



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΤΜΗΜΑ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ**

**Δημιουργία Συστήματος Επίβλεψης και Αναγνώρισης
Ανωμαλιών σε Φάρμα Τρισδιάστατων Εκτυπωτών**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Σκιαδάς Λυκούργος
Αριθμός Μητρώου : 71445181**

Επιβλέπων : Γρηγόριος Νικολάου, Καθηγητής ΠΑΔΑ

Αθήνα, Ιούνιος 2021

Μέλη εξεταστικής επιτροπής

- 1. Νικολάου Γρηγόριος**
- 2. Βασιλειάδου Σουλτάνα**
- 3. Δρόσος Χρήστος**

| | |
|--|----------|
| Ευχαριστίες | 5 |
| Περίληψη | 6 |
| Abstract | 7 |
| Κεφάλαιο 1 | |
| Εισαγωγή | 8 |
| 1.1 Διαδίκτυο των πραγμάτων | 10 |
| 1.2 Ταχεία Πρωτοτυποποίηση | 11 |
| 1.2.1 Αφαιρετική μέθοδος | 12 |
| 1.2.2 Προσθετική μέθοδος | 12 |
| 1.3 Ιστορική αναδρομή της 3D Εκτύπωσης | 13 |
| 1.3.1 Η Γέννηση | 14 |
| 1.3.2 Η εξέλιξη | 16 |
| 1.3.1 Η Ωρίμανση | 17 |
| 1.4 Η Ιδέα | 18 |
| Κεφάλαιο 2 | |
| Τεχνολογίες Τρισδιάστατων Εκτυπωτών | 19 |
| 2.1 Laser SLA - Stereolithography | 19 |
| 2.1.1 SLA-DLP - Digital Light Processing | 20 |
| 2.2 FDM-Fusion Deposition Modeling | 22 |
| 2.3 SLS - Selective Laser Sintering | 24 |
| Κεφάλαιο 3 | |
| Η Δομή ενός FDM 3D Printer | 26 |
| 3.1 Άξονες Κίνησης | 27 |
| 3.1.1 Γραμμικοί Ράβδοι | 27 |
| 3.1.2 Γραμμικές Ράγες | 28 |
| 3.1.3 V-Wheels | 29 |
| 3.2 Βηματικοί κινητήρες | 31 |
| 3.2.1 Extruders | 34 |
| 3.3 Θερμαντικά Στοιχεία | 36 |
| 3.3.1 HotEnd | 36 |
| 3.3.2 Heated Bed | 38 |
| 3.4 Mainboard | 40 |
| 3.4.1 Marlin | 40 |
| 3.4.2 G-Code | 41 |
| 3.4.3 3D Model Processing | 42 |
| Κεφάλαιο 4 | |
| Ανάλυση απαιτήσεων εφαρμογής | 44 |
| 4.1 Απαιτήσεις Σχεδιασμού | 44 |
| 4.2 Απαιτήσεις Αισθητηρίων | 45 |
| 4.3 Απαιτήσεις Δεδομένων Εκμάθησης | 46 |
| 4.4 Απαιτήσεις Διασύνδεσης και Διεπαφών | 47 |
| Κεφάλαιο 5 | |

| | |
|--|------------|
| Συλλογή δεδομένων Μέσω Μικροελεγκτή | 48 |
| 5.1 Arduino | 48 |
| 5.2 Accelerometer | 50 |
| 5.2.1 Στερέωση Επιταχυνσιόμετρου | 51 |
| 5.2.2 Συνδεσμολογία/ Κώδικας Επιταχυνσιόμετρου | 53 |
| 5.3 Encoder | 55 |
| 5.3.1 Στερέωση Encoder | 56 |
| 5.3.2 Συνδεσμολογία/ Κώδικας Encoder | 59 |
| 5.4 Συγχώνευση | 63 |
| Κεφάλαιο 6 | |
| Μηχανική Μάθηση - Anomaly Detection | 66 |
| 6.1 Autoencoder | 67 |
| 6.2 Datasets | 69 |
| 6.2.1 Συλλογή Δεδομένων | 69 |
| 6.2.2 Ανάλυση Χαρακτηριστικών Dataset | 71 |
| 6.3 Autoencoder Training | 74 |
| 6.3.1 Εκπαίδευση του Μοντέλου | 75 |
| 6.3.2 Μετατροπή του Μοντέλου | 82 |
| Κεφάλαιο 7 | |
| Διακομιστής Συγκέντρωσης και Απεικόνισης Δεδομένων | 83 |
| 7.1 Hardware Διακομιστή | 83 |
| 7.2 Παραμετροποίηση Διακομιστή | 84 |
| 7.3 Σχεδίαση Ιστοσελίδας | 87 |
| 7.3.1 Κώδικας Διαχείρισης Βάσης Δεδομένων | 87 |
| 7.3.2 Κώδικας Ιστοσελίδας | 90 |
| 7.4 Συλλογή δεδομένων | 100 |
| Κεφάλαιο 8 | |
| Ολοκλήρωση Συστήματος | 103 |
| 8.1 Κώδικας Συγχώνευσης | 105 |
| 8.2 Επεξήγηση λειτουργίας | 110 |
| 8.3 Μελλοντικές Βελτιώσεις | 111 |
| Βιβλιογραφικές Αναφορές | 113 |

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ/ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/η κάτωθι υπογεγραμμένος/η Λυκούργος Σκιαδάς του Δημητρίου, με αριθμό μητρώου 71445181 φοιτητής/τρια του Πανεπιστημίου Δυτικής Αττικής της Σχολής..... του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής., δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της πτυχιακής/διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο/Η Δηλών/ούσα



Ευχαριστίες

Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας, θα ήθελα να ευχαριστήσω θερμά τον επιβλέπων καθηγητή μου, Κ. Γρηγόριο Νικολάου, για την επιστημονική του καθοδήγηση, τις υποδείξεις του, την επιμονή του, το αμείωτο ενδιαφέρον του, τη συμπαράστασή του και τη συνεχή του υποστήριξη. Όπως θα ήθελα να ευχαριστήσω όλους του καθηγητές του τμήματος Βιομηχανικής Σχεδίασης και Παραγωγής του Πανεπιστημίου Δυτικής Αττικής, οι οποίοι με το εκπαιδευτικό τους πνεύμα με βοήθησαν να κάνω τα πρώτα μου βήματα σαν μηχανικός.

Επίσης, θα ήθελα να ευχαριστήσω όλους τους συμφοιτητές που με έχουν στηριξει αυτά τα χρόνια. Συγκεκριμένα, θα ήθελα να ευχαριστήσω την συνάδελφό Ελένη Δέσποινα Καμάρα, για ανταλλαγή απόψεων και την συμβολή της στην υλοποίηση της παρούσας εργασίας.

Τέλος θα ήθελα να ευχαριστήσω την οικογένειά μου, για την συνεχή συμπαράσταση τους σε όλα τα χρόνια της ζωής μου.

Περίληψη

Από τα πρώτα της βήματα, η τρισδιάστατη εκτύπωση(3D Printing) αποτέλεσε σημαντικό εύρημα για τον τεχνολογικό κόσμο, αποτελεί μια τεχνολογία οποία έχει δείξει σημαντική εξέλιξη τα τελευταία 30 χρόνια, όπως και οι μέθοδοι, αυτής της τεχνολογίας, που αξιοποιούνται ώστε να παραχθεί ένα τελικό αποτέλεσμα ή προϊόν. Μια από τις πιο γνωστές μεθόδους τρισδιάστατης εκτύπωσης είναι η μέθοδος εναπόθεσης τήγματος(FDM-Fused Deposition Modeling), η οποία ειδικά την τελευταία δεκαετία, έπειτα από την λήξη της πατέντας υπ' αριθμόν US5340433A¹, έχει γίνει ευρέως διαδεδομένη και εμπορικά διαθέσιμη με πολύ χαμηλό κόστος. Αυτή η μέθοδος χρησιμεύει κυρίως στην γρήγορη και οικονομική παραγωγή πρωτοτύπων και κατασκευών δίχως τον κίνδυνο υψηλού κόστους σχεδίασης, όπως για παράδειγμα στην δημιουργία καλουπιού που χρησιμοποιείται στην χύτευση με έγχυση(Injection Molding). Σε συνδυασμό με τα υλικά που μπορούν να χρησιμοποιηθούν με αυτήν την μέθοδο, της τρισδιάστατης εκτύπωσης, υπάρχει μια πληθώρα εφαρμογών όπου αυτή μπορεί να φανεί χρήσιμη.

Σε αυτή την διπλωματική εργασία, θα αναφερθούμε στην μαζική παραγωγή πρωτότυπων αντικειμένων από μια συστοιχία τρισδιάστατων εκτυπωτών τύπου FDM(Φαρμα Εκτυπωτών) και θα πραγματοποιήσουμε μια ολοκληρωμένη μελέτη ως προς τα συνηθέστερα σφάλματα τα οποία μπορούν να προκύψουν κατά την παραγωγική διαδικασία, πως μπορούμε να τα εντοπίσουμε, δίχως την άμεση παρακολούθηση, και πώς μπορούμε να παρακολουθήσουμε γενικές πληροφορίες μέσω μιας δικτυακής διεπαφής όπου και θα φανερώνονται στοιχεία για το κάθε μηχάνημα που βρίσκεται εντός αυτής. Αξιοποιώντας μεθόδους μηχανικής μάθησης θα μπορούμε έγκαιρα να εντοπίζουμε ανωμαλίες του συστήματος οι οποίες θα μπορούσαν να αυξήσουν το κόστος παραγωγής ή και να προκαλέσουν μόνιμες βλάβες σε κάποιο από τα μηχανήματα.

Τα αποτελέσματα αυτής την έρευνας τα οποία και θα δούμε αναλυτικότερα στα τελευταία κεφάλαια, μπορούν να μας υποδείξουν για το πόσο σημαντικός είναι ο έλεγχος και η επίβλεψη των μηχανημάτων παραγωγής στα πλαίσια μιας παραγωγικής μονάδας και πως η αυτοματοποίηση της διαδικασίας επίβλεψης αποτελεί σημαντικό παράγοντα για την εύκολη λειτουργία, πρόβλεψη και αντιμετώπιση προβλημάτων της συστοιχίας, τα οποία, εν συνεχεία οδηγούν και σε καλύτερα αποτελέσματα από πλευράς παραγόμενων προϊόντων.

¹ <https://patents.google.com/patent/US5340433/en11>

Abstract

From the first steps 3D Printing technology, is considered an important find in the technological world, a technology that has shown significant development over the last 30 years, as well as the methods used to produce a final result or product. One of the most well-known 3D Printing methods is the FDM-Fused Deposition Modeling method, which especially in the last decade, after the expiration of the patent number US5340433A, has become widespread and commercially available at a very low cost. This method is mainly used for the rapid and economical production of prototypes and assemblies without the risk of high cost, such as the injection molding method. Taking into consideration the materials that can be used, with this 3D Printing method, there is a big variety of use cases.

In this thesis, we will be referring to the mass production of prototype objects from an array of FDM 3D Printers (3D Printing Farm) and will carry out a comprehensive study of the most common errors that may occur during the production process, how can we detect those errors, without direct monitoring and how can we monitor general information through a web interface where data will be displayed for each machine that consists our array. Utilizing machine learning methods we will be able to detect system malfunctions in time which could increase production cost or cause permanent damage to any of the machines

The results of this research, which we will see in more detail in the last chapters, can show us how important it is to control and supervise production machines within a production unit and that the automation of the supervision process is an important factor for easy operation, forecasting and troubleshooting, which, in turn, leads to better results in the production process.

Κεφάλαιο 1

Εισαγωγή

Ακολουθώντας την ανθρώπινη ιστορία και την εξέλιξη της τεχνολογίας, είναι άξιο να σημειώσουμε πως η καινοτομία και η εφευρετικότητα αποτελεί ένα από τα σημαντικότερα χαρακτηριστικά μας. Από την δημιουργία απλών πέτρινων εργαλείων², της κατώτερης παλαιολιθικής εποχής, έως και πολύπλοκα μηχανικά συστήματα τα οποία έχουν την δυνατότητα “αίσθησης” και “απόφασης”, το ανθρώπινο γένος έχει αποδείξει πως πάντα καταφέρνει να βρίσκει τρόπους ενίσχυσης των περιορισμένων φυσικών δυνατοτήτων του με αποτέλεσμα να καταφέρνει να πετύχει αυτό που επιθυμεί με λιγότερο κόπο.

Σημαντική περίοδος, η οποία αποτελεί την γενέτειρα του τεχνολογικού κόσμου που γνωρίζουμε σήμερα είναι η 1η βιομηχανική επανάσταση³ η οποία προέκυψε τα τέλη του 18ου αιώνα(1765). Η Βιομηχανίες της εποχής άλλαξαν, χάρις την χρήση μηχανών για την λειτουργία της παραγωγικής διαδικασίας. Η εφεύρεση και η χρήση των ατμομηχανών οδήγησε στην ραγδαία ανάπτυξη του σιδηροδρομικού δικτύου με αποτέλεσμα να μπορούν να δημιουργηθούν μεγαλύτερα οικονομικά δίκτυα, όπου η πρόσβαση σε αυτά κατέστη γρηγορότερη. Αυτού του είδους η εξέλιξη ήταν ο λόγος όπου εθνική οικονομία της κάθε χώρας μετατοπίστηκε από την καλλιέργεια στην μηχανοποίηση.

Σχεδόν έναν αιώνα αργότερα, η 1η βιομηχανική επανάσταση θα αποτελέσει σημαντικό θεμέλιο για την 2η βιομηχανική επανάσταση⁴, η οποία ξεκίνησε στα τέλη του 19ου αιώνα(1870), στην οποία παρατηρούμε σημαντικές τεχνολογικές εξελίξεις όπως η αξιοποίηση του ηλεκτρισμού, την εγκατάσταση τηλεφωνικών στύλων κατα μήκος των σιδηροδρομικών γραμμών ώστε να μπορεί να υπάρχει γρηγορότερη επικοινωνία και καλύτερη οργάνωση μεταξύ των οικονομικών δικτύων και η χρήση μηχανών εσωτερικής καύσης που έδωσαν ζωή στα πρώτα μηχανοκίνητα οχήματα προσωπικής χρήσης. Πέραν των τεχνολογικών εξελίξεων που ακολούθησαν εκείνη την περίοδο, σημαντικό είναι να αναφερθεί πως σημαντικές εξελίξεις ακολούθησαν και στο κομμάτι της διαχείρισης των εργαζομένων αλλά και της οργάνωσης των παραγωγικών φάσεων, ώστε αυτές να είναι ταχύτερες.



Εικόνα 1.0.3
Γραμμή παραγωγής της εποχής



Εικόνα 1.0.3
Γραμμή παραγωγής της Ford

²<https://www.smithsonianmag.com/science-nature/oldest-known-stone-tools-unearthed-kenya-180955341/>

³ <https://www.history.com/topics/industrial-revolution/industrial-revolution>

⁴ <https://medium.com/@elmiljan/the-second-industrial-revolution-6c0142e103cd>

Στα μέσα του 20ου αιώνα⁵, ξεκινάει η 3η βιομηχανική επανάσταση, γνωστή και ως ψηφιακή επανάσταση, οι τεχνολογικές εξελίξεις που αναπτύχθηκαν κατά αυτή την περίοδο, έφεραν την ανθρωπότητα όλο και πιο κοντά στην σημερινή της καθημερινότητα. Αυτή ξεκίνησε από την εφεύρεση του τρανζίστορ⁶, το οποίο και επέκτεινε περαιτέρω την εξέλιξη των ηλεκτρονικών υπολογιστικών συστημάτων που γνωρίζουμε σήμερα. Η σημαντική ανάπτυξη της επιστήμης των ηλεκτρονικών, επιτάχυναν ολο ένα και περισσότερο τις τεχνολογίες της τηλεφωνίας, της απομακρυσμένης διασύνδεσης ηλεκτρονικών υπολογιστών(Internet), της ρομποτικής και την εξέλιξη προγραμματιζόμενων τεχνολογιών ελέγχου, όπως τα PLC και τους μικροελεγκτές. Αυτές οι καινοτομίες και οι εξελίξεις τους για τα επόμενα χρόνια, οδήγησαν τις βιομηχανικές μονάδες, να επιτύχουν έναν πολύ υψηλό βαθμό αυτοματοποίησης και ελέγχου της παραγωγικής διαδικασίας, με αποτέλεσμα, για άλλη μια φορά η ταχύτητα και η ποιότητα παραγωγής να βελτιωθούν ραγδαία.

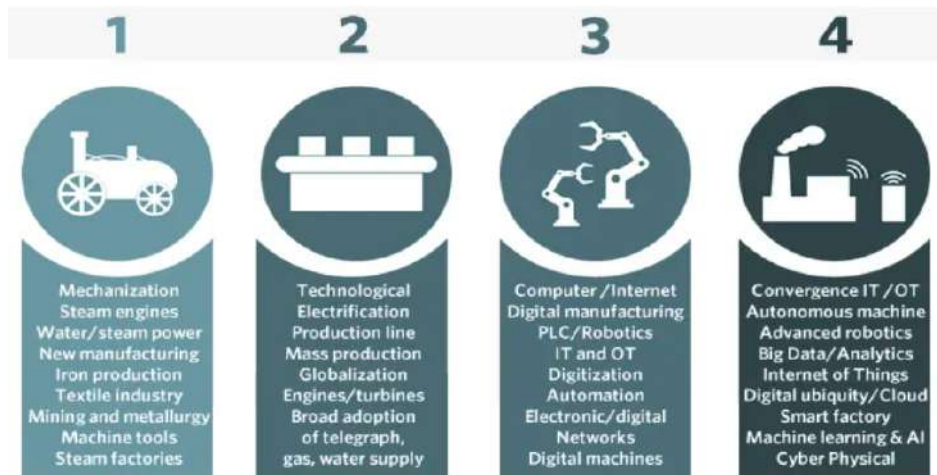
Πλησιάζοντας ολοένα και περισσότερο την σήμερα ημέρα, μπορούμε να παρατηρήσουμε πως διανύουμε μια ακόμα βιομηχανική επανάσταση η οποία, αξιοποιεί τις υποδομές των προηγούμενων ώστε να υπάρξει ακόμα υψηλότερο επίπεδο αυτοματοποίησης και ελέγχου τόσο στην απλή καθημερινότητα όσο και στον κόσμο της βιομηχανίας. Η συγκεκριμένη επανάσταση, ή αλλιώς Βιομηχανική επανάσταση 4.0⁷, έχει σαν κύριες εξελίξεις την διασύνδεση και επικοινωνία συσκευών μέσω του διαδικτύου(Internet of Things-IoT) και την ανάπτυξη συστημάτων τεχνητής νοημοσύνης. Η 4η Βιομηχανική επανάσταση στηρίζεται έντονα στα δεδομένα(Big Data), τόσο στη συλλογή τους όσο και στην ανάλυση αυτών. Ένα απλό παράδειγμα, αξιοποίησης αυτών των δεδομένων είναι ανάπτυξη και εκπαίδευση συστημάτων που φέρουν κάποιου είδους τεχνητή νοημοσύνη με σκοπό την καλύτερη διαχείριση αποφάσεων. Η τεχνητή νοημοσύνη βέβαια, δεν χρησιμοποιείται πλέον μόνο στην παραγωγική διαδικασία, σημαντική εξέλιξη έχει υπάρξει ακόμα και στην αξιοποίηση των δεδομένων που έχουν συλλεχθεί για την δημιουργία καταναλωτικών μοντέλων, κυρίως υπο μορφή διαφημίσεων ή προτάσεων που θα μπορούσαν να ταιριάζουν στις ανάγκες του καταναλωτή.

Παρατηρώντας, την πορεία όλων αυτών των βιομηχανικών επαναστάσεων, μπορούμε να παρατηρήσουμε πως η ανάγκη για ταχύτερη, οικονομικότερη και ασφαλέστερη παραγωγή βρίσκεται στο επίκεντρο του βιομηχανικού κόσμου. Αυτό έχει αποτελέσει έμπνευση για την ανακάλυψη τεχνολογιών και διαδικασιών για τον συγκεκριμένο σκοπό. Μια από αυτές τις τεχνολογίες είναι και η τρισδιάστατη εκτύπωση.

⁵<https://trailhead.salesforce.com/en/content/learn/modules/learn-about-the-fourth-industrial-revolution/meet-the-three-industrial-revolutions>

⁶<https://web.archive.org/web/20110813004951/http://www.ideafinder.com/history/inventions/transistor.htm>

⁷ <https://www.i-scoop.eu/industry-4-0/>



Εικόνα 1.0.3
Η ιστορία των βιομηχανικών επαναστάσεων

1.1 Διαδίκτυο των πραγμάτων

Το διαδίκτυο των πραγμάτων ή πιο διαδεδομένα Internet of Things(IoT), μπορούμε να ορίσουμε πως αναφέρεται σε μια πληθώρα συσκευών οι οποίες έχουν την δυνατότητα σύνδεσης με το διαδίκτυο ή σύνδεσης μεταξύ αυτών μέσω κάποιας ασύρματης τεχνολογίας.

Η καθημερινότητα μας, πλέον, εμπεριέχει μια μεγάλη ποικιλία διασυνδεδεμένων συσκευών. Ο λόγος που ο αριθμός αυτών συνεχίζει να αυξάνεται με τόσο έντονο ρυθμό είναι οι καινοτομίες τόσο στις διαδικτυακές υποδομές και τεχνολογίες όσο και η μείωση του κόστους των μικροελεγκτών, με αποτέλεσμα ένας μεγάλος αριθμός συσκευών να αποκτά μια καινούργια σειρά από ιδιότητες.

Οι έξυπνες κινητές συσκευές(Smartphones) και η εξέλιξη του διαδικτύου κατέστησε ένα ιδανικό περιβάλλον για το διαδίκτυο των πραγμάτων. Οι συσκευές που το απαρτίζουν, συνήθως, συλλέγουν δεδομένα, ο ρόλος που έχουν μερικές από αυτές είναι να αποφασίζουν και να λειτουργούν ανάλογα, δίχως απαραίτητα να υπάρχει άμεση ανθρώπινη παρέμβαση. Άλλες φορές προβάλλουν αυτά τα δεδομένα μέσω κάποιας διεπαφής, συνήθως με την βοήθεια κάποιας ιστοσελίδας ή εφαρμογής. Η συλλογή των δεδομένων, γίνεται με την βοήθεια αισθητηρίων τα οποία τοποθετούνται ανάλογα το σύστημα-περιβάλλον και τι καλείται η συσκευή να επιβλέπει.

Αυτού του είδους οι συσκευές έχουν υποστηρίξει αρκετά την αυτοματοποίηση ορισμένων διαδικασιών, τόσο στην απλή καθημερινότητα όσο και σε βιομηχανικά σενάρια, όπως η μαζική παραγωγή, μειώνοντας έτσι σημαντικά τον χρόνο επεξεργασίας των δεδομένων που συλλέγονται κάνοντας έτσι ευκολότερη την διαπίστωση ορθής λειτουργίας τόσο της συσκευής όσο και του ελεγχόμενου συστήματος-περιβάλλοντος.

Αναφορικά, μερικά από τα πεδία εφαρμογής⁸ είναι :

- I. Ιατρική και υγειονομική περίθαλψη(IoMT)⁹
- II. Οικιακός αυτοματισμός
- III. Γεωργία
- IV. Μαζική παραγωγή
- V. Διαχείριση ενέργειας
- VI. Διαχείριση συντήρησης
- VII. Διαχείριση στόλου μεταφορών
- VIII. Logistics αποθηκών/ Καταγραφή αποθεμάτων¹⁰
- IX. Αυτόνομη Οδήγηση

1.2 Ταχεία Πρωτοτυποποίηση

Στην εποχή μας οι παραγωγικές μονάδες είναι τόσο εξελιγμένες και ευέλικτες που το κόστος παραγωγής προϊόντων ολοένα και μειώνεται. Αυτό έχει σαν αποτέλεσμα, να υπάρχουν εργαλεία και υπηρεσίες στην αγορά τα οποία είναι ευρέως διαθέσιμα στο καταναλωτικό κοινό λόγω του, πλέον, χαμηλού κόστους τους. Αυτή η εξέλιξη έχει δώσει μια καινούργια πνοή στον κόσμο της σχεδίασης προϊόντων, μιας και γίνεται εφικτή η σχεδίαση και υλοποίηση διαφόρων καινοτομιών με πολύ χαμηλότερο κόστος και χρόνο εκτέλεσης.

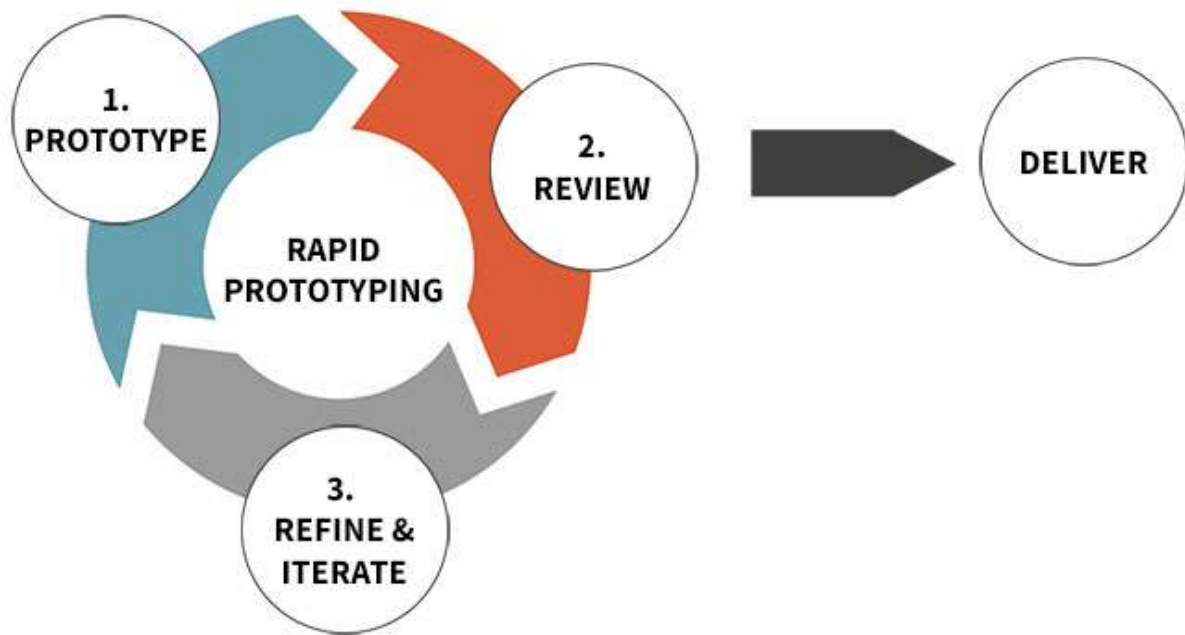
Η ανάπτυξη προϊόντων, προϋποθέτει ορθή παραμετροποίηση των τμημάτων του προϊόντος τόσο αισθητικά όσο και λειτουργικά με την βοήθεια ειδικού CAD(Computer Aided Design) λογισμικού. Λιγότερα λάθη σημαίνουν καλύτερη εφαρμογή των τμημάτων, μεταξύ τους και εν τέλη ευκολότερη συναρμολόγηση, από ένα προϊόν το οποίο απαιτεί κάποιου είδους επιπλέον παρέμβαση πέρα από τις προγραμματιζόμενες φάσεις παραγωγής.

Η ανάγκη για τον έλεγχο του πρωτότυπου, οδήγησε στην δημιουργία μεθόδων ταχείας πρωτοτυποποίησης, οι οποίες παρόλο που δεν είναι σχεδιασμένες για εκτέλεση μαζικής παραγωγής, είναι ιδανικές για γρήγορες υλοποιήσεις σχεδίων που οδηγούν σε μια πρώτη εκτίμηση της ιδέας του προϊόντος και αποτελεσματικότητας της φάσης της σχεδίασης. Κύριος σκοπός είναι η μελέτη για το αν το συγκεκριμένο προϊόν είναι παραγωγικά και οικονομικά συμφέρον. Οι μέθοδοι αυτοί, κατατάσσονται σε προσθετικές και αφαιρετικές μεθόδους παραγωγής και οι δύο εκμεταλλεύονται σχέδια, υλοποιημένα με λογισμικά CAD τα οποία συνήθως κατα την πρώτη φάση επεξεργασίας τους εξάγονται υπο την μορφή .STL ώστε να μετατραπούν, αργότερα, σε ειδικές εντολές τύπου G-Code.

⁸ <https://www.fractal.com/en/blog/the-9-most-important-applications-of-the-internet-of-things>

⁹ <https://healthtechmagazine.net/article/2020/01/how-internet-medical-things-impacting-healthcare-performance>

¹⁰ Συνήθως χρησιμοποιείται η τεχνολογία RFID



Εικόνα 1.2.1
Ο κύκλος της ταχείας πρωτοτυποποίησης
για την δημιουργία προϊόντων

1.2.1 Αφαιρετική μέθοδος παραγωγής

Η συγκεκριμένη μέθοδος παραγωγής είναι ίσως από τις πρώτες μεθόδους ταχείας επεξεργασίας υλικών σε ένα βιομηχανικό περιβάλλον, ακολουθεί την διαδικασία της κοπής ή αφαίρεσης υλικού από μια μάζα, με σκοπό την δημιουργία ενός στοιχείου με συγκεκριμένες διαστάσεις και μορφολογία.

Μια μεγάλη καινοτομία¹¹ στην συγκεκριμένη διαδικασία, ήταν η εφαρμογή ηλεκτρικών κινητήρων σε κάθε έναν από τους τρεις άξονες (x,y,z) και ο έλεγχος αυτών από κάποιον μικρο-ελεγκτή ή υπολογιστή. Οι μηχανές CNC(Computer Numerical Control), αξιοποιούν εντολές τύπου G-Code και μια μεγάλη ποικιλία εργαλείων κατεργασίας για να εκτελέσουν με πολύ μεγάλη ακρίβεια, κινήσεις, ώστε να γίνεται όσο το δυνατόν καλύτερα η αφαίρεση του υλικού. Συνήθως αποτελούνται από 3 άξονες κίνησης, αν και σε βιομηχανικό επίπεδο συχνά συναντάμε 5 άξονες.

Μερικά από τα εργαλεία κατεργασίας ή αλλιώς τα άκρα κοπής:

- I. Μοτερ με ειδικό άκρο συσφιξης κοπτικών εργαλείων
- II. Δέσμη φωτός ή CO₂¹² - Laser
- III. Water jet cutter¹³

¹¹ <https://mainstaymfg.com/the-origins-of-cnc-machining/>

¹² <https://www.3erp.com/blog/three-main-types-of-lasers-for-cutting/>

¹³ <https://all3dp.com/2/cnc-water-jet-all-you-need-to-know/>

1.2.2 Προσθετική μέθοδος παραγωγής

Μια από τις πιο γνωστές μεθόδους ταχείας πρωτοτυποποίησης, η οποία εκμεταλλεύεται αρκετές τεχνολογίες, είναι αυτή της προσθετικής μεθόδου παραγωγής. Συνήθως στην συγκεκριμένη κατηγορία, ανήκουν οι γνωστοί σε όλους μας, τρισδιάστατοι εκτυπωτές οι οποίοι χρησιμοποιούν διάφορες τεχνικές και υλικά για να εναποθέσουν μια ποσότητα υλικού, το οποίο συνήθως είναι κάποιο πολυμερές με ή και χωρίς επιπλέον προσμίξεις.

Οι τεχνολογίες που την απαρτίζουν έχουν την ιδιότητα παραγωγής αντικειμένων των οποίων η μορφολογία μπορεί να είναι και περίπλοκη, κάτι το οποίο είναι πάρα πολύ δύσκολο να επιτευχθεί από την αφαιρετική μέθοδο ακόμα και από το ιδανικό σενάριο των 5 αξόνων κίνησης.

Τα υλικά που χρησιμοποιούνται για αυτήν την τεχνολογία, διαφέρουν από τεχνολογία σε τεχνολογία, μια και η εναπόθεση τους πραγματοποιείται από διαφορετικές τεχνικές. Μερικά από αυτά μπορούν να έχουν μορφή:

- I. Νήματος
- II. Υγρού (Resin)
- III. Σκόνης

Στα επόμενα κεφάλαια θα αναφέρουμε περισσότερες λεπτομέρειες για αυτά που προαναφέρθηκαν, όπου και θα εξηγήσουμε με λεπτομέρειες τον τρόπο διαχείρισης των συγκεκριμένων υλικών αλλά και τις τεχνολογίες που απαρτίζουν την προσθετική μέθοδο παραγωγής.

1.3 Ιστορική αναδρομή της 3D Εκτύπωσης

Η τρισδιάστατη εκτύπωση δεν πρόκειται για μια πρόσφατη καινοτομία, η γέννηση της χρονολογείται από την δεκαετία των 80's. Όπως και κάθε καινοτομία, έτσι και η τρισδιάστατη εκτύπωση έπρεπε να περάσει από ορισμένες φάσεις ωρίμανσης, ώστε να καταφέρει να φτάσει στα επίπεδα αποδοτικότητας που γνωρίζουμε σήμερα. Τα στάδια της :

- I. Το βρεφικό, ήταν από την χρονολογία 1981 έως και 1999
- II. Το εφηβικό, ήταν από την χρονολογία 1999 έως και 2010
- III. Το ενηλικίωσης, από το 2011 έως και σήμερα

1.3.1 Η Γέννηση

Τα πρώτα τεχνολογικά βήματα που οδήγησαν στην τρισδιάστατη εκτύπωση που γνωρίζουμε σήμερα, έγιναν το 1981 από τον Dr. Hideo Kodama ο οποίος εργαζόταν στο Δημοτικό Ινστιτούτο Βιομηχανικής Έρευνας της Ναγκόγια, όπου και έγινε αναφορά για την δημιουργία ενός συστήματος ταχείας πρωτοτυποποίησης η οποία αξιοποιούσε φωτοπολυμερή σαν βασικό υλικό. Έτσι πραγματοποιήθηκε και η πρώτη τρισδιάστατη εκτύπωση ενός στερεού αντικειμένου σε διαδοχικές στρώσεις.

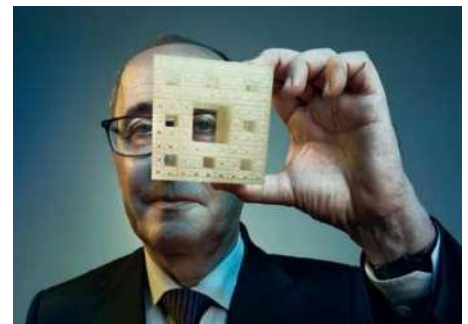
Αν και αρκετοί επιστήμονες προσπάθησαν να αναπτύξουν τα επόμενα βήματα της τεχνικής που χρησιμοποιήθηκε από τον Dr. Hideo Kodama, όπως και να δηλώσουν διπλώματα ευρεσιτεχνίας για αυτήν.

Μια ομάδα Γάλλων επιστημόνων, ήταν οι πρώτοι που ξεκίνησαν να χτίζουν τον πρώτο τρισδιάστατο εκτυπωτή, ο οποίος χρησιμοποιούσε την μέθοδο της στερεολιθογραφίας για την παραγωγή απτών αντικειμένων. Όλα ξεκίνησαν, όταν ένα από τα μέλη της ομάδας, ο Alain le Méhauté ήθελε να δημιουργήσει ένα Φράκταλ αντικείμενο το οποίο ήταν απίθανο να υλοποιηθεί από κάποια άλλη μέθοδο παραγωγής. Έπειτα από συζήτηση με τον Olivier de Witt, ο οποίος εργαζόταν με ακτίνες λέιζερ για την Cilas, ξεκίνησαν τα πρώτα πειράματα για την δημιουργία του φρακταλ αντικείμενου. Τα πρώτα πειράματα δεν είχαν κάποιο αποτέλεσμα και έτσι παρουσίασαν την ιδέα τους στον Jean-Claude André, ο οποίος είχε την ιδέα το αντικείμενο να χτίζεται στρώση προς στρώση(layer-by-layer). Η διαδικασία για την απόκτηση διπλώματος ευρεσιτεχνίας, ξεκίνησε τρεις εβδομάδες νωρίτερα από τον Chuck hull, για τον οποίο θα μιλήσουμε παρακάτω. Δυστυχώς, η ιδέα τους δεν λήφθηκε σοβαρά από το Γαλλικό Εθνικό Κέντρο Επιστημονικής Έρευνας(CNRS), για το οποίο εργαζόταν ο Jean-Claude André, διότι θεωρούσαν πως δεν θα έχει κάποιο σημείο εφαρμογής για εκείνους. Η ιδέα εγκαταλείφθηκε από τους δημιουργούς της μιας και δεν υπήρχαν περαιτέρω διαθέσιμοι οικονομικοί πόροι.

Το 1984, ο Charles (Chuck) Hull, εφήυρε την στερεολιθογραφία, η οποία έδινε την δυνατότητα σε σχεδιαστές να δημιουργούν τρισδιάστατα μοντέλα μέσω ψηφιακών δεδομένων, τα οποία χρησιμοποιούνταν για την δημιουργία απτών αντικειμένων. Η σημαντικότερη καινοτομία του ήταν η δημιουργία του STL File Format, το οποίο μεταφράζει ένα σχήμα σε ψηφιακά δεδομένα, όπου και έπειτα αυτό χρησιμοποιείται για να παραχθεί η γλώσσα λειτουργίας(G-CODE) ώστε να υλοποιηθεί ένα αντικείμενο. Η τεχνολογία της στερεολιθογραφίας(SLA) εκμεταλλεύεται



Εικόνα 1.3.0
Ο Hideo Kodama



Εικόνα 1.3.1
Ο Alain le Méhauté που κρατάει το Φράκταλ αντικείμενο που σχεδίασε



Εικόνα 1.3.2
Το ομοίωμα του Chuck Hull, μπροστά από τον πρώτο εμπορικά διαθέσιμο SLA εκτυπωτή

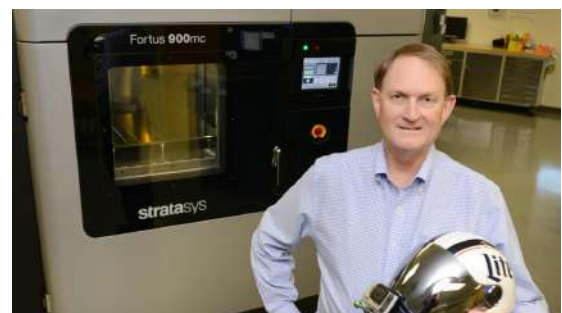
την ευαισθησία των φωτοπολυμερών στην υπεριώδη ακτινοβολία, για την σκλήρυνση του υλικού το οποίο είναι αρχικά σε υγρή μορφή. Η ακτίνα υπεριώδους φωτός, κινείται επιλεκτικά με σκοπό να σκληρύνει το υλικό σε στρώσεις(layer-by-layer) δημιουργώντας έτσι ένα τελικό, στερεό, αντικείμενο συγκεκριμένου μεγέθους και μορφολογίας. Εκείνη την εποχή κατέθεσε και το δίπλωμα ευρεσιτεχνίας προς αξιολόγηση και αυτό έγινε δεκτό το 1986, όπου και έγινε συνιδρυτής της εταιρείας 3D Systems. Στις αρχές του 1988 έγινε εμπορικά διαθέσιμος ο πρώτος εκτυπωτής SLA, από την 3D Systems, με το όνομα SLA-1.

Το 1988 ο Carl Deckard με την υποστήριξη του καθηγητή Joseph Beaman, στα πλαίσια του πανεπιστημίου του Τέξας, κατάφεραν να αποκτήσουν δίπλωμα ευρεσιτεχνίας για την τεχνολογία SLS(Selective Laser Sintering). Την ίδια εποχή, ο Carl ίδρυσε την εταιρία DTM στην οποία και μετέφερε το δίπλωμα ευρεσιτεχνίας το 1989, έτσι λοιπόν, άρχισε ο ανταγωνισμός της DTM και της 3D systems στο πεδίο της προσθετικής μεθόδου παραγωγής. Δυστυχώς, παρόλο που η τεχνολογία είχε τραβήξει το ενδιαφέρον μεγάλων εταιριών η πώληση των μηχανημάτων δεν ήταν τόσο κερδοφόρα με αποτέλεσμα, τα οικονομικά της προβλήματα ολο ένα και να πληθαίνουν. Το 2001, η συγκεκριμένη εταιρεία εξαγοράστηκε από την 3D Systems έναντι 45 εκ. Δολαρίων.



Εικόνα 1.3.3
Άρθρο που εκδόθηκε με αφιέρωμα την τρισδιάστατη εκτύπωση, με τους Carl Deckard και Dr. Joseph Beaman

Μια ακόμη σπουδαία ανακάλυψη για τον κόσμο της τρισδιάστατης εκτύπωσης, η οποία είναι και η πιο συνήθης μέθοδος λειτουργίας των περισσότερων τρισδιάστατων εκτυπωτών της αγοράς, ήταν η τεχνολογία FDM(Fused Deposition Modeling), από τον Scott Grump. Όλα ξεκίνησαν, όταν ήθελε να φτιάξει ένα παιχνίδι-βατραχάκι για την 2 χρονών κόρη του, αξιοποιώντας λίγο πλαστικό και ένα πιστόλι σιλικόνης, διαπίστωσε πως μπορεί να δημιουργήσει ένα τρισδιάστατο αντικείμενο, όμως το αποτέλεσμα ήταν ακριβώς το αντίθετο από το επιθυμητό. Εκεί, λοιπόν, σκέφτηκε πως θέλει αυτή η διαδικασία να είναι αυτοματοποιημένη, δηλαδή ένα πιστόλι σιλικόνης πάνω σε έναν κινητό άξονα XYZ. Όσο συνέχιζε τα πειράματα, εκείνος και η γυναίκα του υπέβαλαν αίτηση για δίπλωμα ευρεσιτεχνίας το 1989. Το 1992, ίδρυσε την εταιρεία Stratasys, όπου και είναι ο πρόεδρος του διοικητικού της συμβουλίου μέχρι και σήμερα.



Εικόνα 1.3.4
Ο Scott Grump με ένα από τα μηχανήματα που έχει δημιουργήσει η εταιρία του.

1.3.2 Η Εξέλιξη

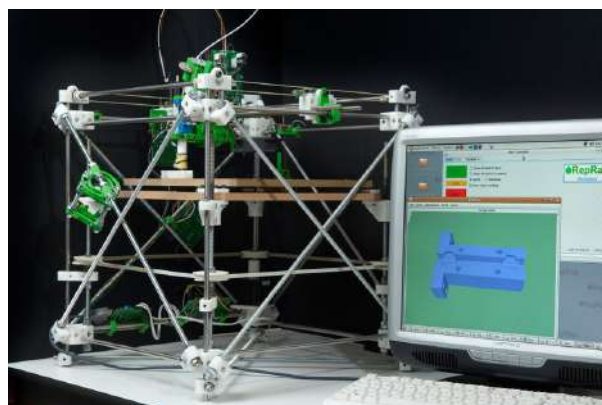
Κατά το εφηβικό της στάδιο, η τρισδιάστατη εκτύπωση δεν ήταν ακόμα γνωστή στο ευρύ κοινό, μιας και αποτελούσε μια τεχνολογία πολύ νέα με πολύ λίγους μηχανικούς να εξειδικεύονται σε εκείνη.

Το 1999, έγιναν οι πρώτες σοβαρές καινοτομίες στην δημιουργία οργάνων μέσω τρισδιάστατων εκτυπωτών. Επιστήμονες από το πανεπιστήμιο αναγεννητικής ιατρικής του Wake Forest, έφτιαξαν αρχικά ένα σύνθετο ικρίωμα μιας ουροδόχου κύστης, όπου και το κάλυψαν με τα κύτταρα του ασθενή, ώστε να μην υπάρχει μεγάλη πιθανότητα απόρριψης του οργάνου από τον οργανισμό του. Άλλες δημιουργίες που ακολούθησαν, στο πεδίο της ιατρικής ήταν το πρώτο τρισδιάστατο εκτυπωμένο προσθετικό πόδι, το οποίο αποτελούνταν από πολλά σύνθετα μέρη, τα πρώτα βιο-εκτυπώμενα αιμοφόρα αγγεία και το πρώτο συνθετικό νεφρό αξιοποιώντας την τεχνική που είχαμε αναφέρει προηγουμένως με την εναπόθεση κυττάρων του ασθενούς οργανισμού. Αυτές οι ανακαλύψεις αποτέλεσαν θεμέλιο για τα επόμενα βήματα αυτού του τρόπου δημιουργίας σύνθετων οργάνων και προσθετικών μελών. Η συγκεκριμένη τεχνολογία, όντας σύνθετη μέχρι και σήμερα, απαιτεί πάρα πολλά χρόνια ανάπτυξης, ώστε τα αποτελέσματα που παράγει να είναι αποδεκτά και ασφαλή για χρήση σε ασθενείς.

