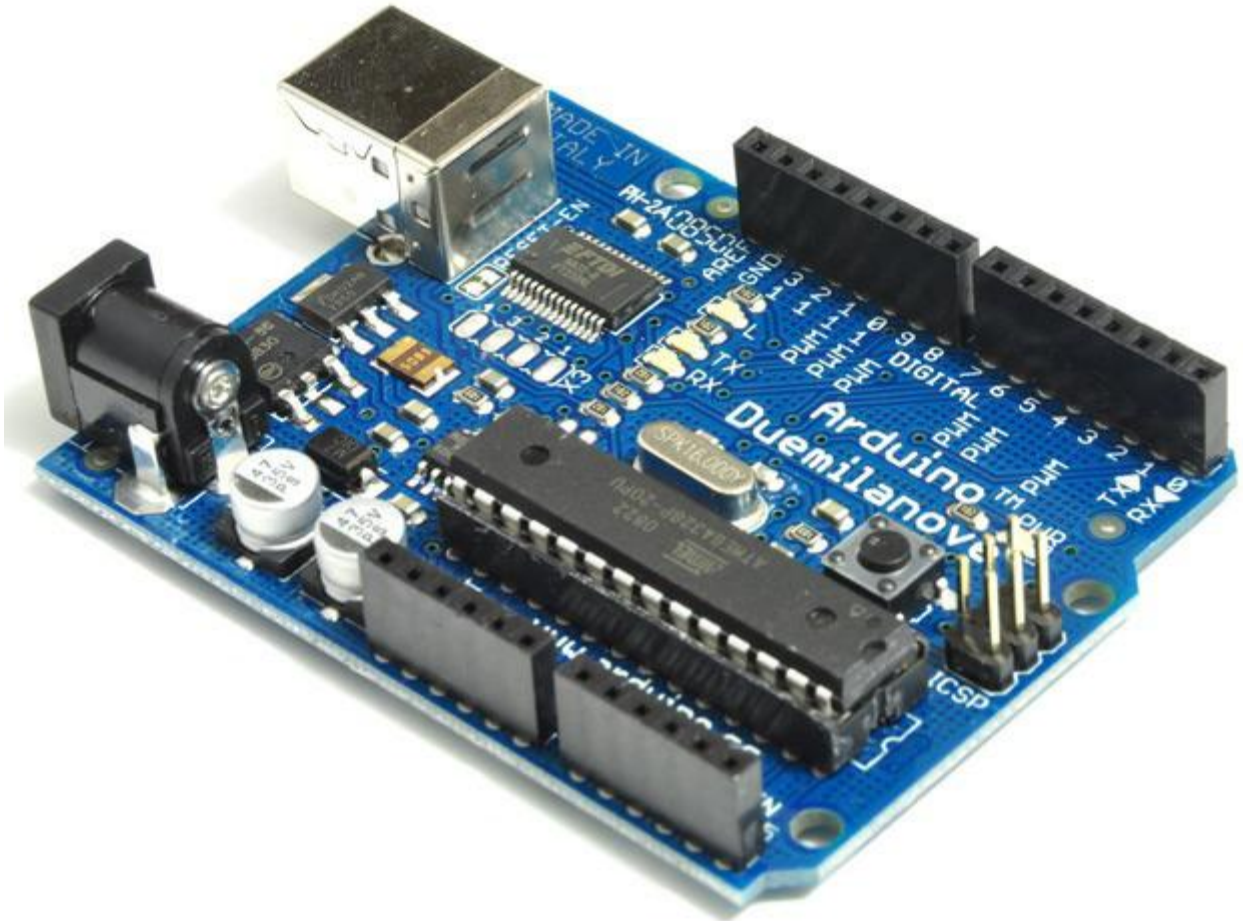


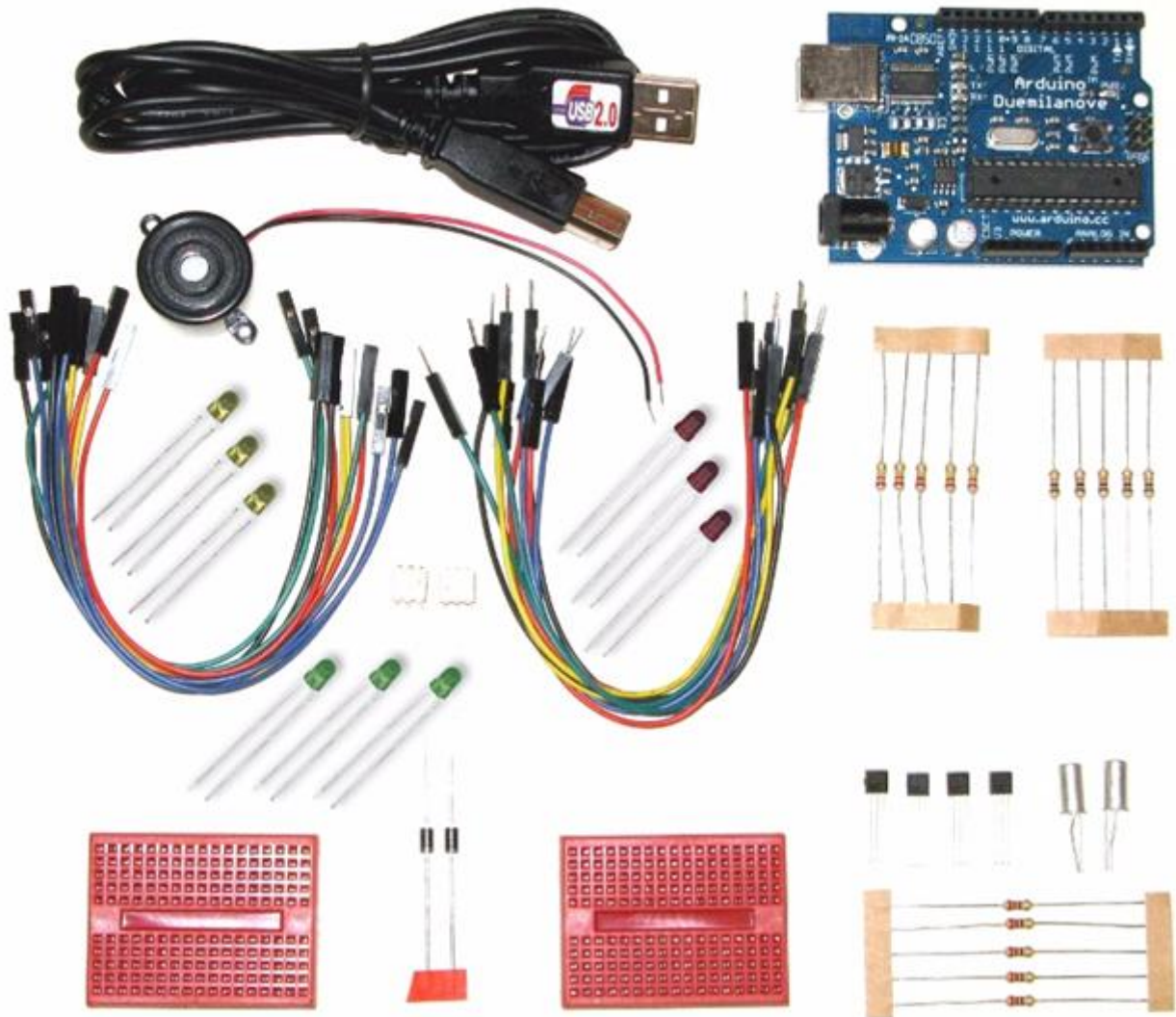
## Εισαγωγή στο Arduino

Όπως το περιγράφει ο δημιουργός του, το Arduino είναι μια «ανοικτού κώδικα» πλατφόρμα «πρωτοτυποποίησης» ηλεκτρονικών βασισμένη σε ευέλικτο και εύκολο στη χρήση hardware και software που προορίζεται για οποιονδήποτε έχει λίγη προγραμματιστική εμπειρία, στοιχειώδεις γνώσεις ηλεκτρονικών και ενδιαφέρεται να δημιουργήσει διαδραστικά αντικείμενα ή περιβάλλοντα.



