



# **ΤΕΙ ΠΕΙΡΑΙΑ**

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ**

**ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

**Π.Μ.Σ. “ΕΦΑΡΜΟΣΜΕΝΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ”**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Έξυπνο Σπίτι**

**Ανδρέας Δ. Μάρκελλος**

**Εισηγητής: Δρ Ιωάννης Έλληνας, Καθηγητής**

**ΑΘΗΝΑ 2017**



**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Έξυπνο Σπίτι**

**Ανδρέας Δ. Μάρκελλος**

**A.M. ais0079**

**Εισηγητής:**

**Δρ Ιωάννης Έλληνας, Καθηγητής**

**Ημερομηνία εξέτασης / /**



## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Ανδρέας Μάρκελλος, του Δημητρίου, με αριθμό μητρώου ais0079 φοιτητής του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονεμίσει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»



## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η παρούσα πτυχιακή εργασία αφιερώνεται στη σύζυγό μου Ζωγραφιά για την υπομονή και την στήριξη της καθ' όλη τη διάρκεια της εκπόνησης αυτής της πτυχιακής εργασίας.

Επίσης ευχαριστώ τον κ. Ιωάννη Έλληνα, ο οποίος υπήρξε καθηγητής μου και στις προπτυχιακές σπουδές μου, που δέχτηκε να εκπονήσω την παρούσα εργασία στο πλαίσιο του δικού του μαθήματος.

Τέλος θέλω να ευχαριστήσω και όλους τους υπόλοιπους καθηγητές που είχα κατά τη διάρκεια αυτού του μεταπτυχιακού προγράμματος.





## ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία ασχολείται με την ανάπτυξη ενός συστήματος το οποίο θα παρέχει υπηρεσίες αυτοματισμού σε ένα σπίτι, έτσι ώστε να ελέγχονται η θερμοκρασία η υγρασία η ποιότητα του αέρα και ο φωτισμός έκτακτης ανάγκης στο εσωτερικό του. Για το σκοπό αυτό χρησιμοποιήθηκε ο μικροελεγκτής Arduino MEGA 2560 R3, ένα shield κάρτας δικτύου για σύνδεση στο τοπικό δίκτυο, αισθητήρες, τα απαιτούμενα ρελέ και μια οθόνη απεικόνισης των μετρήσεων και των μηνυμάτων. Ο χειριστής θα μπορεί να ελέγχει τις συνθήκες που επικρατούν στο σπίτι χρησιμοποιώντας έναν οποιοδήποτε φυλλομετρητή, συνδεδεμένος στον web server της εφαρμογής, από οποιονδήποτε ηλεκτρονικό υπολογιστή, tablet ή κινητό το οποίο θα έχει πρόσβαση στο τοπικό δίκτυο ή στο διαδίκτυο.

## ABSTRACT

This present thesis deals with the development of a system that provides automation services in a home, so that temperature, humidity, air quality and emergency light can be checked. For this purpose a microcontroller Arduino MEGA 2560 R3, a network shield module, sensors, required relays and a display that depict the measurements and messages are been used. The operator can control the conditions inside home using a browser, linked to the web server application, from any computer, tablet or mobile phone that can be connected to the local network or to the internet.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Μικροελεγκτές.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: μικροελεγκτής, Arduino, έξυπνο σπίτι.

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>1. ΕΙΣΑΓΩΓΗ</b> .....	17
1.1 Περιγραφή του αντικειμένου της πτυχιακής εργασίας .....	17
1.2 Ιστορική αναδρομή .....	18
1.3 Απομακρυσμένη σύνδεση στην εφαρμογή .....	18
<b>2. ΤΑ ΕΞΑΡΤΗΜΑΤΑ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ &amp; ΤΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥΣ</b> ....	21
2.1 Γενικά .....	21
2.2 Arduino Mega 2560 R3 .....	22
2.3 Ethernet Shield W5100 .....	23
2.4 Αισθητήρες .....	24
2.4.1 Φωτεινότητας LDR – GL5539 .....	24
2.4.2 Θερμοκρασίας & Υγρασίας Χώρου (DHT-11) .....	25
2.4.3 Αισθητήρας Ποιότητας Αέρα .....	26
2.5 Οθόνη LCD .....	26
2.6 Real Time Clock (RTC) DS3231 .....	28
2.7 4_Relay Module .....	29
2.8 Άλλα Εξαρτήματα .....	30
2.8.1 Led's .....	30
2.8.2 Buzzer .....	30
2.8.3 Button .....	31
2.8.4 Αντιστάσεις .....	32
2.8.5 PCB .....	32
2.8.6 Pin Header .....	33
2.8.7 Αποστάτες, Βίδες .....	34
2.8.8 Μετασχηματιστής .....	35
2.8.9 Micro SD .....	36
2.8.10 Καλώδια .....	36
2.8.11 Ξύλινη βάση .....	37
<b>3. ΣΥΝΔΕΣΜΟΛΟΓΙΑ ΕΞΑΡΤΗΜΑΤΩΝ ΚΑΙ ΑΙΣΘΗΤΗΡΙΩΝ</b> .....	39
3.1 Γενικά .....	39

3.2 Ethernet shield .....	39
3.3 Οθόνη LCD .....	40
3.4 Αισθητήρας Θερμοκρασίας & Υγρασίας Χώρου (DHT-11) .....	41
3.5 Αισθητήρας Φωτεινότητας LDR – GL5539 .....	42
3.6 Αισθητήρας Ποιότητα Αέρα MQ135 .....	43
3.7 Real Time Clock (RTC) DS3231 .....	44
3.8 Buzzer .....	45
3.9 Button .....	46
3.10 Led's .....	47
3.11 4_Relay module .....	48
3.12 Σύνδεση περιφερειακών συσκευών .....	49
3.13 Όλη η κατασκευή .....	50
<b>4. Η ΚΑΤΑΣΚΕΥΗ ΚΑΙ Η ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ .....</b>	<b>51</b>
4.1 Η Κατασκευή .....	51
4.2 Η λειτουργία του συστήματος .....	53
4.2.1 Γενικά .....	53
4.2.2 Αρχεία της micro SD κάρτας .....	55
<b>5. ΤΟ WEB ΠΕΡΙΒΑΛΛΟΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ .....</b>	<b>57</b>
5.1 Γενικά .....	57
5.2 Σύνδεση (login) .....	57
5.3 Αυτόματος έλεγχος των συνθηκών .....	58
5.4 Μη αυτόματος έλεγχος των συνθηκών .....	59
5.5 Εμφάνιση των μετρήσεων .....	60
5.6 Αλλαγή ημερομηνίας και ώρας .....	60
5.7 Τα γραφήματα των μετρήσεων .....	61
5.8 Το web περιβάλλον σε smartphone ή tablet .....	62
<b>6. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ .....</b>	<b>65</b>
6.1 Σύνοψη της πτυχιακής εργασίας .....	65
6.2 Προοπτικές .....	65
<b>ΠΑΡΑΡΤΗΜΑ Α΄, Ο κώδικας του Arduino .....</b>	<b>68</b>

**ΠΑΡΑΡΤΗΜΑ Β΄, Ο κώδικας των web σελίδων ..... 71**

**ΒΙΒΛΙΟΓΡΑΦΙΑ ..... 75**

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

<b>Σχήμα 3.2:</b> Συνδεσμολογία της Ethernet shield .....	37
<b>Σχήμα 3.3:</b> Συνδεσμολογία της οθόνης LCD .....	38
<b>Σχήμα 3.4:</b> Συνδεσμολογία του DHT-11 .....	39
<b>Σχήμα 3.5:</b> Συνδεσμολογία του GL5539 .....	40
<b>Σχήμα 3.6:</b> Συνδεσμολογία του MQ135 .....	41
<b>Σχήμα 3.7:</b> Συνδεσμολογία του DS3231 .....	42
<b>Σχήμα 3.8:</b> Συνδεσμολογία του Buzzer .....	43
<b>Σχήμα 3.9:</b> Συνδεσμολογία του Button .....	44
<b>Σχήμα 3.10:</b> Συνδεσμολογία των Led .....	45
<b>Σχήμα 3.11:</b> Συνδεσμολογία του 4_Relay module .....	46
<b>Σχήμα 3.12:</b> Σύδεση περιφερειακών συσκευών .....	47
<b>Σχήμα 3.13:</b> Συνδεσμολογία όλων των εξαρτημάτων .....	48
<b>Σχήμα 4.1:</b> Το πλάνο της κατασκευής .....	49

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

<b>Πίνακας 2.1:</b> Τα εξαρτήματα της κατασκευής .....	20
<b>Πίνακας 2.8:</b> Αντιστάσεις .....	30

## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

- AJAX** Asynchronous JavaScript And XML
- I<sup>2</sup>C** Inter Integrated Circuit
- LDR** Light Dependent Resistor
- MSB** Most Significant Bit
- PCB** Printed Circuit Board
- PWM** Pulse Width Modulation
- RTC** Real Time Clock
- SCL** Serial Clock Line
- SDA** Serial Data Line
- SPI** Serial Peripheral Interface
- SRAM** Static Random-Access Memory
- UART** Universal Asynchronous Receiver/Transmitter
- USB** Universal Serial Bus
- UTP** Unshielded Twisted Pair





## ΚΕΦΑΛΑΙΟ 1

### ΕΙΣΑΓΩΓΗ

#### 1.1 Περιγραφή του αντικειμένου της πτυχιακής εργασίας

Αντικείμενο της παρούσας πτυχιακής εργασίας είναι η κατασκευή μίας συσκευής η οποία τοποθετημένη σε μία οικία να μπορεί να δώσει πληροφορίες και μετρήσεις για τις συνθήκες που επικρατούν στο χώρο, αλλά και συνδεδεμένη με το τοπικό δίκτυο να μπορεί ο χρήστης να συνδεθεί απομακρυσμένα με τη συσκευή, να ενημερώνεται και να ελέγχει αυτές τις συνθήκες.

Για το σκοπό αυτό θα χρησιμοποιηθεί ο μικροελεγκτής Arduino Mega 2560 R3 ο οποίος με τη βοήθεια των κατάλληλων αισθητηρίων και των συνδεδεμένων με αυτόν ρελέ, να ελέγχει διάφορες συσκευές του χώρου στο οποίο είναι συνδεδεμένος. Στην παρούσα κατασκευή χρησιμοποιούνται οι εξής αισθητήρες: φωτεινότητας, θερμοκρασίας, υγρασίας και ποιότητας αέρα. Παράλληλα μέσω τεσσάρων κατάλληλων ανεξάρτητων ρελέ μπορούμε να ελέγξουμε μία λάμπα έκτακτης ανάγκης, ένα θερμαντικό σώμα (π.χ. θερμοπομπό), έναν αφυγραντήρα και έναν εξαεριστήρα.

Οι μετρήσεις που λαμβάνει ο μικροελεγκτής μεταφέρονται στον ενσωματωμένο web server της εφαρμογής, ο οποίος όταν είναι προσβάσιμος από το δίκτυο που έχει συνδεθεί ο μικροελεγκτής, έχουμε πλήρη έλεγχο της εφαρμογής από οποιονδήποτε φυλλομετρητή.

#### 1.2 Σύντομη ιστορική αναδρομή

Εδώ και αρκετά χρόνια υπάρχει μεγάλο ενδιαφέρον για τους τρόπους που μπορεί ένα σπίτι να γίνει αυτοπονημένο, δηλαδή να μπορούν να ελεγχθούν οι ηλεκτρονικές και ηλεκτρικές συσκευές του κεντροποιημένα και σε ιδανική περίπτωση χωρίς καν να χρειάζεται η φυσική παρουσία κάποιου μέσα στο σπίτι. Ο κατακλυσμός των σπιτιών πια από ηλεκτρικές και ηλεκτρονικές συσκευές αλλά και η ραγδαία ανάπτυξη της τεχνολογίας προφανώς κάνει επιτακτική την ανάγκη ελέγχου όλων αυτών των συσκευών από μία και μόνη συσκευή είτε τοπικά είτε απομακρυσμένα.

Στην περίπτωση που κάποιος θέλει να έχει κεντροποιημένο τον έλεγχο των συσκευών του ενώ βρίσκεται στο σπίτι, απλά με συσκευές οι οποίες μπορούν να ελεγχθούν ασύρματα, μπορεί μέσω του κινητού του τηλεφώνου, μιας ταμπλέτας ή ενός υπολογιστή με το κατάλληλο λογισμικό (παρέχεται μαζί με τις εν λόγω συσκευές) να ενεργοποιήσει ή να απενεργοποιήσει τις συσκευές, όπως για παράδειγμα λαμπτήρες, πρίζες (και κατά συνέπεια ηλεκτρικές συσκευές), άνοιγμα και κλείδωμα θυρών, ρολών κ.α..

Πολλές μεγάλες εταιρίες αυτοματισμού δίνουν πια ολοκληρωμένες λύσεις χρησιμοποιώντας διάφορα πρότυπα που έχουν αναπτυχθεί τα τελευταία χρόνια. Σκοπός όλες οι ηλεκτρονικές και ηλεκτρικές συσκευές σε ένα σπίτι ή σε ένα κτήριο να είναι ελεγχόμενες κεντρικά και απομακρυσμένα.

### **1.3 Απομακρυσμένη σύνδεση στην εφαρμογή**

Σε περίπτωση που θέλουμε να έχουμε πρόσβαση στον web server της εφαρμογής ενώ δεν βρισκόμαστε στο χώρο, δηλαδή μέσω του διαδικτύου, θα πρέπει είτε να έχουμε μια στατική διεύθυνση από τον πάροχό μας, ή να ρυθμίσουμε κατάλληλα τον δρομολογητή μας ώστε να είναι ανιχνεύσιμος από το διαδίκτυο μέσω διαφόρων υπηρεσιών που υπάρχουν γι' αυτόν το σκοπό.

Υπάρχουν αρκετοί τρόποι που μας επιτρέπουν να έχουμε πρόσβαση απομακρυσμένα στον Arduino χωρίς να έχουμε κάποια στατική ip διεύθυνση. Πολλοί ιστοχώροι προσφέρουν δωρεάν απομακρυσμένη σύνδεση στις σελίδες τους, έχουν cloud στο οποίο ο Arduino στέλνει τα δεδομένα και διαβάζει εντολές από εκεί, με συνέπεια κάποιος χρησιμοποιώντας το cloud απομακρυσμένα να προγραμματίζει και να πληροφορείται για την κατάσταση κάποιων μετρήσεων ή την κατάσταση κάποιων pin του Arduino. Με αυτό τον τρόπο δεν χρειάζεται να φτιάξουμε καμία web εφαρμογή για τον έλεγχο του Arduino και αποφεύγουμε και όλα τα άλλα προβλήματα απομακρυσμένης σύνδεσης σε αυτόν. Η ανάγκη απομακρυσμένης σύνδεσης για έλεγχο μικροελεγκτών, καμερών, συναγερμών κτλ. σε ένα χώρο έχει ως αποτέλεσμα νέοι τέτοιοι ιστοχώροι να δημιουργούνται συνέχεια, προσπαθώντας να καλύψουν αυτές τις ανάγκες, για παράδειγμα: <https://thingspeak.com/>, <http://www.nearbus.net/> κ.α.. Σε αυτές τις περιπτώσεις

συνήθως χρειάζεται να τρέξουν προγράμματα υποστήριξης αυτής της επικοινωνίας στον μικροελεγκτή μας.

Άλλες υπηρεσίες στο διαδίκτυο με τις οποίες μπορούμε να έχουμε απομακρυσμένη πρόσβαση σε κάποια συσκευή στο τοπικό δίκτυό μας όπως είναι οι DynDNS και NoIP αρκεί να υποστηρίζονται από το δρομολογητή μας. Έχοντας δημιουργήσει λογαριασμό σε αυτές ενημερώνεται περιοδικά η εξωτερική φυσική διεύθυνση που έχει ο δρομολογητής μας και κάθε φορά που θέλουμε να συνδεθούμε απομακρυσμένα με το τοπικό μας δίκτυο προωθούμαστε στη σωστή φυσική διεύθυνση.

Η παρούσα εφαρμογή σχεδιάστηκε να δουλεύει όσο το δυνατόν ανεξάρτητα με το δεδομένο ότι συνδεόμαστε σε αυτή μόνο μέσω της Ethernet κάρτας δικτύου. Χρησιμοποιούνται κάποια έτοιμα κομμάτια κώδικα τα οποία καλούνται κατευθείαν από το διαδίκτυο για την απεικόνιση των διαγραμμάτων και της γραφικής αναπαράστασης των μετρήσεων, αλλά αυτό γίνεται για το λόγο ότι αν αποθηκευτούν στην SD κάρτα και καλούνται από εκεί, η εφαρμογή μας θα αργεί αρκετά να απεικονίσει τα δεδομένα.



## ΚΕΦΑΛΑΙΟ 2

### ΤΑ ΕΞΑΡΤΗΜΑΤΑ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ ΚΑΙ ΤΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥΣ

#### 2.1 Γενικά

Για την υλοποίηση της κατασκευής απαιτούνται τα παρακάτω εξαρτήματα και υλικά που φαίνονται στον πίνακα:

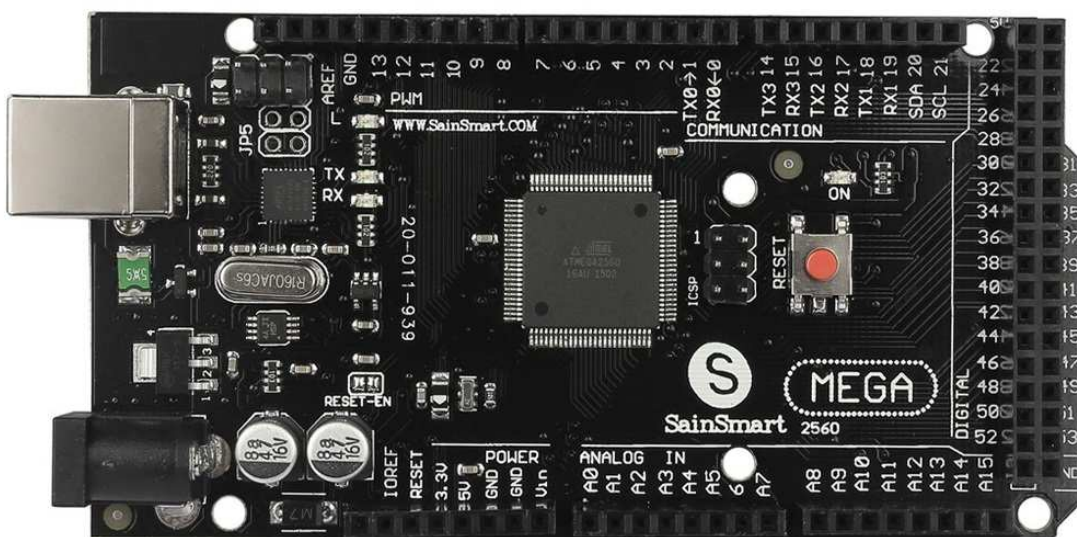
Πίνακας 2.1: Τα εξαρτήματα της κατασκευής.

Εξάρτημα	Ποσότητα
Arduino Mega 2560 R3	1
Ethernet Shield W5100	1
LCD οθόνη με I <sup>2</sup> C interface	1
4 Relay module	1
Real Time Clock DS3231	1
Φωτοαντίσταση GL5539	1
Αισθητήρας θερμοκρασίας & Υγρασίας DHT11	1
Αισθητήρας ποιότητας αέρα MQ135	1
Buzzer	1
Button	1
LED λευκό	1
LED κόκκινο	1
LED μπλε	1
LED κίτρινο	1
Αντίσταση 10ΚΩ	1
Αντιστάσεις 2,2ΚΩ	2
Αντιστάσεις 330Ω	2
Αντιστάσεις 220Ω	2
PCB	1
Pin Headers	2
Αποστάτες	14
Βίδες	14
Τροφοδοτικό	1
Micro SD	1
Καλώδιο UTP straight	1
Καλώδιο UTP cross over	1

Καλώδια jumpers	27
Καλώδιο usb	1
Ξύλινη βάση	1
Μπαταρία CR2032	1

Παρακάτω περιγράφονται αναλυτικά όλα τα εξαρτήματα της κατασκευής και τα χαρακτηριστικά τους.

