



# Προγραμματισμός σε Arduino

---

*Arduino UNO & Innoesys Educational Shield*

[www.devobox.com](http://www.devobox.com)

Ηλεκτρονικά Εξαρτήματα &  
Υλικά Κατασκευής Πρωτοτύπων  
Λέανδρου 79, 10443, Κολωνός  
+30 210 51 55 513, [info@devobox.com](mailto:info@devobox.com)

---

ARDUINO UNO.....	3
INNOESYS EDUCATIONAL SHIELD .....	3
ΠΡΙΝ ΞΕΚΙΝΗΣΟΥΜΕ .....	4
ΣΥΝΔΕΣΗ .....	5
1. ΑΣ ΑΝΑΨΟΥΜΕ ΤΑ LED.....	8
2. ΡΥΘΜΙΖΟΜΕΝΗ/ΑΥΤΟΜΑΤΗ ΦΩΤΕΙΝΟΤΗΤΑ.....	9
3. Ο ΣΥΝΑΓΕΡΜΟΣ .....	11
4. ΑΝΑΨΕ ΤΟΝ ΑΝΕΜΙΣΤΗΡΑ .....	12
5. ΑΝΕΜΙΣΤΗΡΑΣ ΜΕ ΔΙΑΚΟΠΤΗ .....	12
6. ΠΑΤΑ ΜΕ ΝΑ ΔΕΙΣ ΤΙ ΚΑΝΩ!.....	13
7. ΖΕΣΤΗ ΚΑΝΕΙ ΣΗΜΕΡΑ!.....	15

# Arduino UNO

Το Arduino είναι μια μικρή συσκευή (μικροεπεξεργαστής) που συνδέεται με USB στον υπολογιστή. Με την συσκευή αυτή καθηγητές, χομππίστες, μαθητές και φοιτητές μαθαίνουν προγραμματισμό και μπορούν να κατασκευάσουν αυτοματισμούς και εφαρμογές (πχ ρομποτικής).

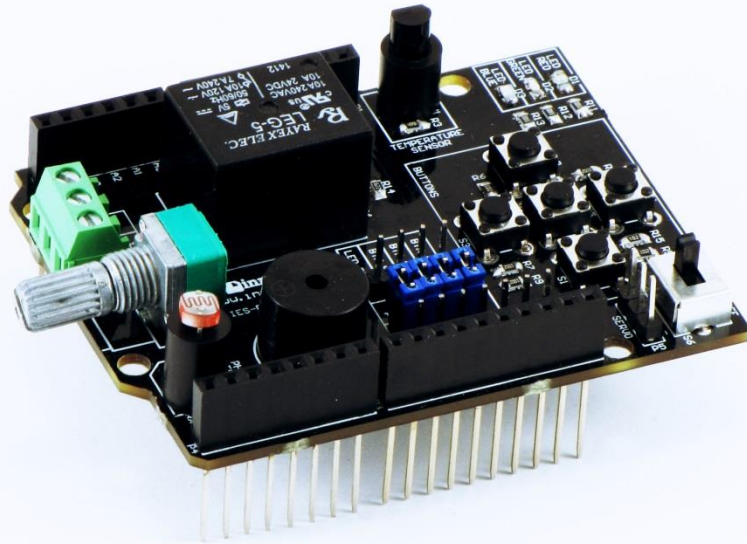


Η πλακέτα του Arduino αποτελείται από έναν μικρό επεξεργαστή ανοικτού κώδικα στον οποίο μπορεί κανείς να προγραμματίσει, ακόμη κι αν είναι αρχάριος, σε περιβάλλον Windows, Linux και MAC OS X. Στην πλακέτα του Arduino UNO βλέπουμε εισόδους και εξόδους στις οποίες μπορούν να συνδεθούν εξωτερικά εξαρτήματα/περιφερειακά ( Led, ποτενσιόμετρα, ρελέ, αισθητήρες και πολλά άλλα) για να μας δώσουν δεδομένα (π.χ. αισθητήρας θερμοκρασίας) ή για να ορίσουμε εμείς δεδομένα (π.χ. άναψε LED)

