

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΦΥΣΙΚΗΣ
ΑΥΤΟΜΑΤΟΠΟΙΗΣΗ ΑΙΣΘΗΤΗΡΙΩΝ ΣΥΣΤΗΜΑΤΩΝ

2^η ΕΝΟΤΗΤΑ

Εισαγωγή στο Σχεδιασμό Κυκλωμάτων με Arduino

Κωνσταντίνος Τσίκνας

ktsik@teiemt.gr

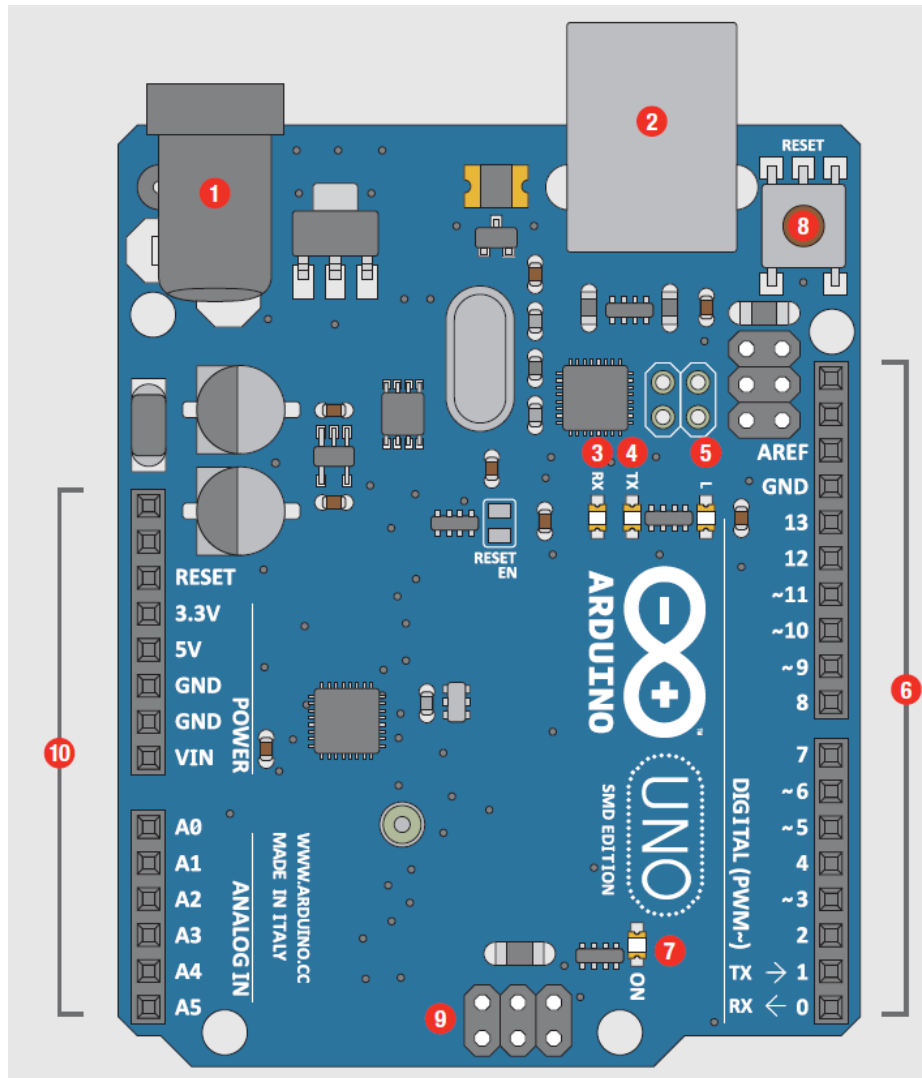
Τί είναι το Arduino

- Ηλεκτρονική πλατφόρμα ανοικτού κώδικα και σχεδιασμού, που βασίζεται σε ευέλικτο και εύκολο στη χρήση υλικό και λογισμικό.



