

Μαριάντζελα Κομνηνού  
ΙΤΥΕ-ΔΙΟΦΑΝΤΟΣ

## Λίγα λόγια για τον οδηγό

Ο στόχος του συνοπτικού οδηγού είναι να υπάρχει ένα κείμενο αναφοράς για το υλικό του Arduino, τα βασικά του στοιχεία, τα εξαρτήματα και αισθητήρες που μπορούν να συνδεθούν, οι εντολές που αφορούν το Arduino και τα βασικά της γλώσσας C για τον προγραμματισμό του.

Περισσότερες πληροφορίες μπορούν να αναζητηθούν στα αναλυτικότερα κείμενα «Τα βασικά της C για το Arduino» και «Τα βασικά του Arduino». Τα κείμενα απευθύνονται σε όποιον θέλει να μάθει να «παίζει» με αυτή την ανοιχτή πλατφόρμα και να φτιάξει τα δικά του projects.



## Περί ανοικτότητας λογισμικού και υλικού

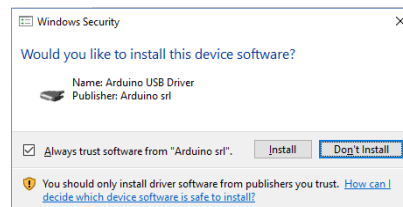
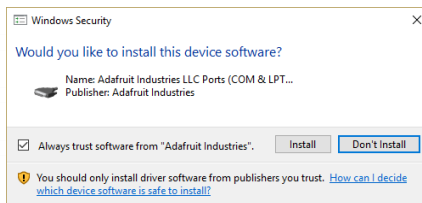
Το Arduino ως υλικό, το περιβάλλον και οι βιβλιοθήκες προγραμματισμού του και τα παραδείγματα διατίθενται με άδεια Ελεύθερου Λογισμικού και Υλικού (Open Source) και είναι διαθέσιμα στον καθένα με τους κανόνες του Creative Commons. Γιατί έτσι; Γιατί η γνώση και η ευχαρίστηση της δημιουργίας πρέπει να δίνεται απλόχερα και ελεύθερα σε όποιον την θέλει. Για το λόγο αυτό η δουλειά αυτή είναι με τους κανόνες CC ShareAlike (sa) και CC NonCommercial (nc) Περισσότερες πληροφορίες για το Creative Commons (CC) στην ιστοσελίδα <https://creativecommons.org/>

# Περιβάλλον προγραμματισμού

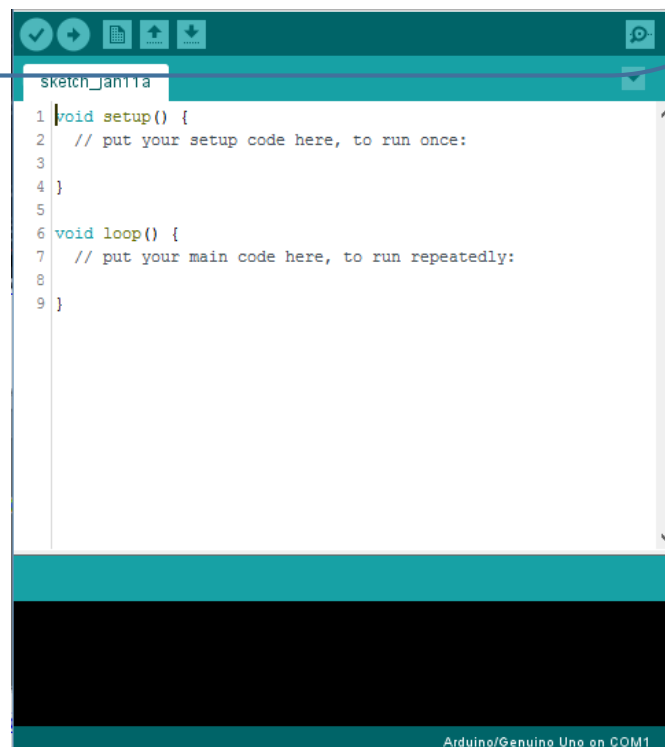
## Εγκατάσταση IDE

Η εγκατάσταση για περιβάλλον λειτουργικού συστήματος Windows γίνεται με τα ακόλουθα βήματα:

- Βήμα 1.** «Κατεβάζουμε» το λογισμικό για το λειτουργικό μας σύστημα από το <https://www.arduino.cc/en/Main/Software> από το σύνδεσμο Windows Installer
- Βήμα 2.** Εκτελούμε το αρχείο Arduino-rr-windows.exe που μόλις κατεβάσαμε (rr) είναι η τρέχουσα έκδοση π.χ. 1.8.1)
- Βήμα 3.** Επιβεβαιώνουμε πως θα εγκαταστήσουμε τους οδηγούς για τις σειριακές θύρες και τις USB θύρες.



- Βήμα 4.** Εκτελούμε την συντόμευση που έχει δημιουργηθεί στην επιφάνεια εργασίας
- Βήμα 5.** Συνδέουμε την πλακέτα Arduino μέσω του USB καλωδίου με τον υπολογιστή και εμφανίζεται το παράθυρο προσθήκης νέου υλικού και αναζητούμε τους κατάλληλους οδηγούς στο path C:\Program Files (x86)\Arduino\drivers



## Άλλα περιβάλλοντα

### Προγραμματιστικά

PlatformIO IDE <http://platformio.org/platformio-ide>

### Βασισμένα σε Scratch

S4A <http://s4a.cat/>

Ardublock <https://sourceforge.net/projects/ardublock/?source=navbar>

mBlock <http://www.mblock.cc/>

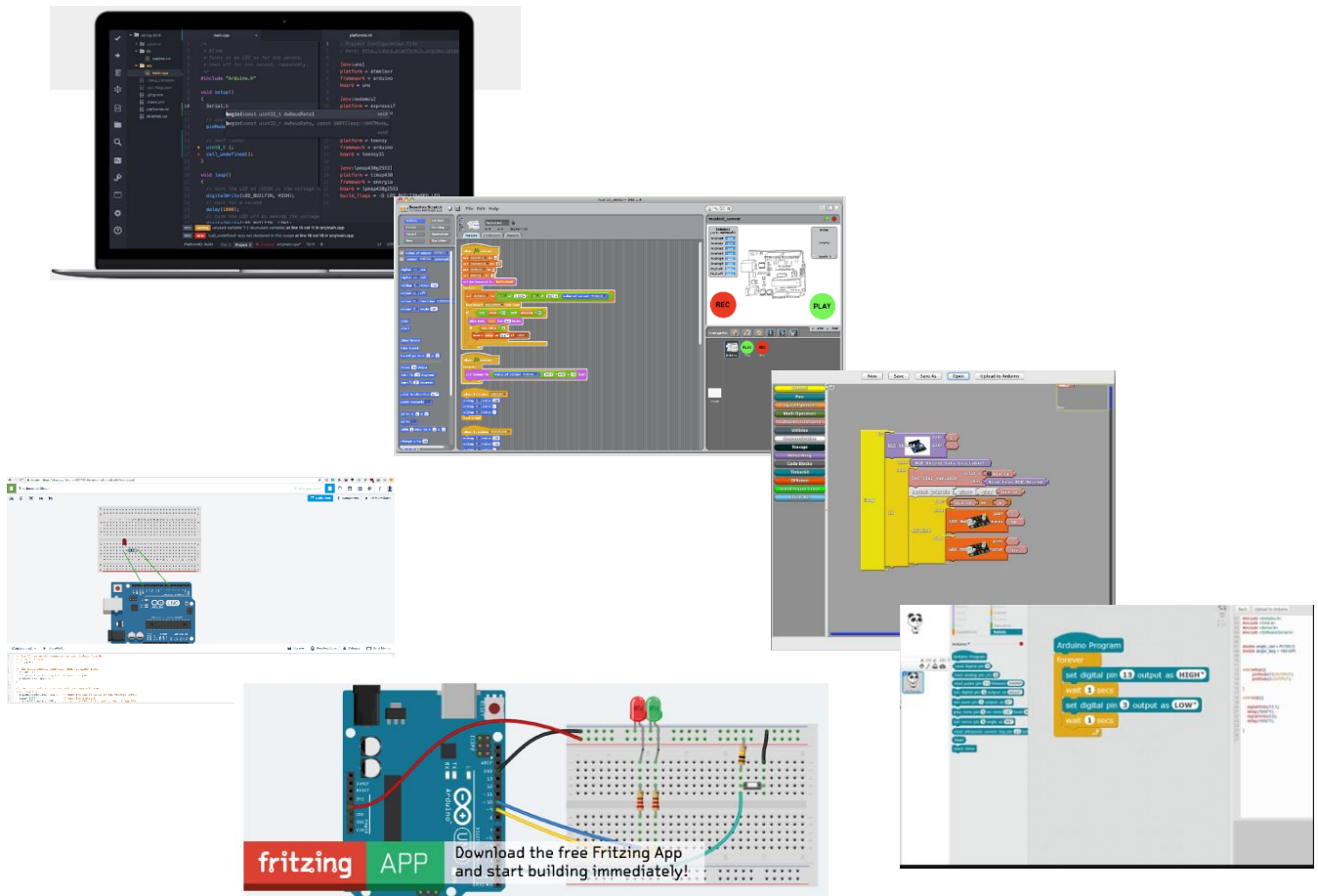
Minibloq <http://blog.minibloq.org/p/download.html>