Με την ανάπτυξη του διαδικτύου, κατά τις αρχές του 21ου αιώνα, η γνώση απέκτησε ψηφιακή μορφή, πλέον οι απόψεις και οι τεχνικές γνώσεις μπορούσαν να συζητηθούν ελεύθερα μέσω του διαδικτύου. Εκεί λοιπόν, αναπτύχθηκε η φιλοσοφία της ελεύθερου λογισμικού και κυρίως τις καινοτομίες ανοιχτής πηγής (Open Source), οι οποίες δημιούργησαν κοινότητες δημιουργών με σκοπό ένα κοινό αποτέλεσμα. Ακολουθώντας την λογική του Open Source, γινόταν διαμοιρασμός γνώσης μεταξύ ενδιαφερόμενων αλλά και συνεισφορά σε κάποιο έργο, με αποτέλεσμα, να γίνετε ανάπτυξη καινοτομιών βήμα προς βήμα, τις οποίες μπορεί ο οποιοσδήποτε να έχει πρόσβαση, ακολουθώντας όμως μερικές άδειες, ώστε να μην υπάρχει κερδοφορία εις βάρος των δημιουργούν που κατέχουν ορισμένα δικαιώματα, γνωστά και ως Creative Commons (CC). Έτσι και η τρισδιάστατη εκτύπωση προσέλκυσε το ενδιαφέρον αυτού του κινήματος με πρωτοπόρο τον Dr. Adrian Bowyer, το 2005, ο οποίος είχε την ιδέα δημιουργίας ενός μηχανήματος το οποίο είχε την ιδιότητα να “αναπαράγεται”, δηλαδή να εκτυπώνει το ίδιο τα πλαστικά κομμάτια που ενώνουν τα υπόλοιπα μέρη της κατασκευής.



Εικόνα 1.3.5
Ένα τρισδιάστατο εκτυπωμένο προσθετικό πόδι.



Εικόνα 1.3.6
Ο πρώτος Reprap 3D Printer, ονόματι Darwin.

Έτσι, ιδρύθηκε το Replication Rapid-Prototyper Project ή Reprap Project, το οποίο είναι ένα από τα μεγαλύτερα project ανάπτυξης hardware και λογισμικού στον κόσμο του Do-it-yourself 3D Printing μέχρι και σήμερα.

Προς το τέλος αυτής της περιόδου, η τρισδιάστατη εκτύπωση γινόταν ολοένα και πιο γνωστή, χάρις τις πρώτες μηχανές SLS οι οποίες έγιναν εμπορικά διαθέσιμες και την ανάπτυξη περισσότερων υλικών που μπορούσαν να χρησιμοποιηθούν. Ειδικά το 2009, έπειτα από την λήξη του διπλώματος ευρεσιτεχνίας της FDM τεχνολογίας εκτύπωσης, που είχε αναπτύξει ο Scott Grump, άρχισαν να γίνονται διαθέσιμα μηχανήματα σε μορφή kit στο ευρύ κοινό.

1.3.3 Η Ωρίμανση

Στο στάδιο της ενηλικίωσης της, η τρισδιάστατη εκτύπωση εξελίσσεται ολοένα και γρηγορότερα, μιας και η τεχνολογία έχει κάνει άλματα από την εποχή που πρωτοεμφανίστηκε. Το κόστος παραγωγής, και εν συνεχεία πώλησης, έχει μειωθεί δραματικά, με αποτέλεσμα ολοένα και περισσότερα μηχανήματα να βρίσκονται στην διάθεση μιας πολύ μεγάλης γκάμας ανθρώπων, από χομπίστες μέχρι και επαγγελματίες. Βέβαια, δεν θα μπορούσαν να χρησιμοποιηθούν, αν οι διεπαφές τους δεν ήταν φιλικές προς τους χρήστες οποιασδήποτε κατηγορίας, εδώ είναι επίσης σημαντικό να αναφέρουμε πως και η ανάπτυξη των προγραμμάτων σχεδίασης τρισδιάστατων μοντέλων έχει δώσει μια μεγάλη ώθηση, ώστε χρήστες να μπορέσουν να εκμεταλλευτούν την συγκεκριμένη τεχνολογία. Η ακρίβεια, αυτών των μηχανημάτων έχει βελτιωθεί σημαντικά και όσο πάει βελτιώνεται, καταφέροντας έτσι, να παράγουν αποτελέσματα με σωστή διαστασιολόγηση όπως όρισε ο χρήστης στο ψηφιακό πρωτότυπο.

Η τρισδιάστατη εκτύπωση, έχει δώσει σημαντική έμπνευση στην ανάπτυξη καινοτομιών σε πολλούς τομείς, άλλοτε σε παραγωγικό στάδιο ή και σε στάδιο πρωτοτυποποίησης. Πλέον, έχουν αρχίσει να δημιουργούνται τρισδιάστατα εκτυπωμένα σπίτια, με αποτέλεσμα το κόστος να μειώνεται σημαντικά και έτσι να υπάρχει μεγαλύτερη υποστήριξη οικογενειών με χαμηλό εισόδημα. Η ιατρική, όπως και στην εφηβική εποχή της τεχνολογίας, συνεχίζει να ανακαλύπτει και να αναπτύσει καινοτομίες οι οποίες έχουν σημαντικό ρόλο στην υποστήριξη βιολογικών οργανισμών, με την δημιουργία τεχνητών οργάνων και μελέτη λειτουργικότητας αυτών, ώστε να μπορέσει στο μέλλον να αξιοποιηθεί δίχως αμφιβολία ή κίνδυνο.

Τα υλικά που χρησιμοποιούνται σήμερα, για την παραγωγή τρισδιάστατων αντικειμένων, φέρουν πολλές ιδιότητες, δίνοντας έτσι επιλογή ανάλογα το πεδίο εφαρμογής. Μερικά είναι σχεδιασμένα για να αντέχουν στις καταπονήσεις και στις εξωτερικές καιρικές συνθήκες, άλλα έχουν αντιμικροβιακές ιδιότητες και χρησιμοποιήθηκαν κατα την παραγωγή μασκών και ασπίδων προστασίας την



Εικόνα 1.3.7
Ακροφύσιο εναπόθεσης υγρού
σκυροδέματος

περίοδο της πανδημίας του κορονοϊου, για τα περισσότερα υλικά θα αναφερθούμε λεπτομερώς σε επόμενο κεφάλαιο. Εν τέλη, παρατηρούμε πως η τρισδιάστατη εκτύπωση αποτελεί μια μέθοδο παραγωγής πρωτοτύπων με μόνο περιορισμό την φαντασία του χρήστη.

1.4 Η Ιδέα

Η παρούσα εργασία καλείται να δώσει λύση σε φάρμες εκτυπωτών όπου ο ορθός έλεγχος του κάθε μηχανήματος και του αποτελέσματος του είναι χρονοβόρος και δύσκολος. Τα σφάλματα παραγωγής και η ένδειξη κακής λειτουργίας που μπορούν να προκύψουν, λόγω του μεγάλου αριθμού μηχανημάτων, δεν μπορούν εύκολα ή έγκαιρα να εντοπιστούν μέχρι να είναι πολύ αργά. Για αυτό το λόγο θα πρέπει να δημιουργήσουμε ένα μέσο επίβλεψης με σκοπό την ταχύτερη πρόγνωση προβλήματος και αντιμετώπισης αυτού. Η έρευνα που θα πραγματοποιήσουμε βασίζεται σε 3 βασικούς πυλώνες :

- I.** Θα αξιολογήσουμε ένα μοντέλο μηχανικής μάθησης, το οποίο έχει εκπαιδευτεί συγκεκριμένα ώστε να αναγνωρίζει την εσφαλμένη λειτουργία κάποιου βηματικού κινητήρα, θα μπορέσουμε να αναγνωρίσουμε δυσκολίες στην κίνηση κάποιου άξονα ή της ροής του υλικού.
- II.** Θα παρακολουθούμε τη ροή του υλικού, κάθε μηχανήματος, θα γνωρίζουμε πόσο καταναλώθηκε και πόσο έχει απομείνει σε περίπτωση που χρειαστεί να αλλάξει το καρούλι του υλικού. Υποβοηθώντας έτσι το σύστημα προμηθειών της εγκατάστασης, για την έγκαιρη παραγγελία υλικών.
- III.** Χρησιμοποιώντας το τοπικό δίκτυο θα παρουσιάζουμε όλες τις συλλεγμένες πληροφορίες από τα μηχανήματα της φάρμας σε μία φιλική προς τον χρήστη διεπαφή.
- IV.** Θα γίνεται παρακολούθηση δεδομένων που λαμβάνονται από τα αισθητήρια που θα τοποθετήσουμε πάνω στους τρισδιάστατους εκτυπωτές και σε περίπτωση κάποιας κρίσιμης κατάστασης θα γίνεται και η κατάλληλη ενημέρωση έτσι ώστε να γίνουν και τα ανάλογα βήματα για να αντιμετωπιστούν τα όποια προβλήματα έχουν παρουσιαστεί

Κεφάλαιο 2

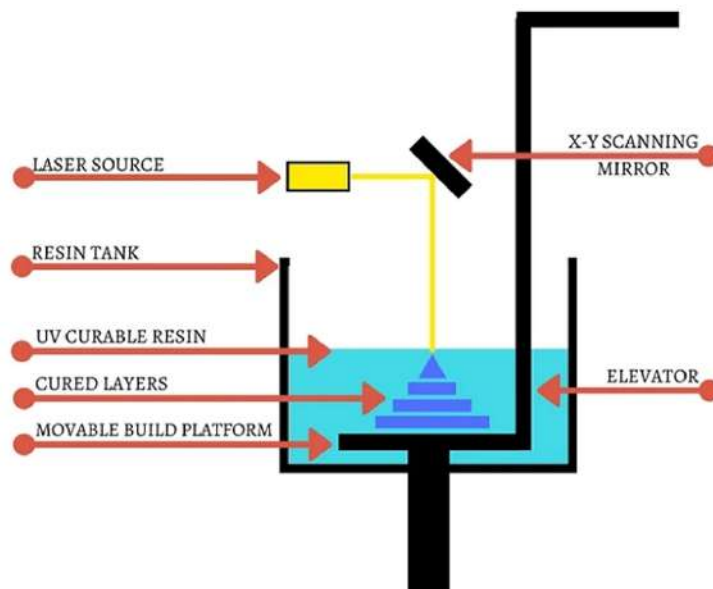
Τεχνολογίες Τρισδιάστατων Εκτυπωτών

Έχοντας αναφέρει προηγουμένως στην ιστορική εξέλιξη της τρισδιάστατης εκτύπωσης, παρατηρήσαμε πως υπάρχει μια πληθώρα τεχνολογιών η οποία προσπαθεί να συμβάλει στην εύρεση γρήγορων και αποδοτικών λύσεων στον τομέα της προσθετικής μεθόδου παραγωγής. Σε αυτό το κεφάλαιο θα αναφερθούμε συνοπτικά στις τεχνολογίες που απαρτίζουν αυτήν την μέθοδο, σύμφωνα με τις πιο πρόσφατες εξελίξεις του συγκεκριμένου χώρου.

2.1 Laser SLA - Stereolithography

Η στερεολιθογραφία, αποτελεί ιστορικά ένα σημαντικό θεμέλιο για την τρισδιάστατη εκτύπωση όπως την γνωρίζουμε σήμερα. Με την ανάπτυξη της έγινε εφικτή η εξέλιξη καινούργιων και διαφορετικών τεχνολογιών προσθετικής παραγωγής αντικειμένων, με αποτέλεσμα, να υπάρχει η δυνατότητα επιλογής τρόπου παραγωγής ανάλογα τις ανάγκες και τους περιορισμούς.

Όπως αναφέραμε και προηγουμένως, η στερεολιθογραφική μέθοδος τρισδιάστατης εκτύπωσης εκμεταλλεύεται την ευαισθησία ορισμένων πολυμερών, στο φως, έτσι πραγματοποιώντας στοχευμένη έκθεση αυτών σε υπεριώδες γίνεται τοπική σκλήρυνση του υλικού και έτσι χτίζεται το επιθυμητό τρισδιάστατο σχέδιο στρώση προς στρώση.



Εικόνα 2.1.0
Κύρια μέρη ενός τρισδιάστατου εκτυπωτή, τύπου SLA

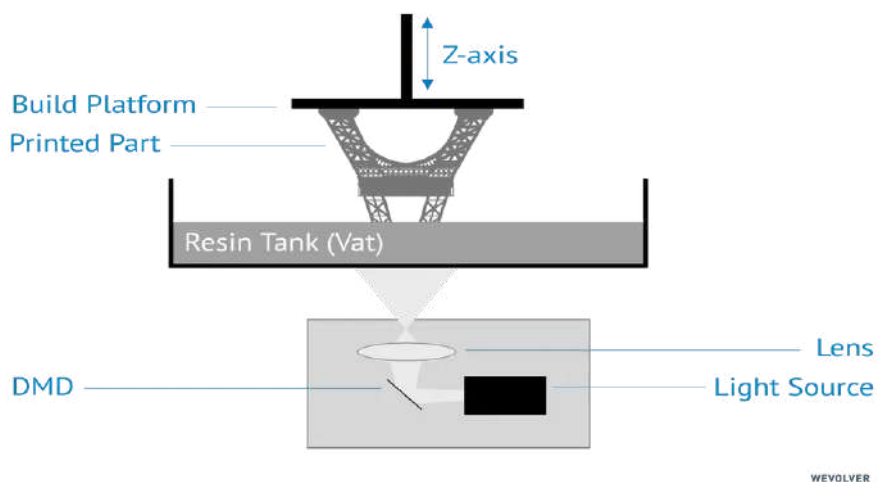
Για την λειτουργία ενός SLA εκτυπωτή, όπως παρατηρείται και στην εικόνα παραπάνω, χρειάζονται :

- I. Ένα δοχείο στο οποίο θα υπάρχει το φωτοευαίσθητο πολυμερές, δηλαδή η φωτοδραστική υγρή ρητίνη.
- II. Μια επιφάνεια η οποία θα μπορεί να κινείται, σε έναν άξονα, συγκεκριμένα τον Z.
- III. Ένα υψηλής ισχύος, υπεριώδες laser το οποίο κατευθύνεται με την βοήθεια καθρεπτών οι οποίοι αναλαμβάνουν τους ρόλους των αξόνων X και Y..
- IV. Μια υπολογιστική διεπαφή, συνήθως μια ειδικά σχεδιασμένη πλακέτα με κάποιον μικροελεγκτή, η οποία μπορεί να διαχειριστεί την κίνηση της επιφάνειας εκτύπωσης και την λειτουργία της υπεριώδους φωτοδέσμης.

Η συγκεκριμένη τεχνολογία, αξιοποιεί μια δέσμη φωτός για να εναποθέσει εκλεκτικά τα διαδοχικά επίπεδα που θα αποτελέσουν το στερεό αντικείμενο, αποτελεί την ακριβέστερη μέθοδο τρισδιάστατης εκτύπωσης έχοντας τις αυστηρότερες ανοχές διαστάσεων, ακολουθώντας έτσι, πιο πιστά το πρωτότυπο CAD σχέδιο.

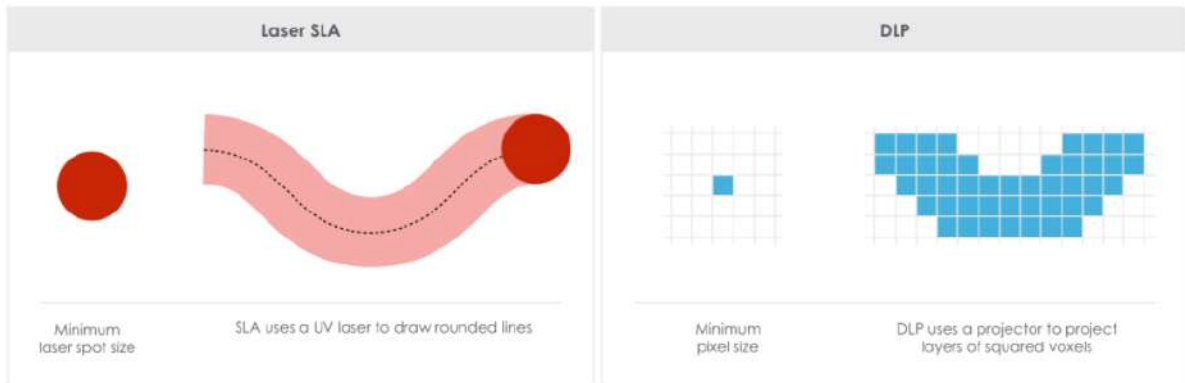
2.1.1 SLA-DLP - Digital Light Processing

Η τεχνολογία DLP, ακολουθεί πιστά τις αρχές λειτουργίας της τεχνολογίας SLA, χρησιμοποιώντας ακριβώς τις ίδιες ιδιαιτερότητες και αρχές λειτουργίας, δηλαδή, του ενός άξονα κίνησης και του δοχείου συλλογής της ρητίνης, με την μονη διαφορά ότι η σκλήρυνση του υγρού πολυμερούς γίνεται από μια συσκευή ψηφιακού προβολέα, για παράδειγμα μια οθόνη LCD.



Εικόνα 2.1.1
Κύρια μέρη ενός τρισδιάστατου εκτυπωτή, τύπου DLP

Η συγκεκριμένη τεχνολογία επειδή χρησιμοποιεί μια ψηφιακή οθόνη για την προβολή του φωτός για την ταυτόχρονη σκλήρυνση, καθιστά την ανάλυση του κάθε στρώματος δέσιμο με την ανάλυση της οθόνης σε τετραγωνικά εικονοστοιχεία (pixels). Αυτό σημαίνει πως η ταχύτητα ολοκλήρωσης ενός τρισδιάστατου μοντέλου είναι πολύ πιο γρήγορη σε σχέση με έναν εκτυπωτή τεχνολογίας Laser SLA μιας και η έκθεση του κάθε στρώματος γίνεται ταυτόχρονα.



Εικόνα 2.1.2
Σύγκριση στρωμάτων εκτύπωσης

Όσον αφορά το τελικό αποτέλεσμα, η χρήση Laser SLA προσφέρει ένα καλύτερο φινίρισμα στην επιφάνεια του τρισδιάστατου εκτυπωμένου αντικειμένου, εξαιτίας των εικονοστοιχείων, ακολουθώντας έτσι και πιο πιστά τις διαστάσεις του σχεδίου CAD.



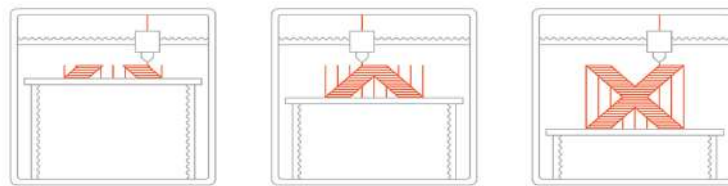
Εικόνα 2.1.2
Σύγκριση τεχνολογιών εκτύπωσης

2.2 FDM-Fusion Deposition Modeling

Η κατασκευή με εναπόθεση υλικού, αποτελεί την διασημότερη μέθοδο τρισδιάστατης εκτύπωσης, διότι υπήρξε η πρώτη που μπόρεσε να εισέλθει σχετικά νωρίς στην αγορά λόγω του κινήματος ανοιχτού κώδικα που είχε υποστηρίξει την μείωση του κόστους παραγωγής και λειτουργίας.

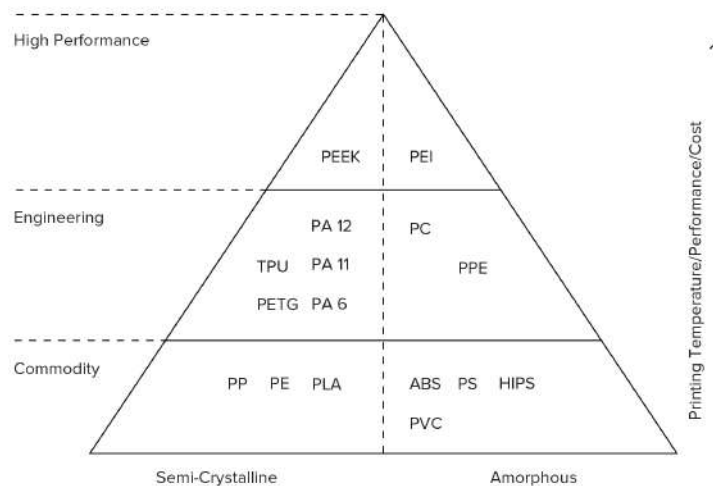
Η συγκεκριμένη τεχνολογία, είναι πολύ πιο εύκολη στην λειτουργία της σε σχέση με άλλες, ειδικά στην χρήση της από μη εξειδικευμένο προσωπικό. Σε αυτό έχει συμβάλει, σημαντικά, η κοινότητα ελεύθερου λογισμικού, προσπαθώντας στην βελτίωση του λογισμικού που φέρει η υπολογιστική μονάδα του μηχανήματος, όπως και η διεπαφή με τον τελικό χρήστη, προσφέροντας έτσι μια πληθώρα λειτουργιών και παραμέτρων.

Η βασική αρχή λειτουργίας αυτού του τύπου εκτυπωτών, είναι η χρήση ενός θερμαινόμενου ακροφυσίου, το οποίο μπορεί και κινείται εντός ενός συστήματος τριών αξόνων, με σκοπό την ρευστοποίηση του σκληρού νήματος, το οποίο αποτελείται από ένα θερμοπλαστικό πολυμερές, με σκοπό αυτο να απλωθεί επιλεκτικά στην επιφάνεια στερέωσης της εκτύπωσης και έτσι στρώση προς στρώση να δημιουργηθεί το τρισδιάστατο αντικείμενο.



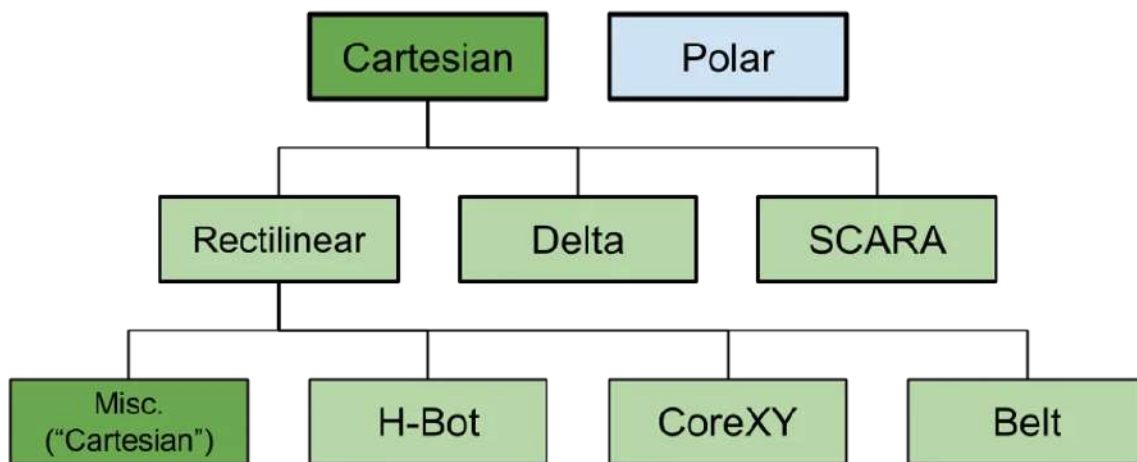
Εικόνα 2.2.1
Διαδικασία FDM εκτύπωσης

Ένα σημαντικό πλεονέκτημα που φέρει η συγκεκριμένη τεχνολογία είναι το ευρύ φάσμα διαθέσιμων υλικών που κυκλοφορούν στην αγορά. Αυτό σημαίνει, πως σε σχέση με την στερεολιθογραφία, υπάρχει μεγαλύτερη ποικιλία μηχανικών και αισθητικών ιδιοτήτων.



Εικόνα 2.2.2
Πυραμίδα θερμοπλαστικών υλικών

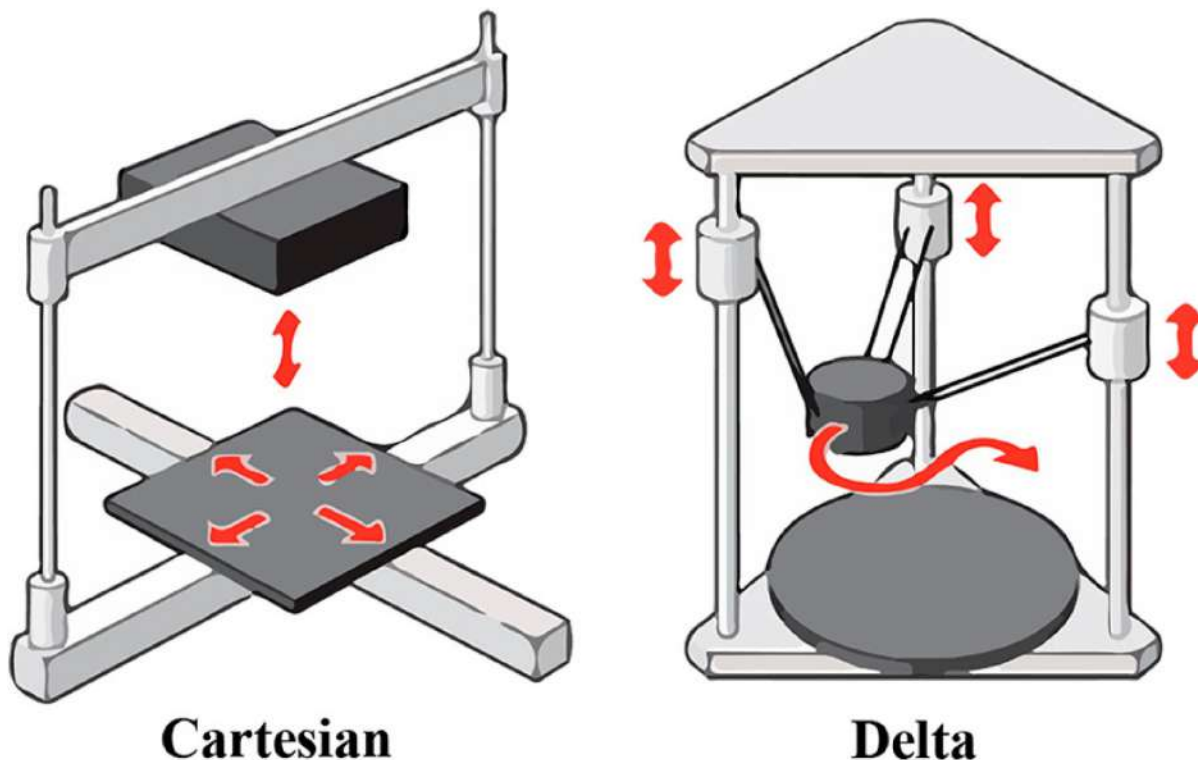
Χρησιμοποιώντας την συγκεκριμένη μέθοδο εκτύπωσης, κατά την σχεδίαση του τρισδιάστατου αντικειμένου, πρέπει, αφού επιλέγει το υλικό σύμφωνα με τις ιδιότητες και τα πλεονεκτήματα που προσφέρει, να θεωρηθούν και όλες οι δυσκολίες που μπορεί να υπάρξουν κατά την διαδικασία της εκτύπωσης, όπως για παράδειγμα η συρρίκνωση η οποία μπορεί να προκαλέσει προβλήματα κυρίως στις διαστάσεις του αντικειμένου ή ακόμη και στην επαφή του πάνω στην επιφάνεια εκτύπωσης, όπου ειδικά το τελευταίο μπορεί να αποτελέσει πρόβλημα στην ολοκλήρωση του αντικειμένου.



Εικόνα 2.2.3
Τύποι εκτυπωτών σύμφωνα με την κινηματική τους αρχή

Η κίνηση του ακροφυσίου που εναποθέτει το πλαστικό μπορεί να πραγματοποιηθεί από μια πληθώρα κινηματικών αρχών. Στην αγορά συνήθως θα συναντήσουμε μηχανήματα τα οποία ακολουθούν τις καρτεσιανές συντεταγμένες, δηλαδή η κίνηση του ακροφυσίου ή και της επιφάνειας εκτύπωσης γραμμικά σε ένα σύστημα τριών αξόνων. Οι κύριες διαφορές στα καρτεσιανά μηχανήματα παρατηρούνται κυρίως στο πως κινείται ο κάθε άξονας, δηλαδή :

- I. Τουλάχιστον ένας κινητήρας να κινεί κάθε άξονα
- II. Δυο κινητήρες αναλαμβάνουν ταυτόχρονα την κίνηση X και Y άξονα και τουλάχιστον ένας τον Z, αυτός ο τύπος ονομάζεται COREXY
- III. Τρεις κινητήρες αναλαμβάνουν την κίνηση και των τριών αξόνων, μεταφράζοντας τρεις γωνίες σε καρτεσιανές συντεταγμένες, ονομάζονται Delta.



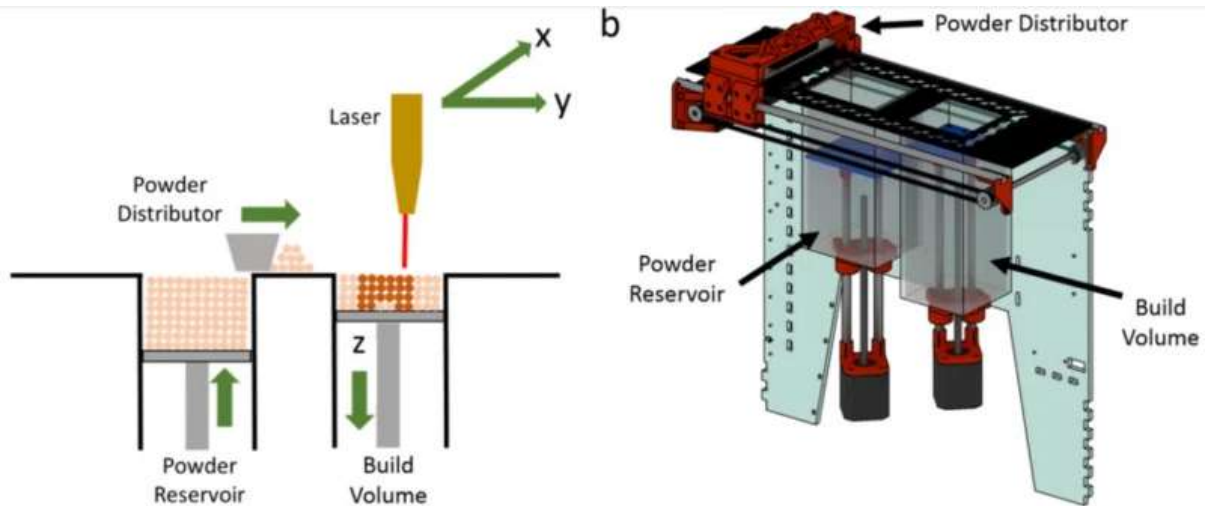
Εικόνα 2.2.4
Σύγκριση καρτεσιανού και Delta εκτυπωτή

Σε σύγκριση με άλλες μεθόδους η τεχνολογία FDM αποτελεί την οικονομικότερη και ταχύτερη λύση πρωτοτυποποίησης, σε σχέση με όλες τις άλλες της αγοράς. Αν και το φινιρίσμα στην επιφάνεια των εκτυπωμένων αντικειμένων δεν είναι ιδανική, διότι μπορούν εύκολα να διακριθούν οι στρώσεις, έχει την ικανότητα να υλοποιεί μεγάλα αντικείμενα σε πολύ μικρότερο χρόνο σε σύγκριση με την τεχνολογία SLA.

2.3 SLS - Selective Laser Sintering

Η διαδικασία της επιλεκτικής ενσωμάτωσης με Laser, αν και όχι τόσο γνωστή στο απλό καταναλωτικό κοινό λόγω του υψηλού κόστους της, αποτελεί μια ακόμη σημαντική καινοτομία στον κόσμο της τρισδιάστατης εκτύπωσης, μιας και είναι η μόνη που μπορεί να διαχειριστεί πολυμερή υλικά όπως το Nylon, με σχετικά μεγάλη ευκολία. Τα αποτελέσματα των κατασκευών θυμίζουν αποτελέσματα που μπορούν να υπάρξουν αποκλειστικά μέσω του injection Moulding με την χρήση καλουπιού.

Αναλυτικότερα, η συγκεκριμένη τεχνολογία αξιοποιεί ένα υψηλής ισχύος Laser, όπως το Laser διοξειδίου του άνθρακα, ώστε να τήξει σωματίδια μεταξύ τους. Η πρώτη ύλη, της συγκεκριμένης τεχνολογίας μπορεί να είναι πλαστικά, μέταλλα, κεραμικά ή γυαλί σε μορφή σκόνης.



Εικόνα 2.2.5
Δομή SLS τρισδιάστατου εκτυπωτή

Τα κύρια μέρη αυτού του είδους εκτυπωτή, είναι δύο δεξαμενές, ένας διανομέας υλικού και το Laser. Στην πρώτη δεξαμενή υπάρχει η ακατέργαστη πρώτη ύλη, η βάση αυτής μπορεί να κινείται πάνω σε έναν άξονα, ώστε να προωθεί το υλικό, προς τα πάνω, στον διανομέα. Ο διανομέας με την σειρά του απλώνει το υλικό πάνω από την επιφάνεια της δεύτερης δεξαμενής, όπου μετά περνάει η ακτίνα Laser και προβάλλεται πάνω από την επιφάνεια, που θα έχει το τρισδιάστατο αντικείμενο, την δισδιάστατη απεικόνιση της κάθε στρώσης που απαιτείται για να ολοκληρωθεί ολόκληρο το τρισδιάστατο σχέδιο.

Αποτελεί μια μέθοδο η οποία, ειδικά σε υλικά όπως το μέταλλο, είναι ενεργειακά οικονομικότερη για τον σχηματισμό πρωτοτύπων σε σχέση με άλλες τις αγορές, όπως για παράδειγμα η παραγωγή με την βοήθεια καλουπιού ή η χρήση φρέζας, όπου με την αφαιρετική μέθοδο παραγωγής θα σχηματιζόταν το επιθυμητό αντικείμενο. Όσον αφορά τα υλικά που χρησιμοποιούνται στην συγκεκριμένη μέθοδο, ότι δεν σκληρυνθεί, μπορεί να ξαναχρησιμοποιηθεί σε αντίθεση με την αφαιρετική διαδικασία, όπου το γρέζι που μένει δεν μπορεί να χρησιμοποιηθεί ξανά άμεσα.

Κεφάλαιο 3

Η Δομή ενός FDM 3D Printer

Έχοντας μιλήσει γενικώς για τους τρισδιάστατους εκτυπωτές, τι μορφή μπορούν να έχουν και τι υλικά διαχειρίζονται, θα εμβαθύνουμε, τώρα, πιο συγκεκριμένα σε αυτούς που καλούμαστε να χρησιμοποιήσουμε για την διεκπεραίωση της συγκεκριμένης διπλωματικής εργασίας. Για τον πειραματισμό και την ολοκλήρωση του πρωτοτύπου του συστήματος, θα χρησιμοποιήσουμε έναν Ender-3 της εταιρείας Creality, ο οποίος, πρόκειται για έναν αρκετά οικονομικό τρισδιάστατο εκτυπωτή με αρκετές δυνατότητες, κάτι που τον χρήζει ιδανικό για λειτουργία φάρμας εκτυπωτών.

Πιο συγκεκριμένα, πρόκειται για ένα μηχάνημα το οποίο φέρει τετραγωνισμένο σκελετό, από προφίλ αλουμινίου και ακολουθεί το καρτεσιανό σύστημα συντεταγμένων για την κίνηση των τριών αξόνων. Περισσότερα στοιχεία που απαρτίζουν έναν τρισδιάστατο εκτυπωτή θα αναφέρουμε στα επόμενα υποκεφάλαια ώστε να αποκτήσουμε μια πλήρη εικόνα όσον αφορά τις διαφοροποιήσεις που φέρουν αυτά τα μηχανήματα μεταξύ τους. Έτσι θα μπορέσουμε να έχουμε μια σφαιρική άποψη όσον αφορά πιθανά σημεία βελτίωσης, συχνών βλαβών και που θα μπορούσαμε να τοποθετήσουμε ορισμένα αισθητήρια, με σκοπό την επίβλεψη της ορθής λειτουργίας.



Εικόνα 3.0.1
Ο Τρισδιάστατος εκτυπωτής που χρησιμοποιούμε για την
διεκπεραίωση της εργασίας

3.1 Άξονες Κίνησης

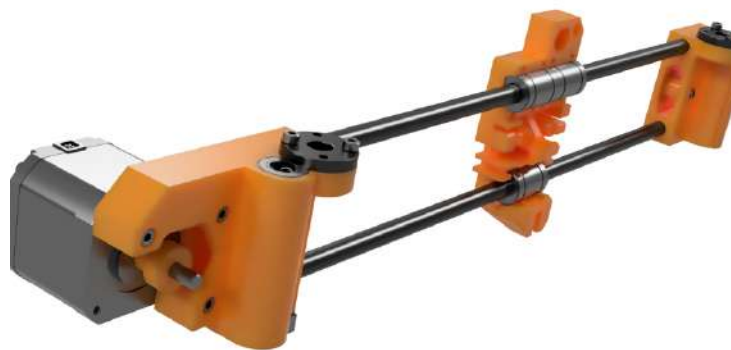
Για να δημιουργήσουμε ένα τρισδιάστατο αντικείμενο πρέπει να έχουμε και αντίστοιχα τρεις άξονες, στους οποίους πάνω θα πραγματοποιεί κίνηση το ακροφύσιο του εκτυπωτή και έτσι θα υπάρχει η ορθή εναπόθεση υλικού η οποία θα οδηγήσει στο επιθυμητό αποτέλεσμα.

Ο κάθε άξονας του συστήματος είναι τοποθετημένος σε έναν ή περισσότερους φορείς, οι οποίοι έχουν σαν τελικό σκοπό την τοποθέτηση του θερμαινόμενου ακροφυσίου στην κατάλληλη θέση, εντός του χώρου εργασίας, ώστε να να εκχυθεί κατάλληλη ποσότητα υλικού στις σωστές συντεταγμένες. Καθένας από αυτούς, διαθέτει έναν ή περισσότερους κινητήρες με τους οποίους πραγματοποιείται η διαδικασία της κίνησης από κάποιο σύστημα μετάδοσης το οποίο μετατρέπει την περιστροφική, του κινητήρα, σε γραμμική.

Ένας επιπλέον, κινητήρας χρησιμοποιείται, για την προώθηση του νήματος προς το θερμαινόμενο ακροφύσιο. Αναλόγως τον τύπο του μηχανήματος χρησιμοποιούνται οι αντίστοιχοι μέθοδοι στερέωσης και μεταφοράς του φορτίου του άξονα. Εκεί χρησιμοποιούνται γραμμικά ή ακτινικά ρουλεμάν για την ελεγχόμενη ολίσθηση πάνω τους με σκοπό να μειωθεί το φορτίο των κινητήρων όσο το περισσότερο δυνατόν.

3.1.1 Γραμμικοί Ράβδοι

Από τα πρώτα μηχανήματα που δημιουργήθηκαν μέσω του Reprap project χρησιμοποιήθηκαν γραμμικές ράβδοι και γραμμικών ρουλεμάν για την κίνηση των τριών αξόνων. Σαν καινούργια τεχνολογία, για την εποχή, δεν υπήρχαν διαθέσιμα τόσα υλικά, εξειδικευμένα, με τέτοιο τρόπο ώστε, να γίνει με κάποιον άλλο, αποδοτικότερο τρόπο, η συναρμολόγηση και λειτουργία ενός τρισδιάστατου εκτυπωτή. Έτσι λοιπόν έγινε η χρήση υλικών τα οποία ήταν ευρέως διαθέσιμα όπως οι ντίζες και οι λείοι άξονες για τον σκελετό του μηχανήματος.



Εικόνα 3.1.0
Ο άξονας X, ενός καρτεσιανού μηχανήματος

Όπως είναι λογικό, ένα τέτοιο σύστημα για να λειτουργήσει ορθά και να μειωθεί σημαντικά η τριβή που δημιουργείται κατά την κίνηση πρέπει να είναι όπως καταλλήλως λαδωμένο. Η λίπανση είναι απαραίτητη για την αύξηση της μακροζωίας του συστήματος.

Ένα τέτοιο σύστημα κίνησης επιφέρει και κάποιες δυσκολίες κυρίως στο κομμάτι της συντήρησης. Αυτό συμβαίνει διότι το λάδι, το οποίο βρίσκεται πάνω στους άξονες, μπορεί να προσελκύσει σωματίδια, όπως η σκόνη, και στην συνέχεια να την μεταφέρει εντός του ρουλεμάν, αχρηστευοντας το.

Μία λεπτομέρεια, την οποία θα πρέπει να προσέξουμε, είναι πως αυτοί οι άξονες όπως είδαμε στην παραπάνω εικόνα είναι λεπτοί συνήθως 8mm. Όσο λοιπόν ένα μηχάνημα λειτουργεί μπορεί να δεχθεί ορισμένες δυνάμεις οι οποίες, αν επιμείνουν για μεγάλο χρονικό διάστημα, θα προκαλέσουν κάποιου είδους στρεύλωση, η οποία θα εμποδίζει την ομαλή κίνηση και έτσι θα επηρεάζει αρνητικά τον κινητήρα, λόγω του φορτίου.

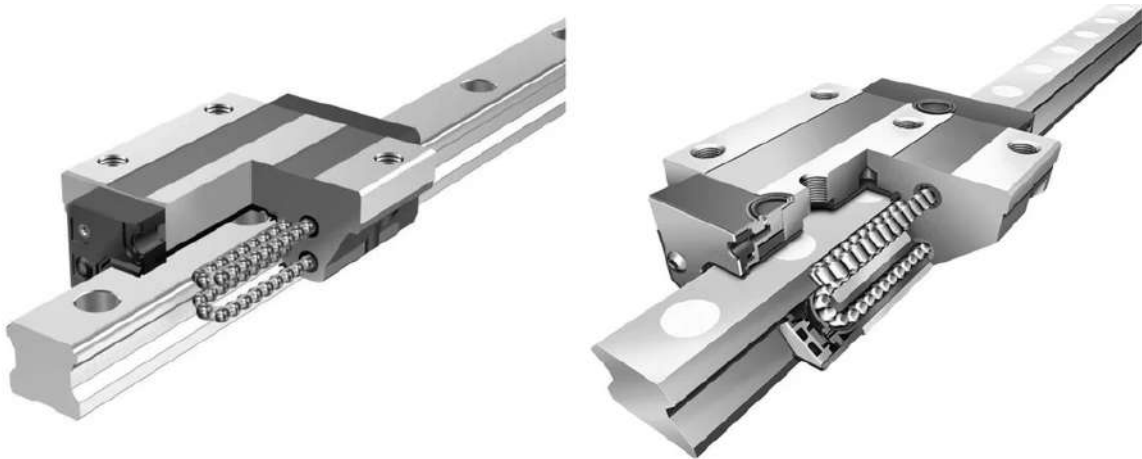


Εικόνα 3.1.1
Γραμμικό ρουλεμαν

3.1.2 Γραμμικές Ράγες

Όταν η τρισδιάστατη εκτύπωση έκανε εμφανή την παρουσία της στην αγορά άρχισαν, χάρις το RepRap κίνημα, να επινοούνται τρόποι βελτίωσης της λειτουργίας της. Αυτό σήμαινε καλύτερα και στιβαρότερα υλικά, ευρύτερη διαθεσιμότητα σημαντικών εξαρτημάτων και υλικών εκτύπωσης, όπως, και μεγαλύτερη ευκολία ως προς τη

συναρμολόγηση των κυριων μελών του σκελετού του μηχανήματος. Μία σημαντική βελτίωση η οποία επινοήθηκε για να αντιμετωπίσει τα προβλήματα που είχαν οι γραμμικοί άξονες, ήταν οι γραμμικές ράγες, οι οποίες λειτουργούσαν με παρόμοια λογική αλλά με σημαντικές βελτιώσεις. Πλέον δεν γινότανε ολίσθηση πάνω σε λείους άξονες αλλά σε ειδικά διαμορφωμένα προφίλ, όπως βλέπουμε παρακάτω :



Εικόνα 3.1.2
Γραμμικοί Άξονες

Στους συγκεκριμένους άξονες και φορείς δεν υπάρχει μεγάλο σφάλμα κίνησης του φορέας(Carriage), αντίθετα δηλαδή, από τα στρογγυλά γραμμικά ρουλεμάν τα οποία έχουν αρκετό τζόγο στην κίνηση τους. Σε αντίθεση με τους λείους άξονες όπου είναι απαραίτητη η χρήση δύο αξόνων για να παραμείνει σταθερός ο φορέας που κινεί το φορτίο του άξονα, στις γραμμικές ράγες μπορεί να χρησιμοποιηθεί μόνο μία ράγα.