- Υπάρχει ένα κύκλωμα που χρησιμοποιεί μικροελεγκτή, το οποίο μας δίνει ένα αριθμό πυλών οι οποίες μπορεί να λειτουργήσουν είτε ως εισοδοι είτε ως εξοδοι στα κυκλώματά μας.
- Αυτές τις εισόδους ή εξόδους μπορούμε να τις διαχειριστούμε γράφοντας κώδικα στο περιβάλλον προγραμματισμού Arduino IDE που έχει βασιστεί στη γλώσσα C/C++.

Δομικά μέρη της πλακέτας



Κάτοψη πλακέτας Arduino Uno

1. Εξωτερική τροφοδοσία (Power)

2. Θύρα USB. Για την επικοινωνία με τον υπολογιστή και την τροφοδοσία της πλακέτας.

3, 4, 5. LED (Tx, Rx, Troubleshooting)

6. Pin **ψηφιακής** εισόδου/εξόδου (0-13)

Βασικοί ακροδέκτες ψηφιακής επικοινωνίας με εξωτερικές συσκευές

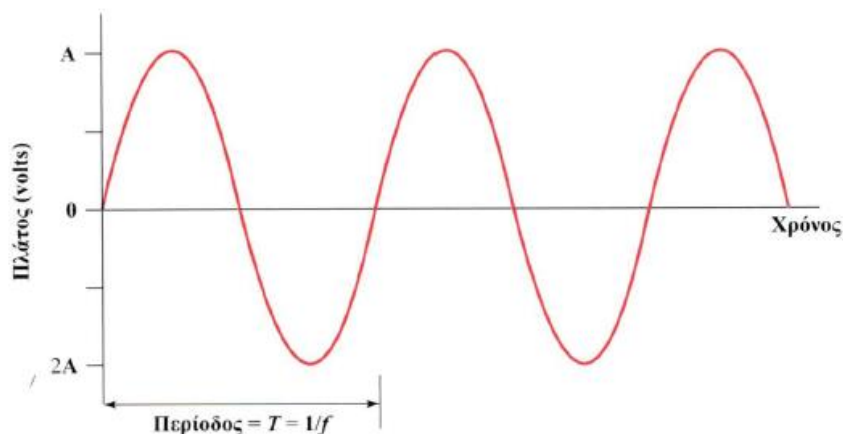
7. LED τροφοδοσίας

8. Reset button

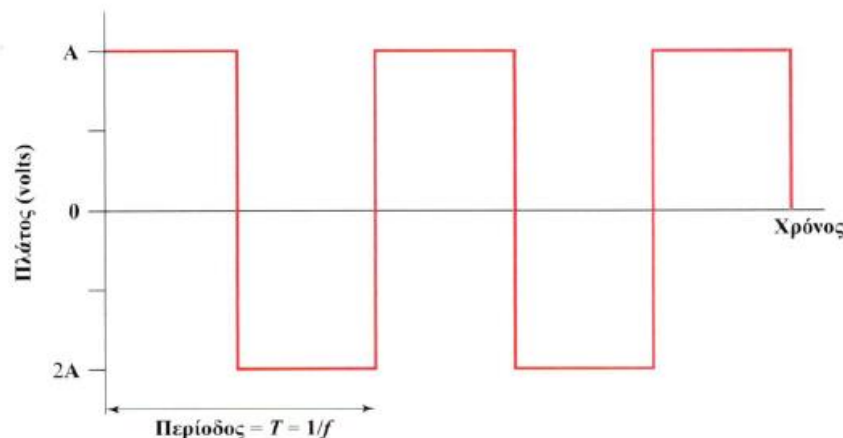
10. Pin **αναλογικής** εισόδου (A0-A5)

Διάβασμα αναλογικών σημάτων, τροφοδοσίας (5V, 3V) και γείωσης (GND).

