

Διεθνές Πανεπιστήμιο της Ελλάδος

Τμήμα Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών

---

Διπλωματική Εργασία Πρόγραμμα Μεταπτυχιακών Σπουδών στη Ρομποτική

---

# Κατασκευή συστήματος πλήρωσης στερεών προϊόντων προς συσκευασία

Καρακύριος Νικόλαος  
Διεθνές Πανεπιστήμιο της Ελλάδος

Σέρρες, 30-3-2021

Εργασία που υποβλήθηκε στο  
Πρόγραμμα Μεταπτυχιακών σπουδών στην Ρομποτική,  
του Διεθνούς Πανεπιστημίου της Ελλάδος,  
για τη μερική εκπλήρωση υποχρεώσεων για το Δίπλωμα Ειδίκευσης στη  
Ρομποτική

Επιβλέπων Καθηγητής: Ιωάννης Καλόμοιρος

## Πρόλογος - Ευχαριστίες

Η παρούσα διπλωματική εργασία με θέμα “Κατασκευή συστήματος πλήρωσης στερεών προϊόντων προς συσκευασία” πραγματοποιήθηκε στο πλαίσιο των σπουδών μου στο Μεταπτυχιακό Πρόγραμμα Σπουδών στη ρομποτική του τμήματος πληροφορικής, υπολογιστών και τηλεπικοινωνιών του Διεθνούς Πανεπιστημίου της Ελλάδος.

Πρώτα από όλους θέλω να ευχαριστήσω τη σύζυγο μου, Κωνσταντίνα, που με την υπομονή, τη δύναμη αλλά κυρίως την αγάπη της, με βοήθησε να δώσω τον καλύτερο μου εαυτό στη μάθηση, την έρευνα και την πειραματική διαδικασία που χρειάστηκε αυτή η εργασία.

Θέλω επίσης να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Καλόμοιρο για τη στήριξη, την καθοδήγηση και την εμπιστοσύνη του σχετικά με αυτήν την εργασία που ήταν τόσο σημαντική για μένα.

Τέλος, θέλω να ευχαριστήσω την οικογένεια μου που με στήριξε με κάθε τρόπο για να μπορέσω και εγώ να κάνω αυτό που αγαπάω πράξη, εκπονώντας αυτήν την εργασία και περιγράφοντας ένα μαγικό ταξίδι, από μια αρχική ιδέα σε μια ολοκληρωμένη οντότητα, με διαδραστικότητα, σκοπό και μέλλον.

# Περίληψη

Η παρούσα διπλωματική εργασία έχει να κάνει με το πρόβλημα που αντιμετωπίζουν βιομηχανίες στη συσκευασία των προϊόντων τους. Τα κύρια προβλήματα που αντιμετωπίζουν έχουν να κάνουν με την αξιοπιστία των σταθμών παραγωγής, την δυνατότητα να προσαρμοστεί σε κάποια αλλαγή στο προϊόν αλλά και φυσικά την ταχύτητα παραγωγής. Μέχρι σήμερα, το μεγαλύτερο κομμάτι των γραμμών παραγωγής ελέγχεται από προγραμματιζόμενους λογικούς ελεγκτές (PLC) ή απλά ηλεκτρικά κυκλώματα. Η 4η βιομηχανική επανάσταση (Industry 4.0) άλλαξε τις απαιτήσεις στις σύγχρονες γραμμές παραγωγής με το βασικότερο από όλα να είναι η σύνδεση στο διαδίκτυο. Συνδέοντας τη γραμμή παραγωγής στο διαδίκτυο οι επιπρόσθετες δυνατότητες είναι πάρα πολλές αλλά και πολύ σημαντικές για τη βιομηχανία. Η συλλογή δεδομένων, η γρήγορη και έγκυρη ενημέρωση καθώς και η απομακρυσμένη λειτουργία, ήταν διαχρονικές ανάγκες των γραμμών παραγωγής, όμως ένας παράγοντας ήταν απαγορευτικός για την κάλυψη τους, το κόστος. Η εξέλιξη της τεχνολογίας, μειώνοντας το κόστος σε διάφορα εξαρτήματα αλλά και υπηρεσίες πλέον έδωσαν την δυνατότητα για εξέλιξη σε αυτόν τον τομέα.

Στοχεύοντας λοιπόν σε έναν από τους παράγοντες που κρατούσε πίσω όλα αυτά τα χρόνια τη βιομηχανία, στην παρούσα εργασία μελετήθηκε η λύση του προβλήματος πλήρωσης στερεών προϊόντων προς συσκευασία χρησιμοποιώντας ένα φθηνό μικροελεγκτή τύπου Arduino και έναν υπολογιστή τσέπης τύπου Raspberry Pi. Εκτός από το κόστος, πλεονέκτημα των μικροελεγκτών Arduino είναι πως υπάρχει πάρα πολύ μεγάλη κοινότητα που γράφει προγράμματα και τα ανεβάζει στο διαδίκτυο με στόχο να γίνουν πιο διαδεδομένοι και να κατακτήσουν μεγαλύτερο κομμάτι της αγοράς. Περνώντας στο Raspberry Pi, το λειτουργικό που υποστηρίζει είναι διάφορες εκδόσεις Linux και υποστηρίζει διάφορες γλώσσες προγραμματισμού, μια από τις δημοφιλέστερες που χρησιμοποιήθηκε επίσης για την ανάπτυξη του συστήματος αλληλεπίδρασης μηχανής - χρήστη (Human - machine interface HMI) είναι η “Python” με την χρήση της βιβλιοθήκης “tkinter”. Η κατασκευή των μηχανολογικών εξαρτημάτων έγινε χρησιμοποιώντας ένα σχεδιαστικό πρόγραμμα με τρισδιάστατη απεικόνιση και τη δυνατότητα εξαγωγής “αρχείων κοπής” για την παραγωγή των εξαρτημάτων. σε εργαλειομηχανές (laser, στράντζα) καθώς και σε τρισδιάστατους εκτυπωτές.

Δοκιμάζοντας την κατασκευή, τα πρώτα αποτελέσματα ήταν ικανοποιητικά. Η επικοινωνία λειτουργούσε άψογα, η ακρίβεια και η επαναληψιμότητα του συστήματος ήταν υπεράνω των αποδεκτών ορίων και η ταχύτητα του συστήματος υπερβαίνει τις απαιτήσεις. Ελάχιστο προβλήματα παρουσιάστηκαν λόγω του υλικού (πλαστικό) καθώς δεν είχε τις κατάλληλες μηχανικές ιδιότητες ώστε να αντέξει τα φορτία που δεχόταν. Αποτέλεσμα ήταν μερικές μικρές δυσλειτουργίες.

# Summary

This thesis deals with the filling problem of packaging lines. The main problems they are facing are reliability, adaptability and capacity. Till now, the majority of packaging lines are using PLCs and basic electronics. Industry 4.0 brought the capability to the packaging lines of internet connectivity. By this ability, numerous benefits have been developed that are really important for the industry. Data collection, fast and safe update, and remote control are requirements that packaging lines missed for a long time, but the main burden that no solution could be achieved was the cost. Technology improvement brought cost reduction and nowadays cheaper solutions may fit to our needs.

This thesis will deal with the use of arduino and raspberry pi for developing a filling station. Apart from the arduino price, arduino's users community is also a crucial factor since they contribute open source software. Raspberry Pi is a small size computer which works with linux OS and can be programmed using various programming languages. One of the most popular, "python", was used for development in this thesis. The human machine interface (HMI) was developed by using the tkinter library. Finally, Solidworks was used for mechanical 3D modeling and production of the real model and both CNC - Laser, 3D printer and manual machines.

Tests were successful, communication worked properly and the system's accuracy and repeatability was at a good level, additionally the system's speed was above the limits. A minor problem was faced due to the material of the guides (plastic), which had not the proper mechanical properties to handle the forces and sometimes (rarely) made the system unreliable.

## Περιεχόμενα:

Κεφάλαιο Πρώτο: Το πρόβλημα στις γραμμές συσκευασίας-παραγωγής	8
Ενότητα 1.1 Ιστορική αναδρομή	8
Ενότητα 1.2 Η διαδικασία της συσκευασίας	10
Ενότητα 1.3 Σημαντικές πληροφορίες για το προϊόν και το σύστημα συσκευασίας της παρούσας εργασίας	11
Κεφάλαιο δεύτερο: Συστήματα πλήρωσης στερεών προϊόντων	13
Ενότητα 2.1 Jorner Machinery - Γραμμές παραγωγής	13
Ενότητα 2.2 Hulmann - Γραμμές παραγωγής	18
Ενότητα 2.3 Ερασιτεχνικά συστήματα με χρήση μικροελεγκτών	21
Κεφάλαιο τρίτο: Μεθοδολογία επίλυσης προβλήματος	26
Ενότητα 3.1 Ανάλυση συστήματος	26
Ενότητα 3.2 Διαδικασία σχεδιασμού - παραγωγής	27
Ενότητα 3.3 Εμπορεύσιμα εξαρτήματα	33
Κεφάλαιο τέταρτο: Σχέδια / τρισδιάστατη απεικόνιση	41
Ενότητα 4.1 Μηχανολογικά σχέδια / 3D	42
Ενότητα 4.2 Ηλεκτρολογικά σχέδια	53
Κεφάλαιο πέμπτο: Διαγράμματα ροής δεδομένων	55
Ενότητα 5.1 Διαδρομή προϊόντος	55
Ενότητα 5.2 Μονάδα ελέγχου υπολογιστή - περιβάλλον χρήστη	56
Ενότητα 5.2.2 Αρχικοποίηση προγράμματος	58
Ενότητα 5.2.3 Συναρτήσεις	58
Ενότητα 5.2.4 Frames	60
Ενότητα 5.2.5 Entries, Buttons, Grid	60
Ενότητα 5.3 Μικροελεγκτής	62
Ενότητα 5.3.1 Μονάδα ελέγχου κινητήρα	62
Ενότητα 5.3.2 Αρχικοποίηση μικροελεγκτή arduino	65
Ενότητα 5.3.3 Σειριακή επικοινωνία υπολογιστή - μικροελεγκτή	67
Ενότητα 5.3.4 Εντολές μικροελεγκτή	69
Κεφάλαιο έβδομο: Δοκιμές	70
Κεφάλαιο όγδοο: Αξιολόγηση αποτελεσμάτων	77
Παράρτημα Α: Κώδικας Arduino	82

Παράρτημα Β: Κώδικας Python	85
Παράρτημα Γ: Πίνακας σημάτων	88
Παράρτημα Δ: Κατάλογος εικόνων	89
Παράρτημα Δ: Κατάλογος Σχημάτων	92

# Κεφάλαιο Πρώτο: Το πρόβλημα στις γραμμές συσκευασίας-παραγωγής

## Ενότητα 1.1 Ιστορική αναδρομή

Το πρόβλημα το οποίο στοχεύει να λύσει αυτή η εργασία είναι η πλήρωση στερεών προϊόντων προς συσκευασία χρησιμοποιώντας υλικά και τεχνικές καινοτόμα και κυρίως χαμηλού κόστους, συμβατά με καινοτόμες γραμμές παραγωγής. Οι καινοτόμες γραμμές παραγωγής (Industry 4.0) έχουν ως στόχο, ο χειρισμός και η εξαγωγή δεδομένων από την μηχανή να γίνεται μέσω διαδικτύου. Τα οφέλη από αυτό το δεδομένο για μια βιομηχανία είναι πολλά. Η ανάγκη των εργοστασίων να μεταβούν στην τέταρτη βιομηχανική επανάσταση, όπως λέγεται ( Industry 4.0), ήρθε κυρίως από τον μεγάλο ανταγωνισμό που έφερε το άνοιγμα του εμπορίου. Την δεκαετία του 80, υπήρχαν πάρα πολλές βιοτεχνίες στην Ελλάδα και αρκετά εργοστάσια που κατα κύριο λόγο παρήγαγαν, αν όχι ένα προϊόν, προϊόντα από μια και μόνο πρώτη ύλη (π.χ. πλαστικό, σίδηρο, δέρμα, γάλα, υφάσματα κ.α.). Ο λόγος που μπορούσαν να επιβιώσουν αυτές οι βιοτεχνίες στην αγορά ήταν πως το αγοραστικό κοινό δεν είχε την επιλογή να παραγγείλει κάτι από το εξωτερικό, δεν είχε καν την επιλογή να δει που υπάρχει στον κόσμο εργοστάσιο που να φτιάχνει το προϊόν το οποίο έψαχνε. Αυτήν την δυνατότητα την έδωσε στον κόσμο το διαδίκτυο. Σήμερα, ότι και να χρειάζεται το καταναλωτικό κοινό, έχουν αναπτυχθεί εργαλεία που όχι μόνο διαφημίζουν τα εργοστάσια και τους εμπόρους παγκοσμίως, αλλά επίσης συγκρίνουν τιμές, δίνουν στο καταναλωτικό κοινό δυνατότητα να σχολιάσει την αγοραστική του εμπειρία κ.α. Με αυτά τα δεδομένα, δεν είναι τόσο απλή η παραγωγή/συσκευασία καθώς ο ανταγωνισμός πλέον είναι παγκόσμιος και οι γραμμές παραγωγής έχουν μελετηθεί αρκετά από ανθρώπους, έτσι ώστε να μην γίνεται καμία επιλογή υλικού, μηχανής, διαδικασίας και εγκατάστασης στην τύχη. Η μελέτη και βελτιστοποίηση γραμμών παραγωγής/συσκευασίας είναι ένα μεγάλο επιστημονικό πεδίο που δεν θα αναλυθεί στην παρούσα εργασία. Ενδεικτικά θα αναφερθούν μόνο κάποιες πηγές για την μελέτη και βελτιστοποίησης γραμμών παραγωγής που ξεκίνησαν στην Ιαπωνία και την Toyota που έφερε την ονομασία “Λιτή παραγωγή” (Lean manufacturing) από το 1930 με το “Toyota Production System, TPS”. Στην συνέχεια η Motorola ανέπτυξε την μέθοδο 6σ (six sigma) η οποία ακόμα και σήμερα παίζει καθοριστικό ρόλο στην μελέτη μονάδων παραγωγής. Ακολούθησαν πολλές ακόμα μελέτες πάνω στο θέμα που είναι από τα πιο σημαντικά για την βιομηχανία.

Καθοριστικό ρόλο στην βιωσιμότητα των εργοστασίων παίζει η κερδοφορία τους, για αυτό και καθίσταται επιτακτική ανάγκη μιας γραμμής παραγωγής, να μπορεί να προσαρμόζεται στις ανάγκες των πωλήσεων. Η προσαρμογή αυτή μπορεί να έχει να κάνει με την ταχύτητα παραγωγής, την δυνατότητα παραγωγής διαφόρων κωδικών προϊόντος καθώς και την ανίχνευση σφαλμάτων / δυσλειτουργιών στην γραμμή. Στις παλιές γραμμές παραγωγής, υπήρχε ένας κινητήρας, ο οποίος με αυξομειώσεις στροφών και μετάδοση της κίνησης, κινούσε όλους τους επιμέρους σταθμούς. Αυτό καθιστά την ρύθμιση της ταχύτητας εύκολη, όμως και καθόλου αξιόπιστη καθώς εμπλέκονται πολλά μηχανικά μέρη που με την φθορά αποκτούν ανοχές και προκαλούν αστοχίες στην γραμμή. Οι νέες γραμμές έχουν τοποθετήσει έναν κινητήρα σε κάθε σταθμό, συνήθως με κωδικοποιητή πάνω στον άξονα του, ώστε να γίνεται άμεσα η μετάδοση κίνησης σε κάθε σταθμό, ο συγχρονισμός των σταθμών παραγωγής γίνεται από τον προγραμματιζόμενο λογικό ελεγκτή (PLC) και είναι ακόμα πιο εύκολη η αλλαγή στις ρυθμίσεις την μηχανής (π.χ. προπορεία σταθμών, διάρκεια ενεργοποιητών κ.α.). Η δυνατότητα παραγωγής διαφόρων κωδικών προϊόντος έχει να κάνει με τα φυσικά όρια των υλικών κατασκευής της γραμμής παραγωγής/συσκευασίας (μηχανικές αντοχές, προδιαγραφές εξωτερικών συστημάτων κ.α.) και με τον τρόπο προγραμματισμού της γραμμής συσκευασίας/παραγωγής. Είναι πολύ συχνό φαινόμενο σε γραμμές παραγωγής/συσκευασίας να μην είναι δυνατή η ρύθμιση διαφόρων μεταβλητών χωρίς την αλλαγή σημαντικού όγκου κώδικα. Τέλος, η ανίχνευση σφαλμάτων (troubleshooting) είναι ένα θέμα που ταλαιπωρεί πάρα πολύ τις κατασκευαστικές εταιρείες μηχανών και ξοδεύουν υπέρογκα ποσά στέλνοντας τεχνικούς για παρακολούθηση γραμμών. παραγωγής/συσκευασίας, μερικές φορές χωρίς να έχουν αποτέλεσμα και άλλες φορές χωρίς να υπάρχει πρόβλημα στην γραμμή παραγωγής/συσκευασίας.



## Ενότητα 1.2 Η διαδικασία της συσκευασίας

Για να μελετηθεί λοιπόν ένας ρομποτικός σταθμός παραγωγής/συσκευασίας, πρέπει πρώτα να αναλυθεί τι είναι η γραμμή παραγωγής/συσκευασίας. Η γραμμή παραγωγής/συσκευασίας λοιπόν είναι ένα σύνολο από μηχανήματα που ως στόχο έχουν την μεταποίηση πρώτων υλών για την παραγωγή ενός τελικού προϊόντος. Η γραμμή παραγωγής/συσκευασίας λοιπόν αποτελείται από επιμέρους υποσυστήματα, τα οποία το καθένα έχει και έναν ξεχωριστό σκοπό. Ξεκινώντας από έναν από τους πιο σημαντικούς σταθμούς μιας γραμμής παραγωγής/συσκευασίας είναι ο σταθμός μεταφοράς. Ο σταθμός μεταφοράς έχει ως στόχο να μεταφέρει το προϊόν από την είσοδο του (είσοδο της συσκευασίας) μέχρι και την έξοδο του τελικού προϊόντος. Ο εν λόγω σταθμός είναι πολύ σημαντικός καθώς η ακρίβεια του επηρεάζει όλους τους υπόλοιπους σταθμούς παραγωγής.

Ένας επίσης σημαντικός σταθμός είναι αυτός που τροφοδοτεί την γραμμή παραγωγής/συσκευασίας με τα υλικά συσκευασίας. Μερικές γραμμές παραγωγής/συσκευασίας παίρνουν έτοιμα υλικά συσκευασίας (π.χ. κύπελλα), και κάποιες άλλες φτιάχνουν την συσκευασία σε εσωτερικό σταθμό διαμόρφωσης (από ρολό πλαστικό που ξετυλίγεται, διαμορφώνονται θήκες για να γεμίσουν, να σφραγιστούν με το προϊόν και μετά να κοπούν).

Ένας ακόμα σημαντικός σταθμός είναι αυτός της υγιεινής, στον σταθμό αυτόν γίνεται ο καθαρισμός / αποστείρωση της συσκευασίας και η απομάκρυνση ξένων σωμάτων. Συνήθως τέτοιοι σταθμοί υπάρχουν σε βιομηχανίες τροφίμων και φαρμάκων. Μετά τους σταθμούς υγιεινής, κατά κανόνα βρίσκονται οι σταθμοί πλήρωσης για να είναι όσο το δυνατό μικρότερος ο χρόνος που κινείται η συσκευασία αποστειρωμένη και χωρίς να έχει σφραγιστεί στην γραμμή παραγωγής/συσκευασίας. Στον σταθμό πλήρωσης το προϊόν μπαίνει στην συσκευασία και θα συνεχίσει για να πάει προς σφράγιση. Ανάλογα με τα χαρακτηριστικά του υλικού προκύπτουν και οι απαιτήσεις από τον σταθμό πλήρωσης. Τα σημαντικά χαρακτηριστικά ενός υλικού που πρέπει να συσκευαστεί είναι η κατάσταση του (υγρό / στερεό / αέριο), οι ιδιότητες του (μηχανικές αντοχές / ψαθυρότητα / βάρος / ιξώδες κ.α.) και πολλά άλλα, διαφορετικά ανάλογα την περίπτωση.

Ένα σημαντικό θέμα των γραμμών παραγωγής που γίνεται όλο ένα και πιο περίπλοκο είναι η επικοινωνία. Η επικοινωνία στις γραμμές παραγωγής γίνεται είτε με κωδικοποίηση κάποιων μηνυμάτων με ψηφιακά σήματα είτε με επικοινωνίες τύπου ethernet όταν υπάρχει μεταφορά δεδομένων. Ένας σταθμός παραγωγής για να ενσωματωθεί σε μια γραμμή παραγωγής πρέπει να είναι έτοιμος να της στείλει τα δεδομένα που θα χρειαστεί. Εφόσον χρησιμοποιείται το Raspberry Pi για τον έλεγχο του Arduino δεν υπάρχει πρόβλημα στην αποστολή δεδομένων στο διαδίκτυο.

## Ενότητα 1.3 Σημαντικές πληροφορίες για το προϊόν και το σύστημα συσκευασίας της παρούσας εργασίας

Αναλύοντας την περίπτωση της παρούσας εργασίας, στα στερεά προϊόντα προς συσκευασία, σημαντικό ρόλο παίζει το βάρος, ο όγκος, το σχήμα, οι μηχανικές αντοχές και η ψαθυρότητα. Επίσης, σημαντικό ρόλο παίζουν και οι ανοχές στις διαστάσεις των προϊόντων καθώς η οδήγηση τους από το δοχείο αναμονής προς την συσκευασία του γίνεται εξαναγκασμένα, με διάφορες μεθόδους (βαρύτητα, πεπιεσμένο αέρα, ηλεκτρομαγνητισμό, κίνηση μηχανισμών κ.α.) και είναι συχνό φαινόμενο οι εμπλοκές. Εκτός από εμπλοκές, στην εξαναγκασμένη μεταφορά συχνά εμφανίζεται και θραύση, όταν έχουμε ψαθυρά προϊόντα, ή παραμόρφωση όταν έχουμε μη ψαθυρά. Η πρόκληση σε έναν νέο σταθμό πλήρωσης στερεών προϊόντων είναι να μπορεί να μεταφέρει τα προϊόντα στην συσκευασία τους γρήγορα, με ασφάλεια και να μπορεί να εντοπίσει την εμπλοκή πριν αυτή να εμφανιστεί στην γραμμή. Αυτό μπορεί να γίνει ελέγχοντας τον όγκο των προϊόντων, το σχήμα τους και την κατάσταση στον οδηγό στα σημεία που είναι επίφοβη η εμπλοκή των προϊόντων. Τέτοια σημεία είναι γωνίες, ενώσεις εξαρτημάτων και στενώσεις.

Η αποθήκευση των προϊόντων θα γίνει σε ένα μικρό τετράγωνο δοχείο που στο κάτω μέρος του θα έχει μια τρύπα, στο σχήμα του προϊόντος, μεγαλύτερη κατά 1 χιλιοστό σε κάθε διάσταση, έτσι ώστε να έχει ελεύθερη συναρμογή με το προϊόν, όμως να μην χωράνε να περάσουν δύο από εκεί. Η δουλειά του οδηγού, που βρίσκεται κάτω από το δοχείο, είναι να κρατάει στοιχισμένα τα προϊόντα, ώστε να περιμένουν το επόμενο στάδιο. Στο επόμενο στάδιο, υπάρχει ένας κύλινδρος που περιστρέφεται και στον άξονα του εκτός από τον κινητήρα έχει και έναν κωδικοποιητή έτσι ώστε να ρυθμίζεται η θέση του. Το ιδιαίτερο με αυτόν τον κύλινδρο είναι πως από την περίμετρο του έχει αφαιρεθεί το αρνητικό του προϊόντος προς συσκευασία, έτσι, όταν κάτω από τον οδηγό έρχεται κάποιο “αυλάκι”, αυτό γεμίζει με το υλικό προς συσκευασία και το παρασύρει μαζί του. Ο οδηγός όμως συνεχίζει κατα την περίμετρο του κυλίνδρου, για να κρατάει το προϊόν μέσα στο αυλάκι, μέχρι το σημείο που από κάτω θα υπάρχει η συσκευασία του. Τα σημεία που πρέπει να μελετηθούν πολύ σε συστήματα πλήρωσης όπως αυτό είναι οι γεωμετρίες του οδηγού και του κινούμενου κυλίνδρου, ο τρόπος ελέγχου του κινητήρα και του κωδικοποιητή και η γεωμετρία της κατασκευής έτσι ώστε να εξασφαλίζεται η σταθερότητα του συστήματος. Τέτοιου τύπου συστήματα πλήρωσης μπορούν να χρησιμοποιηθούν σε οποιαδήποτε γραμμή συσκευάζονται υλικά εύθραυστα, ψαθυρά ή στα οποία υπάρχουν αυστηρές προδιαγραφές για τον αριθμό των προϊόντων μέσα σε κάθε συσκευασία. Πολύ μεγάλη εφαρμογή τέτοιων συστημάτων υπάρχει στις φαρμακοβιομηχανίες όπου και από εκεί βρέθηκαν και αρκετά παραδείγματα προς ανάλυση.

Η παρούσα εργασία επίσης, θα συνεισφέρει στην παρακαταθήκη μιας ακόμα μελέτης συστήματος το οποίο χρησιμοποιεί μικροελεγκτές τύπου Arduino / Raspberry Pi , με σκοπό να γίνουν ακόμα πιο διαδεδομένα στην αγορά και το αντικείμενο της ρομποτικής στην Ελλάδα να μπορέσει να μπει στην Ελληνική βιομηχανία καθώς υπάρχουν πάρα πολλές ανάγκες εκσυγχρονισμού γραμμών παραγωγής/συσκευασίας για τις οποίες το μεγάλο πρόβλημα είναι το κόστος. Η μελέτη, κατασκευή και δοκιμή ρομποτικών συστημάτων από Ελληνικά πανεπιστήμια είναι η απόδειξη της πραγματικά σημαντικής, σκληρής και δύσκολης δουλειάς που γίνεται μέχρι σήμερα σε πολλά ιδρύματα στην Ελλάδα και ,γιατί όχι, να δώσουν το ερέθισμα για νέες μελέτες, επενδύσεις και καινοτομίες.