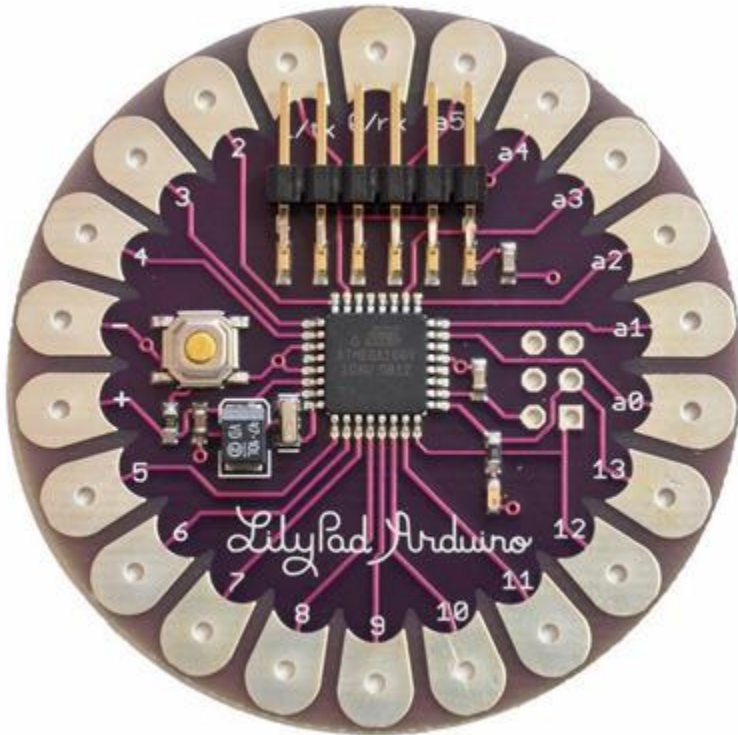
Στην ουσία, πρόκειται για ένα ηλεκτρονικό κύκλωμα που βασίζεται στον μικροελεγκτή ATmega της Atmel και του οποίου όλα τα σχέδια, καθώς και το software που χρειάζεται για την λειτουργία του, διανέμονται ελεύθερα και δωρεάν ώστε να μπορεί να κατασκευαστεί από τον καθένα (απ' όπου και ο περίεργος -για hardware- χαρακτηρισμός «ανοικτού κώδικα»). Αφού κατασκευαστεί, μπορεί να συμπεριφερθεί σαν ένας μικροσκοπικός υπολογιστής, αφού ο χρήστης μπορεί να συνδέσει επάνω του πολλαπλές μονάδες εισόδου/εξόδου και να προγραμματίσει τον μικροελεγκτή να δέχεται δεδομένα από τις μονάδες εισόδου, να τα επεξεργάζεται και να στέλνει κατάλληλες εντολές στις μονάδες εξόδου. Μάλιστα κάποιος θα μπορούσε να ισχυριστεί – και θα ήταν ένας αρκετά πετυχημένος παραλληλισμός – ότι λειτουργικά το Arduino μοιάζει πολύ με το NXT Brick των Lego Mindstorms NXT. Άλλωστε η ρομποτική είναι μια από τις πολλές εφαρμογές στις οποίες το Arduino διαπρέπει. Το Arduino βέβαια, δεν είναι ούτε ο μοναδικός, ούτε και ο καλύτερος δυνατός τρόπος για την δημιουργία μιας οποιασδήποτε διαδραστικής ηλεκτρονικής συσκευής. Όμως το κύριο πλεονέκτημά του είναι η τεράστια κοινότητα που το

υποστηρίζει και η οποία έχει δημιουργήσει, συντηρεί και επεκτείνει μια ανάλογο μεγέθους online γνωσιακή βάση. Έτσι, παρότι ένας έμπειρος ηλεκτρονικός μπορεί να προτιμήσει διαφορετική πλατφόρμα ή εξαρτήματα ανάλογα με την εφαρμογή που έχει στον νου του, το Arduino, με το εκτενές documentation, καταφέρνει να κερδίσει όλους αυτούς των οποίων οι γνώσεις στα ηλεκτρονικά περιορίζονται στα όσα λίγα έμαθαν στο σχολείο.



Ακριβώς επειδή απευθύνεται κυρίως σε αρχάριους των ηλεκτρονικών και επειδή, παρά τις αναλυτικότερες οδηγίες που υπάρχουν, δεν έχουν όλοι τις γνώσεις και τα μέσα να κατασκευάσουν μια ηλεκτρονική πλακέτα, κυκλοφορούν έτοιμες, προκατασκευασμένες πλακέτες Arduino που μπορείτε να προμηθευτείτε για περίπου €25. Με λίγα χρήματα παραπάνω μάλιστα, οι περισσότεροι προμηθευτές διαθέτουν Arduino Starter Kit, τα οποία, εκτός από το ίδιο το Arduino, περιέχουν διάφορα άλλα εξαρτήματα και εργαλεία που μπορεί να χρειαστείτε για τις πρώτες σας εφαρμογές (όπως το απαραίτητο καλώδιο USB για την σύνδεση με τον υπολογιστή, ράστερ, καλώδια, LED, διακόπτες, ποτενσιόμετρα, αντιστάσεις, διόδους, τρανζίστορ κ.λπ.). Μπορείτε να βρείτε μια λίστα με προμηθευτές του Arduino σε όλο τον κόσμο, κάνοντας κλικ στον σύνδεσμο Buy στον [επίσημο ιστοτόπο του Arduino](#). Στην λίστα δεν υπάρχει έλληνας προμηθευτής αλλά πολλά καταστήματα του εξωτερικού στέλνουν και στην Ελλάδα με αρκετά λογικά μεταφορικά κόστη. Αν πάλι θέλετε

οποσδήποτε να το προμηθευτείτε από την Ελλάδα, το [ελληνικό ηλεκτρονικό κατάστημα BuyARobot](#) διαθέτει το Arduino αλλά δεν διαθέτει Starter Kit ή συνοδευτικά shield (θα διαβάσετε για αυτά σε σχετικό πλαίσιο στην συνέχεια).



Αυτό που μπορεί να σας μπερδέψει ψάχνοντας να αγοράσετε το Arduino σε αυτά τα καταστήματα είναι οι διαφορετικές εκδόσεις στις οποίες κυκλοφορεί, επίσημες και ανεπίσημες. Από τις επίσημες εκδόσεις (Duemilanove, Diecimila, Nano, Mega, Bluetooth, LilyPad, Mini, Mini USB, Pro, Pro Mini, Serial και Serial SS) συνιστάται κυρίως η αγορά του Arduino Duemilanove ή τουλάχιστον των Diecimila ή Mega επειδή διαθέτουν υποδοχή USB και είναι συμβατές με τα shield. Για τους ίδιους λόγους, από τις ανεπίσημες εκδόσεις (Freeduino, Boarduino, Sanguino, Seeduino, BBB, RBBB κ.α.) συνιστάται μόνο το Freeduino v1.16 και το Seeduino. Δείτε το σχετικό πλαίσιο με τις κυριότερες διαφορές ανάμεσα στις εκδόσεις που προτείνονται [εδώ](#). Εκτός και αν επισημανθεί διαφορετικά, όσα θα αναφερθούν παρακάτω αφορούν την πλέον πρόσφατη βασική έκδοση του Arduino, δηλαδή το Arduino Duemilanove.

### **Μικροελεγκτής – η καρδιά του Arduino**

Το Arduino βασίζεται στον ATmega328, έναν 8-bit RISC μικροελεγκτή, τον οποίο χρονίζει στα 16MHz. Ο ATmega328 διαθέτει ενσωματωμένη μνήμη τριών τύπων:

- 2Kb μνήμης SRAM που είναι η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματά σας για να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. κατά το runtime. Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή αν γίνει reset.

