws", "Se auto to", "programma to", "robot kineitai", "mono  
otan", "einai sto fws.", "Afou eisagetai", "tis kinhseis",  
"xrhsimopointas ", "ta plhktra ", "pathste PLAY" , "gia na arxisei", "h kinhsh,pathste", "STOP  
gia na", "stamathsei h'gia", "na diagrapsete", "mia kinhsh."};
```



```
char* Odhgies4[]={ "Kinhsh sto ", "skotadi", "Se auto to", "programma to", "robot  
kineitai", "mono sto skotadi", "Afou eisagetai", "tis kinhseis",  
"xrhsimopointas ", "ta plhktra ", "pathste PLAY" ,"gia na arxisei", "h kinhsh,pathste", "STOP  
gia na", "stamathsei h'gia", "na diagrapsete", "mia kinhsh."};
```

```
//μεταβλητές που χρησιμοποιούνται στο μενού
```

```
int menuState=0;  
int menu0=1;  
int menu1;  
int menu2;  
int menu3;  
int menu4;  
int menuout=0;  
unsigned long previousMillismenu = 0;  
const long intervalmenu = 500;  
int graphState=LOW ;
```

```
//μεταβλητές που χρησιμοποιούνται στο δεύτερο πρόγραμμα
```

```
const int sampleWindow = 250;  
unsigned int knock;  
unsigned long start;  
unsigned int peakToPeak ;  
unsigned int signalMax ;  
unsigned int signalMin ;  
double volts;  
int j3=0;
```

```
//μεταβλητές που χρησιμοποιούνται στα προγράμματα τρία και τέσσερα
```

```
float at1;  
float kt1;  
int arxt=0;  
unsigned long previousMillis4 = 0;  
const long interval4 = 1000;
```

```
//millis Button+Delete
unsigned long previousMillisButton = 0;
const long intervalButton = 300;
unsigned long previousMillisDelete = 0;
const long intervalDelete = 300;
```

```
//lcd
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
//button
const int buttonUP = 22;
const int buttonDOWN = 23;
const int buttonRIGTH = 24;
const int buttonLEFT = 25;
const int buttonDelete = 26;
const int buttonPlay = 27;
int buttonStateUP = 0;
int buttonStateDOWN = 0;
int buttonStateRIGTH = 0;
int buttonStateLEFT = 0;
int buttonStateDelete = 0;
int buttonStatePlay = 0;
```

```
//LED
const int ledUP = 42;
const int ledDOWN = 43;
const int ledRIGHT = 44;
const int ledLEFT = 45;
```

- Με την εντολή «byte data[8]» δημιουργούμε ένα χαρακτήρα με σκοπό να τον εμφανίσουμε στη οθόνη LCD, «data» είναι το όνομα που θα δώσουμε στο χαρακτήρα. Ο κάθε χαρακτήρας αποτελείται από 8 byte, ένα byte για κάθε σειρά του χαρακτήρα. Κάθε σειρά απαρτίζεται από πέντε ψηφία, όταν θέλουμε να εμφανιστεί το ψηφίο του δίνουμε τη τιμή «1» αλλιώς τη τιμή «0». Για να μπορεί να χρησιμοποιηθεί ο χαρακτήρα μετά την εντολή byte πρέπει και να τον δηλώσουμε με την εντολή «lcd.createChar(num, data)»[23]

```
//BELAKIA
```

```
byte up[8] = {
  B00100,
  B01010,
  B10001,
  B11111,
  B00100,
  B00100,
  B00100,
};
```

```
byte down[8] = {
  B00100,
  B00100,
  B00100,
  B11111,
  B10001,
  B01010,
  B00100,
};
```

```
byte righth[8] = {
  B00100,
  B00110,
  B11101,
  B00110,
  B00100,
```

```
B00000,  
B00000,  
B00000,  
};
```

```
byte left[8] = {  
    B00100,  
    B01100,  
    B10111,  
    B01100,  
    B00100,  
    B00000,  
    B00000,  
    B00000,  
};
```

```
byte speaker[8] = {  
    B00001,  
    B00011,  
    B00111,  
    B11111,  
    B11111,  
    B00111,  
    B00011,  
    B00001,  
};
```

```
byte sun1[8] = {  
    B00100,  
    B10101,  
    B01110,  
    B10001,  
    B10101,  
    B10001,  
    B01110,  
};
```

```
B00000,  
};
```

```
byte sun2[8] = {  
  B00000,  
  B00000,  
  B00000,  
  B00000,  
  B10001,  
  B10001,  
  B01110,  
  B10101,  
};
```

```
byte epilogh[8] = {  
  B00000,  
  B00000,  
  B01110,  
  B11111,  
  B01110,  
  B00000,  
  B00000,  
};
```

- Ρουτίνα Setup εκτελείται μία φορά κατά την εκκίνηση του κώδικα ,και χρησιμοποιείται για την αρχικοποίηση και τον ορισμό των μεταβλητών.

