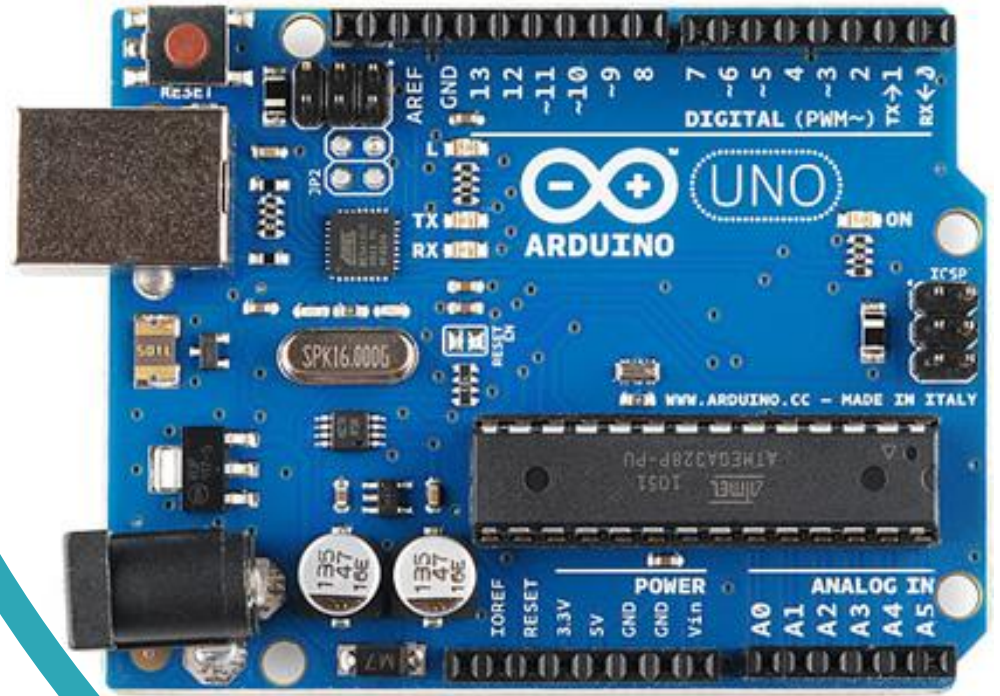




ARDUINO

ΚΟΥΤΣΙΟΥΚΗ ΕΥΑΓΓΕΛΙΑ

Εισαγωγή στο Arduino Uno	4
Τι είναι το Arduino	4
Πώς δουλεύει το Arduino	4
Τα μέρη του Arduino Uno	5
Τι χρειαζόμαστε για να δουλέψουμε	6
Γιατί το χρησιμοποιούμε στο μάθημα.....	6
Το περιβάλλον του Arduino IDE – βασικά κουμπιά	7
Ρυθμίσεις πριν την πρώτη χρήση	7
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1 - Εισαγωγή στο Arduino Uno με Arduino IDE	8
ΕΦΑΡΜΟΓΗ 1: Άναμμα ενός LED	8
ΕΦΑΡΜΟΓΗ 2: Εναλλαγή δύο LED	11
ΕΦΑΡΜΟΓΗ 3: LED με κουμπί.....	14
Μικρό Τεστ Ανατροφοδότησης	17
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2 - Φωτεινός Σηματοδότης Ελέγχου Κυκλοφορίας Οχημάτων και Πεζών	18
ΕΦΑΡΜΟΓΗ 1: Σηματοδότης Οχημάτων (3 LED)	18
ΕΦΑΡΜΟΓΗ 2: Σηματοδότης Οχημάτων & Πεζών.....	22
ΕΦΑΡΜΟΓΗ 3: Σηματοδότης με Κουμπί Προτεραιότητας Πεζών	26
Μικρό Τεστ Ανατροφοδότησης	33
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 3 - ΤΡΕΧΑΝΤΗΡΙ ΜΕ LED	34
ΕΦΑΡΜΟΓΗ 1: ΔΕΞΙΑ ΟΛΙΣΘΗΣΗ.....	34
ΕΦΑΡΜΟΓΗ 2: ΑΡΙΣΤΕΡΗ ΟΛΙΣΘΗΣΗ	37
ΕΦΑΡΜΟΓΗ 3: «ΠΗΓΑΙΝΕ-ΕΛΑ» (Ping-Pong)	39
Μικρό Τεστ Ανατροφοδότησης	42
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 4 - ΡΥΘΜΙΣΗ ΕΝΤΑΣΗΣ ΦΩΤΕΙΝΟΤΗΤΑΣ	43
ΕΦΑΡΜΟΓΗ 1: Μέτρηση τάσης με τη βοήθεια ποτενσιόμετρου και χρήση της σειριακής οθόνης.....	43
ΕΦΑΡΜΟΓΗ 2: Κύκλωμα ελέγχου φωτεινότητας LED, με τη βοήθεια ποτενσιόμετρου.....	45
Μικρό Τεστ Ανατροφοδότησης	48
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 5 - ΑΝΙΧΝΕΥΤΗΣ ΣΚΟΤΟΥΣ ΜΕ ΦΩΤΟΑΝΤΙΣΤΑΣΗ	49
ΕΦΑΡΜΟΓΗ: Σε συνθήκες σκότους ανάβει ένα LED, ενώ σε συνθήκες φωτός το LED παραμένει σβηστό.	49

Μικρό Τεστ Ανατροφοδότησης	53
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 6 - ΕΝΔΕΙΚΤΗΣ LED ΕΝΟΣ ΨΗΦΙΟΥ	54
ΕΦΑΡΜΟΓΗ 1: Διαδοχική και συνεχόμενη απεικόνιση των αριθμών 0–9 σε έναν ενδείκτη LED επτά τμημάτων.	54
ΕΦΑΡΜΟΓΗ 2: Διαδοχική απεικόνιση των αριθμών 0–9 με κάθε πάτημα ενός μπουτόν.....	58
Μικρό Τεστ Ανατροφοδότησης	62
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 7 - ΕΜΦΑΝΙΣΗ ΜΗΝΥΜΑΤΟΣ ΣΕ ΟΘΟΝΗ ΥΓΡΩΝ ΚΡΥΣΤΑΛΛΩΝ (LCD 2x16)...	63
ΕΦΑΡΜΟΓΗ: Εμφάνιση μηνύματος σε οθόνη LCD 2x16	63
Μικρό Τεστ Ανατροφοδότησης	67
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 8 - ΜΕΤΡΗΤΗΣ ΘΕΡΜΟΚΡΑΣΙΑΣ	68
ΕΦΑΡΜΟΓΗ: Μέτρηση της θερμοκρασίας	68
Μικρό Τεστ Ανατροφοδότησης	72
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 9 - ΣΥΝΑΓΕΡΜΟΣ ΕΝΕΡΓΟΠΟΙΟΥΜΕΝΟΣ ΑΠΟ ΗΧΟ	73
ΕΦΑΡΜΟΓΗ : Δημιουργία ενός απλού συναγερμού κρότου	73
Μικρό Τεστ Ανατροφοδότησης	77
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 10 - Ανίχνευση απόστασης με αισθητήρα υπερήχων	78
ΕΦΑΡΜΟΓΗ 1: Εντοπισμός αντικειμένου που πλησιάζει στα 5cm. Όταν πλησιάσει το αντικείμενο, ηχεί ο βομβητής.....	78
ΕΦΑΡΜΟΓΗ 2: Έξι LED ανάβουν ανάλογα με την απόσταση	82
Μικρό Τεστ Ανατροφοδότησης	87
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 11 - ΕΛΕΓΧΟΣ ΣΕΡΒΟΚΙΝΗΤΗΡΑ ΜΕ ΠΟΤΕΝΣΙΟΜΕΤΡΟ	88
ΕΦΑΡΜΟΓΗ : Έλεγχος γωνίας στροφής σερβοκινητήρα με ποτενσιόμετρο.....	88
Μικρό Τεστ Ανατροφοδότησης	92
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 12 - ΨΗΦΙΑΚΟ ΖΑΡΙ.....	93
ΕΦΑΡΜΟΓΗ 1: Ψηφιακό Ζάρι με LED	93
ΕΦΑΡΜΟΓΗ 2: Ψηφιακό Ζάρι με 7 – Segment Display.....	97
Μικρό Τεστ Ανατροφοδότησης	100
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 13 - ΗΛΕΚΤΡΟΝΙΚΟ ΖΑΡΙ ΜΕ 7-SEGMENT DISPLAY ΚΑΙ BUZZER	101
ΕΦΑΡΜΟΓΗ: Ηλεκτρονικό Ζάρι με 7-Segment Display και Buzzer.....	101
Μικρό Τεστ Ανατροφοδότησης	106

Εισαγωγή στο Arduino Uno

◆ Τι είναι το Arduino;

- Το Arduino είναι μια μικρή πλακέτα που λειτουργεί σαν μικρός υπολογιστής.
- Ο «εγκέφαλος» του Arduino Uno είναι ο μικροελεγκτής ATmega328P, που εκτελεί τις εντολές μας.

Μπορεί να:

- Διαβάζει δεδομένα από αισθητήρες (π.χ. θερμοκρασία, φως, κουμπιά).
- Στέλνει εντολές σε εξαρτήματα (π.χ. LED, μοτέρ, οθόνες).

💡 *Χρησιμοποιείται σε:*

ρομποτική, αυτοματισμούς, συστήματα ασφαλείας, IoT (έξυπνες συσκευές), αλλά και σε σχολικά πειράματα.

◆ Πώς δουλεύει το Arduino;

Η λειτουργία του Arduino βασίζεται σε τρεις απλές κινήσεις:

1. Γράφουμε τον κώδικα στον υπολογιστή (Arduino IDE).
2. Συνδέουμε το Arduino Uno με USB και φορτώνουμε τον κώδικα.
3. Το Arduino εκτελεί το πρόγραμμα συνεχώς (loop).

💡 *Παράδειγμα:*

Αν γράψουμε «άναψε το LED», το Arduino θα στείλει ρεύμα στο LED και θα το ανάψει.

Αν γράψουμε «όταν πατηθεί το κουμπί → άναψε το LED», θα το κάνει αυτόματα κάθε φορά που πατιέται το κουμπί.

◆ Τα μέρη του Arduino Uno

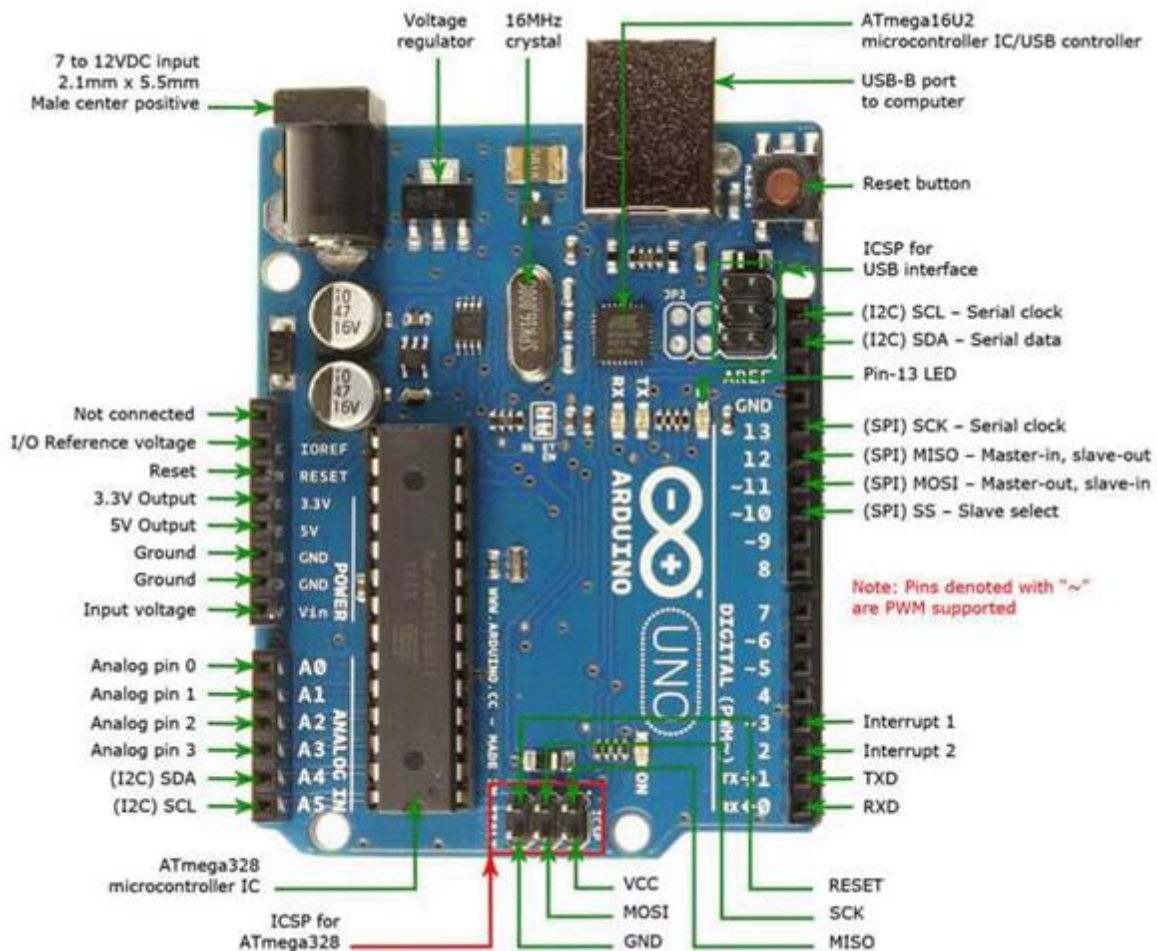
- Ψηφιακά pin (D0–D13):

Για λειτουργίες ON/OFF. Παράδειγμα: άναμμα/σβήσιμο LED.

- Αναλογικά pin (A0–A5):

Για μετρήσεις τιμών. Παράδειγμα: ποτενσιόμετρο → τιμή από 0 έως 1023.

- 5V / 3.3V: παροχή ρεύματος σε εξαρτήματα.
- GND (Γείωση): απαραίτητη σύνδεση για να κλείσει το κύκλωμα.
- USB θύρα: σύνδεση με τον υπολογιστή (τροφοδοσία + προγραμματισμός).



◆ Τι χρειαζόμαστε για να δουλέψουμε;

- Arduino Uno
- Καλώδιο USB
- Breadboard (πίνακας δοκιμών)
- Καλώδια jumper (αρσενικά–θηλυκά)
- Εξαρτήματα (LED, αντιστάσεις, κουμπιά, αισθητήρες, μοτέρ, οθόνες κ.ά.)

◆ Γιατί το χρησιμοποιούμε στο μάθημα;

- ✓ Για να μάθουμε προγραμματισμό με πράξη
- ✓ Για να γνωρίσουμε ηλεκτρονικά κυκλώματα με ασφάλεια (χαμηλή τάση 5V)
- ✓ Για να δημιουργούμε δικές μας έξυπνες κατασκευές
- ✓ Γιατί είναι εύκολο και ανοιχτό σε όλους – υπάρχουν άπειρα παραδείγματα και κοινότητες υποστήριξης

💡 Συμπέρασμα

Το Arduino είναι ένα εργαλείο μάθησης και δημιουργίας.

Μας δίνει τη δυνατότητα να μετατρέψουμε ιδέες σε πραγματικά κυκλώματα με πολύ απλό και πρακτικό τρόπο.



Το περιβάλλον του Arduino IDE – βασικά κουμπιά

Όταν ανοίξουμε το Arduino IDE, βλέπουμε ένα παράθυρο όπου γράφουμε τον κώδικά μας. Πάνω αριστερά υπάρχουν τα πιο σημαντικά κουμπιά:

1.  Verify (Επαλήθευση)
 - Ελέγχει αν ο κώδικας έχει λάθη.
 - Δεν τον στέλνει ακόμα στην πλακέτα.
 - Αν όλα είναι σωστά → εμφανίζει μήνυμα “Compilation completed”.

2.  Upload (Μεταφόρτωση)
 - Στέλνει το πρόγραμμα από τον υπολογιστή στην πλακέτα Arduino.
 - Αυτό είναι το πιο σημαντικό κουμπί: “φορτώνει” το πρόγραμμα.

3.  New (Νέο)
 - Δημιουργεί ένα νέο, άδειο αρχείο κώδικα (sketch).

4.  Open (Άνοιγμα)
 - Ανοίγει παλιό πρόγραμμα (sketch) που έχουμε αποθηκεύσει.

5.  Save (Αποθήκευση)
 - Αποθηκεύει το πρόγραμμα που γράψαμε για να μην χαθεί.



Ρυθμίσεις πριν την πρώτη χρήση

Πριν φορτώσουμε προγράμματα πρέπει να ρυθμίσουμε:

- Board (Πλακέτα) → Arduino Uno (από το μενού Tools → Board).
- Port (Θύρα) → επιλέγουμε την USB θύρα στην οποία είναι συνδεδεμένο το Arduino.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1

Εισαγωγή στο Arduino Uno με Arduino IDE

Στόχοι

- Να μάθουμε πώς συνδέουμε και προγραμματίζουμε ένα LED.
- Να κατανοήσουμε τη χρήση κουμπιού (button) για έλεγχο LED.
- Να προγραμματίσουμε δύο LED να αναβοσβήνουν εναλλάξ.

ΕΦΑΡΜΟΓΗ 1: Άναμμα ενός LED

Θεωρία

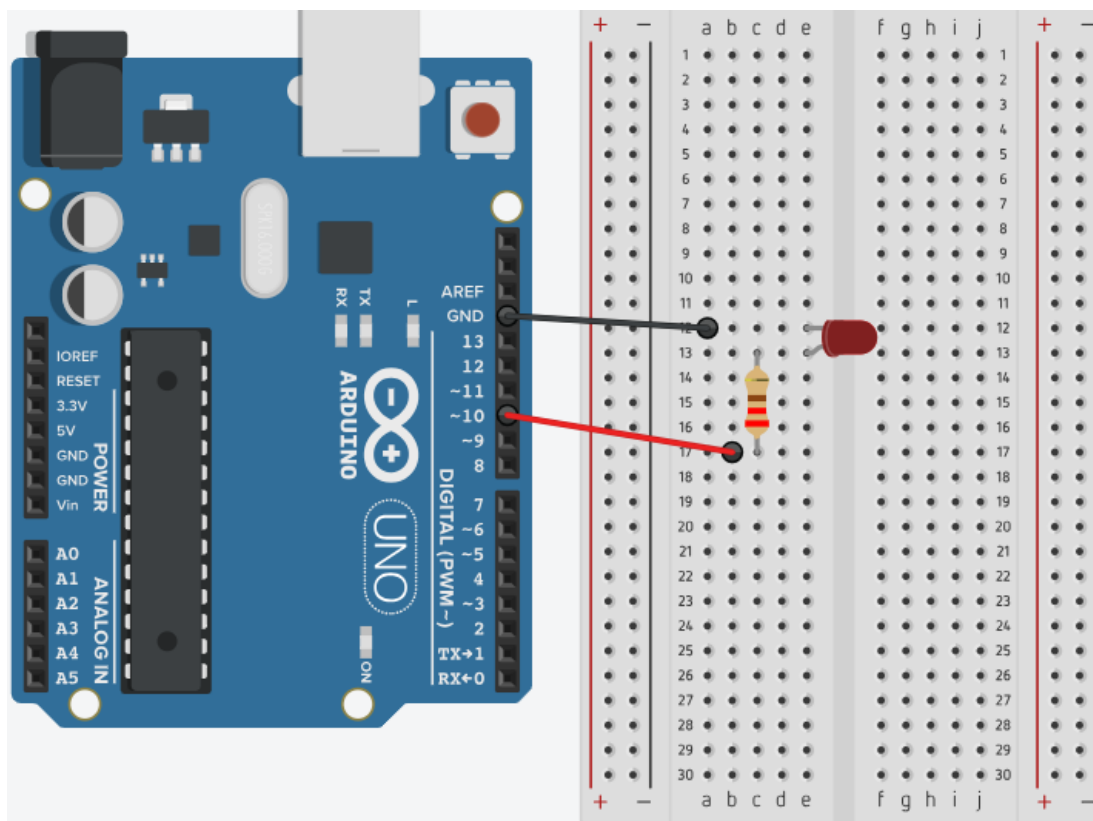
Το LED είναι δίοδος που ανάβει όταν περάσει ρεύμα. Χρειάζεται αντίσταση για προστασία, για να περιορίσουμε το ρεύμα και να μην καεί.

- Μακρύ ποδαράκι (Άνοδος, +) → στο pin μέσω αντίστασης
- Κοντό ποδαράκι (Κάθοδος, -) → GND

Υλικά

- 1 × Arduino Uno
- 1 × LED
- 1 × Αντίσταση 220 Ω
- Καλώδια σύνδεσης (jumper wires)
- Breadboard
- Καλώδιο USB για σύνδεση του Arduino με τον υπολογιστή

🔌 Σχέδιο συνδεσμολογίας



🔧 Βήματα Σύνδεσης

1. Τοποθετήστε το LED στο breadboard.
2. Συνδέστε την άνοδο (+, μακρύ ποδαράκι) σε σειρά με την αντίσταση.
3. Το άλλο άκρο της αντίστασης τοποθετήστε το στο pin 10 του Arduino.
4. Συνδέστε την κάθοδο (-, κοντό ποδαράκι) στο GND του Arduino.

★ Tip: Αν το LED δεν ανάβει, πιθανόν το έχετε βάλει ανάποδα.

☞ Δηλαδή:

Το LED θα συνδεθεί στο pin 10 του Arduino μέσω αντίστασης.

★ Στο breadboard η αντίσταση και το LED συνδέονται σε σειρά.



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επέλεξε Arduino Uno.
4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επέλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.
7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περίμενε να εμφανιστεί το μήνυμα “Upload completed”.



Κώδικας

```
// Εφαρμογή 1: Άναμμα ενός LED

int led = 10;    // Δηλώνουμε ότι στο pin 10 έχουμε το LED

void setup() {
  pinMode(led, OUTPUT); // Ορίζουμε το pin 10 σαν έξοδο (OUTPUT)
}

void loop() {
  digitalWrite(led, HIGH); // Στέλνουμε ρεύμα στο LED → ανάβει
  delay(1000);             // Περιμένουμε 1000ms (1 δευτερόλεπτο)

  digitalWrite(led, LOW);  // Σταματάμε το ρεύμα → σβήνει
  delay(1000);             // Περιμένουμε πάλι 1 δευτερόλεπτο
}
```

Επεξήγηση εντολών:

- `int led = 10;` → αποθηκεύουμε τον αριθμό του pin σε μια μεταβλητή.
- `pinMode(pin, OUTPUT);` → δηλώνουμε ότι το pin είναι έξοδος.
- `digitalWrite(pin, HIGH);` → στέλνουμε τάση 5V → LED ανάβει.
- `digitalWrite(pin, LOW);` → μηδενίζουμε την τάση → LED σβήνει.
- `delay(ms);` → καθυστέρηση σε χιλιοστά δευτερολέπτου (1000 ms = 1 sec).

Έλεγχος Λειτουργίας

Το LED πρέπει να αναβοσβήνει κάθε 1 δευτερόλεπτο.

Αν δεν λειτουργεί:

- ✓ Ελέγξτε αν η πολικότητα του LED είναι σωστή.
- ✓ Ελέγξτε αν έχετε συνδέσει σωστά το pin 10 και το GND.
- ✓ Ελέγξτε στο IDE αν επιλέξατε το σωστό Board και Port.

