



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

Ενσωματωμένα Συστήματα

Ενότητα: ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ: CIRC-03

Δρ. Μηνάς Δασυγένης

mdasyg@ieee.org

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

<http://arch.ict.e.uowm.gr/mdasyg>

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα του Πανεπιστημίου Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Περιεχόμενα

1.Σκοπός της άσκησης.....	4
2.Παραδοτέα.....	4
3.Κατασκευή του κυκλώματος CIRC-03.....	5
4.Προγραμματισμός του κυκλώματος CIRC-03.....	6
5.Παραμετροποίηση του κυκλώματος CIRC-03.....	6
5.1Αλλαγή της συνάρτησης ελέγχου του μοτέρ (Έλεγχος της ταχύτητας).....	6

1. Σκοπός της άσκησης

- Χρήση του αναπτυξιακού περιβάλλοντος του Arduino¹ για τη συγγραφή και τη μεταφόρτωση προγραμμάτων στην πλακέτα του Arduino Uno.
- Δημιουργία κυκλώματος ελέγχου ενός μοτέρ.

2. Παραδοτέα

- Ένα αρχείο zip με τα project **CIRC_03**, **CIRC_03_c1**, **CIRC_03_c2** που θα δημιουργήσετε.
- Ένα screenshot από το arduino IDE που δείχνει ότι η μεταγλώττιση (compile) έγινε με επιτυχία και ταυτόχρονα εμφανίζει το μέγεθος του δυαδικού σχεδίου για κάθε κύκλωμα.
- Ένα video 5-10 sec επίδειξης του κυκλώματος μαζί με ηχητική περιγραφή για κάθε κύκλωμα.






Χρόνος ολοκλήρωσης εργαστηρίου: 30 λεπτά.

Τα pins του Arduino είναι κατάλληλα για να ελέγχουν μικρά ηλεκτρονικά στοιχεία όπως τα LEDs, όμως όταν έχουμε να κάνουμε με μεγαλύτερα στοιχεία, όπως ένα μοτέρ, απαιτείται η χρήση εξωτερικού τρανζίστορ. Τα τρανζίστορ είναι πολύ χρήσιμα, γιατί μας επιτρέπουν να ρυθμίζουμε μεγάλης έντασης ρεύματα, χρησιμοποιώντας ρεύματα μικρότερης έντασης. Το τρανζίστορ έχει τρία pins. Για ένα NPN τρανζίστορ όπως αυτό που θα χρησιμοποιήσουμε σ' αυτό το εργαστήριο, συνδέετε το φορτίο στο συλλέκτη, και τον εκπομπό στη γείωση. Όταν ένα μικρής έντασης ρεύμα περνάει από τη βάση στον εκπομπό, θα δημιουργηθεί ένα μεγαλύτερης έντασης ρεύμα και θα κινηθεί το μοτέρ. Το τρανζίστορ που θα χρησιμοποιηθεί είναι ένα P2N2222AG, το οποίο είναι ένα σύνηθες τρανζίστορ γενικής χρήσης. Τα σημαντικά στοιχεία του, είναι η μέγιστη τάση (40 V) και το μέγιστο ρεύμα (200 mA) τα οποία είναι αρκετά υψηλά για να θέσουν σε λειτουργία το μοτέρ μας. Όσον αφορά τη δίοδο 1N4001 μπορείτε να ανατρέξετε στο σχετικό λήμμα της [Wikipedia](#) για να δείτε για ποιό λόγο χρησιμοποιείται.

¹ Χρησιμοποιήθηκε υλικό από το [SparkFun Inventors Kit for Arduino](#)

3. Κατασκευή του κυκλώματος CIRC-03

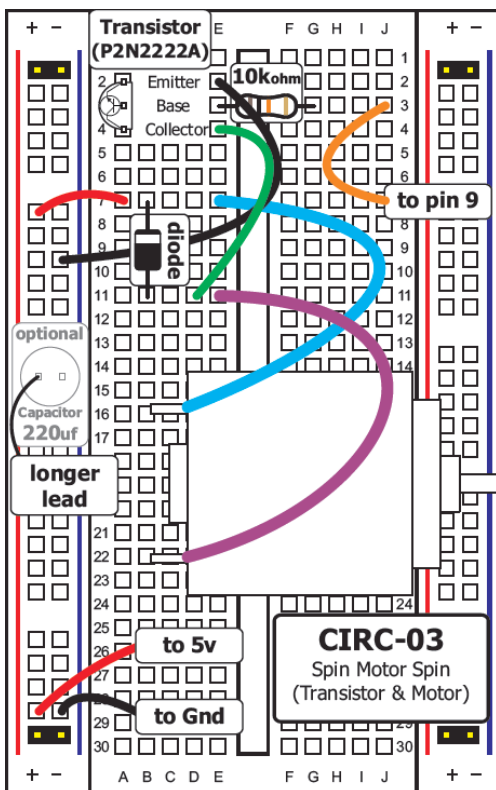
Για την εκπόνηση του κυκλώματος CIRC-03 απαιτούνται τα εξής μέρη:

Στοιχείο	Περιγραφή και ποσότητα
	1x Τρανζίστορ P2N2222AG (TO92)
	8x Καλώδια
	1x Μοτέρ
	1x Αντίσταση 10k Ohm (Καφέ-Μαύρο-Πορτοκαλί)
	1x Δίοδος (1N4001)

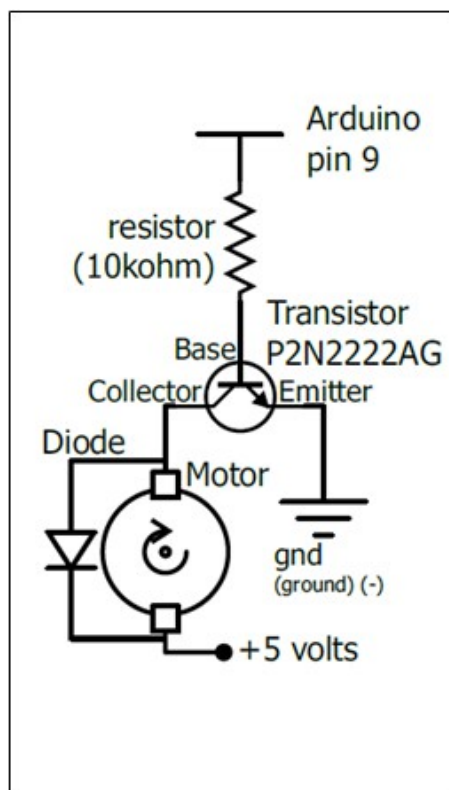
Τα φυλλάδια δεδομένων (datasheets) των υλικών βρίσκονται παρακάτω:

- [Τρανζίστορ P2N2222AG \(TO92 \)](#)
- [Αντίσταση 10k Ohm](#)
- [Δίοδος \(1N4001 \)](#)
- [Μοτέρ](#)

Κατασκευάστε το κύκλωμα, σύμφωνα με τις παρακάτω σχηματικές αναπαραστάσεις:



Σχήμα 1



Σχήμα 2

Συνδέστε τη **βάση** του τρανζίστορ με μία **αντίσταση 10k Ohm** και στη συνέχεια συνδέστε **την άλλη άκρη της αντίστασης με το pin 9** του Arduino. Ο **εκπομπός** του τρανζίστορ συνδέεται στη **γείωση**, ενώ ο **συλλέκτης** ακριβώς **πριν τη δίοδο**, όπως φαίνεται στο Σχήμα 2. Τέλος στη δίοδο συνδέεται και το **μοτέρ** (το κόκκινο καλώδιο στην πλευρά που συνδέεται με την **τάση +5 V**, ενώ το **μαύρο** στην πλευρά που έχουμε συνδέσει τον **συλλέκτη** του τρανζίστορ).

Ο πυκνωτής που φαίνεται στο Σχήμα 1 δεν είναι απαραίτητος για τη λειτουργία του κυκλώματος και μπορεί να παραληφθεί. Η τοποθέτηση του χρειάζεται μόνο στην περίπτωση που παρατηρήσετε ότι το Arduino κάνει επανεκκίνηση μόνο του.

Τέλος συνδέστε, στις κατάλληλες θέσεις του breadboard, ένα καλώδιο στο pin τροφοδοσίας 5V του Arduino και ένα καλώδιο σε ένα από τα τρία pin γείωσης (Gnd) που βρίσκονται πάνω στο Arduino.

4. Προγραμματισμός του κυκλώματος CIRC-03

Αντιγράψτε τον παρακάτω κώδικα στο προγραμματιστικό περιβάλλον του Arduino.

(Εναλλακτικά κατεβάστε τον από [εδώ](#))

Κώδικας προγραμματισμού του κυκλώματος

```
*| Arduino Experimentation Kit Example Code|
*| CIRC-03 .: Spin Motor Spin .: (Transistor and Motor) |
*-----
*
* The Arduinos pins are great for driving LEDs however if you hook
* up something that requires more power you will quickly break them.
* To control bigger items we need the help of a transistor.
* Here we will use a transistor to control a small toy motor
*
* http://tinyurl.com/d4wht7
*
*/

int motorPin = 9; // define the pin the motor is connected to
// (if you use pin 9,10,11 or 3you can also control speed)

/*
 * setup() - this function runs once when you turn your Arduino on
 * We set the motors pin to be an output (turning the pin high (+5v) or low
 (ground) (-))
 * rather than an input (checking whether a pin is high or low)
 */
void setup()
{
  pinMode(motorPin, OUTPUT);
}

/*
 * loop() - this function will start after setup finishes and then repeat
 * we call a function called motorOnThenOff()
 */

void loop() // run over and over again
{
  motorOnThenOff();
  //motorOnThenOffWithSpeed();
  //motorAcceleration();
}
```