## Κεφάλαιο δεύτερο: Συστήματα πλήρωσης στερεών προϊόντων

Σε αυτό το κεφάλαιο θα παρουσιαστούν σύγχρονα συστήματα πλήρωσης και θα αναλυθεί ο τρόπος λειτουργίας τους. Εξειδικευμένα ζητήματα για προϊόντα δεν γίνονται γνωστά από τις εταιρείες οπότε δεν υπάρχει πρόσβαση σε λεπτομέρειες. Θα γίνει περιγραφή του τρόπου λειτουργίας τους, ανάλυση των επιμέρους συστημάτων τους και θα μια αναφορά στα πλεονεκτήματα και μειονεκτήματα της κάθε λύσης που βρέθηκε στο σύστημα πλήρωσης προϊόντων.

### Ενότητα 2.1 Jorner Machinery - Γραμμές παραγωγής

Η Jorner Machinery Co., Ltd. ιδρύθηκε το 1989 στην Κίνα και δραστηριοποιείται στον τομέα των φαρμάκων. Ακολουθούν κάποιες λύσεις που έχει φτιάξει η Jorner για το πρόβλημα την πλήρωσης στερεών προϊόντων προς συσκευασία.

1) Το σύστημα πλήρωσης της εικόνας 2.1 απαρτίζεται από ένα ανοξείδωτο δοχείο προϊόντος, έναν δονητικό διάδρομο, μια βούρτσα, και την ταινία μεταφοράς της συσκευασίας.



Εικόνα 2.1: Chute alu-alu Blister Feeder, πηγή: <https://jornen.com/>  
Βλέπε αναφορά [7]

Στο σύστημα πλήρωσης της εικόνας 2.1, ο χειριστής βάζει το προϊόν στο ανοξείδωτο δοχείο και αυτό με το δονητικό ταμί στον πάτο του, το σπρώχνει προς τον οδηγό. Φυσικά, θα πρέπει να έχει κάποιον αισθητήρα για τον έλεγχο στάθμης, έτσι ώστε να καταλαβαίνει πότε έχει γεμίσει ο οδηγός, και να σταματάει ο δονητικός κινητήρας, δυστυχώς όμως είναι μια πληροφορία που δεν την δίνει ο κατασκευαστής. Στην συνέχεια, τα προϊόντα στοιβάζονται στην αρχή του οδηγού και μια περιστροφική βούρτσα τα ανακατεύει με σκοπό να περάσουν αυτά μέσα από τα αυλάκια του οδηγού. Μέσα στα αυλάκια, τα προϊόντα στοιχίζονται και το ένα σπρώχνει το άλλο μέχρι το τελευταίο που πατάει μέσα στην συσκευασία. Η συσκευασία έχει τέτοιο σχήμα που χωράει ένα προϊόν σε κάθε θήκη, έτσι είναι εφαρμόσιμη αυτή η λύση πλήρωσης προϊόντων. Καθώς προχωράει η ταινία μεταφοράς, φέρνει κάτω από τον οδηγό νέα άδεια κελιά και αυτά γεμίζουν με προϊόν λόγω της βαρύτητας. Έτσι, έχουν πετύχει μια φθηνή λύση για το σύστημα πλήρωσης χωρίς όμως να έχουν την δυνατότητα να αλλάξουν τον αριθμό των προϊόντων που θα μπει σε κάθε συσκευασία.

2) Το σύστημα της εικόνας 2.2 απαρτίζεται πάλι από ένα ανοξείδωτο δοχείο προϊόντος, ένα δονητικό ταμί μεταφοράς υλικού, έναν σωλήνα μεταφοράς υλικού, έναν οδηγό σε σχήμα ταπιού ο οποίος έχει και λειτουργίες με πεπιεσμένο αέρα και την ταινία μεταφοράς



Εικόνα 2.2: Tube blister feeder , πηγή: <https://jornen.com/>  
Βλέπε αναφορά [7]



Εικόνα 2.3: Tube blister feeder , πηγή: <https://jornen.com/>  
Βλέπε αναφορά [7]

Ο σταθμός πλήρωσης της εικόνας 2.3, όπως και ο προηγούμενος, έχουν ένα ανοξείδωτο δοχείο προϊόντος, ένα δονητικό ταψί μεταφοράς στον πάτο του και μετά η μεταφορά γίνεται με εύκαμπτο σωλήνα καθώς ο οδηγός κινείται. Ο οδηγός κινείται γιατί γεμίζει “παρτίδες” από την ταινία μεταφοράς και όχι κάθε σειρά. Όπως φαίνεται στην εικόνα, σε κάθε κύκλο λειτουργίας του συστήματος πλήρωσης, ο οδηγός αφήνει 20 σειρές προϊόντος πάνω στην ταινία μεταφοράς. Για να εξασφαλίσουν πως σε κάθε κύκλο όλες οι οπές θα είναι γεμάτες με προϊόν και αυτό θα έχει την σωστή θέση και προσανατολισμό για να μπει στην ταινία μεταφοράς γεμίζουν έναν αρκετά μεγάλο όγκο με προϊόν πάνω από τον οδηγό και αυτό το ανακατεύουν με πεπιεσμένο , αποστειρωμένο αέρα έτσι ώστε να υπάρχουν λιγότερες πιθανότητες να μην μείνει κανένα αυλάκι του οδηγού χωρίς προϊόν. Αυτή η φιλοσοφία πλήρωσης προϊόντων απαιτεί πάρα πολλές δοκιμές για την ροή αέρα και την γεωμετρία του οδηγού έτσι ώστε να έχουν την μεγαλύτερη δυνατότητα παραγωγής με τα λιγότερα σφάλματα. Επίσης, σε αυτό το σύστημα πλήρωσης δεν υπάρχει τρόπος να βάλεις δύο προϊόντα σε μια συσκευασία.

3) Το σύστημα της εικόνας 2.4 / 2,5 απαρτίζεται από ένα ανοξείδωτο δοχείο προϊόντος, ένα δονητικό ταγί μεταφοράς υλικού, ένα δεύτερο δονητικό ταγί μεταφοράς υλικού με ειδικές γεωμετρίες και συχνότητες δόνησης ώστε να δίνει μια κυκλική φορά στα προϊόντα, έναν οδηγό, έναν κύλινδρο με το αρνητικό του προϊόντος να λείπει από την περίμετρο του σε συγκεκριμένες θέσεις και την ταινία μεταφοράς



Εικόνα 2.4: Blister machine DPH260 , πηγή: <https://jornen.com/>  
Βλέπε αναφορά [7]



Εικόνα 2.5: Blister machine DPH260 , πηγή: <https://jornen.com/>  
Βλέπε αναφορά [7]

Το ιδιαίτερο στον σταθμό πλήρωσης της εικόνας 2.5 είναι πως έχει την δυνατότητα να ρυθμίσει την ποσότητα προϊόντων σε κάθε συσκευασία. Στην εικόνα βέβαια φαίνεται μια συσκευασία που μπορεί να υποδεχτεί ένα μόνο προϊόν, όμως αν το αυλάκι στην συσκευασία χωρούσε περισσότερα, αυτό το σύστημα πλήρωσης θα μπορούσε να την τροφοδοτήσει με αυτά. Η γενική φιλοσοφία είναι παρόμοια με τις προηγούμενες, μπαίνει το προϊόν στο ανοξείδωτο δοχείο, ένα δονητικό ταψί που μεταφέρει το προϊόν στο δεύτερο δονητικό ταψί το οποίο κάνει το προϊόν να περιστρέφεται, με τέτοιο τρόπο που γεμίζει τα αυλάκια του οδηγού. Στην συνέχεια ο οδηγός είναι πάνω από τον περιστρεφόμενο κύλινδρο και όταν αυτός περιστρέφεται, “γεμίζει” τα αυλάκια του με το προϊόν που πέφτει μέσα σε αυτά λόγω της βαρύτητας. Στην συνέχεια ακολουθεί ακόμα ένας οδηγός που κρατάει το προϊόν μέσα στον κύλινδρο μέχρι να φτάσει στο σημείο που έχει συσκευασία από κάτω, όπου και πέφτει μέσα σε αυτήν.



## Ενότητα 2.2 Hulmann - Γραμμές παραγωγής

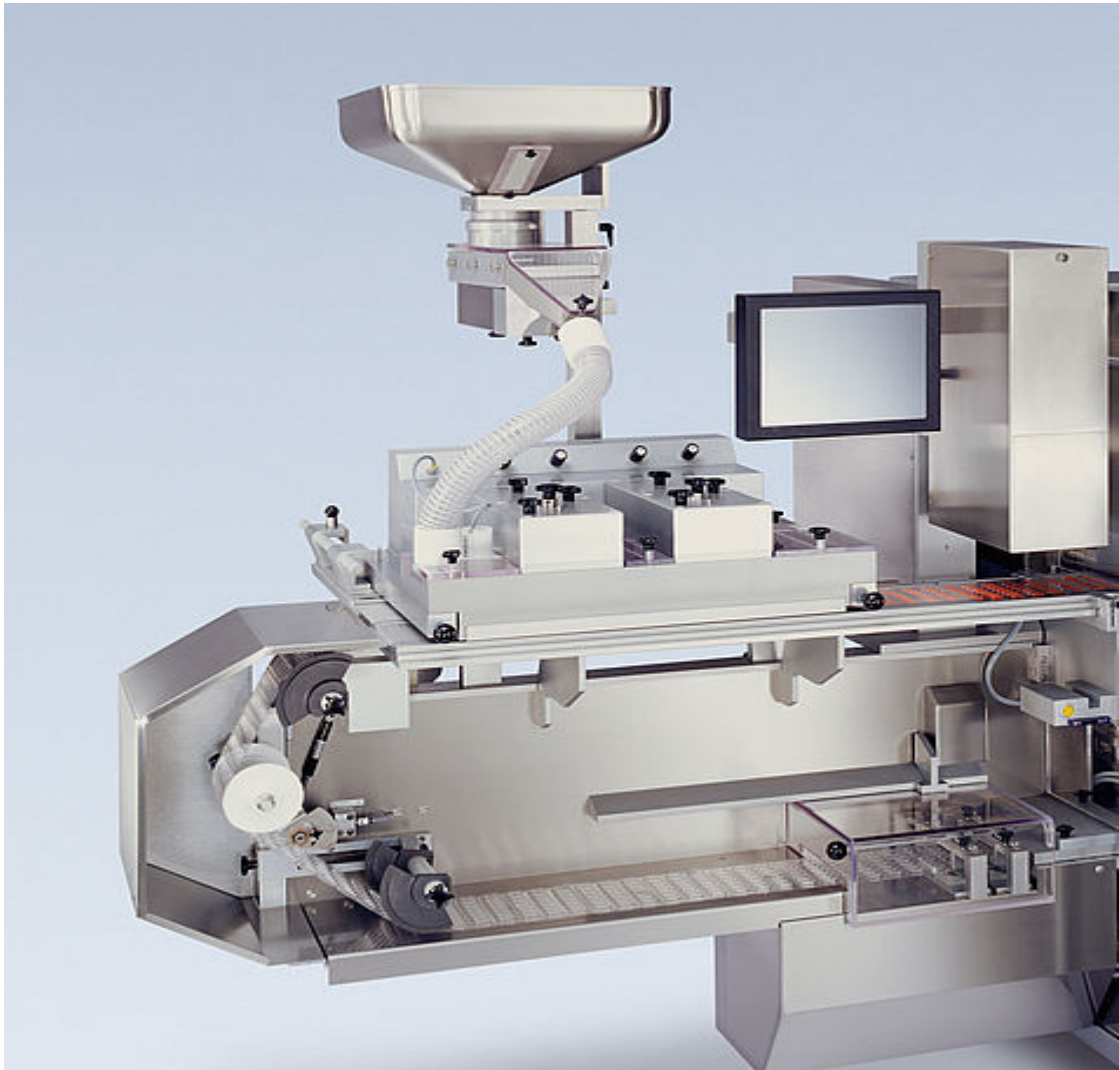
Η εταιρία Hulmann ιδρύθηκε το 1948 από τον Josef Uhlmann και παράγαν μηχανολογικά καλούπια ακριβείας. Με το πέρας των χρόνων, δραστηριοποιήθηκε στις γραμμές συσκευασίας και παραγωγής και σήμερα κατέχει σημαντικό μερίδιο της αγοράς παγκοσμίως.

Μια από τις γραμμές συσκευασίας της εταιρίας Uhlmann, η Blister machine B200, χρησιμοποιεί τον σταθμό πλήρωσης προϊόντων της εικόνας 2.6 ο οποίος φαίνεται να φέρει βούρτσες, σωλήνες αναρρόφησης καθώς και σταθμό απομάκρυνσης των προϊόντων που περισσεύουν πάνω από το φιλμ. Είναι ένας σταθμός παραγωγής που μπορεί να καλύψει αρκετά υψηλές ταχύτητες παραγωγής όμως σε ψαθυρά προϊόντα μπορεί να εμφανιστεί θραύση άρα και να υπάρχουν αρκετές αστοχίες στο υλικό.



Εικόνα 2.6: Blister machine B200 , πηγή:<https://www.hulmann.de/>:  
Βλέπε αναφορά [8]

Μια ακόμα από τις γραμμές συσκευασίας της εταιρίας Uhlmann, η Blister machine B1240, χρησιμοποιεί τον σταθμό πλήρωσης προϊόντων της εικόνας 2.7 ο οποίος φαίνεται να φέρει βούρτσες, σωλήνες αναρρόφησης καθώς και σταθμό απομάκρυνσης των προϊόντων που περισσεύουν πάνω από το φιλμ. Φαίνεται αρκετά όμοια με την προηγούμενη καθώς ανήκουν στην ίδια κατηγορία απλώς διαφέρει η δυνατότητα παραγωγής του κάθε σταθμού.



Εικόνα 2.7: Blister machine B1240 , πηγή: <https://www.uhlmann.de/>:  
Βλέπε αναφορά [8]

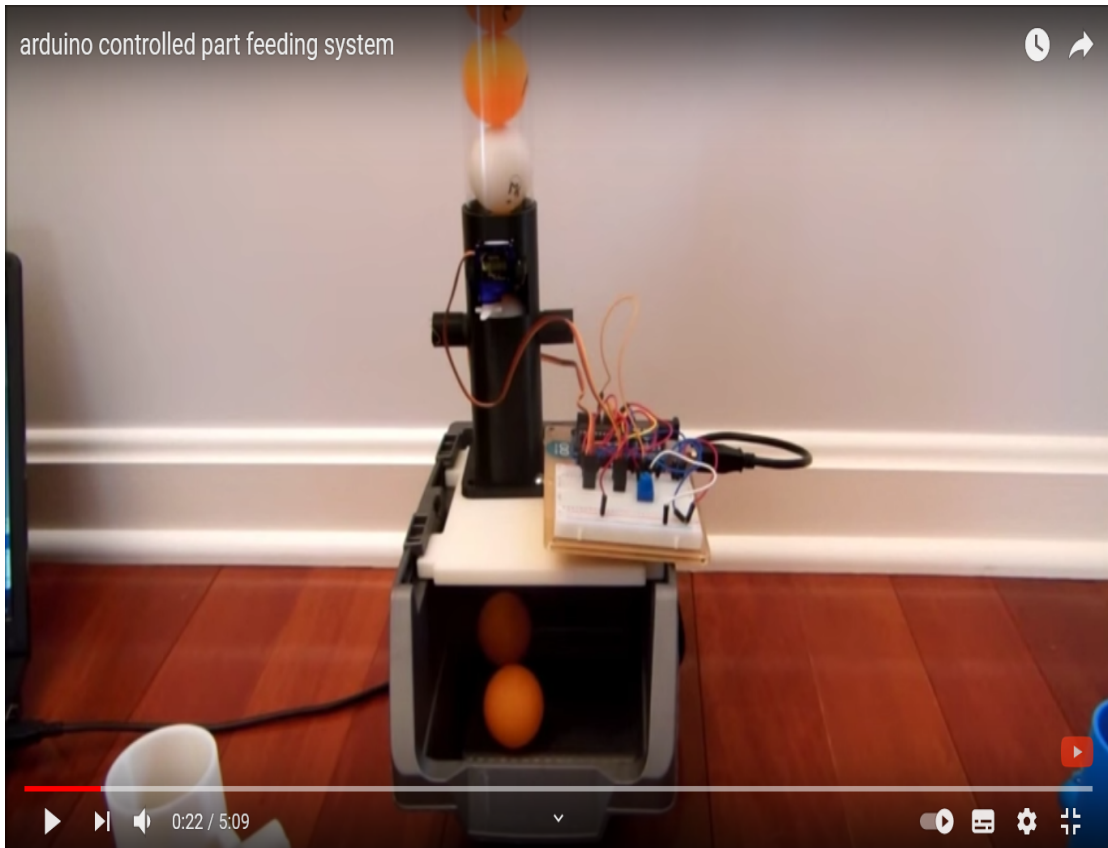
Η Uhlmann δίνει ακόμα την δυνατότητα στους πελάτες της να διαμορφώσουν αυτοί την γραμμή παραγωγής με διάφορους σταθμούς (π.χ. από διαφορετικές οικογένειες προϊόντων της εταιρίας). Έτσι, ονομάζοντας το σύστημα πλήρωσης “δονητικό τροφοδότη” (vibration feeder) παρουσιάζει τον σταθμό πλήρωσης της εικόνας 2.9.



Εικόνα 2.8: Vibration feeder , πηγή:<https://www.hulmann.de/>:  
Βλέπε αναφορά [8]

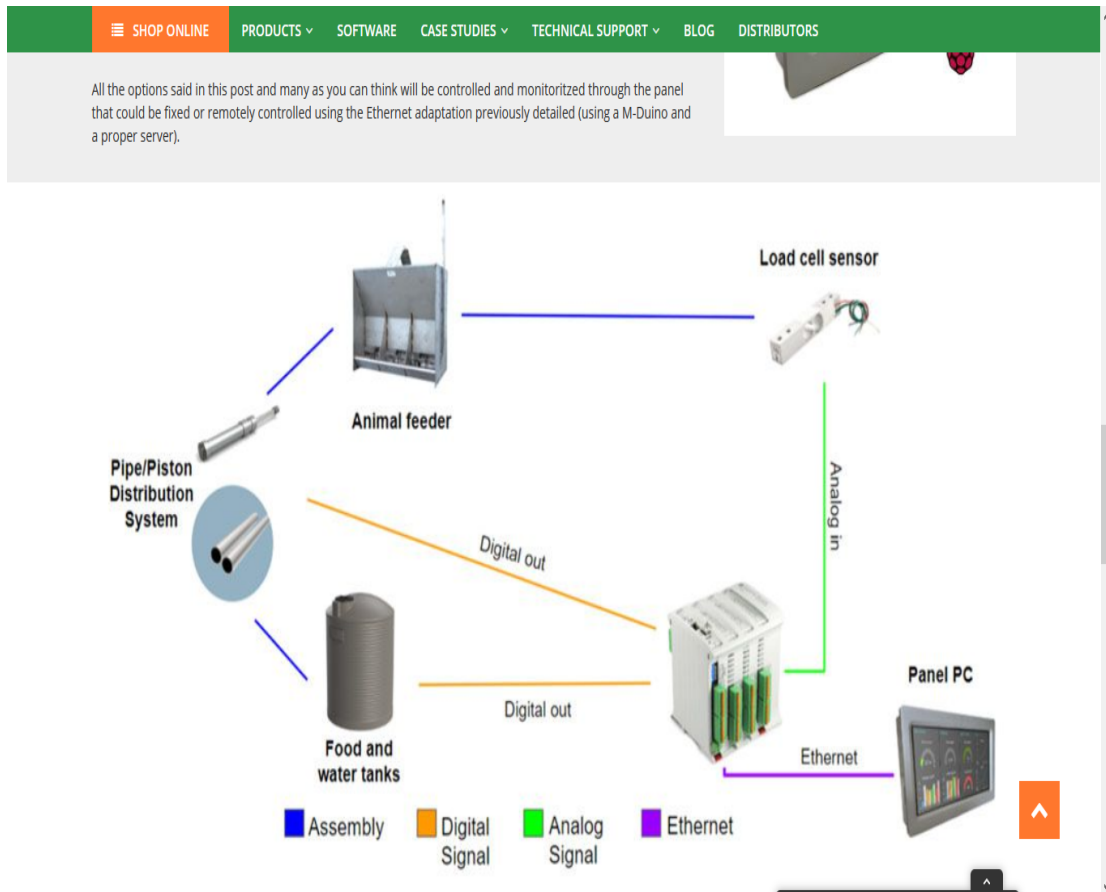
## Ενότητα 2.3 Ερασιτεχνικά συστήματα με χρήση μικροελεγκτών

Με τα συστήματα τροφοδότησης όμως δεν ασχολούνται μόνο μεγάλες εταιρίες για τις βιομηχανίες, παρακάτω θα παρουσιαστεί η δουλειά ενός ερασιτέχνη που ασχολείται με την ανάπτυξη συστημάτων με την χρήση Arduino. Έφτιαξε λοιπόν ένα σύστημα, χρησιμοποιώντας και αυτός εξαρτήματα φτιαγμένα με την χρήση τρισδιάστατου εκτυπωτή, το οποίο θα τροφοδοτεί με μπαλάκια του πινκ - πονκ με συγκεκριμένη χρονική καθυστέρηση. Το σύστημα χρησιμοποιεί δύο σερβο κινητήρες για να ελέγξει την ποσότητα πλήρωσης και μια φωτοευαίσθητη αντίσταση σε συνδυασμό με μια λάμπα. Στις πηγές θα παρατεθεί η αναφορά για το βίντεο με την παρουσίαση του προϊόντος.



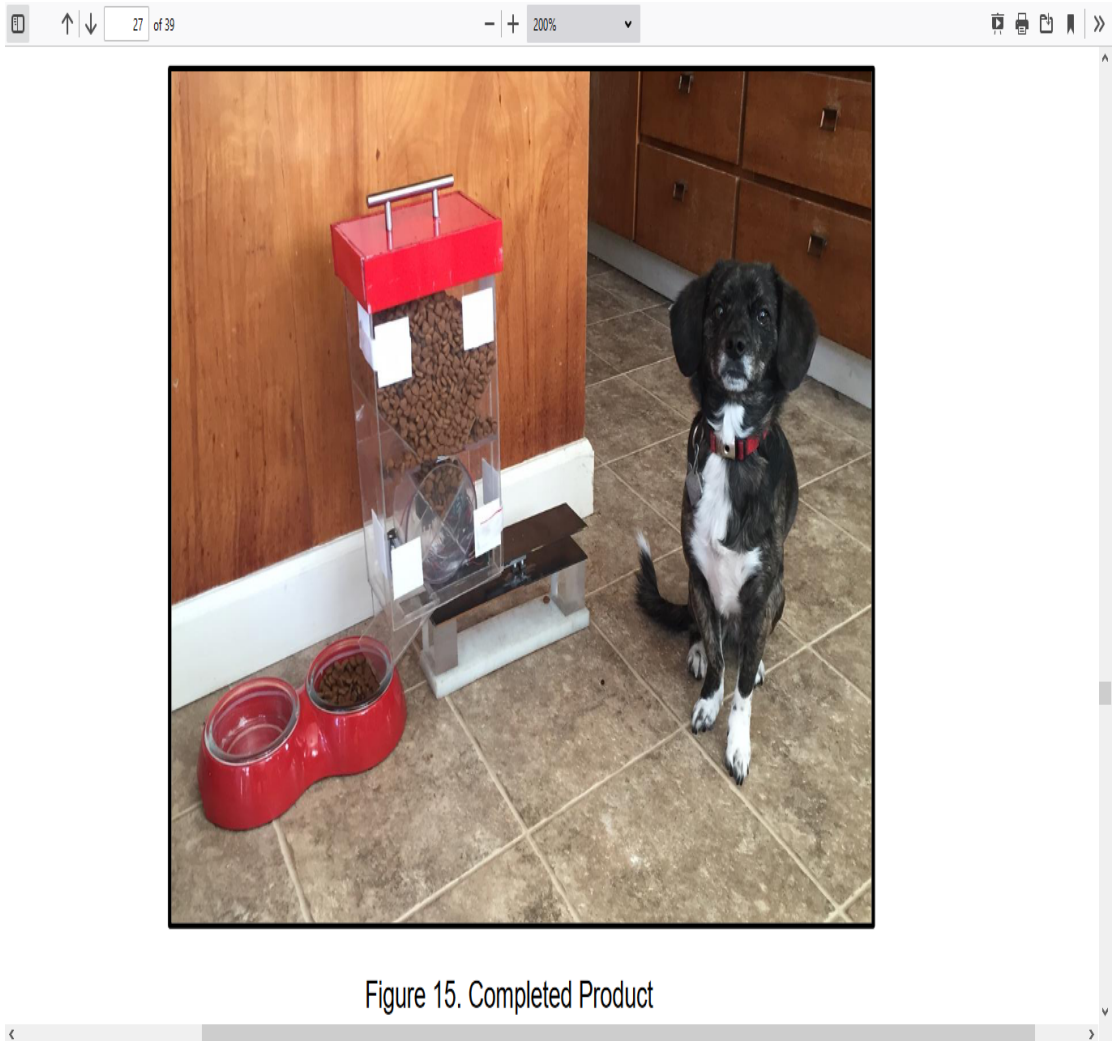
Εικόνα 2.9: Arduino controlled part feeding system ,  
πηγή:<https://www.youtube.com/watch?v=L47OBXGYRtE>  
Βλέπε αναφορά [9]

Η εταιρεία Industrial Shields παρέχει εξατομικευμένες λύσεις σε διάφορα προβλήματα με την χρήση open source συστημάτων. Στην εικόνα 2.10 φαίνεται μια λύση πλήρωσης τροφής και νερού σε ζώα με την χρήση δυναμοκυψέλης, απομακρυσμένης σύνδεσης κ.α.



