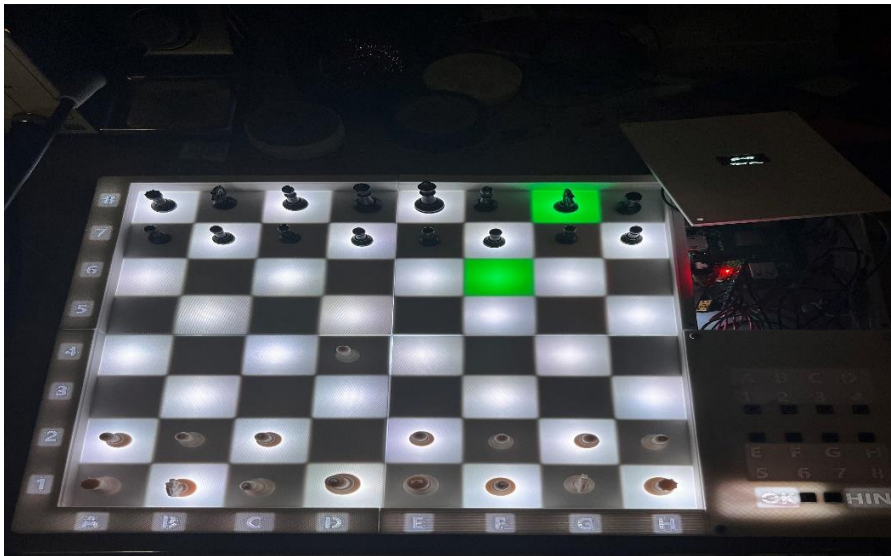


ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
«Έξυπνη Σκακιέρα»



Του φοιτητή
Σαββίδη Νικόλαου
Αρ. Μητρώου: 516115

Επιβλέπων
Ονοματεπώνυμο Άγγελος
Γιακουμής
Βαθμίδα Λέκτορας

Ημερομηνία

Τίτλος Π.Ε. ΕΞΥΠΝΗ ΣΚΑΚΙΕΡΑ
Κωδικός Π.Ε. 22165
Όνοματεπώνυμο φοιτητή ΣΑΒΒΙΔΗΣ ΝΙΚΟΛΑΟΣ
Όνοματεπώνυμο εισηγητή ΑΓΓΕΛΟΣ ΓΙΑΚΟΥΜΗΣ
Ημερομηνία ανάληψης Π.Ε. 18/03/2022
Ημερομηνία περάτωσης Π.Ε. ...

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Σαββίδη Νικόλαου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Αφιέρωση»

Αφιερώνω αυτή την πτυχιακή εργασία στην οικογένειά μου για την υποστήριξή της όλα τα χρόνια της ακαδημαϊκής μου σταδιοδρομίας, καθώς και για την οικονομική ενίσχυσή τους για την επίτευξη των στόχων μου.

Πρόλογος

Η πτυχιακή εργασία με τίτλο έξυπνη σκακιέρα επιλέχθηκε με σκοπό τη δημιουργία μιας πρωτότυπης συσκευής, με πολλαπλές προοπτικές εξέλιξης, η οποία θα μπορεί να προσφέρει στον άνθρωπο την ικανοποίηση αλλά και ψυχαγωγία να αλληλοεπιδράσει με συγκεκριμένη μορφή τεχνητής νοημοσύνης. Ο απώτερος σκοπός αυτής της πτυχιακής εργασίας είναι πρώτα να διασκεδάσει ο παίκτης αλλά παράλληλα να του προσφέρει γνώσεις για το δημοφιλές παιχνίδι σκάκι. Συνοπτικά, ο άνθρωπος θα μπορεί να παίξει σκάκι σε μία κανονική σκακιέρα χρησιμοποιώντας κανονικά κομμάτια για σκάκι με αντίπαλο την σκακιστική μηχανή-νοημοσύνη Stockfish, η οποία αφού ο παίκτης επιλέξει την κίνηση του, θα γνωστοποιεί σε αυτόν χρησιμοποιώντας χρωματικές ενδείξεις επάνω στη σκακιέρα, την κίνηση που θέλει να πραγματοποιήσει ο παίκτης εκ μέρους της μηχανής. Τέλος, επέλεξα το συγκεκριμένο θέμα καθώς ήταν αυτή η επιθυμία μου από το 2019 ξεκινώντας πάλι με την σειρά μου να είμαι ενεργός παίκτης και ανακαλύπτοντας ενδιαφέρον projects επάνω στην τεχνολογική του ανάπτυξη.

Περίληψη

Η εκπαιδευτική έξυπνη σκακιέρα είναι ένα ενδιαφέρον μέσο εκμάθησης και αυτό-βελτίωσης στο δημοφιλές σε όλους μας παιχνίδι σκάκι. Αναπτύσσει μορφές αυτοματισμού, προγραμματισμό και ηλεκτρονικό ενδιαφέρον όσο αφορά την κατασκευή της. Μας δίνει την δυνατότητα προσομοίωσης πραγματικών παιχνιδιών με δυνατότητα βοήθειας κατά την διάρκεια της παρτίδας. Θα υπάρχει έλεγχος του παιχνιδιού κάθε στιγμή, όπως και φωτεινές ενδείξεις βοήθειας όταν αυτές ζητούνται από τον παίκτη. Σίγουρα μπορεί να υποστηρίξει μελλοντικές βελτιώσεις που μπορούν να την κάνουν μια ακόμα πιο αυτοματοποιημένη κατασκευή, όμως στην εν λόγω εργασία σκοπός είναι να συμβάλει σημαντικά και ο άνθρωπος. Να μπορεί να συμμετέχει ο ίδιος ώστε να έχει την κλασική εμπειρία του σκάκι με μια δόση έξτρα τεχνολογίας.

«Smart Chessboard»

«Nikolaos Savvidis»

Abstract

The educational smart chessboard is an interesting means of learning and self-improvement in the popular game of chess. It develops forms of automation, programming and electronic interest in its construction. It allows us to simulate real games with hint function during the game. There will be a referee function running at all times in the background, as well as hint indication lights when requested by the player. It can certainly support future improvements that can make it an even more automated build, but in this work the goal is for humans to contribute significantly as well. To be able to participate themselves and to have the classic chess game experience with a dose of extra technology.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την οικογένειά μου για την υποστήριξή της όλα τα χρόνια της ακαδημαϊκής μου πορείας και για την οικονομική της υποστήριξη ώστε να είμαι σε θέση να επιτύχω τους στόχους μου. Τους καθηγητές μου που μου κίνησαν το ενδιαφέρον να μάθω και να εξελιχθώ ώστε να γίνω καλύτερος άνθρωπος αλλά και επαγγελματίας στην μετέπειτα πορεία μου. Τέλος, θα ήθελα να ευχαριστήσω την σύντροφό μου για την υποστήριξή της κατά την συγγραφή της συγκεκριμένης εργασίας.

Περιεχόμενα

Πρόλογος.....	v
Περίληψη.....	vi
Abstract	vii
Ευχαριστίες	viii
Περιεχόμενα	ix
Κατάλογος Εικόνων	xi
Κατάλογος Πινάκων.....	xii
Κατάλογος Σχημάτων	xii
Κατάλογος Μοντέλων.....	xii
Συνομογραφίες.....	xiii
Κεφάλαιο 1ο: Δημιουργία εξαρτημάτων σκακιέρας	1
1.1 Εισαγωγή.....	1
1.2 3D Εκτυπωτής.....	1
1.3 Ρυθμίσεις.....	3
1.4 Λειτουργικό εκτυπωτή	3
1.5 Υλικό εκτύπωσης	4
1.6 Εκτυπώσεις μοντέλων στο πρόγραμμα προσομοίωσης	5
1.6.1 Συναρμολόγηση σκακιέρας.....	9
1.7 Επίλογος.....	16
Κεφάλαιο 2ο: Υλικά και τα χαρακτηριστικά τους	17
2.1 Εισαγωγή.....	17
2.2 Μικροϋπολογιστής.....	17
2.3 Υπολογιστής.....	20
2.3.1 Τροφοδοτικό.....	22
2.4 Φωτισμός και κουμπιά	23
2.5 Λογικό επίπεδο.....	23
2.6 Οθόνη OLED.....	25
2.7 Επίλογος.....	26
Κεφάλαιο 3ο: Συναρμολόγηση ηλεκτρονικού υλικού.....	27
3.1 Εισαγωγή.....	27
3.2 Φωτισμός.....	27
3.3 Κουμπιά και συνδεσμολογία	28

3.4	Συνδέσεις μικροϋπολογιστή	28
3.5	Προσαρμογή λογικής κατάστασης.....	30
3.6	Σύνδεση οθόνης.....	32
3.7	Επίλογος.....	33
Κεφάλαιο 4ο: Εγκατάσταση λογισμικού και προγραμματισμός Arduino		34
4.1	Εισαγωγή.....	34
4.2	Γλώσσα και προγραμματισμός Arduino	34
4.3	Γλώσσα raspberry pi	37
4.3.1	Stockfish.....	39
4.3.2	Maxchess.....	39
4.3.3	Autostart	40
4.4	Σειριακές επικοινωνίες.....	40
4.5	Επίλογος.....	41
Κεφάλαιο 5ο: Συμπεράσματα και μελλοντικές βελτιώσεις		43
5.1	Συμπεράσματα.....	43
5.2	Μελλοντικές βελτιώσεις.....	48
Βιβλιογραφία.....		51

Κατάλογος Εικόνων

Εικόνα 1.1: Κομμάτια 3D εκτυπωτή.....	1
Εικόνα 1.2: 3D εκτυπωτής.....	2
Εικόνα 1.3: Ρολό PLA τρισδιάστατης εκτύπωσης.....	4
Εικόνα 1.4: Κομμάτι βάσης αρχική στρώση.....	5
Εικόνα 1.5: Εκτυπωμένο κομμάτι οθόνης.....	6
Εικόνα 1.6: Κομμάτι βάσης μεγάλο.....	8
Εικόνα 1.7: Συναρμολόγηση της βάσης.....	9
Εικόνα 1.8: Εκτύπωση πρώτης στρώσης.....	11
Εικόνα 1.9: Βάση κομμάτι μεγάλο.....	12
Εικόνα 1.10: Βάση κομμάτι μεγάλο.....	13
Εικόνα 1.11: Κομμάτι μεγάλο βάσης.....	14
Εικόνα 1.12: Πίνακας ελέγχου.....	15
Εικόνα 2.1: Arduino Nano.....	17
Εικόνα 2.2: Arduino Nano.....	18
Εικόνα 2.3: Συνδέσεις πίνακα ελέγχου και Arduino Nano.....	19
Εικόνα 2.4: Υπολογιστής Raspberry Pi.....	20
Εικόνα 2.5: Ανεμιστήρας εργασίας.....	21
Εικόνα 2.6: Raspberry Pi συνδέσεις.....	22
Εικόνα 2.7: Τροφοδοτικό Raspberry pi.....	23
Εικόνα 2.8: Μηχανικό button.....	23
Εικόνα 2.9: Logic level shifter.....	24
Εικόνα 2.10: Κομμάτι κώδικα σχετικό με την επικοινωνία των συσκευών.....	25
Εικόνα 2.11: Οθόνη OLED.....	25
Εικόνα 3.1: Ταινία LED.....	27
Εικόνα 3.2: Μετατροπέας λογικού επιπέδου.....	30
Εικόνα 3.3: Block διάγραμμα των συνδέσεων των ηλεκτρονικών στοιχείων της κατασκευής.....	33
Εικόνα 4.1: Κώδικας υπεύθυνος για την απεικόνιση των τετραγώνων της σκακιέρας και εφέ εκκίνησης συσκευής.....	35
Εικόνα 4.2: Δήλωση του pin 3 ως διακοπή HINT.....	35
Εικόνα 4.3: Hint interrupt.....	37
Εικόνα 4.4: Συνάρτηση για αποστολή text στην οθόνη OLED.....	38
Εικόνα 4.5: Περιβάλλον stockfish στην πλατφόρμα LICHESS.....	39
Εικόνα 4.6: GPIOs.....	41
Εικόνα 5.1: Εκκίνηση της σκακιέρας.....	44
Εικόνα 5.2: Επιλογή τύπου παιχνιδιού.....	44
Εικόνα 5.3: Επιλογή δυσκολίας παιχνιδιού.....	45
Εικόνα 5.4: Επιλογή χρόνου κίνησης υπολογιστή.....	45
Εικόνα 5.5: Έναρξη παιχνιδιού.....	45
Εικόνα 5.6: Κίνηση D2-D4.....	46
Εικόνα 5.7: Απάντηση υπολογιστή G8-F3.....	46
Εικόνα 5.8: Επιλογή Hint με εμφάνιση της προτεινόμενης κίνησης με μπλε χρώμα.....	47
Εικόνα 5.9: Μη έγκυρη κίνηση από τον παίκτη.....	47

Κατάλογος Πινάκων

Πίνακας 3.1: Interrupt pins που μπορούν να χρησιμοποιηθούν ως διακοπές κατά τα διάφορα μοντέλα Arduino.....	29
-------------------------------------------------------------------------------------------------------------	----

Κατάλογος Σχημάτων

Σχήμα 3.1: Λογικά επίπεδα	31
Σχήμα 3.2: Εναλλακτική απεικόνιση σύνδεσης οθόνης με το raspberry pi	32

Κατάλογος Μοντέλων

Μοντέλο 1.1: Αριστερό κομμάτι βάσης.....	5
Μοντέλο 1.2: Κομμάτι οθόνης OLED	6
Μοντέλο 1.3: Led control panel	7
Μοντέλο 1.4: Αριστερή πάνω γωνία	7
Μοντέλο 1.5: Δεξί κομμάτι βάσης	7
Μοντέλο 1.6: Αριστερή κάτω γωνία	10
Μοντέλο 1.7: Αριστερό πάνω κομμάτι βάσης	10
Μοντέλο 1.8: E-H κομμάτι.....	11
Μοντέλο 1.9: Κάλυμμα επιφάνειας παιχνιδιού.....	11
Μοντέλο 1.10: Δεξί πάνω κομμάτι βάσης.....	12
Μοντέλο 1.11: Πίνακας ελέγχου μπροστινό μέρος.....	15
Μοντέλο 1.12: Επιφάνεια παιχνιδιού	16

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΠΙΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
3D	3 Dimensions
OLED	Organic Light-Emitting Diodes
PLA	Polylactic acid
FDM	Fused Deposition Modeling
SCARA	Selective Compliance Assembly Robot Arm
SLA	Stereolithography
SLS	Selective Laser Sintering
CNC	Computer Numerical Control
STL	Standard Triangle Language/Standard Tessellation Language
LEDs	Light-emitting diodes
GPIOs	general purpose inputs and outputs
OE	Output Enable

Κεφάλαιο 1ο: Δημιουργία εξαρτημάτων σκακιέρας

1.1 Εισαγωγή

Στο κεφάλαιο αυτό θα μιλήσουμε για τα αρχικά βήματα που ακολουθήθηκαν για τη δημιουργία του φυσικού υλικού της σκακιέρας. Θα αναφερθούμε στον τρισδιάστατο εκτυπωτή που χρησιμοποιήθηκε και στις απαραίτητες ρυθμίσεις που έγιναν σε αυτόν για το σωστό καλιμπράρισμα [1] του, καθώς και για το λογισμικό στο οποίο χρειάστηκε να γίνουν κάποιες αλλαγές ώστε να μπορούν να ξεκινήσουν οι εκτυπώσεις. Επίσης, θα μιλήσουμε για τον τύπο υλικού που χρησιμοποιήθηκε και για το πρόγραμμα ρύθμισης των 3D μοντέλων της σκακιέρας και των παραμέτρων τους.

1.2 3D Εκτυπωτής

Κάνοντας μια ιστορική αναδρομή μαθαίνουμε πως στις αρχές του 1980 ο Charles Hull εφεύρε τον πρώτο τρισδιάστατο εκτυπωτή. Αυτή την ανακάλυψη την περιέγραψε ως stereolithography (STL) ή αλλιώς ως μια εκτύπωση στρώσεων ενός υλικού οι οποίες στοιβάζονται η μια πάνω στην άλλη με τον σκοπό να δημιουργήσουν ένα τρισδιάστατο μοντέλο. Σε αυτό των τύπο εκτύπωσης μπορεί να περιλαμβάνονται πληροφορίες όπως το χρώμα του αντικειμένου, την υφή του αντικειμένου, το πάχος των στρωμάτων που θα δημιουργήσουν το αντικείμενο κ.τ.λ. Για να παρουσιαστεί στο κοινό αυτή η καινούρια τεχνολογία εκτυπώθηκαν οι σκελετοί από συμβατικά γυαλιά ώστε να επιδείξουν τις ικανότητες αυτής την τεχνολογίας να αντεπεξέρχεται απέναντι σε πολύπλοκα σχήματα, γωνίες, καμπύλες και γενικότερα σε αρκετές λεπτομέρειες [2].



Εικόνα 1.1: Κομμάτια 3D εκτυπωτή

Τα πρώτα βήματα προς την ολοκλήρωση της Π.Ε. έγιναν με την δημιουργία των φυσικών κομματιών της σκακιέρας. Τα συγκεκριμένα κομμάτια ήταν σε πλήθος δεκαπέντε.

Τέσσερα κομμάτια χωρισμένα με τετράγωνα στο εσωτερικό τους για την βάση της σκακιέρας, τέσσερα κομμάτια διαπερατά από το φως για τον χώρο παιχνιδιού, ένα κομμάτι για την οθόνη τύπου organic light-emitting diodes (OLED) [3], δύο κομμάτια για τον πίνακα ελέγχου της σκακιέρας και τέσσερα κομμάτια για την περιφερειακή κάλυψη στα οποία αναγράφονται οι αριθμοί και τα νούμερα των συντεταγμένων.

Το εργαλείο που χρησιμοποιήθηκε για την εκτύπωση των δεκαπέντε κομματιών, ήταν ο εκτυπωτής μοντέλου Ender 3 Pro της εταιρίας Creality. Ο συγκεκριμένος εκτυπωτής μας δίνει την δυνατότητα να εκτυπώσουμε σε διαστάσεις έως 235mm x 235mm x 250mm, αφού όμως κάνουμε τις απαραίτητες παραμετροποιήσεις στο λειτουργικό του σύστημα Marlin Firmware [4], ώστε να μεγαλώσουμε την περιοχή εκτύπωσης και να την διευρύνουμε στο μέγιστο.



Εικόνα 1.2: 3D εκτυπωτής

Θα υπάρξουν αναφορές για την παραπάνω διαδικασία στα επόμενα κεφάλαια. Η ανάλυση του συγκεκριμένου εκτυπωτή σε σύγκριση πάντα και με το κόστος αγοράς του είναι αξιότιμη.

1.3 Ρυθμίσεις

Για την ορθή λειτουργία του 3D εκτυπωτή ακολουθήθηκαν τα απαραίτητα βήματα για την ρύθμιση των αξόνων τριών διαστάσεων (x,y,z) [4].

Στον άξονα του x βρίσκεται η κεφαλή του εκτυπωτή και για να την ρυθμίσουμε σφίγγουμε την ζώνη του κινητήρα, που είναι υπεύθυνος για την κίνηση στον άξονα του x, ώστε να μπορεί με άνεση η κεφαλή να κινείται κατά μήκος του.

Στον άξονα z εγκαταστάθηκε αισθητήρας ανίχνευσης ύψους ώστε να σταματάει την κατακόρυφη κίνηση της κεφαλής στο καθορισμένο σημείο μηδέν που ορίζουμε εμείς, ώστε το υλικό που εξάγεται από την άκρη της κεφαλής να απλώνεται και να κολλάει ομοιόμορφα επάνω στην επιφάνεια εκτύπωσης.

Στον άξονα y βρίσκεται η επιφάνεια εκτύπωσης η οποία χρειάζεται να ζεσταθεί στην κατάλληλη θερμοκρασία που ορίσαμε στους 60 έως 80 Celsius, για να μπορέσει το υλικό μας τύπου Polylactic acid (PLA) να κολλήσει επάνω κατά την διάρκεια της τρισδιάστατης εκτύπωσης.