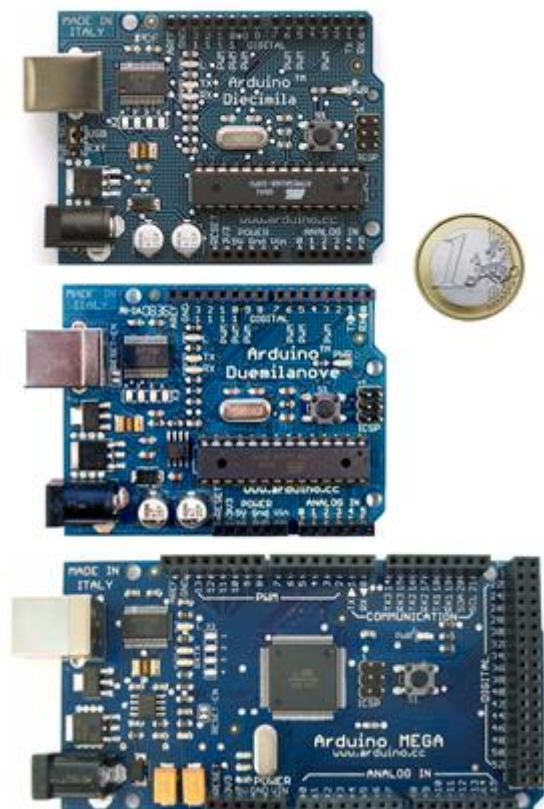
- 1Kb μνήμης EEPROM η οποία μπορεί να χρησιμοποιηθεί για «ωμή» εγγραφή/ανάγνωση δεδομένων (χωρίς datatype) ανά byte από τα προγράμματά σας κατά το runtime. Σε αντίθεση με την SRAM, η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset οπότε είναι το ανάλογο του σκληρού δίσκου.
- 32Kb μνήμης Flash, από τα οποία τα 2Kb χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware αυτό που στην ορολογία του Arduino ονομάζεται bootloader είναι αναγκαίο για την εγκατάσταση των δικών σας προγραμμάτων στον μικροελεγκτή μέσω της θύρας USB, χωρίς δηλαδή να χρειάζεται εξωτερικός hardware programmer. Τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν στον υπολογιστή σας. Η μνήμη Flash, όπως και η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset. Επίσης, ενώ η μνήμη Flash υπό κανονικές συνθήκες δεν προορίζεται για χρήση runtime μέσα από τα προγράμματά σας, λόγω της μικρής συνολικής μνήμης που είναι διαθέσιμη σε αυτά (2Kb SRAM + 1Kb EEPROM), έχει σχεδιαστεί μια βιβλιοθήκη που επιτρέπει την χρήση όσου χώρου περισσεύει (30Kb μείον το μέγεθος του προγράμματός σας σε μεταγλωττισμένη μορφή).

### **Διαφορές στις προτεινόμενες εκδόσεις του Arduino**

Το **Arduino Diecimila** έχει ουσιαστικά δύο βασικές διαφορές με το **Duemilanova**:

- Βασίζεται στον μικροελεγκτή *ATmega168*, ο οποίος διαθέτει ακριβώς την μισή μνήμη από τον *ATmega328*, δηλαδή 1Kb SRAM, 512bytes EEPROM και 16Kb Flash (14 ελεύθερα λόγω του bootloader).
- Δεν επιλέγει αυτόματα μεταξύ της εξωτερικής τροφοδοσίας και της τροφοδοσίας μέσω της θύρας USB. Το **Diecimila** διαθέτει ειδικό *jumper* με το οποίο μπορείτε να επιλέξετε χειροκίνητα την πηγή τροφοδοσίας.

Το **Arduino Mega** είναι η πιο εξελιγμένη έκδοση με τον μικροελεγκτή *ATmega1280* και αρκετά μεγαλύτερο μέγεθος. Οι διαφορές του από το **Duemilanova** είναι:

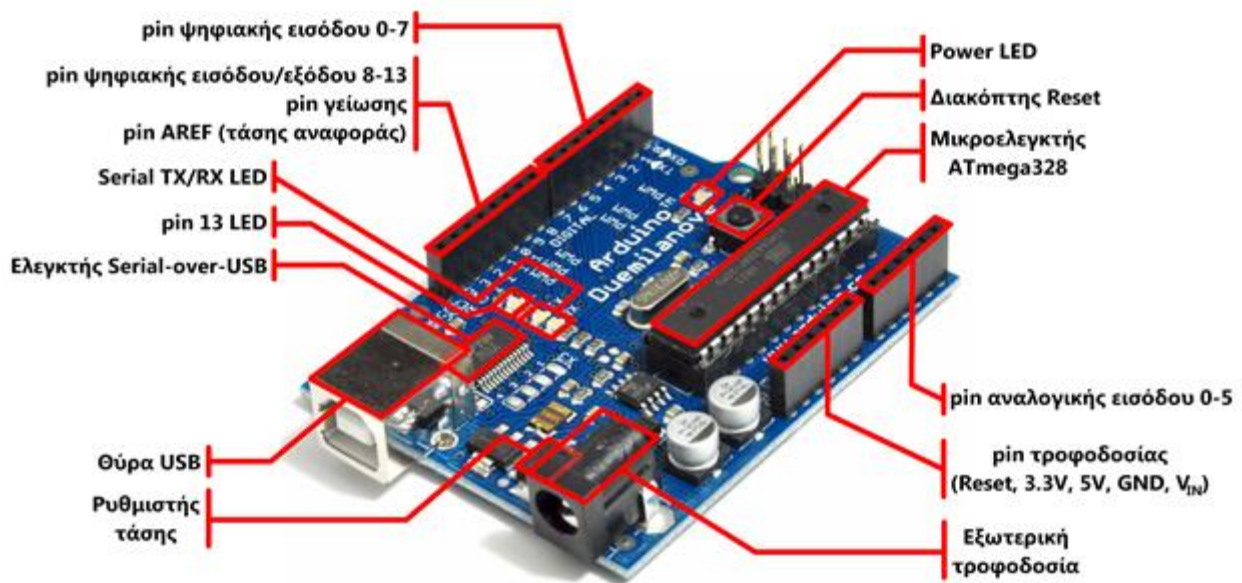


- Τετραπλάσια μνήμη (8Kb SRAM, 4Kb EEPROM, 128Kb Flash).
- 40 επιπλέον ψηφιακά pin εισόδου/εξόδου (σύνολο 54)
- 10 επιπλέον pin αναλογικής εισόδου (σύνολο 16)
- Υποστήριξη ψευδοαναλογικής εξόδου PWM σε 8 ακόμα ψηφιακά pin (σύνολο 14 PWM pin)
- Υποστήριξη εξωτερικού interrupt σε 4 ακόμα ψηφιακά pin (σύνολο 6 interrupt)
- 3 επιπλέον σειριακά interface (σύνολο 4) από τα οποία το ένα προωθείται στον ελεγκτή Serial-Over-USB όπως στο Duemilanove για σύνδεση με τον υπολογιστή.

Σημειώστε ότι το Arduino Mega είναι συμβατό με τα περισσότερα shield που έχουν κυκλοφορήσει για το Arduino αλλά όχι με το Ethernet Shield, το οποίο είναι ένα αρκετά σημαντικό μειονέκτημα για όσους θέλουν να φτιάξουν εφαρμογές με πρόσβαση στο internet ή σε κάποιο άλλο δίκτυο. Από τις ανεπίσημες εκδόσεις, το **Freeduino 1.16** και το **Seeeduino** βασίζονται στο Diecimila οπότε ισχύουν οι ίδιες διαφορές που έχει αυτό με το Duemilanove. Το Freeduino είναι ακριβής κλώνος του Diecimila, ενώ το Seeeduino είναι μια βελτιωμένη έκδοση του Diecimila με κύρια διαφορά την προσθήκη 2 επιπλέον pin αναλογικής εισόδου.

## **Είσοδοι – Έξοδοι**

Καταρχήν το Arduino διαθέτει σειριακό interface. Ο μικροελεγκτής ATmega υποστηρίζει σειριακή επικοινωνία, την οποία το Arduino προωθεί μέσα από έναν ελεγκτή Serial-over-USB ώστε να συνδέεται με τον υπολογιστή μέσω USB. Η σύνδεση αυτή χρησιμοποιείται για την μεταφορά των προγραμμάτων που σχεδιάζονται από τον υπολογιστή στο Arduino αλλά και για αμφίδρομη επικοινωνία του Arduino με τον υπολογιστή μέσα από το πρόγραμμα την ώρα που εκτελείται.



