



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΤΟΜΕΑΣ ΜΗΧΑΝΟΛΟΓΙΚΩΝ ΚΑΤΑΣΚΕΥΩΝ ΚΑΙ ΑΥΤΟΜΑΤΟΥ

ΕΛΕΓΧΟΥ

**Διπλωματική Εργασία: « Μελέτη Και Κατασκευή
Αυτόματου Συστήματος Ιστοκινέτας Μικρών Δειγμάτων »**

Μιχαλοπούλου Μαρία

Επιβλέπων Καθηγητής: Λεωνίδας Αλεξόπουλος

Αθήνα, Ιανουάριος 2014



Περίληψη

Η παρούσα διπλωματική εργασία αφορά στην κατασκευή μιας αυτόματης ιστοκινέτας η οποία θα λαμβάνει μέρος στις ιστολογικές εφαρμογές του εργαστηρίου για μικρά δείγματα.

Πιο συγκεκριμένα, η εργασία περιλαμβάνει τον σχεδιασμό της συσκευής ξεκινώντας από τον ιδεασμό και φτάνοντας στο τρισδιάστατο συναρμολογημένο μοντέλο. Από το τρισδιάστατο μοντέλο έγιναν τα δισδιάστατα σχέδια από τα οποία μπόρεσαν να πραγματοποιηθούν οι κατάλληλες κατεργασίες για την κατασκευή. Κάποιες κατεργασίες έλαβαν μέρος στο εργαστήριο Εμβιομηχανικής και Συστημικής Βιολογίας του τμήματος Μηχανολόγων Μηχανικών του Εθνικού Μετσόβιου Πολυτεχνείου και κάποιες άλλες σε εξωτερικά μηχανουργεία. Στη συνέχεια έγινε η συναρμολόγηση και ο έλεγχος των ηλεκτρονικών διατάξεων μέσω του προγραμματισμού ενός μικροεπεξεργαστή Arduino. Τέλος, πραγματοποιήθηκε η κόλληση των καλωδίων, των ηλεκτρονικών στοιχείων στην πλακέτα αλλά και η τελική τοποθέτηση όλων πάνω στη συσκευή.

Όταν ολοκληρώθηκε η συσκευή και ήταν έτοιμη για λειτουργία πραγματοποιήθηκε η πρώτη εφαρμογή ενός δείγματος χόνδρου για να αποδειχθεί ότι τα ιστολογικά αποτελέσματα με την χειροκίνητη και την μηχανοκίνητη λειτουργία ήταν ίδια. Τα αποτελέσματα έχουν να κάνουν με τον βαθμό οστεοαρθρίτιδας και κατά πόσον ο βαθμός αυτός φαίνεται να παραμένει ίδιος και στις δύο περιπτώσεις.





Abstract

This diploma thesis concerns the construction of an automatic histokinette system which will take part in laboratory's histological applications for small samples.

More specifically, the thesis involves the design of the device from ideation to the creation of an assembled three-dimensional model. The appropriate manufacturing processes for the construction were executed with the use of two-dimensional drawings which were designed according to the three-dimensional model. Some of the processes took place in the mechanical workshop of Bioengineering and Systems Biology Lab of the Department of Mechanical Engineering of the National Technical University of Athens while others in external mechanical workshops. In cessation, the required assembly was conducted as well as the required control for the electronic devices which was achieved through the programming of a microprocessor of type Arduino. At the end, the gluing of the wires and the electronic components on the circuit board was performed and all components were finally placed on the device.

When the device was completed and ready for operation, the first application of a cartilage sample was held in order to verify and prove that the histological results were the same either with manual or the motorized operation. The results were related to the degree of osteoarthritis and whether the degree appears to be the same in both cases.





Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου, κ. Λεωνίδα Αλεξόπουλο για τη στήριξη του όλους αυτούς τους μήνες σε ό,τι χρειαζόταν για την εργασία και την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου την συγκεκριμένη διπλωματική εργασία. Ήταν μια ευκαιρία για εμένα να δω στην πράξη πως, από μία ιδέα και μια επιθυμητή λειτουργία, μπορούμε να φτάσουμε σε μια έτοιμη συσκευή που συμβάλει στην καλή καθημερινή ροή του εργαστηρίου και διευκολύνει τους χρήστες αποβάλλοντάς τους επιπρόσθετες εργασίες που έπρεπε να κάνουν.

Όπως αναφέρεται και στην Εισαγωγή στη συνέχεια, το πρώτο μέρος της εργασίας ξεκίνησε στα πλαίσια του μαθήματος Εμβιομηχανικής και Βιοϊατρικής Τεχνολογίας . Για την εργασία αυτή είχα την χαρά να συνεργάζομαι με τον συμφοιτητή μου Νικόλαο Καψαμπέλη τοποθετώντας τα θεμέλια για την παρούσα διπλωματική εργασία. Εκ μέρους και των δυο μας, θα ήθελα να πω ένα μεγάλο ευχαριστώ στον διδακτορικό φοιτητή του εργαστηρίου, Γεώργιο Κανακάρη για την πολύτιμη βοήθεια και καθοδήγηση σε ένα αρχικά σχεδόν άγνωστο μονοπάτι για εμάς, εκείνο του σχεδιασμού για κατασκευή και βιομηχανική παραγωγή, της έρευνας αλλά και της κατασκευής σε μετέπειτα στάδιο.

Στην συνέχεια, θα ήθελα να ευχαριστήσω ιδιαίτερος τον Νικόλαο Γεωργίου για την πολύτιμη βοήθεια του σε θέματα που αφορούσαν τον προγραμματισμό και κυρίως τον ηλεκτρονικό σχεδιασμό του συστήματος και φυσικά για την υπομονή που έδειξε σε τυχόν απορίες που είχα, αφού ήταν η πρώτη φορά που ασχολήθηκα με τέτοιου είδους θέματα σε πρακτικό επίπεδο. Επιπλέον, δεν θα μπορούσα να παραλείψω τις συμβουλές του και την βοήθειά του και σε κατασκευαστικά ζητήματα όταν χρειάστηκαν.

Επίσης, θα ήθελα να ευχαριστήσω την διπλωματούχο Μηχανολόγο Μηχανικό Ελισάβετ Χατζοπούλου, καθώς και την Βιολόγο Θεώνη Κατωπόδη, που με βοήθησαν να εξασκηθώ στις πειραματικές διαδικασίες και με μύησαν γενικότερα στο χώρο των βιολογικών και ειδικότερα των ιστολογικών εφαρμογών.

Τέλος, δεν θα μπορούσα να παραλείψω να πω ένα πολύ μεγάλο ευχαριστώ σε όλα τα παιδιά του εργαστηρίου αλλά και στην οικογένειά μου η οποία τόσα χρόνια ήταν δίπλα μου και με στήριζε σε κάθε μου βήμα και σε κάθε μου δυσκολία.





Περιεχόμενα

Περίληψη.....	iii
Abstract.....	v
Ευχαριστίες.....	vii
Περιεχόμενα	1
1. Εισαγωγή.....	3
2. Θεωρητικό Υπόβαθρο.....	5
2.1. Ιστολογία	5
2.2. Οστεοαρθρίτιδα.....	5
2.3. Τυπική Διαδικασία Εφαρμογών.....	9
2.4. Διαβάθμιση Οστεοαρθρίτιδας.....	13
3. Τρέχουσες Πληροφορίες και Ανάγκες που προκύπτουν	19
3.1. Σκοπός της Συσκευής.....	19
3.2. Υπάρχουσες Τεχνολογίες.....	19
3.3. Ανάγκες.....	23
4. Προδιαγραφές Συστήματος	25
5. Περιγραφή και Σχεδιασμός Μηχανολογικής Διάταξης	27
5.1. Σύστημα Περιστροφής Δοχείων.....	28
5.2. Σύστημα Γραμμικής Κίνησης.....	31
5.3. Σύστημα Περιστροφής Βραχίονα.....	32
5.4. Σύστημα Θερμαινόμενων Δοχείων.....	34
5.5. Σύστημα Προφύλαξης Διαλυμάτων	34
5.6. Συστήματα Ελέγχου	35
6. Κατασκευή	37
6.1. Κομμάτια και Κατεργασίες Δύο Διαστάσεων.....	37



6.2. Κομμάτια και Κατεργασίες Τριών Διαστάσεων.....	45
6.3. Συναρμολόγηση.....	50
7. Περιγραφή και Σχεδιασμός Ηλεκτρονικών Διατάξεων.....	55
7.1. Έλεγχος Κινητήρων Συνεχούς Ρεύματος με Driver L293D.....	56
7.2. Έλεγχος Βηματικού Κινητήρα με Easy Driver.....	57
7.3. Έλεγχος Θέσης με Μηχανικό Κωδικοποιητή (encoder).....	58
7.4. Έλεγχος Λειτουργίας Θερμαντήρων (heaters).....	60
7.5. Λειτουργία Αισθητήρων Θερμοκρασίας.....	61
7.6. Έλεγχος Θέσης με Φωτοευαίσθητους Μικροαισθητήρες.....	62
7.7. Έλεγχος Θέσης με Μικροδιακόπτες.....	64
7.8. Λειτουργία Οθόνης Υγρών Κρυστάλλων (LCD).....	65
8. Προγραμματισμός και Περιγραφή του Κώδικα.....	67
9. Εφαρμογές που Πραγματοποιήθηκαν.....	69
9.1. Εφαρμογή με Χειροκίνητη Διαδικασία.....	69
9.2. Εφαρμογή με το Σύστημα Ιστοκινέτας.....	71
9.3. Αποτελέσματα και Σύγκριση των Δύο Εφαρμογών.....	73
10. Τελικά Συμπεράσματα.....	79
11. Βιβλιογραφία.....	81
Παράρτημα Α': Οδηγίες Χρήσης Ιστοκινέτας.....	83
Παράρτημα Β': Πρωτόκολλο Ιστοκινέτας.....	85
Παράρτημα Γ': Ιστολογικό Πρωτόκολλο.....	87
Παράρτημα Δ': Κατασκευαστικά Σχέδια.....	91
Παράρτημα Ε': Πίνακας Υλικών.....	93
Παράρτημα ΣΤ': Κώδικας σε Περιβάλλον Arduino.....	95
Παράρτημα Ζ': Έτοιμες Βιβλιοθήκες που Χρησιμοποιήθηκαν στον Κώδικα.....	113
Παράρτημα Η': Τελικό Διάγραμμα Ηλεκτρονικού Κυκλώματος.....	129
Παράρτημα Θ': Datasheets Ηλεκτρονικών Εξαρτημάτων (Τεχνικά Δελτία).....	131



1. Εισαγωγή

Η παρούσα διπλωματική εργασία εκπονήθηκε στο εργαστήριο Εμβιομηχανικής και Βιοϊατρικής Τεχνολογίας της σχολής Μηχανολόγων Μηχανικών του Εθνικού Μετσοβίου Πολυτεχνείου.

Η εργασία αφορά τη μελέτη και την κατασκευή μιας εργαστηριακής συσκευής που λαμβάνει χώρα σε ιστολογικές πειραματικές διαδικασίες. Η συσκευή αυτή ονομάζεται ιστοκινέτα και είναι υπεύθυνη για την επεξεργασία του ιστού για να γίνει ιστολογική εξέταση. Η επεξεργασία αυτή όταν γινόταν χειροκίνητα απαιτούσε από τον πειραματιστή την διαρκή προσοχή του για περίπου 7,5 ώρες ενώ με την ιστοκινέτα γίνεται πλέον αυτόματα, χωρίς την απαραίτητη συνεχή επίβλεψη καθόλη αυτή τη διάρκεια.

Στην εργασία αυτή, αρχικά, θα αναλυθεί το απαραίτητο βιολογικό υπόβαθρο όσον αφορά στην ιστολογία, τις εφαρμογές της αλλά και τη διαδικασία, θα περιγραφεί ο σχεδιασμός των μηχανολογικών και των ηλεκτρονικών διατάξεων, ο τρόπος κατασκευής, ο προγραμματισμός, αλλά και οι ιστολογικές εφαρμογές που έγιναν στο εργαστήριο ώστε να αποδειχθεί ότι η συσκευή δουλεύει σωστά.

Σε αυτό το σημείο θα πρέπει να αναφερθεί ότι το πρώτο τμήμα της διπλωματικής ήταν το σχεδιαστικό μέρος, το οποίο έγινε στα πλαίσια του μαθήματος Εμβιομηχανικής και Βιοϊατρικής Τεχνολογίας, μάθημα του 9^{ου} εξαμήνου, με διδάσκοντες τον κ. Χριστόφορο Προβατίδη και τον κ. Λεωνίδα Αλεξόπουλο. Περιληπτικά, η διαδικασία που ακολουθήθηκε είναι αναρτημένη στο «wikispace» του μαθήματος στον παρακάτω σύνδεσμο: « <https://biotech-ntua.wikispaces.com/carousel+ntua> » και περιλαμβάνει και κάποιες αρχικές ιδέες που υπήρξαν αλλά και τους λόγους για τους οποίους δεν υλοποιήθηκαν, κάτι που δεν θεωρήθηκε απαραίτητο να αναφερθεί στην παρούσα εργασία.





2. Θεωρητικό Υπόβαθρο

Για να γίνει κατανοητή η λειτουργία και η διαδικασία που ακολουθεί η ιστοκινέτα, αλλά και οι λόγοι για τους οποίους είναι χρήσιμη, θα πρέπει πρώτα να αναφερθεί το βιολογικό υπόβαθρο στο οποίο στηρίζεται.

2.1. Ιστολογία

Ιστολογία είναι ο κλάδος της Βιολογίας που μελετά την εσωτερική μορφολογία ζωικών και φυτικών ιστών εξετάζοντας λεπτές χρωματισμένες τομές. Έτσι, μπορεί να γίνει αντιληπτή μια γενικότερη εικόνα του ιστού, να μελετηθούν αλλαγές όσον αφορά στη μορφή του στα διάφορα στάδια της ζωής του, ή, σε περίπτωση ασθένειας, να διαγνωστεί κάποια ασθένεια ή να γίνει βιοψία για καρκίνο. Σημειώνεται ότι βιοψία δεν γίνεται μόνο με απώτερο σκοπό να διαγνωστεί η ύπαρξη καρκίνου ή όχι, αλλά και για να διαπιστώσει ο θεράπων ιατρός, κάθε φορά, εάν πρέπει να δοθεί αγωγή ή εάν η αγωγή που έχει ήδη δοθεί στον ασθενή είναι σωστή ή αν θα πρέπει να υπάρξει κάποια αλλαγή. Πιο συγκεκριμένα, όταν ο ιστός εξετάζεται για ασθένεια τότε οι διαδικασίες αυτές υπάγονται στον κλάδο της Ιστοπαθολογίας.

Μπορούν, λοιπόν, να αναφερθούν πολλά παραδείγματα εφαρμογών της ιστολογίας. Στην παρούσα εργασία θα αναπτυχθεί το παράδειγμα της οστεοαρθρίτιδας.

2.2. Οστεοαρθρίτιδα

Στο εργαστήριο Εμβιομηχανικής και Συστημικής Βιολογίας της σχολής Μηχανολόγων Μηχανικών, τα ιστολογικά πειράματα χρησιμοποιούνται κυρίως για την μελέτη της οστεοαρθρίτιδας, μια ασθένεια που ταλαιπωρεί πάνω από ένα εκατομμύριο Έλληνες, όπως αναφέρθηκε στο 65ο Πανελλήνιο Ορθοπεδικό Συνέδριο, τον Οκτώβρη του 2009, [11] 27 εκατομμύρια ανθρώπους στις Ηνωμένες Πολιτείες Αμερικής, 8 εκατομμύρια στη Μεγάλη Βρετανία και που αποτελεί μία από τις συχνότερες ασθένειες που αφορούν στην τρίτη ηλικία. Επιπροσθέτως, οι αρθροπλαστικές επεμβάσεις έχουν πολύ μεγάλο κόστος, γεγονός που τις καθιστά αδύνατες, για τις περισσότερες ομάδες ανθρώπων, να πραγματοποιηθούν. Άρα συμπεραίνεται η σημαντικότητα της μελέτης μιας τέτοιας ασθένειας.

Η οστεοαρθρίτιδα είναι μια ασθένεια η οποία έχει ως αποτέλεσμα την καταστροφή του χόνδρου των αρθρώσεων (articular cartilage) καθώς επίσης συχνά και του υποχονδρίου οστού (subchondral bone). Αυτό έχει ως αποτέλεσμα να αυξάνεται η τριβή μεταξύ των οστών προκαλώντας πόνο, δυσκαμψία και διάχυση υγρού στην άρθρωση, τις περισσότερες φορές. Ως αλυσιδωτή αντίδραση των παραπάνω, ο

ασθενής μειώνει την κίνηση του, λόγω του πόνου, και μπορεί να υπάρχουν προβλήματα χαλάρωσης σε γειτονικούς μύες και συνδέσμους. Η συγκεκριμένη ασθένεια παρατηρείται εντονότερα στις αρθρώσεις που καταπονούνται περισσότερο και παραλαμβάνουν μεγαλύτερο φορτίο, δηλαδή στα γόνατα (**Σχήμα 2.1.**) και στο ισχίο (**Σχήμα 2.2.**), χωρίς να εξαιρούνται όμως αρθρώσεις όπως εκείνες των δακτύλων, του καρπού και των μεσοσπονδύλιων δίσκων.



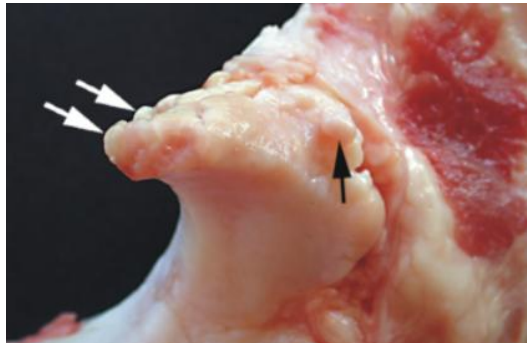
Σχήμα 2.1.: Επιβάρυνση γονάτου (knee) [12]

Τα αίτια ποικίλουν και μπορεί να βασίζονται σε παράγοντες όπως (για τις πρωτοπαθείς αρθροπάθειες) η ηλικία, καθώς παρατηρείται συχνότερη εμφάνιση μετά το 50^ο έτος του ανθρώπου, η έντονη και σκληρή σωματική άσκηση, ειδικά όταν επιβαρύνονται σημεία όπως τα γόνατα και οι γοφοί, το αυξημένο σωματικό βάρος αλλά και η γενετική προδιάθεση, η οποία μάλιστα παρατηρείται αυξημένη στους δίδυμους. Επίσης, οστεοαρθρίτιδα μπορεί να παρατηρηθεί και ως αποτέλεσμα άλλων ασθενειών ή τραυματισμών όπως (για τις δευτεροπαθείς αρθροπάθειες), δυσπλασία, φλεγμονώδεις παθήσεις, σηπτική αρθρίτιδα, καταστροφή της άρθρωσης από κάκωση κ.ά..



Σχήμα 2.2.: Επιβάρυνση ισχίου (hip) [12]

Ο αρθρικός χόνδρος αποτελείται από κολλαγόνο, πρωτεογλυκάνες και χονδροκύτταρα. Η ασθένεια ξεκινά όταν ελαττώνονται οι πρωτεογλυκάνες, γεγονός που έχει ως αποτέλεσμα να μειώνεται η ενυδάτωση του αρθρικού χόνδρου. Στο πρώτο στάδιο, λοιπόν, παρατηρείται αύξηση της δραστηριότητας των χονδροκυττάρων προκειμένου να συνθέσουν τις χαμένες πρωτεογλυκάνες. Το κολλαγόνο ποσοτικά παραμένει το ίδιο αλλά έχει αλλοιωθεί η ποιότητά του και ο ρυθμός κατά τον οποίο παράγεται. Συνοπτικά, η αντοχή του χόνδρου μειώνεται και οι ίνες του κολλαγόνου αρχίζουν να δέχονται μεγαλύτερη πίεση. Στο δεύτερο στάδιο, τα χονδροκύτταρα είναι περισσότερα και μεγαλύτερα και δεν μπορούν να εξασφαλίσουν τις ενεργειακές απαιτήσεις τους. Αυτό έχει ως συνέπεια την αυτολυτική ενζυμική καταστροφή των χονδροκυττάρων για να εξασφαλίσουν τροφή στα κύτταρα που απέμειναν. Οι απώλειες βασικών ουσιών απογυμνώνουν τις ίνες κολλαγόνου (τύπου II), οι οποίες χάνουν την αντοχή τους και διασπώνται ολοσχερώς. Σε αυτό το σημείο αρχίζει η παραγωγή ενός άλλου τύπου κολλαγόνου (τύπου I) το οποίο οδηγεί την ασθένεια σε μη αναστρέψιμα μονοπάτια. Το οστό, σε μια προσπάθεια να βελτιώσει την απορρόφηση των πιέσεων, δημιουργεί διάφορες εξοχές που μοιάζουν με ακάθια, τα λεγόμενα οστεόφυτα (**Σχήμα 2.3.**). Στο τρίτο και τελικό στάδιο ο αρθρικός χόνδρος εξαφανίζεται τελείως και αρχίζει η παραμόρφωση του υποχόνδριου οστού το οποίο γίνεται παχύτερο και σκληρότερο και σπάει τοπικά δημιουργώντας τη μορφή σφουγγαριού.



Σχήμα 2.3.: Δημιουργία οστεόφυτων [Wikipedia]

Τρόποι θεραπείας της οστεοαρθρίτιδας είναι:

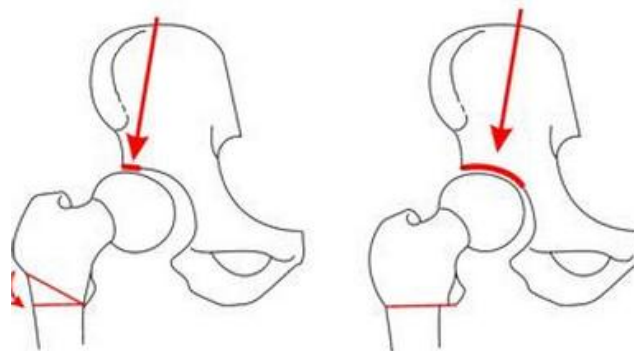
- Φαρμακευτική αγωγή: βασίζεται κυρίως σε αντιφλεγμονώδεις ουσίες όπως στεροειδή αντιφλεγμονώδη (κορτιζόνη), ή Μη Στεροειδή Αντιφλεγμονώδη (ΜΣΑΦ). Σε κάθε περίπτωση η χορήγηση φαρμάκων γίνεται τις ημέρες που υπάρχει έξαρση των συμπτωμάτων καθώς η χρόνια χορήγησή τους μπορεί να επιφέρει σοβαρότερη αλλοίωση της άρθρωσης.
- Υποστηρικτική αγωγή: όπως απώλεια βάρους για μείωση του φορτίου στην άρθρωση, ήπια άσκηση για σωστή ενδυνάμωση των γειτονικών μυών και συνδέσμων, φυσικοθεραπείες και

αποφόρτιση της πάσχουσας άρθρωσης με την βοήθεια μαστουινού ή ειδικού δυναμικού κηδεμόνα.

- Χειρουργική αγωγή: αρχικά, χρησιμοποιούνται οι αρθροπλαστικές επεμβάσεις κατά τις οποίες αντικαθίσταται εξολοκλήρου η άρθρωση με μια τεχνητή. Η μέθοδος αυτή έχει πολύ καλά αποτελέσματα όσον αφορά στις αρθρώσεις γόνατος (Σχήμα 2.4.), ισχίου, ώμου, αγκώνα και αρθρώσεων δακτύλων. Στις περιπτώσεις καρπού, αστραγάλου ή σπονδυλικής στήλης η αρθροπλαστική δεν έχει ικανοποιητικά αποτελέσματα και έτσι χρησιμοποιείται η μέθοδος της αρθρόδεσης κατά την οποία δεν υφίσταται πλέον η λέξη άρθρωση και η κίνηση περιορίζεται, μιας και τα δυο μέλη μένουν κολλημένα για πάντα. Άλλη μέθοδος είναι η οστεοτομία, με την οποία αλλάζουμε τον άξονα του οστού (Σχήμα 2.5.) έτσι ώστε να μειωθεί ο πόνος που δημιουργείται από την πίεση του φορτίου σε κάποιο συγκεκριμένο σημείο της άρθρωσης. Αυτή τη μέθοδο την εφαρμόζουμε σε πιο αρχικά στάδια για να αποτρέψουμε την εξάπλωση του προβλήματος σε όλη την άρθρωση.



Σχήμα 2.4.: Τεχνητή άρθρωση γόνατου [20]



Σχήμα 2.5.: Αλλαγή άξονα με οστεοτομία [21]



2.3. Τυπική Διαδικασία Εφαρμογών

Για να μελετηθεί η οστεοαρθρίτιδα, για να βρεθεί το στάδιο στο οποίο βρίσκεται και πόσο σοβαρό είναι το πρόβλημα, θα πρέπει να δημιουργηθούν ιστολογικές τομές ώστε με τη βοήθεια φωτοmikροσκοπίου να φανεί η δομή του χόνδρου.

A) Αρχικά, πρέπει να γίνει η **μονιμοποίηση** (fixation) του ιστού ακριβώς μετά την αφαίρεσή του από τον οργανισμό. Έτσι θα αποφευχθεί η αυτόλυση του ιστού και θα κρατήσει σταθερή τη δομή του.

1) Οι μονιμοποιητικές ουσίες ποικίλουν και η κάθε μία έχει τα πλεονεκτήματα και τα μειονεκτήματά της. Γι' αυτό το λόγο χρησιμοποιούνται διάφορα μίγματα αυτών ώστε να έχουμε το καλύτερο δυνατό αποτέλεσμα. Από τα πιο απλά και συνηθισμένα μονιμοποιητικά διαλύματα είναι το αλατούχο διάλυμα φορμαλίνης 10%. Το πρωτόκολλο που χρησιμοποιείται στο εργαστήριο, αναφέρει ότι πρέπει να αφήνεται ο ιστός στο παραπάνω διάλυμα για 4 ώρες.

2) Στην περίπτωση ιστών που έχουν μέρος από οστό πρέπει να γίνει **απασβεστοποίηση** έτσι ώστε να αφαιρεθεί το ασβέστιο και το οστό να γίνει πιο ευλύγιστο και να μπορεί να κοπεί αργότερα σε τομές για παρατήρηση στο μικροσκόπιο. Προτεινόμενη ουσία είναι η Cal-Ex, Fisher Scientific και το δείγμα αφήνεται για 6 με 12 ώρες ή και για ολόκληρο το βράδυ (για μικρά δείγματα).

3) Έπειτα, αφήνεται να τρέξει νερό βρύσης έτσι ώστε να ξεπλυθούν τα δείγματα, από 30 έως 60 λεπτά.

4) Τέλος, τα κρατάμε σε 70% διάλυμα αιθανόλης για όσο θέλουμε.

B) Στην συνέχεια, αφού τοποθετηθούν οι ιστοί στις ειδικές αναλώσιμες, πλαστικές κασετίνες στις οποίες έχει αναγραφεί το όνομα του δείγματος για την αποφυγή λαθών, λαμβάνει χώρα η **επεξεργασία ιστών** (tissue processing), όπου οι ιστοί μπαίνουν διαδοχικά στα παρακάτω διαλύματα:

1) 70% αιθανόλη για 30 λεπτά

2) 96% αιθανόλη για 30 λεπτά



- 3) 96% αιθανόλη για 30 λεπτά
- 4) 96% αιθανόλη για 30 λεπτά
- 5) 100% αιθανόλη για 30 λεπτά
- 6) 100% αιθανόλη για 30 λεπτά
- 7) 100% αιθανόλη για 30 λεπτά
- 8) Διάλυμα ξυλόλης για 1 ώρα
- 9) Διάλυμα ξυλόλης για 1 ώρα
- 10) Λιωμένη Παραφίνη στους 58 °C για 1 ώρα
- 11) Λιωμένη Παραφίνη στους 58 °C για 1 ώρα

Τα διαλύματα 1 έως και 7 είναι υπεύθυνα για την αφυδάτωση των ιστών με τη βοήθεια της αιθανόλης, ενώ τα διαλύματα 8 και 9 για τον καθαρισμό τους από τυχόν λίπη. Έπειτα, η θερμότητα προκαλεί εξάτμιση της ξυλόλης και τα κενά που αφήνονται γεμίζουν με παραφίνη, η οποία είναι υπεύθυνη για την σταθεροποίηση των ιστών.

Γ) Σε επόμενο στάδιο πρέπει να δημιουργηθούν τα **καλούπια παραφίνης**, σε σχήμα κύβου, στα οποία ενσωματώνονται οι ιστοί (embedding), έτσι ώστε να μπορέσουν να γίνουν δύσκαμπτοι και να τοποθετηθούν στο μικροτόμο.

Για την διαδικασία αυτή είναι απαραίτητη μία θερμή πλάκα στους 60 °C, μια ψυχρή πλάκα στους 4 °C, τα μεταλλικά καλούπια στα οποία θα φτιαχτούν οι κύβοι, λιωμένη παραφίνη στους 58 °C και λαβίδες για να πιάνουμε τους ιστούς.

Αρχικά, οι πλαστικές κασετίνες τοποθετούνται για λίγο στην θερμή πλάκα για να φύγουν τυχόν υπολείμματα κεριού γιατί το καλούπι μπορεί να μην γίνει ομοιόμορφο. Έπειτα, αναλόγως με το δείγμα και το μέγεθος του καλουπιού, τοποθετείται μικρή ποσότητα λιωμένης παραφίνης στο κατάλληλο μεταλλικό καλούπι, αφήνεται για 2 δευτερόλεπτα περίπου για να στερεοποιηθεί λίγο ο πάτος, τοποθετείται με τη λαβίδα ο ιστός στον πάτο μέχρι να κολλήσει. Αν δεν κολλάει ο ιστός γιατί έχει προλάβει ο πάτος να στερεοποιηθεί τελείως, το καλούπι πρέπει να έρθει λίγο σε επαφή με τη θερμή πλάκα για να ξαναλιώσει η παραφίνη και επαναλαμβάνεται η διαδικασία. Στη συνέχεια, το καλούπι πρέπει να σκεπαστεί με την πλαστική κασετίνα και να γεμιστεί με παραφίνη. Έπειτα, τοποθετείται πάνω στην ψυχρή πλάκα μέχρι να στερεοποιηθεί τελείως. Όταν θα είναι έτοιμο το μεταλλικό καλούπι θα βγει με μεγάλη ευκολία.

Δ) Έπειτα, τα καλούπια είναι έτοιμα για κοπή στον μικροτόμο και τη δημιουργία **τομών**.



Γι' αυτό το στάδιο χρειάζεται ένα δοχείο με πάγο, ένα δοχείο με κρύο νερό, μια δεξαμενή με ζεστό απεσταγμένο νερό (46-48 °C), διηθητικό χαρτί πάνω στον πάγο, φορτισμένα πλακίδια για να κολλήσουν οι τομές, στα οποία έχει αναγραφεί το όνομα του δείγματος για την αποφυγή λαθών, μια λαβίδα και δύο πινέλα. Το ένα πινέλο βοηθάει στην εύκολη απομάκρυνση των σωστών τομών από τον μικροτόμο ενώ το άλλο βοηθάει στον καθαρισμό του.

Τα καλούπια τοποθετούνται στο δοχείο με τον πάγο πριν κοπούν, για να μην μαλακώνει το κερί λόγω θερμοκρασίας περιβάλλοντος και να κοπεί πιο εύκολα. Επίσης, και κατά τη διάρκεια της κοπής μπορεί να τοποθετηθεί πάγος στο καλούπι για τον ίδιο λόγο, πολύ περισσότερο όταν η θερμοκρασία δωματίου είναι πολύ υψηλή. Στη συνέχεια, αφού το καλούπι μπει σωστά στη θέση του, στο μικροτόμο, τοποθετηθεί η ασφάλινη λεπίδα και ασφαλιστούν οι μοχλοί, κόβονται οι χοντρές τομές (15-25 μm) στην άκρη της λεπίδας. Αφού η λεπίδα φτάσει κοντά στον ιστό, πρέπει να κοπούν οι λεπτές τομές (3-5 μm) με το μεσαίο τμήμα της λεπίδας. Το τμήμα της λεπίδας πρέπει να αλλάζεται για να μένει πιο κοφτερό το τμήμα της λεπίδας που κόβει τις τομές του ιστού, που είναι και το επιθυμητό αποτέλεσμα. Έπειτα, συλλέγεται η «καλή» τομή, τοποθετείται στο δοχείο με το κρύο νερό και με τη βοήθεια του πλακιδίου τη μεταφέρεται στη δεξαμενή με το ζεστό νερό για να ανοίξει καλά και ξανασυλλέγεται με την πλάκα έτσι ώστε να κολλήσει καλά πάνω της. Καλό θα είναι να τιναχθεί ελαφρώς η πλάκα, πάνω στο διηθητικό χαρτί, με προσοχή, για να φύγει το νερό και αφήνεται να στεγνώσει όρθια.

Ε) Στο στάδιο της **χρώσης** χρησιμοποιούμε κυρίως, στο εργαστήριό μας, δύο είδη βαφής, το συνδυασμό της αιματοξυλίνης με την εωσίνη (Alum Haematoxylin and Eosin) και το κυανό της τολουιδίνης (Toluidine blue) που αποτελούν βασικές χρωστικές (εκτός από την εωσίνη που αποτελεί όξινη χρωστική), το οποίο σημαίνει ότι χρωματίζουν τα συστατικά των ιστών που είναι βασίφιλα.

Για το μείγμα της αιματοξυλίνης με την εωσίνη ακολουθείται η παρακάτω διαδικασία:

- 1) Τα πλακίδια με τους ιστούς ζεσταίνονται στον κλίβανο, στους 65 °C για 30 λεπτά.
- 2) Για να αφαιρεθεί το κερί από τον ιστό και να τον ενυδατώσουμε ακολουθούνται με τη σειρά τα παρακάτω βήματα:
 - 4 x Διήθηση σε ξυλόλη για 5 λεπτά (σύνολο 20 λεπτά)
 - 1 πέρασμα με μια σταγόνα ακετόνης
 - 3 x Διήθηση σε 100% αιθανόλη για 5 λεπτά (σύνολο 15 λεπτά)
 - 3 x Διήθηση σε 96% αιθανόλη για 5 λεπτά (σύνολο 15 λεπτά)
 - 1 x Διήθηση σε 70% αιθανόλη για 4 λεπτά



- 1 x Διήθηση σε 50% αιθανόλη για 4 λεπτά
 - Ξεπλένουμε με τρεχούμενο νερό βρύσης
- 3) Το πλακίδιο βυθίζεται στην αιμοτοξυλίνη για 2 λεπτά εάν είναι φρέσκια ή για 4 λεπτά αν δεν είναι.
 - 4) Ξεπλένεται με τρεχούμενο νερό βρύσης.
 - 5) Βυθίζεται για 1-2 δευτερόλεπτα σε όξινη αλκοόλη.
 - 6) Ξεπλένεται σε νερό βρύσης για 3 λεπτά, μέχρι να γίνουν οι τομές μπλε.
 - 7) Βυθίζεται το πλακίδιο στην εωσίνη για 3 λεπτά.
 - 8) Ξεπλένεται με τρεχούμενο νερό βρύσης για 2 λεπτά.
 - 9) Αφυδατώνουμε γρήγορα περνώντας το πλακίδιο από τις αλκοόλες.

Για το κυανό της τολουιδίνης ακολουθείται η παρακάτω διαδικασία:

- 1) Τα πλακίδια με τους ιστούς ζεσταίνονται στον κλίβανο, στους 65 °C για 30 λεπτά.
- 2) Για να αφαιρεθεί το κερί από τον ιστό και να τον ενυδατώσουμε ακολουθούμε με τη σειρά τα παρακάτω βήματα:
 - 4 x Διήθηση σε ξυλόλη για 5 λεπτά (σύνολο 20 λεπτά)
 - 1 πέρασμα με μια σταγόνα ακετόνης
 - 3 x Διήθηση σε 100% αιθανόλη για 5 λεπτά (σύνολο 15 λεπτά)
 - 3 x Διήθηση σε 96% αιθανόλη για 5 λεπτά (σύνολο 15 λεπτά)
 - 1 x Διήθηση σε 70% αιθανόλη για 4 λεπτά
 - 1 x Διήθηση σε 50% αιθανόλη για 4 λεπτά
 - Ξεπλένεται με τρεχούμενο νερό βρύσης
- 3) Το πλακίδιο βυθίζεται σε απεσταγμένο νερό για 10 λεπτά.
- 4) Εφαρμόζεται 0,25% w/v κυανό της τολουιδίνης στην πλευρά του πλακιδίου που είναι η τομή, ή το πλακίδιο βυθίζεται σε δοχείο με την παραπάνω χρωστική ουσία. Το κρατάμε έτσι για 5 λεπτά.
- 5) Ξεπλένεται σε απεσταγμένο νερό.
- 6) Οι τομές παρατηρούνται στο μικροσκόπιο για να διαπιστωθεί ότι έχουν χρωματιστεί καλά. Εάν ναι, η διαδικασία συνεχίζεται αλλιώς το πλακίδιο ξανατοποθετείται λίγο στη χρωστική.
- 7) Γρήγορη αφυδάτωση περνώντας το πλακίδιο από τις αλκοόλες (όχι τις ξυλόλες) του βήματος 2 αλλά με την ανάποδη σειρά, κρατώντας το πλακίδιο για 10 δευτερόλεπτα στο καθένα.
- 8) Βυθίζεται στην ξυλόλη για 5 λεπτά.
- 9) Τοποθετείται το προστατευτικό γυαλί με την βοήθεια της ειδικής κόλλας.



