



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών

3&4^η Εργαστηριακή Άσκηση
Εισαγωγή στο Arduino

Μέλη ομάδας	Αριθμός Μητρώου
Γιώργος Χαραλάμπους	03119706
Δωροθέα Κουμίδου	03119712
Σπύρος Αντώνης	03117649

Επιβλέπων:
Σπυρίδων Αγγελόπουλος

Αθήνα
2023

ΣΤΟΧΟΣ

Ο στόχος της εργαστηριακής άσκησης είναι η εισαγωγή στο Arduino IDE και η εξοικείωση με τη πλακέτα Arduino UNO κατά τη κατασκευή κυκλωμάτων με αναλογικές ή ψηφιακές εισόδους/εξόδους. Η κυκλωματική διάταξη και προσομοίωση των παραδειγμάτων έγινε με τη βοήθεια της ιστοσελίδας Tinkercad(<https://www.tinkercad.com/circuits>) και παρουσιάζεται ο αντίστοιχος κώδικας και η περιγραφή της λειτουργίας του κυκλώματος σε κάθε περίπτωση.

Παράδειγμα 1: Blink

Κώδικας Arduino

```
Blink.ino
1 // the setup function runs once when you press reset or power the board
2 void setup() {
3   // initialize digital pin LED_BUILTIN as an output.
4   pinMode(LED_BUILTIN, OUTPUT);
5 }
6
7 // the loop function runs over and over again forever
8 void loop() {
9   //Ερώτημα 1
10  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
11  delay(1500);
12  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
13  delay(500);
14  //Ερώτημα 2
15  //S.0.S --> .....-/-.-.-/.....-
16  //      0 1 2 3 4 5 6 7 8
17  for(int i=0; i<9;i++){
18    if(i==2 || i==8){
19      digitalWrite(LED_BUILTIN, HIGH);
20      delay(500);
21      digitalWrite(LED_BUILTIN, LOW);
22      delay(1500);
23    }
24    else if(i==3 || i==4){
25      digitalWrite(LED_BUILTIN, HIGH);
26      delay(1500);
27      digitalWrite(LED_BUILTIN, LOW);
28      delay(500);
29    }
30    else if(i==5){
31      digitalWrite(LED_BUILTIN, HIGH);
32      delay(1500);
33      digitalWrite(LED_BUILTIN, LOW);
34      delay(1500);
35    }
36    else{
37      digitalWrite(LED_BUILTIN, HIGH);
38      delay(500);
39      digitalWrite(LED_BUILTIN, LOW);
40      delay(500);
41    }
42  }
43 }
44 }
45
```