Αν και καλή λύση, η οποία προσφέρει ομαλότερη και ακριβέστερη κίνηση, οι γραμμικές ράγες έχουν αυξημένο κόστος σε σχέση με τους απλούς λείους άξονες και τα στρογγυλά γραμμικά ρουλεμάν, σε σχέση με αυτά που προσφέρουν, οπότε και οι κατασκευαστές προτιμούν συνήθως την οικονομικότερη λύση.

3.1.3 V-Wheels

Το κόστος των τρισδιάστατων εκτυπωτών μειώθηκε χάρη την τυποποίηση συναρμολόγησης των επιμέρους εξαρτημάτων που το απαρτίζουν αυτό έγινε κυρίως με την χρήση προφίλ αλουμινίου τα οποία μπορούσαν να παραχθούν οικονομικά και με αρκετά μεγάλη ταχύτητα. Οπότε εκεί έπρεπε να πραγματοποιηθεί μετατροπή του τρόπου κίνησης των τριών αξόνων ώστε να εφαρμόζον πάνω σε αυτά τα προφίλ, με αποτέλεσμα να δημιουργηθούν τα V-Wheels.

Οι συγκεκριμένες ροδες κίνησης απαρτίζονται από ένα ακτινικό ρουλεμάν και ένα κέλυφος από νάιλον σε μορφή V, αντίστοιχα και ο άξονας στον οποίο κινούνται απάνω φέρει και αυτός ένα προφίλ τύπου V Με σκοπό η ρόδα να μαγκώνει εντός του προφίλ και έτσι να μην υπάρχει ο κίνδυνος ολίσθησης προς κάποια κατεύθυνση πέρα από την προκαθορισμένη.



Εικόνα 3.1.3
Εφαρμογή V-Wheel σε V-Slot aluminum profile



Εικόνα 3.1.4
Ο Φορέας του Extruder πάνω στον άξονα X

Στην εικόνα 3.1.4, παρατηρούμε την εφαρμογή του φορέα στον άξονα X του τρισδιάστατου εκτυπωτή που χρησιμοποιούμε για την συγκεκριμένη εργασία, με τα V-Wheels να είναι εφαρμοσμένα στην επιφάνεια του προφίλ.

Σε σύγκριση με τις προηγούμενες μεθόδους η συντήρηση δεν αποτελεί συχνό πρόβλημα διότι τα ρουλεμάν είναι κλειστά με ένα μεταλλικό δαχτυλίδι να καλύπτει τις μικρές σφαίρες που κινούνται στο εσωτερικό του, δηλαδή δεν έχουν άμεση επαφή με το περιβάλλον και διαθέτουν γράσο το εσωτερικό τους οπότε δεν υπάρχει περίπτωση να εισέλθουν ακαθαρσίες οι οποίες θα δυσκολέψουν την περιστροφή του ρουλεμάν και εν συνεχεία την κίνηση του άξονα. Σε σχέση με τους λείους άξονες τα προφίλ αλουμινίου είναι στιβαρά, συνήθως έχουν μέγεθος 20mm x 20mm ή 40mm x 40mm για τις συγκεκριμένες κατασκευές, με αποτέλεσμα να μην υπάρχει η ανησυχία στρεβλώσης του από κάποια καθετη ασκούμενη εξωτερική δύναμη κατά μήκος τους.

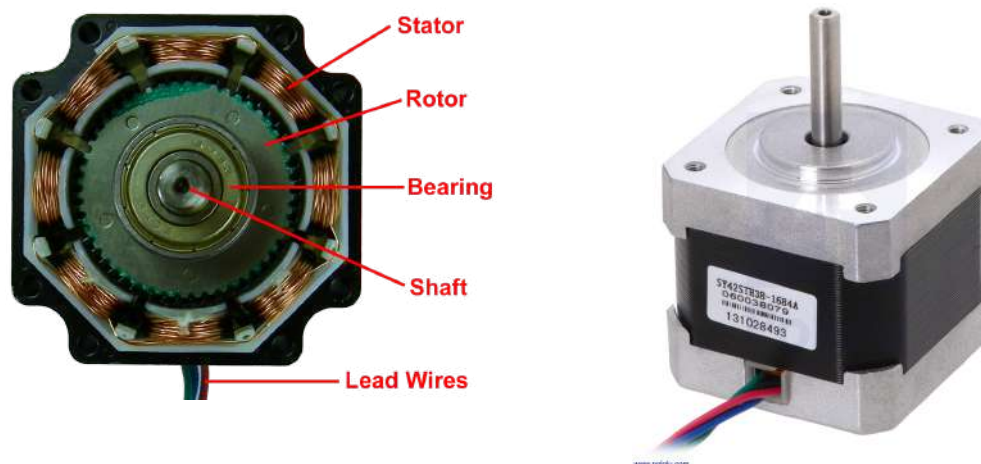
Ένα αρνητικό φαινόμενο το οποίο μπορεί να συναντήσουμε στα V-Wheels είναι πως μετά από αρκετούς κύκλους εργασίας μπορεί να υπάρξει μικρή φθορά στο καπάκι του ρουλεμάν το οποίο είναι φτιαγμένο από νάιλον και έτσι να δημιουργηθεί ένας μικρός τζόγος, αυτό βέβαια μπορεί να αντιμετωπιστεί με την αλλαγή του πλαστικού μέρους.

Ο εκτυπωτής που διαθέτουμε για τη συγκεκριμένη έρευνα διαθέτει αυτό το σύστημα κίνησης για όλους τους άξονες του.

3.2 Βηματικοί κινητήρες

Έχοντας αναλύσει προηγουμένως τους άξονες που διαθέτει ένας τρισδιάστατος εκτυπωτής και πώς πραγματοποιείται η ολίσθηση αυτών, ήρθε η ώρα μελετήσουμε πώς οι φορείς των αξόνων κινούνται στο σύστημα. Η κίνηση των αξόνων πραγματοποιείται με την βοήθεια βηματικών κινητήρων μετατρέποντας την περιστροφική τους κίνηση σε γραμμική όπως αναφέραμε και προηγουμένως.

Ο λόγος που γίνεται χρήση βηματικών κινητήρων σε ένα σύστημα όπως ένας τρισδιάστατος εκτυπωτής, είναι διότι, για την σωστή εκτέλεση του τρισδιάστατου σχεδίου το οποίο έχει σχεδιαστεί με συγκεκριμένες διαστάσεις, από κάποιο πρόγραμμα cad, πρέπει να έχει ακρίβεια κινήσεων και ροπή τόσο στην κίνηση όσο και στην συγκράτηση της θέσης. Κάτι το οποίο κινητήρες συνεχούς ρεύματος άλλης κατηγορίας δεν μπορούν να πετύχουν, όπως οι κινητήρες συνεχούς ταχείας περιστροφής ή ακόμα και κινητήρες σέρβο. Συγκεκριμένα οι κινητήρες σέρβο παρόλο που διαθέτουν έλεγχο θέσης δεν διαθέτουν άλλα χαρακτηριστικά όπως ροπή και η συγκράτηση θέσης, στοιχεία απαραίτητα, για την ορθή λειτουργία ενός τρισδιάστατου εκτυπωτή. Τα χαρακτηριστικά αυτά αξιοποιούνται κυρίως για την κίνηση στον άξονα Z. Χάρης την ροπή συγκράτησης, ο κινητήρας, έχει μεγάλη αδράνεια περιστροφής, αν η περιστροφή αυτού δεν είναι επιθυμητή, και στην δική μας περίπτωση επιθυμούμε ο άξονας Z να παραμείνει σταθερός έως ότου τελειώσει το layer που αφορά το συγκεκριμένο ύψος του χώρου εργασίας.



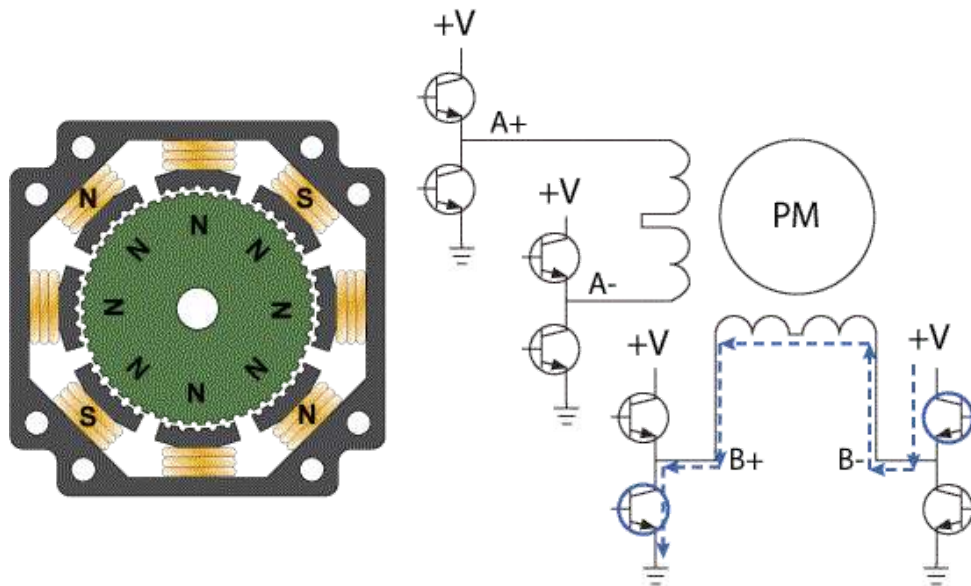
Εικόνα 3.2.1
Εσωτερικά(αριστερά) και εξωτερικά(δεξιά) ενός βηματικού
κινητήρα

Ο ρότορας του βηματικού κινητήρα φέρει μαγνητικά πολωμένες οδοντώσεις οι οποίες κατά την ενεργοποίηση ενός ζεύγους πηνίων του στάτορα, τις έλκει μέσω του μαγνητικού πεδίου που δημιουργείται. Ο έλεγχος των πηνίων, με σκοπο τον βηματισμό του κινητήρα, γίνεται μέσω ειδικών κυκλωμάτων γνωστών και ως stepper drivers. Οι περισσότεροι κινητήρες που χρησιμοποιούνται για τον έλεγχο ενός τρισδιάστατου εκτυπωτή είναι τύπου NEMA17 και ο ρότορας τους έχει 200 οδοντώσεις, που σημαίνει πως για να ολοκληρωθεί μία πλήρη περιστροφή του, πρέπει η ενεργοποίηση των πηνίων να γίνει με διαδοχικό τρόπο έτσι ώστε να γίνουν 200 βηματισμοί των 1.8° .

Γενικά, οι βηματισμοί έχουν τις εξής μορφές :

- I. Πλήρης Βηματισμός (Full-step), 1.8°
- II. Ημι-βηματισμός (Half-step), 0.9° , μειωμένη ροπή κατα 30%
- III. Μικρο-βηματισμός (Microstepping), μειωμένη ροπή κατα ακόμη 30% σε σχέση με τον Ημι-βηματισμό.

Το Microstepping αποτελεί μία καινούργια τεχνολογία η οποία χρησιμοποιείται σε μία ευρεία κατηγορία τρισδιάστατων εκτυπωτών με σκοπό την μεγαλύτερη ακρίβεια κινήσεων. Συνήθως αυτό επιτυγχάνεται με τον έλεγχο του ρευματος, που διέρχεται σε ένα ζεύγος πηνίων. Η δύναμη του μαγνητικού πεδίου να είναι αναλόγως μικρότερη, σε σχέση με το μέγιστο ρεύμα που χρησιμοποιείται, ανα φάση, για έναν πλήρη βηματισμό, έτσι δεν γίνεται πλήρης έλξη της μαγνητικά πολωμένης οδόντωσης του ρότορα οπότε και ο βηματισμός του, αποτελεί υποδιαίρεση ενός πλήρους βήματος.

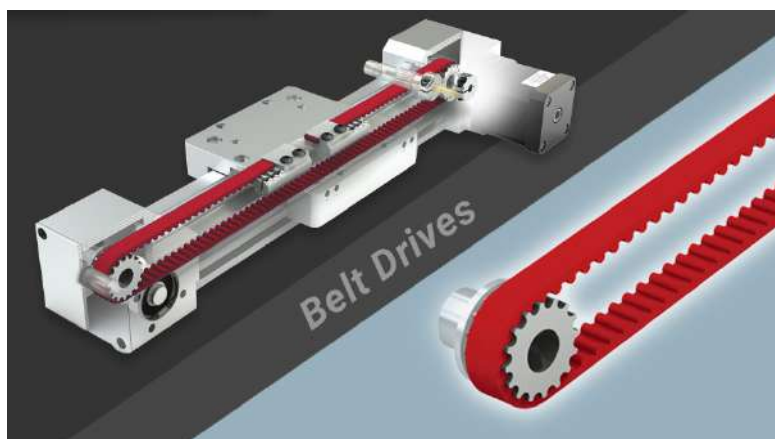


Εικόνα 3.2.2
Απεικόνιση τρόπου λειτουργίας διπολικού βηματικού κινητήρα

Για να μετατρέψουμε την περιστροφική κίνηση των κινητήρων σε γραμμική, έτσι ώστε να μπορέσουμε να δώσουμε την δυνατότητα στους φορείς των αξόνων να κινηθούν σύμφωνα με το καρτεσιανό σύστημα, πρέπει να χρησιμοποιήσουμε κάποιο είδος μετάδοσης κίνησης. ο τρόπος κατά τον οποίο γίνεται η επιλογή του τρόπου μετάδοσης κίνησης εξαρτάται από το φορτίο το οποίο καλείται ο κινητήρας να κινήσει και το κόστος κατασκευής. Μερικοί τρόποι μετάδοσης είναι :

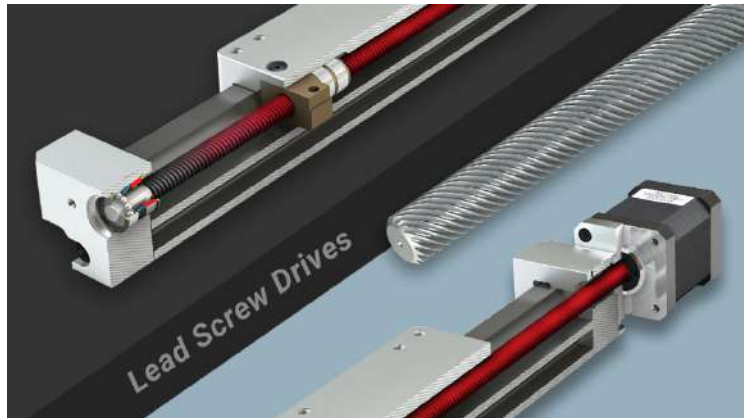
- I. Ιμάντας με οδοντώσεις (συνήθως τύπου GT2)
- II. Τραπεζοειδής Ντίζα (Σπανίως και απλές ντίζες)

Συνήθως στους άξονες X και Y χρησιμοποιούνται ιμάντες κίνησης όπου στον κινητήρα προστίθεται ένα γρανάζι οδήγησης. Αυτό είναι διαμορφωμένο με τέτοιο τρόπο ώστε να εφαρμόζουν οι οδοντώσεις του ιμάντα και έτσι να μην γλιστράει και χάνει βήματα. Οι ιμάντες χρησιμοποιούνται σε αυτούς τους άξονες επειδή δεν κινούν μεγάλο φορτίο και αποτελούν την οικονομικότερη λύση.



Εικόνα 3.2.3
Σύστημα κίνησης φορέα με την χρήση ιμάντα

Όσον αφορά τον άξονα Z, συνήθως, χρησιμοποιείτε μια ή παραπάνω τραπεζοειδείς ντίζες και τα αντίστοιχα περικόχλια τους. Αυτός κουβαλάει και τον άξονα X όπου υπάρχει το θερμαινόμενο ακροφύσιο και το σύστημα προώθησης υλικού το οποίο μπορεί να μεταφραστεί σε παραπάνω βάρος, σε σχέση με μεμονωμένους τους άξονες X και Y. Σε αυτή την περίπτωση το τραπεζοειδές προφίλ της ντίζας υποβοηθά στην συγκράτηση της θέσης των δύο αξόνων.



Εικόνα 3.2.4
Σύστημα κίνησης φορέα με την τραπεζοειδή ντίζα

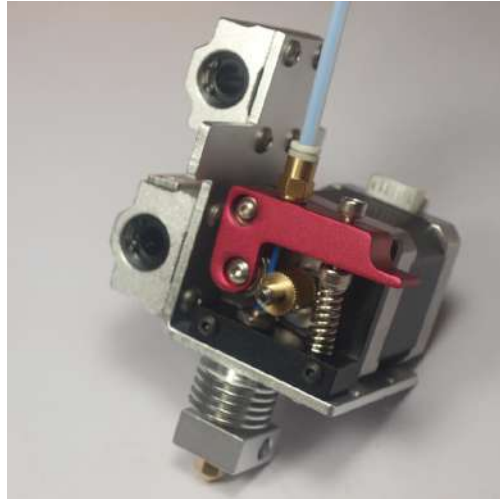
Στο μηχάνημα που χρησιμοποιούμε στη συγκεκριμένη έρευνα οι άξονες X και Y κινούνται μέσω συστήματος μάντα με την βοήθεια V-Wheels, για την ολίσθηση του φορέα κάθε άξονα, οι οποίοι κουβαλάνε το θερμαινόμενο ακροφύσιο και την θερμαινόμενη επιφάνεια εκτύπωσης αντίστοιχα. Για τον άξονα Z χρησιμοποιείται μια τραπεζοειδής ντίζα διότι, λόγω της στερέωσης των V-Wheels και των προφίλ αλουμινίου, είναι ελάχιστη η πιθανότητα γωνιακής απόκλισης, αν και οικονομική λύση, θα πρέπει να επιτηρείται συχνά μιας και οι κραδασμοί αρκετων παραγωγικών κύκλων μπορούν να λασκάρουν τις βίδες στερέωσης

3.2.1 Extruders

Για την κίνηση του υλικού, χρησιμοποιούμε επίσης έναν βηματικό κινητήρα στο οποίο τον άξονα έχουμε στερεώσει ένα μικρό γρανάζι με πολύ κοφτερές οδοντώσεις, το οποίο γραπώνει το υλικό και το προωθεί προς το ακροφύσιο. Οι προωθητές ή Extruders είναι δυο κατηγοριών.

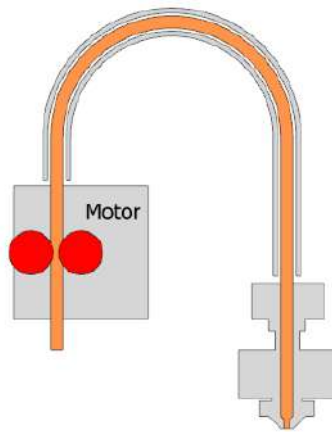
Στην πρώτη κατηγορία ανήκουν extruders οι οποίοι βρίσκονται πάνω στον φορέα του θερμαινόμενου ακροφυσίου ή Hotend. Επειδή ο βηματικός κινητήρας βρίσκεται ακριβώς από πάνω από το θερμαινόμενο άκρο, η δύναμη που απαιτείται για να προωθηθεί το υλικό είναι μικρή που σημαίνει πως υπάρχει και η δυνατότητα χρήσης μικρότερου, αρα οικονομικότερου, κινητήρα. Λόγω της ευκολότερης προώθησης, είναι εφικτή η χρήση περίπλοκων υλικών, όπως τα εύκαμπτα υλικά τα οποία είναι από τα δυσκολότερα να

εκτυπωθούν με την μέθοδο FDM. Επίσης, η ανάκληση υλικού κατά την διαδικασία της εκτύπωσης, μια χρήσιμη ενέργεια ώστε να μην υπάρχει υπερχειλίση αυτού, είναι γρηγορότερη με αποτέλεσμα τα σφάλματα τέτοιου είδους να είναι ελάχιστα. Ένα σημαντικό αρνητικό αυτής της κατηγορίας, είναι η αύξηση τους βάρους και του όγκου του φορέα του άξονα X, σε μηχανήματα όπως το δικό μας, το οποίο απαιτεί χαμηλότερη ταχύτητα κίνησης ώστε να αποφευχθούν σφάλματα στην μορφολογία του εκτυπωμένου αντικειμένου.



Εικόνα 3.2.
Direct Drive Extruder τοποθετημένο στο φορέα του
θερμαινόμενου ακροφυσίου

Η άλλη κατηγορία ονομάζεται Bowden Extruder και η κύρια διαφορά του με την προηγούμενη, είναι η τοποθέτηση του πάνω στο σκελετό του μηχανήματος και όχι πάνω από το Hotend του άξονα X. Η σύνδεση του με το θερμαινόμενο ακροφύσιο γίνεται με έναν λεπτό σωλήνα PTFE, ο οποίος είναι ανθεκτικός στις υψηλές θερμοκρασίες, που μπορεί να φτάσει το ακροφύσιο κατά την λειτουργία του, και ο ρόλος του είναι να καθοδηγήσει περιορισμένα το υλικό προς τα εκεί. Κύριο πλεονέκτημα είναι η μείωση του βάρους και όγκου, του άξονα X, το οποίο μας επιτρέπει την γρηγορότερη κίνηση του φορέα του ακροφυσίου και έτσι να ολοκληρώνονται γρηγορότερα οι εκτυπώσεις δίχως μορφολογικά σφάλματα που θα μπορούσαμε να συναντήσουμε σε ένα Direct Drive σύστημα. Μερικά από τα μειονεκτήματα αυτού του τρόπου προώθησης είναι πως, απαιτείται μεγαλύτερη ροπή για την κίνηση του υλικού προς τον Extruder, οπότε χρειάζεται να έχουμε και μεγαλύτερης ισχύος κινητήρα. Επιπροσθέτως, δύσκολα υλικά, όπως τα ευκαμπτα, που αναφέραμε και προηγουμένως είναι πάρα πολύ δύσκολο να εκτυπωθούν, οπότε αυτό αποτελεί σημαντικό περιορισμό. Αυτό είναι και το σύστημα προώθησης που χρησιμοποιείται από τον εκτυπωτή που έχουμε στην διάθεσή μας προς μελέτη.



Bowden

Εικόνα 3.2.6
Bowden Extruder

3.3 Θερμαντικά Στοιχεία

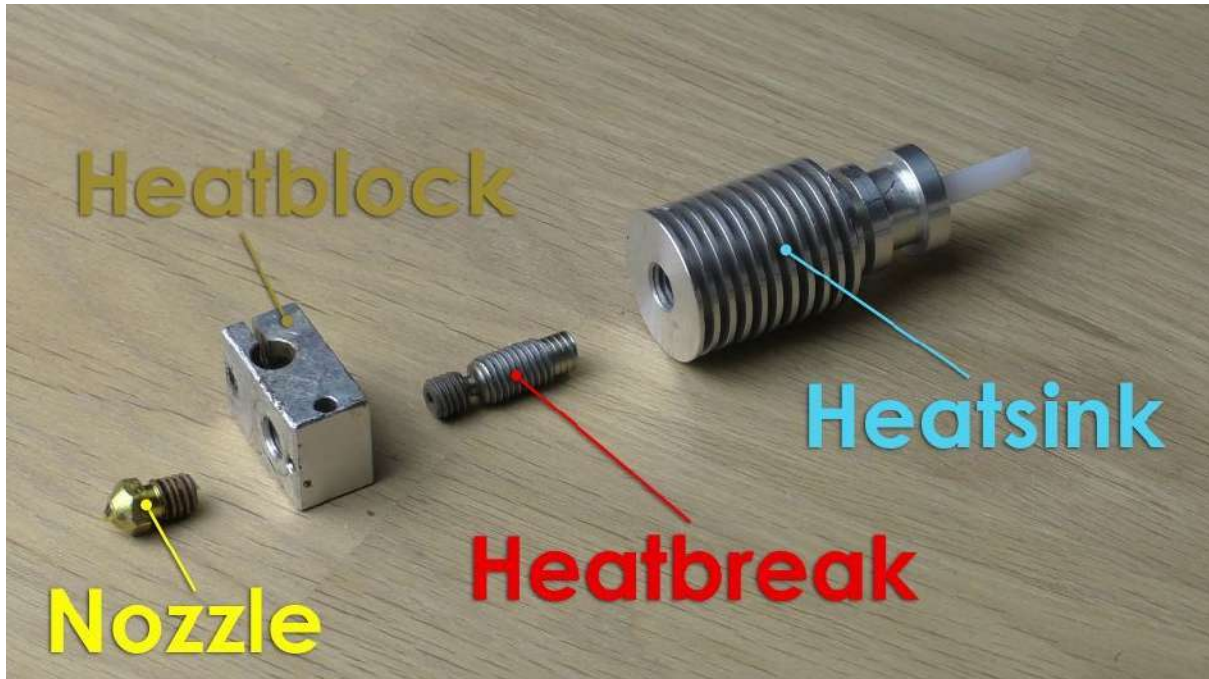
Έχοντας αναφερθεί στο κομμάτι της κινηματικής ενός τρισδιάστατου εκτυπωτή θα αναφερθούμε σε ένα άλλο κρίσιμο σημείο για την ορθή λειτουργία του, αυτή της επεξεργασίας του πλαστικού νήματος. Στους τρισδιάστατους εκτυπωτές της σημερινής αγοράς παρατηρούμε δύο θερμαντικά στοιχεία, το πρώτο είναι το θερμαινόμενο ακροφύσιο από το οποίο περνάει το υλικό για να ρευστοποιηθεί και το άλλο είναι η θερμαινόμενη επιφάνεια η οποία έχει σαν ρόλο την συγκράτηση και τη θερμική μετάβαση του τρισδιάστατου εκτυπωμένου αντικειμένου.

3.3.1 HotEnd

Το θερμαινόμενο ακροφύσιο ή αλλιώς Hotend αποτελεί το κύριο, και ίσως, το πολυπλοκότερο στοιχείο επεξεργασίας και προώθησης του στερεού πλαστικού νήματος. Στο κεφάλαιο 3 αναφερθήκαμε συνοπτικά για τα συνηθέστερα υλικά που χρησιμοποιούμε στην τρισδιάστατη εκτύπωση σήμερα. όπως είδαμε και εκεί κάθε υλικό έχει τις δικές του ιδιαιτερότητες με αποτέλεσμα να υπάρχουν διαφορές στην θερμοκρασία υγροποίησης του και στον τρόπο που αυτό προωθείται από τον extruder.

Τα κύρια στοιχεία που απαρτίζουν ένα Hotend είναι :

- I. Το ακροφύσιο - Nozzle
- II. Θερμικό μπλοκ - Heat Block
- III. Σωλήνας μεταγωγής θερμότητας - Heatbreak
- IV. Ψύκτρα - Heatsink



Εικόνα 3.3.1
Τα στοιχεία που απαρτίζουν ένα hotend

Στο Heat Block του extruder τοποθετείτε μία κεραμική αντίσταση ισχύος η οποία το θερμαίνει μαζί με το ακροφύσιο, από το οποίο θα ρέει το υλικό σε υγρή μορφή. Ο έλεγχος της θερμοκρασίας του Heat Block και εν τέλει του ακροφυσίου γίνεται μέσω ενός θερμίστορ το οποίο βιδώνεται πάνω του. Η θερμοκρασία του ακροφυσίου μπορεί να φτάσει μέχρι και τους 285°C αν και πολύ λίγα υλικά απαιτούν θερμοκρασίες υψηλότερες των 265°C. Τα nozzles που χρησιμοποιούνται συνήθως είναι φτιαγμένα από χαλκό σε περίπτωση που γίνεται όμως χρήση πιο σκληρών υλικών με προσμίξεις τότε προτιμούνται nozzles από stainless Steel ή και από ρουμπινι το οποίο είναι τοποθετημένο στο άκρο του ακροφυσίου.



Εικόνα 3.3.2
Τα διαφορετικά είδη θερμαινόμενων ακροφυσίων

Ο σωλήνας μεταγωγής θερμότητας, που βρίσκεται ενδιάμεσα του Heat Block και της ψύκτρας, αποτελεί ένα σημαντικό στοιχείο για την ομαλή κίνηση του νήματος από την ψυχρή περιοχή στην Θέρμη δίχως το υλικό να λιώνει νωρίτερα, κατά τη διαδρομή του, και έτσι να προκαλεί δυσκολίες στη ροή. Τέλος η ψύκτρα χρησιμοποιείται για να δημιουργήσει την ψυχρή περιοχή, συνήθως για την διατήρηση της ψυχρής περιοχής χρησιμοποιείτε κάποιο σύστημα ψύξης όπως ένας ανεμιστήρας.



Εικόνα 3.3.3
Συναρμολογημένος Extruder

3.3.2 Heated Bed

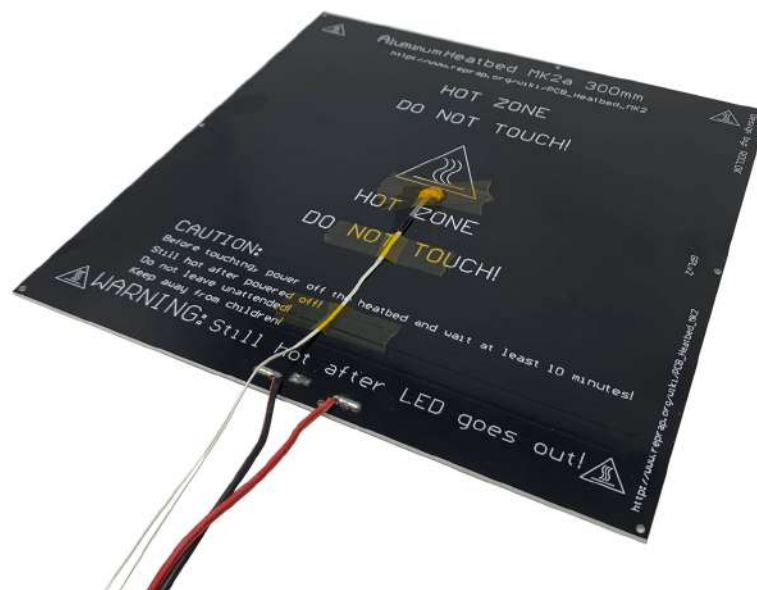
Όταν ακόμα η FDM τρισδιάστατη εκτύπωση ήταν σε πρωτότυπο στάδιο, η προσκόλληση του λιωμένου πλαστικού σε μία επιφάνεια γινόταν με την εφαρμογή κάποιου είδους αδύναμη κόλλα(σε μορφή στικ), αυτό γινόταν έτσι ώστε το εκτυπωμένο αντικείμενο να μην ξεκολλήσει από τη θέση του ενώ γίνεται εκτύπωση. Αρκετές φορές όμως λόγω μικρών σφαλμάτων στην απόσταση μεταξύ θερμαινόμενου ακροφυσίου και της επιφάνειας εκτύπωσης το αντικείμενο το οποίο σχηματιζόταν ξεκολλούσε από αυτήν σε κάποιο συγκεκριμένο σημείο ή εξ ολοκλήρου. Αυτό γινόταν, διότι, η επιφάνεια είχε θερμοκρασία, πολύ χαμηλότερη από αυτή του ακροφυσίου, το οποίο ξεπερνούσε συνήθως τους 180°C με αποτέλεσμα η ψύξη του πλαστικού να γίνεται απότομα και έτσι αυτό να συστέλλεται και να μην εφαρμόζει σωστά στην επιφάνεια.

Για να λυθεί αυτό το πρόβλημα επινοήθηκαν οι θερμαινόμενες επιφάνειες οι οποίες λόγω της θερμοκρασίας που αποκτούν μπορούν να προσφέρουν ομαλότερη ψύξη στο πλαστικό που εγχύεται από το ακροφύσιο. Έτσι, η στερεοποίηση του λιωμένου πλαστικού γίνεται

ομαλότερα χωρίς αυτό να συστέλλεται και να ξεκολλάει από την επιφάνεια. Μία θερμαινόμενη επιφάνεια εκτύπωσης Αποτελείται από δύο στελέχη :

Το πρώτο είναι μία πλακέτα PCB, η οποία είναι σχεδιασμένη ώστε να λειτουργεί σαν μία μεγάλη αντίσταση μικρής ωμικότητας

Το δεύτερο είναι μία πολύ λεπτό μεταλλικό φύλλο ώστε να συγκρατείται πιο εύκολα η θερμοκρασία αλλά και παράλληλα να προσφέρει μηχανική αντοχή την οποία δεν έχει από μόνη της η πλακέτα pcb η οποία είναι κατασκευασμένη από fiberglass τύπου FR-4.



Εικόνα 3.3.3
Θερμαινόμενη επιφάνεια εκτύπωσης

Οι θερμαινόμενες επιφάνειες έχουν επιτρέψει την χρήση πολλαπλών δευτερέων επιφανειών, που βοηθούν στην καλύτερη προσκόλληση του εκτυπωμένου αντικειμένου.

Μερικοί τύποι αυτών των είναι :

- I. Φιλμ πολυαιθεριμιδίου
- II. BuildTak
- III. Γυαλί

Τα δύο πρώτα είδη επιφανειών μπορούν να στερεωθούν πάνω σε λεπτά φύλλα Ελατηρίου όπου μπορεί να γίνεται και πιο εύκολα η αφαίρεση του τρισδιάστατα εκτυπωμένου αντικειμένου σε σύγκριση με το γυαλί. Όλες οι παραπάνω επιφάνειες που προαναφέραμε πέραν από το κομμάτι της στερέωσης προσφέρουν και ομορφότερο αποτέλεσμα ειδικά στο κομμάτι του πλαστικού που έρχεται σε επαφή με αυτές.

3.4 Mainboard

Πέρα από την μηχανική και θερμική ιδιότητα ενός τρισδιάστατου εκτυπωτή δεν μπορούμε να παραμερίσουμε το κομμάτι που ελέγχει τα πάντα, δηλαδή τον εγκέφαλο του ή αλλιώς MainBoard.

Μια MainBoard αναλαμβάνει τους ρόλους :

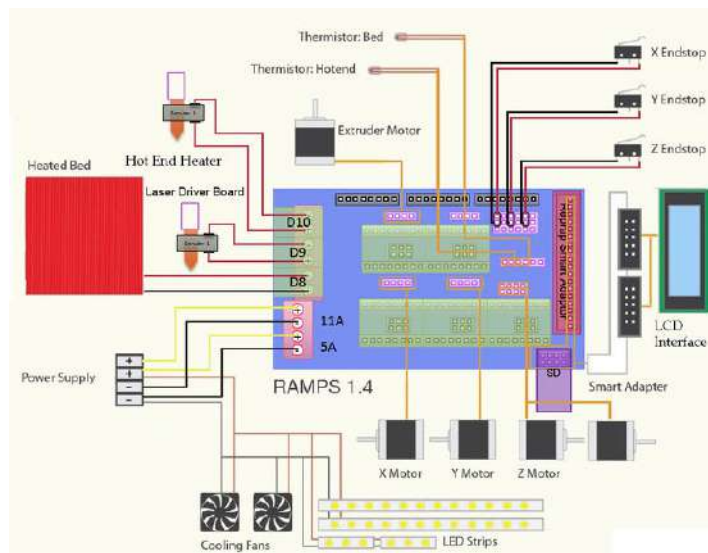
Κίνηση των αξόνων

Ελέγχου θερμαντικών στοιχείων με χρήση PID ανατροφοδότησης

Μετάφραση G-Code σε κίνηση.

Διεπαφής για τον ευκολότερο έλεγχο του συστήματος

Οι πρώτες MainBoard που χρησιμοποιήθηκαν, την εποχή που εξελισσόταν το Reprap Project, ήταν οι πλακέτες Arduino Mega, με μια επιπλέον πλακέτα η οποία έφερε θέσεις για Stepper Drivers και ειδικά Mosfet για τον έλεγχο των θερμαντικών στοιχείων. Για την λειτουργία όλων αυτών, δημιουργήθηκε το Marlin λογισμικό.



Εικόνα 3.4.1
Σχηματικό συνδεσμολογίας RAMPS 1.4

3.4.1 Marlin

Το Marlin, πρόκειται για ένα λογισμικό ανοιχτού κώδικα, το οποίο προέρχεται από παρόμοια Project το Sprinter και το GRBL, και έγινε ανεξάρτητο το 2011. Το συγκεκριμένο λογισμικό προσφέρει μια πληθώρα χαρακτηριστικών, μερικά από αυτά :

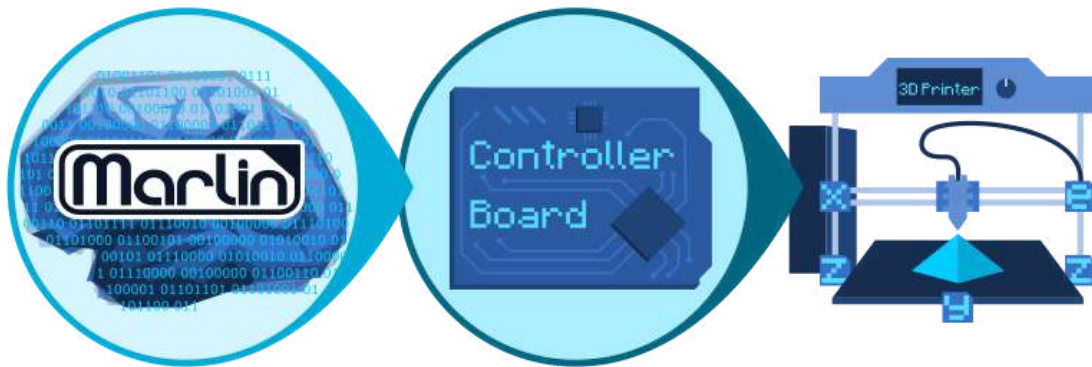
- I. Μεγάλη ποικιλία εντολών G-code.
- II. Λειτουργία τρισδιάστατων εκτυπωτών διαφόρων κινηματικών μοντέλων(cartesian, delta, corexy κ.α.)
- III. Επίβλεψη υπερθέρμανσης και άμεση διακοπή λειτουργίας
- IV. Ευθυγράμμιση επιφάνειας εκτύπωσης, μέσω λογισμικού
- V. Διεπαφή μέσω LCD

- VI. Εκτύπωση μέσω κάρτας μνήμης
- VII. Συνέχεια εκτύπωσης έπειτα από διακοπή γενικής τροφοδοσίας

Αυτό το λογισμικό χρησιμοποιείται σχεδόν από όλους τους εκτυπωτές της αγοράς, διότι η ανάπτυξη του, σαν λογισμικό ανοικτού κώδικα, γίνεται άμεσα από τους χρήστες αυτών των εκτυπωτών και χάρις την χρήση Forum μπορεί να γίνει εύκολα ο εντοπισμος σημαντικών προβλημάτων. Δεν μπορούμε να παραβλέψουμε και το γεγονός ότι, λόγω της φύσης του, έχει υπάρξει σημαντική μείωση του κόστους των τρισδιάστατων εκτυπωτών μιας και δεν αποτελεί κόστος ανάπτυξης για τις εταιρίες.

Ένα μεγάλο πλεονέκτημα αυτού του λογισμικού, είναι πως παρόλο που δημιουργήθηκε, πρωταρχικά για 8-Bit μικροελεγκτές έχει δοθεί και η δυνατότητα χρήσης ισχυρότερων εκτυπωτών 32-Bit, οι οποίοι μπορούν να φέρουν σημαντικές βελτιώσεις.

Με απλά λόγια, το λογισμικό αναλαμβάνει την μετάφραση της γλώσσας G-Code σε κίνηση και εν τέλη ένα εκτυπωμένο αντικείμενο.



Εικόνα 3.4.1
Λογισμικό ανοικτού κώδικα Marlin

3.4.2 G-Code

Η G-Code είναι μία προγραμματιστική γλώσσα η οποία χρησιμοποιείται ευρέως σε μηχανές CNC. G-code σημαίνει γεωμετρικός κώδικας(Geometric code) και τον χρησιμοποιούμε για να υποδείξουμε σε ένα μηχάνημα τι να κάνει ή πώς να κάνει κάτι. Αυτό συμπεριλαμβάνει τον τρόπο τον οποίο θα κινηθεί γεωμετρικά το άκρο εργασίας, στην δική μας περίπτωση το θερμαινόμενο ακροφύσιο, με ποια ταχύτητα και πόσο υλικό πρέπει να εναποθέσει ανά στρώση έως ότου ολοκληρωθεί το τρισδιάστατο αντικείμενο.

G## X## Y## Z## F## E##

Εικόνα 3.4.3
G-Code Format

Όπως βλέπουμε και από την παραπάνω εικόνα το format μιας εντολής g-code Είναι πολύ απλό. Αρχικά προηγείται η γενική εντολή G, αναλαμβάνει όλες τις εντολές που έχουν να κάνουν με την κίνηση του μηχανήματος. Έπειτα τοποθετούμε τις συντεταγμένες κινήσεις των αξόνων μετά από τις εντολές X, Y, Z, όπου το άκρο εργασίας θα κινηθεί ανάλογα την εντολή G που έχει δοθεί. Για παράδειγμα έχει προηγηθεί η εντολή G90, τότε, ότι συντεταγμενες δοθούν για τους άξονες X,Y,Z η κίνηση θα γίνει στο απόλυτο σύστημα συντεταγμένων από την αρχή των αξόνων(Home position), αλλιώς αν έχει προηγηθεί εντολή G91, τότε η κίνηση θα γίνει από το σημείο που βρίσκεται το άκρο εργασίας. Οι τιμές που δίνονται στην εντολή F, αφορούν την ταχύτητα κίνησης που θα πραγματοποιηθεί ώστε το ακρο εργασίας να φτάσει στις συγκεκριμένες συντεταγμένες που έχουν δοθεί για τους τρεις προηγούμενους άξονες. Τέλος, η παράμετρος E αφορά το υλικό, δηλαδή πόσο θα πρέπει να έχει προχωρήσει ανα κάθε εντολή κίνησης των αξόνων.

```

;Sliced by ideaMaker 3.6.0.4300, 2021-01-10 13:27:10 UTC+0200
;Dimension: 230.000 230.000 250.000 0.400
;Plate Shape: 0
;Extruder Offset #1: 25.000 0.000
;Filament Diameter #1: 1.750
;Filament Diameter #2: 1.750
;Filament Compensation #1: 96.00
;Filament Compensation #2: 100.00
;Filament Density #1: 1240.00
;Filament Density #2: 1240.00
;Bounding Box: 80.866 150.654 61.142 168.857 0.000 12.000
;Model Gap: 0.000
M221 T0 S96.00
M104 T0 S220.00
M109 T0 S220.00
T0
G21
G90
M82
M107
G28 X0 Y0
G28 Z0
G1 E15.0 F7200.0
G92 E0
G1 F140 E29
G1 X20 Y0 F140 E15
G92 E0
G1 F7200.0
M117 Printing...
;end of Start GCode
;LAYER:0
;Z:0.300
;HEIGHT:0.300
M106 S0
M140 S60
G1 F2700 E-5.7500
G0 F7200 X82.451 Y62.864
G0 F3600 Z0.300
;TYPE:SKIIRT
;WIDTH:0.400
G1 F2100 E0.0800
G1 F1500 X82.755 Y62.523 E0.1023
G1 X83.367 Y62.019 E0.1411
G1 X83.753 Y61.766 E0.1637
G1 X84.484 Y61.437 E0.2029
G1 X85.130 Y61.254 E0.2357
G1 X85.724 Y61.165 E0.2651
G1 X86.100 Y61.142 E0.2835
G1 X140.027 Y61.154 E2.9201
G1 X140.710 Y61.223 E2.9537

```

Εικόνα 3.4.4
Παράδειγμα G-Code

3.4.3 3D Model Processing

Αφού αναφερθήκαμε γενικά για τα στοιχεία που απαρτίζουν έναν τρισδιάστατο εκτυπωτή θα ασχοληθούμε με τα αρχεία που χρειάζεται ένας τρισδιάστατος εκτυπωτής για να εκτελέσει ένα τρισδιάστατο σχέδιο.

