



Ανάπτυξη ευφυούς ηλεκτρονικού ελεγκτή και  
κατάλληλου αλγορίθμου για βέλτιστο έλεγχο φωτισμού  
σε χώρους γραφείων

Φουτράκης Παντελής

Πολυτεχνείο Κρήτης

Τμήμα Ηλεκτρονικών Μηχανικών & Μηχανικών Υπολογιστών

2 Σεπτεμβρίου 2013

Εγκρίθηκε από την τριμελή επιτροπή:

1. Καθ. Καλαϊτζάκης Κωνσταντίνος (Επιβλέπων)
2. Καθ. Σταυρακάκης Γεώργιος
3. Επικ. Καθ. Κολοκοτσά Διονυσία

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κ. Καλαϊτζάκη Κωνσταντίνο καθώς και την καθηγήτρια μου κα. Κολοκοτσά Διονυσία για την αμέριστη βοήθεια και την πολύτιμη καθοδήγησή τους κατά την διάρκεια εκπόνησης της διπλωματικής μου εργασίας. Τέλος θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους μου για την στήριξη τους όλα αυτά τα χρόνια των σπουδών μου.

## Περίληψη

Τα φορτία ψύξης σε χώρους γραφείων με μεγάλες προσόψεις τζαμιών είναι αυξημένα, λόγω της διείσδυσης της ηλιακής ακτινοβολίας από τα παράθυρα, η οποία υπερθερμαίνει τα δωμάτια. Η ηλιακή ακτινοβολία παρέχει φυσικό φωτισμό στα δωμάτια, ο οποίος μπορεί να είναι μεγαλύτερος του αναμενόμενου τους καλοκαιρινούς μήνες και να δημιουργεί προβλήματα θάμπωσης. Με στόχο να ισορροπήσουμε με βέλτιστο τρόπο την μείωση των φορτίων ψύξης και του φυσικού φωτισμού μπορεί να εγκατασταθεί ένας ελεγκτής που θα επιλέγει με τον καλύτερο τρόπο την θέση των σκιάστρων και την λειτουργία ή όχι του τεχνητού φωτισμού. Ο ελεγκτής που υλοποιήθηκε, με βάση το arduino, μπορεί να εγκατασταθεί σε κτίρια γραφείων και να ελέγχει το επίπεδο των σκιάστρων και τη γωνία αυτών καθώς και το άνοιγμα/κλείσιμο του τεχνητού φωτισμού των γραφείων. Η εφαρμογή αυτή χρησιμοποιεί ως εισόδους έναν εσωτερικό και έναν εξωτερικό αισθητήρα φωτεινότητας, έναν αισθητήρα παρουσίας, μια κάρτα μνήμης για ανάγνωση των κανόνων της ασαφούς λογικής (fuzzy logic) και για αποθήκευση των δεδομένων και έναν προσαρμογέα xbee που υλοποιεί ένα ασύρματο δίκτυο μέσω του πρωτοκόλλου zigbee για επικοινωνία (είσοδο/ έξοδο) με τον υπολογιστή. Ο ελεγκτής βασίζεται σε κανόνες ασαφούς λογικής και οι παράμετροι ασαφοποίησης βελτιστοποιήθηκαν με τη χρήση των αλγορίθμων βελτιστοποίησης της MATLAB.

## **Abstract**

Cooling loads in office buildings with large glazing facades are increase due to solar radiation penetrating the windows and over-heating the rooms. Moreover solar radiation provides natural lighting in the rooms, which might be higher than expected in the summer months and causes glare problems. In order to balance optimally between the reduction of cooling loads and natural lighting a controller can be applied to select the best combination between the position of blinds and whether the artificial lights will operate or not.

The developed controller, arduino based, is applied in an office building and is controlling the level of venetian blinds and their angle. The application is using as input an indoor and outdoor illuminance sensor, and an occupancy sensor. The controller is based on fuzzy rules and the fuzzification parameters are optimized offline using matlab's optimization algorithms.

## Περιεχόμενα

Περίληψη .....	iii
Περιεχόμενα.....	v
Πίνακας εικόνων .....	vii
1 Εισαγωγή.....	1
2 Υπάρχουσα τεχνολογία για έλεγχο φώτων και σκιάστρων.....	5
3 Κατασκευή Arduino σε breadboard .....	8
3.1 Περιγραφή του μικρό-ελεγκτή Arduino.....	8
3.2 Υλοποίηση του Arduino σε breadboard.....	9
4 Κατασκευή Arduino σε PCB .....	20
5 Εξοπλισμός που τοποθετείται στον μικροελεγκτή .....	26
5.1 Αισθητήρας Παρουσίας (PIR sensor) .....	26
5.2 Αισθητήρας φωτεινότητας.....	28
5.3 Κάρτα μνήμης .....	32
5.3.1 Κύκλωμα κάρτας μνήμης στο breadboard .....	32
5.3.2 Προσαρμογέας της κάρτας μνήμης στο PCB.....	35
5.3.3 Λειτουργικότητα της κάρτας μνήμης.....	37
5.4 Προσαρμογέας XBee.....	38
5.4.1 Περιγραφή πρωτοκόλλου zigbee .....	38
5.4.2 Διάταξη XBee στο breadboard .....	42
5.4.3 Προσαρμογέας XBee στο PCB.....	43
5.4.4 Λειτουργικότητα του XBee .....	44
6 Κατασκευή και σχεδίαση shield για τον μικρό-ελεγκτή Arduino	46

6.1	Σύνδεση συστήματος και διάταξη στο χώρο .....	48
7	Σχεδίαση και υλοποίηση ασαφούς λογικής .....	49
8	Προγραμματισμός του συστήματος .....	54
8.1	Διάγραμμα ροής .....	54
8.2	Πηγαίος κώδικας .....	55
9	Αποτελέσματα .....	67
10	Συμπεράσματα.....	69
11	Μελλοντικές εφαρμογές .....	70
	Βιβλιογραφία .....	71

## Πίνακας εικόνων

Εικόνα 1: Arduino σε breadboard: Τα εξαρτήματα του arduino .....	10
Εικόνα 2: Arduino σε breadboard:Γραμμές τάσεις .....	10
Εικόνα 3: Arduino σε breadboard:Ένωση των γραμμών τάσης του πάνω με του κάτω μέρους .....	11
Εικόνα 4: Arduino σε breadboard:Προσθήκη του 7805 και των πυκνωτών .....	11
Εικόνα 5: Arduino σε breadboard: LED ένδειξης τροφοδοσίας .....	12
Εικόνα 6: Arduino σε breadboard:Είσοδος τροφοδοσίας του κυκλώματος από μπαταρία .....	12
Εικόνα 7: Atmega 328P pin mapping .....	13
Εικόνα 8: Arduino σε breadboard:Προσθήκη μικρό-επεξεργαστή και της τροφοδοσίας του .....	14
Εικόνα 9: Arduino σε breadboard:Προσθήκη κουμπιού reset.....	14
Εικόνα 10: Arduino σε breadboard: Προσθήκη LED στο pin 13 .....	15
Εικόνα 11: Arduino σε breadboard:Arduino στο breadboard.....	16
Εικόνα 12: FT232 pin mapping .....	17
Εικόνα 13: Arduino σε breadboard:Σύνδεση της τροφοδοσίας του FT232 .....	17
Εικόνα 14: Arduino σε breadboard:Σύνδεση καλωδίων για τα σήματα Rx, Tx.....	18
Εικόνα 15: Arduino top view .....	20
Εικόνα 16: Arduino bottom view .....	21
Εικόνα 17: Διάγραμμα τρυπήματος.....	21
Εικόνα 18: Λίστα εξαρτημάτων .....	22
Εικόνα 19: Διάγραμμα τοποθέτησης των εξαρτημάτων .....	23
Εικόνα 20: Arduino S3V3 (Revision 2) top view .....	23
Εικόνα 21: Arduino S3V3 (Revision 2) bottom view .....	24



Εικόνα 22: Τα arduino που κατασκευάστηκαν στο εργαστήριο ηλεκτρονικής .....	24
Εικόνα 23: Το ηλεκτρολογικό σχέδιο του arduino.....	25
Εικόνα 24: PIR sensor.....	26
Εικόνα 25: Λεπτομέρειες PIR sensor .....	27
Εικόνα 26: Σύνδεση του PIR sensor στο arduino .....	28
Εικόνα 27: Αισθητήρας φωτεινότητας (TEMT6000) .....	28
Εικόνα 28: Αξιολογήσεις του αισθητήρα .....	28
Εικόνα 29: Βασικά χαρακτηριστικά του αισθητήρα .....	29
Εικόνα 30: Collector Light Current vs Illuminance .....	29
Εικόνα 31: Relative Spectral Responsivity vs Wavelength.....	30
Εικόνα 32: Relative Radiant Sensitivity vs Angular Displacement....	30
Εικόνα 33: Ηλεκτρολογικό διάγραμμα του TEMT6000 .....	31
Εικόνα 34: Σύνδεση του TEMT6000 sensor στο arduino .....	32
Εικόνα 35: Voltage regulator 3V3.....	33
Εικόνα 36: CD4050BCN level shifter .....	33
Εικόνα 37: Breakout Board for SD-MMC Card .....	33
Εικόνα 38: Διάταξη του κυκλώματος της κάρτας μνήμης στο breadboard .....	34
Εικόνα 39: Σύνδεση της κάρτας μνήμης και των αισθητήρων με το arduino.....	34
Εικόνα 40: Ηλεκτρολογικό σχέδιο του PCB.....	36
Εικόνα 41: Προσαρμογέας της κάρτας μνήμης για σύνδεση με το shield .....	37
Εικόνα 42: Τα επίπεδα του πρωτοκόλλου ZigBee .....	42
Εικόνα 43: Τοποθέτηση του XBee στο breadboard .....	42
Εικόνα 44: Σύνδεση του XBee με το arduino .....	43
Εικόνα 45: Ηλεκτρολογικό σχέδιο του προσαρμογέα Xbee .....	43
Εικόνα 46: PCB του προσαρμογέα Xbee.....	44

Εικόνα 47: Pin mapping του XBee adaptor .....	44
Εικόνα 48: Ηλεκτρολογικό σχέδιο του shield.....	47
Εικόνα 49: PCB του shield που σχεδιάστηκε .....	48
Εικόνα 50: Κανόνες του συστήματος για τα τεχνητά φώτα .....	51
Εικόνα 51: Κανόνες του συστήματος για τα σκίαστρα .....	51
Εικόνα 52: Δομή για σκίαστρα και τεχνητά φώτα .....	53
Εικόνα 53: Διάγραμμα ροής .....	54
Εικόνα 54: Fuzzy logic μοντέλο του συστήματος .....	67
Εικόνα 55: Το σύστημα υλοποιημένο στο breadboard .....	67
Εικόνα 56: Η σχεδιασμένη πλακέτα για τους απαραίτητους αισθητήρες.....	68
Εικόνα 57: Οπτικοποίηση των κανόνων από τη Matlab για τον φωτισμό .....	68
Εικόνα 58: Οπτικοποίηση των κανόνων από τη Matlab για τα σκίαστρα.....	68

## 1 Εισαγωγή

Η αύξηση της ενεργειακής αποδοτικότητας των κτιρίων αποτελεί μείζον θέμα στις μέρες μας, καθώς συμβάλει ταυτόχρονα στη μείωση της κατανάλωσης των συμβατικών πηγών ενέργειας, ελάττωση του ενεργειακού κόστους για τους ιδιοκτήτες κτιρίων και ελάττωση των αερίων της ατμόσφαιρας που συμβάλλουν στην υπερθέρμανση του πλανήτη. Τα κτίρια καταναλώνουν το 40% της ενέργειας που χρησιμοποιείται παγκοσμίως και είναι ευρέως αποδεκτό ότι αν ληφθούν μέτρα και γίνουν αλλαγές στην λειτουργικότητα των κτιρίων θα πετύχουμε σημαντική εξοικονόμηση. Επιπλέον αναμένεται από τα κτίρια να πετύχουν πολύ υψηλά πρότυπα στο κοντινό μέλλον. Οφείλουν να κατασκευαστούν με μια βιώσιμη λύση, ώστε να πετύχουν ενεργειακή ισορροπία, να έχουν υγιές και άνετο περιβάλλον, να ανταποκρίνονται καλά στο ενεργειακό δίκτυο και να είναι οικονομικά στην κατασκευή και στη συντήρηση.

Σε ένα τυπικό κτίριο γραφείων, η εξοικονόμηση ενέργειας για τον φωτισμό είναι σημαντικό πρόβλημα καθώς ο φωτισμός ευθύνεται για το 40% της συνολικής κατανάλωσης ηλεκτρισμού σε ολόκληρο το κτίριο. Όμως τα επίπεδα του φωτισμού επηρεάζουν τους εργαζόμενους στα γραφεία κι ένα ακατάλληλο επίπεδο στο φωτισμό μειώνει την παραγωγικότητα. Ένα σύστημα ελέγχου φωτισμού για ένα κτίριο γραφείων πρέπει να πετύχει την κατάλληλη εξοικονόμηση με προσεκτικό και ακριβή έλεγχο του φωτός. Σε κάθε περίπτωση η εξοικονόμηση ενέργειας δεν πρέπει να διακινδυνεύσει την άνεση των χρηστών του κτιρίου. Η έλλειψη άνεσης έχει άμεση επίπτωση στην παραγωγικότητα των χρηστών και έμμεση επίπτωση στην ενεργειακή αποδοτικότητα του κτιρίου. Παράλογες απαιτήσεις από τους χρήστες αποβαίνουν καταστροφικές για την ενεργειακή αποδοτικότητα.

Συστήματα ενεργειακής διαχείρισης κτιρίων (Building energy management systems (BEMS)) συμβάλλουν στην ενεργειακή αποδοτικότητα, ενώ παράλληλα διατηρούν την άνεση και τα επίπεδα του φωτισμού στα επιθυμητά επίπεδα. Τα περισσότερα BEMS αποθηκεύουν δεδομένα για την ενεργειακή ανάλυση, έλεγχο και βελτιστοποίηση της ενεργειακής διαχείρισης, διατηρούν την άνεση και τα επίπεδα του φωτισμού στα επιθυμητά επίπεδα. Αν και η εγκατάσταση των BEMS συνήθιζε να είναι αποδοτική μόνο σε νέα κτίρια λόγω των αυξημένων απαιτήσεων καλωδίωσης για την επικοινωνία, πρόσφατες υλοποιήσεις στον αυτοματισμό κτιρίων και στον τομέα ελέγχου με την εισαγωγή διάφορων μέσων μετάδοσης, βοήθησε σημαντικά την διάδοση της ενεργειακής διαχείρισης και στα ήδη υπάρχοντα κτίρια. Κύριος στόχος των BEMS είναι να ικανοποιήσουν τη θερμική και οπτική άνεση, την ποιότητα του αέρα στο εσωτερικό του κτιρίου, καθώς και την μείωση της ενεργειακής κατανάλωσης. Τεχνητή νοημοσύνη (AI) έχει εφαρμοστεί σε προηγμένα συστήματα ελέγχου του περιβάλλοντος ενός κτιρίου όπως συστήματα ελέγχου της θερμοκρασίας, του φωτός και της ποιότητας του αέρα. Με σημαντικό πλεονέκτημα την εφαρμογή σε μη γραμμικά ή ασαφούς δυναμικότητας συστήματα, μέθοδοι ελέγχου βασισμένες στην τεχνητή νοημοσύνη ενσωματωμένες στα Fuzzy Logic (FL), Adaptive Neuro-Fuzzy Inference System (ANFIS), and Artificial Neural Network (ANN) έχουν δημιουργήσει επιτυχώς περισσότερο άνετα περιβάλλοντα σε κτίρια. Στην οπτική άνεση, προηγμένες λογικές που χρησιμοποιούν μοντέλα AI ερευνώνται για βέλτιστο έλεγχο φωτισμού και βελτιωμένη ενεργειακή απόδοση.

Σε ερευνητικό επίπεδο ο τομέας μοντελοποίησης της συμπεριφοράς των εργαζομένων σε ένα κτίριο και ο τομέας της ανάπτυξης έξυπνων ελεγκτών (ειδικά για φώτα και σκίαστρα) εξελίσσονται συγχρόνως. Από

τη μία υπάρχει μεγάλη ποικιλία αλγορίθμων ελέγχου για σκίαστρα και φώτα και από την άλλη πολλά στοχαστικά μοντέλα έχουν αναπτυχθεί να μιμούνται την συμπεριφορά ενός εργαζομένου όσον αφορά την χρήση των φώτων, σκιάστρων και του εξαερισμού. Με την συνδυασμένη χρήση των στοχαστικών μοντέλων και των αλγορίθμων ελέγχου για τον φωτισμό και τα σκίαστρα πετυχαίνουμε ακριβέστερη προσομοίωση και μπορεί να προβλεφθεί καλύτερα η αποδοτικότητα του ελεγκτή. Με τον έλεγχο των σκιάστρων και των φώτων όχι μόνο βελτιώνεται η κατανάλωση ηλεκτρικής ενέργειας για το φωτισμό αλλά και η κατανάλωση ενέργειας για τη θέρμανση και την ψύξη. Εσωτερικά σκίαστρα χρησιμοποιούνται ευρέως σε χώρους γραφείων για να ελέγξουν τη θερμική συμβολή της ηλιακής ακτινοβολίας και να εμποδίσουν την οπτική δυσφορία. Οι ιδιότητες τους έχουν σημαντική επίπτωση στη διαθεσιμότητα του ηλιακού φωτός στα γραφεία και εξαιτίας αυτού στις ενεργειακές απαιτήσεις για φωτισμό, θέρμανση και ψύξη. Πρόσφατες μελέτες στην αυτόματη λειτουργία των συσκευών σκίασης δείχνουν μεγάλες δυνατότητες για εξοικονόμηση ενέργειας.

Η επιστημονική κοινότητα και οι κυβερνήσεις είναι ενημερωμένες αρκετά χρόνια τώρα για το πρόβλημα υπερθέρμανσης του πλανήτη. Συνέπεια αυτού είναι ένας ευρωπαϊκός στόχος να μειωθούν τα αέρια που συμβάλλουν στο φαινόμενο του θερμοκηπίου κατά 20% μέχρι το 2020, ενώ παράλληλα να επιτραπεί η δημογραφική και οικονομική ανάπτυξη. Αυτό μπορεί να επιτευχθεί μόνο εάν η κατανάλωση ενέργειας βελτιστοποιηθεί. Στα κτίρια τα συστήματα θέρμανσης καταναλώνουν πάνω από 50% που σημαίνει περίπου το 23% της συνολικής κατανάλωσης ενέργειας. Αν και η τάση είναι να κατασκευαστούν νέα αποδοτικά σε ενέργεια κτίρια, μείωση στην συνολική ενεργειακή κατανάλωση δεν μπορεί να επιτευχθεί χωρίς να βελτιστοποιηθούν τα ήδη

υπάρχοντα κτίρια. Καθώς οι ανακαινίσεις είναι δαπανηρές και χρονοβόρες, ένα προηγμένο σύστημα ελέγχου προτείνεται ως καλύτερη λύση. Η πρόκληση για το σύστημα ελέγχου του φωτισμού είναι η εύρεση του ιδανικού σημείου μεταξύ της οπτικής άνεσης του χρήστη και της ελάχιστης κατανάλωσης ενέργειας.