Η επιφάνεια εκτύπωσης λόγω του υλικού της (γυαλί) θα παρουσίαζε διαστολή όταν θα έφτανε την επιθυμητή θερμοκρασία εκτύπωσης, έτσι αφού ζεστάθηκε, ρυθμίστηκε η απόστασή της και στις τέσσερις γωνίες της επιφάνειας εκτύπωσης καθώς και στο κέντρο της να είναι στο 1mm. Με αυτό τον τρόπο μπορούμε να εξασφαλίσουμε τον σωστό σχεδιασμό των τρισδιάστατων μοντέλων της έξυπνης σκακιέρας.

1.4 Λειτουργικό εκτυπωτή

Σε αυτό το κεφάλαιο θα μιλήσουμε για το λειτουργικό σύστημα του τρισδιάστατου εκτυπωτή, την εφαρμογή του σε άλλα μηχανήματα σχεδίασης ή/και εκτύπωσης και για την συμβατότητα του σε αρκετές επιλογές hardware. Το Marlin [4] είναι ένα λειτουργικό ανοιχτού τύπου κώδικα που έχει σχεδιαστεί κυρίως για τρισδιάστατους εκτυπωτές τεχνολογίας Fused Deposition Modeling (FDM) [5].

Το λειτουργικό Marlin εκτελεί τις εντολές κωδικοποιημένες καθώς και τα σύνολα οδηγιών με μορφοποίηση G-code [6], ως λογισμικό ρομπότ στον πίνακα ελέγχου του 3D εκτυπωτή. Όπου στον οποίο διαχειρίζονται όλες οι δραστηριότητες, του τρισδιάστατου εκτυπωτή, σε πραγματικό χρόνο. Συμπεριλαμβανομένης της αποστολής συντεταγμένων κίνησης στους βηματικούς κινητήρες των αξόνων x,y,z μέσω των βηματικών προγραμμάτων οδήγησης, ελέγχου θερμοκρασιών στοιχείων, αισθητήρων, φώτων, παρακολούθηση απόστασης από την επιφάνεια εκτύπωσης, οθόνες LCD και κουμπιά.

Το Marlin [4] υποστηρίζει μια μεγάλη ποικιλία από πλακέτες και πολλές πλατφόρμες ρομπότ για τρισδιάστατους εκτυπωτές, εκ των οποίων είναι οι καρτεσιανοί εκτυπωτές, δηλαδή, οι εκτυπωτές στους άξονες x και y, εκτυπωτές τύπου Delta [7] και εκτυπωτές τύπου Selective Compliance Assembly Robot Arm (SCARA) [8], καθώς και ορισμένους λιγότερο συμβατικών τεχνολογιών όπως εκτυπωτές τύπου Hangprinter [9].

Εκτός από την τρισδιάστατη εκτύπωση, η Marlin [4] χρησιμοποιείται επίσης για πολλούς άλλους τύπους μηχανών, όπως εκτυπωτές τύπου Stereolithography (SLA) [10], τύπου Selective Laser Sintering (SLS)

[11], Computer Numerical Control (CNC) [12] mills και μηχανές κοπής με χρήση δέσμης laser (laser cutters).

1.5 Υλικό εκτύπωσης

Το υλικό που χρησιμοποιήθηκε για την εκτύπωση των τρισδιάστατων μοντέλων της έξυπνης σκακιέρας ήταν τύπου PLA. Για την παραγωγή του PLA χρησιμοποιείται 65% λιγότερη ενέργεια από ότι η παραγωγή των συμβατικών πλαστικών. Επίσης παράγει 68% λιγότερα αέρια θερμοκηπίου και δεν τοξίνες. Αυτά είναι μερικά από τα χαρακτηριστικά του συγκεκριμένου υλικού που το καθιστούν ένα αρκετά οικολογικό υλικό χωρίς πολλές επιπτώσεις στο περιβάλλον. Το PLA σαν υλικό είναι αρκετά ευαίσθητο στην θερμοκρασία. Αρχίζει να γίνεται σημαντικά εύκαμπτο σε θερμοκρασία που κυμαίνεται περίπου στους 126 °F ή 52 °C [13].



Εικόνα 1.3: Ρολό PLA τρισδιάστατης εκτύπωσης

Στα θετικά:

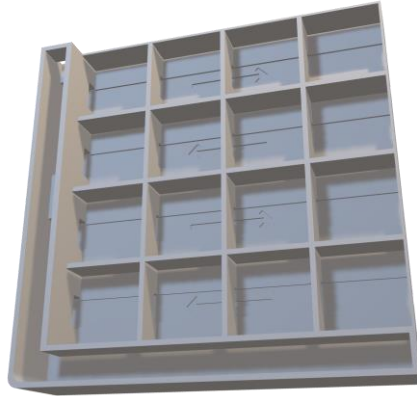
- Φίλικό προς το περιβάλλον
- Εύκολο στη χρήση για τρισδιάστατες εκτυπώσεις
- Ασφαλές για να χρησιμοποιηθεί στην δημιουργία σκευών φαγητού ή ιατρικού εξοπλισμού
- Υπάρχει μεγάλη ποικιλία σε χρώματα και χαρακτηριστικά για την επίτευξη διαφορετικών εφέ αλλά και ιδιοτήτων

Στα αρνητικά:

- Χαμηλή θερμική αντοχή
- Συγκριτικά με άλλα υλικά δεν είναι τόσο δυνατό
- Η επεξεργασία του μπορεί να επιφέρει δυσκολίες

1.6 Εκτυπώσεις μοντέλων στο πρόγραμμα προσομοίωσης

Για την προσομοίωση αλλά και παραμετροποίηση των τρισδιάστατων εκτυπώσεων χρησιμοποιήθηκε το πρόγραμμα Ultimaker-Cura. Με το συγκεκριμένο εργαλείο μπορούμε να παραμετροποιήσουμε τα τρισδιάστατα μοντέλα μας σε ένα πολύ μεγάλο βαθμό, όπως μπορούμε να ρυθμίσουμε και ποικίλα χαρακτηριστικά του μηχανήματος όπου θα κάνουμε τις τρισδιάστατες εκτυπώσεις. Τα 3D μοντέλα αποκτήθηκαν από την ιστοσελίδα Thingiverse, το οποίο εξειδικεύεται στα 3D μοντέλα, και προσαρμόστηκαν με τέτοιο τρόπο στο Slicer ώστε να πληρούν τις προδιαγραφές που έθεσα για την εν λόγω πτυχιακή εργασία.



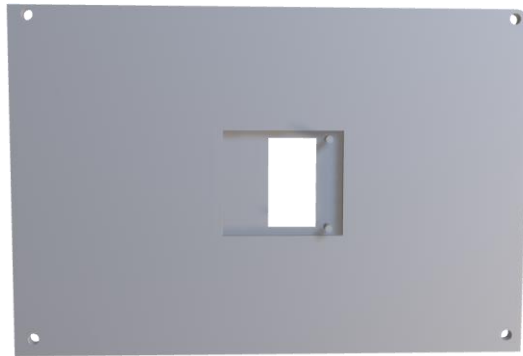
Μοντέλο 1.1: Αριστερό κομμάτι βάσης



Εικόνα 1.4: Κομμάτι βάσης αρχική στρώση

Κεφάλαιο 1

Το πρόγραμμα χρησιμοποιεί αρχεία τύπου Standard Triangle Language ή Standard Tessellation Language (STL). Επί της ουσίας ένα αρχείο STL είναι ένα τριγωνικό μοντέλο επιφάνειας [14].



Μοντέλο 1.2: Κομμάτι οθόνης OLED



Εικόνα 1.5: Εκτυπωμένο κομμάτι οθόνης

Αφού εισάγουμε το αρχείο μας μέσα στο πρόγραμμα επεξεργασίας των τρισδιάστατων μοντέλων και κάνουμε τις απαραίτητες παραμετροποιήσεις τότε το αρχείο μας, επεξεργάζεται με την διαδικασία του τεμαχισμού όπου αυτό έχει σαν αποτέλεσμα να δημιουργήσει την διαδρομή που πρέπει να ακολουθηθεί από την θερμαινόμενη κεφαλή του τρισδιάστατου εκτυπωτή μας για την επίτευξη της σωστής εκτύπωσης του τρισδιάστατου μοντέλου.



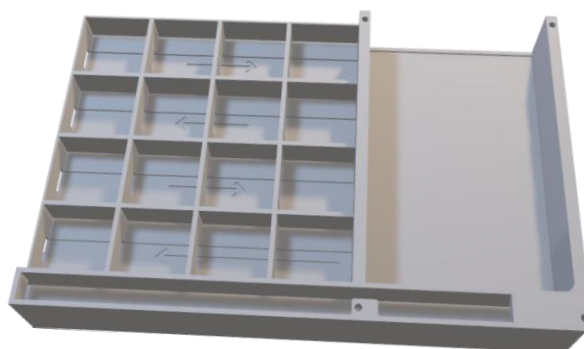
Μοντέλο 1.3: Led control panel

Ουσιαστικά οι πληροφορίες που παράγει το πρόγραμμα επεξεργασίας με την διαδικασία του τεμαχισμού αποθηκεύονται ως ένα αρχείο CNC που μπορεί να αξιοποιήσει ένας τρισδιάστατος εκτυπωτής.

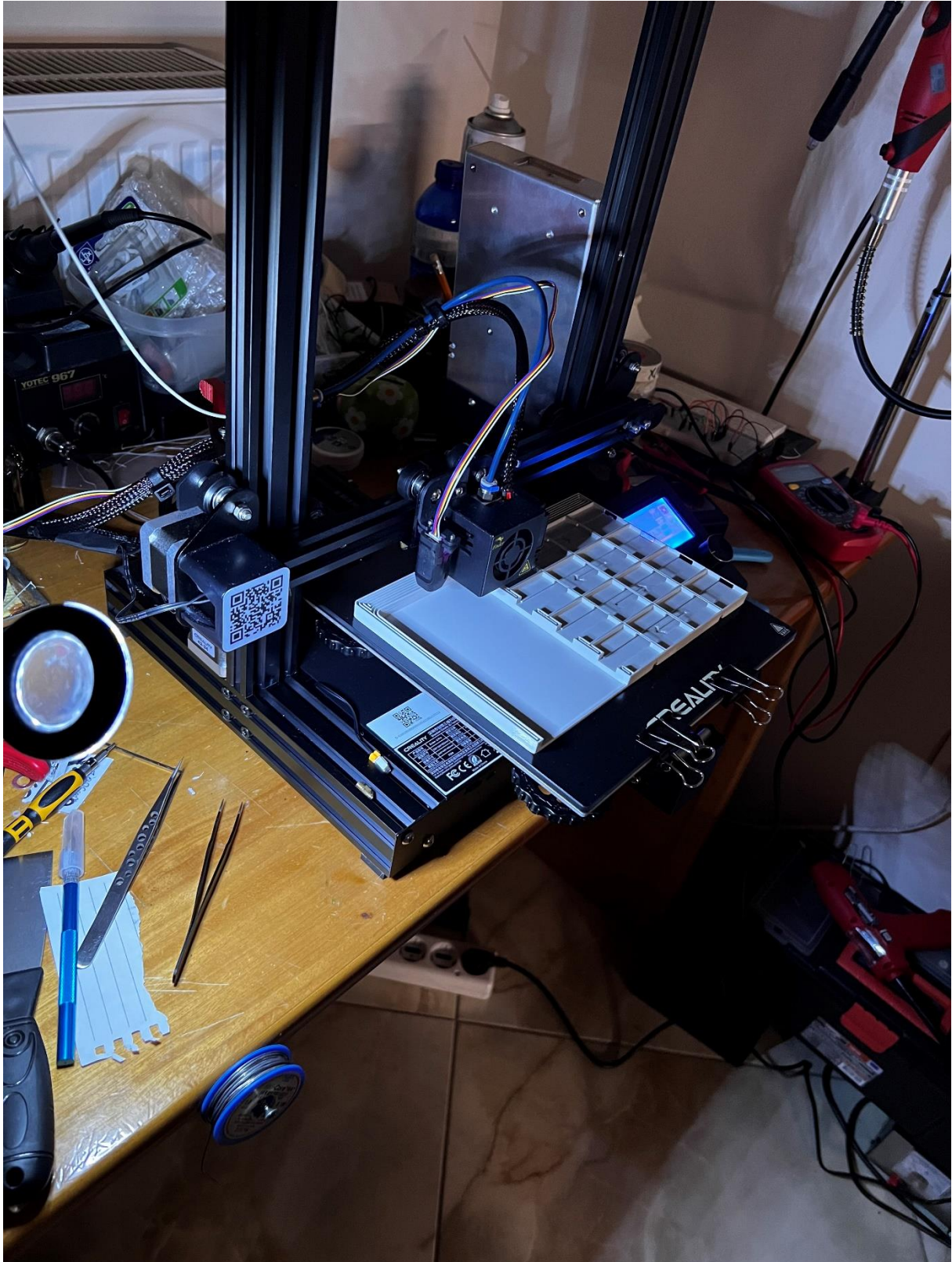


Μοντέλο 1.4: Αριστερή πάνω γωνία

G-code [6] είναι ο όρος που χρησιμοποιούμε για να χαρακτηρίσουμε ένα αρχείο κειμένου το οποίο περιέχει εντολές για να κατευθύνει το μηχάνημα της τρισδιάστατης εκτύπωσης.



Μοντέλο 1.5: Δεξί κομμάτι βάσης



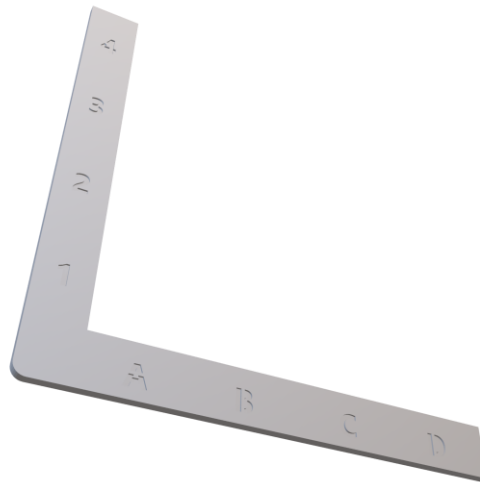
Εικόνα 1.6: Κομμάτι βάσης μεγάλο

1.6.1 Συναρμολόγηση σκακιέρας

Για την διαδικασία της συναρμολόγησης της βάσης της έξυπνης σκακιέρας χρησιμοποιήθηκε κυανοακρυλική κόλλα για να σταθεροποιήσει τα κομμάτια μεταξύ τους. Τα τέσσερα κομμάτια που αποτελούν την βάση της έξυπνης σκακιέρας κολλήθηκαν σε ζευγάρια ξεκινώντας από τα δύο μικρότερα και έπειτα τα άλλα δύο μεγαλύτερα. Η διαδικασία ήταν γρήγορη και αποτελεσματική.



Εικόνα 1.7: Συναρμολόγηση της βάσης

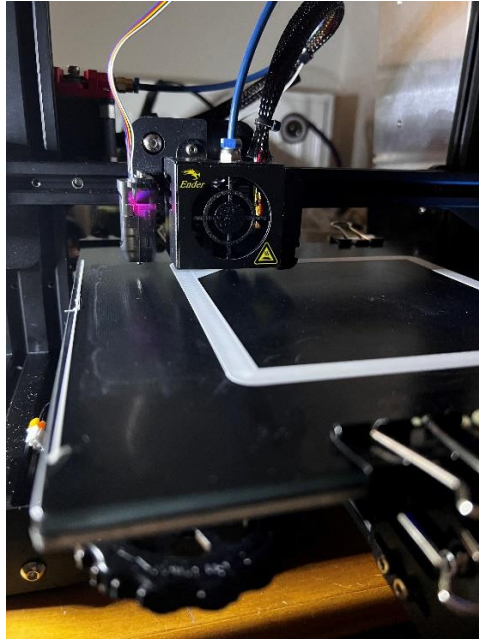


Μοντέλο 1.6: Αριστερή κάτω γωνία

Αφού πλέον η βάση ήταν σταθερή ξεκίνησε η διαδικασία εγκατάστασης των Light-emitting diodes (LEDs) [15].



Μοντέλο 1.7: Αριστερό πάνω κομμάτι βάσης



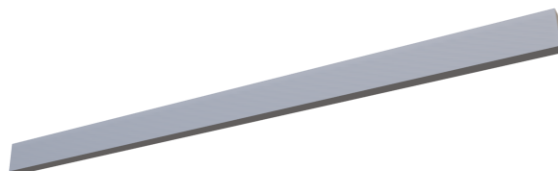
Εικόνα 1.8: Εκτύπωση πρώτης στρώσης

Από την ταινία η οποία ενσωματώνει προγραμματιζόμενα LEDs χρησιμοποιήθηκαν οκτώ ταινίες των οκτώ LED για τον κύριο φωτισμό της βάσης της έξυπνης σκακιέρας. Οι ταινίες συνδεσμολογήθηκαν σε σειρά ξεκινώντας από την κάτω δεξιά πλευρά της βάσης και καταλήγοντας στην επάνω δεξιά. Στην αρχή της ταινίας έγινε η κόλληση του καλωδίου εισόδου δεδομένων για τον φωτισμό της βάσης.



Μοντέλο 1.8: E-H κομμάτι

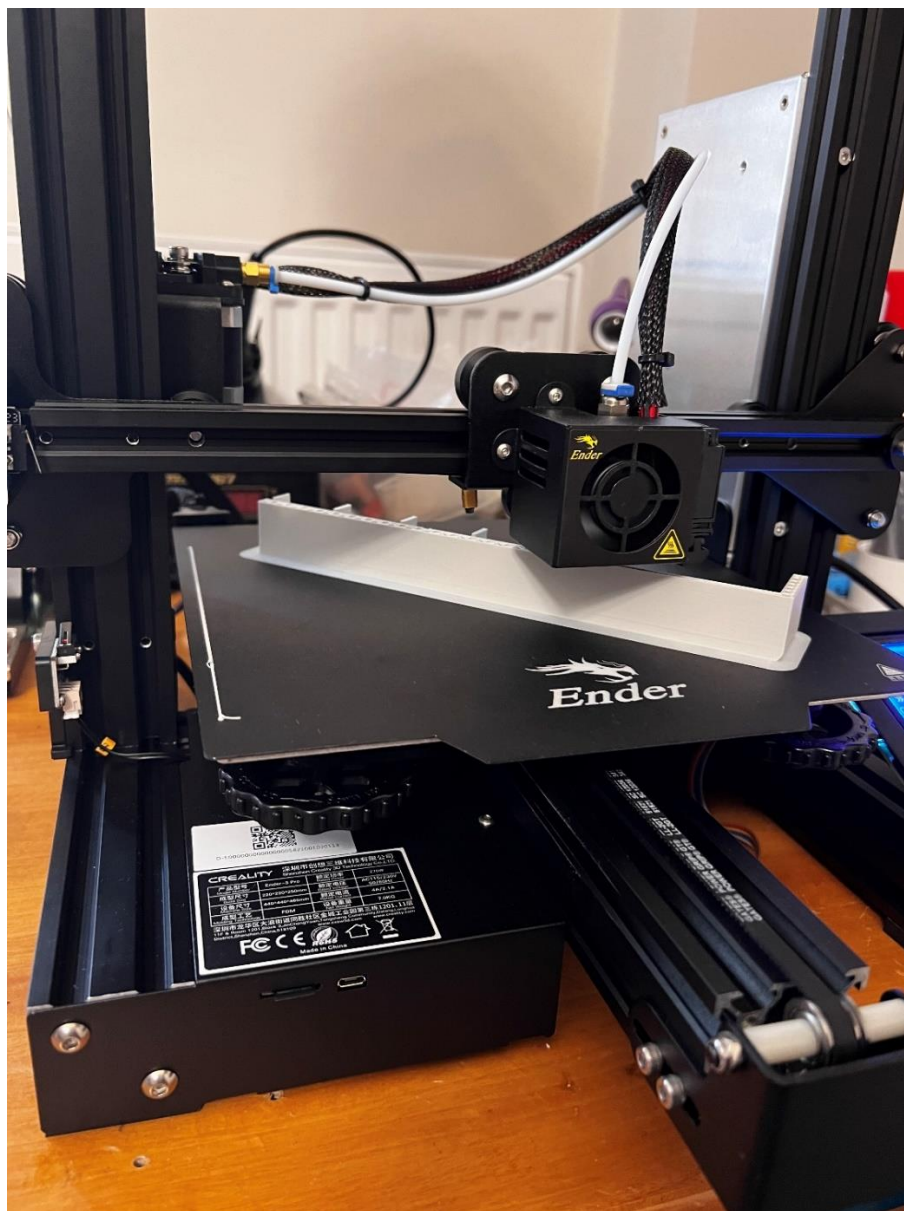
Έπειτα ξεκίνησε η διαδικασία κατασκευής του πίνακα ελέγχου της έξυπνης σκακιέρας. Με τη χρήση της κόλλας που χρησιμοποιήθηκε για την βάση κολλήθηκαν δέκα διακόπτες τύπου button. Αφού σταθεροποιήθηκαν έγινε η κόλληση για να τοποθετηθεί η κοινή γείωσή τους χρησιμοποιώντας πολύ μικρού μήκους και πάχους πολύκλωνο καλώδιο.