```
void setup() {  
  Serial.begin(9600);  
  lcd.begin(16,2);
```

```
  //sensor
```

```

pinMode(A8, INPUT);
//button
pinMode(buttonUP, INPUT);
pinMode(buttonDOWN, INPUT);
pinMode(buttonRIGTH, INPUT);
pinMode(buttonLEFT, INPUT);
pinMode(buttonDelete, INPUT);
pinMode(buttonPlay, INPUT);

//LED
pinMode(ledUP, OUTPUT);
pinMode(ledDOWN, OUTPUT);
pinMode(ledRIGHT, OUTPUT);
pinMode(ledLEFT, OUTPUT);

//belakia
lcd.createChar(1, up);
lcd.createChar(2, down);
lcd.createChar(3, rigth);
lcd.createChar(4, left);
lcd.createChar(5, epilogh);
lcd.createChar(6, speaker);
lcd.createChar(7,sun1);
lcd.createChar(8,sun2);
}

```

- Η ρουτίνα Loop εκτελείται επανειλημμένα και εκεί γράφεται ο κύριος κώδικας.

```
void loop() {
```

- Η παρακάτω εντολή if κάνει κάθε φορά που πατάμε ένα από τα πλήκτρα κίνησης να ανάβει το αντίστοιχο led ,εκτός αν τρέχει κάποιο από τα προγράμματα ήδη.

```

if(memorystart != 1 && memorystart2 != 1 && memorystart3 != 1 && memorystart4 != 1){
digitalWrite(ledUP,buttonStateUP );

```

```
digitalWrite(ledDOWN,buttonStateDOWN );
digitalWrite(ledRIGHT,buttonStateRIGTH );
digitalWrite(ledLEFT,buttonStateLEFT );
};
```

- Στη συνέχεια καλείται η ρουτίνα menu που ανάλογα με το τι θα επιλεγεί εκεί από το χρήστη, θα εκτελεστεί και το αντίστοιχο πρόγραμμα.

```
//menu
if(menuout!=1){
  menu();
};
```

- Αν έχει επιλεγεί από τη ρουτίνα menu καλείται η ρουτίνα ODHGIES.

```
if(odhgies==1){
  ODHGIES();
}
```

- Αν έχει επιλεγεί από τη ρουτίνα menu ξεκινάει το πρόγραμμα 1.

```
if(program1==1){
```

- Καλείται η ρουτίνα Display η οποία εμφανίζει στην LCD τον πίνακα κινήσεων.

```
lcd.setCursor(0, 0);
lcd.write("Entoles:");
Display();
```

- Διαβάζεται το κουμπί Delete ,άμα πατηθεί καλείται η ρουτίνα Backspace.

```
buttonStateDelete = digitalWrite(buttonDelete);
if (buttonStateDelete == HIGH) {
```

```
Backspace();  
};
```

- Διαβάζεται το κουμπί Play ,άμα πατηθεί ξεκινάει το πρόγραμμα 1.

```
buttonStatePlay = digitalRead(buttonPlay);  
if (buttonStatePlay == HIGH) {  
    memorystart = 1;  
};  
if ( memorystart == 1 ) {  
    Start();  
};
```

- Σε όλη τη διάρκεια του προγράμματος 1 ,άμα δεν έχουν συμπληρωθεί 20 κινήσεις, καλείται η ρουτίνα pinakas.

```
if (i < 20) {  
    pinakas();  
};  
};
```

```
//=====
```

- Αν έχει επιλεγεί από τη ρουτίνα menu ξεκινάει το πρόγραμμα 2.

```
if(program2==1){
```

- Καλείται η ρουτίνα Display η οποία εμφανίζει στην LCD τον πίνακα κινήσεων.

```
lcd.setCursor(0, 0);  
lcd.write("Entoles:");  
Display();
```

- Διαβάζεται το κουμπί Delete ,άμα πατηθεί καλείται η ρουτίνα Backspace.

```
buttonStateDelete = digitalRead(buttonDelete);
```



```

if (buttonStateDelete == HIGH) {
  Backspace();
};

```

- Σε όλη τη διάρκεια του προγράμματος 2 ,άμα δεν έχουν συμπληρωθεί 20 κινήσεις, καλείται η ρουτίνα pinakas.

```

if (i < 20) {
  pinakas();
};

```

- Διαβάζεται το κουμπί Play ,άμα πατηθεί ξεκινάει το πρόγραμμα 2.

```

buttonStatePlay = digitalRead(buttonPlay);
if (buttonStatePlay == HIGH) {
  memorystart2 = 1;
};
if ( memorystart2 == 1) {
  Start2();
  sample();
}

```

- Κατά την εκκίνηση του προγράμματος 2 καλούνται οι ρουτίνες Start2() και sample(). Η ρουτίνα sample υπολογίζει το πλάτος του παλμού ενός ήχου και το μετατρέπει σε volt. Αν τα volt ξεπερνάνε τη τιμή 1σημαίνει ότι υπάρχει δυνατό ήχος. Τότε σταματάει η κίνηση που εκτελείται και ξεκινάει η επόμενη.[12]

```

if (volts >=1.0){
  j3=j3+1;
  stop1();
}
}
};

```

```
//=====
```

- Αν έχει επιλεγεί από τη ρουτίνα menu ξεκινάει το πρόγραμμα 3.

```
if(program3==1){
```

- Καλείται η ρουτίνα Display η οποία εμφανίζει στην LCD τον πίνακα κινήσεων.