Αναλογικό – ψηφιακό σήμα



(α) ημιτονοειδής κυματομορφή



(β) τετραγωνική κυματομορφή

- **Αναλογικό σήμα:** Συνεχής κυματομορφή
- **Ψηφιακό σήμα:** Διακριτές τιμές (0 V, 5 V)

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΣΗΜΑΤΟΣ

- **Πλάτος (A):** μέγιστη τιμή του σήματος
- **Περίοδος (T):** ο ελάχιστος χρόνος που απαιτείται για να εκτελεστεί ένας πλήρης κύκλος του σήματος (για περιοδικά σήματα)
- **Συχνότητα (f)** ο αριθμός των επαναλήψεων ανά sec
- **Φάση (φ):** η σχετική θέση στο χρόνο
- **Μήκος κύματος (λ):** Η απόσταση που καλύπτεται από ένα κύκλο. Ισχύει

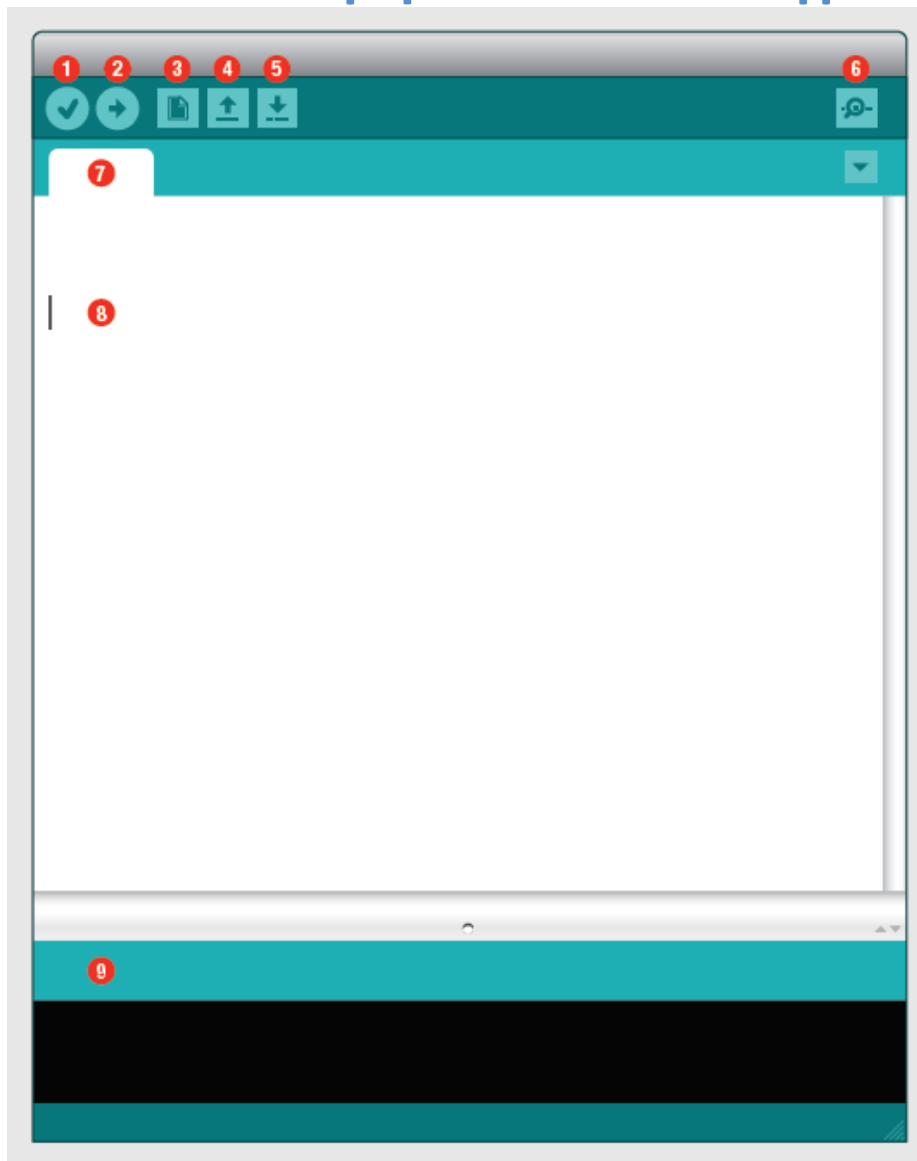
$\lambda = c/f$, όπου c η ταχύτητα μεάδοσης του κύματος $c = 3 \times 10^8 \text{ m/s}$