Εικόνα 2.10: Λύση από την Industrial shields στο πρόβλημα τροφής - νερού σε μονάδα ζώων, πηγή:  
<https://www.industrialshields.com/case-study-animal-feeding-automation-industrial-plc-based-arduino>  
Βλέπε αναφορά [10]

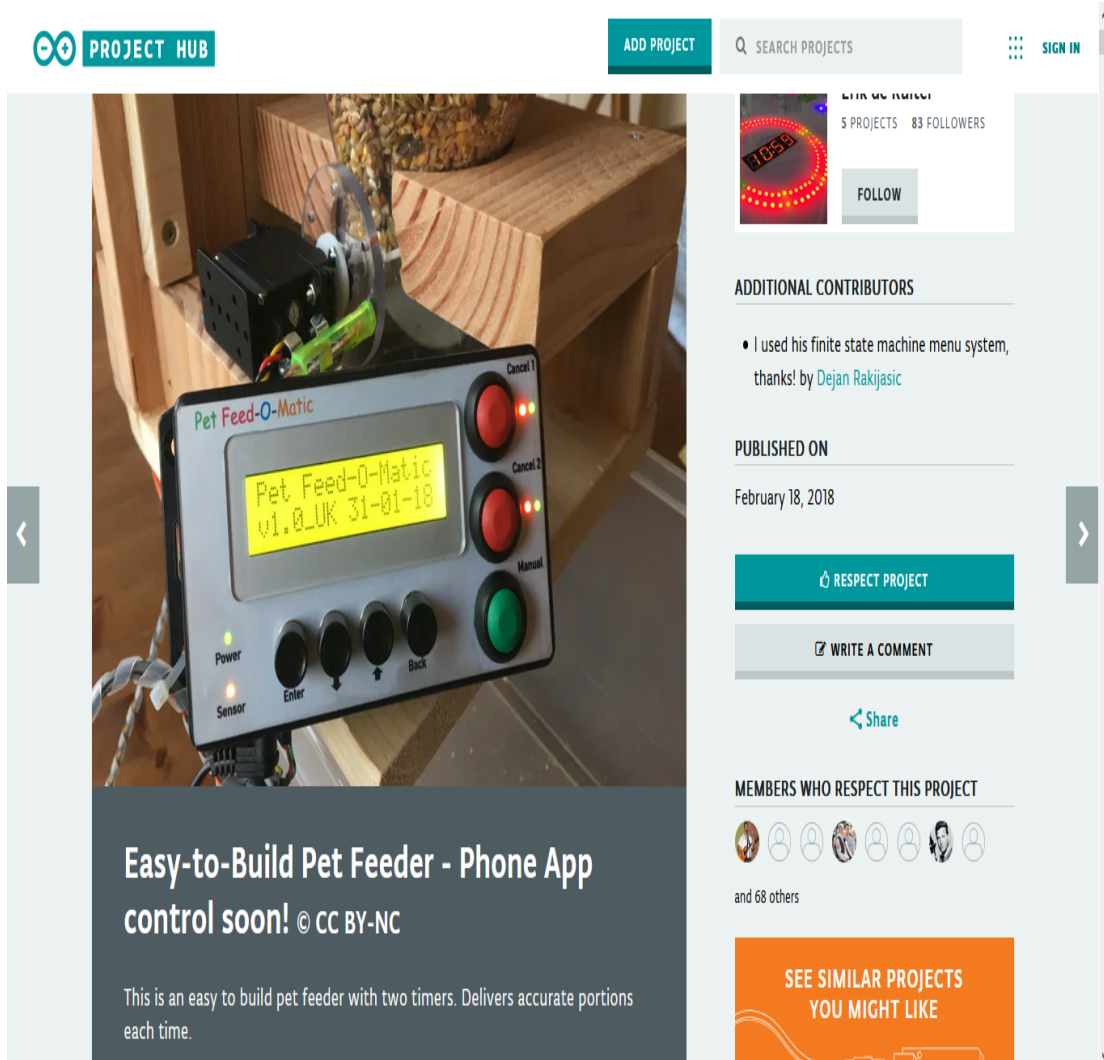
Ένας ακόμα φοιτητής, ο Soroush Sedigh, στο Πολυτεχνείο της Καλιφόρνια σχεδίασε ένα σύστημα αυτόματης πλήρωσης τροφής με την χρήση δυναμοκυψέλης και arduino για την περάτωση της διπλωματικής του εργασίας. Στην εικόνα 2.11 φαίνεται μια εικόνα μέσα από την εργασία του με το τελικό προϊόν.



Εικόνα 2.11: Σύστημα πλήρωσης τροφής με χρήση arduino και δυναμοκυψέλης  
<https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1155&context=braesp>

Βλέπε αναφορά [11]

Ακόμα και η επίσημη ιστοσελίδα της Arduino έχει ασχοληθεί με το θέμα και έχει δημοσιεύσει οδηγό για την κατασκευή “Pet feeder” καθώς υπόσχεται ακόμα και να διαθέσει εφαρμογή στο κινητό για τον τηλεχειρισμό του.

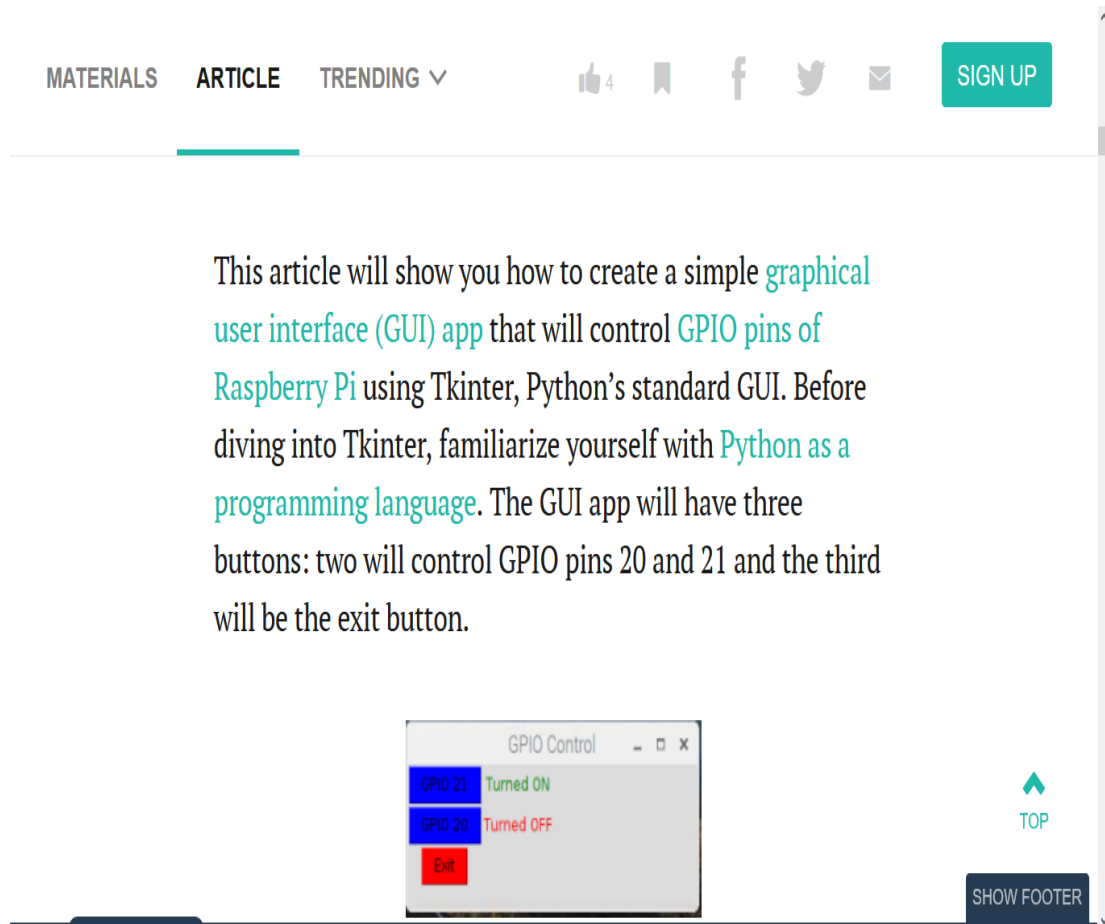


Εικόνα 2.12: Pet feeder by Arduino.cc

<https://create.arduino.cc/projecthub/edr1924/easy-to-build-pet-feeder-phone-app-control-soon-86c1ef>

Βλέπε αναφορά [12]

Περνώντας στην ανάπτυξη περιβάλλοντος για την διαδραστικότητα του χρήστη με το πρόγραμμα υπάρχει στην γλώσσα python η βιβλιοθήκη tkinter με την οποία μπορείς εύκολα να εμφανίσεις κουμπιά, και άλλα στοιχεία χρήσιμα για τον χρήστη, στην οθόνη καθώς και να τα προγραμματίσεις ανάλογα με τις ανάγκες. Στην εικόνα 2.12 φαίνεται ένα άρθρο που περιγράφει αναλυτικά πως φτιάχνεται ένα πρόγραμμα που θα ανάψει και θα σβήνει δύο LED.



Εικόνα 2.13: tkinter UI

<https://maker.pro/raspberry-pi/tutorial/how-to-use-tkinter-to-control-raspberry-pi-gpio-pins>

Βλέπε αναφορά [13]



## Κεφάλαιο τρίτο: Μεθοδολογία επίλυσης προβλήματος

Σε αυτό το κεφάλαιο θα αναλυθεί η μεθοδολογία που χρησιμοποιήθηκε για την επίλυση του προβλήματος. Αναφορά θα γίνει στα εργαλεία που χρησιμοποιήθηκαν για να σχεδιαστεί η μηχανολογική κατασκευή, στα μηχανήματα που χρειάστηκαν για να παραχθούν τα επιμέρους εξαρτήματα της, στα εργαλεία που χρησιμοποιήθηκαν για να αναπτυχθεί το λογισμικό και στους ελεγκτές που προγραμματίστηκαν με το λογισμικό αυτό.

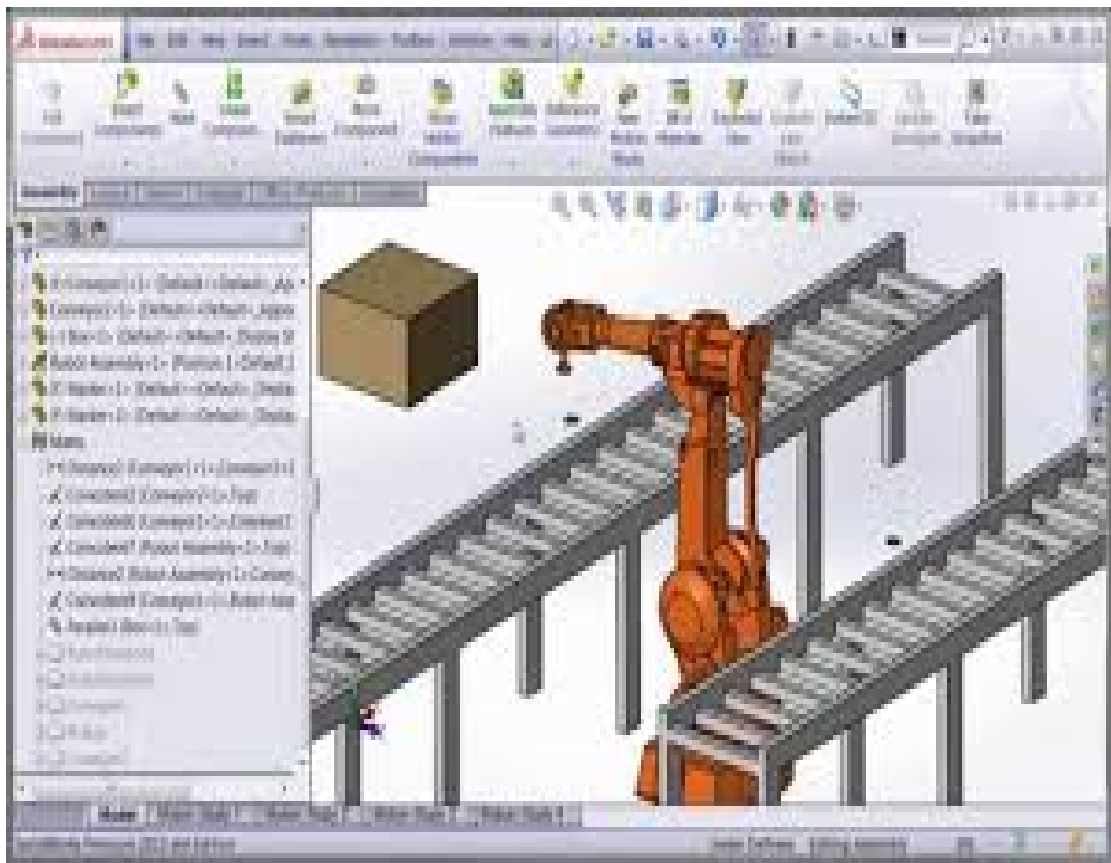
### Ενότητα 3.1 Ανάλυση συστήματος

Ξεκινώντας λοιπόν από την ιδέα ενός ρομποτικού συστήματος πλήρωσης στερεών προϊόντων, μια πρωταρχική ανάγκη είναι η απομακρυσμένη λειτουργία. Αυτή την ανάγκη την καλύπτει απόλυτα η επιλογή του μικροελεγκτή Arduino για τον έλεγχο των αισθητήρων και των κινητήρων και η επιλογή του Raspberry Pi για την ανάπτυξη γραφικού περιβάλλοντος για τον χρήστη και τον έλεγχο του Arduino. Χρειάζεται λοιπόν, ο σταθμός παραγωγής να είναι συνδεδεμένος σε κάποιο δίκτυο για διάφορους λόγους όπως διάγνωση βλαβών, καταγραφή δεδομένων κ.α. Μια ακόμη δυνατότητα που είναι απαραίτητη για να μπορεί ο σταθμός να χρησιμοποιηθεί για διαφορετικά προϊόντα είναι η ρύθμιση της ταχύτητας του κινητήρα. Η ρύθμιση της ταχύτητας του κινητήρα είναι απαραίτητη τόσο για την ακεραιότητα των προϊόντων όσο και για την μέγιστη δυνατότητα παραγωγής σε κάθε προϊόν. Η παρακολούθηση της θέσης του άξονα είναι ένα από τα πιο σημαντικά ζητήματα αυτού του συστήματος καθώς μόνο έτσι μπορεί να ρυθμιστεί η ποσότητα πλήρωσης προϊόντων, αφήνοντας την δυνατότητα στον χρήστη να την μεταβάλλει εύκολα.

Η παρακολούθηση της θέσης του άξονα γίνεται από τον κωδικοποιητή και τον μικροελεγκτή arduino. Η μεταφορά των προϊόντων από το δοχείο αναμονής προς τον κύλινδρο διαχωρισμού γίνεται με την βαρύτητα και τον οδηγό και η πλήρωση της συσκευασίας γίνεται με την περιστροφή του κυλίνδρου διαχωρισμού. Ο χειρισμός του συστήματος γίνεται από το περιβάλλον αλληλεπίδρασης ανθρώπου - μηχανής (Human - machine interface) που αναπτύχθηκε στην γλώσσα Python με την βιβλιοθήκη tkinter και λειτουργεί πάνω στο Raspberry Pi.

## Ενότητα 3.2 Διαδικασία σχεδιασμού - παραγωγής

Το πρώτο βήμα που έγινε για την επίλυση του προβλήματος, μετά βέβαια από την μελέτη του προηγούμενου κεφαλαίου, ήταν η ανάπτυξη τρισδιάστατων μοντέλων στο σχεδιαστικό πρόγραμμα Solidworks για να πάρει μορφή το σύστημα. Στο Solidworks εκτός από ανάπτυξη τρισδιάστατων μοντέλων, υπάρχει η δυνατότητα έκδοσης μηχανολογικών σχεδίων, την δημιουργία αρχείων που είναι συμβατά για εργαλειομηχανές (.DXF αρχεία), την δημιουργία αρχείων που είναι συμβατά με τρισδιάστατους εκτυπωτές (.STL, .OBJ) και πολλά ακόμα. Έχοντας αναπτύξει το τρισδιάστατο μοντέλο του συστήματος, σειρά έχει η δημιουργία των αρχείων (DXF, STL) για την κατασκευή των εξαρτημάτων σε αυτόματες εργαλειομηχανές και τρισδιάστατους εκτυπωτές καθώς και η έκδοση μηχανολογικών σχεδίων για την κατεργασία υλικών σε χειροκίνητες εργαλειομηχανές.



Εικόνα 3.1: Σχεδιαστικό περιβάλλον Solidworks

Από αυτόματες εργαλειομηχανές χρησιμοποιήθηκε ένα CNC - Laser από την επιχείρηση Χατζηπαντελής Α.Ε. για την κοπή των τεμαχίων μαλακού χάλυβα στις διαστάσεις και με τις γεωμετρίες που απαιτούσε το τρισδιάστατο μοντέλο. Να επισημανθεί πως το CNC - Laser έχει την δυνατότητα κατεργασίας (κοπής) φύλλων διαφόρων υλικών (μαλακός χάλυβας, ανοξείδωτος χάλυβας και αλουμίνιο) με ελευθερία κινήσεων της κεφαλής σε 2 άξονες, για την παραγωγή εξαρτημάτων που απαιτούν επεξεργασία σε δύο ή παραπάνω επίπεδα χρειάζεται η χρήση κέντρων κατεργασίας.



Εικόνα 3.2: CNC - Laser

Εκτός από το CNC - Laser, από αυτόματες εργαλειομηχανές χρησιμοποιήθηκε και ο τρισδιάστατος εκτυπωτής πλαστικού Ender 3 της Creality. Οι τρισδιάστατοι εκτυπωτές παράγουν εξαρτήματα (κυρίως από διάφορα πλαστικά) χρησιμοποιώντας μια θερμική κατεργασία του πλαστικού και την κίνηση της κεφαλής εκτύπωσης (extruder) με ελευθερία κινήσεων σε 3 άξονες (x,y,z). Έτσι, ξεκινώντας από κάτω προς τα πάνω “χτίζει” το εξάρτημα που έχει σχεδιαστεί στο υπολογιστή. Ο τρισδιάστατος εκτυπωτής αγοράστηκε για τον σκοπό της διπλωματικής εργασίας όμως υπάρχουν διάφορες επιχειρήσεις που παρέχουν τις υπηρεσίες σχεδίασης και εκτύπωσης εξαρτημάτων με τρισδιάστατους εκτυπωτές.



Εικόνα 3.3: Τρισδιάστατος εκτυπωτής

Μια από τις χειροκίνητες εργαλειομηχανές που χρησιμοποιήθηκαν ήταν η στράντζα. Η στράντζα είναι μια εργαλειομηχανή που αντιγράφει το προφίλ του μαχαιριού σε ένα φύλλο μετάλλου με της χρήση υψηλής πίεσης. Συνήθως, τα προφίλ που χρησιμοποιούν στις στράντζες είναι τύπου “V” με διαφορετικές γωνίες και ύψος. Με την στράντζα συνήθως φτιάχνονται εξαρτήματα που χρειάζεται να μην είναι συγκολλητά και επίσης η γεωμετρία περιγράφεται από γωνίες σε διάφορα επίπεδα. Για την διπλωματική εργασία χρησιμοποιήθηκε η στράντζα στην επιχείρηση Χατζηπαντελής Α.Ε.



Εικόνα 3.4: Στράντζα

Επίσης, από χειροκίνητες εργαλειομηχανές χρησιμοποιήθηκε ο τόρνος. Στον τόρνο καταργάζονται κυλινδρικά εξαρτήματα κυρίως (άξονες, σωλήνες κ.α.) με την μέθοδο αφαίρεσης υλικού. Το προς καταργασία εξάρτημα τοποθετείται στον σφιγκτήρα (λεγόμενο και τσοκ) και αυτό περιστρέφεται καθώς ο χειριστής κινεί ένα μαχαίρια για την αφαίρεση του υλικού από το εξάρτημα. Με τον τόρνο συνήθως δημιουργούν σπειρώματα σε άξονες, διαβαθμίσεις διαμέτρου και τρύπες . Για την εργασία χρησιμοποιήθηκε ο τόρνος στην επιχείρησης Κόμβος Α.Ε.



Εικόνα 3.5: Τόρνος

Ακόμα μια μηχανή που χρησιμοποιήθηκε για την παραγωγή εξαρτημάτων είναι η ηλεκτροκόλληση. Η ηλεκτροκόλληση χρησιμοποιείται για την συγκόλληση δύο μεταλλικών εξαρτημάτων με την χρήση ρεύματος. Για την συγκόλληση εξαρτημάτων χρησιμοποιήθηκε η ηλεκτροκόλληση στην επιχείρηση Κόμβος Α.Ε.



Εικόνα 3.6 : Ηλεκτροκόλληση

### Ενότητα 3.3 Εμπορεύσιμα εξαρτήματα

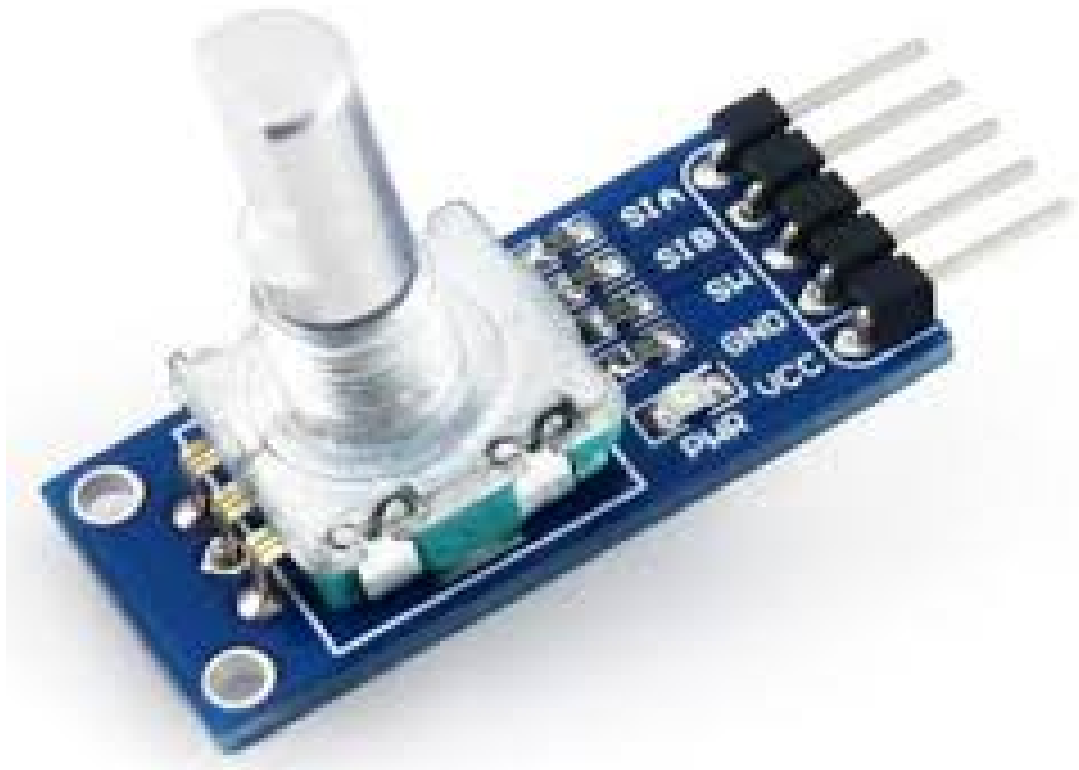
Σειρά, μετά από τα μηχανολογικά εξαρτήματα, έχουν τα ηλεκτρολογικά εξαρτήματα όπως ο κινητήρας. Για την κίνηση του άξονα χρησιμοποιήθηκε ένας κινητήρας συνεχούς ρεύματος με μειωτήρα στην έξοδο του. Οι κινητήρες συνεχούς ρεύματος έχουν το πλεονέκτημα πως ρυθμίζεται εύκολα η φορά περιστροφής τους χωρίς εξειδικευμένα ηλεκτρονικά κυκλώματα. Εκτός από την φορά περιστροφής, με το κατάλληλο ολοκληρωμένο κύκλωμα γίνεται και η ρύθμιση της ταχύτητας περιστροφής του. Σημαντικά μεγέθη που αφορούν τον κινητήρα είναι η ταχύτητα περιστροφής, η τάση λειτουργίας, η ροπή και το υλικό κατασκευής αυτού και του μειωτήρα αν έχει. Ο κινητήρας για την παρούσα διπλωματική αγοράστηκε από την εταιρία Grobotronics.



Εικόνα 3.7 : Ηλεκτροκινητήρας συνεχούς ρεύματος με μειωτήρα στην έξοδο



Εκτός από τον κινητήρα που κινεί τον άξονα, υπάρχει και ο κωδικοποιητής που ελέγχει την θέση του άξονα. Ο κωδικοποιητής έχει ένα ηλεκτρικό κύκλωμα που ανάλογα με την θέση ενός ρότορα αλλάζει κάποιες ψηφιακές εξόδους, έτσι σε συνδυασμό με το κατάλληλο λογισμικό και τον κατάλληλο ελεγκτή μπορεί το πρόγραμμα να ξέρει την θέση του άξονα. Σημαντικές πληροφορίες για έναν κωδικοποιητή είναι ο τύπος του (οπτικός, ηλεκτρικός, μαγνητικός), η ακρίβεια του και η διάρκεια ζωής του. Ο κωδικοποιητής για την παρούσα εργασία αγοράστηκε από την εταιρία Grobotronics.