Η σύνδεση αυτών των περιφερειακών γίνεται με συγκεκριμένο τρόπο για το κάθε εξάρτημα και θέλει ο χρήστης να έχει βασική γνώση θεωρίας ηλεκτρονικών. Σε αυτόν όμως τον οδηγό δεν θα ασχοληθούμε με την σύνδεση των περιφερειακών συσκευών, που σημαίνει ότι μπορούν αρχάριοι χρήστες να κάνουν τα πρώτα τους βήματα εύκολα και απλά. Για τον σκοπό αυτό θα χρησιμοποιήσουμε το Educational Shield της Innoesys

## Innoesys Educational Shield

Το Educational Shield της Innoesys αποτελείται από αρκετά περιφερειακά εξαρτήματα κατάλληλα συνδεδεμένα, έτσι ώστε ο αρχάριος χρήστης να είναι σε θέση να τα χρησιμοποιήσει χωρίς να κάνει συνδεσμολογίες και χωρίς να χρειάζονται γνώσεις θεωρίας ηλεκτρονικών.



Ας δούμε λοιπόν τι περιλαμβάνει το συγκεκριμένο shield.

- LED (R+G+B): Τρία LED διαφορετικού χρώματος συνδεδεμένα στις ψηφιακές εξόδους D6 (Κόκκινο), D9 (Πράσινο) και D10 (Μπλε).
- Buttons: Πέντε buttons συνδεδεμένα στις ψηφιακές εισόδους D7 (S1), D8 (S2), D11 (S3), D12 (S4) και D13 (S5)
- Διακόπτης On-Off: Ένας διακόπτης δύο θέσεων (On-Off) στην είσοδο ψηφιακή D4
- Ρελέ: Ένα ρελέ συνδεδεμένο στην ψηφιακή έξοδο D2
- Ποτενσιόμετρο: Ένα ποτενσιόμετρο συνδεδεμένο στην αναλογική είσοδο A1
- Buzzer: Ένα buzzer για ηχητική σήμανση στην ψηφιακή έξοδο D5
- Φωτοαντίσταση: Μια φωτοαντίσταση στην αναλογική είσοδο A2
- Αισθητήρα θερμοκρασίας: Ένας αισθητήρας θερμοκρασίας περιβάλλοντος στην αναλογική είσοδο A0
- Σύνδεσμος Σερβοκινητήρα: Σύνδεσμο για σερβοκινητήρα στην ψηφιακή έξοδο D3

Αναλυτικά για την πλήρη περιγραφή υλικών και δυνατοτήτων ανατρέξτε στο datasheet της εταιρείας ([www.innoesys.com](http://www.innoesys.com))

## Πριν ξεκινήσουμε

Όπως είπαμε και προηγουμένως το Arduino UNO έχει κάποιες εισόδους/εξόδους. Ανάλογα με το περιφερειακό που θα δουλέψουμε θα πρέπει να ορίσουμε αν είναι είσοδος ή έξοδος. Για παράδειγμα ένας αισθητήρας θερμοκρασίας για να μας δώσει τιμές πρέπει

να συνδεθεί σε «είσοδο», ενώ αντίθετα το LED για να μπορέσουμε να το ανάψουμε θα πρέπει να του δώσουμε κατάσταση, άρα να συνδεθεί σε «έξοδο». Θα το δούμε όμως αναλυτικότερα παρακάτω.