Τα μοντέλα που έχουμε σκοπό να εκτυπώσουμε, αρχικά θα πρέπει να τα σχεδιάσουμε με κάποιο εργαλείο CAD, τα πιο γνωστά του χώρου της τρισδιάστατης εκτύπωσης, σήμερα, είναι το solidworks και το Fusion360, χωρίς αυτά να είναι οι μόνες επιλογές. Αφού, ολοκληρώσουμε το επιθυμητό μας σχέδιο με τις διαστάσεις που έχουμε ορίσει πρέπει να

αποθηκεύσουμε το σχέδιο σε αρχείο τύπου STL, το οποίο μας είναι γνώριμο από το κεφάλαιο της εισαγωγής. Ο STL τύπος αρχείου ή αλλιώς Standard Triangle Language, είναι είναι ο πιο ευρέως χρησιμοποιούμενος στον κόσμο της ταχείας πρωτοτυποποίησης προσθετικής μεθόδου. Κύριο χαρακτηριστικό, όπως περιγράφεται και από το όνομα του τύπου, είναι πως το τρισδιάστατο αρχείο περιγράφεται μορφολογικά μέσω εμπλεκόμενων τριγώνων ώστε να υπάρξει μεγαλύτερη ευελιξία στην μετάφραση περίπλοκων γωνιών με κλίση και σχημάτων.

Αφού σχεδιαστεί το μοντέλο και αποθηκευτεί σε τύπο .STL τότε παραμένει ένα στάδιο πριν την υλοποίηση του σχεδίου από τον τρισδιάστατο εκτυπωτή, αυτό του Slicing. Ένα Slicing πρόγραμμα αναλαμβάνει να επεξεργαστεί το τρισδιάστατο μοντέλο ώστε να το μετατρέψει σε εντολές, κατανοητές από έναν τρισδιάστατο εκτυπωτή, για να χτίσει το μοντέλο στρώση προς στρώση. Οι εντολές αυτές είναι γραμμένες σε G-code όπως αναφέραμε και προηγουμένως.

Μέσα σε ένα slicing software μπορούμε να παραμετροποιήσουμε τις ρυθμίσεις με τις οποίες θα πραγματοποιηθεί η εκτύπωση, μερικές από αυτές είναι :

- I. Ταχύτητα εκτύπωσης
- II. Θερμοκρασία των θερμαντικών στοιχείων
- III. Ροή υλικού
- IV. Τοποθέτηση και παράμετροι υποστηρικτικού υλικού(Supports)
- V. Παραμετροποίηση ποσοστού και τύπου γεμίματος(Σε περίπτωση που για λόγους οικονομίας δεν θέλουμε να το κάνουμε μασιφ)
- VI. Ρυθμός ψύξης υλικού



Εικόνα 3.4.5
Ροή επεξεργασίας ενός τρισδιάστατου μοντέλου προς
εκτύπωση

Κεφάλαιο 4

Ανάλυση απαιτήσεων εφαρμογής

Σε αυτό το κεφάλαιο, θα αναλύσουμε περισσότερο, την κύρια ιδέα της εφαρμογής, την οποία προαναφέραμε συνοπτικά. Έχοντας αναλύσει τα κύρια στοιχεία με τα οποία θα ασχοληθούμε για την ανάπτυξη της συγκεκριμένης εφαρμογής, θα ορίσουμε τις απαιτήσεις που θέλουμε να έχει το σύστημα μας, ορίζοντας έτσι, μια πορεία ανάπτυξης της ώστε να είναι παράλληλα λειτουργική αλλά και εύκολη ως προς την χρήση.

4.1 Απαιτήσεις Σχεδιασμού

Η εφαρμογή που θα αναπτύξουμε στα επόμενα κεφάλαια, καλείται να αντιμετωπίσει σημαντικά παραγωγικά προβλήματα στα οποία βρίσκονται αντιμέτωπες οι φάρμες τρισδιάστατων εκτυπωτών τύπου FDM.

Για μεγαλύτερη ευκολία, θα αναφερθούμε πρώτα για τα συνηθή παραγωγικά προβλήματα ή βλάβες που μπορεί να αντιμετωπίσει ένας εκτυπωτής της συγκεκριμένης τεχνολογίας. Όντας ένα μηχάνημα με πολλά ηλεκτρομηχανικά μέρη, υπάρχουν πολλά σημεία εξέτασης όσον αφορά τις βλάβες που μπορούν να προκύψουν, συνοπτικά αυτές μπορεί να είναι :

- I. Στον έλεγχο θερμοκρασίας του ακροφυσίου
- II. Στον έλεγχο θερμοκρασίας της επιφάνειας εκτύπωσης
- III. Στην κίνηση του υλικού μέσα από το θερμαινόμενο ακροφύσιο(Clogging)
- IV. Μηχανική στρέβλωση
- V. Μηχανική χαλάρωση ενώσεων λόγω δονήσεων λειτουργίας

Σαν πρώτη εικόνα μπορούμε να παρατηρήσουμε πως οι βλάβες που μπορούν να προκύψουν μπορεί να είναι μικρές με αντίκτυπα την δημιουργία μικρο-σφαλμάτων στα παραγόμενα προϊόντα έως και σοβαρά προβλήματα, όπως η εκδήλωση πυρκαγιάς.

Το σοβαρότερο πρόβλημα, δηλαδή η πυρκαγιά από τρισδιάστατο εκτυπωτή, μπορεί να προκύψει από κάποιο χαλασμένο Thermistor με αποτέλεσμα αν ο εκτυπωτής βρίσκεται σε λειτουργία παραγωγής, να μην δέχεται σωστή επιστροφή στον PID ελεγκτή του, και ολοένα να αυξάνει την θερμοκρασία, το οποίο υπο κακές συνθήκες θα μπορούσε να προκαλέσει σπινθηρισμό. Βέβαια, αν και πλέον οι περισσότεροι εκτυπωτές έχουν την λειτουργία του Thermal Shutdown, δηλαδή της διακοπής λειτουργίας σε περίπτωση μη σωστής απόκρισης της θερμοκρασίας σε ένα μικρό χρονικό διάστημα. Υπάρχουν ακόμα πολλά μηχανήματα τα οποία δεν φέρουν αυτή την ρύθμιση, οπότε καλούμαστε να είμαστε προσεκτικοί ώστε η εφαρμογή μας να επιβλέπει όλες τις περιπτώσεις.

Ένα σημαντικό στοιχείο για την ορθή λειτουργία ενός τρισδιάστατου εκτυπωτή είναι η ομαλή ροή του υλικού μέσα από το θερμαινόμενο ακροφύσιο, ώστε να μπορεί να παραχθεί ένα αντικείμενο το οποίο δεν υπάρχει περίπτωση να φέρει προβλήματα στην μορφολογία του και ακολουθεί πιστά την σχεδίαση που έχει πραγματοποιηθεί σε προηγούμενο στάδιο πριν την έναρξη της παραγωγής. Πολλές φορές μπορεί να εμφανιστούν δυσκολίες στην

ροή(Filament Clogs), υπάρχει περίπτωση να μην παρατηρηθεί κατά την έναρξη της διαδικασίας εκτύπωσης αλλά μετά από κάποιο χρονικό διάστημα, αυτό συναντάται όταν:

- I. Οι ρυθμίσεις ροής δεν είναι σωστές,
- II. Η θερμοκρασία του ακροφυσίου δεν είναι σωστή, μπορεί να βρίσκεται πολύ ψηλά ή πολύ χαμηλά,
- III. Το υλικό που χρησιμοποιείται υπέστη αλίωση, συνήθως λόγω της υγρασίας του χώρου έπειτα από κάποιο χρονικό διάστημα έκθεσης του,
- IV. Το ακροφύσιο έχει βουλώσει λόγω κάποια ακαθαρσίας,
- V. Υπάρχει βλάβη στον extruder, ο οποίος προωθεί το υλικό.

Από τα παραπάνω καταλαβαίνουμε πόσο σημαντική είναι η επίβλεψη της ροής του υλικού για να μπορέσει το μηχάνημα να βγάλει ένα αποδεκτό αποτέλεσμα μιας και μπορούν να φανερωθούν αρκετά σημεία βλάβης τα οποία χρήζουν άμεση επισκευή.

Μιλώντας για μια φάρμα εκτυπωτών δεν μπορούμε να παραβλέψουμε και τους οικονομικούς παράγοντες κέρδους και σφάλματος. Κύρια εξοδα για μια τέτοια παραγωγική μονάδα είναι :

- I. Υλικά εκτύπωσης, σε μορφή νήματος,
- II. Ώρες λειτουργίας των μηχανημάτων(Uptime),
- III. Κατανάλωση ρεύματος,
- IV. Εξαρτήματα για επισκευές βλαβών

Καταλαβαίνουμε, λοιπόν, πως είναι απαραίτητο ένα σύστημα μέτρησης Uptime και κατανάλωσης υλικού, αλλά και επίβλεψης εναπομειναντως ποσοτητας του εκάστοτε καρουλιού. Η επίβλεψη της κατάστασης λειτουργίας ενός εκτυπωτή αποτελεί αναγκαία για τον εντοπισμό ανωμαλιών κατα την λειτουργία, που θα μπορούσαν να οδηγήσουν σε πρόωρο εντοπισμό πηγής σφάλματος και έτσι να μειώσει τόσο τον χρόνο που ένα μηχάνημα θα μείνει ανενεργό λόγω βλάβης αλλά και την έγκαιρη επισκευή ενός σημείου που θα μπορούσε να προκαλέσει ακόμη μεγαλύτερη οικονομική ζημιά.

Μια παραγωγική μονάδα, όμως, λόγω του όγκου και τον αριθμό μηχανημάτων προς έλεγχο και λειτουργία, χρειάζεται μια διεπαφή στην οποία συγκεντρώνονται όλες οι πληροφορίες που είναι απαραίτητες για την λειτουργία της. Αυτή η διεπαφή, θα πρέπει να είναι απλή και εύκολη ως προς την χρήση, ενώ παράλληλα να διαθέτει σενάρια κινδύνου στα οποία θα αναλαμβάνει να ειδοποιήσει άμεσα σε περίπτωση που κάποιο από αυτά γίνει αντιληπτό.

4.2 Απαιτήσεις Αισθητηρίων

Για να μπορούμε να αντιλαμβανόμαστε την κατάσταση του τρισδιάστατου εκτυπωτή κατα την λειτουργία του πρέπει να αξιοποιήσουμε ορισμένα αισθητήρια ηλεκτρονικά ή μηχανικά ώστε να μπορέσουμε να διακρίνουμε σφάλματα και καταστάσεις.

Αρχικά, μιας και χρειάζεται να εντοπίζουμε αδυναμίες ροής του υλικού πρέπει να σκεφτούμε έναν τρόπο να καταλαβαίνουμε πότε αυτό συμβαίνει. Όταν ένα υλικό δεν μπορεί να περάσει από το ακροφύσιο, δημιουργεί μια δυσκολία στον βηματικό κινητήρα προώθησης. Ο κινητήρας όπως γνωρίζουμε φέρει ένα γρανάτζι με οδοντώσεις για την

προώθηση του υλικού, όταν το υλικό δεν κινηθεί ενώ ο βηματικός προσπαθεί να πραγματοποιήσει έναν βηματισμό, υπάρχει περίπτωση να γλιστρήσει, δηλαδή να χάσει το βήματος και έτσι να ακουστεί ένας χαρακτηριστικός γδούπος. Όταν χάνεται ένα βήμα, κινητήρας δονείται, λόγω της αντίστασης που δέχτηκε, σε μεγαλύτερη συχνότητα, όμως επειδή το σύστημα μας προορίζεται για φάρμα εκτυπωτών ο ήχος, που παράγεται από τον κινητήρα, δεν είναι εύκολα διακριτός, οπότε θα πρέπει να βρούμε τρόπο να αντιλαμβανόμαστε τις δονήσεις λειτουργίας και αδυναμίας προώθησης με κάποιο άλλο μέσο. Για να το καταφέρουμε αυτό θα πρέπει να χρησιμοποιήσουμε ένα επιταχυνσιομετρο τύπου MEMS(Micro-electromechanical system) ώστε να μπορέσουμε να έχουμε μετρήσεις κίνησης και ως προς τους τρεις άξονες.

Αν κοιτάξουμε τα Datasheet ορισμένων βηματικών κινητήρων θα παρατηρήσουμε πως η θερμοκρασία περιβάλλοντος είναι από $-10\text{ }^{\circ}\text{C}$ έως και $50\text{ }^{\circ}\text{C}$ και η μέγιστη θερμοκρασία λειτουργίας είναι $85\text{ }^{\circ}\text{C}$, ο λόγος που έχουμε αυτόν τον περιορισμό είναι επειδή υπάρχει περίπτωση να χαλάσει η μόνωση του χαλκού εντός του μοτερ με αποτέλεσμα να υπάρξουν βραχυκυκλώματα μεταξύ των σπειρωμάτων και έτσι να μην λειτουργεί σωστά. Οπότε, καλό θα ήταν με το σύστημα μας να μπορούμε να παρακολουθούμε την θερμοκρασία και να ειδοποιούμε σε περίπτωση κάποιας κρίσιμης τιμής.

| | |
|---------------------|-----------------------|
| Temperature Rise | 85°C Max.(2 phase on) |
| Ambient Temperature | -20°C~+50°C |

Εικόνα 5.1.1
Θερμοκρασία περιβάλλοντος και θερμοκρασία λειτουργίας βηματικού κινητήρα

Τέλος, θα πρέπει να μπορούμε να μετρήσουμε άμεσα το υλικό που χρησιμοποιείται κατα την διάρκεια μιας εκτύπωσης. Για να το καταφέρουμε αυτό, θα χρησιμοποιήσουμε έναν Incremental Rotary Encoder ο οποίος θα πρέπει να εφαρμόζει πάνω στον άξονα του μοτέρ με σκοπό να μετράει τις περιστροφές που κάνει και έπειτα με μια απλή μαθηματική έκφραση να βρίσκουμε πόσο υλικό έχει καταναλωθεί.

4.3 Απαιτήσεις Δεδομένων Εκμάθησης

Για την αναγνώριση ανώμαλης κατάστασης(Anomaly Detection), θα χρησιμοποιήσουμε ένα μοντέλο μηχανικής μάθησης, έτσι θα καταφέρουμε να εντοπίζουμε πότε υπάρχει αδυναμία ροής. Ένα τέτοιο μοντέλο για να μπορεί να κάνει ορθή αναγνώριση πρέπει να εκπαιδευτεί με κάποιου είδους δεδομένα. Τα δεδομένα που θα χρησιμοποιήσουμε για να τροφοδοτήσουμε το μοντέλο μας, θα είναι μετρήσεις από το επιταχυνσιομετρο το οποίο θα είναι στερεωμένο πάνω στο μοτερ που κινεί το υλικό.

Για να αποκτήσουμε αυτά τα δεδομένα θα πρέπει να πραγματοποιήσουμε μερικές εκτυπώσεις, μερικές από αυτές θα πρέπει να έχουν τις ορθές ρυθμίσεις, στις οποίες ξέρουμε πως δεν θα υπάρχει αδυναμία ροής του υλικού, ενώ μερικές άλλες θα πρέπει να μην έχουν τις σωστές ρυθμίσεις ώστε να προκαλέσουμε επίτηδες αδυναμία ροής και έτσι να υπάρχουν στοιχεία στα δεδομένα μας τα οποία θα εκφράσουν ανώμαλη κατάσταση.

Περισσότερες λεπτομέρειες, όσον αφορά τα δεδομένα που θα χρησιμοποιήσουμε και τι είδους επεξεργασία θα τους υποβάλλουμε θα αναφέρουμε σε επόμενο κεφάλαιο.

4.4 Απαιτήσεις Διασύνδεσης και Διεπαφών

Το σύστημα μας, αφού αναγνωρίσει κάποιο σφάλμα πρέπει να το γνωστοποιήσει με κάποιον τρόπο στον τελικό χρήστη του συστήματος. Για να γίνει αυτό πρέπει να ορίσουμε τον τρόπο κατά τον οποίο οι ελεγκτές των μηχανημάτων θα επικοινωνούν με έναν κεντρικό διακομιστή ο οποίος θα έχει την ευθύνη να προβάλλει αυτά τα δεδομένα και να διαχειρίζεται τα σφάλματα που εντοπίζονται από τα αισθητήρια.

Μιας και κάθε εκτυπωτής θα έχει τα δικά του αισθητήρια, θα πρέπει να χρησιμοποιήσουμε έναν τρόπο συγκεντρώσεως των μετρήσεων τους στον διακομιστή. Δημιουργώντας ένα τοπικό δίκτυο, και αξιοποιώντας το πρωτόκολλο MQTT(Message Queuing Telemetry Transport), θα αποστέλλουμε όλες τις μετρήσεις μας ώστε να υπάρξει μετέπειτα επεξεργασία και διαχείριση.

Αφού γίνει η συλλογή των δεδομένων, θα πρέπει αυτά να φιλτραριστούν, ώστε να βρούμε αν υπάρχει κάποιο σφάλμα. Με τον εντοπισμό σφάλματος, θέλουμε έναν άμεσο τρόπο να ενημερώνεται κάποιος υπεύθυνος ώστε να δράσει ανάλογα και έτσι να αποφευχθεί κάποιου είδους βλάβη ή ελαττωματική εκτύπωση. Η ενημέρωση θα πραγματοποιείται με αποστολή μηνύματος(SMS), μέσω μιας συσκευής GSM, ώστε αυτή να γίνεται εγκαίρως ακόμη και σε περίπτωση που δεν παρακολουθείται η διεπαφή.

Η διεπαφή που θα προβάλλει τα δεδομένα και τις ανωμαλίες του συστήματος, πρέπει να είναι σχεδιασμένη έτσι ώστε να γίνεται άμεσα αντιληπτή μια αρνητική κατάσταση, για αυτό πρέπει να γίνει και η απαραίτητη χρήση χρωματικού κώδικα για την άμεση αναγνώριση. Η προβολή όλων αυτών, θα πρέπει να είναι προσβάσιμη από μια πληθώρα συσκευών τόσο εντός του τοπικού δικτύου όσο και από κάποια εξωτερική συσκευή, οπότε η διεπαφή μας θα φιλοξενείται από μια ιστοσελίδα, που θα λειτουργεί εντός τους διακομιστή.

Κεφάλαιο 5

Συλλογή δεδομένων Μέσω Μικροελεγκτή

Το σύστημα μας, όπως αναφέραμε προηγουμένως, πρέπει να δέχεται σαν είσοδο τα δεδομένα ορισμένων αισθητηριών ώστε να γίνουν αντιληπτές μερικές ενέργειες ή συνθήκες. Για να επιτύχουμε την παραμετροποίηση των αισθητηριών και την λήψη των δεδομένων που μας παρέχουν, πρέπει να χρησιμοποιήσουμε έναν μικροελεγκτή, ο οποίος με την σειρά του θα στέλνει σειριακά τα δεδομένα ώστε να γίνει περαιτέρω επεξεργασία τους μέσω του μοντέλου μηχανικής μάθησης, το οποίο θα εξηγήσουμε περισσότερο στο επόμενο κεφάλαιο.

Μιας και το σύστημα που φτιάχνουμε βρίσκεται σε πρωτότυπο στάδιο, πρέπει να ορίσουμε τι είδους μικροελεγκτή θα χρησιμοποιήσουμε, σε τι γλώσσα θα γράψουμε τον κώδικα μας, τι περιφερειακά θα χρειαστούν για την ορθή του λειτουργία και εν τέλη θα πρέπει να σχεδιάσουμε και μια πλακέτα(PCB) ώστε όλα αυτά τα υλικά να συγκεντρωθούν. Για να κάνουμε τις πρώτες δοκιμές μας, αν το σύστημα τελικά μπορεί να υλοποιηθεί, αρεστό θα ήταν να χρησιμοποιήσουμε μια οικονομική πλακέτα προτυποποίησης και η επιλογή μας θα είναι ένα Arduino uno.

Επιπροσθέτως, στο συγκεκριμένο κεφάλαιο θα αναφέρουμε όλα τα αισθητήρια που θα χρησιμοποιήσουμε, θα σχηματίσουμε τον κώδικα σύμφωνα με τον οποίο θα συλλέγουμε τις πληροφορίες και έτσι θα ξεκινήσουμε το πρώτο σκέλος του συστήματος μας.

5.1 Arduino

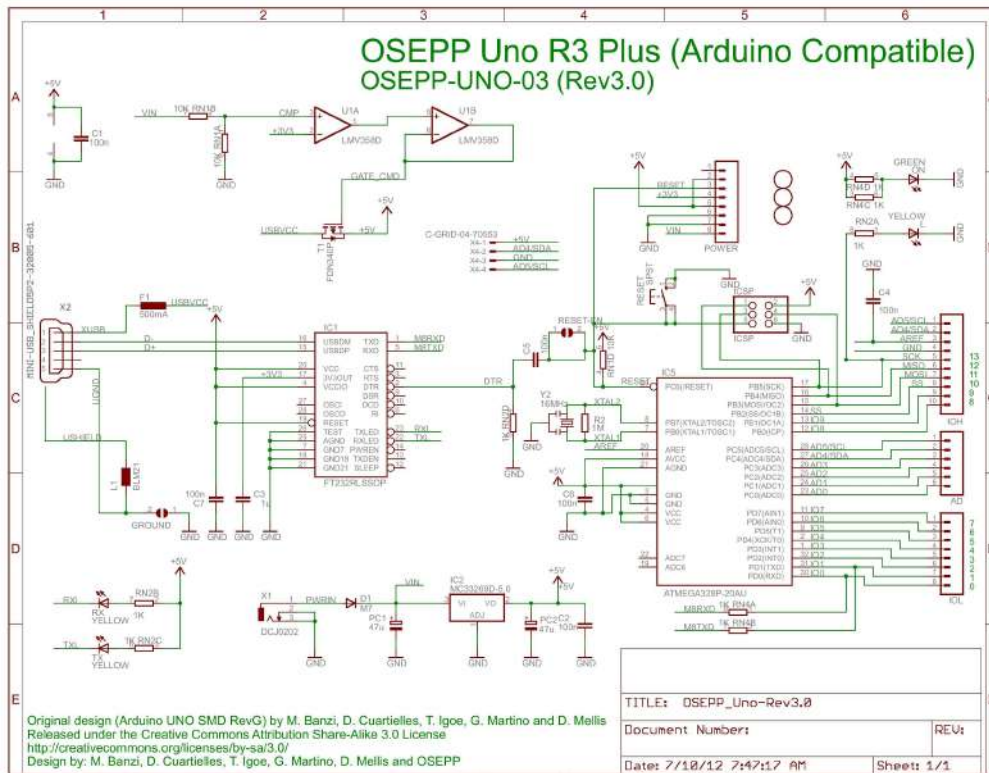
Για να μπορέσουμε να πάρουμε δεδομένα από τα αισθητήρια μας και να μεταφέρουμε στο μοντέλο μηχανικής μάθησης, θα χρησιμοποιήσουμε την πλατφόρμα ανοικτού κώδικα Arduino, η οποία βασίζεται σε υλικά και λογισμικό εύκολης χρήσης. Με λίγα λόγια, μιας και πρόκειται για μια πρωτοβουλία ανοικτού κώδικα, θα μπορούμε πολύ εύκολα να βρούμε βιβλιοθήκες για τον κώδικα μας και εξαρτήματα/αισθητήρια τα οποία είναι συμβατά μειώνοντας σημαντικά τον χρόνο και το κόστος πρωτοτυποποίησης του συστήματος.

Στην δική μας περίπτωση θα χρησιμοποιήσουμε την πλακέτα Arduino Uno, μιας και είναι μια πλακέτα με αρκετές δυνατότητες, κατάλληλες για την χρήση που θέλουμε να κάνουμε. Η συγκεκριμένη πλακέτα βασίζεται στον ATmega328p ο οποίος είναι ένας 8-bit μικροελεγκτής της εταιρίας ATMEL. Ο μικροελεγκτής, χρονίζεται με την βοήθεια κρυστάλλου στα 16MHZ, όπου και δίνετε η δυνατότητα στους χρήστες να μην τον αξιοποιήσουν και να χρησιμοποιήσουν τον εσωτερικό χρονισμό του σε περίπτωση που αυτό το απαιτεί η εφαρμογή τους. Η τροφοδοσία της πλακέτας μπορεί να γίνει με τρεις τρόπους :
Απευθείας τροφοδοσία μέσω USB,
Χρησιμοποιώντας την φύσα(Barrel jack) με μέγιστη προτεινόμενη τάση τα 12V,
Συνδέοντας το Vin της πλακέτας με μέγιστη τάση εισόδου τα 12V



Εικόνα 5.1.1
Arduino Uno

Αξίζει να σημειωθεί, πως η πλακέτα φέρει κύκλωμα διαχείρισης τροφοδοσίας σε περίπτωση που υπάρχουν περισσότερες από μια μέθοδοι τροφοδοσίας. Σύμφωνα με το Datasheet για το ATmega328p, οι εισοδοι και οι εξοδοι του, έχουν όριο τα 40mA, κάτι το οποίο καλό θα ήταν να προσέξουμε σε περίπτωση που το αισθητήριο ή η συσκευή που ελέγχουμε χρειάζεται μεγαλύτερη τροφοδότηση ρεύματος για να λειτουργήσει ορθά. Ας δούμε, λοιπόν, και το ολοκληρωμένο κύκλωμα της πλακέτας :



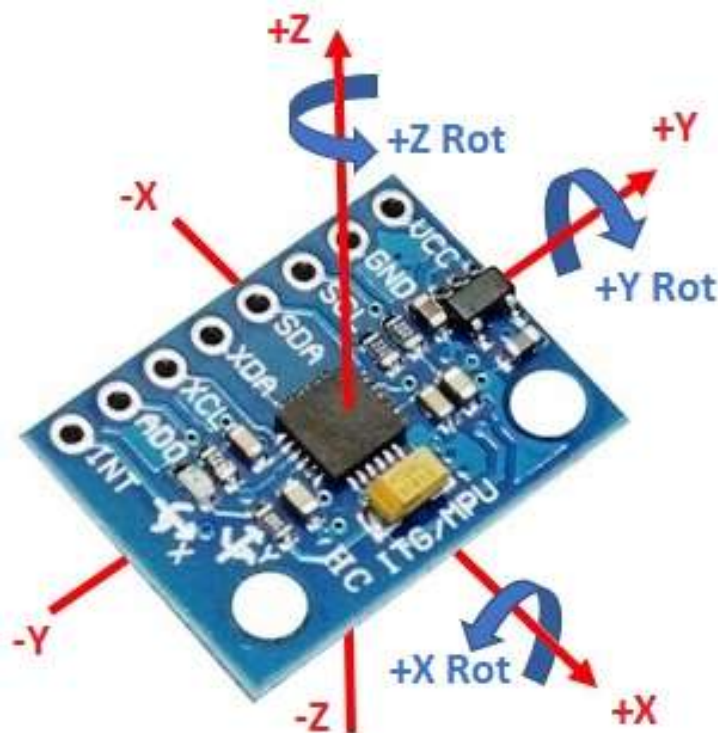
Εικόνα 5.1.2
 Το σχηματικό μιας πλακέτας Arduino Uno

Τέλος, από πλευράς Hardware, ένα στοιχείο που μας εξυπηρετεί είναι η δυνατότητα του να μεταφέρει δεδομένα σειριακά σε κάποιον υπολογιστή ή κάποια USB Host συσκευή. Στις πρωτότυπες εκδόσεις, τον ρόλο του USB to UART Driver τον κάνει ένας μικρότερος μικροελεγκτής, συνήθως ένας ATmega16U, αλλά σε συμβατές πλακέτες τον συγκεκριμένο ρόλο τον αναλαμβάνει το FT232RL το οποίο είναι και η οικονομική εναλλακτική.

5.2 Accelerometer

Για να μπορέσουμε να παρακολουθήσουμε τις δονήσεις των βηματικών κινητήρων μέσω του μικροελεγκτή μας, θα πρέπει να χρησιμοποιήσουμε ένα επιταχυνσιομετρο, με το οποίο θα μπορέσουμε γραφικά να αναπαραστήσουμε τις μετρήσεις μας με κάποιο Plotter και έτσι να αναγνωρίσουμε τις διαφορές μεταξύ κανονικής και ανώμαλης λειτουργίας. Έπειτα από δοκιμές και έρευνα αγοράς, ένα αισθητήριο που πληροί τα κριτήρια μας, είναι το ολοκληρωμένο MPU-6050 το οποίο μπορούμε να βρούμε, εύκολα στην εγχώρια αγορά, σε μορφή Module, δηλαδή ήδη τοποθετημένο σε πλακέτα PCB με τα απαραίτητα περιφερειακά που χρειάζεται για να λειτουργήσει.

Το συγκεκριμένο αισθητήριο, διαθέτει γυροσκοπιο, επιταχυνσιομετρο και αισθητήριο θερμοκρασίας. Στην δική μας περίπτωση θα χρησιμοποιήσουμε το επιταχυνσιομετρο και το αισθητήριο θερμοκρασίας για να παρακολουθούμε και την άνοδο της θερμοκρασίας του βηματικού κινητήρα κατα την διάρκεια λειτουργίας, ώστε να δούμε αν ποτέ ξεπεραστούν οι μέγιστες τιμές που έχουν ορισθεί από τον κατασκευαστή, οι οποίες θα έδειχναν πιθανότητα βλάβης.



Εικόνα 5.2.1
MPU6050 Module

5.2.1 Στερέωση Επιταχυνσιόμετρου

Στο κεφάλαιο 4, είχαμε αναφέρει κάποια κύρια στοιχεία που φέρουν οι εκτυπωτές τύπου FDM. Στην δική μας περίπτωση, έχουμε στην διάθεση μας ένα μηχάνημα της εταιρίας Creality, τον Ender-3, ο οποίος φέρει σύστημα Bowden προώθησης υλικού(Extruder), που σημαίνει πως ο βηματικός κινητήρας βρίσκεται σε άλλη θέση από το ακροφύσιο. Οπότε, κατα την σχεδίαση της βάσης του αισθητηρίου ,την οποία και θα εκτυπώσουμε, θα πρέπει να εξετάσουμε το πώς είναι τοποθετημένος ο κινητήρας και τι περιορισμούς μπορεί να συναντήσουμε.

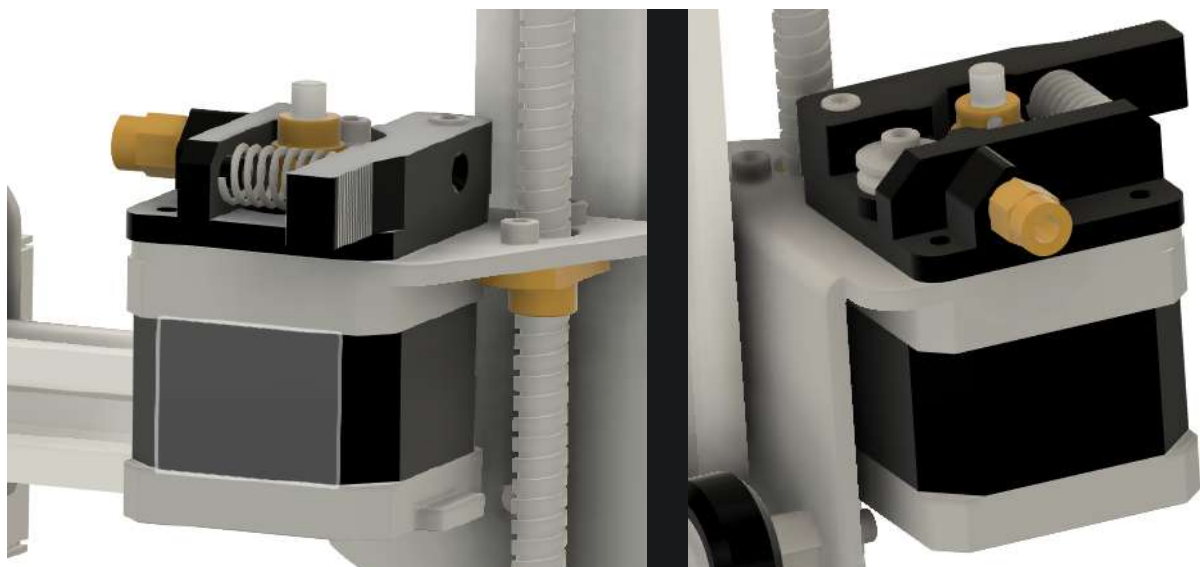
Κατα την σχεδίαση, θα ακολουθήσουμε την modular λογική, δηλαδή, θα πρέπει να υλοποιήσουμε κάτι το οποίο θα μπορεί να τοποθετηθεί σε μια πληθώρα μηχανημάτων δίχως κόπο και κάποια περίπλοκη συναρμολόγηση. Για να επιτευχθεί αυτό:

Η βάση μας θα πρέπει να κουμπώνει πάνω στον κινητήρα δίχως την χρήση εργαλείων

Θα πρέπει να είναι αρκετά μικρή ώστε να μπορεί να εγκατασταθεί και σε άλλους τύπους μηχανημάτων

Θα πρέπει να φέρει το αισθητήριο σε άμεση επαφή με τον κινητήρα για την ακριβέστερη μέτρηση δονήσεων και θερμοκρασίας.

Για την σχεδίαση, θα χρησιμοποιήσουμε το πρόγραμμα σχεδίασης Fusion 360, το οποίο είναι δωρεάν για μη επαγγελματική χρήση. Αρχικά θα πρέπει να παρατηρήσουμε την τοποθέτηση του extruder πάνω στον σκελετό του μηχανήματος ώστε να ορίσουμε τι βρίσκεται δίπλα στο μοτερ που πιθανότατα να ενοχλούσε την εφαρμογή της βάσης.



Εικόνα 5.2.2
Ο Extruder του Ender-3

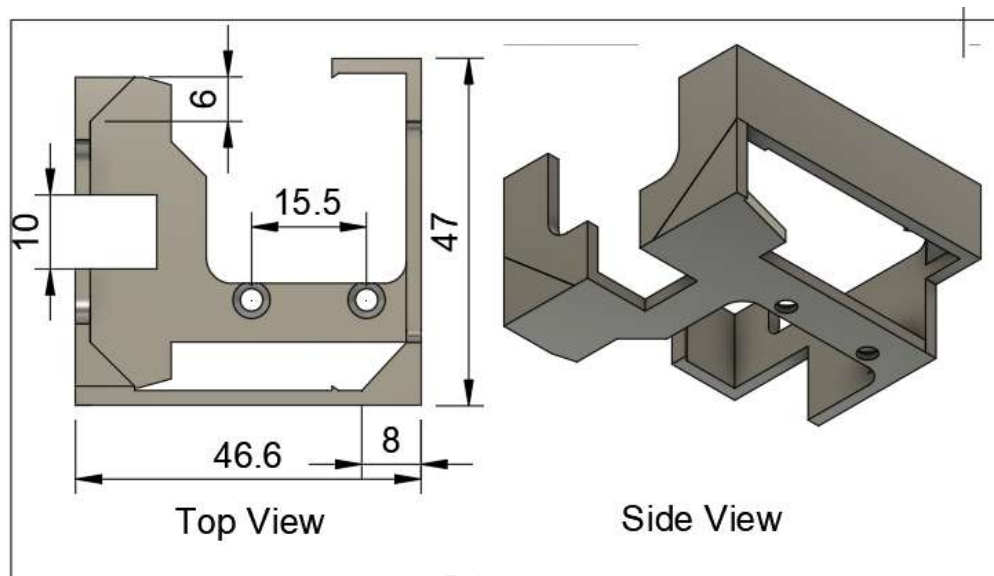
Από τις παραπάνω εικόνες παρατηρούμε, ήδη, ορισμένους περιορισμούς, τους οποίους θα μας δυσκολεψουν, όπως είναι:

Ο σύνδεσμος καλωδίων του κινητήρα,

Η βίδα που συγκρατεί το ένα ρουλεμάν κίνησης του Z άξονα

Ο περιορισμένος χώρος μεταξύ σκελετού και κινητήρα.

Έχοντας αυτές τις συνθήκες, το σχέδιο μας θα είναι διαμορφωμένο ως εξής :



Εικόνα 5.2.3
Προσχέδιο της βάσης του επιταχυνσιομετρου MPU-6050

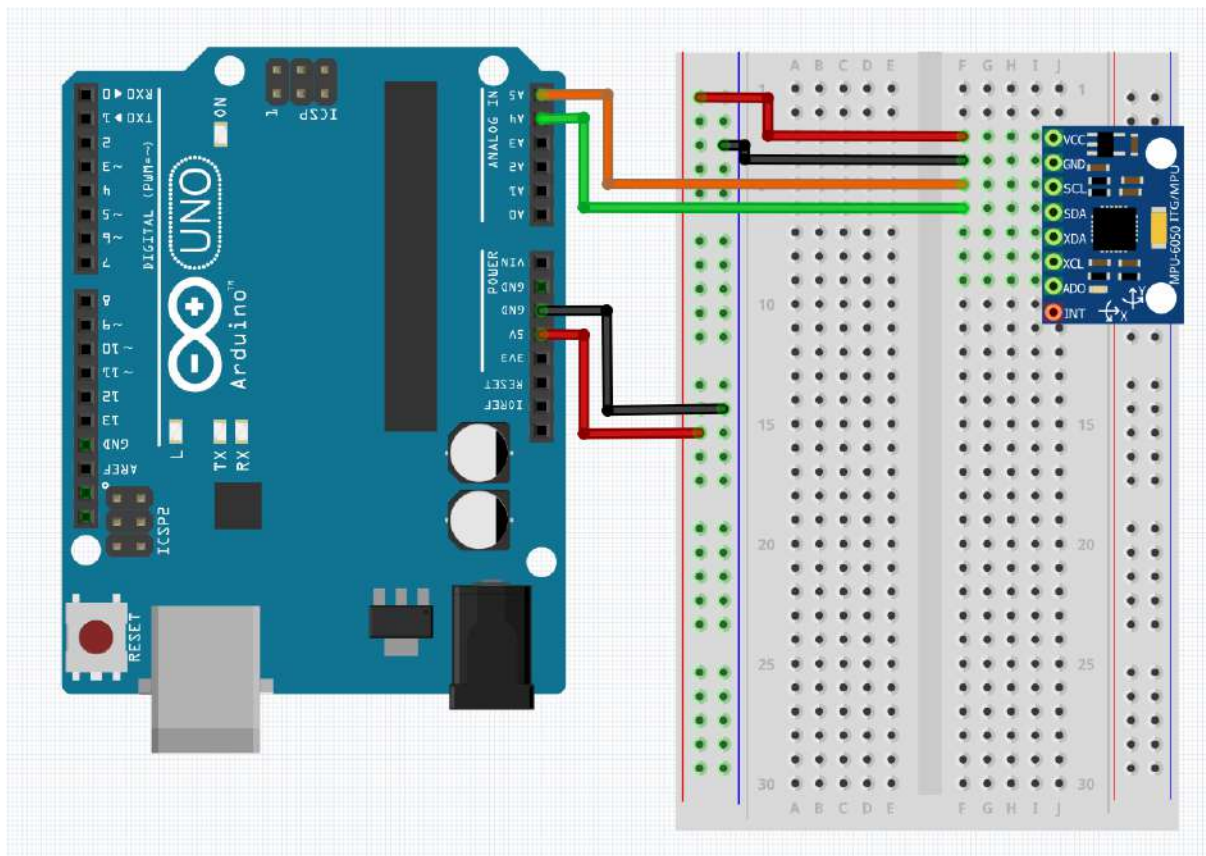
Αφου ολοκληρώσουμε το σχέδιο, παραμένει μόνο να το εκτυπώσουμε και να το δοκιμάσουμε πάνω στον βηματικό μας κινητήρα.



Εικόνα 5.2.4
Τρισδιάστατη απεικόνιση εφαρμογής της βάσης του
επιταχυνσιομετρου MPU-6050

5.2.2 Συνδεσμολογία/ Κώδικας Επιταχυνσιόμετρου

Αφού φτιάξαμε μια βάση για το αισθητήριο μας, παραμένει να ολοκληρώσουμε τις καλωδιώσεις μεταξύ του Arduino Uno και του αισθητηρίου. Όπως θα δούμε παρακάτω απαιτείται τροφοδοσία 5V, την οποία και δίνουμε απευθείας από την κύρια τροφοδοσία της πλακέτας. Οι ενώσεις SCL και SDA, οι οποίες απαιτούνται για την επικοινωνία μέσω του πρωτοκόλλου I2C, θα συνδεθούν αντίστοιχα στα ADC5 και ADC4 pins του μικροελεγκτή ή A5 και A4 όπως αναγράφεται πάνω στην πλακέτα. Το κύκλωμα σύνδεσης του αισθητηρίου με την πλακέτα Arduino uno είναι η εξής :



Εικόνα 5.2.5
Συνδεσμολογία Arduino Uno και MPU6050

Έχοντας ολοκληρώσει το κύκλωμα, παραμένει μόνο να προγραμματίσουμε την πλακέτα μας ώστε να επικοινωνεί και να διαβάζει τα δεδομένα που συλλέγει κατά την λειτουργία του. Για να δοκιμάσουμε, σε πρώτο στάδιο αν το αισθητήριο μας λειτουργεί, θα συντάξουμε ένα κομμάτι κώδικα ώστε να επιβεβαιώσουμε την λειτουργία του.

Πρώτα, πρέπει να συμβουλευτούμε το εγχειρίδιο του κατασκευαστή του MPU6050 ώστε η επικοινωνία να είναι ορθή και τα συλλεγμένα δεδομένα να μετατραπούν ορθώς σε μονάδες επιτάχυνσης g (G-Force). Από το εγχειρίδιο, παρατηρούμε πως το αισθητήριο μας μπορεί να ρυθμιστεί ως προς το μέγιστο εύρος κλιμακας, στην δική μας περίπτωση, δεν θα χρειαστεί να υπερβούμε τα +2g. Διαθέτει 16-bit ADCs, το οποίο υποστηρίζει τις ταυτόχρονες λήψεις δεδομένων επιτάχυνσης, θερμοκρασίας και γυροσκοπιου σε υψηλή ανάλυση. Όπως,

είναι λογικό για το πρωτόκολλο I2C, οι συσκευές-σκλάβοι έχουν διευθύνσεις ώστε να αναγνωρίζουν αν η πληροφορία που στέλνει ο μικροελεγκτής προορίζεται για εκείνες, το MPU6050 έχει διεύθυνση 0x68 όταν ο ακροδέκτης D0 είναι γειωμένος, δηλαδή λογικό “0” και 0x69 όταν δέχεται τροφοδοσία που ορίζεται ως λογικό “1”, αυτό μας βοηθά σε περίπτωση που έχουμε παραπάνω από ένα ίδια αισθητήρια. Γνωρίζοντας όλες τις παραπάνω πληροφορίες, θα συντάξουμε τον κώδικα μας :

```
#include<Wire.h>

const int MPU_addr=0x68; // I2C address of the MPU-6050
double AcX,AcY,AcZ,Tmp;

void setup()
{
    Wire.begin();// Startup I2C
    Wire.beginTransmission(MPU_addr);// Insert Device Address
    Wire.write(0x6B); // PWR_MGMT_1 register
    Wire.write(0); // set to zero (wakes up the MPU-6050)
    Wire.endTransmission(true);

    //Open Serial Communication
    Serial.begin(9600);
}

void loop()
{
    // I2C Communication with the MPU6050 Module
    Wire.beginTransmission(MPU_addr);
    Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
    Wire.endTransmission(false);
    Wire.requestFrom(MPU_addr,8,true); // request a total of 8 registers
    AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
    AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
    AcZ=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
    Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)

    //We have to divide the RAW Accel Data with 16384.5 to convert to G-force, for easier Handling, According to the Datasheet
    Serial.print(AcX/16384,5);
    Serial.print(" ");

    Serial.print(AcY/16384,5);
    Serial.print(" ");

    Serial.print(AcZ/16384,5);
    Serial.print(" ");

    Serial.print(Tmp/340.00+36.53); //equation for temperature in //degrees C from datasheet
    Serial.print(" ");

    delay(33);
}
```

Εικόνα 5.2.6
Κώδικας λειτουργίας MPU6050

Ο συγκεκριμένος κώδικας, θα μας βοηθήσει στο να συλλέξουμε δεδομένα κίνησης του βηματικού προώθησης υλικού. Μιας και έχουμε την δυνατότητα σειριακής επικοινωνίας με τον υπολογιστή, θα χρησιμοποιήσουμε το πρόγραμμα CoolTerm, που πρόκειται για ένα εργαλείο σειριακής επικοινωνίας, για να αποθηκεύσουμε δεδομένα δονήσεων για τους τρεις άξονες, με σκοπό να εντοπίσουμε συνθήκες ορθής και ανώμαλης λειτουργίας. Έτσι θα δημιουργήσουμε ένα Dataset που θα μας χρησιμεύσει στην εκπαίδευση του μοντέλου μηχανικής μάθησης.