Περιγραφή κώδικα

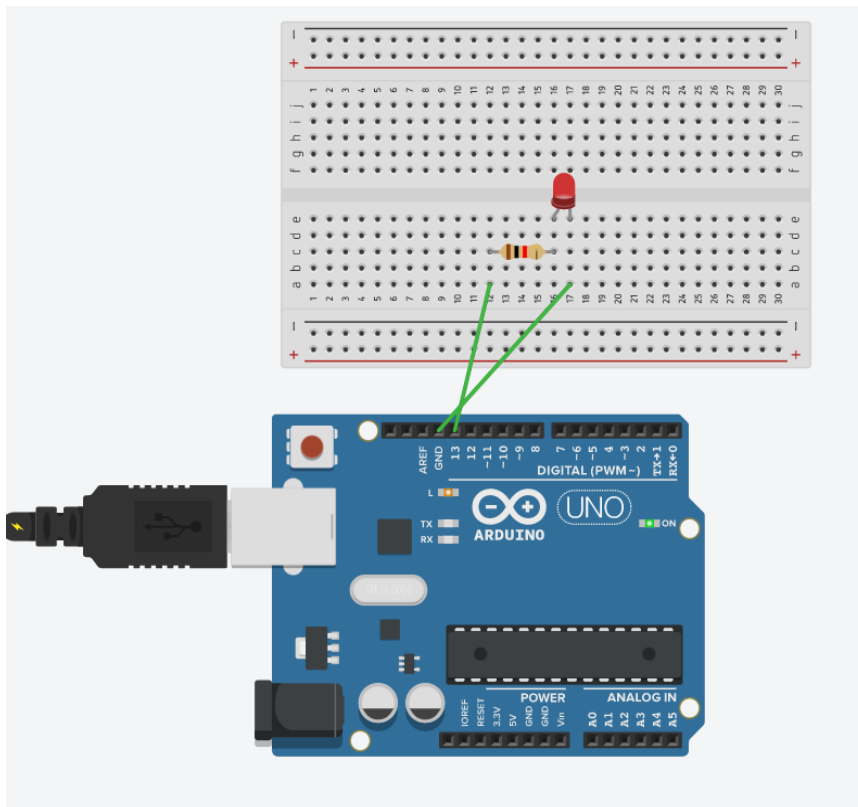
Στο παράδειγμα αυτό αρχικά ζητείται να δημιουργηθεί κώδικας που να ανάβει το ενσωματωμένο LED Arduino του UNO για 1.5sec και να σβήνει για 0.5sec. Αυτό επιτυγχάνεται με τη χρήση της εντολής **delay()**, έτσι ώστε να παραμείνει το LED στην ίδια κατάσταση on/off, σημειώνοντας ότι η χρονική διάρκεια μετρείται σε **milliseconds(ms)**. Στη συνέχεια το Arduino εκπέμπει SOS σε σήματα Morse, λαμβάνοντας υπόψη ότι η παύση μεταξύ συμβόλων αντιστοιχεί σε τελεία(.) δηλαδή 0.5s και η παύση μεταξύ γραμμάτων αντιστοιχεί σε παύλα(-) δηλαδή 1.5s.

Παράδειγμα 2: PhysicalPixel

Κώδικας Arduino

```
PhysicalPixel.ino
1  const int ledPin = 13; // the pin that the LED is attached to
2  int incomingByte;      // a variable to read incoming serial data into
3
4  void setup() {
5      // initialize serial communication:
6      Serial.begin(9600);
7      // initialize the LED pin as an output:
8      pinMode(ledPin, OUTPUT);
9  }
10
11 void loop() {
12     // see if there's incoming serial data:
13     if (Serial.available() > 0) {
14         // read the oldest byte in the serial buffer:
15         incomingByte = Serial.read();
16         // if it's a capital H (ASCII 72), turn on the LED:
17         if (incomingByte == 'H' || incomingByte == 'h') {
18             digitalWrite(ledPin, HIGH);
19         }
20         // if it's an L (ASCII 76) turn off the LED:
21         else if (incomingByte == 'L' || incomingByte == 'l') {
22             digitalWrite(ledPin, LOW);
23         }
24         else{
25             Serial.print("Error: Write H/h for turn on OR L/l for turn off");
26         }
27     }
28 }
29
```

