

**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΟΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ 1500**

**ΜΕΛΕΤΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ  
ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ  
ΜΕΤΡΗΣΗΣ, ΑΠΕΙΚΟΝΙΣΗΣ ΚΑΙ ΕΛΕΓΧΟΥ  
ΘΕΡΜΟΚΡΑΣΙΑΣ ΚΑΙ ΦΩΤΙΣΜΟΥ**

**ΒΑΡΕΛΑΣ ΝΙΚΟΛΑΟΣ 6340  
ΜΩΡΑΓΓΗΣ ΝΙΚΟΛΑΟΣ 6508**

**ΕΠΟΠΤΕΥΟΝ ΚΑΘΗΓΗΤΗΣ: ΜΠΙΣΔΟΥΝΗΣ ΛΑΜΠΡΟΣ**

**ΠΑΤΡΑ 2016**







## ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη .....	5
Πρόλογος .....	6
Εισαγωγή .....	8
Τί είναι το Arduino .....	9
Φωτοαντίσταση – LDR .....	31
Αισθητήρας θερμοκρασίας DS18B20 .....	33
Στήσιμο κατασκευής, δοκιμές και επεξήγηση .....	36
Περιγραφή και επεξήγηση κώδικα .....	43
Στάδια υλοποίησης εργασίας .....	48
Πίνακας απαρτιών .....	52
Πηγές .....	53

## ΠΕΡΙΛΗΨΗ

Με αυτό το σύστημα ο χρήστης θα έχει τη δυνατότητα να βάλει σε λειτουργία ένα σύστημα θέρμανσης ή ψύξης (τα οποία είναι απαραίτητα για τη σωστή λειτουργία του συστήματος). Αυτό το σύστημα ανάλογα τη θερμοκρασία του χώρου και τη θερμοκρασία που θα έχει ορίσει ο χρήστης η οποία θα είναι πλήρως συμβατή από αυτόν θα ενεργοποιούνται ή θα απενεργοποιούνται τα αντίστοιχα ρελέ. Επίσης θα υπάρχει και ένας αισθητήρας φωτεινότητας ο οποίος θα είναι εξωτερικά του συστήματος και μέσω μιας επέκτασης θα μπορεί να τοποθετηθεί όπου θέλει ο χρήστης, ο αισθητήρας αυτός θα ενεργοποιεί ένα ρελέ για το άνοιγμα ή κλείσιμο του φωτισμού για κάποια όρια τα οποία θα είναι προκαθορισμένα. Επιπλέον θα υπάρχει μία οθόνη στην οποία θα απεικονίζονται η θερμοκρασία περιβάλλοντος και τα όρια της θερμοκρασίας, θέρμανσης και ψύξης που θα έχει ορίσει ο χρήστης. Επίσης μέσω αυτής της οθόνης και των αντίστοιχων κουμπιών πάνω στο σύστημα θα γίνεται και ο έλεγχος των ορίων θερμοκρασίας.

## ΠΡΟΛΟΓΟΣ

Ο αυτοματισμός ερευνά τη συμπεριφορά δυναμικών συστημάτων μοντελοποιώντας τα με τα μεθοδολογικά και μαθηματικά εργαλεία της επεξεργασίας σήματος. Έτσι μεταχειρίζεται τα συστήματα ως μαύρα κουτιά με είσοδο και έξοδο. Ως είσοδος θεωρείται ένα σήμα, αναλογικό ή ψηφιακό, συλλεγόμενο από κάποιο σημείο του συστήματος. Τα ενδιάμεσα κουτιά αναπαριστούν τις διάφορες διαταράξεις που επηρεάζουν το σήμα, όπως τριβές στους ενεργοποιητές, αποτέλεσμα των στοιχείων του ελέγχου που παρεμβάλλονται, τους ελεγκτές.

Αυτά τα αποτελέσματα συνήθως αναπαρίστανται με μαθηματικές συναρτήσεις, τις συναρτήσεις μεταφοράς. Μία συνάρτηση μεταφοράς προσδιορίζει ένα σύστημα και τον τρόπο που μεταβάλλει κάθε σήμα εισόδου. Η έξοδος του συστήματος ονομάζεται αναφορά και ανταποκρίνεται στην τιμή του σήματος κατόπιν ενεργοποίησης των προηγούμενων συναρτήσεων μεταφοράς σε αυτήν. Όταν μια ή περισσότερες μεταβλητές εξόδου ενός συστήματος πρέπει να ακολουθήσουν την τιμή κάποιας αναφοράς που μεταβάλλεται με τον χρόνο, χρειάζεται να προστεθεί ένας ελεγκτής που να χειρίζεται τις τιμές των σημάτων εισόδου έως ότου επιτευχθεί η επιθυμητή έξοδος.

Στην πτυχιακή αυτή έχουμε έναν αυτοματισμό με εισόδους δύο αισθητήρες και τέσσερα κουμπιά και τρεις εξόδους που ενεργοποιούν τρεις γραμμές στο δίκτυο ηλεκτρισμού. Η επεξεργασία σήματος και η μοντελοποίηση γίνεται μέσω μίας μικροεπεξεργαστικής κάρτας.





## ΕΙΣΑΓΩΓΗ

Ο αυτοματισμός (ή αυτοματοποίηση) αφορά δύο έννοιες μη σχετιζόμενες μεταξύ τους. Αρχικά, σημαίνει την τυποποίηση μίας διαδικασίας μέσω της εύρεσης καλώς ορισμένων βημάτων τα οποία πρέπει να ακολουθηθούν για να παραχθεί κάποιο επιθυμητό αποτέλεσμα. Έτσι ο αυτοματισμός δεν είναι τίποτα άλλο παρά η εύρεση ενός αλγορίθμου για την επίλυση ενός προβλήματος, ή η κατασκευή ενός αυτόνομου μηχανισμού που εκτελεί αυτόν τον αλγόριθμο για κάποια είσοδο χωρίς ανθρώπινη παρέμβαση.

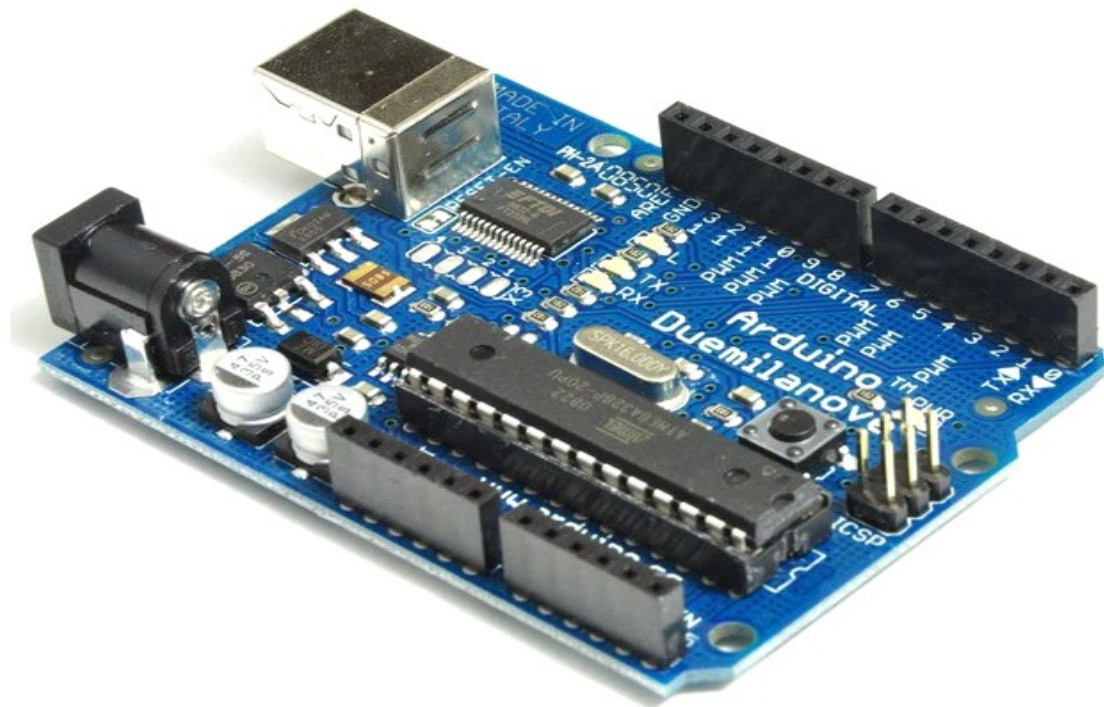
Ο αυτοματισμός στηρίζεται εννοιολογικά στη θεωρία ελέγχου και στους μηχανισμούς ανάδρασης, επομένως αποτέλεσε μία κύρια αφετηρία της κυβερνητικής. Σε αντίθεση με την τελευταία όμως, ο αυτοματισμός έχει έναν αυστηρά εφαρμοσμένο χαρακτήρα και στην πράξη αξιοποιεί ποικιλία εξειδικευμένων προϊόντων ηλεκτρονικής και τεχνολογίας πληροφοριών (π.χ. μικροελεγκτές, συστήματα πραγματικού χρόνου, τεχνολογίες CAx). Η σημασία του αυτοματισμού είναι μεγάλη στη βιομηχανία, όπου μειώνει σημαντικά την ανάγκη για ανθρώπινη παρέμβαση (π.χ. σε τηλεμετρίες, αυτόματο έλεγχο γραμμών παραγωγής κλπ).

Σε αυτή τη πτυχιακή εργασία έχει δημιουργηθεί μία συσκευή με την οποία ελέγχεται η θερμοκρασία σε ένα χώρο και στον ίδιο χώρο στόχος είναι να υπάρχει πάντα φως.

Ο προγραμματιζόμενος ελεγκτής που θα πραγματοποιήσει αυτόν τον αυτοματισμό είναι η κάρτα Arduino Uno Rev3.

## ΤΙ ΕΙΝΑΙ ΤΟ ARDUINO

Το Arduino Uno είναι ένας single-board μικροελεγκτής, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++ ). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Pure Data, SuperCollider.



## ΜΙΚΡΟΕΛΕΓΚΤΗΣ – Η ΚΑΡΔΙΑ ΤΟΥ ARDUINO UNO

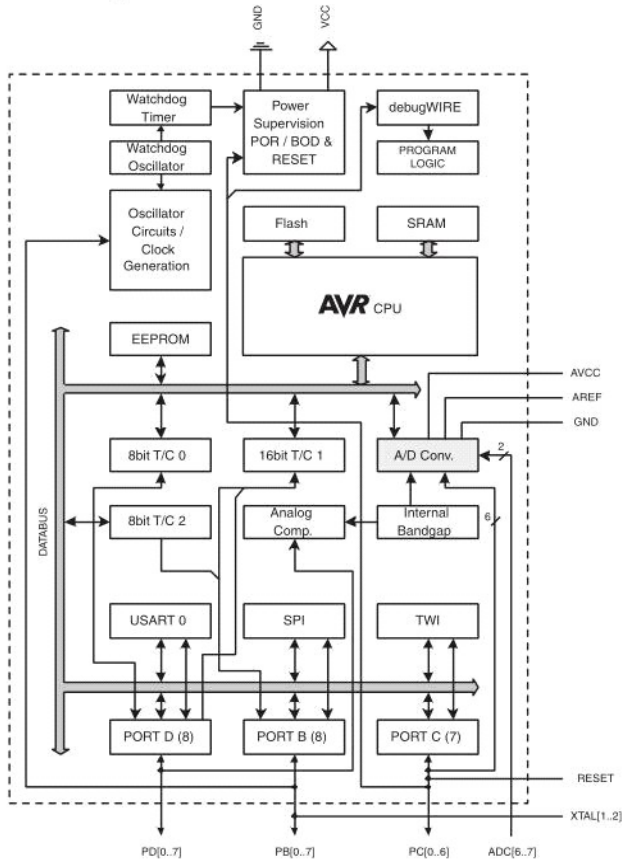
Το Arduino Uno βασίζεται στον ATmega328, έναν 8-bit RISC μικροελεγκτή, τον οποίο χρονίζει στα 16MHz. Ο ATmega328 διαθέτει ενσωματωμένη μνήμη τριών τύπων:

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.

