

**ομάδα ρομποτικής
1ου ΓΕΛ Φλώρινας
2015-2016**

**“ρομποτικός βραχίονας -
ιμπρεσιονιστής”**



ομάδα ρομποτικής

1ο ΓΕΛ Φλώρινας σχ. έτος 2015-2016

τα μέλη της ομάδας:

*Βασιλειάδης Χάρης
Βελιάνης Σπύρος
Γεωργιάδης Λάζαρος
Δαΐρετζής Κωνσταντίνος
Δόντσιος Ανέστης
Ιωάννου Βασίλης
Μιαούλης Ραφαήλ-Εμμανουήλ
Μπαλασόπουλος Μάριος
Παπαδάκης Στέργιος
Τοσουνιδης Αλέξανδρος
Τσάιας Πέτρος
Τσωτσος Δημητρης*

υπεύθυνος καθηγητής:

Παπαδόπουλος Τρ. Δημήτριος (ΠΕ20)

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Ποιοι είμαστε	4
Γενικά - εισαγωγικά για την κατασκευή μας	4
Τι είναι οι σερβοκινητήρες και πως λειτουργούν	6
Arduino	7
Υλικό (Hardware)	7
Λογισμικό	8
Η Γλώσσα Processing	10
Υλικά	11
Κατασκευή	12
Αρχή λειτουργίας	12
Θεωρητικό υπόβαθρο (Αντίστροφη κινηματική / Inverse kinematics)	12
Ψηφιακές εικόνες και απλές επεξηγήσεις	15
Περιγραφή εφαρμογής / λογισμικού σε processing	16
Περιγραφή λογισμικού ελέγχου (firmware) arduino	18
Τομείς εφαρμογής	19
Βελτιώσεις - παρατηρήσεις	19
παράρτημα: φωτογραφίες	20

Ποιοι είμαστε

Είμαστε η ομάδα ρομποτικής του 1ου Γενικού Λυκείου Φλώρινας και ξεκινήσαμε σαν ομάδα την ενασχόλησή μας με θέματα τεχνολογίας το σχ. έτος 2015-2016.

Είμαστε μια ομάδα μαθητών από όλες τις τάξεις του σχολείου, που στον ελεύθερο χρόνο μας και μέσω των πολιτιστικών προγραμμάτων, ασχολούμαστε με θέματα τεχνολογίας σε τομείς της ζωής μας που έχουν αλλάξει δραματικά και συνεχώς αλλάζουν με την προσθήκη αυτόματων μηχανών - ρομπότ.

Μελετάμε και κατασκευάζουμε “ρομπότ” που μπορούν να χρησιμοποιηθούν σε κάθε τομέα της ζωής μας, και με τον τρόπο αυτόν προσπαθούμε να υλοποιήσουμε έναν συνεργατικό τρόπο μελέτης - έρευνας που τα τελευταία χρόνια είναι γνωστός ως **STEMA** (Science Technology Engineering Maths Art) (Επιστήμη / Τεχνολογία / Μηχανική / Μαθηματικά / Τέχνες). Εμπλεκόμενοι με κάθε τομέα ξεχωριστά, προσπαθούμε να αποκτήσουμε μια σφαιρική γνώση πάνω στο αντικείμενο με το οποίο ασχολούμαστε κάθε φορά.

Δεν παραβλέπουμε βέβαια και τις επιπτώσεις σε διάφορους τομείς της ζωής μας, κουλτούρα, οικονομία κλπ από τη χρήση των ρομπότ στην σημερινή κοινωνία και στο μέλλον.

Βασικός μας επίσης στόχος είναι η χρήση όσο το δυνατόν μεγαλύτερου ποσοστού ανακυκλωμένων ή ανακυκλώσιμων απλών υλικών στην κατασκευή μας, είτε από κατεστραμμένες ηλεκτρονικές / ηλεκτρικές μηχανές, είτε από άχρηστα (scrap) καθημερινά υλικά που προορίζονται για τον κάδο ανακύκλωσης.

Γενικά - εισαγωγικά για την κατασκευή μας

Η κατασκευή μας αφορά έναν ρομποτικό βραχίονα, ο οποίος έχει τα εξής **χαρακτηριστικά**:

- εκτύπωση με γραμμές / πινελιές σαν **ιμπρεσσιονιστής**
- Φιλική στο περιβάλλον αφού κατασκευάστηκε από ανακυκλωμένες κατά βάση συσκευές
- Μπορεί να εκτυπώσει με κοινό μολύβι, στυλό, μαρκαδόρο, κάρβουνο ή ότι άλλο καθημερινό μέσο γραφής ή ζωγραφικής φαντασθούμε
- Υπερβολικά φθηνός
- Δυνατότητα εκτύπωσης bitmap αρχείων
- Πολύ απλό λογισμικό
- Μικρό, εύκολα μετακινήσιμο,
- Εκτύπωση μόνο με την θύρα usb χωρίς άλλη πηγή τροφοδοσίας.

Σύντομη περιγραφή κατασκευής :

Ο “ρομπο-ιμπρεσσιονιστής” μπορεί να δεχθεί μια φωτογραφία από κάμερα, web-camera ή αρχείο εικόνας και να το αποτυπώσει στο χαρτί / καμβά ή σε όποια επιφάνεια επιθυμούμε.