Κυκλωματική διάταξη



Περιγραφή κώδικα

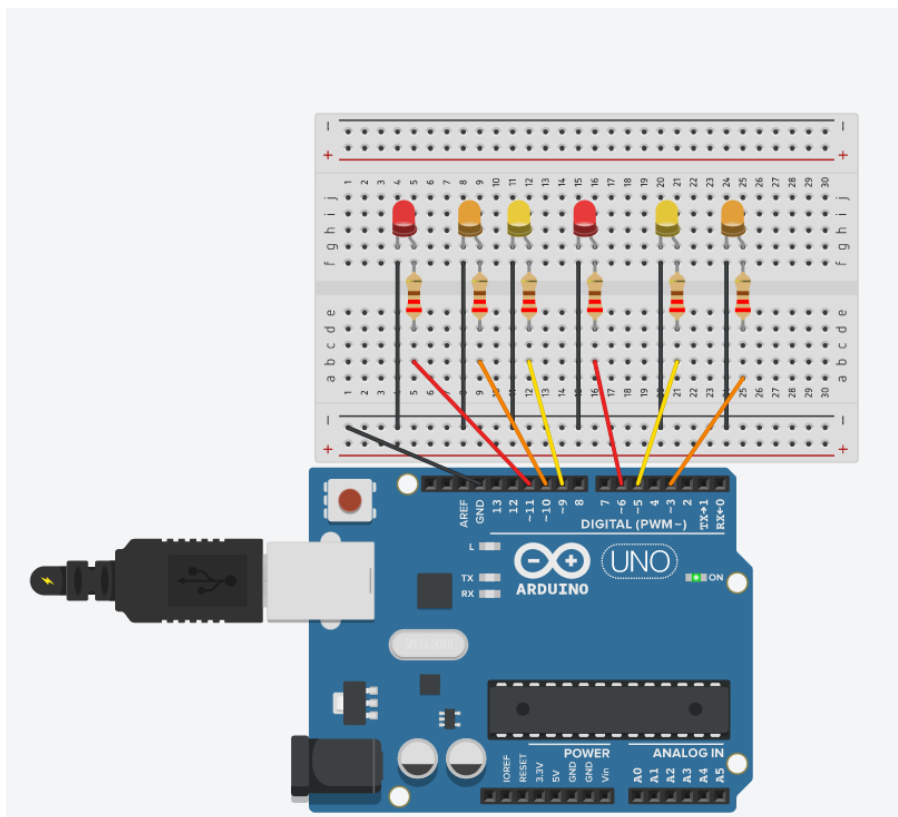
Το παράδειγμα αυτό εξετάζει τον έλεγχο ενός LED μέσω σειριακής επικοινωνίας Arduino-υπολογιστή. Η ταχύτητα μεταφοράς δεδομένων έχει οριστεί στα 9600bps και είναι η πιο σύνηθες προς χρήση. Αν στο Serial Monitor εισαχθούν οι χαρακτήρες 'H' ή 'h' τότε το λαμπάκι ανάβει, ενώ αν εισαχθούν οι χαρακτήρες 'L' ή 'l' το λαμπάκι σβήνει. Σε άλλη περίπτωση εμφανίζεται μήνυμα λάθους.

Παράδειγμα 3: Προσομοίωση φλόγας

Κώδικας Arduino

```
3_LED_Fire.ino
1  int ledPin1 = 9; //Pin 1ου LED
2  int ledPin2 = 10; //Pin 2ου LED
3  int ledPin3 = 11; //Pin 3ου LED
4  int ledPin4 = 6; //Pin 4ου LED
5  int ledPin5 = 5; //Pin 5ου LED
6  int ledPin6 = 3; //Pin 6ου LED
7  int pins[6] = {9,10,11,6,5,3};
8
9  void setup() {
10     for(int i=0; i<6;i++){
11         pinMode(pins[i], OUTPUT); //Ορισμός ως Pin εξόδου
12     }
13     randomSeed(analogRead(A0)); //Λήψη τυχαίων αριθμών βάσει της εισόδου A0
14 }
15 void loop() {
16     for(int i=0; i<6;i++){
17         analogWrite(pins[i], random(0, 130) + 125); //Τυχαίο duty cycle μεταξύ των τιμών 125 & 255
18     }
19     delay(random(0, 120)); //Τυχαία καθυστέρηση μεταξύ 0 & 120 ms
20 }
21
22
```

