

Πρώιμος Μηχανοτρονικός Σχεδιασμός Εργαλειομηχανών Μικροκατεργασιών Στερεολιθογραφίας

Σπαθούλας Δημήτρης AM: 92206

25 Ιανουαρίου 2023



Περίληψη

Η ιδέα για την εργασία ήρθε καθώς ήθελα να κάνω κάτι διαφορετικό πάνω στο μάθημα, αλλά και σχετικό με τα γενικά μου ενδιαφέροντα, στα οποία συμπεριλαμβάνεται και το συγκεκριμένο μάθημα. Το κύριο ενδιαφέρον και βαρύτητα της εργασίας βρίσκεται στην υλοποίησή του κώδικα και στις ποικίλες χρήσεις που θα μπορούσε να έχει.

1 Εισαγωγή

Οι εργαλειομηχανές ορίζονται ως μια πλήρη ηλεκτρομηχανολογική διάταξη - άρα και μηχανοτρονική - η οποία χρησιμοποιείται για την επεξεργασία και δημιουργία αντικειμένων στον τρισδιάστατο χώρο[1]. Παρακάτω εξετάζεται και υλοποιείται ένας μηχανοτρονικός σχεδιασμός μιας εργαλειομηχανής CNC λειτουργίας για μικροκατεργασίες μέσω του ARDUINO μέσω μιας διάταξης μικροηλεκτρονικών συστημάτων. Η δημιουργία του τρισδιάστατου μοντέλου της μηχανής χρήζει μια ολόκληρη και διαφορετική έρευνα και προσέγγιση. Παρακάτω θα γίνει εμφανές ότι ο υλοποιημένος κώδικας μπορεί με μικρές παραλλαγές να χρησιμοποιηθεί και για διαφορετικές μηχανές και κατεργασίες όπως πλάνισμα, ηλεκτροσυγκόλληση, μηχανές 3d printing αλλά και σε φρέζες, δράπανα, φρεζοδράπανα, όπως και πολλές άλλες. Το μοντέλο όπως είναι ανεπτυγμένο μπορεί επιπρόσθετα να γίνει και σε κινήσεις ρομποτικού βραχίονα.

2 Σχεδιασμός

2.1 Μικροκατεργασίες και παραδοχές του σχεδίου

Ο όρος μικροκατεργασία χρησιμοποιείται για να αποδώσει οποιαδήποτε διεργασία γίνεται πάνω σε ένα αντικείμενο σε κλίμακά μικρών, η οποία ξεκινάει στα $1 \cdot 10 \exp(-6)$ m και φτάνει στην κλίμακα των nano[1]. Άρα είναι προφανές ότι για να μπορέσει να ελεγχθεί η εργασία θα πρέπει να οριστεί μία κλίμακα η οποία είναι πολλαπλάσια της κατεργασίας. Στο πρόγραμμα υπάρχει επιτάχυνση με την οποία κινούνται οι βηματικοί κινητήρες για ευκολία στην εξέταση, ωστόσο στην πραγματική υλοποίηση θα έπρεπε αν αποφευχθεί λόγω ακρίβειας (σε σημειακές κατεργασίες). Πλέον μέσω της ασαφής άλγεβρας και τεχνητής νοημοσύνης αυτό το πρόβλημα μπορεί να εξαλειφθεί πλήρως. Μία τέτοια προσέγγιση έχει γίνει στο μάθημα “ ευφυή συστήματα ” με την χρήση των SIMULINK, FUZZY LOGIC για έναν κινητήρα συνεχούς ρεύματος και τάσης αναφοράς 12 V .

Για να υπάρχει αποδοτικότερη διαχείρισή δεδομένων ο κώδικας έχει ως βασική προϋπόθεσή ότι το ελάχιστο μήκος διάνυσης βήματος στους κινητήρες είναι η διαφορά ανάμεσα στις διακριτές θέσης ενός πίνακα με αποδέκτες τιμές το 0 και 1 (βλ. κεφάλαιο 6). Αυτό βέβαια μειώνει δραστικά την διακριτότητα και ανάλυση του αλγορίθμου, για αυτό και προτείνεται παρακάτω περαιτέρω μελέτη. Το βήμα αυτό θεωρήθηκε μια πλήρη περιστροφή (η διάνυση 6.28 ακτινίων στο μοναδιαίο διάνυσμα κάθετο της επιφάνειας της διατομής του άξονα) των βηματικών κινητήρων που χρησιμοποιήθηκαν. Τέλος ως στόμιο ή χοάνη έχει θεωρηθεί ένας σερβομηχανισμός ο οποίος μετακινείται ανάλογα ως κλειστός ή ανοικτός (βλ. κεφάλαιο 6). Για πιο μεγάλη ευκολία και κατανόηση έχουν χρησιμοποιηθεί LED χρωμάτων που αντιστοιχούν σε συγκεκριμένες καταστάσεις λειτουργίας, αλλά και εμφάνιση δεδομένων παράλληλα με την λειτουργία του προγράμματος.

