

# 2

## Εφαρμογές για Αρχάριους

### ΚΕΦΑΛΑΙΟ - 2

Η ανάπτυξη οποιασδήποτε εφαρμογής προϋποθέτει την εξοικείωση με ορισμένες βασικές έννοιες όπως είναι το κύκλωμα, το ρεύμα, η τάση, το ψηφιακό σήμα, κλπ. Το καλό είναι ότι χρειάζεται να γνωρίζουμε ελάχιστα πράγματα για να ξεκινήσουμε. Αυτό είναι άλλωστε και το πλεονέκτημα του Arduino. Σε αυτό το κεφάλαιο θα γνωρίσουμε το απλούστερο δυνατό κύκλωμα που περιλαμβάνει ένα LED και μια αντίσταση. Στόχος είναι να αναλύσουμε τόσο τη λειτουργία του κυκλώματος όσο και τον τρόπο που μπορούμε να το ελέγξουμε μέσω του Arduino.

### 2.1 Εφαρμογή 1 Γνωριμία με τη δίοδο LED

**Τι είναι το LED;**

Η δίοδος LED ή απλά το LED (εικόνα 2.1) είναι ένας ημιαγωγός που όταν τον πολώσουμε ορθά φωτοβολεί (βγάζει φως). Πιο απλά, πρόκειται για ένα ηλεκτρονι-



Εικόνα 2.1 Το LED

κό εξάρτημα με δύο ακροδέκτες (+,-) που όταν συνδεθούν σε μια πηγή βγάζει φως. Ο θετικός ακροδέκτης του LED έχει ελαφρώς μεγαλύτερο μήκος ώστε να είναι εύκολο να τον αναγνωρίσουμε.

### Πηγή ενέργειας

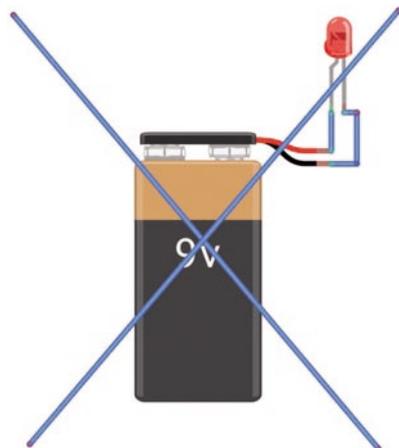
Ως πηγή ενέργειας μπορεί να χρησιμοποιηθεί μια απλή μπαταρία (εικόνα 2.2) ή μια ηλεκτρονική διάταξη (π.χ. τροφοδοτικό) που μπορεί να δώσει στην έξοδό της συνεχή τάση (DC).



Εικόνα 2.2 Μπαταρία

### Σύνδεση LED στην πηγή;

Η παρακάτω σύνδεση (εικόνα 2.3) είναι λανθασμένη παρά το γεγονός ότι έχει συνδεθεί το (-) του LED στο (-) της πηγής και το (+) στο (+) της πηγής, επειδή απαιτείται περιορισμός του ρεύματος.



Εικόνα 2.3 LED με μπαταρία

### Τι ρεύμα διαρρέει το LED;

Με την παραπάνω σύνδεση θα διαρρέει το LED σχεδόν το μέγιστο ρεύμα που μπορεί να δώσει η πηγή αφού δεν υπάρχει κάποιος περιορισμός σε αυτό (το LED

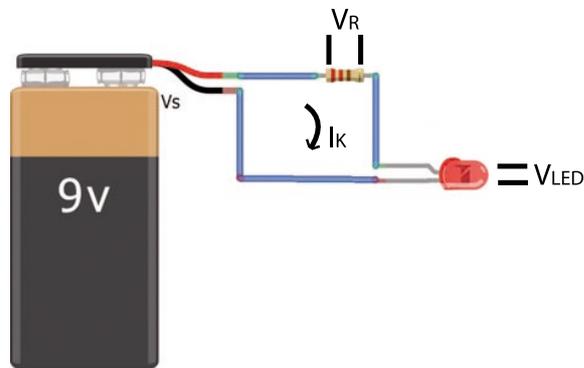
παρουσιάζει μικρή αντίσταση). Το αποτέλεσμα είναι η καταστροφή τουλάχιστον του LED.

