ΣΚΟΠΟΣ

θα ασχοληθούμε με τους κινητήρες servo (σερβοκινητήρες) και stepper (βηματικοί κινητήρες).

ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ

Οι κινητήρες Servo

Τι είναι

Οι κινητήρες αυτοί έχουν την δυνατότητα περιστροφής από 0-180 μοίρες. Χρησιμοποιούνται συνήθως για

- Να κάνουν έλεγχο της περιοχής για εμπόδια
- Να ελέγχουν βραχίονα
- Όπου χρειάζεται κίνηση πχ περιστροφή ηλιακών πάνελ

Το μόνο που έχουμε να κάνουμε εμείς είναι να ορίσουμε την θέση (γωνία) στην οποία θέλουμε να σταματήσει ο κινητήρας όπως θα δούμε παρακάτω

Τρόπος συνδεσμολογίας

Κάθε servo έχει τρεις επαφές για σύνδεση:

1. Γείωση (GND), μαύρο καλώδιο ή καφέ
2. Θετική (VCC), κόκκινο καλώδιο τυπικά από 4.8V-6.0V
3. Ελέγχου (PWM), κίτρινο ή πορτοκαλί καλώδιο, σήμα ελέγχου για τον καθορισμό θέσης (δηλαδή μπορεί να συνδεθεί στις θέσεις digital PWM(Power Motor) 3,5,6,9,10 και 11.

Πρακτική υλοποίηση

Θεωρητικά ο σερβοκινητήρας μπορεί να κινηθεί από 0 έως 179 μοίρες. Πρακτικά στους φτηνούς κινητήρες 0-20 μοίρες δεν μπορεί να κινηθεί και βουίζει οπότε καλό είναι να αποφεύγεται αυτό το διάστημα για να μην υπερθερμανθεί ο κινητήρας.

Μπορούμε να δούμε σε ποιο διάστημα μπορεί να κινηθεί ο κινητήρας δίνοντάς του μια εντολή μετακίνησης και κοιτάζοντας αν βουίζει προσπαθώντας να φτάσει εκεί. Αν βουίζει καλό είναι να αποφεύγεται αυτή η τιμή.

Προγραμματισμός

```
1  #include <Servo.h>  //Πρέπει να συμπεριληφθεί
2  Servo myservo; //Ορισμός μεταβλητής για το χειρισμό του σερβοκινητήρα
3
4  void setup() {
5       myservo.attach(9);   //Ο σερβοκινητήρας συνδεδεμένος στο pin 9
6  }
7
8  void loop() {
9      for ( int pos = 20; pos <= 179; pos += 1) {
10           //Οι τιμές που μπορεί να πάρει η εντολή write είναι από
   20-179
11         //Πηγαίνει τον σερβοκινητήρα τη θέση pos μοίρες
12         myservo.write( pos);
13         delay(30);            //Εισάγει καθυστέρηση 30ms
14     }
15 }
```

Η κίνηση του κινητήρα γίνεται με μεγάλη ταχύτητα (περίπου 150msec για 90 μοίρες) μπορούμε να την ελέγξουμε προσθέτοντας delay

Προσοχή:

- Ένας λόγος για να «κάψουμε» έναν κινητήρα είναι να του δώσουμε εντολή περιστροφής αλλά κάποιο εμπόδιο να μην το αφήσει να φτάσει στις μοίρες που θέλουμε με αποτέλεσμα μετά από κάποιο χρονικό διάστημα να ζεσταθεί και να καεί.
- Πρέπει να λάβουμε υπόψιν το χρονικό διάστημα που απαιτείται ώστε το servo να περιστραφεί στην επιθυμητή γωνία. Για το συγκεκριμένο servo χρειάζονται περίπου 150msec για 90 μοίρες. Οπότε σε κάθε περιστροφή πρέπει να εισάγουμε και μια ανάλογη καθυστέρηση delay.
- Στο παραπάνω παράδειγμα για να μετακινηθεί από τη θέση 20 στη θέση 180 χρειάζονται 160*30 msec => 4,8 δευτερόλεπτα

Βηματικοί κινητήρας (stepper motor)

Θεωρία

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

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

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. Ένα τυπικό νούμερο είναι 4 βήματα και 200 βήματα για μια πλήρεις περιστροφή.

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. Παίρνει ενδιαφέρον όταν λέμε μόνο για να πάει ένα βήμα και γίνεται ακριβώς 1,8 βαθμούς.

Χαρακτηριστικά κινητήρα 28BYJ-48

Ο κινητήρας αυτός σε μία πλήρη περιστροφή έχει 32 βήματα.