```
lcd.setCursor(0, 0);
lcd.write("Entoles:");
Display();
```

- Διαβάζεται το κουμπί Delete ,άμα πατηθεί καλείται η ρουτίνα Backspace.

```
buttonStateDelete = digitalRead(buttonDelete);
if (buttonStateDelete == HIGH) {
Backspace();
};
```

- Σε όλη τη διάρκεια του προγράμματος 3 ,άμα δεν έχουν συμπληρωθεί 20 κινήσεις, καλείται η ρουτίνα pinakas.

```
if (i < 20) {
pinakas();
};
```

- Διαβάζεται το κουμπί Play ,άμα πατηθεί ξεκινάει το πρόγραμμα 3.Κατά την εκκίνηση του προγράμματος 3 διαβάζεται η αναλογική είσοδος A8,η πρώτη τιμή που διαβάζεται δηλώνεται ως αρχική τιμή «at» . Στη συνέχεια διαβάζεται επανειλημμένα η αναλογική είσοδος A8 και δηλώνεται ως καινούργια τιμή «kt».Όταν η αρχική τιμή είναι μικρότερη της καινούργιας καλείται η ρουτίνα Start2 κάθε κίνηση εκτελείται για χρόνο t=interval4.

```
buttonStatePlay = digitalRead(buttonPlay);
if (buttonStatePlay == HIGH) {
memorystart3 = 1;
};
```

```

if ( memorystart3 == 1){
  if(arxt==0 && i!=0){
    at1=analogRead(A8);
    arxt=1;
  }
  kt1=analogRead(A8);
  if(at1-kt1<=10){
    unsigned long currentMillis4 = millis();
    Start2();
    if (currentMillis4 - previousMillis4 >= interval4) {
      j3=j3+1;
      stop1();
      previousMillis4 = currentMillis4;
    }else{
      stop1();
    }
  }
}
};
//=====

```

- Αν έχει επιλεγεί από τη ρουτίνα menu ξεκινάει το πρόγραμμα 3.

```

if(program4==1){

```

- Καλείται η ρουτίνα Display η οποία εμφανίζει στην LCD τον πίνακα κινήσεων

```

lcd.setCursor(0, 0);
lcd.write("Entoles:");
Display();

```

- Διαβάζεται το κουμπί Delete ,άμα πατηθεί καλείται η ρουτίνα Backspace.

```

buttonStateDelete = digitalRead(buttonDelete);
if (buttonStateDelete == HIGH) {

```

```
Backspace();  
};
```

- Σε όλη τη διάρκεια του προγράμματος 3 ,άμα δεν έχουν συμπληρωθεί 20 κινήσεις, καλείται η ρουτίνα pinakas.

```
if (i < 20) {  
pinakas();  
};
```

- Διαβάζεται το κουμπί Play ,άμα πατηθεί ξεκινάει το πρόγραμμα 4.Κατά την εκκίνηση του προγράμματος 4 διαβάζεται η αναλογική είσοδος A8,η πρώτη τιμή που διαβάζεται δηλώνεται ως αρχική τιμή «at» . Στη συνέχεια διαβάζεται επανειλημμένα η αναλογική είσοδος A8 και δηλώνεται ως καινούργια τιμή «kt».Όταν η αρχική τιμή είναι μεγαλύτερη της καινούργιας καλείται η ρουτίνα Start2 κάθε κίνηση εκτελείται για χρόνο t=interval4.

```
buttonStatePlay = digitalRead(buttonPlay);  
if (buttonStatePlay == HIGH) {  
memorystart4 = 1;  
};  
if ( memorystart4 == 1){  
if(arxt==0 && i!=0){  
at1=analogRead(A8);  
arxt=1;  
}  
kt1=analogRead(A8);  
if(kt1-at1<=10){  
Start2();  
unsigned long currentMillis4 = millis();  
  
if (currentMillis4 - previousMillis4 >= interval4) {  
j3=j3+1;  
stop1();  
previousMillis4 = currentMillis4;
```

```

Serial.println("move");}
}else{
stop1();
Serial.println("stop");}
}
};
}

```

- Κατά τη ρουτίνα ODHGIES ο χρήστη πλοηγείτε μεταξύ των πινάκων Odhgies1, Odhgies2 Odhgies3 και Odhgies4 με τη χρήση των πλήκτρων κινήσεων .

```

void ODHGIES(){

unsigned long currentMillisButton = millis();

buttonStateUP = digitalRead(buttonUP);
buttonStateDOWN = digitalRead(buttonDOWN);
buttonStateRIGTH = digitalRead(buttonRIGTH);
buttonStateLEFT = digitalRead(buttonLEFT);

if (currentMillisButton - previousMillisButton >= intervalButton) {
if (buttonStateUP == HIGH && grammh!=0) {
grammh=grammh-1;
lcd.clear();
};
if (buttonStateDOWN == HIGH && grammh!=15) {
grammh=grammh+1;
lcd.clear();
};
if (buttonStateRIGTH == HIGH && t!=4) {
t=t+1;
grammh=0;
lcd.clear();
};
};
}

```