### Τι κάνει το LED να ανάβει;

Η πηγή τάσης δημιουργεί το ρεύμα. Όταν το LED διαρρέεται από ρεύμα τότε ανάβει (φωτοβολεί).

### Πώς θα περιορίσω το ρεύμα;

Το ρεύμα περιορίζεται με αντιστάσεις (εικόνα 2.4). Η αντίσταση μετριέται σε  $\Omega\mu$  (συμβολίζεται με  $\Omega$ ).



Εικόνα 2.4 Κύκλωμα με αντίσταση και LED

### Η πρώτη ανάλυση

Η τάση της πηγής  $V_S$  διαμοιράζεται στα στοιχεία που είναι συνδεδεμένα. Θα ισχύει δηλαδή ότι:

$$V_S = V_R + V_{LED}$$

( $V_R$ =πτώση τάσης στην αντίσταση,  $V_{LED}$ =πτώση τάσης στο LED)

Από το νόμο του  $\Omega\mu$  γνωρίζουμε ότι το ρεύμα μπορεί να υπολογιστεί αν διαιρέσουμε την τάση στο σημείο που εξετάζουμε με την αντίσταση. Το ρεύμα στο κύκλωμα είναι ίδιο παντού (ένας κλάδος ροής). Επειδή δεν ξέρουμε την αντίσταση του LED (μπορεί να υπολογιστεί αργότερα) θα υπολογίσουμε το ρεύμα που διαρρέει την αντίσταση, άρα το ρεύμα του κυκλώματος  $I_K$ .

Ποια είναι όμως η τάση στα άκρα της αντίστασης; Είναι όση απομένει αν αφαιρέσουμε από την τάση της πηγής την πτώση τάσης της διόδου LED.

Αφού ισχύει ότι  $V_S = V_R + V_{LED}$ , τότε  $V_R = V_S - V_{LED}$

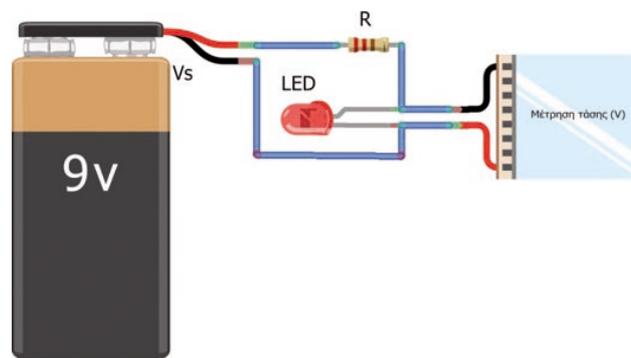
Έτσι, το ρεύμα θα είναι:

$$I_K = \frac{V_S - V_{LED}}{R}$$

Θεωρητικά μπορούμε να υποθέσουμε ότι η δίοδος LED είναι ιδανική οπότε δεν θα έχει πτώση τάσης (μηδενική αντίσταση). Σε αυτή την περίπτωση η τάση της πηγής θα ισούται με την πτώση τάσης στην αντίσταση και το ρεύμα θα είναι  $I_k = V_s/R$ .

### Μέτρηση τάσης

Μπορούμε να κάνουμε μετρήσεις στο κύκλωμα χρησιμοποιώντας ένα πολύμετρο (εικόνα 2.5). Η μέτρηση της τάσης (μετριέται σε **Volt** και συμβολίζεται με **V**) γίνεται τοποθετώντας τους ακροδέκτες του πολύμετρου στα άκρα του στοιχείου. Στην εικόνα 2.5 φαίνεται ο τρόπος μέτρησης στα άκρα της δίοδου LED.



Εικόνα 2.5 Μέτρηση τάσης

Η μέτρηση στη δίοδο LED μας δίνει

$$V_{LED} = 2.26V$$

### Υπολογισμός ρεύματος

Στο κύκλωμα έχουμε διαλέξει μια αντίσταση  $220\Omega$ . Για να υπολογίσουμε το ρεύμα του κυκλώματος θα πρέπει να διαιρέσουμε την πτώση τάσης στην αντίσταση με τα  $220\Omega$ . Στον υπολογισμό θα χρησιμοποιήσουμε την μέτρηση τάσης στα άκρα της δίοδου LED. Επιπλέον, η μέτρηση στην πηγή δίνει  $9.3V$ . Έτσι, το ρεύμα του κυκλώματος θα είναι:

$$I_k = \frac{V_s - V_{LED}}{R} = \frac{9.3V - 2.26V}{220\Omega} = \frac{7.04V}{220\Omega} = 0.032A$$



