

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΟΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ 1676

Ανάπτυξη συστήματος αυτοματισμών θερμοκηπίου

Greenhouse automation system development

Ανδρέας Μπαρζούκας 6425

Εποπτεύον καθηγητής: Μπισδούνης Λάμπρος

Πάτρα 2018

Περιεχόμενα:	Σελίδες:
1. Περίληψη	3
2. Πρόλογος	3-4
3. Αισθητήρια	4-18
4. Arduino	18
4.1 Γενικά για το Arduino	18-21
4.2 Arduino uno r3	21-28
4.3 Arduino Shields	28-29
5. Γλώσσα προγραμματισμού C	29
5.1 Γενικά για την C	29-49
6. Τροφοδοσία Arduino	49-51
7. Διαδικασία και Υλοποίηση Διπλωματικής	51
7.1 Κώδικας Διπλωματικής	51-59
7.2 LCD Screen 16x2	60-62
7.3 Πρόγραμμα Arduino IDE	62-68
7.4 Πίνακας Υλικών	68-69
7.5 Κομμάτια Project Διπλωματικής	69-79
7.6 Διαδικασία Υλοποίησης κατασκευής	79-83
Πηγές	84-85

1. Περίληψη

Με αυτό το σύστημα θα γίνεται αυτόματος έλεγχος θερμοκρασίας, αυτόματο πότισμα και αυτόματο άνοιγμα φωτός. Ο έλεγχος θερμοκρασίας θα γίνεται με ένα "κλιματιστικό"(ανεμιστηράκι στην προκειμένη περίπτωση) το οποίο θα ενεργοποιείται και ταυτόχρονα θα ανάβει και ένα πράσινο LED, όποτε ένα αισθητήριο θερμοκρασίας ανιχνεύει το ανώτατο προκαθορισμένο όριο θερμοκρασίας. Όταν θα πέφτει η θερμοκρασία, θα σβήνει το ανεμιστηράκι. Επίσης, ένα αισθητήριο υγρασίας εδάφους, θα ανιχνεύει την σχετική υγρασία και όταν εντοπίζει την υγρασία να βρίσκεται στο κατώτατο όριο υγρασίας, και πιο χαμηλά, θα ενεργοποιεί μια αντλία και θα στέλνει νερό για να ποτιστεί το φυτό. Όταν το επίπεδο υγρασίας ανέβει, τότε η αντλία νερού θα κόβει αυτόματα. Επιπλέον, ένας αισθητήρας φωτός θα ανιχνεύει την ένταση του φωτός και θα ανάβει μια λάμπα. Η ένταση του φωτός διακρίνεται σε 3 επίπεδα. Ένα πολύ φωτεινό, ένα μεσαίας φωτεινότητας και ένα χαμηλής φωτεινότητας.

2. Πρόλογος

Η ανθρωπότητα και ο τρόπος ζωής των ανθρώπων, άλλαξε ριζικά, μετά την δημιουργία και την ένταξη των αυτοματισμών στην ζωή τους. Όλα ξεκίνησαν τον 20ο αιώνα, όταν οι αυτοματισμοί εντάχθηκαν στην βιομηχία αλλά και στην καθημερινή ζωή των ανθρώπων. Αυτό αυτόματα συνεπαγόταν με την κατάργηση πολλών θέσεων εργασίας στον τομέα της βιομαχίας, οι οποίες, ως επί το πλείστον συσχετιζονταν με την παραγωγή. Ένα παράδειγμα, είναι η συναρμολόγηση παιχνιδιών, η οποία στις αρχές συνήθιζε να πραγματοποιείτε με την εκμετάλλευση του ανθρωπίνου δυναμικού. Επίσης ένα άλλο παράδειγμα στο οποίο φαίνεται η σημασιολογία της ένταξης των αυτοματισμών είναι στην συναρμολόγηση οχημάτων(μικρότερο προς μηδαμινό ποσοστό λάθους σε σχέση με το ποσοστό λάθους του ανθρωπίνου παράγοντα). Οι αυτοματισμοί, μπορούν να χρησιμοποιηθούν με πολλούς τρόπους και κυρίως με διάφορα κομμάτια εξοπλισμού. Ένα είδος τμήματος εξοπλισμού, είναι το ρελέ και ο συνδιασμός του με διακόπτες και αισθητήρια αλλά και με PLC. Ένα άλλο παράδειγμα είναι η χρήση Arduino. Το Arduino(UNO, 101, MEGA κτλ), χρησιμοποιείται σε συνδιασμό με γλώσσα προγραμματισμού(C, C++), μαζί με διάφορους αισθητήρες(φωτός, κίνησης κ.λπ.).

Πιο συγκεκριμένα, όσον αφορά τον αυτοματισμό:

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 μεταφορών 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. Σε αυτήν την διπλωματική έχουμε έναν αυτοματισμό με 3 εισόδους και 11 εξόδους. Μια για τον αισθητήρα φωτός, μια για τον αισθητήρα υγρασίας και άλλη μια για τον αισθητήρα θερμοκρασίας. Μια έξοδος για τον ανεμιστήρα, 3 για τα 3 LED του διαιρέτη τάσης, 6 για την LCD οθόνη, μια έξοδος για το LED της αντλίας (πιο συγκεκριμένα για την προσομοίωση του ποτίσματος). Η επεξεργασία σήματος και η μοντελοποίηση γίνεται μέσω ενός μικροεπεξεργαστή (κάρτα Arduino UNO).

3. Αισθητήρια

Γενικά:

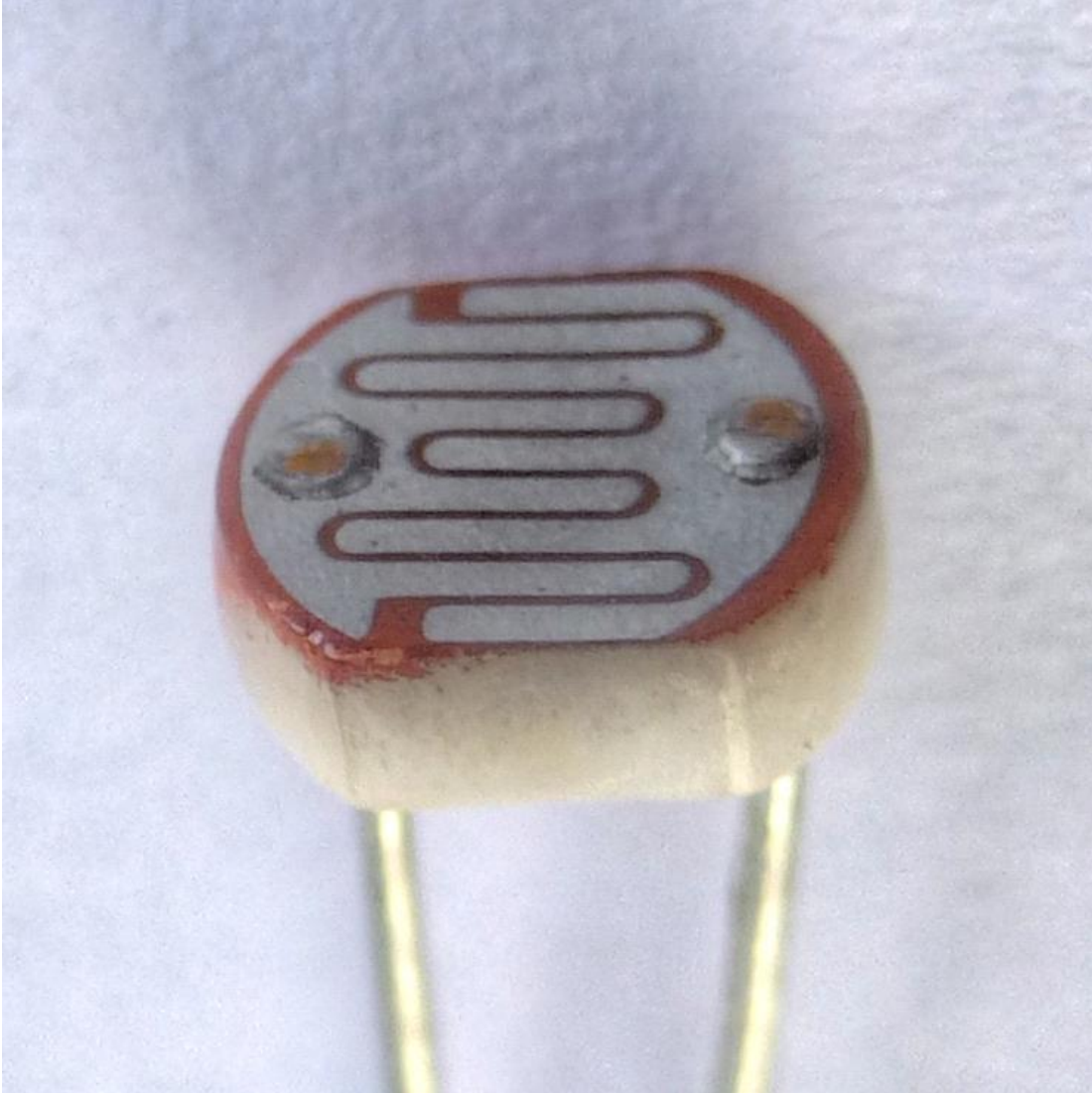
Αισθητήρας ονομάζεται μία συσκευή που ανιχνεύει ένα φυσικό μέγεθος και παράγει από αυτό μία μετρήσιμη έξοδος. Για παράδειγμα, το υδραργυρικό θερμομέτρο μετατρέπει τη μετρούμενη θερμοκρασία σε διαστολή, η οποία μπορεί να αναγνωστεί από ένα βαθμονομημένο σωλήνα. Οι αισθητήρες χρησιμοποιούνται σε καθημερινά αντικείμενα, όπως κουμπιά ανελκυστήρων ευαίσθητα στην αφή και λάμπες φωτισμού που εκπέμπουν λαμπρότερα ή απαλότερα αγγίζοντας τη βάση τους. Υπάρχουν αναρίθμητες ακόμη χρήσεις που οι περισσότεροι άνθρωποι δεν αντιλαμβάνονται. Εφαρμογές τους συναντούμε στα αυτοκίνητα, σε μηχανές, στην αεροναυπηγική, την ιατρική, τη βιομηχανία και τη ρομποτική.

Χαρακτηριστικά αισθητήρων:

- Εύρος: Τα όρια στα οποία η συσκευή λειτουργεί αξιόπιστα.
- Ακρίβεια: Η εγγύτητα της τιμής εξόδου προς τη τιμή εισόδου.
- Σφάλμα: Η διαφορά ανάμεσα στη μετρούμενη τιμή και τη πραγματική τιμή.
- Ανοχή: Το μέγιστο σφάλμα που μπορεί να δημιουργήσει ο αισθητήρας.
- Διακριτική ικανότητα: Η μικρότερη αλλαγή τιμής εισόδου που μπορεί να ανιχνεύσει.
- Ευαισθησία: Η σχέση της αλλαγής εξόδου προς τη αλλαγή εισόδου, είναι ίση με τη διαφορά των τιμών της εξόδου προς τη διαφορά των αντίστοιχων τιμών εισόδου.
- Βαθμονόμηση: Η βαθμολόγηση της κλίμακας σε μονάδες.
- Νεκρή ζώνη: Το μέγιστο ποσό αλλαγής της εισόδου που δεν επιφέρει αλλαγή στην έξοδο.
- Γραμμικότητα: Ο βαθμός στον οποίο η γραφική παράσταση της εξόδου προσεγγίζει ευθεία ως προς την είσοδο του αισθητήρα.
- Απόκριση: Ο χρόνος που απαιτείται για να λάβει τη τελική τιμή η έξοδος.
- Καθυστερήση: Η καθυστέρηση της αλλαγής της εξόδου ως προς την είσοδο.

- Ευστάθεια: Η μεταβολή της εξόδου σε μεγάλη χρονική περίοδο, χωρίς μεταβολή της εισόδου και των συνθηκών.
- Υστέρηση: Η διαφορά στην έξοδο όταν η κατεύθυνση της μεταβολής της εισόδου αντιστραφεί.
- Επαναληψιμότητα: Η παραγωγή του ίδιου αποτελέσματος, σε διαφορετικές χρονικές στιγμές, με την ίδια είσοδο.
- Ολίσθηση: Η μεταβολή των χαρακτηριστικών του αισθητήρα με το χρόνο και το περιβάλλον.
- Στατικό σφάλμα: Σταθερό σφάλμα σε όλο το εύρος λειτουργίας, το οποίο μπορεί να αντισταθμιστεί.
- Χρόνος λειτουργίας: Ο εκτιμώμενος χρόνος λειτουργίας στα πλαίσια των προδιαγραφών του.

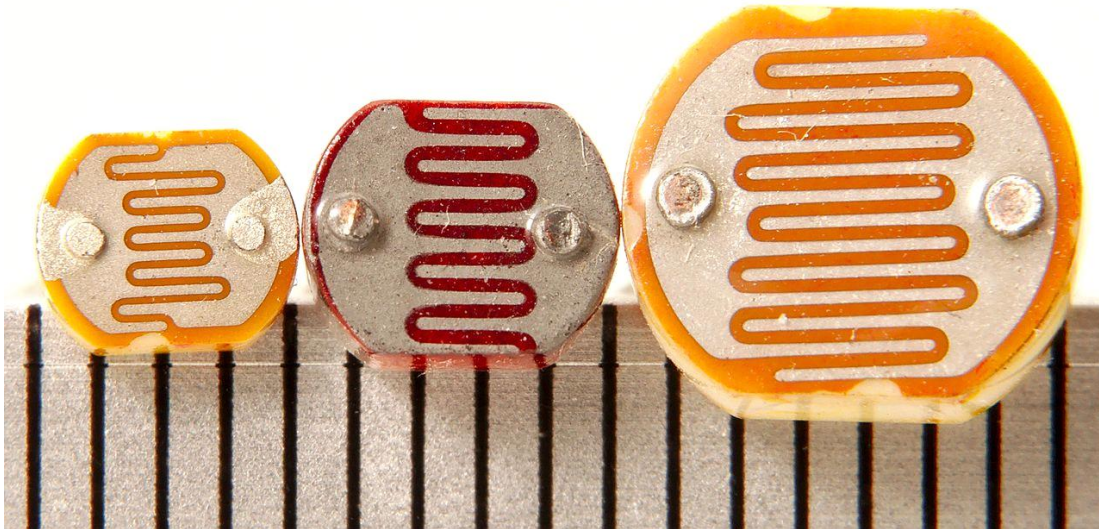
Αισθητήριο φωτός(photoresistor)



Η φωτοαντίσταση είναι μια μεταβλητή αντίσταση, η τιμή της οποίας αλλάζει ανάλογα με το φως που πέφτει πάνω σε αυτή. Τη φωτοαντίσταση τη συναντάμε σε κυκλώματα τα οποία θέλουν με κάποιο τρόπο να αναγνωρίσουν αν το περιβάλλον είναι φωτεινό ή σκοτεινό. Για παράδειγμα λάμπες που ανάβουν αυτόματα σε χαμηλό φωτισμό. Μια φωτοαντίσταση είναι κατασκευασμένη από έναν ημιαγωγό υψηλής αντίστασης. Στο σκοτάδι η φωτοαντίσταση μπορεί να έχει αντίσταση υψηλή έως αρκετά ΜΩ (megaohms), ενώ στο φως, μπορεί να έχει αντίσταση τόσο χαμηλή έως και μερικά εκατοντάδες Ω (ohms). Εάν το φως στην φωτοαντίσταση ξεπεράσει μια ορισμένη συχνότητα, τα φωτόνια, απορροφούνται από τον ημιαγωγό δίνοντας στα δεσμευμένα ηλεκτρόνια, αρκετή ενέργεια για να πηδήξουν μέσα στην ζώνη αγωγιμότητας. Έχει σαν αποτέλεσμα τα ελεύθερα ηλεκτρόνια να μεταφέρουν ηλεκτρισμό και κατ'αυτόν τον

τρόπο να χαμηλώσουν την αντίσταση. Τα εύρος της αντίστασης και της ευαισθησίας μιας φωτοαντίστασης, διαφέρουν αρκετά μεταξύ των διαφορετικών συσκευών. Επιπλέον, χαρακτηριστικοί τύποι φωτοαντιστάσεων μπορεί να αντιδράσουν αρκετά διαφορετικά στα φωτόνια μέσα σε συγκεκριμένου μήκους κύματος ζώνες. Μια φωτοηλεκτρική συσκευή μπορεί να είναι είτε εγγενής είτε εξωγενής. Ένας εγγενής ημιαγωγός έχει τους δικούς του φορείς φόρτισης και δεν είναι ένας αποδοτικός ημιαγωγός. Για παράδειγμα, η σιλικόνη. Στις εγγενής συσκευές, τα μόνα διαθέσιμα ηλεκτρόνια βρίσκονται στην ζώνη σθένους, και συνεπώς τα φωτόνια πρέπει να έχουν αρκετή ενέργεια για να διεγείρουν τα ηλεκτρόνια μέσα σε όλο το κενό ζώνης. Οι εξωγενείς συσκευές έχουν "ακαθαρσίες", οι οποίες επίσης λέγονται προσθετικά, που έχουν προστεθεί, των οποίων η ενέργεια εδάφους είναι πιο κοντά στην ζώνη μεταφοράς, καθώς τα ηλεκτρόνια δεν μπορούν να πηδήξουν πολύ μακριά. Χαμηλότερης ενέργειας φωτόνια (που έχουν μεγαλύτερο μήκος κύματος και χαμηλότερες συχνότητες) είναι ικανά να ενεργοποιήσουν την συσκευή. Εάν σε ένα δείγμα σαν την σιλικόνη, τα άτομα του έχουν αντικατασταθεί με άτομα φωσφόρου, θα υπάρξουν επιπλέον ηλεκτρόνια διαθέσιμα για αγωγή. Αυτό είναι ένα παράδειγμα ενός εξωγενή ημιαγωγού.

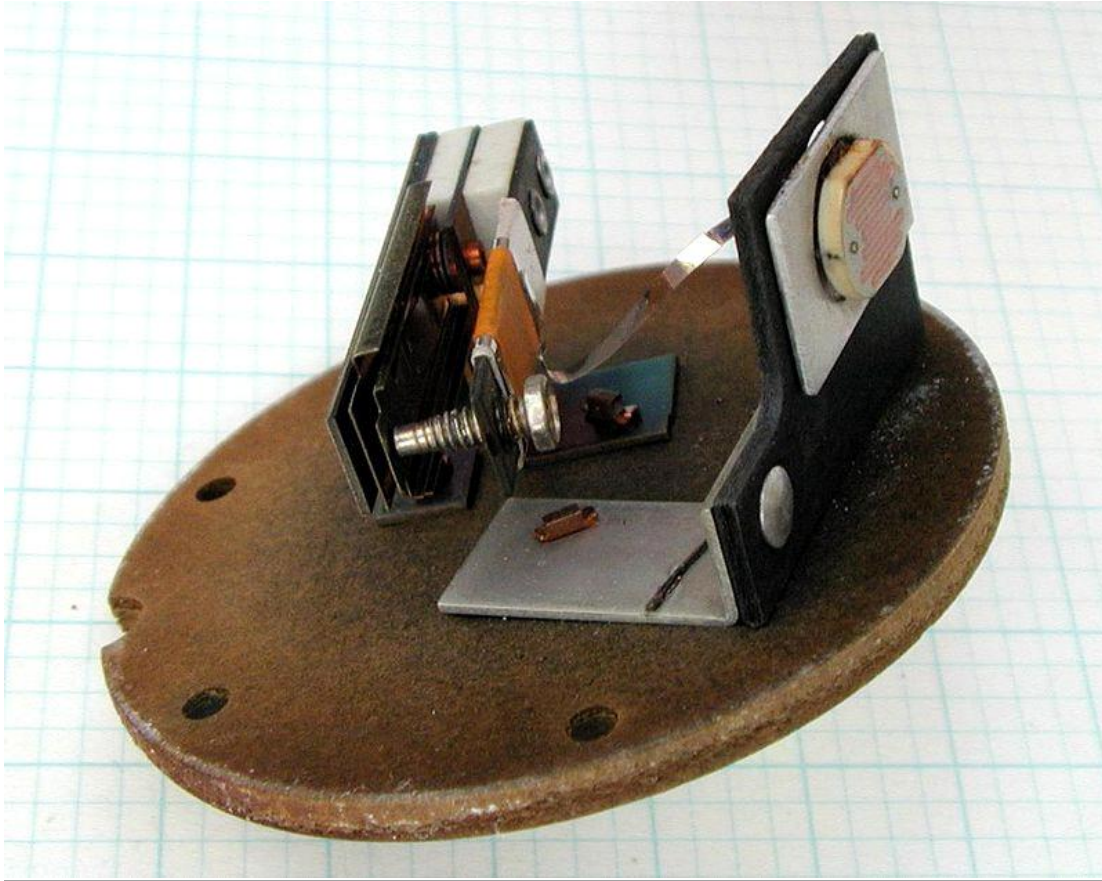
Δομή





Οι φωτοαντιστάσεις είναι συσκευές που έχουν μικρότερη ευαισθησία στην φωτεινότητα σε σχέση με της φωτοδιόδους ή τα φωτοτρανζίστορ. Τα δύο τελευταία εξαρτήματα είναι πραγματικές ημιαγώγιμες συσκευές, ενώ οι φωτοαντιστάσεις είναι παθητικά εξαρτήματα και δεν έχουν PN-junction. Η ευαισθησία φωτός σε κάθε φωτοαντίσταση μπορεί να ποικίλει, εξαρτώμενη στην περιβαλλοντική θερμοκρασία, κανοντάς τα ακατάλληλα για εφαρμογές που απαιτούν ακριβής μέτρηση ή ευαισθησία στα πρωτόνια φωτός. Επίσης, οι φωτοαντιστάσεις, σε έναν βαθμό παρουσιάζουν καθυστέρηση μεταφοράς δεδομένων μεταξύ της έκθεσης στο φως και την επακόλουθη μείωση στην αντίσταση, συνήθως γύρω στα 10 msec(millisecond). Ο χρόνος καθυστέρησης όταν αλλάζει από φωτεινό σε σκοτεινό περιβάλλον είναι ακόμα πιο μεγάλο και συχνά φτάνει και μέχρι και το ένα δευτερόλεπτο. Αυτό τα κάνει ακατάλληλα για την ανίχνευση ταχέων αναλαμπών φωτός, αλλά μερικές φορές χρησιμοποιείται για την εξομάλυνση της απόκρισης της συμπίεσης του ήχου.