Με γρανάζια 64 στροφές του κινητήρα αντιστοιχούν σε 1 στροφή που βλέπει ο χρήστης οπότε τελικά 1 στροφή είναι 64*32=2048 βήματα.

Ο κινητήρας μπορεί να κάνει 1-500 στροφές το λεπτό ανάλογα με τη ρύθμιση της ταχύτητας. Αυτό αντιστοιχεί σε 60000 msec έως 120 msec η κάθε στροφή

Επομένως ο πιο γρήγορος χρόνος είναι περίπου 8 δευτερόλεπτα.

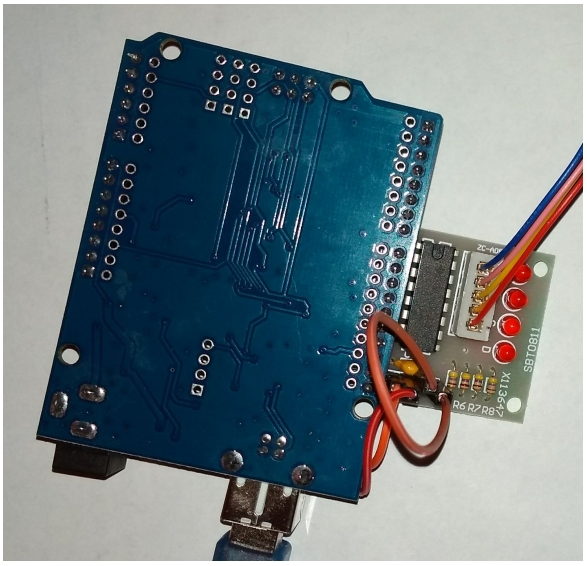
Αντίθετα αν θέλουμε να δει ο χρήστης μία στροφή κάθε 10 δευτερόλεπτα τότε αυτό αντιστοιχεί σε 6 στροφές/λεπτό για το χρήστη και 6*64=384 στροφές/λεπτό του εσωτερικού κινητήρα (επειδή τα γρανάζια κάνουν τη διαίρεση με το 64). Άρα θα πρέπει να βάλουμε ταχύτητα 384.

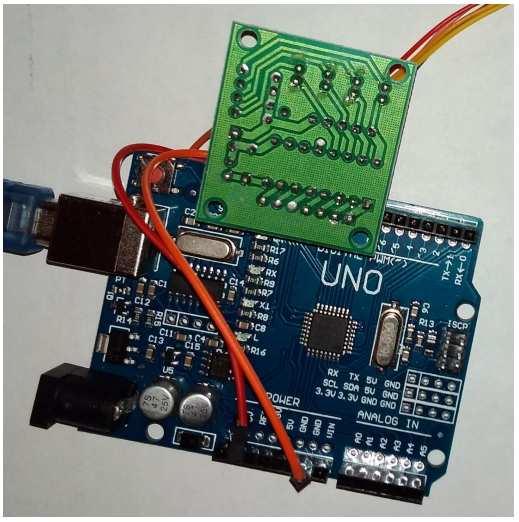
1ος τρόπος σύνδεσης. Με χρήση πλακέτας οδηγού

- Η μία πλευρά της πλακέτας έχει μια πρίζα 5 καλωδίων, όπου το καλώδιο από το βηματικό μοτέρ αγκιστρώνει. Το καλώδιο του κινητήρα πηγαίνει μόνο με έναν τρόπο
- 4 LED για να δείξει ποιο πηνίο τροφοδοτείται.
- η πλακέτα έχει είσοδο για ξεχωριστή τροφοδοσία του κινητήρα 5-12 V 1A, καθώς ο κινητήρας μπορεί να χρειαστεί περισσότερο ρεύμα από ότι μπορεί να δώσει ο μικροελεγκτής και ενδεχομένως να τον βλάψει. Μικροί βηματικοί κινητήρες μπορούν να τροφοδοτηθούν απευθείας από τα 5V του Arduino
- Τα pin on/off δίπλα στην τροφοδοσία πρέπει να είναι βραχυκυκλωμένα.

Arduino	Controller
pin8	IN1
pin9	IN2
pin10	IN3
pin11	IN4
GND	-(ΠΛΗΝ)
5V	+(ΣΥΝ)

Η συνδεσμολογία





Για να συνδεθεί η πλακέτα του ελεγκτή του step motor με την πλακέτα του Arduino χρειάζονται καλώδια που είναι θηλυκά στο ένα άκρο και αρσενικά στο άλλο. Τα IN1,IN2,IN3,IN4 κουμπώνουν απευθείας πάνω στις εξόδους 8-11 του Arduino. Το - και + του ελεγκτή πρέπει να πάνε το GND και στα 5V του Arduino. Επίσης τα δύο διπλανά ποδαράκια πρέπει να βραχυκυκλώσουν για αυτό αν δεν υπάρχει jumper θα χρειαστεί να βραχυκυκλώσουν με ένα καλώδιο που να είναι θηλυκό και στα δύο άκρα.