Κυκλωματική διάταξη



Περιγραφή κώδικα

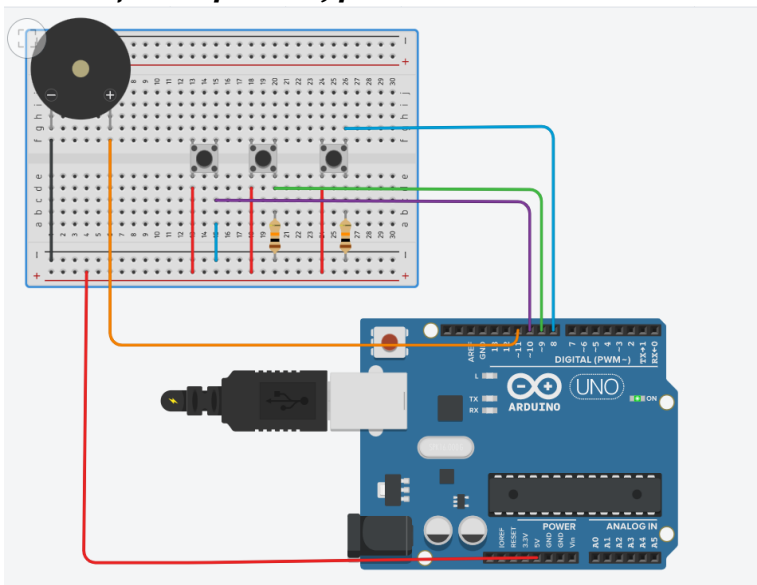
Το παράδειγμα αυτό είναι μια απλή προσομοίωση του φωτός μιας φωτιάς με τη χρήση 6 LEDs. Οι αριθμοί της καθυστέρησης και της φωτεινότητας για κάθε LED έχουν οριστεί τυχαία με τη χρήση της εντολής **random()**. Αξίζει να σημειωθεί ότι τα pins εξόδου που έχουν επιλεγθεί πρέπει είναι PWM pins(), αφού η έξοδος μας είναι αναλογική και παίρνει τιμές από 0 μέχρι 255. Επιπλέον με την χρήση ενός for-loop δεν χρειάζεται επανάληψη των εντολών **pinMode()** και **analogWrite()** για κάθε LED.

Παράδειγμα 4: Piano

Κώδικας Arduino

```
5_Piano.ino
1  int speaker_pin = 11;
2  int Re_pin = 10;
3  int Fa_pin = 9;
4  int Sol_pin = 8;
5
6  int notes[] = {294, 349, 392};
7
8  void setup() {
9    pinMode(Re_pin, INPUT);
10   pinMode(Fa_pin, INPUT);
11   pinMode(Sol_pin, INPUT);
12   pinMode(speaker_pin, OUTPUT);
13 }
14
15 void loop() {
16   while (digitalRead(Re_pin) == HIGH)
17   {
18     tone(speaker_pin, notes[0], 200);
19   }
20   while (digitalRead(Fa_pin) == HIGH)
21   {
22     tone(speaker_pin, notes[1], 200);
23   }
24   while (digitalRead(Sol_pin) == HIGH)
25   {
26     tone(speaker_pin, notes[2], 200);
27   }
28 }
29
```