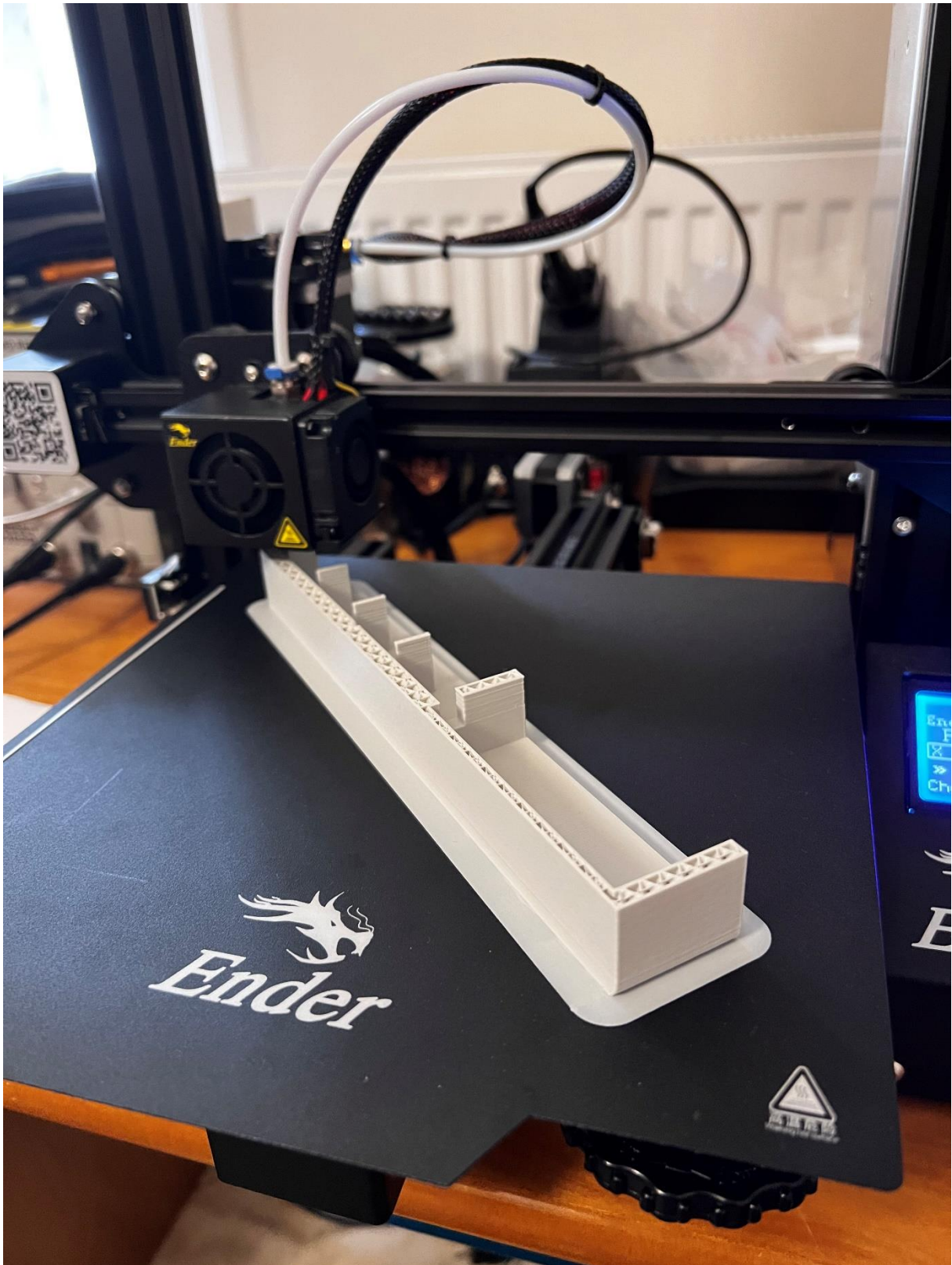
Μοντέλο 1.9: Κάλυμμα επιφάνειας παιχνιδιού



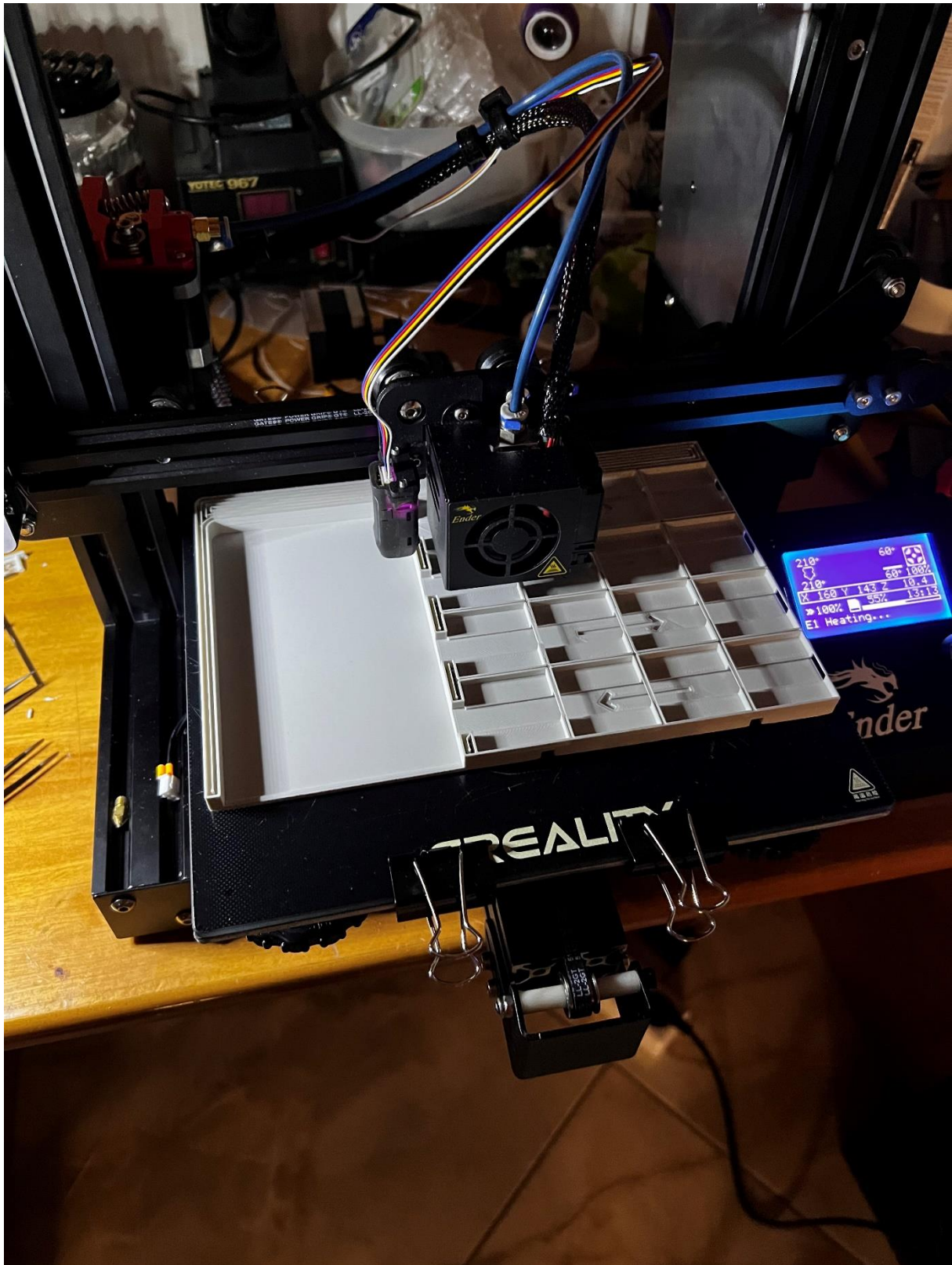
Μοντέλο 1.10: Δεξί πάνω κομμάτι βάσης



Εικόνα 1.9: Βάση κομμάτι μεγάλο

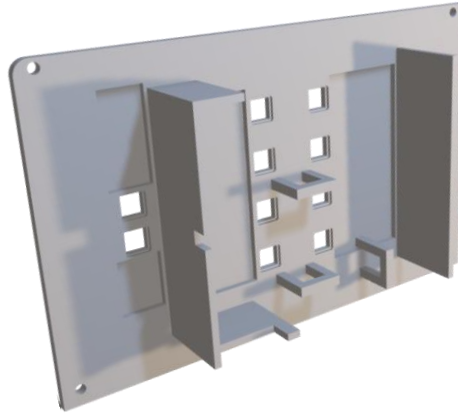


Εικόνα 1.10: Βάση κομμάτι μεγάλο

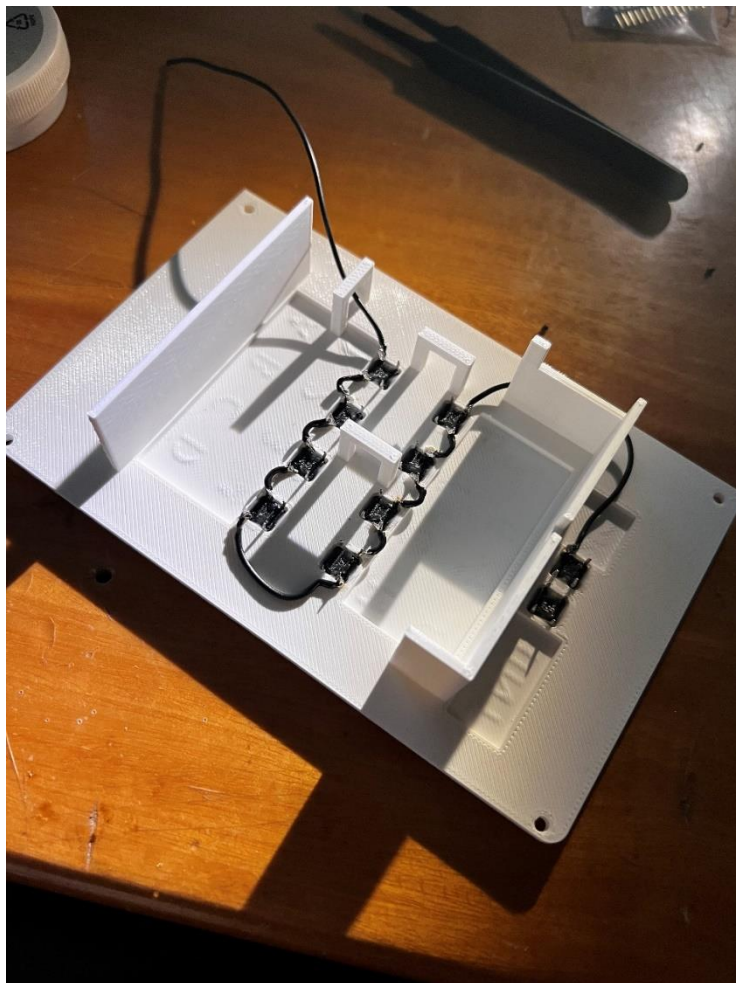


Εικόνα 1.11: Κομμάτι μεγάλο βάσης

Για να ολοκληρωθεί η εγκατάσταση του πίνακα ελέγχου ακολούθησε η κόλληση δέκα καλωδίων σε κάθε ένα από τα κουμπιά με σκοπό να συνδεθούν μετά στο Arduino Nano. Θα γίνει επεξήγηση της διαδικασίας κόλλησης των LEDs και buttons σε παρακάτω κεφάλαια.



Μοντέλο 1.11: Πίνακας ελέγχου μπροστινό μέρος



Εικόνα 1.12: Πίνακας ελέγχου



Μοντέλο 1.12: Επιφάνεια παιχνιδιού

1.7 Επίλογος

Συνοψίζοντας, σε αυτό το κεφάλαιο αναφερθήκαμε στις γενικές παραμετροποιήσεις αλλά και ενέργειες που ακολουθήθηκαν με σκοπό την υλοποίηση των φυσικών κομματιών από τα οποία αποτελείτε η κατασκευή της έξυπνης σκακιέρας. Είδαμε το μηχανήμα των τρισδιάστατων εκτυπώσεων, το λειτουργικό του σύστημα καθώς και το πρόγραμμα με το οποίο γίνεται η διαδικασία τεμαχισμού των τρισδιάστατων μοντέλων. Τελικά καταλήξαμε στην κόλληση της βάσης της σκακιέρας με σκοπό την ενσωμάτωση των ηλεκτρονικών στοιχείων της Π.Ε. όπως θα δούμε στα κεφάλαια που θα ακολουθήσουν.

Κεφάλαιο 2ο: Υλικά και τα χαρακτηριστικά τους

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα μιλήσουμε για το ηλεκτρονικό υλικό που χρησιμοποιήθηκε για την Π.Ε. έξυπνη σκακίερα αλλά και τους λόγους που επιλέχθηκαν τα συγκεκριμένα υλικά. Θα μιλήσουμε για το ενεργειακό κόστος της συσκευής, την ταχύτητα με την οποία λειτουργεί αλλά και την λογική κατάσταση κατά την οποία λαμβάνονται οι εντολές από τον υπολογιστή Raspberry Pi 3A+.

Θα γίνει αναφορά στην οθόνη τύπου Organic Light-emitting diode OLED όπως και στο φωτισμό της συσκευής.

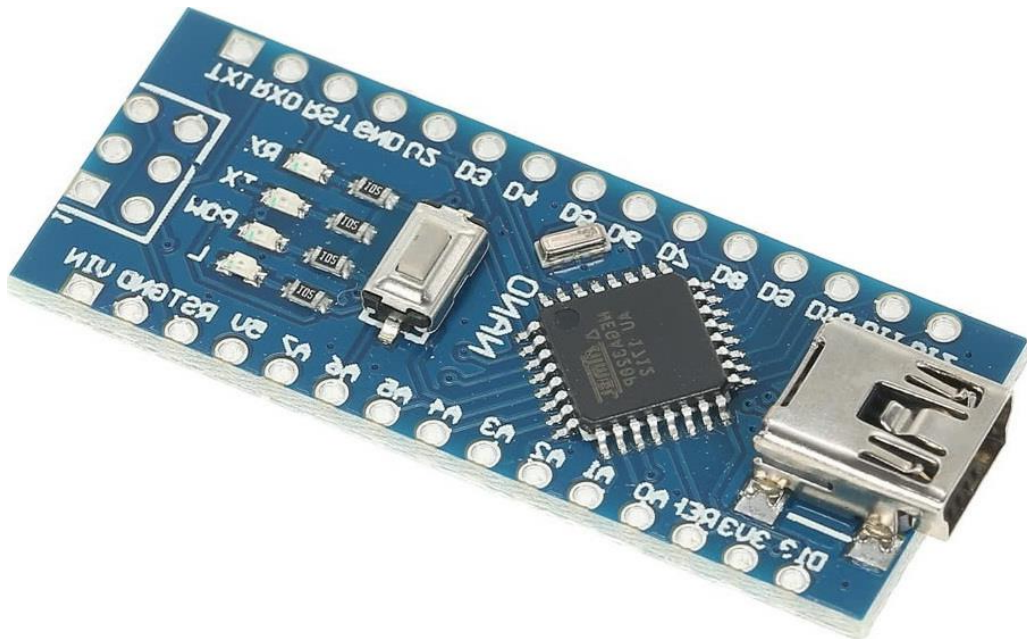
2.2 Μικροϋπολογιστής

Τα τετράγωνα της έξυπνης σκακίερας δεν υπάρχουν ζωγραφισμένα στο ταμπλό παιχνιδιού, βρίσκονται στο εσωτερικό της κατασκευής. Τα φυσικά τετράγωνα εμφανίζονται με την βοήθεια των LEDs όταν αυτά είναι αναμμένα και σβηστά διαδοχικά κατά μήκος της σκακίερας.

Ο φωτισμός της κατασκευής ελέγχεται από ένα μικροϋπολογιστή τύπου Arduino Nano που χρησιμοποιεί έναν επεξεργαστή ATmega328. Ο συγκεκριμένος μικροϋπολογιστής λόγω του μικρού του μεγέθους αλλά και της πολύ χαμηλής ενεργειακής κατανάλωσής του εξυπηρετεί ώστε η συσκευή να μην έχει μεγάλες ενεργειακές απαιτήσεις [16].

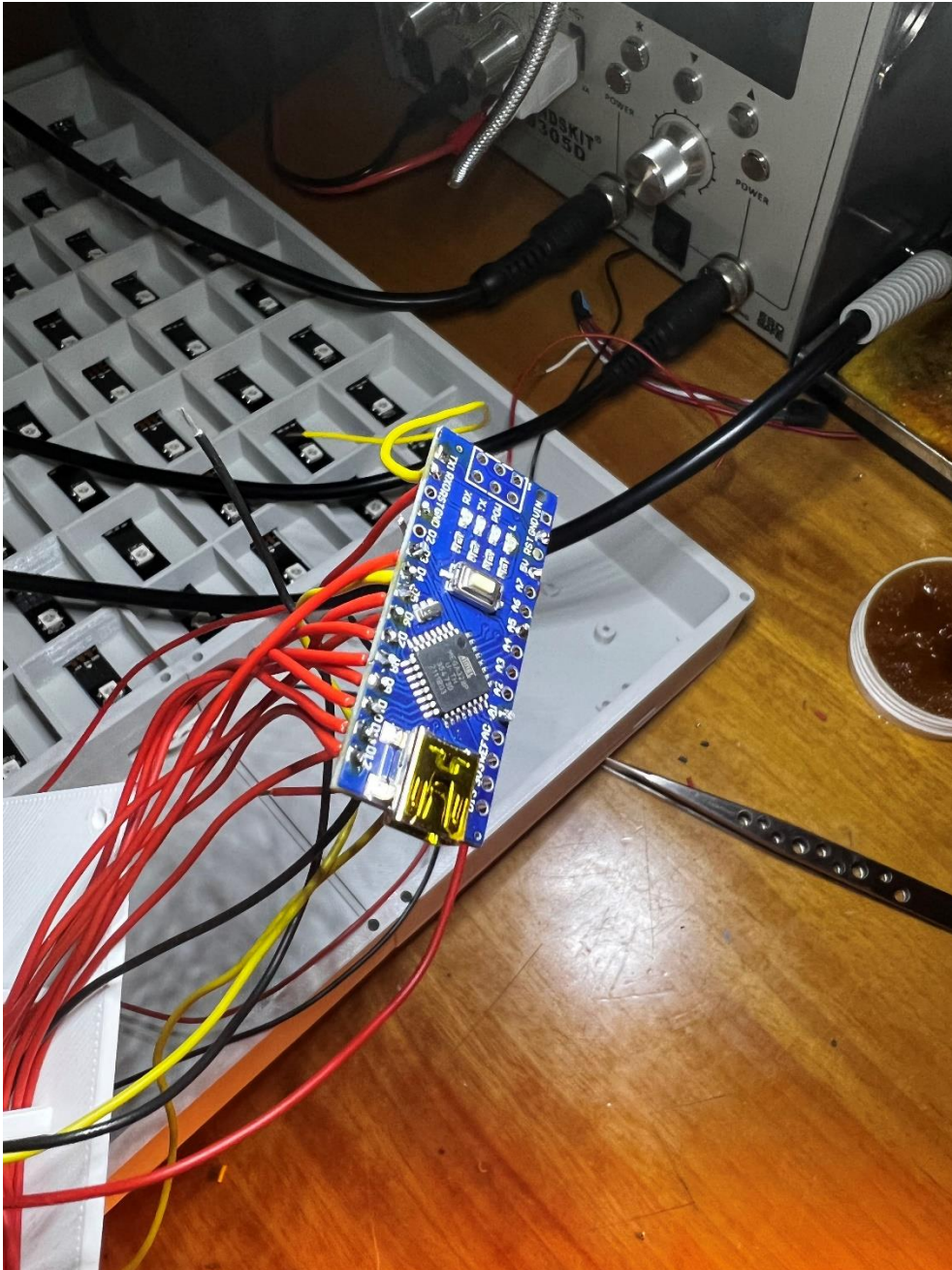
Τα buttons ελέγχονται και αυτά από τον μικροϋπολογιστή ώστε η σκακίερα να γνωρίζει κάθε φορά ποιο κουμπί έχει πατηθεί. Συγκεκριμένα, το κάθε κουμπί έχει ένα γράμμα και έναν αριθμό.

