Robotics & Automation Society - IEEE NTUA SB

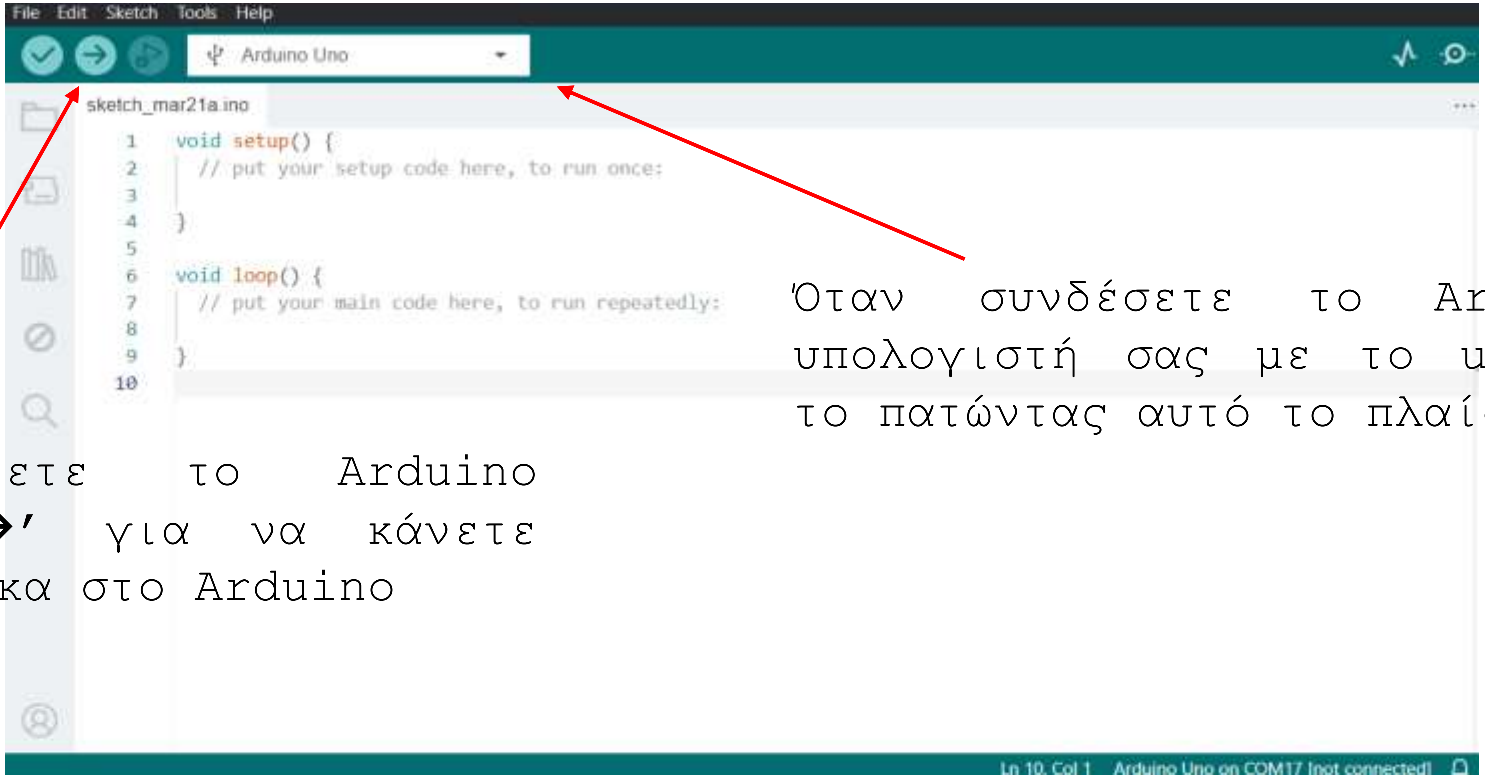
Αυτόνομη Κίνηση Ρομποτικού Βραχίονα

Από τη Θεωρία στην Πράξη

ROBOTALK 2025

Arduino IDE

Εύκολη διασύνδεση Arduino - Υπολογιστή

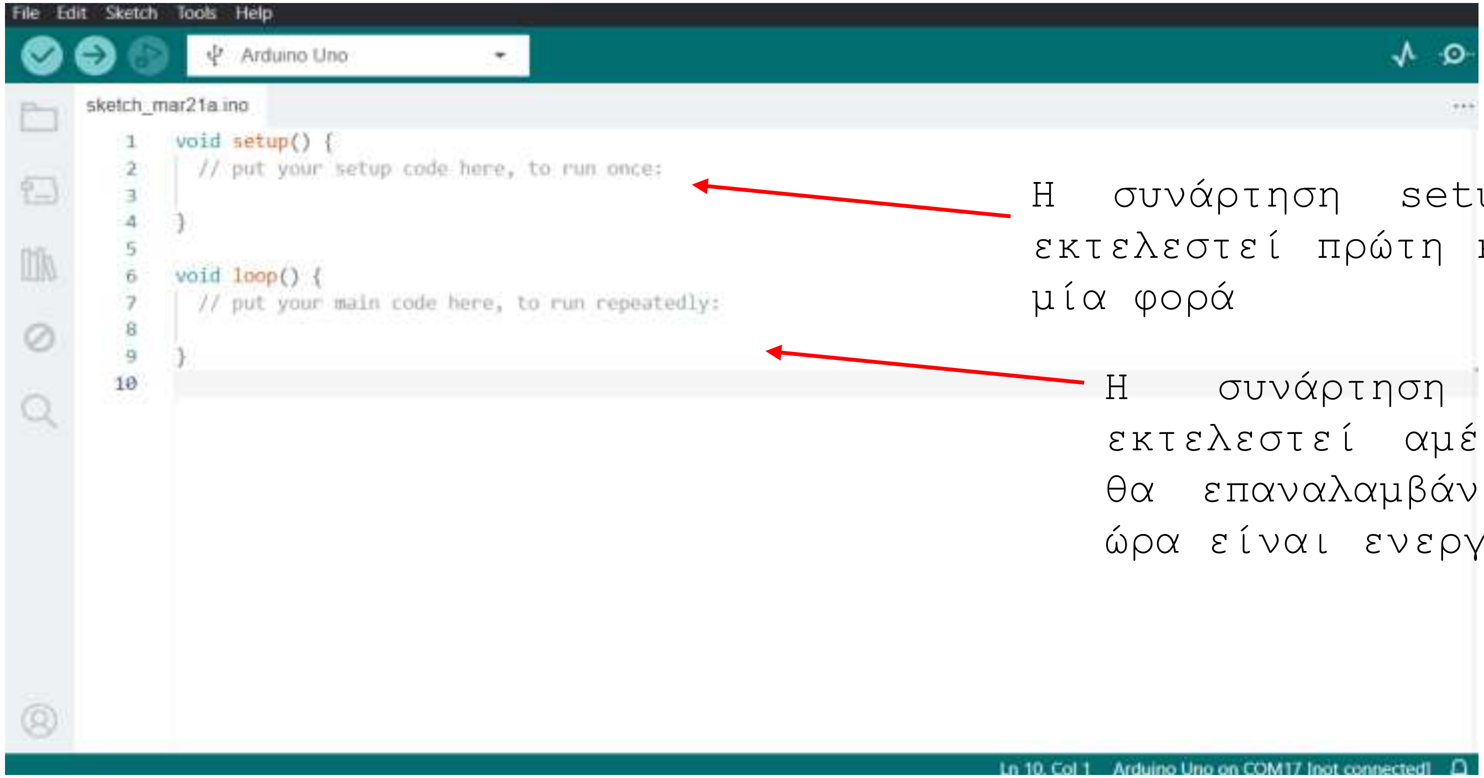


Όταν συνδέσετε το Arduino στον υπολογιστή σας με το usb διαλέξτε το πατώντας αυτό το πλαίσιο