ΕΦΑΡΜΟΓΗ 2: Εναλλαγή δύο LED

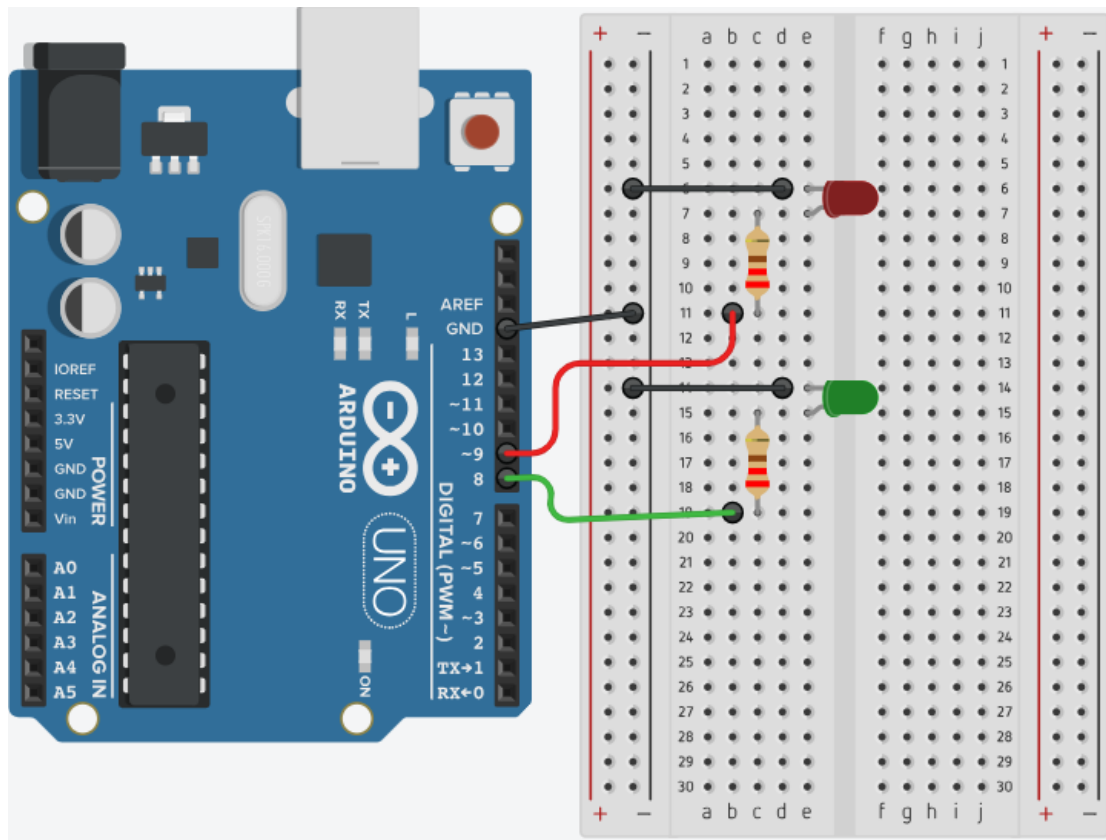
Θεωρία

Σε αυτή την εφαρμογή θα χρησιμοποιήσουμε δύο LED (π.χ. κόκκινο και πράσινο). Θα ανάβουν εναλλάξ, δηλαδή όταν το πρώτο είναι αναμμένο, το δεύτερο θα είναι σβηστό, και αντίστροφα. Έτσι καταλαβαίνουμε πώς μπορούμε να ελέγχουμε πολλαπλές εξόδους με το Arduino.

Υλικά

- 1 × Arduino Uno
- 2 × LED (π.χ. κόκκινο & πράσινο)
- 2 × Αντιστάσεις 220Ω
- Breadboard
- Καλώδια σύνδεσης

🔌 Σχέδιο συνδεσμολογίας



🔧 Βήματα Σύνδεσης

1. Τοποθετήστε τα δύο LED στο breadboard.
2. Συνδέστε την άνοδο (+) του κόκκινου LED στο pin 9 μέσω αντίστασης.
3. Συνδέστε την άνοδο (+) του πράσινου LED στο pin 8 μέσω αντίστασης.
4. Συνδέστε τις καθόδους (-) και των δύο LED στη γείωση (GND) του Arduino.

📌 Δηλαδή:

- Κόκκινο LED → pin 9
- Πράσινο LED → pin 8
- Και τα δύο LED συνδέονται στη γείωση (GND) με αντίσταση 220Ω.

★ Για να ξεχωρίζετε εύκολα τα LED, χρησιμοποιήστε διαφορετικά χρώματα.



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επέλεξε Arduino Uno.
4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επέλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.
7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περίμενε να εμφανιστεί το μήνυμα “Upload completed”.



Κώδικα

```
// Εφαρμογή 2: Εναλλαγή δύο LED

int led1 = 9;           // Το κόκκινο LED στο pin 9
int led2 = 8;           // Το πράσινο LED στο pin 8

void setup() {
  pinMode(led1, OUTPUT); // Ορίζουμε το pin 9 ως έξοδο
  pinMode(led2, OUTPUT); // Ορίζουμε το pin 8 ως έξοδο
}

void loop() {
  digitalWrite(led1, HIGH); // Ανάβει το κόκκινο LED
  digitalWrite(led2, LOW);  // Σβήνει το πράσινο LED
  delay(1000);              // Περιμένει 1 δευτερόλεπτο

  digitalWrite(led1, LOW);  // Σβήνει το κόκκινο LED
  digitalWrite(led2, HIGH); // Ανάβει το πράσινο LED
  delay(1000);              // Περιμένει 1 δευτερόλεπτο
}
```

Επεξήγηση εντολών:

- `int led1 = 9;` → δηλώνουμε μεταβλητή για το κόκκινο LED.
- `int led2 = 8;` → δηλώνουμε μεταβλητή για το πράσινο LED.
- `digitalWrite(led1, HIGH);` → ανάβει το κόκκινο LED.
- `digitalWrite(led2, HIGH);` → ανάβει το πράσινο LED.

Έλεγχος Λειτουργίας

- ✓ Το κόκκινο και το πράσινο LED πρέπει να ανάβουν εναλλάξ κάθε 1 δευτερόλεπτο.

Αν δεν λειτουργεί:

- ✓ Ελέγξτε τα pins (9 και 8).
- ✓ Ελέγξτε αν οι αντιστάσεις είναι σωστά τοποθετημένες.

ΕΦΑΡΜΟΓΗ 3: LED με κουμπί

Θεωρία

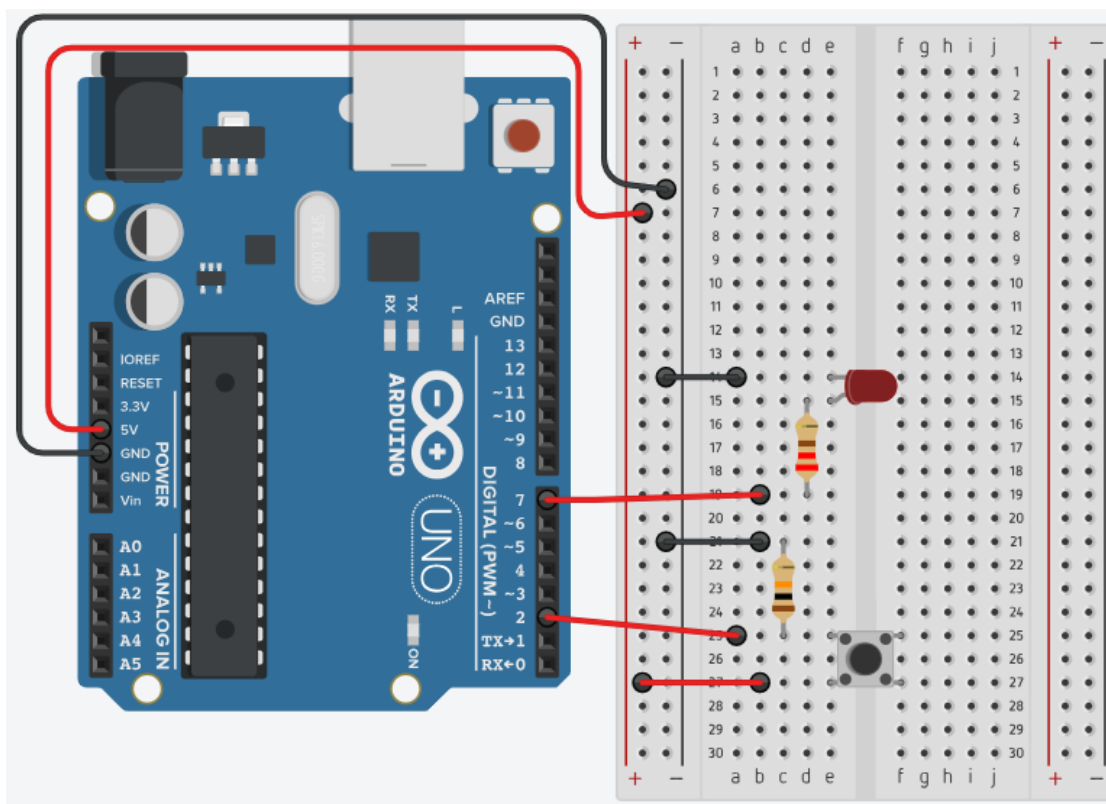
Το κουμπί (push button) λειτουργεί σαν διακόπτης:

- Όταν το πατάμε, το κύκλωμα κλείνει και στέλνει σήμα στο Arduino.
- Όταν το αφήνουμε, το κύκλωμα ανοίγει και η είσοδος γίνεται LOW.
- Θα συνδέσουμε ένα κουμπί στο pin 2 και θα ανάβουμε/σβήνουμε ένα LED στο pin 7.

Υλικά

- 1 × Arduino Uno
- 1 × LED (κόκκινο)
- 1 × Αντίσταση 220Ω (για το LED)
- 1 × Push button
- 1 × Αντίσταση 10kΩ (pull-down)
- Breadboard
- Καλώδια σύνδεσης

🔌 Σχέδιο συνδεσμολογίας



🔧 Βήματα Σύνδεσης

1. Συνδέστε το LED στο pin 7 μέσω αντίστασης 220Ω.
2. Το κουμπί τοποθετείται στο breadboard (κάθετα).
3. Ένα ποδαράκι του κουμπιού πάει στο pin 2.
4. Το ίδιο ποδαράκι πάει και στη γείωση (GND) μέσω αντίστασης 10kΩ (pull-down).
5. Το άλλο ποδαράκι πάει στο +5V.

☞ Δηλαδή:

LED → pin 7

Κουμπί → pin 2 (με αντίσταση 10kΩ προς GND)

★ Αν δεν βάλετε την pull-down αντίσταση, το Arduino μπορεί να «μπερδεύεται» και να νομίζει ότι πατάτε το κουμπί ενώ δεν το έχετε πατήσει.



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επέλεξε Arduino Uno.
4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επέλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.
7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περίμενε να εμφανιστεί το μήνυμα "Upload completed".



Κώδικας

```
// Εφαρμογή 3: LED με κουμπί

int led = 7;           // LED στο pin 7
int button = 2;        // Κουμπί στο pin 2
int buttonState = 0;   // Μεταβλητή για την κατάσταση του κουμπιού

void setup() {
  pinMode(led, OUTPUT);    // Το pin 7 ως έξοδος
  pinMode(button, INPUT);  // Το pin 2 ως είσοδος
}

void loop() {
  buttonState = digitalRead(button);    // Διαβάζουμε το κουμπί

  if (buttonState == HIGH) {           // Αν πατηθεί το κουμπί
    digitalWrite(led, HIGH);           // Ανάβει το LED
  } else {
    digitalWrite(led, LOW);            // Σβήνει το LED
  }
}
```


Επεξήγηση εντολών:

- `digitalRead(button);` → ελέγχει αν το κουμπί είναι πατημένο.
- `if (buttonState == HIGH)` → εκτελεί την εντολή μόνο όταν το κουμπί πατηθεί.

Έλεγχος Λειτουργίας

- ✓ Όταν πατάτε το κουμπί → LED ανάβει.
- ✓ Όταν το αφήνετε → LED σβήνει.

Μικρό Τεστ Ανατροφοδότησης

1. Σε ποιο pin συνδέσαμε το LED στην πρώτη εφαρμογή;
α) Pin 13
β) Pin 10
γ) Pin GND
δ) Pin 5V
2. Τι κάνει η εντολή `pinMode(10, OUTPUT);` ;
α) Ορίζει το pin 10 ως είσοδο
β) Ορίζει το pin 10 ως έξοδο
γ) Ανάβει το LED στο pin 10
δ) Δεν κάνει τίποτα
3. Τι λογικό σήμα (HIGH ή LOW) στέλνει το κουμπί στο Arduino όταν το πατάμε;
α) LOW
β) HIGH
γ) 0V
δ) Δεν στέλνει σήμα
4. Ποια εντολή χρησιμοποιούμε για να στείλουμε τάση σε ένα pin και να ανάψει το LED;
α) `digitalRead()`
β) `digitalWrite()`
γ) `delay()`
δ) `setup()`
5. Αν το LED δεν ανάβει, ποιο από τα παρακάτω ΔΕΝ είναι σωστή ενέργεια ελέγχου;
α) Να ελέγξω αν έβαλα σωστά την πολικότητα του LED
β) Να ελέγξω αν έχω φορτώσει σωστά τον κώδικα
γ) Να αλλάξω το χρώμα του LED
δ) Να ελέγξω αν το pin έχει οριστεί ως OUTPUT

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2

Φωτεινός Σηματοδότης Ελέγχου Κυκλοφορίας Οχημάτων και Πεζών

Στόχοι

Με την ολοκλήρωση του φύλλου οι μαθητές θα μπορούν:

- Να κατανοούν τη λειτουργία ενός φωτεινού σηματοδότη.
- Να σχεδιάζουν και να υλοποιούν κυκλώματα με περισσότερα από ένα LED.
- Να προσθέτουν κουμπί (push button) για έλεγχο κυκλώματος.
- Να γράφουν και να κατανοούν προγράμματα στο Arduino Uno που συνδυάζουν πολλαπλές εξόδους και μία είσοδο.
- Να εφαρμόζουν λογική ακολουθίας (sequences) σε πραγματικά κυκλώματα.

ΕΦΑΡΜΟΓΗ 1: Σηματοδότης Οχημάτων (3 LED)

Θεωρία

Ο φωτεινός σηματοδότης χρησιμοποιεί τρία χρώματα:

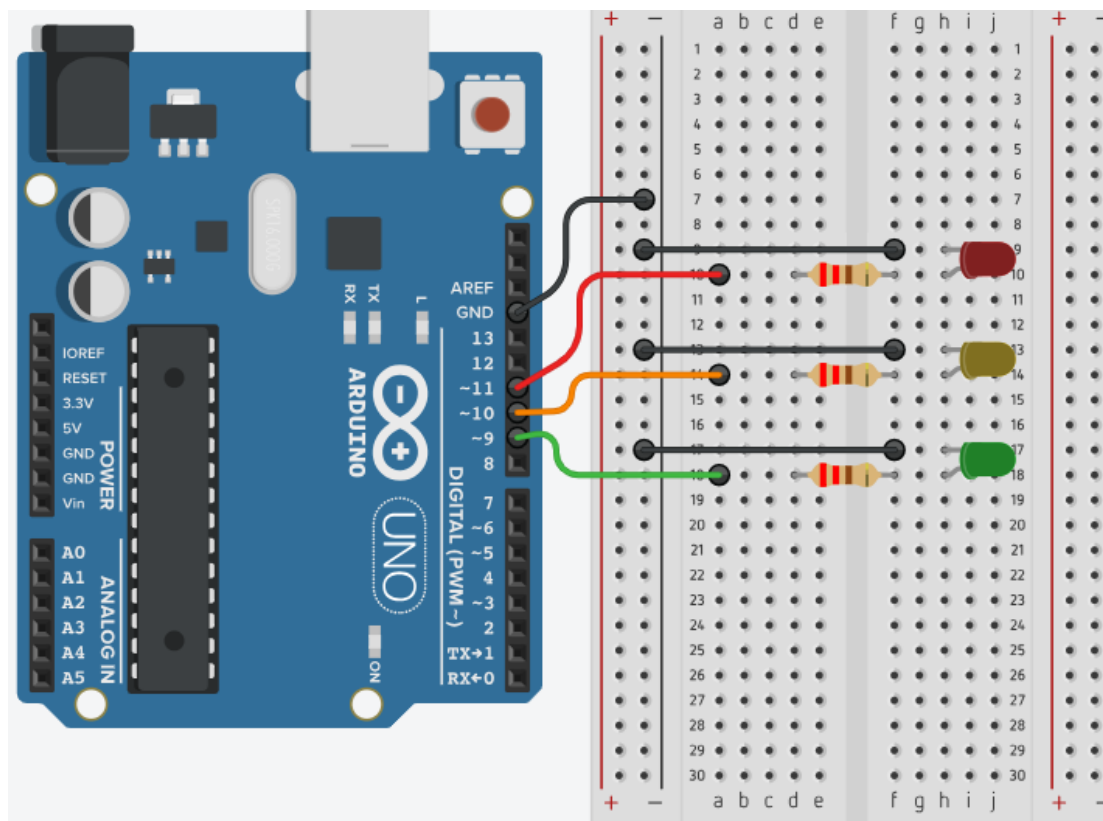
- Κόκκινο: Σταματούν τα οχήματα.
- Κίτρινο: Προειδοποίηση αλλαγής.
- Πράσινο: Κίνηση επιτρέπεται.

Στην εφαρμογή αυτή θα ανάβουν διαδοχικά 3 LED που αντιπροσωπεύουν τα φανάρια των οχημάτων.

Υλικά

- 1 Arduino Uno
- 1 Breadboard
- 3 LED (κόκκινο, κίτρινο, πράσινο)
- 3 αντιστάσεις 220Ω
- Καλώδια σύνδεσης (jumper wires)

🔧 Σχέδιο συνδεσμολογίας



🔧 Βήματα Σύνδεσης

1. Συνδέστε το κόκκινο LED στο Pin 11 με αντίσταση 220Ω προς GND.
2. Συνδέστε το κίτρινο LED στο Pin 10 με αντίσταση 220Ω προς GND.
3. Συνδέστε το πράσινο LED στο Pin 9 με αντίσταση 220Ω προς GND.
4. Ελέγξτε τις πολικότητες: το μακρύ ποδαράκι του LED πάει στην έξοδο του Arduino.

🔧 Δηλαδή:

Κόκκινο LED → Pin 11

Κίτρινο LED → Pin 10

Πράσινο LED → Pin 9

Κάθε LED συνδέεται σε σειρά με αντίσταση 220Ω και μετά στη γείωση (GND).

★ Αν τα LED δεν ανάβουν, πιθανότατα έχει αντιστραφεί η πολικότητα.



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επέλεξε Arduino Uno.
4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επέλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.
7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περίμενε να εμφανιστεί το μήνυμα “Upload completed”.



Κώδικας

```
// Εφαρμογή 1: Σηματοδότης Οχημάτων (3 LED)

// Ορισμός των pin για τα LED

int redCar = 11;      // Κόκκινο LED στο pin 11
int yellowCar = 10;   // Κίτρινο LED στο pin 10
int greenCar = 9;     // Πράσινο LED στο pin 9

void setup() {
    // Ορίζουμε τα pin ως εξόδους
    pinMode(redCar, OUTPUT);
    pinMode(yellowCar, OUTPUT);
    pinMode(greenCar, OUTPUT);
}

void loop() {
    // Πράσινο: τα αυτοκίνητα περνούν
    digitalWrite(greenCar, HIGH);    // Ανάβει πράσινο
    delay(4000);                     // 4 δευτερόλεπτα
```

```
digitalWrite(greenCar, LOW); // Σβήνει

// Κίτρινο: προειδοποίηση
digitalWrite(yellowCar, HIGH);
delay(2000);           // 2 δευτερόλεπτα
digitalWrite(yellowCar, LOW);

// Κόκκινο: τα αυτοκίνητα σταματούν
digitalWrite(redCar, HIGH);
delay(4000);           // 4 δευτερόλεπτα
digitalWrite(redCar, LOW);
}
```

Επεξήγηση εντολών:

- pinMode(..., OUTPUT) → δηλώνουμε ότι το pin θα δίνει σήμα.
- digitalWrite(pin, HIGH) → ανάβει LED.
- digitalWrite(pin, LOW) → σβήνει LED.
- delay(ms) → καθυστέρηση σε χιλιοστά του δευτερολέπτου.

Έλεγχος Λειτουργίας

- ✓ Ελέγξτε ότι ανάβει πρώτα το πράσινο, μετά το κίτρινο, και μετά το κόκκινο LED.
- ✓ Αν η σειρά είναι λάθος → αλλάξτε τα καλώδια ή ελέγξτε τον κώδικα.

◆ ΕΦΑΡΜΟΓΗ 2: Σηματοδότης Οχημάτων & Πεζών

📖 Θεωρία

Εκτός από τα οχήματα, πρέπει να ρυθμίζεται και η κυκλοφορία των πεζών.

Ένας δεύτερος μικρός σηματοδότης προστίθεται με κόκκινο και πράσινο LED:

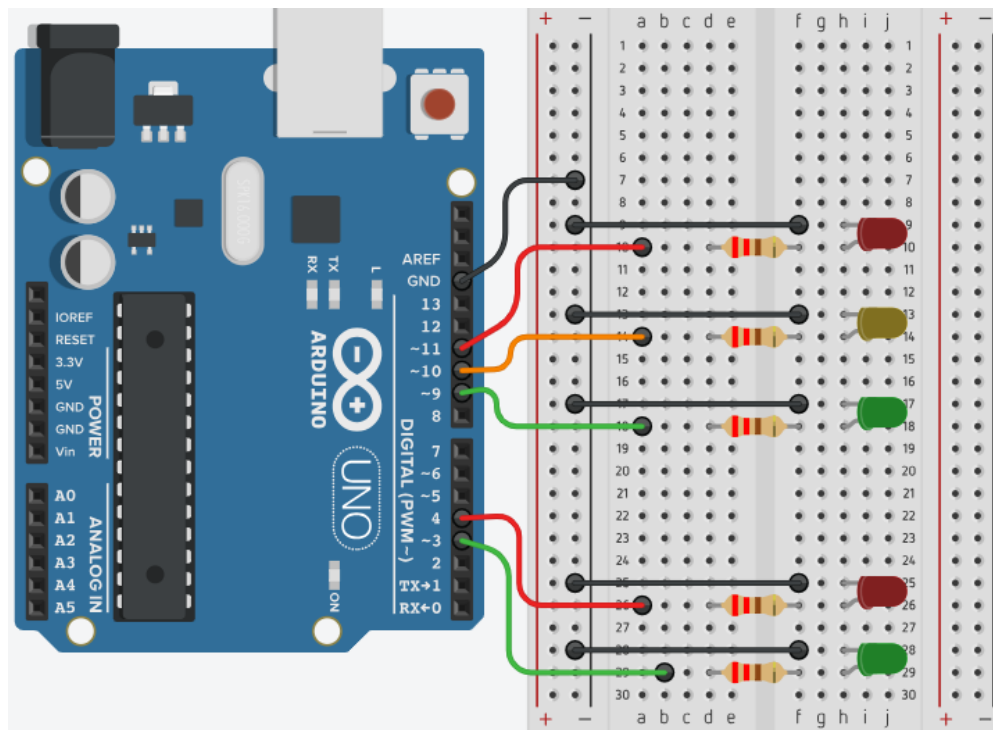
- Κόκκινο: Ο πεζός σταματά.
- Πράσινο: Ο πεζός περνά.

Στην εφαρμογή αυτή συγχρονίζουμε τα φανάρια ώστε να υπάρχει ασφαλής ρύθμιση και για τα δύο.

🔧 Υλικά

- 1 Arduino Uno
- 1 Breadboard
- 5 LED (3 για οχήματα, 2 για πεζούς)
- 5 αντιστάσεις 220Ω
- Καλώδια σύνδεσης

🔌 Σχέδιο συνδεσμολογίας





Βήματα Σύνδεσης

1. Συνδέστε τα 3 LED οχημάτων (11, 10, 9) όπως στην εφαρμογή 1.
2. Προσθέστε το κόκκινο LED πεζών στο Pin 4 με αντίσταση 220Ω προς GND.
3. Προσθέστε το πράσινο LED πεζών στο Pin 3 με αντίσταση 220Ω προς GND.
4. Ελέγξτε τις πολικότητες (το μακρύ ποδαράκι πάει στο pin).

☞ Δηλαδή:

Οχήματα:

- Κόκκινο → Pin 11
- Κίτρινο → Pin 10
- Πράσινο → Pin 9

Πεζοί:

- Κόκκινο → Pin 4
- Πράσινο → Pin 3



Για να είναι πιο ρεαλιστικό, όταν το κόκκινο των οχημάτων είναι αναμμένο, το πράσινο των πεζών πρέπει να ανάβει.



