

Έλεγχος DC κινητήρα με την πλατφόρμα Arduino και το MATLAB® Support Package for Arduino® Hardware

Εισαγωγή

Η εργασία αυτή πραγματεύεται τον έλεγχο ενός DC κινητήρα με τη χρήση της πλατφόρμας μικροελεγκτή Arduino. Στόχος του συγκεκριμένου εγγράφου είναι η περιγραφή του hardware και του software που συναποτελούν το περιβάλλον ελέγχου του κινητήρα.

Το περιβάλλον ελέγχου του κινητήρα

Το περιβάλλον ελέγχου του κινητήρα αποτελείται από μία πλατφόρμα μικροελεγκτή Arduino Uno, ένα κύκλωμα τροφοδοσίας του κινητήρα, το οποίο είναι συγχρόνως αποδέκτης των καταστάσεων του κινητήρα (θέση και ταχύτητα) και αποστολέας του σήματος ελέγχου σε αυτόν, καθώς και ένα interface μεταξύ του Arduino Uno και του προαναφερθέντος κυκλώματος. Αξίζει να σημειωθεί ότι το interface τροφοδοτείται και αυτό από το κύκλωμα τροφοδοσίας του κινητήρα.

To Software

Ο προγραμματισμός του εκάστοτε ελεγκτή γίνεται σε φορητό υπολογιστή, ο οποίος επικοινωνεί με το Arduino Uno με ένα A to B καλώδιο USB, και μπορεί να πραγματοποιηθεί με δύο τρόπους :

- Χρήση του MATLAB® Support Package for Arduino® Hardware και προγραμματισμός του ελεγκτή στο MATLAB. Σε αυτή την περίπτωση το Arduino συλλέγει την πληροφορία των καταστάσεων του κινητήρα με τη βοήθεια του interface και την αποστέλλει στο MATLAB μέσω της σειριακής θύρας. Το πρόγραμμα υπολογίζει το επιθυμητό σήμα ελέγχου και το αποστέλλει στο Arduino με τον ίδιο τρόπο.
- Προγραμματισμός του ελεγκτή στο Arduino IDE και φόρτωση του προγράμματος στο Arduino. Σε αυτή την περίπτωση όλοι οι υπολογισμοί που απαιτούνται για τον προσδιορισμό του επιθυμητού σήματος ελέγχου πραγματοποιούνται από τον μικροεπεξεργαστή του Arduino Uno.

Στην περίπτωσή μας προτιμήθηκε η πρώτη μεθοδολογία για λόγους ευκολότερου προγραμματισμού, αλλά κυρίως γιατί παρέχει τη δυνατότητα συλλογής μετρήσεων κατά τη διάρκεια εκτέλεσης του προγράμματος.

Παρατηρήσεις για το MATLAB® Support Package for Arduino® Hardware

Αξίζει να αναφερθούμε σε ορισμένες παρατηρήσεις που κάναμε κατά τη χρήση του παραπάνω πακέτου της MATLAB.

1. Η έκδοση που βρίσκεται εδώ και υποστηρίζεται μόνο για εκδόσεις MATLAB από R2014a και πάνω είναι πιθανό να παρουσιάσει ορισμένα προβλήματα συμβατότητας. Προς το παρόν χρησιμοποιήστε την έκδοση που βρίσκεται εδώ.
2. Η χρήση του MATLAB ως εργαλείο προγραμματισμού για τον έλεγχο ενός συστήματος εισάγει μία χρονική καθυστέρηση στον κλειστό βρόχο (της τάξης των 40 ms), η οποία δεν είναι κρίσιμη για τις εφαρμογές που θα εξεταστούν στα πλαίσια του εργαστηρίου, αλλά είναι, ωστόσο άξια αναφοράς.

Οδηγίες για τη ρύθμιση της επικοινωνίας μεταξύ MATLAB και Arduino

Παρακάτω θα περιγράψουμε τον τρόπο με τον οποίο ο χρήστης μπορεί να ρυθμίσει την επικοινωνία μεταξύ MATLAB και Arduino. Τα βήματα που ακολουθούν απαιτείται να γίνουν μόνο 1 φορά και πριν από τη συγγραφή οποιουδήποτε κομματιού κώδικα για Arduino στο MATLAB.