2.2 Αλγοριθμική Πολυπλοκότητα και ευκολία εξέτασης

Λόγο σαφών περιορισμών έχει θεωρηθεί βέλτιστο το παρακάτω μοντέλο πίνακα ορισμού διαστάσεων μέσω ανάδρασης με τις θέσεις του να γεμίζουν με τιμές μέσω συνάρτηση τυχαίας γεννήτριας (βλ. κεφάλαιο 6).

3 Ανάλυση Χρησιμοποιούμενου Υλισμικού

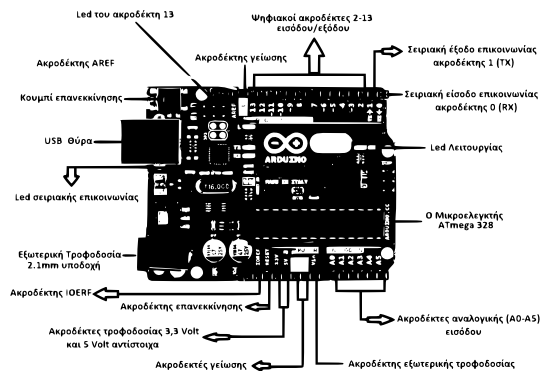
3.1 ARDUINO BOARD

Παρακάτω αναλύεται σε φωτογραφία η διάταξη της πλακέτας που χρησιμοποιήθηκε για την εργασία[2].

3.2 Μικροηλεκτρονικά Αναλώσιμα

Για τον σχεδιασμό χρησιμοποιήθηκαν :

- Δύο βηματικοί κινητήρες[3]



Σχήμα 1: ARDUINO UNO SPECS

- Ένας σερβοκινητήρας[4]
- Ένας πιεστικός διακόπτης[5]
- Τρεις δίοδοι φωτοεκπομπής[6]
- Μια πειραματική πλακέτα[6]
- Καλώδια M/M και F/F [6]
- Αντιστάσεις 330 Ω [6]

*ΣΗΜΕΙΩΣΗ: ΣΤΙΣ ΠΑΡΑΠΑΝΩ ΣΕΛΙΔΕΣ ΒΡΙΣΚΕΤΑΙ Η ΜΕΜΟΝΟΜΕΝΗ ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ ΚΑΘΕ ΣΤΟΙΧΕΙΟΥ ΜΕ ΚΑΠΟΙΑ ΒΑΣΙΚΑ ΣΧΕΔΙΑ.

4 Μαθηματική Ανάλυση - Ανυσματικός Λογισμός - Δυναμική Μηχανισμών

Το παραπάνω μοντέλο λόγω προ-αναφερθέντων αιτιών βασίζεται περισσότερο στην ηλεκτρονική παρά στην μηχανολογική του προσέγγιση, καθώς υπάρχει περιορισμός πόρων. Η αντίστροφη κινηματική εμφανίζεται σε πάρα πολλές πτυχές της λειτουργίας. Για αυτό και έχουν γίνει σαφής παραδοχές για την σχετική κίνηση των στοιχείων. Η στατική και η δυναμική καταπόνηση επάνω στα μηχανολογικά στοιχεία, αλλά και οι δείκτες αστοχίας τους, έχουν θεωρηθεί μη περιοριστικά στην παρούσα εργασία.

5 Σύνθεση Στοιχείων Στην πλακέτα

Παρακάτω γίνεται λεπτομερής αναφορά στην σύνδεση των μικροηλεκτρονικών στοιχείων στο ARDUINO UNO .

Η πειραματική πλακέτα είναι ο κοινός κόμβος όλων. Έχει γίνει σύνδεση της τάσης 5 V στην στήλη με την κόκκινη χρωματική ένδειξη και το σύμβολο '+', ταυτόχρονα η γείωση της πλακέτας συνδέθηκε με την στήλη υπό το σύμβολο '-'. Ο πρώτος βηματικός κινητήρας συνδέεται στις εισόδους 8-9-10-11, ο δέκτης τάσης συνδέθηκε στην πειραματική πλακέτα παράλληλα της στήλης '+' όπως και η γείωση στην στήλη '-'. Αντίστοιχα συνδέθηκε και ο δεύτερος με εισόδους στο ARDUINO UNO τις 2-3-5-6.

