

Arduino

ΕΙΣΑΓΩΓΗ ΣΤΟ ARDUINO

A. Ανδρεάτος

Κεφ. 1/ Εισαγωγή - Τι είναι το Arduino;

Σύμφωνα με τον δημιουργό του, το Arduino είναι μια πλατφόρμα ανάπτυξης ηλεκτρονικών κατασκευών ανοικτών τεχνολογιών (υλικού και λογισμικού) 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 το Arduino είναι πλακέττα που περιέχει κάποιον μικροελεγκτή ATmega της Atmel, σταθεροποιημένο τροφοδοτικό και κύκλωμα σύνδεσης σε θύρα USB. Τα σχέδια του Arduino 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 χαρακτηρισμός «ανοικτού υλικού»). Το Arduino διαθέτει εισόδους και εξόδους που αντιδρούν όπως προβλέπει το πρόγραμμα που γράψαμε και φορτώσαμε στον μικροελεγκτή με τη βοήθεια του υπολογιστή, μέσω θύρας USB. Η γλώσσα προγραμματισμού που χρησιμοποιεί (Wiring) είναι παρόμοια με την C, εύκολη στη σύνταξη και

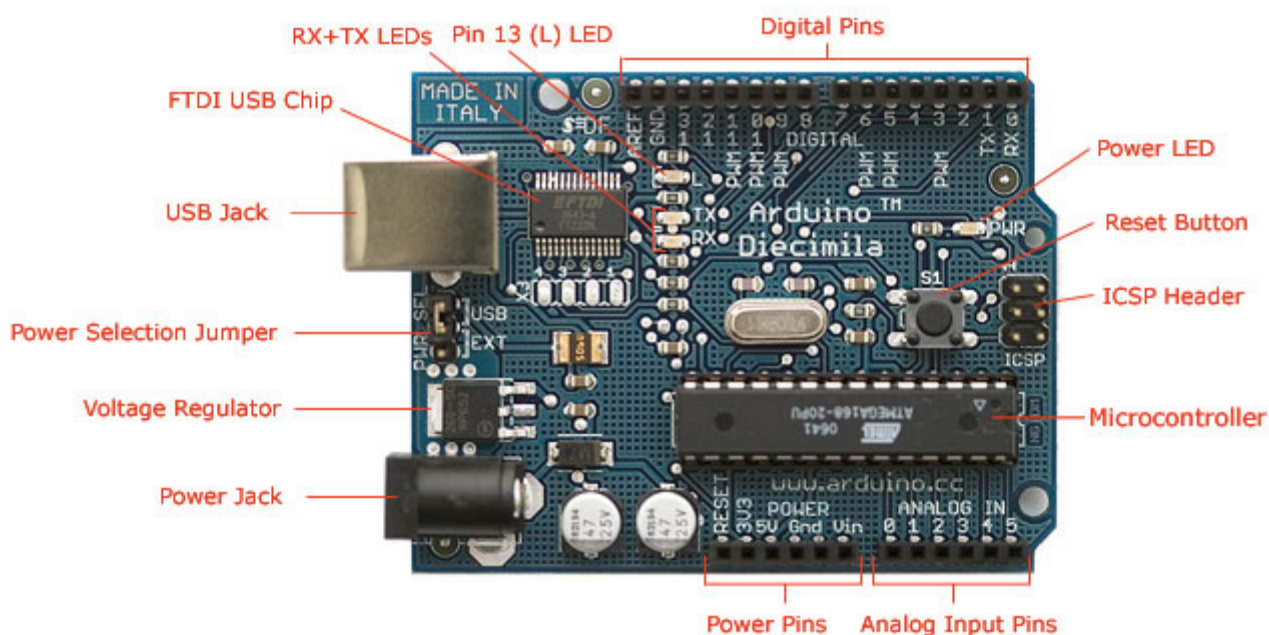
διατίθεται με άδεια χρήσης GPL για πλατφόρμες Linux, MAC και Windows.

Ο χρήστης μπορεί να συνδέσει επάνω στο Arduino πολλαπλές μονάδες εισόδου/ εξόδου όπως αισθητήρες, πλήκτρα, πληκτρολόγιο, οθόνες, LED, ρελαί, τρανζίστορ, κινητήρες κλπ, και να προγραμματίσει τον μικροελεγκτή να δέχεται δεδομένα από τις μονάδες εισόδου, να τα επεξεργάζεται και να στέλνει κατάλληλες εντολές στις μονάδες εξόδου. Η ρομποτική και οι αυτοματισμοί είναι δύο από τις πολλές περιοχές στις οποίες το Arduino διαπρέπει.

Αυτό που έκανε το Arduino ξεχωριστό στην αρχή είναι ότι όλο το κύκλωμα της πλακέτας διατίθεται με άδεια χρήσης Creative Commons, πράγμα που σημαίνει ότι ο καθένας μπορεί να κατασκευάσει την δική του πλακέτα όπως αυτός θέλει. Φυσικά για τους αρχάριους το να κατασκευάσουν την πλακέτα μόνοι τους είναι μάλλον δύσκολο, οπότε είναι προτιμώτερο να αγοράσουν μια έτοιμη πλακέτα Arduino από το Διαδίκτυο η οποία διατίθεται σε πάρα πολύ προσιτή τιμή (6 – 30 €). Πολλοί κατασκευαστές έβγαλαν στη συνέχεια αντίστοιχα προϊόντα σε αντίστοιχες (χαμηλές) τιμές. Αυτό όμως που κάνει το Arduino ξεχωριστό είναι τα χιλιάδες παραδείγματα ανοικτού κώδικα που μπορεί να βρή κανείς είτε ενσωματωμένα στο Arduino IDE είτε στο Διαδίκτυο (στην επίσημη ιστοσελίδα <https://www.arduino.cc> και σε ιστοσελίδες ερασιτεχνών και απλών χρηστών). Ως αποτέλεσμα, μπορούμε εύκολα να κατασκευάσουμε χιλιάδες κατασκευές αναλόγως φυσικά και της εμπειρίας μας στα ηλεκτρονικά.

Οι δυνατότητες που προσφέρει το Arduino είναι πάρα πολλές. Μπορούμε να το χρησιμοποιήσουμε σε εφαρμογές ρομποτικής και γενικότερα σε αυτοματισμούς καταφέροντας έτσι πάρα πολλά όπως: την κίνηση κινητήρων servo, stepper και DC, τη λήψη πληροφοριών από διάφορους αισθητήρες (θερμοκρασίας, υγρασίας, υπερύθρων κ.α), την αμφίδρομη σειριακή επικοινωνία μεταξύ Arduino και PC χρησιμοποιώντας γλώσσες προγραμματισμού (όπως Java και python), την αναπαραγωγή και αντίληψη ήχων και πάρα πολλά ακόμα.

Στο site του Arduino θα ανακαλύψετε πάρα πολλές πληροφορίες όσον αφορά τον προγραμματισμό της πλακέτας. Η πλακέτα Arduino μέχρι στιγμής διατίθεται σε 20 βασικές παραλλαγές οι οποίες αναφέρονται σε διαφορετικές χρήσεις η κάθε μια, ανάλογα με τις ανάγκες της εφαρμογής μας.



Photograph by SparkFun Electronics. Used under the Creative Commons Attribution Share-Alike 3.0 license.

Εικόνα 1-1: Το πρώτο Arduino (Diecimila)

Το Arduino βέβαια, δεν είναι ούτε ο μοναδικός, ούτε και ο καλύτερος δυνατός τρόπος δημιουργίας 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 ανάλογου 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 εκτενή και ανοικτή τεκμηρίωση που το συνοδεύει, καταφέρνει να κερδίσει όλους αυτούς τους χρήστες που διαθέτουν περιορισμένες γνώσεις ηλεκτρονικών.

το Arduino κυκλοφορεί σε διαφορετικές εκδόσεις, επίσημες και ανεπίσημες. Από τις επίσημες εκδόσεις (UNO, Due, Leonardo, Ethernet, Duemilanove, Diecimila, Nano, Mega, Bluetooth, LilyPad, Mini, Mini USB, Pro, Pro Mini, Serial και Serial SS) συνιστάται κυρίως η αγορά του Arduino UNO ή τουλάχιστον των Duemilanove 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.

Εκτός και αν επισημανθεί διαφορετικά, όσα θα αναφερθούν παρακάτω αφορούν τις πλέον δημοφιλείς εκδόσεις του Arduino, δηλαδή το Arduino Duemilanove και το Arduino Uno.

1.1/ Μικροελεγκτής – η καρδιά του Arduino

Το Arduino βασίζεται στον ATmega328, ένα μικροελεγκτή RISC των 8-bit, 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 σ' 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 BIOS.
- 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 (“φορτωτής”) 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).

1.2/ Διαφορές μεταξύ διαφόρων εκδόσεων του Arduino

Το Arduino Diecimila έχει ουσιαστικά δύο βασικές διαφορές με το Duemilanove: 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).