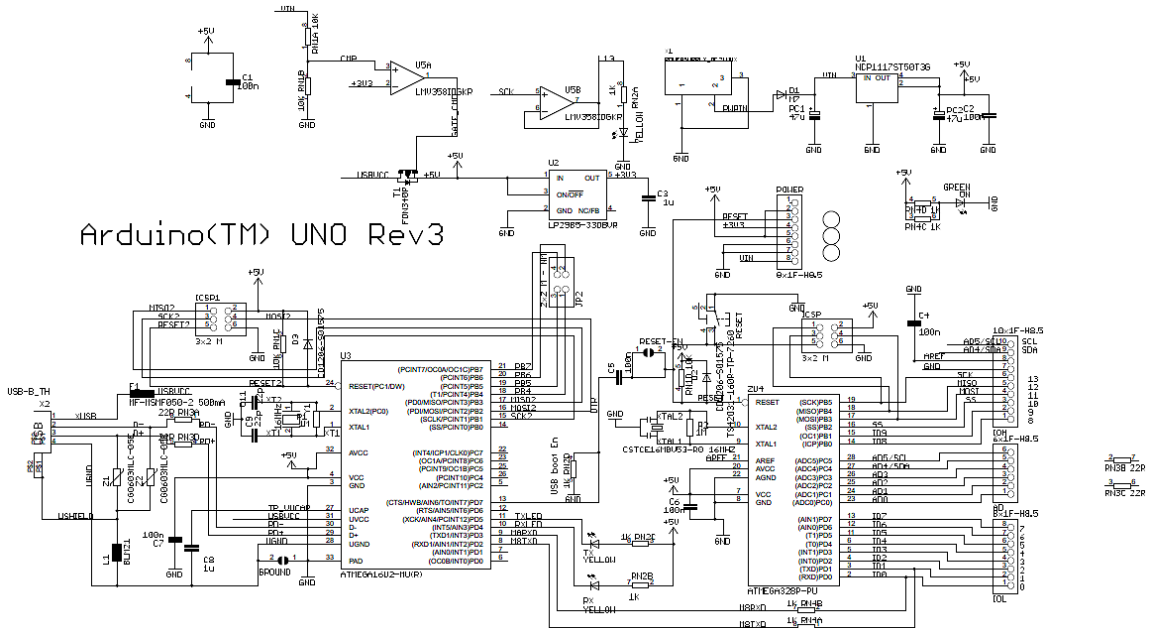
1Kb μνήμης EEPROM η οποία μπορεί να χρησιμοποιηθεί για «ωμή» εγγραφή/ανάγνωση δεδομένων (χωρίς datatype) ανά byte από τα προγράμματά σας κατά το runtime. Σε αντίθεση με την SRAM, η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset οπότε είναι το ανάλογο του σκληρού δίσκου.

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).

## ΤΟ BLOCK ΚΑΙ ΤΟ ΣΧΕΔΙΟ ΤΟΥ ARDUINO UNO



Arduino™ UNO Rev3



## ΤΑ ΣΗΜΑΝΤΙΚΟΤΕΡΑ ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΟΥ ARDUINO

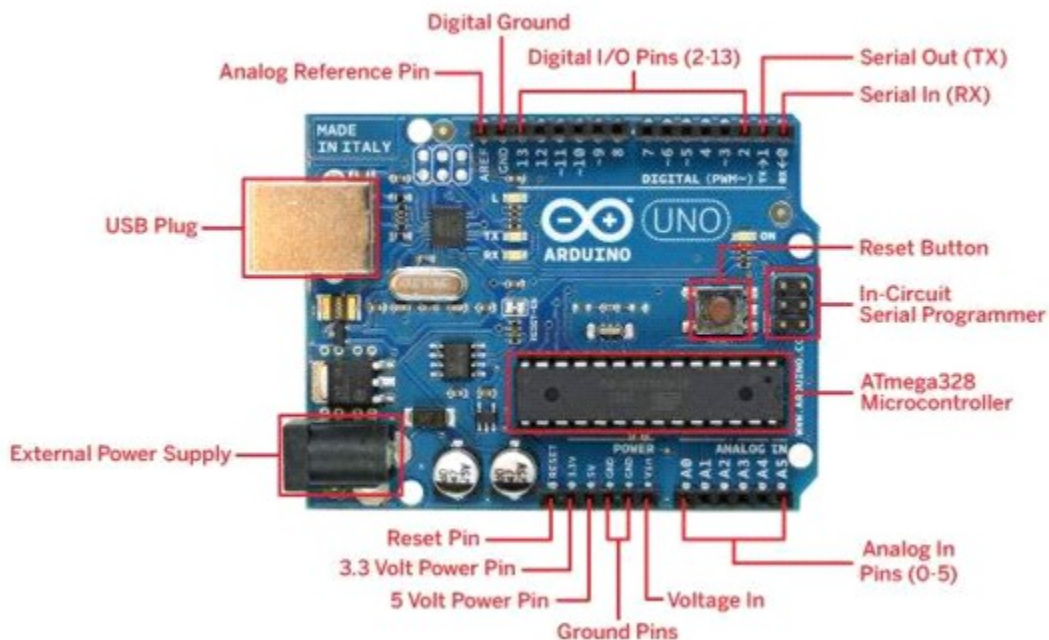
- 1) Το Χαμηλό κόστος. Το κόστος μιας πλακέτας ανέρχεται σε μερικά ευρώ. Πιο συγκεκριμένα το Arduino uno που είναι η ναυαρχίδα της Arduino μπορεί να αγοραστεί μεσώ ίντερνετ για 10-15€. Επίσης υπάρχουν πακέτα ξεκινήματος τα οποία ξεκινούν από 30 € και τα οποία εκτός από την πλακέτα προμηθεύουν μια γκάμα από ηλεκτρονικά (καλώδια, αισθητήρες, ηλεκτροκινητήρες, led ...) για τις πρώτες δόκιμες-κατασκευές.
- 2) Είναι Ανεξαρτήτου πλατφόρμας (cross-platform). Το πρόγραμμα του arduino εκτελείται και στα τρία λειτουργικά συστήματα (windows, macintosh, linux) αγκαλιάζοντας όλο το εύρος των χρηστών προσωπικών υπολογιστών
- 3) Η Απλότητα. Ίσως το πιο σημαντικό πλεονέκτημα του arduino είναι η απλότητα του. Μέσα σε λίγες ώρες ο άπειρος χρήστης μπορεί να δημιουργήσει την πρώτη του κατασκευή. Αποτελεί ιδανικό δημιουργικό εργαλείο για την απόκτηση ηλεκτρονικών και μηχανικών δεξιοτήτων καθώς επίσης και δημιουργική απασχόληση για τους εφήβους.
- 4) Οι εκδόσεις του Arduino. Η arduino είναι μια οικογένεια από πλακέτες μικροελεγχτών που σκοπό έχουν να κάνουν ευκολότερη την κατασκευή διαδραστικών αντικείμενων. Κάθε έκδοση καλύπτει διαφορετικές ανάγκες και έχει διαφορετικές δυνατότητες. Έκτος από τις διαφορετικές εκδόσεις υπάρχουν και οι πλακέτες επέκτασης (shield's) που έρχονται να ενισχύσουν και να δώσουν νέες δυνατότητες στις πλατφόρμες του arduino. Όπως γίνεται κατανοητό πάντα θα υπάρχουν διάφορες εκδόσεις του arduino που θα καλύπτουν την τρέχουσα τεχνολογία.
- 5) Η οικογένεια του Arduino. Ένα ακόμα από τα σημαντικότερα του πλεονεκτήματα είναι το πλήθος των ανθρώπων που ασχολούνται με κάθε τομέα του arduino (υλικό και λογισμικό). Έτσι υπάρχουν αμέτρητα forum και ιστοσελίδες που μπορούν να καθοδηγήσουν, βοηθήσουν, διδάξουν και εμπνεύσουν των κάθε χρήστη, έμπειρο ή όχι.
- 6) Το Ανοικτού κώδικα και επέκτασης λογισμικό. Ο καθένας μπορεί να βρει τον πηγαίο κωδικά, να τον μελετήσει και να τον τροποποιήσει σύμφωνα με τις ανάγκες του. Έμπειροι χρήστες μπορούν μέσα από τις βιβλιοθήκες τις C++ άλλα και μέσα από τις βιβλιοθήκες του arduino να γράψουν τον δικό τους κωδικά

καινα τον μοιραστούν. Ενώ αρχάριοι χρήστες μπορούν να χρησιμοποιήσουν αυτόν το κωδικά χωρίς να γνωρίζουν προγραμματισμό χαμηλού επίπεδου.

- 7) Το Ανοικτού κώδικα και επέκτασης υλικό. Τα σχέδια των πλατφορμών είναι ανοικτού κωδικά πράγμα που σημαίνει ότι έμπειροι χρήστες στην ηλεκτρονική μπορούν να επεκτείνουν και να αναβαθμίσουν τις πλατφόρμες. Ακόμα και άπειροι χρήστες μπορούν να κατασκευάσουν την breadboard εκδοσή εάν θέλουν να καταλάβουν πως λειτουργεί και να γλιτώσουν και λίγα χρήματα.

## ΕΙΣΟΔΟΙ – ΕΞΟΔΟΙ

Το Arduino Uno 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 κατάστασή 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 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 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 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 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

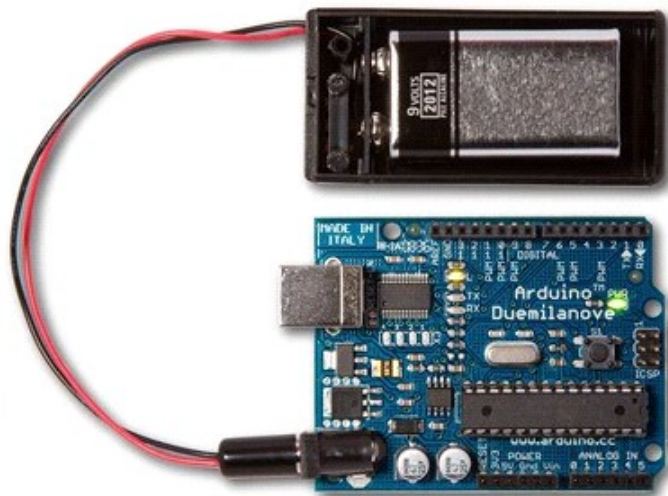
ενσωματωμένο στον μικροελεγκτή. Για παράδειγμα, μπορείτε να τροφοδοτήσετε ένα από αυτά με μια τάση την οποία μπορείτε να κυμάνετε με ένα ποτενσιόμετρο από 0V ως

για μια τάση αναφοράς  $V_{ref}$  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.

## ΤΡΟΦΟΔΟΣΙΑ

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 εφαρμογές μακριά από τον υπολογιστή θα χρειαστείτε ένα τροφοδοτικό -σαν αυτό της εικόνας- για να τροφοδοτήσετε το Arduino με ρεύμα. Μπορείτε να χρησιμοποιήσετε ένα παλιό φορτιστή ή ένα τροφοδοτικό που σας έχει περισσέψει από κάποια άλλη συσκευή, αρκεί να παρέχει από 7 ως 12 Volt. Αν πάλι θέλετε μια εντελώς αυτόνομη εφαρμογή που δεν εξαρτάται ούτε από πρίζα, μια κοινή μπαταρία των 9 Volt συνδεδεμένη όπως στην εικόνα είναι η ιδανική τροφοδοσία.

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:

- 1 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.
- 2 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.
- 3 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.
- 4 Το τέταρτο και το πέμπτο pin, με την ένδειξη GND, είναι φυσικά γειώσεις.
- 5 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 φιν 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.

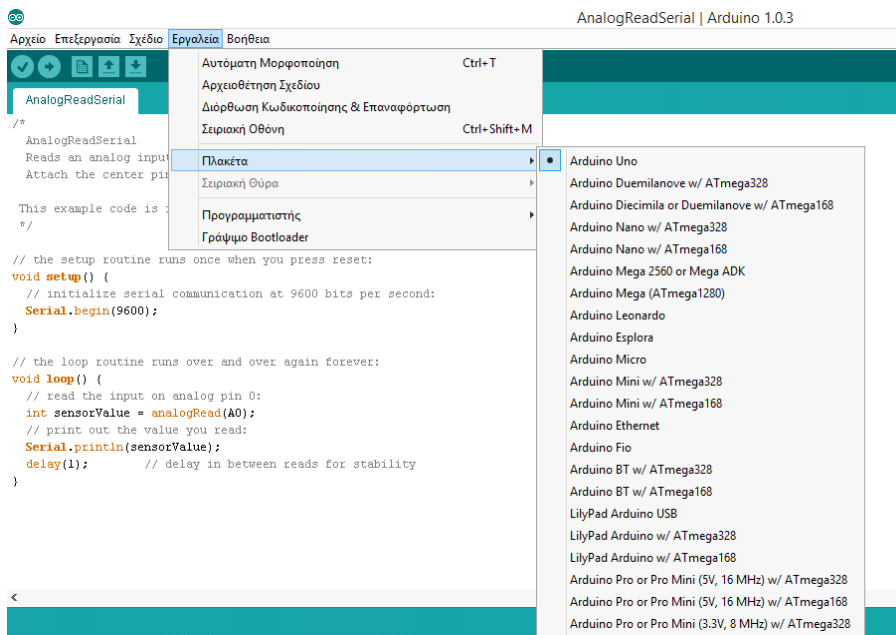
## ΕΝΣΩΜΑΤΩΜΕΝΑ ΚΟΥΜΠΙΑ ΚΑΙ LED

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 ΣΥΝΔΕΣΗ ΜΕ ΤΟΝ ΥΠΟΛΟΓΙΣΤΗ

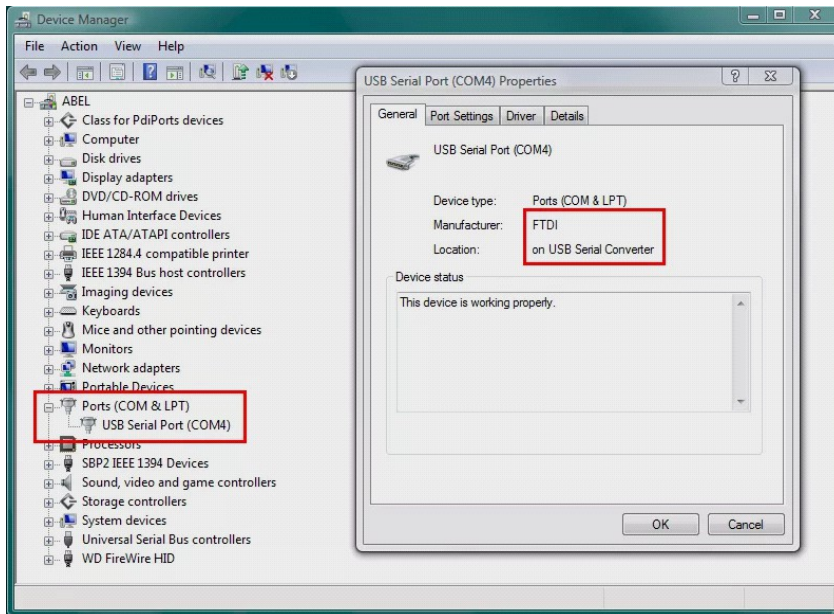
Ότι χρειάζεστε για την διαχείριση του Arduino από τον υπολογιστή σας το παρέχει το Arduino IDE, την τελευταία έκδοση του οποίου μπορείτε να κατεβάσετε από το επίσημο site για καθένα από τα τρία δημοφιλέστερα λειτουργικά συστήματα.