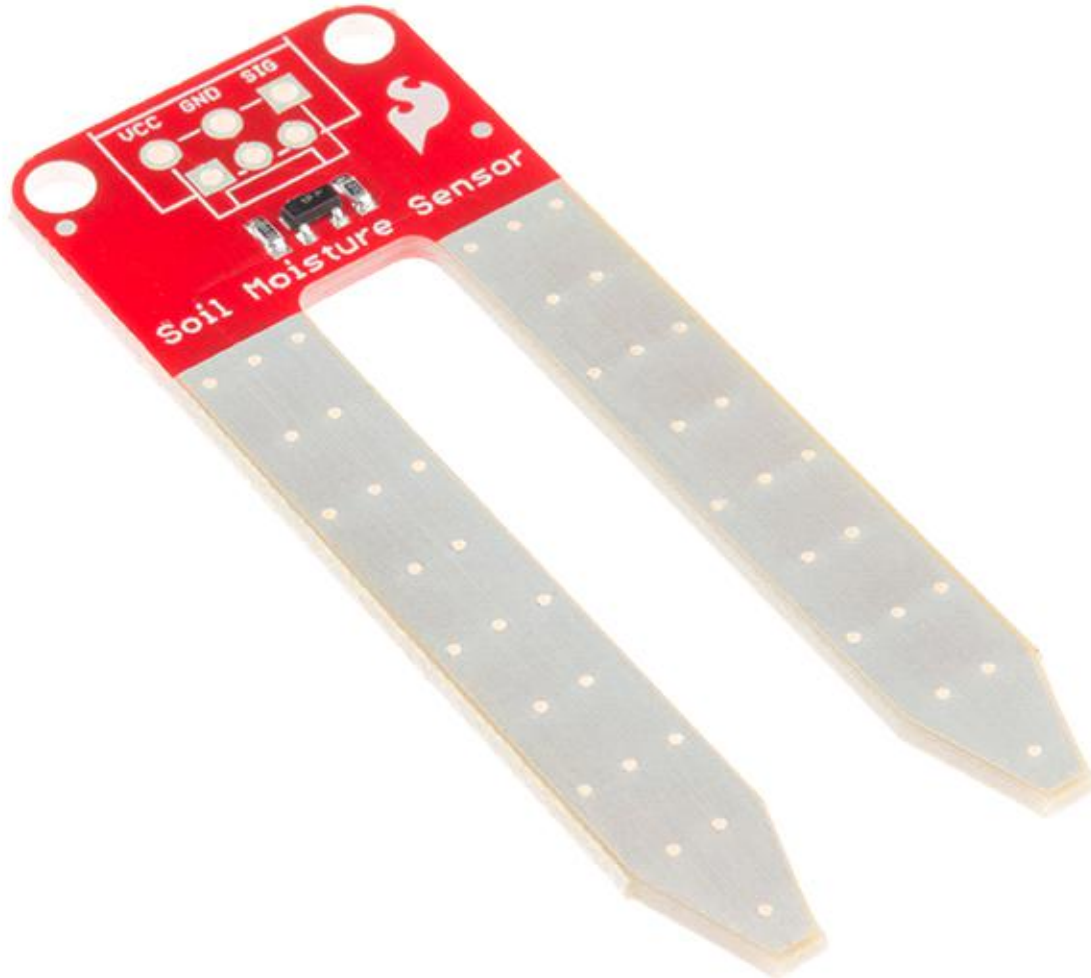
Εφαρμογες



Εικόνα(πανω): τα εσωτερικά εξαρτήματα ενός φωτοηλεκτρικού ελέγχου για ένα τυπικό αμερικάνικο φωτιστικό δρόμου.

Οι φωτοαντιστάσεις διακρίνονται σε πολλούς τύπους. Φτωχότερης ποιότητας κομμάτια σουλφιδίου του καδμίου βρίσκονται σε πολλά προϊόντα όπως κάμερες, μετρητές φωτός, ραδιόφωνα ρολογιού, ξυπνητήρια(σαν τον ανιχνευτή αχτίδας φωτός), φώτα νυκτός, εξωτερικά ρολόγια, ηλιακά φωτιστικά δρόμου κλπ. Οι φωτοαντιστάσεις μπορούν να τοποθετηθούν μέσα σε φωτιστικά δρόμου, για να ελεγχθεί πότε θα ανάβουν τα φώτα. Φώτα του περιβάλλοντος πέφτουν πάνω στον φωτοαισθητήρα προκαλώντας το φωτιστικό δρόμου να σβήσει. Επομένως, συσσωρεύεται ενέργεια εξασφαλίζοντας ότι τα φώτα θα είναι ανοιχτά τις νυχτερινές ώρες. Επίσης χρησιμοποιούνται κάποιοι δυναμικοί συμπιεστές μαζί με μια μικρή λυχνία πυράκτωσης ή μια λάμπα νεον, ή μια δίοδο που εκπέμπει φως για να ελέγξει την μείωση του κέρδους. Η χρήση των CdS και CdSe φωτοαντιστάσεων είναι απαγορευμένες στην Ευρώπη λόγω της απαγόρευσης του RoHS στο κάδμιο. Το σουλφίδιο μόλυβδου και το ανδμονιούχο ίνδιο χρησιμοποιούνται για τη φασματική περιοχή μέσης υπέρυθρης ακτινοβολίας.

Αισθητήριο υγρασίας εδάφους-Sparkfan soil moisture sensor



Η σύνδεση του αισθητηρίου είναι πολύ απλή. Υπάρχουν μόνο 3 pin για να συνδεθούν: VCC, GND, SIG. Πρέπει να υπάρξει παροχή τάσης στο VCC.. Το SIG προσφέρει ένα αναλογικό σήμα το οποίο μπορεί να συνδεθεί με το ADC του μικροεπεξεργαστή. Η μέτρηση που θα διαβαστεί από το SIG εξαρτάται από την τάση με την οποία θα τροφοδοτηθεί ο αισθητήρας. Οι δύο καθετήρες λειτουργούν σαν μεταβλητές αντιστάσεις. Περισσότερο νερό στο χώμα σημαίνει καλύτερη αγωγιμότητα και αποτελέσματα σε μια χαμηλότερη αντίσταση και ένα υψηλότερο SIG out. Οι αναλογικές ενδείξεις θα ποικίλουν ανάλογα με τη τάση χρησιμοποιείται για το Vcc όπως και για την ανάλυση των ADC pins. Συνιστάται η τροφοδοσία του αισθητηρίου να είναι μεταξύ 3.3V-5V.

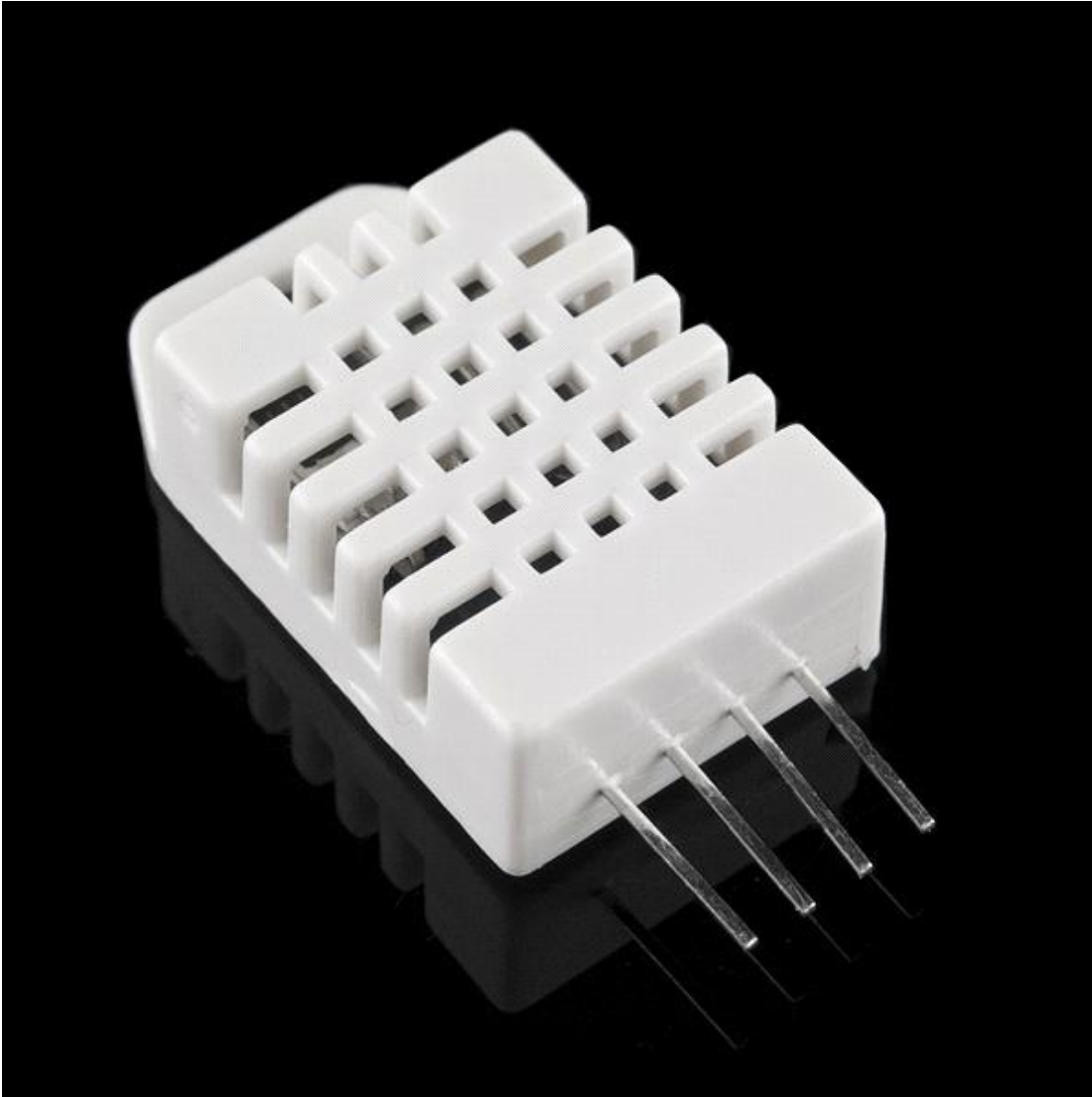
Το αισθητήριο προσφέρει 2 εισόδους, μια ψηφιακή. Το ψηφιακό όριο μπορεί να μεταβληθεί με ένα ποτενσιόμετρο. Οπότε, αν χρησιμοποιηθεί η ψηφιακή έξοδος, τότε θα υπάρχουν μόνο 2 επίπεδα υγρασίας. Υπάρχει ένα μειονέκτημα όσον αφορά την χρήση της ψηφιακής εξόδου. Όποτε η μηχανή θα ξεκινά, και το επίπεδο υγρασίας θα ανεβαίνει κοντά στον αισθητήρα και περνάει το όριο, η αντλία θα απενεργοποιείται. Συνεπώς, το πότισμα θα πραγματοποιείται για ένα μικρό χρονικό διάστημα και το νερό δεν θα φτάσει βαθιά στις ρίζες.

Άλλη μια επιλογή, είναι η χρήση της αναλογικής εξόδου. Το αισθητήριο έχει όρια μεταξύ 0-1023, όπου το 0 είναι για υγρό χώμα και το 1023 είναι για ξερό χώμα. Μπορούμε να θέσουμε 2 όρια, ένα για να αρχίσει το πότισμα και ένα για να σταματήσει (σε αυτό το project θα χρησιμοποιηθεί η αναλογική έξοδος το αισθητήρα υγρασίας χώματος).

Ένα ευρέως γνωστό πρόβλημα με τον αισθητήρα υγρασίας χώματος είναι ο χαμηλός χρόνος ζωής του όταν εκτίθεται σε ένα περιβάλλον με υγρασία. Για να αντιμετωπιστεί αυτό, τα PCB έχουν καλυφθεί με χρυσό τελείωμα(ηλεκτρολυτικό χρυσό με χύτευση νικελίου).

Ακόμα ένας τρόπος επέκτασης του χρόνου ζωής του αισθητήρα είναι να τροφοδοτηθεί μόνο όταν θα λαμβάνεται ένδειξη.

Αισθητήριο θερμοκρασίας-υγρασίας-Humidity and Temperature Sensor DHT22



1. Χαρακτηριστικά και εφαρμογές:

- Υψηλή ακρίβεια
- Χωρητικού τύπου
- Πλήρης αντιστάθμιση θερμοκρασίας
- Μέτρηση υγρασίας και θερμοκρασίας
- Βαθμονομημένο ψηφιακό σήμα
- Εξαιρετική μακροπρόθεσμη σταθερότητα
- Επιπλέον υλικά είναι αχρείαστα

- Μεγάλη απόσταση μετάδοσης σήματος έως 100 μέτρα
- Κατανάλωση μικρής ισχύος
- 4 pin συσκευασμένα

2. Περιγραφή:

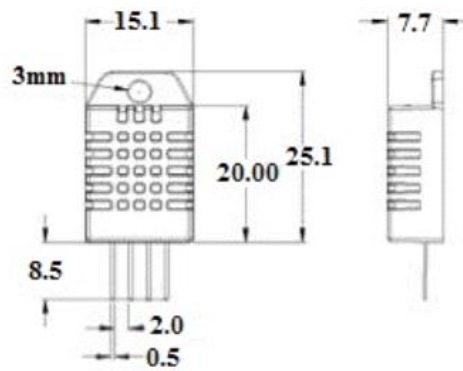
Η έξοδος του RHT03 είναι βαθμονομημένου ψηφιακού σήματος. Χρησιμοποιεί αποκλειστικά τεχνική συλλογής ψηφιακού σήματος και τεχνολογία ανίχνευσης υγρασίας, διαβεβαιώνοντας την σταθεροτητά και την αξιοπιστία του. Τα στοιχεία τα οποία ανιχνεύουν είναι συνδεδεμένα με ένα 8-bit τσιπ υπολογιστή.

Το μικρό μέγεθος, η χαμηλή κατανάλωση και η μεγάλη απόσταση μεταφοράς (100m) δίνει την δυνατότητα στο RHT03 να ταιριάζει με όλα τα είδη των δύσκολων περιπτώσεων των εφαρμογών. Μιας σειράς κατασκευασμένο με 4 pin, καθιστώντας την σύνδεση πολυ βολική.

3. Τεχνικές λεπτομέρειες:

- Μοντέλο --> RHT03
- Τροφοδοσία --> ισχύος 3.3-6V DC
- Σήμα εξόδου --> ψηφιακό σήμα μέσω MaxDetect 1-wire bus
- Αισθητήριο--> Πολυμερές πυκνωτής υγρασίας
- Εύρος λειτουργίας --> υγρασία 0-100%RH; θερμοκρασία -40-80 κελσίου
- Ακρίβεια --> υγρασία +-2%RH(Μέγιστο +-5%RH); θερμοκρασία +-0.5 κελσίου
- Ανάλυση ή ευαισθησία --> υγρασία 0.1%RH; θερμοκρασία 0.1 κελσίου
- Επαναληψιμότητα --> υγρασία +-1%RH; θερμοκρασία +-0.2 κελσίου
- Υστέρηση θερμοκρασίας--> +-0.3%RH
- Μακροπρόθεσμη σταθερότητα--> +-0.5%RH/χρόνο

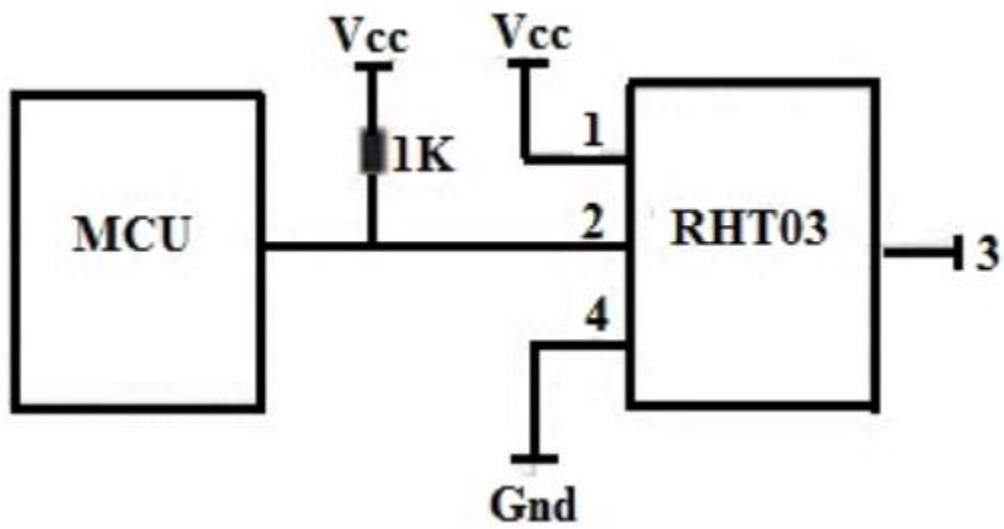
4. Διαστάσεις(μονάδα mm):



Pin sequence number: 1 2 3 4 (from left to right direction).

Pin	Function
1	VDD—power supply
2	DATA—signal
3	NULL
4	GND

5. Ηλεκτρικό διάγραμμα σύνδεσης:



Προσοχή:

Εάν το σήμα από το RHT03 έχει πάντα τάση σε υψηλό επίπεδο, σημαίνει ότι το RHT03 δυσλειτουργεί πρέπει να ελεγχθεί η ηλεκτρική σύνδεση.

Άλλα αισθητήρια που μπορούν να συνδιαστούν με το arduino είναι:

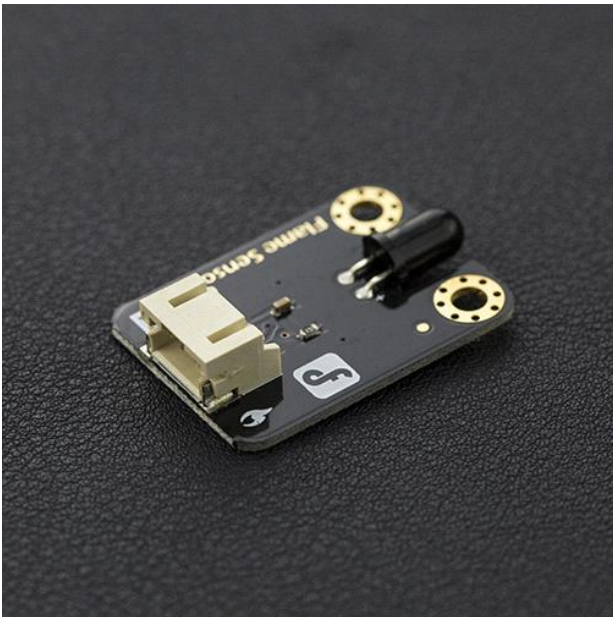
1) Αισθητήρας δαχτυλικού αποτυπώματος



2) Αισθητήρας σύγκρουσης



3) Αισθητήρας φωτιάς



4) Αισθητήρας ήχου



4. Arduino

4.1 Γενικά για το arduino

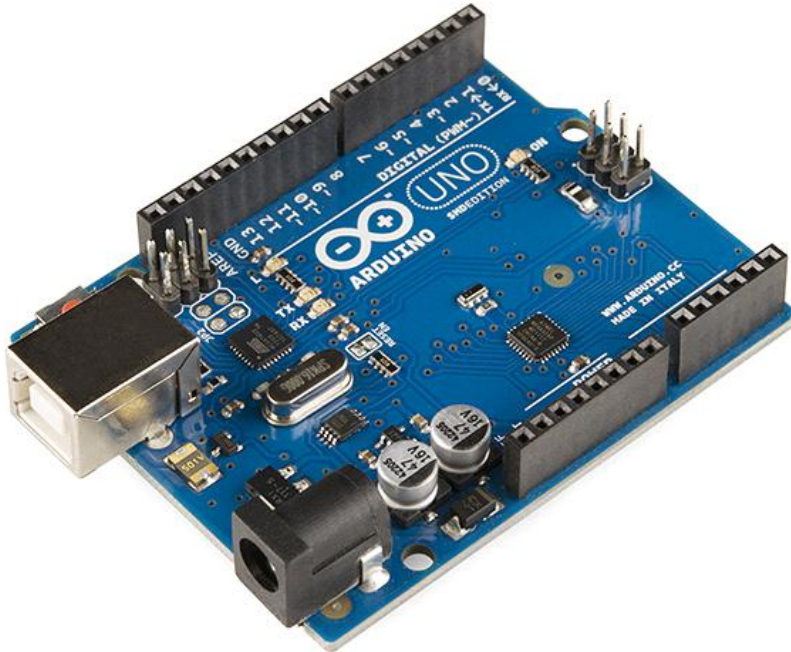
Γενικά

Το Arduino είναι ένας μικροελεγκτής μονής πλακέτας, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, 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++). Οι περισσότερες εκδόσεις του Arduino μπορούν να αγοραστούν προ-συναρμολογημένες. Το διάγραμμα και οι πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους.

