



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Ανάπτυξη πλατφόρμας επικοινωνίας για παρακολούθηση και έλεγχο ενεργειακής αποδοτικότητας ευφυών κατοικιών

Γεώργιος Τσανικλίδης

Επιβλέπων καθηγητής:
Δρ. Καλαϊτζάκης Κωνσταντίνος

Χανιά 2016

Πρόλογος

Οφείλω κατ' αρχήν να εκφράσω τις θερμές μου ευχαριστίες στον επιβλέποντα καθηγητή μου, κ. Καλαϊτζάκη Κωνσταντίνο, για την ανιδιοτελή και συνεχή καθοδήγηση που μου προσέφερε καθ' όλη την διάρκεια της φοίτησής μου και σε όλα τα στάδια της εκπόνησης της παρούσας διπλωματικής εργασίας.

Επίσης θα ήθελα να ευχαριστήσω θερμά τον αναπληρωτή Καθηγητή κ. Κουτρούλη Ευτύχιο, την επίκουρη καθηγήτρια κα Κολοκοτσά Διονυσία και τον υποψήφιο Διδάκτωρα κ. Γομπάκη Κωνσταντίνο για την προσφορά και την βοήθειά τους στην έρευνα των τεχνολογιών που χρησιμοποιήθηκαν.

Ιδιαίτερες ευχαριστίες οφείλω στην μεταπτυχιακή φοιτήτρια Τσιφιντουλάκη Έλλη και στο Πολυτεχνείο Κρήτης.

Τέλος, ένα μεγάλο ευχαριστώ στην οικογένεια μου, τους συγγενείς και τους φίλους μου για την στήριξη τους καθ' όλη τη διάρκεια των σπουδών μου.

Περίληψη

Η παρούσα εργασία πραγματεύεται την υλοποίηση μίας πλατφόρμας παρακολούθησης ενεργειακών δεδομένων μίας ευφυούς κατοικίας από ηλεκτρονικό υπολογιστή ή από έξυπνα κινητά τηλέφωνα. Περιλαμβάνει την κυκλωματική υλοποίηση και προγραμματισμό των σταθμών βάσης αισθητήρων και ενεργοποιητών, καθώς και τον προγραμματισμό της διεπαφής χρήστη της εφαρμογής, μαζί με τον μηχανισμό άντλησης των δεδομένων από αυτήν.

Τα δεδομένα τα οποία συλλέγονται από τους αισθητήρες οι οποίοι είναι εγκατεστημένοι στα δωμάτια της κατοικίας, και εμφανίζονται στην εφαρμογή, είναι η θερμοκρασία, η υγρασία, η φωτεινότητα, η ταχύτητα ανέμου, η περιεκτικότητα του αέρα σε διοξείδιο του άνθρακα, και η ανίχνευση κίνησης. Τα δεδομένα αυτά, στην συνέχεια, δύναται να χρησιμοποιηθούν για αυτοματισμό της λειτουργίας της οικίας, σε συνδυασμό με τους ενεργοποιητές.

Οι ενεργοποιητές οι οποίοι εγκαταστάθηκαν στην οικία περιλαμβάνουν φώτα, κινητήρες πορτών και παραθύρων και ηχεία ειδοποίησης. Τα φώτα και οι κινητήρες μπορούν, επίσης, να ελεγχθούν μέσω της υλοποιηθείσας πλατφόρμας, έπειτα από επιλογή του χρήστη στην εφαρμογή, ενέργεια για την οποία επινοήθηκε και υλοποιήθηκε προγραμματιστικά ένα βασικό πρωτόκολλο.

Η υλοποιηθείσα πλατφόρμα δύναται να επεκταθεί και να παραμετροποιηθεί, ώστε να φιλοξενήσει οποιουδήποτε είδους αισθητήρα, και να εμφανίσει τα δεδομένα από αυτόν στον χρήστη. Αυτό το γεγονός, σε συνδυασμό με την ταχύτητα και την απόδοσή της, την καθιστά κατάλληλη όχι μόνο στο πλαίσιο μίας έξυπνης οικίας, αλλά και σε πλήθος άλλων εφαρμογών του διαδικτύου των πραγμάτων.

Περιεχόμενα

Πρόλογος	2
Περίληψη	3
Περιεχόμενα Σχημάτων	7
Περιεχόμενα Πινάκων	11
Πίνακας συντημήσεων	12
1. Εισαγωγή	14
1.1 Περιγραφή οφελών ευφυούς κατοικίας	14
1.2 Περιγραφή προβλήματος	15
1.3 Στόχοι της Διπλωματικής	16
1.4 Διάρθρωση Διπλωματικής	16
2. Ανάλυση εννοιών, θεωρητικό υπόβαθρο, προδιαγραφές	18
2.1 Η πλακέτα ανάπτυξης και οι προδιαγραφές της	18
2.2 Arduino Mega 2560	18
2.2.1 Arduino Ethernet Shield	20
2.2.2 Το Arduino IDE - Arduino Software	22
2.3 Αισθητήρες και οι προδιαγραφές τους	24
2.4 Αισθητήρας θερμοκρασίας LM35D	25
2.4.1 Συνδεσμολογία LM35D με Arduino	26
2.4.2 Κώδικας Arduino για αισθητήρα LM35D	27
2.4.3 Αισθητήρας υγρασίας HIH-4030	28
2.4.4 Συνδεσμολογία HIH-4030 με Arduino	29
2.4.5 Κώδικας Arduino για αισθητήρα HIH-4030	30
2.5 Αισθητήρας φωτός περιβάλλοντος TEMT6000	31
2.5.1 Συνδεσμολογία TEMT6000 με Arduino	32
2.5.2 Κώδικας Arduino για αισθητήρα TEMT6000	33
2.6 Αισθητήρας ανέμου Wind Sensor Rev. C	34
2.6.1 Συνδεσμολογία Wind Sensor Rev. C με Arduino	36
2.6.2 Κώδικας Arduino για αισθητήρα Wind Sensor Rev. C	38
2.7 Αισθητήρας διοξειδίου του άνθρακα K-30	39
2.7.1 Συνδεσμολογία K-30 με Arduino	40
2.7.2 Κώδικας Arduino για αισθητήρα διοξειδίου του άνθρακα K-30	41
2.8 Αισθητήρας ανίχνευσης κίνησης PIR	42
2.8.1 Συνδεσμολογία PIR Sensor με Arduino	43
2.8.2 Κώδικας Arduino για χρήση αισθητήρα PIR	45
2.9 Κύκλωμα οδήγησης κινητήρων	46

2.9.1	Λειτουργία γέφυρας H (H-Bridge).....	46
2.9.2	Η λειτουργία του κυκλώματος οδήγησης (Motor Driver)	47
2.9.3	Η λειτουργία του ολοκληρωμένου κυκλώματος L293	47
2.9.4	Συνδεσμολογία L293D με Arduino για έλεγχο DC κινητήρων	49
2.9.5	Κώδικας Arduino για οδήγηση δύο κινητήρων.....	50
2.10	Τεχνολογίες δικτύου.....	51
2.11	Τεχνολογίες υλοποίηση διεπαφής χρήστη (User Interface)	52
2.11.1	Δομή σελίδας εφαρμογής με χρήση του jQuery Mobile	53
2.11.2	Χρήση λίστας στοιχείων (Listview) σε jQuery Mobile	54
2.11.3	Χρήση επιλογέων (Buttons) σε jQuery Mobile	55
3.	Πρακτική υλοποίηση πλατφόρμας	57
3.1	Υλοποίηση σταθμού βάσης αισθητήρων	57
3.2	Υλοποίηση σταθμού βάσης ελέγχου κινητήρων και φώτων	67
3.3	Υλοποίηση διεπαφής χρήστη	71
3.3.1	Υλοποίηση κεντρικού μενού διεπαφής χρήστη πλατφόρμας έξυπνου σπιτιού	71
3.3.2	Υλοποίηση διεπαφής χρήστη άντλησης δεδομένων αισθητήρων πλατφόρμας έξυπνου σπιτιού	73
3.3.3	Υλοποίηση διεπαφής χρήστη ελέγχου κινητήρων και φωτισμού έξυπνου σπιτιού	77
3.3.4	Υλοποίηση λοιπών τμημάτων διεπαφής χρήστη εφαρμογής πλατφόρμας έξυπνου σπιτιού	80
3.3.5	Υλοποίηση Android εφαρμογής με χρήση της προτεινόμενης διεπαφής χρήστη.....	81
4.	Συμπεράσματα και Μελλοντική Έρευνα.....	87
4.1	Συμπεράσματα	87
4.2	Μελλοντική Έρευνα.....	88
5.	Βιβλιογραφία	90
6.	Παράρτημα.....	91
6.1	Στιγμιότυπα οθόνης και κώδικας HTML5 εφαρμογής	91
6.1.1	Στιγμιότυπα κεντρικής οθόνης εφαρμογής	91
6.1.2	Κώδικας κεντρικής οθόνης εφαρμογής	92
6.1.3	Στιγμιότυπα οθόνης Ισογείου εφαρμογής	93
6.1.4	Κώδικας οθόνης Ισογείου εφαρμογής	94
6.1.5	Στιγμιότυπα οθόνης Πρώτου Ορόφου εφαρμογής.....	95
6.1.6	Κώδικας οθόνης Πρώτου Ορόφου εφαρμογής	96
6.1.7	Στιγμιότυπα οθόνης μετρήσεων Κήπου εφαρμογής	97
6.1.8	Κώδικας οθόνης μετρήσεων Κήπου εφαρμογής.....	98

6.1.9	Σπιγμιότυπα οθόνης ενεργοποιητών Κήπου εφαρμογής	100
6.1.10	Κώδικας οθόνης ενεργοποιητών Κήπου εφαρμογής	101
6.1.11	Σπιγμιότυπα οθόνης Ευχαριστιών εφαρμογής	103
6.1.12	Κώδικας οθόνης Ευχαριστιών εφαρμογής	104
6.1.13	Σπιγμιότυπα οθόνης Σχετικά (About) εφαρμογής	105
6.1.14	Κώδικας οθόνης Σχετικά (About) εφαρμογής	106
6.1.15	Σπιγμιότυπα οθόνης μετρήσεων Τραπεζαρίας εφαρμογής	107
6.1.16	Κώδικας οθόνης μετρήσεων Τραπεζαρίας εφαρμογής	108
6.1.17	Σπιγμιότυπα οθόνης ενεργοποιητών Τραπεζαρίας εφαρμογής	110
6.1.18	Κώδικας οθόνης ενεργοποιητών Τραπεζαρίας εφαρμογής	111
6.1.19	Σπιγμιότυπα οθόνης μετρήσεων Κουζίνας εφαρμογής	113
6.1.20	Κώδικας οθόνης μετρήσεων Κουζίνας εφαρμογής	114
6.1.21	Σπιγμιότυπα οθόνης ενεργοποιητών Κουζίνας εφαρμογής	116
6.1.22	Κώδικας οθόνης ενεργοποιητών Κουζίνας εφαρμογής	117
6.1.23	Σπιγμιότυπα οθόνης μετρήσεων Σαλονιού εφαρμογής	119
6.1.24	Κώδικας οθόνης μετρήσεων Σαλονιού εφαρμογής	120
6.1.25	Σπιγμιότυπα οθόνης ενεργοποιητών Σαλονιού εφαρμογής	122
6.1.26	Κώδικας οθόνης ενεργοποιητών Σαλονιού εφαρμογής	123
6.1.27	Σπιγμιότυπα οθόνης μετρήσεων Γραφείου εφαρμογής	125
6.1.28	Κώδικας οθόνης μετρήσεων Γραφείου εφαρμογής	126
6.1.29	Σπιγμιότυπα οθόνης ενεργοποιητών Γραφείου εφαρμογής	128
6.1.30	Κώδικας οθόνης ενεργοποιητών Γραφείου εφαρμογής	129
6.1.31	Σπιγμιότυπα οθόνης μετρήσεων Κρεβατοκάμαρας εφαρμογής	131
6.1.32	Κώδικας οθόνης μετρήσεων Κρεβατοκάμαρας εφαρμογής	132
6.1.33	Σπιγμιότυπα οθόνης ενεργοποιητών Κρεβατοκάμαρας εφαρμογής	134
6.1.34	Κώδικας οθόνης ενεργοποιητών Κρεβατοκάμαρας εφαρμογής	135
6.2	Κώδικας Android εφαρμογής	137
6.2.1	Κώδικας Java του MainActivity Android εφαρμογής	137
6.2.2	Κώδικας αρχείου strings.xml Android εφαρμογής	139
6.2.3	Κώδικας αρχείου AndroidManifest.xml Android εφαρμογής	139
6.2.4	Κώδικας Layout (main.xml) Android εφαρμογής	140
6.3	Κώδικας Arduino	141
6.3.1	Κώδικας Arduino σταθμού βάσης αισθητήρων	141
6.3.2	Κώδικας Arduino σταθμού βάσης ενεργοποιητών φώτων/μοτέρ	149
6.3.3	Κώδικας Arduino δοκιμής λειτουργίας κινητήρων και φώτων	156
6.4	Ορισμοί	160

Περιεχόμενα Σχημάτων

Σχήμα 2-1. Το Arduino Mega 2560	19
Σχήμα 2-2. Σχηματική αναπαράσταση Arduino Mega 2560	19
Σχήμα 2-3. Η συμβατή με το Arduino Mega 2560, Ethernet Shield	20
Σχήμα 2-4. Σχηματική αναπαράσταση Ethernet Shield	21
Σχήμα 2-5. Το IDE του Arduino	22
Σχήμα 2-6. Ενδεικτικό Hello World (Blink) στο Arduino IDE	23
Σχήμα 2-7. Ο αισθητήρας θερμοκρασίας LM35D	25
Σχήμα 2-8. Σχηματικό διάγραμμα αισθητήρα LM35	26
Σχήμα 2-9. Συνδεσμολογία LM35D με Arduino	26
Σχήμα 2-10. Σχηματικό συνδεσμολογίας LM35D με Arduino	27
Σχήμα 2-11. Ο αισθητήρας υγρασίας HIH-4030	28
Σχήμα 2-12. Σχηματικό διάγραμμα αισθητήρα υγρασίας HIH-4030.....	28
Σχήμα 2-13. Συνδεσμολογία HIH-4030 με Arduino	29
Σχήμα 2-14. Σχηματικό συνδεσμολογίας HIH-4030 με Arduino	29
Σχήμα 2-15. Ο αισθητήρας φωτός TEMT6000	31
Σχήμα 2-16. Σχηματικό διάγραμμα αισθητήρα TEMT6000	31
Σχήμα 2-17. Συνδεσμολογία TEMT6000 με Arduino	32
Σχήμα 2-18. Σχηματικό συνδεσμολογίας TEMT6000 με Arduino	32
Σχήμα 2-19. Ο αισθητήρας ανέμου Wind Sensor Rev. C	34
Σχήμα 2-20. Σχηματικό διάγραμμα αισθητήρα Wind Sensor Rev. C	35
Σχήμα 2-21. Συνδεσμολογία Wind Sensor Rev. C με Arduino	36
Σχήμα 2-22. Σχηματικό συνδεσμολογίας Wind Sensor Rev. C με Arduino	36
Σχήμα 2-23. Ο αισθητήρας διοξειδίου του άνθρακα K-30	39
Σχήμα 2-24. Συνδεσμολογία K-30 με Arduino	40
Σχήμα 2-25. Σχηματικό συνδεσμολογίας K-30 με Arduino.....	40
Σχήμα 2-26. Ο αισθητήρας ανίχνευσης κίνησης PIR.....	42
Σχήμα 2-27. Σχηματικό διάγραμμα αισθητήρα PIR	43
Σχήμα 2-28. Συνδεσμολογία PIR με Arduino	44
Σχήμα 2-29. Σχηματικό συνδεσμολογίας PIR με Arduino	44
Σχήμα 2-30. Κυκλωματικό διάγραμμα μίας H-bridge (επισημαίνεται με κόκκινο χρώμα)	46
Σχήμα 2-31. Οι δύο βασικές καταστάσεις μίας H-bridge.....	47

Σχήμα 2-32. Το ολοκληρωμένο κύκλωμα L293D	48
Σχήμα 2-33. Σχηματικό διάγραμμα του ολοκληρωμένου κυκλώματος L293D	48
Σχήμα 2-34. Συνδεσμολογία L293 με Arduino για έλεγχο δύο DC κινητήρων	49
Σχήμα 2-35. Σχηματικό συνδεσμολογίας L293 με Arduino για έλεγχο δύο DC κινητήρων	49
Σχήμα 2-36. Port forwarding σε Router Philips CGA5720S/TE	51
Σχήμα 2-37. Γραφικά στοιχεία (Widgets) του jQuery Mobile	52
Σχήμα 2-38. Λίστα στοιχείων (Listview) σε jQuery Mobile	54
Σχήμα 2-39. Επιλογείς (Buttons) σε jQuery Mobile	56
Σχήμα 3-1. Παράσταση συνδεσμολογίας συνόλου αισθητήρων και buzzer σε σταθμό βάσης	57
Σχήμα 3-2. Αναπαράσταση δημιουργίας αρχείου XML και αποστολή του σε έξυπνες συσκευές μέσω του Router	59
Σχήμα 3-3. Χρήση εργαλείου ring για επαλήθευση επικοινωνίας με WebServer που τρέχει στο Arduino	61
Σχήμα 3-4. Επιβεβαίωση δημιουργίας αρχείου XML μέσω Browser	62
Σχήμα 3-5. Το πλήρες υλοποιηθέν API σε XML με τις μετρήσεις όλων των αισθητήρων	62
Σχήμα 3-6. Παράσταση συνδεσμολογίας συνόλου L293, κινητήρων και LEDs σε σταθμό βάσης ελέγχου κινητήρων και φώτων	67
Σχήμα 3-7. Εκτέλεση GET request για τον έλεγχο κινητήρων και φώτων	69
Σχήμα 3-8. Στιγμιότυπα οθόνης υλοποιηθείσας εφαρμογής	71
Σχήμα 3-9. Στιγμιότυπο οθόνης δεδομένων αισθητήρων υλοποιηθείσας εφαρμογής	74
Σχήμα 3-10. Στιγμιότυπο οθόνης ελέγχου ενεργοποιητών υλοποιηθείσας εφαρμογής	77
Σχήμα 3-11. Επιβεβαίωση εκτέλεσης get request στον Server του Arduino κατά την επιλογή "Άνοιξε φως" της προτεινόμενης εφαρμογής	80
Σχήμα 3-12. Μπάρα προόδου άντλησης δεδομένων από Arduino WebServer στο Android	83
Σχήμα 3-13. Στατιστικά στοιχεία κατανομής εκδόσεων του Android, Σεπτέμβριος 2016	86
Σχήμα 6-1. Διεπαφή κεντρικού μενού υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου	91
Σχήμα 6-2. Διεπαφή κεντρικού μενού υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή	91
Σχήμα 6-3. Διεπαφή Ισογείου υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου	93
Σχήμα 6-4. Διεπαφή Ισογείου υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή	93
Σχήμα 6-5. Διεπαφή Πρώτου ορόφου υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου	95
Σχήμα 6-6. Διεπαφή Πρώτου ορόφου υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή	95
Σχήμα 6-7. Διεπαφή μετρήσεων Κήπου υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου	97
Σχήμα 6-8. Διεπαφή μετρήσεων Κήπου υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή	97

Σχήμα 6-9. Διεπαφή ενεργοποιητών Κήπου υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου	100
Σχήμα 6-10. Διεπαφή ενεργοποιητών Κήπου υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή	100
Σχήμα 6-11. Διεπαφή Ευχαριστιών υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου	103
Σχήμα 6-12. Διεπαφή Ευχαριστιών υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή	103
Σχήμα 6-13. Διεπαφή Σχετικά (About) υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου	105
Σχήμα 6-14. Διεπαφή Σχετικά (About) υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή	105
Σχήμα 6-15. Διεπαφή μετρήσεων τραπεζαρίας υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου	107
Σχήμα 6-16. Διεπαφή μετρήσεων τραπεζαρίας υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή	107
Σχήμα 6-17. Διεπαφή ενεργοποιητών τραπεζαρίας υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου	110
Σχήμα 6-18. Διεπαφή ενεργοποιητών τραπεζαρίας υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή	110
Σχήμα 6-19. Διεπαφή μετρήσεων Κουζίνας υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου	113
Σχήμα 6-20. Διεπαφή μετρήσεων Κουζίνας υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή	113
Σχήμα 6-21. Διεπαφή ενεργοποιητών Κουζίνας υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου	116
Σχήμα 6-22. Διεπαφή ενεργοποιητών Κουζίνας υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή	116
Σχήμα 6-23. Διεπαφή μετρήσεων Σαλονιού υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου	119
Σχήμα 6-24. Διεπαφή μετρήσεων Σαλονιού υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή	119
Σχήμα 6-25. Διεπαφή ενεργοποιητών Σαλονιού υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου	122
Σχήμα 6-26. Διεπαφή ενεργοποιητών Σαλονιού υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή	122
Σχήμα 6-27. Διεπαφή μετρήσεων Γραφείου υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου	125
Σχήμα 6-28. Διεπαφή μετρήσεων Γραφείου υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή	125

Σχήμα 6-29. Διεπαφή ενεργοποιητών Γραφείου υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου	128
Σχήμα 6-30. Διεπαφή ενεργοποιητών Γραφείου υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή	128
Σχήμα 6-31. Διεπαφή μετρήσεων Κρεβατοκάμαρας υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου	131
Σχήμα 6-32. Διεπαφή μετρήσεων Κρεβατοκάμαρας υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή	131
Σχήμα 6-33. Διεπαφή ενεργοποιητών Κρεβατοκάμαρας εφαρμογής σε οθόνη κινητού τηλεφώνου .	134
Σχήμα 6-34. Διεπαφή ενεργοποιητών Κρεβατοκάμαρας εφαρμογής σε οθόνη tablet/υπολογιστή ...	134

Περιεχόμενα Πινάκων

Πίνακας 2-1. Επεξήγηση των Pins του Wind Sensor Rev C.....	37
Πίνακας 3-1. Αντιστοιχία Pins του Arduino του πρώτου σταθμού βάσης, με αισθητήρες, τύπους εξόδων, μεταβλητές στο XML, μετρούμενα μεγέθη και δωμάτια.....	66
Πίνακας 3-2. Αντιστοιχία Pins του Arduino του δεύτερου σταθμού βάσης, με συνδεδεμένο κύκλωμα, κανάλι L293, μεταβλητές στο πρωτόκολλο, και δωμάτια.....	70

Πίνακας συντμήσεων

ΑΠΕ: Ανανεώσιμες Πηγές Ενέργειας

ABC: Automatic Background Calibration

ADB: Android Debug Bridge

ADC: Analog to Digital Converter

AJAX: Asynchronous JavaScript And XML

API: Application Programming Interface

DC: Direct Current

DHCP: Dynamic Host Configuration Protocol

DTL: Diode–Transistor Logic

GND: GrouND

HTML: HyperText Markup Language

IC: Integrated Circuit

IP: Internet Protocol

IDE: Integrated Development Environment

LED: Light Emitting Diode

MIT: Massachusetts Institute of Technology

NDIR: Non-Dispersive InfraRed

OSH: Open Source Hardware

PCB: Printed Circuit Board

PIR: Passive InfraRed

PWM: Pulse Width Modulation

RH: Relative Humidity

SPI: Serial Peripheral Interface

SBPC: Single Board Personal Computer

SDK: Software Development Kit

SMD: Surface Mount Device

SPI: Serial Peripheral Interface

SS: Slave Select

TCP: Transmission Control Protocol

TTL: Transistor–Transistor Logic

UART: Universal Asynchronous Receiver/Transmitter

UDP: User Datagram Protocol

U.I./U.X: User Interface/User eXperience

URI: Uniform Resource Identifier

URL: Uniform Resource Locator

UTF: Unicode Transformation Format

1. Εισαγωγή

1.1 Περιγραφή οφελών ευφυούς κατοικίας

Τα τελευταία χρόνια, όλο και περισσότεροι άνθρωποι, θέλοντας να εκμεταλλευτούν τη ραγδαία εξέλιξη των τεχνολογικών επιτευγμάτων και καινοτομιών, προς ικανοποίηση των αναγκών τους και βελτίωση της ποιότητας της ζωής τους, επιλέγουν να κάνουν χρήση όλο και περισσότερων έξυπνων αντικειμένων. Ο όρος έξυπνα αντικείμενα αποδίδεται σε μία ευρεία ποικιλία συσκευών, που ενσωματώνουν υλικό, λογισμικό, δεδομένα και υπηρεσίες. Τα έξυπνα αντικείμενα, με την ενσωμάτωσή τους σε κατοικίες και κτίρια, έχουν δημιουργήσει μία νέα τάση στην εξέλιξη των χώρων διαβίωσης, δουλειάς και διασκέδασης. Η τάση αυτή περιλαμβάνει την αυτοματοποίηση όλων των συστημάτων και των συσκευών, που υφίστανται στους χώρους αυτούς, όπως για παράδειγμα του συστήματος φωτισμού, του συστήματος θέρμανσης και των συσκευών που είναι υπεύθυνες για την ασφάλεια. Η παρακολούθηση και ο έλεγχος όλων αυτών των αυτοματοποιημένων συστημάτων, μέσω ενός υπολογιστή ή ενός κινητού τηλεφώνου, δίνει τελικά στα κτίρια το όνομα έξυπνα κτίρια και στις κατοικίες την ονομασία έξυπνα σπίτια ή ευφυείς κατοικίες (smart homes).

Οι λόγοι που επιλέγουν οι άνθρωποι σήμερα να μετατρέψουν το σπίτι τους σε ένα "έξυπνο σπίτι" δεν περιορίζονται μόνο από την ανάγκη τους για μία άνετη και γεμάτη διευκολύνσεις ζωή. Ένα έξυπνο σπίτι είναι ταυτόχρονα και ένα ασφαλές σπίτι, καθώς δίνει στους ενοίκους του τη δυνατότητα να επιτηρούν τους χώρους του σπιτιού όταν λείπουν, με κάμερες ασφαλείας και αισθητήρες ανίχνευσης κίνησης, οι οποίοι στέλνουν τα δεδομένα που λαμβάνουν στην τοποθεσία του χρήστη. Επιπλέον, έξυπνα συστήματα μέτρησης του επιπέδου CO₂ και της θερμοκρασίας στα δωμάτια του σπιτιού μπορούν να προλάβουν τις δυσάρεστες συνέπειες μίας πυρκαγιάς. Τέλος, ασφάλεια προσφέρει στους χρήστες η δυνατότητα που τους δίνεται να κλειδώνουν τις πόρτες και τα παράθυρα του σπιτιού απομακρυσμένα, με τη βοήθεια των έξυπνων κλειδαριών.

Ακόμα, η μετατροπή ενός σπιτιού σε έξυπνο σπίτι δεν απαιτεί πολύπλοκες διαδικασίες εγκατάστασης, καθώς συνήθως οι έξυπνες συσκευές μπορούν να ενσωματωθούν σε μία κατοικία εύκολα, γρήγορα και χωρίς την ανάγκη κάποιου ειδικού. Εξίσου εύκολη είναι και η χρήση τους, η οποία απαιτεί ελάχιστη εξοικείωση με την τεχνολογία, με μοναδικό προαπαιτούμενο τη γνώση χρήσης ενός σύγχρονου κινητού τηλεφώνου. Ωστόσο, τα οφέλη της χρήσης των έξυπνων τεχνολογιών εντός του ευρύτερου χώρου ενός σπιτιού, επεκτείνονται και στον οικονομικό τομέα, καθώς και στον τομέα της προστασίας του

περιβάλλοντος. Η παρακολούθηση της κατανάλωσης ηλεκτρικής ενέργειας και της τιμολόγησης σε ένα σύστημα φωτισμού ενός έξυπνου σπιτιού, καθώς και η ειδοποίηση σε περίπτωση αδικαιολόγητης χρήσης του δικτύου ύδρευσης, προσφέρει στους ενοίκους του σπιτιού τη δυνατότητα για εξοικονόμηση χρημάτων και πόρων.

Οι έξυπνες συσκευές τείνουν να λειτουργούν με την ελάχιστη απαιτούμενη ενέργεια, προσπαθώντας να αξιοποιήσουν τις ανανεώσιμες πηγές ενέργειας (ΑΠΕ), με αποτέλεσμα να μην οξύνουν -αν όχι να αμβλύνουν- το πρόβλημα της καταστροφής του περιβάλλοντος. Τέλος, η επένδυση στον οικιακό αυτοματισμό είναι σίγουρο ότι θα αποδειχτεί κερδοφόρα σε όσους την επιλέξουν, καθώς πέρα από τη γρήγορη απόσβεση λόγω της εξοικονόμησης ενέργειας που προσφέρει, αυξάνει την αξία μεταπώλησης της κατοικίας.

1.2 Περιγραφή προβλήματος

Ωστόσο, αν και τα πλεονεκτήματα που προσφέρει η υιοθέτηση έξυπνων τεχνολογιών στα πλαίσια μιας κατοικίας είναι πολλά, τα εμπορικά διαθέσιμα συστήματα στον τομέα αυτό τείνουν να εμφανίζουν ορισμένα κοινά μειονεκτήματα.

Αρχικά, το αυξημένο κόστος όλων των εμπορικά διαθέσιμων υλοποιήσεων λειτουργεί ως αποτρεπτικός παράγοντας για την εγκατάσταση έξυπνων συστημάτων σε μία κατοικία, καθώς απαιτούνται πολλά έτη λειτουργίας ώστε να πραγματοποιηθεί απόσβεση των εξόδων μέσω των ενεργειακών οφελών. Τα κόστη συντήρησης, αναβάθμισης ή και αντικατάστασης κάποιου συστατικού του συστήματος λόγω αστοχίας είναι επίσης σημαντικά, εφόσον οι παραπάνω εργασίες δύναται να γίνουν μόνο από τον εκάστοτε επίσημο κατασκευαστή.

Ακόμα, η αλληλεπίδραση των έξυπνων αντικειμένων με χρήστες μη εξοικειωμένους με την τεχνολογία γίνεται όλο και πιο δύσκολη όσο τα εμπορικά συστήματα αυξάνουν την πολυπλοκότητά τους, με στόχο τον εμπλουτισμό των προσφερόμενων υπηρεσιών και όχι με γνώμονα την ευκολία του χρήστη. Σε τέτοιες περιπτώσεις, οι χρήστες σταματούν να χρησιμοποιούν τις έξυπνες συσκευές τους, φοβούμενοι την πρόκληση ατυχημάτων.

Επιπλέον, οι εφαρμογές των επίσημων κατασκευαστών μπορούν να λειτουργήσουν σε μία από τις γνωστές πλατφόρμες και λειτουργικά συστήματα εκ των Windows, Linux, Android, iOS, και είναι "κλειστές", με την έννοια ότι όχι μόνο δε δίνουν πρόσβαση στον πηγαίο τους κώδικα, αλλά διατίθενται αποκλειστικά με την αγορά προϊόντων από τους ίδιους κατασκευαστές.

Το ίδιο συμβαίνει τόσο με το υλικό όσο και με τη διεπαφή προγραμματισμού της εφαρμογής (API), τα οποία δεν προσφέρονται προς χρήση και επεξεργασία. Τέλος, συνηθισμένα προβλήματα των διαθέσιμων εμπορικών συστημάτων στον τομέα των ευφών κατοικιών είναι η μειωμένη ικανότητα επεκτασιμότητας, η έλλειψη ευελιξίας και η ελλιπής διαλειτουργικότητα (interoperability), τα οποία οφείλονται στο μεγάλο ανταγωνισμό των εταιρειών του τομέα αυτού, που δεν έχει επιτρέψει την καθιέρωση παγκοσμίως αποδεκτών προτύπων

1.3 Στόχοι της Διπλωματικής

Κίνητρο για την εκπόνηση της παρούσας εργασίας αποτέλεσε η αντιμετώπιση των παραπάνω προβλημάτων στις υλοποιήσεις των έξυπνων κατοικιών. Στην προτεινόμενη υλοποίηση έχουν τεθεί ως στόχοι οι παρακάτω.

Αρχικά, η προτεινόμενη εφαρμογή θα πρέπει να μπορεί να λειτουργεί σε όλες τις γνωστές πλατφόρμες και λειτουργικά συστήματα (π.χ. Windows, Linux, Android, iOS), να είναι ανοιχτού κώδικα (open-source) το υλικό, το API και το λογισμικό της, και να διατίθεται δωρεάν προς μελέτη, αλλαγή και διανομή. Επιπρόσθετα, πρέπει να είναι εύκολα επεκτάσιμη και ευέλικτη, ώστε να φιλοξενήσει οποιονδήποτε αισθητήρα και ενεργοποιητή εύκολα και χωρίς την ανάγκη πολύωρων και πολύπλοκων διαδικασιών.

Το κόστος της προτεινόμενης υλοποίησης θα πρέπει να είναι το ελάχιστο δυνατό, τόσο κατά την διάρκεια της ανάπτυξης του βασικού συστήματος, όσο και κατά την διαδικασία αναβάθμισής του με επιπλέον αισθητήρες, εάν ο χρήστης το επιλέξει. Επιπλέον, η προτεινόμενη εφαρμογή πρέπει να είναι εύχρηστη και ικανή να χρησιμοποιηθεί ακόμη και από αρχάριους χρήστες, περιλαμβάνοντας απλό λεξιλόγιο και επεξηγηματικές εικόνες.

1.4 Διάρθρωση Διπλωματικής

Η ανάπτυξη και διάρθρωση του κειμένου της παρούσας διπλωματικής, δίνεται παρακάτω: Στην δεύτερη ενότητα ορίζονται αναλυτικότερα οι προδιαγραφές, βάσει των άνωθεν στόχων, και πραγματοποιείται η έρευνα εύρεσης των βέλτιστων λύσεων σε τεχνολογίες υλικού (αισθητήρες, μικροελεγκτές, οδηγούς κινητήρων) και λογισμικού. Γίνεται, επίσης, ενδεικτική χρήση και περιγραφή της λειτουργίας κάθε μεμονωμένου στοιχείου υλικού που επιλέχθηκε για την πλατφόρμα.

Στην τρίτη ενότητα πραγματοποιείται η υλοποίηση της πλατφόρμας, με την σύνδεση όλων των μεμονομένων στοιχείων που επιλέχθηκαν, την συγγραφή και επεξήγηση του κώδικα των σταθμών βάσης, της διεπαφής χρήστη και της Android εφαρμογής.

Η τέταρτη ενότητα περιλαμβάνει τα συμπεράσματα που προέκυψαν από την υλοποίηση της πλατφόρμας, και κατά πόσον επετεύχθησαν οι ορισμένοι στόχοι. Ακόμα, αναφέρονται μελλοντικές εξελίξεις της πλατφόρμας, αλλά και νέοι ερευνητικοί τομείς που σχετίζονται με αυτήν.

Η πέμπτη ενότητα αποτελεί την βιβλιογραφία.

Στην έκτη και τελευταία ενότητα δίνεται το παράρτημα, το οποίο περιέχει τον προτεινόμενο κώδικα που υλοποιήθηκε για την διεπαφή χρήσης της εφαρμογής, συνοδευόμενο από στιγμιότυπα οθόνης, τον κώδικα των δύο σταθμών βάσης, τον κώδικα της Android εφαρμογής, αλλά και βασικούς ορισμούς.

2. Ανάλυση εννοιών, θεωρητικό υπόβαθρο, προδιαγραφές

2.1 Η πλακέτα ανάπτυξης και οι προδιαγραφές της

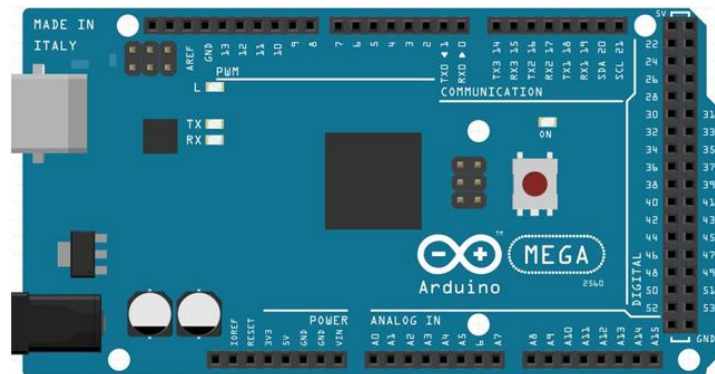
Για την άντληση και επεξεργασία των δεδομένων που αποστέλλονται από τους αισθητήρες, καθώς και ως μονάδα ελέγχου των φώτων και των κινητήρων, κρίνεται απαραίτητη η χρήση ενός μικροελεγκτή σε μία Development Board. Ως προϋπόθεση ορίστηκε η Development Board να αποτελεί OSH (Open Source Hardware), δηλαδή υλικό ανοιχτού κώδικα με ελεύθερα προσβάσιμα κυκλωματικά διαγράμματα, το οποίο επιτρέπει ευκολότερη αντιμετώπιση προβλημάτων αλλά και δυνατότητα παραμετροποίησης και επέκτασης. Ο μικροελεγκτής κρίθηκε προτιμότερη επιλογή από κάποιο SBPC (Single Board PC) όπως το Raspberry Pi, τόσο για το χαμηλότερο κόστος όσο και για την χαμηλότερη κατανάλωση ενέργειάς, τα οποία καθιστούν την προτεινόμενη υλοποίηση περισσότερο συμβατή με τεχνικές συγκομιδής ενέργειας (energy harvesting). Το πλήθος και το είδος των αναλογικών αισθητήρων καθιστούν απαραίτητη την χρήση ενός μικροελεγκτή με 16 αναλογικές εισόδους, με ανάλυση στον ADC του τουλάχιστον 10bit. Απαιτούνται, επίσης, τουλάχιστον 10 ψηφιακά Pins. Η υπολογιστική του ισχύς πρέπει να είναι αρκετή να φιλοξενήσει έναν Web Server, ώστε τα δεδομένα των αισθητήρων να μπορούν να είναι προσβάσιμα μέσω δικτύου, το οποίο ταυτόχρονα καθιστά απαραίτητη την συμβατότητα με κάποιο module επέκτασης με Ethernet Controller. Προτιμότερη είναι επίσης και η συμβατότητα με 5V Logic για την αποφυγή ανάγκης σύνδεσης κυκλωμάτων προσαρμογής, όπως διαίρετες τάσης ή logic level shifters στους αισθητήρες. Τέλος, στις προδιαγραφές της πλακέτας ανάπτυξης τέθηκε ως προϋπόθεση και η συμβατότητα με το Abstraction Layer του Arduino, ώστε ο κώδικας της προτεινόμενης υλοποίησης να είναι συμβατός με το πλήθος των συσκευών που το υποστηρίζουν.

2.2 Arduino Mega 2560

Η μονάδα επεξεργασίας των δεδομένων των αισθητήρων και η μονάδα ελέγχου των φώτων και των κινητήρων που επιλέχθηκαν και είναι εγκατεστημένες στο έξυπνο σπίτι της προτεινόμενης υλοποίησης, είναι δύο Arduino Mega2560 με αντίστοιχες Ethernet Shields. Το Arduino Mega2560 είναι μία πλακέτα που ενσωματώνει τον μικροελεγκτή Atmega2560, της οικογένειας μικροελεγκτών AVR, της εταιρείας Atmel.

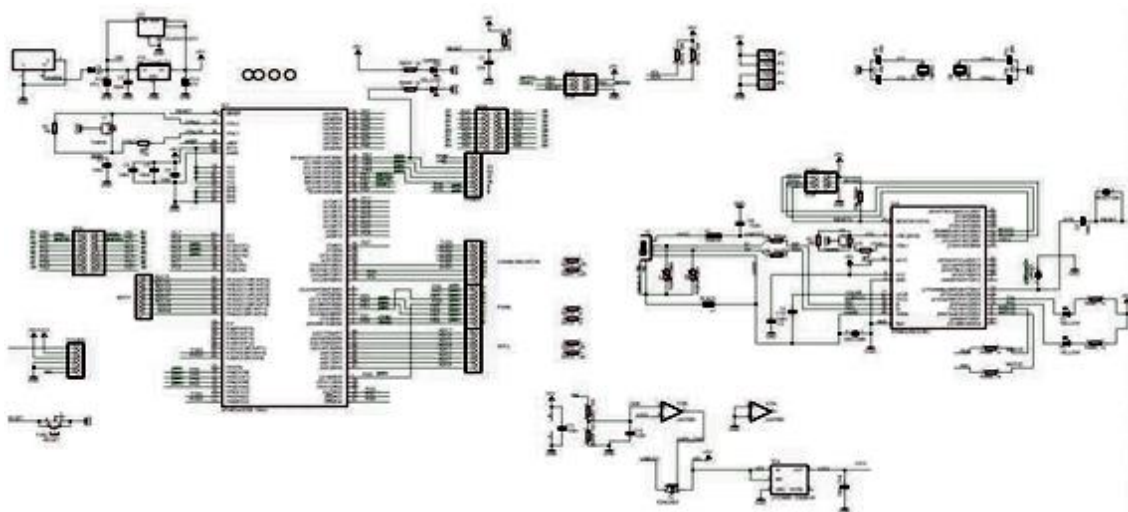
Διαθέτει 54 pins που μπορούν να χρησιμοποιηθούν ως ψηφιακές εισοδοί ή ψηφιακές έξοδοι, τα 15 μπορούν να χρησιμοποιηθούν ως έξοδοι διαμόρφωσης πλάτους παλμού (PWM - Pulse Width Modulation), 16 αναλογικές εισόδους, 4 κυκλώματα ενιαίου ασύγχρονου δέκτη/πομπού (UART - Universal Asynchronous Receiver/Transmitter, κύκλωμα

διαμεσολαβητής στη σειριακή επικοινωνία του μικροελεγκτή με άλλες συσκευές), έναν κρυσταλλικό ταλαντωτή στα 16MHz, μία θύρα ενιαίου σειριακού διαύλου (USB - Universal Serial Bus), μία υποδοχή ρεύματος, 6 pins για σειριακό προγραμματισμό εντός κυκλώματος (ICSP - In Circuit Serial Programming) και ένα κουμπί επαναφοράς (Reset). (Atmel, 2014)



Σχήμα 2-1. Το Arduino Mega 2560

Το Arduino Mega2560, όπως όλες οι πλακέτες Arduino, είναι υλικό ανοιχτού κώδικα, το οποίο έχει σχεδιαστεί για να είναι ευέλικτο και εύχρηστο, ώστε να δίνει τη δυνατότητα σε σχεδιαστές να δημιουργούν διαδραστικά αντικείμενα. Κάθε πλακέτα Arduino μπορεί να έρθει σε επαφή με το περιβάλλον της, λαμβάνοντας σήματα εισόδου, όπως για παράδειγμα φως από έναν αισθητήρα φωτός, το άγγιγμα ενός δαχτύλου πάνω σε ένα κουμπί, ή ακόμα και ένα μήνυμα από κάποιο μέσο κοινωνικής δικτύωσης, και μετατρέποντάς τα σε μία έξοδο, μπορεί να ενεργοποιήσει ένα μοτέρ, ένα LED (Light Emitting Diode), ή και να δημοσιεύσει κάτι στο διαδίκτυο.



Σχήμα 2-2. Σχηματική αναπαράσταση Arduino Mega 2560

Το γεγονός ότι είναι OSH έχει παίξει καθοριστικό ρόλο στο να θεσπιστεί μία παγκόσμια κοινότητα φοιτητών, ερασιτεχνών, καλλιτεχνών και άλλων επαγγελματιών, οι οποίοι με τη συνεισφορά τους έχουν δημιουργήσει ένα τεράστιο όγκο προσβάσιμης γνώσης, που βοηθά τόσο αρχάριους χρήστες όσο και ειδικούς να εξελίσσουν τις ιδέες τους και μαζί με αυτές να εξελίσσουν και το ίδιο το υλικό.

2.2.1 Arduino Ethernet Shield

Η Arduino Ethernet Shield επιτρέπει σε μία πλακέτα Arduino να συνδεθεί στο διαδίκτυο. Είναι βασισμένη στο Ethernet chip W5100, της εταιρείας Wiznet. Το Ethernet Chip W5100 παρέχει μία στοίβα πρωτοκόλλων διαδικτύου (TCP/IP) ικανή για TCP (Transmission Control Protocol) και UDP (User Datagram Protocol), υποστηρίζοντας έως 4 ταυτόχρονες συνδέσεις. (Wiznet, 2015)

Για να συνδεθεί η πλακέτα Arduino στο διαδίκτυο, εκτός από την Arduino Ethernet Shield, είναι αναγκαία η χρήση της βιβλιοθήκης Ethernet στον κώδικα, η οποία κάνει την πλακέτα Arduino να συμπεριφέρεται είτε ως εξυπηρετητής, ο οποίος δέχεται εισερχόμενες συνδέσεις, είτε ως πελάτης, ο οποίος δημιουργεί εξερχόμενες συνδέσεις.