Στο σκάκι οι παίκτες χρησιμοποιούν συντεταγμένες ώστε να επικοινωνούν για τις κινήσεις τους, καθώς και για να καταγράφουν ολόκληρα παιχνίδια με σκοπό να τα μελετούν.



Εικόνα 2.1: Arduino Nano

Η ίδια φιλοσοφία έχει εφαρμοστεί και εδώ με το πρώτο πάτημα να δηλώνει το γράμμα ή αλλιώς την στήλη στην οποία βρίσκεται το κομμάτι και το δεύτερο πάτημα τον αριθμό ώστε να δηλωθεί η σειρά, δηλαδή η ακριβής αρχική θέση του κομματιού. Έπειτα, ακριβώς με τον ίδιο τρόπο εισάγουμε την επιθυμητή κίνηση ή αλλιώς τελική θέση που θέλουμε να κινηθεί αυτό το κομμάτι ώστε να φωτιστούν τα αντίστοιχα LEDs πράσινο χρώμα, στο ταμπλό του παιχνιδιού.

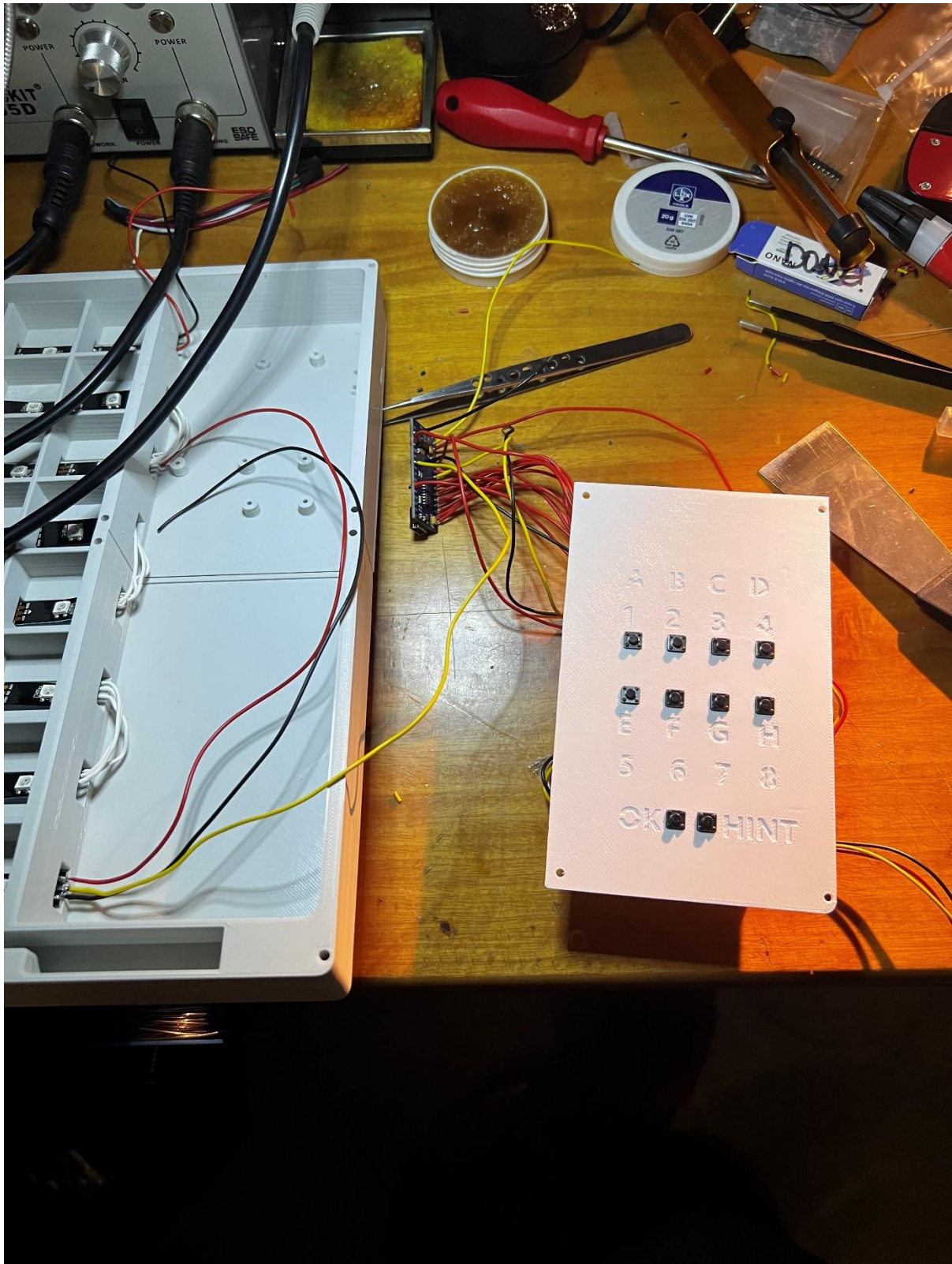


Εικόνα 2.2: Arduino Nano

Αφού πατήσουμε το κουμπί OK και η κίνησή μας είναι έγκυρη τότε και μόνο έχουμε ολοκληρώσει επιτυχώς την κίνησή μας. Σε διαφορετική περίπτωση αν η κίνησή μας δεν είναι σωστή το ταμπλό θα φωτίσει ένα μεγάλο κόκκινο X και στην οθόνη θα εμφανιστεί πως η κίνηση δεν ήταν σωστή.

Επιπλέον, υπάρχει και το κουμπί βοήθειας HINT το οποίο όταν το πατάμε μας προτείνει η σκακιέρα την καλύτερη κίνηση που μπορούμε να επιλέξουμε να κάνουμε.

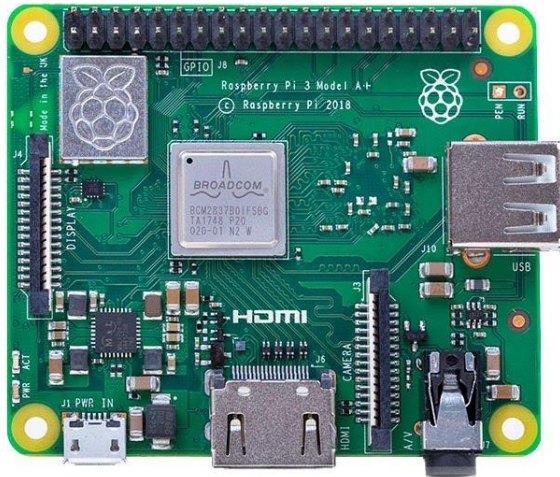
Για τις συνδέσεις στις εισόδους αλλά και εξόδους του μικροϋπολογιστή όπως και την επικοινωνία του με τον υπολογιστή θα μιλήσουμε στα επόμενα κεφάλαια.



Εικόνα 2.3: Συνδέσεις πίνακα ελέγχου και Arduino Nano

2.3 Υπολογιστής

Για την υλοποίηση της σκακιστικής μηχανής, με την οποία θα μπορούμε να αναμετρηθούμε, χρησιμοποιήθηκε ένας υπολογιστής τύπου Raspberry Pi 3A+. Επιλέχθηκε το συγκεκριμένο μοντέλο για τις επιδόσεις του όπως και την χαμηλή ενεργειακή του κατανάλωση. Όπως και για την δυνατότητα σύνδεσης πολλαπλών συσκευών με διαφορετικούς τρόπους επικοινωνίας [17] [18].



3A+

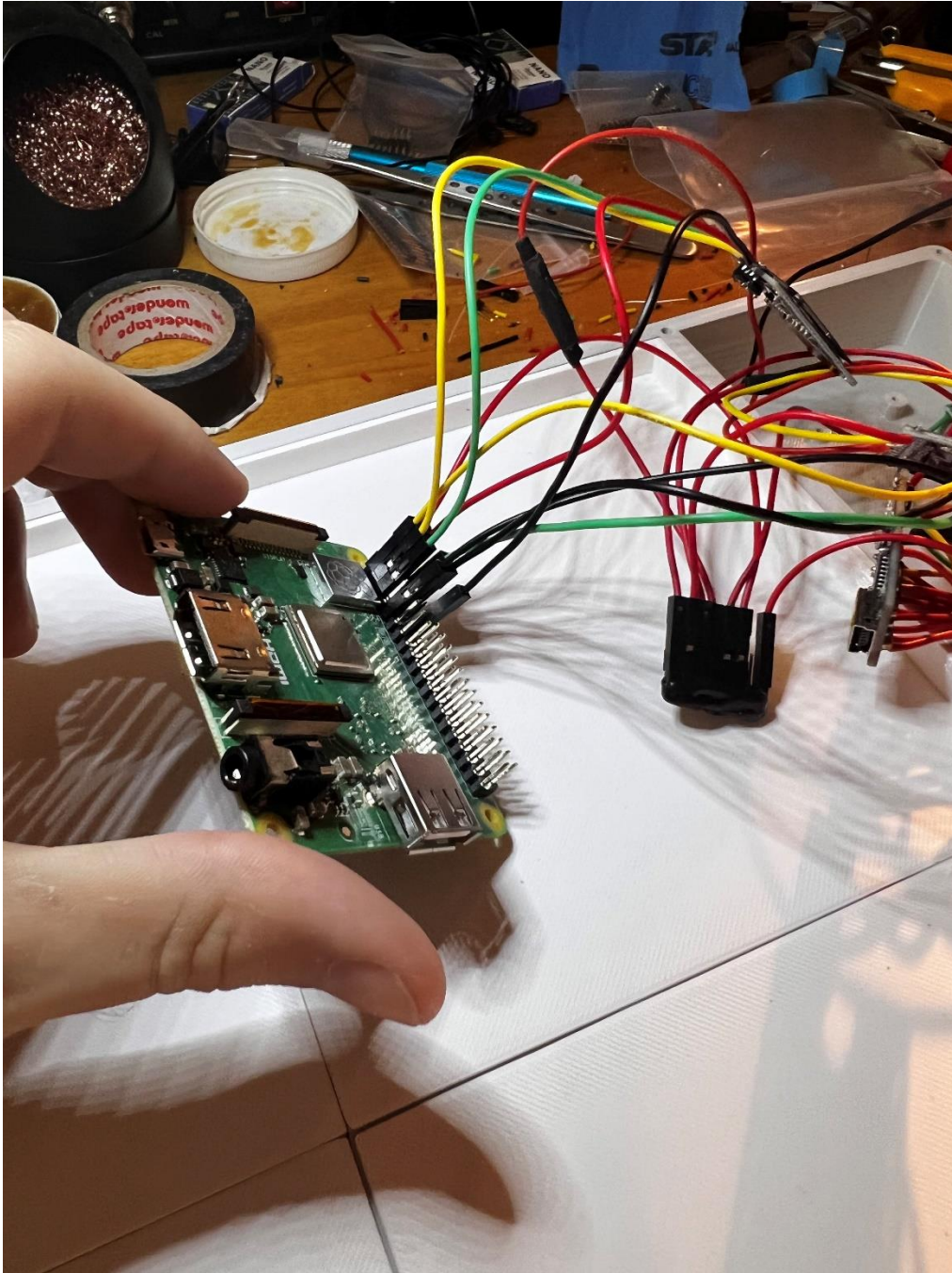
Εικόνα 2.4: Υπολογιστής Raspberry Pi

Σε αυτόν θα συνδεθεί το Arduino Nano που ελέγχει τα LEDs και η οθόνη που θα επικοινωνεί με το Raspberry Pi σειριακά. Επιπλέον ο υπολογιστής που επιλέχθηκε έχει μικρό μέγεθος ώστε να βρίσκεται εντός της κατασκευής. Αξίζει να σημειωθεί πως κατά την δοκιμαστική λειτουργία του εκτιμήθηκε πως η χρήση ψύξης ήταν απαραίτητη για την σταθερή λειτουργία του.



Εικόνα 2.5: Ανεμιστήρας εργασίας

Έτσι εγκαταστάθηκε στον επεξεργαστή ένα ειδικό μπλοκ χαλκού με θερμό αγωγήμη πάστα σε συνδυασμό με ένα ανεμιστήρα μικρού μεγέθους στο εξωτερικό της κατασκευής με προσανατολισμό pull, δηλαδή να «τραβάει» αέρα ή αλλιώς να απάγει την θερμότητα από το εσωτερικό της κατασκευής προς το εξωτερικό περιβάλλον.



Εικόνα 2.6: Raspberry Pi συνδέσεις

2.3.1 Τροφοδοτικό

Ο υπολογιστής μας ώστε να διασφαλιστεί η σωστή λειτουργία του θα πρέπει να του παρέχουμε και την κατάλληλη τροφοδοσία. Στην συγκεκριμένη πτυχιακή εργασία χρησιμοποιήθηκε ένα τροφοδοτικό των 5.1 Volt και 2.5 Ampere που είναι αρκετό για να τροφοδοτήσει τον υπολογιστή καθώς και τα υπόλοιπα ηλεκτρονικά στοιχεία της έξυπνης σκακιέρας.



Εικόνα 2.7: Τροφοδοτικό Raspberry pi

2.4 Φωτισμός και κουμπιά

Για την πτυχιακή εργασία έξυπνη σκακιέρα χρησιμοποιήθηκε συγκεκριμένη ταινία με LEDs για την απεικόνιση της επιφάνειας παιχνιδιού όπως και για τον φυσικό φωτισμό της σκακιέρας.

Η ταινία με τα LEDs είναι προγραμματιζόμενη και υπάρχει η δυνατότητα να ρυθμιστούν όπως χρειάζεται όλα τα LEDs.

Τα κουμπιά που χρησιμοποιήθηκαν είναι μηχανικά και θα αποτελέσουν το κύριο μέρος του πίνακα ελέγχου.



Εικόνα 2.8: Μηχανικό button

2.5 Λογικό επίπεδο

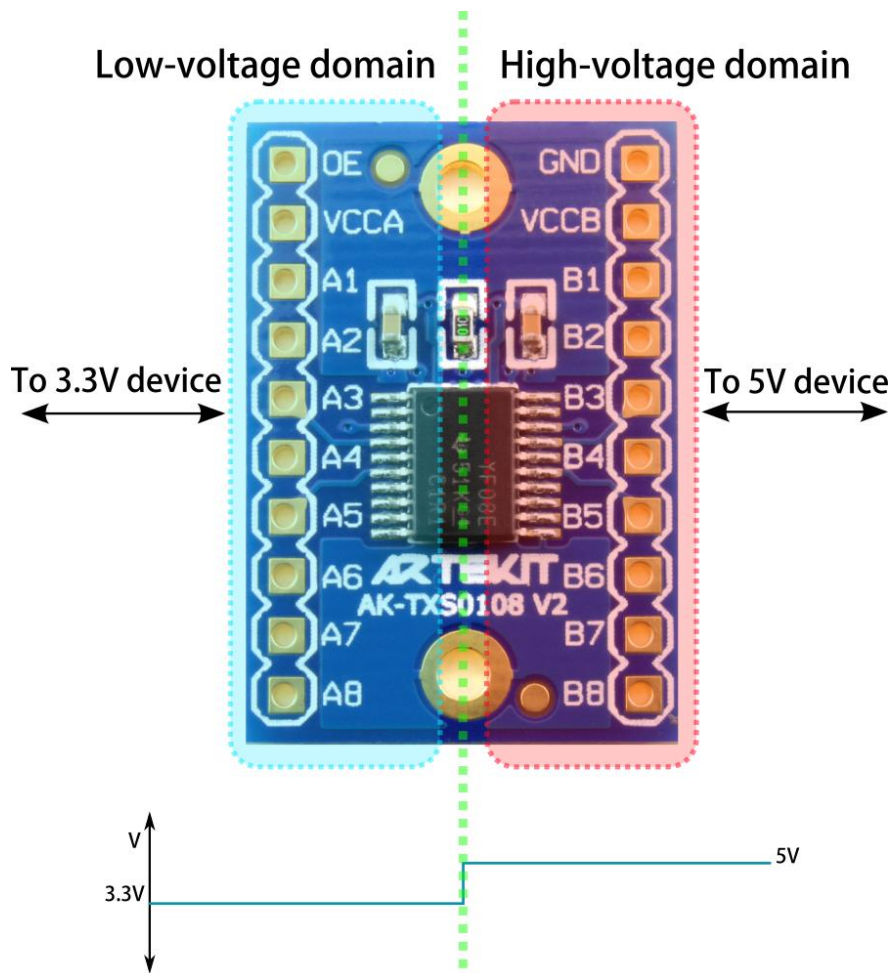
Για να μπορεί να παίζει με τον άνθρωπο η μηχανή-νοημοσύνη του σκάκι το λεγόμενο Stockfish θα πρέπει να επιτευχθεί η σωστή επικοινωνία μεταξύ του Arduino nano με τον υπολογιστή raspberry pi. Όπως και το πρόγραμμα για τον έλεγχο των κινήσεων το οποίο θα επιβλέπει τις κινήσεις του ανθρώπου

και θα καθορίζει αν η κίνηση του στη θέση που έχει επιλέξει συμβαδίζει με τον κανονισμό του παιχνιδιού.

Σε διαφορετική περίπτωση αν ο παίκτης παραβεί με την κίνησή του τους κανόνες τότε η σκακιέρα θα κάνει ένα ανάλογο εφέ και θα εμφανίσει μήνυμα στην οθόνη, όπου θα τον ενημερώνει πως η κίνησή του είναι εσφαλμένη. Υπάρχουν δύο επιλογές συνδεσμολογίας για να επιτύχουμε αυτό τον σκοπό.

Η πρώτη μας επιλογή είναι να δημιουργήσουμε σειριακή επικοινωνία ενεργοποιώντας την κατάλληλη σειριακή πόρτα του raspberry pi. Αυτό μπορούμε να το επιτύχουμε πηγαίνοντας στην εσωτερική διεύθυνση του pi και παραμετροποιώντας το αρχείο cmdline.txt όπου θα πρέπει να διαγράψουμε την γραμμή console=serial0.115200.

Η επικοινωνία μεταξύ του Arduino nano όπως και του Raspberry pi γίνεται σε λογικό επίπεδο. Όμως για να μπορέσουμε να επιτύχουμε την μεταξύ τους επικοινωνία θα πρέπει να χρησιμοποιήσουμε ένα μετατροπέα λογικού επιπέδου. Στη συγκεκριμένη εργασία χρησιμοποιήθηκε ο μετατροπέας **TXS0108E** [19].



Εικόνα 2.9: Logic level shifter

Ο λόγος που απαιτείτε αυτή η προσθήκη είναι επειδή οι δύο μηχανές λειτουργούν σε διαφορετικό λογικό επίπεδο. Συγκεκριμένα το Arduino Nano βρίσκεται στη περιοχή των 5 Volt ενώ το Raspberry pi στα 3,3 Volt. Με τη χρήση του παραπάνω μετατροπέα λογικού επιπέδου μπορούμε να επιτύχουμε αμφίδρομη επικοινωνία μεταξύ των συσκευών της εργασίας [16] [17] [18] [19].