```

if (buttonStateLEFT == HIGH && t!=1 ) {
    t=t-1;
    grammh=0;
    lcd.clear();
};
previousMillisButton = currentMillisButton;
};

if(t==1){
lcd.setCursor(0, 0);
lcd.write(Odhgies1[grammh]);
    lcd.setCursor(0, 1);
lcd.write(Odhgies1[grammh+1]);
}
else if(t==2){
lcd.setCursor(0, 0);
lcd.write(Odhgies2[grammh]);
    lcd.setCursor(0, 1);
lcd.write(Odhgies2[grammh+1]);
}
else if(t==3){
lcd.setCursor(0, 0);
lcd.write(Odhgies3[grammh]);
    lcd.setCursor(0, 1);
lcd.write(Odhgies3[grammh+1]);
}
else if(t==4){
lcd.setCursor(0, 0);
lcd.write(Odhgies4[grammh]);
    lcd.setCursor(0, 1);
lcd.write(Odhgies4[grammh+1]);
}
}
}

```

- Κατά τη ρουτίνα pinakas διαβάζονται τα πλήκτρα κινήσεων και αποθηκεύονται στο πίνακα kinhseis[i] .Κάθε φορά που πατιέται ένα πλήκτρο παράγεται μία νότα.

```
void pinakas() {
```

```
    unsigned long currentMillisButton = millis();
```

```
    buttonStateUP = digitalRead(buttonUP);
```

```
    buttonStateDOWN = digitalRead(buttonDOWN);
```

```
    buttonStateRIGTH = digitalRead(buttonRIGTH);
```

```
    buttonStateLEFT = digitalRead(buttonLEFT);
```

```
    if (currentMillisButton - previousMillisButton >= intervalButton) {
```

```
        if (buttonStateUP == HIGH) {
```

```
            kinhseis[i] = 1;
```

```
            i = i + 1;
```

```
            tone(7, 740,200);
```

```
        };
```

```
        if (buttonStateDOWN == HIGH) {
```

```
            kinhseis[i] = 2;
```

```
            i = i + 1;
```

```
            tone(7,698,200);
```

```
        };
```

```
        if (buttonStateRIGTH == HIGH) {
```

```
            kinhseis[i] = 3;
```

```
            i = i + 1;
```

```
            tone(7,466,200);
```

```
        };
```

```
        if (buttonStateLEFT == HIGH) {
```

```
            kinhseis[i] = 4;
```

```
            i = i + 1;
```

```
            tone(7,494,200);
```

```
        };
```

```
    previousMillisButton = currentMillisButton;
};
};
```

- Κατά τη ρουτίνα Display ,οι κινήσεις που έχουν αποθηκευτεί στο πίνακα kinhseis[i], εμφανίζονται στην οθόνη LCD με μορφή συμβόλων-χαρακτήρων.

```
void Display() {

    lcd.setCursor(0, 1);
    for (int j = 0; j <= i ; j++) {
        if (kinhseis[j] == 1) {
            lcd.write(1);
        } if (kinhseis[j] == 2) {
            lcd.write(2);
        };
        if (kinhseis[j] == 3) {
            lcd.write(3);
        };
        if (kinhseis[j] == 4) {
            lcd.write(4);
        };
        if (kinhseis[j] == 0) {
            lcd.write(" ");
        };
    };
};
```

- Κατά τη ρουτίνα Backspace ,διαγράφεται μία κίνηση από το πίνακα και kinhseis[i].

```
void Backspace() {
    unsigned long currentMillisDelete = millis();
```



```

if (currentMillisDelete - previousMillisDelete >= intervalDelete) {
    i = i - 1;
    kinhseis[i] = 0;
    previousMillisDelete = currentMillisDelete;
};
};

```

- Κατά τη ρουτίνα Start, διαβάζονται οι κινήσεις που έχουν αποθηκευτεί στο πίνακα kinhseis[i], και ανάλογα καλούνται οι ρουτίνες mprosta, pisw, deksia, aristera και stop1.

```

void Start() {

```

```

    for (int j2 = 0; j2 <= i; j2++) {

```

```

        buttonStateDelete = digitalRead(buttonDelete);

```

```

        if (buttonStateDelete == HIGH ) {

```

```

            j2=i;

```

```

        };

```

```

        if (kinhseis[j2] == 1) {

```

```

            mprosta();

```

```

        };

```

```

        if (kinhseis[j2] == 2) {

```

```

            pisw();

```

```

        };

```

```

        if (kinhseis[j2] == 3) {

```

```

            deksia();

```

```

        };

```

```

        if (kinhseis[j2] == 4) {

```

```

            aristera();

```

```

        };

```

```

        if (kinhseis[j2] == 0) {

```

```

            stop1();

```

```

        };

```

```

    };

```

```
memorystart = 0;  
};
```

- Κατά τη ρουτίνα mprosta, δίνεται εντολή στους κινητήρες να κινηθούν μπροστά και να ανάψει το αντίστοιχο led για 1sec καθώς και να παραχθεί ήχος για 0,5sec.