2.4. Διαβάθμιση Οστεοαρθρίτιδας

Για να μπορέσει να γίνει η μελέτη ενός αρθρικού χόνδρου και να βγουν κάποια συμπεράσματα για τη θεραπεία ή την αντιμετώπιση της ασθένειας των ίδιων των ασθενών ή πιθανών μελλοντικών ασθενών, θα πρέπει, έχοντας ως δεδομένα τα δείγματα που θέλουμε να ασχοληθούμε, να είναι γνωστές οι βασικές πληροφορίες για τον οργανισμό από τον οποίο προήλθε το δείγμα, όπως ηλικία, φύλο, πιθανές δραστηριότητες με τις οποίες καταπιανόταν ο ασθενής από τις οποίες μπορεί να επιβαρύνθηκε η άρθρωση ή πιθανά ατυχήματα και σοβαρά χτυπήματα που μπορεί να υπέστη η άρθρωση. Σε συνδυασμό με τα δεδομένα αυτά λοιπόν, θα πρέπει να είναι γνωστό το στάδιο της ασθένειας και πόσο έχει επεκταθεί ώστε για παράδειγμα να διαπιστωθεί τι είδους αντιμετώπιση πρέπει να χρησιμοποιηθεί. Σε αυτό το σημείο θα αναλυθεί ο τρόπος με τον οποίο υπολογίζεται ο συνολικός βαθμός της ασθένειας.

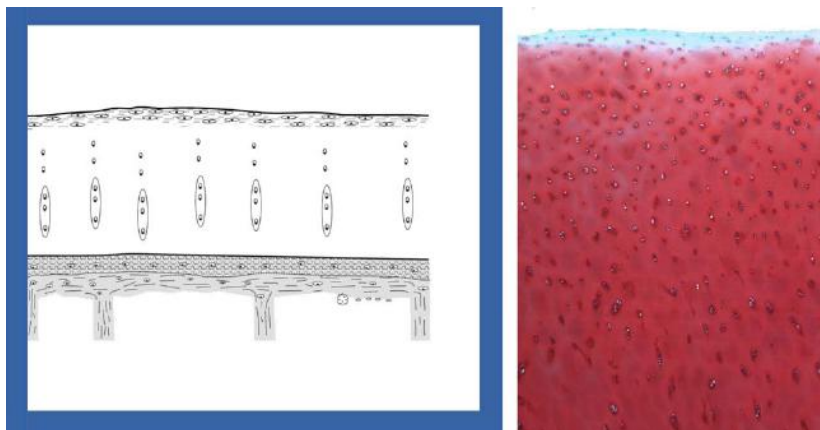
Σύμφωνα με το σύστημα OARSI (Osteoarthritis Research Society International) ο ιστός χαρακτηρίζεται αρχικά από έναν βαθμό (grade) που δείχνει τη σοβαρότητα της ασθένειας με το πόσο έχει προχωρήσει σε επίπεδο βάθους (0 έως 6) και από το στάδιο (stage) στο οποίο βρίσκεται που δείχνει την ποσοστιαία έκταση της ασθένειας (0 έως 4). Ο συνολικός βαθμός (score) μπορεί να υπολογιστεί από τον πολλαπλασιασμό μεταξύ των δύο (score = grade x stage).

Για να γίνει η επιλογή του **βαθμού**, παραθέτονται κάποια βασικά χαρακτηριστικά που μπορούν να παρατηρηθούν στο μικροσκόπιο:

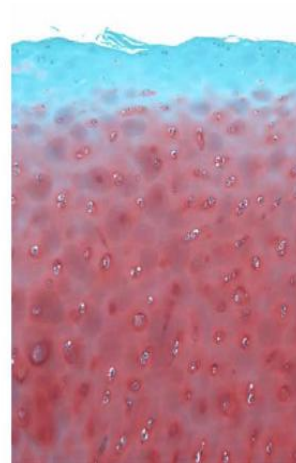
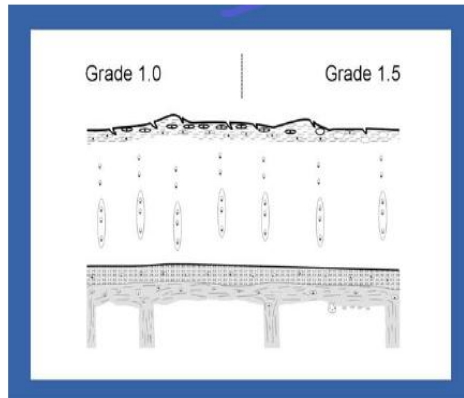
- Βαθμός 0: στην ουσία ο χόνδρος είναι υγιής, έχει λεία επιφάνεια και καλή οργάνωση των κυττάρων. (Σχήμα 2.6.)
- Βαθμός 1: η διαφορά από τον προηγούμενο βαθμό είναι ότι ξεκινούν να δημιουργούνται κάποιες μικρές εκδορές στην επιφάνεια που μπορεί να συνοδεύονται από κάποιο πρήξιμο (οίδημα). Επίσης, παρατηρείται μείωση, και ταυτόχρονα μεγαλύτερη δραστηριότητα πολλαπλασιασμού χονδροκυττάρων, αλλά και έλλειψη ικανότητας της μήτρας να χρωματιστεί καλά από βασοφιλικές χρωστικές. (Σχήμα 2.7.)
- Βαθμός 2: εδώ υπάρχει αποφλοιώση του ιστού (τα κομμάτια αυτά εμφανίζονται σαν νιφάδες ή ινίδια στο αρθρικό υγρό) ενώ το 1/3 περίπου της τομής δεν χρωματίζεται καλά από τις χρωστικές. (Σχήμα 2.8.)
- Βαθμός 3: οι ρωγμές προχωρούν σχεδόν κάθετα σε βαθύτερα στρώματα και να βγάζουν άλλα παρακλάδια υπό διαφορετικές γωνίες. Μπορεί να παρατηρηθεί επίσης εμφανής αποδιοργάνωση των χονδροκυττάρων και η ανικανότητα χρώσης στα 2/3 περίπου της τομής. (Σχήμα 2.9.)

- Βαθμός 4: σε αυτό το σημείο υπάρχει διάβρωση, αποκόλληση των ανώτερων στρωμάτων και δημιουργία κυστών στη μεσαία ζώνη. (Σχήμα 2.10.)
- Βαθμός 5: παρατηρείται απογύμνωση του υαλώδη χόνδρου (ο υαλώδης χόνδρος δεν αιματώνεται, συνεπώς δεν μπορεί να αναπλαστεί αλλά έχει μεγάλη αντοχή στους κραδασμούς) από την αρθρική επιφάνεια. Επίσης, μπορεί να εμφανιστεί ινώδης χόνδρος στην επιφάνεια, δηλαδή χόνδρος που επικοινωνεί με τον οργανισμό μέσω του αίματος αλλά δεν έχει μεγάλη αντοχή στις πιέσεις, χωρίς όμως να παραμορφώνεται η γεωμετρία της κοινής επιφάνειας. (Σχήμα 2.11.)
- Βαθμός 6: χαρακτηρίζεται από παραμόρφωση, το περίγραμμα της αρθρικής επιφάνειας αλλάζει όχι μόνο από κατάγματα στην αρθρική πλάκα αλλά και από έντονη μεταβολική δραστηριότητα της αρθρικής πλάκας του οστού και την ενεργοποίηση του των συνδετικών ιστών κυρίως πλευρικά αλλά και κεντρικά. (Σχήμα 2.12.)

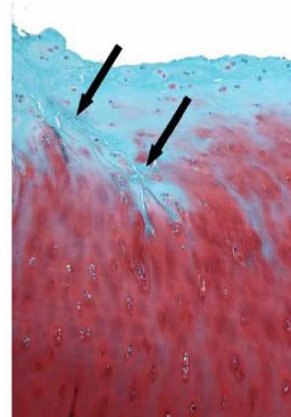
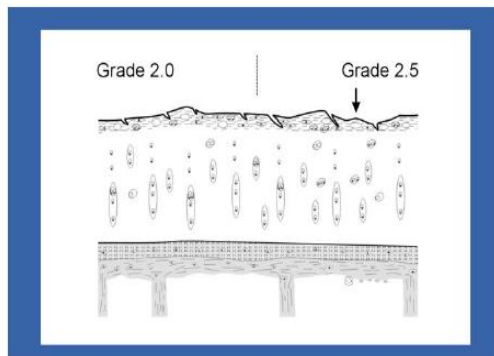
Μερικές φορές, για να τονιστεί ότι η ασθένεια από τη μία μπορεί να έχει χαρακτηριστεί από έναν βαθμό, αλλά από την άλλη οι ενδείξεις για αυτό το βαθμό είναι πιο προχωρημένες, προστίθεται 0,5 στον βαθμό όπως στις παρακάτω εικόνες (Σχήμα 2.6.-2.12.). Εκτός από τον βαθμό 0 που δεν υπάρχει κάποιο νόημα στο να οριστεί βαθμός 0,5. Στην ουσία η οστεοαρθρίτιδα ξεκινά από τον Βαθμό 1.



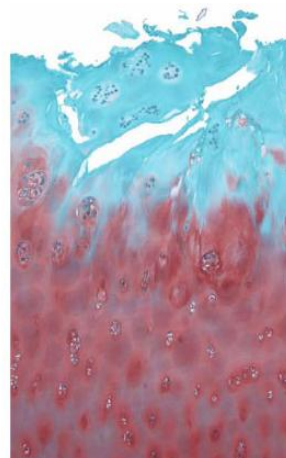
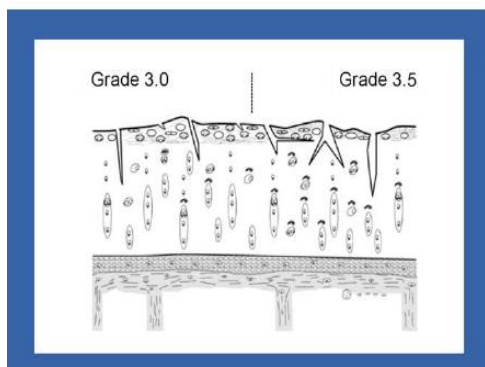
Σχήμα 2.6.: Μορφολογία ιστού με βαθμό 0 [1]



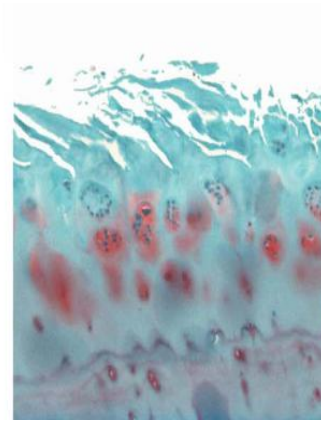
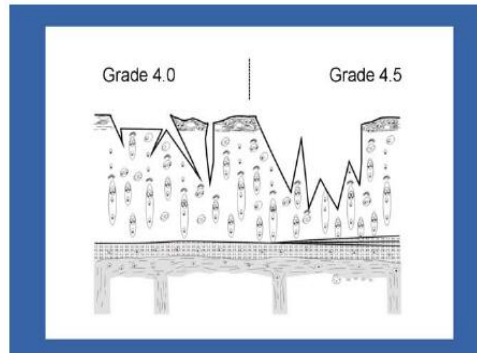
Σχήμα 2.7.: Μορφολογία ιστού με βαθμό 1 και 1,5 [1]



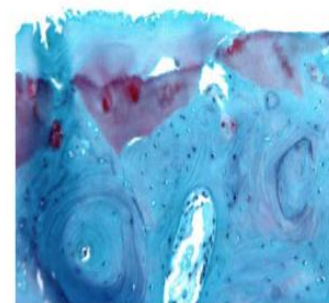
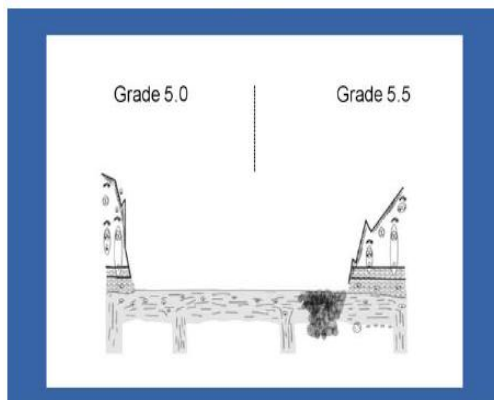
Σχήμα 2.8.: Μορφολογία ιστού με βαθμό 2 και 2,5 [1]



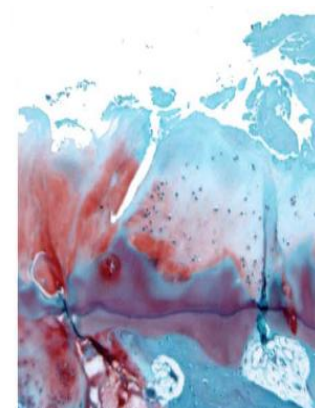
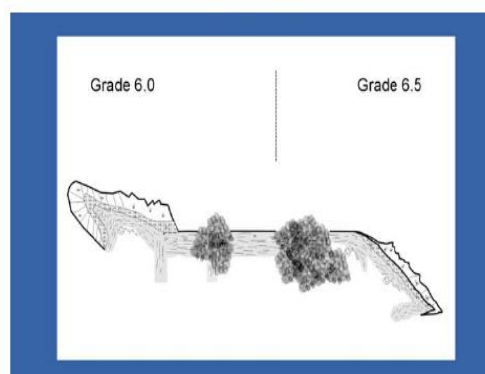
Σχήμα 2.9.: Μορφολογία ιστού με βαθμό 3 και 3,5 [1]



Σχήμα 2.10.: Μορφολογία ιστού με βαθμό 4 και 4,5 [1]



Σχήμα 2.11.: Μορφολογία ιστού με βαθμό 5 και 5,5 [1]



Σχήμα 2.12.: Μορφολογία ιστού με βαθμό 6 και 6,5 [1]



Στη συνέχεια, για να επιλεγεί το **στάδιο** στο οποίο έχει επεκταθεί η ασθένεια, θα πρέπει να παρατηρηθεί η οριζόντια έκταση της εμπλεκόμενης επιφάνειας χόνδρου ανεξάρτητα από τον βαθμό που έχουμε δώσει προηγουμένως [1]:

- Στάδιο 0: υγιής χόνδρος,
- Στάδιο 1: λιγότερο από 10 % της εμπλεκόμενης επιφάνειας,
- Στάδιο 2: 10-25 % της εμπλεκόμενης επιφάνειας,
- Στάδιο 3: 25-50 % της εμπλεκόμενης επιφάνειας,
- Στάδιο 4: περισσότερο από 50 % της εμπλεκόμενης επιφάνειας.

Στον παρακάτω πίνακα (**Πίνακας 2.1.**) φαίνεται ο τελικός βαθμός της ασθένειας πολλαπλασιάζοντας τον βαθμό με το στάδιο που έχει ευρεθεί (δεν υπάρχει λόγος να οριστεί συνολικός βαθμός ενός υγιούς χόνδρου και γι' αυτό δεν λαμβάνεται υπόψιν ο μηδενικός βαθμός και το μηδενικό στάδιο):

Συνολικός Βαθμός	Στάδιο 1	Στάδιο 2	Στάδιο 3	Στάδιο 4
Βαθμός 1	1	2	3	4
Βαθμός 2	2	4	6	8
Βαθμός 3	3	6	9	12
Βαθμός 4	4	8	12	16
Βαθμός 5	5	10	15	20
Βαθμός 6	6	12	18	24

Πίνακας 2.1.: Εύρεση συνολικού βαθμού οστεοαρθρίτιδας





3. Τρέχουσες Πληροφορίες και Ανάγκες που προκύπτουν

Το μόνο που είναι γνωστό είναι ο σκοπός της επιθυμητής συσκευής και κάποιες εικόνες και πληροφορίες για τις υπάρχουσες τεχνολογίες που κυκλοφορούν στην αγορά. Δυστυχώς, για το δεύτερο, δεν ήταν αδύνατο να γίνουν γνωστές λεπτομέρειες για τον τρόπο λειτουργίας και σχεδιασμού της συσκευής.

3.1. Σκοπός της Συσκευής

Ακολουθώντας την διαδικασία ιστολογικών εφαρμογών που αναπτύχθηκε στο **Κεφάλαιο 2.2.**, διαπιστώθηκε ότι υπάρχουν πέντε βασικά στάδια. Το δεύτερο στάδιο από αυτά (B) είναι η επεξεργασία ιστών που περιλαμβάνει την αφυδάτωση, τον καθαρισμό και τον εμποτισμό σε παραφίνη των ιστών. Αυτό το στάδιο είναι πολύ χρονοβόρο για να γίνει χειροκίνητα. Αναγκάζει στην ουσία τον πειραματιστή να ασχολείται τουλάχιστον για 7,5 ώρες με τη διαδικασία, δίνοντας μεγάλη προσοχή στον χρόνο που θα παραμείνει ο ιστός στο κάθε δοχείο, στην τοποθέτηση των ιστών στο σωστό δοχείο, στο να μην ξεχαστεί κάποιο δοχείο από τη διαδικασία και στην ρύθμιση της σωστής θερμοκρασίας για τα θερμαινόμενα δοχεία. Είναι απαραίτητη δηλαδή μία ολόκληρη μέρα εργασίας για την πραγματοποίηση μόνο ενός σταδίου.

Σκοπός λοιπόν αυτής της εργασίας είναι να δημιουργηθεί μία συσκευή που να είναι υπεύθυνη για την επεξεργασία των ιστών χωρίς να χρειάζεται ο πειραματιστής να είναι παρόν καθ' όλη τη διάρκεια και να ασχολείται με τα δείγματα. Το μόνο που θα πρέπει να κάνει είναι να επιβεβαιώσει ότι τα δοχεία έχουν την απαραίτητη ποσότητα διαλύματος και εάν όχι να συμπληρώσει, να τοποθετήσει την κασετίνα με τον ιστό και να ανοίξει το μηχάνημα. Μετά από 7,5 ώρες θα πρέπει η διαδικασία να έχει ολοκληρωθεί σωστά.

3.2. Υπάρχουσες Τεχνολογίες

Με τη βοήθεια του διαδικτύου διαπιστώθηκε ότι υπάρχουν κυρίως δύο είδη συσκευών επεξεργασίας ιστών (tissue processor), ο γραμμικός (**Σχήμα 3.1.**) και ο περιστροφικός (**Σχήμα 3.2.**). Ο γραμμικός μεταφέρει τις κασετίνες με τους ιστούς στο επόμενο δοχείο με γραμμική κίνηση ενώ ο περιστροφικός περιστρέφει τα δοχεία με κυκλική κίνηση έχοντας σταθερό σε ένα σημείο το καλάθι με τις κασετίνες ή περιστρέφει τις κασετίνες κρατώντας σταθερό το καλάθι.



Σχήμα 3.1.: Γραμμικός Επεξεργαστής Ιστών [28]






Σχήμα 3.2.: Περιτροφικός Επεξεργαστής Ιστών [29]

Ο δεύτερος είναι λιγότερο ογκώδης και μπορεί να μεταφερθεί εύκολα. Συνήθως, χρησιμοποιείται για μικρότερη ποσότητα κασετίνων σε μικρά εργαστήρια. Πολλές φορές μάλιστα αναφέρεται η ονομασία καρουζέλ (carousel) για να δηλώσει αυτή την περιστροφική κίνηση, εμπνευσμένη από το γνωστό παιχνίδι με τα περιστρεφόμενα αλογάκια.







Γενικότερα κυκλοφορούν διάφορα είδη ιστοκινέτας από διάφορες εταιρείες. Στον παρακάτω πίνακα (Πίνακας 3.1.) φαίνονται κάποια μοντέλα και κάποια χαρακτηριστικά τους. Θα πρέπει να σημειωθεί ότι όλα από τα παρακάτω μοντέλα έχουν την δυνατότητα χρήσης μπαταρίας σε περίπτωση διακοπής ρεύματος και ρύθμισης της διαδικασίας που θέλει ο χρήστης να γίνει έχοντας την δυνατότητα ρύθμισης της διάρκειας που θα μένουν οι κασετίνες στα δοχεία αλλά και της θερμοκρασίας που θα πρέπει να έχουν τα θερμαινόμενα δοχεία. Η απομάκρυνση των καπνών, που δημιουργούνται από την εξάτμιση




των διαλυμάτων, σε μερικές περιπτώσεις προτείνεται ως επιπλέον λειτουργία που μπορεί να έχει το μηχάνημα και μπορεί να γίνει είτε μέσω ειδικού χώρου, είτε μέσω κενού, είτε μέσω ανεμιστήρα-απορροφητήρα.

Όνομασία	Κατασκευαστική Εταιρεία	Χωρητικότητα Κασετίνων	Όγκος Μηχανήματος	Ιδιαίτερα Χαρακτηριστικά	Εικόνα
TP1020	Leica	100-200	Μικρός	-Εναλλακτικές για έλεγχο καπνού και σύστημα κενού. -Δεν καλύπτεται ολόκληρο με καπάκι	
ASP6025	Leica		Μεσαίος	-Μπαταρία σε περίπτωση διακοπής ρεύματος. -Καλύπτεται εντελώς. -Φίλτρα άνθρακα για τον καπνό. -Αυτόματο γέμισμα δοχείων	
ASP300S	Leica	300	Μεσαίος	-Καλύπτεται εντελώς. -Οθόνη αφής. -Σύστημα κενού για φιλτράρισμα και πίεσης για γρήγορο γέμισμα δοχείων	



PELORIS	Leica		Μεγάλος	-Πολλά δείγματα -Γρήγορα αποτελέσματα	
Excelsior ES	Thermo Scientific	222 (organized) 300 (random)	557440 cm ³ (Μεσαίος)	-Τροφοδοσία σε περίπτωση διακοπής ρεύματος. -Σήμα ότι τα δοχεία έχουν αδειάσει.	
Microm STP 120 Spin	Thermo Scientific	120 Or 240	283725 cm ³ (Μικρός)	-Μπαταρία σε περίπτωση διακοπής ρεύματος. -1 ή 2 καλάθια σε λειτουργία.	
STP 420 ES	Thermo Scientific	360 Or 420 (+60)	609865 cm ³ (Μεσαίος)	-Ανεξάρτητη διπλή διαδικασία ή και ταυτόχρονη αν είναι επιθυμητό. (Το δεύτερο σύστημα έχει θέση για 60 κασετίνες)	
ATP-700	IHC WORLD	144 (2*72)	317418750 cm ³ (Μεγάλος)	-2 καλάθια σε λειτουργία. -Λειτουργία καθυστέρησης έναρξης προγράμματος.	
ATP-700-ST	IHC WORLD	72	317418750 cm ³ (Μεγάλος)	-1 καλάθι σε λειτουργία.	



KD-TS6B	IHC WORLD	(όγκος μικρότερου δοχείου 1 L)	188166 cm ³ (Μικρός)	-Σύστημα κενού για την επιτάχυνση των διαδικασιών.	
KD-TS3D1	IHC WORLD	150	266952 cm ³ (Μικρός)	-Αυτόματος ανεμιστήρας	
KD-TS1A	IHC WORLD	40	189000 cm ³ (Μικρός)	-Μνήμη για δημιουργία διαδικασίας	
Citadel 2000	Shandon (RBS)	100	952201 cm ³ (Μεσαίος)	-Διαθέσιμο σύστημα κενού -Προγραμματισμός έως και 9 μέρες πριν -Χώρος διαφυγής καπνών.	
L12601	EMS		84066 cm ³ (Μικρός)	-Θερμαινόμενα δοχεία με σύστημα Πελτιέ -Σύστημα αντλίας κενού.	

Πίνακας 3.1.: Υπάρχοντα Μοντέλα Ιστοκινέτας

3.3. Ανάγκες

Παρατηρώντας τα μοντέλα ιστοκινέτας που κυκλοφορούν στην αγορά μπορεί να παρατηρηθεί ότι δεν υπάρχει καμία συσκευή η οποία να διευκολύνει μικρά εργαστήρια με λίγα δείγματα. Αυτό συμβαίνει γιατί με τις υπάρχουσες τεχνολογίες ο ερευνητής θα είναι αναγκασμένος για 5 κασετίνες, για παράδειγμα, να λειτουργήσει όλο το σύστημα και να σπαταλάει διαλύματα, αναλώσιμα αλλά και περισσότερη ενέργεια. Επιπλέον, στις περιπτώσεις που μετά τη διαδικασία τα διαλύματα των δοχείων απορρίπτονται και στην θέση τους τοποθετούνται νέα, ή στις περιπτώσεις που κάποια δοχεία έρχονται σε επαφή με τον ατμοσφαιρικό αέρα επειδή δεν είναι τελείως κλειστά, εάν η συσκευή δεν έχει λειτουργήσει για πολλά δείγματα, γίνεται σπατάλη των διαλυμάτων που παίρνουν μέρος στη



διαδικασία, τα οποία είναι εξαιρετικά πτητικά και εξατμίζονται (αιθανόλη αλλά πολύ περισσότερο η ξυλόλη). Πόσο μάλλον όταν τα δοχεία που είναι διαθέσιμα στην αγορά είναι του ενός με δύο λίτρων το λιγότερο.

Ακόμα και αν οι συσκευές που αναφέρθηκαν στον **Πίνακα 3.1.** είναι ικανοποιητικές, το κόστος τους αρχίζει συνήθως από 2000 ευρώ και μπορεί να ανέβει μέχρι και 11000 ευρώ. Γίνεται αντιληπτό λοιπόν ότι αυτές οι ιστοκινέτες είναι δύσκολο να υπάρξουν σε ένα ελληνικό πανεπιστήμιο ή σε ένα μικρό εργαστήριο.

Από τα παραπάνω διαπιστώνεται η ανάγκη δημιουργίας μίας ιστοκινέτας για λίγα δείγματα, με μικρούς όγκους δοχείων, μικρό όγκο στο χώρο και να είναι οικονομικά προσιτή σε μεγαλύτερο φάσμα εργαστηρίων.

Τέλος, επειδή στο εργαστήριο, όταν ξεκίνησε να εκπονείται η εργασία, δεν υπήρχαν πολλά δείγματα για ιστολογική μελέτη, δεν γίνονταν συχνές επεξεργασίες ιστών οπότε γι' αυτό το λόγο λήφθηκε ως αρχικό συμπέρασμα ότι υπήρξε ανάγκη μιας συσκευής που θα δέχεται μία μόνο κασετίνα.



4. Προδιαγραφές Συστήματος

Οι τεχνικές προδιαγραφές της ιστοκινέτας σύμφωνα με τις ανάγκες του συστήματος είναι:

- Ιστοκινέτα περιστρεφόμενου τύπου, καρουζέλ
- Ικανότητα επεξεργασίας μίας μόνο κασετίνας με ιστό, για μικρά δείγματα
- Ελάχιστη χωρητικότητα 11 δοχείων
- 9 γυάλινα δοχεία για τις αιθανόλες
- 2 ανοξείδωτα δοχεία για την παραφίνη
- Εύκολα αποσπώμενα δοχεία για τον καθαρισμό τους
- Θέρμανση της παραφίνης σε θερμοκρασίες μεταξύ 58-60°C
- Ασφάλεια για τους χρήστες και τους γύρω τους
- Μικρός όγκος μηχανήματος για εύκολη μεταφορά και τοποθέτηση
- Δοχεία μικρών όγκων ώστε να χωράει μία κασετίνα
- Καπάκι για την προστασία των διαλυμάτων
- Σχεδιασμός για παραγωγή
- Να έχει το μικρότερο δυνατό κόστος
- Ο πειραματιστής να μην χρειάζεται καθόλου να ασχολείται με την εφαρμογή κατά τη διάρκεια της διαδικασίας
- Ο πειραματιστής να μπορεί να αναγνωρίσει το στάδιο στο οποίο βρίσκεται η διαδικασία οποιαδήποτε στιγμή
- Να ακολουθείται το πρωτόκολλο του Παραρτήματος Β' χωρίς επιπλέον ρύθμιση.





5. Περιγραφή και Σχεδιασμός Μηχανολογικής Διάταξης

Για να γίνει ο σχεδιασμός της συσκευής λήφθηκαν υπόψιν στοιχεία όπως το οικονομικό συμφέρον του εργαστηρίου, τα εργαλεία και τα υλικά στα οποία υπάρχει πρόσβαση μέσω του εργαστηρίου, η ευκολία της συναρμολόγησης και συντήρησης, η ασφάλεια των χρηστών και όλες οι τεχνικές προδιαγραφές που αναφέρθηκαν στο Κεφάλαιο 4.

Για τον πρώτο λόγο, αποφασίστηκε να κατασκευαστούν όσο το δυνατόν περισσότερα κομμάτια δύο διαστάσεων έτσι ώστε να γίνουν με υδροκοπή και όχι με κάποια εργαλειομηχανή, μια φθηνή, απλή και πολύ αποτελεσματική, τις περισσότερες φορές, ανάλογα με την ποιότητά της, μέθοδο. Για τα κομμάτια που αναγκαστικά έπρεπε να γίνουν σε τρεις διαστάσεις δόθηκε προσοχή στο να γίνουν, όσο γινόταν, απλά στην κατασκευή και να μπορούν να κατασκευαστούν με τα μέσα που υπάρχουν στο εργαστήριο. Έγινε προσπάθεια, δηλαδή, να σχεδιαστεί η ιστοκινέτα όσο γίνεται με κομμάτια που μπορούν να επαναληφθούν με ευκολία ώστε να υπάρξει σχεδιασμός για παραγωγή (design for manufacturing) είτε σε περίπτωση καταστροφής κάποιου τεμαχίου, είτε σε περίπτωση εξ'ολοκλήρου παραγωγής της συσκευής.

Τα βασικά στοιχεία που αποφασίστηκε ότι θα πρέπει να έχει η συσκευή είναι ένα σύστημα περιστροφικής κίνησης των δοχείων, ένα σύστημα συγκράτησης της κασετίνας το οποίο θα της επιτρέπει να μπαίνει και να βγαίνει στα δοχεία με μία κάθετη γραμμική κίνηση και ένα σύστημα θερμαινόμενων δοχείων στα οποία θα ζεσταίνεται η παραφίνη σε συγκεκριμένη θερμοκρασία.

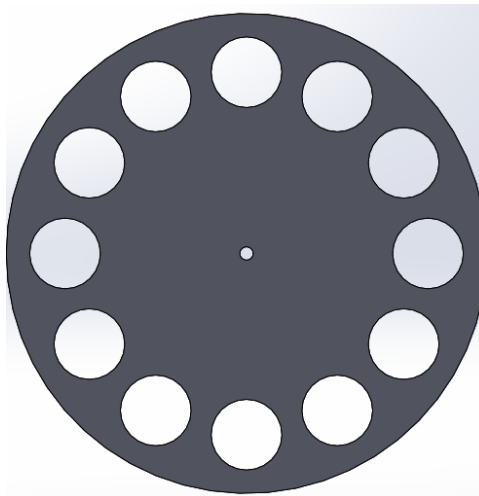
Αρχικά, κρίθηκε σωστό να απομακρυνθούν τα θερμαινόμενα δοχεία από την περιστρεφόμενη κίνηση για να αποφευχθεί η περιστροφή ή η ημιπεριστροφή καλωδίων, να μην βρίσκεται πηγή θερμότητας ακριβώς δίπλα σε πολύ πτητικά υγρά όπως είναι οι αιθανόλες αλλά πολύ περισσότερο η ξυλόλη, η οποία εξατμίζεται ακόμα και από κλειστό μπουκάλι, και για να αποτρεπεί η ανισοκατανομή του βάρους, καθώς το σύστημα θέρμανσης σε συνδιασμό με το σύστημα μόνωσης των θερμαινόμενων δοχείων θα είχε πρωτίστως μεγαλύτερο βάρος και δευτερευόντως μεγαλύτερο όγκο.

Τοποθετώντας τα θερμαινόμενα δοχεία έξω από το σύστημα περιστροφικής κίνησης και έχοντάς τα ακίνητα, θα πρέπει να δοθεί στο σύστημα συγκράτησης της κασετίνας η δυνατότητα να μπορεί να πάει σε αυτά και να μπορεί να κάνει την ίδια κάθετη κίνηση προκειμένου να εισέλθει στην παραφίνη. Γι'αυτό το λόγο τοποθετήθηκε ένα σύστημα περιστρεφόμενου βραχίονα ο οποίος θα μπορεί να πάει σε τρεις συγκεκριμένες θέσεις, μία θέση για όλα τα περιστρεφόμενα δοχεία και δύο θέσεις για τα δύο

ακίνητα, θερμαινόμενα δοχεία. Τα συστήματα αυτά αναλύονται με περισσότερες λεπτομέρειες στα επόμενα υποκεφάλαια (5.1.-5.6.) ενώ τα κατασκευαστικά τους σχέδια βρίσκονται στο Παράρτημα Δ'.

5.1. Σύστημα Περιστροφής Δοχείων

Για το ιστολογικό πείραμα είναι απαραίτητη η περιστροφή εννέα (9) δοχείων στα οποία θα είναι τα διαλύματα και ένα επιπλέον που θα αποτελεί την τελική θέση της κασετίνας, δηλαδή δέκα (10) δοχεία συνολικά. Θεωρήθηκε, λοιπόν, απαραίτητη μια πλατφόρμα για την συγκράτησή τους, δηλαδή ένας δίσκος με ειδικές οπές, με διάμετρο λίγο μεγαλύτερη από εκείνη των δοχείων (διάμετρος δοχείων 38 mm) που θα μπουν τα διαλύματα, της τάξεως δηλαδή των 2 χιλιοστών, για να μπορούν να τοποθετούνται τα δοχεία με ευκολία, όπως φαίνεται στο **Σχήμα 5.1.** Επίσης, δημιουργήθηκε και ένας δεύτερος δίσκος, ο οποίος τοποθετήθηκε πάνω από τον προηγούμενο με μεγαλύτερη διάμετρο για να δημιουργηθεί έτσι μια «πατούρα», ένα σκαλοπάτι δηλαδή, για την καλύτερη ασφάλεια των δοχείων. Ο δίσκος αυτός φαίνεται στο **Σχήμα 5.2.**

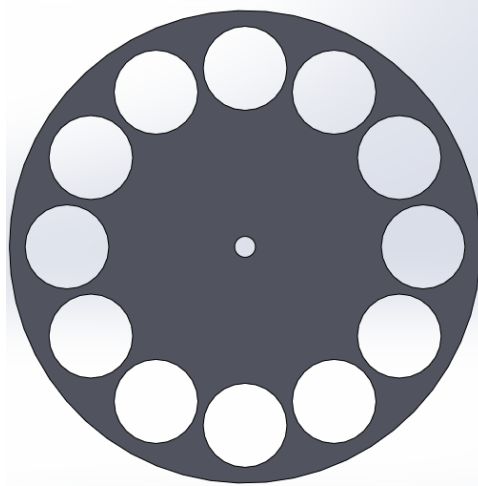


Σχήμα 5.1.: Κάτω δίσκος για τοποθέτηση δοχείων

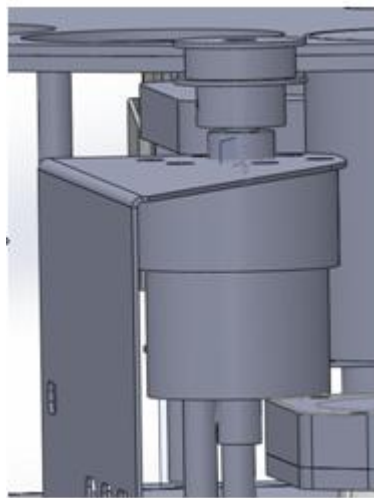
Σε αυτό το σημείο θα πρέπει να σημειωθεί ότι ο δίσκος έγινε δώδεκα (12) θέσεων και όχι δέκα (10) γιατί όταν αργότερα έγινε έρευνα για παραγγελία ενός κωδικοποιητή (encoder), για να μπορεί το σύστημα να γνωρίζει σε ποια ακριβώς θέση βρίσκεται, βρέθηκε ένας 12 θέσεων σε πολύ καλή τιμή και συνέφερε πολύ οικονομικά σε σχέση με εκείνον που βρέθηκε για 10 θέσεις.

Στη συνέχεια, για να χρειαστούν λιγότερα και πιο απλά τεμάχια, αποφασίστηκε ότι η μετάδοση της κίνησης έπρεπε να γίνει από την άκρη, με επαφή του δίσκου με ένα ελαστομερές, το οποίο είναι συνδεδεμένο στον άξονα του κινητήρα (**Σχήμα 5.3.**). Ο κινητήρας συνεχούς ρεύματος (DC) που

χρησιμοποιήθηκε στηρίζεται σε άλλο τεμάχιο, με όνομα αρχείου Solidworks “vasi_thermainomena”, με βίδες M3.