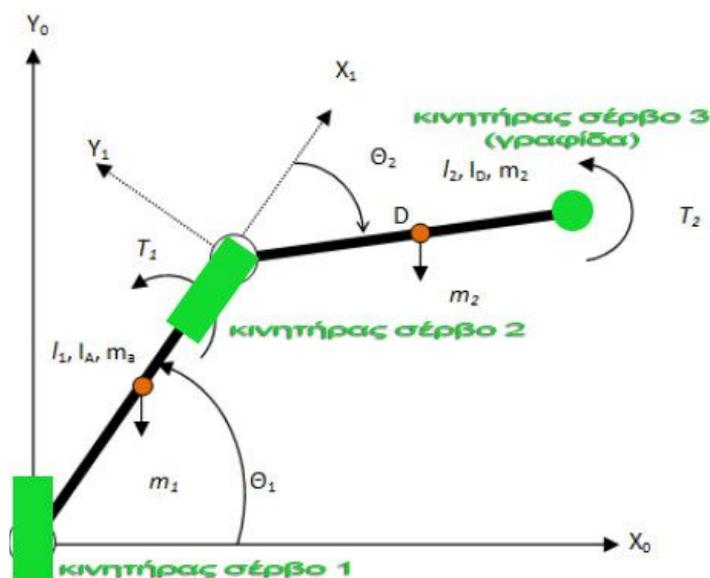
Επίσης, μπορεί να εκτυπώσει κάθετα με μικρές μετατροπές και διαφορετικό αποτέλεσμα με κουκίδες (Pixels) και μεγαλύτερη ακρίβεια.

Ο τρόπος λειτουργίας του, και ο τρόπος κίνησης των κινητήρων βασίζεται στα μαθηματικά και για να υλοποιηθεί χρειασθήκαμε τη βοήθεια των μαθηματικών του σχολείου μας και του Σιμηχανίδη Νικολάου και του Μιντεκίδη Παντελή.

Το λογισμικό αναγνώρισης της εικόνας και δημιουργίας των εντολών εκτύπωσης βασίζεται στην **Processing**, και το υλικολογισμικό (εφαρμογή οδήγησης) στο **Arduino IDE** χρησιμοποιώντας σαν κεντρικό επεξεργαστή έναν **μικροεπεξεργαστή Arduino**.

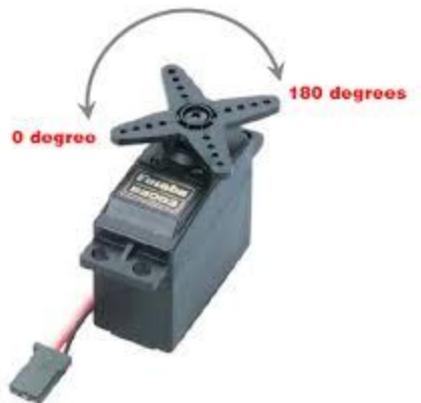
Οι “μύες” του ρομπότ καλλιτέχνη μας αποτελούνται από 3 σερβο-κινητήρες και επομένως έχει 3 βαθμούς ελευθερίας κίνησης. Οι δύο χρησιμοποιούνται ώστε να κινήσουν την γραφίδα στο επιθυμητό σημείο και ο τρίτος χειρίζεται τη γραφίδα θέτοντας την σε κατάσταση εγγραφής ή όχι.

Η βασική λειτουργία αποτυπώνεται στην παρακάτω εικόνα και θα μιλήσουμε αναλυτικότερα παρακάτω. Στόχος δηλαδή είναι η τοποθέτηση 2 ευθ. τμημάτων γνωστού μήκους σε τέτοιες γωνίες ώστε η ακμή του 2ου ευθ. τμήματος να βρίσκεται στο σημείο που επιθυμούμε να ζωγραφίσουμε. Αυτό επαναλαμβάνεται για κάθε σημείο της εικόνας μας.



Τι είναι οι σερβοκινητήρες και πως λειτουργούν

Με απλά λόγια είναι κινητήρες που συνήθως δεν περιστρέφονται 360ο αλλά μόνο 180ο. Ένα παράδειγμα χρήσης τους είναι το σύστημα διεύθυνσης (κατεύθυνση) ενός τηλεκατευθυνόμενου αυτοκινήτου, αεροπλάνου κλπ. Η χρήση τους βέβαια δεν περιορίζεται στα παραπάνω παραδείγματα, αλλά είναι ευρεία σε ρομποτικούς βραχίονες κλπ και γενικά στον τομέα του ελέγχου κίνησης και ειδικότερα σε εφαρμογές ελέγχου θέσεως ταχύτητας και ροπής άξονα.



Το servo είναι ένας κινητήρας που έχει έναν προεξέχοντα άξονα.

Ο άξονας αυτός μπορεί να τοποθετηθεί σε συγκεκριμένες θέσεις, γωνίες αφού δεχθεί κάποια κωδικοποιημένα σήματα. Έχουν μικρο μηχανισμό, ενσωματωμένο σύστημα ελέγχου, και πολύ μεγάλη δύναμη συγκριτικά με το μέγεθος τους.

Η ενέργεια που καταναλώνει είναι ανάλογη με το μηχανικό φορτίο που δέχεται. Δηλαδή όσο λιγότερο φορτίο δίνουμε σε ένα servo τόσο λιγότερη ενέργεια καταναλώνει.

Πώς λειτουργεί ένα servo;

Ο πιο απλός τρόπος χρήσης του κινητήρα σέρβο (και ο τρόπος που χρησιμοποιήσαμε στην εφαρμογή μας) είναι ο εξής: Ζητάμε από το σέρβο να πάει σε κάποια προκαθορισμένη γωνία. πχ η εντολή σέρβο(88) θα αναγκάσει τον άξονα του κινητήρα να ευθυγραμμισθεί στις 88 μοίρες.

Ο μηχανισμός του servo έχει κάποια κυκλώματα ελέγχου κι ένα ποτενσιόμετρο (μεταβλητή αντίσταση) που συνδέονται με τον προεξέχον άξονα. Το κύκλωμα ελέγχου να καταγράφει την τρέχουσα γωνία του servo-μοτέρ. Αν ο άξονας είναι σε σωστή γωνία, τότε το μοτέρ σβήνει. Αν το κύκλωμα καταλάβει ότι η γωνία δεν είναι η σωστή, θα δώσει εντολή να λειτουργήσει το μοτέρ προς τη σωστή κατεύθυνση έτσι ώστε ο άξονας να αποκτήσει τη σωστή γωνία.