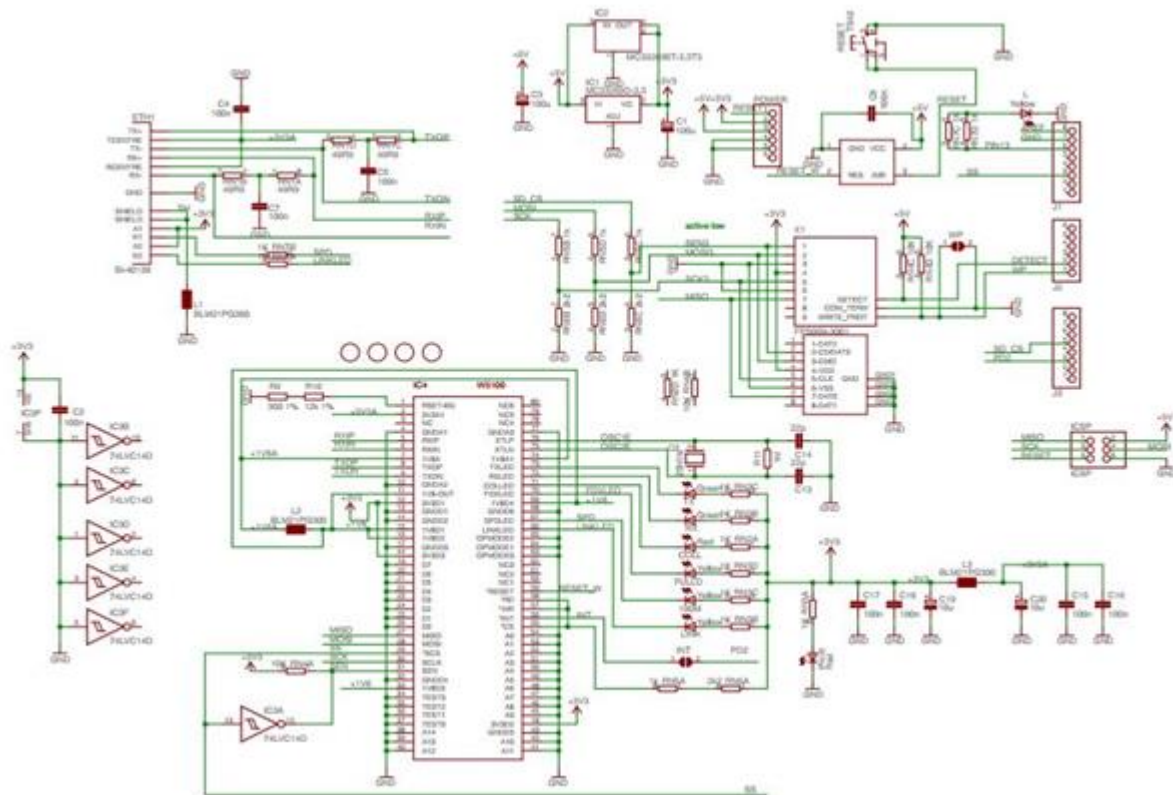


Σχήμα 2-3. Η συμβατή με το Arduino Mega 2560, Ethernet Shield

Η πλακέτα Arduino επικοινωνεί με την Arduino Ethernet Shield χρησιμοποιώντας το δίαυλο σειριακής διασύνδεσης περιφερειακών συσκευών (SPI – Serial Peripheral Interface). Αυτός

βρίσκεται στα pins 11,12,13 στην πλακέτα Arduino Uno και στα pins 50,51,52 στην πλακέτα Arduino Mega2560. Και στις δύο πλακέτες, το pin 10 χρησιμοποιείται ως SS (Slave Select). Στην πλακέτα Arduino Mega2560 το pin 53 δε χρησιμοποιείται για να επιλέγει το ethernet chip W5100, αλλά πρέπει να μένει ορισμένο ως έξοδος, διαφορετικά το SPI interface δε θα λειτουργεί.

Στις πλατφόρμες Arduino είναι δυνατή η τοποθέτηση περισσότερων της μίας Arduino Ethernet Shield, βάζοντας τη μία πάνω στην άλλη.



Σχήμα 2-4. Σχηματική αναπαράσταση Ethernet Shield

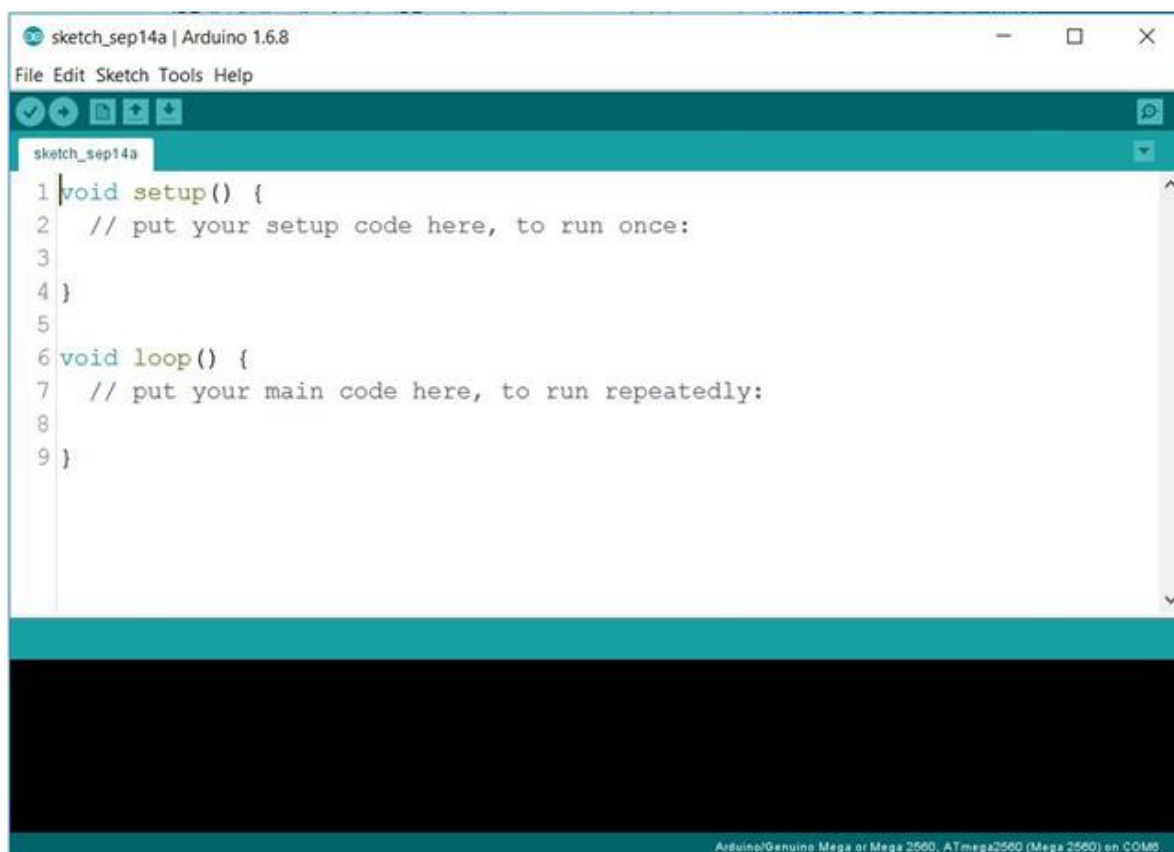
Η Arduino Ethernet Shield περιλαμβάνει μία υποδοχή κάρτας SD, η οποία μπορεί να χρησιμοποιηθεί για να αποθηκεύει αρχεία για εξυπηρετήσεις πέρα από το δίκτυο. Επιπλέον, η Arduino Ethernet Shield έχει έναν υποδοχέα για βύσμα RJ-45, πάνω στον οποίο συνδέεται το καλώδιο δικτύου. Στην παρούσα εργασία, ένα καλώδιο δικτύου ενώνει κάθε μία από τις δύο Arduino Ethernet Shields με ένα δρομολογητή (router), στον οποίο συνδέονται στη συνέχεια ασύρματα το κινητό ή ο υπολογιστής που περιλαμβάνουν την προτεινόμενη εφαρμογή.

Στην παρούσα εργασία, η μία από τις δύο Arduino Ethernet Shields που χρησιμοποιήθηκαν ήταν η Arduino Ethernet Shield 2, η οποία αποτελεί τη νέα έκδοση της Arduino Ethernet Shield. Η διαφορά τους εντοπίζεται στο ότι η νέα έκδοση έχει διαφορετικό ethernet chip, το

W5500 με περιοχή ενδιάμεσης μνήμης (buffer) χωρητικότητας 32KB, έναντι του ethernet chip W5100 της παλαιότερης έκδοσης -όπως προαναφέρθηκε- με χωρητικότητα buffer 16KB. Επίσης, για να είναι δυνατή η επικοινωνία της πλακέτας Arduino με τη νεότερη έκδοση Arduino Ethernet Shield 2 είναι απαραίτητη η χρήση της βιβλιοθήκης Ethernet2.h. Κατά τα άλλα, οι δύο πλακέτες είναι ίδιες, με ελάχιστες εσωτερικές βελτιώσεις στη νεότερη έκδοση, οι οποίες επηρεάζουν περισσότερο το κόστος και ελάχιστα τις επιδόσεις και τη λειτουργικότητά της.

2.2.2 Το Arduino IDE - Arduino Software

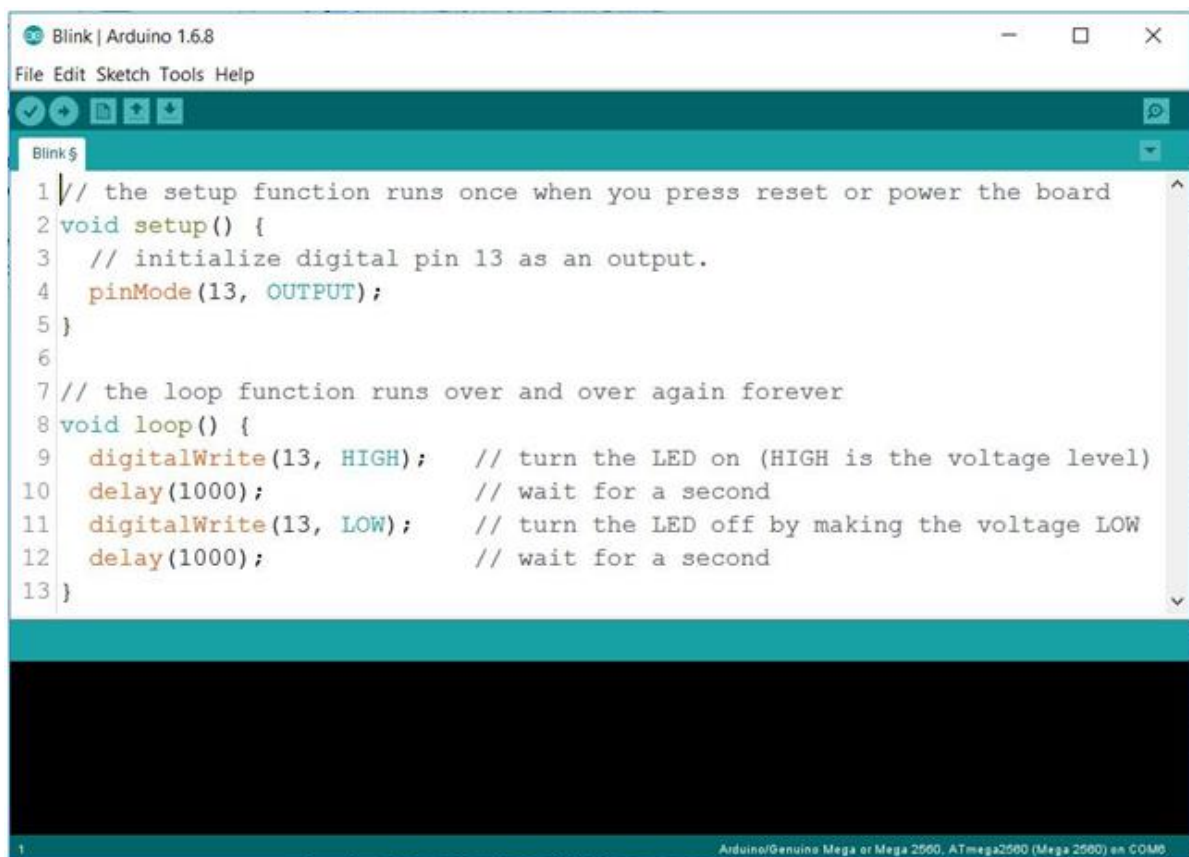
Το ολοκληρωμένο περιβάλλον ανάπτυξης Arduino – ή λογισμικό Arduino – περιέχει ένα πρόγραμμα επεξεργασίας κειμένου για τη σύνταξη του κώδικα, μία περιοχή για μηνύματα, μία κονσόλα κειμένου, μία γραμμή εργαλείων με κουμπιά για διάφορες λειτουργίες και μία σειρά από μενού. Συνδέεται με υλικό Arduino (Arduino hardware) για να μπορεί ο χρήστης να φορτώσει προγράμματα και να επικοινωνεί το υλικό με το λογισμικό.



Σχήμα 2-5. Το IDE του Arduino

Το ολοκληρωμένο περιβάλλον ανάπτυξης Arduino είναι ένα λογισμικό ανοιχτού κώδικα (open-source software), γραμμένο σε Java και βασισμένο στη γλώσσα προγραμματισμού Processing. Η γλώσσα προγραμματισμού που χρησιμοποιεί ο χρήστης είναι η Wiring (γλώσσα προγραμματισμού C++, σε συνδυασμό με βιβλιοθήκες υλοποιημένες επίσης σε C++), η οποία είναι αρκετά εύκολη στη σύνταξη και κατανοητή ακόμη και από κάποιον αρχάριο προγραμματιστή. Επίσης, το περιβάλλον υποστηρίζει τις γλώσσες προγραμματισμού C και C++. Ένα πρόγραμμα γραμμένο στο λογισμικό Arduino (IDE) ονομάζεται sketch και έχει τη μορφή [όνομα_αρχείου].ino.

Ένα απλό sketch που χρησιμοποιείται ως πρώτο παράδειγμα από έναν αρχάριο προγραμματιστή Arduino είναι το Blink.ino, το οποίο ενεργοποιεί για ένα δευτερόλεπτο και στη συνέχεια απενεργοποιεί το led που βρίσκεται πάνω στην πλακέτα Arduino και είναι συνδεδεμένο, σε σειρά με μία αντίσταση, στην ψηφιακή έξοδο 13.



Σχήμα 2-6. Ενδεικτικό Hello World (Blink) στο Arduino IDE

2.3 Αισθητήρες και οι προδιαγραφές τους

Ως αισθητήρας ορίζεται μία διάταξη που ανιχνεύει ένα φυσικό μέγεθος και παράγει από αυτό μία μετρήσιμη έξοδο. Στα ηλεκτρονικά κυκλώματα γίνεται χρήση αισθητήρων, των οποίων η μετρήσιμη έξοδος είναι τάση ή πιο σπάνια ένταση ρεύματος. Με τον τρόπο αυτό δύναται η μέτρηση να οδηγηθεί σε έναν μικροελεγκτή και να εμφανιστεί στον χρήστη, ή βάσει αυτής να πραγματοποιηθεί κάποια ενέργεια. Η χρήση αισθητήρων απαιτείται για την βελτίωση της ενεργειακής αποδοτικότητας σε μία ευφυή κατοικία, συνεπώς και κρίθηκε απαραίτητη και στην προτεινόμενη υλοποίηση.

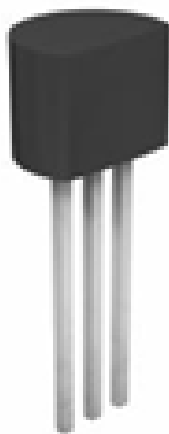
Η θερμοκρασία, η υγρασία, η φωτεινότητα, η ταχύτητα ανέμου, η τυχόν κίνηση, η περιεκτικότητα του αέρα σε διοξείδιο του άνθρακα αποτελούν τα φυσικά μεγέθη για τα οποία έγινε η επιλογή των αισθητήρων, που εγκαταστάθηκαν στην ευφυή κατοικία. Για την επιλογή του μοντέλου κάθε ενός από τους αισθητήρες αυτούς, ελήφθησαν υπ' όψιν παράμετροι όπως το εύρος λειτουργίας τους, η ακρίβειά τους, το κόστος τους, η γραμμικότητα της εξόδου τους, η διακριτική ικανότητά τους, η ευαισθησία τους, η απόκρισή τους, αλλά και το προσδόκιμο χρόνου ζωής τους.

Λοιπές παράμετροι που ελήφθησαν υπ' όψιν σχετίζονται με την χρήση πρωτοκόλλων επικοινωνίας με τους αισθητήρες, όπως τα SPI, I2C, CAN-BUS ή 1-Wire. Τα πρωτόκολλα αυτά επιτρέπουν την σύνδεση πολλών αισθητήρων σε έως τέσσερα Pins, αλλά ταυτόχρονα αυξάνουν εκθετικά το κόστος τους, κυρίως διότι στις διατάξεις τους περιλαμβάνουν και έναν βασικό μικροελεγκτή για την επικοινωνία. Τέλος, αυξάνουν και την απαιτούμενη επεξεργαστική ισχύ στην Development Board.

Οι επιλεγθέντες αισθητήρες στην προτεινόμενη υλοποίηση είναι οι: LM35D (θερμοκρασία), HIH-4030 (υγρασία), TEMT 6000 (φωτεινότητα), Modern Devices Wind Sensor Rev. C (ταχύτητα ανέμου), Parallax PIR (κίνηση) και K-30 (περιεκτικότητα αέρα σε διοξείδιο του άνθρακα).

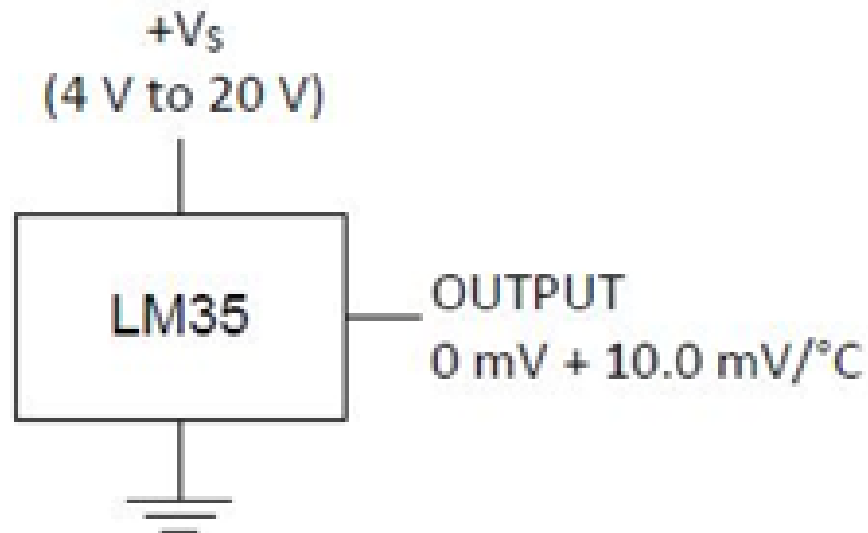
2.4 Αισθητήρας θερμοκρασίας LM35D

Για τη μέτρηση της θερμοκρασίας στους χώρους της κατοικίας χρησιμοποιήθηκε ο αισθητήρας της Texas Instruments, LM35D. Η σειρά αισθητήρων LM35 είναι ακριβείς αισθητήρες θερμοκρασίας ολοκληρωμένου κυκλώματος (integrated-circuit, IC), των οποίων η τάση εξόδου είναι γραμμικά ανάλογη με τη θερμοκρασία σε βαθμούς Κελσίου. Έτσι, ο LM35 έχει ένα πλεονέκτημα σε σχέση με τους γραμμικούς αισθητήρες θερμοκρασίας, οι οποίοι βαθμονομούνται σε βαθμούς Κέλβιν, καθώς δεν απαιτεί από το χρήστη να αφαιρεί μία μεγάλη σταθερή τάση από την έξοδο που παρατηρεί για να επιτύχει την κατάλληλη μετατροπή στην κλίμακα Κελσίου. (Texas Instruments, 2016)



Σχήμα 2-7. Ο αισθητήρας θερμοκρασίας LM35D

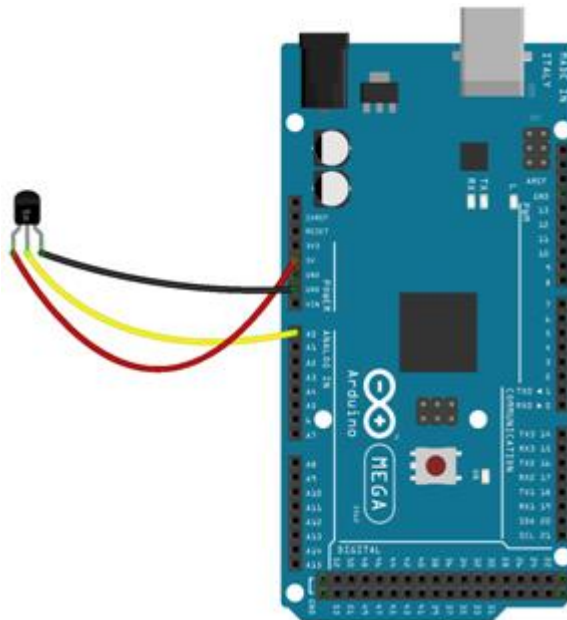
Ο LM35D δεν απαιτεί καμία εξωτερική βαθμονόμηση ή άλλη ρύθμιση και παρέχει τυπικά ακρίβεια $\pm 1.5^{\circ}\text{C}$ σε θερμοκρασία δωματίου στο εύρος θερμοκρασιών 0 έως $+100^{\circ}\text{C}$. Το χαμηλό κόστος εξασφαλίζεται με τη βαθμονόμηση σε επίπεδο wafer (φέτας ημιαγωγού). Η χαμηλή αντίσταση και η γραμμικότητα της εξόδου, καθώς και η ακριβής εγγενής βαθμονόμηση κάνουν τη διεπαφή του χρήστη με τα αποτελέσματα της εξόδου και με το κύκλωμα ελέγχου εξαιρετικά εύκολη. Ο αισθητήρας μπορεί να χρησιμοποιηθεί με μονή τροφοδοσία ή με θετική και αρνητική τροφοδοσία. Καθώς αντλεί μόνο 60μΑ από την τροφοδοσία του, έχει πολύ χαμηλή αυτο-θέρμανση, λιγότερο από 0.1°C σε νηνεμία.



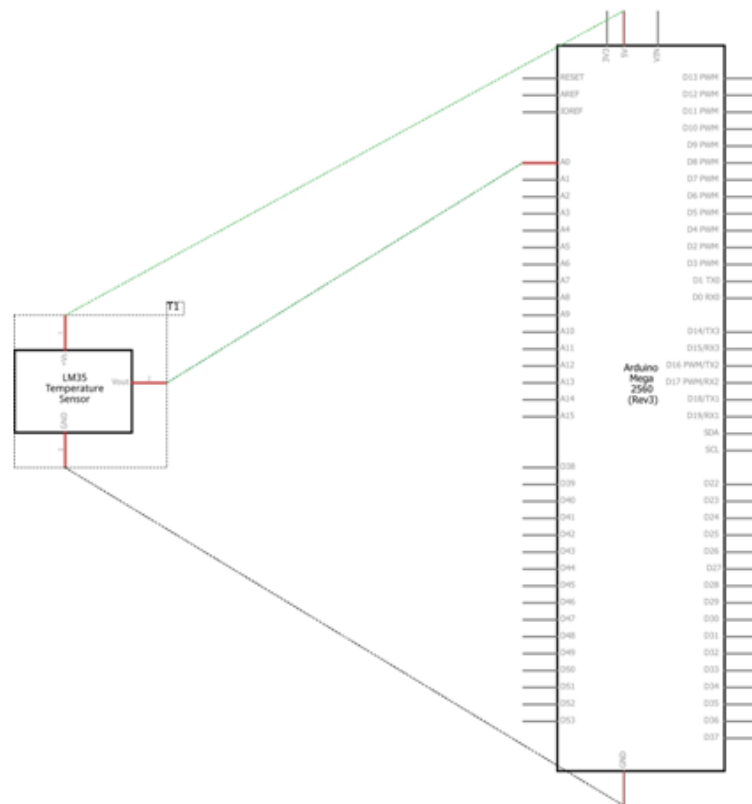
Σχήμα 2-8. Σχηματικό διάγραμμα αισθητήρα LM35

2.4.1 Συνδεσμολογία LM35D με Arduino

Ο LM35D μπορεί να τροφοδοτηθεί από 4-20V. Για την τροφοδοσία του χρησιμοποιήθηκε η έξοδος 5V του Arduino. Η αναλογική έξοδος του αισθητήρα συνδέθηκε στην αναλογική είσοδο A0 του Arduino και το pin GND του αισθητήρα στο GND του Arduino.



Σχήμα 2-9. Συνδεσμολογία LM35D με Arduino



Σχήμα 2-10. Σχηματικό συνδεσμολογίας LM35D με Arduino

2.4.2 Κώδικας Arduino για αισθητήρα LM35D

```
int lm35Pin=A0;
void setup() {
  Serial.begin(9600);
}
void loop() {
  int temperature=calculateTemperature(lm35Pin);
}
int calculateTemperature(int lm35Pin) {
  int sum=0;
  int temp;
  for (int j=0;j<=10;j++){
    temp = (5.0 * analogRead(lm35Pin) * 100.0) / 1024; //temp se kelsiou
    sum+=temp;
  }
  temp=sum/10;
  return temp;
}
```

2.4.3 Αισθητήρας υγρασίας H1H-4030

Για την μέτρηση του ποσοστού της υγρασίας στα δωμάτια χρησιμοποιήθηκε ο αισθητήρας της Honeywell, H1H-4030 και, πιο συγκεκριμένα, το breakout board της Sparkfun. Ο αισθητήρας H1H-4030 μπορεί εύκολα να συνδεθεί με έναν μικροελεγκτή ή κάποια άλλη συσκευή, δίνοντας σχεδόν γραμμική τάξη εξόδου. Εξαιτίας της χαμηλής κατανάλωσής του (τυπικά 200μΑ), είναι ιδανικός για χρήση σε συστήματα που πρέπει να λειτουργούν με μπαταρία. Επιπλέον, η υψηλή ικανότητα εναλλαξιμότητας (interchangeability) του αισθητήρα μειώνει σημαντικά το κόστος παραγωγής και βαθμονόμησής του. (Honeywell, 2016)



Σχήμα 2-11. Ο αισθητήρας υγρασίας H1H-4030

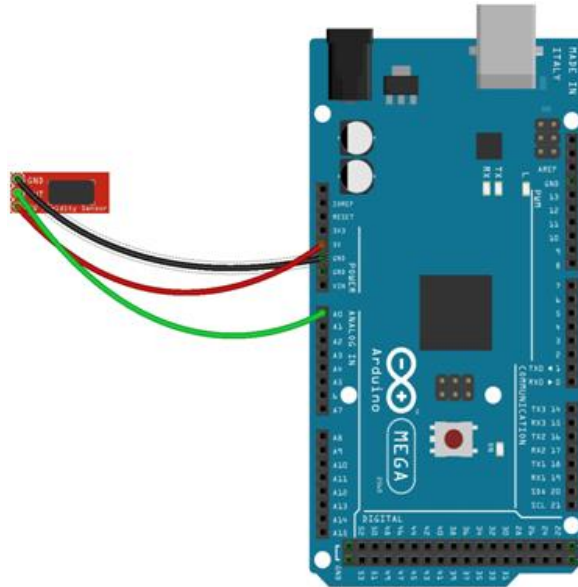
Ο αισθητήρας H1H-4030 εγγυάται άριστη απόδοση στη μέτρηση της σχετικής υγρασίας (RH - Relative Humidity), σε συνδυασμό με χαμηλή τιμή και διατίθεται σε μορφή στοιχείου επιφανειακής τοποθέτησης (SMD - Surface Mount Device). Ο προτεινόμενος τρόπος υπολογισμού της υγρασίας περιλαμβάνει διόρθωση σφάλματος μέσω θερμοκρασίας.



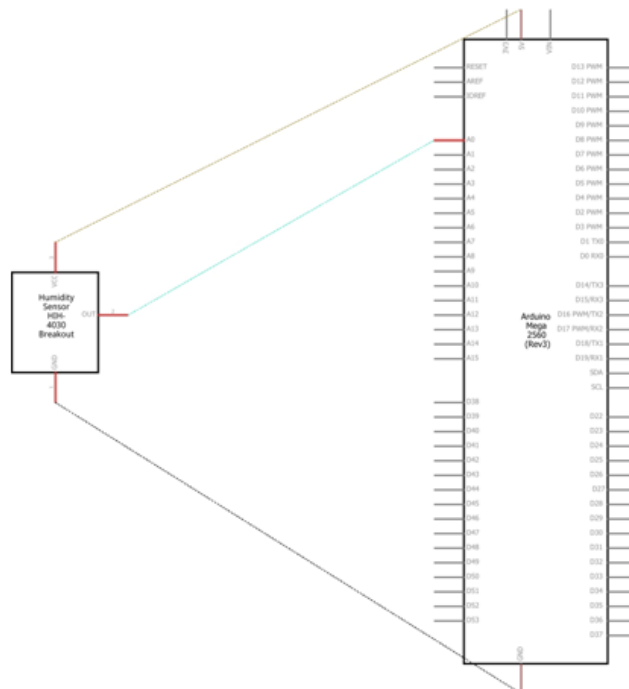
Σχήμα 2-12. Σχηματικό διάγραμμα αισθητήρα υγρασίας H1H-4030

2.4.4 Συνδεσμολογία ΗΙΗ-4030 με Arduino

Βάσει του datasheet, ο ΗΙΗ-4030 μπορεί να τροφοδοτηθεί από 4V έως 5,8V. Για την τροφοδοσία του χρησιμοποιήθηκε η έξοδος 5V του Arduino. Η αναλογική έξοδος του αισθητήρα συνδέθηκε με την αναλογική είσοδο Α0 του Arduino και το pin GND του αισθητήρα στο GND του Arduino.



Σχήμα 2-13. Συνδεσμολογία ΗΙΗ-4030 με Arduino



Σχήμα 2-14. Σχηματικό συνδεσμολογίας ΗΙΗ-4030 με Arduino

2.4.5 Κώδικας Arduino για αισθητήρα HIH-4030

```
int HIH4030_Pin = A0; //analogiko pin sto opoio sindeoume ton aisthitira

void setup() {
    Serial.begin(9600); //arxizoume tin seiriaki epikoinwnia
}

void loop() {
    float temperature = 25; //ipothetoume thermokrasia dwmatiou 25 vathmous
    //kelsiou, kathws ephreazei tin metrisi.

    //Sto teliko project poy metrame tin thermokrasia me allo aisthitira, tha
    //dwsoume tin pragmatiki thermokrasia
    float relativeHumidity = getHumidity(temperature);
    Serial.println(relativeHumidity);
}

float getHumidity(float degreesCelsius) {

    float supplyVolt = 5.0; //metavliti stin opoia dilwnoume tin tasi
    //trofodosias tou aisthitira.

    //Xrhsimopoioume ws trofodosia tin 5V exodo tou arduino
    int HIH4030_Value = analogRead(HIH4030_Pin); //diavasma mesw analogRead
    float voltage = HIH4030_Value / 1023. * supplyVolt;

    // lamvanontas ip opsin tin akriveia toy ADC, metatrepoume tin timi tis
    //exodou tou aisthitira se tasi. Sto arduino exoume akriveia 10bit, ara 2^10
    float sensorRH = 161.0 * voltage / supplyVolt - 25.8;

    //ipologismos sxetikis igrasias vasei datasheet, gia thermokrasia dwmatiou
    //25 vathmous kelsiou
    float trueRH = sensorRH / (1.0546 - 0.00216 * degreesCelsius); //diorthwsi
    //lamvanontas ip opsin kai tin thermokrasia
    return trueRH;
}
```

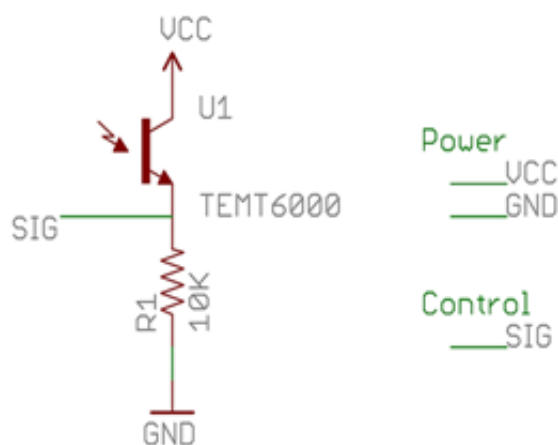
2.5 Αισθητήρας φωτός περιβάλλοντος TEMT6000

Για τη μέτρηση της έντασης του φωτός στους χώρους του σπιτιού χρησιμοποιήθηκε ο αισθητήρας της Vishay, TEMT6000 και, πιο συγκεκριμένα, το breakout board της Sparkfun. Ο αισθητήρας TEMT6000 είναι ένα επιταξιακό φωτοτρανζίστορ πλανάρ, τύπου NPN, από πυρίτιο, μέσα σε ένα μικροσκοπικό διαφανές καλούπι για επιφανειακή τοποθέτηση πάνω σε ένα PCB (Printed Circuit Board). (Vishay, 2011)



Σχήμα 2-15. Ο αισθητήρας φωτός TEMT6000

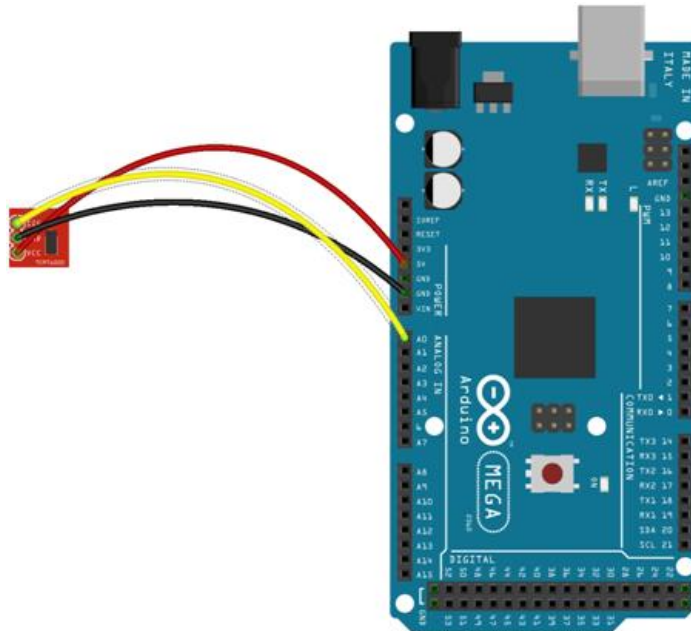
Ο αισθητήρας είναι ευαίσθητος στο ορατό φάσμα. Η λειτουργία του είναι απλή. Όσο περισσότερο το φως που προσπίπτει στον αισθητήρα τόσο μεγαλύτερη η αναλογική τάση στο pin του σήματος εξόδου.



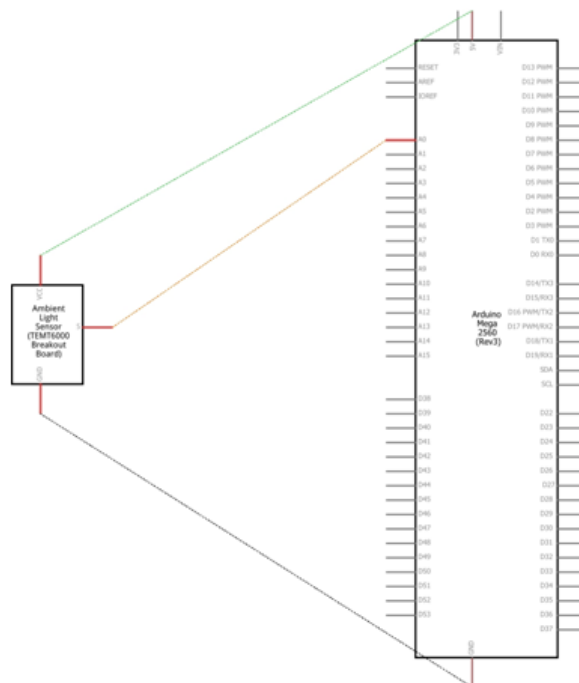
Σχήμα 2-16. Σχηματικό διάγραμμα αισθητήρα TEMT6000

2.5.1 Συνδεσμολογία TEMT6000 με Arduino

Για την τροφοδοσία του TEMT6000 χρησιμοποιήθηκε η έξοδος 5V του Arduino. Η αναλογική έξοδος του αισθητήρα συνδέθηκε στην αναλογική είσοδο A0 του Arduino και το pin GND του αισθητήρα στο GND του Arduino.



Σχήμα 2-17. Συνδεσμολογία TEMT6000 με Arduino



Σχήμα 2-18. Σχηματικό συνδεσμολογίας TEMT6000 με Arduino

2.5.2 Κώδικας Arduino για αισθητήρα TEMT6000

```
int temt6000Pin = A0;

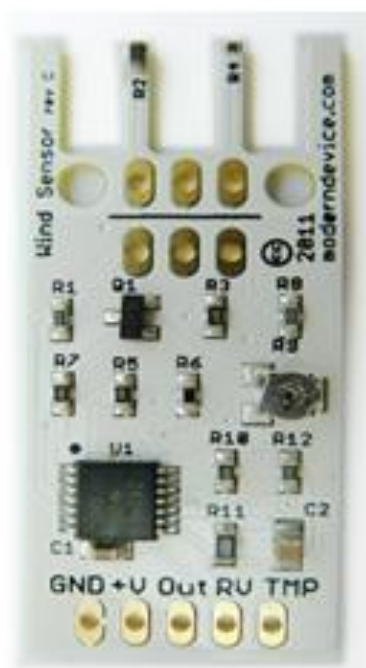
void setup() {
  Serial.begin(9600);
}

void loop() {
  int illuminance = calculateLux(temt6000Pin);
}

int calculateLux(int temt6000Pin) {
  int illuminance = analogRead(temt6000Pin)* 0.9765625;//fwteinotita se lux
  return illuminance;
}
```

2.6 Αισθητήρας ανέμου Wind Sensor Rev. C

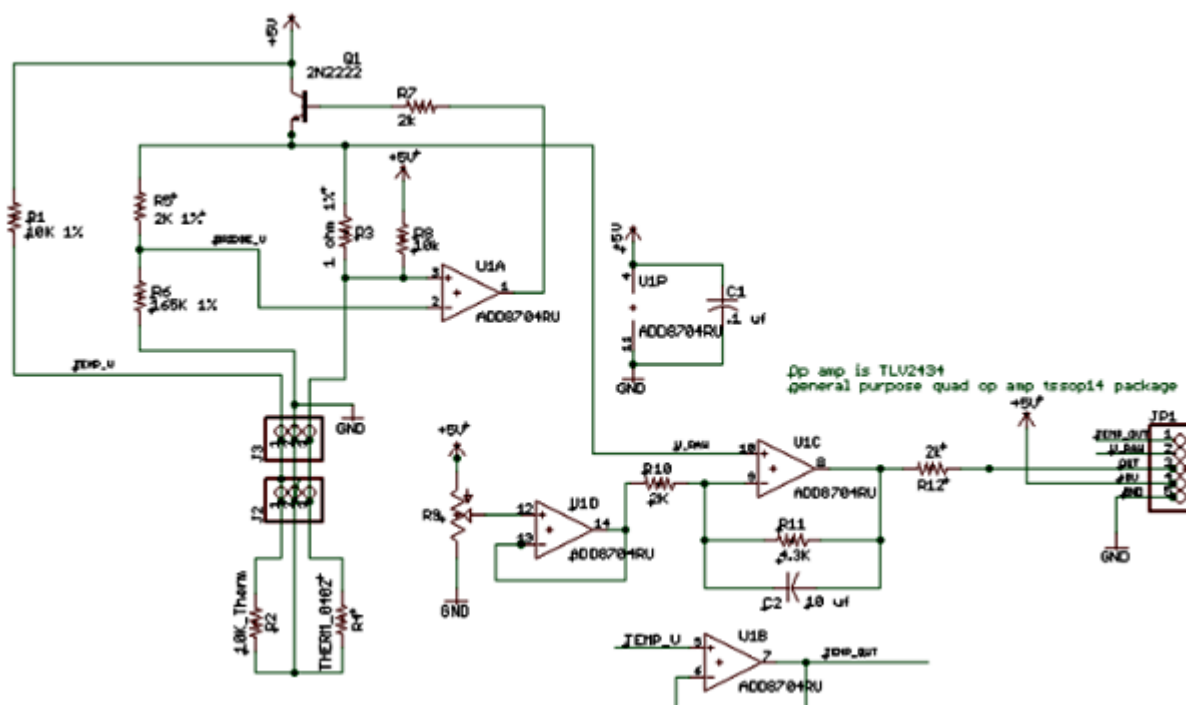
Για τη μέτρηση της ταχύτητας του ανέμου χρησιμοποιήθηκε ο αισθητήρας της Modern Device, Wind Sensor Rev. C. Πρόκειται για ένα θερμικό ανεμόμετρο χαμηλού κόστους, σχεδιασμένο για χρήση σε πειράματα με ηλεκτρονικά. Η λειτουργία του βασίζεται στην τεχνική “hot-wire” (θερμό σύρμα), η οποία περιλαμβάνει τη θέρμανση ενός στοιχείου σε μια σταθερή θερμοκρασία και, στη συνέχεια, μέτρηση της ηλεκτρικής ισχύος που απαιτείται για να διατηρηθεί το θερμασμένο στοιχείο στη θερμοκρασία αυτή, καθώς αλλάζει ο άνεμος. Αυτή η μετρούμενη ηλεκτρική ισχύς είναι ευθέως ανάλογη με το τετράγωνο της ταχύτητας του ανέμου. Η τεχνική “hot-wire” υπερέχει στη μέτρηση χαμηλών έως μεσέων ταχυτήτων ανέμου και είναι η προτιμώμενη τεχνική για ανίχνευση της κίνησης του αέρα σε εσωτερικούς χώρους, όπου τα κυπελλοφόρα ανεμόμετρα -τα οποία τυπικά συναντώνται σε μετεωρολογικού σταθμούς- είναι αναποτελεσματικά. Τα “hot-wire” ανεμόμετρα είναι συνήθως διαθέσιμα σε φορητές συσκευές, που μοιάζουν με πολύμετρα και κοστίζουν περίπου 250€.



Σχήμα 2-19. Ο αισθητήρας ανέμου Wind Sensor Rev. C

Ως πειραματικό εργαλείο, ο αισθητήρας είναι εξαιρετικά ευαίσθητος, με ένα μικρό φύσημα του ανέμου να ανιχνεύεται σε απόσταση 45-60 cm. Ο αισθητήρας μπορεί να απενεργοποιηθεί για εξοικονόμηση ενέργειας, αλλά στην εκκίνηση χρειάζεται να του επιτραπεί να ζεσταθεί για περίπου 10 δευτερόλεπτα για να σταθεροποιηθεί θερμικά, για βέλτιστη ακρίβεια. Αυτό το

χρονικό διάστημα επιτρέπει στο θερμίστορ να φτάσει σε θερμοκρασία λειτουργίας. Επίσης, η τροφοδοσία του αισθητήρα μέσω ενός μικροελεγκτή επηρεάζει την ακρίβειά του και δε συνιστάται, καθώς το ρεύμα που αντλεί ο αισθητήρας μεταβάλλεται με την ταχύτητα του ανέμου. Ο Wind Sensor Rev. C έχει ένα μικρό ποτενσιόμετρο που χρησιμοποιείται για τη βαθμονόμηση του αισθητήρα για μηδενικό άνεμο. Η βαθμονόμηση είναι απλή. Ένα γυάλινο ποτήρι τοποθετείται πάνω από τον αισθητήρα για να μπλοκάρει τη ροή του αέρα προς αυτόν και με κατάλληλη ρύθμιση του ποτενσιομέτρου επιτυγχάνεται το επιθυμητό μηδενικό επίπεδο ανέμου.



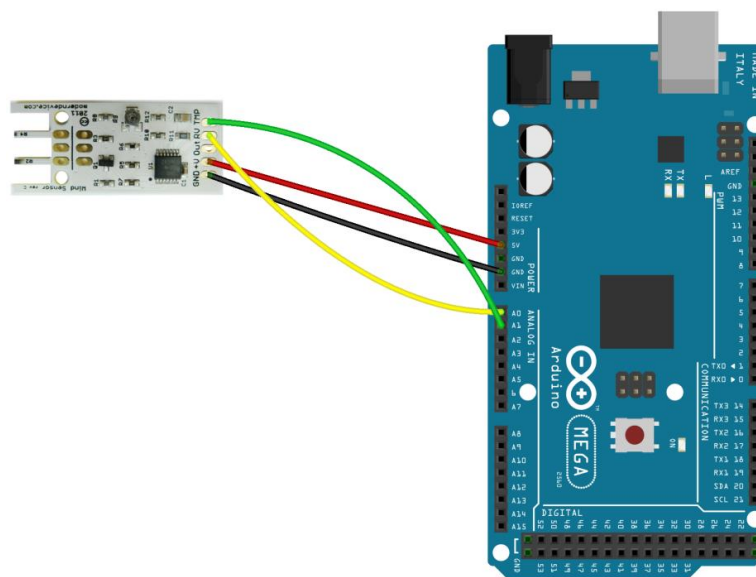
Σχήμα 2-20. Σχηματικό διάγραμμα αισθητήρα Wind Sensor Rev. C

Οι αισθητήρες Wind Sensor Rev. C βαθμονομούνται περίπου στα 0.5V της εξόδου για μηδενικό άνεμο, με 6V παροχή. Ένα χαμηλότερο σημείο βαθμονόμησης, π.χ. 0.2V, θα έχει ως αποτέλεσμα μία λίγο μεγαλύτερη ικανότητα ανίχνευσης. Η χρησιμοποίηση μεγαλύτερης τάσης τροφοδοσίας απαιτεί βαθμονόμηση εκ νέου. Απαγορεύεται η χρήση τάσεων τροφοδοσίας μεγαλύτερων των 10V. Μία ρυθμιζόμενη παροχή συνιστάται.

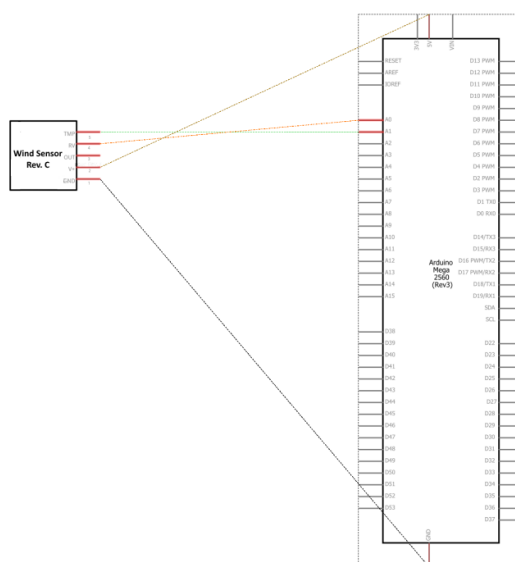
Ο αισθητήρας μπορεί να τροφοδοτηθεί από 4 έως 10 volts και καταναλώνει 20 έως 40 mA, ανάλογα με την ταχύτητα του ανέμου. Το σήμα εξόδου του είναι αναλογικό, με την τιμή του να κυμαίνεται από 0 έως VCC (τάση εισόδου). (Modern Device, 2016)

2.6.1 Συνδεσμολογία Wind Sensor Rev. C με Arduino

Ο Wind Sensor Rev. C έχει 5 pins. Η σύνδεσή του με το Arduino απαιτεί την σύνδεση του GND και της τροφοδοσίας του αισθητήρα στα GND και +5V του Arduino αντίστοιχα, ενώ οι έξοδοι RV και TMP συνδέονται στις εισόδους A0 και A1. Το pin Out μένει ασύνδετο, καθώς η έξοδος αυτή χρησιμοποιεί τη ρύθμιση του ποτενσιόμετρου και έχει χαμηλό σημείου κορεσμού, σε περίπτωση που η θερμοκρασία του περιβάλλοντος αλλάξει πολύ σε σχέση με τη θερμοκρασία βαθμονόμησης του αισθητήρα. Ως pin εξόδου του αισθητήρα χρησιμοποιείται το pin RV.