Εικόνα 3.8: Παλμογεννήτρια

Ακόμα ένα εξάρτημα που χρειάστηκε για την υλοποίηση της εργασίας είναι το ρελέ. Το ρελέ είναι ένας διακόπτης που ελέγχεται από ένα ηλεκτρικό κύκλωμα και χρησιμοποιείται για λόγους προστασίας των ελεγκτών. Η Arduino έχει ολοκληρωμένες πλακέτες με ρελέ πάνω φτιαγμένες για να ελέγχονται από τους ελεγκτές της εύκολα και με κάποια feedback. Σημαντικές πληροφορίες για ένα ρελέ είναι η τάση λειτουργίας/είδος ρεύματος του τυλίγματος και η τάση λειτουργίας/είδος ρεύματος των επαφών. Για την παρούσα εργασία το ολοκληρωμένο με τα ρελέ αγοράστηκε από την Grobotronics.



Εικόνα 3.9: Ρελέ

Περνώντας τώρα στους ελεγκτές, για τον έλεγχο χαμηλού επιπέδου χρησιμοποιήθηκε ο Arduino nano. Το Arduino nano είναι μια πλακέτα που έχει την δυνατότητα να προγραμματιστεί ξανά

και έχει τα παρακάτω χαρακτηριστικά:

- Επεξεργαστή Atmel AVR (ATmega328)
- flash memory 32kb από την οποία τα 2kb δεσμεύονται από το bootloader
- SRAM 2KB
- Διαθέτει 8 αναλογικές εισόδους
- 22 ψηφιακές εισόδους / εξόδους, από τις οποίες οι 8 μπορούν να παράγουν παλμούς (PWM)
- 7-12 v-dc power input

Ο προγραμματισμός της γίνεται χρησιμοποιώντας την σειριακή θύρα του υπολογιστή και το πρόγραμμα Arduino IDE.



Εικόνα 3.10: Arduino Nano



Εικόνα 3.11: Arduino IDE

Όσο για τον έλεγχο υψηλού επιπέδου, επιλέχθηκε ένας υπολογιστής τσέπης Raspberry Pi για την επίλυση του προβλήματος. Το Raspberry Pi model 3 B+ που χρησιμοποιήθηκε στην παρούσα εργασία έχει τα παρακάτω χαρακτηριστικά:

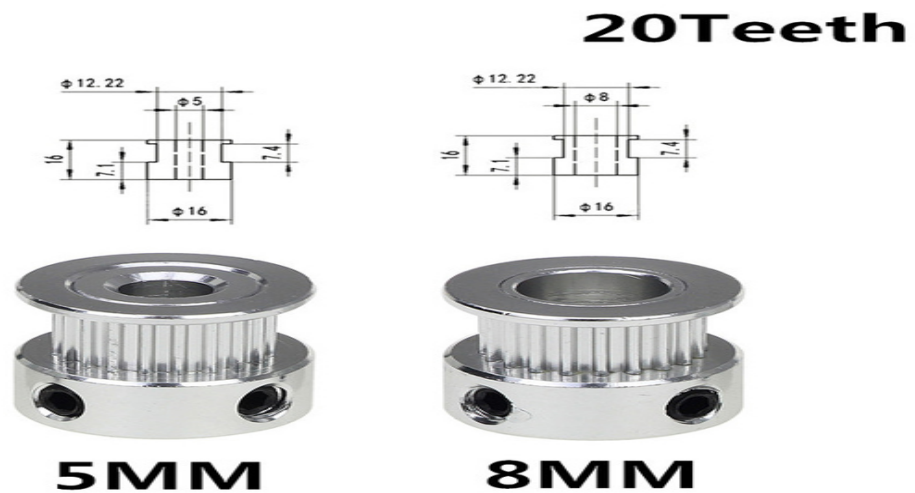
- επεξεργαστή της Cortex - A53 (ARMv8) στα 1.4GHz
- SDRAM 1GB
- Wifi
- Bluetooth 4.2
- Ethernet
- 40 GPIO pins
- 4 USB ports
- Full size - HDMI
- CSI camera port
- DSI touchscreen display port
- Micro SD
- 5v / 2.5A DC power input

Το λειτουργικό που χρησιμοποιεί το Raspberry Pi είναι κάποια μορφή Linux και αποθηκεύεται μαζί με τα δεδομένα στην Micro SD κάρτα.



Εικόνα 3.12: Raspberry Pi Model 3B+

Για την μετάδοση της κίνησης από τον κινητήρα στον κύλινδρο χρησιμοποιήθηκαν τυποποιημένα εξαρτήματα μετάδοσης κίνησης (γρανάζια) που συνεργάζονται με πλαστικό ιμάντα. Ο τύπος των γρανάζιων ήταν GT2 και το υλικό κατασκευής τους αλουμίνιο. Το πλήθος των δοντιών ήταν 20 και 40 στον κινητήρα και στον κινητήριο άξονα αντίστοιχα για να επιτυγχάνεται μια μείωση στροφών ακόμα.



Εικόνα 3.13 Γρανάζια για ιμάντα χρονισμού



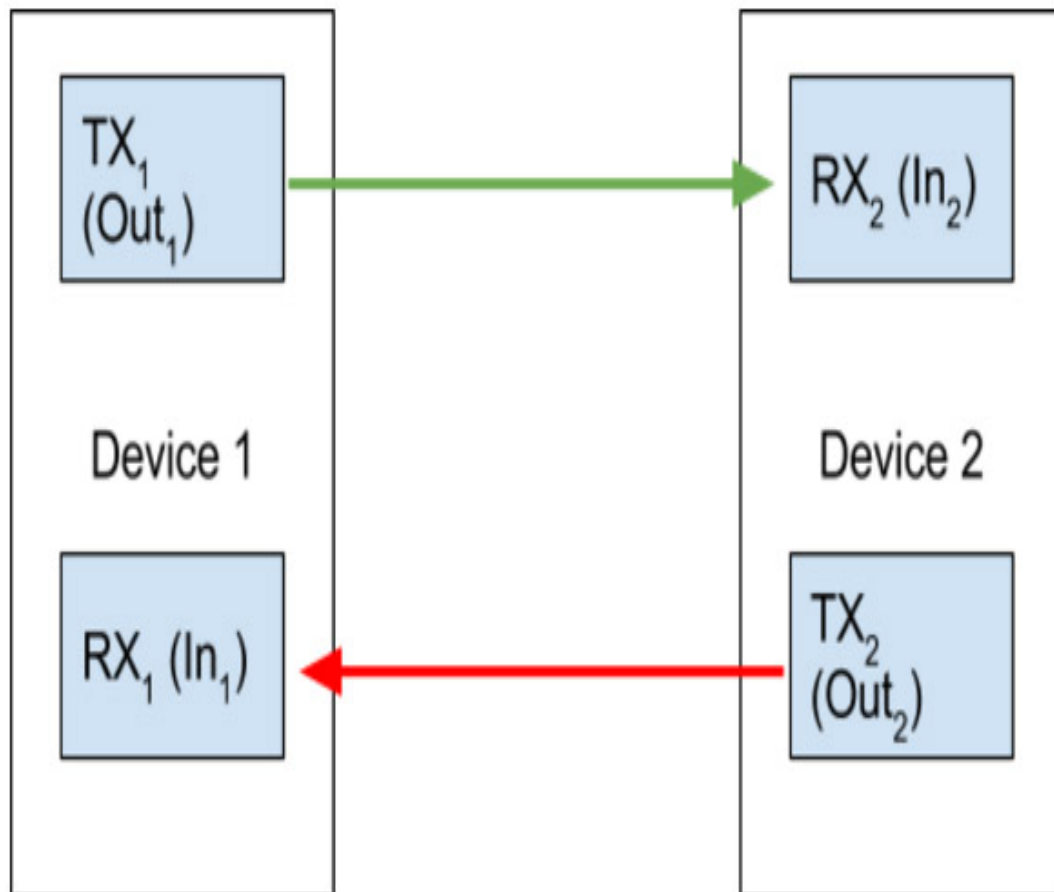
Εικόνα 3.14 Ζεύγος γρανάζιων ιμάντα

Για να τροφοδοτηθεί ο κινητήρας με τάση 12 v -dc επιλέχθηκε ένα τροφοδοτικό από φορητό υπολογιστή καθώς το φορτίο που είχε ο κινητήρας ήταν πολύ μικρό. Τα τροφοδοτικά υπολογιστή είναι πολύ εύκολο να τα βρει κανείς στην αγορά και για τον λόγο αυτό έχουν και πολύ χαμηλή τιμή.



Εικόνα 3.15 τροφοδοτικό υπολογιστή

Τέλος, η σειριακή επικοινωνία που επιλέχθηκε είναι αυτή που χρησιμοποιεί το arduino για να επικοινωνεί με τον υπολογιστή. Η σειριακή επικοινωνία είναι η σύνδεση 4 καλωδίων και στις δύο συσκευές, δυο καλώδια τάσης και δύο καλώδια πληροφορίας. Παρακάτω θα δούμε ένα παράδειγμα συνδεσμολογίας σειριακής επικοινωνίας

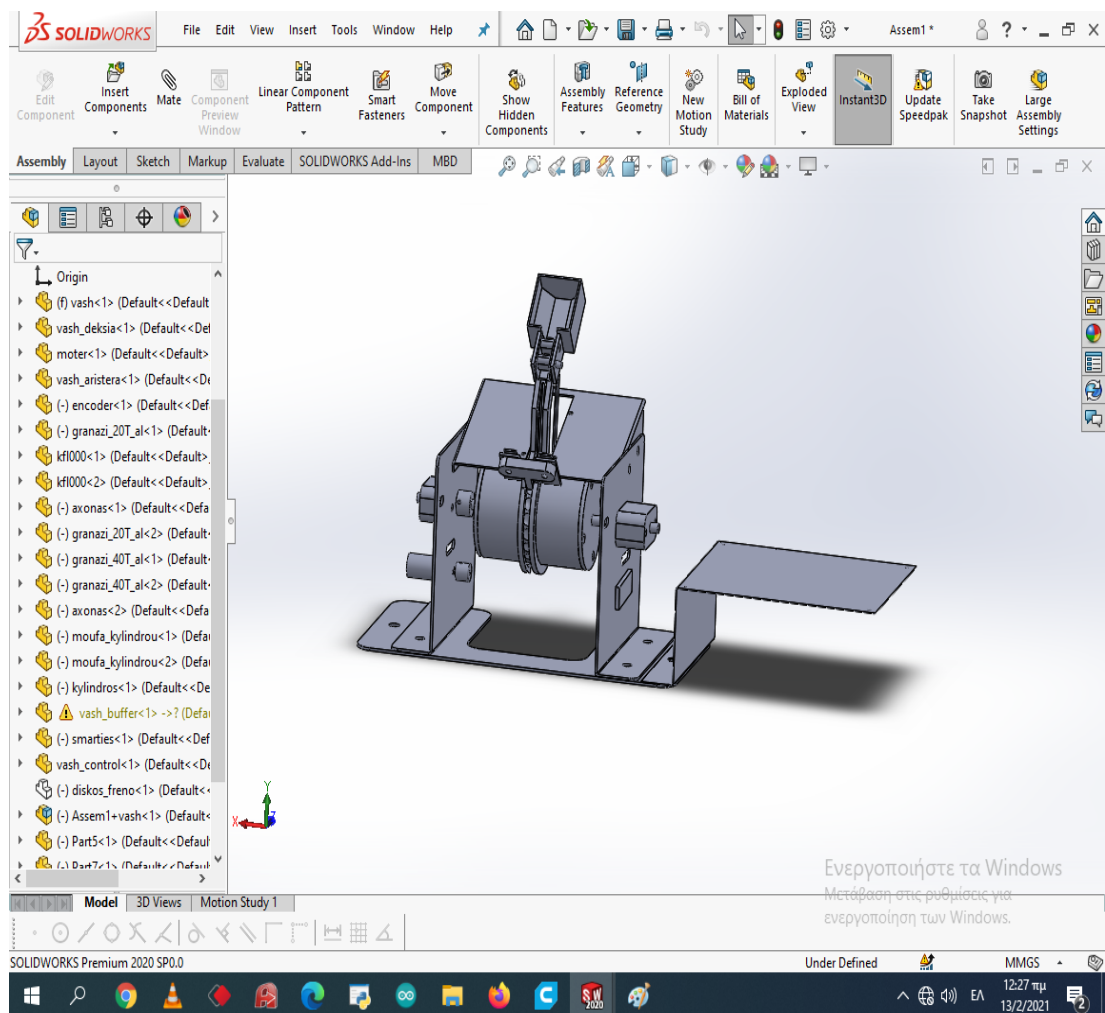


Εικόνα 3.16: Σειριακή επικοινωνία



## Κεφάλαιο τέταρτο: Σχέδια / τρισδιάστατη απεικόνιση

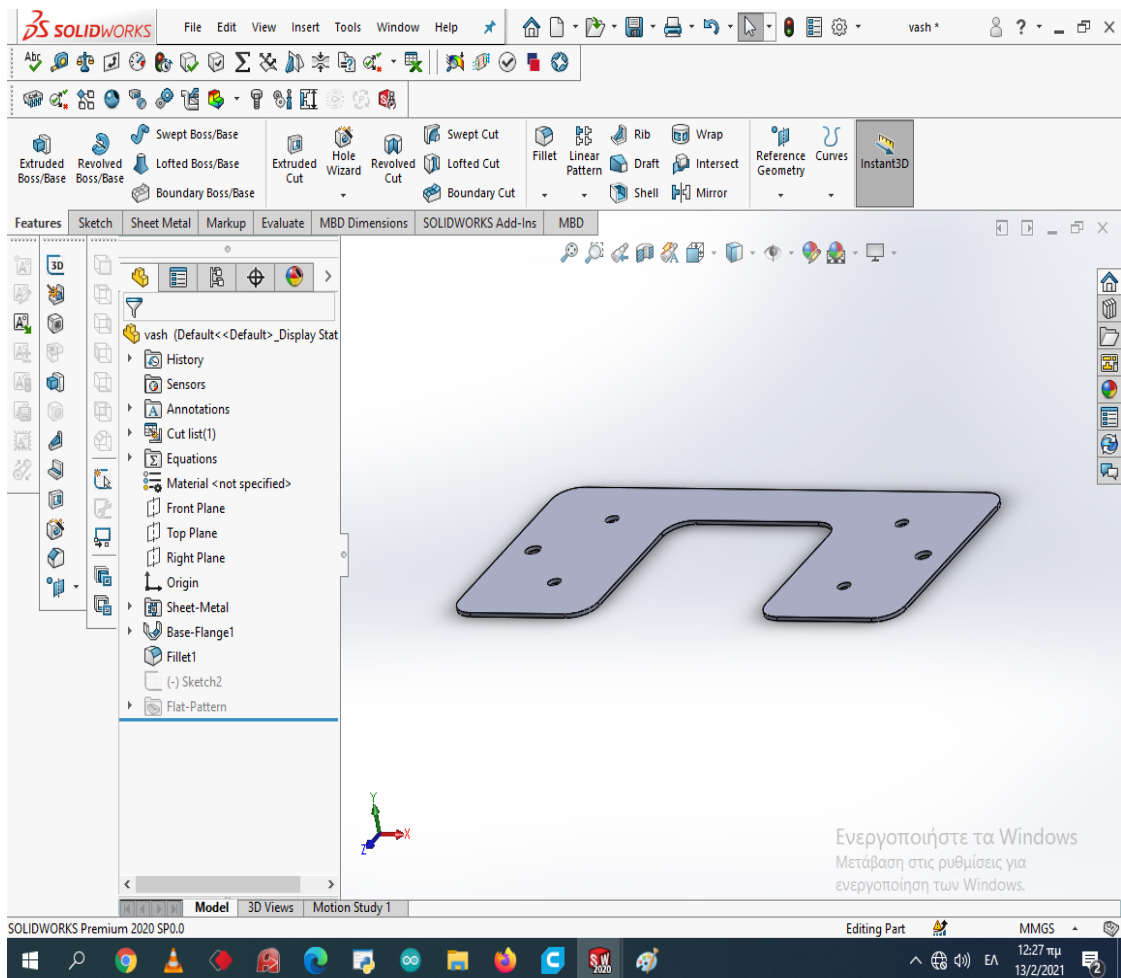
Σε αυτό το κεφάλαιο θα αναλυθούν τα μηχανολογικά και ηλεκτρολογικά σχέδια και θα αναλυθεί ο τρόπος που κατασκευάστηκαν τα πραγματικά εξαρτήματα. Το κάθε εξάρτημα εξυπηρετεί έναν σκοπό στο σύνολο της μηχανής και βάση αυτού σχεδιάστηκε. Εκτός βέβαια από τα εξαρτήματα που κατασκευάστηκαν, υπάρχουν και τα εξαρτήματα που αγοράστηκαν έτοιμα από το εμπόριο, σε αυτά θα γίνει απλά για αναφορά με την φωτογραφία τους και θα αναλυθεί γιατί επιλέχθηκε αυτό το εξάρτημα. Το τρισδιάστατο μοντέλο, τα μηχανολογικά σχέδια καθώς και τα αρχεία κοπής δημιουργήθηκαν με την χρήση του προγράμματος Solidworks.



Εικόνα 4.1 Τρισδιάστατη απεικόνιση συστήματος

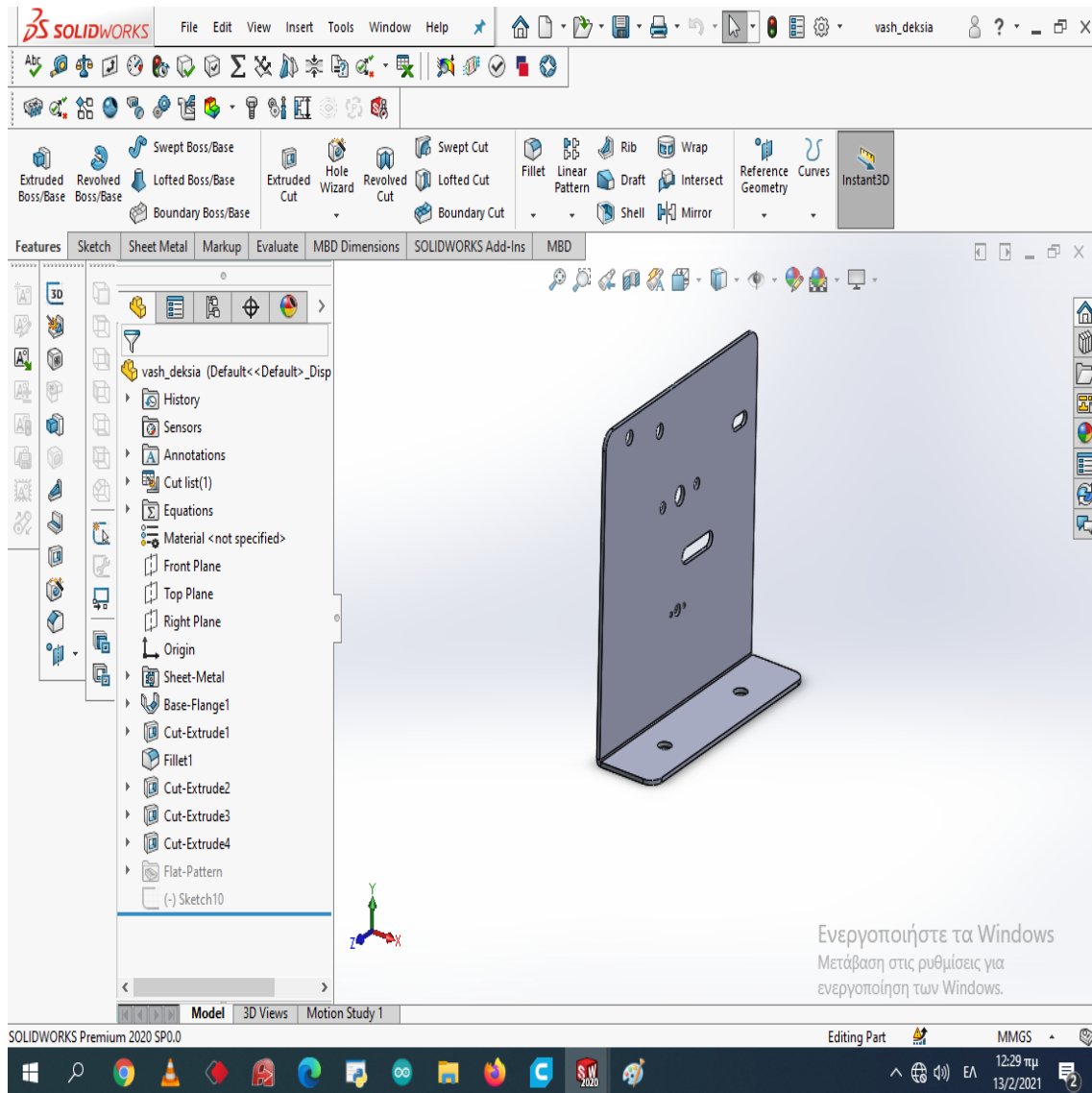
## Ενότητα 4.1 Μηχανολογικά σχέδια / 3D

Ξεκινώντας από το σασί, σκοπός του είναι να εξασφαλίσει την σωστή θέση και προσανατολισμό των επιμέρους εξαρτημάτων του συστήματος απορροφώντας τα μηχανικά φορτία που αυτό θα δέχεται. Το σασί για να είναι απλή η κατασκευή του έγινε σε 5 κομμάτια που ενώνονται με βίδες και αυτά έχουν υποδοχή για τα εξαρτήματα που θα στηρίζουν. Το πρώτο κομμάτι θα λειτουργεί σαν βάση για τα υπόλοιπα τρία που αυτά θα στηρίζουν το καθένα ξεχωριστά εξαρτήματα. Η καταργασία του εξαρτήματος αυτού έγινε στο CNC - Laser. Το υλικό που χρησιμοποιήθηκε είναι μαύρος μαλακός χάλυβας σε φύλλο λαμαρίνας.



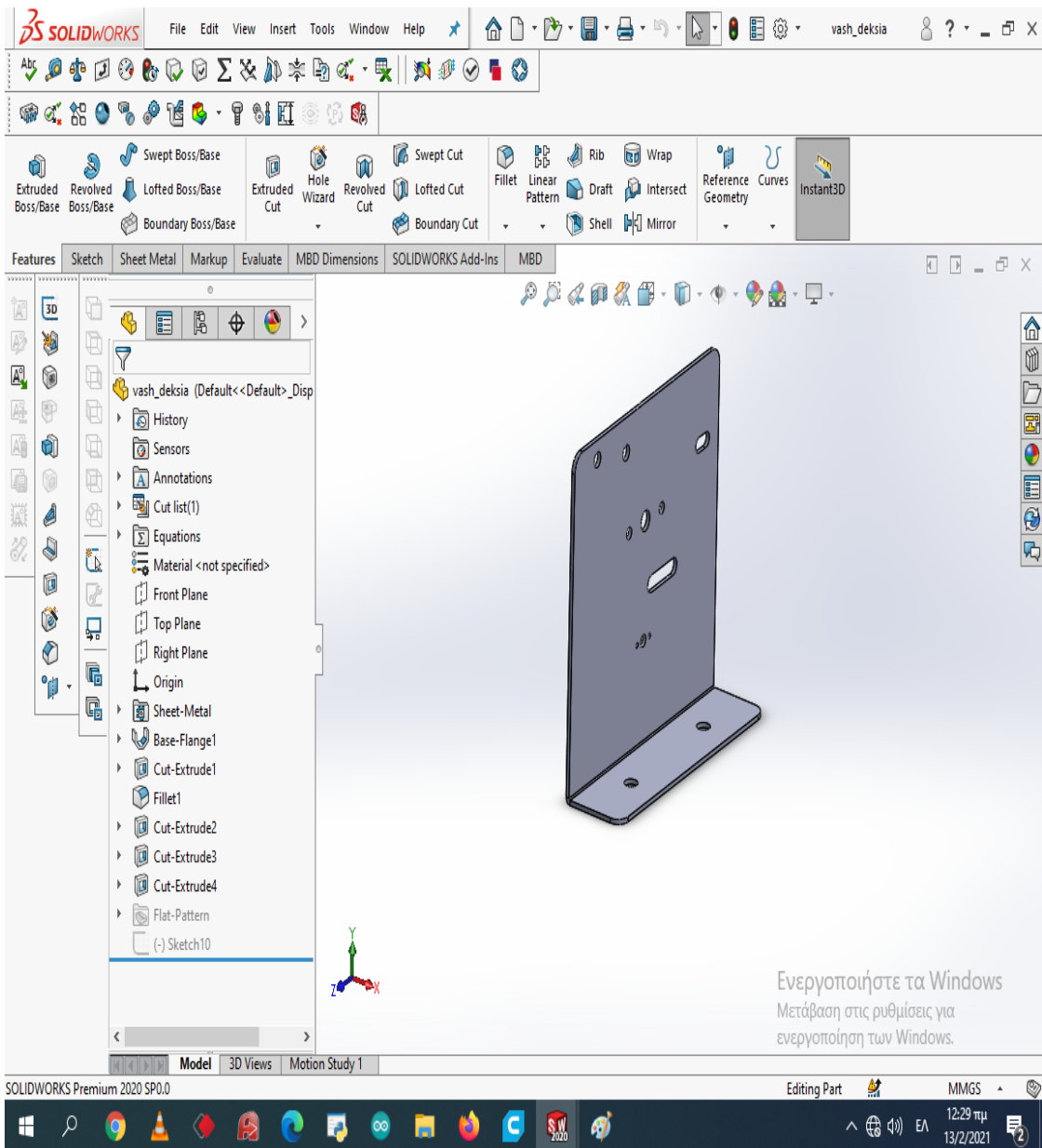
Εικόνα 4.2 Σασί συστήματος (εξάρτημα 1/5)

Το δεύτερο κομμάτι φτιάχτηκε για να στηρίζεται πάνω του ο κινητήρας, το ένα από τα δύο έδρανα καθώς και ο τεντωτήρας του μάντα. Στην γεωμετρία του προστέθηκε και μια γωνία 90 μοίρες για να μπορεί ο άξονας περιστροφής του εδράνου να είναι παράλληλος με το έδαφος. Η καταργασία του εξαρτήματος έγινε στο CNC - Laser και μετά η γωνία δημιουργήθηκε στην στράντζα. Το υλικό που χρησιμοποιήθηκε είναι μαύρος μαλακός χάλυβας σε φύλλο λαμαρίνας.



