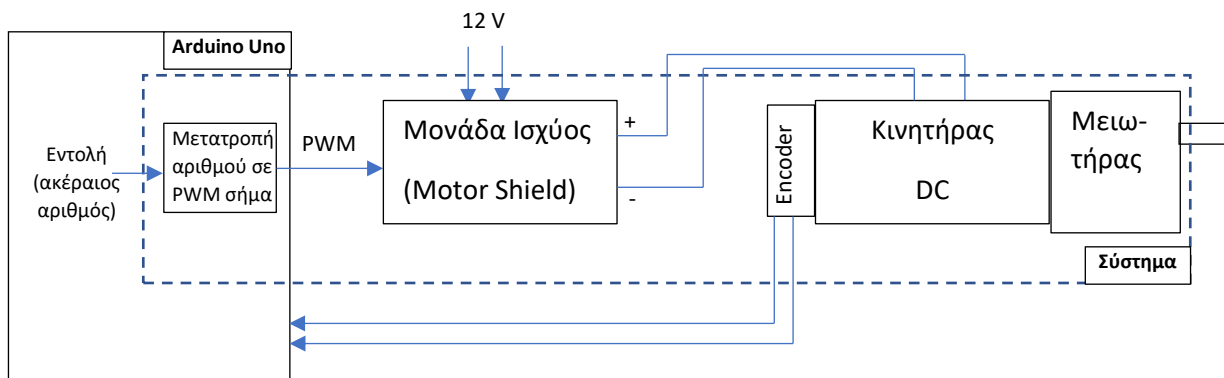


Εργαστηριακή Άσκηση 1

Ταυτοποίηση κινητήρα DC με ενσωματωμένο μειωτήρα

Το σύστημα ελέγχου ταχύτητας και θέσης του εργαστηρίου φαίνεται στο παρακάτω σχήμα. Η συνάρτηση μεταφοράς του συστήματος με είσοδο την εντολή για κίνηση (ακέραιος αριθμός) που δημιουργείται στον μικροελεγκτή και έξοδο την γωνιακή ταχύτητα του κινητήρα (μετρούμενη σε «παλμούς του encoder/δευτερόλεπτο»), είναι :

$$G(s) = \frac{K_{tot}}{Ts + 1} \left(\frac{\text{παλμοί/s}}{\text{"μονάδα εντολής"}} \right)$$



A. Σύνδεση αισθητήρα θέσης (encoder) με τον μικροελεγκτή και μέτρηση γωνίας στροφής.

Τα δυο κανάλια του αισθητήρα, συνδέονται στις ψηφιακές εισόδους 2 και 3 του μικροελεγκτή. Ο αισθητήρας τροφοδοτείται επίσης με την απαραίτητη τάση (5V DC), από τον ίδιο τον μικροελεγκτή. Για κάθε μια περιστροφή του άξονα του κινητήρα (στον οποίο είναι προσαρμοσμένος ο αισθητήρας) ο μικροελεγκτής καταγράφει τον αριθμό 64. Η «διακριτική ικανότητα» συνεπώς μέτρησης γωνίας του συγκεκριμένου αισθητήρα είναι ο αριθμός $64/360^\circ$.

Ο μειωτήρας έχει σχέση μετάδοσης $n=6.3$.

Για την μέτρηση της γωνίας, θα χρησιμοποιήσουμε την βιβλιοθήκη "Encoder.h"¹ που πρέπει να είναι εγκατεστημένη στο περιβάλλον προγραμματισμού.

Ανοίξτε ένα νέο αρχείο στο περιβάλλον προγραμματισμού του Arduino και γράψτε :

```
#include <Encoder.h>
```

```
Encoder Enc(2, 3); (Δηλώνει τις ψηφιακές εισόδους στις οποίες έχει συνδεθεί ο encoder)
```

```
long newP; (Ορίζει ένα «μεγάλο» ακέραιο ο οποίος θα έχει την γωνιακή θέση)
```

¹ Μπορεί κανείς να γράψει και μόνος του το σχετικό προγραμματάκι. Υπάρχουν επίσης πολλές άλλες βιβλιοθήκες ελεύθερα διαθέσιμες που κάνουν την ίδια δουλειά.

```

void setup() {
  Serial.begin(9600); } (Ρυθμίζει την σειριακή θύρα)
  Enc.write(0);      (... αρχική τιμή για τον encoder)
}
void loop() {       (Κάνε συνεχώς τα παρακάτω :)
  newP = Enc.read(); (Διάβαζε την γωνία στροφής ...)
  Serial.println (newP); (...και στείλε την στην σειριακή θύρα)
  delay(10);        (Περίμενε και 10 ms να ξεκουρασθείς)
}

```

Τρέξετε το πρόγραμμα. Ανοίξτε το παράθυρο που δείχνει τι έρχεται στην σειριακή θύρα και δείτε την γωνία να αλλάζει καθώς περιστρέφετε με το χέρι τον άξονα του μειωτήρα.