Η δεύτερη επιλογή ώστε το σκακιστικό πρόγραμμα να έχει επικοινωνία με το Arduino nano είναι με ένα συμβατικό καλώδιο δεδομένων USB. Σε αυτή την περίπτωση θα πρέπει να ελέγξουμε το αρχείο που ξεκινάει με την εκκίνηση του υπολογιστή μας και να κάνουμε την απαραίτητη δήλωση για το πρωτόκολλο επικοινωνίας. Επομένως αυτό που χρειάζεται είναι να αλλάξουμε στην γραμμή 15 του κώδικα που τρέχει το παιχνίδι μας και να αλλάξουμε το ttyAMA0 σε ttyUSB0. Έτσι ενεργοποιούμε την επικοινωνία του υπολογιστή μας μέσω της USB θύρας του.

```
# initiate chessboard
from ChessBoard import ChessBoard
import subprocess, time, serial
maxchess = ChessBoard()

if __name__ == '__main__':
    ser = serial.Serial('/dev/ttyAMA0', 9600, timeout=1) # for Pi Zero use '/dev/ttyAMA0' and for others use '/dev/ttyUSB0'.
    ser.flush()
```

Εικόνα 2.10: Κομμάτι κώδικα σχετικό με την επικοινωνία των συσκευών

Σε αυτή την περίπτωση δεν χρειάζεται ο μετατροπέας λογικού επιπέδου που αναφέρθηκε παραπάνω καθώς η επικοινωνία γίνεται μέσω της θύρας USB και όχι από τα GPIOs του raspberry pi.

2.6 Οθόνη OLED

Για την απεικόνιση των κινήσεων και άλλων μηνυμάτων χρησιμοποιήθηκε μια οθόνη OLED 0.96 ιντσών. Η συγκεκριμένη οθόνη εγκαταστάθηκε στο raspberry pi ενεργοποιώντας την διεπαφή I²C στον υπολογιστή. Η οθόνη είναι αρκετά αποδοτική χωρίς να καταναλώνει πολλαπλούς πόρους από το συνολικό μας σύστημα κάτι που την καθιστά ιδανική επιλογή για την συγκεκριμένη Π.Ε.

Για να ενεργοποιήσουμε την επικοινωνία της οθόνης με το Raspberry pi θα πρέπει να μπούμε στις ρυθμίσεις του υπολογιστή μας χρησιμοποιώντας την εντολή `sudo raspi-config` και έπειτα να μεταβούμε στο απαραίτητο μενού και να επιλέξουμε την ενεργοποίηση της διεπαφής I²C.



Εικόνα 2.11: Οθόνη OLED

Όμως για να είναι λειτουργική η οθόνη μας θα πρέπει να εγκαταστήσουμε ορισμένες βιβλιοθήκες. Στη συγκεκριμένη περίπτωση επειδή χρησιμοποιούμε γλώσσα Python 2.7 εγκαταστάθηκαν και οι αντίστοιχες βιβλιοθήκες που είναι συμβατές με την συγκεκριμένη γλώσσα [20].

2.7 Επίλογος

Σε αυτό το κεφάλαιο αναφέρθηκε όλο το ηλεκτρονικό υλικό που χρησιμοποιήθηκε στην εν λόγω Π.Ε. και διευκρινίστηκαν σημαντικές πληροφορίες για την οθόνη OLED και το λογικό επίπεδο. Επιπλέον έγινε αναφορά και στην τροφοδοσία της συγκεκριμένης εργασίας καθώς και στα σημαντικότερα υλικά που είναι ο υπολογιστής raspberry pi και ο μικροεπεξεργαστής Arduino nano. Σε βάθος για τις διαδικασίες εγκατάστασης και τον προγραμματισμό των ηλεκτρονικών στοιχείων θα ακολουθηθεί σε επόμενο κεφάλαιο.

Κεφάλαιο 3ο: Συναρμολόγηση ηλεκτρονικού υλικού

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα αναπτυχθούν τα βήματα αλλά και οι διαδικασίες που ακολουθήθηκαν ώστε να υλοποιηθεί το κατασκευαστικό κομμάτι της έξυπνης σκακιέρας. Θα γίνει πλήρη επεξήγηση για την συνεισφορά του ηλεκτρονικού υλικού και την συνεργασία του με τα υπόλοιπα στοιχεία της Π.Ε.

3.2 Φωτισμός

Η ταινία που φέρει επάνω της τα προγραμματιζόμενα LEDs χρειάστηκε να τεμαχιστεί σε οκτώ λωρίδες οι οποίες θα περιλαμβάνουν οκτώ LEDs η κάθε μία ξεχωριστά. Εν συνεχεία, ώστε να δημιουργηθεί το απαραίτητο κύκλωμα μεταξύ αυτών, χρειάστηκαν κατά κύριο λόγο κοντά καλώδια μικρής διαμέτρου και τύπου πολύκλιωνα. Να σημειωθεί πως η συγκεκριμένη αναφορά γίνεται για τα LEDs που αποτελούν την επιφάνεια παιχνιδιού. Για τα περιφερειακά LEDs που θα φωτίζουν τις συντεταγμένες αλλά και τον πίνακα ελέγχου της σκακιέρας θα αναφερθούμε παρακάτω στο συγκεκριμένο κεφάλαιο.



Εικόνα 3.1: Ταινία LED

Η τεμαχισμένη ταινία απαιτεί σύνδεση σε σειρά ώστε να μπορούμε να την προγραμματίσουμε και να απευθυνθούμε στο κάθε LED ξεχωριστά. Χρησιμοποιώντας ένα κολλητήριο των 50 Watt κολλήθηκαν στην αρχή της πρώτης ταινίας με τα οκτώ LEDs τρία καλώδια. Το ένα για τα 5 Volt της τροφοδοσίας, ένα για την λογική είσοδο ελέγχου της ταινίας και ένα φυσικά για την γείωση της ταινίας.

Σε αυτό το σημείο θα πρέπει να διευκρινιστεί πως για να ενσωματωθούν στην εργασία και οι οκτώ βασικές ταινίες της επιφάνειας του παιχνιδιού θα πρέπει να τοποθετηθούν με μέθοδο “ZIG-ZAG”, αυτό διότι το σχήμα της σκακιάρας είναι τετράγωνο και οι ταινίες θα πρέπει να είναι τοποθετημένες σε σειρά. Οπότε καταλαβαίνουμε πως τα καλώδια που βρίσκονται στις «κλειδώσεις» της ταινίας πρέπει να είναι αρκετά κοντά σε ύψος και σε διαφορετικό μέγεθος, μεγαλύτερο προς μικρότερο, ώστε να μπορούν να φέρουν στροφή 180 μοιρών εντός της τρισδιάστατης κατασκευής της σκακιάρας.

Έτσι ολοκληρώθηκε η κόλληση των ταινιών, λαμβάνοντας πάντα υπόψη την πολικότητά τους, που θα είναι υπεύθυνες για την απεικόνιση του παιχνιδιού. Κολλήθηκαν στο τέλος της ταινίας που βρίσκεται στην κορυφή δύο καλώδια μεγαλύτερα σε μήκος, ένα για την τροφοδοσία των 5 Volt και ένα για την γείωση του συστήματος.

Για τον φωτισμό του πίνακα ελέγχου χρειάστηκαν 3 ταινίες που να έχουν από 2 LEDs η κάθε μία. Η σύνδεση τους γίνεται σε σειρά όπως και των ταινιών για την επιφάνεια του παιχνιδιού. Κολλήθηκαν σε αυτές καλώδια για την τροφοδοσία τους, την γείωσή τους αλλά και για την ψηφιακή τους είσοδο.

3.3 Κουμπιά και συνδεσμολογία

Τα κουμπιά αποτελούν ίσως ένα από τα βασικότερα τμήματα αυτής της κατασκευής. Ο λόγος είναι πως αποτελούν το μέσω ώστε να δίνουμε της απαραίτητες πληροφορίες για τις κινήσεις μας στην σκακιάρα και συγκεκριμένα στο Arduino Nano.

Στη συγκεκριμένη Π.Ε. χρησιμοποιήθηκαν 10 buttons από τα οποία 8 είναι για τις συντεταγμένες της σκακιάρας και 2 buttons ένα για την έγκριση των κινήσεων του παίκτη και ένα για την χρήση βοήθειας κατά την διάρκεια του παιχνιδιού.

Από την ομάδα των buttons χρειάζεται να γνωρίζουμε πως έχουν συνδεσμολογηθεί σε σειρά όπως έχει αναφερθεί σε προηγούμενο κεφάλαιο. Ως αποτέλεσμα τα buttons χρησιμοποιούν κοινή γείωση το οποίο έχει σαν αποτέλεσμα για την συγκεκριμένη Π.Ε. να εξοικονομήσουμε χώρο εντός της κατασκευής καθώς όλα θα πρέπει να είναι τακτοποιημένα στο εσωτερικό της.

Έπειτα ακολούθησε η κόλληση, ξεχωριστών για το κάθε ένα button, καλωδίων ώστε να μπορεί να είναι ανιχνεύσιμη ανά πάσα στιγμή από τον μικροϋπολογιστή μας η λογική τους κατάσταση. Σε αυτό το σημείο αξίζει να αναφερθεί πως κατά την εκτύπωση του μοντέλου για τον πίνακα ελέγχου σχεδιάστηκαν εσοχές οι οποίες βοήθησαν στην κατανομή και τακτοποίηση των καλωδίων εντός της κατασκευής της Π.Ε.

Μπορούμε να ελέγξουμε την λειτουργία των buttons χρησιμοποιώντας ένα πολύμετρο στην λειτουργία συνέχειας. Συνδέοντας την γείωση των buttons με την γείωση του πολυμέτρου και τα ξεχωριστά καλώδια για το κάθε ένα κουμπί. Πατώντας κάθε φορά το κουμπί που θέλουμε να πάρουμε μέτρηση θα πρέπει να ακούγετε ο χαρακτηριστικός ήχος από το buzzer του πολυμέτρου. Έτσι μπορούμε να εξασφαλίσουμε πως όλα μας τα κουμπιά λειτουργούν κανονικά.

3.4 Συνδέσεις μικροϋπολογιστή

Το ηλεκτρονικό στοιχείο που θα ελέγχει τα LEDs καθώς και buttons της έξυπνης σκακιάρας είναι ένα Arduino Nano με επεξεργαστή ATmega328P. Σε αυτό το σημείο χρειάζεται να ολοκληρωθεί η

εγκατάσταση των buttons και για αυτό θα πρέπει να κολληθούν στις κατάλληλες εισόδους του συγκεκριμένου μικροϋπολογιστή.

Ξεκινώντας με την κόλληση των αυτόνομων καλωδίων των κουμπιών του πίνακα ελέγχου της έξυπνης σκακιέρας έχουμε το κουμπί A1 που συνδέεται στην είσοδο D5 του Arduino Nano. Το κουμπί B2 στην είσοδο D6, το κουμπί C3 στην είσοδο D7. Το κουμπί D4 στην είσοδο D8, το κουμπί E5 στην είσοδο D9, το κουμπί F6 στην είσοδο D10, το κουμπί G7 στην είσοδο D11 και το κουμπί H8 στην είσοδο D12.

Το κουμπί OK υπεύθυνο για την πιστοποίηση των κινήσεων του παίκτη συνδέεται στην είσοδο A1 του Arduino Nano. Επειδή το κουμπί βοήθειας “HINT” θα χρησιμοποιηθεί ως μια διακοπή ή αλλιώς interrupt στον προγραμματισμό του Arduino χρειάζεται να συνδεθεί σε μία από τις δύο εισόδους D2 ή D3 σύμφωνα με τον κατασκευαστή.

Πίνακας 3.1: Interrupt pins που μπορούν να χρησιμοποιηθούν ως διακοπές κατά τα διάφορα μοντέλα Arduino

BOARD	DIGITAL PINS USABLE FOR INTERRUPTS
Uno, Nano, Mini, other 328-based	2, 3
Uno WiFi Rev.2, Nano Every	all digital pins
Mega, Mega2560, MegaADK	2, 3, 18, 19, 20, 21 (pins 20 & 21 are not available to use for interrupts while they are used for I2C communication)
Micro, Leonardo, other 32u4-based	0, 1, 2, 3, 7
Zero	all digital pins, except 4
MKR Family boards	0, 1, 4, 5, 6, 7, 8, 9, A1, A2
Nano 33 IoT	2, 3, 9, 10, 11, 13, A1, A5, A7
Nano 33 BLE, Nano 33 BLE Sense	all pins
Due	all digital pins
101	all digital pins (Only pins 2, 5, 7, 8, 10, 11, 12, 13 work with CHANGE)

Στην συγκεκριμένη περίπτωση χρησιμοποιήθηκε η είσοδος D2 του Arduino Nano για την σύνδεση του κουμπιού “HINT”.

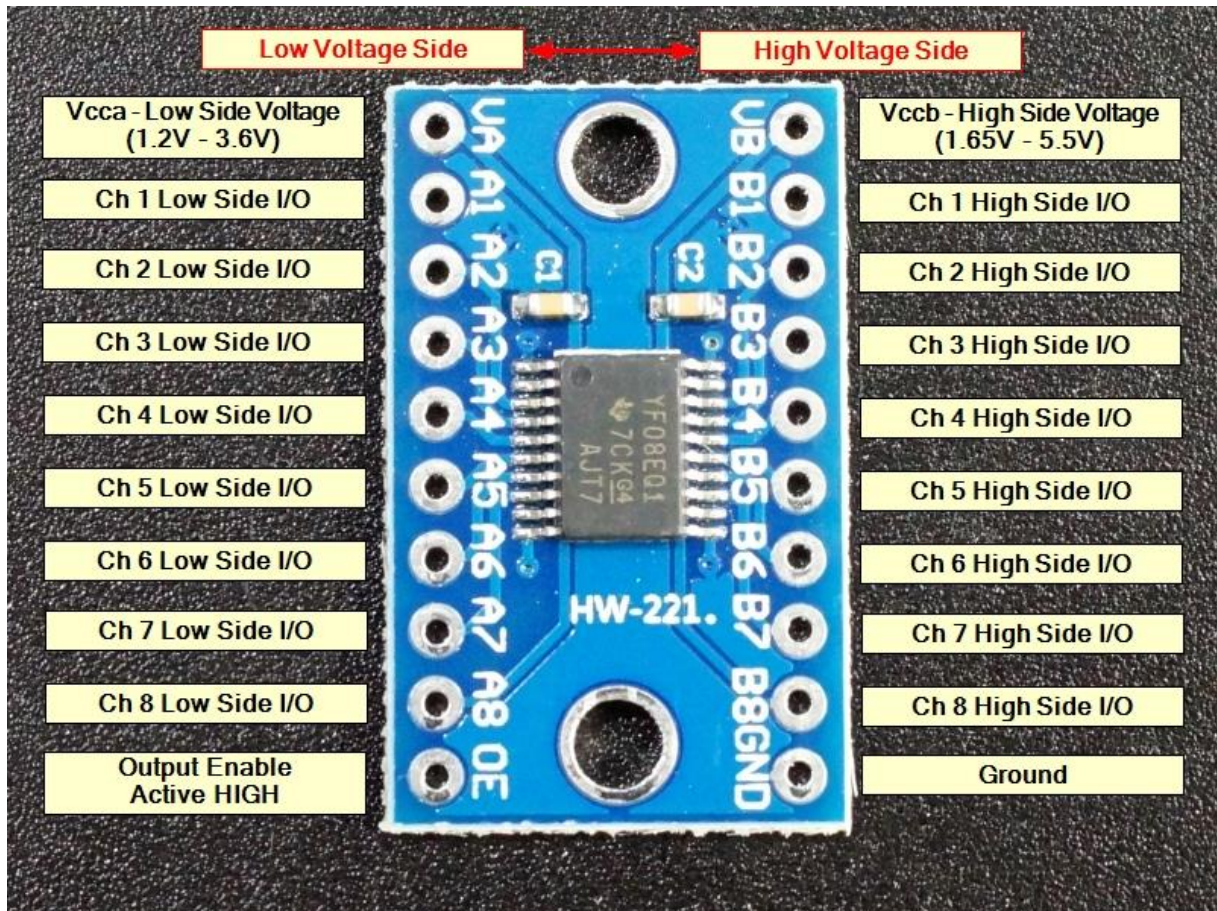
Η κοινή γείωση των καλωδίων συνδέθηκε στην γείωση του Arduino Nano στο επάνω μέρος.

Στο Arduino χρειάζεται να κολληθούν επίσης και τα κυκλώματα που αποτελούνται από τις ταινίες των LEDs, ώστε να μπορούν να προγραμματιστούν. Τα LEDs υπεύθυνα για τον φωτισμό του πίνακα ελέγχου συνδέονται με το Arduino Nano χρησιμοποιώντας την ψηφιακή τους είσοδο από την πρώτη στην σειρά ταινία στο ποδαράκι D4 του Arduino.

Για να συνδεθούν τα LEDs της επιφάνεια παιχνιδιού με το Arduino χρησιμοποιούμε όπως και προηγουμένως το μεσαίο καλώδιο που κολλήθηκε για την ψηφιακή είσοδο τους στο ποδαράκι A5 του Arduino.

3.5 Προσαρμογή λογικής κατάστασης

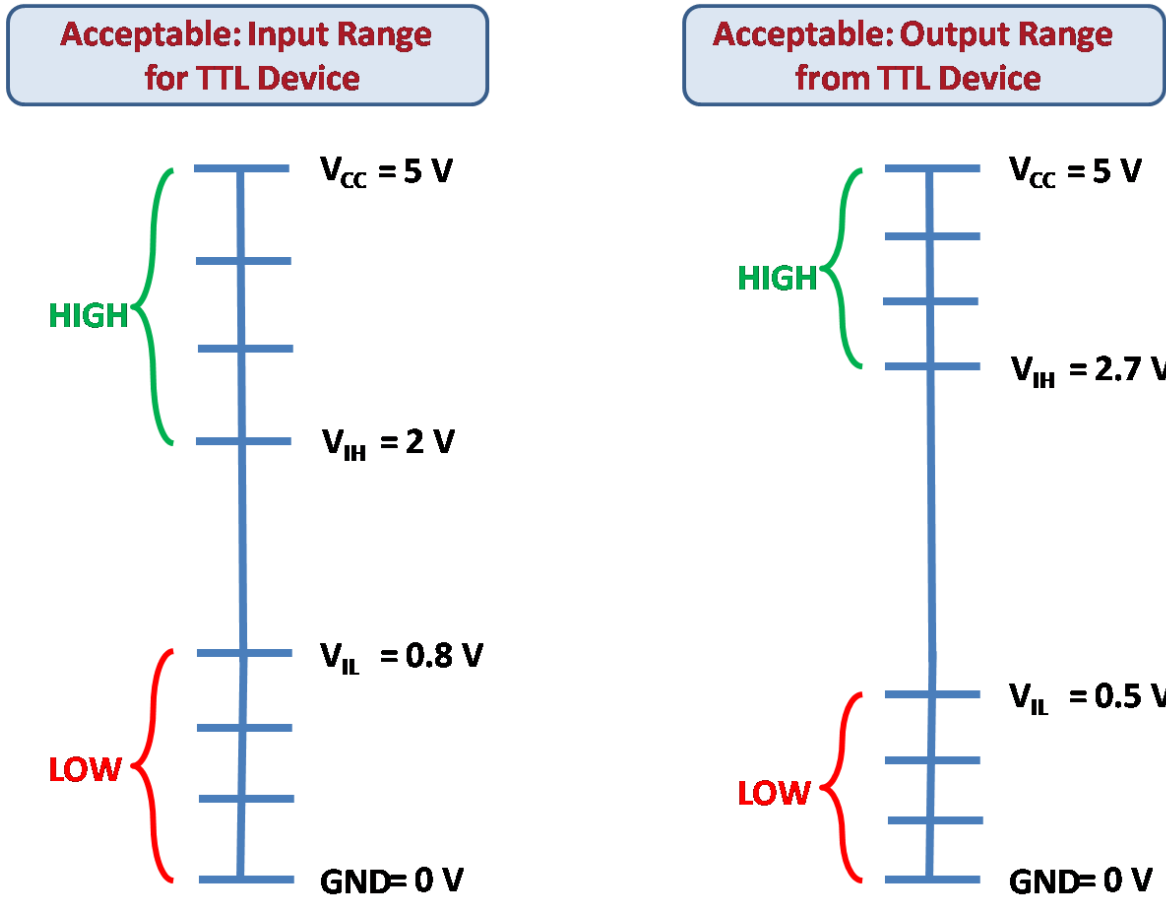
Όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο χρειάζεται να έχουμε αμφίδρομη επικοινωνία του μικροϋπολογιστή Arduino Nano και του υπολογιστή Raspberry pi. Σε αυτό θα μας βοηθήσει ο μετατροπέας λογικού επιπέδου **TXS0108E** [19].