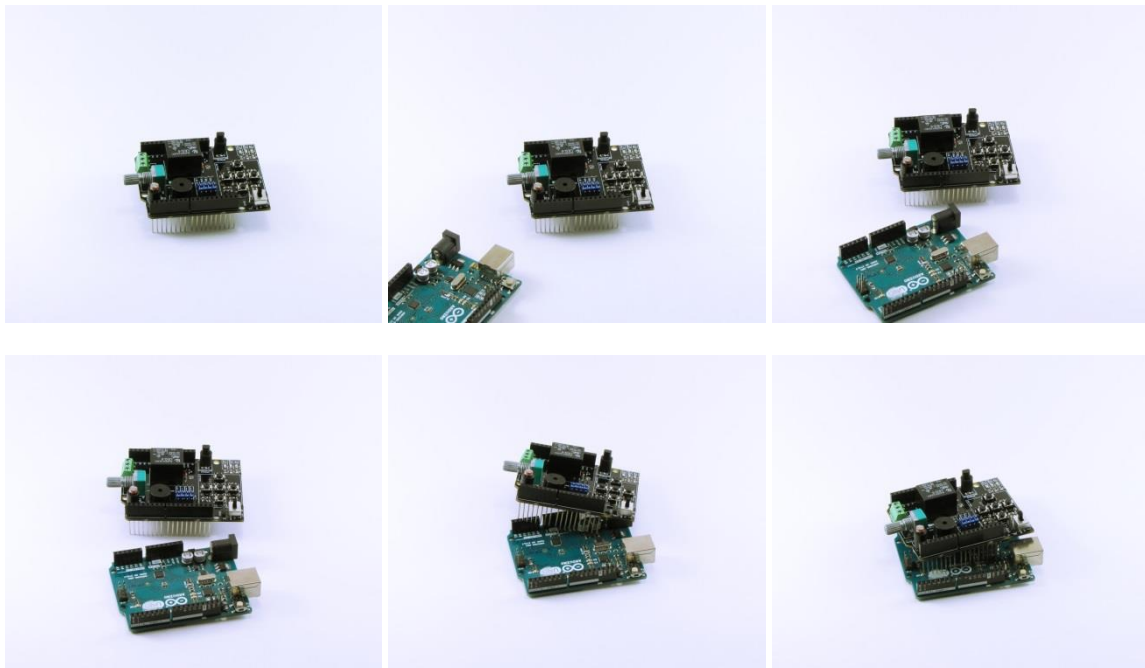
Το Arduino έχει την δυνατότητα να παίρνει από πάνω του πλακέτες (τα λεγόμενα shield) και να μπορεί να τα χρησιμοποιεί απευθείας χωρίς καλώδια και άλλες συνδεσμολογίες. Έτσι λοιπόν το μόνο που χρειαζόμαστε για να ολοκληρώσουμε τα παραδείγματα που ακολουθούν είναι

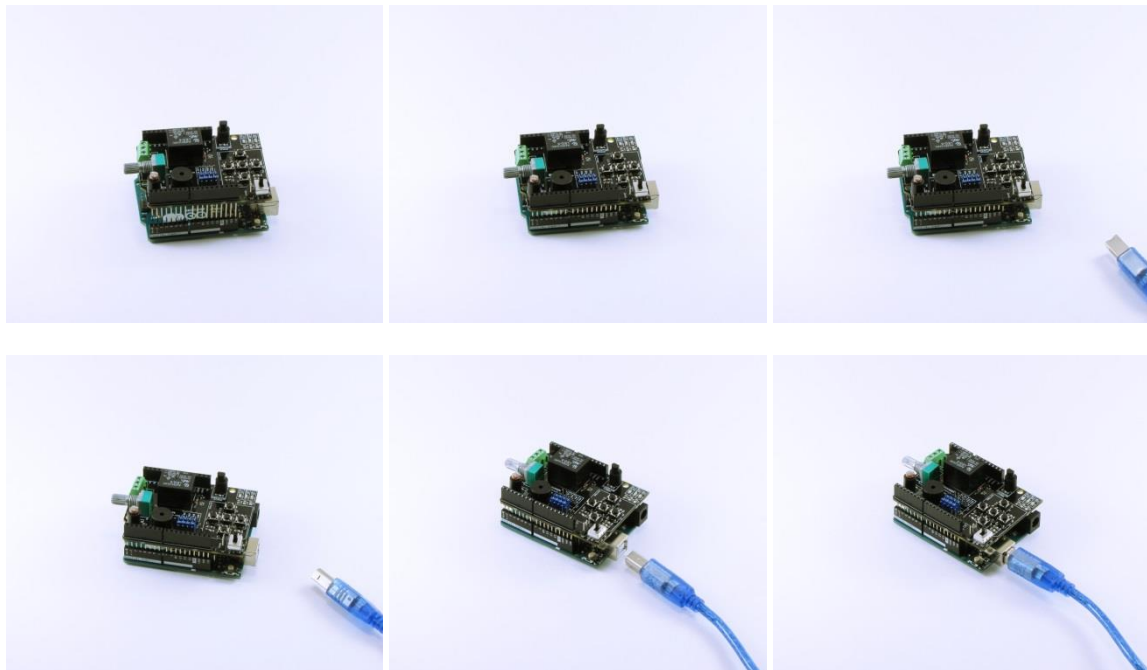
- 1 x [Arduino UNO](#)
- 1 x [USB καλώδιο](#)
- 1 x Innoesys Educational Shield

Για τον προγραμματισμό του Arduino θα χρειαστούμε και το αντίστοιχο περιβάλλον το οποίο μπορείτε να το «κατεβάσετε» από το επίσημο site της εταιρείας (<https://www.arduino.cc/en/Main/Software>) και να το εγκαταστήσετε στον υπολογιστή σας.