Ο σερβοκινητήρας συνδέθηκε στην αναλογική είσοδο A5 (με αντίστοιχη των παραπάνω σύνδεση τάσης-γείωσης). Κάθε LED συνδέθηκε απομονωμένα σε αντίστοιχες αναλογικές θέσεις A0,A1,A2 με μια αντίσταση των 330 Ω σε σειρά με το καθένα. Ο διακόπτης συνδέθηκε με την ψηφιακή είσοδο 7 καθώς και με μία αντίσταση των 10 KΩ. Οι συνδέσεις με την τάση και την γείωση ήταν αντίστοιχες.

6 Ανάλυση Αλγορίθμου

Παρακάτω παρουσιάζεται ο κώδικας.

Πρώτα γίνεται δήλωση των βιβλιοθηκών που χρησιμοποιήθηκαν και των παραμέτρων.

```
// libraries
#include <Servo.h>
#include <AccelStepper.h>
#include <time.h>
#include <Stdlib.h>
#include <stdio.h>

// steppers
AccelStepper step2( 4 ,8,10,9,11);
AccelStepper step1( 4 ,2,5,3,6);
float maxs=512.;
float spd=256.;
float ac=64.;
float s1=0,s2=0;

// servo
int srv=A4; //pin analog
double pos=0.0; // deg 0.-180.
Servo servo; //call

// leds
int R=A0; // dig
int G=A2; // dig
int B=A1; // dig

//button
int b; // 0 || 1
int br=7; // pin

// variables
int i,j;
int n,m;
int dt=1000; // 1 sec
int *a;
```

Έπειτα δηλώνεται η σχέση ανάμεσα σε κάθε είσοδο και την λειτουργία της. Αυτό γίνεται και πάνω, αλλά η πλακέτα δέχεται αυτές τις εντολές στο πεδίο void setup() .

Παρακάτω βρίσκεται το κομμάτι εγκατάστασης των παραμέτρων σε συγκεκριμένες συναρτήσεις της γλώσσας ARDUINO .

```
void setup() {
  // put your setup code here, to run once:
  srand(time(0));
  Serial.begin(9600);

  // motor 1
  step1.setSpeed(spd);
  step1.setAcceleration(ac);
  step1.setMaxSpeed(maxs);
  // motor 2
  step2.setSpeed(spd);
  step2.setAcceleration(ac);
```

```

    step2.setMaxSpeed(maxs);

    // servo
    servo.attach(srv);

    // leds
    pinMode(R,OUTPUT);
    pinMode(G,OUTPUT);
    pinMode(B,OUTPUT);

    // button
    pinMode(br,INPUT);

    analogWrite(R,0);
    analogWrite(G,0);
    analogWrite(B,0);

```

Αρχικά δημιουργείται μία μήτρα τυχαίων τιμών σε κάθε καινούργια έναρξη λειτουργίας, έπειτα καλείται το πεδίο εμφάνισης αποτελεσμάτων. Για κάθε βηματικό κινητήρα σαφοποιείται η ταχύτητα του, η μέγιστη του ταχύτητα και η μέγιστη επιτάχυνση που μπορεί να πάρει. Ο σερβοκινητήρας ενεργοποιείται, οι ενεργοποιητές φωτός συνδέονται ως αναλογικοί έξοδοι. Το κουμπί συνδέεται ως ψηφιακή είσοδο. Τέλος τα LED δέχονται ψηφιακή τιμή μηδέν, δηλαδή είναι σβηστά.

```

    Serial.println("Enter row of matrix (<=5) ");
    while (Serial.available()==0) {

    }

    m=Serial.parseInt();
    Serial.print("rows : ");
    Serial.println(m);
    Serial.println();
    Serial.println("Enter collum number (<=5) ");
    while (Serial.available()==0) {

    }

    n=Serial.parseInt();
    Serial.print("collums : ");
    Serial.println(n);
    Serial.println();
    if ( (n<=0) || (m<=0) ) {
        Serial.println("Size is negative or zero. Please restart program.");
        exit(1);
    }
    int *A=malloc((m*n)*sizeof(int));

    for (i<0;i<m;i++) {
        Serial.print("|");
        for (j=0;j<n;j++) {
            A[(i*n)+j]=rand()%2;
            Serial.print(A[(i*n)+j]);
            Serial.print("|");
        }
        Serial.println();
    }

    Serial.println();