Εικόνα 3.2: Μετατροπέας λογικού επιπέδου

Κολλήθηκαν καλώδια μεταξύ στα ποδαράκια B1 και B2 του μετατροπέα και τα ποδαράκια TX1 και RX0 του Arduino αντίστοιχα. Για την τροφοδοσία του μετατροπέα συνδέθηκε το ποδαράκι VB με την τροφοδοσία των 5 Volt και το ποδαράκι GND με την γείωση. Έπειτα χρειάζεται να κολληθούν στα ποδαράκια VA, A1, A2 του μετατροπέα ξεχωριστά καλώδια.

Σημαντικό είναι να αναφερθεί πως ο συγκεκριμένος μετατροπέας έχει ένα διακόπτη ON/OFF και βρίσκεται στο ποδαράκι OE (Output Enable). Όταν αυτό το ποδαράκι είναι σε λογική κατάσταση LOW τότε όλα τα pins του αποκτούν μεγάλη ωμική αντίσταση. Αυτό έχει αποτέλεσμα να διακοπεί η επικοινωνία μεταξύ των δύο πλευρών του.



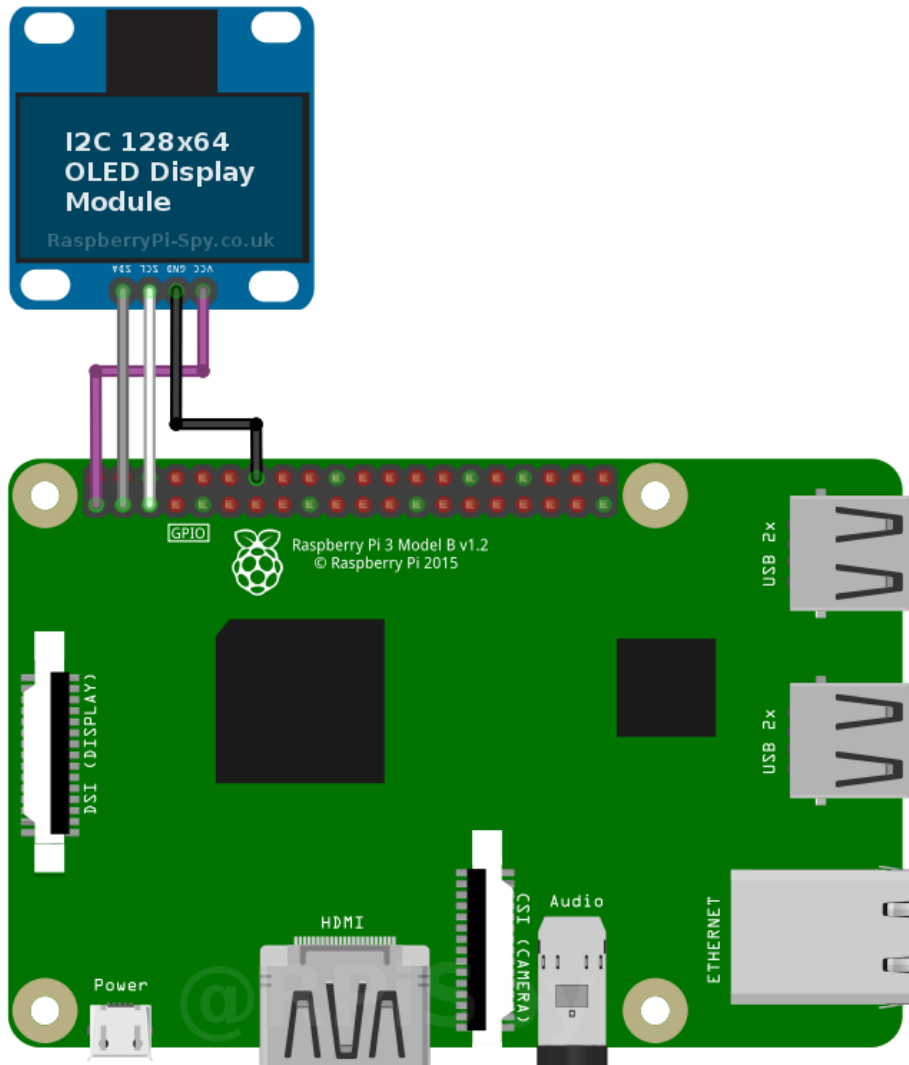
Σχήμα 3.1: Λογικά επίπεδα

Για αυτό χρειάζεται να συνδέσουμε το ποδαράκι OE του μετατροπέα με το ποδαράκι VA με ένα άλλο ξεχωριστό καλώδιο. Έτσι διασφαλίζουμε πως η επικοινωνία μεταξύ των συσκευών μας ξεκινάει μόνο όταν έχει ενεργοποιηθεί ο υπολογιστής μας.

Τα ποδαράκια του μετατροπέα λογικού επιπέδου VA, A2 και A1 συνδέονται στα pins 1, 8 και 10 του Raspberry pi αντίστοιχα.

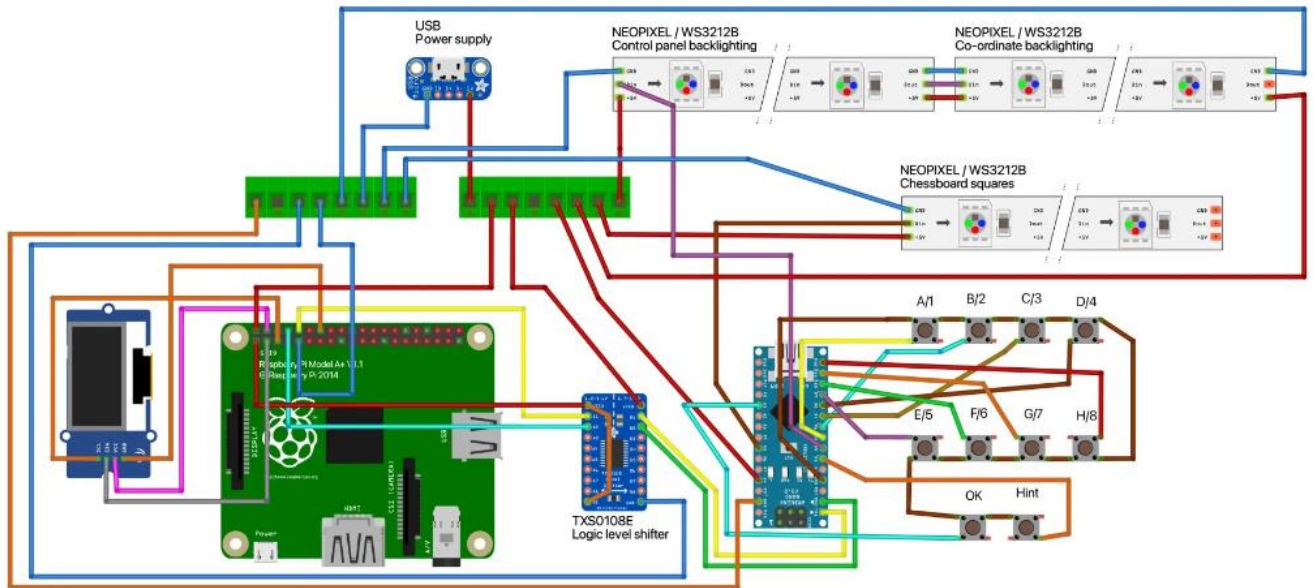
3.6 Σύνδεση οθόνης

Για να μπορέσουμε να απεικονίσουμε τις κινήσεις που γίνονται κατά την διάρκεια του παιχνιδιού, την προειδοποίηση λανθασμένης κίνησης, οδηγίες επανεκκίνησης ή τερματισμού λειτουργίας της σκακιάρας χρειάζεται να συνδέσουμε την οθόνη OLED με το Raspberry pi.



Σχήμα 3.2: Εναλλακτική απεικόνιση σύνδεσης οθόνης με το raspberry pi

Συγκεκριμένα τα pins SDA, VCC, SCL και GND της οθόνης με τα pins 3, 4, 5 και 14 του Raspberry pi αντίστοιχα. Έτσι έχουμε συνδέσει σειριακά την οθόνη για επικοινωνία με τον υπολογιστή μας.



Εικόνα 3.3: Block διάγραμμα των συνδέσεων των ηλεκτρονικών στοιχείων της κατασκευής

3.7 Επίλογος

Στο κεφάλαιο αυτό έγινε εκτενής αναφορά για τα χαρακτηριστικά του ηλεκτρονικού υλικού, τις συνδέσεις μεταξύ αυτών και επεξηγήθηκε η μέθοδος όλων των εργασιών τους. Ουσιαστικά σε αυτό το κεφάλαιο δόθηκαν σημαντικές οδηγίες για την κατασκευή ως προς την λειτουργία και επικοινωνία του φυσικού υλικού αναλύοντας λεπτομερώς όλες τις συνδέσεις και τον λόγο που έγιναν.

Στο επόμενο κεφάλαιο θα επικεντρωθούμε στο λογισμικό περιβάλλον της κατασκευής ώστε να «δέσει» με τις πληροφορίες αυτού του κεφαλαίου. Έτσι, ολοκληρώνεται η παρουσίαση και επεξήγηση του φυσικού υλικού της Π.Ε.

Κεφάλαιο 4ο: Εγκατάσταση λογισμικού και προγραμματισμός Arduino

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνουν αναφορές στο κομμάτι του προγραμματισμού της εν λόγω Π.Ε. και θα δοθούν επεξηγηματικές οδηγίες ως προς την υλοποίηση της. Τα κομμάτια κώδικα που συνεργάζονται σε αυτή την εργασία είναι αποτέλεσμα έρευνας από το διαδίκτυο και έχουν χρησιμοποιηθεί με τέτοιο τρόπο ώστε να υπάρχει βέλτιστη λειτουργία μεταξύ του ηλεκτρονικού υλικού της έξυπνης σκακιέρας.

Όσον αφορά την επικοινωνία αλλά και διαχείριση των εργασιών που εκτελούνται στο παρασκήνιο. Έχουν γίνει αρκετές προσαρμογές σε αυτά, όπως και βελτιστοποίηση κατά τον προγραμματισμό τους. Στις προσαρμογές αυτές θα αναφερθούμε πιο αναλυτικά στην συνέχεια του κεφαλαίου αυτού.

Ακόμη, θα μιλήσουμε αναλυτικά για τον προγραμματισμό των συσκευών μας, την σκακιστική μηχανή-νοημοσύνη, την διαιτησία του παιχνιδιού αλλά και την αυτόματη ενεργοποίηση του κώδικα του Raspberry pi κατά την εκκίνηση της συσκευής.

4.2 Γλώσσα και προγραμματισμός Arduino

Η γλώσσα που χρησιμοποιείται στο Arduino αναφέρεται πολλές φορές στην αγγλική γλώσσα ως sketch. Ουσιαστικά είναι όμως C++ με κάποιες επιπλέον μεθόδους και συναρτήσεις. Για τον προγραμματισμό του Arduino Nano που έχει χρησιμοποιηθεί στην συγκεκριμένη Π.Ε., χρειάστηκε η εγκατάσταση το προγράμματος μετάφρασης της εταιρίας του κατασκευαστή, Arduino IDE.

Ο κώδικας ή αλλιώς το λειτουργικό που κατασκευάστηκε αποτελείται από δύο μέρη. Το πρώτο είναι υπεύθυνο για την δήλωση σημαντικών μεγεθών όπως ο πίνακας παιχνιδιού, η λειτουργία LOADING ή αλλιώς εκκίνησης της συσκευής, ο φυσικός φωτισμός της σκακιέρας, η απεικόνιση των κινήσεων και οι παράμετροι φωτεινών ενδείξεων που συμβαδίζουν πάντα με τους κανονισμούς του παιχνιδιού.

```

void boardsquares(){ //marking the squares of the chessboard
  for(int i = 0; i < 8; i=i+2) {
    for(int j = 0; j < 8; j++) {
      if (j % 2 == 0){
        boardlights.drawPixel(i,j,WHITE);
        boardlights.drawPixel(i+1,j,BLACK);
      }
      else{
        boardlights.drawPixel(i,j,BLACK);
        boardlights.drawPixel(i+1,j,WHITE);
      }
    }
  }
  boardlights.show();
}

int loadingboard(int Squareslighted){ // animate loading status while waiting for pi to start
  int Row = 0;
  int Column = 0;
  if (Squareslighted < 8){
    Row = 0;
    Column = Squareslighted;
  } else if (Squareslighted > 7){
    Row = Squareslighted / 8;
    Serial.println(Row);
    Column = Squareslighted % 8;
    Serial.println(Column);
    Serial.println();
  }
  boardlights.drawPixel(Column,Row,GREEN);
  boardlights.show();
  return Squareslighted + 1;
}

```

Εικόνα 4.1: Κώδικας υπεύθυνος για την απεικόνιση των τετραγώνων της σκακιέρας και εφέ εκκίνησης συσκευής

Το δεύτερο μέρος είναι υπεύθυνο για την σωστή επικοινωνία του Arduino Nano με το Raspberry pi κατά το οποίο θα επιβεβαιώνει ή όχι ανάλογα την εκάστοτε κίνηση με το Arduino με σκοπό την ανανέωση του παιχνιδιού σύμφωνα με την πιο πρόσφατη θέση όλων των κομματιών. Πολύ σημαντικό είναι και η διαχείριση των βοηθειών που κατόπιν επικοινωνίας του Arduino με το Raspberry pi το τελευταίο με την μηχανή νοημοσύνη μας προτείνει με φωτεινή ένδειξη την καλύτερη δυνατή κίνηση για το συγκεκριμένο παιχνίδι σε συγκεκριμένο χρόνο. Αυτό εξασφαλίζει την σωστή ροή των παρτίδων που λαμβάνουν χώρα στην εν λόγω κατασκευή.

```

//setup the hint interrupt pin
attachInterrupt(digitalPinToInterrupt(3), hint, RISING);

```

Εικόνα 4.2: Δήλωση του pin 3 ως διακοπή HINT

Στη κονσόλα που τρέχει ο κώδικας υπάρχουν μηνύματα κειμένου που αποστέλλονται από και προς τις συσκευές καθ' όλη την διάρκεια μιας παρτίδας, περισσότερο για την επιβεβαίωση πως όλα ρέουν ομαλά χωρίς να υπάρχει μεγάλο αντίκτυπο στην αποδοτικότητα του έργου. Αυτό το σημείο δεν είναι ορατό στον καταναλωτή-παίκτη του οποίου η προτεραιότητα είναι να παίζει την παρτίδα του χωρίς αχρείαστους περισπασμούς.

Είναι σημαντικό να αναφερθεί ότι, χρειάστηκε να γίνει σωστή κατανομή στην μνήμη του Arduino κατά τον προγραμματισμό του διότι εμφάνιζε κατά τα αρχικά στάδια δοκιμών παρακώληση των εργασιών που εκτελούνταν στο παρασκήνιο. Αυτό είχε ως συνέπεια να γεμίζει η μνήμη του κατά την μεταφορά πληροφοριών από και προς το Arduino με αποτέλεσμα να δημιουργείτε αστάθεια κατά την λειτουργία του. Στέλνοντας αυτά τα αρχεία να αποθηκεύονται στην flash memory αντί για την κανονική του παρατηρήθηκε σημαντική βελτίωση μεταξύ της επικοινωνίας προς το Raspberry pi. Εξάλλου τα μηνύματα αυτά δεν είναι ορατά από τον παίκτη αλλά απαραίτητα για την αναγνώριση της εισόδου από τον παίκτη.

Εξίσου σημαντικό είναι να αναφερθεί πως υπάρχουν παράμετροι επανεκκίνησης παιχνιδιού ή τερματισμού αυτού με σχετικό κείμενο που εμφανίζεται στην οθόνη και μας ενημερώνει για τις λειτουργίες που έχουμε επιλέξει. Στην παρακάτω φωτογραφία μπορούμε να δούμε τον κώδικα ο οποίος είναι υπεύθυνος για την διακοπή Hint όπου με την οποία μπορούμε να ενεργοποιήσουμε και τις λειτουργίες που αναφέρθηκαν παραπάνω.

```

void hint(){
  static unsigned long last_interrupt_time = 0;
  unsigned long interrupt_time = millis();
  // to prevent button bounce under 200ms
  if (interrupt_time - last_interrupt_time > 200)
  {
    if (digitalRead(A1) == LOW){
      Serial.println(F("Starting a new game..."));
      sendPidata(F(""),F("n"));
      initialBoard(); //reset all the chess piece positions
      ButtonLEDS.setPixelColor(5, 0, 0, 0); //turn off the hint button on the control panel
      ButtonLEDS.fill(cpWHITE, 0, 5); //control panel white color leds
      ButtonLEDS.show();
      int var1 = 0;
      while (var1 < 64){
        var1 = loadingboard(var1);
        delay(25);
      }
      delay(1000);
      boardsquares(); //show board squares
      return;
    } else if (digitalRead(12) == LOW){
      sendPidata(F("shutdown"), F("x"));
    } else if (BestMovefrompi.length() != 1){
      ButtonLEDS.fill(cpBLACK, 0, 4);
      ButtonLEDS.setPixelColor(5, 0, 0, 255); //light up the hint button on the control panel in blue color
      ButtonLEDS.show(); // black out button lights
      lightUpMove(BestMovefrompi, 'H');
      boardsquares(); //show board squares
      ButtonLEDS.setPixelColor(5, 255, 255, 255); //light up the hint button on the control panel in white again
      ButtonLEDS.fill(cpWHITE, 0, 4);
      ButtonLEDS.show(); // light up button lights in white color
    } else {
      Serial.println(F("No hint provided by pi yet.));
    }
  }
  last_interrupt_time = interrupt_time;
}

```

Εικόνα 4.3: Hint interrupt

4.3 Γλώσσα raspberry pi

Με τον συγκεκριμένο υπολογιστή και τις δυνατότητές του που έχουν αναφερθεί σε προηγούμενο κεφάλαιο τρέχουν κατά την εκκίνησή του πολύ σημαντικά προγράμματα για την υλοποίηση των παιχνιδιών. Τα εν λόγω προγράμματα είναι η σκακιστική μηχανή stockfish, το maxchess που εξυπηρετεί ως διαιτητής του παιχνιδιού και ελέγχει κάθε κίνηση. Έπειτα αποστέλλει errors στο Arduino για το αν μια κίνηση βρίσκεται εντός των κανονισμών, σε διαφορετική περίπτωση παραμένει σιωπηλά στο παρασκήνιο εκτελώντας το καθήκον του κάθε φορά.

Η αυτόματη ενεργοποίηση των παραπάνω προγραμμάτων κατά την εκκίνηση του υπολογιστή που έχει ως σκοπό αφού ξεκινήσει η λειτουργία τους να ενημερώνεται μέσω αυτών και το Arduino ώστε να ανανεώσει τις φωτεινές ενδείξεις του.

Επίσης υπάρχει και η βιβλιοθήκη Chessboard έκδοσης v2.05 η οποία είναι κατασκευασμένη από τον John Eriksson και υπάγεται σε σχετική άδεια δημόσιας χρήσης GPL, δηλαδή αποτελεί ελεύθερο ανοικτό λογισμικό προς οποιαδήποτε χρήση είτε αυτό συνεπάγεται στην αναπαραγωγή του, την παραμετροποίηση, τροποποίηση του ή και την διανομή του από τον οποιοδήποτε.