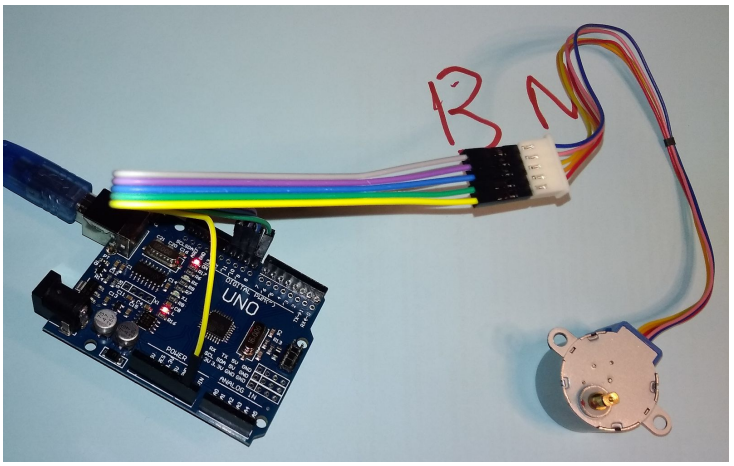
Το πρόγραμμα

```
1  #include <Stepper.h>
2
3  //Αριθμός βημάτων για μια ολόκληρη περιστροφή του κινητήρα
4  const int stepsPerRevolution=32;
5  const int gear=64;
6  //Η πρώτη παράμετρος είναι τα βήματα του κινητήρα
7  //Η 2η-5η δείχνουν σε ποιες εξόδους του Arduino είναι συνδεδεμένος
8  Stepper myStepper (stepsPerRevolution, 8 , 10 , 9, 11 );
9
10 void setup() {
11     Serial.begin( 9600);
12     //Ταχύτητα περιστροφής του κινητήρα σε rpm (περιστροφές το λεπτό)
13     myStepper.setSpeed ( 384);
14 }
15
16 void loop() {
17     long t = millis();    //Έναρξη χρόνου
18     //Κινεί το κινητήρα για μια πλήρη περιστροφή
19    myStepper.step( stepsPerRevolution*gear);
20    Serial.println( millis()-t);      //Τυπώνω το χρόνο που
   χρειάστηκε
21    delay(2000);
22 }
```

2ος τρόπος σύνδεσης. Χωρίς τη χρήση ελεγκτή

Η συνδεσμολογία

Από τα 5 καλώδια του κινητήρα το κόκκινο είναι το κοινό και το καθένα από τα 4 υπόλοιπα δίνει ρεύμα στο αντίστοιχο πηνίο. Οπότε συνδέουμε το κόκκινο στο GND του Arduino και τα 4 επόμενα σε 4 εξόδους π.χ. 8,9,10,11.



Το πρόγραμμα

Το παρακάτω πρόγραμμα προκαλεί μια δεξιόστροφη περιστροφή στον κινητήρα σε 2*4096 msec, περιμένει για 3 δευτερόλεπτα και συνεχίζει.

```
#define IN1  8
#define IN2  9
#define IN3  10
#define IN4  11
int Steps = 0;
boolean Direction = 1;

void setup() {
   Serial.begin(115200);
   pinMode(IN1, OUTPUT);
   pinMode(IN2, OUTPUT);
   pinMode(IN3, OUTPUT);
   pinMode(IN4, OUTPUT);
}

void loop() {
   Direction=1;
   for(int i=0; i < 4096; i++) {
      stepper(1);
    delayMicroseconds( 2000);
   }
   delay( 3000);
}

void stepper(int xw) {
   for (int x=0;x<xw;x++){
      switch(Steps){
        case 0:
           digitalWrite(IN1, LOW);
           digitalWrite(IN2, LOW);
           digitalWrite(IN3, LOW);
           digitalWrite(IN4, HIGH);
           break;
         case 1:
           digitalWrite(IN1, LOW);
           digitalWrite(IN2, LOW);
           digitalWrite(IN3, HIGH);
           digitalWrite(IN4, HIGH);
           break;
        case 2:
```

```
            digitalWrite(IN1, LOW);
            digitalWrite(IN2, LOW);
            digitalWrite(IN3, HIGH);
            digitalWrite(IN4, LOW);
            break;
         case 3:
            digitalWrite(IN1, LOW);
            digitalWrite(IN2, HIGH);
            digitalWrite(IN3, HIGH);
            digitalWrite(IN4, LOW);
            break;
         case 4:
            digitalWrite(IN1, LOW);
            digitalWrite(IN2, HIGH);
            digitalWrite(IN3, LOW);
            digitalWrite(IN4, LOW);
            break;
         case 5:
            digitalWrite(IN1, HIGH);
            digitalWrite(IN2, HIGH);
            digitalWrite(IN3, LOW);
            digitalWrite(IN4, LOW);
            break;
         case 6:
            digitalWrite(IN1, HIGH);
            digitalWrite(IN2, LOW);
            digitalWrite(IN3, LOW);
            digitalWrite(IN4, LOW);
            break;
         case 7:
            digitalWrite(IN1, HIGH);
            digitalWrite(IN2, LOW);
            digitalWrite(IN3, LOW);
            digitalWrite(IN4, HIGH);
            break;
         default:
            digitalWrite(IN1, LOW);
            digitalWrite(IN2, LOW);
            digitalWrite(IN3, LOW);
            digitalWrite(IN4, LOW);
            break;
      }
      SetDirection();
   }
}

void SetDirection(){
   if( Direction==1)
      Steps++;
   if( Direction==0)
      Steps--;
   if( Steps>7)
      Steps=0;
    if(Steps<0)
      Steps=7;
}
```

