



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**"Υλοποίηση εκτυπωτή διάστασης A2, με μικροελεγκτή
Arduino και χρήση βηματικών κινητήρων "**

Νταλούκας Χρήστος 2088 - Φυκούρας Ηλίας 7242

ΕΠΙΒΛΕΠΩΝ: Καρέλλης Δημήτριος, Επίκουρος καθηγητής

ΠΑΤΡΑ 2021

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Πάτρα, 23/9/2019, αρ. πτυχιακής: 1747

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Καρέλλης Δημήτριος
2. Βλαχόπουλος Πέτρος,
3. Σταθάτος Ηλίας

Υπεύθυνη Δήλωση Φοιτητών

Βεβαιώνουμε ότι είμαστε συγγραφείς αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχαμε για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης έχουμε αναφέρει τις όποιες πηγές από τις οποίες κάναμε χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνουμε ότι αυτή η εργασία προετοιμάστηκε από εμάς προσωπικά ειδικά για τη συγκεκριμένη εργασία.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Πελοποννήσου δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων των συγγραφέων εκ μέρους του Τμήματος.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία των φοιτητών Νταλούκα Χρήστου και Φυκούρα Ηλία που την εκπόνησαν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης οι συγγραφείς /δημιουργοί εκχωρούν στο Πανεπιστήμιο Πελοποννήσου, την αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση των συγγραφέων /δημιουργών. Οι συγγραφείς /δημιουργοί διατηρούν το σύνολο των ηθικών και περιουσιακών τους δικαιωμάτων.

Περίληψη

Στα πλαίσια της παραπάνω διπλωματικής εργασίας, αναζητήθηκε μια προσιτή και οικονομική λύση, για την δημιουργία ενός εκτυπωτή δύο διαστάσεων μεγέθους A2, με χρήση μικροελεγκτή Arduino.

Ο έλεγχος της κίνησης του συστήματος σχεδιάσης στις δύο διαστάσεις, γίνεται με χρήση κυλινδρικών μεταλλικών αξόνων. Η κίνηση των αξόνων πετυχαίνεται με χρήση ταινίας (ιμάντων), οι οποίοι ελέγχονται με βηματικούς κινητήρες.

Υπάρχει η δυνατότητα έγχρωμης, εκτύπωσης με τοποθέτηση χρωματιστών μολυβιών και διαφορετικής ποιότητας γραμμών, ανάλογα με την εκτυπωτική γραφίδα που χρησιμοποιούμε.

Η όλη κατασκευή είναι φορητή, μικρού βάρους και τοποθετήθηκε πάνω σε αλουμινένιο σασί, το οποίο με τη σειρά του τοποθετείται σε σταθερή λεία επιφάνεια, π.χ. γραφείο, στην οποία προσαρμόζεται το προς εκτύπωση χαρτί.

Ο προγραμματισμός του μικροελεγκτή Arduino, έγινε σε γλώσσα προγραμματισμού C, στο περιβάλλον ανάπτυξης λογισμικού “Arduino IDE”, που είναι το “official” περιβάλλον για εφαρμογές Arduino.

Για τον προσδιορισμό της ακριβούς θέσης εκτύπωσης έγινε χρήση της computer numerical control (CNC) προσέγγισης, δηλαδή της αντιστοίχισης κάθε σημείου της επιφάνειας σε δύο αριθμούς που είναι οι συντεταγμένες X και Y, ως προς ένα σημείο που ορίζουμε ως αρχή των αξόνων. Αυτό επετεύχθει με πρόγραμμα που δημιουργήσαμε στο περιβάλλον της Visual Basic, μέσα από το οποίο, η κάθε γραμμή της εικόνας, αναλύεται στις συντεταγμένες των σημείων της.

Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε τον καθηγητή μας και επιβλέποντα της διπλωματικής, κο. Καρέλλη Δημήτριο, για την ανοιχτή και θερμή προσέγγιση αρχικά τις ιδέας μας, αλλά και την άριστη συνεργασία στην συνέχεια, κατά την εκπόνηση της παρούσας εργασίας.

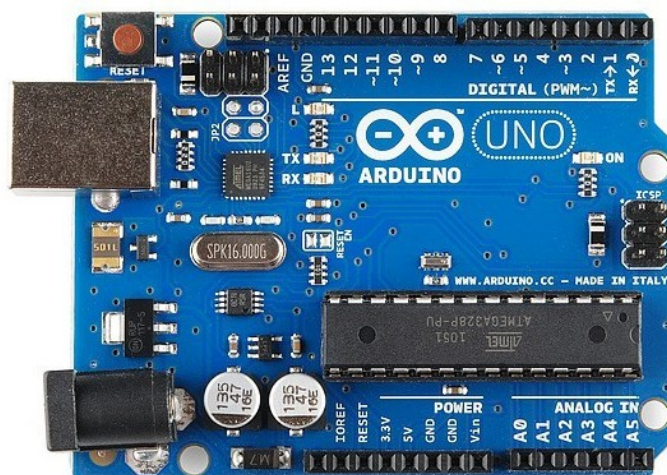
Επίσης οφειλή και στις οικογένειές μας, που ακαταπρόλητα, ηθικά και υλικά στήριξαν την ενασχόλησή μας.

Πίνακας περιεχομένων

1 Βασικά Στοιχεία του Arduino.....	6
1.1 Μικροελεγκτής του Arduino.....	7
1.2 Είσοδοι & Έξοδοι.....	8
1.3 Τροφοδοσία.....	11
1.4 Είδη Arduino.....	11
1.5 Ηλεκτρονικά εξαρτήματα που συνδέονται στο Arduino.....	13
2 Βηματικοί κινητήρες (stepper motors).....	17
2.1 Γενικές πληροφορίες. Τι είναι ο βηματικός κινητήρας.....	17
2.2 Δομή ενός βηματικού κινητήρα.....	19
2.3 Αρχή λειτουργίας βηματικού κινητήρα.....	20
2.4 Χαρακτηριστικά κινητήρα NEMA-17 size - 200 steps/rev, 12V 350mA.....	22
2.5 Κύκλωμα οδηγού κινητήρα.....	23
3 Περιβάλλον Προγραμματισμού του Arduino.....	25
3.1 Εγκατάσταση λογισμικού Arduino.....	25
3.2 Άλλα περιβάλλοντα προγραμματισμού.....	27
4 Περιγραφή κατασκευής.....	30
4.1 Υλικά.....	30
4.2 Υλοποίηση κατασκευής.....	30
4.3 Δοκιμές – έλεγχοι κατασκευής.....	31
5 Λειτουργία εκτυπωτή.....	34
5.1 Περιγραφή λειτουργίας.....	34
5.2 Πρόσθετες λειτουργίες.....	38
6 Συμπεράσματα και προτάσεις περαιτέρω μελέτης ή ανάπτυξης.....	42
7 Βιβλιογραφία.....	43
Παράρτημα Α: Διευθύνσεις internet.....	44
Παράρτημα Β : Κώδικας Συστήματος και προγραμμάτων εγκατάστασης.....	45
Β1 Κώδικας VB.NET.....	45
Β2 Κώδικας Arduino.....	75

1 Βασικά Στοιχεία του Arduino

Το Arduino είναι μια «ανοικτού κώδικα» πλατφόρμα «πρωτοτυποποίησης» συστήματος ελέγχου ηλεκτρονικών, βασισμένη σε ευέλικτο και εύκολο στη χρήση υλικό και λογισμικό. Η πλατφόρμα απευθύνεται σε οποιονδήποτε έχει λίγη προγραμματιστική εμπειρία, στοιχειώδεις γνώσεις ηλεκτρονικών και ενδιαφέρεται να δημιουργήσει διαδραστικά συστήματα ή περιβάλλοντα ελέγχου και αντίδρασης σε εξωτερικά ερεθίσματα ή συνθήκες, αλλά και σε αυτούς που έχουν τη διάθεση να φτιάξουν τις δικές τους κατασκευές από το μηδέν.



Εικόνα 1: Πλακέτα Arduino Uno

Το Arduino ουσιαστικά πρόκειται για μία ηλεκτρονική πλακέτα που βασίζεται στον μικροελεγκτή ATmega της Atmel και του οποίου το λογισμικό αλλά κι όλα τα σχέδια που χρειάζεται για την λειτουργία του, διανέμονται ελεύθερα και δωρεάν ώστε να δίνεται η δυνατότητα να κατασκευαστεί από τον καθένα. Πάνω του μπορούμε να συνδέουμε πολλές μονάδες εισόδου (π.χ. αισθητήρες, επαφές) και εξόδου (π.χ. Led, μοτέρ), να επεξεργαζόμαστε τα δεδομένα τους και να στέλνονται οι κατάλληλες οδηγίες στις εξόδους, ώστε να εφαρμόζεται μία αντίδραση στο ερέθισμα. Μπορούμε να πούμε ότι το Arduino λειτουργεί σαν υπολογιστής ή καλύτερα σαν ελεγκτής καταστάσεων δράσης και αντίδρασης.

Η γλώσσα προγραμματισμού που χρησιμοποιεί είναι η Wiring, η οποία είναι αρκετά εύκολη στη σύνταξη και διατίθεται σε πλατφόρμες Linux, MAC και Windows με αδεια χρήσης GPL.

Αυτό όμως που κάνει το Arduino ακόμα πιο σημαντικό, είναι ότι όλο το κύκλωμα της πλακέτας διατίθεται με άδεια χρήσης Creative Commons, πράγμα που σημαίνει ότι ο καθένας μπορεί να κατασκευάσει την δική του πλακέτα όπως αυτός θέλει.

Φυσικά για τους αρχάριους το να κατασκευάσουν την πλακέτα μόνοι τους, ίσως ακουστεί λίγο τραβηγμένο, οπότε είναι ευκολότερο να αγοράσουν μια έτοιμη πλακέτα Arduino από το διαδίκτυο, η οποία διατίθεται σε πάρα πολύ προσιτή τιμή.

Αν και μικροσκοπικό (7x5 cm) οι δυνατότητες που προσφέρει είναι πάρα πολλές. Μπορούμε να το χρησιμοποιήσουμε σε εφαρμογές ρομποτικής και γενικότερα σε αυτοματισμούς καταφέροντας έτσι πάρα πολλά όπως: την κίνηση servo, stepper και DC κινητήρων, τη λήψη πληροφοριών από διάφορους αισθητήρες (θερμοκρασίας, υγρασίας, υπερύθρων κ.α), την αμφίδρομη σειριακή επικοινωνία μεταξύ Arduino και PC χρησιμοποιώντας γλώσσες προγραμματισμού (όπως Java και python), όπως επίσης την αναπαραγωγή και αντίληψη ήχων.

Φυσικά οι δυνατότητες του Arduino δεν σταματούν εκεί, στο site του Arduino (<http://arduino.cc/>) υπάρχει μια μεγάλη κοινότητα, με αρκετές πληροφορίες όσον αφορά τις εκδόσεις, την αγορά και τον προγραμματισμό της πλακέτας.

1.1 Μικροελεγκτής του Arduino

Το Arduino βασίζεται κυρίως σε επεξεργαστές (CPU) ATMEL όπως π.χ. ο ATmega328, που είναι ένας 8-bit RISC μικροελεγκτής, που έχει ρολόι που «τρέχει» στα 16MHz. Οι επεξεργαστές διαθέτουν ενσωματωμένη μνήμη τριών τύπων και ο ATmega328 έχει:

- ✓ 2Kb μνήμης SRAM που είναι η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματά μας ώστε να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. κατά την διάρκεια που «τρέχει» το πρόγραμμα (runtime). Αυτή η μνήμη όμως, χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή αν γίνει reset, όπως άλλωστε συμβαίνει και σε έναν υπολογιστή.

- ✓ 1Kb μνήμης EEPROM η οποία μπορεί να χρησιμοποιηθεί για απευθείας εγγραφή/ανάγνωση δεδομένων (χωρίς datatype) ανά byte από τα προγράμματά μας κατά το runtime. Σε αντίθεση με την SRAM, η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset, έτσι μπορούμε να το παρομοιάσουμε με το σκληρό δίσκο ενός υπολογιστή.

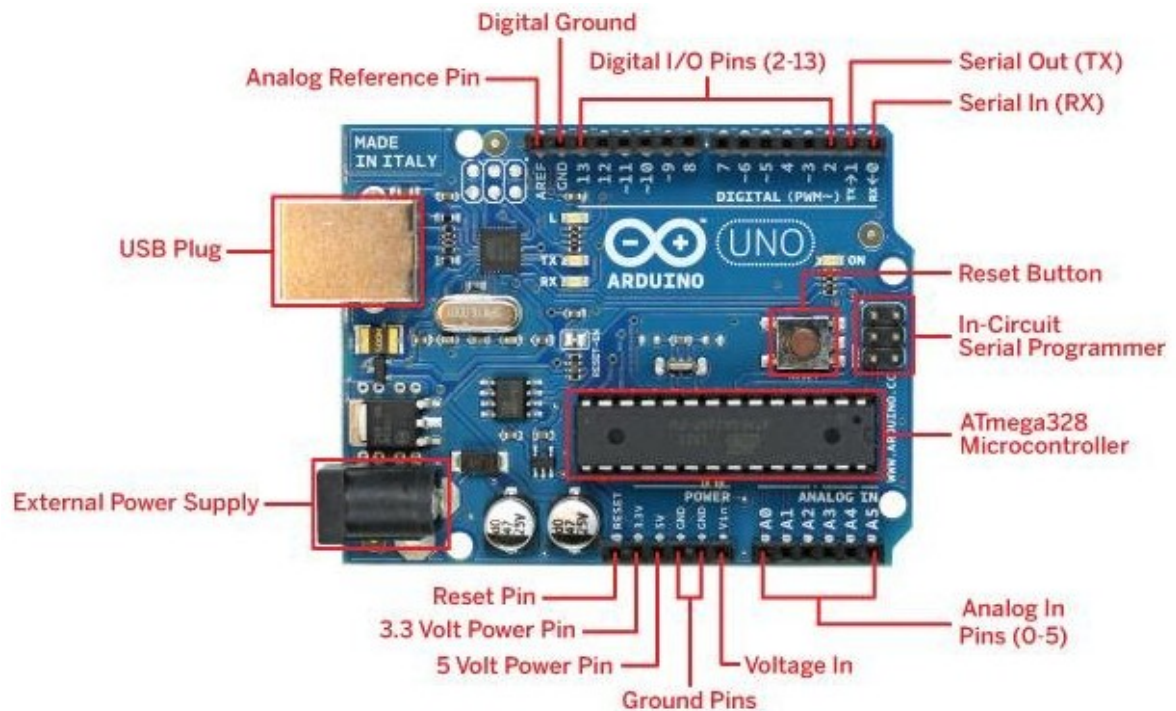
✓ 32Kb μνήμης Flash, από τα οποία τα 2Kb χρησιμοποιούνται από το λογισμικό συστήματος (firmware) του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware, αυτό που στη «γλώσσα» του Arduino ονομάζεται bootloader είναι αναγκαίο για την εγκατάσταση των δικών μας προγραμμάτων στον μικροελεγκτή μέσω της θύρας USB, χωρίς δηλαδή να χρειάζεται εξωτερικός hardware programmer ενώ, τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για να αποθηκεύονται αυτά ακριβώς τα προγράμματα, αφού πρώτα μεταγλωττιστούν στον υπολογιστή. Σημειώνεται πως ούτε η μνήμη Flash χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset.

Υπάρχουν πολλών ειδών πλακέτες (boards) Arduino, (12 βασικές παραλλαγές), οι οποίες αναφέρονται σε διαφορετικές χρήσεις η κάθε μια, ανάλογα με τις ανάγκες της εφαρμογής μας και με διαφορετικούς μικροεπεξεργαστές (CPUs) όπως ATmega168P, ATmega328P, ATmega2560, ATmega32u4, ATSAM3X8E και πολλά άλλα. Οι διαφορές τους είναι στην ταχύτητα της CPU (8MHz-84MHz), στις αναλογικές εισόδους (4 ως 16) και εξόδους (0 ως 2), τις ψηφιακές εισόδους (14 ως 54) και εξόδους (4 ως 15), το μέγεθος των τριών διαφορετικών ειδών της μνήμης, την τάση τροφοδοσίας και λειτουργίας, κλπ.

1.2 Είσοδοι & Έξοδοι

Συνήθως τα περισσότερα Arduino διαθέτουν σειριακό interface. Ο μικροελεγκτής ATmega υποστηρίζει σειριακή επικοινωνία, την οποία το Arduino προωθεί μέσα από έναν ελεγκτή Serial-over-USB ώστε να συνδέεται με τον υπολογιστή μέσω USB. Αυτή η σύνδεση χρησιμοποιείται με σκοπό τη μεταφορά των προγραμμάτων (upload) που γράφονται από τον υπολογιστή στο Arduino αλλά και για αμφίδρομη επικοινωνία Arduino και υπολογιστή μέσα από το πρόγραμμα, όσο αυτό εκτελείται.

Αναλυτικότερα, στην αριστερή πλευρά του Arduino (βλ. Εικόνα 2) είναι τοποθετημένα 14 θηλυκές θύρες (pins), αριθμημένα από 0 ως 13, που μπορούν να λειτουργούν ως ψηφιακές εισόδους και εξόδους (Input/Output-IO), για τη συγκεκριμένη πλακέτα στα 5V και καθένα μπορεί να παρέχει ή να δεχτεί το πολύ 40mA ρεύμα.



Εικόνα 2: Περιγραφή της πλακέτας του ArduinoUno

Αν είναι ψηφιακή έξοδος, το pin μπορεί να τεθεί από το πρόγραμμά μας σε κατάσταση HIGH ή LOW, οπότε το Arduino θα ξέρει αν πρέπει να διοχετεύσει ή όχι το ρεύμα λειτουργίας του στο συγκεκριμένο pin. Συνεπώς, μπορούμε να ανάψουμε και να σβήσουμε ένα LED συνδεδεμένο στο συγκεκριμένο pin. Αντιστοίχως, εάν ρυθμιστεί ένα από αυτά τα pin ως ψηφιακή είσοδος, τότε, μπορούμε με την κατάλληλη εντολή να διαβάσουμε την κατάστασή του (HIGH ή LOW). Μερικά από αυτά τα 14 pin, εκτός από ψηφιακές είσοδοι/έξοδοι έχουν και δεύτερη λειτουργία. Συγκεκριμένα:

Τα pin 0 και 1 λειτουργούν ως υποδοχή RX και αποστολή TX της σειριακής όταν το πρόγραμμά μας ενεργοποιεί την σειριακή θύρα. Συνεπώς, όταν το πρόγραμμα στέλνει δεδομένα στην σειριακή, αυτά προωθούνται και στην θύρα USB μέσω του ελεγκτή Serial-Over-USB αλλά και στο pin 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή (π.χ. ένα δεύτερο Arduino στο δικό του pin 1).

Τα pin 2 και 3 λειτουργούν και ως εξωτερικά interrupts (interrupt 0 και 1 αντίστοιχα). Αυτό σημαίνει πως μπορούμε να τα ρυθμίσουμε μέσω του προγράμματος με σκοπό να λειτουργούν αποκλειστικά ως ψηφιακές είσοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική

ροή του προγράμματος σταματάει άμεσα κι εκτελείται μια συγκεκριμένη συνάρτηση. (Αξίζει να σημειωθεί πως είναι πολύ χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.)

Τα pin 3, 5, 6, 9, 10 και 11 μπορούν να λειτουργήσουν και ως ψευδοαναλογικές εξόδους με το σύστημα PWM (Pulse Width Modulation), δηλαδή με το ίδιο σύστημα που διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων. Συνεπώς, μπορούμε να συνδέσουμε παραδείγματος χάρη ένα LED σε κάποιο από αυτά τα pin και να ελέγχουμε πλήρως την φωτεινότητά του με ανάλυση 8bit (256 καταστάσεις από 0-σβηστό ως 255-πλήρως αναμμένο) αντί να μπορούμε απλά να το αναβοσβήνουμε όπως συμβαίνει με τις υπόλοιπες ψηφιακές εξόδους.

Στην δεξιά πλευρά του Arduino (Εικόνα 2), με τη σήμανση ANALOG IN, βρίσκεται ακόμη μια σειρά από 6 pin, αριθμημένα από το 0 ως το 5. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος κάνοντας χρήση του ADC (Analog to Digital Converter) που είναι ενσωματωμένο στον μικροελεγκτή. Παραδείγματος χάρη, μπορούμε να τροφοδοτήσουμε ένα από αυτά με μια τάση η οποία μπορεί να κυμαίνεται από 0V ως μια τάση αναφοράς V_I η οποία, αν δεν κάνουμε κάποια αλλαγή είναι προ ρυθμισμένη στην τάση λειτουργίας της πλακέτας, εδώ στα 5V, με τη χρήση π.χ. ενός ποτενσιόμετρου. Τότε, μέσα από το πρόγραμμά μας μπορούμε να "διαβάσουμε" την τιμή του pin ως ένα ακέραιο αριθμό ανάλυσης 10bit, από 0 (όταν η τάση στο pin είναι 0V) μέχρι 1023 (όταν η τάση στο pin είναι 5V). Η τάση αναφοράς μπορεί να ρυθμιστεί με μια εντολή στο 1.1V, ή σε όποια τάση επιθυμούμε (μεταξύ 2 και 5V) τροφοδοτώντας εξωτερικά με αυτή την τάση το pin με την σήμανση AREF που βρίσκεται στην απέναντι πλευρά της πλακέτας. Έτσι, αν τροφοδοτήσουμε το pin AREF με 3.3V και στην συνέχεια δοκιμάσετε να διαβάσουμε κάποιο pin αναλογικής εισόδου στο οποίο εφαρμόζετε τάση 1.65V, το Arduino θα μας επιστρέψει την τιμή 512. Τέλος, καθένα από τα 6 αυτά pin, με κατάλληλη εντολή μέσα από το πρόγραμμα μπορεί να μετατραπεί σε ψηφιακό pin εισόδου/εξόδου όπως τα 14 που βρίσκονται στην απέναντι πλευρά και τα οποία περιγράφηκαν πριν. Σε αυτή την περίπτωση τα Pin μετονομάζονται από 0~5 σε 14~19 αντίστοιχα.