Εικόνα 1-2: Arduino Diecimila, Duemilanove & Mega
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 ATmega128 και αρκετά μεγαλύτερο μέγεθος. Οι διαφορές του από το Duemilanove είναι:

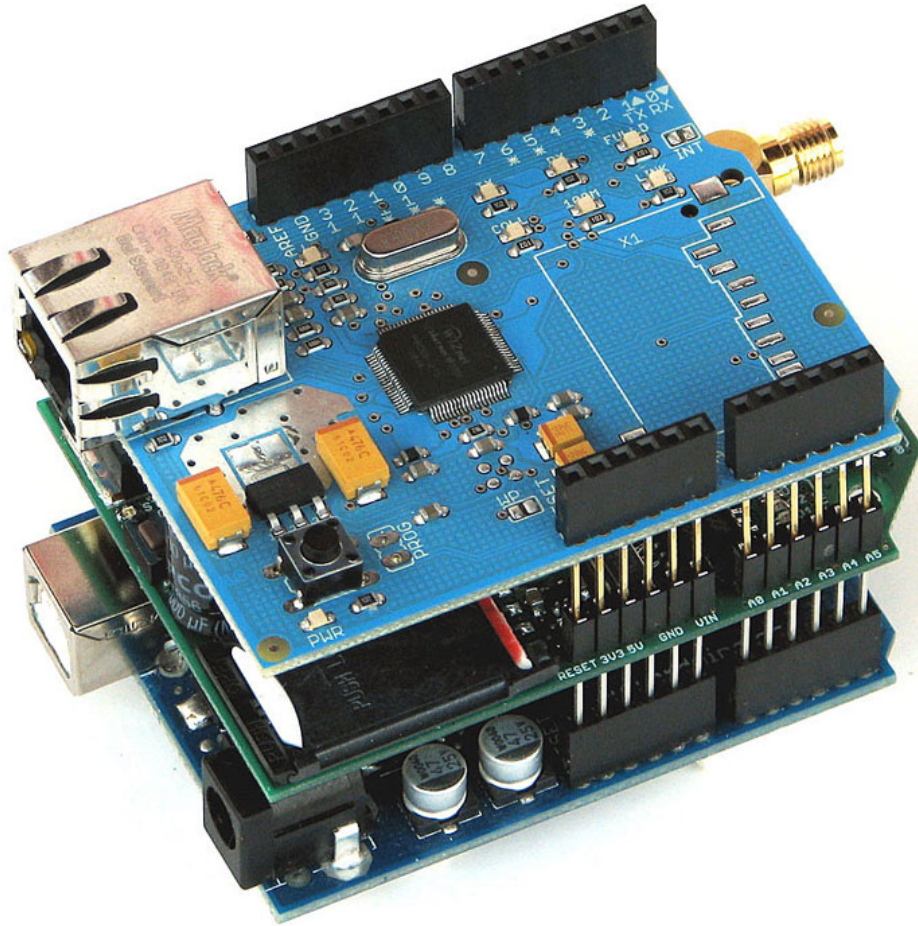
- Τετραπλάσια μνήμη (8Kb SRAM, 4Kb EEPROM, 128Kb Flash).
- 40 επιπλέον ψηφιακά pin εισόδου/εξόδου (σύνολο 54)
- 10 επιπλέον pin αναλογικής εισόδου (σύνολο 16)
- Υποστήριξη ψευδαναλογικής εξόδου PWM σε 8 ακόμα ψηφιακά pin (σύνολο

- 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.

Τα μοντέλλα Diecimila και Duemilanove δεν διατίθενται πλέον. Το σημερινό βασικό μοντέλλο είναι το UNO. Το Mega υπάρχει ακόμα με πιο καινούργιο επεξεργαστή.

1.3/ Επέκταση δυνατοτήτων του Arduino με τα Shield

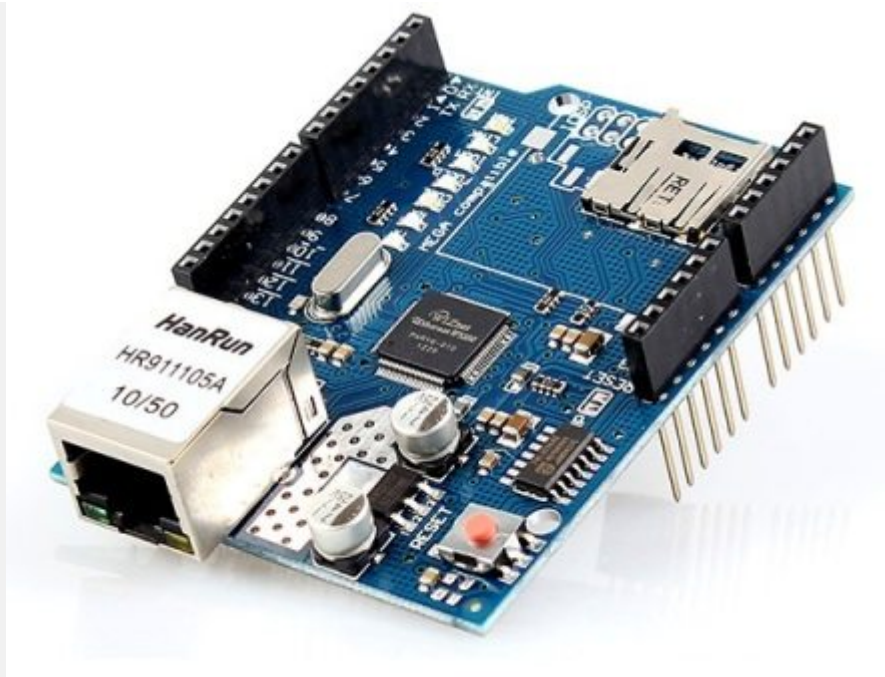
Τα shield είναι ολοκληρωμένες πλακέτες που είναι σχεδιασμένες ώστε να κουμπώνουν πάνω στο Arduino προεκτείνοντας την λειτουργικότητά του. Τα shield είναι σαν τα περιφερειακά ενός ΗΥ: επεκτείνουν τις δυνατότητές του. Τα shield είναι σχεδιασμένα ώστε αφού κουμπωθούν πάνω στο Arduino να προωθούν τις υποδοχές του στην ανώτερη πλακέττα (το shield) ώστε να μπορείτε να συνδέσετε επιπλέον τα δικά σας εξαρτήματα ή να κουμπώσετε και επόμενο shield.



Εικόνα 1-3: Arduino με δύο shield

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. Βασίζεται σε ολοκληρωμένο κύκλωμα ελεγκτή δικτύου [Ethernet controller] που υλοποιεί το πρωτόκολλο Ethernet. Ενσωματώνει υποδοχής κάρτας μνήμης micro-SD. Μπορεί να υλοποιήσει ακόμα και Web server! Συνήθως χρησιμοποιούνται οι εξής δύο ελεγκτές: W5100 και ENC28J60 . Προσφάτως κυκλοφόρησε το Arduino Ethernet που ενσωματώνει τον ελεγκτή δικτύου αλλά το κόστος του είναι σχετικά υψηλό.



Εικόνα 1-4: Arduino Ethernet shield

WiFi shield: Όμοιο με το Ethernet shield, χωρίς φυσικά το καλώδιο.

XBee: Μονάδα ασύρματου

πομποδέκτη. <http://en.wikipedia.org/wiki/XBee>

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.

Διάφορα Motor Shield: Σας επιτρέπουν να οδηγήσετε εύκολα μοτέρ διάφορων τύπων (απλά DC, servo, stepper κ.λπ.) από το Arduino.

ProtoShield: Μια προσχεδιασμένη πλακέτα πειραματισμού και ανάπτυξης προτύπων κατασκευών (“προτυποποίησης”) στις διαστάσεις του Arduino και τρύπες για εξαρτήματα, που μας βοηθάει να φτιάξουμε το δικό μας shield.



Εικόνα 1-5: Arduino με Ethernet 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 μ' 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