Κεφάλαιο 4

Στο λογισμικό που έχει αναφερθεί παραπάνω υπάρχει η δυνατότητα ασύρματης επικοινωνίας μέσω του διαδικτύου μεταξύ δύο ίδιων έξυπνων σκακιερών. Λόγο του κόστους κατασκευής δυστυχώς δεν υλοποιήθηκε η δεύτερη για την παρουσίαση της συγκεκριμένης δυνατότητας. Παρ' αυτά θεωρητικά θα πρέπει να δουλεύει κανονικά όταν στην εκκίνηση της συσκευής επιλέξουμε την λειτουργία Online όπου μέσω της υπηρεσίας ADAFRUIT έχοντας εκχωρήσει το μοναδικό κλειδί μας, θα μπορούμε να συνδέσουμε τις δύο σκακιέρες μέσω του διαδικτύου για Online παιχνίδι.

```
def sendToScreen(line1,line2,line3,size = '14'):
    """Send three lines of text to the small OLED screen"""
    screenScriptToRun = ["python3", "/home/pi/SmartChess/RaspberryPiCode/printToOLED.py", '-a '+ line1, '-b '+ line2, '-c '+ line3, '-s '+ size]
    subprocess.Popen(screenScriptToRun)

#Choose a mode of gameplay on the Arduino
time.sleep(1)
sendtoboard("ChooseMode")
print ("Waiting for mode of play to be decided on the Arduino")
sendToScreen ('Choose opponent:', '1) Against PC', '2) Remote human')
gameplayMode = getboard()[1:].lower()
print ("Requested gameplay mode:")
print(gameplayMode)

if gameplayMode == 'stockfish':
    while True:
        sendtoboard("ReadyStockfish")

        # get intial settings (such as level)
        print ("Waiting for level command to be received from Arduino")
        sendToScreen ('Choose computer','difficulty level:', '(0 -> 8)')
        skillFromArduino = getboard()[1:3].lower()
        print ("Requested skill level:")
        print(skillFromArduino)

        # get intial settings (such as move time)
        print ("Waiting for move time command to be received from Arduino")
        sendToScreen ('Choose computer','move time:', '(0 -> 8)')
        movetimeFromArduino = getboard()[1:].lower()
        print ("Requested time out setting:")
        print(movetimeFromArduino)
```

Εικόνα 4.4: Συνάρτηση για αποστολή text στην οθόνη OLED

Στη παραπάνω εικόνα βλέπουμε την δημιουργία μιας συνάρτησης υπεύθυνης για την ενεργοποίηση της οθόνης κατά την οποία θα αποστέλλονται τα μηνύματα κειμένου προς αυτή. Επίσης μπορούμε να διακρίνουμε και τα μηνύματα κειμένου που λαμβάνουμε κατά την επιλογή παιχνιδιού και χρόνου εάν αυτή είναι η σκακιστική μηχανή Stockfish.

Για την απεικόνιση των μηνυμάτων στην οθόνη LCD χρησιμοποιήθηκε κώδικας που κατασκευάστηκε από παραδείγματα στο διαδίκτυο από την εταιρία Adafruit. Μπορεί να χρησιμοποιηθεί σε υπολογιστές με βάση το λογισμικό Linux που χρησιμοποιούν την Cpython με την Adafruit Blinka ώστε να μπορούν να υποστηριχθούν οι βιβλιοθήκες CircuitPython. Η CircuitPython δεν μπορεί να υποστηρίξει όμως την απεικόνιση εικόνων, όμως είναι ικανή για την απεικόνιση μηνυμάτων όπου αυτός είναι και ο σκοπός αυτής της Π.Ε.

4.3.1 Stockfish

Το stockfish είναι μια παντοδύναμη μηχανή για το παιχνίδι σκάκι. Είναι ένα λογισμικό ανοικτού τύπου που διατίθεται δωρεάν στο διαδίκτυο, όπου στο οποίο είναι αποθηκευμένα σκακιστικά παιχνίδια από αμνημόνευτων χρόνων με σκοπό τον υπολογισμό πιθανοτήτων και κέρδους μέσα σε μεγάλο βάθος κινήσεων [21].



Εικόνα 4.5: Περιβάλλον stockfish στην πλατφόρμα LICHESSE

Αυτό εξυπηρετεί στην μελέτη παρτίδων όταν χρησιμοποιείται σαν μια μηχανή ανάλυσης σκακιστικών παιχνιδιών αλλά και ως μηχανή-νοημοσύνη ώστε να μπορείς να παίξεις μαζί του σκάκι.

Αποτελεί ένα πολύ χρήσιμο εργαλείο της σκακιστικής κοινότητας και χρησιμοποιείται επίσης από μεγάλες εταιρίες του χώρου όπως το Chess.com και το Lichess.

4.3.2 Maxchess

Το πρόγραμμα maxchess χρησιμοποιεί την βιβλιοθήκη chessboard για να παίξει σκάκι με την μηχανή stockfish. Είναι κατασκευασμένο από πολλαπλά παραδείγματα του διαδικτύου και αποτελεί ένα ανοικτού τύπου λογισμικό. Είναι γραμμένο σε γλώσσα Python έκδοσης 2.7 .

Δέχεται τις εντολές εισόδου από μεθόδους χειριστήριου όπου στην προκειμένη περίπτωση έχουν αντικατασταθεί από τις σειριακές επικοινωνίες μεταξύ του Arduino Nano και του Raspberry pi.

4.3.3 Autostart

Το συγκεκριμένο αρχείο περιέχει ένα text το οποίο αποθηκεύεται σε φάκελο υπεύθυνο για την παραμετροποίηση του raspberry pi.

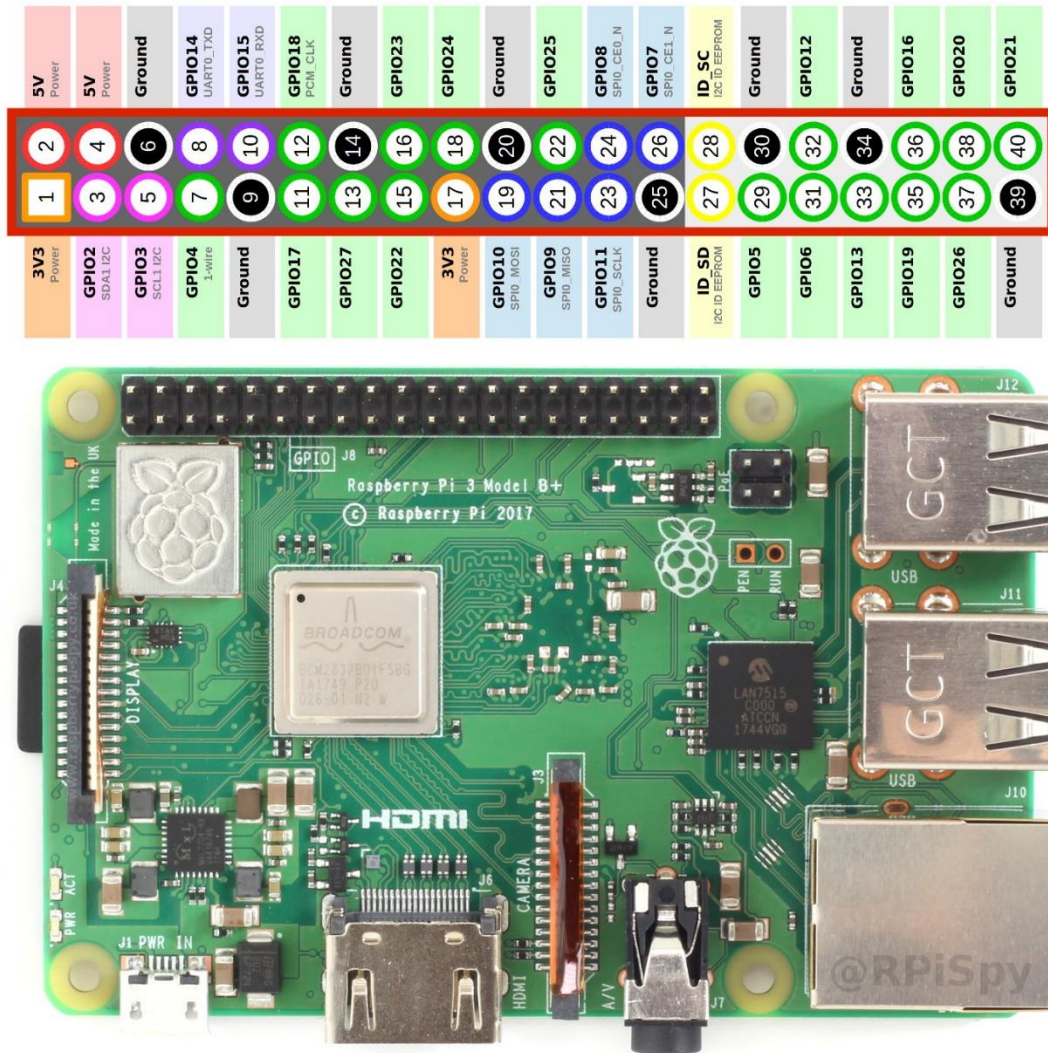
Σε αυτό το αρχείο είναι δηλωμένη μια διαδρομή στην μνήμη του raspberry pi. Σε αυτή με τη χρήση του συγκεκριμένου text δηλώνεται η γλώσσα την οποία χρησιμοποιούμε, δηλαδή python 2 και το πρόγραμμα StartChessGame, το οποίο ενεργοποιεί τα επιμέρους βασικά κομμάτια κώδικα που αναφέρθηκαν παραπάνω.

Τέλος, ενεργοποιείται μόλις ολοκληρωθεί η διαδικασία εκκίνησης του υπολογιστή και έπειτα ακολουθεί ένα εφέ φόρτωσης του παιχνιδιού στην σκακιέρα μας μέσω του Arduino nano. Πλέον χρειάζεται να ακολουθήσουμε τις οδηγίες στην οθόνη LED για να ξεκινήσουμε ένα καινούριο παιχνίδι.

4.4 Σειριακές επικοινωνίες

Ο κύριος τρόπος επικοινωνίας των εξωτερικών συσκευών που συνδέονται με το Raspberry pi γίνεται μέσω των general purpose inputs and outputs (GPIOs) που φέρει ο υπολογιστής. Μέσω αυτών ο υπολογιστής μπορεί να δέχεται εντολές εισόδου και εξόδου, ώστε να υποστηριχθούν οι επιμέρους εντολές και προγράμματα που αναφέρθηκαν από και προς τις ηλεκτρονικές συσκευές της εργασίας [22].

Στην πράξη μέσω αυτών επικοινωνούν όλες μας οι συσκευές και το μόνο που μπορεί να διαφέρει είναι τα πρωτόκολλα επικοινωνίας που χρησιμοποιούνται από αυτές.



Εικόνα 4.6: GPIOs

Για την επικοινωνία της OLED θόνης με τον υπολογιστή Raspberry pi χρησιμοποιήθηκε πρωτόκολλο επικοινωνίας I²C. Το εν λόγω πρωτόκολλο επικοινωνίας χαρακτηρίζεται από ευκολία στη χρήση χωρίς απώλειες πληροφοριών κατά την επικοινωνία. Μέσω αυτού το διάλυο επικοινωνίας μπορεί το raspberry pi να στέλνει τα μηνύματα στην οθόνη με σκοπό την εμφάνισή τους από αυτή [23].

Ουσιαστικά, επιτυγχάνουμε έτσι την ανανέωσή της με καινούρια μηνύματα όποτε αυτό χρειάζεται προς ενημέρωση του παίκτη σχετικά με πληροφορίες που αφορούν το παιχνίδι, αλλά και τις επιμέρους λειτουργίες επιλογής παιχνιδιού, δυσκολίας, παράβασης κανονισμών, επανεκκίνησης και τερματισμού λειτουργίας [22].

4.5 Επίλογος

Σε αυτό το κεφάλαιο ολοκληρώθηκε η επεξήγηση για το πως επικοινωνούν μεταξύ τους οι ηλεκτρονικές συσκευές της Π.Ε. Δόθηκαν επεξηγήσεις για τα επιμέρους προγράμματα όπου συνεργατικά δουλεύουν

Κεφάλαιο 4

στο παρασκήνιο και επιτυγχάνουν έτσι την ροή ενός σκακιστικού παιχνιδιού. Μάθαμε πληροφορίες για την σκακιστική μηχανή-νοημοσύνη, τις εφαρμογές της στην σκακιστική κοινότητα και το πως μπορούμε να την εκμεταλλευτούμε ώστε να δημιουργήσουμε ενδιαφέρον projects που σκοπό έχουν την εκμάθηση του παιχνιδιού σκάκι. Δόθηκαν πληροφορίες σχετικές με την διατησία των παιχνιδιών όπως και τα απαραίτητα εργαλεία και βιβλιοθήκες που χρησιμοποιούνται για την επίτευξη αυτού του στόχου.

Κεφάλαιο 5^ο: Συμπεράσματα και μελλοντικές βελτιώσεις

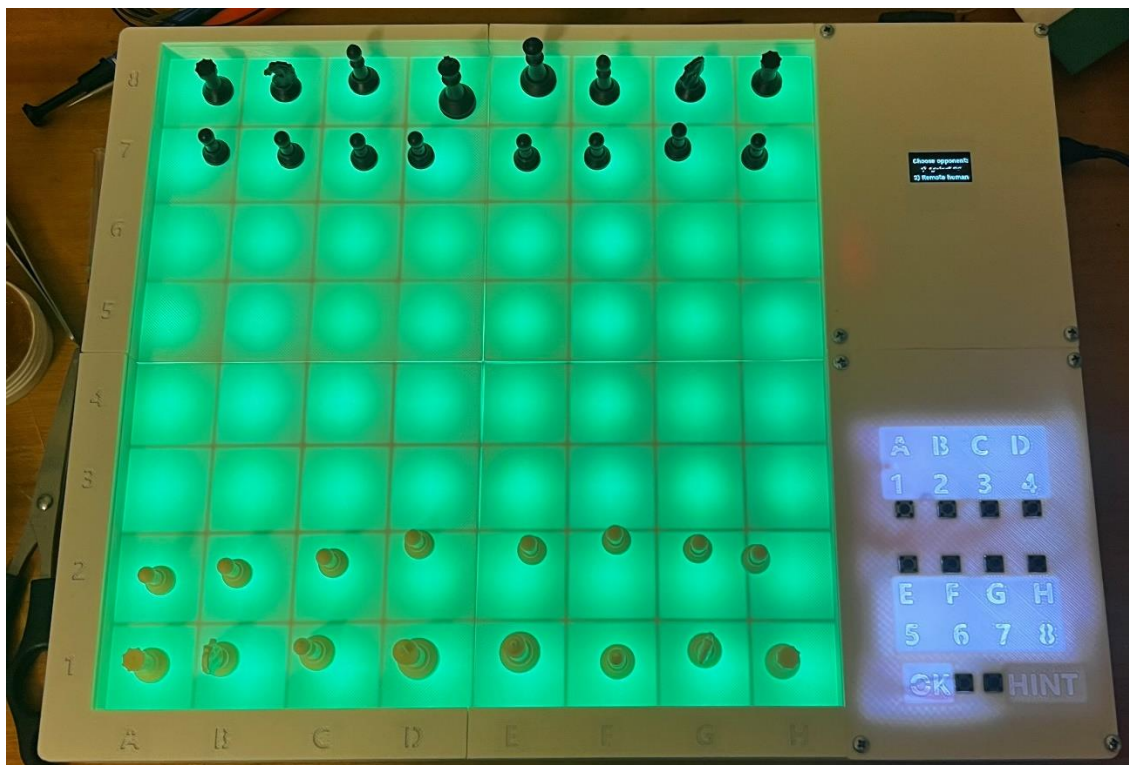
5.1 Συμπεράσματα

Η πτυχιακή εργασία είχε να επιφέρει μια μεγάλη επιμορφωτική εμπειρία κατά την έρευνα και υλοποίησή της. Ξεκίνησε από το μηδέν με βασικές γνώσεις που αποκτήθηκαν κατά την διάρκεια των σπουδών. Πάντα υπήρχε και θα υπάρχει το ερέθισμα της αυτό-βελτίωσης που θα ανοίγει νέες πόρτες και ορίζοντες για ένα ευδόκιμο επαγγελματικό μέλλον. Με την συγκεκριμένη Π.Ε. έγινε η διαπίστωση πως οι δημιουργίες μας μπορούν να βελτιώσουν ή ακόμα και να εξελίξουν συνανθρώπους μας είτε αποτελούν κομμάτι της ψυχαγωγίας είτε κάποιο εργαλείο του παραγωγικού τομέα. Προσωπικά με ενδιαφέρει ο χώρος της ψυχαγωγίας είτε αυτή προσφέρεται αυτούσια όπως είναι, είτε να επιφέρει επιμορφωτικό ενδιαφέρον για την βελτίωση ικανοτήτων και προσωπικής εξέλιξης. Η εργασία μου λοιπόν είχε έχει και θα έχει επιμορφωτικό χαρακτήρα και θα μπορεί μελλοντικά να προσαρμοστεί στις ανάγκες του εκάστοτε καταναλωτή. Καλώς ή κακώς πολλές φορές συναντάμε όλοι μας οικονομικά εμπόδια τα οποία στην προκειμένη περίοδο να μην είναι δυνατό να καταπολεμηθούν όσο αποτελεσματικά και εάν θέλουμε. Ίσως με την πάροδο του χρόνου να δημιουργηθεί το κατάλληλο κλίμα που θα μου επιτρέψει να ολοκληρώσω τις ιδέες μου όπως και τους προβληματισμούς μου επάνω σε κατασκευαστικές ιδέες που θα μου δημιουργηθούν. Αν μη τι άλλο είμαι ευγνώμων γιατί μέσα από αυτή την εργασία έμαθα πράγματα τα οποία ενδεχομένως να μην είχα το κατάλληλο ερέθισμα ώστε να κάνω την αναζήτηση και εφαρμογή αυτών. Ο στόχος μου έχει εκπληρωθεί στην δημιουργία μιας βελτιωμένης ηλεκτρονικής σκακιέρας που θα μπορεί να χρησιμοποιηθεί ως ένα εκπαιδευτικό μέσο αν φυσικά αυτό κριθεί απαραίτητο. Είναι κοινά αποδεκτό ότι, ο χώρος του σκάκι χαρακτηρίζεται από μια μακροχρόνια παράδοση που φέρει μαζί του το παιχνίδι ανά τους αιώνες και ενδεχομένως να δυσκολεύει τον κλασικό παίκτη να εκπληρώσει το επόμενο τεχνολογικό βήμα. Έτσι, προσπάθησα να σεβαστώ αυτή την παράδοση και να υλοποιήσω μια κατασκευή που ναι μεν δεν αποτελεί μια κλασική εμπειρία παιχνιδιού, αλλά μια πιο σύγχρονη έκδοση που παρομοιάζεται με μια παραδοσιακή παρτίδα.

Παρακάτω ακολουθούν μερικές εικόνες της τελικής κατασκευής που παρουσιάζουν ένα παιχνίδι, όπως και τους χρήσιμους μηχανισμούς της σκακιέρας για τον έλεγχο κινήσεων και βοήθειας.