## **2 Υπάρχουσα τεχνολογία για έλεγχο φωτών και σκιάστρων**

Οι ελεγκτές σκιάστρων ήταν ή εξαρτώμενοι από το χρόνο ή βασισμένοι σε μια τιμή κατωφλίου, για παράδειγμα την ηλιακή ακτινοβολία. Με το πέρασμα των χρόνων περισσότερες μεταβλητές εισόδου προστέθηκαν και το αποτέλεσμα ήταν να μην είναι πλέον γραμμικά εξαρτώμενοι από τις μεταβλητές εισόδου. Τα τελευταία χρόνια αυτοματοποιημένες προσαρμογές του χρήστη και της αποδοχής του προσέλκυσαν ερευνητές και προγραμματιστές στον τομέα. Στις πρώτες προσεγγίσεις τα σκιάστρα προσαρμόζονταν με βάση μία μεταβλητή εισόδου. Σε άλλο μοντέλο τα σκιάστρα χαμήλωναν αν το επίπεδο της ηλιακής ακτινοβολίας ξεπερνούσε τα  $233 \text{ W/m}^2$ . Με αυτόν τον απλό κανόνα ήταν δυνατό να ρίξει το μέσο PPD (προβλεπόμενο ποσοστό των χρηστών δυσαρεστημένων με τη θερμότητα του περιβάλλοντος) από το 22% στο 13%. Περισσότερο ευφυείς ελεγκτές περιλαμβάνουν περισσότερες της μίας μεταβλητές και είναι κλειστού βρόχου αλγόριθμοι. Οι (Trobac Lah et al. 2006) δημιούργησαν ένα FL σύστημα για τη διαχείριση των σκιάστρων με στόχο την διατήρηση των επιθυμητών επιπέδων του ηλιακού φωτός στο εσωτερικό του κτιρίου. Οι εισοδοί γι' αυτό το σύστημα είναι η εσωτερική φωτεινότητα, η ολική και η ανακλώμενη ηλιακή ακτινοβολία, καθώς και η παρούσα θέση των σκιάστρων. Το σύστημα εφαρμόστηκε σε ένα θάλαμο δοκιμής και επέδειξε μεγάλη αξιοπιστία στον έλεγχο της εσωτερικής φωτεινότητας συγκριτικά με τη διαθέσιμη ηλιακή ακτινοβολία. Η θερμική άνεση δεν ελεγχόταν από αυτό το σύστημα.

Άλλη βελτίωση είναι η προσαρμογή του συστήματος στις ανάγκες των χρηστών. Οι (Guillemin and Morel 2002) πρότειναν ένα σύστημα με ενσωματωμένο αυτό-προσαρμοζόμενο ελεγκτή για τη θέρμανση, τη σκίαση και το φωτισμό. Οι εισοδοί του ελεγκτή είναι η τρέχουσα ώρα, η

εσωτερική και εξωτερική θερμοκρασία, η ηλιακή ακτινοβολία, η παρουσία ή απουσία χρηστών στο δωμάτιο και επιλεγμένες ρυθμίσεις από τους ίδιους τους χρήστες. Ο ελεγκτής αποτελείται από δυο τεχνητά νευρωνικά δίκτυα (ANN) για την πρόβλεψη της θερμοκρασίας του δωματίου και του καιρού, FL για τον έλεγχο της θέρμανσης και FL για τον έλεγχο του τεχνητού φωτισμού και των σκιάστρων. Για την συνεχή βελτιστοποίηση το σύστημα κάθε βράδυ με τη χρήση γενετικών αλγορίθμων ψάχνει το πιο αποδοτικό σενario παραμέτρων για τους ελεγκτές.

Η (Kolokotsa et al. 2006) παρουσιάζει έναν FL ελεγκτή για την εσωτερική θερμική και οπτική άνεση και την ποιότητα του αέρα βασισμένη στον EIB (European Installation Bus) και στη Matlab. Το σύστημα δοκιμάστηκε και υλοποιήθηκε σε ένα πειραματικό θάλαμο (μεγέθους 1m x 1m x 2m) εξοπλισμένο με αισθητήρες για εσωτερική και εξωτερική θερμοκρασία, υγρασία, φωτεινότητα, ροή αέρα και συγκέντρωση CO<sub>2</sub>. Ο fuzzy ελεγκτής έχει ως είσοδο της ακόλουθες παραμέτρους: Predicted Mean Vote (PMV), εξωτερική θερμοκρασία, απαιτήσεις θέρμανσης ή ψύξης, άνοιγμα παραθύρου, εσωτερική φωτεινότητα, επίπεδο του τεχνητού φωτισμού και θέση των σκιάστρων. Ως εξόδους έχει: το επίπεδο του τεχνητού φωτισμού και των σκιάστρων. Το σύστημα δοκιμάστηκε για 3 μέρες και έδειξε καλή συμπεριφορά στη διατήρηση των επιθυμητών επιπέδων. Στις περισσότερες δημοσιεύσεις φαίνεται ότι οι προτεινόμενοι ελεγκτές σε θέμα κατανάλωσης ενέργειας είναι ανώτεροι από τους απλούς ελεγκτές ανοίγματος-κλεισίματος. Αυτό αποδεικνύει ότι η ανάπτυξη ελεγκτών φωτισμού διατηρεί τον στόχο της ταυτόχρονης παροχής άνετου περιβάλλοντος και της εξοικονόμησης ενέργειας. Όμως η προσαρμογή και η αποδοχή του χρήστη είναι το κέντρο του ενδιαφέροντος. Μια άλλη τάση είναι η εισαγωγή περισσότερων παραμέτρων στον ελεγκτή. Σε επίπεδο εργαστηρίου αυτό



μπορεί να μην αποτελεί πρόβλημα αλλά σε μια πραγματική εφαρμογή οι επιπλέον αισθητήρες συχνά αυξάνουν το κόστος, περιπλέκουν την εγκατάσταση και είναι απωθητικοί για τους χρήστες.

Οι (Kurian et al. 2008) ανέπτυξαν ένα συνδυασμένο σύστημα για τον έλεγχο των φώτων και των σκιάστρων. Το σύστημα αυτό αποτελείται από τρεις FL ελεγκτές: θάμπωσης, οπτικής άνεσης και εξοικονόμηση ενέργειας (δεδομένης απουσίας χρηστών). Για τον έλεγχο των σκιάστρων και του τεχνητού φωτισμού ένα adaptive neuro-fuzzy inference σύστημα (ANFIS) υλοποιήθηκε. Το σύστημα συγκρίθηκε με ένα απλό ελεγκτή ανοίγματος-κλεισίματος και επέφερε μείωση κατανάλωσης της ενέργειας από 5% έως 60%. Ο έλεγχος του φωτισμού δεν οφείλει μόνο να βελτιστοποιεί την κατανάλωση ενέργειας αλλά και να προσφέρει αρκετό φωτισμό για τη συμβολή σε ένα άνετο περιβάλλον εργασίας. Επιπροσθέτως πρέπει να τονιστεί ότι ο φυσικός φωτισμός έχει θετική επίδραση στο σώμα και αυξάνει την παραγωγικότητα, πράγμα πολύ σημαντικό για χώρους γραφείων.

Αν και είναι προφανές ότι τα συστήματα ελέγχου του φωτισμού έχουν θετική επίδραση στην κατανάλωση ενέργειας, όταν διατηρούν υψηλά επίπεδα φωτισμού η σύγκριση με τα επίπεδα οικονομίας είναι σχετικά δύσκολη. Αυτό οφείλεται στις διαφορετικές ρυθμίσεις και στις απαιτήσεις των χρηστών.

### **3 Κατασκευή Arduino σε breadboard**

#### **3.1 Περιγραφή του μικρό-ελεγκτή Arduino**

Το arduino ανήκει σε μια οικογένεια μικρό-ελεγκτών (μικροσκοπικοί υπολογιστές) και εμπεριέχει ένα περιβάλλον ανάπτυξης λογισμικού που κάνει εύκολο για το χρήστη την συγγραφή προγραμμάτων τα οποία αλληλεπιδρούν με το φυσικό κόσμο. Εφαρμογές που μπορεί κάποιος να υλοποιήσει με το arduino μπορούν να αισθανθούν και να ανταποκριθούν στην αφή, στον ήχο, στη θέση, στη θερμότητα και στο φως. Αυτός ο τύπος της τεχνολογίας (συχνά αναφέρεται ως physical computing) χρησιμοποιείται σ' όλου του είδους τις εφαρμογές από τα έξυπνα κινητά έως ηλεκτρονικά συστήματα αυτοκινήτων. Το arduino δίνει τη δυνατότητα σε οποιονδήποτε που ενδιαφέρεται, ακόμα και σε ανθρώπους χωρίς γνώση προγραμματισμού ή ηλεκτρονικών κυκλωμάτων, να χρησιμοποιήσουν αυτή την πληθωρική και περίπλοκη τεχνολογία. Ακόμη και το περιβάλλον του arduino έχει σχεδιαστεί ώστε να είναι προσιτό και στον αρχάριο χρήστη που δεν έχει γνώσεις προγραμματισμού ή ηλεκτρονικών κυκλωμάτων. Το arduino έχει χρησιμοποιηθεί για τη δημιουργία ποικίλων συστημάτων, συμπεριλαμβάνοντας μουσικά όργανα, ρομπότ (<http://arduino.cc/en/Main/Robot>), παιχνίδια και διαδραστικά ρούχα (<http://arduino.cc/en/Main/arduinoBoardLilyPad>). Το arduino χρησιμοποιείται σε πολλά εκπαιδευτικά προγράμματα ανά τον κόσμο αλλά και από designers και καλλιτέχνες που δημιουργούν πρωτότυπα με ευκολία αλλά δεν θέλουν να έχουν εις βάθος κατανόηση των τεχνικών λεπτομερειών πίσω από τις δημιουργίες τους. Επειδή έχει σχεδιαστεί να χρησιμοποιείται και από αρχάριους χρήστες, το λογισμικό του περιλαμβάνει αρκετά παραδείγματα κώδικα ώστε να επιδείξει πώς να χρησιμοποιήσει ο χρήστης τις διάφορες δυνατότητες της πλακέτας. Αν και εύκολο στη χρήση το υποβόσκον υλικό του arduino λειτουργεί στο

ίδιο επίπεδο εξυπνάδας που οι μηχανικοί εφαρμόζουν όταν υλοποιούν ενσωματωμένα συστήματα. Άνθρωποι που ήδη δουλεύουν πάνω σε μικρό-ελεγκτές έλκονται από το arduino λόγω των ευέλικτων δυνατοτήτων ανάπτυξης και της γρήγορης υλοποίησης των ιδεών τους. Το arduino είναι ευρέως γνωστό για το υλικό του, όμως χρειάζεται λογισμικό για να προγραμματιστεί το υλικό. Τόσο το υλικό όσο και το λογισμικό φέρουν την ονομασία arduino. Ο συνδυασμός των δύο επιτρέπει τη δημιουργία σχεδίων που αισθάνονται και ελέγχουν τον φυσικό κόσμο. Το λογισμικό είναι δωρεάν, ανοιχτού κώδικα και υπάρχει διαθέσιμο για όλες τις πλατφόρμες λειτουργικών συστημάτων. Οι πλακέτες είναι φθηνές για αγορά αλλά μπορεί ο χρήστης να φτιάξει τις δικές του (τα σχέδια του υλικού είναι επίσης προσβάσιμα δωρεάν). Επιπροσθέτως υπάρχει μια ενεργή και υποστηρικτική κοινότητα του arduino που είναι προσβάσιμη μέσω των arduino forums και της wiki (γνωστή ως arduino playground). Τα forums και η wiki προσφέρουν παραδείγματα καθώς και λύσεις σε προβλήματα, ενώ παράλληλα προσφέρουν έμπνευση και καθοδήγηση όσο υλοποιεί ο χρήστης τα δικά του σχέδια.

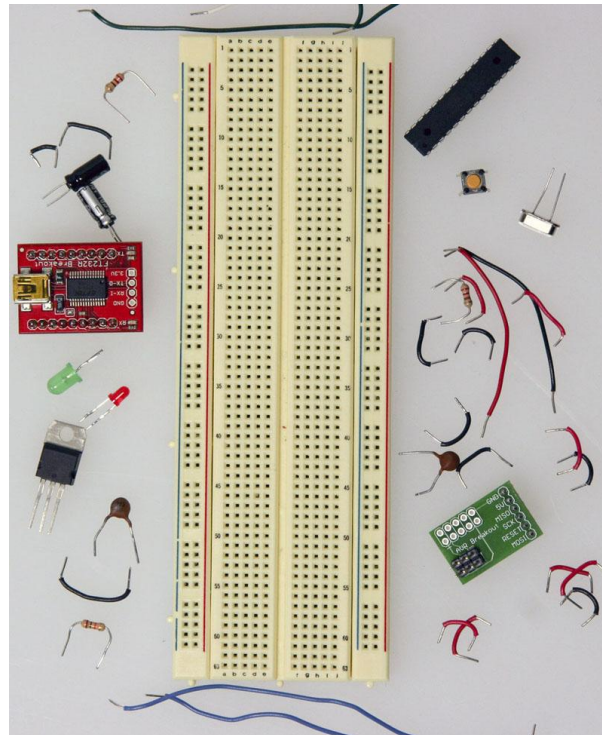
### 3.2 Υλοποίηση του Arduino σε breadboard

Για την υλοποίηση του arduino στο breadboard χρειαζόμαστε:

Table 1

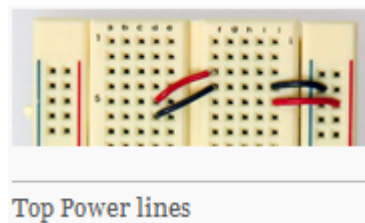
Breadboard
7805 Voltage regulator
2 LEDs
2 220 Ohm resistors
10k Ohm resistor
2 10 uF capacitors
16 MHz clock crystal

2 22 pF capacitors
small momentary normally open ("off") button
Atmel Atmega8/168/328/328P AVR microcontroller
FT232 USB Breakout board



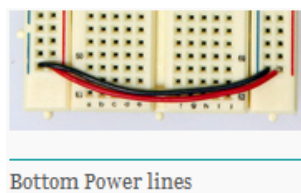
Εικόνα 1: Arduino σε breadboard: Τα εξαρτήματα του arduino (<http://arduino.cc/en/Main/Standalone>)

Προσθέτουμε τα καλώδια της τάσης (power) και της γείωσης (ground) στο σημείο που θα τοποθετηθεί ο ρυθμιστής τάσης (voltage regulator).



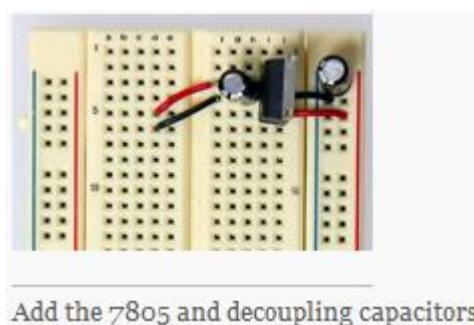
Εικόνα 2: Arduino σε breadboard:Γραμμές τάσεις (<http://arduino.cc/en/Main/Standalone>)

Ενώνουμε τις γραμμές της τάσης και της γείωσης με αυτές του κάτω μέρους του breadboard, ώστε να αναπτύξουμε το arduino στο χώρο του breadboard αξιοποιώντας καλύτερα το χώρο.



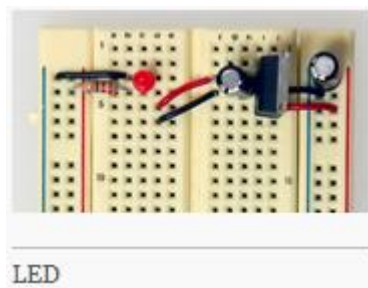
**Εικόνα 3: Arduino σε breadboard: Ένωση των γραμμών τάσης του πάνω με του κάτω μέρους**  
(<http://arduino.cc/en/Main/Standalone>)

Προσθέτουμε τον ρυθμιστή τάσης (7805 Voltage regulator), ο ρυθμιστής είναι της συσκευασίας TO-220 που σημαίνει ότι στην αριστερή είσοδο είναι η είσοδος της τάσης από την μπαταρία, στη μεσαία είσοδο είναι η γείωση και στα δεξιά είναι η έξοδος της επιθυμητής τάσης 5V (τάση λειτουργίας του μικρό-ελεγκτή). Τοποθετούμε 2 πυκνωτές των 10uF, ώστε να σταθεροποιούν την τάση σε περίπτωση αυξομειώσεων και να παρέχεται έτσι στον μικρό-ελεγκτή σταθερή τάση. Ένας μεταξύ της εισόδου στο αριστερό μέρος του ρυθμιστή και της γείωσης και ο δεύτερος μεταξύ της εξόδου στο δεξί μέρος με τη γείωση. Η ασημένια λωρίδα με το σημάδι του μείον σηματοδοτεί το μέρος του πυκνωτή που συνδέεται στη γείωση.



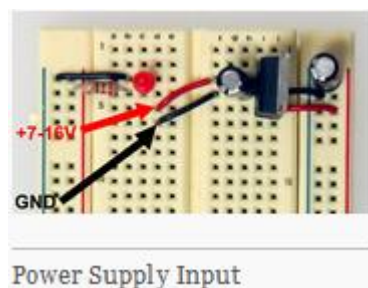
**Εικόνα 4: Arduino σε breadboard: Προσθήκη του 7805 και των πυκνωτών**  
(<http://arduino.cc/en/Main/Standalone>)

Προσθέτουμε έπειτα ένα LED και μια αντίσταση των 220-ohm όπως φαίνεται στην παρακάτω εικόνα που θα δείχνει αν το κύκλωμα τροφοδοτείται με ενέργεια.



Εικόνα 5: Arduino σε breadboard: LED ένδειξης τροφοδοσίας (<http://arduino.cc/en/Main/Standalone>)

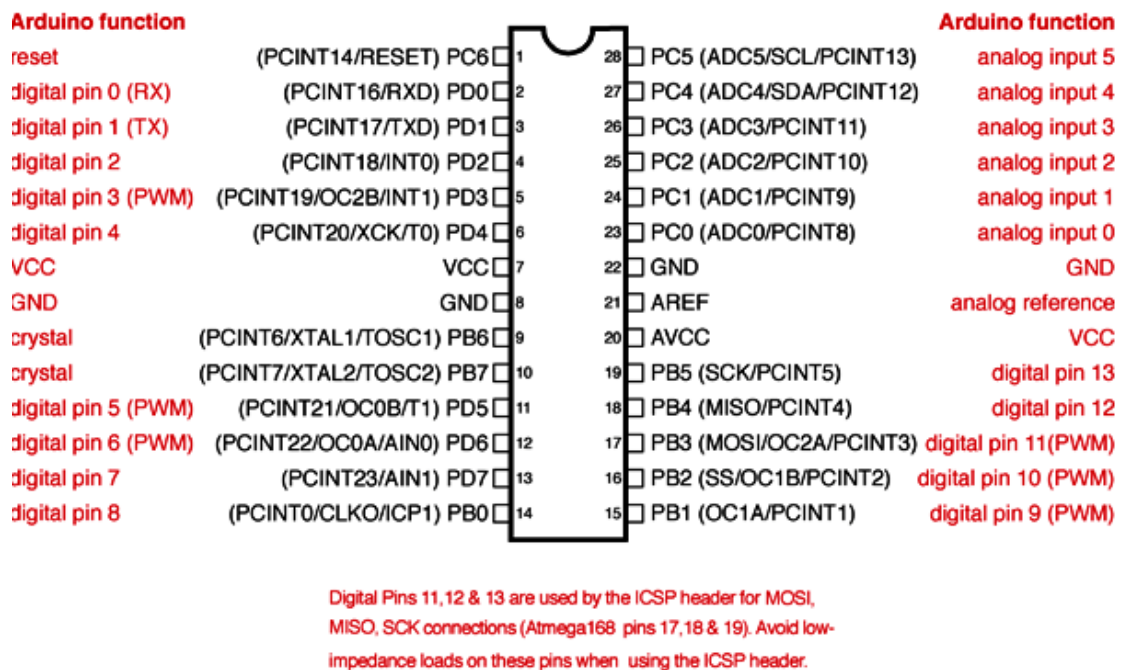
Τα κόκκινα και μαύρα καλώδια υποδεικνύουν που βρίσκεται η τάση και που η γείωση αντίστοιχα. Τα καλώδια αυτά στα αριστερά του ρυθμιστή τάσης είναι το σημείο που θα συνδεθεί η μπαταρία η οποία θα πρέπει να είναι μεταξύ των 7-16V, ιδανικά χρησιμοποιούμε μια μπαταρία των 9V.