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επέλεξε Arduino Uno.
4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επέλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.
7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περίμενε να εμφανιστεί το μήνυμα "Upload completed".

Κώδικας

```
// Εφαρμογή 2: Σηματοδότης Οχημάτων & Πεζών
```

```
// Pin για οχήματα
```

```
int redCar = 11;
```

```
int yellowCar = 10;
```

```
int greenCar = 9;
```

```
// Pin για πεζούς
```

```
int redPed = 4;
```

```
int greenPed = 3;
```

```
void setup() {
```

```
  pinMode(redCar, OUTPUT);
```

```
  pinMode(yellowCar, OUTPUT);
```

```
  pinMode(greenCar, OUTPUT);
```

```
  pinMode(redPed, OUTPUT);
```

```
  pinMode(greenPed, OUTPUT);
```

```
}
```

```
void loop() {
```

```
  // 1. Οχήματα πράσινο - Πεζοί κόκκινο
```

```
  digitalWrite(greenCar, HIGH);
```

```
  digitalWrite(redPed, HIGH);
```

```
  delay(4000);
```

```
  digitalWrite(greenCar, LOW);
```

```
  digitalWrite(redPed, LOW);
```



```
// 2. Οχήματα κίτρινο
digitalWrite(yellowCar, HIGH);
delay(2000);
digitalWrite(yellowCar, LOW);

// 3. Οχήματα κόκκινο - Πεζοί πράσινο
digitalWrite(redCar, HIGH);
digitalWrite(greenPed, HIGH);
delay(4000);

digitalWrite(redCar, LOW);
digitalWrite(greenPed, LOW);
}
```

Επεξήγηση εντολών:

- Ο σηματοδότης πεζών είναι συγχρονισμένος αντίστροφα με των οχημάτων.
- Όταν το κόκκινο των οχημάτων είναι ON → το πράσινο των πεζών είναι ON.

Έλεγχος Λειτουργίας

- ✓ Ελέγξτε ότι όταν τα οχήματα έχουν πράσινο, οι πεζοί έχουν κόκκινο.
- ✓ Όταν τα οχήματα έχουν κόκκινο, οι πεζοί έχουν πράσινο.

◆ ΕΦΑΡΜΟΓΗ 3: Σηματοδότης με Κουμπί Προτεραιότητας Πεζών

Θεωρία

Σε πραγματικές διασταυρώσεις, οι πεζοί πατάνε ένα κουμπί για να ζητήσουν να περάσουν.

Το κουμπί (push button) είναι ένας διακόπτης στιγμιαίας επαφής:

- Όταν το πατάμε → το κύκλωμα κλείνει.
- Όταν το αφήνουμε → το κύκλωμα ανοίγει.

Στον Arduino χρησιμοποιούμε την ειδική εντολή INPUT_PULLUP ώστε να μην χρειάζεται εξωτερική αντίσταση.

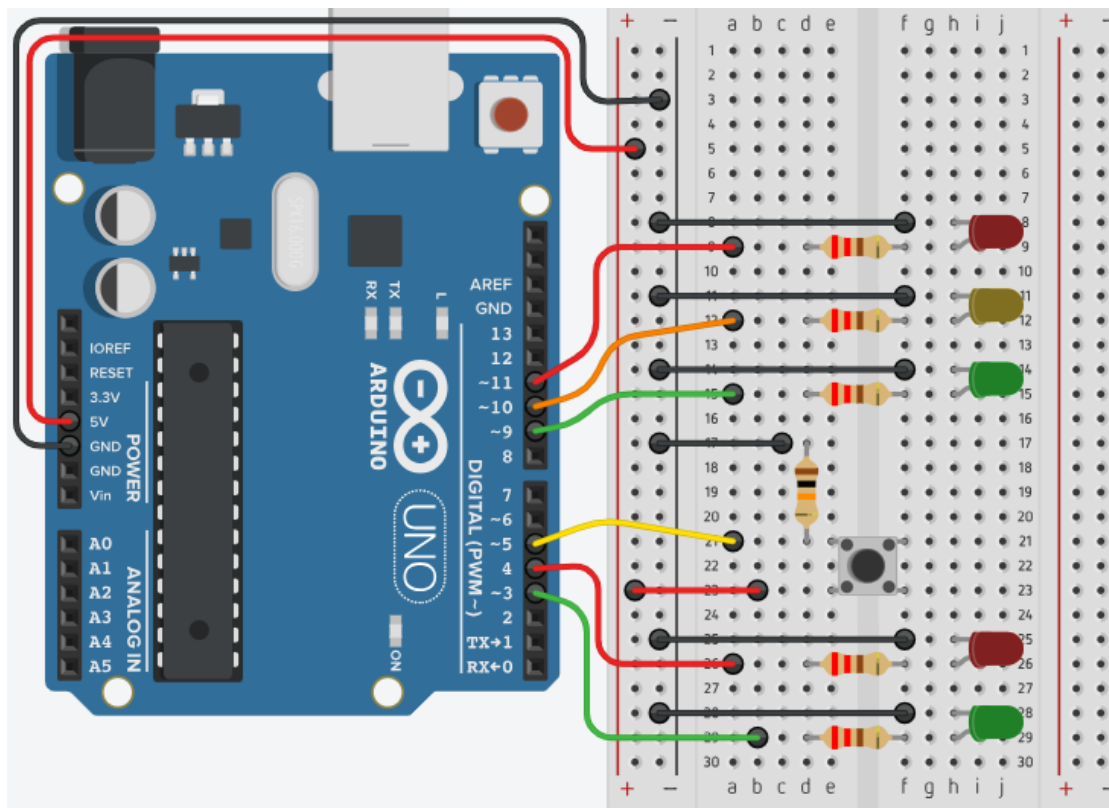
 Αυτό σημαίνει:

- Χωρίς πάτημα: το pin «βλέπει» HIGH (λογικό 1).
- Με πάτημα: το pin συνδέεται στη γείωση και «βλέπει» LOW (λογικό 0).

Υλικά

- 1 Arduino Uno
- 1 Breadboard
- 5 LED (3 για οχήματα, 2 για πεζούς)
- 5 αντιστάσεις 220Ω
- 1 αντίσταση 10ΚΩ
- 1 κουμπί (push button)
- Καλώδια σύνδεσης

🔌 Σχέδιο συνδεσμολογίας



🔧 Βήματα Σύνδεσης

1. Συνδέστε τα LED οχημάτων (11, 10, 9) όπως πριν.
2. Συνδέστε τα LED πεζών (4 και 3) όπως στην εφαρμογή 2.
3. Τοποθετήστε το κουμπί στο breadboard.
4. Ένα ποδαράκι του κουμπιού → Pin 5.
5. Το άλλο ποδαράκι → GND.
6. Συνδέστε το GND του breadboard με το GND του Arduino.

👉 Δηλαδή:

Οχήματα:

- Κόκκινο → Pin 11
- Κίτρινο → Pin 10
- Πράσινο → Pin 9

Πεζοί:

- Κόκκινο → Pin 4

- Πράσινο → Pin 3
- Κουμπί → Pin 5

✦ Το κουμπί μπαίνει κάθετα στο breadboard. Συνδέστε το ένα ποδαράκι στο Pin 5 και το άλλο στη γραμμή GND.



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επέλεξε Arduino Uno.
4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επέλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.
7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περίμενε να εμφανιστεί το μήνυμα “Upload completed”.



Κώδικας

```
// Εφαρμογή 3: Σηματοδότης με Κουμπί Προτεραιότητας Πεζών
```

```
// Φανάρια για αυτοκίνητα
```

```
int redCar = 11;
```

```
int yellowCar = 10;
```

```
int greenCar = 9;
```

```
// Φανάρια για πεζούς
```

```
int redPed = 4;
```

```
int greenPed = 3;
```

```
// Κουμπί για πεζούς
```

```
int buttonPin = 5;
```

```

    // Μεταβλητή για να θυμόμαστε αν υπάρχει αίτημα από πεζούς
    bool pedestrianRequest = false;

    void setup() {
        // Ορισμός pin ως είσοδοι/έξοδοι
        pinMode(redCar, OUTPUT);
        pinMode(yellowCar, OUTPUT);
        pinMode(greenCar, OUTPUT);
        pinMode(redPed, OUTPUT);
        pinMode(greenPed, OUTPUT);
        pinMode(buttonPin, INPUT_PULLUP); // κουμπί με pull-up (πατημένο = LOW)

        // Αρχική κατάσταση: αυτοκίνητα πράσινο, πεζοί κόκκινο
        digitalWrite(greenCar, HIGH);
        digitalWrite(redCar, LOW);
        digitalWrite(yellowCar, LOW);
        digitalWrite(redPed, HIGH);
        digitalWrite(greenPed, LOW);
    }

    void loop() {
        // Έλεγχος κουμπιού
        // Αν πατηθεί το κουμπί, καταγράφουμε το αίτημα (χωρίς να αλλάξουμε αμέσως τα φανάρια)
        if (digitalRead(buttonPin) == LOW) {
            pedestrianRequest = true;
        }
    }

```

```

// === ΚΑΝΟΝΙΚΟΣ ΚΥΚΛΟΣ ΦΑΝΑΡΙΩΝ ===

// 1) Αυτοκίνητα πράσινο, πεζοί κόκκινο
digitalWrite(greenCar, HIGH);
digitalWrite(redPed, HIGH);
delay(4000);           // 4 δευτερόλεπτα πράσινο στα αυτοκίνητα

// Αν έχει ζητηθεί διάβαση από πεζούς, μετά το πράσινο προχωράμε σε φάση αλλαγής
if (pedestrianRequest) {
    // Σβήνουμε πράσινο αυτοκινήτων
    digitalWrite(greenCar, LOW);

    // 2) Κίτρινο για αυτοκίνητα (προειδοποίηση)
    digitalWrite(yellowCar, HIGH);
    delay(2000);
    digitalWrite(yellowCar, LOW);

    // 3) Κόκκινο για αυτοκίνητα
    digitalWrite(redCar, HIGH);

    // 4) Πράσινο για πεζούς
    digitalWrite(redPed, LOW);
    digitalWrite(greenPed, HIGH);
    delay(5000); // 5 δευτερόλεπτα για να περάσουν οι πεζοί

    // 5) Επαναφορά: Πεζοί κόκκινο, αυτοκίνητα κόκκινο για λίγο (νεκρός χρόνος ασφαλείας)
    digitalWrite(greenPed, LOW);
    digitalWrite(redPed, HIGH);
    delay(2000);
}

```

```

        // 6) Επαναφορά: αυτοκίνητα πράσινο, πεζοί κόκκινο
        digitalWrite(redCar, LOW);
        digitalWrite(greenCar, HIGH);

        // Μηδενίζουμε το αίτημα
        pedestrianRequest = false;
    } else {
        // Αν ΔΕΝ υπάρχει αίτημα, συνεχίζουμε κανονικό κύκλο

        // Σβήνουμε πράσινο αυτοκινήτων
        digitalWrite(greenCar, LOW);
        digitalWrite(redPed, LOW);

        // Κίτρινο
        digitalWrite(yellowCar, HIGH);
        delay(2000);
        digitalWrite(yellowCar, LOW);

        // Κόκκινο για αυτοκίνητα, πράσινο για πεζούς
        digitalWrite(redCar, HIGH);
        digitalWrite(greenPed, HIGH);
        delay(4000);

        // Επαναφορά
        digitalWrite(redCar, LOW);
        digitalWrite(greenPed, LOW);
    }
}

```

Επεξήγηση εντολών:

- Το κουμπί δεν αλλάζει αμέσως το φανάρι· καταγράφεται σαν "αίτημα".
- Όταν τελειώσει το πράσινο των αυτοκινήτων, τότε γίνεται η μετάβαση (πράσινο → κίτρινο → κόκκινο).
- Μόνο τότε ανάβει πράσινο για τους πεζούς.
- Μετά από 5s, ξαναγυρνάει σε αυτοκίνητα πράσινο / πεζοί κόκκινο.

Έλεγχος Λειτουργίας

- ✓ Αν δεν πατηθεί το κουμπί, ο σηματοδότης λειτουργεί κανονικά (όπως στην εφαρμογή 2).
- ✓ Αν πατηθεί το κουμπί, τα οχήματα σταματούν (κόκκινο) και οι πεζοί περνούν (πράσινο).
- ✓ Μετά από λίγο, όλα επανέρχονται στον κανονικό κύκλο.



Μικρό Τεστ Ανατροφοδότησης

Επιλέξτε τη σωστή απάντηση:

1. Ποια LED χρησιμοποιούμε για τον έλεγχο των οχημάτων στην εφαρμογή 1;
α) Κόκκινο, Κίτρινο, Πράσινο
β) Κόκκινο, Πράσινο
γ) Κόκκινο, Μπλε
δ) Κίτρινο, Μπλε

2. Σε ποια pin συνδέεται ο φωτεινός σηματοδότης των πεζών στην εφαρμογή 2;
α) Pin 11 και Pin 10
β) Pin 9 και Pin 8
γ) Pin 4 και Pin 3
δ) Pin 6 και Pin 5

3. Τι κάνει η εντολή INPUT_PULLUP όταν χρησιμοποιείται για το κουμπί;
α) Ανάβει το LED αυτόματα
β) Δηλώνει ότι το pin είναι έξοδος
γ) Κρατά το pin σε HIGH αν δεν πατηθεί το κουμπί
δ) Δίνει τάση 5V στο κουμπί

4. Τι πρέπει να συμβεί όταν ο πεζός πατήσει το κουμπί στην εφαρμογή 3;
α) Τα οχήματα συνεχίζουν κανονικά
β) Τα οχήματα σταματούν και οι πεζοί περνούν
γ) Σβήνουν όλα τα LED
δ) Αναβοσβήνει το κίτρινο φανάρι

5. Γιατί χρησιμοποιούμε καθυστέρηση (delay) στο πρόγραμμα;
α) Για να αλλάζει το χρώμα των LED σε σωστούς χρόνους
β) Για να δηλώσουμε ποια pin είναι έξοδοι
γ) Για να λειτουργεί το κουμπί
δ) Για να τροφοδοτείται το κύκλωμα με 5V

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 3

ΤΡΕΧΑΝΤΗΡΙ ΜΕ LED

Στόχοι

- Να κατανοήσουν οι μαθητές/τριες τη χρήση πινάκων (arrays) για χειρισμό πολλαπλών εξόδων.
- Να γράφουν βρόχους (for) για επαναλαμβανόμενες ενέργειες.
- Να δημιουργούν εφέ κίνησης με LED (δεξιά ολίσθηση, αριστερή ολίσθηση, πήγαινε-έλα).
- Να ελέγχουν συνδέσεις σε breadboard και να διευθετούν προβλήματα λειτουργίας.

ΕΦΑΡΜΟΓΗ 1: ΔΕΞΙΑ ΟΛΙΣΘΗΣΗ

Θεωρία

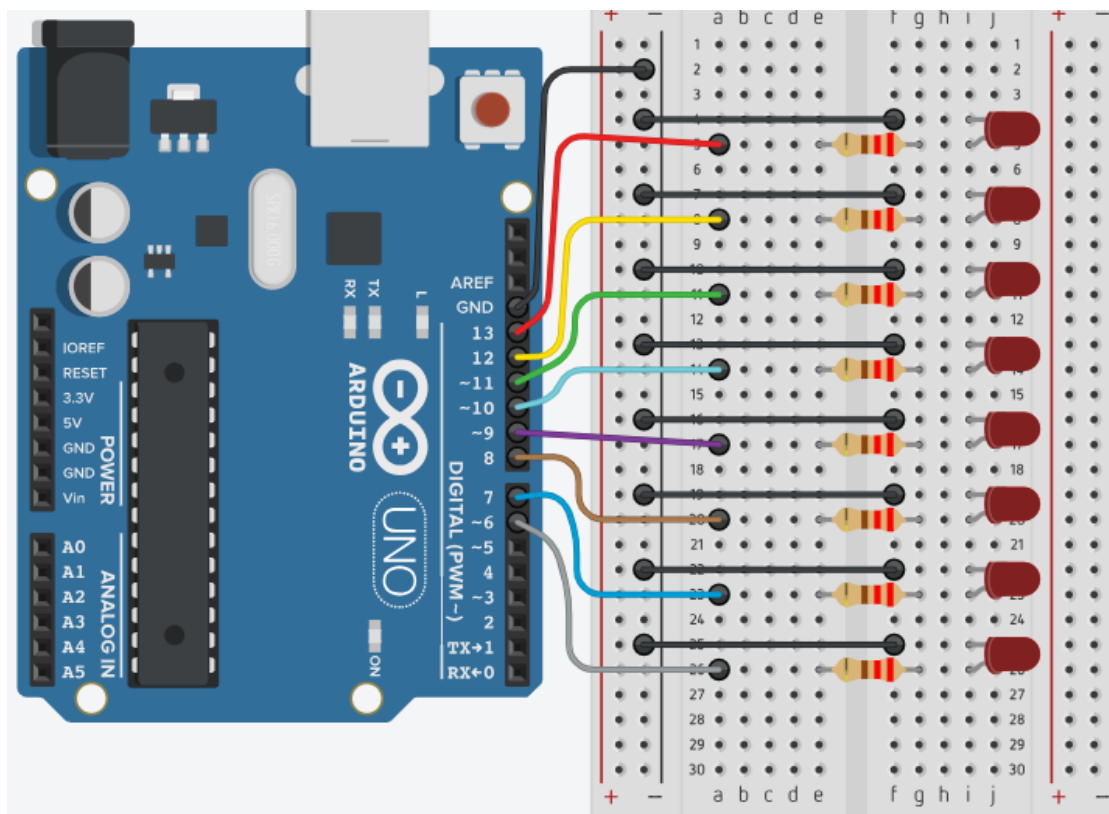
Το τρεχάτηρι (running light) ανάβει διαδοχικά LED ώστε να δίνει την αίσθηση μιας κινούμενης φωτεινής κουκκίδας. Εδώ η κίνηση είναι από αριστερά προς τα δεξιά (pin 13 προς pin 6).

Υλικά

- 1 × Arduino Uno
- 1 × Breadboard
- 8 × LED (π.χ. διάφορα χρώματα)
- 8 × Αντιστάσεις 220Ω
- Jumper wires (καλώδια σύνδεσης)
- Καλώδιο USB για σύνδεση Arduino ↔ υπολογιστή

★ Προτεινόμενη σειρά pin: το πρώτο LED στο Pin 13, τα επόμενα στα 12, 11, 10, 9, 8, 7, 6 (13 → 6).

🔌 Σχέδιο συνδεσμολογίας



🔧 Βήματα Σύνδεσης

1. Τοποθετήστε τα 8 LED στο breadboard σε μία σειρά.
2. Συνδέστε την άνοδο (+, μακρύ ποδαράκι) του πρώτου LED σε μία τρύπα της σειράς.
3. Βάλτε σε σειρά την αντίσταση 220Ω μεταξύ της άνοδου και του jumper που πάει στο pin 13.
4. Επαναλάβετε για τα υπόλοιπα LED με pins 12, 11, 10, 9, 8, 7, 6.
5. Συνδέστε όλες τις καθόδους (-, κοντό ποδαράκι) μαζί στη γραμμή GND του breadboard.
6. Συνδέστε τη GND γραμμή του breadboard στο GND του Arduino.
7. Έλεγξε εξωτερικά τις συνδέσεις πριν τροφοδοτήσεις.

📌 Δηλαδή:

- LED1 (αριστερό) → Pin 13 → αντίσταση 220Ω → LED άνοδος → κάθοδος → GND
- LED2 → Pin 12 → αντίσταση → ... → GND
- ...
- LED8 (δεξί) → Pin 6 → αντίσταση → ... → GND

(Κάθε LED έχει τη δική του αντίσταση σε σειρά. Γείωση (GND) του breadboard συνδέεται με GND του Arduino.)

★ Το LED έχει πολικότητα — μακρύ = +, κοντό = -. Αν ένα LED δεν ανάβει, γύρισέ το ανάποδα.



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επέλεξε Arduino Uno.
4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επέλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.
7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περίμενε να εμφανιστεί το μήνυμα “Upload completed”.



Κώδικας

```
// Εφαρμογή 1: Δεξιά ολίσθηση (13 -> 6)

int leds[] = {13, 12, 11, 10, 9, 8, 7, 6}; // Πίνακας με pins των 8 LED
int n = 8; // Πλήθος LED
int delayTime = 200; // Χρόνος ανάβ/σβηγ σε ms (ρυθμίζεται)

void setup() {
  for (int i = 0; i < n; i++) {
    pinMode(leds[i], OUTPUT); // Κάθε pin ορίζεται ως έξοδος
  }
}

void loop() {
  for (int i = 0; i < n; i++) {
    digitalWrite(leds[i], HIGH); // Ανάβει το LED στη θέση i
    delay(delayTime); // Περιμένει delayTime ms
    digitalWrite(leds[i], LOW); // Σβήνει το LED πριν προχωρήσει
  }
}
```

Επεξήγηση εντολών:

- `int leds[] = {...};` — ο πίνακας (array) κρατάει τα pin των LED, ώστε να τα διαχειριζόμαστε εύκολα με βρόχο.
- `pinMode(pin, OUTPUT);` — ορίζει το pin ως έξοδο (θα στέλνει τάση στο LED).
- `digitalWrite(pin, HIGH);` — στέλνει 5V στο pin (ανάβει LED).
- `digitalWrite(pin, LOW);` — σβήνει το pin (0V).
- `delay(ms);` — σταματά το πρόγραμμα για ms χιλιοστά του δευτερολέπτου.

Έλεγχος Λειτουργίας

- ✓ Τα LED πρέπει να ανάβουν ένα-ένα από το pin 13 προς το pin 6 με καθυστέρηση `delayTime`.
- ✓ Αν κάποιο LED δεν ανάβει, ελέγξτε την πολικότητα και την αντίσταση.
- ✓ Δοκιμάστε να αλλάξετε `delayTime` (π.χ. 100 ms ή 400 ms) και δείτε την αλλαγή ταχύτητας.

ΕΦΑΡΜΟΓΗ 2: ΑΡΙΣΤΕΡΗ ΟΛΙΣΘΗΣΗ

Θεωρία

Ίδιο κύκλωμα με την Εφαρμογή 1, αλλά η κατεύθυνση αντιστρέφεται: τα LED ανάβουν από το δεξί άκρο (pin 6) προς το αριστερό (pin 13).

Υλικά

(ίδια με Εφαρμογή 1 — δεν χρειάζεστε παραπάνω εξαρτήματα)

Σχέδιο συνδεσμολογίας

(ίδιο με Εφαρμογή 1 — τα ίδια pins 13..6 χρησιμοποιούνται, απλώς η σειρά στο πρόγραμμα αλλάζει)

Βήματα Σύνδεσης

(ίδια όπως στην Εφαρμογή 1 — ελέγξτε ότι οι συνδέσεις είναι όπως πριν)



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επέλεξε Arduino Uno.
4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επέλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.
7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περίμενε να εμφανιστεί το μήνυμα “Upload completed”.



Κώδικας

//Εφαρμογή 2: Αριστερή ολίσθηση (6 -> 13)

```
int leds[] = {13, 12, 11, 10, 9, 8, 7, 6};
```

```
int n = 8;
```

```
int delayTime = 200;
```

```
void setup() {
```

```
  for (int i = 0; i < n; i++) {
```

```
    pinMode(leds[i], OUTPUT);
```

```
  }
```

```
}
```

```
void loop() {
```

```
  // Ξεκινάμε από το τελευταίο στοιχείο (index n-1) και πηγαίνουμε προς το 0
```

```
  for (int i = n - 1; i >= 0; i--) {
```

```
    digitalWrite(leds[i], HIGH);    // Ανάβει το LED
```

```
    delay(delayTime);              // Περιμένει
```

```
    digitalWrite(leds[i], LOW);    // Σβήνει
```

```
  }
```

```
}
```

Επεξήγηση εντολών:

Σημείωση για τον βρόχο for:

Ο βρόχος for (int i = n-1; i >= 0; i--) ξεκινάει από i = 7 και μειώνει μέχρι i = 0. Όταν i γίνει -1, η συνθήκη i >= 0 αποτυγχάνει και ο βρόχος τερματίζει — αυτό είναι σωστό και αναμενόμενο.