Για την ημιτονοειδή κυματομορφή του σχήματος ισχύει **$S(t) = A\eta\mu(2\pi ft + \varphi)$**

Ψηφιακές - Αναλογικές είσοδοι - έξοδοι

- Για **ψηφιακή είσοδο**, χρησιμοποιούμε τις 14 ψηφιακές **0..13**. Όταν δουλεύουν ψηφιακά, η είσοδος μπορεί να είναι ή 0 ή 5V, με τον χαρακτηρισμό LOW ή HIGH
- Για **ψηφιακή έξοδο**, χρησιμοποιούμε τις 14 ψηφιακές **0..13**. Όταν δουλεύουν ψηφιακά, η έξοδος μπορεί να είναι 0 ή 5V, με τον χαρακτηρισμό LOW ή HIGH
- Για **αναλογική είσοδο**, δηλαδή να διαβάσουμε τιμές ρεύματος στο διάστημα 0 έως 5V, χρησιμοποιούμε τις έξι αναλογικές θύρες **A0..A5**.
- Για **αναλογική έξοδο**, μπορούμε να χρησιμοποιήσουμε τις έξι **PWM** ψηφιακές θύρες (**3, 5, 6, 9, 10, 11**).

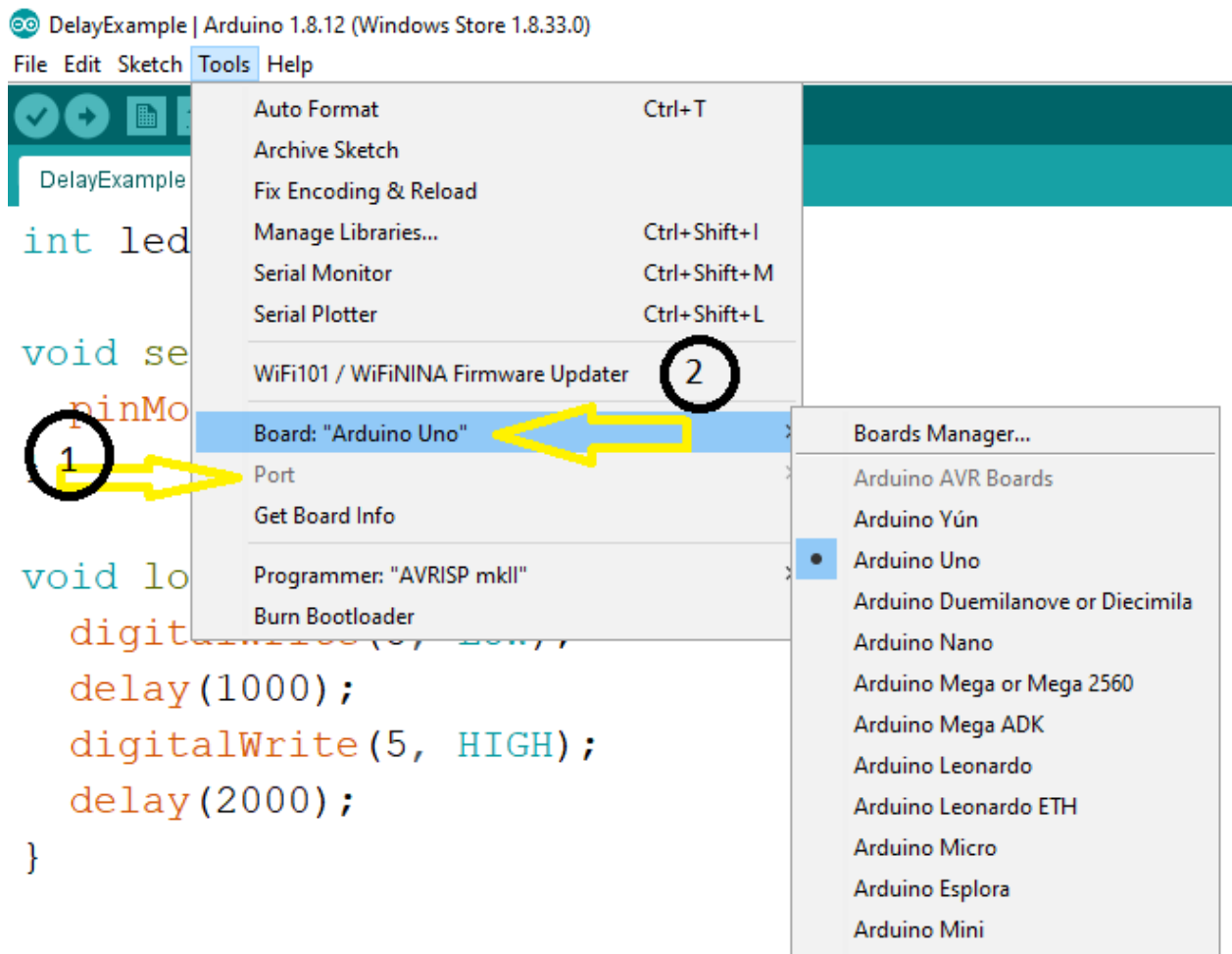
Το λειτουργικό σύστημα



- Είναι ελεύθερο λογισμικό
<http://arduino.cc/en/Guide/Windows>
- Βασικά κουμπιά
 1. **Verify** Μεταφράζει τον κώδικα και ελέγχει για λάθη στη σύνταξη
 2. **Upload**: Φορτώνει τον κώδικα στην πλακέτα
 3. **New**: Δημιουργία νέου εγγράφου
 4. **Open**: Ανοίγμα υφιστάμενου εγγράφου
 5. **Save**: Σώσιμο