```
void mprosta() {  
  Serial.println("Mprosta");  
  digitalWrite(34,LOW );  
  digitalWrite(35,HIGH );  
  digitalWrite(32,LOW );  
  digitalWrite(33,HIGH );  
  digitalWrite(ledUP, HIGH);  
  tone(7, 740,500);  
  delay(1000);  
  
  digitalWrite(ledUP,LOW );  
  digitalWrite(34,LOW );  
  digitalWrite(35,LOW );  
  digitalWrite(32,LOW );  
  digitalWrite(33,LOW );  
};
```

- Κατά τη ρουτίνα pisw, δίνεται εντολή στους κινητήρες να κινηθούν πίσω και να ανάψει το αντίστοιχο led για 1sec καθώς και να παραχθεί ήχος για 0,5sec.

```
void pisw() {  
  Serial.println("Piso");  
  digitalWrite(34,HIGH );  
  digitalWrite(35,LOW );  
  digitalWrite(32,HIGH );  
  digitalWrite(33,LOW );
```

```
digitalWrite(ledDOWN,HIGH );  
tone(7,698,500);  
delay(1000);
```

```
digitalWrite(ledDOWN,LOW );  
digitalWrite(34,LOW );  
digitalWrite(35,LOW );  
digitalWrite(32,LOW );  
digitalWrite(33,LOW );  
};
```

- Κατά τη ρουτίνα deksia, δίνεται εντολή στους κινητήρες να στρίψουν δεξιά και να ανάψει το αντίστοιχο led για 1sec καθώς και να παραχθεί ήχος για 0,5sec.

```
void deksia() {  
Serial.println("Deksia");  
digitalWrite(34,LOW );  
digitalWrite(35,HIGH );  
digitalWrite(32,HIGH);  
digitalWrite(33,LOW);  
digitalWrite(ledRIGHT,HIGH );  
tone(7,466,500);  
delay(450);
```

```
digitalWrite(ledRIGHT,LOW );  
digitalWrite(34,LOW );  
digitalWrite(35,LOW );  
digitalWrite(32,LOW );  
digitalWrite(33,LOW );  
};
```

- Κατά τη ρουτίνα aristera, δίνεται εντολή στους κινητήρες να στρίψουν αριστερά και να ανάψει το αντίστοιχο led για 1sec καθώς και να παραχθεί ήχος για 0,5sec.

```

void aristera() {
  Serial.println("Aristera");
  digitalWrite(34,HIGH );
  digitalWrite(35,LOW );
  digitalWrite(32,LOW);
  digitalWrite(33,HIGH);
  digitalWrite(ledLEFT,HIGH );
  tone(7,494,500);
  delay(450);

  digitalWrite(ledLEFT,LOW );
  digitalWrite(34,LOW );
  digitalWrite(35,LOW );
  digitalWrite(32,LOW );
  digitalWrite(33,LOW );
};

```

- Κατά τη ρουτίνα stop1 σταματάει κάθε κίνηση και σβήνουν όλα τα led.

```

void stop1() {
  Serial.println("Stop");
  digitalWrite(34,LOW );
  digitalWrite(35,LOW );
  digitalWrite(32,LOW );
  digitalWrite(33,LOW );
  delay(10);
  digitalWrite(ledRIGHT,LOW );
  digitalWrite(ledLEFT,LOW );
  digitalWrite(ledUP,LOW );
  digitalWrite(ledDOWN,LOW );
};

```

- Κατά τη ρουτίνα Start2, διαβάζονται οι κινήσεις που έχουν αποθηκευτεί στο πίνακα kinhseis[i], και ανάλογα καλούνται οι ρουτίνες mprosta2, pisw2, deksia2, aristera2 και stop1.

```

void Start2() {
  if (j3<i) {

    buttonStateDelete = digitalRead(buttonDelete);
    if (buttonStateDelete == HIGH ) {
      j3=i;
    };
    if (kinhseis[j3] == 1) {
      mprosta2();
    };
    if (kinhseis[j3] == 2) {
      pisw2();
    };
    if (kinhseis[j3] == 3) {
      deksia2();
    };
    if (kinhseis[j3] == 4) {
      aristera2();
    };
    if (kinhseis[j3] == 0) {
      stop1();
    };
    }else{
      memorystart2 = 0;
      memorystart3 =0;
      memorystart4 = 0;

      j3=0;
    }
  };
}

```

```

void sample(){
  start= millis(); // Start of sample window
  peakToPeak = 0; // peak-to-peak level
  signalMax = 0;
  signalMin = 1024;
  // collect data for 250 miliseconds
  //apothikeush tou max kai tou min se xrono 250 milisecond
  while (millis() - start < sampleWindow)
  {  knock = analogRead(A4);
    if (knock < 1024) //This is the max of the 10-bit ADC so this loop will include all
readings
    {
      if (knock > signalMax)
      {
        signalMax = knock; // save just the max levels
      }
      else if (knock < signalMin)
      {
        signalMin = knock; // save just the min levels
      }
    }
  }
  peakToPeak = signalMax - signalMin; // max - min = peak-peak amplitude /υπολογισμος του
platous tou palmou
  volts = (peakToPeak * 3.3) / 1024; // convert to volts

  };

```

- Κατά τη ρουτίνα mprosta2, δίνεται εντολή στους κινητήρες να κινηθούν προς τα μπροστά και να ανάψει το αντίστοιχο led.