Ιστορικό

Το 2005 ένα σχέδιο ξεκίνησε προκειμένου να φτιαχτεί μία συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν πιο φθηνή από άλλα πρωτότυπα συστήματα διαθέσιμα εκείνη την περίοδο. Οι ιδρυτές Massimo Banzι και David Cueartielles ονόμασαν το σχέδιο από τον Arduino της Inrea και ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Inrea, κωμόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας - την ίδια περιοχή στην οποία στεγαζόταν η εταιρία υπολογιστών Olivetti. Το σχέδιο Arduino είναι μία διακλάδωση της πλατφόρμας Wiring για λογισμικό ανοικτού κώδικα και προγραμματίζεται χρησιμοποιώντας μια γλώσσα βασισμένη στο Wiring (σύνταξη και βιβλιοθήκες), παρόμοια με την C++ με απλοποιήσεις και αλλαγές, καθώς και ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Arduino IDE).

Η πλακέτα του Arduino UNO Rev3



Πλατφόρμα

Μία πλακέτα Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR (ATmega328 και ATmega168 στις νεότερες εκδόσεις, ATmega8 στις παλαιότερες) και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωσή του σε άλλα κυκλώματα. Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz. Ο μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα bootloader, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής. Σε εννοιολογικό επίπεδο, στην χρήση του Arduino software stack, όλα τα boards προγραμματίζονται με μία RS-232 σειριακή σύνδεση, αλλά ο τρόπος που επιτυγχάνεται αυτό διαφέρει σε κάθε hardware εκδοχή. Τα σημερινά Arduino προγραμματίζονται μέσω USB και αυτό καθίσταται δυνατό μέσω της εφαρμογής προσαρμογών chip USB-to-Serial όπως το FTDI FT232. Κάποιες παραλλαγές, όπως το Arduino mini και το ανεπίσημο Boarduino, χρησιμοποιούν ένα αφαιρούμενο USB-to-Serial καλώδιο ή board, Bluetooth ή άλλες μεθόδους. (Όταν χρησιμοποιείται με παραδοσιακά εργαλεία microcontroller αντί για το Arduino IDE, χρησιμοποιείται πρότυπος προγραμματισμός AVR ISP).

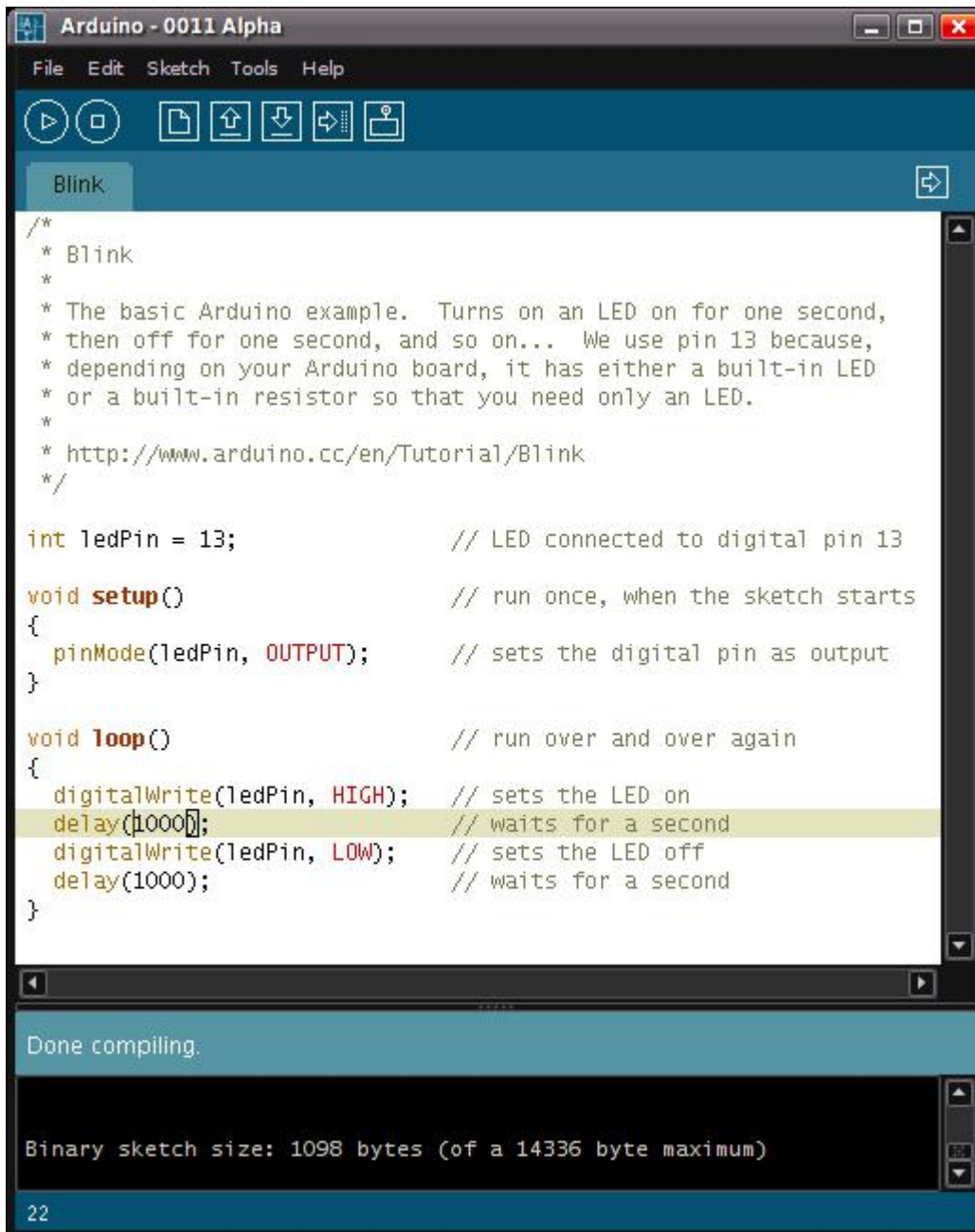
Ο πίνακας Arduino εκθέτει τα περισσότερα I/O(input/output) pin μικροελεγκτή για χρήση, σε σχέση με άλλα κυκλώματα. Τα Diecimila, Duemilanove και το τρέχον Uno παρέχουν 14 ψηφιακά I/O(input/output) pin, έξι από τα οποία μπορούν να παράγουν pulse-width διαμορφωμένα σήματα, και έξι αναλογικά δεδομένα. Διάφορες εφαρμογές ασπίδων plug-in είναι εμπορικά διαθέσιμες. Το Arduino nano και τα Arduino-Compatible Bare Bones Board και Boarduino Board ενδέχεται να παρέχουν male header pins στο κάτω μέρος του board προκειμένου να συνδέονται σε Breadboards. Υπάρχουν πολλά boards συμβατά με και προερχόμενα από Arduino boards. Κάποια είναι λειτουργικά ισάξια με ένα Arduino και μπορεί να χρησιμοποιηθούν εναλλακτικά. Πολλοί χρησιμοποιούν το βασικό Arduino με την προσθήκη καινοτόμων output drivers, συχνά για την χρήση σχολικής μόρφωσης για να απλοποιήσουν την κατασκευή buggies και μικρών robot. Άλλες είναι ηλεκτρικά ισάξια αλλά αλλάζουν τον παράγοντα μορφής, επιτρέποντας κάποιες φορές την συνεχόμενη χρήση των Shields ενώ κάποιες όχι. Κάποιες παραλλαγές έχουν τελείως διαφορετικούς επεξεργαστές, με ποικίλα επίπεδα συμβατότητας.

Λογισμικό

Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java, που λειτουργεί σε πολλές πλατφόρμες και προέρχεται από το IDE για τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring. Έχει σχεδιαστεί για να εισαγάγει στον προγραμματισμό τους καλλιτέχνες και τους νέους που δεν είναι εξοικειωμένοι με την ανάπτυξη λογισμικού. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης και ο συνδυασμός αγκυλών και είναι επίσης σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα με ένα μόνο πάτημα. Δεν υπάρχει συνήθως καμία ανάγκη να επεξεργαστούν αρχεία make ή να τρέξουν προγράμματα σε ένα περιβάλλον γραμμής εντολών. Ένα πρόγραμμα ή κώδικας που γράφτηκε για Arduino ονομάζεται σκίτσο (sketch). Τα Arduino προγράμματα είναι γραμμένα σε C ή C++. Το Arduino IDE έρχεται με μια βιβλιοθήκη λογισμικού που ονομάζεται "Wiring", από το πρωτότυπο σχέδιο Wiring, γεγονός που καθιστά πολλές κοινές λειτουργίες εισόδου/εξόδου πολύ πιο εύκολες. Οι χρήστες πρέπει μόνο να ορίσουν δύο λειτουργίες για να κάνουν ένα πρόγραμμα κυκλικής εκτέλεσης:

- `setup()`:μία συνάρτηση που τρέχει μία φορά στην αρχή του προγράμματος η οποία αρχικοποιεί τις ρυθμίσεις
- `loop()`:μία συνάρτηση που καλείται συνέχεια μέχρι η πλακέτα να απενεργοποιηθεί

Παράδειγμα προγράμματος γραμμένο σε C



The image shows a screenshot of the Arduino IDE interface. The window title is "Arduino - 0011 Alpha". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for running, stopping, saving, and other functions. The main editor area displays the "Blink" sketch code. The code includes a multi-line comment explaining the sketch, followed by the `int` declaration for the LED pin, the `void setup()` function, and the `void loop()` function. The `delay(1000);` lines in the `loop()` function are highlighted in yellow. Below the editor, a status bar shows "Done compiling." and "Binary sketch size: 1098 bytes (of a 14336 byte maximum)". The page number "22" is visible in the bottom left corner.

```
Arduino - 0011 Alpha
File Edit Sketch Tools Help

Blink

/**
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;           // LED connected to digital pin 13

void setup()              // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()               // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);              // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);              // waits for a second
}

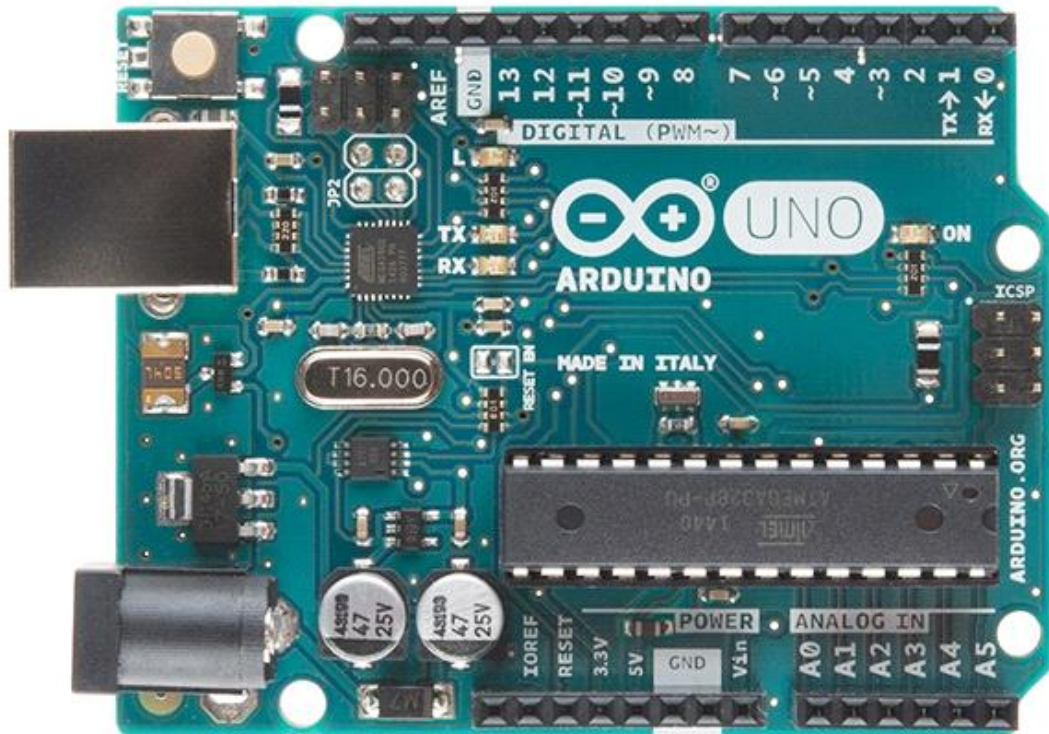
Done compiling.

Binary sketch size: 1098 bytes (of a 14336 byte maximum)

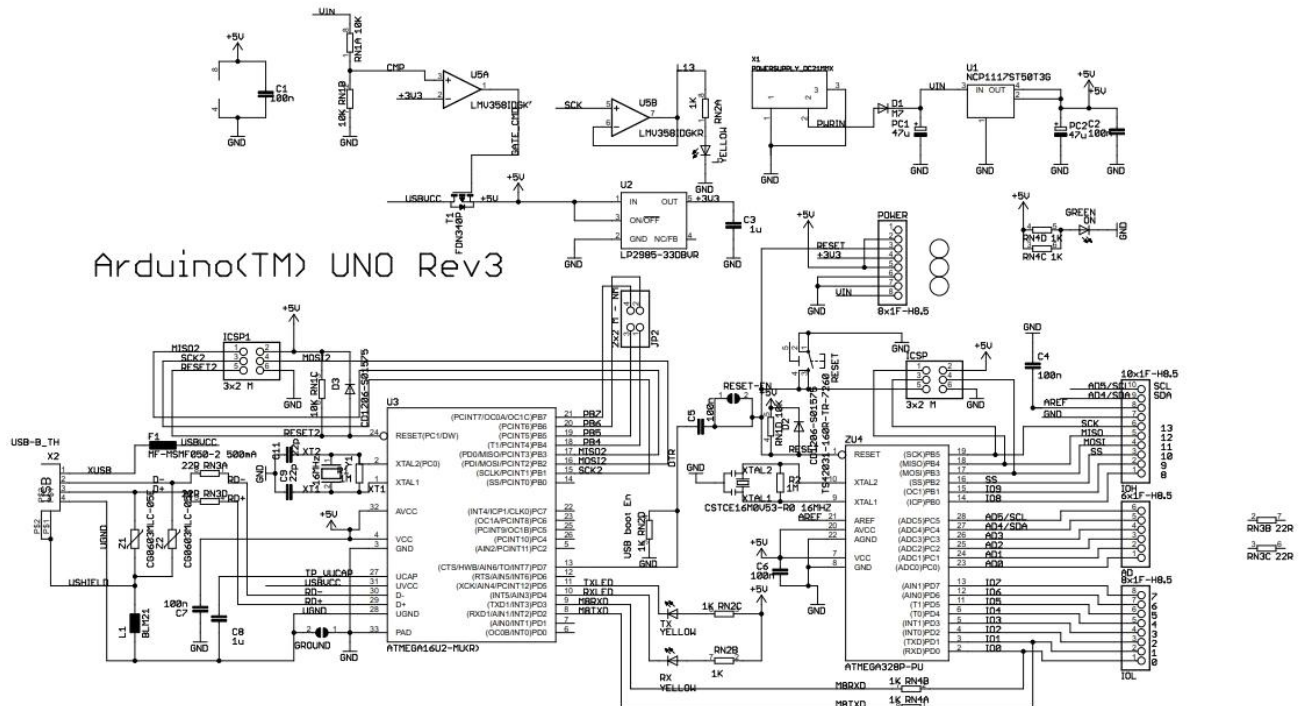
22
```

4.2 Arduino Uno rev3

Arduino Uno rev3



Διαγραμμα



Το Arduino uno r3 είναι είναι ένας μικροελεγκτής βασισμένος στον ATmega328P. Έχει 14 ψηφιακές εισόδους/εξόδους pins (από τις οποίες οι 6 μπορούν να χρησιμοποιηθούν σαν PWM έξοδοι), 6 αναλογικές εισόδους, έναν 16 MHz quartz κρύσταλλο, μια θύρα USB για σύνδεση, ένα καλώδιο σύνδεσης, ένα ICSP header και ένα κουμπί reset. Περιέχει ότι χρειάζεται για τα υποστηρίξει τον μικροελεγκτή. Απλά πρέπει να συνδεθεί σε έναν υπολογιστή με ένα USB καλώδιο ή να τροφοδοτηθεί με ένα AC-to-DC αντάπτορα ή μπαταρία. Ο χρήστης μπορεί να το χρησιμοποιήσει χωρίς να ανησυχεί μήπως γίνει κάποιο ζημιά. Το χειρότερο σενάριο είναι να καεί το τσιπ και μπορεί να αντικατασταθεί για ένα μικρό χρηματικό ποσό.

Τεχνικά χαρακτηριστικά

1. Μικροελεγκτής	ATmega328P
2. Τάση λειτουργίας	5V
3. Τάση εισόδου (συνιστάται)	7-12V
4. Τάση εισόδου (όριο)	6-20V
5. Ψηφιακά I/O Pins	14(Οι 6 είναι PMW έξοδοι)
6. ψηφιακά I/O Pins	6
7. Αναλογικά Pins εισόδου	6

DC ρεύμα ανα I/O Pin	20 mA
DC ρεύμα για το 3.3V Pin	50 mA
Μνήμη Flash	32 KB (ATmega328P)
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Ταχύτητα ρολογιού	16 MHz
LED_BUILTIN	13
Μήκος	68.6 mm
πλάτος	53.4 mm
Βάρος	25 g

Προγραμματισμός

Το Arduino Uno μπορεί να προγραμματιστεί με το λογισμικό Arduino (IDE). Γίνεται επιλογή "Arduino/Genuino Uno από την επιλογή Tools > Board menu (σύμφωνα με τον μικροεπεξεργαστή). Ο ATmega328 στο Arduino Uno έρχεται προγραμματισμένο με έναν bootloader ο οποίος επιτρέπει την αναφόρτωση του νέου κώδικα σε αυτό χωρίς την παρέμβαση εξωτερικού προγραμματιστή. Επικοινωνεί χρησιμοποιώντας πρωτόκολλο STK500.

Τροφοδοσία

Η πλακέτα Arduino Uno μπορεί να τροφοδοτηθεί μέσω σύνδεσης USB ή με εξωτερική τροφοδοσία. Η πηγή τροφοδοσία επιλέγεται αυτόματα. Εξωτερική τροφοδοσία μπορεί να γίνει είτε από αντάπτορα απο AC-DC είτε από μπαταρία. Ο αντάπτορας μπορεί να συνδεθεί τοποθετώντας ένα καλώδιο 2.1mm στην θύρα τροφοδοσίας της πλακέτας. Οι ακροδέκτες από την μπαταρία μπορούν να τοποθετηθούν στην γείωση(GND) και στο Vin pin header του τροφοδοτικού. Η πλακέτα μπορεί να λειτουργήσει με μια εξωτερική τροφοδοσία από 7 μέχρι 12 volts. Εάν τροφοδοτηθεί με λιγότερο από 7V, το pin των 5V μπορεί να την τροφοδοτήσει και ίσως με λιγότερο απο 5 volts και η πλακέτα μπορεί να

γίνει ασταθής. Εάν χρησιμοποιηθούν πάνω από 12 V, τότε ο ρυθμιστής τάσης μπορεί να υπερθερμανθεί και να κάνει ζημιά στην πλακέτα. Τα συνιστάμενα όρια είναι 7 με 12.

- Vin. Η τάση εισόδου του Arduino/Genuino χρησιμοποιεί εξωτερική πηγή τροφοδοσίας(σε σχέση με τα 5 volts από την σύνδεση USB ή κάποια άλλη ρυθμιζόμενη πηγή ενέργειας). Μπορεί εάν δωθεί τροφοδοσία μέσω αυτού του pin, ή εάν τροφοδοτηθεί χρησιμοποιώντας βίσμα, να δωθεί πρόσβαση μέσω αυτού του Pin.
- 5V. Αυτό το pin εξάγει 5V ρυθμισμένα από τον ρυθμιστή στην πλακέτα. Η πλακέτα μπορεί να τροφοδοτηθεί με ενέργεια είτε από την DC υποδοχή τροφοδοσίας (7-12V), από την υποδοχή USB (5V), ή από το VIN pin στην πλακέτα (7-12V). Τροφοδοτώντας με τάση μέσω των 5V ή 3.3V pins παρακάμπτει τον ρυθμιστή, και μπορεί να κάνει ζημιά στην πλακέτα.
- 3.3. Η 3 volt τροφοδοσία παράγεται από τον ρυθμιστή της πλακέτας. Η μέγιστη λήψη ρεύματος είναι 50 mA.
- GND. Ακροδέκτες γείωσης
- IOREF. Αυτό το pin στην πλακέτα Arduino/Genuino παρέχει την τάση με την οποία λειτουργεί ο μικροελεγκτής. Μια σωστά ρυθμισμένη ασπίδα μπορεί να διαβάσει την τάση του IOREF pin και να επιλέξει την κατάλληλη πηγή ενέργειας ή να ενεργοποιήσει τους μετασχηματιστές στις εξόδους για να λειτουργήσει με τα 5V ή τα 3.3V.