1.3 Τροφοδοσία


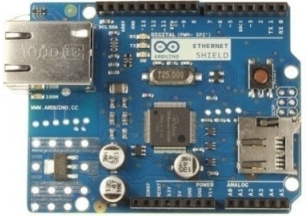
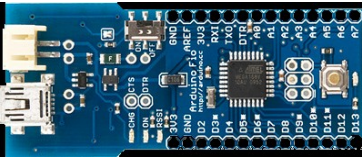
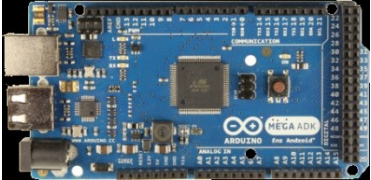
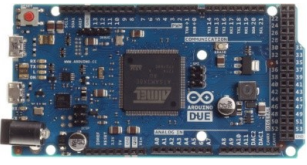
Στην πλειοψηφία τους, οι πλακέτες Arduino μπορούν να τροφοδοτηθούν από μπαταρία ή εξωτερικό τροφοδοτικό, ακόμα κι από το ίδιο USB που χρησιμοποιούμε για να μεταφέρουμε το πρόγραμμα. Η τάση που μπορούμε να εφαρμόσουμε στο Arduino είναι από 5 (ή 3.3 για κάποια boards) έως 12V και μπορεί να προέρχεται από ένα κοινό μετασχηματιστή του εμπορίου, από μπαταρίες ή οποιαδήποτε άλλη πηγή DC, χωρίς να αντιμετωπίζουμε κίνδυνο να καεί η πλακέτα.

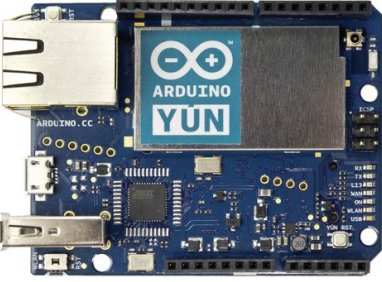

Δίπλα από τα pin αναλογικής εισόδου, υπάρχει μια ακόμα συστοιχία από 6 pin με την σήμανση POWER. Η λειτουργία του καθενός έχει ως εξής:

- Το πρώτο, με την ένδειξη RESET, όταν γειωθεί (σε οποιοδήποτε από τα 3 pin με την ένδειξη GND που υπάρχουν στο Arduino) έχει ως αποτέλεσμα την επανεκκίνηση του Arduino.
- Το δεύτερο, με την ένδειξη 3.3V, μπορεί να τροφοδοτήσει τα εξαρτήματά μας με τάση 3.3V. Η τάση αυτή δεν προέρχεται από την εξωτερική τροφοδοσία αλλά παράγεται από τον ελεγκτή Serial_over_USB και έτσι η μέγιστη ένταση που μπορεί να παρέχει είναι μόλις 50mA.
- Το τρίτο, με την ένδειξη 5V, μπορεί να τροφοδοτήσει τα εξαρτήματά μας με τάση 5V. Ανάλογα με τον τρόπο τροφοδοσίας του ίδιου του Arduino, η τάση αυτή προέρχεται είτε άμεσα από την θύρα USB (που ούτως ή άλλως λειτουργεί στα 5V), είτε από την εξωτερική τροφοδοσία αφού αυτή περάσει από ένα ρυθμιστή τάσης για να την “φέρει” στα 5V.
- Το τέταρτο και το πέμπτο pin, με την ένδειξη GND, είναι φυσικά γειώσεις.
- Το έκτο και τελευταίο pin, με την ένδειξη Vin έχει διπλό ρόλο. Σε συνδυασμό με το pin γείωσης δίπλα του, μπορεί να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας του Arduino, στην περίπτωση που δεν μας βολεύει να χρησιμοποιήσουμε την υποδοχή του φιν των 2.1mm. Αν όμως έχουμε ήδη συνδεδεμένη εξωτερική τροφοδοσία μέσω του φιν, μπορούμε να χρησιμοποιήσουμε αυτό το pin για να τροφοδοτήσουμε εξαρτήματα με την πλήρη τάση της εξωτερικής τροφοδοσίας (7~12V), πριν αυτή περάσει από τον ρυθμιστή τάσης όπως γίνεται με το pin των 5V.

1.4 Είδη Arduino


Στον πίνακα που ακολουθεί παρουσιάζονται ορισμένα είδη Arduino που υπάρχουν, με τα βασικά τεχνικά χαρακτηριστικά τους. Στην παρούσα διπλωματική, επιλέχθηκε για την υλοποίηση της κατασκευής, το Arduino UNO R3.



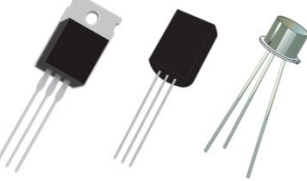



Είδος Arduino	Τεχνικά Χαρακτηριστικά
 <p data-bbox="421 636 644 663">Εικόνα 3: Arduino Uno</p>	<p data-bbox="863 371 1145 398">Μικροελεγκτής: ATmega328</p> <p data-bbox="863 403 1078 430">Τάση λειτουργίας: 5V</p> <p data-bbox="863 434 1358 461">Ψηφιακές εισοδοί / έξοδοι: 14 (6 με PWM έξοδο)</p> <p data-bbox="863 465 1238 492">Ρεύμα DC ανά είσοδο / έξοδο: 40 mA</p> <p data-bbox="863 497 1082 524">Αναλογικές εισοδοί: 6</p> <p data-bbox="863 528 1193 555">Μνήμη Flash: 32 KB (ATmega328)</p> <p data-bbox="863 560 1114 586">SRAM: 2 KB (ATmega328)</p> <p data-bbox="863 591 1142 618">EEPROM: 1 KB (ATmega328)</p> <p data-bbox="863 622 1145 649">Ταχύτητα ρολογιού: 16 MHz</p>
 <p data-bbox="405 958 660 985">Εικόνα 4: ArduinoEthernet</p>	<p data-bbox="863 696 1145 723">Μικροελεγκτής: ATmega328</p> <p data-bbox="863 728 1078 754">Τάση λειτουργίας: 5V</p> <p data-bbox="847 759 1358 808">Ψηφιακές εισοδοί / έξοδοι: 14 (4 παρέχουν PWM έξοδο)</p> <p data-bbox="863 813 1238 840">Ρεύμα DC ανά είσοδο / έξοδο: 40 mA</p> <p data-bbox="863 844 1082 871">Αναλογικές εισοδοί: 6</p> <p data-bbox="863 875 1193 902">Μνήμη Flash: 32 KB (ATmega328)</p> <p data-bbox="863 907 1114 934">SRAM: 2 KB (ATmega328)</p> <p data-bbox="863 938 1142 965">EEPROM: 1 KB (ATmega328)</p> <p data-bbox="863 969 1145 996">Ταχύτητα ρολογιού: 16 MHz</p> <p data-bbox="863 1001 1310 1028">W5100 TCP/IP Embedded Ethernet Controller</p>
 <p data-bbox="427 1330 635 1357">Εικόνα 5: Arduino Fio</p>	<p data-bbox="863 1050 1158 1077">Μικροελεγκτής: ATmega328P</p> <p data-bbox="863 1081 1098 1108">Τάση λειτουργίας: 3.3V</p> <p data-bbox="863 1113 1358 1140">Ψηφιακές εισοδοί / έξοδοι: 14 (6 με PWM έξοδο)</p> <p data-bbox="863 1144 1238 1171">Ρεύμα DC ανά είσοδο / έξοδο: 40 mA</p> <p data-bbox="863 1176 1082 1202">Αναλογικές εισοδοί: 8</p> <p data-bbox="863 1207 1058 1234">Μνήμη Flash: 32 KB</p> <p data-bbox="863 1238 978 1265">SRAM: 2 KB</p> <p data-bbox="863 1270 1007 1296">EEPROM: 1 KB</p> <p data-bbox="863 1301 1134 1328">Ταχύτητα ρολογιού: 8 MHz</p>
 <p data-bbox="392 1621 676 1648">Εικόνα 6: Arduino Mega ADK</p>	<p data-bbox="863 1386 1158 1413">Μικροελεγκτής: ATmega2560</p> <p data-bbox="863 1417 1078 1444">Τάση λειτουργίας: 5V</p> <p data-bbox="863 1449 1369 1476">Ψηφιακές εισοδοί / έξοδοι: 54 (15 με PWM έξοδο)</p> <p data-bbox="863 1480 1238 1507">Ρεύμα DC ανά είσοδο / έξοδο: 40 mA</p> <p data-bbox="863 1512 1098 1538">Αναλογικές εισοδοί: 16</p> <p data-bbox="863 1543 1070 1570">Μνήμη Flash: 256 KB</p> <p data-bbox="863 1574 978 1601">SRAM: 8 KB</p> <p data-bbox="863 1606 1007 1632">EEPROM: 4 KB</p> <p data-bbox="863 1637 1145 1664">Ταχύτητα ρολογιού: 16 MHz</p>
 <p data-bbox="424 1872 644 1899">Εικόνα 7: Arduino Due</p>	<p data-bbox="863 1682 1174 1709">Μικροελεγκτής: AT91SAM3X8E</p> <p data-bbox="863 1713 1098 1740">Τάση λειτουργίας: 3.3V</p> <p data-bbox="863 1744 1225 1771">Συνολικό DC ρεύμα εξόδου: 130 mA</p> <p data-bbox="863 1776 1358 1803">Ψηφιακές εισοδοί/έξοδοι: 54 (12 με PWM έξοδο)</p> <p data-bbox="863 1807 1098 1834">Αναλογικές εισοδοί: 12</p> <p data-bbox="863 1839 1134 1865">Αναλογικές έξοδοι: 2 (DAC)</p> <p data-bbox="863 1870 1070 1897">Μνήμη Flash: 512 KB</p> <p data-bbox="863 1901 1270 1928">SRAM: 96 KB (two banks: 64KB and 32KB)</p> <p data-bbox="863 1933 1145 1960">Ταχύτητα ρολογιού: 84 MHz</p>




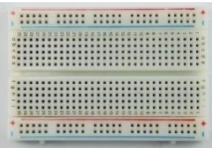
 <p>Εικόνα 8: Arduino Leonardo</p>	<p>Μικροελεγκτής: ATmega32u4 Τάση λειτουργίας: 5V Ψηφιακές εισοδοί / έξοδοι: 20 (7 παρέχουν PWM έξοδο) Αναλογικές εισοδοί: 12 Ρεύμα DC ανά είσοδο / έξοδο: 40 mA Μνήμη Flash: 32 KB (ATmega32u4) SRAM: 2.5 KB (ATmega32u4) EEPROM: 1 KB (ATmega32u4) Ταχύτητα ρολογιού: 16 MHz</p>
 <p>Εικόνα 9: Arduino Yún</p>	<p>Μικροελεγκτής: ATmega32u4 Τάση λειτουργίας: 5V Ψηφιακές εισοδοί / έξοδοι: 20 (7 παρέχουν PWM έξοδο) Ρεύμα DC ανά είσοδο / έξοδο: 40 mA Αναλογικές εισοδοί: 12 Μνήμη Flash: 32 KB SRAM: 2.5 KB EEPROM: 1 KB Ταχύτητα ρολογιού: 16 MHz Μικροεπεξεργαστής Linux: Processor Atheros AR9331 Αρχιτεκτονική: MIPS @400MHz Τάση λειτουργίας: 3.3V Ethernet: IEEE 802.3 10/100Mbit/s Wi-Fi: IEEE 802.11b/g/n Card Reader: Micro-SD only RAM: 64 MB DDR2 Μνήμη Flash: 16 MB</p>
 <p>Εικόνα 10: Arduino Micro</p>	<p>Μικροελεγκτής: ATmega32u4 Τάση λειτουργίας: 5V Ψηφιακές εισοδοί / έξοδοι: 20 (7 παρέχουν PWM έξοδο) Ρεύμα DC ανά είσοδο / έξοδο: 40 mA Αναλογικές εισοδοί: 12 Μνήμη Flash: 32 KB (ATmega32u4) SRAM: 2.5 KB (ATmega32u4) EEPROM: 1 KB (ATmega32u4) Ταχύτητα ρολογιού: 16 MHz</p>

1.5 Ηλεκτρονικά εξαρτήματα που συνδέονται στο Arduino

Για να χρησιμοποιηθούν τα ηλεκτρονικά εξαρτήματα με το Arduino δε χρειάζεται προηγούμενη εμπειρία. Παρακάτω παρατίθενται κάποια από τα βασικά εξαρτήματα ώστε να μπορούν να αναγνωριστούν και να κατανοηθεί η χρήση τους.

<p>LED</p> 	<p>Τι κάνει Εκπέμπει φως όταν μικρό ρεύμα περάσει μέσα του (μόνο στη μια κατεύθυνση) Πως το αναγνωρίζουμε Μοιάζει με μικρή λάμπα.</p>	<p>Αριθμός επαφών 2 (ένα μακρύτερο που συνδέεται με το θετικό πόλο) Τι να προσέξουμε -Λειτουργεί μόνο στη μια κατεύθυνση -Απαιτεί αντιστάτη</p>
---	--	---

<p>ΔΙΟΔΟΣ (DIODE)</p> 	<p>Τι κάνει Είναι το αντίστοιχο ηλεκτρονικό στοιχείο μιας μονόδρομης βαλβίδας. Επιτρέπει το ρεύμα να περνάει προς μια κατεύθυνση αλλά όχι και αντίστροφα.</p> <p>Πως το αναγνωρίζουμε Συνήθως είναι κύλινδρος με επαφές που προεξέχουν από τα δύο άκρα του</p>	<p>Αριθμός επαφών 2 Τι να προσέξουμε Δουλεύει μόνο προς τη μια κατεύθυνση</p>
<p>ΑΝΤΙΣΤΑΤΗΣ (RESISTOR)</p> 	<p>Τι κάνει Περιορίζει την ποσότητα ρεύματος που περνά μέσα από ένα κύκλωμα.</p> <p>Πως το αναγνωρίζουμε Είναι κύλινδρος με εξογκωμένα άκρα και επαφές στα δύο άκρα του.</p>	<p>Αριθμός επαφών 2 Τι να προσέξουμε Να επιλέγουμε το σωστό αντιστάτη ανάλογα το κύκλωμα.</p>
<p>ΤΡΑΝΖΙΣΤΟΡ (TRANSISTOR)</p> 	<p>Τι κάνει Χρησιμοποιεί ένα μικρό ρεύμα για να ενεργοποιήσει ή ενισχύσει ένα μεγαλύτερο ρεύμα.</p> <p>Πως το αναγνωρίζουμε Είναι συνήθως μισός μαύρος κύλινδρος με 3 ποδαράκια-επαφές</p>	<p>Αριθμός επαφών 3(βάση, συλλέκτης, εκπομπή) Τι να προσέξουμε Πρέπει να συνδέονται τα ποδαράκια-επαφές με το σωστό τρόπο. Συνήθως χρειάζεται ένας αντιστάτης στην επαφή της Βάσης</p>
<p>ΠΟΤΕΝΣΙΟΜΕΤΡΟ (POTENTIOMETER)</p> 	<p>Τι κάνει Παρέχει διαφορετική αντίσταση ανάλογα με τη θέση της γωνίας του δείκτη.</p> <p>Πως το αναγνωρίζουμε Έχει μία βίδα ή επιφάνεια που περιστρέφεται και έχει ένα βελάκι επάνω του.</p>	<p>Αριθμός επαφών 3 Τι να προσέξουμε Να μην αγοραστεί κατά λάθος ποτενσιόμετρο λογαριθμικής κλίμακας.</p>
<p>ΚΟΥΜΠΙ (PUSH BUTTON)</p> 	<p>Τι κάνει Ενεργοποιεί το κύκλωμα όταν πατηθεί.</p> <p>Πως το αναγνωρίζουμε Μικρό τετράγωνο με οδηγούς στο κάτω μέρος κι ένα κουμπί στην κορυφή.</p>	<p>Αριθμός επαφών 4 Τι να προσέξουμε Είναι τετράγωνο οπότε μπορούν να εισαχθούν με 90 μοίρες γωνία.</p>
<p>ΠΙΕΖΟ ΣΤΟΙΧΕΙΟ (PIEZO ELEMENT)</p> 	<p>Τι κάνει Ένας παλμός ρεύματος το ενεργοποιεί. Η ροή των παλμών θα το αναγκάσει να εκπέμψει έναν ήχο.</p> <p>Πως το αναγνωρίζουμε Συνήθως είναι ένα μαύρο βαρελάκι κι άλλες φορές απλά ένας χρυσός δίσκος.</p>	<p>Αριθμός επαφών 2 Τι να προσέξουμε Είναι αδύνατο να μη χρησιμοποιηθεί σωστά.</p>

<p>DC ΜΟΤΕΡ</p> 	<p>Τι κάνει Περιστρέφεται όταν περνά ρεύμα. Πως το αναγνωρίζουμε Μοιάζει με μοτέρ</p>	<p>Αριθμός επαφών 2 Τι να προσέξουμε Πρέπει να χρησιμοποιείται τρανζίστορ το οποίο να ταιριάζει στο μοτέρ.</p>
<p>ΦΩΤΟΑΝΤΙΣΤΑΤΗΣ</p> 	<p>Τι κάνει Παράγει μια μεταβλητή αντίσταση η οποία εξαρτάται από την ποσότητα του φωτός που δέχεται. Πως το αναγνωρίζουμε Συνήθως είναι ένας μικρός δίσκος με διάφανη κορυφή και μια καμπυλωτή γραμμή από κάτω.</p>	<p>Αριθμός επαφών 2 Τι να προσέξουμε Πρέπει να είναι περάσει από έναν διαιρετή τάσης πριν δώσει αποτέλεσμα.</p>
<p>Καλώδια σύνδεσης</p> 	<p>Τι κάνει Καλώδια σύνδεσης Πως το αναγνωρίζουμε Χρωματιστά καλώδια με μαύρα άκρα</p>	<p>Αριθμός επαφών 2 Τι να προσέξουμε Υπάρχουν 3 ειδών αρσενικά στα 2 άκρα (M-M), αρσενικό-θηλυκό (M-F), και θηλυκά και στα 2 άκρα (F-F)</p>
<p>Βοηθητική πλακέτα</p> 	<p>Τι κάνει Βοηθητική πλακέτα (breadboard) Πως το αναγνωρίζουμε Ορθογώνια πλακέτα με τρύπες ομαδοποιημένα σε 4 σειρές και πολλές στήλες</p>	<p>Αριθμός επαφών Ανάλογα με το μέγεθος Τι να προσέξουμε Οι 2 πάνω και κάτω σειρές (με σύμβολα +, -) είναι συνδεδεμένα εσωτερικά σε γραμμές. Οι εσωτερικές 2 ομάδες είναι συνδεδεμένες κατά στήλες</p>

Επιπλέον υπάρχουν και πολλά ειδικά εξαρτήματα και αισθητήρες όπως ενδεικτικά τα παρακάτω:

<p>Αισθητήρας Θερμοκρασίας Υγρασίας</p> 	<p>Τι κάνει Μετρά Θερμοκρασία και Υγρασία περιβάλλοντος Πως το αναγνωρίζουμε Έχει ένα λευκό διάτρητο κουτί</p>	<p>Αριθμός επαφών 3 Τι να προσέξουμε Υπάρχει και με μικρότερη ακρίβεια και είναι μπλε</p>
<p>Οθόνη</p> 	<p>Τι κάνει Δείχνει στην οθόνη δύο γραμμών χαρακτήρες και αριθμούς για να βλέπουμε μετρήσεις Πως το αναγνωρίζουμε Έχει μία οθόνη ορθογώνιου παραλληλόγραμμου πάνω σε πλακέτα</p>	<p>Αριθμός επαφών 16 Τι να προσέξουμε Υπάρχουν με οπίσθιο φωτισμό και χωρίς. Επίσης υπάρχουν TFT οθόνες (επαφής ή όχι) διαφόρων μεγεθών που μπορούν να έχουν γραφικά και να απεικονίζουν περισσότερους χαρακτήρες</p>

<p>WiFi επικοινωνίας</p> 	<p>Τι κάνει Δημιουργεί ασύρματη επικοινωνία wifi με access point και συνδέεται στο διαδίκτυο</p> <p>Πως το αναγνωρίζουμε Έχει στο ένα άκρο μια χρυσή τεθλασμένη γραμμή (ζικ-ζακ) που είναι η κεραία του</p>	<p>Αριθμός επαφών 24</p> <p>Τι να προσέξουμε Υπάρχουν πολλές εκδόσεις με διαφορετικά χαρακτηριστικά. Απαιτούνται ιδιαίτερες γνώσεις προγραμματισμού και δικτύων.</p>
<p>Αισθητήρας μονοξειδίου του άνθρακα MQ7</p> 	<p>Τι κάνει Αισθητήρας για ανίχνευση μονοξειδίου του άνθρακα</p> <p>Πως το αναγνωρίζουμε Γράφει MQ-7</p>	<p>Αριθμός επαφών 6</p> <p>Τι να προσέξουμε Χρειάζεται να συνδεθούν Pins μεταξύ τους</p>
<p>Αισθητήρας καπνού MQ2</p> 	<p>Τι κάνει Αισθητήρας για εύφλεκτα αέρια (Μεθάνιο, Βουτάνιο, LPG, καπνό)</p> <p>Πως το αναγνωρίζουμε Γράφει MQ-2</p>	<p>Αριθμός επαφών 4</p> <p>Τι να προσέξουμε Κάποιες υλοποιήσεις έχουν μεταβλητή αντίσταση σε ποτενσιόμετρο για να ρυθμίζεται η ευαισθησία του</p>

2 Βηματικοί κινητήρες (stepper motors)



Εικόνα 11: Βηματικός κινητήρας NEMA-17

2.1 Γενικές πληροφορίες. Τι είναι ο βηματικός κινητήρας.

Ένας τύπος κινητήρα που χρησιμοποιείται για τον ακριβή έλεγχο της θέσης του άξονά του είναι ο βηματικός κινητήρας ή stepper motor.

Ο βηματικός κινητήρας είναι μια συσκευή ηλεκτρομηχανικής κίνησης η οποία μετατρέπει ψηφιακό σήμα σε μηχανική κίνηση. Είναι μια ειδική έκδοση μιας σύγχρονης μηχανής, στην οποία ο ρότορας είναι ένας μόνιμος μαγνήτης, ενώ ο στάτης αποτελείται από κάποια τυλίγματα. Σε αντίθεση όμως με τους συγχρόνους κινητήρες, οι βηματικοί κινητήρες έχουν μεγάλο αριθμό ζευγών πόλων.

Είναι μοτέρ που χρειάζεται συνεχές (DC) ρεύμα για την λειτουργία του, δεν έχει καρβουνάκια και η κατασκευή του είναι συμπαγής με αντοχή και οικονομική. Τον βρίσκουμε συνήθως σε μικρές ισχύς και διαστάσεις, σε εκτυπωτές, scanner, ελεγκτές θέσης ηλεκτροβαλβίδων, φρέζες CNC και αλλού.

