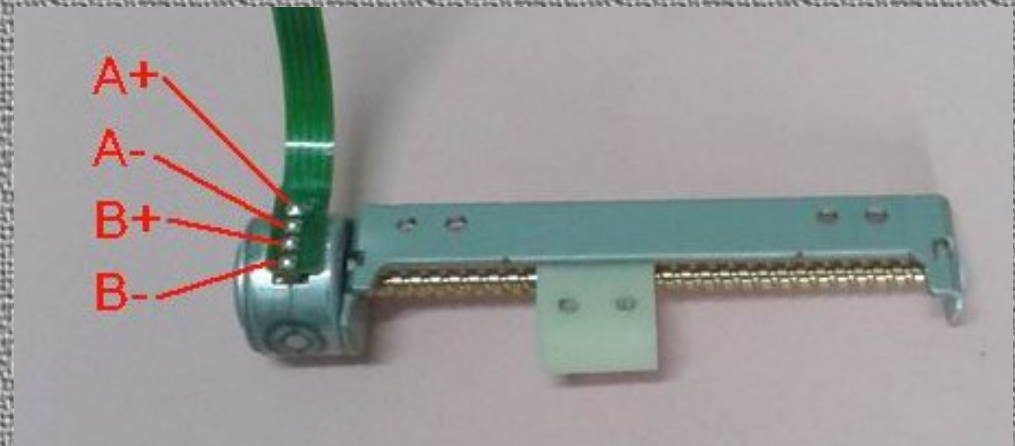
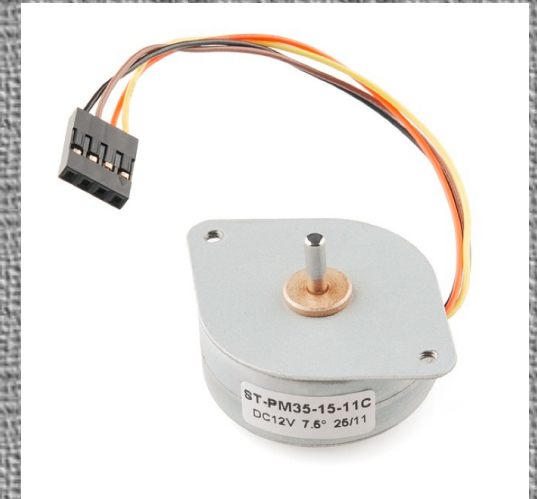