Αφότου επιλέξετε το Arduino πατήστε το '→' για να κάνετε Upload τον κώδικα στο Arduino

Arduino IDE

Το Arduino IDE μας δίνει την δυνατότητα να γράφουμε προγράμματα σε γλώσσα C/C++, τα οποία στην συνέχεια τρέχουν στον μικροελεγκτή Arduino UNO.



Η συνάρτηση setup() θα εκτελεστεί πρώτη και μόνο μία φορά

Η συνάρτηση loop() θα εκτελεστεί αμέσως μετά και θα επαναλαμβάνεται για όση ώρα είναι ενεργό το Arduino

Ανοίξτε το
RoboTalk 2025 Code Schema

File Edit Sketch Tools Help

Arduino Uno

RoboTalk_2025_Code_Schema.ino

```
#include <Servo.h>

Servo joint_1, joint_2;
float q1, q2;
float x[20];
float y[20];
int size;
String data;

// Parameters
float l1=10;
float l2=9.5;


/* ------------------------------ FUNCTIONS ------------------------------ */

void Inverse_Kin(float px, float py) {

}

void draw_line(float x_start, float y_start, float x_end, float y_end, float u) {

}

/* ------------------------------ SETUP ------------------------------ */

void setup() {
  joint_1.attach(9);
  joint_2.attach(10);

  Inverse_Kin(0,10);

  Serial.begin(9600);
}

/* ------------------------------ LOOP ------------------------------ */

void loop() {
  //---------- COMMUNICATION ----------//
  size = 0;
  while (true) {
    if (Serial.available() > 0) {
      data = Serial.readStringUntil('\n');  // Read until newline
      if (data == "stop"){
        Serial.println("ACK");
        break;
      }

      int commaIndex = data.indexOf(',');
      if(commaIndex != -1) {
        Serial.println("ACK");

        x[size] = data.substring(0, commaIndex).toFloat();
        y[size] = data.substring(commaIndex + 1).toFloat();
        size++;
      }
    }
  }
  //---------- KINEMATICS ----------//

}
```

Στην συνέχεια θα συμπληρώσουμε τις συναρτήσεις **Inverse_Kin()** και **draw_line()**, και θα προσθέσουμε κώδικα κάτω από το section **KINEMATICS** στην loop().

Robotic Arms

Βιομηχανία

Ιατρική

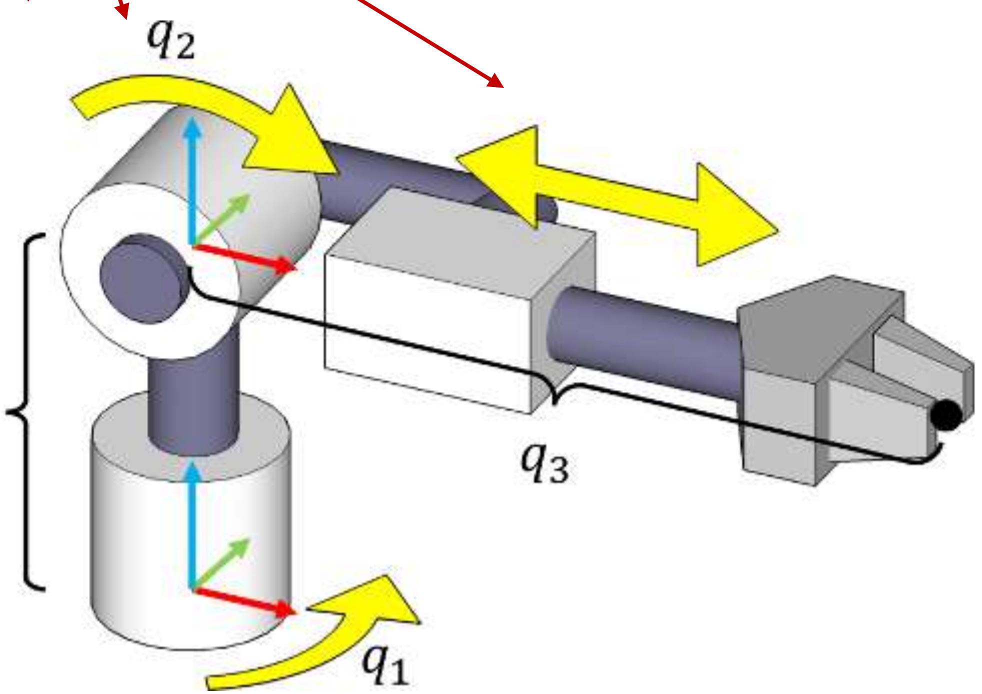
Αρθρώσεις

Οι αρθρώσεις είναι τα σημεία του ρομποτικού βραχίονα που του επιτρέπουν να κινηθεί με ένα συγκεκριμένο τρόπο