## Σύνδεση

Η διαδικασία σύνδεσης είναι πολύ απλή. Τοποθετούμε το Educational Shield επάνω στο Arduino. Φροντίζουμε τα pins να είναι σωστά ευθυγραμμισμένα και πιέζουμε για να «κουμπώσουν» σωστά στις θέσεις τους. Στην συνέχεια απλά συνδέουμε με το USB καλώδιο το Arduino με τον Η/Υ. Δείτε στις παρακάτω φωτογραφίες αναλυτικότερα.

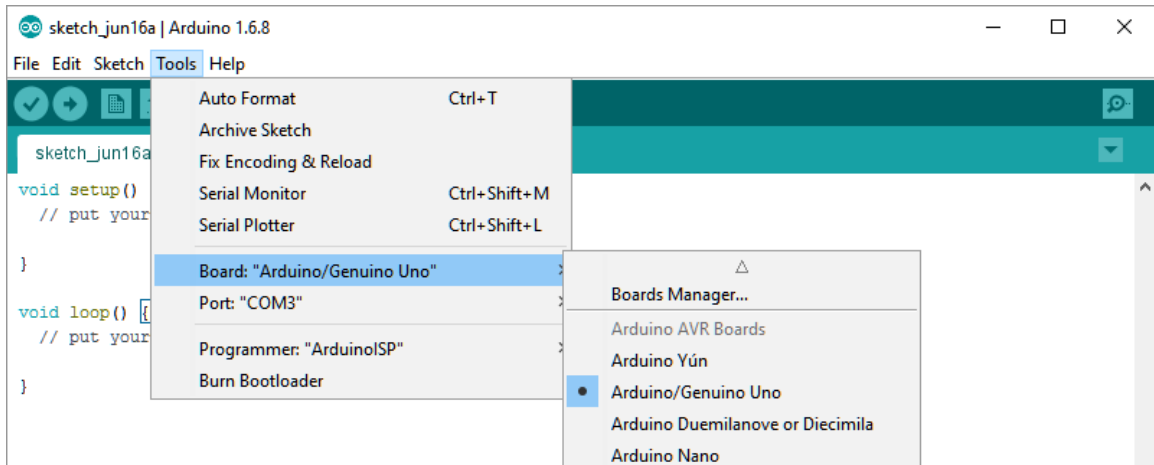




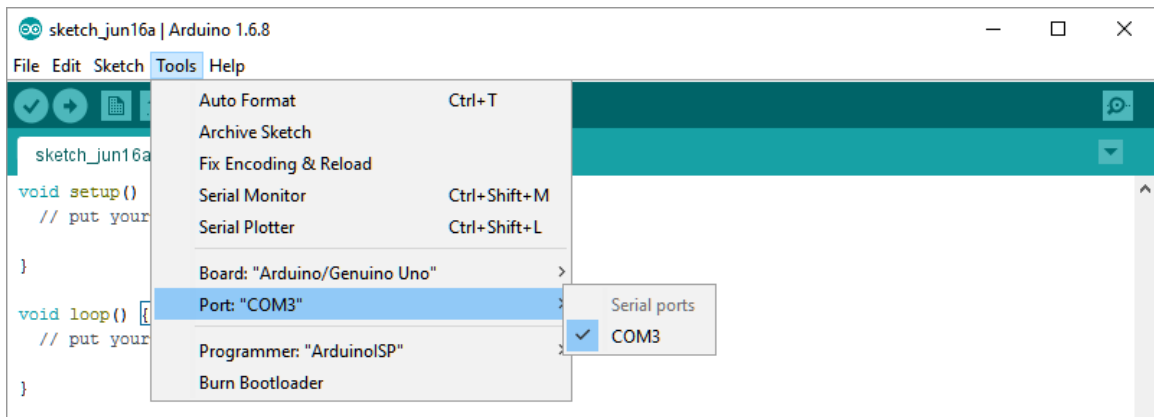
Τέλος ανοίγουμε την εφαρμογή της Arduino που έχουμε εγκαταστήσει και βλέπουμε το παρακάτω παράθυρο. Εδώ παρατηρούμε δύο μέρη στα οποία μπορούμε να γράψουμε τον κώδικά μας. Στο πρώτο μέρος `void setup()` κάνουμε τους ορισμούς μας και τις ρυθμίσεις. Αυτό το μέρος του κώδικα θα «τρέξει» μόνο μία φορά όταν βάλουμε τροφοδοσία στο Arduino. Στο δεύτερο μέρος `void loop()` γράφουμε τον κώδικα που θέλουμε να επαναλαμβάνεται συνεχώς.

```
sketch_jun16a | Arduino 1.6.8
File Edit Sketch Tools Help
sketch_jun16a
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
```