```

void mprosta2() {
  Serial.println("Mprosta");

```

```
digitalWrite(34,LOW );
digitalWrite(35,HIGH );
digitalWrite(32,LOW );
digitalWrite(33,HIGH );
digitalWrite(ledUP, HIGH);
};
```

- Κατά τη ρουτίνα pisc2, δίνεται εντολή στους κινητήρες να κινηθούν προς τα πίσω και να ανάψει το αντίστοιχο led.

```
void pisc2() {
  Serial.println("Piso");
  digitalWrite(34,HIGH );
  digitalWrite(35,LOW );
  digitalWrite(32,HIGH );
  digitalWrite(33,LOW );
  digitalWrite(ledDOWN,HIGH );
};
```

- Κατά τη ρουτίνα deksia2, δίνεται εντολή στους κινητήρες να στρίψουν δεξιά και να ανάψει το αντίστοιχο led.

```
void deksia2() {
  Serial.println("Deksia");
  digitalWrite(34,LOW );
  digitalWrite(35,HIGH );
  digitalWrite(32,HIGH);
  digitalWrite(33,LOW);
  digitalWrite(ledRIGHT,HIGH );
};
```

- Κατά τη ρουτίνα aristera2, δίνεται εντολή στους κινητήρες να στρίψουν αριστερά και να ανάψει το αντίστοιχο led.

```

void aristera2() {
  Serial.println("Aristera");
  digitalWrite(34,HIGH );
  digitalWrite(35,LOW );
  digitalWrite(32,LOW);
  digitalWrite(33,HIGH);
  digitalWrite(ledLEFT,HIGH );
};

```

- Η ρουτίνα menu διαβάζει τα κουμπιά κινήσεως και ανάλογα με την επιλογή που έχει γίνει δείχνει το τίτλο του κάθε προγράμματος μαζί με σύμβολα που το περιγράφουν. Όταν επιλεγεί το κουμπί Play επιλέγεται το πρόγραμμα που εμφανίζεται εκείνη την ώρα στην οθόνη.

```

void menu(){

  buttonStateUP = digitalRead(buttonUP);
  buttonStateDOWN = digitalRead(buttonDOWN);
  delay(150);

  if (buttonStateUP == HIGH && menu0==1) {
    menuState=0;
  };
  if (buttonStateDOWN == HIGH && menu1==0) {
    menuState=1;
  };
  if (buttonStateUP == HIGH && menu1==1) {
    menuState=0;
  };
  if (buttonStateDOWN == HIGH && menu1==1) {
    menuState=2;
  };
  if (buttonStateUP == HIGH && menu2==1) {

```



```

    menuState=1;
};
if (buttonStateDOWN == HIGH && menu2==1) {
    menuState=3;
};
    if (buttonStateUP == HIGH && menu3==1) {
        menuState=2;
    };
if (buttonStateDOWN == HIGH && menu3==1) {
    menuState=4;
};
if (buttonStateUP == HIGH && menu4==1) {
    menuState=3;
};
if (buttonStateDOWN == HIGH && menu4==1) {
    menuState=4;
};

////////
if (menuState ==0) {
    menu0=1;
    menu1=0;
    menu2=0;
    menu3=0;
    menu4=0;
}

if (menuState ==1) {
    menu0=0;
    menu1=1;
    menu2=0;
    menu3=0;
    menu4=0;
}

```

```
if (menuState == 2) {  
    menu0=0;  
    menu1=0;  
    menu2=1;  
    menu3=0;  
    menu4=0;  
}
```

```
if (menuState == 3) {  
    lcd.clear();  
    menu0=0;  
    menu1=0;  
    menu2=0;  
    menu3=1;  
    menu4=0;  
}
```

```
if (menuState == 4) {  
    lcd.clear();  
    menu0=0;  
    menu1=0;  
    menu2=0;  
    menu3=0;  
    menu4=1;  
}
```

```
//==Grafika
```

```
unsigned long currentMillismenu = millis();
```

```
if (currentMillismenu - previousMillismenu >= intervalmenu) {  
    previousMillismenu = currentMillismenu;  
    if (graphState == LOW) {  
        lcd.clear();  
        graphState = HIGH;  
    } else {
```

```

    lcd.clear();
    graphState = LOW;
}
}

if (menu0==1){
if(graphState==LOW){

lcd.setCursor(1, 0);
lcd.write("Odhgies Xrhshs");
    lcd.setCursor(5, 1);
    lcd.write("(");
    lcd.write(7);
    lcd.write(".");
    lcd.write(7);
    lcd.write(")");
}else if(graphState==HIGH){
lcd.setCursor(1, 0);
lcd.write("Odhgies Xrhshs");
    lcd.setCursor(5, 1);
    lcd.write("(");
    lcd.write(7);
    lcd.write(".");
    lcd.write(8);
    lcd.write(")");
}
}

if (menu1==1){
if(graphState==LOW){
    lcd.setCursor(2, 0);
    lcd.write("Aplh kinsh");
    lcd.setCursor(0, 1);
    lcd.write(4);

```