## 2.2 Arduino Mega 2560 R3



Εικόνα 2.2: Arduino Mega 2560 R3

Η πλακέτα του Arduino Mega αποτελείται από ένα μικροελεγκτή ATmega2560 και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωσή του σε άλλα κυκλώματα. Περιλαμβάνει 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. Έχει 54 ψηφιακά pin εισόδου & εξόδου (εκ των οποίων 14 μπορούν να χρησιμοποιηθούν ως έξοδοι PWM), 16 αναλογικές εισόδους, 4 UARTs (σειριακές θύρες), μια σύνδεση USB, μια υποδοχή ρεύματος και ένα κουμπί επαναφοράς. Τα pin 3, 5, 6, 9, 10 και 11 του Arduino μπορούν να λειτουργήσουν και ως ψευδοαναλογικές έξοδοι με το σύστημα PWM,

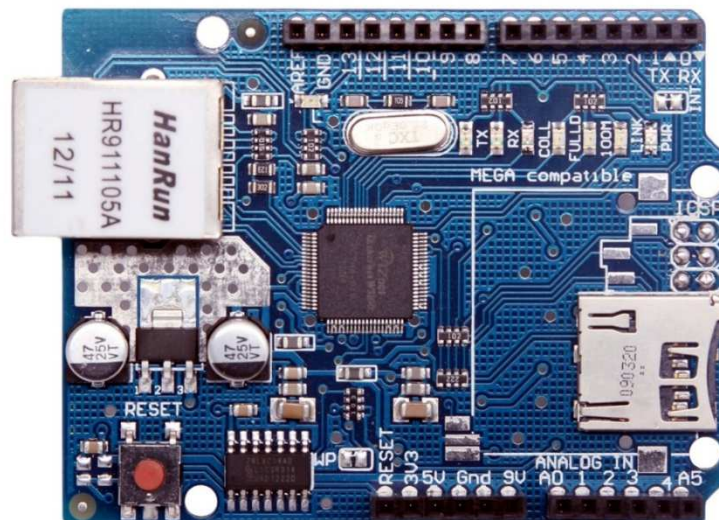
δηλαδή το ίδιο σύστημα που διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων.

Στη περίπτωση της εφαρμογής μας δεσμεύονται κάποια pins από την Ethernet shield για τις ανάγκες επικοινωνίας της η οποία κουμπώνει πάνω στο board του Arduino βλ. Κεφ 3.2.

Χαρακτηριστικά:

- Microcontroller: ATmega2560
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limit): 6-20V
- Digital I/O Pins: 54 (of which 15 provide PWM output)
- Analog Input Pins: 16
- DC Current per I/O Pin: 20mA
- DC Current for 3.3V Pin: 50mA
- Flash Memory: 256 KB of which 8 KB used by bootloader
- SRAM: 8KB
- EEPROM: 4KB
- Clock Speed: 16MHz

### 2.3 Ethernet Shield W5100



Εικόνα 2.3: Ethernet Shield W5100

Το shield αυτό συνδέεται απευθείας πάνω στον Arduino και χρησιμοποιεί τα ψηφιακά pins 4,10,50,51,52,53. Αυτές οι ακίδες δεν μπορούν να χρησιμοποιηθούν για γενική I/O. Το shield ενσωματώνει πάνω του και μια υποδοχή για SD κάρτες. Ο προγραμματιστής πρέπει να μεριμνήσει ώστε να μην έχουμε ταυτόχρονη χρήση της κάρτας δικτύου και της SD υποδοχής διότι χρησιμοποιούν κοινά pins.

Χαρακτηριστικά:

- χρήση ως διακομιστής ή πελάτης
- συνδέεται απευθείας στον Arduino
- ελεγκτής: Wiznet W5100.
- παρέχει μια υποδοχή κάρτας micro SD, η οποία μπορεί να χρησιμοποιηθεί για αποθήκευση αρχείων
- παρέχει πρωτόκολλο IP ικανό για TCP και UDP.
- υποστηρίζει έως και τέσσερις ταυτόχρονες συνδέσεις

## 2.4 Αισθητήρες

### 2.4.1 Ισθητήρας Φωτεινότητας LDR – GL5539

Το στοιχείο αυτό είναι μια φωτοαντίσταση. Ανάλογα με το φως που υπάρχει στον περιβάλλοντα χώρο μεταβάλλεται και η εσωτερική αντίστασή του. Έτσι μετρώντας αναλογικά αυτήν την αντίσταση μπορούμε να μετρήσουμε και τη φωτεινότητα του περιεχόμενου χώρου.



Εικόνα 2.4.1: LDR – GL5539

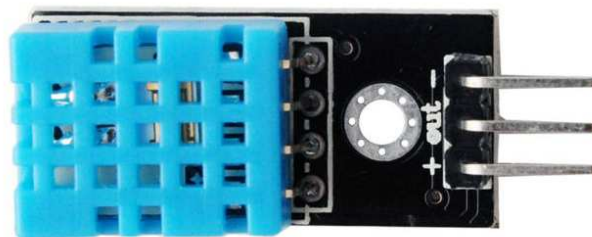


Χαρακτηριστικά:

- Μέγιστη τάση: (V-DC) 150V
- Μέγιστη κατανάλωση ενέργειας: 100mW
- Θερμοκρασία λειτουργίας: -30/+70°C
- Αντίσταση σε φωτεινότητα 10 lux: 30-90KΩ
- Αντίσταση στο σκοτάδι: 5MΩ

### 2.4.2 Αισθητήρας Θερμοκρασίας & Υγρασίας Χώρου (DHT-11)

Το DHT11 είναι μια βασικός, ψηφιακός αισθητήρας θερμοκρασίας και υγρασίας εξαιρετικά χαμηλού κόστους. Στέλνει ένα ψηφιακό σήμα δεδομένων σε κάποιο ψηφιακό pin. Είναι αρκετά απλό στην χρήση του, αλλά απαιτεί καθορισμένο χρονοδιάγραμμα για να λάβει τα δεδομένα. Ένα μειονέκτημα αυτού του αισθητήρα είναι ότι μπορούμε να πάρουμε νέα δεδομένα κάθε 2 δευτερόλεπτα.



Εικόνα 2.4.2: DHT-11

Χαρακτηριστικά:

- Εύρος μέτρησης υγρασίας: 20%~90%RH
- Εύρος μέτρησης Θερμοκρασίας: 0~60°C
- Ακρίβεια μέτρησης υγρασίας:  $\pm 5\%$ RH
- Ακρίβεια μέτρησης Θερμοκρασίας:  $\pm 2^\circ\text{C}$
- Χρόνος απόκρισης: <5sec
- Τάση λειτουργίας: 5V

### 2.4.3 Αισθητήρας Ποιότητας Αέρα MQ135

Ο αισθητήρας αυτός μπορεί να ανιχνεύει μια σειρά από αέρια από τα οποία τα περισσότερα είναι επιβλαβή για την υγεία όπως: αμμωνία, αρωματικές ενώσεις, θείο, βενζόλιο, καπνό, αλκοόλη. Έχει τη δυνατότητα αναλογικής και ψηφιακής εξόδου. Στην εφαρμογή θα χρησιμοποιήσουμε την αναλογική έξοδο του αισθητήρα (A0).



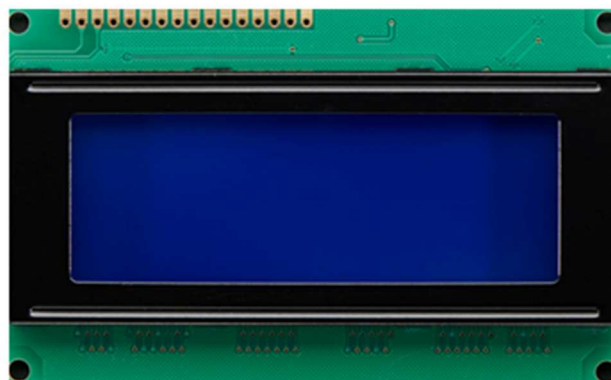
Εικόνα 2.4.3: Αισθητήρας ποιότητας αέρα MQ135

Χαρακτηριστικά:

- 2 διαφορετικές εξόδους (αναλογική & ψηφιακή)
- Τάση λειτουργίας: DC 5V

### 2.5 Οθόνη LCD

Μια απαραίτητη συσκευή για την άμεση πληροφόρησή μας είναι η οθόνη. Εκεί εμφανίζονται όλες οι πληροφορίες των μετρήσεων των αισθητήρων αλλά και απεικόνιση τυχών λαθών που μπορεί να υπάρξουν κατά τη λειτουργία της εφαρμογής.



Εικόνα 2.5.1: Οθόνη LCD.

Αυτή η οθόνη LCD επιλέχθηκε διότι έχει 4 γραμμές και 20 χαρακτήρες ανά γραμμή και μπορεί να εμφανίσει επαρκώς όλες τις τρέχουσες μετρήσεις των αισθητηρίων της εφαρμογής ταυτόχρονα. Τα λευκά εικονοστοιχεία της την κάνουν ευανάγνωστη από μεγάλη σχετικά απόσταση (η φωτεινότητα ρυθμίζεται εύκολα από έναν ροοστάτη). Η οθόνη επικοινωνεί με τον μικροελεγκτή βάση του σειριακού πρωτοκόλλου I<sup>2</sup>C, το οποίο κάνει πολύ εύκολη και απλή την σύνδεσή της με τον μικροελεγκτή αφού χρειάζονται μόνο 4 καλώδια.

Χαρακτηριστικά:

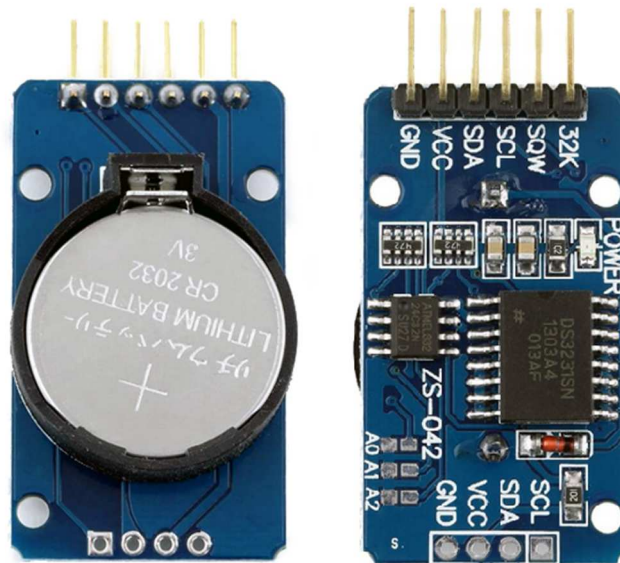
- Τάση λειτουργίας: (V-DC) 5V
- 4 γραμμές
- 20 χαρακτήρες ανά γραμμή
- Μπλε φόντο με λευκά εικονοστοιχεία
- Επικοινωνία με τον Arduino βάση του πρωτοκόλλου I<sup>2</sup>C (έχει ενσωματωμένο το I<sup>2</sup>C Serial Interface).



Εικόνα 2.5.2: I<sup>2</sup>C Serial Interface.

Το I<sup>2</sup>C είναι πρωτόκολλο μεταφοράς δεδομένων (0 ή 1), χρησιμοποιεί μόνο δύο καλώδια (τα οποία είναι αμφίδρομης κατεύθυνσης), τα SCL και SDA. Η γραμμή SCL είναι η γραμμή ρολογιού, ενώ η SDA είναι η γραμμή των δεδομένων. Τα δεδομένα μεταφέρονται σε ομάδες των 8 bit ξεκινώντας πάντα η μετάδοση με το MSB.

## 2.6 Real Time Clock (RTC) DS3231



Εικόνα 2.6: RTC DS3231.

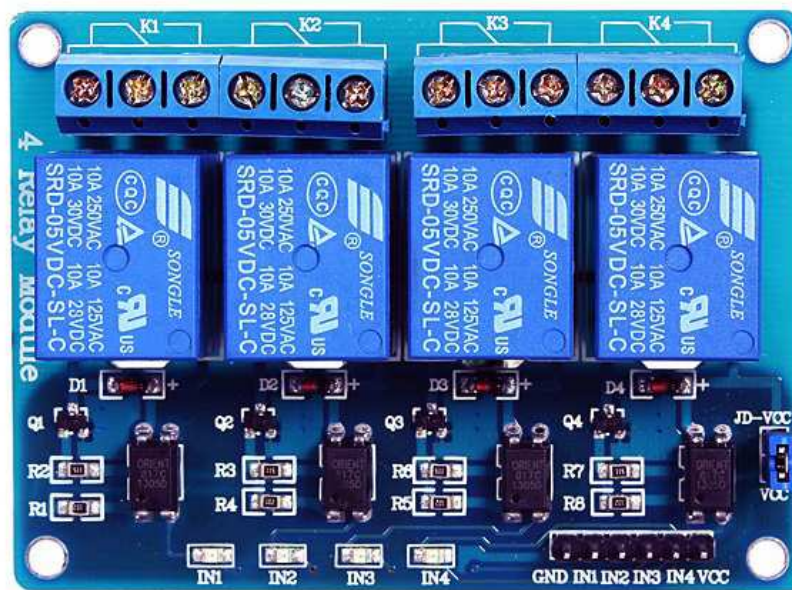
Το DS3231 είναι ένα χαμηλού κόστους, εξαιρετικά ακριβές I<sup>2</sup>C ρολόι πραγματικού χρόνου (RTC). Η συσκευή ενσωματώνει μπαταρία, έτσι και χωρίς τροφοδοσία διατηρεί ακριβή χρονομέτρηση. Υπολογίζει: δευτερόλεπτα, λεπτά, ώρες, ημέρα, μήνα και έτος. Το ρολόι λειτουργεί είτε με σύστημα 24 ωρών είτε με AM/PM της μορφής 12 ωρών. Τα δεδομένα μεταφέρονται σειριακά μέσω ενός διπλής κατεύθυνσης διαύλου I<sup>2</sup>C. Η τάση λειτουργίας του κυμαίνεται από 3,3 - 5V.

Η μπαταρία που δέχεται κανονικά είναι η LIR2032 η οποία είναι μια επαναφορτιζόμενη μπαταρία ιόντων λιθίου 3,6V, μπορεί εναλλακτικά όμως να δεχθεί και την απλή μπαταρία λιθίου CR2032 των 3V.

Παραπάνω αναφέρεται ότι κανονικά απαιτείται η χρήση μιας επαναφορτιζόμενης μπαταρίας διότι το ίδιο το κύκλωμα του DS3231 είναι έτσι σχεδιασμένο ώστε να φορτίζει την αναρτημένη μπαταρία κατά τη διάρκεια της λειτουργίας του, τροφοδοτούμενο από εξωτερική πηγή. Αν τώρα χρησιμοποιήσουμε την εναλλακτική της CR2032 θεωρητικά μετά από πολλές ώρες συνεχούς λειτουργίας του DS3231, τροφοδοτούμενο από εξωτερική πηγή μπορεί να καταστραφεί η μπαταρία.

## 2.7 4\_Relay Module

Για τον έλεγχο των περιφερειακών συσκευών δηλαδή της λάμπας, του θερμοπομπού, του αφυγραντήρα και του εξαεριστήρα οι οποίες δουλεύουν στα 220V, χρησιμοποιούμε ένα module το οποίο έχει 4 ανεξάρτητα ρελέ. Τροφοδοτείται από τάση 5V και μπορεί να οδηγήσει ανάλογα με την τάση στις εισόδους του IN1, IN2, IN3, IN4, τα αντίστοιχα ρελέ. Δίπλα ακριβώς στις εισόδους υπάρχουν και 4 κόκκινα led τα οποία σηματοδοτούν τον σπλισμό ή όχι της κάθε εξόδου.



Εικόνα 2.7: 4 Relay module.

Χαρακτηριστικά:

- 4 κανάλια
- Τάση λειτουργίας ρελέ: (V-DC) 5V
- Ρεύμα λειτουργίας ρελέ: 15-20mA
- Ρελέ υψηλής τάσης AC: 250V/10A
- DC: 30V/10A
- Οπτοαποζεύκτες εισόδου

## 2.8 Άλλα εξαρτήματα

### 2.8.1 Led's



Εικόνα 2.8.1: Led.

Στην κατασκευή χρησιμοποιούνται τα εξής led:

- λευκό, το οποίο όταν ανάψει προειδοποιεί για χαμηλό φωτισμό περιβάλλοντα χώρου (ταυτόχρονα ανάβει η λάμπα έκτακτης ανάγκης)
- κόκκινο, ανάβει όταν η θερμοκρασία πέσει κάτω από το καθορισμένο όριο (ταυτόχρονα ανάβει ο θερμοπομπός)
- μπλε, το οποίο ανάβει όταν η υγρασία ανέβει πάνω από το καθορισμένο όριο (ταυτόχρονα ανάβει ο αφυγραντήρας)
- κίτρινο, όταν η ποιότητα του αέρα δεν είναι ικανοποιητική, δηλαδή το ποσοστό των επιβλαβών αερίων έχει περάσει την προκαθορισμένη τιμή (ταυτόχρονα ανάβει ο εξαεριστήρας)

### 2.8.2 Buzzer

Μόλις δώσουμε τάση στον ένα ακροδέκτη του, ακούγεται το γνωστό «beep» για όση ώρα την εφαρμόσουμε στο buzzer. Στην εφαρμογή μας το χρησιμοποιούμε για να ειδοποιηθούμε και ακουστικά ότι:

- η θερμοκρασία είναι σε χαμηλά επίπεδα, κάτω του προκαθορισμένου ορίου
- η υγρασία είναι υψηλή, πάνω του προκαθορισμένου ορίου

Η χρήση του buzzer χρησιμεύει ως επί το πλείστον όταν λειτουργούμε την εφαρμογή μας σε manual mode. Τότε οι συνθήκες μέσα στο σπίτι μπορούν να ξεφύγουν αρκετά από τις καθορισμένες ως κανονικές. Όταν είμαστε σε auto mode συνθήκες ελέγχονται διαρκώς και διορθώνονται, άρα σε αυτή την περίπτωση δουλεύει μέχρι οι διορθώσεις των συνθηκών φτάσουν τα επιθυμητά επίπεδα.



Εικόνα 2.8.2: Buzzer.

Χαρακτηριστικά:

- Resonant frequency: 2.8kHz
- Operating current: 12mA
- Operating temperature: -20 - 60°C
- Diameter: 31mm
- Height: 10mm
- Operating voltage: 3 - 20V DC
- Mounting holes pitch: 39mm
- Sound level: 90dB (d=0.3 m)
- Resonant frequency tolerance:  $\pm 500\text{Hz}$

### 2.8.3 Button

Μόλις πατηθεί το κουμπί, βραχυκυκλώνει τους 2 ακροδέκτες του. Στην εφαρμογή μας το χρησιμοποιούμε για να ενεργοποιήσουμε ή να απενεργοποιήσουμε την οθόνη (η οθόνη απενεργοποιείται αυτόματα μετά από 5 λεπτά κανονικής λειτουργίας).

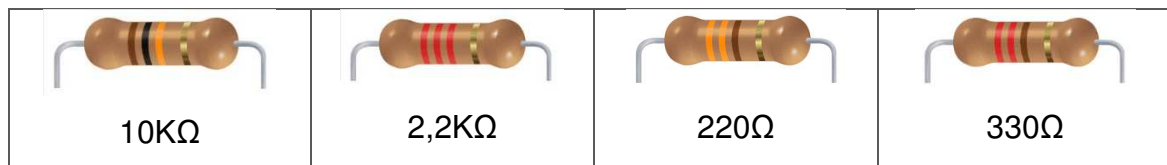


Εικόνα 2.8.3: Button.

## 2.8.4 Αντιστάσεις

Οι αντιστάσεις που χρησιμοποιήθηκαν στην κατασκευή μας για την προστασία των κυκλωμάτων είναι οι εξής:

Πίνακας 2.8: Αντιστάσεις.



