



**ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

---

## **Ενσωματωμένα Συστήματα**

**Ενότητα:** ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ: CIRC-14

Δρ. Μηνάς Δασυγένης

[mdasyg@ieee.org](mailto: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



## Περιεχόμενα

1.Σκοπός της άσκησης.....	4
2.Παραδοτέα.....	4
3.Κατασκευή του κυκλώματος CIRC-14.....	4
4.Προγραμματισμός του κυκλώματος CIRC-14.....	7
5.Παραμετροποίηση του κυκλώματος CIRC-14.....	7
5.1Μείωση της υπερβολικής παρουσίας του κόκκινου χρώματος.....	7
5.2Χρήση HSB (Hue, Saturation & Brightness) για την απεικόνιση των χρωμάτων.....	7
5.3Προσαρμοσμένα κουμπιά.....	8

## 1. Σκοπός της άσκησης

- Χρήση του αναπτυξιακού περιβάλλοντος του Arduino<sup>1</sup> για τη συγγραφή και τη μεταφόρτωση προγραμμάτων στην πλακέτα του Arduino Uno.
- Δημιουργία κυκλώματος ελέγχου ενός RGB LED με τη χρήση ποτενσιόμετρου μεμβράνης.

## 2. Παραδοτέα





- Ένα αρχείο zip με τα project **CIRC\_14**, **CIRC\_14\_c1**, **CIRC\_14\_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: 25 λεπτά.**

Το ποτενσιόμετρο μεμβράνης λειτουργεί ακριβώς όπως ένα ποτενσιόμετρο, αλλά είναι επίπεδο, λεπτό ευλύγιστο και δεν έχει περιστρεφόμενο διακόπτη. Η αντίσταση του μεταβάλλεται απλά ασκώντας πίεση σε διαφορετικά σημεία της επιφάνειάς του. Η αντίσταση κυμαίνεται από 100 μέχρι 10K Ohm και η τιμή που επιστρέφεται μπορεί να χρησιμοποιηθεί για να υπολογιστεί η θέση που πιέστηκε στην επιφάνειά του. Στο παρόν κύκλωμα θα χρησιμοποιηθεί για να ελέγξουμε τα χρώματα ενός RGB LED.

## 3. Κατασκευή του κυκλώματος CIRC-14

Για την εκπόνηση του κυκλώματος CIRC-14 απαιτούνται τα εξής μέρη:

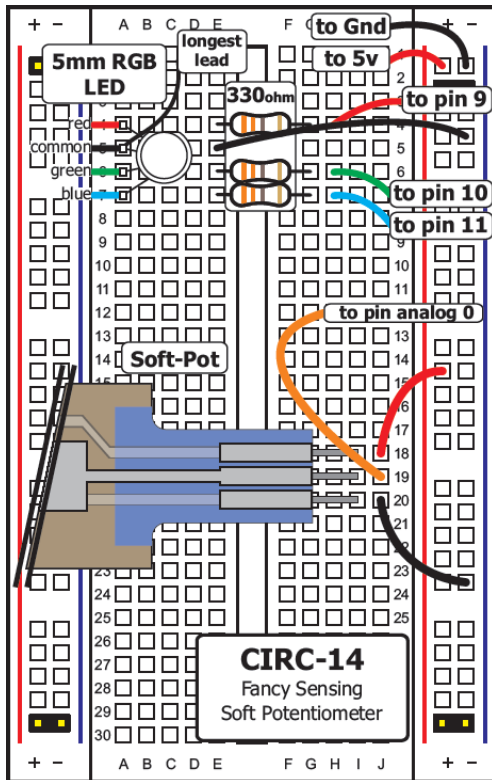
Στοιχείο	Περιγραφή και ποσότητα
	1x <u>RGB LED</u>
	9x <u>Καλώδια</u>
	3x <u>Αντιστάσεις 330 Ohm</u> ( <u>Πορτοκαλί – Πορτοκαλί – Καφέ</u> )
	1x <u>Ποτενσιόμετρο Μεμβράνης</u>

<sup>1</sup> Χρησιμοποιήθηκε υλικό από το [SparkFun Inventors Kit for Arduino](#)

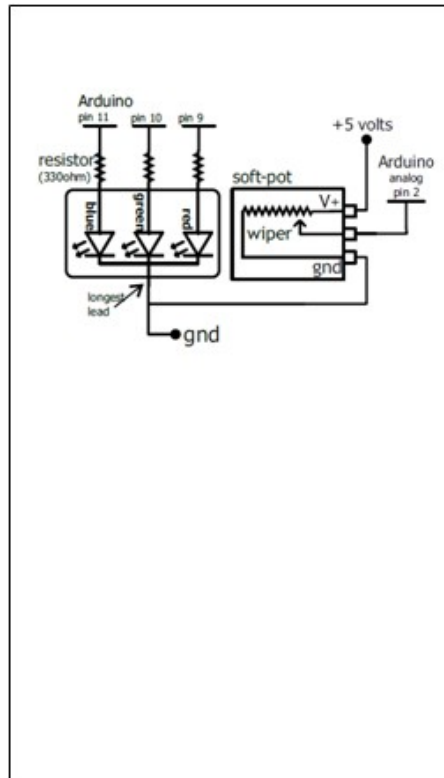
Τα φυλλάδια δεδομένων (datasheets) των υλικών βρίσκονται παρακάτω:

- [RGB LED](#)
- [Αντίσταση 330 Ohm](#)
- [Ποτενσιόμετρο Μεμβράνης](#)

Κατασκευάστε το κύκλωμα, σύμφωνα με τις παρακάτω σχηματικές αναπαραστάσεις:



Σχήμα 1



Σχήμα 2

Τοποθετήστε το RGB LED όπως φαίνεται στο Σχήμα 1.

(Τα pins του LED αναλύονται στο Σχήμα 2).

Στη συνέχεια συνδέστε από μία **αντίσταση 330 Ohm** σε **κάθε pin που αντιστοιχεί σε κάποιο χρώμα**, και έπειτα κάθε αντίσταση με το ανάλογο pin του Arduino. Επίσης συνδέστε και το pin της γείωσης.

Τέλος τοποθετήστε το ποτενσιόμετρο μεμβράνης στο breadboard και συνδέστε το μεσαίο pin στο A0 του Arduino, και τα άλλα δυο στα +5V και τη γείωση αντίστοιχα.

**ΠΡΟΣΟΧΗ:** το ποτενσιόμετρο μεμβράνης **ΔΕΝ** πρέπει να λυγίζει. Με το δάχτυλο σας το ακουμπάτε σε διάφορα σημεία ελαφρά.

## 4. Προγραμματισμός του κυκλώματος CIRC-14

Αντιγράψτε τον παρακάτω κώδικα στο προγραμματιστικό περιβάλλον του Arduino.

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

```

                                Κώδικας προγραμματισμού του κυκλώματος
/*-----
 *| Experimentation Kit for Arduino Example Code|
 *| CIRC-14: Fancy Sensing:(Soft Potentiometer) |
 *-----
 *
 * Will fade an RGB LED from Red-Green-Blue in relation to the
 * soft pot value
 *
 */

// LED leads connected to PWM pins

const int RED_LED_PIN=9; //Red LED Pin
const int GREEN_LED_PIN=10; //Green LED Pin
const int BLUE_LED_PIN=11; //Blue LED Pin

void setup(){
//no need for any code here
}

void loop(){
  int sensorValue=analogRead(0); //read the Soft Pot

  int redValue=constrain(map(sensorValue,0,512,255,0),0,255);
//calculate the red Value (255-0 over the range 0-512)

  int greenValue=constrain(map(sensorValue,0,512,0,255),0,255)-
constrain(map(sensorValue,512,1023,0,255),0,255);
//calculate the green value (0-255 over 0-512 & 255-0 over 512-1023)

  int blueValue=constrain(map(sensorValue,512,1023,0,255),0,255);
//calculate the blue value 0-255 over 512-1023

  // Display the requested color

  analogWrite(RED_LED_PIN, redValue);
  analogWrite(GREEN_LED_PIN, greenValue);
  analogWrite(BLUE_LED_PIN, blueValue);
}

```

Αποθηκεύστε το ως CIRC\_14 και στη συνέχεια φορτώστε το πρόγραμμα στο Arduino.

Σε περίπτωση που το LED δεν ανάβει, ή δείχνει λάθος χρώματα, ελέγξτε αν το κύκλωμά σας είναι σωστό.

Αν λαμβάνετε περίεργα αποτελέσματα, ίσως πιέζετε το ποτενσιόμετρο σε περισσότερες από μία θέσεις. Αυτό είναι φυσιολογικό και μπορεί να χρησιμοποιηθεί για την επίτευξη ιδιαίτερων αποτελεσμάτων.

## 5. Παραμετροποίηση του κυκλώματος CIRC-14

### 5.1 Μείωση της υπερβολικής παρουσίας του κόκκινου χρώματος

Ενδέχεται το κόκκινο χρώμα του LED να είναι εντονότερο από τα άλλα και αυτό να έχει ως συνέπεια τα χρώματα που προκύπτουν να μην είναι τόσο ισορροπημένα. Αν συμβαίνει κάτι τέτοιο, μπορείτε να το διορθώσετε είτε βάζοντας μια αντίσταση περισσότερων Ohm στο pin του κόκκινου χρώματος, είτε αλλάζοντας την παρακάτω γραμμή στο πρόγραμμά σας, από:

```
analogWrite(RED_LED_PIN, redIntensity);
```

σε:

```
analogWrite(RED_LED_PIN, redIntensity/3);
```

Αποθηκεύστε (Ctrl+s) και φορτώστε ξανά το πρόγραμμα στο Arduino, για να επιβεβαιώσετε ότι το πρόβλημα λύθηκε.