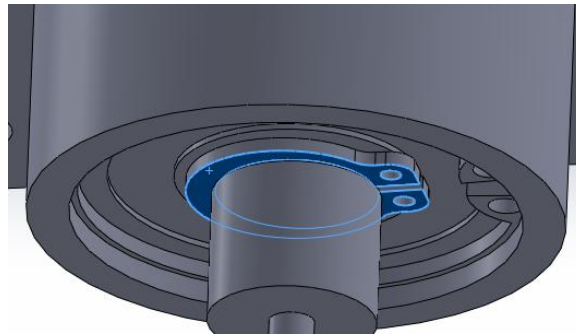
Σχήμα 5.2.: Πάνω δίσκος για τοποθέτηση δοχείων



Σχήμα 5.3.: Ελαστομερές και δίσκος σε επαφή

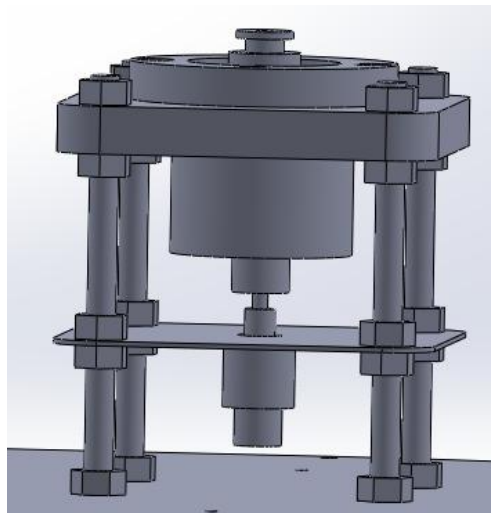
Η στήριξη των δίσκων έγινε κάτω από το κέντρο τους με τη βοήθεια μιας φωλιάς ρουλεμάν (BGRBB6201ZZ-40) που αγοράστηκε έτοιμη με δύο ρουλεμάν βαθείας αύλακος (B6201), τα οποία είναι υπεύθυνα για την παραλαβή ακτινικών φορτίων. Η φωλιά χρειάζεται να βιδωθεί και για αυτό το λόγο δημιουργήθηκε το τεμάχιο με όνομα αρχείου Solidworks “vasi_gia_housing”. Για να υπάρξει παραλαβή και αξονικών φορτίων δημιουργήθηκε ένας άξονας με ειδική εσοχή (Solidworks Part: “aksonaki_housing”) στην οποία τοποθετήθηκε δαχτυλίδι (Solidworks Part: “stwn12_2_03”) στο κάτω ρουλεμάν για συγκράτηση, όπως φαίνεται στην κάτω όψη του συστήματος φωλιά-άξονα στο Σχήμα

5.4. Για την στήριξη του συστήματος φωλιά-άξονας-βάση χρησιμοποιήθηκαν ντίζες και παξιμάδια Μ6. Για την ενδιάμεση στήριξη των ντιζών κατασκευάστηκε το τεμάχιο με όνομα αρχείου Solidworks “endiamesi_sygkratisi” που στερεώθηκε με τη βοήθεια παξιμαδιών Μ6 πάνω και κάτω. Ενώ για την στήριξη των δίσκων στον άξονα, έτσι ώστε να τον περιστρέφουν και αυτόν, τοποθετήθηκε ειδική βίδα που αγοράστηκε έτοιμη (dbgs5).

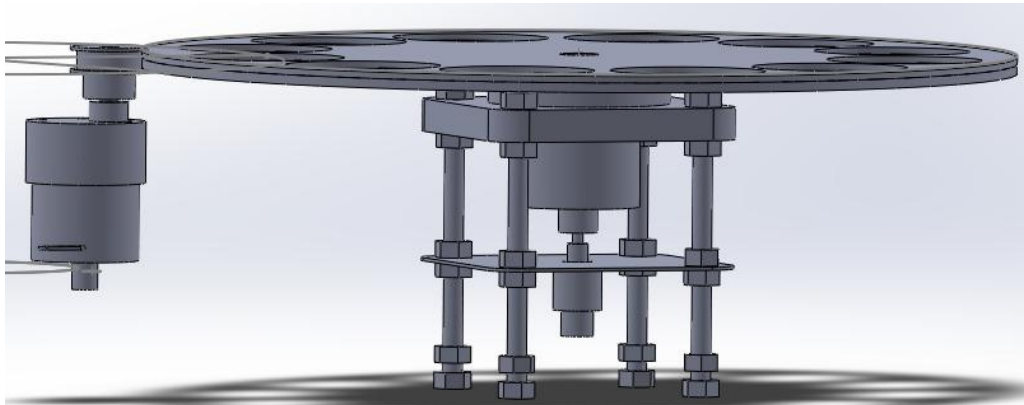


Σχήμα 5.4.: Δακτύλιος για την παραλαβή αξονικών φορτίων

Στο **Σχήμα 5.5.** φαίνεται το σύστημα στήριξης των δίσκων ενώ στο **Σχήμα 5.6.** φαίνεται ολόκληρο το σύστημα περιστροφικής κίνησης των δίσκων, και στα δύο σχήματα φαίνεται και ο κωδικοποιητής που συνδέεται με τον άξονα με μία σκουληκόβίδα.



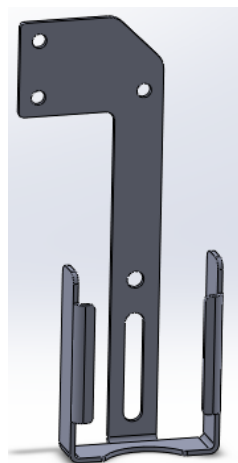
Σχήμα 5.5.: Σύστημα στήριξης δίσκων



Σχήμα 5.6.: Σύστημα περιστροφής δίσκων

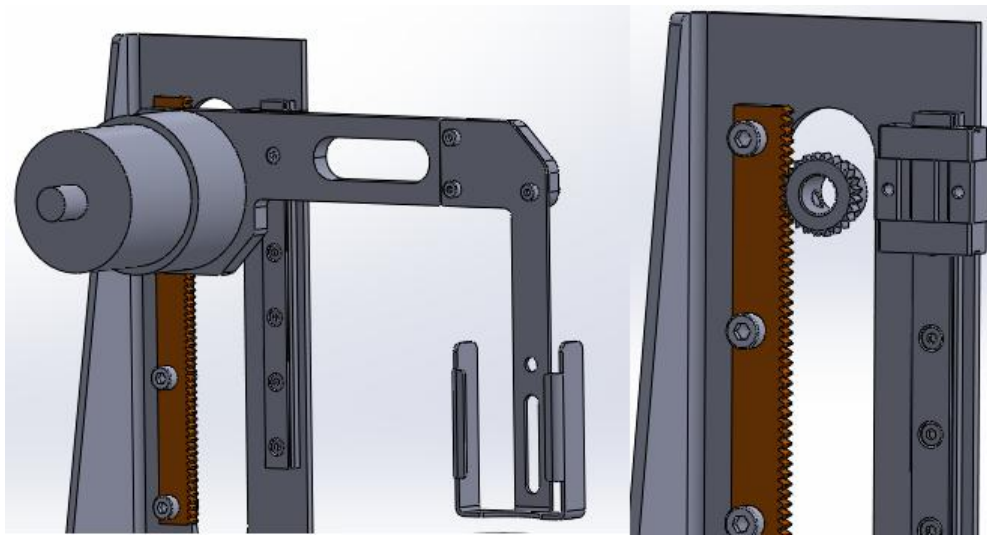
5.2. Σύστημα Γραμμικής Κίνησης

Το σύστημα συγκράτησης της κασετίνας, με το δείγμα, πρέπει να μπορεί να πραγματοποιεί μια κάθετη γραμμική κίνηση, να έχει δηλαδή μια συμπεριφορά παρόμοια με έναν ανελκυστήρα, για να μπορούν τα δείγματα να βυθιστούν στα διάφορα διαλύματα. Αρχικά, δημιουργήθηκε ένα τεμάχιο, στο οποίο θα μπορεί να τοποθετηθεί η κασετίνα, με ειδικές στρατζαριστές διαμορφώσεις ώστε να τοποθετείται, έτσι ώστε να μην φύγει την ώρα που θα μπαίνει στο δοχείο, αλλά ταυτόχρονα να μπορεί να αφαιρεθεί χωρίς δυσκολία. Το τεμάχιο αυτό, που έχει όνομα αρχείου Solidworks “asanser”, έγινε από ανοξείδωτο χάλυβα για να έχει μεγαλύτερη αντοχή στα χημικά διαλύματα με τα οποία θα έρθει σε επαφή πολλές φορές και φαίνεται στο Σχήμα 5.7..



Σχήμα 5.7.: Τεμάχιο για την τοποθέτηση της κασετίνας

Ο κινητήρας συνεχούς ρεύματος που είναι υπεύθυνος για αυτή την κίνηση τοποθετήθηκε στο τεμάχιο που έχει όνομα αρχείου Solidworks “stiriksi_asanser” και η σύνδεση έγινε με βίδες. Από την άλλη μεριά του συγκεκριμένου τεμαχίου συνδέθηκε το τεμάχιο που φαίνεται στο **Σχήμα 5.7.** με βίδες και παξιμάδια M2. Στον άξονα του κινητήρα τοποθετείται μικρό γρανάζι με τη βοήθεια σκουληκόβιδας, το οποίο σε συνδιασμό με έναν σταθερό οδοντωτό κανόνα εξασφαλίζει την γραμμική κίνηση (κρεμαγιέρα). Για καλύτερη οδήγηση και για στήριξη του συστήματος αυτού, χρησιμοποιήθηκε γραμμικός οδηγός με ενσωματωμένα ρουλεμάν που επιτρέπουν την γραμμική κίνηση. Στο **Σχήμα 5.8.** φαίνεται ολόκληρο το σύστημα της κίνησης αλλά και το σύστημα κανόνα-οδήγησης ξεχωριστά. Για να γνωρίζει το σύστημα πότε πρέπει να σταματήσει την πάνω ή την κάτω κίνηση προστέθηκαν δύο (2) μηχανικοί διακόπτες, ένας πάνω και ένας κάτω για να εισάγουν τα όρια στο σύστημα.

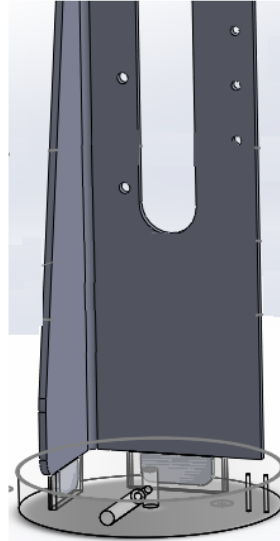


Σχήμα 5.8.: Σύστημα γραμμικής κίνησης

5.3. Σύστημα Περιστροφής Βραχίονα

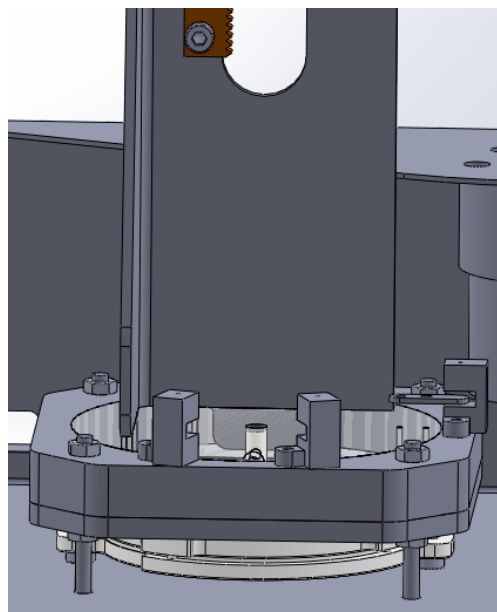
Για να περιστραφεί ο βραχίονας στις τρεις επιθυμητές θέσεις χρησιμοποιήθηκε βηματικός κινητήρας τύπου ultra flat, δηλαδή πολύ λεπτός έτσι ώστε να μην πιάνει πολύ χώρο στο σύστημα. Αυτός με τη σειρά του προσδένεται από τις άκρες του στο τεμάχιο που έχει όνομα αρχείου Solidworks “plaka_stiriksis_kinitira” με βίδες στο σπείρωμα του τεμαχίου, ενώ αφήνεται να περάσει ανεμπόδιστα ο άξονας του κινητήρα από την κεντρική του τρύπα. Πάνω από αυτό το τεμάχιο προσδένεται ένα άλλο με όνομα αρχείου Solidworks “vasi_kinit” στο οποίο βιδώνονται τρεις (3) φωτοαισθητήρες, για να γνωρίζει το σύστημα που είναι οι τρεις επιθυμητές θέσεις του βραχίονα, και στο εσωτερικό του αφήνεται να ολισθαίνει ένας κύλινδρος από plexi glass το οποίο δουλεύει σαν έδρανο ολίσθησης

(όνομα αρχείου Solidworks “edrano_olisthisis”). Σε αυτό το έδρανο λοιπόν, έγιναν ειδικές εσοχές στις οποίες ο κάθετος βραχίονας (όνομα αρχείου Solidworks “kathetos_vraxionas”) κουμπώνει με τις ειδικές προεξοχές του (Σχήμα 5.9.). Στο κέντρο του εδράνου βρίσκεται μία οπή από την οποία θα περάσει ο άξονας του κινητήρα και θα σφίξει σε αυτή με τη βοήθεια σκουληκόβιδας.



Σχήμα 5.9.: Σύνδεση εδράνου ολίσθησης με κάθετο βραχίονα

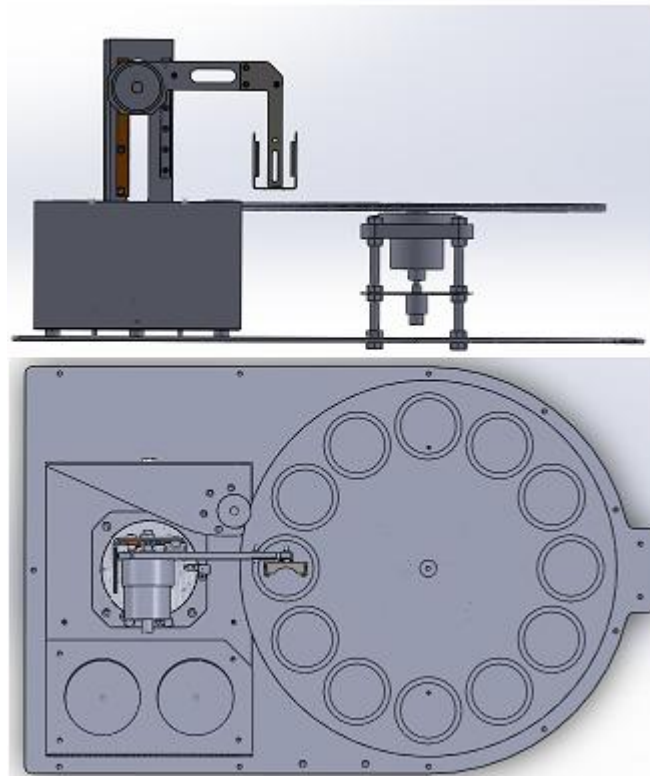
Όλο το σύστημα στήριξης του κινητήρα και του βραχίονα στηρίζεται με ντίζες M3 και παξιμάδια στο τεμάχιο με όνομα αρχείου Solidworks “vasi_thermainomena”, όπως φαίνεται στο Σχήμα 5.10..



Σχήμα 5.10.: Σύστημα περιστροφής βραχίονα

5.4. Σύστημα Θερμαινόμενων Δοχείων

Τα δύο δοχεία που θα περιέχουν την παραφίνη επιλέχθηκαν να είναι από ανοξείδωτο χάλυβα για να μην έχουν προβλήματα με τις συνεχείς αυξήσεις και πτώσεις της θερμοκρασίας. Κάτω από κάθε δοχείο τοποθετήθηκε ένας θερμαντήρας, ενώ μέσα στο δοχείο είναι στερεωμένος ένας αισθητήρας θερμότητας. Για να μην υπάρχουν πολλές απώλειες στο σύστημα τα δύο δοχεία αυτά τοποθετήθηκαν σε ειδικά διαμορφωμένο κουτί από μονωτικό υλικό, και συγκεκριμένα από πυριτικό ασβέστιο. Στις σπές υπάρχει χώρος για τον θερμαντήρα ενώ δημιουργήθηκαν ειδικές σπές για να απομακρυνθούν τα καλώδια. Το μονωτικό κουτί συνδέεται με το τεμάχιο που έχει όνομα αρχείου Solidworks “vasi_thermainomena”. (Σχήμα 5.11.)

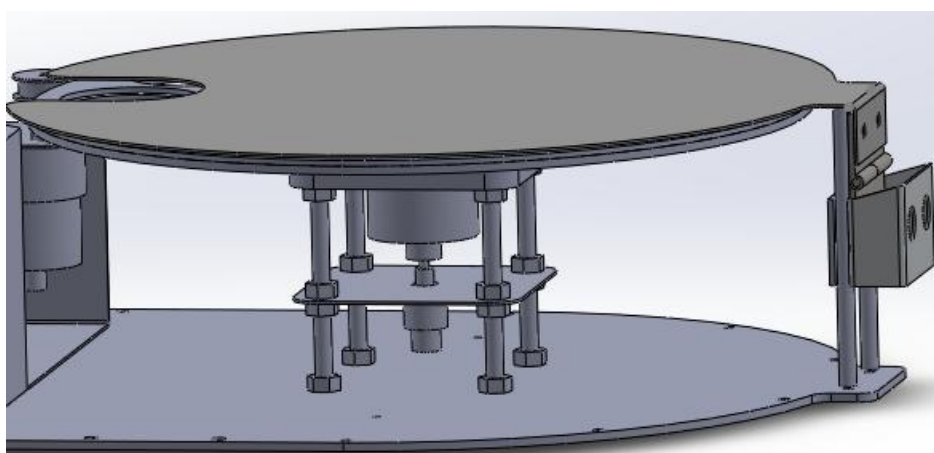


Σχήμα 5.11.: Σύστημα με τους τρεις κινητήρες και το μονωτικό κουτί

5.5. Σύστημα Προφύλαξης Διαλυμάτων

Τα διαλύματα που περιέχουν τα δοχεία, δηλαδή οι αιθανόλες και η ξυλόλη, είναι ιδιαίτερος πτητικά όπως έχει προαναφερθεί, και αρκετά ακριβά με αποτέλεσμα να μην υφίσταται η πολυτέλεια της σπατάλης των διαλυμάτων. Για να μην εξατμίζονται λοιπόν, αλλά και για να μην μπαίνει σκόνη ή άλλα

σκουπίδια στα διαλύματα κατασκευάστηκε καπάκι με διαμόρφωση τέτοια ώστε να είναι ανοιχτό από πάνω μόνο το δοχείο αυτό που χρησιμοποιείται την συγκεκριμένη στιγμή, επιτρέποντας τη βύθιση της κασετίνας στο διάλυμα του εκάστοτε σταδίου της διαδικασίας. Το καπάκι αυτό ανοιγοκλείνει χειροκίνητα με μια άρθρωση (μεντεσές). Για να μπορεί το καπάκι να σταματάει κάπου και να μην πέφτει στο τραπέζι, κόπηκε ένας κεκλιμένος σωλήνας ορθογωνικής διατομής ώστε να στέκεται εκεί. Ο σωλήνας αυτός βιδώνεται με μια πλάκα, ανάμεσα από τα οποία υπάρχει πεπλατυσμένος άξονας τον οποίο σφήγγουν, για να στηρίζονται, καθώς βιδώνονται (Σχήμα 5.12.).



Σχήμα 5.11.: Σύστημα προστασίας διαλυμάτων

5.6. Συστήματα Ελέγχου

Λόγω πολυπλοκότητας του συστήματος θεωρήθηκε απαραίτητο να εισέλθουν κάποια εξαρτήματα για την ασφάλεια των κινήσεων και την ακρίβεια της διαδικασίας.

Αρχικά, κρίσιμο σημείο για τη σωστή λειτουργία της συσκευής είναι η αναγνώριση του σταδίου στο οποίο βρίσκεται η εκτελούμενη διαδικασία. Είναι, δηλαδή, ανάγκη τόσο του χρήστη, όσο και του λογισμικού που θέτει σε λειτουργία τη συσκευή, να μπορεί να προσδιορίσει τη θέση της κασετίνας, δηλαδή το διάλυμα στο οποίο είναι βυθισμένη. Για το σκοπό αυτό χρησιμοποιείται στον άξονα του περιστρεφόμενου δίσκου κωδικοποιητής 12 θέσεων, όπως έχει ήδη αναφερθεί, για την ενημέρωση του προγράμματος. Έπειτα, για την ενημέρωση του χρήστη, υπάρχει οθόνη υγρών κρυστάλλων στην οποία αναγράφεται ακριβώς ποιο είναι το στάδιο της διαδικασίας, για παράδειγμα «ETHANOL 70%» (αιθανόλη 70%), σε ποιο δοχείο βρίσκεται, για παράδειγμα «BEAKER 3», πόσος χρόνος μένει για το συγκεκριμένο δοχείο σε λεπτά και δευτερόλεπτα και πόσος χρόνος έχει περάσει συνολικά από την αρχή της διαδικασίας, και σε ώρες, λεπτά και δευτερόλεπτα.



Στη συνέχεια, οι μηχανικοί διακόπτες που βιδώθηκαν στον κάθετο βραχίονα, δίνουν τη δυνατότητα να γνωρίζει το σύστημα τα όρια της γραμμικής κάθετης κίνησης ώστε να σταματάει ο κινητήρας. Ενώ για την περιστροφή του βραχίονα, οι τρεις φωτοдиодοι, που φαίνονται και στο **Σχήμα 5.10.**, γνωστοποιεί στο σύστημα τις τρεις επιθυμητές θέσεις που θέλουμε να σταματήσει όταν πρέπει. Στο έδρανο ολίσθησης λοιπόν, στηρίχθηκε μία μικρή πλάκα με όνομα αρχείου Solidworks “syndesi_fwtdiodwn”, η οποία κινείται μαζί με το έδρανο και όταν περάσει ανάμεσα από τη φωτοдиодο διακόπτει την ακτίνα και έτσι γίνεται γνωστό ότι τη συγκεκριμένη χρονική στιγμή ο βραχίονας έχει φτάσει στη συγκεκριμένη θέση.

Τέλος, μείζον ζήτημα για τη διαδικασία είναι η σταθερή θερμοκρασία των θερμαινόμενων δοχείων. Η θερμοκρασία πρέπει να παραμένει σταθερή καθόλη τη διάρκεια του συγκεκριμένου σταδίου στους 58 με 60 °C χωρίς να ξεφεύγει από αυτά τα όρια. Με έναν αδιάβροχο αισθητήρα θερμοκρασίας που τοποθετήθηκε στο εξωτερικό του δοχείου κάθε στιγμή γνωστοποιείται η συγκεκριμένη θερμοκρασία. Επιπλέον, η θερμοκρασία αυτή γίνεται γνωστή και στον χρήστη με τη βοήθεια της οθόνης υγρών κρυστάλλων. Προκειμένου να αποφασιστεί σε ποιες ακριβώς θερμοκρασίες έπρεπε να κλείνουν και να ξανανοίγουν, ενδεχομένως, οι θερμαντήρες, έγιναν αρκετά πειράματα ώστε να φτάσουμε στο επιθυμητό αποτέλεσμα.

6. Κατασκευή

Για την κατασκευή του συστήματος ιστοκινέτας, όπως έχει ήδη αναφερθεί, βασική προϋπόθεση ήταν να γίνουν όσο το δυνατόν περισσότερα κομμάτια δύο διαστάσεων για την πιο εύκολη μαζική παραγωγή. Αυτό βέβαια δεν σημαίνει ότι αποφεύχθηκαν τελείως τα κομμάτια τριών διαστάσεων ή των κομματιών που ενώ κόπηκαν σε δύο διαστάσεις έπειτα έπρεπε να υποστούν κάποια 3D κατεργασία όπως το φρεζάρισμα.

6.1. Κομμάτια και Κατεργασίες Δύο Διαστάσεων

Τα κομμάτια εκείνα που έγιναν με κατεργασία δύο διαστάσεων πραγματοποιήθηκαν με υδροκοπή σε μηχανουργείο έξω από το εργαστήριο της σχολής. Η υδροκοπή είναι μία σχετικά φθηνή κατεργασία, η οποία ανάλογα με την ποιότητά της έχει και τα ανάλογα αποτελέσματα. Γι' αυτό τον λόγο, στον σχεδιασμό λήφθηκε υπόψιν, για τις διαστάσεις, το πιθανό σφάλμα της κοπής. Έτσι, αποφεύχθηκαν να κοπούν κομμάτια σε υδροκοπή που θα είχαν μεγάλη ακρίβεια. Αυτός ήταν και ένας λόγος για τον οποίο η συναρμολόγηση μεταξύ των περισσότερων κομματιών έγινε με βίδες και περικόχλια, έτσι ώστε να μπορούν να σχεδιαστούν τρύπες μεγαλύτερες από την διάσταση των βιδών ώστε να περνούν από αυτές με ευκολία, παρά το μεγάλο σφάλμα που μπορεί να έχει καμιά φορά η υδροκοπή στα αποτελέσματά της.

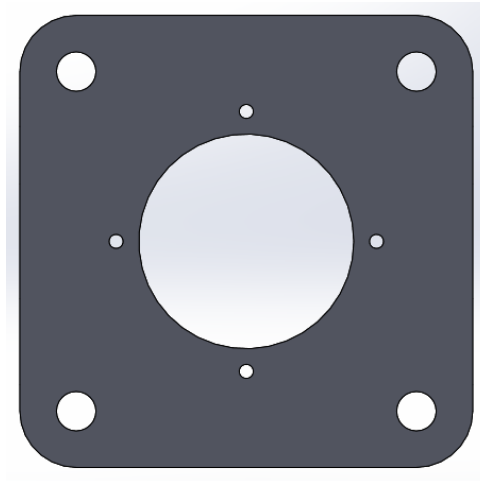
Εκτός από την υδροκοπή, άλλη κατεργασία που χρησιμοποιήθηκε σε μερικά από τα δύο διαστάσεων κομμάτια, ήταν η κοπή με λέιζερ και το στραντζάρισμα. Κάποια στρανταρίσματα τα ανέλαβε το ίδιο μηχανουργείο που έκανε και την υδροκοπή και κάποια άλλα έγιναν στο εργαστήριο της σχολής με εξοπλισμό του εργαστηρίου. Οι κοπές με ακτίνες λέιζερ έγιναν στο εργαστήριο.

Στη συνέχεια θα επεξηγηθεί ο τρόπος κατασκευής όλων των κομματιών δύο διαστάσεων χωρισμένα με βάση το σύστημα στο οποίο ανήκουν.

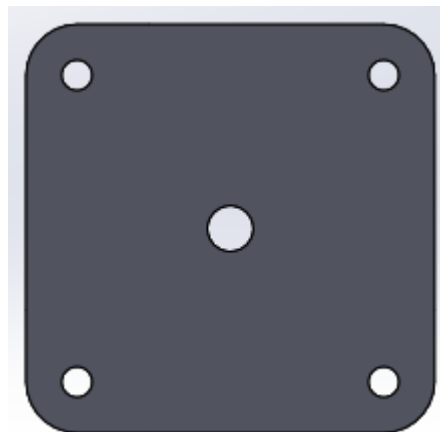
Ξεκινώντας από το σύστημα περιστροφής των δοχείων:

- Ο «diskos_katw» έγινε με υδροκοπή και το υλικό του είναι αλουμίνιο (Σχήμα 5.1.). Σκοπός του είναι να στηρίζονται και να παρασύρονται μαζί του στην περιστροφική κίνηση τα δοχεία των διαλυμάτων.
- Ο «diskos_panw» έγινε με υδροκοπή και το υλικό του είναι αλουμίνιο (Σχήμα 5.2.). Σκοπός του είναι η δημιουργία πατούρας, δηλαδή σκαλοπατιού προκειμένου να υπάρχει μεγαλύτερη ασφάλεια στο χείλος των δοχείων.

- Η «vasi_gia_housing» έγινε με υδροκοπή και το υλικό της είναι αλουμίνιο (Σχήμα 6.1). Σκοπός της είναι να βιδωθεί σε αυτή και να συγκρατηθεί το σύστημα με τα δύο ρουλεμάν και το κέλυφός τους.
- Η «endiamesi_sygkratysi» έγινε με υδροκοπή και το υλικό της είναι αλουμίνιο (Σχήμα 6.2). Σκοπός της είναι να οδηγήσει τους σπειρωτομημένους άξονες που χρησιμοποιήθηκαν και να τους συγκρατήσει με τη βοήθεια περικοχλίων καθώς οι άξονες αυτοί έχουν μεγάλο μήκος.



Σχήμα 6.1.: Βάση για δέσιμο του κελύφους με τα ρουλεμάν



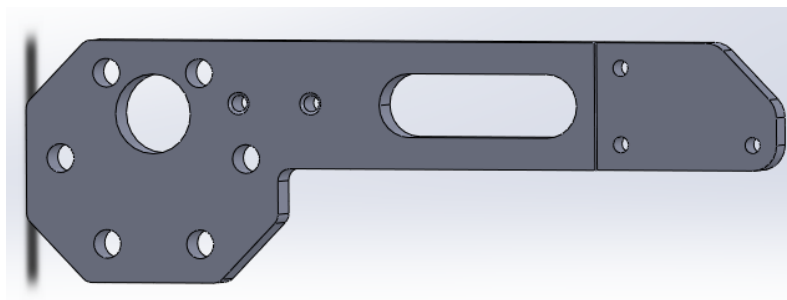
Σχήμα 6.2.: Ενδιάμεση συγκράτηση για σπειροτομημένους άξονες.

Από το σύστημα γραμμικής κίνησης:

- Το «asanser» έγινε αρχικά με υδροκοπή και έπειτα στραντζάρισμα 90 μοιρών (Σχήμα 5.7.). Το υλικό του είναι ανοξείδωτος χάλυβας για καλύτερη συμπεριφορά στα χημικά και για να είναι πιο στιβαρό, ενώ έχει κοπεί και αλουμίνιο για το ίδιο κομμάτι. Επειδή το πάχος του είναι ενός χιλιοστού οπότε είναι εύθραυστο και επειδή μπαينوβγαίνει μια κασετίνα κάθε φορά που

γίνεται μία διαφορετική εφαρμογή, θεωρήθηκε απαραίτητο να γίνουν κάποια κομμάτια επιπλέον σε περίπτωση ατυχήματος. Έτσι, εκτός από το κομμάτι του ανοξείδωτου χάλυβα που ήδη είναι συνδεδεμένο στη συσκευή, υπάρχει άλλο ένα από ανοξείδωτο χάλυβα και άλλα δύο από αλουμίνιο, σε απόθεμα. Σκοπός του είναι να υποδεχτεί την κασετίνα με τον ιστό.

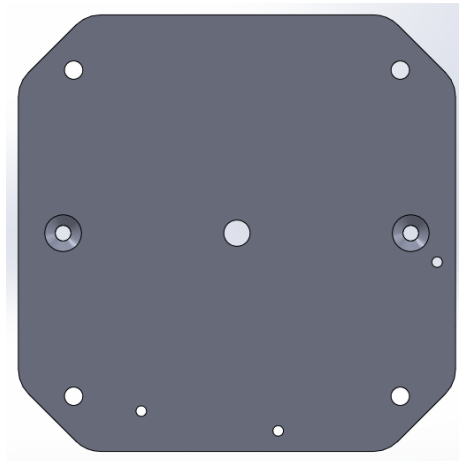
- Η «stiriksi_asanser» έγινε με υδροκοπή και το υλικό του είναι αλουμίνιο (**Σχήμα 6.3.**). Στη συνέχεια έγινε κατεργασία στη φρέζα όπως θα αναφερθεί και στο υποκεφάλαιο 6.2.. Σκοπός της είναι να μεσολαβήσει μεταξύ του συστήματος κάθετης κίνησης και της συγκράτησης της κασετίνας.



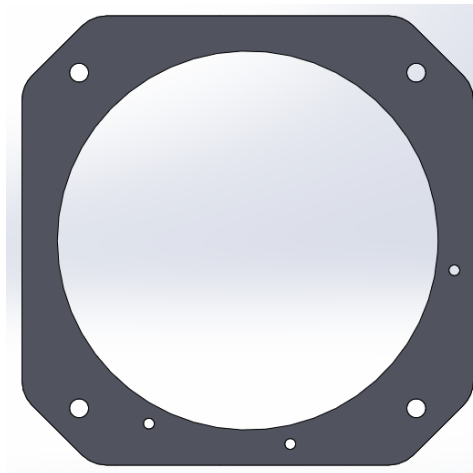
Σχήμα 6.3.: Στήριξη για την υποδοχή της κασετίνας

Από το σύστημα περιστροφής βραχίονα:

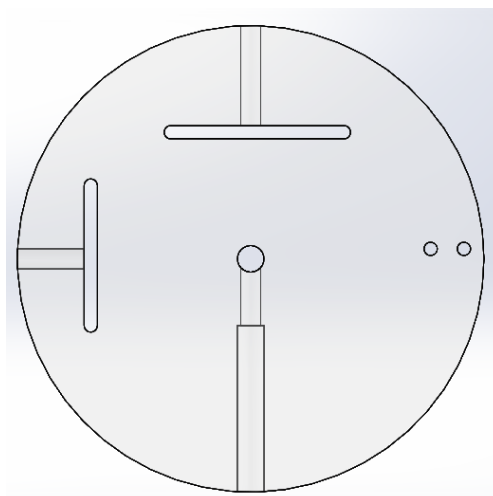
- Η «plaka_stiriksis_kinitira» έγινε με υδροκοπή και το υλικό της είναι αλουμίνιο (**Σχήμα 6.4.**). Σκοπός της είναι να βιδωθεί σε αυτή βίδα με την οποία στηρίζεται ο βηματικός κινητήρας.
- Η «vasi_kinit» έγινε με υδροκοπή και το υλικό της είναι αλουμίνιο (**Σχήμα 6.5.**). Σκοπός της είναι να δημιουργήσει το σωστό επίπεδο για να τοποθετηθούν οι φωτοαισθητήρες και να βιδωθούν αλλά και να αφήσει χώρο στο έδρανο ολίσθησης να περιστρέφεται.
- Το «edrano_olisthisis» έγινε με κοπή σε λέιζερ και το υλικό του είναι από πλαστικό (plexi-glass) (**Σχήμα 6.6.**). Στο λέιζερ έγινε το περίγραμμα, η κεντρική οπή, οι δύο μικρές οπές και τα δύο αυλάκια για να τοποθετηθεί ο κάθετος βραχίονας. Οι υπόλοιπες κατεργασίες αναφέρονται στο επόμενο υποκεφάλαιο (6.2.) γιατί θεωρούνται κατεργασίες τριών διαστάσεων. Σκοπός του είναι να σφινωθεί σε αυτό ο κάθετος βραχίονας (και ο κινητήρας) ώστε να του δοθεί η δυνατότητα να περιστρέφεται μαζί με τον κινητήρα.
- Ο «kathetos_vraxionas» έγινε με υδροκοπή και το υλικό του είναι αλουμίνιο (**Σχήμα 6.7.**). Έπειτα, έγινε στραντζάρισμα του τεμαχίου σε 90 μοίρες. Σκοπός του είναι ο συγκράτηση όλου του συστήματος κάθετης κίνησης και η σύνδεσή του με το σύστημα περιστροφής του βραχίονα.



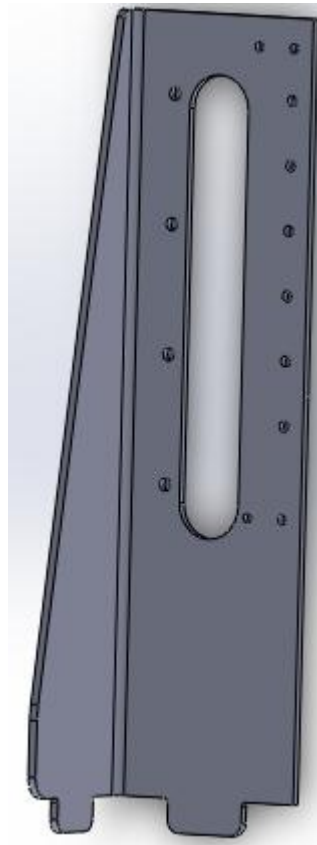
Σχήμα 6.4.: Πλάκα στήριξης βηματικού κινητήρα



Σχήμα 6.5.: Βάση κινητήρα για υποδοχή φωτοαισθητήρων



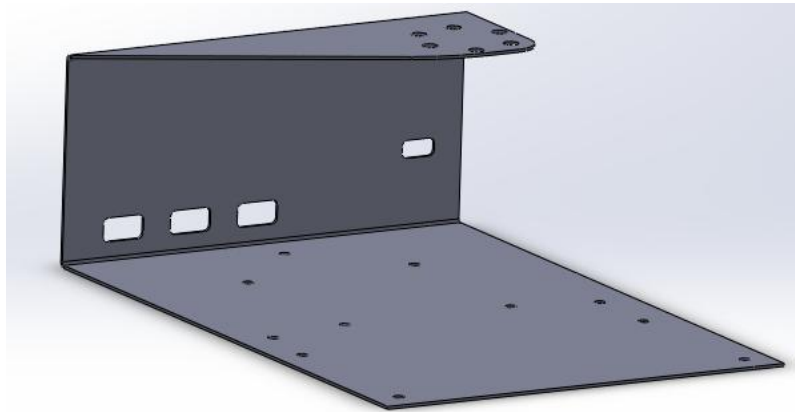
Σχήμα 6.6.: Έδρανο ολίσθησης



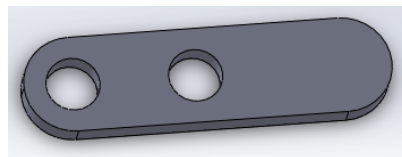
Σχήμα 6.7.: Κάθετος βραχίονας

Από το σύστημα θερμαινόμενων δοχείων:

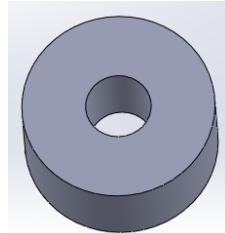
- Η «vasi_thermainomenwn» έγινε με υδροκοπή και το υλικό της είναι αλουμίνιο (Σχήμα 6.8.). Στην συνέχεια ακολούθησαν δύο στρατζαρίσματα σε 90 μοίρες το καθένα. Σκοπός της είναι να στηρίζει τον κινητήρα των δίσκων, το σύστημα του βραχίονα αλλά και το μονωτικό κουτί για τα θερμαινόμενα δοχεία.
- Η «syndesi_fwtoiodwn» έγινε με υδροκοπή και το υλικό της είναι αλουμίνιο (Σχήμα 6.9.). Σκοπό της είναι να διακόπτει την ακτίνα των φωτοαισθητήρων. Πολλές φορές αναφέρεται και ως “dog”.
- Οι «apostates» έγιναν με ακτίνες λέιζερ και το υλικό τους είναι πλαστικό (plexi-glass) (Σχήμα 6.10.). Σκοπός τους είναι να διατηρήσουν την σωστή και σταθερή απόσταση μεταξύ της βάσης των θερμαινόμενων και της βάσης του συστήματος ώστε να χωρέσουν και μερικά περικόχλια που χρειάζονται για τη σύνδεση κάποιων τεμαχίων.
- Ο «apost_phot» έγινε με κοπή σε λέιζερ και το υλικό του είναι πλαστικό (plexi-glass) (Σχήμα 6.11.). Σκοπός του είναι να δημιουργήσει την κατάλληλη απόσταση μεταξύ εδράνου ολίσθησης και σύνδεσης φωτοδιόδων έτσι ώστε να μπορεί να διακόπτεται η ακτίνα.