Πρόκειται για κινητήρες DC μόνιμου μαγνήτη, οι οποίοι έχουν τα εξής χαρακτηριστικά λειτουργίας: μπορούν να περιστραφούν και στις δυο κατευθύνσεις, να κινηθούν με ακρίβεια κατά συγκεκριμένη γωνία, να διατηρήσουν ροπή συγκράτησης υπό μηδενική ταχύτητα και να ελεγχθούν με ψηφιακά κυκλώματα. Η ακριβής κίνηση κατά τμήματα συγκεκριμένης γωνίας, που ονομάζονται βήματα (steps), γίνεται με την εφαρμογή ψηφιακών παλμών σε ένα ηλεκτρικό κύκλωμα οδήγησης. Ο αριθμός και ο ρυθμός των παλμών ελέγχουν τη θέση και την ταχύτητα του άξονα του κινητήρα.

Το όνομα του: "βηματικός" προέρχεται από τον τρόπο λειτουργίας του, δίνοντας συνεχή τάση χωρίς μεταβολή της κάνει ένα και μόνο μικρό βήμα συγκεκριμένης γωνίας. Σε αυτή την θέση μένει "κολλημένο". Δίνοντας μετά τροφοδοσία με άλλο συνδυασμό σύνδεσης ή πολικότητας ο άξονας κάνει άλλο ένα βήμα. Μετά από έναν συγκεκριμένο αριθμό βημάτων οι εντολές επαναλαμβάνονται. Γενικά οι βηματικοί κινητήρες κατασκευάζονται με δυνατότητα 12, 24, 72, 144, 180 και 200 βημάτων ανά πλήρη περιστροφή, που σημαίνει ότι ο άξονας περιστρέφεται 30°, 15°, 5°, 2.5°, 2° και 1.8° ανά βήμα. Εάν υπάρχει ανάγκη ακριβέστερης κίνησης, είναι δυνατό να σχεδιαστούν κυκλώματα μικροβηματισμού (micro_stepping), που να επιτρέπουν πολύ περισσότερα βήματα ανά περιστροφή και ο αριθμός μπορεί να φθάνει, ή και να ξεπερνά τα 10000 steps/rev.

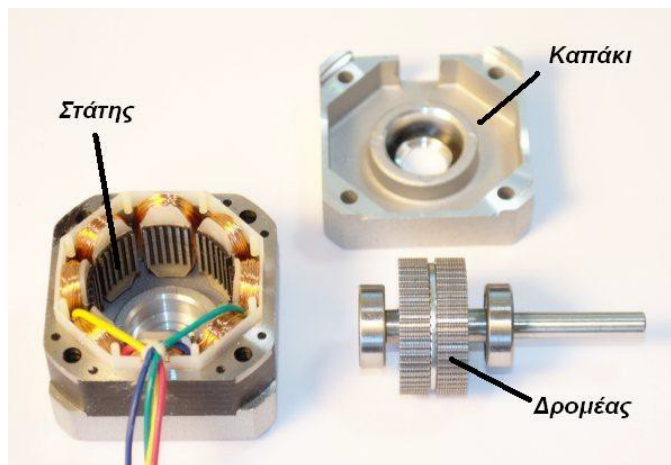
Οι βηματικοί κινητήρες είναι είτε μονοπολικόι απαιτώντας μόνο μια πηγή τροφοδοσίας, είτε διπολικόι απαιτώντας δυο πηγές ενέργειας ή μια πηγή με δυνατότητα αλλαγής της πολικότητας. Τροφοδοτούνται από πηγές DC και χρειάζονται ψηφιακά κυκλώματα για την εκκίνηση των διαδικασιών ενεργοποίησης των πηνίων, ώστε να περιστραφεί ο κινητήρας. Η ανάδραση δεν είναι πάντα απαραίτητη, αλλά η χρήση ενός κωδικοποιητή, ή κάποιου άλλου αισθητήρα θέσης μπορεί να εξασφαλίσει ακρίβεια στις εφαρμογές, που ο ακριβής έλεγχος θέσης αποτελεί βασική προϋπόθεση. Το πλεονέκτημα της λειτουργίας χωρίς ανάδραση είναι, ότι δε χρειάζεται το σύστημα ελέγχου κλειστού βρόχου. Γενικά οι βηματικοί κινητήρες αποδίδουν λιγότερο από 1 hp (746 W), με αποτέλεσμα να χρησιμοποιούνται μόνο σε εφαρμογές ελέγχου θέσης χαμηλής ισχύος.

Ένας βηματικός κινητήρας, που διατίθεται στο εμπόριο, αποτελείται από έναν μεγάλο αριθμό πόλων, που καθορίζει έναν αντίστοιχα μεγάλο αριθμό θέσεων ισορροπίας του κινητήρα. Στην περίπτωση ενός βηματικού κινητήρα μόνιμου μαγνήτη, ο στάτορας αποτελείται από πόλους με περιέλιξη, ενώ οι πόλοι του ρότορα είναι μόνιμοι μαγνήτες. Η διέγερση διαφορετικών συνδυασμών περιελίξεων μετακινεί και σταθεροποιεί τον ρότορα σε διαφορετικές θέσεις.

Όταν χρειαζόμαστε ακρίβεια και επαναληψιμότητα, ένας βηματικός κινητήρας είναι πάντα η λύση. Με τον τρόπο που έχει σχεδιαστεί, ένα stepper μπορεί να μετακινηθεί από το ένα βήμα στο επόμενο και να στερεωθεί σε αυτή τη θέση. Ένας τυπικός κινητήρας έχει 200 βήματα ανά περιστροφή, αν πούμε ότι ο κινητήρας έχει 100 βήματα προς μία κατεύθυνση, θα γυρίσει ακριβώς 180 μοίρες.

2.2 Δομή ενός βηματικού κινητήρα

Παρακάτω απεικονίζεται η δομή ενός υβριδικού βηματικού κίνητρα τεσσάρων φάσεων. Να σημειωθεί ότι η δομή και η κατασκευή για κάθε διαφορετικό είδος ενός βηματικού κινητήρα, είναι διαφορετική.



Εικόνα 12: Ανοιγμένος υβριδικός βηματικός κινητήρας

Ο δρομέας (ή ρότορας)

Ο δρομέας όπως γνωρίζουμε από όλους του κινητήρες είναι το κινητό μέρος του κινητήρα. Ο δρομέας στον συγκεκριμένο βηματικό κινητήρα του σχήματος είναι κατασκευασμένος από δυο δίσκους, που μοιάζουν σαν γρανάζια, ένας από τους οποίους είναι ο βόρειος πόλος και ο άλλος ο νότιος πόλος. Όταν λοιπόν τοποθετήσουμε τους δυο αυτούς δίσκους μαζί έχουμε εναλλάξ στις οδοντώσεις βόρεια και νότια πολικότητα. Σε άλλα είδη βηματικών κινητήρων, ο δρομέας μπορεί να μην έχει οδοντώσεις, αλλά να είναι απλά ένας μόνιμος μαγνήτης.

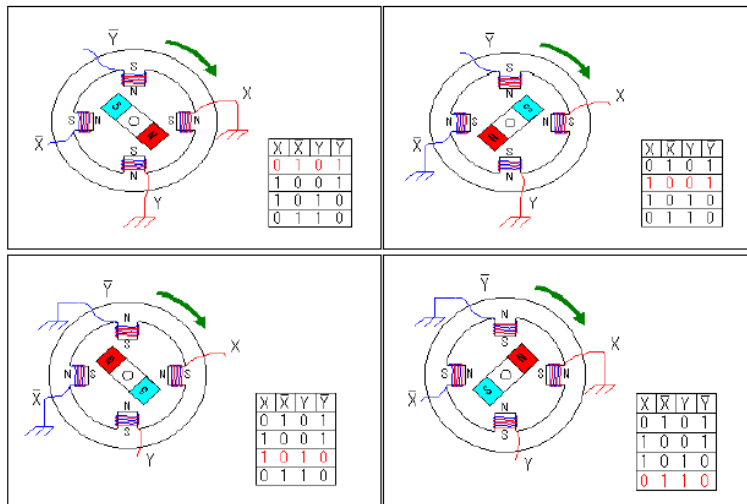
Ο στάτης

Ο στάτης από την άλλη είναι το ακίνητο μέρος του κινητήρα. Περιέχει τους πόλους οι οποίοι καταλήγουν σε κάποιους ακροδέκτες. Οι πόλοι που έχει ο συγκεκριμένος βηματικός κινητήρας είναι 4, δηλαδή έχουμε 8 τυλίγματα όπου κάθε ζεύγος αυτών κάνουν και 1 πόλο. Τα τυλίγματα όπως βλέπουμε στο σχήμα είναι και εδώ οδοντωτά, παρόλα αυτά σε άλλα είδη βηματικών κινητήρων, μπορεί να μην έχουν οδοντώσεις. Οι ακροδέκτες που προέρχονται από τα τυλίγματα θα πρέπει να συνδεθούν σε έναν οδηγό και έναν ελεγκτή, για να δοθεί παλμός σε αυτά, αφού όπως τονίσαμε παραπάνω, οι συγκεκριμένοι κινητήρες λειτουργούν δίνοντάς τους παλμούς.

2.3 Αρχή λειτουργίας βηματικού κινητήρα

Για να λειτουργήσει ένας τέτοιος βηματικός κινητήρας χρειάζεται μια μονάδα έλεγχου, η οποία ενεργοποιεί τα τυλίγματα ξεχωριστά με βάση μια συγκεκριμένη ακολουθία παλμών. Άρα ο βηματικός κινητήρας λειτουργεί με παλμούς, κάθε παλμός δίνει εντολή να ενεργοποιηθεί ένα τύλιγμα και έτσι ο δρομέας έλκετε από το συγκεκριμένο τύλιγμα που ενεργοποιήθηκε, κάνοντας έτσι ένα βήμα, και από αυτό η ονομασία του βηματικός κινητήρας.

Ανάλογα και με τον κινητήρα και κάποιες άλλες παραμέτρους, ο βηματικός κινητήρας κάνει ένα βήμα με μια συγκεκριμένη γωνία. Για να κάνει λοιπόν ο κινητήρας μια πλήρη περιστροφή 360° θα χρειαστεί έναν συγκεκριμένο αριθμό παλμών, έτσι όπως απεικονίζεται στο παρακάτω σχήμα χρειάζεται 4 παλμούς και για κάθε παλμό κάνει 270° . Στην παρακάτω εικόνα απεικονίζεται το πώς ακριβώς, γίνεται αυτή η λειτουργία βήμα βήμα.



Εικόνα 13: Τρόπος λειτουργίας διπολικού βηματικού κινητήρα

Ας υποθέσουμε λοιπόν όπως απεικονίζεται στην εικόνα 13, ότι έχουμε έναν βηματικό κινητήρα που έχει 2 ζεύγη πόλων δηλαδή 4 διαφορετικά τυλίγματα. Ονομάζουμε τα τυλίγματα Y, \bar{Y}, X, \bar{X} ανάλογα με την πολικότητα που έχουν. Αφού κατανοήσουμε ότι όταν έχουμε 1 έχουμε και παλμό άρα και ενεργοποίηση του τυλίγματος, ενώ όταν έχουμε 0 δεν υπάρχει παλμός, συνεχίζουμε σε μια σειρά ενεργοποίησης τυλιγμάτων.

Κάθε φορά που ένα ή δυο τυλίγματα (ανάλογα με το είδος βηματισμού που έχουμε) ενεργοποιούνται, ο δρομέας μας θα κινηθεί προς την αντίστοιχη πολικότητα των τυλιγμάτων (που βρίσκονται στον στάτη). Έτσι έχοντας μια ακολουθία από διαδοχικούς παλμούς στα τυλίγματα πετυχαίνουμε την κίνηση του κινητήρα.

Στο παράδειγμα μας, η διαδοχή με την οποία ενεργοποιούνται τα τυλίγματα, οδηγούν τον κινητήρα να περιστρέφεται δεξιόστροφα. Για να περιστραφεί αριστερόστροφα θα πρέπει να ενεργοποιήσουμε τα τυλίγματα αντίθετα από αυτό το παράδειγμα.

Ο βηματικός κινητήρας λόγω της αρχής λειτουργίας του, έχει πλεονέκτημα σε εφαρμογές που χρειάζονται απόλυτος έλεγχος κίνησης, όχι μόνο σε περιστρεφόμενη κίνηση αλλά και σε διαδοχικές κινήσεις, δεξιόστροφης και αριστερόστροφης φοράς, σε εναλλασσόμενες ταχύτητες.

2.4 Χαρακτηριστικά κινητήρα NEMA-17 size - 200 steps/rev, 12V 350mA

Εμείς θα χρησιμοποιήσουμε το μοντέλο βηματικού κινητήρα, NEMA-17 size - 200 steps/rev, 12V 350mA

- Ο κινητήρας αυτός σε μία πλήρη περιστροφή έχει 200 βήματα.
- Είναι διπολικός κινητήρας και συνδέεται με 4 καλώδια, για να στραφεί 1,8° ανά βήμα, για να πετύχουμε, ομαλή κίνηση και σωστή ροπή συγκράτησης.
- Οι προδιαγραφές του βηματικού κινητήρα είναι να έχει μέγιστο ρεύμα 350mA, έτσι ώστε να μπορεί να κινείται εύκολα με shield driver for Arduino και βάση.

Τεχνικές λεπτομέρειες

- Διπολικός 4 καλωδίων
- 200 βήματα ανά περιστροφή, 1,8 μοίρες
- Πηνίο # 1: Κόκκινο και κίτρινο σύρμα ζεύγος.
- Πηνίο # 2 Πράσινο & Καφέ / Γκρι σύρμα ζεύγος.
- Διπολικός βηματικός, απαιτεί 2 γεμάτες Η-γέφυρες!

42 mm τετράγωνο σώμα

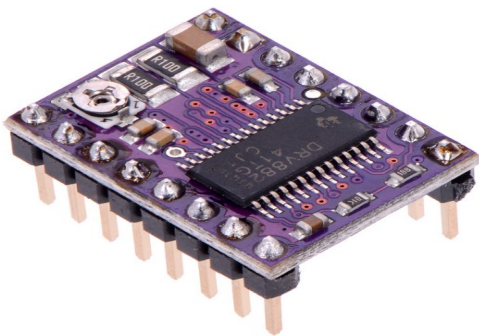
- 31mm τετράγωνες οπές τοποθέτησης,
- Μετρικές βίδες 3mm (M3)
- Άξονας κίνησης διαμέτρου 5 χιλ., μήκους 24 χλστ., με κατεργασμένο επίπεδο
- 12V ονομαστική τάση (μπορούμε να το οδηγήσουμε σε χαμηλότερη τάση, αλλά η ροπή θα πέσει) σε μέγιστο ρεύμα 350mA
- 20 N • cm, 2 Kg • cm, ροπή συγκράτησης ανά φάση
- 35 ohm ανά περιέλιξη

2.5 Κύκλωμα οδηγού κινητήρα

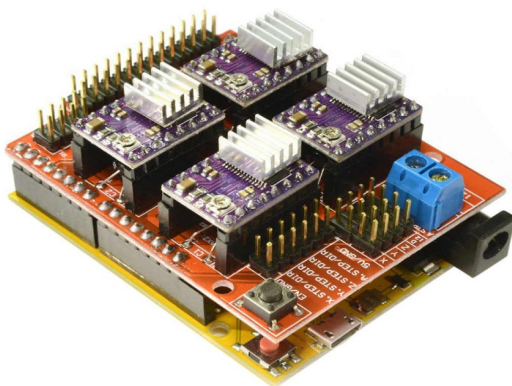
Για τον έλεγχο του χρειάζεται κύκλωμα οδηγού (driver), που έχει τα απαραίτητα ηλεκτρονικά που χρειάζονται για την τροφοδοσία και σύνδεση των συνήθως δυο πηνίων του.

Πώς συνδέεται ο βηματικός κινητήρας

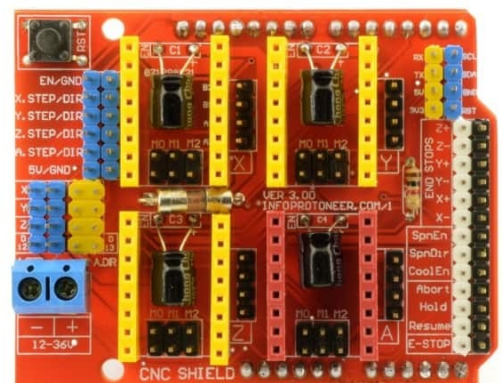
Πάνω στο Arduino, προσαρμόζεται ένα Arduino CNC Shield V3 (Εικόνα 15), το οποίο είναι μια πλακέτα επέκτασης. Αυτή η πλακέτα επέκτασης, μπορεί να χρησιμοποιηθεί για χαρακτηριστικά μηχανήματα και εκτυπωτές 3D. Έχει συνολικά 4 slots και μπορεί να οδηγήσει τέσσερις A4988 ή DRV8825 drivers βηματικών κινητήρων (εικόνα 14). Κάθε βηματικός κινητήρας χρειάζεται μόνο δύο θύρες I/O.



Εικόνα 14: Ο Driver DR8825 του βηματικού κινητήρα

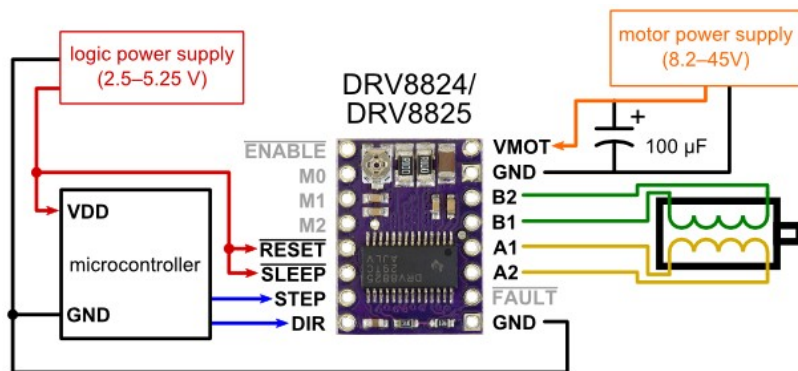


Εικόνα 16: Το Arduino CNC Shield V3, με τους 4 Drivers DRV8825

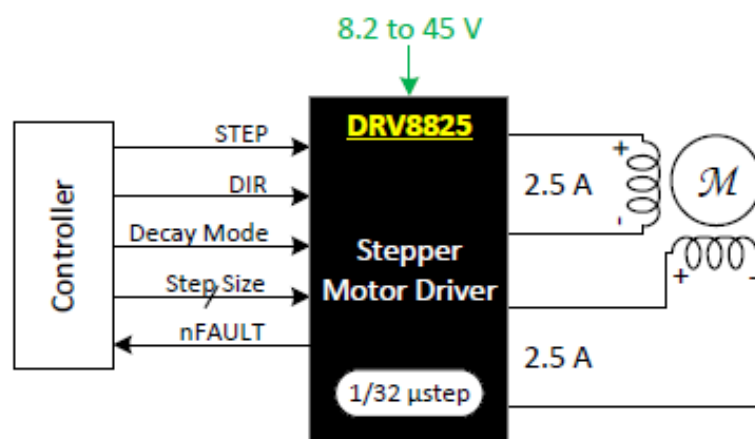


Εικόνα 15: Το Arduino CNC Shield V3

Στο επόμενο σχήματα περιγράφεται ο τρόπος σύνδεσης του βηματικού κινητήρα, μέσω του driver με το Arduino.



Εικόνα 17: Κύκλωμα σύνδεσης μικροελεγκτή με τον driver DR8825 βηματικού κινητήρα



Εικόνα 18 Σύνδεση driver DR8825 με controller

3 Περιβάλλον Προγραμματισμού του Arduino

Το περιβάλλον προγραμματισμού που συνήθως χρησιμοποιείται για το Arduino διατίθεται δωρεάν από τον ιστότοπο <https://www.arduino.cc/en/Main/Software> και είναι κατάλληλο για λειτουργικά συστήματα Windows, Mac OS X και Linux.

Στη συνέχεια και τις παρούσες ανάγκες γίνεται χρήση του Arduino IDE

3.1 Εγκατάσταση λογισμικού Arduino

Η εγκατάσταση για περιβάλλον λειτουργικού συστήματος Windows γίνεται με τα ακόλουθα βήματα:

Βήμα 1. «Κατεβάζουμε» το λογισμικό για το λειτουργικό μας σύστημα από το <https://www.arduino.cc/en/Main/Software> κάτω από την ετικέτα Download the Arduino IDE

Βήμα 2. Εκτελούμε το αρχείο `arduino-1.8.12-windows.exe`, που μόλις κατεβάσαμε (είναι η τρέχουσα έκδοση π.χ. 1.8.12)

Βήμα 3. Συνδέουμε την πλακέτα Arduino μέσω του USB καλωδίου με τον υπολογιστή και εμφανίζεται το παράθυρο προσθήκης νέου υλικού και αναζητούμε τους κατάλληλους οδηγούς στο path `C:\Program Files (x86)\Arduino\drivers`

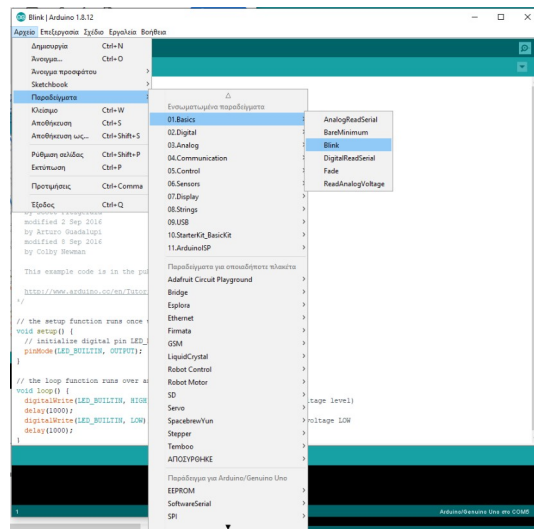


Εικόνα 19: Το περιβάλλον προγραμματισμού Arduino IDE

Βήμα 4. Μετά τη σύνδεση του Arduino board στην USB του υπολογιστή και την εκκίνηση του Arduino IDE επιλέγουμε στο Tools-Port την νέα σειριακή θύρα που έχει εμφανιστεί π.χ. COM3

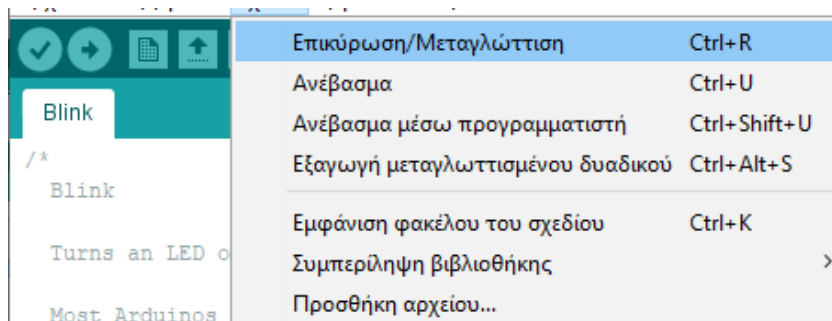
Βήμα 5. Επιλέγουμε τον τύπο του Arduino board που έχουμε στη διάθεσή μας από το Tools-Board

Στο περιβάλλον προγραμματισμού υπάρχουν αρκετά έτοιμα παραδείγματα με σχόλια στον κώδι-



Εικόνα 20: Επιλέγουμε το παράδειγμα Blink που μπορούμε να τα επιλέξουμε από το File-Examples. Για παράδειγμα στο Examples-01 Basics, μπορούμε να επιλέξουμε το παράδειγμα Blink. (Εικόνα 19). Περιέχει έτοιμο τον κώδικα με σχόλια.