Έλεγχος Λειτουργίας

- ✓ Τα LED πρέπει να ανάβουν από το pin 6 προς το pin 13 (δεξιά → αριστερά).
- ✓ Δοκιμάστε διάφορα delayTime για να δείτε την επίδραση στην ταχύτητα.

ΕΦΑΡΜΟΓΗ 3: «ΠΗΓΑΙΝΕ-ΕΛΑ» (Ping-Pong)

Θεωρία

Το πήγαινε-έλα συνδυάζει τις δύο προηγούμενες ολισθήσεις σε μια συνεχή κίνηση: τα LED «τρέχουν» δεξιά και μετά επιστρέφουν αριστερά, δημιουργώντας ομαλό εφέ. Σχεδιάζουμε τον κώδικα ώστε να μην ξανα-αναβοσβήνει δύο φορές το άκρο LED όταν αλλάζει κατεύθυνση για πιο ομαλό αποτέλεσμα.

Υλικά

(ίδια με Εφαρμογή 1)

Σχέδιο συνδεσμολογίας

(ίδια με Εφαρμογή 1 — χρησιμοποιούμε τα ίδια pins)

Βήματα Σύνδεσης

(ίδια με Εφαρμογή 1 — χρησιμοποιούμε τα ίδια pins)



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επέλεξε Arduino Uno.
4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επέλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.
7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περίμενε να εμφανιστεί το μήνυμα “Upload completed”.



Κώδικας

```
//Εφαρμογή 3: Πήγαινε-Έλα (ping-pong)

int leds[] = {13, 12, 11, 10, 9, 8, 7, 6};
int n = 8;
int delayTime = 200;

void setup() {
  for (int i = 0; i < n; i++) {
    pinMode(leds[i], OUTPUT);
  }
}

void loop() {
  // Πρώτα πηγαίνουμε αριστερά -> δεξιά (index 0 -> n-1)
  for (int i = 0; i < n; i++) {
    digitalWrite(leds[i], HIGH);
    delay(delayTime);
    digitalWrite(leds[i], LOW);
  }
```



```

// Τώρα επιστρέφουμε δεξιά -> αριστερά αλλά ΑΠΟΦΕΥΓΟΥΜΕ
// να ενεργοποιήσουμε ξανά το ακραίο LED που μόλις σβήσαμε.
// Γι' αυτό ξεκινάμε από n-2 (index 6) και φτάνουμε στο 1.
for (int i = n - 2; i >= 1; i--) {
    digitalWrite(leds[i], HIGH);
    delay(delayTime);
    digitalWrite(leds[i], LOW);
}
}

```

Επεξήγηση εντολών:

Γιατί n-2 μέχρι 1 στο δεύτερο βρόχο;

Τελειώνοντας τον πρώτο βρόχο, το LED στο index n-1 (pin 6) έχει ήδη αναβεί και σβήσει. Αν επιστρέψουμε άμεσα από n-1 πάλι, θα γίνει διπλό «κλικ» στο άκρο. Ξεκινώντας από n-2 πετυχαίνουμε ομαλή επιστροφή χωρίς διπλό-αναβόσβημα. Το ίδιο αποφεύγουμε να φτάσουμε μέχρι 0 στην επιστροφή (εδώ επιλέξαμε ≥ 1) ώστε η κίνηση να έχει ομαλό ρυθμό — αν προτιμάτε πιο πλήρη επιστροφή, αλλάξτε το σε $i \geq 0$.

Έλεγχος Λειτουργίας

- ✓ Το φως πρέπει να φαίνεται να κινείται αριστερά→δεξιά και μετά ομαλά πίσω (δεξιά→αριστερά) χωρίς διπλά ακραία flashes.
- ✓ Αν το εφέ δεν είναι ομαλό, δοκιμάστε:
 - Να μειώσετε delayTime (γρηγορότερο εφέ) ή να το αυξήσετε (πιο αργό).
 - Να αλλάξετε τα όρια του δεύτερου βρόχου (π.χ. $i \geq 0$) αν θέλετε το άκρο να ανάβει και στην επιστροφή.



Μικρό Τεστ Ανατροφοδότησης

1. Τι είναι ο πίνακας `int leds[] = {13,12,...};` ;
α) Μια λίστα με pin που χρησιμοποιούμε.
β) Ένα είδος LED.
γ) Μια συνάρτηση του Arduino.
2. Στην Εφαρμογή 1, από ποιο pin ξεκινάει η ολίσθηση;
α) Pin 6
β) Pin 13
γ) Pin 8
3. Τι κάνει η εντολή `digitalWrite(leds[i], HIGH);` ;
α) Σβήνει το LED
β) Ανάβει το LED (στέλνει 5V)
γ) Αλλάζει το `delayTime`
4. Στην Εφαρμογή 3 γιατί ξεκινάμε την επιστροφή από n-2 αντί για n-1;
α) Για να αποφύγουμε διπλό ανάβω στο άκρο LED και να έχουμε πιο ομαλό εφέ.
β) Επειδή το n-1 δεν είναι έγκυρος δείκτης.
γ) Για να κάνει το πρόγραμμα `compile`.
5. Τι θα συμβεί αν αφαιρέσουμε τις εντολές `digitalWrite(leds[i], LOW);` ;
α) Τα LED θα μείνουν αναμμένα μετά που άναψαν.
β) Τα LED δεν θα λειτουργήσουν.
γ) Το `delayTime` θα μηδενιστεί.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 4

ΡΥΘΜΙΣΗ ΕΝΤΑΣΗΣ ΦΩΤΕΙΝΟΤΗΤΑΣ

Στόχοι

Μετά την ολοκλήρωση του φύλλου, οι μαθητές/τριες θα μπορούν:

- Να κατανοούν τη λειτουργία του ποτενσιόμετρου ως μεταβλητή αντίσταση/διαιρέτης τάσης.
- Να χρησιμοποιούν τις αναλογικές εισόδους του Arduino για μέτρηση τάσης.
- Να εξάγουν δεδομένα στη σειριακή οθόνη.
- Να ελέγχουν την ένταση φωτεινότητας ενός LED με τη βοήθεια ποτενσιόμετρου (PWM).
- Να σχεδιάζουν κυκλώματα με ποτενσιόμετρα και LED.
- Να υλοποιούν προγράμματα που συνδέουν είσοδο-έξοδο (αναλογική μέτρηση – αναλογική έξοδος).

◆ ΕΦΑΡΜΟΓΗ 1: Μέτρηση τάσης με τη βοήθεια ποτενσιόμετρου και χρήση της σειριακής οθόνης

Θεωρία

Το ποτενσιόμετρο είναι ένας ρυθμιζόμενος διαιρέτης τάσης. Διαθέτει τρεις ακροδέκτες:

- Ο ένας άκρο συνδέεται στο +5V,
- Ο άλλος στη Γείωση (GND),
- Ο μεσαίος είναι η κινητή επαφή (wiper) που δίνει την τάση εξόδου.

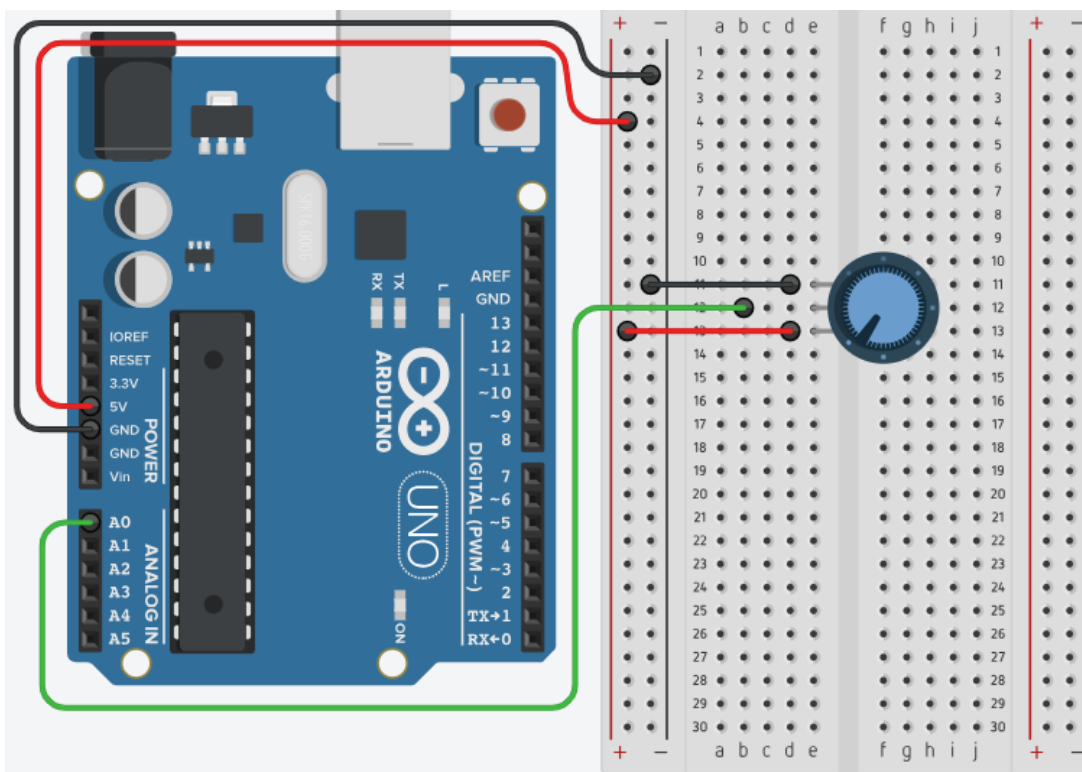
Η έξοδος αυτή μπορεί να πάρει οποιαδήποτε τιμή από 0V έως 5V.

Το Arduino διαθέτει αναλογικές εισόδους (A0–A5) που μπορούν να μετρούν αυτή την τάση. Η εντολή `analogRead()` επιστρέφει τιμές 0–1023, οι οποίες αντιστοιχούν σε 0–5V.

Υλικά

- 1 × Arduino Uno
- 1 × Ποτενσιόμετρο 10kΩ
- Καλώδια σύνδεσης
- Breadboard

🔌 Σχέδιο συνδεσμολογίας



🔧 Βήματα Σύνδεσης

1. Συνδέστε το δεξί άκρο του ποτενσιόμετρου στο +5V του Arduino.
2. Συνδέστε το αριστερό άκρο στο GND.
3. Συνδέστε τον μεσαίο ακροδέκτη στο pin A0 του Arduino.



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Ανοίξτε το λογισμικό Arduino IDE.
2. Δημιουργήστε νέο αρχείο.
3. Αντιγράψτε τον κώδικα.
4. Επιλέξτε από το μενού Εργαλεία → Πλακέτα → Arduino Uno.
5. Επιλέξτε τη σωστή θύρα (COM).
6. Πατήστε Μεταφόρτωση.
7. Ανοίξτε τη Σειριακή Οθόνη για να δείτε τις τιμές.

Κώδικας

// ΕΦΑΡΜΟΓΗ 1: Μέτρηση τάσης με τη βοήθεια ποτενσιόμετρου και χρήση της σειριακής οθόνης

```
int potPin = A0;           // δηλώνουμε το pin του ποτενσιόμετρου
int potValue = 0;          // μεταβλητή για την αποθήκευση της μέτρησης

void setup() {
  Serial.begin(9600);       // ξεκινά η σειριακή επικοινωνία στα 9600 baud
}

void loop() {
  potValue = analogRead(potPin); // διαβάζουμε την τιμή (0-1023) από το A0
  Serial.println(potValue);      // εμφανίζουμε τη μέτρηση στη σειριακή οθόνη
  delay(200);                   // μικρή καθυστέρηση για σταθερή ανάγνωση
}
```

Έλεγχος Λειτουργίας

- ✓ Περιστρέψτε το ποτενσιόμετρο.
- ✓ Στη σειριακή οθόνη πρέπει να βλέπετε τιμές από περίπου 0 έως 1023.

◆ ΕΦΑΡΜΟΓΗ 2: Κύκλωμα ελέγχου φωτεινότητας LED, με τη βοήθεια ποτενσιόμετρου

Θεωρία

Το Arduino δεν μπορεί να δώσει πραγματική αναλογική έξοδο, αλλά χρησιμοποιεί την τεχνική PWM (Pulse Width Modulation).

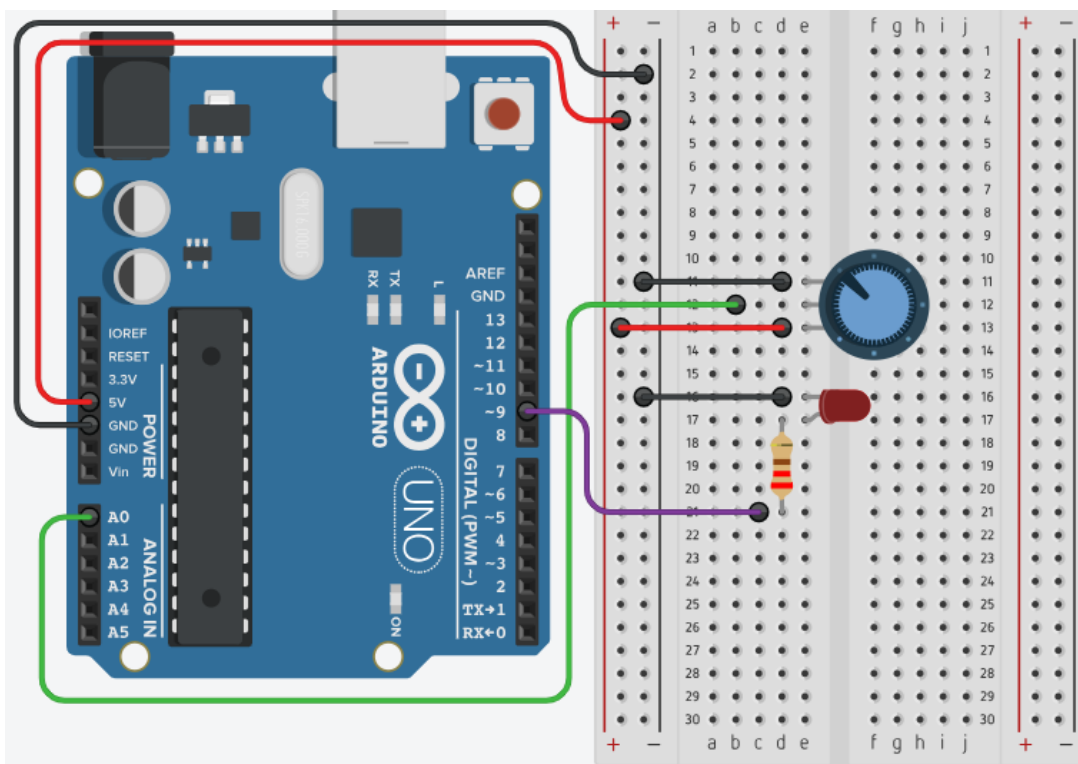
Με την εντολή `analogWrite()` παράγεται σήμα με παλμούς, του οποίου η διάρκεια (duty cycle) ελέγχει τη φωτεινότητα του LED.

- `analogRead()` → διαβάζει τιμή 0–1023 (είσοδος)
- `map()` → μετατρέπει την τιμή αυτή σε νέα κλίμακα (π.χ. 0–255)
- `analogWrite()` → στέλνει την τιμή PWM στην έξοδο (π.χ. pin 9)

Υλικά

- 1 × Arduino Uno
- 1 × Ποτενσιόμετρο 10kΩ
- 1 × LED
- 1 × Αντίσταση 220Ω
- Καλώδια σύνδεσης
- Breadboard

Σχέδιο συνδεσμολογίας



Βήματα Σύνδεσης

1. Συνδέστε το ποτενσιόμετρο όπως στην Εφαρμογή 1 (μεσαίος ακροδέκτης → A0).
2. Συνδέστε την άνοδο του LED στο pin 9 (μέσω αντίστασης 220Ω).
3. Συνδέστε την κάθοδο του LED στη γείωση (GND).



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επέλεξε Arduino Uno.
4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επέλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.
7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περίμενε να εμφανιστεί το μήνυμα “Upload completed”.



Κώδικας

// Εφαρμογή 2: Κύκλωμα ελέγχου φωτεινότητας LED, με τη βοήθεια ποτενσιόμετρου

```
int potPin = A0;           // είσοδος ποτενσιόμετρου
int ledPin = 9;            // έξοδος LED με PWM
int potValue = 0;          // αποθήκευση τιμής ποτενσιόμετρου
int ledValue = 0;          // αποθήκευση τιμής για το LED

void setup() {
  pinMode(ledPin, OUTPUT); // δηλώνουμε το pin 9 ως έξοδο
}

void loop() {
  potValue = analogRead(potPin); // διαβάζουμε τιμή (0-1023)
  ledValue = map(potValue, 0, 1023, 0, 255); // μετατροπή στην κλίμακα 0-255
  analogWrite(ledPin, ledValue); // αποστολή PWM στο LED
  delay(50);                     // μικρή καθυστέρηση για σταθερότητα
}
```

Έλεγχος Λειτουργίας

- ✓ Περιστρέψτε το ποτενσιόμετρο.
- ✓ Το LED πρέπει να αλλάζει ομαλά φωτεινότητα από σβηστό έως μέγιστη ένταση.

Μικρό Τεστ Ανατροφοδότησης

Επιλέξτε τη σωστή απάντηση (μία για κάθε ερώτηση):

1. Τι τιμές επιστρέφει η εντολή `analogRead()`; ;
 - α. 0–5V
 - β. 0–1023
 - γ. 0–255
 - δ. Μόνο 0 ή 1
2. Ποια εντολή χρησιμοποιείται για να στείλουμε σήμα PWM σε LED;
 - α. `digitalWrite()`
 - β. `analogWrite()`
 - γ. `Serial.println()`
 - δ. `pinMode()`
3. Το ποτενσιόμετρο λειτουργεί ως:
 - α. Σταθερή αντίσταση
 - β. Διαρέτης τάσης
 - γ. Πηγή ρεύματος
 - δ. Διακόπτης
4. Σε ποια κλίμακα μετατρέπουμε συνήθως την τιμή του ποτενσιόμετρου για PWM;
 - α. 0–5
 - β. 0–1023
 - γ. 0–255
 - δ. 0–100
5. Αν περιστρέψουμε το ποτενσιόμετρο προς το +5V, το LED:
 - α. Σβήνει
 - β. Αυξάνει τη φωτεινότητά του
 - γ. Μειώνει τη φωτεινότητά του
 - δ. Δεν αλλάζει

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 5

ΑΝΙΧΝΕΥΤΗΣ ΣΚΟΤΟΥΣ ΜΕ ΦΩΤΟΑΝΤΙΣΤΑΣΗ

Στόχοι

Μετά την ολοκλήρωση του φύλλου, οι μαθητές/τριες θα μπορούν:

- Να κατανοούν τη λειτουργία της φωτοαντίστασης (LDR).
- Να χρησιμοποιούν αναλογικές εισόδους του Arduino για την ανίχνευση φωτός/σκότους.
- Να σχεδιάζουν κυκλώματα που ενεργοποιούν εξόδους (π.χ. LED) βάσει εισόδων (π.χ. LDR).
- Να υλοποιούν και να ελέγχουν πρόγραμμα σε Arduino με συνθήκες (if – else).

ΕΦΑΡΜΟΓΗ : Σε συνθήκες σκότους ανάβει ένα LED, ενώ σε συνθήκες φωτός το LED παραμένει σβηστό.

Θεωρία

Η φωτοαντίσταση (Light Dependent Resistor – LDR) είναι ένας αισθητήρας που αλλάζει την τιμή της αντίστασής του ανάλογα με το φως:

- Σε έντονο φωτισμό → μικρή αντίσταση → μεγαλύτερη τάση στην είσοδο.
- Σε σκοτάδι → μεγάλη αντίσταση → μικρότερη τάση στην είσοδο.

Με τον συνδυασμό LDR + αντίστασης 10kΩ σχηματίζεται διαιρέτης τάσης, ο οποίος στέλνει στο Arduino μια μεταβλητή τάση ανάλογη με την ένταση του φωτός.

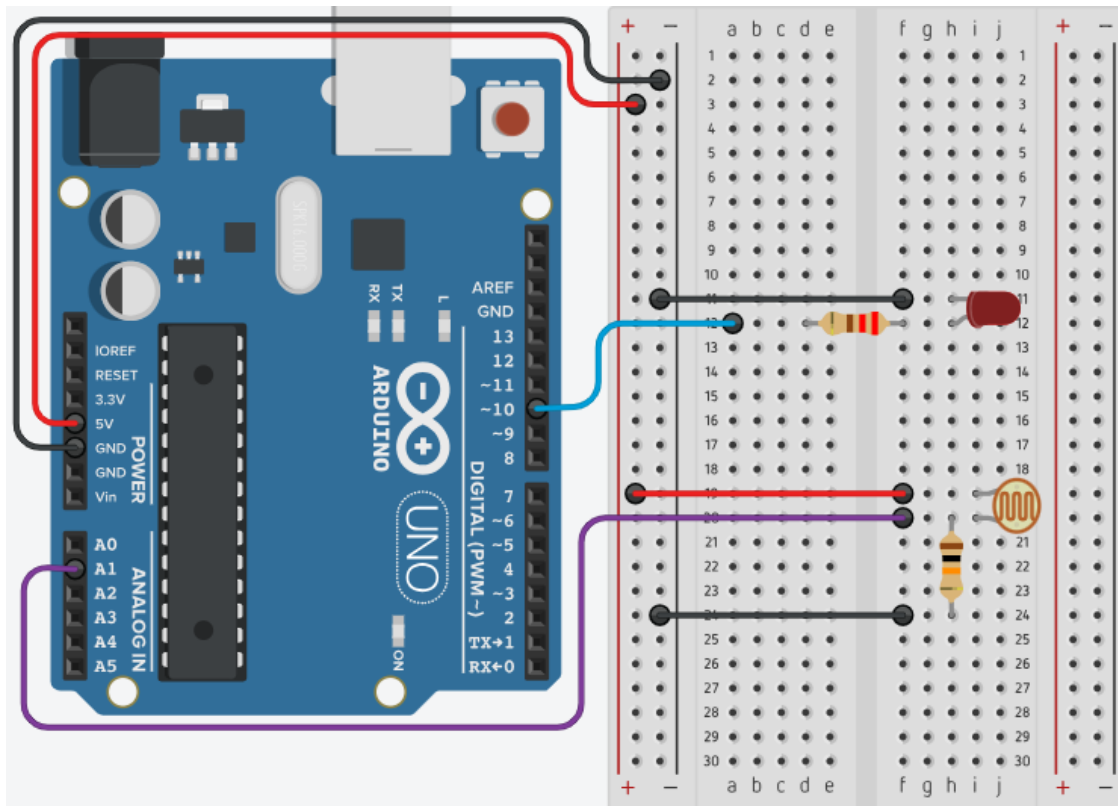
Το Arduino «διαβάζει» αυτήν την τάση μέσω της αναλογικής εισόδου (A1).

Υλικά

- 1 × Arduino Uno
- 1 × Φωτοαντίσταση (LDR)
- 1 × Αντίσταση 10kΩ (για τον διαιρέτη τάσης)
- 1 × LED

- 1 × Αντίσταση 220Ω
- Καλώδια σύνδεσης
- Breadboard

🔌 Σχέδιο συνδεσμολογίας



🔧 Βήματα Σύνδεσης

1. Συνδέστε το LED στο pin 10 μέσω αντίστασης 220Ω προς GND.
2. Συνδέστε τη φωτοαντίσταση:
 - Το ένα άκρο στο +5V.
 - Το άλλο άκρο στον ακροδέκτη A1 και στην αντίσταση 10kΩ.
 - Το άλλο άκρο της αντίστασης 10kΩ στο GND.

✦ Έτσι δημιουργείται ο διαιρέτης τάσης.