Κώδικας προγραμματισμού του κυκλώματος

```
/*
 * motorOnThenOff() - turns motor on then off
 * (notice this code is identical to the code we used for
 * the blinking LED)
 */
void motorOnThenOff(){
  int onTime = 2500; //the number of milliseconds for the motor to turn on for
  int offTime = 1000; //the number of milliseconds for the motor to turn off for
  digitalWrite(motorPin, HIGH); // turns the motor On
  delay(onTime); // waits for onTime milliseconds
  digitalWrite(motorPin, LOW); // turns the motor Off
  delay(offTime); // waits for offTime milliseconds
}

/*
 * motorOnThenOffWithSpeed() - turns motor on then off but uses speed values as
 well
 * (notice this code is identical to the code we used for
 * the blinking LED)
 */
void motorOnThenOffWithSpeed(){
  int onSpeed = 200; // a number between 0 (stopped) and 255 (full speed)
  int onTime = 2500; //the number of milliseconds for the motor to turn on for
  int offSpeed = 50; // a number between 0 (stopped) and 255 (full speed)
  int offTime = 1000; //the number of milliseconds for the motor to turn off for
  analogWrite(motorPin, onSpeed); // turns the motor On
  delay(onTime); // waits for onTime milliseconds
  analogWrite(motorPin, offSpeed); // turns the motor Off
  delay(offTime); // waits for offTime milliseconds
}

/*
 * motorAcceleration() - accelerates the motor to full speed then
 * back down to zero
 */
void motorAcceleration(){
  int delayTime = 50; //milliseconds between each speed step
  //Accelerates the motor
  for(int i = 0; i < 256; i++){ //goes through each speed from 0 to 255
    analogWrite(motorPin, i); //sets the new speed
    delay(delayTime); // waits for delayTime milliseconds
  }

  //Decelerates the motor
  for(int i = 255; i >= 0; i--){ //goes through each speed from 255 to 0
    analogWrite(motorPin, i); //sets the new speed
    delay(delayTime); // waits for delayTime milliseconds
  }
}
```

Αποθηκεύστε το ως CIRC_03 και στη συνέχεια φορτώστε το πρόγραμμα στην πλακέτα.

Αν το μοτέρ δεν δουλεύει, ελέγξτε ξανά το κύκλωμα σας, καθώς και τις παραμέτρους του προγραμματιστικού περιβάλλοντος του Arduino (όπως έχουν περιγραφεί στο CIRC_01, ενότητα 2).

Ελέγξτε επίσης αν τα τρανζίστορ είναι τα αναγραφόμενα P2N2222AG (TO92) και αν δεν είναι, ελέγξτε αν τα pins τους έχουν την ίδια λειτουργία με αυτά του P2N2222AG (TO92), από το site του κατασκευαστή (σε ορισμένα τρανζίστορ ο εκπομπός και ο συλλέκτης είναι ανάποδα).

Τέλος, αν το Arduino κάνει επανεκκίνηση μόνο του, πρέπει να τοποθετήσουμε στο κύκλωμα τον πυκνωτή που αναφέρθηκε στο α μέρος.

5. Παραμετροποίηση του κυκλώματος CIRC-03

5.1 Αλλαγή της συνάρτησης ελέγχου του μοτέρ (Έλεγχος της ταχύτητας)

Σε προηγούμενο εργαστήριο είδαμε την ικανότητα του Arduino να ελέγχει τη φωτεινότητα ενός LED. Θα χρησιμοποιήσουμε την ίδια ικανότητα για να ελέγξουμε την ταχύτητα του μοτέρ μας. Το Arduino το πετυχαίνει αυτό χρησιμοποιώντας διαμόρφωση πλάτους παλμού (PWM), μόνο που αντί να μεταβάλλει απευθείας την τάση που προέρχεται από το ριπ, μεταβαίνει μεταξύ των δύο καταστάσεων πολύ γρήγορα. Για παράδειγμα, εάν το Arduino διαμορφώνει το πλάτος του παλμού στο 50% βλέπουμε το φως αχνό 50%, επειδή τα μάτια μας δεν είναι αρκετά γρήγορα για να το δουν να ανάβει και να σβήνει. Το ίδιο χαρακτηριστικό λειτουργεί και με τρανζίστορ.

Στη συνάρτηση `loop()` υπάρχουν τρεις συναρτήσεις, δύο εκ των οποίων είναι σχόλια. Αφού διαβάσετε και κατανοήσετε τι λειτουργία έχει η κάθε συνάρτηση, αλλάξτε τη συνάρτηση ελέγχου του μοτέρ.

Στη συνέχεια αποθηκεύστε και τρέξτε το πρόγραμμα για καθεμιά από τις συναρτήσεις. (*CIRC_03_c1 για την δεύτερη συνάρτηση και CIRC_03_c2 για την τρίτη*).