5.3 Encoder

Για να καταφέρουμε να μετράμε πόσο υλικό περνάει μέσα από τον extruder προς το ακροφύσιο, θα πρέπει να χρησιμοποιήσουμε μια μέθοδο η οποία κατά προτίμηση δεν θα επηρεάζει την ροή του υλικού, να μην φέρει δηλαδή αντίσταση κατά την κίνηση του. Ένα μηχανικό αισθητήριο που θα μπορούσε να εξυπηρετήσει τον σκοπό μας, είναι το Encoder.

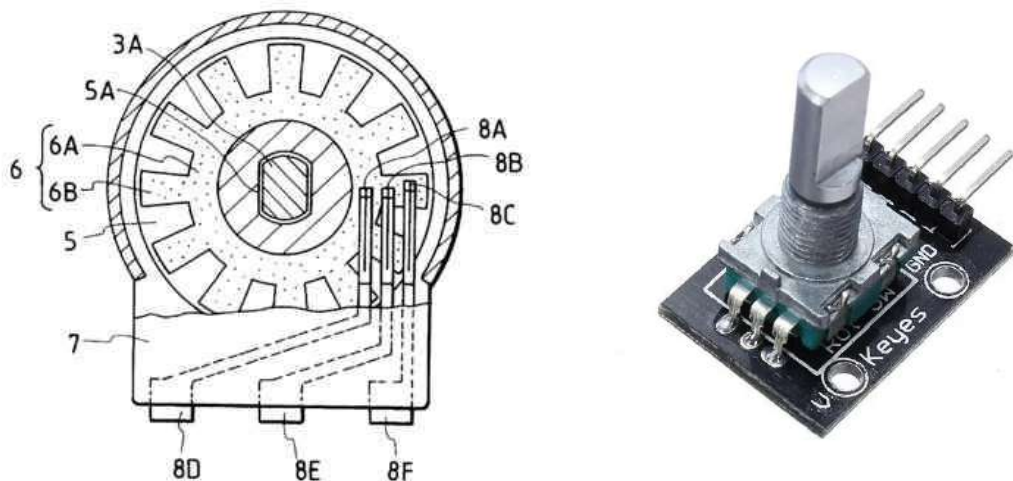
Ένα Encoder, ή περιστροφικός κωδικοποιητής, πρόκειται για μια συσκευή εισόδου η οποία έχει την δυνατότητα να περιστρέφεται συνεχώς προς οποιαδήποτε κατεύθυνση. Κατά την περιστροφή, παράγει ψηφιακούς παλμούς δυο φάσεων, με τους οποίους μπορούμε να αναγνωρίσουμε την κίνηση αλλά και την κατεύθυνση περιστροφής.

Στην καθημερινότητα μας μπορούμε να τα συναντήσουμε σε συστήματα με κάποιου είδους ελεγχόμενη διεπαφή, ώστε να γίνετε πλοήγηση σε κάποιο μενού, η χρήση τους όμως, δεν περιορίζεται μόνο εκεί. Όπως θα δούμε, στην δική μας περίπτωση θα το χρησιμοποιήσουμε σαν μετρητή απόστασης όπου θα υπολογίζουμε το υλικό που εισέρχεται ανά μια φάση εκτύπωσης και θα αποθηκεύουμε τα αποτελέσματα, με σκοπό να μπορούμε να έχουμε την συνολική εικόνα κατανάλωσης ενός καρουλιού όπως και να αντιμετωπίσουμε τυχόν διακοπές τροφοδοσίας, που θα μπορούσε να συμβεί και σε μια κανονική βιομηχανική μονάδα.

Υπάρχουν διαφορετικά είδη περιστροφικών κωδικοποιητών, μερικοί εκ αυτών :

- I. Αυξητικοί περιστροφικοί κωδικοποιητές (Incremental Rotary Encoders)
- II. Οπτικοί κωδικοποιητές (Optical Encoder)
- III. Μαγνητικοί κωδικοποιητές (Magnetic Encoder)

Στην δική μας περίπτωση, θα χρησιμοποιήσουμε έναν Αυξητικό περιστροφικό κωδικοποιητή, 20 σημείων(ή βημάτων) ανά περιστροφή. Ο άξονας του κωδικοποιητή θα μας είναι χρήσιμος για την στερέωση του πάνω στον άξονα κίνησης του βηματικού κινητήρα ώστε να μετράμε τις περιστροφές που θα κάνει, κάτι το οποίο δεν θα επιβαρύνει την κίνηση του κινητήρα και του υλικού. Όταν θα πραγματοποιήσουμε την συνδεσμολογία και την συγγραφή του κώδικα θα πρέπει να δώσουμε προσοχή στο φαινόμενο της αναπήδησης διακόπτη, η οποία μπορεί να μας δημιουργήσει ψευδείς μετρήσεις, παρακάτω θα δούμε τον τρόπο επίλυσης του συγκεκριμένου προβλήματος.

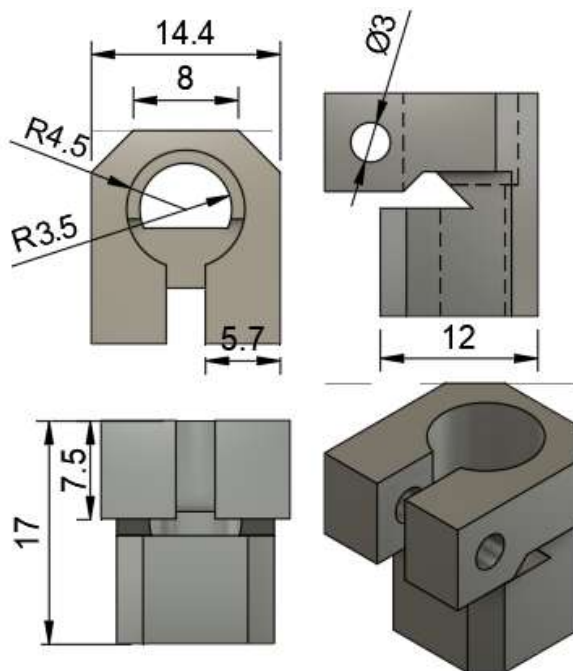


Εικόνα 5.3.1
Incremental Rotary Encoder

5.3.1 Στερέωση Encoder

Πριν ξεκινήσουμε τις δοκιμές με το Encoder, θα πρέπει να σχεδιάσουμε μια βάση με την οποία αυτό θα παραμένει σταθερό και να σκεφτούμε έναν εύκολο τρόπο τοποθέτησης πάνω στο Extruder Assembly. Όπως και πριν τα αντικείμενα στερέωσης θα τα σχεδιάσουμε μέσω του προγράμματος Fusion 360 και θα τα εκτυπώσουμε με την βοήθεια ενός τρισδιάστατου εκτυπωτή από PLA.

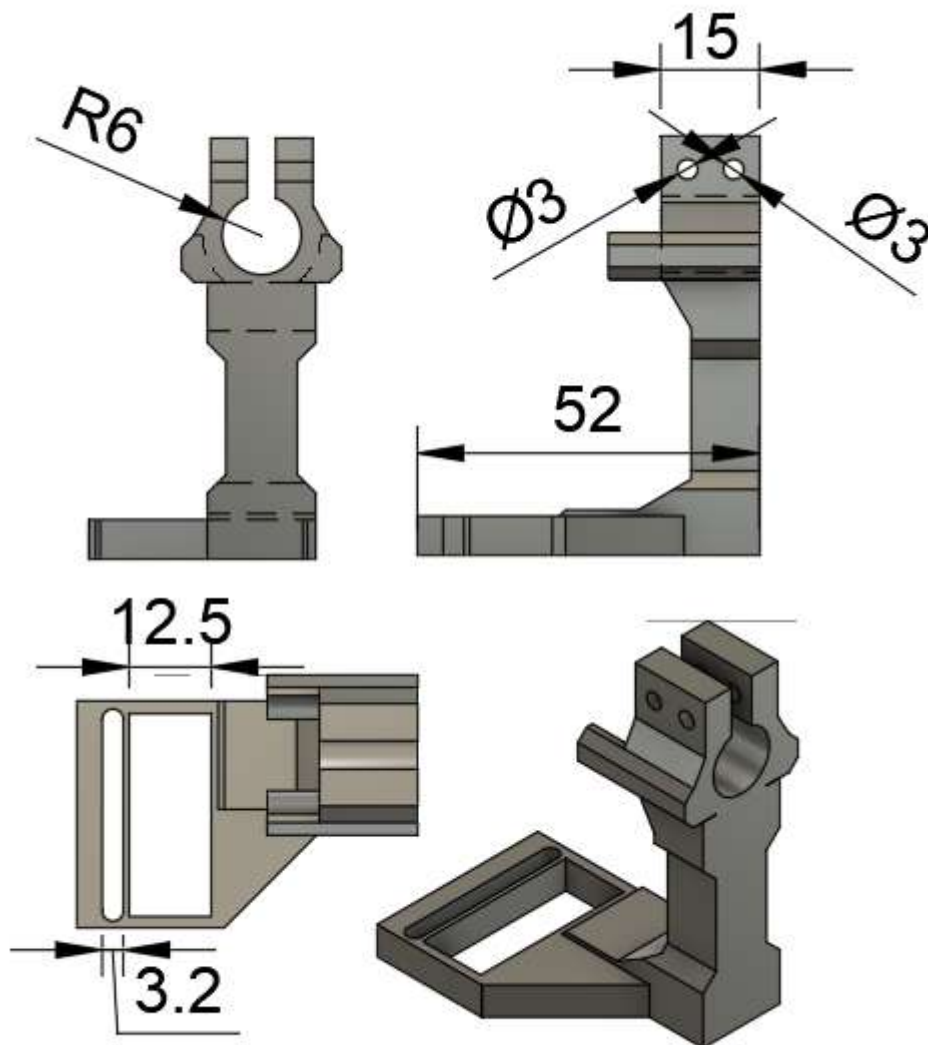
Πρώτα θα σχεδιάσουμε τον σύνδεσμο μεταξύ του άξονα του Encoder και του άξονα του βηματικού κινητήρα, όπου πάνω βρίσκεται και το Extruder Gear, το οποίο προωθεί το υλικό και η μορφή του είναι ως εξής :



Εικόνα 5.3.2
Σύνδεσμος κινητήρα με Encoder

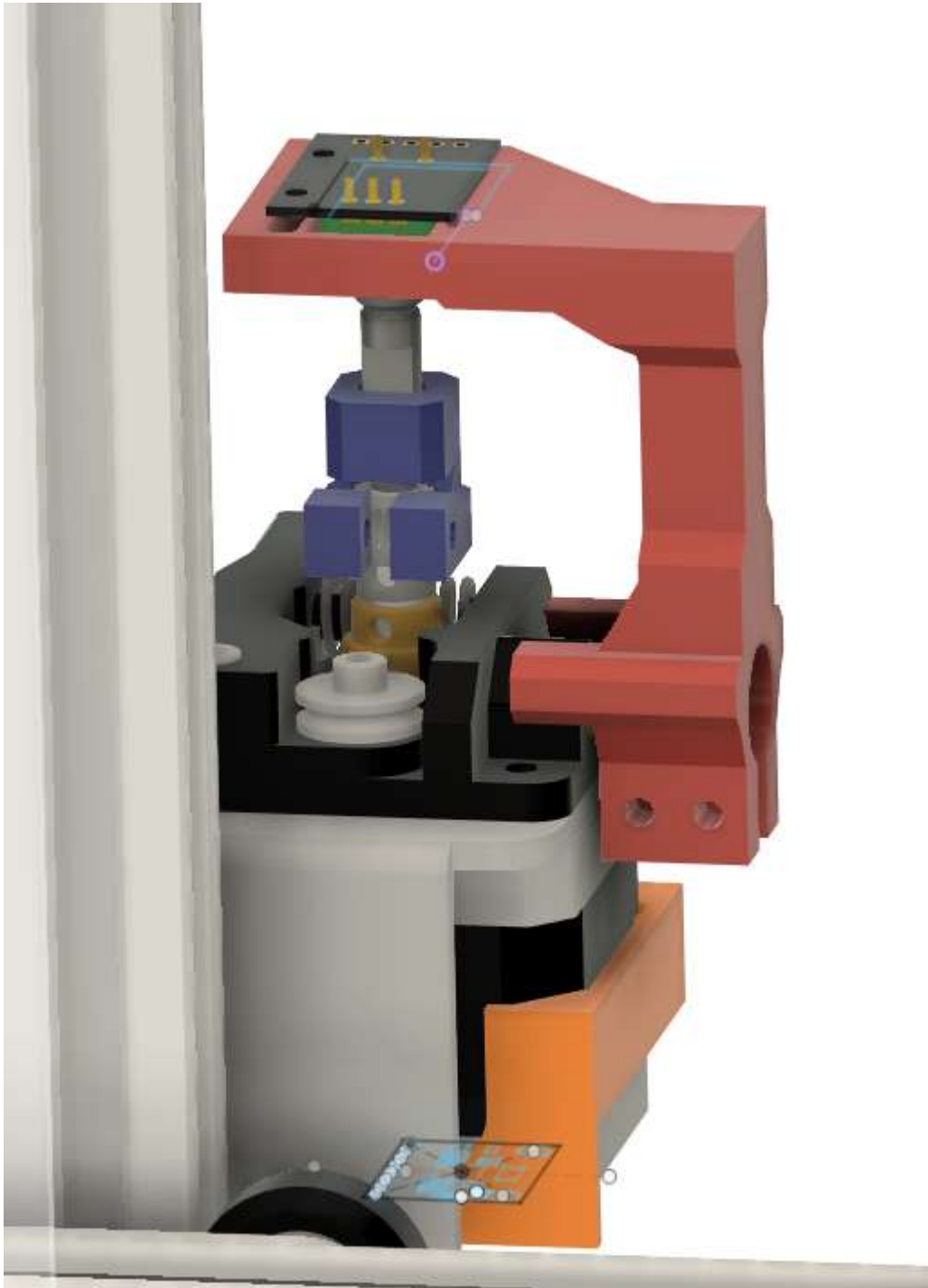
Όπως παρατηρούμε, το άκρο που συνδέεται με το Encoder το σχεδιάσαμε σύμφωνα με το προφίλ του άξονα του, ώστε να μπαίνει σφηνωτά, αντίθετα επειδή ο κινητήρας μας έχει στρογγυλό άξονα, σχεδιάσαμε έναν τρόπο σύσφιξης, όπου χρησιμοποιούμε μια βίδα και ένα παξιμάδι τύπου M3. Όπως παρατηρούμε όμως από τις μετρήσεις, η άκρη που συνδέεται στον κινητήρα είναι αρκετά μεγαλύτερη από τον άξονα του που έχει διάμετρο 5mm, αυτό έγινε για να χρησιμοποιήσουμε ένα επιπλέον γρανάτζι κίνησης, ώστε να γίνει ορθή σύσφιξη πάνω στον κινητήρα. Η σύνδεση απευθείας στον κινητήρα θα μπορούσε να οδηγήσει σε λανθασμένες μετρήσεις λόγω ολίσθησης σε περίπτωση ατέλειας του πλαστικού του συνδέσμου. Το μήκος κατα το οποίο εισέρχεται το δευτερεύον γρανάτζι, είναι υπολογισμένο έτσι, ώστε η σύσφιξη να γίνεται πάνω στις οδοντώσεις, με σκοπό να μην υπάρχει η περίπτωση ανεπιθύμητων ολισθήσεων.

Αφού στερεώθηκε το Encoder πάνω στο μοτέρ, πρέπει να σχεδιάσουμε μια επιπλέον βάση όπου θα το κρατάει σταθερό για να γίνεται σωστά η περιστροφή του άξονα. Η βάση, θα στερεώνεται πάνω στον πνευματικό σύνδεσμο που συγκρατεί τον σωλήνα PTFE που καθοδηγεί το υλικό προς το θερμαινόμενο ακροφύσιο. Η βάση μας θα έχει την παρακάτω μορφή :



Εικόνα 5.3.3
Βάση σταθεροποίησης Encoder

Προτού εκτυπώσουμε τον σύνδεσμο και την βάση, μπορούμε να κάνουμε επαλήθευση σχεδίασης, εφαρμόζοντας τα σε ένα ψηφιακό μοντέλο του εκτυπωτή, όπως κάναμε δηλαδή και για το επιταχυνσιομετρο προηγούμενος.



Εικόνα 5.3.4
Προσομοίωση τοποθέτησης συνδέσμου και βάσης Encoder

Για επιπλέον συγκράτηση θα χρειαστούμε 4 βίδες των 20mm και 4 παξιμάδια τύπου M3. Οι πρώτες 2 βίδες θα σφίξουν την βάση πάνω στο πνευματικό σύνδεσμο και οι άλλες 2 θα στερεώσουν το encoder πάνω στην βάση ώστε να μην υπάρχει τζόγος κατά την κίνηση

του άξονα του κινητήρα και έτσι κάποιο σφάλμα στις μετρήσεις. Στο στοιχείο της αστάθειας έπρεπε να δοθεί προσοχή, διότι, ο κινητήρας δεν σπρώχνει μόνο το υλικό προς μια φορά αλλά εν συνεχεία μπορεί να το τραβήξει πίσω, ώστε να αποφευχθεί η καταλάθος πτώση υλικού εκεί που δεν θα έπρεπε να είναι. Αυτή η ιδιαιτερότητα σε συνδυασμό με το φαινόμενο αναπήδησης διακόπτη, θα μπορούσαν να επιστρέψουν λανθασμένους παλμούς.

5.3.2 Συνδεσμολογία/ Κώδικας Encoder

Αφού σχεδιάσαμε και υλοποιήσαμε την βάση και τον σύνδεσμο, τώρα μας δίνετε η δυνατότητα δοκιμής της κατασκευής μας από πλευρά κώδικα.

Πρώτα όμως, θα σχεδιάσουμε την συνδεσμολογία μας μεταξύ Arduino Uno και Encoder.

Εδώ λοιπόν καλούμαστε, να βρούμε μέθοδο αντιμετώπισης του θορύβου που δημιουργείται από το διακοπτικό φαινόμενο που φέρουν οι μηχανικοί κωδικοποιητές.

Στον κώδικα μας, θα χρησιμοποιήσουμε την μέθοδο Polling, δηλαδή αναμονής συνθήκης,

για να διαβάσουμε τα σήματα που δεχόμαστε από το Encoder. Θα μπορούσαμε να

χρησιμοποιήσουμε και την μέθοδο interrupt, δηλαδή, με την άνοδο του σήματος προς το

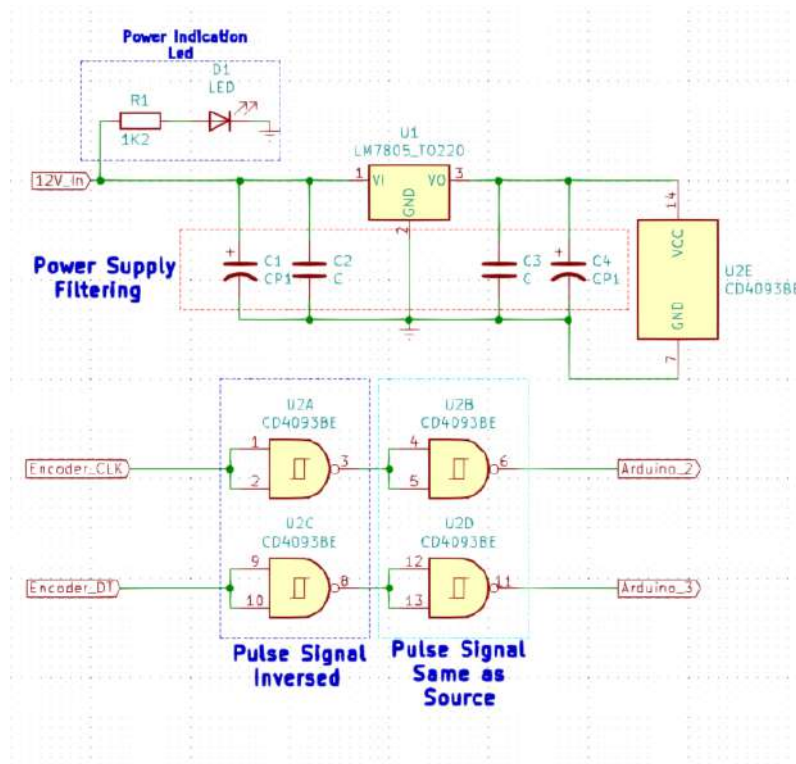
λογικό "1", να διακόπτουμε την λειτουργία της κύριας συνάρτησης Loop() για να κάνουμε

ανάγνωση εισερχόμενου σήματος, αλλά αυτή η μέθοδος μπορεί να μας διακόπτει

λανθασμένα την συνάρτηση μας λόγω κάποιου στοιχείου θορύβου.

Στην δική μας περίπτωση θα χρησιμοποιήσουμε το παρακάτω κύκλωμα ώστε να

αντιμετωπίσουμε το διακοπτικό φαινόμενο :

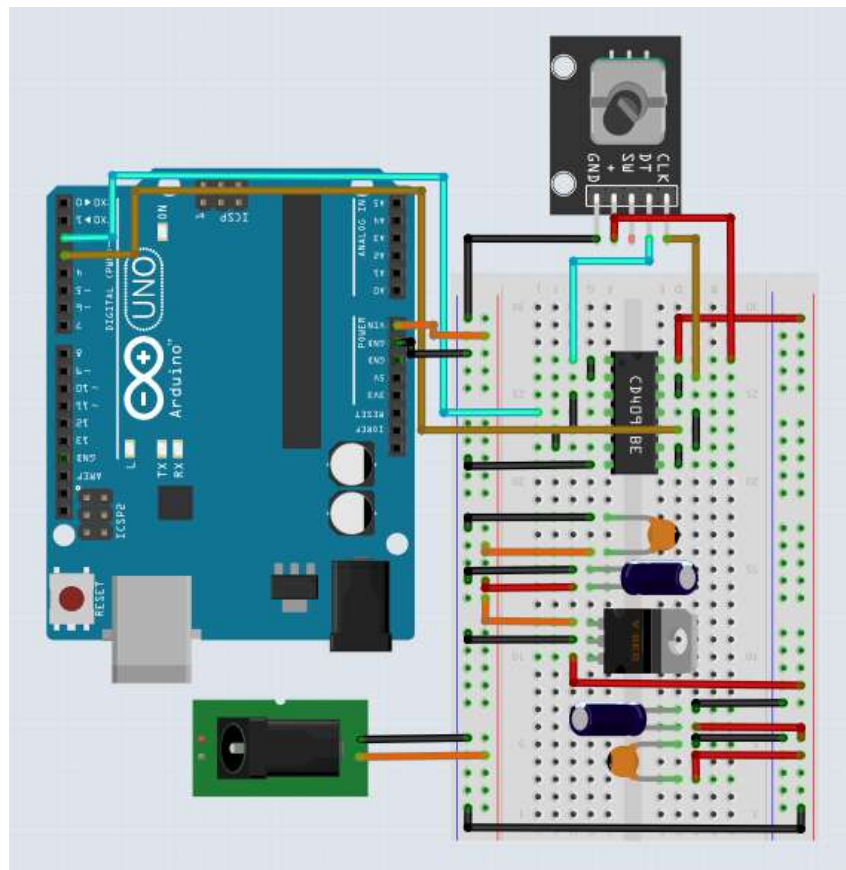


Εικόνα 5.3.5
Κύκλωμα αντιμετώπισης διακοπτικών πηγών θορύβου

Κατά το πρώτο σκέλος, θα χρησιμοποιήσουμε το ολοκληρωμένο κύκλωμα CD4093BE, το οποίο περιλαμβάνει 4 λογικές πύλες τύπου NAND με Schmitt-trigger λειτουργία σε κάθε είσοδο του κυκλώματος. Χάρης το Schmitt-trigger καταφέρνουμε να αντιμετωπίσουμε μικρο-διακυμάνσεις του σήματος, χαρακτηριστικό της διακοπτικής λειτουργίας, επειδή το σήμα μετατρέπεται σε λογικό “0” ή “1” ανάλογα την τάση εισόδου, αποφεύγοντας έτσι ενδιάμεσες και “λογικα θορυβώδης” τιμές εξόδου της πύλης.

Επιπλέον, γνωρίζουμε πως ο ρυθμιστής τάσης της πλακέτας Arduino Uno, είναι γραμμικός, που σημαίνει ότι όσο ανεβαίνει το φορτίο που απαιτείται από το σύστημα τόσο ανεβαίνει και η θερμοκρασία που εκπέμπεται από το ολοκληρωμένο, επιπλέον επειδή είναι της μορφής SMD με πακέτο SOT-223, δεν υπάρχει η κατάλληλη μάζα για ορθή απαγωγή θερμότητας, ειδικά σε εγκαταστάσεις με υψηλές περιβάλλον θερμοκρασίες. Αυτό, μπορεί, ανάλογα τις συνθήκες περιβάλλοντος που βρίσκεται το σύστημα να οδηγήσει στην ενεργοποίηση του Thermal Shutdown, δηλαδή της διακοπής λειτουργίας λόγω υψηλής θερμοκρασίας, και έτσι να διακοπεί η τροφοδοσία προς το σύστημα που θα μπορούσε να οδηγήσει και σε προβλήματα ως προς την ορθή λειτουργία του ολοκληρωμένου συστήματος. Για να επιλύσουμε αυτή την ανησυχία και παράλληλα να υποβοηθήσουμε το υπόλοιπο κύκλωμα από πλευράς θορύβου τροφοδοσίας, θα χρησιμοποιήσουμε έναν επιπλέον ρυθμιστή τον L7805 του οποίου θα έχουμε φροντίσει να φιλτράρουμε την είσοδο και έξοδο του.

Αφού ολοκληρώσαμε σχεδιαστικά το κύκλωμα, τώρα θα πραγματοποιήσουμε την συνδεσμολογία του με το Arduino Uno :



Εικόνα 6.3.5
Συνδεσμολογία κυκλώματος με το Encoder και το Arduino Uno

Έχοντας ολοκληρώσει την συνδεσμολογία, τώρα θα συντάξουμε τον κώδικα μας, ώστε να πραγματοποιήσουμε και δοκιμές ορθής λειτουργίας.

Όπως αναφέραμε και προηγουμένως, θα χρησιμοποιήσουμε την μέθοδο Polling για την ανάγνωση των σημάτων που θα δίνει το Encoder. Αφού το στερεώσαμε πάνω από τον άξονα του βηματικού κινητήρα προώθησης υλικού θα πρέπει να μετρήσουμε την διάμετρο του οδοντωτού γραναζιού, το οποίο σπρώχνει το υλικό, και με την εξίσωση της περιφέρειας κύκλου να βρίσκουμε πόσο υλικό έχει περάσει. Ο κώδικας διαμορφώνεται ως εξής :

```
#define CLK_PIN 2
#define DATA_PIN 3

int N=20;//Encoder Full rotation
const float pi = 3.1415;
const float r=0.00549;//Radius of extruder Gear
static int16_t state=0,counter=0;
float Filament_Distance_m ;

void setup()
{
  //Pin Config for encoder pins
  pinMode(CLK_PIN,INPUT);
  pinMode(DATA_PIN,INPUT);

  //Open Serial Communication
  Serial.begin(9600);
}

void loop()
{
  //Encoder Reading
  state=(state<<1) | digitalRead(CLK_PIN) | 0xe000;

  if (state==0xf000)
  {
    state=0x0000;
    if(digitalRead(DATA_PIN))
      counter--;
    else
      counter++;
  }

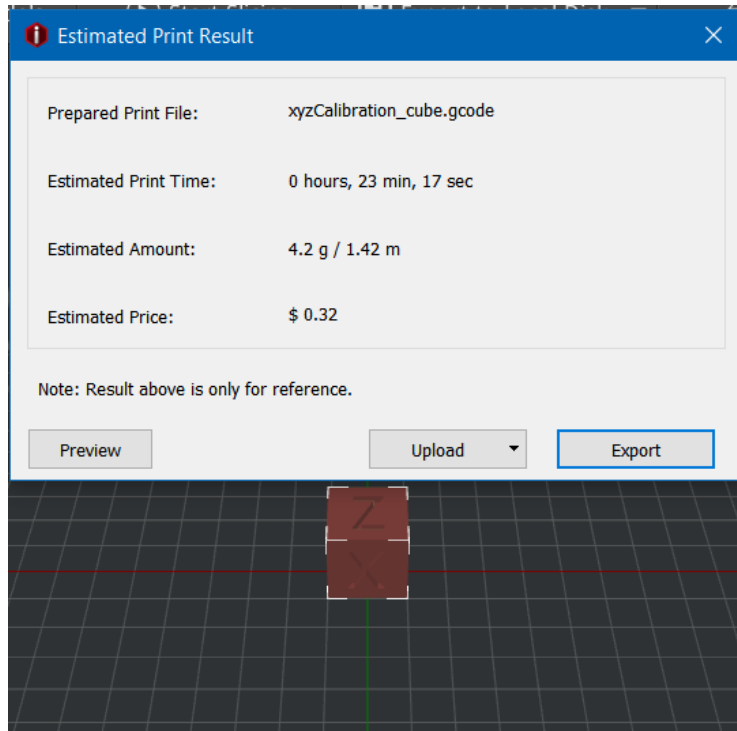
  Filament_Distance_m = ((2*pi*r)/N) * counter;

  Serial.println(Filament_Distance_m, 4);//conversion to meters

  delay(33);
}
```

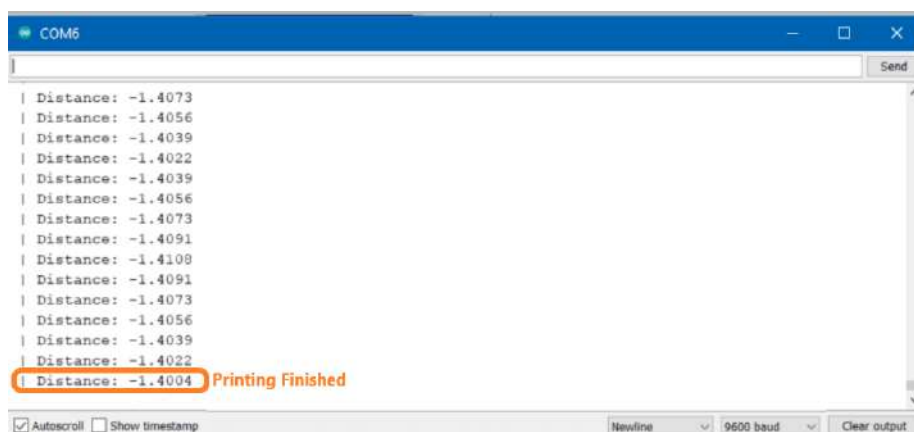
Εικόνα 5.3.7
Κώδικας λειτουργίας Encoder

Αφού ανεβάσουμε τον κώδικα στον μικροελεγκτή και έχουμε τοποθετήσει το encoder στην βάση του, θα πρέπει να πραγματοποιήσουμε μια εκτύπωση ώστε να επαληθεύσουμε την λειτουργία του και αν οι μετρήσεις είναι σωστές ως προς το υλικό που προωθείτε. Για να το καταφέρουμε αυτό, θα τυπώσουμε έναν κύβο διαστάσεων 20mm x 20mm x 20mm, με την βοήθεια κάποιου Slicing προγράμματος, στην δική μας περίπτωση το ideaMaker θα δούμε πόση θα είναι περίπου η κατανάλωση υλικού :



Εικόνα 5.3.8
Κατανάλωση δοκιμαστικής εκτύπωσης

Αφού από το πρόγραμμα βλέπουμε πως θα υπάρξει κατανάλωση 1.42 μέτρων υλικού, θα πρέπει και οι μετρήσεις που θα κάνουμε να έρθουν πάρα πολύ κοντά σε αυτήν την τιμή, μιας και είναι λογικό να μην είναι ακριβώς η ίδια λόγω μικρών σφαλμάτων στο ίδιο το νήμα ή κάποιου άλλου σφάλματος. Συνδέοντας σειριακά το Arduino Uno, και παρακολουθώντας τα σειριακά του μηνύματα, παρατηρούμε το εξής :

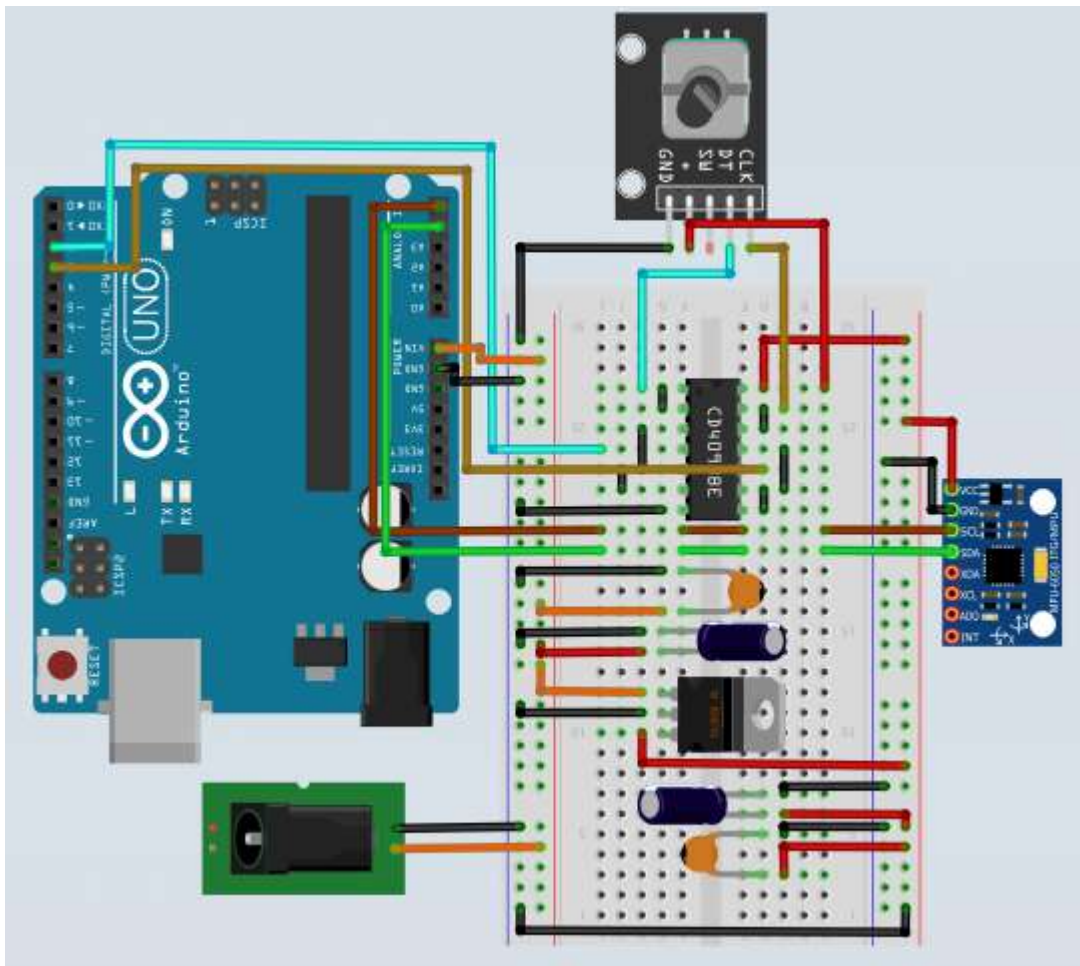


Εικόνα 5.3.9
Επαλήθευση κατανάλωσης υλικού μέσω σειριακής παρακολούθησης

Με την παραπάνω επαλήθευση, οι οποία πειραματικά έγινε αρκετές φορές, πριν την συγγραφή αυτού του υποκεφαλαίου, μπορούμε να προχωρήσουμε στο τελικό στάδιο αυτής της φάσης σχεδίασης του συστήματος και να συγκροτήσουμε τα δύο παραπάνω τμήματα.

5.4 Συγχώνευση

Έχοντας ολοκληρώσει μεμονωμένα την παραμετροποίηση και την δοκιμή του επιταχυνσιομετρου και του Encoder αλλά και την ορθή στερέωση αυτών, ήρθε η ώρα να συγκροτήσουμε τα δύο αυτά τμήματα, ώστε να μπορέσουμε να περάσουμε στην επόμενη φάση σχεδίασης του συστήματος μας. Αρχικά θα δούμε το σχηματικό μας, ολοκληρωμένο :



Εικόνα 5.4.1
Ολοκληρωμένη συνδεσμολογία συστήματος μετρήσεων

Έχοντας ολοκληρώσει την παραπάνω συνδεσμολογία, θα συνθέσουμε τον τελικό κώδικα από τον οποίο θα συλλέγουμε δεδομένα και θα τα αποστέλλουμε σειριακά, σε ένα μικρο-υπολογιστή, ώστε να γίνει η διαχείριση αυτών, από το μοντέλο μηχανικής μάθησης. Το σύστημα μας πέρα από την ανάλυση και την αναγνώριση ανεπιθύμητων στοιχείων κατα την λειτουργία, λειτουργεί και σαν λογιστικό εργαλείο με το να επιβλέπει πόσο υλικό έχει παραμείνει σε ένα συγκεκριμένο καρούλι. Για να επιτύχουμε όμως, την βέλτιστη λειτουργία του συστήματος μας όσον αφορά την επίβλεψη υλικού, θα πρέπει να προνοήσουμε για κάποιο σενάριο διακοπής τροφοδοσίας, δηλαδή δεν θέλουμε η μεταβλητές μας να βρίσκονται στην μνήμη Flash του μικροελεγκτή, διότι έτσι μπορεί να χαθούν. Οπότε, θα πρέπει το υλικό το οποίο έχουμε μετρήσει ανα κάποιες φάσεις εργασίας, να αποθηκεύεται στην μόνιμη μνήμη του μικροελεγκτή, δηλαδή την EEPROM. Έτσι θα μπορούμε με σιγουριά να γνωρίζουμε, πόσο υλικό μας έχει απομείνει και αν το μηχάνημα που έχει αυτό το υλικό μπορεί να συνεχίσει τις εκτυπώσεις. Η τελική μορφή του κώδικα :

```
//Include Libraries
#include<Wire.h>
#include <EEPROM.h>
#include <avr/eeprom.h>
//definitions for Encoder pins
#define CLK_PIN 2
#define DATA_PIN 3

const int MPU_addr=0x68; // I2C address of the MPU-6050
double AcX,AcY,AcZ,Tmp;

int N=20;//Encoder Full rotation
const float pi = 3.1415;
const float r=0.00549;//Radius of extruder Gear
static int16_t state=0,counter=0;
float Filament_Distance_m ;
float temp ; // Stepper Motor Temperature
char command; // Serial Command for saving to EEPROM

void setup()
{
//Pin Config for encoder pins
pinMode(CLK_PIN,INPUT);
pinMode(DATA_PIN,INPUT);

//I2C Setuo
Wire.begin();
Wire.beginTransmission(MPU_addr);
Wire.write(0x6B); // PWR_MGMT_1 register
Wire.write(0); // set to zero (wakes up the MPU-6050)
Wire.endTransmission(true);

//Open Serial Communication
Serial.begin(9600);
//After Boot, read the last value of Filament recorded
eeprom_read_block((void*)&counter, (void*)0, sizeof(counter));
}
```

Εικόνα 5.4.2
Ολοκληρωμένος κώδικας λειτουργίας, 1ο σκελος

```

void loop()
{
  //(Serial)
  {
    command = Serial.read();
    if(command == 's')
    {
      eeprom_update_block((const void*)&counter, (void*)0, sizeof(counter));
      // After saving reset the command
      command = '0';
    }
  }
  //I2C Communication with the MPU6050 Module
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr,8,true); // request a total of 8 registers
  AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
  AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
  AcZ=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
  Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)

  //Encoder Reading
  state=(state<<1) | digitalRead(CLK_PIN) | 0x000;

  if (state==0x000)
  {
    state=0x000;
    if(digitalRead(DATA_PIN))
      counter--;
    else
      counter++;
  }
  Filament_Distance_m = ((2*pi*r)/N) * counter; //Convert counter measurement to meters of Filament

  //We have to divide the RAW Accel Data with 16384.5 to convert to G-force, for easier Handling, According to the Datasheet
  Serial.print(AcX/16384.5);
  Serial.print(" ");
  Serial.print(AcY/16384.5);
  Serial.print(" ");
  Serial.print(AcZ/16384.5);
  Serial.print(" ");
  Serial.print(Tmp/340.00+36.53); //equation for temperature in degrees C from datasheet
  Serial.print(" ");
  Serial.println(Filament_Distance_m, 4); //conversion to meters

  delay(33);
}

```

Εικόνα 5.4.3
Ολοκληρωμένος κώδικας λειτουργίας, 2ο σκελος

Έχοντας ολοκληρώσει, πλέον, όλες τις κατασκευές και τον κώδικα που αφορούν την ανάγνωση αισθητηρίων που τοποθετήσαμε πάνω στον τρισδιάστατο εκτυπωτή, ολοκληρώσαμε το πρώτο, θεμέλιο, σκέλος του συστήματός μας. Ο λόγος που είναι τόσο σημαντικό, είναι επειδή, χάρις αυτό θα μπορούσαμε να φτιάξουμε τα Dataset που είναι απαραίτητα για την εκπαίδευση του μοντέλου μηχανικής μάθησης. Περισσότερο, θα αναλύσουμε τα στοιχεία αυτά, στο επόμενο κεφάλαιο, μιας και πέρα από την εκπαίδευση του μοντέλου, πρέπει να διαχειριστούμε τα δεδομένα με σκοπό την ομαδοποίηση τους σε δεδομένα ορθής ή ανώμαλης λειτουργίας.

Κεφάλαιο 6

Μηχανική Μάθηση - Anomaly Detection

Σε μία εγκατάσταση όπου υπάρχουν πολλά μηχανήματα, αν υπάρχει πρόβλημα ροής υλικού, δεν θα μπορούμε να το αντιληφθούμε άμεσα λόγω του θορύβου που θα επικρατεί, μιας και σε ένα ρεαλιστικό σενάριο πλήρους φόρτου εργασίας, όλα τα μηχανήματα θα είναι λειτουργικά. Ο εντοπισμός του μηχανήματος με το σφάλμα δεν θα μπορεί να γίνει άμεσα αντιληπτό, η ταχύτητα με την οποία θα βρεθεί το πρόβλημα, θα μας βοηθήσει ώστε να αποφύγουμε εκτυπωμένα αντικείμενα τα οποία δεν είναι ποιοτικά αποδεκτά. Σε κάποιο χειρότερο σενάριο, αν δεν γίνει έγκαιρα αντιληπτή, κάποια ανώμαλη κατάσταση, υπάρχει περίπτωση να δημιουργηθεί βλάβη η οποία μπορεί να είναι χρονοβόρα και να απαιτεί κάποιο καινούργιο εξάρτημα, το οποίο ανεβάζει το κόστος παραγωγής.

Για τον έλεγχο ανωμαλιών που μπορεί να αντιμετωπίσει το σύστημα προώθησης υλικού ενός τρισδιάστατου εκτυπωτή θα χρησιμοποιήσουμε έναν ειδικό αλγόριθμο μηχανικής μάθησης.

Η μηχανική μάθηση αποτελεί μια εφαρμογή τεχνητής νοημοσύνης (AI), η οποία δίνει τη δυνατότητα σε ορισμένα συστήματα να εκπαιδεύονται από μόνα τους μέσω της εμπειρίας και των δεδομένων που συλλέγουν χωρίς να απαιτείται η ανθρώπινη παρέμβαση ή να πραγματοποιείται επαναπρογραμματισμός του κυρίου αλγόριθμου λειτουργίας σύμφωνα με τα καινούργια δεδομένα που μπορεί να συλλέχθηκαν. Στη δική μας περίπτωση θα χρησιμοποιήσουμε κάποιο τέτοιο μοντέλο ώστε να εντοπίσουμε ανωμαλίες στο σύστημα. Οι κύριες κατηγορίες αλγορίθμων μηχανικής μάθησης χωρίζονται σε δύο σε αυτές που χρειάζονται επίβλεψη και σε αυτές που δεν χρειάζονται.

Οι αλγόριθμοι που δεν απαιτούν την επίβλεψη λειτουργίας έχουν εκπαιδευτεί από ένα ήδη έτοιμο σύνολο δεδομένων το οποίο ονομάζουμε dataset. Αυτό το σύνολο δεδομένων αποτελείται από στοιχεία τα οποία δεν είναι κατηγοριοποιημένα και ο αλγόριθμος προσπαθεί να εντοπίσει διαφορές.