```
lcd.write(" ");
lcd.write(1);
lcd.write(" ");
lcd.write(2);
lcd.write(" ");
lcd.write(3);
lcd.write(" ");
lcd.write(4);
lcd.write(" ");
lcd.write(1);
lcd.write(" ");
lcd.write(2);
lcd.write(" ");
lcd.write(3);
}else if(graphState==HIGH){
lcd.setCursor(2, 0);
lcd.write("Aplh kinsh");
lcd.setCursor(0, 1);
lcd.write(3);
lcd.write(" ");
lcd.write(2);
lcd.write(" ");
lcd.write(1);
lcd.write(" ");
lcd.write(4);
lcd.write(" ");
lcd.write(3);
lcd.write(" ");
lcd.write(2);
lcd.write(" ");
lcd.write(1);
lcd.write(" ");
lcd.write(4);
}
```

```

}
if (menu2==1){
  if(graphState==LOW){
    lcd.setCursor(1, 0);
    lcd.write("Kinhsh me Hxo");
    lcd.setCursor(4, 1);
    lcd.write(6);
    lcd.write(" ) )");
  }else if(graphState==HIGH){
    lcd.setCursor(1, 0);
    lcd.write("Kinhsh me Hxo");
    lcd.setCursor(4, 1);
    lcd.write(6);
    lcd.write(" ) ) ");
  };
};

```

```

if (menu3==1){
  if(graphState==LOW){
    lcd.setCursor(1, 0);
    lcd.write("Kinhsh sto fws");
    lcd.setCursor(3, 1);
    lcd.write(1);
    lcd.write(1);
    lcd.setCursor(5, 1);
    lcd.write("(");
    lcd.write(7);
    lcd.write(".");
    lcd.write(7);
    lcd.write(")");
  }else if(graphState==HIGH){
    lcd.setCursor(1, 0);
    lcd.write("Kinhsh sto fws");

```

```

lcd.setCursor(5, 1);
lcd.write("(");
lcd.write(8);
lcd.write(".");
lcd.write(8);
lcd.write(")");
}
};

if (menu4==1){
if(graphState==LOW){
lcd.setCursor(0, 0);
lcd.write("Kinhsh sto skotadi");
lcd.setCursor(5, 1);
lcd.write("(");
lcd.write(7);
lcd.write(".");
lcd.write(7);
lcd.write(")");
}else if(graphState==HIGH){
lcd.setCursor(0, 0);
lcd.write("nhsh sto skotadi");
lcd.setCursor(3, 1);
lcd.write(2);
lcd.write(2);
lcd.setCursor(5, 1);
lcd.write("(");
lcd.write(8);
lcd.write(".");
lcd.write(8);
lcd.write(")");
}
};

```