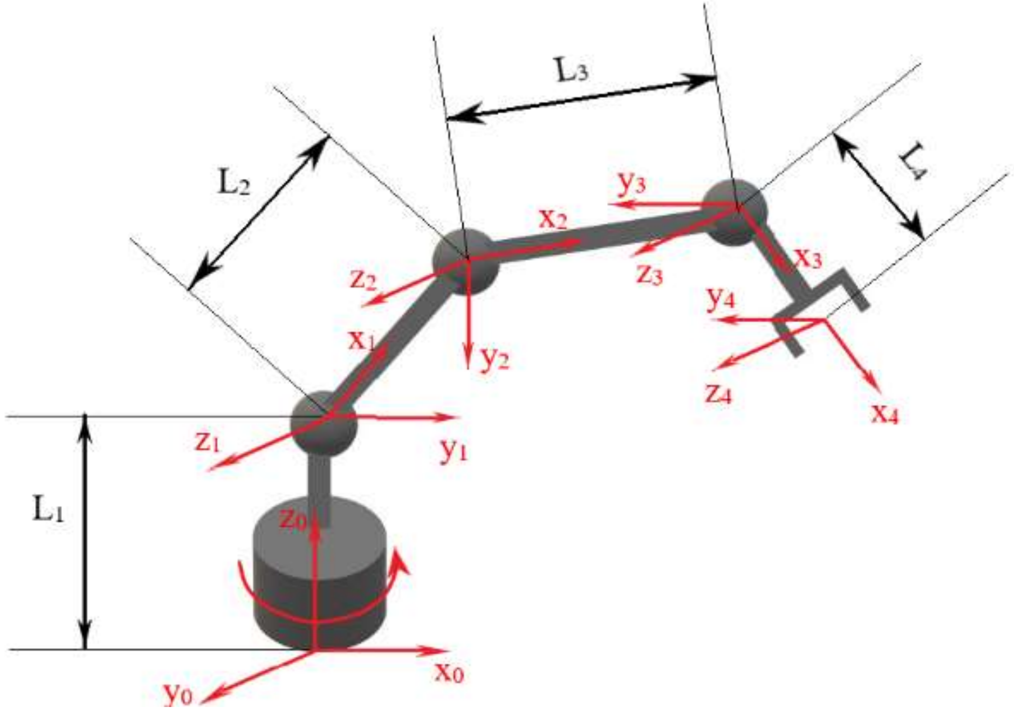
Περιστροφική Άρθρωση

Πρισματική Άρθρωση

- **Πρισματική Άρθρωση:** Μετακίνηση κατά μήκος ενός άξονα, βαθμός ελευθερίας = μετατόπιση
- **Περιστροφική Άρθρωση:** Περιστροφή γύρω από έναν άξονα βαθμός ελευθερίας = γωνία



Κινηματική Ανάλυση



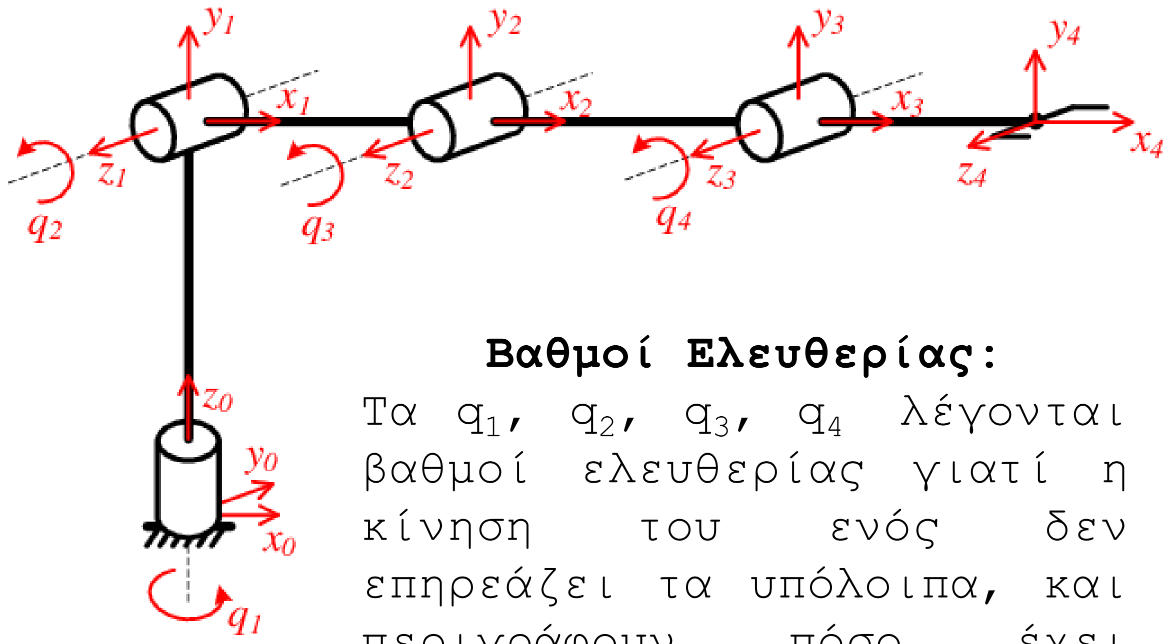
Πλαίσια Αναφοράς:

Δίνουν την πληροφορία της θέσης και τοποθέτησης στον χώρο των αρθρώσεων και του τελικού εργαλείου δράσης του βραχίονα

<u>Base Frame:</u> $O_0-x_0y_0z_0$

<u>End Effector Frame:</u> $O_4-x_4y_4z_4$

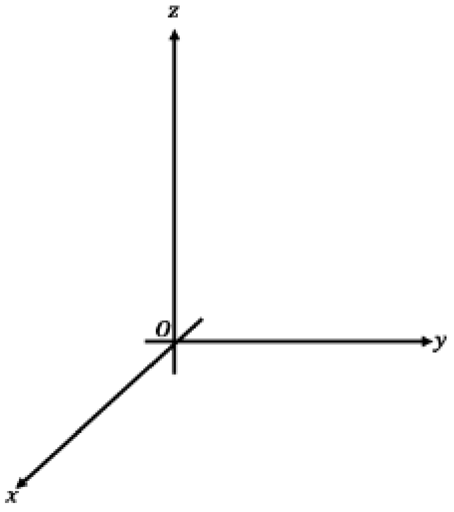
<u>Ενδιάμεσα Frames:</u> $O_1-x_1y_1z_1$, O_2-



Βαθμοί Ελευθερίας:

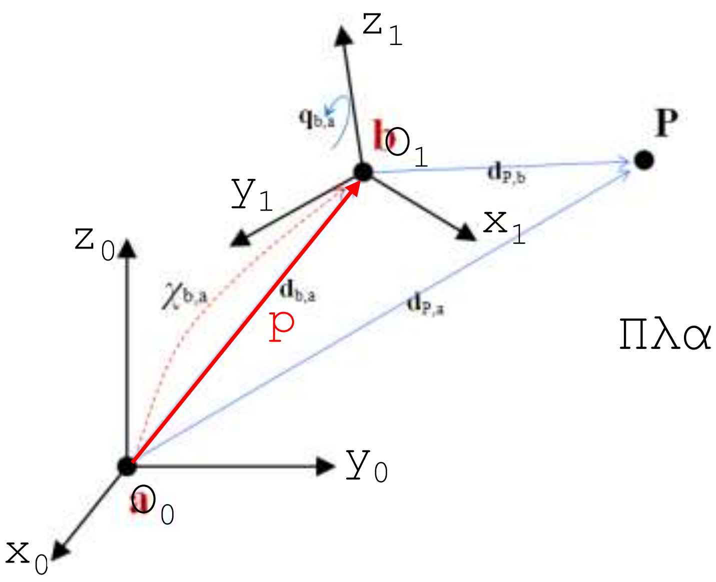
Τα q_1, q_2, q_3, q_4 λέγονται βαθμοί ελευθερίας γιατί η κίνηση του ενός δεν επηρεάζει τα υπόλοιπα, και περιγράφουν πόσο έχει μεταβληθεί μία άρθρωση

Πλαίσια Αναφοράς



$$R = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \qquad z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

Ορθογώνιος Πίνακας: Οι στήλες του είναι κάθετες ανά 2



Πλαίσιο O_1-$x_1y_1z_1$

Θέση: $p = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}$

Προσανατολισμός $R_1^0 = \begin{bmatrix} x_{11} & y_{11} & z_{11} \\ x_{12} & y_{12} & z_{12} \\ x_{13} & y_{13} & z_{13} \end{bmatrix}$