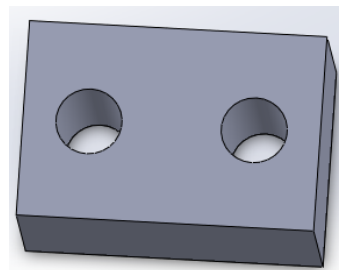
Σχήμα 6.8.: Βάση για θερμαινόμενα δοχεία και στήριξη κινητήρα δίσκων



Σχήμα 6.9.: Σύνδεση φωτοδιόδων για διακοπή ακτίνας



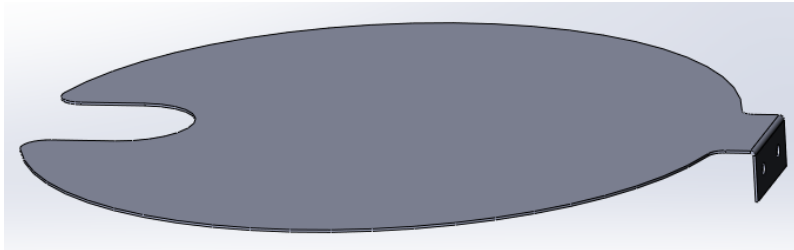
Σχήμα 6.10.: Αποστάτες για τη βάση θερμαινόμενων δοχείων



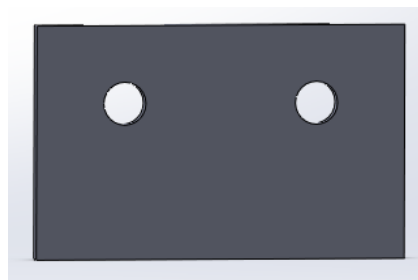
Σχήμα 6.11.: Αποστάτης για τη σύνδεση φωτοδιόδων

Από το σύστημα προφύλαξης διαλυμάτων:

- Το «ΚΑΡΑΚΙ» έγινε με υδροκοπή και το υλικό του είναι αλουμίνιο (Σχήμα 6.12.). Σκοπός του είναι να προστατεύει τα διαλύματα από σκουπίδια, σκόνη και να αποτρέπει την γρήγορη εξάτμισή τους.
- Η «Plaka_desimatos_mentese» έγινε με υδροκοπή και το υλικό της είναι αλουμίνιο (Σχήμα 6.13.). Σκοπός της είναι να σφίξει ο μεντεσές και το καπάκι, με βίδες, πάνω στον σωλήνα ορθογωνικής διατομής και στα πεπλατυσμένα αξονάκια.



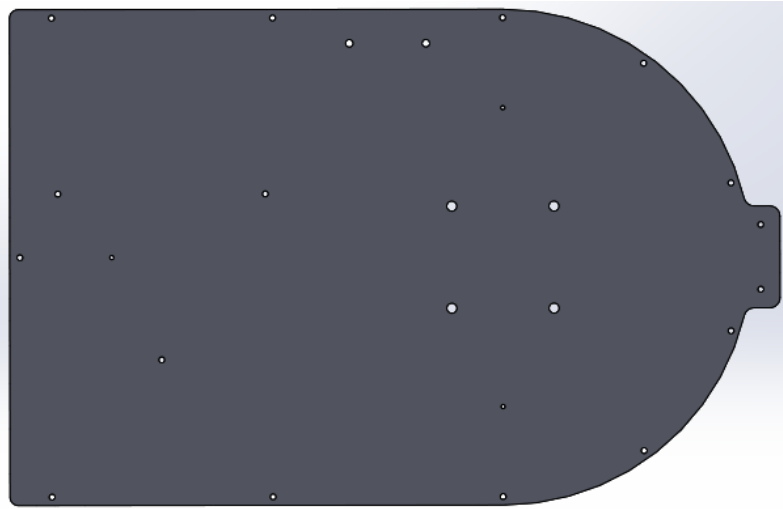
Σχήμα 6.12.: Καπάκι για προστασία διαλυμάτων



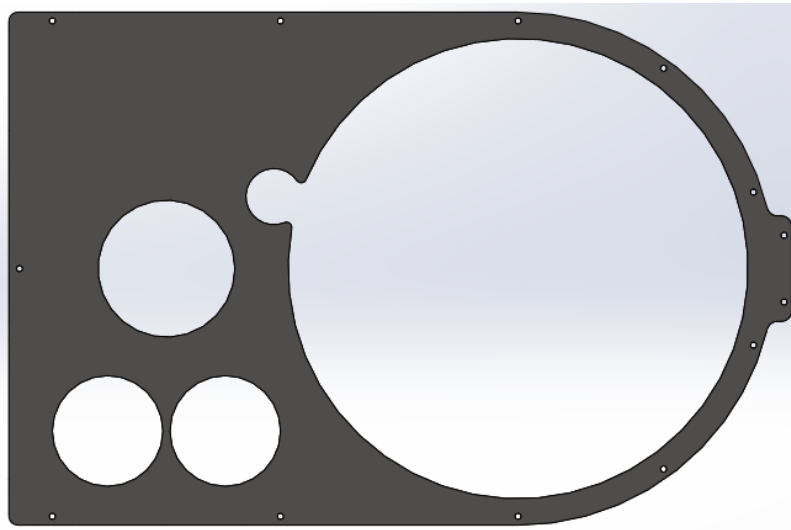
Σχήμα 6.13.: Πλάκα για σφίξιμο του μεντεσέ

Από τα όρια συστήματος:

- Η «vasi_systimatos» έγινε με υδροκοπή και το υλικό της είναι αλουμίνιο (Σχήμα 6.14.). Σκοπός της είναι να δημιουργήσει το επίπεδο πάνω στο οποίο θα βρίσκονται όλα τα συστήματα. Επίσης σε αυτή τη βάση τοποθετήθηκαν 3 ποδαράκια για να στέκεται όλη η συσκευή.
- Το «kalymma_systimatos» έγινε με υδροκοπή και το υλικό του είναι ανοξείδωτος χάλυβας με την όψη ενός καθρέφτη (Σχήμα 6.15.). Σκοπός του είναι να προστατεύσει τα ηλεκτρονικά και τα καλώδια από διαλύματα που μπορεί να πέσουν κατά λάθος αλλά και να έχει η συσκευή ένα καλύτερο αισθητικό αποτέλεσμα.



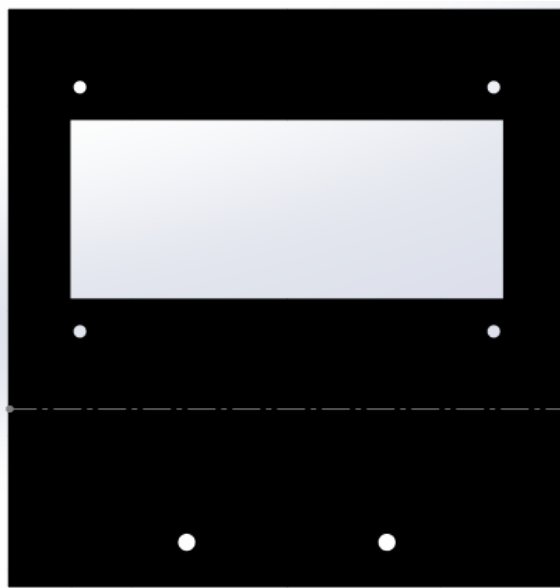
Σχήμα 6.14.: Βάση συστήματος



Σχήμα 6.15.: Κάλυμμα συστήματος για προστασία

Από το σύστημα LCD:

- Η «thiki_LCD» έγινε με κοπή σε λέιζερ και το υλικό της είναι πλαστικό (plexi-glass) μαύρου χρώματος για να έχει καλύτερο αισθητικό αποτέλεσμα . Στη συνέχεια έγινε στραντζάρισμα σε 90 μοίρες. Σκοπός της είναι να συγκρατήσει την οθόνη υγρών κρυστάλλων ώστε να πληροφορείται ο χρήστης για την διαδικασία κατά τη διάρκειά της. Στο **Σχήμα 6.16.** φαίνεται το κομμάτι ανοιχτό, χωρίς να έχει υποστεί το στραντζάρισμα.



Σχήμα 6.16.: Θήκη για τοποθέτηση οθόνης

6.2. Κομμάτια και Κατεργασίες Τριών Διαστάσεων

Τα κομμάτια εκείνα που έγιναν με κατεργασία τριών διαστάσεων πραγματοποιήθηκαν με συμβατικό τόρνο, συμβατική φρέζα, φρέζα-cnc, πρίονι, λειαντικός τροχός, τρυπάνι και σπειροτόμο στο μηχανουργείο του εργαστηρίου της σχολής. Όλες αυτές οι εργαλειομηχανές και τα εργαλεία έχουν πολύ καλή ακρίβεια και γι' αυτό χρησιμοποιήθηκαν κάποιες φορές, όπως θα διαπιστωθεί και στη συνέχεια, για να πραγματοποιηθούν καλύτερες επιφάνειες για συνεργασία.

Για την κατεργασία σε φρέζα-cnc χρησιμοποιήθηκε το πρόγραμμα Solid-Cam σε συνεργασία με το Solidworks. Εκεί, αφού σχεδιαστεί το κομμάτι, του δίνονται οι αντίστοιχες εντολές με τις κατεργασίες που πρέπει να γίνουν και δημιουργείται αυτόματα ο G-κώδικας για τη cnc.

Σε όλες τις εργαλειομηχανές και τα εργαλεία που χρησιμοποιήθηκαν σημαντικό ρόλο διαδραμάτισε ο τρόπος στήριξης του τεμαχίου (τρόπος δεσίματος), ο οποίος κάποιες φορές μπορεί να γίνει πολύ δύσκολος για συγκεκριμένο συνδιασμό τεμαχίου, εργαλείου και κατεργασίας. Γι' αυτό το λόγο κατά το σχεδιασμό έπρεπε να ληφθεί υπόψιν και ο τρόπος δεσίματος του τεμαχίου στη μέγγενη ώστε να καταστεί δυνατή η κατεργασία.

Επίσης, στα τεμάχια εκείνα που έπρεπε να ανοιχτούν σπειρώματα έπρεπε να ληφθεί υπόψιν, κατά την υδροκοπή ή την κοπή σε λείζερ, ότι οι οπές εκείνες έπρεπε να γίνουν μικρότερες για να ανοιχτούν, σε

αντίθεση με τις περαστές οι οποίες, όπως αναφέρθηκε σε προηγούμενο κεφάλαιο έπρεπε να γίνουν λίγο μεγαλύτερες από τη διάμετρο της βίδας για να χωράει χωρίς πρόβλημα.

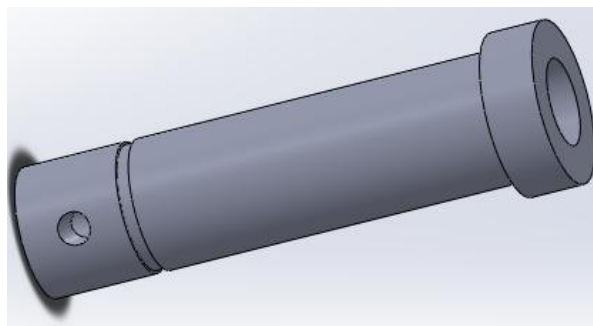
Στη συνέχεια θα επεξηγηθεί ο τρόπος κατασκευής όλων των κομματιών τριών διαστάσεων χωρισμένα με βάση το σύστημα στο οποίο ανήκουν, όπως και στο προηγούμενο υποκεφάλαιο.

Ξεκινώντας από το σύστημα περιστροφής των δοχείων:

- Η «ntiza_M6» είναι στην ουσία σπειροτομημένος άξονας (ή ατέρμονη βίδα ή ντίζα) όπως αυτός του **Σχήματος 6.17.** που αγοράστηκε έτοιμος σε μεγάλο μήκος και κόπηκε στο πριόνι στο επιθυμητό μήκος. Η λείανσή του στις άκρες έγινε στον λειαντικό τροχό. Σκοπός της είναι να στηρίζει όλο το σύστημα των ρουλεμάν, τους δίσκους και τον κωδικοποιητή.
- Το «aksonaki_housing» έγινε σε συμβατικό τόρνο και το υλικό του είναι αλουμίνιο (**Σχήμα 6.18.**). Πρώτα κόπηκε άξονας Φ15 στο πριόνι, σε μήκος λίγο μεγαλύτερο από το επιθυμητό ώστε να κοπεί το πρόσωπο στον τόρνο και το μήκος να έχει μεγαλύτερη ακρίβεια. Στον τόρνο έγιναν και οι υπόλοιπες διαμορφώσεις εκτός από την οπή και το σπείρωμα για την σκουληκόβίδα που έγιναν με τρυπάνι και με σπειροτόμο αντίστοιχα. Σκοπός του είναι να περιστρέφει τον κωδικοποιητή μαζί με τους δίσκους και να στηρίζει τη φωλιά με τα ρουλεμάν με τη βοήθεια δακτυλίου.



Σχήμα 6.17.: Ατέρμονη βίδα (ντίζα)



Σχήμα 6.18.: Άξονας για περιστροφή δίσκων

Από το σύστημα γραμμικής κίνησης:

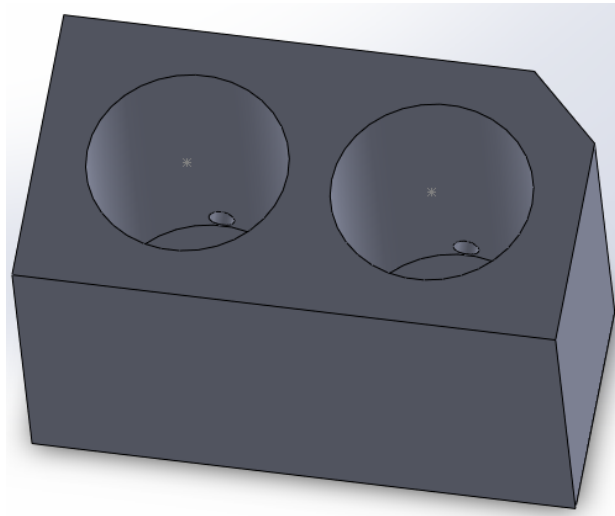
- Η «stiriksi_asanser» υπέστει, μετά την υδροκοπή, όπως αναφέρθηκε, και μια κατεργασία σε φρέζα-cnc ώστε να δημιουργηθεί το άνισο επίπεδο που φαίνεται στο **Σχήμα 6.3.**, στο οποίο θα εφαρμόσει η υποδοχή της κασετίνας (ασανσέρ).

Από το σύστημα περιστροφής βραχίονα:

- Στο «edrano_olisthisis» (**Σχήμα 6.6.**) μετά την υδροκοπή, όπως έχει αναφερθεί, χρειάστηκαν να γίνουν κάποιες οπές περιμετρικά με σπειρώματα, για τη σύσφιξη του βραχίονα και του κινητήρα στο τεμάχιο με σκουλικόβιδες, με τρυπάνι και σπειροτόμο.
- Στη «stiriksi_kinitira» (**Σχήμα 6.4.**), μετά την υδροκοπή, χρειάστηκε να γίνει μια λοξοτόμηση (chamfer) στις δύο τρύπες που θα υποδεχτούν τη βίδα για να στηριχτεί ο κινητήρας. Επειδή αυτή η βίδα δεν έπρεπε να εξέχει από το επίπεδο της πλάκας, αφού θα μπει από πάνω η «vasi_kinit» επιλέχθηκε η κεφαλή της να είναι λοξή (ή αλλιώς φρεζάτη) και γι'αυτό χρειάστηκε η λοξοτόμηση της τρύπας.
- Στη «vasi_kinit» (**Σχήμα 6.5.**), μετά την υδροκοπή, ανοίχτηκε η κεντρική οπή στον τόρνο ώστε να είναι καλύτερης ποιότητας η επιφάνεια επαφής με το έδρανο ολίσθησης.

Από το σύστημα θερμαινόμενων δοχείων:

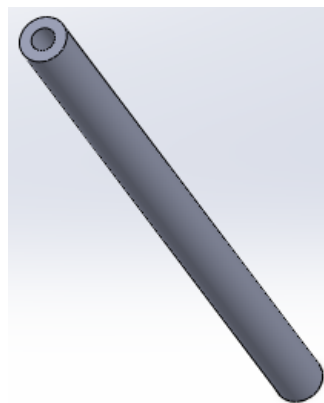
- Η «vasi_gia_inox» (**Σχήμα 6.19.**) έγινε από πλάκες ανθρακικού ασβεστίου οι οποίες κόπηκαν σε πριόνι στις επιθυμητές διαστάσεις και διαμορφώσεις, καθώς το υλικό αυτό θα έσπαγε εάν τοποθετούνταν στη φρέζα. Οι πλάκες αυτές στη συνέχεια κολλήθηκαν μεταξύ τους με δυνατή κόλλα και στο τέλος συγκρατήθηκαν με πολλές στρώσεις ταινίας θερμοαντοχής. Τέλος, με τρυπάνι ανοίχθηκαν οπές για να περαστούν τα καλώδια των θερμοαντήρων και για να σταθούν οι αισθητήρες θερμοκρασίας.



Σχήμα 6.18.: Μονωτικό περίβλημα για θερμαινόμενα δοχεία

Από τα όρια συστήματος:

- Το «kolonaki» είναι από αλουμίνιο (Σχήμα 6.19.). Αγοράστηκε άξονας μεγάλου μήκους διαμέτρου Φ6, κόπηκε σε λίγο μεγαλύτερο μήκος από το επιθυμητό σε πριόνι, κατεργάστηκαν τα πρόσωπα στον συμβατικό τόρνο, έπειτα έγιναν οι οπές και πάλι στον τόρνο και τέλος δημιουργήθηκαν τα σπειρώματα με τον σπειροτόμο. Σκοπός των 11 αξόνων αυτών είναι να στηρίζουν το κάλυμμα του συστήματος.

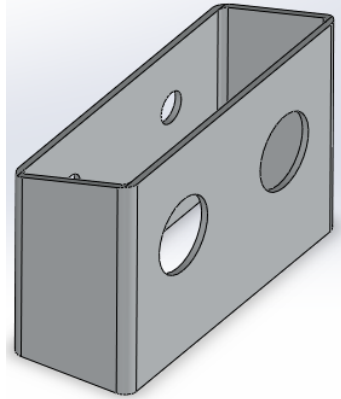


Σχήμα 6.19.: Άξονες για στήριξη καλύμματος

Από το σύστημα προφύλαξης διαλυμάτων:

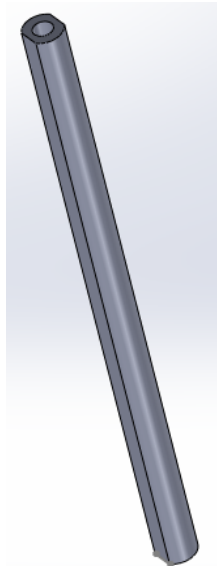
- Ο «swlinas_orthogwnikis_diatomis», είναι από αλουμίνιο, αγοράστηκε έτοιμος και έπειτα κόπηκε υπό γωνία με τη βοήθεια μέγγενης στο πριόνι (Σχήμα 6.20.). Στη συνέχεια με τρυπάνι ανοίχτηκαν δύο μικρές οπές για να στερεωθεί με κοχλία ο μεντεσές και δύο πιο μεγάλες από

την άλλη μεριά ώστε να χωράει να περάσει εργαλείο που θα σφίξει την βίδα. Σκοπός του είναι να στηρίζει το καπάκι όταν είναι ανοιχτό ώστε να μην πέφτει κάτω.



Σχήμα 6.20.: Σωλήνας ορθογωνικής διατομής κομμένος υπό γωνία

- Η «stiriksi_mentese» (**Σχήμα 6.21.**) προέρχεται από άξονα αλουμινίου και υπέστη τις ίδιες κατεργασίες με εκείνες του τεμαχίου «kolonaki» μόνο που έγινε και η παραλληλία των δύο πλευρών με συμβατική φρέζα. Σκοπός τους είναι να δημιουργήσουν τα παράλληλα επίπεδα ώστε να μπορεί να στηριχθεί το σύστημα της προστασίας διαλυμάτων και να σφίξει ο μεντεσές με την πλάκα μεντεσέ.

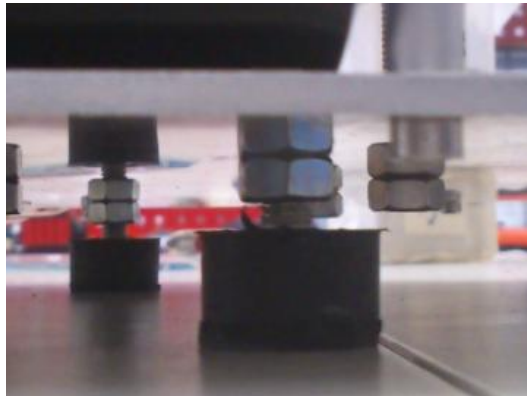


Σχήμα 6.21.: Σωλήνας στήριξης μεντεσέ

6.3. Συναρμολόγηση

Η συναρμολόγηση της συσκευής είναι το πιο βασικό κομμάτι μιας κατασκευής καθώς μπορεί να φέρει στην επιφάνεια λάθη που δεν ήταν δυνατό να φανούν ή να κατανοηθούν. Σκοπός είναι η ιδέα να γίνει πραγματικότητα και ο τρόπος σύνδεσης των κομματιών να κάνουν λειτουργική τη συσκευή.

Αρχικά, τοποθετήθηκαν στη μεγάλη βάση του συστήματος (Σχήμα 6.14.) τα τρία ποδαράκια για τη στήριξη ολόκληρης της συσκευής (Σχήμα 6.22.). Έπειτα, βιδώθηκαν οι δίσκοι μεταξύ τους και συναρμολογήθηκε το σύστημα συγκράτησης των δίσκων, δηλαδή η φωλιά με τα δύο ρουλεμάν, η βάση για τη φωλιά (housing), η ενδιάμεση συγκράτηση, ο κωδικοποιητής, ο άξονας και η αντίστοιχη σκουληκόβιδα, οι σπειροτομημένοι άξονας και τα αντίστοιχα περικόχλια που χρειάστηκαν, όπως φαίνεται στο Σχήμα 6.23..

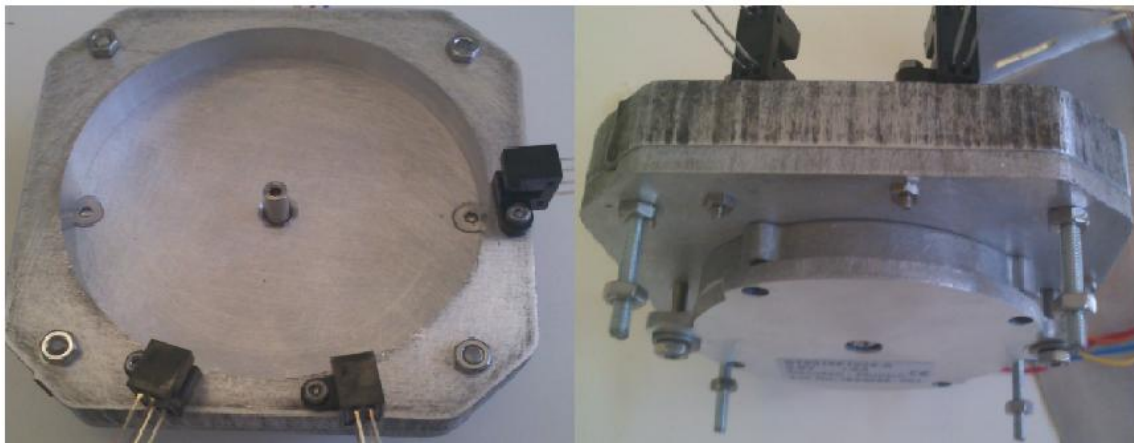


Σχήμα 6.22.: Ποδαράκια βάσης

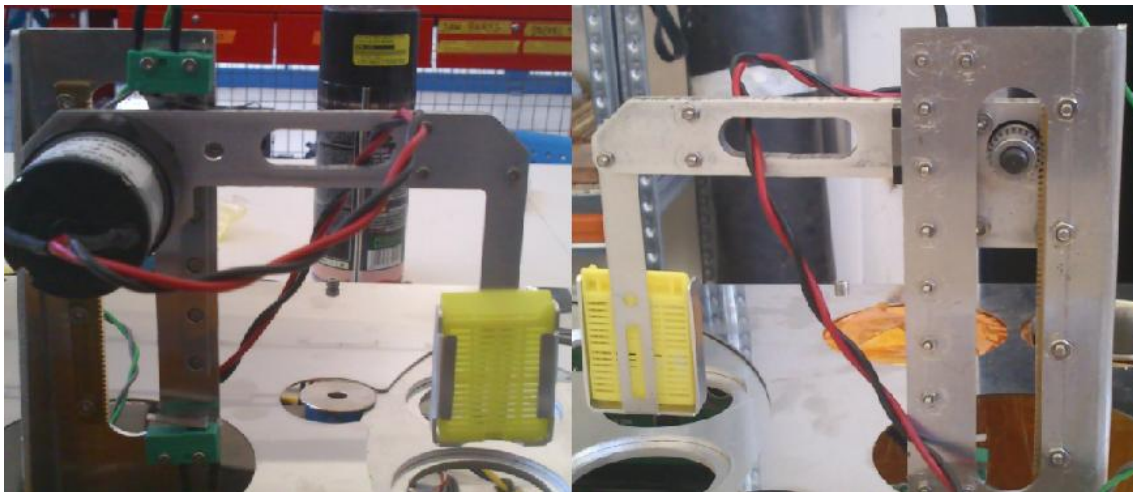


Σχήμα 6.23.: Συναρμολογημένο σύστημα στήριξης δίσκων

Στη συνέχεια, συναρμολογήθηκαν ξεχωριστά το σύστημα του βηματικού κινητήρα και το σύστημα του βραχίονα. Στο πρώτο συνδέθηκε πρώτα ο κινητήρας στη στήριξη κινητήρα και μετά οι φωτοαισθητήρες στη βάση του κινητήρα όπως φαίνεται στο **Σχήμα 6.24.** Στο δεύτερο συνδέθηκε πρώτα το ρουλεμάν γραμμικής κίνησης στη στήριξη του «ασανσέρ» με φρεζάτες βίδες, έπειτα ο κινητήρας με πέντε από βίδες (από τις έξι) και το ασανσέρ. Μετά, βιδώθηκε ο κανόνας και ο διάδρομος με το ρουλεμάν γραμμικής κίνησης πάνω στον κάθετο βραχίονα και τέλος βιδώθηκαν οι αποστάτες με τους διακόπτες και το γρανάζι με την σκουληκόβιδα, όπως φαίνεται στο **Σχήμα 6.25.**



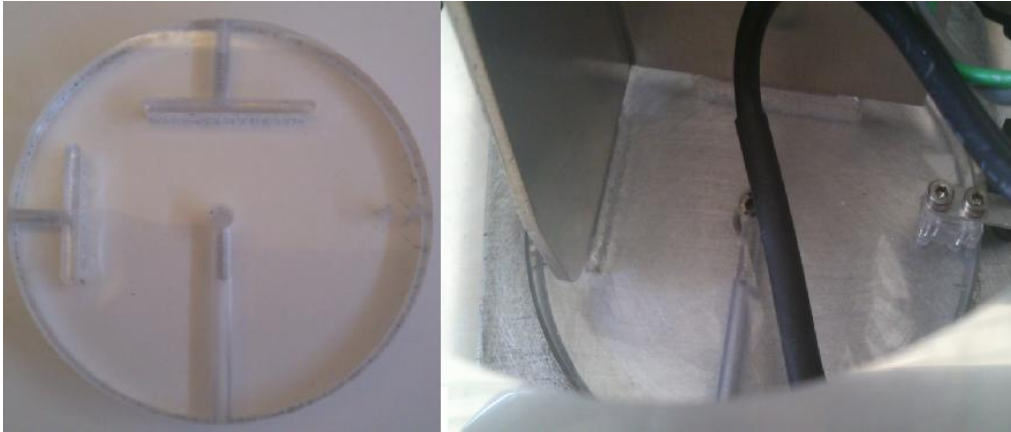
Σχήμα 6.24.: Συναρμολογημένο σύστημα βηματικού κινητήρα



Σχήμα 6.25.: Συναρμολογημένο σύστημα βραχίονα

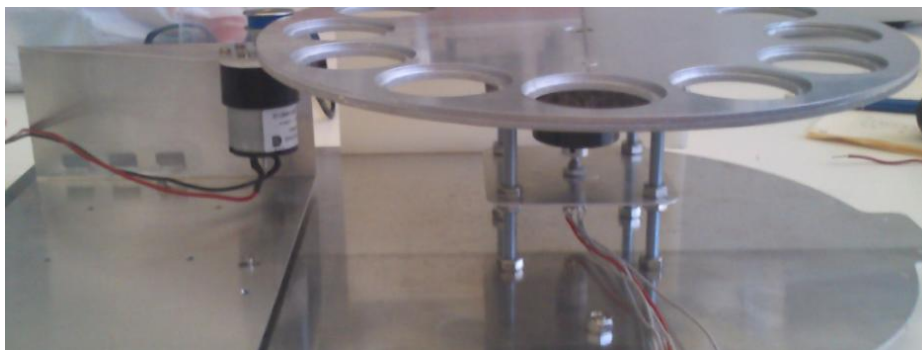
Τα δύο παραπάνω συστήματα ενώνονται μεταξύ τους με το έδρανο ολίσθησης, στο οποίο έχει τοποθετηθεί ο αποστάτης και η σύνδεση με τις φωτοдиодους, και σκουληκόβιδες, όπως φαίνεται στο

Σχήμα 6.26. Σημειώνεται ότι για να βιδωθούν οι σκουληκόβιδες έπρεπε να ξεβιδωθεί η βάση του κινητήρα (αφού είχε περαστεί από το κάτω μέρος του βραχίονα γιατί μετά δεν θα χωρούσε να περάσει) και οι φωτοαισθητήρες και να ξαναβιδωθούν εκ των υστέρων.



Σχήμα 6.26.: Συναρμολογημένο σύστημα εδράνου ολίσθησης

Στη συνέχεια, τοποθετείται η βάση στην οποία δένεται ο κινητήρας για την περιστροφή των δίσκων και το ελαστομερές, το οποίο πρέπει να βρίσκεται σε επαφή μαζί τους, όπως στο **Σχήμα 6.27.** Σημειώνεται ότι χρησιμοποιήθηκαν αποστάτες με κοχλίες και περικόχλια για να κρατήσουν αυτή τη βάση σε απόσταση από τη μεγάλη βάση του συστήματος.

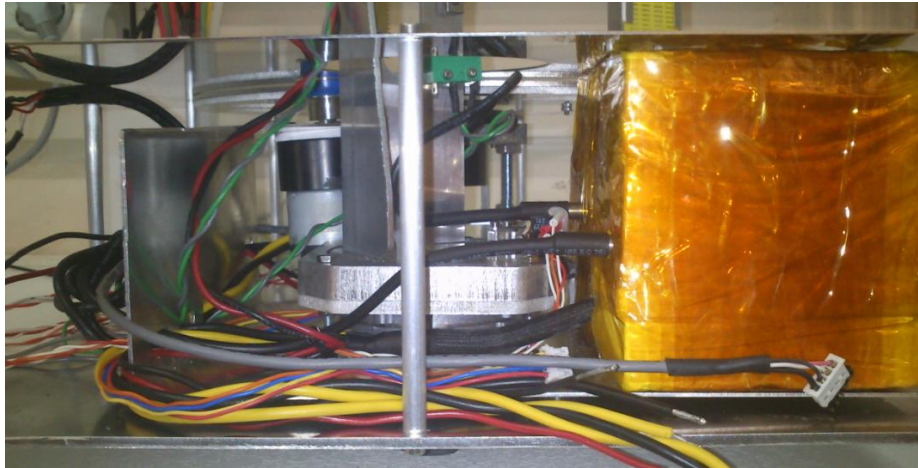


Σχήμα 6.27.: Συναρμολογημένο σύστημα κινητήρα δίσκων

Έπειτα, συναρμολογείται ξεχωριστά το σύστημα των θερμαινόμενων δοχείων τοποθετώντας πάνω από την πρώτη πλάκα τους θερμαντήρες και από πάνω τις άλλες τρεις πλάκες που είναι ενωμένες μεταξύ τους με κόλλα και ταινία θερμοαντοχής. Όλο το «κουτί» αυτό με το μονωτικό υλικό και τους θερμαντήρες δέθηκε όλο μαζί με ταινία θερμοαντοχής και κολλήθηκε με κόλλα στη βάση. Τέλος,

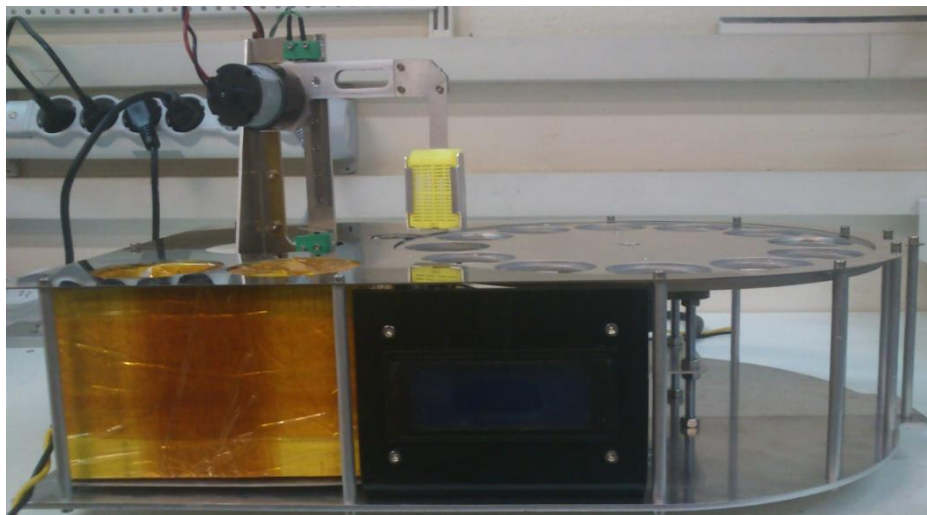
σφινώθηκαν στις ειδικές οπές οι αισθητήρες θερμοκρασίας οι οποίοι θα λαμβάνουν τη θερμοκρασία από το εξωτερικό περίβλημα των δοχείων.

Το σύστημα αυτό με ολόκληρο το σύστημα του βραχίονα φαίνεται στο **Σχήμα 6.28.**



Σχήμα 6.28.: Συναρμολογημένο σύστημα θερμαινόμενων δοχείων και βραχίονα

Στο τέλος βιδώθηκε η οθόνη υγρών κρυστάλλων και τα περιμετρικά ποδαράκια τα οποία στηρίζουν το κάλυμμα του συστήματος. Στο **Σχήμα 6.29.** φαίνεται το τελικό συναρμολογημένο σύστημα ιστοκινέτας.