Σχήμα 2-21. Συνδεσμολογία Wind Sensor Rev. C με Arduino



Σχήμα 2-22. Σχηματικό συνδεσμολογίας Wind Sensor Rev. C με Arduino

Στον παρακάτω πίνακα συνοψίζεται η λειτουργία των pins του αισθητήρα ανέμου Wind Sensor Rev. C.

Πίνακας 2-1. Επεξήγηση των Pins του Wind Sensor Rev C.	
Ground	Γείωση
+V	Τροφοδοσία – 5 έως 10 volts – συνιστάται ρυθμιζόμενη παροχή για καλύτερα αποτελέσματα
Out	Τάση εξόδου. Είναι η τάση στο RV πολλαπλασιασμένη κατά περίπου 3 και ολισθημένη κατά τη ρύθμιση του ποτενσιομέτρου. Αυτή η τάση εξόδου θα κορεστεί (φτάσει το Ground ή το +V) αν η θερμοκρασία του περιβάλλοντος αλλάξει πολύ σε σχέση με τη θερμοκρασία όταν βαθμονομήθηκε ο αισθητήρας. Η ευαισθησία αυτής της εξόδου μπορεί να αλλάξει αλλάζοντας την αντίσταση R11. Μεγαλύτερη ευαισθησία θα κάνει τον αισθητήρα να κορεστεί πιο εύκολα με την αλλαγή της θερμοκρασίας του περιβάλλοντος.
RV	Αυτή είναι η τάση μετά τον A/D μετατροπέα – και δε θα κορεστεί. Επίσης, δε θα πέσει κάτω από 1,8V για θερμοκρασίες δωματίου. Αυτή η τάση δεν επηρεάζεται από τη βαθμονόμηση του ποτενσιομέτρου. Η έξοδος του αισθητήρα είναι λογαριθμική, το οποίο σημαίνει ότι ο αισθητήρας μπορεί να εντοπίσει πολύ μικρές κινήσεις του αέρα, καθώς επίσης και ότι δε θα κορεστεί μέχρι η ροή αέρα να φτάσει περίπου τα 60 μίλια την ώρα. Το ενδεικτικό Arduino sketch για χρήση με τον αισθητήρα αυτό έχει κατασκευαστεί να χρησιμοποιεί το pin αυτό ως έξοδο.
TMP	Μία έξοδος θερμοκρασίας. Αυτό είναι απλά μία αντίσταση σε ένα κύκλωμα διαιρέτη τάσης και ένα θερμίστορ. Θα διαβάζει περίπου 2.8V σε θερμοκρασία δωματίου, θα πέφτει για μεγαλύτερες θερμοκρασίες και θα ανεβαίνει για χαμηλότερες θερμοκρασίες. Δε θα κορεστεί ποτέ. Ένας αλγόριθμος για να μεταφράσει την τάση σε θερμοκρασία περιέχεται στο sketch παράδειγμα.

2.6.2 Κώδικας Arduino για αισθητήρα Wind Sensor Rev. C

```
#define analogPinForRV    1    //to rv sindeetai sto analogiko pin1
#define analogPinForTMP   0    //to tmp sindeetai sto analogiko pin0

float WindSpeed_KPH;
const float zeroWindAdjustment = .2; //gia calibration tou aisthitira

int TMP_Therm_ADunits; //i thermokrasia apo to thermistor tou windsensor
float RV_Wind_ADunits; //i exodos RV tou windsensor
float RV_Wind_Volts;
int TempCtimes100;
float zeroWind_ADunits;
float zeroWind_volts;
float WindSpeed_MPH;

void setup() {
    Serial.begin(9600);
}

void loop() {
    float aeras = calculateWindspeed();
    Serial.println("WindSpeed KPH:");
    Serial.println(aeras);
    delay(1000);
}

float calculateWindspeed() {
    TMP_Therm_ADunits = analogRead(analogPinForTMP);
    RV_Wind_ADunits = analogRead(analogPinForRV);
    RV_Wind_Volts = (RV_Wind_ADunits * 0.0048828125);
    TempCtimes100 = (0.005 * ((float)TMP_Therm_ADunits *
    (float)TMP_Therm_ADunits)) - (16.862 * (float)TMP_Therm_ADunits) + 9075.4;
    zeroWind_ADunits = -0.0006 * ((float)TMP_Therm_ADunits *
    (float)TMP_Therm_ADunits) + 1.0727 * (float)TMP_Therm_ADunits + 47.172;
    zeroWind_volts = (zeroWind_ADunits * 0.0048828125) - zeroWindAdjustment;
    WindSpeed_MPH = pow(((RV_Wind_Volts - zeroWind_volts) / .2300) , 2.7265);
    WindSpeed_KPH = WindSpeed_MPH * 1.60934; //metatropi apo mph se kmh
    return WindSpeed_KPH;
}
```

2.7 Αισθητήρας διοξειδίου του άνθρακα K-30

Ο αισθητήρας διοξειδίου του άνθρακα K-30 είναι ένας χαμηλού κόστους, υπέρυθρος και χωρίς απαιτήσεις συντήρησης πομπός, ο οποίος είναι κατάλληλος για ενσωμάτωση σε συσκευές που απαιτούν παρακολούθηση δεδομένων CO₂. Επιπλέον, είναι ένας ακριβής αισθητήρας, που μπορεί να χρησιμοποιηθεί από κατασκευαστές υλικού, οι οποίοι εύκολα και χωρίς μεγάλες επενδύσεις μπορούν να τον εντάξουν στα προϊόντα τους. Το γεγονός ότι είναι μικρού μεγέθους και χαμηλών απαιτήσεων ισχύος τον κάνει να μπορεί εύκολα να αποτελέσει ένα πρόσθετο στοιχείο σε εξοπλισμό που βασίζεται σε μικροελεγκτή.

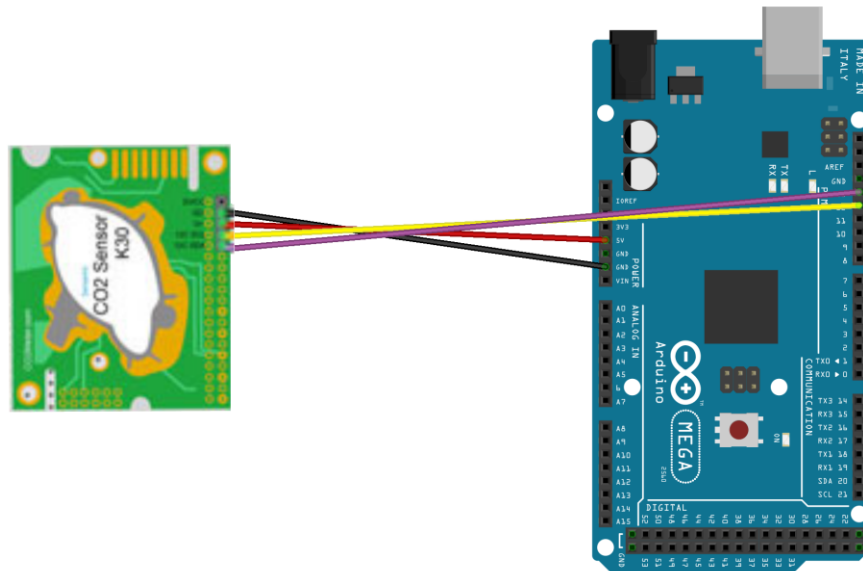


Σχήμα 2-23. Ο αισθητήρας διοξειδίου του άνθρακα K-30

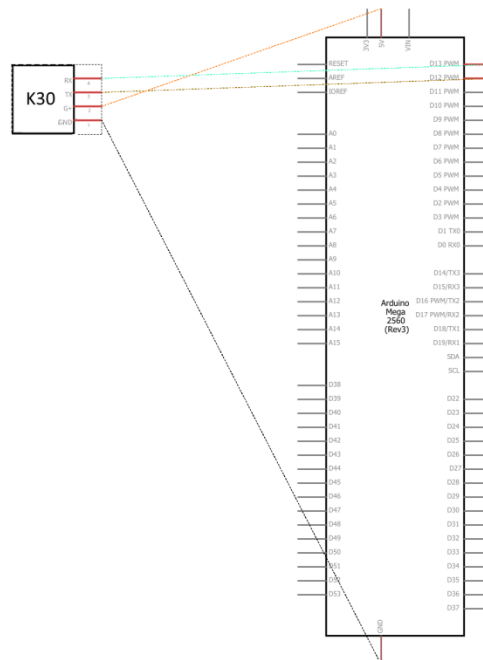
Ο αισθητήρας CO₂ K-30 είναι ιδανικός για εφαρμογές σε κατοικίες, σε μονάδες διαχείρισης αέρα, καθώς και σε συστήματα συναγερμού, καθώς μπορεί να λειτουργήσει σε θερμοκρασίες 0 έως +50 ° C, σε επίπεδα υγρασίας 0-95% RH και έχει διάρκεια ζωής μεγαλύτερης των 15 χρόνων. Επίσης, ο χρόνος που απαιτείται για να ζεσταθεί είναι λιγότερος από 1 λεπτό και μπορεί να μετρήσει περιεκτικότητες CO₂ από 0-10000 ppm, με ακρίβεια ±30ppm. Η μέθοδος που χρησιμοποιεί για να μετρήσει την περιεκτικότητα του αέρα σε CO₂ είναι κυματοδηγός τεχνολογία υπέρυθρης ακτινοβολίας χωρίς διάχυση (NDIR – Non-Dispersive Infrared) με αυτόματο αλγόριθμο βαθμονόμησης (ABC – Automatic Background Calibration). Υπάρχουν πολλοί διαφορετικοί τρόποι να συνδεθεί ο αισθητήρας CO₂ K-30 σε ένα σύστημα-υποδοχέα. Στην παρούσα εργασία έχει συνδεθεί με το Arduino χρησιμοποιώντας επικοινωνία UART (με ρυθμό μετάδοσης δεδομένων τα 9600bits/sec), η οποία περιλαμβάνει σύνδεση των pins G+ (τροφοδοσία), G0 (γείωση), TxD και RxD. (CO₂meter, 2015)

2.7.1 Συνδεσμολογία K-30 με Arduino

Ο αισθητήρας CO₂ K-30 μπορεί να τροφοδοτηθεί από 5-14V και αντλεί κατά μέσο όρο 40mA. Για την τροφοδοσία του (G+) χρησιμοποιήθηκε η έξοδος 5V του Arduino. Επιπλέον, το pin TxD του αισθητήρα συνδέθηκε με το D13 του Arduino, το pin RxD του αισθητήρα με το D12 του Arduino και το pin G0 του αισθητήρα στο GND του Arduino.



Σχήμα 2-24. Συνδεσμολογία K-30 με Arduino



Σχήμα 2-25. Σχηματικό συνδεσμολογίας K-30 με Arduino

2.7.2 Κώδικας Arduino για αισθητήρα διοξειδίου του άνθρακα K-30

Η βιβλιοθήκη k.Series είναι απαραίτητη για τη σωστή λειτουργία του αισθητήρα

```
#include "kSeries.h"
kSeries K_30(13,12);

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  double co2 = K_30.getCO2('p'); //epistrefei se ppm tin timi tou co2

  Serial.print("Co2 ppm = ");
  Serial.println(co2);
  delay(1500);
}
```

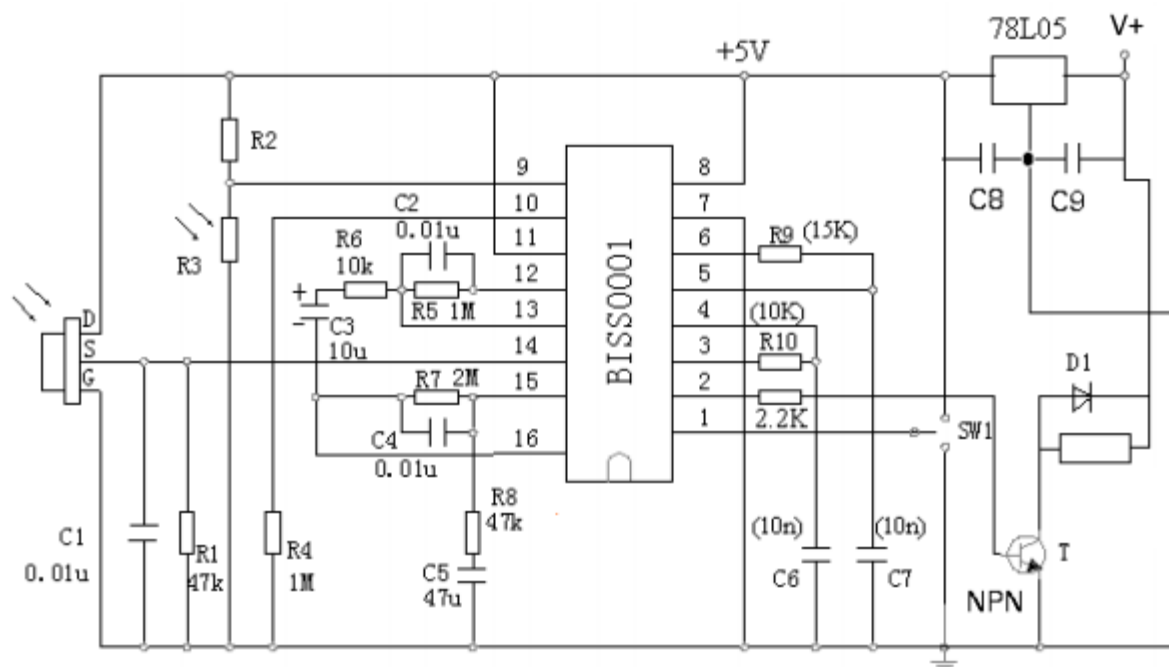
2.8 Αισθητήρας ανίχνευσης κίνησης PIR

Για την ανίχνευση κίνησης στους χώρους της κατοικίας επιλέχθηκε και χρησιμοποιήθηκε ο παθητικός αισθητήρας υπέρυθρων PIR (Passive Infra-Red) Sensor, της εταιρείας Parallax. Πρόκειται για μία πυροηλεκτρική συσκευή, η οποία εντοπίζει την κίνηση αντιλαμβανόμενη αλλαγές των επιπέδων της υπέρυθρης ακτινοβολίας στα περιβάλλοντα αντικείμενα. Ο αισθητήρας ελέγχει για ξαφνικές αλλαγές σε ένα υπέρυθρο (IR) πρότυπο περιβάλλον. Όταν εντοπίσει κίνηση, ο PIR Sensor βγάζει στην έξοδό του ένα σήμα υψηλής τάσης (HIGH), το οποίο στη συνέχεια μπορεί να γίνει είσοδος σε ένα μικροελεγκτή ή να τροφοδοτήσει ένα εξωτερικό φορτίο. (Parallax, 2014)



Σχήμα 2-26. Ο αισθητήρας ανίχνευσης κίνησης PIR

Ο αισθητήρας PIR μπορεί να ανιχνεύσει ένα άτομο σε απόσταση έως 9 μέτρων ή σε απόσταση έως 4,5 μέτρων αν η ευαισθησία είναι ρυθμισμένη στο κατώτατο σημείο. Η ψηφιακή έξοδος του επηρεάζεται -εκτός από τη ρύθμιση ευαισθησίας- από το μέγεθος και τις θερμικές ιδιότητες των αντικειμένων που βρίσκονται κοντά στον αισθητήρα, καθώς επίσης και από τις περιβαλλοντικές συνθήκες, συμπεριλαμβανομένων της θερμοκρασίας του περιβάλλοντος και τις πηγές φωτός.

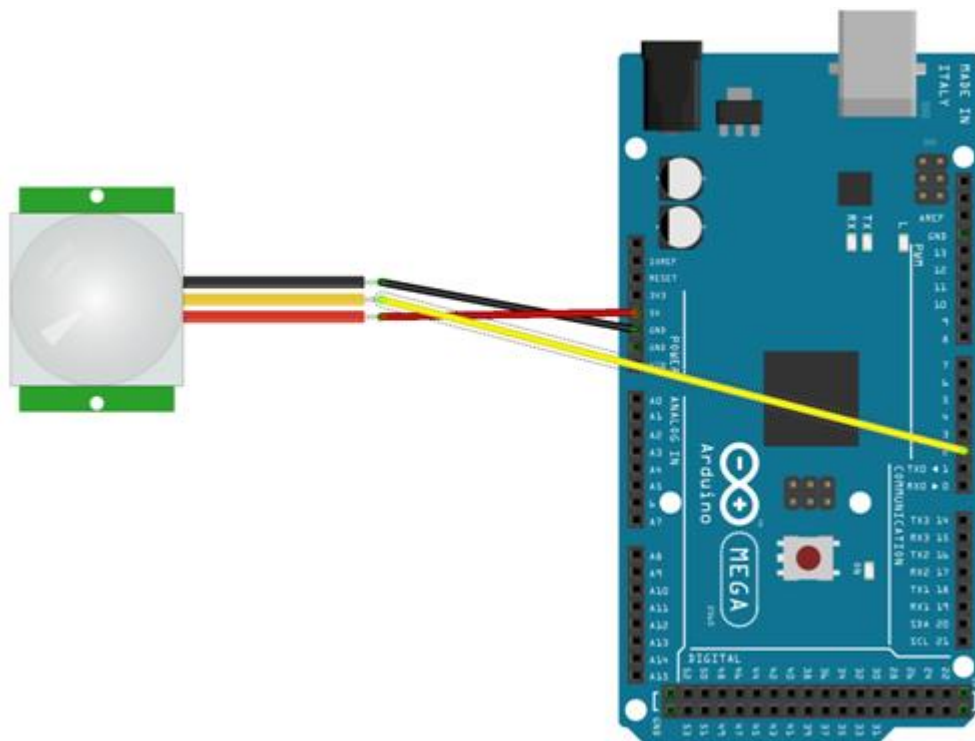


Σχήμα 2-27. Σχηματικό διάγραμμα αισθητήρα PIR

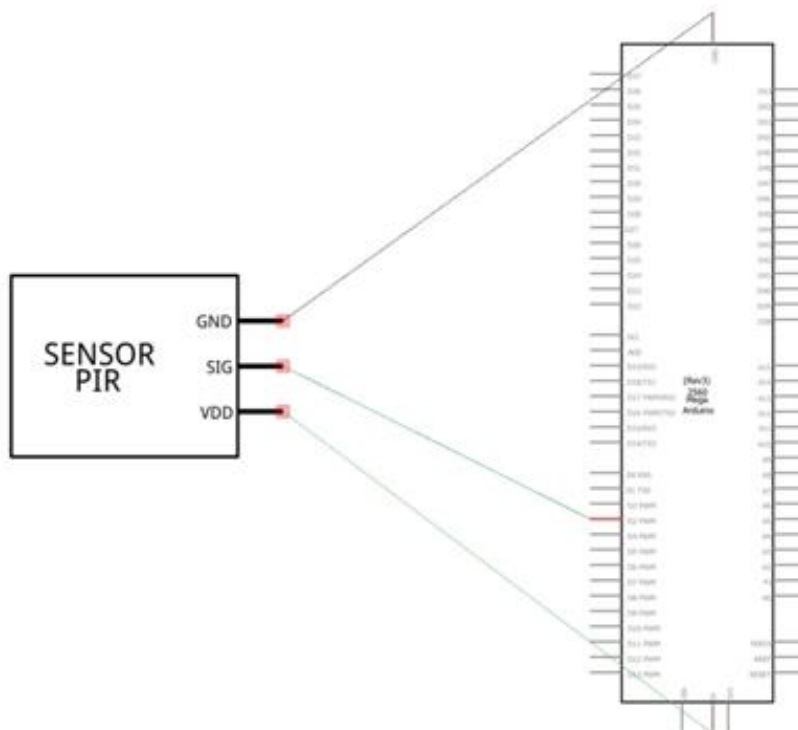
Για τη σωστή λειτουργία του, ο αισθητήρας PIR απαιτεί ένα μικρό χρονικό διάστημα (έως 40 δευτερόλεπτα) προετοιμασίας, εντός του οποίου αναγνωρίζει το περιβάλλον του. Κατά τη διάρκεια αυτής της προετοιμασίας, τα δύο LEDs που βρίσκονται κάτω από το πλαστικό κάλυμμα είναι αναμμένα και πρέπει το πεδίο θέασης του αισθητήρα να παραμένει όσο το δυνατόν πιο ακίνητο.

2.8.1 Συνδεσμολογία PIR Sensor με Arduino

Ο αισθητήρας μπορεί να τροφοδοτηθεί από 3-6 V, αντλεί 130μΑ όταν δεν εντοπίζει κίνηση και 3mA (χωρίς φορτίο) όταν εντοπίζει κάποια αλλαγή στο πρότυπο περιβάλλον. Για την τροφοδοσία του χρησιμοποιήθηκε η έξοδος 5V του Arduino. Η ψηφιακή έξοδος του αισθητήρα συνδέθηκε στην ψηφιακή είσοδο D2 του Arduino και το pin GND του αισθητήρα στο GND του Arduino. Επίσης, χρησιμοποιήθηκε το onboard LED 13 του Arduino ως έξοδος, το οποίο ενεργοποιείται κάθε φορά που ο αισθητήρας εντοπίζει κίνηση και μένει ανοιχτό μέχρι η έξοδος του αισθητήρα να γίνει ξανά LOW.



Σχήμα 2-28. Συνδεσμολογία PIR με Arduino



Σχήμα 2-29. Σχηματικό συνδεσμολογίας PIR με Arduino

2.8.2 Κώδικας Arduino για χρήση αισθητήρα PIR

```
int ledPin = 13;           // to onboard led
int inputPin = 2;          // to pin simatos
int pirState = LOW;        // arxiki katastasi
int val = 0;               // status

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(inputPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  val = digitalRead(inputPin);
  if (val == HIGH) {
    digitalWrite(ledPin, HIGH);
    if (pirState == LOW) {
      // ενεργοποιήθηκε
      Serial.println("Ανιχνεύτηκε κίνηση");

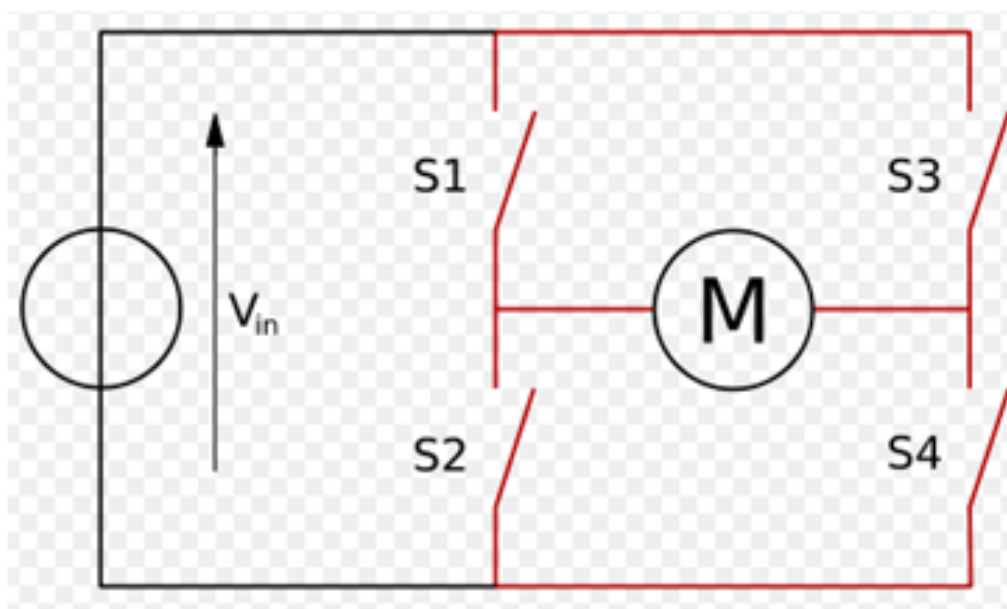
      pirState = HIGH;
    }
  } else {
    digitalWrite(ledPin, LOW);
    if (pirState == HIGH) {
      Serial.println("Σταμάτησε η κίνηση");
      pirState = LOW;
    }
  }
}
```

2.9 Κύκλωμα οδήγησης κινητήρων

Για τον έλεγχο των παραθύρων της έξυπνης κατοικίας στην παρούσα εργασία χρησιμοποιήθηκαν DC (Direct Current) κινητήρες τάσης λειτουργίας 6-9 V, κάθε ένας από τους οποίους αντλεί έως 400mA, σε συνδυασμό με 4 L293D ολοκληρωμένα κυκλώματα οδήγησης κινητήρων, κάθε ένα από τα οποία περιέχει 2 H-bridges.

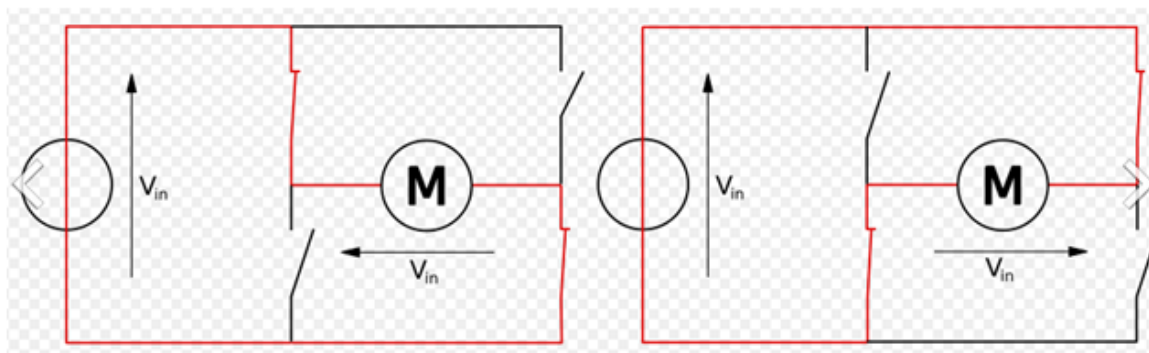
2.9.1 Λειτουργία γέφυρας H (H-Bridge)

Μία γέφυρα H (H-bridge) είναι ένα ηλεκτρονικό κύκλωμα που επιτρέπει σε μία τάση να εφαρμοστεί σε ένα φορτίο και προς τις δύο κατευθύνσεις. Χρησιμοποιείται συνήθως σε ρομποτικά συστήματα για να επιτρέπει στους DC κινητήρες να κινηθούν προς τα μπροστά και προς τα πίσω.



Σχήμα 2-30. Κυκλωματικό διάγραμμα μίας H-bridge (επισημαίνεται με κόκκινο χρώμα)

Όταν οι διακόπτες S1 και S4 είναι κλειστοί (και οι S2 και S3 ανοιχτοί), μία θετική τάση εφαρμόζεται στα άκρα του κινητήρα. Ανοίγοντας τους διακόπτες S1 και S4 και κλείνοντας τους διακόπτες S2 και S3, η τάση αυτή αντιστρέφεται, επιτρέποντας στον κινητήρα να λειτουργήσει με την αντίθετη φορά.



Σχήμα 2-31. Οι δύο βασικές καταστάσεις μίας H-bridge

Οι διακόπτες S1 και S2 δεν πρέπει ποτέ να κλείνουν ταυτόχρονα, καθώς αυτό προκαλεί βραχυκύκλωμα στη πηγή τάσης εισόδου. Το ίδιο ισχύει και για τους διακόπτες S3 και S4. Ο κινητήρας σταματάει να λειτουργεί όταν οι διακόπτες S1 και S3 είναι κλειστοί (και οι S2 και S4 ανοιχτοί) ή όταν είναι κλειστοί οι διακόπτες S2 και S4 (και οι S1 και S3 ανοιχτοί).

2.9.2 Η λειτουργία του κυκλώματος οδήγησης (Motor Driver)

Κύκλωμα οδήγησης (Motor driver) ονομάζεται ένας ενισχυτής ρεύματος, ο οποίος λαμβάνει ένα σήμα χαμηλού ρεύματος από ένα μικροελεγκτή και παράγει ένα αναλογικά μεγαλύτερο σήμα ρεύματος, το οποίο μπορεί να ελέγξει έναν κινητήρα. Ο Motor Driver L293D είναι ένα κύκλωμα κατάλληλο να ελέγξει δύο μικρούς κινητήρες, κάνοντάς τους να κινηθούν και προς τις δύο κατευθύνσεις. Επιλέγεται σε σχέση με άλλα κυκλώματα οδήγησης κινητήρων καθώς περιλαμβάνει διόδους οι οποίες προστατεύουν τα τρανζίστορς, που είναι υπεύθυνα για τον έλεγχο των κινητήρων, από αιχμές της τάσης, που προκαλούνται όταν το πηνίο του κινητήρα σταματά να διαρρέεται από ρεύμα.

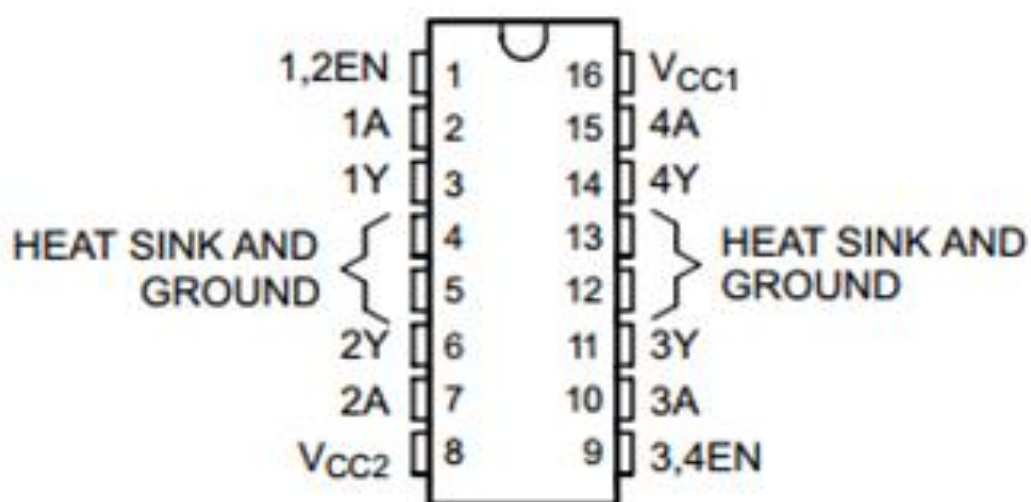
2.9.3 Η λειτουργία του ολοκληρωμένου κυκλώματος L293

Το L293D είναι ένα μονολιθικό ολοκληρωμένο κύκλωμα (IC - Integrated Circuit) υψηλής τάσης εισόδου και ρεύματος εξόδου, που διαθέτει 4 κανάλια οδήγησης. Είναι σχεδιασμένο να δέχεται DTL (Diode-Transistor Logic) ή TTL (Transistor-Transistor Logic) λογική, να οδηγεί επαγωγικά φορτία, όπως ρελέ, DC και βηματικούς κινητήρες, και να τροφοδοτεί τρανζίστορς ισχύος.



Σχήμα 2-32. Το ολοκληρωμένο κύκλωμα L293D

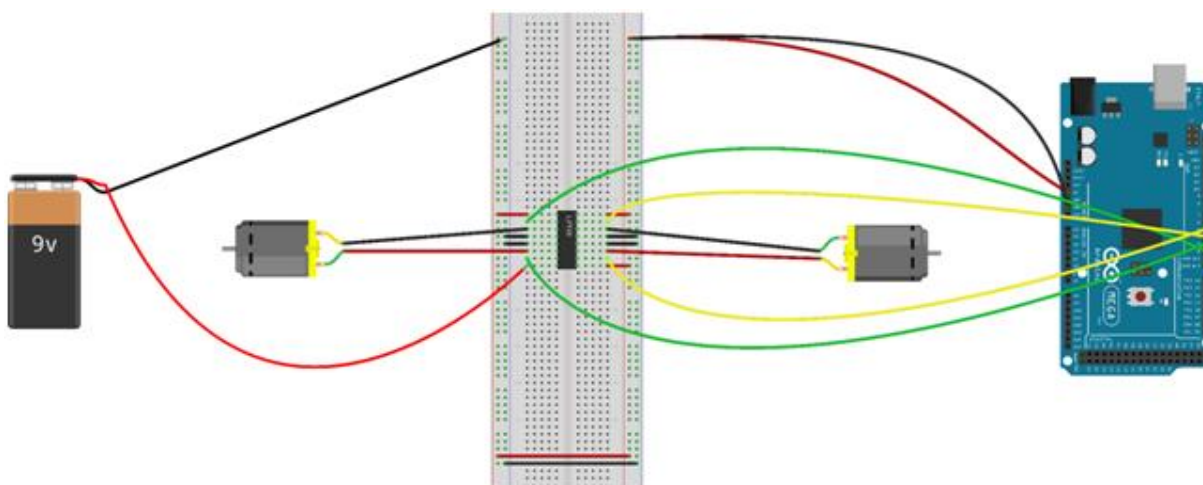
Κάθε ζευγάρι καναλιών είναι εξοπλισμένο με μία είσοδο ενεργοποίησης, ώστε να απλοποιείται η χρησιμοποίησή τους ως δύο H-bridges. Μία ξεχωριστή είσοδος τροφοδοσίας παρέχεται για τη λογική, επιτρέποντας τη λειτουργία σε χαμηλότερη τάση. Το L293D μπορεί να ελέγξει μέχρι 2 κινητήρες με απαιτήσεις τάσης 4,5 – 36 V, τα οποία αντλούν έως 600mA. Η θερμοκρασία λειτουργίας του L293D είναι 0-70°C.



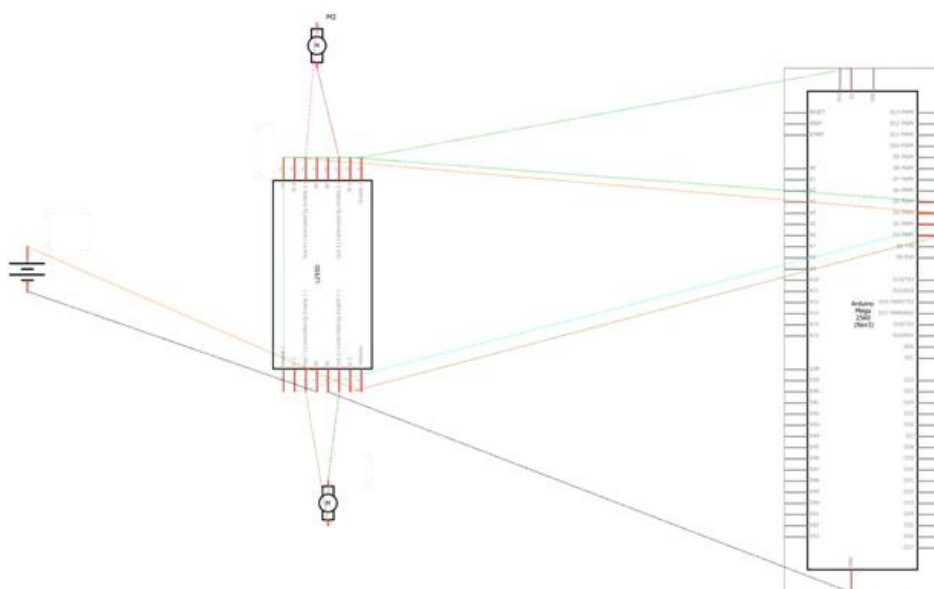
Σχήμα 2-33. Σχηματικό διάγραμμα του ολοκληρωμένου κυκλώματος L293D

2.9.4 Συνδεσμολογία L293D με Arduino για έλεγχο DC κινητήρων

Οι είσοδοι ενεργοποίησης του ολοκληρωμένου L293D (1,2EN / 3,4EN), καθώς και η ξεχωριστή είσοδος τροφοδοσίας για τη λογική (V_{CC1}) συνδέθηκαν στην έξοδο 5 V του Arduino. Οι κινητήρες συνδέθηκαν στις εξόδους του κυκλώματος οδήγησης (ο ένας στα pins 3 και 6 και ο δεύτερος στα pins 11 και 14). Για τον έλεγχο των κινητήρων χρησιμοποιήθηκαν τα pins εισόδου 2,7 και 10,15, τα οποία συνδέθηκαν στις ψηφιακές εξόδους 2,3 και 4,5 αντίστοιχα. Η είσοδος V_{CC2} του L293D συνδέθηκε με μπαταρία 9V, η οποία είναι υπεύθυνη για την τροφοδοσία των κινητήρων. Το V_{CC2} δύναται να δεχθεί από 4,5 V έως 36 V. Τα pins 4,5 και 12,13 του ολοκληρωμένου L293D, καθώς και ο αρνητικός πόλος της μπαταρίας, συνδέθηκαν στο GND του Arduino.



Σχήμα 2-34. Συνδεσμολογία L293 με Arduino για έλεγχο δύο DC κινητήρων



Σχήμα 2-35. Σχηματικό συνδεσμολογίας L293 με Arduino για έλεγχο δύο DC κινητήρων

2.9.5 Κώδικας Arduino για οδήγηση δύο κινητήρων

```
int motor_left[] = {2, 3};
int motor_right[] = {4, 5};

void setup() {
  Serial.begin(9600);
  for(int i = 0; i < 2; i++){
    pinMode(motor_left[i], OUTPUT);
    pinMode(motor_right[i], OUTPUT);}
}

void loop() {
  drive_forward();
  delay(2000);
  motor_stop();
  drive_backward();
  delay(2000);
  motor_stop();
}

void motor_stop(){
  digitalWrite(motor_left[0], LOW);
  digitalWrite(motor_left[1], LOW);
  digitalWrite(motor_right[0], LOW);
  digitalWrite(motor_right[1], LOW);
  delay(25);
}

void drive_forward(){
  digitalWrite(motor_left[0], HIGH);
  digitalWrite(motor_left[1], LOW);
  digitalWrite(motor_right[0], HIGH);
  digitalWrite(motor_right[1], LOW);
}

void drive_backward(){
  digitalWrite(motor_left[0], LOW);
  digitalWrite(motor_left[1], HIGH);
  digitalWrite(motor_right[0], LOW);
  digitalWrite(motor_right[1], HIGH);
}
```

2.10 Τεχνολογίες δικτύου

Για την σύνδεση των βασισμένων στο Arduino σταθμών βάσης, με το εσωτερικό δίκτυο της ευφυούς κατοικίας -και εν δυνάμει και με το διαδίκτυο- απαιτείται η χρήση ενός Modem/Router. Στις προδιαγραφές που απαιτούνται για την προτεινόμενη υλοποίηση περιλαμβάνονται το να είναι χαμηλού κόστους, να διαθέτει τουλάχιστον 2 θύρες Ethernet στην λειτουργία ως Switch για τους δύο σταθμούς βάσης, και να διαθέτει WiFi ώστε να συνδέει το ενσύρματο δίκτυο με ασύρματες συσκευές (κινητά τηλέφωνα, laptops). Τα περισσότερα τυπικά Modem/Routers που δίνονται από τους παρόχους υπηρεσιών διαδικτύου πληρούν τις άνωθεν προϋποθέσεις. Στην προτεινόμενη υλοποίηση χρησιμοποιήθηκε το Philips CGA5720S/TE.

Στο εσωτερικό δίκτυο η IP διεύθυνση του εν λόγω Router, ορίστηκε η 192.168.1.1, με τις υπόλοιπες συσκευές που συνδέονται σε αυτό να δύνανται να λάβουν διευθύνσεις από την 192.168.1.2 έως 192.168.1.254. Για τους σταθμούς βάσεις αισθητήρων και ενεργοποιητών επιλέχθηκαν οι στατικές εσωτερικές διευθύνσεις 192.168.1.21 και 192.168.1.22 αντίστοιχα. Λοιπές συσκευές που συνδέονται στο εν λόγω δίκτυο λαμβάνουν IPs εντός του συγκεκριμένου εύρους μέσω του δυναμικού μηχανισμού παραχώρησης IP διευθύνσεων (DHCP - Dynamic Host Configuration Protocol).

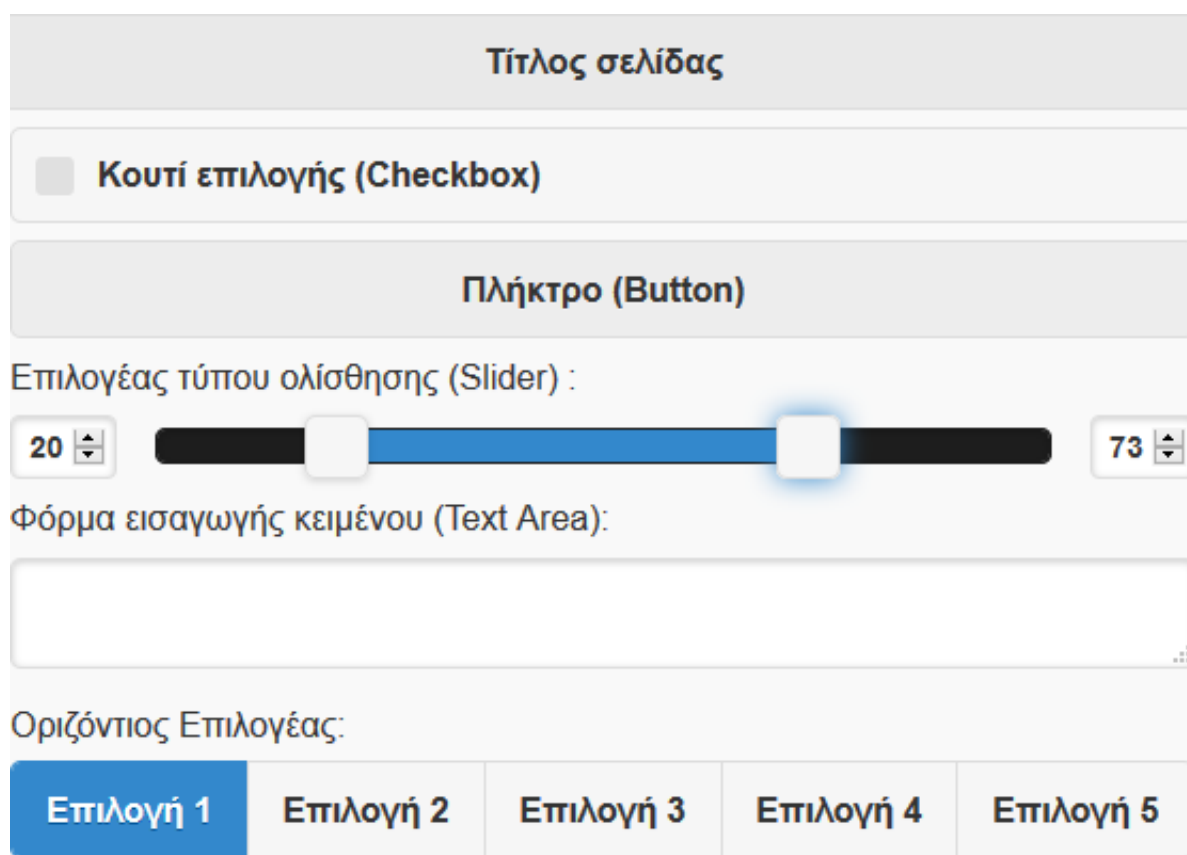
Σε περίπτωση που επιθυμούμε τα δεδομένα των σταθμών βάσης να είναι προσβάσιμα από οποιοδήποτε σημείο του κόσμου μέσω Internet, και όχι μόνο εντός του τοπικού δικτύου εντός της εμβέλειας του WiFi και των καλωδίων Ethernet, απαιτείται η χρήση του Port Forwarding. Με τον τρόπο αυτό την εξωτερική IP διεύθυνση που μας δίνεται από τον πάροχο, την αντιστοιχούμε σε IP διεύθυνση εντός του εσωτερικού του δικτύου. Στο παράδειγμα που δίνεται παρακάτω, η [εξωτερική IP]:80, αντιστοιχεί στην 192.168.1.21:80, του σταθμού βάσης αισθητήρων. Οι επιπτώσεις ασφαλείας της πρόσβασης μέσω διαδικτύου στο εσωτερικό δίκτυο της ευφυούς οικίας, δεν εξετάστηκαν στην παρούσα εργασία.

No.	Lan IP Address	Protocol Type	Lan Port	Public Port	Enable	
0	192.168.1.21	TCP	80	80	<input checked="" type="checkbox"/>	Add/Clean

Σχήμα 2-36. Port forwarding σε Router Philips CGA5720S/TE

2.11 Τεχνολογίες υλοποίηση διεπαφής χρήστη (User Interface)

Για την υλοποίηση της διεπαφής χρήστη κρίθηκε απαραίτητη η χρήση του jQuery Mobile: Ένός συνόλου βιβλιοθηκών ανοιχτού κώδικα, βασισμένες στο πρότυπο HTML 5 (HyperText Markup Language), οι οποίες επιτρέπουν την κατασκευή και παραμετροποίηση του U.I. (User Interface) εφαρμογών. Αποτελεί ανεξάρτητο πλατφόρμας λογισμικό (cross platform), συνεπώς είναι συμβατό με όλα τα σύγχρονα λειτουργικά συστήματα ηλεκτρονικών υπολογιστών (Windows, Linux) και κινητών συσκευών (Android, iOS, Windows Phone). Μεταξύ των δυνατοτήτων του συγκαταλέγονται η αυτόματη προσαρμογή σε οθόνες διαφορετικού μεγέθους και ανάλυσης (Responsive Design), η υποστήριξη χειρονομιών σε οθόνες αφής (Gestures) τα εφέ μετάβασης μεταξύ οθονών (Page Transitions), αλλά και τα προ-προγραμματισμένα γραφικά στοιχεία (Widgets). Η πρόσβαση στα γραφικά στοιχεία αλλά και ο προγραμματισμός της λειτουργίας τους πραγματοποιείται μέσω του παρεχόμενου API, το οποίο επίσης μπορούμε να επεκτείνουμε. (Firtman, 2012). Η άδεια με την οποία παρέχεται το jQuery Mobile είναι η ελεύθερη MIT (Massachusetts Institute of Technology) License, η οποία επιτρέπει την χρήση του και στην παρούσα εργασία, αλλά και σε τυχόν εμπορική χρήση.