Εδώ το μόνο που χρειάζεται να κάνουμε για να επιβεβαιώσουμε την σωστή σύνδεση του Arduino είναι στο menu → tools → Board και να επιλέξουμε το Arduino UNO



Και στη συνέχεια στο menu → tools → Port να επιλέξουμε την θύρα στην οποία έχει συνδεθεί.



Είμαστε έτοιμοι για προγραμματισμό. Πάμε λοιπόν στα παραδείγματα!

# 1. Ας ανάψουμε τα LED

Το Educational Shield που έχουμε συνδέσει στο Arduino έχει συνδεδεμένα τα LED στις ψηφιακές εξόδους D6 (Κόκκινο), D9 (Πράσινο) και D10 (Μπλε).

Πρώτα λοιπόν θα πρέπει να ορίζουμε αυτές τις ψηφιακές σαν εξόδους. Αυτό γίνεται με την εντολή `pinMode()` και το ορίζουμε στην περιοχή του `void setup()`. Για τη συγκεκριμένη περίπτωση θα πρέπει να βάλουμε:

```
pinMode(6, OUTPUT);  
pinMode(9, OUTPUT);  
pinMode(10, OUTPUT);
```

Τώρα το επόμενο βήμα μας είναι να αναβοσβήσουμε τα LED! Αυτό γίνεται με την εντολή `digitalWrite()` και το ορίζουμε στην περιοχή του `void loop()`, για να επαναλαμβάνεται. Συνεπώς για να αναβοσβήσουμε ένα LED θα πρέπει να βάλουμε:

```
digitalWrite(6, HIGH);  
digitalWrite(6, LOW);
```

Αυτό όμως θα επαναλαμβάνεται πολύ γρήγορα, οπότε θα ήταν καλό να προσθέσουμε και μια μικρή καθυστέρηση με την εντολή `delay()`

Ας δούμε λοιπόν όλο τον κώδικα της δοκιμής μας:

```
void setup() {  
  // put your setup code here, to run once:  
  
  // initialize digital pins  
  pinMode(6, OUTPUT); //Red  
  pinMode(9, OUTPUT); //Green  
  pinMode(10, OUTPUT); //Blue  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
  digitalWrite(6, HIGH);  
  delay(1000); //Delay for 1 sec  
  digitalWrite(6, LOW);  
  delay(1000); //Delay for 1 sec  
}
```

Επιλέγουμε λοιπόν Upload(→) για να ενημερωθούν οι αλλαγές/προσθήκες στο Arduino UNO





Για να δούμε τον κώδικά και για τα υπόλοιπα LED:

```
void setup() {  
  // put your setup code here, to run once:  
  
  // initialize led pins  
  pinMode(6, OUTPUT); //Red  
  pinMode(9, OUTPUT); //Green  
  pinMode(10, OUTPUT); //Blue  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
  digitalWrite(6, HIGH);  
  delay(1000); //Delay for 1 sec  
  digitalWrite(6, LOW);  
  delay(1000); //Delay for 1 sec  
  
  digitalWrite(9, HIGH);  
  delay(1000); //Delay for 1 sec  
  digitalWrite(9, LOW);  
  delay(1000); //Delay for 1 sec  
  
  digitalWrite(10, HIGH);  
  delay(1000); //Delay for 1 sec  
  digitalWrite(10, LOW);  
  delay(1000); //Delay for 1 sec  
}
```

Επιλέγουμε λοιπόν Upload(→) για να ενημερωθούν οι αλλαγές/προσθήκες στο Arduino UNO



## 2. Ρυθμιζόμενη/Αυτόματη φωτεινότητα

Με την βοήθεια του ποτενσιόμετρου, συνδεδεμένο στην αναλογική είσοδο A1, μπορούμε να ελέγχουμε την ένταση ενός LED. Για να το δούμε αναλυτικά.