Η κατηγορία των αλγορίθμων η οποία απαιτεί επίβλεψη εκπαιδεύεται, σε αντίθεση με την προηγούμενη, με δεδομένα τα οποία είναι κατηγοριοποιημένα. Με την εκπαίδευση του αλγόριθμου γίνεται εφικτή η πρόβλεψη των τιμών εξόδου από τα εισαγόμενα δεδομένα του του συστήματος. Ο αλγόριθμος μπορεί να συγκρίνει την έξοδο του με την σωστή προοριζόμενη έξοδο και να υπολογίσει το σφάλμα ώστε να τροποποιηθεί το μοντέλο αναλόγως

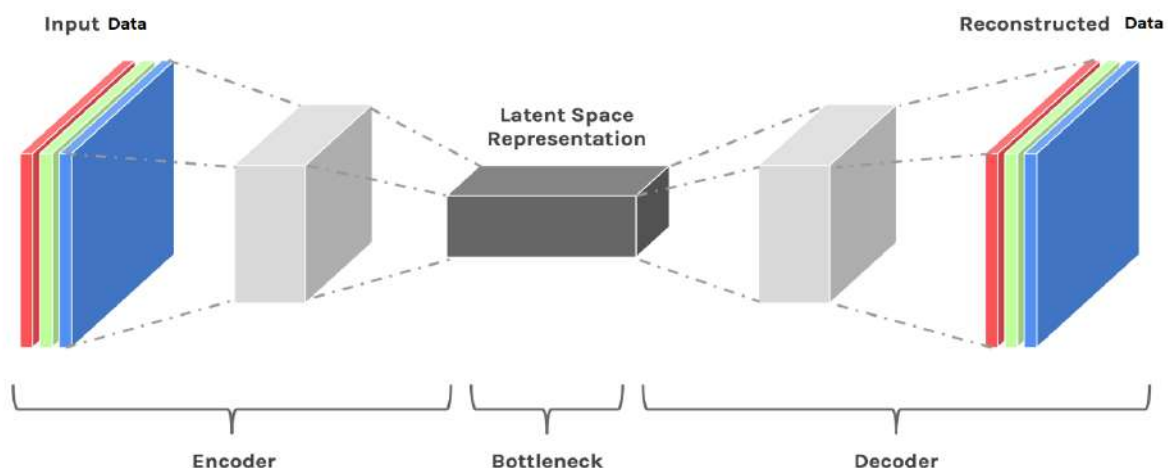
Ανίχνευση ανωμαλιών ή ανίχνευση ακραίων τιμών, καλείται η αναγνώριση ορισμένων προτύπων από ένα σύνολο δεδομένων που εμφανίζουν διαφορετική συμπεριφορά από αυτή που αναμένεται από το σύστημα ως η ιδανική κατάσταση λειτουργίας. Για την ανίχνευση ανωμαλιών απαιτείται ένας πολύ μεγάλος όγκος δεδομένων έτσι ώστε το μοντέλο να μπορεί να αναγνωρίζει μοτίβα και χαρακτηριστικά στα καινούργια δεδομένα που δέχεται, στην δική μας περίπτωση, από το Arduino που συλλέγει δεδομένα από τα αισθητήρια που τοποθετήσαμε στον εκτυπωτή.

6.1 Autoencoder

Στην δική μας περίπτωση επιθυμούμε το μοντέλο που θα δημιουργήσουμε να συγκρίνει τις κυματομορφές που σχηματίζονται από τα δεδομένα που λαμβάνουμε από το επιταχυνσιόμετρο και να αντιλαμβάνεται αν η λειτουργία του βηματικού κινητήρα είναι ορθή. Για αυτό θα δημιουργήσουμε έναν μοντέλο μηχανικής μάθησης τύπου Autoencoder.

Το Autoencoder είναι ένα τεχνητό νευρωνικό δίκτυο το οποίο μαθαίνει να συμπιέζει και να κωδικοποιεί τα δεδομένα που δέχεται και έπειτα να τα ανασχηματίζει σε μορφή παρόμοια της αρχικής. Αν η ανακατασκευή των δεδομένων φέρει μικρά ή καθόλου σφάλματα σημαίνει πως το σύστημα λειτουργεί ορθά. Αν όμως το μέσο τετραγωνικό σφάλμα είναι αρκετά μεγαλύτερο από μία προκαθορισμένη τιμή σημαίνει πως στο σύστημα υπάρχει μία ανωμαλία. Ένας Autoencoder αποτελείται από :

- I. Τον κωδικοποιητή - Encoder
- II. Το ενδιάμεσο σημείο κωλύματος - Bottleneck
- III. Τον αποκωδικοποιητή - Decoder
- IV. Την απώλεια ανοικοδόμησης - Reconstruction Loss



Εικόνα 6.1.1
Διάγραμμα λειτουργίας ενός Autoencoder

Στο μοντέλο μηχανικής μάθησης, ο κωδικοποιητής εκπαιδεύεται έτσι ώστε τα εισερχόμενα δεδομένα να μειώνονται όσον αφορά τις διαστάσεις τους και να συμπιέζονται σε μια κωδικοποιημένη αναπαράσταση σε έναν λανθάνων χώρο. Εάν θεωρήσουμε τα εισερχόμενα δεδομένα ως x και τον κωδικοποιητή ως E τότε η αναπαράσταση του λανθάνοντος χώρου εξόδου, s , θα είναι :

$$s = E(x)$$

Το ενδιάμεσο σημείο εμπεριέχει την συμπιεσμένη αναπαράσταση των εισερχόμενων δεδομένων. Ο αποκωδικοποιητής του μοντέλου εκπαιδεύεται ώστε να ανασχηματίζει τα εισερχόμενα δεδομένα, με βάση την συμπιεσμένη αναπαράσταση που βρίσκεται σε έναν

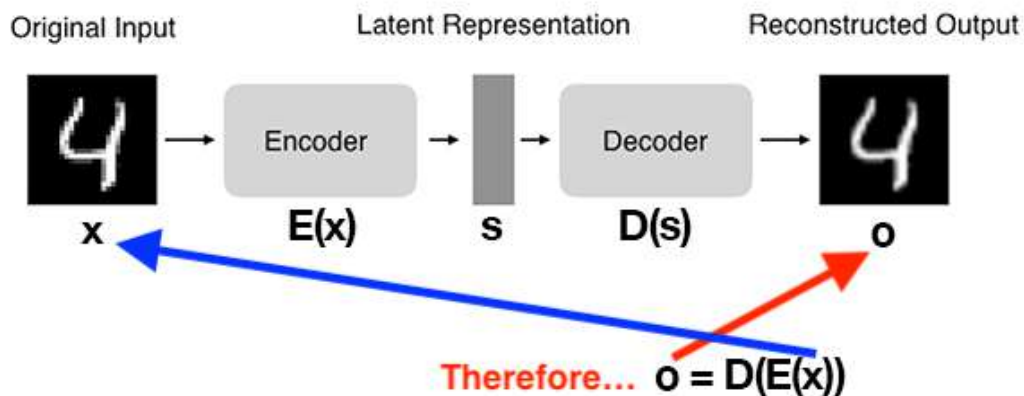
λανθάνων χώρο, προσπαθώντας να είναι όσο το δυνατόν ακριβέστερος, ώστε τα αποτελέσματα του να είναι ίδια με τα εισερχόμενα. Αν ο λανθάνων χώρος χαρακτηρίζεται ως s , ο αποκωδικοποιητής ως D και η έξοδος του o , τότε :

$$o = D(s)$$

Η απώλεια ανοικοδόμησης, είναι το μέτρο που περιγράφει το σφάλμα, με το οποίο κρίνουμε αν ο ανασχηματισμός των δεδομένων έγινε αποδοτικά και αν αυτά είναι παρόμοια με τα δεδομένα εισόδου.

Κατά την εκπαίδευσή του μοντέλου χρησιμοποιείται η μέθοδος Back-Propagation με σκοπό την ελαχιστοποίηση του σφάλματος κατά την ανοικοδόμηση. Στην δική μας περίπτωση θα χρησιμοποιήσουμε τη μηχανική μάθηση για να μειώσουμε τα σφάλματα της παραγωγής αλλά και να εντοπίζουμε αδύναμα σημεία λειτουργίας γρήγορα. Σύμφωνα με τις προηγούμενες μαθηματικές εκφράσεις, μπορούμε να ορίσουμε πως η διαδικασία της εκπαίδευσης εκφράζεται μαθηματικά ως :

$$o = D(E(x))$$



Εικόνα 6.1.2
Μαθηματική έκφραση Autoencoder

Όπως καταλαβαίνουμε από τα παραπάνω στοιχεία οι Autoencoders έχουν σαν ρόλους:

- I. Μείωση διαστάσεων
- II. Αφαίρεση θορύβου από τα εισαγόμενα δεδομένα
- III. Ανίχνευση ανωμαλιών/ Ακραίων τιμών

Στην περίπτωση της παρούσα έρευνας, χρησιμοποιείται η μέθοδος Autoencoder, επειδή, αφού συλλέξουμε τα απαραίτητα δεδομένα από το επιταχυνσιομετρο, τα οποία και θα εκφράζουν μια κανονική κατάσταση λειτουργίας, θα εκπαιδεύσουμε το μοντέλο με αυτά. Αυτό έχει σαν κύριο σκοπό, την αναγνώριση κανονικών καταστάσεων λειτουργίας, οι οποίες θα επιστρέφουν μηδενική ή ελάχιστη απώλεια ανοικοδόμησης. Αν υπάρξει κάποια κατάσταση η οποία δεν είναι στα πλαίσια αναγνώρισης από το μοντέλο, θα κατηγοριοποιείται ως ανωμαλία.

6.2 Datasets

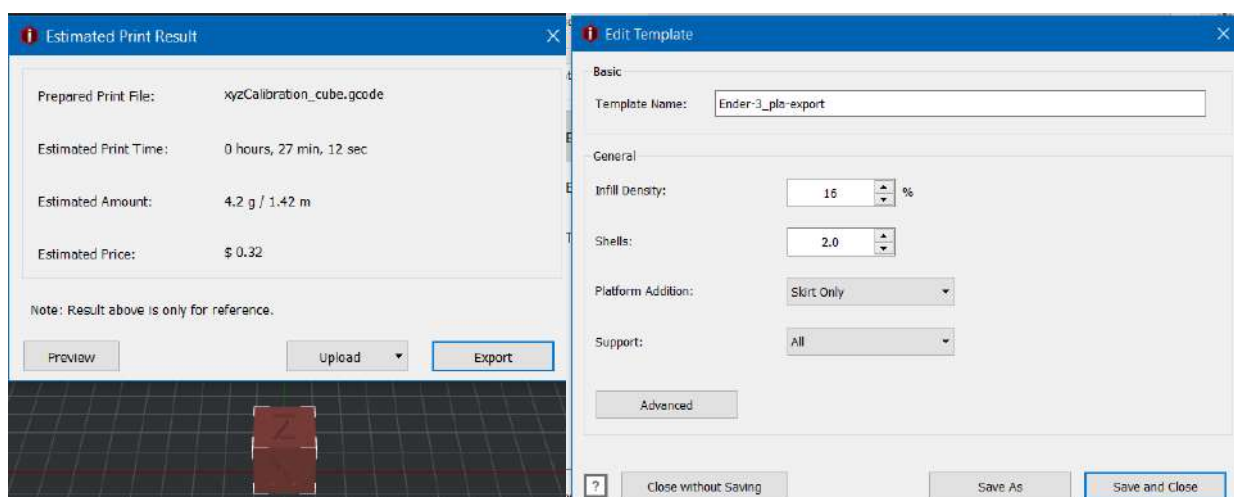
Κάθε μοντέλο μηχανικής μάθησης χρειάζεται και τα κατάλληλα δεδομένα ώστε να λειτουργήσει ορθά, μιας και πρόκειται για μία διαδικασία η οποία περιλαμβάνει τον εμπειρικό παράγοντα. Πέραν της ποιότητας, όμως, χρειαζόμαστε και ποσότητα.

Στην δική μας περίπτωση θα πρέπει να συλλέξουμε τα δεδομένα, από τη συσκευή που φτιάξαμε στο προηγούμενο κεφάλαιο με το Arduino UNO, να αποθηκεύσουμε τα δεδομένα σε φύλλα επεξεργασίας ανα εκτυπωμένο μοντέλο και να τα συγκρίνουμε ώστε να διακρίνουμε στοιχεία που εκδηλώνουν ορθή λειτουργία. Αυτά τα δεδομένα που θα συλλέξουμε θα τα χρησιμοποιήσουμε ώστε να εκπαιδεύσουμε το μοντέλο μηχανικής μάθησης.

6.2.1 Συλλογή Δεδομένων

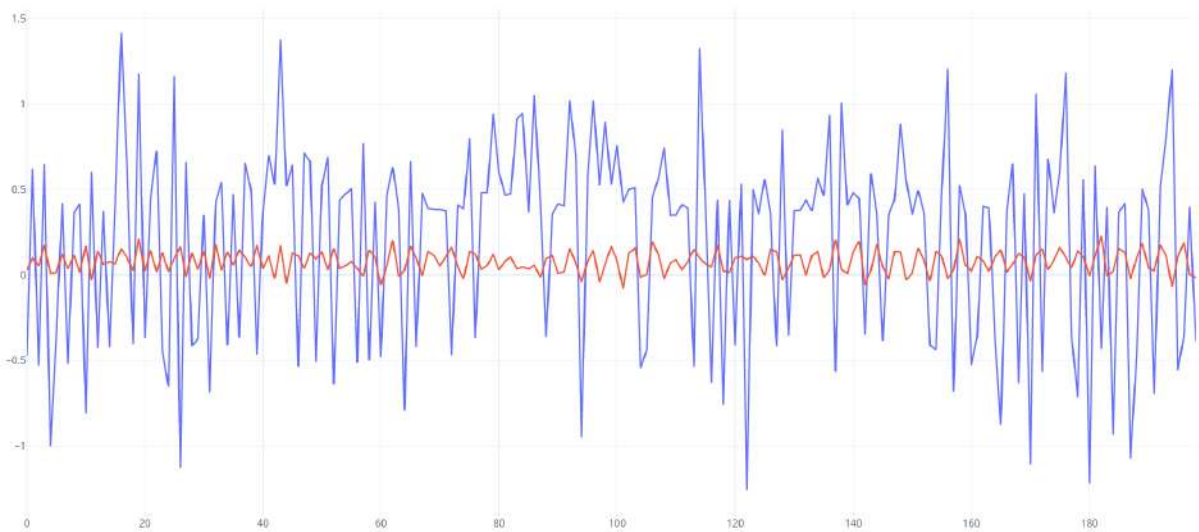
Για να δημιουργήσουμε τα dataset, που χρειαζόμαστε για να εκπαιδεύσουμε το μοντέλο μηχανικής μάθησης, χρειάζεται να συγκεντρώσουμε δεδομένα που εκφράζουν τη συμπεριφορά του βηματικού κινητήρα ανά έναν σταθερό κύκλο παραγωγής, δηλαδή μέχρι να ολοκληρωθεί η παραγωγή ενός τρισδιάστατα εκτυπωμένου αντικειμένου.

Για να πραγματοποιήσουμε σωστά τη συγκεκριμένη διαδικασία συλλογής, θα πρέπει να επαναλάβουμε την εκτύπωση ενός συγκεκριμένου αντικειμένου πολλές φορές. Το αντικείμενο που θα χρησιμοποιήσουμε θα είναι ένα απλό τετράγωνο 20mm x 20mm x 20mm, ώστε να μπορέσουμε να επαναλάβουμε πολλές φορές το πείραμα, στο ίδιο περιβάλλον αλλά και με τις ίδιες παραμέτρους, ώστε να σιγουρευτούμε ότι τα δεδομένα που θα συλλέξουμε ακολουθούν μία λογική και ένα μοτίβο, τα οποία θα μας εκφράσουν ορισμένα σημεία ενδιαφέροντος που μπορεί να εκμεταλλευτεί το μοντέλο. Αφού συλλέξουμε αρκετά δεδομένα θα τα συγκρίνουμε με μετρήσεις που εκφράζουν ανωμαλία στο σύστημα μας.



Εικόνα 6.2.1
Παράμετροι πειραματικής εκτύπωσης

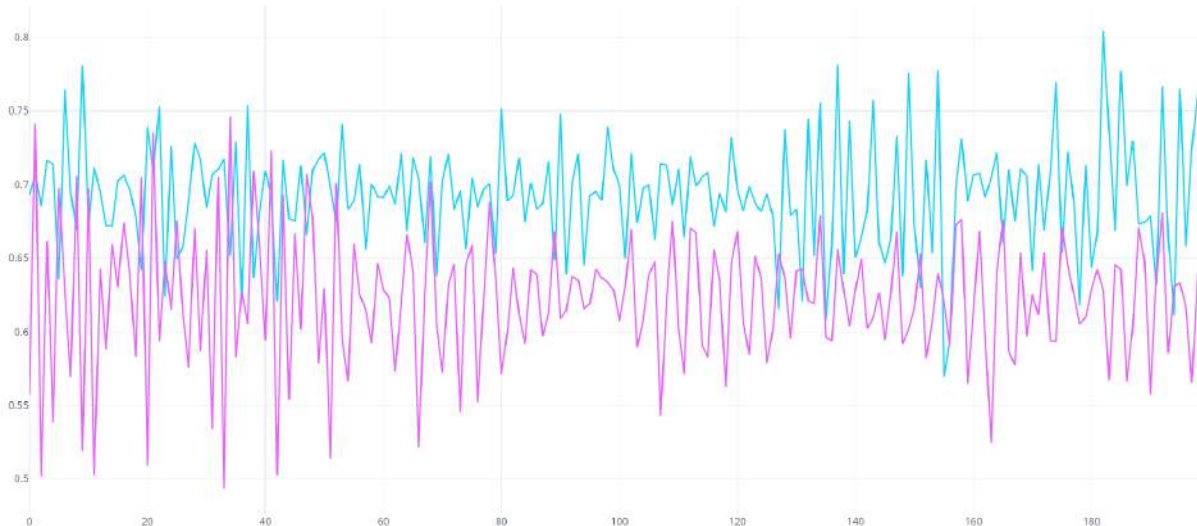
Για τις πειραματικές μετρήσεις Θα χρησιμοποιήσουμε το προφίλ εκτύπωσης το οποίο έχουμε κρίνει πώς βγάζει το βέλτιστο αποτέλεσμα ωπου το τρισδιάστατα εκτυπωμένο αντικείμενο είναι ποιοτικά αρεστό. Όσον αφορά τις μετρήσεις τις οποίες θα εκφράζουν την ανωμαλία του συστήματος, δηλαδή την δυσκολία ροής του υλικού η οποία μπορεί να δημιουργήσει σημαντικές ατέλειες στο τελικό προϊόν, αυτές θα τις συλλέξουμε προκαλώντας κάποια κάποια εξωτερική ανωμαλία. Για να το πετύχουμε αυτό, θα χαμηλώσουμε πάρα πολύ τη θερμοκρασία του ακροφυσίου, κατα την διάρκεια της εκτύπωσης ώστε να προσομοιώσουμε βλάβη στην αντίσταση θέρμανσης του Heating Block, με σκοπό να μην προλαβαίνει το νήμα να λιώσει σωστά και έτσι να προκαλεί αντίσταση κίνησης στον κινητήρα. Αυτή η ανωμαλία στο σύστημα, θα κάνει τον κινητήρα να χάσει βήματα και έτσι να υπάρξει ένα overshoot στην κυματομορφή της δονήσεως σε κάποιον από τους τρεις άξονες αναφοράς.



Εικόνα 6.2.2
Σύγκριση κανονικής(κόκκινη) και ανώμαλης(μπλε)
κυματομορφής λειτουργίας για τον άξονα X



Εικόνα 6.2.3
Σύγκριση κανονικής(μωβ) και ανώμαλης(πρασινη)
κυματομορφής λειτουργίας για τον άξονα Y



Εικόνα 6.2.4
Σύγκριση κανονικής(ροζ) και ανώμαλης(γαλαζιο)
κυματομορφής λειτουργίας για τον άξονα Z

Αξιολογώντας τις παραπάνω κυματομορφές, οι οποίες ανήκουν στο dataset που θα χρησιμοποιήσουμε για να εκπαιδύσουμε το μοντέλο μηχανικής μάθησης, παρατηρούμε σημαντικές διαφορές στις κυματομορφές των αξόνων X και Y.

6.2.2 Ανάλυση Χαρακτηριστικών Dataset

Τα δεδομένα που συλλέχθηκαν μέσω του Arduino UNO από τις εκτυπώσεις που κάναμε δημιουργήσαμε μία αρκετά ικανοποιητική βιβλιοθήκη δεδομένων ώστε να μπορούμε να συγκρίνουμε επιθυμητές καταστάσεις λειτουργίας με τις μη επιθυμητές. Πέρα από τις διαφορές που παρατηρήσαμε στις κυματομορφές που συγκρίναμε, τώρα από όλα τα αρχεία που έχουμε στη διάθεση μας θα μπορούμε να τα συγκρίνουμε για να βρούμε τις κύριες διαφοροποιήσεις που θα μπορούσαν να αξιοποιηθούν από το μοντέλο μηχανικής μάθησης. Μερικές διαφοροποιήσεις, από το σύνολο μετρήσεων ορθής λειτουργίας σε σχέση με τις μετρήσεις ανώμαλης συμπεριφοράς είναι :

- I. Διαφορές πλάτους - Variance
- II. Κέρτωση - Kurtosis
- III. Λοξότητα - Skew

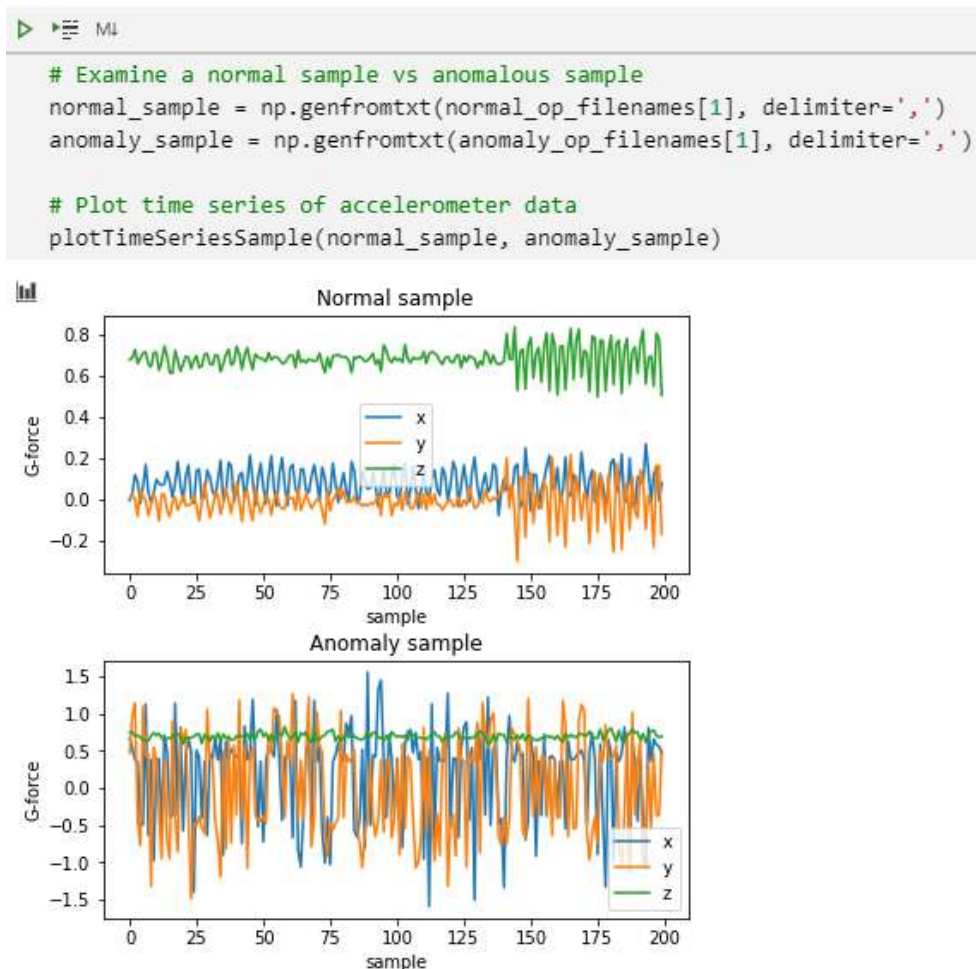
Θα χρησιμοποιήσουμε ένα jupyter notebook και την γλώσσα python, ώστε να τρέξουμε μικρά κομμάτια κώδικα. Θα εισάγουμε όλα μας τα δεδομένα με σκοπό να τα συγκρίνουμε μεταξύ τους και θα τα προβάλλουμε σε γραφικές παραστάσεις αξιοποιώντας τη γραφική βιβλιοθήκη matplotlib, που χρησιμοποιείται και από το MATLAB. Μιας και έχουν συλλεχθεί πολλά δεδομένα, θα πρέπει να τα ανακατέψουμε ώστε να βεβαιωθούμε πως οι διαφορές είναι διακριτές ακόμα και σε τυχαίες περιπτώσεις και δεν πρόκειται για τυχαίο

συμβάν. Πρέπει να σημειωθεί πως για κάθε δείγμα περιλαμβάνονται 200 μετρήσεις που έχουν συλλεχθεί από το επιταχυνσιόμετρο.

```
M4  
  
# Create normal and anomaly filename lists  
normal_op_filenames = createFilenameList(normal_op_list)  
anomaly_op_filenames = createFilenameList(anomaly_op_list)  
print('Number of normal samples:', len(normal_op_filenames))  
print('Number of anomaly samples:', len(anomaly_op_filenames))  
  
Number of normal samples: 889  
Number of anomaly samples: 1453
```

Εικόνα 6.2.5
Δείγματα ορθής και ανώμαλης λειτουργίας

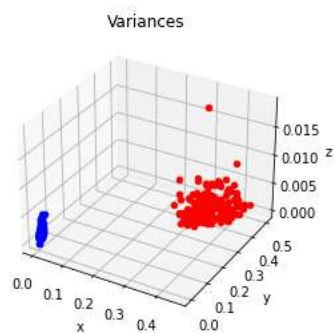
Εν συνεχεία, θα συγκρίνουμε δείγματα ορθής και ανώμαλης λειτουργίας, χρησιμοποιώντας όλες τις τιμές των αξόνων σε ένα γράφημα, ώστε να σχηματιστεί μια πλήρης εικόνα.



Εικόνα 6.2.6
Σύγκριση ορθής λειτουργίας και ανώμαλης κατάστασης

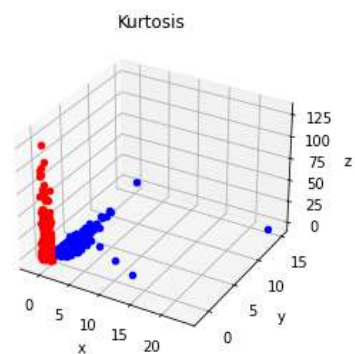
Έχοντας παρατηρήσει τα στοιχεία που διαφοροποιούν μία ορθή από μία ανώμαλη κατάσταση θα πρέπει τώρα, να μελετήσουμε τις διαφοροποιήσεις, που είχαμε αναφέρει προηγουμένως, τις οποίες μπορεί να έχουν τα δείγματα προς μελέτη χρησιμοποιώντας διαγράμματα διασποράς.

```
MI
# Make a 3D scatterplot of variances
num_samples = 200
normal_samples = []
anomaly_samples = []
for i in range(num_samples):
    normal_sample = np.genfromtxt(normal_op_filenames[i], delimiter=',')
    anomaly_sample = np.genfromtxt(anomaly_op_filenames[i], delimiter=',')
    normal_samples.append(np.var(normal_sample, axis=0))
    anomaly_samples.append(np.var(anomaly_sample, axis=0))
plotScatterSamples(normal_samples, anomaly_samples, num_samples, title='Variances')
```



Εικόνα 6.2.7
Διάγραμμα διασποράς διαφορών μεταξύ δειγμάτων

```
MI
# Make a 3D scatterplot of kurtosis
num_samples = 200
normal_samples = []
anomaly_samples = []
for i in range(num_samples):
    normal_sample = np.genfromtxt(normal_op_filenames[i], delimiter=',')
    anomaly_sample = np.genfromtxt(anomaly_op_filenames[i], delimiter=',')
    normal_samples.append(stats.kurtosis(normal_sample))
    anomaly_samples.append(stats.kurtosis(anomaly_sample))
plotScatterSamples(normal_samples, anomaly_samples, num_samples, title='Kurtosis')
```

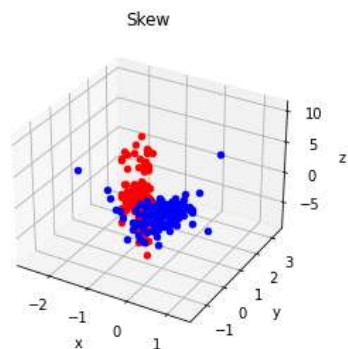


Εικόνα 6.2.8
Διάγραμμα διασποράς κύρτωσης μεταξύ δειγμάτων

```

▶ ML
# Make a 3D scatterplot of skew
num_samples = 200
normal_samples = []
anomaly_samples = []
for i in range(num_samples):
    normal_sample = np.genfromtxt(normal_op_filenames[i], delimiter=',')
    anomaly_sample = np.genfromtxt(anomaly_op_filenames[i], delimiter=',')
    normal_samples.append(stats.skew(normal_sample))
    anomaly_samples.append(stats.skew(anomaly_sample))
plotScatterSamples(normal_samples, anomaly_samples, num_samples, title='Skew')

```



Εικόνα 6.2.9
Διάγραμμα διασποράς λοξότητας μεταξύ δειγμάτων

Από τα παραπάνω χαρακτηριστικά που μπορούμε να διακρίνουμε, από την σύγκριση των δεδομένων, είναι πρωτοφανές πώς το χαρακτηριστικό της διαφοροποίησης θα αποτελέσει ένα σημαντικό στοιχείο στην εκπαίδευση του Autoencoder μιας και οι ακραίες τιμές είναι εύκολα διακριτές.

6.3 Autoencoder Training

Για να πραγματοποιήσουμε την εκπαίδευση του μοντέλου μηχανικής μάθησης θα πρέπει να χρησιμοποιήσουμε ορισμένα μαθηματικά εργαλεία και βιβλιοθήκες οι οποίες θα μας βοηθήσουν στο να χτίσουμε το μοντέλο μας ευκολότερα.

Αρχικά θα χρησιμοποιήσουμε μία δωρεάν βιβλιοθήκη ανοικτού κώδικα, ειδικά σχεδιασμένη για μηχανική μάθηση, η οποία αναπτύχθηκε από την Google και χρησιμοποιείται και από εκείνη. Η βιβλιοθήκη ονομάζεται Tensorflow και μπορεί να χρησιμοποιηθεί σε ένα ευρύ φάσμα εργασιών, αν και έχει ιδιαίτερη έμφαση στην εκπαίδευση και την εξαγωγή συμπερασμάτων Deep learning νευρωνικών δικτύων. Πρόκειται για μία συμβολική βιβλιοθήκη μαθηματικών με βάση τη ροή δεδομένων, όπως για παράδειγμα τη ροή δεδομένων από αισθητήρια, και τον διαφοροποιημένο προγραμματισμό.

Ένα ακόμη εργαλείο που θα χρησιμοποιήσουμε είναι το Keras, το οποίο είναι μία βιβλιοθήκη ανοικτού κώδικα η οποία προσφέρει δυνατότητες διεπαφής με την γλώσσα Python για τα τεχνητά νευρωνικά δίκτυα. Η συγκεκριμένη βιβλιοθήκη αποτελεί διεπαφή για

την βιβλιοθήκη Tensorflow κάνοντας την χρήση της πιο φιλική και επεκτάσιμη στα χέρια του χρήστη.

Οι δύο παραπάνω βιβλιοθήκες αποτέλεσαν σημαντικό πυλώνα για την ανάπτυξη των νευρωνικών δικτύων απλής κατασκευής, χρησιμοποιώντας από έναν απλό υπολογιστή έως και έναν 8-bit μικροελεγκτή για την ανάπτυξη σημαντικών καινοτομιών στον κόσμο της τεχνολογίας. Αυτό έγινε εφικτό με την χρήση της γλώσσας python η οποία είναι υψηλού επιπέδου και διαθέτει μια μεγάλη ποικιλία από βιβλιοθήκες, επιτρέποντας έτσι σε πολλούς χρήστες να αξιοποιήσουν εύκολα την βιβλιοθήκη tensorflow και έτσι να δώσουν μια άλλη διάσταση στην κατασκευή τους.

6.3.1 Εκπαίδευση του Μοντέλου

Χρησιμοποιώντας πάλι ένα jupyter notebook, Θα σχηματίσουμε τον κώδικα με τον οποίο θα εκπαιδεύσουμε το μοντέλο μηχανικής μάθησης, ενώ παράλληλα, θα το εξετάσουμε ώστε να επαληθεύσουμε τη αποτελεσματικότητα του και αν οι καταστάσεις ορθής λειτουργίας και ανώμαλες καταστάσεις είναι διακριτές. Τον έλεγχο του μοντέλου θα τον κάνουμε προτού το αποθηκεύσουμε σε περίπτωση που χρειαστεί παραπάνω εκπαίδευση ή κάποια άλλη αλλαγή στις ρυθμίσεις με τις οποίες πραγματοποιούμε την εκπαίδευση.

Σαν πρώτο βήμα θα ορίσουμε πόσα από τα δεδομένα μας θα χρησιμοποιηθούν για να πραγματοποιηθεί η εκπαίδευση και πόσα από αυτά θα πραγματοποιήσουν τον ρόλο επαλήθευσης της λειτουργικότητας του μοντέλου.

```
ML
# Settings
dataset_path = 'Motor_Dataset' # Directory where raw accelerometer data is stored

normal_op_list = ['Motor_good']
anomaly_op_list = ['Motor_anomaly']

val_ratio = 0.2 # Percentage of samples that should be held for validation set
test_ratio = 0.2 # Percentage of samples that should be held for test set
raw_scale = 1 # Multiply raw values to fit into integers
sensor_sample_rate = 200 # Hz
desired_sample_rate = 50 # Hz
sample_time = 0.64 # Time (sec) length of each sample
samples_per_file = 128 # Expected number of measurements in each file (truncate to this)
max_measurements = int(sample_time * sensor_sample_rate)
downsample_factor = int(samples_per_file / desired_sample_rate)
win_len = int(max_measurements / downsample_factor)

keras_model_name = 'models\stepper_var_1' # Will be given .h5 suffix
sample_file_name = 'D:\My Files\University\Διπλωματική 2020\Practical\mechanical vibration stepper logs\MPU-6050 data\test_auto\test
samples\var_1_normal_anomaly_samples' # Will be given .npz suffix
rep_dataset_name = 'D:\My Files\University\Διπλωματική 2020\Practical\mechanical vibration stepper logs\MPU-6050 data\test_auto\test
samples\var_1_normal_anomaly_test_set' # Will be given .npz suffix

print('Max measurements per file:', max_measurements)
print('Downsample factor:', downsample_factor)
print('Window length:', win_len)

Max measurements per file: 128
Downsample factor: 2
Window length: 64
```

Εικόνα 6.3.1
Κατανομή Dataset και γενικές ρυθμίσεις

```
▶ MI
# Create list of filenames
def createFilenameList(op_list):

    # Extract paths and filenames in each directory
    op_filenames = []
    num_samples = 0
    for index, target in enumerate(op_list):
        samples_in_dir = listdir(join(dataset_path, target))
        samples_in_dir = [join(dataset_path, target, sample) for sample in samples_in_dir]
        op_filenames.append(samples_in_dir)

    # Flatten list
    return [item for sublist in op_filenames for item in sublist]
```

```
▶ MI
# Create normal and anomaly filename lists
normal_op_filenames = createFilenameList(normal_op_list)
anomaly_op_filenames = createFilenameList(anomaly_op_list)
print('Number of normal samples:', len(normal_op_filenames))
print('Number of anomaly samples:', len(anomaly_op_filenames))
```

Number of normal samples: 889
Number of anomaly samples: 1453

Εικόνα 6.3.2
Δημιουργία λίστας και καταμέτρηση όλων των δεδομένων
του Dataset που έχουν μορφή αρχείου .csv

```
▶ MI
# Shuffle lists
random.shuffle(normal_op_filenames)
random.shuffle(anomaly_op_filenames)
```

```
▶ MI
# Calculate validation and test set sizes
val_set_size = int(len(normal_op_filenames) * val_ratio)
test_set_size = int(len(normal_op_filenames) * test_ratio)
```

Εικόνα 6.3.3
Ανάμειξη των δεδομένων της κάθε λίστας, ώστε να
αξιοποιηθούν περισσότερες περιπτώσεις κατά το στάδιο
της εκπαίδευσης

```
▶ MI
# Function: extract Variance features from sample
def extract_features(sample, max_measurements=0, scale=1):

    features = []

    # Truncate sample
    if max_measurements == 0:
        max_measurements = sample.shape[0]
    sample = sample[0:max_measurements]

    # Scale sample
    sample = scale * sample
    # Variance
    features.append(np.var(sample, axis=0))

    return np.array(features).flatten()#flatten the multidimensional array
                                     #into one dimension |
```

Εικόνα 6.3.4
Εξαγωγή χαρακτηριστικών διαφοροποίησης

```
▶ MI
# Break dataset apart into train, validation, and test sets
num_samples = len(normal_op_filenames)
filenames_val = normal_op_filenames[:val_set_size]
filenames_test = normal_op_filenames[val_set_size:(val_set_size + test_set_size)]
filenames_train = normal_op_filenames[(val_set_size + test_set_size):]

# Print out number of samples in each set
print('Number of training samples:', len(filenames_train))
print('Number of validation samples:', len(filenames_val))
print('Number of test samples:', len(filenames_test))

# Check that our splits add up correctly
assert(len(filenames_train) + len(filenames_val) + len(filenames_test)) == num_samples

Number of training samples: 535
Number of validation samples: 177
Number of test samples: 177
```

Εικόνα 6.3.5
Διαχωρισμός δεδομένων σε δείγματα εκπαίδευση,
επαλήθευσης και δοκιμής

```

> MI
# Build model
# Based on: https://medium.com/@curiously/
credit-card-fraud-detection-using-autoencoders-in-keras-tensorflow-for-hackers-part-vii-20e0c85301bd
encoding_dim = 2 # Number of nodes in first layer
model = models.Sequential([
    layers.InputLayer(input_shape=sample_shape),
    layers.Dense(encoding_dim, activation='relu'),
    layers.Dropout(0.2),

    layers.Dense(*sample_shape, activation='relu')
])

# Display model
model.summary()

Model: "sequential_2"

```

| Layer (type) | Output Shape | Param # |
|---------------------|--------------|---------|
| dense_4 (Dense) | (None, 2) | 8 |
| dropout_2 (Dropout) | (None, 2) | 0 |
| dense_5 (Dense) | (None, 3) | 9 |

```

Total params: 17
Trainable params: 17
Non-trainable params: 0

```

Εικόνα 6.3.6
Δημιουργία Autoencoder με ένα ενδιάμεσο Bottleneck layer

```

> MI
# Add training parameters to model
model.compile(optimizer='Adam',
              loss='mse')

> MI
# Train model (note Y labels are same as inputs, X)
history = model.fit(x_train,
                   x_train,
                   epochs=2000,
                   batch_size=100,
                   validation_data=(x_val, x_val),
                   verbose=1)

```

```

-----
6/6 [=====] - 0s 3ms/step - loss: 8.7458e-06 - val_loss: 9.7051e-06
Epoch 1989/2000
6/6 [=====] - 0s 3ms/step - loss: 8.4518e-06 - val_loss: 9.6763e-06
Epoch 1990/2000
6/6 [=====] - 0s 3ms/step - loss: 8.7501e-06 - val_loss: 9.7082e-06
Epoch 1991/2000
6/6 [=====] - 0s 3ms/step - loss: 8.7376e-06 - val_loss: 9.8060e-06
Epoch 1992/2000
6/6 [=====] - 0s 3ms/step - loss: 9.1291e-06 - val_loss: 9.7113e-06
Epoch 1993/2000
6/6 [=====] - 0s 3ms/step - loss: 8.7410e-06 - val_loss: 9.9231e-06
Epoch 1994/2000
6/6 [=====] - 0s 3ms/step - loss: 8.8425e-06 - val_loss: 9.7671e-06
Epoch 1995/2000
6/6 [=====] - 0s 3ms/step - loss: 8.5837e-06 - val_loss: 9.6823e-06
Epoch 1996/2000
6/6 [=====] - 0s 3ms/step - loss: 8.6832e-06 - val_loss: 9.8133e-06
Epoch 1997/2000
6/6 [=====] - 0s 3ms/step - loss: 8.6388e-06 - val_loss: 9.7382e-06
Epoch 1998/2000
6/6 [=====] - 0s 3ms/step - loss: 8.9941e-06 - val_loss: 9.6913e-06
Epoch 1999/2000
6/6 [=====] - 0s 3ms/step - loss: 8.7898e-06 - val_loss: 9.7720e-06
Epoch 2000/2000
6/6 [=====] - 0s 3ms/step - loss: 8.8909e-06 - val_loss: 9.7643e-06

```

Εικόνα 6.3.7
Εκπαίδευση του μοντέλου

```

> ML

# Plot results
loss = history.history['loss']
val_loss = history.history['val_loss']

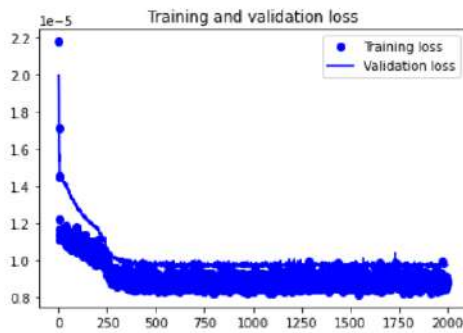
epochs = range(1, len(loss) + 1)

plt.figure()

plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()

plt.show()

```



Εικόνα 6.3.8
Γραφική παρουσίαση της διαδικασίας εκπαίδευσης

```

> ML

# Calculate MSE from validation set
predictions = model.predict(x_val)
normal_mse = np.mean(np.power(x_val - predictions, 2), axis=1)
print('Average MSE for normal validation set:', np.average(normal_mse))
print('Standard deviation of MSE for normal validation set:', np.std(normal_mse))
print('Recommended threshold (3x std dev + avg):', (3*np.std(normal_mse)) + np.average(normal_mse))
fig, ax = plt.subplots(1,1)
ax.hist(normal_mse, bins=20, label='normal', color='blue', alpha=0.7)

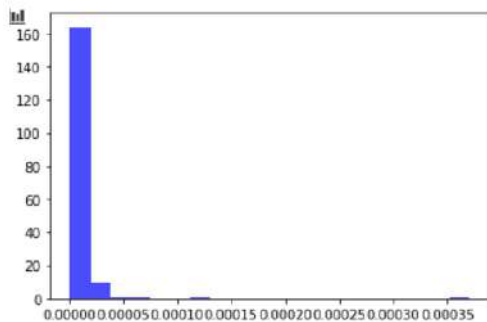
```

Average MSE for normal validation set: 9.764259172784184e-06
Standard deviation of MSE for normal validation set: 2.9762988303499013e-05
Recommended threshold (3x std dev + avg): 9.905322408328123e-05

```

(array([[164., 9., 1., 1., 0., 0., 1., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 1.]],
array([9.94175471e-07, 1.94399660e-05, 3.78857565e-05, 5.63315470e-05,
        7.47733374e-05, 9.32231279e-05, 1.11668918e-04, 1.30114709e-04,
        1.48560499e-04, 1.67006290e-04, 1.85452080e-04, 2.03897871e-04,
        2.22343661e-04, 2.40789452e-04, 2.59235242e-04, 2.77681033e-04,
        2.96126823e-04, 3.14572614e-04, 3.33018404e-04, 3.51464195e-04,
        3.69909985e-04]),
<BarContainer object of 20 artists>)

```



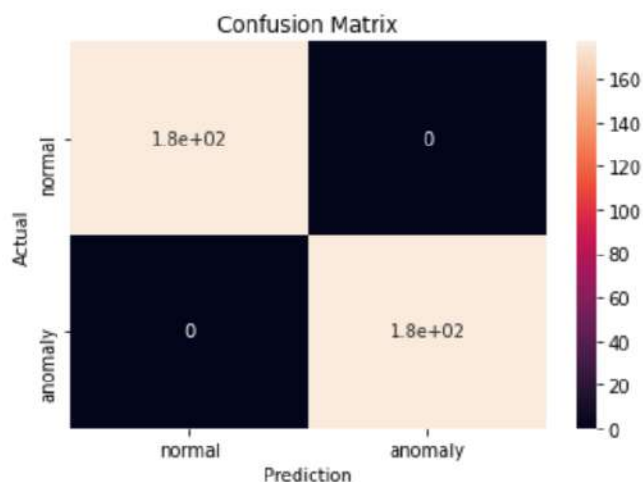
Εικόνα 6.3.9
Υπολογισμός μέσου τετραγωνικού σφάλματος


```

# Create confusion matrix
df_cm = pd.DataFrame(cm, index=['normal', 'anomaly'], columns=['normal', 'anomaly'])
plt.figure()
sn.heatmap(df_cm, annot=True)
plt.title('Confusion Matrix')
plt.xlabel('Prediction')
plt.ylabel('Actual')

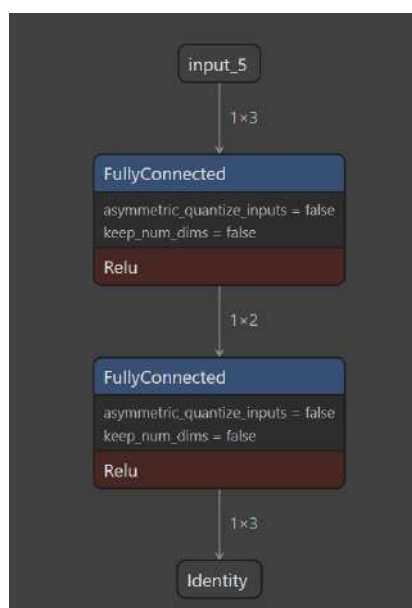
```

Text(33.0, 0.5, 'Actual')



Εικόνα 6.3.11
Η μήτρα σύγχυσης του μοντέλου μηχανικής μάθησης που εκπαιδεύσαμε

Αφού έχουμε ολοκληρώσει και αποθηκεύσει το μοντέλο μπορούμε να επιβεβαιώσουμε τη μορφή του χρησιμοποιώντας το πρόγραμμα Netron, το οποίο μπορεί να το διαβάσει, και έτσι να δούμε τις τρεις στρώσεις του Autoencoder που συνθέσαμε.