Για να τρέξει το πρόγραμμα επιλέγουμε Επικύρωση / Μεταγλώττιση, για να γίνει ο έλεγχος σφαλμάτων και να «μεταγλωττιστεί» όπως λέμε το πρόγραμμα C σε γλώσσα που καταλαβαίνει ο επεξεργαστής του Arduino και μετά Ανέβασμα, για να «φορτωθεί» το πρόγραμμα στη μνήμη του Arduino και να αρχίσει να εκτελείται. Όταν γίνει αυτό βλέπουμε το LED να αναβοσβήνει.

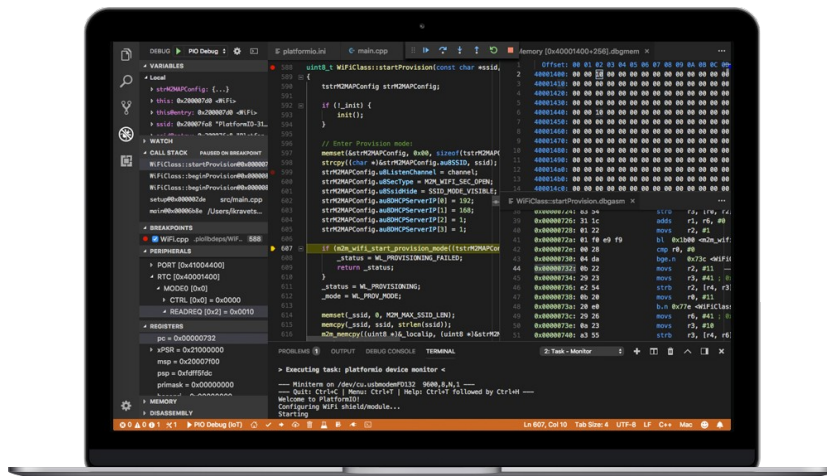


Εικόνα 21: Επικύρωση / Μεταγλώττιση

3.2 Άλλα περιβάλλοντα προγραμματισμού

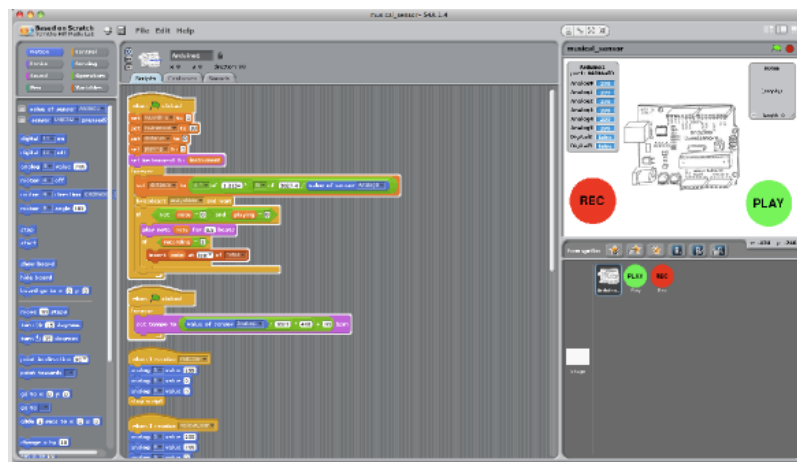
Λόγω της ευκολίας δημιουργίας πληθώρας πρωτότυπων projects έχει δημιουργηθεί μεγάλη κοινότητα για τα Arduino και φυσικά έχουν δημιουργηθεί πολλά περιβάλλοντα προγραμματισμού με χρήση C/C++/C#, για όσους γνωρίζουν τα βασικά του προγραμματισμού ή με προγραμματισμό με χρήση δομικών στοιχείων βασισμένα σε Scratch (programming building blocks), για όσους δεν έχουν καθόλου εμπειρία, αλλά και προσομοιωτών (simulators) για δοκιμές πριν την υλοποίηση, όπως: Ακολουθούν ενδιαφέροντα περιβάλλοντα προγραμματισμού

PlatformIO IDE <http://platformio.org/platformio-ide>



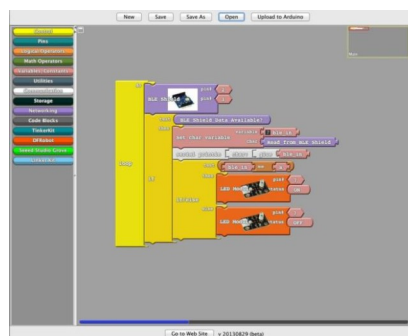
Εικόνα 22: PlatformIO IDE

S4A <http://s4a.cat/>



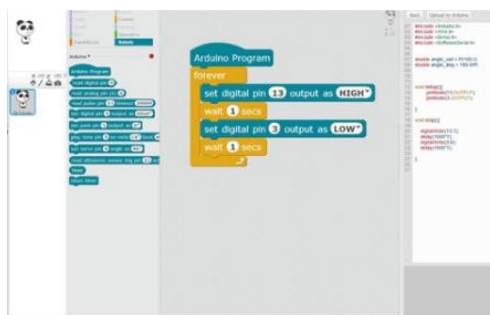
Εικόνα 23: S4A

Ardublock <https://sourceforge.net/projects/ardublock/?source=navbar>



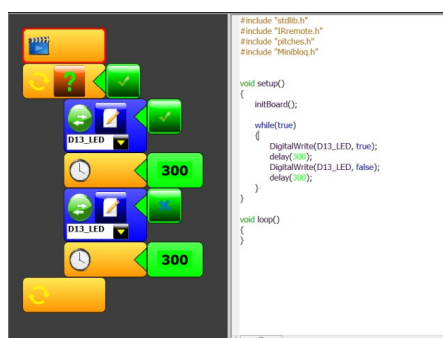
Εικόνα 24: Ardublock

mBlock <http://www.mblock.cc/>



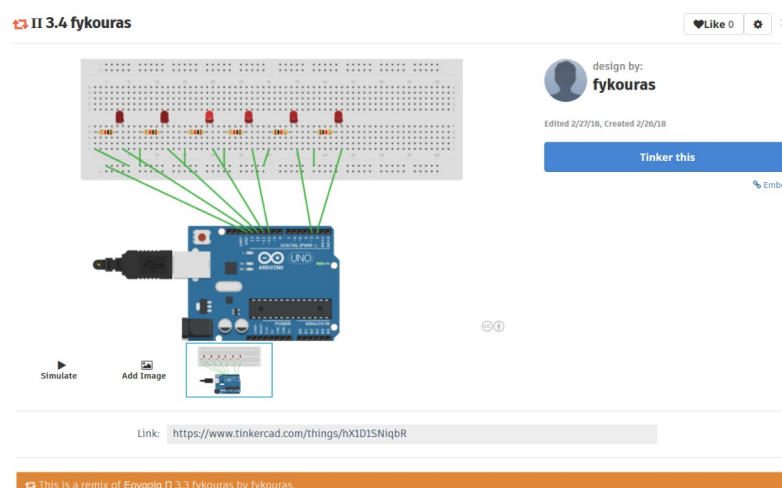
Εικόνα 25: mBlock

Minibloq <http://blog.minibloq.org/p/download.html>



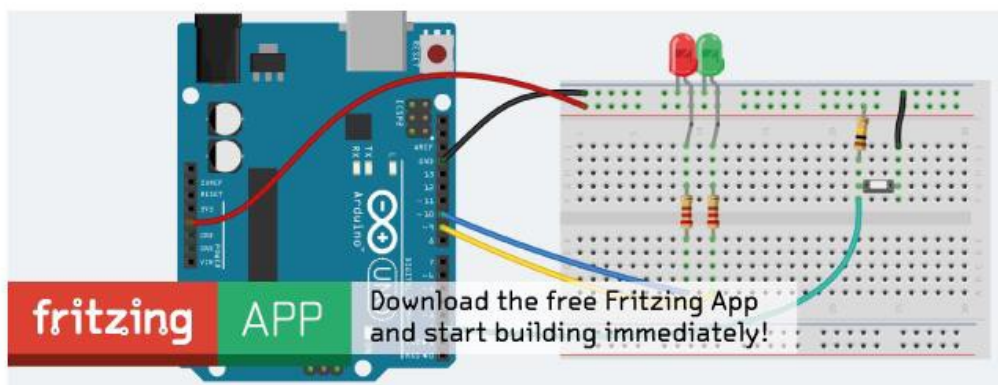
Εικόνα 26: Minibloq

Προσωμοιωτής: <https://www.tinkercad.com/circuits>



Εικόνα 27: Προσωμοιωτής tinkercad

Ταυτόχρονα υπάρχουν και σχεδιαστικά εργαλεία που βοηθούν στη δημιουργία σχηματικών, όπως το fritzing (<http://fritzing.org/home/>)



Εικόνα 28: fritzing

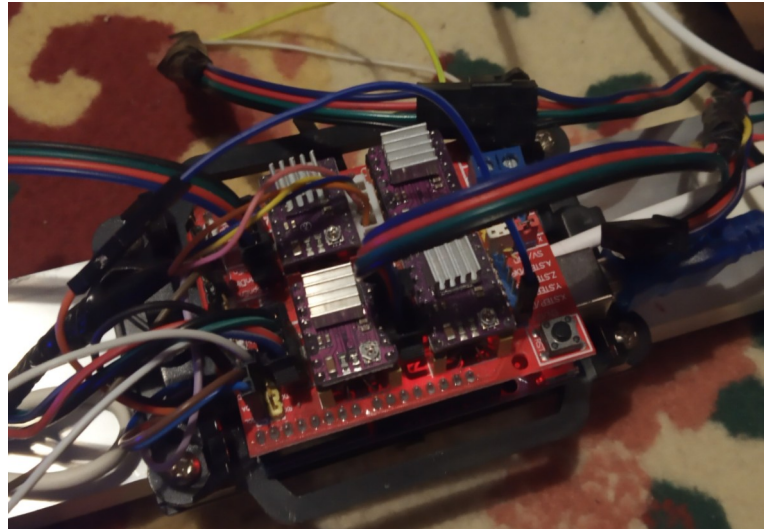
4 Περιγραφή κατασκευής

4.1 Υλικά

- ✓ Σασί: 2 ράβδοι αλουμινίου 72cm X 5cm X 2,5cm
2 ράβδοι αλουμινίου 59cm X 5cm X 2,5cm
- ✓ 2 κυλινδρικοί ράβδοι στον άξονα των X & 2 στον άξονα των Y
- ✓ 2 μάντες κατά μήκος του άξονα των X & 1 κατά μήκος του άξονα των Y
- ✓ 1 Arduino uno
- ✓ 1 CNC V3 Shield
- ✓ 4 Drivers DR8825 βηματικού κινητήρα
- ✓ 3 κινητήρες - step motor NEMA-17 size - 200 steps/rev, 12V 350mA
- ✓ 1 κινητήρας step motor 28BYJ 48 5VDC 15031801



Εικόνα 30: Ο βηματικός κινητήρας στον άξονα Y



Εικόνα 29: Οι 4 drivers DR8825 βηματικού κινητήρα πάνω στο Arduino μας

4.2 Υλοποίηση κατασκευής

Ενώσαμε τις αλουμινένιες ράβδους και δημιουργήσαμε ένα ορθογώνιο πλαίσιο, εξωτερικών διαστάσεων 77cm επί 64 cm. Η σύνδεση των ράβδων έγινε με αλουμινένιες βίδες. Το ορθογώνιο

πλαίσιο θα λειτουργήσει σαν “σασί”, στην κατασκευή μας. Κάτω από τις ράβδους προστέθηκε ελαστικό υλικό για την απορρόφηση των κραδασμών κατά την εκτύπωση.

Πάνω σε κάθε μία από τις μεγάλες πλευρές του αλουμινένιου ορθογωνίου σασί, δηλαδή στον άξονα X, τοποθετήθηκε μια κυλινδρική ράβδος και ένας ιμάντας.

Τοποθετήσαμε τους δύο κινητήρες (NEMA-17 size - 200 steps/rev, 12V 350mA), στα άκρα της μεγάλης διάστασης του ορθογωνίου, έτσι ώστε να μετραδίδουν κίνηση μέσω του ιμάντα, στον άξονα X.

Κάθετα στις μεγάλες κυλινδρικές ράβδους, στον άξονα Y, τοποθετήσαμε τις δύο μικρότερες κυλινδρικές ράβδους. Πάνω τους, προσαρμόσαμε τον τρίτο κινητήρα (NEMA-17 size - 200 steps/rev, 12V 350mA) και ένα ιμάντα που παίρνει κίνηση από αυτόν και μπορεί να κινήσει το υποσύστημα γραφής, στον άξονα Y.

Πάνω στις κυλινδρικές ράβδους του άξονα Y, τοποθετήθηκε ο κινητήρας (28BYJ 48 5VDC 15031801) (εικόνα 30), για να ελέγχει την κίνηση στον άξονα Z, δηλαδή το πάνω – κάτω της βάσης που προσαρμόσαμε για να δέχεται μολύβι, στυλό, ή μαρκαδόρο (γραφίδα), αυτό που ονομάσαμε νωρίτερα υποσύστημα γραφής.

4.3 Δοκιμές – έλεγχοι κατασκευής

Επιλέξαμε δύο κυλινδρικούς άξονες και δύο κινητήρες στον άξονα των X, έναν σε κάθε βραχύονα, γιατί στις δοκιμές που κάναμε με έναν, είδαμε ότι δεν μπορούσαμε να πετύχουμε ικανοποιητική ακρίβεια στον έλεγχο της συντεταγμένης X.

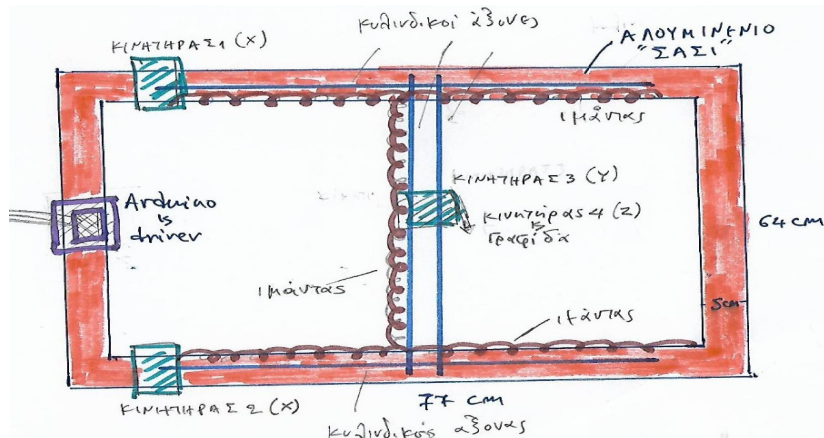
Επιλέξαμε δύο κυλινδρικούς άξονες στην διεύθυνση Y, για λόγους στατικών της κατασκευής μας, αλλά και για καλύτερο έλεγχο της συντεταγμένης Y.

Στον έλεγχο της κίνησης στον άξονα Z, θα μπορούσε να επιλεγεί και κινητήρας servo, αλλά προτιμήθηκε ο step motor 28BYJ 48 που λειτουργεί στα 5 Volt.

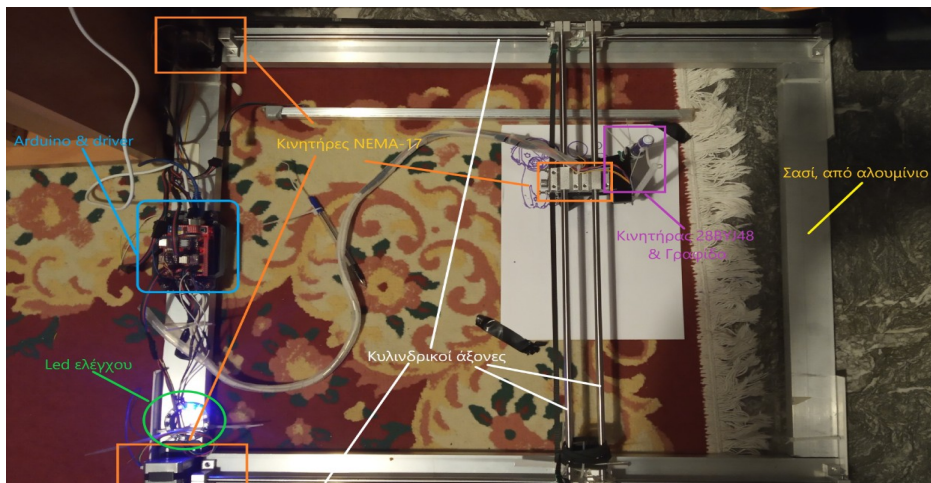
Θα πρέπει να αναφέρουμε ότι καταλήξαμε στην συγκεκριμένη υλοποίηση, αφού δοκιμάσαμε πολλές άλλες. Για παράδειγμα η πρώτη υλοποίηση έγινε με ξύλινο σασί, αλλά αποδείχθηκε με σοβαρά προβλήματα στην στατικότητα.

Μια άλλη σκέψη που εγκαταλήφθηκε ήταν να γίνει η όλη κατασκευή πάνω σε στιβαρή επίπεδη επιφάνεια. Αυτό θα δημιουργούσε σοβαρό πρόβλημα φορητότητας του εκτυπωτή, κάτι που δεν το θέλαμε.

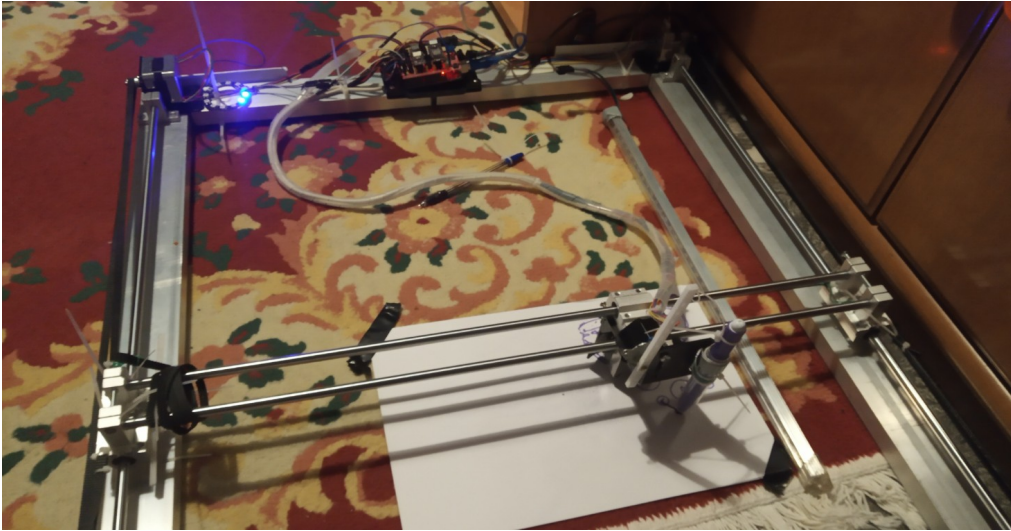
Ακολουθούν προσχέδιο και φωτογραφίες της τελικής υλοποίησης



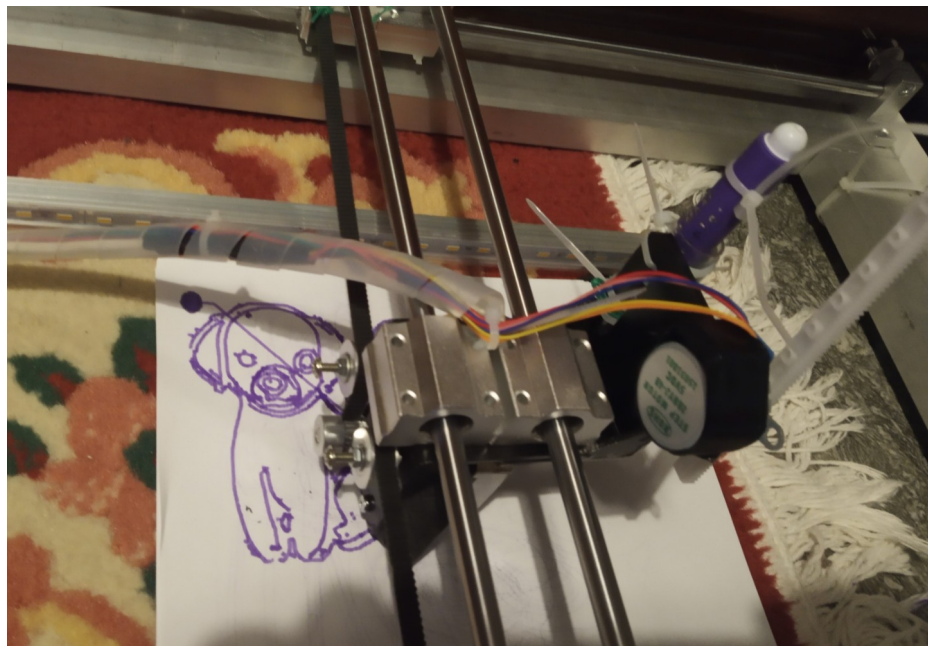
Εικόνα 31: Προσχέδιο κατασκευής



Εικόνα 32: Κάτοψη τελικής κατασκευής



Εικόνα 33: Ο άξονας Y, με τους δύο κινητήρες και τη γραφίδα σε πρώτο πλάνο



Εικόνα 34: Βραχύωνας γραφίδας

5 Λειτουργία εκτυπωτή

5.1 Περιγραφή λειτουργίας

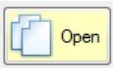
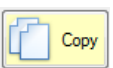
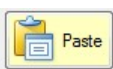



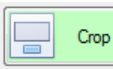

Χρησιμοποιήσαμε δυο περιβάλλοντα προγραμματισμού. Την Visual Basic και το περιβάλλον του Arduino IDE, που είναι περιβάλλον γλώσσας C.