 6. **Serial monitor**: Στένει μηνύματα μέσω της σειριακής θύρας. Χρήσιμο για debugging
 7. **Sketch name**: Το όνομα του τρέχοντος εγγράφου
 3. **Code area**: Η περιοχή που γράφεται ο κώδικας
 9. **Message area**: Εδώ εμφανίζονται τα μηνύματα λάθους

Τα δύο πρώτα βασικά βήματα

- Αφού έχει συνδεθεί η πλακέτα στον υπολογιστή
ΒΗΜΑ 1º: επιλέγουμε τη θύρα (συνήθως ανιχνεύεται αυτόματα)
ΒΗΜΑ 2º: Επιλέγουμε την κατάλληλη έκδοση Arduino (Uno, Nano, Mega κλπ)



Τυπική μορφή προγράμματος Wiring C

```
void setup() {  
  
    //θέτει το Pin 10 του Arduino ως είσοδο  
    pinMode(10, INPUT);  
  
}  
  
void loop()  
  
    //διαβάζει την ψηφιακή είσοδο  
    int digitalInput = digitalRead(10);  
  
}
```

- **setup()** - μπαίνουν όλες τις εντολές που πρέπει να τρέξουν μία φορά, όταν ενεργοποιείται η μονάδα
 - Όταν δηλαδή δίνουμε ρεύμα ή όταν πατηθεί το πλήκτρο reset.
- **loop()** - εδώ γράφεται το πρόγραμμα. Οι εντολές που υπάρχουν θα τρέξουν κι όταν φτάσει στο τέλος θα ενεργοποιηθεί ξανά η loop(), συνεχίζοντας από την αρχή της, και ξανά.
 - Αυτό θα συμβαίνει συνεχώς, όσο έχει ρεύμα το Arduino ή μέχρι να πατηθεί το πλήκτρο reset.