Σχήμα 2-37. Γραφικά στοιχεία (Widgets) του jQuery Mobile

2.11.1 Δομή σελίδας εφαρμογής με χρήση του jQuery Mobile

Η χρήση του jQuery Mobile σε μία εφαρμογή απαιτεί την ενσωμάτωση των τριών βιβλιοθηκών του (jquery.mobile-[έκδοση].min.css, jquery-[έκδοση].min.js, jquery.mobile-[έκδοση].min.js). Η πρώτη βιβλιοθήκη έχει κατάληξη .css και περιέχει γραφικά στοιχεία και πληροφορίες εμφάνισης, ενώ οι δύο βιβλιοθήκες με κατάληξη .js περιέχουν κώδικα javascript ο οποίος επιδρά στα γραφικά στοιχεία του .css. (Dutson, 2013) Παρακάτω δίνεται ο κώδικας της βασικής δομής μίας σελίδας HTML με ενσωματωμένες τις βιβλιοθήκες του jQuery Mobile, ο οποίος αποτέλεσε και την βάση για την προτεινόμενη εφαρμογή που υλοποιήθηκε.

```
<!DOCTYPE html>
<html>

<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="jquery.mobile-[version].min.css" />
    <script src="jquery-[version].min.js"></script>
    <script src="jquery.mobile-[version].min.js"></script>
</head>

<body>

<div data-role="page"> <!-- αρχή εμφανιζόμενης σελίδας -->

    <div data-role="header">
        <h1>Τίτλος σελίδας</h1>
    </div> <!-- Τίτλος -->

<!-- Τμήμα ενσωμάτωσης περιεχομένου -->

</div> <!-- τέλος εμφανιζόμενης σελίδας -->

</body>

</html>
```

Τα στοιχεία <meta> τοποθετούνται στο τμήμα <head> </head> ενός αρχείου HTML για να παρέχουν ορισμένες πληροφορίες στον μηχανισμό (φυλλομετρητή ή WebView) που τα εμφανίζει. Ο ορισμός του charset ως UTF-8 αφορά την κωδικοποίηση και απαιτείται στην προτεινόμενη εφαρμογή για την σωστή εμφάνιση των Ελληνικών χαρακτήρων. Ο ορισμός του πλάτους του περιεχομένου ως device-width, initial-scale=1 αφορά το αρχικό επίπεδο μεγέθυνσης. Το εμφανιζόμενο στον χρήστη UI αρχίζει στον ορισμό της σελίδας ως <div data-role="page">. Ο τίτλος της κάθε σελίδας της εφαρμογής εμφανίζεται μέσα σε <h1> </h1> tags και δηλώνεται ως <div data-role="header">.

2.11.2 Χρήση λίστας στοιχείων (Listview) σε jQuery Mobile

Στην κατασκευή του UI της προτεινόμενης εφαρμογής κρίνεται απαραίτητη η χρήση λιστών, τόσο για δυνατότητα του χρήστη να επιλέξει το δωμάτιο το οποίο θέλει να ελέγξει, όσο και για την εμφάνιση των δεδομένων των αισθητήρων. Για την κατηγοριοποίηση των στοιχείων των λιστών χρησιμοποιούνται διαχωριστές. Ο χρήστης επίσης γνωρίζει εάν κάποια επιλογή της λίστας παραπέμπει σε υπομενού, μέσω του εικονιδίου στα δεξιά.

Τίτλος σελίδας	
Στοιχείο 1	>
Στοιχείο 2	>
Στοιχείο 3	>
Διαχωριστής:	
Στοιχείο 4	>
Διαχωριστής:	
Στοιχείο 5	

Σχήμα 2-38. Λίστα στοιχείων (Listview) σε jQuery Mobile

Παρακάτω δίνεται ο κώδικας δημιουργίας της άνωθεν λίστας στοιχείων / δεδομένων σε jQuery Mobile.

```
<div data-role="main" class="ui-content">    <!--αρχή περιεχομένου-->

    <ul data-role="listview">    <!--αρχή λίστας-->
        <li><a href="#">Στοιχείο 1</a></li>
        <li><a href="#">Στοιχείο 2</a></li>
        <li><a href="#">Στοιχείο 3</a></li>

        <li data-role="list-divider">Διαχωριστής:</li>
        <!--διαχωρισμός στοιχείων-->
        <li><a href="#">Στοιχείο 4</a></li>
        <!--διαχωρισμός στοιχείων-->
        <li data-role="list-divider">Διαχωριστής:</li>
        <!--διαχωρισμός στοιχείων-->
        <li data-icon="false"><b>Στοιχείο 5</b></li>
    </ul>                                <!--τέλος λίστας-->

</div>                                <!--τέλος περιεχομένου-->
```

Η δημιουργία μίας λίστας στοιχείων, απαιτεί την χρήση του `<ul data-role="listview">` tag το οποίο συνεπάγεται μη ταξινομημένη αλφαβητικά λίστα (unordered list). Κάθε στοιχείο της λίστας ορίζεται μέσα στα `` `` tags. Σε περίπτωση που είναι επιθυμητό κάποιο στοιχείο της λίστας να παραπέμπει σε κάποια άλλη σελίδα (π.χ. κάποιο υπομενού στην εφαρμογή) χρησιμοποιείται μέσα σε αυτό το ``. Σε περίπτωση που ένα στοιχείο της λίστας δεν παραπέμπει σε άλλη σελίδα, κρίνεται σκόπιμη η χρήση του `data-icon="false"` για την αφαίρεση του εικονιδίου μετάβασης (βλ. Στοιχείο 5 σε Σχήμα 2-38). Για την κατηγοριοποίηση των δεδομένων χρησιμοποιούνται διαχωριστές (list dividers), οι οποίοι τοποθετούνται ως στοιχεία της λίστας με `data-role="list-divider"`. Το τέλος της λίστας ορίζεται με το κλείσιμο του `ul` tag (``).

2.11.3 Χρήση επιλογέων (Buttons) σε jQuery Mobile

Για τον έλεγχο παραμέτρων αλλά και για την ενεργοποίηση / απενεργοποίηση συσκευών μέσω της προτεινόμενης εφαρμογής, κρίνεται απαραίτητη η τοποθέτηση επιλογέων (Buttons) στην διεπαφή χρήστη. Οι επιλογείς δύναται να καταλαμβάνουν το πλήρες πλάτος της οθόνης του χρήστη, είτε μέρος αυτής εάν διαχωριστούν.



Σχήμα 2-39. Επιλογείς (Buttons) σε jQuery Mobile

Παρακάτω δίνεται ο κώδικας δημιουργίας της παραπάνω οθόνης με ενεργοποιητές σε jQuery Mobile.

```
<div data-role="main" class="ui-content">    <!--αρχή περιεχομένου-->

<fieldset class="ui-grid-a">
    <div class="ui-block-a"><button class="ui-btn ui-btn-a">
Ενεργοποίησε Α </button></div>
    <div class="ui-block-b"><button class="ui-btn ui-btn-b">
Απενεργοποίησε Α </button></div>
    <div class="ui-block-a"><button class="ui-btn ui-btn-a">
Ενεργοποίησε Β </button></div>
    <div class="ui-block-b"><button class="ui-btn ui-btn-b">
Απενεργοποίησε Β </button></div>
</fieldset>

    <button class="ui-btn ui-btn-a">Ενεργοποιητής Γ</button>
    <button class="ui-btn ui-btn-b">Ενεργοποιητής Δ</button>

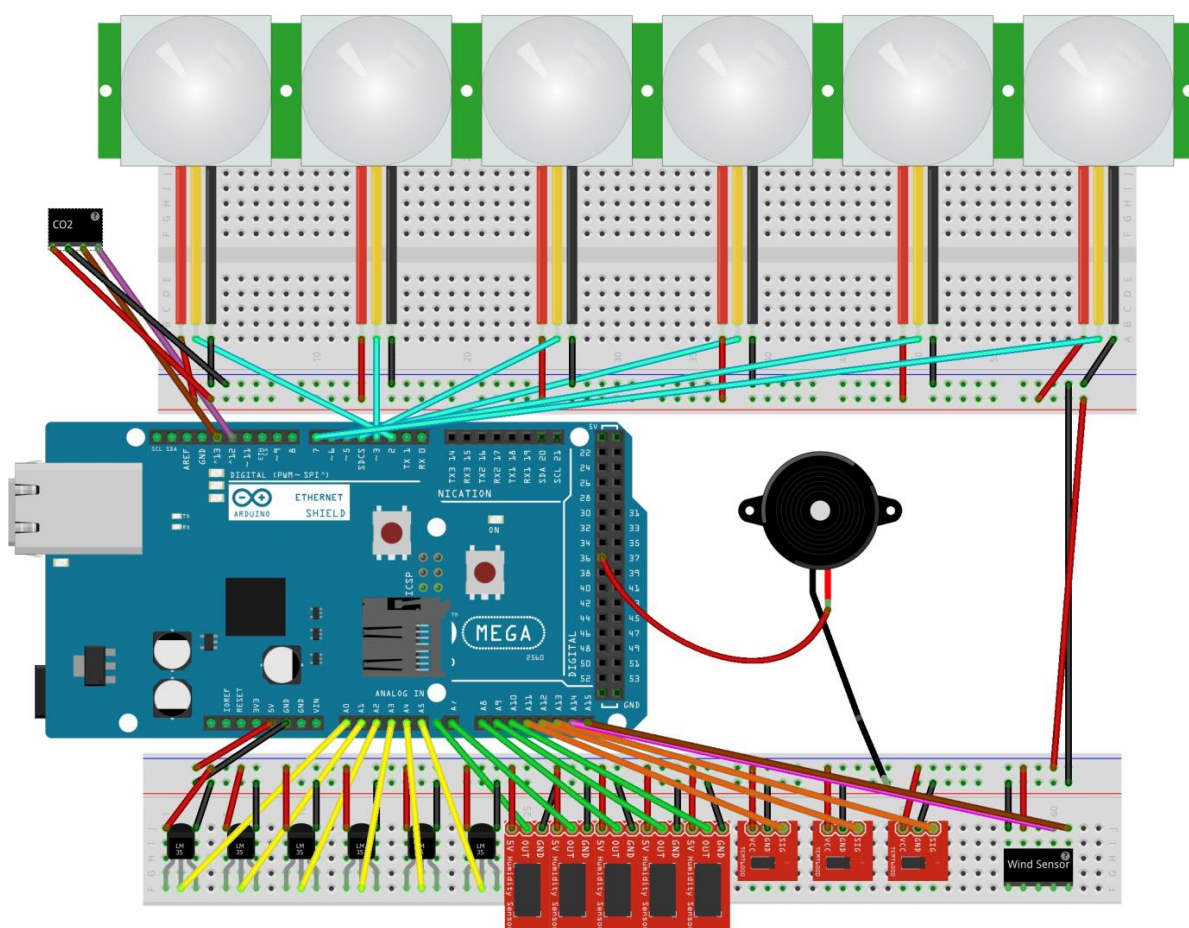
</div>    <!--τέλος περιεχομένου-->
```

Κάθε ενεργοποιητής ορίζεται με τα <button> </button> tags. Εάν είναι επιθυμητός διαχωρισμός του πλάτους της οθόνης, χρησιμοποιούνται οι κλάσεις ui-block-a και ui-block-b, μέσα σε κάθε μία εκ των οποίων τοποθετούνται τα button tags. Η αλλαγή του χρώματος εμφάνισης των ενεργοποιητών πραγματοποιείται μέσω της κλασης ui-btn-a και ui-btn-b.

3. Πρακτική υλοποίηση πλατφόρμας

3.1 Υλοποίηση σταθμού βάσης αισθητήρων

Ο πρώτος σταθμός βάσης υλοποιήθηκε με χρήση του μικροελεγκτή ATmega2560 του Arduino Mega και του Ethernet ελεγκτή (Controller) Wiznet 5100 της Ethernet Shield, τα οποία περιγράφησαν στα προηγούμενα κεφάλαια. Σε αυτόν συνδέονται όλοι οι αισθητήρες (έξι αισθητήρες θερμοκρασίας, πέντε αισθητήρες υγρασίας, τρεις αισθητήρες φωτεινότητας, ένας αισθητήρας διοξειδίου του άνθρακα, έξι αισθητήρες κίνησης, και ένας αισθητήρας ταχύτητας ανέμου), όπως και το ηχείο (buzzer) το οποίο χρησιμοποιείται για ηχητική ειδοποίηση στην λειτουργία συναγερμού, σε περίπτωση εντοπισμού κίνησης από αισθητήρα PIR. Η ενδεικτική συνδεσμολογία, ανάγνωση και κανονικοποίηση δεδομένων από κάθε μεμονωμένο αισθητήρα που χρησιμοποιήθηκε έχει περιγραφεί στο Κεφάλαιο 2. Συνολικά η συνδεσμολογία όλων των αισθητήρων στον σταθμό βάσης παρίσταται στην παρακάτω εικόνα.



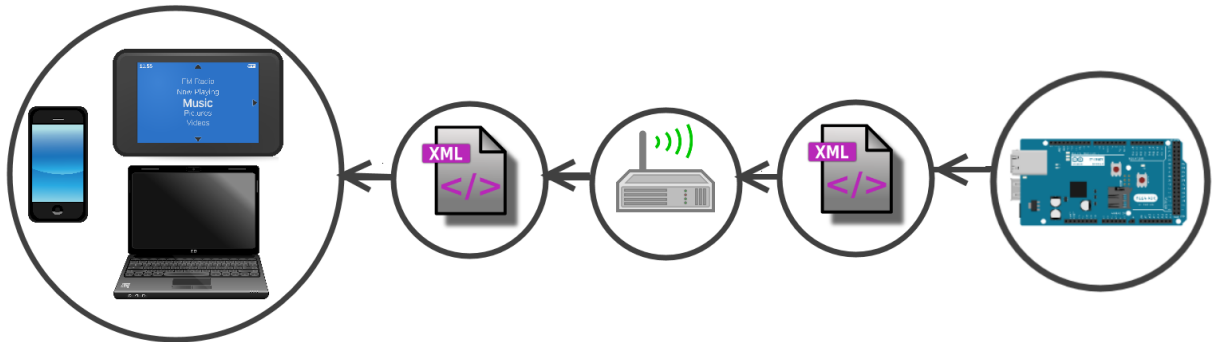
Σχήμα 3-1. Παράσταση συνδεσμολογίας συνόλου αισθητήρων και buzzer σε σταθμό βάσης

Στα αναλογικά Pins A0 έως A5 συνδέονται οι έξοδοι των έξι αισθητήρων θερμοκρασίας LM35. Στα Pins A6 έως A10 συνδέονται οι έξοδοι των τεσσάρων αισθητήρων υγρασίας HIH4030. Στα Pins A11 έως A13 συνδέονται οι έξοδοι των τριών αισθητήρων φωτός TEMT6000. Στα Pins A14 και A15 συνδέονται οι έξοδοι TMP και RV αντίστοιχα, του ενός αισθητήρα ταχύτητας ανέμου Wind Sensor Rev. C. Τα ψηφιακά Pins 0 και 1 αφέθηκαν σκοπίμως αχρησιμοποίητα, καθότι αντιστοιχούν στο TX και RX της σειριακής θύρας, η οποία χρησιμοποιήθηκε κατά την διαδικασία της αποσφαλμάτωσης. Στα ψηφιακά Pins 2 έως 7 συνδέθηκαν οι έξοδοι των αισθητήρων PIR. Στα ψηφιακά Pins 12 και 13, συνδέθηκαν τα RX και TX του αισθητήρα διοξειδίου του άνθρακα K-30, ο οποίος χρησιμοποιεί το πρωτόκολλο SPI. Στο ψηφιακό Pin 36 συνδέθηκε το Buzzer συναγερμού. Τα λοιπά ψηφιακά Pins δεν χρησιμοποιήθηκαν.

Η δήλωση των Pins στον κώδικα του Arduino έγινε ως εξής:

```
const int lm35PinTrapezaria = A0;
const int lm35PinKouzina = A1;
const int lm35PinSaloni = A2;
const int lm35PinGrafeio = A3;
const int lm35PinKrevatokamara = A4;
const int lm35PinKhpos = A5;
const int hih4030PinTrapezaria = A6;
const int hih4030PinKouzina = A7;
const int hih4030PinSaloni = A8;
const int hih4030PinGrafeio = A9;
const int hih4030PinKhpos = A10;
const int temt6000PinKouzina = A11;
const int temt6000PinSaloni = A12;
const int temt6000PinKhpos = A13;
const int windSensorTempPin = A14;
const int windSensorRVPin = A15;
const int pirPinTrapezaria = 2;
const int pirPinKouzina = 3;
const int pirPinSaloni = 4;
const int pirPinGrafeio = 5;
const int pirPinKrevatokamara = 6;
const int pirPinKhpos = 7;
const int co2rxPin = 12;
const int co2txPin = 13;
const int buzzerPin = 36;
```

Για να είναι προσβάσιμες οι άνωθεν μετρήσεις αισθητήρων από εφαρμογές σε υπολογιστές και έξυπνα κινητά τηλέφωνα, κρίθηκε απαραίτητη η υλοποίηση κώδικα Web Server στο Arduino με την Ethernet Shield, ο οποίος μέσω ενός API που επινοήθηκε θα παρέχει τα δεδομένα σε μορφή XML. (Trevennor, 2014) Στην συνέχεια, με την χρήση αυτού του API και κάνοντας parse τα δεδομένα του XML, θα είναι εφικτή η εμφάνιση των πληροφοριών και δεδομένων αισθητήρων στην προτεινόμενη υλοποιηθείσα εφαρμογή.



Σχήμα 3-2. Αναπαράσταση δημιουργίας αρχείου XML και αποστολή του σε έξυπνες συσκευές μέσω του Router

Ο βασικός προτεινόμενος κώδικας δημιουργίας Web Server που να παράγει ως έξοδο ένα στατικό XML, δίνεται παρακάτω.

```
#include <SPI.h>
#include <Ethernet.h> //ethernet i ethernet2 analogws wiznet w5100 i w5500
                        //se ethernet shield

byte mac[] = {0x90, 0x90, 0x90, 0x90, 0x90, 0x90 };
IPAddress ip(192, 168, 1, 21);

EthernetServer server(80);
String readString;

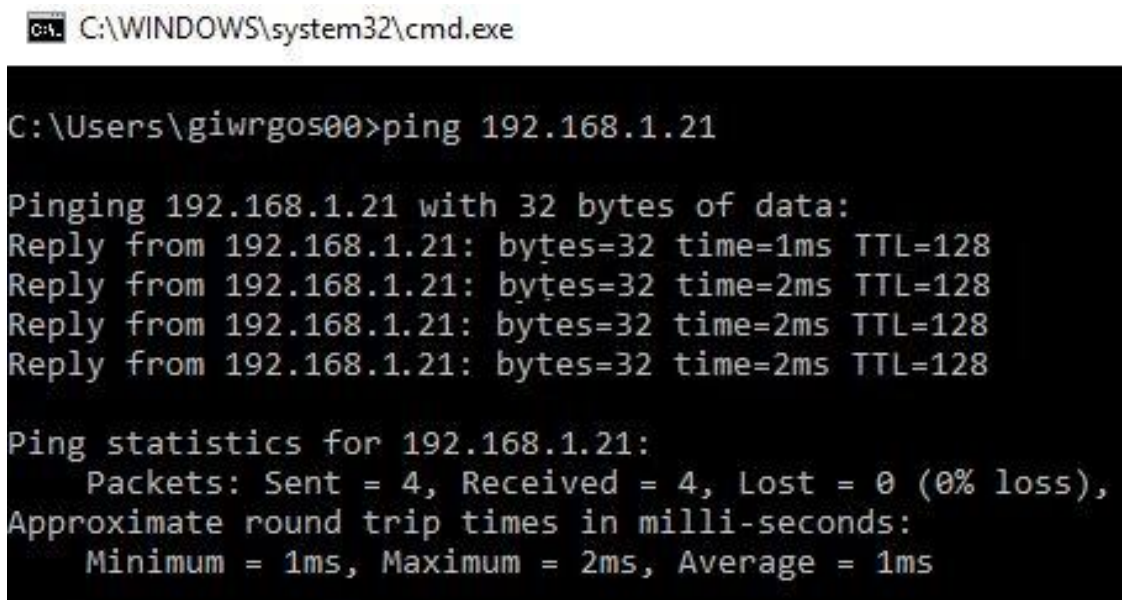
void setup()
{
  Ethernet.begin(mac, ip);
  server.begin();
  //Serial.begin(9600); //για debugging
}
```

```
void loop()
{
  EthernetClient client = server.available(); // eiserxomenoi clients
  if (client)
  {
    while (client.connected())
    {
      if (client.available())
      {
        char c = client.read();
        if (readString.length() < 100) //diavasma olou tou mikous tou req
        {
          readString += c;
          Serial.write(c);
          if (c == '\n') { //emfanisi toy HTML/XML
            client.println("HTTP/1.1 200 OK");
            client.println("Server: Arduino");
            client.println("Access-Control-Allow-Origin: *");
            client.println("Connection: close");
            client.println("Content-Type: text/xml\r\n");
            client.print("<note>"); //edw arxizei to xml arxeio mou
            client.print("<metavlititi>");
            client.print("timi metavlitis"); //edw se eisagwgika, meta oxi
            client.print("</metavlititi>");
            client.print("</note>");
            delay(1);
            //delay gia statherotita
            client.stop();
            //diagrafi gia nea ananewsi
            readString = "";
            //delay gia statherotita
            delay(1);
            client.stop();
          }
        }
      }
    }
  }
}
```

Στο άνωθεν κομμάτι κώδικα, αρχικά δηλώνονται οι βιβλιοθήκες των οποίων θα γίνει χρήση (SPI.h και Ethernet.h), στην συνέχεια η MAC Address και η IP διεύθυνση που θα αντιστοιχηθούν στον Ethernet Controller του Arduino, τα οποία απαιτείται να είναι μοναδικά στο δίκτυο. Εντός της void setup, εκτελείται η Ethernet.begin() με παραμέτρους τα δηλωθέντα Mac Address και IP, για εκκίνηση των παραμέτρων δικτύου. Εντός της void loop, της ρουτίνας κώδικα η οποία εκτελείται κατ' επανάληψη, πραγματοποιείται έλεγχος για το εάν υπάρχει εισερχόμενος client, και στην περίπτωση αυτή γίνεται ανάγνωση του Request. Τέλος, μέσω των client.print() γίνεται εμφάνιση του XML, το οποίο δηλώνεται ως εξής με το Content-Type: text/xml\r\n. Εντός του XML πρέπει να ακολουθείται η μορφή των tags ως εξής:

```
<metrisi>[τιμή μεταβλητής]</metrisi>
<metrisi2>[τιμή μεταβλητής]</metrisi2>
<allipliroforia>[τιμή μεταβλητής]</allipliroforia>
<katiallo>[τιμή μεταβλητής]</katiallo>
```

Η επιτυχής επικοινωνία με τον WebServer του Arduino δύναται να επαληθευτεί με την χρήση του εργαλείου ping στην διεύθυνση του Server (στην παρούσα εργασία αποτελεί την 192.168.1.21) ως εξής:



```
C:\WINDOWS\system32\cmd.exe

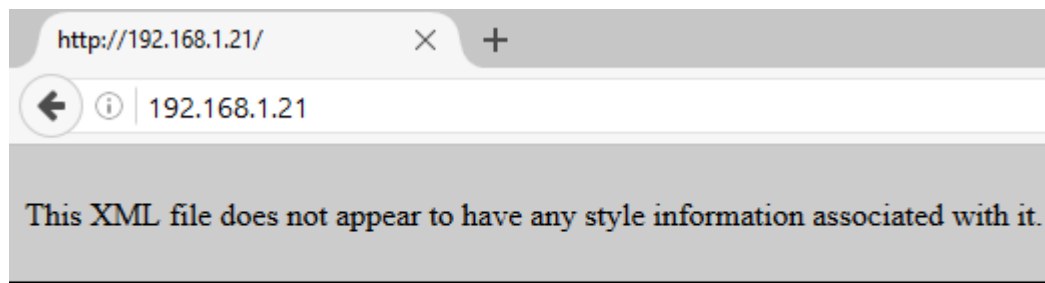
C:\Users\giwrgos00>ping 192.168.1.21

Pinging 192.168.1.21 with 32 bytes of data:
Reply from 192.168.1.21: bytes=32 time=1ms TTL=128
Reply from 192.168.1.21: bytes=32 time=2ms TTL=128
Reply from 192.168.1.21: bytes=32 time=2ms TTL=128
Reply from 192.168.1.21: bytes=32 time=2ms TTL=128

Ping statistics for 192.168.1.21:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms
```

Σχήμα 3-3. Χρήση εργαλείου ping για επαλήθευση επικοινωνίας με WebServer που τρέχει στο Arduino

Με την χρήση ενός browser δύναται να επαληθευτεί και η επιτυχής δημιουργία του XML. (Deitel, 2001). Ο άνωθεν κώδικας Arduino παρήγαγε τον WebServer με το παρακάτω XML:



```

- <note>
  <metavliti>timi metavlitis</metavliti>
</note>

```

Σχήμα 3-4. Επιβεβαίωση δημιουργίας αρχείου XML μέσω Browser

Το πλήρες υλοποιηθέν API σε XML παρουσιάζεται παρακάτω:



```

- <note>
  <temp_trapezaria>26</temp_trapezaria>
  <temp_kouzina>25</temp_kouzina>
  <temp_saloni>26</temp_saloni>
  <temp_grafeio>24</temp_grafeio>
  <temp_krevatokamara>27</temp_krevatokamara>
  <temp_khpos>23</temp_khpos>
  <hum_trapezaria>57</hum_trapezaria>
  <hum_kouzina>49</hum_kouzina>
  <hum_saloni>55</hum_saloni>
  <hum_grafeio>56</hum_grafeio>
  <hum_khpos>63</hum_khpos>
  <fws_kouzina>72</fws_kouzina>
  <fws_saloni>63</fws_saloni>
  <fws_khpos>84</fws_khpos>
  <wind>2</wind>
  <pir_trapezaria>Ησυχία</pir_trapezaria>
  <pir_kouzina>Ησυχία</pir_kouzina>
  <pir_saloni>Ησυχία</pir_saloni>
  <pir_grafeio>Ησυχία</pir_grafeio>
  <pir_krevatokamara>Ησυχία</pir_krevatokamara>
  <pir_khpos>Κλέφτης</pir_khpos>
  <co2>554</co2>
</note>

```

Σχήμα 3-5. Το πλήρες υλοποιηθέν API σε XML με τις μετρήσεις όλων των αισθητήρων

Για την δημιουργία του πλήρους XML το οποίο περιέχει όλες τις μετρήσεις από τους αισθητήρες, και όχι στατικό περιεχόμενο, απαιτείται η κλήση συναρτήσεων υπολογισμού μετρήσεων από τους αισθητήρες που περιγράφησαν στο Κεφάλαιο 2 και η εμφάνιση της επιστραφείσας μεταβλητής εντός των tags. Ενδεικτικά:

```
client.print("<temp_trapezaria>");  
int tempTrapezaria = calculateTemperature(lm35PinTrapezaria);  
client.print(tempTrapezaria);  
client.print("</temp_trapezaria>");  
client.print("<temp_kouzina>");  
int tempKouzina = calculateTemperature(lm35PinKouzina);  
client.print(tempKouzina);  
client.print("</temp_kouzina>");
```

Από την άνωθεν ρουτίνα η πρώτη γραμμή εμφανίζει το αρχικό tag <temp_trapezaria>. Στην συνέχεια, καλείται η συνάρτηση υπολογισμού της θερμοκρασίας σε βαθμούς Κελσίου, calculateTemperature(), με παράμετρο τον αριθμό του Pin της τραπεζαρίας, και το αποτέλεσμα της τοποθετείται στην μεταβλητή tempTrapezaria. Στην επόμενη γραμμή εμφανίζεται η τιμή της εν λόγω μεταβλητής. Έπειτα εμφανίζεται το tag κλεισίματος </temp_trapezaria>. Αντιστοίχως για όλα τα υπόλοιπα δωμάτια, με διαφορά την παράμετρο που περνιέται στην συνάρτηση υπολογισμού, καθώς για να υπολογιστεί και να εμφανιστεί η θερμοκρασία στην Κουζίνα απαιτείται να περαστεί στην συνάρτηση calculateTemperature() ο αριθμός του Pin του LM35 που βρίσκεται στην Κουζίνα. Όσον αφορά τον αισθητήρα υγρασίας, όπως αναφέρθηκε στο Κεφάλαιο 2.4.3, προτείνεται η χρήση του τύπου διόρθωσης σφάλματος μέσω θερμοκρασίας. Δηλαδή μειώνεται το σφάλμα κατά την μέτρηση της υγρασίας εφ' όσον υπάρχει διαθέσιμη μέτρηση θερμοκρασίας εκείνη την στιγμή. Συνεπώς για να υπολογιστεί με μικρό σφάλμα η υγρασία στην κουζίνα θα γίνει κλήση της συνάρτησης calculateHumidity() και μέσα σε αυτήν θα περαστεί και η μέτρηση της θερμοκρασίας στο ίδιο δωμάτιο που πραγματοποιήθηκε από άλλον αισθητήρα, και ο αριθμός του Pin του H1H4030, όπως παρουσιάζεται παρακάτω:

```
client.print("<hum_kouzina>");  
float humKouzina = calculateHumidity(tempKouzina, hih4030PinKouzina);  
client.print(humKouzina);  
client.print("</hum_kouzina>");
```

Όσον αφορά τον PIR, στην έτοιμη συνάρτηση `digitalRead()` που παρέχει το abstraction layer του Arduino, δίνεται ως παράμετρος ο αριθμός του Pin του αισθητήρα αυτού του τύπου και η μεταβλητή που επιστρέφεται ελέγχεται για το εάν είναι HIGH ή LOW ως εξής:

```
client.print("<pir_saloni>");  
int proximitySaloni = digitalRead(pirPinSaloni);  
if (proximitySaloni == LOW)  
    {  
        client.print("Ησυχία");  
    }  
else  
    {  
        client.print("Κλέφτης");  
    }  
client.print("</pir_saloni>");
```

Στην περίπτωση που γίνει HIGH συνεπάγεται ότι ο αισθητήρας έχει ανιχνεύσει κίνηση, οπότε και θα εμφανίσει στο αρχείο XML το string "Κλέφτης" ανάμεσα από τα αντίστοιχα tags.

Ενδεικτικά ο κώδικας για εμφάνιση δεδομένων από έναν αισθητήρα κάθε κατηγορίας, και την μέτρηση να τοποθετείται εντός tags στο XML, δίνεται παρακάτω.

```
client.print("<temp_saloni>");  
int tempSaloni = calculateTemperature(lm35PinSaloni);  
client.print(tempSaloni);  
client.print("</temp_saloni>");  
[...]  
  
client.print("<hum_saloni>");  
float humSaloni = calculateHumidity(tempSaloni, hih4030PinSaloni);  
client.print(humSaloni);  
client.print("</hum_saloni>");  
[...]  
  
client.print("<fws_saloni>");  
int fwsSaloni = calculateLux(temt6000PinSaloni);  
client.print(fwsSaloni);  
client.print("</fws_saloni>");  
[...]
```



```
client.print("<wind>");
int aeras = calculateWind(windSensorTempPin, windSensorRVPin);
client.print(aeras);
client.print("</wind>");
[...]

client.print("<pir_saloni>");
int proximitySaloni = digitalRead(pirPinSaloni);
if (proximitySaloni == LOW)
{
    client.print("Ήσυχία");
}
else
{
    client.print("Κλέφτης");
}
client.print("</pir_saloni>");
[...]

client.print("<co2>");
double valCO2 = kSensor.getCO2('P');
client.print(valCO2);
client.print("</co2>");
```

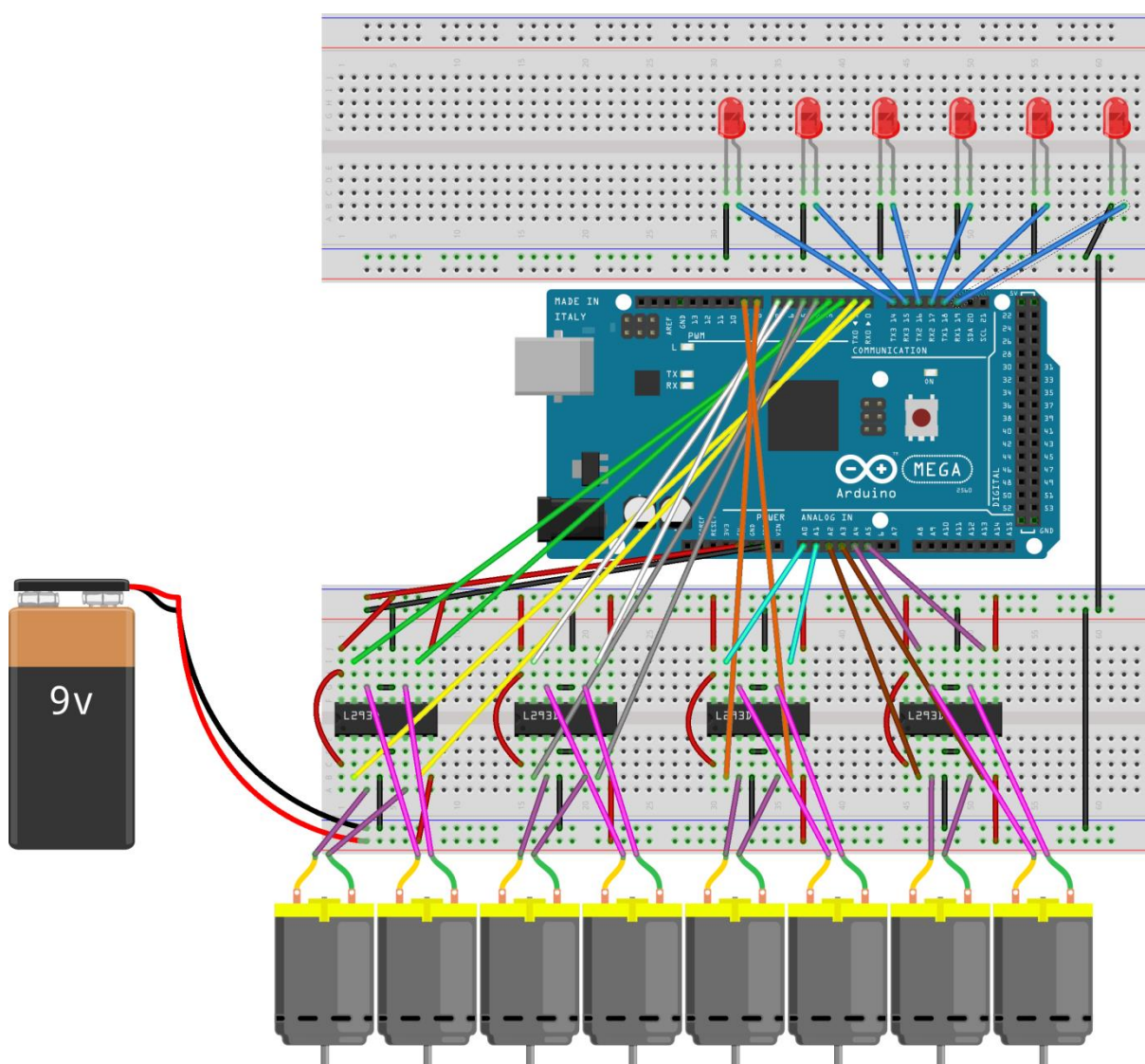
Ο πλήρης προτεινόμενος κώδικας του σταθμού βάσης αισθητήρων που περιλαμβάνει τις δηλώσεις μεταβλητών και αντικειμένων, τον κώδικα αρχικοποίησης και δήλωσης των εισόδων και εξόδων, την ρουτίνα δημιουργία και ανανέωσης του webserver, τις ρουτίνες ανάγνωσης και κανονικοποίησης των δεδομένων από τους αισθητήρες, αλλά και την ρουτίνα εμφάνισής τους στο XML, για το σύνολο των αισθητήρων κάθε κατηγορίας (έξι LM35, πέντε H1H4030, τρεις TEMT6000, έξι Parallax PIR, έναν Wind Sensor Rev C και έναν K30), δίνεται στο παράρτημα 6.3.1.

Η πλήρης αντιστοιχία κάθε pin του Arduino με κάθε αισθητήρα, τον τύπο των εξόδων τους, τα μετρούμενα μεγέθη τους, το όνομα των μεταβλητών τους στο XML, και με τα δωμάτια του έξυπνου σπιτιού στα οποία είναι τοποθετημένοι, δίνεται παρακάτω.

Πίνακας 3-1. Αντιστοιχία Pins του Arduino του πρώτου σταθμού βάσης, με αισθητήρες, τύπους εξόδων, μεταβλητές στο XML, μετρούμενα μεγέθη και δωμάτια					
Pin	Αισθητήρας	Τύπος Εξόδου	Μεταβλητή (tag) σε XML	Μετρούμενο μέγεθος	Δωμάτιο
A0	LM35	An. Out	temp_trapezaria	θερμοκρασία	Τραπεζαρία
A1	LM35	An. Out	temp_kouzina	θερμοκρασία	Κουζίνα
A2	LM35	An. Out	temp_saloni	θερμοκρασία	Σαλόνι
A3	LM35	An. Out	temp_grafeio	θερμοκρασία	Γραφείο
A4	LM35	An. Out	temp_krevatokamara	θερμοκρασία	Υπνοδωμάτιο
A5	LM35	An. Out	temp_khpos	θερμοκρασία	Κήπος
A6	HIH4030	An. Out	hum_trapezaria	υγρασία	Τραπεζαρία
A7	HIH4030	An. Out	hum_kouzina	υγρασία	Κουζίνα
A8	HIH4030	An. Out	hum_saloni	υγρασία	Σαλόνι
A9	HIH4030	An. Out	hum_grafeio	υγρασία	Γραφείο
A10	HIH4030	An. Out	hum_khpos	υγρασία	Κήπος
A11	TEMT6000	An. Out	fws_kouzina	φωτεινότητα	Κουζίνα
A12	TEMT6000	An. Out	fws_saloni	φωτεινότητα	Σαλόνι
A13	TEMT6000	An. Out	fws_khpos	φωτεινότητα	Κήπος
A14	Wind Sensor Rev C.	An. Out TMP	wind	Ταχύτητα ανέμου	Σαλόνι
A15		An. Out RV			
D2	Parallax PIR	D. Out	pir_trapezaria	κίνηση	Τραπεζαρία
D3	Parallax PIR	D. Out	pir_kouzina	κίνηση	Κουζίνα
D4	Parallax PIR	D. Out	pir_saloni	κίνηση	Σαλόνι
D5	Parallax PIR	D. Out	pir_grafeio	κίνηση	Γραφείο
D6	Parallax PIR	D. Out	pir_krevatokamara	κίνηση	Υπνοδωμάτιο
D7	Parallax PIR	D. Out	pir_khpos	κίνηση	Κήπος
D12	K30	D. Out SPI (RX)	co2	διοξείδιο του άνθρακα	Σαλόνι
D13		D.Out SPI (TX)			

3.2 Υλοποίηση σταθμού βάσης ελέγχου κινητήρων και φώτων

Για την υλοποίηση του δεύτερου σταθμού βάσης έγινε χρήση του μικροελεγκτή ATmega2560 του Arduino Mega -ομοίως με τον πρώτο σταθμό βάσης- και του Ethernet ελεγκτή (Controller) Wiznet 5500 της Ethernet Shield 2. Στο Arduino αυτό συνδέονται τα έξι φώτα LED των δωματίων και οι τέσσερις γέφυρες οδήγησης L293 οι οποίες ελέγχουν την λειτουργία των οκτώ κινητήρων των πορτών και παραθύρων. Η χρήση της εν λόγω γέφυρας οδήγησης έχει περιγραφεί στο Κεφάλαιο 2.9. Η τροφοδοσία των κινητήρων πραγματοποιείται μέσω ξεχωριστού τροφοδοτικού 9V, και αποτελεί την μέγιστη ασφαλή τάση τροφοδοσίας τους.



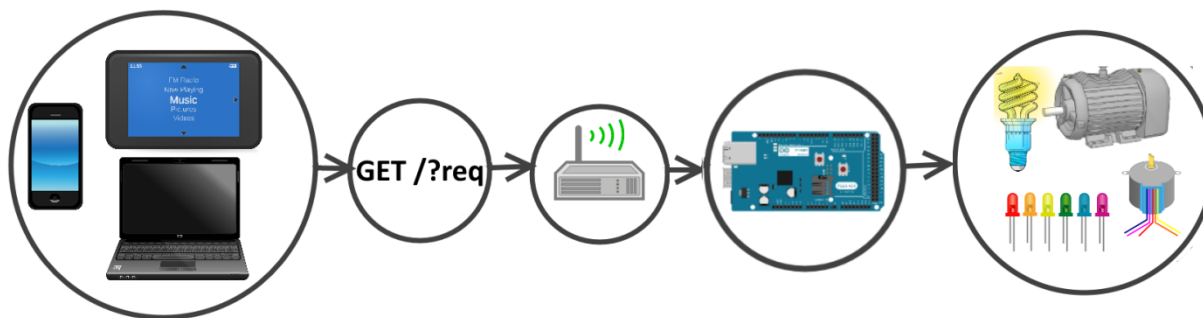
Σχήμα 3-6. Παράσταση συνδεσμολογίας συνόλου L293, κινητήρων και LEDs σε σταθμό βάσης ελέγχου κινητήρων και φώτων

Στα Pins D0 έως D9 και A0 έως A5 συνδέονται οι είσοδοι των L293, ενώ στα Pins D14 έως D19 συνδέονται τα LED του φωτισμού. Όλα τα άνωθεν Pins είναι προγραμματισμένα στην προτεινόμενη υλοποίηση να λειτουργούν ως ψηφιακές έξοδοι, και να έχουν την αρχική κατάστασή τους σε LOW.

Ομοίως με τον σταθμό βάσης αισθητήρων, στον σταθμό βάσης ελέγχου κινητήρων και φώτων, απαιτείται η εκτέλεση ρουτίνας δημιουργίας ενός Web Server στο Arduino με την Ethernet Shield, όπως παρουσιάστηκε στο Κεφάλαιο 3.1. Στην συνέχεια, ελέγχουμε τα GET requests τα οποία εκτελεί ο χρήστης μέσω της εφαρμογής ή ενός browser στην IP του Arduino, και για κάθε διαφορετικό request, εκτελείται διαφορετική ρουτίνα ελέγχου μηχανισμών φώτων ή κινητήρων πορτών/παραθύρων, ως εξής:

```
[...]
else if (readString.indexOf("?epai") > 0) //parathiro ipnodwmatioy
{
    digitalWrite(A4, HIGH);
    digitalWrite(A5, LOW);
    delay(2000);
    digitalWrite(A4, LOW);
    digitalWrite(A5, LOW);
}
else if (readString.indexOf("?rpai") > 0) //parathiro ipnodwmatioy
{
    digitalWrite(A4, LOW);
    digitalWrite(A5, HIGH);
    delay(2000);
    digitalWrite(A4, LOW);
    digitalWrite(A5, LOW);
}
[...]
else if (readString.indexOf("?aft") > 0) //fws trapezaria on
{
    digitalWrite(14, HIGH);
}
else if (readString.indexOf("?kft") > 0) //fws trapezaria off
{
    digitalWrite(14, LOW);
}
[...]
```

Ο έλεγχος της παραμέτρου του GET request πραγματοποιείται μέσω της μεθόδου `indexof()` στην μεταβλητή `readString`.



Σχήμα 3-7. Εκτέλεση GET request για τον έλεγχο κινητήρων και φώτων

Για την παράμετρο του GET request κρίθηκε απαραίτητη η επινοήση και υλοποίηση ενός βασικού πρωτοκόλλου. Συγκεκριμένα ο πρώτος χαρακτήρας του GET request δύναται να είναι "a" (άνοιγμα), το οποίο περιγράφει ενεργοποίηση σε φως, "k" (κλείσιμο), το οποίο περιγράφει απενεργοποίηση σε φως, "e" (enable), το οποίο περιγράφει ενεργοποίηση κινητήρα προς την φορά ανοίγματος και r (reverse enable), το οποίο περιγράφει ενεργοποίηση κινητήρα προς την αντίθετη φορά. Οι επόμενοι, ένας, δύο, ή τρεις χαρακτήρες περιγράφουν το είδος του μηχανισμού. Ενδεικτικά: f (φως), pa (παράθυρο), mrv (μπαλκονόπορτα βόρεια). Ο τελευταίος χαρακτήρας περιγράφει το δωμάτιο στο οποίο είναι τοποθετημένος ο μηχανισμός. Ενδεικτικά: t (τραπεζαρία), g (γραφείο) κ.ο.κ. Για παράδειγμα το request `GET /?afg` αντιστοιχεί σε άνοιγμα στο φως του γραφείου.

Ο πλήρης προτεινόμενος κώδικας του σταθμού βάσης ελέγχου κινητήρων και φώτων που περιλαμβάνει τις δηλώσεις μεταβλητών και αντικειμένων, τον κώδικα αρχικοποίησης και δήλωσης των εξόδων, την ρουτίνα δημιουργία και ανανέωσης του webserver, τις ρουτίνες ελέγχου των GET requests, και της εκτέλεσης των διεργασιών ενεργοποίησης ή απενεργοποίησης των Pins, δίνεται στο παράρτημα 6.3.2.