Επιπλέον, στην πάνω πλευρά του Arduino βρίσκονται 14 θηλυκά pin, αριθμημένα από 0 ως 13, που μπορούν να λειτουργήσουν ως ψηφιακές εισοδοι και έξοδοι. Λειτουργούν στα 5V και καθένα μπορεί να παρέχει ή να δεχτεί το πολύ 40mA.

Ως ψηφιακή έξοδος, ένα από αυτά τα pin μπορεί να τεθεί από το πρόγραμμά σας σε κατάσταση HIGH ή LOW, οπότε το Arduino θα ξέρει αν πρέπει να διοχετεύσει ή όχι ρεύμα στο συγκεκριμένο pin. Με αυτόν τον τρόπο μπορείτε λόγω χάρη να ανάψετε και να σβήσετε ένα LED που έχετε συνδέσει στο συγκεκριμένο pin. Αν πάλι ρυθμίσετε ένα από αυτά τα pin ως ψηφιακή είσοδο μέσα από το πρόγραμμά σας, μπορείτε με την κατάλληλη εντολή να διαβάσετε την κατάσταση του (HIGH ή LOW) ανάλογα με το αν η εξωτερική συσκευή που έχετε συνδέσει σε αυτό το pin διοχετεύει ή όχι ρεύμα στο pin (με αυτόν τον τρόπο λόγω χάρη μπορείτε να «διαβάζετε» την κατάσταση ενός διακόπτη). Μερικά από αυτά τα 14 pin, εκτός από ψηφιακές εισοδοι/έξοδοι έχουν και δεύτερη λειτουργία. Συγκεκριμένα:

- Τα pin 0 και 1 λειτουργούν ως RX και TX της σειριακής όταν το πρόγραμμά σας ενεργοποιεί την σειριακή θύρα. Έτσι, όταν λόγω χάρη το πρόγραμμά σας στέλνει δεδομένα στην σειριακή, αυτά προωθούνται και στην θύρα USB μέσω του ελεγκτή Serial-Over-USB αλλά και στο pin 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή (π.χ. ένα δεύτερο Arduino στο δικό του pin 1). Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμά σας ενεργοποιήσετε το σειριακό interface, χάνετε 2 ψηφιακές εισόδους/εξόδους.
- Τα pin 2 και 3 λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Με άλλα λόγια, μπορείτε να τα ρυθμίσετε μέσα από το πρόγραμμά σας ώστε να λειτουργούν αποκλειστικά ως ψηφιακές εισοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει \*άμεσα\* και εκτελείται μια

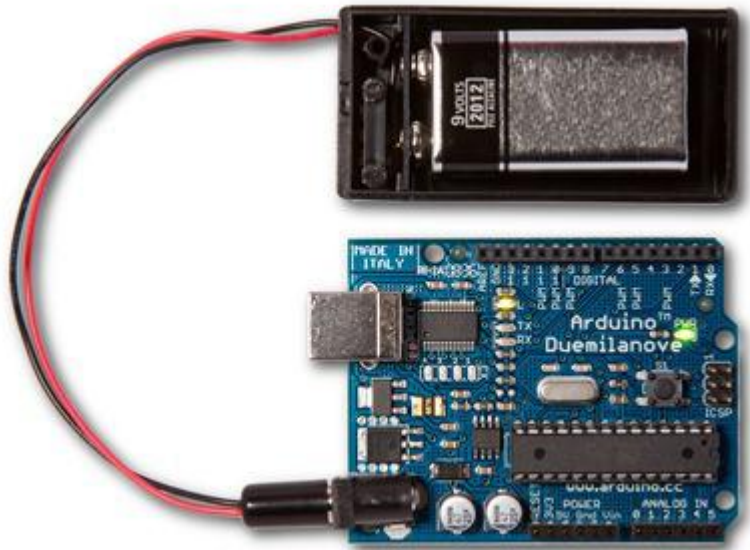
συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.

- Τα pin 3, 5, 6, 9, 10 και 11 μπορούν να λειτουργήσουν και ως ψευδοαναλογικές έξοδοι με το σύστημα PWM (Pulse Width Modulation), δηλαδή το ίδιο σύστημα που διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων. Έτσι, μπορείτε να συνδέσετε λόγω χάρη ένα LED σε κάποιο από αυτά τα pin και να ελέγξετε πλήρως την φωτεινότητά του με ανάλυση 8bit (256 καταστάσεις από 0-σβηστό ως 255-πλήρως αναμμένο) αντί να έχετε απλά την δυνατότητα αναμμένο-σβηστό που παρέχουν οι υπόλοιπες ψηφιακές έξοδοι. Είναι σημαντικό να καταλάβετε ότι το PWM δεν είναι πραγματικά αναλογικό σύστημα και ότι θέτοντας στην έξοδο την τιμή 127, δεν σημαίνει ότι η έξοδος θα δίνει 2.5V αντί της κανονικής τιμής των 5V, αλλά ότι θα δίνει ένα παλμό που θα εναλλάσσεται με μεγάλη συχνότητα και για ίσους χρόνους μεταξύ των τιμών 0 και 5V.

Στην κάτω πλευρά του Arduino, με τη σήμανση ANALOG IN, θα βρείτε μια ακόμη σειρά από 6 pin, αριθμημένα από το 0 ως το 5. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος κάνοντας χρήση του ADC (Analog to Digital Converter) που είναι ενσωματωμένο στον μικροελεγκτή. Για παράδειγμα, μπορείτε να τροφοδοτήσετε ένα από αυτά με μια τάση την οποία μπορείτε να κυμάνετε με ένα ποτενσιόμετρο από 0V ως μια τάση αναφοράς Vref η οποία, αν δεν κάνετε κάποια αλλαγή είναι προρυθμισμένη στα 5V. Τότε, μέσα από το πρόγραμμά σας μπορείτε να «διαβάσετε» την τιμή του pin ως ένα ακέραιο αριθμό ανάλυσης 10-bit, από 0 (όταν η τάση στο pin είναι 0V) μέχρι 1023 (όταν η τάση στο pin είναι 5V). Η τάση αναφοράς μπορεί να ρυθμιστεί με μια εντολή στο 1.1V, ή σε όποια τάση επιθυμείτε (μεταξύ 2 και 5V) τροφοδοτώντας εξωτερικά με αυτή την τάση το pin με την σήμανση AREF που βρίσκεται στην απέναντι πλευρά της πλακέτας. Έτσι, αν τροφοδοτήσετε το pin AREF με 3.3V και στην συνέχεια δοκιμάσετε να διαβάσετε κάποιο pin αναλογικής εισόδου στο οποίο εφαρμόζετε τάση 1.65V, το Arduino θα σας επιστρέψει την τιμή 512. Τέλος, καθένα από τα 6 αυτά pin, με κατάλληλη εντολή μέσα από το πρόγραμμα μπορεί να μετατραπεί σε ψηφιακό pin εισόδου/εξόδου όπως τα 14 που βρίσκονται στην απέναντι πλευρά και τα οποία περιγράφηκαν πριν. Σε αυτή την περίπτωση τα pin μετονομάζονται από 0~5 σε 14~19 αντίστοιχα.

## Τροφοδοσία

Το Arduino μπορεί να τροφοδοτηθεί με ρεύμα είτε από τον υπολογιστή μέσω της σύνδεσης USB, είτε από εξωτερική τροφοδοσία που παρέχεται μέσω μιας υποδοχής φισ των 2.1mm (θετικός πόλος στο κέντρο) και βρίσκεται στην κάτω-αριστερή γωνία του Arduino.