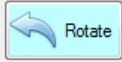




Το Software που δημιουργήσαμε, αποτελείται από δύο προγράμματα. Αυτό που βρίσκεται στο Arduino σε γλώσσα C και αυτό που βρίσκεται στον υπολογιστή, σε γλώσσα Visual Basic..

Αρχικά δημιουργούμε μια εικόνα ή ένα σκίτσο σε λογισμικό της αρεσκείας μας, ή αναζητούμε κάτι αντίστοιχο στο Internet, το οποίο θέλουμε στη συνέχεια να εκτυπώσουμε.

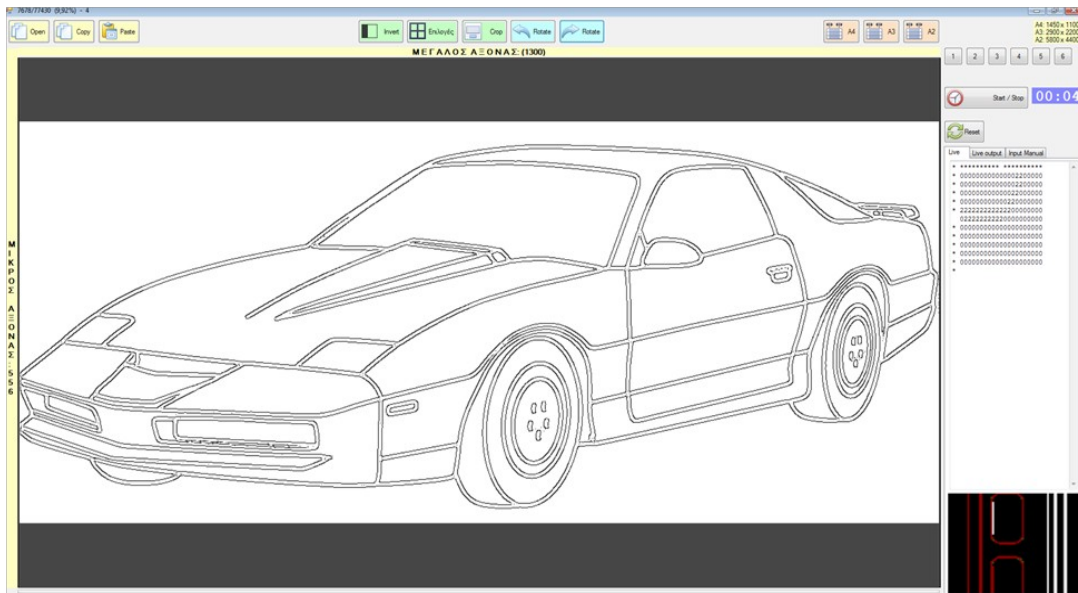
Το πρόγραμμά μας στον υπολογιστή, γραμμένο σε Visual Basic, διαβάζει τις εικόνες και τις μετατρέπει σε εντολές κινήσεις στα moter του εκτυπωτή. Το πρόγραμμα στο Arduino διαβάζει αυτές τις εντολές και τις μετατρέπει σε κίνηση στα moter στους 2 άξονες (X και Y) και τη γραφίδα πάνω ή κάτω (άξονας Z).

Ακολουθεί πίνακας με τις λειτουργίες του προγράμματός μας σε Visual Basic (Version 1).

	Ανοίγει μία εικόνα από τον δίσκο
	Αντιγράφει την εικόνα που φαίνεται στην οθόνη στην μνήμη του υπολογιστή για να μπορεί να την επεξεργαστεί με κάποιο άλλο πρόγραμμα
	Κάνει επικόλληση την εικόνα από την μνήμη στην οθόνη του προγράμματος
	Αλλάζει τα χρώματα από μαύρο σε άσπρο και το αντίθετο. ΠΡΕΠΕΙ το background να είμαι μαύρο για να δουλέψει σωστά το πρόγραμμα
	Δημιουργεί περιγράμματα όπου χρειάζεται η εικόνα. 
	Κόβει το περιττό κενό που υπάρχει γύρω από την εικόνα 

	Περιστρέφει την εικόνα προς τα αριστερά
	Περιστρέφει την εικόνα προς τα δεξιά
	Προσαρμόζει την εικόνα να χωράει στο μέγεθος A4
	Προσαρμόζει την εικόνα να χωράει στο μέγεθος A3
	Προσαρμόζει την εικόνα να χωράει στο μέγεθος A2

Στο **1ο βήμα**, το πρόγραμμά μας σε Visual Basic (εικόνα 35), παίρνει την εικόνα και ψάχνει να βρει την πιο κοντινή γραμμή. Στη συνέχεια φτιάχνει την “κίνηση” που έχει η γραμμή αυτή, δηλα-

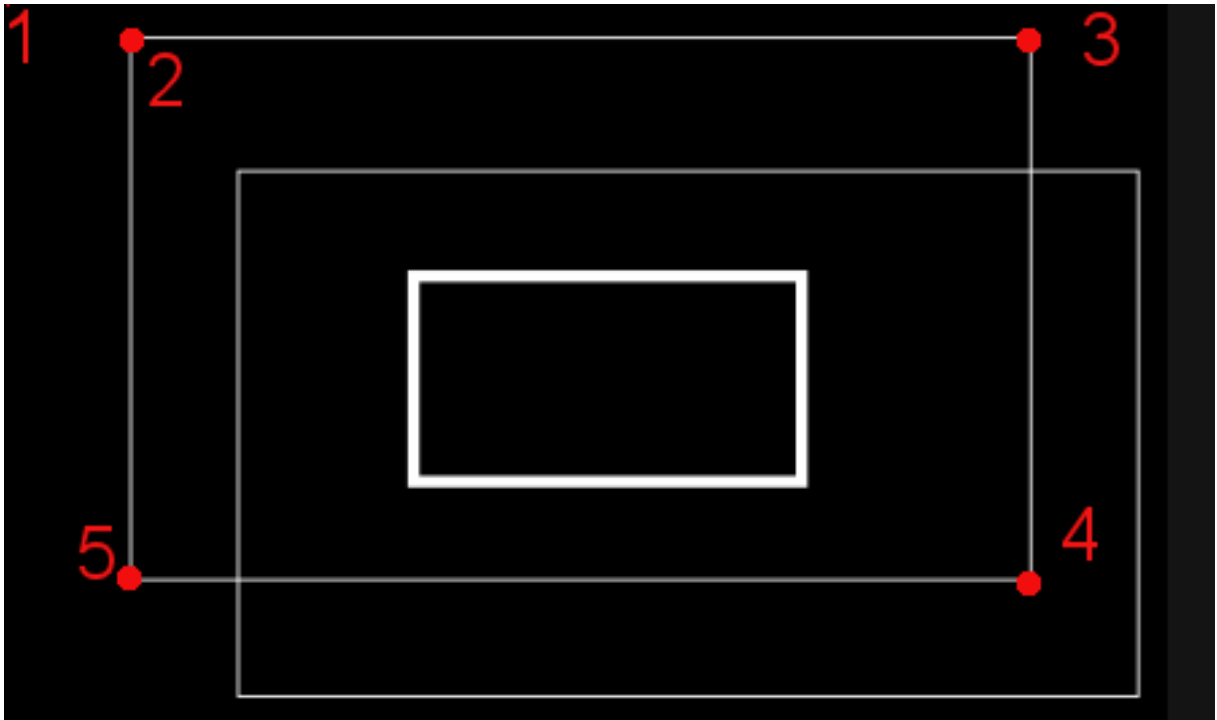


Εικόνα 35: Βήμα 1, επιλογή στίχου από τον κώδικα της Visual Basic
 δή καταγράφει τις συντεταγμένες των σημείων που αποτελούν την γραμμή.

Στο **2ο βήμα**, το πρόγραμμά μας σε Visual Basic, στέλνει στο Arduino τις προηγούμενες συντεταγμένες, όπου εκεί αναλαμβάνει ο κώδικας της C, να σχεδιάσει την γραμμή (Εικόνα 38).

Αναλυτικότερα ο τρόπος λειτουργίας του προγράμματος που δημιουργήσαμε, στον υπολογιστή σε Visual Basic, είναι ο παρακάτω:

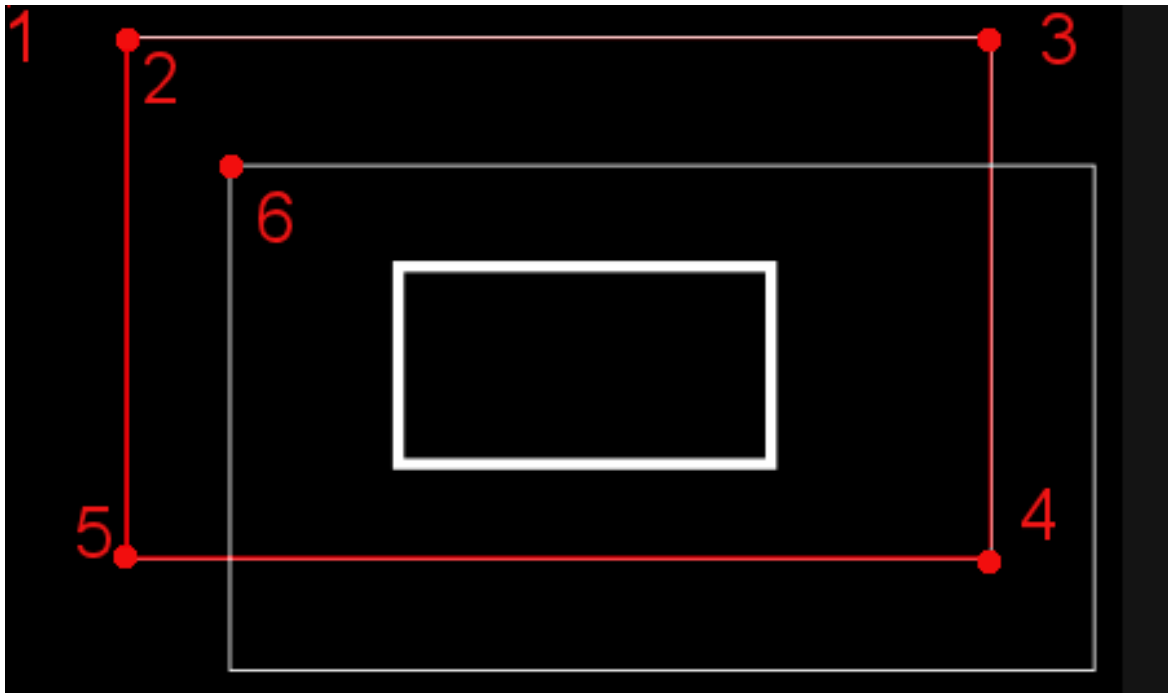
Έστω ότι έχουμε το πιο κάτω σχήμα προς εκτύπωση (εικόνα 36).



Εικόνα 36: Το προς εκτύπωση σχήμα

- A. Το πρόγραμμα ξεκινάει από το σημείο 1 με συντεταγμένες (0,0).
- B. Ψάχνοντας κυκλικά ψάχνει το πρώτο κοντινό σημείο που έχει χρώμα (άσπρο) και το βρίσκει στο σημείο 2. (έστω το σημείο 2 έχει συντεταγμένες (50,10)). Το πρόγραμμα δίνει την εντολή 'm50.10' στο Arduino. Αυτή η εντολή θα μετακινήσει τον άξονα X κατά 50 μονάδες και τον Y κατά 10 μονάδες. (μια μονάδα αντιστοιχεί 5 μοίρες περιστροφή στο moter)
- Γ. Στην συνέχεια δίνει εντολή να κατέβει η γραφίδα, γιατί από εδώ και περά θα γίνεται σχεδίαση.
- Δ. Το πρόγραμμα μετά κινείται δεξιά, αριστερά, πάνω ή κάτω και βρίσκει την πιο μεγάλη ευθεία. Στην περίπτωση μας θα πάει στην θέση 3 που έχει π.χ. συντεταγμένη (800.10). Άρα θα δώσει εντολή 'M750.0' (που σημαίνει ότι απλά θα κινηθεί προς την μία κατεύθυνση κατά 750 μονάδες).
- Ε. Στην συνέχεια επαναλαμβάνεται το βήμα Δ και θα φτάσει στα σημεία 4,5 και μετά στο 2.

Κάθε φορά που ολοκληρώνει κάποιο άσπρο κομμάτι και το μεταφέρει στο Arduino αυτό γίνεται κόκκινο στην οθόνη (Εικόνα 37).



Εικόνα 37: Το σχήμα 2-3-4-5, βρίσκεται στο Arduino

Αφού λοιπόν φτάσει πάλι στο σημείο 2 και δεν υπάρχει δίπλα κάποιο άλλο σημείο, τότε δίνει εντολή να σηκώσει τον στυλό. Στη συνέχεια, το πρόγραμμα ξεκινάει πάλι από το σημείο 2. (Θα πάει στο σημείο 6 και θα κάνει τα ίδια)

Όταν ολοκληρωθούν όλα τα σημεία, τότε το πρόγραμμα μεταφέρει το μοτέρ στην θέση (0,0) και σταματάει το πρόγραμμα.

Τα moter στο Arduino μπορούν να κινηθούν και χειροκίνητα από τον υπολογιστή (π.χ. για να το πάμε σε όποια θέση θέλουμε, για να ξεκινήσει από εκεί να ζωγραφίζει), πατώντας απλά τα βελάκια του υπολογιστή.

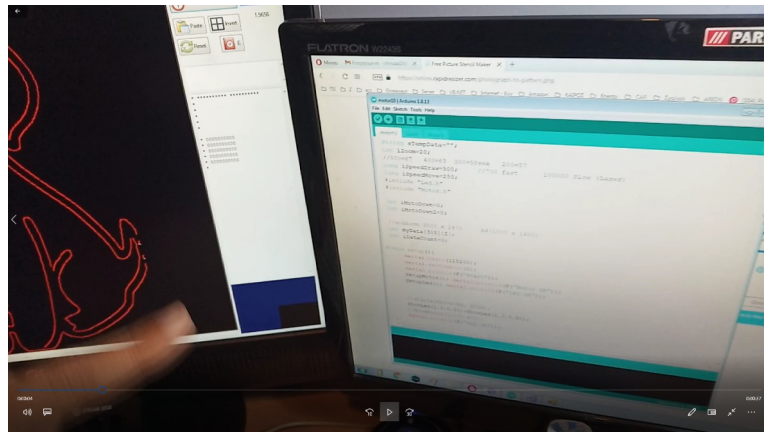
Αναλυτικότερα ο τρόπος λειτουργίας του προγράμματος στο Arduino είναι ο παρακάτω:

Το πρόγραμμα σε γλώσσα C, στο περιβάλλον του Arduino, διαβάζει μέσω συριακής σύνδεσης με τον υπολογιστή κάποιες εντολές και μπορεί έτσι να μετακινήσει τα moter κατά τον άξονα X, Y, Z.

Οι εντολές που δέχεται είναι οι εξής:

- `mx.y`: Μετακινεί τα moter στον άξονα X ή Y για x ή y μονάδες γρήγορα. π.χ. (m100.-30), μετακινεί τον άξονα X κατά 100 μονάδες και τον Y κατά -30 μονάδες.
- `Mx.y`: ίδια με την `mx.y` απλά μετακινεί τα moter πιο αργά. (αυτή η εντολή χρησιμοποιείται όταν είναι για ζωγράφισμα ενώ η `mx.y` όταν θέλουμε απλά να πάμε σε μια θέση).
- `z`: Μετακινεί τη γραφίδα κάτω.
- `Z`: Μετακινεί τη γραφίδα πάνω.
- `r`: Κάνει διαγραφή της μνήμης στο Arduino.
- `R`: Κάνει reset το Arduino.

Επίσης στο Arduino υπάρχει και ένας αναλογικός μοχλός ο οποίος μετακινεί τα moter χωρίς την χρήση σύνδεσης με υπολογιστή. Τον μοχλό όσο πιο πολύ τον πατάμε, τόσο πιο γρήγορα μετακινείται το αντίστοιχο μοτέρ. Στο Arduino υπάρχουν συνδεδεμένα και κάποια Led, τα οποία δείχνουν αν η γραφίδα είναι πάνω ή κάτω και με πόση ταχύτητα κινείται ο άξονας X και Y.

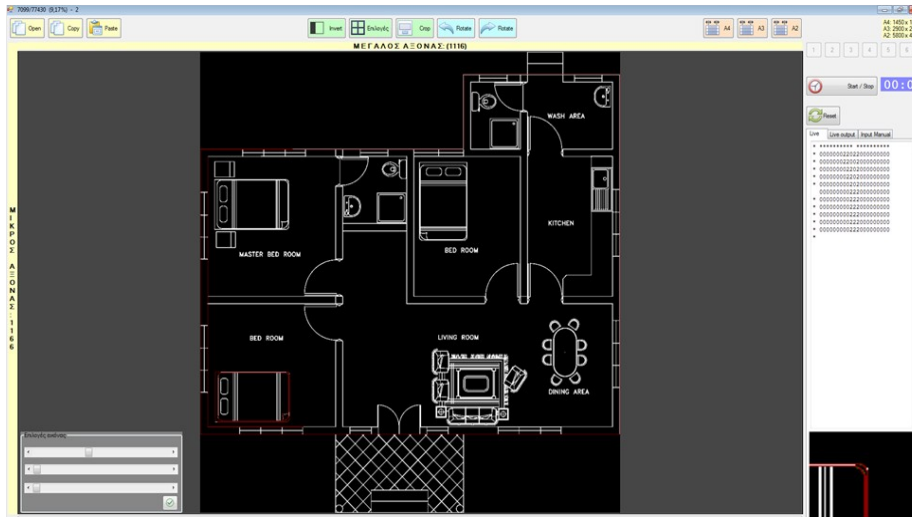


Εικόνα 38: Βήμα 2, έχει αναλάβει ο κώδικας της C στο Arduino

5.2 Πρόσθετες λειτουργίες

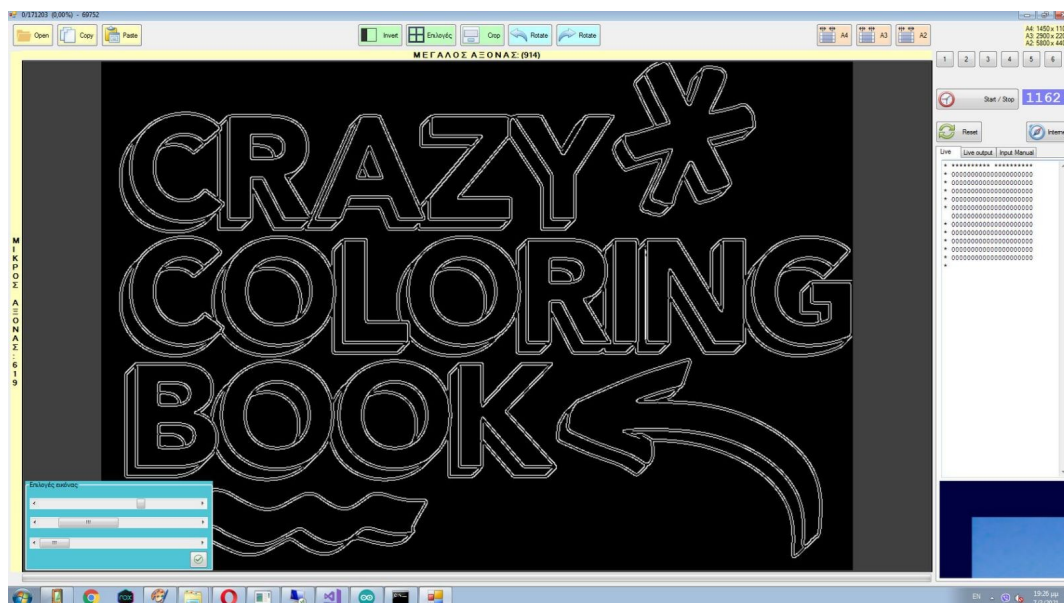
Παράλληλα τρέχουν και κάποιες βοηθητικές λειτουργίες. Ο κώδικας της Visual Basic, μας δείχνει την γραμμή που έχει εκτυπωθεί με κόκκινο χρώμα και την προς εκτύπωση με λευκό, ώστε να έχουμε τον έλεγχο της εκτύπωσης (εικόνα 39).

Στην κατασκευή έχουμε τοποθετήσει, έναν κυκλικό δίσκο με led, όπου μπορούμε να παρακολουθούμε εάν η γραφίδα είναι κάτω, άρα γράφει (κόκκινο χρώμα led). Εάν είναι στον αέρα, άρα δεν γράφει πράσινο χρώμα led και τέλος η ταχύτητα ενεργοποίησης των led, δίνει την ταχύτητα κίνησης της γραφίδας.



Εικόνα 39: Σχέδιο σε διαδικασία εκτύπωσης

Επιπρόσθετα έχουμε τοποθετήσει ένα χειριστήριο (joystick), με το οποίο μπορούμε να ελέγξουμε αναλογικά την κίνηση της γραφίδας. Πρακτικά δηλαδή ελέγχουμε την κίνηση της γραφίδας και στους τρεις άξονες μέσω joystick και μπορούμε να σχεδιάσουμε με το χέρι, οτιδήποτε. Στις εικόνες, 35, 38,39 και 40 φαίνεται σε λειτουργία, η εφαρμογή που δημιουργήσαμε, σε Visual Basic. Την αναβαθμίζουμε συνεχώς και ήδη βρίσκεται στην έκδοση 2, με προσθήκες στη λειτουργία της, αλλά και στη γραμμή εργαλείων (εικόνα 40).