Κυκλωματική διάταξη



Περιγραφή κώδικα

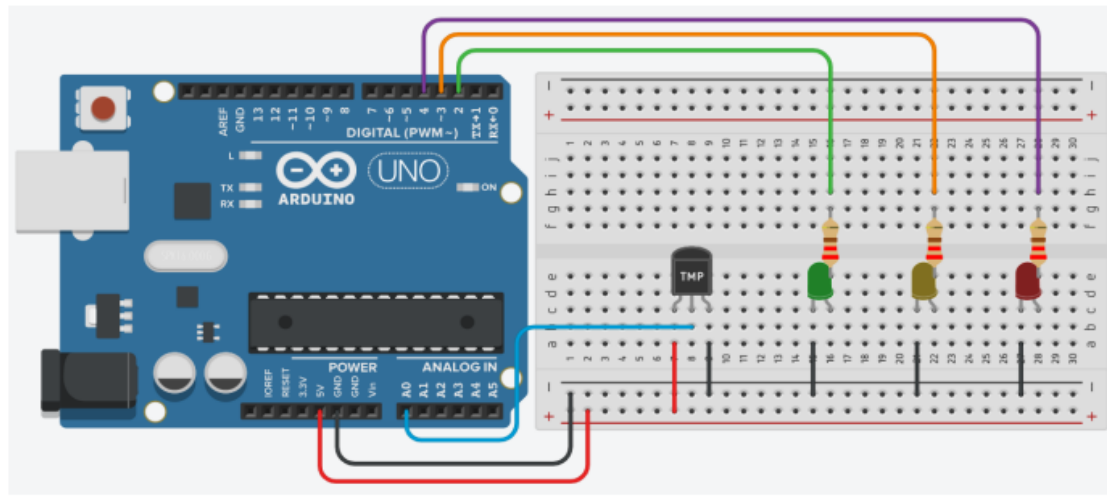
Η κυκλωματική διάταξη αναπαράγει τις νότες Ρε, Φα , και Σολ με το πάτημα των αντίστοιχων πλήκτρων, που λειτουργούν ουσιαστικά ως διακόπτες. Η κάθε νότα αντιστοιχεί σε μία συγκεκριμένη ακουστική συχνότητα, η οποία στέλνεται στο buzzer μέσω του αντίστοιχου κουμπιού. Η αντίσταση των 10kΩ ρυθμίζει την ένταση του ήχου στο ηχείο. Η αφαίρεση της ή η προσθήκη μίας μεγαλύτερης αντίστασης, αυξάνει ή μειώνει αντίστοιχα την ένταση του ήχου.

Παράδειγμα 5: Αισθητήρας θερμοκρασίας TMP36

Κώδικας Arduino

```
6_TMP36.ino
1  int green_pin = 2;
2  int orange_pin = 3;
3  int red_pin = 4;
4  int tempPin = A0;
5
6  void setup()
7  {
8      pinMode(green_pin, OUTPUT);
9      pinMode(orange_pin, OUTPUT);
10     pinMode(red_pin, OUTPUT);
11     Serial.begin(9600);
12 }
13
14 void loop()
15 {
16     int tempValue = analogRead(tempPin);
17     float temp = tempValue * 5000.0 / 1024.0;
18     float temp_C = (temp - 500) / 10;
19     Serial.println(tempValue);
20     Serial.print("Temp = ");
21     Serial.print(temp_C);
22     Serial.println("C");
23     if (temp_C <= 25) {
24         digitalWrite(green_pin, HIGH);
25         digitalWrite(orange_pin, LOW);
26         digitalWrite(red_pin, LOW);
27     }
28     else if (temp_C > 25 && temp_C <= 35) {
29         digitalWrite(green_pin, HIGH);
30         digitalWrite(orange_pin, HIGH);
31         digitalWrite(red_pin, LOW);
32     }
33     else {
34         digitalWrite(green_pin, HIGH);
35         digitalWrite(orange_pin, HIGH);
36         digitalWrite(red_pin, HIGH);
37     }
38     delay(1000);
39 }
40
```

Κυκλωματική διάταξη



Περιγραφή κώδικα

Το πρόγραμμα αυτό διαβάζει την έξοδο ενός αισθητήρα θερμοκρασίας TMP36 και ανάλογα με τη θερμοκρασία ανάβουν διαδοχικά τα LED σύμφωνα με τον πιο κάτω πίνακα:

Θερμοκρασία	LEDs
$\theta \leq 25$	Πράσινο
$25 < \theta \leq 35$	Πράσινο & Πορτοκαλί
$\theta > 35$	Πράσινο & Πορτοκαλί & Κόκκινο