Το Arduino IDE είναι βασισμένο σε Java και συγκεκριμένα παρέχει:

- 1 Ένα πρακτικό περιβάλλον για την συγγραφή των προγραμμάτων σας (τα οποία ονομάζονται sketch στην ορολογία του Arduino) με συντακτική χρωματική σήμανση.
- 2 Αρκετά έτοιμα παραδείγματα.
- 3 Μερικές έτοιμες βιβλιοθήκες για προέκταση της γλώσσας και για να χειρίζεστε εύκολα μέσα από τον κώδικά σας τα εξαρτήματα που συνδέετε στο Arduino.
- 4 Τον compiler για την μεταγλώττιση των sketch σας.
- 5 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.
- 6 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.



Αν ο οδηγός του ελεγκτή Serial-over-USB είναι σωστά εγκατεστημένος στα Windows, το Arduino θα αναγνωρίζεται από τον Device Manager όπως στην εικόνα. Εκεί μπορείτε να δείτε και τον αριθμό της εικονικής σειριακής θύρας που του ανατέθηκε. 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 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.

## ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

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 παρακατω:

### Δομές ελέγχου ροής

- 1 if (δομή ελέγχου μίας συνθήκης)
- 2 if ... else (δομή ελέγχου πολλαπλών συνθηκών)
- 3 or (δομή επαναληπτικού ελέγχου συνθήκης)
- 4 while (δομή επαναληπτικού ελέγχου συνθήκης)
- 5 do ... while (δομή επαναληπτικού ελέγχου συνθήκης)
- 6 switch ... case (δομή ελέγχου περιπτώσεων)
- 7 break (εντολή διακοπής μιας επαναληπτικής δομής)
- 8 continue (εντολή παράλειψης της τρέχουσας επανάληψης)
- 9 return (εντολή επιστροφής από μία συνάρτηση)
- 10 goto (εντολή μετάβασης σε κάποιο σημείο του κώδικα)

## Αριθμητικοί τελεστές

- 1 = (τελεστής εκχώρησης)
- 2 + (τελεστής πρόσθεσης)
- 3 - (τελεστής αφαίρεσης)
- 4 \* (τελεστής πολλαπλασιασμού)
- 5 / (τελεστής διαίρεσης)
- 6 % (τελεστής υπόλοιπου ακεραίας διαίρεσης)

## Λογικοί τελεστές

- 1 && (λογική σύζευξη)
- 2 || (λογική διάζευξη)
- 3 ! (λογική άρνηση)

## Δυαδικοί τελεστές

- 1 & (δυαδική σύζευξη)
- 2 | (δυαδική διάζευξη)
- 3 ^ (δυαδική αποκλειστική διάζευξη)
- 4 ~ (δυαδική άρνηση)
- 5 << (δυαδική αριστερή ολίσθηση)
- 6 >> (δυαδική δεξιά ολίσθηση)

## Τελεστές αύξησης και μείωσης

- 1 ++ (αύξηση κατά μία ακέραιη μονάδα)
- 2 -- (μείωση κατά μία ακέραιη μονάδα)

## Σύνθετοι τελεστές

- 1 +=, -=, \*=, /=, %= (σύνθετοι αριθμητικοί τελεστές)
- 2 &=, |=, ^=, ~=, <<=, >>= (σύνθετοι δυαδικοί τελεστές)

## Τελεστές σύγκρισης

- 1 == (ισότητα)
- 2 != (ανισότητα)
- 3 < (μικρότερο)
- 4 > (μεγαλύτερο)
- 5 <= (μικρότερο ή ίσο)
- 6 >= (μεγαλύτερο ή ίσο)

## Τελεστές δεικτών

- 1 \* (τελεστής απόκτησης περιεχομένου)
- 2 & (τελεστής απόκτησης διεύθυνσης)

## Σταθερές

- 1 HIGH (τιμή υψηλής στάθμης για μία επαφή εισόδου ή εξόδου)
- 2 LOW (τιμή χαμηλής στάθμης για μία επαφή εισόδου ή εξόδου)

- 3 false (λογικό επίπεδο ψεύδους σε μία συνθήκη)
- 4 true (λογικό επίπεδο αλήθειας σε μία συνθήκη)
- 5 INPUT (χρησιμοποιείται για τον ορισμό μίας επαφής ως είσοδο)
- 6 OUTPUT (χρησιμοποιείται για τον ορισμό μίας επαφής ως έξοδο)
- 7 A0, ..., A5 (συμβολοσταθερές για τις αναλογικές επαφές εισόδου)

### Τύποι δεδομένων

- 1 boolean (λογική δυαδική τιμή)
- 2 char (προσημασμένος χαρακτήρας 8 ψηφίων)
- 3 unsigned char (μη προσημασμένος χαρακτήρας 8 ψηφίων)
- 4 byte (μη προσημασμένος χαρακτήρας 8 ψηφίων)
- 5 int (προσημασμένος ακέραιος αριθμός 16 ψηφίων)
- 6 unsigned int (μη προσημασμένος ακέραιος αριθμός 16 ψηφίων)
- 7 word (μη προσημασμένος ακέραιος αριθμός 16 ψηφίων)
- 8 long (προσημασμένος ακέραιος αριθμός 32 ψηφίων)
- 9 unsigned long (μη προσημασμένος ακέραιος αριθμός 32 ψηφίων)
- 10 float, double (αριθμός κινητής υποδιαστολής απλής ακρίβειας)
- 11 String (αντικείμενο αλφαριθμητικού με χρήσιμες μεθόδους)
- 12 Ως αλφαριθμητικό μπορεί να θεωρηθεί και ο πίνακας χαρακτήρων



## Συναρτήσεις μετατροπής τύπων

- 1 char(), byte()
- 2 int(), word(), long()
- 3 float(), double()

## Συναρτήσεις εισόδου και εξόδου

- 1 pinMode() (ορίζει μια επαφή ως είσοδο ή έξοδο)

## Συναρτήσεις ψηφιακής εισόδου και εξόδου

- 1 digitalWrite() (γράφει σε μία ψηφιακή επαφή εξόδου)
- 2 digitalRead() (διαβάζει από μία ψηφιακή επαφή εισόδου)

## Συναρτήσεις αναλογικής εισόδου και εξόδου

- 1 analogReference() (ορίζει την τάση αναλογικής αναφοράς)
- 2 analogWrite() (γράφει PWM σήματα σε μία επαφή εξόδου)
- 3 analogRead() (διαβάζει από μία αναλογική επαφή εισόδου)

## Προηγμένες συναρτήσεις εισόδου και εξόδου

- 1 tone() (παράγει ένα τετραγωνικό σήμα ορισμένης συχνότητας)
- 2 noTone() (διακόπτει την παραγωγή τετραγωνικών σημάτων)
- 3 shiftOut() (ολισθαίνει τα ψηφία μιας τιμής σε μία επαφή εξόδου)
- 4 pulseIn() (επιστρέφει την διάρκεια σε μs ενός παλμού HIGH ή LOW)

## Συναρτήσεις χρόνου

- 1 `millis()` (διάρκεια εκτέλεσης του προγράμματος σε ms)
- 2 `micros()` (διάρκεια εκτέλεσης του προγράμματος σε μs)
- 3 `delay()` (παύση προγράμματος - η διάρκεια δίδεται σε ms)
- 4 `delayMicroseconds()` (παύση προγράμματος - η διάρκεια δίδεται σε μs)

## Μαθηματικές και Τριγωνομετρικές συναρτήσεις

- 1 `max()` (βρίσκει τον μεγαλύτερο ανάμεσα σε δύο αριθμούς)
- 2 `min()` (βρίσκει τον μικρότερο ανάμεσα σε δύο αριθμούς)
- 3 `abs()` (επιστρέφει την απόλυτη τιμή ενός αριθμού)
- 4 `constrain()` (ελέγχει για υπερχείλιση ή υποχείλιση ορίων)
- 5 `map()` (πραγματοποιεί γραμμικό μετασχηματισμό ορίων)
- 6 `pow()` (επιστρέφει το αποτέλεσμα μίας δύναμης)
- 7 `sqrt()` (επιστρέφει την ρίζα ενός αριθμού)
- 8 `sin()` (υπολογίζει το ημίτονο ενός αριθμού)
- 9 `cos()` (υπολογίζει το συνημίτονο ενός αριθμού)
- 10 `tan()` (υπολογίζει την εφαπτομένη ενός αριθμού)

## Συναρτήσεις γεννήτριας ψευδοτυχαίων αριθμών

- 1 `random()` (δίδεται ένας νέος αριθμός από την γεννήτρια)
- 2 `randomSeed()` (θέτει τον σπόρο της γεννήτριας παραγωγής)

### Συναρτήσεις επεξεργασίας δυαδικών αριθμών

- 1 lowByte() (επιστρέφει το δεξιότερο byte μίας μεταβλητής)
- 2 highByte() (επιστρέφει το αριστερότερο byte μίας μεταβλητής)
- 3 bitRead() (διαβάζει ένα συγκεκριμένο ψηφίο μίας μεταβλητής)
- 4 bitWrite() (γράφει σε ένα συγκεκριμένο ψηφίο μιας μεταβλητής)
- 5 bitSet() (γράφει την τιμή 1 σε κάποιο ψηφίο μίας μεταβλητής)
- 6 bitClear() (γράφει την τιμή 0 σε κάποιο ψηφίο μιας μεταβλητής)
- 7 bit() (υπολογίζει μία συγκεκριμένη δύναμη με βάση το 2)

### Συναρτήσεις χρήσης ρουτινών εξυπηρέτησης διακοπών

- 1 attachInterrupt() (ενεργοποιεί μία ρουτίνα εξυπηρέτησης διακοπής)
- 2 detachInterrupt() (απενεργοποιεί μία ρουτίνα εξυπηρέτησης διακοπής)

### Συναρτήσεις ενεργοποίησης και απενεργοποίησης διακοπών

- 1 interrupts() (ενεργοποιεί τα σήματα διακοπής)
- 2 noInterrupts() (απενεργοποιεί τα σήματα διακοπής)

### Υποστήριξη σειριακής επικοινωνίας

- 1 Serial (αντικείμενο σειριακής επικοινωνίας με χρήσιμες μεθόδους)

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

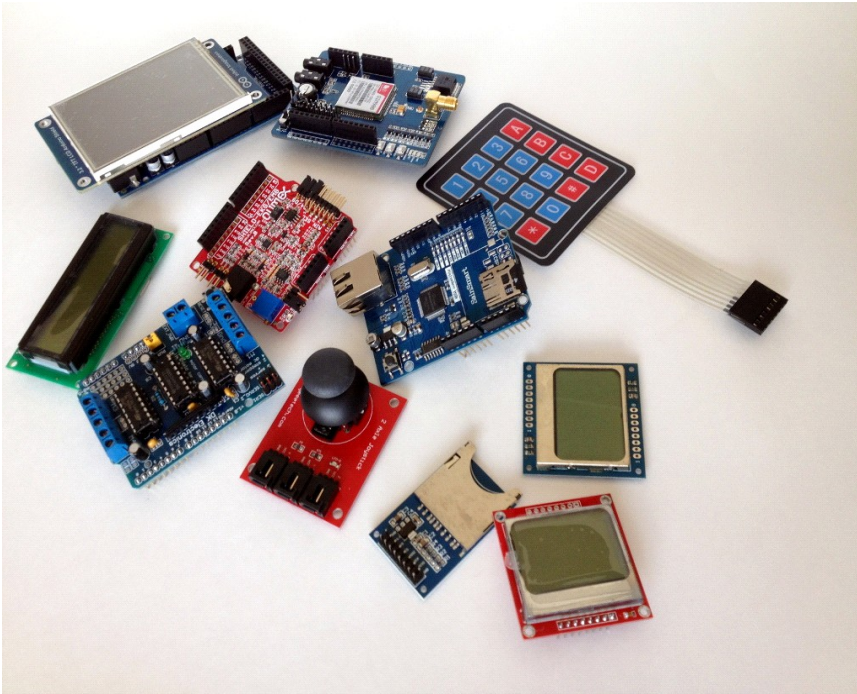
```
// Ενσωματώσεις βιβλιοθηκών, δηλώσεις μεταβλητών...
void setup()
{
  // ...
}
void loop()
{
  // ...
}
// Υπόλοιπες συναρτήσεις...
```

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 μπορέσετε να δημιουργήσετε το sketch ακόμα και για κάποιο αρκετά περίπλοκο project.