Ομογενής Μετασχηματισμός

Κρατάμε και την πληροφορία της **Θέσης** και του **Προσανατολισμού** με την ομογενή μήτρα:

$$
\begin{array}{cc}
 & \begin{array}{cc} \text{Προσανατολισμός} & \text{Θέση} \end{array} \\
A_1^0 = & \left[\begin{array}{ccc|c} x_{11} & y_{11} & z_{11} & p1 \\ x_{12} & y_{12} & z_{12} & p2 \\ x_{13} & y_{13} & z_{13} & p3 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]
\end{array}
$$

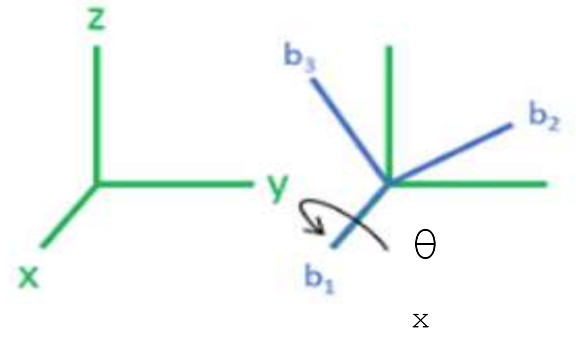
! Ο πολλαπλασιασμός δύο ομογενών πινάκων είναι ένας ομογενής

πίνα

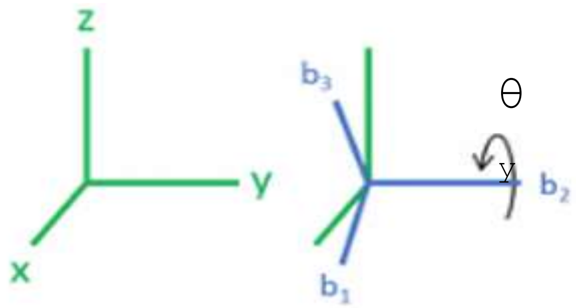
$$
\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & A_1 \\ a_{21} & a_{22} & a_{23} & A_2 \\ a_{31} & a_{32} & a_{33} & A_3 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \cdot \left[\begin{array}{ccc|c} b_{11} & b_{12} & b_{13} & B_1 \\ b_{21} & b_{22} & b_{23} & B_2 \\ b_{31} & b_{32} & b_{33} & B_3 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc|c} c_{11} & c_{12} & c_{13} & C_1 \\ c_{21} & c_{22} & c_{23} & C_2 \\ c_{31} & c_{32} & c_{33} & C_3 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]
$$

Απλοί Μετασχηματισμοί: Περιστροφές

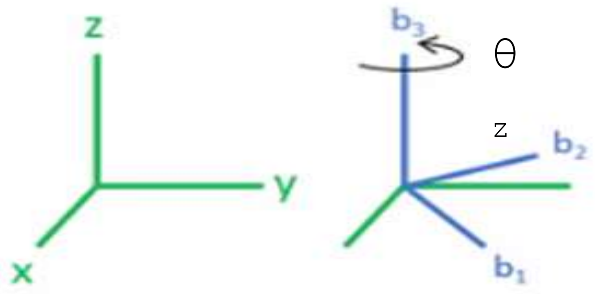
$$\mathrm{Rot}(x,\boldsymbol{\theta}_x) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\boldsymbol{\theta}_x & -\sin\boldsymbol{\theta}_x & 0 \\ 0 & \sin\boldsymbol{\theta}_x & \cos\boldsymbol{\theta}_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



$$\mathrm{Rot}(y,\boldsymbol{\theta}_y) = \begin{pmatrix} \cos\boldsymbol{\theta}_y & 0 & \sin\boldsymbol{\theta}_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\boldsymbol{\theta}_y & 0 & \cos\boldsymbol{\theta}_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

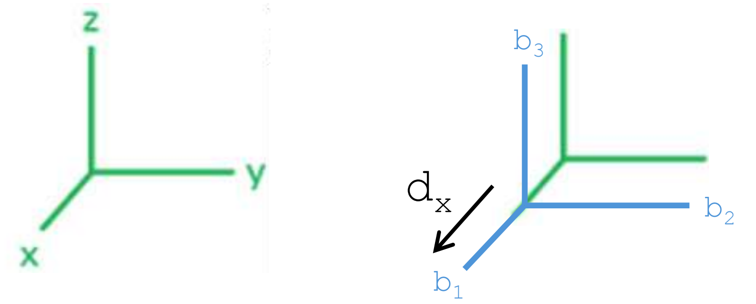


$$\mathrm{Rot}(z,\boldsymbol{\theta}_z) = \begin{pmatrix} \cos\boldsymbol{\theta}_z & -\sin\boldsymbol{\theta}_z & 0 & 0 \\ \sin\boldsymbol{\theta}_z & \cos\boldsymbol{\theta}_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

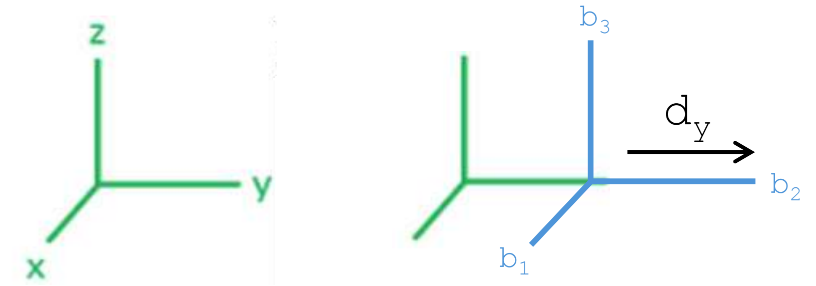


Απλοί Μετασχηματισμοί: Μετατοπίσεις

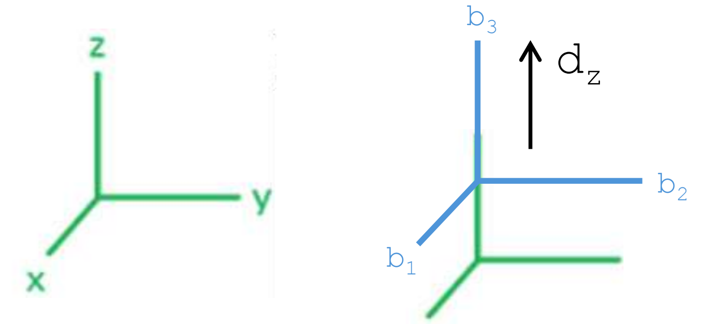
$$\mathrm{Tra}(x,\mathbf{d}_x) = \begin{pmatrix} 1 & 0 & 0 & \mathbf{d}_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



$$\mathrm{Tra}(y,\mathbf{d}_y) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \mathbf{d}_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



$$\mathrm{Tra}(z,\mathbf{d}_z) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \mathbf{d}_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