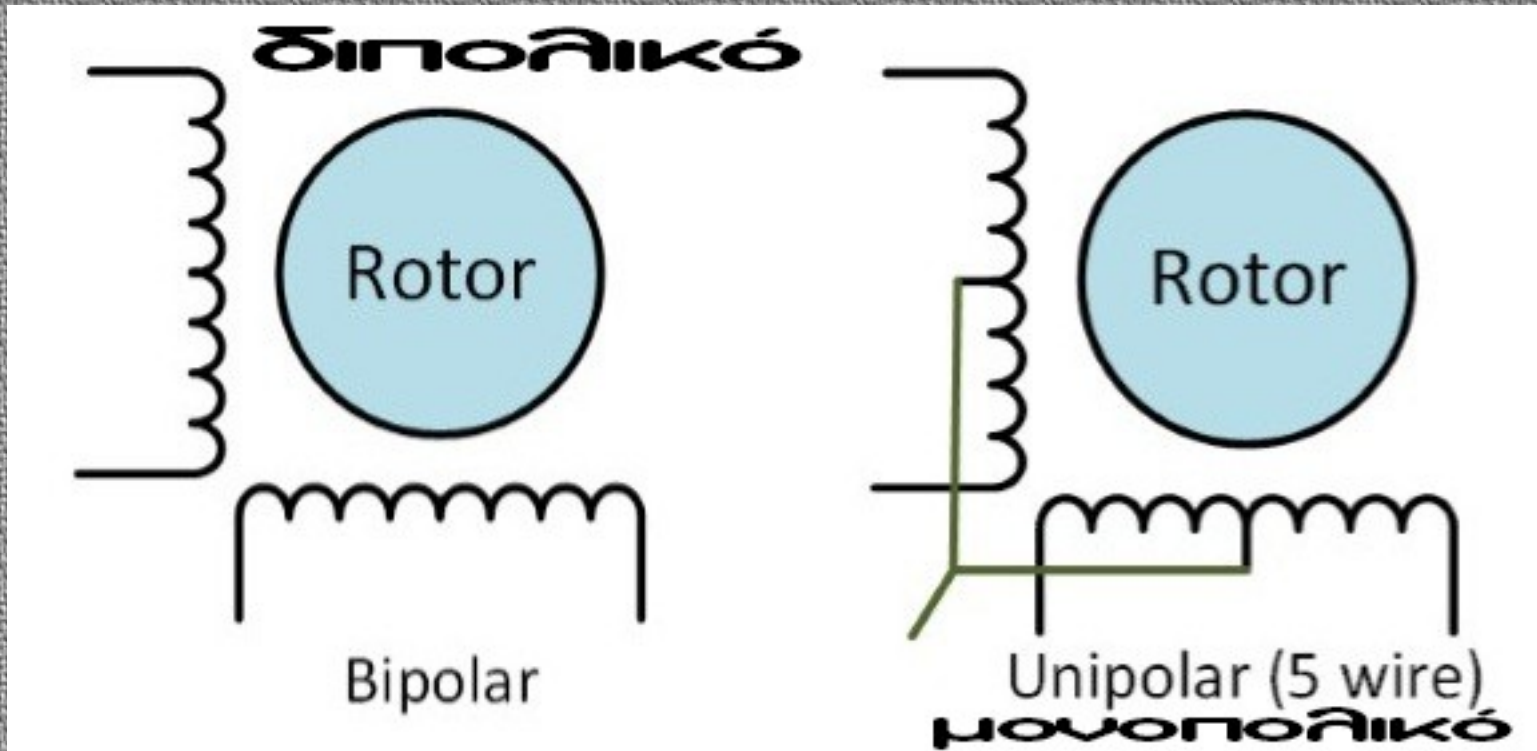
Βηματικοί κινητήρες

STEP ANGLE	STEPS PER REVOLUTION
0.72	500
1.8	200
2.0	180
2.5	144
5.0	72
7.5	48
15	24

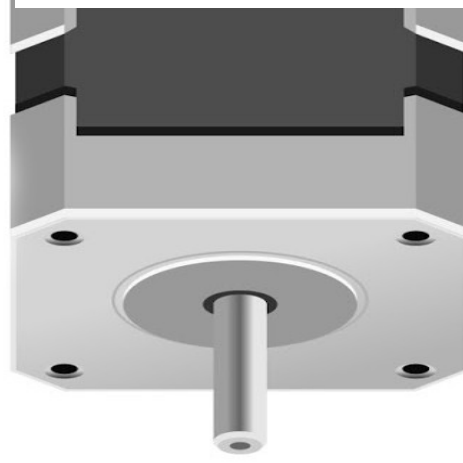