Ο άξονας του servo μπορεί να περιστραφεί περίπου από 180ο - 210ο . Η γωνία αυτή ποικίλει ανάλογα με τον κατασκευαστή . Τα περισσότερα servo χρησιμοποιούνται για να ελέγχουν μια κυκλική κίνηση μεταξύ 0ο και 180ο και δεν μπορούν να περιστραφούν περισσότερο εξαιτίας ενός μηχανικού στοπ που υπάρχει εγκατεστημένο στο κύριο (προεξέχων) γρανάζι.

Η ενέργεια που καταναλώνει το μοτέρ, είναι ανάλογη με τη διαδρομή που πρέπει να κάνει. Δηλαδή, αν ο άξονας πρέπει να περιστραφεί πολύ, το μοτέρ θα εκτελέσει την κίνηση με μεγάλη ταχύτητα (και επομένως ενέργεια). Αν πρέπει να περιστραφεί λίγο, το μοτέρ θα εκτελέσει την κίνηση με μικρότερη ταχύτητα (και επομένως ενέργεια).

Arduino



Μιας και το “μυαλό” και το κέντρο ελέγχου της κατασκευής μας βασίζεται σε έναν μικροεπεξεργαστή Arduino, ας αναφερθούμε λίγο αναλυτικότερα στο τι είναι η πλατφόρμα arduino.

Το Arduino είναι ένας μικροελεγκτής μονής πλακέτας, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Pure Data, SuperCollider. Οι περισσότερες εκδόσεις του Arduino μπορούν να αγοραστούν προ-συναρμολογημένες· το διάγραμμα και πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους.

Υλικό (Hardware)

Μία πλακέτα Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR (ATmega328 και ATmega168 στις νεότερες εκδόσεις, ATmega8 στις παλαιότερες) και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωσή του σε άλλα κυκλώματα. Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz (ή κεραμικό αντηχητή σε κάποιες παραλλαγές). Ο

μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα bootloader, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής.

Σε εννοιολογικό επίπεδο, στην χρήση του Arduino software stack, όλα τα boards προγραμματίζονται με μία RS-232 σειριακή σύνδεση, αλλά ο τρόπος που επιτυγχάνεται αυτό διαφέρει σε κάθε hardware εκδοχή. Οι σειριακές πλάκες Arduino περιέχουν ένα απλό level shifter κύκλωμα για να μετατρέπει μεταξύ σήματος επιπέδου RS-232 και TTL. Τα τωρινά Arduino προγραμματίζονται μέσω USB· αυτό καθίσταται δυνατό μέσω της εφαρμογής προσαρμοστικών chip USB-to-Serial όπως το FTDI FT232. Κάποιες παραλλαγές, όπως το Arduino mini και το ανεπίσημο Boarduino, χρησιμοποιούν ένα αφαιρούμενο USB-to-Serial καλώδιο ή board, Bluetooth ή άλλες μεθόδους. (Όταν χρησιμοποιείται με παραδοσιακά εργαλεία microcontroller αντί για το Arduino IDE, χρησιμοποιείται πρότυπος προγραμματισμός AVR ISP).

Ο πίνακας Arduino εκθέτει τα περισσότερα microcontroller I/O pins για χρήση από άλλα κυκλώματα. Τα Diecimila, Duemilanove και το τρέχον Uno παρέχουν 14 ψηφιακά I/O pins, έξι από τα οποία μπορούν να παράγουν pulse-width διαμορφωμένα σήματα, και έξι αναλογικά δεδομένα. Αυτά τα pins βρίσκονται στην κορυφή του πίνακα μέσω female headers 0.1 ιντσών (2,2mm). Διάφορες εφαρμογές ασπίδων plug-in είναι εμπορικώς διαθέσιμες.

Το Arduino nano και το Arduino-Compatible Bare Bones Board και Boarduino Board ενδέχεται να παρέχουν male header pins στο κάτω μέρος του board προκειμένου να συνδέονται σε Breadboards. Υπάρχουν πολλά boards συμβατά με και προερχόμενα από Arduino boards. Κάποια είναι λειτουργικά ισάξια με ένα Arduino και μπορεί να χρησιμοποιηθούν εναλλακτικά. Πολλοί είναι το βασικό Arduino με την προσθήκη καινοτόμων output drivers, συχνά για την χρήση σχολικής μόρφωσης για να απλοποιήσουν την κατασκευή buggies και μικρών robot. Άλλες είναι ηλεκτρικά ισάξια αλλά αλλάζουν τον παράγοντα μορφής, επιτρέποντας κάποιες φορές την συνεχόμενη χρήση των Shields ενώ κάποιες όχι. Κάποιες παραλλαγές είναι τελείως διαφορετικοί επεξεργαστές, με ποικίλα επίπεδα συμβατότητας.

Λογισμικό



```

Arduino - 9011 Alpha
File Edit Sketch Tools Help
Sketch
/*
 * LED
 *
 * The basic Arduino example. Turns an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/LED
 */
int ledPin = 13; // LED connected to digital pin 13

void setup() // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop() // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000); // waits for 1 second
  digitalWrite(ledPin, LOW); // sets the LED off
  delay(1000); // waits for 1 second
}
Done compiling
Binary sketch size: 2096 bytes (of a 32768 byte maximum)

```

Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java, που λειτουργεί σε πολλές πλατφόρμες και προέρχεται από το IDE για τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring. Έχει σχεδιαστεί για να εισαγάγει τον προγραμματισμό στους καλλιτέχνες και τους νέους που δεν είναι εξοικειωμένοι με την ανάπτυξη λογισμικού. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης και ο συνδυασμός αγκύλων και είναι επίσης σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα με ένα μόνο κλικ. Δεν υπάρχει συνήθως καμία ανάγκη να επεξεργαστείτε αρχεία make ή να τρέξετε προγράμματα σε ένα περιβάλλον γραμμής εντολών. Ένα πρόγραμμα ή κώδικας που γράφτηκε για Arduino ονομάζεται σκίτσο (sketch)[16].