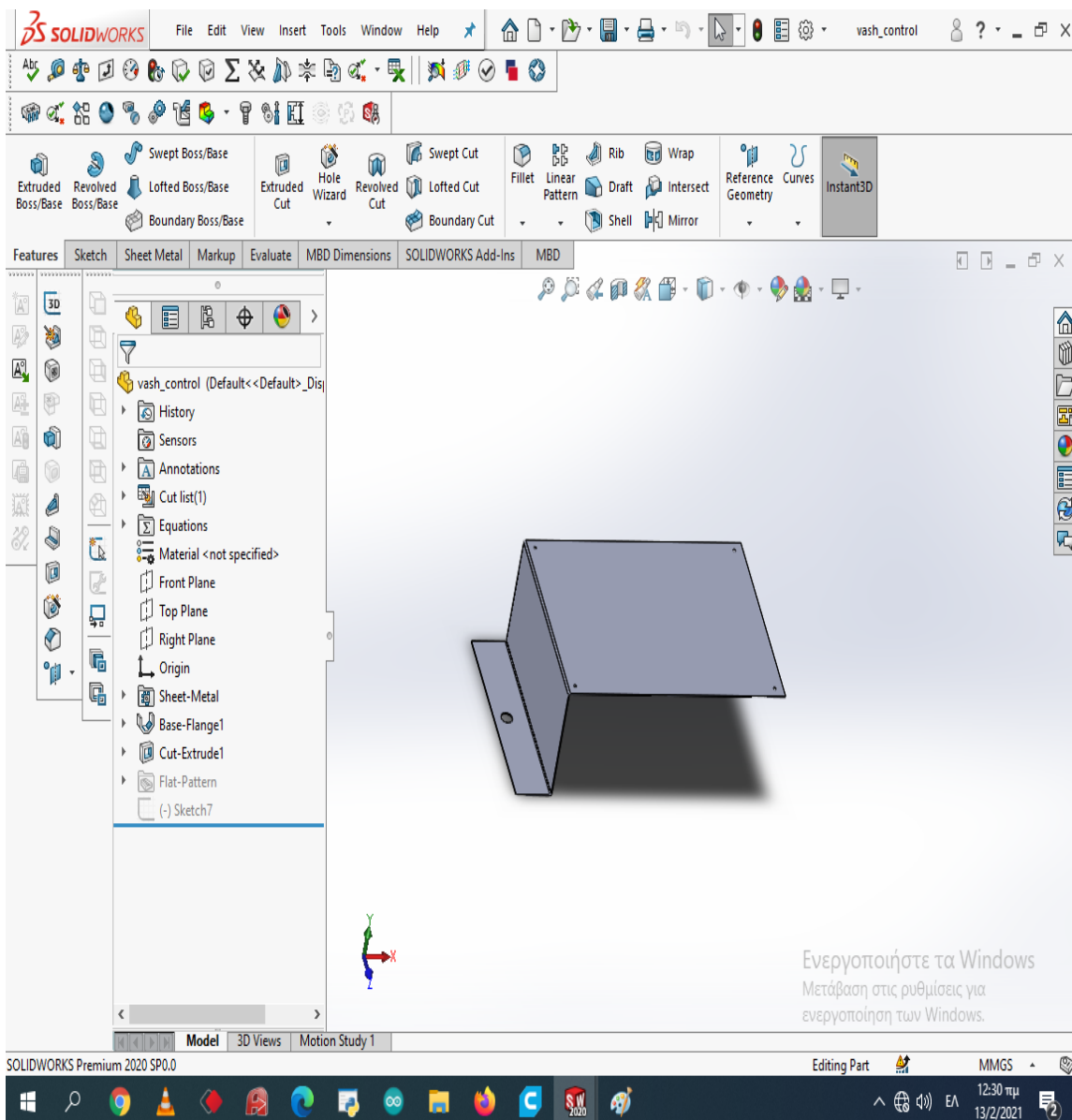
Εικόνα 4.3 Σασί συστήματος (εξάρτημα 2/5)

Το τρίτο κομμάτι φτιάχτηκε για να στηρίζεται πάνω του ο κωδικοποιητής, το άλλο έδρανο καθώς και ο τενωτήρας του μιάντα. Όπως και στο προηγούμενο εξάρτημα, στην γεωμετρία του προστέθηκε γωνία 90 μοίρες για να μπορεί ο άξονας περιστροφής του εδράνου να είναι παράλληλος με το έδαφος. Η κατεργασία του εξαρτήματος έγινε στο CNC - Laser και μετά η γωνία δημιουργήθηκε στην στράντζα. Το υλικό που χρησιμοποιήθηκε είναι μαύρος μαλακός χάλυβας σε φύλλο λαμαρίνας.



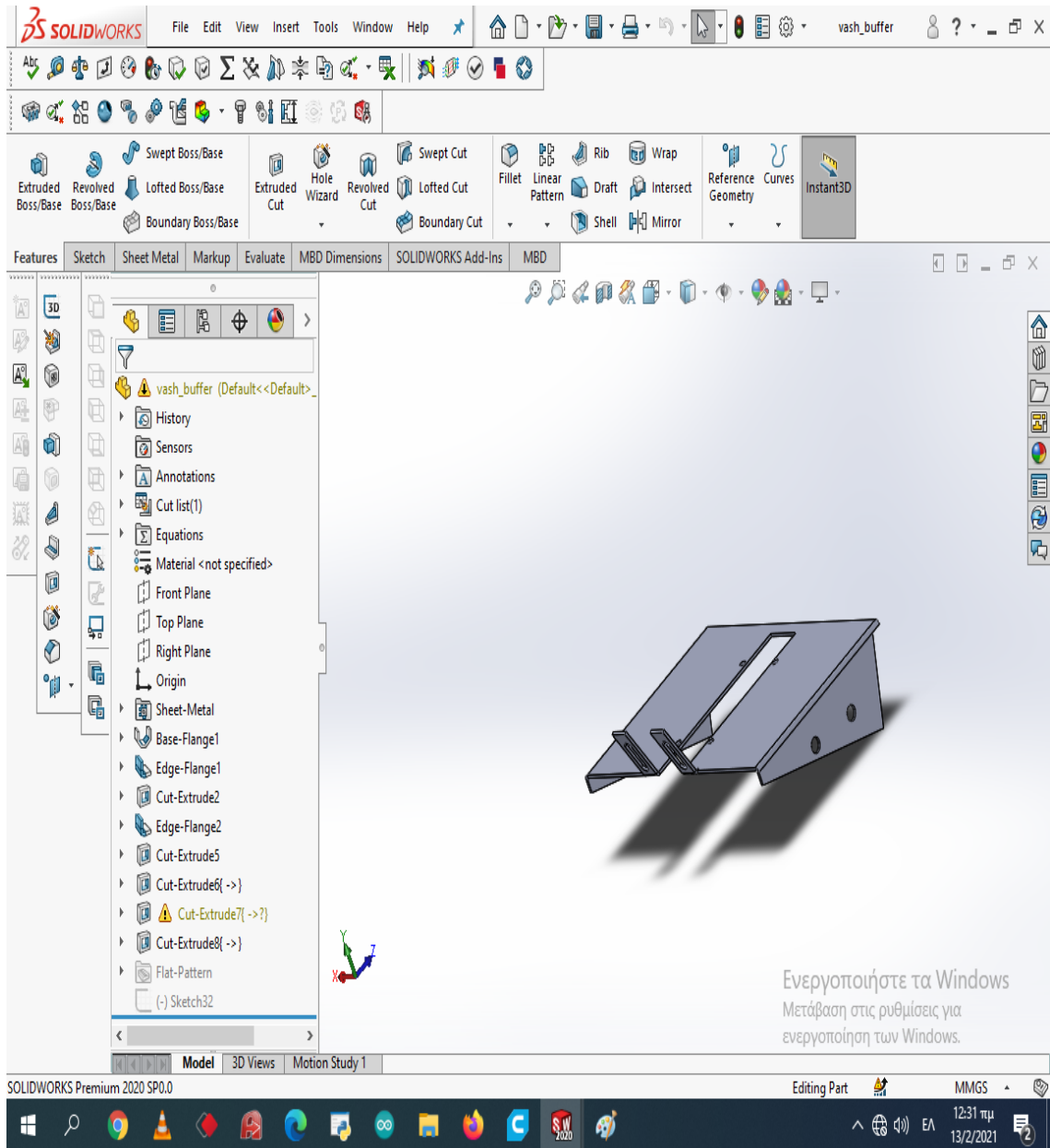
Εικόνα 4.4 Σασί συστήματος (εξάρτημα 3/5)

Το τέταρτο κομμάτι φτιάχτηκε για να στηρίζονται πάνω του τα ηλεκτρολογικά εξαρτήματα και να λειτουργεί σαν ένας πίνακας με σκοπό την εύκολη μεταφορά της κατασκευής. Στην κομμάτι αυτό πάνω είναι κολλημέρα τα ρελέ, το τροφοδοτικό, ο ελεγκτής και ο ρυθμιστής ταχύτητας. Εκτός από αυτά εκεί πάνω στηρίζονται και τα καλώδια που βρίσκονται μέσα στον “τακτοποιητή καλωδίων” που εξυπηρετεί την καλή αισθητική του τελικού ηλεκτρολογικού κυκλώματος. Η καταργασία του εξαρτήματος έγινε στο CNC - Laser και οι γωνίες δημιουργήθηκαν στην στράντζα. Το υλικό που χρησιμοποιήθηκε είναι μαύρος μαλακός χάλυβας σε φύλλο λαμαρίνας.



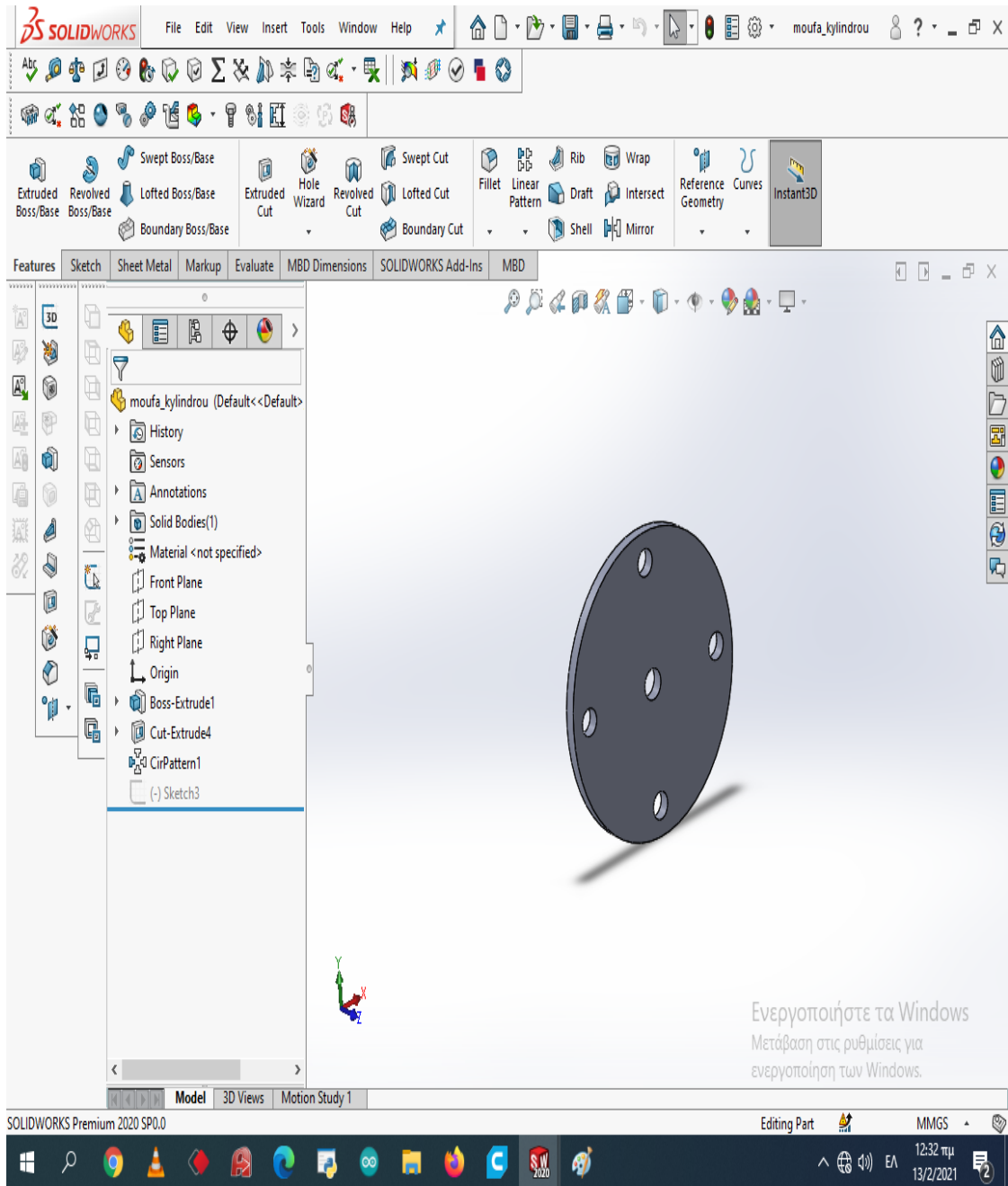
Εικόνα 4.5 Σασί συστήματος (εξάρτημα 4/5)

Το πέμπτο και τελευταίο κομμάτι είναι αυτό που φτιάχτηκε για να στηρίζει τους οδηγούς και το δοχείο αναμονής του προϊόντος. Εκτός από αυτά, βοηθάει και στην μηχανική απόσβεση των κραδασμών του συστήματος. Η καταργασία του εξαρτήματος έγινε στο CNC - Laser και οι γωνίες δημιουργήθηκαν στην στράντζα. Το υλικό που χρησιμοποιήθηκε είναι μαύρος μαλακός χάλυβας σε φύλλο λαμαρίνας.



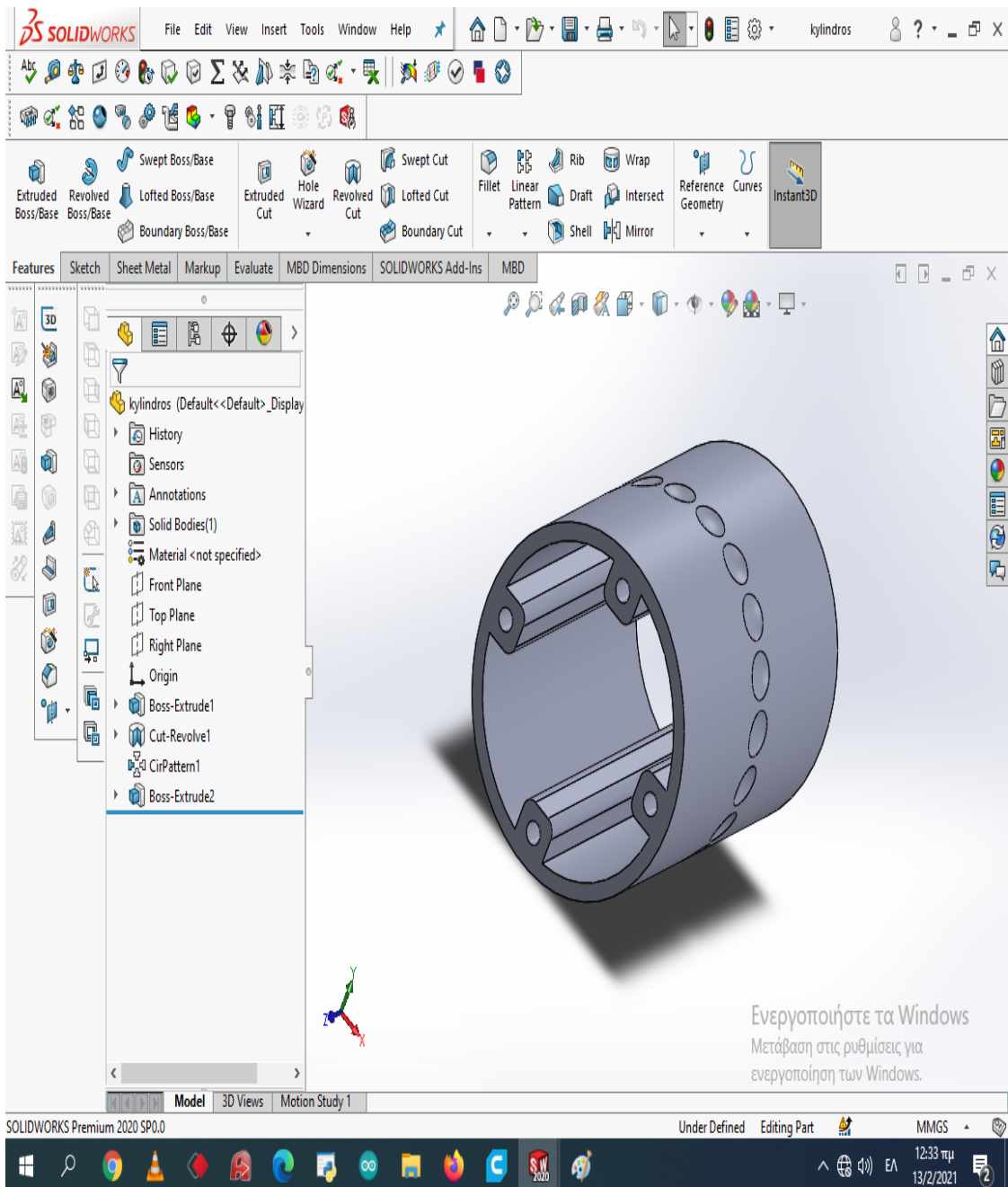
Εικόνα 4.6 Σασί συστήματος (εξάρτημα 5/5)

Εκτός από την βάση, δημιουργήθηκαν και δυο εξαρτήματα που θα κολληθούν πάνω στους άξονες και πάνω τους θα βιδωθεί ο κύλινδρος, οι λεγόμενες “μούφες”. Η καταργασία των εξαρτημάτων έγινε στο CNC - Laser και η συγκόλληση τους με τους άξονες έγινε με την χρήση ηλεκτροκόλλησης σύρματος. Το υλικό που χρησιμοποιήθηκε είναι μαύρος μαλακός χάλυβας σε φύλλο λαμαρίνας.



Εικόνα 4.7 Εξαρτήματα σύνδεση αξόνων με τον κύλινδρο

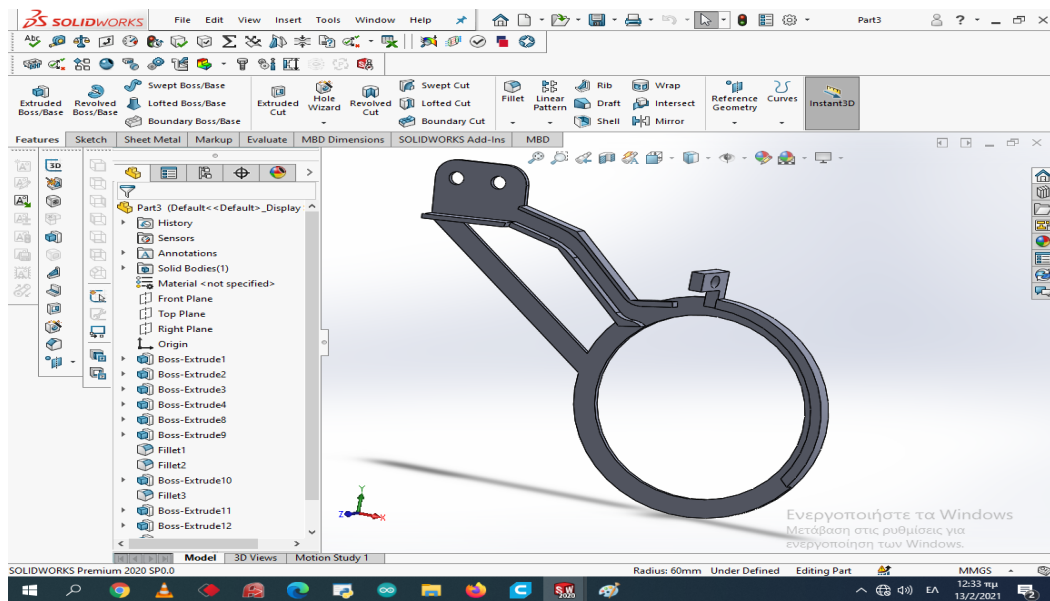
Ο κύλινδρος είναι από τα πιο βασικά εξαρτήματα του συστήματος. Σκοπός του είναι να περιστρέφεται, να υποδέχεται στα αυλάκια που έχει πάνω του το προϊόν και σε συνδυασμό με τους οδηγούς να μεταφέρουν το προϊόν στην συσκευασία του. Η κατασκευή του κυλίνδρου λόγω πολύπλοκης γεωμετρίας έγινε με την χρήση του τρισδιάστατου εκτυπωτή χρησιμοποιώντας υλικό PETG.



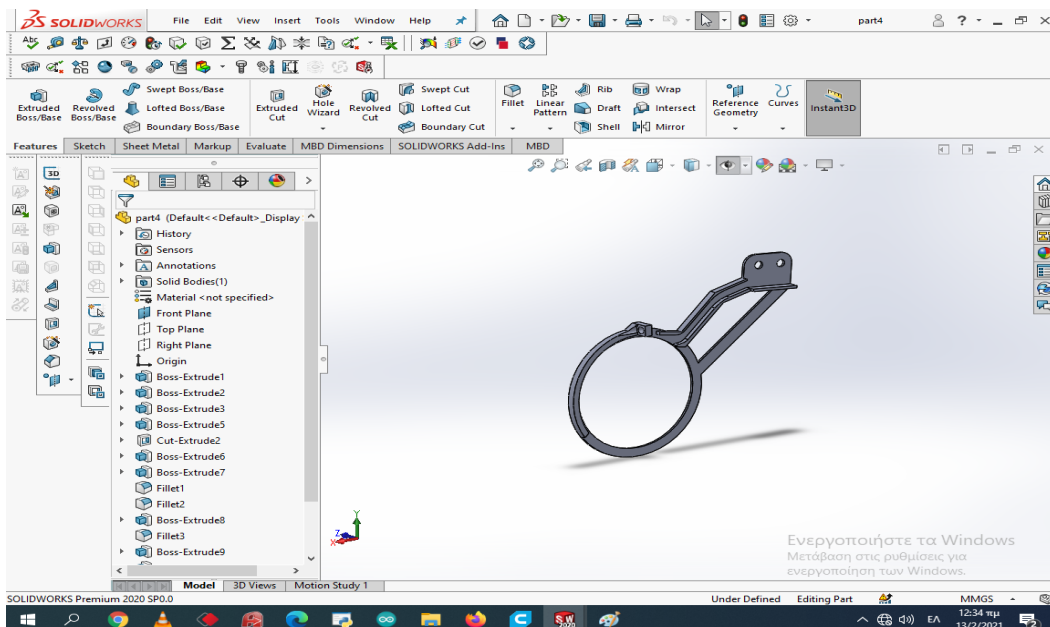
Εικόνα 4.8 Κύλινδρος



Οι οδηγοί τώρα του προϊόντος για να μπορέσουν να κρατήσουν το προϊόν μέσα στο αυλάκι έχουν μια ελεύθερη συναρμογή με τον κύλινδρο περιστροφής, τέτοια όμως που δεν επιτρέπει στο προϊόν όταν είναι ανάμεσα στο αυλάκι και στον οδηγό να μην παρασύρεται από την περιστροφή του κυλίνδρου. Η κατασκευή των οδηγών έγινε επίσης στον τρισδιάστατο εκτυπωτή με την χρήση του ίδιου υλικού PETG.