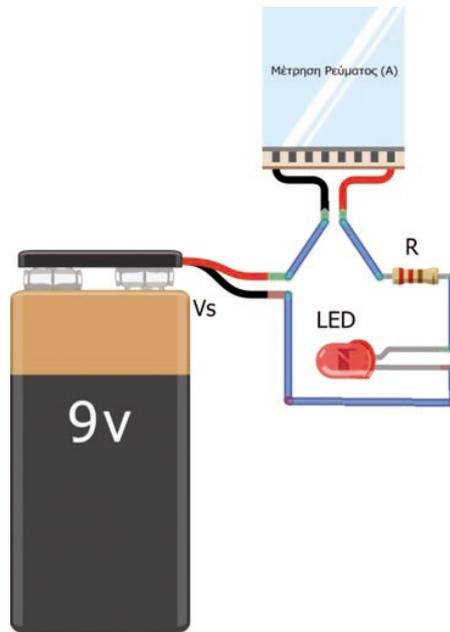
**Σημείωση:** Για λόγους απλότητας δεν έχουμε επεκταθεί σε θεωρητική ανάλυση που περιλαμβάνει τον τύπο της δίοδου (π.χ. πυριτίου, γερμανίου) καθώς και την τάση που η δίοδος άγει στην ορθή πόλωση.

### Μέτρηση ρεύματος κυκλώματος

Για να μετρήσουμε το ρεύμα (μετριέται σε **Ampere** που συμβολίζεται με **A**) θα πρέπει να συνδέσουμε το πολύμετρο σε σειρά, δηλαδή να διακόψουμε το κύκλωμα το οποίο θα κλείνει μέσα από το όργανο μέτρησης. Θα πρέπει να σημειωθεί ότι για να μετρηθεί ρεύμα θα πρέπει να υπάρχει φορτίο στο κύκλωμα που να δημιουργεί πτώση τάσης. Σε αντίθετη περίπτωση θα είναι σαν να βραχυκυκλώνουμε την πηγή τάσης με επικίνδυνα αποτελέσματα. Η εικόνα 2.6 δείχνει τη μέτρηση του ρεύματος στο κύκλωμα. Η μέτρηση του ρεύματος θα πρέπει να επαληθεύει τον προηγούμενο υπολογισμό.

Η μέτρηση του ρεύματος μας δίνει  $I_K=0.032A$  ή  $I_K=32mA$

Η κλίμακα mA χρησιμοποιείται για την καλύτερη αποτύπωση του αποτελέσματος. Ας θυμηθούμε τα προθέματα ( $m=1 \times 10^{-3}$ ,  $\mu=1 \times 10^{-6}$ ,  $n=1 \times 10^{-9}$ ,  $p=1 \times 10^{-12}$ ). Δηλαδή  $32mA = ((1 \times 10^{-3}) \times 32) A = (0.001 \times 32)A$