1. Ακολουθούμε τη διαδικασία εγκατάστασης του Arduino IDE και των απαιτούμενων drivers ανάλογα με το λειτουργικό σύστημα του υπολογιστή μας ([Windows](#),[Linux](#),[Mac OS X](#)).
2. Κατεβάζουμε το .zip αρχείο που βρίσκεται εδώ και το κάνουμε unzip σε έναν φάκελο του υπολογιστή μας στον οποίο έχουμε δικαιώματα εγγραφής και ανάγνωσης.
3. Ανοίγουμε στο Arduino IDE το αρχείο adio.pde το οποίο βρίσκεται στην τοποθεσία ArduinoIO/pde/adio.
4. Επιλέγουμε τη θύρα (COM) στην οποία έχει αντιστοιχηθεί το Arduino (Εργαλεία/Σειριακή θύρα) και φορτώνουμε στο Arduino το adio.pde.
5. Ανοίγουμε το MATLAB με δικαιώματα διαχειριστή και εκτελούμε το πρόγραμμα `install_arduino.m`. Αυτό θα προσθέσει στο path του MATLAB την τοποθεσία των m files που απαιτούνται για την επικοινωνία με το Arduino.

Είμαστε πλέον έτοιμοι να γράψουμε κώδικα για Arduino σε γλώσσα MATLAB.

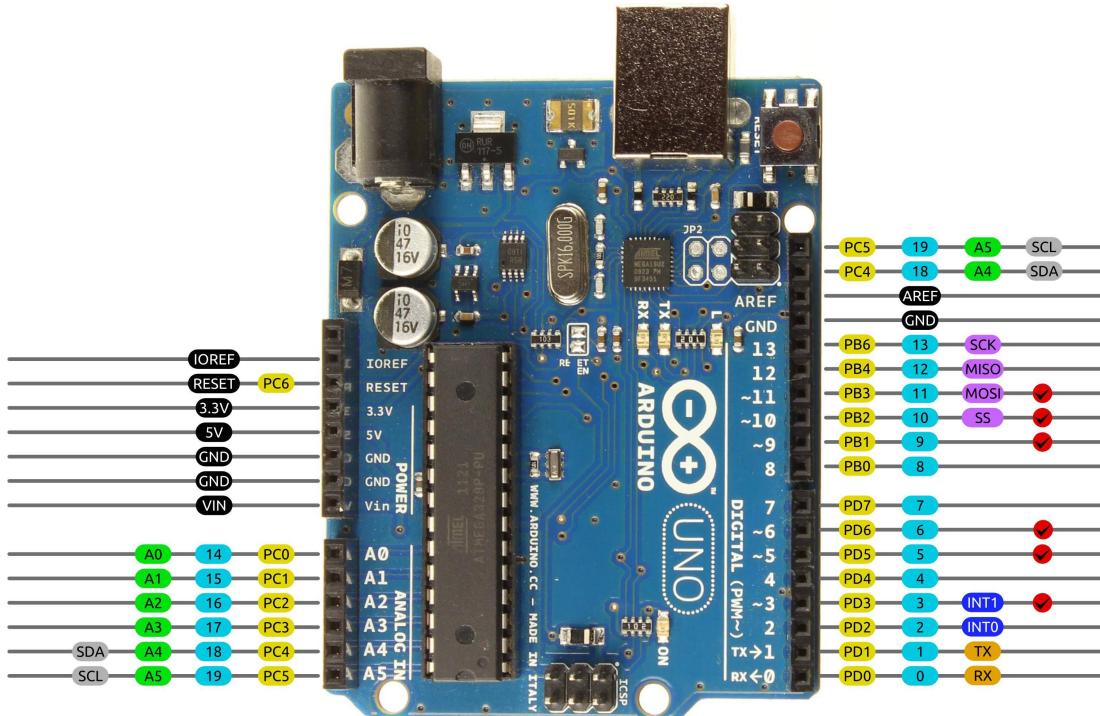
Βασικές εντολές

Ακολούθως θα παρουσιάσουμε τη σύνταξη ορισμένων βασικών εντολών που θα χρειαστούν κατά τη συγγραφή προγραμμάτων για Arduino σε MATLAB.

```
a=arduino('COMX');
```

Στην αρχή όλων των προγραμμάτων μας πρέπει να δημιουργήσουμε ένα αντικείμενο του τύπου arduino και να το αναθέσουμε σε μια μεταβλητή. (όπου X βάζουμε τον αριθμό της θύρας του υπολογιστή μας στην οποία έχει αντιστοιχηθεί το Arduino)