## SHEILDS

Τα sheilds είναι ολοκληρωμένες πλακέτες που είναι σχεδιασμένες ώστε να κουμπώνουν πάνω στο Arduino προεκτείνοντας την λειτουργικότητά του. Είναι η hardware αντίστοιχη έννοια των plugi, addon και extension που υπάρχουν στο software.



Μερικά από τα πιο δημοφιλή shield που κυκλοφορούν στο εμπόριο για Arduino είναι:

- 1 Ethernet shield: Δίνει στο Arduino την δυνατότητα να δικτυωθεί σε ένα LAN ή στο internet μέσω ενός τυπικού καλωδίου Ethernet.
- 2 WiFi shield: Όμοιο με το Ethernet shield, χωρίς φυσικά το καλώδιο.
- 3 Διάφορα shield οθόνης: Προσθέτουν οθόνη στο Arduino. Κυκλοφορούν από απλές οθόνες τύπου calculator μέχρι OLED touchscreen υψηλής ανάλυσης τύπου iPhone.
- 4 Wave shield: Δίνει στο Arduino την δυνατότητα να παίζει ήχους/μουσική από κάρτες SD.
- 5 GPS shield: Προσθέτει GPS δυνατότητες στο Arduino (εντοπισμό στίγματος).

- 6 Διάφορα Motor Shields: Σας επιτρέπουν να οδηγήσετε εύκολα μωτέρ διάφορων τύπων (απλά DC, servo, stepper κ.λπ.) από το Arduino.
- 7 ProtoShield: Μια προσχεδιασμένη πλακέτα πρωτοτυποποίησης, συμβατή στις διαστάσεις του Arduino και χωρίς εξαρτήματα για να φτιάξετε το δικό σας shield.

## ΆΛΛΕΣ ΚΑΡΤΕΣ ΤΗΣ ΟΙΚΟΓΕΝΕΙΑΣ ARDUINO

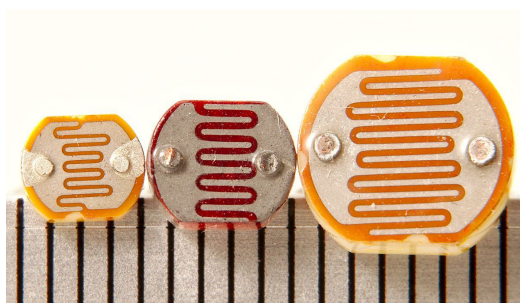
Η Arduino εκτός από την κύρια και πιο διαδεδομένη κάρτα της, η οποία είναι η Uno, έβγαλε και άλλες. Η κάθε μία έχει κάτι διαφορετικό να προσφέρει όπως τάση τροφοδοσίας digital I/Os, ταχύτητα ρολογιού, αναλογικές εισόδους κ.λ.π.

Παρακάτω φαίνονται σε πίνακα οι πιο διαδεδομένες από τις κάρτες αυτές και τα χαρακτηριστικά τους.

Πλακέτα	Μικρ/στής	Τροφοδοσία	Digital I/Os	Analog In.	Flash Mem.	Clock
Uno	ATmega328	7-12V	14 (6 PWM)	6	32 KB	16 MHz
Mega	ATmega2560	7-12V	54 (15 PWM)	16	256 KB	16 MHz
Mini	ATmega328	7-9 V	14 (6 PWM)	8	32 KB	16 MHz
Micro	ATmega32u4	7-12V	20 (7 PWM)	12	32 KB	16 MHz
DUE	AT91SAM3X8E	7-12V	54 (12 PWM)	12	512 KB	84 MHz

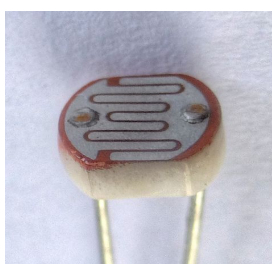
## ΦΩΤΟΑΝΤΙΣΤΑΣΗ – LDR

Μια φωτοαντίσταση (ή LDR, ή φωτοκύτταρο) είναι μια ελαφριά ελεγχόμενη μεταβλητή αντίσταση. Η αντίσταση μιάς φωτοαντίστασης μειώνεται με την αύξηση της έντασης του προσπίπτοντος φωτός. Μια φωτοαντίσταση μπορεί να εφαρμοστεί σε φωτοευαίσθητα κυκλώματα ανίχνευσης, και κυκλώματα μεταγωγής φως – σκοτάδι.



Μια φωτοαντίσταση είναι κατασκευασμένη από έναν ημιαγωγό υψηλής αντοχής. Στο σκοτάδι μπορεί να έχει μια αντίσταση τόσο υψηλή όσο αρκετά Megohms ( $M\Omega$ ), ενώ στο φως μπορεί να έχει μια αντίσταση τόσο χαμηλή όσο μερικές εκατοντάδες ohms. Αν η ένταση του προσπίπτοντος φωτός σε μια φωτοαντίσταση υπερβαίνει μια ορισμένη συχνότητα, τα φωτόνια που απορροφούνται από τον ημιαγωγό δίνουν στα δεσμευμένα ηλεκτρόνια αρκετή ενέργεια για να μεταβούν στη ζώνη αγωγιμότητας. Τα προκύπτοντα ελεύθερα ηλεκτρόνια άγουν το ηλεκτρικό ρεύμα, μειώνοντας έτσι την αντίσταση. Το εύρος αντίστασης και ευαισθησίας σε μία φωτοαντίσταση μπορεί να διαφέρει ουσιαδώς μεταξύ ανόμοιων αισθητηριών.

Επιπλέον, μοναδικές φωτοαντιστάσεις μπορεί να αντιδρούν διαφορετικά σε φωτόνια εντός ορισμένων ζωνών μήκους κύματος.



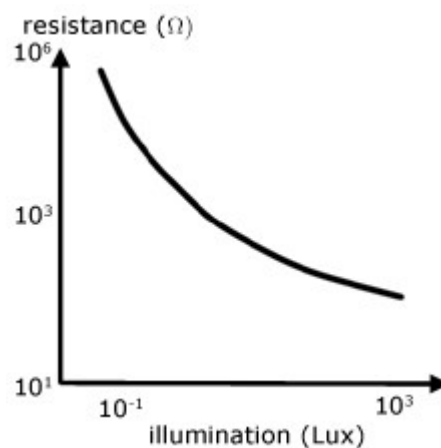
Μία φωτοηλεκτρική διάταξη μπορεί να είναι είτε ενδογενούς ή εξωγενούς. Ένας ενδογενής ημιαγωγός έχει τον δικό φορέα φορτίου της και δεν είναι ένας αποτελεσματικός ημιαγωγός. Για παράδειγμα, το πυρίτιο. Στην ενδογενή συσκευή τα μόνα διαθέσιμα ηλεκτρόνια είναι στη ζώνη σθένους, και ως εκ τούτου, το φωτόνιο πρέπει να έχουν αρκετή ενέργεια για να διεγείρουν τα ηλεκτρόνια σε ολόκληρο το διάκενο ζώνης. Οι εξωγενείς συσκευές έχουν ακαθαρσίες, που ονομάζεται επίσης προσμίξεις, των οποίων η ενέργεια θεμελιώδης κατάστασης είναι πιο κοντά στην ζώνη αγωγιμότητα. Δεδομένου ότι τα ηλεκτρόνια δεν έχουν τόσο μακριά για να πηδήσουν,



φωτόνια χαμηλότερης ενέργειας (δηλαδή, μεγαλύτερα μήκη κύματος και χαμηλότερες συχνότητες) είναι επαρκείς για να ενεργοποιήσουν τη συσκευή. Εάν ένα δείγμα του πυριτίου είχε μερικά άτομα φωσφόρου (ακαθαρσίες), θα υπήρχαν επιπλέον ηλεκτρόνια διαθέσιμα για αγωγή. Αυτό είναι ένα παράδειγμα ενός εξωγενούς ημιαγωγού.

## ΣΚΕΠΤΙΚΟ ΤΟΥ ΣΧΕΔΙΑΣΜΟΥ

Οι φωτοαντιστάσεις είναι λιγότερο ευαίσθητες στο φως από τις φωτοδιόδους οι τα φωτοτρανζίστορ: τα δύο τελευταία συστατικά είναι μόνο ημιαγωγοί ενώ μια φωτοαντίσταση είναι ένα παθητικό στοιχείο και δεν έχει PN διαστύρωση. Η αντίσταση κάθε φωτοαντίστασης μπορεί να ποικίλει ευρέως ανάλογα με την θερμοκρασία του περιβάλλοντος, καθιστώντας την ακατάλληλη για εφαρμογές που απαιτούν ακριβή μέτρηση φωτός ή ευαισθησία φωτός.



Επίσης οι φωτοαντιστάσεις παρουσιάζουν έναν ορισμένο βαθμό λανθάνουσας κατάστασης μεταξύ της έκθεσης στο φως και την επακόλουθη μείωση στην αντίσταση, συνήθως περίπου 10 χιλιοστά του δευτερολέπτου. Η χρονική καθυστέρηση κατά την μεταβίβαση από φωτεινά σε σκοτεινά περιβάλλοντα είναι ακόμη μεγαλύτερη, συχνά όσο ένα δευτερόλεπτο. Αυτή η ιδιότητα τα καθιστά ακατάλληλα για την ανίχνευση στα γρήγορα φώτα που αναβοσβήνουν, αλλά μερικές φορές χρησιμοποιούνται για να εξομαλύνουν την απόκριση της συμπίεσης του ηχητικού σήματος.

## ΕΦΑΡΜΟΓΕΣ

Οι φωτοαντιστάσεις έρχονται σε πολλούς τύπους. Ακριβά κύτταρα θείουχου κάδμιου μπορούν να βρεθούν σε πολλά στοιχεία καταναλωτών, όπως μετρητές φωτός κάμερας, ραδιόφωνα με ρολόι, συστήματα συναγερμού (ως ανιχνευτής σε δέσμη φωτός), φώτα νύχτας, εξωτερικά ρολόγια, ηλιακές λάμπες δρόμου, κ.λ.π.

Φωτοαντιστάσεις μπορούν να βρεθούν σε φώτα δρόμου για τον έλεγχο, όταν το φως είναι αναμμένο. Το φως του περιβάλλοντος που εκτίθεται στην φωτοαντίσταση προκαλεί τα φώτα του

δρόμου να απενεργοποιηθούν. Έτσι εξοικονομείται ενέργεια εξασφαλίζοντας ότι τα φώτα είναι αναμμένα μόνο κατά τις ώρες του σκότους.

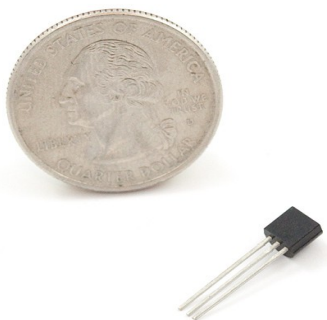
Η χρήση των CdS και CdSe φωτοαντιστάσεων είναι αυστηρά περιοσμένη στην Ευρώπη εξαιτίας της απαγόρευσης του RoHS για το κάδμιο.

Οι φωτοαντιστάσεις από μόλυβδο σουλφίδιο (PbS) και ίνδιο αντιμονιούχο (InSb) χρησιμοποιούνται για την φασματική περιοχή των υπερύθρων.

Οι φωτοαγωγοί από γερμάνιο χαλκό είναι από τους καλύτερα διαθέσιμους μακρινούς ενιχνευτές υπερύθρων και χρησιμοποιούνται για την υπέρυθη αστρονομία και υπέρυθη φασματοσκοπία.

## ΑΙΣΘΗΤΗΡΑΣ ΘΕΡΜΟΚΡΑΣΙΑΣ DS18B20

Ψηφιακός αισθητήρας θερμοκρασίας DS18B20 1-Wire από την Maxim-IC. Υπολογίζει βαθμούς Κελσίου (°C) από -55°C έως 125°C (+/-0.5°C) με ακρίβεια 9 - 12bit. Κάθε αισθητήρας χαρακτηρίζεται από μία διεύθυνση 16-Bit. Λόγω των διευθύνσεων που έχουν οι αισθητήρες αυτοί μπορούν να συνδεθούν πολλοί μαζί σε ένα μόνο καλώδιο (1-wire protocol).