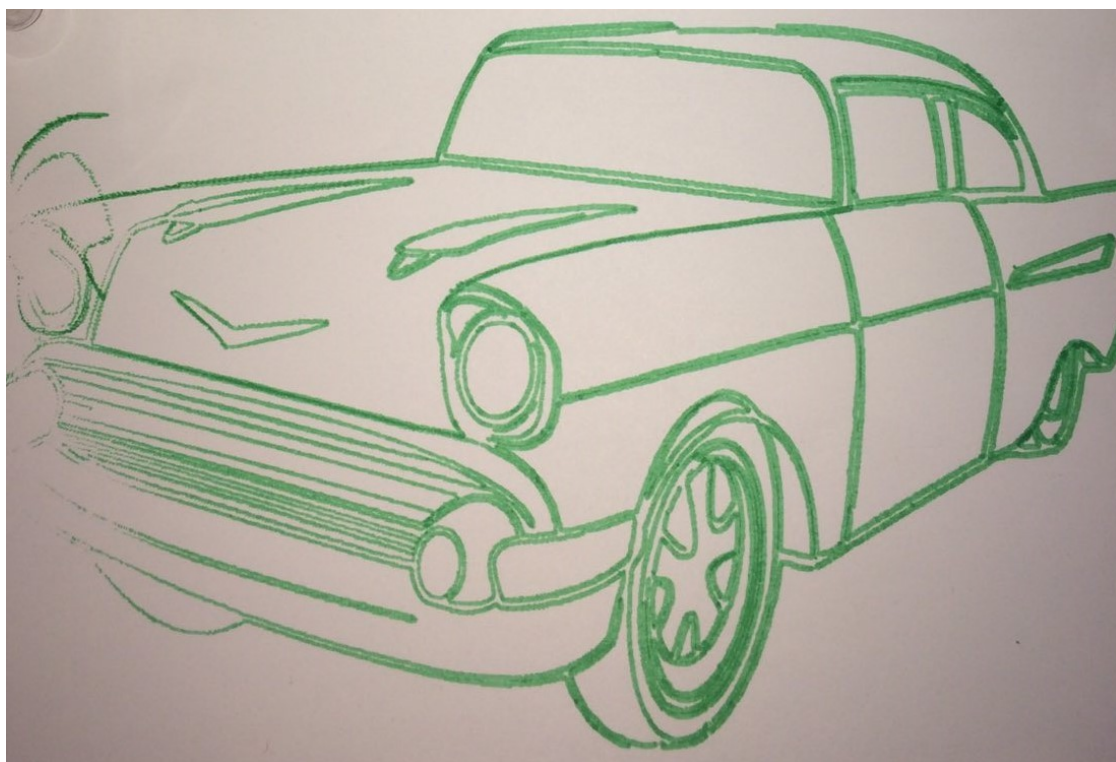


Εικόνα 40: Η εφαρμογή μας στην version 2

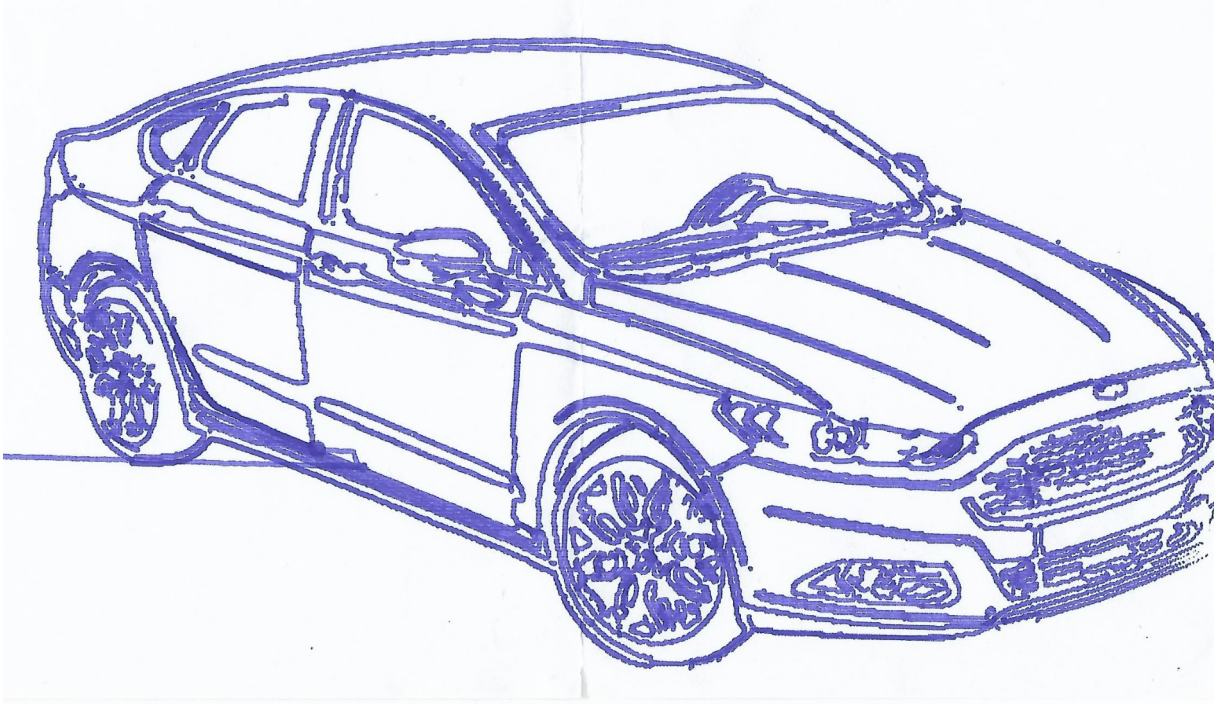
Ακολουθούν δοκιμαστικές εκτυπώσεις με τον εκτυπωτή μας:



Εικόνα 41: Δοκιμαστική εκτύπωση 1, με τον εκτυπωτή μας



Εικόνα 42: Δοκιμαστική εκτύπωση 2, με τον εκτυπωτή μας



Εικόνα 43: Δοκιμαστική εκτύπωση 3, με τον εκτυπωτή μας



Εικόνα 44: Δοκιμαστική εκτύπωση 4, με τον εκτυπωτή μας

6 Συμπεράσματα και προτάσεις περαιτέρω μελέτης ή ανάπτυξης

Ο στόχος του εγχειρήματος για την δημιουργία μιας κατασκευής, που ενώ έχει μικρό κόστος και φορητότητα, μπορεί να ανταποκριθεί με αξιοπιστία σε εκτυπώσεις διάστασης μέχρι και A2, είναι εφικτός με χρήση του μικροελεγκτή Arduino.

Η κατασκευή, έχει στοιχεία επεκτασιμότητας σε πολλά επίπεδα.

Μία πρώτη νέα εκδοχή θα μπορούσε να οδηγήσει σε εκτυπώσεις σε κατακόρυφο επίπεδο.

Μία δεύτερη, στην δημιουργία μιας αντίστοιχης κατασκευής, που αντί να σχεδιάζει σε χαρτί, να χαράσσει με χρήση laser, σε ξύλο. Πάνω σε αυτήν την εκδοχή, πειραματιστήκαμε και εμείς με επιτυχία, πάνω σε χαρτί.

Επίσης, σε μελλοντικό χρόνο μια αντίστοιχη κατασκευή με χρήση Arduino, θα μπορούσε χρησιμοποιώντας εργαλείο κοπής να κόβει σε προεπιλεγμένο σχήμα κομμάτια ξύλου, π.χ. να δημιουργεί τα κομμάτια για τις φιγούρες στον караγκιόζη, ή χρησιμοποιώντας εργαλείο διάτρησης, να ανοίγει τρύπες σε ξύλα, ώστε αυτά να συνενωθούν για την κατασκευή επίπλων.

7 Βιβλιογραφία

- [1] Brian W. Kernighan – Dennis M. Ritchie, 1990, Γλώσσα προγραμματισμού C, Εκδόσεις: Κλειδάριθμος
- [2] Harvey M. Deitel & Paul J. Deitel, 2012, Visual Basic 2010 Προγραμματισμός, Εκδόσεις: Μ. Γκιούρδας
- [3] Peter Aitken & Bradley L. Jones, 1997, Εγχειρίδιο της C, Εκδόσεις: Μ. Γκιούρδας
- [4] Βηματικοί Κινητήρες – Κατασκευή και Μεθοδολογία Ελέγχου, Παντής Γρηγόριος, Διπλωματική εργασία, Τ.Ε.Ι. Δυτικής Ελλάδας, Σ.ΤΕ.ΕΦ., 2017
- [5] Διερεύνηση λειτουργίας και κατασκευή οδήγησης βηματικού κινητήρα μέσω παράλληλης θύρας, Απόστολος Κ. Σεργιάδης, Διπλωματική εργασία, Ε.Μ.Π., 2009
- [6] Μελέτη και κατασκευή βηματικού κινητήρα , Κύργιος Λεωνίδας - Ανδρέας Χοντάης , Διπλωματική εργασία, Τμήμα Ηλεκτρολόγων Μηχανικών Τ.Ε.Ι. Πειραιά, 2017
- [7] Μελέτη και υλοποίηση συστήματος ελέγχου βηματικού κινητήρα με χρήση ARDUINO, Σιατής Χρήστος, Διπλωματική εργασία, Τμήμα Μηχανικών Πληροφορικής, Τ.Ε.Ι. Ηπείρου, 2014
- [8] Μηχατρονικά Συστήματα Ι, Δρ. Φασουλές Γιάννης, Εργαστήριο Συστημάτων Ελέγχου Ρομποτικής, Τμήμα Μηχανολόγων Μηχανικών Τ.Ε.Ι. Κρήτης
- [9] A Micro-Controller Controlled 3 Axis CNC Machine for Engraving and Designing, Tanzim Ahmed Khan - Mohammad Samiul Haque Samiu - Md. Shahabuddin Sani - Md. Mozammel Haque, BRAC University, Dhaka, Bangladesh, 2017

Παράρτημα Α: Διευθύνσεις internet

- [1] Official site: <https://www.arduino.cc/>, Οκτώβριος 2019
- [2] Βικιπαίδεια – Wikipedia, [https://en.wikipedia.org/wiki/C_\(programming_language\)](https://en.wikipedia.org/wiki/C_(programming_language)), Οκτώβριος 2019
- [3] Βικιπαίδεια – Wikipedia, <https://el.wikipedia.org/wiki/Arduino>, Οκτώβριος 2019
- [4] <https://ts.sch.gr/docs/odigies-egkatastasis-diaxisirisis/363-odigoι-gia-xrisi-arduino>, Οδηγός χρήσης Arduino, 17/8/2017, τροποποιήθηκε 18/6/2020, ΕΑΙΤΥ, Μ. Κομνηνού
- [5] Βικιπαίδεια – Wikipedia, https://en.wikipedia.org/wiki/Visual_Basic, Οκτώβριος 2019
- [6] <https://bit.ly/2rVFISP>, Προγραμματίζοντας με τον μικροελεγκτή Arduino, Εμμανουήλ Πουλάκης, Ηράκλειο Ιανουάριος 2015, [Ανακτήθηκε 1 Οκτωβρίου 2019]
- [7] Μαθήματα C++, <https://www.cprogramming.com/tutorial.html>, Νοέμβριος 2019
- [8] Free interactive C tutorial, <https://www.learn-c.org/>, Νοέμβριος 2019
- [9] C++ Tutorial, <https://www.w3schools.com/cpp/default.asp>, Νοέμβριος 2019
- [10] Visual Basic Tutorial, <https://www.vbtutor.net/index.php/visual-basic-2017-tutorial/>, Νοέμβριος 2019
- [11] Εκμάθηση Visual Basic, <https://www.homeandlearn.co.uk/NET/vbNet.html>, Νοέμβριος 2019
- [12] Ιστοσελίδα δημιουργίας και επεξεργασίας εικόνων, <https://online.rapidresizer.com/photograph-to-pattern.php>, Ιανουάριος 2020
- [13] Ιστοσελίδα με πληροφορίες για drivers βηματικών κινητήρων: <https://www.pololu.com/category/120/stepper-motor-driver>, Ιανουάριος 2020
- [14] DRV8825 Stepper Motor Controller IC datasheet, <https://www.ti.com/product/DRV8825>, έκδοση Ιούλιος 2014

Παράρτημα Β : Κώδικας Συστήματος και προγραμμάτων εγκατάστασης

B1 Κώδικας VB.NET

```
Dim bmpTemp As Bitmap
Dim bArray As Boolean = False
Dim iTimeSend As Double
Dim iCurX As Integer = 0, iCurY As Integer = 0
Dim picData(0, 0) As Byte, iHeight As Integer = 0, iWidth As Integer = 0,
iCountData As Long, iCurData As Long = 0, iTimer As Double
Dim myBitmap As Bitmap, bOKRead As Boolean
Dim WithEvents comArduino As IO.Ports.SerialPort
Dim iPaxos As Integer = 1
Dim iSpeed As Integer = 10
Dim xMouse As Integer, yMouse As Integer, iCountSend As Integer
Dim iClock As Double
```

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

```
'Φόρτωση των ρυθμίσεων από το αρχείο settings.txt (θύρα - και τρόπος αποστολής στο Arduino)
```

```
    Dim sCom As String = ""
    Using sr As New StreamReader("settings.txt")
        sCom = sr.ReadLine
        If Val(sr.ReadLine) = 1 Then bArray = True
    EndUsing
```

```
    Try
```

```
        Using sr As New StreamReader("settings.txt")
            comArduino = My.Computer.Ports.OpenSerialPort("COM"&
sCom, 115200)
        EndUsing
```

```

        comArduino.Close()
        comArduino.Open()
    Catch ex As Exception
        MsgBox("Πρόβλημα με την COM"& sCom)
    EndTry
    LoadPhoto(0)

```

```
EndSub
```

```
PrivateSub cmdStart_Click(sender As Object, e As EventArgs) Handles cmd-
Start.Click
```

```

    'Κουμπι Start / Stop
    If cmd1.Enabled = False Then
        cmd1.Enabled = True
        cmd2.Enabled = True
        cmd3.Enabled = True
        cmd4.Enabled = True
        cmd5.Enabled = True
        cmd6.Enabled = True
        ExitSub
    EndIf

    txtData.Text = ""
    ComputeImage()
    iClock = Microsoft.VisualBasic.Timer
    cmd1.Enabled = False
    cmd2.Enabled = False
    cmd3.Enabled = False
    cmd4.Enabled = False
    cmd5.Enabled = False
    cmd6.Enabled = False
    Dim x As Integer = 0, y As Integer = 0, b As Boolean = False, dx As-
Integer = 0, dy As Integer = 0
    iTimer = Microsoft.VisualBasic.Timer
    SendArduino("r")

    Do
        MoveSmart2(x, y, dx, dy)

```

```

    If iCountData - iCurData < 2 ThenExitDo
    ShowZoomPic(x, y)
    If dx <> 0 Or dy <> 0 Then SendArduino("m"& dx & "."& dy)
    If x < 0 Or y < 0 ThenExitDo

    SendArduino("z")
    GetLine(x, y)
    SendArduino("Z")

    For iy = 1 To iHeight - 2
        For ix = 1 To iWidth - 2
            If picData(ix - 1, iy) <> 1 And picData(ix, iy)
= 1 And picData(ix + 1, iy) <> 1 And
            picData(ix - 1, iy - 1) <> 1 And picData(ix, iy
- 1) <> 1 And picData(ix + 1, iy - 1) <> 1 And
            picData(ix - 1, iy + 1) <> 1 And picData(ix, iy
+ 1) <> 1 And picData(ix + 1, iy + 1) <> 1 Then
                PlotData(ix, iy, Color.Green)
            EndIf
        Next
    Next

    pic.Refresh()
    Application.DoEvents()

    If iCountData - iCurData < 2 ThenExitDo
    If cmd1.Enabled = TrueThenExitDo
Loop
SendArduino("m"& -x & "."& -y)
cmd1.Enabled = True
cmd2.Enabled = True
cmd3.Enabled = True
cmd4.Enabled = True
cmd5.Enabled = True
cmd6.Enabled = True
SendArduino("E")
SendArduino("R")

EndSub

```

```
PrivateSub Timer1_Tick(sender AsObject, e As EventArgs) Handles  
Timer1.Tick
```

```
    Timer1.Enabled = False  
    cmdStart_Click(Nothing, Nothing)
```

```
EndSub
```

```
PrivateSub MoveSmart2(ByRef x AsInteger, ByRef y AsInteger, ByRef dx  
AsInteger, ByRef dy AsInteger)
```

```
'Ρουτίνα που ψάχνει απο το σημείο X,Y κάποιο κοντινό που έχει χρώμα  
άσπρο για εκτύπωση
```

```
'Η ρουτίνα κάνει τετράγωνα γύρω γύρω απο το σημείο X,Y αυξάνοντας  
κάθε φορά την απόσταση μέχρι να βρεί κάποιο σημείο με χρώμα
```

```
    Dim b AsBoolean = False, d AsLong  
    Dim x0 AsInteger = x, y0 AsInteger = y, x1 AsInteger = x, y1 As-  
Integer = y  
    dx = 0 : dy = 0
```

```
    If picData(x, y) = 1 ThenExitSub
```

```
    d = iPaxos  
    If picData(x, y) = 1 ThenGoTo iExit
```

```
    Do
```

```
        y0 = y1 - d
```

```
        x0 = x1
```

```
        y = y0
```

```
        For x = x0 To x0 + d           'Δεξιά
```

```
            If x >= 0 And y >= 0 And x < iWidth And y < iHeight
```

```
ThenIf picData(x, y) = 1 ThenGoTo iExit
```

```
        Next
```

```
        x = x0 + d : x0 = x
```

```
        For y = y0 To y0 + d * 2       'Κάτω x2
```

```
            If x >= 0 And y >= 0 And x < iWidth And y < iHeight
```

```
ThenIf picData(x, y) = 1 ThenGoTo iExit
```



```

        Next
        y = y0 + d * 2 : y0 = y
        For x = x0 To x0 - d * 2 Step -1      'Αριστερά x2
            If x >= 0 And y >= 0 And x < iWidth And y < iHeight
ThenIf picData(x, y) = 1 ThenGoTo iExit
        Next
        x = x0 - d * 2 : x0 = x
        For y = y0 To y0 - d * 2 Step -1    'Πάνω x2
            If x >= 0 And y >= 0 And x < iWidth And y < iHeight
ThenIf picData(x, y) = 1 ThenGoTo iExit
        Next
        y = y0 - d * 2 : y0 = y
        For x = x0 To x0 + d                'Δεξιά
            If x >= 0 And y >= 0 And x < iWidth And y < iHeight
ThenIf picData(x, y) = 1 ThenGoTo iExit
        Next
        x = x0 + d
        d = d + 1
        If d > Width + iHeight ThenExitDo
    Loop
    x = x1
    y = y1

iExit:
    dx = x - x1
    dy = y - y1

EndSub

```

```

PrivateSub PlotData(x AsInteger, y AsInteger, Optional iColor As Color
= Nothing)

```

```

    'Κάθε φορά που βρίσκω κάποιο σημείο για εκτύπωση τότε το κάνω κόκ-
κίνο στην οθόνη
    For y0 = y - (iPaxos - 1) To y + (iPaxos - 1)
        If y0 >= 0 And y0 < iHeight Then
            For x0 = x - (iPaxos - 1) To x + (iPaxos - 1)
                If x0 >= 0 And x0 < iWidth Then

```

```

        If picData(x0, y0) = 1 Then
            iCurData = iCurData + 1

            If x = x0 And y = y0 Then
                picData(x0, y0) = 2
            Else
                picData(x0, y0) = 3
            EndIf
            If iColor = NothingThen iColor = Col-
or.DarkRed
            myBitmap.SetPixel(x0, y0, iColor)
        EndIf
    EndIf
Next
EndIf
Next
EndSub

```

```

PrivateFunction FindMaxRight(x AsInteger, y AsInteger) AsInteger

```

'Ρουτίνα που με βάση το σημείο Χ,Υ (που έχει άσπρο χρώμα) πάει Δεξιά μέχρι να βρεί μαύρο. Έτσι βρίσκω πόσο θα ζωγραφίσω

```

    FindMaxRight = 0
    IfpicData(x + iPaxos, y) = 1 Then
        x = x + iPaxos
        Do
            FindMaxRight = FindMaxRight + 1
            x = x + iPaxos
            If x >= iWidth ThenExitDo
            If picData(x, y) <> 1 ThenExitDo
        Loop
    EndIf
EndFunction

```

```

PrivateFunction FindMaxDown(x AsInteger, y AsInteger) AsInteger

```

'Ρουτίνα που με βάση το σημείο X,Y (που έχει άσπρο χρώμα) πάει Κάτω μέχρι να βρεί μαύρο

```
FindMaxDown = 0
If picData(x, y + iPaxos) = 1 Then
    y = y + iPaxos
    Do
        FindMaxDown = FindMaxDown + 1
        y = y + iPaxos
        If y = iHeight ThenExitDo
        If picData(x, y) <> 1 ThenExitDo
    Loop
EndIf
```

EndFunction

PrivateFunction FindMaxLeft(x AsInteger, y AsInteger) AsInteger

'Ρουτίνα που με βάση το σημείο X,Y (που έχει άσπρο χρώμα) πάει Αριστερά μέχρι να βρεί μαύρο

```
FindMaxLeft = 0
If x > 0 Then
    If picData(x - iPaxos, y) = 1 Then
        x = x - iPaxos
        Do
            FindMaxLeft = FindMaxLeft + 1
            x = x - iPaxos
            If x < 0 ThenExitDo
            If picData(x, y) <> 1 ThenExitDo
        Loop
    EndIf
EndIf
```

EndFunction

PrivateFunction FindMaxUp(x AsInteger, y AsInteger) AsInteger

'Ρουτίνα που με βάση το σημείο X,Y (που έχει άσπρο χρώμα) πάει Πάνω μέχρι να βρεί μαύρο

```
FindMaxUp = 0
If y > 0 Then
```

```

        If picData(x, y - iPaxos) = 1 Then
            y = y - iPaxos
            Do
                FindMaxUp = FindMaxUp + 1
                y = y - iPaxos
                If y < 0 ThenExitDo
                If picData(x, y) <> 1 ThenExitDo
            Loop

        EndIf
    EndIf

EndFunction

PrivateSub cmdPaste_Click(sender AsObject, e As EventArgs) Handles cmd-
Paste.Click

    'Κάνω επικόλληση την εικόνα απο το πρόχειρο στην οθόνη
    cmdPaste.Enabled = False
    LoadPhoto(-2)
    cmdPaste.Enabled = True

EndSub

PrivateSub GetLine(ByRef x0 AsInteger, ByRef y0 AsInteger)

    'Κεντρική ρουτίνα που ζωγραφίζει πηγαίνοντας Πάνω, Κάτω, Δεξιά, Αριστε-
    ρά
    'Κάθε φορά που τελειώνει μια ευθεία στέλνει την αντίστοιχει εντολή στο
    Arduino
    'Όταν βρεί μαύρο φεύγει απο την ρουτίνα

iRestart:
    If iCountData - iCurData < 2 ThenExitSub

    Dim x AsInteger = x0, y AsInteger = y0, d AsInteger = 0, a AsIn-
    teger