Arduino Uno R3 Pinout



AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT

```
analogWrite(a,X,Y)
```

Ορισμένα από τα digital pins του Arduino (σημειωμένα με το σύμβολο ~) μπορούν να παράγουν αναλογικές τάσεις από 0 έως 5 V με τη μορφή PWM. Η τιμή X

(ακέραιος) αντιστοιχεί στον αριθμό του pin που θέλουμε να χρησιμοποιήσουμε, ενώ η τιμή Y στην τάση που θέλουμε να παράγουμε. Το Y είναι επίσης ακέραιος και παίρνει τιμές από 0 έως 255. Η τιμή 0 αντιστοιχεί σε 0 V ενώ η τιμή 255 σε 5 V.

```
Y=analogRead(a,X);
```

Από τα analog pins του Arduino μπορούμε να μετρήσουμε αναλογικές τάσεις από 0 έως 5 V που αντιστοιχούν σε ακέραιους αριθμούς από 0 έως 1023. Η τιμή X (ακέραιος) αντιστοιχεί στον αριθμό του pin εισόδου, ενώ η τιμή Y αντιστοιχεί στην ακέραια τιμή που μετράμε.

To Hardware

Το κύκλωμα τροφοδοσίας δεν χρήζει κάποιας ιδιαίτερης περιγραφής, ενώ πληροφορίες για την πλατφόρμα Arduino Uno μπορούν να βρεθούν [εδώ](#).

To Interface μεταξύ του κυκλώματος τροφοδοσίας και του Arduino Uno

Η ύπαρξη ενός ενδιάμεσου κυκλώματος (interface), το οποίο παρεμβάλλεται ουσιαστικά μεταξύ του DC κινητήρα και του Arduino Uno είναι αναγκαία. Πιο συγκεκριμένα, ο κινητήρας τροφοδοτείται με τάση $\pm 24V$ και η τάση – θέσης (όπως αυτή προκύπτει από το ποτενσιόμετρο εξόδου) κυμαίνεται από 0 V έως περίπου +12.5 V (με κατάλληλη τροποποίηση της συνδεσμολογίας του ποτενσιόμετρου εξόδου), ενώ η τάση της ταχογεννήτριας κυμαίνεται περίπου μεταξύ -10 V και +10 V (η διατήρηση του προσήμου είναι απαραίτητη ως ένδειξη της φοράς περιστροφής του κινητήρα). Το Arduino Uno, ωστόσο, μπορεί να δεχθεί και να παραγάγει μόνο τάσεις που κυμαίνονται από 0 V έως +5 V. Συνεπώς, είναι απαραίτητη η κατάλληλη μετατροπή (διαίρεση και μετατόπιση) των τάσεων – καταστάσεων του κινητήρα, ώστε αυτές να είναι διαχειρίσιμες από το μικροελεγκτή και, στη συνέχεια, η ανύψωση και πιθανή αντιστροφή του σήματος ελέγχου του κινητήρα, το οποίο παράχθηκε από το μικροελεγκτή.

Επιμέρους τμήματα του interface

Ρυθμιστές τάσης από τα +24 V στα +15 V και +5 V και από τα -24 V στα -15 V

Προκειμένου να ελαχιστοποιηθεί η επιφάνεια που καταλαμβάνει το interface, επιλέξαμε να χρησιμοποιήσουμε ολοκληρωμένα κυκλώματα τα οποία

περιλαμβάνουν 4 τελεστικούς ενισχυτές το καθένα και συγκεκριμένα τα LM324n και LM348n. Η τροφοδοσία των παραπάνω ολοκληρωμένων κυκλωμάτων απαιτούνταν να γίνει στα (0 V) - (+15 V) και στα (-15 V) – (+15 V) αντίστοιχα. Επομένως, κρίθηκε αναγκαία η μετατροπή των διαθέσιμων τάσεων τροφοδοσίας ($\pm 24V$) στις απαιτούμενες με τη χρήση ρυθμιστών τάσης και συγκεκριμένα των 7815 (για τα +15 V) και 7915 (για τα -15 V). Η χρήση των +5 V που προκύπτουν από το ρυθμιστή τάσης 7805 θα περιγραφεί παρακάτω.