✦ Η LDR και η αντίσταση 10kΩ πρέπει να είναι σε σειρά, με το κοινό σημείο να πηγαίνει στο A1.



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επέλεξε Arduino Uno.
4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επέλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.
7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περίμενε να εμφανιστεί το μήνυμα “Upload completed”.



Κώδικας

// Εφαρμογή : Σε συνθήκες σκότους ανάβει ένα LED, ενώ σε συνθήκες φωτός το LED παραμένει σβηστό.

```
int ldrPin = A1;           // είσοδος της φωτοαντίστασης
int ledPin = 10;           // έξοδος LED
int ldrValue = 0;          // αποθήκευση της μέτρησης φωτός

void setup() {
  pinMode(ledPin, OUTPUT); // ορίζουμε το pin 10 ως έξοδο
  Serial.begin(9600);       // ενεργοποιούμε τη σειριακή επικοινωνία για έλεγχο
}

void loop() {
  ldrValue = analogRead(ldrPin); // διαβάζουμε την τιμή από την LDR (0-1023)
  Serial.println(ldrValue);      // εμφανίζουμε τη μέτρηση στη σειριακή οθόνη

  if (ldrValue < 400) {          // αν το φως είναι χαμηλό (σκοτάδι)
    digitalWrite(ledPin, HIGH); // άναψε το LED
  } else {                       // αν υπάρχει φως
```

```
digitalWrite(ledPin, LOW); // σβήσε το LED
}

delay(200);                // μικρή καθυστέρηση για σταθερότητα
}
```

★ Η τιμή 400 είναι ένα ενδεικτικό «όριο φωτεινότητας». Μπορεί να προσαρμοστεί ανάλογα με τις συνθήκες.

✓ Έλεγχος Λειτουργίας

- ✓ Ανοίξτε τη σειριακή οθόνη για να δείτε τις τιμές που μετρά η φωτοαντίσταση.
- ✓ Σκεπάστε τη φωτοαντίσταση με το χέρι → το LED πρέπει να ανάψει.
- ✓ Φωτίστε τη φωτοαντίσταση → το LED πρέπει να σβήσει.



Μικρό Τεστ Ανατροφοδότησης

Επιλέξτε τη σωστή απάντηση:

1. Η φωτοαντίσταση μειώνει την αντίστασή της:
 - α. Στο σκοτάδι
 - β. Στο φως
 - γ. Όταν τη θερμάνουμε
 - δ. Όταν τη συνδέσουμε ανάποδα
2. Τι τιμές επιστρέφει η `analogRead()`; από το A1;
 - α. 0–5
 - β. 0–255
 - γ. 0–1023
 - δ. 0 ή 1
3. Τι ρόλο παίζει η αντίσταση 10kΩ στη συνδεσμολογία;
 - α. Για να μην καεί το LED
 - β. Σχηματίζει διαιρέτη τάσης με τη LDR
 - γ. Σταθεροποιεί την τάση 5V
 - δ. Κάνει το κύκλωμα πιο γρήγορο
4. Ποια εντολή χρησιμοποιούμε για να ανάψει το LED;
 - α. `analogWrite(ledPin, HIGH);`
 - β. `digitalWrite(ledPin, HIGH);`
 - γ. `pinMode(ledPin, OUTPUT);`
 - δ. `Serial.println(ledPin);`
5. Αν θέλουμε το LED να ανάβει σε δυνατό φως, πρέπει να:
 - α. Αντιστρέψουμε τη λογική στην `if`
 - β. Αλλάξουμε την αντίσταση της LDR
 - γ. Μετακινήσουμε το LED στο GND
 - δ. Χρησιμοποιήσουμε άλλο pin

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 6

ΕΝΔΕΙΚΤΗΣ LED ΕΝΟΣ ΨΗΦΙΟΥ

Στόχοι

Μετά την ολοκλήρωση του φύλλου, οι μαθητές/τριες θα μπορούν:

- Να γνωρίζουν τη λειτουργία του ενδείκτη 7 τμημάτων (7-segment display).
- Να συνδέουν σωστά τα τμήματα με το Arduino.
- Να εμφανίζουν αριθμούς από 0 έως 9.
- Να χρησιμοποιούν κουμπί για ελεγχόμενη αλλαγή ενδείξεων.
- Να αναπτύσσουν προγράμματα με πίνακες (arrays) και συνθήκες.

◆ ΕΦΑΡΜΟΓΗ 1: Διαδοχική και συνεχόμενη απεικόνιση των αριθμών 0–9 σε έναν ενδείκτη LED επτά τμημάτων.

Θεωρία

Ο ενδείκτης 7 τμημάτων (7-segment display) αποτελείται από 7 LED (a–g) που σχηματίζουν έναν αριθμό, συν ένα επιπλέον LED (dp) για την υποδιαστολή.

Υπάρχουν δύο τύποι:

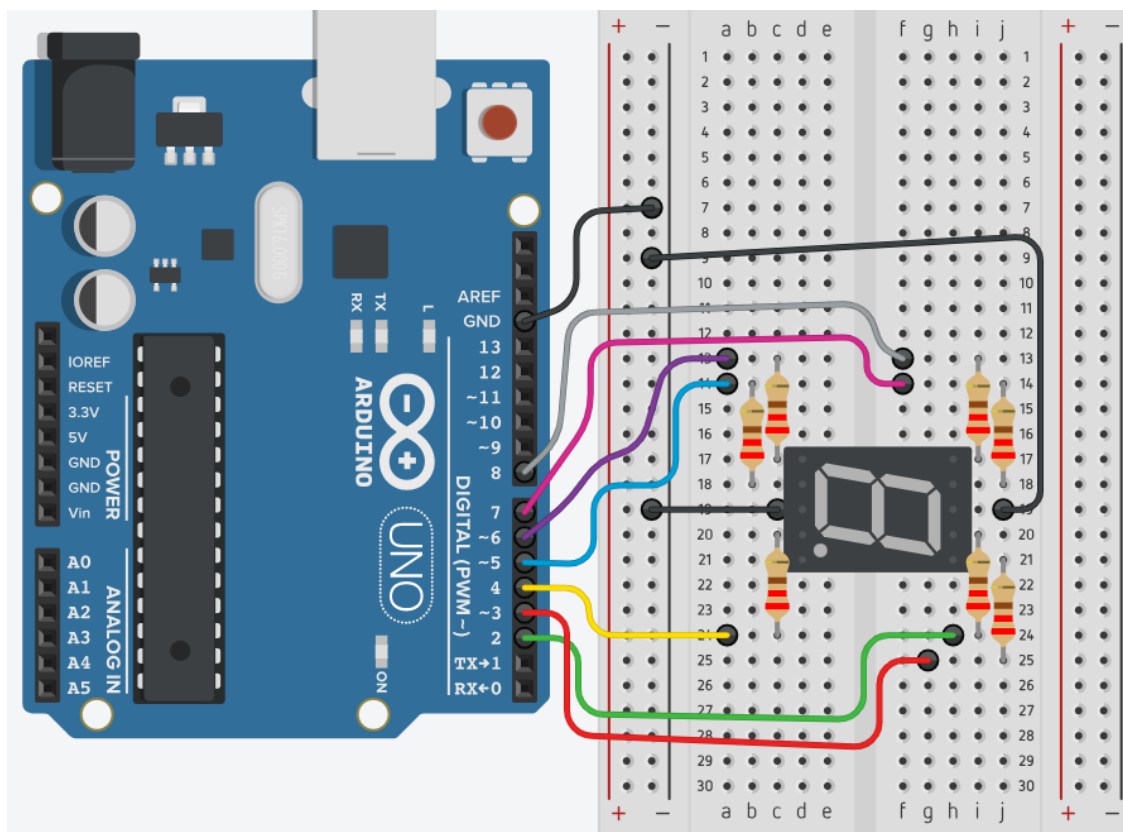
- Κοινού Καθοδικού (Common Cathode): όλα τα LED έχουν κοινό το GND.
- Κοινού Ανοδικού (Common Anode): όλα τα LED έχουν κοινό το +5V.

Στην άσκηση χρησιμοποιούμε κοινής καθόδου, οπότε κάθε τμήμα ανάβει όταν στο pin του δώσουμε HIGH.

Υλικά

- 1 × Arduino Uno
- 1 × Ενδείκτης 7 τμημάτων (common cathode)
- 7 × Αντιστάσεις 220Ω
- 1 × Breadboard
- Καλώδια σύνδεσης

🔌 Σχέδιο συνδεσμολογίας



🔧 Βήματα Σύνδεσης

1. Τοποθετήστε τον ενδείκτη στο breadboard.
2. Εντοπίστε το κοινό pin (cathode) και συνδέστε το στο GND του Arduino.
3. Συνδέστε κάθε τμήμα (a–g) σε διαφορετικό pin του Arduino μέσω αντίστασης 220Ω.
4. Ελέγξτε ότι οι συνδέσεις δεν βραχυκυκλώνουν.

☞ Δηλαδή:

a → pin 2

b → pin 3

c → pin 4

d → pin 5

e → pin 6

f → pin 7

g → pin 8

GND → GND Arduino

★ Για να βρείτε ποιο ποδαράκι είναι ποιο τμήμα, συμβουλευτείτε το datasheet του 7-seg.



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επέλεξε Arduino Uno.
4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επέλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.
7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περίμενε να εμφανιστεί το μήνυμα “Upload completed”.



Κώδικας

```
// Εφαρμογή 1: Διαδοχική και συνεχόμενη απεικόνιση των αριθμών 0–9

// Ορισμός pins για τα τμήματα του 7-segment
int segmentPins[7] = {2, 3, 4, 5, 6, 7, 8};

// Πίνακας που δείχνει ποια τμήματα ανάβουν για κάθε ψηφίο 0-9
// Σειρά: a, b, c, d, e, f, g
int digits[10][7] = {
  {1,1,1,1,1,0},    // 0
  {0,1,1,0,0,0},    // 1
  {1,1,0,1,1,0},    // 2
  {1,1,1,1,0,0},    // 3
```



```

{0,1,1,0,0,1,1},    // 4
{1,0,1,1,0,1,1},    // 5
{1,0,1,1,1,1,1},    // 6
{1,1,1,0,0,0,0},    // 7
{1,1,1,1,1,1,1},    // 8
{1,1,1,1,0,1,1}     // 9
};

void setup() {
    // Ορίζουμε όλα τα segment pins ως εξόδους
    for(int i=0; i<7; i++) {
        pinMode(segmentPins[i], OUTPUT);
    }
}

void loop() {
    for(int num=0; num<10; num++) {    // εμφάνιση 0-9
        showDigit(num);
        delay(1000);    // 1 δευτερόλεπτο ανά αριθμό
    }
}

// Συνάρτηση που ανάβει τα τμήματα για έναν αριθμό
void showDigit(int num) {
    for(int i=0; i<7; i++) {
        digitalWrite(segmentPins[i], digits[num][i]);
    }
}

```

✓ Έλεγχος Λειτουργίας

- ✓ Ο ενδείκτης θα εμφανίζει συνεχόμενα τους αριθμούς 0 έως 9.
- ✓ Αν παρατηρηθεί λάθος στα τμήματα, ελέγξτε αν τα pins έχουν συνδεθεί σωστά.

◆ ΕΦΑΡΜΟΓΗ 2: Διαδοχική απεικόνιση των αριθμών 0–9 με κάθε πάτημα ενός μπουτόν.

📖 Θεωρία

Με το μπουτόν εισάγουμε έλεγχο από τον χρήστη.

Κάθε φορά που πατάμε, αυξάνεται ο αριθμός κατά 1 και εμφανίζεται στον ενδείκτη.

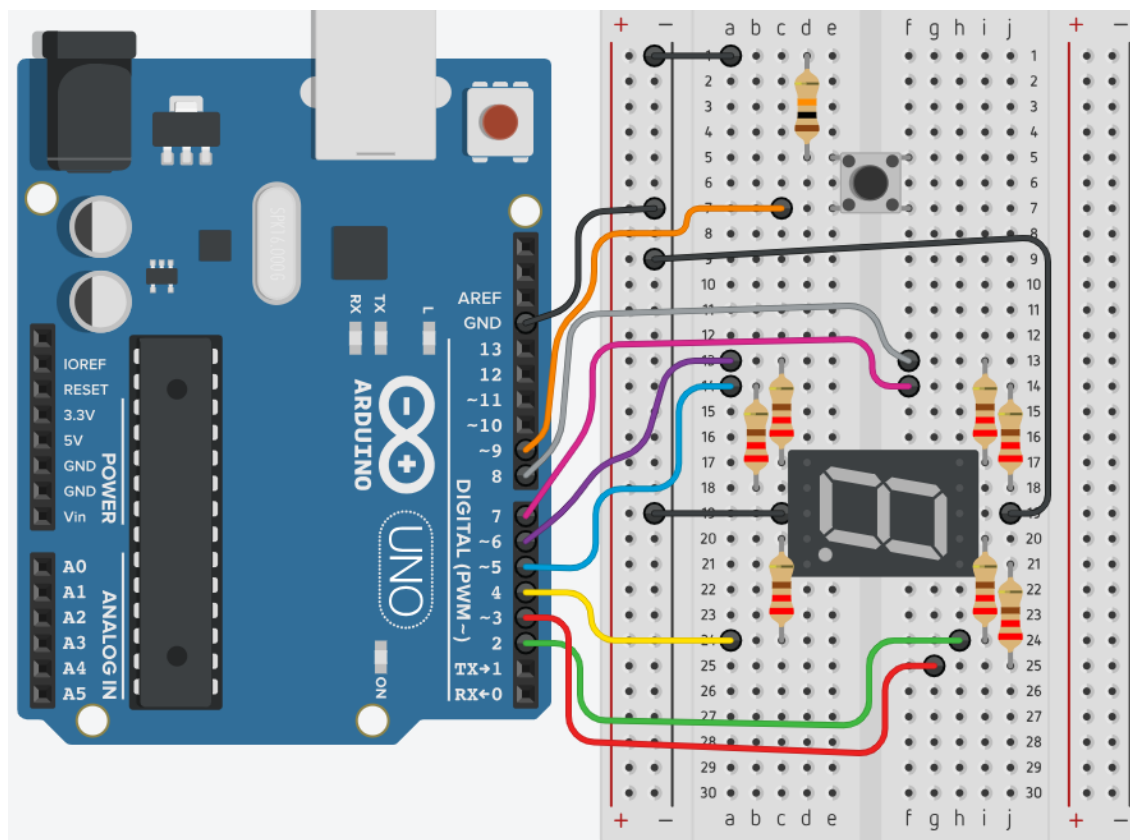
Όταν φτάσουμε στο 9, επιστρέφουμε στο 0.

✂ Υλικά

(επιπλέον από πριν)

- 1 × Μπουτόν
- 1 × Αντίσταση 10kΩ (pull-down)

🔌 Σχέδιο συνδεσμολογίας



🔧 Βήματα Σύνδεσης

1. Το μπουτόν στο pin 9.
2. Το άλλο άκρο στο GND.
3. Χρήση αντίστασης 10kΩ προς GND (pull-down).

★ Αν χρησιμοποιήσουμε `pinMode(buttonPin, INPUT_PULLUP)`, δεν χρειάζεται εξωτερική αντίσταση, αλλά το λογικό σήμα αντιστρέφεται (πατημένο = LOW).



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επίλεξε Arduino Uno.

4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επίλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.
7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περίμενε να εμφανιστεί το μήνυμα “Upload completed”.

Κώδικας

// Εφαρμογή 2: Διαδοχική απεικόνιση των αριθμών 0–9 με κάθε πάτημα ενός μπουτόν.

```
int segmentPins[7] = {2, 3, 4, 5, 6, 7, 8};

int buttonPin = 9;

int number = 0;           // αρχικός αριθμός
int lastState = LOW;      // αποθήκευση προηγούμενης κατάστασης

int digits[10][7] = {
  {1,1,1,1,1,1,0},       // 0
  {0,1,1,0,0,0,0},       // 1
  {1,1,0,1,1,0,1},       // 2
  {1,1,1,1,0,0,1},       // 3
  {0,1,1,0,0,1,1},       // 4
  {1,0,1,1,0,1,1},       // 5
  {1,0,1,1,1,1,1},       // 6
  {1,1,1,0,0,0,0},       // 7
  {1,1,1,1,1,1,1},       // 8
  {1,1,1,1,0,1,1}        // 9
};

void setup() {
  for(int i=0; i<7; i++) {
    pinMode(segmentPins[i], OUTPUT);
  }
}
```

```

pinMode(buttonPin, INPUT_PULLUP);           // χρησιμοποιούμε το εσωτερικό pull-up
showDigit(number);
}

void loop() {
  int buttonState = digitalRead(buttonPin);

  if(buttonState == LOW && lastState == HIGH) {      // αν ανιχνευτεί πάτημα
    number++;                                         // αύξησε αριθμό
    if(number > 9) number = 0;                       // επιστροφή στο 0
    showDigit(number);                             // εμφάνισε τον νέο αριθμό
    delay(200);                                     // καθυστέρηση για αποφυγή διπλοπατήματος (debounce)
  }

  lastState = buttonState;
}

void showDigit(int num) {
  for(int i=0; i<7; i++) {
    digitalWrite(segmentPins[i], digits[num][i]);
  }
}

```

✓ Έλεγχος Λειτουργίας

- ✓ Πατώντας το κουμπί, ο ενδείκτης προχωράει στον επόμενο αριθμό.
- ✓ Μετά το 9, επιστρέφει στο 0.



Μικρό Τεστ Ανατροφοδότησης

Επιλέξτε τη σωστή απάντηση:

1. Ο ενδείκτης 7-segment αποτελείται από:
 - α. 5 τμήματα
 - β. 7 τμήματα
 - γ. 8 τμήματα
 - δ. 10 τμήματα

2. Τι κοινό έχουν όλα τα LED σε έναν ενδείκτη κοινού καθόδου;
 - α. Το GND
 - β. Το +5V
 - γ. Το pin 13
 - δ. Την ίδια αντίσταση

3. Σε τι χρησιμεύει ο πίνακας `digits[][]` στον κώδικα;
 - α. Αποθηκεύει τον αριθμό που πληκτρολογούμε
 - β. Καθορίζει ποια τμήματα ανάβουν για κάθε αριθμό
 - γ. Ελέγχει το κουμπί
 - δ. Φορτώνει το πρόγραμμα

4. Γιατί χρησιμοποιούμε `INPUT_PULLUP` στο κουμπί;
 - α. Για να ανάβει πιο δυνατά
 - β. Για να μη χρειάζεται εξωτερική αντίσταση
 - γ. Για να σβήνει το LED
 - δ. Για να μετράει παλμούς

5. Τι συμβαίνει όταν ο μετρητής φτάσει στο 9;
 - α. Σταματάει
 - β. Συνεχίζει στο 10
 - γ. Επιστρέφει στο 0
 - δ. Σβήνει όλα τα LED

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 7

ΕΜΦΑΝΙΣΗ ΜΗΝΥΜΑΤΟΣ ΣΕ ΟΘΟΝΗ ΥΓΡΩΝ ΚΡΥΣΤΑΛΛΩΝ (LCD 2x16)

Στόχοι

- Να γνωρίσουμε την οθόνη LCD 2x16.
- Να μάθουμε πώς συνδέεται με το Arduino Uno.
- Να εμφανίσουμε ένα μήνυμα στην οθόνη.

ΕΦΑΡΜΟΓΗ: Εμφάνιση μηνύματος σε οθόνη LCD 2x16

Θεωρία

Η οθόνη LCD (Liquid Crystal Display) 2x16 μπορεί να εμφανίσει 2 γραμμές με 16 χαρακτήρες η καθεμία.

Για να επικοινωνήσει με το Arduino, χρησιμοποιούμε το ολοκληρωμένο I2C interface, που επιτρέπει να ελέγχουμε την LCD μόνο με 2 καλώδια (SDA και SCL).

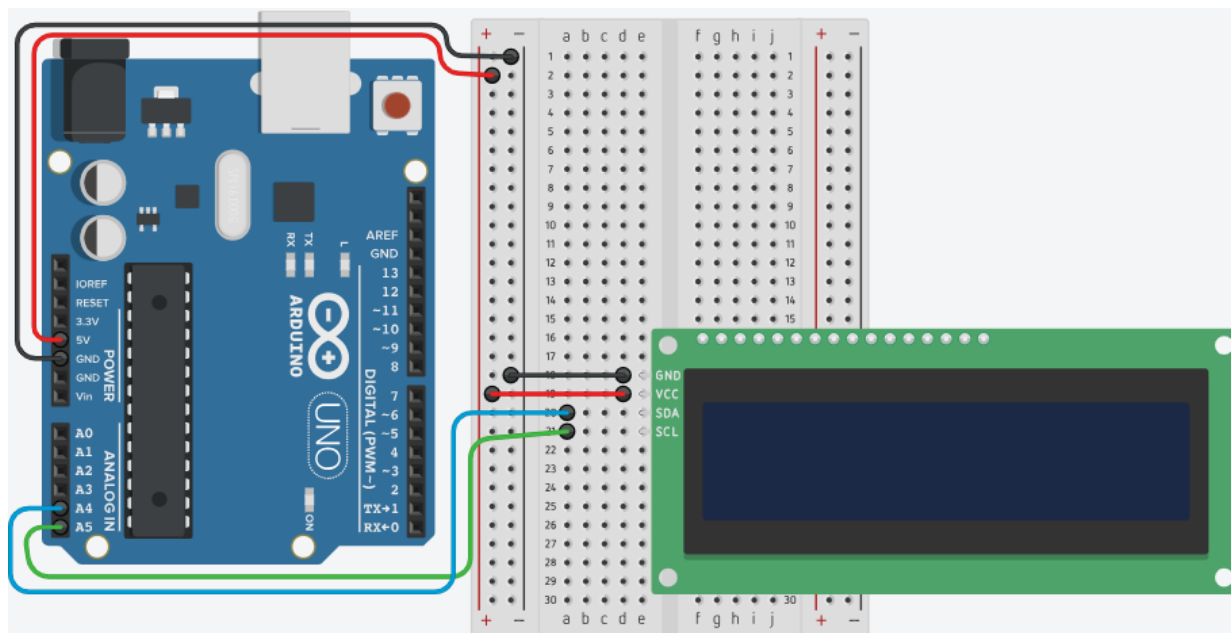
Τα πλεονεκτήματα της LCD:

- Μπορεί να εμφανίσει κείμενα, αριθμούς και σύμβολα.
- Εύκολη επικοινωνία με το Arduino μέσω της βιβλιοθήκης LiquidCrystal_I2C.

Υλικά

- 1 × Arduino Uno
- 1 × Οθόνη LCD 2x16 με αντάπτορα I2C
- Καλώδια σύνδεσης (jumper wires)
- Breadboard
- Καλώδιο USB για σύνδεση του Arduino με τον υπολογιστή

🔌 Σχέδιο συνδεσμολογίας



🔧 Βήματα Σύνδεσης

1. Σύνδεσε την οθόνη LCD στο breadboard.
2. Σύνδεσε τα καλώδια τροφοδοσίας (VCC → 5V, GND → GND).
3. Σύνδεσε το SDA της LCD στο pin A4 του Arduino.
4. Σύνδεσε το SCL της LCD στο pin A5 του Arduino.
5. Σύνδεσε το Arduino με τον υπολογιστή μέσω USB.

🔍 Δηλαδή:

- VCC → 5V (Arduino)
- GND → GND (Arduino)
- SDA → A4 (Arduino Uno)
- SCL → A5 (Arduino Uno)

★ Αν η οθόνη δεν δείχνει χαρακτήρες, ρύθμισε το μικρό ποτενσιόμετρο στο πίσω μέρος (contrast).



Εγκατάσταση Βιβλιοθήκης LiquidCrystal I2C

1. Άνοιξε το Arduino IDE.
2. Πήγαινε στο Sketch > Include Library > Manage Libraries...
3. Στο πεδίο αναζήτησης γράψε: LiquidCrystal I2C
4. Επίλεξε LiquidCrystal I2C by Frank de Brabander και πάτησε Install.
5. Αυτόματα θα κατέβει και η Wire.h που χρειάζεται για την επικοινωνία I2C.