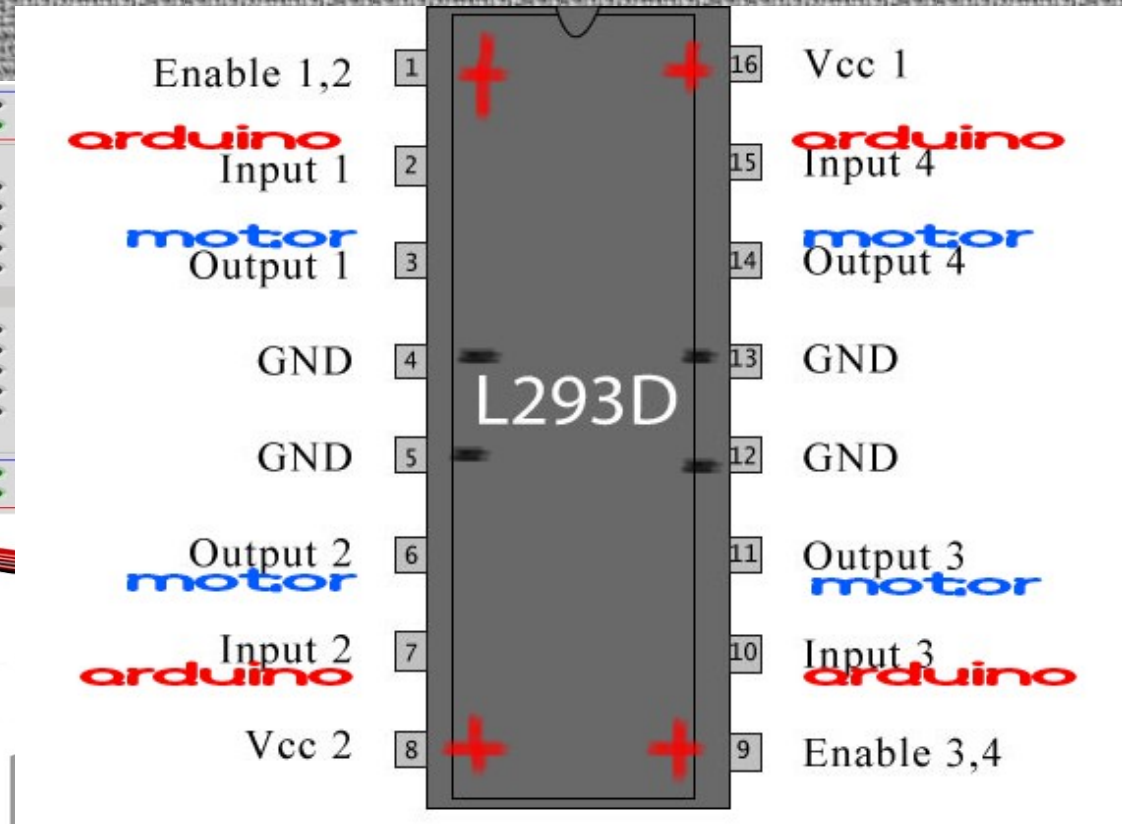
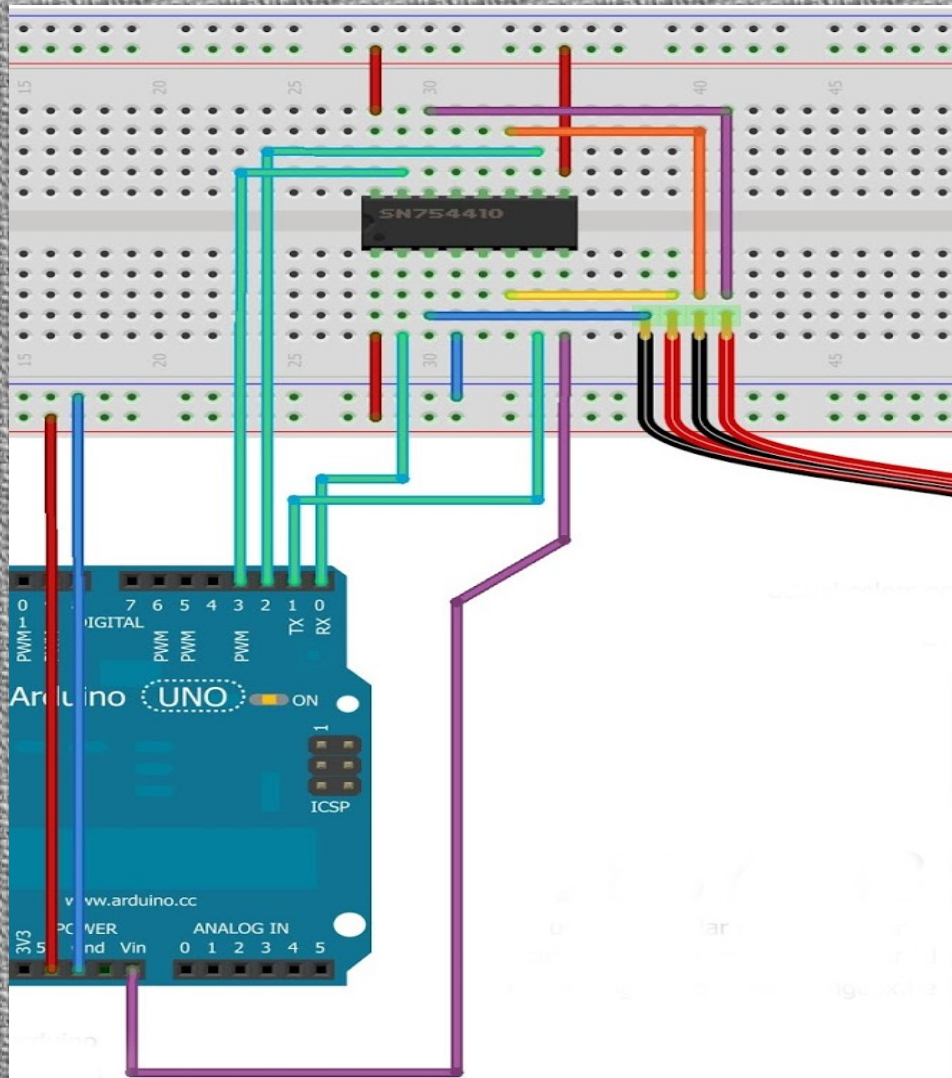
$$\text{Steps per second} = \frac{\text{rpm} \times \text{steps per revolution}}{60}$$