Διαχείριση της τάσης – Θέσης

Χωρίς καμία τροποποίηση το ποτενσιόμετρο εξόδου δίνει τάσεις μεταξύ περίπου -12.5 V και +12.5 V. Προκειμένου να αποφύγουμε την ανάγκη μετατόπισης της τάσης – θέσης και με μόνο μειονέκτημα την απώλεια αναλυτικότητας, συνδέουμε το ένα άκρο του ποτενσιόμετρου στη γείωση αντί των -24 V, οπότε παίρνουμε τάσεις μεταξύ 0 V και +12.5 V. Η τάση αυτή δεν μπορεί να τροφοδοτηθεί σε κάποιο analog input pin του Arduino, γι' αυτό και την υποτριπλασιάζουμε περνώντας την από ένα διαιρέτη τάσης 1/3. Επίσης, παρεμβάλλουμε έναν ακολουθητή τάσης (buffer) πριν την τροφοδοτήσουμε στο Arduino.

Διαχείριση της τάσης – ταχύτητας

Η ταχογεννήτρια δίνει τάσεις μεταξύ περίπου -10 V και +10 V. Η τάση αυτή δεν μπορεί να τροφοδοτηθεί σε κάποιο analog input pin του Arduino, και λόγω μεγέθους αλλά και λόγω προσήμου. Γι' αυτό αρχικά την υποδιπλασιάζουμε με ένα διαιρέτη τάσης 1/2 και στη συνέχεια, αφού την περάσουμε από έναν ακολουθητή τάσης, την τροφοδοτούμε σε έναν τελεστικό ενισχυτή ο οποίος είναι συνδεδεμένος ως αθροιστής με μοναδιαίο κέρδος προκειμένου να προσθέσουμε ένα offset +5 V (τα +5 V προκύπτουν από το ρυθμιστή τάσης που αναφέραμε παραπάνω). Τέλος, υποδιπλασιάζουμε ξανά την τάση χρησιμοποιώντας ένα διαιρέτη τάσης 1/2 και την τροφοδοτούμε σε ένα analog input pin του Arduino, αφού την περάσουμε ξανά από έναν ακολουθητή τάσης. Συνολικά η τάση της ταχογεννήτριας υφίσταται τις ακόλουθες μετατροπές : [-10,+10] → [-5,+5] → [0,+10] → [0,+5].

Παραγωγή του σήματος ελέγχου

Όπως αναφέραμε και προηγουμένως, το Arduino μπορεί να παραγάγει τάσεις μεταξύ 0 V και +5 V. Επομένως, είναι αδύνατο να κατασκευάσουμε ένα σήμα

ελέγχου που θα μπορεί να παίρνει και θετικές και αρνητικές τιμές με ένα μόνο τελεστικό ενισχυτή. Για το σκοπό αυτό έχουμε 2 τελεστικούς ενισχυτές που διπλασιάζουν την τάση εισόδου τους και είναι συνδεδεμένοι ο ένας σε αναστρέφουσα και ο άλλος σε μη αναστρέφουσα συνδεσμολογία. Στη συνέχεια, προσθέτουμε τις δύο τάσεις εξόδου των τελεστικών με τη βοήθεια ενός τελεστικού ενισχυτή σε συνδεσμολογία αθροιστή με μοναδιαίο κέρδος. Έτσι, αν επιθυμούμε να έχουμε αρνητικό σήμα ελέγχου (από -10 V έως 0 V) τροφοδοτούμε τον τελεστικό που είναι σε αναστρέφουσα συνδεσμολογία με μία τάση μεταξύ 0 V και +5 V, ενώ ο άλλος τελεστικός ενισχυτής τροφοδοτείται με μηδενική τάση. Το ανάποδο συμβαίνει όταν θέλουμε να παραγάγουμε ένα θετικό σήμα ελέγχου.

[Μετατροπή του PWM σε DC](#)

Το Arduino Uno δεν παράγει πραγματική (αναλογική) τάση, αλλά ορισμένα pins του χρησιμοποιούν [PWM](#) προκειμένου να προσομοιώσουν την επιθυμητή τιμή αναλογικής τάσης. Ωστόσο, ο ενισχυτής ισχύος που παρεμβάλλεται μεταξύ της εξόδου του interface και του κινητήρα απαιτεί ως είσοδο DC τάση. Για το λόγο αυτό παρεμβάλλουμε ένα απλό φίλτρο RC.

[Είσοδοι στην πλακετα του interface](#)

A) Από την πλευρά του κινητήρα

- +24 V, -24 V, GND από το τροφοδοτικό
- MOTOR VEL IN : η τάση που προέρχεται από την ταχογεννήτρια
- MOTOR POS IN : η τάση που προέρχεται από το ποτενσιόμετρο εξόδου.