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Εγκατέστησε τη βιβλιοθήκη LiquidCrystal_I2C (από το Library Manager).
4. Πήγαινε: Tools → Board → Arduino Uno.
5. Επέλεξε τη σωστή θύρα (π.χ. COM3).
6. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
7. Πάτησε ✓ Verify για έλεγχο.
8. Πάτησε → Upload για να φορτωθεί στο Arduino.



Κώδικας

```
// Εφαρμογή: Εμφάνιση μηνύματος σε LCD 2x16 με I2C

#include <Wire.h>           // Βιβλιοθήκη για I2C επικοινωνία
#include <LiquidCrystal_I2C.h> // Βιβλιοθήκη για LCD με I2C

LiquidCrystal_I2C lcd(0x27, 16, 2); // Διεύθυνση LCD (0x27), 16 στήλες, 2 γραμμές

void setup() {
  lcd.init();           // Αρχικοποίηση της LCD
  lcd.backlight();      // Ενεργοποίηση φωτισμού LCD
  lcd.setCursor(0, 0);  // Θέση κέρσορα: στήλη 0, γραμμή 0 (πρώτη γραμμή)

  lcd.print("Hello, Arduino!"); // Εμφάνιση μηνύματος στην πρώτη γραμμή
```

```
lcd.setCursor(0, 1);           // Μετακίνηση στη δεύτερη γραμμή
lcd.print("LCD 2x16 Demo");    // Εμφάνιση κειμένου στη δεύτερη γραμμή
}

void loop() {
    // Δεν χρειάζεται κάτι εδώ, το μήνυμα μένει στην οθόνη
}
```

Επεξήγηση εντολών:

- lcd.init(); → αρχικοποιεί την LCD.
- lcd.backlight(); → ανοίγει το φωτισμό της οθόνης.
- lcd.setCursor(x, y); → ορίζει σε ποια στήλη (x) και γραμμή (y) θα εμφανιστεί το κείμενο.
- lcd.print("..."); → εμφανίζει το μήνυμα που γράφουμε μέσα στα εισαγωγικά.

Έλεγχος Λειτουργίας

- ✓ Ανέβασε τον κώδικα στο Arduino.
- ✓ Η οθόνη πρέπει να δείξει το μήνυμα:
 - 1η γραμμή: Hello, Arduino!
 - 2η γραμμή: LCD 2x16 Demo
- ✓ Αν δεν φαίνονται γράμματα, ρύθμισε το ποτενσιόμετρο στο πίσω μέρος της LCD.



Μικρό Τεστ Ανατροφοδότησης

Επιλέξτε τη σωστή απάντηση:

1. Πόσους χαρακτήρες μπορεί να εμφανίσει μια LCD 2x16;
 - α) 16 χαρακτήρες σε 1 γραμμή
 - β) 32 χαρακτήρες σε 2 γραμμές
 - γ) 2 χαρακτήρες σε 16 γραμμές
 - δ) 8 χαρακτήρες σε 4 γραμμές

2. Ποια pins του Arduino Uno χρησιμοποιούνται για την επικοινωνία I2C;
 - α) D2 και D3
 - β) D10 και D11
 - γ) A4 και A5
 - δ) RX και TX

3. Τι κάνει η εντολή `lcd.setCursor(0,1);`;
 - α) Γράφει τον χαρακτήρα στην πρώτη γραμμή, πρώτη στήλη
 - β) Μετακινεί τον κέρσορα στη δεύτερη γραμμή, πρώτη στήλη
 - γ) Καθαρίζει την οθόνη
 - δ) Ενεργοποιεί τον φωτισμό

4. Τι χρειάζεται να κάνουμε αν η οθόνη δείχνει κενό ή ακατανόητα σύμβολα;
 - α) Να αλλάξουμε την πλακέτα Arduino
 - β) Να ρυθμίσουμε το ποτενσιόμετρο (contrast) της LCD
 - γ) Να ξαναγράψουμε τον κώδικα από την αρχή
 - δ) Να βγάλουμε το SDA pin

5. Ποια βιβλιοθήκη χρησιμοποιείται για την LCD με I2C;
 - α) Servo.h
 - β) Wire.h και LiquidCrystal_I2C.h
 - γ) EEPROM.h
 - δ) Stepper.h

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 8

ΜΕΤΡΗΤΗΣ ΘΕΡΜΟΚΡΑΣΙΑΣ

Στόχοι

Με το πέρας της άσκησης οι μαθητές/τριες θα μπορούν να:

- Συνδέουν αισθητήρα θερμοκρασίας με το Arduino.
- Διαβάζουν τιμές θερμοκρασίας από τον αισθητήρα.
- Εμφανίζουν τις μετρήσεις σε οθόνη υγρών κρυστάλλων (LCD 2x16).
- Ελέγχουν τη λειτουργία του κυκλώματος.

◆ ΕΦΑΡΜΟΓΗ: Μέτρηση της θερμοκρασίας

Θεωρία

Ο αισθητήρας LM35 δίνει τάση ανάλογη με τη θερμοκρασία.

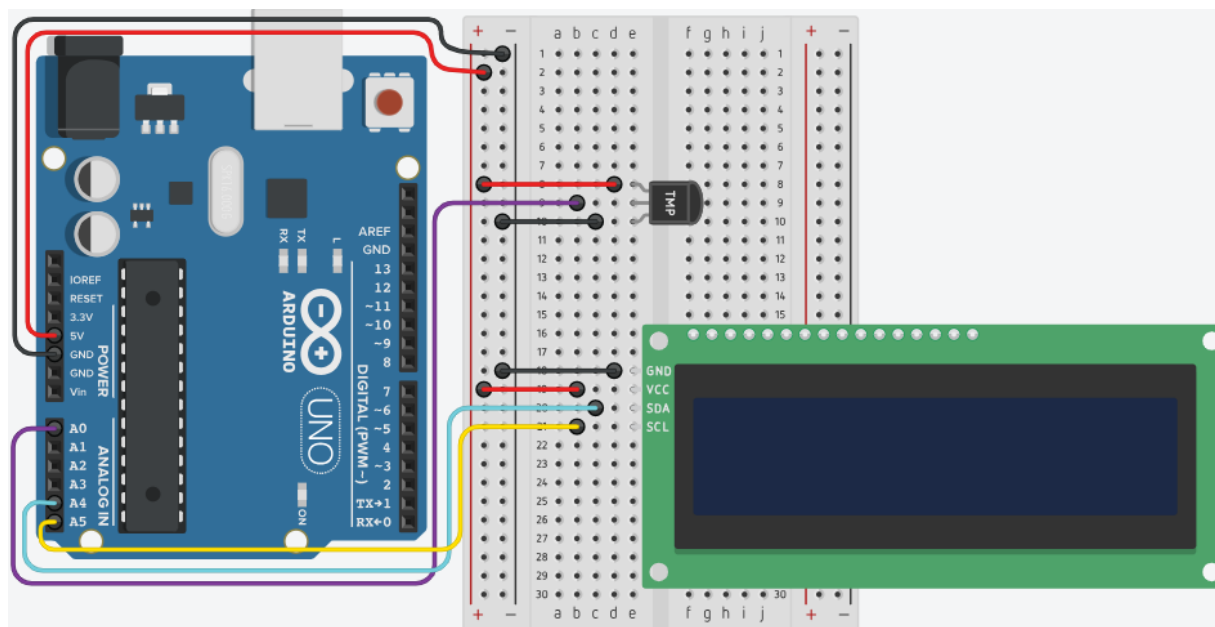
- Κάθε 10 mV αντιστοιχεί σε 1°C.
- Η έξοδος του συνδέεται σε αναλογική είσοδο του Arduino (π.χ. A0).

Η LCD (2x16) συνδέεται με το Arduino μέσω I2C (A4 → SDA, A5 → SCL). Για την εμφάνιση δεδομένων χρησιμοποιούμε τις βιβλιοθήκες Wire.h και LiquidCrystal_I2C.h.

Υλικά

- 1 x Arduino Uno
- 1 x Αισθητήρας θερμοκρασίας LM35
- 1 x Οθόνη LCD 2x16 με I2C
- 1 x Breadboard
- Καλώδια σύνδεσης (jumper wires)

🔌 Σχέδιο συνδεσμολογίας



🔧 Βήματα Σύνδεσης

1. Σύνδεση Αισθητήρα LM35

Ο LM35 έχει 3 ποδαράκια:

- Αριστερό: VCC (τροφοδοσία 5V)
- Μεσαίο: OUT (σήμα)
- Δεξί: GND (γείωση)

Σύνδεσε:

- το αριστερό ποδαράκι στον ακροδέκτη 5V του Arduino,
- το μεσαίο ποδαράκι στον ακροδέκτη A0 (αναλογική είσοδος),
- το δεξί ποδαράκι στον ακροδέκτη GND.

2. Σύνδεση Οθόνης LCD (16x2) με I2C module

- GND της LCD → GND του Arduino
- VCC της LCD → 5V του Arduino
- SDA της LCD → A4 του Arduino
- SCL της LCD → A5 του Arduino

★ Πρόσεξε τη σωστή κατεύθυνση του LM35 – η επίπεδη πλευρά να «κοιτάει» μπροστά σου.

✦ Αν η οθόνη δεν δείχνει καθαρά χαρακτήρες, γύρισε ελαφρά το μικρό ποτενσιόμετρο (trimmer) στο πίσω μέρος της LCD.



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επέλεξε Arduino Uno.
4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επέλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.
7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περιμένε να εμφανιστεί το μήνυμα “Upload completed”.



Κώδικας

```
// Εφαρμογή: Μέτρηση της θερμοκρασίας

#include <Wire.h>           // Βιβλιοθήκη για επικοινωνία I2C
#include <LiquidCrystal_I2C.h> // Βιβλιοθήκη για την LCD

LiquidCrystal_I2C lcd(0x27, 16, 2); // Διεύθυνση 0x27, LCD 16x2
int sensorPin = A0;         // Ο LM35 συνδεδεμένος στην αναλογική είσοδο A0
float tempC;                // Μεταβλητή για τη θερμοκρασία

void setup() {
  lcd.init();               // Αρχικοποίηση της LCD
  lcd.backlight();          // Ενεργοποίηση φωτισμού LCD
}

void loop() {
```

```

int sensorValue = analogRead(sensorPin);    // Διαβάζουμε την τιμή από τον αισθητήρα
tempC = (sensorValue * 5.0 * 100.0) / 1024;  // Μετατροπή σε βαθμούς Κελσίου

lcd.clear();                                // Καθαρίζουμε την οθόνη
lcd.setCursor(0, 0);                        // Θέση: πρώτη γραμμή
lcd.print("Thermometro");                  // Τίτλος
lcd.setCursor(0, 1);                        // Θέση: δεύτερη γραμμή
lcd.print("Temp: ");
lcd.print(tempC);                           // Εμφάνιση θερμοκρασίας
lcd.print(" C");                           // Μονάδα μέτρησης

delay(1000);                               // Αναμονή 1 δευτερόλεπτο
}

```

✓ Έλεγχος Λειτουργίας

- ✓ Βεβαιώσου ότι όλα τα GND είναι ενωμένα μεταξύ τους (Arduino, LM35, LCD).
- ✓ Έλεγξε ότι οι τροφοδοσίες (+5V) πάνε σωστά σε LCD και αισθητήρα.



Μικρό Τεστ Ανατροφοδότησης

Επιλέξτε τη σωστή απάντηση:

1. Ποια είναι η έξοδος του αισθητήρα LM35;
 - α) Ψηφιακό σήμα
 - β) Αναλογικό σήμα (τάση)
 - γ) Θερμική αντίσταση
 - δ) Παλμός

2. Πόσα mV αντιστοιχούν σε 1°C για τον LM35;
 - α) 1 mV
 - β) 5 mV
 - γ) 10 mV
 - δ) 100 mV

3. Σε ποια είσοδο συνδέεται ο LM35 στο Arduino Uno;
 - α) Ψηφιακή είσοδος D2
 - β) Αναλογική είσοδος A0
 - γ) PWM pin 9
 - δ) TX pin

4. Ποιες βιβλιοθήκες χρειάζονται για να λειτουργήσει η LCD με I2C;
 - α) Wire.h και LiquidCrystal_I2C.h
 - β) Servo.h
 - γ) EEPROM.h
 - δ) Stepper.h

5. Αν η θερμοκρασία αυξηθεί, τι πρέπει να δούμε στην LCD;
 - α) Σταθερή τιμή
 - β) Αύξηση της ένδειξης °C
 - γ) Μείωση της ένδειξης °C
 - δ) Σφάλμα

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 9

ΣΥΝΑΓΕΡΜΟΣ ΕΝΕΡΓΟΠΟΙΟΥΜΕΝΟΣ ΑΠΟ ΗΧΟ

Στόχοι

Με το πέρας της άσκησης οι μαθητές/τριες θα μπορούν να:

- Συνδέουν αισθητήρα ήχου (μικρόφωνο) με Arduino.
- Αναγνωρίζουν το κατώφλι ενεργοποίησης από τον αισθητήρα.
- Ενεργοποιούν έναν βομβητή (buzzer) και ένα LED μέσω του Arduino.
- Δημιουργούν ένα απλό κύκλωμα συναγερμού.

◆ ΕΦΑΡΜΟΓΗ : Δημιουργία ενός απλού συναγερμού κρότου

Θεωρία

Ο αισθητήρας ήχου διαθέτει μικρόφωνο και ενισχυτή.

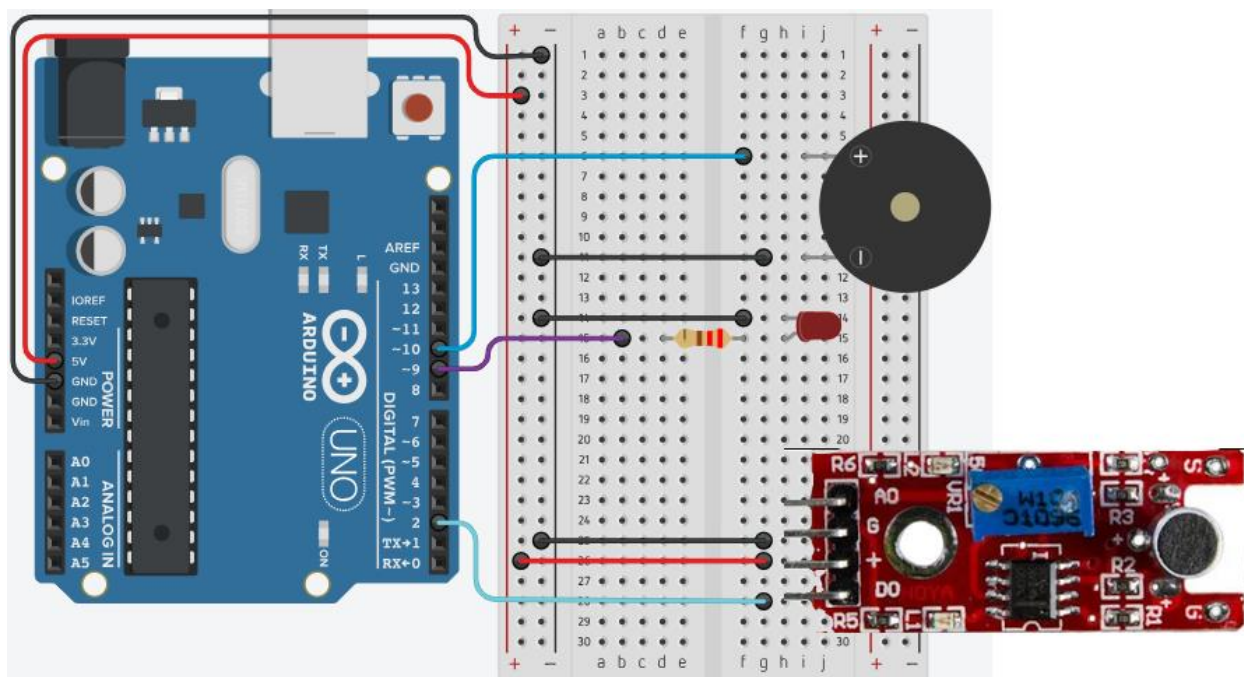
- Στην ψηφιακή του έξοδο (D0) δίνει λογικό HIGH (1) όταν ο ήχος ξεπερνάει ένα κατώφλι (ρυθμιζόμενο με ποτενσιόμετρο).
- Στην αναλογική του έξοδο (A0) μπορεί να δώσει την ένταση του ήχου.

Στην εφαρμογή αυτή αρκεί να χρησιμοποιήσουμε την ψηφιακή έξοδο (D0) ώστε όταν ανιχνεύεται ήχος, να ενεργοποιούνται buzzer και LED.

Υλικά

- 1 x Arduino Uno
- 1 x Αισθητήρας ήχου (microphone sound sensor)
- 1 x LED (flash)
- 1 x Αντίσταση 220Ω για το LED
- 1 x Buzzer (βομβητής)
- 1 x Breadboard
- Καλώδια σύνδεσης (jumper wires)

🔌 Σχέδιο συνδεσμολογίας



🔧 Βήματα Σύνδεσης

1. Αισθητήρας ήχου:
Ο αισθητήρας ήχου έχει συνήθως 3 pin:
 - VCC → 5V Arduino
 - GND → GND Arduino
 - OUT ή D0 → Pin 2 Arduino
2. LED :
 - Το μακρύ ποδαράκι (Άνοδος + (θετικό pin)) του LED συνδέεται σε ψηφιακή έξοδο του Arduino Pin 9 μέσω αντίστασης 220Ω
 - Το κοντό ποδαράκι (Κάθοδος – (αρνητικό pin)) του LED πάει στο GND.
3. Buzzer:
 - + (θετικό pin) → σε ψηφιακή έξοδο του Arduino Pin 10
 - – (αρνητικό pin) → GND Arduino

★ Ανάλογα με το module, η έξοδος δίνει HIGH όταν ανιχνεύσει έντονο ήχο (χτύπημα, παλαμάκι κ.λπ.).

★ Αν είναι παθητικός buzzer, μπορούμε αργότερα να παίξουμε και με tone() για διάφορους ήχους.



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επέλεξε Arduino Uno.
4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επέλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.
7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περίμενε να εμφανιστεί το μήνυμα “Upload completed”.



Κώδικας

```
// Εφαρμογή : Δημιουργία ενός απλού συναγερμού κρότου

int soundSensor = 2;           // Ψηφιακή έξοδος αισθητήρα ήχου
int ledPin = 9;                // LED
int buzzerPin = 10;           // Buzzer

void setup() {
  pinMode(soundSensor, INPUT);  // Ο αισθητήρας ήχου ως είσοδος
  pinMode(ledPin, OUTPUT);      // LED ως έξοδος
  pinMode(buzzerPin, OUTPUT);   // Buzzer ως έξοδος
}

void loop() {
  int soundState = digitalRead(soundSensor); // Διαβάζουμε την έξοδο του αισθητήρα

  if (soundState == HIGH) {      // Αν ανιχνευθεί ήχος
    digitalWrite(ledPin, HIGH);  // Ανάβει το LED
    digitalWrite(buzzerPin, HIGH); // Ενεργοποιείται ο βομβητής
  }
}
```

```
} else {                                     // Αν δεν ανιχνευθεί ήχος  
    digitalWrite(ledPin, LOW);               // Σβήνει το LED  
    digitalWrite(buzzerPin, LOW);           // Σταματά ο βομβητής  
}  
}
```

Επεξήγηση εντολών:

- pinMode(...) → Ορίζει αν το pin είναι είσοδος ή έξοδος.
- digitalRead(...) → Διαβάζει αν το pin έχει HIGH (1) ή LOW (0).
- digitalWrite(...) → Στέλνει HIGH (άναμμα/ενεργοποίηση) ή LOW (σβήσιμο/απενεργοποίηση) στο pin.

Έλεγχος Λειτουργίας

- ✓ Χτυπήστε παλαμάκια ή δημιουργήστε έντονο ήχο κοντά στον αισθητήρα.
- ✓ Ελέγξτε αν το LED αναβοσβήνει και ο βομβητής ηχεί κατά την ανίχνευση.
- ✓ Ρυθμίστε το ποτενσιόμετρο του αισθητήρα ώστε να ρυθμίσετε την ευαισθησία.



Μικρό Τεστ Ανατροφοδότησης

Επιλέξτε τη σωστή απάντηση:

1. Τι κάνει ο αισθητήρας ήχου;
 - α) Μετράει θερμοκρασία
 - β) Μετατρέπει την ένταση του ήχου σε ηλεκτρικό σήμα
 - γ) Ελέγχει φωτεινότητα
 - δ) Παράγει ήχο
2. Τι επιστρέφει η ψηφιακή έξοδος (D0) του αισθητήρα ήχου;
 - α) Συνεχή αναλογική τιμή της έντασης
 - β) HIGH ή LOW, ανάλογα αν ο ήχος ξεπέρασε το κατώφλι
 - γ) Πάντα HIGH
 - δ) Πάντα LOW
3. Σε ποιο pin του Arduino συνδέσαμε το LED στην εφαρμογή;
 - α) Pin 2
 - β) Pin 9
 - γ) Pin 10
 - δ) Pin A1
4. Ποια εντολή χρησιμοποιούμε για να διαβάσουμε την έξοδο του αισθητήρα ήχου;
 - α) digitalWrite()
 - β) analogRead()
 - γ) digitalRead()
 - δ) pinMode()
5. Τι πρέπει να κάνουμε αν ο συναγερμός ενεργοποιείται πολύ εύκολα ή καθόλου;
 - α) Να αλλάξουμε τον κώδικα
 - β) Να αλλάξουμε το Arduino
 - γ) Να ρυθμίσουμε το ποτενσιόμετρο του αισθητήρα
 - δ) Να αλλάξουμε το LED

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 10

Ανίχνευση απόστασης με αισθητήρα υπερήχων

Στόχοι

- Να γνωρίσουν οι μαθητές τον αισθητήρα υπερήχων HC-SR04.
- Να κατανοήσουν πώς μετράμε αποστάσεις με Arduino.
- Να συνδέουν περιφερειακά (led, βομβητή) με το Arduino.
- Να ελέγχουν τη λειτουργία της κατασκευής με βάση μετρήσεις από αισθητήρες.
- Να εξοικειωθούν με την εντολή `pulseIn()` και τον υπολογισμό απόστασης.

◆ ΕΦΑΡΜΟΓΗ 1: Εντοπισμός αντικειμένου που πλησιάζει στα 5cm. Όταν πλησιάσει το αντικείμενο, ηχεί ο βομβητής.

Θεωρία

Ο αισθητήρας υπερήχων HC-SR04 μετρά αποστάσεις στέλνοντας ένα σήμα υπερήχων από το ποδαράκι TRIG. Το σήμα επιστρέφει και λαμβάνεται από το ECHO.