```

```

'*****ΣτυλόςΚΑΤΩ*****
If x < 0 Or y < 0 ThenExitSub
If picData(x, y) = 1 Then
    PlotData(x, y)
EndIf

Do
    If cmd1.Enabled = TrueThenExitDo

    a = 0
    Dim aR AsInteger = FindMaxRight(x, y)
    Dim aL AsInteger = FindMaxLeft(x, y)
    Dim aU AsInteger = FindMaxUp(x, y)
    Dim aD AsInteger = FindMaxDown(x, y)
    If aR > 0 Then a = 1
    If aL > aR Then a = 2
    If aU > aL Then a = 3
    If aD > aU Then a = 4
    If aR = 0 And aL = 0 And aU = 0 And aD = 0 Then
        Dim dx AsInteger = 0, dy AsInteger = 0, x1 AsInteger =
x, y1 AsInteger = y

        MoveSmart2(x1, y1, dx, dy)
        If iCountData - iCurData < 2 ThenExitSub
        If Math.Abs(dx) = 1 And Math.Abs(dy) = 1 Then
            Dim TempX AsInteger = x1, TempY AsInteger = y1,
nCount AsInteger = 0
            Do
                If TempX < 0 ThenExitDo
                If TempY < 0 ThenExitDo
                If picData(TempX, TempY) = 1 Then
                    PlotData(TempX, TempY)
                    nCount = nCount + 1
                Else
                    ExitDo
                EndIf
                TempX = TempX + dx
                TempY = TempY + dy
                If TempX < 0 ThenExitDo
                If TempY < 0 ThenExitDo

```

```

        Loop
        If nCount = 0 ThenExitDo
        If dx <> 0 Or dy <> 0 Then SendArduino("M"& dx *
nCount & "."& dy * nCount)
        x0 = x1 + dx * (nCount - 1) : y0 = y1 + dy *
(nCount - 1)

        GoTo iRestart
    EndIf
    x0 = x
    y0 = y
    '*****Στυλός ΠΑΝΩ*****
    ExitSub
EndIf
If a = 0 ThenStop

Dim bFound AsBoolean = False
'Μετακίνηση δεξιά
If picData(x + iPaxos, y) = 1 And a = 1 Then
    x = x + iPaxos
    bFound = True
    d = 0
    Do
        PlotData(x, y)
        d = d + iPaxos
        x = x + iPaxos
        If x = iWidth ThenExitDo
        If picData(x, y) <> 1 ThenExitDo
    Loop
    x = x - iPaxos
    If d > 0 Then
        If d > 0 Then
            SendArduino("M"& Math.Abs(d) & ".0")
        Else
            SendArduino("M-"& Math.Abs(d) & ".0")
        EndIf
        GoTo iNext
    EndIf
EndIf
EndIf

```

```

'Μετακίνηση αριστερά
'a = 2
If x > 0 Then
    If picData(x - iPaxos, y) = 1 And a = 2 Then
        x = x - iPaxos
        bFound = True
        d = 0
        Do
            PlotData(x, y)
            d = d + iPaxos
            x = x - iPaxos
            If x < 0 ThenExitDo
            If picData(x, y) <> 1 ThenExitDo
        Loop
        x = x + iPaxos
        If d > 0 Then
            If d > 0 Then
                SendArduino("M-" & Math.Abs(d) & ".0")
            Else
                SendArduino("M" & Math.Abs(d) & ".0")
            EndIf
            GoTo iNext
        EndIf
    EndIf
EndIf

```

```

'Μετακίνηση επάνω
'a = 3
If y > 0 Then
    If picData(x, y - iPaxos) = 1 And a = 3 Then
        y = y - iPaxos
        bFound = True
        d = 0
        Do
            PlotData(x, y)
            d = d + iPaxos
            y = y - iPaxos
            If y < 0 ThenExitDo
            If picData(x, y) <> 1 ThenExitDo
        Do

```

```

        Loop
        y = y + 1
        If d > 0 Then
            If d > 0 Then
                SendArduino("M0."& Math.Abs(d))
            Else
                SendArduino("M0."& Math.Abs(d))
            EndIf
            GoTo iNext
        EndIf
    EndIf
EndIf

```

'Μετακίνηση κάτω

'a = 4

```

If picData(x, y + iPaxos) = 1 And a = 4 Then

```

```

    y = y + iPaxos

```

```

    bFound = True

```

```

    d = 0

```

```

    Do

```

```

        PlotData(x, y)

```

```

        d = d + iPaxos

```

```

        y = y + iPaxos

```

```

        If y = iHeight ThenExitDo

```

```

        If picData(x, y) <> 1 ThenExitDo

```

```

    Loop

```

```

    y = y - iPaxos

```

```

    If d > 0 Then

```

```

        If d > 0 Then

```

```

            SendArduino("M0."& Math.Abs(d))

```

```

        Else

```

```

            SendArduino("M0."& Math.Abs(d))

```

```

        EndIf

```

```

        GoTo iNext

```

```

    EndIf

```

```

EndIf

```

iNext:

```

ShowZoomPic(x, y)

```

```

If bFound = FalseThenExitDo

```



```

Loop
x0 = x
y0 = y

```

```
EndSub
```

```
PrivateSub cmdExecute_Click(sender As Object, e As EventArgs) Handles
cmdExecute.Click
```

'Εδώ μπορούμε να ανοίξουμε ένα αρχείο με εντολές (που δημιουργείται όταν εκτυπώνω) και να το τυπώσω χωρίς να έχω την εικόνα

```

txtData.Text = ""
cmdExecute.Enabled = False
Dim bmp AsNew Bitmap(pic.Image)
Dim s AsString = txtInput.Text
Dim x AsInteger = 0, y AsInteger = 0
For i = 0 To Split(s, vbCrLf).Count - 1
    Dim sCom AsString = Split(s, vbCrLf)(i)
    SendArduino(sCom)
    SelectCase Microsoft.VisualBasic.Left(sCom, 1)
        Case"m"
            x = x + Val(Split(Mid(sCom, 2), ".")(0))
            y = y + Val(Split(Mid(sCom, 2), ".")(1))
        Case"M"
            Dim xx AsInteger = Val(Split(Mid(sCom, 2), ".")
(0)), yy AsInteger = Val(Split(Mid(sCom, 2), ".")(1))
            Do
                bmp.SetPixel(x, y, Color.Red)
                If xx > 0 Then x = x + 1 : xx = xx - 1
                If yy > 0 Then y = y + 1 : yy = yy - 1
                If xx < 0 Then x = x - 1 : xx = xx + 1
                If yy < 0 Then y = y - 1 : yy = yy + 1
                bmp.SetPixel(x, y, Color.Red)
                pic.Image = bmp
                If xx = 0 And yy = 0 ThenExitDo
            Loop
        Case"z"
            If Int(Rnd() * 10) > 8 Then pic.Refresh()
        Case"Z"

```

```

        'pic.Refresh()
    EndSelect

Next
cmdExecute.Enabled = True

EndSub

PrivateSub cmdReset_Click(sender As Object, e As EventArgs) Handles cmdReset.Click

    'Κάνω RESET το Arduino
    Try
        iClock = Microsoft.VisualBasic.Timer
        cmdReset.Enabled = False
        SendArduino("R")
        cmdReset.Enabled = True
        comArduino.Close()
        comArduino.Open()
    Catch ex As Exception

    EndTry

EndSub

PrivateSub ShowZoomPic(x As Integer, y As Integer)
    'Ρουτίνα που μεγενθύνει το κομμάτι που τυπώνεται τώρα και το δείχνει στην οθόνη
    'Επίσης εμφανίζει και την εικόνα σε μορφή πίνακα (με τιμές 0,1,2,3)

    Dim destinationbmp AsNew Bitmap(picZoom.Width, picZoom.Height)
    Dim gr As Graphics = Graphics.FromImage(destinationbmp)

    Dim d As Integer = 40
    Dim selectionrectangle AsNew Rectangle(x - d, y - d, d * 2, d * 2)
    Dim destinationrectangle AsNew Rectangle(0, 0, picZoom.Width, picZoom.Height)

```

```
gr.DrawImage(pic.Image, destinationrectangle, selectionrectangle,  
GraphicsUnit.Pixel)
```

```
picZoom.Image = destinationbmp
```

```
Dim s AsString = "* ***** *" & vbCrLf & "* ", a AsIn-  
teger, b AsInteger
```

```
For a = y - 5 To y + 5
```

```
For b = x - 10 To x + 10
```

```
'If picData(b, a) <> 0 Then Stop
```

```
Try
```

```
    If b >= 0 And a >= 0 And a < iHeight And b <  
iWidth Then s = s & picData(b, a) Else s = s & " "
```

```
Catch ex As Exception
```

```
s = s & " "
```

```
EndTry
```

```
Next
```

```
If a = y - 1 Then
```

```
s = s & vbCrLf & " "
```

```
Else
```

```
s = s & vbCrLf & "* "
```

```
EndIf
```

```
Next
```

```
txtTable.Text = s
```

```
txtTable.Refresh()
```

```
EndSub
```

```
PrivateSub cmdInvert_Click(sender AsObject, e As EventArgs) Handles  
cmdInvert.Click
```

```
'Κάνει αναστροφή χρωμάτων της εικόνας
```

```
cmdInvert.Enabled = False
```

```
Dim bmp AsNew Bitmap(pic.Image)
```

```
    If bmp.PixelFormat <> Imaging.PixelFormat.Format24bppRgb Then
        bmp = AForge.Imaging.Image.Clone(bmp, System.Drawing.Imaging.PixelFormat.Format24bppRgb)
    EndIf
```

```
    bmp =
AForge.Imaging.Filters.Grayscale.CommonAlgorithms.RMY.Apply(bmp)
```

```
    Dim filter AsNew AForge.Imaging.Filters.Invert
    pic.Image = filter.Apply(bmp)
```

```
    cmdInvert.Enabled = True
```

```
EndSub
```

```
PrivateSub cmdRotateR_Click(sender AsObject, e As EventArgs) Handles
cmdRotateR.Click
```

```
    'Περιστρέφει την εικόνα προς τα δεξιά
    cmdRotateR.Enabled = False
    Dim bmp AsNew Bitmap(pic.Image)
    bmp.RotateFlip(RotateFlipType.Rotate90FlipNone)
    pic.Image = bmp
    ShowImageData()
    cmdRotateR.Enabled = True
```

```
EndSub
```

```
PrivateSub cmdRotateL_Click(sender AsObject, e As EventArgs) Handles
cmdRotateL.Click
```

```
    'Περιστρέφει την εικόνα προς τα αριστερά
    cmdRotateL.Enabled = False
    Dim bmp AsNew Bitmap(pic.Image)
    bmp.RotateFlip(RotateFlipType.Rotate270FlipNone)
    pic.Image = bmp
    ShowImageData()
    cmdRotateL.Enabled = True
```

EndSub

```
PrivateSub cmdEdge_Click(sender As Object, e As EventArgs) Handles cmd-Edge.Click
```

```
    bmpTemp = pic.Image  
    GroupImage.Visible = True
```

EndSub

```
PrivateSub cmdCopy_Click(sender As Object, e As EventArgs) Handles cmd-Copy.Click
```

```
    'Αντιγράφω την εικόνα στο πρόχειρο  
    Clipboard.SetImage(pic.Image)
```

EndSub

```
PrivateSub cmdAutoCrop_Click(sender As Object, e As EventArgs) Handles cmdAutoCrop.Click
```

```
    'Ρουτίνα που σβήνει το περιτό κενό γύρω απο την εικόνα και  
    φτιάχνει το background της μαύρο
```

```
    cmdAutoCrop.Enabled = False  
    Dim bmp As Bitmap = pic.Image.Clone  
    Dim bmp2 As Bitmap = bmp.Clone
```

```
    If bmp.GetPixel(0, 0).R > 250 And bmp.GetPixel(0, 0).G > 250 And  
    bmp.GetPixel(0, 0).B > 250 Then
```

```
        bmp = AForge.Imaging.Image.Clone(bmp, System.Drawing.Imag-  
ing.PixelFormat.Format24bppRgb)
```

```
        Dim filter3 AsNew AForge.Imaging.Filters.Invert  
        bmp = filter3.Apply(bmp)  
        bmp2 = bmp.Clone
```

```
    EndIf
```

```
    If bmp.PixelFormat <> Imaging.PixelFormat.Format32bppRgb Then  
        bmp = AForge.Imaging.Image.Clone(bmp, System.Drawing.Imag-  
ing.PixelFormat.Format32bppRgb)
```

```
    EndIf
```

```

    bmp =
AForge.Imaging.Filters.Grayscale.CommonAlgorithms.RMY.Apply(bmp)
    Dim filter2 AsNew AForge.Imaging.Filters.Threshold(100)
    bmp = filter2.Apply(bmp)

    Dim blob AsNew AForge.Imaging.BlobCounter
    blob.ProcessImage(bmp)
    Dim rects As Rectangle() = blob.GetObjectsRectangles()
    cmdExecute.Enabled = True
    Dim x1 AsInteger = -1, y1 AsInteger = -1, x2 AsInteger = -1, y2
AsInteger = -1
    For i = 0 To rects.Count - 1
        If rects(i).X < x1 Or x1 = -1 Then x1 = rects(i).X
        If rects(i).Y < y1 Or y1 = -1 Then y1 = rects(i).Y
        If rects(i).Right > x2 Or x2 = -1 Then x2 = rects(i).Right
        If rects(i).Bottom > y2 Or y2 = -1 Then y2 = rects(i).Bottom
    Next

    pic.Image = bmp2.Clone(New Rectangle(x1, y1, x2 - x1, y2 - y1),
System.Drawing.Imaging.PixelFormat.Format24bppRgb)
    bmp2.Dispose()
    bmp.Dispose()
    ShowImageData()
    cmdAutoCrop.Enabled = True
    GroupImage.Visible = False

EndSub

PrivateSub cmdOpen_Click(sender AsObject, e As EventArgs) Handles cm-
dOpen.Click

    'Άνοιγμα εικόνας
    OpenFileDialog1.Filter = ""
    OpenFileDialog1.FileName = ""
    Dim sFile As DialogResult = OpenFileDialog1.ShowDialog()
    If sFile = Windows.Forms.DialogResult.OK Then
        pic.Image = Image.FromFile(OpenFileDialog1.FileName)
        ComputeImage()
    EndIf

```

EndSub

```
PrivateSub cmdA4_Click(sender As Object, e As EventArgs) Handles  
cmdA4.Click
```

```
    'Κάνω Resize την εικόνα για A4. Η εικόνα γυρνάει αυτόματα σε κάθε-  
τη ή οριζόντια μορφή
```

```
    cmdA4.Enabled = False  
    ResizePic(1450, 1100)  
    cmdA4.Enabled = True
```

EndSub

```
PrivateSub cmdA3_Click(sender As Object, e As EventArgs) Handles  
cmdA3.Click
```

```
    'Κάνω Resize την εικόνα για A3. Η εικόνα γυρνάει αυτόματα σε κάθε-  
τη ή οριζόντια μορφή
```

```
    cmdA3.Enabled = False  
    ResizePic(1450 * 2, 1100 * 2)  
    cmdA3.Enabled = True
```

EndSub

```
PrivateSub cmdA2_Click(sender As Object, e As EventArgs) Handles  
cmdA2.Click
```

```
    'Κάνω Resize την εικόνα για A2. Η εικόνα γυρνάει αυτόματα σε κάθε-  
τη ή οριζόντια μορφή
```

```
    cmdA2.Enabled = False  
    ResizePic(1450 * 4, 1100 * 4)  
    cmdA2.Enabled = True
```

EndSub

```
PrivateSub ResizePic(iWidth As Integer, iHeight As Integer)
```

```
    'Ρουτίνα που κάνει Resize την εικόνα.  
    iWidth = iWidth - 40  
    iHeight = iHeight - 40
```

```

Dim dxNew AsInteger, dyNew AsInteger
If pic.Image.Width > pic.Image.Height Then
    dxNew = iWidth
    dyNew = dxNew * (pic.Image.Height / pic.Image.Width)
Else
    Dim iH AsInteger = iHeight
    iHeight = iWidth
    iWidth = iH

    dyNew = iHeight
    dxNew = dyNew * (pic.Image.Width / pic.Image.Height)
EndIf
Dim bmp AsNew Bitmap(iWidth, iHeight)
Dim gr As Graphics = Graphics.FromImage(bmp)

Dim selectionrectangle AsNew Rectangle(0, 0, pic.Image.Width,
pic.Image.Height)
Dim destinationrectangle AsNew Rectangle(0, 0, dxNew, dyNew)
gr.DrawImage(pic.Image, destinationrectangle, selectionrectangle,
GraphicsUnit.Pixel)
pic.Image = bmp
pic.Refresh()
ShowImageData()

EndSub

```

```

PrivateSub LoadPhoto(iType AsInteger)

```

```

'Ρουτίνα που ανοίγει μια απο τις έτοιμες εικόνες απο τον δίσκο
Dim bmp As Bitmap
If iType = -2 Then
    pic.Image = Clipboard.GetImage()
    If pic.Image IsNothingThenExitSub
Else
    bmp = Image.FromFile(CurDir() &"\"& iType &".bmp")

```



```

        If bmp.PixelFormat <> Imaging.PixelFormat.Format24bppRgb
Then
            bmp = AForge.Imaging.Image.Clone(bmp, System.Draw-
ing.Imaging.PixelFormat.Format24bppRgb)
        EndIf

        Dim filter AsNew AForge.Imaging.Filters.Grayscale(0.2125,
0.7154, 0.0721)
        bmp = filter.Apply(bmp)

        pic.Image = bmp

    EndIf
    ShowImageData()
    GroupImage.Visible = False

EndSub

PrivateSub HScrollBar1_Scroll(sender AsObject, e As ScrollEventArgs)
Handles HScrollBar1.Scroll

    'Φτιάχνει κάποια φίλτρα στην εικόνα
    Dim bmp AsNew Bitmap(bmpTemp)
    bmp =
AForge.Imaging.Filters.Grayscale.CommonAlgorithms.RMY.Apply(bmp)

    If HScrollBar2.Value > 0 Then
        Dim filter AsNew AForge.Imaging.Filters.CannyEdgeDetector
        filter.GaussianSize = HScrollBar2.Value
        'filter.ApplyInPlace(bmp)
        bmp = filter.Apply(bmp)
    EndIf

    If HScrollBar1.Value > 0 Then
        Dim filter2 AsNew AForge.Imaging.Filters.Threshold(HScroll-
Bar1.Value)
        bmp = filter2.Apply(bmp)
    EndIf

    If HScrollBar3.Value > 0 Then

```

```

        Dim filter4 AsNew AForge.Imaging.Filters.Sharpen
        filter4.DynamicDivisorForEdges = HScrollBar3.Value
        bmp = filter4.Apply(bmp)
    EndIf

    pic.Image = bmp

EndSub

PrivateSub HScrollBar2_Scroll(sender AsObject, e As ScrollEventArgs)
Handles HScrollBar2.Scroll
    'Φτιάχνει κάποια φίλτρα στην εικόνα
    HScrollBar1_Scroll(Nothing, Nothing)
EndSub

PrivateSub HScrollBar3_Scroll(sender AsObject, e As ScrollEventArgs)
Handles HScrollBar3.Scroll
    'Φτιάχνει κάποια φίλτρα στην εικόνα
    HScrollBar1_Scroll(Nothing, Nothing)
EndSub

PrivateSub Button2_Click(sender AsObject, e As EventArgs) Handles But-
ton2.Click
    GroupImage.Visible = False
EndSub

PrivateSub ShowImageData()
    lblX.Text = "Μ Ε Γ Α Λ Ο Σ Α Ξ Ο Ν Α Σ: (&pic.Image.Width&)"
    lblY.Text = "Μ Ι Κ Ρ Ο Σ Α Ξ Ο Ν Α Σ: &pic.Image.Height
EndSub

PrivateSubComputeImage()
'Μεταφέρει την εικόνα στην μνήμη σε εναν δισδιάστατο πίνακα.
'Αν το χρώμα είναι απο γκρι και πάνω το κάνει 1 οτιδήποτε άλλο το κάνει
μαύρο
'Όταν ένα σημείο δίνεται για εκτύπωση στην οθόνη γίνεται κόκκινο και
στο πίνακα πέρνει την τιμή 2
'Όταν βρει κάποιο ορφανό σημείο στην οθόνη γίνεται πράσινο και στο πί-
νακα πέρνει την τιμή 3

```

'Ορφανά είναι τα σημεία που στην μέση υπάρχει χρώμα και γύρω γύρω είναι κενό.

```
myBitmap = New Bitmap(pic.Image.Width, pic.Image.Height)

ShowImageData()

cmdStart.Enabled = False
Dim gr As Graphics = Graphics.FromImage(myBitmap)
Dim selectionrectangle AsNew Rectangle(0, 0, pic.Image.Width,
pic.Image.Height)
Dim destinationrectangle AsNew Rectangle(0, 0, pic.Image.Width,
pic.Image.Height)
gr.DrawImage(pic.Image, destinationrectangle, selectionrectangle,
GraphicsUnit.Pixel)

ReDim picData(myBitmap.Width + 1, myBitmap.Height + 1)

iHeight = myBitmap.Height
iWidth = myBitmap.Width

iCountData = 0
iCurData = 0
For y = 0 To iHeight - 1
    For x = 0 To iWidth - 1
        If myBitmap.GetPixel(x, y).R > 150 And myBitmap.Get-
Pixel(x, y).G > 150 And myBitmap.GetPixel(x, y).B > 150 Then
            picData(x, y) = 1
            iCountData = iCountData + 1
        Else
            picData(x, y) = 0
        EndIf
    Next
Next
ShowZoomPic(20, 10)
cmdStart.Enabled = True

EndSub
```

```

PrivateSub SendArduino(s AsString)
'Rουτίνα που πέρνει την εντολή (πχ M10.-35) και την στέλνει στο Arduino
και περιμένει OK για να πάει στην επόμενη
'Sτην περίπτωση που έχουμε επιλέξει μη σειριακή αποστολή στοιχείων τότε
οι εντολές αποθηκεύονται στο Arduino μέχρι
'μέχρι να γίνουν 300. Τότε στέλνει την εντολή E και περιμένει μέχρι να
εκτελεστούν
    ShowClock()
    Text = iCurData & "/" & iCountData & " (" & Format(100 * iCurData /
iCountData, "0.00") & "%) - " & Format(Microsoft.VisualBasic.Timer -
iTimer, "0")

    If Microsoft.VisualBasic.Left(s, 1) = "L" Then iCurX = iCurX -
Val(Mid(s, 2))
    If Microsoft.VisualBasic.Left(s, 1) = "R" Then iCurX = iCurX +
Val(Mid(s, 2))

    If bArray = True Then
        icountSend = icountSend + 1
        Bara.Value = icountSend
        Bara.Refresh()
    EndIf

    txtData.Text = txtData.Text & vbCrLf & s
    If comArduino IsNothing Then
        picZoom.Refresh()
        pic.Image = myBitmap
        'pic.Refresh()
        ExitSub
    EndIf
    'txtOut.Text = txtOut.Text & "@" & s
    'Exit Sub
    If bArray = True Then
        s = Replace(s, "M", "A")
        s = Replace(s, "m", "A")
        s = Replace(s, "z", "@")
        s = Replace(s, "Z", "@")
    EndIf