Αν θέλουμε η περιστροφή να γίνει π.χ. σε 15 δευτερόλεπτα τότε η delayMicroseconds θα πρέπει να πάρει σαν παράμετρο το 15*1000*1000/4096
Αν το Direction γίνει 0 αντί για 1 τότε ο κινητήρας θα κινηθεί αριστερόστροφα.

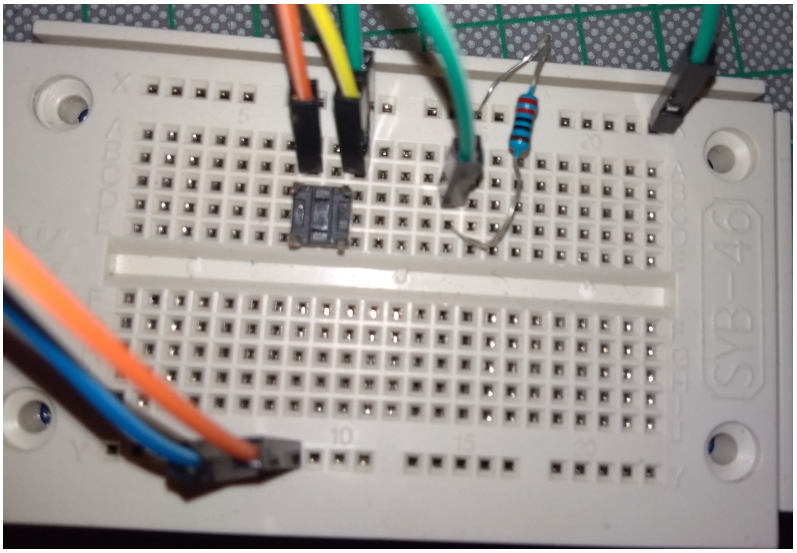
ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ

1η Δραστηριότητα: Περιστροφή σερβοκινητήρα

Να φτιαχτεί κύκλωμα με servo που να πηγαίνει από 20-180 σε 10 sec με βήμα 1 μοίρα και να επιστρέφει στο 20 σε άλλα 6 sec . Αυτό θα ξεκινά και θα σταματά με το πάτημα ενός button. Για τον χρόνο να υπολογίζεται μόνο τα delay και όχι τον χρόνο που χρειάζεται ο κινητήρας να μετακινηθεί από θέση σε θέση.

Παρατηρήσεις

1. Δεν χρειάζεται να εξετάζουμε πάντα την κατάσταση του button. Αρκεί να την εξετάζουμε όταν ο σερβοκινητήρας είναι στην κατάσταση ηρεμίας. Βέβαια αν μπορούσε όταν πατηθεί το κουμπί ο κινητήρας να σταματάει ότι κάνει και να ξεκινάει την παραπάνω κίνηση από τις 20^{o} θα ήταν ακόμη καλύτερα.
2. Για να βρούμε πως πρέπει να τοποθετηθεί πάνω στη breadboard το button αρκεί να το γυρίσουμε ανάποδα. Εκεί φαίνονται οι γραμμές που δείχνουν ποιοι ακροδέκτες είναι συνδεδεμένοι μεταξύ τους. Ουσιαστικά η γραμμή που έχει θα πρέπει να είναι κάθετη προς το breadboard. Χωρίς να το γυρίσουμε ανάποδα ο πιο απλός τρόπος είναι να κάνουμε συνδέσεις μόνο στους δύο διαγώνιους ακροδέκτες (όποιο ζευγάρι θέλουμε).