Η συνάρτηση pinMode(pin, mode)

- **pinMode(Pin, Mode):** Θέτει μια θύρα (pin) σε κατάσταση εισόδου ή εξόδου
Δέχεται δύο ορίσματα:
 - α. **Pin:** Ο αριθμός θύρας (**Pin**) και
 - β. **Mode:** Η κατάσταση λειτουργίας: **INPUT** (είσοδος) ή **OUTPUT**(έξοδος).
 - Τα 14 ψηφιακά pins του Arduino μπορούν δουλεύουν ως ψηφιακές έξοδοι ή είσοδοι, δηλαδή να “θέτουν” ως έξοδο μια τάση με τιμή είτε **0 V** είτε **5 V** ή να τη “διαβάζουν” ως είσοδο.
- Οι συναρτήσεις εισόδου – εξόδου καλούνται εντός της setup() προκειμένου να εκτελεστούν μόνο μία φορά.

```
void setup() {  
  
    //θέτει το Pin 5 του Arduino ως έξοδο  
    pinMode(5, OUTPUT);  
  
    //θέτει το Pin 10 του Arduino ως έξοδο  
    pinMode(10, INPUT);  
  
}
```

Οι συναρτήσεις **digitalRead(pin)** και **analogRead(pin)**

- **digitalRead(pin)**: Διαβάζει και επιστρέφει την τιμή μιας **ψηφιακής** θύρας (0 .. 13)
- **analogRead(pin)**: Διαβάζει και επιστρέφει την τιμή μιας **αναλογικής** θύρας (A0 .. A5)

Δέχονται ένα όρισμα:

Pin: Ο αριθμός θύρας (**0..13**)

- *Η ψηφιακή θύρα θα πρέπει να έχει ορισθεί εντός της **setup()** ως θύρα ΕΙΣΟΔΟ*
- *Η αναλογική θύρα Δεν χρειάζεται να έχει ορισθεί προηγουμένως ως θύρα εισόδου .
Λειτουργεί ως τέτοια εξ' ορισμού.*

```
void setup() {  
  
    //θέτει το Pin 10 του Arduino ως έξοδο  
    pinMode(10, INPUT);  
  
}  
  
void loop() {  
  
    int digitalInput = digitalRead(10);  
  
    int analogInput = analogRead(A0);  
  
}
```

Παράδειγμα χρήσης των **digitalRead()** και **analogRead()**

Η συνάρτηση `digitalWrite(pin, Value)`

- Θέτει μια τιμή (τάση) σε μια **ψηφιακή** θύρα (0 .. 13)

Δέχονται δύο ορίσματα

- **Pin:** Ο αριθμός της ψηφιακής θύρας (0 .. 13)
- **Value:** Τιμή 0 Volt (LOW) ή 5 Volt (HIGH)

Η ψηφιακή θύρα θα πρέπει να έχει ορισθεί εντός της `setup()` ως θύρα ΕΞΟΔΟΥ

```
void setup() {  
  
    //θέτει τα Pin 5, 10 του Arduino ως έξοδο  
    pinMode(5, OUTPUT);  
    pinMode(10, OUTPUT);  
  
}  
  
void loop() {  
  
    //θέτω τάση 0 Volt (LOW) στο pin 5 και 5 Volt (HIGH) στο pin 10  
    int digitalOut1 = digitalWrite(5, LOW);  
    int digitalOut2 = digitalWrite(10, HIGH);  
  
}
```

Παράδειγμα χρήσης της `digitalWrite()`

Η συνάρτηση `delay(time)`

- Δέχονται ένα όρισμα:
time: Ακέραια τιμή - χρόνος σε msec
- Δημιουργεί μια χρονοκαθυστέρηση στο κύκλωμα, τόση όση ο χρόνος που δέχεται σαν όρισμα

```
void setup() {  
    pinMode(5, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(5, LOW);  
    delay(1000);  
    digitalWrite(5, HIGH);  
    delay(2000);  
}
```