```

```

Serial.println();
Serial.println("Enter any character to start proccess");
while (Serial.available()==0) {

}
Serial.println("Initializing...");
mil();
EXEC(A);
EXIT(i);
}

```

Παραπάνω γίνεται είσοδος των γραμμών και των στηλών των διακριτών σημείων που ορίζουν ένα συγκεκριμένο σημείο στο επίπεδο λειτουργίας της μηχανής. Αυτές οι θέσεις δέχονται μέσω συνάρτησης τυχαίων αριθμών στο κλειστό 0,1 τιμές. Ο πίνακας που δημιουργείται είναι ένας δυναμικός πίνακας δύο διαστάσεων ο οποίος, καλείται ως μονοδιάστατος στον κώδικα. Τέλος καλούνται οι συναρτήσεις EXEC(A) EXIT(i) τύπου void .

```

void EXEC(int a[]) {

i=0;
j=0;
s1=step1.currentPosition();
s2=step2.currentPosition();
analogWrite(R,255);
delay(dt*3);
step2.moveTo((n-1)*2048.);
step2.runToPosition();
step2.moveTo(s2*2048.);
step2.runToPosition();

analogWrite(R,0);
analogWrite(G,255);
analogWrite(B,0);
Serial.println("execution starts now!");
Serial.println();
b=digitalRead(br);
while ( ( i < m ) && (b==1) ) {

    step1.moveTo(i*2048.);
    step1.runToPosition();
    Serial.print("pos motor_1 :");
    Serial.println(step1.currentPosition());
    Serial.println();
    b=digitalRead(br);
    if (i%2==0) { j=0;
        Serial.print("row is "); Serial.println(i); b=digitalRead(br);
        while ( ( j<n ) && (b==1) ) {
            if ( a[(i*n)+j]==1 ) { Serial.print("colum is "); Serial.println(j);
                b=digitalRead(br);
                step2.moveTo(j*2048.);
                step2.runToPosition();
                Serial.print("pos of motor 2 : ");
                Serial.println(step2.currentPosition());
                b=digitalRead(br);
                ser();
                b=digitalRead(br);
            }
            j++;
        }
    }
}
}

```

```

        step2.moveTo((n-1)*2048.); step2.runToPosition();
    }

    if (i%2!=0) { j=n-1;
        Serial.print("row is :"); Serial.println(i);
        while ( (j>=0) && (b==1) ) {
            if ( a[(i*n)+j]==1 ) { Serial.print("column is :"); Serial.println(j);
                b=digitalRead(br);
                step2.moveTo(j*2048.);
                step2.runToPosition();
                Serial.print("pos of motor 2 : ");
                Serial.println(step2.currentPosition());
                b=digitalRead(br);
                ser();
                b=digitalRead(br);
            }
            j--;
        }
        step2.moveTo(s2*2048.); step2.runToPosition(); b=digitalRead(br);
    }

    i++; j=0;
}

}

```

Η συνάρτηση EXEC(A) είναι η “ καρδιά ” της εργασίας καθώς εκεί γίνεται όλη η κίνηση των ενεργοποιητών, καθώς και καλείται η συνάρτηση για τον σερβοκινητήρα. Οι αρχικές θέσεις των κινητήρων αποθηκεύονται, έπειτα το πρόγραμμα τίθεται σε μία λειτουργία επεξεργασίας των ορίων των διαστάσεων με τον κινητήρα που μετατοπίζεται κατά στήλες να φτάνει και να γυρνάει πίσω στην αρχική του θέση (το συγκεκριμένο βήμα μπορεί να αποφευχθεί).

Παρακάτω γίνεται έλεγχος για την κατάσταση του κουμπιού. Μέσα στην εντολή ΌΣΟ υπάρχει η συνθήκη του δείκτη γραμμών να είναι μικρότερος της τιμής που έχουμε δώσει μείον μία θέση, αφού θεωρείται ότι η πρώτη θέση στο επίπεδο λειτουργίας είναι και η αρχική. Ταυτόχρονα γίνεται έλεγχος με την ένωση της συνθήκης “ το κουμπί να είναι μη πατημένο ” (δηλαδή 1). Μέσα στην ΌΣΟ ο κινητήρας γραμμών κινείται κατά μια ολόκληρη περιστροφή(64*32) επί την μεταβαλλόμενη τιμή i . Εάν το πηλίκο της διαίρεσης του δείκτη γραμμών με το 2 είναι 0 τότε, ο δεύτερος βηματικός κινητήρας βρίσκεται στην αρχική θέση 0 ως προς τις στήλες. Εάν το πηλίκο δεν είναι 0 τότε, ο δεύτερος κινητήρας βρίσκεται στο ακρότερο πιθανό σημείο των διακριτών θέσεων, δηλαδή στην θέση στήλης n .