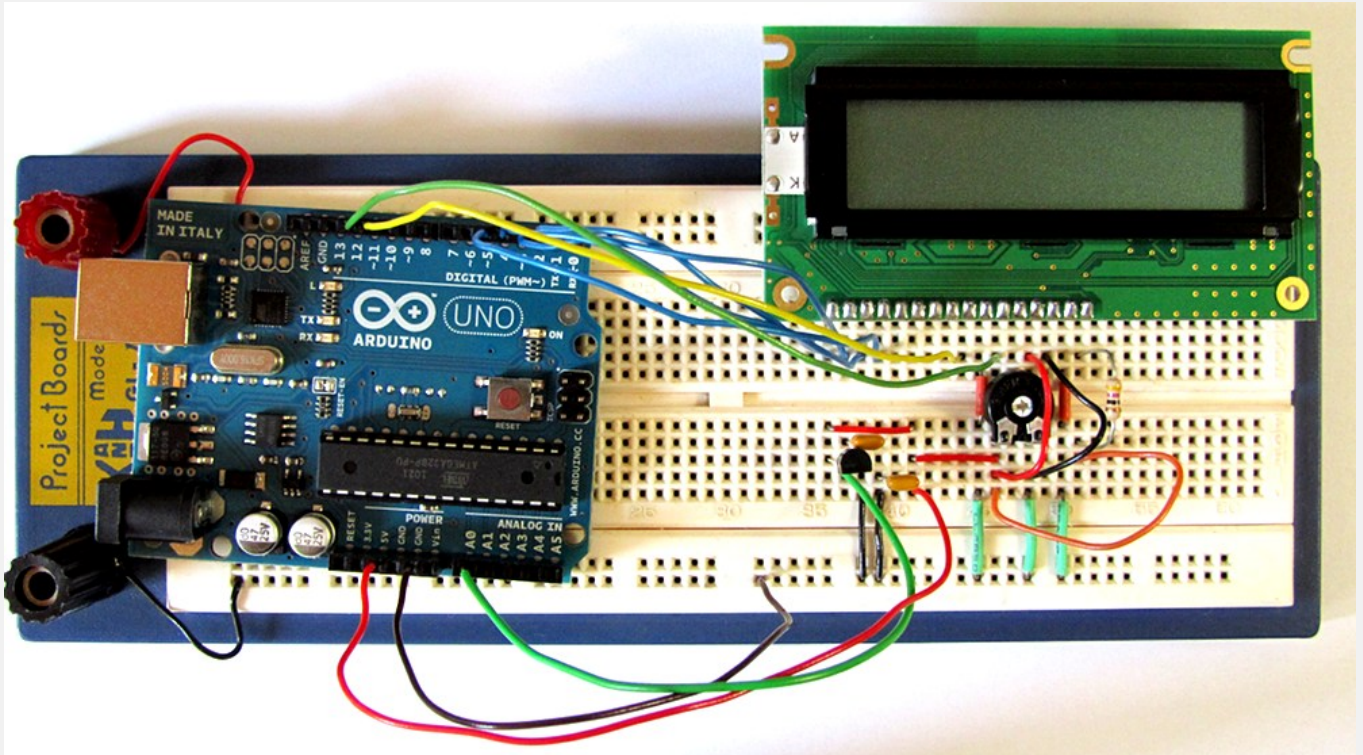
hardware όσο ότι συνοδεύονται συνήθως από έτοιμες βιβλιοθήκες που σας επιτρέπουν να προγραμματίζετε τα sketch σας σε high level.

Έτσι, λόγου 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. 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.

Για κάποιον που θέλει να φτιάξει τη δική του κατασκευή ή ένα εμπορικό προϊόν βασισμένο στον μικροελεγκτή ATmega, τα shield και όλα τα περιφερειακά που μπορεί να συνδέσει στο Arduino μέσω μιας πλακέτας πειραματισμών [breadboard], είναι ένα ενδιάμεσο στάδιο. Όταν καταλήξουμε στο τελικό πρότυπο, το επόμενο βήμα είναι η

κατασκευή μιας δικής μας πλακέτας που θα ενσωματώνει μόνο τα εξαρτήματα που χρειαζόμαστε.

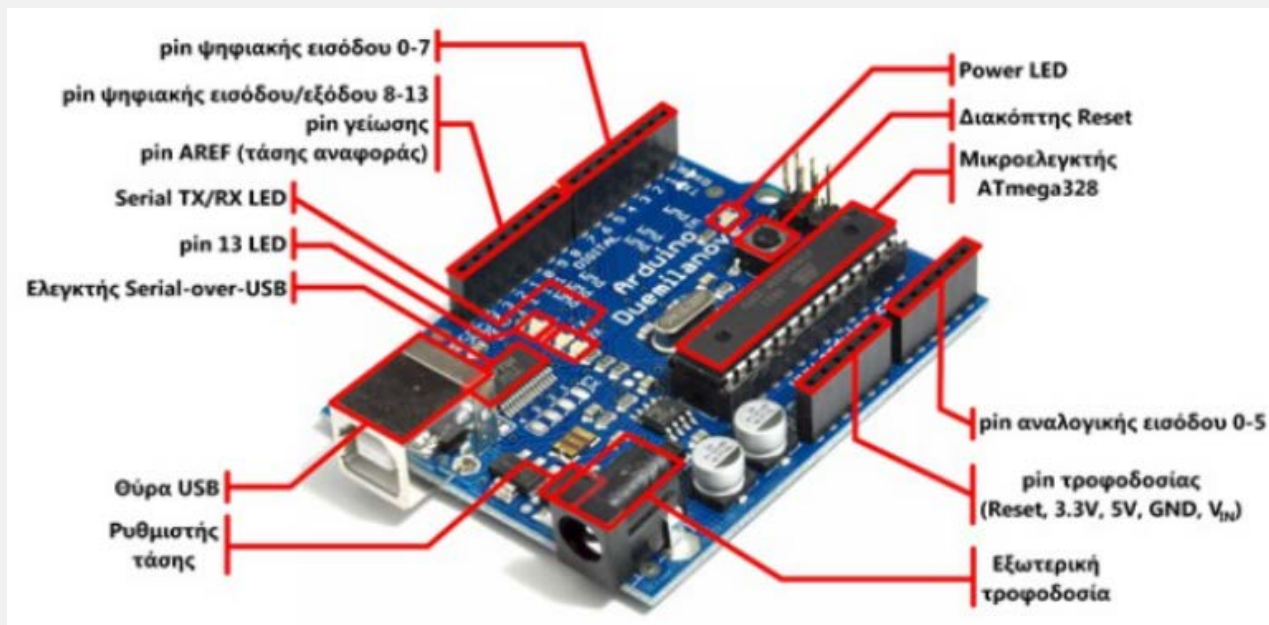


Εικόνα 1-6: Arduino UNO με οθόνη LCD πάνω σε breadboard
Από τις ανεπίσημες εκδόσεις, το Freeduino 1.16 και το Seeduino βασίζονται στο Diecimila οπότε ισχύουν οι ίδιες διαφορές που έχει αυτό με το Duemilanove. Το Freeduino είναι ακριβής κλώνος του Diecimila, ενώ το Seeduino είναι μια βελτιωμένη έκδοση του Diecimila με κύρια διαφορά την προσθήκη 2 επιπλέον pin αναλογικής εισόδου.

1.4/ Είσοδοι – Έξοδοι

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.



Εικόνα 1-7: Ανατομία του Arduino Duemilanove
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. Μ' 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 «διαβάσετε» την κατάσταση ενός διακόπτη).

Μερικά από αυτά τα 14 pin, εκτός από ψηφιακές είσοδοι/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 εξωτερικές είσοδοι διακοπών (interrupt 0 και 1 αντίστοιχα). Μ' άλλα λόγια, μπορείτε να τα ρυθμίσετε μέσα από το πρόγραμμά σας ώστε να λειτουργούν αποκλειστικά ως ψηφιακές είσοδοι 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 ψευδαναλογικές 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 V_{ref} η οποία, αν δεν κάνετε κάποια αλλαγή είναι προρρυθμισμένη 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. Σ' αυτήν την περίπτωση τα pin μετονομάζονται από 0~5 σε 14~19 αντίστοιχα.

1.5/ 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 στα 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 φισ των 2,1mm. Αν όμως έχετε ήδη συνδεδεμένη εξωτερική τροφοδοσία μέσω του φισ, 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.

1.6/ 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.