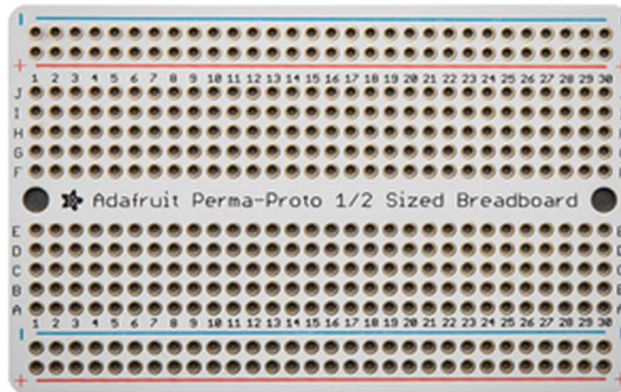
- 10KΩ (για τον αισθητήρα φωτεινότητας)
- 2,2KΩ (για το buzzer και το button)
- 220Ω (για το λευκό και το μπλε led)
- 330Ω (για το κόκκινο και το κίτρινο led)

## 2.8.5 PCB

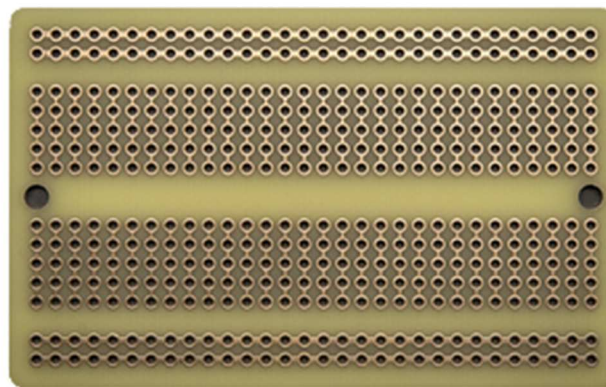
Μια πλακέτα τυπωμένου κυκλώματος (PCB) ηλεκτρικά συνδέει ηλεκτρονικά εξαρτήματα που χρησιμοποιούν αγώγιμα κομμάτια. Είναι χαραγμένα φύλλα χαλκού επάνω σε ένα μη αγώγιμο υπόστρωμα.

Στην κατασκευή χρησιμοποιήθηκε ένα ειδικό PCB διπλής όψης το οποίο έχει βραχυκυκλωμένα pins όπως ένα κοινό breadboard στην πίσω του πλευρά. Αυτό σημαίνει ότι και στην μπροστινή πλευρά οι κάθετα μεταξύ τους οπές στις δύο εσωτερικές ομάδες οπών είναι βραχυκυκλωμένες, όπως επίσης και στις δύο εξωτερικές ομάδες είναι οριζοντίως βραχυκυκλωμένες και χρησιμοποιούνται οι τελευταίες για τη γείωση και την τροφοδοσία. Αυτό κάνει ευκολότερη τη διαδικασία των κολλήσεων αφού χρειάζονται πολύ λιγότερες κολλήσεις και καλώδια για βραχυκυκλώσεις.





Εικόνα 2.8.5a: PCB (μπροστινή όψη).



Εικόνα 2.8.5b: PCB (πίσω όψη).

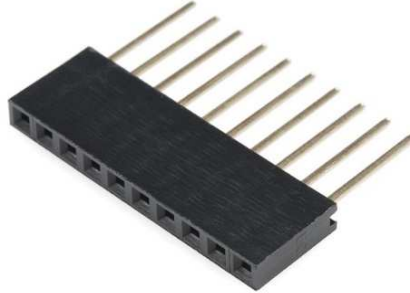
Χαρακτηριστικά:

- like half sized breadboard
- 30 rows of double 5-hole rows
- 4 power rails with positive/negative markings
- 82mm x 55mm
- 1.2mm / 0.047" drill holes
- 3.2mm mounting holes 1.9" apart

## 2.8.6 Pin Header

Για να συνδέσουμε τα αισθητήρια και το ρολόι πάνω στο PCB χωρίς να τα κολλήσουμε, θα χρειαστούμε 2 βάσεις των 10 ακίδων ώστε να κολληθούν αυτές ενώ τα παραπάνω εξαρτήματα να κουμπώνουν πάνω τους. Ένας άλλος

σημαντικός λόγος χρησιμοποίησης των pin headers είναι στην περίπτωση που τοποθετήσουμε την κατασκευή σε κουτί. Τότε είναι αναγκαίο τα αισθητήρια να μπορούν να μεταφερθούν με καλώδια έξω από το κουτί ώστε να παίρνουμε σωστές μετρήσεις.



Εικόνα 2.8.6: Pin Header.

Χαρακτηριστικά:

- Pitch: 2.54mm
- Plastic height: 8.5mm
- Pin Length: about 11mm
- Gender of Connector: Female
- Number of Pins: 10

### 2.8.7 Αποστάτες, βίδες

Ο Arduino, το PCB, η οθόνη LCD και το relay module τοποθετήθηκαν σε μια ξύλινη βάση με τη βοήθεια αποστατών και βιδών ώστε να είναι σταθερά και να μπορούν εύκολα να μετακινηθούν χωρίς κίνδυνο. Τα 10 χιλιοστά του ύψους των αποστατών είναι αρκετά για να χωράνε τα καλώδια βραχυκυκλωμάτων στο PCB αλλά και το I<sup>2</sup>C interface της LCD οθόνης.



Εικόνα 2.8.7α:Αποστάτης.

Χαρακτηριστικά αποστατών:

- type of spacer: screwed spacing bushing
- material: polyamide
- internal thread: M3
- spacer length: 10mm
- sleeve shape: hexagonal
- with internal threaded hole on both ends



Εικόνα 2.8.7b: Βίδες.

Χαρακτηριστικά βιδών:

- thread: M3
- length: 5mm
- type of cut: Phillips
- material: plastic

### 2.8.8 Τροφοδοτικό

Το τροφοδοτικό χρειάζεται για την τροφοδοσία των κυκλωμάτων με την κατάλληλη τάση και την κατάλληλη ένταση όταν δουλεύει ανεξάρτητα, δηλαδή όταν δεν είναι συνδεδεμένη μέσω του διαύλου usb σε κάποιον υπολογιστή.



Εικόνα 2.8.8: Τροφοδοτικό.

Χαρακτηριστικά:

- input voltage: 230V, 50Hz
- output voltage: 9V
- output current: 300mA

### 2.8.9 MicroSD

Η micro SD χρησιμοποιείται για την αποθήκευση όλων των αρχείων των μετρήσεων και των web σελίδων. Η κάρτα τοποθετείται στην Ethernet shield η οποία επιπλέον έχει και υποδοχή για SD κάρτα. Επιλέχθηκε κάρτα υψηλής ταχύτητας μεταφοράς δεδομένων class10 και χωρητικότητας 8GB η οποία είναι και η μεγαλύτερη που μπορεί να δεχθεί η ethernet shield.



Εικόνα 2.8.9: Micro SD

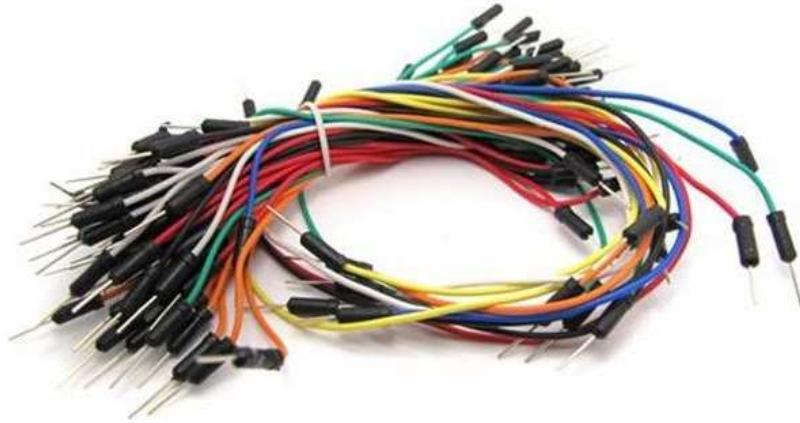
### 2.8.10 Καλώδια

Για τη σύνδεση της κατασκευής με το τοπικό δίκτυο χρησιμοποιούμε UTP cat5 καλώδιο, βλ. κεφ. 3.1.



Εικόνα 2.8.10α: UTP καλώδιο.

Για να συνδέσουμε τα pins του Arduino με αυτά των άλλων εξαρτημάτων και του PCB χρησιμοποιούμε απλά λεπτά και μικρά σε μήκος καλώδια σχεδιασμένα γι' αυτόν ακριβώς το σκοπό. Τα άκρα τους καταλήγουν είτε σε αρσενική είτε σε θηλυκή υποδοχή ανάλογα με τις ανάγκες μας.



Εικόνα 2.8.10b: Jumper Cables.

Χρησιμοποιούμε και ένα καλώδιο USB Type A/B για να μπορούμε να συνδέουμε τον Arduino με υπολογιστή ώστε να τον προγραμματίζουμε. Ταυτόχρονα μπορεί να χρησιμοποιηθεί και ως τροφοδοσία για τον Arduino.



Εικόνα 2.8.10c: USB Type A/B.

### 2.8.11 Ξύλινη Βάση

Απαραίτητη είναι και μία βάση ώστε εκεί να προσαρτηθούν όλα τα εξαρτήματα της κατασκευής. Αυτό το κάνουμε για να είναι όλα σταθερά και να μπορούμε να μετακινούμε την κατασκευή εύκολα και με ασφάλεια.

Για να χρησιμοποιηθεί η βάση χρειάστηκε λείανση με υαλόχαρτο κατά μήκος όλων των ακμών αλλά και στρογγυλοποίηση των γωνιών. Ακόμη για να βιδώσουμε πάνω τα εξαρτήματα έγιναν οι δεκατέσσερις απαραίτητες οπές διαμέτρου 3mm δηλαδή όσο είναι και οπές των εξαρτημάτων για τη στήριξή τους.



Εικόνα 2.8.11: Ξύλινη βάση.

Χαρακτηριστικά:

- υλικό: κόντρα πλακέ
- πάχος: 4mm
- διαστάσεις: 22,5 x 14,0cm

## ΚΕΦΑΛΑΙΟ 3

### ΣΥΝΔΕΣΜΟΛΟΓΙΑ ΕΞΑΡΤΗΜΑΤΩΝ ΚΑΙ ΑΙΣΘΗΤΗΡΙΩΝ

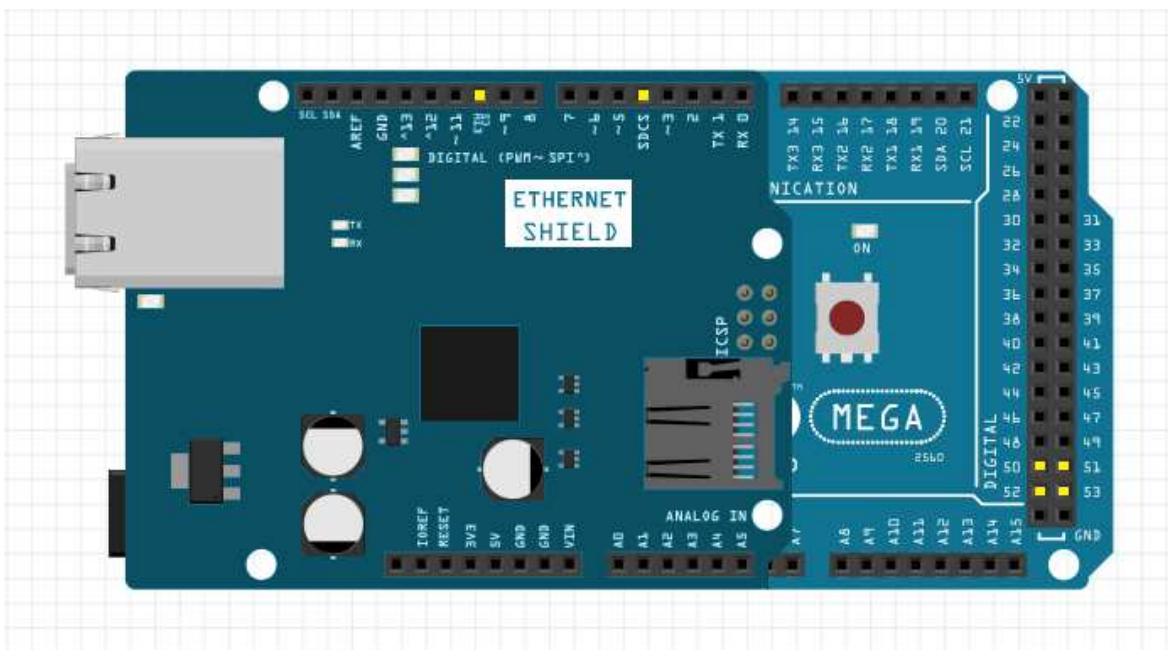
#### 3.1 Γενικά

Για την αναλυτική αναπαράσταση της συνδεσμολογίας των εξαρτημάτων με τον Arduino χρησιμοποιήθηκε το πρόγραμμα fritzing το οποίο είναι ελεύθερο λογισμικό, εύκολο στο χειρισμό και την επεξεργασία των κυκλωμάτων.

#### 3.2 Ethernet Shield

Κουμπώνει πάνω στον Arduino Mega και χρησιμοποιεί τα ψηφιακά pins 4,10,50,51,52,53 (στο σχήμα φαίνονται με κίτρινο χρώμα) όπως και η υποδοχή της SD κάρτας που είναι ενσωματωμένη πάνω της. Άρα αυτά τα pins πρέπει να μην χρησιμοποιηθούν από καμιά άλλη συσκευή της κατασκευής μας.

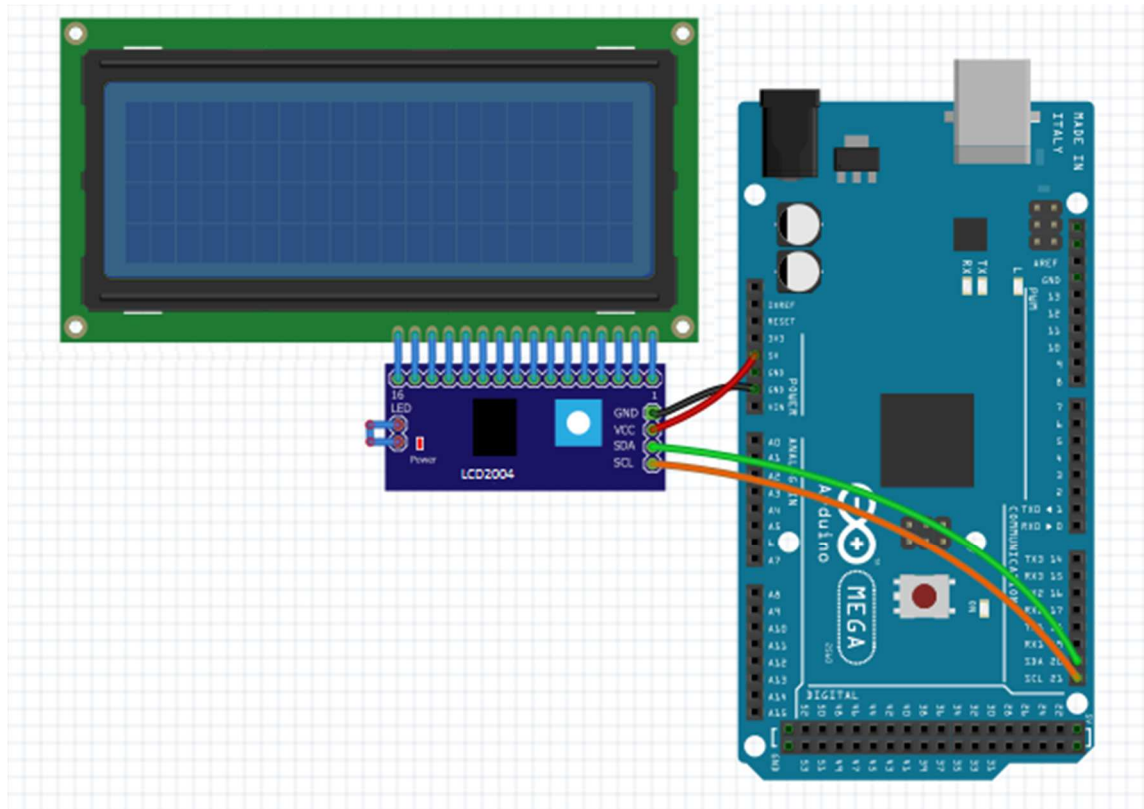
Για να συνδέσουμε την Ethernet shield με το δίκτυό μας (σε ένα switch ή σε έναν router) θα χρησιμοποιήσουμε ένα απλό straight UTP cat5 καλώδιο, αν το συνδέσουμε απευθείας σε έναν υπολογιστή τότε μπορούμε να χρησιμοποιήσουμε και καλώδιο με αντεστραμμένα ζεύγη (crossover).



Σχήμα 3.2: Συνδεσμολογία της Ethernet shield

### 3.3 Οθόνη LCD

Η οθόνη LCD επικοινωνεί με τον Arduino μέσω του πρωτοκόλλου I<sup>2</sup>C (έχει ενσωματωμένο το interface για I<sup>2</sup>C επικοινωνία). Έτσι χρειαζόμαστε μόνο 4 καλώδια για τη λειτουργία της: το SCL και το SDA τα οποία καταλήγουν στα αντίστοιχα pin του arduino, την τροφοδοσία και τη γείωση.

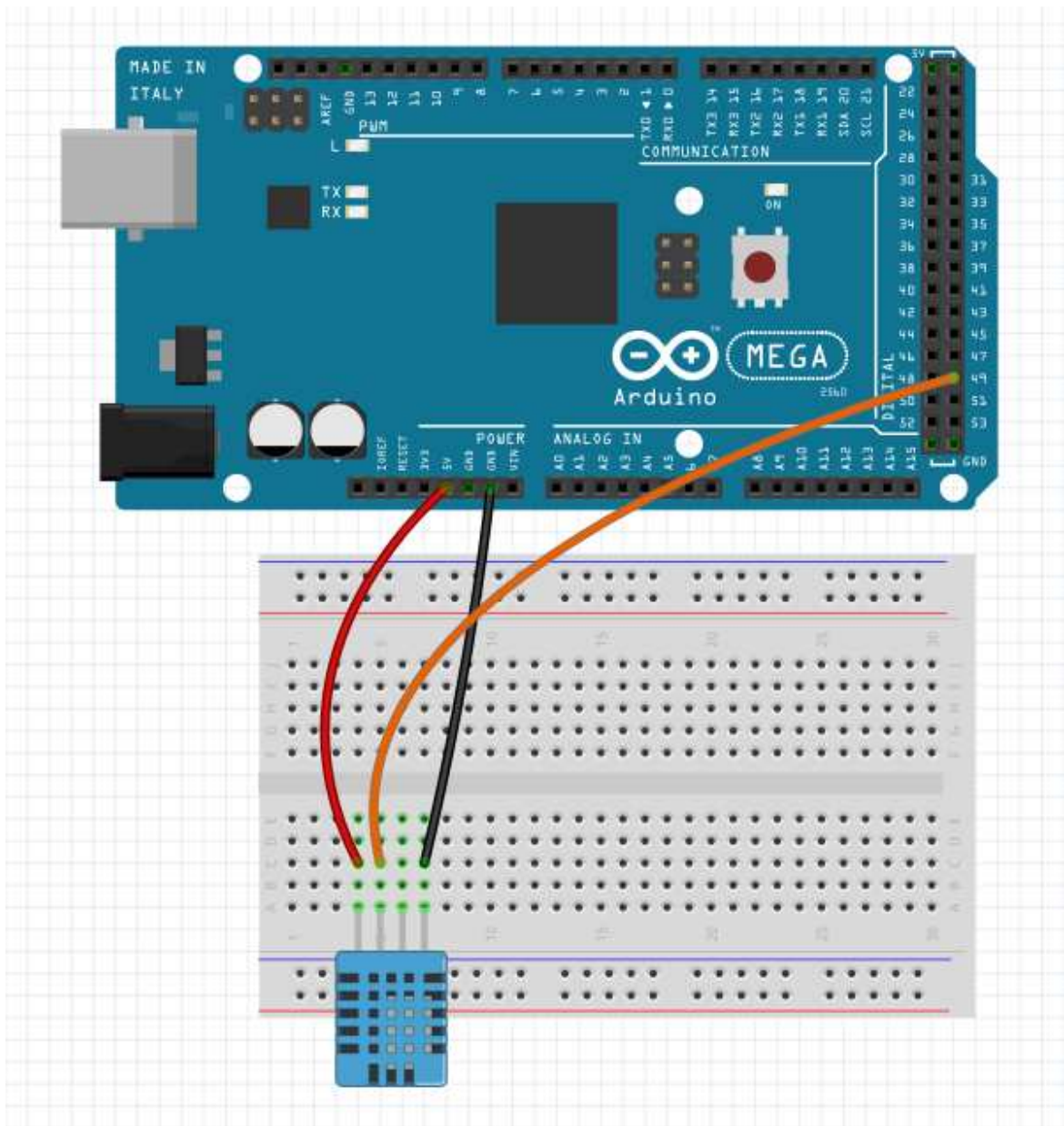


Σχήμα 3.3: Συνδεσμολογία της LCD οθόνης.