### Εξομοιωτής

Autodesk Circuits <https://circuits.io>

### Σχεδιαστικά

fritzing <http://fritzing.org/home/>



# Στοιχεία Arduino

## Πλακέτα

Σε διάφορα μεγέθη, επεξεργαστές, μνήμη, εισόδους και εξόδους. Σίγουρα υπάρχει ένα για το δικό σας Project («Εισαγωγή στο Arduino», σελ. 9)

## Τροφοδοσία

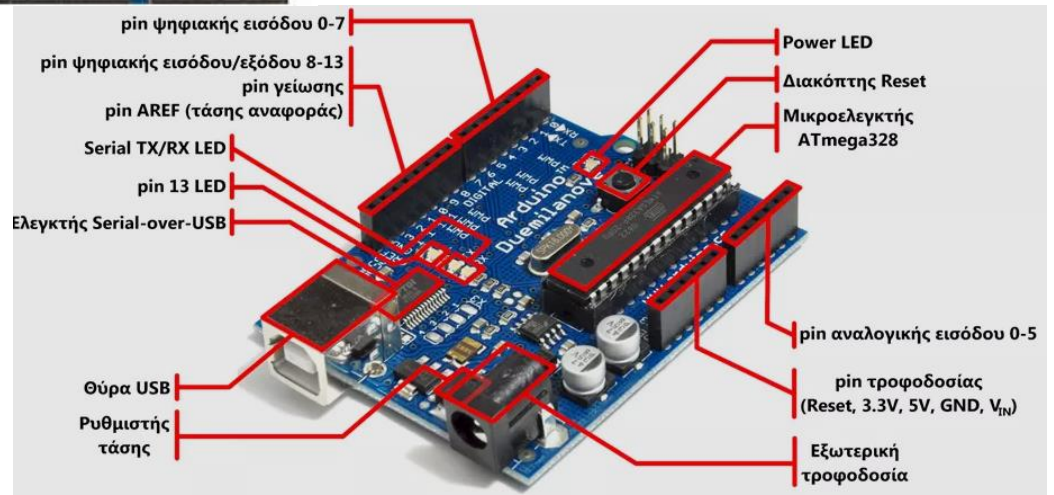
Ανάλογα την πλακέτα και τον επεξεργαστή με 3.3V και 5V. Τροφοδοτούνται με USB ή εξωτερικό τροφοδοτικό ή μπαταρία.