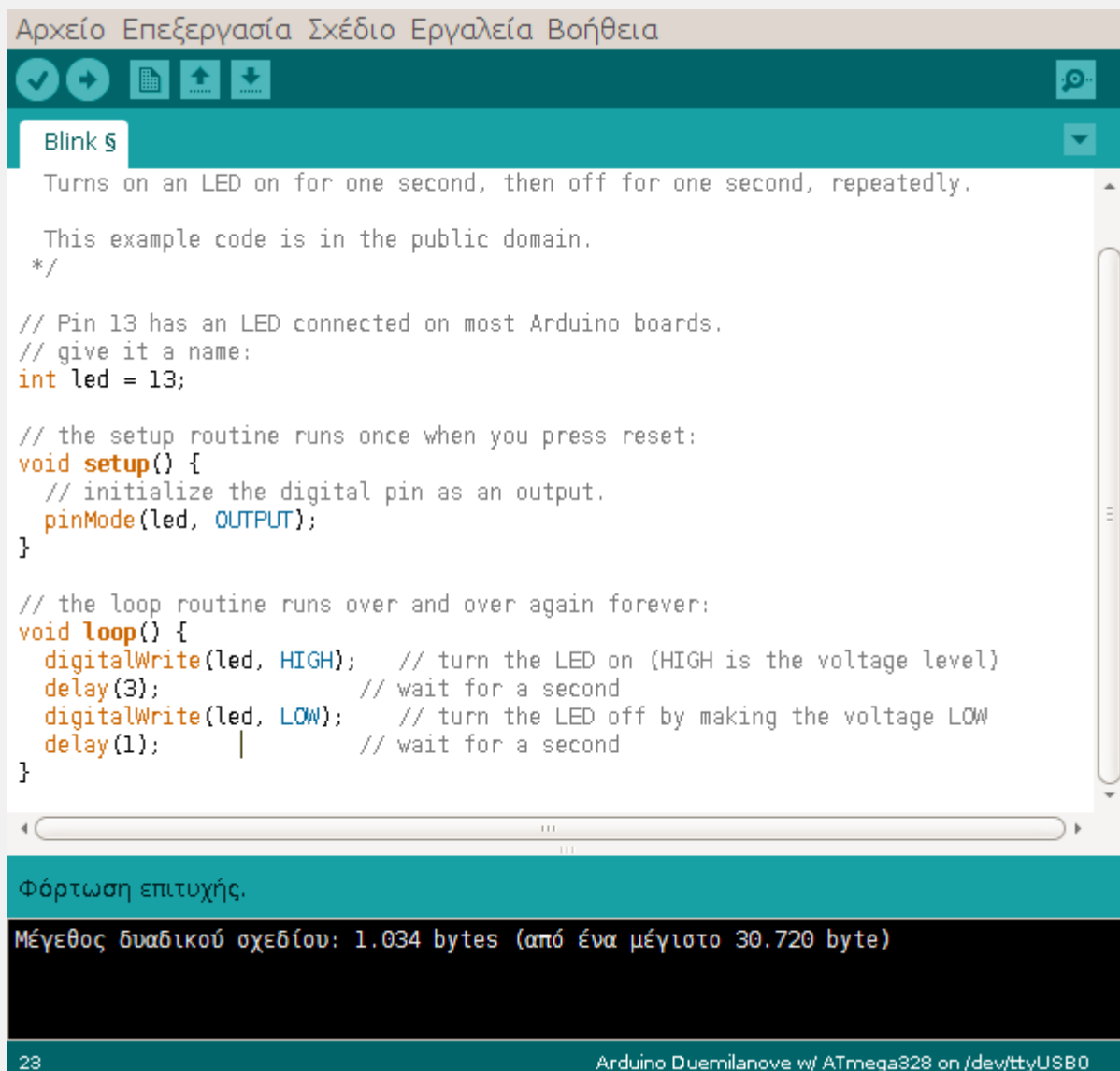
Σημειώστε ότι τα LED αυτά ελέγχονται από τον ελεγκτή Serial-over-USB και συνεπώς δεν λειτουργούν όταν η σειριακή επικοινωνία γίνεται αποκλειστικά μέσω των ψηφιακών pin 0 και 1.

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 συνέδεσαν 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.

Κεφ. 2 Επικοινωνία με τον υπολογιστή - Arduino IDE

2.1/ Το περιβάλλον ανάπτυξης

Το περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java και βασίζεται στο περιβάλλον της γλώσσας Processing (<http://processing.org/>).



```
Αρχείο Επεξεργασία Σχέδιο Εργαλεία Βοήθεια
Blink §
Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.
*/

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(3); // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1); // wait for a second
}

Φόρτωση επιτυχής.
Μέγεθος δυαδικού σχεδίου: 1.034 bytes (από ένα μέγιστο 30.720 byte)
23 Arduino Duernilanove w/ ATmega328 on /dev/ttyUSB0
```

Εικόνα 2-1: Arduino IDE

Ό,τι 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 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.

2.2/ Εγκατάσταση του Arduino IDE

Το Arduino IDE για Mac, Windows και Linux μπορείτε να το κατεβάσετε από το URL:

<http://arduino.cc/en/main/software>

- Στο Linux ποικίλουν οι τρόποι εγκατάστασης του ανάλογα με την διανομή. Σε Ubuntu/ Debian διανομές ανοίξτε το terminal και πληκτρολογήστε “sudo apt-get install arduino”, στη συνέχεια πληκτρολογήστε τον κωδικό σας. Μόλις τελειώσει η διαδικασία το πρόγραμμα έχει εγκατασταθεί. Οι διανομές linux πλέον έχουν όλες σχεδόν κάποιο gui εργαλείο για software managing οπότε μπορείτε πολύ απλά να πάτε εκεί και να το βρείτε σαν arduino ide. Για περισσότερες

πληροφορίες <http://www.arduino.cc/playground/Learning/Linux>.

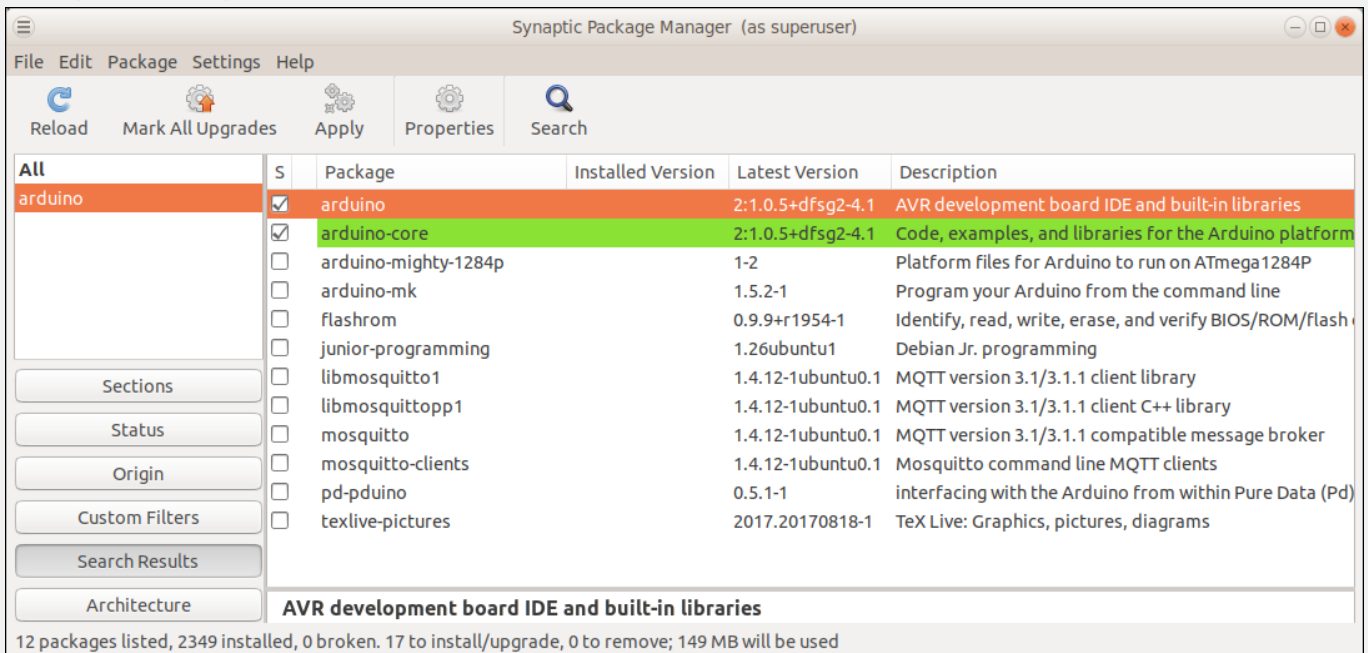
- Το αρχείο που χρειάζεστε σε Mac είναι ένα disk image (dmg). Κάντε διπλό κλικ στο αρχείο αυτό όταν το κατεβάσετε από το Internet. Μέσα στο dmg βρίσκεται η εφαρμογή για το Arduino, αντιγράψτε το κάπου (δεν είναι καλό να το τρέξετε από το dmg) και κάντε διπλό κλικ πάνω του. Το παράθυρο του προγράμματος θα εμφανιστεί.
- Το αρχείο που θα κατεβάσετε για τα windows είναι ένα zip file. Αποσυμπιέστε το αρχείο zip σε οποιαδήποτε κατάλογο π.χ. Program Files/Arduino. Η αποσυμπίεση του αρχείου θα δημιουργήσει έναν φάκελο με όνομα Arduino-00<nn> (όπου <nn> είναι ο αριθμός της έκδοσης του Arduino που κατεβάσατε). Ο κατάλογος αυτός περιέχει το εκτελέσιμο αρχείο Arduino.exe. Κάντε κλικ δύο φορές πάνω του και το παράθυρο του κύριου προγράμματος θα εμφανιστεί. Χρειάζεται επίσης να κατεβάσετε και να εγκαταστήσετε τους drivers του FTDI chip που ταιριάζουν στην έκδοση windows που διαθέτετε (XP/7/..., 32/64 bit κλπ).

Όλα τα καινούργια λειτουργικά συστήματα αναγνωρίζουν αυτόματα το arduino οπότε δεν θα αντιμετωπίσετε κάποιο πρόβλημα με τους drivers.

2.3/ Εγκατάσταση του Arduino IDE σε Linux τύπου deb

Ακολουθούμε τα παρακάτω βήματα:

1. Μέσω Synaptic εγκαθιστούμε Java και Arduino. Το τελευταίο θα συμπεριλάβει τον gcc-avr [gcc compiler για Arduino] και το Arduino IDE. Θα εγκατασταθεί στη θέση: `/usr/share/arduino`.



Εικ. 2-2: Εγκατάσταση Arduino IDE μέσω Synaptic

2. Εναλλακτικά: Κατεβάζουμε τον κώδικα για τη διανομή Λίνουξ που διαθέτουμε [32 ή 64 bit].