Ο χρόνος επιστροφής μετατρέπεται σε απόσταση με τον τύπο:

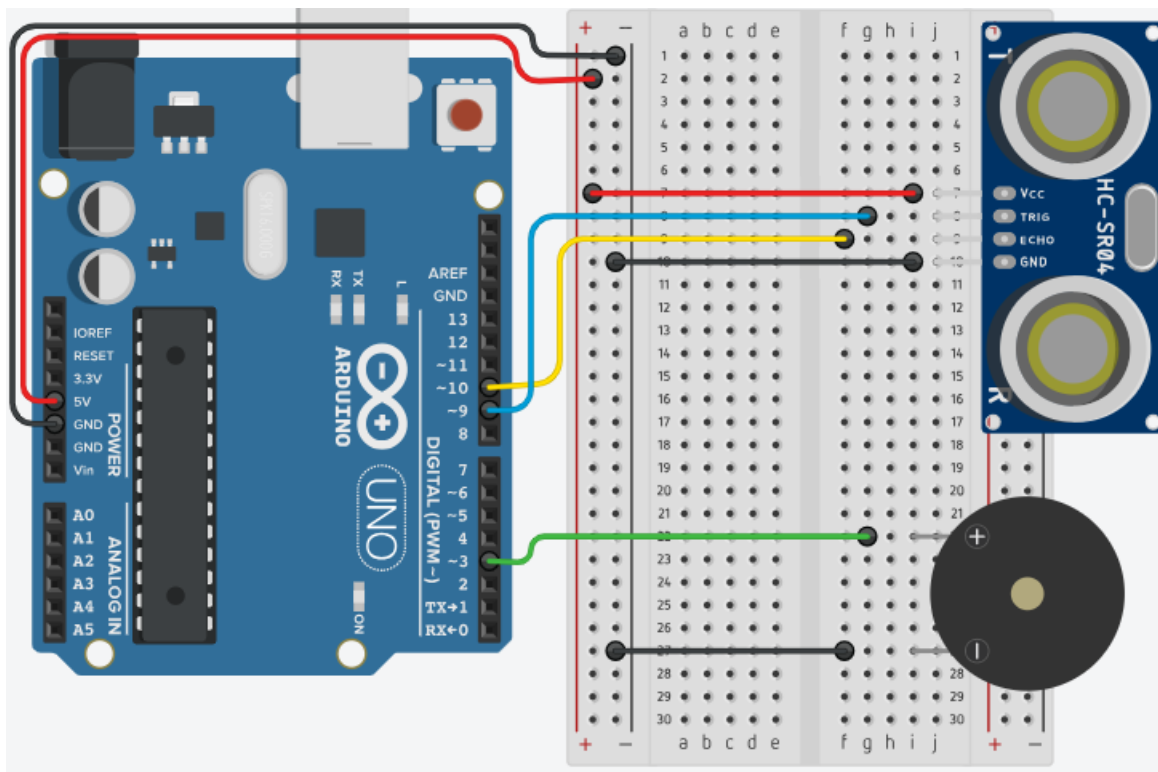
$$\text{Απόσταση (cm)} = \text{Χρόνος} \times 0.034 / 2$$

Το 0.034 cm/μs είναι η ταχύτητα του ήχου, ενώ η διαίρεση με το 2 γίνεται γιατί ο ήχος πάει και επιστρέφει.

Υλικά

- 1 Arduino Uno
- 1 αισθητήρας υπερήχων HC-SR04
- 1 Buzzer
- Καλώδια σύνδεσης
- Breadboard

Σχέδιο συνδεσμολογίας



Βήματα Σύνδεσης

1. Συνδέστε το VCC του HC-SR04 στο 5V του Arduino.
2. Συνδέστε το GND του αισθητήρα στο GND του Arduino.
3. Συνδέστε το TRIG στο pin 9.
4. Συνδέστε το ECHO στο pin 10.
5. Συνδέστε το θετικό του buzzer στο pin 3 και το αρνητικό στο GND.



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επέλεξε Arduino Uno.
4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επέλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.

7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περίμενε να εμφανιστεί το μήνυμα “Upload completed”.

Κώδικας

```
// Εφαρμογή 1: Συναγερμός απόστασης με μεταβλητό ήχο buzzer
```

```
const int trigPin = 9; // TRIG pin του αισθητήρα
const int echoPin = 10; // ECHO pin του αισθητήρα
const int buzzer = 3; // Pin του buzzer

long duration; // Μεταβλητή για τον χρόνο παλμού
int distance; // Μεταβλητή για την απόσταση σε cm

void setup() {
  pinMode(trigPin, OUTPUT); // Ορίζουμε το TRIG ως έξοδο
  pinMode(echoPin, INPUT); // Ορίζουμε το ECHO ως είσοδο
  pinMode(buzzer, OUTPUT); // Ορίζουμε το buzzer ως έξοδο
  Serial.begin(9600); // Έναρξη σειριακής επικοινωνίας
}

void loop() {
  // Στέλνουμε παλμό 10 μικροδευτερολέπτων στο TRIG
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Μετράμε τον χρόνο που το ECHO παραμένει HIGH
  duration = pulseIn(echoPin, HIGH);
```



```

// Υπολογισμός απόστασης
distance = duration * 0.034 / 2;

// Εμφάνιση απόστασης στη σειριακή οθόνη
Serial.print("Απόσταση: ");
Serial.print(distance);
Serial.println(" cm");

// Έλεγχος buzzer με μεταβλητό ήχο ανάλογα με την απόσταση
if (distance <= 5) {
    tone(buzzer, 1000); // Πολύ κοντά: έντονος ήχος (1kHz)
}
else if (distance <= 10) {
    tone(buzzer, 700); // Μέση απόσταση: μέτριος ήχος
}
else if (distance <= 20) {
    tone(buzzer, 400); // Μακρύτερα: χαμηλός ήχος
}
else {
    noTone(buzzer); // Αν το εμπόδιο είναι μακριά, σιωπή
}

delay(200); // Μικρή καθυστέρηση για σταθερότητα μετρήσεων
}

```

✓ Έλεγχος Λειτουργίας

- ✓ Ανοίξτε το Serial Monitor (9600 baud) και παρατηρήστε την ένδειξη απόστασης.
- ✓ Κρατήστε ένα αντικείμενο μπροστά στον αισθητήρα.
- ✓ Απομακρύνετε το αντικείμενο: ο βομβητής θα σταματήσει.

◆ ΕΦΑΡΜΟΓΗ 2 : Έξι LED ανάβουν ανάλογα με την απόσταση

📖 Θεωρία

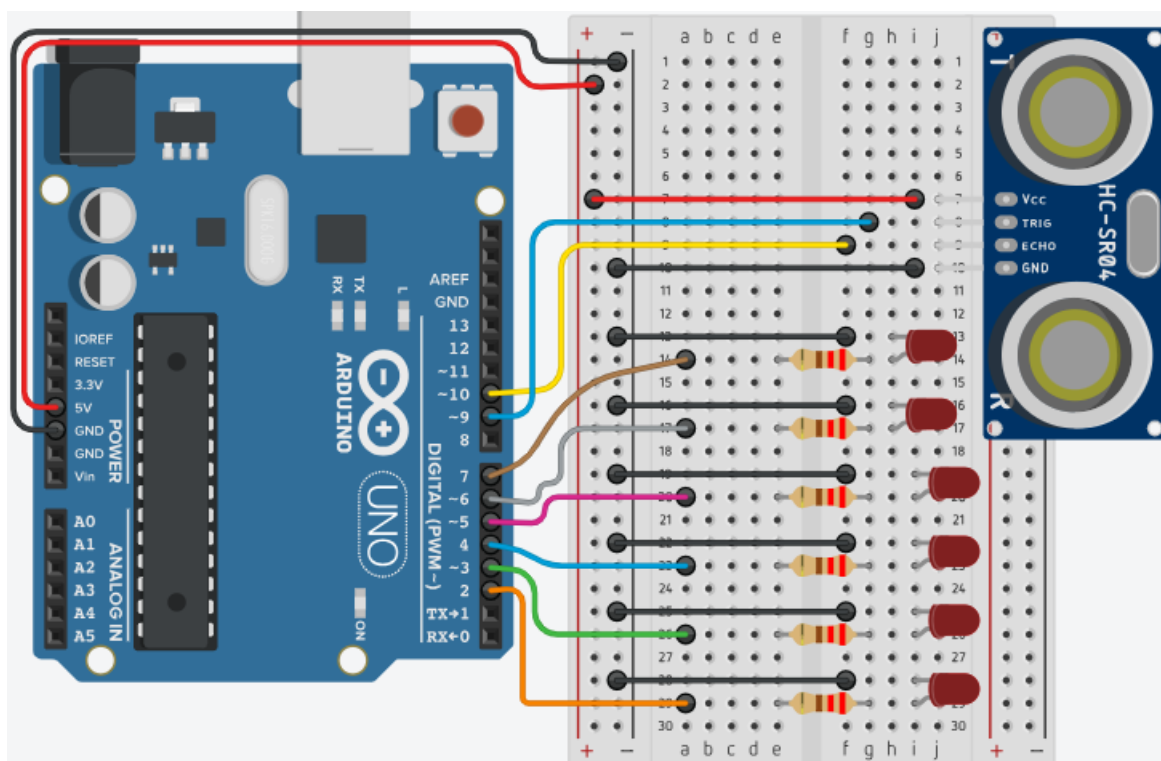
Η ίδια αρχή μέτρησης απόστασης ισχύει.

Σε αυτή την εφαρμογή, χρησιμοποιούμε σειρά LED σαν μπάρα για να δείχνουμε πόσο κοντά βρίσκεται το αντικείμενο.

✂ Υλικά

- 1 Arduino Uno
- 1 αισθητήρας υπερήχων HC-SR04
- 6 LED
- 6 Αντιστάσεις 220Ω
- Καλώδια σύνδεσης
- Breadboard

🔌 Σχέδιο συνδεσμολογίας





Βήματα Σύνδεσης

1. Συνδέστε τον αισθητήρα HC-SR04 όπως στην Εφαρμογή 1.
2. Συνδέστε τα LED στα pins 2 έως 7 με σειρά.
 - Pin 2 → LED1
 - Pin 3 → LED2
 - Pin 4 → LED3
 - Pin 5 → LED4
 - Pin 6 → LED5
 - Pin 7 → LED6
3. Κάθε LED να έχει αντίσταση 220Ω και να συνδεθεί στη γείωση.



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επέλεξε Arduino Uno.
4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επέλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.
7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περίμενε να εμφανιστεί το μήνυμα "Upload completed".



Κώδικας

```
// Εφαρμογή 2: Ανάβουν LED ανάλογα με την απόσταση
```

```
const int trigPin = 9;
```

```
const int echoPin = 10;
```

```
const int leds[] = {2, 3, 4, 5, 6, 7}; // Πίνακας με τα LED
```

```

long duration;
float distance;  // float για μεγαλύτερη ακρίβεια

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // Ορίζουμε όλα τα LED ως εξόδους
  for (int i = 0; i < 6; i++) {
    pinMode(leds[i], OUTPUT);
    digitalWrite(leds[i], LOW);
  }

  Serial.begin(9600);
}

void loop() {
  // Στέλνουμε παλμό trigger
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Μετράμε τον χρόνο επιστροφής με όριο 30ms για αποφυγή κολλήματος
  duration = pulseIn(echoPin, HIGH, 30000);
  // Αν δεν υπάρχει ανάκλαση (timeout)
  if (duration == 0) {

```

```

Serial.println("Καμία ανάκλαση - εκτός εύρους ή λάθος σύνδεση");
for (int i = 0; i < 6; i++) digitalWrite(leds[i], LOW);

delay(300);

return;
}

// Υπολογισμός απόστασης σε cm
distance = (duration * 0.034) / 2.0;

// Εμφάνιση στη σειριακή οθόνη
Serial.print("Απόσταση: ");
Serial.print(distance);
Serial.println(" cm");

// Σβήνουμε όλα τα LED
for (int i = 0; i < 6; i++) {
    digitalWrite(leds[i], LOW);
}

// Ανάβουμε LED ανάλογα με την απόσταση
if (distance <= 30 && distance > 25) digitalWrite(leds[0], HIGH);
else if (distance <= 25 && distance > 20) digitalWrite(leds[1], HIGH);
else if (distance <= 20 && distance > 15) digitalWrite(leds[2], HIGH);
else if (distance <= 15 && distance > 10) digitalWrite(leds[3], HIGH);
else if (distance <= 10 && distance > 5) digitalWrite(leds[4], HIGH);
else if (distance <= 5) digitalWrite(leds[5], HIGH);

delay(300);
}

```

Επεξήγηση εντολών:

- `pulseIn(echoPin, HIGH)` → μετράει σε μικροδευτερόλεπτα πόσο μένει HIGH το Echo.
- `distance = duration / 58;` → μετατροπή χρόνου σε cm.
- `map(distance, 30, 3, 0, 6)` → κλιμακώνει την απόσταση ώστε να αντιστοιχεί σε αριθμό LED.
- `digitalWrite(pin, HIGH/LOW)` → ενεργοποιεί ή σβήνει μια έξοδο.

Έλεγχος Λειτουργίας

- ✓ Κρατήστε ένα αντικείμενο μπροστά από τον αισθητήρα.
- ✓ Όσο πλησιάζετε, θα ανάβουν περισσότερα LED.
- ✓ Όταν το αντικείμενο είναι πολύ κοντά (≤ 5 cm), ανάβουν όλα τα LED.



Μικρό Τεστ Ανατροφοδότησης

Επιλέξτε τη σωστή απάντηση:

1. Ποια pin του αισθητήρα υπερήχων συνδέονται με το Arduino;
α) VCC, GND, DATA
β) TRIG, ECHO, VCC, GND
γ) RX, TX, VCC, GND
δ) CLK, DIO, VCC, GND
2. Τι μετρά ο αισθητήρας υπερήχων;
α) Θερμοκρασία
β) Ένταση φωτός
γ) Απόσταση
δ) Ήχο
3. Σε ποια απόσταση ενεργοποιείται ο βομβητής στην Εφαρμογή 1;
α) 10 cm
β) 5 cm
γ) 3 cm
δ) 1 cm
4. Τι συμβαίνει στην Εφαρμογή 2 όταν το αντικείμενο πλησιάσει στα 3 cm;
α) Σβήνουν όλα τα LED
β) Αναβοσβήνει μόνο το πρώτο LED
γ) Ανάβουν όλα τα LED
δ) Δεν ανάβει κανένα LED
5. Γιατί χρησιμοποιούμε αντιστάσεις στις συνδέσεις με τα LED;
α) Για να τα κάνουμε πιο φωτεινά
β) Για να προστατεύσουμε τα LED από υπερβολικό ρεύμα
γ) Για να λειτουργούν ως διακόπτες
δ) Για να αυξήσουμε την τάση

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 11

ΕΛΕΓΧΟΣ ΣΕΡΒΟΚΙΝΗΤΗΡΑ ΜΕ ΠΟΤΕΝΣΙΟΜΕΤΡΟ

Στόχοι

Μετά το τέλος της δραστηριότητας οι μαθητές/τριες θα είναι ικανοί να:

- Γνωρίζουν τι είναι ο σερβοκινητήρας και πώς λειτουργεί.
- Συνδέουν και ελέγχουν έναν σερβοκινητήρα μέσω Arduino.
- Κατανοούν τη λειτουργία του ποτενσιόμετρου ως αναλογικού αισθητήρα.
- Προγραμματίζουν τον έλεγχο ενός σερβοκινητήρα σε συγκεκριμένες γωνίες.
- Αντιλαμβάνονται τη χρησιμότητα του PWM και της βιβλιοθήκης Servo.h.

◆ ΕΦΑΡΜΟΓΗ : Έλεγχος γωνίας στροφής σερβοκινητήρα με ποτενσιόμετρο

Θεωρία

Ο σερβοκινητήρας (servo motor) είναι ένας μηχανισμός που μπορεί να περιστραφεί με ακρίβεια σε συγκεκριμένες γωνίες, συνήθως από 0° έως 180°.

Αποτελείται από έναν μικρό κινητήρα, έναν μηχανισμό μείωσης ταχύτητας και ένα σύστημα ανάδρασης που του επιτρέπει να γνωρίζει τη θέση του άξονα.

Η σύνδεση με το Arduino γίνεται με τρία καλώδια:

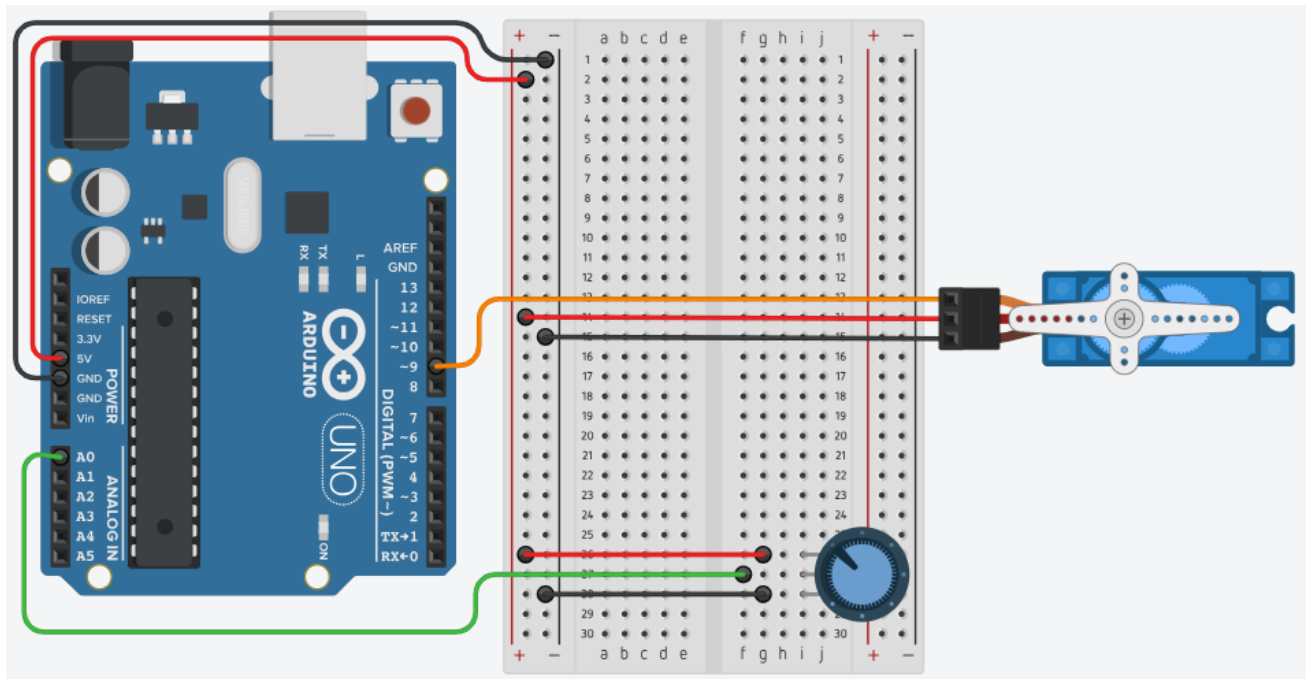
- Καφέ ή Μαύρο: GND (αρνητικός πόλος)
- Κόκκινο: +5V (τροφοδοσία)
- Κίτρινο ή Πορτοκαλί: Σήμα ελέγχου (σύνδεση με ψηφιακό pin)

Το ποτενσιόμετρο λειτουργεί ως διαιρέτης τάσης, παρέχοντας στο Arduino αναλογική τιμή από 0 έως 1023. Αυτή η τιμή αντιστοιχεί στη θέση του άξονα του ποτενσιόμετρου και μπορεί να μετατραπεί σε γωνία του servo.

Υλικά

- 1 Arduino Uno
- 1 Breadboard
- 1 Σερβοκινητήρας (SG90 ή παρόμοιος)
- 1 Ποτενσιόμετρο 10kΩ
- Καλώδια σύνδεσης (jumper wires)

Σχέδιο συνδεσμολογίας



Βήματα Σύνδεσης

1. Τοποθέτησε το ποτενσιόμετρο στο breadboard.
2. Σύνδεσε το μεσαίο ποδαράκι του στο A0.
3. Σύνδεσε το δεξί ποδαράκι στο GND και το αριστερό στο 5V.
4. Σύνδεσε το καφέ (GND) καλώδιο του servo στο GND του Arduino.
5. Σύνδεσε το κόκκινο καλώδιο του servo στο 5V.
6. Σύνδεσε το πορτοκαλί καλώδιο (σήμα) στο Pin 9.
7. Έλεγξε ότι όλες οι γειώσεις (GND) είναι κοινές.

★ Τοποθέτησε το servo σε σταθερή βάση για να μην περιστρέφεται ολόκληρο το σώμα του κατά τη λειτουργία.



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επέλεξε Arduino Uno.
4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επέλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.
7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περίμενε να εμφανιστεί το μήνυμα “Upload completed”.



Κώδικας

```
// Εφαρμογή: Έλεγχος γωνίας στροφής σερβοκινητήρα με ποτενσιόμετρο
#include <Servo.h>      // Εισάγουμε τη βιβλιοθήκη Servo για να ελέγχουμε σερβοκινητήρες

Servo myservo;         // Δημιουργούμε ένα αντικείμενο τύπου Servo
int potPin = A0;        // Ορίζουμε το pin που είναι συνδεδεμένο το ποτενσιόμετρο
int val;               // Μεταβλητή για την ανάγνωση της τιμής του ποτενσιόμετρου

void setup() {
  myservo.attach(9);    // Συνδέουμε το servo στο pin 9
}

void loop() {
  val = analogRead(potPin); // Διαβάζουμε την αναλογική τιμή από το ποτενσιόμετρο (0-1023)
  val = map(val, 0, 1023, 0, 180); // Μετατρέπουμε τη τιμή σε γωνία 0°-180°
  myservo.write(val);      // Θέτουμε το servo στη νέα γωνία
  delay(15);              // Μικρή καθυστέρηση για σταθερότητα
}
```

Επεξήγηση εντολών:

- `#include <Servo.h>`: Ενεργοποιεί τη βιβλιοθήκη ελέγχου servo.
- `myservo.attach(9)`: Καθορίζει σε ποιο pin είναι συνδεδεμένο το servo.
- `analogRead(A0)`: Διαβάζει την αναλογική τιμή από το ποτενσιόμετρο.
- `map(val, 0, 1023, 0, 180)`: Μετατρέπει την αναλογική τιμή σε αντίστοιχη γωνία.
- `myservo.write(val)`: Στρέφει το servo στη ζητούμενη γωνία.

Έλεγχος Λειτουργίας

- ✓ Περιστρέφοντας το ποτενσιόμετρο, ο σερβοκινητήρας πρέπει να αλλάζει γωνία από 0° έως 180°.
- ✓ Αν δεν κινείται:
 - Έλεγξε τη σύνδεση GND και 5V.
 - Βεβαιώσου ότι το pin 9 είναι σωστό.
 - Αν το servo τρεμοπαίζει, δοκίμασε εξωτερική τροφοδοσία 5V (π.χ. 2A).
- ★ Αν το servo κάνει μικρές ταλαντώσεις, δοκίμασε να προσθέσεις έναν πυκνωτή 100μF ανάμεσα στο +5V και GND για σταθερότερη λειτουργία.



Μικρό Τεστ Ανατροφοδότησης

Επιλέξτε τη σωστή απάντηση:

1. Ποια βιβλιοθήκη χρησιμοποιούμε για τον έλεγχο servo κινητήρων;
α) LiquidCrystal.h
β) Servo.h
γ) Ultrasonic.h
δ) Stepper.h
2. Ποιο pin στο Arduino δίνει σήμα ελέγχου στο servo στην άσκησή μας;
α) Pin 3
β) Pin 9
γ) Pin A0
δ) Pin 10
3. Ποια εντολή χρησιμοποιούμε για να περιστρέψουμε τον servo σε συγκεκριμένη γωνία;
α) digitalWrite()
β) analogRead()
γ) myservo.write()
δ) map()
4. Τι κάνει η εντολή map(val, 0, 1023, 0, 180); ;
α) Διαβάζει το ποτενσιόμετρο
β) Μετατρέπει την τιμή 0–1023 σε 0–180°
γ) Εκτυπώνει τιμές
δ) Σβήνει το servo
5. Τι ρόλο παίζει το ποτενσιόμετρο;
α) Ρυθμίζει τη φωτεινότητα του LED
β) Ρυθμίζει τη γωνία του servo
γ) Ενεργοποιεί το Arduino
δ) Αλλάζει την ταχύτητα περιστροφής

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 12

ΨΗΦΙΑΚΟ ΖΑΡΙ

Στόχοι

Μετά το τέλος της δραστηριότητας οι μαθητές/τριες θα είναι ικανοί να:

- Κατανοούν τη χρήση της εντολής `random()` για παραγωγή τυχαίων αριθμών.
- Μαθαίνουν να χρησιμοποιούν ένα κουμπί για ενεργοποίηση λειτουργίας.
- Εξασκούνται στον έλεγχο πολλαπλών LED και 7-τμηματικής οθόνης (7-segment).
- Αναγνωρίζουν τη διαφορά μεταξύ αναλογικής και ψηφιακής απεικόνισης.
- Εφαρμόζουν τη λογική εισόδου-εξόδου για τη δημιουργία ενός ψηφιακού παιχνιδιού τύχης.

ΕΦΑΡΜΟΓΗ 1: Ψηφιακό Ζάρι με LED

Θεωρία

Το Arduino μπορεί να παράγει τυχαίους αριθμούς με την εντολή `random(min, max)`.