### 3.4 Αισθητήρας Θερμοκρασίας & Υγρασίας Χώρου (DHT-11)

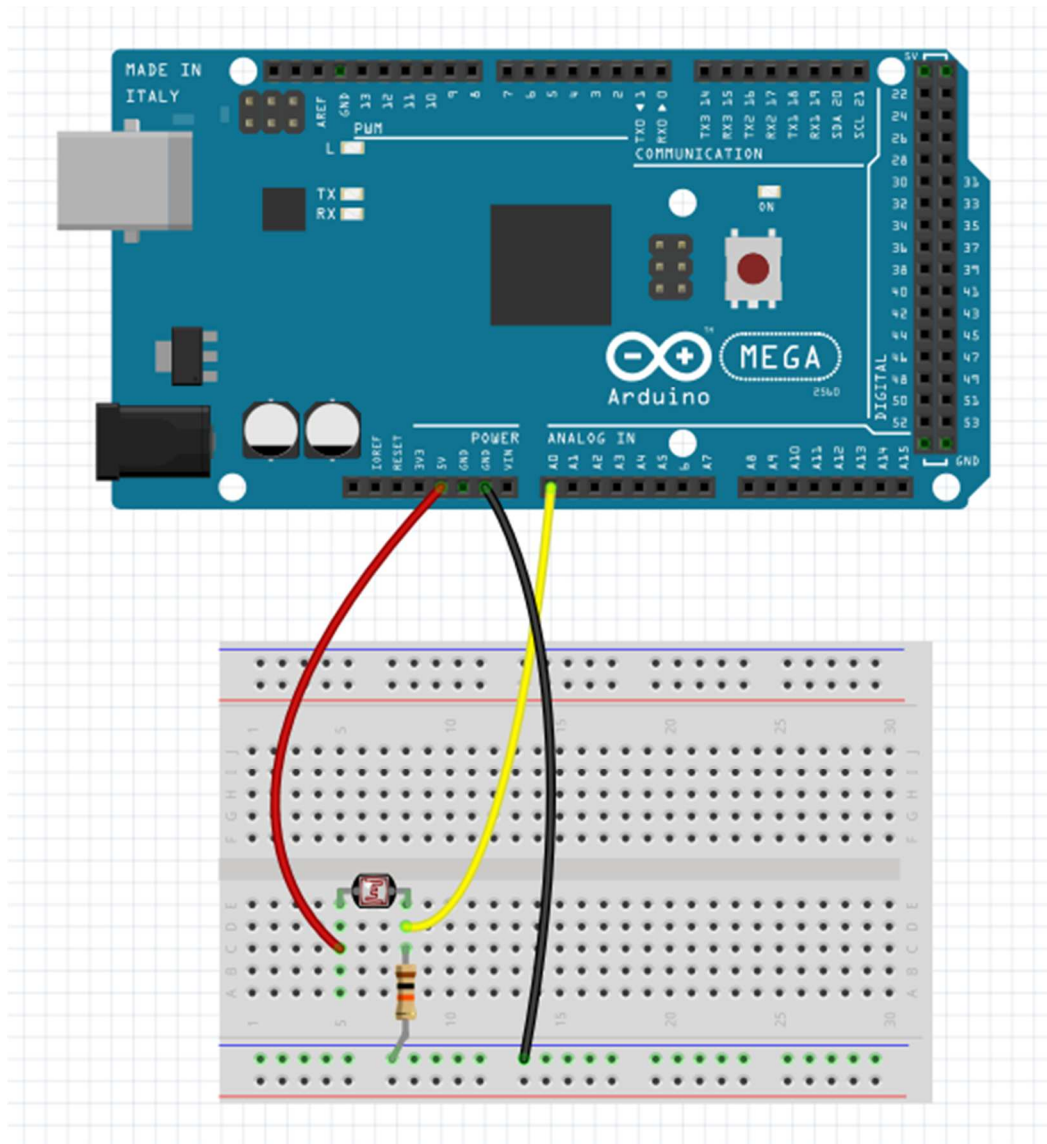
Ο διπλός αυτός αισθητήρας συνδέεται από τον ακροδέκτη του σήματος (DATA) με τον Arduino στο ψηφιακό pin 49, Επίσης στην τροφοδοσία και στη γείωση του Arduino από τους αντίστοιχους ακροδέκτες.



Σχήμα 3.4: Συνδεσμολογία του DHT-11.

### 3.5 Αισθητήρας Φωτεινότητας LDR – GL5539

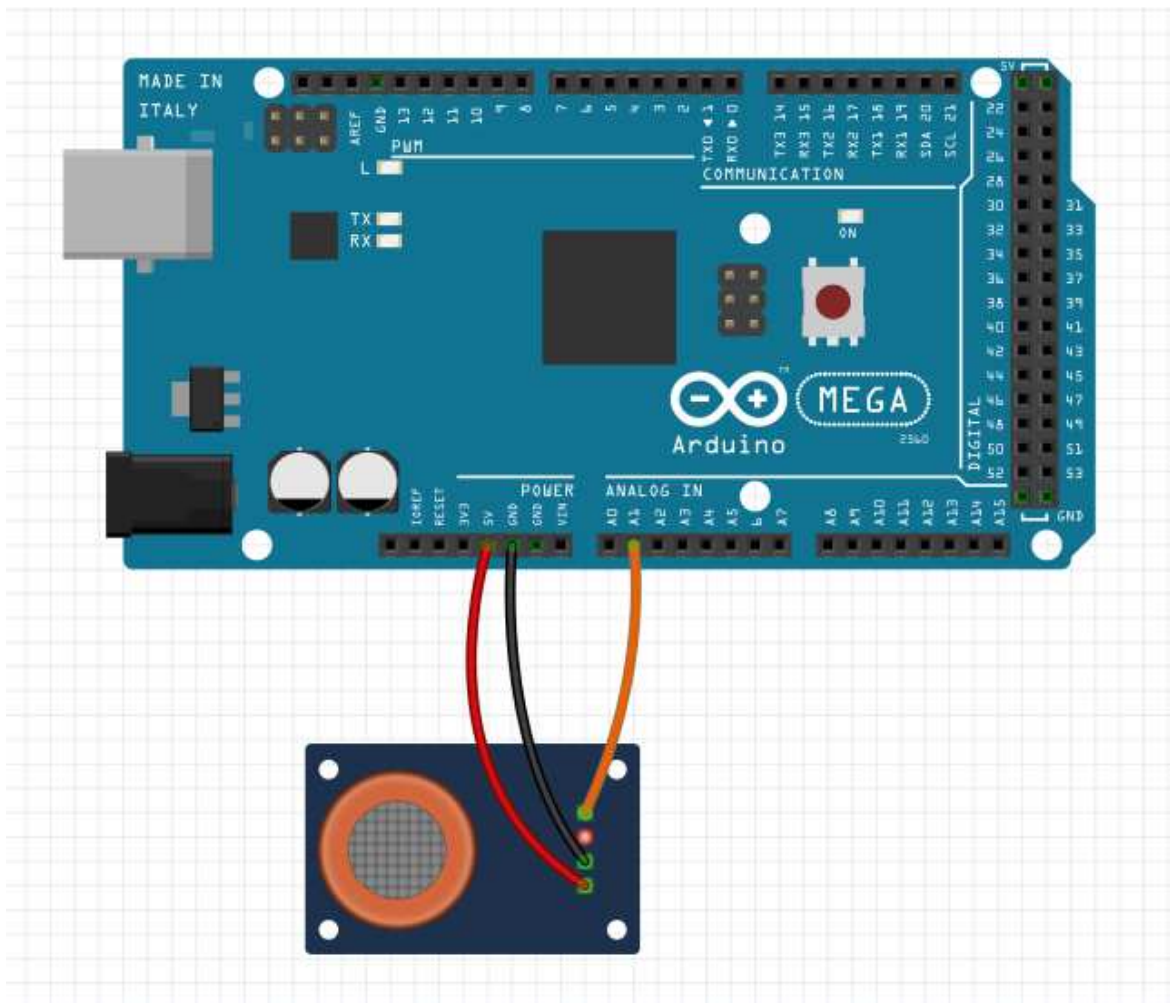
Χρησιμοποιούμε αντίσταση 10KΩ για την προστασία της φωτοαντίστασης. Ο ένας ακροδέκτης του αισθητήρα συνδέεται σε τάση 5V και ο άλλος συνδέεται με τον Arduino για να λάβουμε τις μετρήσεις στην αναλογική είσοδο A0 και μέσω της αντίστασης και στη γείωση.



Σχήμα 3.5: Συνδεσμολογία του GL5539.

### 3.6 Αισθητήρας Ποιότητα Αέρα MQ135

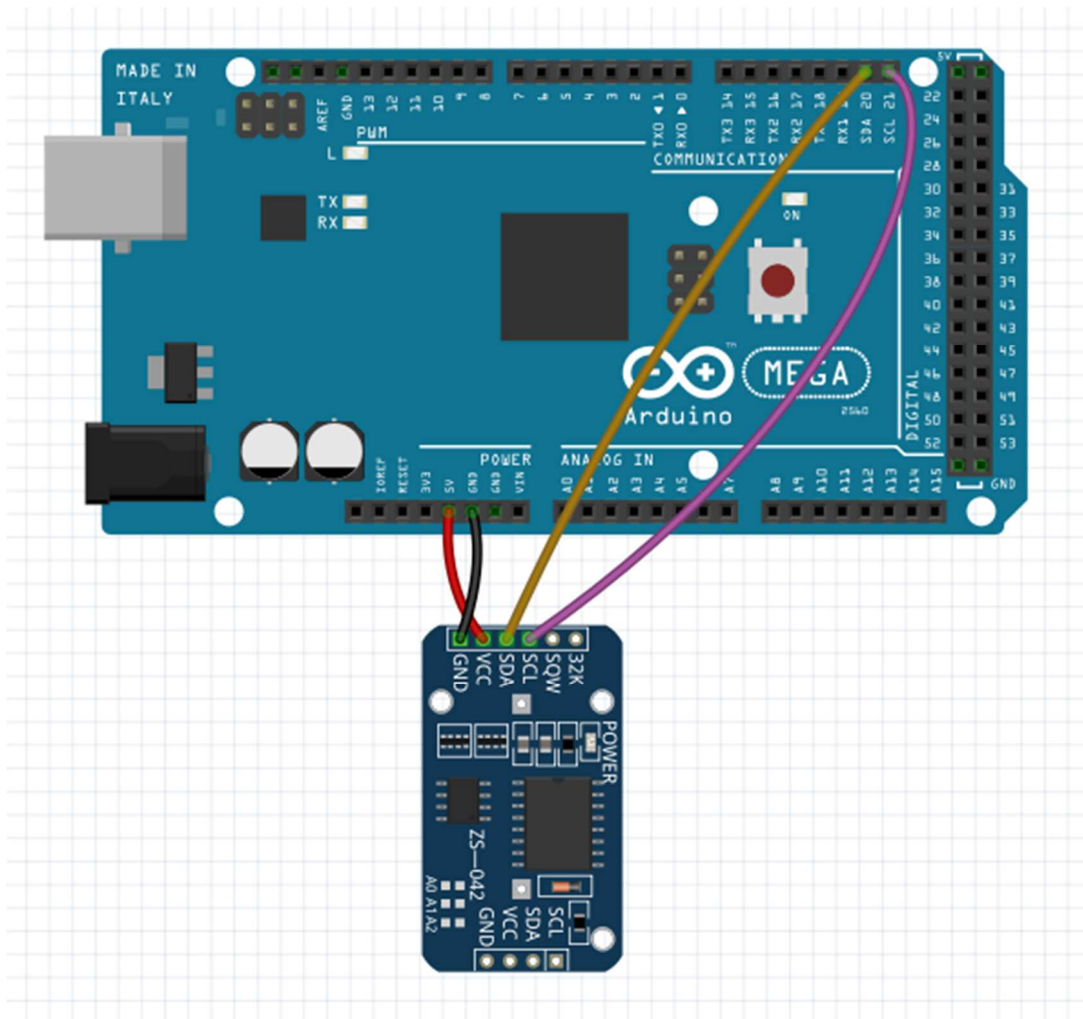
Χρησιμοποιούμε την αναλογική έξοδο της πλακέτας του αισθητηρίου MQ135 και την συνδέουμε στην αναλογική είσοδο A1 του Arduino. Επιπλέον χρειάζεται τροφοδοσία 5V και γείωση.



Σχήμα 3.6: Συνδεσμολογία του MQ135.

### 3.7 Real Time Clock DS3231

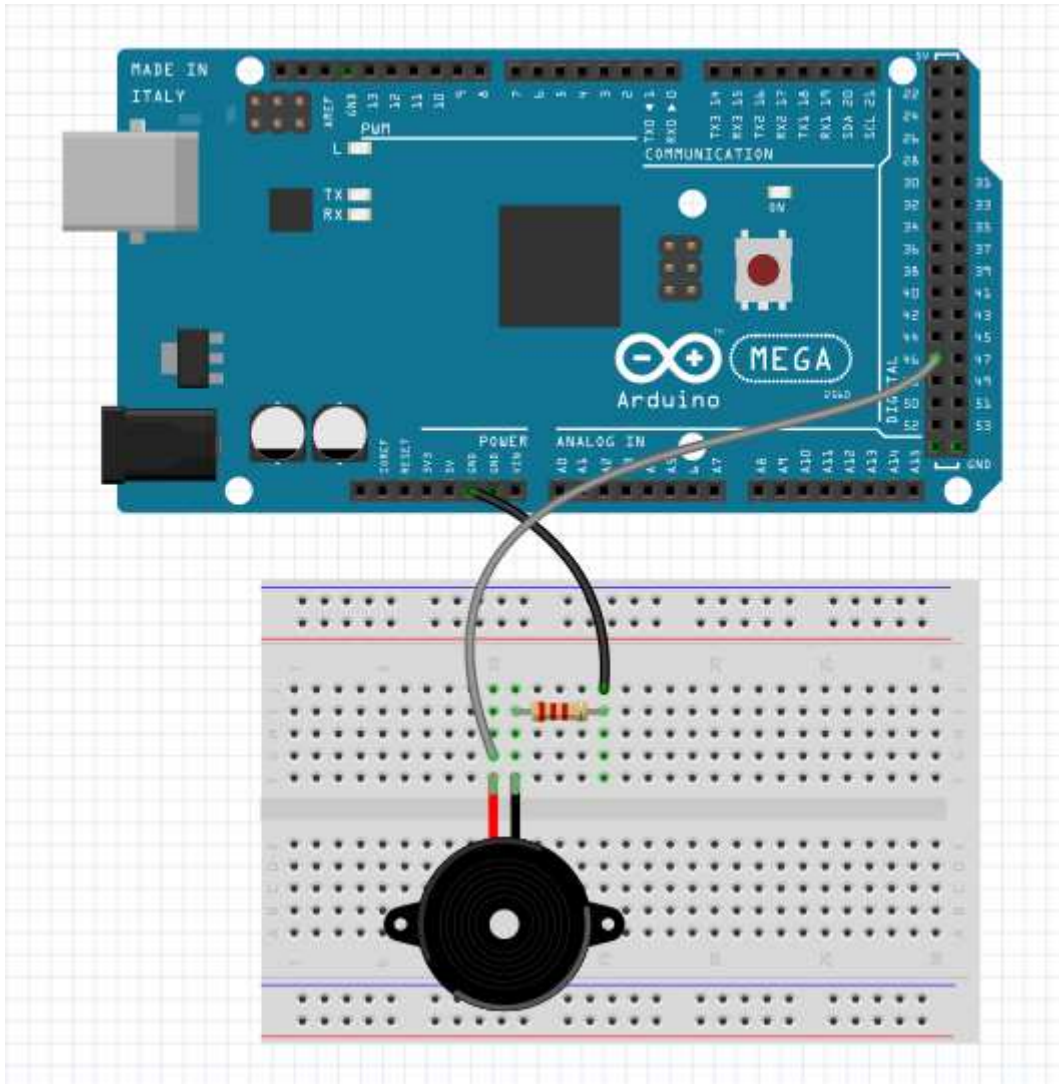
Το Real Time Clock DS3231 χρησιμοποιεί για την επικοινωνία του ένα I<sup>2</sup>C interface, δηλαδή χρειαζόμαστε μόνο 4 καλώδια για την σύνδεσή του. Συνδέουμε τα pin SCL και SDA στα αντίστοιχα pin του Arduino, καθώς επίσης την τροφοδοσία και τη γείωση.



Σχήμα 3.7: Συνδεσμολογία του DS3231.

### 3.8 Buzzer

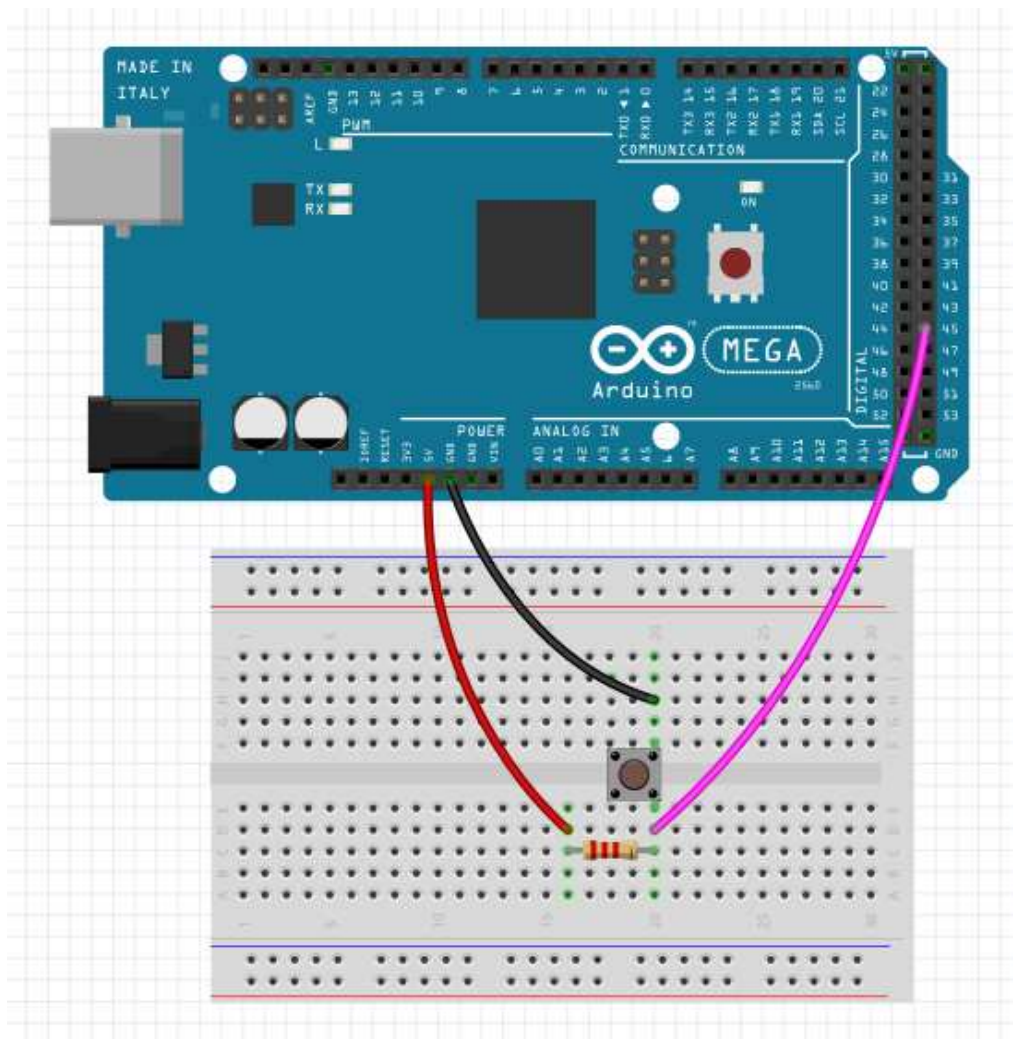
Συνδέουμε τον κόκκινο ακροδέκτη του βομβητή στην ψηφιακή έξοδο 46 του Arduino και τον μαύρο ακροδέκτη στη γείωση μέσω μιας αντίστασης 2,2 ΚΩ.