3. Αποσυμπιέζουμε το αρχείο σε θέση της επιλογής μας [π.χ. στο /home] ως διαχειριστής για πλήρη δικαιώματα εγγραφής. Θα ερωτηθούμε να προστεθεί ο χρήστης μας σε κατάλληλη ομάδα δικαιωμάτων.

Απαντάμε θετικά και δίνουμε τον κωδικό διαχειριστή:

Κάνουμε επανεκκίνηση.

3. Η εκτέλεση του Arduino IDE γίνεται μέσω του μενού : Εφαρμογές > Προγραμματισμός [ή Ηλεκτρονικά] > Arduino IDE. Εναλλακτικά, εκτελούμε τον κώδικα πατώντας με δεξί ποντίκι πάνω στο αρχείο "Arduino" και εκτελείται το script [run]. Εμφανίζεται το παρακάτω παράθυρο:

Εικ. 2-3: Το περιβάλλον της εφαρμογής σε διανομές Linux

2.4/ Ρυθμίσεις του περιβάλλοντος ανάπτυξης

Συνδέουμε την πλακέτα μας στη θύρα USB. Από το περιβάλλον του Arduino IDE επιλέγουμε Tools > Board > {επιλέγουμε το μοντέλο μας}: Οι βασικές ρυθμίσεις που πρέπει να κάνουμε από την στιγμή που ενώσουμε το Arduino στο σύστημα μας είναι:

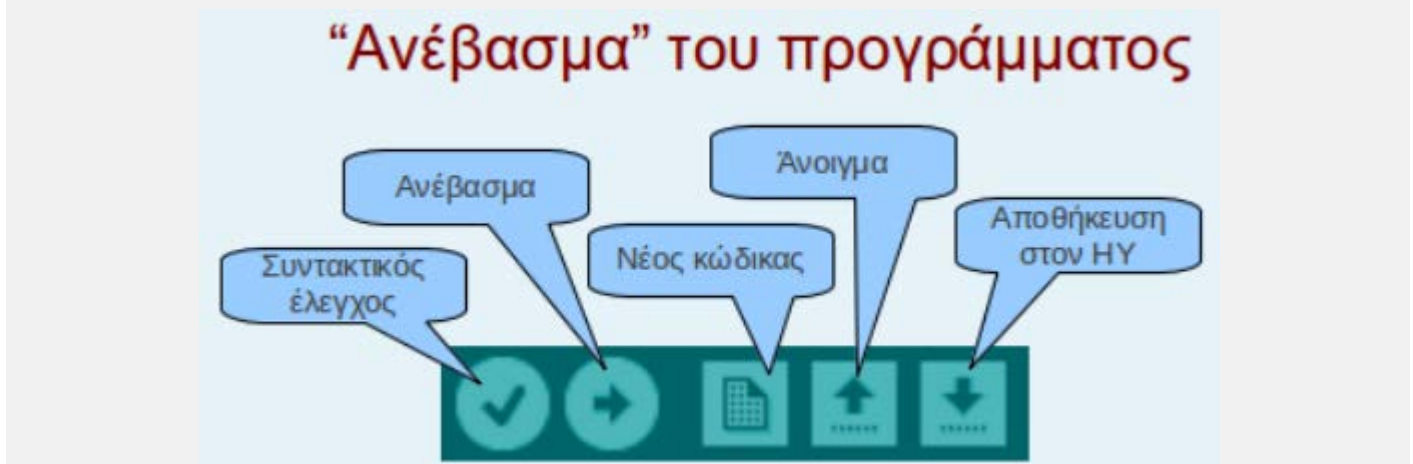
1. Επιλογή πλακέτας. Από το μενού Tools -> Board επιλέγουμε την πλακέτα που έχουμε. Στο συγκεκριμένο οδηγό θα χρησιμοποιήσουμε το Arduino Diecimila, οπότε επιλέγουμε το “Arduino Diecimila, Duemilanove or Nano w/ Atmega168”.
2. Επιλογή σειριακής θύρας. Στη συνέχεια επιλέγουμε με τον ίδιο τρόπο τη θύρα USB (θα είναι κάτι σαν ttyXport). Από το μενού Tools -> Serial Port επιλέγουμε την σειριακή θύρα ή θύρα USB που έχουμε συνδεδεμένο το Arduino (πχ. /dev/ttyUSB0 σε ΛΣ Linux). Σε Windows θα εμφανίζεται ως σειριακή θύρα (π.χ. COM6).

Ρυθμίσεις που αφορούν το μέγεθος του κειμένου, τον φάκελλο αποθήκευσης, χρήση εξωτερικού κειμενογράφου βρίσκονται στη καρτέλα Preferences (File -> Preferences). Για περισσότερες ρυθμίσεις μπορούμε να κάνουμε αλλαγές

το αρχείο preferences.txt (βρίσκεται στον φάκελο του χρήστη (~/.arduino/preferences.txt στο ΛΣ Linux).

Εικ. 2-4: Το περιβάλλον της εφαρμογής σε διανομές Linux
Παρατήρηση: Αν η επιλογή Serial Port του μενού Tools είναι αχνή (όπως στην ανωτέρω εικόνα), σημαίνει πως δεν έχει αναγνωριστεί η θύρα του Arduino. Εφ' όσον έχουμε εγκαταστήσει όλα τα λογισμικά, χρειάζεται επανεκκίνηση.

2.5/ Βασικές λειτουργίες του IDE



Εικ. 2-5: Arduino IDE

2.6/ Χρήση IDE για ετοιμασία ενός Arduino sketch (κώδικα)

Χρησιμοποιήστε το arduino IDE για να κατασκευάσετε, ανοίξετε και τροποποιήσετε sketches που καθορίζουν τι θα κάνει το arduino. Μπορείτε να χρησιμοποιήσετε κουμπιά που βρίσκονται στην κορυφή του IDE για να πραγματοποιήσετε αυτές τις ενέργειες ή τα μενού και τις συντομεύσεις πληκτρολογίου.

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, 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. 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, πρέπει να εμφανίζεται η εικονική σειριακή θύρα (συνήθως COM# για τα Windows, /dev/ttyusbserial## για το MacOS και /dev/ttyUSB## ή /dev/ttyACM## 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>.

2.7/ Εγκατάσταση βιβλιοθηκών

Σε πολλές εφαρμογές απαιτείται το κατέβασμα και η εγκατάσταση βιβλιοθηκών. Αυτές πρέπει να μπουν στη θέση `/usr/share/arduino/libraries`. Γι' αυτό ενεργούμε ως διαχειριστής. Στον φάκελλο `libraries` δημιουργούμε ένα νέο φάκελλο κατά το όνομα της βιβλιοθήκης, π.χ. `Tnout` και αποσυμπιέζουμε εκεί το αρχείο.

Στη συνέχεια, από το Arduino IDE επιλέγουμε την διαδρομή της βιβλιοθήκης που εγκαταστήσαμε και αναζητούμε τον κώδικα (αρχείο `pde` ή `ino` που είναι η νεώτερη ονομασία του κώδικα Arduino).

Κεφ. 3 Γλώσσα προγραμματισμού

Η γλώσσα του Arduino προέρχεται από τις γλώσσες `Wiring` και `Processing`. Η γλώσσα `Wiring` είναι 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++`.

Στην ουσία χρησιμοποιεί κάποιες "έτοιμες" συναρτήσεις οι οποίες είναι γραμμένες σε `C/C++` και συγκεκριμένα στον μεταγλωττιστή `avr-g++` (<http://www.nongnu.org/avr-libc/user-manual/>). Αυτό δεν μας εμποδίζει στο να γράφουμε τις εντολές κατευθείαν σε `C/C++` (βλ. και άρθρα

περιοδικών βιβλιογραφίας), αφού έτσι και αλλιώς ο κώδικας μεταφράζεται σε C πριν μεταγλωττιστεί και μεταφερθεί στο Arduino. Ένα χρήσιμο κείμενο σχετικά με την όλη διαδικασία μπορείτε να διαβάσετε εδώ: Arduino Build Process -

<http://arduino.cc/en/Hacking/BuildProcess>

Σε πλατφόρμα Λίνουξ ως compiler χρησιμοποιείται ο AVR gcc και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc. Γι' αυτό και πρέπει να τις εγκαταστήσουμε στην αρχή.