B) Από την πλευρά του ARDUINO

- ARDUINO PIN9 : είσοδος του τελεστικού σε μη αναστρέφουσα συνδεσμολογία
- ARDUINO PIN10 : είσοδος του τελεστικού σε αναστρέφουσα συνδεσμολογία

[Έξοδοι από την πλακέτα του interface προς το Arduino](#)

- POS TO ARDUINO : η μετασχηματισμένη τάση που προέρχεται από το ποτενσιόμετρο εξόδου
- VEL TO ARDUINO : η μετασχηματισμένη τάση που προέρχεται από την ταχογεννήτρια

Έξοδοι από την πλακέτα του interface προς τον κινητήρα

- OUTPUT TO POWER AMPLIFIER : το σήμα ελέγχου που τροφοδοτείται στον τελεστικό ισχύος

Παραδείγματα εντολών

Συνδυάζοντας τα όσα αναφέρθηκαν σχετικά με τις εντολές που χρησιμοποιεί το MATLAB κατά την επικοινωνία με το Arduino και τον τρόπο με τον οποίο λειτουργεί το Interface παρουσιάζουμε παρακάτω ορισμένα παραδείγματα εντολών που μπορεί να φανούν χρήσιμα κατά την εκτέλεση των πειραμάτων.

Αρχικοποίηση της μεταβλητής a να αναφέρεται στο Arduino.

```
a=arduino('COM3');
```

Απελευθέρωση του Port COM3 από το σύστημα

```
delete(instrfind({'Port'},{'COM3'}))
```

Λήψη των καταστάσεων του συστήματος

```
velocity = analogRead(a,3);  
position = analogRead(a,5);  
x1 = 3 * Vref_Arduino * position/1023;  
x2 = 2 * (2 * velocity * Vref_Arduino / 1023 - V_7805);
```

Με τις παραπάνω εντολές μπορούμε να πάρουμε την αναλογική τιμή της ταχύτητας και της θέσης του κινητήρα. Σε όλα τα συστήματα η ταχύτητα μετριέται στο PIN3 και η θέση στο PIN5. Αξίζει να σημειωθεί η χρήση των μεταβλητών Vref_Arduino και V_7805 που είναι η τάση αναφοράς του DAC του Arduino και η τάση που παράγει ο ρυθμιστής τάσης 7805 αντίστοιχα. Οι τάσεις αυτές θεωρητικά είναι 5 V, αλλά στην πράξη μπορεί να παρουσιάζουν μια απόκλιση της τάξης των ±0.5 V (ειδικά η τάση αναφοράς του DAC όταν το Arduino τροφοδοτείται από μία θύρα USB), οπότε πρέπει να υπολογιστούν σε κάθε εκ νέου σύνδεση της θύρας USB με το Arduino. Ο υπολογισμός τους μπορεί να γίνει απλά μετρώντας με ένα

πολύμετρο τις πραγματική θέση – τάση του κινητήρα όταν αυτός είναι ακίνητος ($x_2 = 0$) και λύνοντας το σύστημα των x_1 και x_2 με αγνώστους τα $V_{ref_arduino}$ και V_{7805} .

Αποστολή του σήματος ελέγχου του κινητήρα

```
if u>0
    analogWrite(a,6,0)
    analogWrite(a,9,min(round(u / 2 * 255 / Vref_arduino) , 255))
else
    analogWrite(a,9,0)
    analogWrite(a,6,min(round(-u / 2 * 255 / Vref_arduino) , 255))
end
```

όπου u είναι η τάση του σήματος ελέγχου σε V, το PIN10 έχει αντιστοιχηθεί στο PIN9 και το PIN9 έχει αντιστοιχηθεί στο PIN6 (δείτε τη σημείωση για την αντιστοίχιση στο χαρτί που περιέχεται στο κουτί του συστήματος ελέγχου). Η χρήση του `min` δεν επιβάλλεται από λόγους ασφαλείας του συστήματος, αφού το πρόγραμμα θα βγάλει σφάλμα αν δοθεί τιμή μεγαλύτερη του 255 στην `analogWrite`, αλλά αποτελεί ένα είδος κορεσμού σε περίπτωση που η επιθυμητή τιμή του σήματος ελέγχου είναι πάνω από 10 V.