## Είσοδοι-Εξοδοι

Υπάρχουν Pins για αναλογικές εισόδους, συνήθως 0 ως 5 (τιμές από 0-1023), Pins που προσδιορίζονται για Ψηφιακές Εισόδους, συνήθως 0-7 (τιμές 0-1023) Pins για Ψηφιακές Εισόδους και Εξόδους, συνήθως 8-13 και κάποια από αυτά μπορούν να χρησιμοποιηθούν για Αναλογικές Εξόδους με PWM (τιμές 0-255). Ο ορισμός γίνεται στο μέρος `setup()`


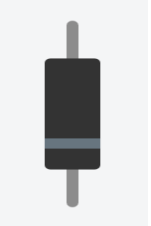
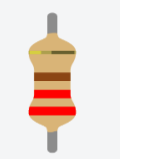
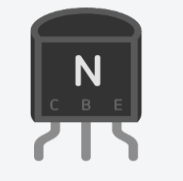

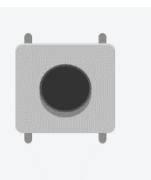
## Σειριακή


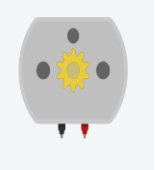


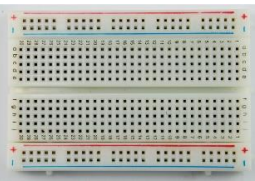
Τα pins 0 και 1 είναι το Rx (Receive) και Tx (Transmit) αντίστοιχα της σειριακής πόρτας. Δεν μπορούν να χρησιμοποιηθούν σαν I/O αν χρησιμοποιούνται για επικοινωνία με τον υπολογιστή.










# Βασικά Ηλεκτρονικά Εξαρτήματα

<b>LED</b> 	<b>Τι κάνει</b> Εκπέμπει φως όταν μικρό ρεύμα περάσει μέσα του (μόνο στη μια κατεύθυνση)  <b>Πως το αναγνωρίζουμε</b> Μοιάζει με μικρή λάμπα.	<b>Αριθμός επαφών</b> 2 (ένα μακρύτερο που συνδέεται με το θετικό πόλο) <b>Τι να προσέξουμε</b> -Λειτουργεί μόνο στη μια κατεύθυνση -Απαιτεί αντιστάτη
<b>ΔΙΟΔΟΣ (DIODE)</b> 	<b>Τι κάνει</b> Είναι το αντίστοιχο ηλεκτρονικό στοιχείο μιας μονόδρομης βαλβίδας. Επιτρέπει το ρεύμα να περνάει προς μια κατεύθυνση αλλά όχι και αντίστροφα.  <b>Πως το αναγνωρίζουμε</b> Συνήθως είναι κύλινδρος με επαφές που προεξέχουν από τα δύο άκρα του	<b>Αριθμός επαφών</b> 2  <b>Τι να προσέξουμε</b> Δουλεύει μόνο προς τη μια κατεύθυνση
<b>ΑΝΤΙΣΤΑΤΗΣ (RESISTOR)</b> 	<b>Τι κάνει</b> Περιορίζει την ποσότητα ρεύματος που περνά μέσα από ένα κύκλωμα.  <b>Πως το αναγνωρίζουμε</b> Είναι κύλινδρος με εξογκωμένα άκρα και επαφές στα δύο άκρα του.	<b>Αριθμός επαφών</b> 2  <b>Τι να προσέξουμε</b> Να επιλέγουμε το σωστό αντιστάτη ανάλογα το κύκλωμα.
<b>ΤΡΑΝΖΙΣΤΟΡ (TRANSISTOR)</b> 	<b>Τι κάνει</b> Χρησιμοποιεί ένα μικρό ρεύμα για να ενεργοποιήσει ή ενισχύσει ένα μεγαλύτερο ρεύμα.  <b>Πως το αναγνωρίζουμε</b> Είναι συνήθως μισός μαύρος κύλινδρος με 3 ποδαράκια-επαφές	<b>Αριθμός επαφών</b> 3(βάση, συλλέκτης, εκπομπή)  <b>Τι να προσέξουμε</b> Πρέπει να συνδέονται τα ποδαράκια-επαφές με το σωστό τρόπο. Συνήθως χρειάζεται ένας αντιστάτης στην επαφή της Βάσης
<b>ΠΟΤΕΝΣΙΟΜΕΤΡΟ (POTENTIOMETER)</b> 	<b>Τι κάνει</b> Παρέχει διαφορετική αντίσταση ανάλογα με τη θέση της γωνίας του δείκτη.  <b>Πως το αναγνωρίζουμε</b> Έχει μία βίδα ή επιφάνεια που περιστρέφεται και έχει ένα βελάκι επάνω του.	<b>Αριθμός επαφών</b> 3  <b>Τι να προσέξουμε</b> Να μην αγοραστεί κατά λάθος ποτενσιόμετρο λογαριθμικής κλίμακας.
<b>ΚΟΥΜΠΙ (PUSH BUTTON)</b> 	<b>Τι κάνει</b> Ενεργοποιεί το κύκλωμα όταν πατηθεί.  <b>Πως το αναγνωρίζουμε</b> Μικρό τετράγωνο με οδηγούς στο κάτω μέρος κι ένα κουμπί στην κορυφή.	<b>Αριθμός επαφών</b> 4  <b>Τι να προσέξουμε</b> Είναι τετράγωνα οπότε μπορούν να εισαχθούν με 90 μοίρες γωνία.