Εικόνα 6: Arduino σε breadboard:Είσοδος τροφοδοσίας του κυκλώματος από μπαταρία (<http://arduino.cc/en/Main/Standalone>)

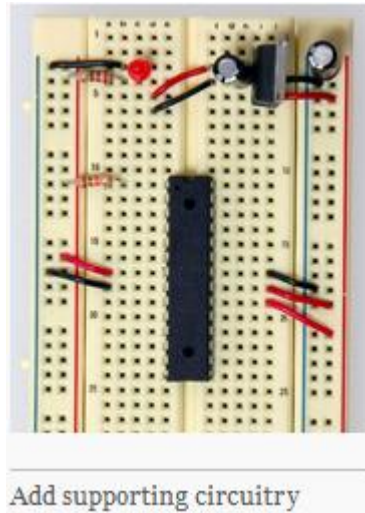
Τώρα που η τροφοδοσία του κυκλώματος είναι έτοιμη προχωράμε στη σύνδεση του μικρό-επεξεργαστή.

Το pin mapping για τον atmega 328P δείχνεται παρακάτω:



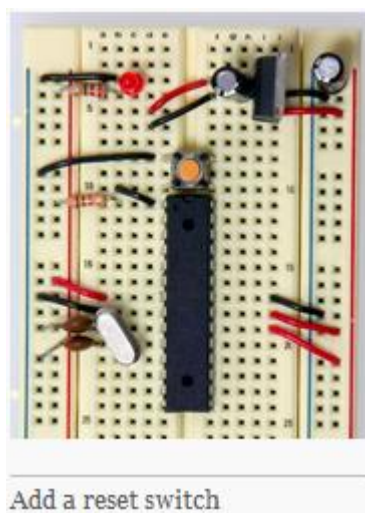
Εικόνα 7: Atmega 328P pin mapping (<http://arduino.cc/en/Main/Standalone>)

Συνδέουμε στο reset μια άκρη της αντίστασης 10k ohm και την άλλη στην τάση των 5V η οποία θα προστατεύει τον μικρό-επεξεργαστή από ανεπιθύμητη επανεκκίνηση. Όταν το reset συνδεθεί με τη γείωση τότε έχουμε επανεκκίνηση του μικρό-επεξεργαστή. Οι συνδέσεις στα ακόλουθα pins (ποδαράκια) (**Pin 7** - Vcc - *Digital Supply Voltage* **Pin 8** - GND **Pin 22** – GND **Pin 21** - AREF - *Analog reference pin for ADC* **Pin 20** - AVcc) φαίνονται στην παρακάτω εικόνα:



**Εικόνα 8: Arduino σε breadboard: Προσθήκη μικρό-επεξεργαστή και της τροφοδοσίας του**  
 (<http://arduino.cc/en/Main/Standalone>)

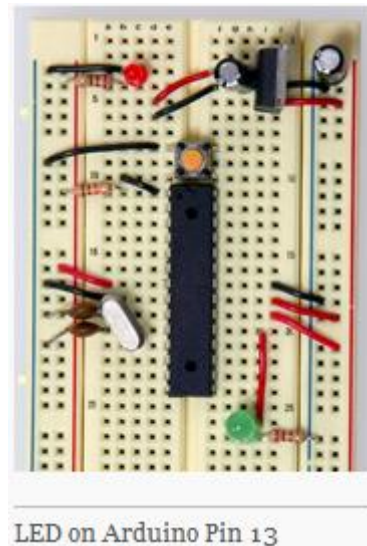
Τοποθετούμε ένα 16 MHz κρύσταλλο (παρέχει στο μικρό-ελεγκτή το ρολόι εισόδου (clock input)) μεταξύ του pin 9 και του pin 10 και ανάμεσα σε κάθε pin από αυτά και την γείωση ένα πυκνωτή 22 pF. Ακόμη τοποθετούμε ένα κουμπί το οποίο κάνει επανεκκίνηση στον μικρό-επεξεργαστή με ένα απλό πάτημα. Συνδέουμε το reset pin του μικρό-επεξεργαστή με το pin του κουμπιού και το άλλο pin του κουμπιού με τη γείωση όπως φαίνεται στην εικόνα.



**Εικόνα 9: Arduino σε breadboard: Προσθήκη κουμπιού reset** (<http://arduino.cc/en/Main/Standalone>)

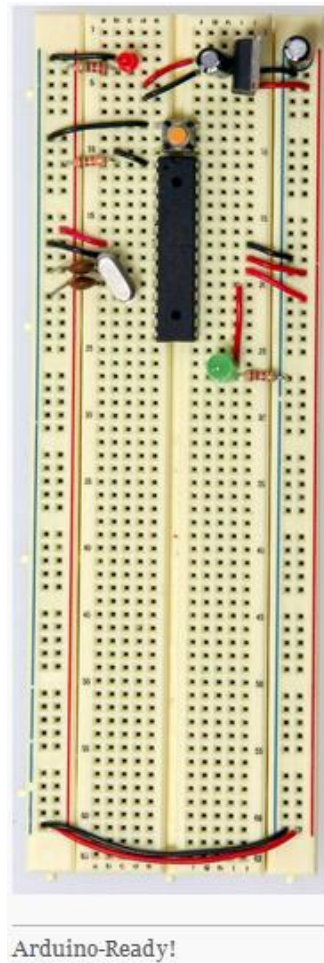


Έπειτα προσθέτουμε στο pin 13 ένα LED με την άνοδο συνδεδεμένη στο pin και την κάθοδο συνδεδεμένη στη γείωση με την προσθήκη ενδιάμεσα μιας αντίστασης, είναι απαραίτητο για να λειτουργήσουν μερικά παραδείγματα του arduino και χρησιμοποιήθηκε αρκετά για επίλυση προβλημάτων κατά τη διάρκεια ανάπτυξης του κώδικα.



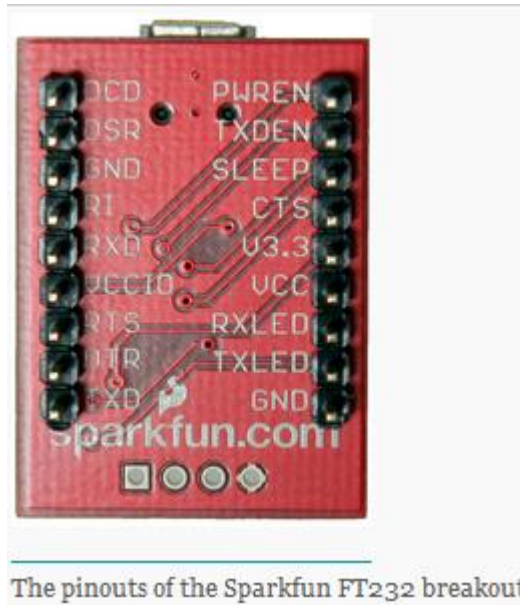
**Εικόνα 10: Arduino σε breadboard: Προσθήκη LED στο pin 13 (<http://arduino.cc/en/Main/Standalone>)**

Τέλος το arduino θα πρέπει να είναι όπως της εικόνας:



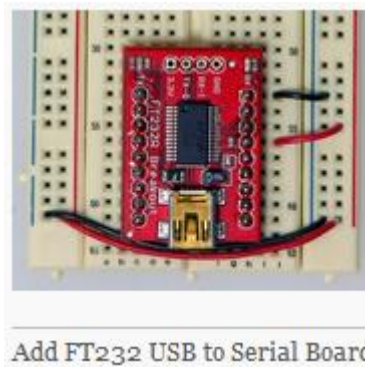
**Εικόνα 11: Arduino σε breadboard:Arduino στο breadboard (<http://arduino.cc/en/Main/Standalone>)**

Για την επικοινωνία του arduino με τον υπολογιστή χρειαζόμαστε τον προσαρμογέα usb (χρησιμοποιείται για τον προγραμματισμό του μικροελεγκτή και την αποστολή και λήψη δεδομένων μέσω της σειριακής θύρας) που φαίνεται στην εικόνα:



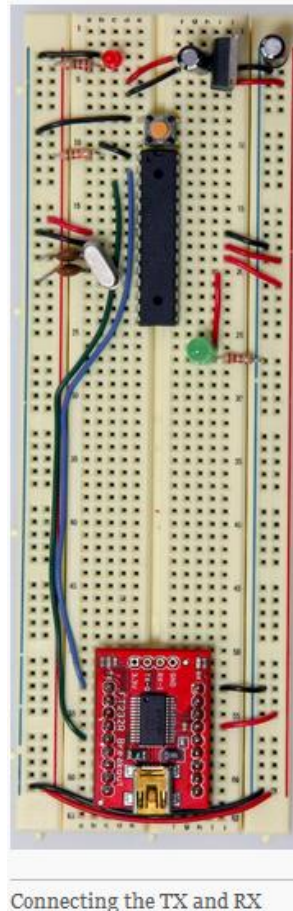
Εικόνα 12: FT232 pin mapping (<http://arduino.cc/en/Main/Standalone>)

Η σύνδεση του στην τάση και στη γείωση είναι:



Εικόνα 13: Arduino σε breadboard:Σύνδεση της τροφοδοσίας του FT232  
(<http://arduino.cc/en/Main/Standalone>)

Για την επικοινωνία με το arduino συνδέουμε το Tx σήμα του usb με το Rx (pin 2) του arduino και το Rx του usb με το Tx (pin 3) του arduino.



**Εικόνα 14: Arduino σε breadboard:Σύνδεση καλωδίων για τα σήματα Rx, Tx**  
(<http://arduino.cc/en/Main/Standalone>)

Αν ο μικρό-επεξεργαστής δεν είναι προγραμματισμένος πρέπει να του περάσουμε τον bootloader πριν μπορέσουμε να τον χρησιμοποιήσουμε για τις εφαρμογές μας.

Χρησιμοποιώντας το arduino mega ως ISP μπορούμε να περάσουμε στο arduino που φτιάξαμε τον arduino uno bootloader. Συνδέουμε τα pins του arduino στο breadboard με αυτά του arduino mega ως εξής:

Slave reset - pin 'reset' του arduino στο breadboard συνδέεται στο pin 53 του arduino mega, MOSI - pin 11 του arduino στο breadboard συνδέεται στο pin 51 του arduino mega, MISO - pin 12 του arduino στο breadboard συνδέεται στο pin 50 του arduino mega, SCK - pin 13 του arduino στο breadboard συνδέεται στο pin 52 του arduino mega. Μέσω

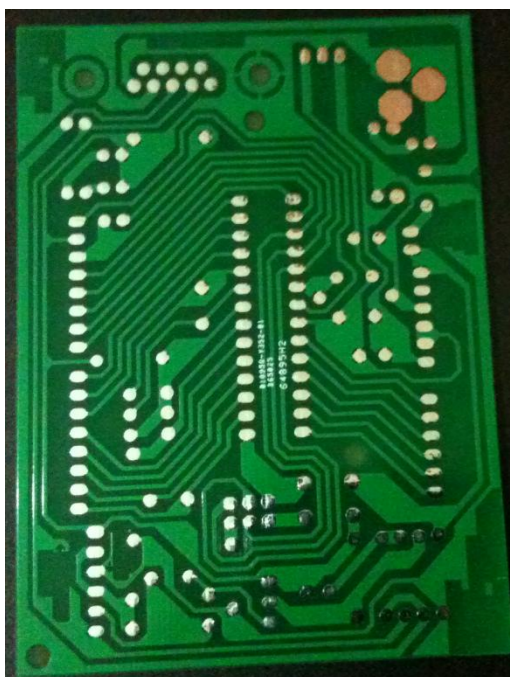
του περιβάλλοντος του arduino, προγραμματίζοντας το arduino mega ως ISP, περνάμε στο arduino που φτιάξαμε τον bootloader και είναι έτοιμο για χρήση.

#### 4 Κατασκευή Arduino σε PCB

Για την κατασκευή του arduino σε πλακέτα (PCB) χρησιμοποιήθηκε ο οδηγός από την ιστοσελίδα του arduino ([www.arduino.cc](http://www.arduino.cc), built by hand Arduino Single-Sided Serial Board (version 3)). Το PCB που χρησιμοποιήθηκε είχε ήδη τυπωμένο το λογικό κύκλωμα.

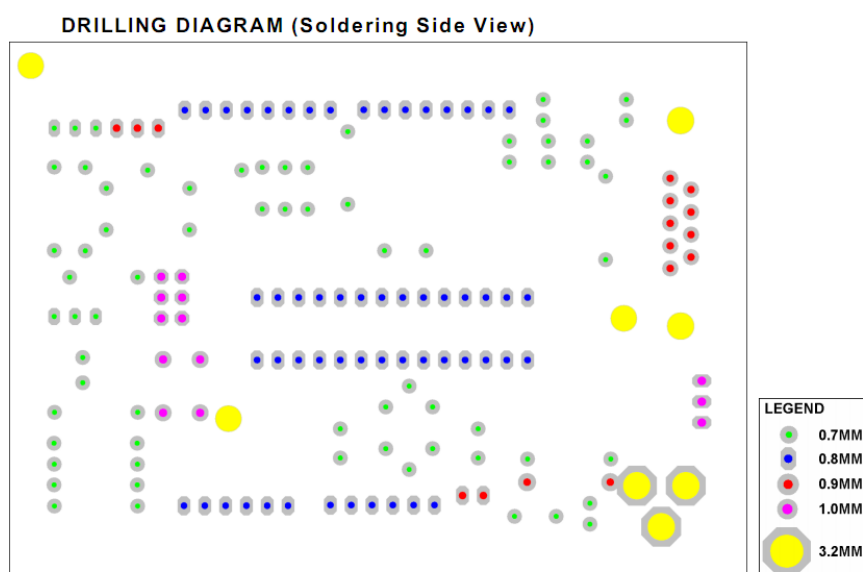


Εικόνα 15: Arduino top view



Εικόνα 16: Arduino bottom view

Τρυπήθηκε σύμφωνα με το πλάνο της arduino:



Εικόνα 17: Διάγραμμα τρυπήματος

Τα εξαρτήματα που απαιτούνται είναι:



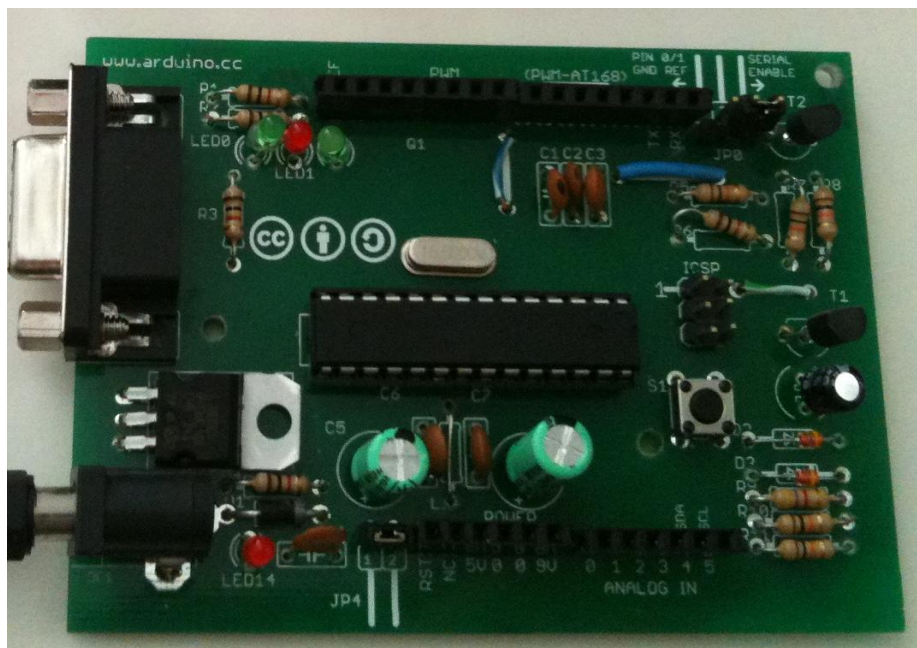
PART LIST FOR ARDUINO SERIAL SINGLE SIDED VERSION 3 (S3V3) - REVISION 2		
POSITION	VALUE	DESCRIPTION
C1	22pF (22 pico Farad)	ceramic disc capacitor
C2	22pF (22 pico Farad)	ceramic disc capacitor
C3	100nF (100 nano Farad - or 0.1 micro Farad)	ceramic or polyester capacitor
C4	100nF (100 nano Farad - or 0.1 micro Farad)	ceramic or polyester capacitor
C5	100μF (100 micro Farad)	electrolytic capacitor
C6	100nF (100 nano Farad - or 0.1 micro Farad)	ceramic or polyester capacitor
C7	100nF (100 nano Farad - or 0.1 micro Farad)	ceramic or polyester capacitor
C8	100μF (100 micro Farad)	electrolytic capacitor
C9	10μF (10 micro Farad)	non-polarized electrolytic capacitor
D1	1N4004	diode
D2	1N4148	diode
D3	1N4148	diode
DC1		2.1mm. DC power jack
IC1		ATMEGA8 (or ATMEGA168)
IC2	7805C	Tension Regulator
ICSP	2x3	male pin header
J1	1x8	female pin header
J2	1x6	female pin header
J3	1x8	female pin header
JP0	1x3	right angle pin header
JP4	1x2	right angle pin header
L1	100μH	leaded inductor
LED0	3 mm.	LED
LED1	3 mm.	LED
LED13	3 mm.	LED
LED14	3 mm.	LED
POWER	1x6	female pin header
Q1		16 MHz crystal
R1	1kohm (1.0 kilo ohm)	Resistor
R2	1kohm (1.0 kilo ohm)	Resistor
R3	1kohm (1.0 kilo ohm)	Resistor
R4	1kohm (1.0 kilo ohm)	Resistor
R5	10kohms (10.0 kilo ohms)	Resistor
R6	1kohm (1.0 kilo ohm)	Resistor
R7	10kohms (10.0 kilo ohms)	Resistor
R8	10kohms (10.0 kilo ohms)	Resistor
R9	4k7ohms (4.7 kilo ohms)	Resistor
R10	10kohms (10.0 kilo ohms)	Resistor
R11	10kohms (10.0 kilo ohms)	Resistor
S1	6x6 mm., 4 terminals	Switch Tactile
T1	BC547	Transistor
T2	BC557	Transistor
X1	9 PIN FEMALE RIGHT ANGLE PC MOUNT	D-SUB CONNECTOR
jumper		0.1" (or 2.54 mm.)
jumper		0.1" (or 2.54 mm.)