Λόγω της καταγωγής της από την C, στην γλώσσα του Arduino μπορούμε να χρησιμοποιούμε ουσιαστικά τις ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους τύπων δεδομένων και τους ίδιους τελεστές όπως και στην C. Επιπλέον, 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:

Πίνακας 3-1

Όρισμα	Είδος	Τύπος	Παράμετροι	Περιγραφή
LOW	Σταθερά	int	-	Έχει τιμή 0 και είναι αντίστοιχη του λογικού 0 (ψευδές)
HIGH	Σταθερά	int	-	Έχει τιμή 1 και είναι αντίστοιχη του λογικού 1 (αληθές)
INPUT	Σταθερά	int	-	Έχει τιμή 0 και είναι αντίστοιχη του λογικού 0

OUTPUT	Σταθερά	-	Έχει τιμή 1 και είναι αντίστοιχη του λογικού 1
pinMode	Εντολή-	(pin,mode)	Καθορίζει αν το συγκεκριμένο ψηφιακό pin θα είναι pin εισόδου ή pin εξόδου ανάλογα με την τιμή που δίνεται στην παράμετρο mode (INPUT ή OUTPUT αντίστοιχα)
digitalWrite	Εντολή-	(pin,pinstatus)	Θέτει την κατάσταση pinstatus (HIGH ή LOW) στο συγκεκριμένο ψηφιακό pin
digitalRead	Συνάρτηση	(pin)	Επιστρέφει την κατάσταση του συγκεκριμένου ψηφιακού pin (0 για LOW, 1 για HIGH)
attachInterrupt	Εντολή-	(interrupt, function, management)	Θέτει σε λειτουργία το συγκεκριμένο interrupt, ώστε να ενεργοποιεί την συνάρτηση function, κάθε φορά που ικανοποιείται η συνθήκη που ορίζεται από την παράμετρο triggermode: LOW (ενεργοποίηση όταν η κατάσταση του pin που αντιστοιχεί στο συγκεκριμένο interrupt γίνει LOW) RISING (όταν

			από LOW γίνεται HIGH) FALLING (όταν από HIGH γίνεται LOW) CHANGE (όταν αλλάξει κατάσταση)
analogRef erence	Εντολή- (type)		Δέχεται τις τιμές DEFAULT, INTERNAL ή EXTERNAL ως παράμετρο (type) για να καθορίσει την τάση αναφοράς (Vref) των αναλογικών εισόδων (5V, 1,1V ή η εξωτερική τάση του pin AREF αντίστοιχα)
analogRea d	Συνάρτι nt ηση	(pin)	Επιστρέφει έναν ακέραιο από 0 έως 1023, ανάλογα με την τάση που τροφοδοτείται το συγκεκριμένο pin αναλογικ ής εισόδου στην κλίμακα 0 ως Vref
analogWri te	Εντολή- (pin, value)		Θέτει το συγκεκριμένο ψηφιακό pin σε κατάσταση ψευδοαναλογικής εξόδου (PWM). Η παράμετρος value καθορίζει ι το πλάτος του παλμού σε σχέση με την περίοδο του παραγόμενου σήματος στην κλίμακα από 0 ως 255 (π.χ. Με value 127, το

millis	Συνάρτηση unsigned long	<p>πλάτος του παλμού είναι ίσο με την ημιπερίοδο) Μετρητής που επιστέφει το χρονικό διάστημα σε ms από την στιγμή που άρχισε η εκτελεστή του προγράμματος. Λόγω του τύπου μεταβλητής (unsigned long δηλ. 32 bit) θα υπερχειλίσει σε 2^{32} ms (περίπου 50 ημέρες), οπότε ο μετρητής θα ξεκινήσει πάλι από το μηδέν</p>
delay	Εντολή- (time)	<p>Σταματά προσωρινά την ροή του προγράμματος για time ms. Η παράμετρος time 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.</p>

3.1/ Γενική δομή προγράμματος Arduino

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...
```

Όπως βλέπουμε υπάρχουν δυο βασικές συναρτήσεις σε ένα τυπικό πρόγραμμα. 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 βρόχος `while(true)`.

Η συνάρτηση `setup()` εκτελείται στην αρχή του προγράμματος και για μία μόνο φορά. Χρησιμοποιείται για τις αρχικοποιήσεις των μεταβλητών, τις δηλώσεις των `pin` (αν θα είναι είσοδος ή έξοδος) και τις αρχικοποιήσεις των βιβλιοθηκών.

Η συνάρτηση loop() κάνει αυτό που λέει και το όνομά της, Ο κώδικας που γράφεται μέσα στη συνάρτηση αυτή επαναλαμβάνεται συνεχώς δίνοντας την δυνατότητα στο πρόγραμμα μας να αλλάζει τιμές και το Arduino να ανταποκρίνεται ανάλογα.

Αν και πρόκειται μόνο για τις πιο βασικές λειτουργίες της γλώσσας του Arduino, με αυτές και με λίγες βασικές γνώσεις C θα μπορέσετε 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.

3.2/ Δομή προγράμματος

Ένα τυπικό πρόγραμμα του Arduino έχει την εξής δομή:

```
// δηλώσεις μεταβλητών
```

```
void setup() {
```

```
// αρχικοποιήσεις
```

```
}
```

```
void loop() {
```

```
// ...
```

```
}
```

3.3/ Μεταβλητές

Μεταβλητή στις γλώσσες προγραμματισμού γενικά ονομάζουμε ένα γλωσσικό αντικείμενο που μπορεί να λάβει διάφορες τιμές, μία κάθε φορά. Οι τιμές μιας μεταβλητής περιορίζονται συνήθως σε ένα τύπο δεδομένων.

3.4/ Τύποι δεδομένων

Οι βασικοί τύποι δεδομένων του Arduino είναι:

1. byte: αποθηκεύει μια αριθμητική τιμή 8-bit χωρίς δεκαδικά ψηφία, παίρνουν τιμές από 0 μέχρι

255.

2. int: ακραίοι, παίρνουν τιμές από -32,768 μέχρι 32767.

3. long: μεγάλου μεγέθους ακαριαίοι, παίρνουν τιμές από -2,147,483,648 μέχρι 2,147,483,647

4. float: πραγματικοί αριθμοί, παίρνουν τιμές από 3.4×10^{-38} μέχρι 3.4×10^{38}

Τις μεταβλητές μπορούμε να τις δηλώσουμε στην αρχή του προγράμματός μας:

```
int myvariable;
```

Μπορούμε επίσης να δώσουμε αρχική τιμή στη μεταβλητή ταυτόχρονα με τη δήλωσή της:

```
int myvariable = 47;
```

5. Πίνακες. Πίνακα ονομάζουμε διάταξη δεδομένων μιας ή περισσότερων διαστάσεων η οποία είναι συγκεκριμένου τύπου δεδομένων. Για παράδειγμα αν έχουμε ένα πίνακα ακραίων 5 θέσεων τον οποίο ονομάζουμε `myarray` τον δηλώνουμε όπως βλέπουμε παρακάτω:

```
int myarray[5];
```

Για να δώσουμε τιμή στο τέταρτο στοιχείο του πίνακα `myarray` γράφουμε:

```
myarray[3] = 12;
```

επίσης μπορούμε να γεμίσουμε τον πίνακα ταυτόχρονα με την δήλωση του:

```
int myarray[] = {12, 45, 32, 61, 55};
```

3.5/ Σταθερές

Σταθερές είναι αντικείμενα τα οποία παίρνουν μόνο μία τιμή, και δηλώνονται μαζί με τις μεταβλητές: πχ: `#define LEDPin 13`.

3.6/ Αριθμητικοί τελεστές

Οι αριθμητικοί τελεστές καλύπτουν τις βασικές πράξεις: πρόσθεση, αφαίρεση, πολλαπλασιασμό, διαίρεση (+, -, *, /). Για παράδειγμα μπορούμε να κάνουμε την πρόσθεση δύο ακέραιων και το αποτέλεσμα να εκχωρηθεί σε μία μεταβλητή:

```
sum = 458 + 954;
```

3.7/ Τελεστές σύγκρισης

Με τους τελεστές σύγκρισης μπορούμε να ελέγξουμε αν μία συγκεκριμένη συνθήκη μεταξύ

μεταβλητών ή σταθερών είναι “Αληθής”. Ποιο συγκεκριμένα υπάρχουν οι παρακάτω τελεστές

σύγκρισης στο Arduino:

$x == y$ το x είναι ίσο με το y

$x != y$ το x είναι άνισο του y

$x < y$ το x είναι μικρότερο με το y

$x > y$ το x είναι μεγαλύτερο με το y

$x <= y$ το x είναι μικρότερο ή ίσο με το y

$x >= y$ το x είναι μεγαλύτερο ή ίσο με το y

3.8/ Λογικοί τελεστές

Με τους λογικούς τελεστές μπορούμε να συγκρίνουμε δύο ή περισσότερες εκφράσεις, δίνοντας

αποτέλεσμα “Αληθής” ή “Ψευδής”. Υπάρχουν τρεις λογικοί τελεστές:

- Λογικό ΚΑΙ $\&\&$ - επιστρέφει “Αληθής” αν όλες οι εκφράσεις είναι “Αληθείς”
- Λογικό Ή $\|\|$ - επιστρέφει “Αληθής” αν μία από τις εκφράσεις είναι “Αληθείς”

- Λογικό ΟΧΙ ! - επιστρέφει “Αληθής” αν η έκφραση είναι “Ψευδής”.

Παράδειγμα:

```
if(x > 0 && x < 5){
```

```
//κώδικας
```

```
}
```

Στο παραπάνω κομμάτι κώδικα γίνεται έλεγχος αν το x είναι μεγαλύτερο από το 0 ΚΑΙ μικρότερο από 5 τότε εκτελείται ο κώδικας που βρίσκεται μέσα στις αγκύλες. Με λίγα λόγια η πρόταση If() ελέγχει αν η συνθήκη μέσα στις παρενθέσεις είναι “Αληθής”.

Ένα άλλο παράδειγμα:

```
if(!x > 0){
```

```
//κώδικας
```

```
}
```

Εδώ ελέγχεται αν το x είναι μεγαλύτερο από 0, αν αυτή η συνθήκη ΔΕΝ ισχύει τότε έχουμε το αποτέλεσμα “Αληθής” και εκτελείται ο κώδικας μέσα στις αγκύλες.

3.9/ Ψηφιακή Έξοδος

Το Arduino Diecimila αποτελείται από δεκατρία ψηφιακά pin, τα οποία μπορούμε να τα χρησιμοποιήσουμε το κάθENA ξεχωριστά, είτε για είσοδο είτε για έξοδο. Μπορούμε να τα

προγραμματίσουμε να συμπεριφέρονται όπως εμείς θέλουμε, αρκεί να κάνουμε τις σωστές δηλώσεις στον κώδικα που θα φορτώσουμε στην πλακέτα.

Η έξοδος του κάθε pin μπορεί να προγραμματιστεί να δίνει τιμές HIGH ή LOW. Λέγοντας HIGH ενώνουμε το δυαδικό '1' και έχουμε τάση εξόδου 5V DC, ενώ το LOW είναι το δυαδικό '0' και έχει τάση εξόδου 0V DC (ground). Παρακάτω θα δούμε ένα παράδειγμα για να μπορέσουμε να κατανοήσουμε τον προγραμματισμό των ψηφιακών pin του Arduino. Θα ενώσουμε ένα LED στη board και θα το προγραμματίσουμε να ανάβει και να σβήνει σε χρονικά διαστήματα ενός δευτερολέπτου.

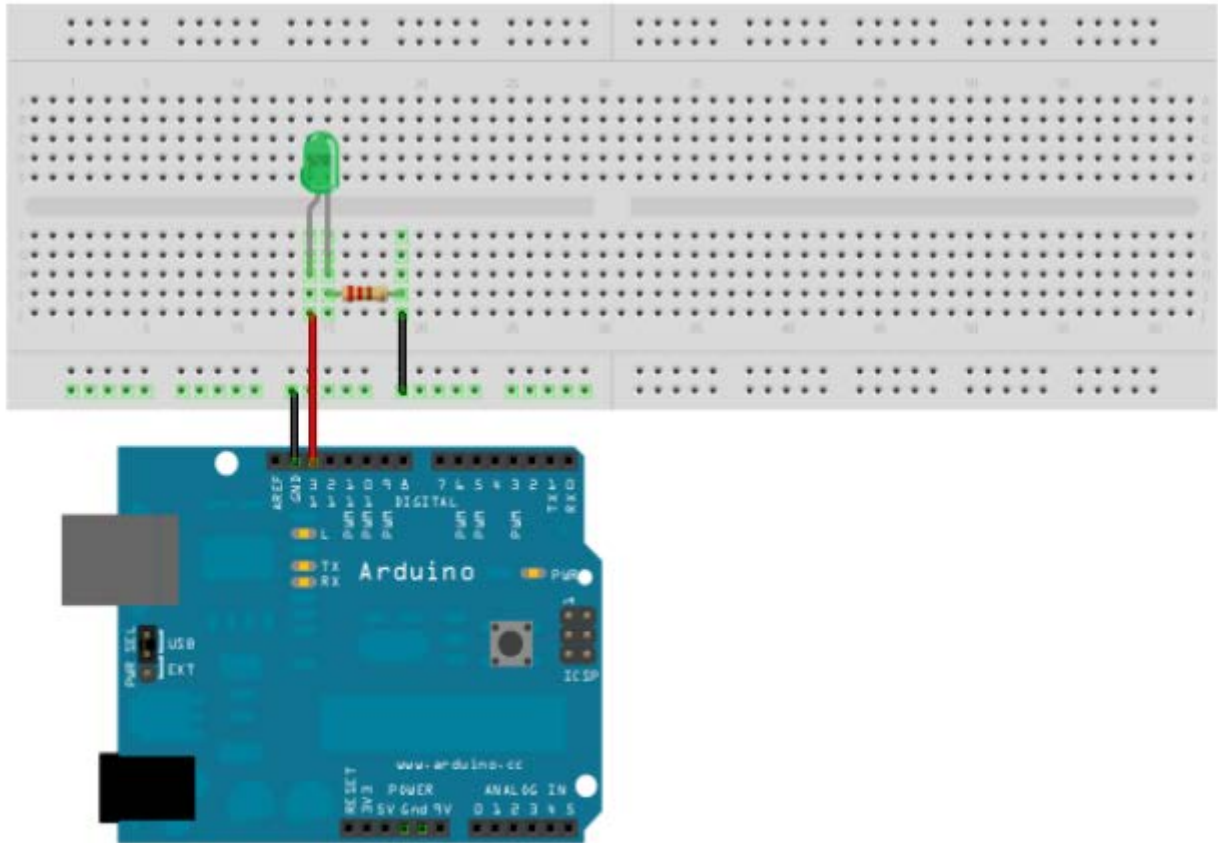
Τα υλικά που θα χρειαστούμε στη πρώτη εφαρμογή είναι:

1 breadboard

1 LED

1 αντίσταση 220Ω (κόκκινο – κόκκινο – καφέ).

Η συνδεσμολογία που κάναμε φαίνεται στο παρακάτω σχήμα:



Εικόνα 3: Φυσική αναπαράσταση του κυκλώματος

Εικόνα 3-1: Αναπαράσταση α' κυκλώματος

Για την φυσική αναπαράσταση του κυκλώματος έγινε χρήση του προγράμματος Fritzing (<http://fritzing.org/>). Ας μελετήσουμε τώρα το πρόγραμμα που θα φορτώσουμε στο Arduino:

```
int LEDPin = 13;

void setup(){

pinMode(LEDPin, OUTPUT);

}

void loop(){

digitalWrite(LEDPin, HIGH);
```

```
delay(1000);  
  
digitalWrite(LEDPin, LOW);  
  
delay(1000);  
  