Μνήμη

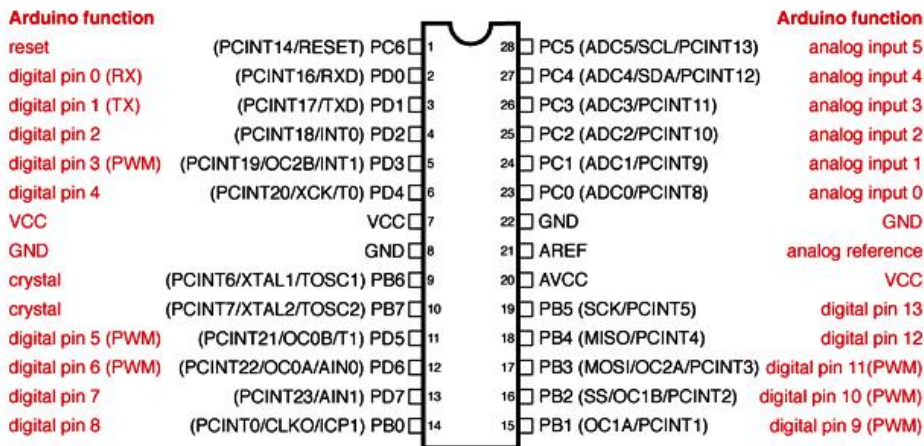
Ο ATmega328 έχει 32 KB (από τα οποία τα 0.5 KB κατειλημένα από το bootloader). Επίσης έχει 2 KB SRAM και 1 KB EEPROM (τα οποία μπορούν να διαβαστούν και να αποθηκευτούν μέσα στην βιβλιοθήκη EEPROM).

Είσοδος και έξοδος

Η χαρτογράφηση μεταξύ των Arduino pins και των θυρών του ATmega328P. Οι χαρτογραφήσεις για τους Atmega8, 168, and 328 είναι πανομοιότυπες.

Χαρτογράφηση pin ATmega328P

Atmega168 Pin Mapping



Digital Pins 11, 12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Το κάθε ένα από τα 14 ψηφιακά pin στο Uno μπορεί να χρησιμοποιηθεί σαν είσοδο και σαν έξοδο, χρησιμοποιώντας τις λειτουργίες `pinMode()` (Για παράδειγμα `pinMode(LED, OUTPUT)`). Λειτουργούν στα 5 volts. Κάθε pin μπορεί να παρέχει ή να λάβει 20 mA σαν την συνισταμένη λειτουργική κατάσταση. Ένα μέγιστο των 40mA είναι η τιμή, η οποία δεν πρέπει να ξεπεραστεί σε κανένα I/O pin για να αποφευχθεί μόνιμη ζημιά στον μικροεπεξεργαστή.

Επιπροσθέτως, μερικά pin έχουν ειδικευμένες λειτουργίες:

- Σειριακή: 0 (RX) και 1 (TX). Χρησιμοποιείται για να λάβει (RX) και να μεταδώσει (TX) TTL σειριακά δεδομένα. Αυτά τα pin είναι συνδεδεμένα στα αντίστοιχα pin του ATmega8U2 USB-to-TTL Serial chip.
- PWM: 3, 5, 6, 9, 10, και 11. Παρέχει 8-bit PWM έξοδο με την `analogWrite()` λειτουργία.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Αυτά τα pin υποστηρίζουν SPI επικοινωνία χρησιμοποιώντας την βιβλιοθήκη SPI.
- LED: 13. Υπάρχει ένα ενσωματωμένο LED το οποίο οδηγείται από το ψηφιακό pin 13.
- TWI: A4 ή SDA pin και A5 ή SCL pin. Υποστηρίζει TWI επικοινωνία χρησιμοποιώντας την `Wire` βιβλιοθήκη.

Το Uno έχει 6 αναλογικές εισόδους, με την ένδειξη από A0 μέχρι A5, όπου η κάθε μια παρέχει 10 bits ανάλυσης (δηλαδή 1024 διαφορετικές τιμές). Από προεπιλογή μετράνε από το έδαφος στα 5 volts, αν και είναι πιθανό να αλλάξει το ανώτερο άκρο της εμβέλειας, χρησιμοποιώντας ο AREF pin και την `analogReference()` λειτουργία.

Υπάρχουν δύο άλλα pin στην πλακέτα:

- AREF. Πηγή τάσης για τις αναλογικές εισόδους. Χρησιμοποιείται με την `analogReference()`.
- Reset. Πατώντας το κουμπί reset θα γίνει reset στον μικροελεγκτή.

Επικοινωνία

Το Arduino/Genuino Uno διαθέτει πολλές εγκαταστάσεις επικοινωνίας με έναν υπολογιστή, άλλη Arduino/Genuino πλακέτα, ή άλλους μικροεπεξεργαστές. Ο ATmega328 παρέχει UART TTL (5V) σειριακή επικοινωνία, η οποία είναι διαθέσιμη στα ψηφιακά pins 0 και 1. Ένας ATmega16U2 διοχετεύει στην πλακέτα αυτή τη σειριακή επικοινωνία μέσω USB και εμφανίζεται σαν μια εικονική θύρα com στο λογισμικό του υπολογιστή. Το 16U2 υλικό-λογισμικό χρησιμοποιεί τα πρότυπα προγράμματα οδήγησης USB COM, και κανένα εξωτερικό πρόγραμμα οδήγησης δεν χρειάζεται. Ωστόσο, στα windows, ένα αρχείο τύπου .inf είναι αναγκαίο. Το λογισμικό Arduino (IDE) εμπεριέχει μια σειριακή οθόνη η οποία επιτρέπει απλά δεδομένα κειμένου να σταλούν από και στην πλακέτα. Τα RX και TX Led στην πλακέτα θα ανάψουν όταν μεταδοθούν δεδομένα μέσω του USB-σε-σειριακο chip και σύνδεση USB στον υπολογιστή(αλλά όχι για σειριακή σύνδεση στα pin 0 και 1). Μια σειριακή βιβλιοθήκη λογισμικού επιτρέπει σειριακή επικοινωνία σε οποιοδήποτε από τα ψηφιακά pin του UNO. Επιπροσθέτως, ο ATmega328 υποστηρίζει I2C (TWI) και SPI επικοινωνία. Το λογισμικό Arduino (IDE) συμπεριλαμβάνει μια wire βιβλιοθήκη για να απλουστεύσει την χρήση του I2C bus. Για επικοινωνία SPI πρέπει να γίνει χρήση της SPI βιβλιοθήκης.

Αυτόματη (λογισμικό) επαναφοράς

Προκειμένου να μην απαιτηθεί φυσικό πάτημα του κουμπιού επαναφοράς πριν το ανέβασμα, η Arduino/Genuino Uno πλακέτα είναι σχεδιασμένη με τέτοιο τρόπο ώστε να επιτρέπει να γίνεται επαναφορά μέσω λογισμικού το οποίο τρέχει σε έναν συνδεδεμένο υπολογιστή. Μια από τις γραμμές ελέγχου ροής υλικού(DTR) του ATmega8U2/16U2 είναι συνδεδεμένη στην γραμμή επαναφοράς του ATmega328 μέσω ενός πυκνωτή των 100 nanofarad. Όταν αυτή η γραμμή επιβεβαιωθεί(ληφθεί χαμηλή), η γραμμή επαναφοράς πέφτει αρκετά για να επαναφέρει το chip. Το λογισμικό Arduino χρησιμοποιεί αυτή την ιδιότητα για να επιτρέψει την μεταφόρτωση του κώδικα απλά πατώντας το κουμπί μεταφόρτωσης, το οποίο είναι πάνω στην γραμμή εργαλείων. Αυτό σημαίνει ότι το bootloader μπορεί να έχει μικρότερο timeout, καθώς η μείωση του DTR μπορεί να είναι καλά συγχρονισμένη με το ξεκίνημα της μεταφόρτωσης. Αυτή η ρύθμιση έχει άλλες επιπτώσεις. Όταν το Uno είναι συνδεδεμένο είτε σε έναν υπολογιστή που τρέχει Mac OS X ή Linux, επαναφέρεται κάθε φορά που πραγματοποιείται σύνδεση από λογισμικό(από USB). Για το επακόλουθο μισό δευτερόλεπτο το bootloader τρέχει στο Uno. Παρόλο που είναι προγραμματισμένο να αγνοεί δύσμορφα δεδομένα(οτιδήποτε άλλο εκτός από την μεταφόρτωση ενός νέου

κώδικα), αυτό θα διακόψει τα πρώτα bytes δεδομένων τα οποία έχουν σταλθεί στην πλακέτα μετά το άνοιγμα μιας σύνδεσης. Εάν ένα sketch το οποίο τρέχει στην πλακέτα λαμβάνει μιας φοράς ρύθμιση παραμέτρων, ή άλλα δεδομένα όταν ξεκινάει για πρώτη φορά, σιγουρεύει ότι το λογισμικό με το οποίο επικοινωνεί, αναμένει ένα δευτερόλεπτο μετά το άνοιγμα της σύνδεσης και πριν γίνει μεταφορά των δεδομένων. Η πλακέτα Uno περιέχει ένα ίχνος το οποίο μπορεί να κοπεί ούτως ώστε να απενεργοποιηθεί η αυτόματη επαναφορά. Η ενδειξή του είναι "RESET-EN". Επίσης είναι δυνατόν να γίνει απενεργοποίηση της αυτόματης επαναφοράς, συνδέοντας μια αντίσταση των 110 ohm από τα 5V στην γραμμή επαναφοράς.

Η αναθεώρηση του revision 3 στην πλακέτα έχει τα εξής χαρακτηριστικά:

- 1.0 pinout: επιπρόσθετα SDA και SCL pin τα οποία είναι κοντά στο AREF pin και 2 άλλα νέα pin τοποθετημένα κοντά στο pin επαναφοράς, το IOREF το οποίο επιτρέπει στις ασπίδες να προσαρμόζονται στην τάση η οποία παρέχεται από τον πίνακα. Στο μέλλον, οι ασπίδες θα είναι συμβατές με τις πλακέτες οι οποίες χρησιμοποιούν AVR, το οποίο λειτουργεί με 5V και με το Arduino Due το οποίο λειτουργεί με 3.3V. Το δεύτερο δεν είναι συνδεδεμένο pin, το οποίο φυλάγεται για μελλοντικές χρήσεις.
- Ισχυρότερο κύκλωμα επαναφοράς(RESET)
Ο Atmega 16u2 αντικατέστησε τον 8U2

4.3 Arduino Shields

Τα shields είναι ολοκληρωμένες πλακέτες που είναι σχεδιασμένες ώστε να κουμπώνουν πάνω στον Arduino προεκτείνοντας την λειτουργικότητα του.

Μερικοί τύποι Arduino Shield

- nAn0 Und3rP4nTs DC Servo-->Δυνατή Nano ασπίδα για DC σερβοκινητήρα κατάλληλο για CNC και ρομποτική. Χρησιμοποιεί PID και έχει μεγάλη H-Bridge!
- IR Remote Shield 1v09 IR Remote Shield 2v14J--> Είναι εύκολο να χρησιμοποιηθούν οι IR απομακρυσμένες ασπίδες με μικρόφωνο, RGB LED, αισθητήρα φωτός, χειριστήριο, κουμπιά και απενεργοποιημένη επανεκκίνηση.
- grblShield - ελεγκτής 3 αξόνων βηματικού μοτέρ για Arduino-->Λειτουργεί με grbl ή έμφυτο Arduino.
- DDS Shield for Arduino-->Ασπίδα για να φιλοξενήσει ένα φθινό δομοστοιχείο AD9850 DDS προκειμένου να παράξει σταθερά, ελέγχιμα σήματα ράδιοσυχνότητας.
- Board of Education Shield--> Συνέδεσε την μαθητική πλακέτα Shield στο δικό σου Arduino για βολική πρωτοτυπία Breadboard και για θύρες για σερβομηχανισμό, δίχως να χρειαστεί συγκόλληση. Οι προγραμματιστές Arduino, οι οποίοι προτιμούν

τους φιλικούς προς την breadboard αισθητήρες και εξαρτήματα της Parallax, τα οποία τους φαίνονται πιο εύκολα για χρήση με την shield.

Άλλες κάρτες της οικογένειας Arduino

Πλακέτα	Microchip	ΤάσηΕισ.	Ψηφ. Έξοδοι	Αν. Είσοδοι	Μνήμη Flash
Uno	ATmega328	7-12V	14(6 PWM)	6	32KB
Mega	ATmega2560	7-12V	54(15 PWM)	16	256KB
Mini	ATmega328	7-9V	14(6 PWM)	8	32KB
Micro	ATmega32u4	7-12V	20(7 PWM)	12	32KB
DUE	AT91SAM3X8E	7-12V	54(12 PWM)	12	512KB

5. Γλώσσα προγραμματισμού C

5.1 Γενικά για την C

1. Τύποι δεδομένων

Η C έχει πέντε βασικούς τύπους δεδομένων:

char (character),

int (integer),

float (floating point),

double (double floating point),

void (no value).

Όλοι οι άλλοι τύποι της C βασίζονται σ' αυτούς. Όλοι οι βασικοί τύποι εκτός από τον τύπο void μπορεί ν' αλλάξουν γράφοντας πριν από τον τύπο τον κατάλληλο μετασχηματισμό. Οι μετασχηματισμοί αυτοί είναι οι: signed, unsigned, long, και short. Το μέγεθος και τα διαστήματα τιμών των τύπων της C εξαρτάται από τον επεξεργαστή. Στον πίνακα δίνουμε τους τύπους δεδομένων όπως ορίζονται από το πρότυπο ANSI.

Τύπος	μέγεθος σε bits
Char	8
Unsigned char	8
Signed char	8
Int	16
Unsigned int	16
Signed int	16
Short int	16
Unsigned short int	8
Signed short int	8
Long int	32
Signed long int	32
Unsigned long int	32
Float	32
Double	64
Long double	128

Δηλώσεις μεταβλητών

Τα αναγνωριστικά στη C μπορούν να έχουν όσους χαρακτήρες θέλουμε. Αν το αναγνωριστικό είναι εξωτερικό όνομα (όνομα συνάρτησης ή καθολική μεταβλητή) τότε μόνο οι έξι πρώτοι χαρακτήρες είναι σημαντικοί διαφορετικά για εσωτερικά ονόματα οι πρώτοι 31 χαρακτήρες είναι σημαντικοί. Τα κεφαλαία γράμματα στην C είναι διαφορετικά από τα μικρά.

Η δήλωση μιας μεταβλητής έχει την γενική μορφή:

<τύπος> <λίστα μεταβλητών>

```
int i=0,j;
```

```
char q='?';
```

```
short int si;
```

```
float f,g;
```

Δηλώσεις μεταβλητών κάνουμε μέσα σε συναρτήσεις (local variables, automatic), στις παραμέτρους μιας συνάρτησης (formal parameters) και έξω απ' όλες τις συναρτήσεις (global variables). Μπορούμε επίσης να δηλώσουμε μία τοπική μεταβλητή μέσα σε μία ενότητα π.χ.

```
if (συνθήκη) {  
  
char x[30];  
  
...  
  
...  
  
}
```

Στη περίπτωση αυτή η εμβέλεια της μεταβλητής είναι η ενότητα στην οποία είναι δηλωμένη. Έτσι αποφεύγουμε πλάγια αποτελέσματα και έχουμε οικονομία χώρου.

Καθολικές μεταβλητές έχουν εμβέλεια σε ολόκληρο το πρόγραμμα και δηλώνονται στην αρχή του κώδικα έξω απ' όλες τις συναρτήσεις. Όταν μια καθολική και μία τοπική μεταβλητή έχουν το ίδιο όνομα τότε μέσα στην εμβέλεια της τοπικής μεταβλητής αναφερόμαστε πάντα στην τοπική. Μεταβλητές που είναι παράμετροι συναρτήσεων συμπεριφέρονται ως τοπικές μεταβλητές της συνάρτησης.

Σταθερές εισάγονται με την προκαθορισμένη λέξη const π.χ.

```
const int i=1;
```

```
const char q='?';
```

Πτητικές (volatile) μεταβλητές πληροφορούν τον μεταγλωττιστή ότι η τιμή τους μπορεί ν' αλλάξει χωρίς αυτό να δηλώνεται σαφώς στο πρόγραμμα.

Τρόποι αποθήκευσης μεταβλητών

Υπάρχουν τέσσερις τρόποι να πληροφορήσουμε τον μεταγλωττιστή πως ν' αποθηκευτεί μια μεταβλητή. Οι τρόποι αυτοί είναι:

1) extern

Επειδή η C υποστηρίζει ξεχωριστή μεταγλώττιση των διαφόρων ενοτήτων ενός μεγάλου προγράμματος θα πρέπει να υπάρχει κάποιος τρόπος που να πληροφορεί τον μεταγλωττιστή ότι ορισμένες μεταβλητές είναι ορισμένες κάπου αλλού. Υπενθυμίζουμε ότι μια καθολική μεταβλητή μπορεί να δηλωθεί μία μόνο φορά.

Αρχείο 1 Αρχείο 2

```
int x,y; extern int x,y;
```

... ..

2) static

Η δήλωση `static` έχει διαφορετικό αποτέλεσμα πάνω σε τοπικές μεταβλητές και διαφορετικό σε καθολικές μεταβλητές.

Μία `static` τοπική μεταβλητή έχει εμβέλεια μέσα στη συνάρτηση που είναι δηλωμένη και κρατά την τιμή της μεταξύ διαδοχικών καλεσμάτων της συνάρτησης. Έτσι μπορεί να χρησιμοποιηθεί σε μία συνάρτηση παραγωγής μιας σειράς αριθμών π.χ.

```
increment(void)
```

```
{  
static int count=0;  
count=count+5;  
return (count);  
}
```

Η αρχικοποίηση της `count` γίνεται μία μόνο φορά, στο πρώτο κάλεσμα της `increment`.

Μία καθολική μεταβλητή `static` έχει εμβέλεια μόνο στο αρχείο στο οποίο είναι δηλωμένη. Αυτό σημαίνει ότι αν και είναι καθολική δεν μπορούν να την δουν ρουτίνες από άλλα αρχεία και ν' αλλάξουν το περιεχόμενό της.

Οι μεταβλητές `static` μας δίνουν την δυνατότητα να αποκρύψουμε ένα μέρος από ένα πρόγραμμα. Αυτό μας βοηθάει πολύ στο γράψιμο μεγάλων προγραμμάτων καθώς επίσης στο γράψιμο συναρτήσεων που θα μπουν σε βιβλιοθήκες.

3) register

Μία μεταβλητή `register` αποθηκεύεται σ' ένα καταχωρητή της CPU αντί για την μνήμη όπως με τις απλές μεταβλητές (συνεπώς οι μεταβλητές αυτές δεν έχουν διεύθυνση). Αυτό σημαίνει ότι πράξεις με μεταβλητές τύπου `register` είναι πολύ πιο γρήγορες εφόσον δεν απαιτείται προσπέλαση στη μνήμη για να δούμε ή ν' αλλάξουμε την τιμή τους. π.χ.

```
register int temp;
```

4) Δηλώσεις σταθερών

Σταθερές χαρακτήρες γράφονται σε απλά εισαγωγικά.

```
char question_mark = '?';
```


Σταθερές τύπου string γράφονται μέσα σε διπλά εισαγωγικά, "---". Έτσι συνήθως τυπώνουμε διάφορα μηνύματα. Παραδείγματα σταθερών για τους άλλους τύπους δίνονται στον παρακάτω πίνακα.

int 1234

long int 38754L

short int 123

unsigned int 62222

float 12.345F

float 1.1e-3F

double -0.9876544

char '?'

Υπάρχουν όμως ορισμένοι χαρακτήρες που δεν τυπώνονται με μια σταθερά τύπου string για παράδειγμα τα διπλά εισαγωγικά. Για τους χαρακτήρες αυτούς έχουμε τις λεγόμενες σταθερές backslash που δίνονται στον επόμενο πίνακα.

Κώδικας	Σημασία
\b	backspace
\f	form feed
\n	νέα γραμμή
\r	carriage return
\t	οριζόντιο tab
\"	διπλά εισαγωγικά
'	απλά εισαγωγικά
\0	Null
\\	backslash
\v	κάθετο tab
\a	alert
\N	οκταδική σταθερά
\xN	δεκαεξαδική σταθερά

Τελεστές

Η C έχει τέσσερις τύπους τελεστών: αριθμητικοί, σύγκρισης (συσχεσιακοί), λογικοί και τελεστές χειρισμού bits (bitwise operators). Παρακάτω δίνουμε ένα πίνακα με όλους τους τελεστές διατεταγμένους σύμφωνα με την προτεραιότητά τους.