<b>ΠΙΕΖΟ ΣΤΟΙΧΕΙΟ (PIEZO ELEMENT)</b> 	<b>Τι κάνει</b> Ένας παλμός ρεύματος το ενεργοποιεί. Η ροή των παλμών θα το αναγκάσει να εκπέμψει έναν ήχο. <b>Πως το αναγνωρίζουμε</b> Συνήθως είναι ένα μαύρο βαρελάκι κι άλλες φορές απλά ένας χρυσός δίσκος.	<b>Αριθμός επαφών</b> 2  <b>Τι να προσέξουμε</b> Είναι δύσκολο να μη χρησιμοποιηθεί σωστά.
<b>DC ΜΟΤΕΡ</b> 	<b>Τι κάνει</b> Περιστρέφεται όταν περνά ρεύμα.  <b>Πως το αναγνωρίζουμε</b> Μοιάζει με μοτέρ	<b>Αριθμός επαφών</b> 2  <b>Τι να προσέξουμε</b> Πρέπει να χρησιμοποιείται τρανζίστορ το οποίο να ταιριάζει στο μοτέρ.
<b>ΦΩΤΟΑΝΤΙΣΤΑΤΗΣ</b> 	<b>Τι κάνει</b> Παράγει μια μεταβλητή αντίσταση η οποία εξαρτάται από την ποσότητα του φωτός που δέχεται. <b>Πως το αναγνωρίζουμε</b> Συνήθως είναι ένας μικρός δίσκος με διάφανη κορυφή και μια καμπυλωτή γραμμή από κάτω.	<b>Αριθμός επαφών</b> 2  <b>Τι να προσέξουμε</b> Πρέπει να είναι περάσει από έναν διαιετη τάσης πριν δώσει αποτέλεσμα.
<b>Καλώδια σύνδεσης</b> 	<b>Τι κάνει</b> Καλώδια σύνδεσης  <b>Πως το αναγνωρίζουμε</b> Χρωματιστά καλώδια με μαύρα άκρα	<b>Αριθμός επαφών</b> 2  <b>Τι να προσέξουμε</b> Υπάρχουν 3 ειδών αρσενικά στα 2 άκρα (M-M), αρσενικό-θηλυκό (M-F), και θηλυκά και στα 2 άκρα (F-F)
<b>Βοηθητική πλακέτα</b> 	<b>Τι κάνει</b> Βοηθητική πλακέτα (breadboard)  <b>Πως το αναγνωρίζουμε</b> Ορθογώνια πλακέτα με τρύπες ομαδοποιημένα σε 4 σειρές και πολλές στήλες	<b>Αριθμός επαφών</b> Ανάλογα με το μέγεθος  <b>Τι να προσέξουμε</b> Οι 2 πάνω και κάτω σειρές (με σύμβολα +, -) είναι συνδεδεμένα εσωτερικά σε γραμμές. Οι εσωτερικές 2 ομάδες είναι συνδεδεμένες κατά στήλες