Τα Arduino προγράμματα είναι γραμμένα σε C ή C++. Το Arduino IDE έρχεται με μια βιβλιοθήκη λογισμικού που ονομάζεται "Wiring", από το πρωτότυπο σχέδιο Wiring, γεγονός που καθιστά πολλές κοινές λειτουργίες εισόδου/εξόδου πολύ πιο εύκολες. Οι χρήστες πρέπει μόνο να ορίσουν δύο λειτουργίες για να κάνουν ένα πρόγραμμα κυκλικής εκτέλεσης:

-setup():μία συνάρτηση που τρέχει μία φορά στην αρχή του προγράμματος η οποία αρχικοποιεί τις ρυθμίσεις

-loop():μία συνάρτηση που καλείται συνέχεια μέχρι η πλακέτα να απενεργοποιηθεί

Η Γλώσσα Processing

Για την δημιουργία των εντολών εκτύπωσης από οποιαδήποτε εικόνα, χρησιμοποιήσαμε την γλώσσα Processing.

Η Processing είναι μια γλώσσα προγραμματισμού ανοικτού κώδικα και παράλληλα ένα προγραμματιστικό περιβάλλον για ανθρώπους που θέλουν να προγραμματίσουν εικόνες, animation και ήχο. Όλα ξεκίνησαν το 2001 όταν δύο απόφοιτοι του πανεπιστημίου MIT, Benjamin Fry και Casey Reas ξεκίνησαν την ανάπτυξη της γλώσσας Processing πάνω σε Java. Παρόλο που η γλώσσα αναπτύχθηκε στη Java, το συντακτικό της είναι απλουστευμένο και το προγραμματιστικό της μοντέλο βασίζεται στα γραφικά. Απώτερος σκοπός των δύο δημιουργών είναι η εκμάθηση προγραμματισμού από αρχάριους χρήστες μέσω ενός οπτικού πλαισίου καθώς και η παροχή ενός επαγγελματικού εργαλείου παραγωγής πολυμεσικών εφαρμογών.

Το περιβάλλον της Processing είναι γραμμένο σε **Java**. Προγράμματα που έχουν γραφτεί σε Processing είναι επίσης μεταφρασμένα σε Java και έπειτα τρέχουν σαν Java προγράμματα. Προγράμματα που είναι γραμμένα σε Java και Processing, συνήθως τρέχουν γρηγορότερα από προγράμματα που βασίζονται σε scripting γλώσσες όπως ActionScript και Lingo, κάτι το οποίο είναι πολύ σημαντικό για εφαρμογές με γραφικά.

Μεγάλες διακρίσεις μεταξύ του Processing και της Java είναι οι βιβλιοθήκες γραφικών του Processing και το απλοποιημένο στυλ προγραμματισμού που δεν προϋποθέτει οι χρήστες να καταλαβαίνουν πιο προηγμένες έννοιες όπως κλάσεις, αντικείμενα, ή animation (ενώ εξακολουθούν να είναι διαθέσιμες για προχωρημένους χρήστες). Τέτοιες τεχνικές λεπτομέρειες πρέπει να είναι ειδικά προγραμματισμένες σε Java, αλλά ολοκληρωμένες σε Processing, κάνοντας τα προγράμματα μικρότερα και ευκολότερα να διαβαστούν.

Υλικά

Για την κατασκευή μας διαλέξαμε να χρησιμοποιήσουμε υλικά που μπορούμε να βρούμε πολύ εύκολα, φθηνά ή μηδενικού κόστους (scrap), με στόχο την ανακύκλωση συσκευών, υλικών.

Τι χρειαστήκαμε:

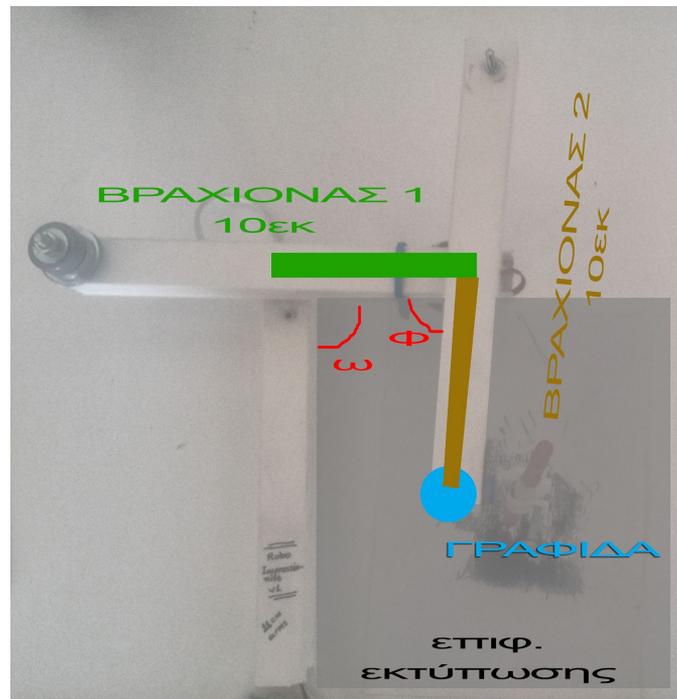
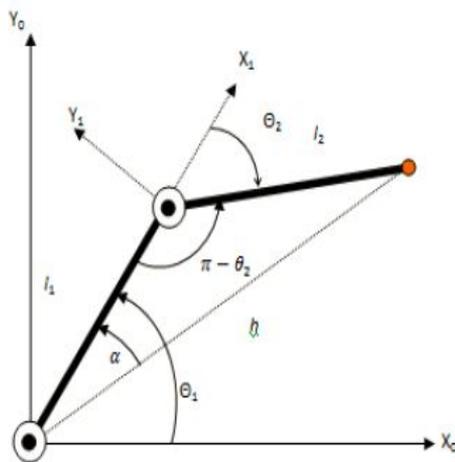
- Τρεις μικρο σερβοκινητήρες τους οποίους μπορούμε να βρούμε στο σύστημα διεύθυνσης παλαιών χαλασμένων τηλεχειριζόμενων παιχνιδιών
- Για τους βραχίονες, κομμάτια (ρετάλια) απο παλιά καναλάκια καλωδίων που βρήκαμε στο σχολείο.
- Arduino μικροεπεξεργαστής
- καλωδιάκια σύνδεσης

ενδεικτικό κόστος όλης της κατασκευής: 5 περίπου ευρώ συνολικά.