Για να μην υπάρχουν προβλήματα, η εξωτερική τροφοδοσία πρέπει να είναι από 7 ως 12V και μπορεί να προέρχεται από ένα κοινό μετασχηματιστή του εμπορίου, από μπαταρίες ή οποιαδήποτε άλλη πηγή DC. Δίπλα από τα pin αναλογικής εισόδου, υπάρχει μια ακόμα συστοιχία από 6 pin με την σήμανση POWER. Η λειτουργία του καθενός έχει ως εξής:

- Το πρώτο, με την ένδειξη RESET, όταν γειωθεί (σε οποιοδήποτε από τα 3 pin με την ένδειξη GND που υπάρχουν στο Arduino) έχει ως αποτέλεσμα την επανεκκίνηση του Arduino.
- Το δεύτερο, με την ένδειξη 3.3V, μπορεί να τροφοδοτήσει τα εξαρτήματά σας με τάση 3.3V. Η τάση αυτή δεν προέρχεται από την εξωτερική τροφοδοσία αλλά παράγεται από τον ελεγκτή Serial-over-USB και έτσι η μέγιστη ένταση που μπορεί να παρέχει είναι μόλις 50mA.
- Το τρίτο, με την ένδειξη 5V, μπορεί να τροφοδοτήσει τα εξαρτήματά σας με τάση 5V. Ανάλογα με τον τρόπο τροφοδοσίας του ίδιου του Arduino, η τάση αυτή προέρχεται είτε άμεσα από την θύρα USB (που ούτως ή άλλως λειτουργεί στα 5V), είτε από την εξωτερική τροφοδοσία αφού αυτή περάσει από ένα ρυθμιστή τάσης για να την «φέρει» στα 5V.
- Το τέταρτο και το πέμπτο pin, με την ένδειξη GND, είναι φυσικά γειώσεις.
- Το έκτο και τελευταίο pin, με την ένδειξη Vin έχει διπλό ρόλο. Σε συνδυασμό με το pin γείωσης δίπλα του, μπορεί να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας του Arduino, στην περίπτωση που δεν σας βολεύει να χρησιμοποιήσετε την υποδοχή του φικ



των 2.1mm. Αν όμως έχετε ήδη συνδεδεμένη εξωτερική τροφοδοσία μέσω του φιν, μπορείτε να χρησιμοποιήσετε αυτό το pin για να τροφοδοτήσετε εξαρτήματα με την πλήρη τάση της εξωτερικής τροφοδοσίας (7~12V), πριν αυτή περάσει από τον ρυθμιστή τάσης όπως γίνεται με το pin των 5V.

## Ενσωματωμένα κουμπιά και LED

Πάνω στην πλακέτα του Arduino υπάρχει ένας διακόπτης micro-switch και 4 μικροσκοπικά LED επιφανειακής στήριξης. Η λειτουργία του διακόπτη (που έχει την σήμανση RESET) και του ενός LED με την σήμανση POWER είναι μάλλον προφανής. Τα δύο LED με τις σημάνσεις TX και RX, χρησιμοποιούνται ως ένδειξη λειτουργίας του σειριακού interface, καθώς ανάβουν όταν το Arduino στέλνει ή λαμβάνει (αντίστοιχα) δεδομένα μέσω USB. Σημειώστε ότι τα LED αυτά ελέγχονται από τον ελεγκτή Serial-over-USB και συνεπώς δεν λειτουργούν όταν η σειριακή επικοινωνία γίνεται αποκλειστικά μέσω των ψηφιακών pin 0 και 1.

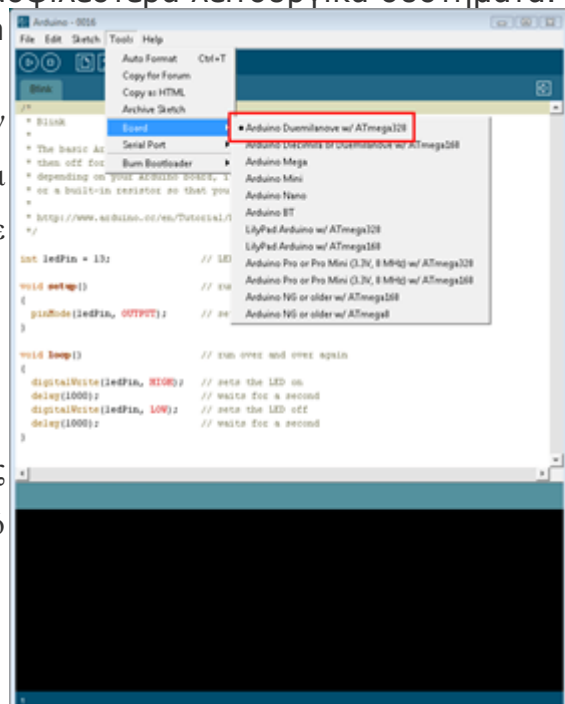
Τέλος, υπάρχει το LED με την σήμανση L. Η βασική δοκιμή λειτουργίας του Arduino είναι να του αναθέσετε να αναβοσβήνει ένα LED (θα το δείτε αυτό στην συνέχεια όταν θα φτιάξετε την πρώτη εφαρμογή σας). Για να μπορείτε να το κάνετε αυτό από την πρώτη στιγμή, χωρίς να συνδέσετε τίποτα πάνω στο Arduino, οι κατασκευαστές του σκέφτηκαν να ενσωματώσουν ένα LED στην πλακέτα, το οποίο σύνδεσαν στο ψηφιακό pin 13. Έτσι, ακόμα και αν δεν έχετε συνδέσει τίποτα πάνω στο φυσικό pin 13, αναθέτοντάς του την τιμή HIGH μέσα από το πρόγραμμά σας, θα ανάψει αυτό το ενσωματωμένο LED.

## Arduino IDE και σύνδεση με τον υπολογιστή

Ότι χρειάζεστε για την διαχείριση του Arduino από τον υπολογιστή σας το παρέχει το Arduino IDE, την τελευταία έκδοση του οποίου μπορείτε [να κατεβάσετε από το επίσημο site](#) για καθένα από τα τρία δημοφιλέστερα λειτουργικά συστήματα.

Το Arduino IDE είναι βασισμένο σε Java και συγκεκριμένα παρέχει:

- ένα πρακτικό περιβάλλον για την συγγραφή των προγραμμάτων σας (τα οποία ονομάζονται sketch στην ορολογία του Arduino) με συντακτική χρωματική σήμανση,
- αρκετά έτοιμα παραδείγματα,
- μερικές έτοιμες βιβλιοθήκες για προέκταση της γλώσσας και για να χειρίζεστε εύκολα μέσα από

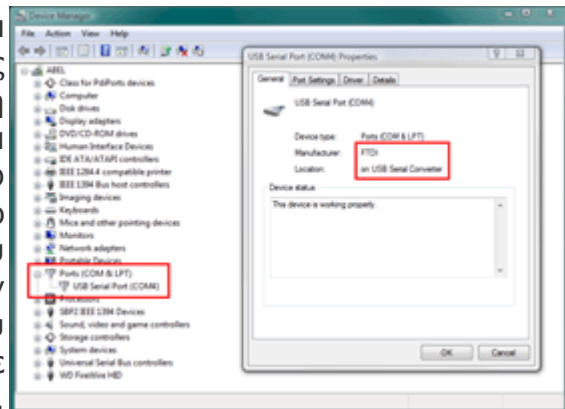


τον κώδικά σας τα εξαρτήματα που συνδέετε στο Arduino,

- τον compiler για την μεταγλώττιση των sketch σας,
- ένα serial monitor που παρακολουθεί τις επικοινωνίες της σειριακής (USB), αναλαμβάνει να στείλει αλφαριθμητικά της επιλογής σας στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για το debugging των sketch σας
- και την επιλογή να ανεβάσετε το μεταγλωττισμένο sketch στο Arduino.

Για τα δύο τελευταία χαρακτηριστικά βέβαια, το Arduino πρέπει να έχει συνδεθεί σε μια από τις θύρες USB του υπολογιστή και, λόγω του ελεγκτή Serial-over-USB, θα πρέπει να αναγνωριστεί από το λειτουργικό σας σύστημα ως εικονική σειριακή θύρα.