Χαρακτηριστικά του αισθητήρα:

1. Μοναδική διεπαφή 1-Wire απαιτεί μόνο μία θύρα για την επικοινωνία.
2. Κάθε συσκευή έχει ένα μοναδικό 64-bit σειριακό κωδικό αποθηκευμένο σε μια ενσωματωμένη ROM.
3. Δεν απαιτεί εξωτερικά εξαρτήματα.
4. Μπορεί να τροφοδοτείται από τη γραμμή δεδομένων. Το φάσμα παροχής ρεύματος είναι 3.0V σε 5.5V.
5. Μετρά θερμοκρασίες από -55°C έως + 125°C (-67°F έως + 257°F).
6. ± 0.5°C ακρίβεια από -10°C έως + 85°C.
7. Αποτελέσματα του θερμομέτρου επιλέγονται από τον χρήστη στα 9-12 bits.
8. Μετατρέπει την θερμοκρασία σε 12-bit ψηφιακή λέξη σε 750 ms (max.).
9. Εντολή αναζήτησης συναγερμού εντοπίζει και αντιμετωπίζει τις συσκευές των οποίων η θερμοκρασία είναι εκτός των προγραμματισμένων ορίων (κατάσταση συναγερμού θερμοκρασίας).
10. Οι εφαρμογές περιλαμβάνουν θερμοστατικούς ελέγχους, βιομηχανικά συστήματα, καταναλωτικά προϊόντα, θερμομέτρα, ή οποιοδήποτε θερμικά ευαίσθητα σύστημα.

## ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΤΡΗΣΕΩΝ ΘΕΡΜΟΚΡΑΣΙΑΣ

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)	DIGITAL OUTPUT (HEX)
+125	0000 0111 1101 0000	07D0h
+85*	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

\*The power-on reset value of the temperature register is +85°C.

## BLOCK ΔΙΑΓΡΑΜΜΑ DS18B20

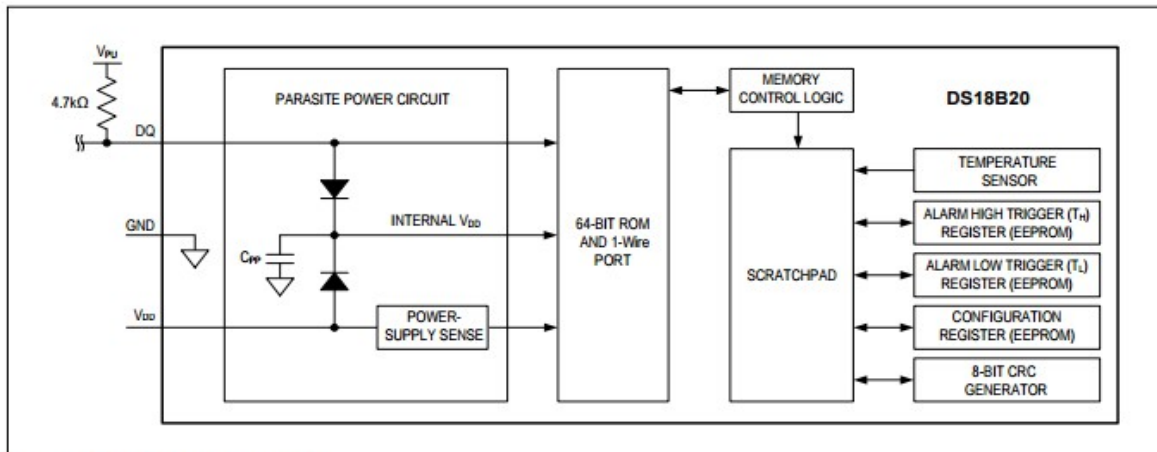


Figure 3. DS18B20 Block Diagram

## ΑΛΛΟΙ ΑΙΣΘΗΤΗΡΕΣ ΠΟΥ ΣΥΝΟΔΕΥΟΥΝ ΤΗΝ ΚΑΡΤΑ ARDUINO

- .1 Αισθητήρας Υπερήχων
- .2 Αισθητήρας Υγρασίας
- .3 Αισθητήρας Καιρού
- .4 Αισθητήρας Επιτάχυνσης

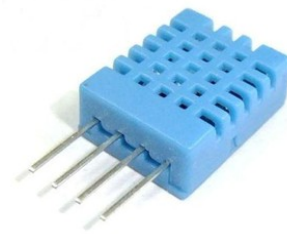
.5 Αισθητήρας Ήχου

.6 Αισθητήρας Πίεσης

.7 Αισθητήρας Αερίων



Εικόνα 1



Εικόνα 2



Εικόνα 3



Εικόνα 4



Εικόνα 5



Εικόνα 6



Εικόνα 7

## ΣΤΗΣΙΜΟ ΚΑΤΑΣΚΕΥΗΣ, ΔΟΚΙΜΕΣ ΚΑΙ ΕΠΕΞΗΓΗΣΗ

Ξεκινώντας την εργασία είναι απαραίτητος ο διαχωρισμός του συνολικού κυκλώματος σε επιμέρους μικρότερα έτσι ώστε να επιτευχθούν δύο στόχοι. Αρχικά να είναι ευκολότερες οι δοκιμές και να μην γίνει μεγάλη συνοχή καλωδίων και εξαρτημάτων σε μια δοκιμαστικά πλακάτα, η οποία συνήθως είναι περιορισμένων υποδοχών όπως στην περίπτωση μελέτης αυτής της πτυχιακής. Ο δεύτερος λόγος είναι η καλύτερη μελέτη των επιμέρους λειτουργιών των κυκλωμάτων. Αυτός ο διαχωρισμός έγινε με βάση το πάσο μάα επιμέρους λειτουργία του συστήματος εξαρτάται από την άλλη.

Στην συγκεκριμένη εργασία υπάρχουν τέσσερα βασικά κυκλώματα τα οποία δεν εξαρτώνται μεταξύ τους. Κάθε κύκλωμα έχει ένα μοναδικό εργαλείο το οποίο είναι κοινό και απαραίτητο για κάθε δοκιμή. Αυτό το εργαλείο είναι η πλακέτα Arduino Uno Rev3 και σε κάθε δοκιμή περιέχει τον απαραίτητο κώδικα για να λειτουργήσει το κάθε επιμέρους κύκλωμα.

Επιμέρους κυκλώματα:

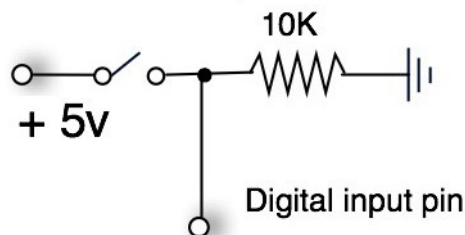
1. Λειτουργία τεσσάρων κουμπιών
2. Λειτουργία αισθητήρων (θερμοκρασίας και φωτεινότητας)
3. Μεταγωγή από ένα σήμα σε τάση δικτύου
4. Λειτουργία lcd οθόνης

Παρακάτω θα δωθούν σε εικόνες αναλυτικά τα κυκλώματα που έγιναν κατά τις δοκιμές με την βοήθεια του 123D CIRCUITS της AYTODESK. Η εφαρμογή αυτή είναι online και διατίθεται δωρεάν. Είναι ειδικά φτιαγμένη για εξομειώσεις με την μικροεπεξεργαστική κάρτα Arduino Uno Rev3. Περιέχει την κάρτα Arduino Uno Rev3 και μία λίστα εξαρτημάτων που χρειάζονται στις περισσότερες εφαρμογές με αυτή τη κάρτα όπως led, αντιστάσεις, μοτεράκια και άλλα.

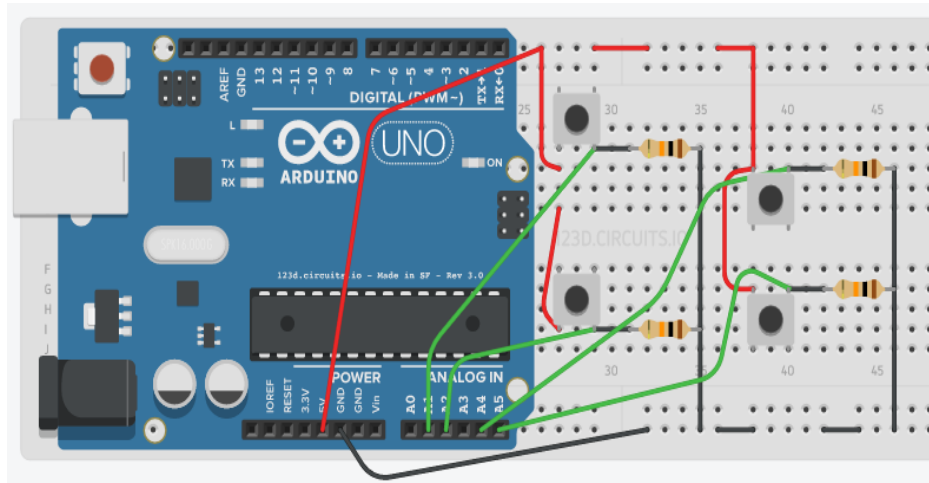
### ΛΕΙΤΟΥΡΓΕΙΑ ΤΕΣΣΑΡΩΝ ΚΟΥΜΠΙΩΝ

Κάθε I/O (είσοδος – έξοδος) της κάρτας Arduino μπορεί να παρέχει ή να δεχτεί μέχρι 5V (dc) και 40mA. Για να μην καεί κάθε pin τοποθετείται και μία προστατευτική αντίσταση μεταξύ pin και γείωσης. Το κουμπί τοποθετείται ανάμεσα στα 5V και στο pin.

Switch connected to a digital input on the arduino



Αυτή είναι η διάταξη του κυκλώματος. Έχουν επιλεγθεί οι αναλογικές εισοδοι A1, A2, A3 και A4. Η επιλογή αυτή έγινε κυρίως επειδή οι ψηφιακές εισοδοι – έξοδοι χρειάζονται για άλλο σκοπό και είναι περιορισμένες. Παρακάτω φαίνεται το κύκλωμα το οποίο φτιάχτηκε για την δοκιμή.



Πέρα από το hardware για να υπάρχει νόημα στο κύκλωμα πρέπει να οριστεί ο στόχος αυτής της διάταξης. Υπάρχουν τέσσερα κουμπιά και με αυτά ο στόχος είναι να ελέγχονται τα όρια μίας θερμοκρασίας σε έναν χώρο στον οποίο θα ανάβουν θερμαντικά και ψηκτικά σώματα. Αν οι προκαθορισμένες τιμές θερμοκρασίας είναι τα θερμαντικά να ανάβουν στους 15°C και τα ψηκτικά στους 30°C τότε υπάρχουν τέσσερα κουμπιά τα οποία κάνουν το εξής: α) ένα κουμπί αυξάνει το πάνω όριο θερμοκρασίας, β) ένα κουμπί μειώνει το πάνω όριο θερμοκρασίας, γ) ένα κουμπί αυξάνει το κάτω όριο θερμοκρασίας, δ) και ένα κουμπί μειώνει το κάτω όριο θερμοκρασίας. Οπότε τα όρια της θερμοκρασίας αυξομειώνονται όπως θέλει κανείς. Όλα αυτά προγραμματίζονται με την βοήθεια της κάρτας Arduino. Στην περίπτωση όμως που κάποιος πατά συνεχώς ένα κουμπί π.χ. το κουμπί που μειώνει το καάτω όριο της θερμοκρασίας η τιμή θα τίνει να φτάσει σε τιμές παράλογες και θα είναι δύσκολο να επανέλθει. Για αυτό το λόγο έχει προγραμματιστεί όταν οι τιμές φτάσουν σε άσκοπες τιμές να επανέρχονται στις αρχικές.

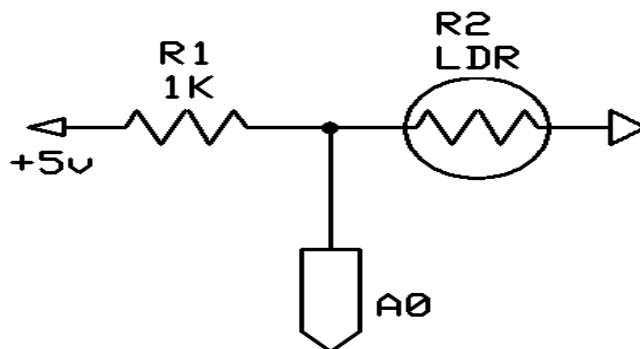
## ΛΕΙΤΟΥΡΓΕΙΑ ΑΙΣΘΗΤΗΡΩΝ

Η συνδεσμολογία για τον αισθητήρα φωτεινότητας ακολουθεί την ίδια λογική με το κύκλωμα των κουμπιών. Όπου στο προηγούμενο κύκλωμα θα έμπνεε κουμπί, τώρα μπαίνει ο αισθητήρας φωτεινότητας που φυσικά δεν έχει πολικότητα αλλά μπαίνει από οποιαδήποτε πλευρά.