Σχήμα 6.29.: Συναρμολογημένο σύστημα ιστοκινέτας



7. Περιγραφή και Σχεδιασμός Ηλεκτρονικών Διατάξεων

Για να τεθεί σε λειτουργία η έτοιμη πλέον, κατασκευαστικά, συσκευή χρειάζεται να ξέρει τι πρέπει να κάνει και ποια είναι η διαδικασία που θα πρέπει να ακολουθήσει. Με άλλα λόγια θα πρέπει να προγραμματιστεί. Ένας αρκετά απλός τρόπος είναι μέσω του συστήματος Arduino, το οποίο αποτελεί ένα ανοικτού τύπου σύστημα μοντελοποίησης, δηλαδή ο κώδικάς του είναι προσβάσιμος στο ευρύ κοινό, χωρίς να είναι απαραίτητη κάποια είδους άδεια για την χρησιμοποίησή του, και έχουν την ικανότητα να κατασκευαστούν ηλεκτρονικές διατάξεις που αντιδρούν με το περιβάλλον με τη βοήθεια αισθητήρων, κινητήρων, LEDs ή οθονών. Με άλλα λόγια είναι η σύνδεση του ψηφιακού με του αναλογικού κόσμου δίνοντας απλά τάση (3.3 V ή 5 V) σε pins ψηφιακά ή αναλογικά.

Στην συγκεκριμένη περίπτωση χρησιμοποιήθηκε Arduino Mega ADK (Σχήμα 7.1.), το οποίο δίνει την δυνατότητα ελέγχου 54άρων ψηφιακών pins, τα οποία μπορούν να λειτουργήσουν είτε ως είσοδοι είτε ως έξοδοι, και 16 αναλογικών pins που λειτουργούν είτε σαν τα ψηφιακά (άρα στην ουσία δημιουργούνται 70 pins) είτε ως είσοδοι λαμβάνοντας δεδομένα από αναλογικούς αισθητήρες. Δίνοντας τάση σε αυτά τα pins ο επεξεργαστής ή όχι, τα θέτει σε κατάσταση HIGH και LOW αντίστοιχα.



Σχήμα 7.1.: Arduino Mega

Το «μυαλό» του Arduino είναι ο μικροεπεξεργαστής του, στον οποίο φορτώνεται ο κώδικας μέσω του υπολογιστή, με την θύρα USB που διαθέτει ο Arduino, η οποία του δίνει και ρεύμα (για τάση 5 Volts 500 mA). Ο μικροεπεξεργαστής μπορεί και επαναπρογραμματίζεται, κρατώντας κάθε φορά το τελευταίο πρόγραμμα που έχει φορτωθεί. Η γλώσσα προγραμματισμού που χρησιμοποιείται είναι βασισμένη στις γλώσσες C και C++ και είναι εύκολη στην χρήση και στην εκμάθηση καθώς υπάρχει μια μεγάλη κοινότητα που χρησιμοποιεί το συγκεκριμένο σύστημα και αναρτά στο διαδίκτυο



ολόκληρους κώδικες και βιβλιοθήκες, οι οποίοι είναι πολύ βοηθητικοί, ειδικά για έναν αρχάριο προγραμματιστή.

Με τη βοήθεια λοιπόν, μιας διάτρητης πλάκας (breadboard), καλωδίων, του Arduino, μιας τροφοδοσίας 12 V και ηλεκτρονικές διατάξεις οι οποίες θα αναλυθούν στη συνέχεια, η συσκευή πήρε ζωή και άρχισε να εκπληρώνει τον σκοπό της.

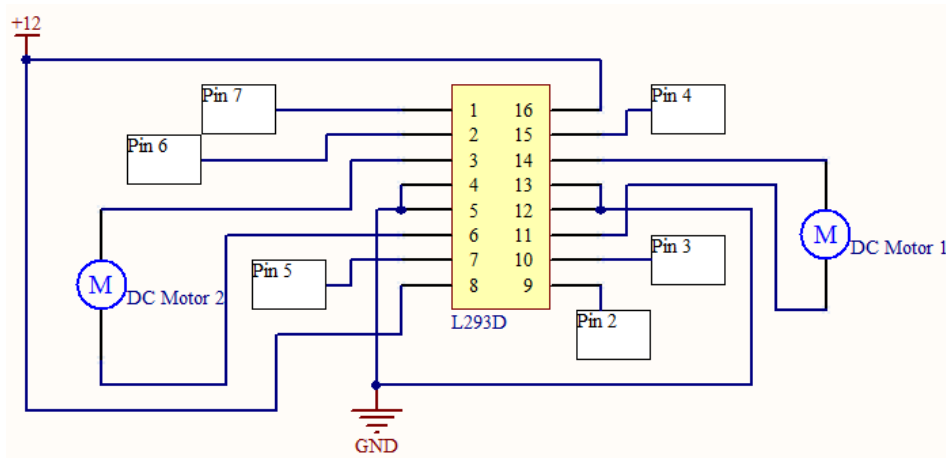
Σημειώνεται ότι τα τεχνικά δελτία όλων των ηλεκτρονικών εξαρτημάτων που χρησιμοποιήθηκαν βρίσκονται στο Παράρτημα Θ'.

7.1. Έλεγχος Κινητήρων Συνεχούς Ρεύματος με Driver L293D

Πρώτος στόχος ήταν να τεθούν σε λειτουργία οι δύο κινητήρες συνεχούς ρεύματος. Οι κινητήρες αυτοί είναι όμοιοι (GHM-01) και το τεχνικό δελτίο τους βρίσκεται στο αντίστοιχο παράρτημα. Η τροφοδοσία που τους είναι απαραίτητη είναι αυτή των 12 Volt. Οι ηλεκτρικοί κινητήρες επιλέχθηκαν να είναι συνεχούς ρεύματος γιατί είναι εύκολος ο έλεγχος της ταχύτητας περιστροφής, καθώς είναι ανάλογη της εφαρμοζόμενης τάσης στο τύμπανο, και της αναπτυσσόμενης ροπής, καθώς είναι ανάλογη του ρεύματος τυμπάνου.

Ο πιο απλός τρόπος για να ελεγχεί η δεξιόστροφη και η αριστερόστροφη κίνηση των κινητήρων είναι μέσω ενός driver. Ο driver αυτός περιέχει μία διάταξη γέφυρας-H (H-bridge), η οποία επιτρέπει την αντιστροφή της φοράς του ρεύματος και κατ'επέκταση και τη φορά λειτουργίας του κινητήρα. Με τον L293 μπορεί να ελεγχθεί ταυτόχρονα η κίνηση και των δύο χρησιμοποιώντας 3 pins για κάθε κινητήρα. Το pin εκείνο που θέτει σε λειτουργία τον κινητήρα (το Pin 2 για τον πρώτο κινητήρα και το Pin 7 για τον δεύτερο) εάν δεν συνδεθεί τότε ο εκείνος είναι πάντοτε ικανός να λειτουργήσει. Όσον αφορά στα άλλα δύο pins, εάν είναι και τα δύο σε όμοια κατάσταση (π.χ. HIGH), ο κινητήρας δεν κινείται, ενώ εάν είναι σε διαφορετική, θα κινείται προς τη μία κατεύθυνση. Αν αντιστρέψουμε τα HIGH και LOW τότε θα έχουμε κίνηση κατά την αντίστροφη φορά. Στην ουσία, με τα HIGH και LOW θέτονται σε κατάσταση ON και OFF αντίστοιχα τα ζεύγη των τεσσάρων τρανζίστορ που απαρτίζουν τη διάταξη και που είναι υπεύθυνα για τη λειτουργία και την φορά περιστροφής (συνήθως τα ζεύγη αυτά είναι τα 1-3 και 2-4).

Παρακάτω, στο **Σχήμα 7.2.** βρίσκεται το διάγραμμα του κυκλώματος με τους κινητήρες, την τροφοδοσία και τα pins του Arduino που χρησιμοποιήσαμε. Σημειώνεται ότι ο L293D μπορεί να λειτουργήσει με τάση από 4,5 έως και 36 Volt.



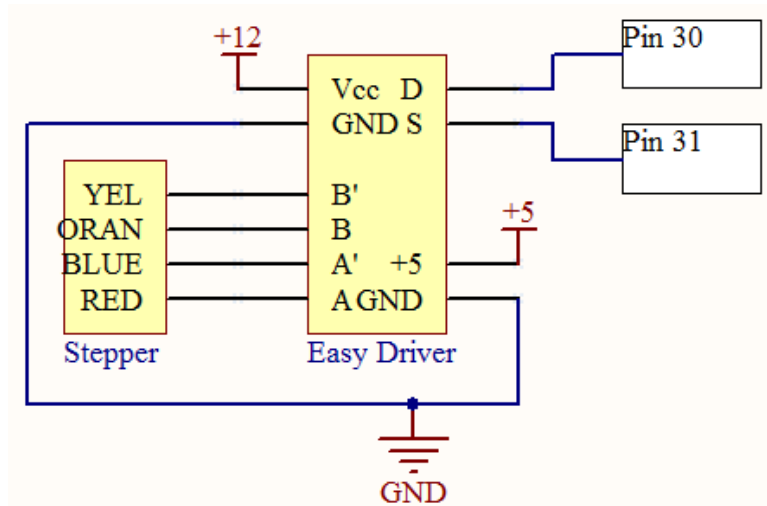
Σχήμα 7.2.: Διάγραμμα L293D

7.2. Έλεγχος Βηματικού Κινητήρα με Easy Driver

Για να ελεγχθεί η κίνηση ενός βηματικού διπολικού κινητήρα με 4 καλώδια, όπως είναι ο κινητήρας που είναι υπεύθυνος για την περιστροφική κίνηση του βραχίονα (ST6318), θα πρέπει να ελεγχθεί η φορά λειτουργίας αλλά και ο αριθμός των βημάτων που θα πραγματοποιήσει ο κινητήρας προκειμένου να εκτελέσει τη λειτουργία του. Αυτό, μέσω του Arduino φυσικά, θα μπορεί να γίνει με τη βοήθεια ενός easy driver. Μπορεί να υποστεί τροφοδοσία για τον κινητήρα από 7 έως 30 Volt, ενώ χρειάζεται τα 5 ή 3.3 Volt από τον Arduino για να δουλέψει. Αυτό που κάνει είναι να δίνει παλμούς (HIGH-LOW) μέσω των Pin που είναι συνδεδεμένα. Πιο συγκεκριμένα, σε αυτή την περίπτωση, το pin 30 είναι υπεύθυνο για την κατεύθυνση ενώ το pin 31 για τον αριθμό των βημάτων που θα πραγματοποιήσει. Η βασική ανάλυση του βήματος που έχει από μόνο του είναι στο 1/8, ενώ υπάρχει δυνατότητα μέσω των pin MS1 και MS2 να γίνει πλήρης, μισή ή στο 1/4. Εδώ χρησιμοποιήθηκε το 1/8, χωρίς να γίνει καμία αλλαγή για την αποφυγή απότομων αλλαγών. Στο κύκλωμα του easy driver είναι τοποθετημένο ένα ποτενσιόμετρο με το οποίο μπορεί να ελεγχθεί η ποσότητα που ρεύματος που θα περνάει (έως 1A). Στην συγκεκριμένη περίπτωση πρέπει να είναι περίπου στο 0,5A καθώς πιο πολύ ανεβάζει πολύ τη θερμοκρασία του κυκλώματος και με πιο λίγο δεν έχει τόση δύναμη ο κινητήρας για να κινήσει το σύστημα.

Στο Σχήμα 7.3. φαίνεται το αντίστοιχο διάγραμμα με τα pins του easy driver που χρησιμοποιήθηκαν. Σημειώνονται επίσης τα χρώματα των καλωδίων του κινητήρα (YEL για κίτρινο/yellow, ORAN για

πορτοκαλί/orange, BLUE για μπλε και RED για κόκκινο) και η σειρά κατά την οποία πρέπει να συνδεθούν.



Σχήμα 7.3.: Διάγραμμα Easy Driver

7.3. Έλεγχος Θέσης με Μηχανικό Κωδικοποιητή (encoder)

Ο κωδικοποιητής (encoder-SERIES 51) που χρησιμοποιήθηκε είναι 12 θέσεων, όπως έχει ήδη αναφερθεί. Χρησιμοποιώντας το διάγραμμα κυκλώματος του τεχνικού του δελτίου, έγιναν οι απαραίτητες κολλήσεις στις ακίδες που αναφέρονται ότι πρέπει να είναι ενωμένες και έτσι δημιουργήθηκαν 5 βασικές γραμμές (Lines) τις οποίες προεκτείναμε με καλώδια. Η μία από αυτές αποτελεί την γείωση (Line 5 ή L5), ενώ με τον κατάλληλο συνδιασμό των υπολοίπων δίνεται η θέση στην οποία βρίσκεται ο δίσκος. Στην παρούσα περίπτωση χρησιμοποιήθηκε ο συμπληρωματικός πίνακας, όπως φαίνεται παρακάτω στον **Πίνακα 7.1.**, για να οριστούν οι θέσεις, χωρίς να σημαίνει ότι δεν θα μπορούσε να χρησιμοποιηθεί ο κανονικός. Όταν τα pins στα οποία έχουμε συνδέσει την κάθε γραμμή δίνουν τις παρακάτω τάσεις τότε συμπεραίνεται και η σωστή θέση του δίσκου.

ΘΕΣΗ	ΓΡΑΜΜΗ 1	ΓΡΑΜΜΗ 2	ΓΡΑΜΜΗ 3	ΓΡΑΜΜΗ 4
1	HIGH	HIGH	HIGH	HIGH
2	LOW	HIGH	HIGH	HIGH
3	HIGH	LOW	HIGH	HIGH
4	LOW	LOW	HIGH	HIGH



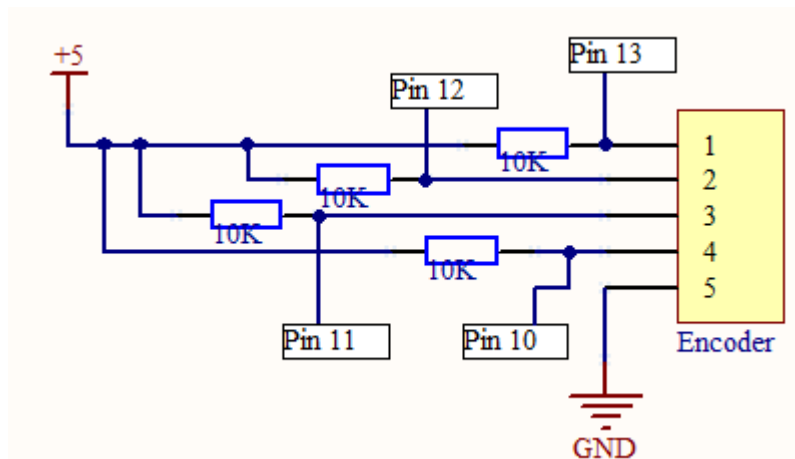
5	HIGH	HIGH	LOW	HIGH
6	LOW	HIGH	LOW	HIGH
7	HIGH	LOW	LOW	HIGH
8	LOW	LOW	LOW	HIGH
9	HIGH	HIGH	HIGH	HIGH
10	LOW	HIGH	HIGH	LOW
11	HIGH	LOW	HIGH	HIGH
12	LOW	LOW	HIGH	LOW

Πίνακας 7.1.: Συμπληρωματικές θέσεις

Στον Πίνακα 7.2. φαίνεται η κάθε γραμμή με ποιο pin είναι συνδεδεμένο ενώ στο Σχήμα 7.4. φαίνεται το διάγραμμα κυκλώματος του κωδικοποιητή. Για τα κυκλώματα αυτά, όπως φαίνεται, χρειάζεται και μία αντίσταση Pull-up, για το κάθε κύκλωμα, των 10 kΩ. Δηλαδή ανάλογα με το αν περνάει ρεύμα στην συγκεκριμένη Γραμμή υπάρχει ή όχι τάση και στο pin. Έτσι, αυτά τα pins χρησιμοποιούνται ως είσοδοι στο σύστημα, δηλαδή λαμβάνεται πληροφορία από αυτά.

ΓΡΑΜΜΗ 1 (L1) / (OUTPUT 1)	PIN 13
ΓΡΑΜΜΗ 2 (L2) / (OUTPUT 2)	PIN 12
ΓΡΑΜΜΗ 3 (L3) / (OUTPUT 4)	PIN 11
ΓΡΑΜΜΗ 4 (L1) / (OUTPUT 8)	PIN 10

Πίνακας 7.2.: Αντιστοίχιση γραμμών-pins



Σχήμα 7.4.: Διάγραμμα Κωδικοποιητή

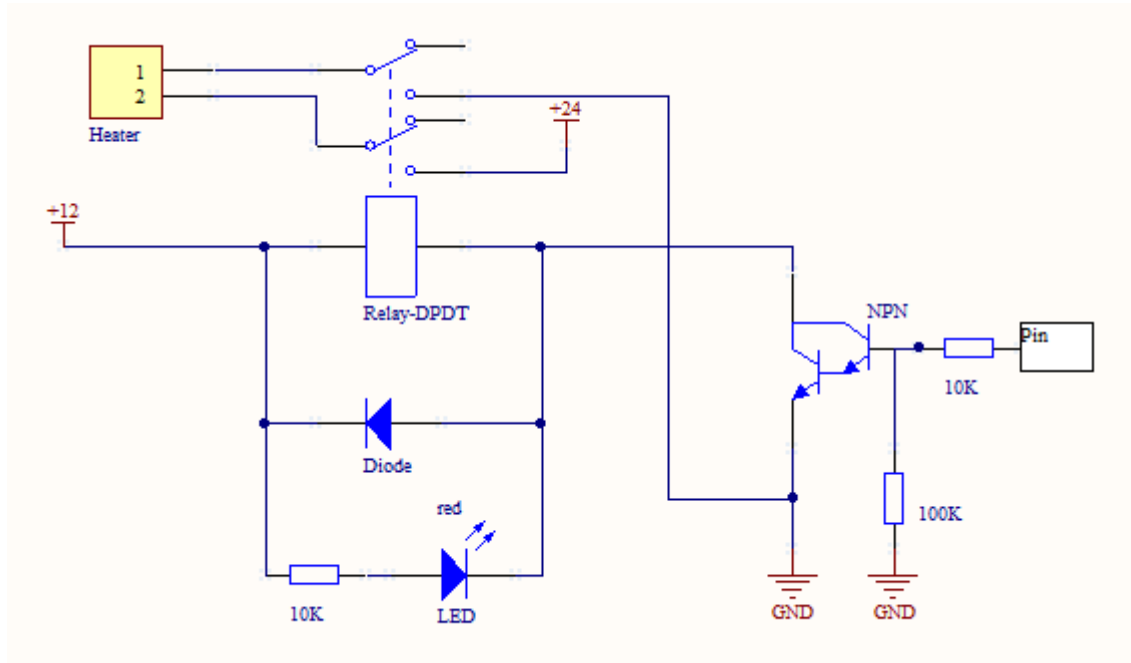
7.4. Έλεγχος Λειτουργίας Θερμαντήρων (heaters)

Οι θερμαντήρες που χρησιμοποιήθηκαν (HP05-1/08-24) για να μπορέσουν να τεθούν σε λειτουργία χρειάζονται από έναν ηλεκτρονόμο (relay) ο οποίος ελέγχεται από ένα άλλο κύκλωμα και είναι ικανός να ελέγχει ένα κύκλωμα εξόδου υψηλότερης ισχύος από το κύκλωμα εισόδου. Ο ηλεκτρονόμος περιέχει ένα πηνίο το οποίο όταν διαρέεται από ρεύμα δημιουργεί μαγνητικό πεδίο, με αποτέλεσμα να έλκει το μεταλλικό διακόπτη προς την κατεύθυνσή του.

Για να ανοίξει το πηνίο, χρειάζεται μεγαλύτερο ρεύμα από αυτό που μπορεί να δώσει ο Arduino και για αυτό το λόγο χρησιμοποιήθηκε ένα τρανζίστορ πυριτίου για ενίσχυση του σήματος (BC517). Συνοπτικά, το τρανζίστορ λειτουργεί ως ενισχυτής καθώς δημιουργείται διαφορά δυναμικού μεγαλύτερη των 0,7 V μεταξύ της βάσης και του εκπομπού. Τότε, υπάρχει κίνηση ηλεκτρονίων από τον εκπομπό προς τη βάση και από τη βάση στο συλλέκτη. Δίνοντας ρεύμα στην βάση και προσθέτοντας τα ηλεκτρόνια του εκπομπού πραγματοποιείται ενίσχυση του ρεύματος.

Επιπλέον, για την ασφάλεια του κυκλώματος συνδέεται παράλληλα με το πηνίο και μία δίοδος πυριτίου (1N4001). Η δίοδος χρησιμοποιείται σε ηλεκτρονόμους συνεχούς ρεύματος και συνδέεται ανάστροφα. Όταν στο πηνίο του ηλεκτρονόμου υπάρχει τάση δημιουργείται μαγνητικό πεδίο. Όταν διακοπεί η τάση τότε το μαγνητικό πεδίο που εξαφανίζεται δημιουργεί στο πηνίο ένα υψηλό αντίθετο ρεύμα. Το ρεύμα αυτό τότε, περνάει μέσα από τη δίοδο και δεν καταστρέφει το τρανζίστορ και το υπόλοιπο κύκλωμα. Επίσης, για να γνωστοποιείται πότε είναι αναμμένος ο θερμαντήρας, έχει συνδεθεί παράλληλα με τη δίοδο ένα LED κόκκινο το οποίο ανάβει. Στο **Σχήμα 7.5**, φαίνεται το διάγραμμα κυκλώματος του κάθε

θερμαντήρα. Το μόνο που αλλάζει είναι ότι για τον πρώτο θερμαντήρα είναι συνδεδεμένο το pin 25 ενώ για το δεύτερο το pin 26.



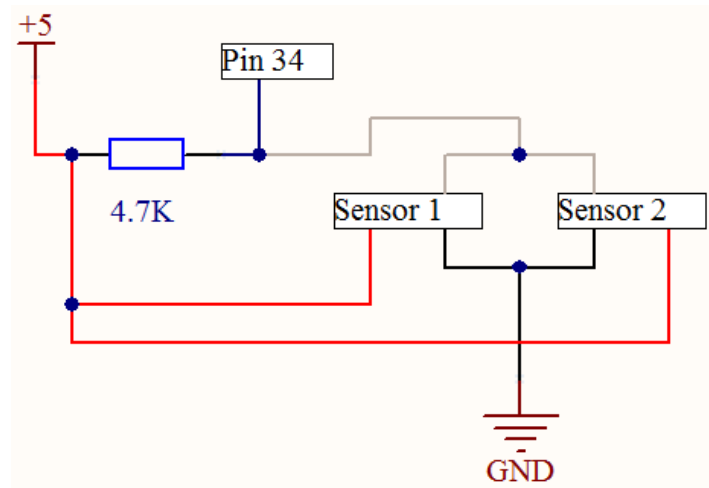
Σχήμα 7.5.: Διάγραμμα Θερμαντήρα

7.5. Λειτουργία Αισθητήρων Θερμοκρασίας

Απαραίτητο για το σύστημα είναι να υπάρχει ενημέρωση για τη θερμοκρασία των θερμαινόμενων δοχείων. Ο στόχος αυτός γίνεται πραγματικότητα μέσω του ευέλικτα, προγραμματιζόμενου αισθητήρα θερμοκρασίας DS18B20. Η συγκεκριμένη έκδοση που χρησιμοποιήθηκε ήταν αδιάβροχη και με μακρύ καλώδιο έτσι ώστε να μετρηθεί η παραφίνη στο εσωτερικού του δοχείου, μακριά από το ηλεκτρικό κύκλωμα. Ο αισθητήρας αυτός μπορεί να μετρήσει θερμοκρασίες μέχρι 125 °C αλλά η ακρίβεια του είναι καλή μέχρι τους 100 °C. Στην περίπτωση του πρωτοκόλλου ιστολογίας δεν χρειάζεται να μετρήσουμε τόσο μεγάλες θερμοκρασίες.

Ένα άλλο χαρακτηριστικό του αισθητήρα αυτού είναι ότι ακολουθεί το πρωτόκολλο συσκευών 1-wire, το οποίο σημαίνει ότι με ένα μόνο καλώδιο, μία αντίσταση, μία τάση και μια γείωση μπορεί να συνδέσει σε παράλληλη σύνδεση όσες συσκευές-σκλάβους (slaves) επιθυμεί ο χρήστης. Έχοντας συνδεδεμένο μόνο ένα pin στον Arduino και χρησιμοποιώντας τις κατάλληλες βιβλιοθήκες είναι δυνατή η λειτουργία και ο έλεγχος πολλών τέτοιων συσκευών. Το διάγραμμα κυκλώματος που χρησιμοποιήθηκε βρίσκεται στο **Σχήμα 7.6.** Στο ίδιο σχήμα μπορεί να γίνει αντιληπτή η συνδεσιμολογία του κάθε καλωδίου του αισθητήρα. Ο αισθητήρας έχει τρία καλώδια, ένα μαύρο που

πάει στη γείωση, ένα κόκκινο που πάει στην τάση και ένα λευκό που μεταφέρει δεδομένα και συνδέεται με το pin.



Σχήμα 7.6.: Διάγραμμα Αισθητήρων Θερμοκρασίας

Οι βιβλιοθήκες που είναι απαραίτητες για μια τέτοια συσκευή είναι:

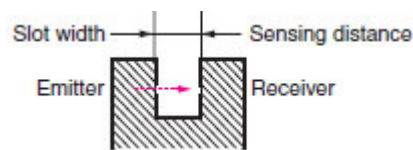
- Η address finder με την οποία μπορεί να βρεθεί η διεύθυνση της συσκευής, η οποία είναι μοναδική, είναι κάτι σαν ταυτότητα της κάθε συσκευής από τον κατασκευαστή της. Η διεύθυνση αυτή θα χρειάστηκε για να γραφεί σε ένα σημείο του κυρίως κώδικα 1-wire.
- Η One Wire, η οποία θέτει σε λειτουργία τον αισθητήρα και πρέπει να υπάρχει στο φάκελο «libraries» του προγράμματος, ώστε να μπορεί να την καλέσει το κυρίως πρόγραμμα.
- Η Dallas Temperature, η οποία και αυτή πρέπει να υπάρχει στο φάκελο «libraries» και είναι υπεύθυνη για την σωστή ανάγνωση της θερμοκρασίας.

Οι παραπάνω βιβλιοθήκες λήφθηκαν έτοιμες από το διαδίκτυο. Υπήρχαν διάφορες διαφορετικές εκδοχές σε κάποιες από αυτές αλλά χρησιμοποιήθηκαν αυτές που ταίριαζαν στη συγκεκριμένη περίπτωση και βρίσκονται στο Παράρτημα Ζ'.

7.6. Έλεγχος Θέσης με Φωτοευαίσθητους Μικροαισθητήρες

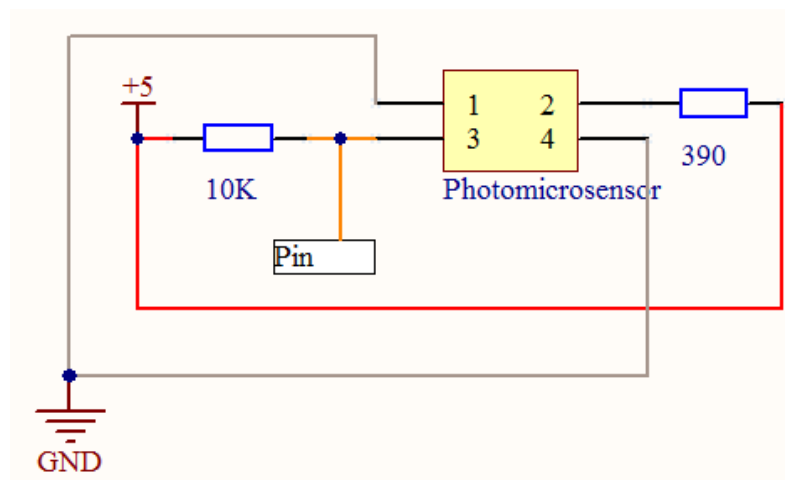
Ο έλεγχος θέσης ενός βηματικού κινητήρα μπορεί να γίνει με έλεγχο του αριθμού των βημάτων του. Για απόλυτη σιγουριά και ακρίβεια όμως, για να ελεγχθεί η θέση του βραχίονα, χρησιμοποιήθηκαν 3 φωτοευαίσθητοι μικροαισθητήρες (Photomicrosensors), ένας για κάθε θέση. Οι αισθητήρες αυτοί μπορεί να είναι δύο τύπων γιατί στέλνουν σήμα με δύο διαφορετικούς τρόπους. Πρώτον, όταν

καλύπτονται τελείως και δεν μπορεί να πλησιάσει το φως (π.χ. ανίχνευση UV ακτίνας) και δεύτερον όταν δημιουργούν μία δέσμη φωτός μέσω φωτοδιόδου, η οποία όταν δεν μπορεί να φτάσει στην απέναντι μεριά της ειδικής σχισμής (δηλαδή από τον εκπομπό/emitter στον συλλέκτη/receiver του εμπεριεχόμενου τρανζίστορ), επειδή εμποδίζεται από κάτι, δίνει σήμα (**Σχήμα 7.7.**). Το μέγεθος της σχισμής στην οποία μπορεί να γίνει αντιληπτή η ακτίνα (sensing distance) είναι χαρακτηριστικό του κάθε αισθητήρα. Σε αυτή την περίπτωση είναι απαραίτητη η χρήση ενός μέσου το οποίο θα διακόπτει, όταν είναι επιθυμητό, την ακτίνα. Στην συγκεκριμένη ιστοκινέτα χρησιμοποιήθηκε αισθητήρας του δεύτερου τύπου.



Σχήμα 7.7.: Λειτουργία Φωτοαισθητήρα

Ο αισθητήρας αυτός είχε τέσσερα (4) pins όπως φαίνεται και στο **Σχήμα 7.8.**. Ένα για την άνοδο της διόδου (Pin 2, κόκκινο καλώδιο), ένα για την κάθοδο (Pin 4, λευκό καλώδιο), ένα για τον εκπομπό (Pin 1, λευκό καλώδιο), και έναν για τον συλλέκτη (Pin 3, πορτοκαλί καλώδιο) (ο εκπομπός εκπέμπει ηλεκτρόνια προς τον συλλέκτη). Διακόπτοντας την επικοινωνία μεταξύ εκπομπού-συλλέκτη θα δοθεί σήμα στο Pin που είναι συνδεδεμένο στον Arduino, το Pin δηλαδή λειτουργεί ως είσοδος του συστήματος. Στο σύστημα υπάρχουν τρία τέτοια κυκλώματα, όπως του **Σχήματος 7.7.**, με αντιστοίχιση Pins-αισθητήρων εκείνη του **Πίνακα 7.3.**



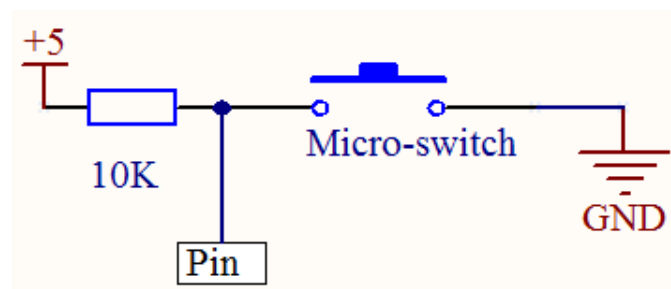
Σχήμα 7.8.: Διάγραμμα Φωτοαισθητήρων

PhotoMicroSensor 1 / (θέση 1, δίσκοι)	PIN 44
PhotoMicroSensor 2 / (θέση 2, θερμαινόμενο 1)	PIN 42
PhotoMicroSensor 3 / (θέση 3, θερμαινόμενο 2)	PIN 40

Πίνακας 7.3.: Αντιστοίχιση αισθητήρων-pins

7.7. Έλεγχος Θέσης με Μικροδιακόπτες

Για την πραγματοποίηση του πρωτοκόλλου είναι απαραίτητος ο έλεγχος της ανώτερης και της κατώτερης επιθυμητής θέσης της κατακότυφης κίνησης της κασετίνας. Γι' αυτό το λόγο βιδώθηκαν, στον κάθετο βραχίονα, δύο (2) μικροδιακόπτες (άνω και κάτω). Κάθε διακόπτης έχει τρεις επαφές. Μία κανονικά ανοιχτή, μία κανονικά κλειστή και μία κοινή. Σε αυτό το κύκλωμα δεν έχει συνδεθεί η κοινή επαφή και αυτό καθιστά το διακόπτη να βρίσκεται στην κανονικά ανοιχτή επαφή. Όπως φαίνεται και στο **Σχήμα 7.9.**, όταν είναι ανοιχτοί, δηλαδή όταν δεν κλείνουν κύκλωμα, δίνουν σήμα στο αντίστοιχο pin, ενώ όταν είναι κλειστοί το ρεύμα περνάει κατευθείαν στη γείωση και δεν δίνει σήμα. Στη συγκεκριμένη περίπτωση το pin που είναι συνδεδεμένο με τον πάνω διακόπτη είναι το pin 9 ενώ με τον κάτω το pin 8. Σημειώνεται ότι χρησιμοποιήθηκε και μία αντίσταση των 10 kΩ όπως φαίνεται στο ίδιο σχήμα.



Σχήμα 7.9.: Διάγραμμα Μικροδιακοπών

7.8. Λειτουργία Οθόνης Υγρών Κρυστάλλων (LCD)

Για να έχει άμεση επαφή ο χρήστης με την εξέλιξη του προγράμματος θεωρήθηκε απαραίτητη η χρήση μιας οθόνης η οποία αρχικά, θα καλοσορίζει το χρήστη, έπειτα θα αναγράφει σε ποιο δοχείο βρίσκεται η κασετίνα, τι περιέχει αυτό το δοχείο, πόση ώρα ακόμα πρέπει να καθίσει σε αυτό και στα τελευταία δοχεία, ποια είναι η θερμοκρασία του δοχείου του γίνεται η επεξεργασία την παρούσα στιγμή ή/και η θερμοκρασία του επόμενου θερμαινόμενου δοχείου. Αυτό σημαίνει ότι μία οθόνη τεσσάρων (4) γραμμών μας είναι αρκετή. Για παράδειγμα, η οθόνη που θα εμφανίζεται στην αρχή του πρώτου δοχείου θα είναι εκείνη του **Πίνακα 7.4.** με μωβ γράμματα, στον οποίο βρίσκεται και η επεξήγηση της κάθε γραμμής στα δεξιά κελιά. Λίγο διαφορετική είναι η οθόνη των δύο θερμαινόμενων δοχείων, η οποία βρίσκεται στον **Πίνακα 7.5.**, που περιέχει και την θερμοκρασία του αντίστοιχου δοχείου αλλά και του επόμενου, όταν εκείνο είναι θερμαινόμενο. Σύμφωνα με τις ανάγκες της οθόνης λοιπόν, χρησιμοποιήθηκε αυτή που φαίνεται στο **Σχήμα 7.10.** (I2C TWI LCD2004 Module) η οποία έχει 4 γραμμές και χωρητικότητα 20 χαρακτήρων ανά γραμμή.



Σχήμα 7.10.: LCD οθόνη 4x20

BEAKER 1 (1/11)	Δοχείο 1 (βρίσκεται στο πρώτο από τα 11 δοχεία)
ETHANOL 70%	Περιέχει αιθανόλη 70%
Time Left: 20 Min	Η κασετίνα πρέπει να μείνει άλλα 20 λεπτά σε αυτό το δοχείο (αντίστροφη χρονομέτρηση σε λεπτά)

Πίνακας 7.4.: Πιθανή οθόνη και επεξήγηση.

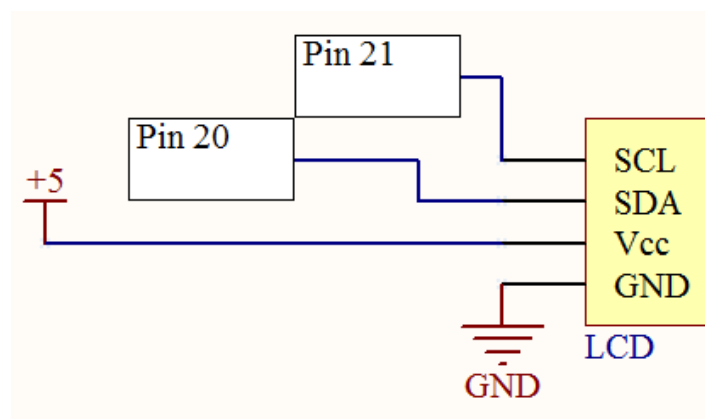
THERMAL 1 (10/11)	Θερμαινόμενο Δοχείο 1 (βρίσκεται στο δέκατο από τα 11)
-------------------	--

	δοχεία)
PARAFFIN 58oC	Περιέχει παραφίνη στους 58 βαθμούς Κελσίου
Time Left: 42 Min	Η κασετίνα πρέπει να μείνει άλλα 42 λεπτά σε αυτό το δοχείο (αντίστροφη χρονομέτρηση σε λεπτά)
Thermal 2: 63oC	Η θερμοκρασία του επόμενου θερμαινόμενου δοχείου είναι 63 βαθμοί Κελσίου

Πίνακας 7.5.: Πιθανή οθόνη θερμαινόμενου δοχείου και επεξήγηση.