Εικόνα 6.3.12
Η μορφή του Autoencoder

6.3.2 Μετατροπή του Μοντέλου

Έχοντας ολοκληρώσει το προηγούμενο στάδιο, αν είμαστε ικανοποιημένοι από τα αποτελέσματα και την αποδοτικότητα του μοντέλου μπορούμε να το αποθηκεύσουμε σε αρχείο .h5, το οποίο εκφράζει ένα μοντέλο που δημιουργήσαμε μέσω της βιβλιοθήκης Keras.

Το επόμενο στάδιο της έρευνας μας περιλαμβάνει δημιουργία μιας διεπαφής, η οποία θα λαμβάνει από κάποια άλλη συσκευή, ένα μικρο-υπολογιστή τα αποτελέσματα του μοντέλου και ορισμένες από τις μετρήσεις, από τα αισθητήρια, που θα βρίσκονται πάνω στους εκτυπωτές. Η συσκευή που θα συλλαμβάνει τα δεδομένα των αισθητηρίων και θα υπολογίζει μέσω του μοντέλου αν υπάρχει κάποια ανώμαλη κατάσταση, θα είναι χαμηλής επεξεργαστικής ισχύος, οπότε επιθυμούμε το μοντέλο να μην το επιβαρύνει και έτσι να μην χρειαστεί να προνοήσουμε σε κάποιου είδους ισχυρότερη ψύξη η οποία πιθανόν να ανέβαζε αρκετά το κόστος της εγκατάστασης των συσκευών. Για να το πετύχουμε, αυτό, πρέπει να το μετατρέψουμε σε Tensorflow lite μοντέλο, ώστε η χαμηλή επεξεργαστική ισχύς να μην προκαλεί υψηλούς λανθάνων χρόνους και το μέγεθος του μοντέλου να είναι σχετικά μικρό ώστε να μην καταλαμβάνει επιπλέον μνήμη στην κύρια μονάδα αποθήκευσης της συσκευής.

```
▶ MI
# Settings
models_path = 'models' # Where we can find the model files (relative path location)
keras_model_name = 'stepper_var_1' # Will be given .h5 suffix
tflite_model_name = 'stepper_var_1' # Will be given .tflite suffix
c_model_name = 'stepper_var_1' # Will be given .h suffix1

▶ MI
# Load model
model = models.load_model(join(models_path, keras_model_name) + '.h5')

▶ MI
# Convert Keras model to a tflite model
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()
open(join(models_path, tflite_model_name) + '.tflite', 'wb').write(tflite_model)

WARNING:tensorflow:From D:\Users\lyk\anaconda3\lib\site-packages\tensorflow\python\training\tracking\tracki
python.keras.engine.training) is deprecated and will be removed in a future version.
Instructions for updating:
This property should not be used in TensorFlow 2.0, as updates are applied automatically.
WARNING:tensorflow:From D:\Users\lyk\anaconda3\lib\site-packages\tensorflow\python\training\tracking\tracki
n.keras.engine.base_layer) is deprecated and will be removed in a future version.
Instructions for updating:
This property should not be used in TensorFlow 2.0, as updates are applied automatically.
INFO:tensorflow:Assets written to: C:\Users\lyk\AppData\Local\Temp\tmpwt5n0re_\assets

1204
```

Εικόνα 6.3.13
Μετατροπή αρχείου Keras σε Tensorflow lite

Κεφάλαιο 7

Διακομιστής Συγκέντρωσης και Απεικόνισης Δεδομένων

Όπως αναφέραμε και στα προηγούμενα κεφάλαια, ο κάθε τρισδιάστατος εκτυπωτής θα έχει ένα δικό του σύστημα επίβλεψης, ώστε να γίνεται αναγνώριση ανωμαλιών αλλά και επίβλεψη χρήσης υλικού. Οι συσκευές που επιβλέπουν τους τρισδιάστατους εκτυπωτές μέσω ενός τοπικού δικτύου θα πρέπει να στέλνουν τα δεδομένα τους κάπου συγκροτημένα. Για να γίνει αυτό πρέπει να χρησιμοποιήσουμε έναν κεντρικό διακομιστή ο οποίος θα έχει τον ρόλο ενός Local server. Εκεί θα πραγματοποιείται η προβολή των δεδομένων μέσω μιας ιστοσελίδας η οποία θα μπορεί να προβληθεί από μία πληθώρα συσκευών, όπως μέσω ενός υπολογιστή ή Smartphone. Έτσι, ο έλεγχος της λειτουργίας των εκτυπωτών, της εγκατάστασης, θα γίνεται πολύ γρήγορα ώστε να εντοπίζουμε γρήγορα και εύκολα κάποιον που θα φέρει πρόβλημα. Πρέπει επίσης, να προνοήσουμε την ενημέρωση βλάβης σε περίπτωση που δεν γίνεται συνεχής παρακολούθηση μέσω της ιστοσελίδας, σαν δικλείδα ασφαλείας, ώστε να γίνεται έγκυρη ενημέρωση.

7.1 Hardware Διακομιστή

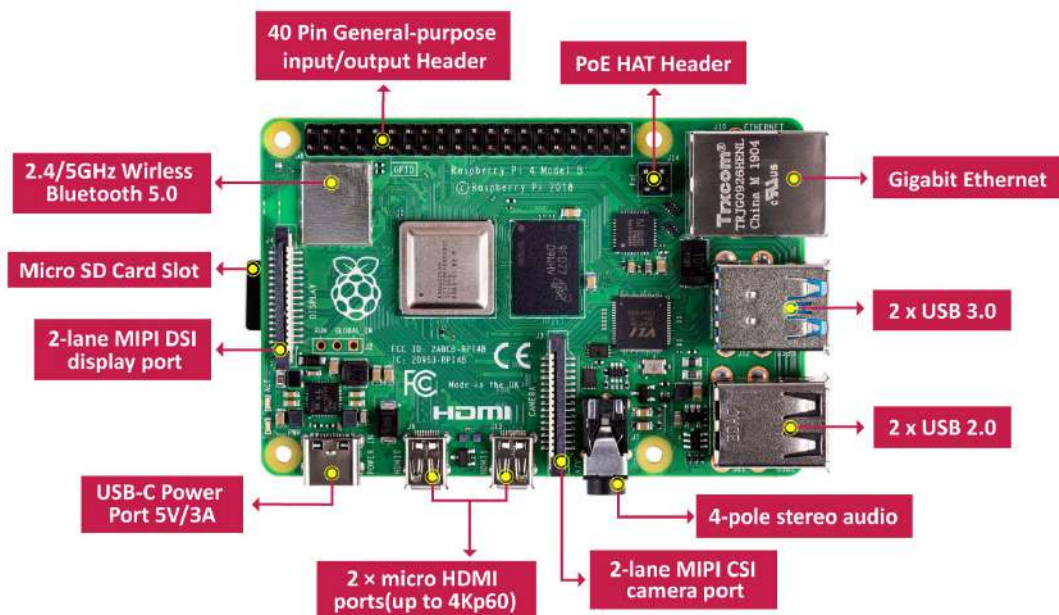
Για την επιλογή των υλικών που θα απαρτίζουν τον διακομιστή θα πρέπει να έχουμε κατασταλάξει στις απαιτήσεις που θέλουμε να έχει το ολοκληρωμένο σύστημα. Μερικά χαρακτηριστικά που επιθυμούμε είναι:

- I. Δυνατότητα σύνδεσης σε τοπικό δίκτυο μέσω ethernet πρωτοκόλλου
- II. Κάποια διανομή Linux
- III. Διαθέσιμη σειριακή επικοινωνία

Για τους σκοπούς της έρευνας, μιας και λειτουργούμε με μία συσκευή θα χρησιμοποιήσουμε έναν μικρό υπολογιστή, το Raspberry Pi 4. Η επιλογή αυτού του συστήματος έγινε λόγω του χαμηλού κόστους του, της ευκολίας προγραμματισμού του και των δυνατοτήτων που προσφέρει.

Πιο συγκεκριμένα στην δική μας περίπτωση διαθέτει 4gb RAM και 32GB αποθηκευτικού χώρου μέσω κάρτας microSD. Όπως είναι λογικό, αν η εγκατάσταση μας αποτελούνταν από πολλά μηχανήματα θα έπρεπε να έχουμε έναν ισχυρότερο server ο οποίος θα μπορούσε να διαχειριστεί το πλήθος των συσκευών και της επεξεργασίας των δεδομένων που παρέχουν.

Ως λειτουργικό θα χρησιμοποιήσουμε το Raspbian το οποίο είναι μία Linux διανομή ειδικά σχεδιασμένη για το Raspberry Pi, η οποία βασίζεται πάνω στο Debian. Ο λόγος που θα το χρησιμοποιούμε είναι επειδή αποτελεί ένα ελαφρύ και δωρεάν λογισμικό. Τα χαρακτηριστικά του Raspbian είναι ιδανικά στην περίπτωσή μας διότι, το λειτουργικό, δεν απαιτεί μεγάλη επεξεργαστική ισχύ, οπότε μπορούμε να αφιερώσουμε και ορισμένους πόρους σε άλλες λειτουργίες που μπορεί να μας ενδιαφέρουν.



Εικόνα 7.1.1
Raspberry pi 4

7.2 Παραμετροποίηση Διακομιστή

Για να παραμετροποιήσουμε το Raspberry Pi, ώστε να λειτουργεί σαν τοπικός διακομιστής ο οποίος θα προβάλλει μέσω μιας ιστοσελίδας όλα τα απαραίτητα δεδομένα που θέλουμε να γνωρίζουμε για τους εκτυπωτές που λειτουργούν στην εγκατάσταση μας, πρέπει πρώτα, να κάνουμε έλεγχο, αν υπάρχει κάποια διαθέσιμη ενημέρωση του λειτουργικού. Αν αυτή υπάρχει θα την εγκαταστήσουμε ώστε το σύστημα μας να είναι πλήρως αναβαθμισμένο, σε περίπτωση που αυτή περιλαμβάνει μια σημαντική διόρθωση. Οι εντολές που θα χρησιμοποιήσουμε για να εγκαταστήσουμε τις αναβαθμίσεις και τα πακέτα που θα χρειαστούμε, θα γίνουν μέσω του Terminal που βρίσκεται στην γραμμή εργαλείων.

```
sudo apt update
```

```
sudo apt full-upgrade
```

Εικόνα 7.2.1
Εντολες αναβάθμισης λειτουργικού Raspbian

Το επόμενο βήμα που θα ακολουθήσουμε έπειτα από την αναβάθμιση του λειτουργικού συστήματος, θα είναι η εγκατάσταση του Apache http server, ο οποίος είναι ένας ανοιχτού κώδικα διακομιστής που έχει την δυνατότητα παροχής περιεχομένου ενός ιστότοπου μέσω του διαδικτύου, δηλαδή, με απλά λόγια, κάνει εφικτή την προβολή μιας ιστοσελίδας όταν καλείται η διεύθυνση του διακομιστή.

```
sudo apt install apache2 -y
```

Εικόνα 7.2.2
Εντολή εγκατάστασης Apache

Για να κάνουμε διαθέσιμες μερικές επιπλέον λειτουργίες, οι οποίες θα μας φανούν χρήσιμες για την επιθυμητή λειτουργία της ιστοσελίδας, θα πρέπει να εγκαταστήσουμε και το πακέτο γλώσσας PHP.

Η PHP είναι μια γλώσσα προγραμματισμού που χρησιμοποιείται στον διακομιστή, η οποία έχει σημαντικό ρόλο στην δημιουργία δυναμικών και διαδραστικών ιστοσελίδων. Αυτή η ιδιαιτερότητα της, την καθιστά “αόρατη” στους χρήστες που προβάλλουν την ιστοσελίδα που λειτουργεί εντός του διακομιστή, σε αντίθεση για παράδειγμα, τις γλώσσες HTML και CSS. Αυτό, την καθιστά ιδανική, για επικοινωνία με βάσεις δεδομένων, την διαχείριση δυναμικού περιεχομένου και την παρακολούθηση συνεδρίων ενός χρήστη. Εμείς θα την αξιοποιήσουμε ώστε να συνδεόμαστε στην τοπική βάση δεδομένων μας και να προβάλλουμε ορισμένες παραμέτρους που μας έχουν σταλεί από τις συσκευές που επιβλέπουν τους εκτυπωτές.

```
sudo apt install php libapache2-mod-php -y
```

Εικόνα 7.2.3
Εντολή εγκατάστασης PHP

Για να δεχόμαστε δεδομένα από τις συσκευές που παρακολουθούν τους τρισδιάστατους εκτυπωτές, θα χρησιμοποιήσουμε το πρωτόκολλο MQTT(Message Queuing Telemetry Transport). Πρόκειται για ένα ελαφρύ σύστημα δημοσίευσης και εγγραφής, που αφορά το application layer της εφαρμογής μας, και αποτελεί ένα απλό πρωτόκολλο ανταλλαγής μηνυμάτων. Αφού ο στόχος μας είναι η χρήση πολλών συσκευών που θα επικοινωνούν με τον κεντρικό διακομιστή, το συγκεκριμένο πρωτόκολλο είναι ιδανικό ώστε να γίνει μείωση φόρτου του δικτύου, αυτό επιτυγχάνεται και λόγω του μικρού Keep-Alive χρόνου, όπου σε άλλη περίπτωση θα μπορούσε να οδηγήσει σε μεγάλες καθυστερήσεις στις λήψεις μηνυμάτων.

Για να παραμετροποιήσουμε τον διακομιστή μας ώστε να δέχεται μηνύματα MQTT, θα πρέπει να εγκαταστήσουμε έναν MQTT μεσίτη, Ο οποίος θα έχει σαν κεντρικό ρόλο τη λήψη μηνυμάτων από τις συσκευές και να τις δρομολογεί Εκεί που χρειάζεται, στη δική μας περίπτωση θέλουμε τα δεδομένα που λαμβάνουμε να τα αποθηκεύουμε σε μία βάση δεδομένων ώστε να τα προβάλλουμε στην ιστοσελίδα που θα σχεδιάσουμε.

```
sudo apt-get install mosquitto mosquitto-clients -y
```

Εικόνα 7.2.4
Εντολή εγκατάστασης MQTT Broker

Αφού ολοκληρώθηκε η εγκατάσταση αυτού πακέτου, μένει να εγκαταστήσουμε ένα σύστημα διαχείρισης δεδομένων χάρη στο οποίο θα μπορούμε να έχουμε άμεση πρόσβαση στα δεδομένα που θα αποστέλλουν συσκευές. Στην δική μας περίπτωση θα εγκαταστήσουμε την MySQL.

```
sudo apt-get install mysql-server php-mysql
```

Εικόνα 7.2.5
Εντολή εγκατάστασης συστήματος διαχείρισης δεδομένων
και το δομοστοιχείο λειτουργίας με την γλώσσα PHP

Η τελευταία αλλαγή που θα πρέπει να πραγματοποιήσουμε για να ολοκληρωθεί το στάδιο της παραμετροποίησης, είναι η ενεργοποίηση της σειριακής λειτουργίας του Raspberry Pi, διότι για την άμεση ειδοποίηση σοβαρής βλάβης θα χρησιμοποιήσουμε ένα GSM module το οποίο συνδέεται σειριακά. Αυτό μπορεί να γίνει από το Raspberry Pi Configuration.



Εικόνα 7.2.6
Ενεργοποίηση σειριακής λειτουργίας μέσω του Raspberry
Pi configuration tab

Για να μπορέσουμε να αποθηκεύσουμε τα δεδομένα στη βάση μας πρέπει να δημιουργήσουμε εντός της mysql μία βάση δεδομένων και έναν πίνακα με το όνομα Deploy ο οποίος θα έχει όλες τις παραμέτρους οι οποίες έχουν ληφθεί από το σύστημα συλλογής δεδομένων. Από τον πίνακα που θα δημιουργήσουμε, στη συνέχεια, θα χτίσουμε έναν κώδικα ο οποίος θα κοιτάει τις γραμμές του πίνακα, κατά τις πιο πρόσφατες, και θα γεμίζει το περιεχόμενο της σελίδας.

7.3 Σχεδίαση Ιστοσελίδας

Ο λόγος που επιλέχθηκε να σχεδιαστεί μία ιστοσελίδα για την προβολή των δεδομένων των εκτυπωτών, είναι διότι, αυτή μπορεί να προβληθεί σε μία πληθώρα συσκευών οι οποίες έχουν σύνδεση στο διαδίκτυο. Αντίθετα, αν σχεδιάζαμε μία διεπαφή, η οποία προβαλλόταν μόνο εντός του διακομιστή, αλλά χωρίς να υπάρχει τρόπος προβολής σε εξωτερικές συσκευές, αυτό θα ήταν μη αποδοτικό, διότι θα έπρεπε κάποιος εκ του εργατικού δυναμικού συνέχεια να την παρακολουθεί.

Με το παρόν σύστημά θέλουμε να πληρούνται όλες οι τηλεμετρικές απαιτήσεις, που θα χρειαζόμασταν για να σιγουρευτούμε πως το μηχάνημα που παρακολουθείτε είναι λειτουργικό και σε καλή κατάσταση. Θα πρέπει επίσης να υπάρξει ένας τρόπος άμεσης ενημέρωσης, ο οποίος δεν θα καθιστά αναγκαία την συνεχή παρακολούθηση της διαδικτυακής διεπαφής από τον χρήστη.

7.3.1 Κώδικας Διαχείρισης Βάσης Δεδομένων

Έχοντας δημιουργήσει τη βάση δεδομένων και οριστικοποιήσει τις στήλες του πίνακα από τον οποίον θα λαμβάνουμε τα δεδομένα για τη σελίδα μας θα πρέπει να φτιάξουμε και τον κώδικα ο οποίος θα αποκτά πρόσβαση στην βάση, μιας και είναι ασφαλισμένη με κωδικό πρόσβασης. Αφού αποκτήσει πρόσβαση στην βάση δεδομένων θα διαβάσουμε τις τελευταίες 25 καταχωρημένες τιμές, για την κάθε στήλη, με βάση την πιο πρόσφατη, κυρίως για τον σχηματισμό της γραφικής παράστασης, ενώ οι πιο πρόσφατες τιμές των μεταβλητών που δεν ανήκουν σε κάποια γραφική απεικόνιση θα αποθηκεύονται και θα προβάλλονται σαν απλά νούμερα.

Ο κώδικας, θα περιλαμβάνεται εντός ενός αρχείου .php και θα είναι αποθηκευμένος στο ίδιο directory, δηλαδή φάκελο, με την ιστοσελίδα που θα δημιουργήσουμε στην συνέχεια.


```

<?php
session_start();#Session start for value pass

echo "Date ".date('d-m-Y');//Print the Date
echo "Time ".date('H-m-s');//Print the time

$conn = mysqli_connect("localhost","lyk","lyk","exampledb");#Database access settings
//Check the connection to the db
if ($connection->connect_error) {
    die ('    Connection error: '.$connection->connect_error);
}
else {
    echo '    Connection to Database -> OK.<br />'; # For debugging purposes
}

$sql = "SELECT * FROM Deploy ORDER BY Datetime DESC LIMIT 25"; # SQL Query for data gathering
$result = $conn->query($sql); //Save the read values

```

Εικόνα 7.3.1
Κώδικας Database connection, σκελος πρώτο

```

$dbdata = array();
$anomalies = 0; //Anomalies Detected
$normal=0;//Normal operation Counter
$total_fil_cons=0.0;//Total Amount of Filament Consumed
$fil_cons=0.0;//Amount Consumed in a Printing Session
$motor_temp=0.0;//Temperature of extruder stepper motor
$operational=0;//Printer operation status
$nozzle=0.0;//Nozzle Temperature
$bed=0.0;//Heated Bed Temperature
$bool = 0;//First value of db save bool

```

Εικόνα 7.3.2
Κώδικας Database connection, σκελος δεύτερο, δήλωση μεταβλητών

Για να μπορέσουμε να είμαστε συγχρονισμένοι στις δοκιμές μας, θα προβάλουμε μέσω php εντολών την τρέχουσα ημερομηνία και ώρα. Έχοντας ανοιχτό το phpmysadmin μπορούμε να επαληθεύσουμε αν οι πληροφορίες που θα βλέπουμε στην ιστοσελίδα θα είναι οι πιο πρόσφατα αποθηκευμένες και αν αυτές προβάλλονται με σωστή σειρά.

| | Datetime | printerID | Anomalies | Normal | Motor_temp | Nozzle_temp | Bed_temp | T_fil_cons | S_fil_cons | Operating |
|--------------------------|---------------------|-----------|-----------|--------|------------|-------------|----------|------------|------------|-----------|
| <input type="checkbox"/> | 2021-05-14 22:29:46 | 0 | 2 | 0 | 37.24 | 0 | 0 | 0.0017 | 0 | 0 |
| <input type="checkbox"/> | 2021-05-14 22:30:07 | 0 | 7 | 0 | 37.24 | 0 | 0 | 0.0017 | 0 | 1 |
| <input type="checkbox"/> | 2021-05-14 22:30:23 | 0 | 11 | 0 | 37.24 | 0 | 0 | 0.0034 | 0.0017 | 1 |
| <input type="checkbox"/> | 2021-05-16 00:13:30 | 0 | 1 | 0 | 37.28 | 0 | 0 | 0 | 0 | 0 |
| <input type="checkbox"/> | 2021-05-14 22:29:54 | 0 | 4 | 0 | 37.28 | 0 | 0 | 0.0017 | 0 | 1 |
| <input type="checkbox"/> | 2021-05-14 22:30:15 | 0 | 9 | 0 | 37.28 | 0 | 0 | 0.0034 | 0.0017 | 1 |
| <input type="checkbox"/> | 2021-05-14 22:30:02 | 0 | 6 | 0 | 37.28 | 0 | 0 | 0.0017 | 0 | 1 |
| <input type="checkbox"/> | 2021-05-14 22:30:19 | 0 | 10 | 0 | 37.33 | 0 | 0 | 0.0034 | 0.0017 | 1 |

Εικόνα 7.3.
Στήλες βάσης δεδομένων ιστοσελίδας

```

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        $timestamp_rest = substr($row["Datetime"], -8);
        $data[] = array($row['Datetime'], (int)$row['Nozzle_temp']);
        $data1[] = array($row['Datetime'], (int)$row['Bed_temp']);
        $dbdata[]=$row;
        if($sbool==0)// Use a Boolean for saving the latest value
        {
            $anomalies=(int)$row['Anomalies'];
            console.log($anomalies);//console log for debugging purposes
            $normal=(int)$row['Normal'];
            $T_fil_cons=(float)$row['T_fil_cons'];
            $S_fil_cons=(float)$row['S_fil_cons'];
            $motor_temp=(float)$row['Motor_temp'];
            $operational=(int)$row['Operating'];
            $Nozzle=(float)$row['Nozzle_temp'];
            $Bed=(float)$row['Bed_temp'];

            $sbool=1;
            console.log($sbool);
        }
    }
}
else {
    echo "0 results";
}
}

```

Εικόνα 7.3.4
Κώδικας Database connection, σκελος τρίτο, ανάγνωση
πίνακα δεδομένων

Η συλλογή των δεδομένων από την βάση γίνεται μέσω εντολών mysql, όπου αποθηκεύουμε σε πίνακες τα τελευταία, πιο πρόσφατα, δηλαδή, 25 στοιχεία των παραμέτρων που λαμβάνουμε από την συσκευή που πραγματοποιεί την δειγματοληψία μέσω των αισθητηρίων.

```

//save the array of temperature values
//for the google chart api
echo "<script>

    var my_2d = ".json_encode(array_reverse($data)).";
    var my_2d1 = ".json_encode(array_reverse($data1)).";
</script>";

mysqli_close($con);

```

Εικόνα 7.3.5
Κώδικας Database connection, σκελος τέταρτο,
αποθήκευση τιμών σε πίνακα για προβολή σε Chart

Για να χρησιμοποιήσουμε το Google chart API θα πρέπει να μετατρέψουμε τις τιμές που συλλέξαμε από τις στήλες της βάσης δεδομένων, σε μορφή JSON. Τέλος, κλείνουμε την σύνδεση με την βάση.

```

//Save all the non-chart values in session variables
//For easier transfer of variables between separate files
$_SESSION["Anomalies"] = "$anomalies";
$_SESSION["Normal"] = "$normal";
$_SESSION["Motor_temp"] = "$motor_temp";
$_SESSION["Nozzle"] = "$Bed";
$_SESSION["Bed"] = "$Bed";
$_SESSION["T_fil_cons"] = "$T_fil_cons";
$_SESSION["S_fil_cons"] = "$S_fil_cons";
$_SESSION["operational"] = "$operational";

$bool=0;//Reset the boolean

?>

```

Εικόνα 7.3.6
Κώδικας Database connection, σκελος πεμπτό,
αποθήκευση τιμών σε Session μεταβλητές

Για να μπορέσει να υπάρχει καλύτερη διακίνηση μεταβλητών μεταξύ αρχείων, θα αποθηκεύσουμε τις υπόλοιπες τιμές που δεν θέλουμε να προβάλλουμε σε γραφική παράσταση, σε session μεταβλητές.

7.3.2 Κώδικας Ιστοσελίδας

Αφού προσαρμόσαμε τον κώδικα, ώστε να έχει πρόσβαση στην βάση δεδομένων και να αποθηκεύει τις τιμές, με τέτοιο τρόπο, ώστε να μπορέσουμε να έχουμε πρόσβαση σε αυτές από την κύρια ιστοσελίδα που θα φτιάξουμε.

Για την δημιουργία της ιστοσελίδας στην οποία θα έχει πρόσβαση ο χρήστης και θα μπορεί να πλοηγηθεί στους διάφορους τομείς της εγκατάστασης που φιλοξενούν τους τρισδιάστατους εκτυπωτές και να αντλήσει τα κατάλληλα δεδομένα θα χρησιμοποιήσουμε έναν συνδυασμό από γλώσσες προγραμματισμού.

Αρχικά, θα χρησιμοποιήσουμε την γλώσσα HTML(Hypertext Markup Language) για να δομήσουμε την ιστοσελίδα και το περιεχόμενο της. Αυτή θα αποτελέσει τον κύριο “σκελετό” της σχεδίασης της οπτικής διεπαφής. Επίσης, έχει την δυνατότητα να υποβοηθηθεί από άλλες τεχνολογικές μεθόδους, άλλες που αφορούν καθαρά την στυλιστική εμφάνιση και άλλες που έχουν να κάνουν με τον προγραμματισμό(scripting), όπως την γλώσσα PHP που χρησιμοποιούμε στο προηγούμενο κεφάλαιο.

Για να μπορέσουμε να αναβαθμίσουμε τις βασικές στυλιστικές επιλογές που προσφέρει HTML, ώστε η σελίδα μας να είναι πιο φιλική ως προς τον χρήστη, θα χρησιμοποιήσουμε και την γλώσσα CSS(Cascading Style Sheets), η οποία καθοδηγεί τα στοιχεία HTML ως προς τον τρόπο που αυτά παρουσιάζονται σε μία οθόνη, στο χαρτί ή σε κάποιο άλλο πολυμέσο. Η ευελιξία της CSS μας βοηθά να ελέγχουμε τον τρόπο τοποθέτησης των στοιχείων σε παραπάνω από μία σελίδα, με το να καλούμε το source file που έχουμε συνθέσει με αυτήν.

Επειδή θέλουμε η ιστοσελίδα μας να είναι διαδραστική και οι ακραίες τιμές που θα μπορεί να έχει το σύστημα να προβάλλονται με τέτοιο τρόπο ώστε να είναι άμεσα διακριτές,

θα χρησιμοποιήσουμε και την γλώσσα προγραμματισμού javascript. Σε σχέση, με την HTML και την CSS, αυτή έχει την ιδιότητα να δώσει στα στοιχισμένα στοιχεία, της ιστοσελίδας, κάποια λειτουργία ώστε να υπάρχει μεγαλύτερη διαδραστικότητα μεταξύ σελίδας και χρήστη. Πιο συγκεκριμένα, θα χρησιμοποιήσουμε, μέσω αυτής το Google Charts API, η οποία πρόκειται για μια διαδραστική υπηρεσία ιστού που μπορεί και σχεδιάζει γραφικές παραστάσεις με βάση τα δεδομένα που έχουμε προσφέρει. Όπως θα δούμε και στην συνέχεια, σε συνδυασμό της CSS και της javascript θα μπορέσουμε να χρησιμοποιήσουμε χρωματική κωδικοποίηση ώστε να μπορέσουμε να εκφράσουμε διαφορετικές καταστάσεις λειτουργίας.

```
<?php
session_start();
?>

<!DOCTYPE >
<html >
<head>
<!--Webpage title -->
<title>3D Printing Farm Monitor</title>
```

Εικόνα 7.3.7
Κύρια ιστοσελίδα, σκελος πρώτο

Για να μπορέσουμε να μεταφέρουμε τις τιμές που λάβαμε από την βάση δεδομένων σε κάποια γραφική απεικόνιση θα πρέπει να ξεκινήσουμε ένα session, ώστε οι μεταβλητές να μπορούν να αποθηκευτούν σε ένα session variable.

```
<style>
html, body
{
    width: 100%;
    margin: 0px;
    padding: 0px;
    overflow-x: scroll;
}

/* Logo size Css */
p.ex1{
    font-size:40px;
}

p.ex2{
    font-family: arial;
    font-size:40px;
    text-align: center;
}

p.ex3{
    font-family: arial;
    font-size:80px;
    text-align: center;
}
```

Εικόνα 7.3.8
Κύρια ιστοσελίδα, σκελος δεύτερο, style tag

```

/* Css for values presentation */
.rcorners1 {
border-radius: 15px 20px 10px 5px;
padding: 25px;
width: 380px;
height: 150px;
}

.rcorners1:hover {
box-shadow: 0 0 11px rgba(33,33,33,.5);
}

.line_box {
border-radius: 15px 20px 0px 0px;
width: 425px;
height: 70px;
border: 4px solid black;
}

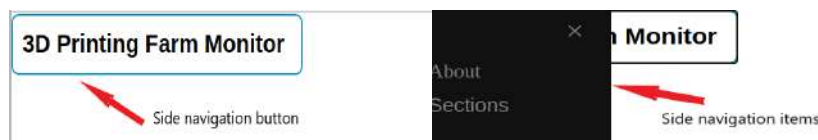
#div1 {
background: linear-gradient(to top right, #00ff99 6%, #ccffcc 100%);//light
}
#div1_fil {
background: linear-gradient(to top right, #00ff99 6%, #ccffcc 100%);//light
}
#div1_temp {
background: linear-gradient(to top right, #00ff99 6%, #ccffcc 100%);//light
}
#div1_op {
background: linear-gradient(to top right, #00ff99 6%, #ccffcc 100%);//light
}
#div2_an {
background: linear-gradient(to top right, #ff3300 6%, #ff9966 100%);//error
}
#div1_stable {
background: linear-gradient(to top right, #00ff99 6%, #ccffcc 100%);//light
}

```

Εικόνα 7.3.9
Κύρια ιστοσελίδα, σκελος τρίτο, style tag

Σχεδιαστικά, επιθυμούμε οι σημαντικές πληροφορίες να φαίνονται έντονα, όπως και αυτές να υποστηρίζονται και από μια χρωματική ένδειξη η οποία θα εκφράζει την ορθή ή την λανθάνουσα λειτουργία. Η χρωματική υποστήριξη των πληροφοριών θα μπορέσει να βοηθήσει, ακόμα και άπειρους στην κατανόηση των πληροφοριών που προβάλλονται.

Κάθε εκτυπωτής έχει την δική του ξεχωριστή σελίδα, όπου θα προβάλλονται οι σημαντικότερες πληροφορίες που συλλέγουμε. Για να μπορέσουμε να πλοηγηθούμε καλύτερα εντός της διεπαφής θα φτιάξουμε ένα side navigation, με σκοπό την καλύτερη οργάνωση των υποσελίδων.



Εικόνα 7.3.10
Side navigation ιστοσελίδας

```

<!--
/* Css for Logo-Button that opens side navigation*/
-->
    .button {
        height : 70px;
        width : 400px;
        background-color: #4CAF50;
        border: none;
        color: white;
        padding: 0px 0px;
        text-align: center;
        text-decoration: none;
        display: inline-block;
        font-size: 10px;
        margin: 2px 1px;
        transition-duration: 0.4s;
        cursor: pointer;
        border-radius: 10px;
    }

    .button1 {
        background-color: white;
        color: black;
        border: 2px solid #008CBA;
    }

    .button1:hover {
        background-color: #008CBA;
        color: white;
    }

```

Εικόνα 7.3.11
Κύρια ιστοσελίδα, σκελος τέταρτο, style tag

```

.sidenav {
    height: 100%;
    width: 0;
    position: fixed;
    z-index: 1;
    top: 0;
    left: 0;
    background-color: #111;
    overflow-x: hidden;
    transition: 0.5s;
    padding-top: 60px;
}

.sidenav a {
    padding: 8px 8px 8px 32px;
    text-decoration: none;
    font-size: 25px;
    color: #818181;
    display: block;
    transition: 0.3s;
}

.sidenav a:hover {
    color: #f1f1f1;
}

.sidenav .closebtn {
    position: absolute;
    top: 0;
    right: 25px;
    font-size: 36px;
    margin-left: 50px;
}

@media screen and (max-height: 450px) {
    .sidenav {padding-top: 15px;}
    .sidenav a {font-size: 18px;}
}

/* Dropdown Button */
.dropbtn {
    background-color: #111;
    color: white;
    padding: 0px;
    font-size: 12px;
    border: none;
}
</style>

```

Εικόνα 7.3.12
Κύρια ιστοσελίδα, σκελος πέμπτο, style tag

```

/* Dropdown Content (Hidden by Default) */
.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f1f1f1;
  min-width: 160px;
  box-shadow: 0px 8px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

/* Links inside the dropdown */
.dropdown-content a {
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
}

/* Change color of dropdown links on hover */
.dropdown-content a:hover {background-color: #ddd;}

/* Show the dropdown menu on hover */
.dropdown:hover .dropdown-content {display: block;}

/* Change the background color of the dropdown button when the dropdown content is shown */
.dropdown:hover .dropbtn {background-color: #111;}

```

Εικόνα 7.3.13
Κύρια ιστοσελίδα, σκελος έκτο, style tag

```

<meta http-equiv="refresh" content="5" >
<meta name="viewport" content="width=1600, initial-scale=1">

<script>
/* Loop through all dropdown buttons to toggle between hiding and showing its dropdown content - This allows the user to have multiple dropdowns without any conflict */
var dropdown = document.getElementsByClassName("dropdown-btn");
var i;

for (i = 0; i < dropdown.length; i++) {
  dropdown[i].addEventListener("click", function() {
    this.classList.toggle("active");
    var dropdownContent = this.nextElementSibling;
    if (dropdownContent.style.display === "block") {
      dropdownContent.style.display = "none";
    } else {
      dropdownContent.style.display = "block";
    }
  });
}
</script>

```

Εικόνα 7.3.14
Κύρια ιστοσελίδα, σκελος έβδομο

```

<script src="http://code.jquery.com/jquery-latest.js"></script>
<script>
//refresh values from db
$(document).ready(function(){
  $("#div_refresh").load("data.php");
  setInterval(function() {
    $("#div_refresh").load("data.php");
  }, 1000);
});
</script>

```

Εικόνα 7.3.15
Κύρια ιστοσελίδα, σκελος ογδοο, ανανέωση τιμών από την
βάση δεδομένων

```

<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">

    // Load the Visualization API and the corechart package.
    google.charts.load('current', {packages: ['corechart']});
    google.charts.setOnLoadCallback(drawChart);

    function drawChart() {

        // Create the data table.
        var data = new google.visualization.DataTable();
        data.addColumn('string', 'Datetime');
        data.addColumn('number', 'Nozzle Temperature');

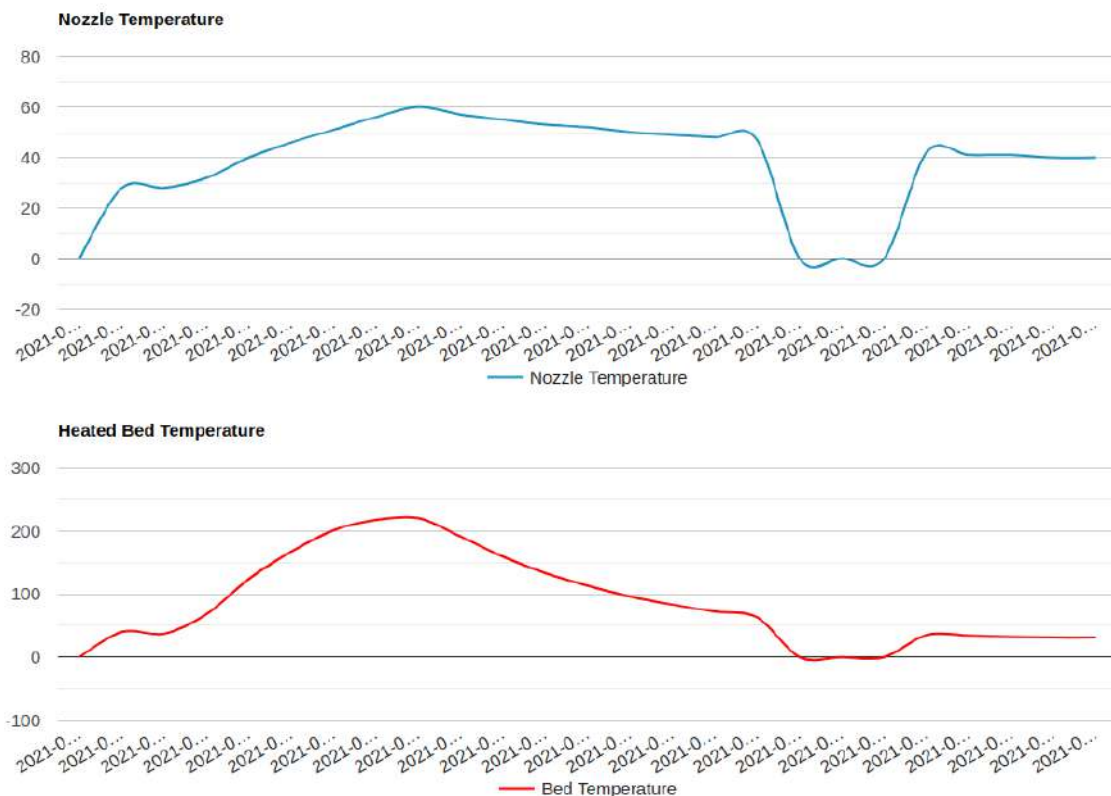
        for(i = 0; i < my_2d.length; i++)
        data.addRow([my_2d[i][0], parseInt(my_2d[i][1])]);

        var data1 = new google.visualization.DataTable();
        data1.addColumn('string', 'Datetime');
        data1.addColumn('number', 'Bed Temperature');

        for(i = 0; i < my_2d1.length; i++)
        data1.addRow([my_2d1[i][0], parseInt(my_2d1[i][1])]);

```

Εικόνα 7.3.16
Κύρια ιστοσελίδα, σκελος ένατο, χρήση Google Chart API



Εικόνα 8.3.17
Γραφικές παραστάσεις θερμοκρασίας μέσω Google Chart API


```

var options = {
  colors: ['#1c91c0'],
  title: 'Nozzle Temperature',
  curveType: 'function',
  width: 1200,
  height: 350,
  legend: { position: 'bottom' }
};
var options1 = {
  colors: ['red'],
  title: 'Heated Bed Temperature',
  curveType: 'function',
  width: 1200,
  height: 350,
  legend: { position: 'bottom' }
};
var chart = new google.visualization.LineChart(document.getElementById('curve_chart'));
var chart1 = new google.visualization.LineChart(document.getElementById('curve_chart1'));
chart.draw(data, options);
chart1.draw(data1, options1);
}

```

Εικόνα 7.3.18
Κύρια ιστοσελίδα, σκελος δέκατο, ονομασία αξόνων

```

//pass session variables to local variables
var Anomalies = "<?php echo $_SESSION['Anomalies'];?>";
var Filament = "<?php echo $_SESSION['T_fil_cons'];?>";
var temp = "<?php echo $_SESSION['Motor_temp'];?>";
var operational = "<?php echo $_SESSION['operational'];?>";

//if anomalies passed the threshold value change the border color
function colour_change_by_val_An()
{
  if (Anomalies < 14){
    document.getElementById('div1').id = 'div1';
    var val = 1;
    console.log(val);
  }
  else if(Anomalies >= 14){
    document.getElementById('div1').id = 'div2_an';
    var val = 2;
    console.log(val);
  }
}

//if filament consumption passed the threshold value change the border color
function colour_change_by_val_fil()
{
  if (Filament < 29){
    document.getElementById('div1_fil').id = 'div1_fil';
    var val = 1;
    console.log(val);
  }
  else if(Filament >= 29){
    document.getElementById('div1_fil').id = 'div2_an';
    var val = 2;
    console.log(val);
  }
}
}

```

Εικόνα 7.3.19
Κύρια ιστοσελίδα, σκελος ενδέκατο

```

//if extruder motor temperature passed the threshold value change the border color
function colour_change_by_val_temp()
{
    if (temp < 10){
        document.getElementById('div1_temp').id = 'div1_temp';
        var val = 1;
        console.log(val);
    }
    else if(temp >= 10){
        document.getElementById('div1_temp').id = 'div2_an';
        var val = 2;
        console.log(val);
    }
}

//if printer is operational change the border color
function colour_change_by_val_op()
{
    if (operational >= 1){
        document.getElementById('div1_op').id = 'div1_op';
        var val = 1;
        console.log(val);
    }
    else if(operational < 1){
        document.getElementById('div1_op').id = 'div2_an';
        var val = 2;
        console.log(val);
    }
}
}
</script>

```

Εικόνα 7.3.20
Κύρια ιστοσελίδα, σκελος δωδέκατο, αλλαγή χρώματος
Background των τιμών που μας ενδιαφέρουν

| | |
|--------------------------------|----------------------------------|
| Anomalies Detected | Normal Conditions |
| 5 | 3 |
| Total Consumed Filament | Session Filament Consumed |
| 0m | 0m |
| Motor(E) Temperature | Operational |
| 28.86 °C | |