# Ειδικά Ηλεκτρονικά Εξαρτήματα

<p><b>Αισθητήρας Θερμοκρασίας Υγρασίας</b></p> 	<p><b>Τι κάνει</b> Μετρά Θερμοκρασία και Υγρασία περιβάλλοντος</p> <p><b>Πως το αναγνωρίζουμε</b> Έχει ένα λευκό διάτρητο κουτί</p>	<p><b>Αριθμός επαφών</b> 3</p> <p><b>Τι να προσέξουμε</b> Υπάρχει και με μικρότερη ακρίβεια και είναι μπλε</p>
<p><b>Οθόνη</b></p> 	<p><b>Τι κάνει</b> Δείχνει στην οθόνη δύο γραμμών χαρακτήρες και αριθμούς για να βλέπουμε μετρήσεις</p> <p><b>Πως το αναγνωρίζουμε</b> Έχει μία οθόνη ορθογώνιου παραλληλόγραμμου πάνω σε πλακέτα</p>	<p><b>Αριθμός επαφών</b> 16</p> <p><b>Τι να προσέξουμε</b> Υπάρχουν με οπίσθιο φωτισμό και χωρίς. Επίσης υπάρχουν TFT οθόνες (επαφής ή όχι) διαφόρων μεγεθών που μπορούν να έχουν γραφικά και να απεικονίζουν περισσότερους χαρακτήρες</p>
<p><b>WiFi επικοινωνία</b></p> 	<p><b>Τι κάνει</b> Δημιουργεί ασύρματη επικοινωνία wifi με access point και συνδέεται στο διαδίκτυο</p> <p><b>Πως το αναγνωρίζουμε</b> Έχει στο ένα άκρο μια χρυσή τεθλασμένη γραμμή (ζικ-ζακ) που είναι η κεραία του</p>	<p><b>Αριθμός επαφών</b> 24</p> <p><b>Τι να προσέξουμε</b> Υπάρχουν πολλές εκδόσεις με διαφορετικά χαρακτηριστικά. Απαιτούνται ιδιαίτερες γνώσεις προγραμματισμού και δικτύων.</p>
<p><b>Αισθητήρας μονοξειδίου του άνθρακα</b></p>  <p><b>MQ7</b></p>	<p><b>Τι κάνει</b> Αισθητήρας για ανίχνευση μονοξειδίου του άνθρακα</p> <p><b>Πως το αναγνωρίζουμε</b> Γράφει MQ-7</p>	<p><b>Αριθμός επαφών</b> 6</p> <p><b>Τι να προσέξουμε</b> Χρειάζεται να συνδεθούν Pins μεταξύ τους</p>
<p><b>Αισθητήρας καπνού</b></p>  <p><b>MQ2</b></p>	<p><b>Τι κάνει</b> Αισθητήρας για εύφλεκτα αέρια (Μεθάνιο, Βουτάνιο, LPG, καπνό)</p> <p><b>Πως το αναγνωρίζουμε</b> Γράφει MQ-2</p>	<p><b>Αριθμός επαφών</b> 4</p> <p><b>Τι να προσέξουμε</b> Κάποιες υλοποιήσεις έχουν μεταβλητή αντίσταση σε ποτενσιόμετρο για να ρυθμίζεται η ευαισθησία του</p>



# Προγραμματισμός Arduino

## Συνοπτικά

Το Arduino προγραμματίζεται με τη γλώσσα C. Η σύντομη περιγραφή που ακολουθεί στοχεύει σε αυτούς που έχουν κάποια μικρή εμπειρία σε προγραμματισμό και θέλουν να γνωρίζουν τις ιδιαιτερότητες των εντολών του Arduino αλλά και του προγραμματισμού σε γλώσσα C.

Αν χρειάζεστε περισσότερες πληροφορίες και παραδείγματα ανατρέξτε στα «Εισαγωγή στο Arduino» και «Τα βασικά της C για το Arduino». Παράλληλα στο διαδίκτυο υπάρχουν πολλές πληροφορίες για τον προγραμματισμό, τη χρήση εξαρτημάτων και αισθητήρων, παραδείγματα και οδηγίες για πολλές κατασκευές, τόσο για αρχάριους όσο και για έμπειρους προγραμματιστές.