Η συγκεκριμένη οθόνη είναι συσκευή I2C (Inter-Integrated Circuit) για να λειτουργήσει χρειάζεται, όπως φαίνεται και στο διάγραμμα κυκλώματος στο **Σχήμα 7.11.**, εκτός από την τάση των 5 Volt και την γείωση, άλλα δύο Pin από τον Arduino, το data και το clock pin. Συνήθως αυτά βρίσκονται στα αναλογικά pin 4 και 5 αντίστοιχα, ενώ στον συγκεκριμένο τύπο Arduino (Mega) βρίσκονται στα 20 και 21 αντίστοιχα. Σε γενικές γραμμές, καθώς το clock pin πάλλεται από low σε high μεταφέρονται δεδομένα μέσω του data pin από τον Arduino στην I2C συσκευή. Ενώ, όταν το clock pin πάλλεται από high σε low μεταφέρονται δεδομένα μέσω του data pin από την I2C συσκευή στον Arduino. Πολλές συσκευές αυτού του τύπου (slaves) μπορούν να ελεγχθούν μόνο από αυτά τα δύο Pins από την κύρια συσκευή (master), καθώς κάθε συσκευή έχει τη δική της μοναδική διεύθυνση.

Επιπρόσθετα, η συσκευή ακολουθεί το TWI (two wire interface) πρωτόκολλο το οποίο απαιτεί μόνο δύο καλώδια για τη λειτουργία μεταξύ των 2 συσκευών τα οποία στη συγκεκριμένη περίπτωση είναι τα data και clock pin. Βασική προϋπόθεση για να πραγματοποιηθούν τα παραπάνω, μέσω του κώδικα, είναι η χρησιμοποίηση της βιβλιοθήκης «LiquidCrystal_I2C» και της βιβλιοθήκης «Wire» οι οποίες πρέπει να βρίσκονται στο φάκελο «libraries» του προγράμματος για να μπορέσουν να καλεστούν από το κύριο πρόγραμμα.



Σχήμα 7.11.: Διάγραμμα κυκλώματος LCD



8. Προγραμματισμός και Περιγραφή του Κώδικα

Σ'ένα πρόγραμμα Arduino υπάρχουν δύο βασικές συναρτήσεις, η “void setup” και η “void loop”. Πριν τη “void setup” σημειώνονται πιθανές βιβλιοθήκες που μπορεί να χρησιμοποιήθηκαν, ορίζονται οι global μεταβλητές, δηλαδή εκείνες που χρησιμοποιούνται σε όλο το πρόγραμμα, τα pins που χρησιμοποιήθηκαν και οι πίνακες που χρειάστηκαν. Έπειτα, στην “void setup”, ορίζονται ποια pins είναι είσοδοι (input), ποια έξοδοι (output), προετοιμάζονται πιθανές συσκευές, και εάν είναι επιθυμητή μια μόνιμη κατάσταση κάποιου pin τότε ορίζεται αναλόγως HIGH ή LOW. Στην “void loop” συνάρτηση, τοποθετείται ο κυρίως κώδικας, ο οποίος θα τρέχει συνεχώς επαναλαμβανόμενα. Άλλες συναρτήσεις που χρειάζεται να δημιουργηθούν μπορούν να τοποθετηθούν πριν ή μετά τη “void loop”. Σε αυτόν τον κώδικα, οποίος βρίσκεται στο Παράρτημα ΣΤ', δημιουργήθηκε η “void printTemperature”, με την οποία δίνεται η θερμοκρασία από τους αισθητήρες σε μια μεταβλητή και η “void rotate” με την οποία λειτουργεί ο βηματικός κινητήρας δίνοντάς του παλμούς. Τα σχόλια είναι σημειωμένα με μπλε χρώμα, για να ξεχωρίζουν, και αναφέρουν τι κάνει στο συγκεκριμένο σημείο ο κώδικας ή/και αιτιολογούν κάποιες επιλογές.

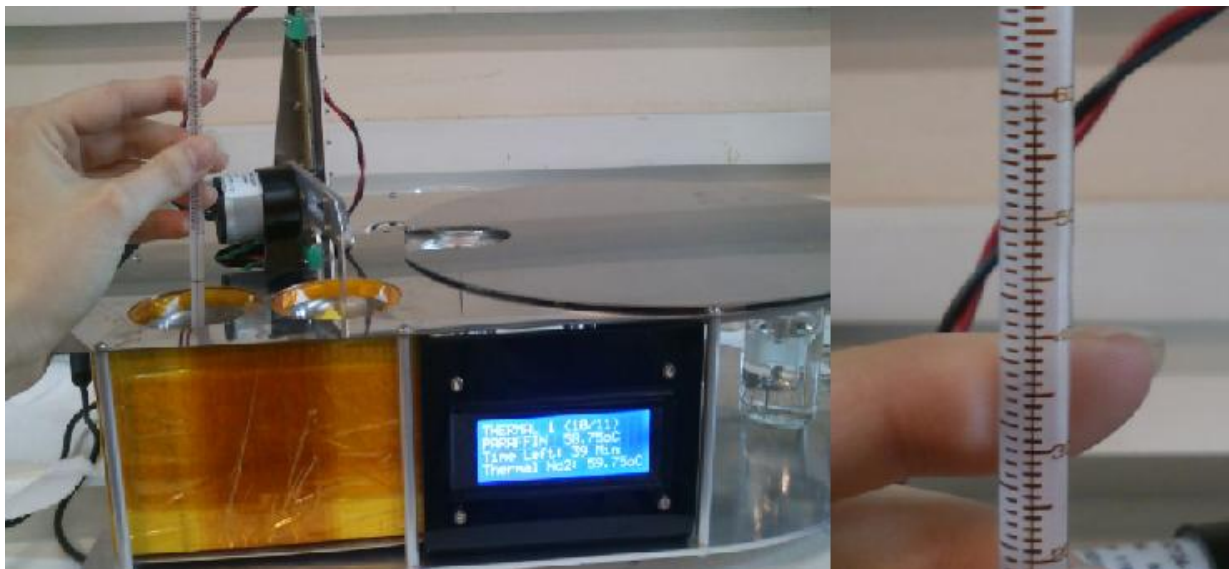
Στον συγκεκριμένο κώδικα συνολικά χρησιμοποιήθηκαν οι βιβλιοθήκες: *OneWire.h*, *DallasTemperature.h*, *Wire.h*, *LiquidCrystal_I2C.h*, οι οποίες βρίσκονται στο Παράρτημα Ζ'.

Ο κώδικας περιμένει τον χρήστη να πατήσει το κουμπί για να ξεκινήσει η “void loop”. Με αυτόν τον τρόπο η συνάρτηση αυτή δεν θα τρέχει συνέχεια παρά μόνο εάν ο χρήστης ξαναπατήσει το κουμπί μετά το τέλος του προγράμματος. Όταν πατηθεί το κουμπί ξεκινάει η αρχικοποίηση του συστήματος θέτοντας τον βραχίονα στην ανώτερη θέση και τοποθετώντας τον στην θέση μπροστά από το δίσκο και φέρνοντας το πρώτο δοχείο μπροστά από τον βραχίονα. Αυτό γίνεται με τη βοήθεια των σημάτων που έρχονται από τους αισθητήρες και δίνοντας το σωστό σήμα στους κινητήρες για να κινηθούν με τη σωστή φορά. Για όσην ώρα συμβαίνει αυτό, η lcd οθόνη καλωσορίζει τον χρήστη.

Στη συνέχεια, επαναλαμβάνεται για όλα τα δοχεία με τα διαλύματα η διαδικασία: ο βραχίονας με την κασετίνα κατεβαίνει κάτω μέχρι να συναντήσει τον διακόπτη, ο οποίος θα του δώσει σήμα για να σταματήσει, περιμένει για κάποια συγκεκριμένη διάρκεια (μισή ή μία ώρα) και μετά ανεβαίνει πάνω μέχρι να τον σταματήσει ο πάνω διακόπτης και με τη βοήθεια του κωδικοποιητή και του κινητήρα για τον δίσκο, ο δίσκος γυρνάει στην επόμενη θέση. Ταυτόχρονα με αυτή τη διαδικασία η lcd οθόνη δείχνει το στάδιο της διαδικασίας, το διάλυμα στο οποίο βρίσκεται η κασετίνα και πόσος χρόνος απομένει για το συγκεκριμένο διάλυμα.

Στα δύο τελευταία δοχεία, τα οποία είναι θερμαινόμενα, δεν χρειάζεται να κουνηθεί άλλη θέση ο δίσκος αλλά πρέπει να μετακινηθεί ο βραχίονας με τη βοήθεια του βηματικού κινητήρα και των μικροφωτοαισθητήρων.

Πρέπει να σημειωθεί ότι οι θερμαντήρες ανοίγουν τρεις ώρες πριν, ενώ η σταθεροποίησή της θερμοκρασίας τους στους 58-59°C γίνεται μία ώρα πριν μπει η κασετίνα στο καθένα. Η σταθεροποίηση γίνεται ανοιγοκλείνοντας τους θερμαντήρες, όταν ανεβαίνει ή πέφτει κάτω από μια συγκεκριμένη θερμοκρασία, η οποία έχει βρεθεί πειραματικά με τη βοήθεια θερμομέτρου. Στην lcd οθόνη αναφέρεται η θερμοκρασία που επόμενου θερμαινόμενου δοχείου καθώς γίνεται η σταθεροποίησή της, στην τέταρτη γραμμή, αλλά και του τρέχοντος θερμαινόμενου δοχείου στην δεύτερη γραμμή. Επειδή ο αισθητήρας θερμοκρασίας βρίσκεται στο εξωτερικό μέρος του δοχείου, στην οθόνη προσθέτουμε τη διαφορά που έχει παρατηρηθεί πειραματικά ώστε να δείχνει στον χρήστη την θερμοκρασία που έχει πραγματικά η παραφίνη. Η πειραματική διαδικασία φαίνεται και στο **Σχήμα 8.1**.



Σχήμα 8.1.: Πειραματική διαδικασία σταθεροποίησης θερμοκρασίας/προβολή σωστής θερμοκρασίας στην οθόνη

9. Εφαρμογές που Πραγματοποιήθηκαν

Για να γίνει η διαπίστωση ότι η συσκευή αυτόματου συστήματος ιστοκινέτας λειτουργεί σωστά, χωρίς λάθος αποτελέσματα, αποφασίστηκε να γίνει, από τον ίδιο ασθενή, μια ιστολογική εφαρμογή με την χειροκίνητη διαδικασία, όπως γινόταν συνήθως στο εργαστήριο, και μια με το αυτόματο σύστημα ιστοκινέτας.

Αρχικά, αφού το δείγμα από γόνατο έφτασε στο εργαστήριο, αφού ήδη είχε ολοκληρωθεί η μονιμοποίηση, κόπηκαν κάποια δισκάκια και τοποθετήθηκαν σε αιθανόλη 70% για να διατηρηθούν περισσότερο καιρό. Αφού όλα τα δείγματα κοπούν στον μικροτόμο και γίνει η βαφή με το κυανό της τολουιδίνης, θα τοποθετηθούν σε πλακίδια και θα ελεγχθούν στο μικροσκόπιο. Θα πρέπει λοιπόν, το δείγμα του χόνδρου από την χειροκίνητη λειτουργία και το δείγμα από την αυτόματη λειτουργία να δείξουν ότι έχουν τον ίδιο βαθμό οστεοαρθρίτιδας, σύμφωνα με το Κεφάλαιο 2.4.. Θα πρέπει να σημειωθεί ότι αφού προέρχονται από τον ίδιο ασθενή και από την ίδια άρθρωση, τα δείγματα βρίσκονται στο ίδιο στάδιο το οποίο είναι το στάδιο 4, διότι η οριζόντια έκταση της εμπλεκόμενης επιφάνειας του χόνδρου είναι περισσότερο από 50 %. Αυτό σημαίνει ότι ο συνολικός βαθμός της οστεοαρθρίτιδας, ο οποίος είναι απαραίτητος για την σύγκριση, δεν επηρεάζεται από το στάδιο, αλλά μόνο από τον βαθμό της οστεοαρθρίτιδας.

9.1. Εφαρμογή με Χειροκίνητη Διαδικασία

Σε μια κασετίνα τοποθετήθηκαν δύο δισκάκια χόνδρου και με τη βοήθεια ενός χρονομέτρου για να γίνεται η αλλαγή από δοχείο σε δοχείο στους σωστούς χρόνους ολοκληρώθηκε η διαδικασία. Τα διαλύματα βρίσκονται μόνιμα σε έναν απαγωγό χημικών διαλυμάτων (Σχήμα 9.9.1.) και με τη βοήθεια μιας λαβίδας τοποθετείται η κασετίνα στα δοχεία.



Σχήμα 9.1.1.: Δοχεία σε απαγωγό χημικών διαλυμάτων

Μετά την ολοκλήρωση της βύθισης της κασετίνας στα διαλύματα ζεσταίνουμε την παραφίνη σε ένα μπρίκι το οποίο βρίσκεται μέσα σε λουτρό νερού πάνω σε έναν θερμαντήρα τύπου κρεπίερας όπως στο **Σχήμα 9.1.2.** . Στο νερό καταλήγει αισθητήρας θερμοκρασίας και ο θερμαντήρας συνδέεται με θερμοστάτη οποίος ανοιγοκλείνει ώστε η θερμοκρασία του νερού να είναι περίπου 60°C , $2-3^{\circ}\text{C}$ πάνω από την επιθυμητή θερμοκρασία της παραφίνης. Σε αυτό το δοχείο-μπρίκι τοποθετείται η κασετίνα για να ολοκληρωθεί το πρωτόκολλο της επεξεργασίας του ιστού.

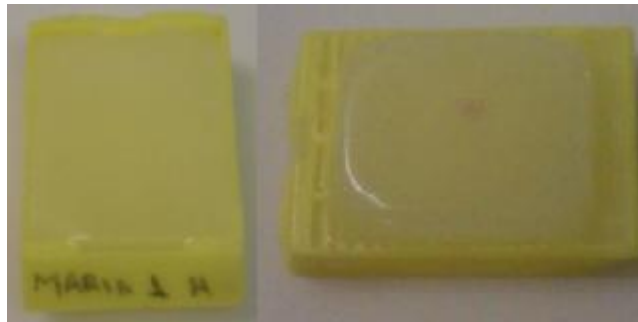


Σχήμα 9.1.2.: Σύστημα θέρμανσης παραφίνης

Στη συνέχεια, ακολουθείται η διαδικασία για τη δημιουργία καλουπιών όπως αναφέρθηκε στο Κεφάλαιο 2.3. παράγραφος Γ' ή στο Παράρτημα Γ'. Στο **Σχήμα 9.1.3.** φαίνονται τα απαραίτητα για τη διαδικασία αυτή, δηλαδή μία θερμή πλάκα στους 60°C , μια ψυχρή πλάκα στους 4°C , τα μεταλλικά καλούπια στα οποία θα φτιαχτούν οι κύβοι, λιωμένη παραφίνη στους 58°C και λαβίδες για να πιάνουμε τους ιστούς. Σημειώνεται ότι φτιάχτηκαν δύο καλούπια, ένα για κάθε δίσκο χόνδρου. Στην κάθε κασετίνα, για το συγκεκριμένο πείραμα, αναφέρεται το όνομα του πειραματιστή, ο αριθμός 1 ή 2 ανάλογα αν είναι το πρώτο ή το δεύτερο δισκάκι και ένα H (Hand) για να δηλώσει τη χειροκίνητη διαδικασία. Ένα παράδειγμα βρίσκεται στο **Σχήμα 9.1.4.**, το οποίο αποτελεί τον έτοιμο κύβο παραφίνης με το δείγμα.



Σχήμα 9.1.3.: Απαραίτητος εξοπλισμός για τη δημιουργία καλουπιών

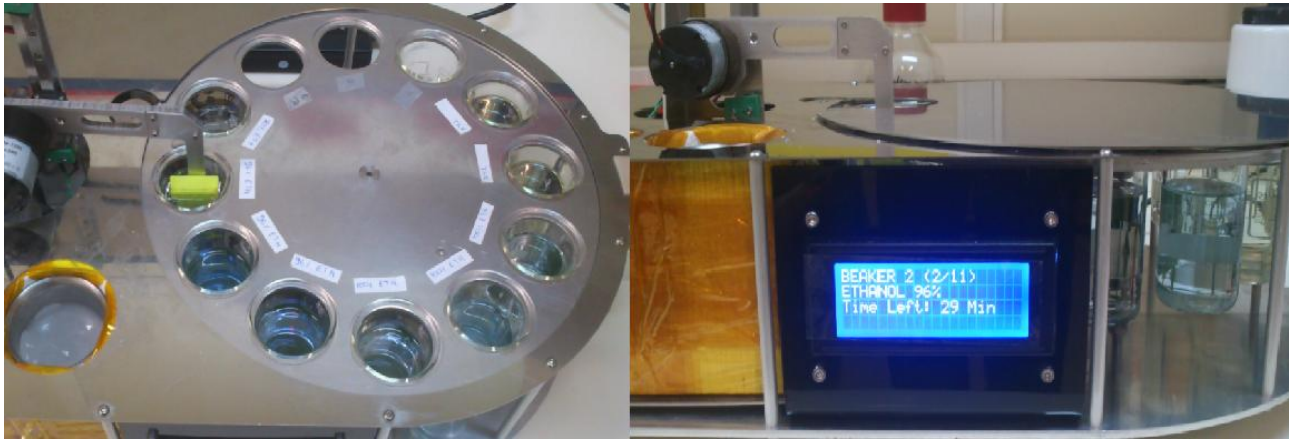


Σχήμα 9.1.4.: Παράδειγμα κύβου παραφίνης (χειροκίνητη διαδικασία)

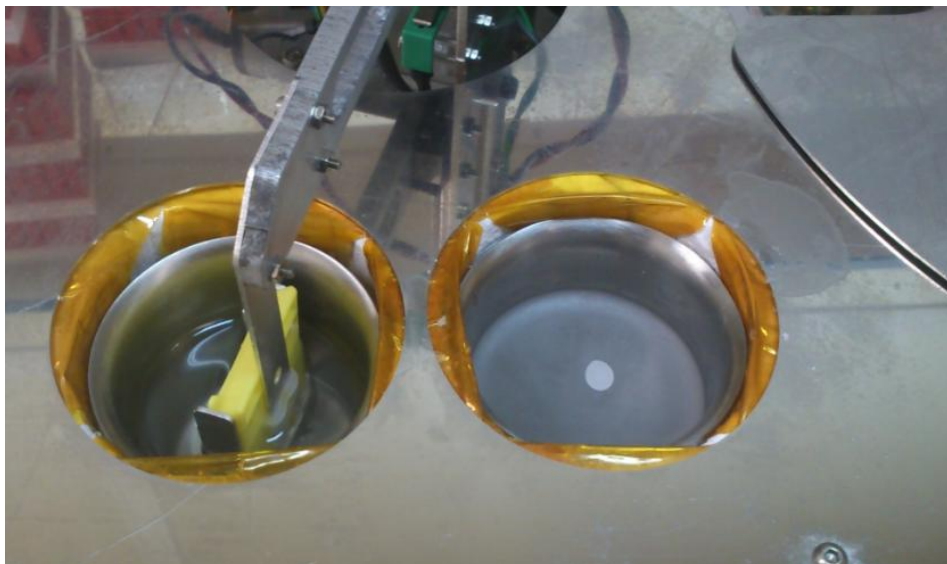
9.2. Εφαρμογή με το Σύστημα Ιστοκινέτας

Και πάλι τοποθετήθηκαν σε μια κασετίνα δύο δισκάκια χόνδρου αλλά αυτή τη φορά η διαδικασία πραγματοποιήθηκε με την αυτόματη συσκευή ιστοκινέτας. Αφού τοποθετήθηκαν τα διαλύματα στα κατάλληλα δοχεία, όπως αναφέρονταν στην κάθε ετικέτα και όπως φαίνεται στο **Σχήμα 9.2.1.**, πατήθηκε το κουμπί της συσκευής για να ξεκινήσει η διαδικασία. Στο συγκεκριμένο σχήμα φαίνεται η διαδικασία στο δεύτερο δοχείο με ανοιχτό καπάκι για να φαίνονται τα διαλύματα με της ετικέτες τους αλλά και με κλειστό καπάκι όπου φαίνεται η οθόνη υγρών κρυστάλλων.

Στο **Σχήμα 9.2.2.** φαίνεται η συσκευή στο τελευταίο στάδιο της διαδικασίας, δηλαδή στο δεύτερο θερμαινόμενο δοχείο, ενώ φαίνεται ότι ήδη ο θερμαντήρας στο προηγούμενο δοχείο έχει κλείσει.



Σχήμα 9.2.1.: Η διαδικασία βρίσκεται στο 2^ο δοχείο (Ανοιχτό/Κλειστό καπάκι)

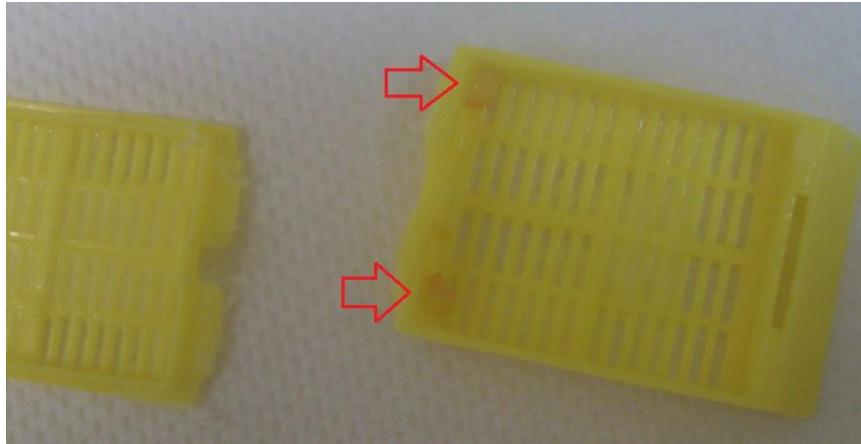


Σχήμα 9.2.2.: Η διαδικασία βρίσκεται στο τελευταίο στάδιο

Ανοίγοντας την κασετίνα παρατηρούμε ότι οι ιστοί έχουν κολλήσει στο κάτω μέρος της κασετίνας (Σχήμα 9.2.3.). Για να μπορέσουμε να τις ξεκολλήσουμε χρησιμοποιήσαμε στο σύστημα της κρεπίερας και τοποθετήσαμε λίγο πάνω την κασετίνα όπως ακριβώς και στην χειροκίνητη διαδικασία. Στη συνέχεια, ακολουθείται κανονικά η διαδικασία για τη δημιουργία καλουπιών όπως αναφέρθηκε στο Κεφάλαιο 2.3. παράγραφος Γ' ή στο Παράρτημα Γ'.

Σημειώνεται και εδώ ότι φτιάχτηκαν δύο καλούπια, ένα για κάθε δίσκο χόνδρου. Στην κάθε κασετίνα, για το συγκεκριμένο πείραμα, αναφέρεται το όνομα του πειραματιστή, ο αριθμός 1 ή 2 ανάλογα αν είναι το πρώτο ή το δεύτερο δισκάκι και ένα C (Carousel) για να δηλώσει τη διαδικασία με τη συσκευή

η οποία είναι τύπου Carousel. Ένα παράδειγμα βρίσκεται στο **Σχήμα 9.2.4.**, το οποίο αποτελεί τον κύβο παραφίνης με το δείγμα πριν ακόμα βγει από το καλούπι του αλλά και μετά.



Σχήμα 9.2.3.: Οι δίσκοι χόνδρου μετά το τέλος της διαδικασίας



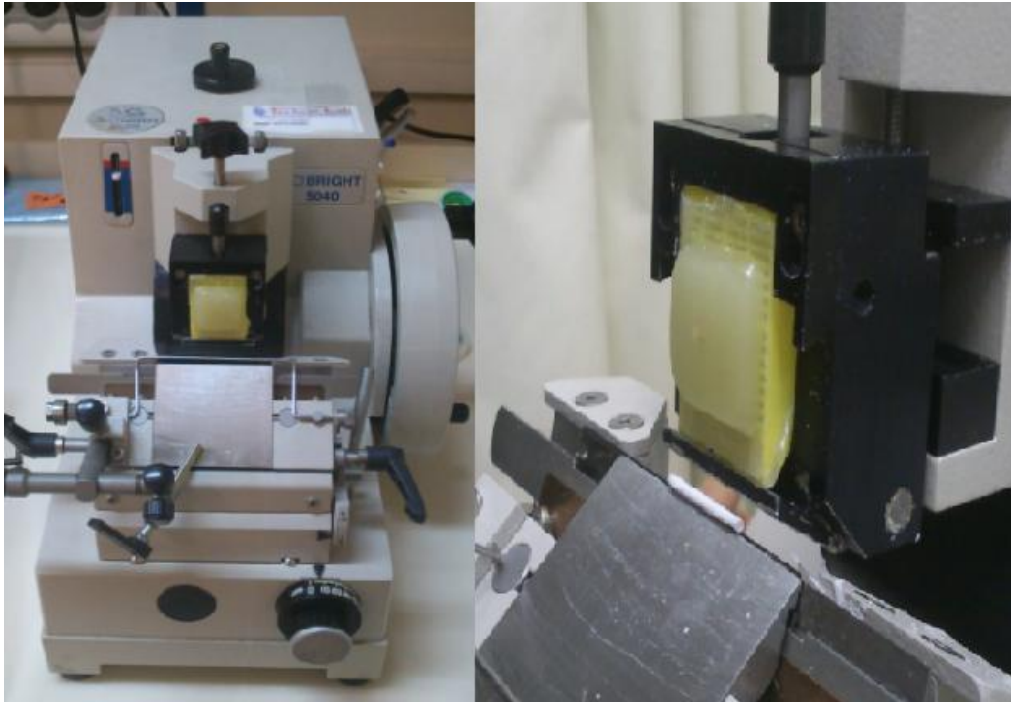
Σχήμα 9.2.4.: Καλούπια Παραφίνης

9.3. Αποτελέσματα και Σύγκριση των Δύο Εφαρμογών

Σκοπός σε αυτό το στάδιο είναι να αποδειχθεί ότι τα αποτελέσματα όσον αφορά στο βαθμό της οστεοαρθρίτιδας παραμένουν ίδια και στην χειροκίνητη διαδικασία και σε εκείνη με την συσκευή χωρίς ιδιαίτερες αλλοιώσεις.

Για να γίνει εμφανές σε ποιο στάδιο οστεοαρθρίτιδας βρίσκεται ο ιστός, πρώτα πρέπει τα καλούπια που δημιουργήθηκαν, να κοπούν στον μικροτόμο για τη δημιουργία τομών, όπως αναφέρεται στο Κεφάλαιο 2.3, παράγραφος Δ' ή/και στο Παράρτημα Γ'. Το κάθε καλούπι τοποθετήθηκε στον μικροτόμο όπως

φαίνεται στο **Σχήμα 9.3.1.** Αφού κόπηκαν οι τομές διαλέχθηκαν οι καλύτερες από το νερό (στο **Σχήμα 9.3.2.**) για να κολληθούν στα πλακίδια, τα οποία φαίνονται και αυτά στο **Σχήμα 9.3.2.** Τα πλακίδια αναγράφουν το όνομα του πειραματιστή, 1 ή 2 και HAND ή CAROUSEL για να γίνει φανερό από ποιο καλούπι προέρχεται η συγκεκριμένη τομή και τέλος η ημερομηνία κοπής.



Σχήμα 9.3.1.: Τοποθέτηση καλουπιού στο μικροτόμο



Σχήμα 9.3.2.: Τομές στο νερό και πλακίδια έτοιμα για την τοποθέτηση των τομών

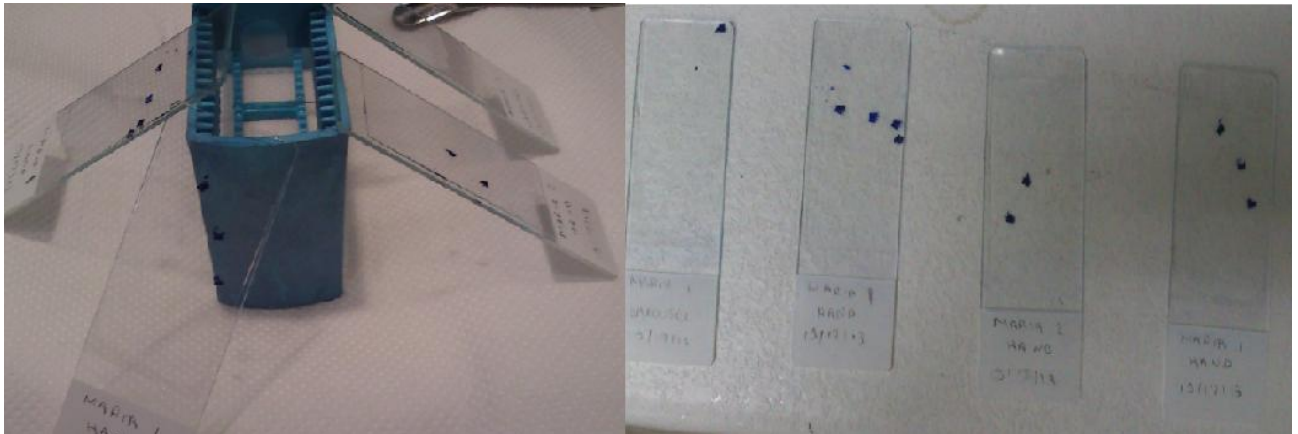
Τα έτοιμα πλακίδια με τις τομές, όπως φαίνονται στο **Σχήμα 9.3.3.**, τοποθετήθηκαν στο εκκολαπτήριο, στους 37°C για όλο το βράδυ και την άλλη μέρα συνεχίστηκε η διαδικασία με την χρώση. Η διαδικασία

αυτή αναφέρεται στο Κεφάλαιο 2.3, παράγραφος Ε' ή/και στο Παράρτημα Γ'. Σημειώνεται ότι για τις συγκεκριμένες εφαρμογές χρησιμοποιήθηκε το κυανό της τολουιδίνης.



Σχήμα 9.3.3.: Τομές στα πλακίδια και μέσα στο εκκολαπτήριο

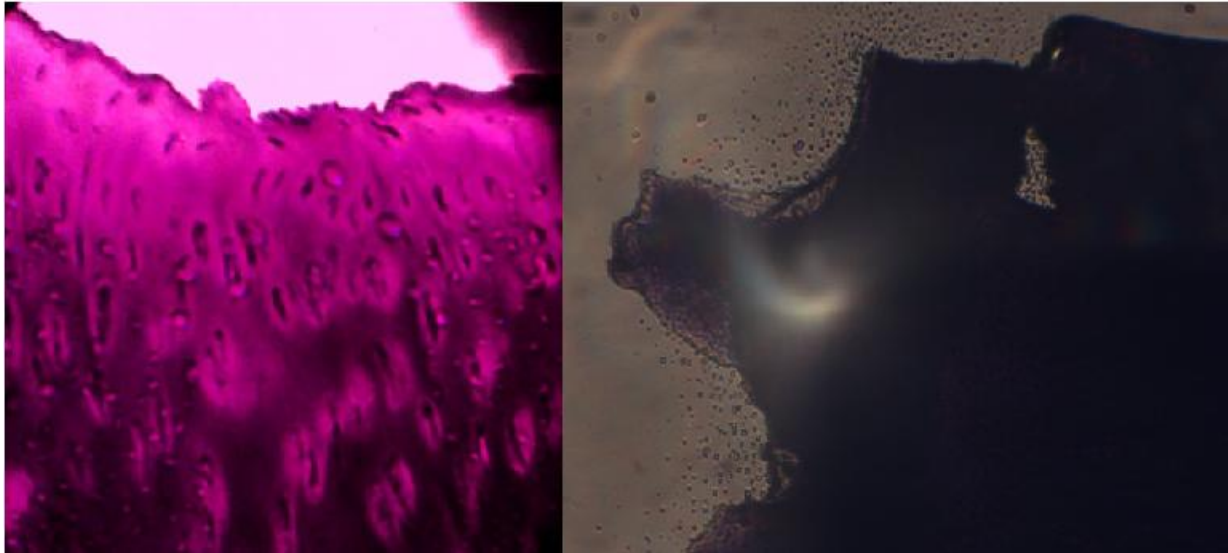
Μετά το τέλος της διαδικασίας της χρώσης, οι τομές των ιστών αφήνονται να στεγνώσουν όπως στο Σχήμα 9.3.4. . Στο συγκεκριμένο καλαθάκι τοποθετήθηκαν όλα τα πλακίδια για να μπορέσουν να βυθιστούν στα χημικά.



Σχήμα 9.3.4.: Ιστοί μετά τη χρώση

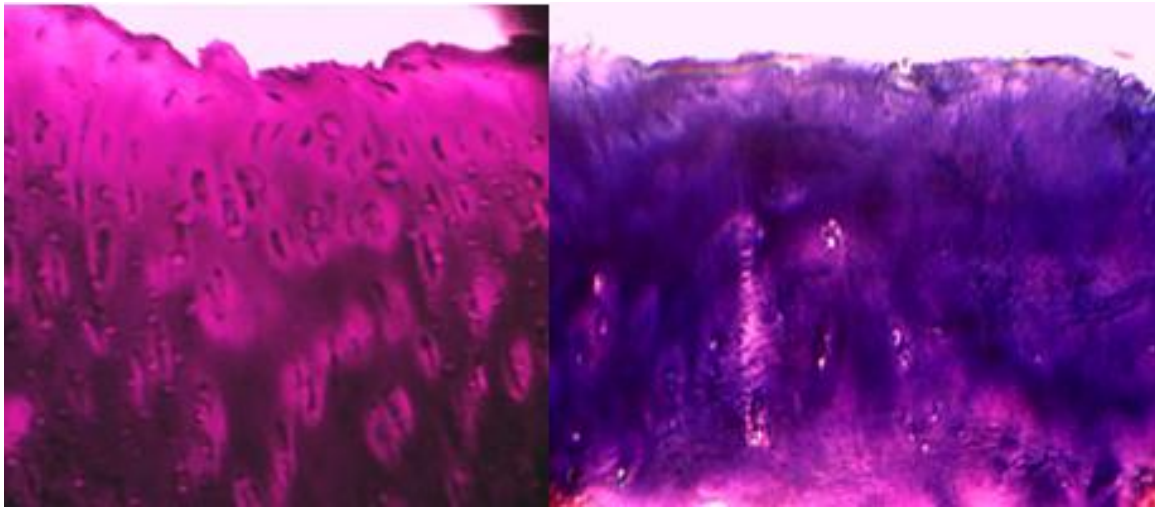
Τέλος, οι τομές ελέγχθηκαν στο μικροσκόπιο από όπου έγινε μια προσπάθεια αποτύπωσης της μεγενθυμένης εικόνας τους σε εικόνα με τη βοήθεια της κάμερας που βρίσκεται συνδεδεμένη στο μικροσκόπιο και με τη βοήθεια κινητού τηλεφώνου. Σημειώνεται ότι με τον πρώτο τρόπο εμφανίζεται στις φωτογραφίες ένα έντονο φως στο κέντρο από το φως του μικροσκοπίου και αυτό έχει ως αποτέλεσμα να θολώνει την εικόνα, ενώ είναι πολύ περιορισμένο το μέγεθος της περιοχής που μπορούμε να αποτυπώσουμε αλλά και το βάθος εστίασης. Επίσης, με τον δεύτερο τρόπο, ενώ οι τομές και τα χρώματα φαίνονται πιο καθαρά και μπορεί να γίνει διάκριση της διαβάθμισης της χρώσης στις

διάφορες περιοχές, γίνεται φανερός και ένας μαύρος κύκλος γύρω από την εικόνα, ο οποίος αποτελεί το μάτι του μικροσκοπίου, λόγω της επαφής του με το κινητό τηλέφωνο για να βγει η φωτογραφία. Για παράδειγμα παρακάτω βλέπουμε την ίδια τομή με τους δύο διαφορετικούς τρόπους αποτύπωσης (Σχήμα 9.3.5.).



Σχήμα 9.3.5.: Αριστερά φωτογραφία με κινητό τηλέφωνο και δεξιά με κάμερα

Στο παρακάτω σχήμα (Σχήμα 9.3.6) παρατηρείται δεξιά η όψη της τομής στην οποία ολοκληρώθηκε το ιστολογικό πρωτόκολλο μέσω της χειροκίνητης διαδικασίας και στην αριστερή θέση απεικονίζεται η τομή του ιστού στην οποία ολοκληρώθηκε το ιστολογικό πρωτόκολλο μέσω της συσκευής. Η συγκεκριμένη τομή ήταν πολύ σκούρα και για να φαίνεται η δομή πιο καλά ξανατοποθετήθηκε στο καθαρό νερό, για αυτό το λόγο φαίνεται πιο μωβ στο χρώμα. Σημειώνεται ότι δεν υπήρχε άλλη τομή καθώς οι υπόλοιπες τομές που προήλθαν από τη συσκευή ήταν πολύ μικρές με αποτέλεσμα να ξεκολλήσουν και να χαθούν κατά τη διάρκεια της διαδικασίας. Το ίδιο όμως συνέβει και με κάποιες τομές που προήλθαν από τη χειροκίνητη διαδικασία, επομένως αυτή η έκβαση του πειράματος δεν έχει σχέση με τη συσκευή και το κατά πόσον λειτούργησε σωστά το πρωτόκολλο. Πιο συγκεκριμένα, όταν υπάρχουν πολύ μικρές τομές ίσως να χρειάζεται να αλλάζουν οι χρόνοι που παραμένουν στα διαλύματα κατά τη διάρκεια της χρώσης, κάτι που δεν ήταν στην ευχαίρεια του πειραματιστή, λόγω έλλειψης εμπειρίας ακολουθήθηκε το κλασικό πρωτόκολλο χρώσης.