Πρώτα λοιπόν θα πρέπει να ορίσουμε το LED σαν έξοδο και το ποτενσιόμετρο σαν είσοδο

```
pinMode(6, OUTPUT); //Red  
pinMode(A1, INPUT);
```

και στην συνέχεια θα πρέπει να διαβάσουμε την τιμή του ποτενσιόμετρου με την εντολή `analogRead()` και να την γράψουμε στο LED με την εντολή `analogWrite()`

```
analogWrite(6, (analogRead(A1) / 4));
```

\*\*\* Σημαντική σημείωση: Οι τιμές που παίρνουμε από το ποτενσιόμετρο είναι 0 -1023 ενώ οι τιμές που μπορούμε να γράψουμε στην έξοδο για το LED είναι 0 – 255. Για τον λόγο αυτό κάνουμε μια διαίρεση με το 4.

Οπότε ο κώδικάς μας είναι ο εξής:

```
void setup() {  
  // put your setup code here, to run once:  
  
  // initialize led pins  
  pinMode(6, OUTPUT); //Red  
  // initialize potentiometer  
  pinMode(A1, INPUT);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
  analogWrite(6, (analogRead(A1) / 4));  
}
```

Το ίδιο μπορούμε να κάνουμε και με την φωτοαντίσταση. Ας προσθέσουμε λοιπόν το μπλε LED, που είναι στην έξοδο 10 και την φωτοαντίσταση στην είσοδο A2

```
pinMode(10, OUTPUT); //Blue  
pinMode(A2, INPUT);
```

και με την κατάλληλη εντολή διαβάσματος των τιμών της φωτοαντίστασης

```
analogWrite(10, (analogRead(A2) / 4));
```

\*\*\* Σημαντική σημείωση: Οι τιμές που παίρνουμε από την φωτοαντίσταση είναι 0 -1023 ενώ οι τιμές που μπορούμε να γράψουμε στην έξοδο για το LED είναι 0 – 255. Για τον λόγο αυτό κάνουμε μια διαίρεση με το 4.

έχουμε τον παρακάτω κώδικα

```
void setup() {  
  // put your setup code here, to run once:  
  
  // initialize led pins  
  pinMode(6, OUTPUT); //Red  
  pinMode(10, OUTPUT); //Blue  
  
  // initialize potentiometer  
  pinMode(A1, INPUT);  
  
  // initialize photoresistor  
  pinMode(A2, INPUT);
```

```

}

void loop() {
  // put your main code here, to run repeatedly:

  analogWrite(6, (analogRead(A1) / 4));
  analogWrite(10, (analogRead(A2) / 4));
}

```

Επιλέγουμε λοιπόν Upload(→) για να ενημερωθούν οι αλλαγές/προσθήκες στο Arduino UNO



### 3. Ο Συναγερμός

Σε αυτό το παράδειγμα θα δούμε την ηχητική σήμανση του Buzzer, που είναι συνδεδεμένο στην ψηφιακή έξοδο D5

Πρώτα ορίζουμε την ψηφιακή έξοδο

```
pinMode(5, OUTPUT);
```

και στην συνέχεια με την εντολή `tone()` μπορούμε να ορίσουμε τον τόνο με τον οποίο θα ηχήσει το buzzer.

```
tone(5, 150, 500);
```

*\*Έξοδος buzzer, Τόνος (0-255), διάρκεια τόνου*

Εφόσον ο τόνος θα ηχήσει για ½ του δευτερολέπτου τότε προσθέτουμε και μια καθυστέρηση 1 δευτερολέπτου για να δημιουργήσουμε διακοπτόμενο ήχο και ο συναγερμός μας έχει πλέον τον κώδικα:

```

void setup() {
  // put your setup code here, to run once:

  // initialize buzzer pin
  pinMode(5, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:

  tone(5, 150, 500);
  delay(1000); //delay for 1 sec
}

```

Επιλέγουμε λοιπόν Upload(→) για να ενημερωθούν οι αλλαγές/προσθήκες στο Arduino UNO