```
//====PLAY
buttonStatePlay = digitalRead(buttonPlay);
if (buttonStatePlay == HIGH) {
    delay(1000);
    if (menu0==1){
        menuout=1;
        odhgies=1;
        lcd.clear();
    };
    if (menu1==1){
        menuout=1;
        program1=1;
        lcd.clear();
    };
    if (menu2==1){
        menuout=1;
        program2=1;
        lcd.clear();
    };
    if (menu3==1){
        menuout=1;
        program3=1;
        lcd.clear();
    };
    if (menu4==1){
        menuout=1;
        program4=1;
        lcd.clear();
    };
};
};
}
```

### 3.5 Σχόλια - Μεθοδολογία Έρευνας

Το λογισμικό του Arduino έχει δύο βασικές ρουτίνες την void Setup και την void Loop. Το void μπροστά από τις ρουτίνες σημαίνει ότι οι ρουτίνες δεν επιστρέφουν κάποια τιμή, η Setup εκτελείται μία φορά κατά την εκκίνηση, και χρησιμοποιείται για την αρχικοποίηση και τον ορισμό των μεταβλητών. Αντίθετα η Loop εκτελείται επανειλημμένα και εκεί είναι που γράφεται ο κύριος κώδικας. Το Arduino είναι ένας απλός επεξεργαστής και μπορεί να τρέξει ένα πρόγραμμα κώδικα τη φορά, σε αντίθεση με άλλους πιο ισχυρούς επεξεργαστές. Για αυτό πρέπει να έχουμε μία πιο προσεκτική προσέγγιση όταν θέλουμε να επιτύχουμε πολυπλεξία.[24]

Όπως αναφέρθηκε και στην αρχή του προηγούμενου κεφαλαίου, το κάθε πρόγραμμα υλοποιήθηκε ξεχωριστά και στη συνέχεια ενοποιήθηκαν όλα μαζί ,παρατηρήθηκε όμως ότι τα προγράμματα επηρέαζαν το ένα το άλλο. Χρησιμοποιήθηκαν διάφορες τεχνικές για να αποφευχθούν τέτοια προβλήματα. Βασική προδιαγραφή ήταν η χρήση πολλών ρουτίνων, με αυτό το τρόπο η ρουτίνα loop έχει λιγότερες γραμμές κώδικα, οι εντολές έχουν ομαδοποιηθεί και είναι πιο συγκεκριμένος ο χρόνος που καλείται μία ρουτίνα.

Κάτι άλλο που δημιούργησε πρόβλημα κατά την ένωση των προγραμμάτων ήταν η εντολή delay. Το delay όταν καλείται κάνει παύση κάθε διεργασίας, «παγώνει» το Arduino στην κατάσταση που βρίσκεται εκείνη τη στιγμή. Για να αντιμετωπιστεί αυτό έγινε χρήση ενός συνόλου εντολών και κυρίως της εντολής millis().

Καθώς η πλοήγηση μεταξύ των προγραμμάτων γίνεται με ένα ενιαίο μενού έγινε χρήση της υποδοχής Reset ,για επιστροφή στην αρχική οθόνη. Με τη χρήση της συγκεκριμένης υποδοχής και όχι με τη χρήση κώδικα επιβεβαιώνουμε ότι όλες η μεταβλητές έχουν επιστρέψει στην αρχική τους τιμή.

Τέλος επειδή το Arduino τροφοδοτεί με ρεύμα μέχρι 5 volt ,ενώ ένα σύστημα πολυπλεξίας χρειάζεται πολύ περισσότερη τάση ,έπρεπε να βρεθεί μία λύση για την τροφοδοσία της κατασκευής. Το βασικότερο πρόβλημα στη τροφοδοσία το είχαν οι κινητήρες , οι οποίοι για να λειτουργήσουν με ικανοποιητική ταχύτητα χρειάζονται μεγαλύτερη τάση από αυτή που μπορεί να παρέχει το Arduino. Πρόβλημα παρουσίασαν και τα υπόλοιπα εξαρτήματα, όπως τα button που είχαν μεγάλο χρόνο απόκρισης. Για το λόγο αυτό έγινε μια διεργασία στη ένωση μπαταρίας με τον διακόπτη ώστε από τα 12 volt της μπαταρίας τα 5 να πηγαίνουν στο Arduino ,ενώ τα περισσευούμενα 7 volt να κατευθύνονται στους κινητήρες .



## 4 Βιβλιογραφία

1. <https://grobotronics.com/shadow-chassis.html>
2. <https://el.wikipedia.org/wiki/Arduino>
3. [https://en.wikipedia.org/wiki/Liquid-crystal\\_display](https://en.wikipedia.org/wiki/Liquid-crystal_display)
4. Electricity 4: AC/DC Motors, Controls, and Maintenance Από τον/την Jeffrey J. Keljik
5. ΗΛΕΚΤΡΟΝΙΚΗ : Αρχές και Εφαρμογές 7η Έκδοση από Albert Malvino και David j.Bates , Εκδόσεις Τζιόλα
6. [https://el.wikipedia.org/wiki/Δίοδος\\_εκπομπής\\_φωτός](https://el.wikipedia.org/wiki/Δίοδος_εκπομπής_φωτός)
7. <http://www.instructables.com/id/Using-an-LDR-Sensor-With-Arduino/>
8. <https://learn.sparkfun.com/tutorials/electret-mic-breakout-board-hookup-guide>
9. <http://www.instructables.com/id/L293D-Motor-Driver/>
10. <https://en.wikipedia.org/wiki/Potentiometer#Television>
11. <https://learn.sparkfun.com/tutorials/resistors>
12. <https://programmingelectronics.com/an-easy-way-to-make-noise-with-arduino-using-tone/>
13. <https://en.wikipedia.org/wiki/Switch>
14. <https://en.wikipedia.org/wiki/Breadboard>
15. <https://learn.sparkfun.com/tutorials/electret-mic-breakout-board-hookup-guide>
16. <https://grobotronics.com/basic-16x2-character-lcd-white-on-blue-5v.html>
17. <https://grobotronics.com/potentiometer-10k-plastic.html>
18. <http://greek.uchidg.com/sale-2038620-high-resistance-e24-510-ohm-carbon-film-1-watt-resistor-non-inductive-resistor.html>
19. <http://learning.grobotronics.com/2014/09/>
20. <http://www.cs.ucy.ac.cy/~annap/ep1664/notes/lecture02-03.pdf>
21. [http://iep.edu.gr:8080/images/school\\_books/24-0530\\_Mixatroniki\\_G-EPAL\\_BM.pdf](http://iep.edu.gr:8080/images/school_books/24-0530_Mixatroniki_G-EPAL_BM.pdf)
22. <https://el.wikipedia.org/wiki/Αισθητήρας>
23. <https://www.arduino.cc/en/Reference/LiquidCrystalCreateChar>
24. <http://www.instructables.com/id/Multi-task-Your-Arduino/>
25. <http://www.hobbytronics.co.uk/led-tactile-button-green>
26. <http://www.faranux.com/product/push-button-switch/>
27. <https://grobotronics.com/led-5mm-950nm-100mw.html>
28. <https://grobotronics.com/photo-resistor-ldr-5mm.html>

29. <https://grobotronics.com/electret-microphone-amplifier-max4466-with-adjustable-gain.html>
30. <https://grobotronics.com/rocker-switch-on-off-red.html>
31. ΣΥΓΧΡΟΝΑ ΣΥΣΤΗΜΑΤΑ ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ 11<sup>η</sup> Έκδοση RICHARD C.DORF -ROBERT H. BISHOP
32. <http://www.ti.com/lit/ds/symlink/l293.pdf>