Σχήμα 9.3.6.: Δεξιά φωτογραφία από τον ιστό με τη χειροκίνητη διαδικασία και αριστερά φωτογραφία από τον ιστό με τη διαδικασία μέσω της συσκευής

Ο βαθμός της οστεοαρθρίτιδας που γίνεται αντιληπτός και στις 2 περιπτώσεις είναι **2** καθώς παρατηρείται, όπως αναφέρεται και στο Κεφάλαιο 2.4., ότι υπάρχει αποφλοίωση του ιστού, η επιφάνεια δεν είναι λεία όπως στο υγιές, ενώ το 1/3 περίπου της τομής δεν χρωματίζεται καλά από τις χρωστικές

Με βάση τα παραπάνω αποτελέσματα διαπιστώνεται ότι το αυτόματο σύστημα ιστοκινέτας που κατασκευάστηκε είναι ικανό να λαμβάνει μέρος στο ιστολογικό πρωτόκολλο χωρίς πρόβλημα.





10. Τελικά Συμπεράσματα

Από την τελική λειτουργία και τις εφαρμογές που πραγματοποιήθηκαν ήταν εφικτό να γίνουν αισθητά κάποια θετικά και κάποια αρνητικά συμπεράσματα.

Όσον αφορά στα θετικά συμπεράσματα ήταν πολύ ευχάριστο το γεγονός ότι η συσκευή λειτούργησε κανονικά και κατάφερε να πραγματοποιήσει το πρωτόκολλο χωρίς λάθη και να ολοκληρωθεί, μετέπειτα, όλη η διαδικασία για την εύρεση του βαθμού της οστεοαρθρίτιδας με επιτυχία.

Κάθε πρωτότυπο μοντέλο συσκευής όμως, όπως το συγκεκριμένο, έχει κάποια λάθη, κυρίως όσον αφορά στον σχεδιασμό του. Τα σχεδιαστικά αυτά λάθη έγιναν αντιληπτά κυρίως κατά τη διάρκεια της συναρμολόγησης αλλά και κατά τη διάρκεια της λειτουργίας της σε πραγματικές συνθήκες. Τα κύρια λάθη που εντοπίστηκαν ήταν:

- 1) μη επαρκής χώρος για τα ηλεκτρονικά και τα καλώδια,
- 2) μη υπολογισμός των τάσεων που θα αναπτύσσονταν μετέπειτα από τα καλώδια με αποτέλεσμα να δυσκολεύουν την κίνηση, και τέλος,
- 3) η όχι άριστη επαφή του κωδικοποιητή με το αξονάκι.

Το πρώτο πρόβλημα αντιμετωπίστηκε σχετικά εύκολα με την πλήρη αξιοποίηση του ελεύθερου χώρου κολλώντας τον επεξεργαστή με το ένα μέρος της πλακέτας στην βάση και το άλλο μέρος της πλακέτας στην κάτω μεριά του καλύμματος του συστήματος. Και στα δύο κομμάτια τοποθετήθηκε plexi-glass ενδιάμεσα ώστε να μην κλείσει κύκλωμα η πλακέτα ή ο επεξεργαστής με το μέταλλο και καταστραφούν τα ηλεκτρονικά.

Το δεύτερο πρόβλημα αντιμετωπίστηκε με τη βοήθεια δεματικών ώστε τα καλώδια να παραμένουν σε συγκεκριμένη περίπτωση θέση ώστε να μην εμποδίζουν τόσο ώστε να δυσκολεύουν την κίνηση. Παρόλαυτα, όπως αναφέρεται και στις οδηγίες για τον χρήστη, σε περίπτωση που ο βραχίονας δυσκολεύεται στην κίνηση προς τα θερμαινόμενα είναι καλό να ελέγχεται η αντίσταση από τα καλώδια, κυρίως εκείνων που βρίσκονται στους διακόπτες.

Το τρίτο πρόβλημα ήταν αυτό που δυσκόλεψε περισσότερο και θέλει προσοχή από τους χρήστες. Η επαφή μεταξύ άξονα και κωδικοποιητή, με τη βοήθεια σκουληκόβιδας, δεν ήταν πολύ καλή, με αποτέλεσμα όταν κάποιος προσπαθούσε να στρέψει τον δίσκο με το χέρι στρεφόταν μεν ο δίσκος, αλλά ο κωδικοποιητής δεν τον ακολουθούσε απαραίτητα στη σωστή θέση. Επιπλέον, η επαφή χανόταν καθώς μετά από κάποιες επαναλήψεις η σκουληκόβιδα ξεβιδωνόταν. Αυτό το γεγονός σε συνδυασμό με



το ότι ο κωδικοποιητής είναι μη συνεχής, δηλαδή έχει μια αρχή και ένα τέλος, έχει ως αποτέλεσμα να παρασέρνεται μόνο ο δίσκος και να χάνεται ο σωστός συνδυασμός δοχείου και θέσης κωδικοποιητή. Σε αυτήν την περίπτωση πρέπει να ξεβιδωθεί η κεντρική βίδα των δίσκων, με δοκιμές να επαναφερθεί η σωστή θέση τους, να ξαναβιδωθεί η βίδα εκ νέου και να βιδωθεί και πάλι η σκουληκόβίδα. Μια ενέργεια που μείωσε αλλά δεν εξάλειψε αυτό το πρόβλημα, ήταν η τοποθέτηση μιας μεγαλύτερης με μήκος σκουληκόβιδας, ώστε να βγαίνει προς τα έξω, και η τοποθέτηση ενός περικοχλίου σε αυτό το μέρος της βίδας που εξείχε, ώστε να σφίγγει κοντά στον άξονα. Όπως αναφέρθηκε, συνεχίζει να υπάρχει το πρόβλημα μη άριστης επαφής, συνεπώς ο χρήστης πρέπει να προσέχει και να μην γυρίζει ποτέ τον δίσκο με το χέρι, αλλά σίγουρα τα σφάλματα μειώθηκαν αισθητά.

Οι παραπάνω παρατηρήσεις λοιπόν, πρέπει να ληφθούν υπόψιν για τον πιθανό επανασχεδιασμό της συσκευής που μπορεί να πραγματοποιηθεί, όπως επίσης και το γεγονός ότι σε ένα εργαστήριο που παραλαμβάνει δείγματα για ιστολογία πολύ συχνά ή/και περισσότερα σε ποσότητα, θα ήταν απαραίτητη μία ιστοκινέτα με μεγαλύτερη χωρητικότητα κασετινών.

Για την τελειοποίηση μιας ιστοκινέτας, όπως η συγκεκριμένη, θα πρέπει να εξετασθεί και να σχεδιαστεί ένας τρόπος ώστε να λαμβάνονται απ'ευθείας οι σωστές θερμοκρασίες στο εσωτερικό των θερμαινόμενων δοχείων, χωρίς να προσθέτονται τα σφάλματα προγραμματιστικά έπειτα από δοκιμές, ή να δημιουργηθεί ένα περιβάλλον για τα δοχεία αυτά απόλυτα μονωμένο, ώστε να μην επηρεάζει καθόλου τις αποκλίσεις της θερμοκρασίας η θερμοκρασία δωματίου. Στην συγκεκριμένη περίπτωση είναι πιθανόν να επηρεαστούν αλλά δεν οι αποκλίσεις θα βρίσκονται σε τέτοια επίπεδα που δεν θα δημιουργηθούν προβλήματα για την πειραματική διαδικασία. Τέλος, για την καλύτερη ροή των διαλυμάτων στο εσωτερικό των ιστών, θα μπορούσε να δημιουργηθεί ένα είδος δόνησης είτε στο τεμάχιο συγκράτησης της κασετίνας, είτε στα ίδια τα δοχεία, μια λειτουργία που είναι διαδεδομένη στις μεγάλες ιστοκινέτες που έχουν ήδη βγει στην αγορά και αφορούν εργαστήρια με μεγάλη ιστολογική δραστηριότητα.



11. Βιβλιογραφία

- [1]: K. P. H. Pritzker M.D., F.R.C.P.C., S. Gay M.D., S. A. Jimenez M.D., K. Ostergaard M.D., Ph.D., J.-P. Pelletier M.D., C.S.P.Q., F.R.C.P.C., P. A. Revell Ph.D., D. Salter M.D., F.R.C.Path and W. B. van den Berg Ph.D., *Osteoarthritis cartilage histopathology: grading and staging*^{1,2}, 2005
OsteoArthritis Research Society International, Published by Elsevier Ltd,
doi:10.1016/j.joca.2005.07.014
- [2]: Cathy S. Carlson a,*, Farshid Guilak ', Thomas P. Vail ', Jean F. Gardin ', Virginia B. Kraus, Synovial fluid biomarker levels predict articular cartilage damage following complete medial meniscectomy in the canine knee, *Journal of Orthopaedic Research* 20 (2002) 92--100 , Received 11 May 2000; accepted 23 April 2001, ELSEVIER
- [3]: Χατζοπούλου Ελισάβετ, Διπλωματική Εργασία: *Συσχέτιση των μηχανικών ιδιοτήτων του αρθρικού χόνδρου με τον βαθμό εκφύλισης του*, Ιούλιος 2012, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Μηχανολόγων Μηχανικών
- [4]: Παυλίδης Μιχάλης, *Ιστολογία*, Νοέμβριος 2002, Πανεπιστήμιο Κρήτης, Τμήμα Βιολογίας
- [5]: Ι.Σ. Χαλκιάδης, *Δίοδος και Κυκλώματα Διόδων*, Σημειώσεις ΤΕΙ Χαλκίδας
- [6]: Linus Wong, *1-Wire*® (Protocol), Company: Semiconductor/Maxim
- [7]: Ελευθεριάδης Μιχαήλ, Πτυχιακή Εργασία: *Εφαρμογές ασύρματης τηλεμετρίας στην αναπτυξιακή πλατφόρμα Arduino*, Σχολή Τεχνολογικών Εφαρμογών, Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων, Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης
- [8]: <http://www.vwlowen.co.uk/arduino/ds18b20/ds18b20.htm>
- [9]: [http://www.dfrobot.com/wiki/index.php?title=I2C_TWI_LCD2004_\(SKU:DFR0154\)](http://www.dfrobot.com/wiki/index.php?title=I2C_TWI_LCD2004_(SKU:DFR0154))
- [10]: [http://www.dfrobot.com/wiki/index.php/I2C_TWI_LCD2004_Module_\(Arduino/Gadgeteer_Compatible\)_\(SKU:DFR0154\)](http://www.dfrobot.com/wiki/index.php/I2C_TWI_LCD2004_Module_(Arduino/Gadgeteer_Compatible)_(SKU:DFR0154))
- [11]: <http://www.disabled.gr/lib/?p=22553>
- [12]: http://www.athloclinic.gr/index.php?option=com_content&view=article&id=121&Itemid=30&lang=el
- [13]: <http://www.arthritis.org/conditions-treatments/disease-center/osteoarthritis/>
- [14]: <http://www.maximintegrated.com/products/1-wire/flash/overview/index.cfm>
- [15]: <http://www.hacktronics.com/Tutorials/arduino-1-wire-address-finder.html>
- [16]: <http://www.tigoe.com/pcomp/code/circuits/motors/stepper-motors/>
- [17]: <http://arduino.cc/en/Tutorial/LiquidCrystal>
- [18]: <http://electronics.orgfree.com/hlektronika6.html>
- [19]: <http://www.me.umn.edu/courses/me2011/arduino/technotes/microswitch/>
- [20]: http://www.orthoskag.gr/page.php?menu=left&menu_item=11



- [21]: <http://www.sportsurgery.gr/osteo2.html>
- [22]: <http://cgi.di.uoa.gr/~std06100/Welcome.html>
- [23]: http://milesburton.com/Dallas_Temperature_Control_Library
- [24]: <http://www.me.umn.edu/courses/me2011/arduino/technotes/dcmotors/L293/L293.html>
- [25]: http://www.ee.teihal.gr/labs/electronics/web/downloads/Genika_hlektronika_kef04.pdf
- [26]: <http://www.hlektronika.gr/index.php?page=theory?diode>
- [27]: <http://www.leicabiosystems.com/specimen-preparation/tissue-processing/>
- [28]: <http://www.ihcworld.com/products/Hestion-Processor.htm>
- [29]: <http://www.bio-int.com/tissue.php>



ΠΑΡΑΡΤΗΜΑ Α'

Οδηγίες Χρήσης Ιστοκινέτας

- Σηκώστε το καπάκι.
- Ελέγξτε ότι όλα τα δοχεία έχουν την κατάλληλη ποσότητα διαλύματος ή παραφίνης και συμπληρώστε όπου είναι λιγότερο.
- Κλείστε το καπάκι.
- Τοποθετήστε την κασετίνα στην ειδική υποδοχή.
- Βάλτε την ιστοκινέτα στην πρίζα και συνδέστε το usb του επεξεργαστή με έναν υπολογιστή.
- Πατήστε το κουμπί.
- Η διαδικασία βρίσκεται σε εξέλιξη.
- Μετά το τέλος της διαδικασίας βγάλτε την πρίζα και το usb.
- Βγάλτε την κασετίνα από την υποδοχή.
- Καθαρίστε τη συσκευή με ένα λεπτό πινέλο και ξυλόλη, όπου υπάρχει κερί (παραφίνη).

Προσοχή: Μην ακουμπήσετε το θερμαινόμενο δοχείο για λίγη ώρα, μέχρι να κρυώσει. Υπάρχει κίνδυνος εγκαυμάτων.

Σημείωση 1^η: Μην γυρίζετε ποτέ τον δίσκο με το χέρι. Υπάρχει κίνδυνος να χαθεί η ακριβής θέση του encoder και θα ξαναχρειαστεί «καλιμπράρισμα» για να δουλέψει.

Σημείωση 2^η: Σε περίπτωση που ο βραχίονας δυσκολεύεται στην κίνησή του ελέγξτε την πιθανότητα εμπλοκής των καλωδίων.





ΠΑΡΑΡΤΗΜΑ Β'

Πρωτόκολλο Ιστοκινέτας

Αφυδάτωση:

- 1) 70% αιθανόλη για 30 λεπτά
- 2) 96% αιθανόλη για 30 λεπτά
- 3) 96% αιθανόλη για 30 λεπτά
- 4) 96% αιθανόλη για 30 λεπτά
- 5) 100% αιθανόλη για 30 λεπτά
- 6) 100% αιθανόλη για 30 λεπτά
- 7) 100% αιθανόλη για 30 λεπτά

Καθαρισμός:

- 8) Διάλυμα ξυλόλης για 1 ώρα
- 9) Διάλυμα ξυλόλης για 1 ώρα

Στερεοποίηση:

- 10) Λιωμένη Παραφίνη στους 58 °C για 1 ώρα
- 11) Λιωμένη Παραφίνη στους 58 °C για 1 ώρα





ΠΑΡΑΡΤΗΜΑ Γ'

Ιστολογικό Πρωτόκολλο

A) Μονιμοποίηση (fixation)

- 1) Οι ιστοί αφήνονται σε διάλυμα φορμαλίνης 10% για 4 ώρες.
- 2) **Απασβεστοποίηση:** Προτεινόμενη ουσία είναι η Cal-Ex, Fisher Scientific. Το δείγμα αφήνεται για 6 με 12 ώρες ή και για ολόκληρο το βράδυ (για μικρά δείγματα).
- 3) Αφήνεται να τρέξει νερό βρύσης από 30 έως 60 λεπτά.
- 4) Κρατάμε τους ιστούς σε 70% διάλυμα αιθανόλης για όσο θέλουμε.

B) Επεξεργασία ιστών (tissue processing)

Αφυδάτωση:

- 1) 70% αιθανόλη για 30 λεπτά
- 2) 96% αιθανόλη για 30 λεπτά
- 3) 96% αιθανόλη για 30 λεπτά
- 4) 96% αιθανόλη για 30 λεπτά
- 5) 100% αιθανόλη για 30 λεπτά
- 6) 100% αιθανόλη για 30 λεπτά
- 7) 100% αιθανόλη για 30 λεπτά

Καθαρισμός:

- 8) Διάλυμα ξυλόλης για 1 ώρα
- 9) Διάλυμα ξυλόλης για 1 ώρα

Σταθεροποίηση:

- 10) Λιωμένη Παραφίνη στους 58 °C για 1 ώρα
- 11) Λιωμένη Παραφίνη στους 58 °C για 1 ώρα

Γ) Καλούπια παραφίνης (embedding)

Απαραίτητα:

- μία θερμή πλάκα στους 60 °C,
- μια ψυχρή πλάκα στους 4 °C,



- τα μεταλλικά καλούπια στα οποία θα φτιαχτούν οι κύβοι,
 - λιωμένη παραφίνη στους 58 °C και
 - λαβίδες για να πιάνουμε τους ιστούς.
- 1) Οι πλαστικές κασετίνες τοποθετούνται για λίγο στην θερμή.
 - 2) Αναλόγως με το δείγμα και το μέγεθος του καλουπιού, τοποθετείται μικρή ποσότητα λιωμένης παραφίνης στο κατάλληλο μεταλλικό καλούπι.
 - 3) Αφήνεται για 2 δευτερόλεπτα περίπου για να στερεοποιηθεί λίγο ο πάτος.
 - 4) Τοποθετείται με τη λαβίδα ο ιστός στον πάτο μέχρι να κολλήσει.
 - 5) Το καλούπι πρέπει να σκεπαστεί με την πλαστική κασετίνα και να γεμιστεί με παραφίνη.
 - 6) Τοποθετείται πάνω στην ψυχρή πλάκα μέχρι να στερεοποιηθεί τελείως.
 - 7) Όταν θα είναι έτοιμο το μεταλλικό καλούπι θα βγει με μεγάλη ευκολία.

Δ) Δημιουργία τομών (microtome)

Απαραίτητα:

- Ένα δοχείο με πάγο,
 - ένα δοχείο με κρύο νερό,
 - μια δεξαμενή με ζεστό απεσταγμένο νερό (46-48 °C),
 - διηθητικό χαρτί πάνω στον πάγκο,
 - φορτισμένα πλακίδια,
 - μια λαβίδα και
 - δύο πινέλα.
- 1) Τα καλούπια τοποθετούνται στο δοχείο με τον πάγο.
 - 2) Το καλούπι τοποθετείται σωστά στη θέση του, στο μικροτόμο.
 - 3) Τοποθετείται η ατσάλινη λεπίδα.
 - 4) Ασφαλίζονται οι μοχλοί.
 - 5) Κόβονται χοντρές τομές (15-25 μm) στην άκρη της λεπίδας.
 - 6) Αφού η λεπίδα φτάσει κοντά στον ιστό, κόβονται λεπτές τομές (3-5 μm) με το μεσαίο τμήμα της λεπίδας.
 - 7) Συλλέγεται η «καλή» τομή.
 - 8) Τοποθετείται στο δοχείο με το κρύο νερό.
 - 9) Με τη βοήθεια του πλακιδίου μεταφέρεται στη δεξαμενή με το ζεστό νερό για να ανοίξει καλά.



- 10) Ξανασυλλέγεται με την πλάκα έτσι ώστε να κολλήσει πάνω της.
- 11) Τινάζουμε ελαφρώς η πλάκα, πάνω στο διηθητικό χαρτί, με προσοχή, για να φύγει το νερό .
- 12) Αφήνεται να στεγνώσει όρθια.

E) Χρόση (staining)

Για το μείγμα της αιματοξυλίνης με την εωσίνη:

- 1) Τα πλακίδια με τους ιστούς ζεσταίνονται στον κλίβανο, στους 65 °C για 30 λεπτά.
- 2) Για να αφαιρεθεί το κερί από τον ιστό και να τον ενυδατώσουμε ακολουθούνται με τη σειρά τα παρακάτω βήματα:
 - 4 x Διήθηση σε ξυλόλη για 5 λεπτά (σύνολο 20 λεπτά)
 - 1 πέρασμα με μια σταγόνα ακετόνης
 - 3 x Διήθηση σε 100% αιθανόλη για 5 λεπτά (σύνολο 15 λεπτά)
 - 3 x Διήθηση σε 96% αιθανόλη για 5 λεπτά (σύνολο 15 λεπτά)
 - 1 x Διήθηση σε 70% αιθανόλη για 4 λεπτά
 - 1 x Διήθηση σε 50% αιθανόλη για 4 λεπτά
 - Ξεπλένουμε με τρεχούμενο νερό βρύσης
- 3) Το πλακίδιο βυθίζεται στην αιματοξυλίνη για 2 λεπτά εάν είναι φρέσκια ή για 4 λεπτά αν δεν είναι.
- 4) Ξεπλένεται με τρεχούμενο νερό βρύσης.
- 5) Βυθίζεται για 1-2 δευτερόλεπτα σε όξινη αλκοόλη.
- 6) Ξεπλένεται σε νερό βρύσης για 3 λεπτά, μέχρι να γίνουν οι τομές μπλε.
- 7) Βυθίζεται το πλακίδιο στην εωσίνη για 3 λεπτά.
- 8) Ξεπλένεται με τρεχούμενο νερό βρύσης για 2 λεπτά.
- 9) Γρήγορη αφυδάτωση περνώντας το πλακίδιο από τις αλκοόλες (όχι τις ξυλόλες) του βήματος 2 αλλά με την ανάποδη σειρά, κρατώντας το πλακίδιο για 10 δευτερόλεπτα στο καθένα.

Για το κυανό της τολουιδίνης:

- 1) Τα πλακίδια με τους ιστούς ζεσταίνονται στον κλίβανο, στους 65 °C για 30 λεπτά.
- 2) Για να αφαιρεθεί το κερί από τον ιστό και να τον ενυδατώσουμε ακολουθούμε με τη σειρά τα παρακάτω βήματα:
 - 4 x Διήθηση σε ξυλόλη για 5 λεπτά (σύνολο 20 λεπτά)
 - 1 πέρασμα με μια σταγόνα ακετόνης



- 3 x Διήθηση σε 100% αιθανόλη για 5 λεπτά (σύνολο 15 λεπτά)
 - 3 x Διήθηση σε 96% αιθανόλη για 5 λεπτά (σύνολο 15 λεπτά)
 - 1 x Διήθηση σε 70% αιθανόλη για 4 λεπτά
 - 1 x Διήθηση σε 50% αιθανόλη για 4 λεπτά
 - Ξεπλένεται με τρεχούμενο νερό βρύσης
- 3) Το πλακίδιο βυθίζεται σε απεσταγμένο νερό για 10 λεπτά.
- 4) Εφαρμόζεται 0,25% w/v κυανό της τολουιδίνης στην πλευρά του πλακιδίου που είναι η τομή, ή το πλακίδιο βυθίζεται σε δοχείο με την παραπάνω χρωστική ουσία. Το κρατάμε έτσι για 5 λεπτά.
- 5) Ξεπλένεται σε απεσταγμένο νερό.
- 6) Οι τομές παρατηρούνται στο μικροσκόπιο για να διαπιστωθεί ότι έχουν χρωματιστεί καλά. Εάν ναι, η διαδικασία συνεχίζεται αλλιώς το πλακίδιο ξανατοποθετείται λίγο στη χρωστική.
- 7) Γρήγορη αφυδάτωση περνώντας το πλακίδιο από τις αλκοόλες (όχι τις ξυλόλες) του βήματος 2 αλλά με την ανάποδη σειρά, κρατώντας το πλακίδιο για 10 δευτερόλεπτα στο καθένα.
- 8) Βυθίζεται στην ξυλόλη για 5 λεπτά.
- 9) Τοποθετείται το προστατευτικό γυαλί με την βοήθεια της ειδικής κόλλας.



ΠΑΡΑΡΤΗΜΑ Δ'

Κατασκευαστικά Σχέδια

- Συναρμολογημένο Σύστημα (synarmologimeno_systima)
- Άξονας Συγκράτησης Δίσκων (aksonaki_housing)
- Αποστάτες για Φωτοδιόδους (apost_phot)
- Αποστάτες για την Μικρή Βάση (apostates)
- Αποστάτες για τους Μικροδιακόπτες (apostates_switch)
- Ασανσέρ για την Τοποθέτηση της Κασετίνας (asanser)
- Κάτω Δίσκος (diskos_katw)
- Πάνω Δίσκος (diskos_panw)
- Έδρανο Ολίσθησης (edrano_olisthisis)
- Ενδιάμεση Συγκράτηση (endiamesi_sygkratisi)
- Κάλυμμα Συστήματος (kalymma_systimatos)
- Κάθετος Βραχίονας (kathetos_vraxionas)
- Πλάκα Στήριξης Κινητήρα (plaka_stiriksis_kinitira)
- Κανόνας (rgeab0_5_300_n_2_03)
- Στήριξη για το Ασανσέρ (stiriksi_asanser)
- Έλασμα για τις Φωτοδιόδους (syndesi_fwtodiodwn)
- Θήκη για την Συγκράτηση της Οθόνης (thiki_LCD - Copy)
- Βάση για την Φωλιά των Ρουλεμάν (vasi_gia_housing)
- Μικρή Βάση για το Μονωτικό Υλικό των Θερμαινόμενων Δοχείων (vasi_gia_inox_1)
- Βάση Βηματικού Κινητήρα (vasi_kinit)
- Βάση Συστήματος (vasi_systimatos)
- Βάση Μονωτικού Υλικού για την Τοποθέτηση των Θερμαινόμενων Δοχείων (vasi_thermainomenwn)
- Καπάκι (ΚΑΡΑΚΙ)
- Ρουλεμάν -B6201 (Bearings)
- Housing -BGRBB6201ZZ-40 (Housing)





ΠΑΡΑΡΤΗΜΑ Ε'

Πίνακας Υλικών

- Αλουμίνιο 5083
- Ανοξειδωτος Χάλυβας SS304
- Πλάκες Πυριτικού Ασβεστίου [$k=0.05 \text{ W}/(m.K)$, at $25^{\circ}C$]
- Διαφανές Plexiglass (PMMA)
- Μαύρο Plexiglass (PMMA)
- Βίδες και Περικόχλια διαφόρων μεγεθών (M3,M5)
- Δύο (2) Ανοξειδωτα Δοχεία, ύψους 60 mm, διαμέτρου 50 mm, των 100 ml (BEAKER WITH RIM SS 100ML LOW FORM)
- Εννέα (9) Γυάλινα Δοχεία ύψους 70 mm, διαμέτρου 38 mm, των 50 ml (BEAKER DURAN BOROSILICATE GLASS, WITHOUT SPOUT, TALL FORM), DIN 12331 & ISO 3819
- Τρεις (3) Φωτοαισθητήρες (opto switch, slotted, EE-SX129, OMRON)
- Ένας (1) Κωδικοποιητής 12 θέσεων (adjustable stop mechanical encoder, Series 51, 12 positions, Grayhill)
- Δύο (2) Θερμαντήρες των 50 Watt (HP05-1/10-24) (PTC element enclosure heater,50W, ΩDBK)
- Δύο (2) Κινητήρες Συνεχούς Ρεύματος (DC) των 12 Volt (GHM-01, 200 rpm, Lynxmotion)
- Ένας (1) Βηματικός Κινητήρας των 12 Volt (ST6318, ultraflat stepper motor, Nanotec)
- Διάτρυτη Πλακέτα
- Δύο (2) Δίοδοι 1N40001
- Δύο (2) Τρανζίστορ Darlington (BC517)
- Δύο (2) Μικροδιακόπτες (MP2 Series, H01)
- Δύο (2) Relay G2RL-2, 12Volt
- Ένα (1) Ψηφιακό Θερμόμετρο 1-Wire (DS18B20, Maxim Integrated)
- Ένας (1) Driver L293d
- Ένας (1) Easy Driver (V43)
- Μία (1) Οθόνη LCD (HD44780)
- Θερμοαγώγιμη επιφάνεια (30CMx3CMx0.3MM Thermal Pad CPU GPU DDR RAM CHIP XBOX360)
- Ρουλεμάν (B6201)
- Housing (BGRBB6201ZZ-40)





ΠΑΡΑΡΤΗΜΑ ΣΤ'

Κώδικας σε Περιβάλλον Arduino

```
//CAROUSEL

//libraries used:
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

//for lcd:
#if defined(ARDUINO) && ARDUINO >= 100
#define printByte(args) write(args);
#else
#define printByte(args) print(args,BYTE);
#endif

//Set one wire Pin:
// Data wire is plugged into pin 34 on the Arduino (red-->Vdd, black-->GND, white-->DQ)
#define ONE_WIRE_BUS 34

// Setup a oneWire instance to communicate with any OneWire devices
OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);

// Assign the addresses of your 1-Wire temp sensors. ---->>> SEE 1WIRE ADDRESS FINDER....
DeviceAddress thermSensor2 = { 0x28, 0x98, 0xD4, 0xEB, 0x03, 0x00, 0x00, 0x76 };
DeviceAddress thermSensor1 = { 0x28, 0x10, 0xEC, 0xEB, 0x03, 0x00, 0x00, 0x87 };

//-----LCD-----
LiquidCrystal_I2C lcd(0x20,20,4); // set the LCD address to 0x20 for a 20 chars and 4 line display
```



//-----

//declaration of float variables:

float tempC1; //temperature of the first sensor,for the first thermal beaker

float tempC2; //temperature of the second sensor,for the second thermal beaker

//declaration of long variables:

unsigned long timer1 = 1800000; //30min-1800000ms -how much time we want the cassette to stay into the beaker (dehydration)

unsigned long timer2 = 3600000; //60min-3600000ms -how much time we want the cassette to stay into the beaker (cleaning-fixation)

//Set Pins:

int motor1Pin1 = 3; // pin 2 on L293D

int motor1Pin2 = 4; // pin 7 on L293D

int enable1Pin = 2; // pin 1 on L293D

int motor2Pin1 = 5; // pin 10 on L293D

int motor2Pin2 = 6; // pin 15 on L293D

int enable2Pin = 7; // pin 16 on L293D

int switchupPin = 9; //switch-up

int switchdownPin = 8; //switch-down

int encoder1Pin = 13; //output 1 of encoder

int encoder2Pin = 12; //output 2 of encoder

int encoder3Pin = 11; //output 4 of encoder

int encoder4Pin = 10; //output 8 of encoder

int slot1Pin = 44; //slot 1 to disk

int slot2Pin = 42; //slot 2 to thermal 1

int slot3Pin = 40; //slot 3 to thermal 2

int buttonPin = 45; //push-button to start the program

int motor3Pin1 = 30; //direction Pin - easy driver



```
int motor3Pin2 = 31; //step Pin - easy driver
int enable3Pin = 32; //enable Pin - easy driver

int heater1Pin = 25; //pin for heater 1
int heater2Pin = 26; //pin for heater 2

//declaration of Matrixes:

//matrix of photosensors:
int slotMatrix [3] = { slot1Pin, slot2Pin, slot3Pin};

//matrix of x% v/v of ethanol for lcd printing:
int ethanolMatrix [7] = {70, 96, 96, 96, 100, 100, 100};

//matrix of encoder's position status (see datasheet):
int statusMatrix[12][4] = {

{LOW, LOW, LOW, LOW},

{HIGH, LOW, LOW, LOW},

{LOW, HIGH, LOW, LOW},

{HIGH, HIGH, LOW, LOW},

{ LOW, LOW, HIGH, LOW},

{ HIGH, LOW, HIGH, LOW},

{ LOW, HIGH, HIGH, LOW},

{ HIGH, HIGH, HIGH, LOW},

{LOW, LOW, LOW, HIGH},
```



{HIGH, LOW, LOW, HIGH},

{LOW, HIGH, LOW, HIGH},

{HIGH, HIGH, LOW, HIGH},

};

//Function for printing temperature:_____

void printTemperature(DeviceAddress deviceAddress)

{

float tempC = sensors.getTempC(deviceAddress);

}

//_____

void setup() {

// set the switch as an input:

pinMode(switchupPin, INPUT);

pinMode(switchdownPin, INPUT);

//set the push-button as an input:

pinMode(buttonPin, INPUT);

//set photosensor as an input:

pinMode(slot1Pin, INPUT);

pinMode(slot2Pin, INPUT);

pinMode(slot3Pin, INPUT);

// set encoder as an input:

pinMode(encoder1Pin, INPUT);

pinMode(encoder2Pin, INPUT);

pinMode(encoder3Pin, INPUT);

pinMode(encoder4Pin, INPUT);