Σχήμα 3.8: Συνδεσμολογία του Buzzer.

### 3.9 Button

Συνδέουμε τον ένα ακροδέκτη στην ψηφιακό pin 45 και μετά στη τάση μέσω μιας αντίστασης 2,2KΩ, ενώ τον άλλο ακροδέκτη απευθείας στη γείωση.

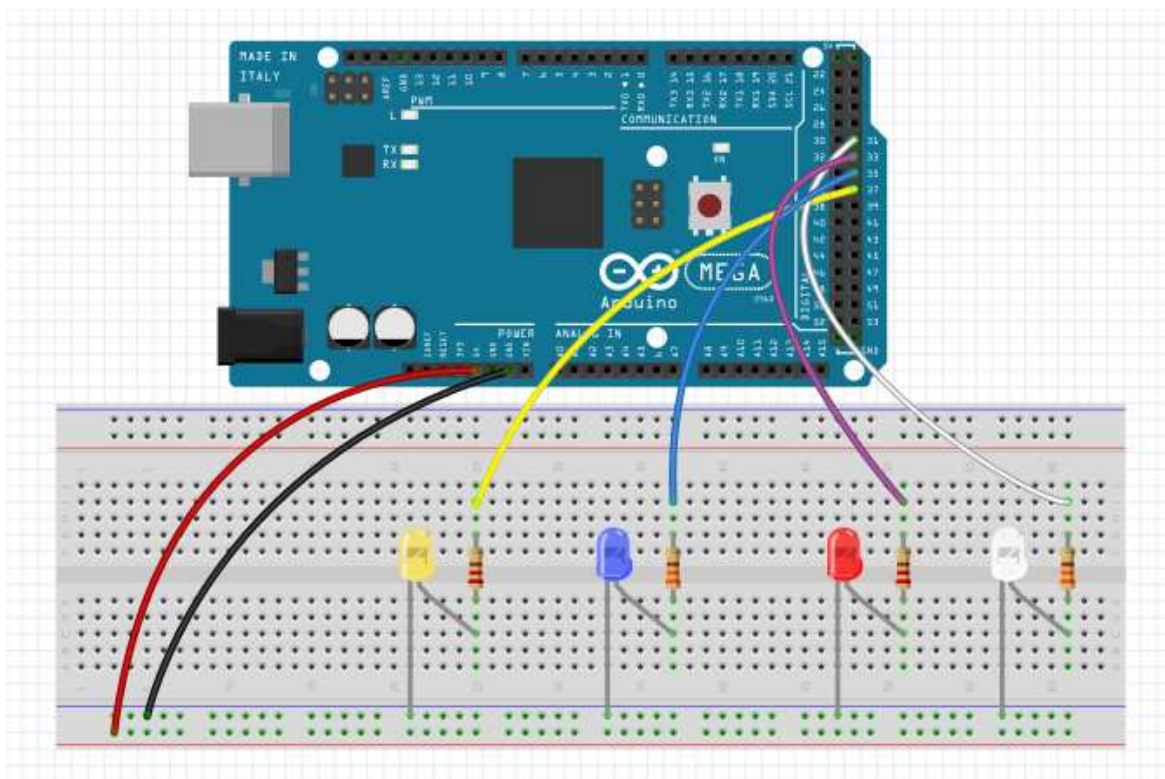


Σχήμα 3.9: Συνδεσμολογία του Button.

### 3.10 LED's

Για να έχουμε άμεση οπτική πληροφόρηση για το ποια περιφερειακή συσκευή δουλεύει χρησιμοποιούμε τέσσερα led διαφορετικού χρώματος:

- με το άσπρο led ειδοποιούμε για την ενεργοποίηση του φωτισμού έκτακτης ανάγκης.
- με το κόκκινο led ειδοποιούμε για την ενεργοποίηση του θερμοπομπού.
- με το μπλε led ειδοποιούμε για την ενεργοποίηση του θερμοπομπού.
- με το κίτρινο led ειδοποιούμε για την ενεργοποίηση του εξαεριστήρα.

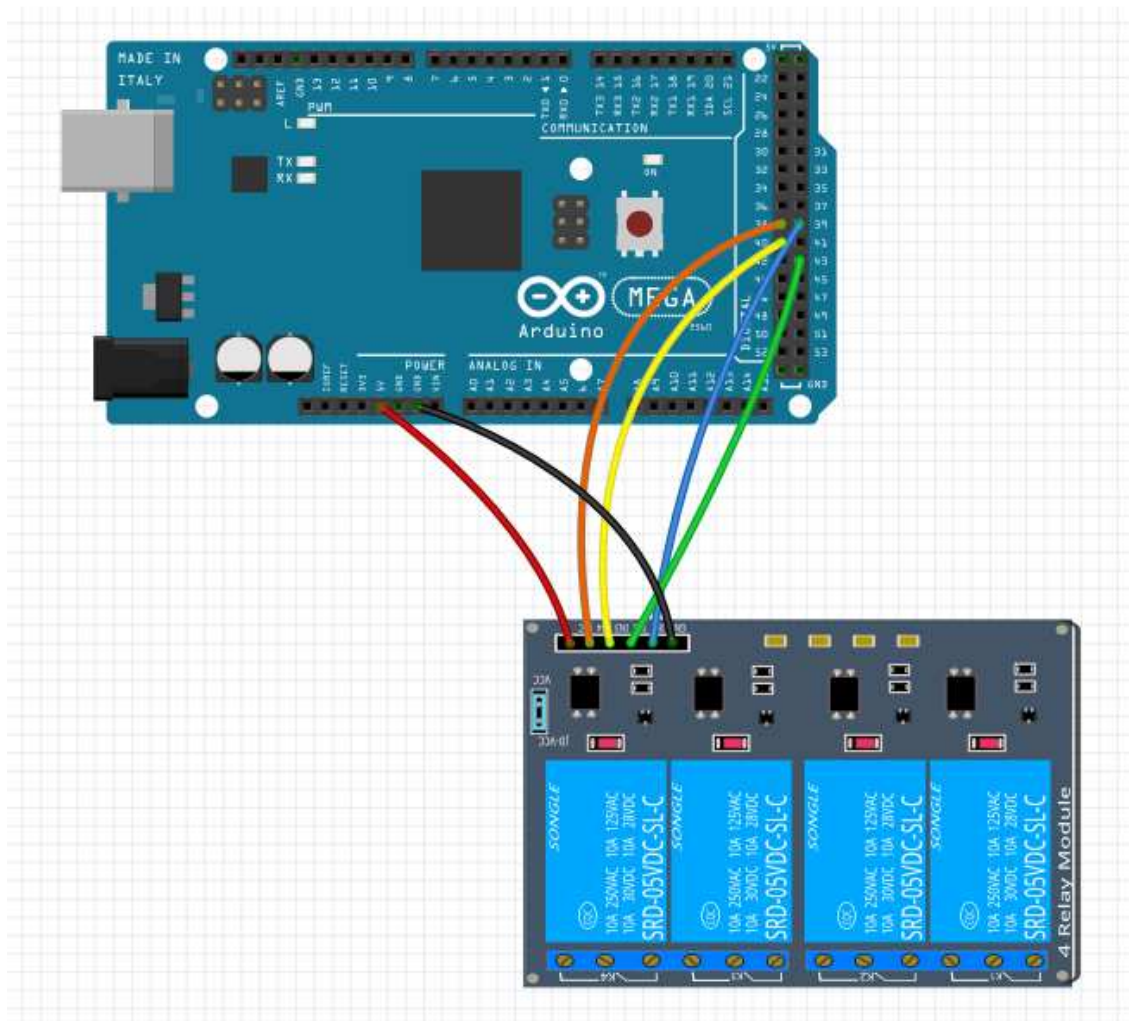


Σχήμα 3.10: Συνδεσμολογία των led.

### 3.11 4\_Relay Module

Για να μπορέσουμε να έχουμε τον έλεγχο λειτουργίας του εξαεριστήρα, του αφυγραντήρα, του θερμοπομπού και του φωτισμού έκτακτης ανάγκης οι οποίοι χρειάζονται 220V, θα πρέπει να χρησιμοποιήσουμε τα ρελέ του module για τις παρακάτω τέσσερις περιπτώσεις:

- 1<sup>ο</sup> ρελέ συνδεδεμένο με το pin 39 του Arduino για τον έλεγχο του φωτισμού έκτακτης ανάγκης.
- 2<sup>ο</sup> ρελέ συνδεδεμένο με το pin 43 του Arduino για τον έλεγχο του θερμοπομπού.
- 3<sup>ο</sup> ρελέ συνδεδεμένο με το pin 40 του Arduino για τον έλεγχο του αφυγραντήρα.
- 4<sup>ο</sup> ρελέ συνδεδεμένο με το pin 38 του Arduino για τον έλεγχο του εξαεριστήρα.



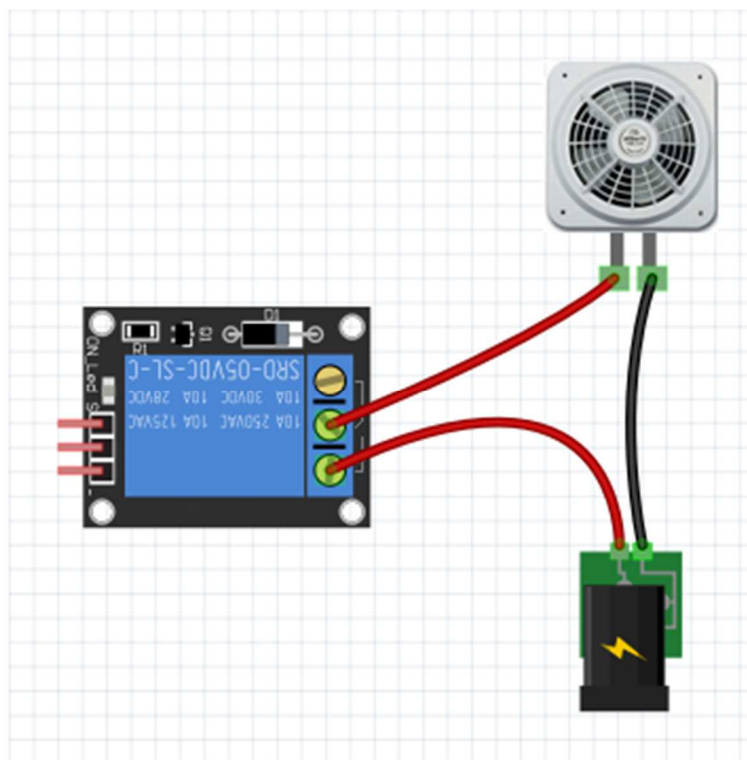
Σχήμα 3.11: Συνδεσμολογία του 4\_relay module.



Τα ρελέ οπλίζουν στέλνοντας στις εισόδους του module μηδενική τάση. Άρα όταν θέλουμε μια περιφερειακή συσκευή να είναι αναμμένη εφαρμόζουμε 0V, ενώ όταν θέλουμε να σβήσει 5V. Η κατασκευή έχει τη δυνατότητα αλλάζοντας τα καλώδια που συνδέουν εξ' ορισμού τα ρελέ με τον Arduino και συνδέοντας άλλα που ενώνονται με τους ακροδέκτες των led, να αλλάξουμε τη συμπεριφορά της όπλισης των ρελέ κάνοντάς τα να οπλίζουν εφαρμόζοντας τάση 5V.

### 3.12 Σύνδεση περιφερειακών συσκευών

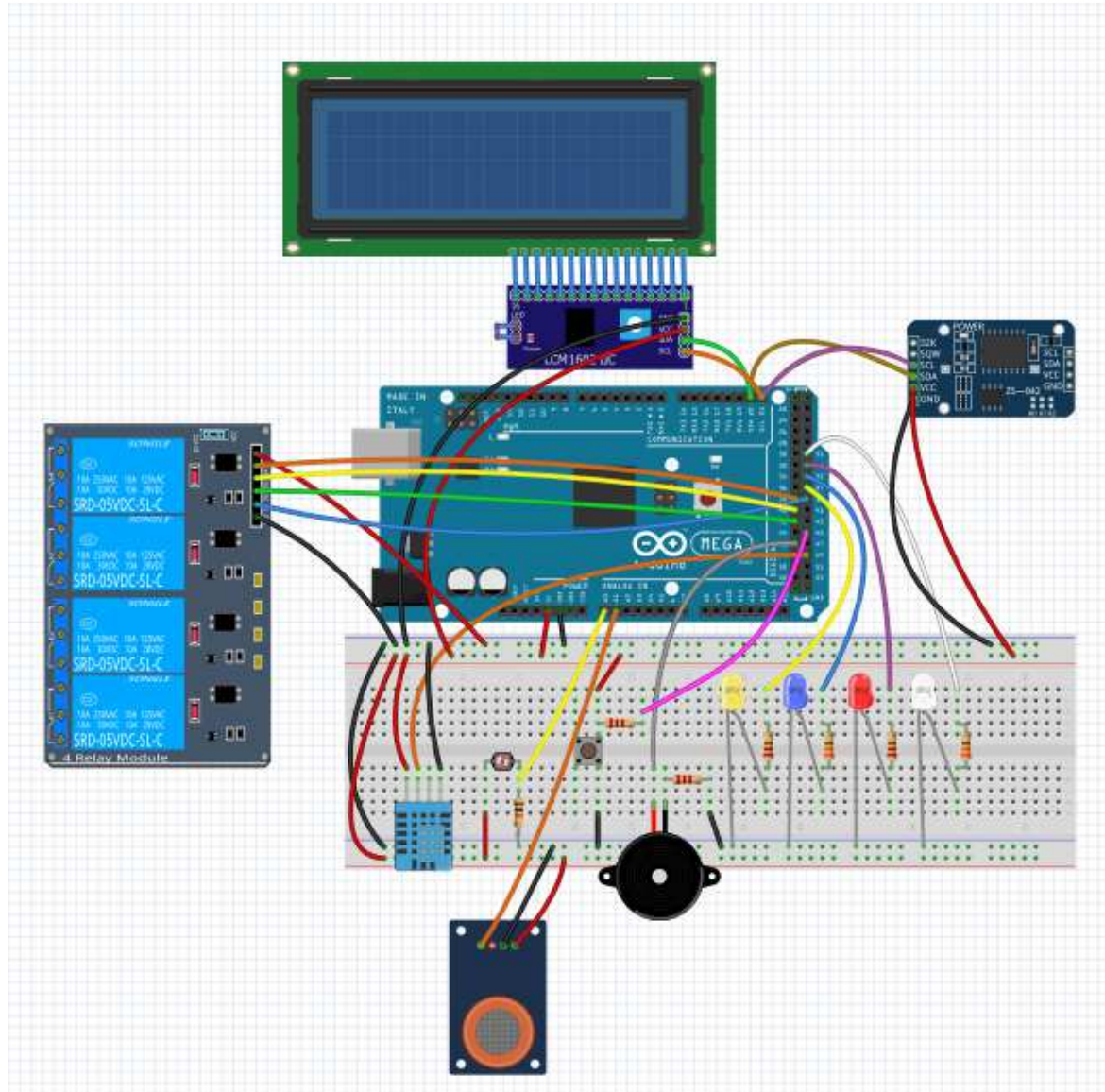
Η συνδεσμολογία ώστε να τοποθετήσουμε μια περιφερειακή συσκευή όπως η λάμπα είναι η εξής: Η συσκευή είναι συνδεδεμένη στη γείωση και στην μεσαία επαφή του ρελέ. Μόλις γειωθεί ο ακροδέκτης του ρελέ από τον Arduino, αμέσως το ρελέ βραχυκυκλώνει τη μεσαία επαφή με την αριστερή με αποτέλεσμα η περιφερειακή συσκευή να τροφοδοτείται με ρεύμα 220V. Με αυτόν τον τρόπο ελέγχουμε και τις τέσσερις περιφερειακές συσκευές μας. Η συνδεσμολογία αυτή είναι normal open.



Σχήμα 3.12: Σύνδεση περιφερειακών συσκευών.

### 3.13 Όλη η κατασκευή

Στο παρακάτω σχήμα φαίνεται η συνδεσμολογία όλων των εξαρτημάτων της κατασκευής:



Σχήμα 3.13: Συνδεσμολογία όλων των εξαρτημάτων.

## ΚΕΦΑΛΑΙΟ 4

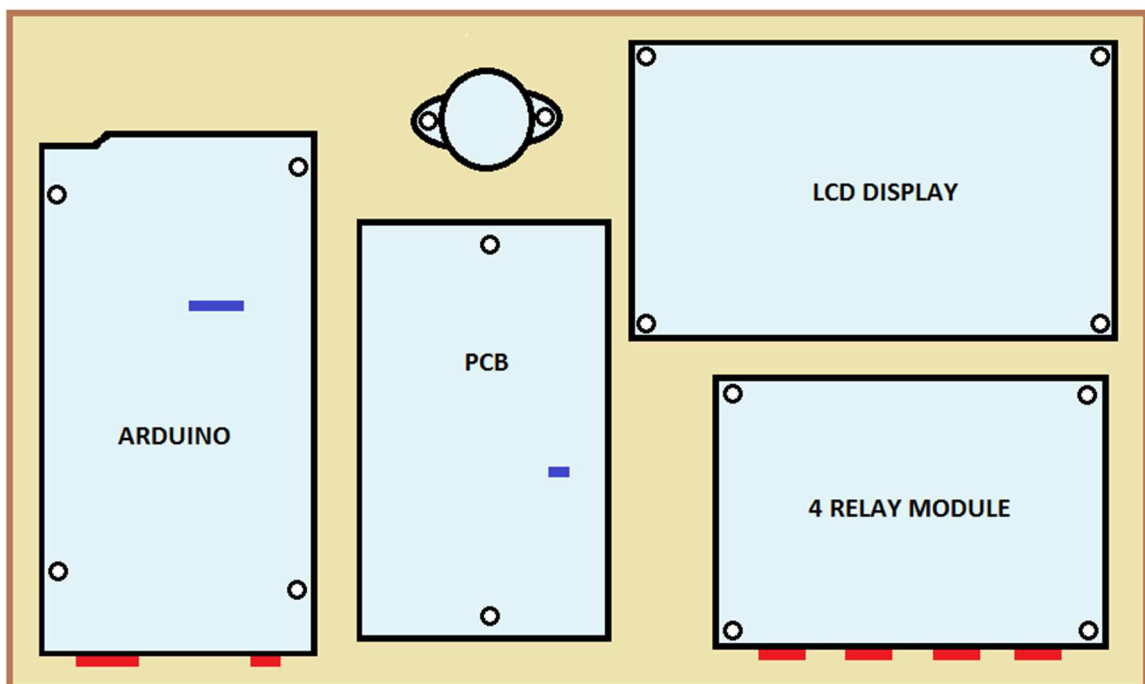
### Η ΚΑΤΑΣΚΕΥΗ & Η ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

#### 4.1 Η κατασκευή

Για να δομηθεί η κατασκευή σε ένα ενιαίο κομμάτι, όλα τα εξαρτήματα πρέπει να προσαρτηθούν πάνω σε μία ξύλινη βάση. Για να γίνει αυτό ορίζονται τα εξής κριτήρια:

- όλες οι απαιτούμενες εξωτερικές συνδέσεις (κόκκινα σημεία) να είναι μόνο από μία πλευρά της κατασκευής
- όσο το δυνατόν μικρότερο συνολικό μέγεθος των διαστάσεων της βάσης
- όσο το δυνατόν μικρότερο μέγεθος καλωδίων για τη σύνδεση των κυρίων εξαρτημάτων μεταξύ τους

Αυτό επιτυγχάνεται με την τοπολογία του παρακάτω σχήματος. Όλες οι απαιτούμενες εξωτερικές συνδέσεις δηλαδή η τροφοδοσία, η σύνδεση σε δίκτυο και οι συνδέσεις των ρελέ με τις περιφερειακές συσκευές γίνονται μόνο από μία πλευρά της βάσης με αποτέλεσμα να έχουμε εύκολη τοποθέτηση της βάσης, ακόμα και ανάρτησή της σε τοίχο, αλλά και μια ενιαία διαφυγή καλωδίων από την κατασκευή.