}
```

Στη πρώτη γραμμή δηλώνουμε μια ακέραια μεταβλητή την LEDPin και της δίνουμε την αρχική τιμή 13, ο λόγος που έγινε αυτή η δήλωση είναι για να μας διευκολύνει αν δεν θυμόμαστε σε ποιο pin συνδέσαμε το LED.

Στη συνάρτηση setup() η εντολή pinMode() χρησιμοποιείτε για να δηλώσουμε την χρήση του pin 13, δηλαδή αν θα το χρησιμοποιήσουμε σαν είσοδο (input) ή σαν έξοδο (output). Ας πάμε τώρα στη συνάρτηση loop() που όπως είπαμε ποιο πριν χρησιμοποιείτε για την επανάληψη του κώδικα, δίνοντάς μας τη ευκαιρία αν θέλουμε να ανανεώσουμε τις τιμές των μεταβλητών και να κάνουμε το Arduino να ανταποκρίνεται στις αλλαγές αυτές.

Εδώ συναντάμε την εντολή digitalWrite() που είναι υπεύθυνη στο να δώσει τιμές HIGH ή LOW στο pin που δηλώνουμε μέσα στις παρενθέσεις. Η εντολή delay() είναι υπεύθυνη για την παύση του προγράμματός μας για χρόνο που δηλώνουμε μέσα στις παρενθέσεις, το χρόνο αυτό τον καθορίζουμε σε microseconds (1000 ms = 1sec).

“Ανέβασμα” του προγράμματος



Εικόνα 3-2: Τα πλήκτρα του Arduino IDE

Αφού ελέγξουμε τον κώδικά για τυχόν λάθη πατώντας το κουμπί του συντακτικού ελέγχου (✓), πατάμε το κουμπί “ανεβάσματος” (==>) για να γίνει η μεταγλώττιση και το ανέβασμα του δεκαεξαδικού αρχείου στο μικροελεγκτή. Αν όλα πήγαν καλά τότε θα δούμε το LED να αναβοσβήνει σε διαστήματα ενός δευτερολέπτου.

3.10 Ψηφιακή Είσοδος

Στο προηγούμενο παράδειγμα είδαμε πώς μπορούμε να δηλώσουμε ένα pin του Arduino και να το χρησιμοποιήσουμε σαν έξοδο. Τώρα θα μελετήσουμε τον τρόπο που μπορούμε να εισάγουμε δεδομένα σε ένα pin. Όπως είδαμε στην έξοδο, έτσι και στην είσοδο η τιμές που μπορεί να δεχτεί ένα pin είναι HIGH ή LOW. Την τιμή HIGH μπορούμε να την πάρουμε από το pin 5V του Arduino ενώ την τιμή LOW μπορούμε να την πάρουμε από τα pin GND (ground, 0V). Για την επιλογή της τιμής 0V ή 5V θα μας βοηθήσει η χρήση ενός διακόπτη. Παρακάτω θα προσπαθήσουμε με την βοήθεια ενός διακόπτη να ανάβουμε ένα LED για ένα δευτερόλεπτο. Τα υλικά που θα χρειαστούμε σ' αυτήν την εφαρμογή είναι:

1 breadboard

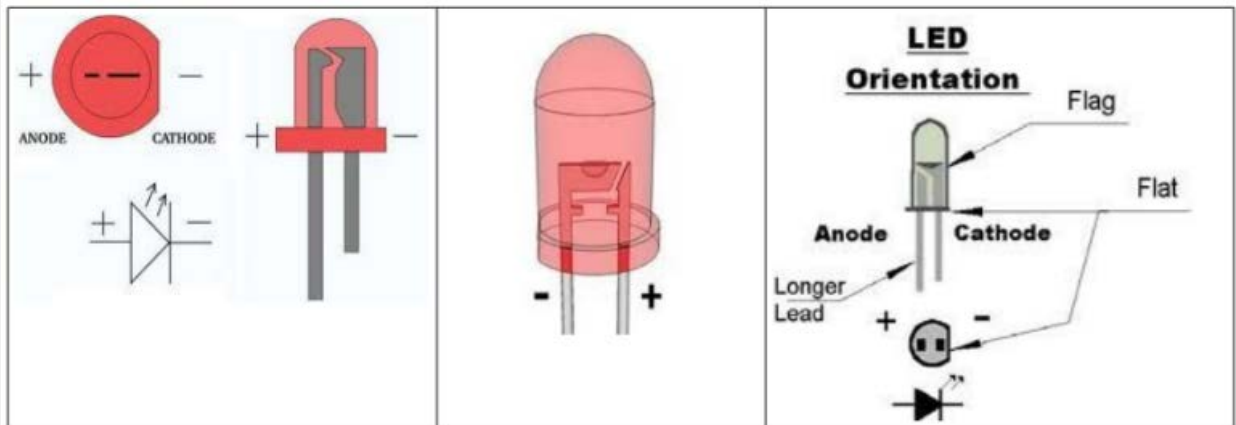
1 LED

1 αντίσταση 220Ω

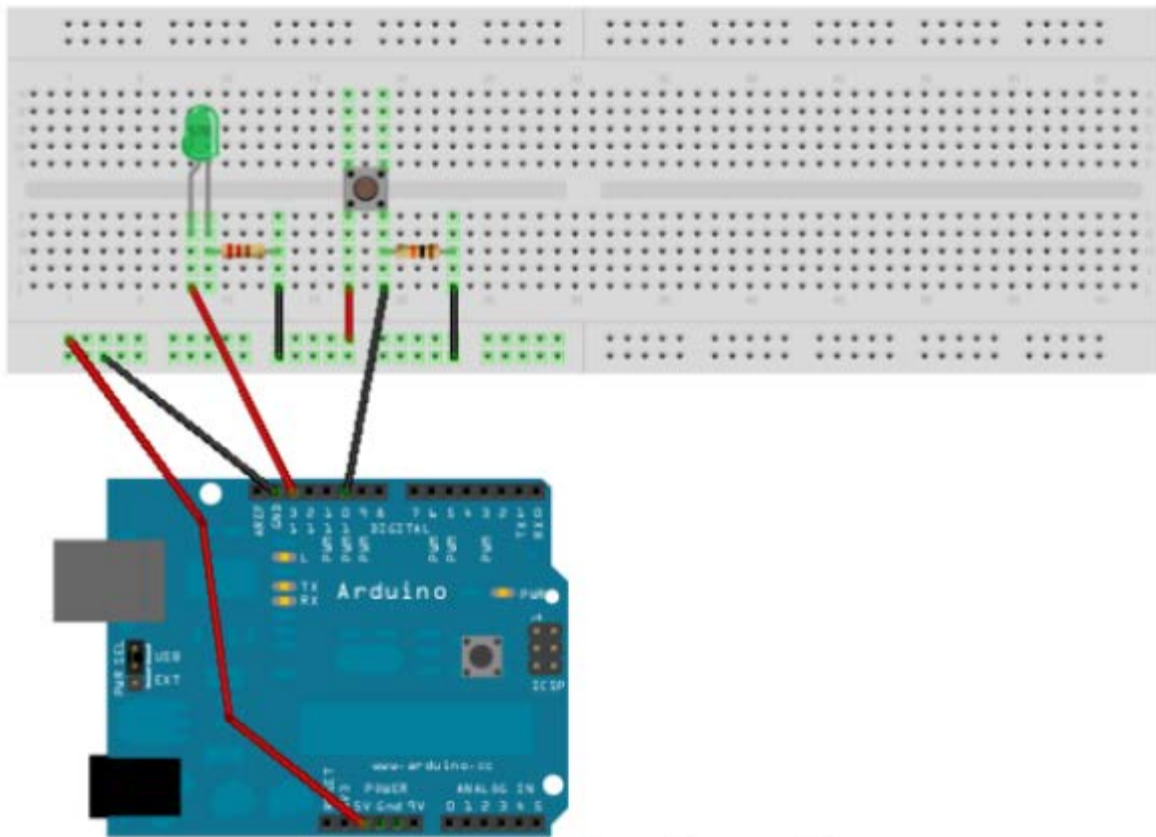
1 διακόπτη

1 αντίσταση 10kΩ (καφέ – μαύρο – πορτοκαλί).

Η συνδεσμολογία που κάναμε φαίνεται στο παρακάτω σχήμα:



Εικόνα 3-3: Πολικότητα LED



Εικόνα 3-4: Αναπαράσταση β' κυκλώματος
Το πρόγραμμα που φορτώσαμε στο Arduino είναι το παρακάτω:

```
int LEDPin = 13;  
  
int inPin = 10;  
  
void setup(){  
  
pinMode(LEDPin, OUTPUT);  
  
pinMode(inPin, INPUT);  
  
}  
  
void loop(){
```

```
if (digitalRead(inPin) == HIGH){  
  
digitalWrite(LEDPin, HIGH);  
  
delay(1000);  
  
digitalWrite(LEDPin, LOW);  
  
delay(1000);  
  
}  
  
}
```

Όπως παρατηρείτε, τις περισσότερες εντολές τις μελετήσαμε στο προηγούμενο παράδειγμα. Αυτό που μένει να μελετήσουμε είναι η συνθήκη `if (digitalRead(inPin) == HIGH){}`.

Η εντολή `digitalRead(inPin)` διαβάζει τη κατάσταση του pin 10 του Arduino. Αν πιέσουμε τον

διακόπτη το pin 10 παίρνει τη τιμή HIGH, δηλαδή κλείνει το κύκλωμα με 5V (βλ. Κύκλωμα), αν

αφήσουμε τον διακόπτη τότε το pin γειώνετε (GND) άρα έχουμε κατάσταση LOW στο pin.

Άρα όπως βλέπουμε πιο πάνω, ο κώδικας που βρίσκετε μέσα στην `if()` θα εκτελεστεί μόλις

ανιχνευτούν 5V στο ψηφιακό pin.

3.11 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:

```
int LEDPin = 13;

void setup()

{

pinMode(LEDPin, OUTPUT);

}

void loop()

{

}

digitalWrite(LEDPin, HIGH);
```

```
delay(1000);  
  
digitalWrite(LEDpin, LOW);  
  
delay(1000 );  
  
}
```

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.

3.12 Ασκήσεις

Άσκηση 1η

Χρησιμοποιώντας 6 LED να υλοποιήσετε μια εφαρμογή η οποία να τα κάνει να αναβοσβήνουν με τη σειρά κάνοντας παύση μισού δευτερολέπτου κάθε φορά. Η σειρά θα είναι η εξής:

Σειρά εκτέλεσης	Κατάσταση led
1	
2	
3	
4	
5	
6	

Εικόνα 3-5: Α' άσκηση - Σειρά αναβοσβησίματος LED
Στη συνέχεια αν θέλουμε μπορούμε να αλλάξουμε την σειρά εκτέλεσης σε:

1, 2, 3, 4, 5, 6, 5, 4, 3, 2, 1.

Να χρησιμοποιηθούν τα ψηφιακά pin 2 ως 7 για τα LED.

* Σε μελλοντικό στάδιο θα χρησιμοποιήσουμε βρόχο για να μειώσουμε το μέγεθος του κώδικα αλλά και να αυτοματοποιούμε την διαδικασία της εκτέλεσης.

Άσκηση 2

Αφού μελετήσετε την εντολή `random()` από το reference του Arduino (Help -> Reference) να υλοποιήσετε ένα ηλεκτρονικό ζάρι, χρησιμοποιώντας 6 LED και ένα διακόπτη. Κάθε φορά που θα πατάτε τον διακόπτη να παράγεται ένας τυχαίος αριθμός από το 1 ως το 6 και στη συνέχεια να ανάβει το ανάλογο LED.

Τα LED πρέπει να τοποθετηθούν το ένα δίπλα από το άλλο στη breadboard ώστε το κάθε ένα να αντιστοιχεί σε ένα αριθμό, για το πρώτο LED ο αριθμός 1, για το δεύτερο ο αριθμός 2 κ.ο.κ. Να χρησιμοποιηθούν τα ψηφιακά pin 2 ως 7 για τα LED, ενώ ο διακόπτης να τοποθετηθεί στο ψηφιακό pin 10.