Εικόνα 2.6 Μέτρηση ρεύματος

### Πώς θα κάνουμε το LED να αναβοσβήνει;

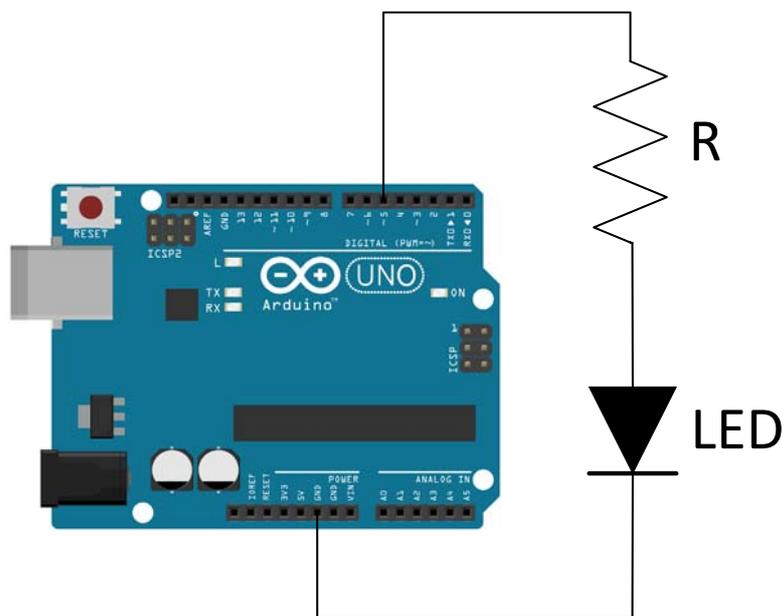
Η εναλλαγή της τάσης της πηγής θα κάνει το LED να αναβοσβήνει ( $0V$ =σβηστό,  $9V$ =αναμμένο). Φυσικά το LED θα ανάβει και με μικρότερες πηγές και το ρεύμα που διαρρέει το κύκλωμα θα είναι διαφορετικό. Στη δική μας περίπτωση, μπορούμε να συνδέσουμε ένα διακόπτη ο οποίος στην ουσία θα συνδέει και θα αποσυνδέει την πηγή από το κύκλωμα. Προφανώς όμως μια τέτοια λύση είναι εξαιρετικά δύσκολη αν φανταστούμε ότι θα πρέπει να αλλάζουμε την κατάσταση του διακόπτη με το χέρι.

Έτσι, θα πρέπει να βρεθεί κάποια λύση ώστε η κατάσταση του κυκλώματος να αλλάξει αυτόματα. Αυτό μπορεί να γίνει με δύο τρόπους:

- (α) Δημιουργία ειδικού ηλεκτρονικού κυκλώματος που να μπορεί να τροφοδοτεί το LED σε δύο επίπεδα (0V και 5V). Ένα τέτοιο κύκλωμα μπορεί να αποτελείται από ένα ολοκληρωμένο 555 αλλά απαιτεί περισσότερες γνώσεις ηλεκτρονικών.
- (β) Χρήση μιας ψηφιακής και αυτοματοποιημένης πλατφόρμας που να επιτρέπει τον έλεγχο του LED πιο εύκολα (π.χ. με απλό προγραμματισμό).

### Φιλοσοφία ελέγχου LED με το Arduino

Στην εικόνα 2.7 φαίνεται το προηγούμενο κύκλωμα μόνο που στη θέση της μπαταρίας έχει τοποθετηθεί το Arduino. Ο ακροδέκτης (+) του LED συνδέεται σε μια έξοδο του Arduino ενώ το (-) συνδέεται στη γείωση (Gnd).



Εικόνα 2.7 Έλεγχος LED με το Arduino

Όπως θα δούμε και αργότερα, ο ακροδέκτης 5 του Arduino (ένας από τους πολλούς ψηφιακούς ακροδέκτες που υποστηρίζει) μπορεί να δώσει ψηφιακή έξοδο 0V ή 5V. Η στάθμη εξόδου καθορίζεται μέσω προγράμματος. Όπως φαίνεται και στην εικόνα 2.7, το LED δεν συνδέεται απευθείας στην έξοδο του Arduino διότι αυτό θα είχε ως αποτέλεσμα να καταστραφεί τόσο το LED όσο και πιθανά η έξοδος αφού θα δημιουργούσε το μέγιστο σχεδόν ρεύμα. Και σε αυτή την περίπτωση ισχύει ο ίδιος υπολογισμός για το ρεύμα του κυκλώματος με μόνη διαφορά ότι η πηγή τάσης είναι 5V αντί για 9V.