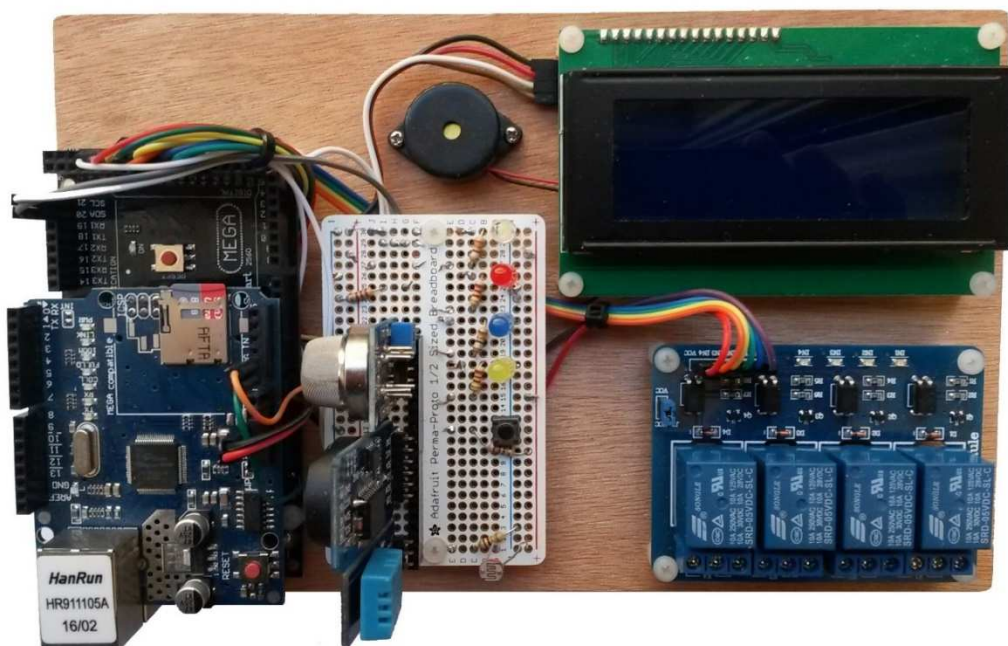
Σχήμα 4.1: Το πλάνο της κατασκευής.

Επίσης υπάρχουν άλλα δύο σημεία στα οποία πρέπει να έχουμε εύκολη πρόσβαση (μπλε σημεία):

- Το πρώτο είναι η κάρτα micro SD η οποία όποτε χρειάζεται να αναδιαμορφώσουμε τα αρχεία που περιέχει πρέπει να αφαιρεθεί ώστε να συνδεθεί σε κάποιον υπολογιστή και αφού είναι έτοιμη να τοποθετηθεί ξανά στη θέση της. Σε αυτό το σημείο το μόνο που μπορούμε να προβλέψουμε αφού η υποδοχή της micro SD είναι προσαρτημένη στην Ethernet shield, είναι να χρησιμοποιήσουμε μόνο τα ψηφιακά pins 22-49 τα οποία βρίσκονται στο δεξιό άκρο του Arduino ώστε τα καλώδια να μην περνάνε κοντά από την υποδοχή.
- Το δεύτερο είναι το κουμπί που είναι τοποθετημένο πάνω στο PCB και είναι υπεύθυνο για την ενεργοποίηση & απενεργοποίηση της οθόνης. Τοποθετήθηκε σε ένα ελεύθερο σημείο γενικά πάνω στην πλακέτα.

Ακόμα με την παρακάτω τοποθέτηση των υλικών εκμεταλλευόμαστε στον μεγαλύτερο βαθμό που θα μπορούσαμε το χώρο, με συνέπεια η βάση να είναι αρκετά μικρή δε διαστάσεις.

Η εικονιζόμενη τοποθέτηση των κύριων εξαρτημάτων πετυχαίνει μικρότερα σε μήκος καλώδια για τη σύνδεση μεταξύ τους, λαμβάνοντας υπ' όψη τα σημεία απ' όπου φεύγουν τα καλώδια από το κάθε κύριο εξάρτημα αλλά και τη θέση που καταλήγουν πάνω στο PCB.



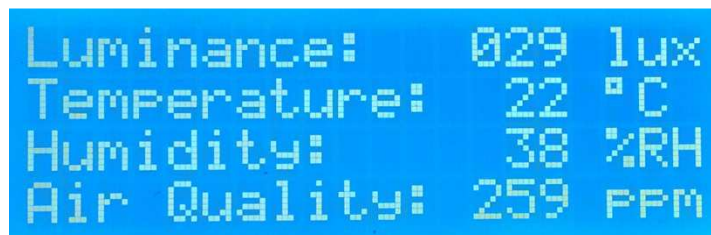
Εικόνα 4.1: Η κατασκευή.

## 4.2 Η λειτουργία του συστήματος

### 4.2.1 Γενικά

Αναλυτικά τα βήματα για τη λειτουργία της κατασκευής έχει ως εξής:

- Αν δεν έχουμε φορτώσει τον κώδικα στον Arduino, φορτώνουμε τον κώδικα για τον προγραμματισμό του. Ο κώδικας αποτελείται από αρκετά ξεχωριστά αρχεία τα οποία καλεί το κύριο .ino αρχείο. Αυτό γίνεται με τη βοήθεια του USB Type A/B καλωδίου το οποίο συνδέει τον Arduino με τον υπολογιστή που έχει τα αρχεία. Ο κώδικας φορτώνεται και τρέχει στον Arduino αν δεν υπάρχουν λάθη. Για τις επόμενες φορές που θα επανεκκινήσει ο Arduino δεν χρειάζεται επαναπρογραμματισμό ούτε φυσικά σύνδεση με ηλεκτρονικό υπολογιστή.
- Για την web λειτουργία της εφαρμογής μας πρέπει να έχουμε αποθηκεύσει και τα κατάλληλα αρχεία στην κάρτα micro SD η οποία είναι ενσωματωμένη στο Ethernet shield. (σε περίπτωση που δεν υπάρχει κάρτα δεν προχωράει το πρόγραμμα παρακάτω, είναι προαπαιτούμενη αρχικοποίηση).
- Ξεκινάει αμέσως η λειτουργία της εφαρμογής και διαβάζουμε στην οθόνη τις αρχικοποιήσεις των συσκευών και αφού ολοκληρωθούν αυτές, τις τρέχουσες τιμές των μετρήσεων που έχουμε λάβει από τα αισθητήρια, ανανεωμένες διαρκώς κάθε 30 δευτερόλεπτα.



```
Luminance:    029 lux
Temperature:  22 °C
Humidity:     38 %RH
Air Quality:  259 PPM
```

Εικόνα 4.2.1: Απεικόνιση των τελευταίων μετρήσεων.

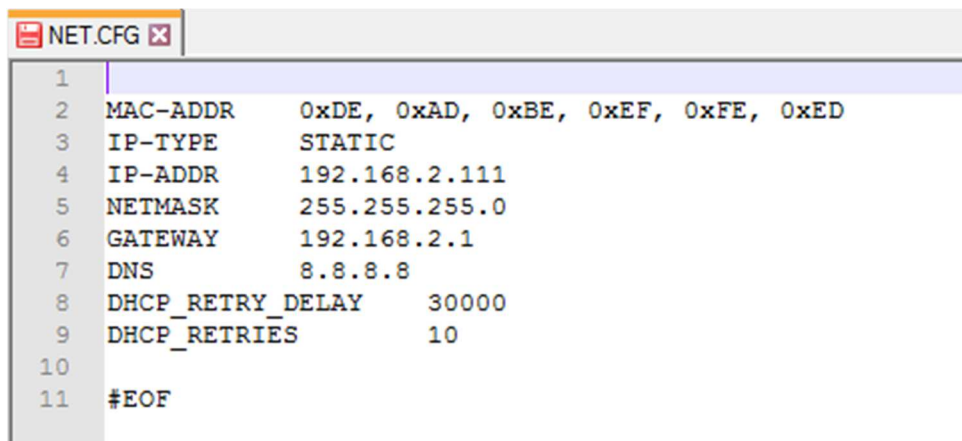
- Σε περίπτωση που κάποια ή και οι δύο από τις τιμές των μετρήσεων θερμοκρασίας και υγρασίας έχουν τιμές πέρα από τα όρια που έχουμε θέσει, ακούγεται από τον βομβητή ένας στιγμιαίος διακριτικός βόμβος κάθε 30 δευτερόλεπτα για να μας ειδοποιήσει ηχητικά ότι κάποια ή κάποιες από τις ελεγχόμενες συνθήκες στον περιβάλλοντα χώρο έχει βγει εκτός ορίων.

- Η οθόνη σβήνει αυτόματα μετά από 5 λεπτά. Για να την ενεργοποιήσουμε ξανά για το ίδιο χρονικό διάστημα πρέπει να πιέσουμε το κουμπί το οποίο είναι πάνω στο pcb.
- Το πρόγραμμα (όταν βρισκόμαστε σε Auto mode, βλ. κεφ. 5.2) ελέγχει συνέχεια τις μετρήσεις και όπου χρειάζεται επεμβαίνει και ανάβει ή σβήνει τις περιφερειακές συσκευές ώστε να μην ξεπερνιούνται τα όρια που έχουμε θέσει.
- Για να έχουμε τον έλεγχο των συνθηκών θα πρέπει να έχουμε συνδέσει το Ethernet shield με έναν ηλεκτρονικό υπολογιστή ή με το τοπικό δίκτυο μέσω του κατάλληλου UTP καλωδίου. Πρόσβαση θα έχουμε στο web κομμάτι της εφαρμογής από οποιονδήποτε φυλλομετρητή απλά πληκτρολογώντας στο πεδίο διευθύνσεων την ip διεύθυνση του συστήματός μας όταν έχουμε συνδέσει τον Arduino απευθείας με έναν ηλεκτρονικό υπολογιστή είτε πρόκειται σε πρόσβαση μέσω δικτύου από οποιονδήποτε ηλεκτρονικό υπολογιστή, κινητό ή tablet τα οποία θα βρίσκονται συνδεδεμένα στο ίδιο δίκτυο.
- Μόλις συνδεθούμε με το web κομμάτι της εφαρμογής εμφανίζεται ένα login παράθυρο. Για να έχουμε πρόσβαση στην web εφαρμογή πρέπει να πληκτρολογήσουμε σωστά το username και το password.
- Από το web κομμάτι της εφαρμογής μπορούμε να αλλάξουμε τα όρια των συνθηκών.
- Βασική ακόμα λειτουργία του προγράμματος είναι ότι μπορούμε να θέσουμε τον έλεγχο σε manual mode, δηλαδή ανεξάρτητα από τις τιμές που έχουμε θέσει ως όρια (οι οποίες αγνοούνται), να ανοιγοκλείνουμε κατά βούληση χειροκίνητα τις περιφερειακές συσκευές, έχοντας φυσικά και πρόχειρες τις τρέχουσες τιμές των συνθηκών.
- Στο web κομμάτι της εφαρμογής μπορούμε επίσης να ρυθμίσουμε εκ νέου την ώρα του ρολογιού μας, σε περίπτωση που δεν ταυτίζεται με την τρέχουσα ώρα.
- Στη δεύτερη σελίδα της εφαρμογής μπορούμε να δούμε γραφικά την πορεία όλων των μετρήσεων (γράφημα γραμμών) αλλά και την τρέχουσα τιμή τους (μετρητές - gauges).

### 4.2.2 Αρχεία της micro SD κάρτας

Στην SD κάρτα υπάρχει ένας φάκελος «config» ο οποίος περιέχει 3 αρχεία πολύ χρήσιμα για τη διαμόρφωση του δικτύου, για την σύνδεση στην εφαρμογή (login), και τέλος για τη διαχείριση των αρχείων των δεδομένων που παράγονται.

- το αρχείο net.cfg στο οποίο καθορίζουμε αν η ip του συστήματός μας θα είναι στατική ή μέσω της υπηρεσίας DHCP, αν είναι στατική καθορίζουμε την διεύθυνση και τη μάσκα, δίνουμε τις διευθύνσεις του DNS και του κεντρικού δρομολογητή και τέλος καθορίζουμε το χρόνο που θα περιμένει να αποδοθεί μια διεύθυνση μέσω της υπηρεσίας DHCP καθώς και πόσες φορές να το κάνει αυτό. Τέλος εκεί ορίζουμε και την MAC address της κάρτας δικτύου.



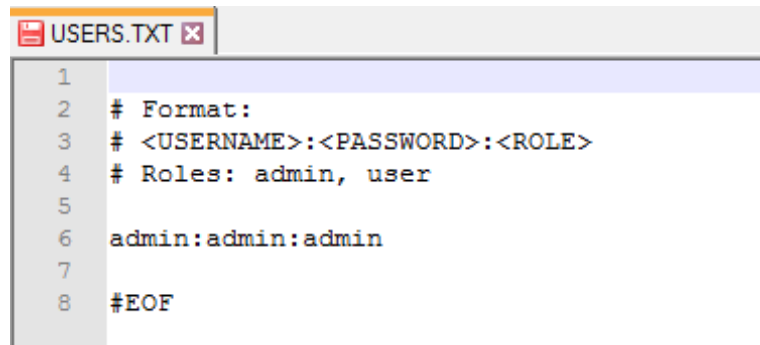
```

1
2 MAC-ADDR      0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
3 IP-TYPE       STATIC
4 IP-ADDR       192.168.2.111
5 NETMASK       255.255.255.0
6 GATEWAY       192.168.2.1
7 DNS           8.8.8.8
8 DHCP_RETRY_DELAY 30000
9 DHCP_RETRIES  10
10
11 #EOF

```

Εικόνα 4.2.2a: Το αρχείο NET.CFG.

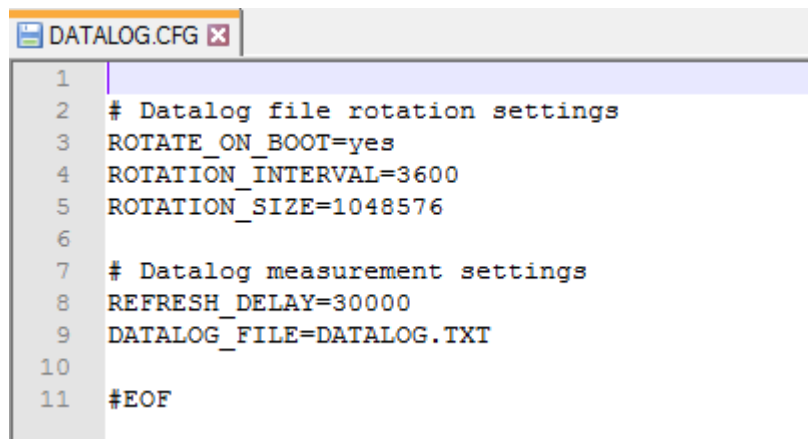
- το αρχείο users.txt στο οποίο περιέχεται το username το password αλλά και ο ρόλος του συνδεδεμένου χρήστη (admin ή user). Για την είσοδο στο web κομμάτι της εφαρμογής το username και το password φαίνονται στην Εικόνα 4.2.2. Η ιδιότητα του ρόλου δεν χρησιμοποιείται στην εφαρμογή. Σε περίπτωση που χρειαστούμε να δώσουμε ρόλο user σε ένα συνδεδεμένο χρήστη θα μπορούσαμε εύκολα να περιορίσουμε τις δυνατότητες απομακρυσμένου ελέγχου και να περιορίσουμε την πρόσβασή του μόνο στην πληροφόρηση, με πολύ μικρές αλλά αναγκαίες επεμβάσεις στον κώδικα.



```
1  
2 # Format:  
3 # <USERNAME>:<PASSWORD>:<ROLE>  
4 # Roles: admin, user  
5  
6 admin:admin:admin  
7  
8 #EOF
```

Εικόνα 4.2.2b: Το αρχείο USERS.TXT.

- το αρχείο datalog.cfg το οποίο έχει τις παραμέτρους για τη διαχείριση των αρχείων των μετρήσεων, όπως αν θα αλλάζει αρχείο εγγραφής κατά το boot ή κάθε πόσα δεύτερα θα κάνει αυτή την αλλαγή.



```
1  
2 # Datalog file rotation settings  
3 ROTATE_ON_BOOT=yes  
4 ROTATION_INTERVAL=3600  
5 ROTATION_SIZE=1048576  
6  
7 # Datalog measurement settings  
8 REFRESH_DELAY=30000  
9 DATALOG_FILE=DATALOG.TXT  
10  
11 #EOF
```

Εικόνα 4.2.2c: Το αρχείο DATALOG.CFG.

Στον φάκελο «web» της κάρτας SD υπάρχουν:

- τα δύο html αρχεία της εφαρμογής (index.html και chart.html)
- το logo αρχείο εικόνας (smho-logo.png)
- το favicon εικονίδιο (favicon.ico)
- ένα απαραίτητο αρχείο javascript (gauge.min.js)

Ο φάκελος «data» είναι αρχικά κενός, αλλά μόλις η εφαρμογή εκτελεστεί, εκεί αποθηκεύονται όλες οι μετρήσεις που λαμβάνουμε από τα αισθητήρια σε μορφή αρχείων .txt.



## ΚΕΦΑΛΑΙΟ 5

### ΤΟ ΠΕΡΙΒΑΛΛΟΝ WEB ΤΗΣ ΕΦΑΡΜΟΓΗΣ

#### 5.1 Γενικά

Το web περιβάλλον της εφαρμογής αποτελείται από 2 σελίδες:

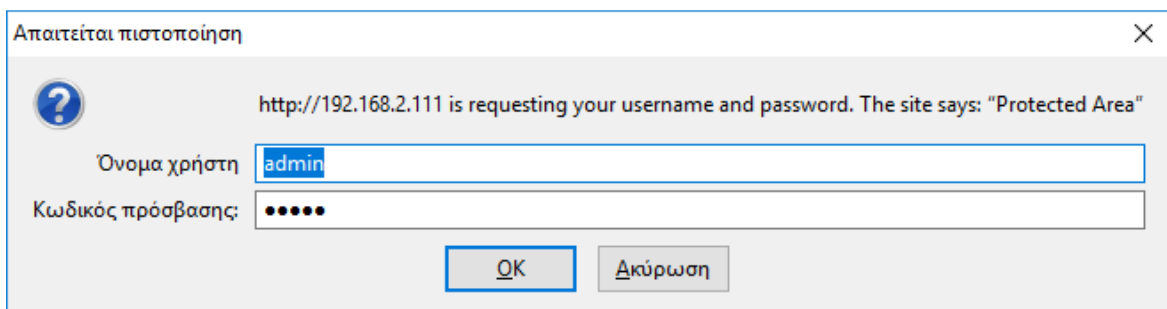
- την κύρια και αρχική σελίδα (index.htm) η οποία περιέχει όλες αριθμητικά τις μετρήσεις, τις επιλογές ελέγχου των συνθηκών και τις ρυθμίσεις της ώρας και της ημερομηνίας
- την σελίδα γραφικής απεικόνισης των μετρήσεων (chart.htm)

Τα δύο αυτά αρχεία βρίσκονται στην SD κάρτα στο φάκελο «web», στο Ethernet Shield της εφαρμογής.

Ακόμα μέσα σε αυτόν το φάκελο υπάρχουν διάφορα αρχεία χρήσιμα για την εμφάνιση των σελίδων, όπως και το αρχείο favicon.ico το οποίο είναι το εικονίδιο που εμφανίζεται δίπλα στον τίτλο της σελίδας του φυλλομετρητή, το αντίστοιχο αρχείο για το logo των σελίδων.

#### 5.2 Σύνδεση (login)

Για να μπορέσουμε να έχουμε πρόσβαση στο web κομμάτι της εφαρμογής χρειάζεται να κάνουμε την απαραίτητη σύνδεση δίνοντας το κατάλληλο username και password. Αν όντως εισαχθούν σωστά εμφανίζεται η αρχική σελίδα (index.html) και έχουμε πρόσβαση σε όλες τις λειτουργίες της.