Σημειώνεται εδώ ότι χρησιμοποιείται τάση τροφοδοσίας 5V και ανάλυση ADC Arduino UNO 10bits. Αυτό σημαίνει ότι η αναλογική τιμή του αισθητήρα παίρνει τιμές από 0-5V με ενδιάμεσο αριθμό δειγμάτων ίσο με $2^{10}=1024$. Αλλάζοντας τώρα την τάση τροφοδοσίας και την ανάλυση έχουμε ένα καινούριο εύρος τιμών από 0-3.3V και πλέον $2^{12}= 4096$ δείγματα. Αυτό μπορεί να γίνει με αλλαγή της μεταβλητής temp, δηλαδή:

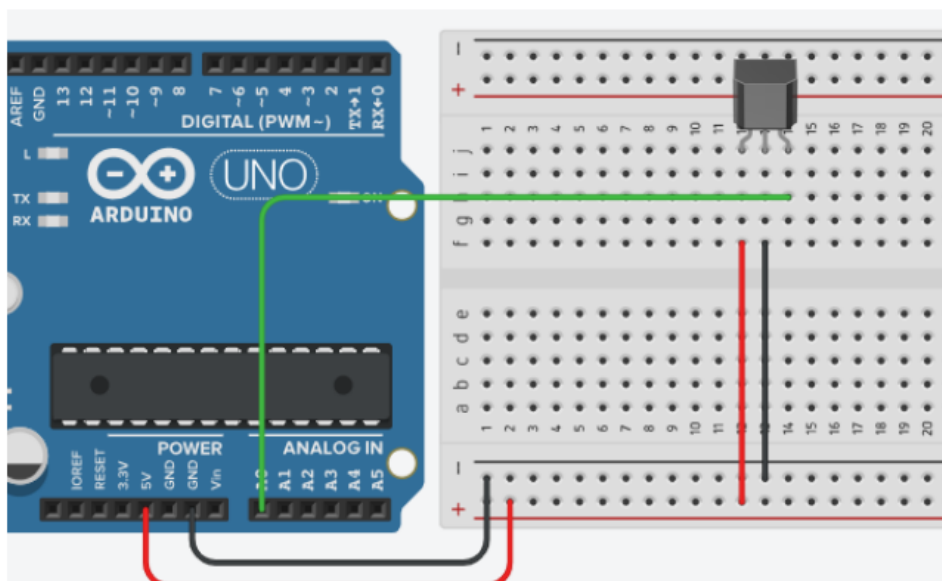
```
float temp = tempValue * 3300.0 / 4096.0.
```


Παράδειγμα 6: Hall SS49

Κώδικας Arduino

```
7_Hall_SS49.ino
1  int pin = A0;           //Pin εισόδου αισθητήρα Hall
2  int vcc = 5000;         //Τάση τροφοδοσίας αισθητήρα Hall σε mV
3  int ledpin=3;           //Pin Led
4
5  int value;              //Μεταβλητή τιμής στο εύρος 0-1023
6  int voltage;            //Μεταβλητή τάσης στο εύρος 0-Vcc
7  int field;              //Μεταβλητή τιμής πεδίου σε mT
8  int led_intensity;      //Μεταβλητή έντασης φωτεινότητας LED
9
10 void setup() {
11   pinMode(ledpin, OUTPUT);
12   Serial.begin(9600); //Εναρξη σειριακής επικοινωνίας
13 }
14
15 void loop() {
16   value = analogRead(pin); //Ανάγνωση τιμής του αισθητήρα Hall
17   voltage = value * (vcc / 1024.0); //Μετατροπή σε τιμή τάσης (mV)
18   field = map(voltage, 1000, 4000, -100, 100); //Υπολογισμός μαγνητικού πεδίου (mT)
19   led_intensity= map(abs(field),0,100,0,255);
20
21   if(abs(field)>5){
22     | | analogWrite(ledpin, led_intensity);
23   }
24   if(voltage != 2500){ // Όταν Vout=2,5V ==> απουσία μαγνητικού πεδίου
25     //Εμφάνιση αποτελεσμάτων
26     Serial.print("Value: ");
27     Serial.print(value);
28     Serial.print("\t");
29     Serial.print(voltage);
30     Serial.println(" mV");
31     Serial.print("Magnetic Field: ");
32     Serial.print(field);
33     Serial.println(" mT");
34     Serial.println("=====");
35     delay(100); //Λήψη νέας τιμής κάθε 100msec
36   }
37   else{
38     Serial.print(" No magnetic Field: ");
39     Serial.print(field);
40     Serial.println(" mT");
41     Serial.println("=====");
42   }
43 }
```