```

```

s = Replace(s, "N", "m")
Try
    bOKRead = False
    iTimeSend = Microsoft.VisualBasic.Timer
    comArduino.WriteLine(s)
Catch ex As Exception

EndTry

Dim iTimer2 As Double = Microsoft.VisualBasic.Timer
Do
    If bOKRead = True Then Exit Do
    Application.DoEvents()
    If Microsoft.VisualBasic.Timer - iTimer2 > 10 Then Exit Do
Loop
If Microsoft.VisualBasic.Timer - iTimer2 > 10 Then Exit Sub

If bArray = False Then
    picZoom.Refresh()
    '
    pic.Image = myBitmap
    GoTo iNext
EndIf

s = "E"
If s = "E" Then
    Do
        If bOKRead = True Then Exit Do
        Application.DoEvents()
    Loop
Else
    'icountSend = icountSend + 1
    'lblCount.Text = "Count: " & icountSend
EndIf

If icountSend > 299 Then
    pic.Image = myBitmap
    bOKRead = False
    picZoom.Refresh()
    pic.Refresh()

```

```

        comArduino.WriteLine("E")
        icountSend = 0
        Do
            If bOKRead = True Then ExitDo
            Application.DoEvents()
        Loop
        ExitSub
    EndIf
iNext:
    bOKRead = False
    lblTimer.Text = Format(Microsoft.VisualBasic.Timer - iTimeSend,
"0.0000")
EndSub

PrivateSub comArduino_DataReceived(sender As Object, e As SerialDataRe-
ceivedEventArgs) Handles comArduino.DataReceived

'Eδω διαβάζουμε ότι στοιχεία έρχονται απο το Arduino όπως πχ το OK
    Do
        Dim sIn As String = comArduino.ReadLine
        If Microsoft.VisualBasic.Left(sIn, 1) = "@" Then
            Invoke(Sub() Bara.Value = Val(Mid(sIn, 2)))
            Invoke(Sub() Bara.Refresh())
            Invoke(Sub() Bara.Refresh())
            Invoke(Sub() Bara.Refresh())
        EndIf
        If Microsoft.VisualBasic.Left(sIn, 2) = "OK" Then
            bOKRead = True
        Else
            'Dim s As String = sIn & vbCrLf & txtArduino.Text
            'Me.BeginInvoke(Sub() txtArduino.Text = s)
            'Debug.Print(sIn)
        EndIf
        If comArduino.BytesToRead = 0 Then ExitDo
    Loop

EndSub

```

```

PrivateSub Form1_KeyDown(sender AsObject, e As KeyEventArgs) HandlesMe.KeyDown

    'Μπορούμε να μετακινήσουμε τα moter και με τα βελάκια απο τον υπο-
    λογιστή
    If e.KeyCode = Keys.Add Then iSpeed = iSpeed + 1
    If e.KeyCode = Keys.Subtract And iSpeed > 1 Then iSpeed = iSpeed -
1
    lblSpeed.Text = "Speed: "& iSpeed

    If e.KeyCode = Keys.Up Then SendArduino("N0.-"& iSpeed)
    If e.KeyCode = Keys.Down Then SendArduino("N0."& iSpeed)

    If e.KeyCode = Keys.Left Then SendArduino("N-"& iSpeed & ".0")
    If e.KeyCode = Keys.Right Then SendArduino("N"& iSpeed & ".0")
    'Text = Now.ToLongTimeString

EndSub

PrivateSub pic_MouseMove(sender AsObject, e As MouseEventArgs) Handles
pic.MouseMove
    xMouse = e.X
    yMouse = e.Y
    'Text = e.X & " - " & e.Y
EndSub

PrivateSub Form1_FormClosing(sender AsObject, e As FormClosingEven-
tArgs) HandlesMe.FormClosing
    Try
        comArduino.Close()
    Catch ex As Exception

    EndTry

EndSub

PrivateSub ShowClock()

```

```

Dim iSec AsInteger = Microsoft.VisualBasic.Timer - iClock
Dim iMin AsInteger = Int(iSec / 60)
iSec = iSec - 60 * Int(iSec / 60)
lblClock.Text = Format(iMin, "00") & ":" & Format(iSec, "00")
lblClock.Refresh()

```

EndSub

```

PublicSharedFunction AforgeAutoCrop(ByVal selectedImage As Bitmap) As
System.Drawing.Image

```

```

    'Ρουτίνα που κάνει autocrop την εικόνα

    Dim autoCropImage As Bitmap = Nothing
    Try
        autoCropImage = selectedImage
        ' create grayscale filter (BT709)
        Dim filter AsNew AForge.Imaging.Filters.Grayscale(0.2125,
0.7154, 0.0721)
        Dim grayImage As Bitmap = filter.Apply(autoCropImage)
        ' create instance of skew checker
        Dim skewChecker AsNew AForge.Imaging.DocumentSkewChecker
        ' get documents skew angle
        Dim angle AsDouble = skewChecker.GetSkewAngle(grayImage)
        ' create rotation filter
        Dim rotationFilter AsNew AForge.Imaging.Filters.RotateBilin-
ear((angle * -1))
        rotationFilter.FillColor = Color.White
        ' rotate image applying the filter

        Dim rotatedImage As Bitmap = rotationFilter.Apply(grayImage)

        Dim filter2 AsNew AForge.Imaging.Filters.ContrastStretch
        rotatedImage = filter2.Apply(rotatedImage)

        Dim filter3 AsNew AForge.Imaging.Filters.Threshold(25)
        filter3.ApplyInPlace(rotatedImage)

```



```

    Dim bc AsNew AForge.Imaging.BlobCounter
    bc.FilterBlobs = True
    bc.MinWidth = 50
    bc.MinHeight = 50
    bc.ObjectsOrder = AForge.Imaging.ObjectsOrder.Size
    bc.ProcessImage(rotatedImage)
    Dim rects() As Rectangle = bc.GetObjectsRectangles
    If rects.Count <> 0 Then

        Dim x1 AsInteger = -1, y1 AsInteger = -1, x2 AsInteger
= -1, y2 AsInteger = -1
        For i = 0 To rects.Count - 1
            If rects(i).X < x1 Or x1 = -1 Then x1 =
rects(i).X
            If rects(i).Y < y1 Or y1 = -1 Then y1 =
rects(i).Y
            If rects(i).Right > x2 Or x2 = -1 Then x2 =
rects(i).Right
            If rects(i).Bottom > y2 Or y2 = -1 Then y2 =
rects(i).Bottom
        Next
        frmMain.Text = x1 &" - "&y1
        autoCropImage = rotatedImage.Clone(New Rectangle(x1,
y1, x2 - x1, y2 - y1), selectedImage.PixelFormat)
    Else
        autoCropImage = Nothing
    EndIf

    Catch ex As Exception
        MessageBox.Show(ex.Message)
    EndTry

    Return autoCropImage
EndFunction

PrivateSub cmd1_Click(sender AsObject, e As EventArgs) Handles
cmd1.Click
    LoadPhoto(1)
EndSub

```

```
PrivateSub cmd2_Click(sender AsObject, e As EventArgs) Handles  
cmd2.Click  
    LoadPhoto(2)  
EndSub
```

```
PrivateSub cmd3_Click(sender AsObject, e As EventArgs) Handles  
cmd3.Click  
    LoadPhoto(3)  
EndSub
```

```
PrivateSub cmd4_Click(sender AsObject, e As EventArgs) Handles  
cmd4.Click  
    LoadPhoto(4)  
EndSub
```

```
PrivateSub cmd5_Click(sender AsObject, e As EventArgs) Handles  
cmd5.Click  
    LoadPhoto(5)  
EndSub
```

```
PrivateSub cmd6_Click(sender AsObject, e As EventArgs) Handles cmd6.Click  
    LoadPhoto(6)  
EndSub
```

B2 Κώδικας Arduino

```
*****main*****
String sTempData="";
int iZoom=20;
//500=67  400=63  300=58sex  200=57
long iSpeedDraw=400;      //700 fast      100000 Slow (Lazer)
long iSpeedMove=250;
#include"Led.h"
#include"Motor.h"

int iMotoDown=0;
int iMotoDown2=0;

//ανάλυση 3500 x 2475      A4(1100 x 1450)
int myData[305][2];
int iDataCount=0;

void setup() {
    Serial.begin(115200);
    Serial.setTimeout(1);
    Serial.println(F("Start"));
    SetupMotor(); Serial.println(F("Motor OK"));
    SetupLed(); Serial.println(F("LED OK"));

    //digitalWrite(EN, HIGH);
    ShowLed(1,0,0,50); ShowLed(2,0,0,50);
    //MoveMotor(0,100,400);
    Serial.println(F("ALL OK"));
}

void AddLocate(int x,int y) {
    myData[iDataCount][0]=x;
    myData[iDataCount][1]=y;
    iDataCount=iDataCount+1;
    //Serial.println(iDataCount);
    Serial.println(F("OK"));
}
```

```

long iHow=0;
unsignedlong iTime=0;
int iDo=0;
String sCom="";
int iZDown=0, bDown=0;
voidloop(){
    if ((digitalRead(13)==0) && (bDown==0)) {
        bDown=1;
        if (iZDown==0) {iZDown=1;MoveMotorZ(100,2000);} else {iZ-
Down=0;MoveMotorZ(-100,2000);}
    }
    if ((digitalRead(13)==1) && (bDown==1)) {bDown=0;}

    int iA4 =analogRead(A4); int iA5 =analogRead(A5);
    if ((iA4<400) || (iA5<400) || (iA4>600) || (iA5>600)) {
        //Serial.print("A4:"); Serial.print(iA4);
        //Serial.print("  A5:"); Serial.print(iA5);
        //Serial.print("  D13:"); Serial.println(digitalRead(13));

        int x0=0, y0=0, iSpeed=500; //map(value, fromLow, fromHigh,
toLow, toHigh)
        if (iA4<400) {x0=-1; iSpeed = map(iA4, 0, 400, iSpeedMin,
iSpeedMax);}
        if (iA4>600) {x0=1; iSpeed =map(iA4, 600, 1024, iSpeedMax,
iSpeedMin);}

        if (iA5<=400) {y0=-1; iSpeed =map(iA5, 0, 400, iSpeedMin,
iSpeedMax);}
        if (iA5>=600) {y0=1; iSpeed =map(iA5, 600, 1024, iSpeedMax,
iSpeedMin);}
        //x0=0;

        int iSpeed2=map(iSpeed, iSpeedMin, iSpeedMax, 50, 1);

        MoveMotor(x0*iSpeed2,y0*iSpeed2,iSpeed);

        //Serial.print("x0:"); Serial.print(x0);
        //Serial.print("  y0:"); Serial.print(y0);
        //Serial.print("  speed:"); Serial.print(iSpeed);
        //Serial.print("  speed2:"); Serial.println(iSpeed2);
    }
}

```

```

if (Serial.available()>0) {
    sCom="";
    while (Serial.available() > 0){
        sCom = sCom +Serial.readStringUntil('\n');
    }
    //Serial.println(sCom);
    iDo=1;
}
if (iDo==1) {
    iDo=0;
    String s2;
    char str;
    str=sCom.charAt(0);
    s2=sCom.substring(1);
    int iNum=s2.toInt();
    int x=0; int y=0;
    //Serial.print(F("Command: ")); Serial.println(sCom);
    //Serial.print(F("str: ")); Serial.println(str);
    //Serial.print(F("s2: ")); Serial.println(s2);
    switch (str) {
        case'z':
            if (iZDown==0) {iZDown=1;MoveMotorZ(100,1000);}
            Serial.println("OK");
            break;
        case'Z':
            if (iZDown==1) {iZDown=0;MoveMotorZ(-100,1000);}
            Serial.println("OK");
            break;

        case'R':
            asm volatile ("jmp 0");
            break;
        case'r':
            ShowLed(1,50,0,50);ShowLed(2,50,0,50);
            delay(100);
            iDataCount=0;
            iMotoDown=0;
            iMotoDown2=0;
            ShowLed(1,0,0,50);ShowLed(2,0,0,50);
    }
}

```

```

        Serial.println("OK");
        break;

    case 'M':
        //iDataCount=0;
        //iMotoDown=2;
        //iMotoDown2=0;
        //x=getNumber(String(s2), '.', 0);
        //y=getNumber(String(s2), '.', 1);
        //AddLocate(x,y);
        //Execute();
        x=iZoom*getNumber(String(s2), '.', 0);
        y=iZoom*getNumber(String(s2), '.', 1);
        MoveMotor(-x,-y,iSpeedDraw);
        Serial.println("OK");
        break;

    case 'm':
        //iDataCount=0;
        //iMotoDown=3;
        //iMotoDown2=0;
        //x=getNumber(String(s2), '.', 0);
        //y=getNumber(String(s2), '.', 1);
        //AddLocate(x,y);
        //Execute();
        x=iZoom*getNumber(String(s2), '.', 0);
        y=iZoom*getNumber(String(s2), '.', 1);
        MoveMotor(-x,-y,iSpeedMove);
        Serial.println("OK");
        break;

    case 'A':
        x=getNumber(String(s2), '.', 0);
        y=getNumber(String(s2), '.', 1);
        AddLocate(x,y);
        break;

    case '@':
        AddLocate(9999,9999);
        break;

    case 'E':

```

```

        Execute();
        break;
    }
}

}

void Execute() {
    int x=0; int y=0;
    if (iDataCount==0) {iDataCount=0; iMotoDown2=0; Serial
al.println(F("OK")); return;}

    //Serial.print(F("Count:"));Serial.println(iDataCount);
    digitalWrite(EN,LOW);
    if (iMotoDown==1) {iMotoDown2=0; MoveMotorZ(100,1000);}
    for (int i=0;i<iDataCount;i=i+1) {
        Serial.print(F("@")); Serial.println(iDataCount-i);
        x = myData[i][0];
        y = myData[i][1];

        if ((x==9999) && (y==9999)) {
            if (iMotoDown==0)
                {iMotoDown=1; MoveMotorZ(100,1000);}
            else
                {iMotoDown=0; MoveMotorZ(-100,1000);}
        }
        else
        {
            x = iZoom*x;
            y = iZoom*y;
            if (iMotoDown==0) {MoveMotor(-x,-y,iSpeedMove);}
            if (iMotoDown==1) {MoveMotor(-x,-y,iSpeedDraw);}

            if (iMotoDown==2) {MoveMotor(-x,-y,iSpeedDraw);}
            if (iMotoDown==3) {MoveMotor(-x,-y,iSpeedMove);}
        }
    }

    if (iMotoDown==1) {iMotoDown2=1; MoveMotorZ(-100,1000);}
}

```

```

    digitalWrite(EN,HIGH);
    iDataCount=0;
    iMotoDown2=0;
    Serial.println(F("OK"));
}

```

```

String getValue(String data, char separator, int index)
{
    int found = 0;
    int strIndex[] = {
        0,-1
    };
    int maxIndex = data.length() - 1;
    for (int i = 0; i <= maxIndex && found <= index; i++) {
        if (data.charAt(i) == separator || i == maxIndex) {
            found++;
            strIndex[0] = strIndex[1] + 1;
            strIndex[1] = (i == maxIndex) ? i + 1 : i;
        }
    }
    return found > index ? data.substring(strIndex[0], strIndex[1])
    : "";
}

```

```

//πέρνει απο ενα string ενα κομμάτι και το κάνει νούμερο
int getNumber(String data, char separator, int index)
{
    String xval = getValue(data, separator, index);
    if (xval == "")
    {
        return -1;
    }
    else
    {
        return xval.toInt();
    }
}

```



```
*****LED*****
```

```
#include<FastLED.h>
CRGB leds[12];

void ShowLed (int iLed,int r,int g,int b) {
    if (iLed==-1) {
        for (int i=0;i<=11;i=i+1) {
            ShowLed(i,r,g,b);
            //delay(20);
        }
    }
    else
    {leds[iLed] =CRGB(r,g,b);
    FastLED.show();
    }
}
```

```
void SetupLed() {
    FastLED.addLeds<NEOPIXEL, 12>(leds, 12);
    ShowLed(-1,0,0,0);

    for (int i=0;i<=11;i=i+1) {
        ShowLed(i,50,0,50);
        delay(50);
    }
    ShowLed(-1,0,0,0);
}
```

```
*****Motor*****
```

```
//Step pin
#define X_STP      2
#define Y_STP      3
#define Z_STP      4      //****      ZSTEP/DIR (blue)

//Direction pin
#define X_DIR      5
#define Y_DIR      6
#define Z_DIR      7      //****      ZSTEP/DIR (white)

#define EN         8

//limit pin
//#define X_LMT     9
//#define Y_LMT    10
```

```

//#define Z_LMT      11      ****LED (white

//#define A_STP 12          **** Spindle Enabled (white
//#define A_DIR 13         **** Spindle Direction (white

int iLocateX=0, iLocateY=0;
unsignedlong iTimeMotorOFF=0;

void SetupMotor() {
  pinMode(A0,OUTPUT); digitalWrite(A0,LOW); //Reset/Abort
  pinMode(A1,OUTPUT); digitalWrite(A1,LOW); //Feed Hold
  pinMode(A2,OUTPUT); digitalWrite(A2,LOW); //Cycle Start/Re-
  sume
  pinMode(A3,OUTPUT); digitalWrite(A3,LOW); //Coolan Enabled

  pinMode(X_DIR,OUTPUT); pinMode(X_STP,OUTPUT); //pinMode(X_LMT,
  INPUT);
  pinMode(Y_DIR,OUTPUT); pinMode(Y_STP,OUTPUT); //pinMode(Y_LMT,
  INPUT);
  pinMode(Z_DIR,OUTPUT); pinMode(Z_STP,OUTPUT); //pinMode(Z_LMT,
  INPUT);

  //pinMode (A_DIR, OUTPUT);
  //pinMode (A_STP, OUTPUT);
  pinMode(EN,OUTPUT);

  //digitalWrite(X_LMT, HIGH);
  //digitalWrite(Y_LMT, HIGH);

  pinMode(13,INPUT);
  digitalWrite(EN,HIGH);
}

void DelayMotor(long iSpeed) {
  //Serial.print("Speed:");Serial.println(iSpeed);
  if (iSpeed<16383) {delayMicroseconds(iSpeed);}
  else
  {delay(iSpeed/1000);}
}

int iSpeedMax=2500, iSpeedMin=150;
void MoveMotor(int x,int y,long iSpeed) {
  //iTimeMotorOFF=millis();
  digitalWrite(EN,LOW);

```

```

//Serial.print("Move X: "); Serial.print(x); Serial.print("  Y:
"); Serial.print(y);
//Serial.print("  Speed: "); Serial.println(iSpeed);
//Serial.print("  t1: "); Serial.print(millis());
unsignedlong iTimeM=millis();

int iColor=map(iSpeed, iSpeedMin, iSpeedMax, 4,-1);
//Serial.print("  iColor: "); Serial.print(iColor);
boolean dirX, dirY;
if (x>0) {dirX=false; for (int i=0;i<=iColor;i=i+1) {leds[2+i]
=CRGB(40,0,0);}}
if (x<0) {dirX=true; for (int i=0;i<=iColor;i=i+1) {leds[2+i]
=CRGB(0,40,0);}}
if (y>0) {dirY=true; for (int i=0;i<=iColor;i=i+1) {leds[7+i]
=CRGB(40,0,0);}}
if (y<0) {dirY=false; for (int i=0;i<=iColor;i=i+1) {leds[7+i]
=CRGB(0,40,0);}}
FastLED.show();

//if (y>0) {dirY=false;} else {dirY=true;}
digitalWrite(X_DIR, dirX); digitalWrite(Y_DIR, dirY);

x=abs(x);y=abs(y);
//int xx=abs(x), yy=abs(y);
//int xy=0, yx=0;
//if (xx>yy) {xy=xx/yy;yx=yy/xx;} else {xy=yy/xx;yx=xx/yy;}

//Serial.print("xx: "); Serial.print(xx); Serial.print("  yy:
"); Serial.print(yy);
//Serial.print("  xy: "); Serial.print(xy); Serial.print("
yx: "); Serial.println(yx);

//int n=0; int y1=0;
//digitalWrite(EN, LOW);
do {
  if (x>0) {digitalWrite(X_STP,HIGH);}
  if (y>0) {digitalWrite(Y_STP,HIGH);}

  DelayMotor(iSpeed);

  if (x>0) {x=x-1; digitalWrite(X_STP,LOW);}
  if (y>0) {y=y-1; digitalWrite(Y_STP,LOW);}

  //Serial.print("x:");Serial.print(x); Serial.print("  y:"); Se-
rial.println(y);
} while ((x>0) || (y>0));

```

```

    //Serial.print(" - "); Serial.print(millis());
    Serial.print("("); Serial.print(millis()-iTimeM); Serial.
    al.println(")");
    for (int i=2;i<=11;i=i+1) {leds[i] =CRGB(0,0,0);} Fas-
    tLED.show();
    digitalWrite(EN,HIGH);
}

void MoveMotorZ(int stepsZ,long iSpeed) {
    //iTimeMotorOFF=millis();
    digitalWrite(EN,LOW);

    //Serial.println("MoveMotorZ");
    boolean dirZ;
    if (stepsZ<0) {dirZ=false; ShowLed(0,0,40,0);
    ShowLed(1,0,40,0);}
    if (stepsZ>0) {dirZ=true; ShowLed(0,40,0,0);
    ShowLed(1,40,0,0);}
    digitalWrite(Z_DIR, dirZ);

    for (int i = 0; i <abs(stepsZ); i++) {
        digitalWrite(Z_STP,HIGH);
        DelayMotor(iSpeed);
        digitalWrite(Z_STP,LOW);
    }

    digitalWrite(EN,HIGH);
    iTimeMotorOFF=millis();
}

```