Εικόνα 18: Λίστα εξαρτημάτων

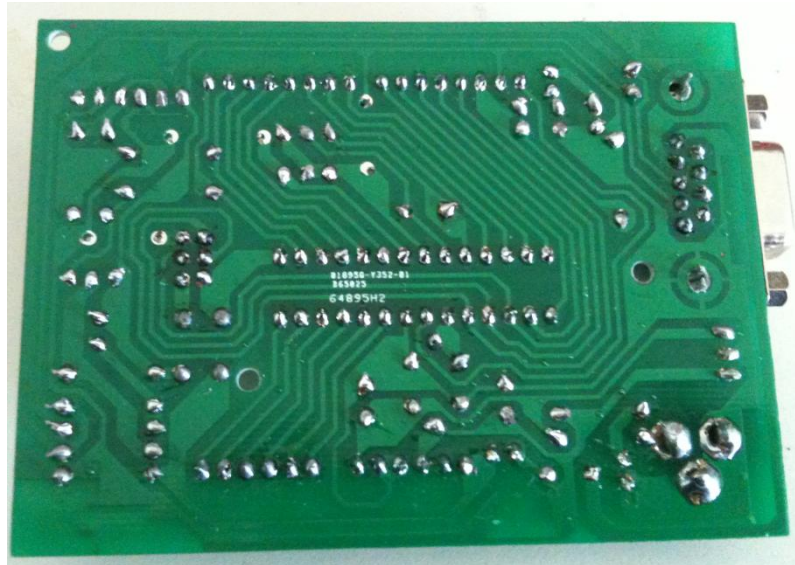
Τα εξαρτήματα που αναφέρθηκαν κολλήθηκαν στο PCB με την τεχνική του soldering με βάση την ακόλουθη εικόνα:



Το τελικό arduino που κατασκευάστηκε δείχνεται ακολούθως:

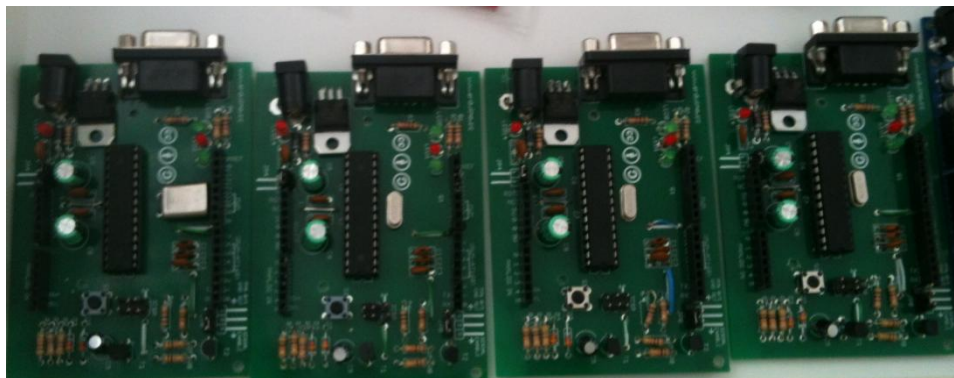


23



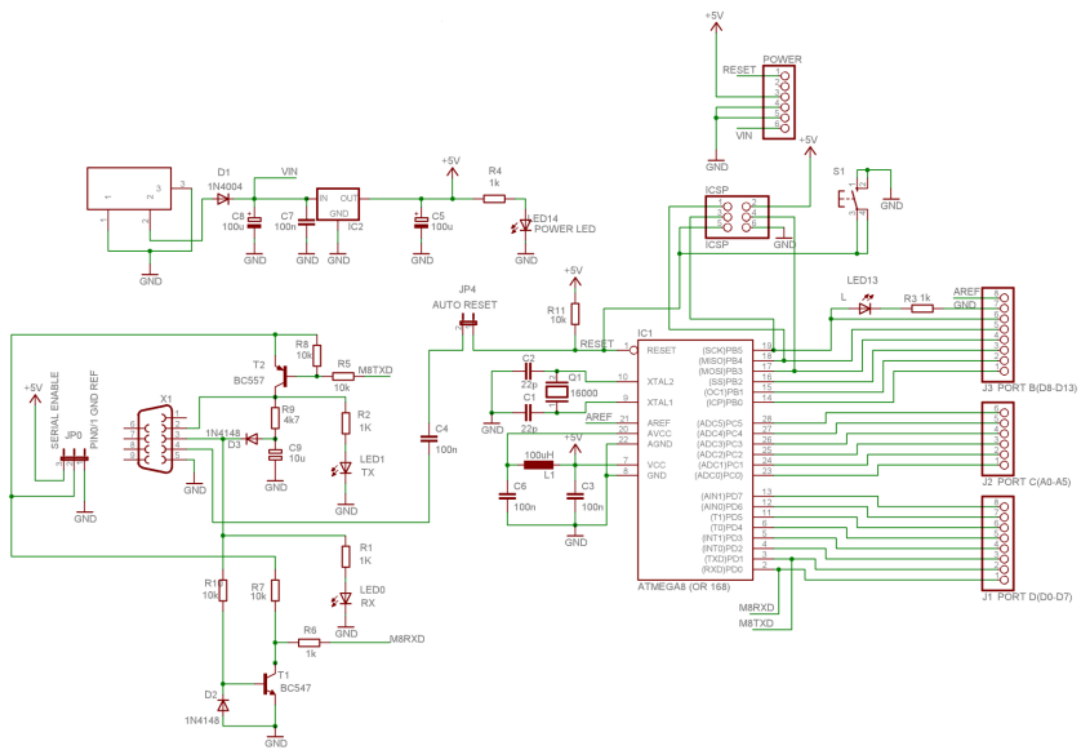
**Εικόνα 21: Arduino S3V3 (Revision 2) bottom view**

Με τον τρόπο αυτό δημιουργήθηκαν συνολικά τέσσερα όμοια arduino.



**Εικόνα 22: Τα arduino που κατασκευάστηκαν στο εργαστήριο ηλεκτρονικής**

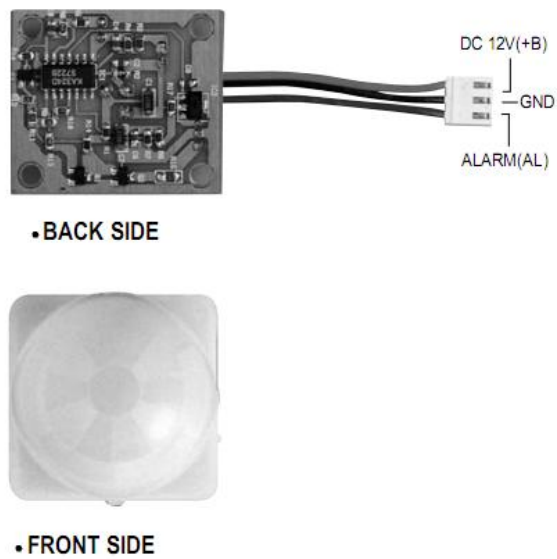
Το ηλεκτρολογικό σχέδιο των παραπάνω arduino είναι το εξής:



**Εικόνα 23: Το ηλεκτρολογικό σχέδιο του arduino**  
<http://arduino.cc/en/Main/ArduinoBoardSerialSingleSided3>

## 5 Εξοπλισμός που τοποθετείται στον μικροελεγκτή

### 5.1 Αισθητήρας Παρουσίας (PIR sensor)



Εικόνα 24: PIR sensor

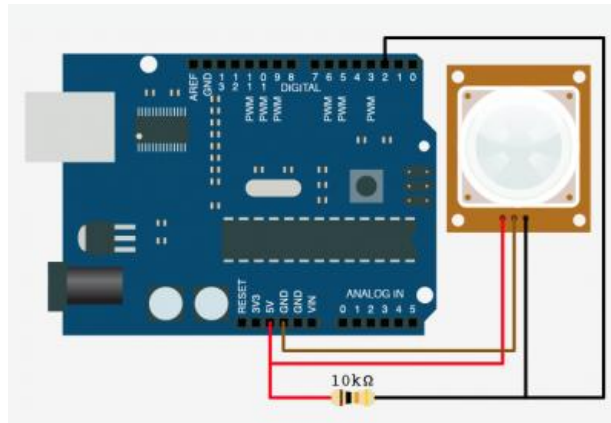
Specifications				
ITEM		Specification	Unit	Condition
Sensor Type		Dual Element		
Housing		TO 5		
Element Size		2×1	mm	
Spacing		1	mm	
Responsivity	Min Typ	3.2 4.0	xv/w	7...14mm, 1Hz, 100°C (One element cover)
Match	Max	<10	%	7...14mm, 1Hz, 100°C (Both element expose)
Noise	Typ Max	20 50	μVp-p V	25°C, 0.4...10Hz
Effect Voltage	Min Max	0.2 1.5		Re=47XO
Window Material		Silicon, coated		
Spectral Range	Transmission Blocking	T>30 average T<0.1	%	7...14mm <5mm
Operating Voltage		12	V	
Operating Temperature		-10~40	°C	
Storage Temperature		-40~80	°C	

Εικόνα 25: Λεπτομέρειες PIR sensor

Ο αισθητήρας παρουσίας του συστήματος που υλοποιήθηκε είναι ένας απλός στη χρήση αισθητήρας κίνησης. Όταν πάρει τάση από το κύκλωμα μετά από 1-2 δευτερόλεπτα «φωτογραφίζει» το δωμάτιο, εάν κάτι κινηθεί μέσα σ' αυτό το διάστημα το alarm pin θα ρίξει την τάση. Δουλεύει σε εύρος τάσης 5-12V. Το alarm pin είναι ανοιχτός συλλέκτης που σημαίνει ότι χρειάζεται μια αντίσταση μεταξύ του pin και της τάσης 5V.

Ο αισθητήρας παρουσίας βρίσκει εφαρμογή σε αντικλεπτικά συστήματα, συστήματα ασφαλείας, φωτισμό σκοτεινών χώρων, αυτόματες πόρτες, έξυπνα κινητά. Στην εφαρμογή αυτή χρησιμοποιείται για να ανιχνεύσει την ύπαρξη χρήστη στο δωμάτιο και να ενεργοποιήσει τον αλγόριθμο για να ξεκινήσει ο έλεγχος του φωτισμού και των σκιάστρων.

Μια τυπική σύνδεση ενός τέτοιου αισθητήρα με το arduino δείχνεται παρακάτω:



Εικόνα 26: Σύνδεση του PIR sensor στο arduino

## 5.2 Αισθητήρας φωτεινότητας



Εικόνα 27: Αισθητήρας φωτεινότητας (TEMT6000)

### Absolute Maximum Ratings

$T_{amb} = 25^{\circ}\text{C}$ , unless otherwise specified

Parameter	Test condition	Symbol	Value	Unit
Collector Emitter Voltage		$V_{CEO}$	6	V
Emitter Collector Voltage		$V_{ECO}$	1.5	V
Collector current		$I_C$	20	mA
Total Power Dissipation	$T_{amb} \leq 55^{\circ}\text{C}$	$P_{tot}$	100	mW
Junction Temperature		$T_j$	100	$^{\circ}\text{C}$
Operating Temperature Range		$T_{amb}$	- 40 to + 85	$^{\circ}\text{C}$
Storage Temperature Range		$T_{stg}$	- 40 to + 85	$^{\circ}\text{C}$
Soldering Temperature	$t \leq 3 \text{ s}$	$T_{sd}$	260	$^{\circ}\text{C}$
Thermal Resistance Junction/ Ambient		$R_{thJA}$	450	K/W

Εικόνα 28: Αξιολογήσεις του αισθητήρα

## Basic Characteristics

$T_{amb} = 25\text{ }^{\circ}\text{C}$ , unless otherwise specified

Parameter	Test condition	Symbol	Min	Typ.	Max	Unit
Collector Emitter Breakdown Voltage	$I_C = 0.1\text{ mA}$	$V_{CEO}$	6			V
Collector dark current	$V_{CE} = 5\text{ V}, E = 0$	$I_{CEO}$		3	50	nA
Collector-emitter capacitance	$V_{CE} = 0\text{ V}, f = 1\text{ MHz}, E = 0$	$C_{CEO}$		16		pF
Collector Light Current	$E_v = 20\text{ lx}$ , standard light A	$I_{ca}$	3.5	10	16	$\mu\text{A}$
	$E_v = 100\text{ lx}$ , standard light A	$I_{ca}$		50		$\mu\text{A}$
Angle of Half Sensitivity		$\phi$		$\pm 60$		deg
Wavelength of Peak Sensitivity		$\lambda_p$		570		nm
Range of Spectral Bandwidth		$\lambda_{0.1}$		360 to 970		nm
Collector Emitter Saturation Voltage	$E_v = 20\text{ lx}, 0.45\text{ }\mu\text{A}$	$V_{CEsat}$		0.1		V

Εικόνα 29: Βασικά χαρακτηριστικά του αισθητήρα

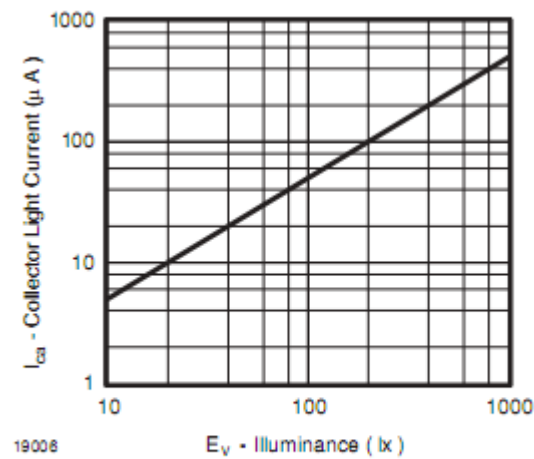


Figure 1. Collector Light Current vs. Illuminance

Εικόνα 30: Collector Light Current vs Illuminance

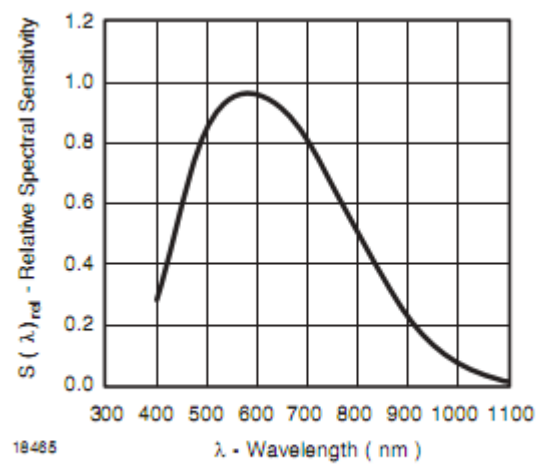


Figure 2. Relative Spectral Responsivity vs. Wavelength

Εικόνα 31: Relative Spectral Responsivity vs Wavelength

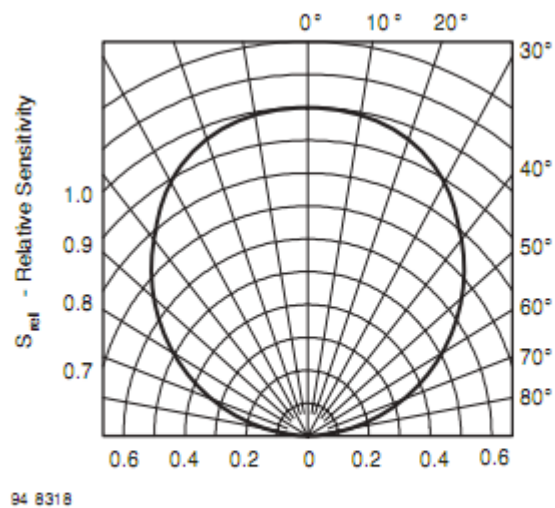


Figure 3. Relative Radiant Sensitivity vs. Angular Displacement

Εικόνα 32: Relative Radiant Sensitivity vs Angular Displacement



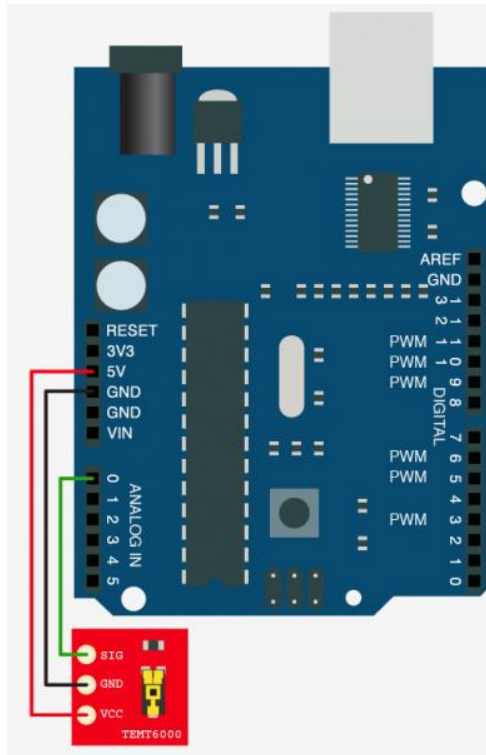


Εικόνα 33: Ηλεκτρολογικό διάγραμμα του TEMT6000

Ο αισθητήρας φωτεινότητας του συστήματος είναι ένα φωτοτρανζίστορ τύπου NPN. Λειτουργεί όπως ένα τυπικό τρανζίστορ, όσο μεγαλύτερο το προσπίπτον φως τόσο μεγαλύτερη τάση εμφανίζεται στο pin signal. Αξίζει να αναφερθεί ότι ο αισθητήρας είναι ευαίσθητος στο ορατό φως.

Ο αισθητήρας φωτεινότητας βρίσκει εφαρμογή σε συστήματα συναγερμού, εξοικονόμησης ενέργειας, ανίχνευσης φωτεινότητας. Στο σύστημα που υλοποιήθηκε χρησιμοποιείται ένας για την μέτρηση της φωτεινότητας του δωματίου, ένας δεύτερος για την μέτρηση της φωτεινότητας του εξωτερικού φυσικού φωτισμού και μέσα από τη σύγκριση των δύο τιμών το σύστημα προβαίνει σε ενέργειες για εξοικονόμηση ενέργειας και παροχής άνετου περιβάλλοντος εργασίας.

Μια τυπική σύνδεση ενός τέτοιου αισθητήρα με το arduino δείχνεται παρακάτω:



Εικόνα 34: Σύνδεση του TEMT6000 sensor στο arduino

### 5.3 Κάρτα μνήμης

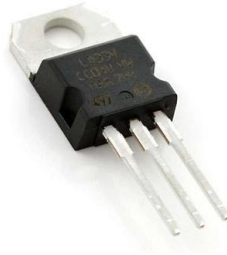
#### 5.3.1 Κύκλωμα κάρτας μνήμης στο breadboard

##### 5.3.1.1 Εξαρτήματα του κυκλώματος

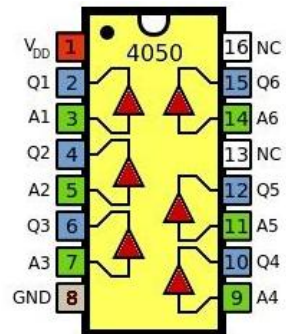
Για το κύκλωμα της κάρτας μνήμης χρειαζόμαστε:

Table 2

Voltage regulator 3V3
CD4050BCN level shifter
Breakout Board for SD-MMC Card
Δύο ηλεκτρολυτικούς πυκνωτές των 10uF



**Εικόνα 35: Voltage regulator 3V3**

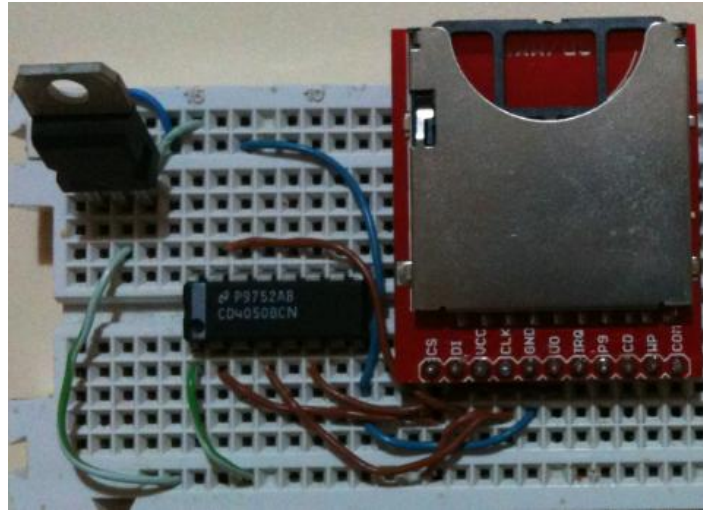


**Εικόνα 36: CD4050BCN level shifter**

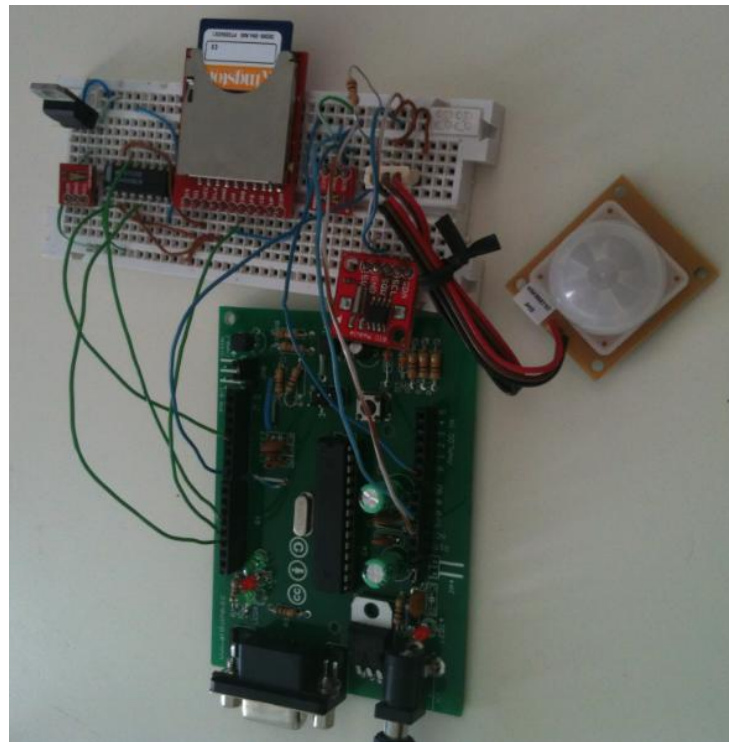


**Εικόνα 37: Breakout Board for SD-MMC Card**

### 5.3.1.2 Διάταξη στο breadboard



Εικόνα 38: Διάταξη του κυκλώματος της κάρτας μνήμης στο breadboard



Εικόνα 39: Σύνδεση της κάρτας μνήμης και των αισθητήρων με το arduino

Συνδέουμε το GND στο ground, τα 3.3V της κάρτας μνήμης στα 3.3V του ρυθμιστή τάσης, το CLK στο Pin 13 του Arduino, το DO στο pin 12, το DI στο pin 11, και το CS στο pin 4, αφού πρώτα τα περάσουμε μέσα από το level shifter.