Προτεραιότητα	Τελεστές
Υψηλότερη	<code>() [] -></code> <code>! ~ ++ -- (type) * & sizeof</code> <code>* / %</code> <code>- +</code> <code><< >></code> <code>< <= > >=</code> <code>== !=</code> <code>&</code> <code>^</code> <code> </code> <code>&&</code> <code> </code> <code>?</code> <code>= += -= *= /=</code>
Χαμηλότερη	

Συμβατότητα με εκχωρήσεις

Ο τελεστής εκχώρησης έχει την γενική μορφή

`<αναγνωριστικό> = <παράσταση>`

Κατά την εκχώρηση η τιμή της παράστασης μετατρέπεται στον τύπο της μεταβλητής στο αριστερό μέρος της παράστασης. Για παράδειγμα:

```
int i;
```

```
char ch;
```

```
float f;
```

```
void function(void)
```

```
{
```

```
ch=i; /* ch= ο χαρακτήρας που αντιστοιχεί στο δεύτερο
```

```
byte του i */
```

```
i=f; /* i= το ακέραιο μέρος του f */
```

```
f=ch; /* f= ο αριθμός που αντιστοιχεί στο ένα byte του ch */
```

```
f=i; /* f= ο αριθμός που αντιστοιχεί στα δύο bytes του int */
```

```
}
```

Όταν σε μία παράσταση υπάρχουν τελεσταίοι διαφορετικών τύπων τότε μετασχηματίζονται στον τύπο του "ισχυρότερου" τελεσταίου. Στο παρακάτω σχήμα φαίνονται όλοι οι μετασχηματισμοί τύπων που γίνονται κατά τον υπολογισμό της παράστασης

```
r=(ch / i) + (f * d) - (f + i)
```

```
-----
```

```
i d f
```

```
-----
```

```
d
```

```
-----
```

```
d
```

Μετασχηματισμός του τύπου μιας παράστασης μπορεί να γίνει προσδιορίζοντας τον νέο τύπο μέσα σε παρένθεση πριν από την παράσταση. π.χ.

```
(τύπος)<παράσταση>
```

Για παράδειγμα

```
int i;
```

```
(float) i/2;
```

Τέλος στη C μπορούμε να έχουμε πολλαπλή εκχώρηση $x=y=z=0$;

Αριθμητικοί τελεστές

-	αφαίρεση, πρόσημο
+	πρόσθεση
*	πρόσθεση
/	διαίρεση
%	(mod)
--	ελάττωση μεταβλητής κατά 1
++	αύξηση μεταβλητής κατά 1

Οι τελεστές -- και ++ μπορεί να τοποθετηθούν μπροστά ή μετά ένα τελεσταίο. Η εντολή --x; ισοδυναμεί με την $x:=x-1$ αλλά η αφαίρεση εκτελείται πριν χρησιμοποιήσουμε την τιμή της x. Όμοια και η εντολή ++x;

Η εντολή x--; ισοδυναμεί με την $x:=x-1$ αλλά η αφαίρεση εκτελείται αφού χρησιμοποιήσουμε την τιμή της x.

```
x=3; x=3;
```

```
y=++x; y=x++;
```

αποτέλεσμα: $y=4$ ($x=4$) αποτέλεσμα: $y=3$ ($x=4$)

Συσχεσιακοί και λογικοί τελεστές

>

>=

<

<=

== ίσον

!= διάφορο του ίσον

&& AND

|| OR

! NOT

Στη C true είναι μια τιμή διαφορετική του μηδενός και false είναι το μηδέν.

Τελεστές χειρισμού bits (bitwise operators)

Χειρισμός των bits σημαίνει την δυνατότητα επέμβασης στα bits ενός byte ή μιας λέξης που αντιστοιχούν στους τύπους char και int.

& AND

| OR

^ XOR

~ συμπλήρωμα ως προς 1

>> shift right

<< shift left

Για παράδειγμα η παράσταση (ch & 127) όπου ch είναι τύπου char εκχωρεί την τιμή 0 στο parity bit.

ch: 1 0 0 1 1 1 0 1

127: 0 1 1 1 1 1 1 1

& 0 0 0 1 1 1 0 1

Ο τελεστής << έχει τον τύπο: variable << number και μετακινεί τα όλα τα bits της μεταβλητής προς τ' αριστερά number θέσεις. Οι κενές θέσεις που δημιουργούνται από τα δεξιά αντικαθίστανται με 0. π.χ.

x=5; x: 0 0 0 0 0 1 0 1

x=x << 2; 0 0 0 1 0 1 0 0

(αποτέλεσμα x=20)

Συντομεύσεις (Shorthands)

Η C σε ορισμένες περιπτώσεις εκχώρησης μας επιτρέπει να συντομεύσουμε τον κώδικα. Έτσι μία παράσταση της μορφής:

<μεταβλητή> =<μεταβλητή> <τελεστής><παράσταση>

μπορεί να γραφτεί ως:

<μεταβλητή> <τελεστής> = <παράσταση>

Για παράδειγμα οι εντολές:

```
year+=11; year=year+11;
```

```
i-=3; ισοδυναμούν με i=i-3;
```

```
x*=10; x=x*10;
```

```
y/=5; y=y/5;
```

"Hello", Το πρώτο πρόγραμμα C

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
printf("Hello there\n");
```

```
return 0;
```

```
}
```

Παρατηρήσεις

Το σώμα του προγράμματος περικλείεται στις αγκύλες "{", "}".

Κάθε εντολή της C τελειώνει με ";" (semicolon).

Σχόλια μιας γραμμής εισάγονται "//". Σχόλια περισσοτέρων γραμμών περικλείονται με /*.....*/

Η πρώτη γραμμή #include <stdio.h> ενσωματώνει το αρχείο standard I/O στο πρόγραμμά μας. Το αρχείο αυτό περιέχει συναρτήσεις I/O.

Το κύριο πρόγραμμα αρχίζει με "int main()" και επιστρέφει (return) 0.

Η εντολή printf εμφανίζει στην οθόνη την συμβολοσειρά "Hello there" και μετά την χαρακτήρα αλλαγής γραμμής (\n).

Παραδείγματα χρήσης απλών μεταβλητών

1)

```
#include <stdio.h>

int main()
{
    int a,b;
    int c=20;
    a=30;
    printf("Type the value of b: ");
    scanf("\n%d", &b);
    printf("value of b is %d ", b);
    b=a-c;
    printf("value of a-c is %d", b);
    return 0;
}
```

Αποτελέσματα

Type the value of b: 100

value of b is 100%d

value of a-c is 10

2)

```
#include <stdio.h>

void main()
{
    double r1, r2;
    long r3;
```

```

unsigned int r4;
r1=4.0/3; //float ---> double
r2=4/3; //int ---> double
r3=r1; //double ---> long
r4=-r1; //double ---> unsigned int
printf("double: 4.0/3= ", r1);
printf("double: 4/3= ", r2 );
printf("long: 4.0/3= ", r3 );
printf("unsigned int: -(4.0/3)=", r4 );
}

```

Αποτελέσματα

double: 4.0/3=1.333333

double: 4.0/3=1.000000

long: 4.0/3=1

unsigned int: -(4.0/3)=1

3)

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
int m=24, n=2;
```

```
int r1, r2, r3, boolean;
```

```
r1=r2=3+n++*24-(m-=25);
```

```
r3=r1--;
```

```
boolean=r1>r3 && 1 || n<(m*=12);
```

```
printf("r1= %d\n", r1);
```

```
printf("r2= %d\n", r2);
```



```
printf("r3= %d\n", r3);  
if (boolean) printf("boolean = TRUE\n");  
else  
printf("boolean = FALSE\n" );  
}
```

Αποτελέσματα

r1=51

r2=52

r3=52

FALSE

Συγκεκριμένα όσον αφορά το Arduino:

FUNCTIONS

Digital I/O

digitalRead()

digitalWrite()

pinMode()

Analog I/O

analogRead()

analogReference()

analogWrite()

Zero, Due & MKR Family

analogReadResolution()

analogWriteResolution()

Advanced I/O

noTone()

pulseIn()

pulseInLong()

shiftIn()

shiftOut()

tone()

Time

delay()

delayMicroseconds()

micros()

millis()

Math

abs()

constrain()

map()

max()

min()

pow()

sq()

sqrt()

Trigonometry

cos()

sin()

tan()

Characters

isAlpha()

isAlphaNumeric()

isAscii()

isControl()

isDigit()

isGraph()

isHexadecimalDigit()

isLowerCase()

isPrintable()

isPunct()

isSpace()

isUpperCase()

isWhitespace()

Random Numbers

random()

randomSeed()

Bits and Bytes

bit()

bitClear()

bitRead()

bitSet()

bitWrite()

highByte()

lowByte()

External Interrupts

attachInterrupt()

detachInterrupt()

Interrupts

interrupts()

noInterrupts()

Communication

Serial

stream

USB

Keyboard

Mouse

VARIABLES

Constants

Floating Point Constants

Integer Constants

HIGH | LOW

INPUT | OUTPUT | INPUT_PULLUP

LED_BUILTIN

true | false

Constants

Floating Point Constants

Integer Constants

HIGH | LOW

INPUT | OUTPUT | INPUT_PULLUP

LED_BUILTIN

true | false

Conversion

byte()

char()

float()

int()

long()

word()

Data Types

String

String()

array

bool

boolean

byte

char

double

float

int

long

short

unsigned char

unsigned int

unsigned long

void

word

Variable Scope & Qualifiers

const

scope

static

volatile

Utilities

PROGMEM

sizeof()

STRUCTURE

Sketch

loop()

setup()

Control Structure

break

continue

do...while

else

for

goto

if...else

return

switch...case

while

Further Syntax

#define (define)

#include (include)

/* */ (block comment)

// (single line comment)

;(semicolon)

{ } (curly braces)

Comparison Operators

!= (not equal to)

< (less than)

<= (less than or equal to)

== (equal to)

> (greater than)

>= (greater than or equal to)

Arithmetic Operators

% (remainder)

* (multiplication)

+ (addition)

- (subtraction)

/ (division)

= (assignment operator)

Boolean Operators

! (logical not)

&& (logical and)

|| (logical or)

Bitwise Operators

& (bitwise and)

<< (bitshift left)

>> (bitshift right)

^ (bitwise xor)

| (bitwise or)

~ (bitwise not)

Compound Operators

&= (compound bitwise and)

*= (compound multiplication)

++ (increment)

+= (compound addition)

-- (decrement)

-= (compound subtraction)

/= (compound division)

^= (compound bitwise xor)

|= (compound bitwise or)

Pointer Access Operators

& (reference operator)

* (dereference operator)

6. Τροφοδοσία Arduino

Ο Arduino μπορεί να τροφοδοτηθεί είτε με καλώδιο και απευθείας σύνδεση στον υπολογιστή, είτε με 12V power supply με barrel jack(Εικόνα 1), είτε με Battery Button Power Cable Tieline(Εικόνα 2) στο οποίο συνδέω μια μπαταρία αλκαλική των 9V, είτε με ένα 9v battery holder τοποθετώντας 6 μπαταρίες AA(Εικόνα 3).

Εικόνα 1(Barrel Jack)



Εικόνα 2(Battery Button Power Cable Tieline)



Εικόνα 3(9V Battery Holder)



Για να γίνει τροφοδοσία εκτός υπολογιστή, μπορεί να χρησιμοποιηθεί ένας παλιός φορτιστής ή ένα περισσευούμενο τροφοδοτικό, αρκεί η τροφοδοσία να κυμαίνεται μεταξύ 7-12 V. Για χρήση χωρίς φορτιστή, μπορεί να χρησιμοποιηθεί παραδείγματος χάριν ένα Battery Button Power Cable Tieline με μια μπαταρία αλκαλική των 9V είτε ένα battery holder ή και ένα battery holder χρησιμοποιώντας 6 μπαταρίες AA(εικόνα 3).

7 Διαδικασία και Υλοποίηση Διπλωματικής

7.1 Κώδικας Διπλωματικής

Δηλώσεις μεταβλητών & σταθερών:

Με την εντολή "LiquidCrystal lcd(2, 3, 4, 5, 6, 7);" δηλώνω τις εισόδους με τις οποίες συνδέω μερικές εξόδους της οθόνης lcd. Η εντολή έχει ως εξής: "LiquidCrystal lcd(Rs, E, D4, D5, D6, D7);" Με την μεταβλητή thresholdUp δηλώνω το ανώτερο όριο(υγρασίας υψηλού επιπέδου όπου το χώμα είναι υγρό) υγρασίας του αισθητήρα υγρασίας εδάφους και με την μεταβλητή thresholdDown δηλώνω το κατώτερο όριο (υγρασίας χαμηλού επιπέδου όπου το χώμα είναι στεγνό). Όπου έχει γίνει ορισμός σαν είσοδος A1 παραδείγματος χάριν(int humsensorPin = A1 πχ) , γίνεται αναφορά στις αναλογικές εισόδους του Arduino Uno. Όπου έχει γίνει ορισμός εξόδου ή εισόδου με απλό νούμερο(πχ const int greenLedPin = 9;), αυτός ο ορισμός αφορά τις ψηφιακές εξόδους(οι οποίες μπορούν να χρησιμοποιηθούν και σα είσοδοι χρησιμοποιώντας αντί για OUTPUT, INPUT. π.χ pinMode(Humsensor, INPUT).). Με την εντολή "int <<word>>;" γίνεται ορισμός μια λέξης η οποία επρόκειτο να χρησιμοποιηθεί στις συνθήκες και ένα

παράδειγμα είναι το εξής" humidity = analogRead(humsensorPin);" όπου η εντολή για να γίνει η δήλωση είναι "int humidity;"(όπου humsensorPin η είσοδος για τον αισθητήρα υγρασίας εδάφους). Στην εντολή " humidity = analogRead(humsensorPin);" γίνεται ο ορισμός της λέξης "humidity" ως η νέα μεταβλητή που επρόκειτο να χρησιμοποιηθεί στις συνθήκες. Εδώ υπάρχουν 2 περιπτώσεις. Στην μια χρησιμοποιούμε το analogRead εφόσον έχουμε αναλογική είσοδο και στην άλλη περίπτωση που είναι της ψηφιακής εισόδου χρησιμοποιούμε το digitalRead. Επιπλέον, σε μερικά κομμάτια εξοπλισμού, όπως την οθόνη Lcd 16x2 και την αισθητήρα θερμοκρασίας DHT22, πρέπει να προσθέσουμε τις ανάλογες βιβλιοθήκες για να λειτουργήσουν.

```
#include <DHT.h>
```

```
#include <DHT_U.h>
```

```
#include <Adafruit_Sensor.h>
```

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(2, 3, 7, 6, 5, 4);
```

```
//humidity
```

```
int humsensorPin = A1;
```

```
int thresholdUp = 1000;
```

```
int thresholdDown = 780;
```

```
int LED = 8;
```

```
//temperature
```

```
#define DHTPIN 13 //Define sensor pin(pin 13)
```

```
#define DHTTYPE DHT22 //What sensor use
```

```
DHT dht(DHTPIN, DHTTYPE); //Create sensor object
```

```
boolean a=LOW,b=HIGH;

float h=0,t=0;

const int fan = 12; //fan pin for temperature control
```

```
long previousMillis = 0;

long interval = 2000; //Read sensor each 2 seconds
```

```
//Light
```

```
const int greenLedPin = 9;

const int blueLedPin = 10;

const int redLedPin = 11;
```

```
int lightSensorPin = A0;
```

```
int Value = 0;
```

Δηλώσεις εισόδων-εξόδων και αρχικών καταστάσεων:

Πρωτίστως, στην περίπτωση αυτή χρησιμοποιείται η εντολή "lcd.begin(16,2);" για να γίνει η εκκίνηση της οθόνης υγρών κρυστάλλων lcd 16x2. Επιπλέον, μέσω της εντολής "pinMode(X, OUTPUT-INPUT);" γίνεται ο καθορισμός για το αν μια ψηφιακή έξοδος θα χρησιμοποιηθεί σαν είσοδος η έξοδος. Προκειμένου να χρησιμοποιηθεί σαν είσοδος θα χρησιμοποιηθεί η εντολή "pinMode(X, OUTPUT);", ενώ για να χρησιμοποιηθεί μια ψηφιακή έξοδος σαν είσοδος θα χρησιμοποιηθεί η εντολή "pinMode(X, INPUT);". Με την εντολή "Serial.begin(9600);" εγκαθιστώ επικοινωνία της πλακέτας Arduino με τον υπολογιστή και το νούμερο όπως για παράδειγμα, στην προκειμένη περίπτωση το 9600 είναι η ταχύτητα μετάδοσης δεδομένων. Άλλες ταχύτητες μετάδοσης είναι οι εξής: 300, 600, 1200, 2400, 4800, 14400, 19200, 28800, 38400, 57600, 115200. Με την εντολή digitalWrite(X, LOW) ορίζω την αρχική κατάσταση της εξόδου(όπου "X" το αντικείμενο, το οποίο θα έχει την αρχική κατάσταση. Παραδείγματος χάριν, fan ή led).

```
void setup() {

Serial.begin(9600);

lcd.begin(16,2);

//temperature

pinMode(fan, OUTPUT);//Change to output my pins
```

```

pinMode(LED, OUTPUT);
dht.begin();//Start DHT22 sensor
digitalWrite(LED, LOW);//
digitalWrite(fan,LOW);//Turn off fan
//Light
pinMode(greenLedPin, OUTPUT);
pinMode(blueLedPin, OUTPUT);
pinMode(redLedPin, OUTPUT);
}

```

Επαναλαμβανόμενος κώδικας:

Μέσα στον επαναλαμβανόμενο κώδικα θα γίνουν όλες οι προκαθορισμένες διεργασίες και χρησιμοποιώντας μετά από κάθε κομμάτι του project την εντολή "delay(time);" θα ορίσουμε τον χρόνο καθυστέρησης της επανεκκίνησης της κάθε επανάληψης.

```

void loop() {

//humidity
{
int humidity;
int percentage;
humidity = analogRead(humsensorPin);
percentage = humidity*0.09765625;//Soil's moisture percentage equation
lcd.setCursor(9,0);
lcd.print("h%=");
lcd.setCursor(12,0);
lcd.print(percentage);

if (humidity <= thresholdDown)

```

```

{
  lcd.setCursor(0,1);
  lcd.print("Needs watering");
  digitalWrite(LED, HIGH);
  Serial.println("Low hum level");
} else if (humidity >= thresholdUp)
{
  digitalWrite(LED, LOW);
  lcd.setCursor(0,1);
  lcd.print("It's watered");
  Serial.println("High hum level");
  delay(2000);

}
}

//temperature
{
  unsigned long currentMillis = millis();//time elapsed

  if(currentMillis - previousMillis > interval) //Comparison between the elapsed time
and the time in which the action is to be executed
  {
    previousMillis = currentMillis; //"Last time is now"

    h = dht.readHumidity();//humidity value

    t = dht.readTemperature();//temperature value centigrades if you want fahrenheit
change to

```

```

//t = dht.readTemperature(true);
//Below is for print data sensors in lcd
lcd.setCursor(0,0);
lcd.print("t=");
lcd.setCursor(3,0);
lcd.print(t);

if(t>=31 && a==LOW)//if temperature above of 25 degrees
{
    delay(1000);
    digitalWrite(fan,HIGH);//Active air conditioner
    a=HIGH;
    b=LOW;

}
else if(t<31&&b==LOW)//if temperature is under 23 degrees
{
    delay(1000);
    digitalWrite(fan,LOW);//Turn off air conditioner
    a=LOW;
    b=HIGH;
}

}
}

```



```
//Light
{
  Value = analogRead(lightSensorPin);
  if(Value < 50){

    digitalWrite(redLedPin, HIGH);
    digitalWrite(greenLedPin, LOW);
    digitalWrite(blueLedPin, LOW);
  }
  else if(Value >= 50 && Value <= 100){

    digitalWrite(redLedPin, LOW);
    digitalWrite(greenLedPin, LOW);
    digitalWrite(blueLedPin, HIGH);
  }
  else{

    digitalWrite(redLedPin, LOW);
    digitalWrite(greenLedPin, HIGH);
    digitalWrite(blueLedPin, LOW);
  }
  delay(500);

}
}
```

Τοποθέτηση λέξεων και μετρήσεων στην οθόνη:

Για να γίνει η τοποθέτηση των μετρήσεων στην οθόνη, έγινε καθορισμός των ορίων μέσω της εντολής `lcd.setCursor(x,y)`, όπου `x` η σημείο της γραμμής και `y` η γραμμή. Στην προκειμένη περίπτωση, λόγω των μικρών διαστάσεων της οθόνης, μπορώ να χρησιμοποιήσω 2 γραμμές. Η μια είναι η 0 και η άλλη η 1. Η 0 είναι η πρώτη(πάνω γραμμή) και η 1 η δεύτερη(κάτω γραμμή). Όσον αφορά την θέση, που χρησιμοποιούμε την μεταβλητή `x`, χρησιμοποιείται για να ορίσουμε την αρχή, στην οποία θα εμφανιστεί στην οθόνη ή η μέτρηση, η οποία θα παρθεί από ένα από τα αισθητήρια, ή μια λέξη η μια φράση την οποία θα την καταχωρήσουμε μέσω της εντολής `"lcd.println("word-phrase")`.