Παράδειγμα χρήσης της `delay()`

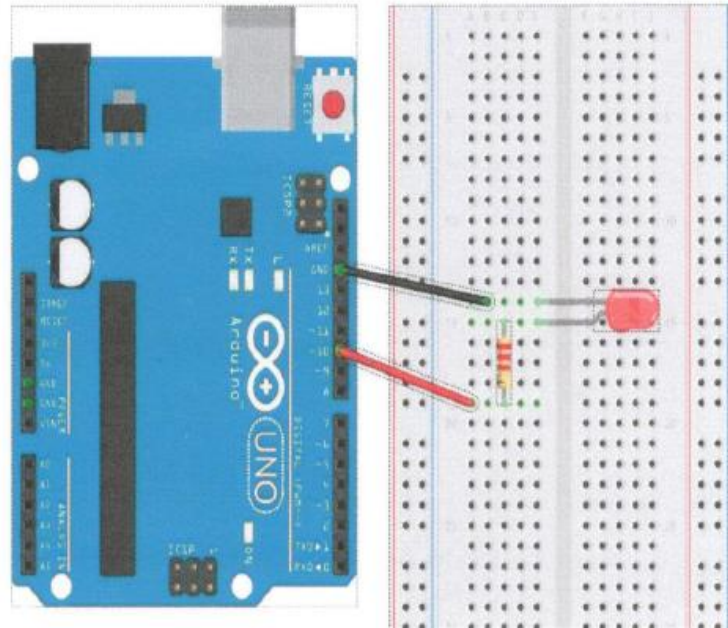
Πρόγραμμα για το αναβόσβημα LED

```
void setup() {  
  
    //  
    pinMode(5, OUTPUT);  
}  
  
void loop() {  
  
    //  
    digitalWrite(5, HIGH);  
  
    //  
    delay(2000);  
  
    //  
    digitalWrite(5, LOW);  
  
    //  
    delay(2000);  
}
```

Το πρόγραμμα του διπλανού σχήματος αρχικά θέτει την ψηφιακή θύρα (pin) **5** ως θύρα εξόδου και κατόπιν θέτει τάση 5 Volt (**High**) και 0 Volt (**Low**)

Ενδιάμεσα εφαρμόζει καθυστέρηση (delay) 2 Δευτερολέπτων.

Εάν η θύρα 5 οδηγηθεί σε ένα led τότε αυτό θα Αναβοσβήνει κάθε δύο sec.



Δηλώσεις μεταβλητών - σταθερών

- **boolean**: με τιμές το 0 και 1 (ή True – False)
- **int**: ακέραιος (από -32768 έως και 32767)
- **long**: ακέραιος (-2147483648 έως και 2147483647)
- **float**: δεκαδικός αριθμός
- **char**: ένας χαρακτήρας (μεγέθους ενός byte)

Δηλώσεις σταθερών: Τοποθετώντας τη λέξη **const** (constant) στην αρχή

Σταθερές



```
const int ledPin = 5;  
  
const int delay = 2000;  
  
float inpVoltage = 2.2  
  
bool isLedOn = true;  
  
char YesNo = 'Y';
```

Δηλώσεις μεταβλητών - σταθερών. Οι σταθερές (**const**) δεν μπορούν να αλλάξουν τιμή μέσα στο πρόγραμμα)

Το πρόγραμμα με δηλώσεις μεταβλητών - σταθερών

```
void setup() {  
  
    //  
    pinMode(5, OUTPUT);  
}  
  
void loop() {  
  
    //  
    digitalWrite(5, HIGH);  
  
    //  
    delay(2000);  
  
    //  
    digitalWrite(5, LOW);  
  
    //  
    delay(2000);  
  
}
```

```
const int ledPin = 5;  
  
const int delay = 2000;
```



Δηλώσεις

```
void setup() {  
  
    //  
    pinMode(ledPin, OUTPUT);  
}  
  
void loop() {  
  
    //  
    digitalWrite(ledPin, HIGH);  
  
    //  
    delay(delay);  
  
    //  
    digitalWrite(ledPin, LOW);  
  
    //  
    delay(delay);  
  
}
```

Ερωτήματα

- Θα μπορούσαν να δηλωθούν εντός της `setup()` ή εντός της `loop()`; Αν ναι τότε γιατί δηλώθηκαν πάνω από τη `setup()`;
- Για ποιο λόγο χρησιμοποιήθηκαν σταθερές έναντι μεταβλητών;
- Σε τι διευκολύνει η χρήση μεταβλητών – σταθερών στο πρόγραμμα;