## 2.2 Εφαρμογή 2

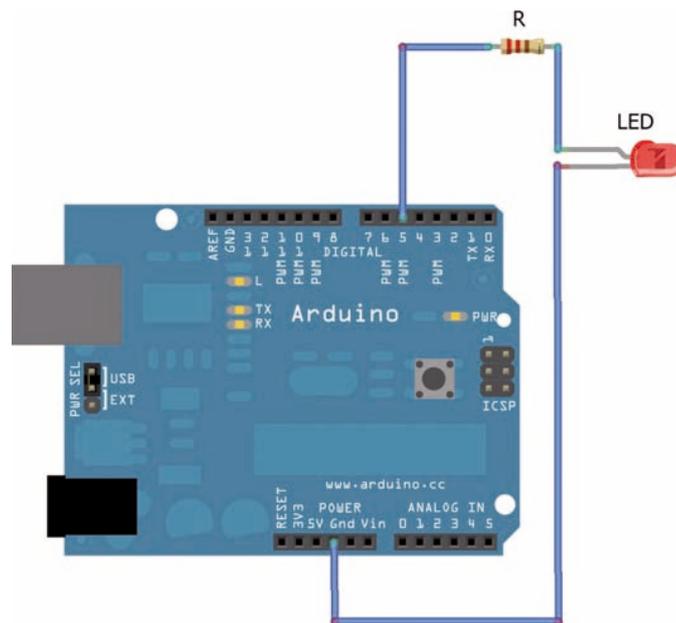
### Έλεγχος LED με το Arduino

#### 2.2.1 Σκοπός της εφαρμογής

Σε αυτή την εφαρμογή θα μάθουμε όχι μόνο να ελέγχουμε το LED αλλά και με ποιον τρόπο μπορούμε να διαχειριστούμε σε πρώτο στάδιο τους ακροδέκτες του Arduino. Επιπλέον, θα παρουσιαστούν τα ελάχιστα τμήματα των προγραμμάτων που θα αναπτύσσουμε για τις εφαρμογές μας.

#### 2.2.2 Το κύκλωμα

Το κύκλωμα (εικόνα 2.8) αποτελείται από μια αντίσταση ( $220\Omega$ ) και ένα LED (εκτός του Arduino).



Εικόνα 2.8 Μορφή κυκλώματος

Δεδομένου ότι η έξοδος του Arduino δίνει 0V ή 5V, μπορούμε να υπολογίσουμε το ρεύμα του κυκλώματος (για τον υπολογισμό θεωρούμε την πτώση τάσης στο πραγματικό LED περίπου 2V).

$$I_K = \frac{V_S - V_{LED}}{R} = \frac{5V - 2V}{220\Omega} = \frac{3V}{220\Omega} = 0.0136A$$

Για να γίνει το LED πιο φωτεινό θα πρέπει να διαρρεύσει περισσότερο ρεύμα το κύκλωμα. Αυτό σημαίνει ότι θα πρέπει να επιλέξουμε μικρότερη αντίσταση. Μπορούμε να επιλέξουμε χοντρικά μια μικρότερη αντίσταση (π.χ. 150Ω) ή να κάνουμε ακριβή υπολογισμό βάσει συγκεκριμένου ρεύματος που επιθυμούμε να διαρρεύσει στο κύκλωμα.

Αν η αντίσταση επιλεγεί να είναι 150Ω, τότε το  $I_K$  θα είναι:

$$I_K = \frac{V_S - V_{LED}}{R} = \frac{5V - 2V}{150\Omega} = \frac{3V}{150\Omega} = 0.02A$$

### 2.2.3 Ελάχιστα συστατικά του προγράμματος

Κάθε εφαρμογή απαιτεί κύκλωμα και κώδικα. Ο κώδικας (πρόγραμμα) εκτελείται από τον μικροελεγκτή και το αποτέλεσμα εμφανίζεται στο κύκλωμα. Η δυνατότητα ελέγχου εφαρμογών μέσω προγραμματισμού προσφέρει μεγάλη ευκολία για το σχεδιαστή. Σε αντίθετη περίπτωση θα έπρεπε να αναπτυχθεί ένα πολύπλοκο ηλεκτρονικό κύκλωμα (εφόσον δεν υπάρχει το Arduino).

Ο κώδικας που αναπτύσσεται για το Arduino βασίζεται στη γλώσσα προγραμματισμού Wiring C η οποία υποστηρίζει τους κλασικούς κανόνες του δομημένου προγραμματισμού.

Κάθε πρόγραμμα αποτελείται από δύο τουλάχιστον συναρτήσεις (ή διαδικασίες στην πραγματικότητα εφόσον δεν επιστρέφουν κάποιο αποτέλεσμα).

Η πρώτη συνάρτηση ονομάζεται **setup** και ο κώδικας που περιλαμβάνει εκτελείται μια μόνο φορά κατά την εφαρμογή της τροφοδοσίας στο Arduino. Συνήθως χρησιμοποιείται για αρχικοποίηση.

Η δεύτερη συνάρτηση ονομάζεται **loop** και εκτελείται συνέχεια (σαν βρόχος) όσο υπάρχει τροφοδοσία στο Arduino.