### 5.2 Χρήση HSB (*Hue, Saturation & Brightness*) για την απεικόνιση των χρωμάτων

Ένας εναλλακτικός τρόπος απεικόνισης των χρωμάτων πέρα από το RGB είναι το HSB.

Επισκεφθείτε το σχετικό λήμμα στη [Wikipedia](#) για περισσότερες πληροφορίες.

Για να μπορείτε να χρησιμοποιήσετε κι εδώ το ίδιο σύστημα, δημιουργήστε ένα καινούριο Arduino Sketch και αντιγράψτε τον παρακάτω κώδικα

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

#### Κώδικας προγραμματισμού του κυκλώματος

```
//LED leads connected to PWM pins
const int RED_LED_PIN=9;
const int GREEN_LED_PIN=10;
const int BLUE_LED_PIN=11;

//http://www.kasperkamperman.com/blog/arduino/arduino-programming-hsb-to-rgb/

void getRGB(int hue, int sat, int val, int colors[3]){
  /*convert hue, saturation and brightness ( HSB/HSV ) to RGB
  The dim_curve is used only on brightness/value and on saturation
  (inverted).
  This looks the most natural.
  */

  // val=dim_curve[val];
  // sat=255-dim_curve[255-sat];

  int r;
  int g;
  int b;
  int base;

  if (sat==0){//Acromatic color (gray).Hue doesn't mind.
    colors[0]=val;
```

### Κώδικας προγραμματισμού του κυκλώματος

```
colors[1]=val;
colors[2]=val;
}else{

base=((255-sat)*val)>>8;

switch(hue/60) {
case 0:
  r = val;
  g = (((val-base)*hue)/60)+base;
  b = base;
break;

case 1:
  r = (((val-base)*(60-(hue%60)))/60)+base;
  g = val;
  b = base;
break;

case 2:
  r = base;
  g = val;
  b = (((val-base)*(hue%60))/60)+base;
break;

case 3:
  r = base;
  g = (((val-base)*(60-(hue%60)))/60)+base;
  b = val;
break;

case 4:
  r = (((val-base)*(hue%60))/60)+base;
  g = base;
  b = val;
break;

case 5:
  r = val;
  g = base;
  b = (((val-base)*(60-(hue%60)))/60)+base;
break;
}
colors[0]=r;
colors[1]=g;
colors[2]=b;
}
}

int rgb_colors[3];
/// -----

int redIntensity;
int greenIntensity;
int blueIntensity;

void setup() {

}
```



### Κώδικας προγραμματισμού του κυκλώματος

```
void loop() {
  int sensorValue = analogRead(0);
  getRGB(map(sensorValue, 0, 1023, 0, 255), 0xff, 0xff, rgb_colors);
  redIntensity = rgb_colors[0];
  greenIntensity = rgb_colors[1];
  blueIntensity = rgb_colors[2];

  // Display the requested color
  analogWrite(RED_LED_PIN, 255-redIntensity);
  analogWrite(GREEN_LED_PIN, 255-greenIntensity);
  analogWrite(BLUE_LED_PIN, 255-blueIntensity);
}
```

Στη συνέχεια αποθηκεύστε το ως CIRC\_14\_c1 και φορτώστε το στο Arduino για να επιβεβαιώσετε την ορθή λειτουργία του.

## 5.3 Προσαρμοσμένα κουμπιά

Το ποτενσιόμετρο μεμβράνης, μπορεί να χρησιμοποιηθεί για να δημιουργήσετε προσαρμοζόμενα κουμπιά. Για να το πετύχετε αυτό θα πρέπει να ορίσετε ζώνες τιμών στις οποίες θα αντιστοιχεί μία συγκεκριμένη λειτουργία.

Χρησιμοποιήστε το παρακάτω ενδεικτικό κομμάτι κώδικα και τη **Σειριακή οθόνη (Ctrl+Shift+M)** (ανατρέξτε στον κώδικα του εργαστηρίου 10 για να θυμηθείτε πως εκτυπώνεται κάτι ως έξοδος στη σειριακή οθόνη) προκειμένου να προσδιορίσετε τις επιθυμητές τιμές για 2 ή περισσότερα κουμπιά.

```
if(analogRead(0)>minValue && analogRead(0)<maxValue){
  analogWrite(RED_LED_PIN, YourValue);
  analogWrite(GREEN_LED_PIN, YourValue);
  analogWrite(BLUE_LED_PIN, YourValue); }
```

...

Τροποποιήστε σχετικά τον κώδικα για δημιουργία 3 ζωνών. Κάθε ζώνη να αντιστοιχηθεί με ένα χρώμα της επιλογής σας.

Στη συνέχεια αποθηκεύστε το ως CIRC\_14\_c2 και φορτώστε το στο Arduino για να επιβεβαιώσετε την ορθή λειτουργία του.