3. Για να ανέβει το βίντεο στο δίκτυο είναι πιο εύκολο να γίνει από το κινητό πατώντας την Αποστολή στο Drive.

2η Δραστηριότητα: Μεταβολή της ταχύτητας περιστροφής του σερβοκινητήρα με ποτενσιόμετρο

Να φτιαχτεί ένα κύκλωμα που θα περιλαμβάνει σερβοκινητήρα και ποτενσιόμετρο. Το ποτενσιόμετρο θα ελέγχει την ταχύτητα με την οποία θα κινείται ο σερβοκινητήρας. Η ελάχιστη ταχύτητα για να φτάσει από 20-180 μοίρες και το αντίστροφο να είναι 2 sec και η μέγιστη 8 sec. Το βήμα θα είναι μια μοίρα. Για απλοποίηση του κυκλώματος δεν χρειάζεται διακόπτης αλλά ο σερβοκινητήρας ας κινείται πάντα. Η αλλαγή στο ποτενσιόμετρο θέλουμε να επηρεάζει άμεσα την ταχύτητα περιστροφής.

Ο πιο απλός τρόπος είναι το ποτενσιόμετρο να ελέγχει το delay. Με αυτόν τον τρόπο αν μεταβληθεί το ποτενσιόμετρο ενώ κινείται ο κινητήρας θα μεταβληθεί άμεσα και η ταχύτητα του κινητήρα.

Το ποτενσιόμετρο θα σας δώσει τιμές από 0 έως 1023. Για να υπολογίσετε το delay θα χρειαστείτε απλά μαθηματικά. Αν δεν θέλετε να χρησιμοποιήσετε απλά μαθηματικά υπάρχει η συνάρτηση map η οποία μετατρέπει μια τιμή που ανήκει σε ένα διάστημα εισόδου σε μία άλλη τιμή που να ανήκει σε ένα άλλο διάστημα.

```
map(value, fromLow, fromHigh, toLow, toHigh)
```

Π.χ. αν η μεταβλητή x παίρνει τιμές από 0 έως 200 ενώ εμείς θέλουμε να την χρησιμοποιήσουμε σε μία άλλη συνάρτηση που παίρνει τιμές από 50 έως 90 τότε θα γράψουμε

```
y = map(x, 0, 200, 50, 90)
```

3η Δραστηριότητα: Κίνηση βηματικού κινητήρα

Να γίνει κύκλωμα με stepper motor (μπορείτε να χρησιμοποιήσετε ή να μην χρησιμοποιήσετε τον αντίστοιχο ελεγκτή) που την ταχύτητά του θα την ελέγχουμε με ποτενσιόμετρο Ελάχιστη ταχύτητα 1 στροφή κάθε 20 δευτερόλεπτα και μέγιστη ταχύτητα 1 στροφή κάθε 8 δευτερόλεπτα.

Για να είναι φανερή η περιστροφή του κινητήρα τοποθετήστε κάποιο αντικείμενο στο γρανάζι του.

Για να αλλάζει άμεσα η ταχύτητα του κινητήρα με την αλλαγή στο ποτενσιόμετρο ο πιο απλός τρόπος είναι στο loop να διαβάζει την τιμή του ποτενσιομέτρου και να κάνει ο κινητήρας μόνο ένα βήμα.

4η Δραστηριότητα: Δημιουργία ρουλέτας με βηματικό κινητήρα

Σκοπός

Πατώντας ένα button θέλουμε να επιλέγεται ένας τυχαίος αριθμός από το 1 έως το 360 και ο κινητήρας να κάνει 1 πλήρη περιστροφή και μετά να σταματάει στο τυχαίο νούμερο που επιλέχτηκε. Η περιστροφή κάθε φορά θα αρχίζει από το σημείο που σταμάτησε και πάντα δεξιόστροφα.

Κατασκευή

Για να φαίνεται προς το που δείχνει ο κινητήρας θα προσαρμόσετε ένα αντικείμενο στο γρανάζι του (π.χ ένα μικρό μανταλάκι) .

Με μία σελίδα Α4 θα φτιάξετε τη βάση που πάνω της θα είναι ζωγραφισμένος ένας κύκλος με ακτίνες ανά 60 μοίρες . Στις έξι πλευρές αναγράψτε τις εντολές προς τους παίκτες όπως στην εικόνα (για να φαίνεται η οδηγία του παιχνιδιού).