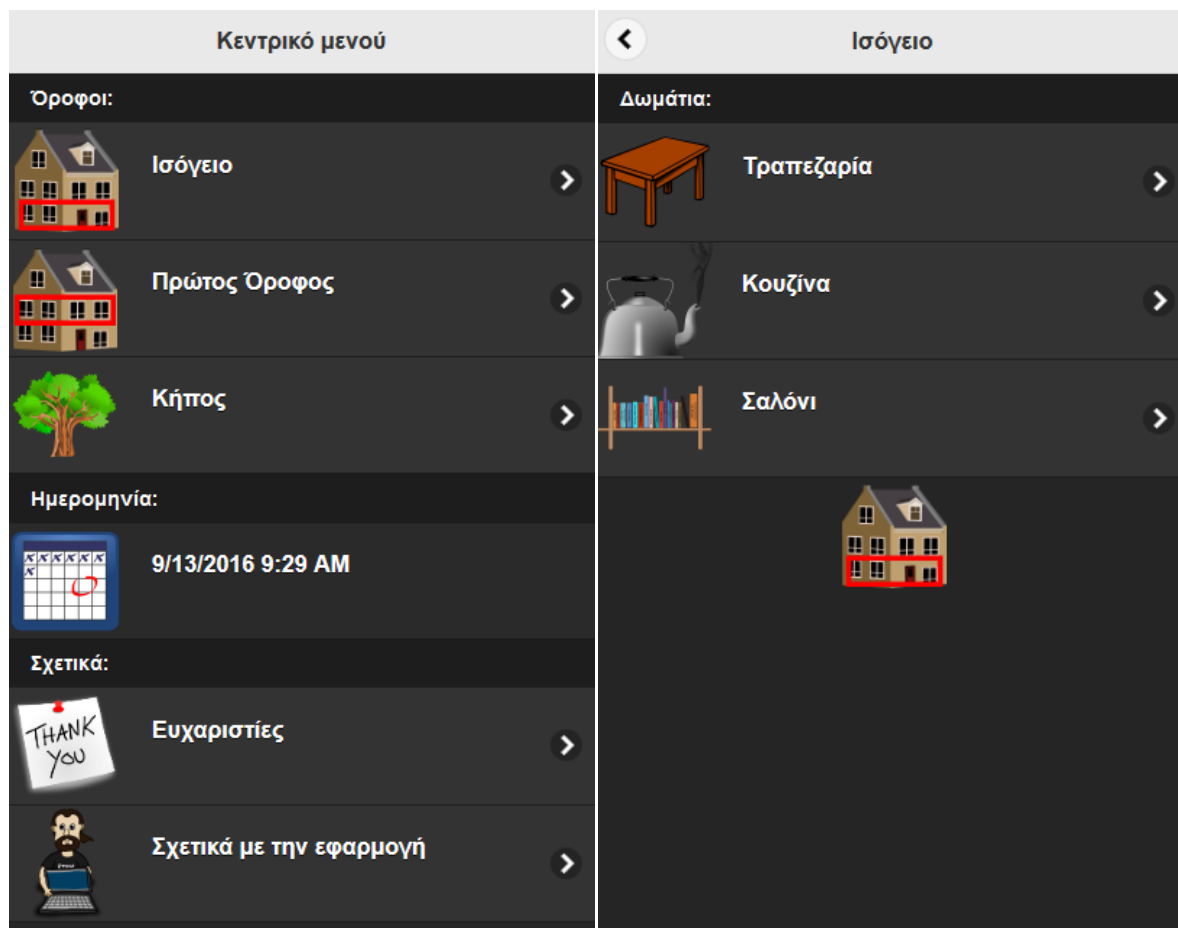
Η πλήρης αντιστοιχία κάθε pin του Arduino με τα κυκλώματα στα οποία συνδέονται, το κανάλι του κυκλώματος στο οποίο συνδέονται, τις μεταβλητές στο υλοποιηθέν πρωτόκολλο που περιγράφουν την λειτουργία τους, αλλά και τους μηχανισμούς και τα δωμάτια στα οποία βρίσκονται, δίνεται στον πίνακα παρακάτω:

Πίνακας 3-2. Αντιστοιχία Pins του Arduino του δεύτερου σταθμού βάσης, με συνδεδεμένο κύκλωμα, κανάλι L293, μεταβλητές στο πρωτόκολλο, και δωμάτια					
Pin	A/A και Κύκλωμα	Κανάλι	Μεταβλητή σε πρωτόκολλο για άνοιγμα	Μεταβλητή σε πρωτόκολλο για κλείσιμο	Δωμάτιο και μηχανισμός
0	1 - L293	1	?epat	?rpat	παράθυρο τραπεζαρίας
1					
2		2	?epak	?rpak	παράθυρο κουζίνας
3					
4	2 - L293	1	?empk	?rmpk	μπαλκονόπορτα κουζίνας
5					
6		2	?empns	?rmpns	ν. μπαλκονόπορτα σαλονιού
7					
8	3 - L293	1	?empvs	?rmpvs	β. μπαλκονόπορτα σαλονιού
9					
A0		2	?epag	?rpag	παράθυρο γραφείου
A1					
A2	4 - L293	1	?empg	?rmpg	μπαλκονόπορτα γραφείου
A3					
A4		2	?epai	?rpai	παράθυρο υπνοδωματίου
A5					
14	1 - LED	1	?aft	?kft	φως τραπεζαρίας
15	2 - LED	1	?afk	?kfk	φως κουζίνας
16	3 - LED	1	?afs	?kfs	φως σαλονιού
17	4 - LED	1	?afg	?kfg	φως γραφείου
18	5 - LED	1	?afm	?kfm	φως υπνοδωματίου
19	6 - LED	1	?afp	?kfp	φως κήπου

3.3 Υλοποίηση διεπαφής χρήστη

3.3.1 Υλοποίηση κεντρικού μενού διεπαφής χρήστη πλατφόρμας έξυπνου σπιτιού

Η κεντρική οθόνη της διεπαφής χρήστη της υλοποιηθείσας εφαρμογής αποτελείται από μία λίστα με τις επιλογές Ισόγειο, Πρώτος Όροφος, Κήπος, Ημερομηνία, Ευχαριστίες και Σχετικά (About). Οι επιλογές Ισόγειο και Πρώτος Όροφος παραπέμπουν σε άλλες λίστες με τις ονομασίες των δωματίων που περιλαμβάνουν. Η επιλογή Κήπος παραπέμπει κατευθείαν στις ενδείξεις αισθητήρων του κήπου και στους ενεργοποιητές του, χωρίς να παρεμβάλλεται κάποιο υπομενού. Τα στοιχεία της λίστας κατηγοριοποιούνται μέσω διαχωριστών (Όροφοι, Ημερομηνία, Σχετικά). Το στοιχείο της λίστας ημερομηνία δεν παραπέμπει σε κάποιο υπομενού, συνεπώς έχει αφαιρεθεί από αυτό ο σχετικός δείκτης στο δεξιό μέρος του UI. Για την εμφάνιση της ημερομηνίας και της ώρας στο κεντρικό μενού υλοποιήθηκε σχετική συνάρτηση. Όλες οι εικόνες που χρησιμοποιήθηκαν στην εφαρμογή είναι υπό άδεια Public Domain, το οποίο συνεπάγεται την ελεύθερη χρήση τους τόσο στην παρούσα εργασία όσο και για εμπορικές εφαρμογές.



Σχήμα 3-8. Στιγμιότυπα οθόνης υλοποιηθείσας εφαρμογής

Παρακάτω δίνεται ο κώδικας για την προτεινόμενη υλοποίηση του κεντρικού μενού της εφαρμογής, ο οποίος δημιουργεί το άνωθεν UI.

```
<div data-role="page" data-theme="b">

    <div data-role="header" data-position="fixed" data-theme="a">
        <h1>Κεντρικό μενού</h1>
    </div>

    <div data-role="main" class="ui-content">
        <ul data-role="listview">
            <li data-role="list-divider">Όροφοι:</li>
            <li><a href="isogeio.html">
                <h4>Ισόγειο</h4></a></li>
            <li><a href="prwtosorofos.html">
                <h4>Πρώτος Όροφος</h4></a></li>
            <li><a href="khpos.html" data-ajax="false">
                <h4>Κήπος</h4></a></li>
            <li data-role="list-divider">Ημερομηνία:</li>
            <li data-icon="false">
                <h4 id="demo"></h4></li>
            <li data-role="list-divider">Σχετικά:</li>
            <li><a href="thanks.html">
                <h4>Ευχαριστίες</h4></a></li>
            <li><a href="about.html">
                <h4>Σχετικά με την εφαρμογή</h4></a></li>
        </ul>
    </div>

</div>
```

Το flag data-theme="b" το οποίο έχει τοποθετηθεί στο div της σελίδας παραπέμπει σε αλλαγή θέματος (χρώματος εμφάνισης μαύρο) του περιεχομένου της σελίδας. Αντίστοιχα, στον τίτλο έχει ορισθεί data-theme="a" το οποίο παραπέμπει σε διαφορετικό χρώμα εμφάνισης. Τα flags data-ajax="false" (ή εναλλακτικά το rel="external") το οποίο έχει τοποθετηθεί στην επιλογή Κήπος, καθότι είναι επιθυμητό η συγκεκριμένη επιλογή να μην φορτωθεί μέσω AJAX -το οποίο αποτελεί την default ρύθμιση του jquery mobile-. Η φόρτωση μέσω AJAX δημιουργούσε πρόβλημα στην ανανέωση των δεδομένων των αισθητήρων. Το αφορά την εκάστοτε εικόνα που χρησιμοποιείται σε κάθε στοιχείο της λίστας.

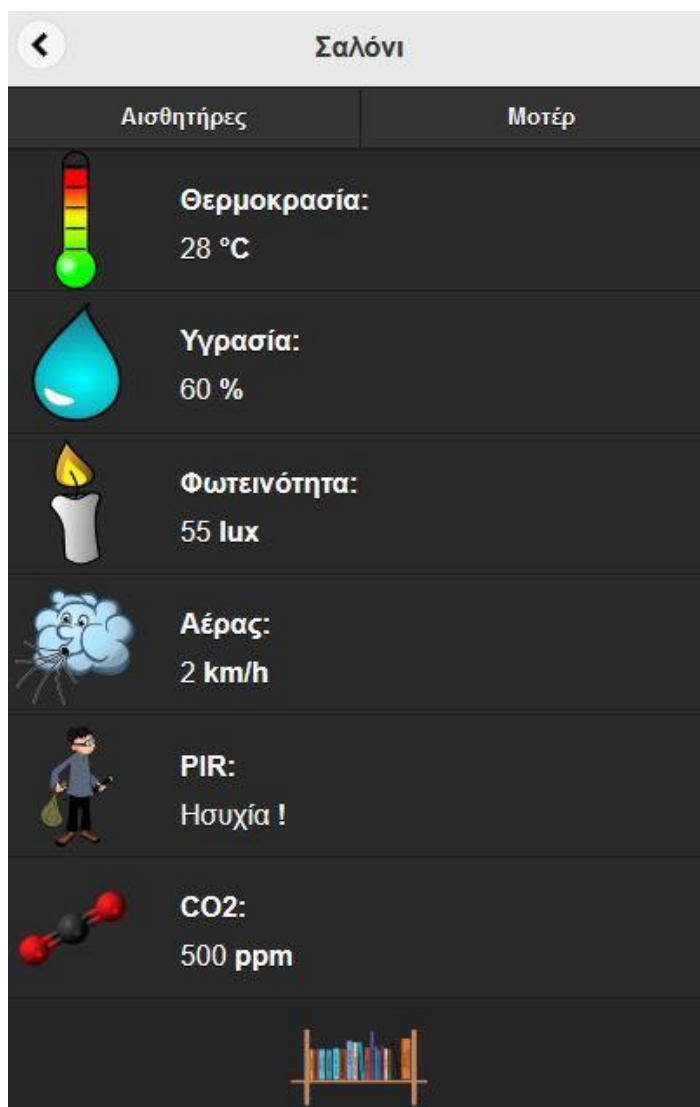
Ο κώδικας για την εμφάνιση της ημερομηνίας και της ώρας στο κεντρικό μενού δίνεται παρακάτω:

```
<script>
    var dt = new Date();
    var d = dt.toLocaleDateString();
    var t = dt.toLocaleTimeString();
    t = t.replace(/\u200E/g, '');
    t = t.replace(/^(^[\d]*\d{1,2}:\d{1,2}):\d{1,2}([\d]*)$/ , '$1$2');
    var result = d + ' ' + t;
    document.getElementById("demo").innerHTML = result;
</script>
```

Είναι επιθυμητό η ημερομηνία και η ώρα που επιστρέφεται στην εφαρμογή να μην περιλαμβάνει τα δευτερόλεπτα, η σωστή εμφάνιση των οποίων θα απαιτούσαν συνεχή ανανέωση της εφαρμογής. Η συνάρτηση `toLocaleTimeString` επιστρέφει την ώρα σε μορφή ΩΩ:ΛΛ:ΔΔ, συνεπώς απαιτείται η χρήση Regular Expressions ώστε να αφαιρεθούν. Το `\u200E` αποτελεί τον Unicode χαρακτήρα που ορίζει γραφή από αριστερά προς τα δεξιά. Στο τέλος η ημερομηνία που παρέχεται μέσω της συνάρτησης `toLocaleDateString()` και η ώρα που διαμορφώθηκε, μέσω του Regular Expression, τοποθετούνται στην μεταβλητή `result` και δύναται να εμφανιστούν στην σελίδα, μέσω του Id (Identifier) που της δόθηκε.

3.3.2 Υλοποίηση διεπαφής χρήστη άντλησης δεδομένων αισθητήρων πλατφόρμας έξυπνου σπιτιού

Για την εμφάνιση των μετρήσεων των αισθητήρων κάθε δωματίου υλοποιήθηκε κατάλληλη διεπαφή χρήστη (UI) σε μορφή λίστας δεδομένων. Η παρούσα εφαρμογή δύναται να εμφανίσει δεδομένα θερμοκρασίας, υγρασίας, φωτεινότητας, έντασης ανέμου, κίνησης και διοξειδίου του άνθρακα. Η λίστα των μετρήσεων είναι υλοποιημένη με τέτοιο τρόπο, ώστε να εμφανίζει τα δεδομένα κάθε αισθητήρα συνοδευόμενα από το χαρακτηριστικό τους εικονίδιο και την μονάδα μέτρησής τους. Κάθε δωμάτιο του σπιτιού μπορεί να περιέχει και λιγότερους εκ των άνωθεν αισθητήρες, συνεπώς το UI έχει προσαρμοσθεί για κάθε περίπτωση δωματίου από τον μέγιστο αριθμό αισθητήρων μέχρι τον ελάχιστο. Το επιλεγθέν δωμάτιο εμφανίζεται στον τίτλο της εφαρμογής.



Σχήμα 3-9. Στιγμιότυπο οθόνης δεδομένων αισθητήρων υλοποιηθείσας εφαρμογής

Στην περίπτωση των οθονών εμφάνισης δεδομένων, δεν απαιτείται απλώς η εμφάνιση ενός στατικού μενού, αλλά και η άντληση δεδομένων από τον webserver ο οποίος τρέχει στο Arduino με την Ethernet Shield και παράγει το προτεινόμενο API. Η μορφή του προτεινόμενου API είναι xml, με τα tags να προσδιορίζουν τόσο το είδος της μέτρησης όσο και του δωματίου από τα οποία προέρχονται. Συνεπώς απαιτείται και η υλοποίηση μηχανισμού parsing του xml για την τοποθέτηση κάθε μέτρησης σε μεταβλητές προσβάσιμες από το UI.

Παρακάτω δίνεται ο κώδικας για την προτεινόμενη υλοποίηση του τμήματος εμφάνισης δεδομένων αισθητήρων.

```

<div data-role="page" data-theme="b">

<div data-role="header" data-position="fixed" data-theme="a" role="banner"
  class="ui-header ui-bar-a ui-header-fixed slidedown">

  <a href="isogeio.html" rel="external" class="ui-btn-left
    ui-alt-icon ui-nodisc-icon ui-btn ui-icon-carat-l ui-btn-icon-notext
    ui-corner-all" data-role="button" role="button">Back</a>

  <h1 class="ui-title" role="heading" aria-level="1">Σαλόνι</h1>
</div>

<div data-role="navbar">
  <ul>
    <li><a href="saloni.html" rel="external">Αισθητήρες</a></li>
    <li><a href="salonimotors.html" rel="external">Μοτέρ</a></li>
  </ul>
</div>

<div data-role="main" class="ui-content">
  <ul data-role="listview">
    <li><h4>Θερμοκρασία:</h4>
      <span id="temp_saloni"></span><b> °C</b></li>
    <li><h4>Υγρασία:</h4>
      <span id="hum_saloni"></span><b> %</b></li>
    <li><h4>Φωτεινότητα:</h4>
      <span id="fws_saloni"></span><b> lux</b></li>
    <li><h4>Αέρας:</h4>
      <span id="wind"></span><b> km/h</b></li>
    <li><h4>PIR:</h4>
      <span id="pir_saloni"></span><b> !</b></li>
    <li><h4>CO2:</h4>
      <span id="co2"></span><b> ppm</b></li>
  </ul>

</div>
<center></center>
</div>

```

Οι οθόνες μετρήσεων βρίσκονται εντός υπο-μενού της εφαρμογής, συνεπώς τοποθετήθηκε πλήκτρο μετάβασης προς προηγούμενα μενού δίπλα στον τίτλο. Με τον τρόπο αυτό, εξασφαλίζεται η σωστή περιήγηση ακόμα και στις πλατφόρμες που δεν έχουν ξεχωριστό πλήκτρο μετάβασης. Κάτω από τον τίτλο τοποθετήθηκαν καρτέλες μετάβασης (navbar) με την χρήση του <div data-role="navbar">. Εντός της λίστας, οι μετρήσεις εμφανίζονται μέσω του . Ενδεικτικά, οι μεταβλητές οι οποίες

εμφανίζονται στο Σαλόνι, έχουν ως id τα temp_saloni, hum_saloni, fws_saloni, wind, pir_saloni και co2. Ο κώδικας για τον μηχανισμό τοποθέτησης των μετρήσεων του xml στις μεταβλητές με τα άνωθεν ids, παρουσιάζεται παρακάτω.

```
<script>
var xmlhttp, xmlDoc;
xmlhttp = new XMLHttpRequest();
xmlhttp.open("GET", "http://192.168.1.21", false);
xmlhttp.send();
xmlDoc = xmlhttp.responseXML;

document.getElementById("temp_saloni").innerHTML=xmlDoc.
getElementsByTagName("temp_saloni")[0].childNodes[0].nodeValue;

document.getElementById("hum_saloni").innerHTML=xmlDoc.
getElementsByTagName("hum_saloni")[0].childNodes[0].nodeValue;

document.getElementById("fws_saloni").innerHTML=xmlDoc.
getElementsByTagName("fws_saloni")[0].childNodes[0].nodeValue;

document.getElementById("wind").innerHTML=xmlDoc.
getElementsByTagName("wind")[0].childNodes[0].nodeValue;

document.getElementById("pir_saloni").innerHTML=xmlDoc.
getElementsByTagName("pir_saloni")[0].childNodes[0].nodeValue;

document.getElementById("co2").innerHTML=xmlDoc.
getElementsByTagName("co2")[0].childNodes[0].nodeValue;

</script>
```

Κάνοντας GET request στην IP στην οποία τρέχει ο WebServer του Arduino, επιστρέφεται ως απάντηση (response) το xml αρχείο το οποίο περιέχει ανάμεσα σε tags όλες τις μετρήσεις. (π.χ. <co2>[μέτρηση]</co2>). Η μέθοδος getElementsByTagName() χρησιμοποιείται, ώστε να αντληθούν από το xmlDoc οι πληροφορίες ανάμεσα στα tags, ενώ η getElementById() χρησιμοποιείται, ώστε να δοθεί ένα id σε μία μεταβλητή. Το id αυτό χρησιμοποιείται στο HTML τμήμα μέσω του για να εμφανιστεί η συγκεκριμένη πληροφορία στην λίστα (Listview) του jQuery Mobile.

Οι πληροφορίες στην εκάστοτε σελίδα κρίνεται απαραίτητο να ανανεώνονται αυτόματα, ώστε ο χρήστης να έχει σε πραγματικό χρόνο επίβλεψη των μετρήσεων. Για τον λόγο αυτό, υλοποιήθηκε συνάρτηση αυτόματης ανανέωσης ανά τακτά χρονικά διαστήματα. Στην προτεινόμενη υλοποίηση η ανανέωση πραγματοποιείται κάθε πέντε δευτερόλεπτα, ενώ αν ο χρήστης επιθυμεί μπορεί, επιλέγοντας την καρτέλα "αισθητήρες", να ανανεώσει τις μετρήσεις πριν το χρονικό διάστημα αυτό. Ο προτεινόμενος κώδικας ανανέωσης των μετρήσεων δίνεται παρακάτω:

```

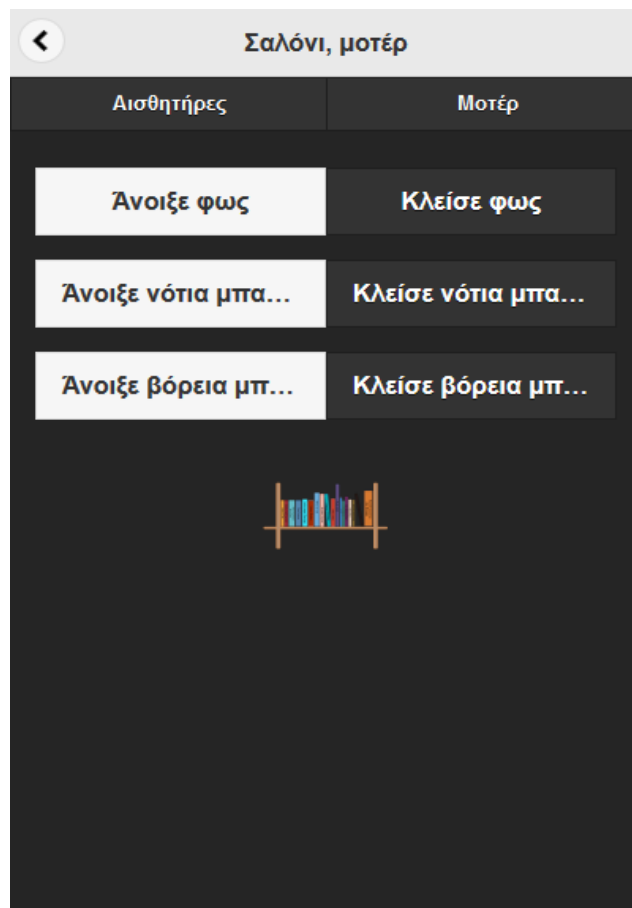
<script>
function autoRefresh()
{
    window.location = window.location.href;
}
setInterval('autoRefresh()', 5000); // 5000ms, 5 δεψτερολεπτα
</script>

```

Η συνάρτηση autoRefresh ανανεώνει όλη την σελίδα η οποία εμφανίζεται στον χρήστη, ενώ μέσω της setInterval ορίζεται ο ρυθμός ανανέωσης.

3.3.3 Υλοποίηση διεπαφής χρήστη ελέγχου κινητήρων και φωτισμού έξυπνου σπιτιού

Στην προτεινόμενη υλοποίηση ο χρήστης δύναται να ελέγξει των φωτισμό σε κάθε δωμάτιο, αλλά και την ενεργοποίηση και αντίστροφη ενεργοποίηση των κινητήρων των παραθύρων.



Σχήμα 3-10. Στιγμιότυπο οθόνης ελέγχου ενεργοποιητών υλοποιηθείσας εφαρμογής

Παρακάτω δίνεται ο κώδικας για την προτεινόμενη υλοποίηση του UI του τμήματος ελέγχου ενεργοποιητών (κινητήρων πορτών/παραθύρων και φώτων) της εφαρμογής.

```
<div data-role="page" data-theme="b">

<div data-role="header" data-position="fixed" data-theme="a" role="banner"
  class="ui-header ui-bar-a ui-header-fixed slidedown">

  <a href="isogeio.html" rel="external" class="ui-btn-left ui-alt-icon
  ui-nodisc-icon ui-btn ui-icon-carat-l ui-btn-icon-notext
  ui-corner-all" data-role="button" role="button">Back</a>

  <h1 class="ui-title" role="heading" aria-level="1">Σαλόνι, μοτέρ</h1>

</div>

<div data-role="navbar">
  <ul>
    <li><a href="saloni.html" rel="external" >Αισθητήρες</a></li>
    <li><a href="salonimotors.html" rel="external">Μοτέρ</a></li>
  </ul>
</div>

<div data-role="main" class="ui-content">
  <fieldset class="ui-grid-a">
    <div class="ui-block-a"><button class="ui-btn ui-btn-a"
      id="button0">Άνοιξε φως</button></div>
    <div class="ui-block-b"><button class="ui-btn ui-btn-b"
      id="button1">Κλείσε φως</button></div>
    <div class="ui-block-a"><button class="ui-btn ui-btn-a"
      id="button2">Άνοιξε νότια μπαλκονόπορτα</button></div>
    <div class="ui-block-b"><button class="ui-btn ui-btn-b"
      id="button3">Κλείσε νότια μπαλκονόπορτα</button></div>
    <div class="ui-block-a"><button class="ui-btn ui-btn-a"
      id="button4">Άνοιξε βόρεια μπαλκονόπορτα</button></div>
    <div class="ui-block-b"><button class="ui-btn ui-btn-b"
      id="button5">Κλείσε βόρεια μπαλκονόπορτα</button></div>
  </fieldset>
</div>
<center></center>

</div>
```

Στο UI της προτεινόμενης εφαρμογής τοποθετείται ένα πλέγμα (grid) από πλήκτρα (buttons), κάθε ένα εκ των οποίων καταλαμβάνει το μισό πλάτος της οθόνης μέσω της χρήσης των κλάσεων ui-block-a και ui-block-b. Αντίστοιχα μέσω των κλάσεων ui-button-a και

ui-button-b, επιλέγουμε διαφορετικό θέμα (χρώμα) των πλήκτρων ενεργοποίησης και απενεργοποίησης. Σε κάθε πλήκτρο τοποθετείται και διαφορετικό (μοναδικό) id, ώστε να είναι εφικτό μέσω του συγκεκριμένου id να προγραμματισθεί κάποια διεργασία κατά το πάτημά του.

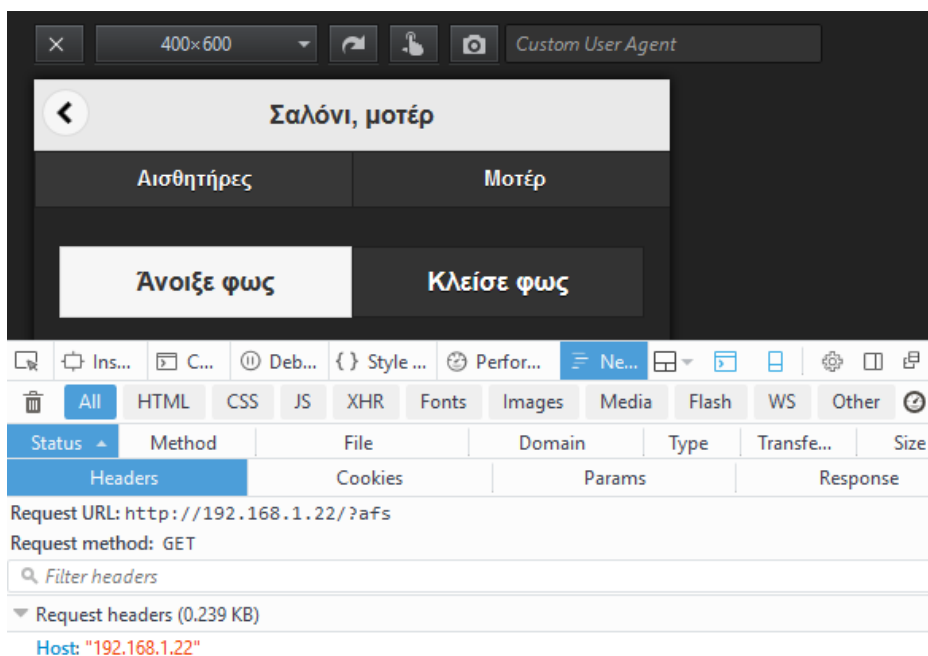
Παρακάτω δίνεται ο προτεινόμενος κώδικας του μηχανισμού ο οποίος επιτρέπει την εκτέλεση μίας διεργασίας κατά την επιλογή ενός από τα άνωθεν πλήκτρα του UI.

```
<script>
```

```
$(document).ready(function() {  
    $("#button0").click(function() {  
        $.get("http://192.168.1.22/?afs", function(data, status) {  
        });  
    });  
    $("#button1").click(function() {  
        $.get("http://192.168.1.22/?kfs", function(data, status) {  
        });  
    });  
    $("#button2").click(function() {  
        $.get("http://192.168.1.22/?empns", function(data, status) {  
        });  
    });  
    $("#button3").click(function() {  
        $.get("http://192.168.1.22/?rmpns", function(data, status) {  
        });  
    });  
    $("#button4").click(function() {  
        $.get("http://192.168.1.22/?empvs", function(data, status) {  
        });  
    });  
    $("#button5").click(function() {  
        $.get("http://192.168.1.22/?rmpvs", function(data, status) {  
        });  
    });  
});
```

```
</script>
```

Η μέθοδος `click()` εκτελείται όταν επιλεγθεί ένα button, βάσει του id του, το οποίο έχει ορισθεί στο UI. Όταν επιλεγθεί, πραγματοποιείται ένα get request στον Web Server που εκτελείται στο Arduino, με το οποίο είναι συνδεδεμένα τα LEDs και οι κινητήρες των πορτών/παραθύρων. Η παράμετρος του κάθε get request (π.χ. `/?afs`), η οποία λαμβάνεται από το Arduino, ακολουθεί το προτεινόμενο πρωτόκολλο που υλοποιήθηκε. Κάθε παράμετρος που στέλνεται στον Web Server του Arduino αντιστοιχεί σε ενεργοποίηση ή απενεργοποίηση διαφορετικού κινητήρα πόρτας/παραθύρου ή φωτός.



Σχήμα 3-11. Επιβεβαίωση εκτέλεσης get request στον Server του Arduino κατά την επιλογή "Άνοιξε φως" της προτεινόμενης εφαρμογής

3.3.4 Υλοποίηση λοιπών τμημάτων διεπαφής χρήστη εφαρμογής πλατφόρμας έξυπνου σπιτιού

Ομοίως, στο παράρτημα της παρούσας εργασίας παρέχεται ο κώδικας και στιγμιότυπα οθόνης της προτεινόμενης εφαρμογής για όλα τα υλοποιηθέντα μενού (Κεντρικό μενού 6.1.1, Ισόγειο 6.1.3, Πρώτος Όροφος 6.1.5, Ευχαριστίες 6.1.11, Σχετικά με την εφαρμογή 6.1.13), οθόνες ενδείξεων αισθητήρων (Ενδείξεις Τραπεζαρίας 6.1.15, Ενδείξεις Κουζίνας 6.1.19, Ενδείξεις Σαλονιού 6.1.23, Ενδείξεις Γραφείου 6.1.27, Ενδείξεις Κρεβατοκάμαρας 6.1.31, Ενδείξεις Κήπου 6.1.7) και οθόνες ελέγχου φωτισμού και κινητήρων (Ενεργοποιητές Τραπεζαρίας 6.1.17, Ενεργοποιητές Κουζίνας 6.1.21, Ενεργοποιητές Σαλονιού 6.1.25, Ενεργοποιητές Γραφείου 6.1.29, Ενεργοποιητές Κρεβατοκάμαρας 6.1.33, Ενεργοποιητές Κήπου 6.1.9).

3.3.5 Υλοποίηση Android εφαρμογής με χρήση της προτεινόμενης διεπαφής χρήστη

Η επιλογή τεχνολογιών διαδικτύου (HTML5 και jQuery Mobile) στην υλοποίηση της διεπαφής χρήστη και του μηχανισμού άντλησης δεδομένων συνεπάγεται την δυνατότητα εκτέλεσης σε πλήθος διαφορετικών πλατφορμών και λειτουργικών συστημάτων με λίγες ή καθόλου αλλαγές. Η υλοποίηση Android εφαρμογής με τον άνωθεν μηχανισμό και το UI απαιτεί την χρήση ενός container τύπου WebView, κώδικα φόρτωσης και εμφάνισης μπάρας προόδου (progress bar), κώδικα χειρισμού του πλήκτρου "Πίσω" (Back Button Handling), καθώς και ορισμένες αλλαγές για πλήρη εκμετάλλευση ορισμένων ειδικών δυνατοτήτων της πλατφόρμας οι οποίες δεν υπάρχουν με την ίδια μορφή στους ηλεκτρονικούς υπολογιστές (π.χ. αλλαγή εμφάνισης από portrait σε landscape της εφαρμογής βάσει των ενδείξεων του orientation sensor της κινητής συσκευής, χρήση αδειών κ.α.). Για την υλοποίηση της Android εφαρμογής χρησιμοποιήθηκε το προτεινόμενο από την Google περιβάλλον, το Android Studio. Αποτελεί ένα IDE βασισμένο στο IntelliJ, στο οποίο ενσωματώνονται όλα τα εργαλεία (adb, sdks, editors, compilers, emulators) για ανάπτυξη και αποσφαλμάτωση εφαρμογών στην πλατφόρμα. Ο βασικός κώδικας Java για χρήση της κλάσης WebView και εμφάνιση του υλοποιηθέντος προτεινόμενου UI της εφαρμογής δίνεται παρακάτω. Τα αρχεία .html, .js, .css και .png που χρησιμοποιήθηκαν στην κατασκευή του UI τοποθετούνται στον φάκελο /assets/ του Android Project.

```
public class SmartHomeActivity extends Activity {  
    WebView webview;  
  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        setContentView(R.layout.main);  
  
        webview = (WebView) findViewById(R.id.web_engine);  
        webview.setWebViewClient(new NewWebViewClient());  
        webview.getSettings().setJavaScriptEnabled(true); //js  
        webview.getSettings().setAllowFileAccess(true); //assets prosvasi  
        webview.setScrollBarStyle(WebView.SCROLLBARS_OUTSIDE_OVERLAY);  
        //webview.loadUrl("http://tuc.gr"); //για debugging  
        webview.loadUrl("file:///android_asset/mainmenu.html");  
    }  
}
```

Μέσω της μεθόδου `setJavaScriptEnabled(true)` ενεργοποιούμε την Javascript εντός του μηχανισμού `WebView`, και μέσω της μεθόδου `setAllowFileAccess(true)` του επιτρέπουμε τοπική πρόσβαση στα αρχεία της εφαρμογής. Τέλος για να εμφανίσει το `WebView` το αρχείο `mainmenu.html` το οποίο βρίσκεται μέσα στα `/assets/`, καλείται η μέθοδος `loadUrl("file:///android_asset/mainmenu.html ")`.

Απαιτείται, επίσης, ο παρακάτω προτεινόμενος κώδικας Java για τον προσδιορισμό της χρήσης του πλήκτρου επιστροφής (Back) το οποίο διαθέτουν όλες οι συσκευές με λειτουργικό σύστημα Android.

```
@Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        if ((keyCode == KeyEvent.KEYCODE_BACK) && webview.canGoBack()) {
            webview.goBack();
            return true;
        }
        return super.onKeyDown(keyCode, event);
    }
```

Γίνεται σύγκριση του `KeyCode` το οποίο πατήθηκε, με την σταθερά `KEYCODE_BACK`, και, στην συνέχεια, μέσω της μεθόδου `canGoBack()` της κλάσης `WebView`, ελέγχουμε αν το `WebView` διαθέτει ιστορικό επιστροφής σε προηγούμενη σελίδα. Αν και οι δύο αυτές συνθήκες είναι αληθείς, εκτελείται η μέθοδος `goBack()`. Με τον τρόπο αυτό, έστω ότι ο χρήστης έχει επιλέξει το μενού του Ισογείου, αν χρησιμοποιήσει το πλήκτρο επιστροφής της συσκευής του, θα μεταφερθεί ξανά στο κεντρικό μενού.

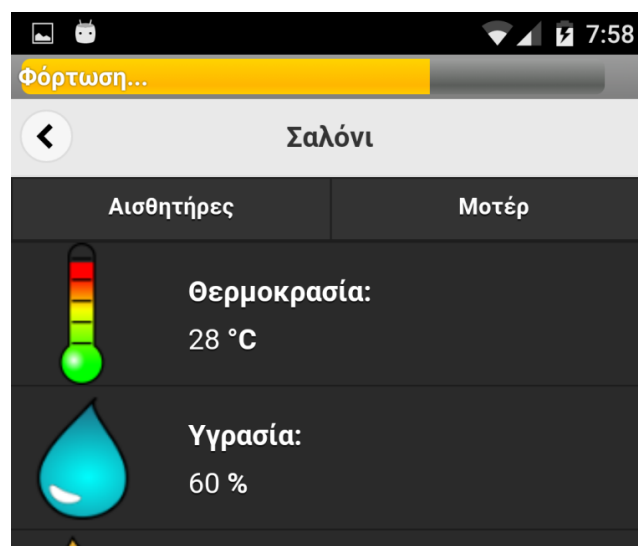
Ένας ακόμα μηχανισμός, ο οποίος υλοποιήθηκε σε Java, για την σωστή μεταφορά της εφαρμογής στο Android, περιλαμβάνει τον έλεγχο και την διαχείριση των επιλογών αφής του χρήστη. Ο μηχανισμός αυτός προβλέπει να ανοίγουν όλες οι σελίδες εντός του `WebView`, εκτός από την περίπτωση που η διεύθυνση (URL - Uniform Resource Locator) αρχίζει με το "mailto:", όπως συμβαίνει στο πεδίο επικοινωνίας της οθόνης About της προτεινόμενης εφαρμογής. Στην περίπτωση αυτή, είναι επιθυμητό το `mailto url` να το χειριστεί κάποια άλλη εξωτερική εφαρμογή εγκατεστημένη στο Android όπως το Gmail. Η κώδικας του μηχανισμού αυτού δίνεται παρακάτω.

```
webView.setWebViewClient(new NewWebViewClient());
```

```
private class NewWebViewClient extends WebViewClient {
    @Override
    public boolean shouldOverrideUrlLoading(WebView view, String url) {
        if ((url != null && (url.startsWith("mailto:")))) {
            view.getContext().startActivity(
                new Intent(Intent.ACTION_VIEW, Uri.parse(url)));
            return true;
        } else {
            return false;
        }
    }
}
```

Αρχικά, πραγματοποιείται επέκταση της κλάσης `WebViewClient`. Στην συνέχεια, μέσα στην μέθοδο `shouldOverrideUrlLoading()`, γίνεται έλεγχος αν το url αρχίζει με το string "mailto:". Στην περίπτωση αυτή, ο χειρισμός (handling) του url θα γίνει από τον σχετικό μηχανισμό του Android, ο οποίος θα εμφανίσει όλες τις εγκατεστημένες εφαρμογές που δύναται να χειριστούν URIs (Uniform Resource Identifier) της μορφής αυτής.

Στις εφαρμογές που αντλούν δεδομένα από κάποιον διακομιστή (Server) θεωρείται προτεινόμενη πρακτική η τοποθέτηση μπάρας προόδου της φόρτωσης των δεδομένων.



Σχήμα 3-12. Μπάρα προόδου άντλησης δεδομένων από Arduino WebServer στο Android

Για την εμφάνιση της άνωθεν μπάρας, στην προτεινόμενη εφαρμογή υλοποιήθηκε το παρακάτω κομμάτι κώδικα Java.

```
final Activity MyActivity = this;

webView.setWebChromeClient (new WebChromeClient () {

    public void onProgressChanged (WebView view, int progress)
    {

        MyActivity.setTitle ("Φόρτωση ...");
        MyActivity.setProgress (progress * 100);

        if (progress == 100)
            MyActivity.setTitle (R.string.app_name);
    }

});
```

Στην μεταβλητή progress τύπου integer τοποθετείται η επί τοις εκατό πρόοδος της φόρτωσης των δεδομένων. Στην συνέχεια, πραγματοποιείται έλεγχος για την περίπτωση κατά την οποία η μεταβλητή λάβει την τιμή εκατό, το οποίο συνεπάγεται ολοκλήρωση της φόρτωσης. Στην περίπτωση αυτή τοποθετείται το όνομα της εφαρμογής στην γραμμή τίτλου, οπότε και ο χρήστης γνωρίζει ότι έχει ολοκληρωθεί η διαδικασία.

Κάθε εφαρμογή Android πρέπει να έχει στον αρχικό της φάκελο ένα αρχείο με όνομα AndroidManifest.xml, το οποίο περιέχει πληροφορίες απαραίτητες για την σωστή εκτέλεσή της και το οποίο απαιτείται για την μεταγλώττισή (compilation) της. Οι πληροφορίες αυτές διαβάζονται από την Android συσκευή, η οποία επιχειρεί να εκτελέσει την εφαρμογή. Τέτοιες πληροφορίες είναι η έκδοση της εφαρμογής, το μοναδικό όνομα πακέτου της (package name), οι άδειες (permissions) οι οποίες ζητάει, την παλαιότερη έκδοση του Android την οποία υποστηρίζει, την νεότερη έκδοση στην οποία τρέχει σωστά, τα activities και τις βιβλιοθήκες που περιέχει και άλλα. Παρακάτω δίνεται τμήμα του αρχείου Manifest που υλοποιήθηκε για την προτεινόμενη εφαρμογή και περιγράφει παραμέτρους απαραίτητες για την λειτουργία της. Το πλήρες Manifest δύναται να βρεθεί στο παράρτημα 6.2.3 της παρούσας εργασίας.

```
package="com.giwrgos.diplwmatikismarhome"
[...]
```

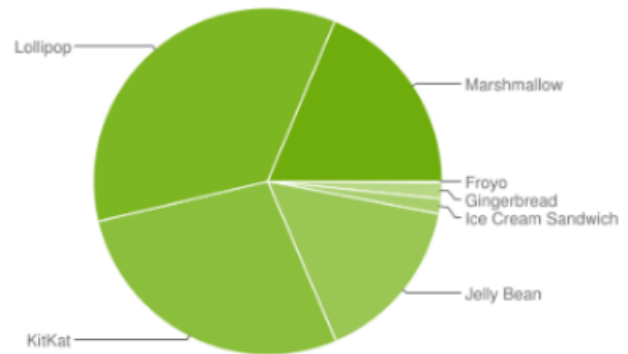
```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
[...]
```

```
<application
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    [...]
<activity
    android:name=".MainActivity"
    android:label="@string/app_name"
    android:configChanges="orientation|screenSize">
    <intent-filter>
        [...]
    </intent-filter>
</activity>
</application>
</manifest>
```

Αρχικά, δίνεται ένα όνομα πακέτου στην εφαρμογή, το οποίο ακολουθεί την τεχνική των namespaces της Java. Στην συνέχεια, αναφέρονται οι άδειες (permissions) οι οποίες απαιτούνται για να λειτουργήσει η εφαρμογή. Στην παρούσα υλοποίηση απαιτούνται οι `android.permission.INTERNET` και `android.permission.ACCESS_NETWORK_STATE`, καθώς η προτεινόμενη εφαρμογή είναι απαραίτητο να συνδέεται με το δίκτυο για να αντλεί δεδομένα από τον Web Server που εκτελείται στο Arduino. Η δυνατότητα να περιστραφεί η οθόνη βάσει μετρήσεων από τον αισθητήρα προσανατολισμού, χωρίς να γίνεται κάθε φορά επανεκκίνηση της `mainActivity`, αποκτάται μέσω του `android:configChanges` με παραμέτρους τα `"orientation|screenSize"`.

Μιας και είναι επιθυμητό μία Android εφαρμογή να είναι συμβατή με όσο το δυνατόν περισσότερες συσκευές, αποφεύχθηκε η χρήση APIs που εισήχθησαν σε πολύ πρόσφατες εκδόσεις του Android. Κάτι τέτοιο θα καθιστούσε την εφαρμογή ασύμβατη με παλαιότερες συσκευές. Η υλοποιηθείσα προτεινόμενη εφαρμογή δύναται να εγκατασταθεί και να εκτελεστεί στις εκδόσεις Android 2.3 Gingerbread έως και Android 7.0 Nougat, το οποίο σύμφωνα με τα τελευταία στατιστικά κατανομής εκδόσεων (Σεπτέμβριος 2016), αντιστοιχεί στο 99,9% των ενεργών συσκευών με πρόσβαση στο Google Play. (Google, 2016)

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	1.5%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.4%
4.1.x	Jelly Bean	16	5.6%
4.2.x		17	7.7%
4.3		18	2.3%
4.4	KitKat	19	27.7%
5.0	Lollipop	21	13.1%
5.1		22	21.9%
6.0	Marshmallow	23	18.7%



Σχήμα 3-13. Στατιστικά στοιχεία κατανομής εκδόσεων του Android, Σεπτέμβριος 2016

Ο πλήρης κώδικας της προτεινόμενης υλοποίησης Android εφαρμογής, με όλα τα βασικά (mainActivity.java) και βοηθητικά αρχεία απαραίτητα για το compilation (AndroidManifest.xml, layout/main.xml, values/strings.xml) δίνονται στο παράρτημα 6.2 της παρούσας εργασίας.

4. Συμπεράσματα και Μελλοντική Έρευνα

4.1 Συμπεράσματα

Στην παρούσα διπλωματική εργασία αναπτύχθηκε μία πλήρης πλατφόρμα επικοινωνίας για παρακολούθηση και έλεγχο ευφυών κατοικιών, με σκοπό την βελτιστοποίηση της ενεργειακής αποδοτικότητάς τους. Η υλοποίηση αποτελείται από το δίκτυο των αισθητήρων οι οποίοι είναι εγκατεστημένοι στα δωμάτια, (θερμοκρασίας, υγρασίας, φωτεινότητας, διοξειδίου του άνθρακα, ανέμου και κίνησης), το δίκτυο των ενεργοποιητών (LEDs, κινητήρες παραθύρων, ηχείο), τους σταθμούς βάσης (δύο Arduino Mega με Ethernet Shields) με τον προτεινόμενο κώδικα, και την εφαρμογή (προτεινόμενο UI και μηχανισμός επικοινωνίας) για υπολογιστές και έξυπνες φορητές συσκευές.

Η εφαρμογή επιτρέπει την παρακολούθηση όλων των παραμέτρων των εγκατεστημένων αισθητήρων αλλά και τον απομακρυσμένο έλεγχο των ενεργοποιητών. Ο ρυθμός ανανέωσης των δεδομένων δύναται να αλλαχθεί ανεξάρτητα για κάθε δωμάτιο και για κάθε αισθητήρα, ώστε ο χρήστης να έχει σαφή εικόνα των δεδομένων σε πραγματικό χρόνο. Ταυτόχρονα, το σύστημα προσφέρει και δυνατότητες αυτοματοποίησης, μιας και οι ενδείξεις δύναται να χρησιμοποιηθούν ως δεδομένα για τους ενεργοποιητές. Ενδεικτικά υλοποιήθηκε η ανίχνευση κίνησης του αισθητήρα PIR να ενεργοποιεί αυτόματα το ηχείο του συναγερμού.

Η προτεινόμενη υλοποίηση επιτεύχθηκε με ελάχιστο κόστος σε σύγκριση με τις εμπορικά διαθέσιμες λύσεις. Το σύστημα δύναται να επεκταθεί απεριόριστα, καθώς επιτρέπει την σύνδεση όσων σταθμών βάσεων (συνεπώς και αισθητήρων και ενεργοποιητών) είναι επιθυμητό στην εκάστοτε ευφυή οικία. Η επέκταση του συστήματος δύναται να επιτευχθεί με χρήση του προτεινόμενου κώδικα σταθμών βάσης, όχι μόνο με το Arduino Mega που χρησιμοποιήθηκε, αλλά και με οποιαδήποτε άλλη Development Board χρησιμοποιεί το abstraction layer του Arduino. Συνεπώς, αποτελεί μία πλήρως παραμετροποιήσιμη λύση, καθώς διαφορετικές Development Boards, χρησιμοποιώντας το υλοποιηθέν API, μπορούν και επικοινωνούν με την προτεινόμενη εφαρμογή. Ταυτόχρονα, με το υλοποιηθέν API, ανεξάρτητοι προγραμματιστές δύναται να αναπτύξουν εφαρμογές συμβατές με τους σταθμούς βάσης.