Μονοπολικά // διπολικά



- Η-bridge (ενισχυτές transistor arrays)
 Τρόπος σύνδεσης χωρίς driver



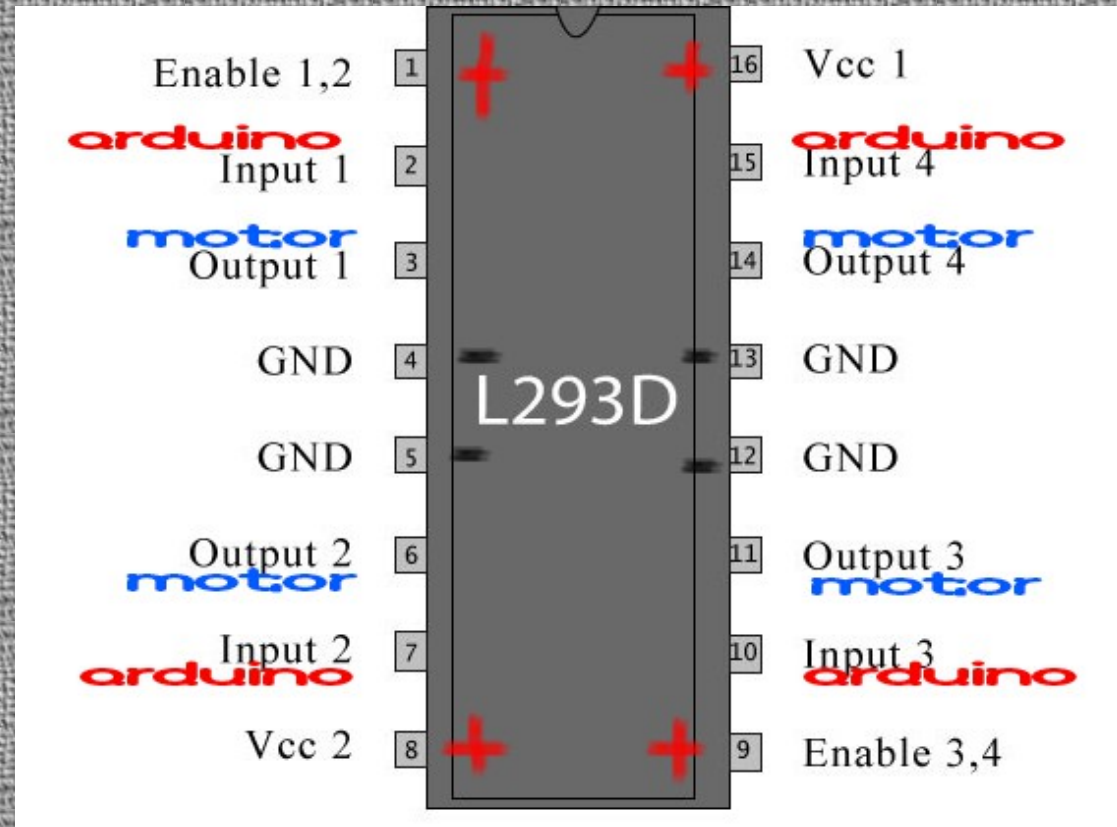
- H-bridge (ενισχυτές transistor arrays)
Κώδικας arduino – βιβλιοθήκη Stepper.h

```
#include <Stepper.h>
```

```
// (steps,pin1,pin2,pin3,pin4)  
Stepper myStepper(20,2,3,4,5);
```

```
void setup() {  
}
```

```
void loop() {  
  myStepper.setSpeed(100);  
  myStepper.step(200);  
}
```