Ο κώδικας που ακολουθεί δείχνει τα ελάχιστα συστατικά ενός προγράμματος.

```
void setup ()
{
  //εντολές που εκτελούνται μια φορά
}

void loop ()
{
  //εντολές που εκτελούνται συνέχεια
}
```

Όπου υπάρχουν τα σύμβολα (//) πρόκειται για σχόλια.

## 2.2.4 Προγραμματισμός εξόδου για έλεγχο του LED

Όπως φαίνεται και στην πλακέτα του Arduino, υπάρχουν αριθμημένοι ακροδέκτες οι οποίοι χρησιμοποιούνται για την ανταλλαγή σημάτων με εξωτερικές συσκευές και κυκλώματα.

Πιο συγκεκριμένα, οι ακροδέκτες 0 έως 13 υποστηρίζουν την ανταλλαγή ψηφιακών σημάτων στάθμης 0V ή 5V. Εμείς διαλέξαμε στην τύχη τον ακροδέκτη 5. Σύμφωνα με τις στάθμες σήματος που υποστηρίζονται, θα έχουμε αναμμένο το LED δίνοντας 5V στην έξοδο στην οποία είναι συνδεδεμένο και σβηστό θέτοντας την έξοδο στα 0V.

Ο έλεγχος του LED προϋποθέτει δύο βήματα: (α) Καθορισμός του ακροδέκτη 5 ως εξόδου και (β) ρύθμιση της στάθμης σήματος (π.χ. 5V → 0V → 5V, κλπ ώστε να ανάβει και να σβήνει).

Ο καθορισμός του ακροδέκτη 5 μπορεί να γίνει εντός της συνάρτησης `setup`, ενώ η ρύθμιση της στάθμης για το LED εντός της `loop`. Έτσι, εξασφαλίζουμε τον συνεχή έλεγχο του LED όσο το Arduino είναι συνδεδεμένο στην τροφοδοσία.

Ο καθορισμός συγκεκριμένου ακροδέκτη για είσοδο ή έξοδο γίνεται με την εντολή

```
pinMode(αριθμός_ακροδέκτη, τύπος)
```

Για τους ψηφιακούς ακροδέκτες, ο `αριθμός_ακροδέκτη` μπορεί να βρίσκεται στο διάστημα 0 έως 13 (σε περίπτωση Arduino MEGA, κλπ, έχουμε περισσότερες επιλογές αφού οι ακροδέκτες είναι περισσότεροι).

Οι βασικές τιμές της παραμέτρου `τύπος` μπορεί να είναι `INPUT`, `OUTPUT` (σε ειδικές περιπτώσεις υπάρχει και ο τύπος `INPUT_PULLUP`). Στη δική μας περίπτωση ο τύπος θα είναι `OUTPUT`.

Έτσι, για να καθοριστεί ο ψηφιακός ακροδέκτης 5 ως εξόδου θα δώσουμε την εντολή

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

Το κυρίως πρόγραμμα (εντός της συνάρτησης `loop`) θα περιλαμβάνει εντολές που καθορίζουν τη στάθμη του σήματος στον ακροδέκτη 5. Η εντολή που καθορίζει τη στάθμη σήματος σε ένα ψηφιακό ακροδέκτη εξόδου ονομάζεται `digitalWrite` και συντάσσεται ως εξής:

```
digitalWrite(ακροδέκτης, στάθμη)
```

Η στάθμη μπορεί να παίρνει τις τιμές `HIGH` (5V) ή `LOW` (0V). Στην πραγματικότητα βέβαια η στάθμη `LOW` δεν αντιστοιχεί ακριβώς στα 0V. Η χρήση της εντολής `digitalWrite` για να ανάβουμε και να σβήνουμε το LED θα έχει την ακόλουθη μορφή:

```
digitalWrite(5, HIGH)
```

για να ανάψει το LED, και

```
digitalWrite(5, LOW)
```

για να σβήσει το LED.

Αν τοποθετηθούν οι δύο αυτές εντολές η μία μετά την άλλη, δεν θα προλαβαίνουμε να δούμε το LED να ανάβει και να σβήνει. Αυτό σημαίνει ότι πρέπει να υπάρχει μια χρονική καθυστέρηση πριν το LED ανάψει ή σβήσει. Αυτό επιτυγχάνεται με τη χρήση της εντολής `delay` η οποία δημιουργεί χρονική καθυστέρηση. Η εντολή συντάσσεται

```
delay(πλήθος ms)
```

για παράδειγμα, η χρονική καθυστέρηση ενός δευτερολέπτου επιτυγχάνεται αν για παράμετρο βάλουμε τον αριθμό 1000 (1000ms = 1sec).