Στην δυνατότητα επέκτασης, στην ευελιξία και την παραμετροποιησιμότητα της προτεινόμενης πλατφόρμας, συμβάλλει και το γεγονός ότι η πλήρης υλοποίηση πραγματοποιήθηκε με χρήση τεχνολογιών (υλικού και λογισμικού) ανοιχτού κώδικα. Η διαθεσιμότητα των σχηματικών διαγραμμάτων και των σημείων ελέγχου (test points), όσον

αφορά το υλικό, του κώδικα των βιβλιοθηκών του UI και του Arduino, καθιστά ευκολότερη την αντιμετώπιση προβλημάτων και την εξέλιξη. Οι βιβλιοθήκες του Arduino, επίσης, καθιστούν την πλατφόρμα συμβατή με όλα τα γνωστά πρωτόκολλα δικτύων αισθητήρων (SPI, I2C, 1-Wire, CAN-BUS). Έτσι, δύναται να συνδεθούν σχεδόν όλοι οι αισθητήρες που κυκλοφορούν στο λιανικό εμπόριο, το οποίο με την σειρά του καθιστά και το κόστος της εξέλιξης / αναβάθμισης πολύ μικρότερο σε σύγκριση με τις εμπορικές υλοποιήσεις κατασκευαστών.

Το σύστημα υλοποιήθηκε με fail-safe λογική: Η τυχόν αστοχία ενός μεμονωμένου υποσυστήματος (αισθητήρα, ενεργοποιητή ή και ολόκληρου σταθμού βάσης) δεν λειτουργεί με καταστροφικό τρόπο προς το υπόλοιπο σύνολο, το οποίο εξακολουθεί να λειτουργεί.

Το σύνολο της πλατφόρμας λειτουργεί με ικανοποιητική ταχύτητα και σταθερότητα, αλλά και με μικρή κατανάλωση ενέργειας.

Τέλος, η προτεινόμενη εφαρμογή ελέγχου υλοποιήθηκε με ανεξαρτήτου πλατφόρμας τεχνολογίες (cross platform), το οποίο καθιστά εφικτή την εκτέλεσή της σε όλα τα σύγχρονα λειτουργικά συστήματα υπολογιστών, (Windows, Linux), αλλά και έξυπνων φορητών συσκευών (Android, iOS).

4.2 Μελλοντική Έρευνα

Η υλοποιηθείσα πλατφόρμα δύναται να αποτελέσει έναυσμα σε λοιπούς ερευνητικούς τομείς του διαδικτύου των πραγμάτων, αλλά και γενικότερα.

Αρχικά θεωρείται κρίσιμο να εξεταστεί η σύνδεση της πλατφόρμας -και όλης της ευφυούς οικίας- με ΑΠΕ (Ανανεώσιμες Πηγές Ενέργειας) για περαιτέρω μείωση του περιβαλλοντικού αντίκτυπού της. Ο έλεγχος των παραμέτρων των ΑΠΕ δύναται να πραγματοποιείται μέσω της προτεινόμενης πλατφόρμας (π.χ. έλεγχος φόρτισης μπαταριών, έλεγχος κατάστασης μπαταριών, αποδοτικότητα φωτοβολταϊκών, τρέχουσα κατανάλωση κτλ).

Η χρήση της παρούσας πλατφόρμας σε βιομηχανικό περιβάλλον θα απαιτούσε πλήθος αλλαγών στην υλοποίηση, τόσο στην επιλογή βιομηχανικά πιστοποιημένων και ανθεκτικών αισθητήρων, όσο και στην εμβάθυνση σε παραμέτρους όπως ο προσδόκιμος χρόνος ζωής του συστήματος, η κρισιμότητα σωστής λειτουργίας, η ηλεκτρομαγνητική συμβατότητα και θωράκιση.

Επιπλέον, ενδιαφέρον παρουσιάζουν οι αλλαγές οι οποίες θα απαιτούνταν στην πλατφόρμα, για χρήση σε εφαρμογές εξαιρετικά χαμηλής κατανάλωσης και συνδυασμό με energy harvesting μεθόδους. Κάτι τέτοιο προϋποθέτει την ελαχιστοποίηση του duty cycle και την εκμετάλλευση των λειτουργιών χαμηλής κατανάλωσης (low power modes) του μικροελεγκτή, τα οποία δεν εξετάστηκαν στην παρούσα εργασία.

Η συλλογή ενεργειακών στατιστικών δεδομένων από πλήθος χρηστών της εφαρμογής -αφού αρχικά εξεταστούν οι ηθικές και νομικές προεκτάσεις- θα είχε επίσης ενδιαφέρον, καθότι θα μπορούσαν να προκύψουν χρήσιμες πληροφορίες, οι οποίες θα μείωναν τον φόρτο των σταθμών παροχής ενέργειας.

Τέλος, ως μελλοντικός τομέας έρευνας αξίζει να αναφερθεί η διασφάλιση της ασφαλούς επικοινωνίας και η προστασία των ευαίσθητων δεδομένων που ανταλλάσει η παρούσα πλατφόρμα και η ασφάλεια λοιπών εμπορικών λύσεων στον χώρο του διαδικτύου των πραγμάτων.

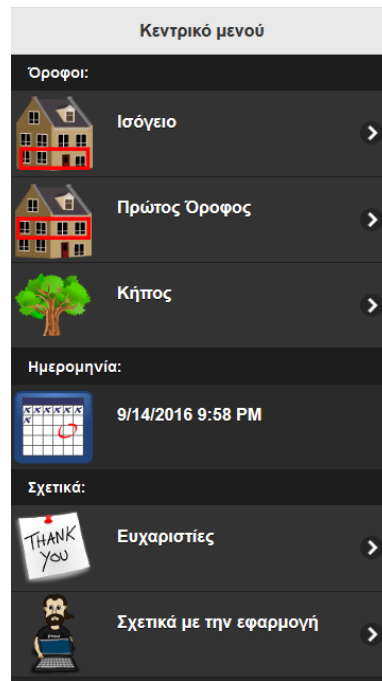
5. Βιβλιογραφία

- [1] Atmel. (2014, 02 10). *Atmel Datasheets*. Ανάκτηση από ATmegaV-2560 Datasheet: http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf
- [2] CO2meter. (2015, 05 01). *CO2 Meter Datasheet*. Ανάκτηση από K-30 Datasheet: <http://co2meters.com/Documentation/Datasheets/DS30-01%20-%20K30.pdf>
- [3] Deitel, H. M. (2001). *XML: How to program*. Upper Saddle River, New Jersey: Prentice Hall.
- [4] Dutson, P. (2013). *Query mobile: In 24 hours*. Indianapolis: Sams.
- [5] Firtman, M. R. (2012). *JQuery mobile: Up and running*. Sebastopol: O'Reilly Media.
- [6] Google. (2016, 08 05). *Android Developer Statistics*. Ανάκτηση από Google Android Dashboard: <https://developer.android.com/about/dashboards>
- [7] Honeywell. (2016, 05 07). *Honeywell Datasheets*. Ανάκτηση από H1H-4030 Datasheet: <http://sensing.honeywell.com/honeywell-sensing-hih4030-4031-series-product-sheet-009021-4-en.pdf>
- [8] Modern Device. (2016, 08 01). *Modern Device Technical Notes*. Ανάκτηση από Wind Sensor Rev. C Technical Notes: <https://moderndevice.com/product/wind-sensor/>
- [9] Parallax. (2014, 03 27). *Parallax Datasheets*. Ανάκτηση από PIR Datasheet: <https://www.parallax.com/sites/default/files/downloads/555-28027-PIR-Sensor-Product-Guide-v2.3.pdf>
- [10] Texas Instruments. (2016, 08 01). *Texas Instruments Datasheets*. Ανάκτηση από Texas Instruments LM35 Datasheet: <http://www.ti.com/lit/ds/symlink/lm35.pdf>
- [11] Texas Instruments. (2016, 01 01). *Texas Instruments Datasheets*. Ανάκτηση από L293X Datasheet: <http://www.ti.com/lit/ds/symlink/l293.pdf>
- [12] Trevennor, A. (2014). *Arduino smart home automation*. Berkeley: Apress.
- [13] Vishay. (2011, 08 23). *Vishay Datasheets*. Ανάκτηση από TEMT6000 Datasheet: <http://www.vishay.com/docs/81579/temt6000.pdf>
- [14] Wiznet. (2015, 10 5). *Wiznet Datasheets*. Ανάκτηση από Wiznet 5100: http://www.wiznet.co.kr/wp-content/uploads/wiznethome/Chip/W5100/Document/W5100_Datasheet_v1.2.6.pdf

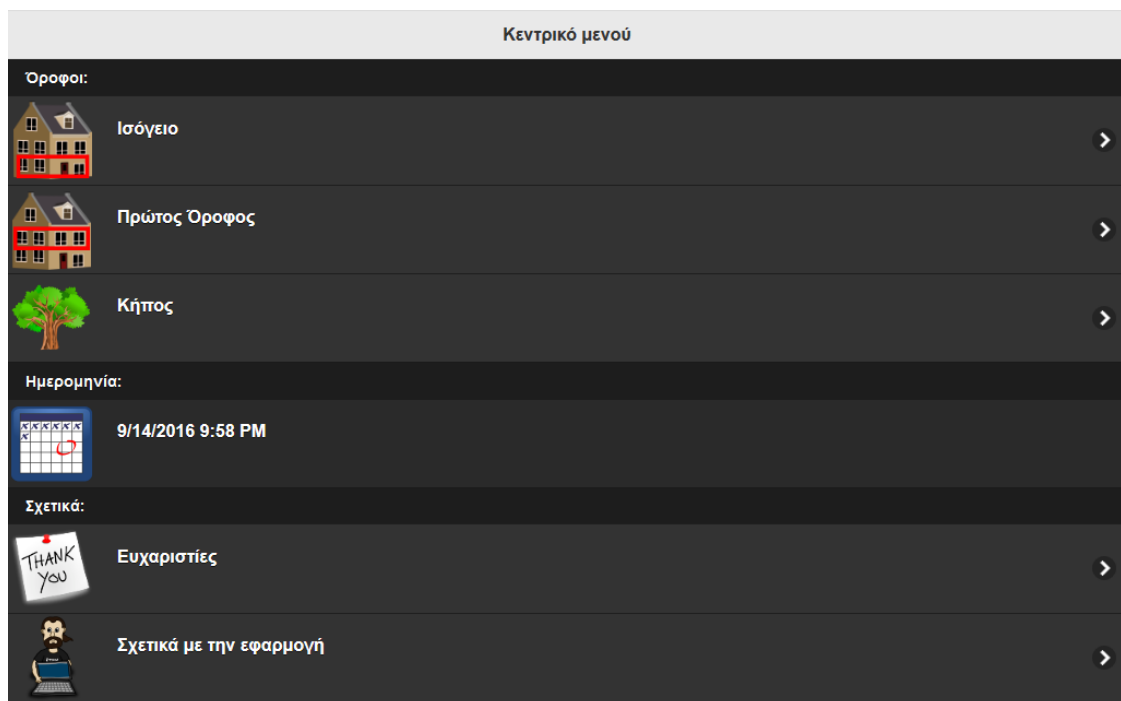
6. Παράρτημα

6.1 Στιγμιότυπα οθόνης και κώδικας HTML5 εφαρμογής

6.1.1 Στιγμιότυπα κεντρικής οθόνης εφαρμογής



Σχήμα 6-1. Διεπαφή κεντρικού μενού υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου



Σχήμα 6-2. Διεπαφή κεντρικού μενού υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή

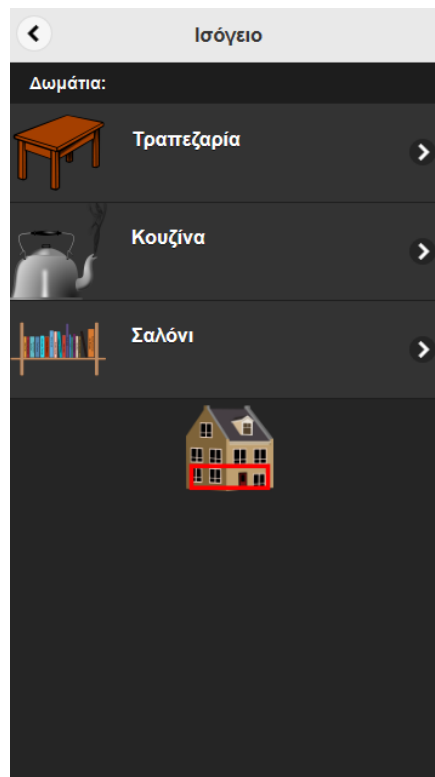
6.1.2 Κώδικας κεντρικής οθόνης εφαρμογής

```

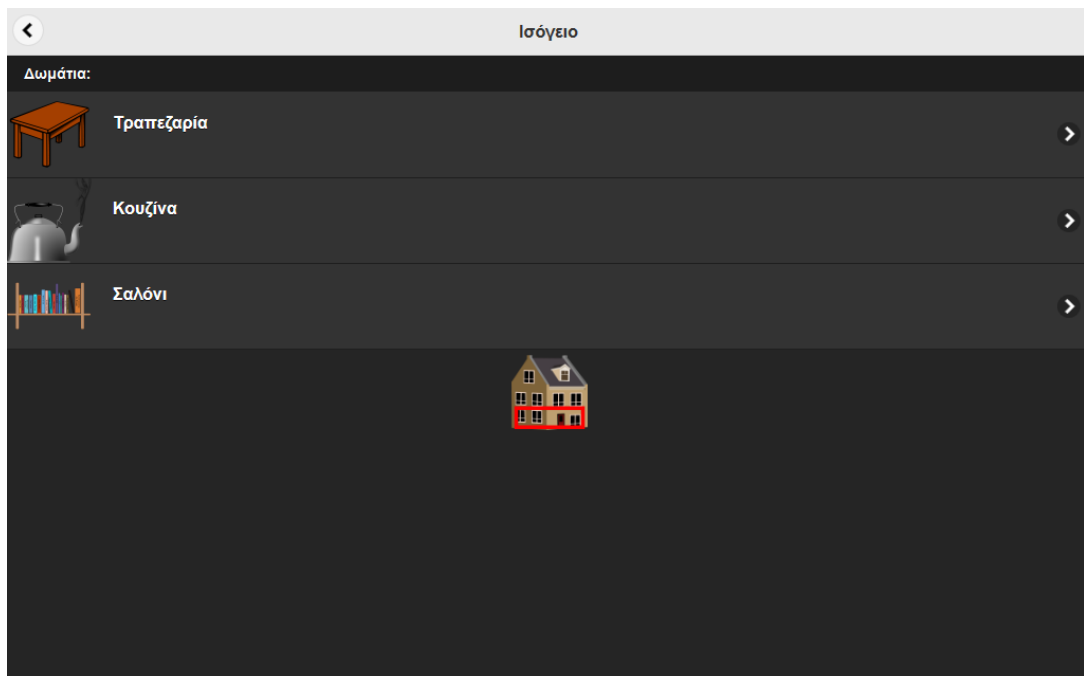
<!DOCTYPE html>
<html>
<head><meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="jquery.mobile-1.4.5.min.css" />
    <script src="jquery-1.11.1.min.js"></script>
    <script src="jquery.mobile-1.4.5.min.js"></script></head>
<body>
<div data-role="page" data-theme="b">
    <div data-role="header" data-position="fixed" data-theme="a">
        <h1>Κεντρικό μενού</h1></div>
    <div data-role="main" class="ui-content">
        <ul data-role="listview">
            <li data-role="list-divider">Όροφοι:</li>
            <li><a href="isogeio.html">
            <h4>Ισόγειο</h4></a></li>
            <li><a href="prwtosorofos.html">
            <h4>Πρώτος Όροφος</h4></a></li>
            <li><a href="khpos.html" data-ajax="false">
             <h4>Κήπος</h4></a></li>
            <li data-role="list-divider">Ημερομηνία:</li>
            <li data-icon="false">
            <h4 id="demo"></h4></li>
            <li data-role="list-divider">Σχετικά:</li>
            <li><a href="thanks.html">
            <h4>Ευχαριστίες</h4></a></li>
            <li><a href="about.html">
            <h4>Σχετικά με την εφαρμογή</h4></a></li>
        </ul>
    </div>
</div>
<script>
    var dt = new Date();
    var d = dt.toLocaleDateString();
    var t = dt.toLocaleTimeString();
    t = t.replace(/\u200E/g, '');
    t = t.replace(/^(^[\d]*\d{1,2}:\d{1,2}):\d{1,2}([\d]*)$/ , '$1$2');
    var result = d + ' ' + t;
    document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>

```

6.1.3 Στιγμιότυπα οθόνης Ισογείου εφαρμογής



Σχήμα 6-3. Διεπαφή Ισογείου υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου



Σχήμα 6-4. Διεπαφή Ισογείου υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή

6.1.4 Κώδικας οθόνης Ισογείου εφαρμογής

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="jquery.mobile-1.4.5.min.css" />
    <script src="jquery-1.11.1.min.js"></script>
    <script src="jquery.mobile-1.4.5.min.js"></script>
  </head>

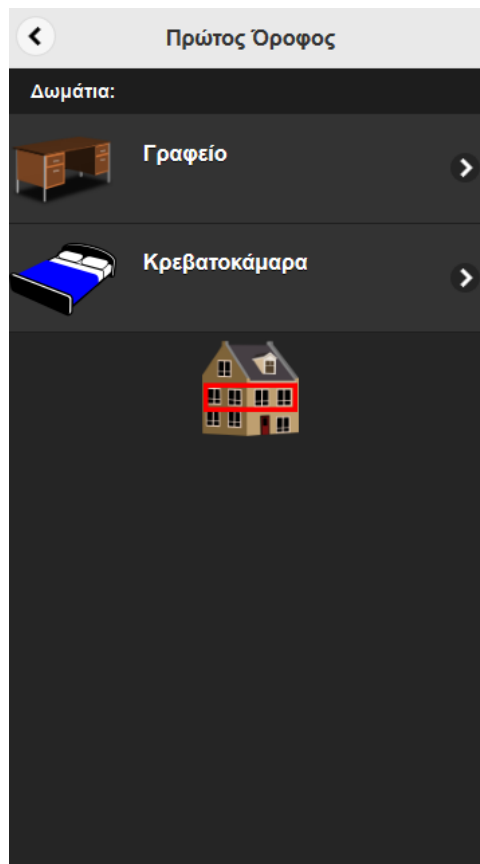
  <body>
    <div data-role="page" data-theme="b">

      <div data-role="header" data-position="fixed" data-theme="a" role="banner"
        class="ui-header ui-bar-a ui-header-fixed slidedown">
        <a href="mainmenu.html" rel="external" class="ui-btn-left ui-alt-icon
          ui-nodisc-icon ui-btn ui-icon-carat-l ui-btn-icon-notext ui-corner-all"
          data-role="button" role="button">Back</a>
        <h1 class="ui-title" role="heading" aria-level="1">Ισόγειο</h1>
      </div>

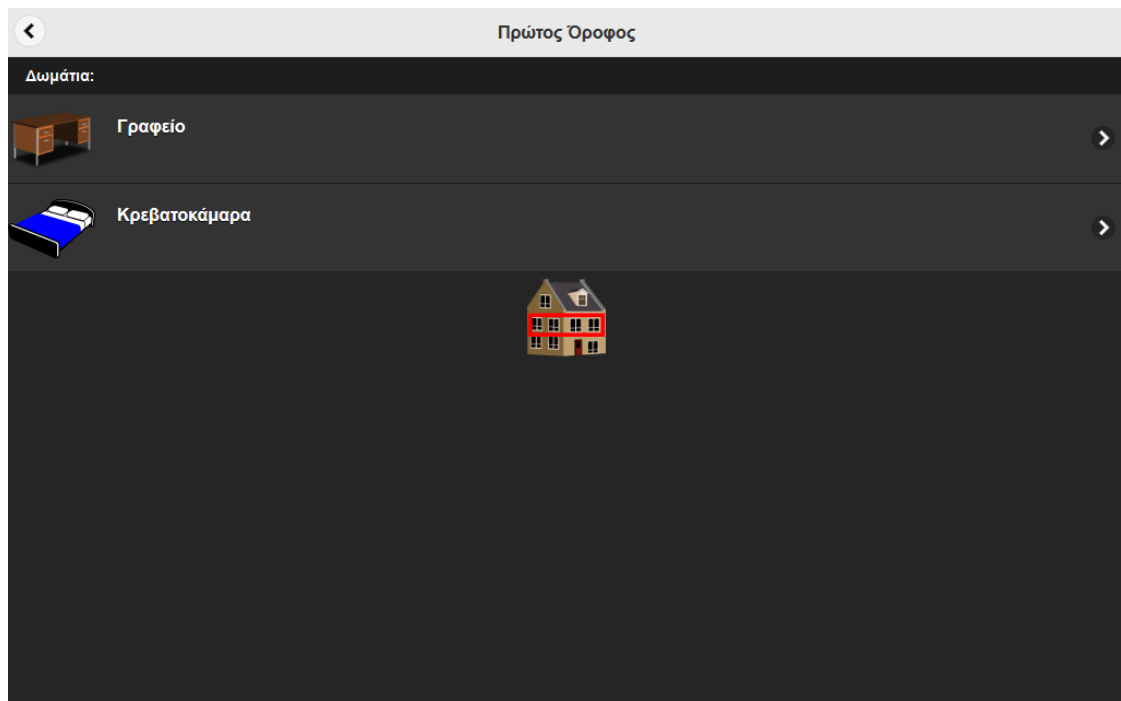
      <div data-role="main" class="ui-content">
        <ul data-role="listview">
          <li data-role="list-divider">Δωμάτια:</li>
          <li><a href="trapezaria.html" rel="external">
            <h4>Τραπεζαρία</h4></a></li>
          <li><a href="kouzina.html" rel="external">
            <h4>Κουζίνα</h4></a></li>
          <li><a href="saloni.html" rel="external">
            <h4>Σαλόνι</h4></a></li>
        </ul>
      </div>
      <center></center>
    </div>

  </body>
</html>
```

6.1.5 Στιγμιότυπα οθόνης Πρώτου Ορόφου εφαρμογής



Σχήμα 6-5. Διεπαφή Πρώτου ορόφου υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου



Σχήμα 6-6. Διεπαφή Πρώτου ορόφου υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή

6.1.6 Κώδικας οθόνης Πρώτου Ορόφου εφαρμογής

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="jquery.mobile-1.4.5.min.css" />
    <script src="jquery-1.11.1.min.js"></script>
    <script src="jquery.mobile-1.4.5.min.js"></script>
  </head>

  <body>
    <div data-role="page" data-theme="b">

      <div data-role="header" data-position="fixed" data-theme="a"
        role="banner" class="ui-header ui-bar-a ui-header-fixed slidedown">
        <a href="mainmenu.html" rel="external" class="ui-btn-left ui-alt-icon ui-
        nodisc-icon ui-btn ui-icon-carat-l ui-btn-icon-notext ui-corner-all" data-
        role="button" role="button">Back</a>
        <h1 class="ui-title" role="heading" aria-level="1">Πρώτος Όροφος</h1>
      </div>

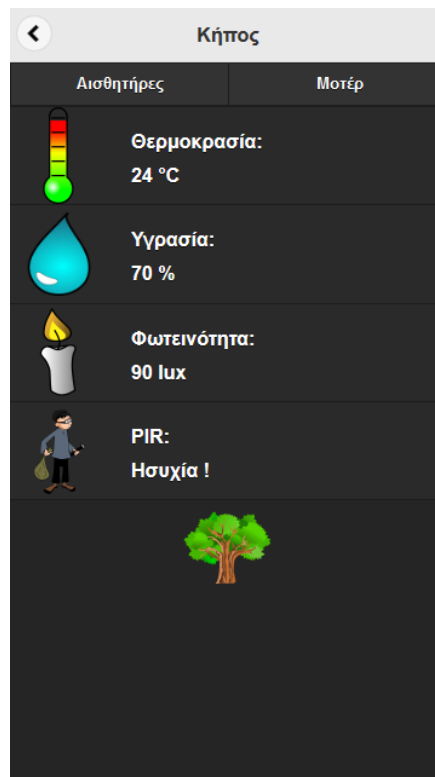
      <div data-role="main" class="ui-content">
        <ul data-role="listview">
          <li data-role="list-divider">Δωμάτια:</li>
          <li><a href="grafeio.html" rel="external">
            <h4>Γραφείο</h4></a></li>
          <li><a href="krevatokamara.html" rel="external">
            <h4>Κρεβατοκάμαρα</h4></a></li>
        </ul>
      </div>

      <center></center>

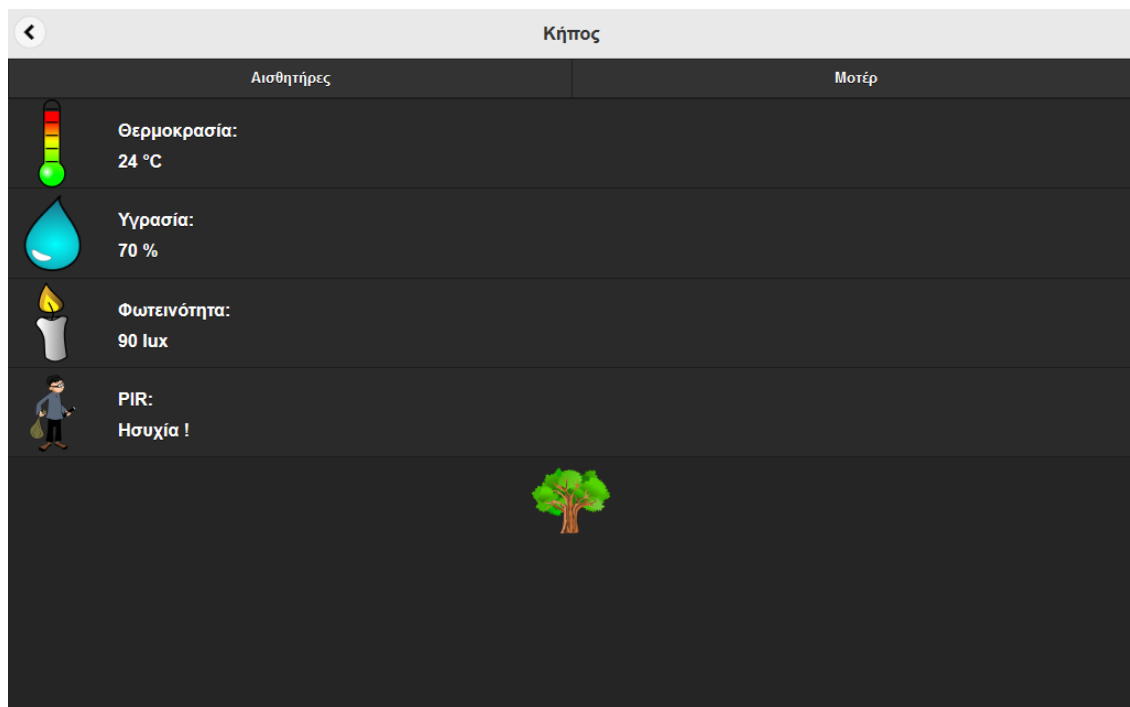
    </div>

  </body>
</html>
```


6.1.7 Στιγμιότυπα οθόνης μετρήσεων Κήπου εφαρμογής



Σχήμα 6-7. Διεπαφή μετρήσεων Κήπου υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου



Σχήμα 6-8. Διεπαφή μετρήσεων Κήπου υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή

6.1.8 Κώδικας οθόνης μετρήσεων Κήπου εφαρμογής

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="jquery.mobile-1.4.5.min.css" />
  <script src="jquery-1.11.1.min.js"></script>
  <script src="jquery.mobile-1.4.5.min.js"></script>

<script>

  function autoRefresh()
  {
    window.location = window.location.href;
  }
  setInterval('autoRefresh()', 5000); //5000ms

</script>
</head>

<body>
<div data-role="page" data-theme="b">

<div data-role="header" data-position="fixed" data-theme="a" role="banner"
  class="ui-header ui-bar-a ui-header-fixed slidedown">

  <a href="mainmenu.html" rel="external" class="ui-btn-left
  ui-alt-icon ui-nodisc-icon ui-btn ui-icon-carat-l ui-btn-icon-notext
  ui-corner-all" data-role="button" role="button">Back</a>

  <h1 class="ui-title" role="heading" aria-level="1">Κήπος</h1>
</div>

  <div data-role="navbar">

    <ul>
      <li><a href="khpos.html" rel="external">Αισθητήρες</a></li>
      <li><a href="khposmotors.html" rel="external">Μοτέρ</a></li>
    </ul>
  </div>
</div>
```

```
<div data-role="main" class="ui-content">
    <ul data-role="listview">
        <li><h4>Θερμοκρασία:</h4>
        <span id="temp_khpos"></span><b> °C</b></li>
        <li><h4>Υγρασία:</h4>
        <span id="hum_khpos"></span><b> %</b></li>
        <li><h4>Φωτεινότητα:</h4>
        <span id="fws_khpos"></span><b> lux</b></li>
        <li><h4>PIR:</h4>
        <span id="pir_khpos"></span><b> !</b></li>
    </ul>

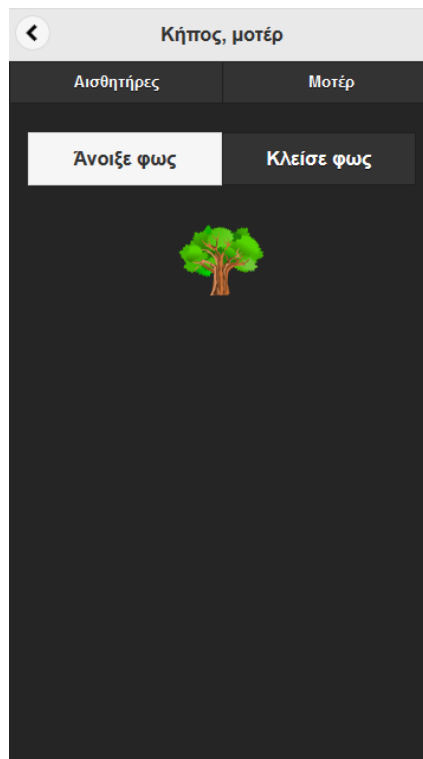
</div>

<center></center>
</div>

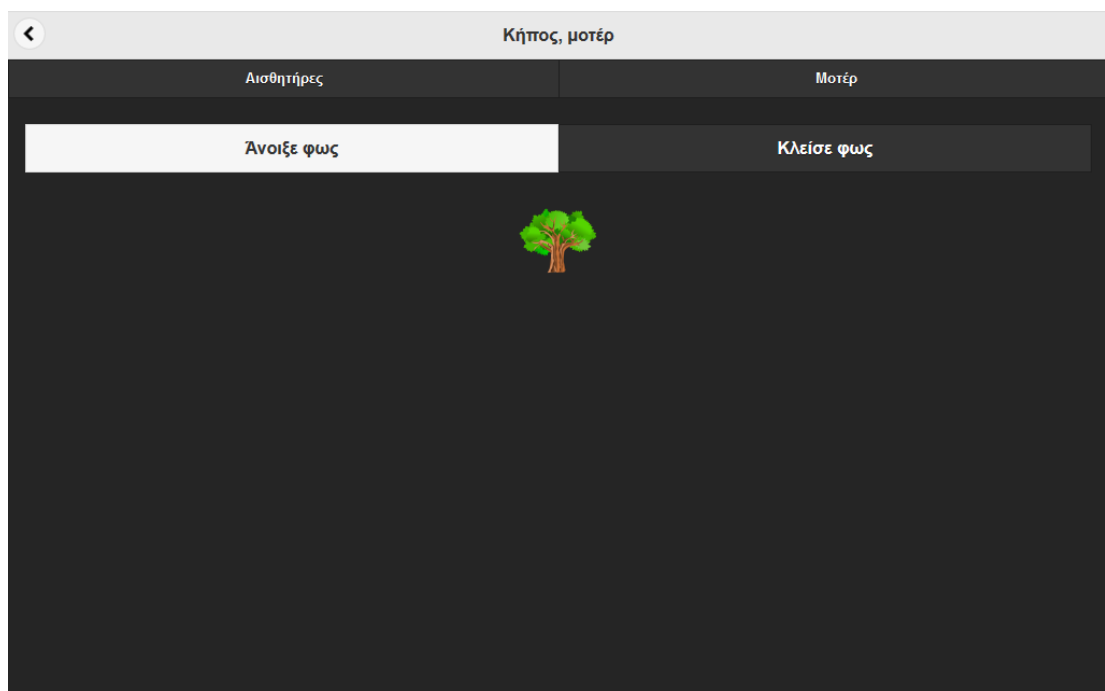
<script>
var xmlhttp, xmlDoc;
xmlhttp = new XMLHttpRequest();
xmlhttp.open("GET", "http://192.168.1.21", false);
xmlhttp.send();
xmlDoc = xmlhttp.responseXML;
document.getElementById("temp_khpos").innerHTML=xmlDoc.
getElementsByTagName("temp_khpos")[0].childNodes[0].nodeValue;
document.getElementById("hum_khpos").innerHTML=xmlDoc.
getElementsByTagName("hum_khpos")[0].childNodes[0].nodeValue;
document.getElementById("fws_khpos").innerHTML=xmlDoc.
getElementsByTagName("fws_khpos")[0].childNodes[0].nodeValue;
document.getElementById("pir_khpos").innerHTML=xmlDoc.
getElementsByTagName("pir_khpos")[0].childNodes[0].nodeValue;
</script>

</body>
</html>
```

6.1.9 Στιγμιότυπα οθόνης ενεργοποιητών Κήπου εφαρμογής



Σχήμα 6-9. Διεπαφή ενεργοποιητών Κήπου υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου



Σχήμα 6-10. Διεπαφή ενεργοποιητών Κήπου υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή

6.1.10 Κώδικας οθόνης ενεργοποιητών Κήπου εφαρμογής

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="jquery.mobile-1.4.5.min.css" />
  <script src="jquery-1.11.1.min.js"></script>
  <script src="jquery.mobile-1.4.5.min.js"></script>

  <script>
    $(document).ready(function() {
      $("#button0").click(function() {
        $.get("http://192.168.1.22/?afp", function(data, status) {
          });
        });
      $("#button1").click(function() {
        $.get("http://192.168.1.22/?kfp", function(data, status) {
          });
        });
      });
    </script>

</head>
<body>
<div data-role="page" data-theme="b">

<div data-role="header" data-position="fixed" data-theme="a" role="banner"
class="ui-header ui-bar-a ui-header-fixed slidedown">
<a href="mainmenu.html" rel="external" class="ui-btn-left ui-alt-icon ui-
nodisc-icon ui-btn ui-icon-carat-l ui-btn-icon-notext ui-corner-all"
data-role="button" role="button">Back</a>
<h1 class="ui-title" role="heading" aria-level="1">Κήπος, μοτέρ</h1>
</div>

<div data-role="navbar">
  <ul>
    <li><a href="khpos.html" rel="external" >Αισθητήρες</a></li>
    <li><a href="khposmotors.html" rel="external">Μοτέρ</a></li>
  </ul>
</div>
```

```
<div data-role="main" class="ui-content">

<fieldset class="ui-grid-a">
  <div class="ui-block-a">
    <button class="ui-btn ui-btn-a" id="button0">Άνοιξε φως</button></div>
    <div class="ui-block-b">
      <button class="ui-btn ui-btn-b" id="button1">Κλείσε φως</button></div>
  </fieldset>

</div>

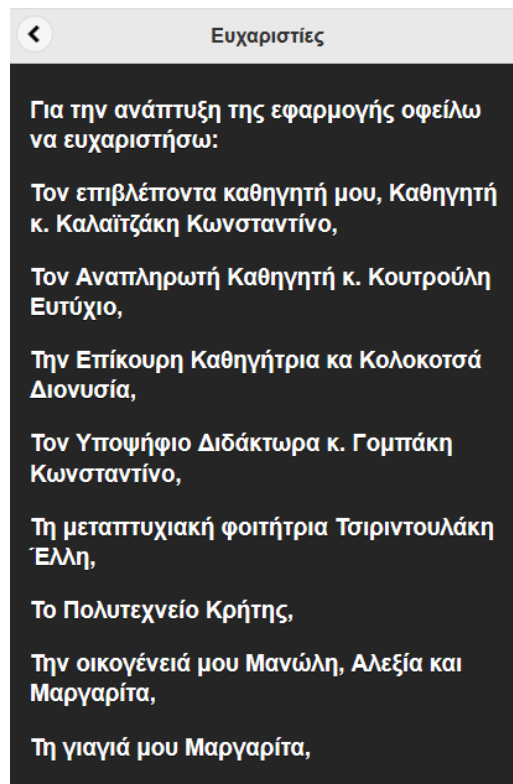
<center></center>

</div>

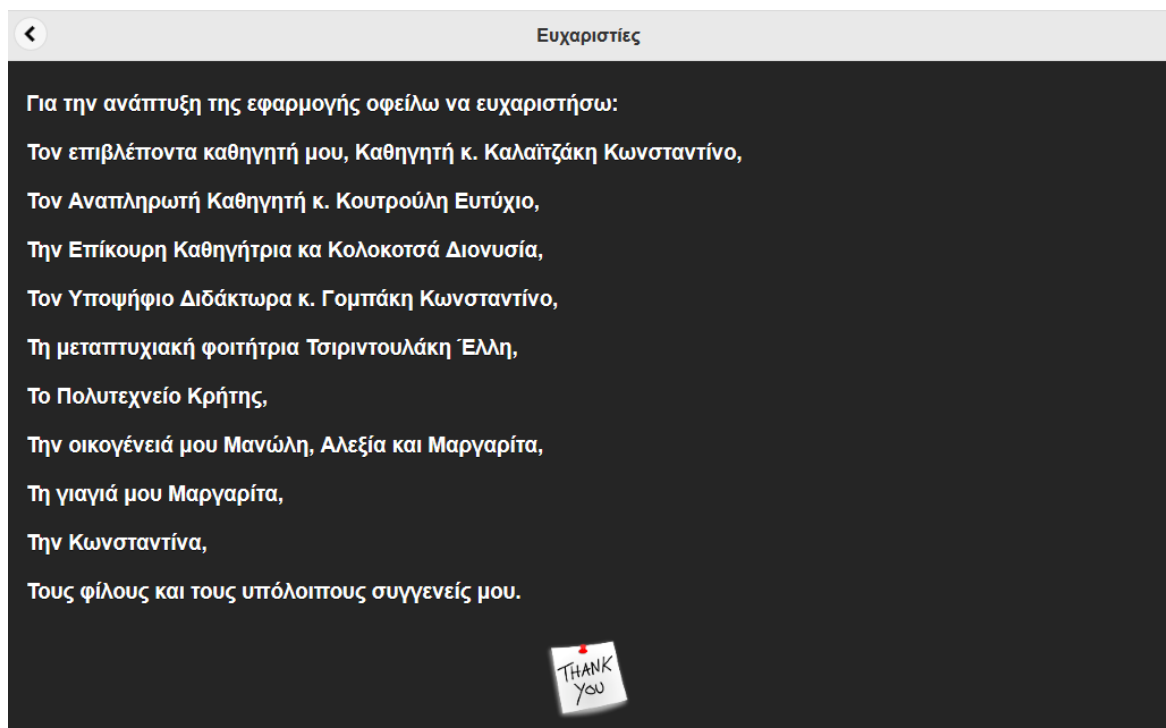
</body>

</html>
```

6.1.11 Στιγμιότυπα οθόνης Ευχαριστιών εφαρμογής



Σχήμα 6-11. Διεπαφή Ευχαριστιών υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου



Σχήμα 6-12. Διεπαφή Ευχαριστιών υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή

6.1.12 Κώδικας οθόνης Ευχαριστιών εφαρμογής

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="jquery.mobile-1.4.5.min.css" />
    <script src="jquery-1.11.1.min.js"></script>
    <script src="jquery.mobile-1.4.5.min.js"></script>
  </head>

  <body>
    <div data-role="page" data-theme="b">

      <div data-role="header" data-position="fixed" data-theme="a" role="banner"
        class="ui-header ui-bar-a ui-header-fixed slidedown">

        <a href="mainmenu.html" rel="external" class="ui-btn-left ui-alt-icon
          ui-nodisc-icon ui-btn ui-icon-carat-l ui-btn-icon-notext ui-corner-all"
          data-role="button" role="button">Back</a>

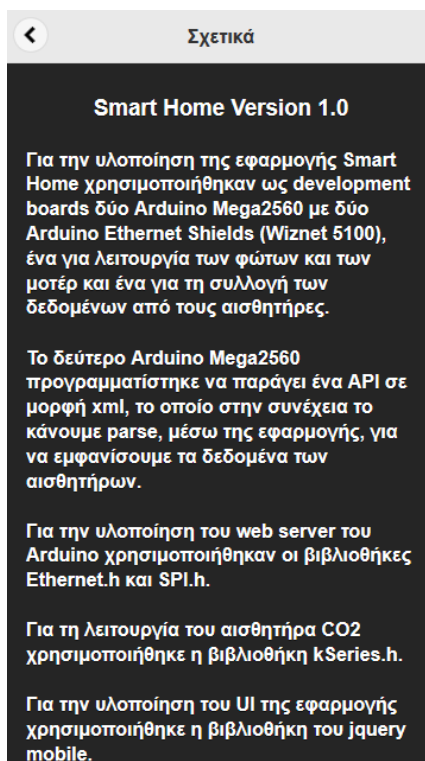
        <h1 class="ui-title" role="heading" aria-level="1">Ευχαριστίες</h1>
      </div>

      <div class="ui-body">
        <h3>Για την ανάπτυξη της εφαρμογής οφείλω να ευχαριστήσω:</h3>
        <h3>Τον επιβλέποντα καθηγητή μου, Καθηγητή κ. Καλαϊτζάκη
          Κωνσταντίνο,</h3>
        <h3>Τον Αναπληρωτή Καθηγητή κ. Κουτρούλη Ευτύχιο,</h3>
        <h3>Την Επίκουρη Καθηγήτρια κα Κολοκοτσά Διονυσία,</h3>
        <h3>Τον Υποψήφιο Διδάκτωρα κ. Γομπάκη Κωνσταντίνο,</h3>
        <h3>Τη μεταπτυχιακή φοιτήτρια Τσιριντουλάκη Έλλη,</h3>
        <h3>Το Πολυτεχνείο Κρήτης,</h3>
        <h3>Την οικογένειά μου Μανώλη, Αλεξία και Μαργαρίτα,</h3>
        <h3>Τη γιαγιά μου Μαργαρίτα,</h3>
        <h3>Την Κωνσταντίνα,</h3>
        <h3>Τους φίλους και τους υπόλοιπους συγγενείς μου.</h3>
      </div>

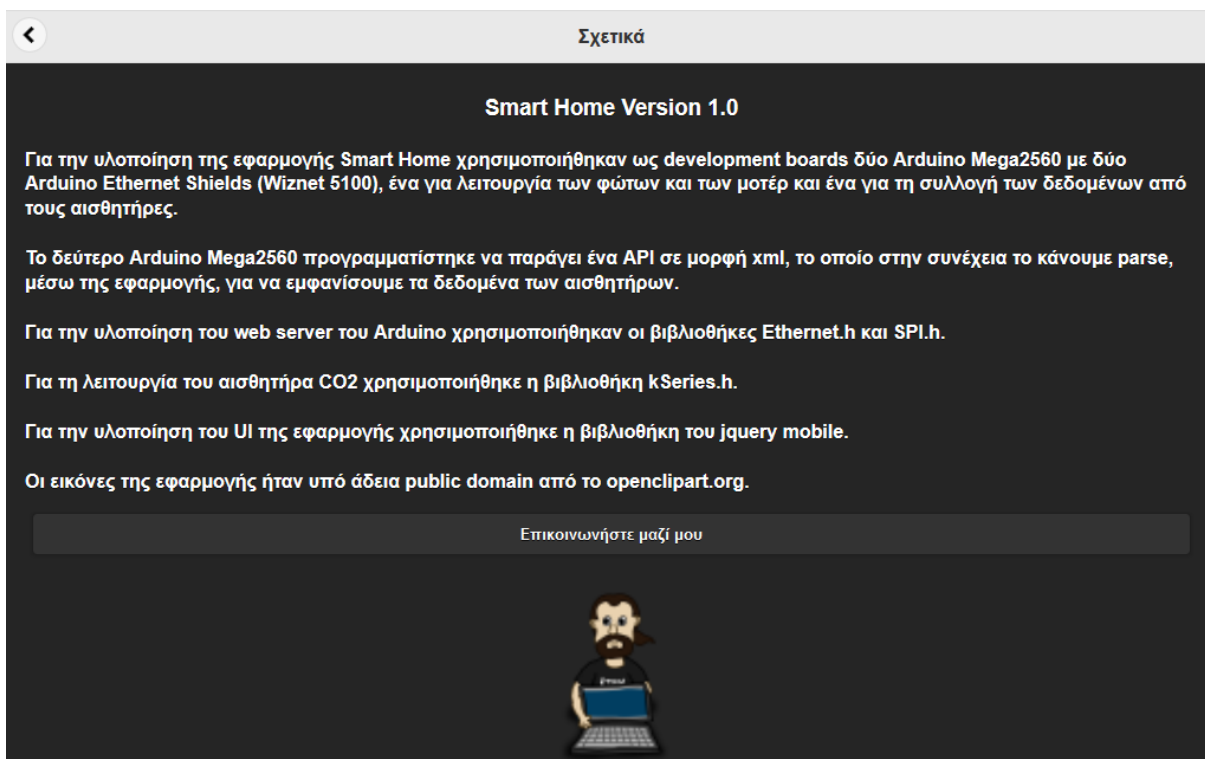
      <center></center>

    </div>
  </body>
</html>
```


6.1.13 Στιγμιότυπα οθόνης Σχετικά (About) εφαρμογής



Σχήμα 6-13. Διεπαφή Σχετικά (About) υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου



Σχήμα 6-14. Διεπαφή Σχετικά (About) υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή

6.1.14 Κώδικας οθόνης Σχετικά (About) εφαρμογής

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="jquery.mobile-1.4.5.min.css" />
    <script src="jquery-1.11.1.min.js"></script>
    <script src="jquery.mobile-1.4.5.min.js"></script>
  </head>
  <body>
    <div data-role="page" data-theme="b">

      <div data-role="header" data-position="fixed" data-theme="a" role="banner"
        class="ui-header ui-bar-a ui-header-fixed slidedown">
        <a href="mainmenu.html" rel="external" class="ui-btn-left ui-alt-icon ui-
        nodisc-icon ui-btn ui-icon-carat-l ui-btn-icon-notext ui-corner-all" data-
        role="button" role="button">Back</a>

        <h1 class="ui-title" role="heading" aria-level="1">Σχετικά</h1>
      </div>

      <div class="ui-body">
        <center><h3>Smart Home Version 1.0</h3></center>

        <h4>Για την υλοποίηση της εφαρμογής Smart Home χρησιμοποιήθηκαν
          ως development boards δύο Arduino Mega2560 με δύο Arduino
          Ethernet Shields (Wiznet 5100), ένα για λειτουργία των φώτων
          και των μοτέρ και ένα για τη συλλογή των δεδομένων από τους
          αισθητήρες.</h4>

        <h4>Το δεύτερο Arduino Mega2560 προγραμματίστηκε να παράγει ένα
          API σε μορφή xml, το οποίο στην συνέχεια το κάνουμε parse,
          μέσω της εφαρμογής, για να εμφανίσουμε τα δεδομένα των
          αισθητήρων.</h4>

        <h4>Για την υλοποίηση του web server του Arduino χρησιμοποιήθηκαν
          οι βιβλιοθήκες Ethernet.h και SPI.h.</h4>

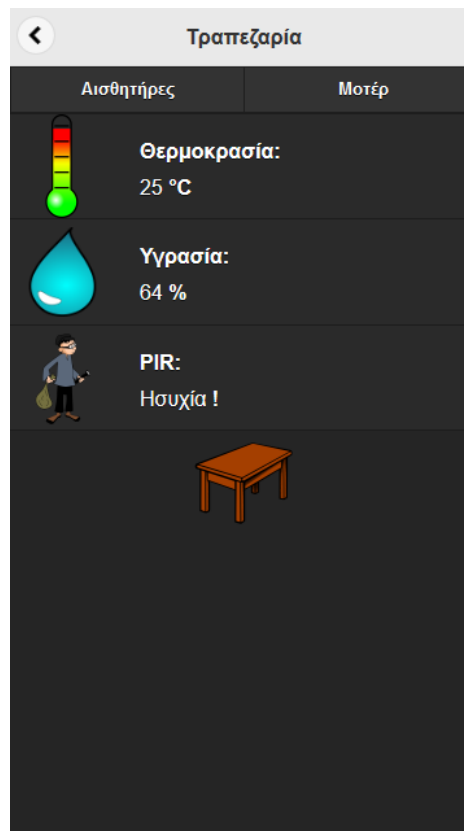
        <h4>Για τη λειτουργία του αισθητήρα CO2 χρησιμοποιήθηκε η
          βιβλιοθήκη kSeries.h.</h4>

        <h4>Για την υλοποίηση του UI της εφαρμογής χρησιμοποιήθηκε η
          βιβλιοθήκη του jquery mobile.</h4>

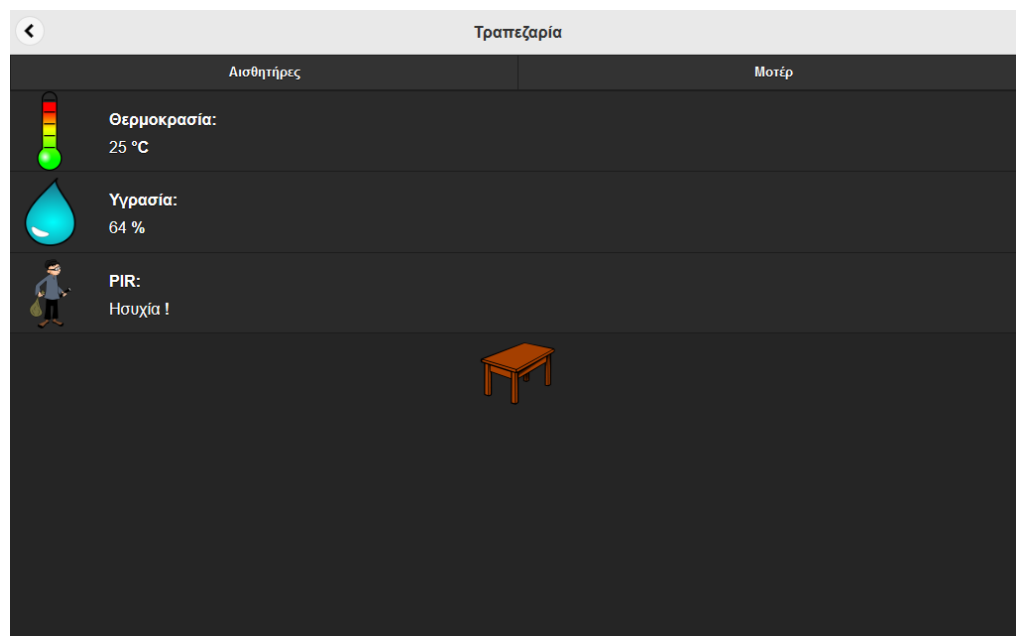
        <h4>Οι εικόνες της εφαρμογής ήταν υπό άδεια public domain από το
          openclipart.org.</h4>
      </div>

      <center></center>
    </div>
  </body>
</html>
```