Για την σύνδεση θα χρειαστείτε ένα καλώδιο USB από Type A σε Type B, όπως αυτό των εκτυπωτών. Για την αναγνώριση από το λειτουργικό θα χρειαστεί να εγκαταστήσετε τον οδηγό του FTDI chip (δηλαδή του ελεγκτή Serial-over-USB) ο οποίος υπάρχει στον φάκελο drivers του Arduino IDE που κατεβάσατε. Την τελευταία έκδοση αυτού του οδηγού μπορείτε επίσης να κατεβάσετε για κάθε λειτουργικό σύστημα [από το site της FTDI](http://www.ftdi.com/).



Σημειώστε ότι στους τελευταίους πυρήνες του Linux υπάρχει εγγενής υποστήριξη του συγκεκριμένου ελεγκτή. Αν όλα έγιναν σωστά, το κεντρικό παράθυρο του Arduino IDE θα εμφανιστεί όταν το εκτελέσετε και στο μενού Tools -> Serial Port θα πρέπει να εμφανίζεται η εικονική σειριακή θύρα (συνήθως COM# για τα Windows, /dev/ttyusbserial## για το MacOS και /dev/ttyusb## για το Linux). Επιλέξτε αυτή την εικονική θύρα και στην συνέχεια επιλέξτε τον τύπο του Arduino σας (Arduino Duemilanove w/ ATmega328) από το μενού Tools -> Board. Το Arduino είναι πλέον έτοιμο να δεχτεί τα sketch σας. Αν εμφανίστηκε οποιοδήποτε πρόβλημα διαβάστε τις αναλυτικές οδηγίες εγκατάστασης για κάθε λειτουργικό σύστημα στη διεύθυνση <http://arduino.cc/en/Guide/HomePage>.

## Γλώσσα προγραμματισμού

Η γλώσσα του Arduino βασίζεται στη γλώσσα Wiring, μια παραλλαγή C/C++ για μικροελεγκτές αρχιτεκτονικής AVR όπως ο ATmega, και υποστηρίζει όλες τις βασικές δομές της C καθώς και μερικά χαρακτηριστικά της C++. Για compiler χρησιμοποιείται ο AVR gcc και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc. Λόγω της καταγωγής της από την C, στην γλώσσα του Arduino μπορείτε να χρησιμοποιήσετε ουσιαστικά τις ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους τύπων δεδομένων και τους ίδιους τελεστές όπως και στην C. Πέρα από αυτές όμως, υπάρχουν κάποιες ειδικές εντολές, συναρτήσεις και

σταθερές που βοηθούν για την διαχείριση του ειδικού hardware του Arduino. Οι πιο σημαντικές από αυτές επεξηγούνται στον πίνακα που ακολουθεί:

Όρισμα	Είδος	Τύπος	Παράμετροι	Περιγραφή
LOW	Σταθερά	int	–	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
HIGH	Σταθερά	int	–	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
INPUT	Σταθερά	int	–	Έχει την τιμή 0 και είναι αντίστοιχη του λογικού false.
OUTPUT	Σταθερά	int	–	Έχει την τιμή 1 και είναι αντίστοιχη του λογικού true.
pinMode	Εντολή	–	( <i>pin, mode</i> )	Καθορίζει αν το συγκεκριμένο ψηφιακό <i>pin</i> θα είναι <i>pin</i> εισόδου ή <i>pin</i> εξόδου ανάλογα με την τιμή που δίνεται στην παράμετρο <i>mode</i> (INPUT ή OUTPUT αντίστοιχα).
digitalWrite	Εντολή	–	( <i>pin, pinstatus</i> )	Θέτει την κατάσταση <i>pinstatus</i> (HIGH ή LOW) στο συγκεκριμένο ψηφιακό <i>pin</i> .
digitalRead	Συνάρτηση	int	( <i>pin</i> )	Επιστρέφει την κατάσταση του συγκεκριμένου ψηφιακού <i>pin</i> (0 για LOW και 1 για HIGH) εφόσον αυτό είναι <i>pin</i> εισόδου.
analogReference	Εντολή	–	( <i>type</i> )	Δέχεται τις τιμές DEFAULT, INTERNAL ή EXTERNAL στην παράμετρο <i>type</i> για να καθορίσει την τάση αναφοράς ( $V_{ref}$ ) των αναλογικών εισόδων (5V, 1.1V ή η εξωτερική τάση με την οποία τροφοδοτείται το <i>pin</i> AREF αντίστοιχα)
analogRead	Συνάρτηση	int	( <i>pin</i> )	Επιστρέφει έναν ακέραιο από 0 έως 1023, ανάλογα με την τάση που τροφοδοτείται το συγκεκριμένο <i>pin</i> αναλογικής εισόδου στην κλίμακα 0 ως $V_{ref}$ .
analogWrite	Εντολή	–	( <i>pin, value</i> )	Θέτει το συγκεκριμένο ψηφιακό <i>pin</i> σε κατάσταση ψευδοαναλογικής εξόδου (PWM). Η παράμετρος <i>value</i> καθορίζει το πλάτος του παλμού σε σχέση με την περίοδο του παραγόμενου σήματος στην κλίμακα από 0 ως 255 (π.χ. με <i>value</i> 127, το πλάτος του παλμού είναι ίσο με μισή περίοδο).
millis	Συνάρτηση	unsigned long	()	Μετρητής που επιστρέφει το χρονικό διάστημα σε ms από την στιγμή που άρχισε η εκτέλεση του προγράμματος. Λάβετε υπόψη ότι λόγω του τύπου

				μεταβλητής (unsigned long δηλ. 32bit) θα γίνει overflow σε $2^{32}$ ms δηλαδή περίπου σε 50 μέρες, οπότε ο μετρητής θα ξεκινήσει πάλι από το μηδέν.
delay	Εντολή	–	(time)	Σταματά προσωρινά την ροή του προγράμματος για <i>time</i> ms. Η παράμετρος <i>time</i> είναι unsigned long (από 0 ως $2^{32}$ ). Σημειώστε ότι παρά την προσωρινή παύση, συναρτήσεις των οποίων η εκτέλεση ενεργοποιείται από interrupt θα εκτελεστούν κανονικά κατά την διάρκεια μιας delay.
attachInterrupt	Εντολή	–	(interrupt,function,triggermode)	<p>Θέτει σε λειτουργία το συγκεκριμένο <i>interrupt</i>, ώστε να ενεργοποιεί την συνάρτηση <i>function</i>, κάθε φορά που ικανοποιείται η συνθήκη που ορίζεται από την παράμετρο <i>triggermode</i>:</p> <ul style="list-style-type: none"> <li>• LOW (ενεργοποίηση όταν η κατάσταση του pin που αντιστοιχεί στο συγκεκριμένο interrupt γίνει LOW)</li> <li>• RISING (όταν από LOW γίνει HIGH)</li> <li>• FALLING (όταν από HIGH γίνει LOW)</li> <li>• CHANGE (όταν αλλάξει κατάσταση γενικά)</li> </ul>
detachInterrupt	Εντολή	–	(interrupt)	Απενεργοποιεί το συγκεκριμένο <i>interrupt</i> .
noInterrupts	Εντολή	–	()	Σταματά προσωρινά την λειτουργία όλων των interrupt
interrupts	Εντολή	–	()	Επαναφέρει την λειτουργία των interrupt που διακόπηκε προσωρινά από μια εντολή noInterrupts.
Serial.begin	Μέθοδος κλάσης	–	(datarate)	Θέτει τον ρυθμό μεταφοράς δεδομένων του σειριακού interface (σε baud)
Serial.println	Μέθοδος κλάσης	–	(data)	Διοχετεύει τα δεδομένα <i>data</i> για αποστολή μέσω του σειριακού

interface. Η παράμετρος *data* μπορεί να είναι είτε αριθμός είτε αλφαριθμητικό.

Επιπλέον, στην γλώσσα του Arduino κάθε πρόγραμμα αποτελείται από δύο βασικές ρουτίνες ώστε να έχει την γενική δομή:

```
// Ενσωματώσεις βιβλιοθηκών, δηλώσεις μεταβλητών...
```

```
void setup()  
{  
  // ...  
}
```

```
void loop()  
{  
  // ...  
}
```

```
// Υπόλοιπες συναρτήσεις...
```

Η βασική ρουτίνα `setup()` εκτελείται μια φορά μόνο κατά την εκκίνηση του προγράμματος ενώ η βασική ρουτίνα `loop()` περιέχει τον βασικό κορμό του προγράμματος και η εκτέλεσή της επαναλαμβάνεται συνέχεια σαν ένας βρόγχος `while(true)`.