Κατασκευή

Αρχή λειτουργίας

Θεωρητικό υπόβαθρο (Αντίστροφη κινηματική / Inverse kinematics)



Ένα απλό διάγραμμα για το βραχίονα ρομπότ δίνεται παραπάνω, όπου η ακραία απόληξη (σημείο που θέλω να φτάσω) αντιπροσωπεύεται από το πορτοκαλί κύκλο. (X_0, y_0) είναι η αρχή συντεταγμένων για 1ο βραχίονα, και (x_1, y_1) συντεταγμένες 2ου βραχίονα.

Το να μετακινηθώ σε ένα νέο σημείο σημαίνει λόγω του τρόπου της λειτουργίας των σερβοκινητήρων, να υπολογίσω απλά τις 2 γωνίες κάθε κινητήρα (κύκλοι με μαύρο κέντρο στην αρχή κάθε άκρου - βραχίονα).

Στην ρομποτική η παραπάνω διαδικασία λέγεται **αντίστροφη κινηματική (Inverse kinematics)**. Όταν ξέρουμε που θέλουμε να πάμε και μπορούμε να υπολογίσουμε πώς και πόσο θα κινηθούν οι κινητήρες (μύες) για να φτάσουμε στο νέο σημείο.

Τα μαθηματικά που χρειαστήκαμε, επιλύθηκαν και μας εξηγήθηκαν από τους μαθηματικούς του σχολείου. Ακολουθεί ο οδηγός κίνησης από τον κο Σιμηχανίδη Νικόλαο και είναι όλοι οι τύποι που χρειαζόμαστε ώστε να υπολογίσουμε τις γωνίες των 2 κινητήρων ώστε να φτάσουμε στο εικονοστοιχείο (pixel) που θέλουμε να αποτυπώσουμε στο χαρτί.

i) $\triangle OM_0M = \text{ισοσκελές}$, $p = OM$, $(OM_0) = (M_0M) = a$

ii) $\omega = \theta + \beta$
 $\psi = \theta - \beta$

iii) $\epsilon\psi\theta = \frac{y}{x} \Rightarrow \theta = \text{το}\int \epsilon\psi\frac{y}{x}$

iv) $\triangle OM_0M$: $p^2 = a^2 + a^2 - 2 \cdot a \cdot a \cdot \sigma\omega M_0 \Rightarrow$
 $\sigma\omega M_0 = \frac{2a^2 - p^2}{2a^2}$
 $\hat{M}_0 + 2\beta = 180^\circ \Rightarrow \hat{M}_0 = 180^\circ - 2\beta$
 $\sigma\omega \hat{M}_0 = \sigma\omega(180 - 2\beta) = -\sigma\omega 2\beta$ } \Rightarrow

$\Rightarrow -\sigma\omega 2\beta = \frac{2a^2 - p^2}{2a^2} \Rightarrow \sigma\omega 2\beta = \frac{p^2 - 2a^2}{2a^2} \Rightarrow$
 $\Rightarrow 2\sigma\omega^2\beta - 1 = \frac{p^2 - 2a^2}{2a^2} \Rightarrow 2\sigma\omega^2\beta = \frac{p^2 - 2a^2}{2a^2} + 1 \Rightarrow$
 $\Rightarrow 2\sigma\omega^2\beta = \frac{p^2}{2a^2} \Rightarrow \sigma\omega^2\beta = \frac{p^2}{4a^2} \Rightarrow \sigma\omega\beta = \frac{p}{2a} \Rightarrow$
 $\Rightarrow \beta = \text{το}\int \sigma\omega \frac{p}{2a}$

(2)

- $x_0^2 + y_0^2 = a^2$
- $(x - x_0)^2 + (y - y_0)^2 = a^2 \Rightarrow \left. \begin{aligned} x^2 + y^2 &= 2(x_0x + y_0y) \\ x_0x + y_0y &= a\rho \cdot \sigma\mu\beta \\ x^2 + y^2 &= \rho^2 \end{aligned} \right\} \Rightarrow$

$\Rightarrow \rho^2 = 2 \cdot a\rho \cdot \sigma\mu\beta \Rightarrow \sigma\mu\beta = \frac{\rho}{2a} \Rightarrow \beta = \tau\omega \int \sigma\mu \frac{\rho}{2a}$

$\left. \begin{aligned} \varepsilon\varphi\omega &= \frac{y_0}{x_0} \\ x_0^2 + y_0^2 &= a^2 \end{aligned} \right\} \Rightarrow \left. \begin{aligned} x_0^2 + x_0^2 \cdot \varepsilon\varphi^2\omega &= a^2 \Rightarrow \\ x_0^2(1 + \varepsilon\varphi^2\omega) &= a^2 \Rightarrow \\ x_0^2 &= \frac{a^2}{1 + \varepsilon\varphi^2\omega} \end{aligned} \right\} \Rightarrow$

$x_0 = \sqrt{\frac{a^2}{1 + \varepsilon\varphi^2\omega}} \quad , \quad 0 \leq \omega < \frac{\pi}{2} \quad \wedge \quad \varepsilon\varphi\omega > 0$

$x_0 = -\sqrt{\frac{a^2}{1 + \varepsilon\varphi^2\omega}} \quad , \quad \frac{\pi}{2} < \omega \leq \pi \quad \wedge \quad \varepsilon\varphi\omega < 0$