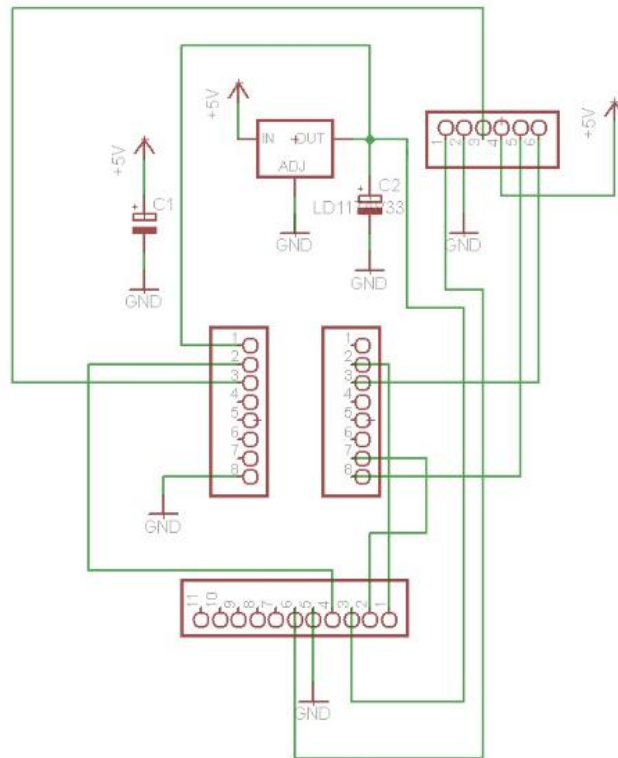
### 5.3.2 Προσαρμογές της κάρτας μνήμης στο PCB

Μέσω του προγράμματος Eagle της Cadsoft σχεδιάστηκε ένα PCB το οποίο ενσωματώνει το κύκλωμα της κάρτας μνήμης που αναπτύχθηκε στο breadboard. Παρακάτω δείχνονται το ηλεκτρολογικό σχέδιο του PCB και το ίδιο το PCB. Χρησιμοποιήθηκαν τα εξαρτήματα που τοποθετήθηκαν στο breadboard.

Table 3

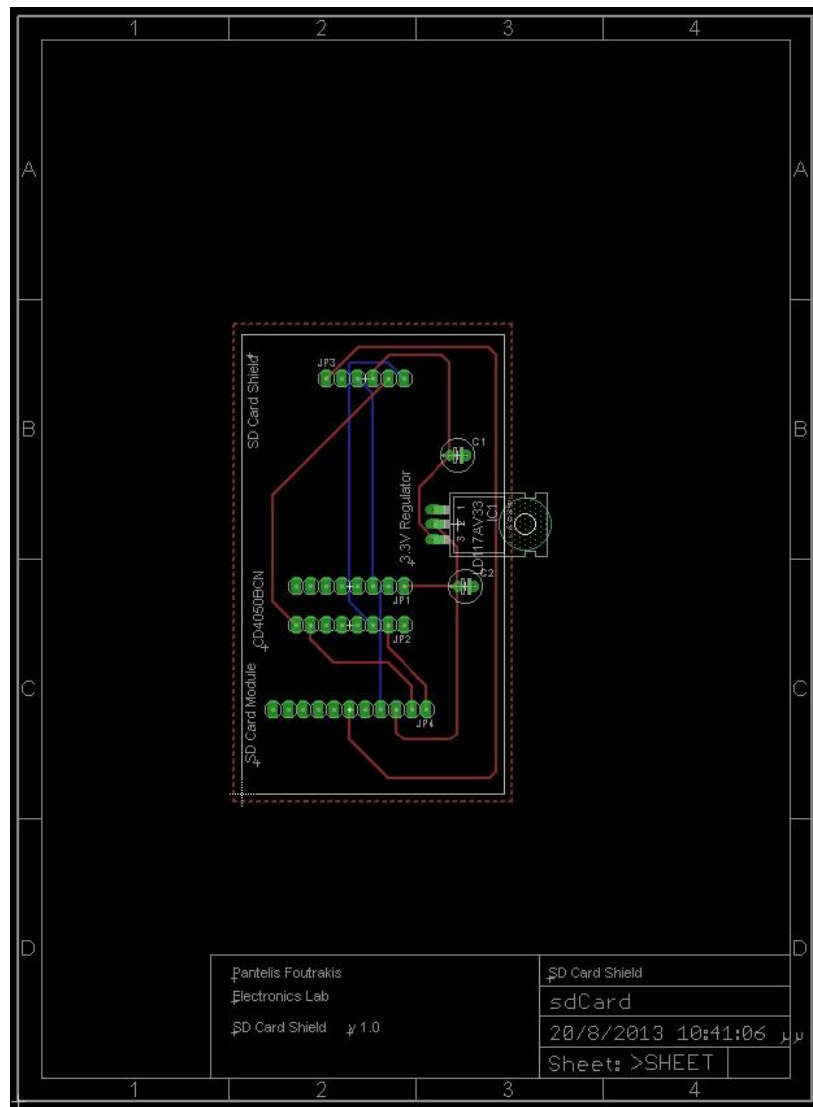
Voltage regulator 3V3
CD4050BCN level shifter
Breakout Board for SD-MMC Card
Δύο ηλεκτρολυτικούς πυκνωτές των 10uF

Η σύνδεσή τους φαίνεται στο ηλεκτρολογικό σχέδιο του PCB. Οι πυκνωτές χρησιμοποιήθηκαν για να κρατούν σταθερή την τάση στα 3.3V. Ο CD4050BCN level shifter τοποθετήθηκε για την μετατροπή των σημάτων του arduino τάσης 5V σε τάση 3.3V που απαιτεί η κάρτα μνήμης. Εξαίρεση αποτελεί το σήμα DO (έξοδος δεδομένων) της κάρτας μνήμης το οποίο συνδέεται χωρίς να περάσει από τον CD4050BCN level shifter στο arduino.



**Εικόνα 40: Ηλεκτρολογικό σχέδιο του PCB**

Τα στοιχεία του PCB διατάχθηκαν με τέτοιο τρόπο, ώστε να καταλαμβάνει το λιγότερο δυνατό χώρο και η σύνδεση των εξαρτημάτων να γίνεται χωρίς τις γραμμές του χαλκού να μπερδεύονται μεταξύ τους. Επιπλέον οι γραμμές του χαλκού σχεδιάστηκαν να στρίβουν υπό γωνία  $45^\circ$  αντί  $90^\circ$  για καλύτερη κατανομή του χώρου και να είναι ανθεκτικότερες σε περίπτωση ατυχήματος κατά την κατασκευή του υλικού. Ακόμη η σύνδεσή του στο shield που σχεδιάστηκε θα γίνεται υπό γωνία  $90^\circ$  ώστε να έχει το σύστημα λιγότερο ύψος.



Εικόνα 41: Προσαρμογέας της κάρτας μνήμης για σύνδεση με το shield

### 5.3.3 Λειτουργικότητα της κάρτας μνήμης

Στο σύστημα που αναπτύχθηκε η κάρτα μνήμης χρησιμοποιείται για αποθήκευση αρχείων απαραίτητων για τη λειτουργία του. Σ' αυτήν αποθηκεύονται σε μορφή αρχείων .txt οι κανόνες της ασαφούς λογικής (FL), οι συναρτήσεις μέλη (membership functions), οι έξοδοι της FL και η έξοδος του συστήματος. Η χρήση της κάρτας μνήμης είναι βολική για το χρήστη καθώς με λίγο κόπο μπορεί να ανακτήσει την κάρτα μνήμης, να επέμβει στα αρχεία και να αλλάξει τους κανόνες και τις συναρτήσεις

μέλη της FL και να επεξεργαστεί τα δεδομένα που παράχθηκαν από το σύστημα.

#### **5.4 Προσαρμογέας XBee**

Επιλέχθηκε η χρήση ενός προσαρμογέα XBee και συνεπώς η χρήση του πρωτοκόλλου ZigBee για την ασύρματη επικοινωνία του συστήματος λόγω των πλεονεκτημάτων που προσφέρει. Ο προσαρμογέας XBee είναι γνωστός για την χαμηλή κατανάλωση ενέργειας του, το χαμηλό κόστος των πομποδεκτών του και το δωρεάν στη χρήση πρωτόκολλό του, την υψηλή αξιοπιστία του (υπάρχει μέθοδος αποφυγής των συγκρούσεων των πακέτων πληροφορίας και επιβεβαίωση της σωστής λήψης τους), την ασφάλεια που παρέχει στην επικοινωνία (χρησιμοποιεί κρυπτογράφηση AES-128 των μηνυμάτων και των κλειδιών ασφαλείας), μικρός χρόνος καθυστέρησης (30 ms για την εύρεση συσκευών, 15 ms για την αναμονή ενεργοποίησης, 15 ms για την πρόσβαση στο κανάλι των ενεργών συσκευών) και μεγάλα μεγέθη δικτύων (ένα δίκτυο Xbee που υλοποιούν το πρωτόκολλο ZigBee έχει ένα συντονιστή και μπορεί να έχει μέχρι και 254 ακόμα συσκευές).

##### **5.4.1 Περιγραφή πρωτοκόλλου zigbee**

Το πρωτόκολλο zigbee είναι ένα πρότυπο χαμηλού κόστους, χαμηλής ισχύος άτακτου ασύρματου δικτύου. Το χαμηλό κόστος επιτρέπει στην τεχνολογία να εφαρμοστεί ευρέως στον ασύρματο έλεγχο και στην παρακολούθηση εφαρμογών. Η χαμηλή ισχύς επιτρέπει την μεγαλύτερη διάρκεια ζωής των μπαταριών και μικρότερες μπαταρίες. Η άτακτη δομή του δικτύου παρέχει μεγαλύτερη αξιοπιστία και μεγαλύτερη εμβέλεια. Οι κατασκευαστές υλικού, που εφαρμόζει το πρωτόκολλο zigbee, τυπικά



συμπεριλαμβάνουν ενσωματωμένους πομποδέκτες και μικρό-επεξεργαστές με μνήμη flash μεταξύ των 60 και των 256 KB. Το πρωτόκολλο zigbee λειτουργεί σε βιομηχανικές, επιστημονικές και ιατρικές ζώνες συχνοτήτων, 868 MHz στην Ευρώπη, 915 MHz στις ΗΠΑ και την Αυστραλία και 2.4 GHz σε ορισμένες περιοχές στον κόσμο. Η μετάδοση δεδομένων ποικίλει από 20 kilobits/second στη ζώνη συχνοτήτων των 868 MHz σε 250 kilobits/second στη ζώνη συχνοτήτων 2.4 GHz. Το επίπεδο δικτύου του πρωτόκολλου zigbee υποστηρίζει τόσο τοπολογία αστέρα και δενδρική δομή, όπως και γενικευμένα άτακτα δίκτυα. Κάθε δίκτυο πρέπει να έχει ένα πομποδέκτη στο ρόλο του συντονιστή (coordinator) με την αρμοδιότητα να δημιουργεί, να ελέγχει τις παραμέτρους και να συντηρεί το δίκτυο. Σε τοπολογίες αστέρα ο coordinator πρέπει να είναι ο κεντρικός κόμβος. Σε δενδρικές δομές και σε άτακτα δίκτυα επιτρέπεται η χρήση πομποδεκτών στο ρόλο του router, οι οποίοι επεκτείνουν την επικοινωνία στο επίπεδο δικτύου. Το πρωτόκολλο zigbee βασίζεται στο φυσικό επίπεδο και στον έλεγχο πρόσβασης στο μέσο, όπως ορίζονται στο πρότυπο IEEE 802.15.4 για χαμηλού ρυθμού WPAN. Το πρωτόκολλο zigbee ολοκληρώνει το πρότυπο προσθέτοντας 4 κύρια μέρη: το επίπεδο δικτύου, το επίπεδο εφαρμογής, τα ZigBee device objects (ZDOs) και αντικείμενα εφαρμογών ορισμένα από τον κατασκευαστή τα οποία επιτρέπουν διαμόρφωση και πλήρη ενσωμάτωση. Η πιο σημαντική βελτίωση είναι η παρουσίαση των ZDOs, τα οποία είναι υπεύθυνα για πολλές λειτουργίες. Αυτές είναι η ανάθεση ρόλων στις συσκευές, ο έλεγχος των αιτήσεων για σύνδεση στο δίκτυο, η εύρεση συσκευών και η ασφάλεια. Το πρωτόκολλο zigbee δεν έχει σκοπό να υποστηρίξει δικτύωση μέσω γραμμών τάσης, αλλά τουλάχιστον να αλληλεπιδρά για ανάγκες έξυπνης μέτρησης και έξυπνων οικιακών συσκευών. Επειδή οι κόμβοι του zigbee μπορούν να μεταβούν από κατάσταση ύπνου σε ενεργή σε λιγότερο από

30 ms η καθυστέρηση είναι μικρή και οι συσκευές ανταποκρίνονται καλύτερα σε σχέση με το Bluetooth όπου η καθυστέρηση είναι περίπου 3 δευτερόλεπτα. Οι κόμβοι του zigbee μπορούν να κοιμούνται την περισσότερη ώρα, επομένως η κατανάλωση ενέργειας είναι χαμηλή με αποτέλεσμα την μεγάλη διάρκεια ζωής της μπαταρίας.

Το πρωτόκολλο zigbee είναι ιδανικό για ενσωματωμένες εφαρμογές που απαιτούν χαμηλό ρυθμό δεδομένων και χαμηλή κατανάλωση ενέργειας. Τυπικές περιπτώσεις που η χρήση του πρωτοκόλλου είναι ιδανική δίνονται ακολούθως: αυτοματισμοί σπιτιών, έξυπνος φωτισμός, ασφάλεια και προστασία, ταινίες και μουσική, βιομηχανικός έλεγχος, ενσωματωμένοι αισθητήρες, συλλογή ιατρικών δεδομένων, συναγερμοί καπνού και εισβολέων, αυτοματισμοί κτιρίων.

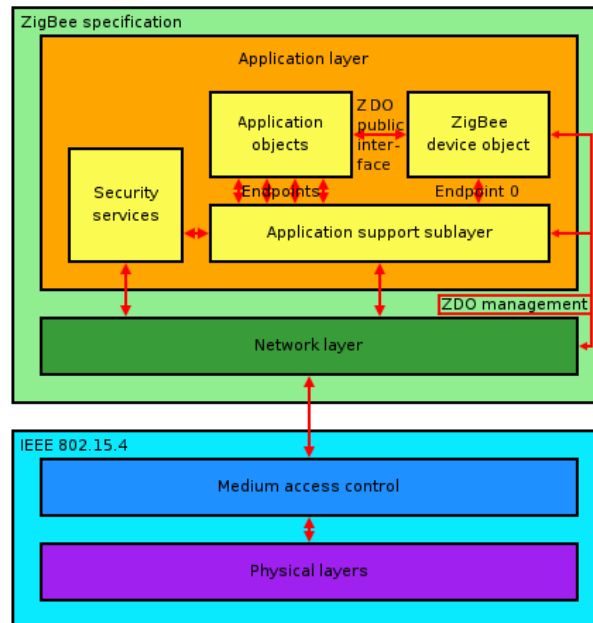
Το πρωτόκολλο zigbee έχει συσκευές 3 διαφορετικών τύπων:

- ZigBee Coordinator (ZC): η πιο ικανή συσκευή, αποτελεί τη βάση του δικτύου και μπορεί να γεφυρώνει επικοινωνία σε διαφορετικά δίκτυα μεταξύ τους. Υπάρχει μόνο ένας στο δίκτυο, καθώς είναι η συσκευή που ξεκίνησε το δίκτυο. Αποθηκεύει πληροφορίες για το δίκτυο και είναι υπεύθυνος για τα κλειδιά ασφαλείας.
- ZigBee Router (ZR): μπορεί να λειτουργήσει ως μεσολαβητής και να μεταφέρει δεδομένα σε άλλες συσκευές.
- ZigBee End Device (ZED): έχει αρκετή λειτουργικότητα ώστε να επικοινωνεί με τον κόμβο γονέα (Coordinator ή Router) και δεν μπορεί να μεταδώσει δεδομένα σε άλλες συσκευές εκτός αυτού. Αυτή η σχέση γονέα-παιδιού επιτρέπει στη συσκευή να είναι σε κατάσταση ύπνου, όταν υπάρχουν δεδομένα για μετάδοση η συσκευή γονιός θα αφυπνίσει τη συσκευή παιδί για την επιτυχή

επικοινωνία. Η ZED απαιτεί τη λιγότερη μνήμη και γι' αυτό είναι οικονομικότερη στην κατασκευή της από τις άλλες συσκευές.

Για την επιτυχημένη επικοινωνία τους οι συσκευές πρέπει να χρησιμοποιούν την ίδια μορφή του δικτύου σε μορφή συμβάσεων. Οι συμβάσεις αυτές ταξινομούνται σε προφίλ. Η ανακάλυψη συσκευών μπορεί να γίνει με διαφορετικές μεθόδους ανάλογα με την διαθέσιμη πληροφορία. Όταν η διεύθυνση δικτύου είναι γνωστή, η IEEE διεύθυνση μπορεί να απαιτηθεί χρησιμοποιώντας αποκλειστική επικοινωνία μεταξύ των δύο συσκευών. Όταν δεν είναι, οι αναφορές μεταδίδονται σε όλους τους κόμβους. Οι ZED θα ανταποκριθούν με την ζητούμενη διεύθυνση, ενώ οι ZC, ZR συσκευές θα αποστείλουν και τις διευθύνσεις των κόμβων που συσχετίζονται με αυτές. Οι ZC διατηρούν πίνακα αναφοράς με τη δομή του δικτύου και τις διευθύνσεις των συσκευών. Ο πίνακας αναφοράς πρέπει να υπάρχει πάντα διαθέσιμος γι' αυτό και οι ZC έχουν μόνιμη παροχή ενέργειας. Η επικοινωνία μπορεί να συμβεί αμέσως μετά τη συσχέτιση. Η άμεση διευθυνσιοδότηση χρησιμοποιεί τις φυσικές διευθύνσεις των πομποδεκτών, ενώ η έμμεση χρησιμοποιεί σχετικά πεδία τα οποία στέλνονται στον coordinator για μετάφραση μέσω του πίνακα αναφοράς για την επικοινωνία. Εκτός από αυτούς τους τρόπους επικοινωνίας υπάρχει η δυνατότητα να γίνει αναμετάδοση σ' όλους τους κόμβους ή σε μια ομάδα αυτών.