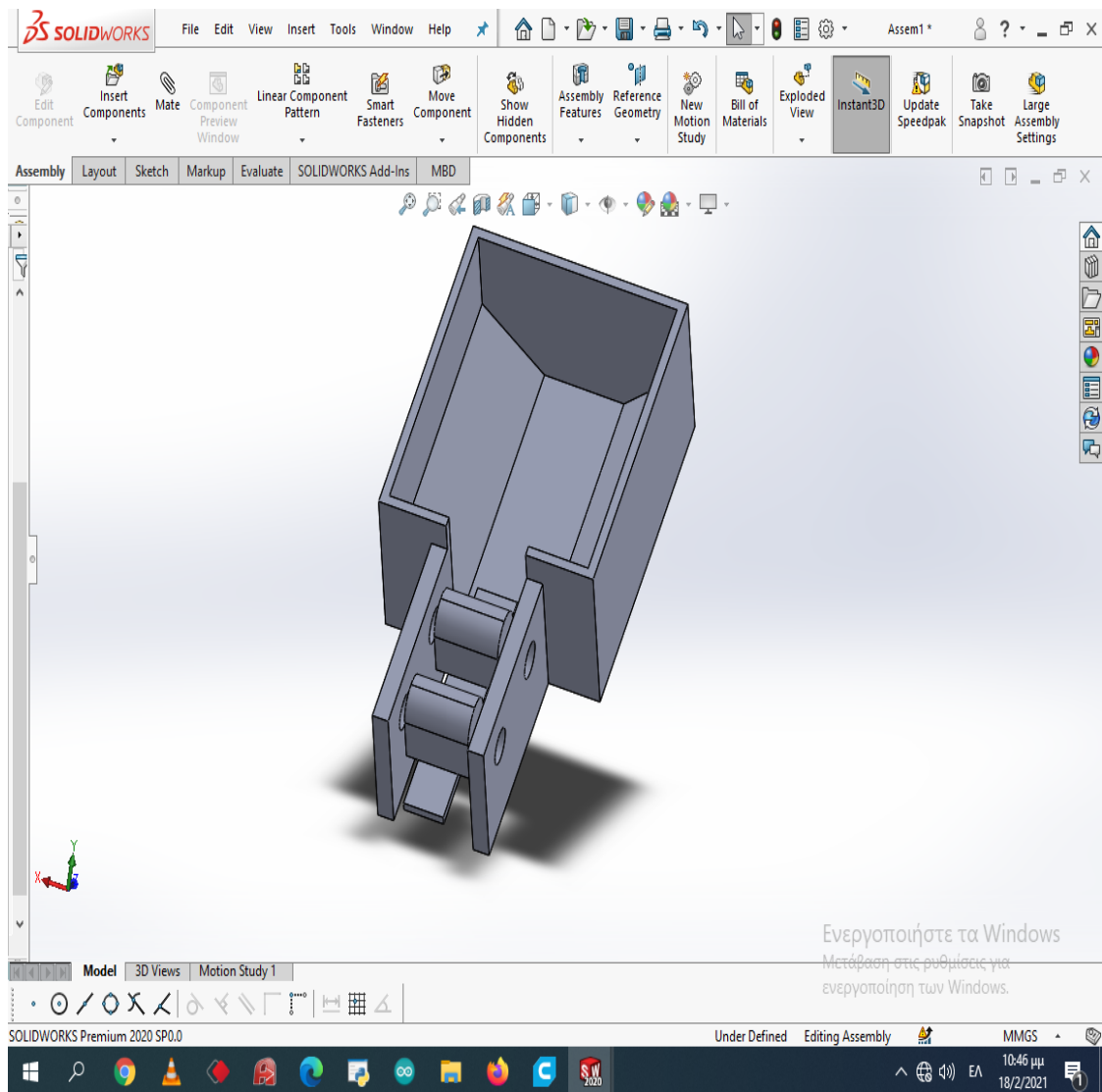


Εικόνα 4.9 Οδηγός (εξάρτημα 1/2)



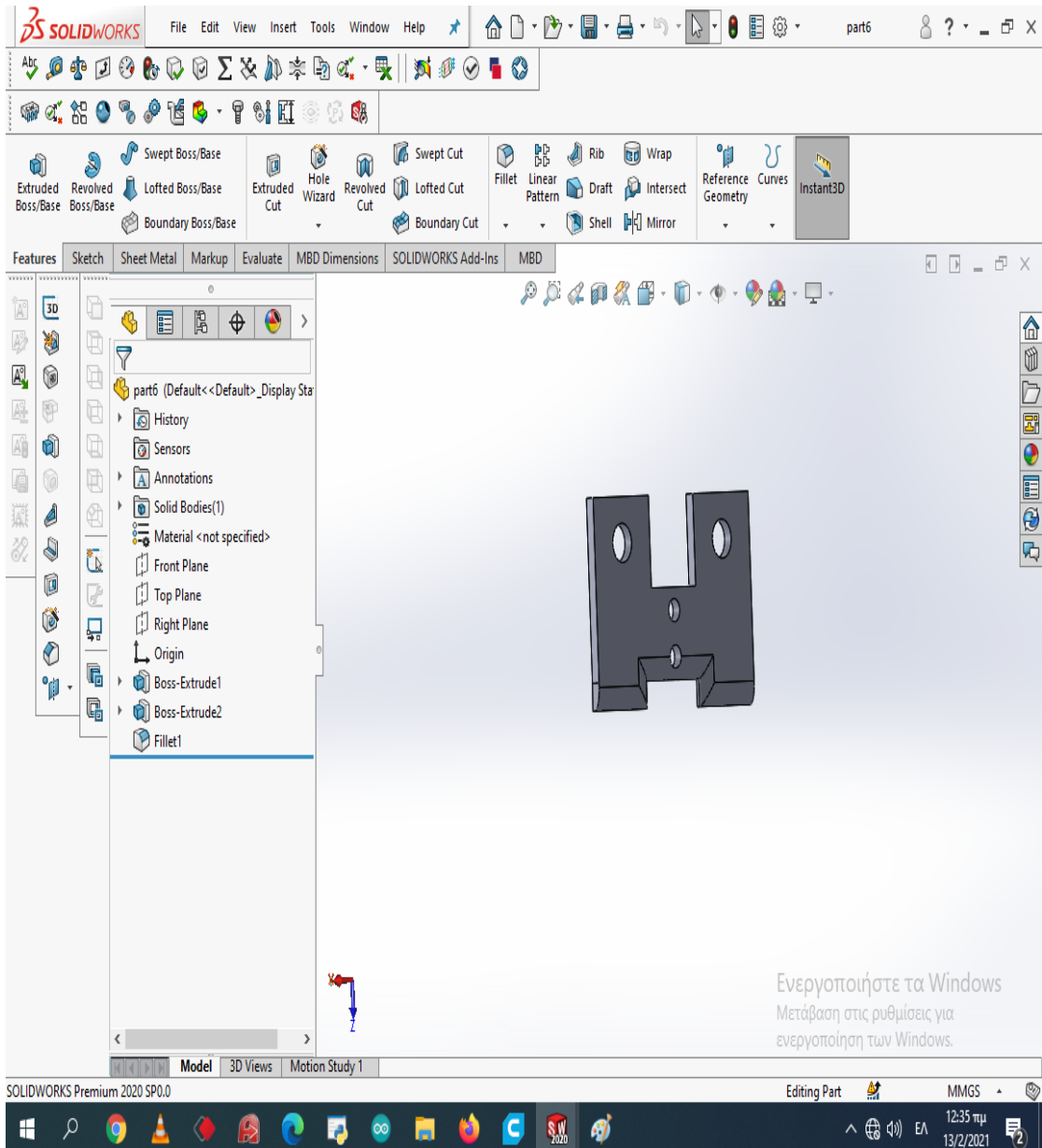
Εικόνα 4.10 Οδηγός (εξάρτημα 2/2)

Το δοχείο των προϊόντων που φτιάχτηκε για τους σκοπούς της εργασίας είναι μικρής χωρητικότητας έτσι ώστε να φανεί η εύρυθμη λειτουργία του συστήματος. Το σημαντικό στην γεωμετρία του είναι πως στο κάτω μέρος του υπάρχει μια τρύπα ώστε να περνάει ένα μόνο προϊόν την φορά και να μην δημιουργούνται εμπλοκές. Η κατασκευή του έγινε με την χρήση τρισδιάστατου εκτυπωτή και για υλικό χρησιμοποιήθηκε το PETG.



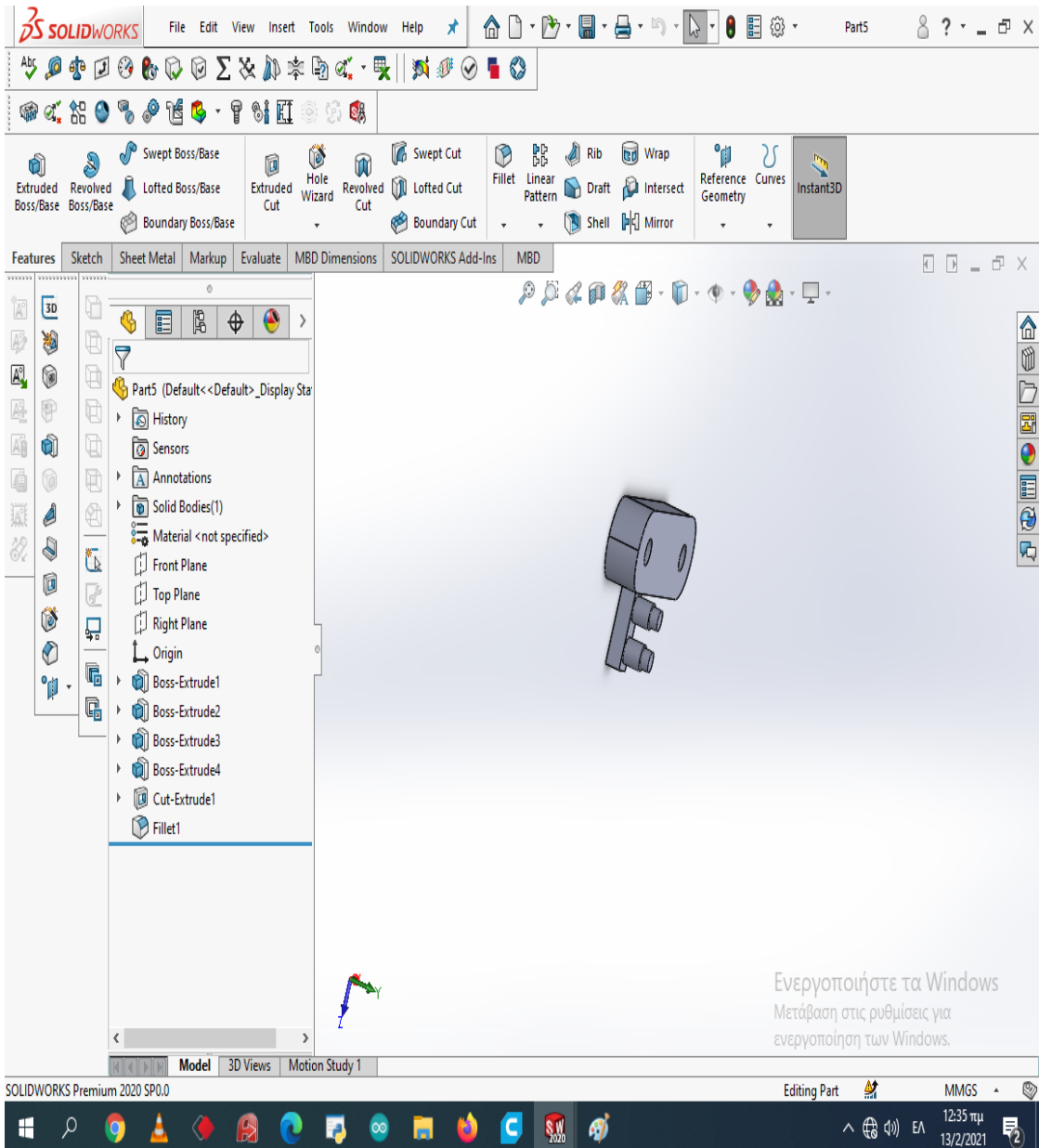
Εικόνα 4.11 Δοχείο προϊόντων

Ανάμεσα στο δοχείο προϊόντων και τους οδηγούς θα πρέπει να εξασφαλιστεί η ομαλή μετάβαση των προϊόντων για να αποφευχθούν οι εμπλοκές για αυτό κατασκευάστηκε ένα εξάρτημα το οποίο θα έχει πιο χαλαρή συναρμογή από τους οδηγούς όμως πάλι δεν θα αφήνει δύο προϊόντα να περάσουν στο ίδιο αυλάκι. Το εξάρτημα αυτό χρησιμοποιείται και για λόγους στήριξης των οδηγών. Η κατασκευή του έγινε με την χρήση τρισδιάστατου εκτυπωτή και το υλικό που χρησιμοποιήθηκε είναι το PETG.



Εικόνα 4.12 Βάση οδηγών

Εκτός από τους οδηγούς, φτιάχτηκε και ένα καπάκι ώστε να γίνει εύκολα η συναρμογή τους για να είναι εύκολη η αλλαγή των οδηγών για αλλαγή format κλπ. Το καπάκι αυτό με την βοήθεια δύο κοχλιών με περικόχλια θα συγκρατεί τους δύο οδηγούς και το σασί του συστήματος έτσι ώστε να κρατάνε οι οδηγοί σταθερό προσανατολισμό και θέση. Η κατασκευή του έγινε με την χρήση τρισδιάστατου εκτυπωτή και για υλικό χρησιμοποιήθηκε το PETG



Εικόνα 4.13 Καπάκι οδηγών

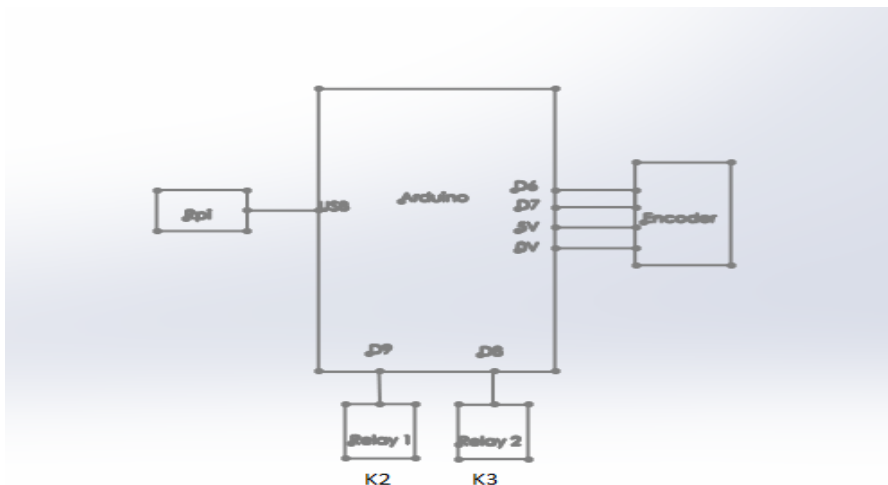
## Ενότητα 4.2 Ηλεκτρολογικά σχέδια

Περνώντας στο ηλεκτρολογικό σχέδιο, στο σύστημα υπάρχει μια τροφοδοσία ρεύματος 230V - AC η οποία συνδέεται στο τροφοδοτικό ώστε να την μετατρέψει σε 12V - DC. Με τάση 12 V - DC τροφοδοτούμε το ολοκληρωμένο κύκλωμα ρύθμισης ταχύτητας του κινητήρα περνώντας το καλώδιο με την υψηλότερη τάση (12V) από ένα ρελέ (K2). Στην συνέχεια, συνδέεται το ο ρυθμιστής ταχύτητας στις επαφές ενός ρελέ (K1) στο οποίο στους κοινούς ακροδέκτες είναι συνδεδεμένος ο κινητήρας. Με τον τρόπο αυτόν πετυχαίνουμε τον έλεγχο της φοράς περιστροφής του κινητήρα ανάλογα με την κατάσταση αυτού του ρελέ. Στην συνέχεια συνδέεται ο μικροελεγκτής arduino με τον κωδικοποιητή, τα ρελέ και τον υπολογιστή Raspberry Pi.

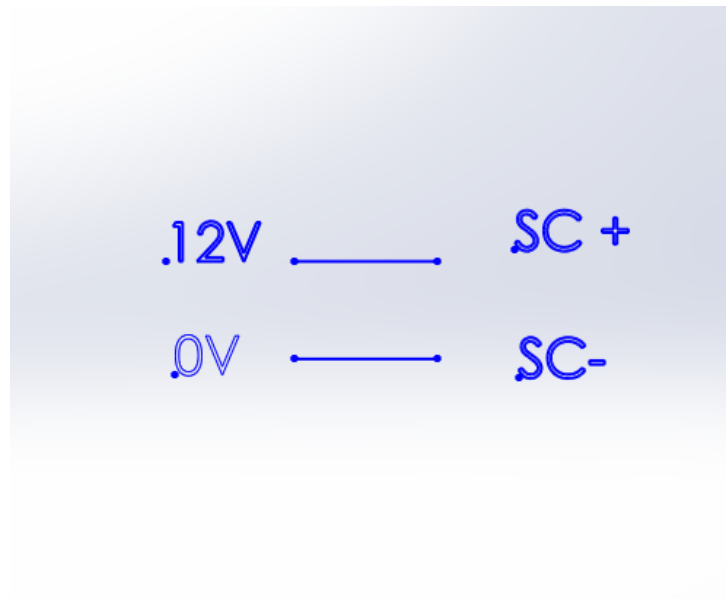
Αναλυτικά τα εξαρτήματα που χρειάζονται για την ηλεκτρολογική σύνδεση:

- Arduino
- Raspberry Pi
- 3 relay (2 relay 5v / 1 relay 12v)
- Ρυθμιστής ταχύτητας
- Κινητήρας
- Τροφοδοτικό 12 V

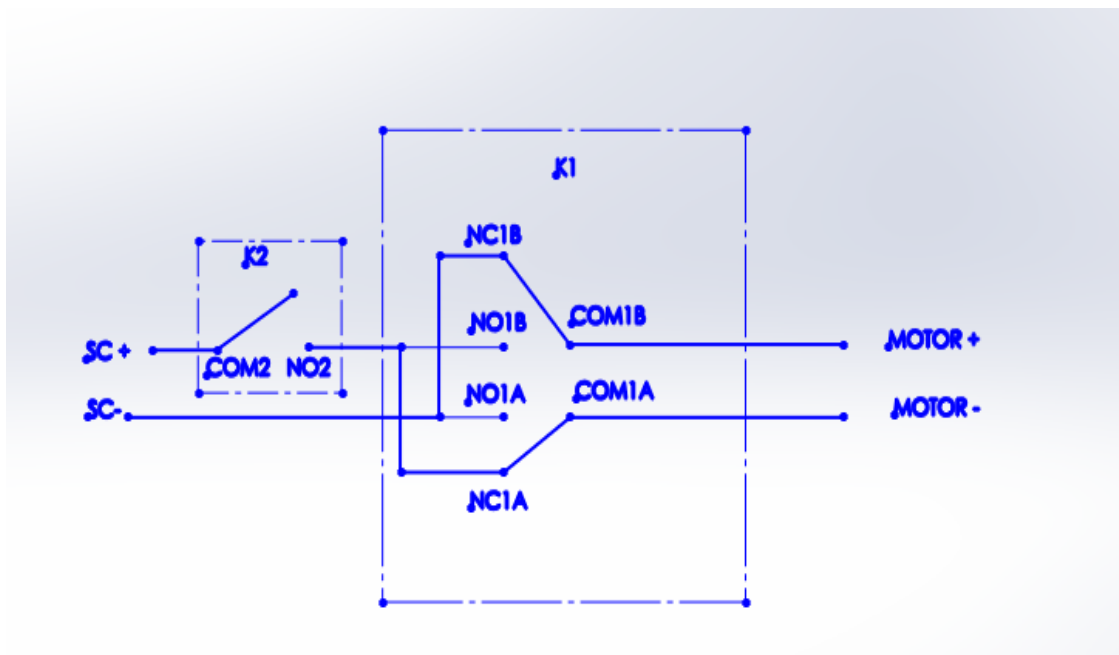
Ακολουθεί το ηλεκτρολογικό σχέδιο του συστήματος:



Σχήμα 4.1 Ηλεκτρολογικό σχέδιο Arduino



Σχήμα 4.2 Ηλεκτρολογικό σχέδιο ρυθμιστή ταχύτητας

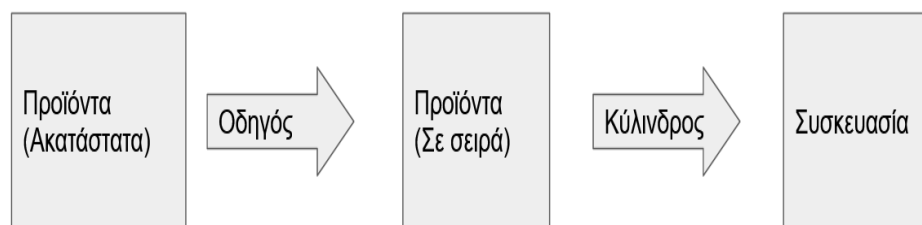


Σχήμα 4.3 Ηλεκτρολογικό σχέδιο ελέγχου κινητήρα

## Κεφάλαιο πέμπτο: Διαγράμματα ροής δεδομένων

Σε αυτό το κεφάλαιο θα παρουσιαστούν διαγράμματα που θα εξηγούν τον τρόπο λειτουργίας του συστήματος καθώς και την ροή των δεδομένων σε όλο το σύστημα. Δεδομένα μπορεί να εισάγει ο χρήστης στο σύστημα, μπορεί βέβαια και να συλλέγει το σύστημα από το περιβάλλον. Ξεκινώντας από το προϊόν, ακολουθεί η διαδρομή και οι φάσεις που περνάει κατά την παραγωγική διαδικασία.

### Ενότητα 5.1 Διαδρομή προϊόντος



Σχήμα 5.1 Διαδρομή προϊόντος

Στο παραπάνω σχήμα φαίνεται πως ξεκινάει το προϊόν μέσα στο δοχείο προϊόντων, σε ακατάστατη σειρά, και γίνεται η τακτοποίηση μέσω του οδηγού. Στην συνέχεια ο κύλινδρος παίρνει ένα ένα προϊόν και τα μεταφέρει στην συσκευασία. Τα προϊόντα είναι επιτακτική ανάγκη της παραγωγής να εισάγονται σε ακατάστατη σειρά για λόγους ταχύτητας, να μην ξεχνάμε πως ένας χειριστής έχει να τροφοδοτεί όλη την μηχανή με πρώτες ύλες και όχι μόνο έναν σταθμό παραγωγής οπότε πρέπει οι ενέργειες που απαιτούνται από τον χειριστή να είναι σύντομες και χωρίς πολλές ποιοτικές απαιτήσεις. Για τον λόγο αυτό, τα προϊόντα εισάγονται σε ακατάστατη μορφή από τον χρήστη στο δοχείο προϊόντων και στην συνέχεια, με την βοήθεια της βαρύτητας και μόνο αυτά πέφτουν προς τα κάτω. Αυτό που κάνει ο οδηγός είναι να δημιουργεί μια στένωση και έτσι να μπορεί να περάσει μόνο ένα προϊόν την φορά από εκεί. Αυτό έχει ως αποτέλεσμα να δημιουργηθεί μια σειρά από στοιχισμένα προϊόντα, το ένα πίσω από το άλλο στον οδηγό τα οποία περιμένουν τον κύλινδρο να περιστραφεί και να τα παρασύρει στην έξοδο.

## Ενότητα 5.2 Μονάδα ελέγχου υπολογιστή - περιβάλλον χρήστη

Στην συνέχεια βλέπουμε την διαδρομή των δεδομένων που σχετίζονται με τον χρήστη.



Σχήμα 5.2 Εντολή χρήστη



Στο σχήμα 5.2 φαίνεται πως η εντολή του χρήστη εισάγεται μέσω του υπολογιστή σε ένα python πρόγραμμα και αυτό εκτελεί κάποιες συναρτήσεις οι οποίες στέλνουν μέσω σειριακής επικοινωνίας δεδομένα στο arduino. Την δυνατότητα να επικοινωνεί ο χρήστης με το πρόγραμμα Python μας την δίνει η βιβλιοθήκη tkinter η οποία δημιουργεί ένα γραφικό περιβάλλον αλληλεπίδρασης με τον χρήστη (Graphical user interface) το οποίο προγραμματίστηκε έτσι ώστε ο χρήστης να μπορεί πατώντας κάποια κουμπιά να στέλνει εντολές στο σύστημα. Αυτές οι εντολές ξεκινάνε από το κουμπί που πατάει ο χρήστης, έχει φτιαχτεί ένα κουμπί για κάθε σετ από τροφοδότηση που θέλει να έχει ο χρήστης (π.χ. 1 τμχ, 2 τμχ, 3 τμχ κλπ) Εφόσον πατήσει το κουμπί ο χρήστης, παίρνει εντολή το πρόγραμμα python να στείλει μέσω της σειριακής ένα κωδικοποιημένο μήνυμα σε bytes στο arduino. Από την μεριά του το arduino, μόλις λάβει τα προκαθορισμένα μηνύματα διαμορφώνει κάποιες μεταβλητές έτσι ώστε να ρίξει τα απαιτούμενα τεμάχια.

Αναλύοντας το πρόγραμμα του υπολογιστή που θα εμφανίζει το περιβάλλον του χρήστη και θα επικοινωνεί μέσω της σειριακής θύρας με τον μικροελεγκτή πρέπει πρώτα να αναφέρουμε τις βιβλιοθήκες που χρειαζόμαστε. Για την σειριακή επικοινωνία η βιβλιοθήκη που έχει η γλώσσα προγραμματισμού python είναι η pyserial. Για την ανάπτυξη γραφικών στο περιβάλλον που βλέπει ο χρήστης η βιβλιοθήκη που χρησιμοποιήθηκε είναι η tkinter. Για τις καθυστερήσεις που χρειάστηκαν η βιβλιοθήκη είναι η time.

```
import serial
```

(Εισάγει την βιβλιοθήκη serial ώστε να μπορεί το πρόγραμμα python εύκολα να στείλει και να δεχθεί δεδομένα με την σειριακή επικοινωνία)

```
import time
```

(Εισάγει την βιβλιοθήκη time ώστε να μπορεί να δώσει εντολές χρονοκαθυστερήσης)

```
from tkinter import *
```

(Εισάγει την βιβλιοθήκη tkinter ώστε να μπορεί να φτιάξει το περιβάλλον του χρήστη)

Μετά την εισαγωγή των βιβλιοθηκών, σειρά στο πρόγραμμα έχει η αρχικοποίηση της σειριακής και του παραθύρου που θα εμφανίζει η tkinter. Για να αναπτυχθεί ένα πρόγραμμα με τις παραπάνω βιβλιοθήκες το οποίο θα μπορεί να λειτουργεί ως συνδετικός κρίκος ανάμεσα στον χειριστή και τον μικροελεγκτή πρέπει να ακολουθηθεί η παρακάτω διαδικασία ώστε να υπάρχει σωστή δομή. Πρέπει να οριστεί ένα παράθυρο στο οποίο θα εκτυπώνονται πάνω του κάποια κουμπιά και αυτά τα κουμπιά να δίνουν εντολή να τρέξει μια συνάρτηση.