$x_0 = 0 \quad , \quad \omega = \frac{\pi}{2}$

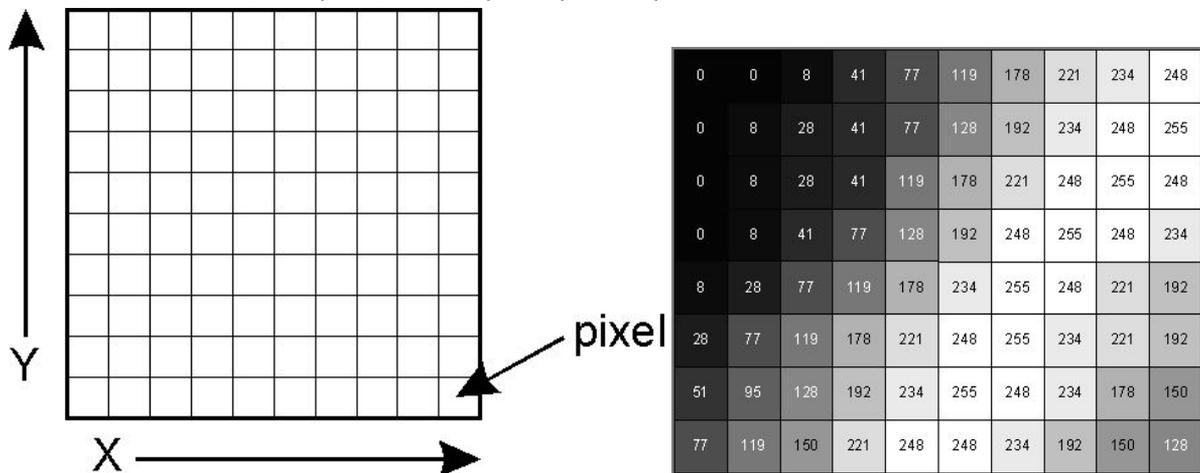
$y_0 = x_0 \cdot \varepsilon\varphi\omega$ με: $x_0 \cdot \varepsilon\varphi\omega > 0$

Ψηφιακές εικόνες και απλές επεξηγήσεις

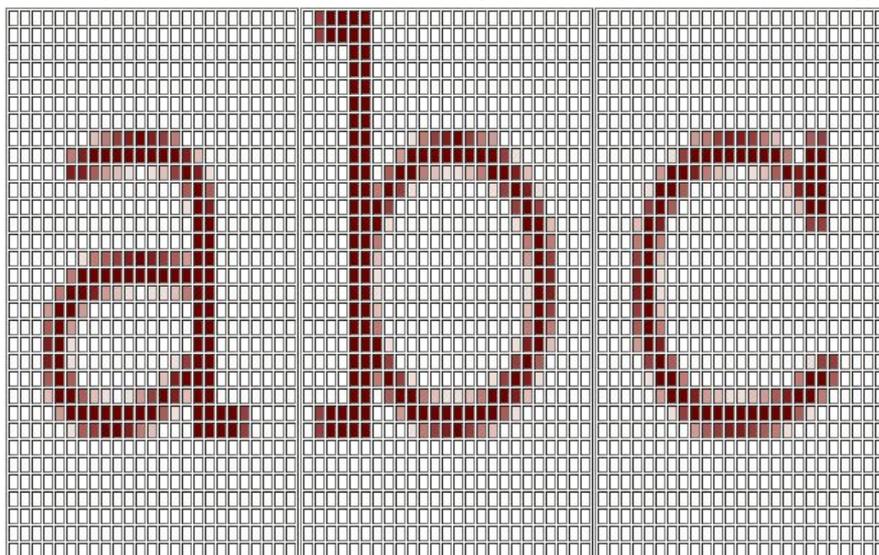
Μια ψηφιακή εικόνα δεν είναι τίποτα περισσότερο από δεδομένα - αριθμούς που δείχνουν παραλλαγές του κόκκινου, πράσινου και μπλε ή αποχρώσεις του γκρι (σε περίπτωση μονοχρωμίας) σε μια συγκεκριμένη θέση σε ένα πλέγμα των pixels.

Τις περισσότερες φορές, θα δείτε αυτά τα εικονοστοιχεία ως μικρογραφία ορθογώνιων να στριμώχνονται μαζί στην οθόνη ενός υπολογιστή. Σε κάθε εικονοστοιχείο στην ουσία αντιστοιχεί ένας αριθμός 0-255 (μαύρο->άσπρο) σε περίπτωση A/M εικόνας, ή 3 αριθμοί 0-255 σε περίπτωση έγχρωμης εικόνας, όπου κάθε χρώμα αντιστοιχεί στην ένταση του Κόκκινου, του Πράσινου και του Μπλέ χρώματος. (σύστημα RGB).

Χρωματίζοντας κατάλληλα κάθε εικονοστοιχείο, αποτυπώνουμε την αρχική εικόνα που θέλουμε να αναπαραστήσουμε σε ψηφιακή μορφή.