Πέρα απο την απλότητα του κυκλώματος όμως υπάρχει ένα εμπόδιο. Όταν υπάρχει πολύ φως η αντίσταση μειώνεται και το αντίστροφο. Αυτό σημαίνει πως όταν δεν υπάρχει φως δεν δέχετε κάποιο σήμα η κάρτα. Δέχεται μόνο όταν υπάρχει φως. Αλλά ο σκοπός της εργασιας είναι όταν βραδιάζει να ενεργοποιείται ένα ρελέ και όχι όταν είναι μέρα. Για να ληθεί αυτό το πρόβλημα προγραμματίζουμε την κάρτα να δίνει σήμα όσο δεν διαρρέεται από ρεύμα το pin που

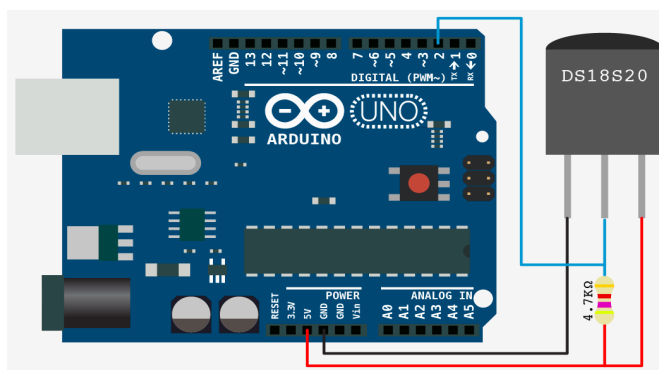
τοποθετήθηκε ο αισθητήρας και να μην δίνει σήμα όσο διαρέεται.

Ο αισθητήρας είναι τοποθετημένος στην αναλογική είσοδο A0. Οι αισθητήρες μπορούν να τοποθετηθούν μόνο στις αναλογικές εισόδους.

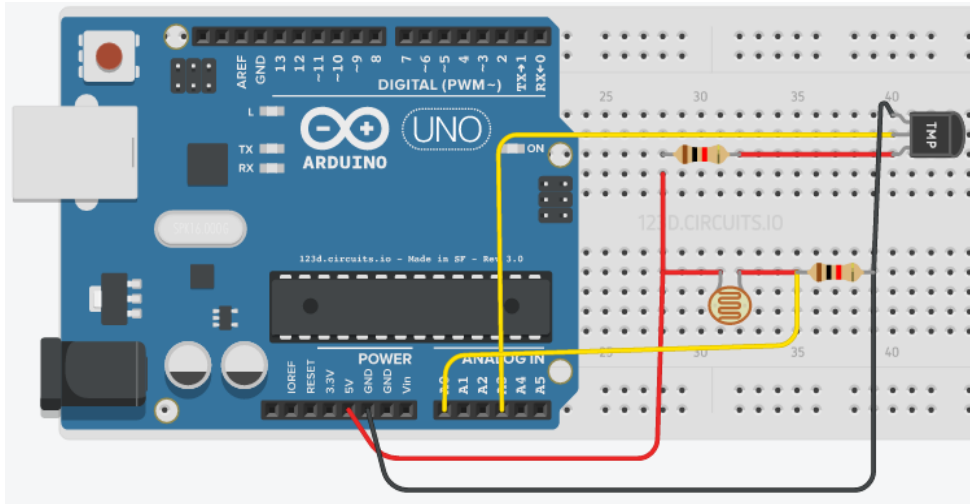


Ο αισθητήρας θερμοκρασίας έχει ανάποδο τρόπο τοποθέτησης πάνω στο κύκλωμα. Ο αισθητήρας αυτός είναι ο DS18B20 και έχει πολλούς τρόπους με τους οποίους μπορεί να συνδεθεί στο κύκλωμα. Κάθε τρόπος γίνεται για διαφορετικούς σκοπούς. Στην εργασία αυτή επιλέχθηκε να τοποθετηθεί σε με τον πιο ασφαλή τρόπο. Τον τρόπο σηλαδή που είναι πιο διαδεδομένος και θα μας δώσει σιγουρα αποτελεσματα και με ακριβεια. Επίσης κάθε τρόπος έχει ανάγκη και απο τον ανάλογο κώδικα. Ο κώδικας θα μελετηθεί παρακάτω. Ο DS18B20 μπορεί να μετρά απο  $-55^{\circ}\text{C}$  μέχρι  $125^{\circ}\text{C}$  και από τους  $-10^{\circ}\text{C}$  μέχρι τους  $85^{\circ}\text{C}$  έχει αστοχία  $\pm 0.5^{\circ}\text{C}$ .

Η συνδεσμολογία είναι ανάποδη από αυτή των προηγούμενων εξαρτημάτων. Τώρα μεταξύ analog pin του arduino και της γείωσης τοποθετείται ο αισθητήρας και όχι κάποια προστατευτική αντίσταση και μεταξύ του pin και της τάσης 5V τοποθετείται η αντίσταση που προστατεύει το pin. Ο αισθητήρας αυτός τοποθετήθηκε στην αναλογική είσοδο A3.



Παρακάτω φαίνεται πως είναι τοποθετημένα τα εξαρτήματα πάνω στην πλακέτα.



## ΜΕΤΑΓΩΓΗ ΑΠΟ ΕΝΑ ΣΗΜΑ ΣΕ ΤΑΣΗ ΔΙΚΤΥΟΥ

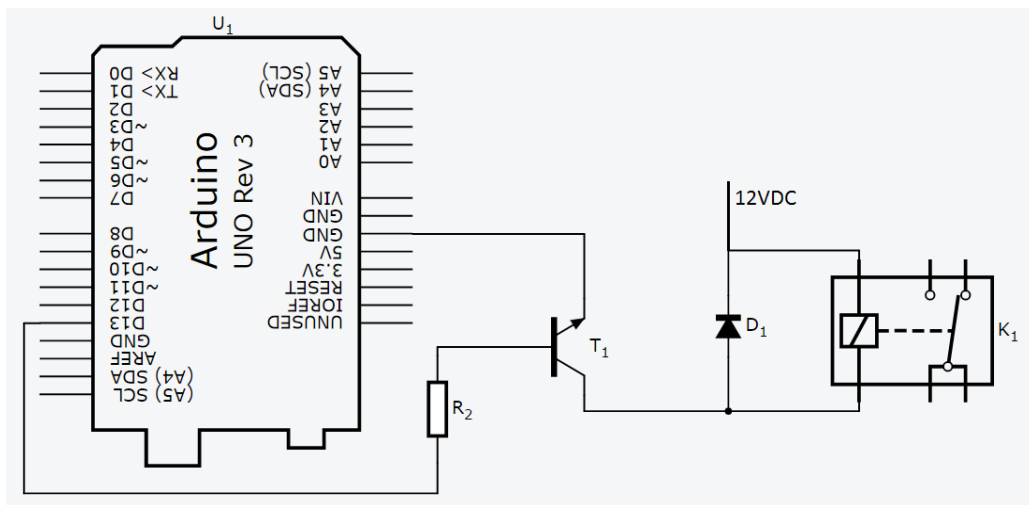
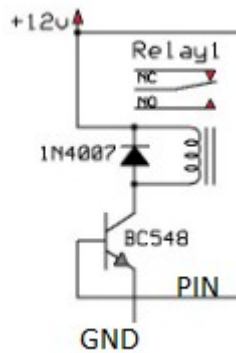
Σχεδόν τελικό στάδιο της κατασκευής είναι η μεταγωγή από ένα απλό σήμα 5V (dc) που βγάζει μία ψηφιακή έξοδο της κάρτας σε τάση 230V (ac). Αυτή η μεταγωγή γίνεται εφαρμόζοντας στην κάθε έξοδο που θέλουμε ένα ρελέ (πλακέτας για την συγκεκριμένη εφαρμογή). Το ρελέ που χρησιμοποιείται σε αυτό το σύστημα είναι ένα ρελέ με είσοδο 12Vdc και 1C/O μεταγωγική επαφή από την οποία δεν χρησιμοποιείται καθόλου η Normal Close. Οπότε χρειάζεται και ένα τρανζίστορ NPN το οποίο παίρνει το σήμα και το ενισχύει για να μπορεί να σπλίζει το ρελέ χωρίς προβλήματα. Η ενίσχυση αυτή γίνεται με μία τάση 12Vdc. Στην περίπτωση όπου δεν υπήρχε η ενίσχυση, το ρελέ θα “υποόπλιζε” και έτσι θα υπήρχε πρόβλημα στο σύστημα. Τέλος υπάρχει η ανάγκη να προστατευτεί η κάρτα από επιρροές του δικτύου. Για αυτό το λόγο τοποθετείται παράλληλα με το ρελέ μία διάδος έτσι ώστε να μην στραφεί ρεύμα για κανέναν λόγο προς το pin της κάρτας και ο μόνος δρόμος να είναι προς την γείωση.

Οι έξοδοι που χρειάστηκαν για αυτό το κομμάτι είναι τρεις.

1. Για ενεργοποίηση θερμαντικών σωμάτων
2. Για ενεργοποίηση ψυκτικών σωμάτων
3. Για ενεργοποίηση γραμμής εξωτερικού φωτισμού

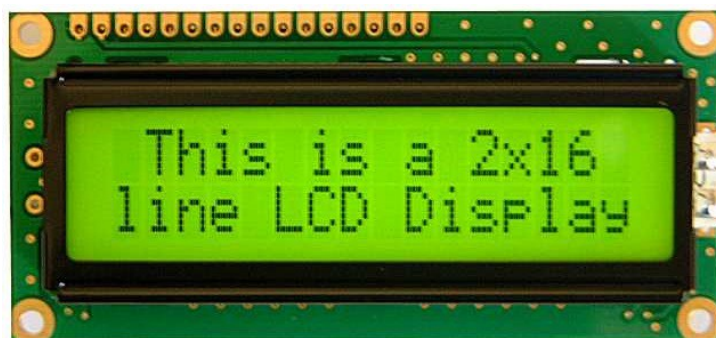
Οι έξοδοι που χρειάστηκαν είναι D8, D9 και D13. Παρακάτω φαίνεται λεπτομερές η συνδεσμολογία για την δοκιμή τους.





## ΛΕΙΤΟΥΡΓΕΙΑ LCD ΟΘΟΝΗΣ

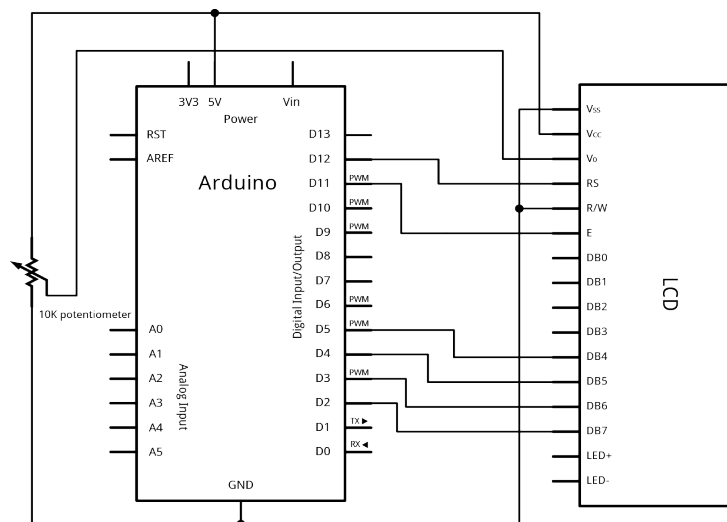
Αφού όλα τα βασικά μέρη της εργασίας έχουν δοκιμαστεί και λειτουργούν με τα επιθυμητά αποτελέσματα, μπαίνει μία οθόνη lcd 16x2 για να φαίνονται τα αποτελέσματα των υπόλοιπων εργασιών συγκεντρωμένα σε ένα σημείο.