6.1.15 Στιγμιότυπα οθόνης μετρήσεων Τραπεζαρίας εφαρμογής



Σχήμα 6-15. Διεπαφή μετρήσεων τραπεζαρίας υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου



Σχήμα 6-16. Διεπαφή μετρήσεων τραπεζαρίας υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή

6.1.16 Κώδικας οθόνης μετρήσεων Τραπεζαρίας εφαρμογής

```
<!DOCTYPE html>
<html>

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="jquery.mobile-1.4.5.min.css" />
  <script src="jquery-1.11.1.min.js"></script>
  <script src="jquery.mobile-1.4.5.min.js"></script>

<script>

  function autoRefresh()
  {
    window.location = window.location.href;
  }

  setInterval('autoRefresh()', 5000); // 5000ms, 5 δεφτερολεπτα

</script>

</head>

<body>

<div data-role="page" data-theme="b">

<div data-role="header" data-position="fixed" data-theme="a" role="banner"
class="ui-header ui-bar-a ui-header-fixed slidedown">
<a href="isogeio.html" rel="external" class="ui-btn-left ui-alt-icon
ui-nodisc-icon ui-btn ui-icon-carat-l ui-btn-icon-notext ui-corner-all"
data-role="button" role="button">Back</a>
<h1 class="ui-title" role="heading" aria-level="1">Τραπεζαρία</h1>
</div>

<div data-role="navbar">
  <ul>
    <li><a href="trapezaria.html" rel="external">Αισθητήρες</a></li>
    <li><a href="trapezariamotors.html" rel="external">Μοτέρ</a></li>
  </ul>
</div>
```

```

<div data-role="main" class="ui-content">

    <ul data-role="listview">
        <li><h4>Θερμοκρασία:</h4>
            <span id="temp_trapezaria"></span><b> °C</b></li>
        <li><h4>Υγρασία:</h4>
            <span id="hum_trapezaria"></span><b> %</b></li>
        <li><h4>PIR:</h4>
            <span id="pir_trapezaria"></span><b> !</b></li>
    </ul>

</div>

<center></center>

</div>

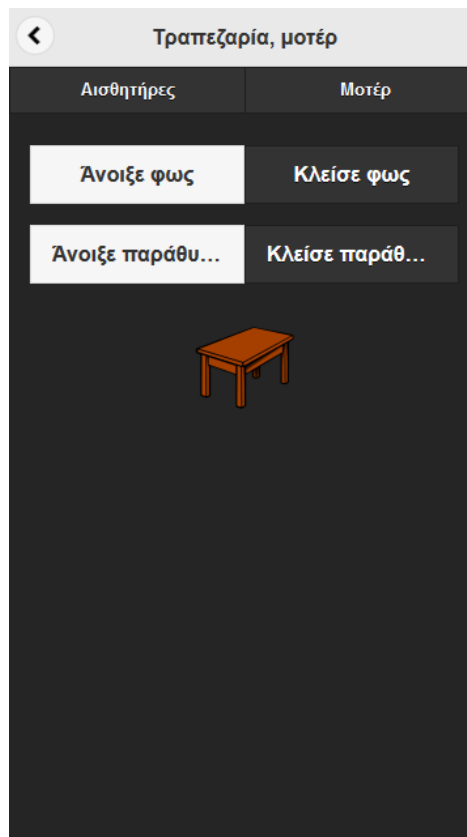
<script>
var xmlhttp, xmlDoc;
xmlhttp = new XMLHttpRequest();
xmlhttp.open("GET", "http://192.168.1.21", false);
xmlhttp.send();
xmlDoc = xmlhttp.responseXML;
document.getElementById("temp_trapezaria").innerHTML=xmlDoc.
getElementsByTagName("temp_trapezaria")[0].childNodes[0].nodeValue;
document.getElementById("hum_trapezaria").innerHTML=xmlDoc.
getElementsByTagName("hum_trapezaria")[0].childNodes[0].nodeValue;
document.getElementById("pir_trapezaria").innerHTML=xmlDoc.
getElementsByTagName("pir_trapezaria")[0].childNodes[0].nodeValue;
</script>

</body>

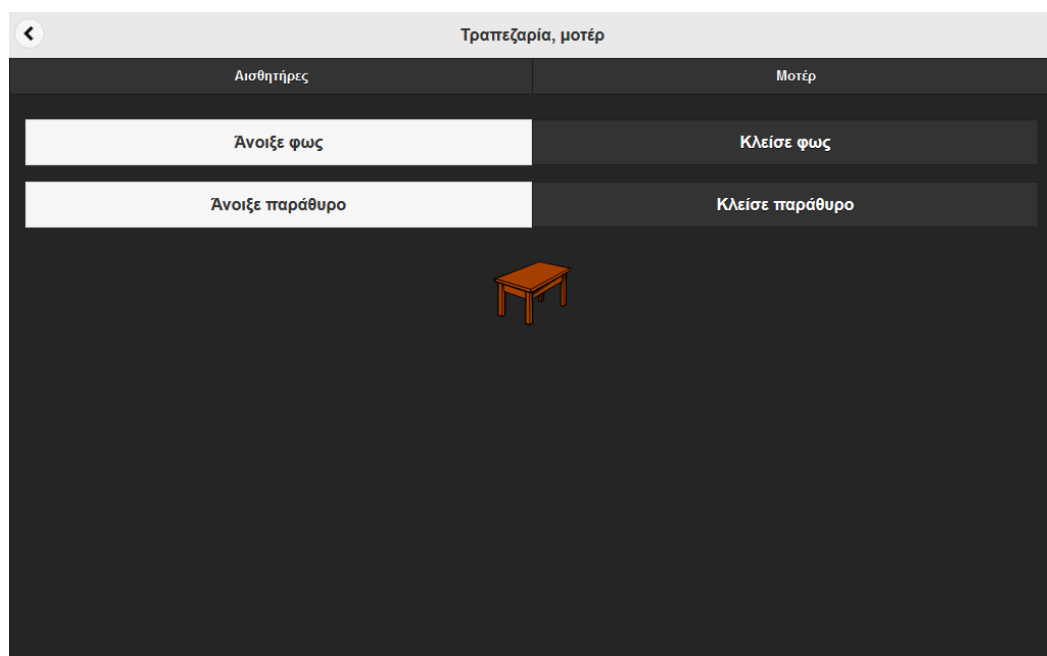
</html>

```

6.1.17 Στιγμιότυπα οθόνης ενεργοποιητών Τραπεζαρίας εφαρμογής



Σχήμα 6-17. Διεπαφή ενεργοποιητών τραπεζαρίας υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου



Σχήμα 6-18. Διεπαφή ενεργοποιητών τραπεζαρίας υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή

6.1.18 Κώδικας οθόνης ενεργοποιητών Τραπεζαρίας εφαρμογής

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="jquery.mobile-1.4.5.min.css" />
    <script src="jquery-1.11.1.min.js"></script>
    <script src="jquery.mobile-1.4.5.min.js"></script>
  </script>
  $(document).ready(function() {
    $("#button0").click(function() {
      $.get("http://192.168.1.22/?aft", function(data, status) {
      });
    });
    $("#button1").click(function() {
      $.get("http://192.168.1.22/?kft", function(data, status) {
      });
    });
    $("#button2").click(function() {
      $.get("http://192.168.1.22/?epat", function(data, status) {
      });
    });
    $("#button3").click(function() {
      $.get("http://192.168.1.22/?rpat", function(data, status) {
      });
    });
  });
</script>
</head>

<body>
<div data-role="page" data-theme="b">

<div data-role="header" data-position="fixed" data-theme="a" role="banner"
class="ui-header ui-bar-a ui-header-fixed slidedown">
<a href="isogeio.html" rel="external" class="ui-btn-left ui-alt-icon
ui-nodisc-icon ui-btn ui-icon-carat-l ui-btn-icon-notext ui-corner-all"
data-role="button" role="button">Back</a>
<h1 class="ui-title" role="heading" aria-level="1">Τραπεζαρία, μοτέρ</h1>
</div>
```

```
<div data-role="navbar">
  <ul>
    <li><a href="trapezaria.html" rel="external" >Αισθητήρες</a></li>
    <li><a href="trapezariamotors.html" rel="external">Μοτέρ</a></li>
  </ul>
</div>

<div data-role="main" class="ui-content">
  <fieldset class="ui-grid-a">
    <div class="ui-block-a"><button class="ui-btn ui-btn-a"
      id="button0">Άνοιξε φως</button></div>
    <div class="ui-block-b"><button class="ui-btn ui-btn-b"
      id="button1">Κλείσε φως</button></div>
    <div class="ui-block-a"><button class="ui-btn ui-btn-a"
      id="button2">Άνοιξε παράθυρο</button></div>
    <div class="ui-block-b"><button class="ui-btn ui-btn-b"
      id="button3">Κλείσε παράθυρο</button></div>
  </fieldset>
</div>

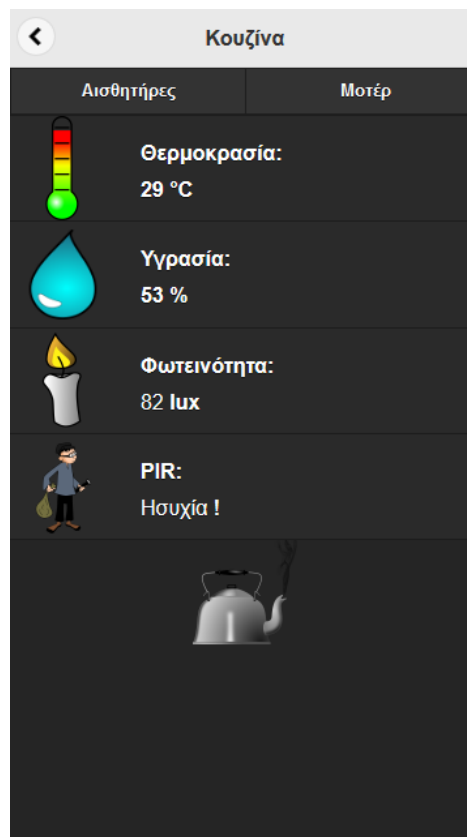
<center></center>

</div>

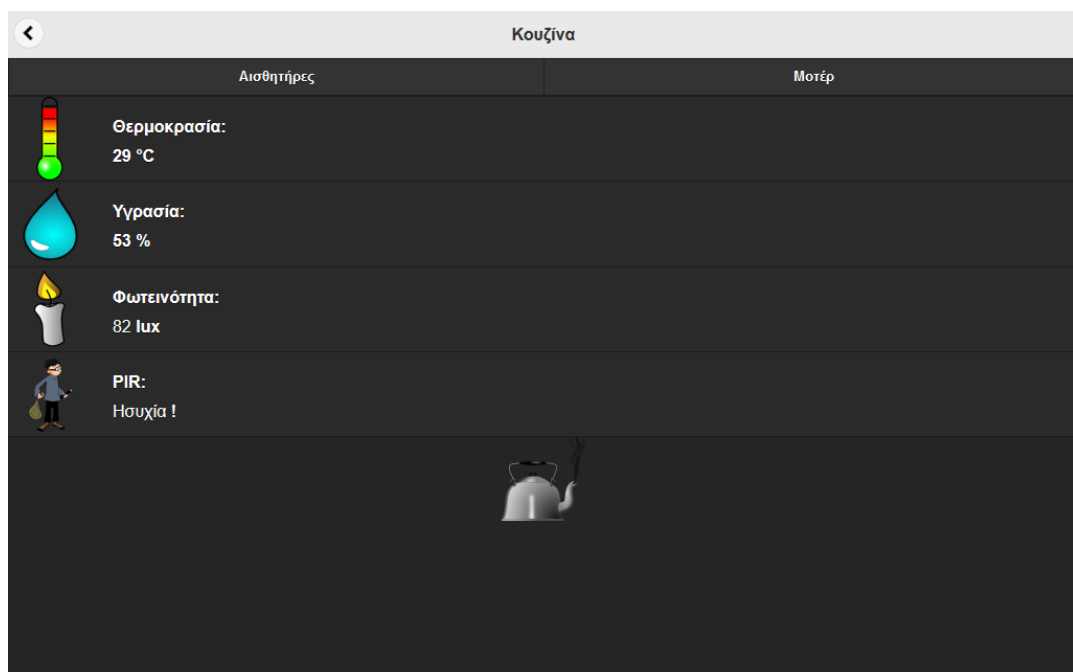
</body>

</html>
```


6.1.19 Στιγμιότυπα οθόνης μετρήσεων Κουζίνας εφαρμογής



Σχήμα 6-19. Διεπαφή μετρήσεων Κουζίνας υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου



Σχήμα 6-20. Διεπαφή μετρήσεων Κουζίνας υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή

6.1.20 Κώδικας οθόνης μετρήσεων Κουζίνας εφαρμογής

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="jquery.mobile-1.4.5.min.css" />
    <script src="jquery-1.11.1.min.js"></script>
    <script src="jquery.mobile-1.4.5.min.js"></script>

  <script>

    function autoRefresh()
    {
      window.location = window.location.href;
    }
    setInterval('autoRefresh()', 5000); // 5000ms, 5 defterolepta

  </script>

</head>

<body>

  <div data-role="page" data-theme="b">

    <div data-role="header" data-position="fixed" data-theme="a" role="banner"
      class="ui-header ui-bar-a ui-header-fixed slidedown">
      <a href="isogeio.html" rel="external" class="ui-btn-left ui-alt-icon ui-
        nodisc-icon ui-btn ui-icon-carat-l ui-btn-icon-notext ui-corner-all" data-
        role="button" role="button">Back</a>
      <h1 class="ui-title" role="heading" aria-level="1">Κουζίνα</h1>
    </div>

    <div data-role="navbar">
      <ul>
        <li><a href="kouzina.html" rel="external">Αισθητήρες</a></li>
        <li><a href="kouzinamotors.html" rel="external">Μοτέρ</a></li>
      </ul>
    </div>

  </div>
```

```
<div data-role="main" class="ui-content">
  <ul data-role="listview">
    <li><h4>Θερμοκρασία:</h4>
    <span id="temp_kouzina"></span><b> °C</b></li>
    <li><h4>Υγρασία:</h4>
    <span id="hum_kouzina"></span><b> %</b></li>
    <li><h4>Φωτεινότητα:</h4>
    <span id="fws_kouzina"></span><b> lux</b></li>
    <li><h4>PIR:</h4>
    <span id="pir_kouzina"></span><b> !</b></li>
  </ul>
</div>

<center></center>

</div>

<script>

var xmlhttp, xmlDoc;
xmlhttp = new XMLHttpRequest();
xmlhttp.open("GET", "http://192.168.1.21", false);
xmlhttp.send();
xmlDoc = xmlhttp.responseXML;

document.getElementById("temp_kouzina").innerHTML=xmlDoc.
getElementsByTagName("temp_kouzina")[0].childNodes[0].nodeValue;

document.getElementById("hum_kouzina").innerHTML=xmlDoc.
getElementsByTagName("hum_kouzina")[0].childNodes[0].nodeValue;

document.getElementById("fws_kouzina").innerHTML=xmlDoc.
getElementsByTagName("fws_kouzina")[0].childNodes[0].nodeValue;

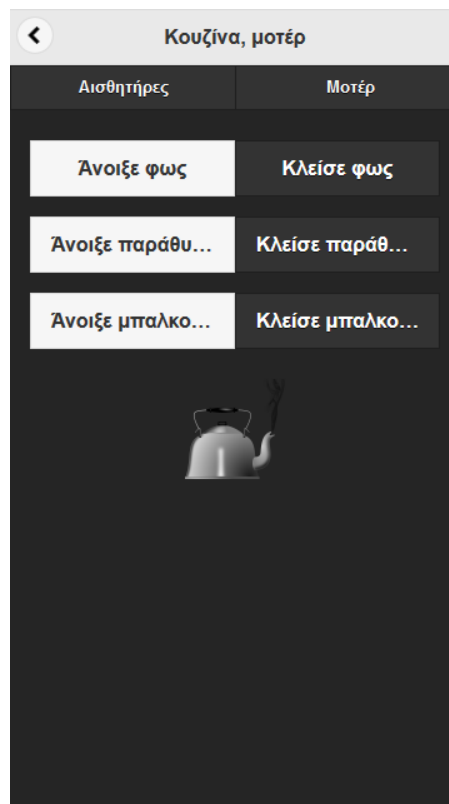
document.getElementById("pir_kouzina").innerHTML=xmlDoc.
getElementsByTagName("pir_kouzina")[0].childNodes[0].nodeValue;

</script>

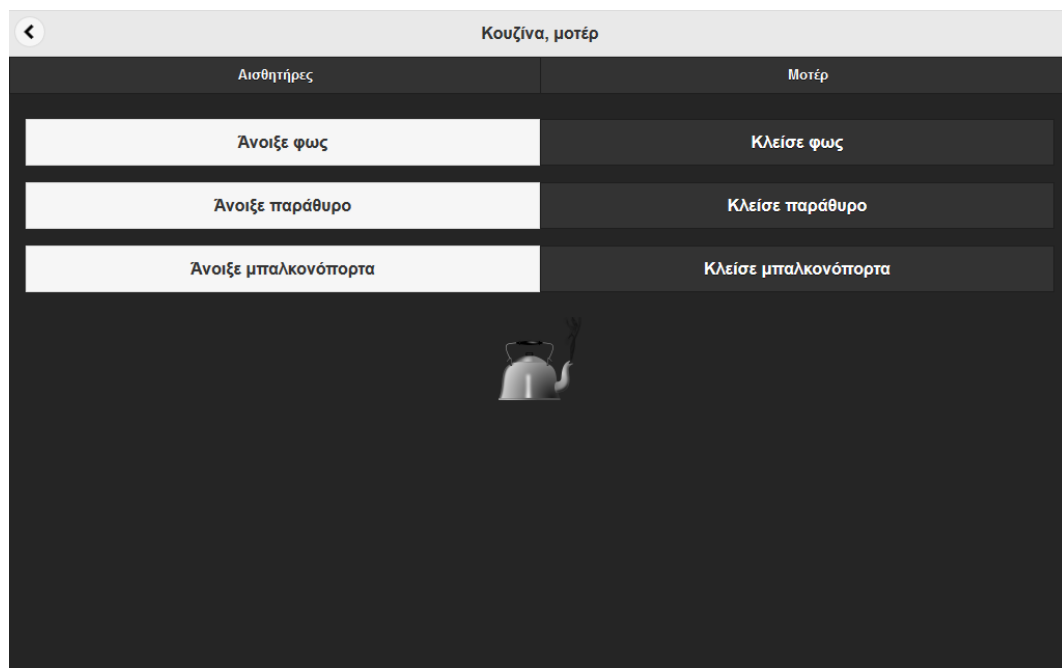
</body>

</html>
```

6.1.21 Στιγμιότυπα οθόνης ενεργοποιητών Κουζίνας εφαρμογής



Σχήμα 6-21. Διεπαφή ενεργοποιητών Κουζίνας υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου



Σχήμα 6-22. Διεπαφή ενεργοποιητών Κουζίνας υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή

6.1.22 Κώδικας οθόνης ενεργοποιητών Κουζίνας εφαρμογής

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="jquery.mobile-1.4.5.min.css" />
    <script src="jquery-1.11.1.min.js"></script>
    <script src="jquery.mobile-1.4.5.min.js"></script>
  </script>
  $(document).ready(function() {
    $("#button0").click(function() {
      $.get("http://192.168.1.22/?afk", function(data, status) {
      });
    });
    $("#button1").click(function() {
      $.get("http://192.168.1.22/?kfk", function(data, status) {
      });
    });
    $("#button2").click(function() {
      $.get("http://192.168.1.22/?epak", function(data, status) {
      });
    });
    $("#button3").click(function() {
      $.get("http://192.168.1.22/?rpak", function(data, status) {
      });
    });
    $("#button4").click(function() {
      $.get("http://192.168.1.22/?empk", function(data, status) {
      });
    });
    $("#button5").click(function() {
      $.get("http://192.168.1.22/?rmpk", function(data, status) {
      });
    });
  });
</script>
</head>
<body>
```

```

<div data-role="page" data-theme="b">

<div data-role="header" data-position="fixed" data-theme="a" role="banner"
class="ui-header ui-bar-a ui-header-fixed slidedown">

<a href="isogeio.html" rel="external" class="ui-btn-left ui-alt-icon
ui-nodisc-icon ui-btn ui-icon-carat-l ui-btn-icon-notext ui-corner-all"
data-role="button" role="button">Back</a>

<h1 class="ui-title" role="heading" aria-level="1">Κουζίνα, μοτέρ</h1>
</div>

<div data-role="navbar">
    <ul>
        <li><a href="kouzina.html" rel="external" >Αισθητήρες</a></li>
        <li><a href="kouzinamotors.html" rel="external">Μοτέρ</a></li>
    </ul>
</div>

<div data-role="main" class="ui-content">
    <fieldset class="ui-grid-a">
        <div class="ui-block-a"><button class="ui-btn ui-btn-a"
id="button0">Άνοιξε φως</button></div>
        <div class="ui-block-b"><button class="ui-btn ui-btn-b"
id="button1">Κλείσε φως</button></div>
        <div class="ui-block-a"><button class="ui-btn ui-btn-a"
id="button2">Άνοιξε παράθυρο</button></div>
        <div class="ui-block-b"><button class="ui-btn ui-btn-b"
id="button3">Κλείσε παράθυρο</button></div>
        <div class="ui-block-a"><button class="ui-btn ui-btn-a"
id="button4">Άνοιξε μπαλκονόπορτα</button></div>
        <div class="ui-block-b"><button class="ui-btn ui-btn-b"
id="button5">Κλείσε μπαλκονόπορτα</button></div>
    </fieldset>
</div>
<center></center>

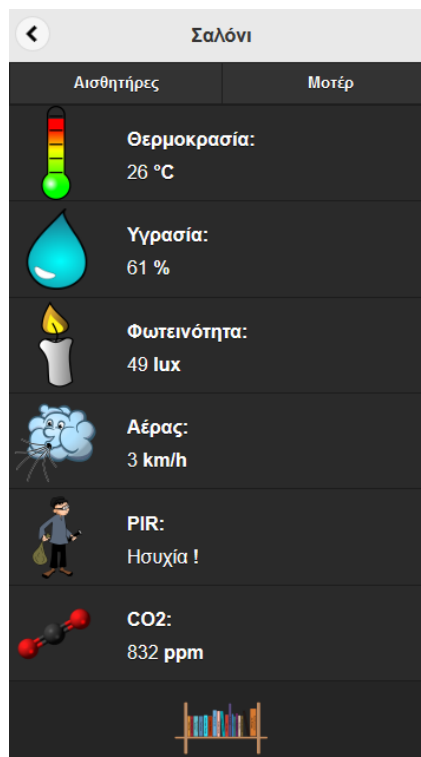
</div>

</body>

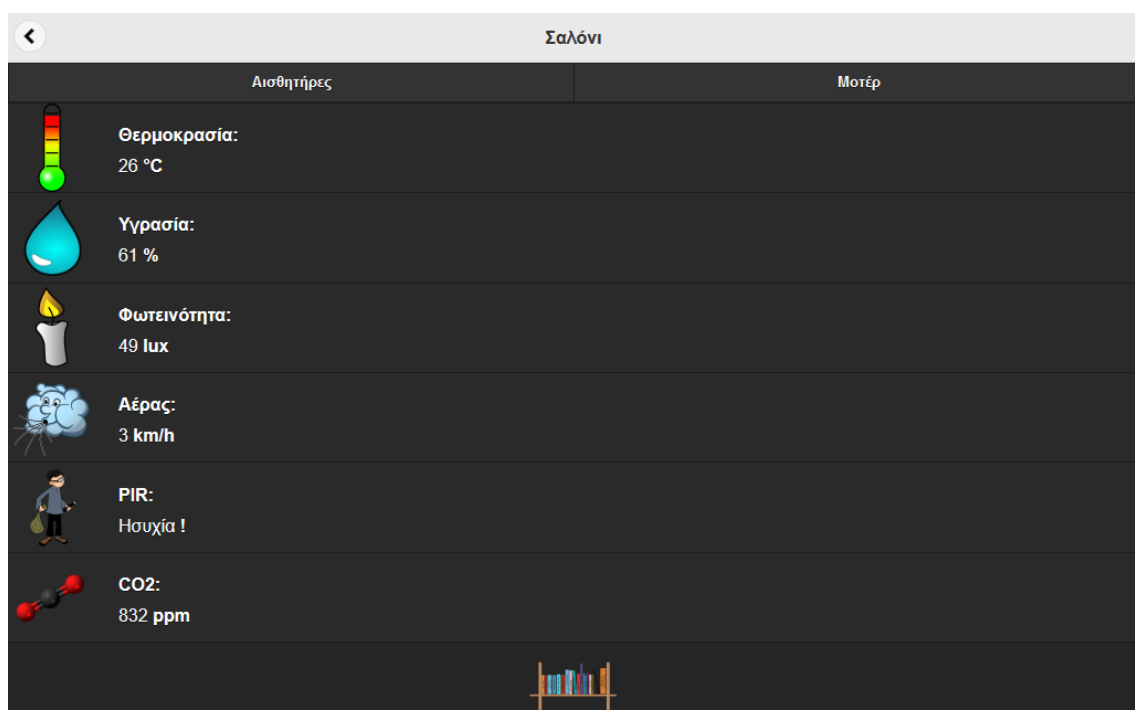
</html>

```

6.1.23 Στιγμιότυπα οθόνης μετρήσεων Σαλονιού εφαρμογής



Σχήμα 6-23. Διεπαφή μετρήσεων Σαλονιού υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου



Σχήμα 6-24. Διεπαφή μετρήσεων Σαλονιού υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή

6.1.24 Κώδικας οθόνης μετρήσεων Σαλονιού εφαρμογής

```
<!DOCTYPE html>

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="jquery.mobile-1.4.5.min.css" />
  <script src="jquery-1.11.1.min.js"></script>
  <script src="jquery.mobile-1.4.5.min.js"></script>

<script>
  function autoRefresh()
  {
    window.location = window.location.href;
  }

  setInterval('autoRefresh()', 5000); // 5000ms, 5 defterolepta
</script>

</head>

<body>

<div data-role="page" data-theme="b">

<div data-role="header" data-position="fixed" data-theme="a" role="banner"
class="ui-header ui-bar-a ui-header-fixed slidedown">
<a href="isogeio.html" rel="external" class="ui-btn-left ui-alt-icon ui-
nodisc-icon ui-btn ui-icon-carat-l ui-btn-icon-notext ui-corner-all" data-
role="button" role="button">Back</a>
<h1 class="ui-title" role="heading" aria-level="1">Σαλόνι</h1>
</div>

<div data-role="navbar">
  <ul>
    <li><a href="saloni.html" rel="external">Αισθητήρες</a></li>
    <li><a href="salonimotors.html" rel="external">Μοτέρ</a></li>
  </ul>
</div>
```



```

<div data-role="main" class="ui-content">
    <ul data-role="listview">
        <li><h4>Θερμοκρασία:</h4>
        <span id="temp_saloni"></span><b> °C</b></li>
        <li><h4>Υγρασία:</h4>
        <span id="hum_saloni"></span><b> %</b></li>
        <li><h4>Φωτεινότητα:</h4>
        <span id="fws_saloni"></span><b> lux</b></li>
        <li><h4>Αέρας:</h4>
        <span id="wind"></span><b> km/h</b></li>
        <li><h4>PIR:</h4>
        <span id="pir_saloni"></span><b> !</b></li>
        <li><h4>CO2:</h4>
        <span id="co2"></span><b> ppm</b></li>
    </ul>
</div>

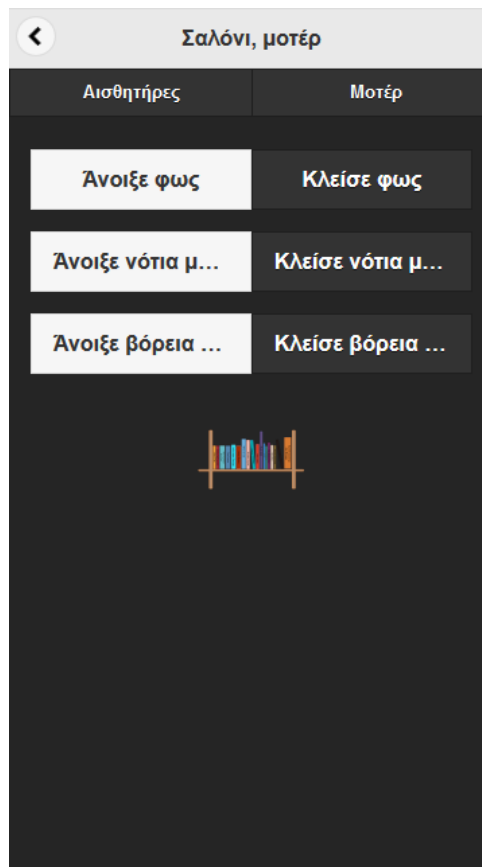
<center></center>
</div>

<script>
var xmlhttp, xmlDoc;
xmlhttp = new XMLHttpRequest();
xmlhttp.open("GET", "http://192.168.1.21", false);
xmlhttp.send();
xmlDoc = xmlhttp.responseXML;
document.getElementById("temp_saloni").innerHTML=xmlDoc.
getElementsByTagName("temp_saloni")[0].childNodes[0].nodeValue;
document.getElementById("hum_saloni").innerHTML=xmlDoc.
getElementsByTagName("hum_saloni")[0].childNodes[0].nodeValue;
document.getElementById("fws_saloni").innerHTML=xmlDoc.
getElementsByTagName("fws_saloni")[0].childNodes[0].nodeValue;
document.getElementById("wind").innerHTML=xmlDoc.
getElementsByTagName("wind")[0].childNodes[0].nodeValue;
document.getElementById("pir_saloni").innerHTML=xmlDoc.
getElementsByTagName("pir_saloni")[0].childNodes[0].nodeValue;
document.getElementById("co2").innerHTML=xmlDoc.
getElementsByTagName("co2")[0].childNodes[0].nodeValue;
</script>

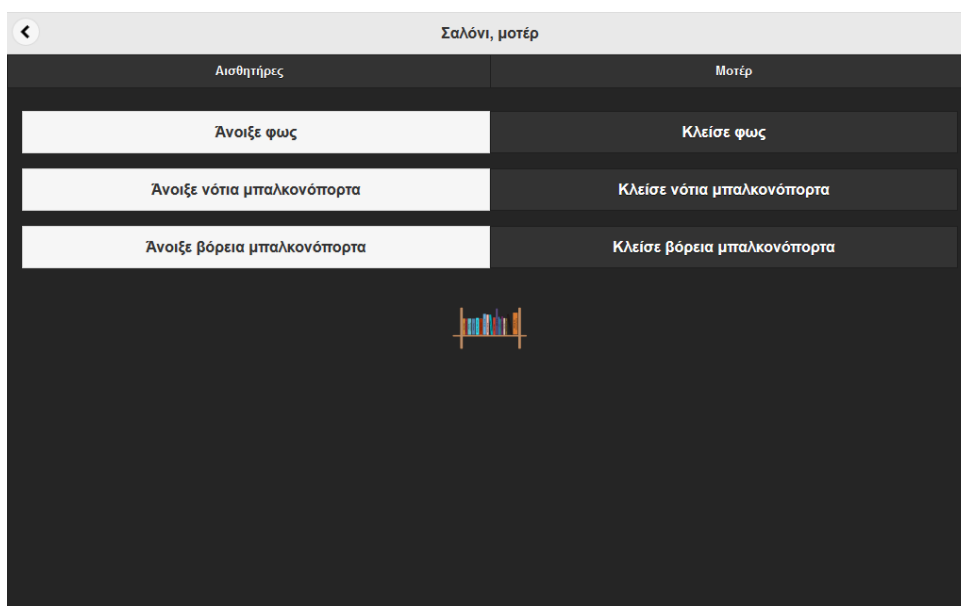
</body>
</html>

```

6.1.25 Στιγμιότυπα οθόνης ενεργοποιητών Σαλονιού εφαρμογής



Σχήμα 6-25. Διεπαφή ενεργοποιητών Σαλονιού υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου



Σχήμα 6-26. Διεπαφή ενεργοποιητών Σαλονιού υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή

6.1.26 Κώδικας οθόνης ενεργοποιητών Σαλονιού εφαρμογής

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="jquery.mobile-1.4.5.min.css" />
    <script src="jquery-1.11.1.min.js"></script>
    <script src="jquery.mobile-1.4.5.min.js"></script>
  </script>
  $(document).ready(function() {
    $("#button0").click(function() {
      $.get("http://192.168.1.22/?afs", function(data, status) {
      });
    });
    $("#button1").click(function() {
      $.get("http://192.168.1.22/?kfs", function(data, status) {
      });
    });
    $("#button2").click(function() {
      $.get("http://192.168.1.22/?empns", function(data, status) {
      });
    });
    $("#button3").click(function() {
      $.get("http://192.168.1.22/?rmpns", function(data, status) {
      });
    });
    $("#button4").click(function() {
      $.get("http://192.168.1.22/?empvs", function(data, status) {
      });
    });
    $("#button5").click(function() {
      $.get("http://192.168.1.22/?rmpvs", function(data, status) {
      });
    });
  });
</script>
</head>
```

```

<body>
<div data-role="page" data-theme="b">

<div data-role="header" data-position="fixed" data-theme="a" role="banner"
class="ui-header ui-bar-a ui-header-fixed slidedown">

<a href="isogeio.html" rel="external" class="ui-btn-left ui-alt-icon
ui-nodisc-icon ui-btn ui-icon-carat-l ui-btn-icon-notext ui-corner-all"
data-role="button" role="button">Back</a>

<h1 class="ui-title" role="heading" aria-level="1">Σαλόνι, μοτέρ</h1>
</div>

<div data-role="navbar">
    <ul>
        <li><a href="saloni.html" rel="external" >Αισθητήρες</a></li>
        <li><a href="salonimotors.html" rel="external">Μοτέρ</a></li>
    </ul>
</div>

<div data-role="main" class="ui-content">
    <fieldset class="ui-grid-a">
        <div class="ui-block-a"><button class="ui-btn ui-btn-a"
id="button0">Άνοιξε φως</button></div>
        <div class="ui-block-b"><button class="ui-btn ui-btn-b"
id="button1">Κλείσε φως</button></div>
        <div class="ui-block-a"><button class="ui-btn ui-btn-a"
id="button2">Άνοιξε νότια μπαλκονόπορτα</button></div>
        <div class="ui-block-b"><button class="ui-btn ui-btn-b"
id="button3">Κλείσε νότια μπαλκονόπορτα</button></div>
        <div class="ui-block-a"><button class="ui-btn ui-btn-a"
id="button4">Άνοιξε βόρεια μπαλκονόπορτα</button></div>
        <div class="ui-block-b"><button class="ui-btn ui-btn-b"
id="button5">Κλείσε βόρεια μπαλκονόπορτα</button></div>
    </fieldset>
</div>
    <center></center>

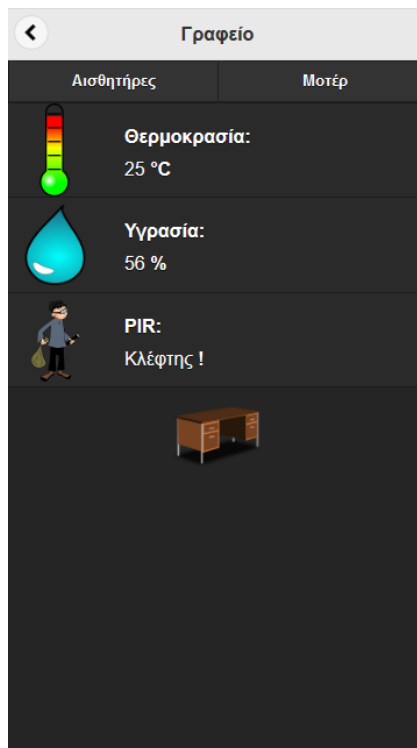
</div>

</body>

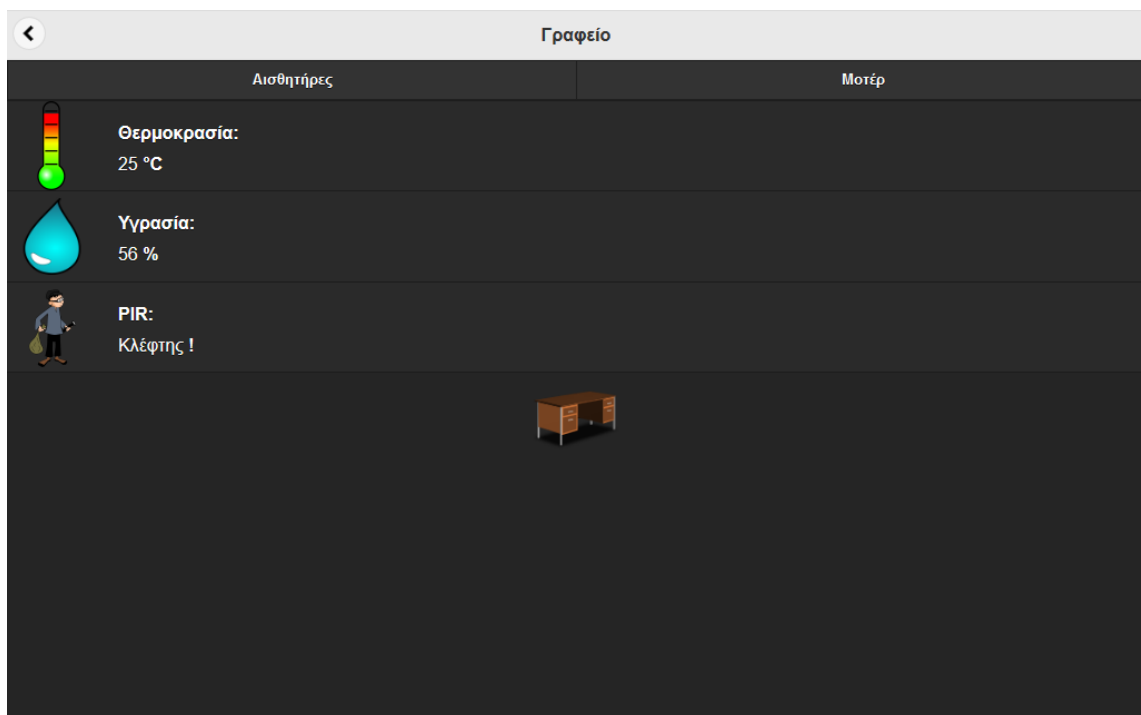
</html>

```

6.1.27 Στιγμιότυπα οθόνης μετρήσεων Γραφείου εφαρμογής



Σχήμα 6-27. Διεπαφή μετρήσεων Γραφείου υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου



Σχήμα 6-28. Διεπαφή μετρήσεων Γραφείου υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή

6.1.28 Κώδικας οθόνης μετρήσεων Γραφείου εφαρμογής

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="jquery.mobile-1.4.5.min.css" />
    <script src="jquery-1.11.1.min.js"></script>
    <script src="jquery.mobile-1.4.5.min.js"></script>

<script>

  function autoRefresh()
  {
    window.location = window.location.href;
  }

  setInterval('autoRefresh()', 5000); // 5000ms, 5defterolepta
</script>

</head>

<body>
<div data-role="page" data-theme="b">

  <div data-role="header" data-position="fixed" data-theme="a" role="banner"
  class="ui-header ui-bar-a ui-header-fixed slidedown">
    <a href="prwtosorofos.html" rel="external" class="ui-btn-left ui-alt-icon
    ui-nodisc-icon ui-btn ui-icon-carat-l ui-btn-icon-notext ui-corner-all"
    data-role="button" role="button">Back</a>
    <h1 class="ui-title" role="heading" aria-level="1">Γραφείο</h1>
  </div>

  <div data-role="navbar">
    <ul>
      <li><a href="grafeio.html" rel="external">Αισθητήρες</a></li>
      <li><a href="grafeiomotors.html" rel="external">Μοτέρ</a></li>
    </ul>
  </div>
</div>
```

```

<div data-role="main" class="ui-content">
    <ul data-role="listview">
        <li><h4>Θερμοκρασία:</h4>
        <span id="temp_grafeio"></span><b> °C</b></li>
        <li><h4>Υγρασία:</h4>
        <span id="hum_grafeio"></span><b> %</b></li>
        <li><h4>PIR:</h4>
        <span id="pir_grafeio"></span><b> !</b></li>
    </ul>

</div>

    <center></center>
</div>

<script>

var xmlhttp, xmlDoc;
xmlhttp = new XMLHttpRequest();
xmlhttp.open("GET", "http://192.168.1.21", false);
xmlhttp.send();
xmlDoc = xmlhttp.responseXML;
document.getElementById("temp_grafeio").innerHTML=xmlDoc.
getElementsByTagName("temp_grafeio")[0].childNodes[0].nodeValue;
document.getElementById("hum_grafeio").innerHTML=xmlDoc.
getElementsByTagName("hum_grafeio")[0].childNodes[0].nodeValue;
document.getElementById("pir_grafeio").innerHTML=xmlDoc.
getElementsByTagName("pir_grafeio")[0].childNodes[0].nodeValue;