Αν και πρόκειται μόνο για τις πιο βασικές λειτουργίες της γλώσσας του Arduino, με αυτές και με λίγες βασικές γνώσεις C θα μπορούσατε να δημιουργήσετε το sketch ακόμα και για κάποιο αρκετά περίπλοκο project (όπως αυτά που θα ακολουθήσουν σε επόμενα τεύχη). Για το πλήρες *reference* πάντως, επισκεφτείτε [την σχετική σελίδα](#) ενώ ακόμα περισσότερες πληροφορίες μπορείτε να βρείτε [στο site της Wiring](#) καθώς και [στο εγχειρίδιο της βιβλιοθήκης AVR Libc](#).

## Hello World!

Έφτασε η στιγμή να δημιουργήσετε το πρώτο σας sketch, το οποίο - παραδοσιακά- πρέπει να εξάγει το μήνυμα «Hello World». Βέβαια -μέχρι να προσθέσετε εσείς μια- το Arduino δεν διαθέτει οθόνη ώστε να εμφανίσει κάποιο μήνυμα. Η μόνη συσκευή εξόδου που είναι ενσωματωμένη στην πλακέτα του Arduino είναι το LED του pin 13. Έτσι, το Arduino σας θα χαιρετίσει την οικουμένη αναβοσβήνοντας απλά το LEDάκι του. Ανοίξτε το IDE του Arduino και -για να μην πληκτρολογείτε- επιλέξτε File -> Sketchbook -> Examples -> Digital -> Blink. Θα ανοίξει ένα sketch με τον παρακάτω κώδικα:

```
1   int ledPin = 13;  
2   void setup()  
3   {  
4     pinMode(ledPin, OUTPUT);  
5   }  
6   void loop()  
7   {  
8     digitalWrite(ledPin, HIGH);  
9     delay(1000);  
10    digitalWrite(ledPin, LOW);  
11    delay(1000);  
    }
```

12

Όπως κάθε "Hello World" πρόγραμμα, το sketch αυτό είναι αρκετά straightforward. Αρχικά, στην ρουτίνα setup() ρυθμίζεται το pin στο οποίο είναι συνδεδεμένο το LED ως pin εξόδου (γραμμή 4). Στην συνέχεια η κύρια ρουτίνα loop(), η εκτέλεση της οποίας επαναλαμβάνεται συνέχεια, ανάβει το LED (γραμμή 8) και στην συνέχεια το σβήνει (γραμμή 10). Δύο εντολές delay ρυθμίζουν τον χρόνο που το LED θα μένει αναμμένο ή σβηστό στις γραμμές 9 και 11 (1000ms δηλαδή 1 δευτερόλεπτο).

Για να δείτε το πρόγραμμα στην πράξη, εφόσον έχετε ήδη συνδέσει το Arduino με τον υπολογιστή επιλέξτε File -> Upload to I/O Board (εναλλακτικά πατήστε Ctrl-U ή κάντε κλικ στο ανάλογο εικονίδιο της toolbar). Με αυτή την ενέργεια, το sketch θα μεταγλωττιστεί και θα σταλεί αυτόματα στο Arduino, γεγονός που μπορείτε να επαληθεύσετε από την δραστηριότητα των TX και RX LED πάνω στην πλακέτα του Arduino. Τα προγράμματα που «ανεβάζετε» στο Arduino εκτελούνται αυτόματα από τον bootloader αμέσως μετά την λήψη τους και έτσι, χωρίς καθυστέρηση, θα πρέπει να δείτε το LED με τη σήμανση 13 να ανάβει και να σβήνει συνεχόμενα με περίοδο 2 δευτερολέπτων, δηλαδή όπως ακριβώς ορίζει το sketch.

Αν επιμένετε ότι ένα LED που αναβοσβήνει δεν αποτελεί πρόπον χαιρετισμό και θέλετε σώνει και καλά να δείτε το "Hello World" γραμμένο, υπάρχει μια λύση. Μπορείτε να το στείλετε μέσω της σειριακής (USB) στον υπολογιστή και να το δείτε στην οθόνη σας. Και σαν bonus, το Arduino θα στέλνει και την κατάσταση του LED στον υπολογιστή. Προσθέστε απλά τις γραμμές:

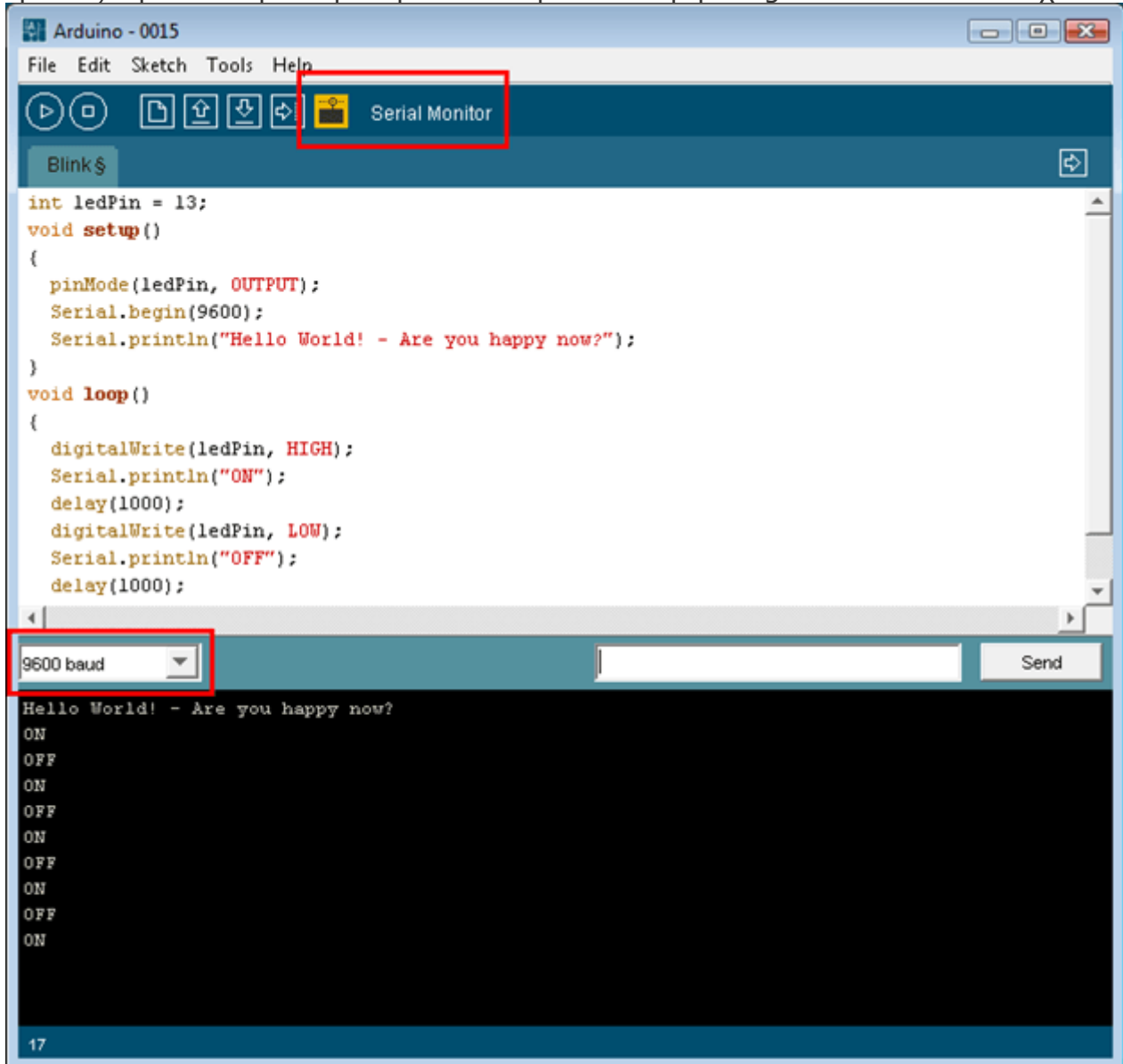
```
Serial.begin(9600);  
Serial.println("Hello World! - Are you happy now?");
```