```
{  
  
int humidity;  
  
humidity = analogRead(humsensorPin);  
  
lcd.setCursor(9,0);  
  
lcd.print("h=");  
  
lcd.setCursor(11,0);  
  
lcd.print(humidity);  
  
  
if (humidity <= thresholdDown)  
{  
    lcd.setCursor(0,1);  
    lcd.print("Low hum level");  
    Serial.println("Low hum level");  
} else if (humidity >= thresholdUp)  
{  
    lcd.setCursor(0,1);  
    lcd.print("High hum level");  
    Serial.println("High hum level");  
    delay(2000);  
}
```

```

}
}
//temperature
{
    unsigned long currentMillis = millis();//time elapsed

    if(currentMillis - previousMillis > interval) //Comparison between the elapsed time
    and the time in which the action is to be executed

    {

        previousMillis = currentMillis; //"Last time is now"

        h = dht.readHumidity();//humidity value

        t = dht.readTemperature();//temperature value centigrades if you want fahrenheit
        change to

        //t = dht.readTemperature(true);

        //Below is for print data sensors in lcd

        lcd.setCursor(0,0);

        lcd.print("t=");

        lcd.setCursor(3,0);

        lcd.print(t);

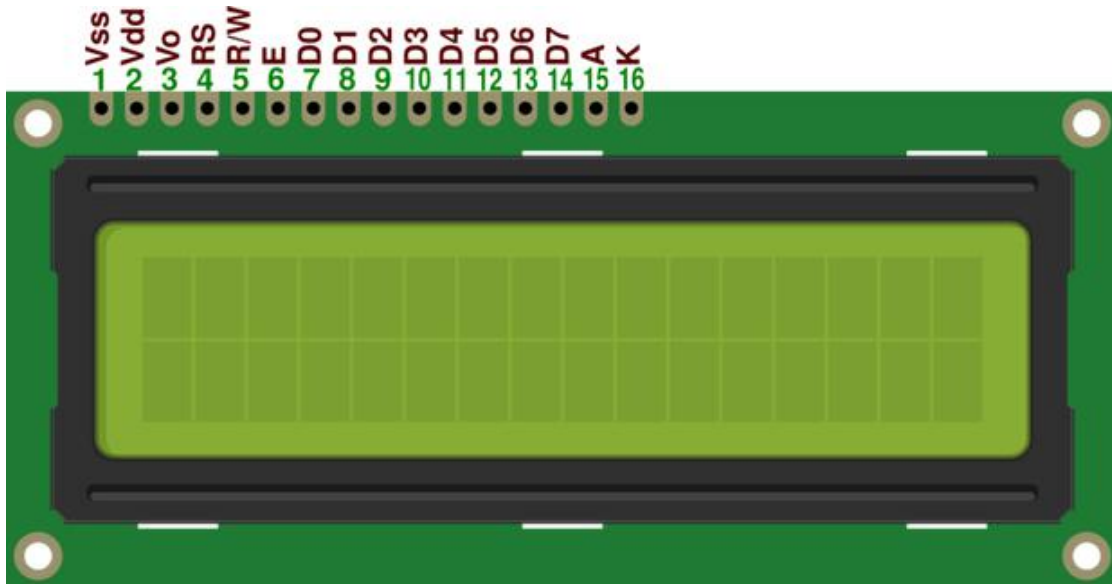
```

Βιβλιοθήκες:

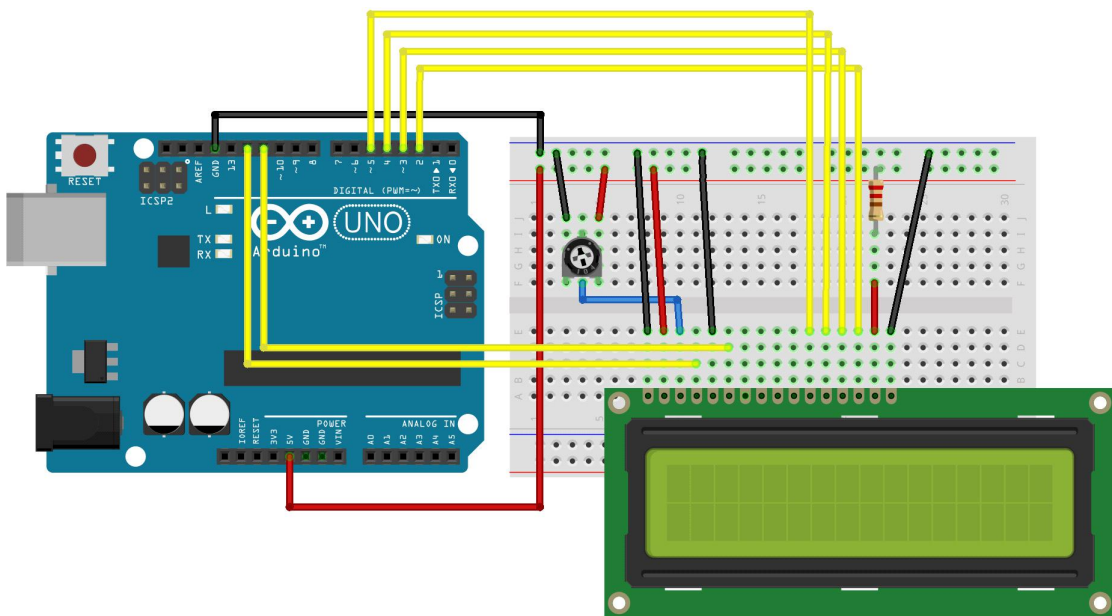
Σε μερικούς αισθητήρες και άλλα κομμάτια εξοπλισμού είμεθα υποχρεωμένοι να δηλώσουμε την ανάλογη βιβλιοθήκη, ούτως ώστε να μπορέσουμε να τα χρησιμοποιήσουμε. Ένα παράδειγμα είναι της lcd 16x2 οθόνης στην οποία πρέπει να προστεθει η ανάλογη βιβλιοθήκη(#include <LiquidCrystal.h>). Ένα επιπλέον παράδειγμα είναι του αισθητήρα DHT22(#include <DHT.h> #include <DHT_U.h>).

7.2 LCD Screen 16x2

Εικόνα



Κύκλωμα



Στιγμιότυπο από την διπλωματικής:



Γενικά:

Εντολές:

`LiquidCrystal lcd(Rs, E, D4, D5, D6, D7);` Όπου από κάθε έξοδο της οθόνης(της συγκεκριμένης εντολής) γίνεται σύνδεση με ψηφιακές εξόδους στον Arduino.

`lcd.begin(16,2);` Για να γίνει η εκκίνηση της οθόνης. Μέσα στην παρένθεση τοποθετούνται οι διαστάσεις της οθόνης.

`lcd.setCursor(x,y);` Με αυτήν την εντολή θέτουμε γραμμή(y) και σημείο πάνω στην γραμμή(x). 0 η πρώτη γραμμή, 1 η δεύτερη και ούτω καθεξής.

`lcd.print(t ή "message");` Με αυτήν την εντολή, ορίζουμε το μήνυμα ή την μέτρηση ενός αισθητήρα που θα αποτυπωθεί πάνω στην οθόνη. Για να αποτυπωθεί κάποια μέτρηση απλά γράφουμε την μεταβλητή στην οποία αποθηκεύεται η μέτρηση. Για την αποτύπωση μια φράσης ή μιας λέξης, χρησιμοποιώ της εξής μορφή `lcd.print("word-phrase);` (όπου t η μέτρηση)

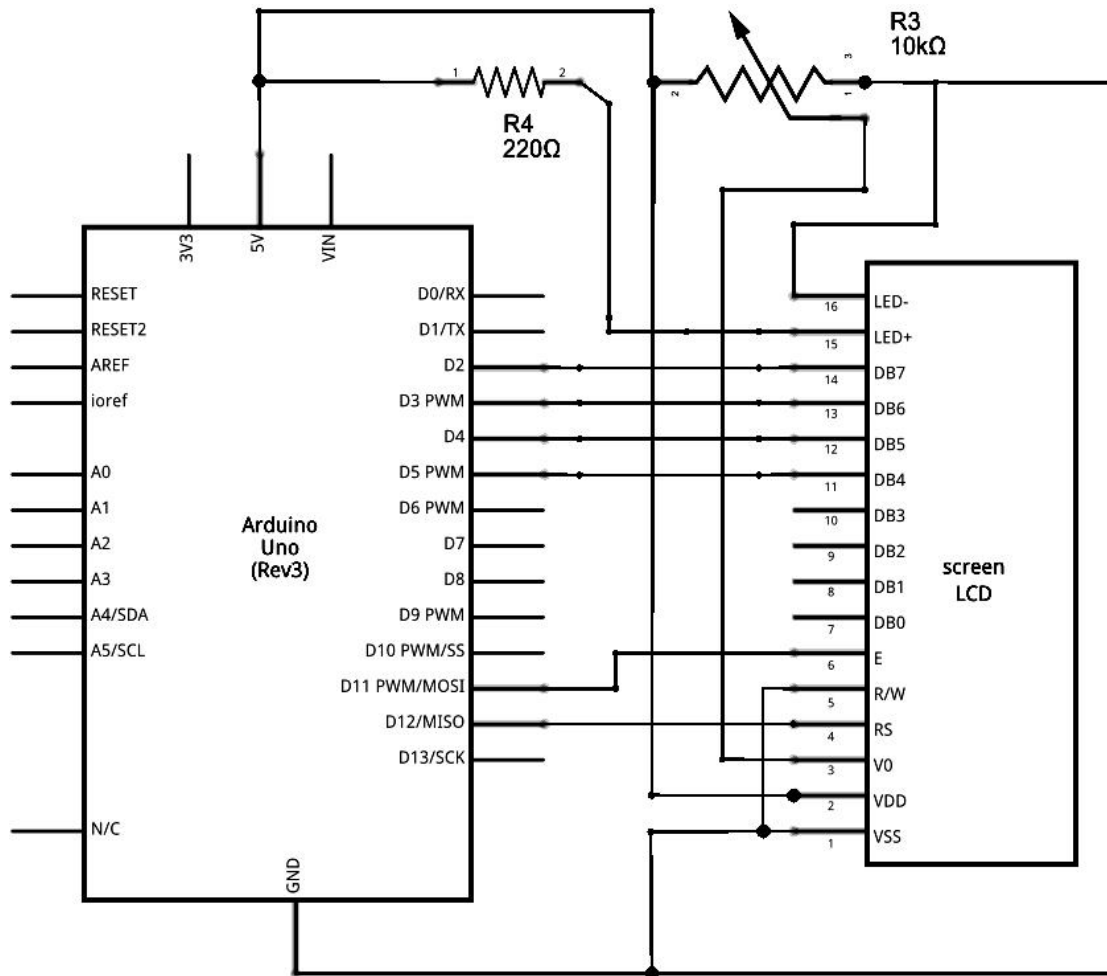
`lcd.clear();` Με αυτήν την εντολή, καθαρίζω την οθόνη και ο κέρσορας τοποθετείται στην πάνω αριστερή γωνία στην πρώτη γραμμή.

`lcd.Blink();` Διαλέγοντας την γραμμή και το σημείο πάνω στην γραμμή που θέλουμε, αναβοσβήνει το κομμάτι του επιλεγμένου κομματιού της γραμμής που επιθυμούμε.

`lcd.noBlink();` Χρησιμοποιώντας αυτήν την εντολή σταματάμε το διαδοχικό άναμα και το σβήσιμο του επιλεγμένου σημείου της γραμμής που πραγματοποιήσαμε με την εντολή `lcd.Blink();`.

Επίσης, για να μπορέσει απο τον χρήστη να γίνει ρύθμιση της φωτεινότητας της οθόνης, τοποθετείται μια αντίσταση 10KΩ(από την έξοδο A στο +5v του arduino και από την έξοδο V_o συνδέσω ένα καλώδιο σε ένα ποτενσιόμετρο των 10KΩ.

Σχεδιάγραμμα



Στο project αυτό χρησιμοποιώ μια αντίσταση των 10KΩ αντί μιας, της τάξης των 220Ω, στην έξοδο Rs την ψηφιακή έξοδο 2, στην έξοδο E την ψηφιακη έξοδο 3, στην έξοδο D4 την ψηφιακη έξοδο 7, στην έξοδο D5 την ψηφιακή έξοδο 6, στην έξοδο D6 την ψηφιακή έξοδο 5 και στην έξοδο D7 την ψηφιακή έξοδο 4 του Arduino.

7.3 Πρόγραμμα Arduino IDE

Με το συγκεκριμένο πρόγραμμα επρόκειτο να γίνει ο προγραμματισμός του Arduino Uno Rev3, καθώς εκεί θα γραφτεί το πρόγραμματιστικό κομμάτι της διπλωματικής. Ο κώδικας μπορεί να γραφτεί είτε σε C είτε σε C++.

Βασικό πλάνο



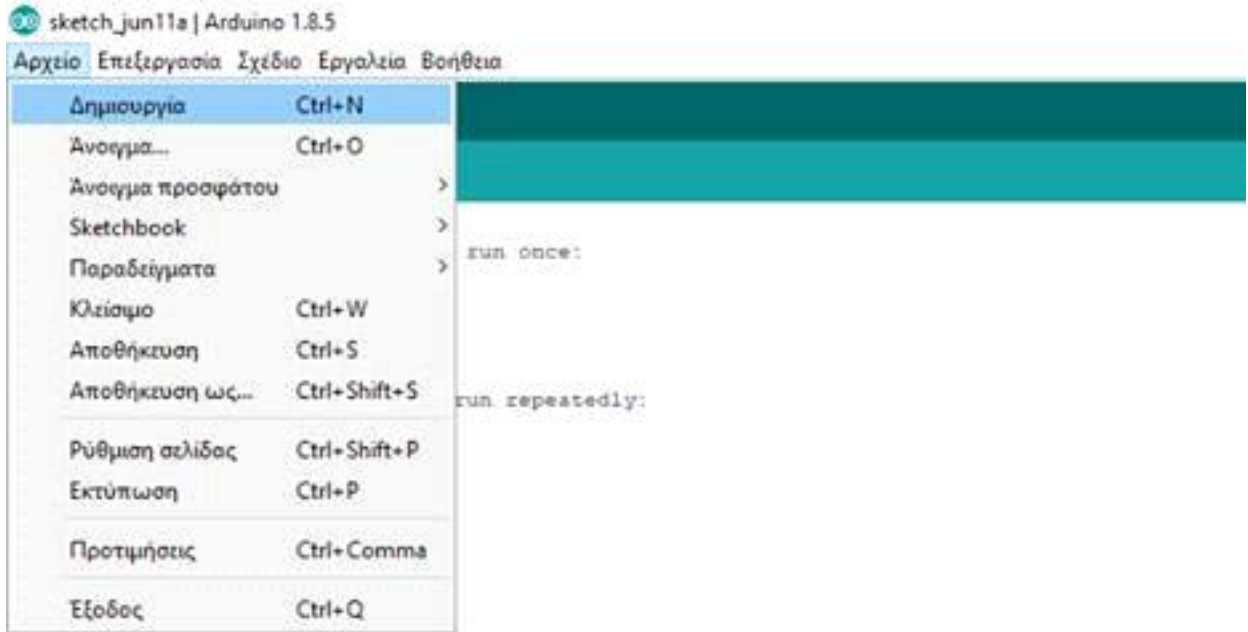
```
sketch_dec07a | Arduino 1.8.3
File Edit Sketch Tools Help
sketch_dec07a
void setup() {
  // put your setup code here, to run once:
}