### Ενότητα 5.2.2 Αρχικοποίηση προγράμματος

Όπως και στον μικροελεγκτή, έτσι και στο πρόγραμμα πρέπει να δηλωθεί σε ποιά σειριακή θύρα θα συνδεθεί, τι ρυθμό μεταφοράς δεδομένων θα έχει και να γίνει η αρχικοποίηση του περιβάλλοντος του χρήστη.

```
serialcomm = serial.Serial('/dev/ttyUSB0', 9600)
```

```
serialcomm.timeout = 1
```

(Αρχικοποιεί την σειριακή επικοινωνία με το arduino στην σειριακή θύρα USB0, με 9600 baud και timeout =1)

```
root=Tk();
```

```
root.title('Msc in Robotics - Smarties feeder User Interface')
```

```
root.geometry('600x400')
```

(Γίνεται η αρχικοποίηση του παραθύρου για τον χρήση με τίτλο και διαστάσεις)

### Ενότητα 5.2.3 Συναρτήσεις

Μετά την αρχικοποίηση πρέπει να οριστούν οι συναρτήσεις τις οποίες θα μπορεί να καλέσει ο χρήστης με το παρόν πρόγραμμα. Αυτές οι συναρτήσεις θα στέλνουν μηνύματα μέσω της σειριακής, θα παίρνουν τιμές από πεδία εισόδου που έχει ο χρήστης ή θα εκτυπώνουν κάποιο μήνυμα στον χρήστη.

```
def myclick1():
```

(Ορίζεται η συνάρτηση που θα εκτελείται από το πρώτο κουμπί)

```
label = Label(frame1, text = "Drop " + str(e.get()) + " Smarties")
```

```
label.grid(row=2, column =0)
```

```
command = e.get()
```

```
serialcomm.write(command.encode())
```

(Το πρώτο κουμπί θα εμφανίζει μια σήμανση που θα γράφει το παρακάτω μήνυμα και θα στέλνει σειριακά στο arduino την μεταβλητή command. Παρατηρείται

πως η μεταβλητή `command` εδώ γίνεται δυναμική καθώς την εισάγει ο χρήστης στο πεδίο `e` και ενημερώνεται το πρόγραμμα από την συνάρτηση `e.get()` )

```
def myclick2():
```

(Ορίζεται η συνάρτηση που θα εκτελείται από το δεύτερο κουμπί)

```
    command = "1"
```

```
    serialcomm.write(command.encode())
```

(Στο δεύτερο κουμπί η μεταβλητή `command` είναι στατική )

```
def myclick3():
```

(Ορίζεται η συνάρτηση που θα εκτελείται από το τρίτο κουμπί)

```
    command = "2"
```

```
    serialcomm.write(command.encode())
```

(Στο δεύτερο κουμπί η μεταβλητή `command` είναι στατική )

```
def myclick4():
```

(Ορίζεται η συνάρτηση που θα εκτελείται από το τέταρτο κουμπί)

```
    command = "3"
```

```
    serialcomm.write(command.encode())
```

(Στο δεύτερο κουμπί η μεταβλητή `command` είναι στατική )

```
def myclick5():
```

(Ορίζεται η συνάρτηση που θα εκτελείται από το πέμπτο κουμπί)

```
    command = "5"
```

```
    serialcomm.write(command.encode())
```

(Στο δεύτερο κουμπί η μεταβλητή `command` είναι στατική )

```
def myclick6():
```

(Ορίζεται η συνάρτηση που θα εκτελείται από το έκτο κουμπί)

```
    command = "6"
```

```
    serialcomm.write(command.encode())
```

(Στο δεύτερο κουμπί η μεταβλητή `command` είναι στατική )

```
def myclick0():
```

(Ορίζεται η συνάρτηση που θα εκτελείται από το δέκατο κουμπί)

```
    command = "0"
```

```
    serialcomm.write(command.encode())
```

(Στο δεύτερο κουμπί η μεταβλητή command είναι στατική )

## Ενότητα 5.2.4 Frames

Για να είναι εύκολη ο δόμηση του περιβάλλοντος του χρήστη, η βιβλιοθήκη tkinter δίνει την δυνατότητα να χωριστεί το παράθυρο σε πολλά frames, όπου frame είναι σαν μια ανεξάρτητη περιοχή μέσα στο παράθυρο. Έτσι στην παρούσα εργασία ακολουθήθηκε η παρακάτω δομή του περιβάλλοντος του χρήστη.

```
frame1 = LabelFrame(root, text="Automatic mode")
```

```
frame1.grid(row=0, column=0, sticky=N)
```

(Φτιάχνει μια περιοχή στο περιβάλλον χρήστη με ονομασία Automatic mode)

```
frame2 = LabelFrame(root, text="Setup ")
```

```
frame2.grid(row=0, column=1, sticky=N)
```

(Φτιάχνει μια περιοχή στο περιβάλλον χρήστη με ονομασία Setup)

```
frame3 = LabelFrame(root, text="Maintenance page")
```

```
frame3.grid(row=0, column=2, sticky=N)
```

(Φτιάχνει μια περιοχή στο περιβάλλον χρήστη με ονομασία Maintenance page)

## Ενότητα 5.2.5 Entries, Buttons, Grid

Στην βιβλιοθήκη tkinter, για να εισάγεις ένα κουμπί υπάρχει η συνάρτηση button στην οποία πρέπει να δηλώσεις που ανήκει αυτό το κουμπί, τι θες να γράφει πάνω και διάφορα άλλα ορίσματα που έχουν να κάνουν με χρώμα, μέγεθος, κατάσταση κ.α. Για να εισάγεις ένα πεδίο εισαγωγής για τον χρήστη υπάρχει η συνάρτηση entry που παίρνει σαν όρισμα μόνο το που ανήκει. Τέλος, για να εμφανιστούν στο παράθυρο υπάρχει η εντολή grid που όπως λέει και η λέξη (πλεγμα) δηλώνεται με σύστημα γραμμών - στηλών η θέση της κάθε οντότητας στο παράθυρο.

```
e = Entry(frame1)
```

```
e.insert(0, "Enter smarties quantity")
```

(Φτιάχνει ένα πεδίο που μπορείς να εισάγεις τον αριθμό των προϊόντων προς πλήρωση)

```
myButton1 = Button(frame1, text="trigger", padx =50, pady = 20,  
command=myclick1)
```

(Δίνει την εντολή να εκτελεστεί η συνάρτηση myclick1)

```
myButton2 = Button(frame2, text="Program 1 \n X1 pieces", padx =50, pady = 20, command=myclick2)
```

(Δίνει την εντολή να εκτελεστεί η συνάρτηση myclick2)

```
myButton3 = Button(frame2, text="Program 2 \n x2 pieces", padx =50, pady = 20, command=myclick3)
```

(Δίνει την εντολή να εκτελεστεί η συνάρτηση myclick3)

```
myButton4 = Button(frame2, text="Program 3 \n X3 pieces", padx =50, pady = 20, command=myclick4)
```

(Δίνει την εντολή να εκτελεστεί η συνάρτηση myclick4)

(Αποθηκευμένα προγράμματα )

```
myButton5 = Button(frame3, text="Manual forward", padx =47, pady = 27, command=myclick5)
```

(Δίνει την εντολή να εκτελεστεί η συνάρτηση myclick5)

```
myButton6 = Button(frame3, text="Manual backward", padx =40.5, pady = 27, command=myclick6)
```

(Δίνει την εντολή να εκτελεστεί η συνάρτηση myclick6)

```
myButton8 = Button(frame3, text="Set encoder to 0", padx =42, pady = 27, command=myclick0)
```

(Δίνει την εντολή να εκτελεστεί η συνάρτηση myclick0)

```
#myButton9 = Button(root, text="Command 20 backward", padx =50, pady = 20, command=myclick9)
```

(Δίνει την εντολή να εκτελεστεί η συνάρτηση myclick9)

```
e.grid(row=0, column=0)
```

```
myButton1.grid(row=1, column=0, sticky=W)
```

```
myButton2.grid(row=0, column=0, sticky=W)
```

```
myButton3.grid(row=1, column=0, sticky=W)
```

```
myButton4.grid(row=2, column=0, sticky=W)
```

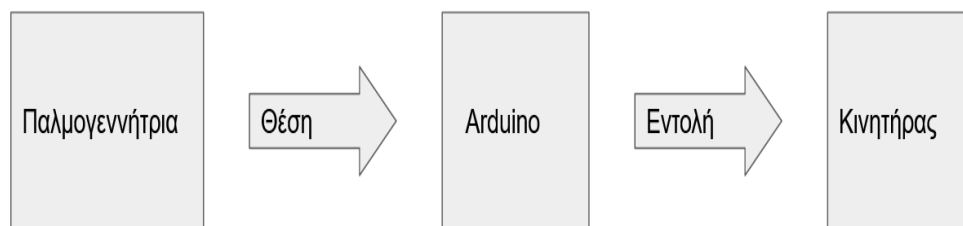
```
myButton5.grid(row=0, column=0, sticky=W)
```

```
myButton6.grid(row=1, column=0, sticky=W)
myButton8.grid(row=3, column=0, sticky=W)
root.mainloop()
```

## Ενότητα 5.3 Μικροελεγκτής

### Ενότητα 5.3.1 Μονάδα ελέγχου κινητήρα

Ο έλεγχος χαμηλού επιπέδου που εκτελεί το arduino φαίνεται στο παρακάτω σχήμα



Σχήμα 5.3 Έλεγχος χαμηλού επιπέδου από το Arduino

Στο σχήμα 5.3 φαίνεται το Arduino που δέχεται την πληροφορία της θέσης από την παλμογεννήτρια και ελέγχει την κατάσταση του κινητήρα. Το arduino για να ελέγξει την θέση του άξονα χρησιμοποιεί δύο μεταβλητές, μια που ενημερώνεται από την παλμογεννήτρια και αντίστοιχα δίνει εντολή στον κινητήρα και ακόμα μια μεταβλητή που ενημερώνεται από την σειριακή. Οπότε όταν ο χρήστης για παράδειγμα θέλει το σύστημα να ρίξει δύο τεμάχια, θα πατήσει το κουμπί με το πρόγραμμα 2, το πρόγραμμα rython θα στείλει μέσω της σειριακής στο arduino ένα μήνυμα, το arduino θα ενημερώσει την μια μεταβλητή και ο βρόγχος ελέγχου που τρέχει θα τροφοδοτήσει τον κινητήρα μέχρις ότου να περιστραφεί τόσο ο άξονας για να γίνουν οι μεταβλητές ίσες. Παρακάτω φαίνεται το κομμάτι κώδικα που πραγματοποιεί τον έλεγχο του κινητήρα.

/CHOOSE

(Αυτό το κομμάτι κώδικα είναι υπεύθυνο για τον έλεγχο του κινητήρα. Ανάλογα με την εντολή (tar) και την πραγματική θέση (counter) αλλάζει και την φορά περιστροφής του κινητήρα)

```
if( counter < tar -1){
  if (start_cw == 0 ){
    start_cw = 1;
    start_ccw =0;
    stp = 0;
    digitalWrite(motor, HIGH);
    digitalWrite(dir, HIGH);
    digitalWrite(motor, LOW);
  }
}
else if( counter > tar+1){
  if (start_ccw == 0){
    start_cw =0;
    start_ccw =1;
    stp = 0;
    digitalWrite(motor, HIGH);
    digitalWrite(dir, LOW);
    digitalWrite(motor, LOW);
  }
}
else if (counter < tar+1 and counter > tar -1) {
  if(stp == 0 ) {
    start_cw = 0;
    start_ccw = 0;
    stp = 1;
  }
}
```

```

    digitalWrite(motor, HIGH);
  }
}

```

Παρατηρείται πως χρησιμοποιούνται μετρητές για την ενεργοποίηση κάθε εντολής έτσι ώστε να μην είναι δυνατόν να ενεργοποιηθεί και η αριστερόστροφη και η δεξιόστροφη περιστροφή του κινητήρα και υπάρξει βραχυκύκλωμα στον ελεγκτή στροφών. Για την ενημέρωση την θέσης του άξονα από την παλμογεννήτρια χρησιμοποιείται το παρακάτω κομμάτι κώδικα

```
//GET_POS
```

(Αυτό το κομμάτι κώδικα είναι υπεύθυνο για να ενημερώνεται το arduino για την πραγματική θέση του άξονα)

```

aState = digitalRead(outputA);
(διαβάζει την έξοδο A)
if(aState != aLastState){
(Aν η έξοδος A είναι διαφορετική από την προηγούμενη μέτρηση)
  if(digitalRead(outputB) != aState){
(Aν και η έξοδος B είναι διαφορετική από την προηγούμενη μέτρηση)
    counter ++;
(Στην μεταβλητή counter πρόσθεσε 1)
    //Serial.print("counter: ");
    //Serial.println(counter);
  }
  else{
    counter --;
(Στην μεταβλητή counter αφάισεσε 1)
    // Serial.print("counter: ");
    // Serial.println(counter);
  }
}
aLastState = aState;

```

(Πριν τελειώσει αυτό το κομμάτι κώδικα ενημερώνει την τελευταία τιμή της εξόδου A με την τρέχουσα )



## Ενότητα 5.3.2 Αρχικοποίηση μικροελεγκτή arduino

Σε αυτό το κεφάλαιο θα αναλυθεί η αρχικοποίηση του μικροελεγκτή arduino καθώς για να λειτουργήσει σωστά πρέπει να το ενημερώσουμε για τα περιφερειακά εξαρτήματα που συνδέσαμε σε αυτό. Για την κάθε μεταβλητή του μικροελεγκτή πρέπει να δηλωθεί ο τύπος της καθώς και το όνομα της το οποίο μπορεί να συνοδεύεται με μια αρχική τιμή.

```
#define outputA 7
```

(Στον ψηφιακό ακροδέκτη 7 ανατίθεται το όνομα output A καθώς συνδέεται στην έξοδο A της παλμογεννήτριας)

```
#define outputB 6
```

(Στον ψηφιακό ακροδέκτη 6 ανατίθεται το όνομα output B καθώς συνδέεται στην έξοδο B της παλμογεννήτριας)

```
#define motor 9
```

(Στον ψηφιακό ακροδέκτη 9 ανατίθεται το όνομα motor καθώς συνδέεται στην εντολή του ρελέ που ενεργοποιεί τον κινητήρα)

```
#define dir 8
```

(Στον ψηφιακό ακροδέκτη 8 ανατίθεται το όνομα dir καθώς συνδέεται στην εντολή του ρελέ που αλλάζει την φορά περιστροφής του κινητήρα)

```
int counter = 0;
```

(Αρχικοποιείται ένας counter με σκοπό της μετατροπής των σημάτων της παλμογεννήτριας σε σχετική θέση του άξονα)

```
int aState;
```

(Αρχικοποιείται μια μεταβλητή που θα αποθηκεύεται σε αυτήν η τιμή από την κατάσταση της εξόδου A της παλμογεννήτριας)

```
int aLastState;
```

(Αρχικοποιείται μια μεταβλητή που θα αποθηκεύεται σε αυτήν η τιμή της προηγούμενης κατάστασης της παλμογεννήτριας)

```
int pos = 0;
```

(Αρχικοποιείται μια μεταβλητή για τον έλεγχο θέσης του άξονα)

```
int tar = 0;
```

(Αρχικοποιείται μια μεταβλητή για τον καθορισμό του στόχου για τον άξονα)

```
int start_cw = 0;
```

(Αρχικοποιείται μια μεταβλητή για την εξασφάλιση ότι δεν θα ενεργοποιηθούν δύο ρελέ μαζί ώστε να κάνουν βραχυκύκλωμα)

```
int start_ccw = 0;
```

(Αρχικοποιείται μια μεταβλητή για την εξασφάλιση ότι δεν θα ενεργοποιηθούν δύο ρελέ μαζί ώστε να κάνουν βραχυκύκλωμα)

```
int stp = 0;
```

(Αρχικοποιείται μια μεταβλητή για την εξασφάλιση ότι δεν θα ενεργοποιηθούν δύο ρελέ μαζί ώστε να κάνουν βραχυκύκλωμα)

Μετά από την αρχικοποίηση του μικροελεγκτή σειρά έχει η συνάρτηση void setup η οποία “τρέχει” μόνο μια φορά και μετά από αυτή ξεκινάει και εκτελείται η void loop. Στην void setup συνήθως μπαίνουν συναρτήσεις αρχικοποίησης καθώς και δηλώσεις ακροδεκτών. Οι δηλώσεις ακροδεκτών είναι απαραίτητες γιατί ο μικροελεγκτής πρέπει να ξέρει αν σε κάποιον ακροδέκτη πρέπει να δώσει η να περιμένει να πάρει ρεύμα για να μην υπάρχει κάποιο βραχυκύκλωμα. Εκτός από τις δηλώσεις ακροδεκτών, σε αυτήν την συνάρτηση συνήθως μπαίνει και το άνοιγμα της σειριακής θύρας στο οποίο δηλώνεται και ο ρυθμός μεταφοράς δεδομένων.

```
void setup() {
```

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

(Ξεκινάει η σειριακή επικοινωνία με 9600 baud)

```
  pinMode(outputA, INPUT);
```

(Ορίζει τον ακροδέκτη outputA ως έξοδο)

```
  pinMode(outputB, INPUT);
```

(Ορίζει τον ακροδέκτη outputB ως έξοδο)

```
  pinMode(motor, OUTPUT);
```

(Ορίζει τον ακροδέκτη motor ως έξοδο)

```
  pinMode(dir, OUTPUT);
```

(Ορίζει τον ακροδέκτη dir ως έξοδο)

```
  aLastState = digitalRead(outputA);
```

(Αρχικοποιεί την τιμή aLastState με μια μέτρηση στον ακροδέκτη outputA)

```
digitalWrite(motor, HIGH);
```

(Αρχικοποιεί την τιμή του ακροδέκτη motor σε 5v)

```
}
```

(Η συνάρτηση αυτή εκτελείται μια φορά από το arduino. )

Μετά από την void setup τρέχει η void loop στην οποία μπαίνει το κύριο κομμάτι του κώδικα.

```
void loop() {
```

(Αυτή η συνάρτηση τρέχει επ άοριστον στο arduino)

### Ενότητα 5.3.3 Σειριακή επικοινωνία υπολογιστή - μικροελεγκτή

Ως μέθοδος επικοινωνίας μεταξύ του υπολογιστή και του μικροελεγκτή επιλέχθηκε η σειριακή επικοινωνία. Η σειριακή επικοινωνία επιτυγχάνεται με δύο καλώδια και στην προκειμένη περίπτωση χρησιμοποιείται το καλώδιο usb. Τα δύο καλώδια είναι το Tx και το Rx τα οποία ουσιαστικά είναι πομπός και δέκτης. (T από Transmitter και R από Receiver) Στην σειριακή επικοινωνία τα δεδομένα μεταφέρονται με την μορφή bytes και αποθηκεύονται σε ένα buffer. Ο μικροελεγκτής arduino έχει την συνάρτηση serial.available() η οποία επιστρέφει τον αριθμό των bytes τα οποία είναι διαθέσιμα προς ανάγνωση στο buffer της σειριακής. Έχοντας λοιπόν προκαθορισμένα μηνύματα, τα οποία έχουν ίδιο μήκος (1 byte) είναι εύκολη η επικοινωνία μεταξύ του μικροελεγκτή και του υπολογιστή. Κάτι χαρακτηριστικό για τα bytes είναι πως ο αριθμός 0 δεν είναι και το byte 0 όμως το 48 για αυτό και όπου διαβάζει κάποιον αριθμό ο μικροελεγκτής, για να μεταφραστεί σε πραγματικό αριθμό αφαιρούμε το 48.

```
//Serial get target
```

(Αυτό το κομμάτι είναι υπεύθυνο για την επικοινωνία με το Raspberry Pi

```
if (Serial.available() > 0) {
```

(Η συνάρτηση `Serial.available()` επιστρέφει τον αριθμό των bytes που είναι στο `income buffer` της σειριακής. Άρα όταν υπάρχουν δεδομένα για να διαβάσει το `arduino`, η συνάρτηση αυτή θα επιστρέψει τον αριθμό των bytes που καταλαμβάνουν αυτά τα δεδομένα)

```
int command = Serial.read() - 48;
```

(Γίνεται η μετατροπή των `char` σε πραγματικού αριθμούς. Το 0 σε μορφή χαρακτήρα είναι το νούμερο 48, το 1 είναι το 49 κλπ)

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

(Εκτυπώνει την μεταβλητή `command` που διάβασε)

### Ενότητα 5.3.4 Εντολές μικροελεγκτή

Έχοντας ολοκληρώσει την επικοινωνία μεταξύ του υπολογιστή και του μικροελεγκτή, πρέπει να μεταφράσει ο μικροελεγκτής την εντολή που πήρε σε κάποια ενέργεια. Έτσι, μετά από δοκιμές φάνηκε πως οι 4 μονάδες από την παλμογεννήτρια αντιστοιχούν σε ένα τεμάχιο, οπότε ανάλογα με την εντολή που παίρνει από τον υπολογιστή ο μικροελεγκτής αλλάζει την τιμή του tar (target). Στην συνέχεια λειτουργεί η συνάρτηση ελέγχου του κινητήρα καθώς το real και το tar πλέον δεν είναι ίδια.

```
if (command == 0) {
    counter = 0;
    tar = 0 ;
}
else if (command == 1) tar = tar - 4; //1
else if (command == 2) tar = tar - 8; //2
else if (command == 3) tar = tar - 12; //3
else if (command == 4) tar = tar - 16; //4
else if (command == 5) tar = tar - 1;
else if (command == 6) tar = tar + 1;
(Ανάλογα με την μεταβλητή command εκτελεί μια από τις παραπάνω εντολές
)

}
}
```

## Κεφάλαιο έβδομο: Δοκιμές

Μετά την κατασκευή και συναρμολόγηση μηχανολογικού και ηλεκτρολογικού εξοπλισμού και τον προγραμματισμό τους, σειρά έχουν οι δοκιμές. Στις δοκιμές στο παρόν σύστημα θα ελεγχθεί αν οι επικοινωνίες μεταξύ των ελεγκτών δουλεύουν σωστά, αν οι συναρτήσεις φέρνουν το σωστό αποτέλεσμα και αν τα προϊόντα φτάνουν στην συσκευασία χωρίς αστοχίες. Για αρχή δοκιμάστηκε η συνάρτηση που γεμίζει με βήμα ένα την συσκευασία και ακολουθούν τα αποτελέσματα:



Εικόνα 6.1 Εισαγωγή 1 τεμαχίου (βήμα 1/3 )



Εικόνα 6.2 Εισαγωγή 1 τεμαχίου (βήμα 2/3)



Εικόνα 6.3 Εισαγωγή 1 τεμαχίου (βήμα 3/3 )

Η ταχύτητα πλήρωσης δεν ήταν μεγάλη και το σύστημα φάνηκε αρκετά αξιόπιστο χωρίς προβλήματα επικοινωνίας. Μηχανολογικά επίσης το σύστημα φάνηκε αρκετά σταθερό με αυτές τις ρυθμίσεις.

Στην συνέχεια δοκιμάστηκε το πρόγραμμα που χρησιμοποιεί βήμα δύο προϊόντων για την πλήρωση της συσκευασίας. Τα αποτελέσματα φαίνονται παρακάτω. Προβλήματα επικοινωνίας πάλι δεν εμφανίστηκαν όμως οι οδηγοί εμφάνισαν κάποια αστάθεια.



Εικόνα 6.4 Εισαγωγή 2 τεμαχίων (βήμα 1/3 )





Εικόνα 6.5 Εισαγωγή 2 τεμαχίων (βήμα 2/3 )



Εικόνα 6.6 Εισαγωγή 2 τεμαχίων (βήμα 3/3)

Τέλος, δοκιμάστηκε και το πρόγραμμα που γεμίζει τις συσκευασίες με βήμα τρία προϊόντα. Σε αυτό το πρόγραμμα εμφανίστηκαν εμπλοκές και χρειάστηκε χειροκίνητη υποβοήθηση για να γεμίσει η συσκευασία. Αυτό δείχνει πως όσο αυξάνεται το βήμα πλήρωσης, τόσο μειώνεται η αξιοπιστία, δηλαδή αυξάνεται η πιθανότητα θραύσης.



Εικόνα 6.7 Εισαγωγή 3 τεμαχίων (βήμα 1/3 )



Εικόνα 6.8 Εισαγωγή 3 τεμαχίων (βήμα 2/3 )



Εικόνα 6.9 Εισαγωγή 3 τεμαχίων (βήμα 3/3 )

Ένα ακόμα πρόβλημα που εμφανίστηκε ήταν η οδήγηση των προϊόντων από το δοχείο στον οδηγό. Η οδήγηση των προϊόντων στον οδηγό δεν γινόταν χωρίς την υποβοήθηση από τον χρήστη και για αυτό μπορεί να ευθύνονται δύο παράγοντες. Πρώτων το υλικό που χρησιμοποιήθηκε ίσως να μην είναι κατάλληλο καθώς δεν εμφανίστηκε επίσης κανένας επαγγελματικός σταθμός πλήρωσης με πλαστικό δοχείο προϊόντων. Δεύτερον, οι γεωμετρίες καθώς και ο τρόπος μεταφοράς (βαρύτητα) φαίνονται να μην είναι και οι πιο κατάλληλες καθώς άλλα συστήματα χρησιμοποιούν δονητικούς κινητήρες, βούρτσες και άλλες τέτοιες μεθόδους. Προτάσεις βελτίωσης θα γίνουν στο επόμενο κεφάλαιο.