<https://www.arduino.cc>

<https://learn.adafruit.com/>

<https://learn.sparkfun.com/tutorials>

<http://www.instructables.com/tag/type-id/category-technology/channel-arduino/>

<https://www.hackster.io/arduino>

<https://thingspeak.com/>

<http://www.esp8266.com/>

<https://www.mysensors.org/>

## Δομή Προγράμματος

Κάθε πρόγραμμα Arduino (sketch) έχει δύο μέρη (συναρτήσεις)

### **Void setup() { }**

Ο κώδικας ανάμεσα στις αγκύλες τρέχει μόνο μία φορά όταν ξεκινά το Arduino. Στο τμήμα αυτό γίνονται οι αρχικοποιήσεις.

### **Void loop() { }**

Η συνάρτηση αυτή εκτελείται μετά την setup συνεχώς μέχρι να βγει η τροφοδοσία ρεύματος. Εδώ τρέχει το κυρίως πρόγραμμα.

## Σύνταξη C – Μεταβλητές και Σχόλια

// γραμμή σχόλιου

/\* πολλές γραμμές με  
σχόλια \*/

{ } Ορίζει μπλοκ εντολών.  
Χρησιμοποιείται σε  
βρόγχους επανάληψης και  
συναρτήσεις

; δηλώνει το τέλος κάθε  
εντολής

**int** (ακέραιος) 2 bytes τιμές από -32768 ως 32767

**long** (μεγάλος ακέραιος) 4 bytes  
τιμές από -2,147,483,648 ως 2,147,483,647

**float** (πραγματικός)  
4 bytes  
τιμές -3.4028235E+38 ως +3.4028235E+38

**char** (χαρακτήρας) 1 byte

**boolean** (λογικό) 1 byte TRUE - FALSE

## Αριθμητικοί Τελεστές

Χρησιμοποιούνται  
για πράξεις σε  
μεταβλητές με  
αριθμούς

= εκχώρηση (x = 5; z = x;)

+, -, \*, / Οι τέσσερις πράξεις

% (modulo) δίνει το υπόλοιπο μια διαίρεσης (πχ 12%10 δίνει 2)

++, -- αυξάνει ή μειώνει αντίστοιχα κατά ένα το περιεχόμενο μίας μεταβλητής (π.χ. αν i=0; i++; μας δίνει 1 στο i)

=, -=, \*=, /=, %= εκχωρεί σε μία μεταβλητή το προηγούμενο περιεχόμενό της μετά την πράξη (πχ. C+=A είναι ισοδύναμο με C=C+A)

## Τελεστές Σύγκρισης

Χρησιμοποιούνται  
για λογικές  
συγκρίσεις

== ισότητα (π.χ. 10 == 5 είναι FALSE, 10==10 είναι TRUE)

!= μη ίσο (π.χ. 10 != 5 TRUE, 10!=10 FALSE)

<, <= μικρότερο, μικρότερο ή ίσο (π.χ. 5<10 TRUE)

>, >= μεγαλύτερο, μεγαλύτερο ή ίσο (π.χ. 5>=10 FALSE)

## Άλλοι Τελεστές

**sizeof()** μέγεθος μιας μεταβλητής (π.χ. `int a`; `sizeof(a)` επιστρέφει 4)

**&** διεύθυνση μιας μεταβλητής (π.χ. `&a` επιστρέφει τη διεύθυνση της μεταβλητής και όχι το περιεχόμενο)

**\*** δείκτης σε μία μεταβλητή

**?:** έλεγχος συνθήκης. Αν η συνθήκη αληθής εκτελείται το πρώτο μέρος, αλλιώς το δεύτερο (π.χ. `(num%2==0)?printf("Αρτιος"):printf("Περιττός");`)

## Έλεγχος Ροής

```
if(condition1) { }  
else if(condition2) { }  
else { }
```

```
while(condition) {  
    εντολές;  
}
```

```
for(int i=0;i<10;i++) { }
```

```
do {  
    εντολές;  
} while(condition)
```

```
switch(n) {  
    case 1:  
        εντολές;  
        break;  
    case 2:  
        εντολές;  
        break;  
    default:  
        εντολές;  
}
```