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

2 Arduino/Genuino Uno on COM3
```

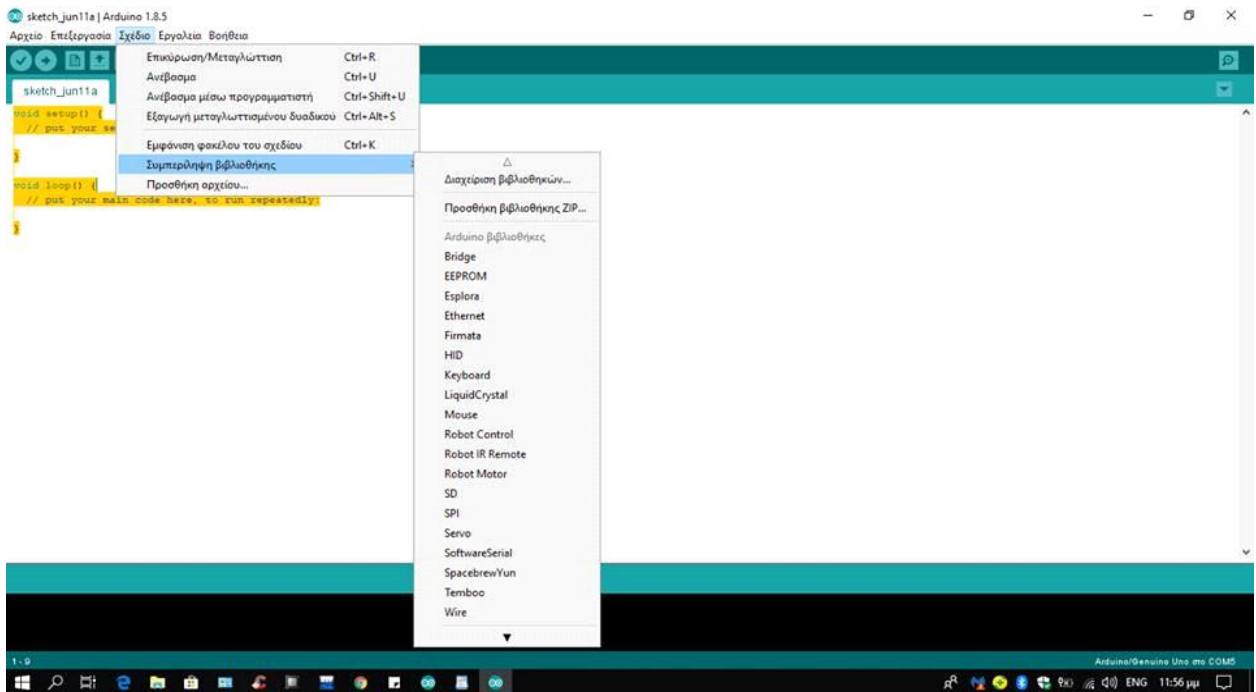
Στο κομμάτι του "void setup() { }" μέσα στις αγκύλες, γίνονται οι δηλώσεις εισόδων-εξόδων καθώς και την εγκατάσταση της επικοινωνίας(serial.begin() της πλακέτας Arduino Uno Rev3 με τον υπολογιστή χρησιμοποιώντας την ανάλογη ταχύτητα(π.χ 9600) για την μετάδοση των δεδομένων.Επίσης γίνεται η εκκίνηση της λειτουργίας της οθόνης lcd (μέσω της εντολής lcd.begin(16,2)).

Δημιουργία νέου project

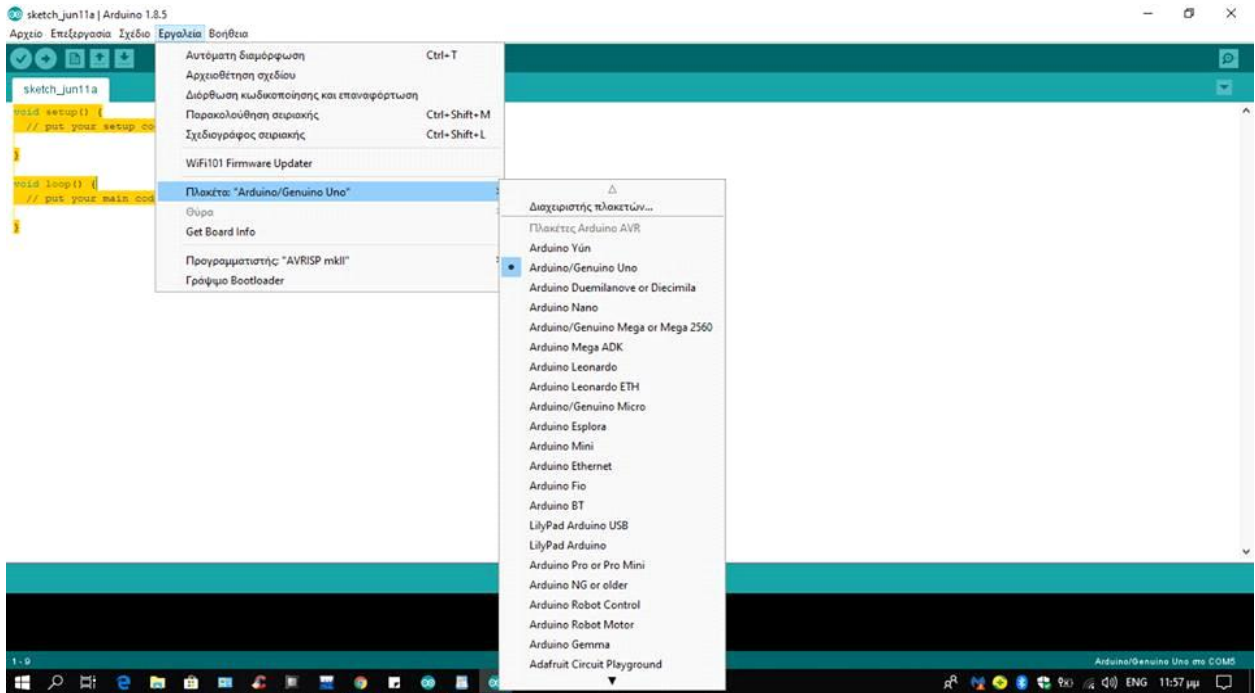


Συμπερίληψη βιβλιοθήκης(αισθητήρα ή άλλου κομματιού εξοπλισμού) στον κώδικα

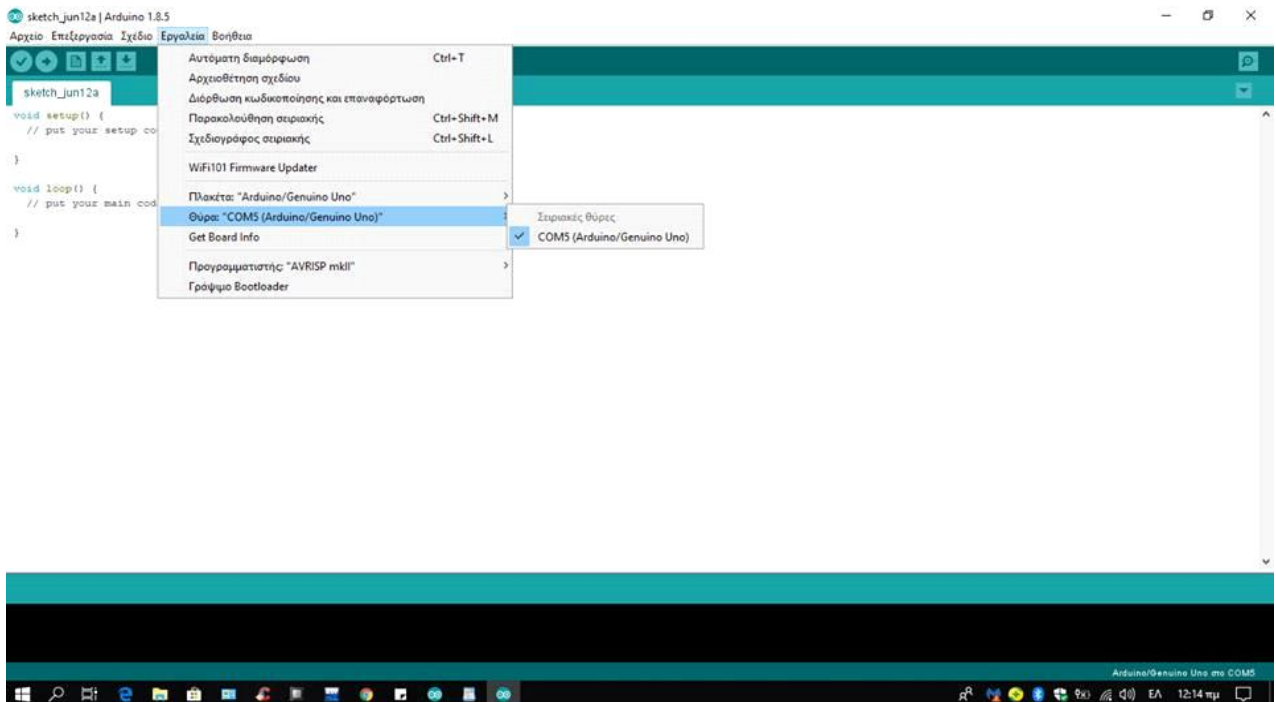
Με την επιλογή "Προσθήκη βιβλιοθήκης ZIP" μπορούμε να προσθέσουμε μια νέα βιβλιοθήκη στην λίστα επιλέγοντας ένα κατεβασμένο αρχείο ZIP της βιβλιοθήκης που επιθυμούμε να προσθέσουμε.



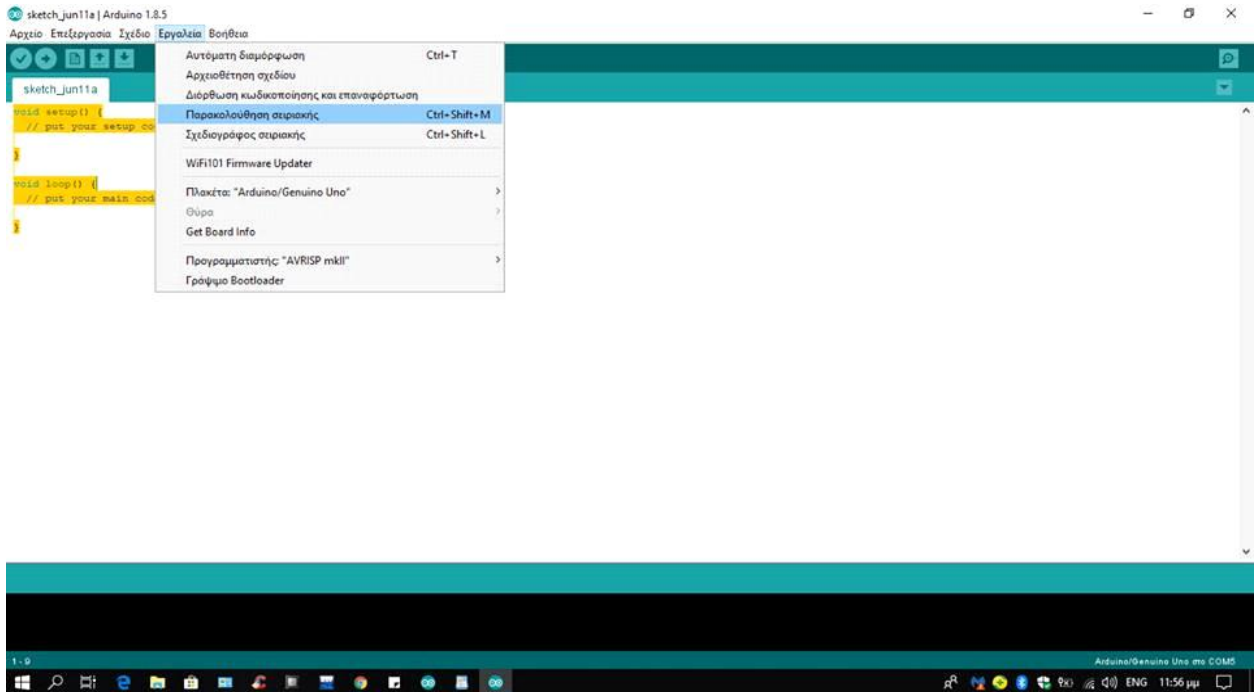
Επιλογή πλακέτας Arduino για σύνδεση



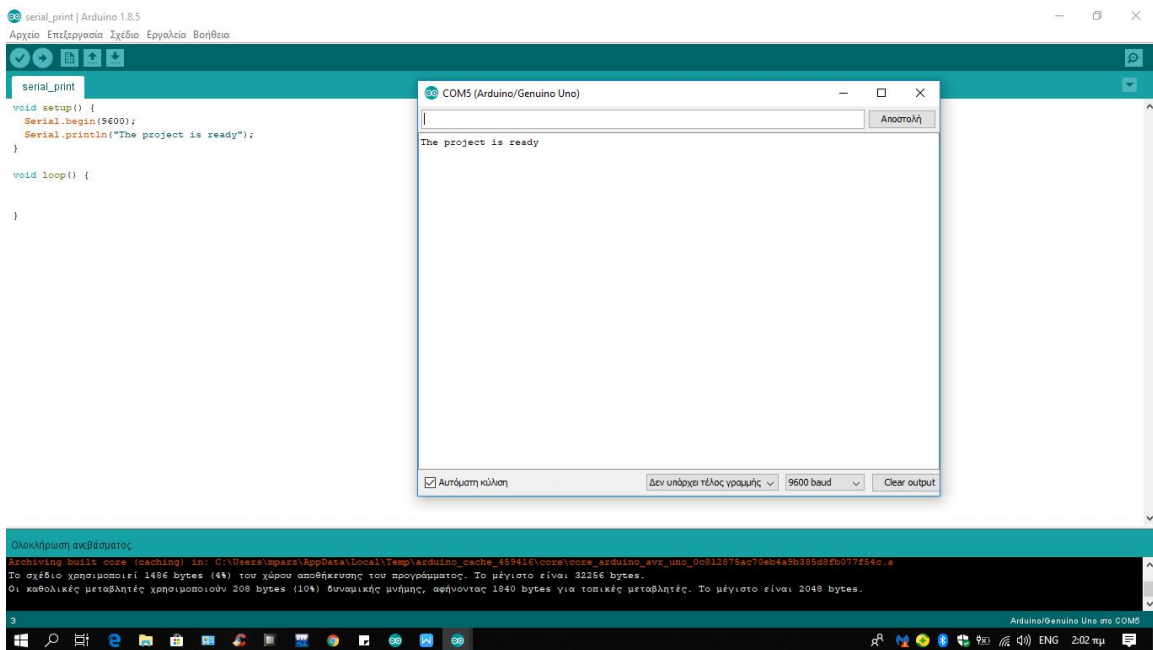
Έλεγχος για σύνδεση Arduino σε θύρα



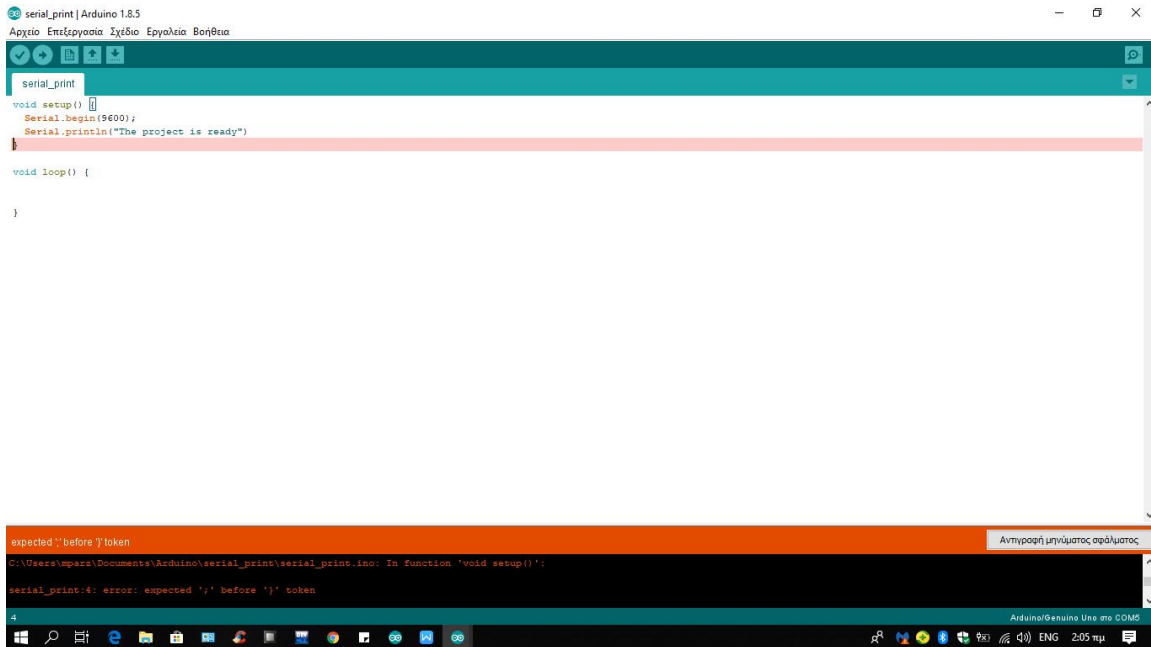
Παρακολούθηση των δεδομένων, τα οποία στέλνονται στο πρόγραμμα απο την πλακέτα Arduino Uno Rev3. Η εμφάνιση των δεδομένων γίνεται με την χρήση της εντολής Serial.println(). Μέσω της εντολής επιλέγουμε για να δωθεί ένα μήνυμα ή μια μέτρηση ενός αισθητήρα. Για να γίνει η εμφάνιση μια φράσης η μιας λέξης η εντολή γράφεται κάπως έτσι "Serial.println("phrase-word")".



Σειριακή παρακολούθηση



Ανάδειξη τυχόντων σφαλμάτων στον κώδικα



The screenshot shows the Arduino IDE interface. The main editor window displays the following code:

```
serial_print
void setup() {
  Serial.begin(9600);
  Serial.println("The project is ready");
}

void loop() {
}
}
```

Below the code editor, the Serial Monitor window is open, showing a compilation error:

```
expected ')' before ')' token
C:\Users\mpazz\Documents\Arduino\serial_print\serial_print.ino: In function "void setup()":
serial_print:4: error: expected ')' before ')' token
```

The error message is highlighted in orange, and a tooltip "Αντιγραφή μηνύματος σφάλματος" (Copy error message) is visible next to it. The Windows taskbar at the bottom shows the time as 2:05 pm and the language as ENG.

Μπάρα με κουμπιά



Το check κουμπί στο πρόγραμμα, πάνω στην μπάρα χρησιμοποιείται για τον έλεγχο του κώδικα. Όταν γραφτεί ο κώδικας, το πατάμε για να γίνει έλεγχος του κώδικα και βγάζει και τυχόν λάθη(δίνεται η γραμμή στην οποία υπάρχει η λανθάνουσα διατύπωση εντολής).

Το πάνω βέλος στην μπάρα με τα κουμπιά, το χρησιμοποιούμε για να ανεβάσουμε τον κώδικα στον Arduino για να τρέξει.

Το διακεκομμένο χαρτί αν πατηθεί, θα δημιουργήσει ένα νέο αρχείο.

Το βέλος που κοιτάζει προς τα πάνω χρησιμοποιείται για να ανοίξει μια υπάρχουσα εργασία.

Το βέλος που κοιτάζει προς τα κάτω, χρησιμοποιείται για να γίνει αποθήκευση του παρόντος αρχείου.

7.4 Πίνακας Υλικών

Υλικά	Πληροφορίες Τεμαχίων	Τεμάχια
Πλακέτα Προγραμ.	Arduino Uno R3	1
Καλώδια breadboard	Jumper Wire Cable for Solderless Breadboard Connections	26
Ανεμιστήρας	12V Fan 60x60x10mm (OEM)	1
Κάρτα Relay	5V 4-Channel Relay Module Board	1
Καλώδια Σύνδεσης M/M	Male to male Dupont Wires	12
Καλώδια Σύνδεσης M/F	Male to Female Dupont Wires	8
Ποτενσιόμετρο	Ρυθμιζόμενο Ποτενσιόμετρο 10ΚΩ	1
Αισθητήρας Υγρασίας	Sparkfan soil moisture sensor	1
Αισθ.Φωτεινότητας	5mm Φωτοαντίσταση LDR	1
Κόκκινο Led	Super Bright 3MM Red LED	1
Τροφοδοτικό	12V 2A DC Power Supply with Barrel Jack	1
Lcd 16x2	1602 16X2 Character LCD Display Module	1
Breadboard	Solderless MB-102 Breadboard 830	1
Μπλε Led	Super Bright 3MM blue LED	2
Πράσινο Led	Super Bright 3MM green LED	1
Ξύλο	Μακέτα πάχους 4mm	5
Αντιστάσεις	330Ω, 100Ω, 220Ω, 10ΚΩ	6
Αισθ.Θερμοκρασίας	DHT22 humidity-temperature	1

Το συνολικό κόστος, για όλα τα υλικά ανέρχεται στα 62,97 ευρώ. Το κόστος αντικειμενικά είναι ρευστό, καθώς οι τιμές από κατάστημα σε κατάστημα διαφοροποιούνται όπως και το κόστος μεταφορικών. Για παράδειγμα, για μια

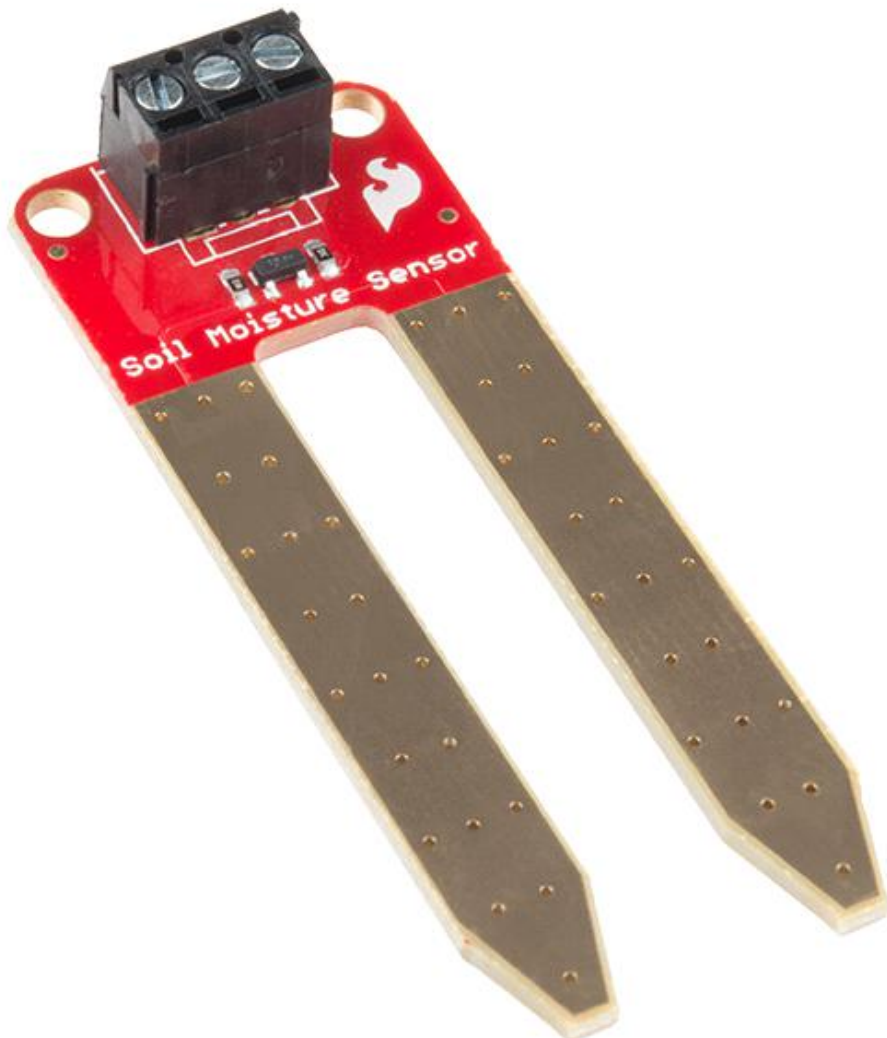
μεταφορά εξοπλισμού από Αθήνα προς πάτρα από την Cableworks τα συνολικά μεταφορικά είναι 3,9 ευρώ, ενώ από την hellas digital είναι 7 ευρώ.

7.5 Κομμάτια Project Διπλωματικής

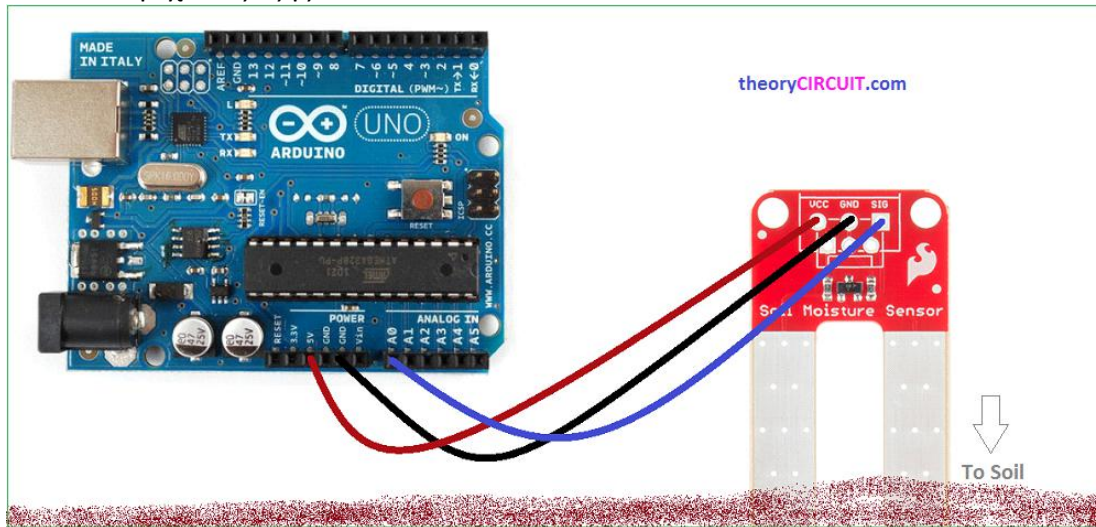
Η διπλωματική αποτελείται από διάφορα κομμάτια.

Υγρασία Εδάφους

Η υγρασία του εδάφους θα γίνεται μέσω του αισθητήρα Sparkfan Soil Moisture Sensor(αναλογική είσοδος A1) και σε μια οθόνη lcd 16x2 θα εμφανίζεται το ποσοστό επί της 100 της υγρασίας.



Η σύνδεση έχει ως εξής:

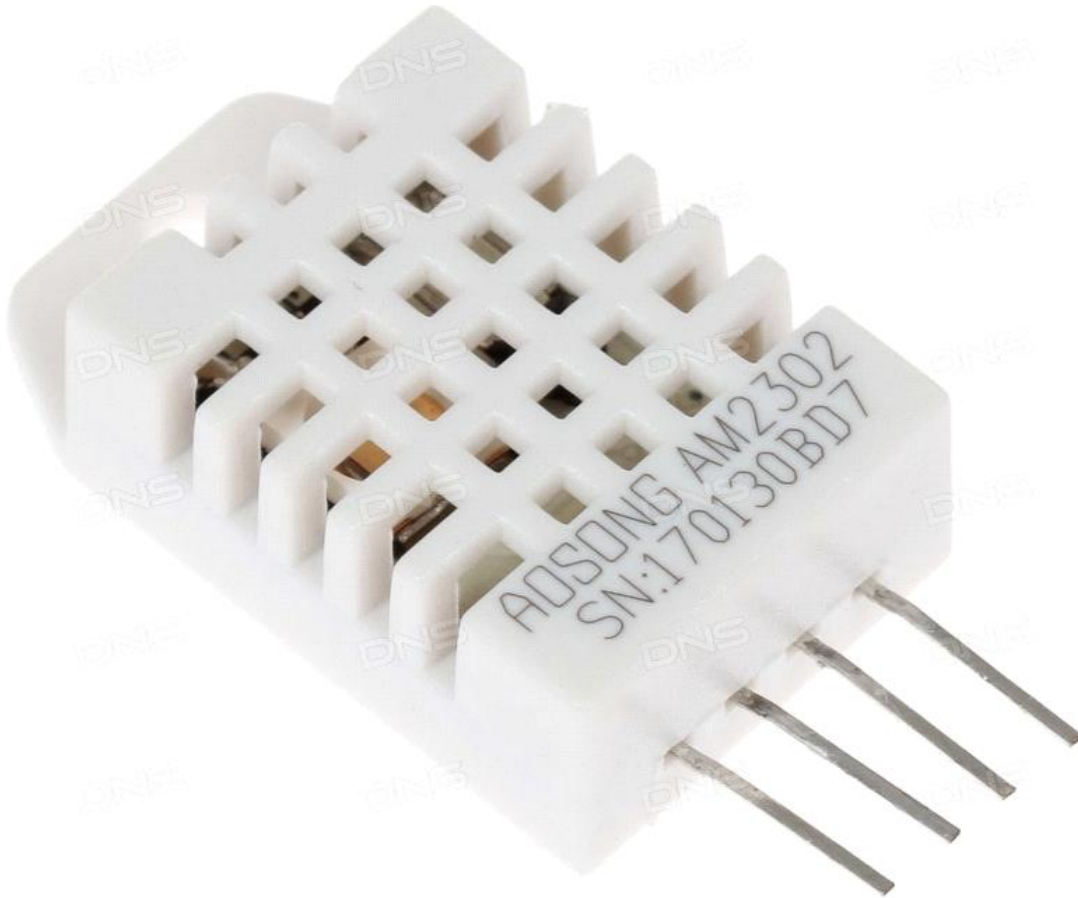


Θα εμφανίζεται ένα μήνυμα στην κάτω γραμμή όταν το φυτό θα χρειάζεται πότισμα, καθώς και όταν θα έχει ποτιστεί.

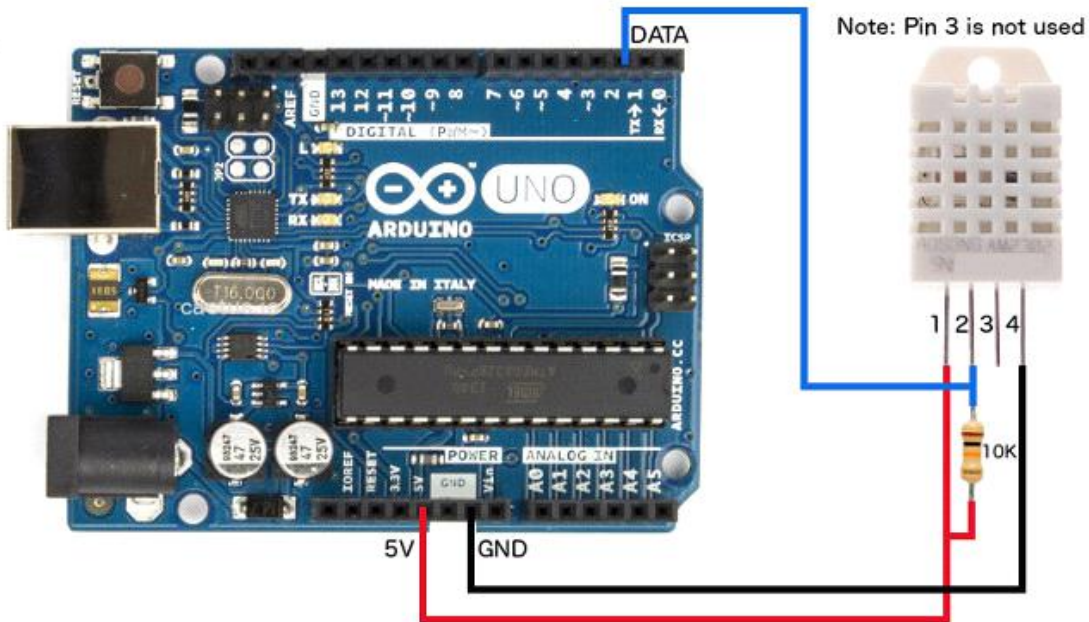


Θερμοκρασία περιβάλλοντος

Η θερμοκρασία περιβάλλοντος θα μετράται από τον αισθητήρα DHT22 και θα εμφανίζεται σε μια οθόνη Lcd 16x2 υγρών κρυστάλλων.



Η σύνδεση έχει ως εξής:



Στο Project της διπλωματικής, χρησιμοποιείται η ψηφιακή έξοδος 13 ως είσοδος του αισθητήρα θερμοκρασίας.

Σε περίπτωση ανόδου της θερμοκρασίας (άνω των 30 βαθμών κελσίου παραδείγματος χάριν) θα ενεργοποιηθεί ένας ανεμιστήρας(εικόνα 1) με την βοήθεια ενός Relay(εικόνα 2), 5V εισόδου το οποίο βγάζει έξοδο 30 Volts και ο ανεμιστήρας δουλεύει στα 12 Volts. Η σύνδεση, προκειμένου να λειτουργήσει ο ανεμιστήρας πρέπει να γίνει με τον εξής τρόπο. Θα συνδεθεί ένα καλώδιο από το com στα 5v, ένα καλώδιο από το κόκκινο της φάσης(του ανεμιστήρα) στην δεξιά θέση, όπου βρίσκεται η θέση NC(normally close), και από το μαύρο της γείωσης, στο σημείο GND(-) πάνω στην breadboard. Επίσης, χρησιμοποιώντας τρία καλώδια dupont M/F, θα συνδέσω το Relay στην πλακέτα Arduino.Θα συνδεθεί ένα καλώδιο από την θέση Vcc στα 5v, ένα ακόμα καλώδιο από την θέση GND στην θέση GND, και άλλο ένα καλώδιο, από την θέση In1 στην ψηφιακή έξοδο 12. Θ χρησιμοποιηθεί το Relay K1.

Εικόνα 1: Ανεμιστήρας



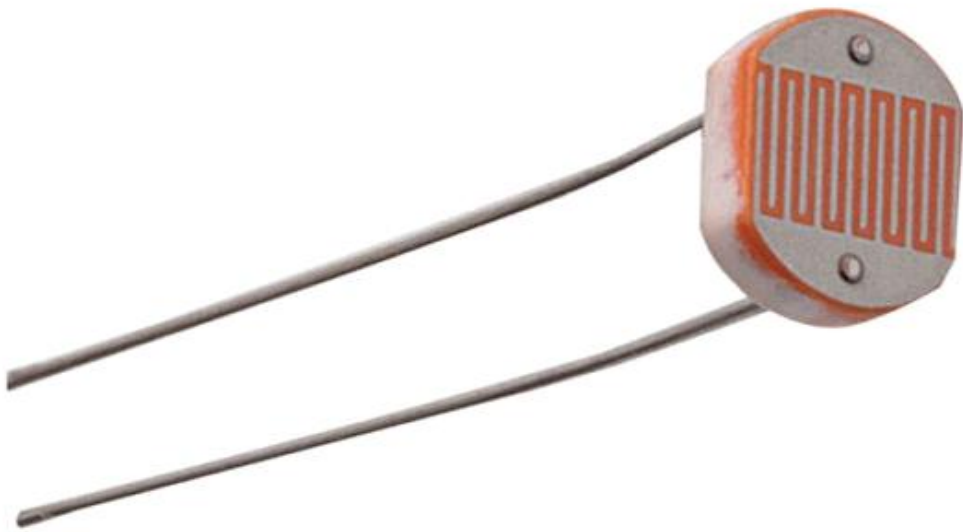
Εικόνα 2: Ρελε



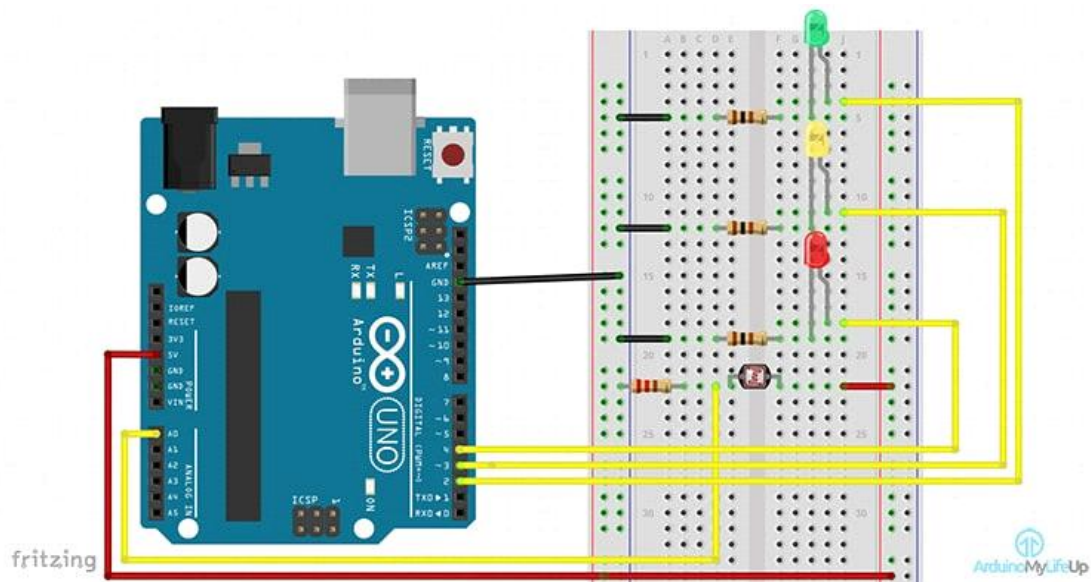
Έλεγχος Λάμπας

Η λειτουργία της λάμπας θα γίνει με έναν διαιρέτη τάσης (Επειδή δεν έφταναν οι έξοδοι, θα γίνει μια προσομοίωση με τα λαμπάκια που ήδη είναι συνδεδεμένα στον διαιρέτη τάσης). Όταν η φωτεινότητα θα είναι πολύ χαμηλή, η λάμπα θα ανάβει, ενώ όταν η φωτεινότητα θα είναι σε μέτριο ή υψηλό επίπεδο θα σβήνει. Στην περίπτωση που η φωτεινότητα θα είναι πολύ υψηλή, η τιμή της αντίστασης θα είναι χαμηλή ενώ όταν η φωτεινότητα θα είναι πολύ χαμηλή, η αντίσταση θα είναι πολύ υψηλή. Επίσης, όταν γίνεται εντοπισμός χαμηλού φωτισμού θα ανάβει ένα κόκκινο LED, υψηλού φωτισμού ένα πράσινο LED και μεσαίου φωτισμού ένα μπλε LED.

Αισθητήρας



Κύκλωμα Διαιρέτη τάσης με Φωτοαντίσταση



Στο Project της διπλωματικής, σαν είσοδος της φωτοαντίστασης χρησιμοποιείται η αναλογική είσοδος A0.

Κώδικες κομματιών διπλωματικής

Κώδικας υγρασίας εδάφους και ελέγχου αντλίας

```
LiquidCrystal lcd(2, 3, 7, 6, 5, 4);
```

```
int humsensorPin = A1;
```

```
int thresholdUp = 1000;
```

```
int thresholdDown = 780;
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  lcd.begin(16,2);
```

```
  pinMode(LED, OUTPUT);
```

```
}
```

```
void loop() {
```

```
  int humidity;
```

```
  int percentage;
```

```
  humidity = analogRead(humsensorPin);
```

```

percentage = humidity*0.09765625;//Soil's moisture percentage equation

lcd.setCursor(9,0);

lcd.print("h%=");

lcd.setCursor(12,0);

lcd.print(percentage);

if (humidity <= thresholdDown)
{
    lcd.setCursor(0,1);

    lcd.print("Needs watering");

    digitalWrite(LED, HIGH);

    Serial.println("Needs watering");
} else if (humidity >= thresholdUp)
{
    digitalWrite(LED, LOW);

    lcd.setCursor(0,1);

    lcd.print("It's watered");

    Serial.println("High hum level");

    delay(2000);
}
}

```

Κώδικας διαίρετη τάσης-Φωτοαντίστασης

```

const int greenLedPin = 9;

const int blueLedPin = 10;

const int redLedPin = 11;

int lightSensorPin = A0;

int Value = 0;

void setup() {

```

```

pinMode(greenLedPin, OUTPUT);

pinMode(blueLedPin,OUTPUT);

pinMode(redLedPin,OUTPUT);
}

void loop() {
Value = analogRead(lightSensorPin);

if(Value < 50){

digitalWrite(redLedPin, HIGH);

digitalWrite(greenLedPin, LOW);

digitalWrite(blueLedPin, LOW);

}

else if(Value >= 50 && Value <= 100){

digitalWrite(redLedPin, LOW);

digitalWrite(greenLedPin, LOW);

digitalWrite(blueLedPin, HIGH);

}

else{

digitalWrite(redLedPin, LOW);

digitalWrite(greenLedPin, HIGH);

digitalWrite(blueLedPin, LOW);

}

delay(500);

}

```

Κώδικας θερμοκρασίας και ελέγχου ανεμιστήρα

```

#include <DHT.h>

#include <DHT_U.h>

```

```

#include <Adafruit_Sensor.h>

#include <LiquidCrystal.h>

LiquidCrystal lcd(2, 3, 7, 6, 5, 4);

#define DHTPIN 13 //Define sensor pin(pin 13)

#define DHTTYPE DHT22 //What sensor use

DHT dht(DHTPIN, DHTTYPE); //Create sensor object

boolean a=LOW,b=HIGH;

float h=0,t=0;

const int fan = 12; //fan pin for temperature control

long previousMillis = 0;

long interval = 2000; //Read sensor each 2 seconds

void setup() {

Serial.begin(9600);

lcd.begin(16,2);

digitalWrite(fan, OUTPUT);

}

void loop() {

    unsigned long currentMillis = millis(); //time elapsed

    if(currentMillis - previousMillis > interval) //Comparison between the elapsed time and the time in
    which the action is to be executed

    {

        previousMillis = currentMillis; // "Last time is now"

        h = dht.readHumidity(); //humidity value

        t = dht.readTemperature(); //temperature value centigrades if you want fahrenheit change to

        //t = dht.readTemperature(true);

        //Below is for print data sensors in lcd

        lcd.setCursor(0,0);

```

```

lcd.print("t=");

lcd.setCursor(3,0);

lcd.print(t);

if(t>=31 && a==LOW)//if temperature above of 25 degrees
{
    delay(1000);

    digitalWrite(fan,HIGH);//Active air conditioner

    a=HIGH;

    b=LOW;

}

else if(t<31&&b==LOW)//if temperature is under 23 degrees
{
    delay(1000);

    digitalWrite(fan,LOW);//Turn off air conditioner

    a=LOW;

    b=HIGH;

}

}

}

```

7.6 Διαδικασία Υλοποίησης κατασκευής

Πιο αναλυτικά: 1.)Κατασκευή 2.)Ανεμιστήρας 3.) Αισθητήρας θερμοκρασίας 4.)Πλακέτα Breadboard 5.)Relay 6.)Γλάστρα 7.)Αισθητήρας υγρασίας χώματος 8.)LED αντλίας9.)LDR 10.)Ολοκληρωμένο project

1.



2.



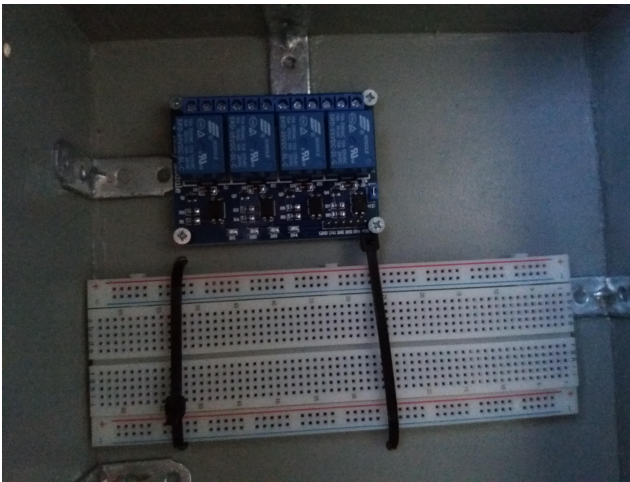
3.



4.



5.



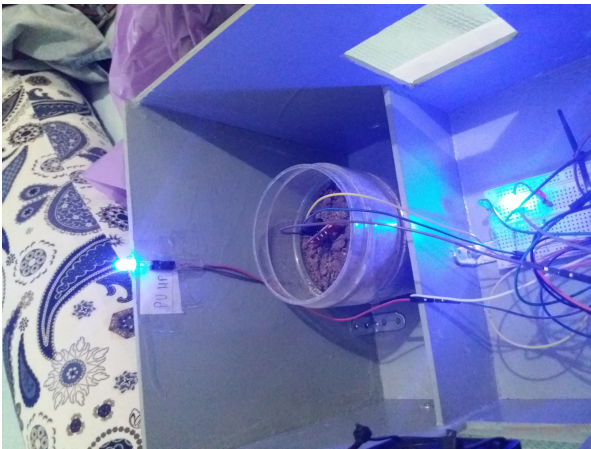
6.



7.



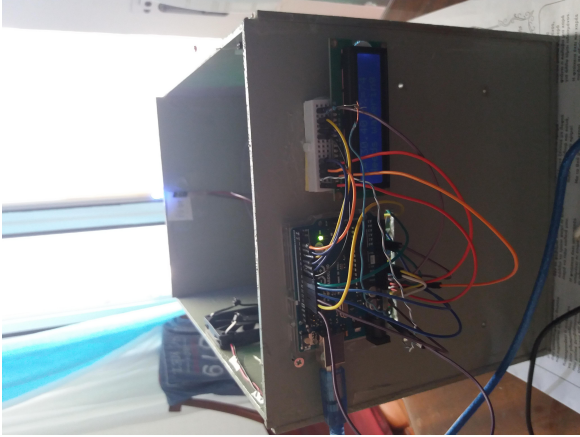
8.



9.



10.



Πηγές:

<https://cdn.sparkfun.com/datasheets/Sensors/Weather/RHT03.pdf>--> DHT22 αισθητήρας θερμοκρασίας

<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>--> Στοιχεία για αυτοματισμό(γενικώς)

https://learn.sparkfun.com/tutorials/soil-moisture-sensor-hookup-guide?_ga=2.11947760.592033775.1522007835-1818640274.1520177714--> Αισθητήρας υγρασίας

<https://en.wikipedia.org/wiki/Photoresistor>--> Αισθητήρας φωτός(φωτοαντίσταση)

<https://en.wikipedia.org/wiki/P%E2%80%93junction>--> Στοιχεία για p-n junction

<https://el.wikipedia.org/wiki/Arduino>--> Γενικά στοιχεία για τον Arduino

https://www.arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf--> Πιο συγκεκριμένα για τον Arduino Uno Rev3

<https://store.arduino.cc/arduino-uno-rev3>συγκεκριμένα--> Πιο συγκεκριμένα για τον Arduino Uno Rev3

<http://www2.aueb.gr/users/kalamboukis/frontistirio.html>--> Για την γλώσσα προγραμματισμού C(παρούσα χρησιμοποιηθέντα γλώσσα)

<https://www.arduino.cc/reference/en/>--> Χρησιμοποιηθέντες εντολές στον Arduino Uno Rev3

<https://el.wikipedia.org/wiki/%CE%91%CE%B9%CF%83%CE%B8%CE%B7%CF%84%CE%A%CF%81%CE%B1%CF%82>--> Γενικά στοιχεία για τους αισθητήρες

<https://www.hellasdigital.gr/electronics/sensors/temperature-sensors/ys-17-flame-sensor-5mm/><https://cdn-learn.adafruit.com/downloads/pdf/adafruit-arduino-lesson-11-lcd-displays-1.pdf>--> Αισθητήρας φωτιάς

<https://www.arduino.cc/en/Tutorial/LiquidCrystalDisplay>--> Εντολές για λειτουργία οθόνης lcd 16x2

<https://www.arduino.cc/en/Tutorial/LiquidCrystalBlin>--> Σχεδιάγραμμα οθόνης lcd 16x2

<https://arduinomylifeup.com/arduino-light-sensor/>--> Project διαίρετη τάσης

<https://www.digikey.com/en/maker/blogs/2018/introduction-to-the-arduino-ide>--> Στοιχεία για πρόγραμμα Arduino IDE

<http://playground.arduino.cc/Main/SimilarBoards#goShie>--> Arduino Shields