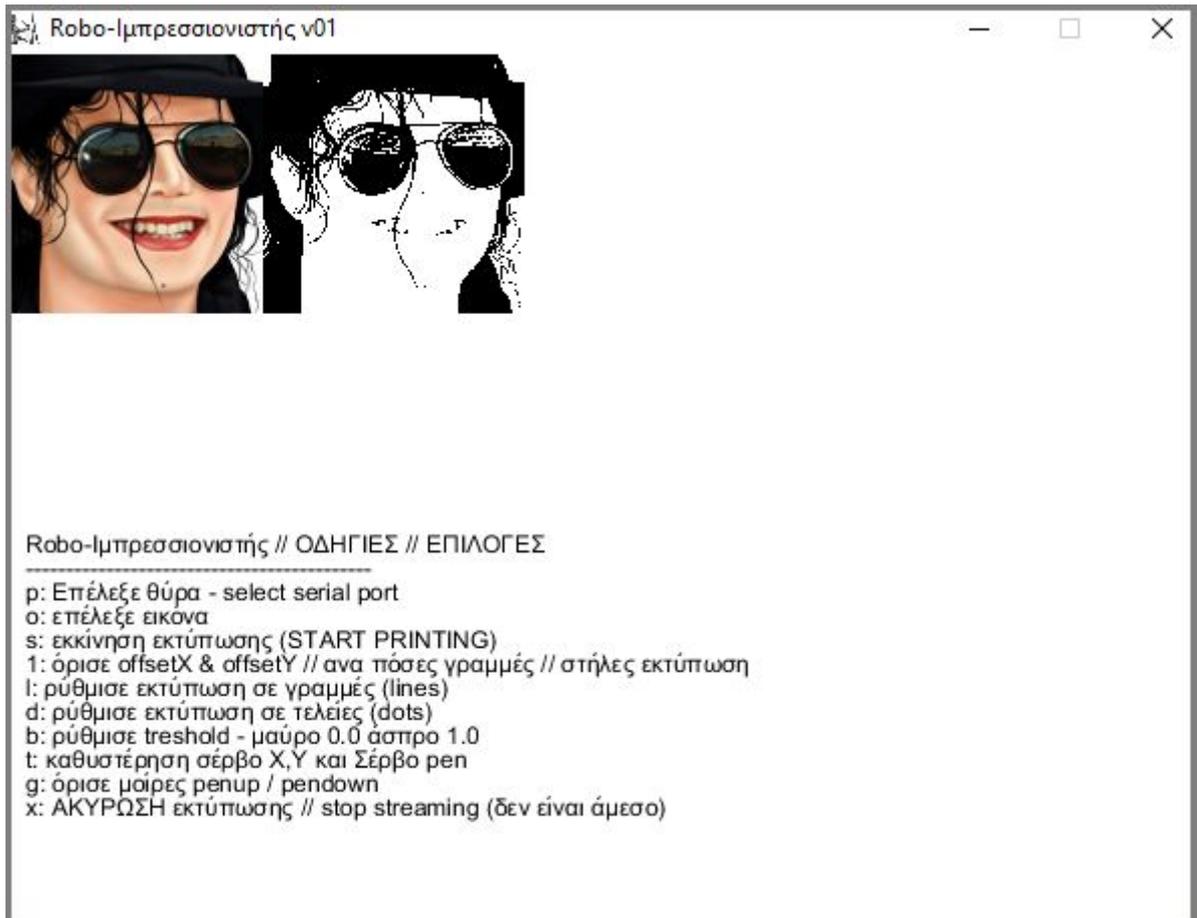


εικόνα, πίνακας εικονοστοιχείων και πίνακας χρώματος Pixel μονόχρωμης εικόνας



χρωματίζοντας κάθε εικονοστοιχείο ξεχωριστά μπορούμε να πετύχουμε μια αποτύπωση της εικόνας που επιθυμούμε.

Περιγραφή εφαρμογής / λογισμικού σε processing



κεντρική οθόνης εφαρμογής

Στόχος της εφαρμογής μας σε processing είναι να επιλέξουμε την εικόνα που θέλουμε να εκτυπώσουμε, και να δημιουργήσουμε ξεχωριστά για κάθε εικονοστοιχείο τις γωνίες που θέλουμε να κινηθεί ο κάθε ένας από τους 2 σερβοκινητήρες που κινούν τους βραχίονες, ώστε να φτάσει η γραφίδα (3ος σερβοκινητήρας) στον προορισμό του, όπου και ανάλογα με το χρώμα του 0 (μαύρο) ή 255 (άσπρο) να επιλέξει αν κάνει μια “πινελιά” στην συγκεκριμένη περιοχή. Με τον τρόπο αυτόν θα επισκεφθούμε κάθε ένα εικονοστοιχείο της εικόνας μας ξεχωριστά.

Για να το επιτύχουμε αυτό, έχουμε τα εξής στάδια:

1. Αλλάζουμε το μέγεθος της αρχικής μας εικόνας σε 300 X 300 pixels (συνήθως οι εικόνες είναι πολύ μεγαλύτερες) ώστε να μπορέσουμε γρηγορότερα να επιτύχουμε την εκτύπωση
2. Στην συνέχεια μετατρέπουμε την εικόνα σε Ασπρόμαυρη χάρη απλότητας και λόγω του ότι ο βραχίονας μας εκτυπώνει με ένα μολύβι / στυλό / μαρκαδόρο. Για να το πετύχουμε χρησιμοποιήσαμε έναν αλγόριθμο αναγνώρισης ακμών (Edge detecting)

ώστε να τονίσουμε και περισσότερο τα χρώματα που “πλησιάζουν” στο μαύρο (0), και να απαλύνουμε τα χρώματα που πλησιάζουν στο άσπρο (255). Περισσότερα: <https://processing.org/examples/edgedetection.html>