Ένα από τα καθοριστικά χαρακτηριστικά του zigbee είναι η δημιουργία ασφαλούς επικοινωνίας, η προστασία της εγκατάστασης και της μεταφοράς των κρυπτογραφημένων κλειδιών και η κρυπτογράφηση των μηνυμάτων και των συσκευών.



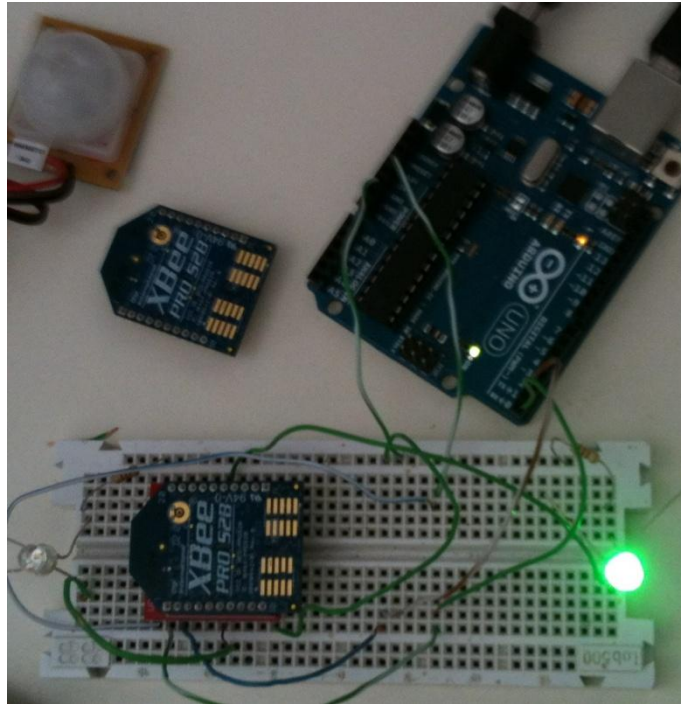
Εικόνα 42: Τα επίπεδα του πρωτοκόλλου ZigBee

#### 5.4.2 Διάταξη XBee στο breadboard

Η σύνδεση του XBee στο breadboard και αυτού με το arduino φαίνεται στις ακόλουθες εικόνες:

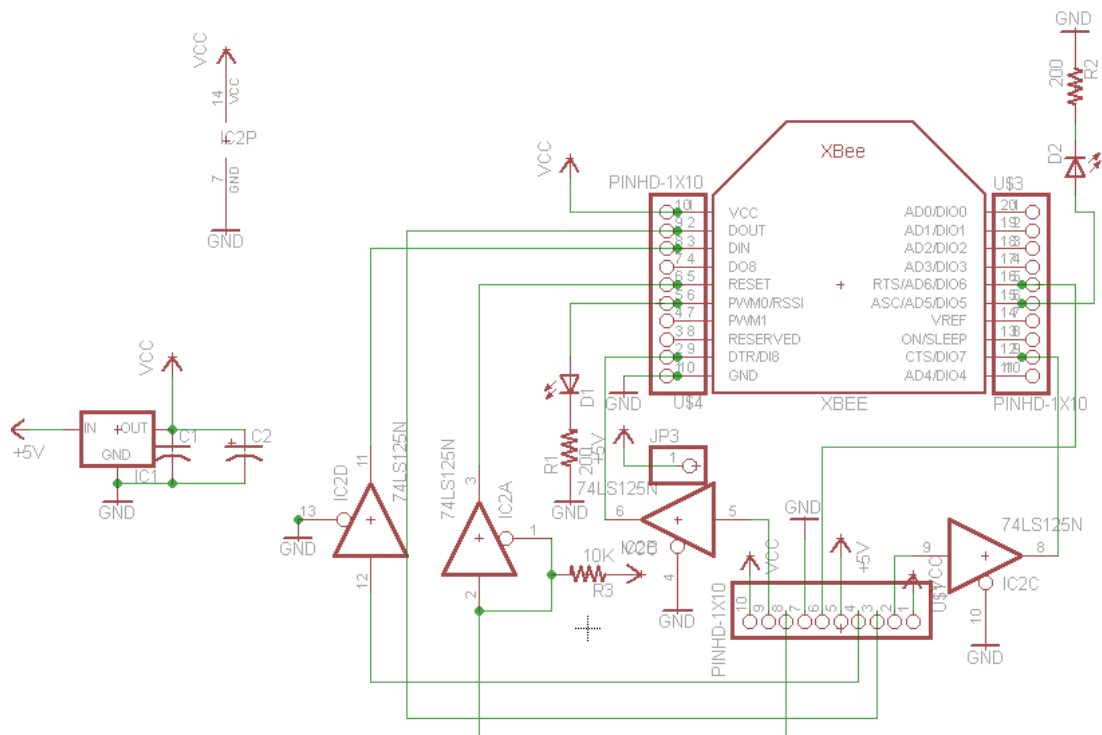


Εικόνα 43: Τοποθέτηση του XBee στο breadboard

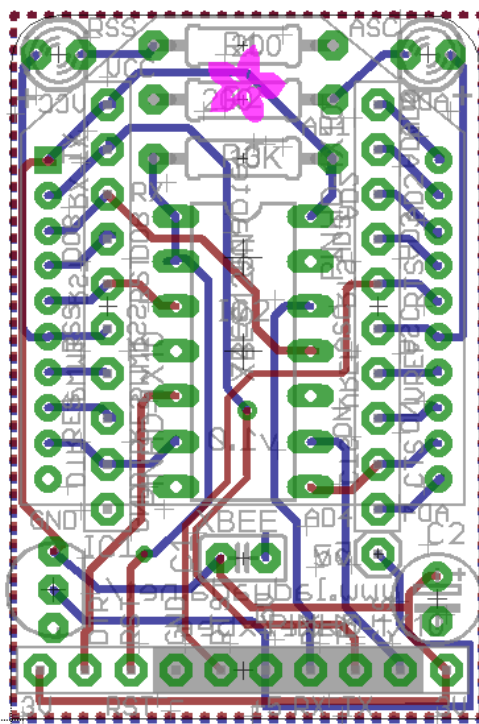


Εικόνα 44: Σύνδεση του XBee με το arduino

### 5.4.3 Προσαρμογές XBee στο PCB

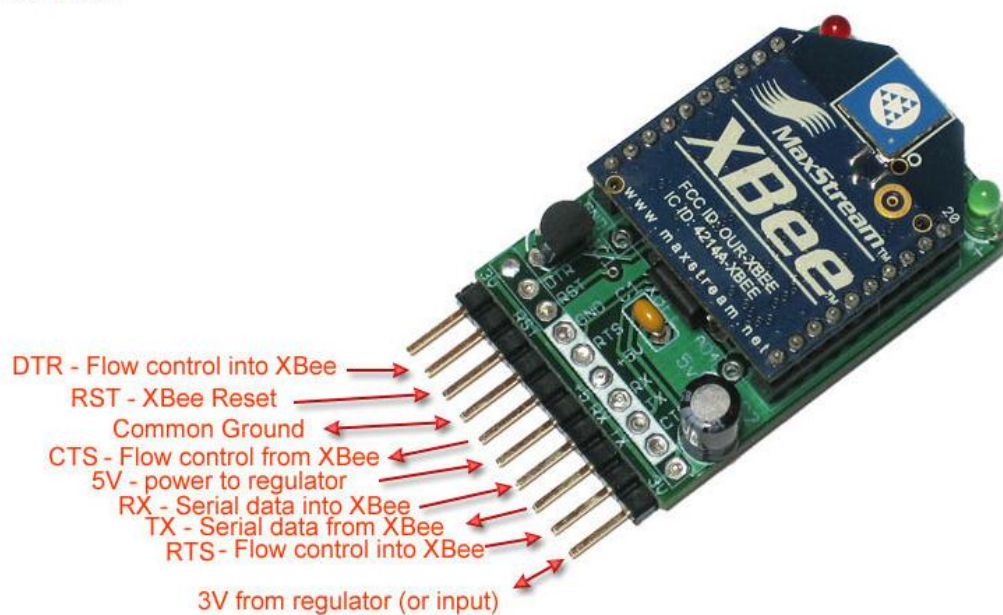


Εικόνα 45: Ηλεκτρολογικό σχέδιο του προσαρμογέα XBee  
(<http://www.ladyada.net/make/xbee/index.html>)



Εικόνα 46: PCB του προσαρμογέα Xbee (<http://www.ladyada.net/make/xbee/index.html>)

#### Reference Pinout



Εικόνα 47: Pin mapping του XBee adaptor (<http://www.ladyada.net/make/xbee/index.html>)

#### 5.4.4 Λειτουργικότητα του XBee

Το σύστημα χρησιμοποιεί το XBee για να επικοινωνεί με τον υπολογιστή. Στέλνει τις εξόδους, τις τιμές των αισθητήρων, την παρούσα θέση των σκιάστρων και την κατάσταση των φώτων στον υπολογιστή

που παρακολουθεί το σύστημα ή σε ένα arduino εξοπλισμένο με μια οθόνη LCD στο εσωτερικό του γραφείου. Ακόμη το σύστημα μπορεί μέσω του XBee να προγραμματιστεί από απόσταση, να στείλει και να λάβει αρχεία από και προς την κάρτα μνήμης. Επίσης οι εργαζόμενοι μπορούν να το χρησιμοποιήσουν για να θέσουν τιμές στην κατάσταση των φώτων και στη θέση των σκιάστρων ανάλογα με τις ανάγκες τους.

## 6 Κατασκευή και σχεδίαση shield για τον μικρό-ελεγκτή Arduino

Μέσω του προγράμματος Eagle της Cadsoft σχεδιάστηκε ένα PCB το οποίο ενσωματώνει το κύκλωμα του συστήματος που αναπτύχθηκε στο breadboard. Περιλαμβάνει τους αισθητήρες, το ρολόι πραγματικού χρόνου και θέσεις για τους προσαρμογείς της κάρτας μνήμης και του XBee. Παρακάτω δείχνονται το ηλεκτρολογικό σχέδιο του PCB και το ίδιο το PCB.

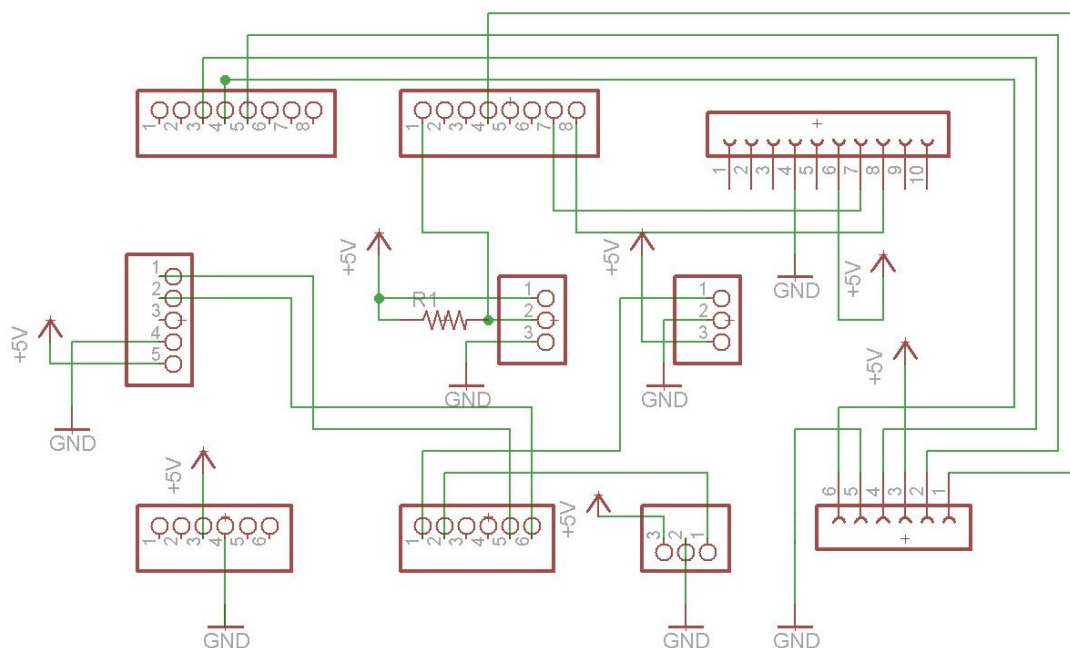
Τα βασικά στοιχεία του shield φαίνονται στον πίνακα:

Table 4

RTC
PIR sensor
Inside illuminance
Outdoor illuminance
Memory card Adapter
Xbee adapter

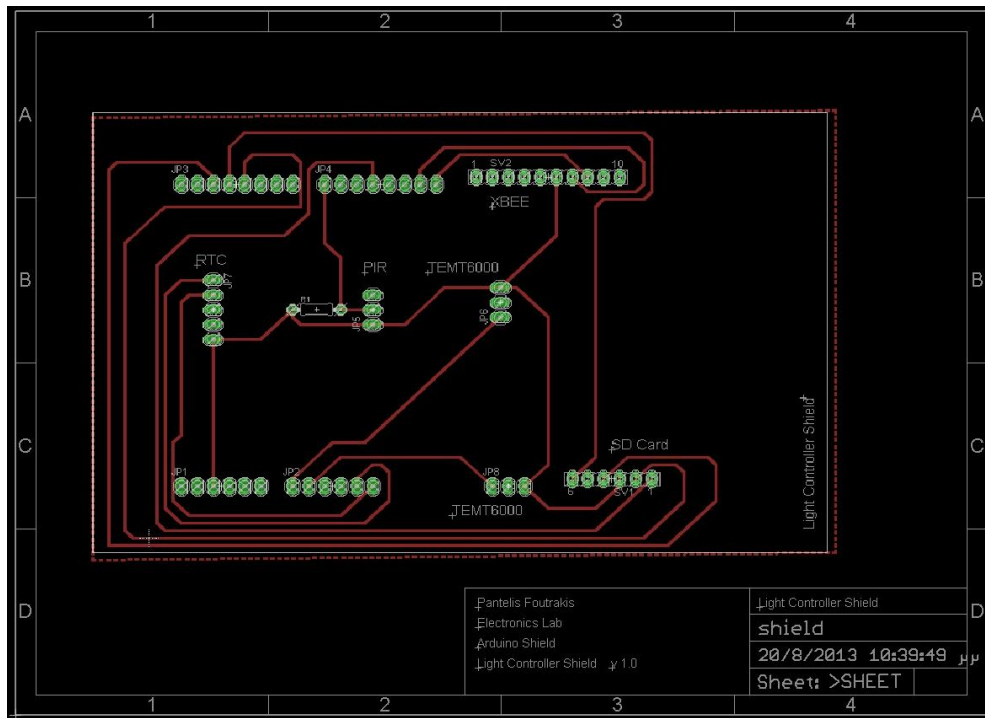
Η σύνδεση του συστήματος που δείχθηκε στο breadboard χρησιμοποιήθηκε για την ανάπτυξη του ηλεκτρολογικού σχεδίου του shield.





**Εικόνα 48: Ηλεκτρολογικό σχέδιο του shield**

Τα στοιχεία του PCB διατάχθηκαν με τέτοιο τρόπο, ώστε να καταλαμβάνει το λιγότερο δυνατό χώρο και η σύνδεση των εξαρτημάτων να γίνεται χωρίς τις γραμμές του χαλκού να μπερδεύονται μεταξύ τους. Επιπλέον οι γραμμές του χαλκού σχεδιάστηκαν να στρίβουν υπό γωνία  $45^\circ$  αντί  $90^\circ$  για καλύτερη κατανομή του χώρου και να είναι ανθεκτικότερες σε περίπτωση ατυχήματος κατά την κατασκευή του υλικού. Λήφθηκε προσοχή στη διάταξη για την αποφυγή της σκίασης του αισθητήρα φωτεινότητας που θα επηρέαζε την ομαλή λειτουργία του συστήματος. Ακόμη η σύνδεσή των προσαρμογέων στο shield που σχεδιάστηκε θα γίνεται υπό γωνία  $90^\circ$  ώστε να έχει το σύστημα λιγότερο ύψος.



Εικόνα 49: PCB του shield που σχεδιάστηκε

## 6.1 . Σύνδεση συστήματος και διάταξη στο χώρο

Το arduino με το shield των αισθητήρων τοποθετείται στην οροφή του δωματίου. Ο αισθητήρας παρουσίας, ο εσωτερικός και εξωτερικός αισθητήρας φωτεινότητας, ο προσαρμογέας της κάρτας μνήμης και του xbee είναι τοποθετημένοι στο shield που σχεδιάστηκε με σκοπό να εφαρμόζει πάνω στο arduino για άμεση χρήση από το σύστημα χωρίς επιπλέον εξωτερικά καλώδια. Ο εξωτερικός αισθητήρας φωτεινότητας τοποθετείται στο εξωτερικό μέρος του παραθύρου του γραφείου και συνδέεται με το υπόλοιπο σύστημα μέσω καλωδίου. Το σύστημα ελέγχου λειτουργεί είτε με μπαταρία των 9V είτε με μετασχηματιστή που συνδέεται σε ηλεκτρική πρίζα του κτιρίου.

## 7 Σχεδίαση και υλοποίηση ασαφούς λογικής

Εάν αναλύσουμε έναν ελεγκτή ασαφούς λογικής, παρατηρούμε ότι υπάρχουν μερικά βασικά στοιχεία που περιγράφουν τη λειτουργία του και τη σχέση εισόδου-εξόδου. Η δημιουργία ενός zero-order Takagi-Sugeno ασαφούς συστήματος απαιτεί τον ορισμό της εισόδου και εξόδου, τους αντίστοιχους τομείς, τις συναρτήσεις μέλη της εισόδου, τις σταθερές τιμές της εξόδου και ένα σύστημα κανόνων.

Με στόχο τον έλεγχο των σκιάστρων και του τεχνητού φωτισμού σε ένα γραφείο με τη χρήση του Takagi-Sugeno ασαφούς συστήματος δημιουργήθηκαν 2 μοντέλα με 2 εισόδους και 1 έξοδο το καθένα, ένα για τα σκιάστρα και ένα για τα τεχνητά φώτα.

Ως είσοδοι χρησιμοποιούνται οι αισθητήρες φωτισμού. Η τιμή του εσωτερικού αισθητήρα φωτεινότητας μείον την σταθερά 500 (η επιθυμητή φωτεινότητα σε ένα δωμάτιο είναι 500 lux, επομένως όταν το δωμάτιο έχει φωτεινότητα 500 lux αφαιρώντας την επιθυμητή τιμή το σφάλμα θα είναι μηδέν) ονομάζεται error και μετριέται σε lux και η τιμή του εξωτερικού αισθητήρα φωτεινότητας σε lux είναι η δεύτερη είσοδος και ονομάζεται outside. Χρησιμοποιούμε συνέχεις, τραπεζοειδούς μορφής συναρτήσεις μέλη για τις εισόδους του συστήματος και έχουν 4 παραμέτρους. Το εύρος για το error είναι [-500 2000] και για το outside είναι [0 1000]. Οι παράμετροι των συναρτήσεων μελών αποθηκεύονται ως πίνακες στην κάρτα μνήμης σε δύο ξεχωριστά αρχεία ένα για κάθε είσοδο.

Οι συναρτήσεις μέλη της εξόδου του zero-order Takagi-Sugeno συστήματος είναι σταθερές και αποθηκεύονται στην κάρτα μνήμης σε ένα αρχείο με τη μορφή διανύσματος στήλης.