## Συναρτήσεις για είσοδο και έξοδο δεδομένων στη C

Στην γλώσσα C χρησιμοποιούνται οι συναρτήσεις `scanf` και `printf`. Στο Arduino η βασική επικοινωνία του προγράμματος με το χρήστη για το διάβασμα και τύπωμα τιμών μεταβλητών γίνεται κυρίως μέσω της σειριακής θύρας.

**scanf()** -Συνάρτηση για είσοδο δεδομένων στο πρόγραμμα

π.χ. αν έχουν γίνει οι δηλώσεις `char ch`; `int a`; `float b`; `char str[5]`; τότε η είσοδος των τιμών των μεταβλητών γίνεται με `scanf("%c%d%f%s", &ch, &a, &b, str)`; Η πρώτη παράμετρος σε εισαγωγικά είναι το αλφαριθμητικό μορφοποίησης ("`%c%d%f`") όπου *%c* **χαρακτήρας**, *%d* **ακέραιος** και *%f* **πραγματικός**, *%s* **σειρά χαρακτήρων-string**), ενώ οι επόμενες είναι οι διευθύνσεις μνήμης των μεταβλητών **ch**, **a**, **b** και **str**. Προσοχή: Επειδή η `str` είναι πίνακας δεν χρειάζεται το `&`

**printf()** -Συνάρτηση για εκτύπωση των τιμών μεταβλητών. Αν έχουν γίνει οι δηλώσεις μεταβλητών όπως παραπάνω τότε η εκτύπωση γίνεται με `printf("character %c integer %d float %f string %s", ch, a, b, str)`;

## Ψηφιακές εντολές

### **pinMode(pin,mode)**

Ορίζει ένα Pin ως είσοδο ή έξοδο (όπου pin:0-19, mode: **INPUT**, **OUTPUT**)

### **digitalRead(pin,value)**

Διαβάζει από το Pin την τιμή value (**0-1023**)

### **digitalWrite(pin,value)**

Το Pin που έχει οριστεί ως **OUTPUT** τίθεται **HIGH** (+5V) ή **LOW** (0V)

## Αναλογικές εντολές

Το Arduino είναι ψηφιακή πλατφόρμα αλλά λειτουργεί και ως αναλογική

### **analogRead(pin,value)**

Διαβάζει από το ορισμένο ως **INPUT** Pin την τιμή value (**0** για 0V-**1023** για 5V)

### **analogWrite(pin,value)**

Το Pin που έχει οριστεί ως **OUTPUT** τίθεται με τιμές **0** (0V) ως **255** (5V)

## Χρονικές εντολές

### **millis()**

επιστρέφει τον αριθμό των milliseconds από τη στιγμή που το Arduino ξεκίνησε τη λειτουργία του

### **micros()**

επιστρέφει τον αριθμό των microseconds από τη στιγμή που το Arduino ξεκίνησε τη λειτουργία του

### **delay(ms)**

γίνεται παύση στο πρόγραμμα για ms χρόνο σε milliseconds

## Βασικές εντολές σειριακής

Στο Arduino η επικοινωνία με το χρήστη γίνεται κυρίως μέσω σειριακής θύρας

### **Serial.begin(speed)**

Αρχικοποιεί την σειριακή με ταχύτητα speed μία από 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, ή 115200 bps

### **Serial.print(val, format)** και **Serial.println(val, format)**

Εκτύπωση της ακέραιας τιμής val όπου format το αριθμητικό σύστημα μ ένα από DEC, HEX, OCT, BIN

### **Serial.read()**

επιστρέφει ως ακέραιο το πρώτο byte που είναι διαθέσιμο στον buffer της σειριακής θύρας ή -1 αν δεν υπάρχουν δεδομένα

### **Serial.available()**

Επιστρέφει τον αριθμό των bytes που είναι διαθέσιμα για διάβασμα στο buffer της σειριακής θύρας