Έτσι, ο κώδικας μπορεί να εμπλουτιστεί ως εξής:

```
digitalWrite(5, HIGH);  
delay(1000);  
digitalWrite(5, LOW);  
delay(1000);
```

Με δεδομένο ότι αυτός ο κώδικας μπαίνει στη συνάρτηση `loop`, το αποτέλεσμα είναι το LED να ανάβει και να σβήνει όσο υπάρχει τροφοδοσία στο Arduino.

Ολοκληρωμένο το πρόγραμμα έχει ως εξής (κώδικας 2.1):

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

Κώδικας 2.1 Έλεγχος LED

### 2.2.5 Βήματα υλοποίησης

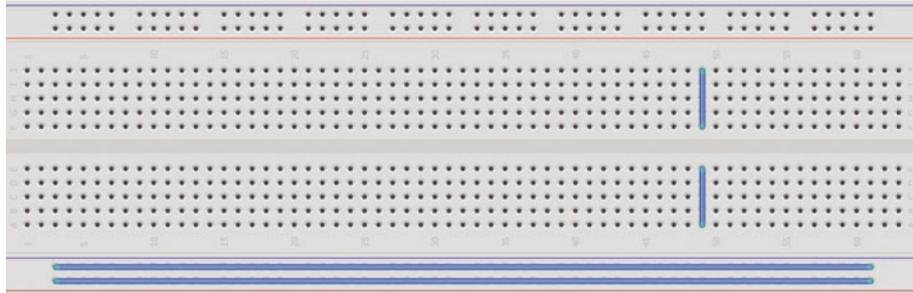
Τα βήματα υλοποίησης για την εφαρμογή ελέγχου του LED αλλά και όλων των εφαρμογών είναι τα ακόλουθα:

#### ***Βήμα 1. Υλοποίηση κυκλώματος (χωρίς εξωτερική τροφοδοσία ή σύνδεση USB)***

Όπως φαίνεται και σε προηγούμενη εικόνα του κυκλώματος, θα απαιτηθούν καλώ-

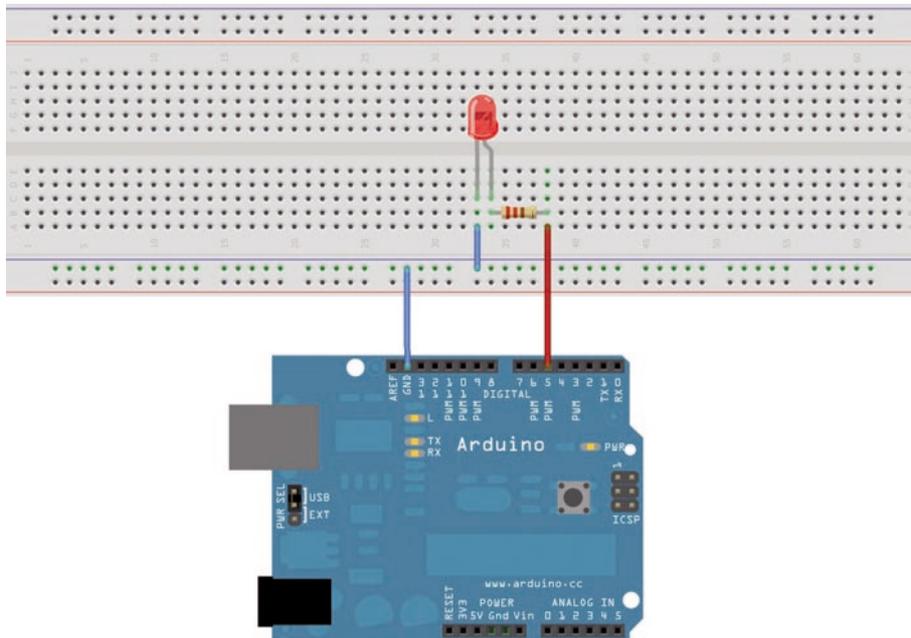
δια για τις συνδέσεις. Οι συνδέσεις καλωδίων με το χέρι ή με το κολλητήρι απαιτούν χρόνο και δεξιοότητα με βασικό μειονέκτημα ότι δύσκολα τροποποιούνται.