## 4. Άναψε τον ανεμιστήρα

Στην ψηφιακή έξοδο D2 του Educational Shield είναι συνδεδεμένο ένα ρελέ. Σε αυτό μπορούμε να δώσουμε εντολή να ανοίξει ή να κλείσει. Για να το κάνουμε αυτό πρώτα πρέπει να ορίσουμε την έξοδό του

```
pinMode(2, OUTPUT);
```

και στην συνέχεια με την εντολή `digitalWrite()` να του αλλάξουμε την κατάσταση

```
digitalWrite(6, HIGH); ή digitalWrite(6, LOW);
```

Ας προσθέσουμε λοιπόν στον κώδικά μας μια εναλλαγή κατάστασης με χρονοκαθυστέρηση 5 δευτερολέπτων. Έτσι το πρόγραμμα έχει ως εξής:

```
void setup() {
  // put your setup code here, to run once:

  // initialize relay pin
  pinMode(2, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:

  digitalWrite(2, HIGH);
  delay(5000);
  digitalWrite(2, LOW);
  delay(5000);
}
```

\*\*\* Αν συνδέσουμε έναν ανεμιστήρα στις κλέμες που καταλήγει το ρελέ, τότε αυτό θα ανοίγει και θα κλείνει ανάλογα με τον χρόνο που θέσαμε στο πρόγραμμα. Συμβουλευτείτε το *datasheet* του Innoesys Educational Shield για την συνδεσμολογία.

Επιλέγουμε λοιπόν Upload(→) για να ενημερωθούν οι αλλαγές/προσθήκες στο Arduino UNO



## 5. Ανεμιστήρας με διακόπτη

Ένας διακόπτης δύο θέσεων (On-Off) είναι συνδεδεμένος στην ψηφιακή D4. Για να τον χρησιμοποιήσουμε θα πρέπει να τον ορίσουμε σαν είσοδο.

```
pinMode(4, INPUT);
```

Διαβάζοντας την κατάσταση του με την εντολή `digitalRead()` μπορούμε να δώσουμε εντολή να γίνει κάποια λειτουργία. Ας πάρουμε τον κώδικα του προηγούμενου παραδείγματος με τον ανεμιστήρα. Σε αυτόν θα προσθέσουμε τον ορισμό την ψηφιακής εισόδου:

```
void setup() {
    // put your setup code here, to run once:

    // initialize relay pin
    pinMode(2, OUTPUT);

    // initialize on-off switch
    pinMode(4, INPUT);
}
```

Αντί λοιπόν να ανοιγοκλείνουμε κάθε 5 δευτερόλεπτα το ρελέ, μπορούμε να δώσουμε απευθείας την κατάσταση του διακόπτη στην ψηφιακή έξοδο του ρελέ με την εντολή:

```
digitalWrite(2, digitalRead(4));
```

Έτσι το πρόγραμμά θα γίνει ως εξής:

```
void setup() {
    // put your setup code here, to run once:

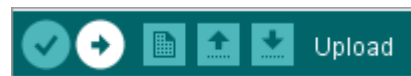
    // initialize relay pin
    pinMode(2, OUTPUT);

    // initialize on-off switch
    pinMode(4, INPUT);
}

void loop() {
    // put your main code here, to run repeatedly:

    digitalWrite(2, digitalRead(4));
}
```

Επιλέγουμε λοιπόν Upload(→) για να ενημερωθούν οι αλλαγές/προσθήκες στο Arduino UNO



## 6. Πάτα με να δεις τι κάνω!

Στο Educational Shield υπάρχουν πέντε buttons συνδεδεμένα στις ψηφιακές εισόδους D7 (S1), D8 (S2), D11 (S3), D12 (S4) και D13 (S5), που μπορούμε να ορίσουμε τι θα κάνουν το καθένα από αυτά. Στο παράδειγμα αυτό θα βάλουμε να ανάβουν τα LED, να ανοίγουν το ρελέ και να δίνουν και μια ηχητική σήμανση.