Εικόνα 7.3.21
Οι μεταβλητές που προβάλλονται στην ιστοσελίδα

```

<script>
function openNav() {
  document.getElementById("mySidenav").style.width = "250px";
}

function closeNav() {
  document.getElementById("mySidenav").style.width = "0";
}
</script>

</head>
<body>

  <div id="mySidenav" class="sidenav">
    <a href="javascript:void(0)" class="closebtn" onclick="closeNav()">&times;</a>

    <a href="#">About</a>
    <div class="dropdown">
      <button class="dropbtn"><a href="#">Sections</a></button>
      <div class="dropdown-content">
        <a href="#">Section A</a>
        <a href="#">Section B</a>
        <a href="#">Section C</a>
      </div>
    </div>
  </div>
</div>
</div>
<!--
Google Chart Placement
-->
<div style="position:relative; left:-80px; top:0px;" id="curve_chart"></div>
<div style="position:relative; left:-80px; top:0px;" id="curve_chart1"></div>

<!--
Anomalies
-->
<div style="position:absolute; left:1000px; top:80px;">
<p class = "rcorners1" id="div1"></p>
</div>

<!--
normals
-->
<div style="position:absolute; left:1400px; top:80px;">
<p class = "rcorners1" id="div1_stable"></p>
<!--
Total Filament
-->
</div>
<div style="position:absolute; left:1000px; top:290px;">
<p class = "rcorners1" id="div1_fil"></p>
</div>

<!--
Session Filament
-->
<div style="position:absolute; left:1400px; top:290px;">
<p class = "rcorners1" id="div1_stable"></p>
</div>

<!--
Motor Temp
-->
<div style="position:absolute; left:1000px; top:500px;">
<p class = "rcorners1" id="div1_temp"></p>
</div>

<!--
Operational
-->
<div style="position:absolute; left:1400px; top:500px;">
<p class = "rcorners1" id="div1_op"></p>
</div>

```

Εικόνα 7.3.22
Κύρια ιστοσελίδα, σκελος δεκατο-τρίτο, τοποθέτηση πμίων
στην σελίδα στοιχισμένα

```

<div style="position:absolute; left:0px; top:780px;">
<div id = "div_refresh"></div>
</div>
<!--
execute colour change according to value
-->
<script>colour_change_by_val_An();colour_change_by_val_fil();colour_change_by_val_temp();colour_change_by_val_op();</script>
Place value inside of coloured box
-->

<div style="position:absolute ; left:1020px; top:70px;"><p class="ex2"><b>Anomalies Detected</b></p></div>
<div style="position:absolute ; left:1190px; top:100px;"><p class="ex3"><b>
<?php
// Echo session variables that were set on previous page
echo $_SESSION["Anomalies"] . "<br>";
?></b></p>
</div>
<div style="position:absolute ; left:999px; top:80px;">
<p class="line_box" ></p>
</div>

<div style="position:absolute ; left:1490px; top:70px;"><p class="ex2"><b>Normal Conditions</b></p></div>
<div style="position:absolute ; left:1620px; top:100px;"><p class="ex3"><b>
<?php
// Echo session variables that were set on previous page
echo $_SESSION["Normal"] . "<br>";
?></b></p>
</div>
<div style="position:absolute ; left:1459px; top:80px;">
<p class="line_box" ></p>
</div>

<div style="position:absolute ; left:1060px; top:265px;"><p class="ex2"><b>Total Consumed </b></p></div>
<div style="position:absolute ; left:1130px; top:300px;"> <p class="ex2"><b>Filament</b></p></div>
<div style="position:absolute ; left:1100px; top:320px;"><p class="ex3"><b>
<?php
// Echo session variables that were set on previous page
echo $_SESSION["T_fil_cons"] . "m<br>";
?></b></p>
</div>
<div style="position:absolute ; left:999px; top:290px;">
<p class="line_box" ></p>
</div>

<div style="position:absolute ; left:1510px; top:265px;"><p class="ex2"><b>Session Filament</b></p></div>
<div style="position:absolute ; left:1570px; top:300px;"> <p class="ex2"><b>Consumed</b></p></div>
<div style="position:absolute ;text-align:center; left:1560px; top:320px;"><p class="ex3"><b>
<?php
// Echo session variables that were set on previous page
echo $_SESSION["S_fil_cons"] . "m<br>";
?></b></p>
</div>
<div style="position:absolute ; left:1459px; top:290px;">
<p class="line_box" ></p>
</div>

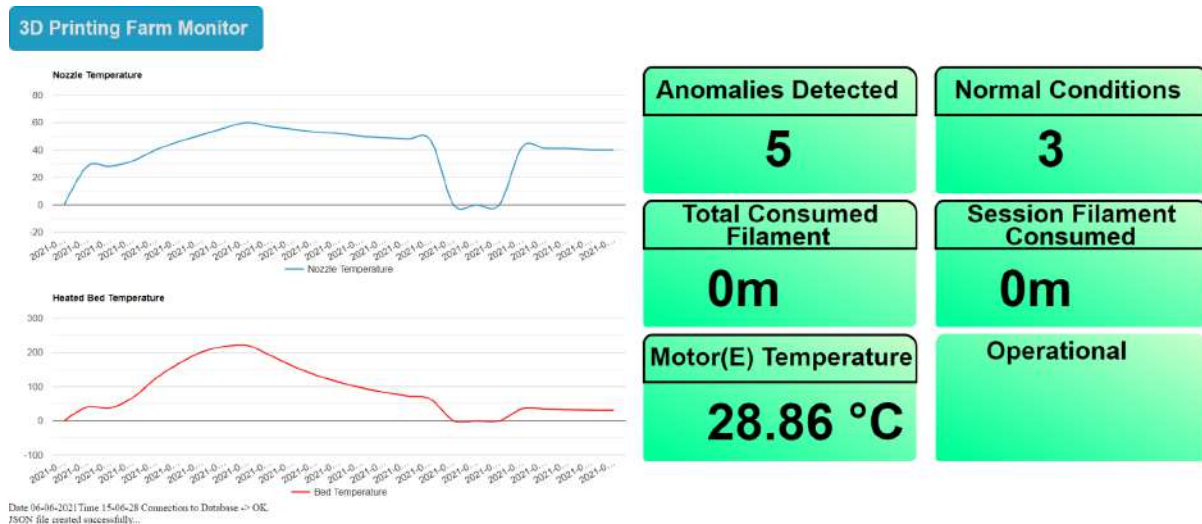
<div style="position:absolute ; left:1540px; top:480px;"><p class="ex2"><b>Operational</b></p></div>

<div style="position:absolute ; left:1010px; top:490px;"><p class="ex2"><b>Motor(E) Temperature</b></p></div>
<div style="position:absolute ; text-align:center; left:1100px; top:530px;"><p class="ex3"><b>
<?php
// Echo session variables that were set on previous page
echo $_SESSION["Motor_temp"] . " °C <br>";

?></b></p>
</div>
<div style="position:absolute ; left:999px; top:500px;">
<p class="line_box" ></p>
</div>
</body>
</html>

```

Εικόνα 7.3.23
Κύρια ιστοσελίδα, σκελεός δεκατο-τέταρτο



Εικόνα 7.3.24
Τελική κύρια ιστοσελίδα

7.4 Συλλογή δεδομένων

Έχοντας σχηματίσει την βασική υποδομή της ιστοσελίδας, πλέον μπορούμε να υλοποιήσουμε και τον τρόπο συλλογής αλλά και αποθήκευσης δεδομένων. Η συλλογή πραγματοποιείται μέσω του συστήματος που έχουμε φτιάξει το οποίο φέρει τα αισθητήρια που θεωρούμε ότι πρέπει να υπάρχουν για την ορθότερη αναγνώριση των καταστάσεων του τρισδιάστατου εκτυπωτή.

Η κάθε συσκευή καλείται να αποθηκεύσει και να στείλει τα δεδομένα που σύλλεξε αξιοποιώντας κάποιο πρωτόκολλο αξιοποιώντας ένα τοπικό δίκτυο. Όπως αναφέραμε και προηγουμένως θα χρησιμοποιήσουμε το πρωτόκολλο MQTT, το οποίο κρίναμε Πώς είναι η βέλτιστη επιλογή για την εφαρμογή μας. Συγκεκριμένα, βασίζεται πάνω στο πρωτόκολλο TCP/IP και πρόκειται για ένα ελαφρύ και ευέλικτο πρωτόκολλο δικτύου το οποίο θα μας δώσει την ελευθερία εξέλιξης σε περίπτωση που επιθυμούμε την περαιτέρω ανάπτυξη του συγκεκριμένου συστήματος. Για την εφαρμογή μας επιθυμούμε ταχεία αποστολή των δεδομένων που συλλέξαμε μέσω της συσκευής έτσι ώστε να γίνεται άμεση απεικόνιση της οποιασδήποτε κατάστασης και αντίστοιχη ενημέρωση σε περίπτωση κάποιου σοβαρού σφάλματος. Το συγκεκριμένο πρωτόκολλο, είναι ιδανικό, μιας και τα πακέτα είναι μικρά αλλά και η σύνδεση μεταξύ client και Broker παραμένει συνέχεια ανοιχτή, καταφέροντας έτσι μια πολύ γρήγορη μετάδοση μηνυμάτων.

Για τον διακομιστή μας, θα πρέπει αν φτιάξουμε ένα πρόγραμμα το οποίο θα λαμβάνει τα δεδομένα μέσω του παραπάνω πρωτοκόλλου και έτσι να τα αποθηκεύει στην βάση δεδομένων, όπου και τελικά θα προβάλλονται στην ιστοσελίδα μας. Για ακραίες περιπτώσεις που θα υπάρχει ένα έντονο σφάλμα, θα χρησιμοποιήσουμε μια συσκευή GSM, με σκοπό την αποστολή μηνύματος, ώστε να γίνει άμεσα επέμβαση στο πρόβλημα από κάποιο τεχνικό. Για την συγγραφή του παρακάτω κώδικα χρησιμοποιήθηκε την γλώσσα Python.

```

import paho.mqtt.client as mqtt

import time,sys
import datetime
import MySQLdb

import json

import serial #GSM Control

MQTT_SERVER = "169.254.199.17"
MQTT_PATH = "test_channel"

GSM_PORT = "/dev/ttyAMA0"

GSM= serial.Serial(GSM_PORT, baudrate=9600, timeout = 0.5)

#START GSM
GSM.write(b'AT'+'\r\n')
time.sleep(2)
GSM.write(b'AT+CPIN=9999'+'\r\n')#Enter Sim card's pin number, future disable pin number
time.sleep(2)
GSM.write("AT+CMGF=1\r") # set to text mode
time.sleep(3)

```

Εικόνα 7.4.1
Κώδικας συλλογής και αποθήκευσης δεδομένων, σκέλος πρώτο

```

# The callback for when the client receives a connect response from the server.
def on_connect(client, userdata, flags, rc):
    print("connected with result code "+str(rc))

    # on_connect() means that if we lose the connection and reconnect then subscriptions will be renewed.
    client.subscribe(MQTT_PATH)

# The callback for when a PUBLISH message is received from the server.
def on_message(client, userdata, msg):
    print(msg.topic+" "+str(msg.payload))
    data = json.loads(msg.payload)
    global ID
    global mysql_space_counter
    ID = int(data["printerID"])
    Anomalies = int(data["Anomalies Counter"])
    Normal = int(data["Normal Counter"])
    try:
        Motor_t = float(data["Motor Temp"])
        Nozzle_t = float(data["Nozzle Temp"])
    except:
        Motor_t = 0
        Nozzle_t = 0
    Bed_t = float(data["Bed Temp"])
    T_fil_cons = float(data["T_Fil Consumed"])
    S_fil_cons = float(data["S_Fil Consumed"])
    Operation = int(data["Operation"])

    print(ID)
    print(Anomalies)
    print(Normal)
    print(Motor_t)
    print(Nozzle_t)
    print(Bed_t)
    print(Operation)

    if ID != 1:
        print("hello")

    db = MySQLdb.connect('127.0.0.1', 'lyk', 'lyk', 'exampledb')

    now = datetime.datetime.now()
    date = str(now.strftime("%Y-%m-%d %H:%M:%S"))

    cursor = db.cursor()

    cursor.execute("select count(*) from Deploy")

    print(list(cursor))
    print(mysql_space_counter)
    if mysql_space_counter > 200 :
        print("clearing up some space.....")
        cursor.execute("DELETE FROM Deploy ORDER BY datetime DESC LIMIT 200")
        mysql_space_counter = 0 #reset after cleaning

    cursor.execute("INSERT INTO Deploy VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s)", (date, ID, Anomalies, Normal, Motor_t, Nozzle_t, Bed_t, T_fil_cons, S_fil_cons, Operation))

    db.commit()
    mysql_space_counter += 1#up by one for every upload, clean after 200 measurements

    print("committed")

    db.close()

    if Anomalies>3:
        GSM.write("AT+CMGS="+306999029979+"\r")
        time.sleep(3)
        msg = "Printer with ID: " + str(ID) + "\r\nAnomalous Behavior\r\n" + date + "\r\nPrinter will Halt, Please attend section: 1A"
        print "Sending SMS Anomalies Detected with printer ID : " + str(ID)
        GSM.flush()
        GSM.write(msg + chr(26))

    clean = 0
    Anomalies = 0

client = mqtt.Client()
client.connect(MQTT_SERVER, 1883, 60)

client.on_connect = on_connect
client.on_message = on_message
#client.on_message = on_message

print("clean", ID)

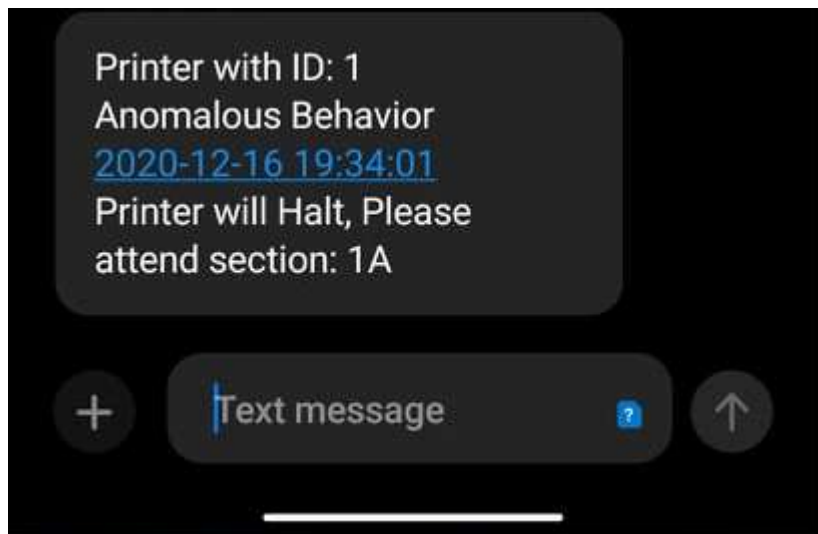
client.loop_forever()

```

Εικόνα 7.4.2
Κώδικας συλλογής και αποθήκευσης δεδομένων, σκέλος δεύτερο

Αφού λάβουμε τα δεδομένα τα ανεβάζουμε στην βάση δεδομένων στις αντίστοιχες στήλες τους. Επιπρόσθετα φροντίζουμε η βάση δεδομένων μας να μην γεμίσει υπερβολικά εξοικονομώντας έτσι σημαντικό χώρο στη μνήμη του μικρο-υπολογιστή, δηλαδή, όταν ξεπεράσουμε ένα ορισμένο πλήθος γραμμών, θα διαγράψουμε μερικές από αυτές, κατα προτεραιότητα τις παλαιότερες. Αυτό το κάνουμε ώστε να σχηματίσουμε ένα καλό θεμέλιο για ένα μεγαλύτερο πλήθος συσκευών που πιθανότατα να υπερ φόρτωναν την βάση μας.

Σε περίπτωση που δεχθούμε πως οι ανώμαλες καταστάσεις ξεπερνούν έναν προκαθορισμένο αριθμό, τότε θεωρείται πως έχει υπάρξει ένα έντονο σφάλμα και θα πρέπει να γίνει άμεση ενημέρωση, οπότε και στέλνουμε ένα SMS ώστε να γίνουν οι άμεσες ενέργειες από το τεχνικό προσωπικό.



Εικόνα 7.4.3
SMS ενημέρωσης σφάλματος

Κεφάλαιο 8

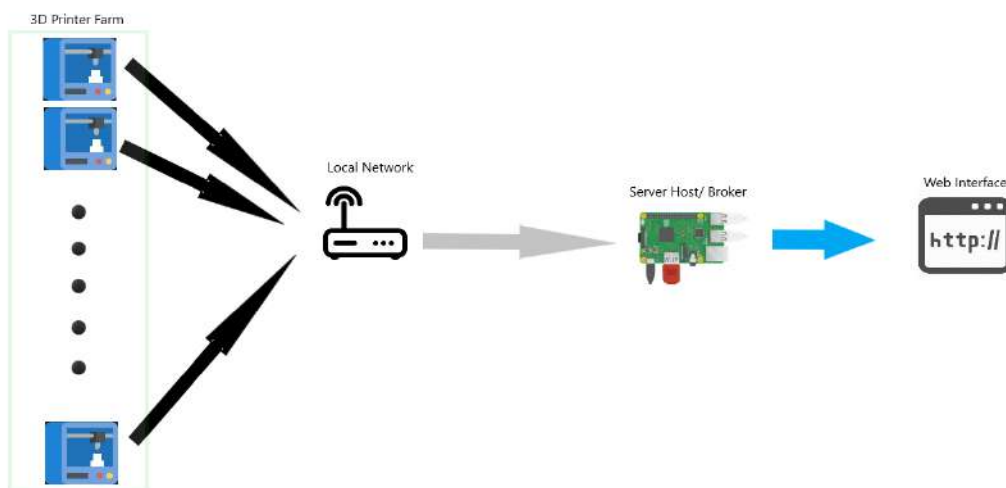
Ολοκλήρωση Συστήματος

Έχοντας ολοκληρώσει όλα σχεδόν τα επιμέρους κομμάτια του συστήματος μας, θα τα συνδυάσουμε. Όπως είδαμε και προηγουμένως τα στοιχεία που θα αποτελέσουν το σύστημα μας είναι:

- I. Συσκευή που αναλαμβάνει την συλλογή δεδομένων από τα αισθητήρια και τα αποστέλλει σειριακά σε μια άλλη υπολογιστική μονάδα.
- II. Μοντέλο μηχανικής μάθησης το οποίο έχει εκπαιδευτεί με τέτοιο τρόπο ώστε να αναγνωρίζει μη-φυσιολογικές καταστάσεις λειτουργίας.
- III. Ένας διακομιστής, ο οποίος συλλέγει όλες τις πληροφορίες, τι επεξεργάζεται και τις προβάλλει σε μια ιστοσελίδα για την ευκολότερη πρόσβαση των πληροφοριών από μια πληθώρα συσκευών.

Πιο συγκεκριμένα, θα φτιάξουμε ένα πρόγραμμα το οποίο θα διαβάζει τα δεδομένα από την συσκευή δειγματοληψίας, λαμβάνοντας τα δεδομένα του επιταχυνσιόμετρου, θα τα επεξεργάζεται με την βοήθεια του μοντέλου μηχανικής μάθησης και θα αποφασίζει αν οι τιμές είναι φυσιολογικές.

Αφού τα συγκεντρώσουμε, με την βοήθεια του πρωτοκόλλου MQTT, θα τα στείλουμε μέσω του δικτύου μας στον διακομιστή, ο οποίος θα τα επεξεργαστεί, θα τα αποθηκεύσει και θα τα προβάλλει στην ιστοσελίδα που σχεδιάσαμε. Πιο συγκεκριμένα η δομή λειτουργίας φαίνεται από την παρακάτω εικόνα.



Εικόνα 8.0.1
Κώδικας συλλογής και αποθήκευσης δεδομένων, σκέλος
δεύτερο

8.1 Κώδικας Συγχώνευσης

Ο κώδικας που θα γράψουμε θα είναι στην γλώσσα Python, αξιοποιώντας την μεγάλη ποικιλία από έτοιμες βιβλιοθήκες. Η ευκολία που μας προσφέρει η συγκεκριμένη γλώσσα προγραμματισμού, είναι ιδανική, μιας και θα έχουμε την ευχέρεια να δοκιμάσουμε το μοντέλο μηχανικής μάθησης και επιπλέον, αφού θα διαβάζουμε τις πληροφορίες που θα αντλούμε από την συσκευή δειγματοληψίας, να τις αποθηκεύουμε σε ένα αρχείο .csv με σκοπό τον οπτικό έλεγχο και επιβεβαίωση λειτουργίας.

```
import serial
import os
import time
import argparse
import json
import threading
import time
import numpy as np
import pandas as pd
import csv
import itertools #for range
import paho.mqtt.publish as publish
import json
import scipy as sp

from os.path import join
from io import BytesIO
from functools import partial
from scipy import stats
from tf.lite_runtime.interpreter import Interpreter
```

Εικόνα 8.1.1
Εισαγωγή βιβλιοθηκών

```
#initialise com port
input_com_port = serial.Serial('com0', 9600, timeout=0.5)
printer_com_port = serial.Serial('com4', 115200, timeout=3)
print("COM Ports OK")
#max samples that are going to be imported to out model
samples_to_read = 80
```

Εικόνα 8.1.2
Δήλωση σειριακών θυρών επικοινωνίας και οριοθέτηση
δειγμάτων

```

samples_read = 0#current sample

normals_counter = 0#Normal operation counter
anomalies_counter = 0#Anomalies Detected
general_counter = 0 #printing session counter

idle_counter = 0#Counter for when a printer is powered On but not printing
operation_counter = 0#Counter for when a printer is printing
operation_status = 0#When 0 the printer is not operating
saved_filament_data_flag = 0 #At the start of the program the fillament wasn't saved to µC EEPROM

Filament_session_flag = 0
starting_filament_value = 1 #When a print has started save the value of the material
ending_filament_val = 0 #The fillament consumed in the current printing session
Filament_consumed_m = 0#Fillament consumed in meters

clean_data = []#Cleaned input data
clean_data_f = []#Data converted to Float
clean_data_p = []#Data converted to Float
sample = []#Sample of readings received from the µC
temperature = 0#Extruder Motor temperature variable

```

Εικόνα 8.1.3
Δήλωση βοηθητικών μεταβλητών και πινάκων
αποθήκευσης δειγμάτων

```

#At first boot of the µC, read the garbage data that can not be used
dummy_serial_read = str(input_com_port.readline().decode('utf-8', errors='ignore'))
dummy_serial_read = str(input_com_port.readline().decode('utf-8', errors='ignore'))
dummy_serial_read = str(input_com_port.readline().decode('utf-8', errors='ignore'))

# Settings for the machine learning model
MODELS_PATH = 'models'#Folder is called "models"
TFLITE_MODEL_FILE = 'stepper_variance' # the name of the model file that was converted to .tflite
MAX_MEASUREMENTS = 80 # Truncate measurements to this number
ANOMALY_THRESHOLD = 0.195#0.000084#A value above the current threshold will be considered an Anomaly
OPERATION_THRESHOLD = 5.3e-06 # Detect if the printer is operational idle is from 5e-07 mostly

# Load model
Interpreter = Interpreter(join(MODELS_PATH, TFLITE_MODEL_FILE) + '.tflite')
Interpreter.allocate_tensors()
input_details = Interpreter.get_input_details()
output_details = Interpreter.get_output_details()

```

Εικόνα .1.4
Καθαρισμός σειριακής θύρας και διαμόρφωση θέσης
φακέλου του μοντέλου μηχανικής μάθησης

```

#//-----DATA SEND SETUP-----//
#JSON format setup
dummy_3d = 0
counter = 0
sensor_data = {'printerID': 0, 'Anomalies Counter':0, 'Normal Counter':0, 'Motor Temp':0,
               | 'Nozzle Temp':0, 'Bed Temp':0, 'T_Fil_Consumed':0, 'S_Fil_Consumed':0, 'Operation':0}

#MQTT configuration
MQTT_SERVER = "169.254.199.17" #Ip of the local server running our webpage
MQTT_PATH = "Production" #test_channel

```

Εικόνα 8.1.5
 Διαμόρφωση δεδομένων σε μορφή JSON και
 παραμετροποίηση διεύθυνσης για την αποστολή των
 δεδομένων

```

#//-----FUNCTION DECLARATION-----//
# Function: extract variances from sample
def extract_features(sample, max_measurements=0):
    #Variance features
    features = []

    # Truncate sample
    if max_measurements == 0:
        max_measurements = sample.shape[0]
    sample = sample[0:max_measurements]

    # Variance of the values
    features.append(np.var(sample, axis=0))

    return np.array(features).flatten()

#Function: Detect if read value is a number
def is_number(string):
    try:
        float(string)
        return True
    except ValueError:
        return False

```

Εικόνα 8.1.6
 Συνάρτηση εξαγωγής χαρακτηριστικών δεδομένων
 επιταχυνσιομετρου και συνάρτηση επαλήθευσης τύπου
 στοιχείου για την εισαγωγή σε πίνακα

```

#//-----MAIN LOOP-----//
#// In this loop, we read through the serial port the data provided by the MPU-6050
#// that is connected to our µController
while 1:

    while samples_read <= samples_to_read:

        #//Read from the COM port
        data = str(input_com_port.readline().decode('utf-8'))

        clean_data.append(data.split())#Split the parameters that are separated by a space
        samples_read = samples_read + 1
        # print(clean_data_f)
        try:
            clean_data_f = [[float(j) for j in e] for e in clean_data]#Convert string values to float
        except:
            # print
            data = str(input_com_port.readline().decode('utf-8'))
            clean_data.append(data.split()) # Split the parameters that are separated by a space
            clean_data_f = [[float(j) for j in e] for e in clean_data] # Convert string values to float
            # print(clean_data_f)

        #Import cleaned data to the sample that is going to be analysed
        try:#Beta testing t oignore data rubbish
            for row in clean_data_f:
                sample.append([float(row[0]),
                               float(row[1]),
                               float(row[2])])

                temperature = row[3]
                Filament_consumed_m = row[4]
        except:
            print("----")

```

Εικόνα 8.1.7
Κύρια επανάληψη, σκέλος πρώτο

```

input_com_port.flushInput()
# Calculate variance for each axis
feature_set = extract_features(np.array(sample),
                               max_measurements=MAX_MEASUREMENTS)

# Make prediction from model
in_tensor = np.float32(feature_set.reshape(1, feature_set.shape[0]))

Interpreter.set_tensor(input_details[0]['index'], in_tensor)
Interpreter.invoke()
output_data = Interpreter.get_tensor(output_details[0]['index'])
pred = output_data[0]

# Calculate MSE (MEAN SQUARED ERROR)
mse = np.mean(np.power(feature_set - pred, 2))
print("MSE:", mse)

```

Εικόνα 8.1.8
Κύρια επανάληψη, σκέλος δεύτερο

```

#//-----NORMAL/ ANOMALY DETECTION-----//
if mse > ANOMALY_THRESHOLD:
    print("ANOMALY DETECTED!")
    anomalies_counter += 1
else:
    print("NORMAL")
    normals_counter += 1
# //-----OPERATION DETECTION-----//
if mse > OPERATION_THRESHOLD: #If the printer is operating
    operation_counter += 1
    idle_counter = 0 # reset the idle counter since the printer is operating
    print("Printer =====> !! Operating !!", operation_counter)
    print(idle_counter)

    if operation_counter > 2:
        print("The printer is operational and Currently printing")
        sensor_data['Operation'] = 1
    if operation_counter == 3:
        operation_status = 1 # The printer is operating
        idle_counter = 0
        print("idle value is zero now")
        Filament_session_flag = 1# A new printing Session began
        starting_filament_value = Filament_consumed_m # save the current filament reading as a starting value
        ending_filament_val = 0 # Since a printing event started clear the session counter

```

Εικόνα 9.1.8
Κύρια επανάληψη, σκέλος τρίτο

```

else:
    idle_counter += 1
    print(idle_counter, operation_counter)
    # For handling the idle_counter
    # and saving after every printing session as not to overstrain the eeprom
    if idle_counter == 40:
        print("Saving Filament Parameters to Device's EEPROM")
        input_com_port.write(b's\n')
        input_com_port.flushInput() # This is mandatory to be able to send data again

        dummy_serial_read = str(input_com_port.readline().decode('utf-8', errors='ignore'))
        dummy_serial_read = str(input_com_port.readline().decode('utf-8', errors='ignore'))
        dummy_serial_read = str(input_com_port.readline().decode('utf-8', errors='ignore'))

        saved_filament_data_flag = 1 # Now we saved our filament data
        operation_status = 0 # The printer is currently not operating

        sensor_data['Operation'] = 0

        normals_counter = 0 # There is no printer operation
        anomalies_counter = 0
        general_counter = 0

        operation_counter = 0
        print("operating value is zero now")

    if Filament_session_flag == 1:
        ending_filament_val = Filament_consumed_m - starting_filament_value
        # IMPORTANT
        #Must not forget to save the value of the session in the database after multiple idle states
        general_counter += 1

```

Εικόνα 8.1.9
Κύρια επανάληψη, σκέλος τέταρτο

```

if Filament_session_flag == 1 :
    ending_filament_val = Filament_consumed_m - starting_filament_value
    # IMPORTANT
    #Must not forget to save the value of the session in the database after multiple idle states
    general_counter += 1

    samples_read = 0#Reset the samples to read
    clean_data.clear()#Clean the received string list
    clean_data_f.clear()#Clean the clean data list
    sample.clear()#Clean the sample list

# //-----PRINTING RESULTS-----//
print("General Session counter : ", general_counter)
print("Normal Condition counter : ", normals_counter)
print("Anomalous Condition counter : ", anomalies_counter)
print("Current Motor Temperature : ", temperature)
print("Filament Consumed in Total: ", Filament_consumed_m)
print("Filament Consumed in Session: ", ending_filament_val)

# //-----IMPORT RESULTS TO JSON-----//
sensor_data['Anomalies Counter'] = anomalies_counter
sensor_data['Normal Counter'] = normals_counter
sensor_data['Motor Temp'] = temperature
sensor_data['S_Fil_Consumed'] = ending_filament_val
sensor_data['T_Fil_Consumed'] = Filament_consumed_m

```

Εικόνα 8.1.10
Κύρια επανάληψη, σκέλος πέμπτο

```

# //-----SEND RESULTS TO MQTT SERVER-----//
dummy_serial_read = str(printer_com_port.readline().decode('utf-8', errors='ignore'))
dummy_serial_read = str(printer_com_port.readline().decode('utf-8', errors='ignore'))
dummy_serial_read = str(printer_com_port.readline().decode('utf-8', errors='ignore'))
# print("connected1")

if printer_com_port.is_open:
    # print("connected2")

    printer_com_port.write(b'M105\n')

    data = str(printer_com_port.readall().decode('utf-8'))
    # print(data)
    clean_data_p.append(data.split())
    # print("connected3")
    for row in clean_data_p:
        sample.append([(row[0]),
                       (row[1]),
                       (row[2])])

        nozzle_temperature = row[3]
        bed_temperature = row[5]

    MQTT_SERVER = "169.254.199.17" # 192.168.1.73"
    MQTT_PATH = "test_channel"
    publish.single(MQTT_PATH, json.dumps(sensor_data), hostname=MQTT_SERVER)
    print("sending data to server...\r\n")

    if dummy_3d > 1:
        sensor_data['Nozzle Temp'] = nozzle_temperature[2:]

        # print(Filament_consumed_m[3:])
        sensor_data['Bed Temp'] = bed_temperature[3:]
    dummy_3d += 1

```

Εικόνα 8.1.11
Κύρια επανάληψη, σκέλος έκτο

8.2 Επεξήγηση λειτουργίας

Ο κώδικας που συνθέσαμε στο παραπάνω υποκεφάλαιο, αφού εισάγει όλες τις απαραίτητες βιβλιοθήκες, ορίζει δύο σειριακές θύρες, η μία της συσκευής δειγματοληψίας και η άλλη του εκτυπωτή. Αφού ορίσουμε τις βοηθητικές μας μεταβλητές, ώστε αν τις αξιοποιήσουμε σε σενάρια Truth ή False της Boolean λογικής θα ορίσουμε και τους μονοδιάστατους πίνακες στους οποίους θα αποθηκεύσουμε τα δεδομένα που θα συλλέξουμε.

Σειριακά θα πρέπει να διαβάσουμε μερικές φορές τον input buffer, μιας και η σειριακή επικοινωνία στην αρχή της χειραψίας μπορεί να φέρει “σκουπίδια”, ή αλλιώς άσχετη πληροφορία με αυτή που περιμένουμε, τα οποία δεν θα μπορέσουμε να περάσουμε από το μοντέλο μηχανικής μάθησης οπότε και θα πρέπει να τα ξεφορτωθούμε. Όσον αφορά την αποστολή των δεδομένων στον διακομιστή, δημιουργούμε έναν πίνακα με μορφολογία json, ορίζουμε την IP διεύθυνση του διακομιστή και την διαδρομή που θα ακολουθήσει.

Για μεγαλύτερη ευκολία, δημιουργήθηκαν δύο συναρτήσεις. Η πρώτη, λαμβάνει τα δεδομένα και εξάγει τα χαρακτηριστικά που είναι απαραίτητα για την αναγνώριση ανωμαλιών λειτουργίας. Η δεύτερη, ελέγχει αν η πληροφορία που δεχτήκαμε από την σειριακή θύρα αποτελεί αριθμό. Αφού διαβάσουμε την σειριακή πληροφορία, θα πρέπει να εισάγουμε τις τιμές σε έναν πίνακα. Για να το κάνουμε αυτό, θα πρέπει να χωρίσουμε τις τιμές, με βάση το κενό που υπάρχει μεταξύ τους, αφού οι τιμές χωριστούν, θα πρέπει να τις μετατρέψουμε από string σε float, ώστε να τις εισάγουμε στο μοντέλο μηχανικής μάθησης.

Αν το μέσο τετραγωνικό σφάλμα είναι μεγαλύτερο από την οριακή τιμή, τότε σημαίνει πως το σύστημα έχει εντοπίσει μια ανώμαλη κατάσταση. Έχοντας λάβει και τα υπόλοιπα δεδομένα από το σύστημα δειγματοληψίας, θα τα τοποθετήσουμε στις αντίστοιχες json τιμές και θα τις στείλουμε στον διακομιστή που φιλοξενεί την ιστοσελίδα, προβάλλοντας έτσι τα δεδομένα παρακολούθησης. Σε περίπτωση που εμφανιστεί ένα έντονο σφάλμα, όπως η θερμοκρασία να υπερβεί του 230 °C, τότε, στέλνουμε στον εκτυπωτή άμεση διακοπή λειτουργίας, ώστε να αποφύγουμε το ενδεχόμενο πυρκαγιάς, σε περίπτωση υπερθέρμανσης του ακροφυσίου.

Όπως είδαμε και στο προηγούμενο κεφάλαιο, ο διακομιστής, αφού λάβει τα δεδομένα ελέγχει πόσες ανώμαλες καταστάσεις έχουν παρουσιαστεί και αν αυτές έχουν υπερβεί ένα προκαθορισμένο όριο, τότε στέλνουμε ένα ενημερωτικό SMS στο υπεύθυνο τεχνικό προσωπικό.

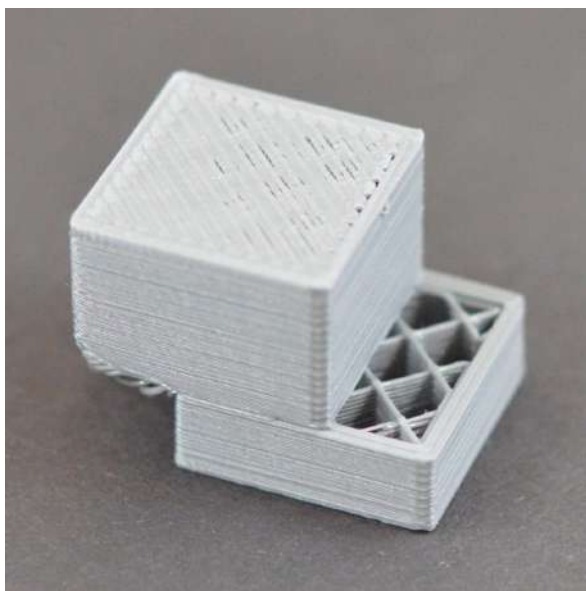
8.3 Μελλοντικές Βελτιώσεις

Έχοντας σχηματίσει την βασική δομή του συστήματος μας, δεν μπορούμε να μην εμπνευστούμε όσον αφορά τα επόμενα βήματα εξέλιξης του. Το σύστημα που δημιουργήσαμε αποτελεί μια σημαντική υποδομή πολλών εφαρμογών και αναβαθμίσεων. Αρχικά, η ιστοσελίδα, η οποία αποτελεί μια διεπαφή μεταξύ ανθρώπου και μηχανής, δεν προσφέρει στοιχεία ελέγχου στον χρήστη και η εφαρμογή της είναι καθαρά πληροφοριακή. Αυτή η αδυναμία περιορίζει αρκετά, την άμεση ενέργεια ειδικά αν πρόκειται για μια εγκατάσταση με πολλά μηχανήματα. Ιδανικά, θα πρέπει να υπάρξει έλεγχος :

- I. Όσον αφορά την παύση και την συνέχιση εκτύπωσης, σε περίπτωση που τελειώνει το υλικό.
- II. Ρύθμιση θερμοκρασίας.
- III. Ρύθμιση ταχύτητας εκτύπωσης.

Αξιοποιώντας το σύστημα για αρκετό καιρό, θα μπορούσαμε να συλλέξουμε στατιστικά κατανάλωσης υλικού, με αυτόν τον τρόπο μπορούμε να υπολογίσουμε έναν μέσο ρυθμό κατανάλωσης και έτσι να υπάρχουν και οι σχετικές ειδοποιήσεις αγοράς υλικού, ώστε να μην υπάρξει ποτε νεκρός χρόνος παραγωγής.

Παρόλο που ένα μεγάλο μέρος των σφαλμάτων οφείλεται στην εσφαλμένη τροφοδοσία υλικού, συγκεκριμένα για την FDM τρισδιάστατη εκτύπωση, η χρήση αισθητηρίων στους κινητήρες δεν μπορούν να μας δώσουν την πλήρη εικόνα του σφάλματος. Ένα από τα σφάλματα που δεν μπορούμε να προβλέψουμε είναι το Layer shifting, το οποίο αν και σπάνιο σφάλμα, μπορεί να μην μπορέσουμε να το εντοπίσουμε σαν ανωμαλία με την μέθοδο μας.



Εικόνα 8.3.1
Layer Shifting σφάλμα εκτύπωσης

Μια τεχνική που θα μπορούσε να λύσει το παραπάνω πρόβλημα, είναι η χρήση μιας κάμερας και η δημιουργία ενός μοντέλου μηχανικής μάθησης το οποίο θα μπορεί να αναγνωρίζει αυτού του είδους τα σφάλματα και έτσι, θα είναι ένας επιπλέον τρόπος εντοπισμού ανώμαλων καταστάσεων. Η παραπάνω τεχνική, αποτελεί μια πολύπλοκη διαδικασία, διότι απαιτείται ένας μεγάλος αριθμός δεδομένων, πιο συγκεκριμένα φωτογραφίες με το παραπάνω σφάλμα. Το σύστημα μας αποτελεί μια πολύ σημαντική βάση για την ανάπτυξη της παραπάνω ιδέας, μιας και μέσω αυτού θα μπορούσαμε να συνδυάσουμε όλα τα δεδομένα μας και έτσι να γίνεται ακριβέστερη η πρόβλεψη κάποιου σφάλματος.

Μια βελτίωση που θα μπορούσε να μειώσει τους νεκρούς χρόνους παραγωγής, θα ήταν η αναγνώριση ολοκλήρωσης ενός κύκλου παραγωγής ενός εκτυπωτή, ώστε να υπάρχει και η σχετική ειδοποίηση για την διαθεσιμότητα του μηχανήματος.

Τέλος, έχοντας δοκιμάσει το μοντέλο μηχανικής μάθησης που χρησιμοποιήσαμε για την ολοκλήρωση του παραπάνω συστήματος, μπορέσαμε να αποκτήσουμε μια εικόνα για την λειτουργία του. Σε μια μελλοντική βελτίωση θα ήταν επιθυμητό να γραφεί ο κώδικας σε γλώσσα C++. μιας και παρατηρήθηκε χαμηλή ταχύτητα επεξεργασίας μέσω της γλώσσας Python. Έχοντας το σύστημα μας, σαν μέτρο σύγκρισης, θα μπορούσαμε να δοκιμάσουμε και άλλους τύπους αλγορίθμου, που πειραματικά μπορεί και να δούμε πως το παραγόμενο μοντέλο είναι αποδοτικότερο από το ήδη υπάρχον.

Βιβλιογραφικές Αναφορές

- <https://www.sculpteo.com/blog/2017/03/01/whos-behind-the-three-main-3d-printing-technologies/>
- <https://reprap.org/wiki/G-code>
- <https://3dinsider.com/3d-printing-history/>
- <https://redshift.autodesk.com/history-of-3d-printing/>
- <https://trailhead.salesforce.com/en/content/learn/modules/learn-about-the-fourth-industrial-revolution/meet-the-three-industrial-revolutions>
- <https://reprap.org/wiki/Darwin>
- <https://www.tctmagazine.com/additive-manufacturing-3d-printing-news/back-story-sls-3d-printing-dr-joe-beaman/>
- <https://www.cmac.com.au/blog/difference-between-3d-printing-rapid-prototyping>
- <https://www.asme.org/topics-resources/content/3d-printing-houses>
- <https://www.starrapid.com/blog/5-reasons-to-use-cnc-for-rapid-prototyping/>
- <https://all3dp.com/2/cnc-water-jet-all-you-need-to-know/>
- <https://pick3dprinter.com/cnc-rapid-prototyping/>
- The Evolution of CNC Machines (2018). Retrieved October 15, 2018, from Engineering Technology Group
- Fitzpatrick, Michael (2019), "Machining and CNC Technology".
- Smid, Peter (2008), *CNC Programming Handbook* (3rd ed.), New York: Industrial Press, ISBN 9780831133474, LCCN 2007045901.
- <https://all3dp.com/1/3d-printer-extruder-nozzle-guide/>
- <https://all3dp.com/2/3d-printer-bed-how-to-choose-the-right-build-plate/>
- <https://www.pbcliner.com/Blog/2020/February/Lead-Screw-or-Belt-Drives>
- <https://www.drdflo.com/pages/Guides/How-to-Build-a-3D-Printer/Carriages.html>
- <https://all3dp.com/2/direct-vs-bowden-extruder-technology-shootout/>
- <https://howtomechatronics.com/tutorials/g-code-explained-list-of-most-important-g-code-commands/>
- <https://hobbyhoarder.net/hotend-explained/>
- <https://all3dp.com/2/linear-rail-3d-printer-really-better-or-just-a-hype/>
- <https://primacreator.com/blogs/news/meet-the-ruby-nozzle>
- <https://academy.autodesk.com/make-3d-printer-human-factors-form-ergonomics-course-series>
- https://reprap.org/wiki/RAMPS_1.4
- <https://all3dp.com/2/what-is-a-3d-slicer-simply-explained/>
- <https://www.omega.com/en-us/resources/stepper-motors>
- <https://marlinfw.org/docs/basics/introduction.html>
- <https://instesre.org/ArduinoElectricalFundamentals.html>
- https://electronoobs.com/eng_arduino_tut31_sch2.php
- <https://pimylifeup.com/raspberry-pi-mysql/>
- <https://pimylifeup.com/raspberry-pi-serial/>

- <https://randomnerdtutorials.com/what-is-mqtt-and-how-it-works/>
- <https://mschoeffler.com/2017/10/05/tutorial-how-to-use-the-gy-521-module-mpu-6050-breakout-board-with-the-arduino-uno/>
- <https://towardsdatascience.com/anomaly-detection-for-dummies-15f148e559c1>
- <https://www.bmc.com/blogs/machine-learning-anomaly-detection/>
- <https://www.digikey.com/en/maker/projects/what-is-edge-ai-machine-learning-iot/4f655838138941138aaad62c170827af>
- <https://mohamedelhafi.medium.com/use-the-mqtt-protocol-to-communicate-data-between-2-raspberry-pi-3d432dea9313>
- <https://blogs.oracle.com/ai-and-datascience/post/introduction-to-anomaly-detection>
- <https://towardsdatascience.com/auto-encoder-what-is-it-and-what-is-it-used-for-part-1-3e5c6f017726>
- <https://www.anodot.com/blog/what-is-anomaly-detection/>
- <https://codeburst.io/2-important-statistics-terms-you-need-to-know-in-data-science-skewness-and-kurtosis-388fef94eeaa>
- <https://www.purplemath.com/modules/meanmode.htm>
- <https://www.kaggle.com/residentmario/keras-optimizers>
- <https://www.digikey.com/en/maker/projects/edge-ai-anomaly-detection-part-1-data-collection/7bb112f76ef644edaedc5e08dba5faae>
- <https://towardsdatascience.com/machine-learning-for-anomaly-detection-and-condition-monitoring-d4614e7de770>
- <https://towardsdatascience.com/how-to-use-machine-learning-for-anomaly-detection-and-condition-monitoring-6742f82900d7>
- <https://towardsdatascience.com/everything-you-need-to-know-about-autoencoders-in-tensorflow-b6a63e8255f0>
- <https://www.pyimagesearch.com/2020/02/17/autoencoders-with-keras-tensorflow-and-deep-learning/>
- <https://developers.google.com/chart>