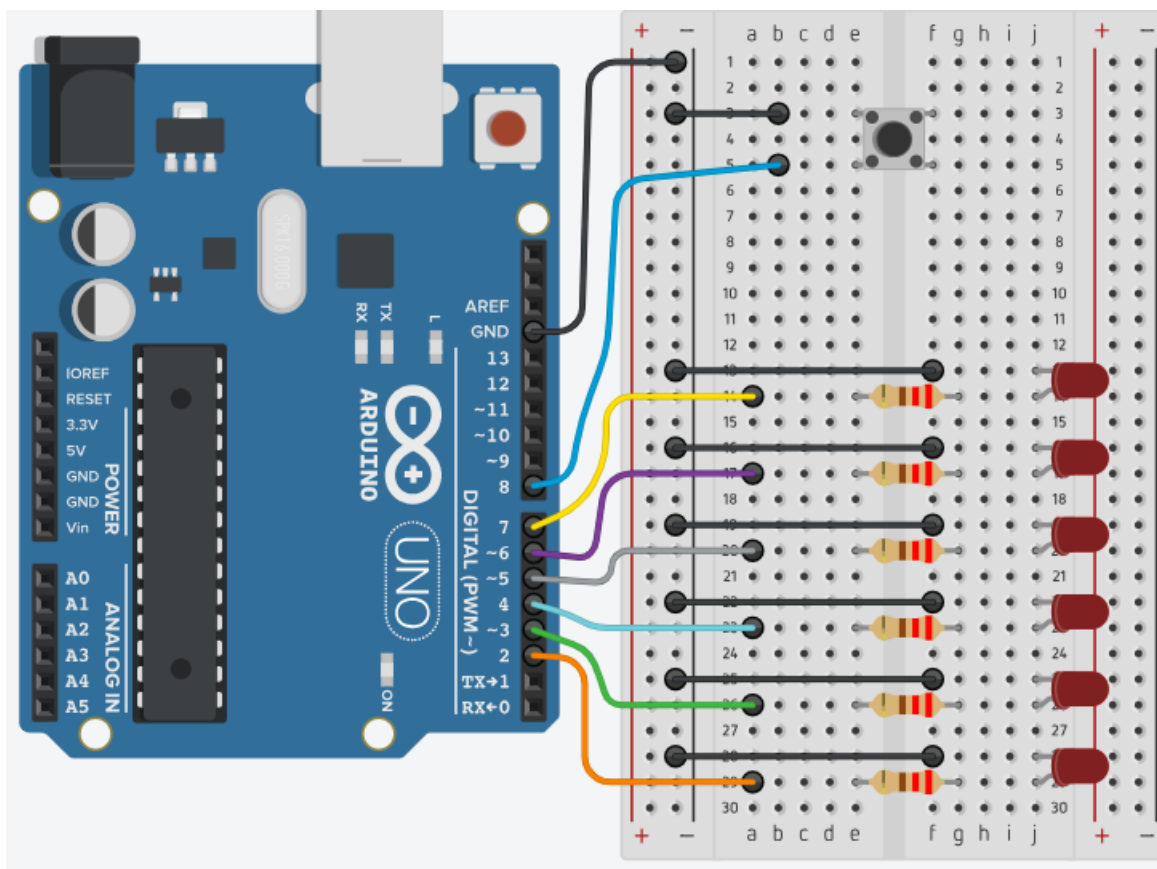
Ο αριθμός χρησιμοποιείται για να ενεργοποιεί τα LED σε διαφορετικό συνδυασμό, όπως οι κουκίδες ενός πραγματικού ζαριού.

Η είσοδος από το κουμπί “ρίχνει” το ζάρι, ενώ η έξοδος στα LED εμφανίζει το αποτέλεσμα.

Υλικά

- 1 Arduino Uno
- 6 LED
- 6 Αντιστάσεις 220Ω
- 1 Push button
- 1 Breadboard
- Καλώδια σύνδεσης -

Σχέδιο συνδεσμολογίας



Βήματα Σύνδεσης

1. Συνδέστε τα LED στα pin 2–7 του Arduino.
2. Συνδέστε κάθε LED σε σειρά με μία αντίσταση 220Ω προς GND.
3. Συνδέστε το κουμπί στο pin 8 και στο GND.
4. Ορίστε στο πρόγραμμα το pin του κουμπιού ως INPUT_PULLUP (ώστε να μην χρειάζεται εξωτερική αντίσταση).
5. Ελέγξτε ότι όλα τα LED ανάβουν αν τα δοκιμάσετε με `digitalWrite(pin, HIGH)`.



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επίλεξε Arduino Uno.

4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επίλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.
7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περίμενε να εμφανιστεί το μήνυμα “Upload completed”.

Κώδικας

```
// Εφαρμογή 1: Ψηφιακό Ζάρι με LED
int leds[] = {2, 3, 4, 5, 6, 7};
int buttonPin = 8;

void setup() {
  for (int i = 0; i < 6; i++) pinMode(leds[i], OUTPUT);
  pinMode(buttonPin, INPUT_PULLUP);
  randomSeed(analogRead(A0));      // Δημιουργία πιο τυχαίων αριθμών
}

void loop() {
  if (digitalRead(buttonPin) == LOW) { // Αν πατηθεί το κουμπί
    int number = random(1, 7);        // Τυχαίος αριθμός 1–6
    showNumber(number);
    delay(200);                       // Μικρή καθυστέρηση για αποφυγή αναπήδησης
  }
}

void showNumber(int n) {
  // Σβήσε όλα τα LED
  for (int i = 0; i < 6; i++) digitalWrite(leds[i], LOW);
}
```

```
// Άναψε ανάλογα με τον αριθμό (όπως στο ζάρι)

switch (n) {

  case 1: digitalWrite(leds[3], HIGH); break;

  case 2: digitalWrite(leds[0], HIGH); digitalWrite(leds[5], HIGH); break;

  case 3: digitalWrite(leds[0], HIGH); digitalWrite(leds[3], HIGH); digitalWrite(leds[5], HIGH); break;

  case 4: digitalWrite(leds[0], HIGH); digitalWrite(leds[1], HIGH); digitalWrite(leds[4], HIGH);
digitalWrite(leds[5], HIGH); break;

  case 5: digitalWrite(leds[0], HIGH); digitalWrite(leds[1], HIGH); digitalWrite(leds[3], HIGH);
digitalWrite(leds[4], HIGH); digitalWrite(leds[5], HIGH); break;

  case 6: for (int i = 0; i < 6; i++) digitalWrite(leds[i], HIGH); break;

}

}
```

✓ Έλεγχος Λειτουργίας

- ✓ Πατώντας το κουμπί, ανάβει διαφορετικός συνδυασμός LED κάθε φορά.
- ✓ Οι αριθμοί 1–6 εμφανίζονται σε τυχαία σειρά.
- ✓ Τα LED αντιπροσωπεύουν σωστά τις “κουκίδες” του ζαριού.

◆ ΕΦΑΡΜΟΓΗ 2: Ψηφιακό Ζάρι με 7 – Segment Display

Θεωρία

Η 7-τμηματική οθόνη (7-segment display) αποτελείται από 7 LED (τμήματα a–g) που σχηματίζουν ψηφία.

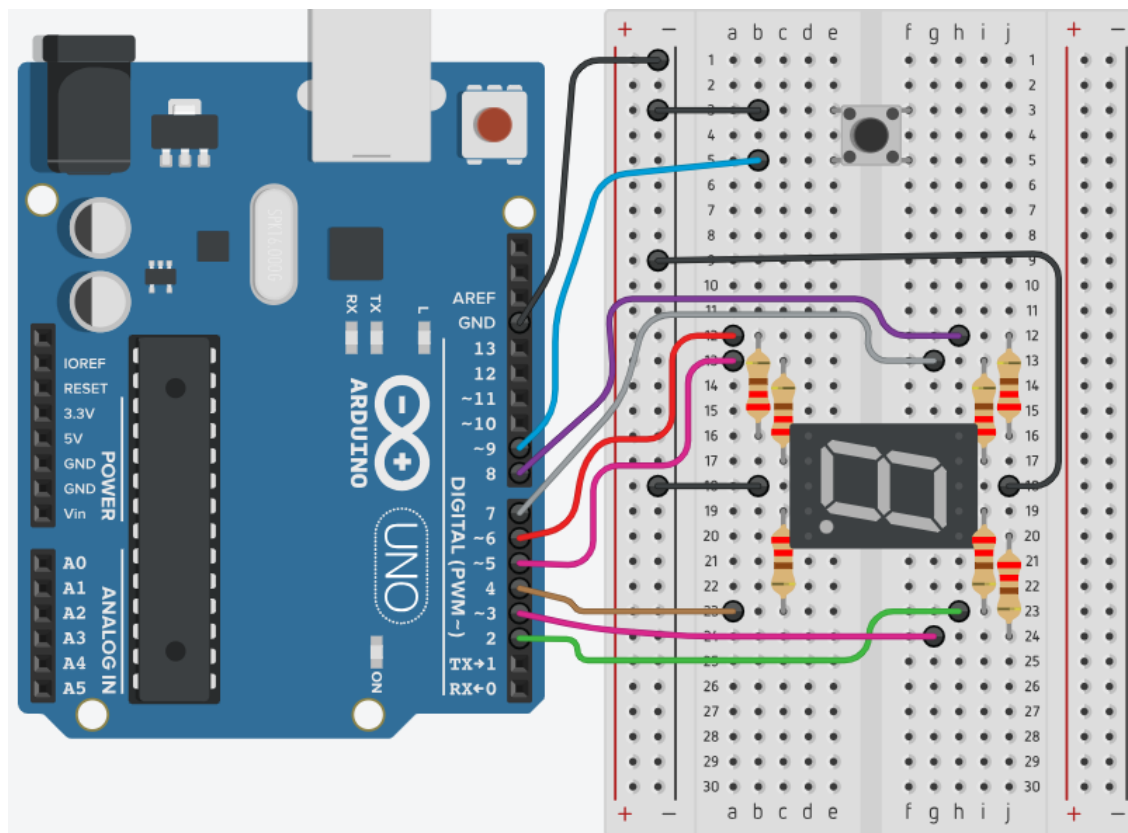
Με ενεργοποίηση των κατάλληλων τμημάτων, μπορούμε να εμφανίσουμε αριθμούς από 0 έως 9.

Στην εφαρμογή, το αποτέλεσμα του “ζαριού” θα εμφανίζεται ως ψηφίο 1–6.

Υλικά

- 1 Arduino Uno
- 1 7-segment display (κοινής καθόδου)
- 7 Αντιστάσεις 220Ω
- 1 Push button
- 1 Breadboard
- Καλώδια σύνδεσης

Σχέδιο συνδεσμολογίας



Βήματα Σύνδεσης

1. Συνδέστε κάθε τμήμα a–g του display σε pin 2–8 του Arduino μέσω αντίστασης 220Ω.
2. Συνδέστε τους δύο κοινούς καθόδους (COM) στο GND.
3. Συνδέστε το κουμπί στο pin 9 και στο GND.
4. Ορίστε το pin του κουμπιού ως INPUT_PULLUP.



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επίλεξε Arduino Uno.
4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επίλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.

7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περιμένε να εμφανιστεί το μήνυμα “Upload completed”.

Κώδικας

```
// Εφαρμογή 2: Ψηφιακό Ζάρι με 7 – Segment Display
int segPins[] = {2, 3, 4, 5, 6, 7, 8};      // a,b,c,d,e,f,g
int buttonPin = 9;

byte numbers[10][7] = {
  {1,1,1,1,1,1,0},      // 0
  {0,1,1,0,0,0,0},      // 1
  {1,1,0,1,1,0,1},      // 2
  {1,1,1,1,0,0,1},      // 3
  {0,1,1,0,0,1,1},      // 4
  {1,0,1,1,0,1,1},      // 5
  {1,0,1,1,1,1,1},      // 6
};

void setup() {
  for (int i=0; i<7; i++) pinMode(segPins[i], OUTPUT);
  pinMode(buttonPin, INPUT_PULLUP);
  randomSeed(analogRead(A0));
}

void loop() {
  if (digitalRead(buttonPin) == LOW) {
    int n = random(1, 7);
```

```

    showNumber(n);
    delay(200);
}
}

void showNumber(int n) {
    for (int i=0; i<7; i++) {
        digitalWrite(segPins[i], numbers[n][i]);
    }
}

```

✓ Έλεγχος Λειτουργίας

- ✓ Με κάθε πάτημα του κουμπιού, το display δείχνει διαφορετικό ψηφίο 1–6.
- ✓ Η οθόνη σβήνει και αλλάζει σωστά σε κάθε ρίψη.
- ✓ Οι αριθμοί εμφανίζονται καθαρά και σωστά σχηματισμένοι.

✦ Αν τα ψηφία εμφανίζονται “αντίστροφα”, αλλάξτε την λογική $1 \leftrightarrow 0$ στην πινάκα numbers.

✦ Αν χρησιμοποιείς common anode, πρέπει να αντιστρέψεις όλα τα bits (HIGH $\leftrightarrow$ LOW).

✦ Μπορείς να προσθέσεις ήχο με buzzer κατά τη ρίψη του ζαριού για πιο διαδραστική εμπειρία.

Μικρό Τεστ Ανατροφοδότησης

1. Ποια εντολή χρησιμοποιείται για την παραγωγή τυχαίων αριθμών;
2. Τι κάνει η εντολή randomSeed(analogRead(A0));;
3. Πώς λειτουργεί το INPUT_PULLUP;
4. Τι είδους LED περιέχει ένα 7-segment display κοινής καθόδου;
5. Ποια είναι η βασική διαφορά μεταξύ common cathode και common anode display;

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 13

ΗΛΕΚΤΡΟΝΙΚΟ ΖΑΡΙ ΜΕ 7-SEGMENT DISPLAY ΚΑΙ BUZZER

Στόχοι

Μετά την ολοκλήρωση του φύλλου, οι μαθητές/τριες θα μπορούν να:

- Κατανοούν τη λειτουργία του 7-segment display κοινής καθόδου.
- Ελέγχουν την απεικόνιση αριθμών μέσω Arduino.
- Χρησιμοποιούν buzzer για ηχητική ειδοποίηση.
- Συνδυάζουν τυχαίους αριθμούς και ήχο σε μία ολοκληρωμένη εφαρμογή.

◆ ΕΦΑΡΜΟΓΗ: Ηλεκτρονικό Ζάρι με 7-Segment Display και Buzzer

Θεωρία

Το 7-segment display είναι μια διάταξη από 7 LED (και ένα προαιρετικό όγδοο για την τελεία).

Κάθε LED αντιστοιχεί σε ένα τμήμα (segment) με ονομασία a, b, c, d, e, f, g.

Ανάβοντας διαφορετικούς συνδυασμούς, σχηματίζονται οι αριθμοί 0–9.

Υπάρχουν δύο τύποι:

- Κοινής ανόδου (CA): όλα τα +5V είναι κοινά.
- Κοινής καθόδου (CC): όλες οι καθόδους (GND) είναι κοινές.

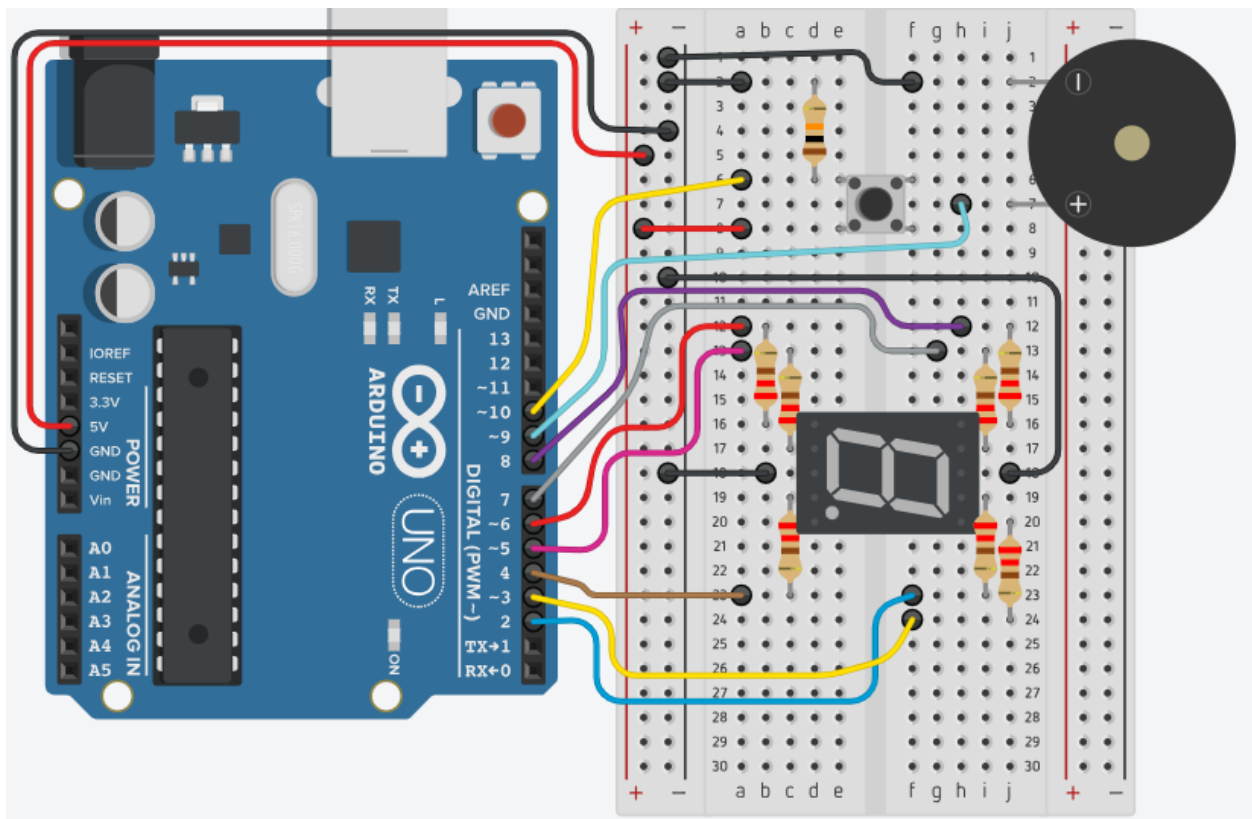
 Στην άσκηση αυτή χρησιμοποιούμε κοινής καθόδου, άρα το GND είναι κοινό και ενεργοποιούμε τα LED με λογικό HIGH (1).

 Το buzzer θα χρησιμοποιηθεί για να παράγει ήχο κάθε φορά που “ρίχνουμε” το ζάρι.

✂ Υλικά

- 1 Arduino Uno
- 1 7-segment display κοινής καθόδου
- 1 κουμπί (push button)
- 1 buzzer
- 7 Αντιστάσεις 220Ω
- 1 Αντίσταση 10KΩ
- Breadboard
- καλώδια

🔌 Σχέδιο συνδεσμολογίας





Βήματα Σύνδεσης

1. Συνδέστε κάθε τμήμα a–g του display σε pin 2–8 του Arduino μέσω αντίστασης 220Ω.
2. Συνδέστε τους δύο κοινούς καθόδους (COM) στο GND.
3. Συνδέστε το Buzzer στο pin 9 και στο GND.
4. Συνδέστε το ένα άκρο από το κουμπί με αντίσταση 10kΩ από Pin 10 προς GND (pull-down) και το άλλο άκρο στο GND.



Φόρτωση προγράμματος στο Arduino Uno (βήμα-βήμα)

1. Σύνδεσε το Arduino Uno με τον υπολογιστή μέσω USB.
2. Άνοιξε το Arduino IDE.
3. Πήγαινε στο Tools (Εργαλεία) → Board (Πλακέτα) → επέλεξε Arduino Uno.
4. Πήγαινε στο Tools (Εργαλεία) → Port (Θύρα) και επέλεξε τη θύρα που εμφανίζει το Arduino (π.χ. COM3 ή /dev/ttyUSB0).
5. Αντέγραψε και επικόλλησε τον παρακάτω κώδικα.
6. Πάτησε το κουμπί ✓ Verify (Επαλήθευση) για να ελέγξεις αν ο κώδικας είναι σωστός.
7. Πάτησε το κουμπί → Upload (Μεταφόρτωση) για να φορτώσει το πρόγραμμα στο Arduino.
8. Περίμενε να εμφανιστεί το μήνυμα “Upload completed”.



Κώδικας

```
// Εφαρμογή : Ηλεκτρονικό Ζάρι με 7-Segment Display και Buzzer
```

```
// Πίνακας με τους συνδυασμούς για κάθε αριθμό 1-6
```

```
int segments[] = {2,3,4,5,6,7,8};
```

```
int buttonPin = 10;
```

```
int buzzerPin = 9;
```

```
byte digits[7][7] = {
```

```
    // a, b, c, d, e, f, g
```

```
{0,0,0,0,0,0,0},    // 0 (δεν χρησιμοποιείται)
```

```
{0,1,1,0,0,0,0},    // 1
```

```
{1,1,0,1,1,0,1},    // 2
```

```

{1,1,1,1,0,0,1},      // 3
{0,1,1,0,0,1,1},      // 4
{1,0,1,1,0,1,1},      // 5
{1,0,1,1,1,1,1}       // 6
};

void setup() {
  for (int i=0; i<7; i++) pinMode(segments[i], OUTPUT);
  pinMode(buttonPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  randomSeed(analogRead(A0));    // Τυχαία εκκίνηση
}

void loop() {
  if (digitalRead(buttonPin) == HIGH) {
    tone(buzzerPin, 800);        // ήχος “ρίψης”
    for (int i=0; i<10; i++) {   // 10 “γυρίσματα” του ζαριού
      int r = random(1,7);
      showNumber(r);
      delay(100 + i*30);        // σταδιακή επιβράδυνση
    }
    noTone(buzzerPin);
  }
}

// Συνάρτηση εμφάνισης αριθμού
void showNumber(int num) {
  for (int i=0; i<7; i++) {
    digitalWrite(segments[i], digits[num][i]);
  }
}

```


Επεξήγηση εντολών:

- `randomSeed(analogRead(A0))`: εξασφαλίζει πιο “τυχαία” νούμερα.
- `tone(buzzerPin, 800)`: παράγει ήχο στα 800Hz.
- `delay(100 + i*30)`: δημιουργεί το εφέ της ρεαλιστικής ρίψης (επιβράδυνση).
- Οι τιμές στα arrays `digits` αντιστοιχούν στα `segments` που πρέπει να ανάβουν (1=on, 0=off).

Έλεγχος Λειτουργίας

- ✓ Φόρτωσε το πρόγραμμα στο Arduino.
- ✓ Πάτησε το κουμπί — θα ακουστεί ήχος και στο 7-segment θα εμφανιστεί τυχαίος αριθμός 1–6.
- ✓ Αν δεν ανάβει τίποτα: έλεγξε αν το display είναι κοινής καθόδου (οι καθόδοι στο GND).
- ✓ Αν ανάβουν λάθος segments: έλεγξε τη σειρά των pins σύμφωνα με το δικό σου display.



Μικρό Τεστ Ανατροφοδότησης

Επιλέξτε τη σωστή απάντηση:

1. Τι τύπου display χρησιμοποιούμε στο κύκλωμα;
 - α. Κοινής ανόδου
 - β. Κοινής καθόδου
 - γ. Πολυγράμματο display
 - δ. OLED

2. Πόσα segments ενεργοποιούνται συνολικά για τον αριθμό “8”;
 - α. 5
 - β. 6
 - γ. 7
 - δ. 8

3. Ποιος εντοπισμός αριθμού κάνει ο κώδικας;
 - α. Από 0 έως 9
 - β. Από 1 έως 6
 - γ. Από 1 έως 9
 - δ. Από 0 έως 5

4. Ποια εντολή δημιουργεί τον ήχο;
 - α. beep()
 - β. playTone()
 - γ. tone()
 - δ. buzz()

5. Αν δεν εμφανίζεται τίποτα στην οθόνη, ποια είναι η πιθανότερη αιτία;
 - α. Κακή σύνδεση GND
 - β. Λάθος εντολή delay()
 - γ. Ελλιπής randomSeed()
 - δ. Σφάλμα μεταγλώττισης