Εάν η τιμή της συγκεκριμένης θέσης του πίνακα σημείων αντιστοιχεί στην τιμή 1, τότε ο κινητήρας στήλης μετακινείται σε εκείνη την θέση και καλείται η συνάρτηση ser() . Αυτά τα βήματα επαναλαμβάνονται μέχρι η πρώτη εντολή ΟΣΟ να γίνει ψευδής.

```

void ser() {

    analogWrite(B,255);
    analogWrite(G,0);
    delay(dt/2);
    servo.write(pos+90.0);
    delay(dt*3);
    analogWrite(B,0);
    analogWrite(G,255);
    servo.write(pos);
    delay(dt);
}

```

Με την μετατόπιση του σερβοκινητήρα κατά ενενήντα μοίρες υποδηλώνεται η κατάσταση ανοιχτού στομίου,

η μπλε δίοδος ανάβει ενώ η πράσινη σβήνει για 3 δευτερόλεπτα, μέχρι να επανέλθει στην αρχική του θέση ο σερβοκινητήρας, δηλαδή το στόμιο να κλείσει.

```
void EXIT(I){
    analogWrite(R,255);
    analogWrite(G,0);
    analogWrite(B,0);
    step2.moveTo(s2);
    step2.runToPosition();
    step1.moveTo(s1);
    step1.runToPosition();

    if (I==m) {
        Serial.println("Procedure Executed Sucessfully");
        Serial.println("Please Exit Program");
    }
    if(I<m) {
        Serial.println("Program Interrupted..."); Serial.println("Please Exit Program");
    }
}
```

Τέλος, η συνάρτηση EXIT() ενεργοποιεί την κόκκινη δίοδο και κλείνει την πράσινη, επιδεικνύοντας λειτουργία επαναφοράς των βηματικών κινητήρων στις αρχικές τους θέσεις. Έπειτα, ανάλογα την τιμή του δείκτη i (εάν έχει κάνει προσπέλαση σε όλες τις τιμές του m-1) εμφανίζεται το αντίστοιχο μήνυμα επιτυχίας ή αποτυχίας της κατεργασίας.

Όλες οι εντολές αυτές υπάρχουν στο πεδίο void setup του arduino , άρα δεν υπάρχει επανάληψη. Θα πρέπει να γίνει επανεκκίνηση του προγράμματος για καινούργια είσοδο διάστασης πίνακα.

7 Επίλογος

Όπως αναφέρθηκε, είναι γνωστό ότι η επίδειξη της λειτουργίας του συγκεκριμένου project είναι δύσκολη, έως και αδύνατη εξ' αποστάσεως.

Δυστυχώς, δεν έχω πλέον άδεια για SOLIDWORKS , αλλά και παράλληλα, ο σχεδιασμός του πλήρη συνόλου μιας τέτοιας μηχανής απαιτεί πολύ παραπάνω χρόνο και γνώσεις που ακόμα δεν διακατέχονται στο σημείο που θα χρειαζόντουσαν.

Τέλος, να αναφέρω ότι η σκέψη και η πλήρης υλοποίηση του κώδικα αποτελούν ολοκληρωτικά δικό μου κόπο. Τουλάχιστον από όσο έχω ψάξει και ξέρω, δεν βρήκα τίποτα αντίστοιχο η πλησίον αυτού.

Σπαθούλας Δημήτρης 18/12/2022 ®

Αναφορές

- [1] Kalpakjian, S. and Schmid. S. (2014) Manufacturing Engineering Technology. 7th Edition, Pearson Publishing Company, Upper Saddle River, New Jersey.
- [2] <https://docs.arduino.cc/hardware/uno-rev3>
- [3] <https://docs.arduino.cc/learn/electronics/stepper-motors>
- [4] <https://docs.arduino.cc/learn/electronics/servo-motors>
- [5] <https://docs.arduino.cc/built-in-examples/digital/Button>
- [6] <https://www.arduino.cc/>