Μπορείτε να επιβεβαιώσετε ότι μετράτε σωστά;

B. Μέτρηση της γωνιακής ταχύτητας περιστροφής.

Θα μετρήσουμε την γωνιακή ταχύτητα, παραγωγίζοντας αριθμητικά την γωνιακή θέση. Η γωνιακή θέση, όπως και όλα τα μεγέθη που μετρούνται με την βοήθεια Η/Υ, μετριέται σε «διακριτές» χρονικές στιγμές, που τις χαρακτηρίζομαι t_i . Ισχύει :

$$\omega(t_i) = \frac{d\theta}{dt} = \lim_{\Delta t \rightarrow 0} \left(\frac{\Delta\theta}{\Delta t} \right) = \lim_{(t_i - t_{i-1}) \rightarrow 0} \left(\frac{\theta_{t_i} - \theta_{t_{i-1}}}{t_i - t_{i-1}} \right)$$

Όπου, t_{i-1} και t_i δύο διαδοχικές χρονικές στιγμές στις οποίες μετρά κανείς θ_{t_i} και $\theta_{t_{i-1}}$ τιμές γωνίας. Το όριο μας λέει ότι αν οι διαδοχικές χρονικές στιγμές απέχουν «λίγο» μεταξύ τους, τότε το πηλίκο $\frac{\theta_{t_i} - \theta_{t_{i-1}}}{t_i - t_{i-1}}$ είναι μια καλή προσέγγιση της γωνιακής ταχύτητάς την χρονική στιγμή t_i .

Για να υλοποιήσουμε τα παραπάνω με την βοήθεια του υπολογιστή, εισάγομε μια καινούργια μεταβλητή που θα έχει τις τιμές της γωνιακής ταχύτητας καθώς και μια ακόμη για να κρατά την τιμή της γωνιακής θέσης την χρονική στιγμή t_{i-1} . Γράφουμε στην αρχή του προγράμματος στην περιοχή ορισμού των μεταβλητών :

```
long vel;
```

```
long pos=0;
```

και τροποποιούμε το πρόγραμμα ως εξής :

```

void loop() {       (Κάνε συνεχώς τα παρακάτω :)
  newP = Enc.read(); (Διάβασε την γωνία στροφής ...)
  vel= (newP - pos)*100; (υπολόγισε την γωνιακή ταχύτητα σε παλμούς/sec ...)
  Serial.println (vel); (...και στείλε την στην σειριακή θύρα)
  Pos = newP;       (.....)
  delay(10);        (Περίμενε και 10 ms, δηλαδή κάνε περίπου 100 μετρήσεις το δευτερόλεπτο)
}

```

}

Γ. Οδήγηση (κίνηση) του κινητήρα με την βοήθεια του Motor Shield

Η μονάδα οδήγησης του κινητήρα (Motor Shield) χρειάζεται, για κάθε κινητήρα που θέλει να κινήσει (μπορεί να κινήσει δύο συγχρόνως), δύο «σήματα» από τον Arduino :

A. Ένα ψηφιακό σήμα που την πληροφορεί για την φορά περιστροφής. Στην περίπτωσή μας έρχεται από το ψηφιακό κανάλι 13 του μικροελεγκτή.

digitalWrite(13, HIGH); (Περιστροφή κατά την μία φορά ...)

digitalWrite(13, LOW); (... περιστροφή κατά την αντίθετη)

Ο μικροελεγκτής μπορεί να χειρισθεί τα ψηφιακά του κανάλια, είτε ως εισόδους είτε ως εξόδους. Πρέπει λοιπόν να τον πληροφορήσουμε νωρίς (βάζοντας την παρακάτω εντολή στην υπορουτίνα setup), ότι το κανάλι 13 είναι έξοδος :

pinMode(13, OUTPUT);

B. Ένα αναλογικό σήμα που την πληροφορεί πόσο γρήγορα πρέπει να κινήσει τον κινητήρα. Στην περίπτωσή μας, το σήμα αυτό έρχεται από το αναλογικό κανάλι 11 και μπορούμε να του δώσουμε τιμή από το πρόγραμμα με την εντολή :

analogWrite(11, «τιμή»); (Όπου τιμή = ακέραιος αριθμός από 0 έως 255)

Δ. Προσδιορισμός της συνάρτησης μεταφοράς

Για να προσδιορίσουμε τη ζητούμενη συνάρτηση μεταφοράς, θα δώσουμε στο σύστημα εντολή βαθμίδας και θα καταγράψουμε την απόκρισή του, την γωνιακή ταχύτητα δηλαδή. Γνωρίζουμε, ότι η απόκριση του συστήματός μας σε είσοδο βαθμίδας A, είναι :

$$\omega(t) = AK_{tot}(1 - e^{-\frac{t}{T}})$$

Όταν η γωνιακή ταχύτητα «σταθεροποιηθεί», η τιμή της είναι :

$$\omega_{ss} = \lim_{t \rightarrow \infty} \omega(t) = \lim_{t \rightarrow \infty} \left[AK_{tot}(1 - e^{-\frac{t}{T}}) \right] = AK_{tot} \left(1 - e^{-\frac{\infty}{T}} \right) = AK_{tot}(1 - 0) = \mathbf{AK_{tot}}$$

Θα βρούμε πειραματικά την τιμή αυτή, οπότε :

$$K_{tot} = \frac{\omega_{ss}}{A}$$

Από την καταγραφή της γωνιακής ταχύτητας, θα βρούμε επίσης τον χρόνο που απαιτείται ώστε η γωνιακή ταχύτητα να φτάσει το 63% της τελικής της τιμής. Ο χρόνος αυτός είναι η σταθερά χρόνου T του συστήματος.

Το παρακάτω πρόγραμμα ξεκινά και σταματά τον κινητήρα συνεχώς και στέλνει στην σειριακής θύρα τις μετρήσεις ταχύτητας.