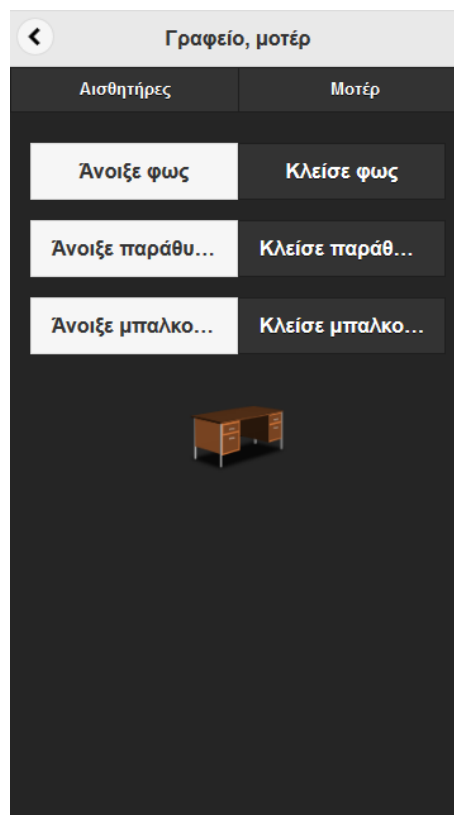
</script>

</body>

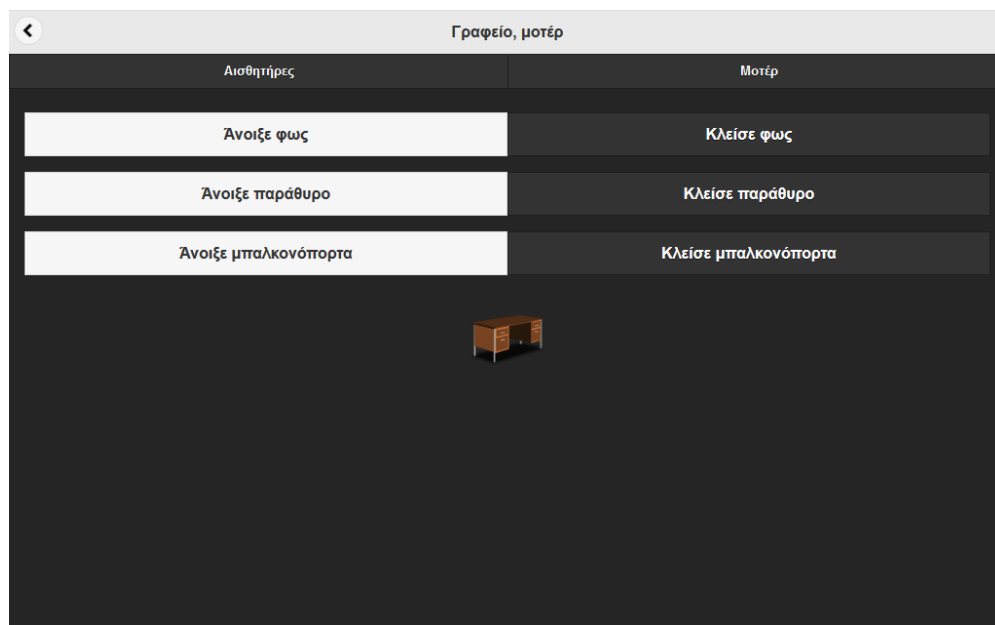
</html>

```

6.1.29 Στιγμιότυπα οθόνης ενεργοποιητών Γραφείου εφαρμογής



Σχήμα 6-29. Διεπαφή ενεργοποιητών Γραφείου υλοποιηθήσας εφαρμογής σε οθόνη κινητού τηλεφώνου



Σχήμα 6-30. Διεπαφή ενεργοποιητών Γραφείου υλοποιηθήσας εφαρμογής σε οθόνη tablet/υπολογιστή

6.1.30 Κώδικας οθόνης ενεργοποιητών Γραφείου εφαρμογής

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="jquery.mobile-1.4.5.min.css" />
    <script src="jquery-1.11.1.min.js"></script>
    <script src="jquery.mobile-1.4.5.min.js"></script>
  </script>

  $(document).ready(function() {
    $("#button0").click(function() {
      $.get("http://192.168.1.22/?afg", function(data, status) {
      });
    });
    $("#button1").click(function() {
      $.get("http://192.168.1.22/?kfg", function(data, status) {
      });
    });
    $("#button2").click(function() {
      $.get("http://192.168.1.22/?epag", function(data, status) {
      });
    });
    $("#button3").click(function() {
      $.get("http://192.168.1.22/?rpag", function(data, status) {
      });
    });
    $("#button4").click(function() {
      $.get("http://192.168.1.22/?empg", function(data, status) {
      });
    });
    $("#button5").click(function() {
      $.get("http://192.168.1.22/?rmpg", function(data, status) {
      });
    });
  });
</script>

</head>
```

```

<body>
<div data-role="page" data-theme="b">

<div data-role="header" data-position="fixed" data-theme="a" role="banner"
class="ui-header ui-bar-a ui-header-fixed slidedown">

<a href="prwtosorofos.html" rel="external" class="ui-btn-left ui-alt-icon
ui-nodisc-icon ui-btn ui-icon-carat-l ui-btn-icon-notext ui-corner-all"
data-role="button" role="button">Back</a>

<h1 class="ui-title" role="heading" aria-level="1">Γραφείο, μοτέρ</h1>
</div>

<div data-role="navbar">
    <ul>
        <li><a href="grafeio.html" rel="external" >Αισθητήρες</a></li>
        <li><a href="grafeiomotors.html" rel="external">Μοτέρ</a></li>
    </ul>
</div>

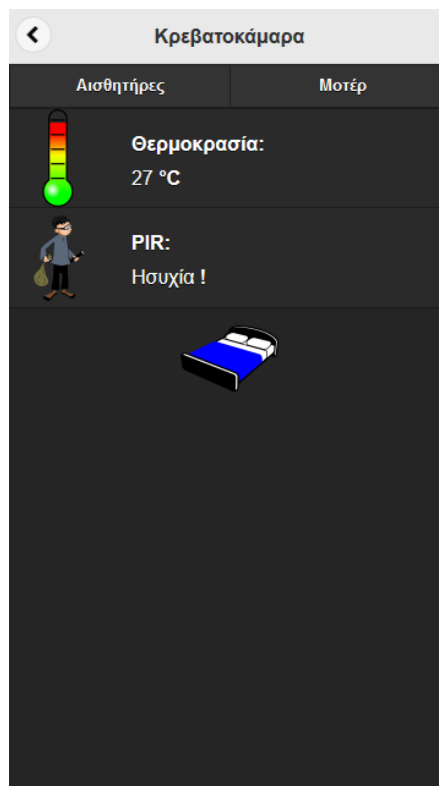
<div data-role="main" class="ui-content">
    <fieldset class="ui-grid-a">
        <div class="ui-block-a"><button class="ui-btn ui-btn-a"
id="button0">Άνοιξε φως</button></div>
        <div class="ui-block-b"><button class="ui-btn ui-btn-b"
id="button1">Κλείσε φως</button></div>
        <div class="ui-block-a"><button class="ui-btn ui-btn-a"
id="button2">Άνοιξε παράθυρο</button></div>
        <div class="ui-block-b"><button class="ui-btn ui-btn-b"
id="button3">Κλείσε παράθυρο</button></div>
        <div class="ui-block-a"><button class="ui-btn ui-btn-a"
id="button4">Άνοιξε μπαλκονόπορτα</button></div>
        <div class="ui-block-b"><button class="ui-btn ui-btn-b"
id="button5">Κλείσε μπαλκονόπορτα</button></div>
    </fieldset>
</div>
<center></center>

</div>

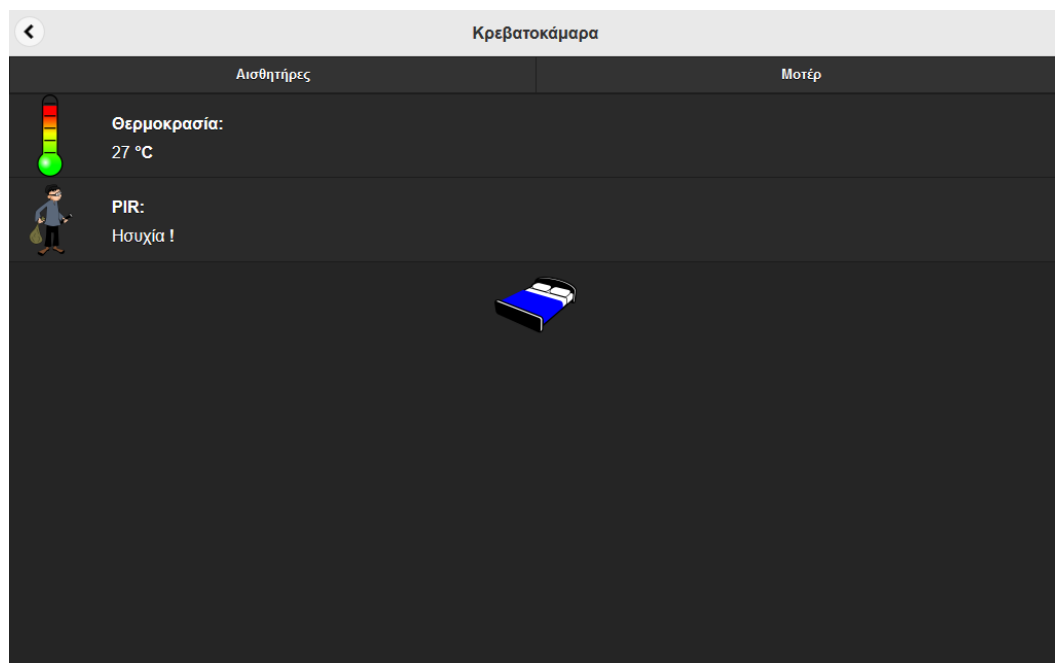
</body>
</html>

```

6.1.31 Στιγμιότυπα οθόνης μετρήσεων Κρεβατοκάμαρας εφαρμογής



Σχήμα 6-31. Διεπαφή μετρήσεων Κρεβατοκάμαρας υλοποιηθείσας εφαρμογής σε οθόνη κινητού τηλεφώνου



Σχήμα 6-32. Διεπαφή μετρήσεων Κρεβατοκάμαρας υλοποιηθείσας εφαρμογής σε οθόνη tablet/υπολογιστή

6.1.32 Κώδικας οθόνης μετρήσεων Κρεβατοκάμαρας εφαρμογής

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="jquery.mobile-1.4.5.min.css" />
    <script src="jquery-1.11.1.min.js"></script>
    <script src="jquery.mobile-1.4.5.min.js"></script>

  <script>

    function autoRefresh()
    {
      window.location = window.location.href;
    }

    setInterval('autoRefresh()', 5000); // 5000ms, 5 defterolepta
  </script>

</head>

<body>
  <div data-role="page" data-theme="b">

    <div data-role="header" data-position="fixed" data-theme="a" role="banner"
      class="ui-header ui-bar-a ui-header-fixed slidedown">
      <a href="prwtosorofos.html" rel="external" class="ui-btn-left ui-alt-icon
        ui-nodisc-icon ui-btn ui-icon-carat-l ui-btn-icon-notext ui-corner-all"
        data-role="button" role="button">Back</a>
      <h1 class="ui-title" role="heading" aria-level="1">Κρεβατοκάμαρα</h1>
    </div>

    <div data-role="navbar">
      <ul>
        <li><a href="krevatokamara.html" rel="external">Αισθητήρες</a></li>
        <li><a href="krevatokamaramotors.html" rel="external">Μοτέρ</a></li>
      </ul>
    </div>
  </div>
```

```
<div data-role="main" class="ui-content">
  <ul data-role="listview">
    <li><h4>Θερμοκρασία:</h4> <span
      id="temp_krevatokamara"></span><b> °C</b></li>
    <li><h4>PIR:</h4> <span
      id="pir_krevatokamara"></span><b> !</b></li>
  </ul>
</div>

<center></center>
</div>

<script>

var xmlhttp, xmlDoc;
xmlhttp = new XMLHttpRequest();
xmlhttp.open("GET", "http://192.168.1.21", false);
xmlhttp.send();
xmlDoc = xmlhttp.responseXML;

document.getElementById("temp_krevatokamara").innerHTML=xmlDoc.
getElementsByTagName("temp_krevatokamara")[0].childNodes[0].nodeValue;

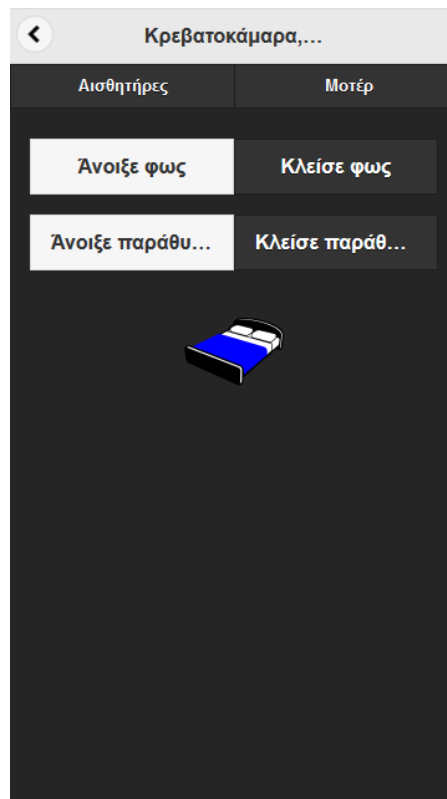
document.getElementById("pir_krevatokamara").innerHTML=xmlDoc.
getElementsByTagName("pir_krevatokamara")[0].childNodes[0].nodeValue;

</script>

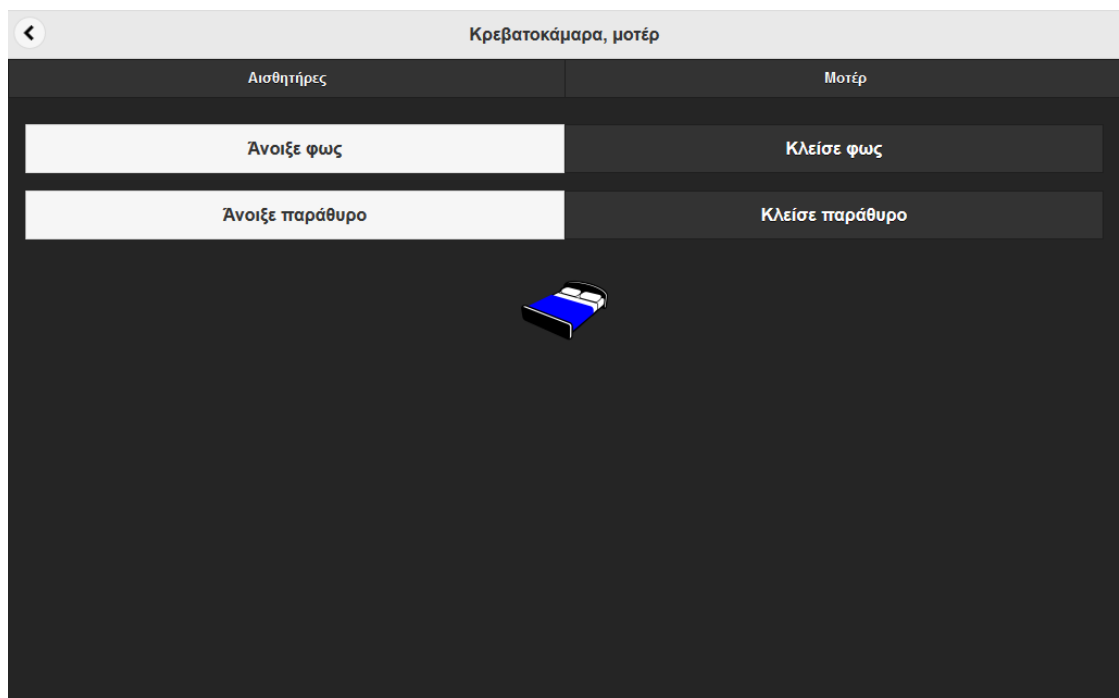
</body>

</html>
```

6.1.33 Στιγμιότυπα οθόνης ενεργοποιητών Κρεβατοκάμαρας εφαρμογής



Σχήμα 6-33. Διεπαφή ενεργοποιητών Κρεβατοκάμαρας εφαρμογής σε οθόνη κινητού τηλεφώνου



Σχήμα 6-34. Διεπαφή ενεργοποιητών Κρεβατοκάμαρας εφαρμογής σε οθόνη tablet/υπολογιστή

6.1.34 Κώδικας οθόνης ενεργοποιητών Κρεβατοκάμαρας εφαρμογής

```

<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="jquery.mobile-1.4.5.min.css" />
    <script src="jquery-1.11.1.min.js"></script>
    <script src="jquery.mobile-1.4.5.min.js"></script>

  <script>
    $(document).ready(function() {
      $("#button0").click(function() {
        $.get("http://192.168.1.22/?afm", function(data, status) {
        });
      });
      $("#button1").click(function() {
        $.get("http://192.168.1.22/?kfm", function(data, status) {
        });
      });
      $("#button2").click(function() {
        $.get("http://192.168.1.22/?epai", function(data, status) {
        });
      });
      $("#button3").click(function() {
        $.get("http://192.168.1.22/?rpai", function(data, status) {
        });
      });
    });
  </script>
</head>

<body>
  <div data-role="page" data-theme="b">

    <div data-role="header" data-position="fixed" data-theme="a" role="banner"
      class="ui-header ui-bar-a ui-header-fixed slidedown">

      <a href="prwtosorofos.html" rel="external" class="ui-btn-left ui-alt-icon
        ui-nodisc-icon ui-btn ui-icon-carat-l ui-btn-icon-notext ui-corner-all"
        data-role="button" role="button">Back</a>

      <h1 class="ui-title" role="heading" aria-level="1">Κρεβατοκάμαρα, μοι έρ</h1>
    </div>
  </div>

```

```
<div data-role="navbar">
  <ul>
    <li><a href="krevatokamara.html" rel="external" >Αισθητήρες</a></li>
    <li><a href="krevatokamaramotors.html" rel="external">Μοτέρ</a></li>
  </ul>
</div>

<div data-role="main" class="ui-content">
  <fieldset class="ui-grid-a">
    <div class="ui-block-a"><button class="ui-btn ui-btn-a"
      id="button0">Άνοιξε φως</button></div>
    <div class="ui-block-b"><button class="ui-btn ui-btn-b"
      id="button1">Κλείσε φως</button></div>
    <div class="ui-block-a"><button class="ui-btn ui-btn-a"
      id="button2">Άνοιξε παράθυρο</button></div>
    <div class="ui-block-b"><button class="ui-btn ui-btn-b"
      id="button3">Κλείσε παράθυρο</button></div>
  </fieldset>
</div>
<center></center>

</div>
</body>

</html>
```


6.2 Κώδικας Android εφαρμογής

6.2.1 Κώδικας Java του mainActivity Android εφαρμογής

```
package com.giwrgos.diplwmatikismarthome;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.Window;
import android.webkit.WebChromeClient;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;

import com.giwrgos.diplwmatikismarthome.R;

public class MainActivity extends Activity {
    WebView webview;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        this.getWindow().requestFeature(Window.FEATURE_PROGRESS);

        setContentView(R.layout.main);
        getWindow().setFeatureInt(Window.FEATURE_PROGRESS,
            Window.PROGRESS_VISIBILITY_ON);

        webview = (WebView) findViewById(R.id.web_engine);
        webview.setWebViewClient(new NewWebViewClient());
        webview.getSettings().setJavaScriptEnabled(true);
        webview.getSettings().setAllowFileAccess(true);
        webview.setScrollBarStyle(WebView.SCROLLBARS_OUTSIDE_OVERLAY);
        //webview.loadUrl("http://tuc.gr"); //για debugging
        webview.loadUrl("file:///android_asset/mainmenu.html");
    }
}
```

```
//mpara proodou
    final Activity MyActivity = this;
    webview.setWebChromeClient(new WebChromeClient() {
        public void onProgressChanged(WebView view, int progress)
        {

            MyActivity.setTitle("Φόρτωση ...");
            MyActivity.setProgress(progress * 100);

            if(progress == 100)
                MyActivity.setTitle(R.string.app_name);
        }
    });

}

//handling urls, uris me mailto pane stin efarmogi email
private class NewWebViewClient extends WebViewClient {
    @Override
    public boolean shouldOverrideUrlLoading(WebView view, String url) {
        if ((url != null && (url.startsWith("mailto:")))) {
            view.getContext().startActivity(
                new Intent(Intent.ACTION_VIEW, Uri.parse(url)));
            return true;
        } else {
            return false;
        }
    }
}

//back button handling
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if ((keyCode == KeyEvent.KEYCODE_BACK) && webview.canGoBack()) {
        webview.goBack();
        return true;
    }
    return super.onKeyDown(keyCode, event);
}
}
```

6.2.2 Κώδικας αρχείου strings.xml Android εφαρμογής

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Smart Home</string>
</resources>
```

6.2.3 Κώδικας αρχείου AndroidManifest.xml Android εφαρμογής

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.giwrgos.diplwmatikismarthome"
    android:versionCode="1"
    android:versionName="1"
    android:installLocation="auto"
    android:hardwareAccelerated="true">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

    <uses-sdk android:minSdkVersion="9"
        android:targetSdkVersion="9"
        android:maxSdkVersion="24" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:allowBackup="false"
        android:hardwareAccelerated="true">

        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:configChanges="orientation|screenSize">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

6.2.4 Κώδικας Layout (main.xml) Android εφαρμογής

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/linearLayout"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <WebView
        android:id="@+id/web_engine"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
</LinearLayout>
```

6.3 Κώδικας Arduino

6.3.1 Κώδικας Arduino σταθμού βάσης αισθητήρων

```
#include <SPI.h>
#include <Ethernet.h> //ethernet i ethernet2 analogws wiznet w5100 i w5500
#include <KSeries.h>

const int lm35PinTrapezaria = A0;
const int lm35PinKouzina = A1;
const int lm35PinSaloni = A2;
const int lm35PinGrafeio = A3;
const int lm35PinKrevatokamara = A4;
const int lm35PinKhpos = A5;
const int hih4030PinTrapezaria = A6;
const int hih4030PinKouzina = A7;
const int hih4030PinSaloni = A8;
const int hih4030PinGrafeio = A9;
const int hih4030PinKhpos = A10;
const int temt6000PinKouzina = A11;
const int temt6000PinSaloni = A12;
const int temt6000PinKhpos = A13;
const int windSensorTempPin = A14;
const int windSensorRVPin = A15;
const int pirPinTrapezaria = 2;
const int pirPinKouzina = 3;
const int pirPinSaloni = 4;
const int pirPinGrafeio = 5;
const int pirPinKrevatokamara = 6;
const int pirPinKhpos = 7;
const int co2rxPin = 12;
const int co2txPin = 13;
const int buzzerPin = 36;

kSeries kSensor(co2rxPin,co2txPin);
byte mac[] = {0x90, 0x90, 0x90, 0x90, 0x90, 0x90 };
IPAddress ip(192, 168, 1, 21);
//IPAddress gateway(192, 168, 1, 1);
//IPAddress subnet(255, 255, 255, 0);

EthernetServer server(80);
String readString;
```

```
void setup()
{
    //Serial.begin(9600);
    initializeAnalogInputs();
    initializeDigitalInputs();
    initializeDigitalOutputs();

    Ethernet.begin(mac, ip);
    //Ethernet.begin(mac, ip, gateway, subnet);
    server.begin();
    //Serial.print("O Server trexei stin IP: ");
    //Serial.println(Ethernet.localIP());
    //Serial.println("Bootup!");
}

void loop()
{
    // listen for incoming clients
    EthernetClient client = server.available();
    if (client)

    {
        Serial.println("new client");

        while (client.connected())
        {
            if (client.available())

            {
                char c = client.read();

                //read char by char HTTP request
                if (readString.length() < 100)

                {
                    readString += c;
                    //Serial.print(c);

                    Serial.write(c);
                }
            }
        }
    }
}
```

```
if (c == '\n') {
    client.println("HTTP/1.1 200 OK");
    client.println("Server: Arduino");
    client.println("Access-Control-Allow-Origin: *");
    client.println("Connection: close");
    client.println("Content-Type: text/xml\r\n");
    client.print("<note>"); //edw arxizei to xml arxeio mou
    client.print("<temp_trapezaria>");
    int tempTrapezaria = calculateTemperature(lm35PinTrapezaria);
    client.print(tempTrapezaria);
    client.print("</temp_trapezaria>");
    client.print("<temp_kouzina>");
    int tempKouzina = calculateTemperature(lm35PinKouzina);
    client.print(tempKouzina);
    client.print("</temp_kouzina>");
    client.print("<temp_saloni>");
    int tempSaloni = calculateTemperature(lm35PinSaloni);
    client.print(tempSaloni);
    client.print("</temp_saloni>");
    client.print("<temp_grafeio>");
    int tempGrafeio = calculateTemperature(lm35PinGrafeio);
    client.print(tempGrafeio);
    client.print("</temp_grafeio>");
    client.print("<temp_krevatokamara>");
    int tempKrevatokamara = calculateTemperature(lm35PinKrevatokamara);
    client.print(tempKrevatokamara);
    client.print("</temp_krevatokamara>");
    client.print("<temp_khpos>");
    int tempKhpos = calculateTemperature(lm35PinKhpos);
    client.print(tempKhpos);
    client.print("</temp_khpos>");
    client.print("<hum_trapezaria>");
    float humTrapezaria = calculateHumidity(tempTrapezaria,
    hih4030PinTrapezaria);
    client.print(humTrapezaria);
    client.print("</hum_trapezaria>");
    client.print("<hum_kouzina>");
    float humKouzina = calculateHumidity(tempKouzina, hih4030PinKouzina);
    client.print(humKouzina);
    client.print("</hum_kouzina>");
    client.print("<hum_saloni>");
```

```
float humSaloni = calculateHumidity(tempSaloni, hih4030PinSaloni);
client.print(humSaloni);
client.print("</hum_saloni>");
client.print("<hum_grafeio>");
float humGrafeio = calculateHumidity(tempGrafeio, hih4030PinGrafeio);
client.print(humGrafeio);
client.print("</hum_grafeio>");
client.print("<hum_khpos>");
float humKhpos = calculateHumidity(tempKhpos, hih4030PinKhpos);
client.print(humKhpos);
client.print("</hum_khpos>");
client.print("<fws_kouzina>");
int fwsKouzina = calculateLux(temt6000PinKouzina);
client.print(fwsKouzina);
client.print("</fws_kouzina>");
client.print("<fws_saloni>");
int fwsSaloni = calculateLux(temt6000PinSaloni);
client.print(fwsSaloni);
client.print("</fws_saloni>");
client.print("<fws_khpos>");
int fwsKhpos = calculateLux(temt6000PinKhpos);
client.print(fwsKhpos);
client.print("</fws_khpos>");
client.print("<wind>");
int aeras = calculateWind(windSensorTempPin, windSensorRVPin);
client.print(aeras);
client.print("</wind>");
client.print("<pir_trapezaria>");
int proximityTrapezaria = digitalRead(pirPinTrapezaria);
if (proximityTrapezaria == LOW)
{
    client.print("Ήσυχ(α)");
}
else
{
    client.print("Κλέφτης");
}
client.print("</pir_trapezaria>");
client.print("<pir_kouzina>");
int proximityKouzina = digitalRead(pirPinKouzina);
if (proximityKouzina == LOW)
```



```
        {
            client.print("Ήσυχία");
        }
        else
        {
            client.print("Κλέφτης");
        }
    }
    client.print("</pir_kouzina>");
    client.print("<pir_saloni>");
    int proximitySaloni = digitalRead(pirPinSaloni);
    if (proximitySaloni == LOW)
    {
        client.print("Ήσυχία");
    }
    else
    {
        client.print("Κλέφτης");
    }
    client.print("</pir_saloni>");
    client.print("<pir_grafeio>");
    int proximityGrafeio = digitalRead(pirPinGrafeio);
    if (proximityGrafeio == LOW)
    {
        client.print("Ήσυχία");
    }
    else
    {
        client.print("Κλέφτης");
    }
    client.print("</pir_grafeio>");
    client.print("<pir_krevatokamara>");
    int proximityKrevatokamara = digitalRead(pirPinKrevatokamara);
    if (proximityKrevatokamara == LOW)
    {
        client.print("Ήσυχία");
    }
    else
    {
        client.print("Κλέφτης");
    }
    client.print("</pir_krevatokamara>");
```

```
client.print("<pir_khpos>");
int proximityKhpos = digitalRead(pirPinKhpos);
if (proximityKhpos == LOW)
{
    client.print("Ηουχία");
    digitalWrite(buzzerPin, LOW);
}
else
{
    client.print("Κλέφτης");
    digitalWrite(buzzerPin, HIGH);
}
client.print("</pir_khpos>");
client.print("<co2>");
double valCO2 = kSensor.getCO2('P');
client.print(valCO2);
client.print("</co2>");
client.print("</note>");

//gia statherotita, 1ms
delay(1);

//stamataei o client
client.stop();

//svinoume ta perioxomena, gia tin epomeni anagnwsi
readString = "";

//ena mikro delay gia ton browser
delay(1);

client.stop();
//Serial.println("aposindethike, gia debugging");

}
}
}
}
}
```

```
void initializeAnalogInputs() {
    pinMode(A0, INPUT);
    pinMode(A1, INPUT);
    pinMode(A2, INPUT);
    pinMode(A3, INPUT);
    pinMode(A4, INPUT);
    pinMode(A5, INPUT);
    pinMode(A6, INPUT);
    pinMode(A7, INPUT);
    pinMode(A8, INPUT);
    pinMode(A9, INPUT);
    pinMode(A10, INPUT);
    pinMode(A11, INPUT);
    pinMode(A12, INPUT);
    pinMode(A13, INPUT);
    pinMode(A14, INPUT);
    pinMode(A15, INPUT);
}

void initializeDigitalInputs() {
    for (int i = 2; i <= 7; i++) {
        pinMode(i, INPUT_PULLUP);
    }
}

void initializeDigitalOutputs() {
    for (int i = 14; i <= 45; i++) {
        pinMode(i, OUTPUT);
    }
}

int calculateTemperature(int lm35Pin) {
    int sum=0;
    int temp;
    for (int j=0;j<=10;j++){
        temp = (5.0 * analogRead(lm35Pin) * 100.0) / 1024; //temp se kelsiou
        sum+=temp;
    }
    temp=sum/10; //mesos oros 10 metrisewn, leitoyrgia psifiakou filtroyu
    return temp;
}
```

```

int calculateLux(int temt6000Pin) {
    int illuminance = analogRead(temt6000Pin) * 0.9765625; //fws se lux
    return illuminance;
}

float calculateHumidity(int tempratureCelsius, int hih4030Pin) {
    //igrasia, ipologismos vasei diorthwsis sfalmatos thermokrasias
    int HIH4030_Value = analogRead(hih4030Pin);
    float voltage = HIH4030_Value / 1023. * 5.0;
    // convert to voltage value
    float sensorRH = 161.0 * voltage / 5.0 - 25.8;
    //to 5 antistixei sto supply voltage tou aisthitira
    float trueRH = sensorRH / (1.0546 - 0.0026 * tempratureCelsius);
    //diorthwsi thermokrasias
    return trueRH;
}

float calculateWind(int windSensorTempPinInput, int windSensorRVPinInput) {
    const float zeroWindAdjustment = .4; //gia calibration tou aisthitira
    int TMP_Therm_ADunits; //i thermokrasia apo to thermistor tou windsensor
    float RV_Wind_ADunits; //i exodos apo ton aisthitira thermokrasias
    float RV_Wind_Volts;
    int TempCtimes100;
    float zeroWind_ADunits;
    float zeroWind_volts;
    float WindSpeed_MPH;
    float WindSpeed_KPH;

    TMP_Therm_ADunits = analogRead(windSensorTempPinInput);
    RV_Wind_ADunits = analogRead(windSensorRVPinInput);
    RV_Wind_Volts = (RV_Wind_ADunits * 0.0048828125);
    TempCtimes100 = (0.005 * ((float)TMP_Therm_ADunits *
    (float)TMP_Therm_ADunits)) - (16.862 * (float)TMP_Therm_ADunits) + 9075.4;
    zeroWind_ADunits = -0.0006 * ((float)TMP_Therm_ADunits *
    (float)TMP_Therm_ADunits) + 1.0727 * (float)TMP_Therm_ADunits + 47.172;
    zeroWind_volts = (zeroWind_ADunits * 0.0048828125) - zeroWindAdjustment;
    WindSpeed_MPH = pow(((RV_Wind_Volts - zeroWind_volts) / .2300) , 2.7265);
    WindSpeed_KPH = WindSpeed_MPH * 1.60934;
    //metatropi apo milia tin wra se kilioetra tin wra
    return WindSpeed_KPH;
}

```

6.3.2 Κώδικας Arduino σταθμού βάσης ενεργοποιητών φώτων/μοτέρ

```
#include <SPI.h>
#include <Ethernet2.h> //ethernet i ethernet2 analogws wiznet w5100 i w5500

byte mac[] = {0x90, 0x90, 0x90, 0x10, 0x20, 0x30 };
IPAddress ip(192, 168, 1, 22);
//IPAddress gateway(192, 168, 1, 1);
//IPAddress subnet(255, 255, 255, 0);

EthernetServer server(80);
String readString;

void setup()
{
    //Serial.begin(9600);
    initializeAnalogOutputs();
    initializeDigitalOutputs();
    initializeallOutputsLow();

    Ethernet.begin(mac, ip);
    //Ethernet.begin(mac, ip, gateway, subnet);
    server.begin();
    //Serial.print("O Server trexei stin IP: ");
    //Serial.println(Ethernet.localIP());
    //Serial.println("Bootup!");
}

void loop()
{
    EthernetClient client = server.available();
    if (client)

    {
        Serial.println("new client");

        while (client.connected())
        {
            if (client.available())

            {
                char c = client.read();
```

```
//diavazoume to http request
if (readString.length() < 100)

{
  readString += c;
  //Serial.print(c);

  Serial.write(c);
  if (c == '\n') {
    client.println("HTTP/1.1 200 OK");
    client.println("Server: Arduino");
    client.println("Access-Control-Allow-Origin: *");
    client.println("Connection: close");
    client.println("Content-Type: text/xml\r\n");
    //dhmiourgia arxeiou xml, gia vevaiwsi epikoinwnias
    client.println("<up>");
    client.println("</up>");
    delay(1);

    //stopping
    client.stop();

    if (readString.indexOf("?epat") > 0) //parathiro trapezarias
    {
      digitalWrite(0, HIGH);
      digitalWrite(1, LOW);
      delay(2000);
      digitalWrite(0, LOW);
      digitalWrite(1, LOW);
    }

    else if (readString.indexOf("?rpat") > 0) //parathiro trapezarias
    {
      digitalWrite(0, LOW);
      digitalWrite(1, HIGH);
      delay(2000);
      digitalWrite(0, LOW);
      digitalWrite(1, LOW);
    }
  }
}
```

```
else if (readString.indexOf("?epak") > 0) //parathrio kouzinas
{
    digitalWrite(2, HIGH);
    digitalWrite(3, LOW);
    delay(2000);
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
}
else if (readString.indexOf("?rpak") > 0) //parathiro kouzinas
{
    digitalWrite(2, LOW);
    digitalWrite(3, HIGH);
    delay(2000);
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
}
else if (readString.indexOf("?empk") > 0) //mpalkonoporta kouzinas
{
    digitalWrite(4, HIGH);
    digitalWrite(5, LOW);
    delay(2000);
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
}
else if (readString.indexOf("?rmpk") > 0) //mpalkonoporta kouzinas
{
    digitalWrite(4, LOW);
    digitalWrite(5, HIGH);
    delay(2000);
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
}
else if (readString.indexOf("?empns") > 0) //n. mpaklon. saloniou
{
    digitalWrite(6, HIGH);
    digitalWrite(7, LOW);
    delay(2000);
    digitalWrite(6, LOW);
    digitalWrite(7, LOW);
}
```

```
else if (readString.indexOf("?rmpns") > 0) //n. mpaklon. saloniou
{
    digitalWrite(6, LOW);
    digitalWrite(7, HIGH);
    delay(2000);
    digitalWrite(6, LOW);
    digitalWrite(7, LOW);
}
else if (readString.indexOf("?empvs") > 0) //v. mpaklon. saloniou
{
    digitalWrite(8, HIGH);
    digitalWrite(9, LOW);
    delay(2000);
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
}
else if (readString.indexOf("?rmpvs") > 0) //v. mpaklon. saloniou
{
    digitalWrite(8, LOW);
    digitalWrite(9, HIGH);
    delay(2000);
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
}
else if (readString.indexOf("?epag") > 0) //parathiro grafeiou
{
    digitalWrite(A0, HIGH);
    digitalWrite(A1, LOW);
    delay(2000);
    digitalWrite(A0, LOW);
    digitalWrite(A1, LOW);
}
else if (readString.indexOf("?rpag") > 0) //parathiro grafeiou
{
    digitalWrite(A0, LOW);
    digitalWrite(A1, HIGH);
    delay(2000);
    digitalWrite(A0, LOW);
    digitalWrite(A1, LOW);
}
```



```
else if (readString.indexOf("?empg") > 0) //mplalkonoporta grafeiou
{
    digitalWrite(A2, HIGH);
    digitalWrite(A3, LOW);
    delay(2000);
    digitalWrite(A2, LOW);
    digitalWrite(A3, LOW);
}
else if (readString.indexOf("?rmpg") > 0) //mpalkonoporta grafeiou
{
    digitalWrite(A2, LOW);
    digitalWrite(A3, HIGH);
    delay(2000);
    digitalWrite(A2, LOW);
    digitalWrite(A3, LOW);
}
else if (readString.indexOf("?epai") > 0) //parathiro ipnodwmatioy
{
    digitalWrite(A4, HIGH);
    digitalWrite(A5, LOW);
    delay(2000);
    digitalWrite(A4, LOW);
    digitalWrite(A5, LOW);
}
else if (readString.indexOf("?rpai") > 0) //parathiro ipnodwmatioy
{
    digitalWrite(A4, LOW);
    digitalWrite(A5, HIGH);
    delay(2000);
    digitalWrite(A4, LOW);
    digitalWrite(A5, LOW);
}
//trapezaria
else if (readString.indexOf("?aft") > 0) //fws trapezaria on
{
    digitalWrite(14, HIGH);
}
else if (readString.indexOf("?kft") > 0) //fws trapezaria off
{
    digitalWrite(14, LOW);
}
```

```
//kouzina
else if (readString.indexOf("?afk") > 0) //fws kouzina on
{
    digitalWrite(15, HIGH);
}
else if (readString.indexOf("?kfk") > 0) //fws kouzina off
{
    digitalWrite(15, LOW);
}

//saloni
else if (readString.indexOf("?afs") > 0) //fws saloni on
{
    digitalWrite(16, HIGH);
}
else if (readString.indexOf("?kfs") > 0) //fws saloni off
{
    digitalWrite(16, LOW);
}

//grafeio
else if (readString.indexOf("?afg") > 0) //fws grafeio on
{
    digitalWrite(17, HIGH);
}
else if (readString.indexOf("?kfg") > 0) //fws grafeio off
{
    digitalWrite(17, LOW);
}

//krevatokamara
else if (readString.indexOf("?afm") > 0) //fws krevatokamara on
{
    digitalWrite(18, HIGH);
}
else if (readString.indexOf("?kfm") > 0) //fws krevatokamara off
{
    digitalWrite(18, LOW);
}
```

```
        //khpos
        else if (readString.indexOf("?afp") > 0) //fws khpou on
        {
            digitalWrite(19, HIGH);
        }
        else if (readString.indexOf("?kfp") > 0) //fws khpou off
        {
            digitalWrite(19, LOW);
        }

        //svisimo gia epomeni anagnwsi
        readString = "";

        delay(1);
        client.stop();
    }
}

}

}

}

}

void initializeAnalogOutputs() {
    pinMode(A0, OUTPUT);
    pinMode(A1, OUTPUT);
    pinMode(A2, OUTPUT);
    pinMode(A3, OUTPUT);
    pinMode(A4, OUTPUT);
    pinMode(A5, OUTPUT);
}

void initializeDigitalOutputs() {
    for (int i = 0; i <= 19; i++) {
        pinMode(i, OUTPUT);
    }
}

void initializeallOutputsLow() { //diasfalisis arxikis katastasis LOW
    for (int i = 0; i <= 19; i++) {
        digitalWrite(i, LOW);
    }
}
```

6.3.3 Κώδικας Arduino δοκιμής λειτουργίας κινητήρων και φώτων

```
void setup() {  
    initializeDigitalOutputs();  
    initializeallOutputsLow();  
}  
  
void loop() {  
    checkMotor1(); //parathiro trapezarias, 1o L923, 1o kanali  
    checkMotor2(); //parathiro kouzinas, 1o L293, 2o kanali  
    checkMotor3(); //mpalkonoporta kouzinas, 2o L293, 1o kanali  
    checkMotor4(); //notio mpalkoni saloniou, 2o L293, 2o kanali  
    checkMotor5(); //vorio mpalkoni saloniou, 3o L293, 1o kanali  
    checkMotor6(); //parathiro grafeiou, 3o L293, 2o kanali  
    checkMotor7(); //mpalkonoporta grafeiou, 4o L293, 1o kanali  
    checkMotor8(); //parathiro krevatokamaras, 4o L293, 2o kanali  
    checkLights(); //energopoihsh kai apenergopoihsh olwn twv fwtwn  
}  
  
void checkMotor1() { //parathiro trapezarias  
    digitalWrite(0, HIGH);  
    digitalWrite(1, LOW);  
    delay(1000);  
    digitalWrite(0, LOW);  
    digitalWrite(1, HIGH);  
    delay(1000);  
    digitalWrite(0, LOW);  
    digitalWrite(0, LOW);  
    delay(1000);  
}  
  
void checkMotor2() { //parathiro kouzinas  
    digitalWrite(2, HIGH);  
    digitalWrite(3, LOW);  
    delay(1000);  
    digitalWrite(2, LOW);  
    digitalWrite(3, HIGH);  
    delay(1000);  
    digitalWrite(2, LOW);  
    digitalWrite(3, LOW);  
    delay(1000);  
}
```

```
void checkMotor3() { //mpalkonoporta kouzinas
    digitalWrite(4,HIGH);
    digitalWrite(5,LOW);
    delay(1000);
    digitalWrite(4,LOW);
    digitalWrite(5,HIGH);
    delay(1000);
    digitalWrite(4,LOW);
    digitalWrite(5,LOW);
    delay(1000);
}
```

```
void checkMotor4() { //notio mpalkoni saloniou
    digitalWrite(6,HIGH);
    digitalWrite(7,LOW);
    delay(1000);
    digitalWrite(6,LOW);
    digitalWrite(7,HIGH);
    delay(1000);
    digitalWrite(6,LOW);
    digitalWrite(7,LOW);
    delay(1000);
}
```

```
void checkMotor5() { //vorio mpalkoni saloniou
    digitalWrite(8,HIGH);
    digitalWrite(9,LOW);
    delay(1000);
    digitalWrite(8,LOW);
    digitalWrite(9,HIGH);
    delay(1000);
    digitalWrite(8,LOW);
    digitalWrite(9,LOW);
    delay(1000);
}
```

```
void checkMotor6() { //parathiro grafeiou
    digitalWrite(A0,HIGH);
    digitalWrite(A1,LOW);
    delay(1000);
    digitalWrite(A0,LOW);
    digitalWrite(A1,HIGH);
    delay(1000);
    digitalWrite(A0,LOW);
    digitalWrite(A1,LOW);
    delay(1000);
}

void checkMotor7() { //mpalkonoporta grafeiou
    digitalWrite(A2,HIGH);
    digitalWrite(A3,LOW);
    delay(1000);
    digitalWrite(A2,LOW);
    digitalWrite(A3,HIGH);
    delay(1000);
    digitalWrite(A2,LOW);
    digitalWrite(A3,LOW);
    delay(1000);
}

void checkMotor8() { //parathiro krevatokamaras
    digitalWrite(A4,HIGH);
    digitalWrite(A5,LOW);
    delay(1000);
    digitalWrite(A4,LOW);
    digitalWrite(A5,HIGH);
    delay(1000);
    digitalWrite(A4,LOW);
    digitalWrite(A5,LOW);
    delay(1000);
}
```

```
void initializeDigitalOutputs() { //ola ta pins 0 ews 19 kai A0 ws A5 exodoi
    for (int i = 0; i <= 19; i++) {
        pinMode(i, OUTPUT);
    }
    pinMode(A0,OUTPUT);
    pinMode(A1,OUTPUT);
    pinMode(A2,OUTPUT);
    pinMode(A3,OUTPUT);
    pinMode(A4,OUTPUT);
    pinMode(A5,OUTPUT);
}

void initializeallOutputsLow() { //exasfalizw arxiki katastasi LOW
    for (int i = 0; i <= 19; i++) {
        digitalWrite(i,LOW);
    }
    pinMode(A0,LOW);
    pinMode(A1,LOW);
    pinMode(A2,LOW);
    pinMode(A3,LOW);
    pinMode(A4,LOW);
    pinMode(A5,LOW);
}

void checkLights() { //elegxos fwtwn
    for (int j = 14; j <= 19; j++) {
        digitalWrite(j,HIGH);
        delay(2000);
        digitalWrite(j,LOW);
        delay(2000);
    }
}
```

6.4 Ορισμοί

Η έννοια Έξυπνο Σπίτι, που χρησιμοποιήθηκε στην παρούσα εργασία, αναφέρεται στον εμπλουτισμό ενός περιβάλλοντος διαβίωσης με τεχνολογία, με σκοπό να προσφέρει βελτιωμένη υποστήριξη στις συνήθειες των κατοίκων του και, κατά συνέπεια, μία βελτιωμένη ποιότητα ζωής.

Ο όρος Έξυπνο Σπίτι δημιουργεί προσδοκίες ενός περιβάλλοντος το οποίο είναι ικανό να αντιδρά έξυπνα, προλαμβάνοντας, "μαντεύοντας" και παίρνοντας αποφάσεις με σημάδια αυτονομίας.

Κατοικίες που χρησιμοποιούν ολοκληρωμένα συστήματα επικοινωνίας για να παρακολουθούν και να διαχειρίζονται την απόδοση του σπιτιού, και να υποστηρίζουν τις επιλογές του τρόπου ζωής των ενοίκων.

Όλα τα συστήματα ευφυών κατοικιών αποτελούνται από τέσσερα βασικά συστατικά, τα οποία χρησιμοποιήθηκαν στην παρούσα εργασία. Οι ορισμοί τους δίνονται παρακάτω:

1. Υποδομή : Ένα μέσο επικοινωνίας για τη μετάδοση της ψηφιακής πληροφορίας. Τα τρέχοντα μέσα περιλαμβάνουν χάλικινα καλώδια, οπτικές ίνες, ασύρματη επικοινωνία και γραμμές ρεύματος, που χρησιμοποιούν τα υφιστάμενα δίκτυα ηλεκτρικής ενέργειας.
2. Αισθητήρες : Χρησιμοποιούνται για να παρακολουθούν τις συνθήκες εντός του σπιτιού, για παράδειγμα αισθητήρες θερμοκρασίας και αισθητήρες υγρασίας.
3. Ενεργοποιητές : Είναι συσκευές που εκτελούν οδηγίες από εφαρμογές. Παραδείγματα περιλαμβάνουν ενεργοποιητές που ανοίγουν παράθυρα, κλείνουν βαλβίδες καλοριφέρ κτλ.
4. Εφαρμογές : Οι εφαρμογές επιτρέπουν στους ανθρώπους να ελέγχουν τα έξυπνα συστήματα στα σπίτια τους. Έχουν τη μορφή αλγορίθμων κωδικοποιημένων μέσα σε προγράμματα. Επιτρέπουν στους χρήστες της κατοικίας να ρυθμίσουν παραμέτρους όπως οι μέγιστες και οι ελάχιστες θερμοκρασίες, τις οποίες οι αισθητήρες και οι ενεργοποιητές θα παρακολουθούν και στις οποίες θα ανταποκρίνονται για να επιτύχουν την απόδοση των ρυθμίσεων του χρήστη.