// set all the other pins you're using as outputs:



```
pinMode(motor1Pin1, OUTPUT);
pinMode(motor1Pin2, OUTPUT);
pinMode(enable1Pin, OUTPUT);
pinMode(motor2Pin1, OUTPUT);
pinMode(motor2Pin2, OUTPUT);
pinMode(enable2Pin, OUTPUT);
pinMode(motor3Pin1, OUTPUT);
pinMode(motor3Pin2, OUTPUT);
pinMode(enable3Pin, OUTPUT);
pinMode(heater1Pin, OUTPUT);
pinMode(heater2Pin, OUTPUT);

// set enablePin high so that motor can turn on:
digitalWrite(enable1Pin, HIGH);
digitalWrite(enable2Pin, HIGH);
digitalWrite(enable3Pin, HIGH);

//open sensors:
sensors.begin();

// set the resolution of sensors to 10 bit ( 8 up to 12)
sensors.setResolution(thermSensor1, 10);
sensors.setResolution(thermSensor2, 10);

//-----LCD-----
lcd.init();           // initialize the lcd
lcd.backlight();
lcd.home();
lcd.clear();
//-----
}

void loop() {
```



```
//push the button to start the procedure in order to be done once. Otherwise the void loop will run all  
the time
```

```
if (digitalRead(buttonPin) == LOW) {
```

```
    //-----lcd-----
```

```
    //welcome the user while the system initialises the histokinette
```

```
    lcd.clear();
```

```
    lcd.setCursor(0, 1);
```

```
    lcd.print(" WELCOME...");
```

```
    //-----
```

```
//initialise:
```

```
//the cassette goes up
```

```
//DC2 --> go up till switch-up
```

```
while (digitalRead(switchupPin) == HIGH) {
```

```
    digitalWrite(motor2Pin1, HIGH); //direction to go up
```

```
    digitalWrite(motor2Pin2, LOW);
```

```
}
```

```
digitalWrite(motor2Pin1, HIGH); //stop motor 2 (both low OR both high)
```

```
digitalWrite(motor2Pin2, HIGH);
```

```
//disk goes to the first beaker (position No2)
```

```
// DC1 --> go disk to position1
```

```
while (digitalRead(encoder1Pin) == LOW || digitalRead(encoder2Pin) == LOW ||
```

```
digitalRead(encoder3Pin) == LOW || digitalRead(encoder4Pin) == LOW) {
```

```
    digitalWrite(motor1Pin2, HIGH); //OPPOSITE DIRECTION FROM THE REST PROGRAM!!
```

```
    digitalWrite(motor1Pin1, LOW); //move disk
```

```
}
```

```
digitalWrite(motor1Pin2, HIGH);
```

```
digitalWrite(motor1Pin1, HIGH); //stop disk (both low OR both high)
```

```
delay (100);
```

```
//DC1 --> go to position2
```



```
while (digitalRead(encoder1Pin) == HIGH || digitalRead(encoder2Pin) == LOW ||
digitalRead(encoder3Pin) == LOW || digitalRead(encoder4Pin) == LOW) {
    digitalWrite(motor1Pin2, LOW);
    digitalWrite(motor1Pin1,HIGH); //move disk
}
digitalWrite(motor1Pin2, HIGH);
digitalWrite(motor1Pin1,HIGH); //stop disk (both low OR both high)
delay (100);

//arm goes to the position in front of disk
//stepper --> go to slot 1
while (digitalRead(slot1Pin) == LOW) { //while there is not anything which interrupt the first
photosensor move the stepper to the right direction
    //rotate a specific number of microsteps (8 microsteps per step)
    //a 200 step stepper would take 1600 micro steps for one full revolution
    rotate(10, 0.01); //reverse - direction to go left / (number of steps, speed)
}
delay (100); //some delays in order to give some more time to the programmer

//Tissue Processing begins...
int i = 0; //counter for beakers (i=0 --> 1st beaker)

//DEHYDRATION!!!

for(i = 0; i < 7; i++){ //do the following steps "{}" for the first 7 beakers

    // open 1st heater when i=5
if (i == 5) {
    digitalWrite(heater1Pin,HIGH);
}

//-----lcd-----
```



```
// set the cursor to column 0, line 0-->1st
lcd.setCursor(0, 0);
// Print the number of beaker
lcd.print(String("BEAKER " + String(i+1) + " (" + String(i+1) + "/" + String(i+1) + ")")); //lcd print
//-----
// set the cursor to column 0, line 1-->2nd
lcd.setCursor(0, 1);
// print the kind of beaker
lcd.print(String("ETHANOL " + String(ethanolMatrix[i]) + "% ")); //lcd print
//-----

//go down
while (digitalRead(switchdownPin) == HIGH) {
  digitalWrite(motor2Pin1, LOW); //direction to go down
  digitalWrite(motor2Pin2, HIGH);
}
digitalWrite(motor2Pin1, HIGH); //stop motor 2 (both low OR both high)
digitalWrite(motor2Pin2, HIGH);
delay (100);

//wait 30 min
unsigned long timer30 = millis(); //save to variable timer30 the number of milliseconds passed since
the programme started
while (millis() - timer30 <= timer1) { //compare the difference in order to know much time you are in
the "while" loop. If the correct time has passed (timer1 or timer2) continue.
  lcd.setCursor(0, 2);
  lcd.print(String("Time Left: " + String(((timer1+timer30-millis())/60000)+1) + " Min ")); //lcd print
}

//go up
while (digitalRead(switchupPin) == HIGH) {
  digitalWrite(motor2Pin1, HIGH); //direction to go up
  digitalWrite(motor2Pin2, LOW);
}
```



```
digitalWrite(motor2Pin1, HIGH); //stop motor 2
digitalWrite(motor2Pin2, HIGH);
delay (100);

//disk go to next position
int y = i + 2; //variable y gives the correct row of statusMatrix/y=i+2 because we have already used
the first 2 separately for initialising

while (digitalRead(encoder1Pin) == statusMatrix[y][0] || digitalRead(encoder2Pin) ==
statusMatrix[y][1] || digitalRead(encoder3Pin) == statusMatrix[y][2] || digitalRead(encoder4Pin) ==
statusMatrix[y][3]) {
    digitalWrite(motor1Pin2,LOW); //move disk
    digitalWrite(motor1Pin1, HIGH);
}
digitalWrite(motor1Pin2, HIGH);
digitalWrite(motor1Pin1,HIGH); //stop disk (both low OR both high)
delay (100);
}

//CLEANING!!!

//when cleaning begins open both heaters

if (i == 7) { //do the following steps "{}" for the 8th beaker
//-----LCD print-----
lcd.setCursor(0, 0);
lcd.print("BEAKER 8 (8/11)");
lcd.setCursor(0, 1);
lcd.print("XYLENE  ");
//-----
digitalWrite(heater1Pin,HIGH); //open heater 1
digitalWrite(heater2Pin,HIGH); //open heater 2
//go down
while (digitalRead(switchdownPin) == HIGH) {
```



```
digitalWrite(motor2Pin1, LOW); //direction to go down
digitalWrite(motor2Pin2, HIGH);
}
digitalWrite(motor2Pin1, HIGH); //stop motor 2 (both low OR both high)
digitalWrite(motor2Pin2, HIGH);
delay (100);

//wait 60 min
unsigned long timer60 = millis(); //save to variable timer60 the number of milliseconds passed since
the programme started
while (millis() - timer60 <= timer2) { //compare the difference in order to know much time you are in
the "while" loop. If the correct time has passed (timer1 or timer2) continue.
    lcd.setCursor(0, 2);
    lcd.print(String("Time Left: " + String(((timer2+timer60-millis())/60000)+1) + " Min  ")); //lcd
print
}

//go up
while (digitalRead(switchupPin) == HIGH) {
    digitalWrite(motor2Pin1, HIGH); //direction to go up
    digitalWrite(motor2Pin2, LOW);
}
digitalWrite(motor2Pin1, HIGH); //stop motor 2
digitalWrite(motor2Pin2, HIGH);
delay (100);

//disk go to next position
int z = i + 2; //variable z gives the correct row of statusMatrix/z=i+2 because we have already used
the first 2 separately for initialising
while (digitalRead(encoder1Pin) == statusMatrix[z][0] || digitalRead(encoder2Pin) ==
statusMatrix[z][1] || digitalRead(encoder3Pin) == statusMatrix[z][2] || digitalRead(encoder4Pin) ==
statusMatrix[z][3]) {
    digitalWrite(motor1Pin2,LOW); //move disk
    digitalWrite(motor1Pin1, HIGH);
```




```
}
digitalWrite(motor1Pin2, HIGH);
digitalWrite(motor1Pin1,HIGH); //stop disk (both low OR both high)
i = i + 1;
}

if (i == 8) { //do the following steps "{}" for the 9th beaker

//-----LCD print-----
lcd.setCursor(0, 0);
lcd.print("BEAKER 9 (9/11)");
lcd.setCursor(0, 1);
lcd.print("XYLENE");
//-----
digitalWrite(heater1Pin,HIGH); //open heater 1
digitalWrite(heater2Pin,HIGH); //open heater 2

//go down
while (digitalRead(switchdownPin) == HIGH) {
digitalWrite(motor2Pin1, LOW); //direction to go down
digitalWrite(motor2Pin2, HIGH);
}
digitalWrite(motor2Pin1, HIGH); //stop motor 2
digitalWrite(motor2Pin2, HIGH);
delay (100);

//wait 60 min
unsigned long timer = millis(); //save to variable timer the number of milliseconds passed since the
programme started
while (millis() - timer <= timer2) { //compare the difference in order to know much time you are in the
"while" loop. If the correct time has passed (timer1 or timer2) continue.
lcd.setCursor(0, 2);
lcd.print(String("Time Left: " + String(((timer2+timer-millis())/60000)+1) + " Min  ")); //lcd print

//Read Temperature:
```



```
delay(2000);
sensors.requestTemperatures();
lcd.setCursor(0, 3);
lcd.print("Thermal No1: "); //lcd print
tempC1 = sensors.getTempC(thermSensor1);
lcd.print(tempC1+ 7.5); //add the difference because the sensor is outside of the thermal beaker in
order the lcd to show the temperature inside and no outside of the beaker
lcd.print("oC");

//stabilise temperature of thermal beaker 1
if (tempC1 > 50.50) {
  digitalWrite(heater1Pin,LOW); //if the temperature of the sensor is more than 50,50 deegres of
Celsius then close the heater
}
else {
  digitalWrite(heater1Pin,HIGH); //if the temperature of the sensor is less or equal to 50,50 deegres of
Celsius then open the heater again
}
}

//go up
while (digitalRead(switchupPin) == HIGH) {
  digitalWrite(motor2Pin1, HIGH); //direction to go up
  digitalWrite(motor2Pin2, LOW);
}
digitalWrite(motor2Pin1, HIGH); //stop motor 2
digitalWrite(motor2Pin2, HIGH);
delay (100);

//disk go to next position
int z = i + 2; //variable z gives the correct row of statusMatrix/z=i+2 because we have already
used the first 2 separately for initialising
while (digitalRead(encoder1Pin) == statusMatrix[z][0] || digitalRead(encoder2Pin) ==
statusMatrix[z][1] || digitalRead(encoder3Pin) == statusMatrix[z][2] || digitalRead(encoder4Pin) ==
statusMatrix[z][3]) {
```



```
digitalWrite(motor1Pin2,LOW); //move disk
digitalWrite(motor1Pin1, HIGH);
}
digitalWrite(motor1Pin2, HIGH);
digitalWrite(motor1Pin1,HIGH); //stop disk
i = i + 1;
}

//FIXATION!!!!
int k = 0; //variable to show the position of the arm using the slotmatrix

if (i == 9) { //do the following steps "{}" for the 10th beaker
digitalWrite(heater1Pin,HIGH); //open heater 1
digitalWrite(heater2Pin,HIGH); //open heater 2
//-----lcd print-----
lcd.setCursor(0, 0);
lcd.print("THERMAL 1 (10/11)");
//-----
//go to the right thermal
k = k + 1;
while (digitalRead(slotMatrix[k]) == LOW) {
rotate(-10, 0.01); //direction to go right
}
delay(100);

//go down
while (digitalRead(switchdownPin) == HIGH) {
digitalWrite(motor2Pin1, LOW); //direction to go down
digitalWrite(motor2Pin2, HIGH);
}
digitalWrite(motor2Pin1, HIGH); //stop motor 2
digitalWrite(motor2Pin2, HIGH);
delay (100);

//wait 60 min
```



```
unsigned long tim = millis(); //save to variable tim the number of milliseconds passed since the
programme started
while (millis() - tim <= timer2) { //compare the difference in order to know much time you are in the
"while" loop. If the correct time has passed (timer1 or timer2) continue.
    lcd.setCursor(0, 2);
    lcd.print(String("Time Left: " + String(((timer2+tim-millis())/60000)+1) + " Min  ")); //lcd print

//Read Temperature...
delay(2000);
sensors.requestTemperatures();
    lcd.setCursor(0, 1);
    lcd.print("PARAFFIN ");
    tempC1 = sensors.getTempC(thermSensor1);
    lcd.print(tempC1 + 7.5); //add the right difference because the sensor is outside of the thermal beaker
in order the lcd to show the temperature inside and no outside of the beaker
    lcd.print("oC");
    lcd.setCursor(0, 3);
    lcd.print("Thermal No2: ");
    tempC2 = sensors.getTempC(thermSensor2);
    lcd.print(tempC2 + 14.50); //add the right difference because the sensor is outside of the thermal
beaker in order the lcd to show the temperature inside and no outside of the beaker
    lcd.print("oC");

//check temperature
//continue to stabilise the temperature of 1st thermal and also stabilise the second one's
//open and close the heater in order to be at the right temperature
if (tempC1 > 50.50) { //if the temperature of the sensor is more than 50,50 deegres of Celsius then
close the heater
    digitalWrite(heater1Pin,LOW);
}
else {
    digitalWrite(heater1Pin,HIGH); //if the temperature of the sensor is less or equal to 50,50 deegres of
Celsius then open the heater again
}
```



```
if (tempC2 > 43.50) { //if the temperature of the sensor is more than 43,50 deegres of Celsius then
close the heater
  digitalWrite(heater2Pin,LOW);
}
else {
  digitalWrite(heater2Pin,HIGH); //if the temperature of the sensor is less or equal to 43,50 deegres of
Celsius then open the heater again
}
}

//go up
while (digitalRead(switchupPin) == HIGH) {
  digitalWrite(motor2Pin1, HIGH); //direction to go up
  digitalWrite(motor2Pin2, LOW);
}
digitalWrite(motor2Pin1, HIGH); //stop motor 2
digitalWrite(motor2Pin2, HIGH);
delay (100);
i = i + 1;
}

if (i == 10) { //do the following steps "{}" for the 11th beaker
  //-----lcd print-----
  lcd.setCursor(0, 0);
  lcd.print("THERMAL 2 (11/11)");
  lcd.setCursor(0, 3);
  lcd.print(" "); //print those spaces in order to clean the previous information on the screen
  //-----
  digitalWrite(heater2Pin,HIGH); //open heater 2
  digitalWrite(heater1Pin,LOW); //close heater 1

  //go to the right thermal
  k = k + 1;
  while (digitalRead(slotMatrix[k]) == LOW) {
```



```
rotate(-10, 0.01); //direction to go right
}
delay(100);

//go down
while (digitalRead(switchdownPin) == HIGH) {
  digitalWrite(motor2Pin1, LOW); //direction to go down
  digitalWrite(motor2Pin2, HIGH);
}
digitalWrite(motor2Pin1, HIGH); //stop motor 2
digitalWrite(motor2Pin2, HIGH);
delay (100);

//wait 60 min
unsigned long tim1 = millis(); //save to variable tim1 the number of milliseconds passed since the
programme started
while (millis() - tim1 <= timer2) { //compare the difference in order to know much time you are in the
"while" loop. If the correct time has passed (timer1 or timer2) continue.
  lcd.setCursor(0, 2);
  lcd.print(String("Time Left: " + String(((timer2+tim1-millis())/60000)+1) + " Min  ")); //lcd print

//Read Temperature
delay(2000);
sensors.requestTemperatures();
lcd.setCursor(0, 1);
lcd.print("PARAFFIN ");
tempC2 = sensors.getTempC(thermSensor2);
lcd.print(tempC2 + 14.50); //add the right difference because the sensor is outside of the thermal
beaker in order the lcd to show the temperature inside and no outside of the beaker
lcd.print(" oC");

//check temperature
//continue to stabilise the temperature of 2nd thermal
if (tempC2 > 43.50) { //if the temperature of the sensor is more than 43,50 deegres of Celsius then
close the heater
```



```
digitalWrite(heater2Pin,LOW);
}
else {
  digitalWrite(heater2Pin,HIGH); //if the temperature of the sensor is less or equal to 43,50 deegres of
Celsius then open the heater again
}
}

//go up
while (digitalRead(switchupPin) == HIGH) {
  digitalWrite(motor2Pin1, HIGH); //direction to go up
  digitalWrite(motor2Pin2, LOW);
}
digitalWrite(motor2Pin1, HIGH); //stop motor 2
digitalWrite(motor2Pin2, HIGH);
delay (100);
i = i + 1;
}

if (i == 11) {
  digitalWrite(heater2Pin,LOW); //close heater 2
//-----lcd print-----
  lcd.clear();
  lcd.setCursor(0, 1);
  lcd.print("  READY!"); //when the procedure is finished
//-----
} //end of “i=11-if”
} //end of “button-if”
} //end of void loop

//rotate function for stepper motor

void rotate(int steps, float speed){
  //rotate a specific number of microsteps (8 microsteps per step) - (negative for reverse movement)
  //speed is any number from .01 -> 1 with 1 being fastest - Slower is stronger
```



```
int dir = (steps > 0)? HIGH:LOW;
steps = abs(steps);

digitalWrite(30,dir);

float usDelay = (1/speed) * 70;

for(int i=0; i < steps; i++){
  digitalWrite(31, HIGH);
  delayMicroseconds(usDelay);

  digitalWrite(31, LOW);
  delayMicroseconds(usDelay);
}
}
```




ΠΑΡΑΡΤΗΜΑ Ζ'

Έτοιμες Βιβλιοθήκες που Χρησιμοποιήθηκαν στον Κώδικα

Βιβλιοθήκη OneWire:

```
#ifndef OneWire_h
#define OneWire_h

#include <inttypes.h>

#if ARDUINO >= 100
#include "Arduino.h" // for delayMicroseconds,
digitalPinToBitMask, etc
#else
#include "WProgram.h" // for delayMicroseconds
#include "pins_arduino.h" // for digitalPinToBitMask, etc
#endif

// You can exclude certain features from OneWire. In theory, this
// might save some space. In practice, the compiler automatically
// removes unused code (technically, the linker, using -fdata-
sections
// and -ffunction-sections when compiling, and Wl,--gc-sections
// when linking), so most of these will not result in any code size
// reduction. Well, unless you try to use the missing features
// and redesign your program to not need them! ONEWIRE_CRC8_TABLE
// is the exception, because it selects a fast but large algorithm
// or a small but slow algorithm.

// you can exclude onewire_search by defining that to 0
#ifndef ONEWIRE_SEARCH
#define ONEWIRE_SEARCH 1
#endif

// You can exclude CRC checks altogether by defining this to 0
#ifndef ONEWIRE_CRC
#define ONEWIRE_CRC 1
#endif

// Select the table-lookup method of computing the 8-bit CRC
// by setting this to 1. The lookup table enlarges code size by
// about 250 bytes. It does NOT consume RAM (but did in very
// old versions of OneWire). If you disable this, a slower
// but very compact algorithm is used.
#ifndef ONEWIRE_CRC8_TABLE
#define ONEWIRE_CRC8_TABLE 1
#endif

// You can allow 16-bit CRC checks by defining this to 1
// (Note that ONEWIRE_CRC must also be 1.)
#ifndef ONEWIRE_CRC16
```



```
#define ONEWIRE_CRC16 1
#endif

#define FALSE 0
#define TRUE 1

// Platform specific I/O definitions

#if defined(__AVR__)
#define PIN_TO_BASEREG(pin)
(portInputRegister(digitalPinToPort(pin)))
#define PIN_TO_BITMASK(pin) (digitalPinToBitMask(pin))
#define IO_REG_TYPE uint8_t
#define IO_REG_ASM asm("r30")
#define DIRECT_READ(base, mask) (((*(base)) & (mask)) ? 1 :
0)
#define DIRECT_MODE_INPUT(base, mask) ((*((base)+1)) &= ~(mask))
#define DIRECT_MODE_OUTPUT(base, mask) ((*((base)+1)) |= (mask))
#define DIRECT_WRITE_LOW(base, mask) ((*((base)+2)) &= ~(mask))
#define DIRECT_WRITE_HIGH(base, mask) ((*((base)+2)) |= (mask))

#elif defined(__MK20DX128__)
#define PIN_TO_BASEREG(pin) (portOutputRegister(pin))
#define PIN_TO_BITMASK(pin) (1)
#define IO_REG_TYPE uint8_t
#define IO_REG_ASM
#define DIRECT_READ(base, mask) (*( (base)+512))
#define DIRECT_MODE_INPUT(base, mask) (*( (base)+640) = 0)
#define DIRECT_MODE_OUTPUT(base, mask) (*( (base)+640) = 1)
#define DIRECT_WRITE_LOW(base, mask) (*( (base)+256) = 1)
#define DIRECT_WRITE_HIGH(base, mask) (*( (base)+128) = 1)

#elif defined(__SAM3X8E__)
// Arduino 1.5.1 may have a bug in delayMicroseconds() on Arduino
Due.
//
http://arduino.cc/forum/index.php/topic,141030.msg1076268.html#msg1076268
// If you have trouble with OneWire on Arduino Due, please check the
// status of delayMicroseconds() before reporting a bug in OneWire!
#define PIN_TO_BASEREG(pin) (&(digitalPinToPort(pin) -
>PIO_PER))
#define PIN_TO_BITMASK(pin) (digitalPinToBitMask(pin))
#define IO_REG_TYPE uint32_t
#define IO_REG_ASM
#define DIRECT_READ(base, mask) (((*( (base)+15)) & (mask)) ?
1 : 0)
#define DIRECT_MODE_INPUT(base, mask) ((* (base)+5) = (mask))
#define DIRECT_MODE_OUTPUT(base, mask) ((* (base)+4) = (mask))
#define DIRECT_WRITE_LOW(base, mask) ((* (base)+13) = (mask))
#define DIRECT_WRITE_HIGH(base, mask) ((* (base)+12) = (mask))
#endif
#define PROGMEM
#define PROGMEM
#endif
```



```
#ifndef pgm_read_byte
#define pgm_read_byte(addr) (*(const uint8_t *) (addr))
#endif

#elif defined(__PIC32MX__)
#define PIN_TO_BASEREG(pin)
(portModeRegister(digitalPinToPort(pin)))
#define PIN_TO_BITMASK(pin) (digitalPinToBitMask(pin))
#define IO_REG_TYPE uint32_t
#define IO_REG_ASM
#define DIRECT_READ(base, mask) (((*(base+4) & (mask)) ? 1 :
0) //PORTX + 0x10
#define DIRECT_MODE_INPUT(base, mask) (*(base+2) = (mask))
//TRISXSET + 0x08
#define DIRECT_MODE_OUTPUT(base, mask) (*(base+1) = (mask))
//TRISXCLR + 0x04
#define DIRECT_WRITE_LOW(base, mask) (*(base+8+1) = (mask))
//LATXCLR + 0x24
#define DIRECT_WRITE_HIGH(base, mask) (*(base+8+2) = (mask))
//LATXSET + 0x28

#else
#error "Please define I/O register types here"
#endif

class OneWire
{
private:
    IO_REG_TYPE bitmask;
    volatile IO_REG_TYPE *baseReg;

#if ONEWIRE_SEARCH
    // global search state
    unsigned char ROM_NO[8];
    uint8_t LastDiscrepancy;
    uint8_t LastFamilyDiscrepancy;
    uint8_t LastDeviceFlag;
#endif

public:
    OneWire( uint8_t pin);

    // Perform a 1-Wire reset cycle. Returns 1 if a device responds
    // with a presence pulse. Returns 0 if there is no device or the
    // bus is shorted or otherwise held low for more than 250uS
    uint8_t reset(void);

    // Issue a 1-Wire rom select command, you do the reset first.
    void select(const uint8_t rom[8]);

    // Issue a 1-Wire rom skip command, to address all on bus.
    void skip(void);
```



```
// Write a byte. If 'power' is one then the wire is held high at
// the end for parasitically powered devices. You are responsible
// for eventually depowering it by calling depower() or doing
// another read or write.
void write(uint8_t v, uint8_t power = 0);

void write_bytes(const uint8_t *buf, uint16_t count, bool power =
0);

// Read a byte.
uint8_t read(void);

void read_bytes(uint8_t *buf, uint16_t count);

// Write a bit. The bus is always left powered at the end, see
// note in write() about that.
void write_bit(uint8_t v);

// Read a bit.
uint8_t read_bit(void);

// Stop forcing power onto the bus. You only need to do this if
// you used the 'power' flag to write() or used a write_bit()
call
// and aren't about to do another read or write. You would rather
// not leave this powered if you don't have to, just in case
// someone shorts your bus.
void depower(void);

#if ONEWIRE_SEARCH
// Clear the search state so that if will start from the
beginning again.
void reset_search();

// Setup the search to find the device type 'family_code' on the
next call
// to search(*newAddr) if it is present.
void target_search(uint8_t family_code);

// Look for the next device. Returns 1 if a new address has been
// returned. A zero might mean that the bus is shorted, there are
// no devices, or you have already retrieved all of them. It
// might be a good idea to check the CRC to make sure you didn't
// get garbage. The order is deterministic. You will always get
// the same devices in the same order.
uint8_t search(uint8_t *newAddr);
#endif

#if ONEWIRE_CRC
// Compute a Dallas Semiconductor 8 bit CRC, these are used in
the
// ROM and scratchpad registers.
static uint8_t crc8(const uint8_t *addr, uint8_t len);
```



```
#if ONEWIRE_CRC16
    // Compute the 1-Wire CRC16 and compare it against the received
    CRC.
    // Example usage (reading a DS2408):
    //     // Put everything in a buffer so we can compute the CRC
    // easily.
    //     uint8_t buf[13];
    //     buf[0] = 0xF0;    // Read PIO Registers
    //     buf[1] = 0x88;    // LSB address
    //     buf[2] = 0x00;    // MSB address
    //     WriteBytes(net, buf, 3);    // Write 3 cmd bytes
    //     ReadBytes(net, buf+3, 10); // Read 6 data bytes, 2 0xFF, 2
CRC16
    //     if (!CheckCRC16(buf, 11, &buf[11])) {
    //         // Handle error.
    //     }
    //
    // @param input - Array of bytes to checksum.
    // @param len - How many bytes to use.
    // @param inverted_crc - The two CRC16 bytes in the received
data.
    //
    //         This should just point into the received
data,
    //
    //         *not* at a 16-bit integer.
    // @param crc - The crc starting value (optional)
    // @return True, iff the CRC matches.
    static bool check_crc16(const uint8_t* input, uint16_t len, const
uint8_t* inverted_crc, uint16_t crc = 0);

    // Compute a Dallas Semiconductor 16 bit CRC. This is required
to check
    // the integrity of data received from many 1-Wire devices. Note
that the
    // CRC computed here is *not* what you'll get from the 1-Wire
network,
    // for two reasons:
    // 1) The CRC is transmitted bitwise inverted.
    // 2) Depending on the endian-ness of your processor, the
binary
    //     representation of the two-byte return value may have a
different
    //     byte order than the two bytes you get from 1-Wire.
    // @param input - Array of bytes to checksum.
    // @param len - How many bytes to use.
    // @param crc - The crc starting value (optional)
    // @return The CRC16, as defined by Dallas Semiconductor.
    static uint16_t crc16(const uint8_t* input, uint16_t len,
uint16_t crc = 0);
#endif
#endif
};

#endif
```



Βιβλιοθήκη AddressFinder:

```
#include <OneWire.h>

//init the one wire interface on pin 8
OneWire ow(8);

void setup(void) {
  Serial.begin(9600);
  lookUpSensors();
}

void lookUpSensors(){
  byte address[8];
  int i=0;
  byte ok = 0, tmp = 0;
  //start the search
  Serial.println("--Search started--");
  while (ow.search(address)){
    tmp = 0;
    //0x10 = DS18S20
    if (address[0] == 0x10){
      Serial.print("Device is a DS18S20 : ");
      tmp = 1;
    } else {
      //0x28 = DS18B20
      if (address[0] == 0x28){
        Serial.print("Device is a DS18B20 : ");
        tmp = 1;
      }
    }
  }
  //display the address, if tmp is ok
  if (tmp == 1){
    if (OneWire::crc8(address, 7) != address[7]){
      Serial.println("but it doesn't have a valid CRC!");
    }
  }
}
```



```
} else {
  //all is ok, display it
  for (i=0;i<8;i++){
    if (address[i] < 9){
      Serial.print("0");
    }
    Serial.print(address[i],HEX);
    if (i<7){
      Serial.print("-");
    }
  }
  Serial.println("");
  ok = 1;
}
} //end if tmp
} //end while
if (ok == 0){
  Serial.println("No devices were found");
}
Serial.println("--Search ended--");
}

void loop(void) {
  //do nothing :)
}
```

[Βιβλιοθήκη DallasTemperature:](#)

```
#ifndef DallasTemperature_h
#define DallasTemperature_h

#define DALLASTEMPLIBVERSION "3.7.2"

// This library is free software; you can redistribute it and/or
// modify it under the terms of the GNU Lesser General Public
// License as published by the Free Software Foundation; either
// version 2.1 of the License, or (at your option) any later version.

// set to true to include code for new and delete operators
#ifndef REQUIRESNEW
#define REQUIRESNEW false
```



```
#endif

// set to true to include code implementing alarm search functions
#ifndef REQUIRESALARMS
#define REQUIRESALARMS true
#endif

#include <inttypes.h>
#include <OneWire.h>

// Model IDs
#define DS18S20MODEL 0x10
#define DS18B20MODEL 0x28
#define DS1822MODEL 0x22

// OneWire commands
#define STARTCONVO      0x44 // Tells device to take a temperature
reading and put it on the scratchpad
#define COPYSCRATCH     0x48 // Copy EEPROM
#define READSCRATCH     0xBE // Read EEPROM
#define WRITESCRATCH    0x4E // Write to EEPROM
#define RECALLSCRATCH   0xB8 // Reload from last known
#define READPOWERSUPPLY 0xB4 // Determine if device needs parasite
power
#define ALARMSEARCH     0xEC // Query bus for devices with an alarm
condition

// Scratchpad locations
#define TEMP_LSB        0
#define TEMP_MSB        1
#define HIGH_ALARM_TEMP 2
#define LOW_ALARM_TEMP 3
#define CONFIGURATION   4
#define INTERNAL_BYTE   5
#define COUNT_REMAIN    6
#define COUNT_PER_C     7
#define SCRATCHPAD_CRC  8

// Device resolution
#define TEMP_9_BIT  0x1F // 9 bit
#define TEMP_10_BIT 0x3F // 10 bit
#define TEMP_11_BIT 0x5F // 11 bit
#define TEMP_12_BIT 0x7F // 12 bit

// Error Codes
#define DEVICE_DISCONNECTED -127

typedef uint8_t DeviceAddress[8];

class DallasTemperature
{
public:

    DallasTemperature(OneWire*);
```




```
// initialise bus
void begin(void);

// returns the number of devices found on the bus
uint8_t getDeviceCount(void);

// Is a conversion complete on the wire?
bool isConversionComplete(void);

// returns true if address is valid
bool validAddress(uint8_t*);

// finds an address at a given index on the bus
bool getAddress(uint8_t*, const uint8_t);

// attempt to determine if the device at the given address is
connected to the bus
bool isConnected(uint8_t*);

// attempt to determine if the device at the given address is
connected to the bus
// also allows for updating the read scratchpad
bool isConnected(uint8_t*, uint8_t*);

// read device's scratchpad
void readScratchPad(uint8_t*, uint8_t*);

// write device's scratchpad
void writeScratchPad(uint8_t*, const uint8_t*);

// read device's power requirements
bool readPowerSupply(uint8_t*);

// get global resolution
uint8_t getResolution();

// set global resolution to 9, 10, 11, or 12 bits
void setResolution(uint8_t);

// returns the device resolution, 9-12
uint8_t getResolution(uint8_t*);

// set resolution of a device to 9, 10, 11, or 12 bits
bool setResolution(uint8_t*, uint8_t);

// sets/gets the waitForConversion flag
void setWaitForConversion(bool);
bool getWaitForConversion(void);

// sets/gets the checkForConversion flag
void setCheckForConversion(bool);
bool getCheckForConversion(void);
```



```
// sends command for all devices on the bus to perform a
temperature conversion
void requestTemperatures(void);

// sends command for one device to perform a temperature conversion
by address
bool requestTemperaturesByAddress(uint8_t*);

// sends command for one device to perform a temperature conversion
by index
bool requestTemperaturesByIndex(uint8_t);

// returns temperature in degrees C
float getTempC(uint8_t*);

// returns temperature in degrees F
float getTempF(uint8_t*);

// Get temperature for device index (slow)
float getTempCByIndex(uint8_t);

// Get temperature for device index (slow)
float getTempFByIndex(uint8_t);

// returns true if the bus requires parasite power
bool isParasitePowerMode(void);

bool isConversionAvailable(uint8_t*);

#ifdef REQUIRESALARMS

typedef void AlarmHandler(uint8_t*);

// sets the high alarm temperature for a device
// accepts a char. valid range is -55C - 125C
void setHighAlarmTemp(uint8_t*, const char);

// sets the low alarm temperature for a device
// accepts a char. valid range is -55C - 125C
void setLowAlarmTemp(uint8_t*, const char);

// returns a signed char with the current high alarm temperature
for a device
// in the range -55C - 125C
char getHighAlarmTemp(uint8_t*);

// returns a signed char with the current low alarm temperature for
a device
// in the range -55C - 125C
char getLowAlarmTemp(uint8_t*);

// resets internal variables used for the alarm search
void resetAlarmSearch(void);
```



```
// search the wire for devices with active alarms
bool alarmSearch(uint8_t*);

// returns true if ia specific device has an alarm
bool hasAlarm(uint8_t*);

// returns true if any device is reporting an alarm on the bus
bool hasAlarm(void);

// runs the alarm handler for all devices returned by alarmSearch()
void processAlarms(void);

// sets the alarm handler
void setAlarmHandler(AlarmHandler *);

// The default alarm handler
static void defaultAlarmHandler(uint8_t*);

#endif

// convert from celcius to fahrenheit
static float toFahrenheit(const float);

// convert from fahrenheit to celsius
static float toCelsius(const float);

#ifdef REQUIRESNEW

// initalize memory area
void* operator new (unsigned int);

// delete memory reference
void operator delete(void*);

#endif

private:
typedef uint8_t ScratchPad[9];

// parasite power on or off
bool parasite;

// used to determine the delay amount needed to allow for the
// temperature conversion to take place
uint8_t bitResolution;

// used to requestTemperature with or without delay
bool waitForConversion;

// used to requestTemperature to dynamically check if a conversion
is complete
bool checkForConversion;

// count of devices on the bus
```



```
uint8_t devices;

// Take a pointer to one wire instance
OneWire* _wire;

// reads scratchpad and returns the temperature in degrees C
float calculateTemperature(uint8_t*, uint8_t*);

void    blockTillConversionComplete(uint8_t*,uint8_t*);

#ifdef REQUIRESALARMS

// required for alarmSearch
uint8_t alarmSearchAddress[8];
char alarmSearchJunction;
uint8_t alarmSearchExhausted;

// the alarm handler function pointer
AlarmHandler *_AlarmHandler;

#endif

};
#endif
```

Βιβλιοθήκη Wire:

```
/*
TwoWire.h - TWI/I2C library for Arduino & Wiring
Copyright (c) 2006 Nicholas Zambetti. All right reserved.

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-
1301 USA

Modified 2012 by Todd Krein (todd@krein.org) to implement repeated
starts
*/

#ifndef TwoWire_h
#define TwoWire_h
```



```
#include <inttypes.h>
#include "Stream.h"

#define BUFFER_LENGTH 32

class TwoWire : public Stream
{
private:
    static uint8_t rxBuffer[];
    static uint8_t rxBufferIndex;
    static uint8_t rxBufferLength;

    static uint8_t txAddress;
    static uint8_t txBuffer[];
    static uint8_t txBufferIndex;
    static uint8_t txBufferLength;

    static uint8_t transmitting;
    static void (*user_onRequest)(void);
    static void (*user_onReceive)(int);
    static void onRequestService(void);
    static void onReceiveService(uint8_t*, int);
public:
    TwoWire();
    void begin();
    void begin(uint8_t);
    void begin(int);
    void beginTransmission(uint8_t);
    void beginTransmission(int);
    uint8_t endTransmission(void);
    uint8_t endTransmission(uint8_t);
    uint8_t requestFrom(uint8_t, uint8_t);
    uint8_t requestFrom(uint8_t, uint8_t, uint8_t);
    uint8_t requestFrom(int, int);
    uint8_t requestFrom(int, int, int);
    virtual size_t write(uint8_t);
    virtual size_t write(const uint8_t *, size_t);
    virtual int available(void);
    virtual int read(void);
    virtual int peek(void);
    virtual void flush(void);
    void onReceive( void (*) (int) );
    void onRequest( void (*) (void) );

    inline size_t write(unsigned long n) { return write((uint8_t)n); }
}

    inline size_t write(long n) { return write((uint8_t)n); }
    inline size_t write(unsigned int n) { return write((uint8_t)n); }
    inline size_t write(int n) { return write((uint8_t)n); }
    using Print::write;
};

extern TwoWire Wire;
```



```
#endif
```

Βιβλιοθήκη LiquidCrystal_I2C:

```
//DFRobot.com
#ifndef LiquidCrystal_I2C_h
#define LiquidCrystal_I2C_h

#include <inttypes.h>
#include "Print.h"
#include <Wire.h>

// commands
#define LCD_CLEARDISPLAY 0x01
#define LCD_RETURNHOME 0x02
#define LCD_ENTRYMODESET 0x04
#define LCD_DISPLAYCONTROL 0x08
#define LCD_CURSORSHIFT 0x10
#define LCD_FUNCTIONSET 0x20
#define LCD_SETCGRAMADDR 0x40
#define LCD_SETDDRAMADDR 0x80

// flags for display entry mode
#define LCD_ENTRYRIGHT 0x00
#define LCD_ENTRYLEFT 0x02
#define LCD_ENTRYSHIFTINCREMENT 0x01
#define LCD_ENTRYSHIFTDECREMENT 0x00

// flags for display on/off control
#define LCD_DISPLAYON 0x04
#define LCD_DISPLAYOFF 0x00
#define LCD_CURSORON 0x02
#define LCD_CURSOROFF 0x00
#define LCD_BLINKON 0x01
#define LCD_BLINKOFF 0x00

// flags for display/cursor shift
#define LCD_DISPLAYMOVE 0x08
#define LCD_CURSORMOVE 0x00
#define LCD_MOVERIGHT 0x04
#define LCD_MOVELEFT 0x00

// flags for function set
#define LCD_8BITMODE 0x10
#define LCD_4BITMODE 0x00
#define LCD_2LINE 0x08
#define LCD_1LINE 0x00
#define LCD_5x10DOTS 0x04
#define LCD_5x8DOTS 0x00

// flags for backlight control
#define LCD_BACKLIGHT 0x08
#define LCD_NOBACKLIGHT 0x00
```



```
#define En B00000100 // Enable bit
#define Rw B00000010 // Read/Write bit
#define Rs B00000001 // Register select bit

class LiquidCrystal_I2C : public Print {
public:
  LiquidCrystal_I2C(uint8_t lcd_Addr,uint8_t lcd_cols,uint8_t
lcd_rows);
  void begin(uint8_t cols, uint8_t rows, uint8_t charsize =
LCD_5x8DOTS );
  void clear();
  void home();
  void noDisplay();
  void display();
  void noBlink();
  void blink();
  void noCursor();
  void cursor();
  void scrollDisplayLeft();
  void scrollDisplayRight();
  void printLeft();
  void printRight();
  void leftToRight();
  void rightToLeft();
  void shiftIncrement();
  void shiftDecrement();
  void noBacklight();
  void backlight();
  void autoscroll();
  void noAutoscroll();
  void createChar(uint8_t, uint8_t[]);
  void setCursor(uint8_t, uint8_t);
#if defined(ARDUINO) && ARDUINO >= 100
  virtual size_t write(uint8_t);
#else
  virtual void write(uint8_t);
#endif
  void command(uint8_t);
  void init();

  ///compatibility API function aliases
void blink_on(); // alias for blink()
void blink_off(); // alias for
noBlink()
void cursor_on(); // alias for
cursor()
void cursor_off(); // alias for
noCursor()
void setBacklight(uint8_t new_val); // alias for
backlight() and nobacklight()
void load_custom_character(uint8_t char_num, uint8_t *rows); //
alias for createChar()
void printstr(const char[]);
```



```
////Unsupported API functions (not implemented in this library)
uint8_t status();
void setContrast(uint8_t new_val);
uint8_t keypad();
void setDelay(int,int);
void on();
void off();
uint8_t init_bargraph(uint8_t graphtype);
void draw_horizontal_graph(uint8_t row, uint8_t column, uint8_t len,
uint8_t pixel_col_end);
void draw_vertical_graph(uint8_t row, uint8_t column, uint8_t len,
uint8_t pixel_col_end);

private:
void init_priv();
void send(uint8_t, uint8_t);
void write4bits(uint8_t);
void expanderWrite(uint8_t);
void pulseEnable(uint8_t);
uint8_t _Addr;
uint8_t _displayfunction;
uint8_t _displaycontrol;
uint8_t _displaymode;
uint8_t _numlines;
uint8_t _cols;
uint8_t _rows;
uint8_t _backlightval;
};

#endif
```




ΠΑΡΑΡΤΗΜΑ Η'

Τελικό Διάγραμμα Ηλεκτρονικού Κυκλώματος

[Βλέπε επόμενη σελίδα]





ΠΑΡΑΡΤΗΜΑ Θ'

Datasheets Ηλεκτρονικών Εξαρτημάτων (Τεχνικά Δελτία)

Στις επόμενες σελίδες βρίσκονται τα τεχνικά δελτία των ηλεκτρονικών εξαρτημάτων, όπως δίνονται από τους κατασκευαστές, για πληρότητα της παρούσας διπλωματικής εργασίας.