Κυκλωματική διάταξη



Περιγραφή κώδικα

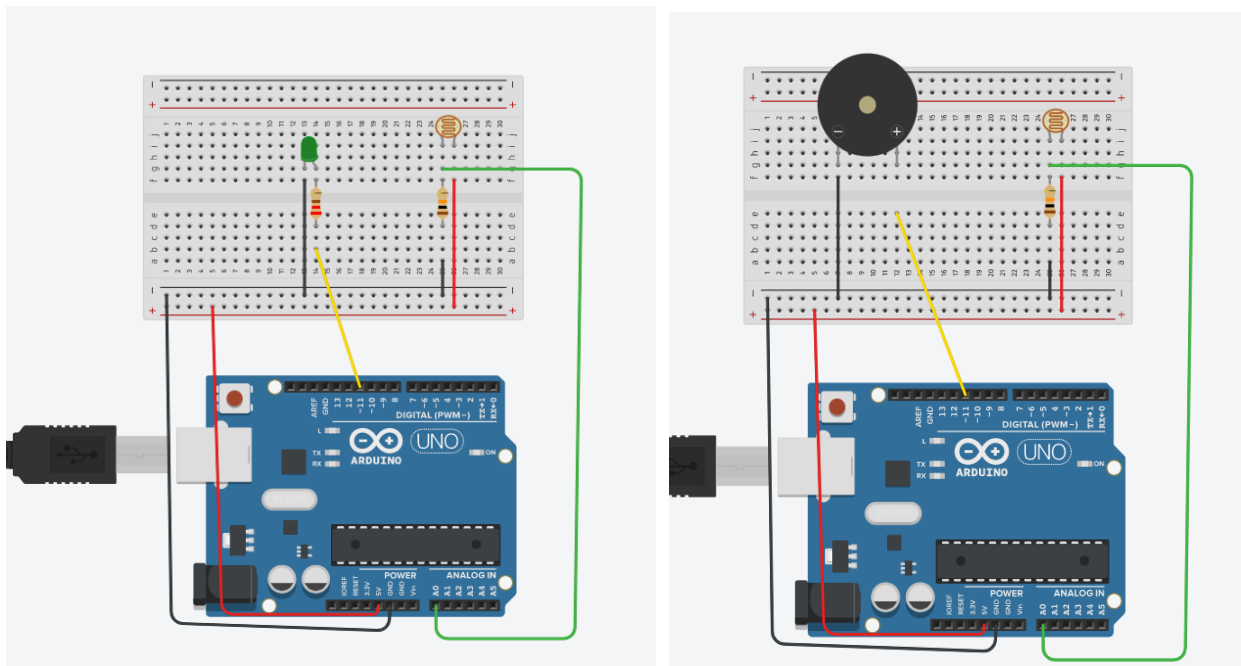
Στο παράδειγμα αυτό έχουμε ένα μαγνητικό αισθητήρα, ο οποίος αναγνωρίζει την παρουσία μαγνήτη και εμφανίζει την αντίστοιχη τιμή του μαγνητικού πεδίου σε mT στο Serial Monitor. Η μετατροπή μεταξύ τάσης και μαγνητικής έντασης γίνεται με τη βοήθεια της εντολής **map()**, όπου τα 4V αντιστοιχούν στα 1000Gauss και τα -1000Gauss σε 1V. Μέσω της σχέσης αυτής γίνεται εύκολα αντιληπτό ότι στα 2.5V έχουμε απουσία μαγνητικού πεδίου. Αν η ένταση του μαγνητικού πεδίου είναι μεγαλύτερη από 5mT=50Gauss, τότε ανάβει μια LED, της οποίας η ένταση είναι ανάλογη με τη τιμή του μαγνητικού πεδίου. Όσο μεγαλύτερο σε απόλυτη τιμή, τόσο μεγαλύτερη η φωτεινότητα της LED, γεγονός που επιτυγχάνεται και πάλι με τη χρήση του **map()**. Οι μετρήσεις στο monitor παρουσιάζονται κάθε 100ms.