Οι κανόνες της FL είναι ένα από τα βασικά στοιχεία του ασαφούς συστήματος. Ένας κανόνας μπορεί να αναπαρασταθεί με 4 στοιχεία. Μ' αυτόν τον τρόπο ένα σύστημα με  $n$  κανόνες, μπορεί να αναπαρασταθεί με ένα πίνακα διαστάσεων  $n \times 4$ . Οι κανόνες που εφαρμόζονται σ' αυτό το σύστημα έχουν την ακόλουθη μορφή:

- If error is A1 and outside is B1 then Y1
- If error is A2 and outside is B1 then Y2
- If error is A3 and outside is B1 then Y3

Για ένα σύστημα δύο εισόδων και μιας εξόδου, αυτή είναι μια γενική μορφή των κανόνων. Έτσι έχουμε, συναρτήσεις μέλη συνδεδεμένα με τις δύο εισόδους, έναν τελεστή που συνδέει τις δύο εισόδους (AND ή OR), μια τιμή της εξόδου, η οποία είναι συνάρτηση μέλος, αν και στο zero-order Takagi-Sugeno σύστημα είναι σταθερά. Με σκοπό να δημιουργήσουμε μια αριθμητική αναπαράσταση των κανόνων, σε κάθε συνάρτηση αντιστοιχούμε έναν αριθμό, για παράδειγμα την τιμή του δείκτη κάθε συνάρτησης. Μ' αυτό τον τρόπο έχουμε 3 στοιχεία σε ένα διάνυσμα γραμμής. Μένει μόνο να ορίσουμε μια αναπαράσταση για τα AND και OR. Αυτό μπορεί να λυθεί ορίζοντας την τιμή 1 για το AND και την τιμή 2 για το OR. Η συνάρτηση μέλος για την πρώτη είσοδο (error) είναι το πρώτο στοιχείο της γραμμής, το δεύτερο στοιχείο είναι η συνάρτηση μέλος της δεύτερης εισόδου (outdoor), το τρίτο είναι η συνάρτηση μέλος της εξόδου και το τέταρτο στοιχείο είναι η τιμή που ορίζει το AND ή το OR. Οπότε ο πρώτος κανόνας αναπαριστάται με το ακόλουθο διάνυσμα [1 1 1 1], ο δεύτερος είναι [2 1 2 1] και ο τρίτος είναι [3 1 3 1].

Επομένως για το σύστημα των κανόνων αποθηκεύουμε δύο αρχεία στην κάρτα μνήμης, ένα για τα σκίαστρα κι ένα για τα φώτα.

Οι κανόνες για τα φώτα και τα σκίαστρα φαίνονται στις ακόλουθες εικόνες:

1. If (error is ZERO) then (lights is stable) (1)
2. If (error is NE) and (outside is low) then (lights is up) (1)
3. If (error is NE) and (outside is normal) then (lights is up) (1)
4. If (error is NE) and (outside is high) then (lights is lup) (1)
5. If (error is SNE) and (outside is low) then (lights is lup) (1)
6. If (error is SNE) and (outside is normal) then (lights is stable) (1)
7. If (error is SNE) and (outside is high) then (lights is ldown) (1)
8. If (error is SPE) and (outside is low) then (lights is ldown) (1)
9. If (error is SPE) and (outside is normal) then (lights is ldown) (1)
10. If (error is SPE) and (outside is high) then (lights is ldown) (1)
11. If (error is PE) and (outside is low) then (lights is down) (1)
12. If (error is PE) and (outside is normal) then (lights is down) (1)
13. If (error is PE) and (outside is high) then (lights is down) (1)

**Εικόνα 50: Κανόνες του συστήματος για τα τεχνητά φώτα**

1. If (error is ZERO) then (shades is stable) (1)
2. If (error is NE) and (outside is low) then (shades is stable) (1)
3. If (error is NE) and (outside is normal) then (shades is up) (1)
4. If (error is NE) and (outside is high) then (shades is up) (1)
5. If (error is SNE) and (outside is low) then (shades is stable) (1)
6. If (error is SNE) and (outside is normal) then (shades is lup) (1)
7. If (error is SNE) and (outside is high) then (shades is lup) (1)
8. If (error is SPE) and (outside is low) then (shades is stable) (1)
9. If (error is SPE) and (outside is normal) then (shades is stable) (1)
10. If (error is SPE) and (outside is high) then (shades is ldown) (1)
11. If (error is PE) and (outside is low) then (shades is stable) (1)
12. If (error is PE) and (outside is normal) then (shades is down) (1)
13. If (error is PE) and (outside is high) then (shades is down) (1)

**Εικόνα 51: Κανόνες του συστήματος για τα σκίαστρα**

Οι συναρτήσεις μέλη για τις εισόδους και εξόδους φαίνονται στους πίνακες 1 και 2:

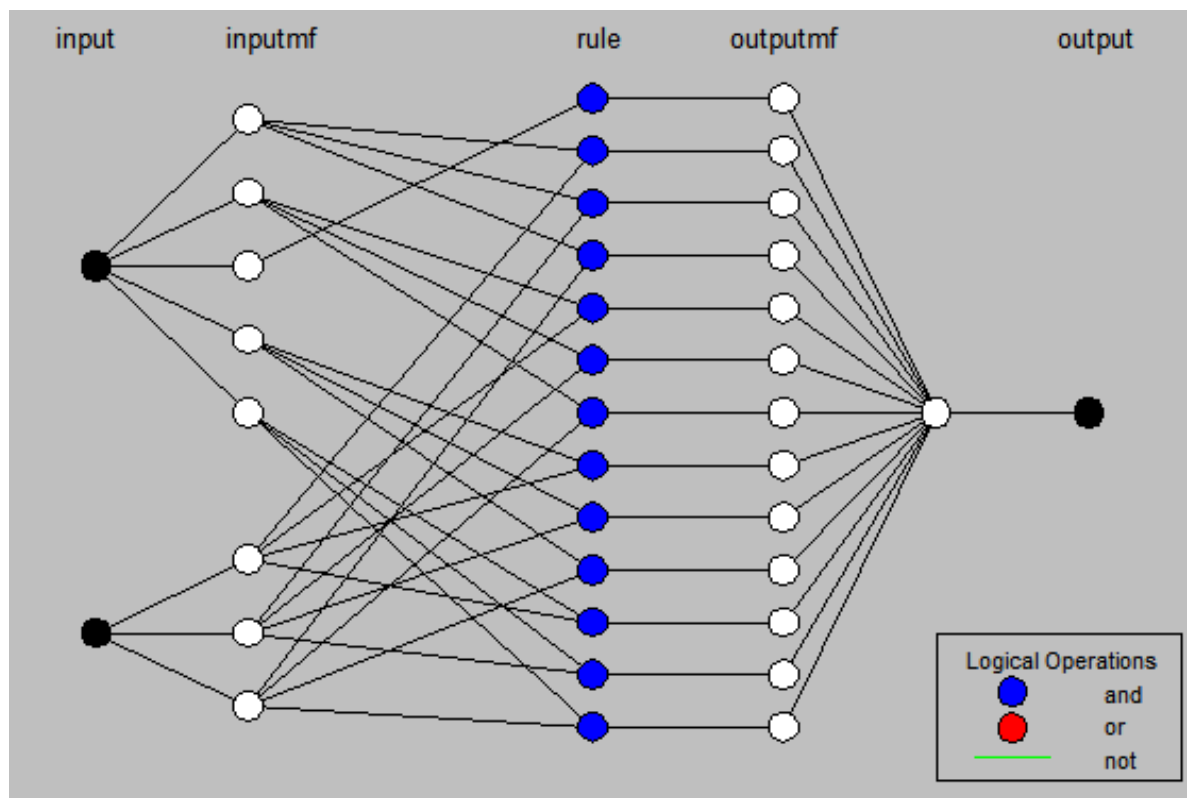
Table 5: Architecture of the fuzzy controller for lights

Type of fuzzy controller	'Sugeno'	
N. of inputs	2: error between current and desired light level, outside light level	
N. of outputs	1: change in the artificial lights state	
Fuzzification parameters error('trapmf')	'NE':	[-1000 -550 -400 -300]
	'ZERO':	[-200 -100 100 250]
	'PE':	[400 550 2200 2500]
	'SNE':	[-400 -300 -200 -100]
	'SPE':	[100 250 400 550]
Fuzzification parameters outside('trapmf')	'low':	[-450 0 250 450]
	'normal':	[250 450 550 950]
	'high':	[550 950 1050 1450]
De-fuzzification parameters (constant)	'down':	[-100]
	'ldown':	[-50]
	'stable':	[0]
	'lup':	[50]
	'up':	[100]

Table 6: Architecture of the fuzzy controller for shades

Type of fuzzy controller	'Sugeno'	
N. of inputs	2: error between current and desired light level, outside light level	
N. of outputs	1: change in the venetian blinds level	
Fuzzification parameters error('trapmf')	'NE':	[-1000 -550 -400 -300]
	'ZERO':	[-200 -100 100 250]
	'PE':	[400 550 2200 2500]
	'SNE':	[-400 -300 -200 -100]
	'SPE':	[100 250 400 550]
Fuzzification parameters outside('trapmf')	'low':	[-450 0 250 450]
	'normal':	[250 450 550 950]
	'high':	[550 950 1050 1450]
De-fuzzification parameters (constant)	'down':	[-100]
	'ldown':	[-50]
	'stable':	[0]
	'lup':	[50]
	'up':	[100]

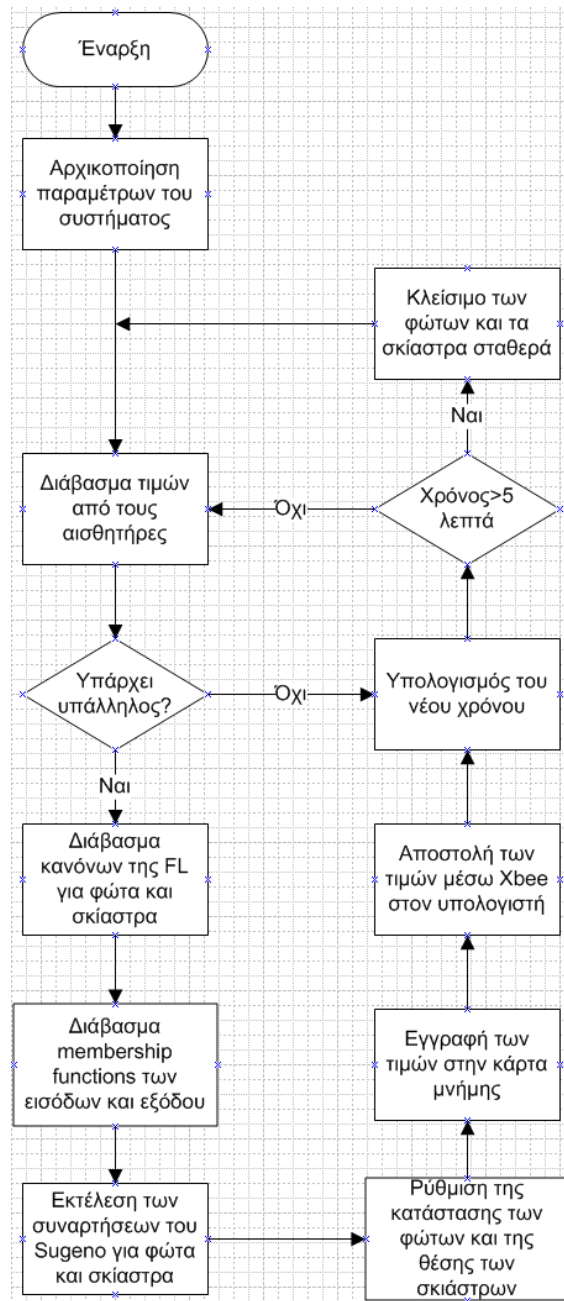
Η δομή του ασαφούς ελεγκτή δείχνεται στην ακόλουθη εικόνα:



Εικόνα 52: Δομή για σκίαστρα και τεχνητά φώτα

## 8 Προγραμματισμός του συστήματος

### 8.1 Διάγραμμα ροής



Εικόνα 53: Διάγραμμα ροής

Κατά την εκκίνηση του συστήματος γίνεται η αρχικοποίηση των παραμέτρων. Μόλις διαβαστούν οι τιμές των αισθητήρων, το σύστημα



ελέγχει αν υπάρχει κάποιος χρήστης στο δωμάτιο. Αν δεν υπάρχει χρήστης τότε ελέγχεται ο χρόνος που έχει περάσει από τη στιγμή που υπήρχε, αν είναι μεγαλύτερος από 5 λεπτά τότε κλείνουν τα φώτα η έξοδος του συστήματος για τα φώτα γίνεται μηδέν και τα σκίαστρα μένουν ως έχουν. Αν ο χρόνος είναι μικρότερος από 5 λεπτά διαβάζονται πάλι οι τιμές των αισθητήρων και ελέγχεται αν έχει έρθει χρήστης στο δωμάτιο, η κατάσταση των φώτων και των σκιάστρων μένει ως έχει. Αν στο δωμάτιο υπάρχει χρήστης τότε διαβάζονται οι κανόνες τις FL για το φωτισμό και τα σκίαστρα, οι συναρτήσεις μέλη των εισόδων και της εξόδου και εκτελείται η συνάρτηση του Sugeno. Με βάση την έξοδο του Sugeno ρυθμίζεται η νέα κατάσταση των φώτων και η νέα θέση των σκιάστρων. Το σύστημα γράφει τις τιμές σε ένα αρχείο στην κάρτα μνήμης και στέλνει τα δεδομένα στον υπολογιστή. Κατόπιν υπολογίζει το νέο χρόνο, αν είναι μικρότερος από 5 λεπτά επαναλαμβάνει τη διαδικασία μετά από αναμονή μερικών δευτερολέπτων, αν είναι μεγαλύτερος από 5 λεπτά κλείνει τα φώτα και διαβάζει τους αισθητήρες μέχρι να μπει κάποιος χρήστης στο χώρο του γραφείου.

## 8.2 Πηγαίος κώδικας

Ο πηγαίος κώδικας για τον προγραμματισμό του συστήματος είναι:

```
/*Elegchos fotismou se xorous grafeion me xrisi tou arduino,  
fuzzy logic controller, sd card kai xbee*/
```

```
#include <SD.h>  
#include <Wire.h>  
#define DS1307_ADDRESS 0x68  
  
int temt1pin = A0; // indoors light sensor  
int temt2pin = A1; // outdoors light sensor  
int motionsensorpin = 7; // pir presence sensor  
int temt1Value = 0;  
int temt2Value = 0;  
int temt1lux = 0; // temt1Value in lux  
int temt2lux = 0; // temt2Value in lux  
int motionsensorValue = 0;  
int error = 0;  
int fuzzy_lights = 0;  
int fuzzy_shades = 0;
```

```

int output_lights = 0;
int output_shades = 0;
String lights_state;
char shades_state = '0';
int mf_input1[5][4];
int mf_input2[3][4];
int mf_output[5][1];
int rules_lights[13][4];
int rules_shades[13][4];
int time = 0;
int time_hour = 0;
int time_minute = 0;
int current_hour = 0;
int current_minute = 0;

// fuzzy sugeno function

int fuzzy_sugeno_function(int x, int y, int (&Rule_matrix)[13][4], int (&mf_input1_matrix)[5][4], int
(&mf_input2_matrix)[3][4], int (&mf_output_matrix)[5][1]){

    int Sy=0;
    int Sm=0;
    int res=0;
    int j=0;
    int A1[5][1];
    int A2[3][1];
    int A[13][1];

    for(j=0; j<5; j++){

        A1[j][0]=f_function(x, mf_input1_matrix[j][0], mf_input1_matrix[j][1], mf_input1_matrix[j][2],
mf_input1_matrix[j][3]);
    }
    for(j=0; j<3; j++){

        A2[j][0]=f_function(y, mf_input2_matrix[j][0], mf_input2_matrix[j][1], mf_input2_matrix[j][2],
mf_input2_matrix[j][3]);
    }

    for(j=0; j<13; j++){

        if(Rule_matrix[j][1]==0){
            A[j][0]=A1[(Rule_matrix[j][0]-1)][0];
        }else if(Rule_matrix[j][3]==1){
            A[j][0]=min_val(A1[(Rule_matrix[j][0]-1)][0], A2[(Rule_matrix[j][1]-1)][0]);
        }else{
            A[j][0]=max_val(A1[(Rule_matrix[j][0]-1)][0], A2[(Rule_matrix[j][1]-1)][0]);
        }
    }

    for(j=0; j<13; j++){

        Sy=Sy+(A[j][0]*(mf_output_matrix[(Rule_matrix[j][2]-1)][0]));
        Sm=Sm+A[j][0];
    }
    if(Sm!=0){
        res=Sy/Sm;
    }else{
        res=0;
    }
}

```

```

    return res;
}

// f(x,a,b,c,d)=max(min((x-a)/(b-a),1,(d-x)/(d-c)),0)

int f_function(int x, int a, int b, int c, int d){

    int z=0;

    if(x<=a){
        z=0;
    }else if(x>=d){
        z=0;
    }else if((x>=b) && (x<=c)){
        z=100;
    }else if((x>a) && (x<b)){
        z=((x-a)*100)/(b-a);
    }else{
        z=((d-x)*100)/(d-c);
    }
    return z;

}

// read rules function

void read_rules_lights(int (&m)[13][4]){

    File myFile;
    int i=0;
    int j=0;
    int num=0;
    int neg=0;
    char c=0;

    myFile = SD.open("lights.txt");

    c=myFile.read();

    while (myFile.available()) {

        if(c==('\r')){
            c=myFile.read();
        }

        while(c==('\n')){
            c=myFile.read();
        }

        while(c==(' ')){
            c=myFile.read();
        }

        while((c!=' ') && (c!='\n') && (c!='\r')){
            if(c==('-')){
                c=myFile.read();
                neg=1;
            }

            num=(num*10)+(c-'0');

```

```

    c=myFile.read();
    if(c==(EOF)){
        break;
    }

    if((c==' ') || (c=='\n') || (c=='\r')){
        break;
    }
}

if(neg==1){
    m[i][j]=-num;
}else{
    m[i][j]=num;
}

j++;
num=0;
neg=0;
}
myFile.close();
}

void read_rules_shades(int (&m)[13][4]){

    File myFile;
    int i=0;
    int j=0;
    int num=0;
    int neg=0;
    char c=0;

    myFile = SD.open("shades.txt");

    c=myFile.read();

    while (myFile.available()) {

        if(c=='\r'){
            c=myFile.read();
        }

        while(c=='\n'){
            c=myFile.read();
        }

        while(c==' '){
            c=myFile.read();
        }

        while((c!=' ') && (c!='\n') && (c!='\r')){
            if(c=='-'){
                c=myFile.read();
                neg=1;
            }

            num=(num*10)+(c-'0');
            c=myFile.read();
            if(c==(EOF)){
                break;
            }
        }
    }
}

```

```

    }

    if((c==' ') || (c=='\n') || (c=='\r')){
        break;
    }
}

if(neg==1){
    m[i][j]=-num;
}else{
    m[i][j]=num;
}

j++;
num=0;
neg=0;
}
myFile.close();
}

// read mf functions

void read_mffunction1(int (&m)[5][4]){

    File myFile;
    int i=0;
    int j=0;
    int num=0;
    int neg=0;
    char c=0;

    myFile = SD.open("input1.txt");

    c=myFile.read();

    while (myFile.available()) {

        if(c=='\r'){
            c=myFile.read();
        }

        while(c=='\n'){
            c=myFile.read();
        }

        while(c==' '){
            c=myFile.read();
        }

        while((c!=' ') && (c!='\n') && (c!='\r')){
            if(c=='-'){
                c=myFile.read();
                neg=1;
            }

            num=(num*10)+(c-'0');
            c=myFile.read();
            if(c==(EOF)){
                break;
            }
        }
    }
}