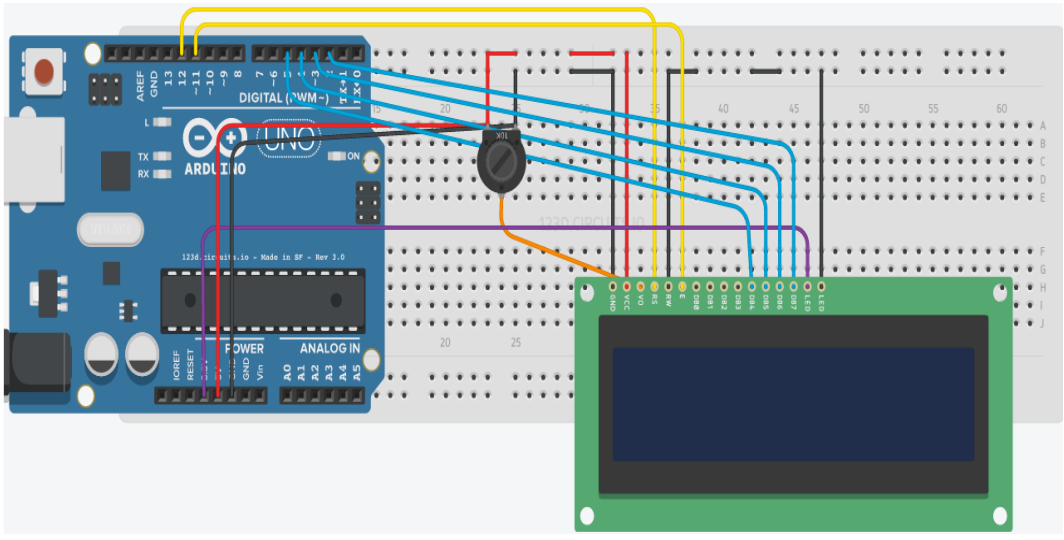
Η οθόνη χαρακτηρίζεται lcd 16x2 white on blue. Ο χαρακτηρισμός lcd σημαίνει πως είναι μία οθόνη υγρών κρυστάλλων (liquid crystal display). Η επιλογή τέτοιας τεχνολογίας οθόνης έγινε γιατί, με βάση τις ανάγκες που χρειαζόντουσαν στην εργασία, μια τέτοια οθόνη είναι επαρκής στις δυνατότητες που της ζητούνται σε αυτή την εφαρμογή, αποτελεί φθινό και ποιοτικό προϊόν, είναι εύκολη στον προγραμματισμό της και έχει χρησιμοποιηθεί σε πολλές εφαρμογές με Arduino. Ο χαρακτηρισμός 16x2 σημαίνει πως στην οθόνη φαίνονται 2 γράμμες όπου η κάθε μια χωρά μέχρι και 16 χαρακτήρες, και ο χαρακτηρισμός white on blue σημαίνει πως τα γράμματα είναι άσπρα σε μπλε φόντο. Η φωτεινότητα των γραμμάτων ρυθμίζεται με ένα ποτενσιόμετρο των 10KOhms.

Αυτή η οθόνη έχει 16 pins τα οποία εκτός από αυτά που είναι γειώσεις και τροφοδοσίες φόντου και γραμμάτων, παίρνουν δεδομένα από την κάρτα προς την οθόνη. Στην άσκηση αυτή δεν χρειάζονται όλες αυτές. Υπάρχουν τέσσερις που αφήνονται χωρίς καμία σύνδεση. Αυτές είναι οι DB0, DB1, DB2, DB3.

Παρακάτω φαίνεται κάθε pin της οθόνης τι χρησιμότητα έχει και ποιά είναι η σειρά τους πάνω στη πλακέτα.



Παρακάτω φαίνεται το κύκλωμα το οποίο δημιουργήθηκε για τις δοκιμές με την οθόνη.



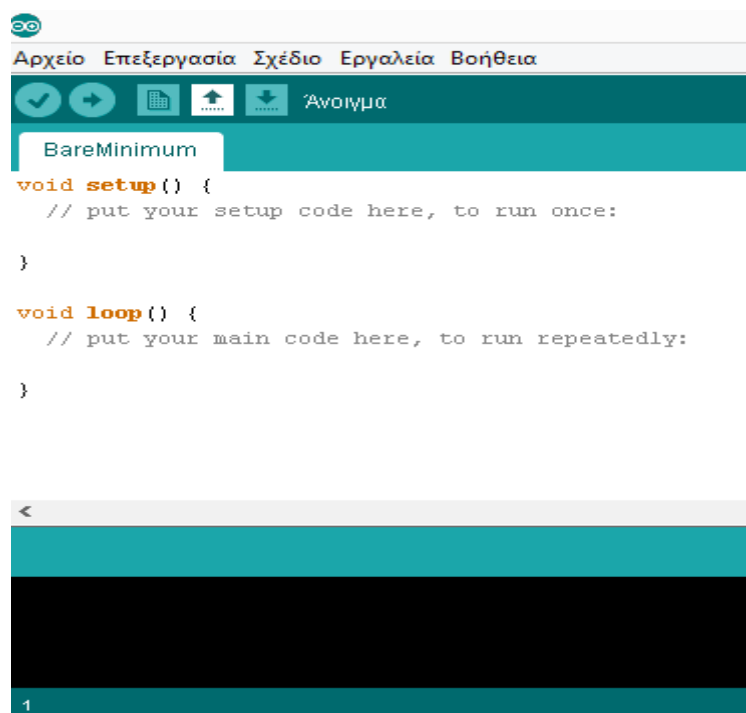
## ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΕΞΗΓΗΣΗ ΚΩΔΙΚΑ

Παρακάτω θα επεξηγηθεί ο κώδικας που διατρέχει την κάρτα Arduino έτσι ώστε να έχουμε τα επιθυμητά αποτελέσματα.

Για να γίνει πιά σαφές ο τρόπος λειτουργίας του κώδικα, είναι αναγκαίος ο χωρισμός του σε κομμάτια τα οποία είναι:

- 1.1 Δήλωση μεταβλητών κ.λ.π.
- 1.2 Προσδιορισμός εισόδων και εξόδων.
- 1.3 Επαναλαμβανόμενος κώδικας.
  - 1.3.1 Διάβασμα και προγραμματισμός αισθητήρα DS18B20
  - 1.3.2 Διάβασμα και προγραμματισμός κουμπιών
  - 1.3.3 Διάβασμα και προγραμματισμός αισθητήρα φωτεινότητας
  - 1.3.4 Τοποθέτηση λέξεων και τιμών στην οθόνη

Πρέπει να ξεκαθαριστεί πρώτα το γενικό πλάνο ενός προγράμματος στο ARDUINO IDE. Το ελάχιστο πρόγραμμα φαίνεται παρακάτω.



```
BareMinimum
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}

1
```

Παρακάτω φαίνεται ο κωδικας που δημιουργήθηκε σε τμήματα

### ΔΗΛΩΣΗ ΜΕΤΑΒΛΗΤΩΝ Κ.Α.Π.

Η δήλωση και δημιουργία των μεταβλητών των σταθερών και των βιβλιοθηκών και γενικά όλα τα είδη των δηλώσεων γίνεται στην αρχή έτσι ώστε να μην υπάρχει σύγχυση στον υπόλοιπο κώδικα.

```
oIokIhromeno_me_lcd$
word x;
const int buttonPin2 = A1;
const int buttonPin3 = A2;
const int buttonPin4 = A4;
const int buttonPin5 = A5;
int buttonPushCounter2 = 30;
int buttonState2 = 0;
int lastButtonState2 = 0;
int buttonPushCounter4 = 15;
int buttonState3 = 0;
int lastButtonState3 = 0;
int buttonState4 = 0;
int lastButtonState4 = 0;
int buttonState5 = 0;
int lastButtonState5 = 0;
#include <OneWire.h>
#include <LiquidCrystal.h>
#include <DallasTemperature.h>
const int red = 9 ;
const int green = 8 ;
#define ONE_WIRE_BUS A3
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
int photocellPin = A0;
int photocellReading;
int LEDpin = 13;
int LEDbrightness;
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

## ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΕΙΣΟΔΩΝ ΚΑΙ ΕΞΟΔΩΝ

Έπειτα προσδιορίζεται και δηλώνεται σε ποιά pin θα τοποθετηθεί το κάθε εξάρτημα και αν αυτό θα είναι είσοδος ή έξοδος. Αυτό το βήμα γράφεται στην void setup όπου δεν διαβάζεται συνεχώς αλλά μόνο στην αρχή του προγράμματος και μετά από κάθε reset. Τα στάδια αυτά (αυτό και το παραπάνω) μπορούν να γραφουν και στην void loop. Απλά χωρίς καμία βελτίωση το πρόγραμμα θα γίνει πιο αργό.

```
oIokIhromeno_me_lcd
void setup(){
  lcd.begin(16, 2);
  pinMode(LEDpin,OUTPUT);
  pinMode(buttonPin2,INPUT);
  pinMode(buttonPin3,INPUT);
  pinMode(buttonPin4,INPUT);
  pinMode(buttonPin5,INPUT);

  pinMode(red,OUTPUT);
  pinMode(green,OUTPUT);
  sensors.begin();
}
```

Προσδιορισμός εισόδων και εξόδων. Οι εντολές αυτές εκτελούνται μόνο μια φορά.

# ΕΠΙΧΡΩΜΕΝΟΣ ΚΩΔΙΚΑΣ

## ΔΙΑΒΑΣΜΑ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΑΙΣΘΗΤΗΡΑ DS18B20 ΚΑΙ ΚΟΥΜΠΙΩΝ

Στην συνέχεια αφού τελιώσει η void setup ο κώδικας συνεχίζεται στην void loop όπου το πρόγραμμα τρέχει “συνεχώς”. Εδώ διαβάζεται η θερμοκρασία ,προσδιορίζονται τα όρια της νεκρής περιοχής της θερμοκρασίας, μετράται σε ποιά κατάσταση βρίσκονται τα κουμπιά και τί δηλώνουν και συσχετίζονται τα όρια της θερμοκρασίας με τα όρια τα οποία έχουν δηλωθεί από τα κουμπιά.

```
oIokhromeno_me_lcd
void loop(){
  sensors.requestTemperatures();
  if (sensors.getTempCByIndex(0)>=buttonPushCounter2){
    digitalWrite(red,HIGH);
  }
  else if ((sensors.getTempCByIndex(0)>=buttonPushCounter4)&&(sensors.getTempCByIndex(0)<buttonPushCounter2)){
    digitalWrite(red,LOW);
    digitalWrite(green,LOW);
  }
  else if(sensors.getTempCByIndex(0)<buttonPushCounter4){
    digitalWrite(green,HIGH);
  }
  buttonState2 = digitalRead(buttonPin2);
  buttonState3 = digitalRead(buttonPin3);
  buttonState4 = digitalRead(buttonPin4);
  buttonState5 = digitalRead(buttonPin5);

  if (buttonState2 != lastButtonState2) {

    if (buttonState2 == HIGH) {

      buttonPushCounter2++;
    }
  }

  if (buttonState3 != lastButtonState3) {

    if (buttonState3 == HIGH) {

      buttonPushCounter2--;
    }
  }
  if (buttonState4 != lastButtonState4) {

    if (buttonState4 == HIGH) {

      buttonPushCounter4++;
    }
  }
  if (buttonState5 != lastButtonState5) {

    if (buttonState5 == HIGH) {

      buttonPushCounter4--;
    }
  }
}
```

Διάβασμα του DS18B20.  
Δήλωση ανώτατων και κατώτατων οριακών θερμοκρασιών για την κατάσταση ηρεμίας.

Διάβασμα των κουμπιών.  
Προγραμματισμός υπολογισμού των δραστηριοτήτων των κουμπιών.

## ΔΙΑΒΑΣΜΑ ΚΑΙ ΤΟΠΟΘΕΤΗΣΗ ΑΙΣΘΗΤΗΡΑ ΦΩΤΕΙΝΟΤΗΤΑΣ

Παρακάτω διαβάζεται ο αισθητήρας, “χαρτογραφείται” έτσι ώστε να δώθει αινεσθηία στο διάβασμα του αισθητήρα και προσδιορίζεται το όριο ενεργοποίησης της εξόδου που επιθυμείτε.

Ο αισθητήρας όμως όταν έχει φως διεγείρει το σύστημα , ενώ όταν δεν έχει το δεν διεγείρει. Όμως δεν είναι επιθυμητό αυτο. Οπότε χρησιμοποιήθηκε μια εντολή που αντιστρέφει αυτή τη σχέση διέγερσης και φωτός.

```
photocellReading = analogRead(photocellPin);  
photocellReading = 1023 - photocellReading;
```

Τώρα όποτε υπάρχει σκοτάδι το σύστημα διεγείρεται οπως είναι επιθυμητό.

```
oIokhromeno_me_lcd
```

```
photocellReading = analogRead(photocellPin);  
photocellReading = 1023 - photocellReading;  
LEDbrightness = map(photocellReading, 0, 1023, 0, 255);
```

```
if (LEDbrightness>200)  
{  
  digitalWrite(LEDpin,HIGH);  
  lcd.setCursor(6, 1);  
  lcd.println("On ");  
}  
else if(LEDbrightness<200)  
{  
  digitalWrite(LEDpin,LOW);  
  lcd.setCursor(6, 1);  
  lcd.println("Off");  
}
```

Διάβασμα του  
φωτοαισθητήρα και  
δήλωση προκαθορισμένης  
τιμής αλλαγής κατάστασης  
εξόδου.

## ΤΟΠΟΘΕΤΗΣΗ ΛΕΞΕΩΝ ΚΑΙ ΤΙΜΩΝ ΣΤΗΝ ΟΘΟΝΗ

Η οθόνη που χρησιμοποιείται είναι 16x2. Οπότε για να χωρέσουν όλα όσα χρειάζονται σε έναν τόσο περιορισμένο χώρο πρέπει να τοποθετηθούν όλα τακτοποιημένα και όχι αυθαίρετα. Έπειτα δηλώνεται η συχνότητα επανάληψης του προγράμματος όπου είναι και το τελικό στάδιο της void loop και ο'λου του προγράμματος.

Παρακάτω φαίνεται ο τελικός κώδικας. Με το που δοκιμαστεί ο κώδικας σε ένα μαύρο πλαίσιο στο κάτω μέρος του compiler φαίνεται η χωρητικότητα του προγράμματος.