Εικόνα 5. 1: Εκκίνηση της σκακιέρας

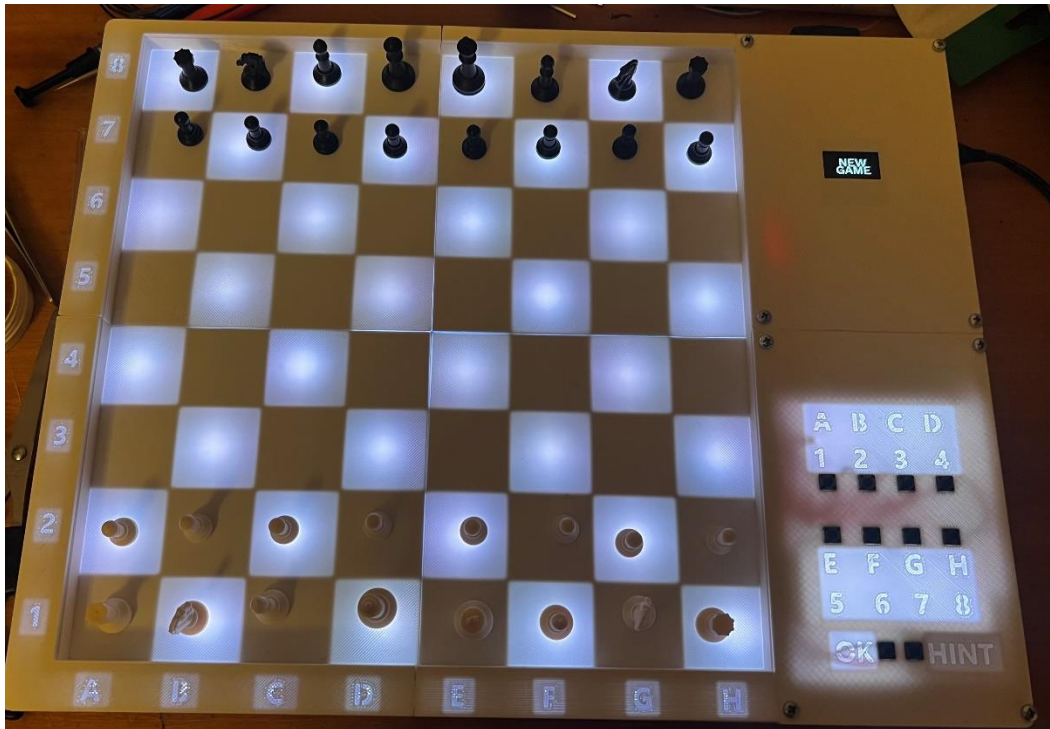


Εικόνα 5. 2: Επιλογή τύπου παιχνιδιού

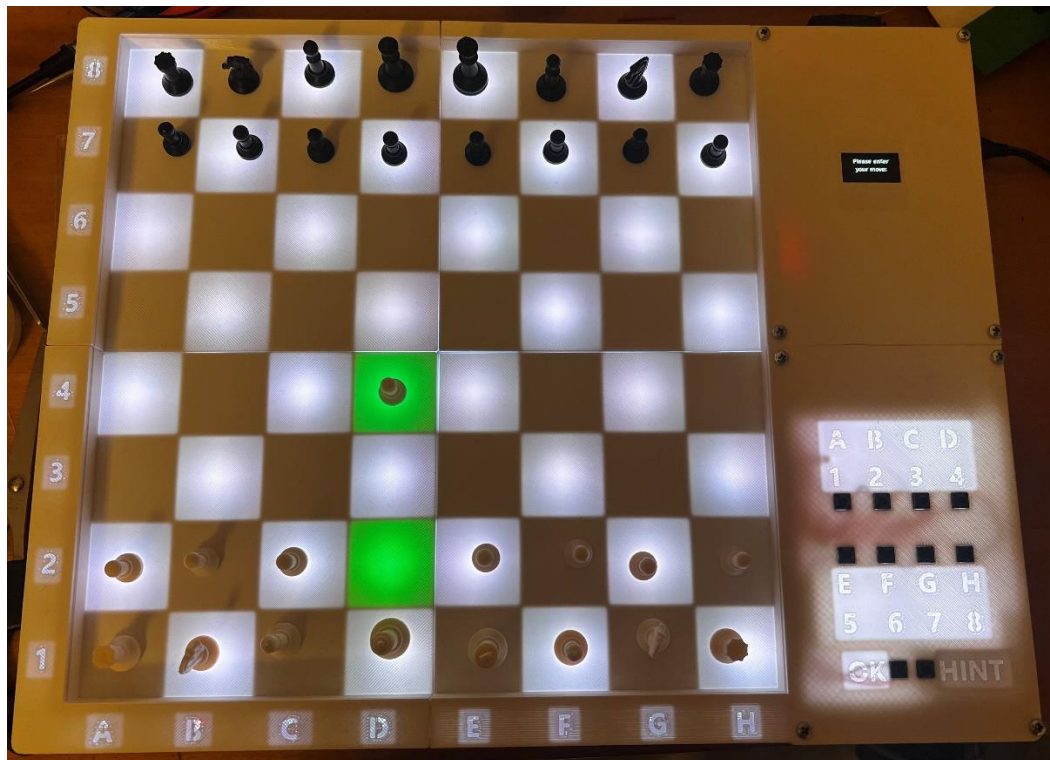


Εικόνα 5. 3: Επιλογή δυσκολίας παιχνιδιού

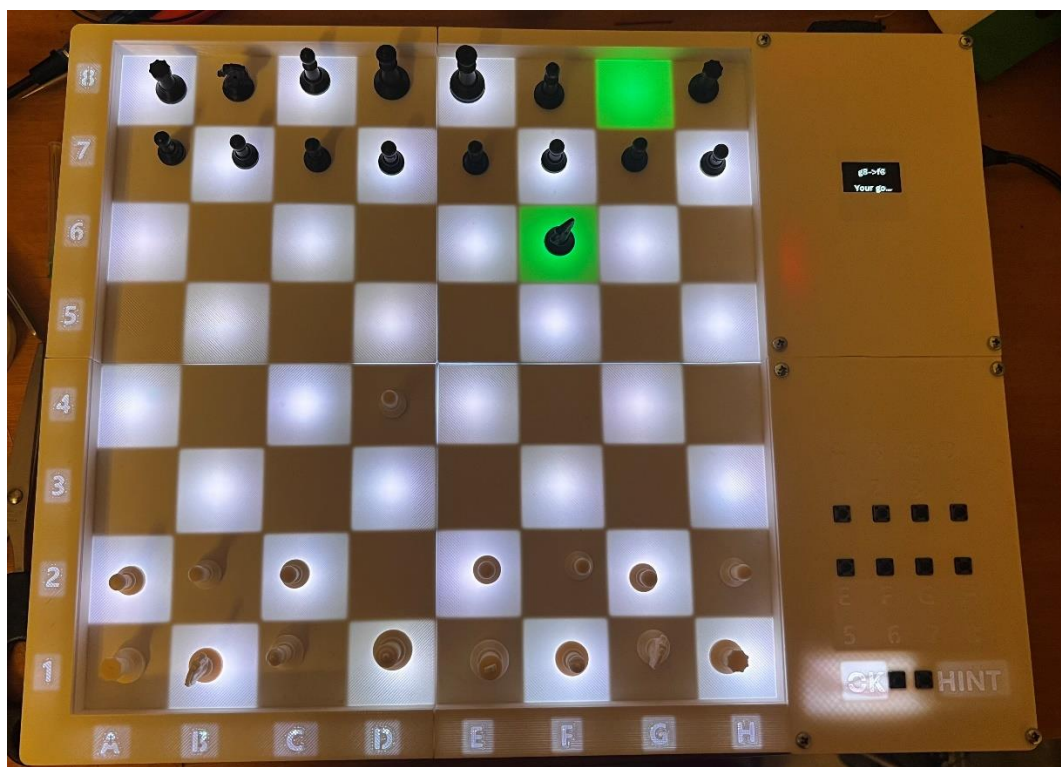
Εικόνα 5. 4: Επιλογή χρόνου κίνησης υπολογιστή



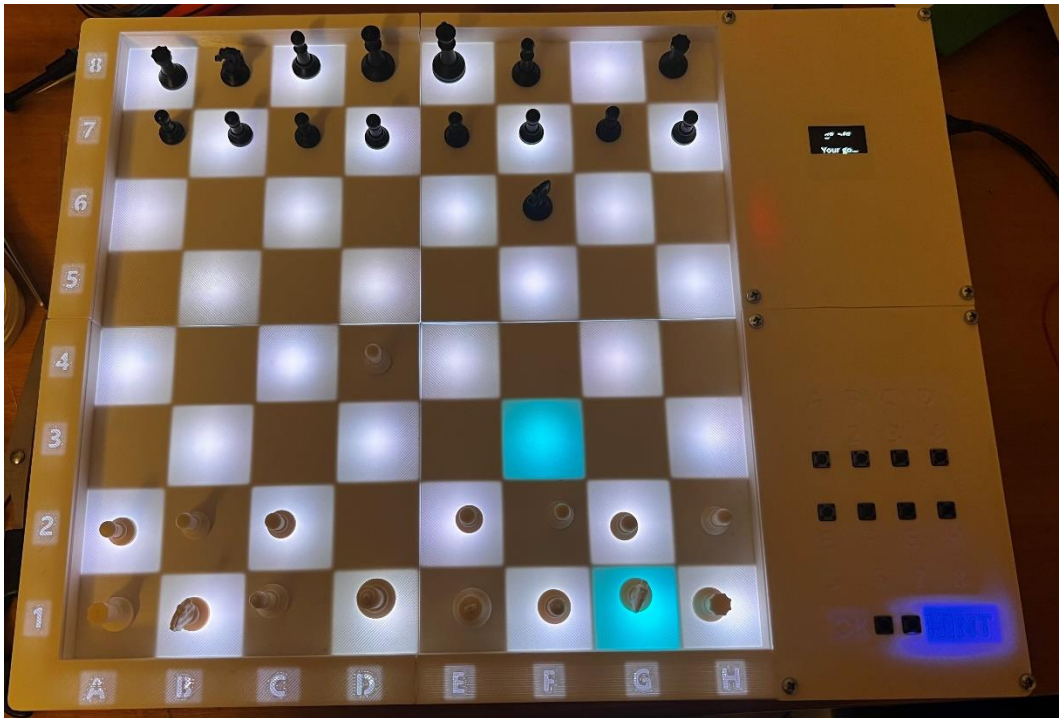
Εικόνα 5. 5: Έναρξη παιχνιδιού



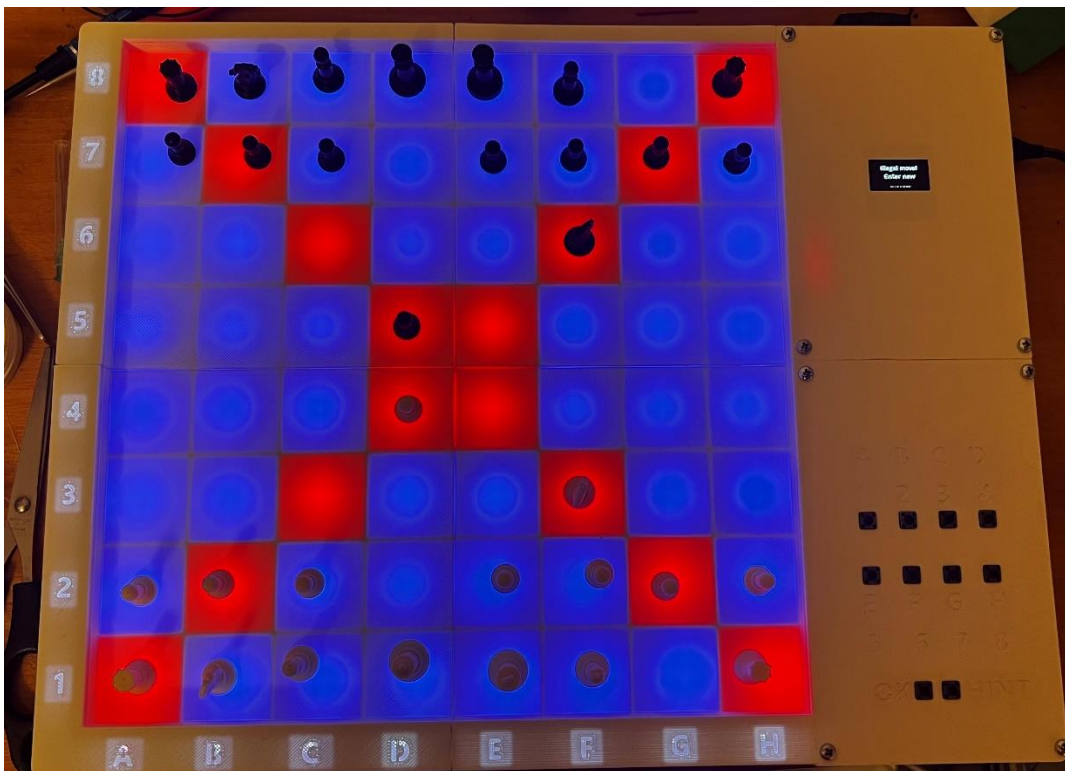
Εικόνα 5. 6: Κίνηση D2-D4



Εικόνα 5. 7: Απάντηση υπολογιστή G8-F3



Εικόνα 5. 8: Επιλογή Hint με εμφάνιση της προτεινόμενης κίνησης με μπλε χρώμα



Εικόνα 5. 9: Μη έγκυρη κίνηση από τον παίκτη

5.2 Μελλοντικές βελτιώσεις

Σίγουρα όπως πολλά πράγματα στην ζωή μας βελτιώνονται και εξελίσσονται τεχνολογικά με την πάροδο του χρόνου έτσι και η συγκεκριμένη εργασία μπορεί να βελτιωθεί. Συγκεκριμένα θα μπορούσε να υπάρξει αυτόματη ανίχνευση των κομματιών με την βοήθεια των αποκαλούμενων Hall sensors όπου λειτουργούν με το φαινόμενο Hall. Το φαινόμενο Hall είναι ένα από τα περισσότερο γνωστά και αξιοποιήσιμα φαινόμενα στερεού τύπου. Ανακαλύφθηκε από τον Hall το 1879 και γνώρισε μεγάλη αναγνωσιμότητα με την παράλληλη ανάπτυξη της τεχνολογίας των ημιαγωγών [24].

Με αυτό τον τρόπο η σκακιέρα δεν θα χρειαζόταν την δική μας είσοδο για τις πληροφορίες που λαμβάνει περί θέσης του εκάστοτε κομματιού. Βέβαια στην πράξη για να γίνει αυτό χρειάζεται η εργασία να δεχθεί μεγάλη αλλαγή στο κομμάτι σχεδίασης ώστε να μπορέσουν αυτοί οι 64 αισθητήρες να τοποθετηθούν στο εσωτερικό της και να προγραμματιστούν. Αυτό συνεπάγεται σε νέα σχεδίαση τρισδιάστατων μοντέλων και εκ νέου τις εκτυπώσεις αυτών. Ο λειτουργικός πυρήνας του λογισμικού κατά βάση θα παρέμενε ο ίδιος αφού πρώτα δηλωθεί ο τρόπος εισαγωγής των πληροφοριών της κίνησης.

Υλικά και εργαλεία που αγοράστηκαν για την Π.Ε.	Κόστος
Creality3D Ender-3 Pro	239 €
3D Prima PrimaValue PLA 1.75mm Light Grey 1kg	23,90 €
3D Prima PrimaValue PLA 1.75mm White 1kg	23,90 €
Creality 3D® CR-Touch	45,53 €
Creativity® MK8 Full Metal Hot End Kit with Extruder Heater Block	8,05 €
Capricorn- Creality Premium PTFE Bowden tube 1.75mm Filament	12 €
Ender-3 Glass Plate with Chemical Surface 235 x 235 mm	16,80 €
3d Hot Bed Springs	3 €
Creality 3D Ender-3 Pro Magnetic Build Surface	16,90 €
Bowden Extruder Feed Kit MK8 Red	9,90 €
DC Fan 5V	2,80 €
40x Jumper wires 15cm female to female all colors	7,20 €
ELEGOO Developer Board Compatible with Arduino IDE Project (3 Board Set without Cable)	27,08 €
Verbatim 44082 16GB Class 10 Micro SDHC with Adapter	7,28 €
3 x Logic Level Converter TXS0108E	7,80 €
Keenso USB Breakout Board Pack of 10 USB Type A Female Breakout Board 2.54 mm	7,02 €
AZDelivery OLED Parent 128 x 64 Pixel 0.96 Inches	16,14 €
BTF-LIGHTING WS2812B LED Strip	30,21 €
Mini Push Button Switches	5 €
Raspberry Pi model 3A+	40 €
Σύνολο	549,51 €

ΒΙΒΛΙΟΓΡΑΦΙΑ

Βιβλία

- [10] C. Schmidleithner and D. M. Kalaskar, "Stereolithography," in *3D Printing*, London, IntechOpen, 2018, pp. 1-22.

Data Sheet

- [16] Arduino, "Arduino Nano," A000005 datasheet, 06 Dec. 2022.
- [18] Raspberry Pi Foundation, "Raspberry Pi 3 Model A+," datasheet, Nov. 2018.
- [19] Texas Instruments , "8-BITBIDIRECTIONALVOLTAGE-LEVELTRANSLATORFOROPEN DRAINANDPUSH-PULLAPPLICATIONS," TXS0108E datasheet, Dec. 2007.

Internet Site

- [20] M. LeBlanc-Williams, "Installing CircuitPython Libraries on Raspberry Pi," Adafruit, 2022.

Paper in Conference Proceedings

- [1] C.-T. Hsieh, "Development of an Integrated System of 3D Printer and Laser Carving," in *International Microsystems, Packaging, Assembly and Circuits Technology conference*, Taipei, Taiwan, 2016.
- [4] S. Moore, P. Armstrong, T. McDonald and M. Yampolskiy, "Vulnerability Analysis of Desktop 3D Printer," in *2016 Resilience Week*, Chicago, IL, USA, 2016.
- [6] A. C. Brown and D. d. Beer, "Development of a stereolithography (STL) slicing and G-Code generation algorithm for an entry level 3-D printer.," in *AFRICON Conference*, Arusha, Tanzania, 2013.
- [11] M. Schmid, A. Amado and K. Wegener, "Polymer powders for selective laser sintering (SLS)," in *AIP Conference Proceedings 1664*, Zürich, Switzerland, 2015.
- [12] M. Hasan, R. Khan, A. T. Noman, H. Rashid, N. Ahmed and T. Reza, "Design and Implementation of a Microcontroller Based Low Cost Computer Numerical Control (CNC) Plotter using Motor Driver Controller," in *International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Cox's Bazar, Bangladesh, 2019.
- [17] A. D. Deshmukh and U. B. Shinde, "A low cost environment monitoring system using raspberry Pi and arduino with Zigbee," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, India, 2017.
- [22] M. Maksimović, V. Vujović, N. Davidović, V. Milošević and B. Perišić, "Raspberry Pi as Internet of Things hardware: Performances and Constraints," in *Proceedings of 1st International Conference on Electrical, Electronic and Computing Engineering*, Vrnjačka Banja, Serbia, 2014.

Journal Articles

- [2] C. Schubert, M. C. van Langeveld and L. A. Donoso, "Innovations in 3D printing: a 3D overview from optics to organs," *British Journal of Ophthalmology*, vol. 98, pp. 159-161, 11 January 2014.
- [3] A. Khazanchi, A. Kanwar, L. SALUJA, A. DAMARA and V. DAMARA, "OLED: A New Display Technology," *International Journal Of Engineering And Computer Science*, vol. 1, no. 2, pp. 75-84, 11 2012.
- [5] C. H. Lee, F. N. B. M. Padzil, S. H. Lee, Z. M. A. Ainun and L. C. Abdullah, "Potential for Natural Fiber Reinforcement in PLA Polymer Filaments for Fused Deposition Modeling (FDM) Additive Manufacturing: A Review," *Polymers*, pp. 1-18, 27 April 2021.
- [7] B. M. Schmitt, C. F. Zirbes, C. Bonin, D. Lohmann, D. C. Lencina and A. d. C. S. Netto, "A Comparative Study of Cartesian and Delta 3D Printers on Producing PLA Parts," *Materials Research*, pp. 883-886, 05 February 2018.
- [8] H. Makino, "Development of the SCARA," *Journal of Robotics and Mechatronics*, vol. 26, no. 1, pp. 5-8, 20 February 2014.
- [9] A. Petsiuk, B. Lavu, R. Dick and J. M. Pearce, "Waste Plastic Direct Extrusion Hangprinter," *Inventions*, vol. 7, no. 3, 19 August 2022.
- [13] Z. Liu, Y. Wang, B. Wu, C. Cui, Y. Guo and C. Yan, "A critical review of fused deposition modeling 3D printing technology in manufacturing polylactic acid parts," *The International Journal of Advanced Manufacturing Technology*, p. 2877–2889, 7 February 2019.
- [14] N.-D. CIOBOTA, "STANDARD TESSELLATION LANGUAGE IN RAPID PROTOTYPING TECHNOLOGY," *The Scientific Bulletin of VALAHIA University - Materials and Mechanics*, 2012.
- [15] M. C. Bourget, "An Introduction to Light-emitting Diodes," *HORTSCIENCE*, vol. 43, no. 7, pp. 1944-1946, December 2008.
- [21] S. Maharaz, N. Polson and A. Turk, "Chess AI: Competing Paradigms for Machine Intelligence," *Entropy*, vol. 24, no. 550, 14 April 2022.
- [23] J. Mankar, C. Darode, K. Trivedi, M. Kanoje and P. Shahare, "REVIEW OF I2C PROTOCOL," *International Journal of Research in Advent Technology*, vol. 2, no. 1, pp. 474-479, January 2014.
- [24] R. S. Popović, "Hall-effect devices," *Sensors and Actuators*, vol. 17, no. 1-2, pp. 39-53, 3 May 1989.