```

```

        if((c==' ') || (c=='\n') || (c=='\r')){
            break;
        }
    }

    if(neg==1){
        m[i][j]=-num;
    }else{
        m[i][j]=num;
    }

    j++;
    num=0;
    neg=0;
}
myFile.close();
}

void read_mfffunction2(int (&m)[3][4]){

    File myFile;
    int i=0;
    int j=0;
    int num=0;
    int neg=0;
    char c=0;

    myFile = SD.open("input2.txt");

    c=myFile.read();

    while (myFile.available()) {

        if(c=='\r'){
            c=myFile.read();
        }

        while(c=='\n'){
            c=myFile.read();
        }

        while(c==' '){
            c=myFile.read();
        }

        while((c!=' ') && (c!='\n') && (c!='\r')){
            if(c=='-'){
                c=myFile.read();
                neg=1;
            }

            num=(num*10)+(c-'0');
            c=myFile.read();
            if(c==(EOF)){
                break;
            }

            if((c==' ') || (c=='\n') || (c=='\r')){
                break;
            }
        }
    }
}

```

```

    }
}

if(neg==1){
    m[i][j]=-num;
}else{
    m[i][j]=num;
}

j++;
num=0;
neg=0;
}
myFile.close();
}

void read_mffunction_out(int (&m)[5][1]){

    File myFile;
    int i=0;
    int j=0;
    int num=0;
    int neg=0;
    char c=0;

    myFile = SD.open("output.txt");

    c=myFile.read();

    while (myFile.available()) {

        if(c==('\r')){
            c=myFile.read();
        }

        while(c==('\n')){
            c=myFile.read();
        }

        while(c==(' ')){
            c=myFile.read();
        }

        while((c!=(' ')) && (c!=('\n')) && (c!=('\r'))){
            if(c==('-')){
                c=myFile.read();
                neg=1;
            }

            num=(num*10)+(c-'0');
            c=myFile.read();
            if(c==(EOF)){
                break;
            }

            if((c==(' ')) || (c==('\n')) || (c==('\r'))){
                break;
            }
        }
    }
}

```

```

    if(neg==1){
        m[i][j]=-num;
    }else{
        m[i][j]=num;
    }

    j++;
    num=0;
    neg=0;
}
myFile.close();
}

// Convert binary coded decimal to normal decimal numbers

byte bcdToDec(byte val) {

    return ( (val/16*10) + (val%16) );
}

// write the results to file data.txt

void write_file(int output_lights, int output_shades, String lights_state, char shades_state, int temt1lux,
int temt2lux){

    File myFile;

    myFile = SD.open("data.txt", FILE_WRITE);

    Wire.beginTransmission(DS1307_ADDRESS);

    byte zero = 0x00;
    Wire.write(zero);
    Wire.endTransmission();

    Wire.requestFrom(DS1307_ADDRESS, 7);

    int second = bcdToDec(Wire.read());
    int minute = bcdToDec(Wire.read());
    int hour = bcdToDec(Wire.read() & 0b111111); //24 hour time
    int weekDay = bcdToDec(Wire.read()); //0-6 -> sunday - Saturday
    int monthDay = bcdToDec(Wire.read());
    int month = bcdToDec(Wire.read());
    int year = bcdToDec(Wire.read());

    myFile.print(monthDay);
    myFile.print("/");
    myFile.print(month);
    myFile.print("/");
    myFile.print(year);
    myFile.print(" ");
    myFile.print(hour);
    myFile.print(":");
    myFile.print(minute);
    myFile.print("\t");
    myFile.print(temt1lux);
    myFile.print("\t");
    myFile.print(temt2lux);
    myFile.print("\t");
    myFile.print(output_lights);

```



```

myFile.print("\t");
myFile.print(output_shades);
myFile.print("\t");
myFile.print(lights_state);
myFile.print("\t");
myFile.println(shades_state);

myFile.close();
}

//print to serial

void printing_serial(int output_lights, int output_shades, String lights_state, char shades_state, int
temt1lux, int temt2lux){

    Wire.beginTransmission(DS1307_ADDRESS);

    byte zero = 0x00;
    Wire.write(zero);
    Wire.endTransmission();

    Wire.requestFrom(DS1307_ADDRESS, 7);

    int second = bcdToDec(Wire.read());
    int minute = bcdToDec(Wire.read());
    int hour = bcdToDec(Wire.read() & 0b111111); //24 hour time
    int weekDay = bcdToDec(Wire.read()); //0-6 -> sunday - Saturday
    int monthDay = bcdToDec(Wire.read());
    int month = bcdToDec(Wire.read());
    int year = bcdToDec(Wire.read());
    delay(5000);
    Serial.println("Arduino UNO");
    Serial.print(monthDay);
    Serial.print("/");
    Serial.print(month);
    Serial.print("/");
    Serial.print(year);
    delay(3000);
    Serial.print(" ");
    Serial.print(hour);
    Serial.print(":");
    Serial.print(minute);
    delay(3000);
    Serial.print("\t");
    Serial.print(temt1lux);
    Serial.print("\t");
    Serial.print(temt2lux);
    Serial.print("\t");
    delay(3000);
    Serial.print(output_lights);
    Serial.print("\t");
    Serial.print(output_shades);
    Serial.print("\t");
    Serial.print(lights_state);
    Serial.print("\t");
    Serial.println(shades_state);

}

//transform to lux function

```

```

int transform_to_lux_function(int z){

    return ((z*0.00488)/(9987*0.5*0.000001));

}

int min_val (int a, int b){

    if (a<=b)
        return a;
    else
        return b;

}

int max_val (int a, int b){

    if (a>=b)
        return a;
    else
        return b;

}

void setup() {

    pinMode(motionsensorpin, INPUT);
    pinMode(10, OUTPUT);
    SD.begin(4);
    Wire.begin();
    Serial.begin(9600);

}

void loop() {

    temt1Value = analogRead(temt1pin);
    temt2Value = analogRead(temt2pin);
    motionsensorValue = digitalRead(motionsensorpin);
    temt1lux = transform_to_lux_function(temt1Value);
    temt2lux = transform_to_lux_function(temt2Value);
    error = temt1lux-500;

    if (motionsensorValue == HIGH)
    {
        time=0;
    }

    if(time<=5){

        // read mf and rules files from sd card

        read_rules_lights(rules_lights);
        read_rules_shades(rules_shades);
        read_mffunction1(mf_input1);
        read_mffunction2(mf_input2);
        read_mffunction_out(mf_output);

        // fuzzy logic controller

```

```

    fuzzy_lights = fuzzy_sugeno_function(error, temt2lux, rules_lights, mf_input1, mf_input2,
mf_output); // calling fuzzy function for lights
    fuzzy_shades = fuzzy_sugeno_function(error, temt2lux, rules_shades, mf_input1, mf_input2,
mf_output); // calling fuzzy function for shades

    output_lights = output_lights+fuzzy_lights;
    output_shades = output_shades+fuzzy_shades;

    if(output_lights<0){
        output_lights=0;
    }else if(output_lights>100){
        output_lights=100;
    }

    if(output_shades<0){
        output_shades=0;
    }else if(output_shades>100){
        output_shades=100;
    }

    if(output_lights>0){
        lights_state = "ON";
    }else{
        lights_state = "OFF";
    }

    if(output_shades>=50){
        shades_state = '+';
    }else{
        shades_state = '-';
    }
}

write_file(output_lights, output_shades, lights_state, shades_state, temt1lux, temt2lux);

delay(5000);
Serial.print("Room 1");
Serial.print("\t");
delay(2000);
printing_serial(output_lights, output_shades, lights_state, shades_state, temt1lux, temt2lux);

Wire.beginTransmission(DS1307_ADDRESS);

byte zero = 0x00;
Wire.write(zero);
Wire.endTransmission();

Wire.requestFrom(DS1307_ADDRESS, 7);

int second = bcdToDec(Wire.read());
int minute = bcdToDec(Wire.read());
int hour = bcdToDec(Wire.read() & 0b111111); //24 hour time
int weekDay = bcdToDec(Wire.read()); //0-6 -> sunday - Saturday
int monthDay = bcdToDec(Wire.read());
int month = bcdToDec(Wire.read());
int year = bcdToDec(Wire.read());

current_hour=hour;
current_minute=minute;

```

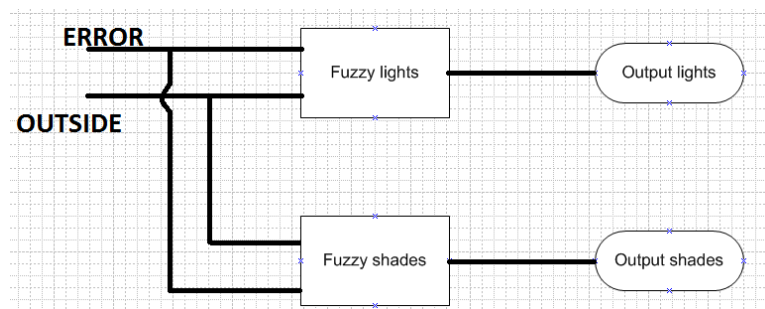
```
time_hour=current_hour-time_hour;
time_minute=current_minute-time_minute;
time=(time_hour*60)+time_minute;

if(time>5){
    output_lights=0;
    lights_state = "OFF";
}

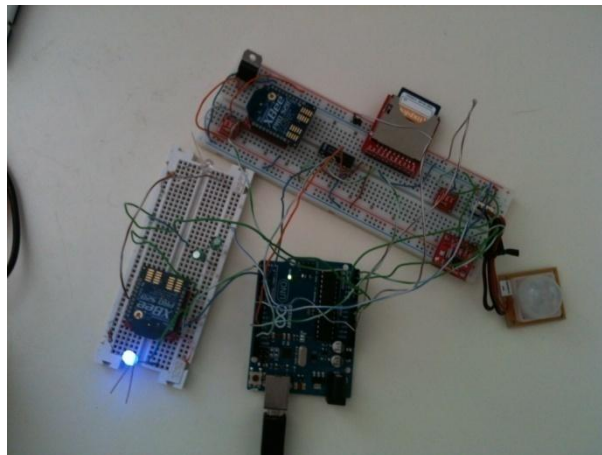
delay(1000);
}
```

## 9 Αποτελέσματα

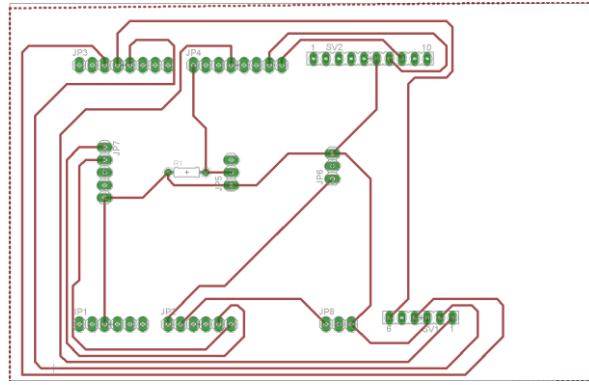
Σ' αυτό το κεφάλαιο παρουσιάζονται τα αποτελέσματα του ελεγκτή που αναπτύξαμε. Αρχικά στην εικόνα βλέπουμε τις εισόδους και τις εξόδους των δύο fuzzy ελεγκτών που υλοποιήθηκαν. Οι συνδεσμολογία του υλικού (των αισθητήρων και του μικρό-ελεγκτή arduino) φαίνονται στην εικόνα.



Εικόνα 54: Fuzzy logic μοντέλο του συστήματος

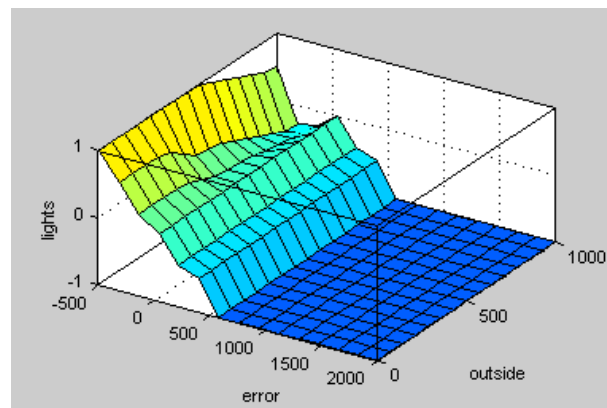


Εικόνα 55: Το σύστημα υλοποιημένο στο breadboard

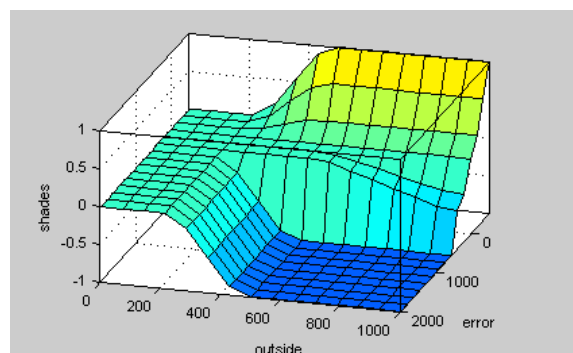


**Εικόνα 56: Η σχεδιασμένη πλακέτα για τους απαραίτητους αισθητήρες**

Η αλλαγή στο επίπεδο του φωτός λόγω της εξωτερικής φωτεινότητας και της διαφοράς της παρούσας από την επιθυμητή φωτεινότητα φαίνεται στην εικόνα , ενώ η διαφορά στο επίπεδο των σκιάστρων λόγω των ίδιων εισόδων φαίνονται στην εικόνα .



**Εικόνα 57: Οπτικοποίηση των κανόνων από τη Matlab για τον φωτισμό**



**Εικόνα 58: Οπτικοποίηση των κανόνων από τη Matlab για τα σκιάστρα**

## 10 . Συμπεράσματα

Ο έξυπνος ελεγκτής χρησιμοποιεί αποδοτικά την ασαφή λογική (FL) για να ελέγχει κατάλληλα τα τεχνητά φώτα και τα σκίαστρα σε ένα χώρο γραφείων απαιτώντας ελάχιστη ενέργεια για τη λειτουργία του. Επικοινωνεί με υπολογιστές και κατορθώνει μεγαλύτερη ακρίβεια και ευελιξία. Αποθηκεύει τους κανόνες της FL και τις συναρτήσεις μέλη στην κάρτα μνήμης προσφέροντας καλύτερη προσβασιμότητα στους χρήστες. Το σύστημα λειτουργεί με μεγάλη αξιοπιστία και μπορεί να ανταποκριθεί σε πραγματικές εφαρμογές. Ακόμη λόγω του μεγέθους του και της εύκολης εγκατάστασής του μπορεί να μετακινηθεί σε άλλους χώρους του κτιρίου γρήγορα και ανέξοδα. Με την παροχή των δεδομένων στον υπολογιστή τα αποτελέσματα μπορούν να χρησιμοποιηθούν για σύγκριση μέσα στη διάρκεια του έτους και για πρόβλεψη μελλοντικών τιμών μέσω εκπαίδευσης νευρωνικού δικτύου. Σημαντικό είναι να αναφερθεί ότι το σύστημα μπορεί να εγκατασταθεί σε οποιοδήποτε κτίριο χωρίς την αναγκαιότητα να ληφθούν υπόψη τα ιδιαίτερα χαρακτηριστικά του κτιρίου και χωρίς να αλλάξει η δομή του συστήματος.

## **11 . Μελλοντικές εφαρμογές**

Η χρήση του arduino και των αισθητήρων σε συστήματα ελέγχου για χώρους γραφείων είναι αρκετά ευέλικτη και δεν έχει ιδιαίτερα απαιτητική εγκατάσταση. Περισσότεροι αισθητήρες μπορούν να προστεθούν και με κατάλληλο προγραμματισμό του arduino μπορούν να εκτελεστούν περισσότερες λειτουργίες με λίγο κόπο. Μπορεί να προστεθούν στο παρόν σύστημα αισθητήρες CO<sub>2</sub>, υγρασίας, θερμοκρασίας και να ελέγχει πλήρως ένα δωμάτιο. Ένα σύστημα με τους παραπάνω αισθητήρες θα παρέχει ένα πλήρως ελεγχόμενο περιβάλλον εργασίας με την ανάγκη λίγης έως και καθόλου ανθρώπινης παρέμβασης και να εξοικονομήσει χώρο καθώς είναι πολύ μικρό και μπορεί να περάσει απαρατήρητο.



## Βιβλιογραφία

- [1]Berson, David M., Felice A. Dunn, and Motoharu Takao. 2002. “Phototransduction by retinal ganglion cells that set the circadian clock.” *Science* 295:1070–73.
- [2]Galasiu, Anca D., Morad R. Atif, and Robert A. MacDonald. 2004. “Impact of window blinds on daylight-linked dimming and automatic on/off lighting controls.” *Solar Energy* 76:523–44.
- [3]Galasiu, Anca D., and Jennifer A. Veitch. 2006. “Occupant preferences and satisfaction with the luminous environment and control systems in daylit offices: a literature review.” *Energy and Buildings* 38:728–42.
- [4]Guillemin, A., and S. Molteni. 2002. “An energy-efficient controller for shading devices self-adapting to the user wishes.” *Building and Environment* 37:1091–97.
- [5]Guillemin, A., and N. Morel. 2001. “An innovative lighting controller integrated in a self-adaptive building control system.” *Energy and Buildings* 33:477–87.
- [6]Guillemin, A., and N. Morel. 2002. “Experimental results of a self-adaptive integrated control system in buildings: a pilot study.” *Solar Energy* 72:397–403.
- [7]Inoue, T., T. Kawase, T. Ibamoto, S. Takakusa, and Y. Matsuo. 1988. “The development of an optimal control system for window shading devices based on investigations in office buildings.” *ASHRAE Transactions* 94:1034–49.

[8]Kolokotsa, D. 2003. “Comparison of the performance of fuzzy controllers for the management of the indoor environment.” *Building and Environment* 38:1439–50.

[9]Kolokotsa, D., D. Rovas, E. Kosmatopoulos, and K. Kalaitzakis. 2010. “A roadmap towards intelligent net zero- and positive-energy buildings.” *Solar Energy* 85:3067–84.

[10]Kolokotsa, D., G. Saridakis, A. Pouliezos, and G. S. Stavrakakis. 2006. “Design and installation of an advanced EIB<sup>TM</sup> fuzzy indoor comfort controller using Matlab.” *Energy and Buildings* 38:1084–92.

[11]Kolokotsa, D., D. Tsiavos, G. S. Stavrakakis, K. Kalaitzakis, and E. Antonidakis. 2001. “Advanced fuzzy logic controllers design and evaluation for buildings’ occupants thermal-visual comfort and indoor air quality satisfaction.” *Energy and Buildings* 33:531–43.

[12]Kurian, C., R. Aithal, J. Bhat, and V. George. 2008. “Robust control and optimisation of energy consumption in daylight--artificial light integrated schemes.” *Lighting Research and Technology* 40:7–24.

[13]Kurian, Cp, S. Kuriachan, J. Bhat, and Rs Aithal. 2005. “An adaptive neuro-fuzzy model for the prediction and control of light in integrated lighting schemes.” *Lighting Research and Technology* 37:343–52.

[14]Leslie, R. P., R. Raghavan, O. Howlett, and C. Eaton. 2005. “The potential of simplified concepts for daylight harvesting.” *Lighting Research and Technology* 37:21–40.

[15]Papantoniou, S. (2012) Neuro-fuzzy model based predictive algorithm for environmental management of buildings.

[16] Trobec Lah, Mateja, Borut Zupančič, Jože Peternelj, and Aleš Krainer. 2006. "Daylight illuminance control with fuzzy logic." *Solar Energy* 80:307–21.

[17] <http://playground.arduino.cc/>

[18] <http://www.arduino.cc>

[19] <http://www.digi.com>

[20] <http://www.edaboard.com>

[21] <http://www.instructables.com>

[22] <http://www.ladyada.net/make/xbee/index.html/>