```
olokhromeno_me_lcd
```

```
lcd.setCursor(0,0);  
lcd.print("Temp:");  
lcd.print(sensors.getTempCByIndex(0));  
lcd.setCursor(11, 0);  
lcd.print("UP:");  
lcd.println(buttonPushCounter2);  
lcd.setCursor(0, 1);  
lcd.print("Light");  
lcd.setCursor(11, 1);  
lcd.print("DN:");  
lcd.println(buttonPushCounter4);  
delay(100);  
}
```

Τοποθέτηση λέξεων και τιμών πάνω στην οθόνη.

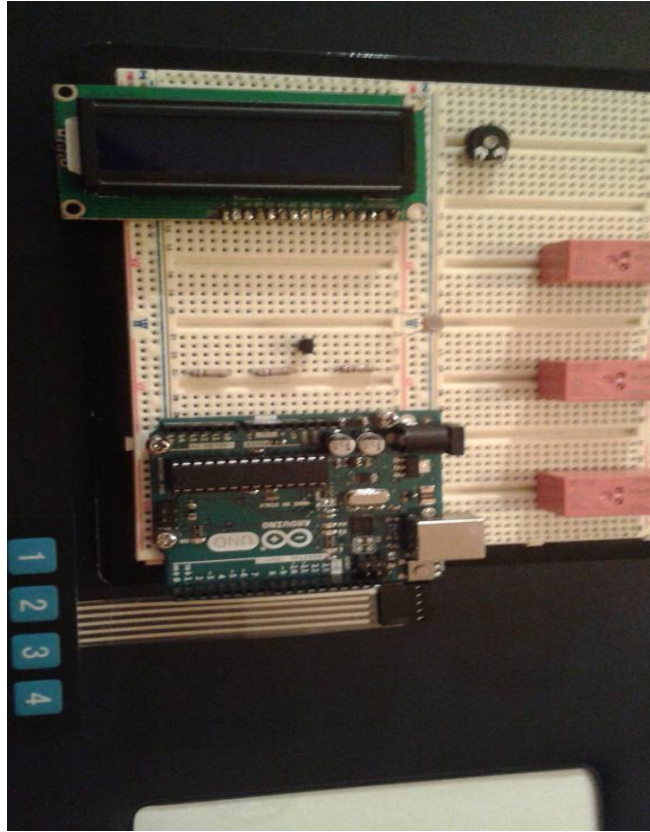
Δήλωση συχνότητας επανάληψης του κώδικα.

```
<
```

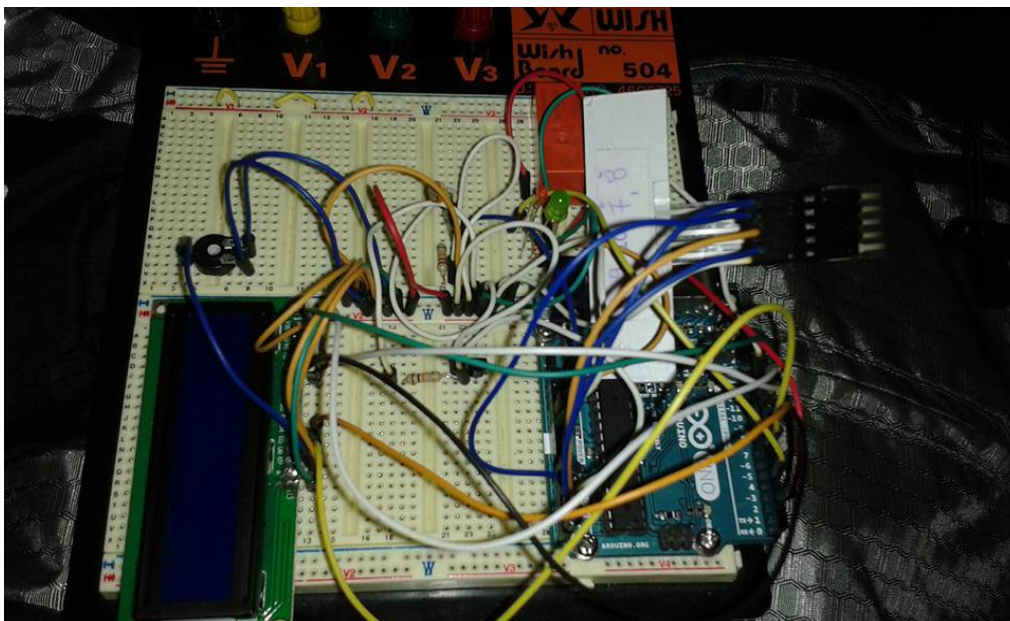
```
102
```



## ΣΤΑΔΙΑ ΥΛΟΠΟΙΗΣΗΣ ΕΡΓΑΣΙΑΣ



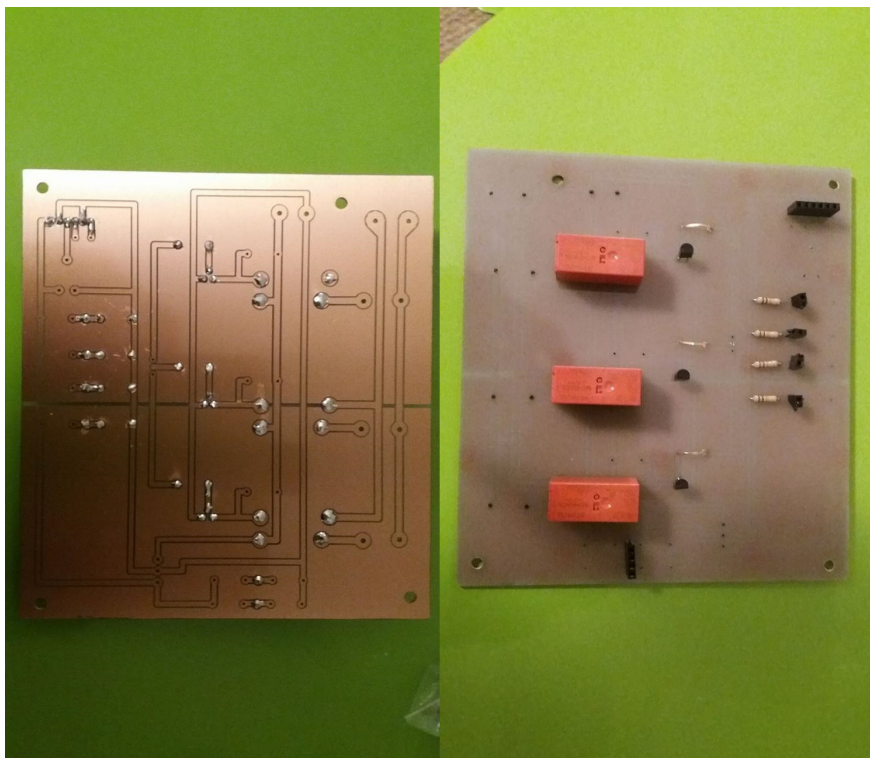
*Εικόνα 1: Τοποθέτηση βασικών εξαρτημάτων σε πλακέτα*



*Εικόνα 2: Δοκιμή μέρος λειτουργιών*



*Εικόνα 3: Πλακέτα έτοιμη προς χάραξη*



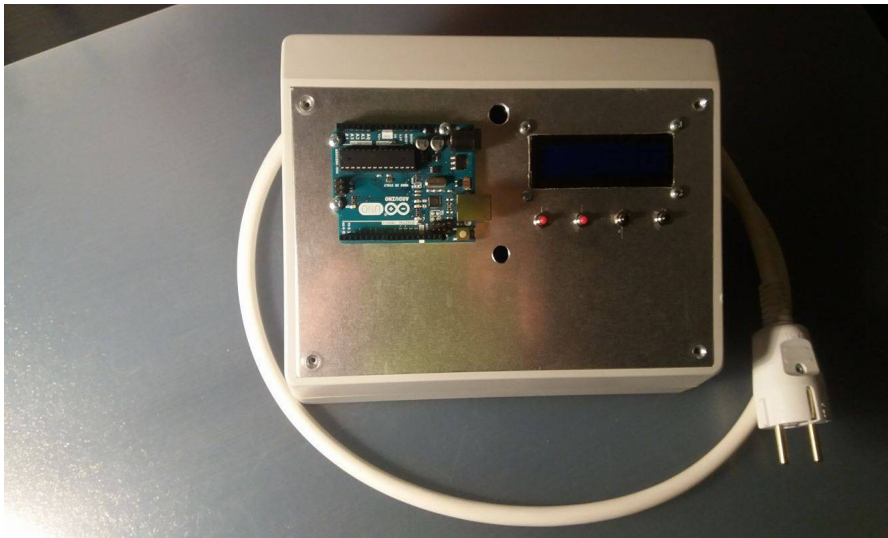
*Εικόνα 4: Αριστερά: Χαραγμένη πλακέτα Δεξιά: Τοποθέτηση εξαρτημάτων*



*Εικόνα 5: Πάνω: Χαραγμένο κουτί κατασκευής εμπρός Κάτω: Χαραγμένο κουτί κατασκευής πίσω*



*Εικόνα 6: Τοποθέτηση πλακέτας στο κουτί κατασκευής*



*Εικόνα 7: Στήσιμο κατασκευής*

## ΠΙΝΑΚΑΣ ΑΠΑΡΤΙΩΝ

Παρακάτω φαίνονται τα υλικά όπου χρειάστηκαν για την πραγματοποίηση του συνολικού κυκλώματος και το κουτί του.

Πίνακας Απαρτιών		
Είδος Τεμαχίου	Περιγραφή Τεμαχίων	TMX
Πλακέτα μικροεπεξεργαστή	Arduino Uno Rev3	1
Οθόνη	LCD 16x2 White on Blue	1
Μπουτόν	Momentary Button (Κόκκινο ή Μαύρο)	4
Καλώδια εντολών	Jumper Wires (15cm)	~ 40
Αισθητήρας θερμοκρασίας	DS18B20	1
Αισθητήρας φωτηνότητας	LDR (Φωτοαντίσταση)	1
Καλώδιο τροφοδοσίας	3x16A	1 μέτρο
Αρσενικό φισ	Φισ για πρίζα οικίας	1
Πλακέτα χαλκου	Single side 10x14cm	1
Ποτενσιόμετρο	0 – 10KOhms	1
Αντιστάσεις	10KOhms	9
Power Jack	230V 16A	3
Ρελέ πλακέτας	RT114012 12VDC	3
Δίοδοι	1N4007	3
Τρανζίστορ	BC548	3
Ρυθμιστής τάσης	LM7805	1
Αρσενικά pins	Pin Headers 1x6 Male	10
Θυληκά pins	Pin Headers 1x6 Female	10
Τροφοδοτικά	Power Supply 12V 2A	1
Κουτί κατασκευής	14,5x21,5x10cm	1
Καλώδιο μεταφοράς δεδομένων	USB A to B	1

Το κόστος για όλη την εργασία είναι περίπου στα 65 €. Η τιμή αυτή μπορεί να διαφέρει καθώς τα αναλώσιμα διαφέρουν στις τιμές απο κατάστημα σε κατάστημα .

## ΠΗΓΕΣ

- [1] Αυτοματισμός, <https://el.wikipedia.org/wiki/%CE%91%CF%85%CF%84%CE%BF%CE%BC%CE%B1%CF%84%CE%B9%CF%83%CE%BC%CF%8C%CF%82>, (Ανάκτηση την 8/5/16).
- [2] Aytodesk 123D Circuits, <https://123d.circuits.io/>, (Ανάκτηση την 6/5/16).
- [3] Εισαγωγή στο Arduino - Το απόλυτο geek toy, <https://deltahacker.gr/arduino-intro/>, (Ανάκτηση την 3/2/16).
- [4] Εισαγωγή στο Arduino, <https://projectmaniacs.wordpress.com/2014/11/22/%CE%B5%CE%B9%CF%83%CE%B1%CE%B3%CF%89%CE%B3%CE%AE-%CF%83%CF%84%CE%BF-arduino/>, (Ανάκτηση την 3/2/16).
- [5] Arduino Home, <https://www.arduino.cc/>, (Ανάκτηση την 17/1/16).
- [6] Photoresistor, <https://en.wikipedia.org/wiki/Photoresistor>, (Ανάκτηση την 6/5/16).
- [7] DS18B20 Datasheet, <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf> (Ανάκτηση την 19/4/16).
- [8] Arduino Projects, <https://www.youtube.com/>, (Ανάκτηση την 10/6/15).
- [9] Grobotronics Robotics & Electronics <http://grobotronics.com/>, (Ανάκτηση την 10/6/15).