Για αυτό το λόγο θα χρησιμοποιήσουμε το γνωστό breadboard το οποίο διαθέτει στο εσωτερικό του ένα τυποποιημένο κύκλωμα (συνδέσεις) και στην επιφάνεια έχει υποδοχές για καλώδια και εξαρτήματα (εικόνα 2.9).



Εικόνα 2.9 Breadboard

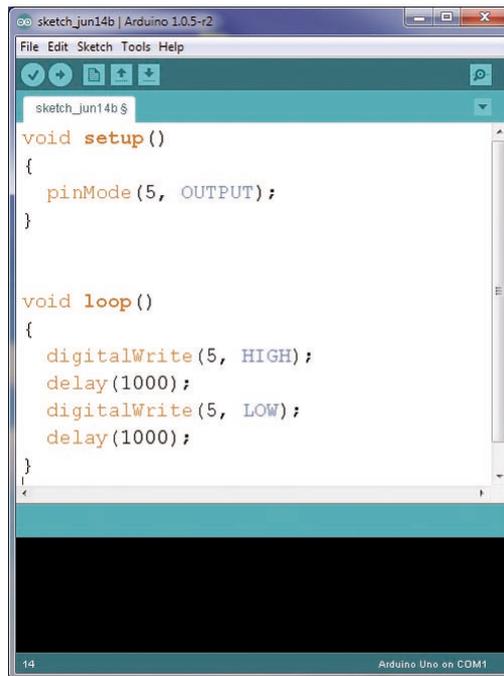
Όπως φαίνεται στην εικόνα 2.9 υπάρχουν δύο τύποι εσωτερικών συνδέσεων. Οι υποδοχές στις κεντρικές λωρίδες είναι συνδεδεμένες κατακόρυφα ενώ στις άκρες οριζόντια. Οι υποδοχές στα άκρα χρησιμοποιούνται για την τροφοδοσία (+/-) ώστε να είναι προσβάσιμη από όλα τα σημεία του breadboard. Στην εικόνα 2.10 φαίνεται η τελική υλοποίηση.



Εικόνα 2.10 Υλοποίηση κυκλώματος

### ***Βήμα 2. Εισαγωγή κώδικα στο περιβάλλον Arduino IDE (εικόνα 2.11)***

Πληκτρολογούμε τον κώδικα στο περιβάλλον IDE το οποίο έχουμε κατεβάσει δωρεάν από τη διεύθυνση <http://arduino.cc>



```
sketch_jun14b | Arduino 1.0.5-r2
File Edit Sketch Tools Help
sketch_jun14b $
void setup()
{
  pinMode(5, OUTPUT);
}

void loop()
{
  digitalWrite(5, HIGH);
  delay(1000);
  digitalWrite(5, LOW);
  delay(1000);
}
```

**Εικόνα 2.11** Ο κώδικας στο περιβάλλον IDE

### ***Βήμα 3. Μεταφορά (μεταφόρτωση) κώδικα***

Η μεταφορά του κώδικα στον μικροελεγκτή του Arduino περιλαμβάνει αυτόματα τη μεταγλώττιση του. Πριν τη μεταφορά θα πρέπει να συνδέσουμε σε μια θύρα USB το Arduino και να ρυθμίσουμε και την ίδια θύρα στο περιβάλλον IDE. Επίσης, θα πρέπει να έχουμε δηλώσει και την έκδοση του Arduino που χρησιμοποιούμε (εικόνες 2.12 και 2.13).

Αφού γίνουν οι κατάλληλες ρυθμίσεις (για να «βλέπει» το περιβάλλον IDE το Arduino board που έχουμε) μεταφέρουμε το πρόγραμμα στον μικροελεγκτή επιλέγοντας **File → Upload** (ή **Αρχείο → Φόρτωση** στην Ελληνική έκδοση) και παρατηρούμε το αποτέλεσμα.

Μπορείτε να αλλάξετε τη γλώσσα του IDE επιλέγοντας **File → Preferences** ή **Αρχείο → Επιλογές** και στη συνέχεια **Editor language**. Οι αλλαγές θα ενεργοποιηθούν με επανεκκίνηση του IDE.