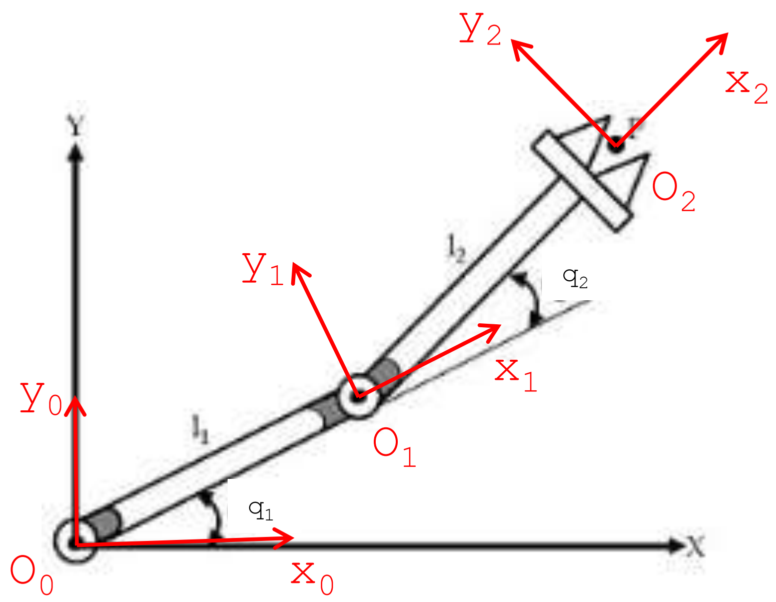
Ευθεία Κινηματική Ανάλυση

Μετατοπίσεις αρθρώσεων $\{q_i\}$ → Μετατόπιση του End Effector (Θέση **p** και Προσανατολισμός **R**)

Μπορούμε να βρούμε θέση και προσανατολισμό όλων των ενδιαμέσων frames και του frame του end effector, ως προς το Base frame, πολλαπλασιάζοντας διαδοχικά τους πίνακες ομογενών μετασχηματισμών

$$A_n^0(q) = A_1^0(q_1) \cdot A_2^1(q_2) \cdot \dots \cdot A_{n-1}^{n-2}(q_{n-1}) \cdot A_n^{n-1}(q_n)$$

Βραχίονας «Ζωγράφος»



Μετάβαση από το $O_0-x_0y_0\ z_0$ στο $O_1-x_1y_1\ z_1$

$$A_1^0(q_1) = Rot(z, q_1) \cdot Tra(x, l_1) = \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 & 0 \\ \sin q_1 & \cos q_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Longrightarrow A_1^0(q_1) = \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 & l_1 \cdot \cos q_1 \\ \sin q_1 & \cos q_1 & 0 & l_1 \cdot \sin q_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Όμοια

$$A_2^1(q_2) = \begin{bmatrix} \cos q_2 & -\sin q_2 & 0 & l_2 \cdot \cos q_2 \\ \sin q_2 & \cos q_2 & 0 & l_2 \cdot \sin q_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Και άρα

$$A_2^0(q_1, q_2) = A_1^0(q_1) \cdot A_2^1(q_2) = \begin{bmatrix} \cos(q_1+q_2) & -\sin(q_1+q_2) & 0 & l_2 \cdot \cos(q_1+q_2) + l_1 \cdot \cos q_1 \\ \sin(q_1+q_2) & \cos(q_1+q_2) & 0 & l_2 \cdot \sin(q_1+q_2) + l_1 \cdot \sin q_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Αντίστροφη Κινηματική Ανάλυση

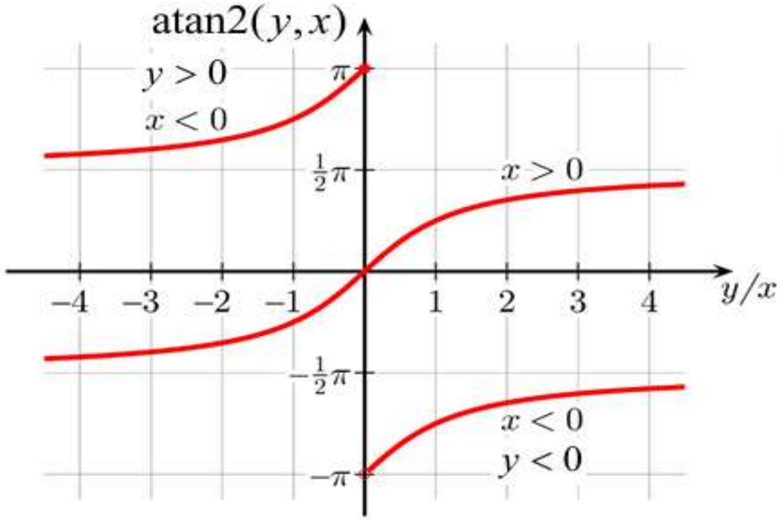
Μετατόπιση του End Effector → Μετατοπίσεις αρθρώσεων $\{q_i\}$ (Θέση **p** και Προσανατολισμός **R**)

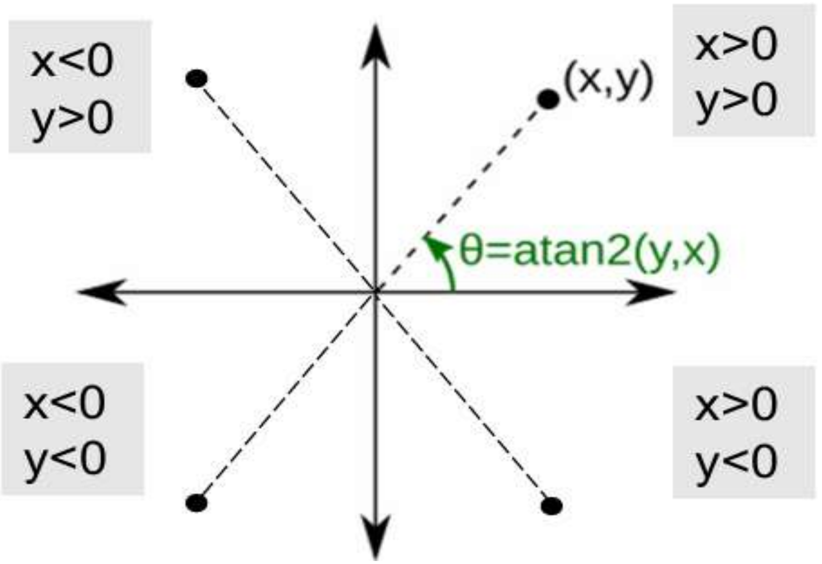
Εν γένει είναι δύσκολο πρόβλημα γιατί:

- Οι εξισώσεις είναι συνήθως μη γραμμικές
- Μπορεί να υπάρχουν περισσότερες από μία λύσεις $\{q_i\}$
- Μπορεί να υπάρχει απειρία λύσεων
- Μπορεί να μην υπάρχει λύση στο πρόβλημα (π.χ. όταν η επιθυμητή θέση δεν είναι προσιτή από τον βραχίονα)

Atan2(y , x)