αμέσως μετά την γραμμή με την εντολή pinMode και πριν κλείσει το άγκιστρο της ρουτίνας setup(). Επίσης, προσθέστε την γραμμή:

```
Serial.println("ON");  
και την γραμμή:
```

```
Serial.println("OFF");
```

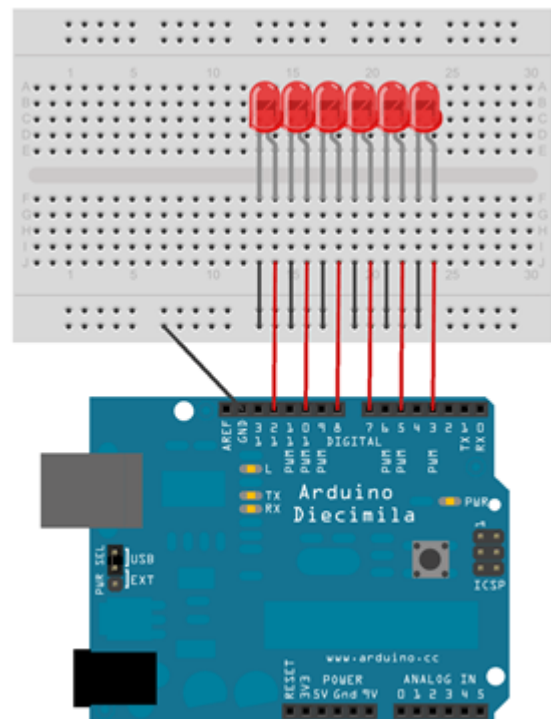
αμέσως μετά την πρώτη και την δεύτερη digitalWrite αντίστοιχα.



Αφού κάνετε τις αλλαγές επιλέξτε όπως και πριν το Upload to I/O Board από το IDE για να γίνει ξανά μεταγλώττιση και να ανέβει το νέο binary στο Arduino.

Αμέσως μετά, κάντε κλικ στο τελευταίο κουμπί της toolbar με επεξήγηση Serial Monitor για να μετατρέψετε το κάτω τμήμα του παραθύρου του IDE σε σειριακή κονσόλα και σύντομα θα δείτε το Arduino να σας στέλνει τα μηνύματά του.

Μπορείτε να πειραματιστείτε με το sketch, να φτιάξετε ωραία pattern με τα οποία αναβοσβήνει το LED, να το βάλετε να στέλνει διαφορετικές πληροφορίες στην σειριακή κ.λπ. αλλά ακόμα και αν μάθετε στο Arduino να σας μιλάει με κώδικα Morse, δεν παύει να είναι ένα LED που αναβοσβήνει και ενίοτε φλυαρεί στην



σειριακή – δεν έχει κάτι σημαντικό να σας πει. Αυτό φυσικά συμβαίνει επειδή δεν έχετε συνδέσει ακόμα περιφερειακά στο Arduino σας και έτσι είναι σαν να έχετε ένα υπολογιστή χωρίς οθόνη, ποντίκι και πληκτρολόγιο. Πλέον όμως έχετε τις απαραίτητες γνώσεις και έτσι στα tutorial που θα ακολουθήσουν στα επόμενα τεύχη τα πράγματα θα γίνουν πολύ πιο ενδιαφέροντα καθώς θα μαθαίνετε πώς να συνδέετε και να αξιοποιείτε διάφορα μεμονωμένα εξαρτήματα ή ολοκληρωμένα shield στο Arduino σας, ώστε να φτιάξετε τις δικές σας real world εφαρμογές. Μέχρι τότε, διασκεδάστε βλέποντας μερικές από τις πιο απίστευτες από αυτές ψάχνοντας με λέξη κλειδί το Arduino στο YouTube και... τα ξαναλέμε...

### **Shield – επειδή δεν είμαστε όλοι ηλεκτρονικοί...**

Τα shield είναι ολοκληρωμένες πλακέτες που είναι σχεδιασμένες ώστε να κουμπώνουν πάνω στο Arduino προεκτείνοντας την λειτουργικότητά του. Είναι η hardware αντίστοιχη έννοια των plugin, addon και extension που υπάρχουν στο software. Μερικά από τα πιο δημοφιλή shield που κυκλοφορούν στο εμπόριο για το Arduino είναι:

- *Ethernet shield*: Δίνει στο Arduino την δυνατότητα να δικτυωθεί σε ένα LAN ή στο internet μέσω ενός τυπικού καλωδίου Ethernet.
- *WiFi shield*: Όμοιο με το Ethernet shield, χωρίς φυσικά το καλώδιο.
- *Διάφορα shield οθόνης*: Προσθέτουν οθόνη στο Arduino. Κυκλοφορούν από απλές οθόνες τύπου calculator μέχρι OLED touchscreen υψηλής ανάλυσης τύπου iPhone.
- *Wave shield*: Δίνει στο Arduino την δυνατότητα να παίζει ήχους/μουσική από κάρτες SD.
- *GPS shield*: Προσθέτει GPS δυνατότητες στο Arduino (εντοπισμό στίγματος).
- *Διάφορα Motor Shields*: Σας επιτρέπουν να οδηγήσετε εύκολα μοτέρ διάφορων τύπων (απλά DC, servo, stepper κ.λπ.) από το Arduino.
- *ProtoShield*: Μια προσχεδιασμένη πλακέτα πρωτοτυποποίησης, συμβατή στις διαστάσεις του Arduino και χωρίς εξαρτήματα για να φτιάξετε το δικό σας shield.





Τα shield είναι σχεδιασμένα ώστε αφού κουμπωθούν πάνω στο Arduino να προωθούν τις υποδοχές του, ώστε να μπορείτε να συνδέσετε επιπλέον τα δικά σας εξαρτήματα ή να κουμπώσετε και επόμενο shield. Φυσικά, το κάθε shield χρησιμοποιεί ορισμένους από τους πόρους συνδεσιμότητας του Arduino και έτσι δεν μπορείτε να συνδέσετε απεριόριστα shield. Μάλιστα κάποια shield μπορεί να μην είναι συμβατά μεταξύ τους γιατί χρησιμοποιούν τα ίδια pin του Arduino για επικοινωνία με αυτό. Επίσης, επειδή κάποια shield δεν προωθούν τις συνδέσεις του Arduino (όπως π.χ. οι οθόνες οι οποίες δεν έχουν νόημα αν τις καλύψετε από πάνω με ένα επόμενο shield), υπάρχουν ειδικά extender shield που κουμπώνουν στο Arduino και δίνουν την δυνατότητα σε δύο άλλα shield να κουμπώσουν πάνω τους, λειτουργώντας σαν πολύπριζα. Όπως και για το ίδιο το Arduino, το βασικό πλεονέκτημα των shield δεν είναι τόσο το προφανές πλεονέκτημα του έτοιμου hardware όσο ότι συνοδεύονται συνήθως από έτοιμες βιβλιοθήκες που σας επιτρέπουν να προγραμματίζετε τα sketch σας σε high level. Έτσι, λόγω χάρη, δεν χρειάζεται να διαβάσετε datasheet ή να γίνετε ηλεκτρονικός για να συνδέσετε και να λειτουργήσετε ένα GPS module πάνω στο Arduino. Απλά συνδέετε το shield, εγκαθιστάτε τη βιβλιοθήκη που το συνοδεύει και χρησιμοποιείτε μια έτοιμη συνάρτηση -του στυλ getLocation- για να πάρετε το γεωγραφικό στίγμα και να το επεξεργαστείτε περαιτέρω στο sketch σας. Τα shield σας λύνουν τα χέρια όταν θέλετε να δημιουργήσετε εύκολα ένα πραγματικά πρακτικό project. Αυτός είναι και ο λόγος που δεν συνιστάται η αγορά κάποιας έκδοσης του Arduino που δεν είναι 100% συμβατή με τα shield.