G-code

- G-code είναι μια γλώσσα προγραμματισμού σε αριθμητικά συστήματα. Κυρίως χρησιμοποιείται στη ρομποτική και στα μηχανήματα κατασκευής με τη βοήθεια Η/Υ.
- Με την G-code ο άνθρωπος λέει στη μηχανή πως να κάνει κάτι. Το "πώς" δηλώνεται με εντολές για το που θα κινηθεί, πόσο γρήγορα και ακολουθώντας ποια διαδρομή.

```
G00 X23.4 Y12.8 Z7
```

```
G01 X23.4 Y12.8 Z7 F2400
```

```
G01 X123
```

```
G02 X90.6 Y13.8 I5 J10
```

Gcode Interpreter #1

```
// ένα απλό Gcode interpreter θα δίνω πχ G00 X12 F123;  
// και θα κινώ το μοτέρ 12 βήματα με ταχύτητα 123  
  
#include "Stepper.h"  
Stepper small_stepper(20, 2, 3, 4, 5);  
  
char line_buffer[200]; // Πίνακας buffer χαρακτήρας - χαρακτήρας οι εντολές που έρχονται  
char c;                // χαρακτήρες σειριακής επικοινωνίας  
int deiktis = 0;  
String inputString = "";  
  
//-----  
void setup() {  
  // αρχή σειριακής επικοινωνίας  
  Serial.begin(9600);  
  // το πολύ 200 bytes  
  inputString.reserve(200);  
}
```

Gcode Interpreter #2

```
void loop() {  
  while (Serial.available()) {  
  
    char inChar = (char)Serial.read(); // διάβασε τον νέο χαρακτήρα  
    inputString += inChar;           // πρόσθεσε τον νέο χαρακτήρα στο τέλος του String  
    deiktis = deiktis + 1; // ή deiktis++;  
  
    if ((inChar == '\n') || (inChar == ';')) { // Αν ο χαρακτήρας είναι ENTER ή ; τελείωσε η εντολή  
      inputString.toCharArray(line_buffer, deiktis); // μετατροπή του string σε πίνακα  
      processIncomingLine(line_buffer, deiktis);  
  
      inputString = ""; // μηδενίζω το string για νέα εντολή  
      deiktis = 0;      // μηδενίζω δείκτη  
      Serial.flush(); // καθαρίζω σειριακή επικοινωνία  
      // και φτου κι απ' την αρχή  
    }  
  }  
}
```

Gcode Interpreter #3

```
void processIncomingLine( char* line, int charNB ) {
    int i = 0;
    int bimata,taxitita;
    char buffer[ 64 ];

    while( i < charNB ) {
        //Serial.println(i);
        if (line[i] == 'G'){
            i = i+1; // i++;
            // αφήνω τους 2 επόμενους χαρακτήρες πχ το 01 μετά το G01
            char* indexX = strchr( line+i, 'X' ); // σε ποιά θέση είναι το "X"
            // strchr(xxx,'K') επιστρέφει σε ποιά θέση του String xxx βρίσκεται ο Χαρακτήρας "K"
            bimata = atoi( indexX + 1); // μετά το X ότι υπάρχει και μοιάζει με αριθμό μετετρεψε το σε ακέραιο
            char* indexF = strchr( line+i, 'F' ); // σε ποιά θέση είναι το "F"
            taxitita = atoi (indexF + 1);
            kinise_moter(bimata,taxitita); // kinise_moter(A,B); υποπρόγραμμα που κινεί το μοτέρ
            // A βήματα με B ταχύτητα
            // θα μπορούσαμε εδώ να κινήσουμε το μοτέρ χωρίς κλήση υποπρογράμματος
            // small_stepper.setSpeed(taxitita);
            // small_stepper.step(bimata);

            Serial.println(bimata);
            Serial.println(taxitita);
        }
        i++;
    }
}
```


Gcode Interpreter #4

```
void kinise_moter(int steps, int tax)
{
    small_stepper.setSpeed(tax);
    //ταχύτητα του μοτέρ
    small_stepper.step(steps);
}
```

μπορούμε να δοκιμάσουμε τις εντολές με τη
χρήση σειριακής οθόνης

Processing