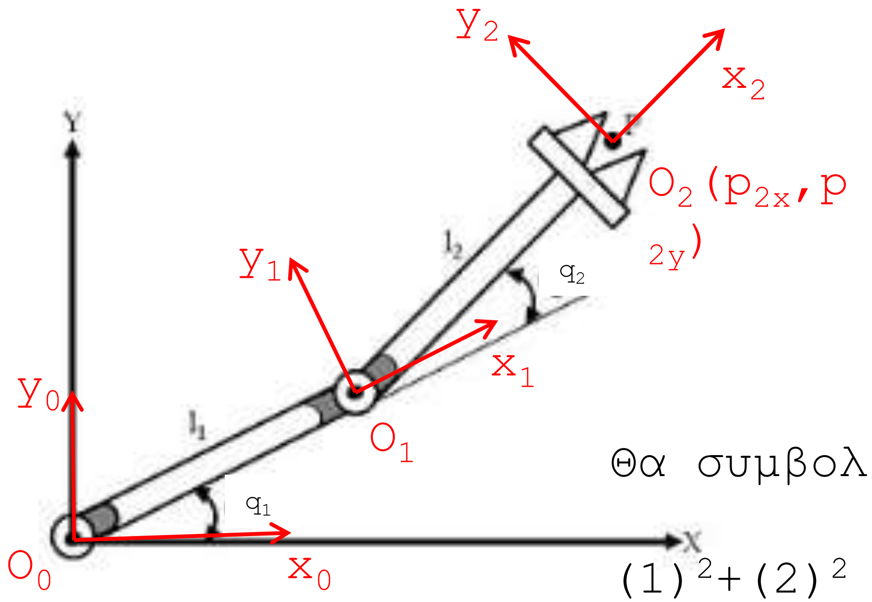
$$\operatorname{atan2}(y,x) = \begin{cases} \arctan(\frac{y}{x}) & \text{if } x > 0, \\ \arctan(\frac{y}{x}) + \pi & \text{if } x < 0 \text{ and } y \geq 0, \\ \arctan(\frac{y}{x}) - \pi & \text{if } x < 0 \text{ and } y < 0, \\ +\frac{\pi}{2} & \text{if } x = 0 \text{ and } y > 0, \\ -\frac{\pi}{2} & \text{if } x = 0 \text{ and } y < 0, \\ \text{undefined} & \text{if } x = 0 \text{ and } y = 0. \end{cases}$$





$$\begin{cases} \arctan(t) : \mathbf{R} \to \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \\ \operatorname{atan2}(y,x) : \mathbf{R}^2 \to (-\pi, \pi] \end{cases}$$

Βραχίονας «Ζωγράφος»



Γνωρίζουμε: l_1 , l_2 , p_{2x} , p_{2y}

Θέλουμε να βρούμε τα: q_1 , q_2

Από την Ευθεία Κινηματική Ανάλυση

γνωρίζουμε ότι:

(1): $p_{2x} = l_2\cos(q1+q2) + l_1\cos(q_1)$

(2): $p_{2y} = l_2\sin(q1+q2) + l_1\sin(q_1)$

Θα συμβολίσουμε το $\cos(q_i)$ ως c_i και το $\sin(q_i)$ ως s_i

$(1)^2+(2)^2 \rightarrow (p_{2x})^2 + (p_{2y})^2 = (l_2^2 c_{12}^2 + l_1^2 c_1^2 + 2\, l_1 l_2 c_1 c_{12}) + (l_2^2 s_{12}^2 + l_1^2 s_1^2 + 2\, l_1 l_2 s_1 s_{12})$

$\rightarrow (p_{2x})^2 + (p_{2y})^2 = l_1^2 + l_2^2 + 2\, l_1 l_2 \cos(q_2)$

$\rightarrow \cos(q_2) = \dfrac{(p_{2x})^2 + (p_{2y})^2 - l_1^2 - l_2^2}{2\, l_1 l_2}$

2 λύσεις

$\rightarrow q_2 = \pm \arccos\left(\dfrac{(p_{2x})^2 + (p_{2y})^2 - l_1^2 - l_2^2}{2\, l_1 l_2}\right)$

Εναλλακτικά

$$\sin(q_2) = \pm\sqrt{1-\cos(q_2)}$$

2 λύσεις

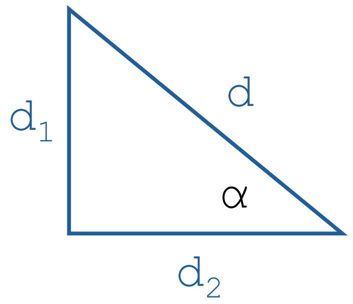
$$\tan(q_2) = \boxed{\frac{\sin(q_2)}{\cos(q_2)} \rightarrow q_2 = \text{atan2}(\ \sin}(q_2)\ ,\ \cos(q_2)\ )$$

Τώρα εφόσον τα c_2 και s_2 είναι γνωστές ποσότητες, για την εύρεση του q_1 έχουμε:

$$(1),(2) \Rightarrow \left\{ \begin{array}{l} (l_1 + l_2\cos q_2)\cos q_1 - (l_2 \sin q_2)\sin q_1 = p_{2x} \\ (l_2 \sin q_2)\cos q_1 + (l_1 + l_2\cos q_2)\sin q_1 = p_{2y} \end{array} \right\} (3)$$

$$\text{Ορίζουμε:} \left\{ \begin{array}{l} d_1 = (l_1 + l_2 \cos q_2) = d\cos\alpha \\ d_2 = (l_2 \sin q_2) = d \sin\alpha \\ d = \sqrt{d_1^2 + d_2^2} \end{array} \right. \quad \Rightarrow \alpha = atan2(d_2, d_1)$$

$$\Rightarrow \alpha = atan2((l_2 \sin q_2), (l_1 + l_2 \cos q_2))$$



Άρα τελικά

$$(3) \Rightarrow \left\{ \begin{array}{l} d\cos(q_1+\alpha) = p_{2x} \\ d\sin(q_1+\alpha) = p_{2y} \end{array} \right\} \Rightarrow \boxed{q_1 = atan2(p_{2y}, p_{2y}) - \alpha}$$

1 λύση για κάθε μία από τις 2 λύσεις του q_2

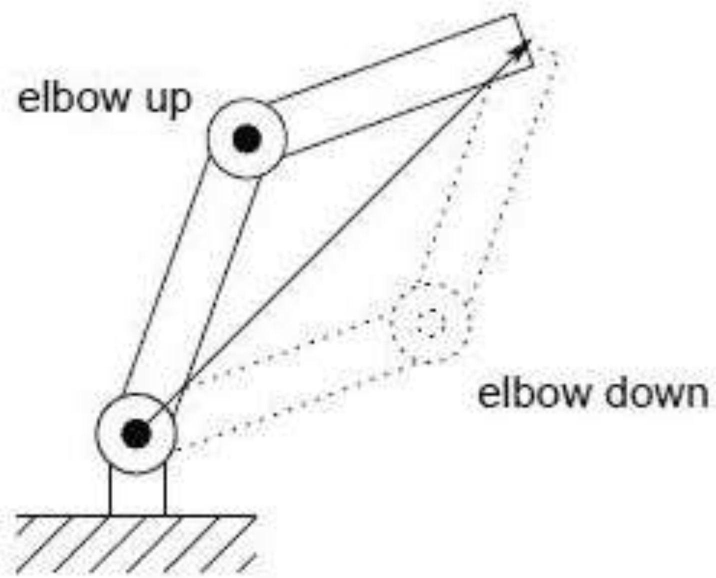
2 Λύσεις

Elbow Up

Για $\sin(q_2)$ = $-\sqrt{1-\cos(q_2)}$

Elbow Down

Για $\sin(q_2)$ = +



Υλοποίηση της Inverse_Kin()

Αυτή η συνάρτηση πρέπει να παίρνει ως παραμέτρους τις συντεταγμένες του σημείου στο οποίο θέλουμε να πάει ο βραχίονας και, αφού υπολογίσει την αντίστροφη κινηματική, να γράφει τα κατάλληλα q_1, q_2 στα servo motors.