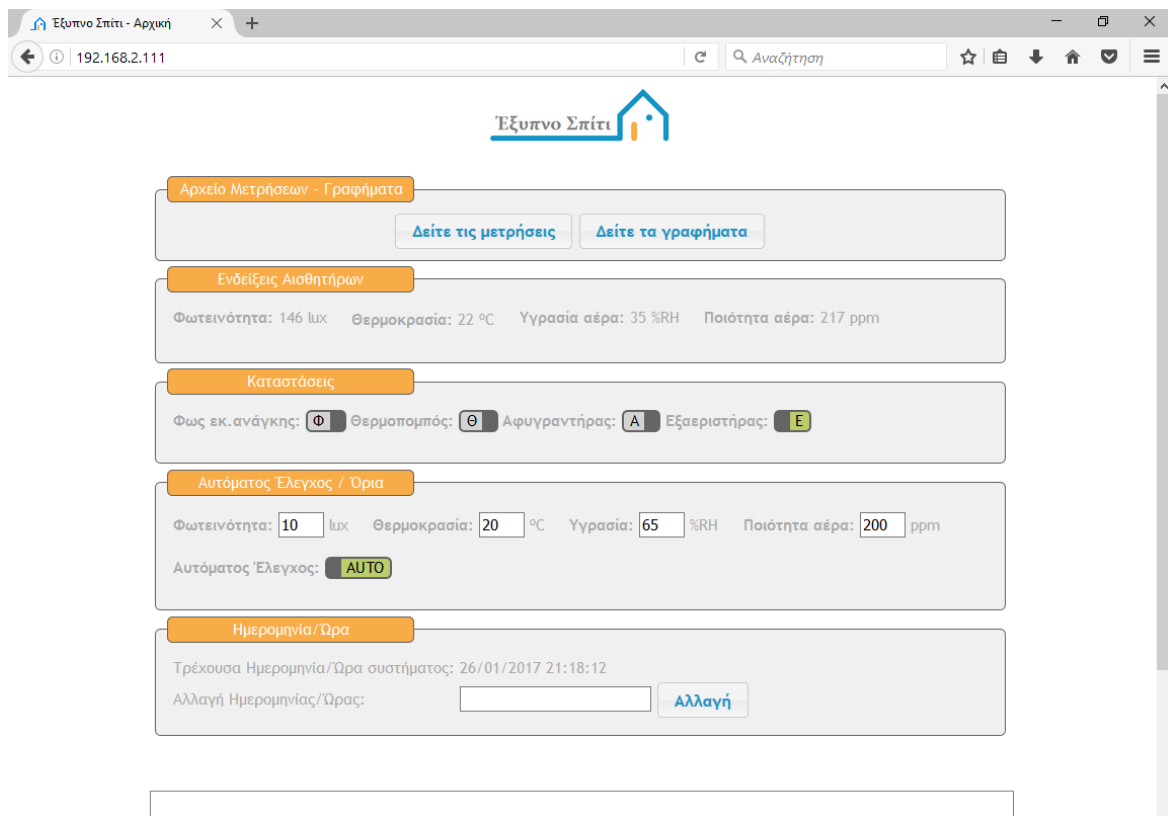


Εικόνα 5.2: login.

### 5.3 Αυτόματος έλεγχος των συνθηκών

Σε αυτή την περίπτωση στο πλαίσιο «Αυτόματος Έλεγχος / Όρια» η επιλογή «ΑΥΤΟ» πρέπει να είναι ενεργοποιημένη. Μόνος του ο μικροελεγκτής θα αναβοσβήνει τις συσκευές (λάμπα, θερμοπομπό, αφυγραντήρα, εξαεριστήρα) έχοντας σαν κρίσιμες τιμές αυτές που του έχουμε καθορίσει στα πεδία «Φωτεινότητα», «Θερμοκρασία», «Υγρασία» και «Ποιότητα Αέρα» του ίδιου πλαισίου.

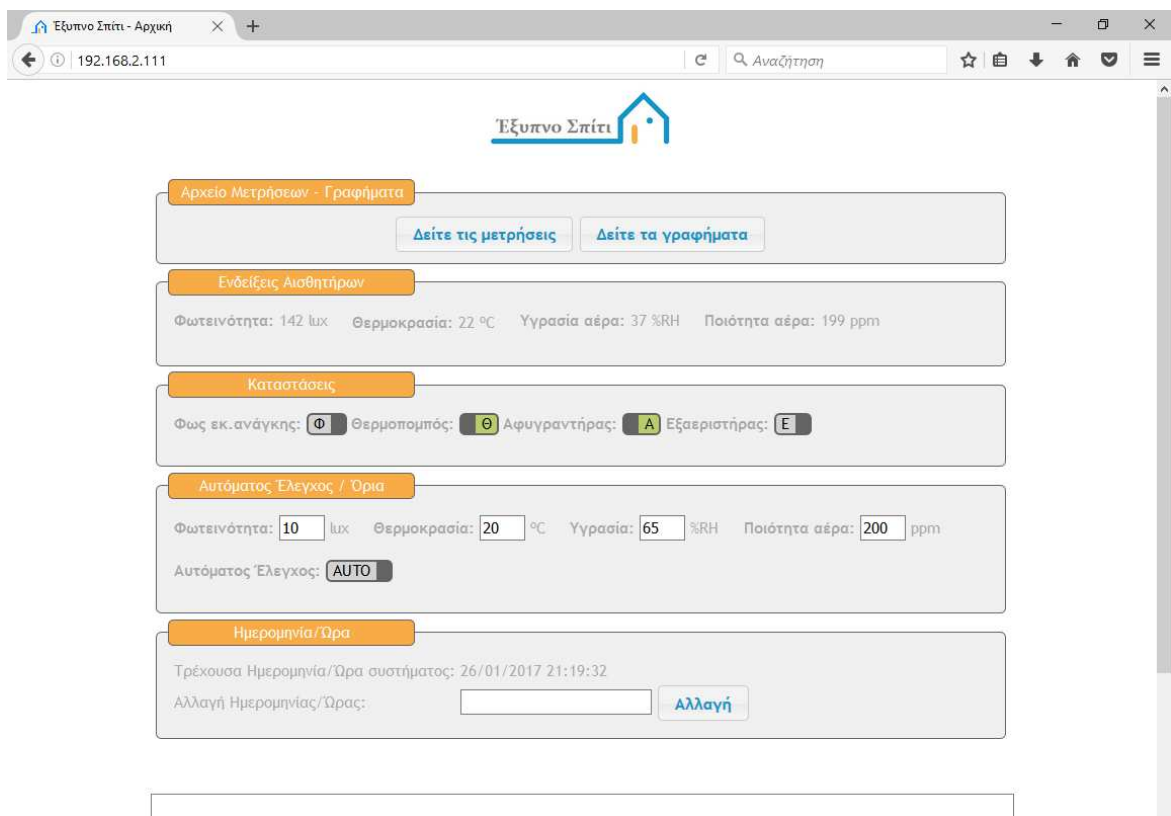
Εάν θέλουμε να αλλάξουμε τα όρια λειτουργίας των συσκευών θα αλλάξουμε τις τιμές των πεδίων κειμένου στις επιθυμητές και εν συνεχεία θα πρέπει να πατήσουμε δύο φορές το κουμπί «ΑΥΤΟ» δηλαδή να πάμε σε manual λειτουργία και αμέσως να ξαναπάμε σε αυτόματη λειτουργία ώστε το πρόγραμμα να στείλει τις νέες κρίσιμες τιμές στον κώδικα του Arduino.



Εικόνα 5.3: Αυτόματος έλεγχος.

## 5.4 Μη αυτόματος έλεγχος των συνθηκών

Όταν στο πλαίσιο «Καταστάσεις» πατήσουμε κάποιον από τους διακόπτες, αυτομάτως μπαίνουμε σε manual mode, δηλαδή απενεργοποιείται η «ΑΥΤΟ» επιλογή του πλαισίου «Αυτόματος Έλεγχος / Όρια». Έχουμε χειροκίνητο έλεγχο των συσκευών ανεξάρτητα από τις τιμές των συνθηκών που επικρατούν στο σπίτι. Το ίδιο γίνεται και όταν πατήσουμε πάνω στο κουμπί «ΑΥΤΟ» (όταν αυτό είναι ενεργοποιημένο), άμεσα σταματάει η αυτόματη λειτουργία, δηλαδή για να λειτουργήσουμε ή όχι τις περιφερειακές συσκευές πρέπει να αλλάξουμε χειροκίνητα την κατάσταση κάθε συσκευής στο πλαίσιο «Καταστάσεις». Για να φύγουμε από την manual λειτουργία πρέπει πάλι να ενεργοποιήσουμε την επιλογή «ΑΥΤΟ».



Εικόνα 5.4: Μη αυτόματος έλεγχος.

## 5.5 Εμφάνιση των μετρήσεων

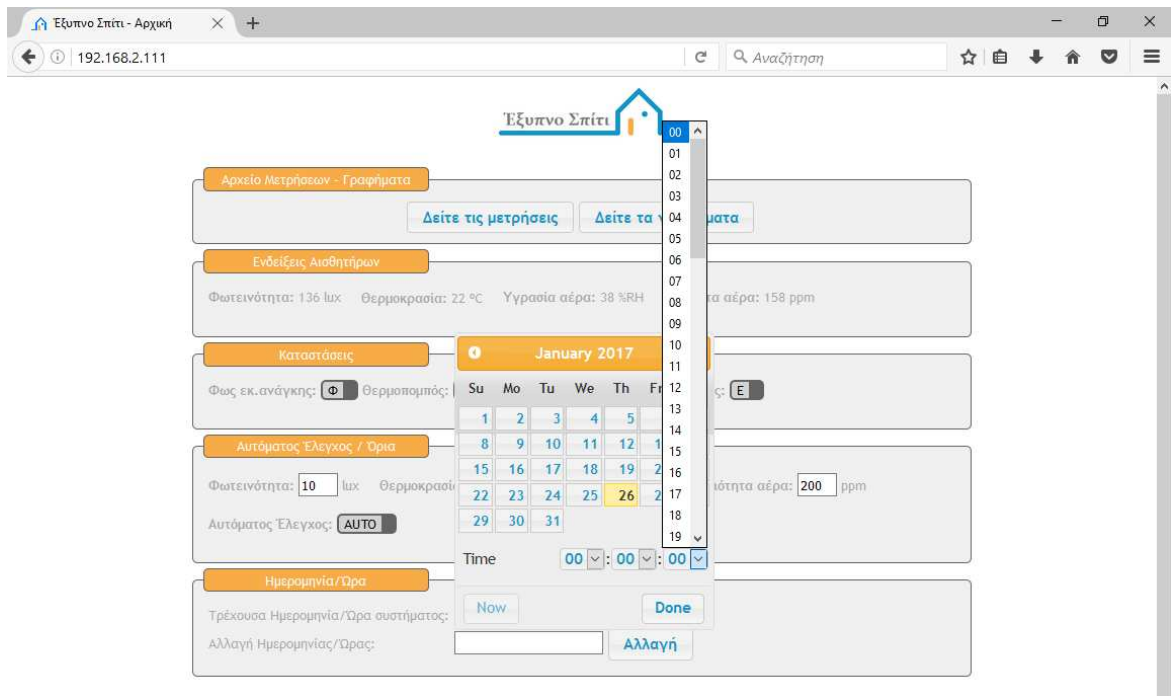
Το αρχείο των μετρήσεων θα εμφανιστεί στο πλαίσιο στο τέλος της σελίδας. Οι τιμές των μετρήσεων εμφανίζονται ομαδοποιημένες με την ακριβή ώρα και ημερομηνία κατά την οποία λήφθηκαν. Το πλαίσιο έχει μπάρα κύλισης και έτσι μπορούμε να δούμε όλες τις μετρήσεις.

Timestamp	Luminosity	Temperature	Humidity	Air quality	
2017-01-26-21-05-33 [1485464733]		150.41	22.00	37.00	204.00
2017-01-26-21-06-03 [1485464763]		147.27	22.00	36.00	201.00
2017-01-26-21-06-33 [1485464793]		150.41	22.00	36.00	210.00
2017-01-26-21-07-03 [1485464823]		146.49	22.00	36.00	206.00
2017-01-26-21-07-33 [1485464853]		134.78	22.00	36.00	213.00

Εικόνα 5.5: Εμφάνιση των μετρήσεων.

## 5.6 Αλλαγή ημερομηνίας και ώρας

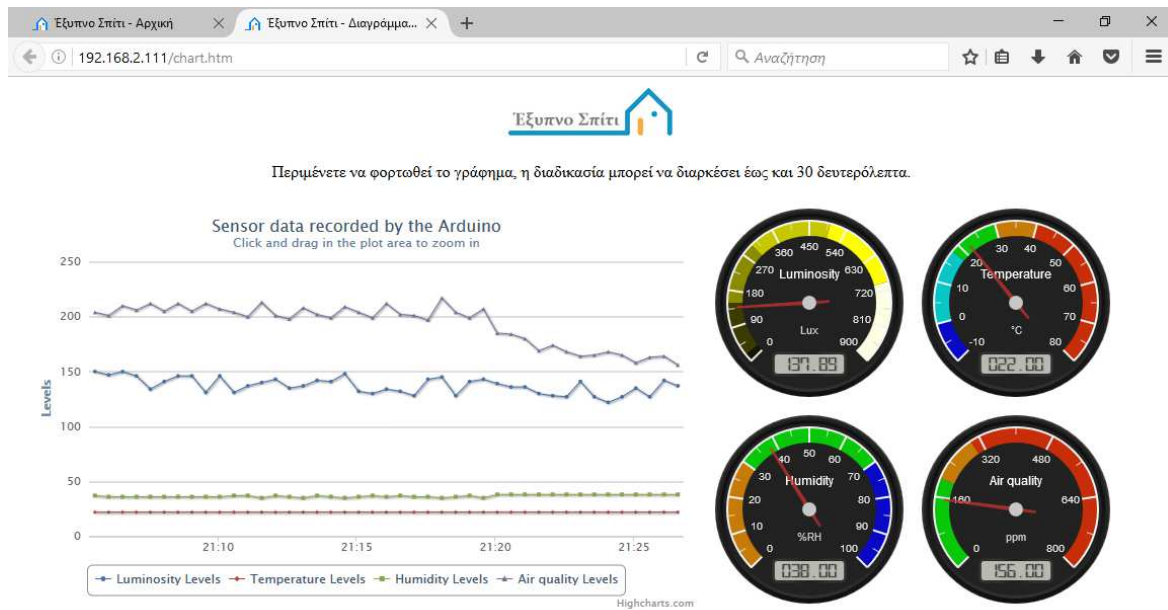
Εάν παρατηρήσουμε οποιοδήποτε πρόβλημα με την ημερομηνία ή την ώρα η οποία αναγράφεται δίπλα στο κείμενο «Τρέχουσα Ημερομηνία / Ώρα συστήματος», μπορούμε πατώντας μέσα στο κενό πεδίο, να αλλάξουμε την ημερομηνία ή την ώρα. Για το σκοπό αυτό αναδύεται ένα ημερολόγιο στο οποίο θα ρυθμίσουμε την πραγματική ώρα και την ημερομηνία. Πατώντας στο κουμπί αλλαγή θα μαζέψει το πρόγραμμα τα δεδομένα που βρίσκονται στο πεδίο κειμένου και θα τα στείλει στον κώδικα του Arduino για να προγραμματίσει εκ νέου και το ρολόι του DS3231.



Εικόνα 5.6: Αλλαγή ημερομηνίας/ώρας.

## 5.7 Τα γραφήματα των μετρήσεων

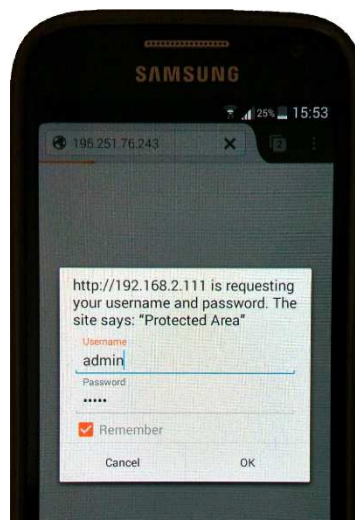
Στη 2<sup>η</sup> σελίδα (σε διαφορετικό tab του φυλλομετρητή) στην οποία εμφανίζονται τα γραφήματα όλων των μετρήσεων, οδηγούμαστε από την κύρια σελίδα όταν πατήσουμε το κουμπί «Δείτε τα γραφήματα» στο πλαίσιο «Αρχείο Μετρήσεων / Γραφήματα». Εκεί περιμένουμε να φορτωθεί το γράφημα για λίγα δευτερόλεπτα και έπειτα γίνεται ανανέωσή του κάθε 30". Βλέπουμε δύο πράγματα: ένα διάγραμμα με τις μετρήσεις όλων των συνθηκών ανά 30 δεύτερα και 4 δείκτες με τις τρέχουσες τιμές των συνθηκών για μια πιο άμεση και εύκολη πληροφόρηση.



Εικόνα 5.7: Η σελίδα chart.htm.

### 5.8 Το web περιβάλλον σε smartphone ή tablet

Μπορούμε να συνδεθούμε με το web κομμάτι της εφαρμογής από κινητό τηλέφωνο ή ταμπλέτα αρκεί να διαθέτουν ένα βασικό φυλλομετρητή. Ο χειρισμός και η πληροφόρηση γίνεται ακριβώς με τον ίδιο τρόπο με πριν ανεξάρτητα με το εάν τρέχουν android, windows ή ios. Η κινητή συσκευή πρέπει να συνδεθεί ασύρματα με το δίκτυό που είναι ήδη συνδεδεμένη η εφαρμογή, ή απευθείας από το internet, ώστε να έχουμε πρόσβαση σε αυτή.



Εικόνα 5.8.1: login.



Εικόνα 5.8.2: index.html.



Εικόνα 5.8.3: chart.htm.





## ΚΕΦΑΛΑΙΟ 6

### ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ

#### 6.1 Σύνοψη της πτυχιακής εργασίας

Η παρούσα εργασία έχει ως στόχο την απομακρυσμένη διαχείριση κάποιων οικιακών συσκευών μέσω οποιουδήποτε φυλλομετρητή, οποιασδήποτε συσκευής ανεξάρτητα λειτουργικού συστήματος και δυνατοτήτων του hardware, οπουδήποτε κι αν βρισκόμαστε. Ο χρήστης που θέλει να συνδεθεί με το web περιβάλλον διαχείρισης των συνθηκών χρειάζεται απλά να έχει πρόσβαση στο internet.

Το περιβάλλον της εφαρμογής είναι απλό και εύκολο στη χρήση, δεν χρειάζεται καμία εξοικείωση με αυτό, έτσι έχουμε άμεση πληροφόρηση και έλεγχο.

Αν χρειαστεί η μεταφορά της συσκευής σε άλλο χώρο δηλαδή σε άλλο δίκτυο μπορούμε εύκολα να ενημερώσουμε την εφαρμογή με τα νέα στοιχεία σύνδεσης απλά αλλάζοντας τα δεδομένα της κάρτας SD χωρίς να απαιτείται κανένας επαναπρογραμματισμός του Arduino.

Ως κατασκευή μπορεί να αναρτηθεί σε τοίχο ή να ενσωματωθεί σε ένα κουτί (διότι τα αισθητήρια μπορούν να μεταφερθούν και έξω από αυτό) και εύκολα να συνδεθεί με όλες τις περιφερειακές συσκευές και τις λοιπές συνδέσεις (τροφοδοσία, δίκτυο).

Ο στόχος του όλου έργου ήταν ο χρήστης αυτής της εφαρμογής να μπορεί να τη χρησιμοποιήσει χωρίς να έχει οποιαδήποτε εξοικείωση με το αντικείμενο και χωρίς να χρειάζεται να συντηρεί ή να ενημερώνει την εφαρμογή.

#### 6.2 Προοπτικές

Η εφαρμογή αυτή θα μπορούσε να ενσωματώσει πολλά ακόμη αισθητήρια, ρελέ ή τεχνολογίες δικτύωσης και σύνδεσης συσκευών (wifi, Bluetooth, Infrared, RF, GPRS κ.α), και να ελέγξει τις ανάλογες συσκευές.

Παραδείγματα επέκτασης των δυνατοτήτων ενός έξυπνου σπιτιού:

- σύνδεση ενός αισθητήρα υγρασίας χώματος ο οποίος να ελέγχει τα επίπεδα υγρασίας ενός πρεβαζιού ή μιας γλάστρας με διάφορα φυτά και μέσω ενός

ρελέ και μιας ηλεκτρικής βάνας νερού να μπορεί ο μικροελεγκτής να ποτίσει τα φυτά όταν ανιχνευθούν πολύ χαμηλά επίπεδα υγρασίας στο χώμα.