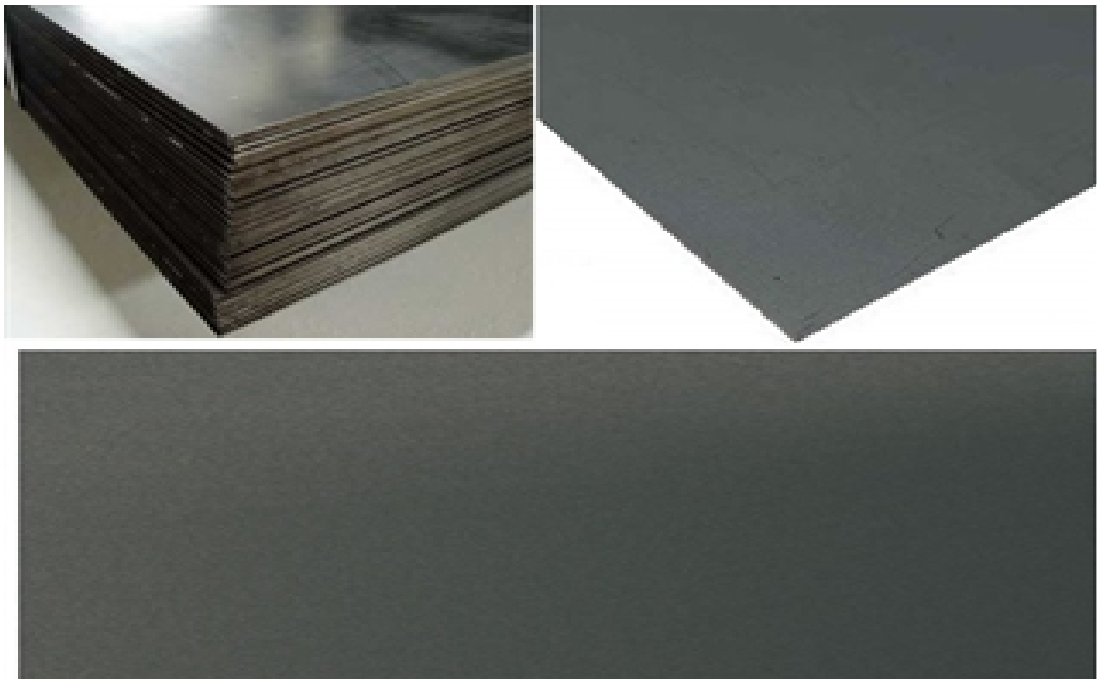
Εικόνα 6.10 Εμπλοκή οδηγών

## Κεφάλαιο όγδοο: Αξιολόγηση αποτελεσμάτων

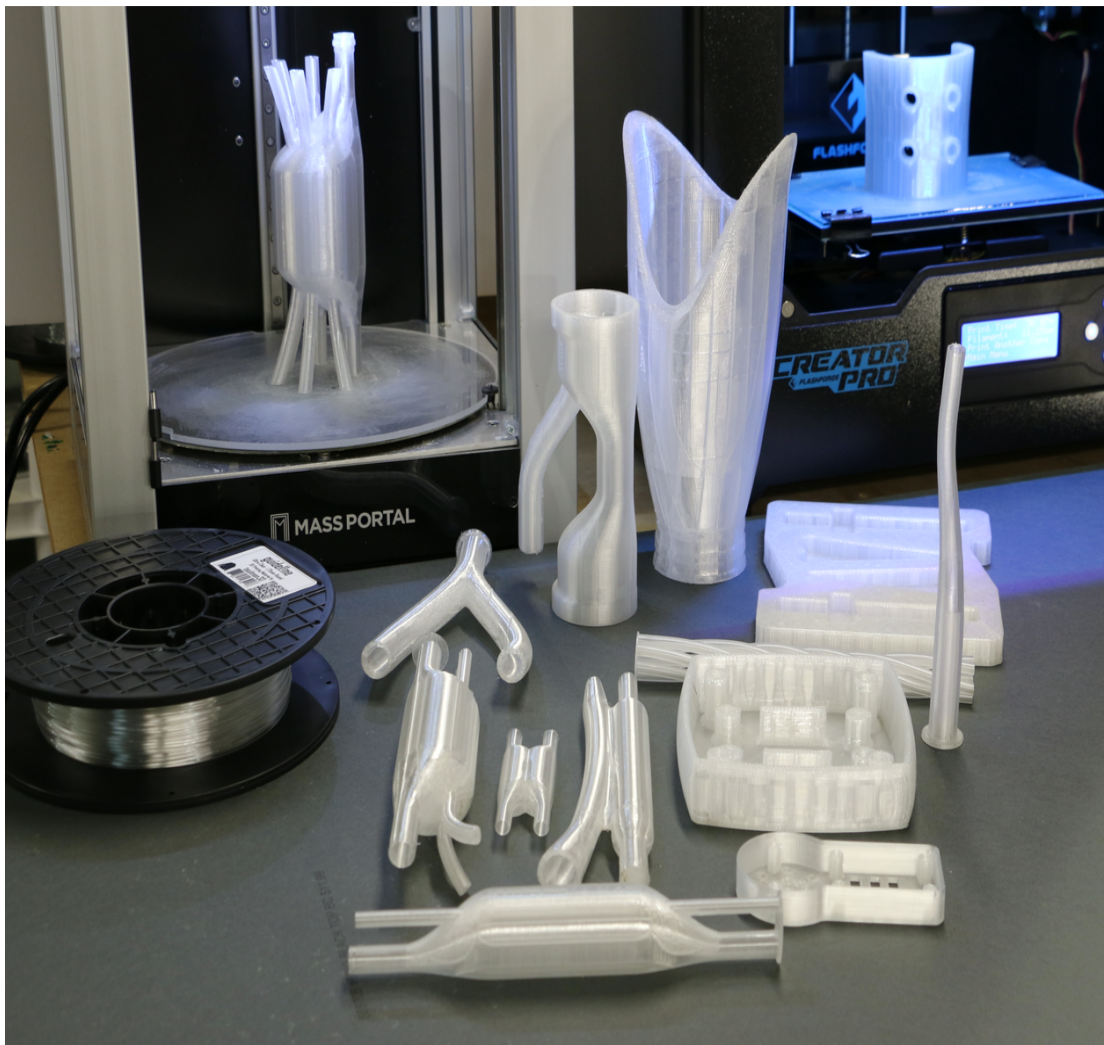
Σε αυτό το κεφάλαιο θα αξιολογηθούν τα αποτελέσματα, θα σχολιαστούν οι πιθανές αιτίες που επηρέασαν καθώς και θα προταθούν καλύτεροι μέθοδοι προσέγγισης παρόμοιων προβλημάτων.

Ξεκινώντας από τα αποτελέσματα, μιλώντας για τον έλεγχο του συστήματος τα αποτελέσματα ήταν πάρα πολύ θετικά. Η επικοινωνία δεν εμφάνισε κάποια δυσλειτουργία και ο χρόνος απόκρισης δεν ήταν μεγάλος. Ενδεικτικά πρέπει να αναφερθεί πως το θέμα της επικοινωνίας είναι αρκετά σύνθετο καθώς μέσα σε έναν ηλεκτρολογικό πίνακα μιας γραμμής συσκευασίας τα μαγνητικά πεδία είναι αρκετά ισχυρά και επηρεάζουν αρκετά τα σήματα. Παρόλο που ο τρόπος επικοινωνίας που χρησιμοποιεί το παρόν σύστημα είναι πολύ απλός, σκόπιμα φυσικά για να μην υπάρχουν δυσλειτουργίες, Κάποιος “θόρυβος” μέσα σε έναν πίνακα θα μπορούσε να το αχριστεύσει.

Περνώντας στην επιλογή των υλικών, η παρούσα μακέτα κατασκευάστηκε για πειραματικούς σκοπούς και τα υλικά φυσικά δεν ήταν ούτε κατάλληλα για τρόφιμα, ούτε τόσο ανθεκτικά όσο θα έπρεπε για να υπάρχει ένα τέλειο αποτέλεσμα. Τα υλικά που χρησιμοποιήθηκαν αναφορικά ήταν μαλακός χάλυβας και πλαστικό PETG.

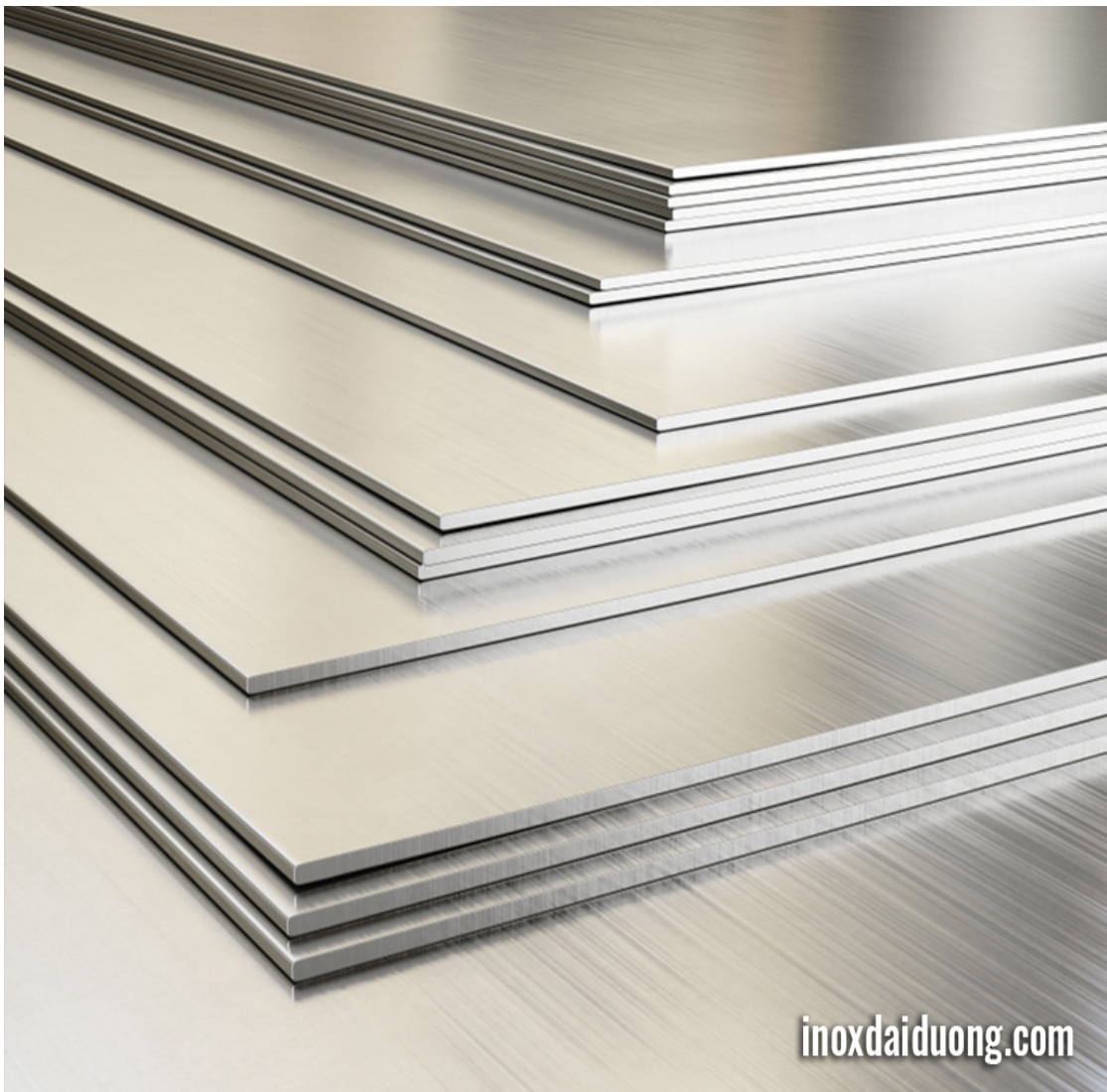


Εικόνα 8.1 Μαλακός χάλυβας



Εικόνα 8.2 PETG 3d print filament

Φυσικά και αυτά τα υλικά δεν βρίσκουν πολλές εφαρμογές στις βιομηχανίες τροφίμων, όχι τουλάχιστον σε μηχανές που έχουν άμεση επαφή με τρόφιμα. Σε άμεση επαφή με τρόφιμα έρχονται υλικά τα οποία μπορούν να πλένονται συχνά χωρίς να διαβρώνονται, τα οποία έχουν μικρή τραχύτητα και άρα δεν κρατάνε βρωμιά πάνω τους και φυσικά υλικά που δεν έχουν πολλές αντιδράσεις με χημικά. Ένα από τα πιο συνηθισμένα υλικά που θα δει κανείς σε μια γραμμή συσκευασίας είναι ο ανοξείδωτος αστάλι, εκτός από ανοξείδωτο χάλυβα συχνά βλέπουμε και αλουμίνιο. Σε πιο σπάνιες περιπτώσεις κάποια εξαρτήματα είναι φτιαγμένα και από πλαστικό.



Εικόνα 8.3 Φύλλα λαμαρίνα από ανοξείδωτο αστάλι





Εικόνα 8.4 Φύλλο αλουμινίου



Εικόνα 8.5 Πλαστικό δοχείο κατάλληλο για τρόφιμα

Εκτός από τα υλικά, ένα ακόμα σημείο που θα μπορούσε να βελτιωθεί η παρούσα εργασία θα ήταν το περιβάλλον του χρήστη. Η χρήση της βιβλιοθήκης tkinter / python έχει αρκετά θετικά όμως καθιστά αρκετά δύσκολο τον έλεγχο της μηχανής, ή ακόμα και την μεταφορά δεδομένων από την μηχανή σε κινητό τηλέφωνο ή tablet. Η ανάπτυξη μιας εφαρμογής σε android studio ακόμα και για σκοπούς οπτικοποίησης δεδομένων θα ήταν μια αρκετά σημαντική συνεισφορά στα προβλήματα της βιομηχανίας.



Εικόνα 8.6 Εφαρμογή σε φορητή συσκευή για οπτικοποίηση δεδομένων

## Παράρτημα Α: Κώδικας Arduino

```
#define outputA 7
#define outputB 6
#define motor 9
#define dir 8

int counter = 0;
int aState;
int aLastState;
int pos = 0;
int tar = 0;
int start_cw = 0;
int start_ccw = 0;
int stp = 0;

void setup() {

    // put your setup code here, to run once:
    Serial.begin(9600);
    pinMode(outputA, INPUT);
    pinMode(outputB, INPUT);
    pinMode(motor, OUTPUT);
    pinMode(dir, OUTPUT);
    aLastState = digitalRead(outputA);
    digitalWrite(motor, HIGH);

}

void loop() {

    //GET_POS
    aState = digitalRead(outputA);
    if(aState != aLastState){
        if(digitalRead(outputB) != aState){
            counter ++;
            //Serial.print("counter: ");
            //Serial.println(counter);
        }
    }
}
```

```

}
else{
    counter --;
    // Serial.print("counter: ");
    // Serial.println(counter);
}
}
aLastState = aState;

//CHOOSE

if( counter < tar -1){
    if (start_cw == 0 ){
        start_cw = 1;
        start_ccw =0;
        stp = 0;
        digitalWrite(motor, HIGH);
        digitalWrite(dir, HIGH);
        digitalWrite(motor, LOW);
    }
}
else if( counter > tar+1){
    if (start_ccw == 0){
        start_cw =0;
        start_ccw =1;
        stp = 0;
        digitalWrite(motor, HIGH);
        digitalWrite(dir, LOW);
        digitalWrite(motor, LOW);
    }
}
else if (counter < tar+1 and counter > tar -1) {
    if(stp == 0 ) {
        start_cw = 0;
        start_ccw = 0;
        stp = 1;

        digitalWrite(motor, HIGH);
    }
}

```

```

}

// += -5   -= -3

//Serial get target
if (Serial.available() > 0) {
  int command = Serial.read() - 48;
  Serial.println(command);

  if (command == 0) {
    counter = 0;
    tar = 0 ;
  }
  else if (command == 1) tar = tar - 4; //1
  else if (command == 2) tar = tar - 8; //2 TEMAXIA
  else if (command == 3) tar = tar - 12; //3 TEMAXIA
  else if (command == 4) tar = tar - 16;
  else if (command == 5) tar = tar - 1;
  else if (command == 6) tar = tar + 1;
  else if (command == 7) tar = tar - 140;
  else if (command == 8) tar = tar - 160;

}
}

```

## Παράρτημα Β: Κώδικας Python

```
import serial
import time
from tkinter import *

serialcomm = serial.Serial('/dev/ttyUSB0', 9600)
serialcomm.timeout = 1

root=Tk();
root.title('Msc in Robotics - Smarties feeder User Interface')
root.geometry('600x400')

def myclick1():
    label = Label(frame1, text = "Drop " + str(e.get()) + " Smarties")
    label.grid(row=2, column =0)
    command = e.get()
    serialcomm.write(command.encode())

def myclick2():
    command = "1"
    serialcomm.write(command.encode())
    time.sleep(0.5)

def myclick3():
    command = "2"
    serialcomm.write(command.encode())
    time.sleep(0.1)

def myclick4():
    command = "3"
    serialcomm.write(command.encode())
```

```

time.sleep(0.1)

def myclick5():
    command = "5"
    serialcomm.write(command.encode())
    time.sleep(0.1)

def myclick6():
    command = "6"
    serialcomm.write(command.encode())
    time.sleep(0.1)

def myclick7():
    command = "7"
    serialcomm.write(command.encode())
    time.sleep(0.1)

def myclick8():
    command = "8"
    serialcomm.write(command.encode())
    time.sleep(0.1)

def myclick9():
    command = "9"
    serialcomm.write(command.encode())
    time.sleep(0.1)

def myclick0():
    command = "0"
    serialcomm.write(command.encode())
    time.sleep(0.1)

frame1 = LabelFrame(root, text="Automatic mode")
frame1.grid(row=0, column=0, sticky=N)

frame2 = LabelFrame(root, text="Setup ")
frame2.grid(row=0, column=1, sticky=N)

frame3 = LabelFrame(root, text="Maintenance page")

```

```

frame3.grid(row=0, column=2, sticky=N)
e = Entry(frame1)
e.insert(0, "Enter smarties quantity")

myButton1 = Button(frame1, text="trigger", padx =50, pady = 20,
command=myclick1)

myButton2 = Button(frame2, text="Program 1 \n X1 pieces", padx =50, pady
= 20, command=myclick2)
myButton3 = Button(frame2, text="Program 2 \n x2 pieces", padx =50, pady =
20, command=myclick3)
myButton4 = Button(frame2, text="Program 3 \n X3 pieces", padx =50, pady
= 20, command=myclick4)

myButton5 = Button(frame3, text="Manual forward", padx =47, pady = 27,
command=myclick5)
myButton6 = Button(frame3, text="Manual backward", padx =40.5, pady =
27, command=myclick6)
myButton8 = Button(frame3, text="Set encoder to 0", padx =42, pady = 27,
command=myclick0)
#myButton9 = Button(root, text="Command 20 backward", padx =50, pady =
20, command=myclick9)
#myButton0 = Button(root, text="Set encoder to 0", padx =50, pady = 20,
command=myclick0)

e.grid(row=0, column=0)
myButton1.grid(row=1, column=0, sticky=W)

myButton2.grid(row=0, column=0, sticky=W)
myButton3.grid(row=1, column=0, sticky=W)
myButton4.grid(row=2, column=0, sticky=W)

myButton5.grid(row=0, column=0, sticky=W)
myButton6.grid(row=1, column=0, sticky=W)
myButton8.grid(row=3, column=0, sticky=W)
root.mainloop()

```



## Παράρτημα Γ: Πίνακας σημάτων

Χαρακτήρες Raspberry Pi	Λειτουργίες Arduino
0	Μηδενισμός μετρητών
1	1 τεμάχιο
2	2 τεμάχια
3	3 τεμάχια
4	4 τεμάχια
5	Μικρό βήμα μπροστά
6	Μικρό βήμα πίσω

## Παράρτημα Δ: Κατάλογος εικόνων

Εικόνα 2.1	Chute alu-alu Blister Feeder
Εικόνα 2.2	Tube blister feeder
Εικόνα 2.3	Tube blister feeder
Εικόνα 2.4	Blister machine DPH260
Εικόνα 2.5	Blister machine DPH260
Εικόνα 2.6	Blister machine B200
Εικόνα 2.7	Blister machine B1240
Εικόνα 2.8	Vibration feeder
Εικόνα 2.9	Arduino controlled part feeding system
Εικόνα 2.10	Λύση από την Industrial shields στο πρόβλημα τροφής - νερού σε μονάδα ζώων
Εικόνα 2.11	Σύστημα πλήρωσης τροφής με χρήση arduino και δυναμοκυψέλης
Εικόνα 2.12	Pet feeder by Arduino.cc
Εικόνα 2.13	tkinter UI
Εικόνα 3.1	Σχεδιαστικό περιβάλλον Solidworks
Εικόνα 3.2	CNC - Laser
Εικόνα 3.3	Τρισδιάστατος εκτυπωτής
Εικόνα 3.4	Στράντζα
Εικόνα 3.5	Τόρνος
Εικόνα 3.6	Ηλεκτροκόλληση
Εικόνα 3.7	Ηλεκτροκινητήρας συνεχούς ρεύματος με μειωτήρα στην έξοδο

Εικόνα 3.8	Παλμογεννήτρια
Εικόνα 3.9	Ρελέ
Εικόνα 3.10	Arduino Nano
Εικόνα 3.11	Arduino IDE
Εικόνα 3.12	Raspberry Pi Model 3B+
Εικόνα 3.13	Γρανάζια για ιμάντα χρονισμού
Εικόνα 3.14	Ζεύγος γραναζιών ιμάντα
Εικόνα 3.15	Τροφοδοτικό υπολογιστή
Εικόνα 3.16	Σειριακή επικοινωνία
Εικόνα 4.1	Τρισδιάστατη απεικόνιση συστήματος
Εικόνα 4.2	Σασί συστήματος (εξάρτημα 1/5)
Εικόνα 4.3	Σασί συστήματος (εξάρτημα 2/5)
Εικόνα 4.4	Σασί συστήματος (εξάρτημα 3/5)
Εικόνα 4.5	Σασί συστήματος (εξάρτημα 4/5)
Εικόνα 4.6	Σασί συστήματος (εξάρτημα 5/5)
Εικόνα 4.7	Εξαρτήματα σύνδεση αξόνων με τον κύλινδρο
Εικόνα 4.8	Κύλινδρος
Εικόνα 4.9	Οδηγός (εξάρτημα 1/2)
Εικόνα 4.10	Οδηγός (εξάρτημα 2/2)
Εικόνα 4.11	Δοχείο προϊόντων
Εικόνα 4.12	Βάση οδηγών
Εικόνα 4.13	Καπάκι οδηγών
Εικόνα 6.1	Εισαγωγή 1 τεμαχίου (βήμα 1/3 )

Εικόνα 6.2	Εισαγωγή 1 τεμαχίου (βήμα 2/3 )
Εικόνα 6.3	Εισαγωγή 1 τεμαχίου (βήμα 3/3 )
Εικόνα 6.4	Εισαγωγή 2 τεμαχίων (βήμα 1/3 )
Εικόνα 6.5	Εισαγωγή 2 τεμαχίων (βήμα 2/3 )
Εικόνα 6.6	Εισαγωγή 2 τεμαχίων (βήμα 3/3 )
Εικόνα 6.7	Εισαγωγή 3 τεμαχίων (βήμα 1/3 )
Εικόνα 6.8	Εισαγωγή 3 τεμαχίων (βήμα 2/3 )
Εικόνα 6.9	Εισαγωγή 3 τεμαχίων (βήμα 3/3 )
Εικόνα 6.10	Εμπλοκή οδηγών
Εικόνα 8.1	Μαλακός χάλυβας
Εικόνα 8.2	PETG 3d print filament
Εικόνα 8.3	Φύλλα λαμαρίνα από ανοξείδωτο ατσάλι
Εικόνα 8.4	Φύλλο αλουμινίου
Εικόνα 8.5	Πλαστικό δοχείο κατάλληλο για τρόφιμα
Εικόνα 8.6	Εφαρμογή σε φορητή συσκευή για οπτικοποίηση δεδομένων

## Παράρτημα Δ: Κατάλογος Σχημάτων

Σχήμα 4.1	Ηλεκτρολογικό σχέδιο Arduino
Σχήμα 4.2	Ηλεκτρολογικό σχέδιο ρυθμιστή ταχύτητας
Σχήμα 4.3	Ηλεκτρολογικό σχέδιο ελέγχου κινητήρα
Σχήμα 5.1	Διαδρομή προϊόντος
Σχήμα 5.2	Εντολή χρήστη
Σχήμα 5.3	Έλεγχος χαμηλού επιπέδου από το Arduino

## Βιβλιογραφία

- [1] Ανάπτυξη εφαρμογών με το Arduino, Παπάζογλου Παναγιώτης Μ. Εκδόσεις Τζιόλα,
- [2] Εξερεύνηση της Python, Καφές Μάνος, Εκδόσεις Κλειδάριθμος
- [3] Δημιουργία 3D μοντέλων με το Solidworks, Κάππος Γιάννης, Εκδόσεις Κλειδάριθμος
- [4] Βασικές αρχές συστημάτων CAD/CAM/CAE . Συγγραφέας: Kunwoo Lee Μεταφραστής: Κώστας Καρανικολός Επιμέλεια: Μιχάλης Μικέδης
- [5] Διοίκηση Παραγωγής και Επιχειρησιακών Λειτουργιών . Σχεδιασμός , Προγραμματισμός και Έλεγχος σε Συστήματα Παραγωγής και Υπηρεσιών, Ξανθόπουλος Αλέξανδρος Κουλουριώτης Δημήτρης, Εκδόσεις Τζιόλα
- [6] Μαθαίνω Python & Tkinter, Κωνσταντίνος Παπαστεργίου, Ιδιωτική Έκδοση <https://www.ebooks4greeks.gr/mathainw-python-tkinter>
- [7] <https://www.jornen.com/> Ιστοσελίδα κατασκευάστριας εταιρίας γραμμών συσκευασίας
- [8] <https://www.hulmann.de/> Ιστοσελίδα κατασκευάστριας εταιρίας γραμμών συσκευασίας
- [9] <https://www.youtube.com/watch?v=L47OBXGYRtE>, Ερασιτέχνης κατασκευαστής συστήματος πλήρωσης με Arduino
- [10] <https://www.industrialshields.com/case-study-animal-feeding-automation-industrial-plc-based-arduino> Εταιρία από την Ισπανία που παρέχει εξατομικευμένες ηλεκτρονικές λύσεις βασισμένες σε Open Source συστήματα.
- [11] <https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1155&context=braesp>  
Διπλωματική εργασία φοιτητή από το Πολυτεχνείο της Καλιφόρνια.
- [12] <https://create.arduino.cc/projecthub/edr1924/easy-to-build-pet-feeder-phone-app-control-soon-86c1ef>  
Pet feeder by arduino.cc
- [13] <https://maker.pro/raspberry-pi/tutorial/how-to-use-tkinter-to-control-raspberry-pi-gpio-pins>  
tkiner UI