Εξισώσεις Αντίστροφης Κινηματικής

$$\cos(q_2) = \frac{(p_{2x})^2 + (p_{2y})^2 - l_1^2 - l_2^2}{2 l_1 l_2}$$

$$\sin(q_2) = \pm\sqrt{1 - \cos(q_2)}$$

$$q_2 = atan2(\sin(q_2), \cos(q_2))$$

$$d_1 = (l_1 + l_2 \cos q_2)$$

$$d_2 = (l_2 \sin q_2)$$

$$\alpha = atan2(d_2, d_1)$$

$$q_1 = atan2(p_{2y}, p_{2y}) - \alpha$$

Συναρτήσεις του Arduino

`pow(a, 2)` → υπολογίζει: a^2

`constrain(cosq2,-1.0,1.0)` → περιορίζει το cosq2 στο (-1.0, 1.0) για να αποφευχθούν bugs

`sqrt(a)` → υπολογίζει: $\sqrt{a}$

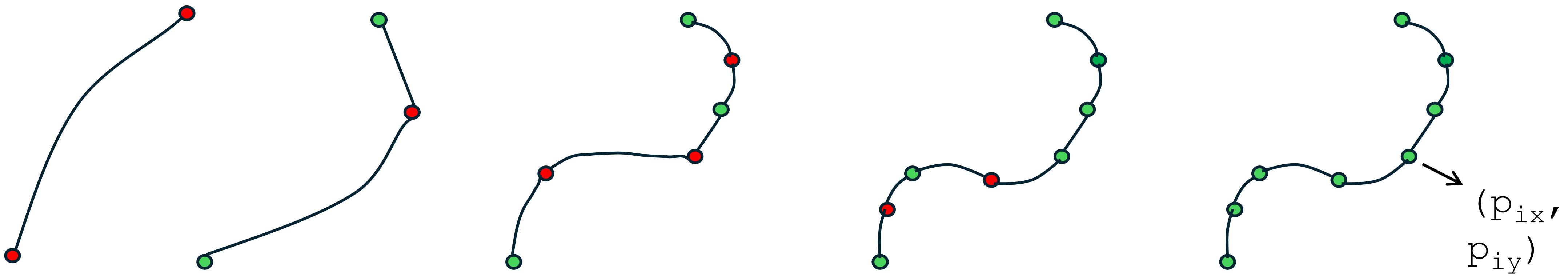
`PI` → είναι η σταθερά $\pi = 3.14 \ldots$ και θα σας βοηθήσει στην μετατροπή rad σε μοίρες

`atan2(y, x)` → υπολογίζει: atan2(y,x)

`joint_1.write(q1)` → γράφει στο servo 1 την γωνία q_1

Σχεδιασμός Τροχιάς

Η τροχιά είναι ένα σύνολο σημείων από τα οποία επιθυμούμε να “περάσει” ο βραχίονας.



Επιθυμητή Διαδρομή στο επίπεδο

$(p_{ix}, p_{iy}) \longrightarrow$ **Αντίστροφη Κινηματική** $\longrightarrow (q_{i1}, q_{i2})$

Σχεδιασμός Ευθείας

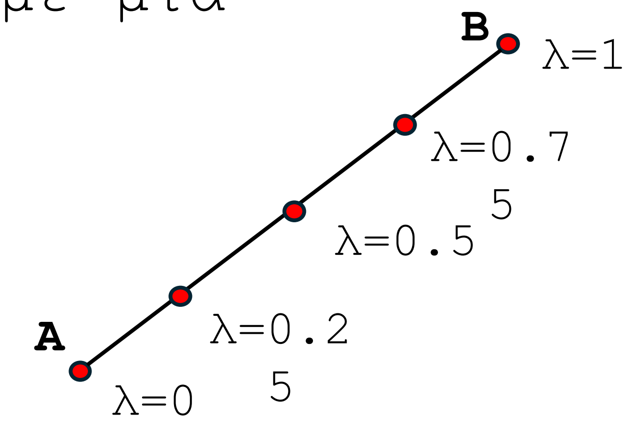
Έστω ότι θέλουμε ο βραχίονας να ζωγραφίσει το ευθύγραμμο τμήμα ΑΒ

Παραμετροποιούμε τα σημεία της ευθείας με μία παράμετρο λ :

$$p(\lambda) = (1-\lambda)\cdot p_A + \lambda\cdot p_B$$

ή

$$p(\lambda) = p_A + \lambda\cdot(p_B - p_A)$$



Καθώς το λ πηγαίνει από 0 έως 1 κινούμαστε πάνω στο ευθύγραμμο τμήμα από το Α έως το Β. Το λ μπορεί να είναι συνάρτηση του χρόνου και αναλόγως με το πόσο γρήγορα κινείται από το 0 στο 1 το ίδιο γρήγορα μεταβαίνει ο ρομποτικός βραχίονας από το Α στο Β.

Υλοποίηση της draw_line()

Αυτή η συνάρτηση θα γράφει διαδοχικά τις κατάλληλες γωνίες στα servo motors, έτσι ώστε ο βραχίονας να περνάει από σημεία της ευθείας. **Η παράμετρος u είναι το λ των εξισώσεων!**

Εξισώσεις Σημείων ευθύγραμμου Τμήματος

$$p_x = p_{Ax} + \lambda \cdot (p_{Bx} - p_{Ax})$$

$$p_y = p_{Ay} + \lambda \cdot (p_{By} - p_{Ay})$$

Συναρτήσεις του Arduino

`Inverse_Kin(px, py)` → Η συνάρτηση που φτιάξαμε

`for(…) {…}` → Ακριβώς όπως στην C/C++

`delay(100)` → Καλέστε αυτή την συνάρτηση αμέσως μετά την Inverse_Kin()