- σύνδεση ρελέ σε διακόπτες φωτισμού και πρίζες ώστε να έχουμε κεντροποιημένο και άμεσο έλεγχο φώτων και άλλων συσκευών όπως μιας καφετιέρας, ενός ανεμιστήρα, ενός κλιματιστικού κ.α.
- με ειδικά ρελέ τα οποία μπορούν να τοποθετηθούν στον ηλεκτρικό πίνακα του σπιτιού να ελέγχονται οι παροχές ρεύματος στην κουζίνα ή στον θερμοσίφωνα.
- σύνδεση κάμερας ώστε όποτε θελήσουμε να έχουμε κάποιο screenshot από το σημείο που είναι τοποθετημένη.
- σύνδεση αισθητηρίων κίνησης για την παρακολούθηση ενός χώρου σε παράλληλη χρήση κάμερας ή φωτισμού ή συναγερμού.
- σύνδεση ενός GPRS Shield ώστε να μπορούμε να δεχόμαστε ειδοποιήσεις στο κινητό μας όταν αυτό χρειάζεται ή να ελέγχουμε και τις συνθήκες μέσω γραπτών μηνυμάτων.
- Μπορούμε επίσης να συνδέσουμε τον Arduino μέσω wifi ή Bluetooth με οποιαδήποτε συσκευή ανάλογης τεχνολογίας και να την ελέγχουμε ασύρματα.
- Τέλος μπορούμε να συνδέσουμε και οποιοδήποτε smartphone ώστε να ελέγχουμε τις συσκευές μέσω αυτού.

Οποιαδήποτε ενσωμάτωση συσκευής φυσικά χρειάζεται και την προσθήκη στο πρόγραμμά μας των κατάλληλων βιβλιοθηκών για την υποστήριξή του αλλά και προσθήκες στον κώδικα για την αρχικοποίησή της και τον έλεγχό της.

Οι δυνατότητες είναι πάρα πολλές και εξαντλούνται όταν εξαντληθούν οι πόροι του μικροελεγκτή μας (ελεύθερα pins, μνήμη). Στην αγορά υπάρχουν μικροελεγκτές με μεγαλύτερες δυνατότητες από τον Arduino Mega, αν είναι απαραίτητη μεγαλύτερη για παράδειγμα μνήμη, απλά θα πρέπει να χρησιμοποιήσουμε άλλον με τις επιθυμητές δυνατότητες.

Στην αγορά υπάρχει πληθώρα επιλογών για την αυτοματοποίηση σπιτιών και κτηρίων. Έχουν αναπτυχθεί ειδικά συστήματα και δίκτυα γι' αυτό το σκοπό που παρέχουν ολοκληρωμένη κάλυψη όλων των απαιτήσεων των χρηστών. Ένα τέτοιο σύστημα είναι το KNX το οποίο είναι και διεθνές πρότυπο, είναι ευέλικτο,

χρησιμοποιεί πολλούς τύπους δικτύωσης και μπορεί να συνδεθεί και με άλλα δίκτυα κτηριακού αυτοματισμού, τηλεφωνικά δίκτυα κ.α. Με αυτό το σύστημα μπορούμε να ελέγξουμε το φωτισμό, τα παράθυρα και τα ρολά, το συναγερμό, την κατανάλωση ενέργειας, τις συσκευές ήχου και εικόνας, τους τηλεχειρισμούς συσκευών, τις λευκές συσκευές και να έχουμε μετρήσεις συνθηκών. Δηλαδή μια πλήρη κάλυψη των πιθανών αναγκών αυτοματοποίησης μιας οικίας ή ενός κτηρίου.

Το έξυπνο σπίτι θα γίνεται συνεχώς πιο «έξυπνο» προσθέτοντας εφαρμογές και αυτοματισμούς. Οι ιδέες είναι πολλές και υλοποιήσιμες. Μερικά παραδείγματα:

- Απομακρυσμένος έλεγχος της εισόδου του σπιτιού με δυνατότητα το άνοιγμα της πόρτας, πολύ χρήσιμο όταν κάποιος ξεχάσει τα κλειδιά του.
- Ψυγείο με ενσωματωμένη κάμερα στο εσωτερικό του. Την ώρα που κάποιος βρίσκεται στο super market και δε θυμάται ποια τρόφιμα του λείπουν μπορεί να δει το εσωτερικό του ψυγείου του και να καταλάβει αμέσως ποια είναι αυτά.
- Σε αντίστοιχη περίπτωση με την προηγούμενη μπορεί να τοποθετηθεί κάμερα στο εσωτερικό ενός φούρνου, να παρακολουθείται το φαγητό απομακρυσμένα, και μόλις είναι σχεδόν έτοιμο να σταματήσει η παροχή ρεύματος στο φούρνο.

Στο μέλλον το σπίτι θα είναι πραγματικά έξυπνο, ο ιδιοκτήτης του θα μπορεί να το ελέγξει απομακρυσμένα και να πληροφορείται για το οτιδήποτε συμβαίνει σε αυτό. Οι δυνατότητες ελέγχου και πληροφόρησης σε ένα έξυπνο σπίτι θα περιορίζονται μόνο όσο τις περιορίζει η φαντασία μας...

## ΠΑΡΑΡΤΗΜΑ Α΄: Ο κώδικας του Arduino

Ο κώδικας για τον Arduino έχει χωριστεί σε αρκετά αρχεία ώστε εύκολα και γρήγορα να εντοπίζουμε τα κομμάτια του κώδικα για κάθε αντικείμενο, τα αρχεία αυτά είναι:

- `smtho.ino`: Είναι το βασικό εκτελέσιμο αρχείο για τον Arduino, το οποίο καλεί όλα τα άλλα αρχεία. Αποτελείται από το `setup` κομμάτι το οποίο κάνει όλες τις απαραίτητες αρχικοποιήσεις το οποίο τρέχει μόνο μία φορά και την έναρξη εκτέλεσης του προγράμματος και ακολουθεί το `loop` κομμάτι το οποίο περιέχει τον κώδικα που επαναλαμβάνεται συνεχώς.
- `sensors.h`: περιέχει τον κώδικα για την αρχικοποίηση των αισθητήρων, τον ορισμό των αναλογικών και ψηφιακών `pin`, διαβάζει τις τιμές των αισθητηρίων, ανάβει ή σβήνει τα `led` και σπλίζει ή όχι τα `relé`. Ακόμα ελέγχει τη λειτουργία του `βομβητή`.
- `net.h`: περιέχει όλο τον κώδικα που χρειάζεται για την αρχικοποίηση της κάρτας δικτύου και τη σύνδεση με το δίκτυο. Τα απαραίτητα στοιχεία που χρειάζεται τα βρίσκει στο αρχείο `NET.CFG` το οποίο βρίσκεται στη κάρτα `SD`.
- `util.h`: είναι ο κώδικας που διαχειρίζεται το ρολόι, δηλαδή την ημερομηνία και την ώρα.
- `lcd.h`: περιέχει τον κώδικα για την αρχικοποίηση της `lcd` οθόνης και την απεικόνιση των μετρήσεων.
- `web.h` & `web-http.h`: στην ουσία αυτά τα δύο αρχεία είναι ο `web server` της εφαρμογής και διαχειρίζονται την επικοινωνία του Arduino με τις δύο `html` σελίδες της.
- `datalog.h`: περιέχει τον κώδικα για τη δημιουργία των αρχείων του καταλόγου των μετρήσεων. Τα απαραίτητα στοιχεία που χρειάζεται τα βρίσκει στο αρχείο `DATALOG.CFG` το οποίο βρίσκεται στη κάρτα `SD`.

Χρειάζονται και είναι ενσωματωμένες στο σύστημα και όλες οι βιβλιοθήκες για τις συσκευές που χρησιμοποιεί η εφαρμογή.

Παρακάτω φαίνεται το «`loop`» κομμάτι του κώδικα του Arduino, όπου φαίνονται οι κλήσεις προς όλα τα άλλα αρχεία της εφαρμογής:

```
void loop() {  
  // update network (net.h)  
  updateNetwork();  
  
  // update datalog file if needed (datalog.h)  
  if (checkRotateDatalogFile()) {  
    rotateDatalogFile();  
  }  
  
  if (datalogUpdateNeeded()) {  
    // take measurements from sensors (sensors.h)  
    updateSensors(measurements);  
  
    // log measurements (datalog.h)  
    updateDatalog(measurements);  
  }  
  
  // evaluate sensor measurement limits (sensors.h)  
  evaluateLimits();  
  
  // update LCD display (lcd.h)  
  updateLcd();  
  
  // listen for incoming connections (web.h)  
  listenForClients();  
}
```

Το loop κομμάτι του κώδικα κάνει τα εξής:

- Διαβάζει μετρήσεις απ' τους αισθητήρες
- Ενημερώνει το αρχείο μετρήσεων και το γράφει στην κάρτα SD
- Αν είναι στην αυτόματη λειτουργία ελέγχει αν πρέπει να αλλάξει την κατάσταση των συσκευών
- Ενημερώνει την οθόνη (LCD)
- Ελέγχει αν υπάρχει επικοινωνία μέσω web και την εξυπηρετεί

Παρακάτω φαίνεται απόσπασμα κώδικα του αρχείου sensors.h. Η evaluateLimits() δουλεύει όταν είμαστε σε Auto mode, συγκρίνει τις τιμές των τελευταίων μετρήσεων (οι οποίες βρίσκονται στον πίνακα measurements[]) με τα όρια που έχουμε βάλει και ανάλογα ανάβει ή σβήνει τις περιφερειακές συσκευές:

```

void evaluateLimits() {
  if ( __AUTO_MODE ) {
    __LUMINOSITY_STATE = measurements[0] < __LUMINOSITY_LIMIT;
    __TEMPERATURE_STATE = measurements[1] < __TEMPERATURE_LIMIT;
    __HUMIDITY_STATE = measurements[2] > __HUMIDITY_LIMIT;
    __AIR_STATE = measurements[3] > __AIR_LIMIT;
    digitalWrite(__LUMINOSITY_PIN, (__LUMINOSITY_STATE==true) ? HIGH : LOW);
    if ( digitalRead(__LUMINOSITY_PIN) == HIGH ){
      digitalWrite(__RELAY1_PIN, LOW);
    }else {
      digitalWrite(__RELAY1_PIN, HIGH);
    }
    digitalWrite(__TEMPERATURE_PIN, (__TEMPERATURE_STATE==true) ? HIGH : LOW);
    if ( digitalRead(__TEMPERATURE_PIN) == HIGH ){
      digitalWrite(__RELAY2_PIN, LOW);
    }else {
      digitalWrite(__RELAY2_PIN, HIGH);
    }
    digitalWrite(__HUMIDITY_PIN, (__HUMIDITY_STATE==true) ? HIGH : LOW);
    if ( digitalRead(__HUMIDITY_PIN) == HIGH ){
      digitalWrite(__RELAY3_PIN, LOW);
    }else {
      digitalWrite(__RELAY3_PIN, HIGH);
    }
    digitalWrite(__AIR_PIN, (__AIR_STATE==true) ? HIGH : LOW);
    if ( digitalRead(__AIR_PIN) == HIGH ){
      digitalWrite(__RELAY4_PIN, LOW);
    }else {
      digitalWrite(__RELAY4_PIN, HIGH);
    }
  }
}

```

Ο κώδικας που ενεργοποιεί το βομβητή (sensors.h), η μεταβλητή `__ALARM_PER_30SEC` ορίζεται στην αρχή του αρχείου, από default έχει οριστεί ίσο με 2, δηλαδή ο βομβητής θα χτυπάει κάθε 1 λεπτό όσο η θερμοκρασία είναι κάτω από 17 °C ή η υγρασία είναι πάνω από 60%:

```

if (*temperature < 17 or *humidity > 60) { //ACTIVATE ALARM
  if ( __ALARM_PER_30SEC > t){
    pinMode(__ALARM_PIN,OUTPUT);
    digitalWrite(__ALARM_PIN,HIGH);
    delay(50);
    digitalWrite(__ALARM_PIN,LOW);
    t=t+1;
  }
  else if(__ALARM_PER_30SEC = t){
    t=1;
  }
}

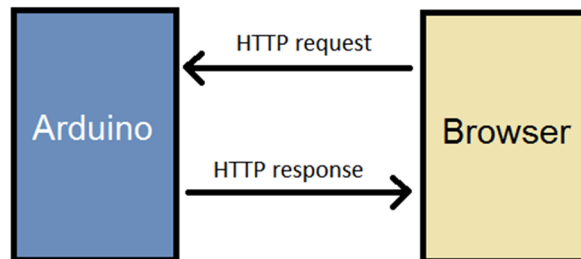
```

## ΠΑΡΑΡΤΗΜΑ Β΄: Ο κώδικας των web σελίδων

Εντολές που δέχεται ο Arduino μέσω της αρχικής web σελίδας:

- Κατέβασμα αρχείου μετρήσεων
- Διάβασμα των τρεχουσών μετρήσεων
- Διάβασμα της κατάστασης των συσκευών (on/off, auto/manual, όρια)
- Έλεγχος των συσκευών (on/off)
- Εμφάνιση / Αλλαγή ημερομηνίας και ώρας

Η index.htm χρησιμοποιεί html και javascript. Η επικοινωνία με τον web server της εφαρμογής γίνεται με την συνάρτηση ajaxGet η οποία χρησιμοποιεί ένα αντικείμενο XMLHttpRequest, δηλαδή μια κλήση ajax η οποία μόλις ένα συμβάν λάβει μέρος στην σελίδα μας (πάτημα ενός κουμπιού), στέλνει ένα request στον web server της εφαρμογής, αυτός επεξεργάζεται την κλήση αυτή με τα νέα δεδομένα και εν συνεχεία απαντάει ανάλογα στην σελίδα. Αυτή επεξεργάζεται την απάντηση και πράττει αναλόγως (π.χ. ανανέωση της σελίδας).



```

function ajaxGet(url, async, onSuccess, onError) {
    // 0. assign defaults to missing arguments
    if (!async) async=true;
    if (!onSuccess) onSuccess=null;
    if (!onError) onError=null;
    // 1. make xhr object
    var request = new XMLHttpRequest();
    // 2. Setup XHR object (url, method, onSuccess callback function, onError function)
    request.open('GET', url, async); //set web server url
    request.onload = function() {
        if (request.status >= 200 && request.status < 400) {
            // Success!
            if (onSuccess && onSuccess!=null) {
                onSuccess(request.responseText);
            }
        } else {
            // We reached our target server, but it returned an error
            if (onError && onError!=null) {
                onError(request.status);
            } else {
                alert('Server returned: '+request.status);
            }
        }
    };
    if (onError && onError!=null) {
        request.onerror = function() {
            onError('Could not connect to server');
        };
    } else {
        request.onerror = function() {
            // There was a connection error of some sort
            alert('Could not connect to server');
        };
    }
    // 3. Connect to server (asynchronously)
    try {
        request.send();
    } catch (e) {
        console.debug('ajaxGet: EXCEPTION: '+e);
        if (onError && onError!=null) {
            onError(''+e);
        }
    }
}
}

```

Όλα αυτά συμβαίνουν χωρίς να επηρεάζεται η σελίδα μέχρι την απάντηση, δηλαδή η επικοινωνία είναι ασύγχρονη. Οι επικοινωνία δηλαδή μεταξύ της web σελίδας και του web server της εφαρμογής γίνεται ανεξάρτητα από τη φόρτωση ή την ανανέωση της web σελίδας στον φυλλομετρητή.

Παρακάτω στην refreshControlState() βλέπουμε πως γίνεται η ανανέωση των στοιχείων που χρειάζεται η index.htm στην περίπτωση που δεχτεί δεδομένα από τον Arduino:



```

function refreshControlState() {
  ajaxGet('/getstate', true,
    function(str) {
      state = str.split(' ')[1];
      values = state.split('/');

      // get current control state
      autoOn = $('#_auto').hasClass('on');
      luxOn = $('#_lux_pin').hasClass('on');
      tempOn = $('#_temp_pin').hasClass('on');
      humOn = $('#_hum_pin').hasClass('on');
      airOn = $('#_air_pin').hasClass('on');
      // update controls state
      if (values[0]==='0' && autoOn || values[0]!='0' && !autoOn) $('#_auto').toggleClass('on');
      if (values[1]==='0' && luxOn || values[1]!='0' && !luxOn) $('#_lux_pin').toggleClass('on');
      if (values[2]==='0' && tempOn || values[2]!='0' && !tempOn) $('#_temp_pin').toggleClass('on');
      if (values[3]==='0' && humOn || values[3]!='0' && !humOn) $('#_hum_pin').toggleClass('on');
      if (values[4]==='0' && airOn || values[4]!='0' && !airOn) $('#_air_pin').toggleClass('on');
      // update controls limits
      $('#_lux_limit').val(values[5]);
      $('#_temp_limit').val(values[6]);
      $('#_hum_limit').val(values[7]);
      $('#_air_limit').val(values[8]);
      //update sensor values
      $('#_lux_value').html(values[9]+' lux');
      $('#_temp_value').html(values[10]+' <sup>o</sup>C');
      $('#_hum_value').html(values[11]+' %RH');
      $('#_air_value').html(values[12]+' ppm');
    },
    function(str) { alert("refreshControlState:\n"+str); }
  );
}

```

Στην `sendControlState` φαίνεται η διαδικασία αποστολής των καταστάσεων των `button` της `index` αλλά και τα επιθυμητά όρια που θέτει ο χρήστης. Κατασκευάζεται ένα `string` το οποίο περιέχει όλες τις πληροφορίες που θέλουμε να στείλουμε στον `Arduino`. Την αποστολή την κάνει η `ajaxGet`.

```

function sendControlState() {
  // get current control state
  autoOn = $('#_auto').hasClass('on')?1:0;
  luxOn = $('#_lux_pin').hasClass('on')?1:0;
  tempOn = $('#_temp_pin').hasClass('on')?1:0;
  humOn = $('#_hum_pin').hasClass('on')?1:0;
  airOn = $('#_air_pin').hasClass('on')?1:0;
  luxLimit = $('#_lux_limit').val();
  tempLimit = $('#_temp_limit').val();
  humLimit = $('#_hum_limit').val();
  airLimit = $('#_air_limit').val();
  url = '/setstate/'+autoOn+'/' + luxOn+'/' + tempOn+'/' + humOn+'/' + airOn+'/'
      + luxLimit+'/' + tempLimit+'/' + humLimit+'/' + airLimit;
  //console.log('>> SEND STATE: '+url);
  ajaxGet(url);
}

```

Δίνεται και το παρακάτω κομμάτι κώδικα από την `Index.htm` και την `chart.htm` αντίστοιχα με τα οποία μορφοποιούμε τα `pins` (τους διακόπτες) και τον `gauge` της υγρασίας ως παραδείγματα κώδικα σχετικού με την γραφική απεικόνιση:

```
{
  sensor: 'Humidity',
  canvas: null,
  width: 300,
  height: 300,
  unit: '%RH',
  minValue: 0,
  maxValue: 100,
  tickNum: 11,
  highlights: [[0, 'rgba(255,153,0,.75)'], [30, 'rgba(0,255,0,.75)'], [70, 'rgba(0,0,255,.75)']],
  gauge: null
},
```



Βλέπουμε ότι μπορούμε να αλλάξουμε τίτλο, μέγεθος, χρώματα, όρια μετρήσεων απεικόνισης, μονάδα μέτρησης.

```
<STYLE>
  a.toggler {
    background: lightgray;
    color: black;
    cursor: pointer;
    border: 2px solid rgb(100,100,100);
    border-right-width: 15px;
    padding: 5px;
    border-radius: 5px;
    text-decoration: none;
    transition: all .5s ease;
  }

  a.toggler.on {
    background: #b9cd6d; <!-- ενεργό (πράσινο)-->
    color: black;
    border-right-width: 2px;
    border-left-width: 15px;
  }
</STYLE>
```



Είναι δυνατή η επεξεργασία των χρωμάτων του φόντου και των γραμμάτων, του περιγράμματος, της θέσης του pin κ.α.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

<https://www.arduino.cc/en/Main/arduinoBoardMega>

<https://www.arduino.cc/en/Reference/EEPROM>

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

<http://www.instructables.com/class/Arduino-Class/>

[http://www.geeetech.com/wiki/index.php/Arduino\\_Mega\\_2560](http://www.geeetech.com/wiki/index.php/Arduino_Mega_2560)

<http://www.ardumotive.com/tutorials.html>

<http://www.ardumotive.com/workshop.html>

<http://www.ardumotive.com/smart-home-en.html>

[http://www.w3schools.com/xml/ajax\\_intro.asp](http://www.w3schools.com/xml/ajax_intro.asp)

<http://www.w3schools.com/js/default.asp>

[http://www.w3schools.com/html/html\\_scripts.asp](http://www.w3schools.com/html/html_scripts.asp)

<http://www.edaboard.com/thread278855.html>