Παράδειγμα 7: Αισθητήρας φωτεινότητας

Κώδικας Arduino

```
8_Light_sensor.ino
1  int incomingByte;
2  int value;
3
4  void setup() {
5      pinMode(11, OUTPUT);
6      Serial.begin(9600);
7      Serial.println("Please enter A for Analog or D for Digital:");
8  }
9
10 void loop() {
11     if (Serial.available() > 0) {
12         incomingByte = Serial.read();
13         if (incomingByte == 'A') {
14             while (Serial.available() == 0) {
15                 analog();
16             }
17         }
18         else if (incomingByte == 'D') {
19             while (Serial.available() == 0) {
20                 digital();
21             }
22         }
23         else {
24             Serial.println("Unknown command");
25         }
26     }
27 }
28
29 void analog() {
30     value = analogRead(A0);
31     Serial.print(value);
32     Serial.print("\t");
33     int analogValue = map(value, 512, 1023, 0, 255);
34     digitalWrite(11, analogValue);
35     Serial.println(analogValue);
36 }
37
38 void digital() {
39     value = analogRead(A0);
40     Serial.print(value);
41     Serial.print("\t");
42     boolean digitalValue = map(value, 0, 600, HIGH, LOW);
43     digitalWrite(11, digitalValue);
44     Serial.println(digitalValue);
45 }
46
```

Κυκλωματική διάταξη



Περιγραφή κώδικα

Σε αυτό το παράδειγμα ζητείται η δημιουργία ενός προγράμματος που διαβάζει ψηφιακά ή αναλογικά την έξοδο μιας φωτοαντίστασης και ανάβει ή σβήνει αντίστοιχα ένα LED (αριστερό κύκλωμα). Η φωτοαντίσταση έχει χαμηλή αντίσταση στο φως και υψηλή αντίσταση σε χαμηλή φωτεινότητα, επομένως η θέση της στο διαιρέτη τάσης, αποσκοπεί σε έλεγχο φωτεινότητας ή συσκότισης. Στο κύκλωμα, μεγάλη φωτεινότητα οδηγεί σε τάσεις εξόδου κοντά στη τάση τροφοδοσίας των 5V (αναλογική τιμή 1023), ενώ σε σκιά/συσκότιση προσεγγίζει τα 0V(αναλογική τιμή 0). Συνεπώς σε αναλογική λειτουργία, το πρόγραμμα μειώνει την ένταση του LED, όσο περισσότερο σκιάζεται ο αισθητήρας. Κατά την ψηφιακή λειτουργία, το λαμπάκι μένει απενεργοποιημένο, ωσότου ο αισθητήρας να περάσει ένα κατώφλι φωτεινότητας.

Κατόπιν, ζητείται η τροποποίηση του κυκλώματος έτσι ώστε η φωτοαντίσταση να μεταβάλλει τη συχνότητα ενός buzzer(δεξιά κύκλωμα). Ο τρόπος λειτουργίας παραμένει ίδιος, ωστόσο χρειάζεται να γίνει αλλαγή στη αναλογική λειτουργία με την εντολή:

```
int analogValue = map(value, 512, 1023, 100, 1000);
```