Διάβασμα Σημείων από το Painter App

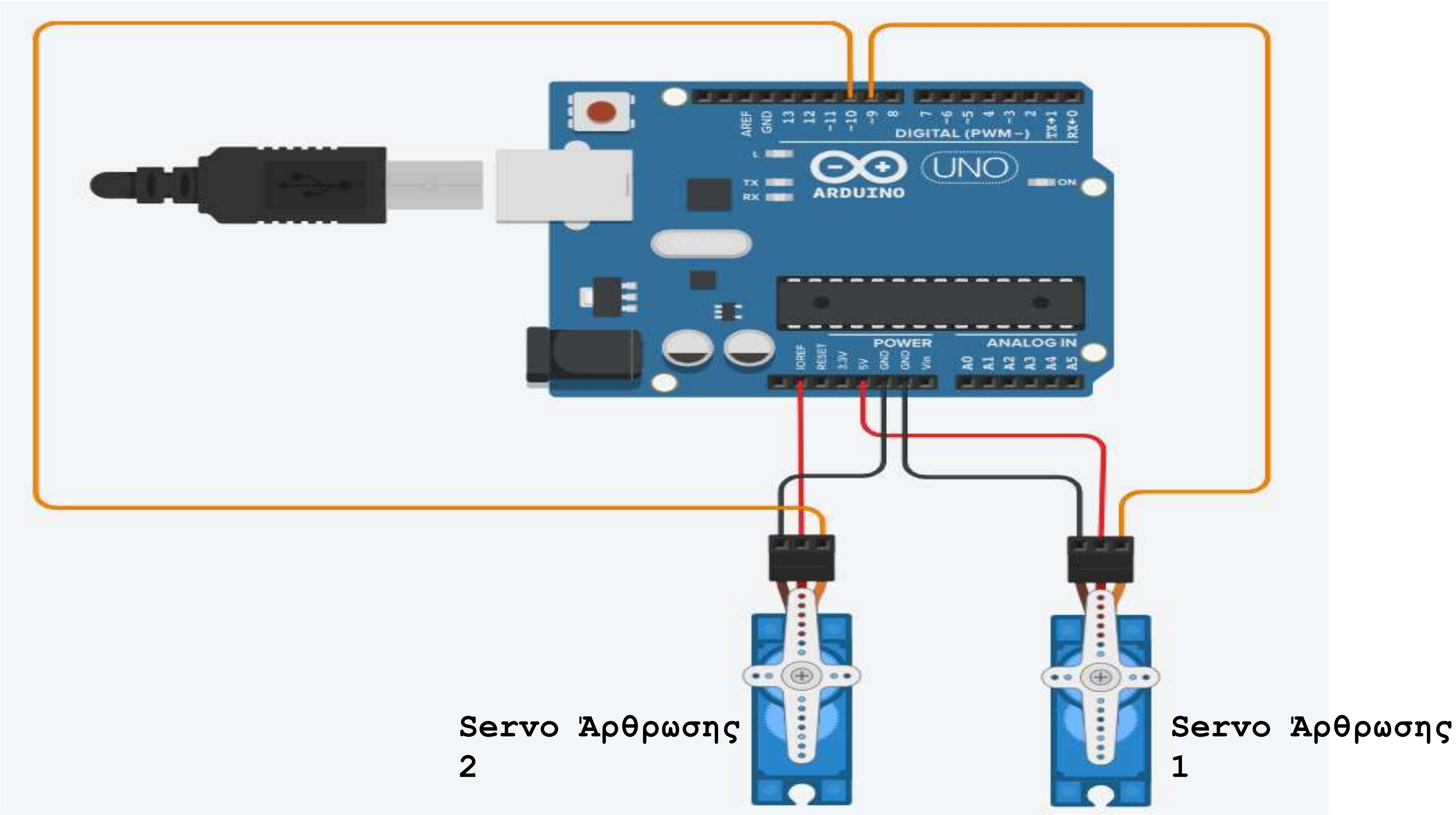
Στο section "COMMUNICATION" του loop() έχουμε υλοποιήσει πρόγραμμα που διαβάζει τα σημεία που θα ζωγραφίσετε στο Painter App, και τα αποθηκεύει στα arrays x[20], y[20] διαδοχικά από το πρώτο στο τελευταίο. Προσέξτε ότι έχουμε default μέγεθος 20 σημεία.

→Φτιάξτε ένα for loop στο section "KINEMATICS" που θα ζωγραφίζει διαδοχικά τα ευθύγραμμα τμήματα του σκίτσου σας καλώντας την συνάρτηση **draw_line()**. Επίσης καλό θα ήταν πριν το for loop να γράψετε:

```
Inverse_Kin(x[0], y[0]);
delay(6000);
```

Για να έχεις χρόνο να τοποθετήσεις τον

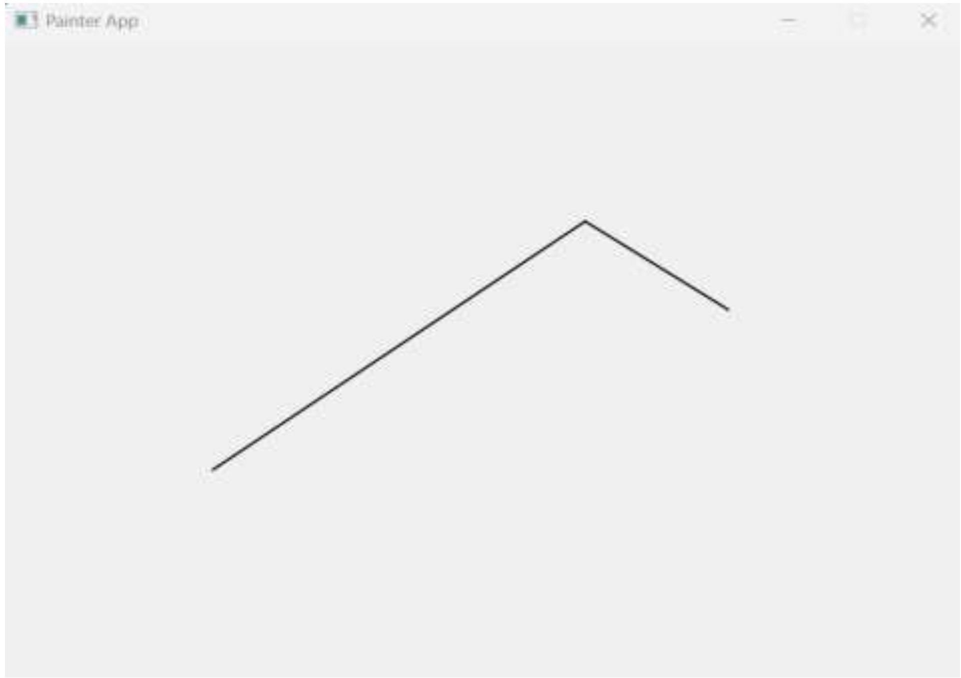
Σύνδεση Arduino με Servos



Painter App

Στο IDE της επιλογής σας τρέξτε το Painter_App.py . Θα ανοίξει ένα παράθυρο στο οποίο πατώντας σε διάφορα σημεία ζωγραφίζεται ευθύγραμμα τμήματα.

Πατώντας **‘Esc’** μπορείτε να σβήσετε το τρέχον σκίτσο σας. Πατώντας το κουμπί **‘s’** δίνετε εντολή στον βραχίονα να ζωγραφίσει το σκίτσο σας.



ROBOTALK 2025

Αυτόνομη Κίνηση Ρομποτικού Βραχίονα

Από τη Θεωρία στην Πράξη

~ Ευχαριστούμε για την προσοχή σας!