Πρώτα πρέπει να ορίσουμε τις εισόδους των button

```
pinMode(7, INPUT); //S1
pinMode(8, INPUT); //S2
pinMode(11, INPUT); //S3
pinMode(12, INPUT); //S4
pinMode(13, INPUT); //S5
```

μετά τις εξόδους των LED

```
pinMode(6, OUTPUT); //Red
pinMode(9, OUTPUT); //Green
pinMode(10, OUTPUT); //Blue
```

Την έξοδο του ρελέ

```
pinMode(2, OUTPUT);
```

Και τέλος την έξοδο του buzzer

```
pinMode(5, OUTPUT);
```

Αφού ολοκληρώσουμε τις αρχικές ρυθμίσεις θα πρέπει να δώσουμε τις τιμές των button στις αντίστοιχες εξόδους. Έτσι έχουμε:

```
digitalWrite(6, digitalRead(7)); //Button S1 -> LED Red
digitalWrite(9, digitalRead(8)); //Button S2 -> LED Green
digitalWrite(10, digitalRead(11)); //Button S3 -> LED Blue
digitalWrite(2, digitalRead(12)); //Button S4 -> Relay
digitalWrite(5, digitalRead(13)); //Button S5 -> Buzzer
```

Και συνολικά ο κώδικας είναι ο εξής:

```
void setup() {
  // put your setup code here, to run once:

  // initialize the buttons
  pinMode(7, INPUT); //S1
  pinMode(8, INPUT); //S2
  pinMode(11, INPUT); //S3
  pinMode(12, INPUT); //S4
  pinMode(13, INPUT); //S5

  // initialize the LED pins
  pinMode(6, OUTPUT); //Red
  pinMode(9, OUTPUT); //Green
  pinMode(10, OUTPUT); //Blue

  // initialize relay pin
  pinMode(2, OUTPUT);

  // initialize the Buzzer
  pinMode(5, OUTPUT);
}
```

```

}

void loop() {
  // put your main code here, to run repeatedly:

  digitalWrite(6, digitalRead(7)); //Button S1 -> LED Red
  digitalWrite(9, digitalRead(8)); //Button S2 -> LED Green
  digitalWrite(10, digitalRead(11)); //Button S3 -> LED Blue
  digitalWrite(2, digitalRead(12)); //Button S4 -> Relay
  digitalWrite(5, digitalRead(13)); //Button S5 -> Buzzer
}

```

Επιλέγουμε λοιπόν Upload(→) για να ενημερωθούν οι αλλαγές/προσθήκες στο Arduino UNO



## 7. Ζέστη κάνει σήμερα!

Στην αναλογική είσοδο A0 βρίσκεται ένας αισθητήρας θερμοκρασίας. Με αυτόν είμαστε σε θέση να μετρήσουμε την θερμοκρασία του περιβάλλοντος.

Για να βλέπουμε τις τιμές που παίρνουμε από τον αισθητήρα θερμοκρασίας θα χρησιμοποιήσουμε την σειριακή έξοδο του Arduino. Αυτό που πρέπει να κάνουμε αρχικά είναι να ορίσουμε την σειριακή και την είσοδο του αισθητήρα με τις εντολές:

```

Serial.begin(9600);
pinMode(A0, INPUT);

```

Εφόσον έχουν αρχικοποιηθεί οι ρυθμίσεις μας το μόνο που χρειάζεται να κάνουμε είναι να διαβάσουμε από τον αισθητήρα τις τιμές με την εντολή `analogRead()` και να τις «τυπώνουμε» στην σειριακή με την εντολή `Serial.println()`.

```

Serial.println((analogRead(0) * 0.004882812 * 100) - 273.15);

```

\*\*\* Οι υπολογισμοί είναι απαραίτητοι για να μετατραπούν τα δεδομένα σε βαθμούς κελσίου.

Έτσι το πρόγραμμά μας έχει ως εξής:

```

void setup() {
  // put your setup code here, to run once:

  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);

  // initialize the temperature Sensor

```

```
pinMode(A0, INPUT);  
  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
  // show the values of the temperature screen in the serial monitor  
  Serial.println((analogRead(0) * 0.004882812 * 100) - 273.15);  
  delay(1000); //delay for 1 sec  
}
```

Επιλέγουμε λοιπόν Upload(→) για να ενημερωθούν οι αλλαγές/προσθήκες στο Arduino UNO



Για να δούμε το αποτέλεσμα πρέπει από το menu → Tools να επιλέξουμε Serial Monitor. Στο παράθυρο που ανοίξει εμφανίζονται οι τιμές του αισθητήρα.