3. Επόμενη βήμα κατ' επιλογή του χρήστη είναι να εφαρμοσθούν κάποια φίλτρα φωτεινότητας αντίθεσης, ώστε να επιτύχει αν επιθυμεί κάθε φορά διαφορετικό αποτέλεσμα.
4. Πλέον η εικόνα μας έχει σε κάθε εικονοστοιχείο έναν αριθμό και μόνο. 0 για μαύρο, ή 255 για άσπρο, αφού έχουμε εξαλείψει κάθε ενδιάμεση απόχρωση του γκρι 1-244 και την έχουμε ενσωματώσει στο πλησιέστερο άσπρο ή μαύρο αντίστοιχα.
5. Το τελευταίο στάδιο είναι να επισκεφθούμε κάθε εικονοστοιχείο ξεχωριστά. Ο αλγόριθμος μας σαρώνει την εικόνα και κάθε εικονοστοιχείο της. Το κάθε εικονοστοιχείο έχει τις συντεταγμένες του στο σύστημα καρτεσιανών συντεταγμένων. Για να φτάσω σε αυτό, αρκεί να υπολογίσω την γωνία (ω) που πρέπει να στραφεί ο 1ος κινητήρας και την γωνία (ϕ) που πρέπει να στραφεί ο 2ος κινητήρας σέρβο (Inverse kinematics / Αντίστροφη κινηματική) και για τον λόγο αυτόν χρησιμοποιήσαμε 4 τύπους από τα μαθηματικά που παραθέσαμε σε προηγούμενη παράγραφο. Μόλις επισκέπτομαι κάθε εικονοστοιχείο, κάθε σημείο του καρτεσιανού συστήματος συντεταγμένων μας, ελέγχω την τιμή του εικονοστοιχείου στο οποίο αντιστοιχεί. Σε περίπτωση που αυτό είναι μαύρο, ο 3ος κινητήρας κατεβάζει και στη συνέχεια ξανα ανεβάζει τη γραφίδα μας, δημιουργώντας μια “πινελιά” στην περιοχή αυτήν, δίνοντας την εικόνα και την τεχνοτροπία ενός ιμπρεσσιονιστή, από όπου και δώσαμε το όνομα στο ρομπότ μας. Αν η περιοχή που επισκεπτόμαστε δεν αντιστοιχεί σε μαύρο αλλά σε άσπρο εικονοστοιχείο, δεν χρειάζεται να δημιουργήσουμε μια “πινελιά” και συνεχίζουμε παρακάτω. Με τον τρόπο αυτόν και αφού τελειώσουμε τη σάρωση και των 300X300 εικονοστοιχείων της δισδιάστατης εικόνας μας, έχουμε “εκτυπώσει” μια αναπαράσταση της αρχικής εικόνας μας. Στην ουσία η γλώσσα processing σε αυτό το στάδιο δεν κινεί του κινητήρες βέβαια, αλλά υπολογίζει την γωνία ω , την γωνία ϕ και το αν χρειάζεται να κάνει μια πινελιά ή όχι στην συγκεκριμένη περιοχή, και αποστέλει στο λογισμικό οδήγησης (Arduino) τις πληροφορίες αυτές μέσω σειριακής επικοινωνίας.

Περιγραφή λογισμικού ελέγχου (firmware) arduino

Το λογισμικό οδήγησης μας μιας και ο επεξεργαστής (εγκέφαλος) του ρομπότ μας είναι το Arduino Uno, δημιουργήθηκε στο περιβάλλον Arduino IDE που με στη σειρά του είναι μια απλή μορφή της γλώσσας C++, όπως και η Processing.

Η λειτουργία του κώδικα είναι πολύ απλή. Περιμένουμε σειριακά εντολές που περιέχουν 3 πληροφορίες: Γωνία 1ου κινητήρα, Γωνία 2ου κινητήρα και αν είναι απαραίτητη η κίνηση του 3ου κινητήρα (γραφίδας). Στη συνέχεια, αναγκάζουμε τον κάθε κινητήρα από τους 3 να εκτελέσει την συγκεκριμένη κίνηση και περιμένουμε την επόμενη εντολή, μέχρι να σαρωθεί κάθε ένα εικονοστοιχείο ξεχωριστά και να δημιουργηθεί η τελική εικόνα μας.

Τομείς εφαρμογής

Τέχνες, Ζωγραφική, αλλά και σαν μηχανήμα CNC με χρήση καλύτερων σερβο-κινητήρων με μεγαλύτερη ακρίβεια.

Από τη στιγμή που μπορεί να “ζωγραφίσει” σε οποιαδήποτε επιφάνεια και με οποιοδήποτε μέσω, είναι ιδανικός να αποτυπώνει σαν πραγματικός καλλιτέχνης με πινέλο σε καμβά, ιδίως αν δώσουμε ενδιάμεσα σε κάθε εντολή κίνησης της γραφίδας, μια κίνηση που θα “γεμίζει” το πινέλο με χρώμα από μια συγκεκριμένη θέση που θα βρίσκεται η παλέτα χρωμάτων μας, επιτυγχάνοντας έτσι και έγχρωμη σχεδίαση.

Βελτιώσεις - παρατηρήσεις

- Με χρήση δυνατότερων κινητήρων θα μπορούσε να αποτελέσει ένα αξιόπιστο CNC μηχανήμα.
- Με χρήση αλγορίθμων βελτιστοποίησης ταχύτητας θα μπορούσαμε να επιτύχουμε πολύ καλύτερους χρόνους εκτύπωσης.
- Στην υλοποίηση μας χρησιμοποιήσαμε πλαστικά παλαιά καναλάκια καλωδίων έχουν κάποια ελαστικότητα. Με χρήση καλύτερων υλικών και μεθόδων δημιουργίας των βραχιόνων κίνησης, το αποτέλεσμα θα μπορούσε να βελτιωθεί πολύ.
- Προσθήκη και άλλων τρόπων ζωγραφικής, όπως πχ αντί για πινελιές, κάθετη εκτύπωση, απελευθερώνοντας κάθετα την γραφίδα, όπου το αποτέλεσμα θα ήταν pixel, pixel και όχι πινελιά πινελιά.

παράρτημα: φωτογραφίες

φωτογραφίες από δραστηριότητές μας και από το “ταξίδι” μας μέχρι την ολοκλήρωση της κατασκευής μας.



Η ομάδα Ρομποτικής του σχολείου στο πολυτεχνείο Βελιγραδίου τμήμα Ηλεκτρολόγων μηχανικών μαζί με τον υπεύθυνο καθηγητή κο Κosta Ιονανovic και τον Νικόλα Τέσλα







European
Maker Week

Promoted and implemented by



2/6
1ο ΓΕΛ Φθιώρινας
12:00'

hacker's lab workshop

Δημιουργία mini CNC (ενός άξονα)
από ανακυκλωμένα υλικά.

- Βηματικά μοτέρ // λειτουργία τύποι μονοπολικά - διπολικά // συνδεσμολογία
- Gcode
- Δημιουργία Gcode Interpreter σε arduino
- Σειριακή επικοινωνία Processing // arduino.

Main partner

In collaboration with



